Shanghai Fudan Microelectronics Group Company Limited

复旦微电子

FM33FG0A 系列
车用 MCU 芯片

产品说明书

2023.8

本资料是为了让用户根据用途选择合适的上海复旦微电子集团股份有限公司（以下简称复旦微电子）的产品而提供的参考资料，不转让属于复旦微电子或者第三者所有的知识产权以及其他权利的许可。
在使用本资料所记载的信息最终做出有关信息和产品是否适用的判断前，请您务必将所有信息作为一个整体系统来进行评价。
采购方对于选择与使用本文描述的复旦微电子的产品和服务全权负责，复旦微电子不承担采购方选择与使用本文描述的产品和服务的责任。除非以书面形式明确地认可，复旦微电子的产品不推荐、不授权、不担保用于包括军事、航空、航天、救生及生命维持系统在内的，由于失效或故障可能导致人身伤亡、严重的财产或环境损失的产品或系统中。
未经复旦微电子的许可，不得翻印或者复制全部或部分本资料的内容。
今后日常的产品更新会在适当的时候发布，恕不另行通知。在购买本资料所记载的产品时，请预先向复旦微电子在当地的销售办事处确认最新信息，并请您通过各种方式关注复旦微电子公布的信息，包括复旦微电子的网站(http://www.fmsh.com/)。
如果您需要了解有关本资料所记载的信息或产品的详情，请与上海复旦微电子集团股份有限公司在当地的销售办事处联系。

商标
上海复旦微电子集团股份有限公司的公司名称、徽标以及“复旦”徽标均为上海复旦微电子集团股份有限公司及其分公司在中国的商标或注册商标。

上海复旦微电子集团股份有限公司在中国发布，版权所有。

章节列表

章节列表 **3**

表目录 **26**

图目录 **28**

1 产品综述 **37**

1.1 概述 37
1.2 芯片结构框图 39
1.3 产品型号列表 40

2 引脚和封装 **41**

2.1 封装和引脚排列 41
2.1.1 LQFP144 封装图（FM33FG0x14A） 41
2.1.2 LQFP100 封装图（FM33FG0x10A） 42
2.1.3 LQFP64 封装图（FM33FG0Ax6A） 43
2.1.4 PA 引脚组数字外设功能 44
2.1.5 PB 引脚组数字外设功能 44
2.1.6 PC 引脚组数字外设功能 45
2.1.7 PD 引脚组数字外设功能 45
2.1.8 PE 引脚组数字外设功能 46
2.1.9 PF 引脚组数字外设功能 46
2.1.10 PG 引脚组数字外设功能 46
2.1.11 PH 引脚组数字外设功能 47
2.1.12 PI 引脚组数字外设功能 47
2.1.13 封装尺寸图 48
2.2 焊接安装说明 52
2.3 MSL 等级 54

3 电参数 **55**

3.1 参数说明 55
3.2 测试标准 55
3.3 极限参数 55
3.4 性能参数 56
3.4.1 典型工作条件 56
3.4.2 功耗参数 56
3.4.3 复位和电源监控 57
3.4.4 内部基准电压 58
3.4.5 低功耗模式唤醒时间 59
3.4.6 外部时钟源特性 60
3.4.7 内部时钟源特性 61
3.4.8 PLL 特性 62
3.4.9 ADC 特性 63
3.4.10 DAC 特性 66
3.4.11 温度传感器 67
3.4.12 模拟比较器特性 68
3.4.13 Flash 存储器特性 69
3.4.14 GPIO 特性 69

4 总线与存储 **71**

4.1 系统总线 71

4.2 存储空间分配 73
4.2.1 概述 73
4.2.2 外设模块寄存器地址分配 74
4.3 RAM 75
4.3.1 概述 75
4.4 CODE FLASH 76
4.4.1 概述 76
4.4.2 特殊信息扇区说明 76
4.4.3 Flash 编程 79
4.4.4 Flash 的内容保护 82
4.5 DATA FLASH 85
4.5.1 概述 85
4.5.2 Debug 接口保护（DBRDP_D） 85
4.5.3 data Flash 编程 86
4.6 寄存器 89
4.6.1 Flash 读取控制寄存器（FLS_RDCR） 90
4.6.2 用户配置字寄存器（FLS_OPTBR） 91
4.6.3 ACLOCK 寄存器 1（FLS_ACLOCK1） 92
4.6.4 ACLOCK 寄存器 2（FLS_ACLOCK2） 92
4.6.5 ACLOCK 寄存器 3（FLS_ACLOCK3） 93
4.6.6 ACLOCK 寄存器 4（FLS_ACLOCK4） 93
4.6.7 CFlash 擦写控制寄存器（CFLS_EPCR） 94
4.6.8 CFlash Key 输入寄存器（CFLS_KEY） 95
4.6.9 CFlash 中断使能寄存器（CFLS_IER） 95
4.6.10 CFlash 标志寄存器（CFLS_ISR） 96
4.6.11 Dflash 擦写控制寄存器（DFLS_EPCR） 96
4.6.12 Dflash Key 输入寄存器（DFLS_KEY） 97
4.6.13 Dflash 中断使能寄存器（DFLS_IER） 98
4.6.14 Dflash 标志寄存器（DFLS_ISR） 98
4.6.15 Code Flash ECC 寄存器（CFLS_ECCR） 99
4.6.16 Data Flash ECC 寄存器（DFLS_ECCR） 100
4.6.17 RAM ECC 寄存器（RAM_ECCR） 101
4.6.18 RAM 原始数据寄存器 0(RAM_RAW0) 102
4.6.19 RAM 原始数据寄存器 1(RAM_RAW1) 102
4.6.20 DFlash 原始数据寄存器 0(DFLS_RAW0) 102
4.6.21 DFlash 原始数据寄存器 1(DLFS_RAW1) 103
4.6.22 HSRAM1 配置寄存器（HSRAM1_CFGR） 103
4.6.23 HSRAM2 配置寄存器（HSRAM2_CFGR） 104
4.6.24 后门访问验证寄存器（FLS_BDVR） 104
4.6.25 后门访问密钥输入寄存器 0（FLS_BDCR0） 105
4.6.26 后门访问密钥输入寄存器 1（FLS_BDCR1） 105
4.6.27 后门访问密钥状态寄存器（FLS_BDSR） 106
5 电源管理单元（PMU） 107
5.1 芯片工作电源 107
5.1.1 电源域划分 107
5.1.2 电源结构图 108
5.1.3 ADC 基准电压 108
5.1.4 片内快速建立基准源（AVREF） 109
5.2 功耗模式 109
5.2.1 概述 109
5.2.2 功耗模式与系统频率 110
5.2.3 Active 模式 110
5.2.4 SLEEP 模式 110

5.2.5 DEEPSLEEP 模式 111
5.3 唤醒源 111
5.3.1 CANFD 异步唤醒 112
5.4 休眠唤醒后的时钟控制 113
5.5 寄存器 114
5.5.1 低功耗控制寄存器（PMU_CR） 114
5.5.2 唤醒时间控制寄存器（PMU_WKTR） 115
5.5.3 唤醒源标志查询寄存器（PMU_WKFR） 116
5.5.4 PMU 中断使能寄存器（PMU_IER） 117
5.5.5 PMU 中断标志寄存器（PMU_ISR） 117
5.5.6 模拟 BUFFER 控制寄存器（PMU_BUFCR） 118
5.5.7 PTAT 控制寄存器（PMU_PTAT_CR） 118
5.5.8 CANFD 唤醒使能寄存器 119
5.5.9 CANFD 唤醒中断寄存器 120

6 处理器（CPU） 121

6.1 概述 121
6.1.1 处理器配置 121
6.2 寄存器 122
6.3 异常和中断 123
6.3.1 中断向量表 123
6.3.2 中断优先级 123
6.3.3 错误处理 123
6.3.4 锁定（Lockup） 124
6.4 MPU 124
6.4.1 MPU 寄存器 125
6.5 调试特性 128
6.5.1 调试功能引脚 128
6.5.2 调试状态下的看门狗控制 129
6.5.3 DEBUG 的复位 129

7 中断向量表 130

8 复位管理单元（RMU） 132

8.1 概述 132
8.2 模块框图 133
8.3 上下电复位 133
8.4 软件复位 135
8.5 NRST 引脚复位 135
8.6 LOC 复位 135
8.7 LOL 复位 136
8.8 寄存器 138
8.8.1 PDR 配置寄存器（RMU_PDRCR） 138
8.8.2 BOR 配置寄存器（RMU_BORCR） 139
8.8.3 复位配置寄存器（RMU_LKPCR） 139
8.8.4 软件复位寄存器（RMU_SOFTRST） 140
8.8.5 复位标志寄存器（RMU_RSTFR） 141
8.8.6 外设复位使能寄存器（RMU_PRSTEN） 142
8.8.7 AHB 外设复位寄存器（RMU_AHBRSTCR） 142
8.8.8 APB 外设复位寄存器 1（RMU_APBRSTCR1） 143
8.8.9 APB 外设复位寄存器 2（RMU_APBRSTCR2） 145

9 独立看门狗（IWDT） 147

9.1 概述 147

9.2 结构框图 147
9.3 IWDT 功能描述 148
9.4 IWDT 窗口功能 148
9.5 IWDT 冻结 149
9.6 寄存器 150
9.6.1 IWDT 清除寄存器（IWDT_SERV） 150
9.6.2 IWDT 配置寄存器（IWDT_CR） 150
9.6.3 IWDT 计数值寄存器（IWDT_CNT） 151
9.6.4 IWDT 窗口寄存器（IWDT_WIN） 152
9.6.5 IWDT 中断使能寄存器（IWDT_IER） 152
9.6.6 IWDT 中断标志寄存器（IWDT_ISR） 152

10 窗口看门狗（WWDT） 154

10.1 功能描述 154
10.2 结构框图 154
10.3 WWDT 工作方式 154
10.4 寄存器 157
10.4.1 WWDT 控制寄存器（WWDT_CR） 157
10.4.2 WWDT 配置寄存器（WWDT_CFGR） 157
10.4.3 WWDT 计数寄存器（WWDT_CNT） 158
10.4.4 WWDT 中断使能寄存器（WWDT_IER） 158
10.4.5 WWDT 中断标志寄存器（WWDT_ISR） 159
10.4.6 WWDT 预分频寄存器（WWDT_PSC） 159

11 时钟管理单元（CMU） 161

11.1 概述 161
11.2 时钟架构 162
11.2.1 时钟树 162
11.2.2 SYSCLK 切换说明 163
11.2.3 时钟安全性 163
11.2.4 主要时钟说明 163
11.2.5 外设模块的总线时钟和工作时钟 164
11.2.6 休眠模式下的外设时钟 165
11.2.7 LSCLK 切换逻辑 166
11.2.8 RTC 时钟 166
11.3 高频晶体振荡电路(XTHF) 167
11.3.1 概述 167
11.3.2 工作方式 167
11.3.3 停振检测（HFDET） 167
11.3.4 XTHF 启动过程 167
11.4 低频晶体振荡电路(XTLF) 168
11.4.1 概述 168
11.4.2 工作方式 168
11.4.3 停振检测（LFDET） 168
11.4.4 XTLF 启动过程 168
11.5 低功耗低频环振(RCLP) 169
11.5.1 概述 169
11.6 锁相环(PLL) 170
11.6.1 概述 170
11.6.2 锁定检测 170
11.6.3 应用注意 170
11.7 低功耗模式下的时钟源 170
11.8 休眠唤醒的时钟处理 170

11.9 寄存器 171
11.9.1 系统时钟配置寄存器（CMU_SYSCLKCR） 172
11.9.2 RCHF 时钟控制寄存器（CMU_RCHCR） 173
11.9.3 RCHF 调校寄存器（CMU_RCHFTR） 174
11.9.4 PLL 控制寄存器（CMU_PLLCR） 174
11.9.5 RCLP 调校寄存器（CMU_RCLPTR） 175
11.9.6 XTLF 控制寄存器（CMU_XTLFCR） 176
11.9.7 LSCLK 选择寄存器（CMU_ LSCLKSEL） 177
11.9.8 XTHF 控制寄存器（CMU_XTHFCR） 177
11.9.9 CMU 中断使能寄存器（CMU_IER） 178
11.9.10 CMU 中断标志寄存器（CMU_ISR） 179
11.9.11 外设总线时钟控制寄存器 1（CMU_ PCLKCR1） 180
11.9.12 外设总线时钟控制寄存器 2（CMU_ PCLKCR2） 180
11.9.13 外设总线时钟控制寄存器 3（CMU_ PCLKCR3） 181
11.9.14 外设总线时钟控制寄存器 4（CMU_ PCLKCR4） 182
11.9.15 外设工作时钟配置寄存器 1（CMU_OPCCR1） 183
11.9.16 外设工作时钟配置寄存器 2（CMU_OPCCR2） 185
11.9.17 外设工作时钟使能寄存器（CMU_OPCCR3） 186
11.9.18 AHB Master 控制寄存器（CMU_AHBMCR） 187
11.9.19 CANFD 时钟配置寄存器（CMU_CFDCR） 187
11.9.20 CANFD 时钟使能寄存器（CMU_CFDER） 188

12 时钟监控（CLM） 190

12.1 概述 190
12.2 结构框图 190
12.3 功能描述 190
12.3.1 工作模式 190
12.3.2 使能与关闭 191
12.3.3 时钟频率监控 191
12.3.4 超时检测 194
12.4 寄存器 194
12.4.1 时钟监控控制寄存器（CLM_CR） 194
12.4.2 时钟监控配置寄存器（CLM_CFGR） 195
12.4.3 时钟监控计数值寄存器（CLM_CNTR） 196
12.4.4 时钟监控中断标志寄存器（CLM_ISR） 196
12.4.5 时钟监控比较高阈值寄存器（CLM_CMPH） 197
12.4.6 时钟监控比较低阈值寄存器（CLM_CMPL） 197

13 电源电压监测（SVD） 199

13.1 概述 199
13.2 结构框图 199
13.3 引脚定义 200
13.4 功能描述 200
13.5 间歇使能模式 202
13.6 外部电源检测 202
13.7 电源检测阈值 203
13.8 寄存器 207
13.8.1 SVD 配置寄存器（SVD_CFGR） 207
13.8.2 SVD 控制寄存器（SVD_CR） 208
13.8.3 SVD 中断使能寄存器（SVD_IER） 208
13.8.4 SVD 状态和标志寄存器（SVD_ISR） 209
13.8.5 SVD 参考电压选择寄存器（SVD_VSR） 209

14 AES 硬件运算单元（AES） 211

14.1 功能描述 211
14.2 工作模式 211
14.3 AES 数据流处理模式 212
14.3.1 ECB 模式 212
14.3.2 CBC 模式 213
14.3.3 暂停模式 215
14.3.4 CTR 模式 216
14.3.5 CTR 模式下的暂停模式 217
14.3.6 GCM 模式 217
14.3.7 MultH 模块 220
14.3.8 推荐的 GCM 流程 221
14.4 数据类型 222
14.5 工作流程 223
14.5.1 模式 1：加密 223
14.5.2 模式 2：密钥扩展 224
14.5.3 模式 3：解密 225
14.5.4 模式 4：密钥扩展+解密 225
14.5.5 使用 MultH 模块 226
14.6 DMA 接口 227
14.6.1 MultH 模块与 DMA 间接口 228
14.7 错误标志 228
14.8 寄存器 229
14.8.1 AES 控制寄存器（AES_CR） 229
14.8.2 AES 中断使能寄存器（AES_IER） 231
14.8.3 AES 中断标志寄存器（AES_ISR） 232
14.8.4 AES 数据输入寄存器（AES_DIR） 232
14.8.5 AES 数据输出寄存器（AES_DOR） 233
14.8.6 AES 秘钥寄存器（AES_KEYx） 233
14.8.7 AES 初始向量寄存器（AES_IVRx） 234
14.8.8 AES MultH 参数寄存器（AES_Hx） 234

15 随机数发生器（TRNG） 235

15.1 概述 235
15.2 功能描述 235
15.2.1 随机数产生 235
15.2.2 工作时钟 236
15.2.3 随机数读取 236
15.2.4 CRC 运算 236
15.3 寄存器 237
15.3.1 随机数控制寄存器 237
15.3.2 随机数/CRC 结果输出寄存器 237
15.3.3 RNG 标志寄存器 238
15.3.4 CRC 控制寄存器 238
15.3.5 CRC 输入数据寄存器 239
15.3.6 CRC 标志寄存器 239

16 模拟比较器（COMPARATOR） 241

16.1 概述 241
16.2 结构框图 241
16.3 功能描述 242
16.3.1 基本功能 242
16.3.2 内部比较基准选择和产生 242
16.3.3 比较器输出滞回 242

16.3.4 时钟和复位 243
16.3.5 窗口功能 243
16.3.6 输出数字滤波 244
16.3.7 功耗和速度模式 245
16.3.8 比较器中断 245
16.3.9 比较器输出和触发信号输出 246
16.4 寄存器 248
16.4.1 COMP1 控制寄存器（COMP1_CR） 248
16.4.2 COMP2 控制寄存器（COMP2_CR） 249
16.4.3 COMP3 控制寄存器（COMP3_CR） 251
16.4.4 COMP4 控制寄存器（COMP4_CR） 252
16.4.5 COMP 中断配置寄存器（COMP_ICR） 254
16.4.6 COMP 中断标志寄存器（COMP_IF） 255
16.4.7 COMP 缓冲器控制寄存器（COMP_BUFCR） 255

17 硬件除法器（HDIV） 257

17.1 概述 257
17.2 工作流程 257
17.3 寄存器 258
17.3.1 被除数寄存器 258
17.3.2 除数寄存器 258
17.3.3 商寄存器 259
17.3.4 余数寄存器 259
17.3.5 状态标志寄存器 260

18 双线串行总线和系统管理总线（I²C-SMBUS） 261

18.1 概述 261
18.2 主要功能 261
18.3 模块框图 262
18.3.1 I²C-SMB 模块框图 262
18.4 时钟和复位 263
18.5 接口时序 264
18.5.1 通信过程 264
18.5.2 接口时序描述 265
18.5.3 总线时序参数表 265
18.6 I²C 功能描述 267
18.6.1 预设从机地址表 267
18.6.2 I2C 初始化过程 268
18.6.3 I2C 主机功能 269
18.6.4 I2C 从机功能 286
18.7 SMBus 功能描述 293
18.7.1 概述 293
18.7.2 SMBus 初始化过程 296
18.7.3 SMBus 主机功能 297
18.7.4 SMBus 超时功能 300
18.7.5 CRC-8 301
18.8 寄存器 302
18.8.1 I2C_SMB 控制寄存器 1（I2CSMB_CR1） 302
18.8.2 I2C_SMB 控制寄存器 2（I2CSMB_CR2） 304
18.8.3 I2C_SMB 中断控制寄存器（I2CSMB_IER） 305
18.8.4 I2C_SMB 状态和标志寄存器（I2CSMB_ISR） 306
18.8.5 I2C_SMB 波特率设置寄存器（I2CSMB_BGR） 307
18.8.6 I2C_SMB 主机时序控制寄存器（I2CSMB_TCR） 307
18.8.7 I2C_SMB 超时寄存器（I2CSMB_TOR） 308

18.8.8 I2C_SMB 接收数据寄存器（I2CSMB_RXBUF）.....309
18.8.9 I2C_SMB 发送数据寄存器（I2CSMB_TXBUF）.....309
18.8.10 I2C_SMB 从机地址寄存器（I2CSMB_SADR）.....310
18.8.11 CRC 数据寄存器（I2CSMB_CRCDR）.....310
18.8.12 CRC LFSR 寄存器（I2CSMB_CRC_LFSR）.....311
18.8.13 CRC 多项式寄存器（I2CSMB_CRC_POLY）.....311

19 双线串行总线主机（I²C MASTER）.....313

19.1 概述.....313
19.2 结构框图.....313
19.3 时钟.....314
19.4 接口时序.....314
19.4.1 接口时序图.....314
19.4.2 接口时序描述.....315
19.5 I²C 工作模式.....317
19.6 I²C 地址格式.....317
19.7 I²C 初始化.....317
19.7.1 主机波特率配置.....317
19.8 I²C 主机功能.....318
19.8.1 7bit 寻址.....318
19.8.2 10bit 寻址.....324
19.8.3 SCL 延展（Slave Clock Stretching）.....327
19.8.4 超时机制.....327
19.8.5 可编程时序.....328
19.9 寄存器.....329
19.9.1 I2C 主机配置寄存器（I2C_MSPCFGR）.....329
19.9.2 I2C 主机控制寄存器（I2C_MSPCR）.....330
19.9.3 I2C 主机中断使能寄存器（I2C_MSPIER）.....331
19.9.4 I2C 主机中断标志寄存器（I2C_MSPISR）.....331
19.9.5 I2C 主机状态寄存器（I2C_MSPSR）.....332
19.9.6 I2C 主机波特率设置寄存器（I2C_MSPBGR）.....333
19.9.7 I2C 主机收发缓冲寄存器（I2C_MSPBUF）.....334
19.9.8 I2C 主机时序控制寄存器（I2C_MSPTCR）.....334
19.9.9 I2C 主机超时寄存器（I2C_MSPTOR）.....335

20 通用异步收发传输器（UART）.....336

20.1 概述.....336
20.2 结构框图.....337
20.3 UART 类型区分.....338
20.4 UART 字符描述.....338
20.5 功能描述.....339
20.5.1 时钟结构.....339
20.5.2 同步策略.....339
20.5.3 位接收采样.....340
20.5.4 数据发送.....341
20.5.5 数据接收.....343
20.5.6 RS232 硬件流控.....344
20.5.7 低功耗休眠唤醒.....345
20.5.8 使用 DMA 进行 UART 收发.....346
20.5.9 DMA 模式下的发送完成中断.....346
20.5.10 半双工单线通信.....347
20.6 波特率发生.....348
20.6.1 波特率发生.....348

20.6.1 波特率自适应 349
20.7 红外调制 349
20.8 接收超时 350
20.9 发送延迟 350
20.10 LIN 总线通信功能 350
20.10.1 概述 350
20.10.2 LIN 主机操作 353
20.10.3 LIN 从机操作 354
20.10.4 LIN 接收采样与噪声检测 360
20.10.5 LIN 错误状态 361
20.10.6 LIN 自测试模式 364
20.11 寄存器 367
20.11.1 红外调制寄存器（UART_IRCR） 368
20.11.2 UARTx 模式寄存器（UARTx_MCR） 368
20.11.3 UARTx LIN 控制寄存器（UARTx_LINCR） 369
20.11.4 UARTx LIN 波特率同步寄存器（UARTx_LINBSR） 370
20.11.5 UARTx LIN 帧超时寄存器（UARTx_LINFTR） 370
20.11.6 UARTx LIN 发送时序寄存器（UARTx_LINTTR） 371
20.11.7 UARTx LINPre-sync 寄存器（UARTx_LINPSR） 372
20.11.8 UARTx LIN Break 长度寄存器（UARTx_LINBKR） 372
20.11.9 UARTx 控制状态寄存器（UARTx_CSR） 373
20.11.10 UARTx 中断使能寄存器（UARTx_IER） 374
20.11.11 UARTx 中断标志寄存器（UARTx_ISR） 376
20.11.12 UARTx 超时和延迟寄存器（UARTx_TODR） 377
20.11.13 UARTx 接收缓冲寄存器（UARTx_RXBUF） 377
20.11.14 UARTx 发送缓冲寄存器（UARTx_TXBUF） 378
20.11.15 UATRx 波特率产生寄存器（UARTx_BGR） 378
21 通用同异步收发传输器（USART） 380
21.1 概述 380
21.2 UART 类型区分 380
21.3 结构框图 381
21.4 USART 字符描述 382
21.4.1 UART 模式 382
21.4.2 智能卡模式 382
21.5 功能描述 383
21.5.1 时钟结构 383
21.5.2 同步策略 383
21.5.3 位接收采样 384
21.5.4 数据发送 385
21.5.5 数据接收 388
21.5.6 RS232 硬件流控 389
21.5.7 低功耗休眠唤醒 390
21.5.8 使用 DMA 进行 UART 收发 390
21.5.9 DMA 模式下的发送完成中断 390
21.5.10 半双工单线通信 391
21.6 波特率发生 391
21.6.1 波特率发生 391
21.6.2 波特率自适应 393
21.7 红外调制 393
21.8 接收超时 394
21.9 发送延迟 394
21.10 寄存器 395

21.10.1 红外调制寄存器（UART_IRCR）..... 395
21.10.2 USARTx 模式寄存器（USARTx_MCR）..... 396
21.10.3 USARTx 控制状态寄存器（USARTx_CSR）..... 397
21.10.4 USARTx 中断使能寄存器（USARTx_IER）..... 398
21.10.5 USARTx 中断标志寄存器（USARTx_ISR）..... 399
21.10.6 USARTx 超时和延迟寄存器（USARTx_TODR）..... 400
21.10.7 USARTx 接收缓冲寄存器（USARTx_RXBUF）..... 401
21.10.8 USARTx 发送缓冲寄存器（USARTx_TXBUF）..... 401
21.10.9 USATRx 波特率产生寄存器（USARTx_BGR）..... 402
21.10.10 USARTx 智能卡状态标志寄存器（USARTx_SCISR）..... 402
21.10.11 USARTx 智能卡中断使能寄存器（USARTx_SCIER）..... 404
21.10.12 USARTx 智能卡帧格式控制寄存器（USARTx_FFCR）..... 404
21.10.13 USARTx 智能卡额外保护时间寄存器（USARTx_EGTR）..... 406
21.10.14 USARTx 智能卡输出时钟分频寄存器（USARTx_CODR）..... 406

22 低功耗 UART（LPUART）..... 407

22.1 概述..... 407
22.2 结构框图..... 408
22.3 引脚定义..... 409
22.4 工作时钟..... 409
22.5 字符描述..... 409
22.6 功能描述..... 411
22.6.1 位接收采样和发送..... 411
22.6.2 接收流程..... 412
22.6.3 发送流程..... 412
22.6.4 休眠模式下的数据接收唤醒..... 412
22.7 寄存器..... 412
22.7.1 LPUARTx 控制状态寄存器（LPUARTx_CSR）..... 413
22.7.2 LPUARTx 中断使能寄存器（LPUARTx_IER）..... 414
22.7.3 LPUARTx 中断标志寄存器（LPUARTx_ISR）..... 415
22.7.4 LPUARTx 波特率调制寄存器（LPUARTx_BMR）..... 416
22.7.5 LPUARTx 接收数据寄存器（LPUARTx_RXBUF）..... 417
22.7.6 LPUARTx 发送数据寄存器（LPUARTx_TXBUF）..... 417
22.7.7 LPUARTx 数据匹配寄存器（LPUARTx_DMR）..... 418

23 串行外设接口（SPI/I2S）..... 419

23.1 概述..... 419
23.2 SPI 功能描述..... 420
23.2.1 SPI 结构框图..... 420
23.2.2 SPI 接口时序..... 420
23.2.3 I/O 配置..... 423
23.2.4 全双工数据通信..... 424
23.2.5 主机发送间隔..... 425
23.2.6 可编程数据字长..... 426
23.2.7 TX-ONLY 模式..... 426
23.2.8 RX-ONLY 模式..... 426
23.2.9 主机 SSN 控制..... 427
23.2.10 数据冲突..... 427
23.2.11 使用 DMA 进行 SPI 收发..... 428
23.3 I2S 功能描述..... 429
23.3.1 I2S 结构框图..... 429
23.3.2 引脚定义..... 430
23.3.3 协议支持..... 430

23.3.4 时钟产生 434
23.3.5 I2S 主机功能 435
23.3.6 I2S 从机功能 437
23.3.7 帧格式错误 438
23.3.8 DMA 支持 438
23.4 寄存器 439
23.4.1 SPI 控制寄存器 1（SPIx_CR1） 439
23.4.2 SPI 控制寄存器 2（SPIx_CR2） 440
23.4.3 SPI 控制寄存器 3（SPIx_CR3） 442
23.4.4 SPI/I2S 中断控制寄存器（SPIx_IER） 443
23.4.5 SPI/I2S 中断标志寄存器（SPIx_ISR） 443
23.4.6 SPI/I2S 发送缓存寄存器（SPIx_TXBUF） 445
23.4.7 SPI/I2S 发送缓存寄存器（SPIx_RXBUF） 445
23.4.8 I2S 配置寄存器（SPIx_I2SCR） 446
23.4.9 I2S 预分频寄存器（SPIx_I2SPR） 447
24 经典控制器区域网络（FSCAN） 448
24.1 概述 448
24.2 结构框图 449
24.3 功能描述 449
24.3.1 时钟和复位 449
24.3.2 Bit timing 450
24.3.3 Bit stream processor 450
24.3.4 控制器工作模式 450
24.3.5 寄存器写权限与控制器模式 452
24.3.6 消息帧存储和消息帧结构（Message storage and structure） 452
24.3.7 控制器初始化 454
24.3.8 消息发送 454
24.3.9 消息接收 456
24.3.10 消息滤波器（Acceptance Filter） 456
24.3.11 错误管理 459
24.3.12 异步唤醒 461
24.4 寄存器 462
24.4.1 CAN 控制寄存器（CAN_CR） 463
24.4.2 CAN 配置寄存器（CAN_MSR） 464
24.4.3 CAN 波特率预分频寄存器（CAN_BRPR） 465
24.4.4 CAN 位时序寄存器（CAN_BTR） 466
24.4.5 CAN 错误计数寄存器（CAN_ECR） 466
24.4.6 CAN 错误状态寄存器（CAN_ESR） 467
24.4.7 CAN 状态标志寄存器（CAN_SR） 468
24.4.8 CAN 接收中断标志寄存器（CAN_RXISR） 469
24.4.9 CAN 接收中断使能寄存器（CAN_RXIER） 470
24.4.10 CAN 发送缓冲控制寄存器（CAN_TXBCR） 471
24.4.11 CAN 发送中断标志寄存器（CAN_TXBSR） 472
24.4.12 CAN 发送缓冲 ID 寄存器（CAN_TXBxIDR, x=0,1,2） 473
24.4.13 CAN 发送缓冲 DLC 寄存器（CAN_TXBxDLCR, x=0,1,2） 473
24.4.14 CAN 发送缓冲 DataWord1 寄存器（CAN_TXBxDW1R, x=0,1,2） 474
24.4.15 CAN 发送缓冲 DataWord2 寄存器（CAN_TXBxDW2R, x=0,1,2） 474
24.4.16 CAN 接收滤波寄存器（CAN_AFR） 475
24.4.17 CAN 接收滤波配置寄存器（CAN_AFCFGR） 475
24.4.18 CAN 接收滤波掩码寄存器 x（CAN_AFMRx） 476
24.4.19 CAN 接收滤波 ID 寄存器 x（CAN_AFIRx） 477
24.4.20 CAN 接收 FIFO ID 寄存器（CAN_RXFIDR） 478
24.4.21 CAN 接收 FIFO DLC 寄存器（CAN_RXFDLCR） 478

24.4.22 CAN 接收 FIFO DataWord1 寄存器（CAN_RXFDW1R）..... 479
24.4.23 CAN 接收 FIFO DataWord2 寄存器（CAN_RXFDW2R）..... 479

25 灵活速率控制器区域网络（RS-CANFD-LITE）..... 481

25.1 概述..... 481
25.2 主要特性..... 481
25.3 模块框图..... 482
25.4 时钟和复位..... 482
25.5 端口..... 483
25.6 工作模式..... 484
25.6.1 概述..... 484
25.6.2 全局模式（global modes）..... 484
25.6.3 通道模式（channel modes）..... 490
25.6.4 全局模式-通道模式转换交互..... 496
25.7 CAN 初始化..... 497
25.7.1 时钟、bit timing 和 baud rate..... 498
24.7.2 发送延迟补偿（TDC）..... 501
24.7.3 复位后的 CAN 模块初始化..... 503
25.8 消息滤波和全局滤波列表（ACCEPTANCE FILTER LIST – AFL）..... 505
24.8.1 Overview..... 505
24.8.2 DLC 滤波功能..... 505
24.8.3 AFL 描述..... 506
24.8.4 环回模式（Loopback mode）..... 508
24.8.5 IDE masking..... 509
24.8.6 通信过程中更新 AFL entry..... 509
25.9 FIFO 和 MESSAGE BUFFER 配置..... 511
24.9.1 Overview..... 511
24.9.2 RX Message Buffers (RXMB)..... 512
24.9.3 FIFO Buffers..... 512
25.10 中断和 DMA..... 515
25.10.1 中断..... 515
25.10.2 DMA（不使用此功能，因为没有 DMA 通道分配）..... 517
25.11 接收和发送..... 518
25.11.1 消息接收..... 518
25.11.2 时间戳 (Timestamp)..... 521
25.11.3 消息发送..... 521
25.11.4 发送优先级..... 522
25.11.5 Normal transmission..... 523
25.11.6 FIFO transmission..... 526
25.11.7 发送队列（TX Queue）..... 528
25.11.8 发送历史列表（TX History List）..... 528
25.11.9 发送数据填充（TX data padding）..... 531
25.12 测试模式..... 531
25.12.1 通道测试模式..... 531
25.12.2 全局模式..... 533
25.13 RAM 配置..... 537
25.14 PNF (PRETENDED NETWORK FILTER)..... 539
25.14.1 Pretended Network Filter mode (PN mode)..... 540
25.14.2 Pretended Network Filter ID only and Acceptance Filter mode..... 540
25.14.3 Acceptance Filter mode..... 541
25.14.4 Pretended Network Filter and Acceptance Filter mode..... 541
25.14.5 滤波功能细节..... 541
25.15 异步唤醒..... 546
25.16 寄存器..... 547

25.16.1 Channel 0 nominal bitrate configuration register (CFDC0NCFG) 550
25.16.2 Channel 0 Control Register (CFDC0CTR) 551
25.16.3 Channel 0 Status Register(CFDC0STS) 553
25.16.4 Channel 0 Error Flag Register (CFDC0ERFL) 554
25.16.5 Channel 0 Data Bitrate Configuration Register (CFDC0DCFG) 557
25.16.6 Channel 0 CAN-FD Configuration Register (CFDC0FDCFG) 558
25.16.7 Channel 0 CAN-FD Control Register (CFDC0FDCTR) 559
25.16.8 Channel 0 CAN-FD Status Register (CFDC0FDSTS) 560
25.16.9 Channel 0 CAN-FD CRC Register 562
25.16.10 Global IP Version Register (CFDGIPV) 562
25.16.11 Global Configuration Register (CFDGCFG) 563
25.16.12 Global Control Register (CFDGCTR) 565
25.16.13 Global Status Register (CFDGSTS) 566
25.16.14 Global Error Flag Register (CFDGERFL) 566
25.16.15 Global TX Interrupt Status Register(CFDGTINSTS) 567
25.16.16 Global Timestamp Counter Register (CFDGTSC) 568
25.16.17 Global Acceptance Filter List Entry Control Register (CFDGAFLECTR) 569
25.16.18 Global Acceptance Filter List Configuration Register (CFDGAFLCFG) 570
25.16.19 Global Acceptance Filter List ID Registers (CFDGAFLIDr) 570
25.16.20 Global Acceptance Filter List Mask Registers (CFDGAFLMr) 571
25.16.21 Global Acceptance Filter List Pointer0 Registers (CFDGAFLP0r) 572
25.16.22 Global Acceptance Filter List Pointer1 Registers (CFDGAFLP1r) 574
25.16.23 RX Message Buffer Number Registers (CFDRMNB) 575
25.16.24 RX Message Buffer New Data Registers (CFDRMND) 576
25.16.25 RX FIFO Configuration/Control Registers a (CFDRFCCa) 576
25.16.26 RX FIFO Status Registers a (CFDRFSTSa) 578
25.16.27 RX FIFO Pointer Control Registers (CFDRFPCTRa) 578
25.16.28 CommonFIFO Configuration/Control Register (CFDCFCC) 579
25.16.29 Common FIFO Status Register (CFDCFSTS) 581
25.16.30 CommonFIFO Pointer Control Registers (CFDCFPCTR) 581
25.16.31 FIFO Empty Status Register (CFDFESTS) 582
25.16.32 FIFO Full Status Register (CFDFFSTS) 583
25.16.33 FIFO Message Lost Status Register (CFDFMSTS) 583
25.16.34 RX FIFO Interrupt Status Register (CFDRFISTS) 584
25.16.35 DMA Transfer Control Register (CFDCDTCT) （不使用） 584
25.16.36 DMA Transfer Status Register (CFDCDTSTS) （不使用） 584
25.16.37 TX Message Buffer Control Registers I (CFDTMCi) 584
25.16.38 TX Message Buffer Status Registers j (CFDTMSTSj) 585
25.16.39 TX Message Buffer Transmission Request Status Register (CFDTMTRSTS) 586
25.16.40 TX Message Buffer Transmission Abort Request Status Register (CFDTMTARSTS) 586
25.16.41 TX Message Buffer Transmission Completion Status Register (CFDTMTCSTS) 587
25.16.42 TX Message Buffer Transmission Abort Status Register (CFDTMTASTS) 587
25.16.43 TX Message Buffer Interrupt Enable Register (CFDTMIEC) 588
25.16.44 TX Queue Configuration/Control Register (CFDTXQCC) 588
25.16.45 TX Queue Status Register (CFDTXQSTS) 589
25.16.46 TX Queue Pointer Control Registers (CFDTXQPCTR) 590
25.16.47 TX History List Configuration/Control Register (CFDTHLCC) 590
25.16.48 TX History List Status Register (CFDTHLSTS) 591
25.16.49 TX History List Access Register0 (CFDTHLACC0) 592
25.16.50 TX History List Access Register1 (CFDTHLACC1) 592
25.16.51 TX History List Pointer Control Register (CFDTHLPCTR) 593
25.16.52 Global Reset Control Register (CFDGRSTC) 593
25.16.53 Global Test Configuration Register (CFDGTSTCFG) 594
25.16.54 Global Test Control Register (CFDGTSTCTR) 594
25.16.55 Global FD Configuration Register (CFDGFDCFG) 595
25.16.56 Global Lock Key Register (CFDGLOCKK) 596
25.16.57 RAM Test Page Access Register (CFDRPGACCk) 596

25.16.58 Global Pretended Network Filter List Entry Control Register (CFDGPFLECTR) 597
25.16.59 Global Pretended Network Filter List Configuration Register (CFDGPFLCFG) 597
25.16.60 Global Pretended Network Filter ID Registers (CFDGPFLIDs) 598
25.16.61 Global Pretended Network Filter List Mask Registers (CFDGPFLMs) 598
25.16.62 Global Pretended Network Filter List Pointer0 Registers (CFDGPFLP0s) 599
25.16.63 Global Pretended Network Filter List Pointer1 Registers (CFDGPFLP1s) 600
25.16.64 Global Pretended Network Filter List Type Registers (CFDGPFLPTs) 601
25.16.65 Global Pretended Network Filter List Payload Data 0 Registers (CFDGPFLPD0s) 602
25.16.66 Global Pretended Network Filter List Payload Data 1 Registers (CFDGPFLPD1s) 603
25.16.67 Global Pretended Network Filter List Payload Mask 0 Registers (CFDGPFLPM0s) 603
25.16.68 Global Pretended Network Filter List Payload Mask 1 Registers (CFDGPFLPM1s) 604
25.16.69 Global AFL Ignore Entry Register (CFDGAFLIGNENT) 604
25.16.70 Global AFL Ignore Control Register (CFDGAFLIGNCTR) 605
25.16.71 RX Message Buffer Interrupt Enable Configuration Register g (CFDRMIEC) 605
25.17 ID 位对齐格式 606
25.18 MESSAGE BUFFER 结构 607
25.18.1 RX Message Buffer Component b (CFDRMBCPb) 608
25.18.2 RX Message Buffer ID Registers (CFDRMID) 609
25.18.3 RX Message Buffer Pointer Registers (CFDRMPTR) 610
25.18.4 RX Message Buffer CAN-FD Status Registers (CFDRMFDSTS) 610
25.18.5 RX Message Buffer Data Field p Registers (CFDRMDFp) 611
25.18.6 RX FIFO Access Message Buffer Component b (CFDRFMBCPb) 611
25.18.7 RX FIFO Access ID Registers (CFDRFID) 612
25.18.8 RX FIFO Access Pointer Register (CFDRFPTR) 613
25.18.9 RX FIFO Access CAN-FD Status Register (CFDRFFDSTS) 613
25.18.10 RX FIFO Access Data Field p Registers (CFDRFDFp) 614
25.18.11 Common FIFO Access Message Buffer Component b (CFDCFMBCPb) 615
25.18.12 Common FIFO Access ID Registers (CFDCFID) 616
25.18.13 Common FIFO Access Pointer Register (CFDCFPTR) 616
25.18.14 Common FIFO Access CAN-FD Status Register (CFDCFFDSTS) 617
25.18.15 Common FIFO Access Data Field p Registers (CFDCFDFp) 618
25.18.16 TX Message Buffer Component b (CFDTMBCPb) 618
25.18.17 TX Message Buffer ID Registers (CFDTMID) 619
25.18.18 TX Message Buffer Pointer Registers (CFDTMPTR) 620
25.18.19 TX Message Buffer CAN-FD Control Register (CFDTMFDCTR) 620
25.18.20 TX Message Buffer Data Field p Registers (CFDTMDFp) 621

26 单边沿半字节传输模块（SENT） 621

26.1 概述 621
26.2 消息格式 622
26.2.1 Data Nibble 622
26.2.2 同步脉冲（Sync Pulse） 623
26.2.3 Status and communication nibble 623
26.2.4 消息帧格式 625
26.2.5 Optional Pause Pulse 625
26.2.6 CRC 校验 625
26.3 结构框图 626
26.4 功能描述 626
26.4.1 发送 Tick 时间 626
26.4.2 发送模式 627
26.4.3 接收模式 627
26.4.4 接收消息诊断 629
26.5 寄存器 630
26.5.1 SENT 控制寄存器（SENT_CR） 630
26.5.2 SENT 配置寄存器（SENT_CFGR） 631
26.5.3 SENT 预分频寄存器（SENT_PSR） 632

26.5.4 SENT 帧长度寄存器（SENT_FTR）..... 632
26.5.5 SENT 状态和标志寄存器（SENT_SR）..... 633
26.5.6 SENT 数据寄存器（SENT_DR）..... 634
26.5.7 SENT 同步脉冲诊断寄存器（SENT_SYNC）..... 635
26.5.8 SENT 同步校准寄存器（SENT_RCAL）..... 635

27 直接存储访问控制器（DMA）..... 637

27.1 概述..... 637
27.2 工作原理..... 638
27.3 结构框图..... 639
27.4 工作流程..... 639
27.5 访问带宽..... 641
27.6 通道控制..... 642
27.6.1 DMA 请求映射..... 642
27.6.2 通道优先级..... 642
27.6.3 传输方向定义..... 643
27.6.4 循环模式..... 643
27.7 寄存器..... 644
27.7.1 DMA 全局控制寄存器（DMA_GCR）..... 644
27.7.2 通道 x 控制寄存器（DMA_CHxCR）..... 645
27.7.3 通道 x 存储器指针寄存器（DMA_CHxMAD）..... 646
27.7.4 通道 7 控制寄存器（DMA_CH7CR）..... 647
27.7.5 通道 7 Flash 指针寄存器（DMA_CH7FLSAD）..... 648
27.7.6 通道 7 RAM 指针寄存器（DMA_CH7RAMAD）..... 648
27.7.7 DMA 中断状态标志寄存器（DMA_ISR）..... 649

28 循环冗余校验（CRC）..... 651

28.1 概述..... 651
28.2 软件配置过程..... 652
28.3 GOLDEN 数据..... 653
28.4 DMA 接口..... 653
28.5 FLASH 数据完整性校验..... 654
28.6 寄存器..... 655
28.6.1 CRC 数据寄存器（CRC_DR）..... 655
28.6.2 CRC 控制状态寄存器（CRC_CR）..... 655
28.6.3 CRC LFSR 寄存器（CRC_LFSR）..... 657
28.6.4 CRC 输出异或寄存器（CRC_XOR）..... 657
28.6.5 CRC 多项式寄存器（CRC_POLY）..... 658

29 高级定时器（ATIM）..... 659

29.1 概述..... 659
29.2 主要特性..... 659
29.3 结构框图..... 660
29.4 功能描述..... 661
29.4.1 定时单元..... 661
29.4.2 定时器工作模式..... 663
29.4.3 重复计数器..... 670
29.4.4 Preload 寄存器..... 671
29.4.5 计数器工作时钟..... 672
29.4.6 内部触发信号（ITRx）..... 677
29.4.7 捕捉/比较通道..... 678
29.4.8 输入捕捉模式..... 680
29.4.9 软件 Force 输出..... 682
29.4.10 输出比较模式..... 683

29.4.11 PWM 输出 684
29.4.12 互补输出和死区插入 686
29.4.13 刹车功能 687
29.4.14 互补输出通道信号状态逻辑表 689
29.4.15 6-step PWM 输出 690
29.4.16 单脉冲输出 691
29.4.17 外部事件清除 OCxREF 693
29.4.18 编码器接口模式（encoder interface） 694
29.4.19 TIM 从机模式 696
29.4.20 DMA 访问 699
29.4.21 DMA Burst 700
29.4.22 输入异或功能 701
29.4.23 Debug 模式 701
29.5 寄存器 702
29.5.1 ATIM 控制寄存器 1 702
29.5.2 ATIM 控制寄存器 2 703
29.5.3 ATIM 从机模式控制寄存器 705
29.5.4 ATIM DMA 和中断使能寄存器 706
29.5.5 ATIM 状态寄存器 708
29.5.6 ATIM 事件产生寄存器 709
29.5.7 ATIM 捕捉/比较模式寄存器 1 710
29.5.8 ATIM 捕捉/比较模式寄存器 2 712
29.5.9 ATIM 捕捉/比较使能寄存器 714
29.5.10 ATIM 计数器寄存器 715
29.5.11 ATIM 预分频寄存器 716
29.5.12 ATIM 自动重载（auto-reload）寄存器 716
29.5.13 ATIM 重复计数寄存器 717
29.5.14 ATIM 捕捉/比较寄存器 1 717
29.5.15 ATIM 捕捉/比较寄存器 2 718
29.5.16 ATIM 捕捉/比较寄存器 3 718
29.5.17 ATIM 捕捉/比较寄存器 4 719
29.5.18 ATIM 刹车和死区控制寄存器 720
29.5.19 ATIM DMA 控制寄存器 721
29.5.20 ATIM DMA 访问寄存器 722
29.5.21 ATIM 刹车输入控制寄存器 722

30 通用定时器（GPTIM） 724

30.1 概述 724
30.2 主要特性 724
30.3 结构框图 725
30.4 功能描述 726
30.4.1 定时单元 726
30.4.2 定时器工作模式 728
30.4.3 计数器工作时钟 735
30.4.4 内部触发信号（ITRx）的捕捉 741
30.4.5 捕捉/比较通道 742
30.4.6 输入捕捉模式 743
30.4.7 软件 Force 输出 745
30.4.8 输出比较模式 745
30.4.9 PWM 输入模式 746
30.4.10 单脉冲输出 747
30.4.11 外部事件清除 OCxREF 749
30.4.12 编码器接口模式（encoder interface） 749

30.4.13 GPTIM 从机模式 750
30.4.14 DMA 访问 753
30.4.15 DMA Burst 754
30.4.16 输入异或功能 754
30.4.17 Debug 模式 754
30.5 寄存器 755
30.5.1 GPTIMx 控制寄存器 1 755
30.5.2 GPTIMx 控制寄存器 2 756
30.5.3 GPTIMx 从机模式控制寄存器 757
30.5.4 GPTIMx DMA 和中断使能寄存器 759
30.5.5 GPTIMx 状态寄存器 761
30.5.6 GPTIMx 事件产生寄存器 762
30.5.7 GPTIMx 捕捉/比较模式寄存器 1 762
30.5.8 GPTIMx 捕捉/比较模式寄存器 2 765
30.5.9 GPTIMx 捕捉/比较使能寄存器 767
30.5.10 GPTIMx 计数器寄存器 768
30.5.11 GPTIMx 预分频寄存器 769
30.5.12 GPTIMx 自动重载（auto-reload）寄存器 769
30.5.13 GPTIMx 捕捉/比较寄存器 1 770
30.5.14 GPTIMx 捕捉/比较寄存器 2 770
30.5.15 GPTIMx 捕捉/比较寄存器 3 771
30.5.16 GPTIMx 捕捉/比较寄存器 4 771
30.5.17 GPTIMx DMA 控制寄存器 772
30.5.18 GPTIMx DMA 访问寄存器 773
30.5.19 GPTIMx ITR 选择寄存器 773
31 定时器阵列单元（TAU） 775
31.1 概述 775
31.2 结构框图 776
31.3 功能描述 777
31.3.1 时钟和复位 777
31.3.2 工作模式 777
31.3.3 输入输出通道 777
31.3.4 16 位计数模式 778
31.3.5 触发信号输出（TRGO） 780
31.3.6 32 位级联模式 780
31.3.7 捕捉功能 785
31.3.8 预分频 789
31.3.9 16 位 PWM 输出 790
31.3.10 级联 PWM 输出 791
31.3.11 外部引脚输入数字滤波 795
31.4 寄存器 796
31.4.1 TAU0 控制寄存器（TAU0_CR） 798
31.4.2 TAU0 通道 x 配置寄存器（TAU0_TxCFGR） 799
31.4.3 TAU0 通道 x 模式寄存器（TAU0_TxMDR） 800
31.4.4 TAU0 重载寄存器（TAU0_TxARR） 801
31.4.5 TAU0 捕捉比较寄存器（TAU0_TxCCR） 801
31.4.6 TAU0 中断使能寄存器（TAU0_TxIER） 802
31.4.7 TAU0 中断标志寄存器（TAU0_TxISR） 802
31.4.8 TAU0 计数值寄存器（TAU0_TxCNTR） 803
31.4.9 TAU1 控制寄存器（TAU1_CR） 803
31.4.10 TAU1 通道 x 配置寄存器（TAU1_TxCFGR） 804
31.4.11 TAU1 通道 x 模式寄存器（TAU1_TxMDR） 805

31.4.12 TAU1 重载寄存器（TAU1_TxARR）.....806
31.4.13 TAU1 捕捉比较寄存器（TAU1_TxCCR）.....806
31.4.14 TAU1 中断使能寄存器（TAU1_TxIER）.....807
31.4.15 TAU1 中断标志寄存器（TAU1_TxISR）.....807
31.4.16 TAU1 计数值寄存器（TAU1_TxCNTR）.....808

32 32 位基本定时器（BSTIM32）.....809

32.1 概述.....809
32.2 主要特性.....809
32.3 结构框图.....809
32.4 功能描述.....810
32.4.1 定时单元.....810
32.4.2 定时器工作模式.....812
32.4.3 计数器工作时钟.....814
32.4.4 Debug 模式.....815
32.5 寄存器.....816
32.5.1 BSTIM32 控制寄存器 1（BSTIM32_CR1）.....816
32.5.2 BSTIM32 控制寄存器 2（BSTIM32_CR2）.....817
32.5.3 BSTIM32 中断使能寄存器（BSTIM32_IER）.....818
32.5.4 BSTIM32 中断标志寄存器（BSTIM32_ISR）.....818
32.5.5 BSTIM32 事件产生寄存器（BSTIM32_EGR）.....819
32.5.6 BSTIM32 计数器寄存器（BSTIM32_CNT）.....819
32.5.7 BSTIM32 预分频寄存器（BSTIM_PSC）.....820
32.5.8 BSTIM32 自动重载寄存器（BSTIM32_ARR）.....820

33 16 位基本定时器（BSTIM16）.....822

33.1 概述.....822
33.2 主要特性.....822
33.3 结构框图.....822
33.4 功能描述.....823
33.4.1 定时单元.....823
33.4.2 定时器工作模式.....825
33.4.3 计数器工作时钟.....827
33.4.1 Debug 模式.....828
33.5 寄存器.....829
33.5.1 BSTIM 控制寄存器 1.....829
33.5.2 BSTIM 控制寄存器 2.....830
33.5.3 BSTIM 中断使能寄存器.....831
33.5.4 BSTIM 状态寄存器.....831
33.5.5 BSTIM 事件产生寄存器.....832
33.5.6 BSTIM 计数器寄存器.....832
33.5.7 BSTIM 预分频寄存器.....833
33.5.8 BSTIM 自动重载（auto-reload）寄存器.....833

34 32 位低功耗定时器（LPTIM32）.....835

34.1 概述.....835
34.2 结构框图.....836
34.3 时钟和复位.....836
34.4 相关引脚.....837
34.5 定时器功能.....837
34.5.1 普通定时器.....837
34.5.2 外部脉冲触发计数.....837
34.5.3 外部异步脉冲计数.....838
34.5.4 Timeout 模式.....838

34.6 捕捉比较功能 839
34.6.1 32bit PWM 839
34.6.2 输入捕捉 840
34.7 触发信号输出 841
34.8 寄存器 842
34.8.1 LPTIM32 配置寄存器（LPTIM32_CFGR） 842
34.8.2 LPTIM32 计数值寄存器（LPTIM32_CNT） 844
34.8.3 LPTIM32 捕捉比较控制和状态寄存器（LPTIM32_CCSR） 844
34.8.4 LPTIM32 目标值寄存器（LPTIM32_ARR） 846
34.8.5 LPTIM32 中断使能寄存器（LPTIM32_IER） 847
34.8.6 LPTIM32 中断标志寄存器（LPTIM32_ISR） 848
34.8.7 LPTIM32 控制寄存器（LPTIM32_CR） 849
34.8.8 LPTIM32 捕捉比较寄存器 1（LPTIM32_CCR1） 850
34.8.9 LPTIM32 捕捉比较寄存器 2（LPTIM32_CCR2） 850
34.8.10 LPTIM32 捕捉比较寄存器 3（LPTIM32_CCR3） 851
34.8.11 LPTIM32 捕捉比较寄存器 4（LPTIM32_CCR4） 851

35 16 位低功耗定时器（LPTIM16） 852

35.1 概述 852
35.2 结构框图 853
35.3 时钟和复位 853
35.4 定时器功能 854
35.4.1 普通定时器 854
35.4.2 外部脉冲触发计数 854
35.4.3 外部异步脉冲计数 855
35.4.4 Timeout 模式 855
35.5 捕捉比较功能 856
35.5.1 16bit PWM 856
35.5.2 输入捕捉 857
35.5.1 输入数字滤波 858
35.6 正交编码器 859
35.7 寄存器 861
35.7.1 LPTIM 配置寄存器（LPTIM16_CFGR） 861
35.7.2 LPTIM 计数值寄存器（LPTIM16_CNT） 863
35.7.3 LPTIM 捕捉比较控制和状态寄存器（LPTIM16_CCSR） 863
35.7.4 LPTIM 目标值寄存器（LPTIM16_ARR） 865
35.7.5 LPTIM 中断使能寄存器（LPTIM16_IER） 865
35.7.6 LPTIM 中断标志寄存器（LPTIM16_ISR） 866
35.7.7 LPTIM 控制寄存器（LPTIM16_CR） 867
35.7.8 LPTIM 捕捉比较寄存器 1（LPTIM16_CCR1） 868
35.7.9 LPTIM 捕捉比较寄存器 2（LPTIM16_CCR2） 868

36 实时时钟（RTC） 870

36.1 概述 870
36.2 结构框图 870
36.3 工作原理 871
36.3.1 时基计数器（LTBC） 871
36.3.2 LTBC 数字调校 872
36.3.3 BCD 时间 873
36.3.4 RTC 使能与停止 874
36.3.5 RTC 时间设置 874
36.3.6 RTC 时间读取 874
36.3.7 闰年判断 875
36.4 寄存器 876

36.4.1 RTC 写使能寄存器（RTC_WER）.....876
36.4.2 RTC 中断使能寄存器（RTC_IER）.....877
36.4.3 RTC 中断标志寄存器（RTC_ISR）.....878
36.4.4 BCD 时间秒寄存器（RTC_BCDSEC）.....879
36.4.5 BCD 时间分钟寄存器（RTC_BCDMIN）.....880
36.4.6 BCD 时间小时寄存器（RTC_BCDHOUR）.....880
36.4.7 BCD 时间天寄存器（RTC_BCDDAY）.....881
36.4.8 BCD 时间星期寄存器（RTC_BCDWEEK）.....881
36.4.9 BCD 时间月寄存器（RTC_BCDMONTH）.....882
36.4.10 BCD 时间年寄存器（RTC_BCDYEAR）.....882
36.4.11 闹钟寄存器（RTC_ALARM）.....883
36.4.12 RTC 时间信号输出寄存器（RTC_TMSEL）.....883
36.4.13 LTBC 数值调整寄存器（RTC_ADJUST）.....884
36.4.14 毫秒计数值寄存器（RTC_SBSCNT）.....885
36.4.15 RTC 控制寄存器（RTC_CR）.....885

37 模数转换器（ADC）.....887

37.1 概述.....887
37.2 结构框图.....888
37.3 输入通道.....888
37.4 单端输入.....889
37.5 工作时序.....890
37.6 功能描述.....892
37.6.1 使用 VDDA 作为基准.....892
37.6.2 温度传感器.....892
37.6.3 温度传感器的斜率和标定.....893
37.6.4 可编程采样时间.....893
37.6.5 转换模式.....895
37.6.6 转换触发.....898
37.6.7 转换停止.....901
37.6.8 过采样和硬件平均.....902
37.6.9 ADC 工作时钟.....902
37.6.10 数据冲突和自动等待.....905
37.6.11 DMA.....906
37.6.12 模拟窗口看门狗（AWD）.....912
37.6.13 ADC 校准.....913
37.7 低功耗模式.....914
37.8 寄存器.....914
37.8.1 ADC 中断和状态寄存器（ADC_ISR）.....915
37.8.2 ADC 中断使能寄存器（ ADC_IER）.....915
37.8.3 ADC 控制寄存器 1（ADC_CR1）.....916
37.8.4 ADC 控制寄存器 2（ADC_CR2）.....917
37.8.5 ADC 校准控制寄存器（ADC_CALR）.....917
37.8.6 ADC 配置寄存器 1（ADC_CFGR1）.....918
37.8.7 ADC 配置寄存器 2（ADC_CFGR2）.....919
37.8.8 ADC 采样时间控制寄存器（ADC_SMTR）.....922
37.8.9 ADC 通道控制寄存器（ADC_CHER）.....923
37.8.10 ADC 内部通道控制寄存器（ADC_ICHER）.....924
37.8.11 ADC 常规组数据寄存器（ADC_RDR）.....925
37.8.12 AWD 阈值寄存器（ADC_HLTR）.....925
37.8.13 ADC 原始数据寄存器（ADC_RAWR）.....926

38 数模转换器（DAC）.....927

38.1 概述.....927

38.2 结构框图 927
38.3 引脚定义 927
38.4 功能描述 928
38.4.1 工作时钟与信号时序 928
38.4.2 DAC 输出模式 928
38.4.3 DAC 触发源选择 930
38.4.4 DAC 输出电压 930
38.4.5 DAC DMA 传输 930
38.4.6 DAC 输出缓冲 931
38.5 寄存器 931
38.5.1 DAC 控制寄存器 1（DAC_CR1） 932
38.5.2 DAC 控制寄存器 2（DAC_CR2） 932
38.5.3 DAC 配置寄存器（DAC_CFGR） 933
38.5.4 DAC 软件触发寄存器（DAC_SWTRGR） 933
38.5.5 DAC 数据保持寄存器（DAC_DHR） 934
38.5.6 DAC 状态标志寄存器（DAC_ISR） 934
38.5.7 DAC 中断使能寄存器（DAC_IER） 935

39 可编程胶合逻辑（PGL） 936

39.1 概述 936
39.2 结构框图 936
39.3 引脚定义 938
39.4 功能描述 938
39.4.1 LUT 真值表 938
39.4.2 LUT 输入 939
39.4.3 LUT 输出 939
39.4.4 滤波和采样 940
39.4.5 中断和触发 941
39.4.6 低功耗模式 941
39.5 寄存器 942
39.5.1 PGL 控制寄存器（PGL_CR） 942
39.5.2 PGL 配置寄存器 0（PGL_CFGR0） 942
39.5.3 PGL 配置寄存器 1（PGL_CFGR1） 944
39.5.4 PGL 配置寄存器 2（PGL_CFGR2） 945
39.5.5 PGL 配置寄存器 3（PGL_CFGR3） 946
39.5.6 PGL 中断使能寄存器（PGL_IER） 947
39.5.7 PGL 中断标志寄存器（PGL_ISR） 947
39.5.8 LUT0 真值表寄存器（PGL_LUT0） 948
39.5.9 LUT1 真值表寄存器（PGL_LUT1） 948
39.5.10 LUT2 真值表寄存器（PGL_LUT2） 949
39.5.11 LUT3 真值表寄存器（PGL_LUT3） 949

40 I/O 端口（GPIO） 952

40.1 概述 952
40.2 引脚类型 952
40.2.1 GPIO，输入输出使能，可控上下拉电阻，可控开漏输出，可控驱动能力 953
40.3 IO 端口功能定义 954
40.3.1 GPIO 输入 954
40.3.2 GPIO 输出 954
40.3.3 数字外设功能 955
40.3.4 模拟功能 955
40.3.5 使用外部晶体引脚 956
40.4 SWD 引脚 956

40.5 WKUPx 引脚 956
40.6 外部引脚中断（EXTI） 957
40.6.1 功能说明 957
40.6.2 应用指南 958
40.7 快速 GPIO 输出 959
40.8 寄存器 960
40.8.1 GPIO 输入使能寄存器（GPIOx_INEN） 965
40.8.2 GPIO 上下拉使能寄存器（GPIOx_PUDEN） 966
40.8.3 GPIO 开漏使能寄存器（GPIOx_ODEN） 966
40.8.4 GPIO 功能选择寄存器（GPIOx_FCR） 967
40.8.5 GPIO 输出数据寄存器（GPIOx_DO） 969
40.8.6 GPIO 输出数据置位寄存器（GPIOx_DSET） 970
40.8.7 GPIO 输出数据复位寄存器（GPIOx_DRST） 970
40.8.8 GPIO 输入数据寄存器（GPIOx_DIN） 971
40.8.9 GPIO 额外数字功能选择寄存器（GPIOx_DFS） 972
40.8.10 GPIO 驱动强度配置寄存器（GPIOx_DSR） 972
40.8.11 EXTI 边沿选择和使能寄存器 A~I（GPIO_EXTIA/B/…/I） 973
40.8.12 EXTI 数字滤波控制寄存器 AB/CD/EF/GH/I（GPIO_EXTIDFAB/CD/EF/GH/I） 974
40.8.13 EXTI 中断标志寄存器（GPIO_EXTIFAB/CD/EF/GH/I） 975
40.8.14 EXTI 输入信号寄存器（GPIO_EXTIDIAB/CD/EF/GH/I） 975
40.8.15 FOUT 配置寄存器（GPIO_FOUTSEL） 976
40.8.16 WKUP 控制寄存器（GPIO_PINWKEN） 977

41 安全功能（SFU） 979

41.1 概述 979
41.2 闪存 CRC 检查 979
41.3 RAM 自检 980
41.4 FLASH ECC 980
41.5 RAM ECC 980
41.6 MPU 980
41.7 RAM 写保护 980
41.8 关键寄存器写保护 980
41.9 非法地址访问检测 983
41.10 时钟频率检测 984
41.11 晶振停振检测 985
41.12 PLL 失锁检测 985
41.13 电源电压检测 985
41.14 IO 输出状态回读 985
41.15 LIN 自测试 986
41.16 CAN 自测试 986
41.17 ADC 测试 986
41.18 DAC 自测试 987
41.19 比较器自测试 987
41.20 温度检测 988
41.21 寄存器 989
41.21.1 存储访问保护控制寄存器（SFU_MAPCR） 989
41.21.2 存储访问保护状态寄存器（SFU_MAPSR） 990

42 专用编程接口 992

42.1 概述 992
42.2 编程器使用 992

43 调试支持 993

43.1 概述 993

43.2 DEBUG 引脚 994
43.2.1 SWD 引脚 994
43.2.2 上拉电阻 994
43.3 SWD 接口协议 994
43.3.1 协议简介 994
43.3.2 传输序列 994
43.3.3 SW-DP ID code 995
43.3.4 主机读操作 996
43.3.5 主机写操作 996
43.4 SWD-DP 寄存器 997
43.4.1 寄存器列表 997
43.5 CORE DEBUG 寄存器 997
43.6 低功耗调试支持 997
43.7 DEBUG 相关的配置项 998
43.8 寄存器 998
43.8.1 MCU DEBUG 配置寄存器 998
43.8.2 HardFault 查询寄存器 999

44 器件签名信息 1001

44.1 唯一器件 ID 1001

版本列表 1002

上海复旦微电子集团股份有限公司销售及服务网点 1003

表目录

表 1-1FM33FG0xxA 型号列表 40
表 3-1FM33FG0A 极限参数 56
表 3-2 FM33FG0A 典型工作条件 56
表 3-3ACTIVE 电流参数 57
表 3-4 SLEEP 电流参数 57
表 3-5 DEEPSLEEP 电流参数 57
表 3-6 复位和电源监控参数 58
表 3-7 内部基准电压参数 59
表 3-8 唤醒时间参数 59
表 3-9 低频晶体振荡器参数 60
表 3-10 高频晶体振荡器参数 61
表 3-11 内部 RC 振荡器参数 61
表 3-12 内部 RC 振荡器参数 62
表 3-13 PLL 参数 62
表 3-14 ADC 参数 65
表 3-15 ADC 采样时间 66
表 3-16 DAC 参数 67
表 3-17 ADC 采样时间 67
表 3-18 温度传感器参数 67
表 3-19 模拟比较器参数 68
表 3-20 FLASH 参数 69
表 3-21 普通 I/O 参数 69
表 3-22 NRST 引脚参数 70
表 3-23 引脚 AC 参数 70
表 4-1 外设模块总线地址列表 75
表 6-1FM33FG0A CPU 配置简表 121
表 6-2 CORTEX-M0 内核寄存器简表 122
表 7-1 中断向量表 131
表 9-1 IWDT 溢出周期表 148
表 10-1 WWDT 溢出周期表 155
表 16-1 比较器工作模式 245
表 18-1 I²C 接口时序要求 266
表 18-2 I²C-SMBUS 从机保留地址定义 268
表 18-3SMBUS 超时事件 294
表 19-1 I²C 接口时序要求 316
表 19-2 I²C 保留地址定义 317
表 20-1 UART 类型列表 338
表 20-2 UART 数据帧格式 339
表 20-3 DMA 发送中断 347
表 20-4 常用时钟频率下波特率计算 349
表 21-1 UART 类型列表 381
表 21-2 UART 数据帧格式 382
表 21-3 DMA 发送中断 391
表 21-4 常用时钟频率下波特率计算 392
表 22-1 LPUART 引脚对应表 409
表 22-2LPUART 数据帧格式 410
表 22-3 LPUART 数据位调制系数 411
表 23-1I2S 典型采样频率计算表 435
表 25-1 RS-CANFD-LITE OVERVIEW 481

表 25- 2 RS-CANFD-LITE 管脚连接 483
表 25- 3 进入 CHANNEL REST/HALT MDOE 的行为 492
表 25- 4 BOEF 和 BORF 置位条件 495
表 25- 5 GLOBAL MODE – CHANNEL MODE TRANSITION INTERACTIONS 497
表 25- 6 BIT TIMING EXAMPLES 499
表 25- 7 NOMINAL BAUD RATE EXAMPLE 501
表 25- 8 NOMINAL AND DATA RATE EXAMPLE 501
表 25- 9AFL 行为与 MME 和 LOOPBACK CONFIGURATION 的关系 509
表 25- 10 FIFO 中断配置与 FIFO 深度 515
表 25- 11 TX MESSAGE BUFFER 发送模式配置 524
表 25- 12FIFO 间隔定时例子 527
表 25- 13BUFFER NUMBER 和 BUFFER TYPE 529
表 25- 14RAM 容量计算 538
表 25- 15RAM 配置举例 539
表 25- 16PNF 和 AFL 配置组合 540
表 25- 17PNF 行为与配置信号关系 541
表 27- 1PDMA 通道映射列表 642
表 29- 1ENCODER INTERFACE 计数方式 694
表 30- 1ENCODER INTERFACE 计数方式 749
表 31- 1 定时器级联触发选择 785
表 31- 2 定时器捕捉源选择 789
表 31- 3 定时器捕捉源选择 789
表 34- 1 LPTIM32 引脚映射 837
表 35- 1 ENCODER INTERFACE 计数方式 859
表 37- 1 ADC 输入通道 889
表 37- 2 温度传感器斜率 893
表 37- 3 ADC 采样时间 894
表 37- 4 DMA 配置与功能 907
表 38- 1 DAC 相关引脚 927
表 38- 2 DAC 触发源 930
表 39- 1 PGL 相关引脚 938
表 39- 2 LUT 输出连接 939
表 40- 1GPIO 功能逻辑定义表 953
表 40- 2FCR 定义表 954
表 41- 1 512K FLASH 版本总线访问控制表 984
表 41- 2 256K FLASH 版本总线访问控制表 984

图目录

图 1-1 芯片结构框图 39
图 2-1 FM33FG0x14A LQFP144 封装图 41
图 2-2 FM33FG0Ax10A LQFP100 封装图 42
图 2-3 FM33FG0Ax6A LQFP64 封装图 43
图 2-4 LQFP144 封装尺寸图 49
图 2-5 LQFP100 封装尺寸图 50
图 2-6 LQFP64 封装尺寸图 51
图 2-7 JEDEC 标准的耐热回流温度曲线 54
图 3-1 ADC 参数说明 63
图 3-2 ADC 通道输入阻抗 65
图 3-3 温度传感器输出曲线 68
图 4-1 系统总线示意图 72
图 4-2 FM33FG0A 总线地址 73
图 5-1 芯片电源结构图 108
图 8-1 芯片复位源框图 133
图 8-2 上下电复位示意图 134
图 8-3 POR, BOR, PDR 复位波形示意图 134
图 9-1 IWDT 结构框图 147
图 9-2 IWDT 窗口示意图 149
图 10-1 WWDT 结构框图 154
图 10-2 WWDT 窗口示意图 156
图 11-1 芯片时钟框图 162
图 12-1 CLM 电路框图 190
图 12-2 参考时钟计数和监控时钟计数 192
图 12-3 监控时钟频率下溢出 192
图 12-4 监控时钟频率上溢出 193
图 12-5 超时检测电路 194
图 13-1 低压检测电路框图 199
图 13-2 低压检测电路工作时序 200
图 13-3 电源检测电路间歇工作模式 201
图 14-1 ECB 模式加密流程 212
图 14-2 ECB 模式解密流程 213
图 14-3 CBC 加密过程 214
图 14-4 CBC 解密过程 215
图 14-5 暂停模式流程 215
图 14-6 CTR 加密流程 216
图 14-7 CTR 解密流程 217
图 14-8 32 位计数器和随机数的存储方式 217
图 14-9 GCM 加密流程 219
图 14-10 GCM 解密流程 220
图 14-11 MULTH 模块框图 221
图 14-12 根据数据类型存储数据的示意图 223
图 14-13 模式 1：加密流程 224
图 14-14 模式 2 示意图 224
图 14-15 模式 3 示意图 225
图 14-16 模式 4 示意图 226
图 14-17 MULTH 模块使用流程示意图 227
图 14-18 输入时 DMA 请求和数据传输示意图 227
图 14-19 输出时 DMA 请求和数据传输示意图 228

图 15-1 真随机数模块框图 235
图 15-2 真随机数模块工作时钟 236
图 16-1 比较器电路框图 241
图 16-2 比较器内建基准缓冲器 242
图 16-3 比较器输出滞回 243
图 16-4 窗口比较器框图 244
图 16-5 窗口比较器波形示意图 244
图 16-6 数字滤波（DFLEN=3）波形示意图 245
图 16-7 比较器中断产生 246
图 16-8 比较器输出逻辑 246
图 16-9 比较器输出上升沿产生触发输出，无滤波 247
图 16-10 比较器输出上升沿下降沿产生触发输出，无滤波 247
图 18-1 I²C-SMB 模块框图 262
图 18-2 I²C 总线时序 264
图 18-3 数据有效时序 264
图 18-4 起始（START）与停止(STOP)命令定义 264
图 18-5 输出应答(ACK) 265
图 18-6I²C 时序参数图例 266
图 18-7 主机向 7 位地址从机写入数据时的帧格式 270
图 18-8I²C 主机软件发送数据流图 271
图 18-9 I²C 主机对 7 位地址从机发送数据流图 272
图 18-10 主机从 7 位地址从机读取数据时的帧格式 272
图 18-11 I²C 软件接收数据流图 273
图 18-12 I²C 从 7 位地址从机读取数据流图 274
图 18-13 I²C 主机接收数据等待 274
图 18-14 双向数据通信帧格式 275
图 18-15 10BIT 寻址，主机向从机写入数据 275
图 18-16I²C 软件发送数据流图 276
图 18-17 10BIT 寻址，主机从从机读取数据 277
图 18-18 I²C 软件发送数据流图 278
图 18-19 I²C 软件发送数据流图 279
图 18-20 I²C 主机 DMA 发送流程图 280
图 18-21 I²C 主机 DMA 接收流程图 282
图 18-22 I²C 多主机时钟同步 283
图 18-23 I²C 多主机数据仲裁 284
图 18-24 主机时序控制 285
图 18-25 I²C 从机数据发送波形 287
图 18-26 I²C 从机数据接收波形 288
图 18-27 I²C 从机数据接收波形（SCLSEN=0，接收溢出） 289
图 18-28 从机地址匹配唤醒 290
图 18-29 I²C 从机 DMA 接收流程图 291
图 18-30 I2C 从机 DMA 发送流程图 292
图 18-31TLOW:SEXT 和 TLOW:MEXT 波形 295
图 18-32 从机地址匹配唤醒 299
图 18-33 从机 START 唤醒 299
图 18-34SMBUS 总线时钟超时事件 300
图 18-35 主机和从机超时事件 301
图 19-1 I²C 主机模块框图 313
图 19-2I²C 总线时序 314
图 19-3 数据有效时序 314
图 19-4 起始（START）与停止(STOP)命令定义 315
图 19-5 输出应答(ACK) 315
图 19-6 主机向 7 位地址从机写入数据时的帧格式 319

图 19-7I²C 软件发送数据流图 320
图 19-8 I²C 主机对 7 位地址从机发送数据流图 321
图 19-9 主机从 7 位地址从机读取数据时的帧格式 321
图 19-10 I²C 软件发送数据流图 322
图 19-11 I²C 从 7 位地址从机读取数据流图 323
图 19-12 双向数据通信帧格式 323
图 19-13 10BIT 寻址，主机向从机写入数据 324
图 19-14I²C 软件发送数据流图 325
图 19-15 10BIT 寻址，主机从从机读取数据 325
图 19-16 I²C 软件发送数据流图 326
图 19-17 I²C 软件发送数据流图 327
图 19-20 主机时序控制 328
图 20-1 UART 接口时序 337
图 20-2 UART 字符描述 338
图 20-3 起始位同步 340
图 20-4 位接收 16 倍采样 340
图 20-5 位接收 8 倍采样 341
图 20-6 UART 异步发送波形 1 342
图 20-7 UART 异步发送波形 2 343
图 20-8 UART 异步发送波形 3 343
图 20-9 两个 UART 之间的硬件流控 344
图 20-10 NRTS 流控 345
图 20-11 NCTS 流控 345
图 20-12 红外调制波形 350
图 20-13 UART 发送延迟 350
图 20-14LIN 总线拓扑 351
图 20-15LIN 总线帧结构 351
图 20-16LIN BREAK FIELD 351
图 20-17LINSYNC FIELD 352
图 20-18LIN 帧 ID 353
图 20-19LIN 总线唤醒信号 354
图 20-21 NORMAL BREAK 356
图 20-22 BREAK ERROR 356
图 20-23 BREAK 超时 357
图 20-24 连续 BREAK 后正常接收 357
图 20-25 LIN 波特率自适应过程 358
图 20-26 波特率误差较大，无 PRE-SYNC 359
图 20-27 LIN 从机接收流程 360
图 20-28 字节间隔和应答间隔 360
图 20-29 LIN BIT SAMPLING 360
图 20-30 LIN BIT ERROR 冲突检测 361
图 20-31 LIN BIT ERROR 监测点 362
图 20-32 PRE-SYNC 后发现 BREAK 长度不足，BREAK ERROR 363
图 20-33 SYNC 段数据错误 364
图 20-34 SYNC 段 0XFF 或 0X00，无 PRE-SYNC 364
图 21-1 USART 模块框图 381
图 21-2 UART 字符描述 382
图 21-3 7816-3 帧结构图 383
图 21-4 起始位同步 384
图 21-5 位接收 16 倍采样 384
图 21-6 位接收 8 倍采样 385
图 21-7 UART 异步发送波形 1 386
图 21-8 UART 异步发送波形 2 387

图 21-9 UART 异步发送波形 3 387
图 21-10 7816 发送和重发 388
图 21-11 两个 UART 之间的硬件流控 389
图 21-12 nRTS 流控 389
图 21-13 nCTS 流控 390
图 21-14 红外调制波形 394
图 21-15 UART 发送延迟 394
图 22-1 LPUART 结构框图 408
图 22-2 字符描述 409
图 23-1 SPI 结构框图 420
图 23-2 SPI 数据/时钟时序图（CPHA=0） 421
图 23-3 SPI 数据/时钟时序图（CPHA=1） 421
图 23-44 线半双工写操作 422
图 23-5 4 线半双工读操作（无 DUMMY CYCLE） 423
图 23-6 4 线半双工读操作（有 DUMMY CYCLE） 423
图 23-7 SPI MASTER/SPI SLAVE 互连 424
图 23-8 SPI MASTER 发送，WAIT=0 426
图 23-9 SPI MASTER 发送，WAIT=1 426
图 23-10 SPI SSN 时序图（SSNM=1，CPHA=0） 427
图 23-11 SPI SSN 时序图（SSNM=0） 427
图 23-12I2S 结构框图 429
图 23-13PHILIPS 标准时序 431
图 23-1432/16-BITS MSB-JUSTIFIED 时序 432
图 23-1524-BITS MSB-JUSTIFIED 时序 432
图 23-16LSB-JUSTIFIED 时序 433
图 23-17PCM 标准时序 434
图 24-1FSCAN 模块结构框图 449
图 24-2 CAN BIT TIMING 450
图 24-3 节点错误状态转移 460
图 25-1 RS-CANFD-LITE 模块框图 482
图 25-2 RS-CANFD-LITE 发送时序 484
图 25-3 全局模式转移路径 484
图 25-4 进入 GLOBAL SLEEP MODE 流程 485
图 25-5 退出 GLOBAL SLEEP MODE 流程 486
图 25-6 进入 GLOBAL RESET MODE 流程 487
图 25-7 退出 GLOBAL RESET MODE 流程 487
图 25-8 进入 GLOBAL HALT MODE 流程 488
图 25-9 退出 GLOBAL HALT MODE 流程 488
图 25-10 进入 GLOBAL OPERATION MODE 流程 489
图 25-11 退出 GLOBAL OPERATION MODE 流程 490
图 25-12 通道模式转移路径 491
图 25-13 CAN CHANNEL 收发子模式 493
图 25-14 节点错误状态转移 494
图 25-15 BUS-OFF 状态下重新初始化发送 496
图 25-16 NOMINAL BIT TIMING 498
图 25-17 DATA BIT TIMING 498
图 25-18 75%采样点设置 499
图 25-19 通过 TSEG 调整补偿采样点 500
图 25-20 FD 帧数据段第二采样点 SSP 502
图 25-21 FD 帧发送延迟补偿 503
图 25-22 TRV_DELAY 测量举例 503
图 25-23 SSP 位置计算 503
图 25-24 硬件复位后的配置流程 504

图 25-25 AFL 的分页访问 507
图 25-26 AFL 配置流程 508
图 25-27 AFL ENTRY 更新流程 510
图 25-28 MESSAGE BUFFER 和 FIFO MAPPING 512
图 25-29 FIFO 配置流程 513
图 25-30 GLOBAL INTERRUPTS BLOCK DIAGRAM 516
图 25-31 CHANNEL INTERRUPTS BLOCK DIAGRAM 517
图 25-32 RXMB 消息访问流程（轮询） 518
图 25-33 RXMB 消息访问流程（中断） 519
图 25-34 RXMB 读取流程 519
图 25-35 FIFO 消息读取流程（轮询） 520
图 25-36 接收 FIFO 读取流程（轮询） 521
图 25-37 发送 BUFFER 配置 522
图 25-38 发送 BUFFER 请求 523
图 25-39 成功发送的请求和标志时序 524
图 25-40 发送撤销的请求和标志时序 525
图 25-41 FIFO 发送请求流程 526
图 25-42 FIFO 间隔定时器结构框图 527
图 25-43 FIFO 间隔定时发送时序 527
图 25-44 TX QUEUE 发送请求 528
图 25-45 THL PREPARATION FLOW 530
图 25-46 THL PROCESSING FLOW 531
图 25-47 软件写保护解锁流程 534
图 25-48 RAM 测试流程 536
图 25-49 RAM 分组 538
图 25-50 PN 模式的消息内容处理 539
图 25-51 PN 模式处理流程 540
图 25-52 PNF 的 RAM 配置 542
图 25-53 PNF 配置流程 543
图 25-54 PNF 操作流程 544
图 25-55 NORMAL MODE 转移 545
图 25-56 REGISTER ENDIANS 545
图 26-1 NIBBLE 波形示意图 623
图 26-2 数据 0 和数据 7 的 NIBBLE 波形 623
图 26-3 短消息格式 OPTION CHANNEL 624
图 26-4 加强消息格式 OPTION CHANNEL 625
图 26-5 消息帧格式 625
图 26-6 SENT 模块结构框图 626
图 27-1 DMA 结构框图 639
图 27-2 DMA 寄存器配置 640
图 27-3 DMA 工作流程 641
图 28-1CRC 运算流程图 652
图 28-2 使用 DMA 对 RAM 中的数据进行 CRC 运算 654
图 29-1 高级定时器结构框图 660
图 29-2 预分频从 1 变为 2 的波形 662
图 29-3 预分频从 1 变为 4 的波形 662
图 29-4 向上计数波形，内部时钟不分频 663
图 29-5 向上计数波形，内部时钟 2 分频 664
图 29-6ARPE=0（ATIM_ARR 没有预装载）时的更新事件 664
图 29-7ARPE=1（ATIM_ARR 预装载）时的更新事件 665
图 29-8 向下计数，内部时钟不分频 666
图 29-9 向下计数，内部时钟 2 分频 666
图 29-10 向下计数，内部时钟 2 分频 667

图 29-11 向下计数，不使用重复计数时的更新事件.....667
图 29-12 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6.....668
图 29-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢).....669
图 29-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出).....669
图 29-15 不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置.....670
图 29-16 ATIM 时钟源框图.....672
图 29-17 内部时钟源模式，时钟分频因子为 1.....672
图 29-18TI2 外部时钟连接例子.....673
图 29-19 外部时钟模式 1 下的时序.....673
图 29-20 外部时钟模式 1 下的时序.....674
图 29-21 外部触发输入框图.....675
图 29-22 外部时钟模式 2 下的时序 1.....675
图 29-23 外部时钟模式 2 下的时序 2.....676
图 29-24 捕获/比较通道(通道 1 输入部分).....678
图 29-25 捕获/比较通道 1 的主电路.....678
图 29-26 捕获/比较通道的输出部分(通道 1 至 3).....679
图 29-27 捕获/比较通道的输出部分(通道 4).....679
图 29-28 PWM 输入捕获模式时序.....680
图 29-29 输出比较模式，翻转 OC1.....683
图 29-30 边沿对齐的 PWM 波形(ARR=7).....684
图 29-31 中央对齐的 PWM 波形(APR=7).....685
图 29-32 带死区插入的互补输出.....686
图 29-33 死区波形延迟大于负脉冲.....686
图 29-34 死区波形延迟大于正脉冲.....686
图 29-35 响应刹车的输出.....688
图 29-36 产生六步 PWM，使用 COM 的例子(OSSR=1).....690
图 29-37 单脉冲模式的例子.....691
图 29-38 ETR 信号清除 ATIM 的 OCxREF.....693
图 29-39 编码器模式下的计数器操作实例.....694
图 29-40 复位模式下的时序.....696
图 29-41 门控模式下的时序.....697
图 29-42 触发器模式下的时序.....698
图 29-43 外部时钟模式 2＋触发模式下的时序.....699
图 30-1 通用定时器架构示意图.....725
图 30-2 预分频从 1 变为 2 的波形.....727
图 30-3 预分频从 1 变为 4 的波形.....727
图 30-4 向上计数波形，内部时钟不分频.....728
图 30-5 向上计数波形，内部时钟 2 分频.....729
图 30-6 ARPE=0（GPTIM_ARR 没有预装载）时的更新事件.....729
图 30-7 ARPE=1（GPTIM_ARR 预装载）时的更新事件.....730
图 30-8 向下计数，内部时钟不分频.....731
图 30-9 向下计数，内部时钟 2 分频.....731
图 30-10 向下计数，内部时钟 2 分频.....732
图 30-11 向下计数，不使用重复计数时的更新事件.....732
图 30-12 中心对齐计数器时序图，GPTIM_PCS=0，GPTIM_ARR=0x6.....733
图 30-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢).....734
图 30-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出).....734
图 30-15 GPTIM 时钟源框图.....735
图 30-16 内部时钟源模式，时钟分频因子为 1.....735
图 30-17 TI2 外部时钟连接例子.....736
图 30-18 外部时钟模式 1 下的时序.....736
图 30-19 外部时钟模式 1 下的时序.....737
图 30-20 外部触发输入框图.....738

图 30-21 外部时钟模式 2 下的时序 1 738
图 30-22 外部时钟模式 2 下的时序 2 739
图 30-23 捕获/比较通道(通道 1 输入部分) 742
图 30-24 捕获/比较通道 1 的主电路 743
图 30-25 捕获/比较通道的输出部分 743
图 30-26PWM 输入捕获模式时序 744
图 30-27 输出比较模式，翻转 OC1 745
图 30-28 边沿对齐的 PWM 波形(ARR=7) 746
图 30-29 中央对齐的 PWM 波形(APR=7) 747
图 30-30 单脉冲模式的例子 748
图 30-31 ETR 信号清除 GPTIM 的 OCxREF 749
图 30-32 编码器模式下的计数器操作实例 750
图 30-33 复位模式下的时序 751
图 30-34 门控模式下的时序 752
图 30-35 触发器模式下的时序 752
图 30-36 外部时钟模式 2＋触发模式下的时序 753
图 31-1TAU 结构框图 776
图 31-2TimerX 结构框图 776
图 31-3 16bit 连续定时功能 778
图 31-4 定时器内部时钟预分频计数 779
图 31-5 定时器外部事件上升沿递增时序 779
图 31-6 定时器外部事件上升下降沿递增，2 分频时序 779
图 31-7 定时器内部信号下降沿递增 780
图 31-8 定时器级联计数（TRIGGER TO INCREMENT） 781
图 31-9 定时器级联计数（TRIGGER TO INCREMENT），从机 Prescale1 = 1 781
图 31-10 定时器级联计数（TRIGGER TO START） 782
图 31-11 定时器级联计数（TRIGGER TO START），从机无法产生溢出 783
图 31-12 定时器级联计数（TRIGGER TO START），从机计数预分频 784
图 31-13 定时器捕捉外部输入信号上升沿 786
图 31-14 定时器 2 分频捕捉外部输入信号下降沿 786
图 31-15 定时器脉冲宽度高电平捕捉 787
图 31-16TI00 二分频计数 790
图 31-17TI00 二分频捕捉 790
图 31-18 16 位 PWM 输出示意图 790
图 31-19 非对称级联 PWM 输出示意图 792
图 31-20 从机 ARR 大于主机的情况 793
图 31-21 非对称非交叠 PWM 输出示意图 794
图 31-22 TIxx 输入数字滤波示意图 795
图 32-1 32 位基本定时器结构框图 809
图 32-2 预分频从 1 变为 2 的波形 811
图 32-3 预分频从 1 变为 4 的波形 811
图 32-4 向上计数波形，内部时钟不分频 812
图 32-5 向上计数波形，内部时钟 2 分频 813
图 32-6ARPE=0（ARR 没有预装载）时的更新事件 813
图 32-7 ARPE=1（ARR 预装载）时的更新事件 814
图 32-8 内部时钟源模式，时钟分频因子为 1 814
图 33-1 16 位基本定时器结构框图 822
图 33-2 预分频从 1 变为 2 的波形 824
图 33-3 预分频从 1 变为 4 的波形 824
图 33-4 向上计数波形，内部时钟不分频 825
图 33-5 向上计数波形，内部时钟 2 分频 826
图 33-6 ARPE=0（ARR 没有预装载）时的更新事件 826
图 33-7 ARPE=1（ARR 预装载）时的更新事件 827

图 33-8 内部时钟源模式，时钟分频因子为 1 827
图 34-1 LPTIM32 结构框图 836
图 34-2 外部 ETR 脉冲上升沿触发计数 838
图 34-3 外部 ETR 脉冲异步计数（下降沿） 838
图 34-4TIMEOUT 模式 839
图 34-5PWM 输出 840
图 34-6 输入信号边沿捕捉 841
图 35-1 LPTIM16 结构框图 853
图 35-2 外部 ETR 脉冲上升沿触发计数 854
图 35-3 外部 ETR 脉冲异步计数（下降沿） 855
图 35-4TIMEOUT 模式 856
图 35-5PWM 输出 857
图 35-6 输入信号边沿捕捉 858
图 35-7 通道输入数字滤波 858
图 35-8 编码器模式下的计数器操作实例 860
图 36-1RTC 结构框图 870
图 36-2LTBC 结构框图 871
图 36-3RTC 时间读取流程图 875
图 37-1 ADC 结构框图 888
图 37-2 单端输入 890
图 37-3 ADC 校准时序 890
图 37-4 ADC 采样转换时序 891
图 37-5 ADC 采样序列时序 891
图 37-6 ADC 输入通道示意图 894
图 37-7 ADC 单次转换全自动触发模式 896
图 37-8 ADC 单次转换半自动触发模式 897
图 37-9 ADC 连续转换模式 898
图 37-10 单次注入触发执行 2 次注入转换 ..... **错误！未定义书签。**
图 37-11 ADC 硬件触发源 900
图 37-12 ADC 注入组硬件触发源 900
图 37-13 常规组转换 STOP 901
图 37-14 常规组和注入组转换 STOP 902
图 37-15 ADC 时钟示意图 905
图 37-16 ADC 自动等待 905
图 37-17 ADC 单次全自动触发+DMA 案例 1 908
图 37-18 ADC 单次全自动触发+DMA 案例 2 909
图 37-19 ADC 单次半自动触发+DMA 910
图 37-20 ADC 全自动触发+DMA 循环模式 911
图 37-21 ADC 连续模式+DMA 循环模式 912
图 37-22 ADC 模拟看门狗阈值示意图 913
图 38-1 DAC 结构框图 927
图 38-2 DAC 工作时序 928
图 38-3 DAC 连续输出模式 929
图 38-4 DAC 触发输出模式 930
图 38-5 触发模式下通过 DMA 更新数据 931
图 39-1LUT 结构框图 937
图 39-2 PGL 结构框图 937
图 39-3 LUT 实现 2 输入 NAND 示意图 939
图 39-4 输出滤波和采样 940
图 39-5 数字滤波 940
图 39-6 使用 LUT0 为 UART 输入做数字滤波 941
图 40-1 普通 GPIO 结构框图 953
图 40-2WKUPx 功能结构框图 956

图 40-3 引脚输入数字滤波 957
图 40-4 EXTI 信号输入示意图 958
图 43-1 CORTEX-M0 调试系统示意图 993

1 产品综述

1.1 概述

FM33FG0A是基于ARM Cortex-M0的低功耗MCU，支持最大512KB程序flash、16KB数据flash和最大64KB RAM；集成12bit SAR-ADC，DAC、CAN-FD控制器，LIN控制器，SENT控制器，等丰富外设；具备超宽工作电压范围和优异的低功耗性能。

主要特性如下：

- 宽电压范围：1.8~5.5V
- 工作温度范围（T_A）：-40℃~+125℃（AEC-Q100 grade1）
- 处理器内核
  - ARM Cortex-M0
  - 支持用户/特权模式
  - 支持MPU
  - 支持中断向量表重定向（VTOR）
  - 最高主频64Mhz
  - SWD调试接口
  - 24bit Systick定时器
  - 单周期乘法器
- 低功耗技术平台
  - 典型运行功耗120uA/MHz@48Mhz
  - Sleep模式：5uA
  - DeepSleep模式，RTC走时+全部RAM保持+CPU内核保持：1.5uA
- 存储器
  - 最大512KB Code Flashw/ ECC，擦写寿命：100,000次
  - 最大16KB Data Flash w/ ECC，擦写寿命：500,000次
  - Flash数据保存时间：10年@85℃
  - 用户代码保护
  - 最大64KB RAM空间，支持ECC
- 丰富的模拟外设
  - 高可靠、可配置BOR电路（支持8级可编程下电复位阈值）
  - 超低功耗PDR电路
  - 可编程电源监测模块（SVD）
  - 2x低功耗模拟比较器

- 12bit 2Msps SAR-ADC，最大31个外部输入通道
- 10bitDAC
- 温度传感器，全温区线性度+/-5℃

- 通用通信接口
  - UART*6，4路支持LIN通信
  - LPUART*3
  - SPI*4，主从模式
  - I2C*4，主机模式
  - I2C_SMBUS*2，主从模式
  - CAN2.0B *1, CANFD *2
  - SENT *2
  - 7通道外设DMA
  - 可编程CRC校验模块
- 定时资源
  - TAU定时器阵列，包含16个独立通道
  - 16bit 4通道通用定时器*2
  - 16bit基本定时器*2
  - 24-bit Systick*1
  - 16bit 2通道低功耗定时器*2
  - 看门狗定时器*2
  - 低功耗实时时钟日历（RTCC），带有数字调校功能，调校精度+/-0.476ppm
- 加密算法
  - AES硬件运算单元，128/192/256-bit
  - AES支持ECB/CBC/CTR/GCM/GMAC模式
  - 真随机数发生器
- 时钟发生电路
  - 片上可配置高速RC振荡器，可配置频率输出8/16/24/32MHz，出厂调校误差+/-0.5%，8/16MHz全温区变化小于+/-2%
  - 低功耗32768Hz晶体振荡器，带有停振检测电路
  - 低功耗低速RC振荡器，32KHz，全温区+/-3%
  - 高频晶体振荡器，4~24MHz
  - PLL，最高输出64MHz
- 封装：LQFP144/LQFP100/LQFP64/LQFP48

1.2 芯片结构框图

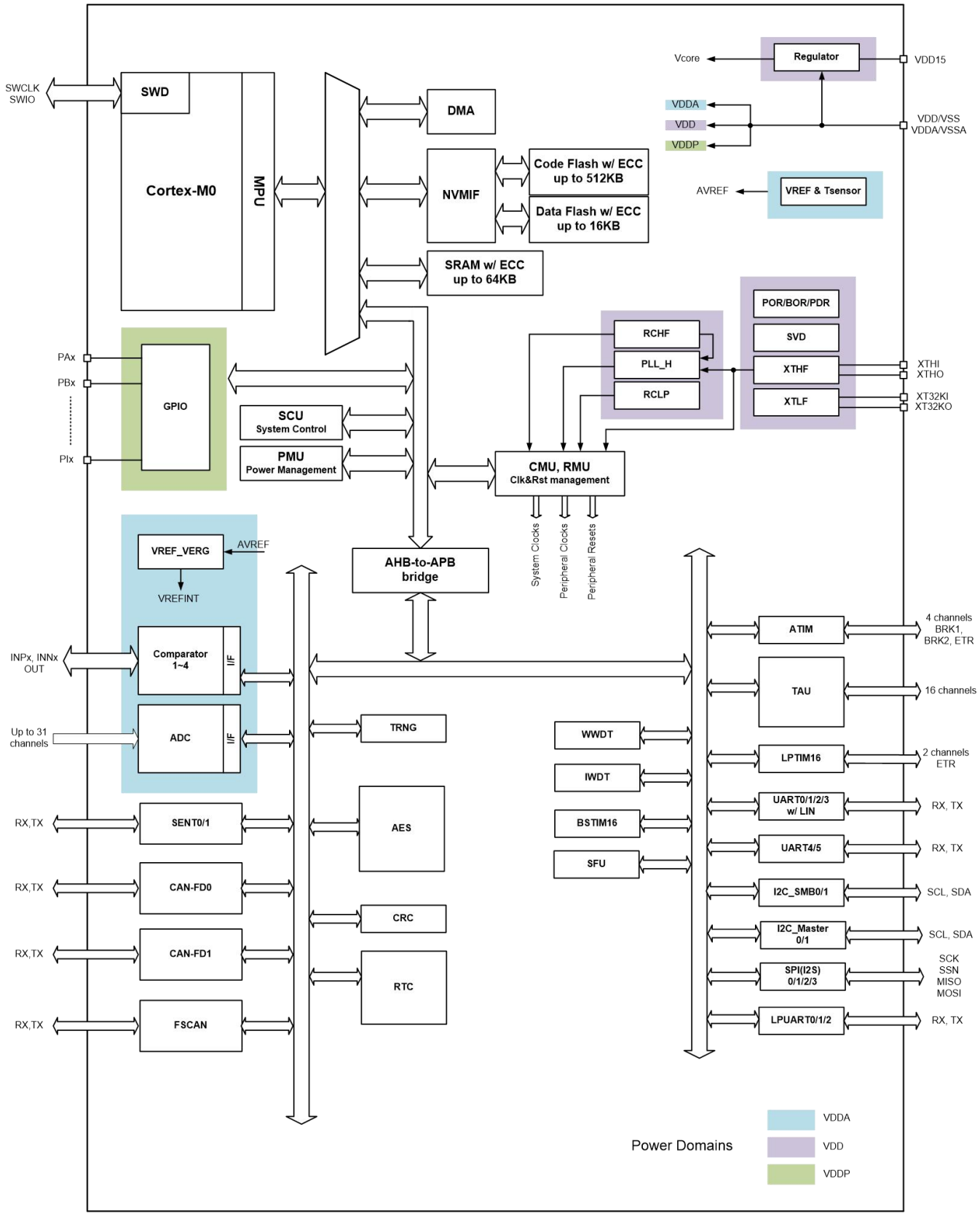


图 1-1 芯片结构框图

1.3 产品型号列表

型号	Code Flash (Kbytes）	Data Flash (Kbytes)	RAM (Kbytes)	封装
FM33FG0614A	512	16	64	LQFP144
FM33FG0610A	512	16	64	LQFP100
FM33FG066A	512	16	64	LQFP64
FM33FG0414A	256	8	32	LQFP144
FM33FG0410A	256	8	32	LQFP100
FM33FG048A	256	8	32	LQFP80

表 1-1FM33FG0xxA 型号列表

2 引脚和封装

2.1 封装和引脚排列

2.1.1 LQFP144 封装图（FM33FG0x14A）

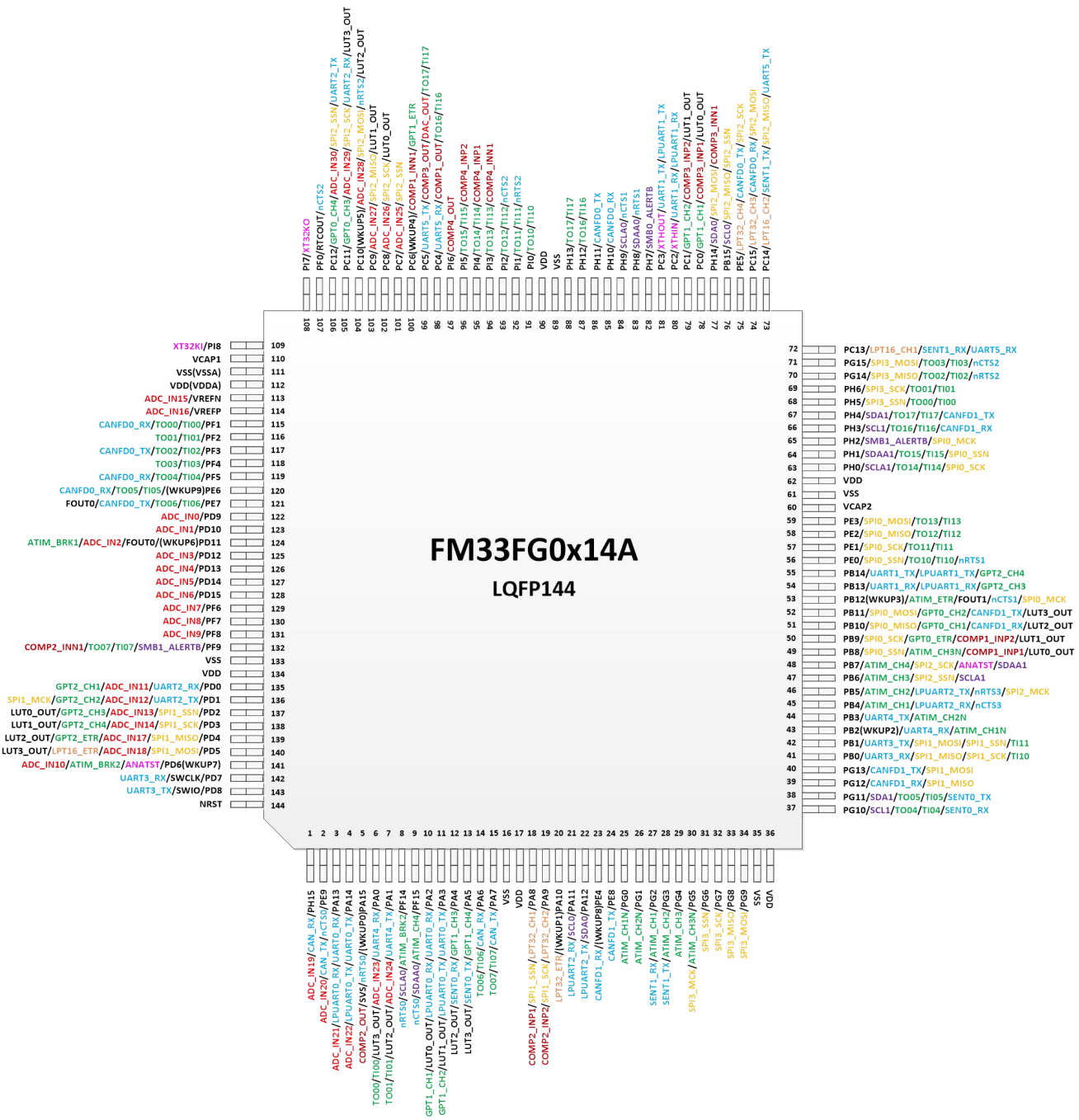


图 2-1 FM33FG0x14A LQFP144 封装图

2.1.2 LQFP100 封装图（FM33FG0x10A）

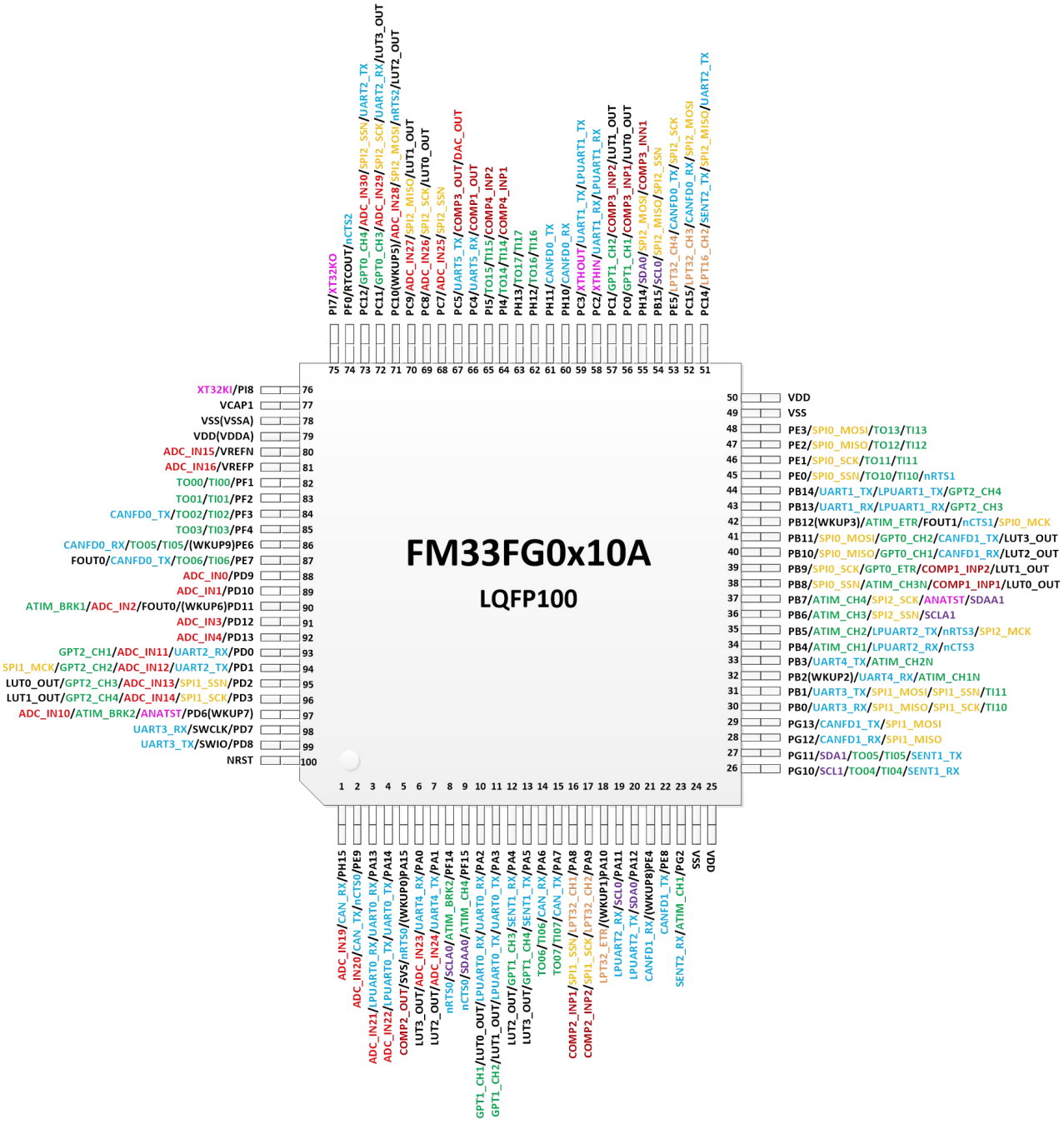


图 2-2FM33FG0Ax10A LQFP100 封装图

2.1.3 LQFP64 封装图（FM33FG0Ax6A）

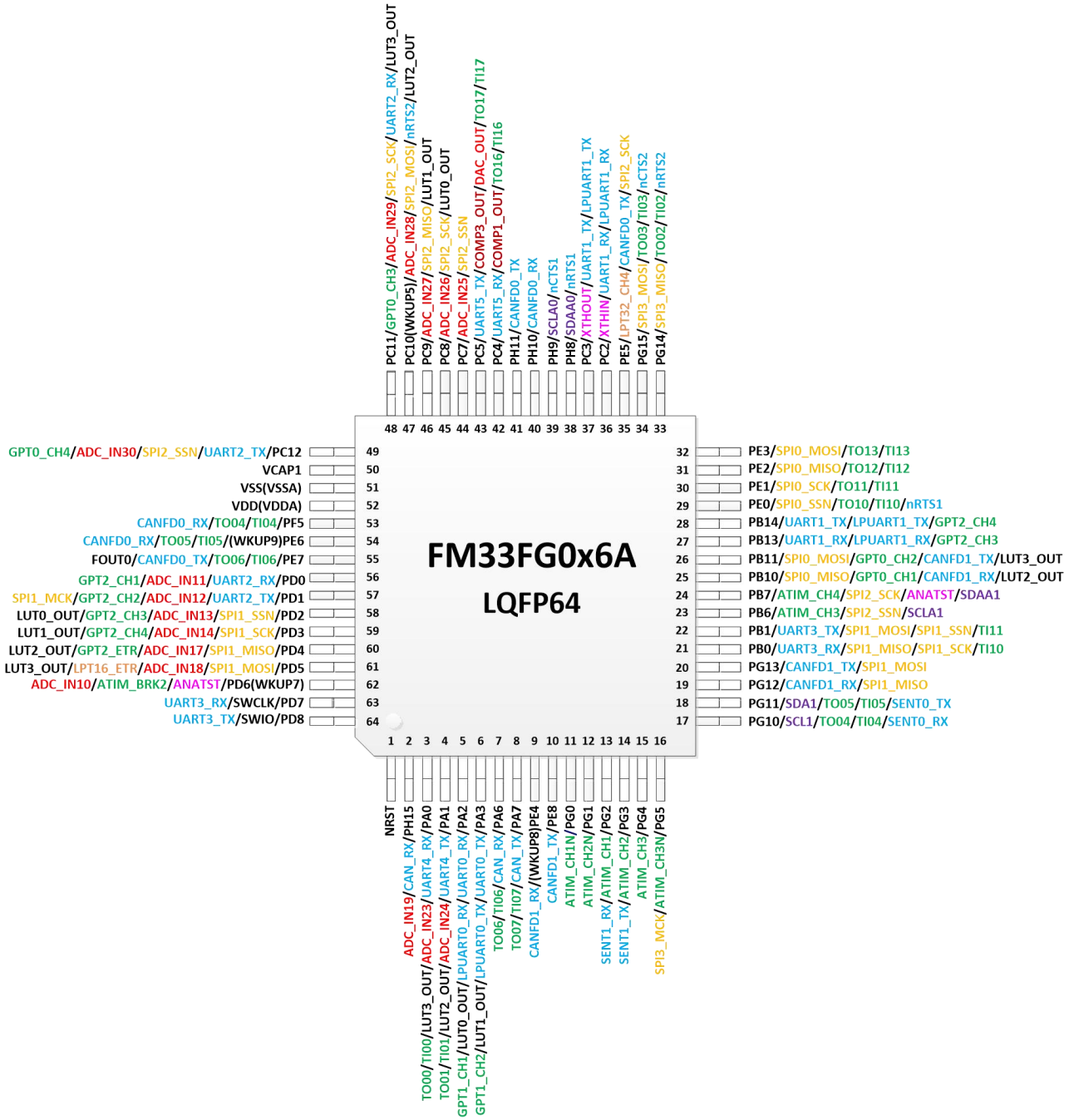


图 2- 3FM33FG0Ax6A LQFP64 封装图

2.1.4 PA 引脚组数字外设功能

GPIO	PADFS			
	00	01	10	11
PA0	UART4_RX	LUT3_OUT	TI00	TO00
PA1	UART4_TX	LUT2_OUT	TI01	TO01
PA2	UART0_RX	LPUART0_RX	LUT0_OUT	GPT1_CH1
PA3	UART0_TX	LPUART0_TX	LUT1_OUT	GPT1_CH2
PA4	GPT1_CH3	SENT0_RX	LUT2_OUT	
PA5	GPT1_CH4	SENT0_TX	LUT3_OUT	
PA6	CAN_RX	TI06	TO06	
PA7	CAN_TX	TI07	TO07	
PA8	LPT32_CH1	SPI1_SSN	canfd1_tx_datarate_en (debug)	
PA9	LPT32_CH2	SPI1_SCK	canfd1_rx_datarate_en (debug)	
PA10	LPT32_ETR	canfd0_tx_datarate_en (debug)		
PA11	SCL0	LPUART2_RX		
PA12	SDA0	LPUART2_TX		
PA13	UART0_RX	LPUART0_RX		
PA14	UART0_TX	LPUART0_TX		
PA15	nRTS0	COMP2_OUT	canfd0_rx_datarate_en (debug)	

2.1.5 PB 引脚组数字外设功能

GPIO	PBDFS			
	00	01	10	11
PB0	UART3_RX	SPI1_MISO	SPI1_SCK	TI10
PB1	UART3_TX	SPI1_MOSI	SPI1_SSN	TI11
PB2	UART4_RX	ATIM_CH1N		
PB3	UART4_TX	ATIM_CH2N		
PB4	ATIM_CH1	LPUART2_RX	nCTS3	
PB5	ATIM_CH2	LPUART2_TX	nRTS3	SPI2_MCK
PB6	ATIM_CH3	SPI2_SSN	SCLA1	
PB7	ATIM_CH4	SPI2_SCK	SDAA1	
PB8	SPI0_SSN	ATIM_CH3N	LUT0_OUT	
PB9	SPI0_SCK	GPT0_ETR	LUT1_OUT	
PB10	SPI0_MISO	GPT0_CH1	CANFD1_RX	LUT2_OUT
PB11	SPI0_MOSI	GPT0_CH2	CANFD1_TX	LUT3_OUT
PB12	ATIM_ETR	FOUT1	nCTS1	SPI0_MCK
PB13	UART1_RX	LPUART1_RX	GPT2_CH3	
PB14	UART1_TX	LPUART1_TX	GPT2_CH4	
PB15	SCL0	SPI2_MISO	SPI2_SSN	

2.1.6 PC 引脚组数字外设功能

GPIO	PCDFS			
	00	01	10	11
PC0	GPT1_CH1	LUT0_OUT		
PC1	GPT1_CH2	LUT1_OUT		
PC2	UART1_RX	LPUART1_RX		
PC3	UART1_TX	LPUART1_TX		
PC4	UART5_RX	COMP1_OUT	TI16	TO16
PC5	UART5_TX	COMP3_OUT	TI17	TO17
PC6	GPT1_ETR			
PC7	SPI2_SSN			
PC8	SPI2_SCK	LUT0_OUT		
PC9	SPI2_MISO	LUT1_OUT		
PC10	SPI2_MOSI	nRTS2	LUT2_OUT	
PC11	GPT0_CH3	SPI2_SCK	UART2_RX	LUT3_OUT
PC12	GPT0_CH4	SPI2_SSN	UART2_TX	
PC13	LPT16_CH1	SENT1_RX	UART5_RX	
PC14	LPT16_CH2	SENT1_TX	SPI2_MISO	UART5_TX
PC15	LPT32_CH3	CANFD0_RX	SPI2_MOSI	

2.1.7 PD 引脚组数字外设功能

GPIO	PDDFS			
	00	01	10	11
PD0	UART2_RX	GPT2_CH1		
PD1	UART2_TX	GPT2_CH2	SPI1_MCK	
PD2	SPI1_SSN	GPT2_CH3	LUT0_OUT	
PD3	SPI1_SCK	GPT2_CH4	LUT1_OUT	
PD4	SPI1_MISO	GPT2_ETR	LUT2_OUT	
PD5	SPI1_MOSI	LPT16_ETR	LUT3_OUT	
PD6	ATIM_BRK2			
PD7	SWCLK	UART3_RX		
PD8	SWIO	UART3_TX		
PD9				
PD10				
PD11	FOUT0	ATIM_BRK1		
PD12				
PD13				
PD14				
PD15				

2.1.8 PE 引脚组数字外设功能

GPIO	PEDFS			
	00	01	10	11
PE0	SPI0_SSN	TO10	TI10	nRTS1
PE1	SPI0_SCK	TO11	TI11	
PE2	SPI0_MISO	TO12	TI12	
PE3	SPI0_MOSI	TO13	TI13	
PE4	CANFD1_RX			
PE5	LPT32_CH4	CANFD0_TX	SPI2_SCK	
PE6	TI05	TO05	CANFD0_RX	
PE7	TI06	TO06	CANFD0_TX	FOUT0
PE8	CANFD1_TX			
PE9	nCTS0	CAN_TX		

2.1.9 PF 引脚组数字外设功能

GPIO	PFDFS			
	00	01	10	11
PF0	RTCOUT	nCTS2		
PF1	TI00	TO00	CANFD0_RX	
PF2	TI01	TO01		
PF3	TI02	TO02	CANFD0_TX	
PF4	TI03	TO03		
PF5	TI04	TO04	CANFD0_RX	
PF6				
PF7				
PF8				
PF9	SMB1_ALERTB	TI07	TO07	
PF14	ATIM_BRK2	SCLA0	nRTS0	
PF15	ATIM_CH4	SDAA0	nCTS0	

2.1.10 PG 引脚组数字外设功能

GPIO	PGDFS			
	00	01	10	11
PG0	ATIM_CH1N			
PG1	ATIM_CH2N			

GPIO	PGDFS			
	00	01	10	11
PG2	ATIM_CH1	SENT1_RX		
PG3	ATIM_CH2	SENT1_TX		
PG4	ATIM_CH3			
PG5	ATIM_CH3N	SPI3_MCK		
PG6	SPI3_SSN			
PG7	SPI3_SCK			
PG8	SPI3_MISO			
PG9	SPI3_MOSI			
PG10	SCL1	TO04	TI04	SENT0_RX
PG11	SDA1	TO05	TI05	SENT0_TX
PG12	CANFD1_RX	SPI1_MISO		
PG13	CANFD1_TX	SPI1_MOSI		
PG14	SPI3_MISO	TO02	TI02	nRTS2
PG15	SPI3_MOSI	TO03	TI03	nCTS2

2.1.11 PH 引脚组数字外设功能

GPIO	PHDFS			
	00	01	10	11
PH0	SCLA1	TO14	TI14	SPI0_SCK
PH1	SDAA1	TO15	TI15	SPI0_SSN
PH2	SMB1_ALERTB	SPI0_MCK		
PH3	SCL1	TO16	TI16	CANFD1_RX
PH4	SDA1	TO17	TI17	CANFD1_TX
PH5	SPI3_SSN	TO00	TI00	
PH6	SPI3_SCK	TO01	TI01	
PH7	SMB0_ALERTB			
PH8	SDAA0	nRTS1		
PH9	SCLA0	nCTS1		
PH10	CANFD0_RX			
PH11	CANFD0_TX			
PH12	TO16	TI16		
PH13	TO17	TI17		
PH14	SDA0	SPI2_MOSI		
PH15	CAN_RX			

2.1.12 PI 引脚组数字外设功能

GPIO	PHDFS			
	00	01	10	11
PI0	TO10	TI10		
PI1	TO11	TI11	nRTS2	
PI2	TO12	TI12	nCTS2	
PI3	TO13	TI13		
PI4	TO14	TI14		

GPIO	PHDFS			
	00	01	10	11
PI5	TO15	TI15		
PI6	COMP4_OUT			
PI7				
PI8				

2.1.13 封装尺寸图

2.1.13.1 LQFP144

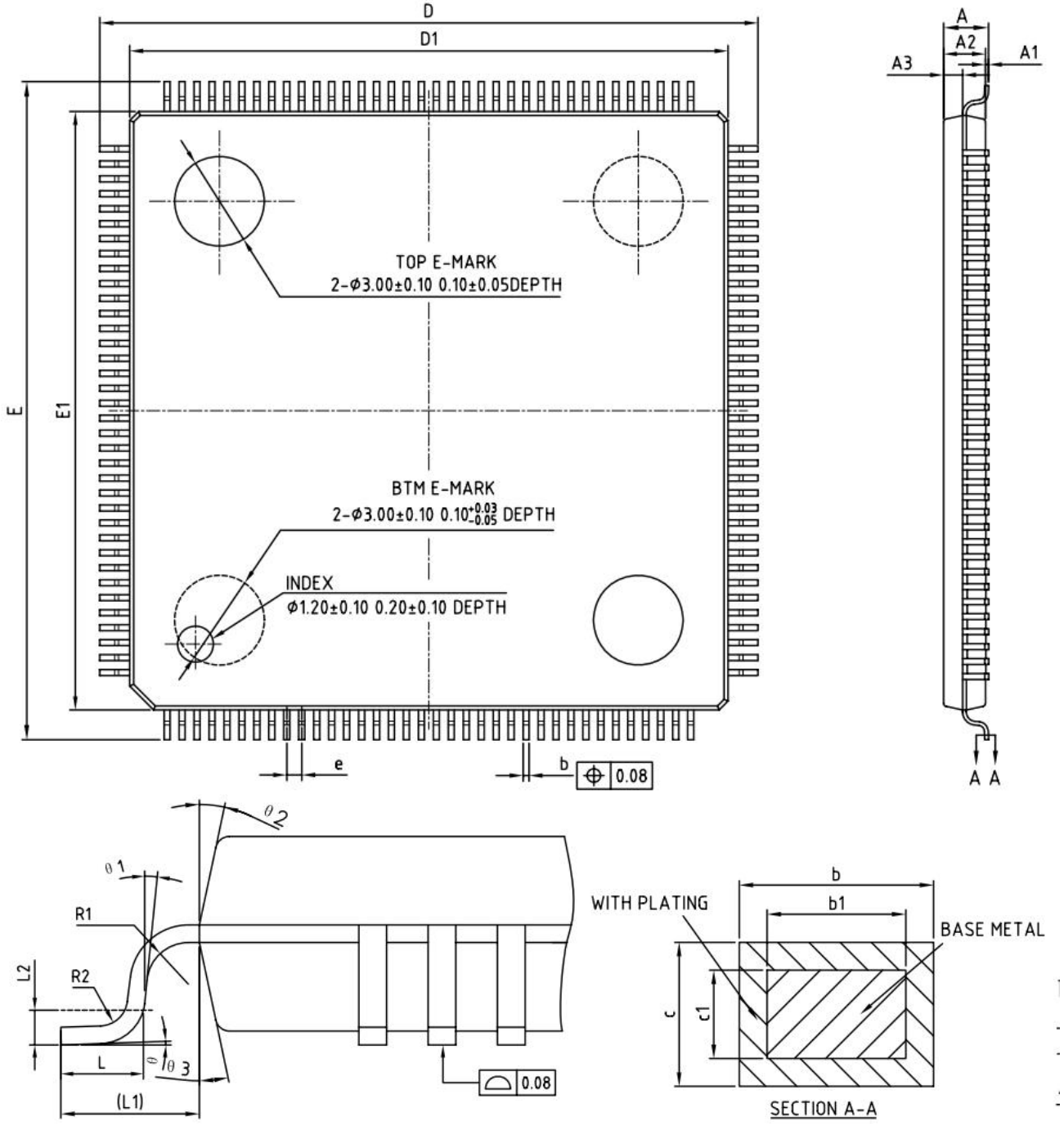
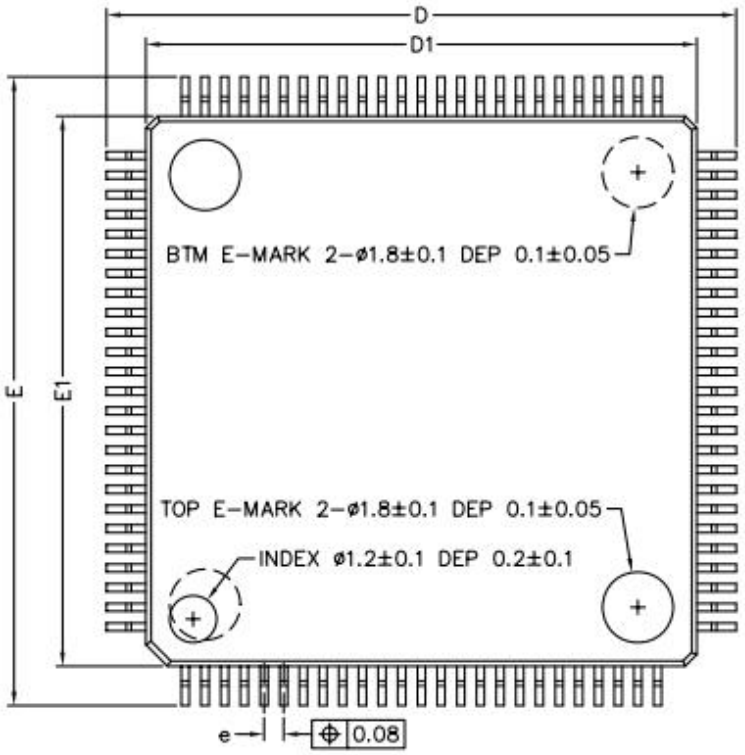


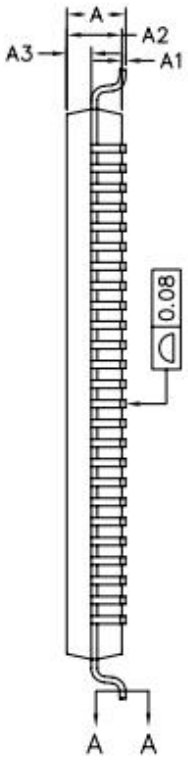
图 2-4 LQFP144 封装尺寸图

COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.17	–	0.27
b1	0.17	0.20	0.23
c	0.127	–	0.18
c1	0.119	0.127	0.135
D	21.80	22.00	22.20
D1	19.90	20.00	20.10
E	21.80	22.00	22.20
E1	19.90	20.00	20.10
e	0.40	0.50	0.60
L	0.45	0.60	0.75
L1		1.00REF	
L2		0.25BSC	
R1	0.08	–	–
R2	0.08	–	–
θ	0°	–	7°
θ 1	0°	–	–
θ 2	11°	12°	13°
θ 3	11°	12°	13°

2.1.13.2 LQFP100





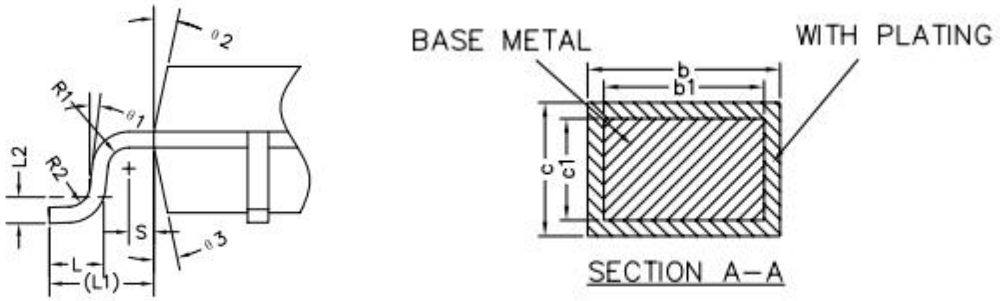


图 2-5 LQFP100 封装尺寸图

Symbol	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	15.80	16.00	16.20
D1	13.90	14.00	14.10
E	15.80	16.00	16.20
E1	13.90	14.00	14.10
e	0.40	0.50	0.60
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

NOTE:
ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026BDD.

2.1.13.3 LQFP64

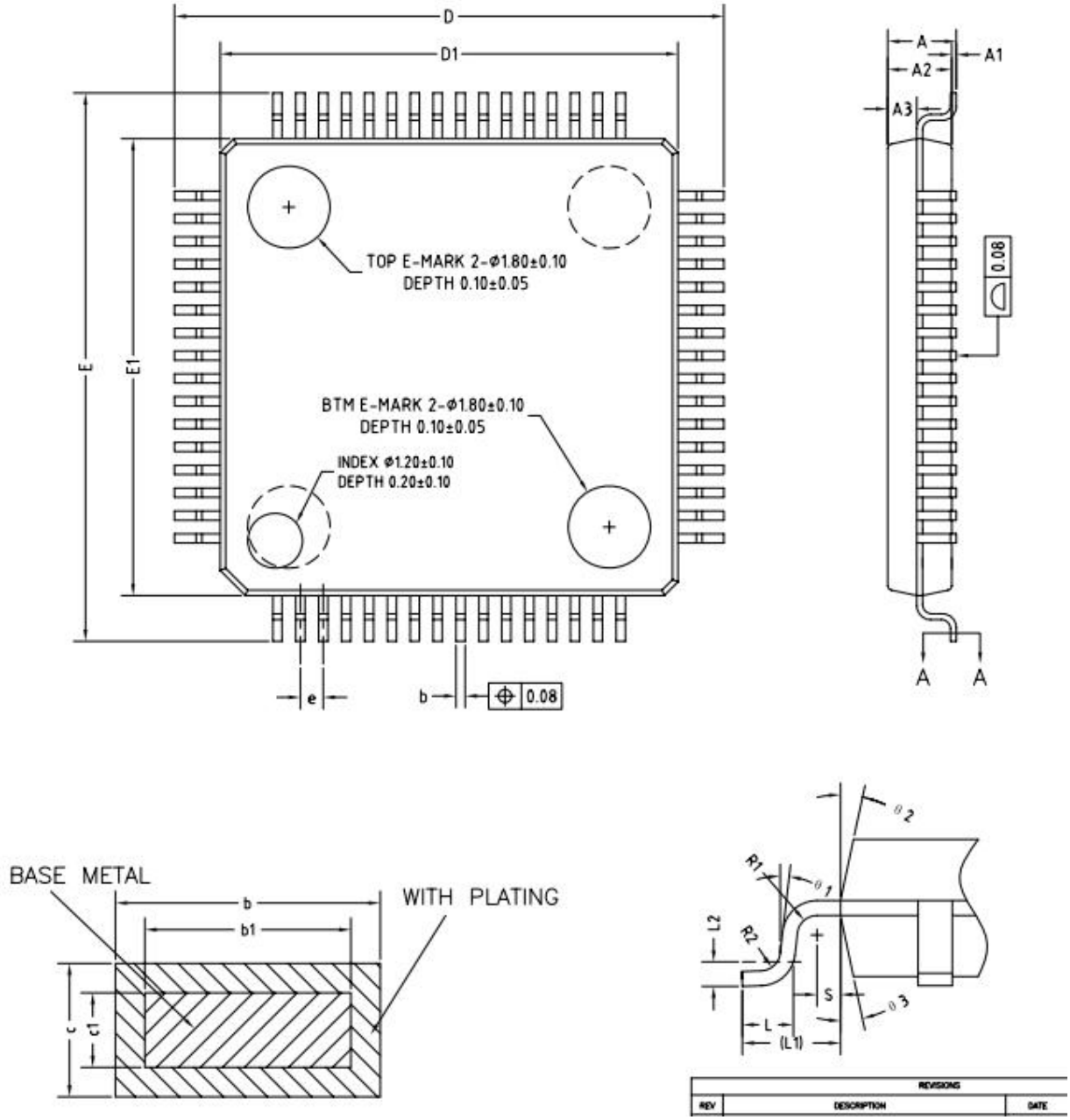


图 2-6 LQFP64 封装尺寸图

Symbol	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	11.80	12.00	12.20

Symbol	MIN	NOM	MAX
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
e		0.50BSC	
L	0.45	0.60	0.75
L1		1.00REF	
L2		0.25BSC	
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

NOTE:
ALL DIMENSIONS REFER TO JEDEC STANDARD MO-220 WMMD-4.

2.2 焊接安装说明

复旦微电子芯片采用无铅工艺封装。回流焊工艺参数建议遵循JEDEC标准进行设定。

根据JEDEC标准J-STD-020，无铅工艺回流焊时的峰值温度设定建议如下表。用户可根据芯片不同厚度和体积的规格，在下表中选择合适的回流焊峰值温度。

封装厚度	塑封体体积 mm^3 <350	塑封体体积 mm^3 350 - 2000	塑封体体积 mm^3 >2000
<1.6mm	260℃	260℃	260℃
1.6~2.5 mm	260℃	250℃	245℃
>2.5mm	250℃	245℃	245℃

下表给出了各种封装形式的回流焊峰值温度：

封装类型	塑封体厚度 mm	塑封体体积 mm^3	回流焊峰值温度
LQFP80	1.4	201.6	260℃

焊接曲线设定请参考JEDEC标准J-STD-020，无铅工艺回流焊温度曲线设定的说明进行设置。

Profile Feature	Pb-Free Assembly
Preheat/Soak Temperature Min (T_{smin}) Temperature Max (T_{smax}) Time (t_s) from (T_{smin} to T_{smax})	 150 °C 200 °C 60-120 seconds
Ramp-up rate (T_L to T_p)	3 °C/second max.
Liquidous temperature (T_L) Time (t_L) maintained above T_L	217 °C 60-150 seconds
Peak package body temperature (T_p)	For users T_p must not exceed the Classification temp in Table 4-2. For suppliers T_p must equal or exceed the Classification temp in Table 4-2.
Time (t_p)* within 5 °C of the specified classification temperature (T_c), see Figure 5-1.	30* seconds
Ramp-down rate (T_p to T_L)	6 °C/second max.
Time 25 °C to peak temperature	8 minutes max.

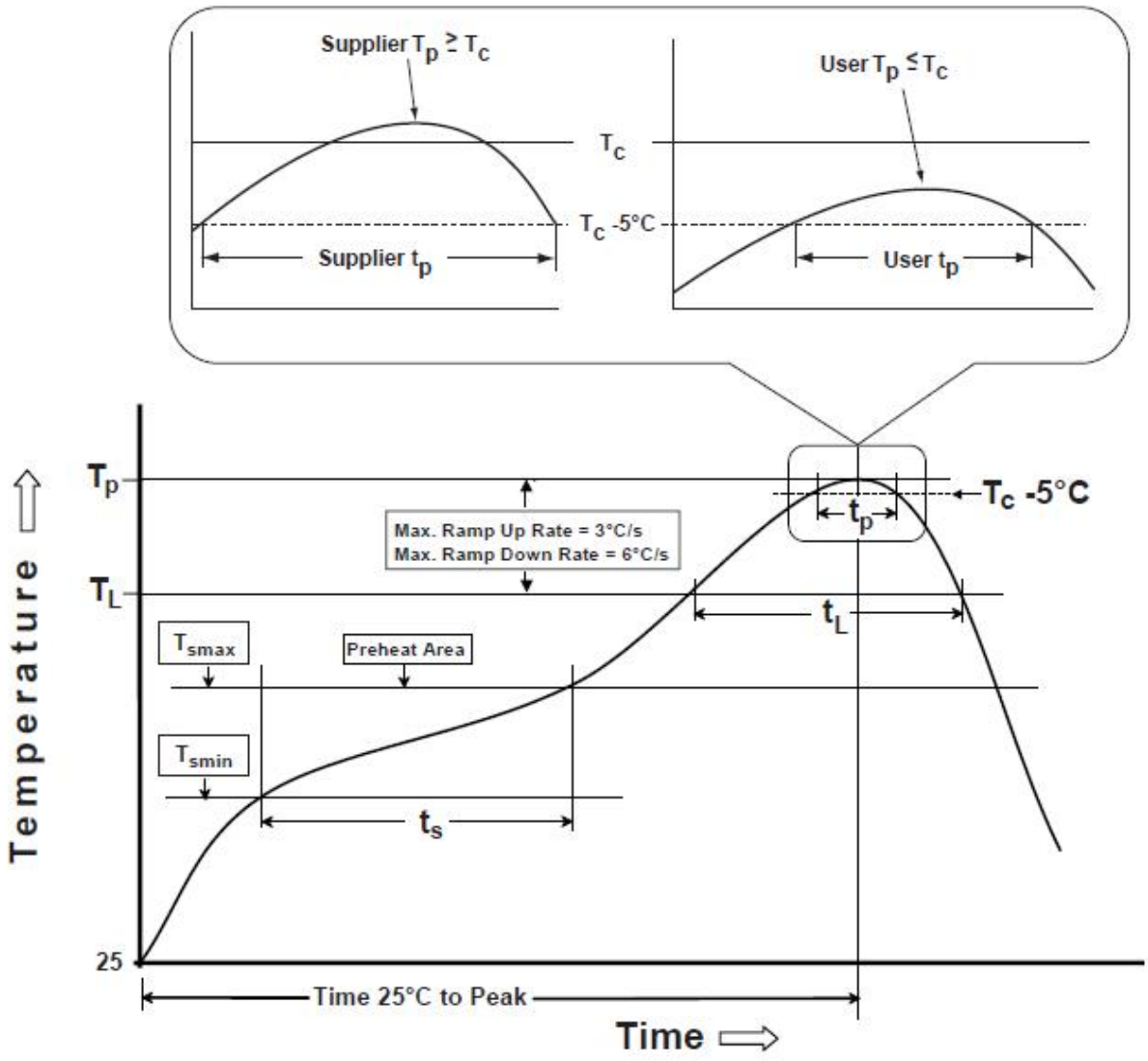


图 2-7JEDEC 标准的耐热回流温度曲线

特别声明：

- 芯片在上板焊接之前，请观察湿度卡是否变色以确认湿敏包装是否完好。
- 如无特殊指定，回流焊次数请勿超过3次

2.3 MSL 等级

FM33FG0A芯片湿敏等级为MSL3，依据JEDEC标准：J-STD-020。请在包装打开置于非干燥环境一周内，进行焊接操作。

3 电参数

3.1 参数说明

除非特别声明，本章节中所罗列的电参数在芯片量产测试过程中，在所标注的环境温度和供电电压下进行测试。

基于特征参数提取、设计仿真得到的参数，在表格备注中声明，这些参数不在量产测试中覆盖。

除非特别声明，typical 参数在 T_A=25C，VDD=5V 条件下，通过测试足够数量的样本获得，这些样本来自于标准量产工艺的晶圆批；typical 参数作为用户设计参考。

3.2 测试标准

FM33FG0xxA 系列 MCU 的量产测试参考 AEC-Q001 和 AEC-Q002 标准执行。

3.3 极限参数

对芯片施加的电压、电流等超过极限参数表定义的最大范围时，可能导致芯片不可恢复的损坏；短时间超过极限参数范围则可能影响芯片的可靠性和工作寿命。

Symbol	Parameter		min	max	unit
V_{DD}-V_{SS}	电源电压 （包含 VDD、VDDA）		-0.3	6.5	V
V_{PIN}	管脚电压		V_{SS}-0.3	6.5	V
VREFP-VREFN	ADC 基准电压		-0.3	6.5	
$\|\triangle V_{SS}\|$	所有地引脚之间的压差		-	50	mV
T_A	环境温度 符合 AEC-Q100 grade1		-40	125	℃
T_J	结温		-40	135	℃
T_{STG}	存储温度		-55	150	℃
HBM	TA=25℃ ESD HBM 模式测试标准符合 JEDEC JS-001	All others	-	+/-4000	V
CDM	ESD CDM 模式 TA=25℃ 测试标准符合 JEDEC JS-002	Corner pins	-	+/-750	V
		Others	-	+/-500	V

Symbol	Parameter	min	max	unit
LU	IO Latchup -(0.5VDD) < VI < (1.5VDD) TA=25℃ 测试标准符合 JESD78E		+/-100	mA
$\sum I_{VDD}$	向芯片 VDD 流入的最大电流(source)		120	mA
$\sum I_{VSS}$	从芯片 VSS 流出的最大电流(sink)		100	mA
$\sum I_{IO}$	所有 IO sink 的最大总和电流		100	mA
	所有 IO source 的最大总和电流		120	mA

表 3-1FM33FG0A 极限参数

3.4 性能参数

3.4.1 典型工作条件

Symbol	Parameter	Conditions	min	max	unit
f_{HCLK}	AHB 时钟频率	T_A=-40~125℃	0	64	MHz
f_{PCLK}	APB 时钟频率		0	64	
VDD	典型工作电压范围		1.8	5.5	V
VDDA	模拟电路工作电压范围		1.8	5.5	V
Vcore	内核 LDO 输出电压 Active 模式	REGC 管脚对地连接 1uF 稳压电容	1.45	1.65	V

表 3-2FM33FG0A 典型工作条件

3.4.2 功耗参数

芯片出厂时的功耗参数在环境温度下测试，高低温电流参数来自于特征参数提取。

测量功耗参数时，MCU 被配置为如下条件：

- 所有功能引脚被配置为 GPIO 模式，并且关闭输入和输出使能，避免引脚浮空漏电
- 除了特别声明的以外，所有外设被关闭，并停止工作时钟
- 常温下的最大功耗数据代表出厂时的测试上限标准
- 常温下的典型功耗数据代表大量样本分布的中心值
- 除非特别声明，所有功耗数据在 VDD=5V 的条件下测试获得

3.4.2.1 Active 模式功耗

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
IDD_{RUN}	运行模式下的功耗，CPU 从 Flash 取指，Coremark	f_{AHB}=8MHz (RCHF) PLL off Flash 0 wait	T_A=25℃	-	2	3	mA
			T_A=125℃	-	2.5	3.5	
		f_{AHB}=64MHz (PLL) PLL on Flash 2 wait	T_A=25℃	-	11.5	15	
			T_A=125℃	-	11.5	15	

表 3-3ACTIVE 电流参数

3.4.2.2 SLEEP 模式功耗

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
I_{sleep}	Sleep 模式电流	BOR、SVD 关闭 RTC 使用 XTLF 走时 CPU、RAM、外设数据保持	T_A=25℃		5.2	9	uA
			T_A=125℃		50	270	

表 3-4 SLEEP 电流参数

3.4.2.3 DEEPSLEEP 模式功耗

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
$I_{deepsleep}$	DeepSleep 模式电流	BOR、SVD 关闭 RTC 使用 XTLF 走时 CPU、RAM、外设数据保持	T_A=25℃	-	1.8	4	uA
			T_A=125℃	-	50	250	

表 3-5 DEEPSLEEP 电流参数

3.4.3 复位和电源监控

芯片的复位和电源监控参数如下表。

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
t_{VDD}	电源上升速度		2		∞	us/V
	电源下降速度	PDR	100		∞	us/V
		BOR	30		∞	us/V
T_{reset_delay}	上电复位延迟时间			0.5		ms
T_{pdr_filter}	下电复位滤波时间			4		us
V_{POR}	上电复位电压[1]	-40℃≤T_A≤125℃	1.4	1.55	1.65	V
V_{BOR_fall}	下电复位电压 -40℃≤T_A≤125℃	BORCFG==2'b000		1.8		V
		BORCFG==2'b001		2.0		
		BORCFG==2'b010		2.2		

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
		BORCFG==2'b011			2.4		
		BORCFG==2'b100			2.7		
		BORCFG==2'b101			3.0		
		BORCFG==2'b110			3.6		
		BORCFG==2'b111			4.0		
V_{PDR}	低功耗下电复位电压[1] T_A=-40~125C	PDRCFG=00			1.4		V
		PDRCFG=01			1.45		
		PDRCFG=10			1.5		
		PDRCFG=11			1.55		
I_{BOR}	BOR 功耗				1		uA
I_{PDR}	PDR 功耗				55		nA
V_{SVD}	电压监测阈值电平	SVD[3:0]=0000	Fall		1.800		V
			Rise		1.900		
		SVD[3:0]=0001	Fall		2.014		V
			Rise		2.114		
		SVD[3:0]=0010	Fall		2.229		V
			Rise		2.329		
		SVD[3:0]=0011	Fall		2.443		V
			Rise		2.543		
		SVD[3:0]=0100	Fall		2.657		V
			Rise		2.757		
		SVD[3:0]=0101	Fall		2.871		V
			Rise		2.971		
		SVD[3:0]=0110	Fall		3.086		V
			Rise		3.186		
		SVD[3:0]=0111	Fall		3.300		V
			Rise		3.400		
		SVD[3:0]=1000	Fall		3.514		V
			Rise		3.614		
		SVD[3:0]=1001	Fall		3.729		V
			Rise		3.829		
		SVD[3:0]=1010	Fall		3.943		V
			Rise		4.043		
		SVD[3:0]=1011	Fall		4.157		V
			Rise		4.257		
		SVD[3:0]=1100	Fall		4.371		V
			Rise		4.471		
		SVD[3:0]=1101	Fall		4.586		V
			Rise		4.686		
		SVD[3:0]=1110	Fall		4.800		V
			Rise		4.900		
		SVD[3:0]=1111	Fall		-		V
			Rise		-		

表 3-6 复位和电源监控参数

[1] 基于特征参数提取

3.4.4 内部基准电压

芯片内建高精度基准电压源，为 ADC 和片外电路提供高精度、高稳定性的参考电压。

芯片出厂时，复旦微电子会在特定的电源电压和温度下，使用片内 ADC 采样基准源输出，并将转换结果保存在芯片的 NVR 中，用户应用中可以将这个转换值作为参考基准使用。

符号	参数说明	总线地址
REF_CAL	ADC 对 AVREF 输出的转换值 测试条件： T_A=30±1℃ VREFP=5V	TBD

内部基准源主要参数：

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
AV_{REF}	基准源输出电压[1]	-40℃≤T_A≤125℃	0.97	1.0	1.03	V
T_{setup}	内部基准源建立时间	-	-	5		us
V_{VREF_MEAS}	出厂时测量转换 AVREF 的 VREFP 电压	-	4.99	5	5.01	V
T_{coeff}	内部基准源温度系数	-40℃≤T_A≤125℃		25	100	ppm/℃
V_{coeff}	内部基准源电压系数	1.8≤VDDA≤5V		1	3	mV/V
T_{S_VREF}	ADC 测量 AVREF 时的采样时间	AVREF buffer 使能	10			us
T_{ADC_BUF}	驱动 ADC 输入的 VREF Buffer 的建立时间		5			us

表 3-7 内部基准电压参数

[1] 基于特征参数提取

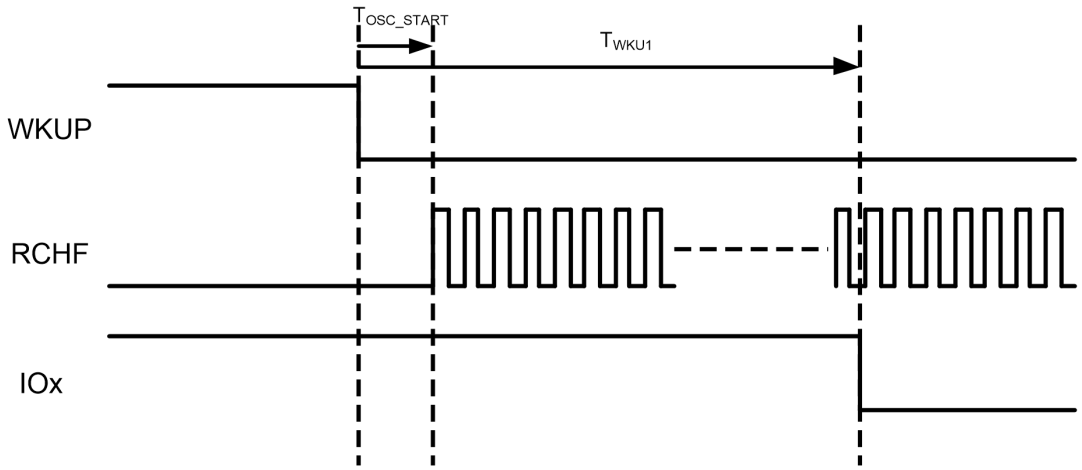
3.4.5 低功耗模式唤醒时间

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
T_{WKU1}	Sleep/DeepSleep 唤醒时间[1]	使用 WKUP 引脚唤醒，PRIMASK=1 禁止中断；CPU 唤醒后执行程序翻转某个 IO 输出，测量 WKUP 信号边沿到 IO 输出翻转之间的时间 F_{SYSCLK}=8Mhz	-	6.5	-	us
T_{WKU2}	LPRUN 模式唤醒时间		-	0	-	us

表 3-8 唤醒时间参数

[1] 基于特征参数提取

典型唤醒事件波形图，仅供设计参考



上图中 T_{OSC_START} 表示唤醒事件到来后 RCHF 环振起振时间，典型值小于 3us

T_{WKU1} 为唤醒事件到来，到程序运行后翻转 IO 的时间，典型值 5.5us。

如果没有通过 PRIMASK 屏蔽中断，则唤醒事件将使 CPU 进入中断服务程序。CPU 进入中断服务程序的过程将额外引入延迟时间。

注意：以上时间评估使用 RCHF 8Mhz 为唤醒后的工作时钟，如果唤醒后选择 16Mhz 或 24Mhz 频率，则唤醒时间相应缩短。

3.4.6 外部时钟源特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
f_{XTLF}	XTLF 振荡频率	外接 32768Hz 晶体		32768		Hz
T_{start}	XTLF 起振时间	外接 32768Hz 晶体 C_{load}=12pF XTLFIPW==3'b000		1	3	s

表 3-9 低频晶体振荡器参数

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
F_{XTHF}	XTHF 振荡频率[1]	VDD=5V		4	-	16	MHz
R_{fb}	反馈电阻[2]	-		-	200	-	KΩ
gm	跨导增益[2]	VDD=5V			TBD		mA/V
		VDD=3.3V			TBD		
VDD_{rise}	XTHF 最低工作电压[1]	8MHz		1.2	-	-	V
		16MHz,		TBD	-	-	
		24MHz		TBD	-	-	
IDD	XTHF 工作电流[1]	8MHz		-	280	-	uA

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
		16MHz	-	TBD	-	
		24MHz	-	TBD	-	
T_{start1}	XTHF 8M 起振时间[1]	VDD=5V，晶振	-	1.3	-	ms
		VDD=5V，陶振	-	0.1	-	ms
T_{start2}	XTHF 16M 起振时间[1]	VDD=5V，晶振	-	TBD	-	ms
		VDD=5V，陶振	-	TBD	-	ms
C_L	负载电容[2]	-	5	-	25	pF

表 3-10 高频晶体振荡器参数

注：

[1] 基于特征参数提取

[2] 基于设计仿真

[3] 推荐使用8~16MHz晶振或陶振，搭配推荐的振荡强度配置，以降低时钟振荡器的功耗和噪声辐射

3.4.7 内部时钟源特性

内部高频 RC 振荡器

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
f_{RCHF}	RCHF 振荡频率	VDD=1.8~5.5V T_A=25C	FSEL==2'b00	7.92	8	8.08	MHz
			FSEL==2'b01	15.84	16	16.16	
			FSEL==2'b10	23.76	24	24.24	
			FSEL==2'b11		RFU		
ACC_{RCHF}[1]	全温区 RCHF 变化范围	VDD=5V	FSEL==2'b00 T_A=-40~125℃	-2.5	-	2.5	%
			FSEL==2'b01 T_A=-40~125℃	-2.5	-	2.5	%
			FSEL==2'b10 T_A=-40~125℃	-4	-	4	%
			FSEL==2'b11 T_A=-40~125℃	-	-	-	%

表 3-11 内部 RC 振荡器参数

[1]：此项指标基于特征参数提取

内部低频 RC 振荡器

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
f_{RCLP}	RCLP 低功耗振荡频率	T_A=25℃	32	32.8	33.5	KHz
		T_A=125℃	29.5	-	36	
		T_A=-40℃	29.5	-	36	
I_{DD_RCLP}	RCLP 功耗[1]	T_A=25℃		350		nA

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
t_{START}	RCLP 启动时间[1]	T_A=25℃		380		us

表 3-12 内部 RC 振荡器参数

[1]：此项指标基于特征参数提取

3.4.8 PLL 特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
F_{PLL}	PLL 输出频率[1]	TA=-40~125℃	32	-	64	MHz
F_{64M}	PLL 输出 64MHz 测试	TA=-40~125℃ 参考时钟为 XTHF8MHz	63.9	64	64.1	MHz
I_{DD_PLL}	PLL 功耗[2]	输入频率 1MHz，输出频率 32Mhz	-	350	-	uA
		输入频率 1MHz，输出频率 64Mhz	-	450	-	
t_{LOCK}	PLL 锁定时间[2]		-	65	-	us

表 3-13 PLL 参数

[1] 基于特征参数提取

[2] 基于设计仿真

3.4.9 ADC 特性

3.4.9.1 参数说明

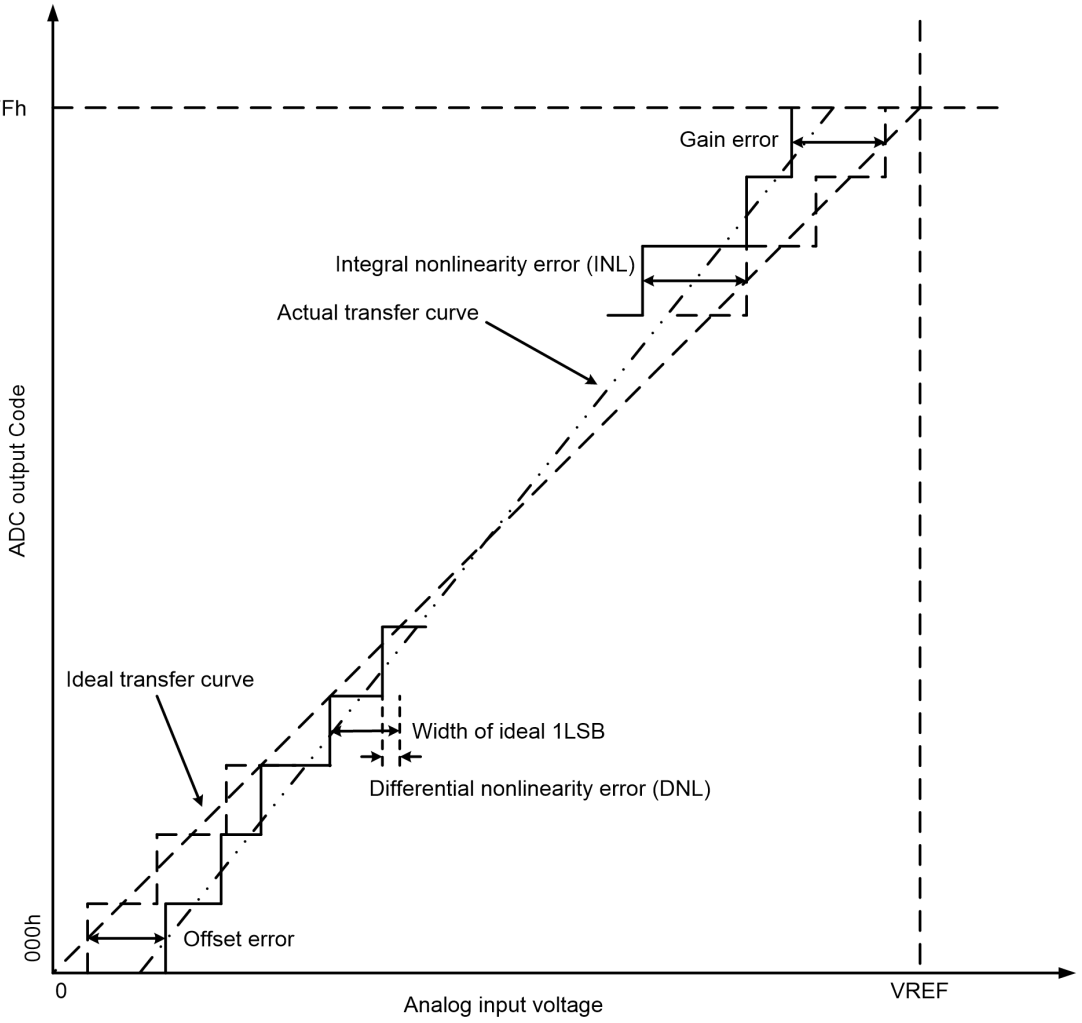


图 3-1 ADC 参数说明

差分非线性（DNL）

DNL表示理想ADC转换曲线1LSB宽度与实际ADC转换曲线1LSB宽度的差值。

积分非线性（INL）

INL表示实际ADC转换曲线与理想ADC转换曲线之间最大偏差。

失调误差（Offset error）

Offset误差表示实际ADC第一个码字跳变的位置与理想ADC第一个码字变化位置之间的差值。

增益误差（Gain error）

Gain error表示满幅输入时实际ADC最后一个码字变化的位置与理想ADC最后一个码字变化位置之间的差值。

3.4.9.2 性能指标

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
VDDA	工作电压范围	-		2.5		5.5	V
VREF+	正参考电压	-		2.5		VDDA	V
VREF-	负参考电压	-		0		0.5	V
T_J	工作结温范围	-		-40		125	℃
V_{AIN}	输入电压范围	-		VREF-		VREF+	V
C_s	采样保持电容	-			3		pF
F_{CLK}	ADC 工作时钟频率	-				32	MHz
F_S	ADC 采样频率	VDDA=2.5~5.5V				2	Msps
T_{SAMP}	采样保持时间	-		2		512	F_{CLK}
T_{CONV}	转换时间	-			14		F_{CLK}
T_{CAL}	自校准时间	-			128	4096	F_{CLK}
IDD	ADC 工作电流	VDDA=3.3V	Fs=1Msps	-	TBD	-	uA
			Fs=2Msps	-	TBD	-	
		VDDA=5V	Fs=1Msps	-	TBD	-	uA
			Fs=2Msps	-	TBD	-	
ADC 动态性能[1]							
ENOB	VDDA=5V VREF+=VDDA F_S=2Msps T_A=25℃ ADCCLK=XTHF	单端模式 F_{AIN}=29KHz		-	10.9	-	bits
	VDDA=3.3V VREF+=VDDA F_S=2Msps T_A=25℃ ADCCLK=XTHF	单端模式 F_{AIN}=29KHz		-	10.8	-	bits
SNDR	信噪失真比 VDDA=5V VREF+=VDDA F_S=2Msps -40℃≤T_A≤125℃	单端模式 F_{AIN}=29KHz		-	66	-	dB
SFDR	无杂散动态范围 VDDA=5V VREF+=VDDA F_S=2Msps F_{AIN}=29KHz -40℃≤T_A≤125℃	单端模式		-	75	-	dB
ADC 静态性能[1]							
DNL	差分非线性 VDDA=5V Fs=1Msps			-0.5	-	0.5	LSB

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
INL	积分非线性 VDDA=5V Fs=1Msps		-2	-	2	LSB
OffsetError	失调误差 校准后			-1.3		LSB
GainError	增益误差 校准后			-0.2		%

表 3-14 ADC 参数

3.4.9.3 输入通道阻抗

下图表示了 ADC 输入通道的阻抗分布。

- ADC_INx 表示快速外部通道
- ADC_INy 表示慢速外部通道
- R_{IO} 表示引脚输入开关阻抗，R_{ADC1} 和 R_{ADC2} 表示 ADC 输入快速通道阻抗和慢速通道阻抗
- C_S 表示 ADC 内部采样电容，典型值 3pF
- 阻抗参数参见后续表格

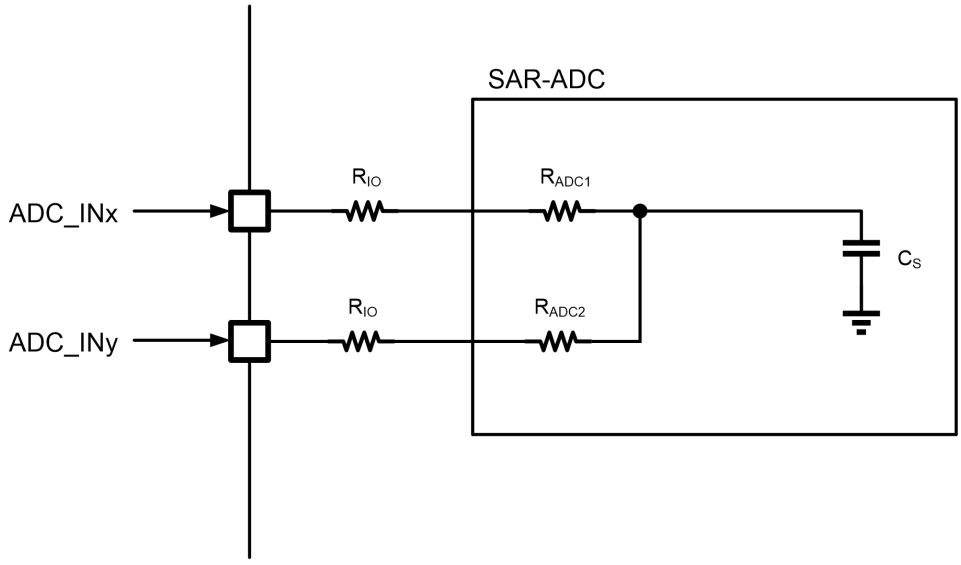


图 3-2 ADC 通道输入阻抗

3.4.9.4 采样时间

ADC 输入信号采样时间最小值由被采样的模拟信号源内阻、信号输入通道阻抗、引脚寄生电容、采样电容共同决定。

ADC 采样外部输入信号的最小采样时间要求根据下式计算：

$$T_{samp} = \ln\left(\frac{2^n}{SA}\right) \times \left(R_{AIN} + R_{ADC} + R_{IO}\right) \times C_{ADC}$$

其中 n=12，SA=0.25LSB（指采样电容上的电压建立到被采样信号电平的 0.25LSB 误差以内），R_{AIN} 表示被采样信号源内阻，R_{IO} 表示输入 IO 阻抗，R_{ADC} 表示 ADC 输入通道阻抗，C_{ADC} 表示 ADC 采样电容。其中 R_{IO} 为 100Ω。

慢速通道的 R_{ADC} 受电源电压、温度和输入信号幅度影响，在输入信号为 VDDA/2 时开关阻抗最大。快速通道则与输入信号幅度无关。以下表格提供了在不同电源、温度条件下，不同通道的 R_{ADC} 参数，用户可以根据这些参数以及信号源特性，计算所需的最小采样时间。

符号	VDDA	温度	参数值			单位
			最小值	典型值	最大值	
R_{ADC}	5V	25C	-	2222	-	Ω
		125C	-	2391	2824	
		-40C	-	2172	2471	
	3.3V	25C	-	2320		
		125C	-	2513	2987	
		-40C	-	2244	2569	
	1.6V	25C	-	2978		
		125C	-	3230	3993	
		-40C	-	2823	3511	

表 3-15 ADC 快速通道内阻

符号	VDDA	温度	参数值			单位
			最小值	典型值	最大值	
R_{ADC}	5V	25C	-	1285	-	Ω
		125C	-	-	1804	
		-40C	-	-	1437	
	3.3V	25C	-	1568		
		125C	-	-	2269	
		-40C	-	-	1795	
	1.6V	25C	-	5123		
		125C	-	-	8188	
		-40C	-	-	17970	

表 3-16 ADC 慢速通道内阻（输入电平 VDDA/2）

3.4.10 DAC 特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
VDDA	工作电压范围	-	2.5	-	5.5	V
VREF	参考电压范围	-	2.5	-	VDDA	V
IDD	DAC 功耗[1] VDDA=VREF=5V 无负载	Middle code (512)		39		uA
		Max code (1023)		14		

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
V_{DAC}	输出电压范围[1]		0		VDDA	V
DNL	差分非线性[1]	Iload<10uA	-0.5		1	LSB
INL	积分非线性[1]		-1		1	LSB
OffsetError	失调误差[1] VDDA=VREF=5V	Code=0x000		3		mV
GainError	增益误差[1] VDDA=VREF=5V	Code=0xFFF		30		mV
t_{setup}	满幅输出建立时间	无负载			1	us

表 3-17 DAC 参数

[1] 基于特征参数提取

3.4.11 温度传感器

芯片出厂时经过温度定标，定标条件是 VDDA=3.0V，TA=30+/-1°C。在此条件下，使用 ADC 采样并转换温度传感器输出电压，将转换结果保存在 Flash 指定地址。

符号	参数说明	测试条件	最小值	典型值	最大值	单位
TS_CAL1	温度传感器标定值 1	VDD=3.0V，T_A=30+/-1°C		TBD		LSB

表 3-18 ADC 采样时间

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
Reso	分辨率[1]	VDDA=VREF+=5V		2.07		LSB/°C
		VDDA=VREF+=3V		3.45		
Slope	输出斜率[1]	T_A=-40~+125° C VDDA=2.5~5.5V		2.53		mV/°C
Linerity	全温区线性度[1]		-		+/-5	°C
I_{DDA}	温度传感器功耗（不含 ADC）[2]	VDDA=5V		0.8		uA
t_{START}	温度传感器启动时间，包含输出 buffer 建立时间[2]				10	us
t_{SAMPLE}	ADC 采样温度传感器输出时要求的采样时间[2]		10	-	-	us

表 3-19 温度传感器参数

[1] 基于特征参数提取

[2] 基于电路设计仿真

温度传感器输出曲线示意图如下。

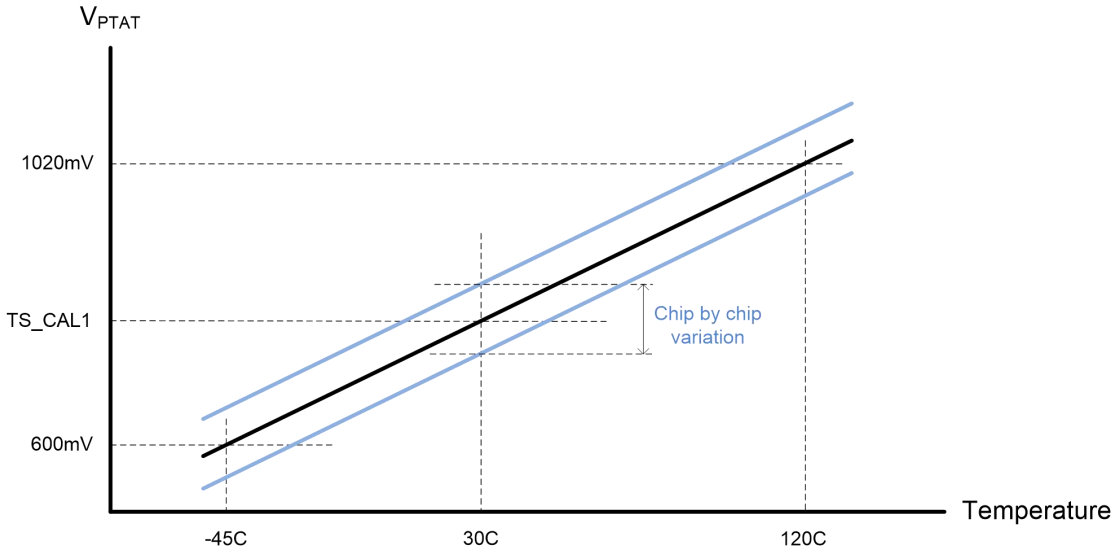


图 3-3 温度传感器输出曲线

温度传感器输出电压只和芯片基底温度有关，而与芯片当前工作电源电压无关。

3.4.12 模拟比较器特性

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
VDDA	比较器工作电压范围	-		2.5	-	5.5	V
VI_{comp1}	比较器输入电压范围	-		0	-	VDDA	V
I_{comp12}	比较器工作电流[2]	高功耗模式			8		uA
		低功耗模式			1.3		
$T_{propagation12}$	比较器传播延迟[1]	VDDA=5V 200mV step 100mV overdrive	高功耗模式		0.6		us
			低功耗模式		1.3		us
$T_{setup12}$	比较器建立时间[2]	VDDA=5V	高功耗模式		3		us
			低功耗模式		17		us

表 3-20 模拟比较器参数

3.4.13 Flash 存储器特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
	Code Flash size		-	-	512K	bytes
	Data flash size				16K	bytes
T_{PROG}	word Program Time		-	25	-	μs
T_{ERASE}	Sector/Block Erase		-	2	-	ms
	Chip Erase		-	8	-	ms
N_{ED}	Sector Endurance for code flash	T_A=125℃	100,000	-	-	Erase/Write cycles
	sector endurance for data flash	T_A=125℃ access time > 80ns ECC on	500,000	-	-	
T_{DR}	Data Retention	After 100K cycling @ T_A=125℃ Qualification compliant to AEC-Q100 equivalent average uselife T_A=85℃	15			yrs

表 3-21 Flash 参数

3.4.14 GPIO 特性

普通 IO

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
V_{IL}	输入低电平		0		0.3V_{DD}	V
V_{IH}	输入高电平		0.7V_{DD}		V_{DD}	V
I_{IL}	输入低漏电	V_{IL}=0V	-1		1	μA
I_{IH}	输入高漏电	V_{IH}=5V	-1		1	μA
V_{OL}	输出低电平	V_{DD}=5V I_{SINK}=20mA		TBD		V
				TBD		
V_{OH}	输出高电平	V_{DD}=5V I_{SOURCE}=20mA		TBD		V
				TBD		
R_{PU}	弱上拉电阻			50		KΩ
R_{PD}	弱下拉电阻			50		KΩ

表 3-22 普通 I/O 参数

NRST 引脚

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
V_{IL}	输入低电平		0		0.3V_{DD}	V
V_{IH}	输入高电平		0.7V_{DD}		V_{DD}	V

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
I_{IL}	输入低漏电	V_{IL}=0V	-1		1	μA
I_{IH}	输入高漏电	V_{IH}=5V	-1		1	μA
R_{PU}	上拉电阻			10		KΩ
$T_{AFILTER}$	模拟滤波长度[1]	VDD=5V		100		ns
$T_{DFILTER}$	数字滤波长度[1]	VDD=2.5~5.5V -40℃≤T_A≤125℃	50		150	us

表 3-23 NRST 引脚参数

注：

[1] 此项参数基于特征参数提取

GPIO AC 特性

IO	符号	参数说明[1]	测试条件	min	max	单位
非 FM+	Fmax	Maximum frequency	C=30pF,2.7V<Vdd<3.6V	-	45	MHz
			C=30pF,1.6V<Vdd<2.7V	-	22	
			C=10pF,2.7V<Vdd<3.6V	-	80	
			C=10pF,1.6V<Vdd<2.7V	-	40	
	Tr/Tf	Output rise and fall time	C=30pF,2.7V<Vdd<3.6V	-	8.7	ns
			C=30pF,1.6V<Vdd<2.7V	-	16.9	
			C=10pF,2.7V<Vdd<3.6V	-	3.4	
			C=10pF,1.6V<Vdd<2.7V	-	6.7	
FM+	Fmax	Maximum frequency	C=50pF,1.6V<Vdd<3.6V	-	10	MHz
	Tf	Output fall time		-	27	ns

表 3-24 引脚 AC 参数

注：

[1] 依据电路仿真，不在量产测试中测试

4 总线与存储

4.1 系统总线

FM33FG0xxA 总线架构包含以下主要部件：

- 两个 Master
  - Cortex-M0 内核
  - DMA 控制器
- 四个 Slave
  - 内部 Flash 存储器
  - 内部 SRAM 存储器
  - GPIO 控制器模块
  - AHB-APB 总线转接桥和系统控制寄存器

FM33FG0A 的系统总线示意图如下，包含一条 AHB-Lite 总线、一条 APB 总线。

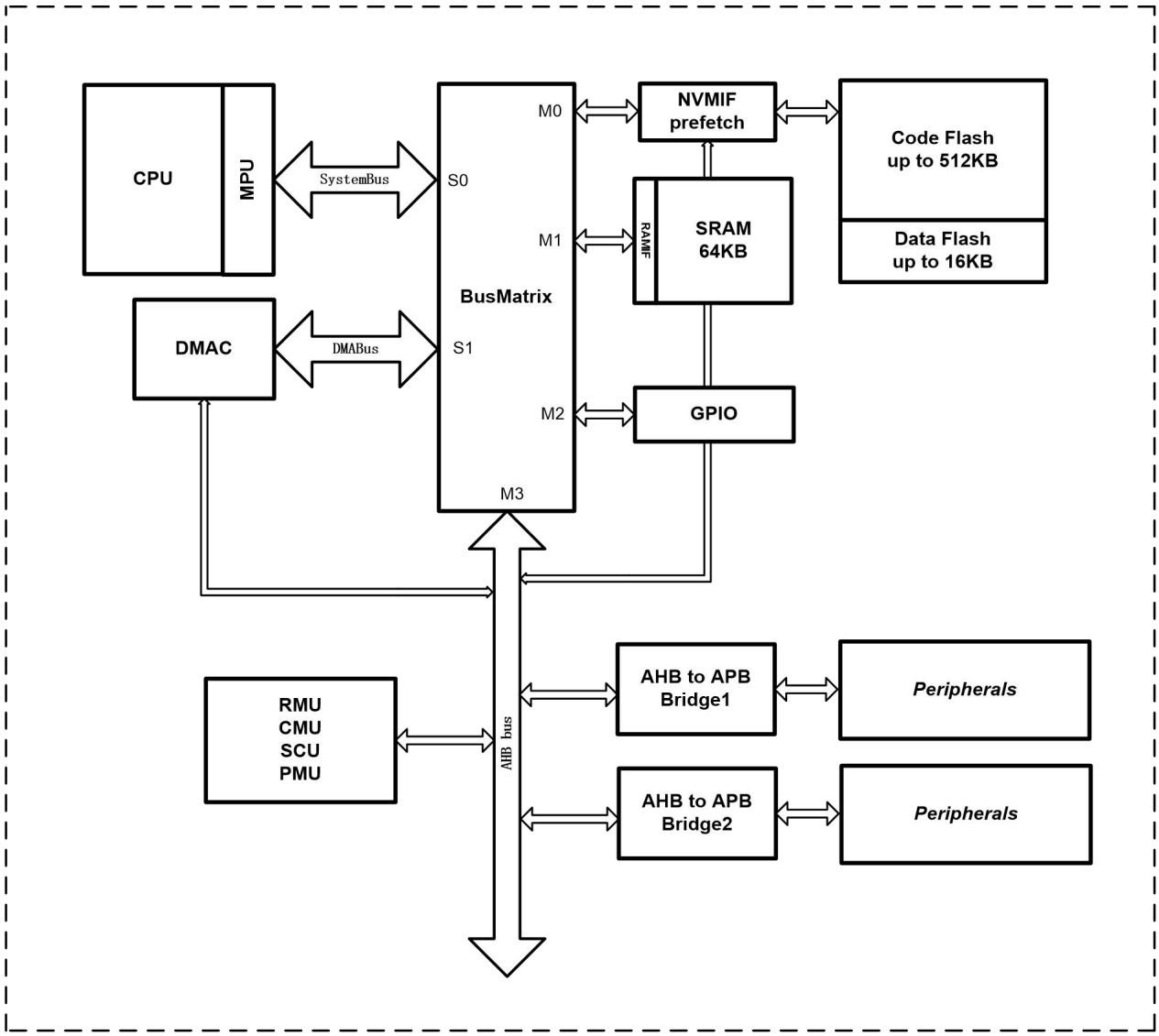

CPU
MPU
SystemBus
S0
M0
NVMIF
prefetch
Code Flash
up to 512KB
Data Flash
up to 16KB
RAMF
SRAM
64KB
M1
BusMatrix
DMAC
DMABus
S1
M2
GPIO
M3
AHB bus
RMU
CMU
SCU
PMU
AHB to APB
Bridge1
Peripherals
AHB to APB
Bridge2
Peripherals

图 4-1 系统总线示意图

4.2 存储空间分配

4.2.1 概述

Flash 页（page）大小为 512 字节，每 4 个 page 组成一个 2K 字节的扇区（sector）。

Flash 包含 4 个 information 扇区，information 为用户配置扇区，用于保存用户配置信息。所有 option 扇区在地址上与 Flash 主区域互相隔离。

当芯片从 Flash 启动时，FM33FG0A 的地址空间分配如下图（512KB code Flash，16KB data flash, 64KB RAM）：

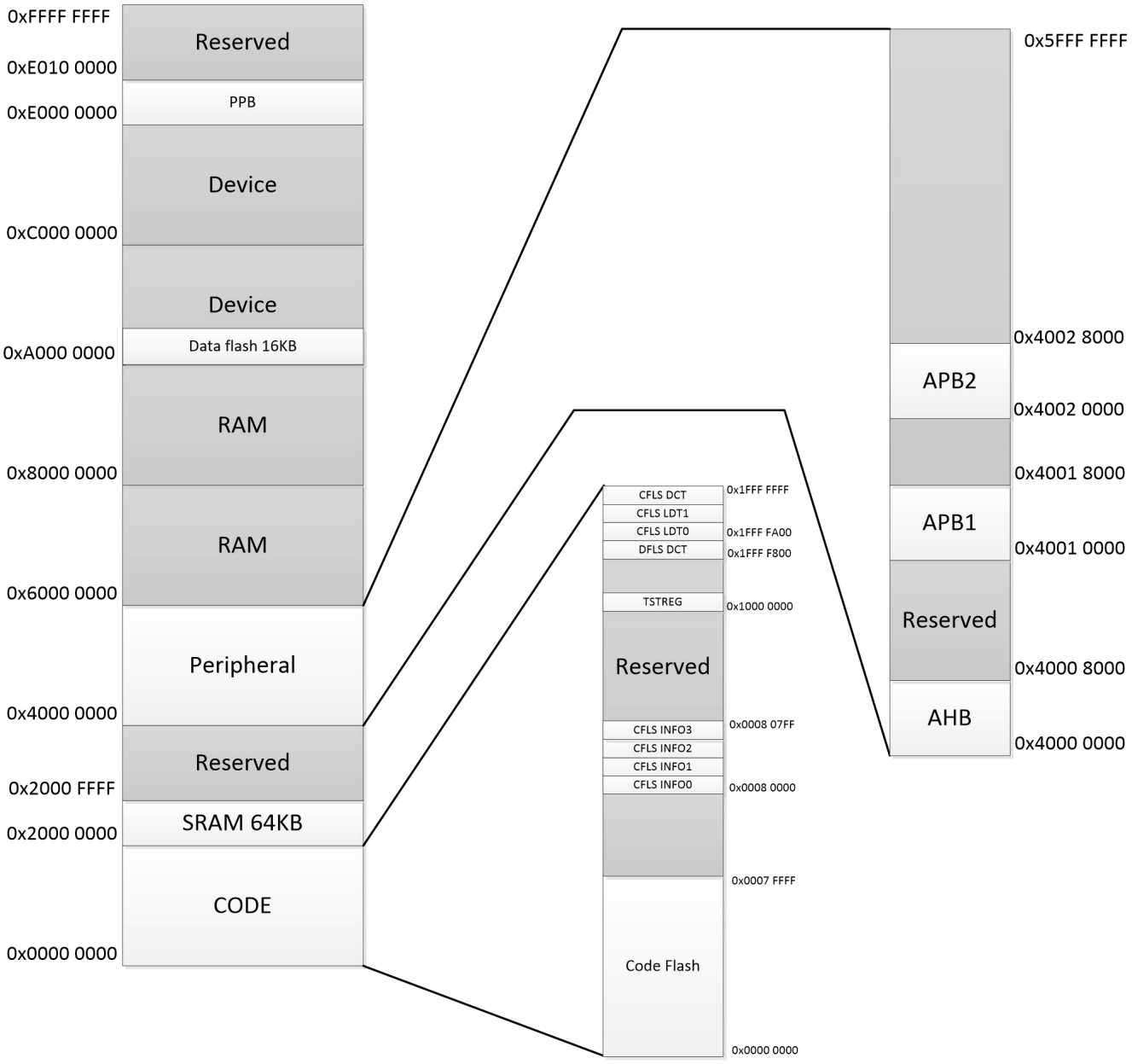


图 4-2FM33FG0A 总线地址

4.2.2 外设模块寄存器地址分配

下表罗列了所有外设模块的地址空间分配范围，每个外设模块占用 1KB 地址空间。

总线	起始地址	空间	外设
AHB	0x0000_0000	512KB	Code Flash
	0x1FFF_FE00	512B	Code Flash DCT cell array
	0x1FFF_FC00	512B	Code Flash LDT1 cell array
	0x1FFF_FA00	512B	Code Flash LDT0 cell array
	0x1FFF_F800	512B	Data Flash DCT cell array
	0x2000_0000	64KB	SRAM
	0x2000_8000	-	-
	0x4000_0000	1KB	SCU
	0x4000_0400	1KB	DMA
	0x4000_0800	1KB	RAMBIST
	0x4000_0C00	1KB	GPIO
	0x4000_1000	1KB	NVMIF
	0x4000_1400	-	-
	0x4000_1800	-	-
	0x4000_1C00	1KB	SFU
	0x4000_2000	1KB	PMU
	0x4000_2400	1KB	CMU
	0x4000_2800	1KB	RMU
	0x4001_0000	64KB	APB
	0xA000_0000	16KB	Data flash
APB1	0x4001_0000	1KB	CRC
	0x4001_0400	1KB	SPI0
	0x4001_0800	1KB	SPI1
	0x4001_0C00	1KB	
	0x4001_1000	1KB	RTC
	0x4001_1400	1KB	IWDT
	0x4001_1800	1KB	WWDT
	0x4001_1C00	1KB	
	0x4001_2000	1KB	UART0
	0x4001_2400	1KB	I2C0
	0x4001_2800	1KB	SVD
	0x4001_2C00	1KB	
	0x4001_3000	1KB	ATIM
	0x4001_3400	1KB	
	0x4001_3800	1KB	AES
	0x4001_3C00	1KB	TRNG
	0x4001_4000	1KB	LPUART0
	0x4001_4400	1KB	
	0x4001_4800	1KB	SPI2
	0x4001_4C00	1KB	GPTIM0
	0x4001_5000	1KB	LPUART2
	0x4001_5400	1KB	COMPx
	0x4001_5800	1KB	
	0x4001_5C00	1KB	ADC
	0x4001_6000	1KB	

	0x4001_6400	1KB	GPTIM1
	0x4001_6800	1KB	UART1
	0x4001_6C00	1KB	PGL
	0x4001_7000	1KB	UART3
	0x4001_7400	1KB	UART4
	0x4001_7800	1KB	UART5
	0x4001_7C00	1KB	UARTIR
	0x4001_8000	1KB	GPTIM2
	0x4001_8400	1KB	-
	0x4001_8800	1KB	LPTIM16
	0x4001_8C00	1KB	ANTEST(BUF4TST)
	0x4001_9000	1KB	DAC
	0x4001_9400	1KB	HDIV
	0x4001_9800	1KB	
	0x4001_9C00	1KB	I2C1
	0x4001_A000	1KB	I2CSMB0
	0x4001_A400	1KB	I2CSMB1
	0x4001_A800	1KB	SPI3
	0x4001_AC00	1KB	
	0x4001_B000	1KB	SENT0
	0x4001_B400	1KB	SENT1
	0x4001_B800	1KB	CLM0
	0x4001_BC00	1KB	CLM1
APB2	0x4002_0000	16KB	CAN-FD0
	0x4002_4000	16KB	CAN-FD1
	0x4002_8000	1KB	FSCAN
	0x4002_8400	1KB	
	0x4002_8800	1KB	

表 4-1 外设模块总线地址列表

4.3 RAM

4.3.1 概述

FM33FG0A 含有一块 64KB RAM，支持 ECC，可实现纠 1 检 2（SEC-DED）。RAM 实际数据位宽是 39 位，包含 32 位有效数据和 7bitECC。当总线以 32bit 访问时，ECC 逻辑自动计算和更新 ECC，读写在一个周期内完成。当总线以 8bit 或者 16bit 访问时，ECC 逻辑访问方式如下：

- 8bit 或者 16bit 读：实际读出 39bit，完成 ECC 运算后，将正确的 8/16bit 数据返回给总线
- 8bit 或者 16bit 写：硬件先读出 32bit 数据，将其中 8/16bit 替换成新数据，然后整体计算 ECC，最终将 39bit 数据写入 RAM，这个过程需要 2 个时钟周期

注意：上电复位后 RAM 中的数据为乱码，此时直接读取极有可能触发校验错误中断，因此软件需在上电后首先对 RAM 内容进行初始化。

4.4 Code Flash

4.4.1 概述

FM33FG0A 使用的 Code Flash 容量为 512KB，支持 ECC，可实现纠 1 检 2（SEC-DED）。

4.4.1.1 ECC

Code 和 data flash 各自有独立的 ECC 使能；上电复位后，data flash ECC 默认使能，code flash ECC 默认关闭。

注意：data flash ECC 和 code flash ECC 需同时使能以正确的产生 ECC 中断。

Flash ECC 中断：

- ECC detection 中断：检测到 2bit 错误，无法纠错，此时产生 NMI 中断，无中断屏蔽手段
- ECC correction 中断：纠正 1bit 错误，不影响正常运行，此时产生的中断可以屏蔽
- Data 和 code flash 可以分别产生 detection 和 correction 标志
- 错误产生时，ADDR_ECC 寄存器记录错误发生的 flash word 地址
- 访问未经编程的数据 flash 将导致 NMI 中断

4.4.2 特殊信息扇区说明

Flash IP 共有 4 个特殊扇区，说明如下

区域	说明	用途
LDT0	FMSH 数据区	保存 FMSH 的调校信息、模式字、测试数据等；软件只读
LDT1	用户选项数据区	用户选项字节（OPTBYTES）
IF	Information 区	4 个 page 共 2KB，供用户使用；软件可以读写

4.4.2.1 LDT0 page

LDT0为FMSH配置信息区，仅能在制造商模式下改写，用户模式下只读。

LDT0 的总线地址是 0x1FFF_FA00~0x1FFF_FBFF；以下参数软件在应用中可以从 LDT0 读取并写入对应的控制寄存器，以实现模拟参数校准。

AHB地址	物理地址 LDTEN[0]=1	[31:16]	[15:0]	说明
0x1FFF_FB08	0x0042	~AVREFCAL	AVREFCAL	3V,30C 下ADC对AVREF的转换值
0x1FFF_FB10	0x0044	~TS_CAL	TS_CAL	3V, 30C下ADC对温度传感器的转换值

0x1FFF_FB20	0x0048	~RCLP_TRIM	RCLP_TRIM	RCLP调校值
0x1FFF_FB34	0x0053	~RCHF32TRIM	RCHF32TRIM	RCHF 32MHz调校值
0x1FFF_FB38	0x0054	~RCHF24TRIM	RCHF24TRIM	RCHF 24MHz调校值
0x1FFF_FB3C	0x0055	~RCHF16TRIM	RCHF16TRIM	RCHF 16MHz调校值
0x1FFF_FB40	0x0056	~RCHF8TRIM	RCHF8TRIM	RCHF 8MHz 调校值(auto-load)

为了保证数据可靠，LDT0中的参数都采用高低半字互为正反码校验的方式保存。软件使用这些参数时，应先进行正反码校验，结果正确的情况下可以使用，否则应保持默认参数。

详细参数格式定义如下：

助记符	[31:16]	[15:0]	功能描述
VREFCAL	{4'h0, ~VREFCAL}	{4'hF, VREFCAL}	VDDA=VREFP=3V+/-10mV, 30C+/-1C 下ADC对AVREF的转换值
RCLP_TRIM	{8'h00, ~trim}	{8'hFF, trim}	trim[7:0]表示8bit调校值
RCHF32TRIM RCHF24TRIM RCHF16TRIM RCHF8TRIM	{9'b0000_0000_0, ~RCHFtrim[6:0]}	{9'b1111_1111_1,RCHFtrim[6:0]}	RCHFtrim[6:0]表示7bit调校值

4.4.2.2 LDT1 page

LDT1为用户配置信息区，可以在用户模式下改写（仅能使用SWD改写，即用户通过编程器改写）。

LDT1的总线地址是0x1FFF_FC00~0x1FFF_FDFF；

AHB addr	Bit[31:16]	Bit[15:0]	Description
0x1FFF_FC00	~OPTBYTES[15:0]	OPTBYTES[15:0]	用户选项字节低半字
0x1FFF_FC04	~OPTBYTES[31:16]	OPTBYTES[31:16]	用户选项字节高半字
0x1FFF_FC08	LOCK1		ACLOCK 配置字，控制最低 128KB flash
0x1FFF_FC0C	LOCK2		ACLOCK 配置字，控制中间 128KB flash
0x1FFF_FC10	LOCK3		ACLOCK 配置字，控制最高 128KB flash
0x1FFF_FC14	LOCK4		ACLOCK 配置字，控制最高 128KB flash

OPTBYTES 选项字节定义如下：

Bitfield	助记符	功能描述	出厂默认
31:24	IWDTEN	复位后 IWDT 默认状态 0x55：使能 IWDT 其他：关闭 IWDT，软件可以开启	0x55
23:20	IWDTSLP	配置 IWDT 在低功耗模式下是否允许停止计数 0xA：在 Sleep/DeepSleep 模式下允许应用停止	0xA

Bitfield	助记符	功能描述	出厂默认
		IWDT 计数 其他：任何模式下禁止应用停止 IWDT	
19:16	BDVEN	Backdoor code verify 使能 0xA：允许 backdoor code verify 其他：禁止 backdoor code verify	0xF
15:8	ACLKEN	应用代码保护使能 0x33：禁止 ACLOCK 其他：使能 ACLOCK	0x33
7:4	DBRDPEN	code flash 调试接口访问保护使能 0xA：关闭调试接口保护 其他：使能调试接口保护	0xA
3:0	DBRDPEN_D	data flash 调试接口访问保护使能 0xA：关闭调试接口保护 其他：使能调试接口保护	0xA

【注】在出厂时，SWD 接口可以任意改写 OPTBYTES；但是一旦 ACLKEN 或 DBRDP 被使能，用户必须通过 SWD 全擦 flash 后才能重新改写 OPTBYTES。

【注】Backdoor code verify 为后门功能，在 BDVEN 有效的情况下，用户可以通过验证后门秘钥解锁 flash 保护。后门秘钥保存在 security page 中且不可读，只有当后门秘钥验证成功后，security page 才可以擦除编程和读取。但是 backdoor code verify 功能不影响 OPTBYTES，即使成功验证了后门密钥，仍然需要 SWD 全擦后才能改写 OPTBYTES

LOCK 配置字节定义如下：

Bitfield	助记符	功能描述	出厂默认
31:0	LOCK1	Block Lock 字 1，每 2it 对应 8KB Block 11：无保护 01、10：软件读写保护，仅取指 00：软件读写保护，仅取指；SWD 读写保护 LOCK1[1:0]对应 Block0（Flash 最低地址 8KB 空间），LOCK1[31:30] 对应 Block15（Flash 地址空间 120~128KB），其他以此类推	0xFFFFFFFF
31:0	LOCK2	Block Lock 字 2，每 2it 对应 8KB Block 11：无保护 01、10：软件读写保护，仅取指 00：软件读写保护，仅取指；SWD 读写保护 LOCK2[1:0] 对应 Block16，LOCK2[31:30] 对应 Block31（Flash 地址空间 248~256KB），其他以此类推	0xFFFFFFFF
31:0	LOCK3	Block Lock 字 3，每 2it 对应 8KB Block 11：无保护 01、10：软件读写保护，仅取指 00：软件读写保护，仅取指；SWD 读写保护 LOCK3[1:0]对应 Block32（Flash 最低地址 8KB 空间），LOCK3[31:30]对应 Block47（Flash 地址空间 376~384KB），其他以此类推	0xFFFFFFFF

Bitfield	助记符	功能描述	出厂默认
31:0	LOCK4	Block Lock 字 4，每 2it 对应 8KB Block 11：无保护 01、10：软件读写保护，仅取指 00：软件读写保护，仅取指；SWD 读写保护 LOCK3[1:0]对应 Block32（Flash 最低地址 8KB 空间），LOCK3[31:30]对应 Block47（Flash 地址空间 376~384KB），其他以此类推	0xFFFFFFFF

4.4.2.3 Information1~3 page

这3个information扇区开放给用户作为数据区使用，SWD和软件均可以读写，共1.5KB。

IF1~3的总线地址是0x0008_0200 ~ 0x0008_07FF

4.4.2.4 Information0 page（security）

IF0是一个security page，这个page用于保存backdoor access code，在backdoor code verify成功前不可读取、不可擦写。IF0的总线地址是0x0008_0000~0x0008_01FF，共512字节。

IF0中保存8字节backdoor access code，在后门秘钥验证流程中，硬件自动读取code并与FLS_BVCRx寄存器比对，如果比对成功，则IF0解锁，可读可写，此时用户可以修改后门秘钥。

AHB address	flash data	default code on shippment	associated BVCR
0x0008_0000	32bit backdoor access code	0x5555 5555	BVCR0
0x0008_0004	32bit backdoor access code	0xAAAA AAAA	BVCR1

注意芯片出厂时需要写入默认code，用户在使用后门功能时，需要将每个芯片的access code修改为独立不重复的编码。

上电后后门访问密钥64bit自动load到寄存器里，这些寄存器软件、swd都是不可见的。后门访问比对直接针对寄存器中的数据进行。

4.4.3 Flash 编程

4.4.3.1 概述

FM33FG0A 支持以下 Flash 编程方法：

- 在系统编程（ISP）：通过 FMSH 专用编程器或者 KEIL 用户界面实施芯片编程，使用 SWD 接口
- 在应用编程（IAP）：通过 bootloader 代码实现芯片自编程，用户可定义任意串口，可用于实现程序在线升级

编程前必须对 Flash 进行擦除。Flash 支持三种擦除操作：全擦、扇区擦、页擦；注意数据 flash 不

受全擦操作影响。

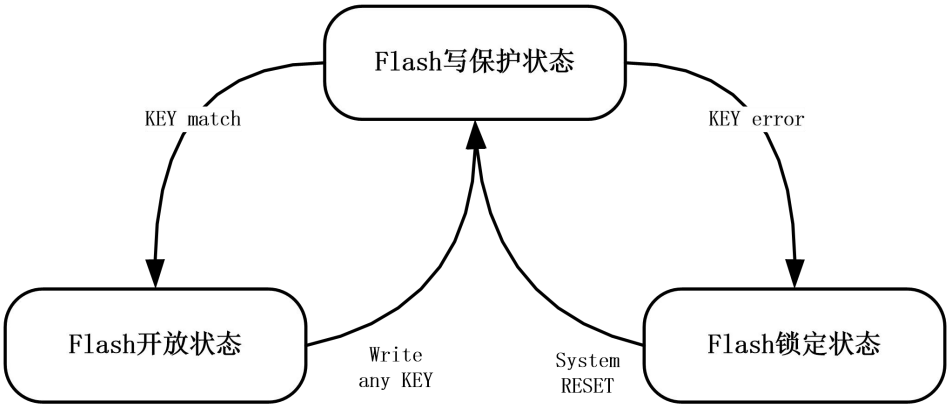
4.4.3.2 Flash 擦写时钟

执行 Flash 擦写时使用校准后的 RCHF 时钟，但是系统时钟可以是任意时钟。NVMIF 根据当前 RCHF 实际频率设置，来产生相应的计时长度。需要支持的 RCHF 频率为 8M、16M 和 24M。

4.4.3.3 Flash 擦写方法

FM33FG0A 支持 Flash 擦除操作，以及单次编程和连续编程。

Flash 擦写前须进行 Key 校验，写入顺序错误或写入值错误，或者在 Flash Key 验证正确之前就进行擦除或编程 Flash 操作将会进入错误状态，并产生相应中断。Flash Key 认证错误之后将禁止擦写 Flash 直到下一次复位。而在正常擦写完成后，向 KEY 寄存器写入任意值都会使状态机返回初始的写保护状态。状态转换如下图：



软件可以通过查询 FLSIF.KEYSTA 来确认当前 Key 输入状态，详情参见寄存器说明。

4.4.3.4 全擦操作（Matrix Erase）

全擦操作只能由 SWD 接口启动，软件禁止进行全擦。全擦操作仅擦除程序 flash。SWD 启动全擦操作流程如下：

- 编程器通过 SWD 配置 ERTYPE 寄存器为 10
- 编程器通过 SWD 清除 PREQ 寄存器，置位 EREQ 寄存器
- 编程器通过 SWD 写入 Flash 全擦 Key：0x9696_9696 和 0x7D7D_7D7D
- SWD 向 Flash 任意地址写擦除请求 0x1234_ABCD
- NVMIF 启动对 Flash 的全擦，并暂停任何 Master 对 Flash 的访问
- 全擦完成后置位中断标志和全擦标志（全擦标志表示程序 flash 全部擦除，任何对程序 flash 的编程将清除此标志）
- 软件确认擦除结束后向 FlashKEY 寄存器写任意值恢复写保护

注意：全擦操作只能擦除Flash的main array，不会影响Fuse Array（DCT/LDT/RED/IF）

4.4.3.5 扇区擦操作（Sector Erase）

SWD 和应用代码都可以执行扇区擦。操作流程如下：

- 配置 ERTYPE 寄存器为 00
- 清除 PREQ 寄存器，置位 EREQ 寄存器
- 写入 Flash 块擦 Key：0x9696_9696 和 0xEAEA_EAEA
- 向需要擦除的 Page 内任意地址写擦除请求 0x1234_ABCD
- NVMIF 检查目标扇区是否属于被 ACLOCK 锁定的 Block，如果没有锁定则启动对目标扇区的擦除，如果被锁定则触发错误标志
- 扇区擦完成后置位中断标志
- 软件确认擦除结束后向 FlashKEY 寄存器写任意值恢复写保护

4.4.3.6 页擦操作（Page Erase）

SWD 和应用代码都可以执行页擦。操作流程如下：

- 配置 ERTYPE 寄存器为 00 或 11
- 清除 PREQ 寄存器，置位 EREQ 寄存器
- 写入 Flash 块擦 Key：0x9696_9696 和 0xEAEA_EAEA
- 向需要擦除的扇区内任意地址写擦除请求 0x1234_ABCD
- NVMIF 检查目标扇区是否属于被 ACLOCK 锁定的 Block，如果没有锁定则启动对目标扇区的擦除，如果被锁定则触发错误标志
- 扇区擦完成后置位中断标志
- 软件确认擦除结束后向 FlashKEY 寄存器写任意值恢复写保护

4.4.3.7 单次编程

单次编程由软件发起，通过总线直接写 Flash，每次操作可执行 byte/half-word/word 编程，操作流程如下：

- 清除 EREQ 寄存器，置位 PREQ 寄存器
- 清除连续编程使能寄存器
- 写入 Flash 编程 Key：0xA5A5_A5A5 和 0xF1F1_F1F1
- 向 Flash 目标地址写数据，如果目标地址被 ACLOCK 锁定，则触发错误标志，如果没有锁定，

则执行编程

- Byte 写入直接完成，half-word 写入则由 NVMIF 自动执行 2 次 byte program，word 编程由 NVMIF 执行 4 次 byte program
- 编程完成后置位中断标志
- 软件确认编程结束后向 FlashKEY 寄存器写任意值恢复写保护

4.4.3.8 连续编程

连续编程指通过 DMA 的 Memory 通道一次向 Flash 写入 half-sector（256 字节）。连续编程时 DMA 从 RAM 指定地址读取数据，Flash 目标编程地址必须是 half-sector 对齐的，也就是 Flash 地址低 6 位为 0。采用这种方式时一次编程的数据长度是固定的，主要用于快速大数据量写入。

在启动连续编程期间，DMA 完全占据 Flash 总线，暂停 CPU 对 Flash 的一切访问。连续编程的操作流程如下：

- 清除 EREQ 寄存器，置位 PREQ 寄存器
- 置位连续编程使能寄存器（DMA 模式使能）
- 向 RAM 中写入 256 字节待编程数据
- 配置 DMA 存储器通道，设定传输方向、读地址和写地址
- 使能 DMA 存储器通道
- 写入 Flash 编程 Key：0xA5A5_A5A5 和 0xF1F1_F1F1
- 软件触发 DMA 存储器通道，DMA 连续 64 次读取 RAM 并对 Flash 编程
- NVMIF 检查被编程扇区是否被 ACLOCK 锁定，如果锁定则触发错误中断并通知 DMA 停止编程
- 256 字节完全编程结束后产生中断，释放 Flash 总线
- 软件确认编程结束后向 FlashKEY 寄存器写任意值恢复写保护

注意：如果 CPU 在 Flash 中取指时进行 Flash 擦写，则 CPU 取指将被暂停，直到擦写操作完成。如果 CPU 跳转到 RAM 中取指运行，则 Flash 擦写不会暂停 CPU 的执行。Flash 擦写过程中，若用户希望在 RAM 中执行代码时仍然能够实时响应中断，应将中断向量表重新映射到 RAM 中。

4.4.4 Flash 的内容保护

Flash 内容保护主要用于保护 Flash 中的用户代码、用户数据和用户配置信息被非授权方读取或篡改。

Flash 保护包含两种类型：Debug 接口读取保护（DBRDP-DeBug ReaD Protection）和应用代码权限保护（ACLOCK-Application Code Block Locking）。Flash 保护的控制通过 LDT1 中的 OPTBYTES

来控制。

4.4.4.1 Debug 接口保护（DBRDP）

DBRDP 的主要目的是防止非授权的第三方通过 debug 接口访问芯片 Flash 内容。

DBRDP 由 NVR0 扇区内的 DBRDPEN 配置字使能或者禁止（0xAA 表示禁止 DBRDP，芯片出厂时默认写为 0xAA）。当 DBRDP 使能时，无法通过 SWD 接口读取或擦写 Flash main array，同时也无法通过 SWD 接口对 RAM 进行访问。

退出 DBRDP 的方法：通过 SWD 对 flash 进行全空间擦写，全擦完成后，SWD 可以任意改写 OPTBYTES 禁止 DBRDP，然后复位芯片；复位完成后，芯片将处于无 debug 保护状态。

4.4.4.2 应用代码保护（ACLOCK）

ACLOCK 的主要目的是防止 hacking code 读取或篡改 Flash 中的 application code。通过 ACLOCK 功能，可以设置 CPU 对 Flash 的某些区域只能进行取指操作，不能 read-as-data，也不能擦写。

ACLOCK 以 Block 为单位工作，即对 Flash 保护的颗粒度是 8KB，整个 Flash 包含 32 个 Blocks，对应每个 Block 有 2bit LOCK 信息。出厂时默认的 LOCK 字为 0xFFFF_FFFF，上电 load 的 LOCK 位全部是 11，即默认的无保护状态；当对应 LOCK 位为 01 或 10 时，此 Block 禁止 CPU 擦写和读取，只能取指；当对应的 LOCK 位为 00 时，当前 Block 禁止 CPU 擦写和读取，同时禁止 SWD 擦写和读取。芯片出厂时 LDT0 中关闭 ACLOCK 功能，用户需要通过编程器使能 ACLOCK，并且用户代码编译时要符合 ACLOCK 配置（比如不能将 literal pool 编译到被 LOCK 的 Block）。

ACLOCK 的功能：

- 无保护：所有 Block 允许 CPU 取指、读取、改写，不限制 SWD 访问
- 读写保护：指定 Block 允许 CPU 取指，不允许 CPU 和 DMA 读取、擦写，不限制 SWD 访问
- 软件和 SWD 保护：指定 Block 允许 CPU 取指，不允许 CPU 和 DMA 读取、擦写，不允许 SWD 读取、擦写

LOCK 位与 Block 访问权限的关系可以参照下表：

LOCK bit	软件读取	软件取指	SWD 读取和擦写
11	允许	允许	允许
01/10	禁止	允许	允许
00	禁止	允许	禁止

ACLOCK 信息在芯片复位时 load 到寄存器中，这些寄存器软件也可以置位，但是不能写 0（即只能提升保护等级）。

ACLOCK 不使能时，LOCK 寄存器内容无效。

注：ACLOCK 的权限控制针对 Flash 各个 Block，与 DBRDP 相互独立。对于 SWD 接口而言，DBRDP 的优先级高于 ACLOCK，即 DBRDP 被使能后，不论 ACLOCK 是否起效，SWD 都无法访问 Flash。

注：不建议 ACLOCK 禁止 1st block 读取，由于 CPU 复位后首先要从 0 地址读取 MSP 指针，ACLOCK 将导致 CPU 无法正常启动。

退出 ACLOCK 的方法：通过 SWD 对 flash 进行全空间擦写，全擦完成后，SWD 可以任意改写 OPTBYTES 禁止 ACLOCK，然后复位芯片；复位完成后，芯片将处于无 ACLOCK 状态。

4.4.4.3 用户模式 Flash 访问权限说明

Flash 空间访问权限分配：

Flash area	DBRDP	LOCK bits (per Block)[3]	Last byte in page	SWD	Application
Main array	ON	00	x	-	R/E/W/F
		01	x	-	对应 Block 只能取指
		10/11	x	-	对应 Block 只能取指
	OFF	00	x	R/E/W	R/E/W/F
		01	x		对应 Block 只能取指
		10/11	x	对应 Block 无法访问	对应 Block 只能取指
DCT LDT0	x	x	x	R	R
LDT1	ON	x	x	R[2]	R
	OFF	x	x	R/E/W	R
IF1,2,3	x	x	x	R/E/W	R/E/W
IF0	x	x	55	R/E	R
			others	R/E/W	R

注：

[1] R: Read, E: Erase, W: Write, F: Fetch

[2] 进行 flash 全擦后可以擦除 LDT1

[3] 这里假设 ACLOCKEN 有效。ACLOCKEN 无效的情况下，LOCK bits 不起作用。

4.4.4.4 后门访问密钥（Backdoor access code）

后门访问密钥用于临时解锁 flash 访问保护，当密钥验证通过后，可以改写 FLS_OPTBR 寄存器中

的 ACLOCKEN 和 DBRDPEN 寄存器，此时可以解除 ACLOCK 锁定和 debugger 接口保护，不管是 SWD 还是软件都可以读取和擦写 flash。

后门秘钥保存在 flash 的 security page 中，共 8 字节。软件或 SWD 将待比对密钥写入后门秘钥寄存器，然后使能后门秘钥验证，如果比对失败，后门验证功能被锁死，直到芯片复位，如果比对成功，则 ACLOCKEN 和 DBRDPEN 寄存器变成可写。

后门密钥禁止全 0 全 1 格式，如果输入密钥为全 0 全 1，则不执行比对直接锁死。

后门密钥验证流程：

- 向 FLS_BDCRx 寄存器写入 8 字节密钥
- 向 FLS_BDVR 寄存器写入 0x55AA1357 启动验证
- 验证结束后，FLS_BDSR 寄存器中 DONE 标志置位，验证是否成功可以查询 FLAG 标志
- 如果 DONE=1&& FLAG=0，验证成功，软件可以改写 ACLOCKEN 和 DBRDPEN；如果 DONE=1 && FLAG=1，验证失败，在芯片复位前无法再次进行验证操作

4.5 Data Flash

4.5.1 概述

FM33FG0A 包含 16KB 数据 flash，擦写寿命可达 50 万次，读取周期为 80ns。由于 code 和 data flash 物理独立，因此 CPU 可以在取指的同时访问 data flash。CPU 对 data flash 只能进行数据访问，不能执行取指操作（XN 属性），对 data flash 的取指操作将触发 hardfault。

上电后默认主频为 8Mhz 的情况下可以实现 data flash 单周期访问，当主频提高后，软件必须正确配置 data flash wait cycle 来保证正确的读取 data flash。

Data flash 被映射到总线的 device 区域，起始地址为 0xA000 0000。

Data flash 可以由 CPU 或 SWD 访问，对 data flash 进行擦写前必须先使能 data flash 编程控制寄存器。通过 option byte 配置，也可以禁止 SWD 接口对 data flash 的读取或擦写权限。

4.5.2 Debug 接口保护（DBRDP_D）

DBRDP 的主要目的是防止非授权的第三方通过 debug 接口访问芯片 data Flash 内容。

DBRDP 由 NVR0 扇区内的 DBRDPEN_D 配置字使能或者禁止（0xA 表示禁止 DBRDP，芯片出厂时默认写为 0xA）。当 DBRDP_D 使能时，无法通过 SWD 接口读取或擦写 data Flash。

退出 DBRDP 的方法：

1） 通过 SWD 对 flash 进行全空间擦写，全擦完成后，SWD 可以任意改写 OPTBYTES 禁止 DBRDP，然后复位芯片；复位完成后，芯片将处于无 debug 保护状态。

2） 使用后门访问密钥，验证成功后可以直接改写 DBRDP_D 寄存器，以解除保护

4.5.3 data Flash 编程

data flash 编程方法与程序 flash 类似

FM33FG0A 支持以下 Data Flash 编程方法：

- 在系统编程（ISP）：通过 FMSH 专用编程器或者 KEIL 用户界面实施芯片编程，使用 SWD 接口
- 在应用编程（IAP）：通过 CPU 代码实现芯片自编程

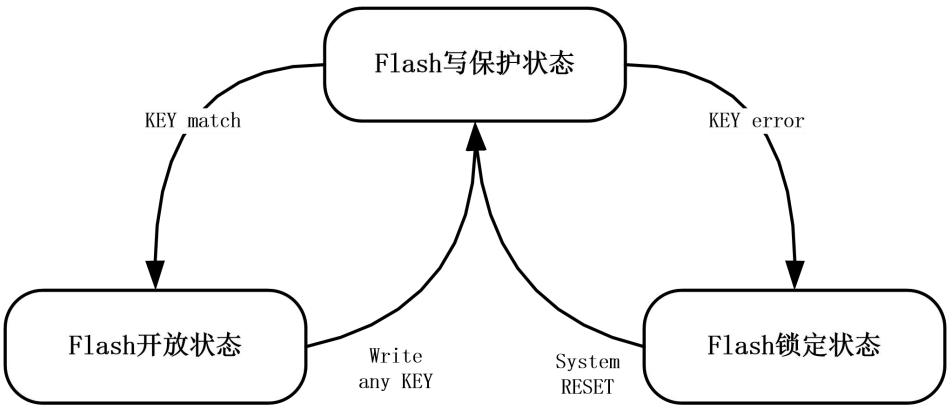
编程前必须对 Flash 进行擦除。Flash 支持三种擦除操作：全擦、扇区擦、页擦

执行 Flash 擦写时使用校准后的 RCHF 时钟，但是系统时钟可以是任意时钟。NVMIF 根据当前 RCHF 实际频率设置，来产生相应的计时长度。需要支持的 RCHF 频率为 8M、16M 和 24M。

擦写时钟独立于 CPU 时钟，两者之间作为异步时钟处理。

FM33FG0A 支持 Flash 擦除操作，以及单次编程和连续编程。

Flash 擦写前须进行 Key 校验，写入顺序错误或写入值错误，或者在 Flash Key 验证正确之前就进行擦除或编程 Flash 操作将会进入错误状态，并产生相应中断。Flash Key 认证错误之后将禁止擦写 Flash 直到下一次复位。而在正常擦写完成后，向 KEY 寄存器写入任意值都会使状态机返回初始的写保护状态。状态转换如下图：



软件可以通过查询 FLSIF.KEYSTA 来确认当前 Key 输入状态，详情参见寄存器说明。

4.5.3.1 扇区擦操作（Sector Erase）

SWD 和应用代码都可以执行扇区擦。操作流程如下：

- 配置 ERTYPE 寄存器为 00
- 清除 PREQ 寄存器，置位 EREQ 寄存器
- 写入 Flash 块擦 Key：0x9696_9696 和 0xEAEA_EAEA
- 向需要擦除的 Page 内任意地址写擦除请求 0x1234_ABCD
- NVMIF 检查目标扇区是否属于被 ACLOCK 锁定的 Block，如果没有锁定则启动对目标扇区的擦除，如果被锁定则触发错误标志
- 扇区擦完成后置位中断标志
- 软件确认擦除结束后向 FlashKEY 寄存器写任意值恢复写保护

注意：当main array中某个sector被冗余修复时，对此sector的擦写实际被映射到冗余扇区

4.5.3.2 页擦操作（Pgae Erase）

SWD 和应用代码都可以执行页擦。操作流程如下：

- 配置 ERTYPE 寄存器为 00 或 11
- 清除 PREQ 寄存器，置位 EREQ 寄存器
- 写入 Flash 块擦 Key：0x9696_9696 和 0xEAEA_EAEA
- 向需要擦除的扇区内任意地址写擦除请求 0x1234_ABCD
- NVMIF 检查目标扇区是否属于被 ACLOCK 锁定的 Block，如果没有锁定则启动对目标扇区的擦除，如果被锁定则触发错误标志
- 扇区擦完成后置位中断标志
- 软件确认擦除结束后向 FlashKEY 寄存器写任意值恢复写保护

注意：当main array中某个page处于被修复的扇区中时，对此page的擦写将被映射到冗余扇区中的对应页

4.5.3.3 单次编程

单次编程由软件发起，通过总线直接写 Flash，每次操作可执行 byte/half-word/word 编程，操作流程如下：

- 清除 EREQ 寄存器，置位 PREQ 寄存器
- 清除连续编程使能寄存器
- 写入 Flash 编程 Key：0xA5A5_A5A5 和 0xF1F1_F1F1
- 向 Flash 目标地址写数据，如果目标地址被 ACLOCK 锁定，则触发错误标志，如果没有锁定，则执行编程
- Byte 写入直接完成，half-word 写入则由 NVMIF 自动执行 2 次 byte program，word 编程由 NVMIF 执行 4 次 byte program
- 编程完成后置位中断标志
- 软件确认编程结束后向 FlashKEY 寄存器写任意值恢复写保护

注意，数据 flash 不支持连续编程

4.6 寄存器

模块起始地址：0x4000_1000

offset 地址	名称	符号
0x00	Flash 读取控制寄存器 （Flash Read Control Register）	FLS_RDCR
0x08	用户配置字寄存器 （Flash Option Bytes Register）	FLS_OPTBR
0x14	CFlash 擦写控制寄存器 （ Program Flash Erase/Program Control Register）	CFLS_EPCR
0x18	CFlash Key 输入寄存器 （Program Flash Key Register）	CFLS_KEY
0x1C	CFlash 中断使能寄存器 （Program Flash Interrupt Enable Register）	CFLS_IER
0x20	CFlash 标志寄存器 （Program Flash Interrupt Status Register）	CFLS_ISR
0x28	Code Flash ECC 寄存器	CFLS_ECCR
0x2C	Data Flash ECC 寄存器	DFLS_ECCR
0x30	RAM ECC 寄存器	RAM_ECCR
0x50	HSRAM 配置寄存器 1	HSRAM1_CFGR
0x54	HSRAM 配置寄存器 2	HSRAM2_CFGR
0x60	ACLOCK 寄存器 1 （Flash Application Code Lock Register1）	FLS_ACLOCK1
0x64	ACLOCK 寄存器 2 （Flash Application Code Lock Register2）	FLS_ACLOCK2
0x68	ACLOCK 寄存器 3 （Flash Application Code Lock Register3）	FLS_ACLOCK3
0x6C	ACLOCK 寄存器 4 （Flash Application Code Lock Register4）	FLS_ACLOCK4
0x90	DFlash 擦写控制寄存器 （ Data Flash Erase/Program Control Register）	DFLS_EPCR
0x94	DFlash Key 输入寄存器 （Data Flash Key Register）	DFLS_KEY
0x98	DFlash 中断使能寄存器 （Data Flash Interrupt Enable Register）	DFLS_IER
0x9C	DFlash 标志寄存器 （Data Flash Interrupt Status Register）	DFLS_ISR
0x100	RAM 原始数据寄存器 0	RAM_RAW0
0x104	RAM 原始数据寄存器 1	RAM_RAW1

0x108	DFlash 原始数据寄存器 0	DFLS_RAW0
0x10C	DFlash 原始数据寄存器 1	DFLS_RAW1
0x200	后门访问密钥验证寄存器	FLS_BDVR
0x204	后门访问密钥输入寄存器 0	FLS_BDCR0
0x208	后门访问密钥输入寄存器 1	FLS_BDCR1
0x20C	后门访问密钥状态寄存器	FLS_BDSR

4.6.1 Flash 读取控制寄存器（FLS_RDCR）

名称	FLS_RDCR							
offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DATA_WAIT				CODE_WAIT			
位权限	R/W-0000				R/W-0000			

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:4	DATA_WAIT	Data Flash 读等待周期配置 0000：无等待 0001：1 wait cycle 0010：2 wait cycles 0011：3 wait cycles … 1111：15 wait cycles Data flash 实际读周期为 DATA_WAIT 个 AHB 时钟周期 Data Flash 最小读周期为 80ns，软件应根据当前系统时钟频率合理配置 DATA_WAIT，确保读周期不短于 80ns 比如，当系统频率 AHBCLK 为 24MHz，对应时钟周期为 41.6ns，则读取数据 flash 至少需要 2 个周期，可以配置 DATA_WAIT 为大于等于 2 的数。
3:0	CODE_WAIT	Code Flash 读等待周期配置 (Wait Cycles) 0000：0 wait cycle 0001：1 wait cycle 0010：2 wait cycles 0011：3 wait cycles … 1111：15 wait cycles CPU 主频小于等于 24MHz 时，不需要开启 wait；主频大于 24M 小于 48Mhz 时使能 1 wait，主频大于 48Mhz 时使能 2 wait

4.6.2 用户配置字寄存器（FLS_OPTBR）

名称	FLS_OPTBR							
offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IWDTSLP	IWDTEN	-					
位权限	R-0	R-1	U-0					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		DBRDPEN_D		ACLOCKEN		DBRDPEN_C	
位权限	U-0		R/W-01		R/W-01		R/W-01	

位号	助记符	功能描述
31	IWDTSLP	IWDT 在休眠模式下是否允许应用暂停计数 (IWDT Sleep) 1：允许应用在休眠模式下暂停 IWDT 计数 0：禁止应用在休眠模式下暂停 IWDT 计数
30	IWDTEN	IWDT 使能 1：上电后 IWDT 开始运行 0：上电后 IWDT 不运行
29:6	-	RFU：未实现，读为 0
5:4	DBRDPEN_D	数据 flash Debug Port 读取保护使能 (Debug Read Protection Enable) 00/01/11：DBRDP 不使能 10：DBRDP 使能 注：正常情况下只读，不可改写；但是后门访问验证成功后可以改写
3:2	ACLOCKEN	应用代码权限锁定使能 (AppCode Lock Enable) 00/01/11：ACLOCK 不使能 10：ACLOCK 使能 注：正常情况下只读，不可改写；但是后门访问验证成功后可以改写
1:0	DBRDPEN_C	程序 flash Debug Port 读取保护使能 (Debug Read Protection Enable) 00/01/11：DBRDP 不使能 10：DBRDP 使能 注：正常情况下只读，不可改写；但是后门访问验证成功后可以改写

4.6.3 ACLOCK 寄存器 1（FLS_ACLOCK1）

名称	FLS_ACLOCK1							
offset	0x60							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK1[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LOCK1[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LOCK1[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCK1[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	LOCK1	ACLOCK 配置寄存器 1，分别用于控制 Block15~Block0 的应用代码读写锁定。每个 Block 大小为 8KB，每个 Block 使用 2bit 进行权限控制。(Lock bits) 11：当前 Block 允许 SWD 和软件读写 01/10：当前 Block 允许 SWD 读写，禁止软件读写，软件可以取指 00：当前 Block 禁止 SWD 读写，禁止软件读写，软件可以取指 所有 bit 软件只能写 0，不能写 1。

4.6.4 ACLOCK 寄存器 2（FLS_ACLOCK2）

名称	FLS_ACLOCK2							
offset	0x64							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK2[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LOCK2[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LOCK2[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCK2[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	LOCK2	ACLOCK 配置寄存器 2，分别用于控制 Block31~Block16 的应用代码读写锁定。每个 Block 大小为 8KB，每个 Block 使用 2bit 进行权限控制。(Lock bits)

位号	助记符	功能描述
		11：当前 Block 允许 SWD 和软件读写 01/10：当前 Block 允许 SWD 读写，禁止软件读写，软件可以取指 00：当前 Block 禁止 SWD 读写，禁止软件读写，软件可以取指 所有 bit 软件只能写 0，不能写 1。

4.6.5 ACLOCK 寄存器 3（FLS_ACLOCK3）

名称	FLS_ACLOCK3							
offset	0x68							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK3[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LOCK3[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LOCK3[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCK3[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	LOCK3	ACLOCK 配置寄存器 3，分别用于控制 Block47~Block32 的应用代码读写锁定。每个 Block 大小为 8KB，每个 Block 使用 2bit 进行权限控制。(Lock bits) 11：当前 Block 允许 SWD 和软件读写 01/10：当前 Block 允许 SWD 读写，禁止软件读写，软件可以取指 00：当前 Block 禁止 SWD 读写，禁止软件读写，软件可以取指 所有 bit 软件只能写 0，不能写 1。

4.6.6 ACLOCK 寄存器 4（FLS_ACLOCK4）

名称	FLS_ACLOCK4							
offset	0x6C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK4[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LOCK4[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LOCK4[15:8]							

位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCK4[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	LOCK4	ACLOCK 配置寄存器 4，分别用于控制 Block63~Block48 的应用代码读写锁定。每个 Block 大小为 8KB，每个 Block 使用 2bit 进行权限控制。(Lock bits) 11：当前 Block 允许 SWD 和软件读写 01/10：当前 Block 允许 SWD 读写，禁止软件读写，软件可以取指 00：当前 Block 禁止 SWD 读写，禁止软件读写，软件可以取指 所有 bit 软件只能写 0，不能写 1。

4.6.7 CFlash 擦写控制寄存器（CFLS_EPCR）

名称	CFLS_EPCR							
offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						ERTYPE	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PREQ	EREQ
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9:8	ERTYPE	程序 Flash 擦除类型配置 (Erase Type) 00/11：Page Erase 01：Sector Erase 10：Chip Erase (SWD only)
7:2	-	RFU：未实现，读为 0
1	PREQ	Program Request 软件置位，硬件完成编程后自动清零 软件置位后，在编程没有开始的情况下，软件可以清零撤销编程请求
0	EREQ	Erase Request 软件置位，硬件完成擦除后自动清零 软件置位后，在擦除没有开始的情况下，软件可以清零撤销擦除请求

4.6.8 CFlash Key 输入寄存器（CFLS_KEY）

名称	CFLS_KEY							
offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	KEY[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	KEY[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	KEY[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	KEY[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:0	KEY	程序 Flash 擦写 Key 输入寄存器，软件或者 SWD 在启动擦写前必须正确地向此地址写入合法 KEY 序列。 (Flash Key)

4.6.9 CFlash 中断使能寄存器（CFLS_IER）

名称	CFLS_IER							
offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				RFUI	AUTHIE	KEYIE	CKIE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PRDIE	ERDIE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11	RFUI	保留位，可读写
10	AUTHIE	Flash 读写权限错误中断使能，1 有效 (Flash Authentication Error Interrupt Enable)
9	KEYIE	Flash KEY 错误中断使能，1 有效 (Flash Key Error Interrupt Enable)
8	CKIE	擦写定时时钟错误中断使能，1 有效 (Erase/Program Clock Error Interrupt Enable)
7:2	-	RFU：未实现，读为 0
1	PRDIE	编程完成标志中断使能，1 有效 (Program Done Interrupt

位号	助记符	功能描述
		Enable)
0	ERDIE	擦写完成标志中断使能，1 有效 (Erase Done Interrupt Enable)

4.6.10 CFlash 标志寄存器（CFLS_ISR）

名称	CFLS_ISR							
offset	0x00000020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				KEYSTA			-
位权限	U-0				R-000			U-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					AUTHERR	KEYERR	CKERR
位权限	U-0					R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PRD	ERD
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:20	-	RFU：未实现，读为 0
19:17	KEYSTA	Flash 擦写 KEY 输入状态 (Flash Key Status) 000：Flash 写保护状态，未输入 KEY 001：全擦解锁状态 010：页擦解锁状态 011：编程解锁状态 100：KEY 错误锁定状态，需要复位才能解锁 101：扇区擦解锁状态 110/111：RFU
16:11	-	RFU：未实现，读为 0
10	AUTHERR	Flash 读写权限错误，读取 LOCK 块数据或对 LOCK 块擦写时置位，软件写 1 清零。(Flash Authentication Error) 1：Flash 访问权限错误 0：Flash 访问没有发生权限错误
9	KEYERR	Flash KEY 错误，硬件置位，软件写 1 清零 (Flash Key Error)
8	CKERR	擦写定时时钟错误，NVMIF 擦写 Flash 时如果 RCHF 未使能，则触发 CKERR 中断，软件写 1 清零。(Erase/Program Clock Error)
7:2	-	RFU：未实现，读为 0
1	PRD	Program Done，编程完成标志，硬件置位，软件写 1 清零
0	ERD	Erase Done，擦写完成标志，硬件置位，软件写 1 清零

4.6.11 Dflash 擦写控制寄存器（DFLS_EPCR）

名称	DFLS_EPCR
offset	0x90

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						ERTYPE	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PREQ	EREQ
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9:8	ERTYPE	DFlash 擦除类型配置（Erase Type） 00/11：Page Erase 01：Sector Erase 10：Chip Erase (SWD only)
7:2	-	RFU：未实现，读为 0
1	PREQ	Program Request 软件置位，硬件完成编程后自动清零 软件置位后，在编程没有开始的情况下，软件可以清零撤销编程请求
0	EREQ	Erase Request 软件置位，硬件完成擦除后自动清零 软件置位后，在擦除没有开始的情况下，软件可以清零撤销擦除请求

4.6.12 Dflash Key 输入寄存器（DFLS_KEY）

名称	DFLS_KEY							
offset	0x94							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	KEY[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	KEY[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	KEY[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	KEY[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:0	KEY	DFlash 擦写 Key 输入寄存器，软件或者 SWD 在启动擦写前必须正确地向此地址写入合法 KEY 序列。 (DFlash Key)

4.6.13 Dflash 中断使能寄存器（DFLS_IER）

名称	DFLS_IER							
offset	0x98							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				-	AUTHIE	KEYIE	CKIE
位权限	U-0				U-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PRDIE	ERDIE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11	-	RFU：未实现，读为 0
10	AUTHIE	Flash 读写权限错误中断使能，1 有效 (Flash Authentication Error Interrupt Enable) *注：该位可写，不可读，请参考勘误表*
9	KEYIE	Flash KEY 错误中断使能，1 有效 (Flash Key Error Interrupt Enable)
8	CKIE	擦写定时时钟错误中断使能，1 有效 (Erase/Program Clock Error Interrupt Enable)
7:2	-	RFU：未实现，读为 0
1	PRDIE	编程完成标志中断使能，1 有效 (Program Done Interrupt Enable)
0	ERDIE	擦写完成标志中断使能，1 有效 (Erase Done Interrupt Enable)

4.6.14 Dflash 标志寄存器（DFLS_ISR）

名称	DFLS_ISR							
offset	0x9C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				KEYSTA			-
位权限	U-0				R-000			U-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				-	AUTHERR	KEYERR	CKERR
位权限	U-0				U-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PRD	ERD

位权限	U-0	R/W-0	R/W-0

位号	助记符	功能描述
31:20	-	RFU：未实现，读为 0
19:17	KEYSTA	DFlash 擦写 KEY 输入状态 (Flash Key Status) 000：Flash 写保护状态，未输入 KEY 001：全擦解锁状态 010：页擦解锁状态 011：编程解锁状态 100：KEY 错误锁定状态，需要复位才能解锁 101：扇区擦解锁状态 110/111：RFU
16:12	-	RFU：未实现，读为 0
11	-	RFU：未实现，读为 0
10	AUTHERR	Flash 读写权限错误，读取 LOCK 块数据或对 LOCK 块擦写时置位，软件写 1 清零。(Flash Authentication Error) 1：Flash 访问权限错误 0：Flash 访问没有发生权限错误
9	KEYERR	Flash KEY 错误，硬件置位，软件写 1 清零 (Flash Key Error)
8	CKERR	擦写定时时钟错误，NVMIF 擦写 DFlash 时如果 RCHF 未使能，则触发 CKERR 中断，软件写 1 清零。(Erase/Program Clock Error)
7:2	-	RFU：未实现，读为 0
1	PRD	Program Done，编程完成标志，硬件置位，软件写 1 清零
0	ERD	Erase Done，擦写完成标志，硬件置位，软件写 1 清零

4.6.15 Code Flash ECC 寄存器（CFLS_ECCR）

名称	CFLS_ECCR							
offset	0x00000028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ECCDD	ECCCD	-	-	RFUI	-		ECCCIE
位权限	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							ADDR_ECC
位权限	U-0							R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ADDR_ECC							
位权限	R-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADDR_ECC							
位权限	R-0							

位号	助记符	功能描述
31	ECCDD	Code flash ECC 检错标志 (ECC Detection in Code flash)，硬件置位，软件写 1 清零 1：Code flash 中检测到 2bit 错误，无法纠正 0：没有检测到 2bit 错误

位号	助记符	功能描述
30	ECCCD	Code flash ECC 纠错标志(ECC Correction in Code flash)，硬件置位，软件写 1 清`零 1：Code flash 中纠正 1bit 错误 0：没有错误
29	-	RFU：未实现，读为 0
28	-	RFU：未实现，读为 0
27	RFUI	保留位
26	-	RFU：未实现，读为 0
25	-	RFU：未实现，读为 0
24	ECCCIE	ECC 纠错中断使能 1：允许 Code Flash ECCCD 产生中断 0：禁止纠错中断
23:17	-	RFU：未实现，读为 0
16:0	ADDR_ECC	ECC 错误地址，只读

4.6.16 Data Flash ECC 寄存器（DFLS_ECCR）

名称	DFLS_ECCR							
offset	0x0000002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ECCDD	ECCCD	-	-	RFUI	-		ECCCIE
位权限	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				ADDR_ECC			
位权限	U-0				R-0			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADDR_ECC							
位权限	R-0							

位号	助记符	功能描述
31	ECCDD	Data flash ECC 检错标志 (ECC Detection in Data flash)，硬件置位，软件写 1 清零 1：数据 flash 中检测到 2bit 错误，无法纠正 0：没有检测到 2bit 错误
30	ECCCD	Data flash ECC 纠错标志(ECC Correction in Data flash)，硬件置位，软件写 1 清`零 1：数据 flash 中纠正 1bit 错误 0：没有错误
29	-	RFU：未实现，读为 0
28	-	RFU：未实现，读为 0
27	RFUI	保留位
26	-	RFU：未实现，读为 0
25	-	RFU：未实现，读为 0
24	ECCCIE	ECC 纠错中断使能

位号	助记符	功能描述
		1：允许 Data Flash ECCCD 产生中断 0：禁止纠错中断
23:12	-	RFU：未实现，读为 0
11:0	ADDR_ECC	ECC 错误地址，只读

4.6.17 RAM ECC 寄存器（RAM_ECCR）

名称	RAM_ECCR							
offset	0x30							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ECCDD	ECCCD	-	-	ECCBE	-		ECCCIE
位权限	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		ADDR_ECC					
位权限	U-0		R-0					
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADDR_ECC							
位权限	R-0							

位号	助记符	功能描述
31	ECCDD	RAM ECC 检错标志 (ECC Detection in RAM)，硬件置位，软件写 1 清零 1：RAM 中检测到 2bit 错误，无法纠正 0：没有检测到 2bit 错误
30	ECCCD	RAM ECC 纠错标志(ECC Correction in RAM)，硬件置位，软件写 1 清零 1：RAM 中纠正 1bit 错误 0：没有错误
29	-	RFU：未实现，读为 0
28	-	RFU：未实现，读为 0
27	ECCBE	ECC bit error 标志 当 ECC 检测发现数据正确但是 ECC 位错误时，此寄存器置位此时 ECC 也会产生检错标志，但是 32bit 数据本身仍是正确的可以使用
26	-	RFU：未实现，读为 0
25	-	RFU：未实现，读为 0
24	ECCCIE	ECC 纠错中断使能 1：允许 ECCCD 和 ECCCC 产生中断 0：禁止纠错中断
23:14	-	RFU：未实现，读为 0
13:0	ADDR_ECC	ECC 错误地址，只读

4.6.18 RAM 原始数据寄存器 0(RAM_RAW0)

名称	RAM_RAW0							
地址	0x100							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RAWDATA							
位权限	R-x							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RAWDATA							
位权限	R-x							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RAWDATA							
位权限	R-x							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RAWDATA							
位权限	R-x							

Bit	助记符	功能描述
31:0	RAWDATA	RAM 读出的 32bit 原始数据（未经纠错）

4.6.19 RAM 原始数据寄存器 1(RAM_RAW1)

名称	RAM_RAW1							
地址	0x104							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	RAWECC						
位权限	U-0	R-x						

Bit	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6:0	RAWECC	RAM 读出的 7bit ECC 原始数据（未经纠错）

4.6.20 DFlash 原始数据寄存器 0(DFLS_RAW0)

名称	DLFS_RAW0							
地址	0x108							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RAWDATA							
位权限	R-x							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RAWDATA							
位权限	R-x							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RAWDATA							
位权限	R-x							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RAWDATA							
位权限	R-x							

Bit	助记符	功能描述
31:0	RAWDATA	DFlash 读出的 32bit 原始数据（未经纠错）

4.6.21 DFlash 原始数据寄存器 1(DLFS_RAW1)

名称	DFLS_RAW1							
地址	0x10C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	RAWECC						
位权限	U-0	R-x						

Bit	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6:0	RAWECC	DFlash 读出的 7bit ECC 原始数据（未经纠错）

4.6.22 HSRAM1 配置寄存器（HSRAM1_CFGR）

名称	HSRAM1_CFGR							
offset	0x50							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				MCS		PDN_SLN	
位权限	U-0				R/W-01		R/W-11	

位号	助记符	功能描述
31:4	-	RFU：未实现，读为 0
3:2	MCS	速度配置，请保持复位值 01
1:0	PDN_SLN	功耗模式配置 11：normal 10：sleep 01/00：power down 注意：当 PDN_SLN=0x，禁止写 11

*此寄存器用于控制CANFD0模块的message RAM

4.6.23 HSRAM2 配置寄存器（HSRAM2_CFGR）

名称	HSRAM2_CFGR							
offset	0x54							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				MCS		PDN_SLN	
位权限	U-0				R/W-01		R/W-11	

位号	助记符	功能描述
31:4	-	RFU：未实现，读为 0
3:2	MCS	速度配置，请保持复位值 01
1:0	PDN_SLN	功耗模式配置 11：normal 10：sleep 01/00：power down 注意：当 PDN_SLN=0x，禁止写 11

*此寄存器用于控制CANFD1模块的message RAM

4.6.24 后门访问验证寄存器（FLS_BDVR）

名称	FLS_BDVR

offset	0x200							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	BDVR							
位权限	W-x							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	BDVR							
位权限	W-x							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BDVR							
位权限	W-x							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BDVR							
位权限	W-x							

位号	助记符	功能描述
31:0	BDVR	Backdoor Verify Register 写入 0x55AA1357 启动验证 验证失败的情况下，重新写入 0x55AA1357 无效

4.6.25 后门访问密钥输入寄存器 0（FLS_BDCR0）

名称	FLS_BDCR0							
offset	0x204							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	BDCR0							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	BDCR0							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BDCR0							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BDCR0							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	BDCR0	Backdoor code Register

4.6.26 后门访问密钥输入寄存器 1（FLS_BDCR1）

名称	FLS_BDCR1							
offset	0x208							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	BDCR1							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	BDCR1							

位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BDCR1							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BDCR1							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	BDCR1	Backdoor code Register

4.6.27 后门访问密钥状态寄存器（FLS_BDSR）

名称	FLS_BDSR							
offset	0x20C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						FLAG	DONE
位权限	U-0						R-0	R-0

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1	FLAG	后门秘钥验证结果，仅 DONE=1 时有效 0：验证成功 1：验证失败
0	DONE	后门秘钥验证完成标志 0：未完成 1：完成验证 软件操作 BDVR 后，需等待 DONE 标志置位

5 电源管理单元（PMU）

5.1 芯片工作电源

5.1.1 电源域划分

- VDD

芯片的主电源（VDD）的典型工作电压范围是 1.8~5.5V。

芯片的 VDD 实际工作电压范围将由 BOR 和 PDR 电路配置共同决定。

注意：在任何情况下不得同时关闭 BOR 和 PDR，这样在芯片掉电时可能由于没有产生正常的复位，而导致重新上电时芯片无法正常工作。

- VDDA

VDDA 是专用的模拟电路电源，主要给 ADC、基准电压等模拟模块供电。VDDA 的工作电压范围是 1.8~5.5V，所有模拟模块在这个电压范围内都可以保证正常工作。

- VCAPx

内部 LDO 输出，外部靠近管脚对地连接 1uF 稳压电容。

5.1.2 电源结构图

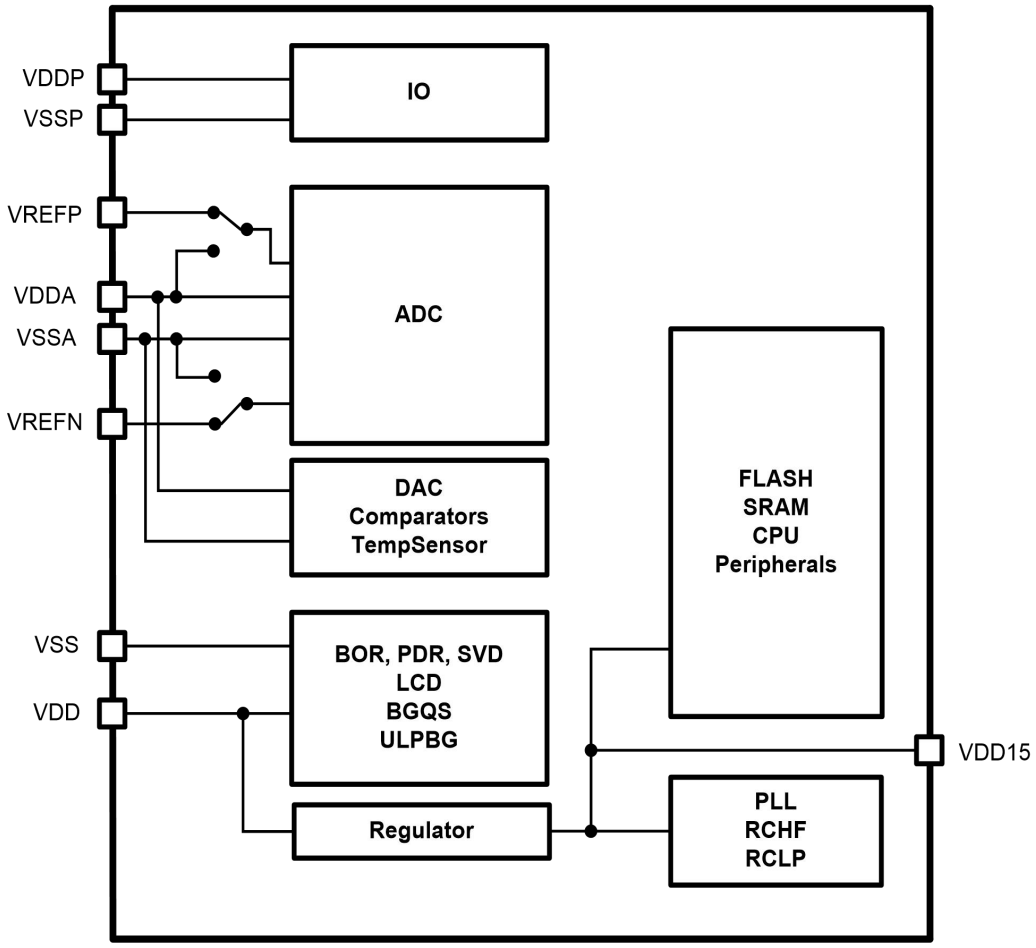


图 5-1 芯片电源结构图

5.1.3 ADC 基准电压

ADC 使用的电源地为 VDD(VDDA)和 VSS(VSSA)，其参考电压是 VREFP、VDDA 或内部产生的 VREFINT。为了提高 ADC 转换精度，降低电源噪声的影响，系统上建议对 VDDA 电源单独滤波，还可以在 PCB 上对 VDDA 和 VSSA 走线进行 shielding，以尽可能屏蔽系统噪声。

在某些低管脚数的封装中，可能没有独立的 VDDA 和 VSSA，在这种情况下，VDDA 和 VDD 在封装内部相连，VSSA 和 VSS 在封装内部相连，由于电源地噪声影响，ADC 的性能可能会有所下降。

外部输入的 VREFN、VREFP 与 PE9、PE10 这两个 GPIO 复用，当需要 ADC 使用外部基准时，必须将 GPIO 配置为模拟通道。

5.1.4 片内快速建立基准源（AVREF）

芯片内部集成一个高精度基准源 AVREF 电压 1.0V，为 LDO、Flash 提供基准电压。

5.2 功耗模式

5.2.1 概述

上电复位后，芯片默认运行在 ACTIVE 模式，此时 CPU 正常从 flash 取指运行，所有外设模块都可以正常工作。芯片支持多种低功耗模式，软件可以在适当的场景下选择合适的低功耗模式，以平衡不同的功耗、性能、唤醒时间和唤醒条件的要求。

芯片支持的功耗模式：

- ACTIVE 模式：正常运行
- SLEEP 模式：CPU 停止，Flash 停止，LDO 工作在超低功耗模式下，仅部分外设可以运行
- DEEPSLEEP 模式：CPU 停止，Flash 停止，关闭基准电压，LDO 工作在超低功耗模式下，仅部分外设可以运行

此外，ACTIVE 模式下的运行功耗也可以通过以下手段降低：

- 降低系统时钟频率
- 关闭不使用的外设的总线时钟和工作时钟

功耗模式	典型功耗	唤醒条件	芯片状态	典型唤醒时间[1]
ACTIVE	150uA/MHz		正常工作	-
SLEEP	6uA	电源检测中断 比较器中断 RTC 定时中断 IO 引脚中断 WKUPx 唤醒 32K 晶振停振 看门狗复位 NRST 引脚复位	CPU 休眠 关闭 RCHF、PLL、ADC 等 保持 AVREF 开启，关闭 LDO，RTC 走时	3us
DEEPSLEEP	1uA		CPU 休眠[2] 关闭 RCHF、PLL、ADC 等 关闭 AVREF，关闭 LDO15，RTC 走时	5us

注：[1] 典型唤醒时间指从唤醒信号到来，到 CPU 开始执行唤醒中断服务程序的时间间隔。

[2] CPU 自身进入休眠的步骤参见 ARMv6-M 架构参考手册

[3] CPU 试图进入低功耗模式时，如果 Flash 正在擦写，则芯片自动等待 Flash 擦写结束后再进入低功耗模式。

5.2.2 功耗模式与系统频率

不同功耗模式下可以接受的系统频率和可用的时钟源如下表所示。应用软件应严格遵守这个表格的规定，在低功耗模式下使用高主频可能导致系统无法正常运行。

功耗模式	CPU 频率	可用时钟源	Flash wait	外设工作时钟
ACTIVE	<=24MHz	All	0	All
	>24MHz, <=48MHz		1	
	>48Mhz, <=64Mhz		2	

5.2.3 Active 模式

芯片正常工作模式。芯片上电复位完成后进入 Active 模式运行，默认的 CPU 频率是 8MHz，最高可以运行到 48MHz。在 Active 模式下所有的数字和模拟外设都可以全速运行。

当主频高于 24MHz 时，必须配置 flash wait。

5.2.4 SLEEP 模式

通过进入 Sleep 模式，可以大幅降低芯片功耗，并处于等待事件唤醒的状态中。

进入 SLEEP 模式

软件按如下步骤进入 SLEEP 模式：

- 配置 PMOD 寄存器为 10
- 执行 WFI 或 WFE 指令

SLEEP 模式下的硬件行为

进入 SLEEP 模式后芯片关闭 CPU 时钟，Flash 进入 STOP 模式，硬件自动关闭 RCHF、PLL、XTHF、TRNG，SVD、比较器仍可以在 SLEEP 模式下工作。

数字外设模块可以使用 XTLF、RCLP 等低速时钟继续工作。

退出 SLEEP 模式

按照以下步骤退出 SLEEP 模式：

- 特定的中断事件发生
- 系统时钟被自动配置为 RCHF
- CPU 被唤醒，根据软件配置，唤醒后可以进入或者不进入中断服务程序

SLEEP 模式下不支持 CVS。

注意：芯片退出 SLEEP 时自动打开主 LDO。

5.2.5 DEEPSLEEP 模式

DEEPSLEEP 是芯片最低功耗模式，此模式下由于关闭了内部基准源（AVREF），因此休眠功耗比 SLEEP 进一步降低大约 3uA。

进入 DEEPSLEEP 模式

软件按如下步骤进入 DEEPSLEEP 模式：

- 置位 VREFOFF 寄存器
- 配置 PMOD 寄存器为 10
- 执行 WFI 或 WFE 指令

DEEPSLEEP 模式下的硬件行为

DEEPSLEEP 模式下，芯片自动关闭 CPU 时钟，关闭内部基准源，Flash 进入 STOP 模式，硬件自动关闭 RCHF、PLL、TRNG；SVD、比较器仍可以在 DEEPSLEEP 模式下工作。

数字外设模块可以使用 XTLF、RCLP 等低速时钟继续工作。

退出 DEEPSLEEP 模式

按照以下步骤退出 DEEPSLEEP 模式：

- 特定的中断事件发生
- 系统时钟被自动配置为 RCHF
- CPU 被唤醒，根据软件配置，唤醒后可以进入或者不进入中断服务程序

5.3 唤醒源

唤醒源	应用	可唤醒模式	
		Sleep	DeepSleep
停振检测	可屏蔽，32786Hz 晶振停振时唤醒芯片	√	√
SVD	可屏蔽，在电源电压跌落至阈值以下或升高至阈值以上时唤醒芯片	√	√
比较器	可屏蔽，用于外部事件唤醒	√	√
RTC	可屏蔽，根据需要的唤醒周期设置	√	√
IO 引脚中断	可屏蔽，用于外部事件唤醒	√	√
Debug	不可屏蔽，用于 debug 唤醒	√	√
LPUART	可屏蔽，接收数据唤醒	√	√
WKUPx 引脚	可屏蔽，用于外部输入唤醒	√	√
NRST	不可屏蔽，用于全局复位	√	√
LPTIM32	可屏蔽，用于定时唤醒	√	√

LPTIM16	可屏蔽，用于定时唤醒	√	√
BSTIM16	可屏蔽，用于定时唤醒	√	√
I2CSMB 从机	可屏蔽，用于从机接收唤醒	√	√
UARTx 唤醒	可屏蔽，检测到 UART0~3 下降沿唤醒信号时唤醒芯片	√	√
LIN 唤醒	可屏蔽，检测到 LIN 唤醒信号时唤醒芯片	√	√
FSCAN 唤醒	可屏蔽，检测到 CAN 总线电平发生隐性->显性变化时唤醒芯片	√	√
CANFD 唤醒	可屏蔽，检测到 CAN 总线电平发生隐性->显性变化时唤醒芯片	√	√

通过Cortex-M0的PRIMASK功能，可以实现以上中断事件唤醒芯片，但是CPU不执行中断处理程序。此时唤醒后CPU将继续从休眠前的指令之后开始运行。

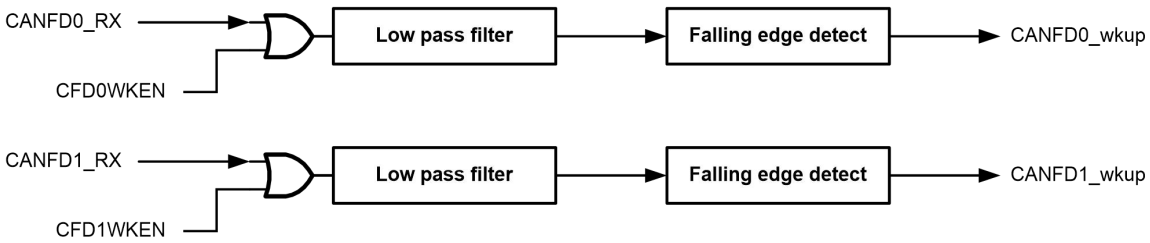
注：芯片从休眠模式唤醒后，软件可以通过查询PMU.WKPFLAG寄存器来快速识别当前的唤醒源，唤醒源的清除需要进入各个外设模块分别完成。

注：FSCAN、CANFD唤醒不需要时钟。FSCAN唤醒中断位于FSCAN模块内，CANFD唤醒中断位于PMU模块内。

5.3.1 CANFD 异步唤醒

CANFD异步唤醒功能与FSCAN唤醒功能相同，只是实现在了PMU模块内部。

唤醒事件同样是检测RX信号从隐性->显性的转换：



使能唤醒功能后，RX下降沿将置位CANFD唤醒中断，同时反映在唤醒源标志查询寄存器中。

不使能唤醒功能的情况下，RX下降沿不会置位CANFD唤醒中断。

5.4 休眠唤醒后的时钟控制

当芯片从Sleep/DeepSleep模式唤醒后，芯片以RCHF为时钟源。寄存器将保留休眠前RCHF的频率配置和trim值，因此唤醒后CPU运行频率将由休眠前软件配置寄存器决定（PMU.WKFSEL），默认为8MHz。最快情况下芯片唤醒后将以24MHz时钟启动。

5.5 寄存器

模块起始地址：0x4000_2000

offset 地址	名称	符号
0x00	低功耗控制寄存器 （Power Management Control Register）	PMU_CR
0x04	唤醒时间控制寄存器 （Wakeup Time Register）	PMU_WKTR
0x08	唤醒源标志查询寄存器 （Wakeup Source Flags Register）	PMU_WKFR
0x0C	PMU 中断使能寄存器 （PMU Interrupt Enable Register）	PMU_IER
0x10	PMU 中断标志寄存器 （PMU Interrupt and Status Register）	PMU_ISR
0x2C	-	-
0x30	-	-
0x34	-	-
0x38	-	-
0x3C	模拟 BUFFER 控制寄存器 （PMU Analog Buffer Control Register）	PMU_BUF_CR
0x50	PTAT 控制寄存器 （PMU PTAT Control Register）	PMU_PTAT_CR
0x54	-	-
0x58	-	-
0x80	CFD 唤醒使能寄存器	PMU_CFDWK_CR
0x84	CFD 唤醒标志寄存器	PMU_CFDWK_SR

5.5.1 低功耗控制寄存器（PMU_CR）

名称	PMU_CR							
offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFUI	-					-	
位权限	R/W-0	U-0					U-0	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				RFUI			
位权限	U-0				R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				WKFSEL		SLPDP	-
位权限	U-0				R/W-00		R/W-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				RFUI		PMOD	
位权限	U-0				R/W-00		R/W-00	

Bit	助记符	功能描述
31	**RFUI**	Dummy 寄存器
30:26	--	未实现，读为 0
25:24	--	未实现，读为 0

Bit	助记符	功能描述
23:12	--	未实现，读为 0
11:10	**WKFSEL**	Sleep/DeepSleep 唤醒后的系统频率 00：RCHF-8MHz 01：RCHF-16MHz 10：RCHF-24MHz 11：RFU
9	**SLPDP**	DeepSleep 控制寄存器 1：DeepSleep 模式使能，下关闭基准电压源 0：常规 Sleep 模式 在 Sleep 下，如果置位了 SLPDP 位即为 DeepSleep 模式； 该位仅在 Sleep 下有效
8	--	未实现，读为 0
7:4	--	未实现，读为 0
3:2	**RFUI**	Dummy 寄存器
1:0	**PMOD**	低功耗模式配置寄存器 00：Active mode 01：RFU 10：Sleep mode / DeepSleep mode 11：RFU

5.5.2 唤醒时间控制寄存器（PMU_WKTR）

名称	PMU_WKTR							
offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					STPCLR	T1A	
位权限	U-0					R/W-0	R/W-01	

Bit	助记符	功能描述
31:3	--	未实现，读为 0
2	**STPCLR**	Flash Stop 唤醒控制 0：Stop 信号等待时钟建立后同步清零 1：Stop 信号异步清零
1:0	**T1A**	可编程额外唤醒延迟 在 Sleep/DeepSleep 模式下，RCHF 时钟到来后，根据此寄存器配置等待额外延迟时间后，再读取 Flash 校验字 00：0us 01：2us 10：4us

Bit	助记符	功能描述
		11：8us

5.5.3 唤醒源标志查询寄存器（PMU_WKFR）

名称	PMU_WKFR							
offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	UART2WKF	UART1WKF	UART0WKF	RTCWKF	SVDWKF	LFDETWKF	UART3WKF	IOWKF
位权限	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	I2CSMB0WKF	I2CSMB1WKF	LPU2WKF	LPU1WKF	LPU0WKF	FSCANWKF	WKP9F	COMPWKF
位权限	R-0	R-0	R-0	R-0	R-0	R-0	R/W-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CFD1WKF	CFD0WKF	LPT32WKF	LPT16WKF	BST32WKF	BST16WKF	DBGWKF	WKPxF
位权限	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WKPxF							
位权限	R/W-00000000							

位	位名	功能描述
31	UART2WKF	UART2 中断唤醒标志，中断撤销时硬件自动清零 包含 LINWKF 和 NEWKF
30	UART1WKF	UART1 中断唤醒标志，中断撤销时硬件自动清零 包含 LINWKF 和 NEWKF
29	UART0WKF	UART0 中断唤醒标志，中断撤销时硬件自动清零 包含 LINWKF 和 NEWKF
28	RTCWKF	RTC 中断唤醒标志，中断撤销时硬件自动清零
27	SVDWKF	SVD 中断唤醒标志，中断撤销时硬件自动清零
26	LFDETWKF	32768Hz 晶体停振中断唤醒标志，中断撤销时硬件自动清零
25	UART3WKF	UART3 中断唤醒标志，中断撤销时硬件自动清零 包含 LINWKF 和 NEWKF
24	IOWKF	IO 中断唤醒标志，中断撤销时硬件自动清零
23	I2CSMB1WKF	I2C_SMB1 中断唤醒标志，中断撤销时硬件自动清零
22	I2CSMB0WKF	I2C_SMB0 中断唤醒标志，中断撤销时硬件自动清零
21	LPU2WKF	LPUART2 中断唤醒标志，中断撤销时硬件自动清零
20	LPU1WKF	LPUART1 中断唤醒标志，中断撤销时硬件自动清零
19	LPU0WKF	LPUART0 中断唤醒标志，中断撤销时硬件自动清零
18	FSCANWKF	FSCAN 中断唤醒标志，中断撤销时硬件自动清零
17	WKP9F	WKUP9 Pin 唤醒标志，软件写 1 清零
16	COMPWKF	比较器 1~4 中断唤醒标志，中断撤销时硬件自动清零
15	CFD1WKF	CANFD1 中断唤醒标志，中断撤销时硬件自动清零
14	CFD0WKF	CANFD0 中断唤醒标志，中断撤销时硬件自动清零
13	LPT32WKF	LPTIM32 中断唤醒标志，中断撤销时硬件自动清零

位	位名	功能描述
12	LPT16WKF	LPTIM16 中断唤醒标志，中断撤销时硬件自动清零
11	BST32WKF	BSTIM32 中断唤醒标志，中断撤销时硬件自动清零
10	BST16WKF	BSTIM16 中断唤醒标志，中断撤销时硬件自动清零
9	DBGWKF	CPU Debugger 唤醒标志，软件写 1 清零
8:0	WKPxF	WKUP0~8 Pin 唤醒标志，软件写 1 清零

5.5.4 PMU 中断使能寄存器（PMU_IER）

名称	PMU_IER							
offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						SLPEIE	-
位权限	U-0						R/W-0	U-0

Bit	助记符	功能描述
31:2	--	未实现，读为 0
1	**SLPEIE**	SLEEP 错误中断使能 1：使能 SLEEP 错误中断 0：禁止 SLEEP 错误中断
0	--	未实现，读为 0

5.5.5 PMU 中断标志寄存器（PMU_ISR）

名称	PMU_ISR							
offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						SLPEIF	-
位权限	U-0						R/W-0	U-0

Bit	助记符	功能描述
31:2	--	未实现，读为 0
1	SLPEIF	SLEEP 错误中断标志，硬件置位，软件写 1 清零 1：在 PMOD=2'h2 后，CPU 执行 WFI/WFE 指令前置位了 SLEEPDEEP 寄存器时置位 0：在 PMOD=2'h2 后，CPU 正确进入 SLEEP
0	--	未实现，读为 0

5.5.6 模拟 BUFFER 控制寄存器（PMU_BUFCR）

名称	PMU_BUFCR							
offset	0x3C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		AVREFBUF_OUTEN	AVREFBUF_EN	VPTATBUFFER_OUTEN	VPTATBUFFER_EN	-	-
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0

位号	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5	AVREFBUF_OUTEN	AVREF 输出缓冲输出使能，ADC 采样 AVREF 时建议使能
4	AVREFBUF_EN	AVREF 输出缓冲使能，ADC 采样 AVREF 时建议使能
3	VPTATBUFFER_OUTEN	Vptat Buffer 模块开关通道输出使能信号，高电平使能有效。(PTAT Buffer Output Enable)
2	VPTATBUFFER_EN	Vptat Buffer 模块使能信号，高电平使能有效。(PTAT Buffer Enable)
1	-	RFU：未实现，读为 0
0	-	RFU：未实现，读为 0

5.5.7 PTAT 控制寄存器（PMU_PTAT_CR）

名称	PMU_PTAT_CR							
地址	0x50							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							PTAT_EN
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	PTAT_EN	温度传感器使能 0：关闭温度传感器 1：使能温度传感器 *注：芯片测量温度前，需要置位 PTAT_EN 和 VPTATBUF_EN、VPTATNUF_OEN，等待输出建立时间后，启动 ADC 采样并计算温度。*

5.5.8 CANFD 唤醒使能寄存器

名称	PMU_CFDCK_CR							
地址	0x80							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CFD1WKIE	CFD0WKIE
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						CFD1WKEN	CFD0WKEN
位权限	U-0						R/W-0	R/W-0

Bit	助记符	功能描述
31:10	--	RFU：未实现，读为 0
9	CFD1WKIE	CANFD1 唤醒中断使能，1 有效
8	CFD0WKIE	CANFD0 唤醒中断使能，1 有效
7:2	--	RFU：未实现，读为 0
1	CFD1WKEN	CANFD1 异步唤醒功能使能，1 有效
0	CFD0WKEN	CANFD0 异步唤醒功能使能，1 有效

5.5.9 CANFD 唤醒中断寄存器

名称	PMU_CFDCK_SR							
地址	0x84							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						CFD1WKIF	CFD0WKIF
位权限	U-0						R/W-0	R/W-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	CFD1WKIF	CANFD1 异步唤醒中断标志，硬件置位，软件写 1 清零
0	CFD0WKIF	CANFD0 异步唤醒中断标志，硬件置位，软件写 1 清零

6 处理器（CPU）

6.1 概述

FM33FG0A 使用的 CPU 内核为 Cortex-M0，符合 ARMv6-M 架构和编程模型；更多信息请参考 ARM 官网 www.arm.com

其基本特性如下：

- 用户/特权模式
- VTOR（中断向量表重定向）
- NVIC 支持 32 个外部中断
- 数据监视点：1
- 硬件断点：4
- 单周期 32-bit 硬件乘法器
- SWD 调试接口

6.1.1 处理器配置

Feature	Options	FM33FG0AConfig
Interrupts	1~32	32
Data endianness	little/big	little
SysTick Timer	Present or absent	Present
watchpoints	0,1,2	1
breakpoints	0,1,2,3,4	4
halting debug support	Present or absent	Present
multiplier	Fast or Small	Fast
Single-Cycle IO	Present or absent	Absent
wake-up interrupt controller(WIC)	Present or absent	Present
Vector Table Offset Register	Present or absent	Present
Unprivileged/Priviledged support	Present or absent	Present
JTAGnSW	JTAG or SWD for DAP	SWD
Memory Protection Unit	Present or absent	Absent

表 6-1FM33FG0A CPU 配置简表

6.2 寄存器

主要内核寄存器列表

名字	描述
R0-R12	通用寄存器
MSP (R13)	堆栈指针；Handler 模式下使用 MSP（Main Stack Pointer），Thread 模式下通过 CONTROL 寄存器选择 MSP 或 PSP（Process Stack Pointer）使用
PSP (R13)	
LR (R14)	Link 寄存器，保存子函数/函数调用/异常处理的返回信息
PC (R15)	程序指针
PSR	包含应用程序状态（APSR）、中断程序状态（IPSR）和程序执行状态（EPSR）
PRIMASK	PRIMASK 用于屏蔽指定优先级及以下的所有中断响应
CONTROL	设置 Thread 模式下使用的堆栈指针

表 6-2 Cortex-M0 内核寄存器简表

寄存器详细定义参见 ARMv6-M 架构参考手册。

6.3 异常和中断

内核的异常和中断管理通过 NVIC 完成。NVIC 的可编程管理寄存器位于 PPB 总线的 SCS 空间内，NVIC 具有如下特性：

- 支持 32 个外部中断，5 个内部异常
- 1 个 NMI 中断
- 支持中断嵌套
- 向量化的异常入口
- 中断屏蔽

处理器内核接受一个异常请求后，首先会将内核寄存器 R0~R3、R12、R14、PC、xPSR 压入堆栈。链接寄存器 LR（R14）被更新为异常返回时使用的特殊值（EXC_RETURN），然后根据异常向量表定位异常处理程序开始执行。注意在异常处理中没有被自动压栈的寄存器，必须通过软件来保存和恢复。

6.3.1 中断向量表

参见**错误!未找到引用源。**章节

6.3.2 中断优先级

处理器支持 3 个固定的最高优先级及 4 个可编程优先级。当两个相同优先级的异常同时发生，则异常编号较小的异常将被首先执行。

6.3.3 错误处理

处理器只支持一种硬件错误处理方式：HardFault 异常。HardFault 优先级-1，只有 NMI 能对其抢占。

HardFault 的触发原因包含以下几种情况：

错误类型	错误条件
存储器相关	总线错误。由于在总线传输中使用了非法地址而产生的总线错误。
	试图在 XN 区域内执行程序
程序错误	执行未定义的指令
	试图切换至 ARM 状态
	试图进行非对齐的存储器访问 在更高优先级异常处理中执行 SVC 指令 执行异常返回时 EXC_RETURN 的值非法 当调试未使能时试图执行 BKPT 指令

FM33FG0A 的 HardFault 触发原因可以通过寄存器查询，以帮助软件开发者定位错误原因。

6.3.4 锁定（Lockup）

当处理器在进行 HardFault 处理的过程中发生了另一个 HardFault，或者 NMI 处理期间发生了 HardFault，则处理器将进入锁定状态（停止执行），并输出 LOCKUP 信号，此时芯片将自动复位处理器内核，而不是等待看门狗溢出。

6.4 MPU

MPU 符合 ARMv6-M Protected Memory System Architecture (PMSAv6)。MPU 支持以下特性：

- 支持 8 个可编程存储区域（Region）
- 支持背景区域特性
- 可交叠的区域，支持 0~7 的区域优先级（0 为最低优先级，7 为最高优先级）
- 访问权限控制
- 输出存储器属性
- 错误权限访问将被阻止并触发 HardFault

对于没有嵌入式 OS 的简单系统，MPU 可以被编程为静态配置，常用功能举例：

- 将部分 RAM 区域设为只读，避免重要数据被意外破坏
- 将堆栈底部空间设置为不可访问，以检测堆栈溢出
- 将 SRAM 区域设置为 XN（不可执行），避免代码注入攻击

对于具有嵌入式 OS 的系统，OS 可以在每次上下文切换（Context Switch）时动态配置 MPU，使得每个应用任务都有不同的 MPU 配置，实现更为复杂的权限管理：

- 定义 SRAM 访问权限，确保应用任务只能访问自己的堆栈空间，避免堆栈泄漏而破坏其他堆栈
- 定义存储器访问权限，使得应用任务只能访问有限的外设
- 限制应用任务只能访问自己的数据池（literal pool）或程序代码

6.4.1 MPU 寄存器

MPU 相关寄存器位于系统控制空间（SCS），包含以下寄存器，注意 MPU 寄存器只支持字访问：

地址	寄存器	功能
0xE000ED90	MPU 类型寄存器（TYPE）	只读，提供 MPU 相关查询信息
0xE000ED94	MPU 控制寄存器（CTRL）	MPU 使能/禁止和背景区域控制
0xE000ED98	MPU 区域编号寄存器（RNR）	选择待配置的 MPU 区域
0xE000ED9C	MPU 基地址寄存器（RBAR）	定义 MPU 区域的基地址
0xE000EDA0	MPU 区域属性和大小寄存器（RASR）	定义 MPU 区域的属性和大小

6.4.1.1 MPU 类型寄存器

Name：MPU_TYPE			
Address：0xE000ED90			
Field	Description	Reset	Access
31:24	-		
23:16	IREGION	0x00	R
15:8	DREGION	0x08	R
7:1	-		
0	I	0	R

6.4.1.2 MPU 控制寄存器

Name：MPU_CTRL			
Address：0xE000ED94			
Field	Description	Reset	Access
31:3	-		
2	PRIVDEFENA 特权等级的默认存储器映射使能，当其为 1 且 MPU 使能时，特权访问会将默认的存储器映射用作背景区域；若此位为 0，则背景区域被禁止，且对不属于任何使能区域的访问会触发 HardFault	0	R/W
1	HFNMIENA 1 – MPU 在 HardFault 和 NMI 处理过程中也是使能的 0 – HardFault 和 NMI 处理中 MPU 不使能	0	R/W
0	ENABLE 1 – 使能 MPU 0 –禁止 MPU	0	R/W

6.4.1.3 MPU 区域编号寄存器

Name：MPU_RNR			
Address：0xE000ED98			
Field	Description	Reset	Access
31:8	-		
7:0	REGION 在设置每个区域之前，写入这个寄存器可以选择要编程的区域；由于处理器只支持 8 个 Region，应避免写入 0-7 以外的值	-	R/W

6.4.1.4 MPU 基地址寄存器

Name：MPU_RBAR			
Address：0xE000ED9C			
Field	Description	Reset	Access
31:8	ADDR 区域的基地址	-	R/W
7:5	-		
4	VALID 1 – Bit[3:0]写入的 REGION 编号会在基地址编程时起作用，同时覆盖 MPU_RNR 中的最低 4bit 0 –MPU_RNR 寄存器中的值会在基地址编程时起作用	-	R/W
3:0	REGION 写入时如果 VALID=1 则覆盖 MPU_RNR[3:0]；读出时返回 MPU_RNR[3:0]	-	R/W

6.4.1.5 MPU 区域属性和大小寄存器

Name：MPU_RASR			
Address：0xE000EDA0			
Field	Description	Reset	Access
31:29	-	-	
28	XN 禁止取指 1 –禁止 CPU 从该区域取指，取指会触发 HardFault 0 –允许从该区域取指	0	R/W
27	-		
26:24	AP	000	R/W

	访问控制		
23:22	-		
21:19	TEX 类型展开域，只支持 000，其他值保留	000	R/W
18	S 可共用	-	R/W
17	C 可缓存	-	R/W
16	B 可缓冲	-	R/W
15:8	SRD 子区域禁止；每个区域被 MPU 平均分割为 8 个子区域，8bit SRD 用于单独使能或禁止每个子区域。 1 –禁止对应的子区域 0 –使能对应的子区域 Bit8 控制最低区域内地址的子区域，Bit15 控制区域内最高地址的子区域	0x00	R/W
7:6	-		
5:1	SIZE 区域大小设置，允许的取值范围是 7-31，对应区域大小为 $2^{(SIZE+1)}$bytes，不支持小于 7 的设置，因为最小区域大小是 256bytes（32bytes*8）	-	R/W
0	ENABLE 区域使能 0 –禁止此区域 1 –使能此区域（在使能 MPU 的前提下）	0	R/W

TEX（类型展开）、S（Shareable）、C（Cachable）、B（Bufferable）表示存储器区域属性，每次数据和指令访问时这些属性都会被输出到总线上，供 write-buffer 或 cache 等总线器件使用。

下面的表格定义了区域属性和访问控制的编码规则：

TEX[a]	C	B	Memory type	Description, or Normal region cacheability	Shareable?
000	0	0	Strongly-ordered	Strongly ordered	Shareable
000	0	1	Device	Shared device	Shareable
000	1	0	Normal	Outer and inner write-through, no write allocate	S bit[b]
000	1	1	Normal	Outer and inner write-back, no write allocate	S bit[b]

a. All other combinations of TEX, C, and B are reserved.
b. Shareable if the S bit is set to 1, Non-shareable if the S bit is set to 0.

AP[2:0]	Privileged access	Unprivileged access	Notes
000	No access	No access	Any access generates a permission fault
001	Read and write	No access	Privileged access only
010	Read and write	Read only	Any unprivileged write generates a permission fault
011	Read and write	Read and write	Full access
100	UNPREDICTABLE	UNPREDICTABLE	Reserved
101	Read-only	No access	Privileged read-only
110	Read-only	Read-only	Privileged or unprivileged read-only
111	Read-only	Read-only	Privileged or unprivileged read-only

6.5 调试特性

处理器支持以下调试特性

- 程序的暂停、恢复及单步执行
- 访问内核寄存器和特殊寄存器
- 硬件断点（4 个）
- 软件断点（不限数量的 BKPT 指令）
- 数据监视点（1 个）
- 动态非侵入式存储器访问（无需停止处理器）
- SWD 接口

Cortex-M0 的调试特性是基于 ARM CoreSight 调试架构的，详情请参考《CoreSight Technology System Design Guide》和《ARM Debug Interface Architecture Specification ADIv5.0 to ADIv5.2》

6.5.1 调试功能引脚

FM33FG0A 使用 SWD 调试接口，用户模式下最少仅需 4 线（NRST, GND, SWIO, SWCLK）即可实现调试功能。2 线调试引脚可以复用为 GPIO，其功能由软件选择配置。

NRST 引脚用于复位芯片，通过 NRST 与 SWD 的配合，可以使芯片复位后 Halt 在第一条指令处。

调试功能引脚的复用说明参见 I/O 控制章节。

6.5.2 调试状态下的看门狗控制

看门狗在调试模式下可以保持使能或关闭。软件或 Debugger 可以通过 MCUDBGCR 寄存器配置看门狗打开或关闭。

6.5.3 DEBUG 的复位

内核的 DEBUG 部分仅受上下电复位影响，其他系统复位源如看门狗、引脚复位、软件复位等，都不会复位 DAP 电路。这样可以在芯片上电后通过引脚复位使 CPU 内核处于复位状态，但是调试器仍可以正常与 DAP 建立通信并设置断点，在复位放开后可以使 CPU 立即进入调试模式。

建议调试器在系统复位时连接内核(在复位向量处设置断点)。

7 中断向量表

CPU 中断向量表如下，

Position	Priority	Priority type	Acronym	Description	Address
0	-	-	MSP 初值	主栈指针初始化地址	0x0000_0000
1	-3	fixed	Reset	复位向量	0x0000_0004
2	-2	fixed	NMI	WKUPx 中断 PMU 中断 存储器非法访问中断 Flash ECC 检错中断 RAM ECC 检错中断	0x0000_0008
3	-1	fixed	HardFault	HardFault 中断向量	0x0000_000C
4-10	-	-	-	Reserved	0x0000_0010~0x0000_002B
11	3	settable	SVC	SVCall 系统服务请求	0x0000_002C
12-13	-	-	-	Reserved	0x0000_0030~0x0000_0037
14	5	settable	PendSV	可挂起系统服务请求	0x0000_0038
15	6	settable	Systick	内部定时器中断向量	0x0000_003C
16	7	settable	WDT	窗口看门狗或独立看门狗中断	0x0000_0040
17	8	settable	SVD	电源监测报警中断	0x0000_0044
18	9	settable	RTC	实时时钟中断	0x0000_0048
19	10	settable	CANFD0	CAN-FD 中断 （包含唤醒中断）	0x0000_004C
20	11	settable	CANFD1		0x0000_0050
21	12	settable	ADC	ADC 转换完成中断	0x0000_0054
22	13	settable	FSCAN	FSCAN 中断	0x0000_0058
23	14	settable	UART0	UART 中断 (UART4/5 即 USART)	0x0000_005C
24	15	settable	UART1		0x0000_0060
25	16	settable	UART2		0x0000_0064
26	17	settable	UART3		0x0000_0068
27	18	settable	UART4/UART5		0x0000_006C
28	19	settable	I2CM	I2C Master0/1 中断	0x0000_0070
29	20	settable	SPI0	SPI 中断	0x0000_0074
30	21	settable	SPI1		0x0000_0078
31	22	settable	SPI2		0x0000_007C
32	23	settable	SPI3		0x0000_0080
33	24	settable	I2C_SMBx	I2C SMBUS0/1 中断	0x0000_0084
34	25	settable	SENTx	SENT 中断	0x0000_0088
35	26	settable	AES	AES 中断	0x0000_008C
36	27	settable	LPTIM/BSTIM	LPTIM/BSTIM 中断 (包含 16 及 32 位)	0x0000_0090
37	28	settable	DMA	DMA 中断	0x0000_0094
38	29	settable	WKUPx	WKUP 引脚中断	0x0000_0098
39	30	settable	TAU	TAU 中断	0x0000_009C

Position	Priority	Priority type	Acronym	Description	Address
			PGL	PGL 中断	
40	31	settable	GPTIMx	GPTIM0/1/2 中断	0x0000_00A0
41	32	settable	COMPx	COMPx 中断	0x0000_00A4
42	33	settable	CLMx	CLM0/1 时钟监控中断	0x0000_00A8
43	34	settable	NVMIF	NVMIF 中断	0x0000_00AC
44	35	settable	ATIM	高级定时器中断	0x0000_00B0
45	36	settable	LPUART	LPUART 中断	0x0000_00B4
46	37	settable	EXTI DAC	外部引脚中断 DAC 中断	0x0000_00B8
47	38	settable	ECCC RAMP FDET	Flash ECC 纠错中断 RAM ECC 纠错中断 XTLF 或 XTHF 停振检测中断	0x0000_00BC

表 7-1 中断向量表

其中WKUPx中断可以接到NMI或者38#入口。通过GPIO模块的GPIO_PINWKEN.WKISEL寄存器来选择中断入口地址。当配置为38#入口时，可以通过PRIMASK将WKUPx中断屏蔽，唤醒后CPU不进入中断服务程序，而是继续从休眠指令处向下执行。

8 复位管理单元（RMU）

8.1 概述

复位电路特点：

- 支持多个复位源，如上下电复位、看门狗复位、软件复位、引脚复位等
- 上下电复位（BOR）监控主电源供电
- BOR 上电复位典型释放电压 1.6V
- BOR 下电复位产生电压软件可配置，支持 8 个档位，可关闭。
- 低功耗下电复位电路（PDR）
- 上下电复位信号经过去抖动和延时，抗干扰能力强

进入复位状态时，所有寄存器都恢复到初始值（除 RTC 内部寄存器）；退出复位状态时，MCU 使用内部 RC 振荡器（RCHF，默认频率 8MHz）作为系统时钟。

8.2 模块框图

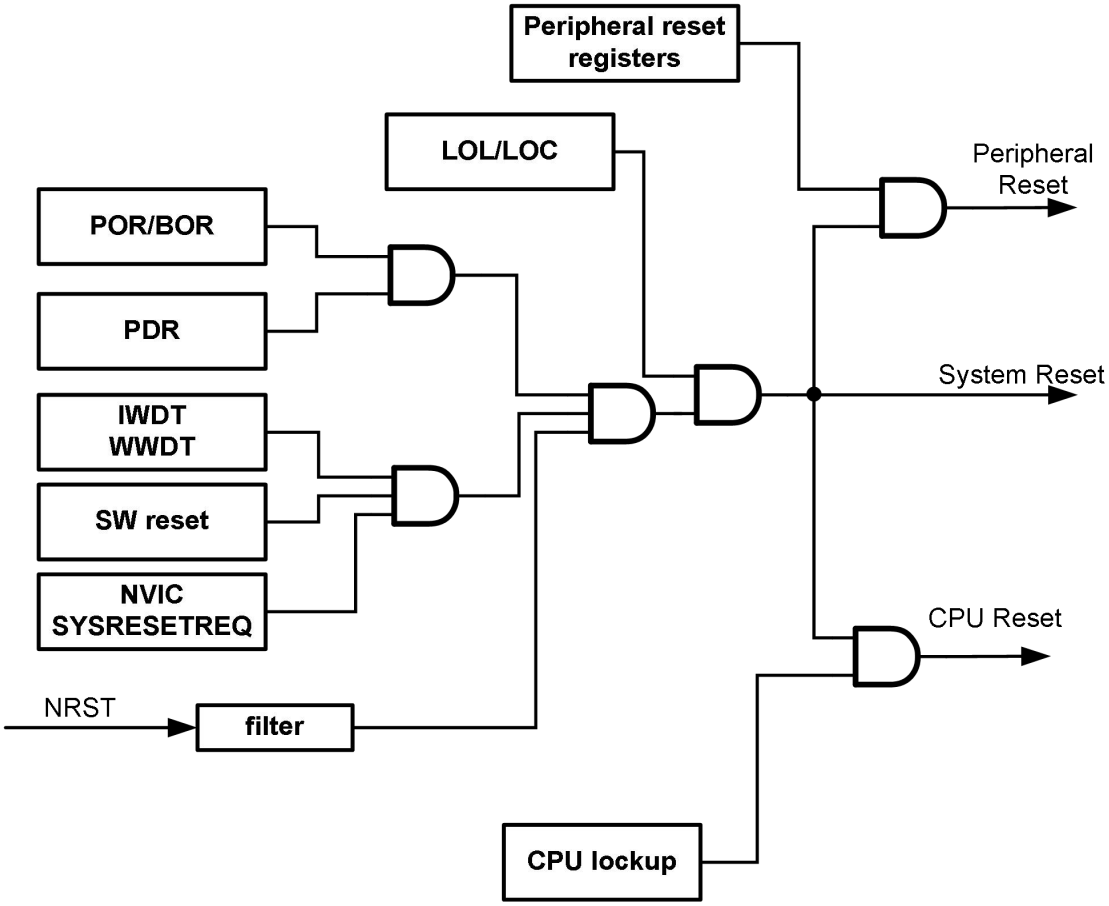


图 8-1 芯片复位源框图

8.3 上下电复位

上下电复位电路监控 VDD 电源，由 BOR、POR 和低功耗 PDR 组成。

- POR：提供上电复位
- BOR：高精度可编程下电复位，8 个档位
- PDR：超低功耗下电复位

为防止电源抖动，保证上电复位电路的抗干扰能力，对上电复位信号进行滤波和延时处理。

BOR下电复位阈值软件可配置，共8档：1.8V、2.0V、2.2V、2.4V、2.7V、3V、3.6V、4V，注意电源回升时有200mV左右的回滞窗口。

芯片上电复位放开后，BOR下电复位默认是关闭的，软件可以开启。

注意：BOR是一个下电复位电路，软件使能BOR之前，必须确保芯片电源已经上升到准备使用的BOR档位阈值之上，否则使能BOR将导致直接复位。比如，上电复位释放后，如果VDD=3V，而软件希

望设置BOR阈值为3.6V，则此时软件不应使能BOR，否则使能后将立即导致复位。建议软件在使能BOR之前，先通过ADC或者SVD检测芯片电源是否达到预期电压。

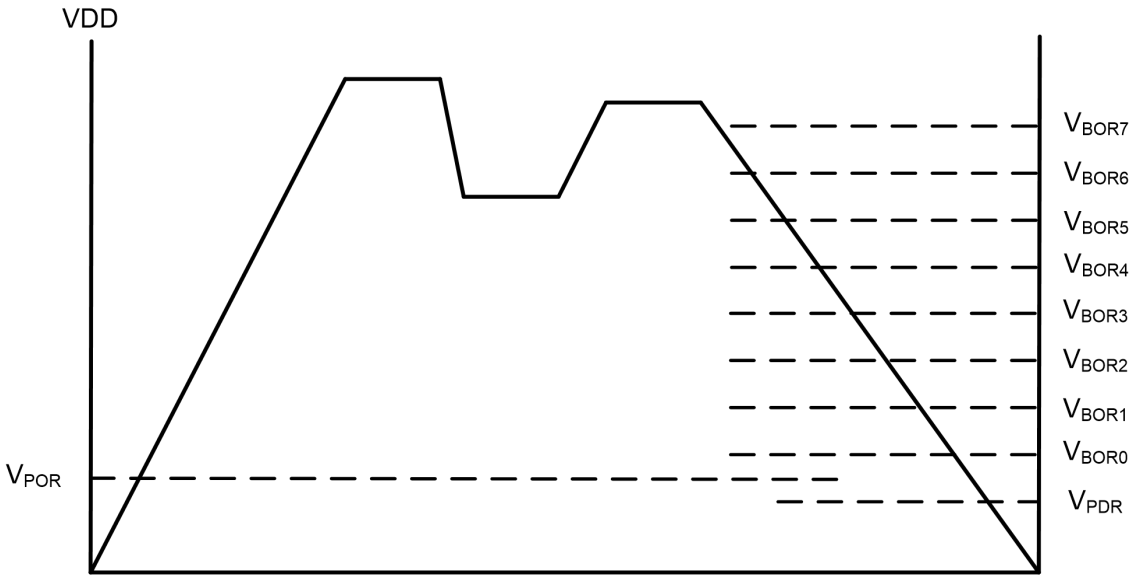


图 8-2 上下电复位示意图

假设上电后软件使能BOR并配置档位3，VDD变化导致芯片系统复位的波形示意图如下

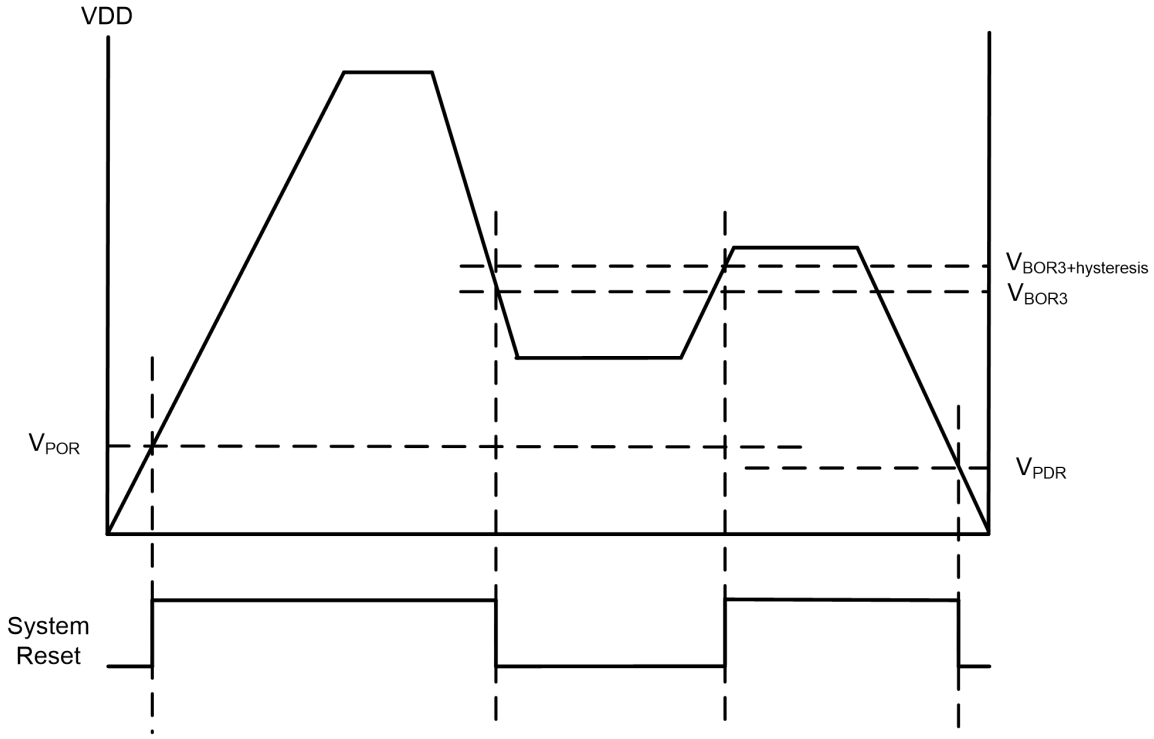


图 8-3 POR, BOR, PDR 复位波形示意图

上电复位滤波时间决定了电源上升的最短时间要求，为了保证安全的上电复位，要求上电速度不能快于2us/V，即电源从0上升到5V，用时不能短于10us。

注意，如果软件关闭BOR，则当电源电压在V_{BOR0}和V_{PDR}之间时，芯片仍在运行。由于芯片全速运行的最低电压是1.65V，为了保证CPU运行可靠，建议使用SVD监控电源，当电源电压较低时，程序应主动进入休眠。

8.4 软件复位

软复位由 CPU 写寄存器发起，操作方式为向 SOFTRST 寄存器写 0x5C5C_AABB。

另外也可以通过操作 Cortex 内核 SCB 模块的 AIRCR 寄存器，置位 SYSRESETREQ bit 的方式来实现全局复位。

8.5 NRST 引脚复位

NRST 是一个双向 PAD，除了外部输入复位信号外，当内部复位源有效时，也会拉低 NRST 引脚，通知系统芯片处于复位状态。此设计符合功能安全要求，当 MCU 处于复位状态时，通过拉低 NRST 引脚通知系统进入安全状态。

内部复位信号源（上下电复位输出、软件复位、LOCKUP 复位、看门狗复位）组合后经过脉冲展宽电路，确保复位有效脉冲宽度至少大于 2 个 RCLP 时钟周期（min 60us），此展宽后脉冲用于驱动 NRST PAD 内的 NMOS 管，拉低外部 NRST 引脚，同时展宽后脉冲用于芯片全局复位。NRST 拉低后也会从输入反馈回来，两者相与后产生全局复位，功能等价。

8.6 LOC 复位

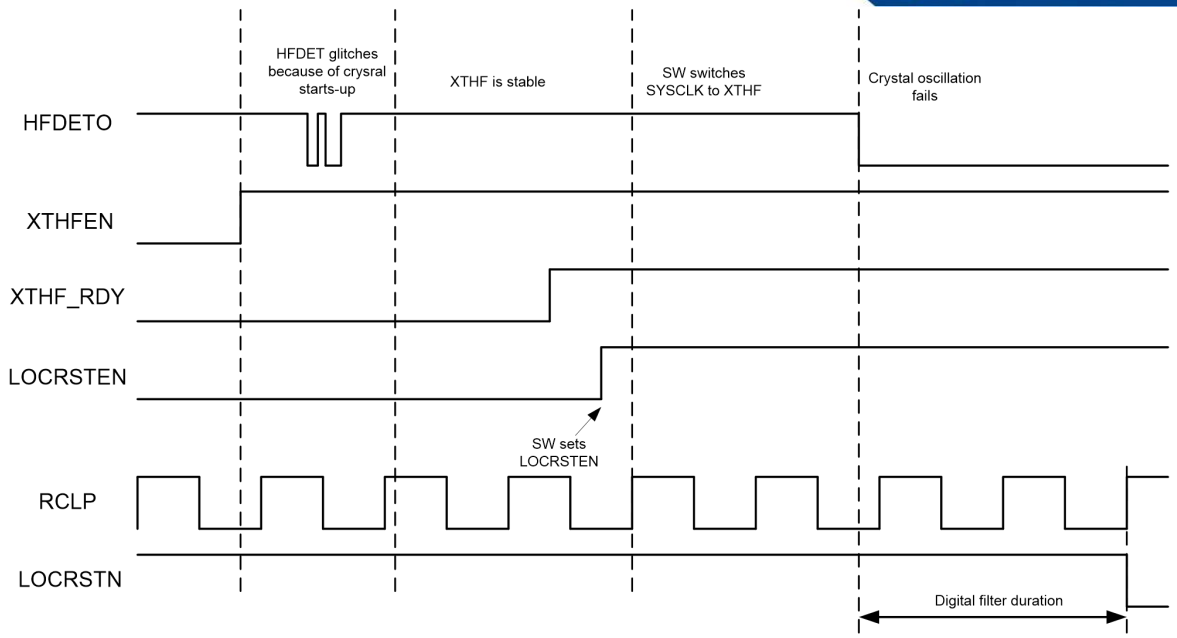
LOC（Loss Of Clock）复位在以下条件满足时复位全芯片：

- SYSCLK 源头选择为 XTHF
- 使能 LOC 复位功能（LOCRSTEN=1）
- HFDET 检测到 XTHF 停振

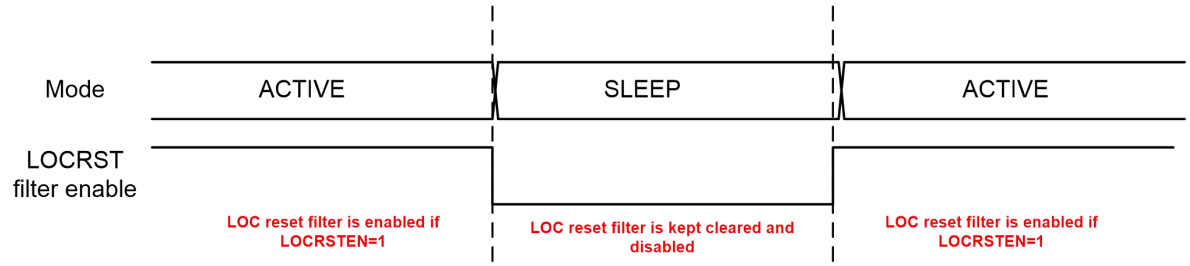
或者

- SYSCLK 源头选择为 XTLF
- 使能 LOC 复位功能（LOCRSTEN=1）
- LFDET 检测到 XTLF 停振

LOC 复位功能波形图如下：



LOCRSTEN 仅在 ACTIVE 模式下有效

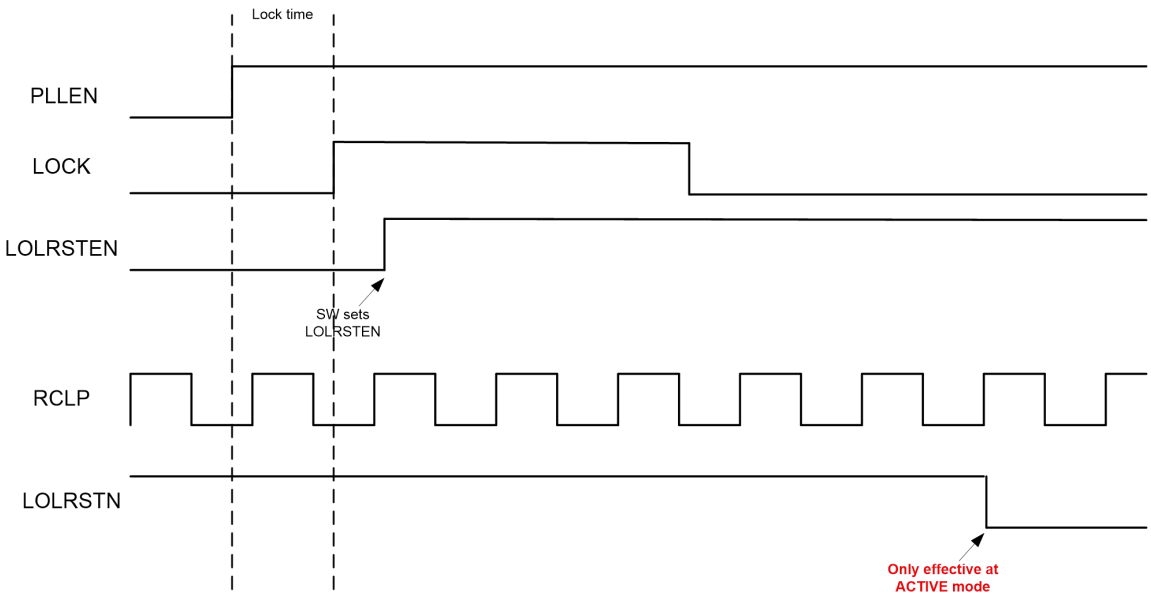


8.7 LOL 复位

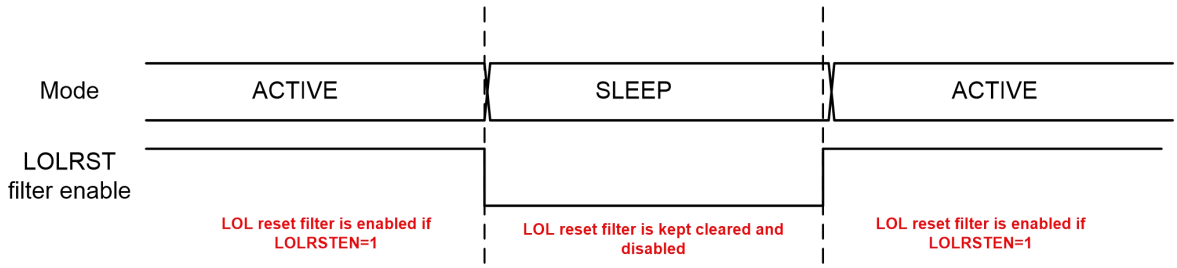
Loss of Lock 复位在以下条件满足时复位全芯片：

- SYSCLK 源头选择为 PLL
- 使能 LOL 复位功能（LOLRSTEN=1）
- PLL 失锁，LOCKED 标志清零

LOL 复位波形如下



注意：LOL 复位功能仅在 Active 模式下有效，进入休眠后自动屏蔽 LOLRSTEN。



8.8 寄存器

模块起始地址：0x40002800

offset 地址	名称	符号
0x00	PDR 配置寄存器	RMU_PDRCR
0x04	BOR 配置寄存器	RMU_BORCR
0x08	复位配置寄存器	RMU_LKPCR
0x0C	软件复位寄存器	RMU_SOFTRST
0x10	复位标志寄存器	RMU_RSTFR
0x14	外设复位控制寄存器	RMU_PRSTEN
0x18	AHB 外设复位寄存器	RMU_AHBRSTCR
0x1C	APB 外设复位寄存器 1	RMU_APBRSTCR1
0x20	APB 外设复位寄存器 2	RMU_APBRSTCR2

8.8.1 PDR 配置寄存器（RMU_PDRCR）

名称	RMU_PDRCR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					CFG		EN
位权限	U-0					R/W-00		R/W-1

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2:1	**CFG**	下电复位电压配置 00—1.4V 01—1.45V 10—1.5V 11—1.55V
0	**EN**	下电复位使能 0：关闭下电复位 1：使能下电复位

8.8.2 BOR 配置寄存器（RMU_BORCR）

名称	RMU_BORCR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			CFG			-	ENB
位权限	U-0			RW-000			U-0	R/W-1

Bit	助记符	功能描述
31:5	--	RFU：未实现，读为 0
4:2	**CFG**	BOR 复位电压配置 000：falling 1.8V reset, rising 2.0V release 001：falling 2.0V reset, rising 2.2V release 010：falling 2.2V reset, rising 2.4V release 011：falling 2.4V reset, rising 2.6V release 100：falling 2.7V reset, rising 2.9V release 101：falling 3.0V reset, rising 3.2V release 110：falling 3.6V reset, rising 3.8V release 111：falling 4.0V reset, rising 4.2V release
1	--	RFU：未实现，读为 0
0	**ENB**	BOR 下电复位使能，上电后默认关闭 0：使能 BOR 下电复位 1：关闭 BOR 下电复位

8.8.3 复位配置寄存器（RMU_LKPCR）

名称	RMU_LKPCR							
地址	0x40000200							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							

名称	RMU_LKPCR							
地址	0x40000200							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOLRSTEN	LOCRSTEN	-				LKUPRST_EN	-
位权限	R/W-0	R/W-0	U-0				R/W-0	U-0

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7	**LOLRSTEN**	PLL 失锁复位使能（仅 Active 模式有效） 1：当检测到 PLL 失锁事件，复位芯片 0：PLL 失锁不会复位芯片，仅产生报警
6	**LOCRSTEN**	晶振停振复位使能 1：当检测到晶振停振，复位芯片 0：晶振停振不会复位芯片，仅产生报警 注，LOCRSTEN=1 时产生 LOC 复位的条件： 1） 系统时钟为 XTHF，XTHF 停振 2） 系统时钟为 XTLF，XTLF 停振 3） 系统时钟为 PLL，PLL 参考时钟为 XTHF，XTHF 停振
5:2	--	RFU：未实现，读为 0
1	**LKUPRST_EN**	LOCKUP 复位使能 1：使能 SC000 LOCKUP 复位 0：屏蔽 SC000 LOCKUP 复位
0	--	RFU：未实现，读为 0

8.8.4 软件复位寄存器（RMU_SOFTRST）

名称	RMU_SOFTRST							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SOFTRST							
位权限	W							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SOFTRST							
位权限	W							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SOFTRST							
位权限	W							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SOFTRST							
位权限	W							

Bit	助记符	功能描述
31:0	**SOFTRST**	软件写 0x5C5C_AABB 触发全局复位

8.8.5 复位标志寄存器（RMU_RSTFR）

名称	RMU_RSTFR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		CLM1_TOT_FLAG	CLM1_UDF_FLAG	CLM1_OVF_FLAG	CLM0_TOT_FLAG	CLM0_UDF_FLAG	CLM0_OVF_FLAG
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	RAMP_FLAG	MAP_FLAG	MDF_FLAG	NRSTN_FLAG	PRCN_FLAG	PORN_FLAG	PDRN_FLAG
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOL_FLAG	LOC_FLAG	SOFTN_FLAG	IWDTN_FLAG	-	WWDTN_FLAG	LKUPN_FLAG	NVICN_FLAG
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:22	--	RFU：未实现，读为 0
21	**CLM1_TOT_FLAG**	CLM1 超时复位标志，高有效，软件写 1 清零
20	**CLM1_UDF_FLAG**	CLM1 下溢复位标志，高有效，软件写 1 清零
19	**CLM1_OVF_FLAG**	CLM1 上溢复位标志，高有效，软件写 1 清零
18	**CLM0_TOT_FLAG**	CLM0 超时复位标志，高有效，软件写 1 清零
17	**CLM0_UDF_FLAG**	CLM0 下溢复位标志，高有效，软件写 1 清零
16	**CLM0_OVF_FLAG**	CLM0 上溢复位标志，高有效，软件写 1 清零
15:14	-	RFU：未实现，读为 0
13	**MAP_FLAG**	非法地址访问复位标志，高有效，软件写 1 清零
12	**MDF_FLAG**	模式诊断超时复位标志，高有效，软件写 1 清零
11	**NRSTN_FLAG**	NRST 引脚复位标志，高有效，软件写 1 清零
10	**PRCN_FLAG**	PRC 复位标志，高有效，软件写 1 清零
9	**PORN_FLAG**	上电复位标志，高有效，软件写 1 清零
8	**PDRN_FLAG**	下电复位标志，高有效，软件写 1 清零
7	**LOL_FLAG**	PLL 失锁复位标志，高有效，软件写 1 清零
6	**LOC_FLAG**	晶振停振复位标志，高有效，软件写 1 清零
5	**SOFTN_FLAG**	软件复位标志，高有效，软件写 1 清零
4	**IWDTN_FLAG**	IWDT 复位标志，高有效，软件写 1 清零
3	--	RFU：未实现，读为 0
2	**WWDTN_FLAG**	WWDT 复位标志，高有效，软件写 1 清零
1	**LKUPN_FLAG**	LOOKUP 复位标志，高有效，软件写 1 清零
0	**NVICN_FLAG**	NVIC 复位标志，高有效，软件写 1 清零

8.8.6 外设复位使能寄存器（RMU_PRSTEN）

名称	RMU_PRSTEN							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PERHRSTEN[31:24]							
位权限	W							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PERHRSTEN[23:16]							
位权限	W							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PERHRSTEN[15:8]							
位权限	W							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PERHRSTEN[7:0]							
位权限	W							

Bit	助记符	功能描述
31:0	PERHRSTEN	外设模块复位使能，32bit 虚寄存器，只写 软件对此地址写 0x1357_9BDF，使能外设复位功能，此后可以通过外设模块复位寄存器复位各个模块 软件对此地址写任意其他数据，将关闭外设复位功能

8.8.7 AHB 外设复位寄存器（RMU_AHBRSTCR）

名称	RMU_AHBRSTCR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							DMARST
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	DMARST	DMA 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

8.8.8 APB 外设复位寄存器 1（RMU_APBRSTCR1）

名称	RMU_APBRSTCR1							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	UART5RST	UART4RST	UART3RST	UART2RST	UART1RST	UART0RST	UARTIRRST	LPUART2RST
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SPI3RST	SPI2RST	SPI1RST	SPI0RST	GPT2RST	GPT1RST	GPT0RST	FSCANRST
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LPUART1RST	LPUART0RST	I2CSMB1RST	I2CSMB0RST	-	-	I2C1RST	I2C0RST
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LPT32RST	BT32RST	PGLRST	ATIMRST	TAU1RST	TAU0RST	LPT16RST	BT16RST
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31	UART5RST	UART5 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
30	UART4RST	UART4 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
29	UART3RST	UART3 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
28	UART2RST	UART2 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
27	UART1RST	UART1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
26	UART0RST	UART0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
25	UARTIRRST	UART 红外调制模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
24	LPUART2RST	LPUART2 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
23	SPI3RST	SPI3 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
22	SPI2RST	SPI2 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位

Bit	助记符	功能描述
		1：复位
21	SPI1RST	SPI1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
20	SPI0RST	SPI0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
19	GPT2RST	GPT2 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
18	GPT1RST	GPT1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
17	GPT0RST	GPT0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
16	FSCANRST	FSCAN 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
15	LPUART1RST	LPUART1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
14	LPUART0RST	LPUART0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
13	I2CSMB1RST	I2CSMB1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
12	I2CSMB0RST	I2CSMB0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
11	--	RFU：未实现，读为 0
10	--	RFU：未实现，读为 0
9	I2C1RST	I2C1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
8	I2C0RST	I2C0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
7	LPT32RST	LPTIM32 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
6	BT32RST	BT32 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
5	PGLRST	PGL 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
4	ATIMRST	ATIM 模块复位，软件写 1 复位，写 0 撤销复位

Bit	助记符	功能描述
		0：不复位 1：复位
3	TAU1RST	TAU1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
2	TAU0RST	TAU0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
1	LPT16RST	LPTIM16 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
0	BT16RST	BT16 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

8.8.9 APB 外设复位寄存器 2（RMU_APBRSTCR2）

名称	APBRST2							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CFD1RSTR	CFD1RSTP	CFD1RSTC	CFD0RSTR	CFD0RSTP	CFD0RSTC	DACRST	ADCCRST
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ADCRST	CLM1RST	CLM0RST	SENT1RST	SENT0RST	AESRST	CRCRST	RNGRST
位权限	R/W-0			U-0		R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				HDIVRST	-	SVDRST	COMPRST
位权限	U-0				R/W-0	U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31	CFD1RSTR	CANFD1 RAMCLK domain reset 0：不复位 1：复位
30	CFD1RSTP	CANFD1 APBCLKdomain reset 0：不复位 1：复位
29	CFD1RSTC	CANFD1 CLKC domain reset 0：不复位 1：复位
28	CFD0RSTR	CANFD0 RAMCLK domain reset 0：不复位 1：复位
27	CFD0RSTP	CANFD0 APBCLK domain reset

Bit	助记符	功能描述
		0：不复位 1：复位
26	CFD0RSTC	CANFD0 CLKC domain reset 0：不复位 1：复位
25	DACRST	DAC 控制器复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
24	ADCCRST	ADC 控制器复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
23	ADCRST	ADC 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
22	CLM1RST	CLM1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
21	CLM0RST	CLM0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
20	SENT1RST	SENT1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
19	SENT0RST	SENT0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
18	AESRST	AES 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
17	CRCRST	CRC 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
16	RNGRST	RNG 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
15:4	--	RFU：未实现，读为 0
3	HDIVRST	HDIV 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
2	--	RFU：未实现，读为 0
1	SVDRST	SVD 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
0	COMPRST	COMP 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

9 独立看门狗（IWDT）

9.1 概述

独立看门狗用于监视系统运行，如果 CPU 运行异常，无法定时清狗，则看门狗在溢出后产生全局复位信号，重启系统，以避免系统锁死。独立看门狗的使能由 OPTBYTES 控制，可配置为上电后自启动或者软件启动。

任何情况下，IWDT 一旦启动就不可关闭，直到芯片复位。

IWDT 使用独立时钟工作（RCLP），RCLP 上电后自动开启，且不能关闭。

为了便于调试，在以下情况下 IWDT 会停止运行：

- 芯片 CP 时，当 TESTN 有效，IWDT 停止运行
- 当芯片处于调试模式时，软件可以通过配置 MCUDBGCR 寄存器在调试过程中暂停 IWDT
- 当 OPTBYTES 中 IWDTSLP 有效时，软件可以在休眠模式下暂停 IWDT 计数

IWDT 核心是一个 12bit 向上计数器，复位后从 0 开始递增，计数到 0xFFF 后触发 IWDT 复位。IWDT 复位是一个全局复位，效果等同于上下电复位。

IWDT 使用 RCLP 工作，确保不会停止工作。IWDT 带有除 128 预分频器，计数器长度为 12bit。

IWDT 支持可编程窗口功能，软件只能在允许的窗口内清狗，窗口外清狗将触发 IWDT 复位。

9.2 结构框图

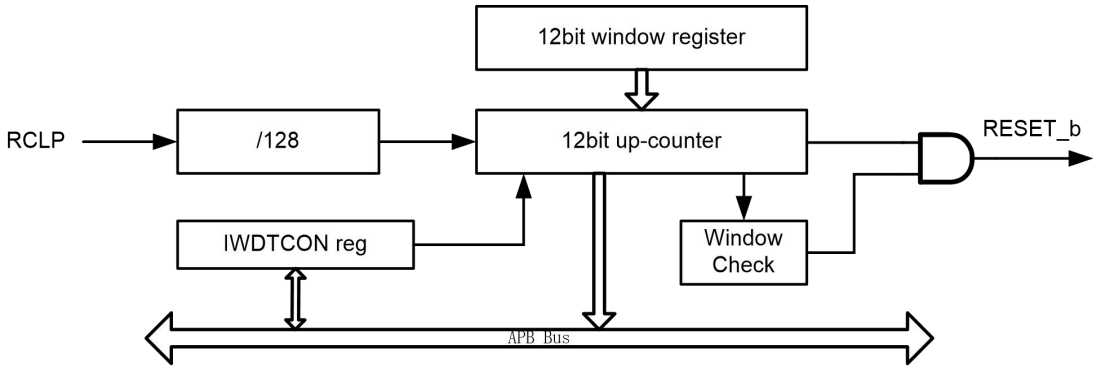


图 9-1 IWDT 结构框图

9.3 IWDT 功能描述

CPU 正常运行时，看门狗应使用较短的溢出周期，而在 SLEEP/DEEPSLEEP 等低功耗模式下，为了使芯片尽可能长时间的停留在低功耗模式下，则看门狗应使用较长的溢出周期。

为了兼容两者的不同应用需求，软件可以实时修改 IWDT 的溢出周期配置。为避免不当操作引发不可预计的后果，软件在更新溢出周期配置时应遵循以下操作步骤：

- 确保看门狗正在运行
- 首先进行一次清狗操作
- 随后改写 IWDTCFG 寄存器，选择合适的溢出周期
- 读 IWDTCFG，确保写入正确
- 溢出周期更新完毕，CPU 正常运行

IWDT 使用 RCLP 工作，内部预分频 128，分频后的计数器溢出长度可配置为 1~4096（共 8 个可用档位），溢出时间长度计算公式如下：

tWWDT = T_{RCLP} * 128 * IWDTOVP

RCLP 频率	溢出长度配置	溢出时间（ms）
32KHz	32	125
	64	250
	128	500
	256	1000
	512	2000
	1024	4000
	2048	8000
	4096	16000

表 9-1 IWDT 溢出周期表

9.4 IWDT 窗口功能

IWDT支持可编程清狗窗口功能。IWDT_WIN寄存器用于定义允许的清狗窗口，只有当计数器计数值大于等于IWDT_WIN的值时，清狗操作才是合法的，在窗口之外清狗将会直接出发IWDT复位。

芯片复位后IWDT_WIN为全0，即默认允许软件在任何位置清狗。

软件可以在IWDT运行过程中实时修改IWDT_WIN寄存器。软件清狗时必须读取并确认当前计数值是否在允许清狗的范围内。

当IWDT计数值进入清狗窗口时，IWDT会触发一个中断标志寄存器，通知软件当前计数值已经进入清狗窗口。

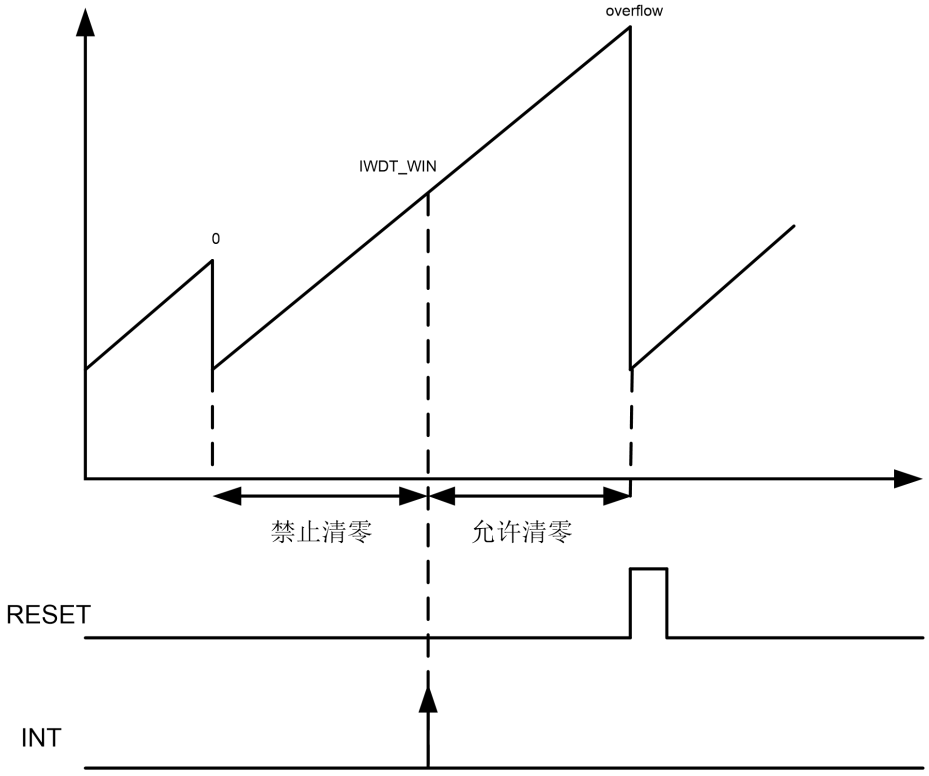


图 9-2 IWDT 窗口示意图

9.5 IWDT 冻结

用户可以通过OPTBYTES配置是否允许IWDT在休眠模式下冻结计数。

当OPTBYTES中的IWDTSLP有效，并且软件将IWDT_FREEZE寄存器置位时，当芯片进入Sleep/DeepSleep模式，IWDT计数值自动冻结（注意不是关闭IWDT，只是计数值保持当前值不再递增）。

9.6 寄存器

模块起始地址：0x4001_1400

offset 地址	名称	符号
0x00	IWDT 清除寄存器 （IWDT Service Register）	IWDT_SERV
0x04	IWDT 配置寄存器 （IWDT Config Register）	IWDT_CR
0x08	IWDT 计数值寄存器 （IWDT Counter Register）	IWDT_CNT
0x0C	IWDT 窗口寄存器 （IWDT Window Register）	IWDT_WIN
0x10	IWDT 中断使能寄存器 （IWDT Interrupt Enable Register）	IWDT_IER
0x14	IWDT 中断标志寄存器 （IWDT Interrupt Status Register）	IWDT_ISR

9.6.1 IWDT 清除寄存器（IWDT_SERV）

名称	IWDT_SERV							
Offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SERV[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SERV[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SERV[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SERV[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:0	**SERV**	上电后 IWDT 状态受 LDT1 区域 IWDTEN 控制。 IWDT 启动后，软件向此地址写入 0x1234_5A5A 时清狗 (IWDT Service Register,write only)

9.6.2 IWDT 配置寄存器（IWDT_CR）

名称	IWDT_CR							
Offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				FREEZE	-		
位权限	U-0				R/W-0	U-0		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					OVP		
位权限	U-0					R/W-001		

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11	FREEZE	IWDT 休眠冻结，仅在 OPTBYTES 中 IWDTSLP 配置有效时起作用(Freeze in Sleep Enable) 1：Sleep/DeepSleep 模式下冻结 IWDT 计数 0：Sleep/DeepSleep 模式下保持 IWDT 运行
10:3	-	RFU：未实现，读为 0
2:0	OVP	配置 IWDT 看门狗溢出时间 (Overflow Period) 000：125ms 001：250ms 010：500ms 011：1s 100：2s 101：4s 110：8s 111：16s

9.6.3 IWDT 计数值寄存器（IWDT_CNT）

名称	IWDT_CNT							
Offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				CNT[11:8]			
位权限	U-0				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:0	CNT	IWDT 当前计数值，软件只读 (IWDT Counter Value,read only) 由于计数器工作时钟与 APB 总线为异步关系，软件读取计数值时应连续读取 2 次以上，为相同值时才认为是稳定结果

9.6.4 IWDT 窗口寄存器（IWDT_WIN）

名称	IWDT_WIN							
Offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				WIN[11:8]			
位权限	U-0				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WIN[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:0	WIN	IWDT 窗口寄存器 (IWDT Window)

9.6.5 IWDT 中断使能寄存器（IWDT_IER）

名称	**IWDT_IER**							
Offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							IE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	IE	IWDT 中断使能 (IWDT Interrupt Enable) 0：中断使能禁止 1：中断使能打开

9.6.6 IWDT 中断标志寄存器（IWDT_ISR）

名称	**IWDT_ISR**
Offset	0x14

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							WINF
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	WINF	IWDT 进入窗口中断标志，写 1 清零 (Window Flag,write 1 to clear) 0：无中断产生 1：计数值进入清狗窗口

10 窗口看门狗（WWDT）

10.1 功能描述

带窗口的看门狗是一个与 CPU 同步运行的看门狗，目的是实时监控 CPU 运行状态，在 CPU 运行异常的情况下复位全芯片，避免不可预计的后果。

WWDT 在芯片上电后默认关闭，软件启动 WWDT 后，不能再关闭，直到下一次复位。低功耗休眠模式下 WWDT 停止运行。

为了保证同步性和实时性，WWDT 使用 CPU 时钟工作，内部有一个预分频电路，以产生同步计数使能信号。

在以下情况时 WWDT 产生 CPU 复位：

- 计数器溢出
- 对 WWDT 清零寄存器写 0xAC 以外的值（可用于触发 CPU 软复位）
- 在窗口关闭期内对 WWDT 清零寄存器写 0xAC

当计数器达到溢出时间的 75%时，会触发一个预警中断。

10.2 结构框图

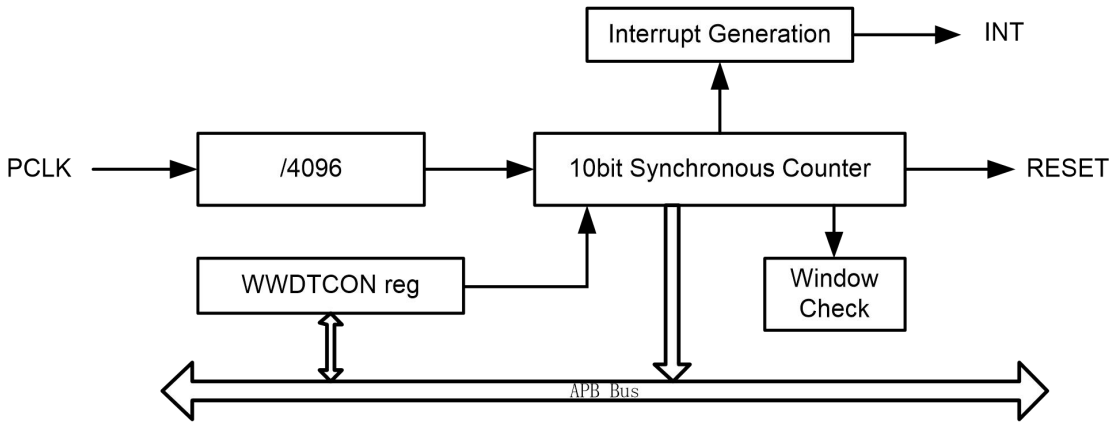


图 10-1WWDT 结构框图

10.3 WWDT 工作方式

WWDT 在芯片复位后默认关闭，软件需对 WWDTCON 寄存器写入 0x5A 来启动 WWDT。WWDT

启动后，如果软件在 open window 内对 WWDTCON 写 0xAC，将清零计数器。WWDT 一旦使能后不能关闭，直到下一次复位，WWDT 复位发生后将会关闭 WWDT。

WWDT 使用 PCLK 工作，内部预分频 4096，分频后的计数器溢出长度可配置为 1~1024（共 8 个可用档位），溢出时间长度计算公式如下：

$tWWDT = T_{APBCLK} * 4096 * N_{CFG}$

下表为计算示例：

APBCLK 频率	溢出长度配置	溢出时间（ms）
48MHz	1	0.085
	4	0.341
	16	1.365
	64	5.461
	128	10.922
	256	21.845
	512	43.69
	1024	87.38
32MHz	1	0.128
	4	0.512
	16	2.048
	64	8.192
	128	16.384
	256	32.768
	512	64.536
	1024	131.072
16MHz	1	0.256
	4	1.024
	16	4.096
	64	16.384
	128	32.768
	256	65.536
	512	129.072
	1024	262.144
8MHz	1	0.512
	4	2.048
	16	8.192
	64	32.768
	128	65.536
	256	131.072
	512	258.144
	1024	524.288

表 10-1 WWDT 溢出周期表

WWDT 只允许在 open window 内进行清除，否则将直接触发复位。使能窗口为计数器的后半周期，

软件在清零看门狗之前应注意查询计数值。

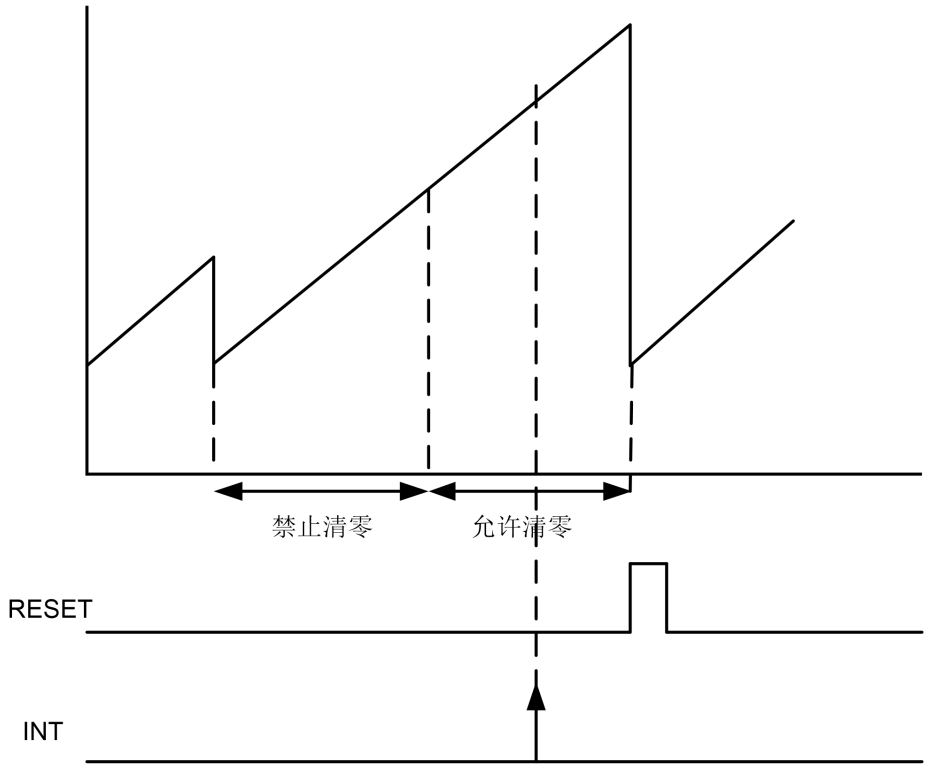


图 10-2WWDT 窗口示意图

10.4 寄存器

模块起始地址：0x4001_1800

offset 地址	名称	符号
0x00	WWDT 控制寄存器 （WWDT Control Register）	WWDT_CR
0x04	WWDT 配置寄存器 （WWDT Config Register）	WWDT_CFGR
0x08	WWDT 计数值寄存器 （WWDT Counter Register）	WWDT_CNT
0x0C	WWDT 中断使能寄存器 （WWDT Interrupt Enable Register）	WWDT_IER
0x10	WWDT 中断标志寄存器 （WWDT Interrupt Status Register）	WWDT_ISR
0x14	WWDT 预分频寄存器 （WWDT Prescaler Register）	WWDT_PSC

10.4.1 WWDT 控制寄存器（WWDT_CR）

名称	WWDT_CR							
Offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CON							
位权限	W-0000 0000							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	CON	当 CPU 向此地址写入 0x5A 时启动 WWDT 定时器 (WWDT Control,write only) 在启动 WWDT 后，当 CPU 向此地址写入 0xAC 时清零计数器

10.4.2 WWDT 配置寄存器（WWDT_CFGR）

名称	WWDT_CFGR							
Offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					CFG		
位权限	U-0					R/W-011		

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2:0	CFG	配置 WWDT 看门狗溢出时间，复位值 011，由于上电后系统时钟默认为 8Mhz，所以默认溢出周期大约 32ms (WWDT Config) 000：T_{PCLK} * 4096 * 1 001：T_{PCLK} * 4096 * 4 010：T_{PCLK} * 4096 * 16 011：T_{PCLK} * 4096 * 64 100：T_{PCLK} * 4096 * 128 101：T_{PCLK} * 4096 * 256 110：T_{PCLK} * 4096 * 512 111：T_{PCLK} * 4096 * 1024

10.4.3 WWDT 计数寄存器（WWDT_CNT）

名称	WWDT_CNT							
Offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CNT[9:8]	
位权限	U-0						R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9:0	CNT	WWDT 计数寄存器值，软件可通过查询此寄存器了解 WWDT 计时进度 (WWDT Counter value,read only)

10.4.4 WWDT 中断使能寄存器（WWDT_IER）

名称	WWDT_IER
Offset	0x0C

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							IE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	IE	WWDT 中断使能 (WWDT Interrupt Enable) 0：中断使能禁止 1：中断使能打开

10.4.5 WWDT 中断标志寄存器（WWDT_ISR）

名称	WWDT_ISR							
Offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							NOVF
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	NOVF	WWDT 75%计时中断标志，写 1 清零 (Near Overflow Flag,write 1 to clear) 0：无中断产生 1：中断标志置位 如果 IE=1，则此寄存器置位将触发中断

10.4.6 WWDT 预分频寄存器（WWDT_PSC）

名称	WWDT_PSC
Offset	0x14

名称	WWDT_PSC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				DIV_CNT[11:8]			
位权限	U-0				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIV_CNT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:0	DIV_CNT	WWDT 的 4096 预分频计数器当前计数值，只读 (WWDT prescaler Divider Counte,read only)

11 时钟管理单元（CMU）

11.1 概述

芯片内包含32.768KHz低频晶体振荡电路(XTLF)、4~16MHz高频晶体振荡器、最高32MHz高频RC振荡器(RCHF)、32KHz低功耗内部环振（RCLP）、高频晶体振荡器和一个锁相环（PLL）。芯片内部的时钟产生模块整合这些时钟源，产生各个模块工作所需要的时钟。

特点：

- 系统主时钟可选多个时钟源
- 时钟可在系统运行中实时切换
- 低频晶体振荡器配备停振检测电路
- 部分外设模块独立工作时钟（与 CPU 和总线时钟解耦）
- CPU 和总线最高频率 64MHz

11.2 时钟架构

11.2.1 时钟树

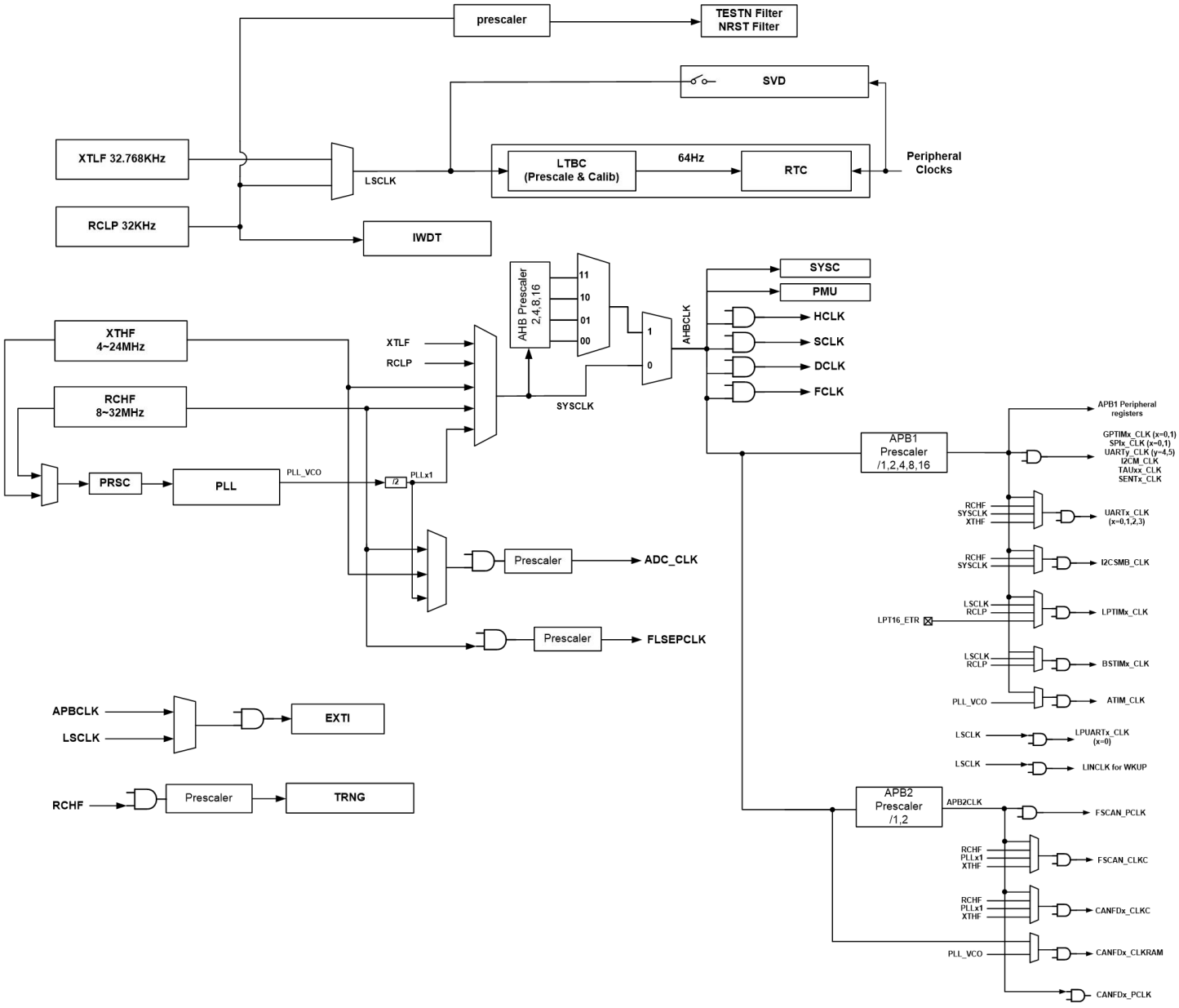


图 11-1 芯片时钟框图

系统主时钟(SYSCLK)可由XTLF、RCHF、XTHF、PLL、RCLP及它们的分频时钟产生。上电默认使用8MHzRCHF的不分频时钟作为系统主时钟，各外设模块的时钟可以分别独立控制。芯片工作时可以只打开需要工作的模块时钟，其他模块的时钟可关闭，以节省功耗。APB总线时钟APBCLK可

以是AHBCLK的分频或同频时钟，用于驱动APB总线上的低速外设。

11.2.2 SYSCLK 切换说明

SYSCLK是系统主时钟，从SYSCLK可以得到AHBCLK和APBCLK等总线时钟以及CPU运行所需的时钟。

SYSCLK选择任何一个时钟源时，硬件都要检查对应时钟源是否开启，如果时钟源没有被使能（或发生停振），则软件切换操作无效，SYSCLKSEL寄存器不会被改写，时钟切换也不会发生，同时SYSCKE_IF中断标志寄存器置位，可以产生中断事件。

目标时钟	切换条件
RCHF	RCHF 使能
XTHF	XTHF 使能且未停振
PLL	PLL 使能，并且： 1， 如果 PLL 参考时钟是 XTHF，则 XTHF 必须使能且未停振 2， 如果 PLL 参考时钟是 RCHF，则 RCHF 必须使能
XTLF	XTLF 未停振
RCLP	-

11.2.3 时钟安全性

当使用XTLF、XTHF为系统时钟时，或者使用PLL为系统时钟并且PLL参考时钟为XTHF时，必须考虑时钟安全性。

在以上条件下，如果发生时钟停振，停振检测输出将强制启动RCHF，并将SYSCLK切换逻辑复位成RCHF通路，同时复位AHBPRES寄存器、复位RCHF档位选择寄存器，确保系统AHB时钟为固定的RCHF 8M不分频。

- SYSCLK为XTLF，当XTLF停振时自动将SYSCLK切换到RCHF
- SYSCLK为XTHF，当XTHF停振时自动将SYSCLK切换到RCHF
- SYSCLK为PLL并且PLL参考时钟选择为XTHF，当XTHF停振时自动将SYSCLK切换到RCHF

以上切换发生的同时，停振检测电路会产生停振中断，通知软件处理异常。

11.2.4 主要时钟说明

时钟	源头	说明
LSCLK	XTLF, RCLP	32KHz 低频系统时钟，晶体停振时可以切换到 RCLP 主要用于 RTC、IWDT、SVD、LCD

时钟	源头	说明
SYSCLK	RCHF, PLL, RCLP, XTLF, XTHF	32K~48MHz，经过分频后得到 AHBCLK
HCLK(AHBCLK)	SYSCLK	AHB 总线时钟，用于驱动 CPU、RAM、Flash 和高速外设
SCLK	SYSCLK	CPU 内核系统时钟
DCLK	SYSCLK	CPU 内核 Debug 时钟（当仿真器连接时这个时钟必须活动）
FCLK	SYSCLK	Free-Running 时钟，提供给 CPU 内核 WIC 模块，以及 APB 桥
APBCLK	AHBCLK	APB 总线时钟，用于驱动低速外设

11.2.5 外设模块的总线时钟和工作时钟

部分外设模块的总线时钟和工作时钟互相独立。

其中总线时钟用于AHB或APB总线访问，在软件访问外设的功能寄存器时，必须先通过外设总线时钟控制寄存器来使能对应的总线时钟。

而外设的工作时钟为外设实际工作使用的时钟，这个时钟可能不同于APBCLK或AHBCLK，外设模块工作前，需要通过外设工作时钟寄存器来选择所需的时钟源，并打开时钟门控。

而对于工作时钟和总线时钟统一的外设模块，则仅需使能总线时钟就可以正常工作了。

模块	总线时钟	工作时钟
独立工作时钟外设		
UARTx (x=0,1,2,3)	APBCLK	APBCLK
		RCHF
		SYSCLK
LPUARTx (x=0,1,2)	APBCLK	LSCLK
		RCHF
I2C_SMBUSx (x=0,1)	APBCLK	APBCLK
		RCHF
		SYSCLK
ATIM	APBCLK	APBCLK
		PLLx2_CLK
LPTIM16 LPTIM32	APBCLK	APBCLK
		LSCLK
		RCLP
		LPT32_ETR, LPT16_ETR
BSTIM16 BSTIM32	APBCLK	APBCLK
		LSCLK
		RCLP

模块	总线时钟	工作时钟
ADC	APBCLK	
		XTHF
		RCHF
		PLL
NVMIF (Flash erase/program)	AHBCLK	RCHF
EXTI (PADCFG)	AHBCLK	AHBCLK
		LSCLK
TRNG	APBCLK	RCHF
IWDT	APBCLK	LSCLK
RTC	APBCLK	LSCLK
FSCAN	APBCLK[1]	XTHF RCHF PLL
CAN-FDx	APBCLK[1]	XTHF PLL PLL_VCO
非独立工作时钟外设		
PMU	AHBCLK	
DMA	AHBCLK	
GPTIMx (x=0,1,2)	APBCLK	
UARTy (y=4,5)	APBCLK	
SPIx (x=0,1,2,3)	APBCLK	
AES	APBCLK	
CRC	APBCLK	
WWDT	APBCLK	
COMPx (x=0,1,2,3)	APBCLK	
DAC	APBCLK	
SENTx	APBCLK	

注[1]：FSCAN/CAN-FDx其总线时钟域为APBCLK，具体对应时钟信号名为APB2CLK；其它外设，总线或工作时钟域同为APBCLK，具体对应具体时钟信号名为APB1CLK；下文同。

11.2.6 休眠模式下的外设时钟

Sleep/DeepSleep模式下，SYSCLK被关闭，因此在休眠模式下AHBCLK和APBCLK都不工作，所有基于AHBCLK或APBCLK的外设都停止工作。但是，使用独立与总线时钟工作的外设仍可以继续工作，比如UARTx、LPUARTx、I2Cx、ATIM、LPTIM32、LPTIM16、BSTIM32、BSTIM16。

为了让上述外设在休眠模式下继续工作，软件需要在休眠前确保上述外设使用SYSCLK和总线时钟

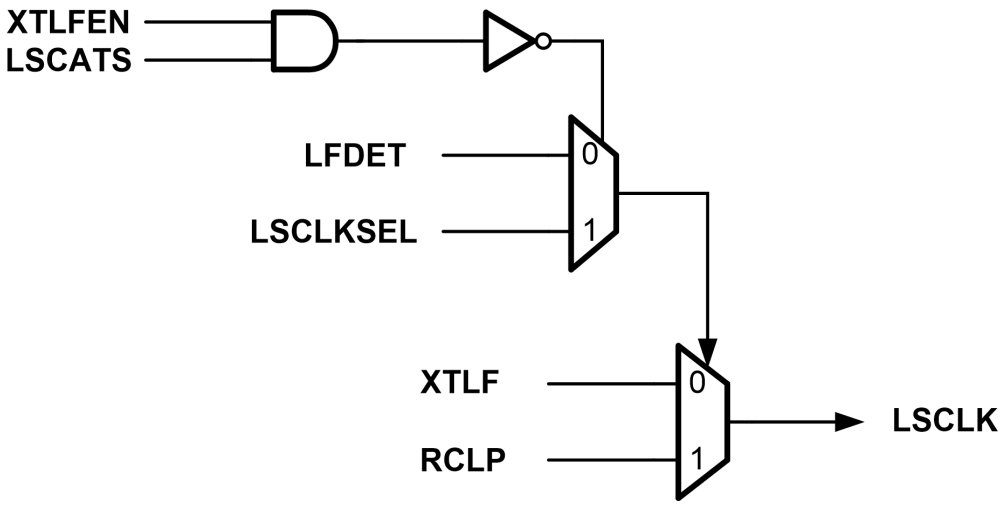
以外的时钟工作。

11.2.7 LSCLK 切换逻辑

LSCLK是供RTC、SVD和LCD驱动使用的低速时钟，典型频率32K左右。LSCLK的源头是XTLF或者RCLP，芯片支持两者之间的自动切换或者由软件手动切换。

LSCLK的自动切换功能，由LSACTS寄存器配置，仅在XTLF使能的情况下有效，此时假设XTLF是主时钟，RCLP是备份时钟，仅用于防止XTLF异常停振。所以LSCATS只在XTLF使能的情况下起作用，当XTLF出现意外停振，FDET输出的停振信号将LSCLK自动切换到RCLP。

在LSCLK自动切换不使能的情况下，软件可以通过LSCLKSEL寄存器实现对LSCLK的手动切换。



芯片上电复位后，XTLFEN=0，LFDET输出不停振，LSCLKSEL默认选中RCLP，XTLF无输出；因此上电后LSCLK默认是RCLP。

此后，软件如果希望使用XTLF，则应当使能XTLF并轮询LFDET输出，直到确认XTLF起振后，将LSCATS置位，或者清零LSCLKSEL。

当LSCLK用作系统时钟（SYSCLK）时，建议软件打开停振自动切换功能。

11.2.8 RTC 时钟

RTC可以使用XTLF、RCLP和XTHF时钟工作。通常情况下，使用32768Hz的XTHF晶振来做RTC高精度计时。使用RCLP时，RTC走时精度无法保证。使用XTHF时，为了得到精确的定时，应使用32768Hz的整数倍频率晶体，比如4.096MHz或者8.192MHz晶体。

XTHF提供给RTC使用前，先经过一个预分频电路，得到尽可能接近32768Hz的时钟。假设使用

4.096MHz晶体，预分频系数为125的情况下，可以得到准确的32768Hz时钟。预分频电路未同步计数器设计，输出分频时钟不保证为50%占空比。

11.3 高频晶体振荡电路(XTHF)

11.3.1 概述

通过外接高频晶体，XTHF能够为MCU提供高精度的高频时钟源。静态和负载电容应尽可能靠近XTHF引脚布置，其中负载电容大小应合理选择，以适配所选用的晶体类型。

XTHF可以适配4~16MHz晶体。软件可以通过XTHFEN寄存器使能或关闭XTHF时钟。

XTHF包含4pF内部负载电容，可以在一定范围内微调振荡频率。

11.3.2 工作方式

XTHF上电后默认关闭。上电复位完成后，软件可以根据需要打开XTHF。由于晶振引脚与GPIO复用，软件使能XTHF前，需要将PA8和PA9引脚配置为模拟功能。

11.3.3 停振检测（HFDET）

FM33FG0A带有片上停振检测电路，与XTHF电路一起使能或关闭。停振检测使能后可以持续检测XTHF输出，当发现XTHF停振时，会产生报警中断，同时产生高级定时器刹车信号；如果XTHF正在被直接或者间接的用作系统工作时钟（直接指SYSCLK选为XTHF，间接指SYSCLK选为PLL同时PLL使用XTHF为输入参考时钟），则停振信号将自动使能RCHF并将SYSCLK切换到RCHF，以避免高频晶体意外停振导致系统异常。

停振检测电路总是与XTHF同时打开或关闭，无法单独关闭，一旦XTHF使能，停振检测电路就会自动打开；当XTHF关闭时，停振检测也会自动关闭，避免误触发停振报警。

HFDET停振检测阈值约200KHz，即XTHF时钟频率低于200KHz时触发停振报警；HFDET工作电流约1.2uA。

11.3.4 XTHF 启动过程

软件使能XTHF后，硬件等待XTHF输出稳定后自动置位RDY标志寄存器。

XTHF关闭或者HFDET检测到停振时，会自动清零计数器以及RDY标志。

软件使能XTHF后应等待RDY标志置位，再将XTHF作为系统时钟使用。

11.4 低频晶体振荡电路(XTLF)

11.4.1 概述

低频晶体振荡电路通过外接32768Hz晶体提供稳定的振荡源，功耗极低，主要用来给实时时钟(RTC)模块提供输入时钟。XTLF的振荡强度可调，用户可根据需要选择振荡强度，达到振荡能力与功耗的平衡。XTLF的反馈电阻集成在芯片内部，用户需要在振荡引脚上外加负载电容。

芯片内部集成了一个停振检测电路（LFDET），用来检测XTLF是否停振。一旦检测到XTLF停振，将产生XTLF停振中断，通知CPU及时处理。

软件可以使能或关闭XTLF。为了提高抗干扰能力，采用4bit的XTLFEN控制位，4bit复位值为0101，必须改写为1010才能关闭XTLF，其他任何数据都会保持XTLF使能。

XTLF也可以选择直接从外部输入时钟工作，输入时钟从XT32KI引脚灌入，此时需要将XTLF模块关闭，将XT32KI引脚配置为GPIO输入，并置位EXCKSEN寄存器。EXCKSEN寄存器用于控制时钟MUX选择GPIO输入还是XTLF输出。

11.4.2 工作方式

XTLF上电后默认关闭，软件启动，默认使用中等强度，以缩短起振时间，相应的振荡功耗也较大。典型的起振时间小于1s。当振荡器充分起振后，软件可以通过配置寄存器降低振荡功耗。

11.4.3 停振检测（LFDET）

FM33FG0A带有片上停振检测电路，使能后可以持续检测XTLF输出，当发现XTLF停振时，产生报警中断，软件可以通过LSCATS寄存器决定是否自动将LSCLK切换到RCLP。

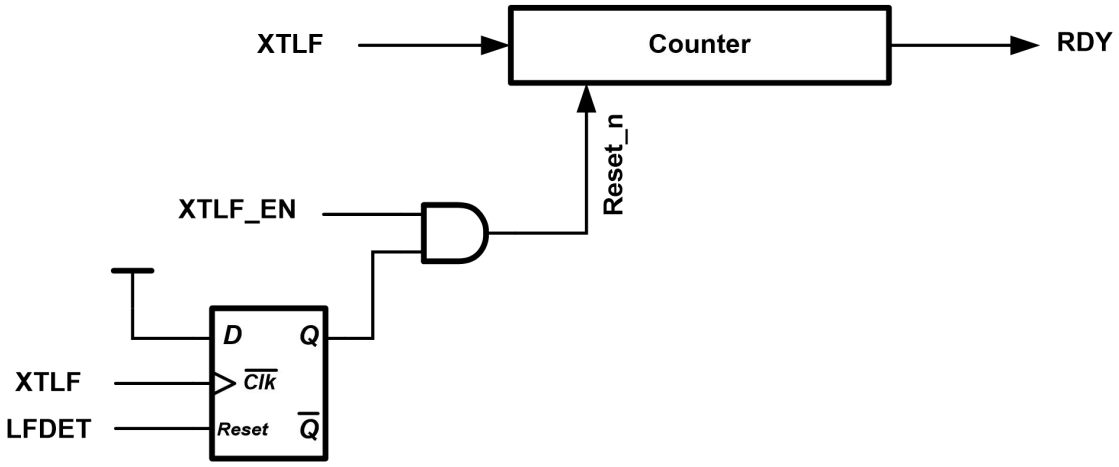
当LSCATS=1时，FDET检测到XTLF停振时，硬件会自动使能LPOSC并将LSCLK切换为RCLP输出；当LSCATS=0时，停振检测只会产生报警中断，并不会自动切换时钟。

在XTLF停振状态下，软件也可以通过置位LSCATS来切换XTLF。

停振检测电路总是与XTLF同时打开或关闭，无法单独关闭，一旦XTLF使能，停振检测电路就会自动打开；当XTLF关闭时，停振检测也会自动关闭，避免误触发停振报警。

11.4.4 XTLF 启动过程

软件使能XTLF后，硬件等待XTLF输出稳定后自动置位RDY标志寄存器。



XTLF关闭或者LFDET检测到停振时，会自动清零计数器以及RDY标志。

软件使能XTLF后应等待RDY标志置位，再使用XTLF时钟。

11.5 低功耗低频环振(RCLP)

11.5.1 概述

RCLP是一个超低功耗内部环振，典型功耗仅400nA左右，典型输出频率32KHz。这个环振主要用作独立看门狗工作时钟，上电后自动启动，不可关闭。

11.6 锁相环(PLL)

11.6.1 概述

锁相环输入参考时钟可以是RCHF或XTHF分频，输入频率固定为1MHz，最高输出频率可达64MHz。软件使用PLL作为系统时钟前，需配置输入参考时钟和倍频系数。

11.6.2 锁定检测

PLL使能后，经过一段时间后输出锁定，此时PLL锁定标志置位。当PLL在工作中发生失锁，失锁标志置位（LOL），此时可以产生中断或者直接复位芯片。

11.6.3 应用注意

出于可靠性考虑，软件需注意以下几点：

- 软件选择PLL输入时必须保证RCHF或XTHF为使能状态
- PLL输出选为SYSCLK时不能关闭PLL
- 软件应等待PLL锁定后再将SYSCLK配置为PLL输出

11.7 低功耗模式下的时钟源

在低功耗模式下，部分时钟源被硬件强制关闭，而另外一部分时钟源则仍可以保持工作。具体参见下表：

时钟源	LPRUN/Sleep/DeepSleep	说明
RCHF	X	硬件强制关闭
PLL	X	
XTHF	X	
RCLP	O	软件配置使能或关闭
XTLF	O	

11.8 休眠唤醒的时钟处理

当芯片从Sleep/DeepSleep模式下唤醒时，硬件自动打开RCHF并恢复到休眠前的频率输出；同时将SYSCLKSEL寄存器复位成00，将系统时钟选为RCHF，而AHBPRES寄存器不会被复位，保持休眠前的状态；因此芯片唤醒后默认将使用RCHF或者其分频时钟工作。

11.9 寄存器

模块起始地址：0x4000_2400

offset 地址	名称	符号
0x00	系统时钟控制寄存器（System Clock Control Register）	CMU_SYSCLKCR
0x04	RCHF 控制寄存器（RCHF Control Register）	CMU_RCHFCR
0x08	RCHF 调校寄存器（RCHF Trim Register）	CMU_RCHFTR
0x0C	PLL 控制寄存器（PLL Control Register）	CMU_PLLCR
0x14	RCLP 调校寄存器（RCLP Trim Register）	CMU_RCLPTR
0x18	XTLF 控制寄存器（XTLF Control Register）	CMU_XTLFCR
0x1C	LSCLK 选择寄存器（LSCLK Select Register）	CMU_LSCLKSEL
0x20	XTHF 控制寄存器（XTHF Control Register）	CMU_XTHFCR
0x2C	CMU 中断使能寄存器（Interrupt Enable Register）	CMU_IER
0x30	CMU 中断标志寄存器（Interrupt Status Register）	CMU_ISR
0x34	外设总线时钟控制寄存器 1（Peripheral bus Clock Control Register1）	CMU_PCLKCR1
0x38	外设总线时钟控制寄存器 2（Peripheral bus Clock Control Register2）	CMU_ PCLKCR2
0x3C	外设总线时钟控制寄存器 3（Peripheral bus Clock Control Register3）	CMU_ PCLKCR 3
0x40	外设总线时钟控制寄存器 4（Peripheral bus Clock Control Register4）	CMU_ PCLKCR 4
0x44	外设工作时钟配置寄存器 1（Peripheral Clock Config Register1）	CMU_OPCCR1
0x48	外设工作时钟配置寄存器 2（Peripheral Clock Config Register 2）	CMU_OPCCR2
0x4C	外设工作时钟配置寄存器 3（Peripheral Clock Config Register 3）	CMU_OPCCR3
0x50	AHB Master 控制寄存器（AHB Master Control Register）	CMU_AHBMCR
0x80	CANFD 时钟配置寄存器（CANFD Clock Config Register）	CMU_CFDCR
0x84	CANFD 时钟使能寄存器	CMU_CFDER

offset 地址	名称	符号
	（CANFD Clock Enable Register）	

11.9.1 系统时钟配置寄存器（CMU_SYSCLKCR）

名称	CMU_SYSCLKCR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				LSCATS	-	SLP_EN EXTI	-
位权限	U-0				R/W-1	U-0	R/W-1	U-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		APB2PRES			APB1PRES		
位权限	U-0		R/W-000			R/W-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					AHBPRES		
位权限	U-0					R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	STCLKSEL		-			SYSCLKSEL		
位权限	R/W-00		U-0			R/W-000		

Bit	助记符	功能描述
31:28	--	RFU：未实现，读为 0
27	LSCATS	LSCLK 自动切换使能 0：当检测到 XTLF 异常停振时，不会自动将 LSCLK 切换到 RCLP，软件可以通过写 LSCLKSEL 寄存器手动切换到 RCLP 1：当检测到 XTLF 异常停振时，自动使能 RCLP 并将 LSCLK 切换到 RCLP
26	--	RFU：未实现，读为 0
25	SLP_ENEXTI	Sleep/DeepSleep 模式下 EXTI 采样设置 1：Sleep/DeepSleep 模式下使能外部引脚中断采样（采样时钟为 LSCLK） 0：Sleep/DeepSleep 模式下禁止外部引脚中断采样（将无法产生 EXTI 中断）
24:22	--	RFU：未实现，读为 0
21:19	APB2PRES	APB2 时钟分频选择 0xx：不分频 100：2 分频 101：4 分频 110：8 分频 111：16 分频
18:16	APB1PRES	APB1 时钟分频选择 0xx：不分频 100：2 分频 101：4 分频 110：8 分频

Bit	助记符	功能描述
		111：16 分频
15:11	--	RFU：未实现，读为 0
10:8	AHBPRES	AHB 时钟分频选择 0xx：不分频 100：2 分频 101：4 分频 110：8 分频 111：16 分频
7:6	STCLKSEL	CPU 内核 systick 工作时钟选择 00：SCLK 01：LSCLK 10：RFU 11：RFU
5:3	--	RFU：未实现，读为 0
2:0	SYSCLKSEL	系统时钟源选择 000：RCHF 001：XTHF 010：PLL 011：RCHF 100：RCHF 101：XTLF 110：RCLP 111：RCHF

11.9.2 RCHF 时钟控制寄存器（CMU_RCHCR）

名称	CMU_RCHFCR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				FSEL			
位权限	U-0				R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-1

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19:16	FSEL	RCHF 频率选择寄存器 0000：8MHz 0001：16MHz 0010：24MHz 0011：32MHz

Bit	助记符	功能描述
		其他：RFU
15:1	--	RFU：未实现，读为 0
0	EN	RCHF 使能寄存器 1：使能 RCHF 0：关闭 RCHF

11.9.3 RCHF 调校寄存器（CMU_RCHFTR）

名称	CMU_RCHFTR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIM							
位权限	R/W-1000 0000							

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	TRIM	RCHF 频率调校寄存器，8'h00 表示频率最低，8'hFF 表示频率最高，调校范围为中心频率+/-30%，调校步长为中心频率 0.5% 上电后芯片自动从 LDT0 读取 8MHz 调校值并写入此寄存器 软件使用其他频率时，可以自行从 LDT0 指定地址读取调校信息并写入此寄存器，从而确保输出频率准确。

11.9.4 PLL 控制寄存器（CMU_PLLCR）

名称	CMU_PLLCR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						DB	
位权限	U-0						R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DB							
位权限	R/W-0001 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCKED	REFPRSC			-	-	INSEL	EN
位权限	R/Dy-0	R/W-000			U-0	U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:26	--	RFU：未实现，读为 0
25:16	DB	PLL 倍频比，输出时钟频率为 1M*(PLLDB+1) 0x1F：输出 32 倍频 0x2F：输出 48 倍频 …
15:8	--	RFU：未实现，读为 0
7	LOCKED	PLL 锁定标志，软件通过查询此寄存器确认 PLL 已经处于锁定状态 1：PLL 已锁定 0：PLL 未锁定
6:4	REFPRSC	PLL 参考时钟预分频（目标是产生 1MHz 参考时钟给 PLL） 000：不分频 001：2 分频 010：4 分频 011：8 分频 100：12 分频 101：16 分频 110：24 分频 111：32 分频
3	--	RFU：未实现，读为 0
2	--	RFU：未实现，读为 0
1	INSEL	PLL 输入选择寄存器 0：RCHF 1：XTHF
0	EN	PLL 使能寄存器 1：使能 PLL 0：关闭 PLL

11.9.5 RCLP 调校寄存器（CMU_RCLPTR）

名称	CMU_RCLPTR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIM							
位权限	R/W-1000 0000							

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0

Bit	助记符	功能描述
7:0	TRIM	RCLP 调校值寄存器 0000 0000：频率最低 …… 1111 1111：频率最高

11.9.6 XTLF 控制寄存器（CMU_XTLFCR）

注意，此寄存器仅受 POR 和 PDR 复位影响，不受 BOR 复位、看门狗复位和软复位影响

名称	CMU_XTLFCR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							BYPASS
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	WAIT			EXCKSEN	EN			
位权限	R/W-111			R/W-0	R/W-0			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		RDY	GMCFG			ICFG	
位权限	U-0		R-0	RW-100			RW-01	

Bit	助记符	功能描述
31:17	--	RFU：未实现，读为 0
16	BYPASS	Bypass 使能信号，高为 bypass 使能，可通过 XTALIN 外部输入时钟，该寄存器仅在 EN=1 时有效
15:13	WAIT	XTLF 启动等待时间配置，单位是 XTLF 时钟周期 000：128 001：256 010：512 011：1024 100：2048 101：4096 110：8192 111：16384
12	EXCKSEN	外部输入低频时钟使能 0：芯片内部使用 XTLF 输出 1：芯片内部使用 XT32KI 引脚输入的外部时钟
11:8	EN	XTLF 使能寄存器，上电默认 XTLF 关闭 1010：关闭 XTLF 和 FDET 0101：使能 XTLF 和 FDET 物理上只有 1bit，当 XTLF 工作时，软件必须写入 1010 才能将其关闭，当 XTLF 不工作时，软件必须写入 0101 才能启动
7:6	--	RFU：未实现，读为 0

Bit	助记符	功能描述
5	RDY	XTLF 稳定标志，硬件置位，只读 关闭 XTLF 或者 XTLF 发生停振时自动清零
4:2	GMCFG	GM 配置信号
1:0	ICFG	XTLF 输入电流源配置信号

11.9.7 LSCLK 选择寄存器（CMU_ LSCLKSEL）

名称	CMU_LSCLKSEL							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							STA
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SEL							
位权限	R/W-xxxx xxxx							

Bit	助记符	功能描述
31:9	--	RFU：未实现，读为 0
8	STA	LSCLK 状态标志寄存器，只读 0：当前 LSCLK 源头是 XTLF 1：当前 LSCLK 源头是 RCLP
7:0	SEL	LSCLK 时钟手动切换寄存器，物理实现上只有 1bit；复位值选中 RCLP； 当 LSCLK 为 XTLF 时，软件对此地址写 0x55，会将 LSCLK 源头切换到 RCLP 当 LSCLK 为 RCLP 时，软件对此地址写 0xAA，会将 LSCLK 源头切换到 XTLF 写任意其他值，不改变当前 LSCLK；此寄存器仅在 LSCATS 为 0 时有效

11.9.8 XTHF 控制寄存器（CMU_XTHFCR）

名称	CMU_XTHFCR							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFUI	-						
位权限	R/W-0	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	WAIT			BYP	HF_CFG			
位权限	R/W-111			R/W-0	R/W-00000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						RDY	EN
位权限	U-0						R-0	R/W-0

Bit	助记符	功能描述
31	RFUI	保留
30:16	--	RFU：未实现，读为 0
15:13	WAIT	XTHF 启动等待时间配置，单位是 XTHF 时钟周期 000：128 001：256 010：512 011：1024 100：2048 101：4096 110：8192 111：16384
12	BYP	Bypass 使能信号，高有效 可通过引脚输入时钟
11:8	HF_CFG	XTHF 振荡强度配置 0000：最弱 1111：最强
7:2	--	RFU：未实现，读为 0
1	RDY	XTHF 稳定标志，硬件置位，只读 关闭 XTHF 或者 XTHF 发生停振时自动清零
0	EN	XTHF 使能寄存器 0：关闭 XTHF 1：使能 XTHF

11.9.9 CMU 中断使能寄存器（CMU_IER）

名称	CMU_IER							
offset	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				-	SYSCSE_IE	HFDET_IE	LFDET_IE

位权限	U-0	U-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3	--	RFU：未实现，读为 0
2	SYSCKE_IE	SYSCLK 时钟选择错误中断使能寄存器，1 有效
1	HFDET_IE	XTHF 高频检测报警中断使能，1 有效
0	LFDET_IE	XTLF 高频检测报警中断使能，1 有效

11.9.10 CMU 中断标志寄存器（CMU_ISR）

名称	CMU_ISR							
offset	0x30							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						HFDETO	LFDETO
位权限	U-0						R-1	R-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				LOLIF	SYSCSE_IF	HFDETIF	LFDETIF
位权限	U-0				R/W-0	R/W-0	R/W -0	R/W -0

Bit	助记符	功能描述
31:10	--	RFU：未实现，读为 0
9	HFDETO	高频晶体停振检测模块输出 1：XTHF 未停振 0：XTHF 停振
8	LFDETO	低频晶体停振检测模块输出 1：XTLF 未停振 0：XTLF 停振
7:4	--	RFU：未实现，读为 0
3	LOLIF	PLL 失锁中断标志 在 PLL 锁定的情况下发生失锁，此标志置位。软件写 1 清零。
2	SYSCSE_IF	SYSCLK 时钟选择错误中断标志。 当被选择的目标时钟没有使能，或者已经停振时，时钟切换被禁止，同时置位此标志寄存器。软件写 1 清零。
1	HFDETIF	高频停振检测中断标志寄存器，XTHF 停振时硬件异步置位，软件写 1 清零；只有在 FFDETO 不为 0 的情况下才能够清除此寄存器
0	LFDETIF	低频停振检测中断标志寄存器，XTLF 停振时硬件异步置位，软件写 1 清零；只有在 FFDETO 不为 0 的情况下才能够清除此寄存器

11.9.11 外设总线时钟控制寄存器 1（CMU_ PCLKCR1）

名称	CMU_PCLKCR1							
offset	0x34							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				CLM1_PCE	CLM0_PCE	COMP_PCE	SVD_PCE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PAD_PCE	ANT_PCE	IWDT_PCE	SCU_PCE	PMU_PCE	RTC_PCE	LPT16_PCE	LPT32_PCE
位权限	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11	CLM1_PCE	CLM1 总线时钟使能，高使能
10	CLM0_PCE	CLM0 总线时钟使能，高使能
9	COMP_PCE	Comparator 总线时钟使能，高使能
8	SVD_PCE	SVD 总线时钟使能，高使能
7	PAD_PCE	PADCFG 总线时钟使能，高使能
6	ANT_PCE	模拟测试 buffer 总线时钟使能，高使能 此寄存器用于控制 BUF4TST 的总线时钟
5	IWDT_PCE	IWDT 总线时钟使能，高使能
4	SCU_PCE	SCU 总线时钟使能，高使能
3	PMU_PCE	PMU 总线时钟使能，高使能
2	RTCA_PCE	RTCA 总线时钟使能，高使能
1	LPT16_PCE	LPTIM16 总线时钟使能，高使能
0	LPT32_PCE	LPTIM32 总线时钟使能，高使能

11.9.12 外设总线时钟控制寄存器 2（CMU_ PCLKCR2）

名称	CMU_PCLKCR2							
offset	0x38							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				PGL_PCE	DAC_PCE	HDIV_PCE	ADC_PCE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WWDT_PCE	RAMBIST_PCE	NVM_PCE	DMA_PCE	-	AES_PCE	RNG_PCE	CRC_PCE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11	PGL_PCE	PGL 总线时钟使能，高使能
10	DAC_PCE	DAC 总线时钟使能，高使能
9	HDIV_PCE	HDIV 总线时钟使能，高使能
8	ADC_PCE	ADC 总线时钟使能，高使能
7	WWDT_PCE	WWDT 总线时钟使能，高使能
6	RAMBIST_PCE	RAMBIST 总线时钟使能，高使能
5	NVM_PCE	NVMIF（Flash 擦写控制器）总线时钟使能，高使能
4	DMA_PCE	DMA 总线时钟使能，高使能
3	--	RFU：未实现，读为 0
2	AES_PCE	AES 总线时钟使能，高使能
1	RNG_PCE	RNG 总线时钟使能，高使能
0	CRC_PCE	CRC 总线时钟使能，高使能

11.9.13 外设总线时钟控制寄存器 3（CMU_ PCLKCR3）

名称	CMU_PCLKCR3							
offset	0x3C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SENT1_PCE	SENT0_PCE	I2CSMB1_PCE	I2CSMB0_PCE	-		I2C1_PCE	I2C0_PCE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0		R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CANFD1_PCE	CANFD0_PCE	-		FSCAN_PCE	LPUART2_PCE	LPUART1_PCE	LPUART0_PCE
位权限	R/W-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	UCIR_PCE	UART5_PCE	UART4_PCE	UART3_PCE	UART2_PCE	UART1_PCE	UART0_PCE
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				SPI3_PCE	SPI2_PCE	SPI1_PCE	SPI0_PCE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31	SENT1_PCE	SENT1 总线时钟使能，高有效
30	SENT0_PCE	SENT0 总线时钟使能，高有效
29	I2CSMB1_PCE	I2C_SMBUS1 总线时钟使能，高有效
28	I2CSMB0_PCE	I2C_SMBUS0 总线时钟使能，高有效
27:26	-	RFU：未实现，读为 0
25	I2C1_PCE	I2C1 总线时钟使能，高有效
24	I2C0_PCE	I2C0 总线时钟使能，高有效

位号	助记符	功能描述
23	CANFD1_PCE	CANFD1 总线时钟使能，高有效
22	CANFD0_PCE	CANFD0 总线时钟使能，高有效
21:20	-	RFU：未实现，读为 0
19	FSCAN_PCE	FSCAN 总线时钟使能，高有效
18	LPUART2_PCE	LPUART2 总线时钟使能，高有效
17	LPUART1_PCE	LPUART1 总线时钟使能，高有效
16	LPUART0_PCE	LPUART0 总线时钟使能，高有效
15	-	RFU：未实现，读为 0
14	UCIR_PCE	UART 红外调制工作时钟使能，高有效
13	UART5_PCE	UART5 总线时钟使能，高有效
12	UART4_PCE	UART4 总线时钟使能，高有效
11	UART3_PCE	UART3 总线时钟使能，高有效
10	UART2_PCE	UART2 总线时钟使能，高有效
9	UART1_PCE	UART1 总线时钟使能，高有效
8	UART0_PCE	UART0 总线时钟使能，高有效
7:4	-	RFU：未实现，读为 0
3	SPI3_PCE	SPI3 总线时钟使能，高有效
2	SPI2_PCE	SPI2 总线时钟使能，高有效
1	SPI1_PCE	SPI1 总线时钟使能，高有效
0	SPI0_PCE	SPI0 总线时钟使能，高有效

11.9.14 外设总线时钟控制寄存器 4（CMU_ PCLKCR4）

名称	CMU_PCLKCR4							
offset	0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							BT16_PCE
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AT_PCE	-	TAU1_PCE	TAU0_PCE	GT2_PCE	GT1_PCE	GT0_PCE	BT32_PCE
位权限	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:9	-	RFU：未实现，读为 0
8	BT16_PCE	BSTIM16 总线时钟使能，高有效
7	AT_PCE	高级定时器总线时钟使能，高有效
6	-	RFU：未实现，读为 0
5	TAU1_PCE	定时器阵列 1 总线时钟使能，高有效
4	TAU0_PCE	定时器阵列 0 总线时钟使能，高有效

位号	助记符	功能描述
3	GT2_PCE	通用定时器 2 总线时钟使能，高有效
2	GT1_PCE	通用定时器 1 总线时钟使能，高有效
1	GT0_PCE	通用定时器 0 总线时钟使能，高有效
0	BT32_PCE	BSTIM32 总线时钟使能，高有效

11.9.15 外设工作时钟配置寄存器 1（CMU_OPCCR1）

名称	CMU_OPCCR1							
offset	0x44							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	EXTICKS	UART3CKS		LPUART2CKS		LPUART1CKS	
位权限	U-0	R/W-0	R/W-00		R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LPUART0CKS		UART2CKS		I2CSMB1CKS		I2CSMB0CKS	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BT16CKS		BT32CKS		LPT16CKS		LPT32CKS	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ATCKS	-	CANCKS		UART1CKS		UART0CKS	
位权限	R/W-0	U-0	R/W-00		R/W-00		R/W-00	

位号	助记符	功能描述
31	-	RFU：未实现，读为 0
30	EXTICKS	EXTI 中断采样时钟选择 1：外部引脚中断使用 LSCLK 采样 0：外部引脚中断使用 AHBCLK 采样 *建议在关闭所有 EXTI 中断的情况下设置，设置完成后再使能 EXTI 中断
29:28	UART3CKS	UART3 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：XTHF
27:26	LPUART2CKS	LPUART1 工作时钟选择 00：LSCLK 01：RCHF 分频（根据 RCHF 档位自动分频到 32768Hz 附近） 10：RFU 11：RFU
25:24	LPUART1CKS	LPUART1 工作时钟选择 00：LSCLK 01：RCHF 分频（根据 RCHF 档位自动分频到 32768Hz 附近） 10：RFU 11：RFU
23:22	LPUART0CKS	LPUART2 工作时钟选择 00：LSCLK

位号	助记符	功能描述
		01：RCHF 分频（根据 RCHF 档位自动分频到 32768Hz 附近） 10：RFU 11：RFU
21:20	UART2CKS	UART2 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：XTHF
19:18	I2CSMB1CKS	I2C_SMB1 主机工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：RFU
17:16	I2CSMB0CKS	I2C_SMB0 主机工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：RFU
15:14	BT16CKS	BSTIM16 工作时钟源选择 00：APBCLK 01：LSCLK 10：RCLP 11：RFU
13:12	BT32CKS	BSTIM32 工作时钟源选择 00：APBCLK 01：LSCLK 10：RCLP 11：RFU
11:10	LPT16CKS	LPTIM16 工作时钟选择 00：APBCLK 01：LSCLK 10：RCLP 11：RFU
9:8	LPT32CKS	LPTIM32 工作时钟选择 00：APBCLK 01：LSCLK 10：RCLP 11：RFU
7	ATCKS	ATIM 工作时钟源选择寄存器 0：APBCLK 1：PLL 两倍频
6	-	RFU：未实现，读为 0
5:4	CANCKS	FSCAN 工作时钟选择（CAN_CLK 工作频率范围是 8~24Mhz） 00：RCHF 01：XTHF 10：PLL 11：APBCLK
3:2	UART1CKS	UART1 工作时钟选择 00：APBCLK

位号	助记符	功能描述
		01：RCHF 10：SYSCLK 11：XTHF
1:0	UART0CKS	UART0 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：XTHF

11.9.16 外设工作时钟配置寄存器 2（CMU_OPCCR2）

名称	CMU_OPCCR2							
offset	0x48							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RTCPRSC							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RTCPRSC		-					RTCCKS
位权限	R/W-00		U-0					R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			RNGPRSC			-	
位权限	U-0			R/W-000			U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			ADCPRSC			ADCCKS	
位权限	U-0			R/W-000			R/W-00	

Bit	助记符	功能描述
31:22	RTCPRSC	使用 XTHF 作为 RTC 时的 XTHF 预分频系数，分频比为： N= RTCPRSC+1 比如 RTCPRSC=124，分频比=125，当 XTHF 外接 4.096MHz 晶体时可以产生 32768Hz 时钟给 RTC
21:17	--	RFU：未实现，读为 0
16	RTCCKS	RTC 工作时钟选择 0：LSCLK 1：XTHF 分频
12:10	RNGPRSC	随机数发生器工作时钟分频 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, 111：RFU
9:5	--	RFU：未实现，读为 0
4:2	ADCPRSC	ADC 工作时钟预分频 000：不分频 001：2 分频

Bit	助记符	功能描述
		010：4 分频 011：8 分频 100：16 分频 101：32 分频 110, 111：RFU
1:0	ADCCKS	ADC 工作时钟选择 00：RFU 01：RCHF 10：XTHF 11：PLL

11.9.17 外设工作时钟使能寄存器（CMU_OPCCR3）

名称	CMU_OPCCR3							
offset	0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTICKE	NVMCKE	LPUART0CKE	LPUART1CKE	LPUART2CKE	-		RNGCKE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		I2CSMB1_CKE	I2CSMB0_CKE	-			ADCCKE
位权限	U-0		R/W-0	R/W-0	U-0			R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ATCKE	-	CANCKE	LINCKE	UART3CKE	UART2CKE	UART1CKE	UART0CKE
位权限	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				BT16CKE	BT32CKE	LPT16CKE	LPT32CKE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31	EXTICKE	外部引脚中断采样时钟使能，高有效
30	NVMCKE	Flash 擦写时钟使能，高有效
29	LPUART0CKE	LPUART0 工作时钟使能，高有效
28	LPUART1CKE	LPUART1 工作时钟使能，高有效
27	LPUART2CKE	LPUART2 工作时钟使能，高有效
26:25	--	RFU：未实现，读为 0
24	RNGCKE	随机数发生器工作时钟使能，高有效 *注意：芯片休眠后此寄存器被复位*
23:22	--	RFU：未实现，读为 0
21	I2CSMB1_CKE	I2C_SMB1 工作时钟使能，高有效
20	I2CSMB0_CKE	I2C_SMB0 工作时钟使能，高有效
19:17	--	RFU：未实现，读为 0
16	ADCCKE	ADC 工作时钟使能，高有效
15	ATCKE	高级定时器工作时钟使能，高有效
14		

Bit	助记符	功能描述
13	CANCKE	FSCAN 总线控制器工作时钟使能，高有效
12	LINCKE	LINCLK 使能，用于唤醒信号检测，高有效
11	UART3CKE	UART3 工作时钟使能，高有效
10	UART2CKE	UART2 工作时钟使能，高有效
9	UART1CKE	UART1 工作时钟使能，高有效
8	UART0CKE	UART0 工作时钟使能，高有效
7:4	--	RFU：未实现，读为 0
3	BT16CKE	BSTIM16 工作时钟使能，高有效
2	BT32CKE	BSTIM32 工作时钟使能，高有效
1	LPT16CKE	LPTIM16 工作时钟使能，高有效
0	LPT32CKE	LPTIM32 工作时钟使能，高有效

11.9.18 AHB Master 控制寄存器（CMU_AHBMCR）

名称	CMU_AHBMCR							
offset	0x50							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFUI	RFUI	-					
位权限	R/W-1	R/W-0	U-0					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							MPRIL
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:30	RFUI	保留位
29:1	--	RFU：未实现，读为 0
0	MPRIL	AHB Master 优先级配置寄存器 0：DMA 优先 1：CPU 优先

11.9.19 CANFD 时钟配置寄存器（CMU_CFDCR）

名称	CMU_CFDCR							
offset	0x80							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CFD1CLKC		CFD1CLKR		CFD0CLKC		CFD0CLKR	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
7:6	CFD1CLKC	CANFD1 通信时钟（CLKC） 00：APBCLK 01：XTHF 10：PLLx1 11：RCHF
5:4	CFD1CLKR	CANFD1 CLK_RAM 选择 00：AHBCLK，此时 CANFD_PCLK=APBCLK=AHBCLK/2 01：PLL_VCO，此时 CANFD_PCLK=APBCLK=AHBCLK 10,11：RFU
3:2	CFD0CLKC	CANFD0 通信时钟（CLKC） 00：APBCLK 01：XTHF 10：PLLx1 11：RCHF
1:0	CFD0CLKR	CANFD0 CLK_RAM 选择 00：AHBCLK，此时 CANFD_PCLK=APBCLK=AHBCLK/2 01：PLL_VCO，此时 CANFD_PCLK=APBCLK=AHBCLK 10,11：RFU

11.9.20 CANFD 时钟使能寄存器（CMU_CFDER）

名称	CMU_CFDER							
offset	0x84							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		CFD1CKCEN	CFD1CKREN	-		CFD0CKCEN	CFD0CKREN
位权限	U-0		R/W-0	R/W-0	U-0		R/W-0	R/W-0

Bit	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5	CFD1CKCEN	CANFD1 通信时钟使能，1 有效
4	CFD1CKREN	CANFD1 RAM 时钟使能，1 有效
3:2	--	RFU：未实现，读为 0
1	CFD0CKCEN	CANFD0 通信时钟使能，1 有效
0	CFD0CKREN	CANFD0 RAM 时钟使能，1 有效

注意：总线寄存器时钟使能在“外设总线时钟控制寄存器3”

12 时钟监控（CLM）

12.1 概述

本芯片共有2个时钟监控模块，用于交叉监控两个时钟之间确定的频率关系，当相对频率出现明显异常时产生报警中断或者全局复位。

12.2 结构框图

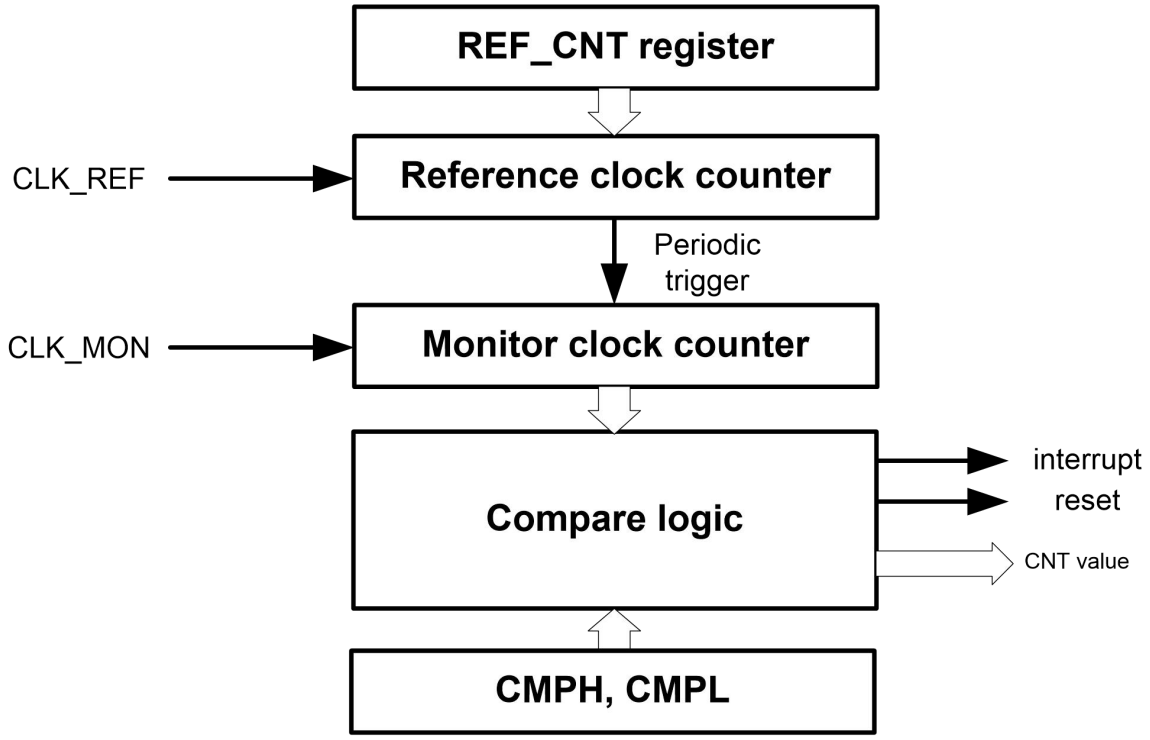


图 12-1CLM 电路框图

其中reference clock counter位宽8bit，monitor clock counter位宽24bit。CLK_REF固定为RCLP，CLK_MON可以从多个高频时钟里选择（RCHF、XTHF、PLL）。

12.3 功能描述

12.3.1 工作模式

CLM模块支持复位模式或中断模式。

在复位模式下（MODE=10），当计数值超过CMPH和CMPL设置的上下限，CLM输出全局复位信号。

在中断模式下（MODE=00/01/11），当计数值超过CMPH和CMPL设置的上下限，CLM输出中断信号。

12.3.2 使能与关闭

CLM模块上电后默认关闭，由软件完成配置后来使能，由于涉及功能安全，CLM控制寄存器受到写保护，以防止寄存器被异常改写，参见SFU章节。建议软件使能CLM后，通过SFU模块启动CLM模块寄存器写保护功能。

当CLK_MON为XTHF和PLL时，由于时钟启动需要时间，软件必须注意在确认监控时钟正常启动后再使能CLM模块，避免CLM出现误动作。

- CLK_MON=XTHF：使能CLM前应查询CMU_XTHFCR.RDY
- CLK_MON=PLL：使能CLM前应查询CMU_PLLCR.LOCKED

CLM与芯片低功耗模式

当芯片进入休眠模式或LPRUN模式时，由于CLK_MON的源头时钟都被自动关闭，此时CLM模块应停止工作，硬件自动关闭CLM，芯片回到ACTIVE模式时，软件需要重新配置并使能CLM模块。

CLM不能在芯片唤醒时自动使能，因为休眠时RCHF、XTHF、PLL都会被硬件强制关闭，唤醒后软件可能修改时钟频率，CLK_MON频率可能与休眠前不同，甚至可能软件没有使能XTHF和PLL，如果CLM自动恢复休眠前的工作状态，则可能导致意外的频率报警。

12.3.3 时钟频率监控

CLM频率监控用于监视CLK_MON的频率是否处于系统允许的正常范围之内，其工作原理是两个独立工作的定时器，分别由CLK_REF和CLK_MON驱动。其中参考时钟计数器的溢出周期由REF_CNT寄存器定义，每次溢出时输出触发信号清零监控时钟计数器，并且在触发点检查监控时钟计数器的值是否处于规定范围之内。

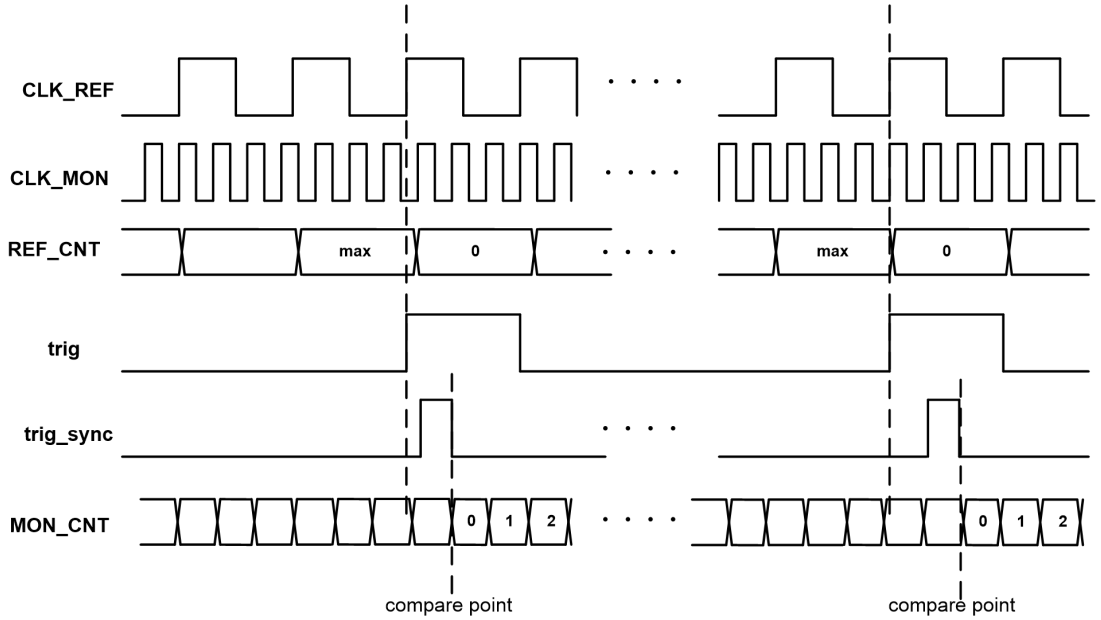


图 12-2 参考时钟计数和监控时钟计数

CMPH和CMPL定义了监控阈值的上下限，在一个监控周期内，如果计数值超过了上下阈值，则说明被监控的时钟频率异常，此时CLM产生异常事件，根据系统要求，此异常可以由软件处理或者直接产生系统复位。

下图中，CLK_MON频率异常，低于正常范围，导致UDF置位。

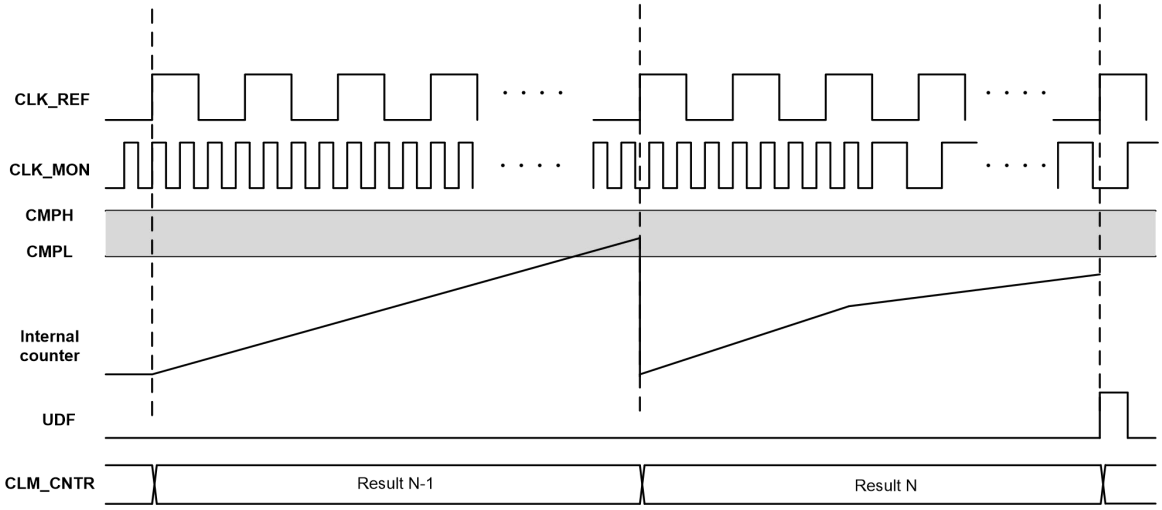


图 12-3 监控时钟频率下溢出

下图为监控时钟频率上溢出的例子。

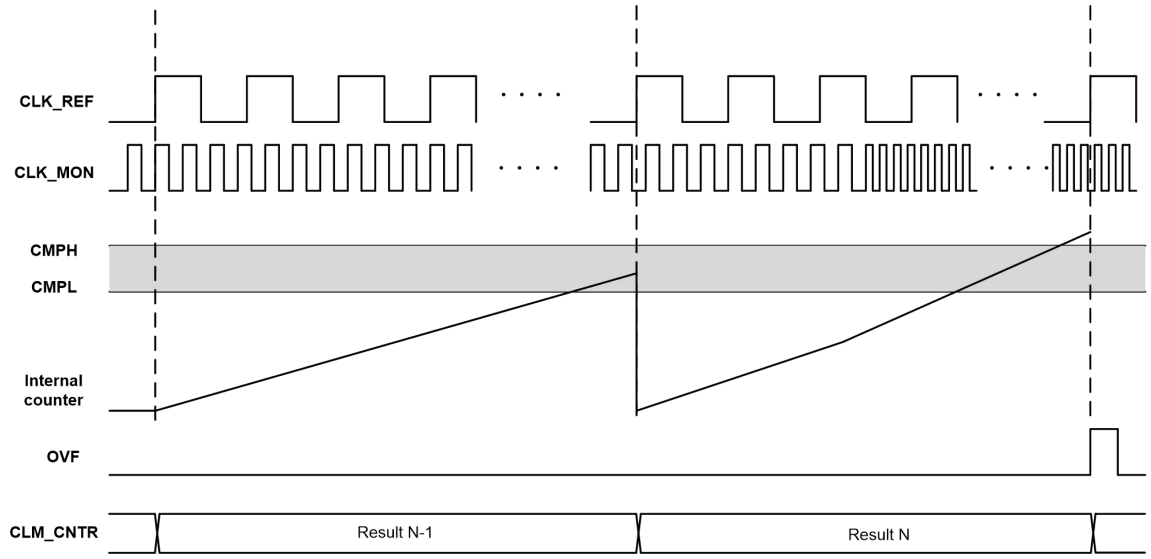


图 12-4 监控时钟频率上溢出

使用时钟监控功能需要注意以下几点：

- 为了避免监控计数器溢出，需要根据CLK_REF和CLK_MON之间的频率关系，合理设置参考时钟计数周期
- 设置CMPH和CMPL时要充分考虑时钟的温度系数，并留有足够余量；设置过小的阈值范围可能导致误报警
- 硬件不负责检查CMPH和CMPL的大小关系，只要计数值高于CMPH或低于CMPL就会报警
- 置位EN寄存器之前，先要完成参考时钟和监控时钟的选择

设置检测参数应考虑系统故障响应时间的要求。最差情况下，检测到系统时钟丢失需要REF_CLK一个完整监控周期，以下面的例子说明：

- 系统使用PLL=64MHz为主时钟工作
- 监控时钟为PLL，PLL的源时钟为XTHF，温度系数小于100ppm
- 参考时钟为RCLP，温度系数小于10%
- 检测PLL时钟丢失的最大故障响应时间是250us
- 考虑RCLP标称频率32KHz，可设置REF_CLK预分频系数为6，即检测周期约180us，考虑温度系数最大不超过200us
- 监控时钟在检测周期内的理想计数值应该是11520
- 设置CMPL=0.8*11520=9200，CMPH=1.2*11520=13800

12.3.4 超时检测

时钟频率监控无法检测到监控时钟完全停止的异常情况，因此需要超时检测机制识别到监控时钟计数器长时间没有变化的情况。CLM使能后，TO计数器使用RCLP计数，溢出周期为15。

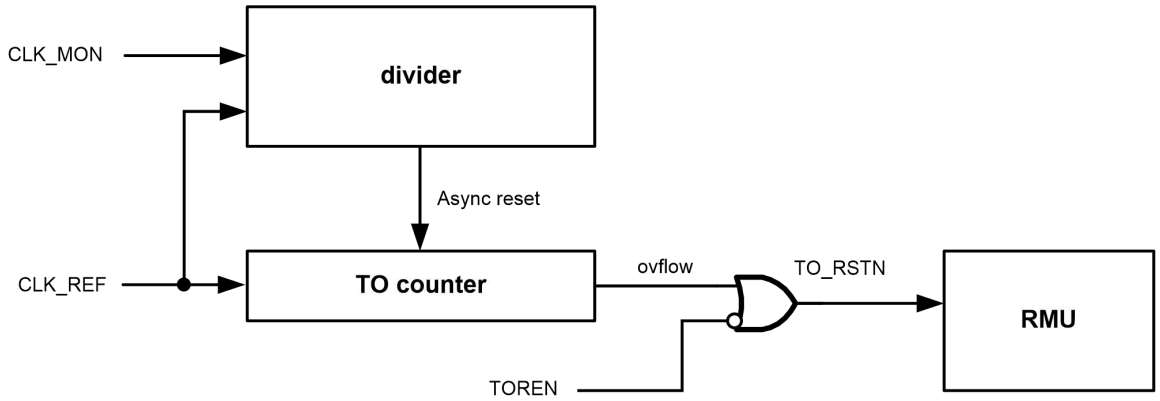


图 12-5 超时检测电路

12.4 寄存器

CLM0模块起始地址：0x4001B800

CLM1模块起始地址：0x4001BC00

offset 地址	名称	符号
0x00	时钟监控控制寄存器	CLM_CR
0x04	时钟监控配置寄存器	CLM_CFGR
0x08	时钟监控计数值寄存器	CLM_CNTR
0x0C	时钟监控中断标志寄存器	CLM_ISR
0x10	时钟监控比较高阈值寄存器	CLM_CMPH
0x14	时钟监控比较低阈值寄存器	CLM_CMPL

12.4.1 时钟监控控制寄存器（CLM_CR）

名称	CLM_CRx (x=0,1)							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TOREN	-						
位权限	R/W-1	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MODE		-		UDFIE	OVFIE	MONIE	EN
位权限	R/W-01		U-0		R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31	TOREN	Time-out reset enable，仅在 EN=1 时有效 0：禁止超时复位输出 1：允许超时复位输出
30:8	--	RFU：未实现，读为 0
7:6	MODE	CLM 工作模式选择 10：复位模式 其他：中断模式
5:4	--	RFU：未实现，读为 0
3	UDFIE	时钟监控计数器下溢出中断使能 0：禁止中断 1：允许中断
2	OVFIE	时钟监控计数器上溢出中断使能 0：禁止中断 1：允许中断
1	MONIE	时钟监控周期中断使能 0：禁止中断 1：允许中断
0	EN	CLM 使能 1：启动 0：关闭，内部计数器复位 *注：当芯片退出 Active 模式时，EN 位自动清零，CLM 关闭并回到初始态；芯片唤醒后，软件需要根据系统时钟设置，重新配置并使能 CLM*

12.4.2 时钟监控配置寄存器（CLM_CFGR）

名称	CLM_CFGRx (x=0,1)							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	REF_CNT							
位权限	R/W-00100000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						MONSEL	
位权限	U-0						R/W-00	

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0

Bit	助记符	功能描述
15:8	REF_CNT	参考时钟周期 检测周期=T_{REF_CLK}*(REF_CNT+1)
7:2	--	RFU：未实现，读为 0
1:0	MONSEL	监控时钟选择 (Monitor clock select) 00：RCHF 01：PLL 10：XTHF 11：XTHF

12.4.3 时钟监控计数值寄存器（CLM_CNTR）

名称	CLM_CNTRx (x=0,1)							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CNT[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

Bit	助记符	功能描述
31:24	--	RFU：未实现，读为 0
23:0	CNT	时钟监控计数器计数值 (Clock calibration counter) 每个监控周期结束后自动更新，供软件查询

12.4.4 时钟监控中断标志寄存器（CLM_ISR）

名称	CLM_ISRx (x=0,1)							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	UDF	OVF	CCL_IF
位权限	U-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	UDF	时钟监控下溢出标志，监控周期内计数值小于监控阈值下限 硬件置位，软件写 1 清零
1	OVF	时钟监控上溢出标志，监控周期内计数值大于监控阈值上限 硬件置位，软件写 1 清零
0	MON_IF	时钟监控周期中断标志 (Clock monitor interrupt flag) 每个时钟监控周期完成后硬件置位，软件写 1 清零

12.4.5 时钟监控比较高阈值寄存器（CLM_CMPH）

名称	CLM_CMPHx (x=0,1)							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CMPH[23:16]							
位权限	R/W-11111111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CMPH[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CMPH[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:24	--	RFU：未实现，读为 0
23:0	CMPH	时钟监控比较值高阈值

12.4.6 时钟监控比较低阈值寄存器（CLM_CMPL）

名称	CLM_CMPLx (x=0,1)							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CMPL[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CMPL[15:8]							
位权限	R/W-0000 0000							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CMPL[7:0]							
位权限	R/W-1111 1111							

Bit	助记符	功能描述
31:24	--	RFU：未实现，读为 0
23:0	CMPL	时钟监控比较值低阈值

13 电源电压监测（SVD）

13.1 概述

电源检测电路主要用来监测外部主电源的供电情况，及时检测到外部主电源欠压或恢复的情况，并给出中断信号。电源检测电路可关断或周期使能以节省功耗。

特点：

- 监测主电源，电压低于或高于设定的阈值时产生中断
- 低压检测范围 1.8V~4.8V，15 级可编程阈值档位，档位间隔 0.214V
- 电压检测迟滞窗口 0.1V
- 可关断或间歇式工作
- 支持 1 个外部通道直接输入与内部基准电压源比较
- 外部通道支持 100mV 窗口

13.2 结构框图

下图是电源检测电路的模块框图。

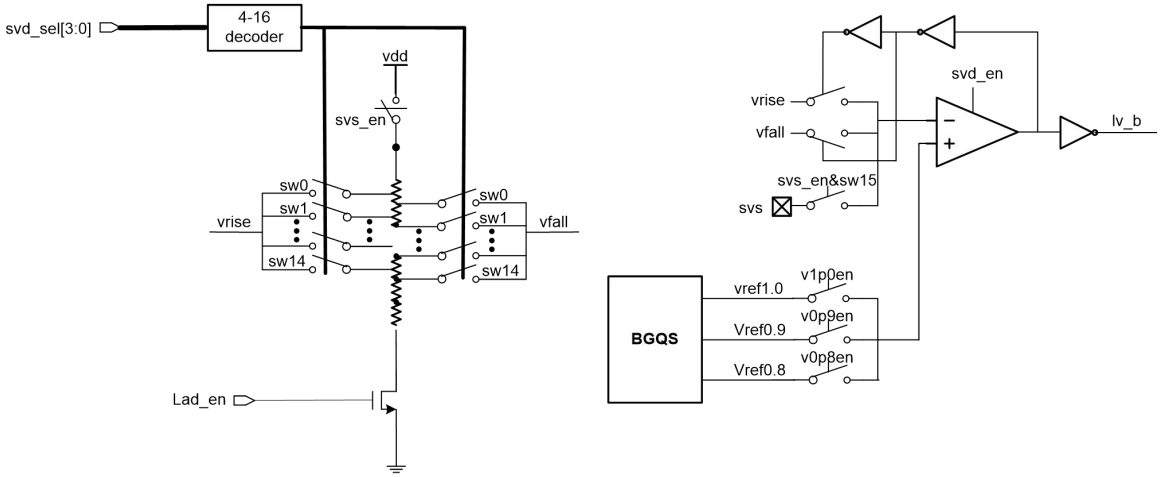


图 13-1 低压检测电路框图

SVD共有15个内部通道和1个外部通道，内部通道用于芯片电源检测，外部通道用于外部输入信号与内部基准电压比较。

SVD工作时序示意图：

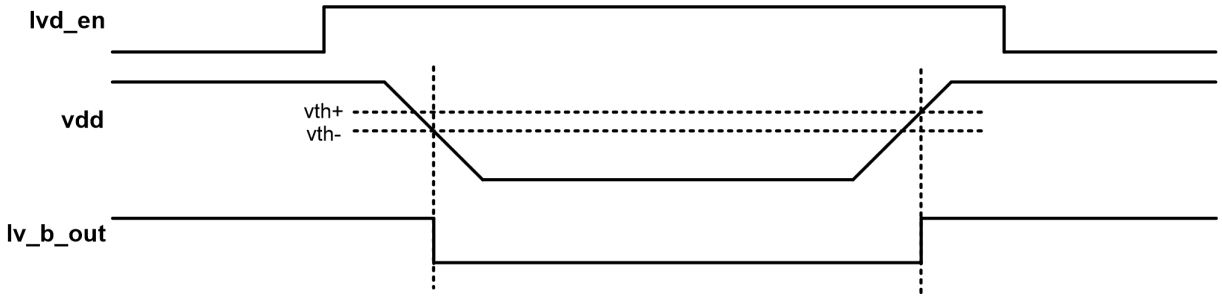


图 13-2 低压检测电路工作时序

13.3 引脚定义

SVD模块可以直接检测芯片电源（VDD），也可以通过1个SVS引脚检测外部电压信号。

检测SVS输入时，需要将GPIO的FCR寄存器配置为11（analog function）。

13.4 功能描述

电源检测电路可以用来检测主电源电压及外部电压。电源电压通过分压电阻产生15级检测电平，检测范围1.8V~4.8V，每级相差0.214V；另外还支持2路外部输入检测通道。VDD分压通过多路选择器送入比较器，与内部参考电压相比较，根据低压报警阈值设置，若待检测电平低于参考电压，引起输出电压跳变，会产生欠压中断，通知MCU及时处理该事件；而当VDD恢复至阈值以上（有大约0.1V迟滞窗口），则会产生欠压恢复中断。

电源检测电路可由软件配置使能或禁止工作。为节省功耗，使能时又可分为常使能和间歇工作两种模式。间歇工作时，可通过设置节DSEF寄存器设置开启时间间隔。

常使能条件下SVD从欠压到过压有0.1V回滞窗口，而间歇使能情况下没有回滞窗口；对于内部通道，可以通过软件配合，即欠压中断后人为设置一个较高的过压阈值，来解决窗口问题。而对于SVS通道，则需要特殊设计，由数字电路锁存上一次间歇窗口的判决结果，作为本次间歇窗口中的阈值选择依据，从而实现SVS的下降阈值和上升阈值选择。相应的，BG需要输出1.0V、0.9V、0.8V三个基准电压。

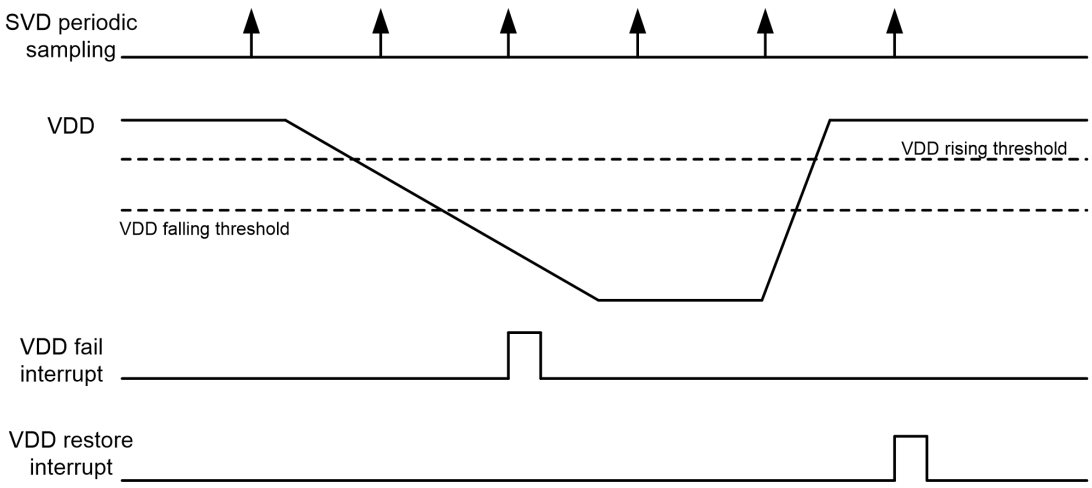


图 13-3 电源检测电路间歇工作模式

间歇工作时，当软件使能SVD的间隙使能后，SVD并不一定会立刻工作，而是要等待下一个开启窗口到来。而常使能情况下，软件开启SVD后经过一到两个LSCLK时钟同步周期后，SVD就会开始工作。SVD开启后到输出稳定建立大约需要100us时间，软件读取SVD输出时需要注意。

如果芯片进入休眠模式后关闭了所有时钟，又希望使用SVD，则需要在休眠前将SVD设置为常使能，并且关闭数字滤波功能。

工作模式说明：

- 在常使能/内部通道模式下，检测阈值有窗口，下降阈值和上升阈值窗口为0.1V，不使能到使能时检测下降阈值。
- 在间歇使能/内部通道模式下，在每次间歇使能启动时（即不使能到使能时）检测下降阈值，因此没有阈值窗口，需要软件配合，即在前次间歇使能检测到欠压时，软件将阈值档位调高一档；在前次间歇使能检测到非欠压时，软件将阈值档位恢复。
- 在常使能/外部通道模式下，输入的基准电压为三档位输入，分别为1.0V、0.9V、0.8V，检测阈值没有窗口，需要软件配合，即在检测到欠压时，软件将阈值档位调高一档；在检测到非欠压时，软件将档位恢复。
- 在间歇使能/外部通道模式下，输入的基准电压为三档位输入，分别为1.0V、0.9V、0.8V，检测阈值没有窗口，需要软件配合，即在前次间歇使能检测到欠压时，软件将阈值档位调高一档；在前次间歇使能检测到非欠压时，软件将档位恢复。

13.5 间歇使能模式

在休眠模式下可以通过间歇使能来降低SVD的平均功耗。在DeepSleep模式下，AVREF默认关闭，此时间歇启动SVD的同时还要启动AVREF，数字电路需要先等待AVREF建立，才能采样SVD输出。AVREF启动时间小于5us，SVD比较器建立时间不超过100us；在等待足够长时间后，数字电路才锁存SVD输出信号。SVD开启窗口中，比较器加AVREF总功耗小于5uA。如果开启间隔设置为1s，则平均电流小于5nA。

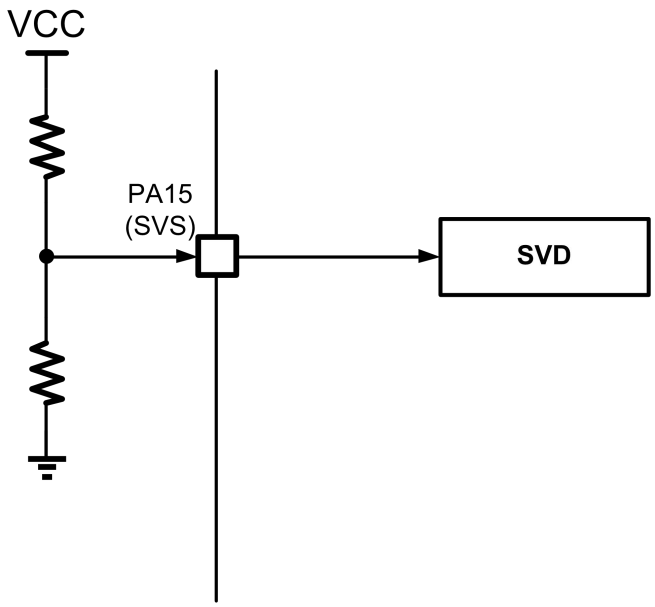
通过间歇使能模式，芯片可以保持对VDD电源供电情况的监视，同时几乎不会带来额外的休眠功耗增加。

13.6 外部电源检测

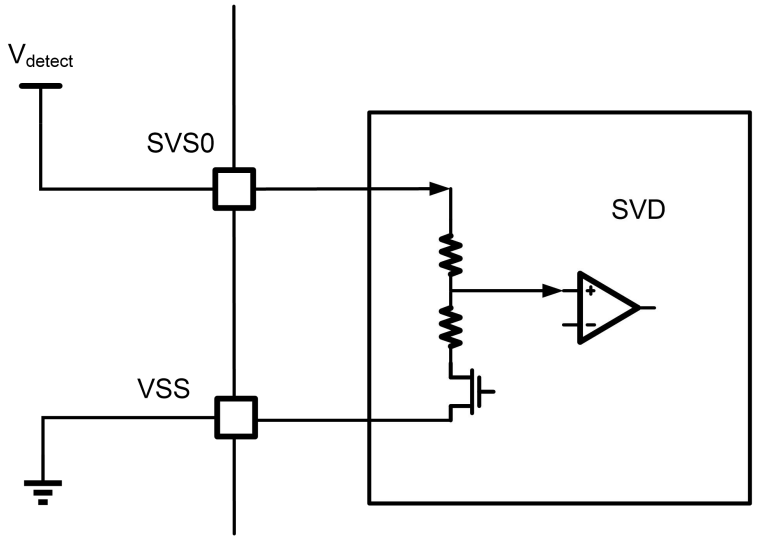
SVD除了可以检测芯片电源，也可以对外部电压信号进行掉电或上电检测。

外部电源检测通过SVS引脚实现，SVS的输入可以采用外部电阻分压或内部电阻分压后，再输入到比较器进行检测。

下图为外部电阻分压的外部电源检测：



下图为内部电阻分压的外部信号检测：



寄存器配置方法如下表：

SVSEN	SVDLVL	说明
0	X	外部电源检测通道关闭，仅检测内部电源电压
1	1111	外部电压输入不做内部分压，直接输入到比较器与内部基准电压比较
	0000~1110	外部电压输入先经过内部电阻分压，然后再输入到比较器与内部基准电压比较 分压后的档位参见后续章节描述

13.7 电源检测阈值

通过SVSEN和SVDLVL寄存器可以选择电压检测对象和检测阈值。

内部电源检测：SVSxEN = 0，{VREF1P0EN, VREF0P9EN, VREF0P8EN} = 100，比较基准1.0V

SVDLVL	上升阈值（V）	下降阈值（V）
0000	1.800	1.900
0001	2.014	2.114
0010	2.229	2.329
0011	2.443	2.543
0100	2.657	2.757
0101	2.871	2.971
0110	3.086	3.186
0111	3.300	3.400
1000	3.514	3.614
1001	3.729	3.829
1010	3.943	4.043
1011	4.157	4.257
1100	4.371	4.471
1101	4.586	4.686

1110	4.800	4.900
1111	N/A	N/A

内部电源检测：SVSxEN = 0，{ VREF1P0EN, VREF0P9EN, VREF0P8EN } = 010，比较基准0.9V

SVDLVL	上升阈值（V）	下降阈值（V）
0000	1.650	1.742
0001	1.846	1.938
0010	2.043	2.135
0011	2.239	2.331
0100	2.436	2.527
0101	2.632	2.723
0110	2.829	2.921
0111	3.025	3.117
1000	3.221	3.313
1001	3.418	3.510
1010	3.614	3.706
1011	3.811	3.902
1100	4.007	4.098
1101	4.204	4.296
1110	4.400	4.492
1111	N/A	N/A

内部电源检测：SVSxEN = 0，{ VREF1P0EN, VREF0P9EN, VREF0P8EN } = 001，比较基准0.8V

SVDLVL	上升阈值（V）	下降阈值（V）
0000	1.500	1.583
0001	1.678	1.762
0010	1.858	1.941
0011	2.036	2.119
0100	2.214	2.298
0101	2.393	2.476
0110	2.572	2.655
0111	2.750	2.833
1000	2.928	3.012
1001	3.108	3.191
1010	3.286	3.369
1011	3.464	3.548
1100	3.643	3.726
1101	3.822	3.905
1110	4.000	4.083
1111	N/A	N/A

外部电压检测：SVSxEN =1，{ VREF1P0EN, VREF0P9EN, VREF0P8EN } = 100，比较基准1.0V

SVDLVL	上升阈值（V）	下降阈值（V）
0000	1.800	1.900

0001	2.014	2.114
0010	2.229	2.329
0011	2.443	2.543
0100	2.657	2.757
0101	2.871	2.971
0110	3.086	3.186
0111	3.300	3.400
1000	3.514	3.614
1001	3.729	3.829
1010	3.943	4.043
1011	4.157	4.257
1100	4.371	4.471
1101	4.586	4.686
1110	4.800	4.900
1111	1.2	1.2

外部电压检测：SVSxEN =1，{ VREF1P0EN, VREF0P9EN, VREF0P8EN } = 010，比较基准0.9V

SVDLVL	上升阈值（V）	下降阈值（V）
0000	1.650	1.742
0001	1.846	1.938
0010	2.043	2.135
0011	2.239	2.331
0100	2.436	2.527
0101	2.632	2.723
0110	2.829	2.921
0111	3.025	3.117
1000	3.221	3.313
1001	3.418	3.510
1010	3.614	3.706
1011	3.811	3.902
1100	4.007	4.098
1101	4.204	4.296
1110	4.400	4.492
1111	1.1	1.1

外部电压检测：SVSxEN =1，{VREF1P2EN, VREF1P1EN, VREF1P0EN} = 001，比较基准0.8V

SVDLVL	上升阈值（V）	下降阈值（V）
0000	1.500	1.583
0001	1.678	1.762
0010	1.858	1.941
0011	2.036	2.119
0100	2.214	2.298
0101	2.393	2.476
0110	2.572	2.655
0111	2.750	2.833

1000	2.928	3.012
1001	3.108	3.191
1010	3.286	3.369
1011	3.464	3.548
1100	3.643	3.726
1101	3.822	3.905
1110	4.000	4.083
1111	1.0	1.0

13.8 寄存器

模块偏移地址：0x4001_2800

offset 地址	名称	符号
0x00	SVD 配置寄存器 （SVD Config Register）	SVD_CFGR
0x04	SVD 控制寄存器 （SVD Control Register）	SVD_CR
0x08	SVD 中断使能寄存器 （SVD Interrupt Enable Register）	SVD_IER
0x0C	SVD 状态和标志寄存器 （SVD Interrupt Status Register）	SVD_ISR
0x10	SVD 参考电压选择寄存器 （SVD reference Voltage Select Register）	SVD_VSR

13.8.1 SVD 配置寄存器（SVD_CFGR）

名称	SVD_CFGR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LVL				DFEN	MOD	ITVL	
位权限	R/W-0000				R/W-1	R/W-0	R/W-00	

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:4	LVL	SVD 报警阈值设置，档位定义参见 13.7 电源检测阈值 (SVD threshold level)
3	DFEN	数字滤波使能（SVDMODE=1 时必须置 1）(Digital Filter Enable) 1：启动 SVD 输出的数字滤波 0：关闭 SVD 输出的数字滤波
2	MOD	SVD 工作模式选择，配置模式后还要置位 SVDEN 才会启动 SVD (SVD Mode) 1：间歇使能模式 0：常使能模式 注意：间歇使能模式下必须开启数字滤波
1:0	ITVL	SVD Interval，SVD 间歇使能间隔 (SVD interval enable period) 00：62.5ms 01：256ms 10：1s 11：4s

13.8.2 SVD 控制寄存器（SVD_CR）

名称	SVD_CR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							TE
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					-	SVS0EN	EN
位权限	U-0					U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:9	--	RFU：未实现，读为 0
8	TE	RFU
7:3	--	RFU：未实现，读为 0
2	--	RFU：未实现，读为 0
1	SVS0EN	SVS0 外部电源检测通道控制信号 (SVS0 external monitor channel enable) 0：SVS0 通道关闭 1：SVS0 通道使能 当 SVS0EN=1 时，根据 SVDLVL 寄存器可以设置 SVS 输入后是否经过内部电阻分压；如果 SVDLVL=1111，则 SVS 输入不做分压，如果 SVDLVL != 1111，则 SVS 输入经过内部电阻分压。
0	EN	SVD 使能 (SVD enable) 1：启动 SVD 0：关闭 SVD

13.8.3 SVD 中断使能寄存器（SVD_IER）

名称	SVD_IER							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	PFIE	PRIE
位权限	U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	PFIE	电源跌落中断使能寄存器（Power Fall interrupt enable） 1：允许电源跌落中断 0：禁止中断
0	PRIE	电源恢复中断使能寄存器（Power Rise interrupt enable） 1：允许电源恢复中断 0：禁止中断

13.8.4 SVD 状态和标志寄存器（SVD_ISR）

名称	SVD_ISR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							SVDO
位权限	U-0							R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SVDR	-					PFF	PRF
位权限	R	U-0					R/W-0	R/W-0

Bit	助记符	功能描述
31:9	--	RFU：未实现，读为 0
8	SVDO	SVD 电源检测输出 1：电源电压高于 SVD 当前阈值 0：电源电压低于 SVD 当前阈值
7	SVDR	SVD 输出锁存信号，数字电路锁存的 SVD 状态
6:2	--	RFU：未实现，读为 0
1	PFF	电源跌落中断标志寄存器，电源电压跌落到 SVD 阈值之下时置位，软件写 1 清零
0	PRF	电源恢复中断标志寄存器，电源电压上升到 SVD 阈值之上时置位，软件写 1 清零

13.8.5 SVD 参考电压选择寄存器（SVD_VSR）

名称	SVD_VSR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit0	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					V1P0EN	V0P9EN	V0P8EN
位权限	U-0					R/W-1	R/W-0	R/W-0

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	V1P0EN	1.0V 基准输入使能信号 (1.0V reference enable) 1：使能 1.0V 基准输入 0：关闭 1.0V 基准输入
1	V0P9EN	0.9V 基准输入使能信号 (0.9V reference enable) 1：使能 0.9V 基准输入 0：关闭 0.9V 基准输入
0	V0P8EN	0.8V 基准输入使能信号 (0.8V reference enable) 1：使能 0.8V 基准输入 0：关闭 0.8V 基准输入

14 AES 硬件运算单元（AES）

14.1 功能描述

AES单元主要功能如下：

- 支持解密密钥扩展
- 支持128bit/192bit/256bit的密钥长度
- 支持ECB，CBC，CTR，GCM
- 支持DMA进行自动数据传输
- 支持GF（2^128）域下的乘法，支持GMAC

14.2 工作模式

AES有4种工作模式，通过配置MODE[1:0]寄存器设置。

模式1：用存储在AES_KEYRx寄存器中的密钥加密。

模式2：密钥扩展，把初始存储在AES_KEYRx寄存器的加密密钥覆盖成在密钥扩展完成后存储在内部寄存器的密钥计算结果。

模式3：用存储在AES_KEYRx寄存器中的解密密钥（预计算的）解密。

模式4：用存储在AES_KEYRx寄存器中的加密密钥进行密钥扩展和解密。（在CTR模式下不使用）

首先通过配置MODE[1:0]寄存器确定工作模式，MODE寄存器必须在AES使能前（EN=0时）才能够配置。KEY寄存器也应该在AES使能前配置。之后配置数据流处理模式寄存器CHMOD[1:0]，在CBC/CTR/GCM模式下还需要配置IV寄存器。

接着可以使能EN，在模式1/模式3/模式4下，AES模块等待软件往AES_DINR寄存器写入输入数据，写4次写完128bit后AES开始计算。在模式2时，使能EN后就马上进行密钥扩展运算了。

计算完成后标志CCF会置起，如果CCFIE=1，会产生一个中断信号。软件再从AES_DOUTR寄存器中读4次共128bit的结果。

AES还支持DMA模式。通过配置DMAOUTEN=1和DMAINEN=1，AES可以配合DMA连续的处理数据，无需CPU的介入。

错误标志RDERR和WRERR会在一次错误的读写操作时置起，如果ERRIE使能，还会产生相应的错

误中断。AES在产生错误后还会继续正常工作。

通过重置EN寄存器能够在任何时候复位AES模块。

14.3 AES 数据流处理模式

AES有4种数据流处理模式：ECB，CBC，CTR，GCM。

14.3.1 ECB 模式

默认的工作模式，该模式下无需使用IV寄存器，每个block单独进行加解密计算。加解密流程如图14-1和图14-2所示。

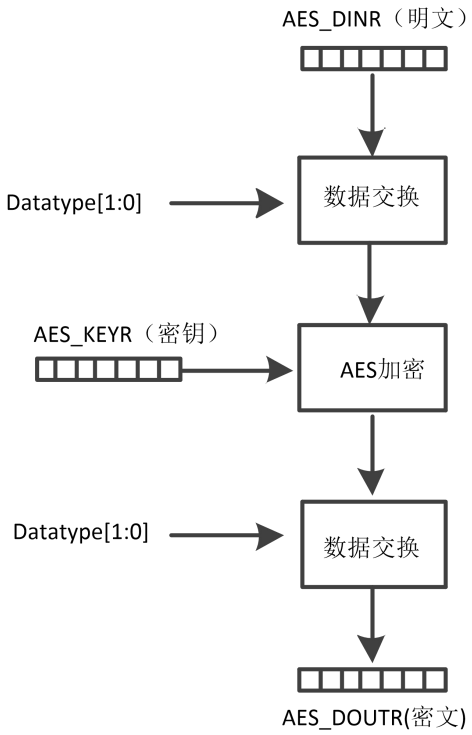


图 14-1 ECB 模式加密流程

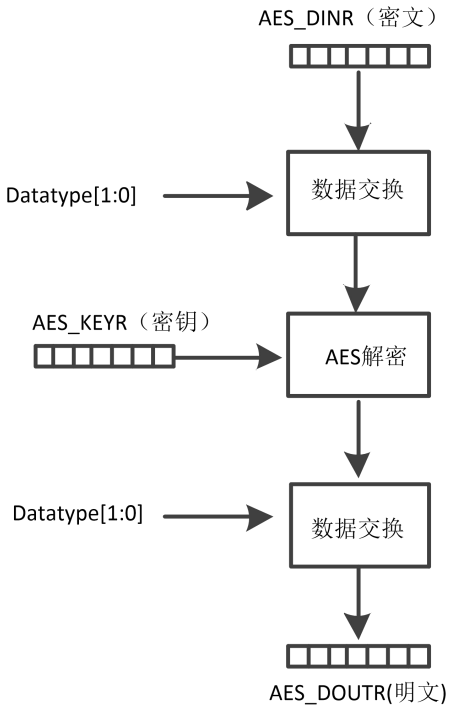


图 14-2 ECB 模式解密流程

14.3.2 CBC 模式

每个block的明文数据与前一block的加密结果异或后作为加密的数据输入。第一个block需要一个初始的IVRx寄存器值。加密时异或操作在加密前而解密时异或操作在加密后。工作流程如图14-3和图14-4所示。

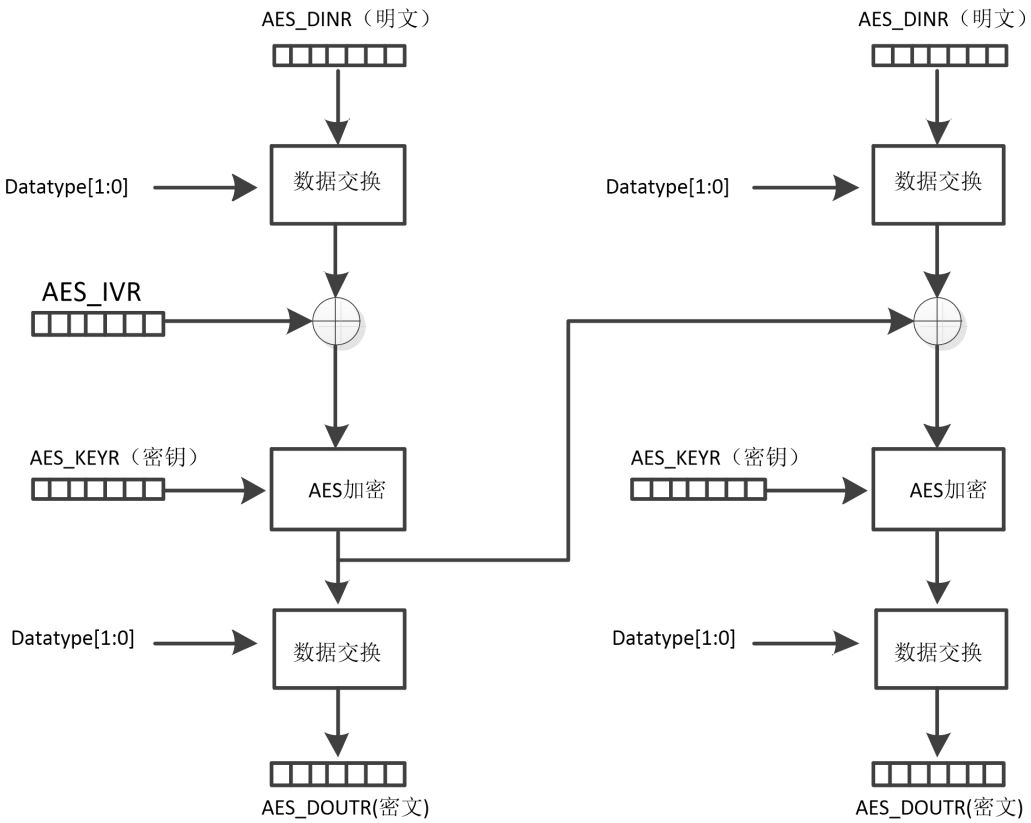


图 14-3 CBC 加密过程

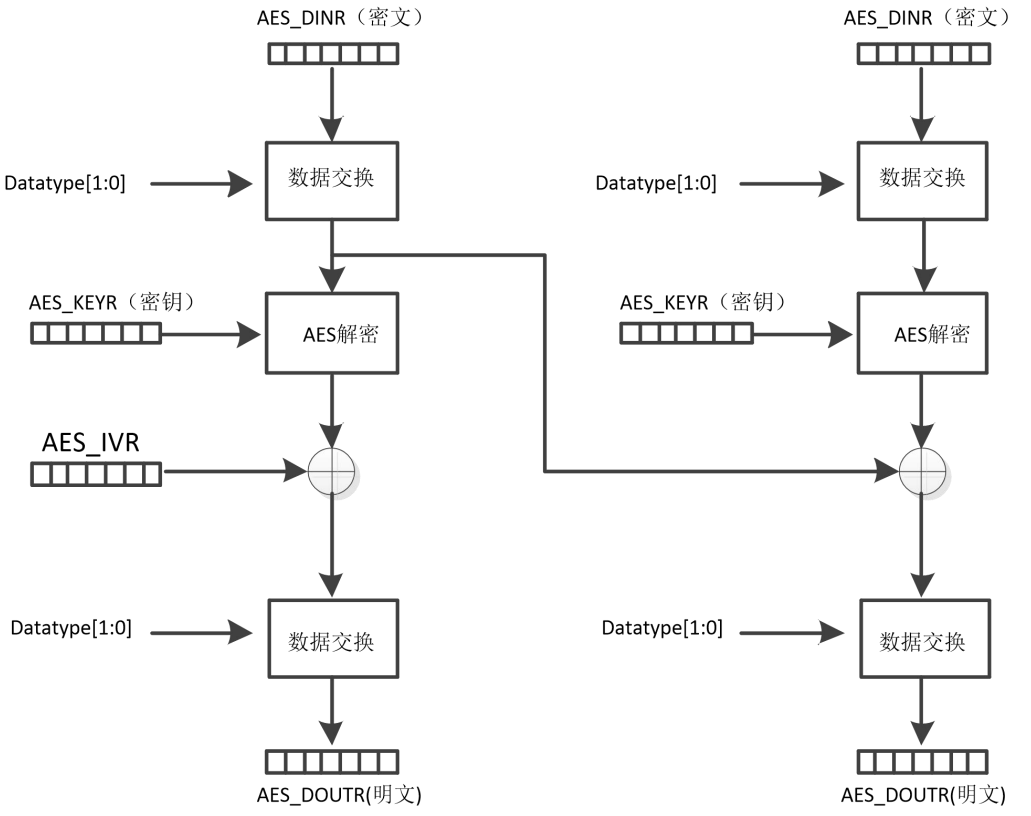


图 14-4 CBC 解密过程

注：在AES工作时读取AES_IVR寄存器的值为0x00000000

14.3.3 暂停模式

如果一个更高优先级的数据需要处理，当前的数据运算是可以暂停的。暂停的数据处理在加解密运算模式下都能够恢复。仅在CPU参与的模式下可用，DMA模式下不可用。

正确的工作流程为：数据在一个block的结果被读完后暂停。

通过对EN bit写0暂停AES。软件读AES_IVRx寄存器中的值并存储，在恢复运算时该值需要被写入AES_IVRx寄存器。

流程如图14-5所示



图 14-5 暂停模式流程

14.3.4 CTR 模式

该模式下，一个32bit的计数器和一个随机数被用作加解密模块的输入。结果与明文数据进行异或。流程如图14-6和图14-7所示。

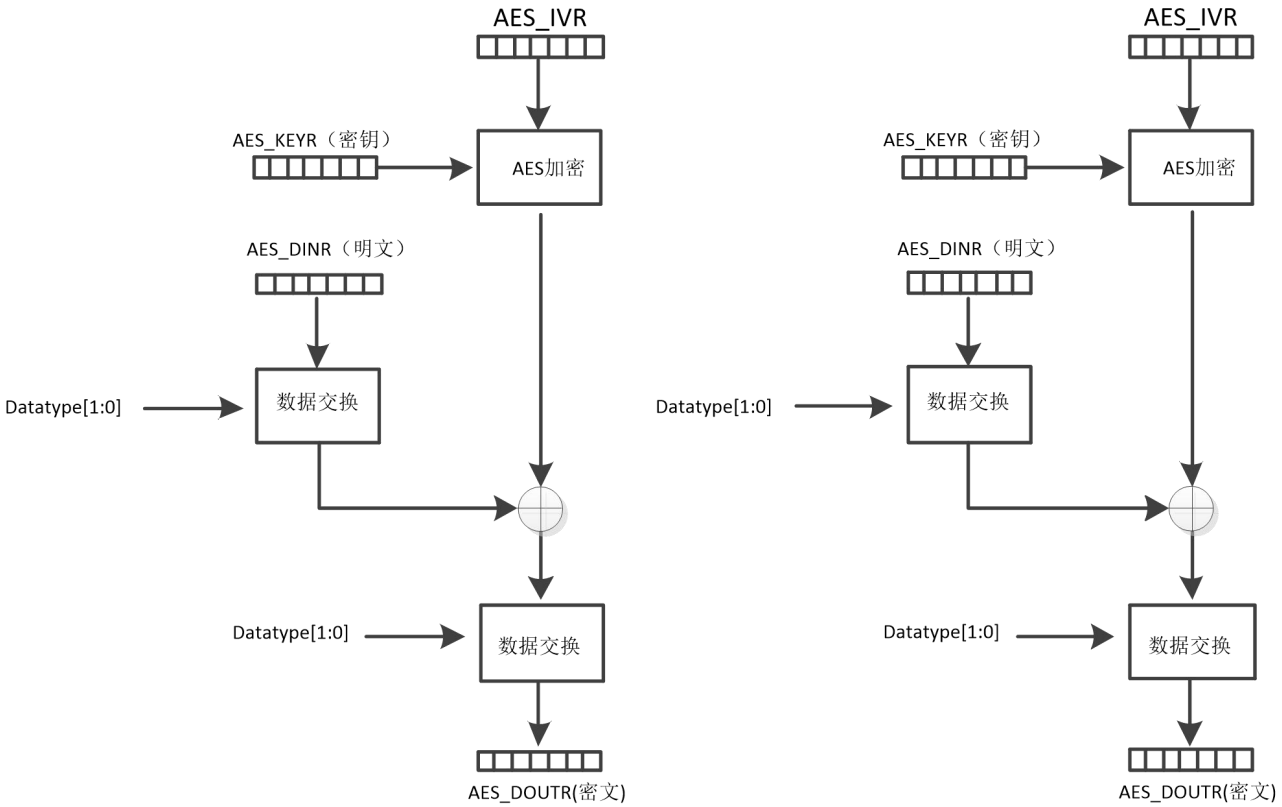


图 14-6 CTR 加密流程

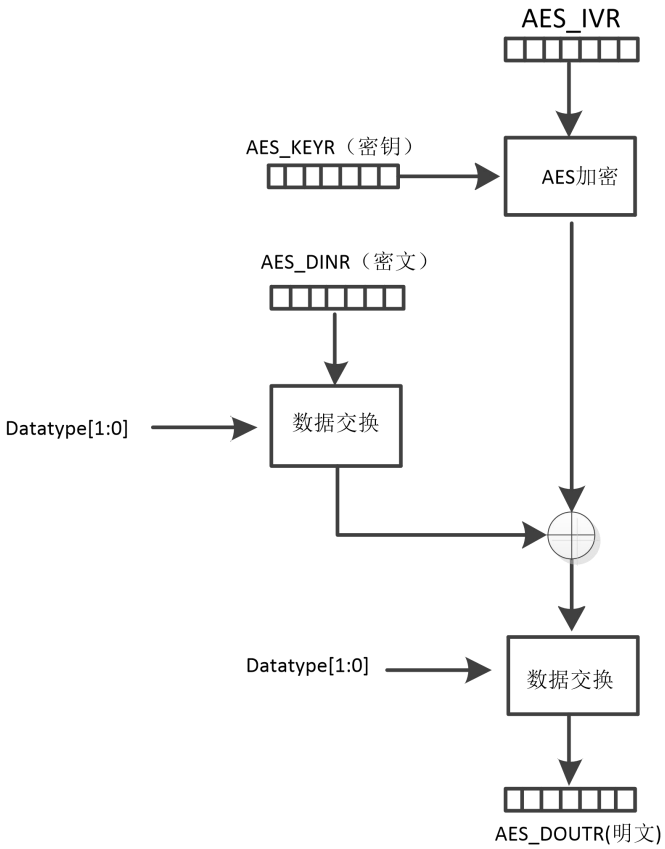


图 14-7 CTR 解密流程

随机数（nonce）和32位计数器存储在IV寄存器中，如图14-8所示



图 14-8 32 位计数器和随机数的存储方式

CTR模式下密钥扩展和解密模式没有意义。

14.3.5 CTR 模式下的暂停模式

与CBC下暂停模式类似。参考CBC下暂停模式。

14.3.6 GCM 模式

具体可以参考文档The Galois/Counter Mode of Operation (GCM)

GCM的加密按照以下公式定义：

$$
\begin{aligned}
H &= E(K, 0^{128}) \\
Y_0 &= \begin{cases} IV \| 0^{31}1 & \text{if len}(IV) = 96 \\ \text{GHASH}(H, \{\}, IV) & \text{otherwise.} \end{cases} \\
Y_i &= \text{incr}(Y_{i-1}) \text{ for } i = 1, \ldots, n \\
C_i &= P_i \oplus E(K, Y_i) \text{ for } i = 1, \ldots, n-1 \\
C_n^* &= P_n^* \oplus \text{MSB}_u(E(K, Y_n)) \\
T &= \text{MSB}_t(\text{GHASH}(H, A, C) \oplus E(K, Y_0))
\end{aligned}
$$

其中GHASH函数的定义为 $\text{GHASH}(H, A, C) = X_{m+n+1}$，其中X的定义为

$$
X_i = \begin{cases} 0 & \text{for } i = 0 \\ (X_{i-1} \oplus A_i) \cdot H & \text{for } i = 1, \ldots, m-1 \\ (X_{m-1} \oplus (A_m^* \| 0^{128-v})) \cdot H & \text{for } i = m \\ (X_{i-1} \oplus C_i) \cdot H & \text{for } i = m+1, \ldots, m+n-1 \\ (X_{m+n-1} \oplus (C_m^* \| 0^{128-u})) \cdot H & \text{for } i = m+n \\ (X_{m+n} \oplus (\text{len}(A) \| \text{len}(C))) \cdot H & \text{for } i = m+n+1. \end{cases}
$$

GCM模式的加解密流程如图14-9，图14-10所示。

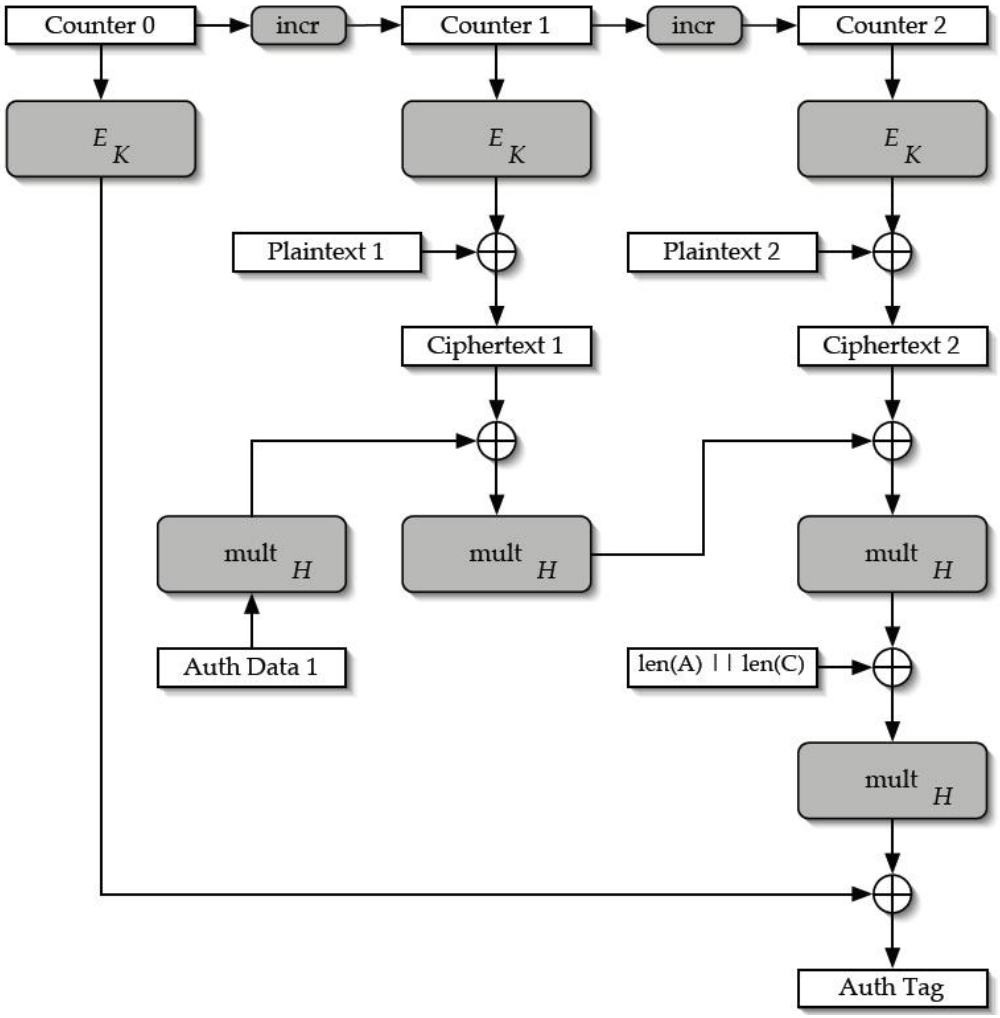


图 14-9 GCM 加密流程

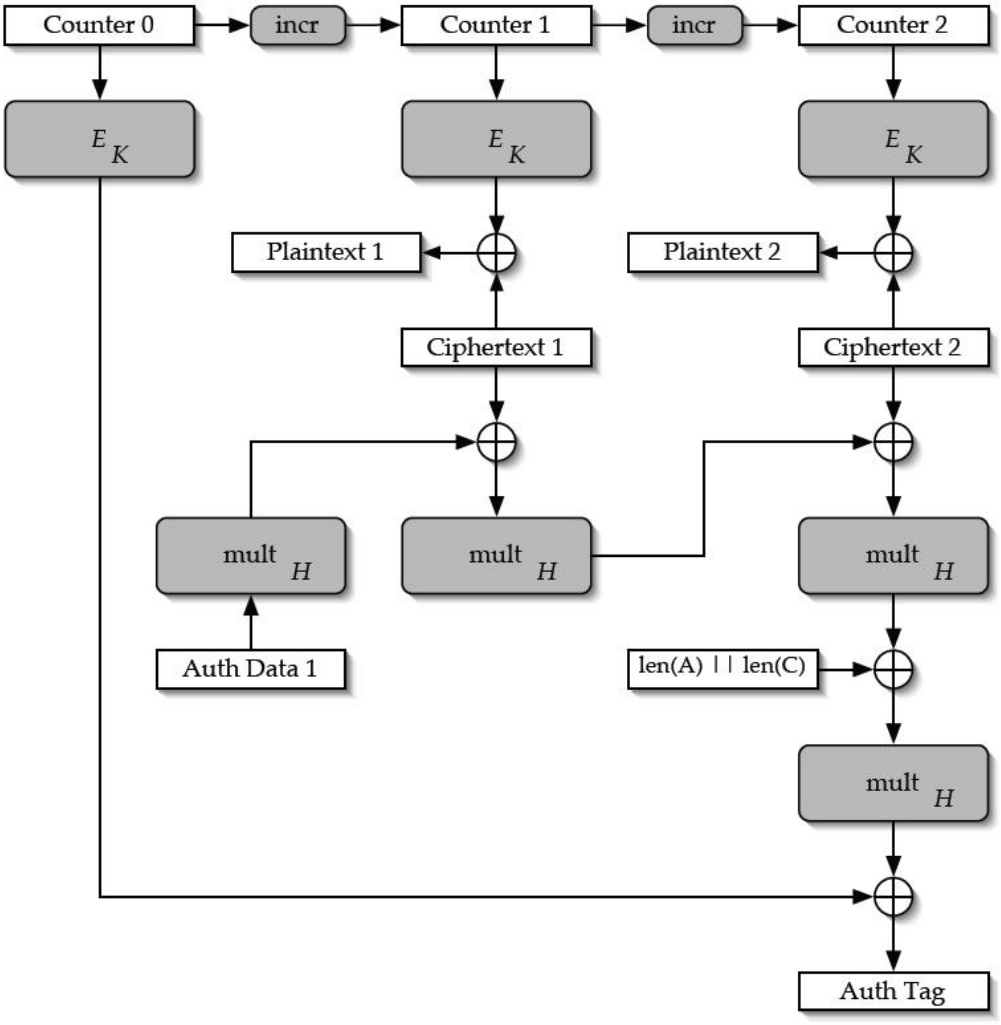


图 14-10 GCM 解密流程

图中Ek表示AES加密模块。multH模块是一个GF（2^128）域上的乘法。Incr表示计数器加一。

GCM模式由软件配合实现，硬件提供一个AES模块和multH模块供软件调度。GCM模式加解密的过程与CTR模式相同。认证过程通过软件调度multH模块实现。

14.3.7 MultH 模块

GF（2^128）上的乘法使用如下算法实现。

Algorithm 1 Multiplication in $GF(2^{128})$. Computes the value of $Z = X \cdot Y$, where X, Y and $Z \in GF(2^{128})$.

$Z \leftarrow 0, V \leftarrow X$
for $i = 0$ to 127 **do**
  if $Y_i = 1$ **then**
    $Z \leftarrow Z \oplus V$
  end if
  if $V_{127} = 0$ **then**
    $V \leftarrow \text{rightshift}(V)$
  else
    $V \leftarrow \text{rightshift}(V) \oplus R$
  end if
end for
return Z

MultH模块的输入输出寄存器复用AES的寄存器。模块框图如图14-11所示。

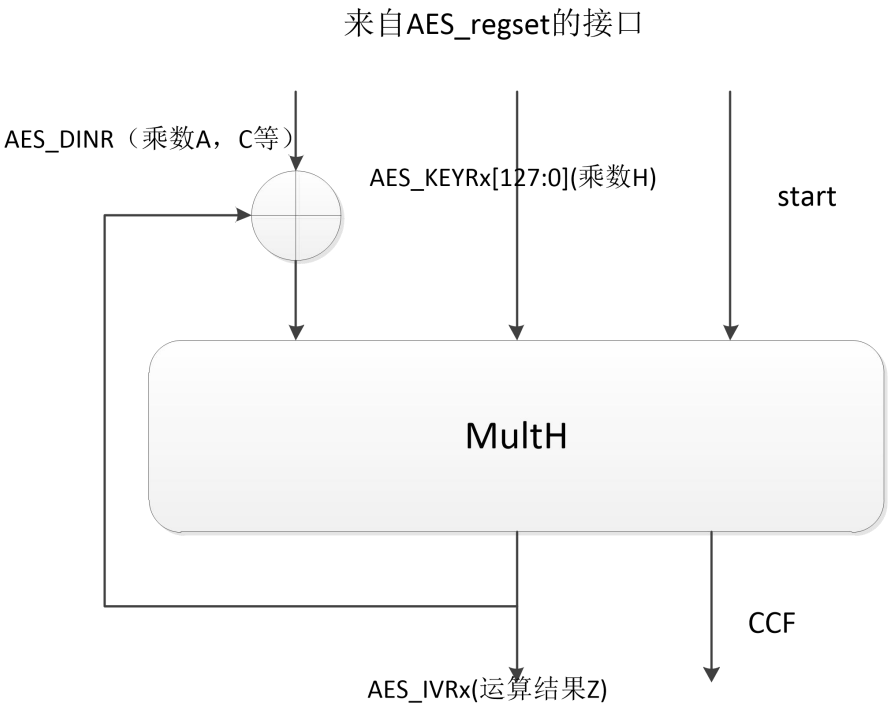


图 14-11 multH 模块框图

multH模块的输入寄存器复用AES的输入寄存器AES_DINR和AES_KEYx的低128bit。输出寄存器复用AES_IVR寄存器。使用时配置CHMOD[1:0]寄存器为MultH模式，接着配置好AES_KEYx和AES_IVR寄存器输入和输出各128bit，使能EN，向AES_DINR输入数据，等待CCF置起即计算完成。

14.3.8 推荐的 GCM 流程

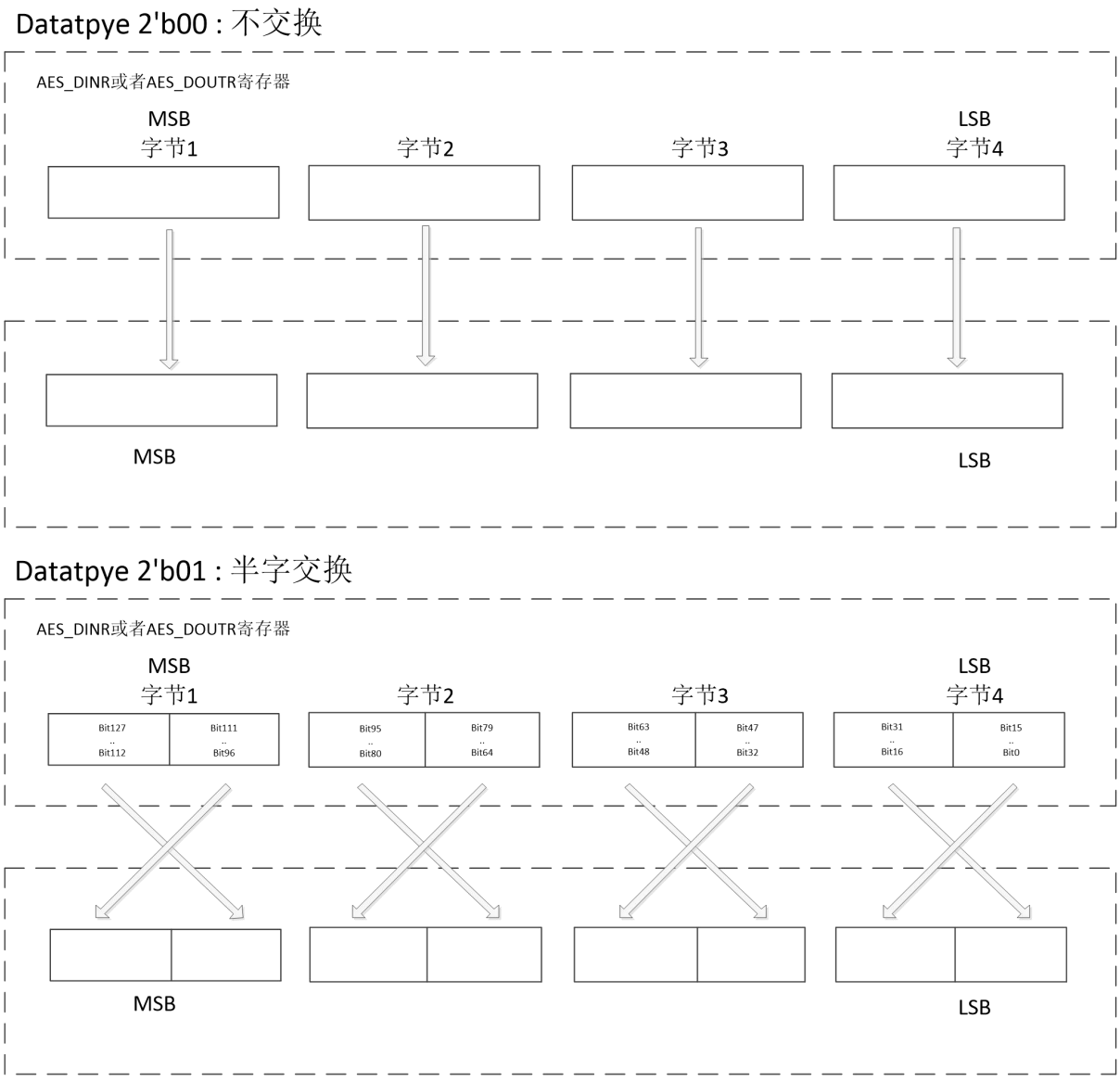
GCM模式的实现需要软硬件配合，本文档提供一种推荐的使用方法。

GCM模式的加解密过程和CTR模式相同。认证过程时仅使用MultH模块而不用AES加解密。

- 调用一次AES模块计算H。并存储。
- 调用一次AES模块计算E（K，Y0），并存储。
- 使用CTR模式开始连续数据的AES加解密操作。IV寄存器初值为Y1
- 使用multH模块连续计算GHASH结果
- 最终GHASH的结果异或上E（K，Y0）即可计算得到tag的值。

14.4 数据类型

AES一次读写32bit数据，每32bit可以根据DATATYPE[1:0]寄存器的设置按照不同的方式交换数据的顺序。如图14-12所示。



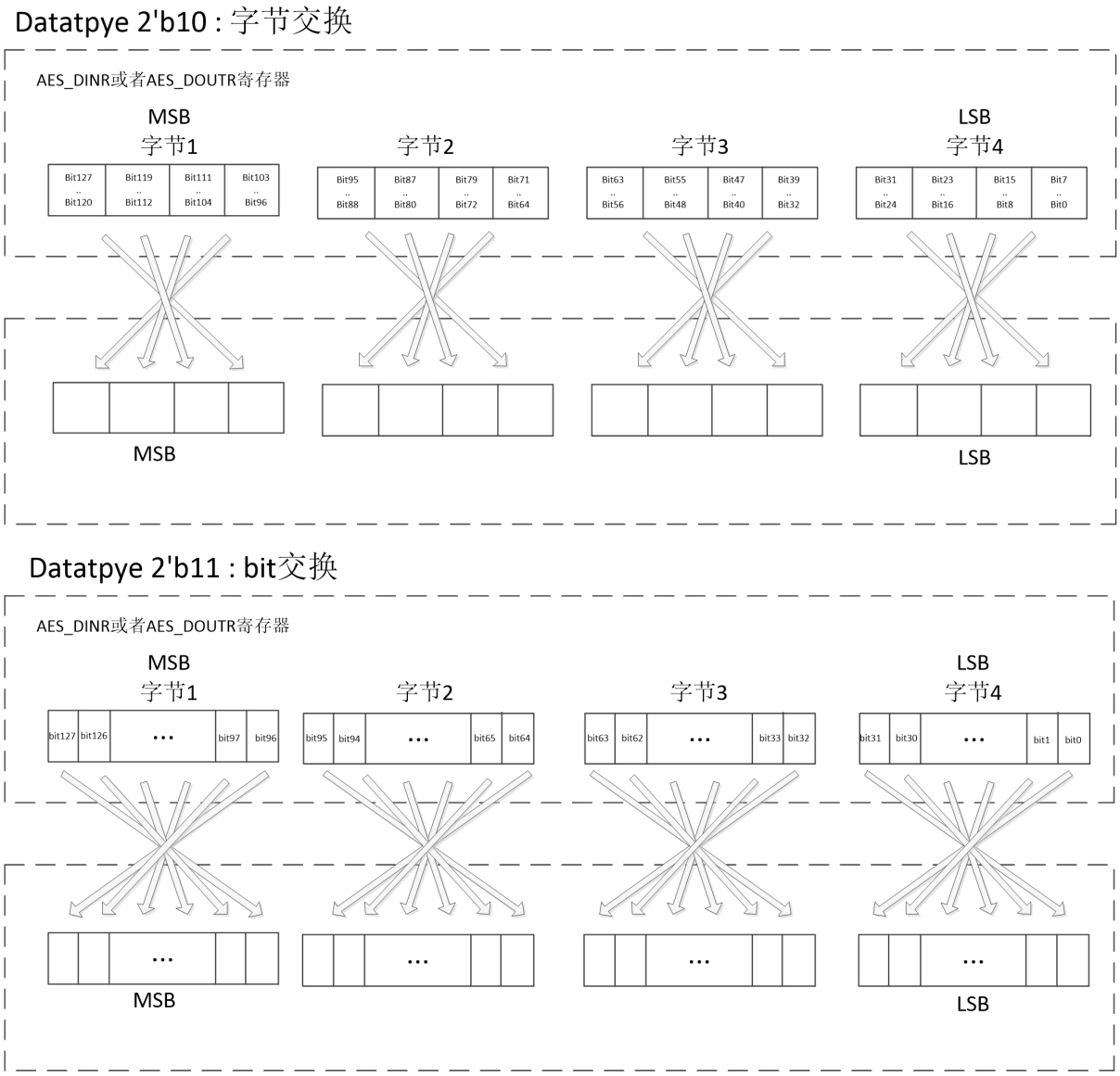


图 14-12 根据数据类型存储数据的示意图

14.5 工作流程

14.5.1 模式 1：加密

- 复位EN 重置AES模块
- 设置模式寄存器mode[1:0]=00，设置流数据处理模式寄存器CHMOD[1:0]
- 写AES_KEYRx寄存器，CTR和CBC模式下写AES_IVRx寄存器
- 写EN=1，使能AES
- 写AES_DINR 寄存器4次
- 等待CCF标志置起

- 从AES_DOUTR分4次读出加密结果
- 对于同一个key，重复步骤5,6,7对接下来的128bit block进行加密

步骤5-7如图所示。

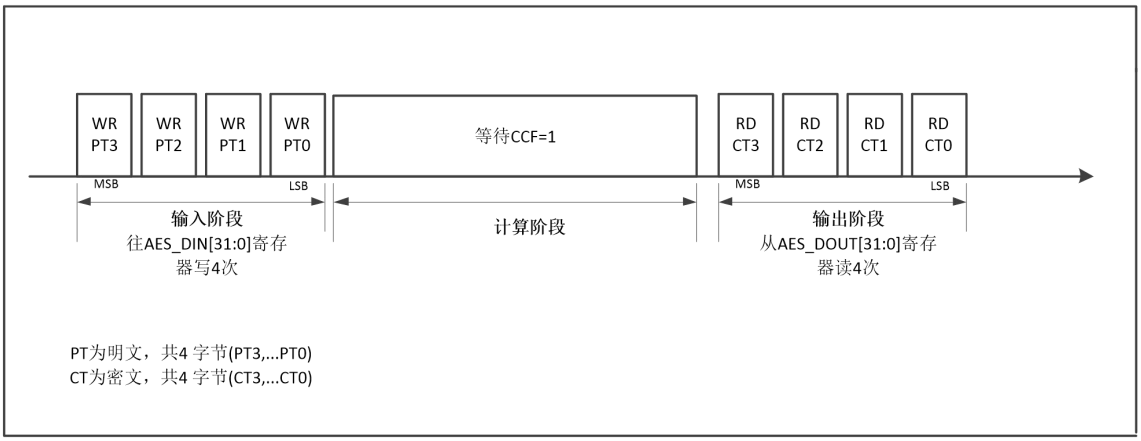


图 14-13 模式 1：加密流程

14.5.2 模式 2：密钥扩展

- 复位EN 重置AES模块
- 设置模式寄存器mode[1:0]=01，CHMOD[1:0]寄存器的值不关心。
- 写AES_KEYRx寄存器。
- 写EN=1，使能AES
- 等待CCF标志置起
- 清除CCF标志，扩展完的key自动写回AES_KEYRx寄存器。如果需要的话可以读取AES_KEYRx寄存器获取结果。想要重新计算扩展密钥，重复步骤3,4,5,6。

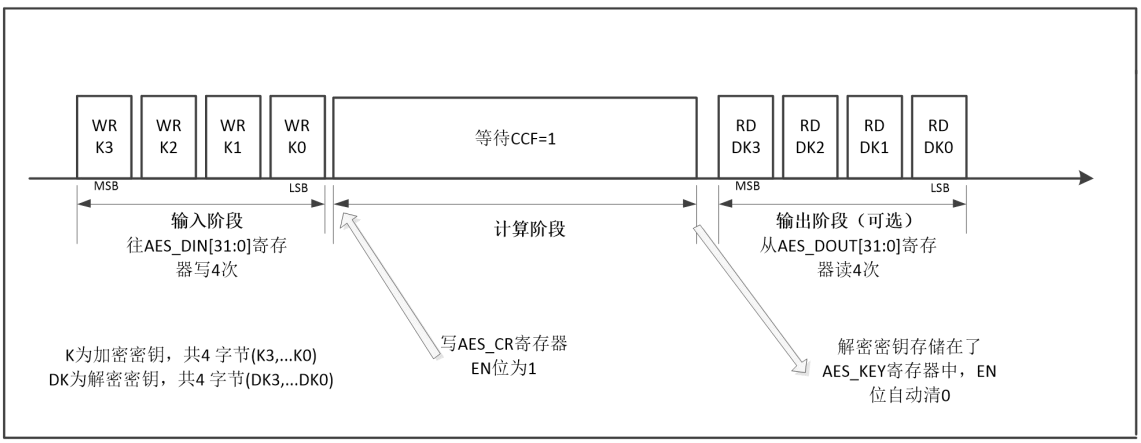


图 14-14 模式 2 示意图

14.5.3 模式 3：解密

- 复位EN 重置AES模块
- 设置模式寄存器mode[1:0]=10，设置流数据处理模式寄存器CHMOD[1:0]
- 写AES_KEYRx寄存器（如果已经通过模式2计算得到了扩展密钥则可跳过这个步骤），CTR和CBC模式下写AES_IVRx寄存器。
- 写EN=1，使能AES
- 写AES_DINR 寄存器4次
- 等待CCF标志置起
- 从AES_DOUTR分4次读出解密结果
- 对于同一个key，重复步骤5,6,7对接下来的128bit block进行解密

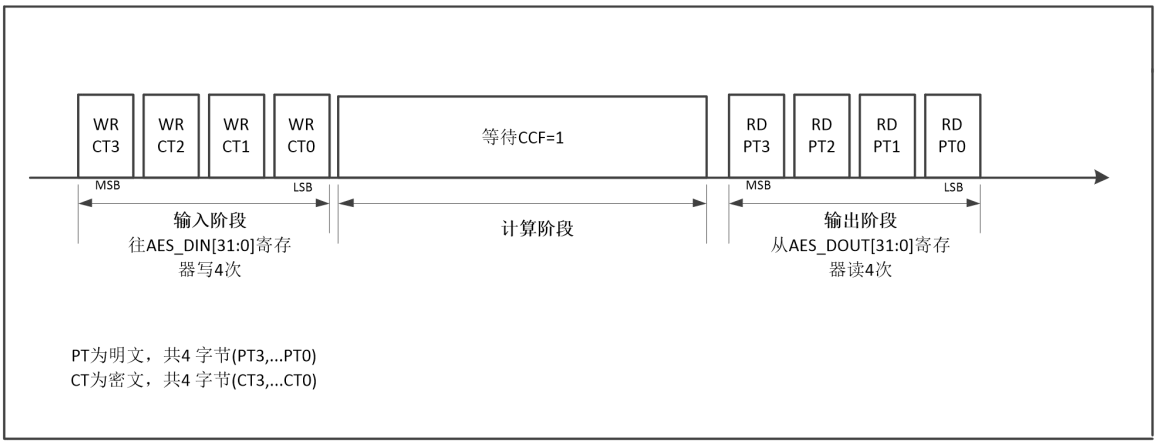


图 14-15 模式 3 示意图

14.5.4 模式 4：密钥扩展+解密

- 复位EN 重置AES模块
- 设置模式寄存器mode[1:0]=11，设置流数据处理模式寄存器CHMOD[1:0]。该模式在CTR模式下被禁止使用。如果设置mode[1:0]=11，CHMOD[1:0]=10，将强制进入CTR解密模式。
- 写AES_KEYRx寄存器，CBC模式下写AES_IVRx寄存器。
- 写EN=1，使能AES
- 写AES_DINR 寄存器4次
- 等待CCF标志置起
- 从AES_DOUTR分4次读出解密结果
- 对于同一个key，重复步骤5,6,7对接下来的128bit block进行解密

注意：该模式下AES_KEYRx寄存器内存储的一直是加密密钥，扩展密钥每次都会在内部被重新计

算而不会被存储到AES_KEYRx寄存器中。

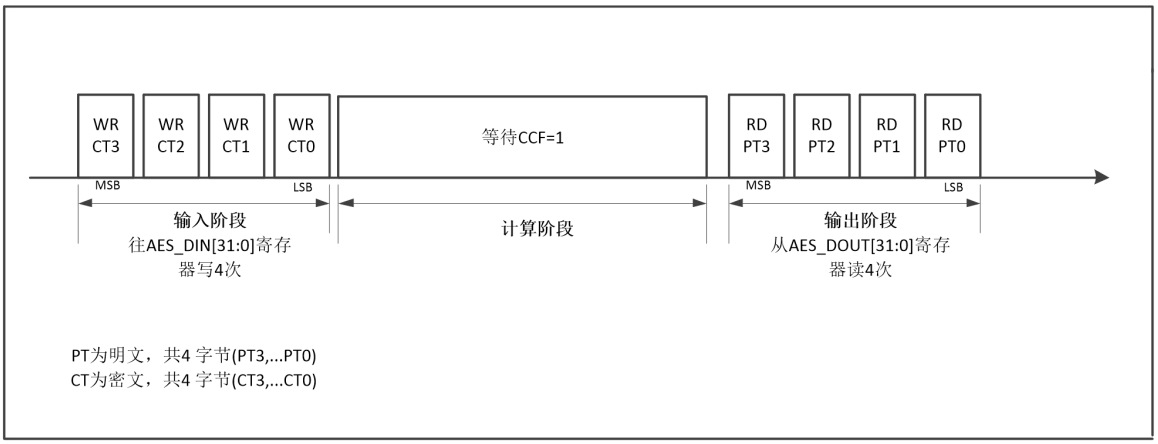


图 14-16 模式 4 示意图

14.5.5 使用 MultH 模块

- 复位EN 重置AES模块。
- 设置流数据处理模式寄存器CHMOD[1:0]=11。该模式下mode[1:0]寄存器的值不能够是01配置在模式2：密钥扩展下。同时配置mode[1:0]=01和CHMOD[1:0]=11会由于mode寄存器优先值更高而进行密钥扩展操作。
- 写AESHx寄存器，若为第一轮计算，则初始值为0x00000000。
- 写EN=1，使能multH模块。
- 写AES_DINR 寄存器4次。MultH模块会把上一次的计算结果异或上AES_DINR寄存器输入的值做为multH模块的一个乘数。所以把上一轮的计算结果赋为0x00000000，即实现了直接把AES_DINR寄存器输入的值做为multH模块的一个乘数的功能。
- 等待CCF标志置起
- 从AES_IVR寄存器中读出计算结果，可以通过IVRSWAP寄存器控制读取的数据是否交换顺序后再放到总线上
- 对于同一个H，重复步骤5,6进行连续计算。即可实现了一个GMAC的功能。

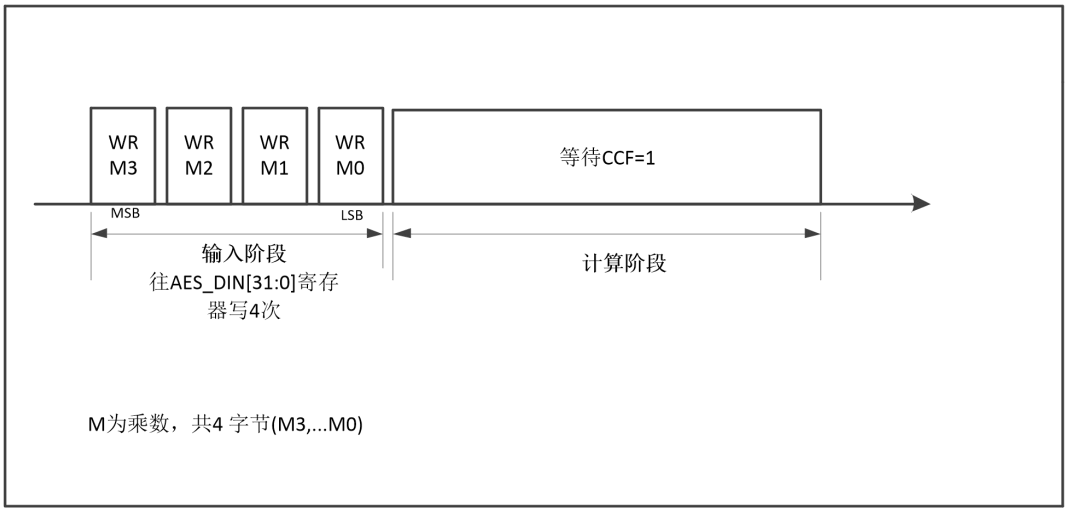


图 14-17 multH 模块使用流程示意图

14.6 DMA 接口

- 一个输入的请求通道：当DMAINEN为1时，每当AES在需要输入数据写入AES_DINR寄存器的时候发起一个DMA的请求。
- 一个输入的请求通道：当DMAOUTEN为1时，每当AES在需要从AES_DOUTR寄存器输出数据的时候发起一个DMA的请求。

每个阶段产生4次请求，在AES模块被关闭前对DMA的请求会一直产生。AES计算完128比特后就自动取新数据进行下次计算。

注意：DMA模式下DMAOUTEN=1时，CCF标志可能为高。

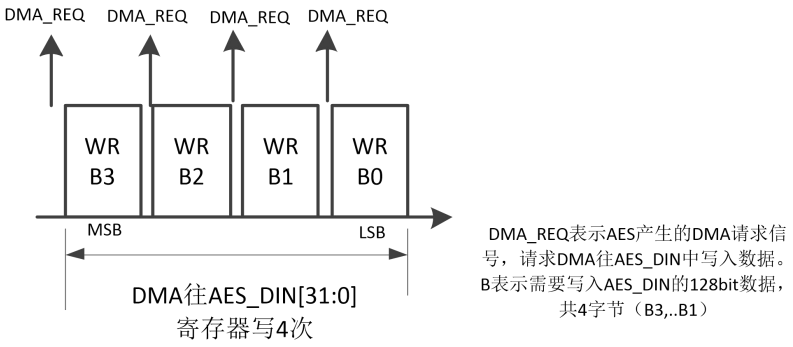


图 14-18 输入时 DMA 请求和数据传输示意图

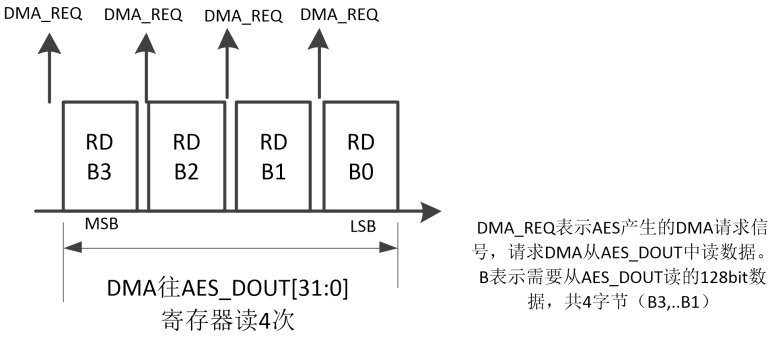


图 14-19 输出时 DMA 请求和数据传输示意图

14.6.1 MultH 模块与 DMA 间接口

MultH计算也可以通过DMA计算。当DMAINEN为1及CHMOD[1:0]=11时，每当AES在需要输入数据写入AES_DINR寄存器的时候发起一个DMA的请求。该模式下配置DMAOUTEN=1无效AES不会产生DMA请求。

14.7 错误标志

在计算和输入阶段发生一个读操作，置起RDERR。

在计算和输出阶段发生一个写操作，置起WRERR。

产生错误后AES模块不会被硬件自动停止，会像正常一样继续运算。

14.8 寄存器

模块基地址：0x4001_3800

offset 地址	名称	符号
0x00	AES 控制寄存器 （AES Control Register）	AES_CR
0x04	AES 中断使能寄存器 （AES Interrupt Enable Register）	AES_IER
0x08	AES 中断标志寄存器 （AES Interrupt Status Register）	AES_ISR
0x0C	AES 数据输入寄存器 （AES Data Input Register）	AES_DIR
0x10	AES 数据输出寄存器 （AES Data Output Register）	AES_DOR
0x14	AES 秘钥寄存器 0 （AES Key Register 0）	AES_KEY0
0x18	AES 秘钥寄存器 1 （AES Key Register 1）	AES_KEY1
0x1C	AES 秘钥寄存器 2 （AES Key Register 2）	AES_KEY2
0x20	AES 秘钥寄存器 3 （AES Key Register 3）	AES_KEY3
0x24	AES 秘钥寄存器 4 （AES Key Register 4）	AES_KEY4
0x28	AES 秘钥寄存器 5 （AES Key Register 5）	AES_KEY5
0x2C	AES 秘钥寄存器 6 （AES Key Register 6）	AES_KEY6
0x30	AES 秘钥寄存器 7 （AES Key Register 7）	AES_KEY7
0x34	AES 初始向量寄存器 0 （AES Initial Vector Register 0）	AES_IVR0
0x38	AES 初始向量寄存器 1 （AES Initial Vector Register 1）	AES_IVR1
0x3C	AES 初始向量寄存器 2 （AES Initial Vector Register 2）	AES_IVR2
0x40	AES 初始向量寄存器 3 （AES Initial Vector Register 3）	AES_IVR3
0x44	AES MultH 参数寄存器 0 （AES MultH parameter Register 0）	AES_H0
0x48	AES MultH 参数寄存器 1 （AES MultH parameter Register 1）	AES_H1
0x4C	AES MultH 参数寄存器 2 （AES MultH parameter Register 2）	AES_H2
0x50	AES MultH 参数寄存器 3 （AES MultH parameter Register 3）	AES_H3

14.8.1 AES 控制寄存器（AES_CR）

名称	AES_CR

offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	KEYLEN		DMAOEN	DMAIEN	IVRSWAP		-
位权限	U-0	R/W-00		R/W-0	R/W-0	R/W-00		U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	CHMOD		MODE		DATATYP		EN
位权限	U-0	R/W-00		R/W-00		R/W-00		R/W-0

Bit	助记符	功能描述
31:15	--	RFU：未实现，读为 0
14:13	KEYLEN	AES 加密密钥长度，AESEN=1 时不可修改。 (Key Length) 00：128bit 01：192bit 10：256bit 11：保留
12	DMAOEN	DMA 数据自动读出使能 (DMA output enable) 0：不开启 1：开启 该位置位后在模式 1，模式 3 和模式 4 下 AES 模块会自动产生 AES->RAM 的传输请求。模式 2 下不会产生。
11	DMAIEN	开启 DMA 数据自动写入使能 (DMA input enable) 0：不开启 1：开启 该位设置为 1 后在模式 1，模式 3 和模式 4 以及 MultH 模式下 AES 模块会自动产生 RAM->AES 的传输请求。模式 2 下不会产生。
10:9	IVRSWAP	IVR 寄存器读出交换功能 (IVR register read-out swapping) 此寄存器仅针对 IVR 寄存器读取，读出的数据在放到系统总线上之前先经过顺序交换处理 00：32bit 数据不交换 01：16bit 数据半字交换 10：8bit 数据字节交换 11：1bit 数据比特交换
8:7	--	RFU：未实现，读为 0
6:5	CHMOD	AES 数据流处理模式，AESEN=1 时不可修改。 (Cipher Mode) 00：ECB 01：CBC 10：CTR 11：使用 MultH 模块
4:3	MODE	AES 工作模式，AESEN=1 时不可修改。

Bit	助记符	功能描述
		(operation MODE) 00：模式 1：加密 01：模式 2：密钥扩展 10：模式 3：解密 11：模式 4：密钥扩展+解密 CTR 模式下配置成模式 4 将自动进入 CTR 的解密模式。即在 CHMOD=2'b10 时配置 MODE=2'b11，AES 将按照 MODE=2'b10 的情形执行。
2:1	DATATYP	选择数据类型，AESEN=1 时不可修改。具体交换规则可参考 AES 数据类型章节。 (Data type) 00：32bit 数据不交换 01：16bit 数据半字交换 10：8bit 数据字节交换 11：1bit 数据比特交换
0	EN	AES 使能 (AES enable) 0：不使能 1：使能 在任何时候清除 AESEN 位都能够复位 AES 模块 在模式 2 下该位会在一次计算完成后硬件自动清 0

14.8.2 AES 中断使能寄存器（AES_IER）

名称	AES_IER							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					WRERR_IE	RDERR_IE	CCF_IE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2	WRERR_IE	写错误中断使能，1 有效。(Write Error interrupt enable)
1	RDERR_IE	读错误中断使能，1 有效。(Read Error interrupt enable)
0	CCF_IE	AES 计算完成中断使能，1 有效。(Cipher Complete Interrupt enable)

14.8.3 AES 中断标志寄存器（AES_ISR）

名称	AES_ISR							
地址	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					WRERR	RDERR	CCF
位权限	U-0					R/W1C-0	R/W1C-0	R/W1C-0

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	WRERR	写错误标志：在计算或输出阶段发生写操作时置位，软件写 1 清零
1	RDERR	读错误标志：在计算或输入阶段发生读操作时置位，软件写 1 清零
0	CCF	AES 计算完成标志，软件写 1 清零 1：计算完成 0：计算没有完成

14.8.4 AES 数据输入寄存器（AES_DIR）

名称	AES_DIR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DIN[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	DIN[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	DIN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	DIN[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	DIN	数据输入寄存器，当 AES 需要输入加解密数据时，应该往该寄存器连续写 4 次。(AES Data Input) 模式 1（加密）：把明文从 MSB 到 LSB 分 4 次写入。 模式 2（密钥扩展）：无需使用数据输入寄存器 模式 3 和模式 4（解密）：把密文从 MSB 到 LSB 分 4 次写入。

Bit	助记符	功能描述
		MultH 模式：把乘数 A 或 C 从 MSB 到 LSB 分 4 次写入。

14.8.5 AES 数据输出寄存器（AES_DOR）

名称	AES_DOR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DOUT[31:24]							
位权限	R-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	DOUT[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	DOUT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	DOUT[7:0]							
位权限	R-0000 0000							

Bit	助记符	功能描述
31:0	DOUT	数据输出寄存器，当 AES 计算完成后，可以分四次读出加解密的结果。(AES Data Output) 模式 1（加密）：把密文从 MSB 到 LSB 分 4 次读出。 模式 2（密钥扩展）：无需使用数据输输出寄存器 模式 3 和模式 4（解密）：把明文从 MSB 到 LSB 分 4 次输出。 MultH 模式：运算结果存储在 IVR 寄存器中，无需读取 AES_DOUTR 寄存器。

14.8.6 AES 秘钥寄存器（AES_KEYx）

名称	AES_KEYx(x=0,1,2,3,4,5,6,7)							
offset	0x14 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	KEYx[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	KEYx[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	KEYx[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	KEYx[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	KEYx	AES 运算秘钥，最长 256bit，AESKEY0 存放秘钥最低 32bit，

Bit	助记符	功能描述
		AESLKEY7 存放秘钥最高 32bit。(AES Key)

14.8.7 AES 初始向量寄存器（AES_IVRx）

名称	AES_IVRx(x=0,1,2,3)							
offset	0x34 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IVRx[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	IVRx[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	IVRx[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	IVRx[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	IVRx	AES 运算 128bit 初始向量，在 MultH 模式下保存运算结果。 (AES Initial Vector Registers)

14.8.8 AES MultH 参数寄存器（AES_Hx）

名称	AES_Hx(x=0,1,2,3)							
地址	0x44 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	Hx[31:24]							
位权限	R/W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	Hx[23:16]							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	Hx[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	Hx[7:0]							
位权限	R/W-00000000							

Bit	助记符	功能描述
31:0	Hx	MultH 运算 128bit 输入 H 参数 (H Parameter) H0 保存 H[31:0]，H3 保存 H[127:96]

15 随机数发生器（TRNG）

15.1 概述

FM33FG0A使用2个Galois真随机噪声源作为真随机数种子，配合简单在线检测（32位全0全1检测）、LFSR后处理、伪随机LFSR共同组成芯片的随机数发生器。

TRNG的启动测试和完整的在线测试功能需要固件实现。

Galois噪声源的采样和LFSR建议使用4MHz时钟。两次取32bit随机数之间的间隔不得小于32个时钟周期。

真随机数发生器通过了FIPS PUB140-2测试，成功率99.9%。

15.2 功能描述

15.2.1 随机数产生

下图为真随机数发生器结构框图。

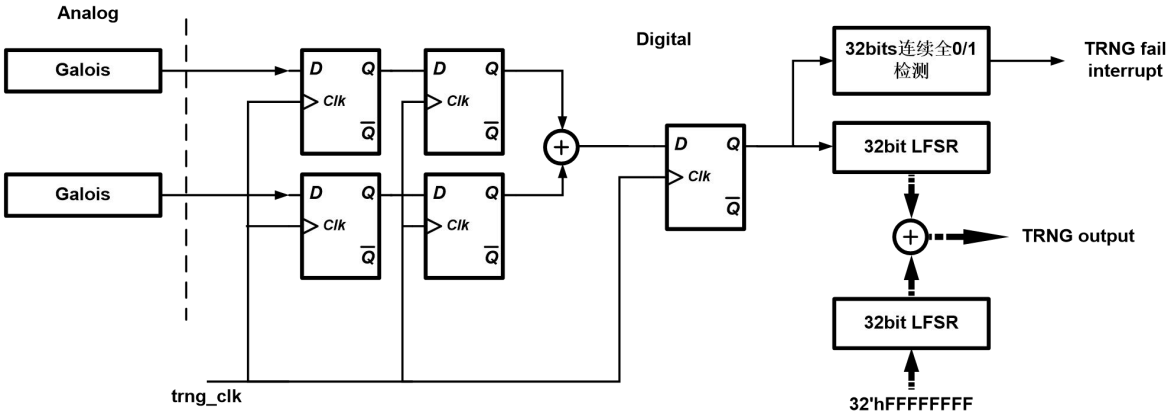


图 15-1 真随机数模块框图

真随机噪声源为2个Galois环振，Galois环振输出在数字电路内部异或并使用系统时钟采样，然后进行LFSR后处理。LFSR后处理之前经过随机数在线检测，如果发现连续32bit全0或全1的情况，则产生TRNG失效报警中断。同时为了避免小概率的真随机数性能不良情况，另外使用一组LFSR以32’hFFFFFFFF为初始值，与后处理LFSR同步运算，并以两组LFSR按位异或后的结果作为最终的32bit随机数输出。

15.2.2 工作时钟

随机数发生器的工作时钟采用RCHF的分频时钟，独立于APBCLK。为了保证随机数质量，一般建议应用使用4M时钟作为随机数工作时钟，并根据4M目标频率配置CMU模块中随机数工作时钟分频寄存器（OPCCON2.RNGPRSC）。工作时钟示意图如下：

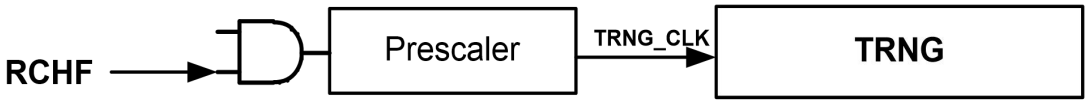


图 15-2 真随机数模块工作时钟

15.2.3 随机数读取

当随机数模块被使能后，真随机噪声源和LFSR后处理模块同时开始工作。软件通过读取RNGOUT寄存器，每次读出32bit随机数。由于LFSR循环移位周期是32cycle，为保证随机数质量，应用应保证两次读取RNGOUT之间的间隔大于32个TRNG_CLK周期。

举例来说，假设TRNG_CLK为4MHz，则两次读取RNGOUT的间隔不应小于8us。

15.2.4 CRC 运算

用作随机数后处理的LFSR也可用于进行CRC计算。

在进行CRC运算时，两组32bit LFSR分别作为输入数据寄存器和CRC运算寄存器，一次可以运算32bit数据的CRC结果。CRC运算前CPU需查询当前LFSR是否被占用，如LFSR空闲，方可以使用CRC功能。

CPU一旦启动CRC运算，LFSR自动置为复位值，随后进行32bit运算，运算结束后清除CRC启动寄存器，不产生中断；软件启动CRC后应连续查询启动寄存器状态，直到运算结束后再读取结果。

CRC多项式：

CRC32=X32+X26+X23+X22+X16+X12+X11+X10+X8+X7+X5+X4+X2+X1+X0

软件操作流程：

- 查询LFSR_BUSY，确认LFSR不在运行中
- 将待运算数据写入CRCDATA0~3
- 置位CRC_EN
- 查询并等待CRC_EN被清零
- 从LFSROUT0~3读出运算结果

15.3 寄存器

模块偏移地址：0x4001_3C00

offset 地址	名称	符号
0x00	随机数控制寄存器	RNG_CR
0x04	随机数/CRC 结果输出寄存器	RNG_DOR
0x10	随机数标志寄存器	RNG_SR
0x14	CRC 控制寄存器	RNG_CRCCR
0x18	CRC 输入数据寄存器	RNG_CRCDIR
0x1C	CRC 标志寄存器	RNG_CRCSR

15.3.1 随机数控制寄存器

名称	RNG_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	EN	RNG 使能寄存器，软件写 1 启动，完成随机数产生后自动清零 1：启动 RNG 0：关闭 RNG

15.3.2 随机数/CRC 结果输出寄存器

名称	**RNG_DOR**							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	OUT[31:24]							
位权限	R							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	OUT[23:16]							
位权限	R							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	OUT[15:8]							
位权限	R							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OUT[7:0]							
位权限	R							

Bit	助记符	功能描述
31:0	OUT	随机数生成结果或 CRC 运算结果寄存器

15.3.3 RNG 标志寄存器

名称	RNG_SR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						RBUSY	RNF
位权限	U-0						R-0	R/W-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	RBUSY	RNG 忙闲标志 1：RNG 正在生成随机数，此时软件不得使用 CRC 功能 0：RNG 空闲
0	RNF	随机数生成失败标志 1：随机数未能通过质量检测 0：随机数通过质量检测

15.3.4 CRC 控制寄存器

名称	RNG_CRCCR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	EN	CRC 使能控制寄存器，软件写 1 启动 CRC，运算完成后硬件自动清零 1：CRC 使能 0：CRC 关闭

15.3.5 CRC 输入数据寄存器

名称	RNG_CRCDIR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IN[31:24]							
位权限	R/W-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	IN[23:16]							
位权限	R/W-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN[15:8]							
位权限	R/W-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IN[7:0]							
位权限	R/W-0							

Bit	助记符	功能描述
31:0	IN	CRC 运算数据输入寄存器

15.3.6 CRC 标志寄存器

名称	RNG_CRCSR							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							DONE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	DONE	CRC 计算完成标志，软件写 0 清零 1：CRC 计算完成 0：CRC 计算未完成

16 模拟比较器（Comparator）

16.1 概述

芯片集成4个比较器，支持以下特性：

- 4个rail-to-rail比较器，支持轨到轨输入，多种功耗模式
- 可编程回滞窗口
- 灵活的输入选择
  - IO引脚输入
  - 内部基准电压及其分压和升压
  - DAC输出
- 中断事件可唤醒MCU
- 输出信号可连接到GPIO，也可以作为触发源连接到定时器、ADC
- 内建输入基准Buffer
- 窗口比较器功能
- 可编程的输出数字滤波功能

16.2 结构框图

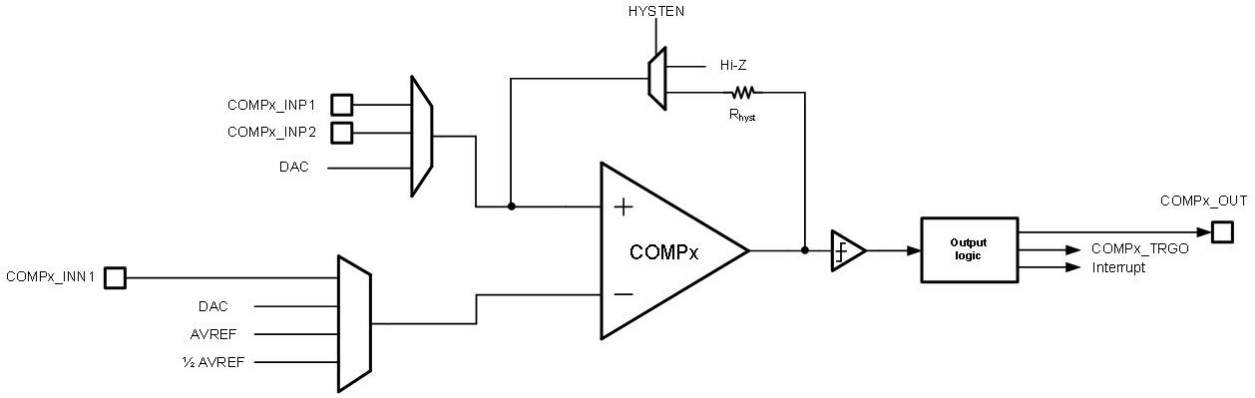


图 16-1 比较器电路框图

比较器结构如上图所示，基准电压入经过BUFFER模块后输出基准电压和基准电压分压。

比较器支持可配置的回滞功能。正端输入包括引脚、DAC，负端输入包括引脚、DAC、基准电压。

比较器输入电压范围0~VDD。高速模式下建立时间小于5us，典型传输延迟小于800ns；低速模式下建立时间小于20us，典型传输延迟小于2us。

16.3 功能描述

16.3.1 基本功能

比较器比较正端输入电压和负端输入电压，当正端电压高于负端电压时，输出逻辑高电平，反之输出逻辑低电平。

正端输入电压可以从多个引脚输入选择，负端输入电压可以选择引脚输入或者内部基准电压。

比较器输出的逻辑信号可以经过数字滤波和极性控制后输出，或产生中断信号。

比较器可以配置为快速模式或者低功耗模式，两种模式下比较器工作电流和传输延迟指标不同，应用上可以根据需要选择合适的工作模式。

16.3.2 内部比较基准选择和产生

比较器负端可以选择输入芯片内部比较基准电压，比较基准来自于AVREF及其分压。

当使用AVREF分压时，必须使能分压buffer，通过BUFENB寄存器来使能比较器模块内建的基准源buffer。

关闭BUFFER的情况下，需要置位BYPASS寄存器才能使基准电压传播到比较器输入端。

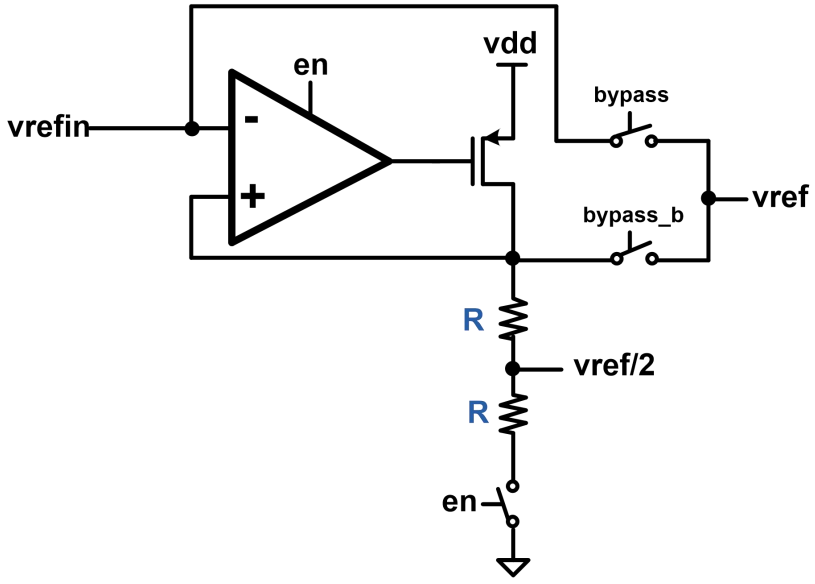


图 16-2 比较器内建基准缓冲器

16.3.3 比较器输出滞回

通过HYSTEN寄存器可以配置比较器是否使能回滞功能，实现原理如下图所示。注意当使能回滞功能时，反馈电阻可能对高阻抗输入信号造成影响，并且回滞窗口大小也与输入信号的内阻有关。

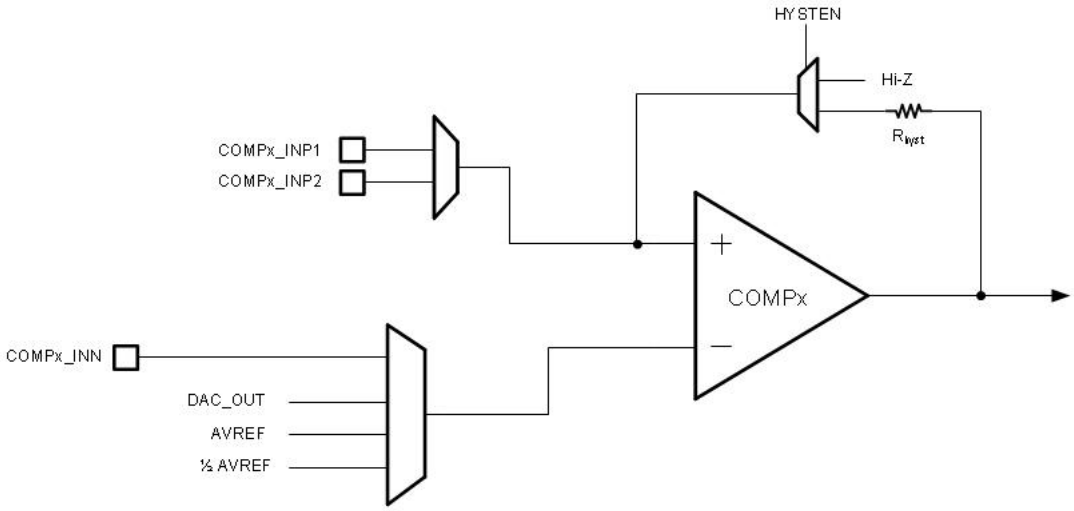


图 16-3 比较器输出滞回

16.3.4 时钟和复位

比较器模块的寄存器时钟由CMU模块提供，复位控制由RMU模块提供。

在操作比较器模块寄存器之前必须在RMU模块中清除比较器复位，并在CMU中使能比较器工作时钟。

比较器本身工作不依赖于时钟，因此可以在各种低功耗模式下工作，应用仅需在进入低功耗之前完成对比较器的寄存器配置。

16.3.5 窗口功能

窗口比较器用于监视输入模拟电压是否处于某个设定的阈值范围之内，当输入信号电平高于高阈值或者低于低阈值时，比较器输出翻转。

窗口比较器功能需要同时使用两个比较器（COMP1和COMP2），被监视的输入电压被同时连接到两个比较器的正端输入，两个比较器的负端输入分别接入高阈值和低阈值电压，两个比较器的输出被异或后输出。窗口模式下有效输出为比较器1的输出信号。

窗口比较器结构如下图所示：

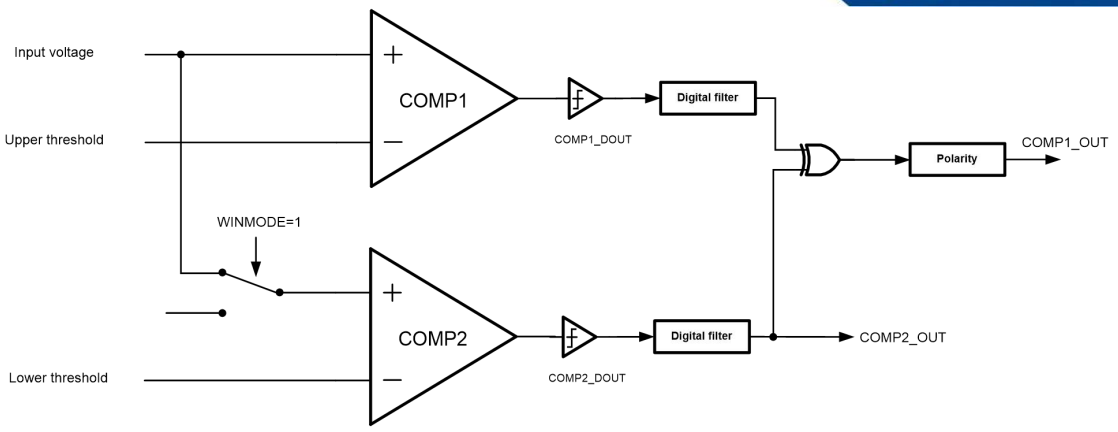


图 16-4 窗口比较器框图

窗口比较器工作时的波形示意图如下（关闭数字滤波，输出极性不取反）。当输入电压超出上下阈值设定的范围时，输出信号翻转，并且产生OOW_IF（Out-Of-Window）中断，而WIN_IF中断在输入信号每次越过阈值时产生。

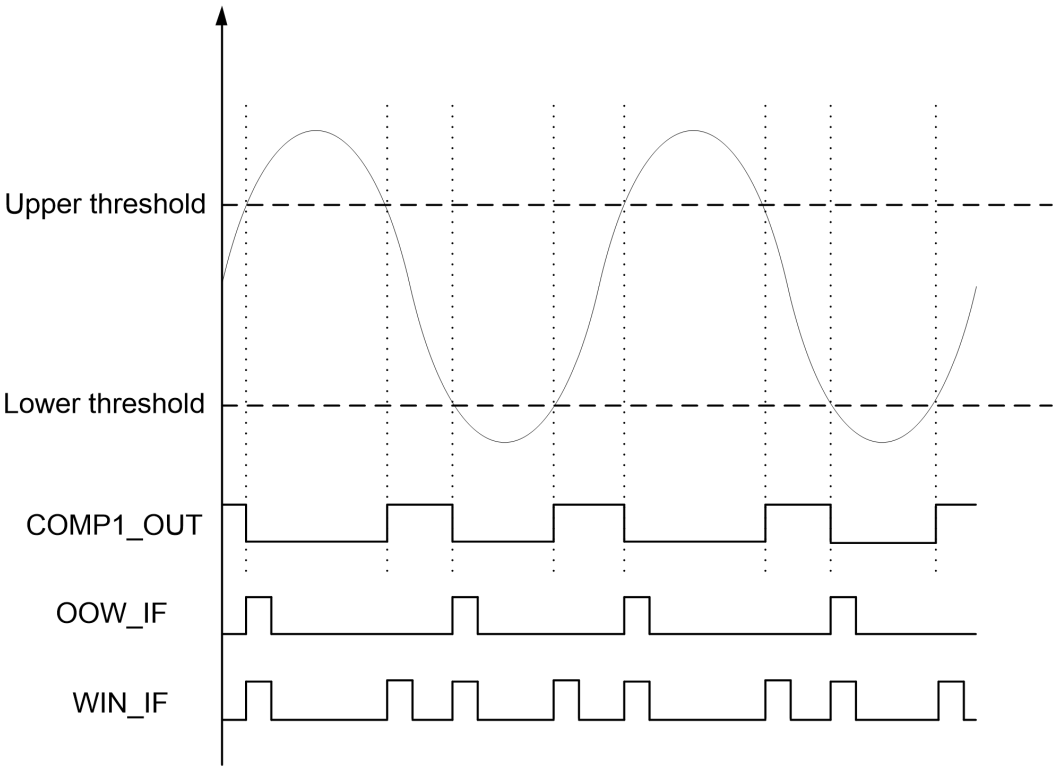


图 16-5 窗口比较器波形示意图

注意：在使用窗口比较器功能时，建议打开回滞功能，避免输出抖动。

16.3.6 输出数字滤波

比较器输出支持数字滤波功能。滤波方式是采用APBCLK连续采样比较器输出，保持DFLEN定义的

周期数为相同电平时，才认为这个电平有效。

在休眠模式下，由于APBCLK关闭，无法使用数字滤波功能，应用中如果需要在休眠模式下使用比较器，软件必须在休眠前关闭比较器数字滤波功能。

下图是DFLEN=3的示意图：

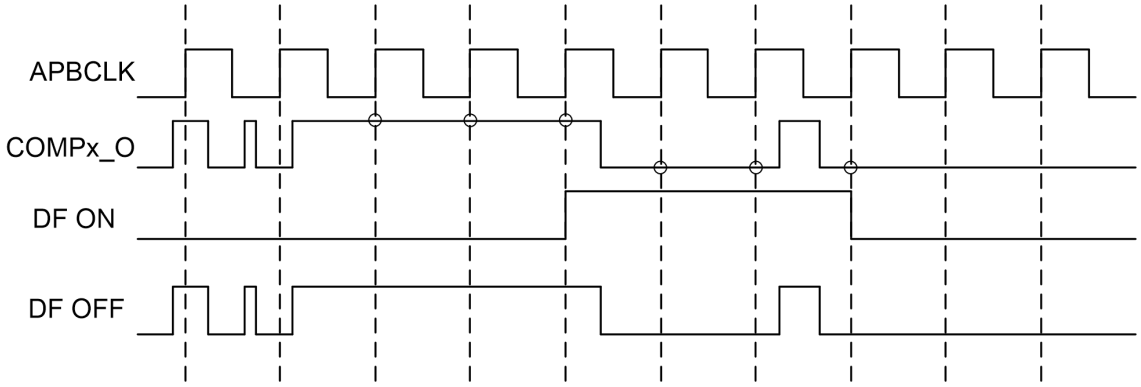


图 16-6 数字滤波（DFLEN=3）波形示意图

16.3.7 功耗和速度模式

比较器支持轨到轨输入，输入电压范围0~VDD，支持低功耗模式、高速模式，用于平衡不同应用下的功耗和延迟。

芯片上电复位后，两个比较器默认处于低功耗模式，应用可以根据需要配置其工作模式。

模式	典型功耗	典型传输延迟	典型建立时间	使用条件
高速模式	8uA	700ns	3us	所有芯片功耗模式下都可以使用
低功耗模式	1.5uA	1.7us	18us	

表 16-1 比较器工作模式

16.3.8 比较器中断

比较器输出可以在上升沿、下降沿上分别产生独立的中断事件。CMPxIE寄存器可以使能或禁止中断输出。CMPxIF标志寄存器在中断事件发生时置位，软件写1清零。软件也可以通过CMPxO寄存器直接读取比较器的输出值。

比较器输出的边沿检测采用异步电路实现，无需工作时钟，因此可以在芯片处于休眠模式的情况下产生中断，唤醒芯片。

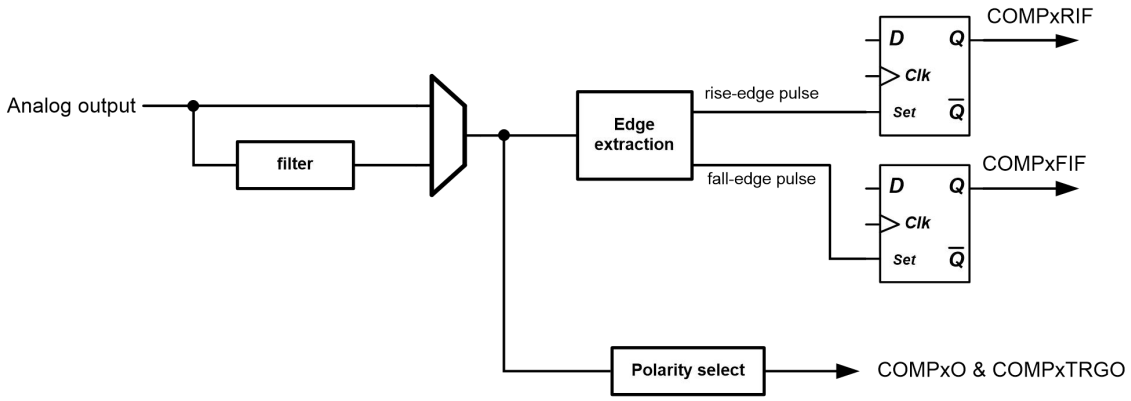


图 16-7 比较器中断产生

16.3.9 比较器输出和触发信号输出

比较器可以直接输出比较结果，还可以给其他外设电路输出触发信号。

输出逻辑示意图如下。

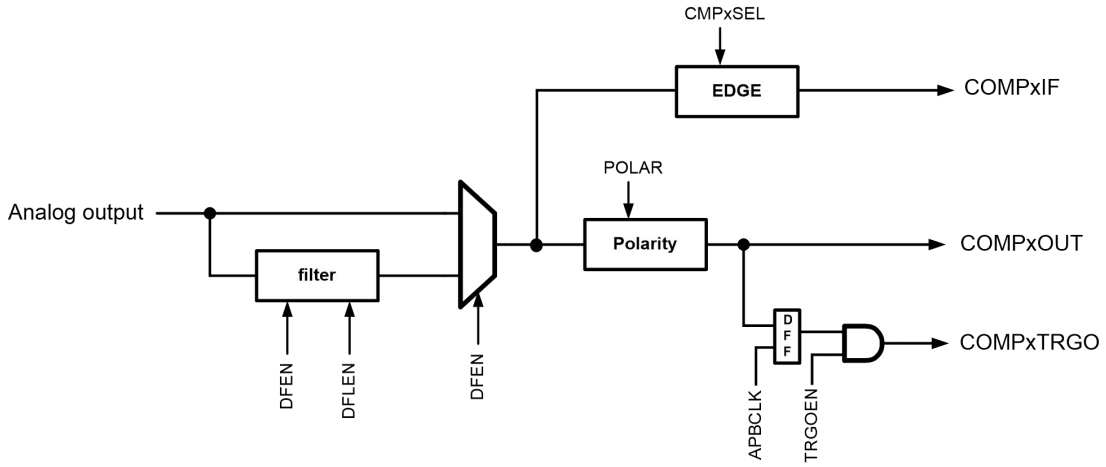


图 16-8 比较器输出逻辑

比较器输出

比较器比较结果（COMPxOUT）可以被软件实时查询，也能够输出到GPIO上。输出信号可以选择是否经过数字滤波。

比较器触发信号输出

可以在比较器输出的上升沿、下降沿分别或同时产生触发信号输出。需要输出触发信号时，必须使能COMP总线时钟，当触发事件发生时，触发信号被同步到APBCLK上升沿输出。触发信号可以被连接到定时器的内部触发输入，或者ADC的内部触发输入。

TRGOEN用于使能或禁止触发信号输出。

下图为比较器输出上升沿产生触发信号的例子。

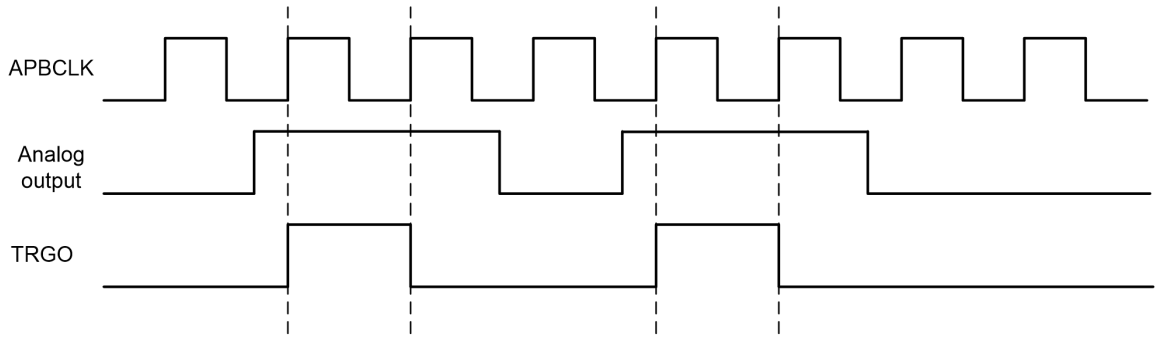


图 16-9 比较器输出上升沿产生触发输出，无滤波

下图为比较器输出上升沿和下降沿都产生触发信号的例子。

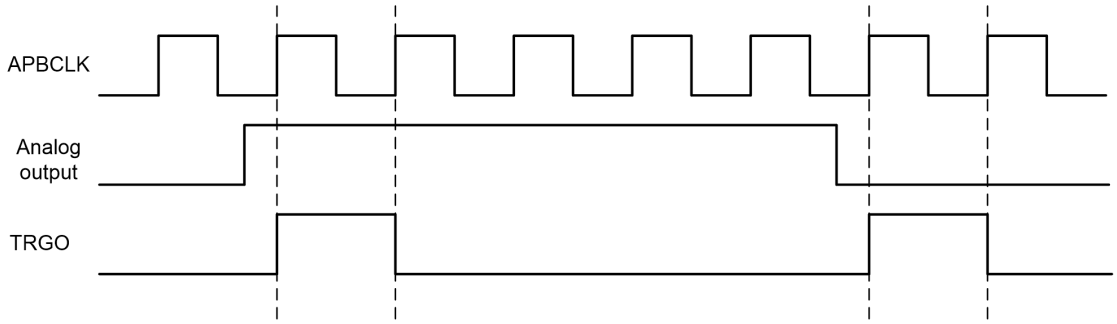


图 16-10 比较器输出上升沿下降沿产生触发输出，无滤波

比较器的触发输出可以被连接到GPTIMx的输入，以便定时器自动记录比较器输出翻转次数。

比较器的触发输出也可以作为启动ADC转换的触发信号，请参考37.6.6转换触发。

16.4 寄存器

模块基地址：0x4001_9C00

offset 地址	名称	符号
0x00	COMP1 控制寄存器	COMP1_CR
0x04	COMP2 控制寄存器	COMP2_CR
0x08	COMP3 控制寄存器	COMP3_CR
0x0C	COMP4 控制寄存器	COMP4_CR
0x10	COMP 中断配置寄存器	COMP_ICR
0x14	COMP 中断标志寄存器	COMP_IF
0x18	COMP 缓冲器控制寄存器	COMP_BUFCR

16.4.1 COMP1 控制寄存器（COMP1_CR）

名称	COMP1_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-					TRGOSEL		TRGOEN
位权限	U-0					R/W-00		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DFLEN					WINMODE	POLAR	DFEN
位权限	R/W-00000					R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MODE		-				HYSTEN	CMP1O
位权限	R/W-00		U-0				R/W-0	R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	V1PSEL			V1NSEL			CMP1EN
位权限	U-0	R/W-000			R/W-000			R/W-0

Bit	助记符	功能描述
31:27	--	RFU：未实现，读为 0
26:25	TRGOSEL	比较器触发信号输出选择 00：比较器输出的上升和下降沿都输出触发信号 01：比较器输出的上升沿输出触发信号 10：比较器输出的下降沿输出触发信号 11：比较器输出的上升和下降沿都输出触发信号
24	TRGOEN	比较器触发信号输出使能 0：禁止触发输出 1：允许触发输出
23:19	DFLEN	比较器 1 输出数字滤波长度配置寄存器。滤波长度周期是 DFLEN+1（最小滤波长度为 3） 00000：3 次 APBCLK 采样 00001：3 次 APBCLK 采样 00010：3 次 APBCLK 采样 00011：4 次 APBCLK 采样 ……

Bit	助记符	功能描述
		11111：32 次 APBCLK 采样
18	WINMODE	比较器窗口模式控制寄存器 0：禁止窗口模式 1：使能窗口模式（COMP1 和 COMP2 联动）
17	POLAR	比较器 1 输出极性控制 0：正向输出 1：取反输出
16	DFEN	比较器 1 输出数字滤波使能 0：禁止输出数字滤波 1：使能输出数字滤波
15:14	MODE	比较器 1 工作模式 00/11：低功耗模式 01：RFU 10：高速模式
13:10	--	RFU：未实现，读为 0
9	HYSTEN	比较器 1 模块迟滞使能信号
8	CMP1O	比较器 1 输出，软件只读
7	--	RFU：未实现，读为 0
6:4	V1PSEL	比较器 1 正极输入选择 000：COMP1_INP1 001：COMP1_INP2 010：DAC 其它：RFU
3:1	V1NSEL	比较器 1 负极输入选择 000：COMP1_INN1 001：DAC 010：AVREF 011：1/2 AVREF 其它：RFU
0	CMP1EN	比较器 1 使能位 0：关闭比较器 1 1：使能比较器 1

16.4.2 COMP2 控制寄存器（COMP2_CR）

名称	COMP2_CR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-					TRGOSEL		TRGOEN
位权限	U-0					R/W-00		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DFLEN					-	POLAR	DFEN
位权限	R/W-00000					U-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MODE		-				HYSTEN	CMP2O

位权限	R/W-00		U-0				R/W-0	R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	V2PSEL			V2NSEL			CMP2EN
位权限	U-0	R/W-000			R/W-000			R/W-0

Bit	助记符	功能描述
31:27	--	RFU：未实现，读为 0
26:25	TRGOSEL	比较器触发信号输出选择 00：比较器输出的上升和下降沿都输出触发信号 01：比较器输出的上升沿输出触发信号 10：比较器输出的下降沿输出触发信号 11：比较器输出的上升和下降沿都输出触发信号
24	TRGOEN	比较器触发信号输出使能 0：禁止触发输出 1：允许触发输出
23:19	DFLEN	比较器 2 输出数字滤波长度配置寄存器。滤波长度周期是 DFLEN+1（最小滤波长度为 3） 00000：3 次 APBCLK 采样 00001：3 次 APBCLK 采样 00010：3 次 APBCLK 采样 00011：4 次 APBCLK 采样 …… 11111：32 次 APBCLK 采样
18	--	RFU：未实现，读为 0
17	POLAR	比较器 2 输出极性控制 0：正向输出 1：取反输出
16	DFEN	比较器 2 输出数字滤波使能 0：禁止输出数字滤波 1：使能输出数字滤波
15:14	MODE	比较器 2 工作模式 00/11：低功耗模式 01：RFU 10：高速模式
13:10	--	RFU：未实现，读为 0
9	HYSTEN	比较器 2 模块迟滞使能信号
8	CMP2O	比较器 2 输出，软件只读
7	--	RFU：未实现，读为 0
6:4	V2PSEL	比较器 2 正极输入选择 000：COMP2_INP1 001：COMP2_INP2 010：DAC 其他：RFU
3:1	V2NSEL	比较器 2 负极输入选择 000：COMP2_INN1 001：DAC 010：AVREF 011：1/2 AVREF 100：RFU 101：RFU

Bit	助记符	功能描述
		110：RFU 111：RFU
0	CMP2EN	比较器 2 使能位 0：关闭比较器 2 1：使能比较器 2

16.4.3 COMP3 控制寄存器（COMP3_CR）

名称	COMP3_CR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-					TRGOSEL		TRGOEN
位权限	U-0					R/W-00		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DFLEN					-	POLAR	DFEN
位权限	R/W-00000					U-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MODE		-				HYSTEN	CMP3O
位权限	R/W-00		U-0				R/W-0	R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	V3PSEL			V3NSEL			CMP3EN
位权限	U-0	R/W-000			R/W-000			R/W-0

Bit	助记符	功能描述
31:27	--	RFU：未实现，读为 0
26:25	TRGOSEL	比较器触发信号输出选择 00：比较器输出的上升和下降沿都输出触发信号 01：比较器输出的上升沿输出触发信号 10：比较器输出的下降沿输出触发信号 11：比较器输出的上升和下降沿都输出触发信号
24	TRGOEN	比较器触发信号输出使能 0：禁止触发输出 1：允许触发输出
23:19	DFLEN	比较器输出数字滤波长度配置寄存器。滤波长度周期是 DFLEN+1（最小滤波长度为 3） 00000：3 次 APBCLK 采样 00001：3 次 APBCLK 采样 00010：3 次 APBCLK 采样 00011：4 次 APBCLK 采样 …… 11111：32 次 APBCLK 采样
18	--	RFU：未实现，读为 0
17	POLAR	比较器输出极性控制 0：正向输出 1：取反输出
16	DFEN	比较器输出数字滤波使能 0：禁止输出数字滤波

Bit	助记符	功能描述
		1：使能输出数字滤波
15:14	MODE	比较器工作模式 00/11：低功耗模式 01：RFU 10：高速模式
13:10	--	RFU：未实现，读为 0
9	HYSTEN	比较器 3 模块迟滞使能信号
8	CMP3O	比较器 3 输出，软件只读
7	--	RFU：未实现，读为 0
7:4	V3PSEL	比较器 3 正极输入选择 000：COMP3_INP1 001：COMP3_INP2 010：DAC 其他：RFU
3:1	V3NSEL	比较器 3 负极输入选择 000：COMP3_INN1 001：DAC 010：AVREF 011：1/2 AVREF 100：RFU 101：RFU 110：RFU 111：RFU
0	CMP3EN	比较器 3 使能位 0：关闭比较器 3 1：使能比较器 3

16.4.4 COMP4 控制寄存器（COMP4_CR）

名称	COMP4_CR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-					TRGOSEL		TRGOEN
位权限	U-0					R/W-00		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DFLEN					-	POLAR	DFEN
位权限	R/W-00000					U-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MODE		-				HYSTEN	CMP4O
位权限	R/W-00		U-0				R/W-0	R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	V4PSEL			V4NSEL			CMP4EN
位权限	U-0	R/W-000			R/W-000			R/W-0

Bit	助记符	功能描述
31:27	--	RFU：未实现，读为 0
26:25	TRGOSEL	比较器触发信号输出选择

Bit	助记符	功能描述
		00：比较器输出的上升和下降沿都输出触发信号 01：比较器输出的上升沿输出触发信号 10：比较器输出的下降沿输出触发信号 11：比较器输出的上升和下降沿都输出触发信号
24	TRGOEN	比较器触发信号输出使能 0：禁止触发输出 1：允许触发输出
23:19	DFLEN	比较器输出数字滤波长度配置寄存器。滤波长度周期是DFLEN+1（最小滤波长度为3） 00000：3 次 APBCLK 采样 00001：3 次 APBCLK 采样 00010：3 次 APBCLK 采样 00011：4 次 APBCLK 采样 …… 11111：32 次 APBCLK 采样
18	--	RFU：未实现，读为 0
17	POLAR	比较器输出极性控制 0：正向输出 1：取反输出
16	DFEN	比较器输出数字滤波使能 0：禁止输出数字滤波 1：使能输出数字滤波
15:14	MODE	比较器工作模式 00/11：低功耗模式 01：RFU 10：高速模式
13:10	--	RFU：未实现，读为 0
9	HYSTEN	比较器 4 模块迟滞使能信号
8	CMP4O	比较器 4 输出，软件只读
7	--	RFU：未实现，读为 0
7:4	V4PSEL	比较器 4 正极输入选择 000：COMP4_INP1 001：COMP4_INP2 010：DAC 其他：RFU
3:1	V4NSEL	比较器 4 负极输入选择 000：COMP4_INN1 001：DAC 010：AVREF 011：1/2 AVREF 100：RFU 101：RFU 110：RFU 111：RFU
0	CMP4EN	比较器 4 使能位 0：关闭比较器 4 1：使能比较器 4

16.4.5 COMP 中断配置寄存器（COMP_ICR）

名称	COMP_ICR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						OOW_IE	WIN_IE
位权限	U-0						R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CMP4SEL		-	CMP4IE	CMP3SEL		-	CMP3IE
位权限	R/W-00		U-0	R/W-0	R/W-00		U-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CMP2SEL		-	CMP2IE	CMP1SEL		-	CMP1IE
位权限	R/W-00		U-0	R/W-0	R/W-00		U-0	R/W-0

Bit	助记符	功能描述
31:26	--	RFU：未实现，读为 0
25	OOW_IE	Out-Of-Window 中断使能
24	WIN_IE	Window 中断使能
23:16	--	RFU：未实现，读为 0
15:14	CMP4SEL	比较器 4 中断源选择 00/11：比较器 4 输出上升或下降沿产生中断 01：比较器 4 输出上升沿产生中断 10：比较器 4 输出下降沿产生中断
13	--	RFU：未实现，读为 0
12	CMP4IE	比较器 4 中断使能，1 有效
11:10	CMP3SEL	比较器 3 中断源选择 00/11：比较器 3 输出上升或下降沿产生中断 01：比较器 3 输出上升沿产生中断 10：比较器 3 输出下降沿产生中断
9	--	RFU：未实现，读为 0
8	CMP3IE	比较器 3 中断使能，1 有效
7:6	CMP2SEL	比较器 2 中断源选择 00/11：比较器 2 输出上升或下降沿产生中断 01：比较器 2 输出上升沿产生中断 10：比较器 2 输出下降沿产生中断
5	--	RFU：未实现，读为 0
4	CMP2IE	比较器 2 中断使能，1 有效
3:2	CMP1SEL	比较器 1 中断源选择 00/11：比较器 1 输出上升或下降沿产生中断 01：比较器 1 输出上升沿产生中断 10：比较器 1 输出下降沿产生中断
1	--	RFU：未实现，读为 0
0	CMP1IE	比较器 1 中断使能，1 有效

*注：为了避免误触发中断，应在关闭中断使能的情况下设置中断源选择寄存器。

16.4.6 COMP 中断标志寄存器（COMP_IF）

名称	COMP_IF							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CMP4IF	CMP3IF
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			OOW_IF	WIN_IF	-	CMP2IF	CMP1IF
位权限	U-0			R/W-0	R/W-0	U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:10	--	RFU：未实现，读为 0
9	CMP4IF	比较器 4 中断标志，硬件置位，软件写 1 清零
8	CMP3IF	比较器 3 中断标志，硬件置位，软件写 1 清零
7:5	--	RFU：未实现，读为 0
4	OOW_IF	Out-of-Window 中断标志，硬件置位，软件写 1 清零
3	WIN_IF	Window 中断标志，硬件置位，软件写 1 清零
2	--	RFU：未实现，读为 0
1	CMP2IF	比较器 2 中断标志，硬件置位，软件写 1 清零
0	CMP1IF	比较器 1 中断标志，硬件置位，软件写 1 清零

16.4.7 COMP 缓冲器控制寄存器（COMP_BUFCR）

名称	COMP_BUFCR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					BUFBYP	-	BUFENB
位权限	U-0					R/W-0	U-0	R/W-1

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0

Bit	助记符	功能描述
2	BUFBYP	缓冲器 bypass 使能 0：不 bypass 1：bypass
1	--	RFU：未实现，读为 0
0	BUFENB	缓冲器使能 0：使能基准缓冲器 1：关闭基准缓冲器 *注：如要使用 AVREF/2 分压，必须使能缓冲器* *注：当 BUFBYP=0 时，以 AVREF 作为基准，必须开启缓冲器*

17 硬件除法器（HDIV）

17.1 概述

HDIV用于帮助软件加速除法运算。

HDIV包含的硬件除法器是一个有符号数整数除法器，可以输出32bit被除数和16bit除数，输出32bit商和32bit余数。

乘法功能在Cortex-M0内核中完成。

特点：

- 有符号整数运算（二进制补码格式）
- 32bit被除数、16bit除数
- 输出32bit商和32bit余数
- 除以0警告
- 一次计算需要8个24MHz周期

17.2 工作流程

软件按照如下步骤调用硬件除法器。

- 向DIVEND寄存器写入32bit被除数（二进制补码）
- 向DIVSOR寄存器写入16bit除数（二进制补码）
- 硬件除法器在软件写入DIVSOR后自动开始运算，同时置位BUSY寄存器
- 软件查询BUSY标志，直到运算完成后BUSY自动清零
- 查询DIV_BY_0标志
- 读取QUOT寄存器中的商
- 读取REMD寄存器中的余数

软件按照以下步骤调用累加器。

- 清零DATA_OUT寄存器
- 向DATA_IN寄存器写入16bit累加数据（二进制补码）
- 此数据被自动累加到DATA_OUT
- 多次写入DATA_IN后从DATA_OUT读取累加结果

17.3 寄存器

模块基地址：0x40019400

offset 地址	名称	符号
0x00	被除数寄存器	DIVEND
0x04	除数寄存器	DIVSOR
0x08	商寄存器	QUOT
0x0C	余数寄存器	REMD
0x10	状态标志寄存器	DIVSR

17.3.1 被除数寄存器

名称	DIVEND							
地址	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DIVEND[31:24]							
位权限	R/W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DIVEND[23:16]							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DIVEND[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIVEND[7:0]							
位权限	R/W-00000000							

位号	位名	说明
31:0	DIVEND	32bit 有符号被除数

17.3.2 除数寄存器

名称	DIVSOR							
地址	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DIVSOR[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIVSOR[7:0]							

位权限	R/W-00000001

位号	位名	说明
31:0	--	RFU：未实现，读为 0
15:0	DIVSOR	16bit 有符号除数

17.3.3 商寄存器

名称	QUOT							
地址	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QUOT[31:24]							
位权限	R							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QUOT [23:16]							
位权限	R							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QUOT [15:8]							
位权限	R							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QUOT [7:0]							
位权限	R							

位号	位名	说明
31:0	QUOT	32bit 有符号商 （仅地址，无实际寄存器，读取时直接返回除法器模块输出）

17.3.4 余数寄存器

名称	REMD							
地址	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	REMD[15:8]							
位权限	R							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	REMD[7:0]							
位权限	R							

位号	位名	说明
31:16	--	RFU：未实现，读为 0
15:0	REMD	16bit 有符号余数 （仅地址，无实际寄存器，读取时直接返回 DW_div 模块输出）

17.3.5 状态标志寄存器

名称	HDIVSR							
地址	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						DIV_BY_0	BUSY
位权限	U-0						R-0	R-0

位号	位名	说明
31:2	--	RFU：未实现，读为 0
1	DIV_BY_0	**除数为 0 标志** 1 = 除数为 0 0 = 除数不为 0
0	BUSY	**运算过程指示** 1 = HDIV 在计算过程中，结果未就绪 0 = 计算完毕，结果就绪 软件在写入除数后，HDIV 开始计算，软件应查询 BUSY 为低后再读取商和余数寄存器

18 双线串行总线和系统管理总线（I²C-SMBus）

18.1 概述

I²C-SMB 模块实现 MCU 与外部 I²C 或 SMBus 器件之间的同步通信。

本模块支持 I²C 的单主机、多主机和从机模式，兼容标准模式（standard-mode 100Kbps）、快速模式（fast-mode 400Kbps）、增强快速模式（fast-mode+ 1Mbps）。

本模块也可以支持 SMBus rev3.0（System Management Bus）和 PMBus（Power Management Bus）协议规范。

本芯片最大支持 2 路独立 I2C-SMBus。

18.2 主要功能

- I2C 协议兼容
  - 主机和从机模式
  - 多主机通信
  - 传输速度支持 standard mode(100Kbps), fast mode(400Kbps)
  - 7 位和 10 位寻址
  - General Call
  - 支持可编程的 setup/hold 时序
  - 支持时钟延展
- SMBus rev 3.0 兼容
  - 硬件 PEC 和 ACK
  - 支持 ARP（Address Resolution Protocol）
  - 支持 Host 和 Device 功能
  - SMBus alert 信号
  - 超时检测和 IDLE 检测
- PMBus rev 1.3 兼容
- 独立时钟工作，不依赖于 APB 总线时钟
- 支持 DMA 数据搬运
- 低功耗特性

- 低功耗从机设计，不需要片上时钟即可完成数据收发
- 从机唤醒功能：支持从机地址匹配唤醒、数据帧接收完成唤醒或 START 检测唤醒

18.3 模块框图

18.3.1 I²C-SMB 模块框图

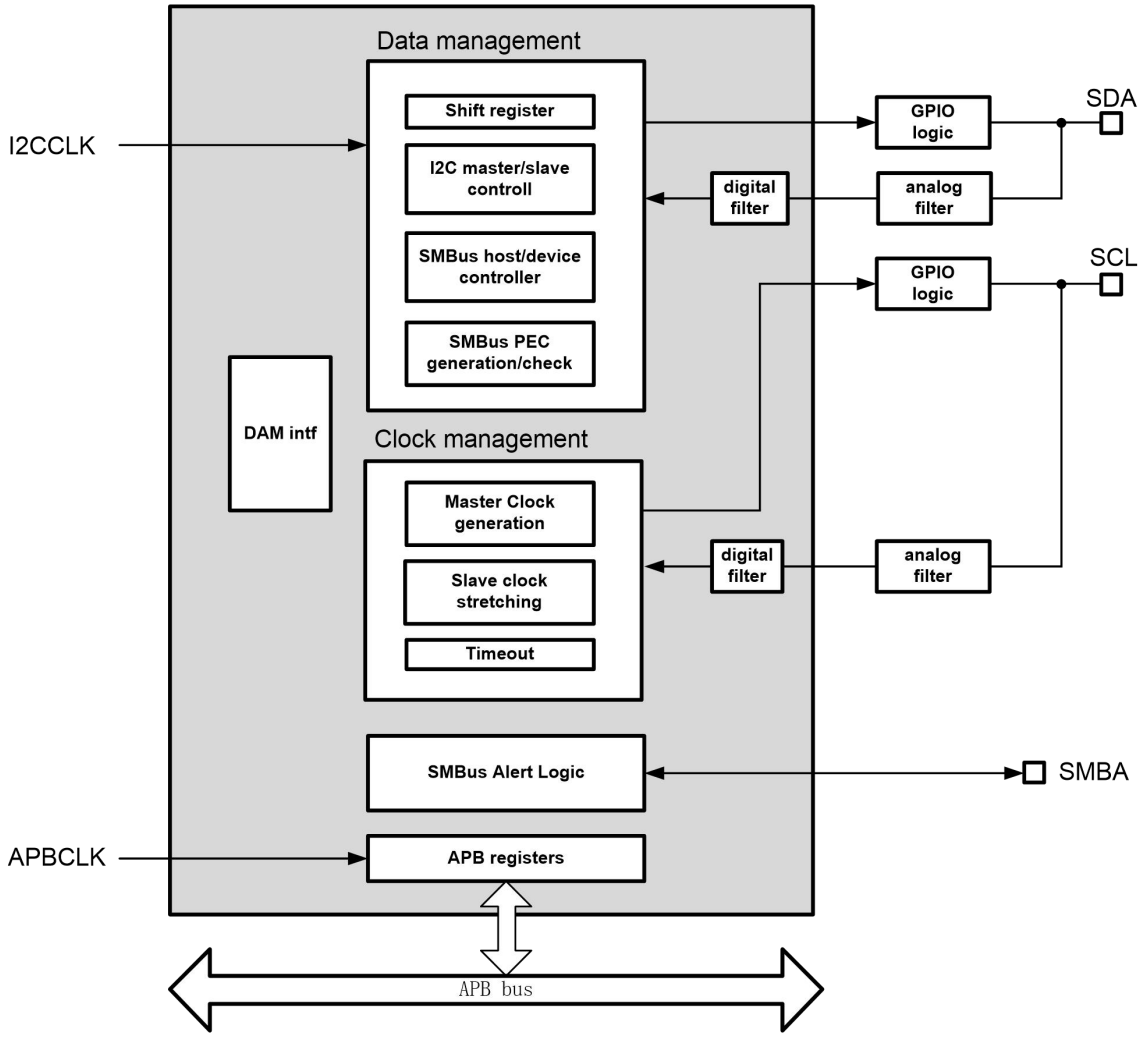


图 18-1 I²C-SMB 模块框图

18.4 时钟和复位

I²C 主机和从机都采用了双时钟结构：

- 主机和从机的总线寄存器时钟用 PCLK 表示，来源于 APBCLK。当 CPU 或者 DMA 需要访问 I²C 内部寄存器时，必须使能 PCLK。
- 主机的数据收发时钟用 I2CCLK 表示，除了可以来源于 APBCLK，还可以来源于 RCHF、SYSCLK，能够独立于 APBCLK 工作。必须使能 I2CCLK 才能进行数据收发。
- 从机通过采样 SCL 输入的有效沿来进行数据收发

PCLK 和 I2CCLK 的控制都在 CMU 模块内完成，进行 I²C 通信前必须正确配置相应的 CMU 控制寄存器。

采用双时钟结构，可以使 I²C 的工作不受限于 APBCLK 的配置，当某些外设需要工作在很高的 APBCLK 频率上时，I²C 仍可以工作在降低的频率上；或者反过来，CPU 工作在较低的频率上，也不影响 I²C 以较高的波特率进行数据通信。

理论上 PCLK 和波特率时钟之间没有相对关系的约束，波特率时钟可以快于或者慢于 PCLK。但是应用需要注意当两者频率相差较大时，CPU 或者 DMA 是否来得及进行数据搬运。

注意：模块工作前需先清除 RMU 模块中的 I²C 复位寄存器（I2CRST）。

18.5 接口时序

18.5.1 通信过程

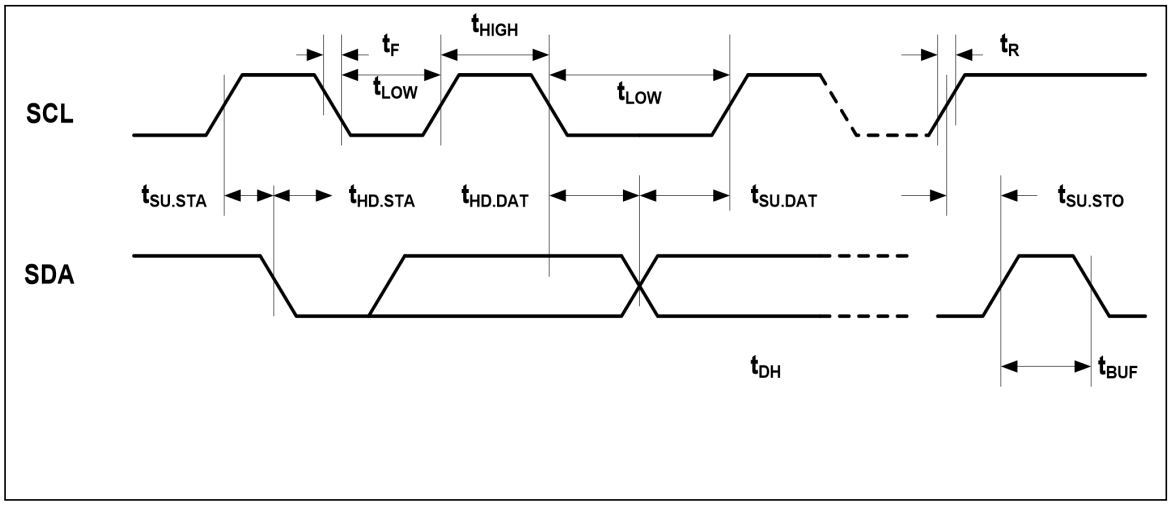


图 18-2 I²C 总线时序

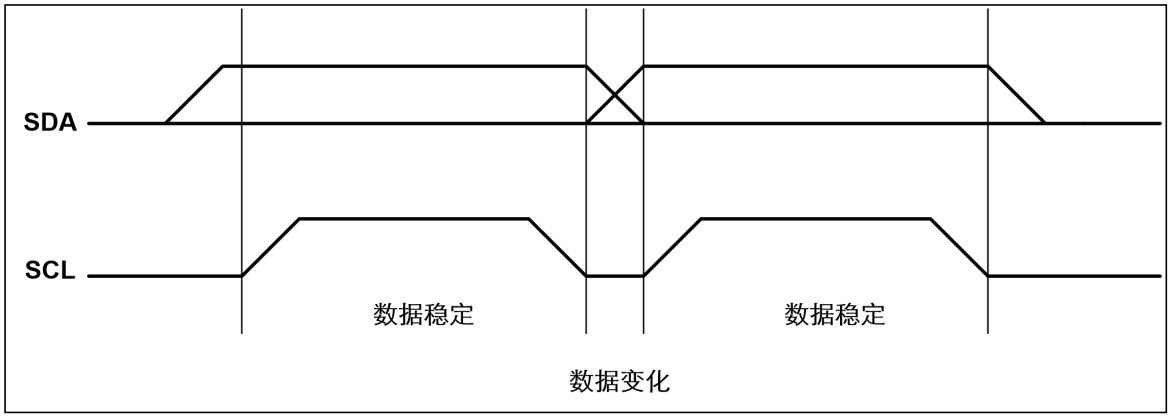


图 18-3 数据有效时序

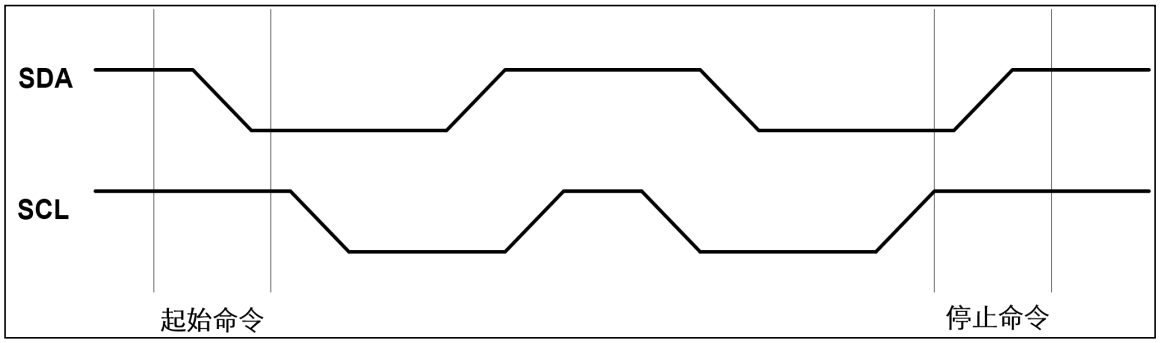


图 18-4 起始（Start）与停止(Stop)命令定义

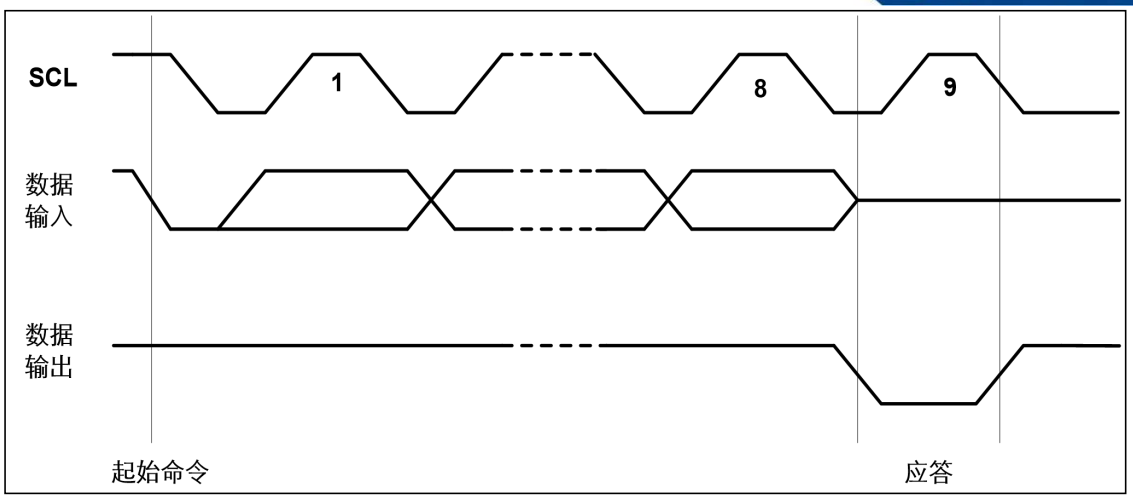


图 18-5 输出应答(ACK)

18.5.2 接口时序描述

时钟有效时序：SDA 引脚通常被外围器件拉高。SDA 引脚的数据应在 SCL 为低时变化（参见图 18-3）；当数据在 SCL 为高时变化，将视为下文所述的一个起始或停止命令。

起始命令：当 SCL 为高，SDA 由高到低的变化被视为起始命令，必须以起始命令作为任何一次读/写操作命令的开始（参见图 18-4）。

停止命令：当 SCL 为高，SDA 由低到高的变化被视为停止命令，在一个读操作后，停止命令会使从机进入等待态低功耗模式（参见图 18-4）。

输出应答：SDA 上的数据都是以 8 位为一组串行输入和输出的，MSB 先发，接收方在收完每个字节后应当在第 9 个周期回发一个回应 acknowledge 位（以下简称 ACK），ACK 的时钟由主机提供。发送方在 ACK 期间悬空 SDA，接收方须将 SDA 拉低，确保 ACK 时钟高电平期间 SDA 为低，形成有效的 ACK 信号（参见图 18-5）。

18.5.3 总线时序参数表

I^2C 协议规定的总线时序参数表如下。

参数	符号	Standard mode(100K)		Fast mode (400K)		Fast mode plus (1M)		单位
		min	max	min	max	min	max	
SCL 时钟频率	F_{SCL}	0	100	0	400	0	1000	kHz
启动条件建立时间	$T_{SU:STA}$	4.7	-	0.6	-	0.26	-	us
启动条件保持时间	$T_{HD:STA}$	4.0	-	0.6	-	0.26	-	us
时钟低电平时间	T_{LOW}	4.7	-	1.3	-	0.5	-	us
时钟高电平时间	T_{HIGH}	4.0	-	0.6	-	0.26		us

参数	符号	Standard mode(100K)		Fast mode (400K)		Fast mode plus (1M)		单位
		min	max	min	max	min	max	
数据输入建立时间	$T_{SU:DAT}$	250	-	100	-	50	-	ns
数据输入保持时间	$T_{HD:DAT}$	0	-	0	-	0	-	us
SDA 和 SCL 上升时间	T_R	-	1000	20	300		120	ns
SDA 和 SCL 下降时间	T_F	-	300	20x(VDD /5.5)	300	20x(VDD /5.5)	120	ns
停止条件建立时间	$T_{SU:STO}$	4.0	-	0.6	-	0.26	-	us
STOP 和 START 之间的总线空闲时间	T_{BUF}	4.7	-	1.3	-	0.5	-	us
总线的容性负载	Cb	-	400	-	400	-	550	pF
数据有效时间	$t_{VD;DAT}$	-	3.45	-	0.9	-	0.45	us
ACK 有效时间	$t_{VD;ACK}$	-	3.45	-	0.9	-	0.45	us
噪声容限低值	V_{nL}	0.1V_{DD}	-	0.1V_{DD}	-	0.1V_{DD}	-	V
噪声容限高值	V_{nH}	0.2V_{DD}	-	0.2V_{DD}	-	0.2V_{DD}	-	V

表 18-1 I²C 接口时序要求

以上各项参数可以在下图中找到对应的图示。

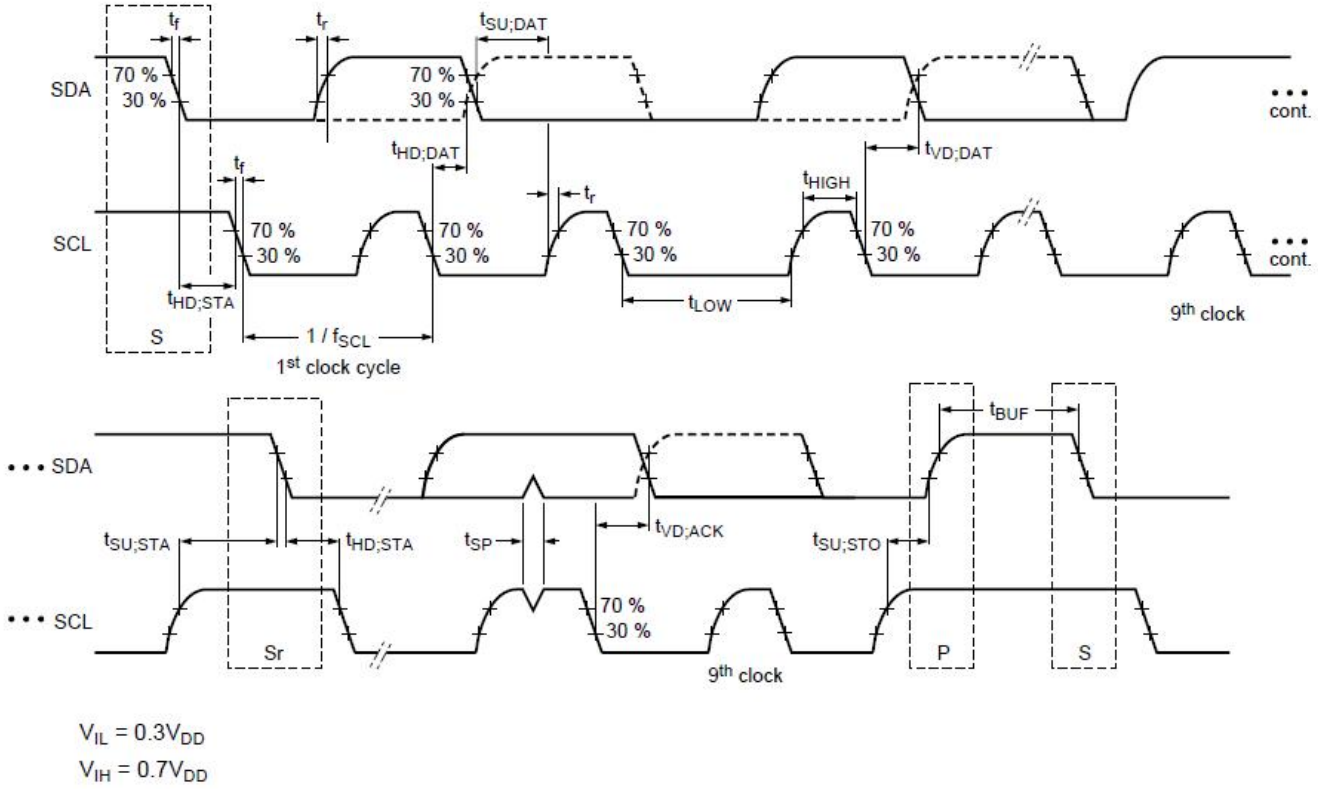


图 18-6 I²C 时序参数图例

18.6 I²C 功能描述

I²C 模块包含主机、从机、IO 接口控制和全局配置寄存器。其中主机部分可以兼容 I²C master 和 SMBu master，从机部分可以兼容 I²C slave 和 SMBus slave；同时此模块还可以支持 SMBus Host 功能。

模块支持以下工作模式。

- I²C 主机发送
- I²C 主机接收
- I²C 从机接收
- I²C 从机发送

芯片上电后 I²C 模块默认关闭，主机和从机都不工作。软件通过置位 EN 寄存器使能 I²C 模块，模块使能后默认处于从机模式，并准备好接收地址和数据。当软件对 START 寄存器写 1 后，模块在总线上发起 START 时序，此时模块自动进入主机模式。当总线仲裁失败，或者发送完 STOP 时序后，模块自动从主机模式回到从机模式，以便于兼容多主机总线应用。

18.6.1 预设从机地址表

I²C和SMBus协议规定了一部分预设地址，模块能够自动识别其中一部分预设地址，其他预设地址由软件进行协议层判断和处理。

对于10bit从机地址应用，即A10EN=1的情况下，要求第一个字节必须以11110开头，否则将触发ADDR_ERROR错误标志。而在A10EN=0的情况下，如果从机收到了11110开头的地址字节，也会置位ADDR_ERROR错误标志。

从机地址	R/W_ bit	描述
0000 000	0	General Call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Reserved for different bus format
0000 011	X	Reserved for future purpose
0000 1XX	X	HS-mode master code
0001 000	X	SMBus host
0001 001	X	Smart Battery Charger
0001 010	X	Smart Battery Selector Smart Battery System Manager
0001 011	X	Smart Battery
0001 100	X	SMBus Alert Response Address
0101 000	1	PMBus ZONE READ

0101 100	X	Reserved by SMBus for LCD controller
0101 101	X	Reserved by SMBus for CCFL Backlight driver
0110 111	0	PMBus ZONE WRITE
1000 0XX	X	Reserved by SMBus for PCMCIA Socket Controller
1000 100	X	Reserved by SMBus for VGA controller
1001 0XX	X	Prototype Address
1100 001	X	SMBus Device Default Address
1111 0XX	X	10bit slave addressing
1111 1XX	X	Reserved for future purpose

表 18- 2 I²C-SMBus 从机保留地址定义

在使能了SMBus功能后，模块能够自动识别SMBus host, SMBus Alert Response Address, SMBus Device Default Address，并给出相应的标志和中断事件。

18.6.2 I2C 初始化过程

进行I²C通信前必须正确的初始化I²C模块，建议软件按照以下步骤进行初始化操作：

- 清零RMU模块的I2CRST寄存器，确保I²C模块不处于复位状态
- 置位CMU模块的I2C_APBEN寄存器，使能I²C模块寄存器总线接口时钟
- 配置CMU模块的I2C_CKSEL和I2C_CKEN寄存器，选择并使能I²C工作时钟（如果是从机模式，不需要这个步骤）
- 根据需要使能或禁止模拟滤波功能（SCL和SDA输入模拟滤波，>50ns）
- 根据需要使能或禁止数字滤波功能

IO配置

需要将用于I²C通信的GPIO配置为数字外设功能。用于I²C通信的IO必须配置为开漏输出模式，支持Fm+模式的IO必须具有20mA的sink能力，支持HS模式的IO则必须具备电流源驱动能力。

请查询芯片数据手册以获得更多IO相关信息。

主机波特率配置

I²C主机需要在使能前配置通信波特率，而从机通信速率由主机发送的SCL决定，因此不需要配置。

BRGH和BRGL波特率配置寄存器用于产生通信波特率。BRGH和BRGL是9 bit波特率分频系数，波特率计算公式如下：

SCL周期T_{SCL}＝T_{BRGH} + T_{BRGL}

其中BRGH定义SCL高电平宽度，BRGL定义SCL低电平宽度

T_{BRGH}＝T_{I2CCLK}×(BRGH+1)

T_{BRGL}＝T_{I2CCLK}×(BRGL+1)

T_{I2CCLK}为I²C工作时钟周期

例如对于100k波特率，T_{SCL}=10us；若I²C工作时钟为8M，则T_{I2CCLK}=125ns。假设要求SCL占空比为50%，即SCL高低电平都是5us宽度，根据以上公式可以计算得到BRGH=BRGL=39

噪声滤波

SDA和SCL输入都支持噪声滤波功能。噪声滤波包含模拟滤波和数字滤波两部分。

模拟滤波的滤波长度不小于50ns，符合I²C协议要求的fast-mode和fast-mode plus模式下对至少50ns噪声脉冲的抑制能力。软件可以通过ANFEN寄存器使能或禁止模拟滤波功能，不论是主机还是从机模式，都可以使能模拟滤波。

数字滤波功能由DFEN寄存器单独使能或禁止，主机和从机都可以使能数字滤波。使能了数字滤波后，SDA和SCL总线输入到模块内部信号，在连续采样规定次数一致后，才会改变，此功能可以抑制一定宽度的毛刺信号，滤波长度由DNF×T_{I2CCLK}决定。

注意，使能数字滤波的情况下，从机必须使能I2CCLK，因此无法在休眠模式下使用。如果从机希望不使用I2CCLK工作，则必须关闭数字滤波。

18.6.3 I2C 主机功能

I²C总线上总是由主机提供同步时钟SCL，SDA数据流方向可以是主机发送从机接收，或者从机发送主机接收。本模块的I²C主机功能支持以下特性：

- 支持多主机总线
- 支持时钟同步
- 支持时钟仲裁
- 7位或10位从机寻址
- 支持从机时钟延展
- 可编程总线时序

- DMA支持

18.6.3.1 7bit 寻址和数据通信流程

在7bit寻址时，主机发送的第一个字节包含从机地址和传输方向位（$R/\overline{W}$），根据$R/\overline{W}$决定后续传输是主机向从机写入数据（$R/\overline{W}$=0）或主机从从机读取数据（$R/\overline{W}$=1）。

名字	Slave Address Byte							
位	7	6	5	4	3	2	1	0
位名	address							R/W

位描述：

位	位名	功能
7-1	address	Slave device address
0	R/W	0: Write 表示发送数据（master 发送） 1: Read 表示请求数据（slave 回发）

主机向从机写入数据

典型的7bit寻址，主机向从机写入数据的帧结构如下图所示。

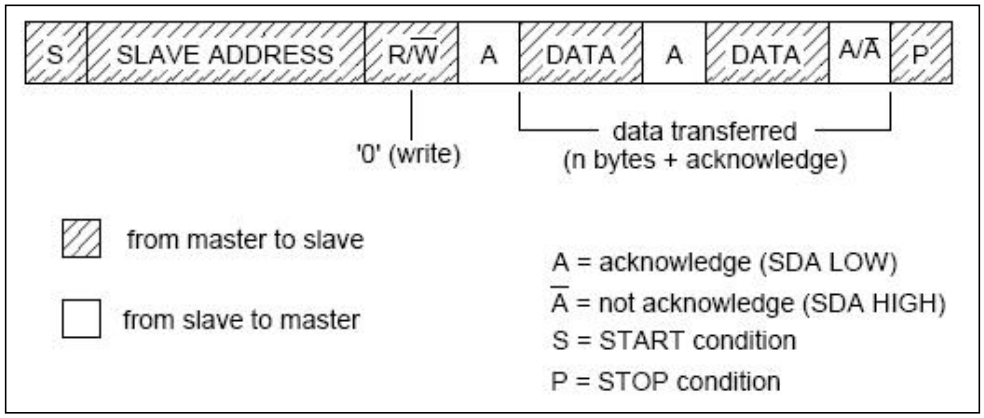


图 18-7 主机向 7 位地址从机写入数据时的帧格式

1、主机清零 RWN 寄存器，并发起 START 时序

2、主机发送从机地址，从机地址包含 7 位从机地址和 1 位 R/W 标志位，发送数据时 R/W 位为 0

4、主机发送第一帧 8 位数据

5、主机在每次发送完 8 位数据后，会在第 9 个 SCL 判断是否检测到有效的 ACK，如果主机检测到 ACK 成功后，会继续输出下一字节数据

6、若从机无法响应 ACK，主机检测到 NACK 后应发送 STOP 时序终止发送

7、主机完成所有数据发送后，发送 STOP 时序

软件启动 I²C 主机发送的操作流程如下图：

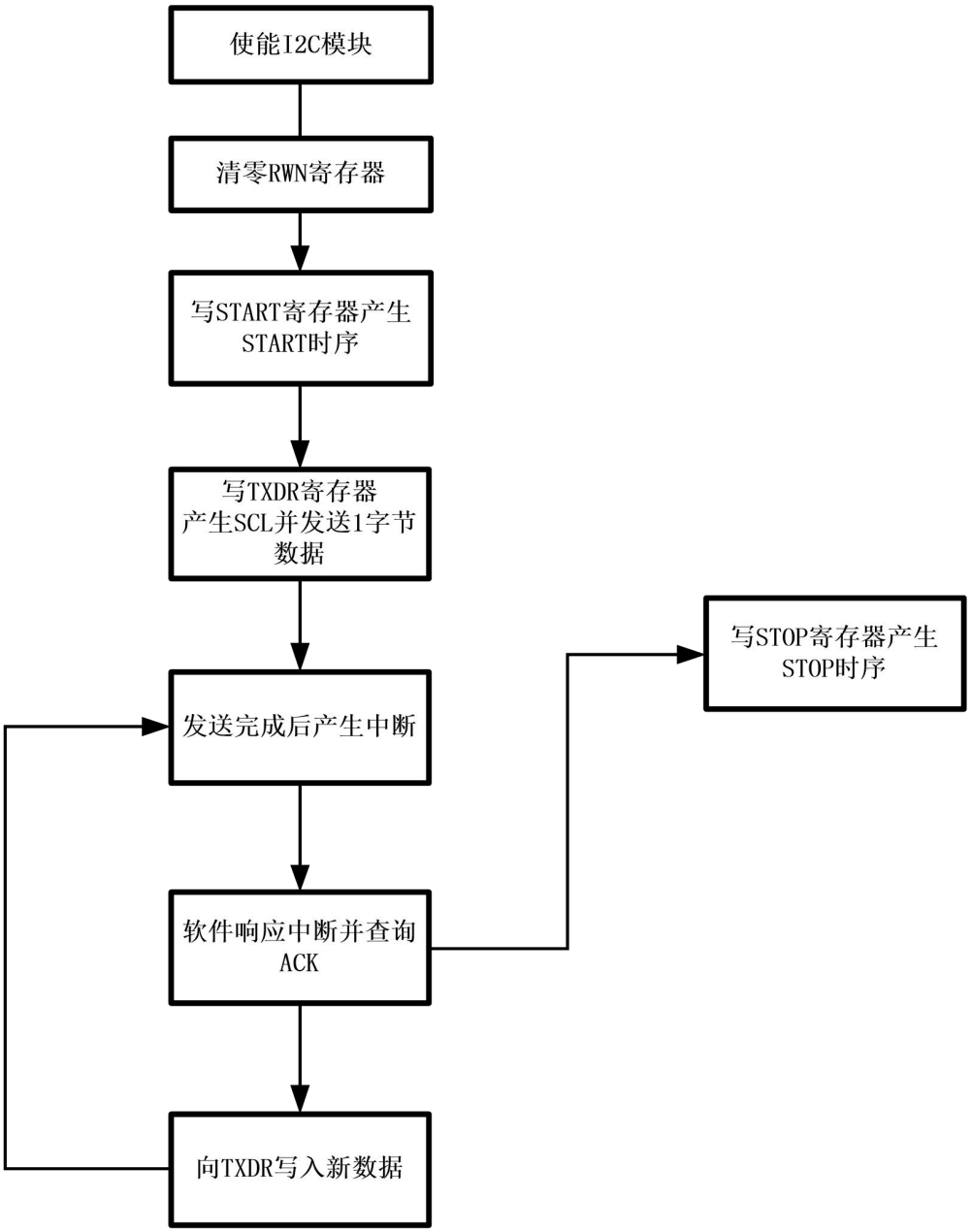


图 18-8I²C 主机软件发送数据流图

I²C主机对7位地址从机写入数据的波形示意图如下：

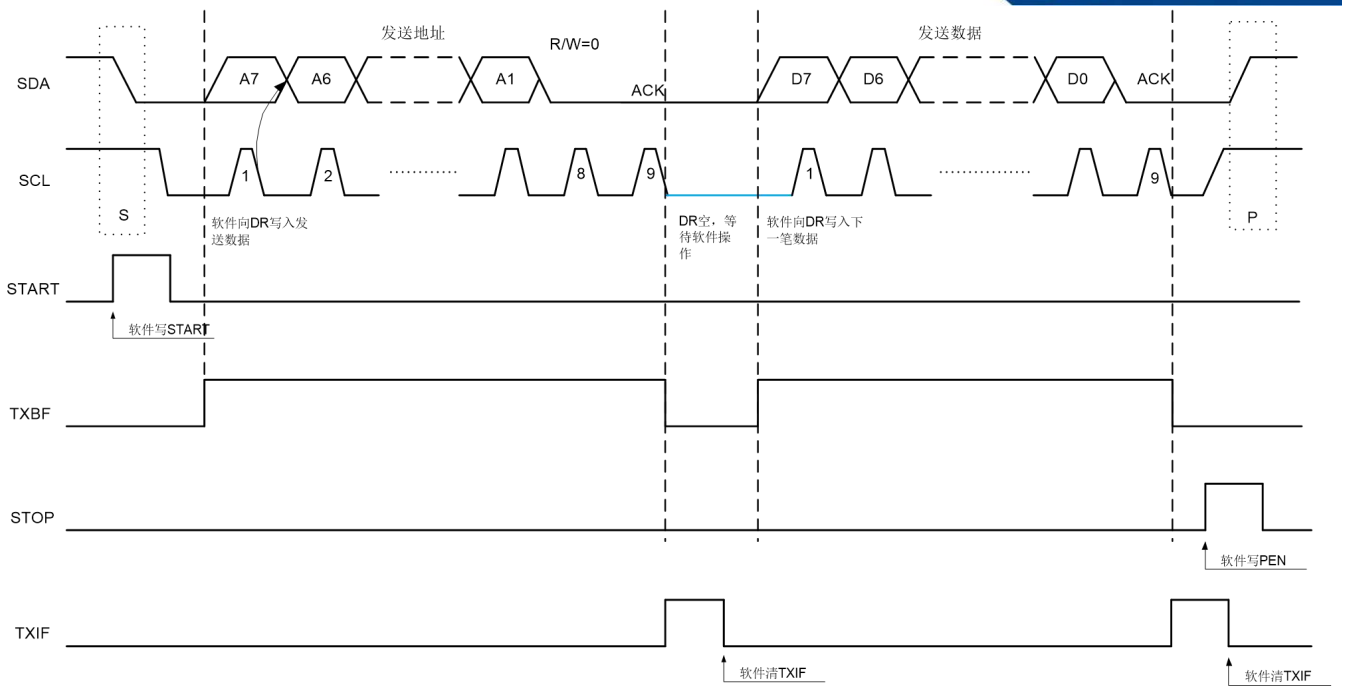


图 18-9 I²C 主机对 7 位地址从机发送数据流图

当数据寄存器DR为空时，硬件会保持SCL为低电平，等待软件将下一笔数据写入DR。如上图SCL蓝色部分所示。

主机从从机读取数据

典型的7bit寻址，主机从从机读取数据的帧格式如下图所示。

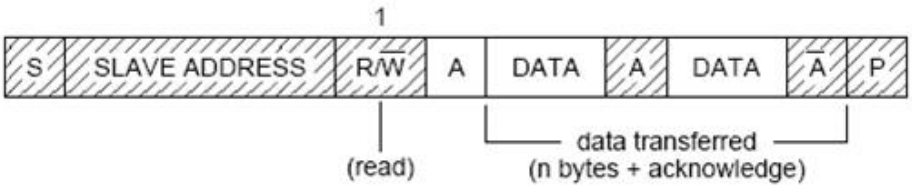


图 18-10 主机从 7 位地址从机读取数据时的帧格式

1、 主机置位 RWN 寄存器并发起 START 时序

2、 主机发送从机地址，从机地址包含 7 位从机地址和 1 位 R/W 标志位，数据读取时 R/W 位为 1

3、 此时重新置位 RWN，主机转为接收状态

4、 主机开始接收第一字节 8 位数据，并在第 9 个 SCL 向从机发送有效 ACK,从而继续读取下一字节 8 位数据

5、 每次接收完 1 字节数据后软件置位 RWN 开始读取下一个字节

6、 主机读取最后一个字节后，在第 9 个 SCL 向从机发送 NACK

7、 主机发送 STOP 时序终止读取

软件启动 I²C 接收的操作流程如下图：

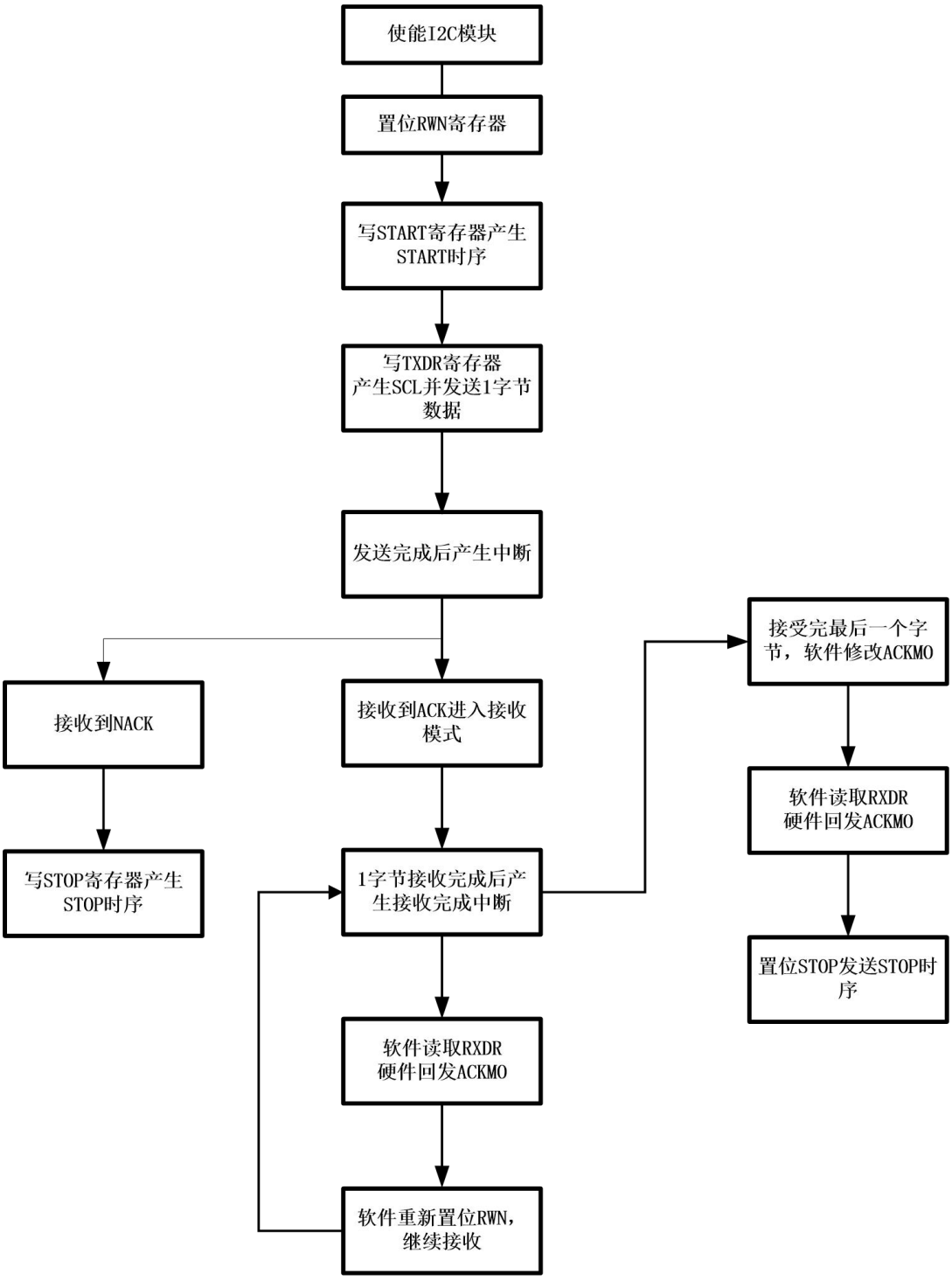


图 18-11 I²C 软件接收数据流图

主机每次接收完从机发送的数据后，根据ACKMO寄存器回发响应。ACKMO复位值为0，即默认状

态下主机回发ACK。如果软件希望主机在接收完成后回发NACK，则需要在前一个字节接收完成中断中将ACKMO寄存器改写为1。ACKMO为1的情况下，主机在发送完响应后会自动清零ACKMO。

I²C主机从7位地址从机读出数据的波形示意图如下：

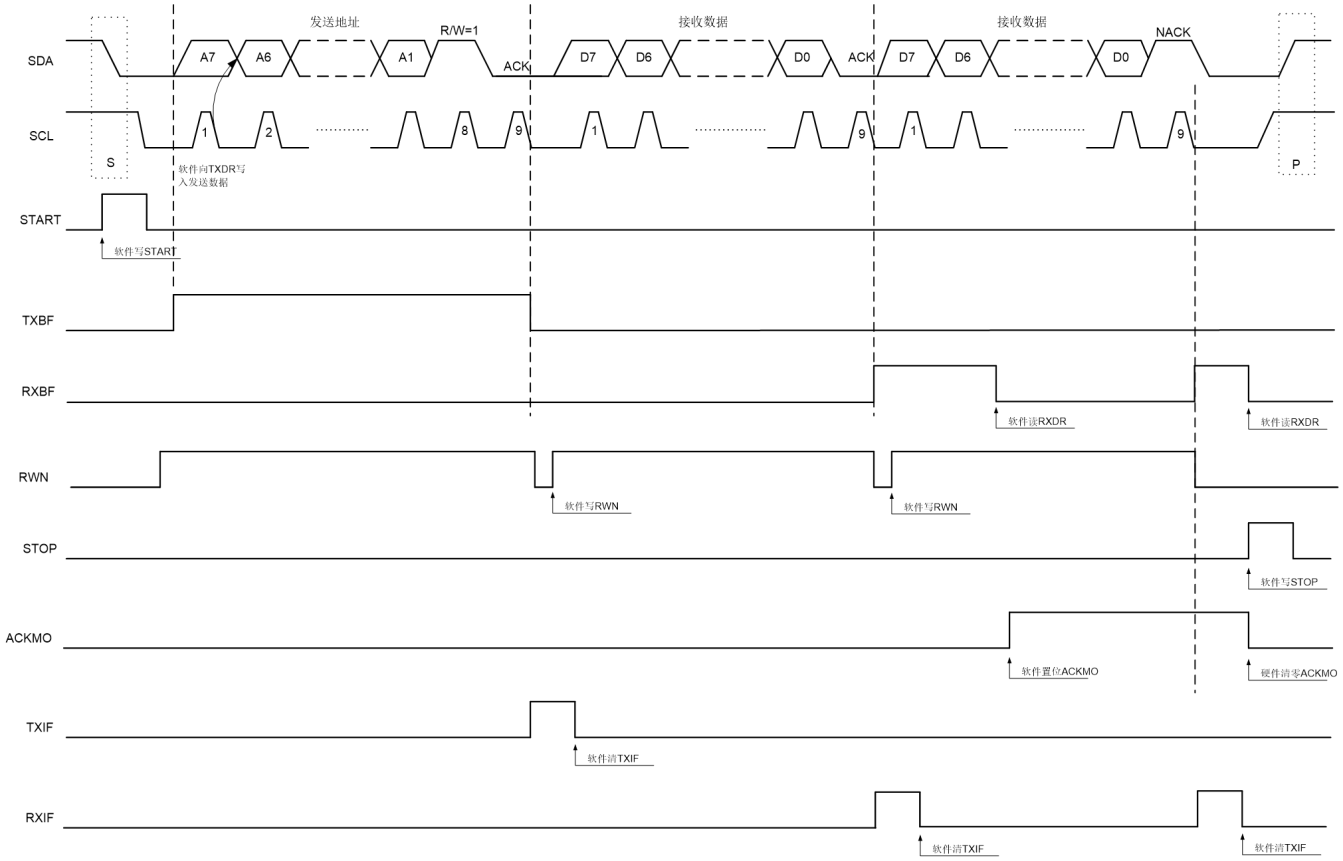


图 18-12 I²C 从 7 位地址从机读取数据流图

在主机连续接收过程中，软件通过置位RWN寄存器读取从机数据。当软件没有写RWN时，主机会保持等待，避免数据接收溢出。

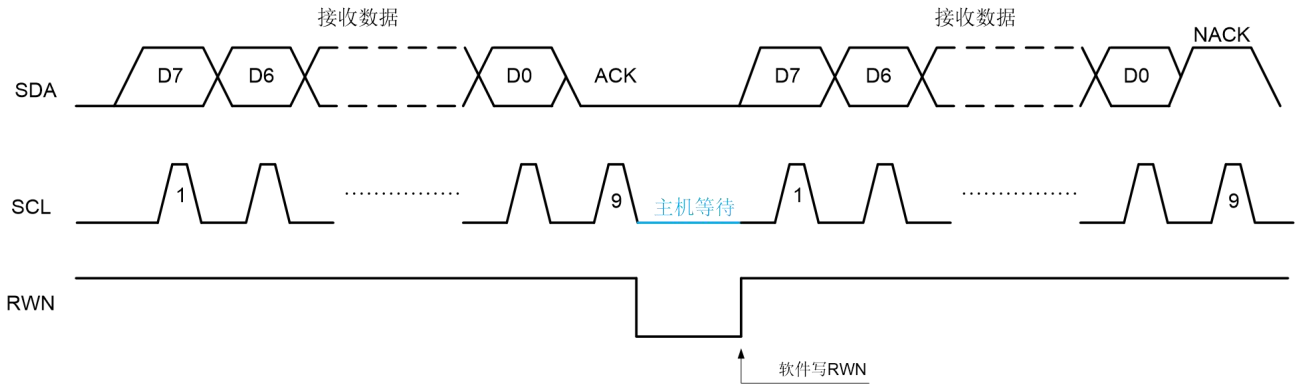


图 18-13 I²C 主机接收数据等待

双向数据传输（组合模式）

典型的双向数据读写流图如下图所示。在主机发送或读取数据过程中，主机可以通过发送 Repeated Start 时序来重新启动一次新的发送或读取通信，所以主机在一次通信中，即可以有数据发送也可以有数据读取。

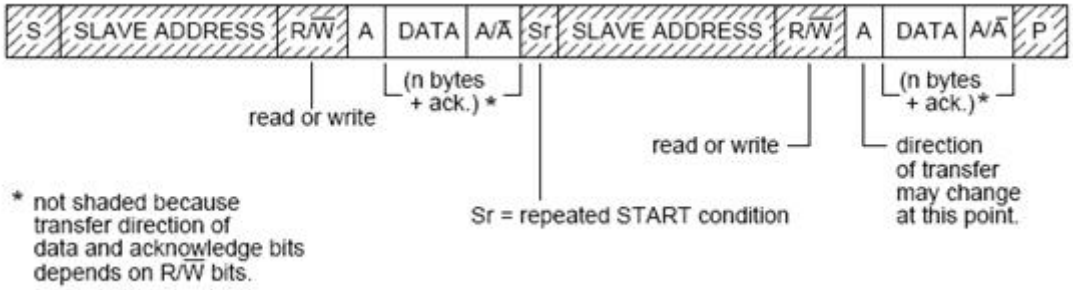


图 18-14 双向数据通信帧格式

组合传输的软件操作流程与单向传输类似，只是在某个字节收发完成后，通过发送ReSTART时序和从机地址字节来修改传输方向。

18.6.3.2 10bit 寻址和数据通信流程

在10bit寻址时，主机发送的第一个字节包含部分从机地址（11110_A9_A8）和传输方向位（$R/\overline{W}$），第二个字节包含剩余从机地址（A7~A0）。两个字节地址发送完成后，再进行数据传输。

主机向从机写入数据

典型的10bit寻址，主机向从机写入数据的数据流图如下图所示。

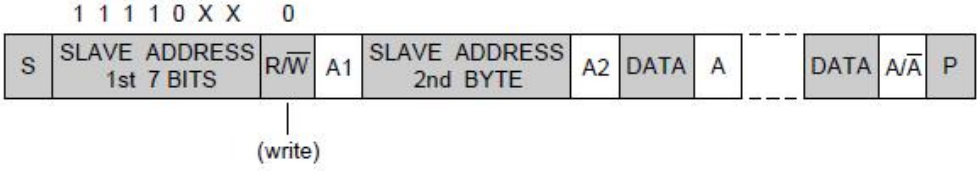


图 18-15 10bit 寻址，主机向从机写入数据

1、主机清零 RWN 寄存器并发起 START 时序

2、主机发送第一个从机地址字节，以 11110 开头，跟随 2bit 从机地址最高位，以及 R/W 标志位，发送数据时 R/W 位为 0

3、主机检查从机回发的 ACK

4、主机发送第二个从机地址字节，包含从机地址的低 8 位

5、主机检查从机回发的 ACK

6、主机继续向从机写入数据

7、主机完成所有数据发送后，发送 STOP 时序

软件启动 I²C 主机发送的操作流程如下图：

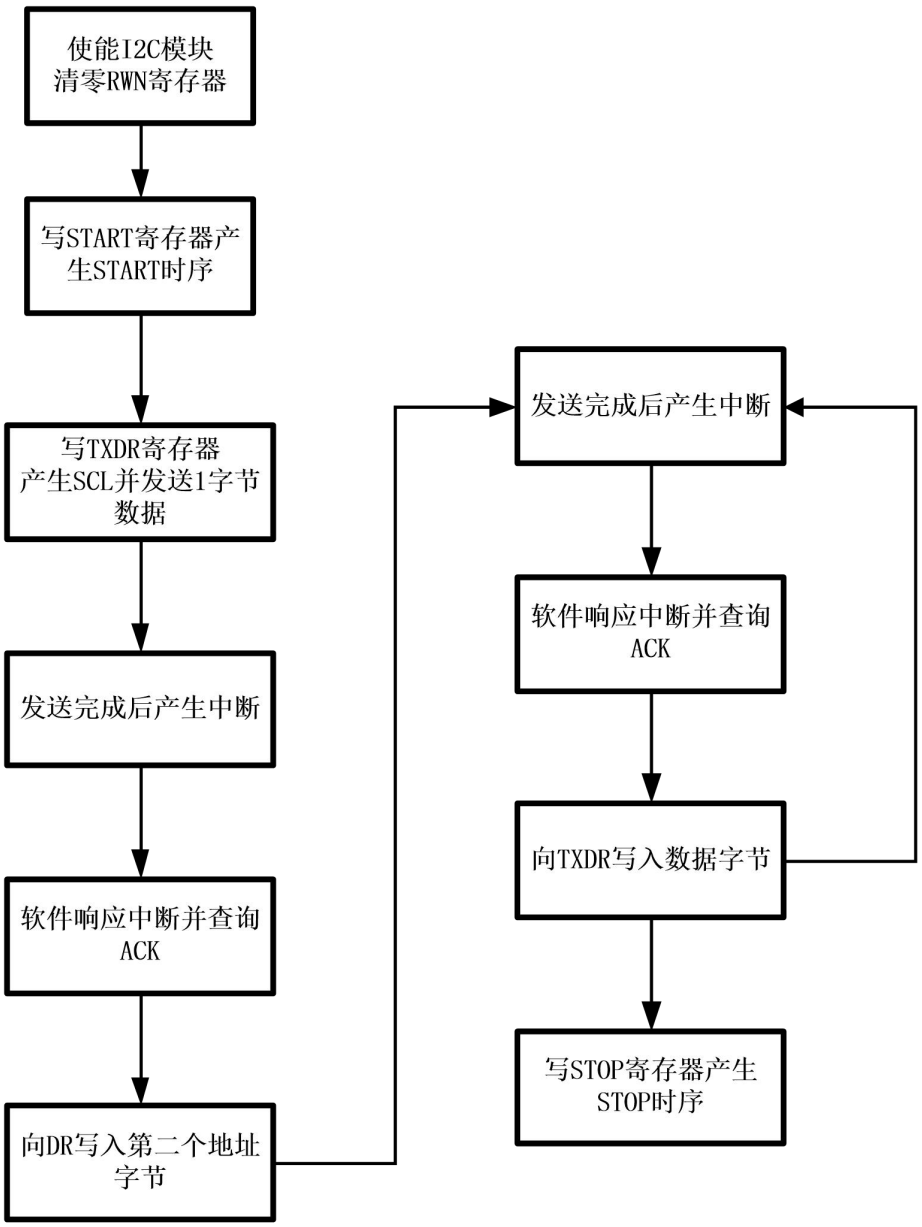


图 18-16I²C 软件发送数据流图

主机从从机读取数据

典型的10bit寻址，主机从从机读取数据的数据流图如下图所示。

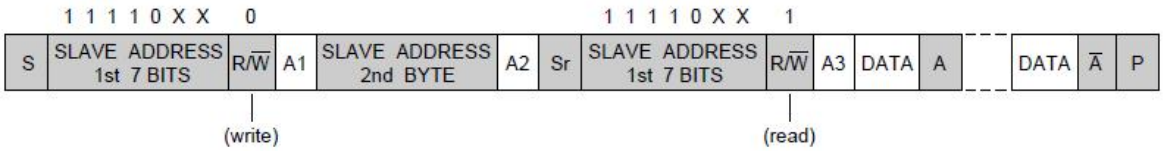


图 18-17 10bit 寻址，主机从从机读取数据

1、 主机清零 RWN 寄存器并发起 START 时序

2、 主机发送第一字节从机地址，包含 5 位前导码 11110、2 位从机地址最高位和 1 位 R/W 标志位

3、 主机发送第二字节从机地址，包含低 8 位地址

4、 主机发送 ReSTART 时序

5、 置位 RWN 寄存器

6、 主机再次发送第一字节从机地址，将 R/W 为改为 1

7、 主机自动转为接收状态

8、 主机开始接收第一字节 8 位数据，并在第 9 个 SCL 向从机发送有效 ACK,从而继续读取下一字节 8 位数据

9、 主机读取最后一个字节后，在第 9 个 SCL 向从机发送 NACK

10、 主机发送 STOP 时序终止读取

软件启动 I^2C 接收的操作流程如下图：

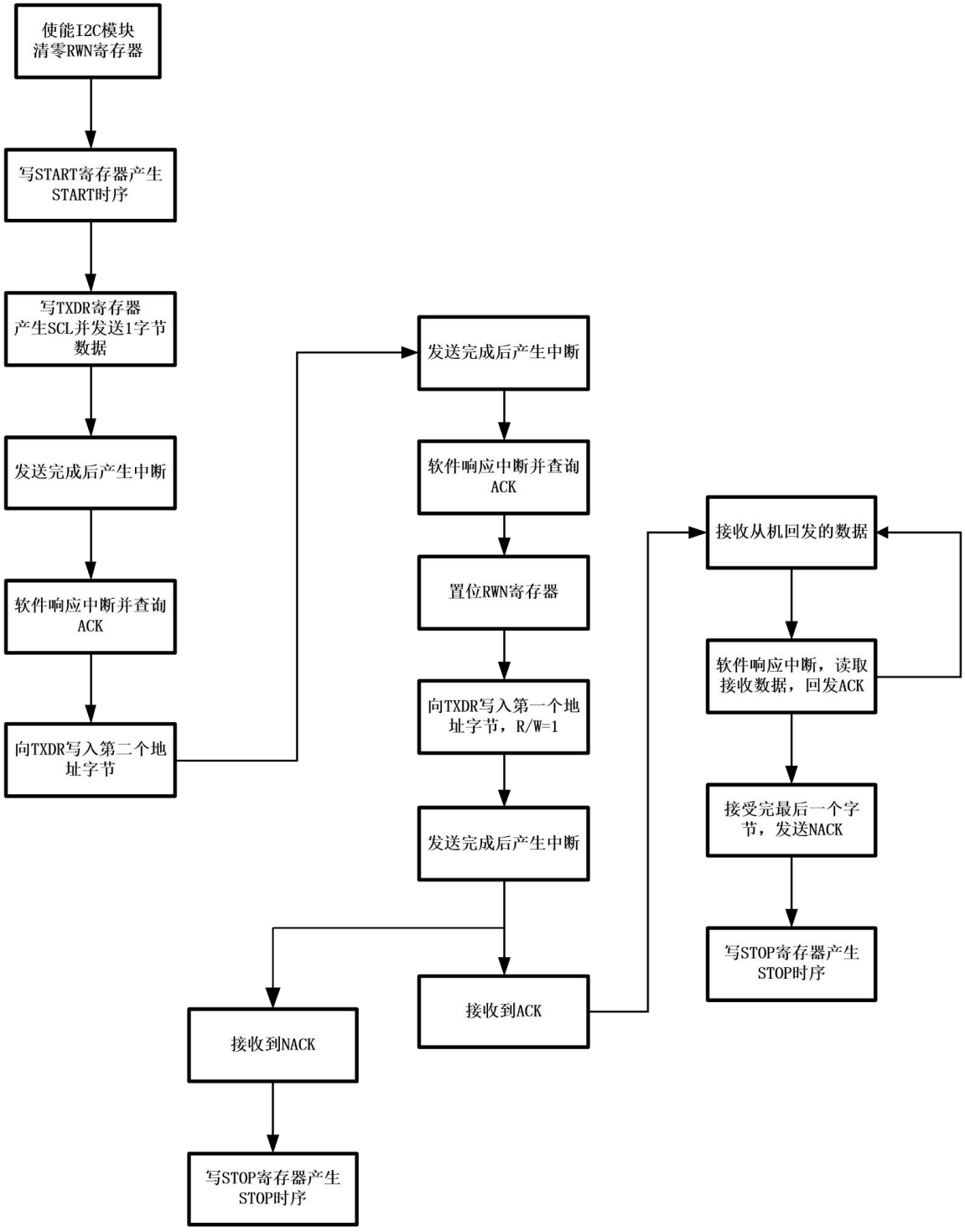


图 18-18 I²C 软件发送数据流图

双向数据传输（组合模式）

典型的双向数据读写流图如下图所示。在主机发送或读取数据过程中，主机可以通过发送 Repeated Start 时序来重新启动一次新的发送或读取通信，所以主机在一次通信中，即可以有数据发送也可以有数据读取。

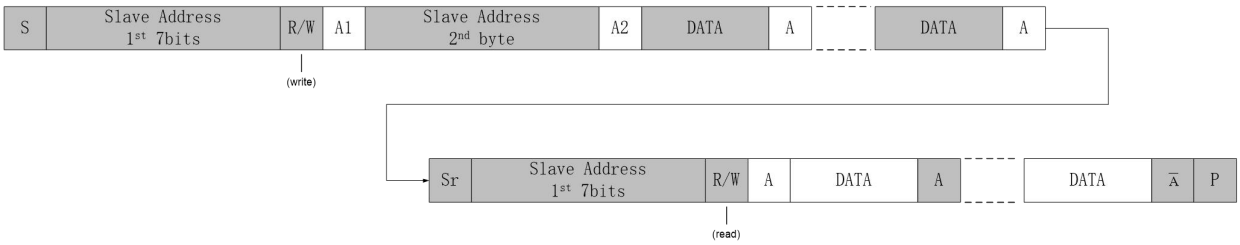


图 18-19 I²C 软件发送数据流图

组合传输的软件操作流程与单向传输类似，只是在某个字节收发完成后，通过发送ReSTART时序和1st从机地址字节来修改传输方向。

18.6.3.3 DMA

I²C 主机支持 DMA，需要注意的是，必须在 I²C 模块的总线时钟（APBCLK）使能的情况下，才能使用 DMA 功能。

主机使用DMA向从机写入数据

主机使用 DMA 发送数据时，包括从机地址字节和发送数据在内的所有数据都需要事先写入 RAM 中，并通过 DMA 请求发送出去。软件应事先将目标 DMA 通道配置为 I2C_TX。

在 DMAEN=1 的情况下，START 置位，如果数据缓存寄存器 TXDR 为空，I²C 模块将产生 DMA 请求，DMA 模块响应请求后将 RAM 中的待发数据写入 TXDR，开始数据发送（第一个字节是从机地址）。DMA 发送模式下，I²C 并不检查发送数据的合法性，软件必须保证 RAM 中的数据是正确的。每个字节发送完成后，I²C 检查从机 ACK，如果 ACK 正确则产生新的 DMA 请求，如果收到 NACK 则产生 NACK 中断，并不再产生 DMA 请求。

当 DMA 完成指定长度的数据发送后，产生 DMA 传输完成中断，此时可以由软件置位 STOP 寄存器产生 STOP 时序，也可以由 I²C 硬件根据 DMA 传输完成信号自动置位 STOP 寄存器产生 STOP 时序。可以通过设置 AUTOEND 寄存器来选择所需的策略。

主机使用 DMA 进行发送的流程如下图：

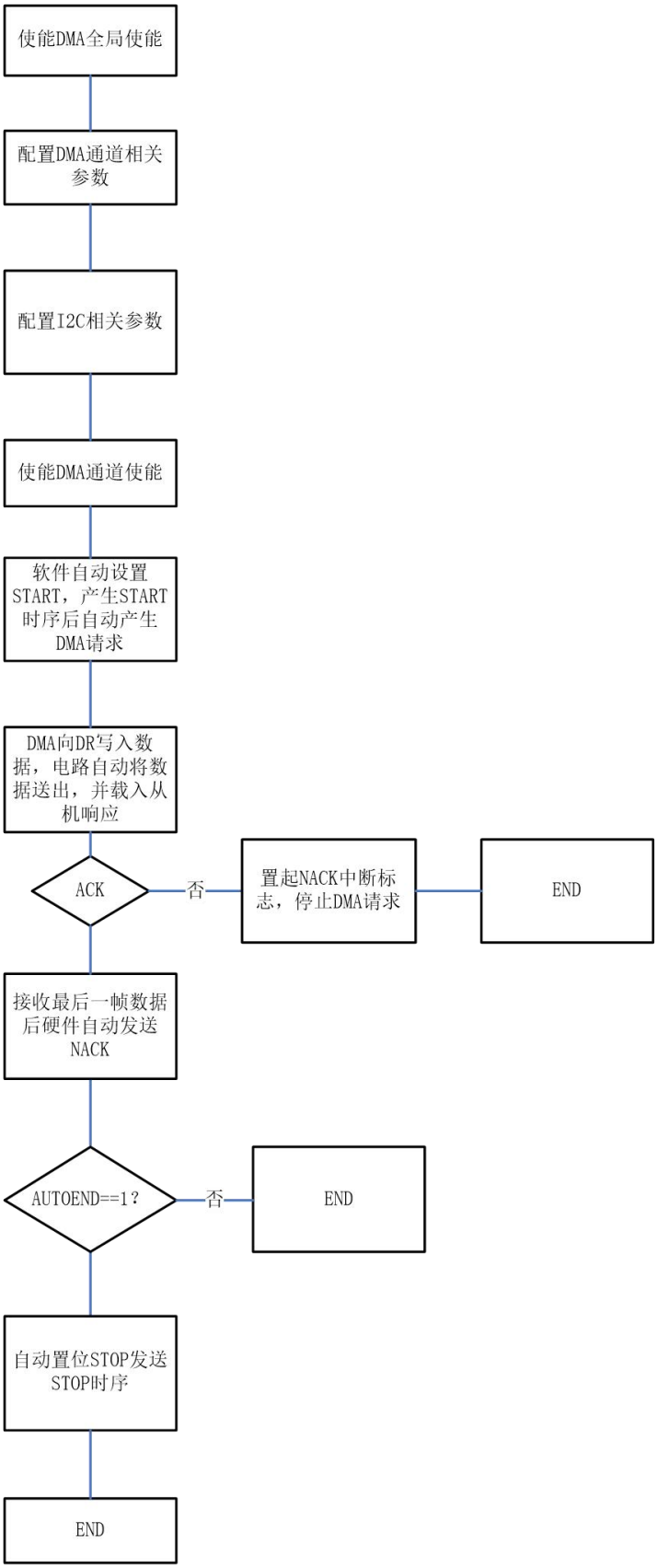


图 18-20 I²C 主机 DMA 发送流程图

主机使用DMA从从机读取数据

这种场景下，从机寻址字节必须由软件发送。软件应事先将目标 DMA 通道配置为 I2C_RX。

软件首先发送完从机地址后，设置 DMAEN=1，然后使能对应的 DMA 通道为 I2C_RX，I²C 自动进入接收模式，并在每个字节接收完成后产生 DMA 请求，通知 DMA 来读取 RXDR 内容，同时向从机回发 ACK。

当 DMA 传输达到指定长度后，DMA 的传输完成标志将通知 I²C 回发 NACK。随后根据 AUTOEND 寄存器配置，可以由软件或硬件置位 STOP 寄存器产生 STOP 时序。

注意：当I2C主机通过DMA进行数据接收时，在不同AUTOEND配置和相同DMA传输长度（CHxTSIZE）配置下，DMA接收字节数会有差别。当AUTOEND=0时，接收字节数为CHxTSIZE+1；当AUTOEND=1时，接收字节数为CHxTSIZE。

主机使用DMA进行接收的流程如下图：

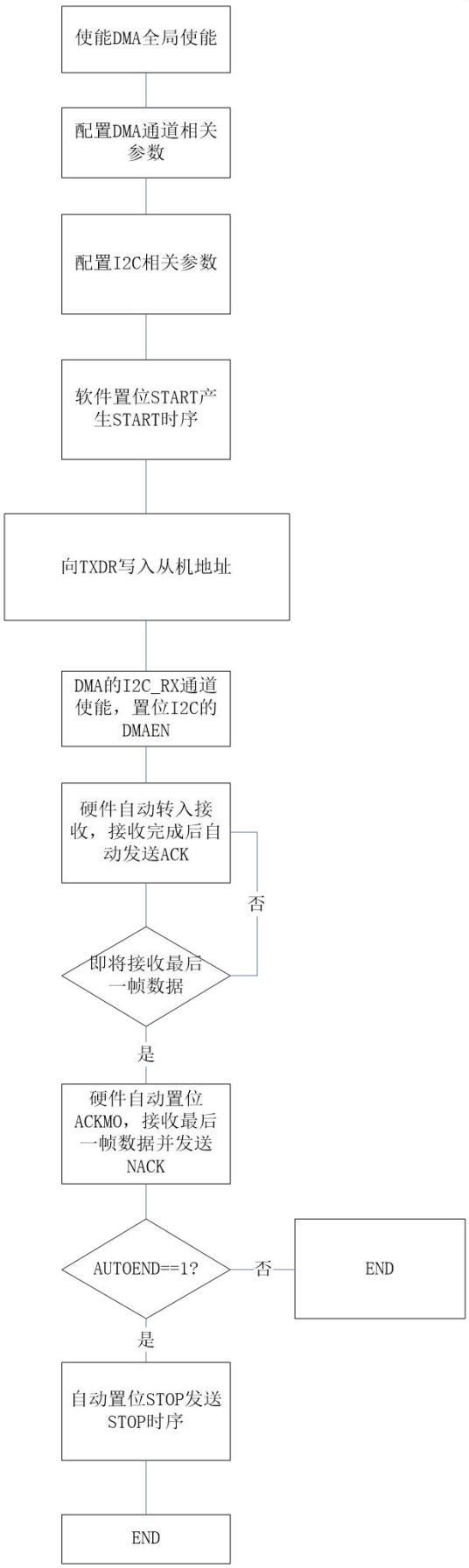


图 18-21 I²C 主机 DMA 接收流程图

18.6.3.4 时钟延展（Slave Clock Stretching）

I²C 总线运行低速从机通过拉低 SCL 的方式通知主机暂停数据通信。I²C 主机必须支持这一特性，因此在每个字节收发起始位置处，主机在尝试发送 SCL 高电平后，需要自动检查总线上 SCL 的实际电平，如果不是高电平，意味着从机正在进行 SCL 延展，主机会持续监控 SCL 电平，直到 SCL 为高，才开始后续操作。

注意：主机只在每字节收发的第一个 SCL 上升沿处进行 SCL 延展检查。如果使能了模拟滤波或数字滤波，则主机检查的 SCL 为总线输入经过滤波后的信号。

18.6.3.5 多主机时钟同步（Clock Synchronization）

时钟同步是I²C协议针对多主机通信的一种同步机制。当多个主机在空闲总线上同时开始传输数据时，时钟同步机制保证了总线时钟不会出现冲突状态。

时钟同步采用线与机制，当多个主机拉低SCL时，SCL低电平时间较短的主机在试图拉高SCL时会发现其他主机仍在驱动SCL低电平，此时主机应保持无驱动的等待状态直到SCL总线被拉高，然后再开始计时SCL高电平宽度。采用线与的方式，总线上的同步时钟的低电平总是由SCL低电平长度最长的主机决定，而SCL高电平则由高电平长度最短的主机决定。

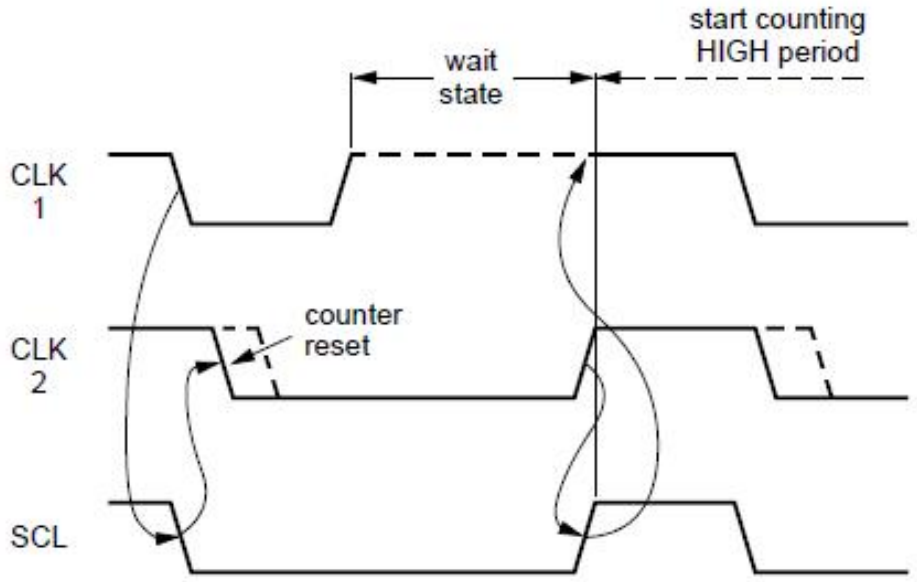


图 18-22 I²C 多主机时钟同步

18.6.3.6 多主机仲裁（Arbitration）

总线仲裁同样仅适用于多主机总线。当多个主机在START条件最小保持时间（$t_{HD,STA}$）内同时发送START条件时，必须通过仲裁机制避免总线冲突并保证数据通信的完整性。

仲裁是基于每个数据bit进行的，当SCL为高电平时，每个主机应实时检测总线上的SDA信号是否与自己发送的数据电平相符。注意到总线的线与特点，因此数据冲突只会在本机发送1，而SDA总线为0的情况下被检测到。此时本主机发现自己输掉仲裁，应主动关闭SDA输出驱动逻辑，便于其他主机继续正常通信，如果模块同时使能了从机模式，模块会立即进入从机接收状态。

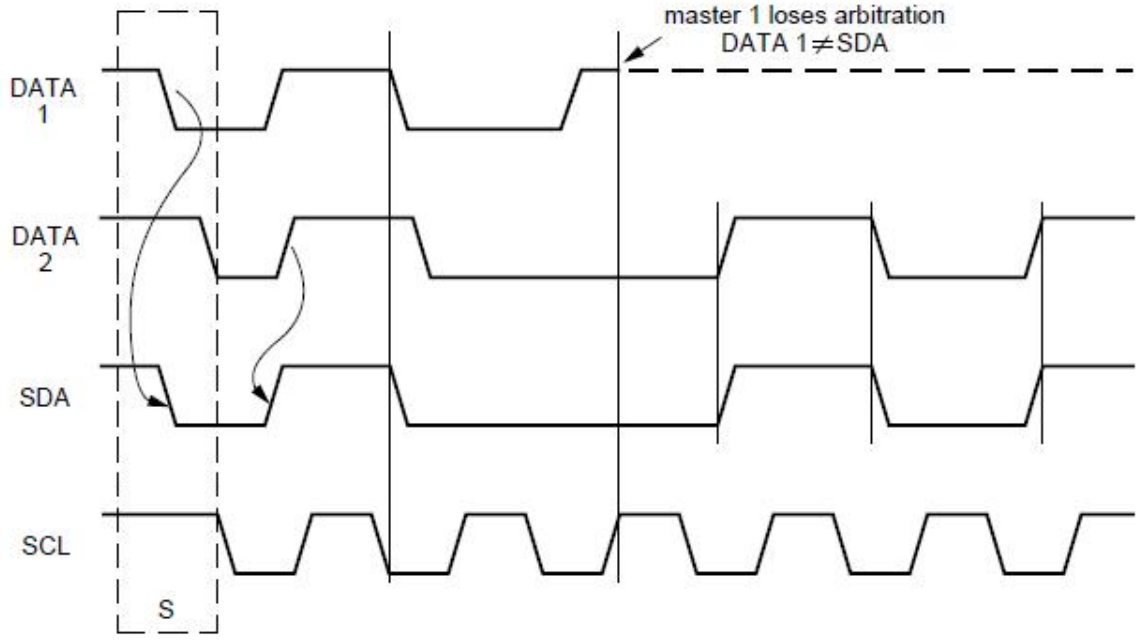


图 18-23 I²C 多主机数据仲裁

18.6.3.7 超时机制

I²C主机还实现了超时机制，即发现从机长时间拉低SCL导致总线无法通信的情况下，产生超时报警中断并返回IDLE状态。

当主机检测到SCL延展，其内部定时器开始计时，主机设定的SCL延展超时的时长最长是4096个SCL周期，假设波特率为100K，则超时周期大约是40ms，如果波特率是400K，则超时周期大约是10ms。通过12bit的TIMEOUT寄存器，软件可以设置超时周期。软件必须在MSPEN为0的情况下设置TIMEOUT寄存器，此寄存复位值为0xFFF，即表示最长4096*T_{SCL}的超时周期，当检测到SCL延展后，TIMEOUT寄存器开始向下递减，当计数到0时，计数停止，TIMEOUT寄存器被复位到0xFFF，同时触发超时中断。因此通过修改TIMEOUT初始值，可以设定超时周期。

$T_{SCL_STRETCHING_TIMEOUT}$=TIMEOUT[11:0] * T_{SCL}

当发生TIMEOUT中断时，建议软件复位I²C模块。

此功能可以被关闭，如果关闭硬件超时，软件也可以通过定时器结合SCL引脚状态判断来自行实现任意长度的超时判决。

18.6.3.8 可编程时序

I²C 模块的主机模式提供了灵活的时序编程特性，允许用户定义 SCL 时钟的低电平宽度、高电平宽度，SDA 数据的建立和保持时间。

通过 BRGH/BRGL 寄存器可以设置 SCL 的低电平和高电平宽度，通过 SDAHD 寄存器可以配置 SDA 数据相对 SCL 时钟脉冲的保持和建立保持时间长度。

SDAHD 时序对于 SDA 数据位和 ACK/NACK 都同样有效。

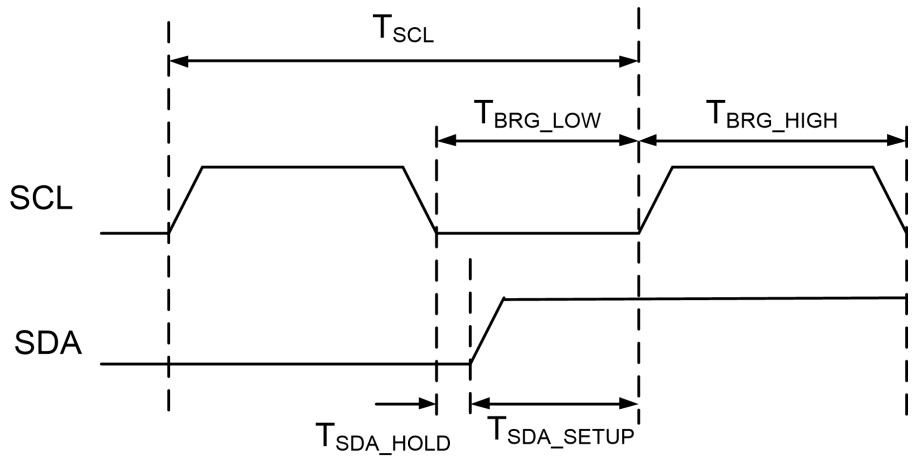


图 18-24 主机时序控制

上图中，T_{SCL}为通信波特率，各个参数可以由以下公式表达：

$T_{SCL} = T_{BRG_LOW} + T_{BRG_HIGH}$

$T_{SDA_SETUP} = T_{BRG_LOW} - T_{SDA_HOLD}$

注意，应用中对BGRH、BRGL和SDAHD寄存器的配置必须满足以下要求，如果违反这些要求将导致异常的总线时序：

BRGH >= 2

BRGL >= 2

BRGL-1 >= SDAHD >= 1

TIMEOUT >= 1

18.6.4 I2C 从机功能

从机接收完 1 字节数据后，产生中断通知 CPU 处理数据，在 CPU 取走数据前硬件可以将 SCL 拉低（软件可以控制是否使能 SCL 延展功能），通知发送方正忙，发送方应暂停发送直到 SCL 放开。若接收方无法响应 ACK，发送方检测 ACK 失败后应发送 P 终止通信或者发送 Sr 开始新的通信。

从机发送完 1 字节数据后，产生中断通知 CPU，硬件拉低 SCL 令主机等待，CPU 响应中断并准备好下一字节数据后再放开 SCL，主机继续发送 SCL 使从机继续数据发送。

18.6.4.1 从机寻址

根据 A10EN 寄存器状态，从机可以支持 7bit 或者 10bit 寻址过程。从机地址由 SLAVE_ADDR 寄存器定义。

对于 10bit 从机地址应用，即 A10EN=1 的情况下，要求第一个字节必须以 11110 开头，否则将触发 ADDR_ERROR 错误标志。而在 A10EN=0 的情况下，如果从机收到了 11110 开头的地址字节，也会置位 ADDR_ERROR 错误标志。

18.6.4.2 从机输入滤波

从机 SCL 和 SDA 输入支持模拟滤波和数字滤波。在使用数字滤波时，必须使能 I2CCLK。如果仅使用模拟滤波，则可以不需要 I2CCLK 工作。

在休眠模式下如果希望 I^2C 从机继续工作，则不能使能数字滤波功能，否则由于数字滤波电路无法工作将导致 I^2C 模块无法检测总线电平变化。

18.6.4.3 从机发送数据

推荐操作流程：

- 从机接收到地址字节（R/W=1），回发ACK，产生地址匹配中断
- 由于R/W=1，硬件自动进行SCL延展，从机进入发送状态
- 软件响应中断，查询R/W标志，确认是从机发送
- 软件将待发送数据写入TXDR
- 硬件自动释放SCL
- 新的SCL到来，SDA开始输出
- 接收ACK并产生发送完成中断

- 重复数据发送过程直到接收到STOP时序，或接收到主机NACK

下图是一个典型的从机数据发送波形示意图：

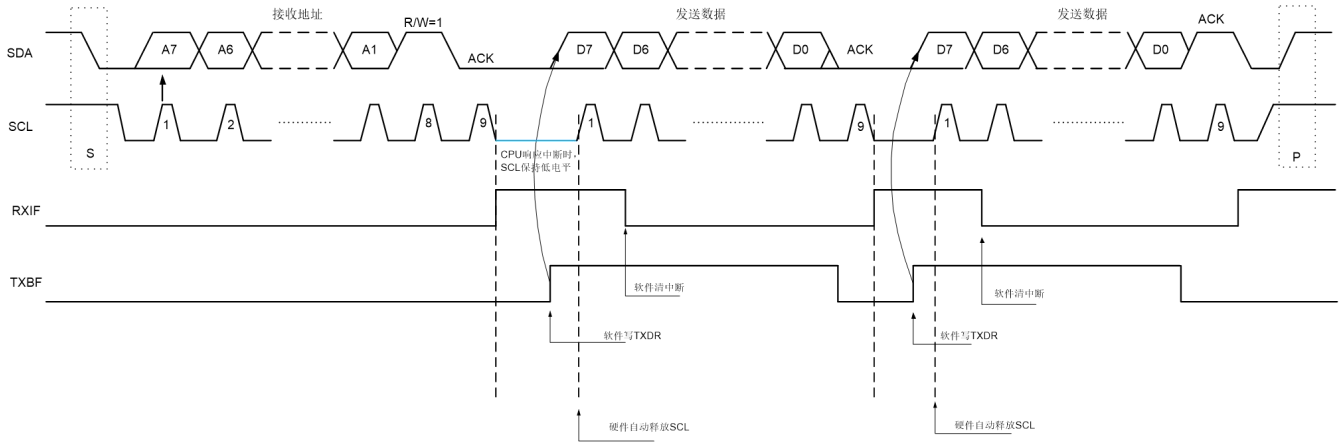


图 18-25 I²C 从机数据发送波形

在从机发送流程中，当从机收到正确地址时，ADM标志置位，地址字节会被写入RXDR。硬件自动拉低SCL信号等待软件写入TXDR，当软件写TXDR后TXBF置位，同时硬件释放SCL。

18.6.4.4 从机接收数据

推荐操作流程：

- 从机接收到地址字节（R/W=0），回发ACK，产生地址匹配中断
- 由于R/W=0，硬件自动进行SCL延展，从机保持接收状态
- 软件响应中断，查询R/W标志，确认是从机接收
- 软件读RXDR硬件自动释放SCL，开始接收数据
- 主机数据字节到来，字节接收完成后硬件置位RXBF标志
- 从机回发ACK，并产生接收完成中断
- 硬件自动进行SCL延展（SCLSEN=1）
- 软件响应中断，读取RXDR，硬件自动清零RXBF标志
- 硬件自动释放SCL
- 重复数据接收过程直到接收到STOP时序，或者软件将ACKEN置0

下图是一个典型的从机数据接收波形示意图（SCLSEN=1）：

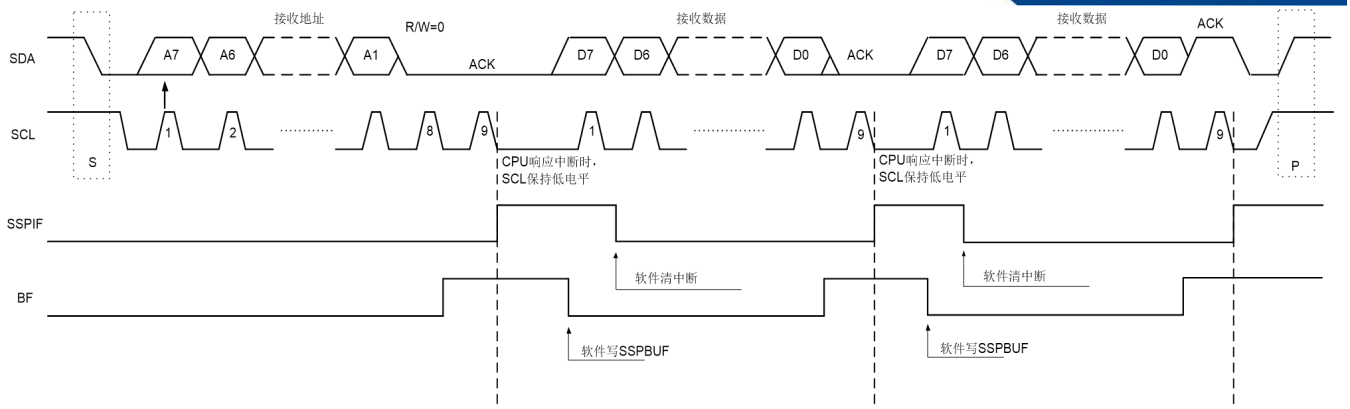


图18-26 I²C从机数据接收波形

从机接收过程中，从机首先收到地址字节，地址匹配的情况下，ADM标志置位，地址字节将被写入RXDR并置位RXBF标志，然后硬件拉低SCL。当软件读取RXDR后，RXBF标志自动清零，硬件释放SCL，可以进行后续数据接收。

注意：从机接收流程中地址字节会被写入RXDR并导致RXBF置位，软件需要读取RXDR来清零RXBF并释放SCL。

从机接收数据可以被动结束通信或主动结束通信。

如果主机主动下发STOP，则从机被动结束本次通信。或者，软件在中断处理程序中将ACKEN寄存器清零，则从机在接收完下一个字节后，将回发NACK，主机接收NACK后将下发STOP结束本次通信。

从机SCL延展

I²C从机默认使能SCL延展（slave clock stretching），但是软件可以关闭这个功能（SCLSEN寄存器）以适应不支持从机SCL延展的主机。

当SCL延展使能的情况下，数据接收完成后，软件只有在SCL延展期间读取接收缓冲区时，才能清零RXBF标志。如果接收中出现了数据溢出，RXOV标志置位，此时硬件回发NACK，并且SCL不再被延展，以便主机下发STOP；RXOV置位的情况下，建议软件等待STOP标志置位，再读取接收缓冲区清零BF标志。

接收数据溢出

当从机接收缓冲区满（BF=1）时，如果又收到新的数据，则发生接收溢出，RXOV标志置位。接收缓冲区中的老数据将被新的数据覆盖。只有在从机关闭了SCL延展功能的情况下，才有可能发生接收数据溢出。

下图是SCLSEN=0情况下发生数据接收溢出的示意图：

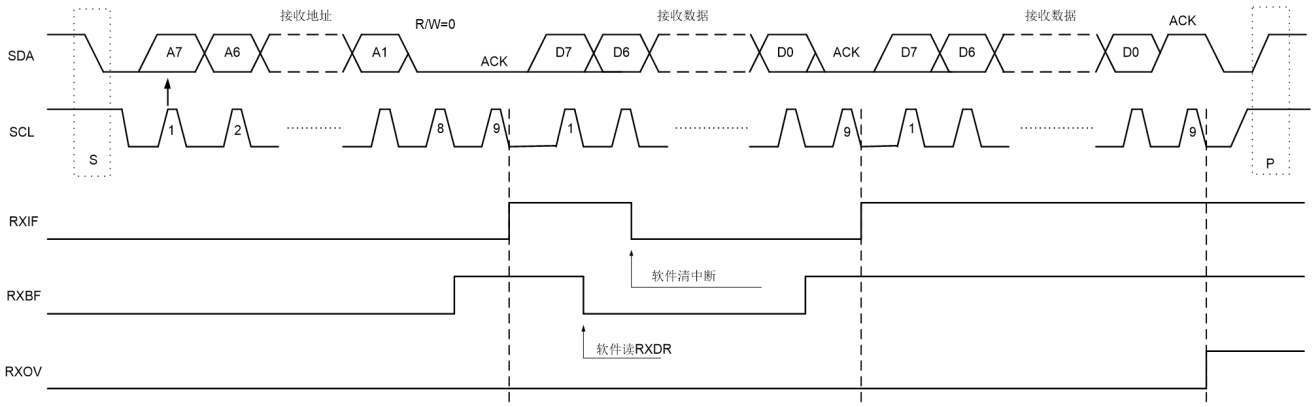


图18-27 I²C从机数据接收波形（SCLSEN=0，接收溢出）

18.6.4.5 从机低功耗接收唤醒

当芯片处于休眠模式时，I²C从机支持START时序唤醒和地址匹配唤醒。

- START时序唤醒RCHF时钟用于从机地址接收
- 从机地址匹配的情况下唤醒MCU进行后续通信

软件也可以选择在检测到START时序时直接唤醒MCU。如果配置为START仅唤醒RCHF时钟，则从机检测到START事件后自动打开RCHF，以便模块能够完成地址字节接收和超时检测。

如果时钟唤醒后发现地址不匹配，则I²C从机不会唤醒MCU，而是自动关闭RCHF。

注意：使用I²C从机休眠唤醒功能时，必须将I2CCLK配置为RCHF，并关闭数字滤波功能。

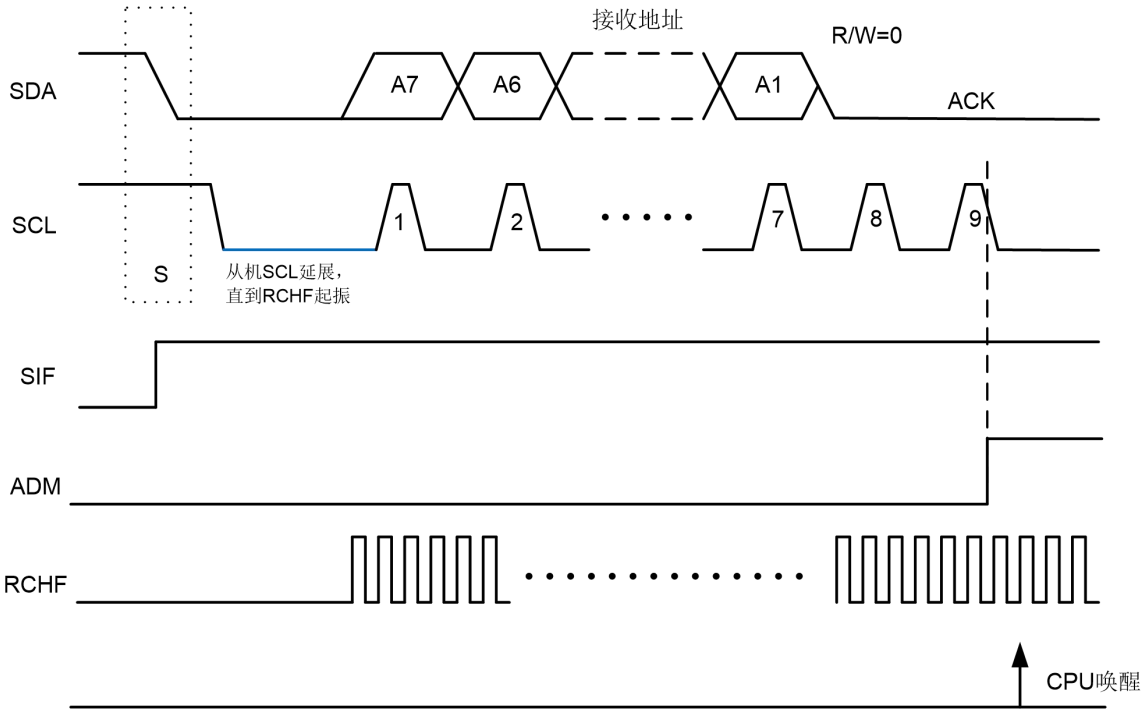


图 18-28 从机地址匹配唤醒

18.6.4.6 DMA

I²C 从机支持 DMA 操作，需要注意的是，必须在 I²C 的总线时钟（APBCLK）使能的情况下才能进行 DMA 操作。总线时钟被用于产生 DMA 请求并接收 DMA 应答。

从机使用DMA接收数据

当 I²C 从机接收到正确的地址后，产生 ADM 中断标志，软件响应中断后，查询接收到的 R/W 位，如果为 0 表示主机准备向从机写入数据。此时软件可以配置特定 DMA 通道为 I2C_RX，并使能 I²C 从机的 DMAEN；随后每次从机完成一个字节的接收，将产生 DMA 请求，通知 DMA 来读取 RXDR。结束 DMA 从机接收有两种可能：

1） 数据传输长度还未达到 DMA 长度配置，主机就下发了 STOP 时序，软件应响应 STOP 中断并主动处理这种情况；

2） 数据传输长度达到 DMA 长度配置，但是由于 DMA 请求是在从机回发 ACK 后产生，所以软件应响应 DMA 传输完成中断，并将 ACKEN 清零，这样从机会在接收完下一个字节后，回发 NACK，结束本次通信。

从机使用 DMA 进行接收的流程如下图：

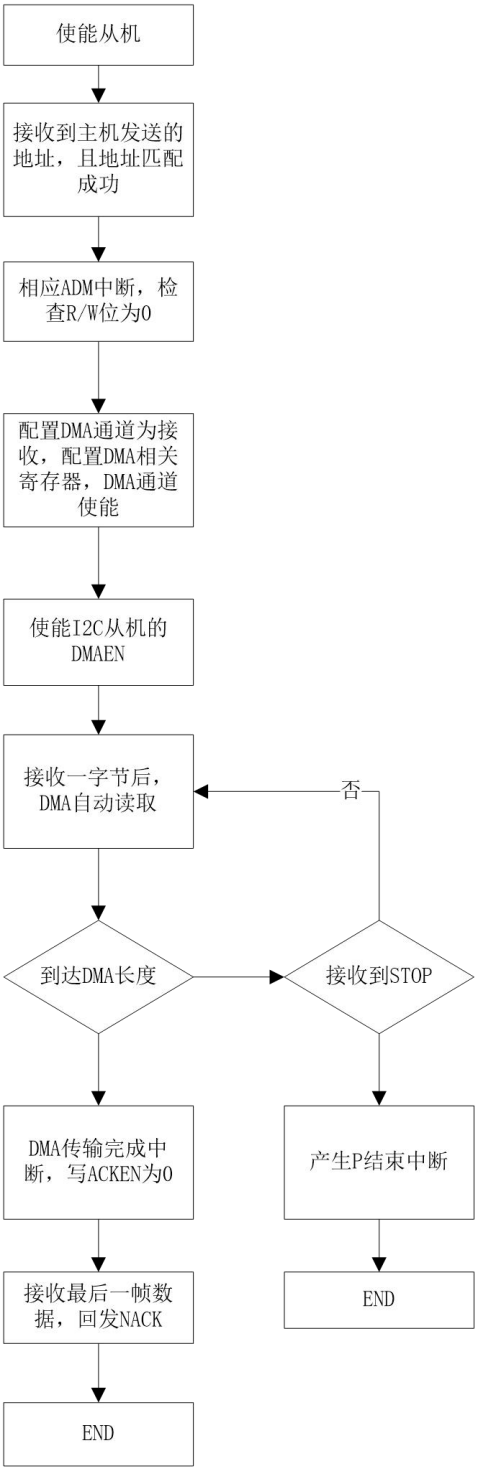


图18-29 I²C从机DMA接收流程图

从机使用DMA发送数据

当 I²C 从机接收到正确的地址后，产生 ADM 中断标志，软件响应中断后，查询接收到的 R/W 位，如果为 1 表示主机准备从从机读出数据。此时软件需要配置特定 DMA 通道为 I2C_TX，并使能 I2C 从机的 DMAEN；随后当从机数据缓存 TXDR 为空时，将产生 DMA 请求，通知 DMA 写入 TXDR。

只有主机回发 NACK 才能结束读取操作。当读取数据长度大于 DMA 设置的传输长度时，由于 DMA 不再响应 I²C 请求，从机将一直拉低 SCL，直到软件关闭 I²C 从机模块。

从机使用 DMA 进行发送的流程如下图：

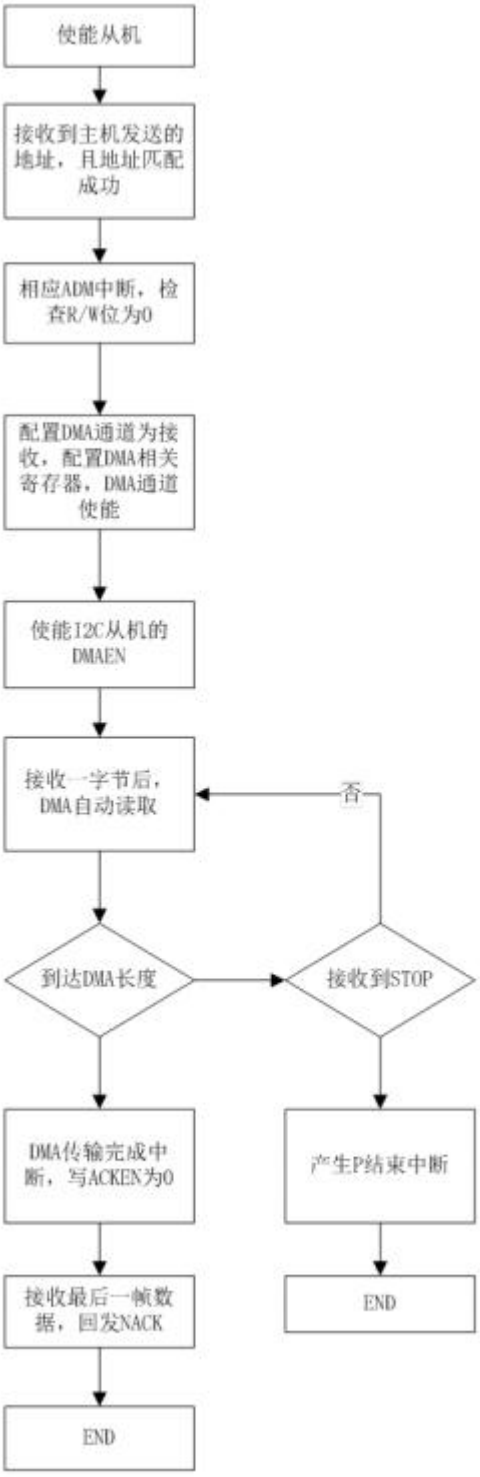


图18-30 I2C从机DMA发送流程图

18.7 SMBus 功能描述

18.7.1 概述

SMBus是基于I²C物理层的双线同步通信协议，可用于系统控制和电源管理任务，支持动态地址分配以实现热插拔应用。本模块支持SMBus rev3.0，更多关于SMBus协议的详情请参见（http://smbus.org）

SMBus规定了三种类型的设备：

- 从机：接收主机发送的命令并给出响应
- 主机：发送命令和时钟，能够启动或终止通信
- Host：一种特殊的主机，能够提供面向系统CPU的接口。Host必须能够作为主机和从机，并支持host notify protocol。整个系统中只允许存在一个host。

本模块可以支持从机、主机、host模式。

SMBus协议层是一个基于命令的操作方式，参见SMBus Specification了解协议定义的命令集。

Address resolution protocol (ARP)

SMBus通过ARP协议可以实现对从机的动态地址分配，因此可以实现系统“热插拔”。本外设模块支持SMBus默认从机地址（device default address: 1100 001）识别，通过置位SMBDEN寄存器来使能此功能。

为了实现ARP，从机软件需要维护一个128位的SMBus系统唯一ID（UDID）；当从机接收到device default address时，中断标志置位通知软件系统主机正在进行ARP枚举协议，软件应完成ARP协议流程以实现动态从机地址分配。

Host Notify protocol

本模块支持host notify协议，当SMBHEN寄存器置位时，外设能够响应SMBus host address（0001 000），并且能够同时作为系统主机和host从机进行通信。

SMBus Alert

SMBus Alert是一个可选的信号，一个SMBus从机可以通过拉低SMBus Alert来向host申请发起通信。Host接收到SMBALERT#信号后，通过Alert Response Address向所有具备SMBALERT#能力的从

机发起访问。只有拉低SMBALERT#的从机会响应Alert Response Address寻址。

在SMBus从机模式下，当ALERTEN=1的情况下，软件可以通过置位ALERT寄存器拉低SMBA引脚。当从机接收到ARA并识别到自己的地址后，从机应清零ALERT寄存器以释放SMBALERT#信号。

在SMBus Host模式下，如果ALERTEN寄存器置位，当检测到SMBA引脚上的下降沿，ALERT标志寄存器置位并可以产生相应的中断事件。

Packet Error Checking (PEC)

SMBus的数据包检错功能通过CRC-8检错码实现，PEC是SMBus器件的可选功能，但是需要支持ARP的器件必须支持PEC。

检错码被附加在数据通信的最后，CRC-8计算被应用于所有的消息字节，包含地址和RW_位，不包含ACK/NACK、START、STOP、ReSTART。

SMBus从机在接收过程中发现PEC错误时，回发NACK指示PEC错误。不过由于ACK/NACK位本身也有可能出现总线传输错误，主机不应将从机回发ACK视为PEC正确。从机模式下本模块可以支持回发或不回发PEC。

PEC使用的CRC-8多项式：$C(x)=x^8 + x^2 + x^1 + 1$

Timeouts

本模块包含一组硬件定时器用于实现SMBus协议规定的三种超时类型。当模块检测到超时事件时，需要复位状态机，停止数据收发，恢复IO默认无驱动上拉状态，并准备好接收新的START事件。

Timeout只检测时钟信号，当总线时钟被拉低后，超时计数器开始工作。

符号	参数	limits		unit
		min	max	
$t_{TIMEOUT}$	SCL 低电平超时	25	35	ms
$t_{LOW:SEXT}$	累积 SCL 低电平扩展时间（slave）	-	25	ms
$t_{LOW:MEXT}$	累积 SCL 低电平扩展时间（master）	-	10	ms

表 18-3SMBus 超时事件

1. $t_{LOW:SEXT}$是一个从机在一次通信中（START到STOP）允许的时钟拉低累积时间
2. $t_{LOW:MEXT}$是一个主机在每个字节内（START-to-ACK, ACK-to-ACK, ACK-to-STOP）允许的时钟拉低累积时间

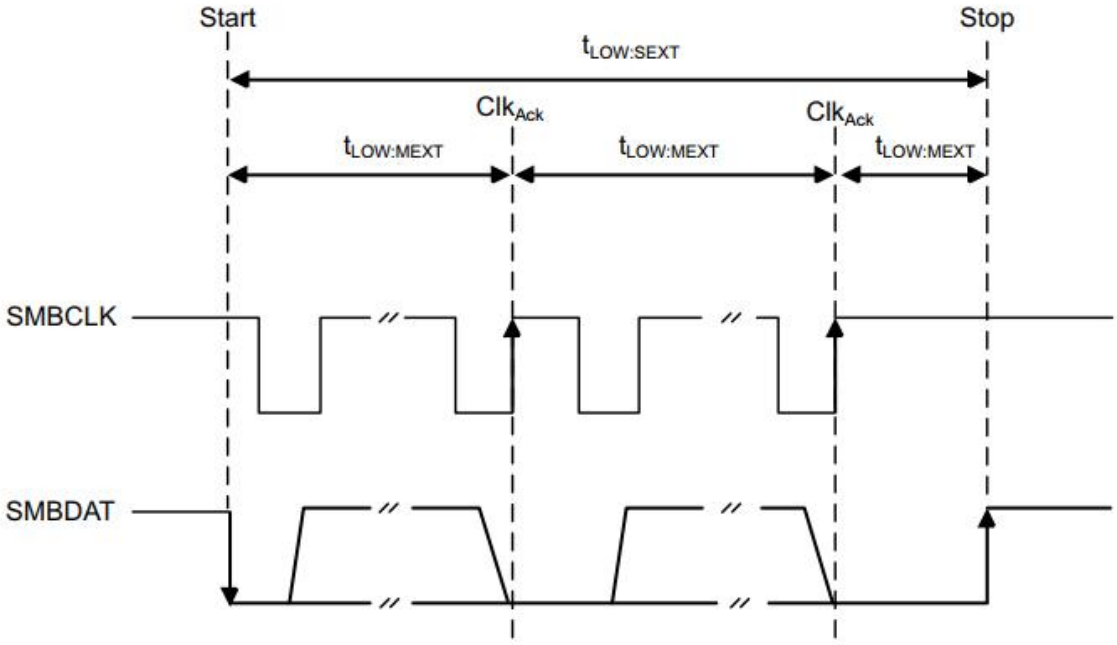


图 18-31$t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 波形

总线IDLE检测

SMBus主机在检测到总线时钟和数据保持高电平超过$t_{HIGH,MAX}$时认为总线已经处于空闲状态，可以发起通信。

本模块支持硬件总线IDLE检测，t_{IDLE}判决时间由TIMEOUT寄存器定义。

18.7.2 SMBus 初始化过程

除了I²C相关的初始化，SMBus特殊的初始化过程包含以下内容。

特殊地址（从机模式）

- SMBus Device Default address (1100 001)通过SMBDEN寄存器使能
- SMBus Host address (0001 000)通过SMBHEN寄存器使能
- Alert Response address (0001 100)通过ALERTEN寄存器使能

Packet Error Checking

PEC校验字节的计算通过模块自带的CRC-8功能实现。

SMBus在收发数据的同时，如果置位了CRCEN，则硬件将接收到的数据bit同步送入CRC-8进行串行移位计算。

在SMBus从机接收模式下，软件应根据PEC比对结果控制PEC字节对应的ACK或者NACK回发。

Timeout检测

超时检测使用TIMEOUTx硬件定时器实现。其中TIMEOUTA是一个12bit定时器，在TIDLE寄存器清零的情况下，TIMEOUTA用于总线SCL低电平超时检测。当SCL拉低时间超过(TIMEOUTA+1)×2048×t_{I2CCLK}时，总线超时标志寄存器被硬件置位，并可以产生中断事件。

另一个TIMEOUTB 12bit定时器用于$t_{LOW:SEXT}$和$t_{LOW:MEXT}$超时检测。当累积SCL延展时间超过(TIMEOUTB+1)×2048×t_{I2CCLK}时，时钟延展超时标志寄存器被硬件置位，并可以产生中断事件。

Bus IDLE检测

当TIDLE寄存器置位时，TIMEOUTA用于总线IDLE检测。如果总线上的SCL和SDA都保持高电平超过(TIMEOUTA+1)×4×t_{I2CCLK}，总线空闲标志寄存器被硬件置位，并可以产生中断事件。

18.7.3 SMBus 主机功能

18.7.3.1 SMBus 主机发送

SMBus主机发送数据可以通过软件或者DMA方式进行。

为了支持PEC功能，I2C_SMBUS模块自带CRC-8校验单元，在使能CRC的情况下（CRCEN=1），CRC-8单元自动完成对发送数据的CRC计算。软件需要在数据发送完成后，从CRCDR读取发送数据的CRC结果并发送出去。

DMA发送时，有两种操作方法：

- 关闭AUTOEND功能，在DMA发送完全部数据字节后，再由软件从CRCDR读取PEC字节并发送出去
- 预先对要发送的数据块预计算CRC结果，紧跟着保存在数据块后面；DMA发送时可以使能AUTOEND，发送长度为数据块长度+1字节CRC

18.7.3.2 SMBus 主机接收

SMBus主机接收数据可以通过软件或者DMA方式进行。

使用软件接收数据时，工作流程与I^2C主机接收类似，主要区别在于PEC功能。接收过程中的CRC计算有两种方案：

- CRCEN=0：软件处理CRC计算，每收到1个字节数据，软件负责将数据写入外设CRC模块进行计算。完成所有数据接收后，软件读取CRC计算结果，与最后收到的PEC字节比较。
- CRCEN=1：SMBus模块内部自动进行CRC计算，调用模块内部的CRC-8单元，每收到1字节数据，同步更新CRC-8结果

对于CRCEN=1的接收模式，由于硬件无法自动区分数据字节和CRC字节，软件应在收完最后一个字节数据，并且在开始接收CRC字节之前，及时关闭CRCEN并读取CRC结果。如果软件不能及时处理，则CRC字节接收将改变数据校验结果。

使用DMA接收数据时，如果使能了CRC，则I2C_SMBUS模块自动完成接收数据的CRC计算，并将结果保存在CRC寄存器中。这种情况下，为了避免CRC字节接收覆盖数据CRC结果，应将DMA接收长度配置为数据字节长度，不含CRC字节。

18.7.3.3 SMBus 从机发送

SMBus从机发送数据流程与I²C从机类似，置位CRCEN后，PEC通过CRC-8单元计算，并由软件完成PEC字节的发送。

DMA发送时，有两种操作方法：

- 关闭AUTOEND功能，在DMA发送完全部数据字节后，再由软件从CRCDR读取PEC字节并发送出去
- 预先对要发送的数据块预计算CRC结果，紧跟着保存在数据块后面；DMA发送时可以使能AUTOEND，发送长度为数据块长度+1字节CRC

18.7.3.4 SMBus 从机接收

SMBus从机接收数据流程与I²C从机类似，PEC计算同样有两种方式，参考SMBus主机接收章节。

18.7.3.5 SMBus 从机唤醒

在低功耗模式下，SMBus可以在接收到START事件时唤醒MCU，也可以只唤醒RCHF时钟用于从机地址接收。如果配置为唤醒MCU，则从机检测到START事件后会将MCU从休眠模式唤醒。如果配置为仅唤醒RCHF时钟，则从机检测到START事件后自动打开RCHF，以便模块能够完成地址字节接收和超时检测。

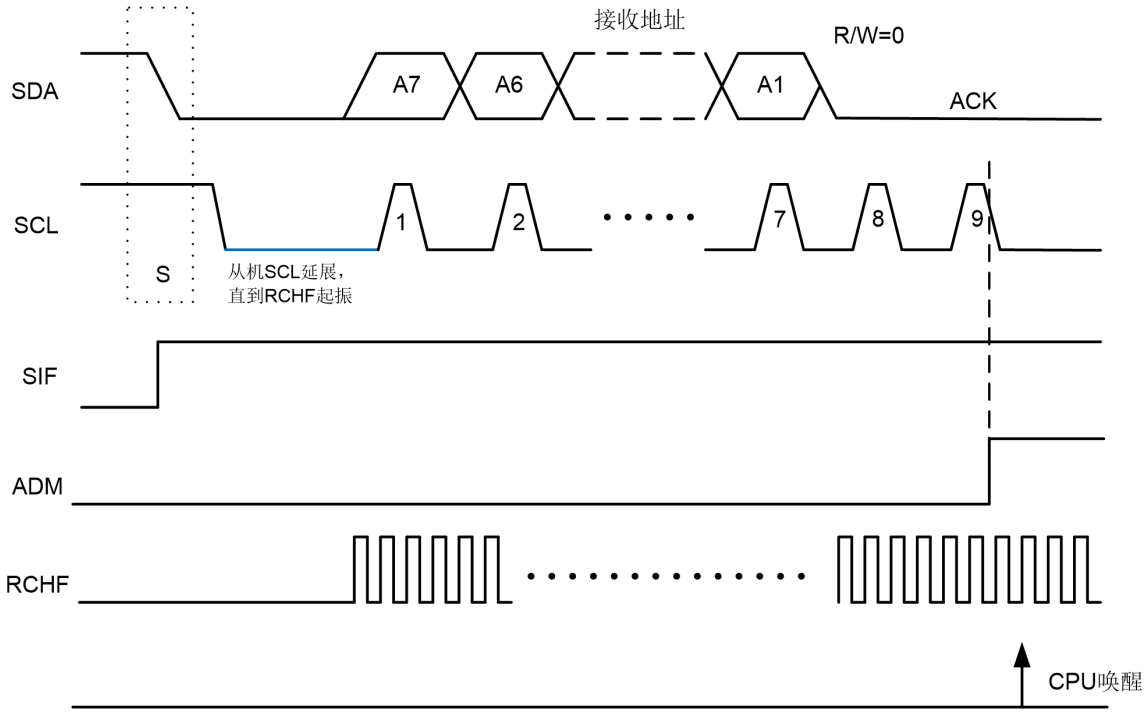


图 18-32 从机地址匹配唤醒

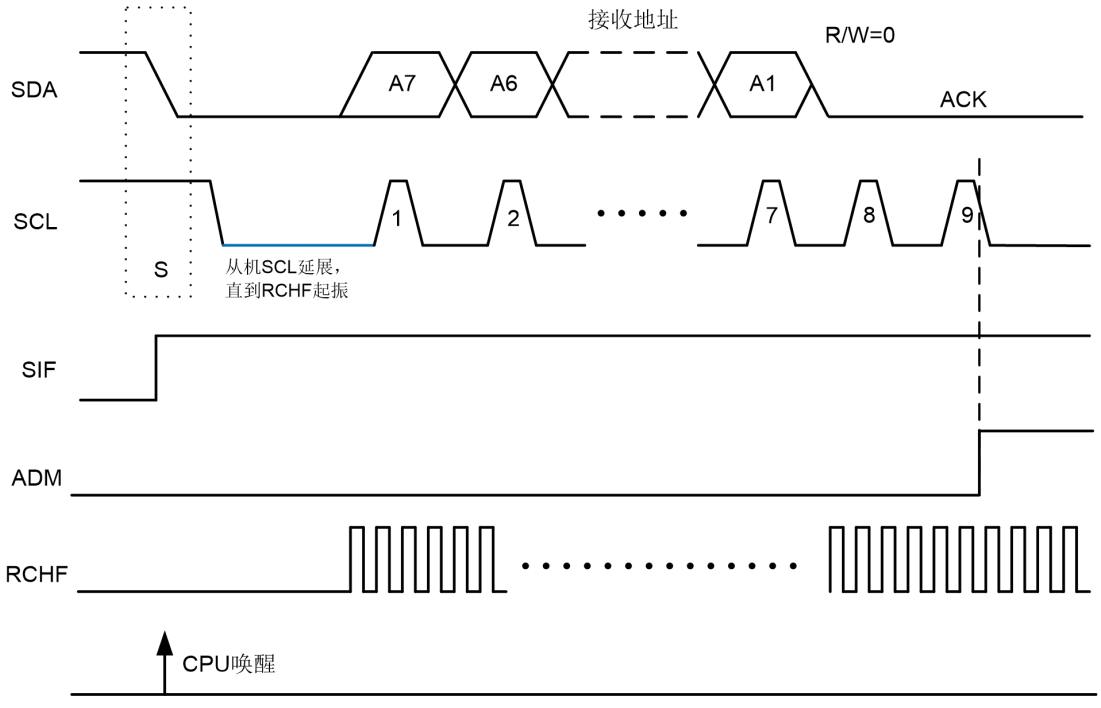


图 18-33 从机 START 唤醒

注意：使用SMBus从机唤醒功能时，必须将I2CCLK配置为RCHF

18.7.4 SMBus 超时功能

SMBus的主机和从机模式都支持$t_{TIMEOUT}$检测，并在检测到超时事件时产生超时中断。

下图显示了总线时钟超时的定义：

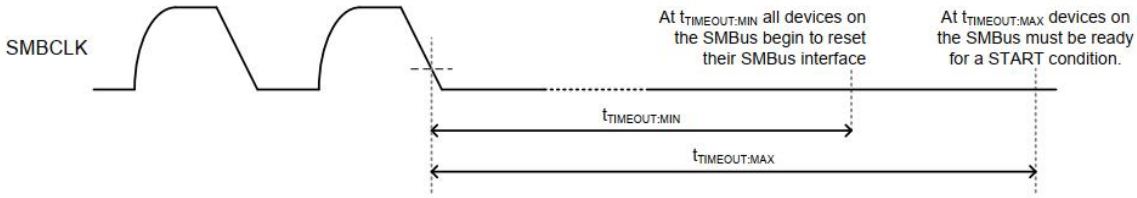


图 18- 34SMBus 总线时钟超时事件

TIMEOUTA寄存器用于定义超时长度，超时时间由下式定义：

$(TIMEOUTA+1)\times2048\times t_{I2CCLK}$

SMBus spec规定的总线超时范围是25~35ms，软件可以根据应用计算合理的TIMEOUTA参数。

当TOEN寄存器置位后，超时计数器在第一次检测到SCL低电平时开始工作，并在每次检测到时钟高电平时自动清零计数器。如果超时计数器溢出，则超时中断寄存器被置位，超时计数器自动关闭，TOEN清零，直到软件下一次启动。

SMBus主机支持$t_{LOW:SEXT}$检测，当主机检测到从机拉低时钟的总时间超过协议规定的上限时，SMBus主机产生超时中断，软件可以选择放弃当前传输，并发送STOP。

SMBus从机支持$t_{LOW:MEXT}$检测，当从机检测到主机拉低时钟的总时间超过协议规定的上限时，SMBus从机产生超时中断，后续行动策略应由软件定义。

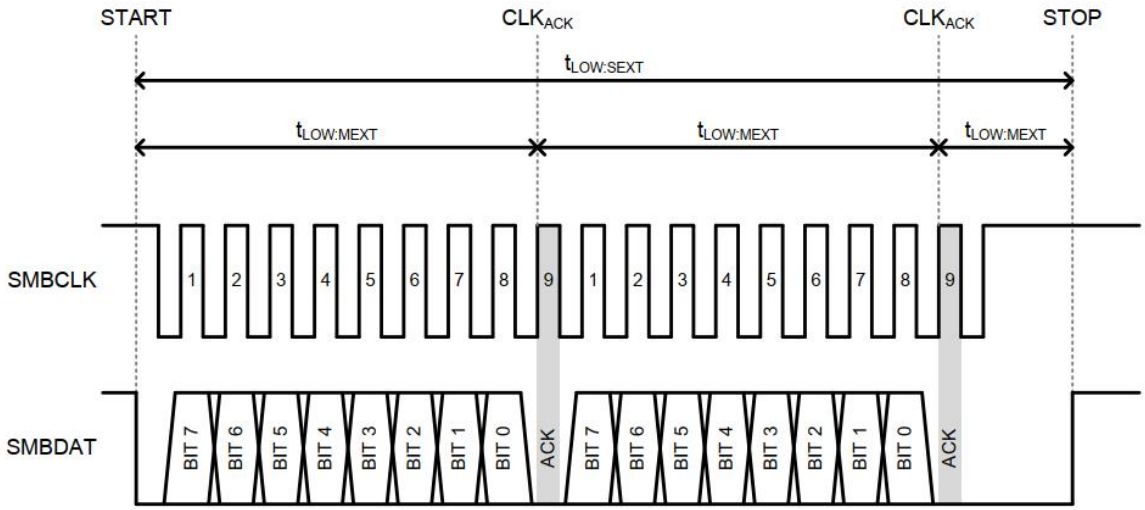


图 18-35 主机和从机超时事件

18.7.5 CRC-8

I2C_SMBUS 模块自带 CRC-8 校验单元，采用串行移位的方式实现，初始值和多项式可配置。以下 Golden 数据供应用中测试及校验使用。

多项式	输入序列	初始值(16 进制)		
		全 0	全 F	6363
		CRC 计算结果（16 进制）		
CRC-8	5A5A	0F	D8	C5
	1223344	F9	28	96

CRC 功能通道置位 CRCEN 寄存器使能，通信结束后再 CRCDR 寄存器中保存计算结果。

18.8 寄存器

I2CSMB0模块基地址：0x4001A000

I2CSMB1模块基地址：0x4001A400

offset 地址	名称	符号
I2C_SMB （模块基地址：）		
0x00000000	I2C_SMB 控制寄存器 1（I2C SMBus Control Register）	I2CSMB_CR1
0x00000004	I2C_SMB 控制寄存器 2（I2C SMBus Control Register）	I2CSMB_CR2
0x00000008	I2C_SMB 中断控制寄存器（ I2C SMBus Intterupt Enable Register）	I2CSMB_IER
0x0000000C	I2C_SMB 状态和标志寄存器（ I2C Master Interrupt Status Register）	I2CSMB_ISR
0x00000010	I2C_SMB 波特率设置寄存器	I2CSMB_BGR
0x00000014	I2C_SMB 主机时序控制寄存器	I2CSMB_TCR
0x00000018	I2C_SMB 超时寄存器	I2CSMB_TOR
0x0000001C	I2C_SMB 接收数据寄存器	I2CSMB_RXBUF
0x00000020	I2C_SMB 发送数据寄存器	I2CSMB_TXBUF
0x00000024	I2C_SMB 从机地址寄存器	I2CSMB_SADR
0x00000028	CRC 数据寄存器	I2CSMB_CRCDR
0x0000002C	CRC LFSR 寄存器	I2CSMB_CRCLFSR
0x00000030	CRC 多项式寄存器	I2CSMB_CRCPOLY

18.8.1 I2C_SMB 控制寄存器 1（I2CSMB_CR1）

名称	I2CSMB_CR1							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	AUTOEND	ALERTEN	SMBDEN	SMBHEN	GCEN	WKUPEN		SCLSEN
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-00		R/W-1
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RXDMAEN	TXDMAEN	-	ANFEN	DNF			
位权限	R/W-0	R/W-0	U-0	R/W-0	R/W-000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				CRCEN	A10EN	-	EN

位权限	U-0	R/W-0	R/W-0	U-0	R/W-0

位号	位名	说明
31:24	--	RFU：未实现，读为 0
23	AUTOEND	主机 DMA 自动终止 (DMA automatic transfer end) 1：DMA 指定长度传输完成后，自动发送 STOP 时序 0：DMA 指定长度传输完成后，等待软件接管
22	ALERTEN	SMBus Alert 引脚功能使能 (SMBus Alert pin enable) 1：使能 Alert 功能 device 模式下可以通过写 ALERT 寄存器拉低 Alert 引脚，并且能够响应 Alert Response Address 0001100x (ACK) host 模式下可以响应 Alert 信号 0：禁止 Alert 功能
21	SMBDEN	SMBus device default address 使能 1：响应 SMBus device default address 1100001x (ACK) 0：不响应 SMBus device default address 1100001x (NACK)
20	SMBHEN	SMBus host address 使能 1：响应 host address 0001 000x (ACK) 0：不响应 host address 0001 000x (NACK)
19	GCEN	General Call 响应使能 1：响应 general call 0000 0000 (ACK) 0：不响应 general call 0000 0000 (NACK)
18:17	WKUPEN	休眠唤醒功能使能(Wakeup Enable) 00：禁止从机休眠唤醒功能 01：START 时序唤醒时钟，不唤醒 MCU 10：START 时序同时唤醒时钟和 MCU 11：START 时序唤醒时钟，地址匹配唤醒 MCU
16	SCLSEN	从机时钟延展功能使能，仅在从机模式下有效(slave SCL stretching enable) 1：使能时钟延展 0：禁止时钟延展
15	RXDMAEN	接收 DMA 使能(Receiver DMA Enable) 1：使能数据接收 DMA 功能 0：禁止数据接收 DMA 功能
14	TXDMAEN	发送 DMA 使能(Transmitter DMA Enable) 1：使能数据发送 DMA 功能 0：禁止数据发送 DMA 功能
13	--	RFU：未实现，读为 0
12	ANFEN	模拟滤波使能 (Analog Noise Filter Enable) 1：使能总线输入模拟滤波 0：禁止模拟滤波
11:8	DNF	数字滤波控制(Digital Noise Filter) 0000：禁止数字滤波 0001：数字滤波长度 1 个 I2CCLK 0010：数字滤波长度 2 个 I2CCLK …… 1111：数字滤波长度 15 个 I2CCLK
7:4	--	RFU：未实现，读为 0
3	CRCEN	CRC 计算使能 1：使能 CRC 计算功能

位号	位名	说明
		0：关闭 CRC 计算
2	A10EN	10bit 地址使能，仅从机模式有效 1：从机使用 10bit 地址 0：从机使用 7bit 地址
1	--	RFU：未实现，读为 0
0	EN	I2C_SMB 模块使能控制位 (module enable) 1：模块使能 0：模块禁止

18.8.2 I2C_SMB 控制寄存器 2（I2CSMB_CR2）

名称	I2CSMB_CR2							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		ALERT	ACKEN	RWN	STOP	RESTART	START
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:6	--	RFU：未实现，读为 0
5	ALERT	SMBus Alert pin 控制寄存器，仅在 ALERTEN=1 时有效 1：SMBus 模块拉低 Alert 引脚 0：SMBus 模块撤销 Alert 引脚
4	ACKEN	主机或从机接收模式下，是否回应 ACK 信号 (ACK output enable) 1：回发 NACK 0：回发 ACK
3	RWN	主机读写方向控制寄存器，此位定义了主机传输方向，应与 Address 字节的 bit0 保持一致，仅主机模式有效。 1：主机读 0：主机写 注意：在主机接收时，每收完 1 个字节数据，硬件会自动清零 RWN，因此连续接收时软件需要在每个字节接收完成后重新置位 RWN
2	STOP	STOP 时序产生使能控制位，软件写 1 发送 STOP 时序，发送完成后硬件自动清零 (Stop Enable)
1	RESTART	Repeated START 时序产生使能控制位，软件写 1 发送 Repeated START 时序，发送完成后硬件自动清零 (Repeated Start Enable)
0	START	START 时序产生使能控制位，软件写 1 发送 START 时序，发送完成后硬件自动清零 (Start Enable)

18.8.3 I2C_SMB 中断控制寄存器（I2CSMB_IER）

名称	I2CSMB_IER							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IDLIE	ERRIE	SIE	PIE	NACKIE	ADIE	TXIE	RXIE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7	IDLIE	总线空闲中断使能寄存器 (Bus Idle interrupt enable) 1：允许总线空闲中断 0：禁止总线空闲中断
6	ERRIE	错误中断使能寄存器 (Error event interrupt enable) 1：允许错误中断 0：禁止错误中断 *当 ERRIE 置位后，以下事件可以触发 CPU 中断* *仲裁失败* *总线错误* *数据溢出* *超时事件* *Alert 检测*
5	SIE	START 中断使能 (START event interrupt enable) 1：允许 START 事件产生中断 0：禁止 START 事件产生中断
4	PIE	STOP 中断使能 (STOP event interrupt enable) 1：允许 STOP 事件产生中断 0：禁止 STOP 事件产生中断
3	NACKIE	主机发送模式下 NACK 中断使能寄存器 (Non-ACK interrupt enable) 1：允许收到 NACK 产生中断 0：禁止产生 NACK 中断
2	ADIE	从机地址匹配中断使能(address match interrupt enable) 1：允许从机地址匹配产生中断 0：禁止从机地址匹配产生中断
1	TXIE	发送完成中断使能(transmit finished interrupt enable) 1：允许发送完成产生中断 0：禁止发送完成产生中断
0	RXIE	接收完成中断使能(receive finished interrupt enable) 1：允许接收完成产生中断 0：禁止接收完成产生中断

18.8.4 I2C_SMB 状态和标志寄存器（I2CSMB_ISR）

名称	I2CSMB_ISR							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							SDIR
位权限	U-0							R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	-	ALIF	TOB	TOA	OVF	BERR	ARLO
位权限	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BUSY	IDLIF	SIF	PIF	NACKIF	ADIF	TXIF	RXIF
位权限	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:17	--	RFU：未实现，读为 0
16	SDIR	从机传输方向，仅从机模式有效，只读；硬件在从机地址匹配后更新 0：主机写操作，从机接收 1：主机读操作，从机发送
15:14	--	RFU：未实现，读为 0
13	ALIF	Alert 标志，SMBus host 模式下有效，检测到 Alert 引脚拉低时硬件置位，软件写 1 清零，写 0 无效
12	TOB	Timeout B 标志，发生 $t_{LOW:MEXT}$ 或 $t_{LOW:SEXT}$ 超时时硬件置位，软件写 1 清零，写 0 无效
11	TOA	Timeout A 标志，发生 SCL 超时硬件置位，软件写 1 清零，写 0 无效
10	OVF	数据溢出标志，在 RXDR 或 TXDR 非空时，有新的数据到来，触发硬件置位，软件写 1 清零，写 0 无效 (Overflow Flag)
9	BERR	总线错误标志，表示传输过程中检测到错误的 START 或 STOP 时序，硬件置位，软件写 1 清零，写 0 无效 (Bus Error)
8	ARLO	仲裁失败标志，硬件置位，软件写 1 清零，写 0 无效 (Arbitration Lost)
7	BUSY	总线忙标志，只读 1：总线正在通信中 0：总线空闲
6	IDLIF	SMBus 总线 IDLE 事件标志，硬件置位，软件写 1 清零，写 0 无效(SMBus IDLE)
5	SIF	START 中断标志，从机模式有效。硬件置位，软件写 1 清零，写 0 无效(START event interrupt flag)
4	PIF	STOP 中断标志，从机模式有效。硬件置位，软件写 1 清零 (STOP event interrupt flag)
3	NACKIF	主机/从机发送模式下 NACK 中断标志寄存器，硬件置位，软件写 1 清零，写 0 无效(Non-ACK interrupt flag)
2	ADIF	从机地址匹配中断标志，硬件置位，软件写 1 清零，写 0 无效(address

位号	位名	说明
		match interrupt flag)
1	TXIF	发送完成中断标志，硬件置位，软件写 1 清零，写 0 无效(transmit finished interrupt flag)
0	RXIF	接收完成中断标志，硬件置位，软件写 1 清零，写 0 无效(receive finished interrupt flag)

18.8.5 I2C_SMB 波特率设置寄存器（I2CSMB_BGR）

名称	I2CSMB_BGR							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							BRGH[8]
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	BRGH[7:0]							
位权限	R/W-00010011							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							BRGL[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BRGL[7:0]							
位权限	R/W-00010011							

位号	位名	说明
31:25	--	RFU：未实现，读为 0
24:16	BRGH	主机发送的 SCL 时钟高电平宽度，以 I2C 工作时钟计数 (Master SCL High level length)
15:9	--	RFU：未实现，读为 0
8:0	BRGL	主机发送的 SCL 时钟低电平宽度，以 I2C 工作时钟计数 (Master SCL Low level length)

18.8.6 I2C_SMB 主机时序控制寄存器（I2CSMB_TCR）

名称	I2CSMB_TCR							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SCLDLY							
位权限	RW-00001000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							SDAHD[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	SDAHD[7:0]
位权限	R/W-00001010

位号	位名	说明
31:24	--	RFU：未实现，读为 0
23:16	SCLDLY	从机模式下，SCL 延展后，软件写入 TXBUF 后到 SCL 释放的时间延迟，以 I2C 工作时钟计数，默认 8 个 clock
15:9	--	RFU：未实现，读为 0
8:0	SDAHD	定义 SDA 相对于 SCL 下降沿的保持时间参数，以 I2C 工作时钟计数 (SDA hold delay) 注意：最小有效值为 1，最大有效值为 BRGL

18.8.7 I2C_SMB 超时寄存器（I2CSMB_TOR）

名称	I2CSMB_TOR							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TEXTEN	-			TIMEOUTB			
位权限	R/W-0	U-0			R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TIMEOUTB							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TOEN	-		TIDLE	TIMEOUTA			
位权限	R/W-0	U-0		R/W-0	R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TIMEOUTA							
位权限	R/W-0000 0000							

位号	位名	说明
31	TEXTEN	累积扩展时钟超时使能 1：主机模式下检测 $t_{LOW:MEXT}$ 超时，从机模式下检测 $t_{LOW:SEXT}$ 超时 0：关闭扩展时钟超时检测
30:28	--	RFU：未实现，读为 0
27:16	TIMEOUTB	用于定义累积扩展时钟超时阈值 主机模式下定义 $t_{LOW:MEXT}$，从机模式下定义 $t_{LOW:SEXT}$ $t_{LOW:EXT}=(TIMEOUTB+1)\times2048\times t_{I2CCLK}$ 注意：禁止在 TEXTEN=1 的情况下改写此寄存器
15	TOEN	时钟超时检测使能 1：SCL 低电平超时检测使能：当 TIDLE=0 时检测 $t_{TIMEOUT}$，TIDLE=1 时检测 t_{IDLE}，阈值由 TIMEOUTA 定义 0：SCL 低电平超时检测禁止
14:13	--	RFU：未实现，读为 0
12	TIDLE	总线 idle 检测使能 1：TIMEOUTA 用于检测总线空闲 0：TIMEOUTA 用于检测 SCL 低电平超时
11:0	TIMEOUTA	TIDLE=0 时用于检测 SCL 超时， $t_{TIMEOUT}=(TIMEOUTA+1)\times2048\times t_{I2CCLK}$ TIDLE=1 时用于检测总线空闲，

		$t_{IDLE}=(TIMEOUTA+1)\times4\times t_{I2CCLK}$ 注意：禁止在 TOEN=1 的情况下改写此寄存器

18.8.8 I2C_SMB 接收数据寄存器（I2CSMB_RXBUF）

名称	I2CSMB_RXBUF							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF							
位权限	R-0000 0000							

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7:0	RXBUF	8bit 接收数据，只读

18.8.9 I2C_SMB 发送数据寄存器（I2CSMB_TXBUF）

名称	I2CSMB_TXBUF							
Offset	0x00000020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF							
位权限	R/W-0000 0000							

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7:0	TXBUF	8bit 发送数据

18.8.10 I2C_SMB 从机地址寄存器（I2CSMB_SADR）

名称	I2CSMB_SADR							
Offset	0x00000024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SAEN	MASK			-		ADDR	
位权限	R/W-0	R/W-000			U-0		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADDR							
位权限	R/W-0000 0000							

位号	位名	说明
31:16	--	RFU：未实现，读为 0
15	SAEN	从机地址使能 (Slave address enable) 1：使能从机地址匹配，ACK 正确地址 0：禁止从机地址匹配，NACK 任何地址
14:12	MASK	从机地址掩码 000：无掩码 001：mask ADDR[0]，即 ADDR[0]不参与地址比较 010：mask ADDR[1:0]，即 ADDR[1:0]不参与地址比较 011：mask ADDR[2:0]，即 ADDR[2:0]不参与地址比较 100：mask ADDR[3:0]，即 ADDR[3:0]不参与地址比较 101：mask ADDR[4:0]，即 ADDR[4:0]不参与地址比较 110：mask ADDR[5:0]，即 ADDR[5:0]不参与地址比较 111：mask ADDR[6:0]，即 ADDR[6:0]不参与地址比较
11:10	--	RFU：未实现，读为 0
9:0	ADDR	10bit 从机地址，当 A10EN=1 时，10bit 地址全部有效，当 A10EN=0 时，只有低 7bit 地址有效

18.8.11 CRC 数据寄存器（I2CSMB_CRCDR）

名称	I2CSMB_CRCDR							
Offset	0x00000028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DR[7:0]							
位权限	R -1111 1111							

位号	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	**DR**	CRCDR 用于在运算结束后保存 CRC 计算结果。(CRC Data Register)

18.8.12 CRC LFSR 寄存器（I2CSMB_CRC_LFSR）

名称	I2CSMB_CRCLFSR							
Offset	0x0000002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LFSR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	**LFSR**	CRC 线性反馈移位寄存器 (Linear Feedback Shift Register) 运算开始前可以由软件写入 CRC 初始值

18.8.13 CRC 多项式寄存器（I2CSMB_CRC_POLY）

名称	I2CSMB_CRCPOLY							
Offset	0x00000030							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	POLY[7:0]							
位权限	R/W-0000 0111							

位号	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	**POLY**	CRC 运算多项式系数 (CRC Polynominals)

19 双线串行总线主机（I²C master）

19.1 概述

I²C 主机模块实现 MCU 与外部 I2C 接口器件之间的同步通信，硬件实现串并转换。支持 I²C 的主机模式，不支持多主机模式。

特点：

- 4 路独立 I²C 主机接口
- 支持主机模式，不支持多主机模式，不支持从机模式
- 支持 7 位或 10 位从机地址
- 传输速度支持 standard mode(100Kbps), fast mode(400Kbps)和 Fm+(1Mbps)

19.2 结构框图

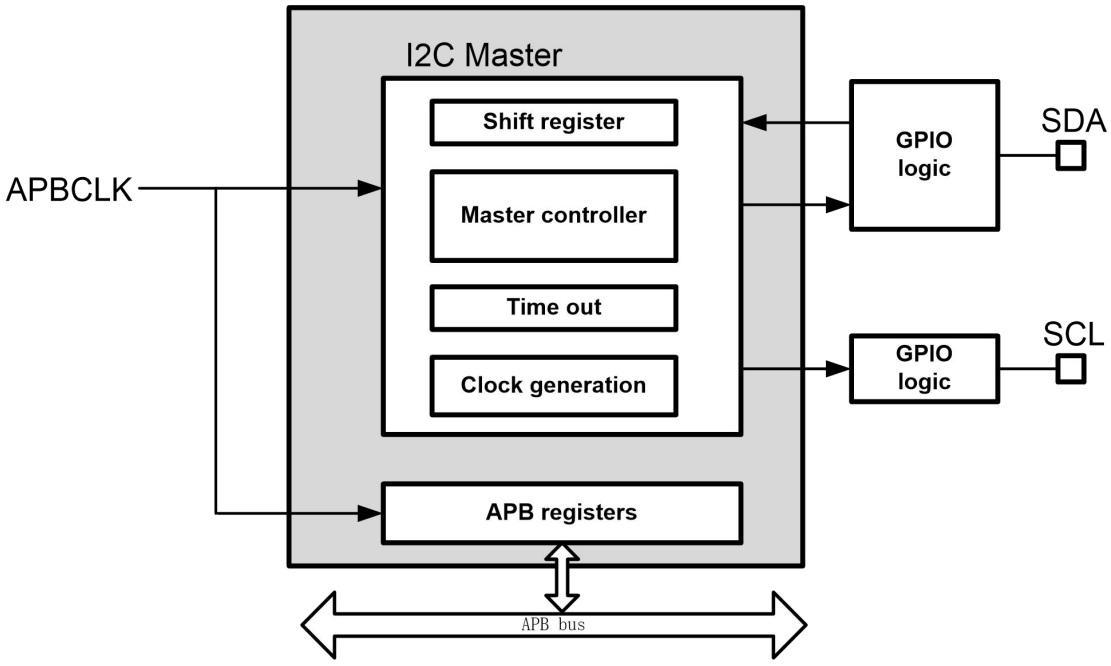


图 19-1 I²C 主机模块框图

19.3 时钟

I²C 主机使用 APBCLK 同步工作。时钟使能控制在 CMU 模块内完成，进行 I²C 通信前必须正确配置相应的 CMU 控制寄存器。

19.4 接口时序

19.4.1 接口时序图

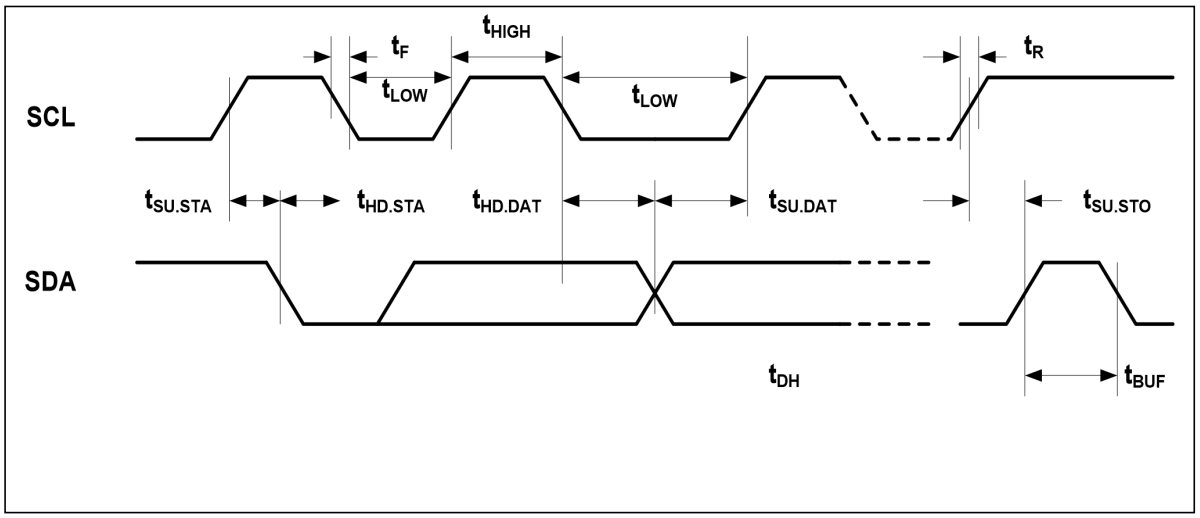


图 19-2 I²C 总线时序

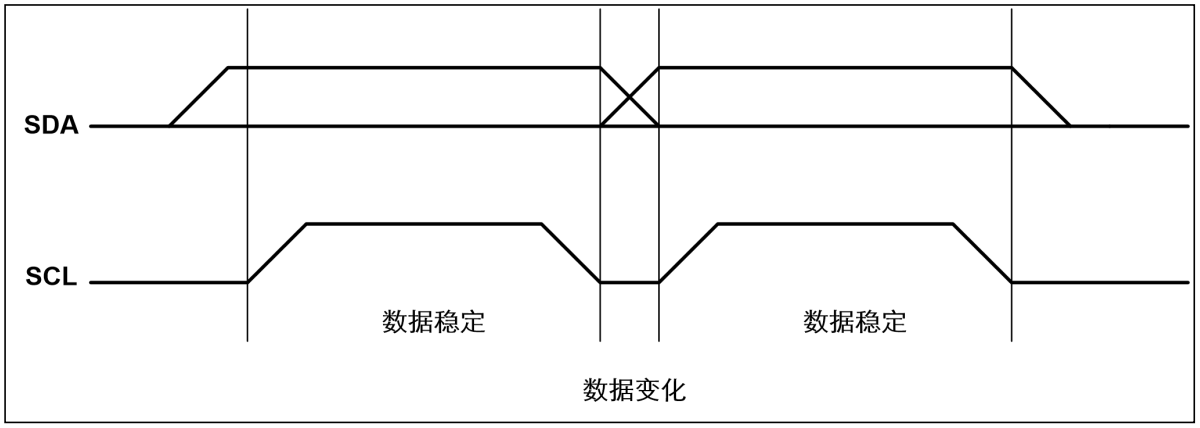


图 19-3 数据有效时序

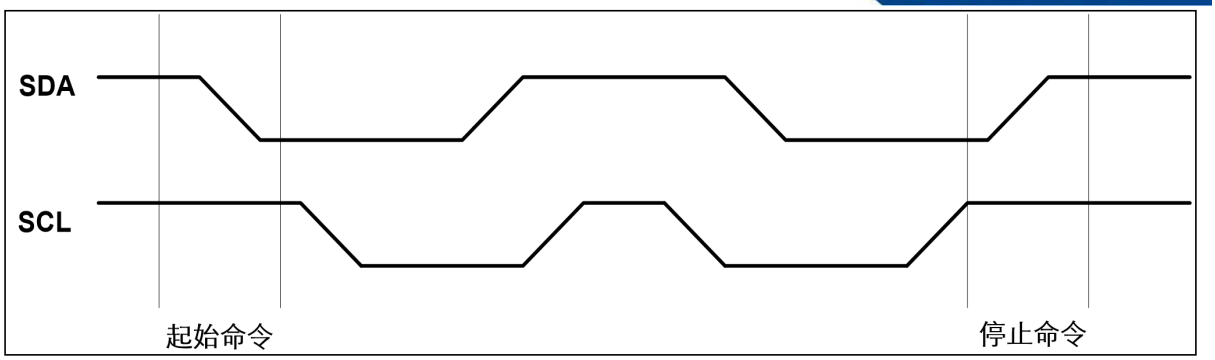


图 19-4 起始（Start）与停止(Stop)命令定义

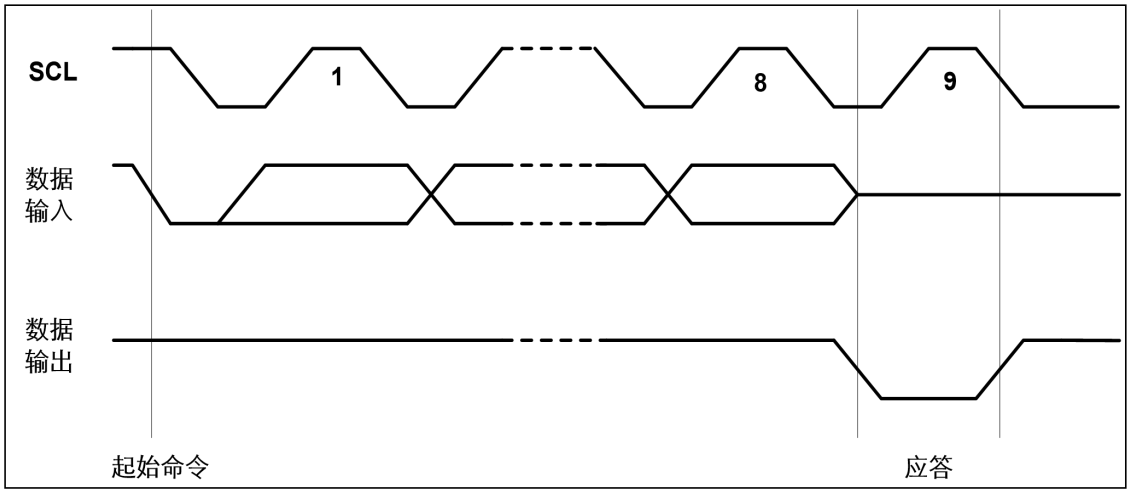


图 19-5 输出应答(ACK)

19.4.2 接口时序描述

时钟有效时序：SDA 引脚通常被外围器件拉高。SDA 引脚的数据应在 SCL 为低时变化；当数据在 SCL 为高时变化，将视为下文所述的一个起始或停止命令。

起始命令：当 SCL 为高，SDA 由高到低的变化被视为起始命令，必须以起始命令作为任何一次读/写操作命令的开始。

停止命令：当 SCL 为高，SDA 由低到高的变化被视为停止命令，在一个读操作后，停止命令会使 EEPROM 进入等待态低功耗模式。

输出应答：SDA 上的数据都是以 8 位为一组串行输入和输出的，MSB 先发，接收方在收完每个字节后应当在第 9 个周期回发一个回应 acknowledge 位（以下简称 ack），ack 的时钟由主机提供。发送方在 ack 期间悬空 SDA，接收方须将 SDA 拉低，确保 ack 时钟高电平期间 SDA 为低，形成有效的 ack 信号。

参数	符号	标准模式（100K）		快速模式(400K)		单位
		最小值	最大值	最小值	最大值	

参数	符号	标准模式（100K）		快速模式(400K)		单位
		最小值	最大值	最小值	最大值	
SCL 时钟频率	F_{SCL}	0	100	0	400	kHz
启动条件建立时间	$T_{SU:STA}$	4.7	—	0.6	—	us
启动条件保持时间	$T_{HD:STA}$	4.0	—	0.6	—	us
时钟低电平时间	T_{LOW}	4.7	—	1.3	—	us
时钟高电平时间	T_{HIGH}	4.0	—	0.6	—	us
数据输入建立时间	$T_{SU:DAT}$	250	—	100(4)	—	ns
数据输入保持时间	$T_{HD:DAT}$	5.0 0(2)	— 3.45(3)	— 0(2)	— 0.9(3)	us us
SDA 和 SCL 上升时间	T_R	—	1000	20+0.1Cb(5)	300	ns
SDA 和 SCL 下降时间	T_F	—	300	20+0.1Cb(5)	300	ns
停止条件建立时间	$T_{SU:STO}$	4.0	—	0.6	—	us
总线空闲时间	T_{BUF}	4.7	—	1.3	—	us
总线的容性负载	Cb	—	400	—	400	Pf
噪声容限低值	V_{nL}	$0.1V_{DD}$	—	$0.1V_{DD}$	—	V
噪声容限高值	V_{nH}	$0.2V_{DD}$	—	$0.2V_{DD}$	—	V

表 19-1 I^2C 接口时序要求

19.5 I²C 工作模式

I²C模块支持以下工作模式：

- 主机接收
- 主机发送

芯片上电后I²C模块默认关闭，主机和从机都不工作。软件需要根据应用选择模块工作模式，通过设置MSPEN来使能主机通信。

19.6 I²C 地址格式

I²C总线协议定义了以下保留地址，对其中多数保留地址，I²C硬件不做合法性判断，由软件保证地址的合法性。

从机地址	R/W_ bit	描述
0000 000	0	General Call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Reserved for different bus format
0000 011	X	Reserved for future purpose
0000 1XX	X	HS-mode master code
1111 1XX	X	Reserved for future purpose
1111 0XX	X	10bit slave addressing

表 19-2 I²C 保留地址定义

19.7 I²C 初始化

进行I²C通信前必须正确的初始化I²C模块，建议软件按照以下步骤进行初始化操作：

- 清零RCC模块的I2CRST寄存器，确保I²C模块不处于复位状态
- 置位RCC模块的I2C_APBEN寄存器，使能I²C模块寄存器总线接口时钟

19.7.1 主机波特率配置

I²C 主机需要在使能前配置通信波特率，而从机不需要配置。

MSPBRG[8:0]波特率配置寄存器用于产生通信波特率。MSPBRG 是 9 bit 波特率分频系数，波特率计算公式如下：

$T_{SCL}=2T_{BRG}$

T_{BRG}＝2×T_{I2CCLK}×(MSPBRG[8:0]+1)；T_{I2CCLK}为 I²C 工作时钟周期，即：

MSPBRG = F_{I2CCLK}/(4*F_{SCL}) – 1

例如对于 100k 波特率，若 I²C 工作时钟为 8M，则 MSPBRG=19。

19.8 I²C 主机功能

FM33FG0A的I²C主机模式不支持多主机总线，因此挂在总线上的其他设备都是从机。总线上总是由主机提供同步时钟SCL，SDA数据流方向可以是主机发送从机接收，或者从机发送主机接收。

I²C总线通信总是由主机发起，主机模式支持7bit或10bit寻址。

19.8.1 7bit 寻址

在7bit寻址时，主机发送的第一个字节包含从机地址和传输方向位（$R/\overline{W}$），根据$R/\overline{W}$决定后续传输是主机向从机写入数据（$R/\overline{W}$=0）或主机从从机读取数据（$R/\overline{W}$=1）。

名字	Slave Address Byte							
位	7	6	5	4	3	2	1	0
位名	address							R/W

位描述：

位	位名	功能
7-1	address	Slave device address
0	R/W	0: Write 表示发送数据（master 发送） 1: Read 表示请求数据（slave 回发）

主机向从机写入数据

典型的7bit寻址，主机向从机写入数据的帧结构如下图所示。

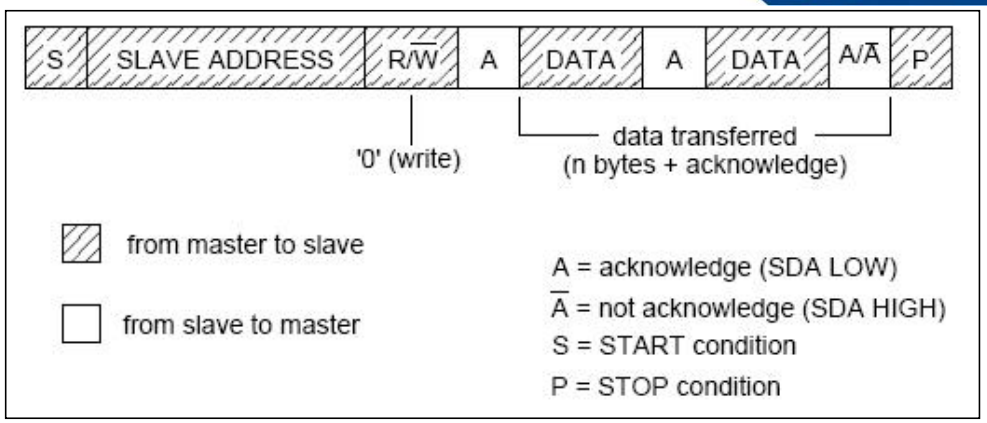


图 19-6 主机向 7 位地址从机写入数据时的帧格式

1、主机发起 START 时序

2、主机发送从机地址，从机地址包含 7 位从机地址和 1 位 R/W 标志位，发送数据时 R/W 位为 0

4、主机发送第一帧 8 位数据

5、主机在每次发送完 8 位数据后，会在第 9 个 SCL 判断是否检测到有效的 ACK，如果主机检测到 ACK 成功后，会继续输出下一字节数据

6、若从机无法响应 ACK，主机检测到 NACK 后应发送 STOP 时序终止发送

7、主机完成所有数据发送后，发送 STOP 时序

软件启动 I2C 主机发送的操作流程如下图：

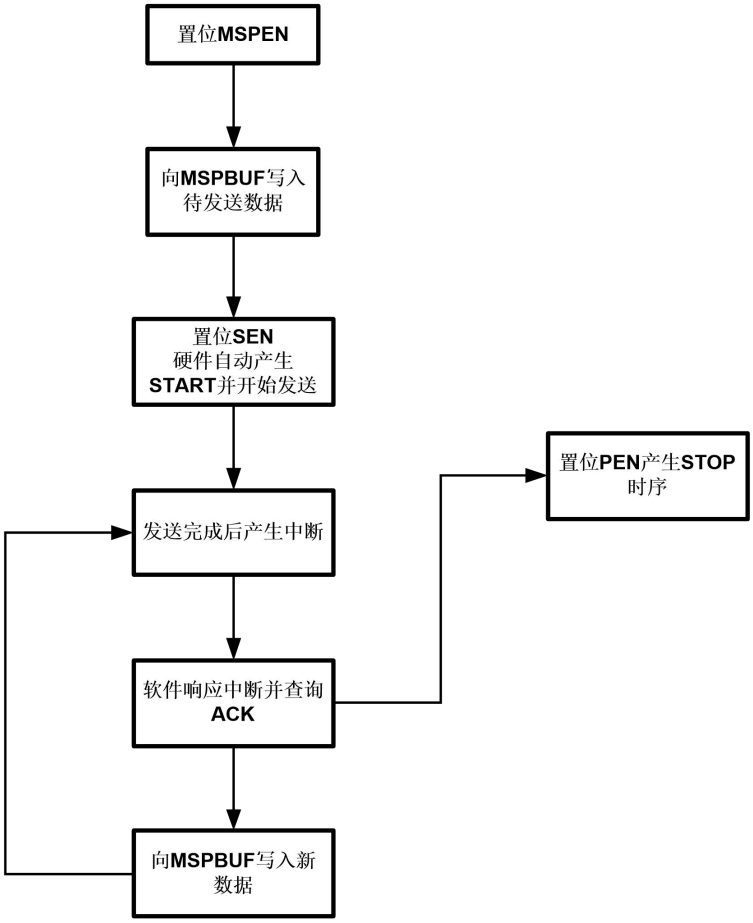


图 19-7I²C 软件发送数据流图

I2C主机对7位地址从机写入数据的波形示意图如下：

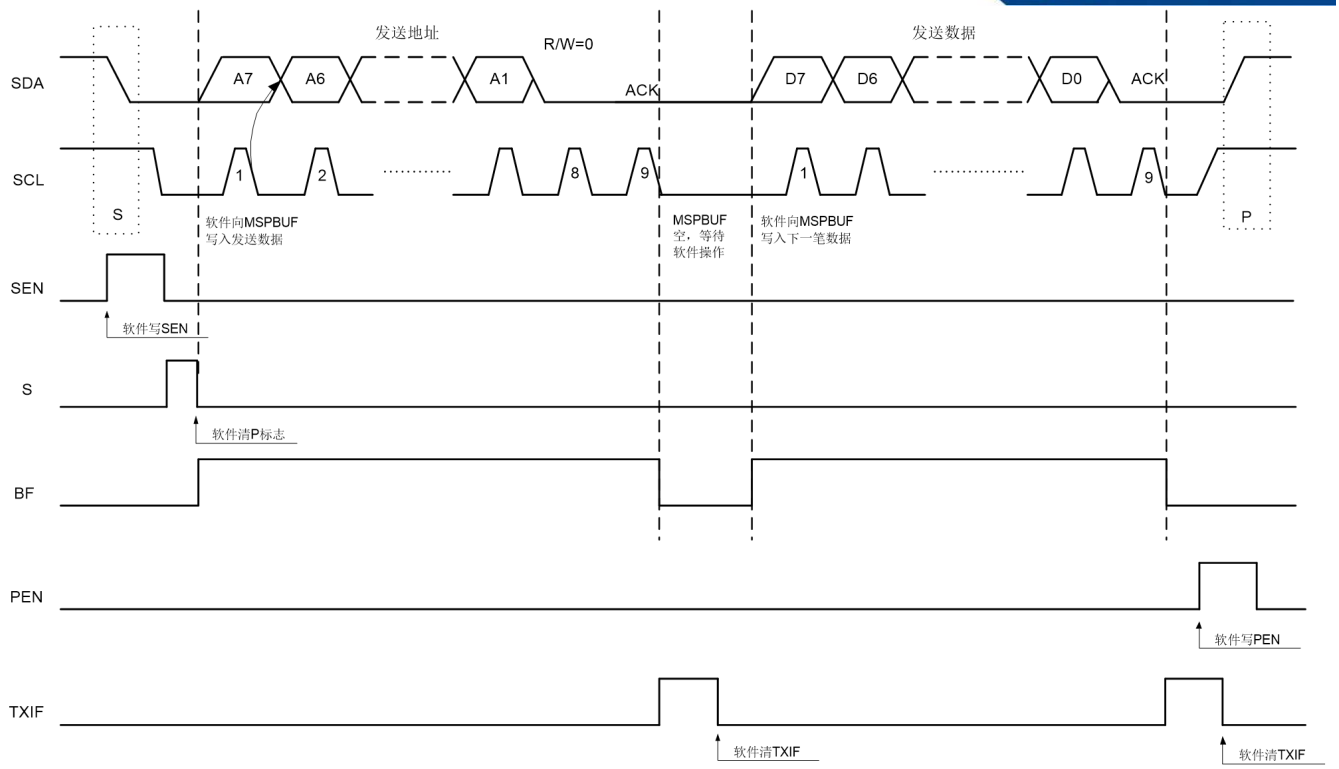


图 19-8 I²C 主机对 7 位地址从机发送数据流图

主机从从机读取数据

典型的7bit寻址，主机从从机读取数据的帧格式如下图所示。

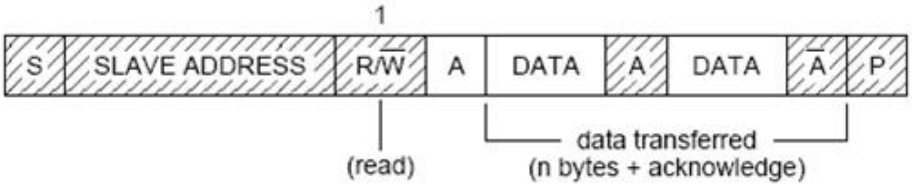


图 19-9 主机从 7 位地址从机读取数据时的帧格式

1、 主机发起 START 时序

2、 主机发送从机地址，从机地址包含 7 位从机地址和 1 位 R/W 标志位，数据读取时 R/W 位为 1

3、 此时设置 MSPCON.RCEN 为 1，主机自动转为接收状态

4、 主机开始接收第一字节 8 位数据，并在第 9 个 SCL 向从机发送有效 ACK,从而继续读取下一字节 8 位数据

5、 主机读取最后一个字节后，在第 9 个 SCL 向从机发送 NACK

6、 主机发送 STOP 时序终止读取

软件启动 I²C 接收的操作流程如下图：

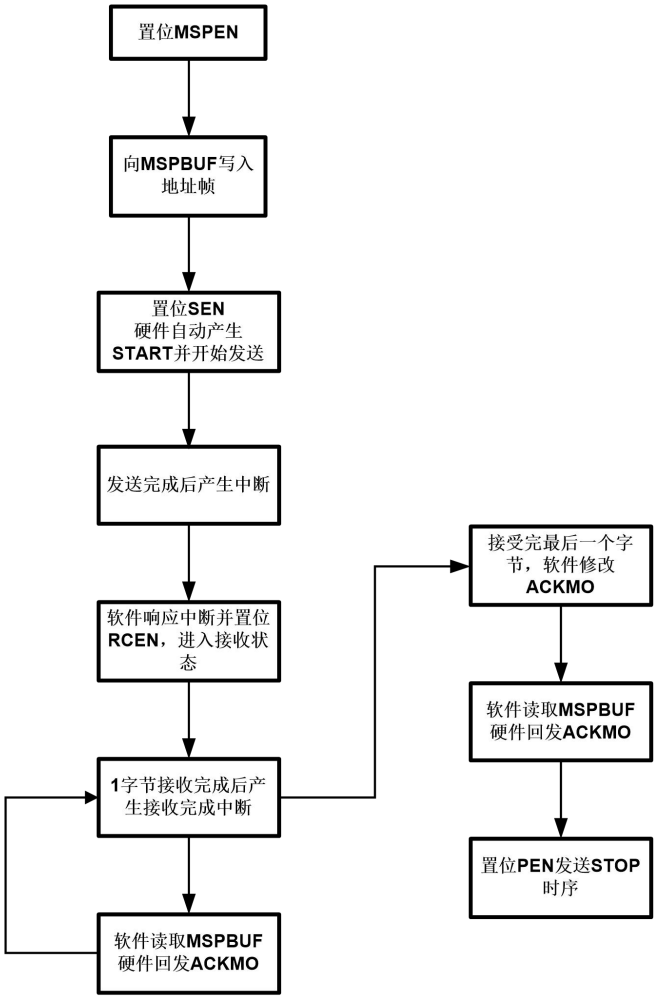


图 19-10 I²C 软件发送数据流图

主机每次接收完从机发送的数据后，根据ACKMO寄存器回发响应。ACKMO复位值为0，即默认状态下主机回发ACK。如果软件希望主机在接收完成后回发NACK，则需要在前一个字节接收完成中断中将ACKMO寄存器改写为1。ACKMO为1的情况下，主机在发送完响应后会自动清零ACKMO。

I²C主机从7位地址从机读出数据的波形示意图如下：

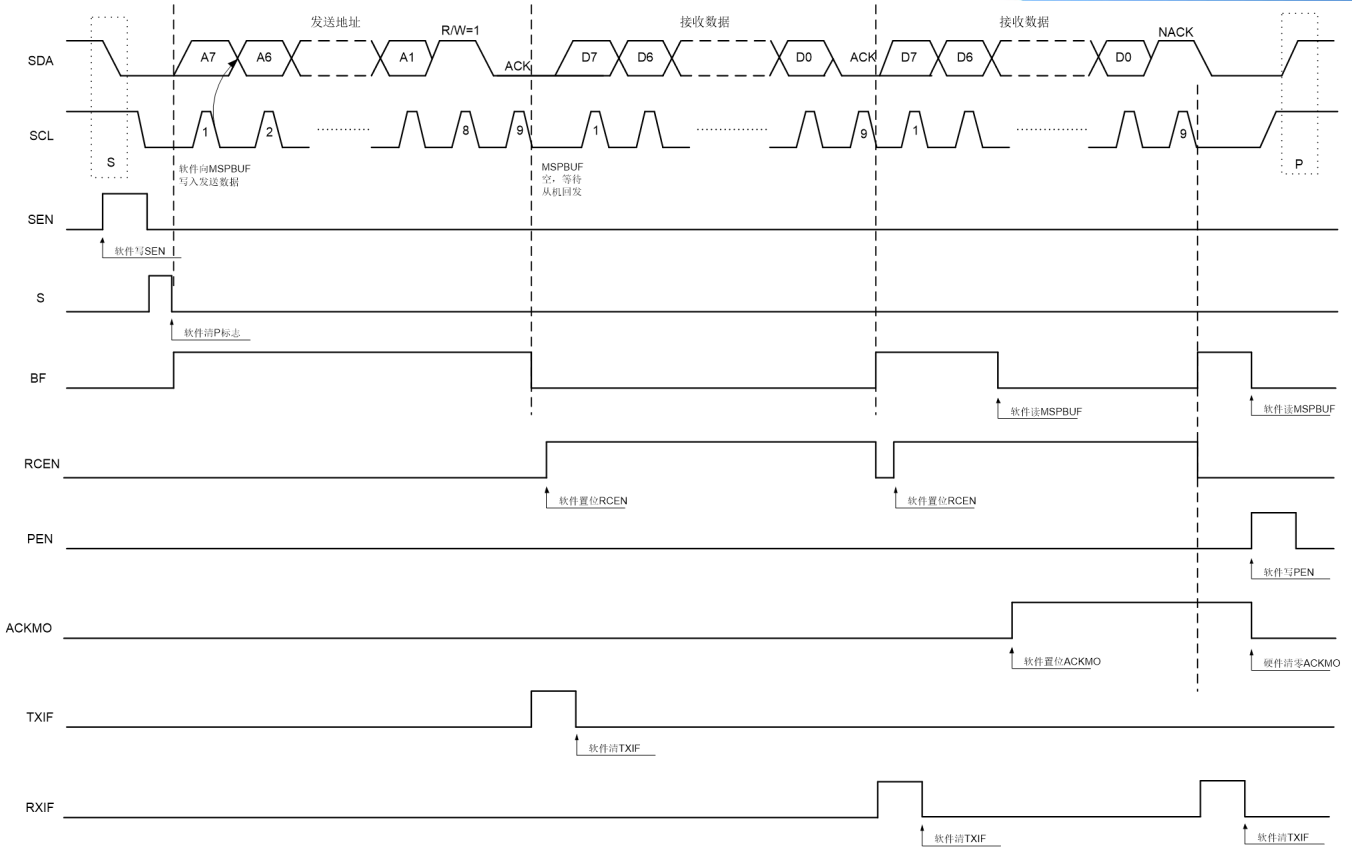


图 19-11 I²C 从 7 位地址从机读取数据流图

双向数据传输（组合模式）

典型的双向数据读写流图如下图所示。在主机发送或读取数据过程中，主机可以通过发送 Repeated Start 时序来重新启动一次新的发送或读取通信，所以主机在一次通信中，即可以有数据发送也可以有数据读取。

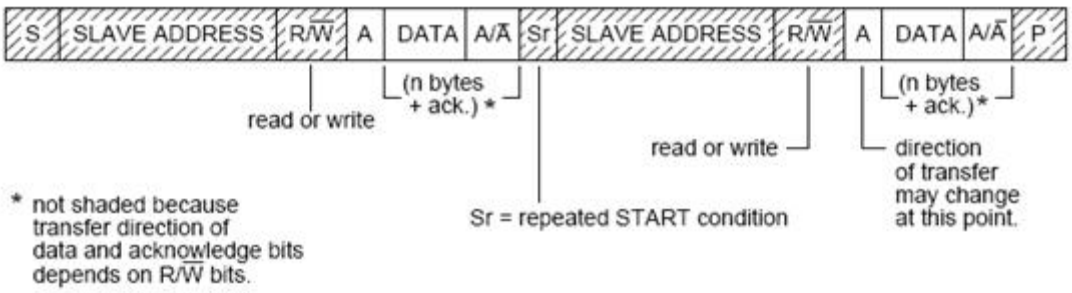


图 19-12 双向数据通信帧格式

组合传输的软件操作流程与单向传输类似，只是在某个字节收发完成后，通过发送ReSTART时序和从机地址字节来修改传输方向。

19.8.2 10bit 寻址

在10bit寻址时，主机发送的第一个字节包含部分从机地址（11110_A9_A8）和传输方向位（$R/\overline{W}$），第二个字节包含剩余从机地址（A7~A0）。两个字节地址发送完成后，再进行数据传输。

主机向从机写入数据

典型的10bit寻址，主机向从机写入数据的数据流图如下图所示。

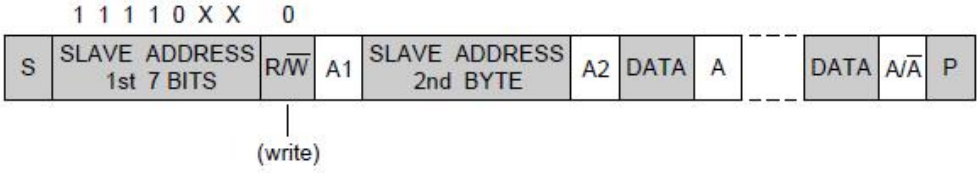


图 19-13 10bit 寻址，主机向从机写入数据

1、主机发起 START 时序

2、主机发送第一个从机地址字节，以 11110 开头，跟随 2bit 从机地址最高位，以及 R/W 标志位，发送数据时 R/W 位为 0

3、主机检查从机回发的 ACK

4、主机发送第二个从机地址字节，包含从机地址的低 8 位

5、主机检查从机回发的 ACK

6、主机继续向从机写入数据

7、主机完成所有数据发送后，发送 STOP 时序

软件启动 I²C 主机发送的操作流程如下图：

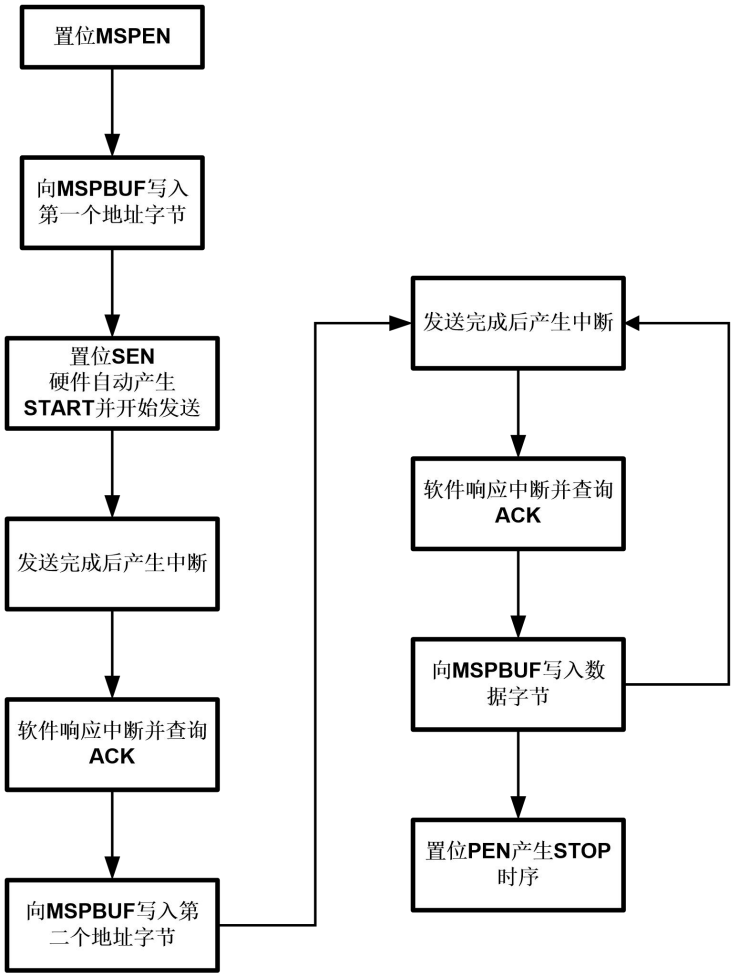


图 19-14I²C 软件发送数据流图

主机从从机读取数据

典型的10bit寻址，主机从从机读取数据的数据流图如下图所示。

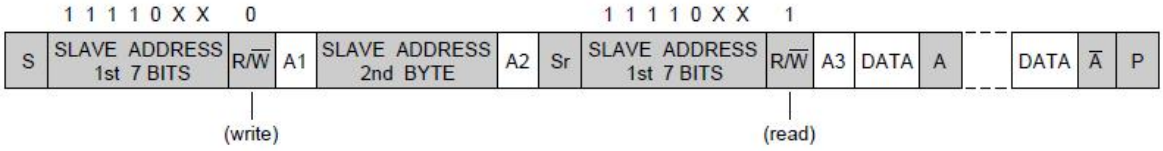


图 19-15 10bit 寻址，主机从从机读取数据

1、 主机发起 START 时序

2、 主机发送第一字节从机地址，包含 5 位前导码 11110、2 位从机地址最高位和 1 位 R/W 标志位，数据读取时 R/W 位为 1

3、 主机发送第二字节从机地址，包含低 8 位地址

4、 主机发送 ReSTART 时序

5、 主机再次发送第一字节从机地址，将 R/W 为改为 0

6、 此时设置 MSPCON.RCEN 为 1，主机转为接收状态

7、 主机开始接收第一字节 8 位数据，并在第 9 个 SCL 向从机发送有效 ACK,从而继续读取下一字节 8 位数据

8、 主机读取最后一个字节后，在第 9 个 SCL 向从机发送 NACK

9、 主机发送 STOP 时序终止读取

软件启动 I²C 接收的操作流程如下图：

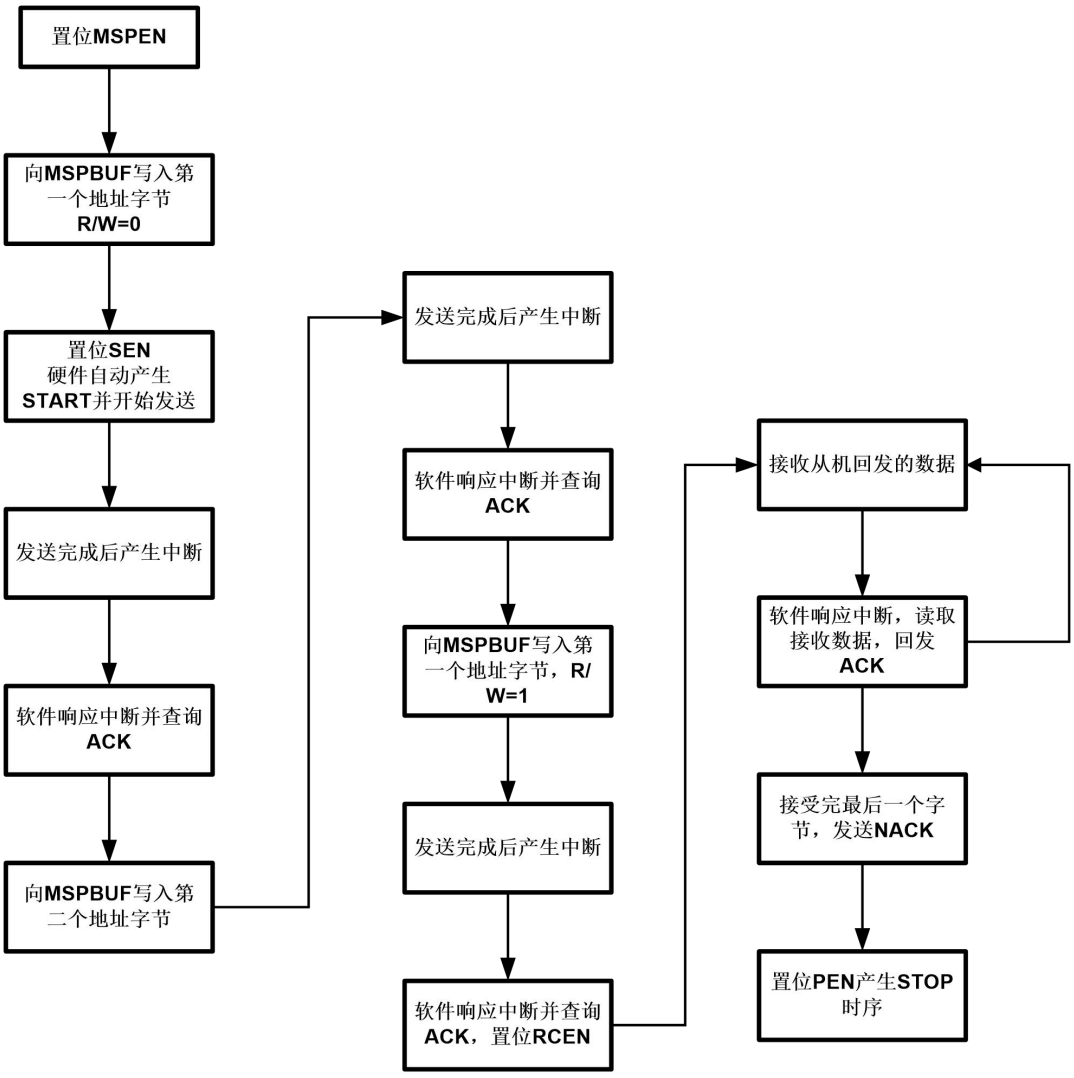


图 19-16 I²C 软件发送数据流图

双向数据传输（组合模式）

典型的双向数据读写流图如下图所示。在主机发送或读取数据过程中，主机可以通过发送 Repeated Start 时序来重新启动一次新的发送或读取通信，所以主机在一次通信中，即可以有数据发送也可以有数据读取。

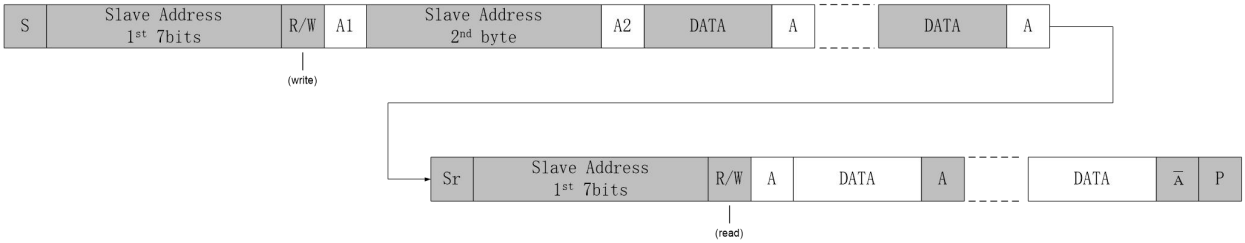


图 19-17 I²C 软件发送数据流图

组合传输的软件操作流程与单向传输类似，只是在某个字节收发完成后，通过发送ReSTART时序和1st从机地址字节来修改传输方向。

19.8.3 SCL 延展（Slave Clock Stretching）

I²C 总线运行低速从机通过拉低 SCL 的方式通知主机暂停数据通信。I²C 主机必须支持这一特性，因此在每个字节收发起始位置处，主机在尝试发送 SCL 高电平后，需要自动检查总线上 SCL 的实际电平，如果不是高电平，意味着从机正在进行 SCL 延展，主机会持续监控 SCL 电平，直到 SCL 为高，才开始后续操作。

注意：主机只在每字节收发的第一个 SCL 上升沿处进行 SCL 延展检查。

19.8.4 超时机制

I²C主机还实现了超时机制，即发现从机长时间拉低SCL导致总线无法通信的情况下，产生超时报警中断并返回IDLE状态。

当主机检测到SCL延展，其内部定时器开始计时，主机设定的SCL延展超时的时长最长是4096个SCL周期，假设波特率为100K，则超时周期大约是40ms，如果波特率是400K，则超时周期大约是10ms。通过12bit的TIMEOUT寄存器，软件可以设置超时周期。软件必须在MSPEN为0的情况下设置TIMEOUT寄存器，此寄存复位值为0xFFF，即表示最长4096*T_{SCL}的超时周期，当检测到SCL延展后，TIMEOUT寄存器开始向下递减，当计数到0时，计数停止，TIMEOUT寄存器被复位到0xFFF，同时触发超时中断。因此通过修改TIMEOUT初始值，可以设定超时周期。

$T_{SCL_STRETCHING_TIMEOUT}$=TIMEOUT[11:0] * T_{SCL}

当发生TIMEOUT中断时，建议软件复位I²C模块。

此功能可以被关闭，如果关闭硬件超时，软件也可以通过定时器结合SCL引脚状态判断来自行实现任意长度的超时判决。

19.8.5 可编程时序

I²C 模块的主机模式提供了灵活的时序编程特性，允许用户定义 SCL 时钟的低电平宽度、高电平宽度，SDA 数据的建立和保持时间。

通过 MSPBRG 寄存器可以设置 SCL 的低电平和高电平宽度，通过 SDAHD 寄存器可以配置 SDA 数据相对 SCL 时钟脉冲的保持和建立保持时间长度。此功能对于发送数据和发送 ACK 都有效。

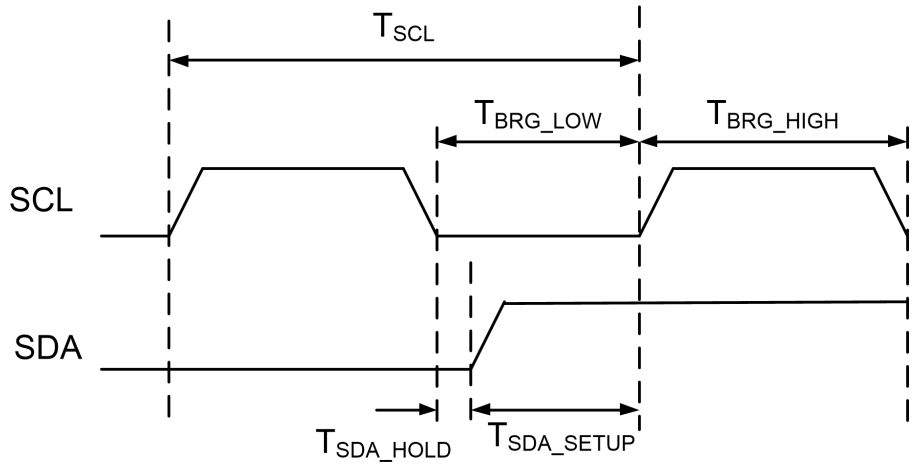


图 19-18 主机时序控制

上图中，T_{SCL}为通信波特率，各个参数可以由以下公式表达：

$T_{SCL} = T_{BRG_LOW} + T_{BRG_HIGH}$

$T_{SDA_SETUP} = T_{BRG_LOW} - T_{SDA_HOLD}$

注意，应用中对MSPBGRH、MSPBRGL和SDAHD寄存器的配置必须满足以下要求，如果违反这些要求将导致异常的总线时序：

MSPBRGH >= 2

MSPBRGL >= 2

MSPBRGL-1 >= SDAHD >= 1

TIMEOUT >= 1

19.9 寄存器

I2C0模块基地址：0x40012400

I2C1模块基地址：0x40019C00

offset 地址	名称	符号
0x00	I2C 主机配置寄存器（I2C Master Config Register）	I2C_MSPCFGR
0x04	I2C 主机控制寄存器（I2C Master Control Register）	I2C_MSPCR
0x08	I2C 主机中断使能寄存器（ I2C Master Intterupt Enable Register）	I2C_MSPIER
0x0C	I2C 主机中断标志寄存器（ I2C Master Interrupt Status Register）	I2C_MSPISR
0x10	I2C 主机状态寄存器（I2C Master Status Register）	I2C_MSPSR
0x14	I2C 主机波特率寄存器（I2C Master Baud rate Generator Register）	I2C_MSPBGR
0x18	I2C 主机收发缓存寄存器（I2C Master transfer Buffer）	I2C_MSPBUF
0x1C	I2C 主机时序控制寄存器（ I2C Master Timing Control Register）	I2C_MSPTCR
0x20	I2C 主机超时寄存器（I2C Master Time-Out Register）	I2C_MSPTOR

19.9.1 I2C 主机配置寄存器（I2C_MSPCFGR）

名称	I2C_MSPCFGR							
offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						AUTOEND	MSP_DMAEN
位权限	U-0						R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						TOEN	MSPEN
位权限	U-0						R/W-0	R/W-0

位号	位名	说明

位号	位名	说明
31:18	--	RFU：未实现，读为 0
17	AUTOEND	主机 DMA 自动终止 (DMA automatic transfer end) 1：DMA 指定长度传输完成后，自动发送 STOP 时序 0：DMA 指定长度传输完成后，等待软件接管
16	MSP_DMAEN	主机 DMA 使能 (Master DMA enable) 0：关闭 DMA 功能 1：使能 DMA 功能
15:2	-	RFU：未实现，读为 0
1	TOEN	SCL 拉低超时使能 (Time Out enable) 1：使能超时功能，超时周期由 MSPTO 寄存器定义 0：关闭超时功能
0	MSPEN	I2C 主机模块使能控制位 (Master enable) 1：I2C 主机使能 0：I2C 主机禁止

19.9.2 I2C 主机控制寄存器（I2C_MSPCR）

名称	I2C_MSPCR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				RCEN	PEN	RSEN	SEN
位权限	U-0				R/W-0	R/W/Dy-0	R/W/Dy-0	R/W/Dy-0

位号	位名	说明
31:4	--	RFU：未实现，读为 0
3	RCEN	主机接收模式下，接收使能位 (Receive enable) 1：主机接收使能 0：接收禁止 主机通信中，软件在发送完地址字节后，通过置位 RCEN 将传输方向切换为主机接收，然后可以接收来自于从机的数据。RCNE 在接收过程中保持为 1，直到软件置位 PEN 发送 STOP 时序。
2	PEN	STOP 时序产生使能控制位，软件写 1 发送 STOP 时序，发送完成后硬件自动清零 (Stop Enable)
1	RSEN	Repeated START 时序产生使能控制位，软件写 1 发送 Repeated START 时序，发送完成后硬件自动清零 (Repeated Start Enable)
0	SEN	START 时序产生使能控制位，软件写 1 发送 START 时序，发送完成后硬件自动清零 (Start Enable)

19.9.3 I2C 主机中断使能寄存器（I2C_MSPIER）

名称	I2C_MSPIER							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	WCOLE	OVTE	SE	PE	NACKE	TXIE	RXIE
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:7	--	RFU：未实现，读为 0
6	WCOLE	WCOL 中断使能寄存器 (Write collision interrupt enable) 1：允许写冲突中断 0：禁止写冲突中断
5	OVTE	SCL 超时中断使能寄存器 (SCL overtime enable) 1：允许超时中断 0：禁止超时中断
4	SE	START 时序中断使能寄存器 (Start interrupt enable) 1：允许 START 时序中断 0：禁止 START 时序中断
3	PE	STOP 时序中断使能寄存器 (Stop interrupt enable) 1：允许 STOP 时序中断 0：禁止 STOP 时序中断
2	NACKE	主机发送模式下 NACK 中断使能寄存器 (Non-ACK interrupt enable) 1：允许收到 NACK 产生中断 0：禁止产生 NACK 中断
1	TXIE	I2C 主机发送完成中断使能(Trasnmit done interrupt enable) 1：允许发送完成中断 0：禁止发送完成中断
0	RXIE	I2C 主机接收完成中断使能(Receive done interrupt enable) 1：允许接收完成中断 0：禁止接收完成中断

19.9.4 I2C 主机中断标志寄存器（I2C_MSPISR）

名称	I2C_MSPISR							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	WCOL	OVT	S	P	ACKSTA	TXIF	RXIF
位权限	U-0	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:7	--	RFU：未实现，读为 0
6	WCOL	写冲突检测位，MCU 只能在完成 START 时序或发送完成一帧读写之后才能写 MSPBUF，否则发生写冲突；硬件置位，软件写 1 清零(Write Collision Interrupt Flag,write 1 to clear) 1：发送写冲突 0：未发生冲突
5	OVT	SCL 超时中断标志，仅在 TOEN 为 1 时工作(SCL OverTime Interrupt Flag) 1：发生 SCL 超时 0：没有发生 SCL 超时
4	S	START 时序发送完成中断标志，硬件置位，软件读取后清零(Start Interrupt flag)
3	P	STOP 时序发送完成中断标志，硬件置位，软件读取后清零(Stop interrupt flag)
2	ACKSTA	主控发送模式下，来自从机的回应信号；当主机发送后收到 NACK，此标志可以产生中断；硬件置位，软件写 1 清零。(Acknowledge Status Flag ,write 1 to clear) 1：从机回应 NACK 0：从机回应 ACK
1	TXIF	I2C 主机发送完成中断标志，硬件置位，软件写 1 清零(Trasnmit done interrupt flag,write 1 to clear) 此标志寄存器在主机接收完从机回发的 ACK 或 NACK 后置位。
0	RXIF	I2C 主机接收完成中断标志，硬件置位，软件写 1 清零(Receive done interrupt flag,write 1 to clear) 此标志寄存器在主机回发完 ACK 或 NACK 后置位。

19.9.5 I2C 主机状态寄存器（I2C_MSPSR）

名称	I2C_MSPSR							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	BUSY	RW_	-	BF	-	ACKMO
位权限	U-0	R-0	R-0	U-0	R-0	U-0	R/W-0

位号	位名	说明
31:6	--	RFU：未实现，读为 0
5	BUSY	I2C 通信状态位 (Busy) 1：接口处于读写状态，正在进行数据传输， 0：已完成数据传输
4	RW_	I2C 传输方向状态位 (Read/Write) 1：主机从从机读取数据 0：主机向从机写入数据
3	-	RFU：未实现，读为 0
2	BF	缓冲器满状态位 (Buffer Full) 接收： 1：接收完成，MSPBUF 满 0：接收未完成，MSPBUF 空 发送： 1：正在发送，MSPBUF 满 0：发送完成，MSPBUF 空
1	-	RFU：未实现，读为 0
0	ACKMO	主控接收模式下，主机回应信号的状态 (Ack Master output) 1：主机回发 NACK 0：主机回发 ACK ***注意：必须在 P 标志寄存器被清零的情况下，软件才能置位 ACKMO***

19.9.6 I2C 主机波特率设置寄存器（I2C_MSPBGR）

名称	I2C_MSPBGR							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							MSPBRGH[8]
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MSPBRGH[7:0]							
位权限	R/W-00010011							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							MSPBRGL[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSPBRGL[7:0]							
位权限	R/W-00010011							

位号	位名	说明
31:25	--	RFU：未实现，读为 0
24:16	MSPBRGH	主机发送的 SCL 时钟高电平宽度，以 I2C 工作时钟计数 (Master SCL High level length)

15:9	-	RFU：未实现，读为 0
8:0	MSPBRGL	主机发送的 SCL 时钟低电平宽度，以 I2C 工作时钟计数 (Master SCL Low level length)

19.9.7 I2C 主机收发缓冲寄存器（I2C_MSPBUF）

名称	I2C_MSPBUF							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SSPBUF							
位权限	R/W-00000000							

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7:0	MSPBUF	MSPBUF[7:0]：数据的读写通过对 MSPBUF 的操作完成。发送时，对 MSPBUF 执行写操作，同时也载入数据收发移位寄存器(MSPSR)；接收时，MSPBUF 与 MSPSR 组成双缓冲结构，读出数据为 MSPBUF 的数据。接收完一个字节的数据，MSPSR 将数据载入 MSPBUF，同时置位 I2CIF。MSPSR 不是直接寄存器，没有物理地址。 (Master data Buffer)

19.9.8 I2C 主机时序控制寄存器（I2C_MSPTCR）

名称	I2C_MSPTCR							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							SDAHD[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SDAHD[7:0]							
位权限	R/W-00001010							

位号	位名	说明
31:9	--	RFU：未实现，读为 0
8:0	SDAHD	定义 SDA 相对于 SCL 下降沿的保持时间参数，以 I2C 工作时钟计数 (SDA hold delay) 注意：最小有效值为 1，最大有效值为 MSPBRGL

19.9.9 I2C 主机超时寄存器（I2C_MSPTOR）

名称	I2C_MSPTOR							
Offset	0x00000020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名					TIMEOUT[11:8]			
位权限					R/W-1111			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TIMEOUT[7:0]							
位权限	R/W-1111 1111							

位号	位名	说明
31:12	--	RFU：未实现，读为 0
11:0	TIMEOUT	定义从机 SCL 低电平延展超时周期，软件可以在 MSPEN=0 的情况下改写 (SCL stretching Time Out) $T_{SCL_STRETCHING_TIMEOUT}$=TIMEOUT[11:0] * T_{SCL}

20 通用异步收发传输器（UART）

20.1 概述

UART串行通信模块特点如下

- 波特率软件可配置
- 6路独立通道（UART0, UART1,UART2, UART3, UART4, UART5）
- 全双工通信口
- 支持半双工单线通信
- 支持RX-TX端口交换
- UART具有数据接收完成/接收错误中断，并提示错误类型
- 可配置数据长度，支持6、7、8、9bits
- 可配置的停止位-支持1个停止位或2个停止位
- 可配置为红外调制输出功能，且载波频率可设置，及载波占空比可设置
- 支持DMA
- 支持接收超时机制
- 支持发送延迟功能

UART支持LIN通信，包括以下特性：

- 唤醒信号的检测和发送
- 同步间隔域（Break Field）检测
- 同步域（Sync Field）检测和波特率自适应

20.2 结构框图

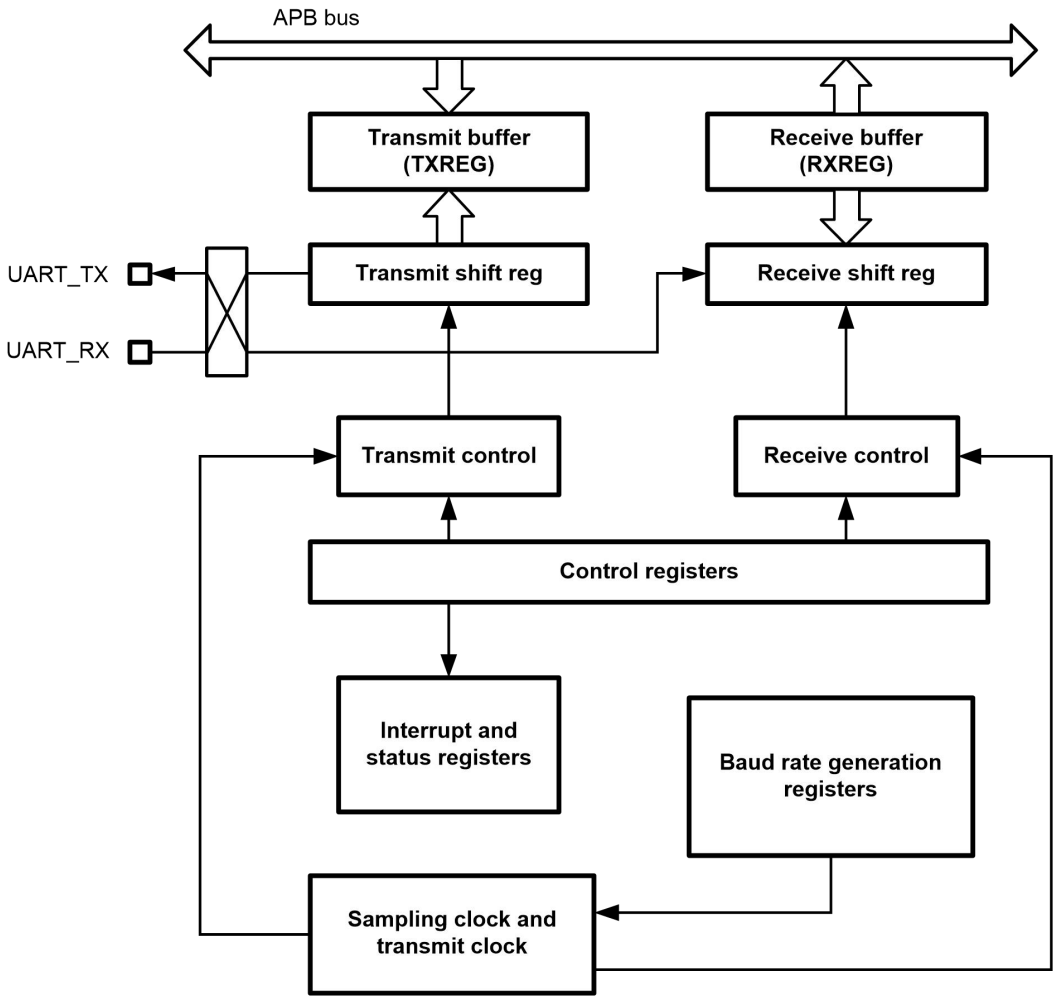


图 20-1 UART 接口时序

20.3 UART 类型区分

芯片集成了多种不同类型的UART（LPUART），其差异如下表所示：

UART 特性	UART (UART0/1/2/3)	USART (UART4/5)	LPUART
DMA 支持	Y	Y	Y
半双工/全双工	Y	Y	Y
红外发射	Y	Y	-
双时钟域（工作时钟独立于总线）	Y	-	Y
半双工单线通信	Y	Y	-
休眠唤醒	Y	-	Y
接收超时	Y	Y	-
发送延迟	Y	Y	-
数据长度	6、7、8、9bits		
LIN support	Y	N	
智能卡模式	N	Y	N

表 20-1 UART 类型列表

20.4 UART 字符描述

UART 传输字符的基本时序如下图所示。每个字符帧包含至少 1bit START 位和至少 1bit STOP 位，数据长度可以配置为 6~9bits，并且可以选择有无校验位。

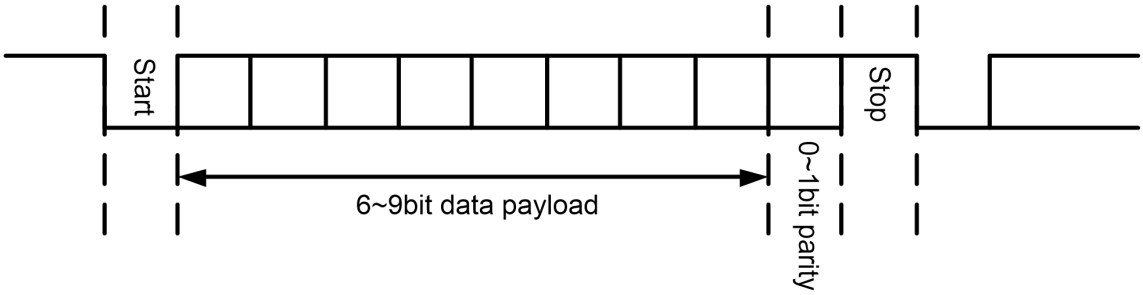


图 20-2 UART 字符描述

UART 支持多种帧格式，由 UARTxCSR.PDSEL 寄存器和 UARTxCSR.PARITY 寄存器控制。见下表：

PDSEL	PARITY	帧格式[1]
00	00	[Start | 7 bits data | Stop]
	01, 10	[Start | 7 bits data | Parity | Stop]

01	00	[Start | 8 bits data | Stop]
	01, 10	[Start | 8 bits data | Parity | Stop]
10	00	[Start | 9 bits data | Stop]
	01, 10	[Start | 9 bits data | Parity | Stop]
11	00	[Start | 6 bits data | Stop]
	01, 10	[Start | 6 bits data | Parity | Stop]

表 20-2 UART 数据帧格式

[1]：Stop 位可能是 1bit 或者 2bit，根据 STOPCFG 寄存器决定

注意 PDSEL 寄存器用于配置帧的数据长度，通信帧长为【起始位+数据位+校验位+停止位】。

20.5 功能描述

20.5.1 时钟结构

UART0 和 UART1 采用了多时钟结构：

- 总线寄存器时钟用 PCLK 表示，来源于 APBCLK。当 CPU 或者 DMA 需要访问 UART 内部寄存器时，必须使能 PCLK
- 数据收发时钟用 UCLK 表示，除了可以来源于 APBCLK，还可以来源于 RCHF、SYSCLK、RC4M，能够独立于 APBCLK 工作。必须使能 UCLK 才能进行数据收发。
- LINCLK 用于实现数据收发以外的 LIN 功能，比如唤醒检测/发送、间隔域检测/发送。LINCLK 使用 LSCLK 工作。

PCLK 和 UCLK 的控制都在 CMU 模块内完成，进行 UART 通信前必须正确配置相应的 CMU 控制寄存器。

采用双时钟结构，可以使 UART0 和 UART1 的工作不受限于 APBCLK 的配置，当某些外设需要工作在很高的 APBCLK 频率上时，UART 仍可以工作在降低的频率上；或者反过来，CPU 工作在较低的频率上，也不影响 UART 以较高的波特率进行数据通信。

理论上 PCLK 和 UCLK 之间没有相对关系的约束，UCLK 可以快于或者慢于 PCLK。但是应用需要注意当两者频率相差较大时，CPU 或者 DMA 是否来得及进行数据搬运。

与 UART0 和 UART1 不同的是，UART2~5 采用单时钟结构，此时 UCLK=PCLK，UART 的数据收发时钟也是来源于 APBCLK 的。

20.5.2 同步策略

UART 支持两种同步策略：起始位同步和重同步。

当 RESYNCEN=0，UART 只执行起始位同步，时钟频率误差将在字节内累积。

当 RESYNCEN=1，UART 执行起始位同步，并在字节内每个下降沿执行重同步。

注意，针对 LIN2.1，协议规定从机不执行重同步，只执行起始位同步，因此应设置 RESYNCEN=0。

通过 start bit 下降沿采样可以实现起始位重同步。使用 UCLK 采样 RXD，当检测到起始位下降沿时，重同步波特率计数器，以实现波特率同步于起始位检测，如下图所示。

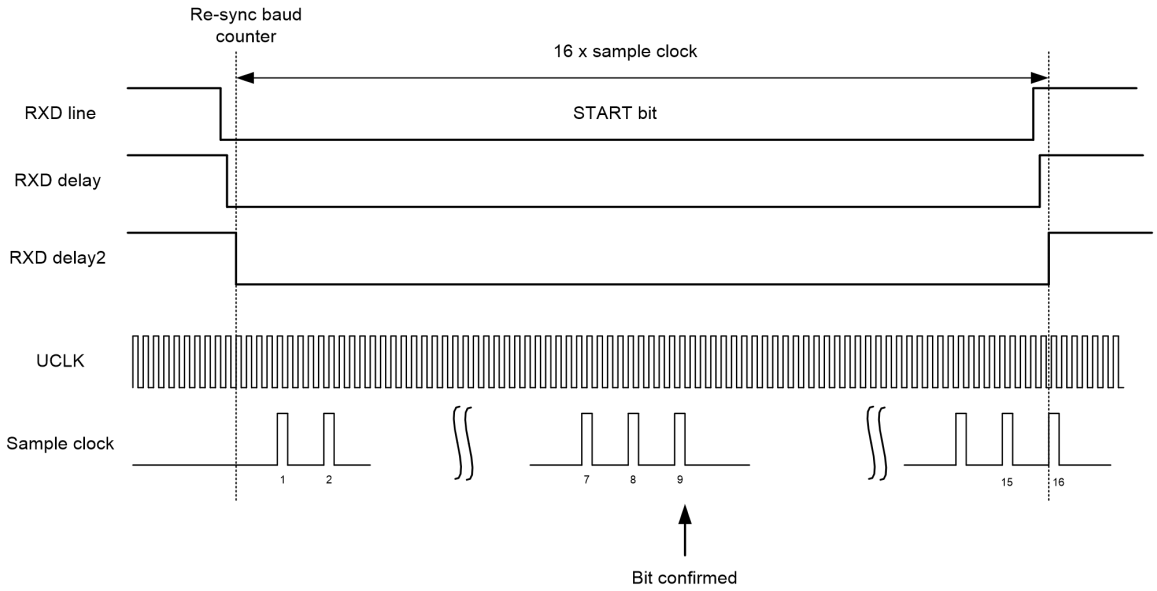


图 20-3 起始位同步

20.5.3 位接收采样

UART 对接收数据进行波特率的 8 倍或 16 倍过采样，并在每个 bit 的中间位置进行三中取二的多数判决，以提高对信号噪声的抑制能力。

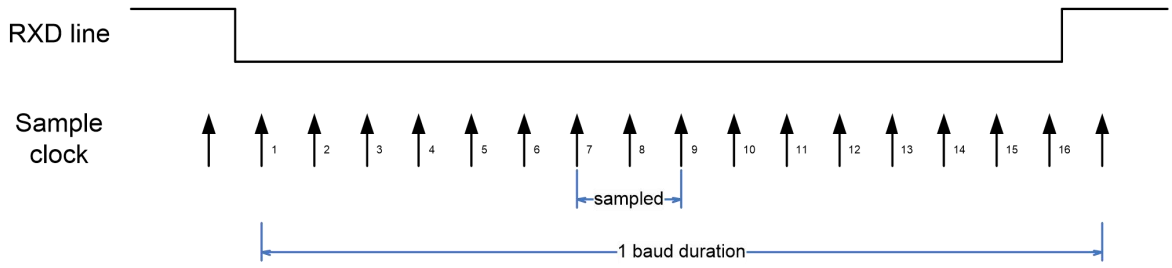


图 20-4 位接收 16 倍采样

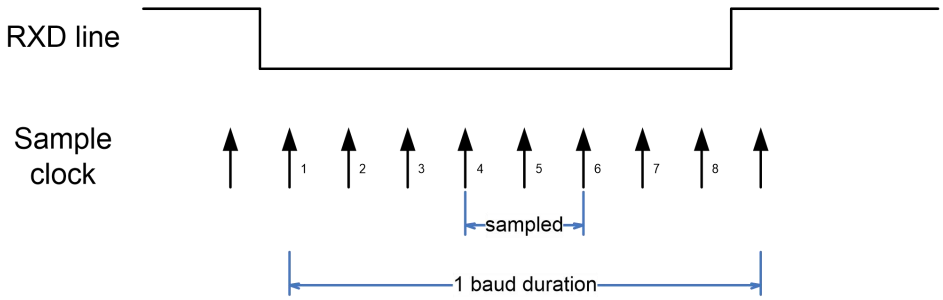


图 20-5 位接收 8 倍采样

接收移位寄存器收到的 bit 位是多数判决的结果。例如三次采样结果是 001，则判决为 0；如果是 011，则判决为 1。

通过 OVSM 寄存器可以配置 UART 接收数据时的过采样倍数。如果 UART 对输入信号进行 16 倍过采样，要求 SPBRG 配置不能小于 16，即 UART 工作时钟必须至少是波特率的 16 倍。如果 UART 对输入信号进行 8 倍过采样，要求 SPBRG 配置不能小于 8，即 UART 工作时钟必须至少是波特率的 8 倍。

选择较小的过采样倍数时，可以获得较高的通信波特率。

20.5.4 数据发送

在发送模式下，UART 的串行数据发送电路主要包括一个发送移位寄存器(TSR)，TSR 功能是将数据逐个移位送出。待发数据必须先写到发送缓冲区中。当软件置位 TXEN 寄存器后，如果发送缓冲区非空，UART 将缓冲区数据载入 TSR 并开始移位输出。

注：由于寄存器操作时钟和波特率时钟是异步关系，当发送开始时，需要等待波特率时钟到来，因此从 TXEN 置位到 UART 开始发送 Start 位之间，有最大 1 个 baud 的延迟。

TXBE 和 TXSE 是发送中断标志位，分别表示发送缓冲区空和 TSR 空，软件可以选择在合适的时间点产生发送完成中断。

一般情况下，一开始 TSR 寄存器是空的，数据的发送需先设定波特率 SPBRG，使能发送模块(设定 TXEN 为 1)，然后写入 TXBUF 寄存器开始发送。也可以在设定好波特率 SPBRG 后，先写入 TXBUF 寄存器，然后再设定 TXEN 使能发送模块来开始数据发送。如果在数据发送过程中将发送模块使能位 TXEN 清 0，那么数据发送工作就会被中断，发送模块也会被复位。

下图为UART异步发送的例子。这个示例中软件首先向TXBUF写入数据，然后通过置位TXEN启动发送。

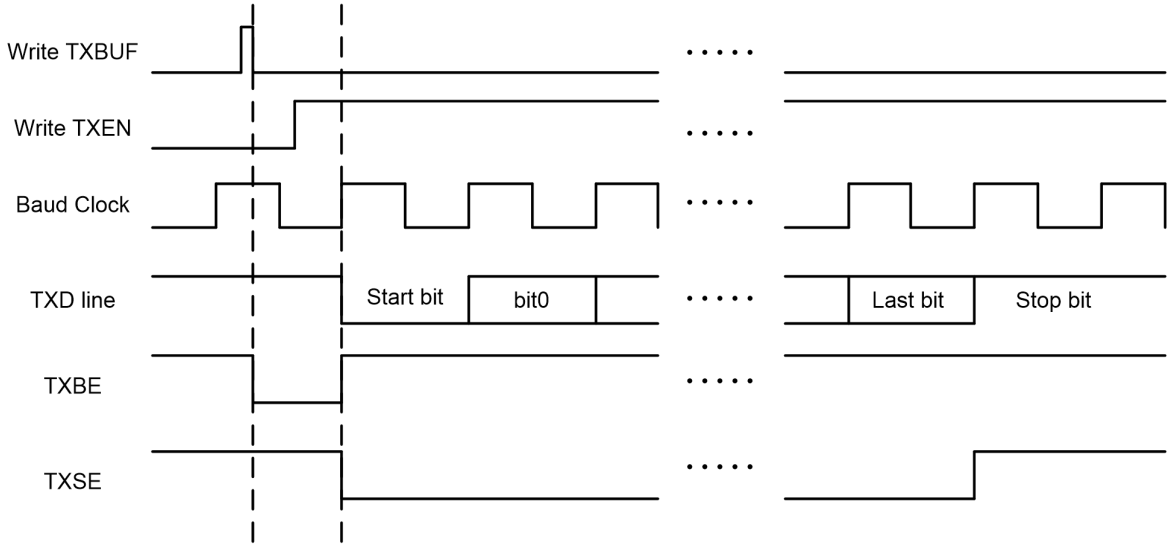


图 20-6 UART 异步发送波形 1

上图中推荐的操作步骤如下：

- 选择合适的波特率，初始化 SPBRG
- 若需要中断，置位 TXSE_IE 或者 TXBE_IE
- 决定数据发送的格式：设置 PDSEL 寄存器，决定发送数据长度；设置 PARITY 寄存器选择是否发送校验位以及校验类型，设置 STOPSEL 寄存器决定发送 1 位还是 2 位停止位
- 如果希望发送的串行数据红外调制，向 IRCON 寄存器写入合适的值来获得相应的调制频率和占空比，并置位 TXIREN
- 将待发送的数据写入 TXBUF 寄存器（自动启动发送）
- 使能发送模块：置位 TXEN

软件也可以先置位TXEN再写入TXBUF，此时UART会在数据写入TXBUF后立刻开始发送流程。

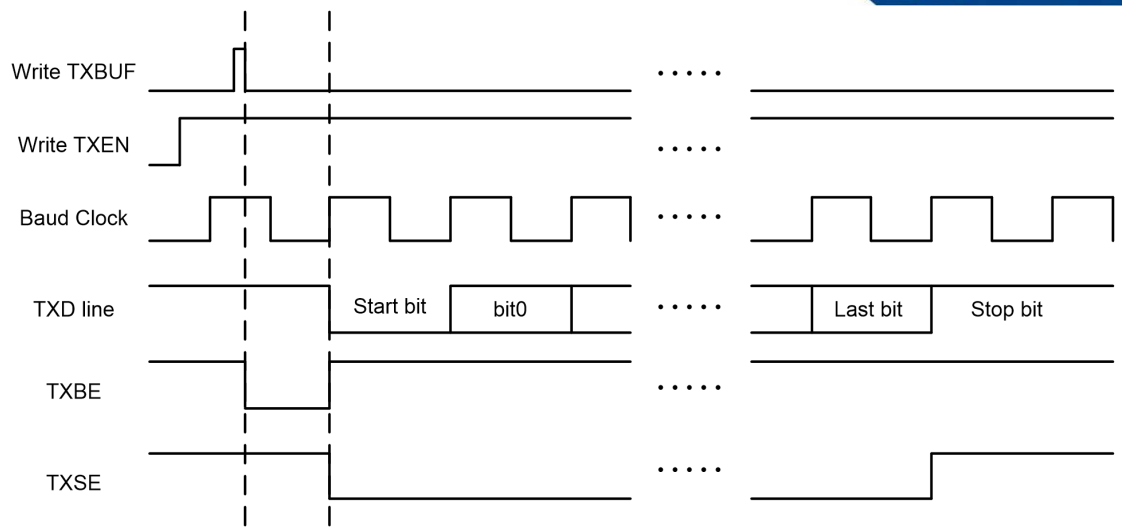


图 20-7 UART 异步发送波形 2

当TXBUF为空时，软件可以立即写入下一个待发送数据，以实现连续无间隔的数据发送。

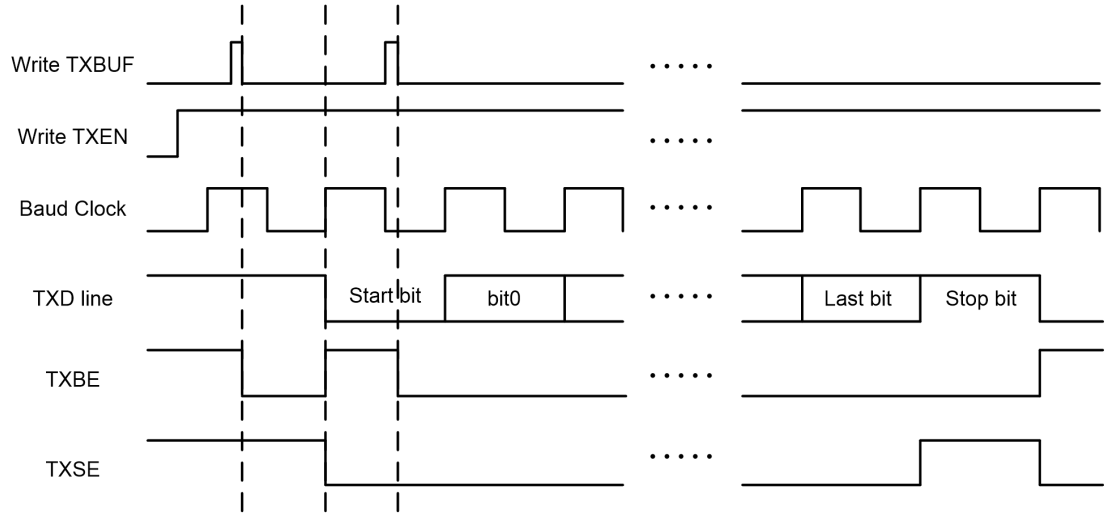


图 20-8 UART 异步发送波形 3

20.5.5 数据接收

UART 的串行数据接收电路主要包括一个接收移位寄存器(RSR)。当接收到停止位后，RSR 就把接收到的数据送入接收缓冲区(RXBUFFER)，传送完成后，在每次接收数据送入接收缓冲区后将中断标志 RXBF 置 1。当接收缓冲区已满时，RSR 接收到一帧数据后仍会将其写入接收缓冲区，即覆盖缓冲区中原有数据，并且再次置位 RXBF，同时发生接收溢出错误，OERR 被置 1；软件写 1 或者读取 RXBUF 都可以清除 OERR 标志。

接收过程中，如果没有检测到正确的停止位，则发生帧格式错，FERR 被置 1；如果发生奇偶校验错，标志位 PERR 被置 1。

推荐的异步接收操作如下：

- 选择合适的波特率，初始化 SPBRG
- 若需要中断，置位 RXBF_IE
- 设置数据接收的格式：设置 PDSEL 寄存器，决定发送数据长度；设置 PARITY 寄存器选择是否发送校验位以及校验类型，设置 STOPSEL 寄存器决定发送 1 位还是 2 位停止位
- 使能接收模块：置位 RXEN
- 在一帧接收完毕时，RXBF 位会置 1，如果 RXBF_IE 位为 1，将会产生中断
- 读取 PERR、FERR、OERR 寄存器，判断是否有数据错误或者溢出
- 读取 RXBUF 寄存器中的接收数据

20.5.6 RS232 硬件流控

UART模式下，通过nCTS输入和nRTS输出，可以实现两个设备之间的数据流控制，如下图：

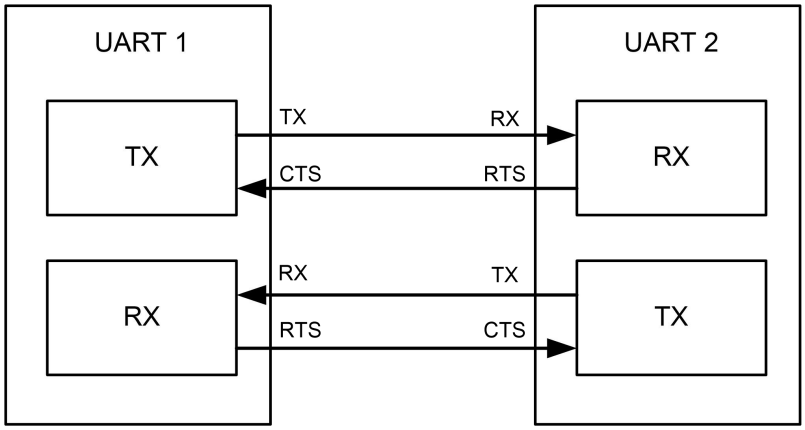


图 20-9 两个 UART 之间的硬件流控

RS232 RTS和CTS流控功能由RTSEN和CTSEN寄存器分别独立控制。

当RTSEN=1，nRTS在UART接收电路准备就绪时保持低电平，当UART数据缓冲区满且接收移位寄存器满，此时无法继续接收数据，nRTS信号拉高指示对方暂停发送新数据。当数据缓冲区被读取后，nRTS重新拉低通知对方可以发送新数据。

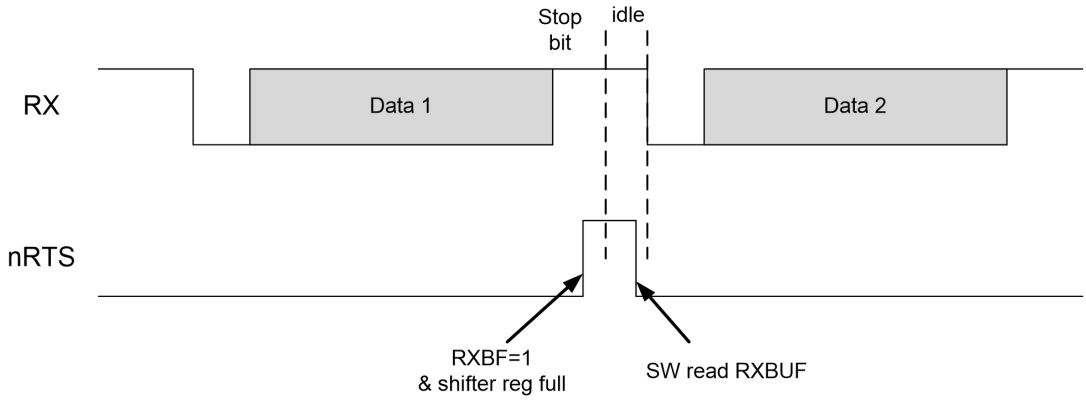


图 20-10nRTS 流控

当CTSEN=1，UART发送数据前先检查nCTS输入，如果nCTS为低电平，表示对方可以接收数据，此时UART正常发送；如果nCTS为高电平，表示对方暂时不能接收，则UART推迟发送直到nCTS变成低电平。在CTSEN=1的情况下，nCTS输入翻转时CTSIF标志自动置位，并且可以产生中断。

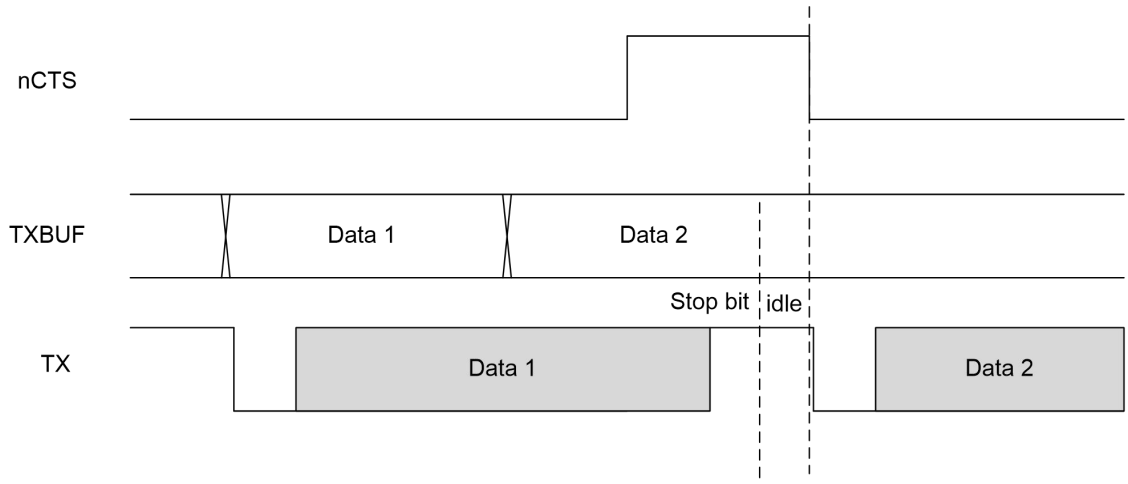


图 20-11nCTS 流控

20.5.7 低功耗休眠唤醒

UART支持两种唤醒模式

1）UART模式：RXD下降沿触发的芯片休眠唤醒。

UART模式下，置位了NEWUP寄存器之后，RXD输入上的下降沿事件（低电平持续时间>100ns）将会使芯片从休眠模式下唤醒，借助这个功能，可以实现UART在休眠模式下接收数据。

软件配置方法如下：

- 配置UART寄存器，使能NEWUP
- 配置UART工作时钟为RCHF，根据需要配置波特率分频寄存器
- 将对应GPIO配置为UART数据接收功能

- 置位RXEN，使能接收
- 软件设置芯片进入休眠，等待UART接收事件

2）LIN模式：从机支持总线唤醒信号检测。

LIN从机模式下，通过置位WKDET_EN寄存器，LIN模块持续检测总线信号，当出现隐性到显性电平转换，并保持显性电平时间长度符合LIN协议要求后，触发LIN唤醒事件。

软件配置方法如下：

- 配置UART寄存器，使能WKDET_EN
- 将对应GPIO配置为UART数据接收功能
- 关闭UART发送和接收使能
- 软件设置芯片进入休眠，等待LIN唤醒事件

20.5.8 使用 DMA 进行 UART 收发

当 UART 模块被使能后，UART 模块在发送缓冲寄存器空和接收缓冲寄存器满时都会自动产生相应的 DMA 请求。应用软件需要事先配置 DMA 通道连接，将特定通道指向 UART 外设，设置 RAM 访问的指针地址，并使能 DMA 通道。此后 DMA 会自动响应 UART 请求，并完成 RAM 和 UART 之间的数据搬运。

应用举例：使用 DMA 进行 UART0 接收

- 将 DMA 通道 1 或 3 配置为 RXD0
- 设置对应通道参数：RAM 指针地址、地址递增递减、通道优先级、传输长度和中断设置等
- 使能对应 DMA 通道
- 配置 UART 模块参数
- 使能 UART 模块接收使能，等待数据接收
- 收到数据后 UART 自动产生 DMA 请求
- DMA 响应请求，读取 UART 接收缓存寄存器，写入指定 RAM 地址

20.5.9 DMA 模式下的发送完成中断

当 UART 通过 DMA 进行数据发送时，DMA 会在指定长度的数据传输完成后产生 DMA 通道中断。但是当通道中断产生时，最后一帧数据刚刚被写入 UART 发送缓冲区，还未被发送出去。

通过配置 DMATXIFCFG 寄存器，可以实现 DMA 传输完成、并且最后一帧数据发送完成的情况下，产生一个发送完成中断（缓冲区空或者移位寄存器空），以便实现所有数据全部发送出去后，再中断 CPU 的应用场景。

软件工作流程说明如下：

- 配置DMA通道为UART发送
- 关闭DMA通道中断使能
- 置位DMATXIFCFG寄存器，仅允许最后一帧数据产生中断输出
- 准备待发送数据，使能DMA
- 置位UART TXBE_IE或TXSE_IE寄存器，允许中断产生
- UART连续发送，直到最后一帧，发送期间不会产生TXBE或TXSE中断
- 最后一帧发送完成后，UART产生TXBE或TXSE中断

下表假设 UART 通过 DMA 发送 N 个帧：

TXBE_IE TXSE_IE	DMATXIFCFG	Frame No.	TXBE TXSE	UART interrupt
0	x	1~N	每帧发送完成后置位	不产生
1	0	1~N	每帧发送完成后置位	不产生
	1	1~N-1	每帧发送完成后置位	不产生
		N	每帧发送完成后置位	产生

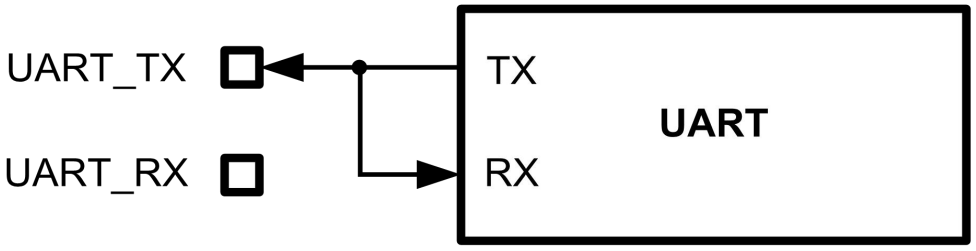
表 20-3 DMA 发送中断

20.5.10 半双工单线通信

当HDSEL寄存器置位时，UART以半双工模式工作，此时仅需一根信号线即可完成通讯。半双工模式仅在UART功能时可以使用，禁止在LIN和智能卡模式下使用，即置位HDSEL时必须保证MODE=0。

当HDSEL置位后：

- TX和RX信号在内部连接
- RX引脚不再需要使用，可以用于GPIO或者其他功能
- 当UART不发送数据时，TX引脚输出使能自动关闭，输入使能自动打开，引脚保持浮空状态，需要通过上拉电阻赋予确定电平
- TX引脚可以配置为开漏输出，数据冲突必须由软件处理
- 数据发送时UART自动屏蔽数据接收



20.6 波特率发生

20.6.1 波特率发生

波特率因子寄存器是一个 16 位的可读写的寄存器，其值 X 为 16—65535 之间的任一整数。

UART 波特率计算公式：

Baud = F_{CLK} / (SPBRG+1)；

注：F_{CLK} 在不同的 UART 中可以是不同的时钟，对于 UART4 和 UART5，F_{CLK} 就是 APBCLK；对于 UART0 和 UART1，F_{CLK} 是独立与 APBCLK 的工作时钟。

为了支持全双工通信，接收和发送波特率单独产生；

下表是常用系统时钟频率下的波特率：

Baud	F_{CLK}=16MHz			F_{CLK}=8MHz		
bps	Actual (bps)	Error%	X+1	Actual (bps)	Error%	X+1
300	300.0019	0.000625	53333	299.9963	-0.00125	26667
1200	1200.03	0.0025	13333	1199.94	-0.005	6667
2400	2399.88	-0.005	6667	2400.24	0.010001	3333
4800	4800.48	0.010001	3333	4799.04	-0.02	1667
9600	9598.08	-0.02	1667	9603.842	0.040016	833
19200	19207.68	0.040016	833	19184.65	-0.07994	417
38400	38369.3	-0.07994	417	38461.54	0.160256	208
57600	57553.96	-0.07994	278	57553.96	-0.07994	139
115200	115107.9	-0.07994	139	115942	0.644122	69
230400	231884.1	0.644122	69	228571.4	-0.79365	35
460800	457142.9	-0.79365	35	470588.2	2.124183	17

Baud	F_{CLK}=24MHz			F_{CLK}=32MHz		
bps	Actual (bps)	Error%	X+1	Actual (bps)	Error%	X+1
300	300	0	80000	299.9991	-0.00031	106667
1200	1200	0	20000	1199.985	-0.00125	26667
2400	2400	0	10000	2400.06	0.0025	13333
4800	4800	0	5000	4799.76	-0.005	6667
9600	9600	0	2500	9600.96	0.010001	3333
19200	19200	0	1250	19196.16	-0.02	1667
38400	38400	0	625	38415.37	0.040016	833
57600	57553.96	-0.07994	417	57553.96	-0.07994	556
115200	115384.6	0.160256	208	115107.9	-0.07994	278
230400	230769.2	0.160256	104	230215.8	-0.07994	139
460800	461538.5	0.160256	52	463768.1	0.644122	69

表 20-4 常用时钟频率下波特率计算

20.6.1 波特率自适应

利用 Timer 的 Capture 功能，可以实现波特率自适应功能。可实现的一种方法为，外部 UART 设备按约定的数据内容(比如 0xF8)发送一帧，由 Timer 对该帧数据的高电平脉宽进行计数，MCU 读取 Timer 捕捉结果计算得到波特率因子，并写入波特率发生寄存器中，作为波特率发生的时钟分频计数值 X 使用。这时接收状态复位，重新等待起始位，以写入的波特率因子所产生的波特率接收数据。参考 Timer 章节。

20.7 红外调制

TZBRG 寄存器保存一个 11 位的分频系数 X，其值为 0~2047 之间的任一整数。所有 UART 共用一个红外调制频率发生器。

红外调制频率计算公式：

$FIR = F_{APBCLK} / (TZBRG+ 1)$

红外调制的方式为：发送数据 0 时调制红外频率，发送数据 1 时不调制。为满足 PNP 和 NPN 两种红外驱动管的需求，寄存器 IRFLAG 位控制红外调制输出的极性。IRFLAG=0 时为正极性输出，适合 PNP 管驱动；IRFLAG=1 时为负极性输出，适合 NPN 管驱动。

TH 寄存器用于配置红外调制占空比

占空比：Y = (TZBRG[10:4] * TH) /(TZBRG+1)

当 TH=4’b0000 时，占空比为 Y = (TZBRG[10:1]+1)/(X+1)；

当 TZBRG[10:4]=7’h00 时，占空比为 Y = TH/(TZBRG[3:0]+ 1)；若此时 TH>TZBRG [3:0]，则红外调制时钟 IRCLK 为固定高电平。

当红外调制极性反向时（IRFLAG=1），占空比也为 1-Y

红外调制波形见下图：

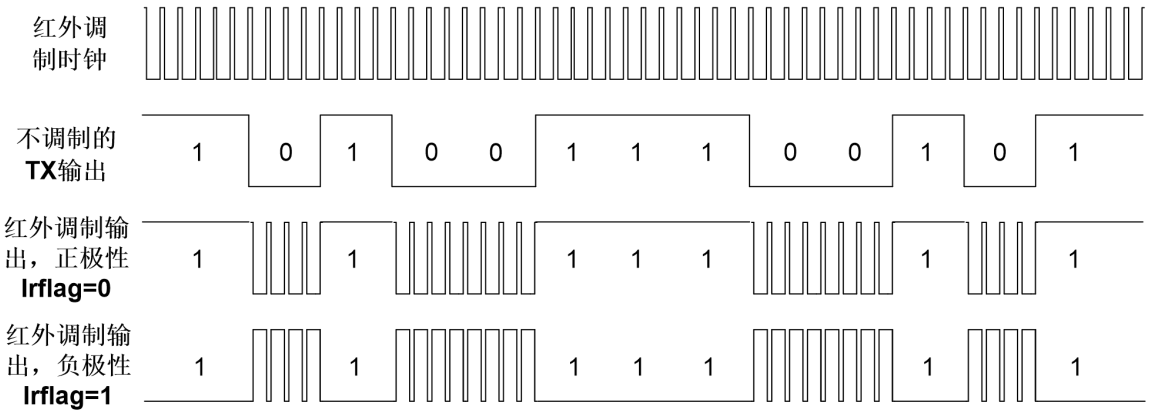


图 20-12 红外调制波形

无论有效电平是 0 还是 1，占空比定义为高电平长度/周期。

20.8 接收超时

针对 MODBUS 等时间敏感型应用，设计了接超时机制。当使能 RXTOEN 寄存器后，超时计数器以波特率时钟计数，当每次收到一个完整的数据帧，将清零超时计数器并重新开始计数。超时溢出的上限值可以由软件配置，最大 255 波特。

20.9 发送延迟

通过 TXDLY_LEN 寄存器，可以控制两个数据帧发送之间的间隔时间，单位是波特。发送延迟是从上一帧最后一个 STOP 位结束，到下一帧起始位之间的间隔。

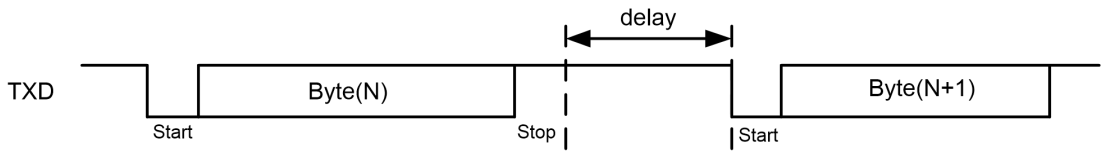


图 20-13 UART 发送延迟

20.10 LIN 总线通信功能

20.10.1 概述

LIN（Local Interconnect Network）是一个低速低成本串行通信协议，通信波特率1~20Kbps，支持单主机和最多15个从机互联。每个节点通过单线收发器接入总线，收发器与控制器之间通过类似UART的接口连接。

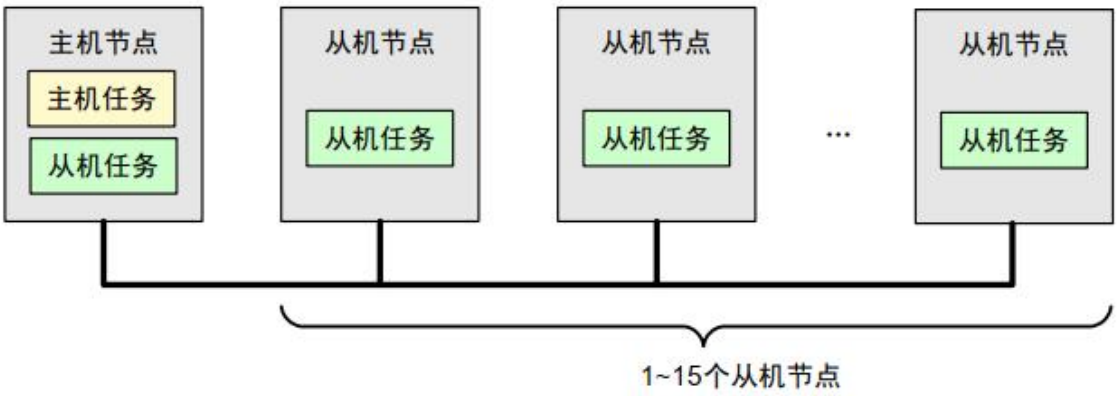


图 20-14LIN 总线拓扑

LIN总线的帧结构包含帧头和应答，帧头总是由主机发送，主机发送帧头后可以继续发送数据，也可以接收从机发送的应答数据。

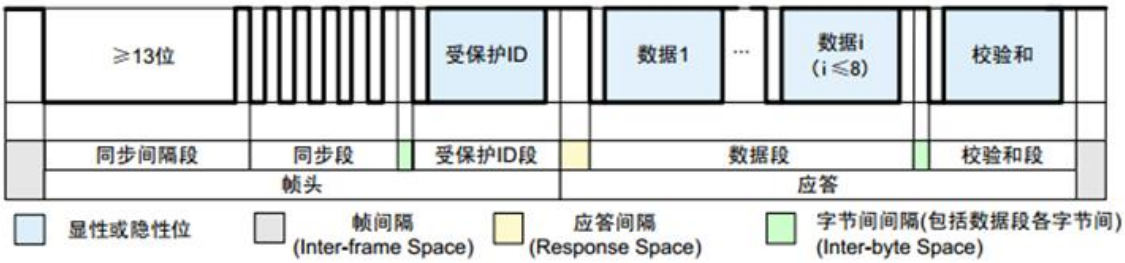


图 20-15LIN 总线帧结构

除了同步间隔段，其他各段都是通过标准UART数据传输格式的格式传输的。在 LIN 帧中，数据传输都是先发送LSB(Least Significant Bit，最低有效位)，最后发送 MSB(Most Significant Bit，最高有效位)。

20.10.1.1 Break Field

同步间隔段（Break Field）为至少13bit低电平和间隔字符组成。总线波特率总是由主机决定的，从机检测到总线上持续至少11bit的显性电平时，认为是帧的开始。



图 20-16LIN break field

20.10.1.2 Sync Field

同步段用于使从机完成波特率时钟同步，从而可以降低总线对从机时钟精度的要求。UART-LIN模块用作LIN从机时支持基于同步段的波特率自适应

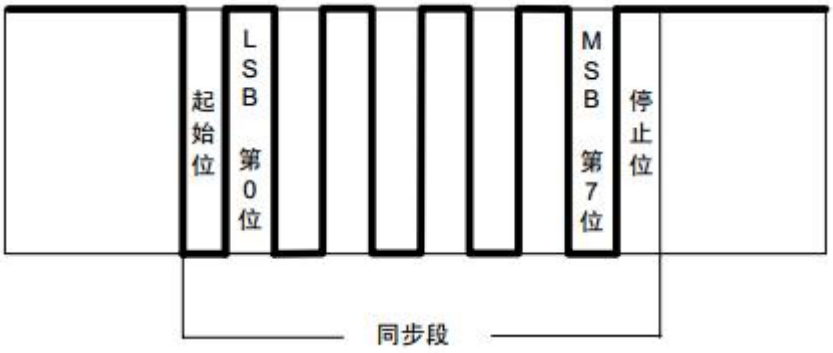


图 20-17LINsync field

LIN 同步以下降沿为判断标志，采用字节 0x55(转换为二进制为 01010101b)。

$$1位时间 = \frac{第7位的下降沿时刻 - 起始位的下降沿时刻}{8}$$

同步段结束后，从机软件根据计数值完成通信波特率的自动调整。

20.10.1.3 PID Field 和 Data Field

PID段的前 6 位叫作帧 ID(Frame ID)，加上两个奇偶校验位后称作Protected ID。

帧 ID 的范围在 0x00～0x3F 之间，共 64 个。帧 ID 标识了帧的类别和目的地。从机任务对于帧头作出的反应(接收/发送/忽略应答部分)都是依据帧 ID 判断的。如果帧 ID 传输错误，将会导致信号无法正确到达目的地，因此引入奇偶校验位。

$$P0 = ID0 \oplus ID1 \oplus ID2 \oplus ID4$$

$$P1 = \neg (ID1 \oplus ID3 \oplus ID4 \oplus ID5)$$

由公式可以看出，PID 不会出现全 0 或全 1 的情况，因此，如果从机节点收到了“0xFF”或“0x00”，可判断为传输错误。

帧的类型		帧 ID
信号携带帧	无条件帧	0x00 ～ 0x3B
	事件触发帧	
	偶发帧	
诊断帧	主机请求帧	0x3C
	从机应答帧(注 1)	0x3D
保留帧		0x3E，0x3F

注：1. 从机应答帧是一个完整的帧，与帧结构中的“应答”(帧的一部分)不同，注意区别。

图 20-18LIN 帧 ID

节点发送的数据位于数据段，包含 1 到 8 个字节，先发送编号最低的字节，编号依次增加。

协议没有规定帧中的哪一部分显示数据长度码的信息，数据的内容和长度是由系统设计者根据帧 ID 事先约定好的。

一般情况下，对于一个帧中的应答，总线上只存在一个发送节点，否则就会出现错误。事件触发帧例外，可能存在零个、一个或多个发布节点。

PID和数据段的收发与标准UART相同，LIN模块无需特殊处理。

20.10.1.4 Checksum Field

校验段对帧中所传输的内容进行校验，校验对象包含：

- 经典校验（LIN 1.x）：所有数据段
- 加强校验（LIN 2.x）：所有数据段以及PID段

校验方法为将校验对象的各字节作带进位二进制加法，并将所得最终的和逐位取反，以该结果作为要发送的校验和。接收方根据校验和类型，对接收数据作相同的带进位二进制加法，最终的和不取反，并将该和与接收到的校验和作加法，如果结果为 0xFF，则校验和无误，这在一定程度上保证了数据传输的正确性。

校验功能由软件实现。

20.10.2 LIN 主机操作

在主机模式下，LIN模块能够向总线发送唤醒信号、间隔段、同步段、数据段。

20.10.2.1 唤醒发送

唤醒信号由软件置位TX_WKUP寄存器发送，信号长度为WKUP_LEN寄存器定义时钟周期数，默认9个LINCLK时钟，约合270us。最大支持256个时钟周期，约7.6ms。LIN协议规定的唤醒信号长度是250us~5ms。当唤醒信号发送完成后，TX_WKUP寄存器被自动清零，同时LINWKTF发送完成标志置位。

20.10.2.2 Break 和 Sync 发送

Break field发送固定为13bit，由软件置位TX_BREAK寄存器发送。

同步段由软件发送0x55实现，数据段和校验段的发送与标准UART数据发送相同。

主机发送完Break和PID后也可以主动转入接收，此时数据接收与标准UART接收相同。

20.10.3 LIN 从机操作

从机模式下，LIN需要支持唤醒信号和间隔段接收，以及基于同步段的波特率自适应。使能LIN从机后唤醒信号和Break检测电路自动使能。

20.10.3.1 唤醒检测

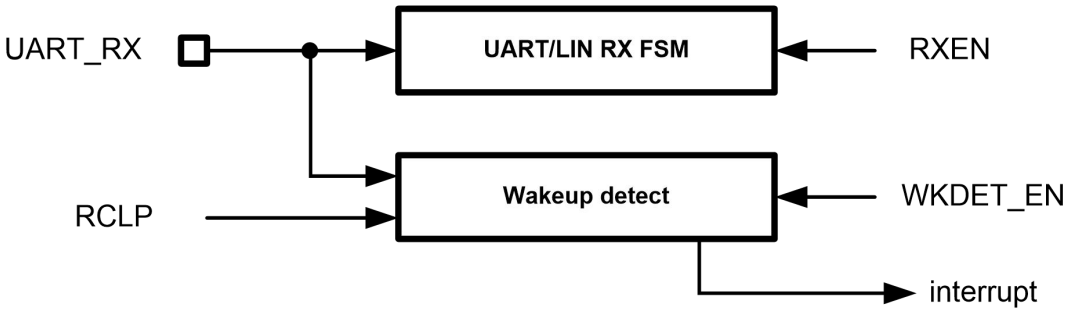
LIN总线处于休眠状态时，主/从机节点都可以向总线上发送唤醒信号，唤醒信号持续 250us～5ms。其余节点（除发送唤醒信号以外的节点）以大于150us为阈值判定唤醒信号。每个从机节点必须在唤醒信号显性脉冲的结束处算起100ms以内准备接收来自主机的命令(帧头)；主机节点也必须被唤醒，100ms之内主机节点发送帧头开始通信。



图 20-19LIN 总线唤醒信号

唤醒信号识别通过一个低速时钟驱动的计数器实现。当软件置位WKDET_EN寄存器后，计数器开始监视RXD输入，RXD为高电平情况下计数器保持复位，RXD下降沿启动计数器开始计数，当RXD保持低电平时间达到WKUP_LEN所定义的时钟个数时，认为唤醒信号有效，产生唤醒中断，并关闭计数器，直到下一个RXD下降沿到来。

唤醒计数器使用LINCLK工作，即RCLP，独立于APBCLK，因此这部分功能可以在休眠模式下使用。RCLP时钟周期大约为30us，软件可以通过WKUP_LEN寄存器设置唤醒计数器的阈值，满足最少150us即可。当RXD保持低电平长度超过WKUP_LEN设置后，LINWKF中断标志寄存器被置位，表示LIN模块识别到总线上有效的唤醒信号。



注意：

WKUP_LEN=N，实际能够唤醒从机的最短电平长度是N-2个RCLP时钟周期。

唤醒条件满足后，唤醒标志将在低电平采样值=N时置位。

唤醒检测仅用于产生标志，当芯片处于休眠模式时，LIN唤醒检测标志可以用于唤醒MCU。

在LIN模块处于正常通信时，应关闭唤醒检测功能（WKDETEN=0）。

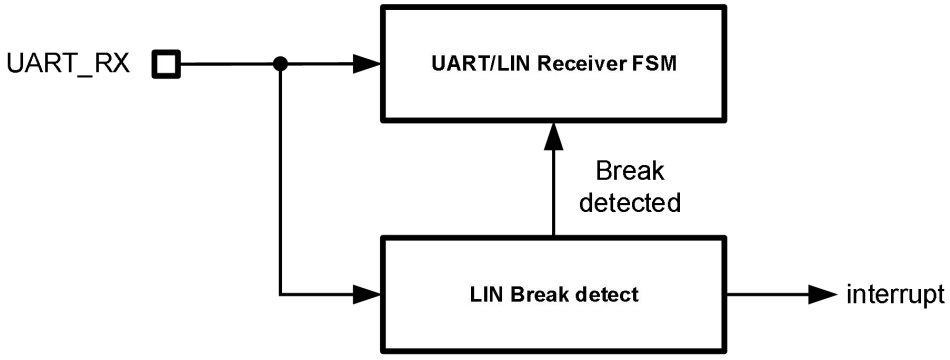
软件应仅在关闭UART收发使能的情况下，打开LIN唤醒检测。

举例：

应用希望检测大于150us的唤醒信号，相当于大约5个RCLP周期，此时需要将WKUP_LEN设置为7，则接收到WKUP低电平长度大于5个RCLP周期时即认为唤醒有效，而唤醒检测标志将在下降沿之后第7个RCLP处置位。

20.10.3.2 Break 检测

Break检测模块独立于LIN从机接收，因此在LIN通信的任何阶段，都能够识别主机发送的Break，并将接收状态机置于初始状态，开始新的一帧接收。



从机处于唤醒状态时（LIN_SLEEP=0），首先要接收Break段。从机根据设定的波特率采样RXD，当接收超过连续11bit低电平，认为检测到合法的Break并置位break检测中断，Break结束中断将在Break结束后置位。

Break低电平阶段使用软件预设的波特率来进行检测，低电平长度超过11 baud即置位break detected标志，检测到上升沿后置位break ended标志，同时记录收到的Break的波特数；随后在SYNC段中，根据前2个下降沿的时间差计算主机baud值（参见Pre-Sync），使用这个值重新验证收到的Break长度是否满足超过11 baud，即应满足：

preset-baud-rate * break-baud-count >sync-baud-rate * 11

如果上式不满足，则置位break error寄存器。

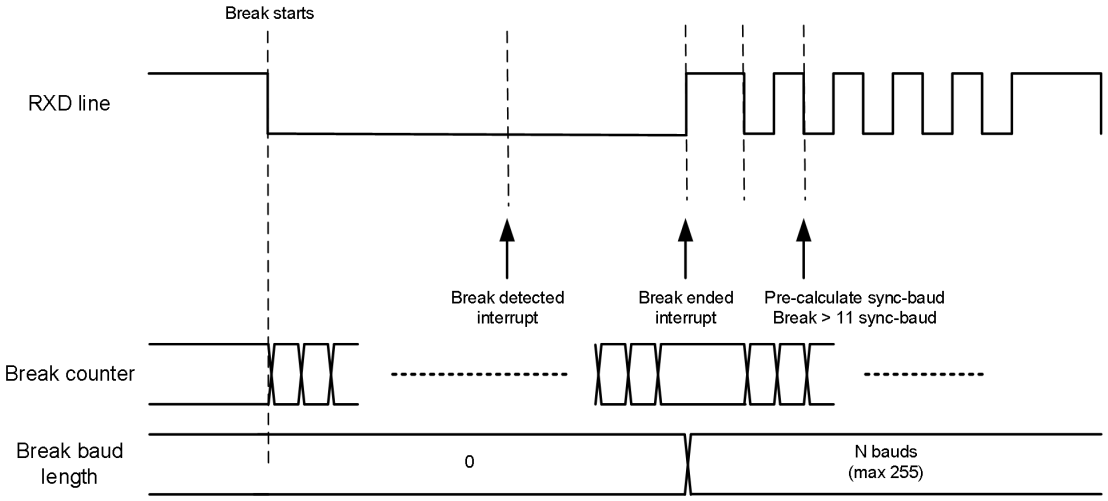


图 20-20Normal Break

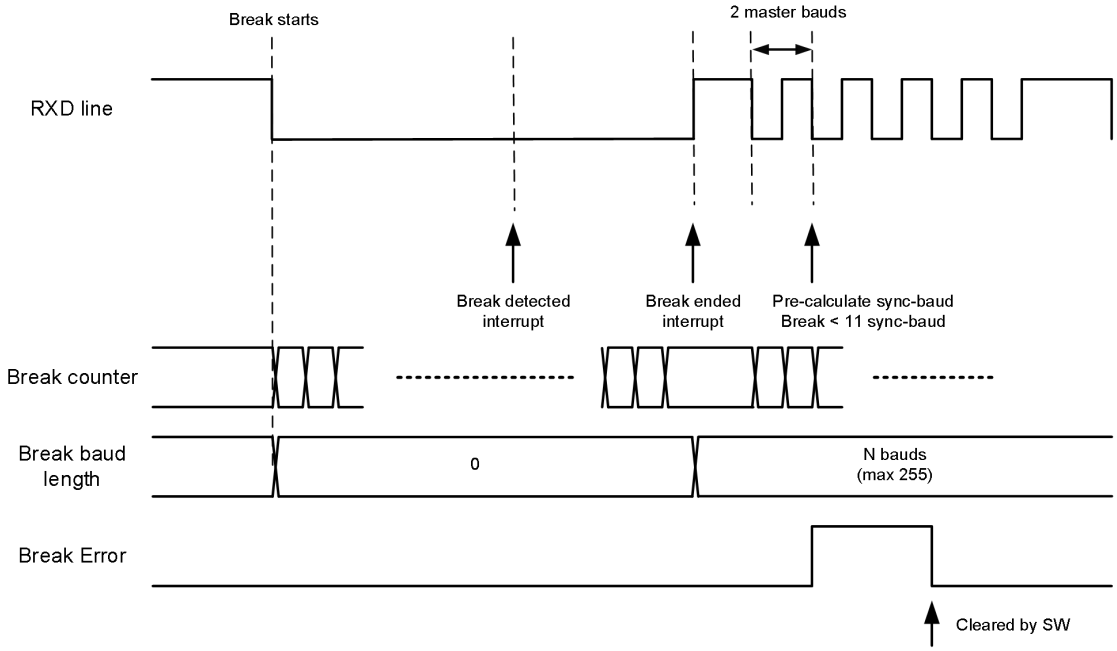


图 20-21Break Error

由于协议没有规定break长度上限，当break持续很长时间显性电平时，LIN模块支持timeout事件中断。timeout长度由寄存器定义（单位baud），通常规定：

timeout time = 48 + (number of data bytes + 1)*14 [baud]

timeout定时器在LIN模块检测到Break下降沿时开始计数，在收到校验和后清零。如果发生了timeout事件，LIN模块置位超时中断、停止当前frame接收，并回到IDLE等待下一个break的下降沿。

因此对于超长break段，LIN将产生超时中断，而不会进行后续的frame接收。发生Break超时后，LIN从机回到IDLE状态，等待下一个RXD下降沿到来。

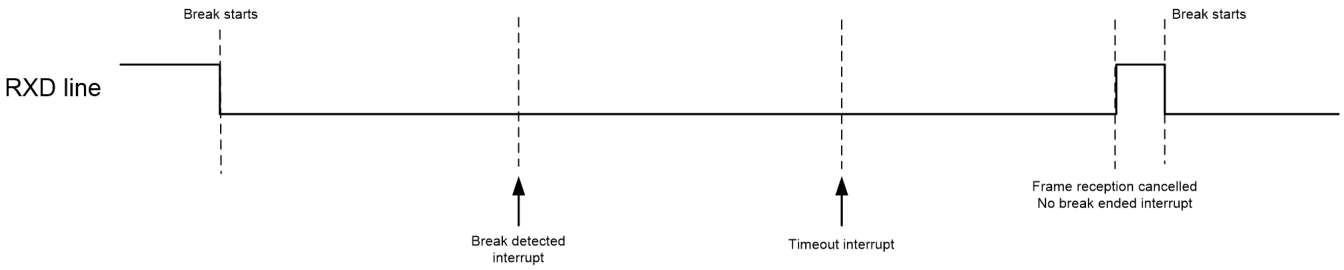


图 20-22Break 超时

与Break接收相关的错误状态：

- Break过长：报超时错误
- Break过短：不识别为Break，LIN从机状态机不动作
- 不发Break直接发送数据：效果等同于Break过短
- 在数据接收过程中收到Break：触发帧格式错误（stop=0，停止位错误）

与Break发送相关的错误状态：

- 发送电平与回环电平不一致：报biterror

特殊情况：

主机发送Break后没有发送SYNC，而是重新发送Break。为了应对这种特殊情况，LIN从机的Break检测应该是并行于接收状态机的，即从机可以在任何状态下识别新的Break并视为主机的一次重新发送过程，此时应终止当前接收过程，准备接收新的帧。

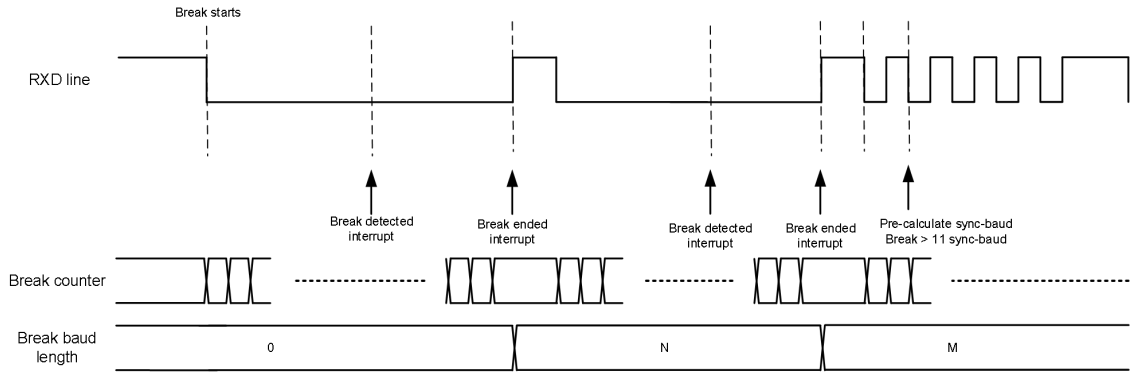


图 20-23 连续 Break 后正常接收

20.10.3.3 波特率同步

在波特率同步状态下，软件置位SYNCEN使能波特率自适应电路，波特率检测模块测量同步段的宽度，软件计算波特率，同步段结束后软件更新波特率寄存器。同步段接收过程中接收电路不工作，

不会将任何数据写入RX DATA寄存器。

如果软件清零SYNCEN，则LIN从机不识别Sync段，不做波特率计算，如果主机发送sync段，从机将作为数据接收。SYNCEN=0可以在必要时改用Timer来实现软件处理的波特率自适应，处理完成后再由LIN硬件进行数据收发。

由于LIN波特率最低1Kbps，考虑到系统主频较高的情况下，需要较长的计数器才能计算8bit总长度，将波特率同步计数器位宽扩展到20bit，这样在波特率计数器为16bit的情况下，足够容纳8个最长baud位的累加长度。

Pre-Sync操作

由于从机和主机之间可能存在较大的初始波特率误差，为了提高同步段识别的正确率，从机首先计算同步段前2个下降沿之间的时间间隔，得到一个初始波特率，后续同步段的接收识别使用这个初始波特率进行，如果识别到正确的0x55，从机将同步段内记录的平均波特率自动更新到BAUD寄存器。如果没有这个功能，当主机从机波特率误差较大，如10%（仍小于协议要求的14%），从机将无法正确接收0x55，此时会导致SYNC段错误。

Pre-sync阶段的2bit是使用预定义波特率接收的，这里的隐含假设是双方误差不会大到影响2bit的接收。而后续到Stop位的8个bit接收都是使用pre-sync后的波特率进行的。

注意：

- *在SYNCEN=1的情况下，如果break段后接收到sync段非0x55，则产生sync错误中断标志。*
- *波特率自适应仅在接收到Break之后执行*

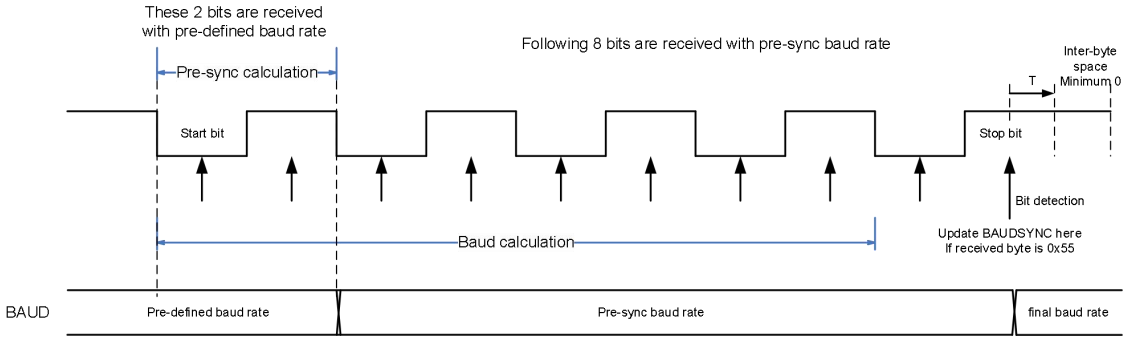


图 20-24LIN 波特率自适应过程

最终的波特率寄存器在stop位检测后更新，更新结果为第一个和第五个下降沿之间的时间间隔除以8得到。

下图举例说明了初始波特率误差过大时，如果没有pre-sync过程，0x55接收将出错（LIN不允许字节内re-sync）

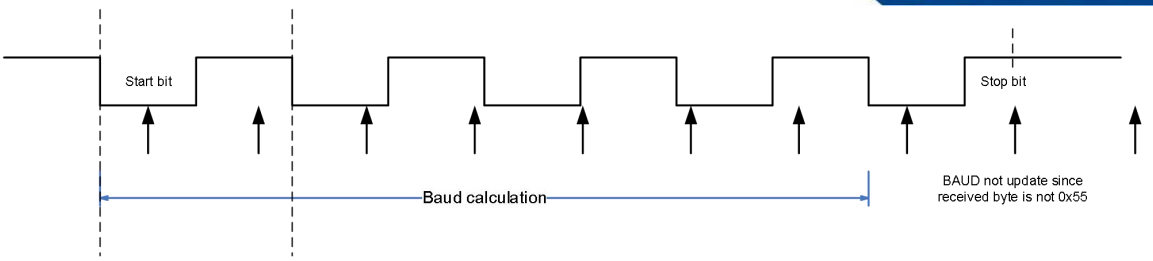


图 20-25 波特率误差较大，无 pre-sync

特殊情况：

主机在Break后发送了0xFF或者0x00，此时从机无法检测到第二个下降沿，这个时候从机仍是按照软件设置的预定义波特率进行接收，不会发生pre-sync，收到的数据也不是0x55，最终报syncerror。

20.10.3.4 从机接收

LIN从机置位RXEN后，状态机开始接收数据，首先做break检测。由于LIN模块并不知道总线数据帧有多少字节，这个信息是总线协议层面事先约定的，所以LIN模块完成break检测和sync检测后会一直保持数据接收状态。软件在收到协议层规定的数据字节和checksum之后，确认当前帧已经接收完毕，此时需要清零RXEN使状态机回到初始态，然后重新置位RXEN使状态机重新进入break检测状态。而wakeup检测是一个独立电路进行的，与LIN模块接收状态机无关。

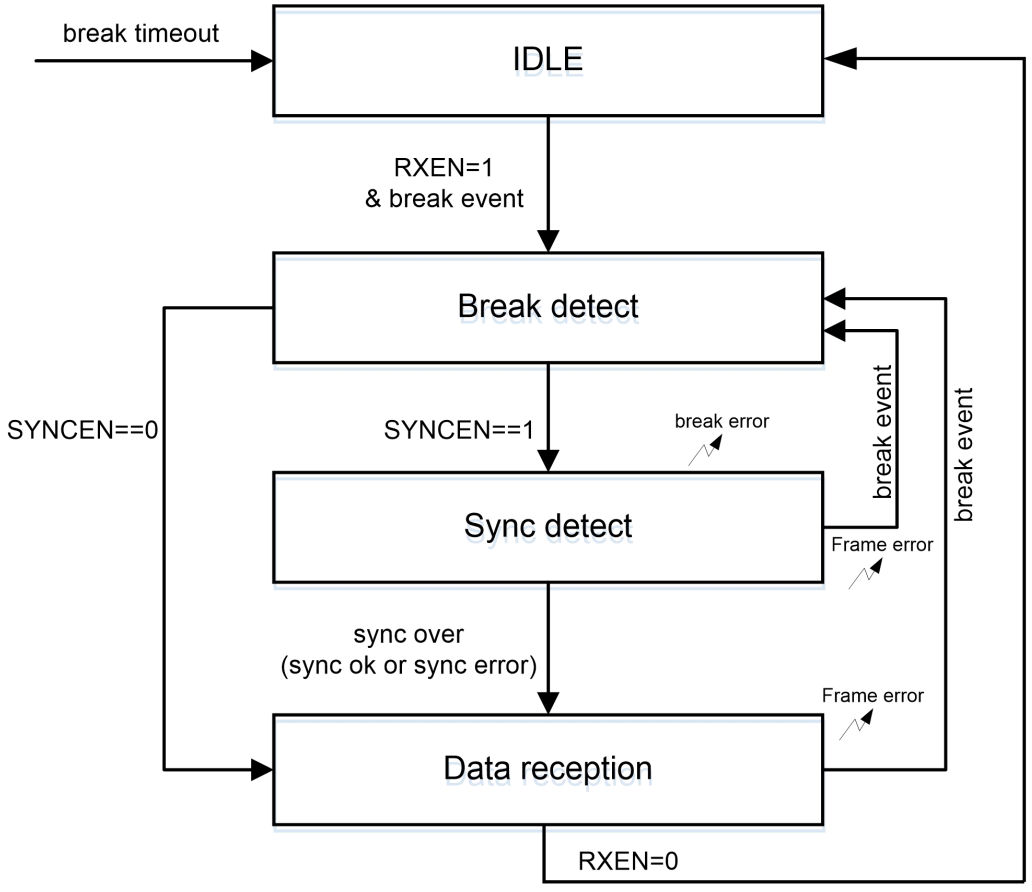


图 20-26LIN 从机接收流程

20.10.3.5 应答发送

LIN从机的数据发送与UART类似，区别是根据IBS寄存器的设置，硬件可能会在发送字节之间插入inter-byte space。根据RSS寄存器的设置，硬件也会在发送完帧头后，插入response space，然后再进行数据发送。

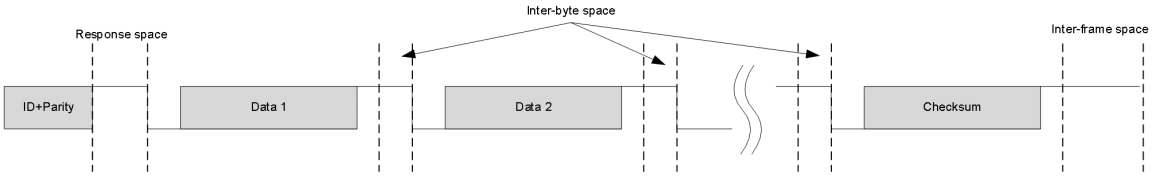


图 20-27 字节间隔和应答间隔

校验和由软件计算，校验和的发送与数据字节相同。

20.10.3.6 从节点休眠

根据LIN2.1协议，当主机保持总线为隐性电平或者显性电平4~10s，无电平转换时，从节点应进入休眠状态。

从节点休眠有两种进入方式：

1）LIN总线保持长时间隐性电平：由于长时间没有检测到Break，MCU主动进入休眠，进入休眠前需要置位WKDETEN寄存器，使能LINwakeup检测功能。

2）LIN总线保持长时间显性电平：从机会检测到Break超时错误，MCU主动进入休眠，进入休眠前需要置位WKDETEN寄存器，使能LINwakeup检测功能。

注意，为了避免休眠中产生错误的break检测，软件休眠前应清零RXEN，关闭数据接收功能。

20.10.4 LIN 接收采样与噪声检测

LIN模式下，数据接收时固定采用16倍过采样模式，OVSM寄存器无效。同时LIN模式支持总线噪声检测，当3个连续的数据采样点得到的采样数据不一致时，NF标志置位，表示检测到总线噪声。即使NF标志置位，不影响3中取2的有效数据判决逻辑。

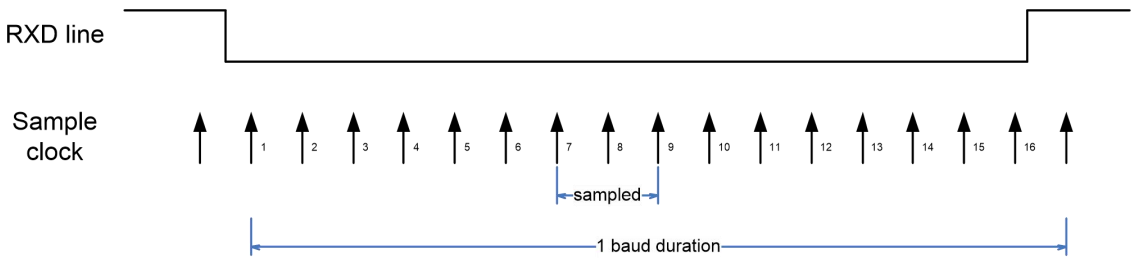


图 20-28LIN bit sampling

数据判决和噪声检测标志：

采样点 7、8、9	数据位判决	噪声标志 NF
000	0	N
001	0	Y
010	0	Y
011	1	Y
100	0	Y
101	1	Y
110	1	Y
111	1	N

注意，LIN2.1 协议中规定 LIN 通信不支持 re-synchronization，因此数据同步仅发生在每个字节的起始位，此时必须设置 RESYNCEN=0.

20.10.5 LIN 错误状态

20.10.5.1 发送数据冲突检测（Bit Error）

配置 BERM 寄存器可以使能发送数据冲突检测功能（仅 LIN 模式下有效）。冲突检测原理是在发送数据时，同时采样 RXD 接收数据，并进行比对，如果收发数据不一致，则置位 bit 冲突中断 BERIF。

数据冲突检测针对主机/从机发送的 break 段、sync 段、数据段和校验和段。

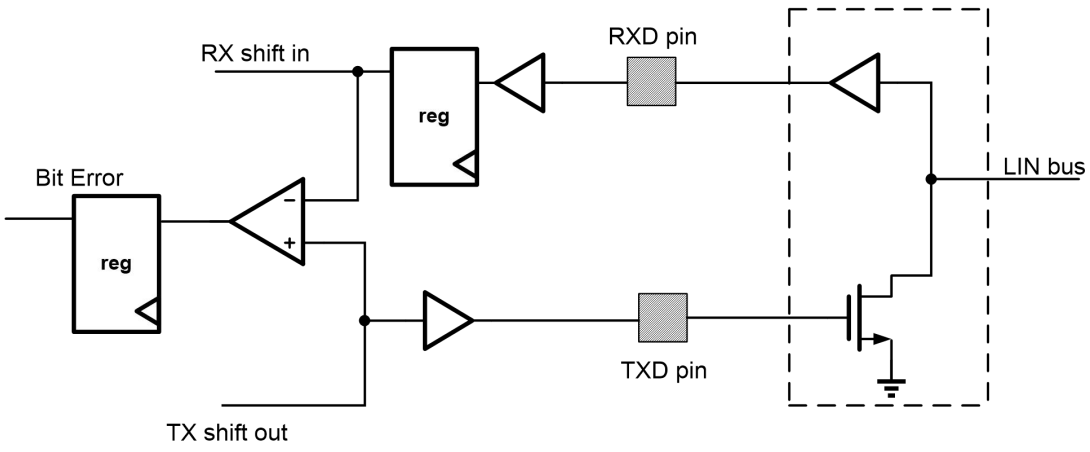


图 20-29LIN bit error 冲突检测

数据冲突检测可以发生在 bit 发送的第 9 个时钟或第 13 个时钟：

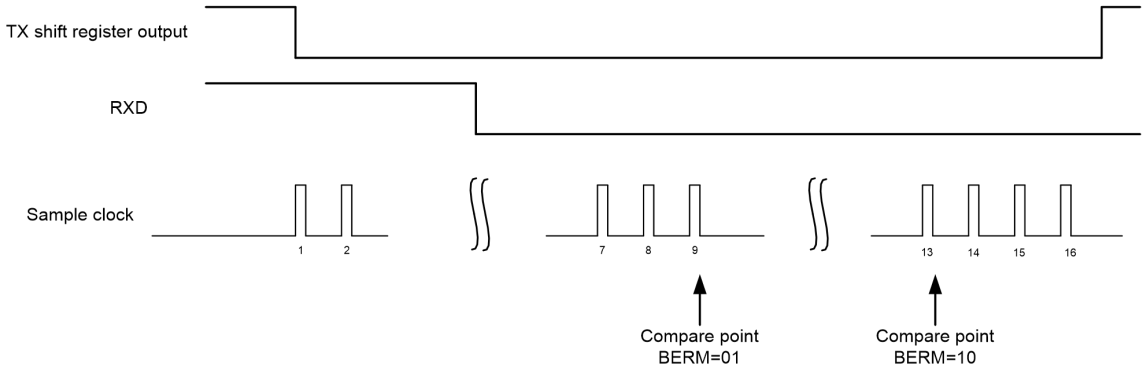


图 20-30LIN bit error 监测点

注意在 sync 段之前由于存在可能的波特率误差，BER 检测时间点可能存在偏差。WKUP 信号不做位冲突检测。

当 BERIF 置位时，LIN 模块采取以下动作：

- 当前发送仍会完成
- 软件重新发送前应先清零 BERIF

注意：由于 WKUP 发送完全独立于 UART 状态机，因此 WKUP 信号不参与发送冲突检测。为了避免 BERIF 误置位，建议在发送唤醒信号时不要使能 BERM 寄存器。

20.10.5.2 timeout 错误

timeout 错误检测 break 连续低电平长度（总线下降沿到上升沿的间隔）是否超出了 FTO 寄存器定义的总 baud 数。FTOEN 寄存器设置是否使能超时功能。

当总线检测到下降沿，LIN 内部超时计数器自动启动计数，检测到上升沿后定时器自动停止并清零。此功能用于检测总线异常的低电平锁死。

根据协议，frame 超时时间通常由下式定义：

frame timeout time = 48 + (number of data bytes + 1)*14 [baud]

这个帧超时检查需要由软件来完成。

20.10.5.3 帧格式错误

数据接收过程中，检测到 stop bit 为低电平。此时停止后续帧数据接收，并置位 FER 标志寄存器。

如果主机在数据帧的中间发送 Break，从机将检测为帧格式错误。

20.10.5.4 break 错误

检测到 break 段错误时 LINBER 标志会置位。LINBER 置位条件是：

[（SYNC 段前两个下降沿长度）/2*11] > [break 长度]

20.10.5.5 sync 错误

在 SYNCEN=1 的情况下，LIN 从机预期会在 break 段后收到 sync 段。如果 break 段之后收到的字节不是 0x55，则从机认为是 sync 段错误，置位 SER 标志，并且可以产生中断。

SYNC 错误有两种可能性：

1） 主机发送的 SYNC 段波特率偏差过大

2） 主机没有发送正确的 SYNC 段

针对错误 1），LIN 从机在接收到 SYNC 段的头两个下降沿时，会自动更新接收使用的波特率，并以新波特率接收后续数据，以校验是否是 0x55。但是即使 0x55 数据正确，波特率过低使得之前收到的 Break 长度不足 11 bit，也是违反协议的（波特率高不违反协议，因为协议只规定了 Break 长度最低值）。因此 LIN 从机需要记录收到的 Break 长度（单位 baud），并且在收到 SYNC 头两个下降沿后判决 Break 长度是否大于 11 个 baud，如不满足则置位 Break 错误标志。

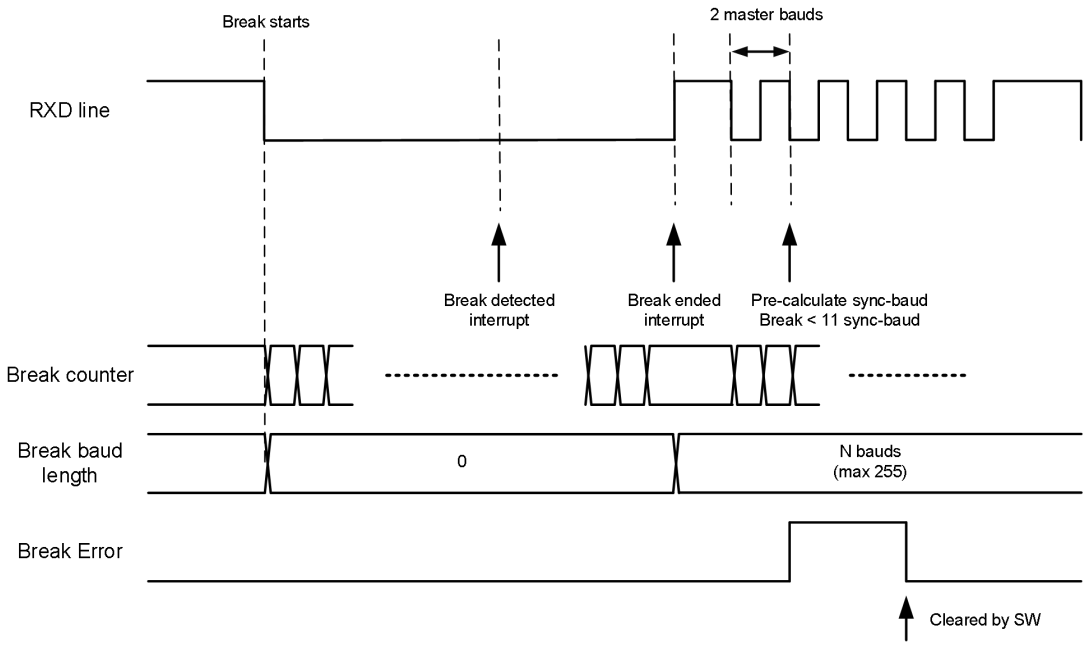


图 20-31Pre-sync 后发现 Break 长度不足，BreakError

针对错误 2），LIN 从机将识别出非 0x55 数据，此时从机置位 SYNC 错误标志。

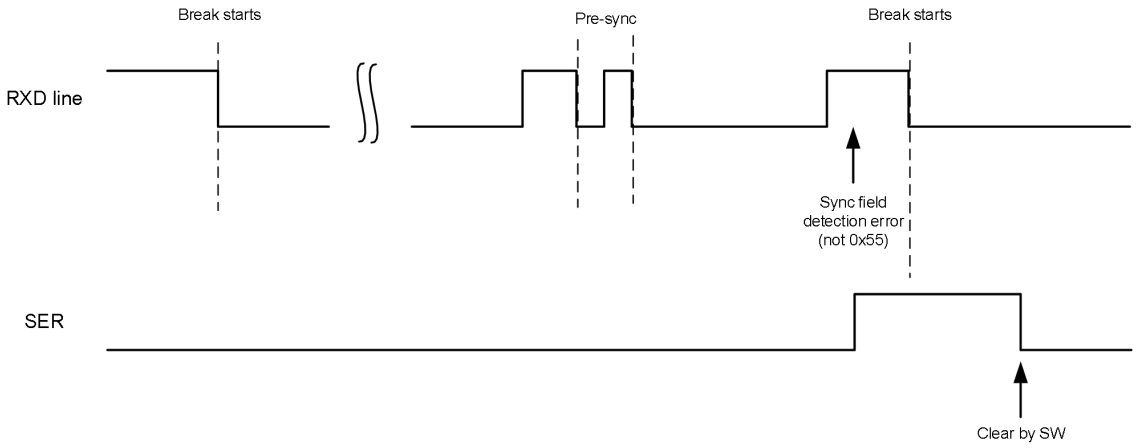


图 20-32SYNC 段数据错误

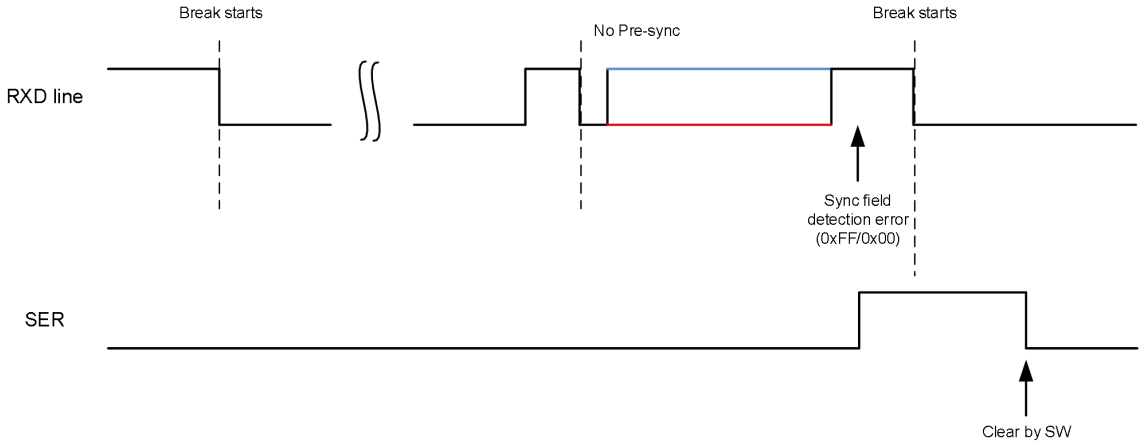
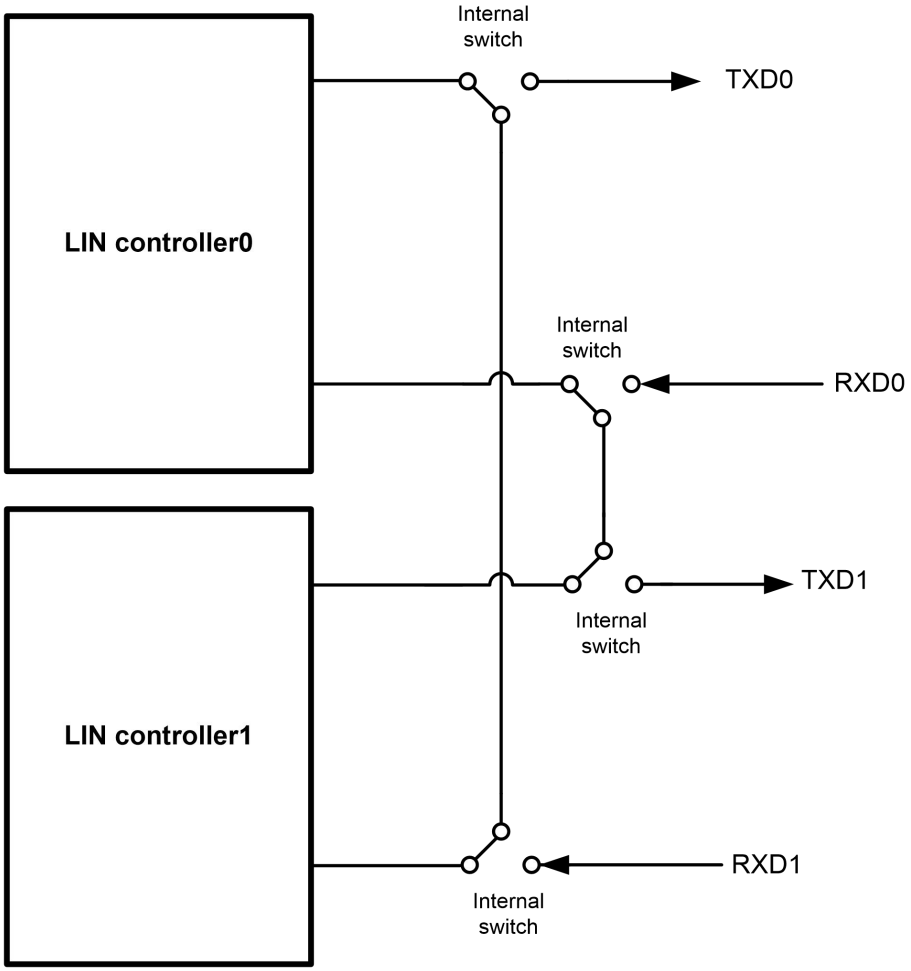


图 20-33SYNC 段 0xFF 或 0x00，无 pre-sync

20.10.6 LIN 自测试模式

LIN 自测试模式需要使用 2 个 LIN 模块实现，LIN 的数据收发与 IO 断开，在 LIN 模块之间实现内部互联：



自测试模式下，通常将 UART0 设置为 LIN 主机，其他设置为 LIN 从机；软件操作主机发送 Break、Sync、数据段、校验和，LIN 从机则自动接收 Break、Sync（自动更新 BAUDSYNC）、数据段和校验和。收发数据正确闭环说明模块功能正常。

模块间自测试互联的配置寄存器仅位于 UART0 模块。

自测试互联设置：

	LINSTC=00	LINSTC=01	LINSTC=10	LINSTC=11
UART0_RXD	self-test disabled	UART1_TXD	UART2_TXD	UART3_TXD
UART0_TXD				
UART1_RXD		UART0_TXD		
UART1_TXD				
UART2_RXD			UART0_TXD	
UART2_TXD				

UART3_RXD				UART0_TXD
UART3_TXD				

LIN 自测试之前，需要配置 LINSTC 寄存器完成内部开关互联，并将参与自测试的模块分别配置成主机和从机。

自测试总是由 UART0 参与和其他模块之间通信来完成。比如配置 LINSTC=01，使用 UART0 和 UART1 进行自测试，流程如下：

- 设置 LINSTC=01
- 设置 UART0 为 LIN 主机，UART1 为 LIN 从机
- UART0 发送 break、sync、data，UART1 完成接收，软件判断收发正常
- 设置 UART0 为 LIN 从机，UART1 为 LIN 主机
- UART1 发送 break、sync、data，UART0 完成接收，软件判断收发正常
- 自测试结束

注意：自测试模式下作为主机的 LIN 模块发送的数据仍会反映在 TX 端口上，为避免自测试过程干扰总线，软件可以将 TX 对应 IO 的外设功能禁用。

20.11 寄存器

UART0模块基地址：0x4001 6800

UART1模块基地址：0x4001 7000

UART2模块基地址：0x4001 7400

UART3模块基地址：0x4001 7800

UARTIR模块基地址：0x4001 7C00

offset 地址	名称	符号
UARTIR 寄存器		
0x00	红外调制寄存器 （Infrared modulation Control Register）	UART_IRCR
UARTx 寄存器		
0x00	UART 控制状态寄存器 （UART Control Status Register）	UART_CSR
0x04	UART 中断使能寄存器 （UART Interrupt Enable Register）	UART_IER
0x08	UART 中断标志寄存器 （UART Interrupt Status Register）	UART_ISR
0x0C	UART 超时和延迟寄存器 （UART Time-Out and Delay Register）	UART_TODR
0x10	UART 接收缓冲寄存器 （UART Receive Buffer）	UART_RXBUF
0x14	UART 发送缓冲寄存器 （UART Transmit Buffer）	UART_TXBUF
0x18	UATR 波特率产生寄存器 （UART Baud rate Generator Register）	UART_BGR
0x20	UART 模式寄存器 （UART Mode Control Register）	UART_MCR
0x24	UART LIN 控制寄存器 （UART LIN Control Register）	UART_LINCR
0x28	UART LIN 波特率同步寄存器 （UART LIN Baud Sync Register）	UART_LINBSR
0x2C	UART LIN 帧超时寄存器 （UART LIN Frame Timeout Register）	UART_LINFTR
0x30	UART LIN 发送时序寄存器 （UART LIN Transmit Timing Register）	UART_LINTTR
0x34	UART LIN Pre-sync 波特率寄存器 （UARTLINPre-sync baud register）	UART_LINPSR
0x38	UART LIN Break 长度寄存器 （UART LIN Break Length Register）	UART_LINBKR
0x80	LIN 自测试开关控制寄存器 （LIN Self-test Cwitch Control Register）	UART_LINSCCR

20.11.1 红外调制寄存器（UART_IRCR）

名称	UART_IRCR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IRFLAG	TH				TZBRG[10:8]		
位权限	R/W-0	R/W-0000				R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TZBRG[7:0]							
位权限	R/W-11010010							

位号	助记符	功能描述
31:16	-	**未实现：读为0**
15	**IRFLAG**	控制红外调制发送数据时的默认输出极性（Infra Red ） 0：正极性 1：负极性
14:11	**TH**	红外占空比调制参数（Tranmission High Duty)
10:0	**TZBRG**	红外调制频率（Transmission Baud Rate)

20.11.2 UARTx 模式寄存器（UARTx_MCR）

名称	**UARTx_MCR**							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					HDSEL	LINMD	MODE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:3	-	未实现：读为0

位号	助记符	功能描述
2	**HDSEL**	半双工单线模式选择 0：双线 UART 1：单线半双工 UART 注：此寄存器仅在 MODE=0 时有效
1	**LINMD**	LIN 模式选择寄存器 0：LIN 主机 1：LIN 从机
0	**MODE**	模块控制寄存器 0：标准 UART 模式 1：LIN 模式

20.11.3 UARTx LIN 控制寄存器（UARTx_LINCR）

名称	UARTx_LINCR							
offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	WKUP_LEN							
位权限	R/W-0000 1000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SYNCEN	FTOEN	BERM		NDET_EN	WKDET_EN	TX_BF	TX_WKUP
位权限	R/W-1	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:16	**-**	未实现：读为0
15:8	**WKUP_LEN**	发送唤醒信号时，此寄存器定义 LIN 总线唤醒信号长度 接收唤醒信号时，此寄存器定义 LIN 唤醒信号识别阈值 单位是 LINCLK，即 RCLP；复位值为 8，表示 9 个时钟周期，大约 270us；最大 256 个时钟周期，大约 7.6ms
7	**SYNCEN**	LIN从机Sync段接收使能 0：不识别sync段，不进行波特率计算 1：识别sync段，进行波特率计算
6	**FTOEN**	LIN Frame timeout使能 0：禁止帧超时检测 1：使能帧超时检测，超时阈值由LINFTO寄存器定义
5:4	**BERM**	LIN总线发送冲突检测配置（不针对WKUP信号） 00/11：关闭发送冲突检测 01：使能发送冲突检测，在发送第9bit处检测 10：使能发送冲突检测，在发送第13bit处检测

位号	助记符	功能描述
3	**NDET_EN**	总线噪声检测使能 (Noise detect enable) 0：关闭LIN总线噪声检测 1：使能LIN总线噪声检测，检测到噪声时NF标志置位
2	**WKDET_EN**	LIN唤醒信号检测使能 0：禁止检测唤醒信号 1：使能检测唤醒信号
1	**TX_BF**	发送 LIN 总线 Break 信号，软件置位，发送完成后硬件自动清零 仅在 LIN 主机模式下有效
0	**TX_WKUP**	发送 LIN 总线唤醒信号，软件置位，发送完成后硬件自动清零 LIN 主机和从机模式下都可以发送唤醒 *注意：TX_BF 和 TX_WKUP 不能同时置位；如果软件同时置位 TX_BF 和 TX_WKUP，发送波形由低电平时间长的一方决定*

20.11.4 UARTx LIN 波特率同步寄存器（UARTx_LINBSR）

名称	UARTx_LINBSR							
offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				BAUD_SYNC			
位权限	U-0				R-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BAUD_SYNC							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BAUD_SYNC							
位权限	R-0000 0000							

位号	助记符	功能描述
31:20	-	未实现：读为0
19:0	**BAUD_SYNC**	同步段计数器，计算同步段的时钟个数，用于波特率同步 软件在检测到 SYNCF 标志后读取此寄存器，除以 8 后得到当前 LIN 波特率

20.11.5 UARTx LIN 帧超时寄存器（UARTx_LINFTR）

名称	UARTx_LINFTR							
offset	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				-			
位权限	U-0				U-0			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	-				FTO			
位权限	U-0				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FTO							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:12	-	未实现：读为0
11:0	**FTO**	Frame timeout 设置 定义低电平最大时间长度

20.11.6 UARTx LIN 发送时序寄存器（UARTx_LINTTR）

名称	UARTx_LINTTR							
offset	0x30							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		IBS		-	RSS		
位权限	U-0		R/W-00		U-0	R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		BDT		BLT			
位权限	U-0		R/W-00		R/W-0000			

位号	助记符	功能描述
31:14	-	未实现：读为0
13:12	**IBS**	inter-byte space，定义数据发送时的数据字节间隔，仅发送时有效 00: 0 baud 01: 1 baud 10: 2 baud 11: 3 baud
11	-	未实现：读为0
10:8	**RSS**	response space，定义 header 到发送数据 response 之间的间隔，仅发送时有效 000: 0 baud 001: 1 baud … 111: 7 baud
7:6	-	未实现：读为0
5:4	**BDT**	Break delimiter 长度 00: 1 baud 01: 2 baud 10: 3 baud 11: 4 baud

位号	助记符	功能描述
3:0	**BLT**	Break 长度 0000: 13 baud 0001: 14 baud ... 1110: 27 baud 1111: 28 baud

20.11.7 UARTx LINPre-sync 寄存器（UARTx_LINPSR）

名称	UARTx_LINPSR							
offset	0x34							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				PRE_SYNC			
位权限	U-0				R-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PRE_SYNC							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PRE_SYNC							
位权限	R-0000 0000							

位号	助记符	功能描述
31:20	**-**	未实现：读为0
19:0	**PRE_SYNC**	同步段预同步（前 2bit）计数结果，此结果用于设定同步段后续 8bit 接收波特率；软件可读取此寄存器用于查询和诊断。

20.11.8 UARTx LIN Break 长度寄存器（UARTx_LINBKR）

名称	UARTx_LINBKR							
offset	0x38							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				-			
位权限	U-0				U-0			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				BREAK_LEN			
位权限	U-0				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BREAK_LEN							
位权限	R-0000 0000							

位号	助记符	功能描述

位号	助记符	功能描述
31:12	-	未实现：读为0
11:0	**BREAK_LEN**	Break 段收到的波特个数，用于 Break 接收结束后的长度记录，供软件查询。

20.11.9 UARTx 控制状态寄存器（UARTx_CSR）

名称	UARTx_CSR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							BUSY
位权限	U-0							R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RTSEN	CTSEN	-				TXIREN	RXTOEN
位权限	R/W-0	R/W-0	U-0				R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RESYNCEN	-	OVSM	IOSWAP	NEWUP	DMATXIFCFG	BITORD	STOPCFG
位权限	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PDSEL		PARITY		RXPOL	TXPOL	RXEN	TXEN
位权限	R/W-01		R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:25	-	未实现：读为0
24	**BUSY**	UART 通信标志，只读 (Busy) 1：UART 正在通信中 0：UART 空闲
23	**RTSEN**	RS232 RTS 流控使能，仅 UART 模式有效 0：关闭 RTS 流控 1：使能 RTS 流控
22	**CTSEN**	RS232 CTS 流控使能，仅 UART 模式有效 0：关闭 CTS 流控 1：使能 CTS 流控
21:18	-	未实现：读为0
17	**TXIREN**	发送红外调制使能 (Transmit Infra-red modulation Enable) 1：使能红外调制发送 0：关闭红外调制发送
16	**RXTOEN**	接收超时使能 (Receive Time-Out Enable) 1：使能接收超时功能 0：关闭接收超时功能
15	**RESYNCEN**	字节内重同步使能 (Re-synchronization enable) 0：禁止重同步，仅作起始位同步 1：允许重同步 注意，LIN通信时保持此bit为0
14	-	未实现：读为0

位号	助记符	功能描述
13	**OVSM**	接收过采样模式选择 (Oversampling mode) 0：16 倍过采样 1：8 倍过采样 *注意：在 LIN 模式下，固定为 16 倍过采样，此寄存器无效*
12	**IOSWAP**	RX 和 TX 引脚交换 0：默认引脚顺序（与封装图一致） 1：交换引脚顺序
11	**NEWUP**	UART RX下降沿唤醒功能使能寄存器（仅UART0和UART1有效）(Negedge Wakeupenable) 1：使能RX下降沿唤醒 0：禁止RX下降沿唤醒
10	**DMATXIFCFG**	DMA发送完成中断使能，仅在UART通过DMA进行发送时有效 (DMA transmit interrupt enable) 1：IE=1的情况下，DMA模式下发送完最后一帧后，允许中断信号输出；最后一帧之前的数据帧发送完成后不允许中断信号输出 0：是否允许中断信号输出仅由IE决定
9	**BITORD**	数据发送/接收时的位顺序 (Bit Order) 0：LSB first 1：MSB first
8	**STOPCFG**	停止位宽度配置，仅对发送帧格式有效，接收时不判断停止位个数 (Stop bit config) 0：1位停止位 1：2 位停止位
7:6	**PDSEL**	每帧的数据长度选择；此寄存器对数据发送和接收同时有效 (Payload data length Select) 00：7 位数据 01：8 位数据 10：9 位数据 11：6 位数据
5:4	**PARITY**	校验位配置；此寄存器对数据发送和接收同时有效 00：无校验位 01：偶校验 10：奇校验 11：RFU
3	**RXPOL**	接收数据极性配置 (Receive Polarity) 0：正向 1：取反
2	**TXPOL**	发送数据极性配置 (Transmit Polarity) 0：正向 1：取反
1	**RXEN**	接收使能，1 有效 (Receive Enable)
0	**TXEN**	发送使能，1 有效 (Transmit Enable)

20.11.10 UARTx 中断使能寄存器（UARTx_IER）

名称	UARTx_IER(x=0~7)							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	TOE_IE	SER_IE	FER_IE	BER_IE	NF_IE	LINSYNC_IE	LINB_IE	LINWK_IE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LINWKTF_IE	LINBEF_IE	LINBER_IE	-		PERR_IE	FERR_IE	OERR_IE
位权限	R/W-0	R/W-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CTSIE	-			RXTO_IE	RXERR_IE	-	RXBF_IE
位权限	R/W-0	U-0			R/W-0	R/W-0	U-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	NEWUP_IE	-					TXBE_IE	TXSE_IE
位权限	R/W-0	U-0					R/W-0	R/W-0

位号	助记符	功能描述
31	**TOE_IE**	LIN timeout error中断使能，1有效
30	**SER_IE**	LIN sync error 中断使能，1 有效
29	**FER_IE**	LIN framing error 中断使能，1 有效???
28	**BER_IE**	LIN 发送冲突中断使能，1 有效
27	**NF_IE**	LIN 噪声检测中断使能，1 有效
26	**LINSYNC_IE**	LIN 同步段中断使能，1 有效
25	**LINB_IE**	LIN Break 段中断使能，1 有效
24	**LINWK_IE**	LIN 唤醒中断使能，1 有效
23	**LINWKTF_IE**	LIN 唤醒完成中断使能，1 有效
22	**LINBEF_IE**	LIN 从机模式，Break 结束中断使能，1 有效
21	**LINBER_IE**	LIN 从机模式，Break Error 中断使能，1 有效
20:19	**-**	未实现：读为 0
18	**PERR_IE**	奇偶校验错误中断使能，1 有效
17	**FERR_IE**	帧格式错误中断使能，1 有效
16	**OERR_IE**	接收缓存溢出错误中断使能，1 有效
15	**CTSIE**	nCTS翻转中断使能，1有效
14:12	**-**	**未实现：读为0**
11	**RXTO_IE**	接收超时中断使能，1 有效 (Receive Time-Out Interrupt Enable) （仅 UART0 和 UART1 有效）
10	**RXERR_IE**	接收错误中断使能，1 有效(Receive Error Interrupt Enable)
9	**-**	**未实现：读为0**
8	**RXBF_IE**	接收缓存满中断使能，1 有效 (Receive Buffer Full Interrupt Enable)
7	**NEWUP_IE**	RX 下降沿异步检测中断使能，1 有效 (Negedge Wakeup Interrupt Enable)
6:2	**-**	**未实现：读为 0**
1	**TXBE_IE**	发送缓存空中断使能，1 有效 (Transmit Buffer Empty Interrupt Enable)
0	**TXSE_IE**	发送缓存空且发送移位寄存器空中断使能，1 有效 (Transmit Shift register Empty Interrupt Enable)

20.11.11 UARTx 中断标志寄存器（UARTx_ISR）

名称	UARTx_ISR(x=0~7)							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TOER	SER	FER	BERIF	NF	LINSYNCF	LINBF	LINWKF
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LINWKTF	LINBEF	LINBER	-		PERR	FERR	OERR
位权限	R/W-0	R/W-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CTSIF	-			RXTO	-		RXBF
位权限	R/W-0	U-0			R/W-0	U-0		R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	NEWKF	-				TX_OERR	TXBE	TXSE
位权限	R/W-0	U-0				R/W-0	R -1	R/W-0

位号	助记符	功能描述
31	**TOER**	LIN timeout error 标志，硬件置位，软件写1清零
30	**SER**	LIN sync error 标志，硬件置位，软件写 1 清零
29	**FER**	LIN framing error 标志，硬件置位，软件写 1 清零
28	**BERIF**	LIN 发送冲突标志（bit error），硬件置位，软件写 1 清零
27	**NF**	LIN 总线噪声检测标志，硬件置位，软件写 1 清零
26	**LINSYNCF**	LIN 同步段接收完成中断标志，硬件置位，软件写 1 清零
25	**LINBF**	LIN 从机模式：LIN Break 段接收完成中断标志，硬件置位，软件写 1 清零 LIN 主机模式：LIN Break 段发送完成中断标志，硬件置位，软件写 1 清零
24	**LINWKF**	LIN 唤醒信号检测标志，识别到有效唤醒信号时硬件置位，软件写 1 清零
23	**LINWKTF**	LIN 唤醒信号发送完成中断标志，硬件置位，软件写 1 清零
22	**LINBEF**	LIN 从机模式，LIN Break 段结束中断标志，硬件置位，软件写 1 清零
21	**LINBER**	LIN 从机模式，LIN Break error 标志，硬件置位，软件写 1 清零
20:19	**-**	未实现：读为0
18	**PERR**	奇偶校验错误中断标志，硬件置位，软件写 1 清零 (Parity Error,write 1 to clear)
17	**FERR**	帧格式错误中断标志，硬件置位，软件写 1 清零 (Frame Error flag,write 1 to clear)
16	**OERR**	接收缓存溢出错误中断标志，当接收缓存满的情况下，收到新的数据时置位；硬件置位，软件写 1 或者读取 RXBUF 时清零 接收溢出时，接收缓冲器中原有的数据被新数据覆盖。 (RX buffer Overflow Error flag,write 1 to clear)
15	**CTSIF**	CTS 信号翻转标志，当 nCTS 输入发生翻转时置位，软件写 1 清零
14:12	**-**	未实现：读为0

位号	助记符	功能描述
11	**RXTO**	接收超时中断标志，硬件置位，软件写 1 清零 (Receive Time-Out flag,write 1 to clear) （仅 UART0 和 UART1 有效）
10:9	**-**	未实现：读为0
8	**RXBF**	接收缓存满中断标志，硬件置位，软件写 1 或者读取 RXBUF 时清零 (Receive Buffer Full flag write 1 to clear)
7	**NEWKF**	RX 下降沿异步检测中断标志，硬件置位，软件写 1 清零 (Negedge Wakeup Flag write 1 to clear) （仅 UART0 和 UART1 有效）
6:3	**-**	未实现：读为 0
2	**TX_OERR**	发送缓存溢出中断标志，硬件置位，软件写 1 清零 (Transmit Buffer Overflowflag)
1	**TXBE**	发送缓存空中断标志，硬件置位，软件写 1 或软件写发送缓存时清零 (Transmit Buffer Empty flag,write 1 to clear)
0	**TXSE**	发送缓冲区空且发送移位寄存器空标志，硬件置位，待发送数据从 TXBUF 写入发送移位寄存器时清零 (Transmit Shift register Empty flag)

20.11.12 UARTx 超时和延迟寄存器（UARTx_TODR）

名称	UARTx_TODR(x=0~5)							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TXDLY_LEN							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXTO_LEN							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:16	**-**	**未实现：读为0**
15:8	**TXDLY_LEN**	发送延迟，最大 255baud (Transmit Delay Length)
7:0	**RXTO_LEN**	接收超时溢出长度，最大 255baud (Receive Time-Out Length)

20.11.13 UARTx 接收缓冲寄存器（UARTx_RXBUF）

名称	UARTx_RXBUF							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							RXBUF[8]
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:9	-	未实现：读为0
8:0	**RXBUF**	接收数据缓冲寄存器数据(Receive buffer)

7位收发时，接收的7bits数据存入RXBUF[6:0]

20.11.14 UARTx 发送缓冲寄存器（UARTx_TXBUF）

名称	UARTx_TXBUF							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							TXBUF[8]
位权限	U-0							W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:9	-	未实现：读为0
8:0	**TXBUF**	发送数据缓冲寄存器数据(Transmit Buffer)

7位收发时，发送的7bits数据写入TXBUF[6:0]

20.11.15 UATRx 波特率产生寄存器（UARTx_BGR）

名称	UARTx_BGR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SPBRG[15:8]							
位权限	R/W-00000011							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SPBRG[7:0]							
位权限	R/W-01000001							

位号	助记符	功能描述
31:16	-	未实现：读为0
15:0	**SPBRG**	波特率产生器寄存器值(Serial Port Baud Rate Generation) 当LIN模式下SYNCEN=1，在PreSync阶段以及成功接收SYNC段后，硬件自动更新此寄存器； 注意，即使SYNCEN=1，软件仍可以改写SPBRG

波特率计算详见波特率发生章节

注：当SPBRG <= 0x000F时，UARTDIV=16'H000F;

当SPBRG >0x000F时，UARTDIV=SPBRG;

21 通用同异步收发传输器（USART）

21.1 概述

USART串行通信模块特点如下

- 波特率软件可配置
- 支持全双工通信口
- 支持半双工单线通信
- 支持RX-TX端口交换
- USART具有数据接收完成/接收错误中断，并提示错误类型
- 可配置数据长度，支持6、7、8、9bits
- 可配置的停止位-支持1个停止位或2个停止位
- 可配置为红外调制输出功能，且载波频率可设置，及载波占空比可设置
- 支持DMA
- 支持接收超时机制
- 支持发送延迟功能

USART支持smart-card模式（ISO7816-3），包括以下特性：

- 具备卡时钟输出端口，输出频率在1MHz~5MHz之间可设
- 位传输方向可配置，支持MSB First或LSB First
- 错误信号宽度可配置为1/1.5/2个ETU
- 发送数据支持传输错误重发机制，重发次数可配置为0~3次
- 支持EGT可设0~256，并支持多种超时中断
- 具有数据接收完成/接收错误中断，并提示错误类型
- 发送中断产生条件可配置为缓冲区空或移位寄存器空
- 支持DMA接口

21.2 UART 类型区分

芯片集成了多种不同类型的UART（LPUART），其差异如下表所示：

UART 特性	UART (UART0/1/2/3)	USART (UART4/5)	LPUART
DMA 支持	Y	Y	Y
半双工/全双工	Y	Y	Y
红外发射	Y	Y	-

双时钟域（工作时钟独立于总线）	Y	-	Y
半双工单线通信	Y	Y	-
休眠唤醒	Y	-	Y
接收超时	Y	Y	-
发送延迟	Y	Y	-
数据长度	6、7、8、9bits		
LIN support	Y	N	
智能卡模式	N	Y	N

表 21-1 UART 类型列表

21.3 结构框图

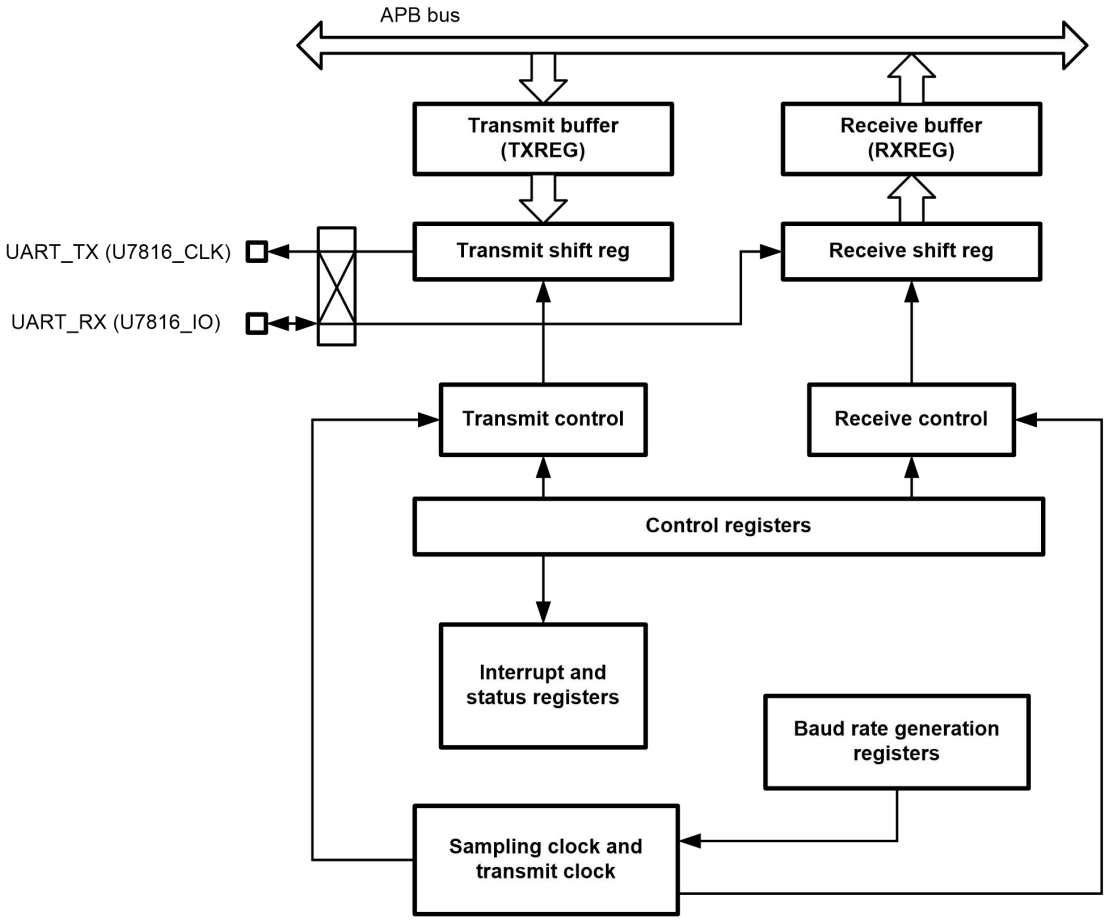


图 21-1 USART 模块框图

21.4 USART 字符描述

21.4.1 UART 模式

UART 传输字符的基本时序如下图所示。每个字符帧包含至少 1bit START 位和至少 1bit STOP 位，数据长度可以配置为 6~9bits，并且可以选择有无校验位。

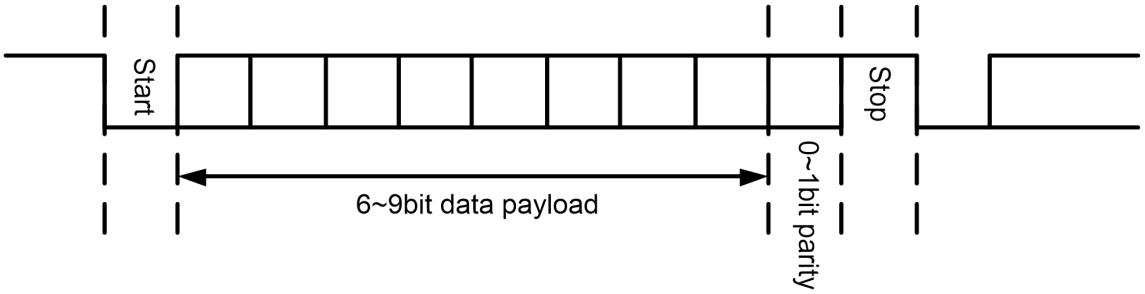


图 21-2 UART 字符描述

UART 支持多种帧格式，由 UARTxCSR.PDSEL 寄存器和 UARTxCSR.PARITY 寄存器控制。见下表：

PDSEL	PARITY	帧格式[1]
00	00	[Start | 7 bits data | Stop]
	01, 10	[Start | 7 bits data | Parity | Stop]
01	00	[Start | 8 bits data | Stop]
	01, 10	[Start | 8 bits data | Parity | Stop]
10	00	[Start | 9 bits data | Stop]
	01, 10	[Start | 9 bits data | Parity | Stop]
11	00	[Start | 6 bits data | Stop]
	01, 10	[Start | 6 bits data | Parity | Stop]

表 21-2 UART 数据帧格式

[1]：Stop 位可能是 1bit 或者 2bit，根据 STOPCFG 寄存器决定

注意 PDSEL 寄存器用于配置帧的数据长度，通信帧长为【起始位+数据位+校验位+停止位】。

21.4.2 智能卡模式

ISO7816-3协议规定的字符格式如下图。

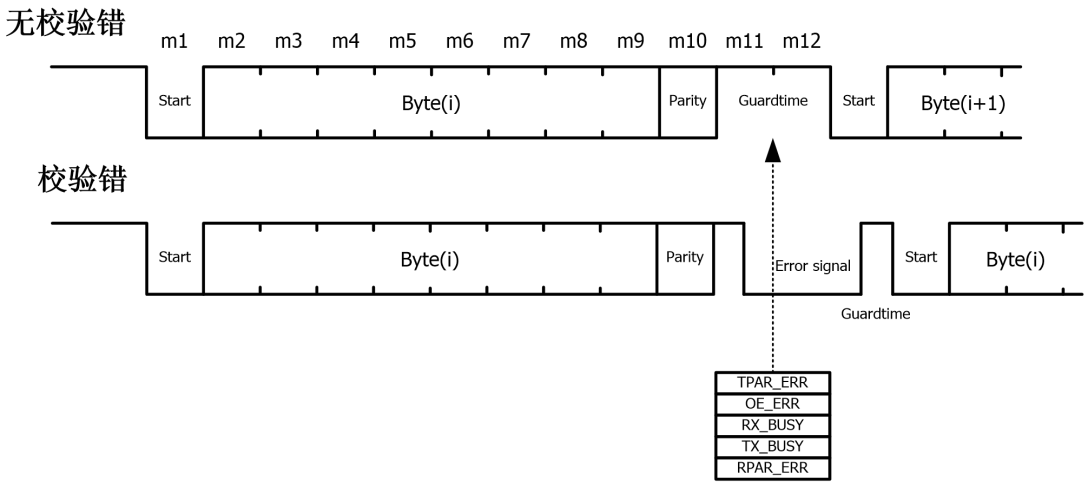


图 21-3 7816-3 帧结构图

参照 7816 协议标准，7816 基本接口时序如下：

- 一个起始位后跟8个数据位及1个校验位，以1-ETU或2-ETU的GUARDTIME结束。
- 单字节数据长度最小为11-ETU或12-ETU。
- 第10.5个ETU接收电路校验接收数据，若校验正确，则插入2个ETU的GUARDTIME，确保数据长度为12-ETU，并在第11个ETU时令RX_BUSY无效并产生可能的OE_ERR标志，完成数据发送；若接收校验出错，则在第10.5-ETU拉低IO，产生ERROR SIGNAL。ERROR SIGNAL最短1个ETU，最长2个ETU。并在第11个ETU根据需要产生RPAR_ERR标志。
- 第11个ETU时发送电路未采样到ERROR SIGNAL，则说明发送数据正确，数据发送完成，令TX_BUSY无效。
- 若第11个ETU发送电路采样到ERROR SIGNAL，则说明发送数据错误，根据设定产生需要的TPAR_ERR或等待2个ETU后重发数据。

UART模块用于智能卡通信时，必须设置数据长度为8bit，并且带校验位。

21.5 功能描述

21.5.1 时钟结构

USART 采用单时钟结构，此时 UCLK=PCLK，USART 的数据收发时钟也是来源于 APBCLK 的。

21.5.2 同步策略

UART 支持两种同步策略：起始位同步和重同步。

当 RESYNCEN=0，UART 只执行起始位同步，时钟频率误差将在字节内累积。

当 RESYNCEN=1，UART 执行起始位同步，并在字节内每个下降沿执行重同步。

通过 start bit 下降沿采样可以实现起始位重同步。使用 UCLK 采样 RXD，当检测到起始位下降沿时，重同步波特率计数器，以实现波特率同步于起始位检测，如下图所示。

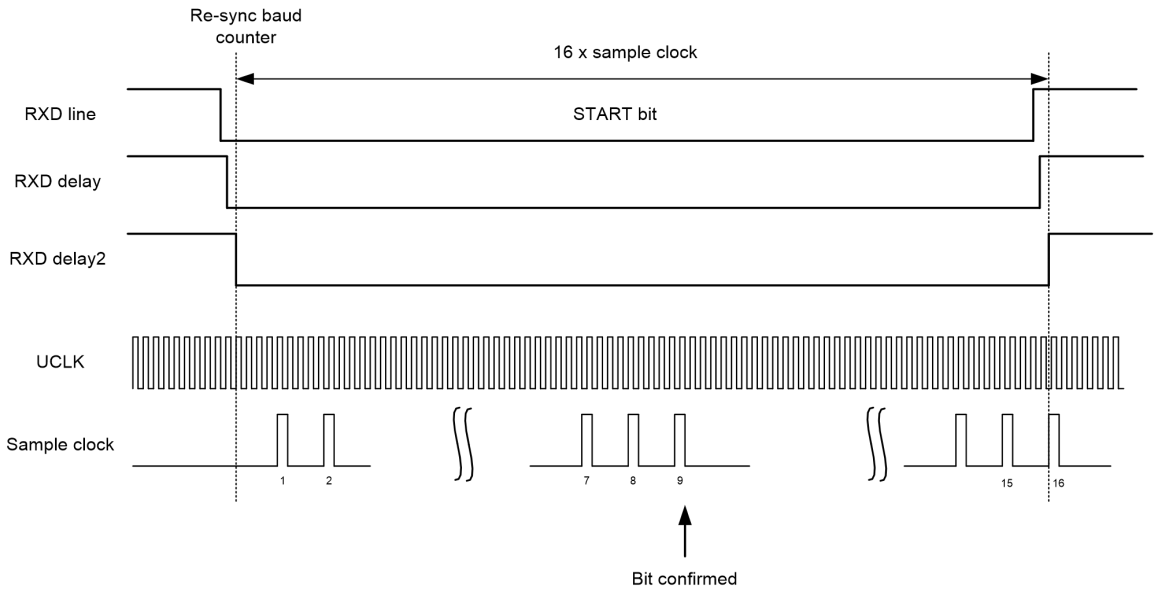


图 21-4 起始位同步

21.5.3 位接收采样

UART 对接收数据进行波特率的 8 倍或 16 倍过采样，并在每个 bit 的中间位置进行三中取二的多数判决，以提高对信号噪声的抑制能力。

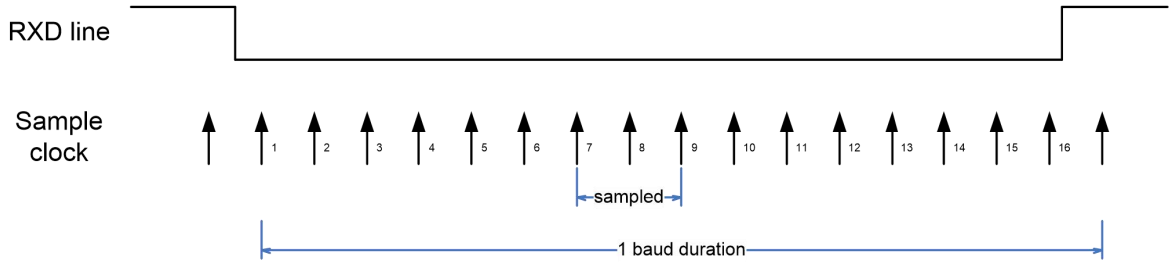


图 21-5 位接收 16 倍采样

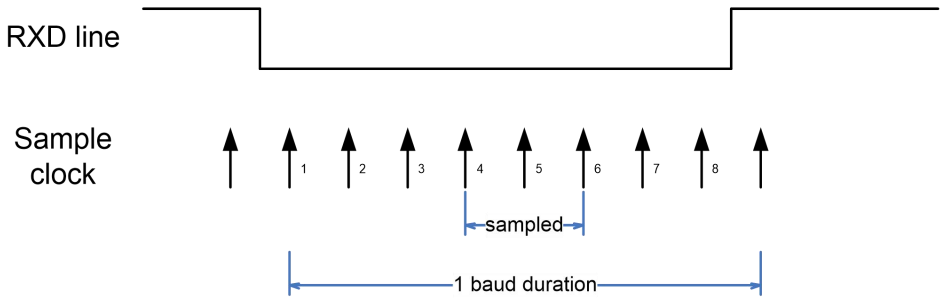


图 21-6 位接收 8 倍采样

接收移位寄存器收到的 bit 位是多数判决的结果。例如三次采样结果是 001，则判决为 0；如果是 011，则判决为 1。

通过 OVSM 寄存器可以配置 UART 接收数据时的过采样倍数。如果 UART 对输入信号进行 16 倍过采样，要求 SPBRG 配置不能小于 16，即 UART 工作时钟必须至少是波特率的 16 倍。如果 UART 对输入信号进行 8 倍过采样，要求 SPBRG 配置不能小于 8，即 UART 工作时钟必须至少是波特率的 8 倍。

选择较小的过采样倍数时，可以获得较高的通信波特率。

21.5.4 数据发送

在发送模式下，UART 的串行数据发送电路主要包括一个发送移位寄存器(TXSR)，TXSR 功能是将数据逐个移位送出。待发数据必须先写到发送缓冲区中。当软件置位 TXEN 寄存器后，如果发送缓冲区非空，UART 将缓冲区数据载入 TXSR 并开始移位输出。

注：由于寄存器操作时钟和波特率时钟是异步关系，当发送开始时，需要等待波特率时钟到来，因此从 TXEN 置位到 UART 开始发送 Start 位之间，有最大 1 个 baud 的延迟。

TXBE 和 TXSE 是发送中断标志位，分别表示发送缓冲区空和 TXSR 空，软件可以选择在合适的时间点产生发送完成中断。

一般情况下，一开始 TXSR 寄存器是空的，数据的发送需先设定波特率 SPBRG，使能发送模块(设定 TXEN 为 1)，然后写入 TXBUF 寄存器开始发送。也可以在设定好波特率 SPBRG 后，先写入 TXBUF 寄存器，然后再设定 TXEN 使能发送模块来开始数据发送。如果在数据发送过程中将发送模块使能位 TXEN 清 0，那么数据发送工作就会被中断，发送模块也会被复位。

下图为UART异步发送的例子。这个示例中软件首先向TXBUF写入数据，然后通过置位TXEN启动发送。

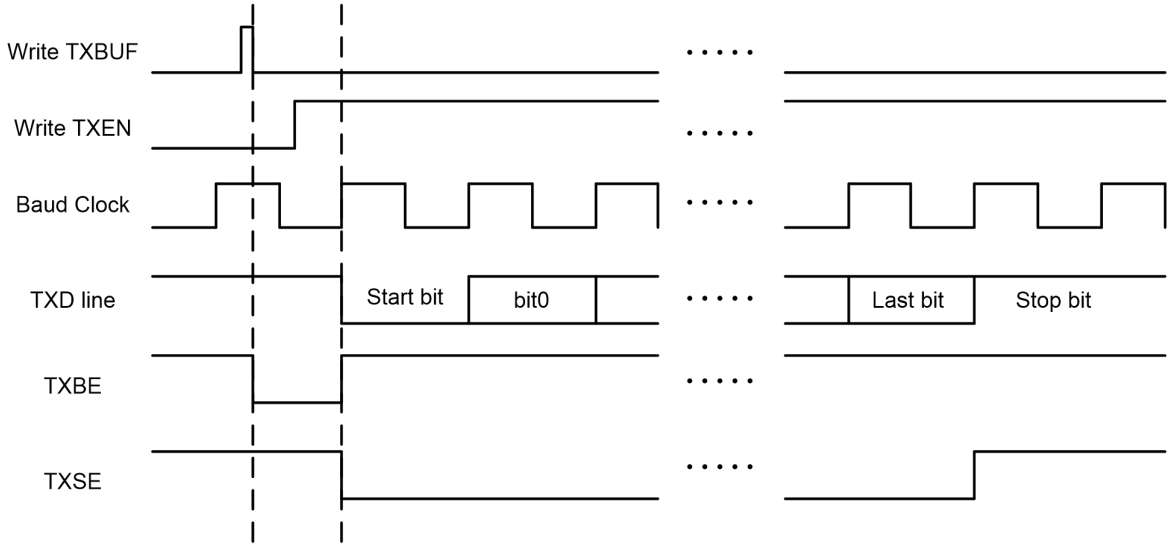


图 21-7 UART 异步发送波形 1

上图中推荐的操作步骤如下：

- 选择合适的波特率，初始化 SPBRG
- 若需要中断，置位 TXSE_IE 或者 TXBE_IE
- 决定数据发送的格式：设置 PDSEL 寄存器，决定发送数据长度；设置 PARITY 寄存器选择是否发送校验位以及校验类型，设置 STOPSEL 寄存器决定发送 1 位还是 2 位停止位
- 如果希望发送的串行数据红外调制，向 IRCON 寄存器写入合适的值来获得相应的调制频率和占空比，并置位 TXIREN
- 将待发送的数据写入 TXBUF 寄存器（自动启动发送）
- 使能发送模块：置位 TXEN

软件也可以先置位TXEN再写入TXBUF，此时UART会在数据写入TXBUF后立刻开始发送流程。

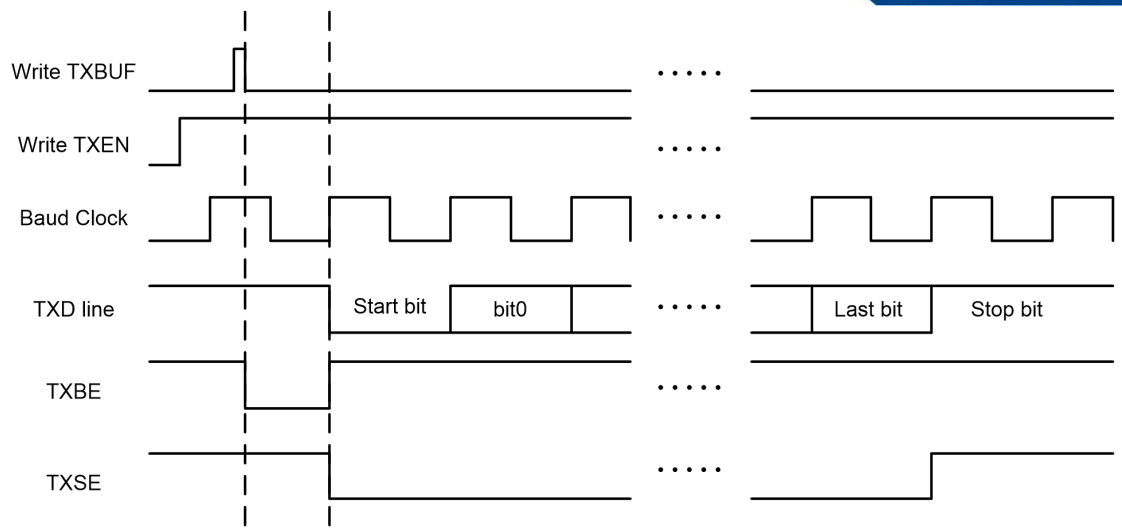


图 21-8 UART 异步发送波形 2

当TXBUF为空时，软件可以立即写入下一个待发送数据，以实现连续无间隔的数据发送。

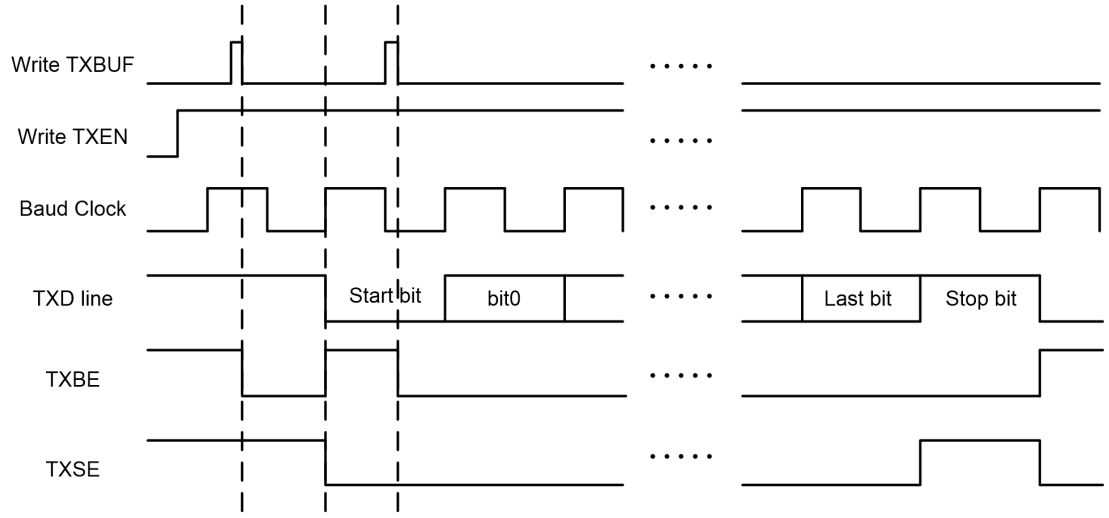


图 21-9 UART 异步发送波形 3

智能卡模式

在智能卡模式下，UART完成数据和校验位发送后，会检测从机是否回发错误信号。如果没有检测到错误信号，则发送停止位结束通信；如果检测到错误信号，则启动自动重发流程。当自动重发超过寄存器设定的次数，但是仍接收到错误信号，则UART放弃重发并置位发送错误中断；如果自动重发后没有检测到错误信号，则发送停止位结束通信。

在发送正常结束后，UART根据寄存器设置等待额外保护时间，并在保护时间结束后才置位发送完成中断。

注意：在错误重发过程中，TXSE不会置位，直到重发结束；此时TXBUF中的数据也不会写入TXSR，

如果TXBF中有数据，则写TXBUF将触发ovflow错误。

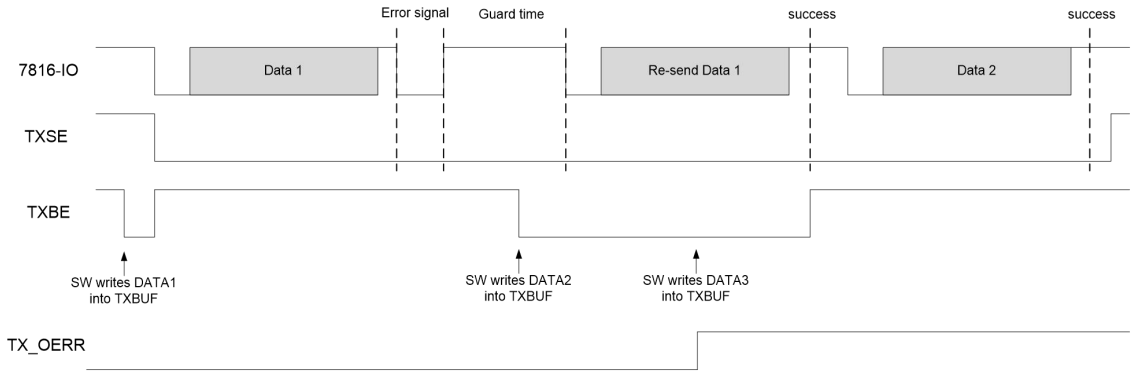


图 21-107816 发送和重发

21.5.5 数据接收

UART 的串行数据接收电路主要包括一个接收移位寄存器(RSR)。当接收到停止位后，RSR 就把接收到的数据送入接收缓冲区(RXBUFFER)，传送完成后，在每次接收数据送入接收缓冲区后将中断标志 RXBF 置 1。当接收缓冲区已满时，RSR 接收到一帧数据后仍会将其写入接收缓冲区，即覆盖缓冲区中原有数据，并且再次置位 RXBF，同时发生接收溢出错误，OERR 被置 1；软件写 1 或者读取 RXBUF 都可以清除 OERR 标志。

接收过程中，如果没有检测到正确的停止位，则发生帧格式错，FERR 被置 1；如果发生奇偶校验错，标志位 PERR 被置 1。

推荐的异步接收操作如下：

- 选择合适的波特率，初始化 SPBRG
- 若需要中断，置位 RXBF_IE
- 设置数据接收的格式：设置 PDSEL 寄存器，决定发送数据长度；设置 PARITY 寄存器选择是否发送校验位以及校验类型，设置 STOPSEL 寄存器决定发送 1 位还是 2 位停止位
- 使能接收模块：置位 RXEN
- 在一帧接收完毕时，RXBF 位会置 1，如果 RXBF_IE 位为 1，将会产生中断
- 读取 PERR、FERR、OERR 寄存器，判断是否有数据错误或者溢出
- 读取 RXBUF 寄存器中的接收数据

智能卡模式

智能卡模式下，UART完成数据和校验位接收后，自动判断校验位是否正确。如果校验位正确，则

判决停止位，并在正确的停止位后置位接收完成中断，或在错误的停止位置位帧格式错误中断。如果校验位错误，则自动回发错误信号并等待对方重发；如果重发次数达到了寄存器设定的上限，仍没有接收到正确的帧，则停止回发错误信号，并置位接收错误中断。

21.5.6 RS232 硬件流控

UART模式下，通过nCTS输入和nRTS输出，可以实现两个设备之间的数据流控制，如下图：

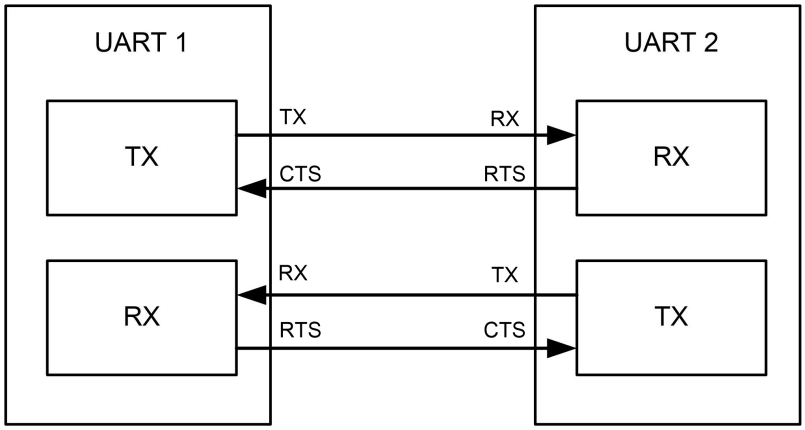


图 21-11 两个 UART 之间的硬件流控

RS232 RTS和CTS流控功能由RTSEN和CTSEN寄存器分别独立控制。

当RTSEN=1，nRTS在UART接收电路准备就绪时保持低电平，当UART数据缓冲区满且接收移位寄存器满，此时无法继续接收数据，nRTS信号拉高指示对方暂停发送新数据。当数据缓冲区被读取后，nRTS重新拉低通知对方可以发送新数据。

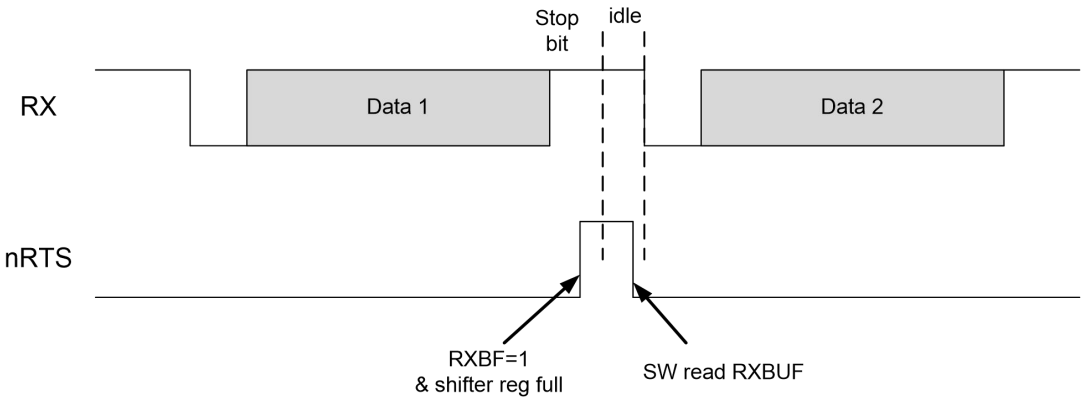


图 21-12nRTS 流控

当CTSEN=1，UART发送数据前先检查nCTS输入，如果nCTS为低电平，表示对方可以接收数据，此时UART正常发送；如果nCTS为高电平，表示对方暂时不能接收，则UART推迟发送直到nCTS变成低电平。在CTSEN=1的情况下，nCTS输入翻转时CTSIF标志自动置位，并且可以产生中断。

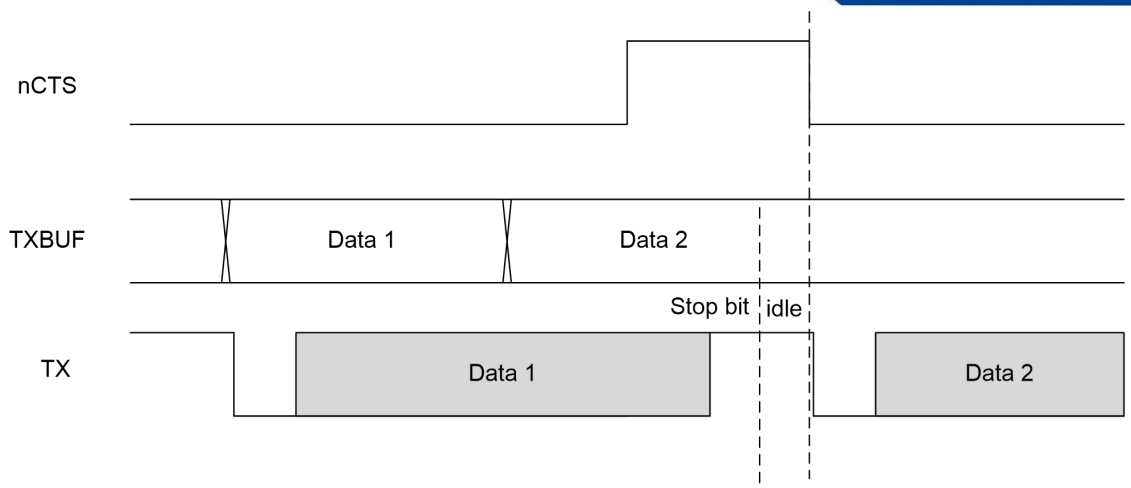


图 21-13nCTS 流控

21.5.7 低功耗休眠唤醒

USART不支持休眠唤醒

21.5.8 使用 DMA 进行 UART 收发

当 UART 模块被使能后，UART 模块在发送缓冲寄存器空和接收缓冲寄存器满时都会自动产生相应的 DMA 请求。应用软件需要事先配置 DMA 通道连接，将特定通道指向 UART 外设，设置 RAM 访问的指针地址，并使能 DMA 通道。此后 DMA 会自动响应 UART 请求，并完成 RAM 和 UART 之间的数据搬运。

21.5.9 DMA 模式下的发送完成中断

当 UART 通过 DMA 进行数据发送时，DMA 会在指定长度的数据传输完成后产生 DMA 通道中断。但是当通道中断产生时，最后一帧数据刚刚被写入 UART 发送缓冲区，还未被发送出去。

通过配置 DMATXIFCFG 寄存器，可以实现 DMA 传输完成、并且最后一帧数据发送完成的情况下，产生一个发送完成中断（缓冲区空或者移位寄存器空），以便实现所有数据全部发送出去后，再中断 CPU 的应用场景。

软件工作流程说明如下：

- 配置DMA通道为UART发送
- 关闭DMA通道中断使能
- 置位DMATXIFCFG寄存器，仅允许最后一帧数据产生中断输出
- 准备待发送数据，使能DMA

- 置位UART TXBE_IE或TXSE_IE寄存器，允许中断产生
- UART连续发送，直到最后一帧，发送期间不会产生TXBE或TXSE中断
- 最后一帧发送完成后，UART产生TXBE或TXSE中断

下表假设 UART 通过 DMA 发送 N 个帧：

TXBE_IE TXSE_IE	DMATXIFCFG	Frame No.	TXBE TXSE	UART interrupt
0	x	1~N	每帧发送完成后置位	不产生
1	0	1~N	每帧发送完成后置位	不产生
	1	1~N-1	每帧发送完成后置位	不产生
		N	每帧发送完成后置位	产生

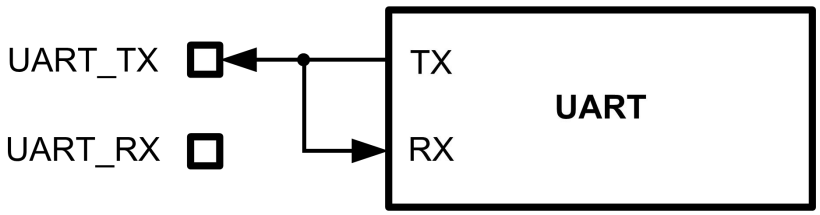
表 21-3 DMA 发送中断

21.5.10 半双工单线通信

当HDSEL寄存器置位时，UART以半双工模式工作，此时仅需一根信号线即可完成通讯。半双工模式仅在UART功能时可以使用，禁止在LIN和智能卡模式下使用，即置位HDSEL时必须保证MODE=0。

当HDSEL置位后：

- TX和RX信号在内部连接
- RX引脚不再需要使用，可以用于GPIO或者其他功能
- 当UART不发送数据时，TX引脚输出使能自动关闭，输入使能自动打开，引脚保持浮空状态，需要通过上拉电阻赋予确定电平
- TX引脚可以配置为开漏输出，数据冲突必须由软件处理
- 数据发送时UART自动屏蔽数据接收



21.6 波特率发生

21.6.1 波特率发生

波特率因子寄存器是一个 16 位的可读写的寄存器，其值 X 为 16—65535 之间的任一整数。

UART 波特率计算公式：

Baud = F_{CLK} / (SPBRG+1)；

注：F_{CLK} 在不同的 UART 中可以是不同的时钟，对于 USART，F_{CLK} 就是 APBCLK；对于 UART-LIN，F_{CLK} 是独立与 APBCLK 的工作时钟。

为了支持全双工通信，接收和发送波特率单独产生；

下表是常用系统时钟频率下的波特率：

Baud	F_{CLK}=16MHz			F_{CLK}=8MHz		
bps	Actual (bps)	Error%	X+1	Actual (bps)	Error%	X+1
300	300.0019	0.000625	53333	299.9963	-0.00125	26667
1200	1200.03	0.0025	13333	1199.94	-0.005	6667
2400	2399.88	-0.005	6667	2400.24	0.010001	3333
4800	4800.48	0.010001	3333	4799.04	-0.02	1667
9600	9598.08	-0.02	1667	9603.842	0.040016	833
19200	19207.68	0.040016	833	19184.65	-0.07994	417
38400	38369.3	-0.07994	417	38461.54	0.160256	208
57600	57553.96	-0.07994	278	57553.96	-0.07994	139
115200	115107.9	-0.07994	139	115942	0.644122	69
230400	231884.1	0.644122	69	228571.4	-0.79365	35
460800	457142.9	-0.79365	35	470588.2	2.124183	17

Baud	F_{CLK}=24MHz			F_{CLK}=32MHz		
bps	Actual (bps)	Error%	X+1	Actual (bps)	Error%	X+1
300	300	0	80000	299.9991	-0.00031	106667
1200	1200	0	20000	1199.985	-0.00125	26667
2400	2400	0	10000	2400.06	0.0025	13333
4800	4800	0	5000	4799.76	-0.005	6667
9600	9600	0	2500	9600.96	0.010001	3333
19200	19200	0	1250	19196.16	-0.02	1667
38400	38400	0	625	38415.37	0.040016	833
57600	57553.96	-0.07994	417	57553.96	-0.07994	556
115200	115384.6	0.160256	208	115107.9	-0.07994	278
230400	230769.2	0.160256	104	230215.8	-0.07994	139
460800	461538.5	0.160256	52	463768.1	0.644122	69

表 21-4 常用时钟频率下波特率计算

智能卡模式下，通信波特率由 UARTx_CODR 和 UARTx_SPBRG 寄存器共同决定。

UARTx_CODR 用于对 APBCLK 预分频得到 7816 接口时钟，并且这个时钟可以输出给外部卡片。

F7816=FAPBCLK/(CLKDIV+1)

7816 通信波特率使用 7816 接口时钟产生：

Baud ＝ F7816/(SPBRG ＋ 1)

智能卡模式下 SPBRG 最小可用值是 0x1，禁止配置 0x0

21.6.2 波特率自适应

利用 Timer 的 Capture 功能，可以实现波特率自适应功能。可实现的一种方法为，外部 UART 设备按约定的数据内容(比如 0xF8)发送一帧，由 Timer 对该帧数据的高电平脉宽进行计数，MCU 读取 Timer 捕捉结果计算得到波特率因子，并写入波特率发生寄存器中，作为波特率发生的时钟分频计数值 X 使用。这时接收状态复位，重新等待起始位，以写入的波特率因子所产生的波特率接收数据。参考 Timer 章节。

21.7 红外调制

TZBRG 寄存器保存一个 11 位的分频系数 X，其值为 0~2047 之间的任一整数。所有 UART 共用一个红外调制频率发生器。

红外调制频率计算公式：

$FIR = F_{APBCLK} / (TZBRG+ 1)$

红外调制的方式为：发送数据 0 时调制红外频率，发送数据 1 时不调制。为满足 PNP 和 NPN 两种红外驱动管的需求，寄存器 IRFLAG 位控制红外调制输出的极性。IRFLAG=0 时为正极性输出，适合 PNP 管驱动；IRFLAG=1 时为负极性输出，适合 NPN 管驱动。

TH 寄存器用于配置红外调制占空比

占空比：Y = (TZBRG[10:4] * TH) /(TZBRG+1)

当 TH=4’b0000 时，占空比为 Y = (TZBRG[10:1]+1)/(X+1)；

当 TZBRG[10:4]=7’h00 时，占空比为 Y = TH/(TZBRG[3:0]+ 1)；若此时 TH>TZBRG [3:0]，则红外调制时钟 IRCLK 为固定高电平。

当红外调制极性反向时（IRFLAG=1），占空比也为 1-Y

红外调制波形见下图：

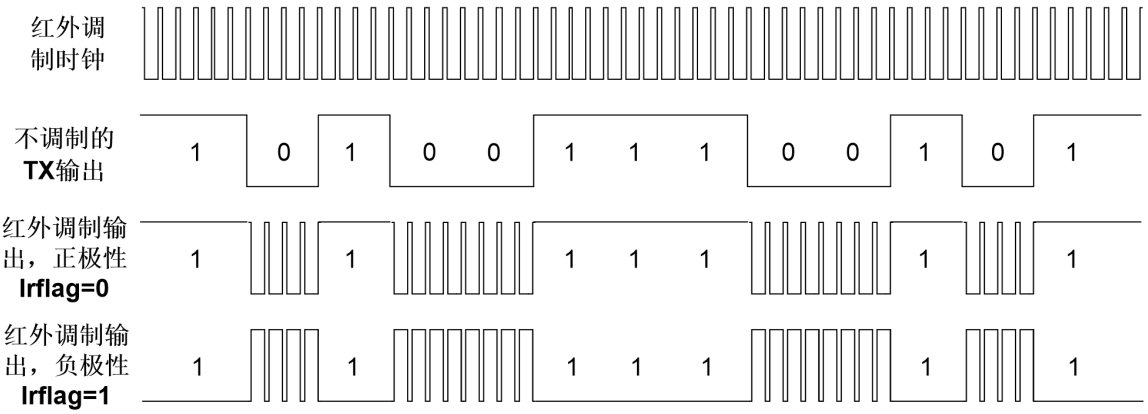


图 21-14 红外调制波形

无论有效电平是 0 还是 1，占空比定义为高电平长度/周期。

21.8 接收超时

针对 MODBUS 等时间敏感型应用，设计了接超时机制。当使能 RXTOEN 寄存器后，超时计数器以波特率时钟计数，当每次收到一个完整的数据帧，将清零超时计数器并重新开始计数。超时溢出的上限值可以由软件配置，最大 255 波特。

21.9 发送延迟

通过 TXDLY_LEN 寄存器，可以控制两个数据帧发送之间的间隔时间，单位是波特。发送延迟是从上一帧最后一个 STOP 位结束，到下一帧起始位之间的间隔。

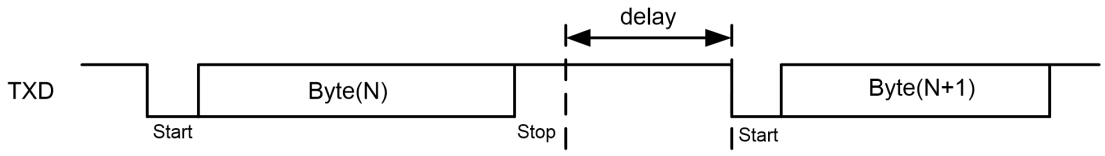


图 21-15 UART 发送延迟

21.10 寄存器

UART4模块基地址：0x4001 1C00

UART5模块基地址：0x4001 2000

UARTIR模块基地址：0x4001 7C00

offset 地址	名称	符号
UARTIR 寄存器（shared）		
0x00	红外调制寄存器 （Infrared modulation Control Register）	UART_IRCR
USARTx 寄存器		
0x00	USART 控制状态寄存器 （USART Control Status Register）	USART_CSR
0x04	USART 中断使能寄存器 （USART Interrupt Enable Register）	USART_IER
0x08	USART 中断标志寄存器 （USART Interrupt Status Register）	USART_ISR
0x10	USART 接收缓冲寄存器 （USART Receive Buffer）	USART_RXBUF
0x14	USART 发送缓冲寄存器 （USART Transmit Buffer）	USART_TXBUF
0x18	USATR 波特率产生寄存器 （USART Baud rate Generator Register）	USART_BGR
0x20	USART 模式寄存器 （USART Mode Control Register）	USART_MCR
0x40	USART 智能卡状态标志寄存器 （USART Smart Card Interrupt Status Register）	USART_SCISR
0x44	USART 智能卡中断使能寄存器 （USART Smart Card Interrupt Enable Register）	USART_SCIER
0x48	USART 智能卡帧格式控制寄存器 （USART Smart Card Frame Format Control Register）	USART_FFCR
0x4C	USART 智能卡额外保护时间寄存器 （USART Smart Card Extra Guard Time Register）	USART_EGTR
0x50	USART 智能卡输出时钟分频寄存器 （USART Smart Card Clock Output Register）	USART_CODR

21.10.1 红外调制寄存器（UART_IRCR）

名称	UART_IRCR
offset	0x00

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IRFLAG	TH				TZBRG[10:8]		
位权限	R/W-0	R/W-0000				R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TZBRG[7:0]							
位权限	R/W-11010010							

位号	助记符	功能描述
31:16	-	**未实现：读为0**
15	**IRFLAG**	控制红外调制发送数据时的默认输出极性 (Infra Red) 0：正极性 1：负极性
14:11	**TH**	红外占空比调制参数 (Tranmission High Duty)
10:0	**TZBRG**	红外调制频率 (Transmission Baud Rate)

21.10.2 USARTx 模式寄存器（USARTx_MCR）

名称	USARTx_MCR							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					HDSEL	-	MODE
位权限	U-0					R/W-0	U-0	R/W-0

位号	助记符	功能描述
31:3	-	未实现：读为0
2	**HDSEL**	UART 半双工单线模式选择 0：双线 UART 1：单线半双工 UART *注：此寄存器仅在 MODE=0 时有效*
1	-	未实现：读为0

位号	助记符	功能描述
0	**MODE**	模块控制寄存器 0：标准 UART 模式 1：智能卡模式（ISO7816-3）

21.10.3 USARTx 控制状态寄存器（USARTx_CSR）

名称	USARTx_CSR(x=0~7)							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							BUSY
位权限	U-0							R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RTSEN	CTSEN	-				TXIREN	RXTOEN
位权限	R/W-0	R/W-0	U-0				R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	CKOE	OVSM	IOSWAP	-	DMATXIFCFG	BITORD	STOPCFG
位权限	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PDSEL		PARITY		RXPOL	TXPOL	RXEN	TXEN
位权限	R/W-01		R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:25	-	未实现：读为0
24	**BUSY**	UART 通信标志，只读 (Busy) 1：UART 正在通信中 0：UART 空闲
23	**RTSEN**	RS232 RTS 流控使能，仅 UART 模式有效 0：关闭 RTS 流控 1：使能 RTS 流控
22	**CTSEN**	RS232 CTS 流控使能，仅 UART 模式有效 0：关闭 CTS 流控 1：使能 CTS 流控
21:18	-	未实现：读为0
17	**TXIREN**	发送红外调制使能 (Transmit Infra-red modulation Enable) 1：使能红外调制发送 0：关闭红外调制发送
16	**RXTOEN**	接收超时使能 (Receive Time-Out Enable) 1：使能接收超时功能 0：关闭接收超时功能
15	-	未实现：读为0
14	**CKOE**	U7816 时钟 CLK 输出使能控制位 1:7816 时钟输出使能 0:7816 时钟输出禁止

位号	助记符	功能描述
13	**OVSM**	接收过采样模式选择 (Oversampling mode) 0：16 倍过采样 1：8 倍过采样 *注意：在 LIN 模式下，固定为 16 倍过采样，此寄存器无效*
12	**IOSWAP**	RX 和 TX 引脚交换 0：默认引脚顺序（与封装图一致） 1：交换引脚顺序
11	**-**	未实现：读为0
10	**DMATXIFCFG**	DMA发送完成中断使能，仅在UART通过DMA进行发送时有效 (DMA transmit interrupt enable) 1：IE=1的情况下，DMA模式下发送完最后一帧后，允许中断信号输出；最后一帧之前的数据帧发送完成后不允许中断信号输出 0：是否允许中断信号输出仅由IE决定
9	**BITORD**	数据发送/接收时的位顺序 (Bit Order) 0：LSB first 1：MSB first
8	**STOPCFG**	停止位宽度配置，仅对发送帧格式有效，接收时不判断停止位个数 (Stop bit config) 0：1位停止位 1：2 位停止位
7:6	**PDSEL**	每帧的数据长度选择；此寄存器对数据发送和接收同时有效 (Payload data length Select) 00：7 位数据 01：8 位数据 10：9 位数据 11：6 位数据
5:4	**PARITY**	校验位配置；此寄存器对数据发送和接收同时有效 00：无校验位 01：偶校验 10：奇校验 11：RFU
3	**RXPOL**	接收数据极性配置 (Receive Polarity) 0：正向 1：取反
2	**TXPOL**	发送数据极性配置 (Transmit Polarity) 0：正向 1：取反
1	**RXEN**	接收使能，1 有效 (Receive Enable)
0	**TXEN**	发送使能，1 有效 (Transmit Enable)

21.10.4 USARTx 中断使能寄存器（USARTx_IER）

名称	**USARTx_IER**							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CTSIE	-			RXTO_IE	RXERR_IE	-	RXBF_IE
位权限	R/W-0	U-0			R/W-0	R/W-0	U-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						TXBE_IE	TXSE_IE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:16	-	**未实现：读为0**
15	**CTSIE**	nCTS翻转中断使能，1有效
14:12	-	**未实现：读为0**
11	**RXTO_IE**	接收超时中断使能，1 有效 (Receive Time-Out Interrupt Enable)
10	**RXERR_IE**	接收错误中断使能，1 有效(Receive Error Interrupt Enable)
9	-	**未实现：读为0**
8	**RXBF_IE**	接收缓存满中断使能，1 有效 (Receive Buffer Full Interrupt Enable)
7:2	-	**未实现：读为 0**
1	**TXBE_IE**	发送缓存空中断使能，1 有效 (Transmit Buffer Empty Interrupt Enable)
0	**TXSE_IE**	发送缓存空且发送移位寄存器空中断使能，1 有效 (Transmit Shift register Empty Interrupt Enable)

21.10.5 USARTx 中断标志寄存器（USARTx_ISR）

名称	**USARTx_ISR**							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					PERR	FERR	OERR
位权限	U-0					R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CTSIF	-			RXTO	-		RXBF
位权限	R/W-0	U-0			R/W-0	U-0		R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	-				TX_OERR	TXBE	TXSE
位权限	U-0	U-0				R/W-0	R -0	R/W-0

位号	助记符	功能描述
31:19	-	未实现：读为0
18	**PERR**	奇偶校验错误中断标志，硬件置位，软件写 1 清零 (Parity Error,write 1 to clear)

位号	助记符	功能描述
17	**FERR**	帧格式错误中断标志，硬件置位，软件写 1 清零 (Frame Error flag,write 1 to clear)
16	**OERR**	接收缓存溢出错误中断标志，当接收缓存满的情况下，收到新的数据时置位；硬件置位，软件写 1 或者读取 RXBUF 时清零 接收溢出时，接收缓冲器中原有的数据被新数据覆盖。 (RX buffer Overflow Error flag,write 1 to clear)
15	**CTSIF**	CTS 信号翻转标志，当 nCTS 输入发生翻转时置位，软件写 1 清零
14:12	**-**	未实现：读为0
11	**RXTO**	接收超时中断标志，硬件置位，软件写 1 清零 (Receive Time-Out flag,write 1 to clear) （仅 UART0 和 UART1 有效）
10:9	**-**	未实现：读为0
8	**RXBF**	接收缓存满中断标志，硬件置位，软件读取 RXBUF 时清零 (Receive Buffer Full flag write 1 to clear)
7:3	**-**	未实现：读为 0
2	**TX_OERR**	发送缓存溢出中断标志，硬件置位，软件写 1 清零 (Transmit Buffer Overflowflag)
1	**TXBE**	发送缓存空中断标志，硬件置位，软件写发送缓存时清零 (Transmit bufferEmpty flag)
0	**TXSE**	发送缓冲区空且发送移位寄存器空标志，硬件置位，待发送数据从 TXBUF 写入发送移位寄存器时清零 (Transmit Shift register Empty flag)

21.10.6 USARTx 超时和延迟寄存器（USARTx_TODR）

名称	USARTx_TODR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TXDLY_LEN							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXTO_LEN							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:16	**-**	**未实现：读为0**
15:8	**TXDLY_LEN**	发送延迟，最大 255baud (Transmit Delay Length)
7:0	**RXTO_LEN**	接收超时溢出长度，最大 255baud (Receive Time-Out Length)

21.10.7 USARTx 接收缓冲寄存器（USARTx_RXBUF）

名称	USARTx_RXBUF							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							RXBUF[8]
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:9	-	未实现：读为0
8:0	**RXBUF**	接收数据缓冲寄存器数据(Receive buffer)

7位收发时，接收的7bits数据存入RXBUF[6:0]

21.10.8 USARTx 发送缓冲寄存器（USARTx_TXBUF）

名称	USARTx_TXBUF							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							TXBUF[8]
位权限	U-0							W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:9	-	未实现：读为0
8:0	**TXBUF**	发送数据缓冲寄存器数据(Transmit Buffer)

7位收发时，发送的7bits数据写入TXBUF[6:0]

21.10.9 USATRx 波特率产生寄存器（USARTx_BGR）

名称	USARTx_BGR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SPBRG[15:8]							
位权限	R/W-00000011							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SPBRG[7:0]							
位权限	R/W-01000001							

位号	助记符	功能描述
31:16	-	未实现：读为0
15:0	SPBRG	波特率产生器寄存器值(Serial Port Baud Rate Generation)

波特率计算详见波特率发生章节
注：当SPBRG <= 0x000F时，UARTDIV=16'H000F;
当SPBRG >0x000F时，UARTDIV=SPBRG;

智能卡模式下，Baud ＝ F_{7816}/(SPBRG ＋ 1)，F_{7816}为智能卡输出工作时钟频率，由CODR寄存器配置

智能卡模式下SPBRG最小可用值是0x1，禁止配置0x0

21.10.10 USARTx 智能卡状态标志寄存器（USARTx_SCISR）

名称	USARTx_SCISR							
offset	0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					WAIT_RPT	TXBUSY	RXBUSY
位权限	U-0					R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				TPARERR	RPARERR	FRERR	OVERR
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TXSE	RXIF	TXIF	ERRIF
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:19	--	RFU：未实现，读为 0
18	WAIT_RPT	U7816 接口发送了错误信号，正在等待对方重发数据；(Waiting for Repeat flag) 状态机进入发送错误信号状态时置位，收到数据起始位或者进入发送状态时硬件清零； 软件只读。
17	TXBUSY	发送数据忙标志。（发送完成后自动清零）(Transmission busy flag) 1：处于数据发送状态，发送移位寄存器正在发送数据。（开始发送起始位置 1，停止位中间清零） 0：数据发送空闲
16	RXBUSY	接收数据忙标志。（接收完成后自动清零）(Receiving busy flag) 1：处于数据接收状态，接收移位寄存器正在接收数据。（收到起始位置 1，收到停止位清零，若接收数据出错需重发，则回发 error signal 时清零。即数据及校验位接收之后，无论是否需要重发，都需要及时清除该标志） 0：数据接收空闲
15:12	--	RFU：未实现，读为 0
11	TPARERR	发送数据奇偶校验错误标志位。硬件置位，写 1 清零 (Transmit Parity Error,write 1 to clear)
10	RPARERR	接收数据奇偶校验错误标志位。硬件置位，写 1 清零 (Receive Parity Error flag,write 1 to clear)
9	FRERR	接收帧格式错误标志位。硬件置位，写 1 清零 (Frame Error flag,write 1 to clear) 1：帧格式有错误，接收到的 frame 字节长度有误或接收到的 frame 或者 stop 位有误 0：接收数据时无奇偶校验错误
8	OVERR	接收溢出错误标志位。硬件置位，写 1 清零 (Receive Overflow Error,write 1 to clear) 1：接收缓冲寄存器未被读出，又接收到新的数据，溢出错误标志有效。原接收缓冲寄存器内数据被新覆盖 0：无溢出错误
7:4	--	RFU：未实现，读为 0
3	TXSE	7816 数据发送完成标志
2	RXIF	接收完成标志(Receive interrupt flag)，U7816 接口控制器每收到 1byte 数据，根据接收的通道相应发出一次中断。硬件置位，读数据接收缓冲寄存器清零 1：接收到 1byte 数据，数据接收缓冲器满 0：未接收到数据，数据接收缓冲器空
1	TXIF	发送缓冲区空标志(Transmit interrupt flag)，上电复位后此标志就自动置位，表示缓冲区空，可以写入数据。软件写入数据后标志自动清除，数据从发送缓存移入移位寄存器后置 1 1：数据发送缓冲器空 0：数据发送缓冲器内有数据待发送
0	ERRIF	错误标志(Error interrupt flag)，寄存器配置出错或传输过程中出错。此 bit 是 TPARERR、RPARERR、FRERR、OVERR 的或。软件通过清除以上错误标志寄存器来清除此 bit。

*仅智能卡模式下有效

21.10.11 USARTx 智能卡中断使能寄存器（USARTx_SCIER）

名称	USARTx_SCIER							
offset	0x44							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TXSEIE	RXIE	TXIE	LSIE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3	TXSEIE	7816 发送完成中断使能 1 =发送完成中断使能 0 = 发送完成中断禁止
2	RXIE	数据接收中断使能位。对应 RXIF 中断标志位 1 = 数据接收中断使能 0 = 数据接收中断禁止
1	TXIE	数据发送中断使能位。对应 TXIF 中断标志位 1 = 数据发送中断使能 0 = 数据发送中断禁止
0	LSIE	线路状态中断使能位。对应 ERRIF 中断标志位 1 = 线路状态中断使能 0 = 线路状态中断禁止

21.10.12 USARTx 智能卡帧格式控制寄存器（USARTx_FFCR）

名称	USARTx_FFCR							
offset	0x48							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				SFREN	ERSW		ERSGD
位权限	U-0				R/W-0	R/W-00		R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	BGTEN	REP_T	PAR	FREN	TREPEN	RREPEN	DICONV
位权限	R/W-0	R/W-0	R/W-00	R/W-0	R/W-1	R/W-1	R/W-0

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11	SFREN	Guard Time 发送长度控制位 1 = Guard time 为 3etu 0 = Guard time 为 2etu
10:9	ERSW	ERROR SIGNAL 宽度选择 (Error Signal Width) 11：ERROR SIGNAL 宽度为 1ETU; 10：ERROR SIGNAL 宽度为 1.5ETU; 01：ERROR SIGNAL 宽度为 2ETU; 00：ERROR SIGNAL 宽度为 2ETU;
8	ERSGD	ERROR SIGNAL 后 GUARDTIME 宽度选择（仅在发送时有效）(Error Signal Guard Time) 1：ERROR SIGNAL 后 GUARDTIME 为 1~1.5ETU。 0：ERROR SIGNAL 后 GUARDTIME 为 2~2.5ETU。 ERROR SIGNAL 宽度为整数 ETU 时 GUARDTIME 为 1.5 或 2.5ETU；ERROR SIGNAL 宽度为 1.5ETU 时 GUARDTIME 为 1 或 2ETU
7	BGTEN	BGT 控制位。控制接收->发送之间是否插入 BGT。BGT 是接收->发送之间需要的最小时间（block guard time enable） 1：BGT 使能，插入 Block guard time(12 etu); 0：BGT 禁止，不插入 Block guard time(12 etu);
6	REP_T	控制接收数据奇偶校验出错时自动重发次数 (Repeated Times) 00：Even 01：Odd 10：Always 1 11：不校验
5:4	PAR	奇偶校验类型选择 (Parity) 00：Even 01：Odd 10：Always 1 11：不校验，处理
3	FREN	Guard Time 接收长度控制位 (Receive short Frame) 1：Guard time 为 1 etu 0：Guard time 为 2 etu
2	TREPEN	发送数据奇偶校验错的处理方式选择 (Transmit Repeat Enable) 1：收到奇偶校验出错标志（error signal），根据 T＝0 协议自动进行回发。在单一 byte 重复发送次数超过 REP_T 后，置 tx_parity_err 标志，进行中断 0：收到 Error signal 时不进行自动回发，置 tx_parity_err 标志，直接中断
1	RREPEN	接收数据奇偶校验错的处理方式选择 (Receive Repeat Enable) 1：奇偶校验错，根据 T=0 协议自动回发 ERROR SIGNAL。单一 BYTE 连续接收次数超过 REP_T 后，置 RX_PARITY_ERR 标志，进行中断

Bit	助记符	功能描述
		0：奇偶校验错，不自动发送 ERROR SIGNAL，置 RX_PARITY_ERR 标志，进行中断
0	DICONV	传输次序，编码方式选择 1 = 反向编码，先收发 MSB；(收发数据+校验位)反逻辑电平 0 = 正向编码，先收发 LSB ； (收发数据+校验位)正逻辑电平

21.10.13 USARTx 智能卡额外保护时间寄存器（USARTx_EGTR）

名称	USARTx_EGTR							
offset	0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXEGT							
位权限	R/W-00000000							

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	TXEGT	发送时插入的 EGT 时间（以 ETU 为单位） (Transmit Extra Guard Time)

21.10.14 USARTx 智能卡输出时钟分频寄存器（USARTx_CODR）

名称	USARTx_CODR (x=4~7)							
offset	0x50							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			CLKDIV				
位权限	U-0			R/W-00011				

Bit	助记符	功能描述

Bit	助记符	功能描述
31:5	--	RFU：未实现，读为 0
4:0	CLKDIV	U7816 时钟输出分频控制寄存器(Clock Divider)，控制 7816 工作时钟分频数。 U7816 工作时钟与 APBCLK 的分频关系： $F_{7816}=F_{APBCLK}/(CLKDIV+1)$ 特殊情况：CLK_DIV 设置成 0 或 1 时，$F_{7816}=F_{APBCLK}/2$ 注：7816 协议规定的工作时钟范围是 1~5MHZ。

22 低功耗 UART（LPUART）

22.1 概述

LPUART 是增强型异步串行通信接口，其工作时钟可以选择 32768Hz 晶振时钟（XTLF）、高频环振时钟（RCHF）。LPUART 可以支持到最高 9600 波特率的数据接收，此时 LPUART 功耗极低，可以在 Sleep/DeepSleep 模式下工作。

特点：

- 异步数据收发
- 3路独立LPUART
- 标准UART帧格式
  - 1bit起始位
  - 可配置数据长度，支持6、7、8、9bits
  - 奇校验、偶校验或无校验位
  - 1或2bit停止位
- 可编程数据极性
- 当工作时钟为XTLF时，支持Sleep/DeepSleep模式下的数据收发
- 中断标志
  - 接收Buffer满
  - 接收Buffer溢出
  - 接收帧格式错误
  - 接收校验位错误
  - START检测
  - 数据匹配
  - 发送完成
- 休眠模式下唤醒芯片

- RXD下降沿唤醒
- 起始位检测唤醒
- 1字节接收完成唤醒
- 1字节数据匹配唤醒

22.2 结构框图

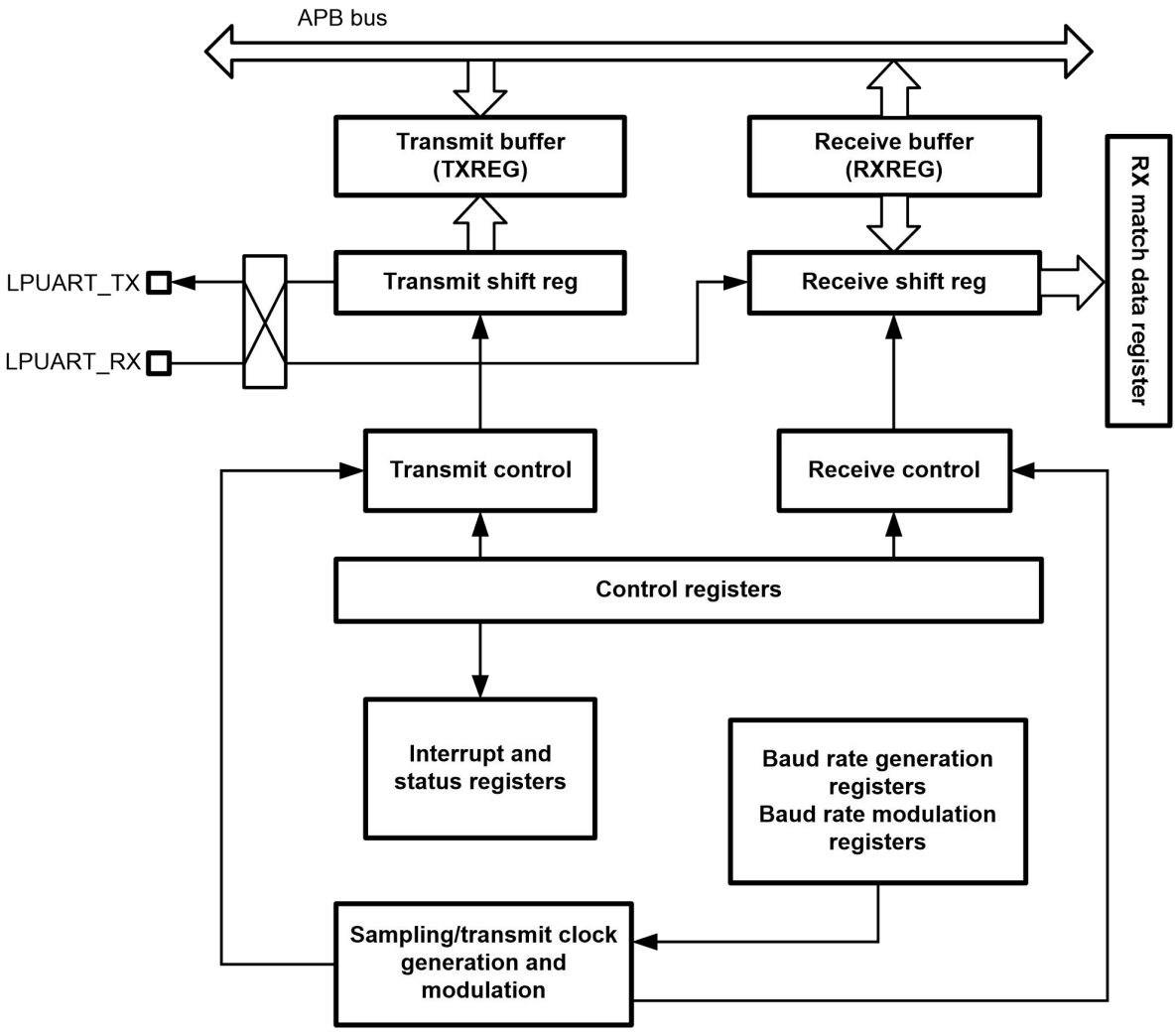


图 22-1 LPUART 结构框图

22.3 引脚定义

LPUART 模块使用 2 个引脚与外部器件通信，每个 UART 的收发信号可能被映射到不同的 GPIO 上。

引脚	UARTx	符号	功能
PA2，PA13	LPUART0	LPUART0_RX	数据接收
PA3，PA14		LPUART0_TX	数据发送
PB13，PC2	LPUART1	LPUART1_RX	数据接收
PB14，PC3		LPUART1_TX	数据发送
PA11，PB4	LPUART2	LPUART2_RX	数据接收
PA12，PB5		LPUART2_TX	数据发送

表 22-1 LPUART 引脚对应表

22.4 工作时钟

LPUART 使用独立于 APBCLK 的时钟进行数据收发，工作前需要在 CMU 模块中配置相关寄存器。

LPUART 使用 XTLF 或 RCLP 工作。由于 RCLP 精度不高，在使用 RCLP 进行 LPUART 通信前必须先进行时钟校准，将 RCLP 校准到+/-1%以内。

22.5 字符描述

LPUART 传输字符的基本时序如下图所示。每个字符帧包含至少 1bit START 位和至少 1bit STOP 位，数据长度可以配置为 6~9bits，并且可以选择有无校验位。

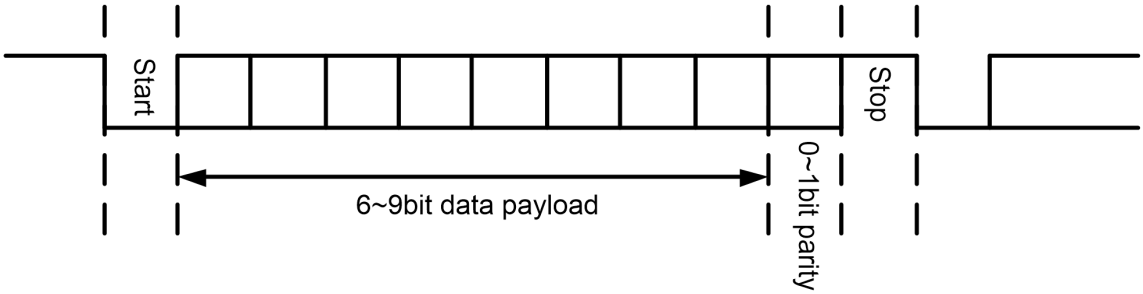


图 22-2 字符描述

LPUART 支持多种帧格式，由 LPUARTxCSR.PDSEL 寄存器和 LPUARTxCSR.PARITY 寄存器控制。见下表：

PDSEL	PARITY	帧格式[1]
00	00	[Start | 7 bits data | Stop]

	01, 10	[Start | 7 bits data | Parity | Stop]
01	00	[Start | 8 bits data | Stop]
	01, 10	[Start | 8 bits data | Parity | Stop]
10	00	[Start | 9 bits data | Stop]
	01, 10	[Start | 9 bits data | Parity | Stop]
11	00	[Start | 6 bits data | Stop]
	01, 10	[Start | 6 bits data | Parity | Stop]

表 22-2LPUART 数据帧格式

[1]：Stop 位可能是 1bit 或者 2bit，根据 STOPCFG 寄存器决定

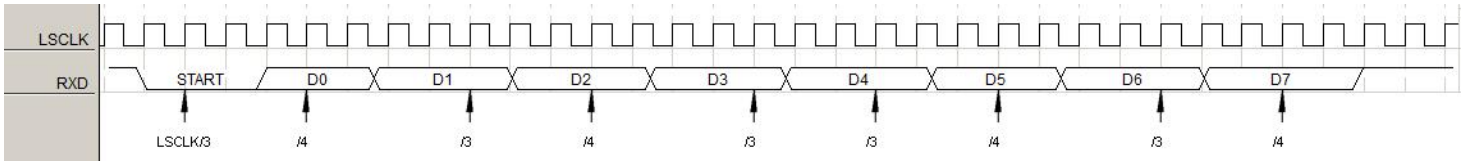
注意 PDSEL 寄存器用于配置帧的数据长度，通信帧长为【起始位+数据位+校验位+停止位】。

22.6 功能描述

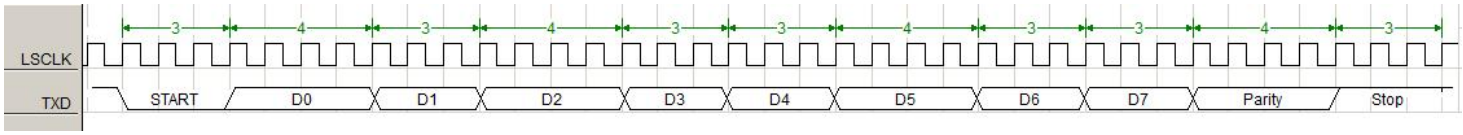
22.6.1 位接收采样和发送

当工作时钟选择为 LSCLK 或 RCHF 分频时，频率仅为 32Khz 左右，此时标准串口无法支持 9600bps 通信，因此需要引入 bit 调制设计。

由于 LPUART 工作时钟不是波特率的整数倍，采用固定分频系数的话会引入累积误差，在接收时采用 3、4 分频交替进行接收，确保在每个 bit 的中间位置采样，每个 bit 采样一次。每个 bit 采用 3 分频还是 4 分频，则由 MCTL 寄存器控制。例如：



类似于 LPUART 接收，LPUART 工作时钟不是波特率的整数倍，采用固定分频系数同样会引入累积误差，在发送时也采用了 3、4 分频交替进行发送，每个 bit 采用 3 分频还是 4 分频，则由 MCTL 寄存器控制。例如：



软件需要根据通信波特率的不同合理配置调制控制寄存器 MCTL，当工作时钟为 32768Hz 时，建议的配置参数表如下：

Baud	MCTL												
	Bit0（start）	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12
9600	0	1	0	0	1	0	1	0	1	0	1	0	0
4800	1	1	0	1	1	1	1	1	0	1	1	1	1
2400	1	1	0	1	1	0	1	1	0	1	1	0	1
1200	0	1	0	0	1	0	0	1	0	0	1	0	0
600	0	1	1	0	1	0	1	1	0	1	1	0	1
300	0	1	0	0	0	0	1	0	0	0	0	1	0

表 22-3 LPUART 数据位调制系数

22.6.2 接收流程

- 配置LPUBAUD寄存器决定波特率
- 根据波特率选择合适的调制参数，配置MCTL寄存器
- 配置LPUCON寄存器，选择帧格式、极性、中断参数等
- 配置LPUEN寄存器打开接收使能
- 等待中断事件

22.6.3 发送流程

- 配置LPUBAUD寄存器决定波特率
- 根据波特率选择合适的调制参数，配置MCTL寄存器
- 配置LPUCON寄存器，选择帧格式、极性、中断参数等
- 配置LPUEN寄存器打开发送使能
- 等待中断事件

22.6.4 休眠模式下的数据接收唤醒

LPUART 支持在 Sleep、DeepSleep 模式下进行数据接收并唤醒芯片。此时芯片功耗极低，并保持对 RXD 引脚的监听，直到特定事件到来后唤醒芯片退出休眠模式。

- 配置LPUBAUD寄存器决定波特率
- 根据波特率选择合适的调制参数，配置MCTL寄存器
- 配置LPUCON寄存器，选择帧格式、极性，通过LPUxCR.RXEV选择唤醒事件为START位、一帧接收完成、一帧数据匹配或RXD下降沿检测
- 配置LPUEN寄存器打开接收使能
- 软件进入Sleep/DeepSleep

22.7 寄存器

LPUART0模块基地址：0x40014000

LPUART1模块基地址：0x40014400

LPUART2模块基地址：0x40015000

offset 地址	名称	符号
LPUART 寄存器		
0x00	LPUART 控制状态寄存器 （LPUART Control Status Register）	LPUART_CSR

offset 地址	名称	符号
0x04	LPUART 中断使能寄存器 （LPUART Interrupt Enable Register）	LPUART_IER
0x08	LPUART 中断标志寄存器 （LPUART Interrupt Status Register）	LPUART_ISR
0x0C	LPUART 波特率调制寄存器 （LPUART Baud rate Modulation Register）	LPUART_BMR
0x10	LPUART 接收缓冲寄存器 （LPUART Receive Buffer Register）	LPUART_RXBUF
0x14	LPUART 发送缓冲寄存器 （LPUART Transmit Buffer Register）	LPUART_TXBUF
0x18	LPUART 数据匹配寄存器 （LPUART data Matching Register）	LPUART_DMR

22.7.1 LPUARTx 控制状态寄存器（LPUARTx_CSR）

名称	LPUARTx _CSR(x=0,1,2)							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							BUSY
位权限	U-0							R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				WKBYTE_CFG	-	RXEV	
位权限	U-0				R/W-0	U-0	R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				IOSWAP	RFUI	BITORD	STOPCFG
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PDSEL		PARITY		RXPOL	TXPOL	RXEN	TXEN
位权限	R/W-00		R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:25	-	未实现：读为0
24	**BUSY**	LPUART 通信标志，只读 (Busy) 1：LPUART 正在通信中 0：LPUART 空闲
23:20	-	未实现：读为0
19	**WKBYTE_CFG**	数据接收唤醒条件配置 (Wakeup Byte Config) 1：接收完1字节，并且奇偶校验和STOP位都正确，才触发唤醒中断 0：接收完1字节，不检查校验位和STOP位，直接触发唤醒中断
18	-	未实现：读为0

位号	助记符	功能描述
17:16	**RXEV**	唤醒中断事件配置，用于控制何种事件下向 CPU 提供唤醒中断 (Receive Wakeup Event) 00：START 位检测唤醒 01：1byte 数据接收完成 10：接收数据匹配成功 11：RXD 下降沿检测
15:12	**-**	未实现：读为0
11	**IOSWAP**	RX 和 TX 引脚交换 (IO swapping) 0：默认引脚顺序（与封装图一致） 1：交换引脚顺序
10	**RFUI**	保留位，可读写
9	**BITORD**	数据发送/接收时的位顺序 (Bit Order) 0：LSB first 1：MSB first
8	**STOPCFG**	停止位宽度配置，仅对发送帧格式有效，接收时不判断停止位个数 (Stop bit Config) 0：1位停止位 1：2位停止位
7:6	**PDSEL**	每帧数据长度选择；此寄存器对数据发送和接收同时有效 (Payload Data length Select) 00：7 位数据 01：8 位数据 10：9 位数据 11：6 位数据
5:4	**PARITY**	校验位配置；此寄存器对数据发送和接收同时有效 (Parity) 00：无校验位 01：偶校验 10：奇校验 11：RFU
3	**RXPOL**	接收数据极性配置 (Receive Polarity) 0：正向 1：取反
2	**TXPOL**	发送数据极性配置 (Transmit Polarity) 0：正向 1：取反
1	**RXEN**	接收使能，1 有效 (Receive Enable)
0	**TXEN**	发送使能，1 有效 (Transmit Enable)

22.7.2 LPUARTx 中断使能寄存器（LPUARTx _IER）

名称	LPUARTx _IER(x=0,1,2)							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			RXEV_IE	-	RXERR_IE	-	RXBF_IE
位权限	U-0			R/W-0	U-0	R/W-0	U-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						TXBE_IE	TXSE_IE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:13	-	未实现：读为0
12	**RXEV_IE**	接收唤醒事件中断使能，1 有效 (Receive Event Interrupt Enable)
11	-	未实现：读为0
10	**RXERR_IE**	接收错误中断使能，1 有效 (Receive Error Interrupt Enable)
9	-	未实现：读为0
8	**RXBF_IE**	接收缓存满中断使能，1 有效 (Receive Buffer Full Interrupt Enable)
7:2	-	未实现：读为 0
1	**TXBE_IE**	发送缓存空中断使能，1 有效 (Transmit Buffer Empty Interrupt Enable)
0	**TXSE_IE**	发送缓存空且发送移位寄存器空中断使能，1 有效 (Transmit Shift register Interrupt Enable)

22.7.3 LPUARTx 中断标志寄存器（LPUARTx _ISR）

名称	LPUARTx _ISR(x=0,1,2)							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							RXEVF
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				TXOV	PERR	FERR	OERR
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							RXBF
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						TXBE	TXSE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:25	-	未实现：读为0
24	**RXEVF**	接收唤醒事件中断标志，硬件置位，软件写 1 清零 (Receive Event Interrupt Flag,write 1 to clear) 中断标志触发源由 LPUxCR.RXEV 寄存器配置。
23:20	-	未实现：读为0

位号	助记符	功能描述
19	**TXOV**	发送缓存溢出错误，硬件置位，软件写1清零 (Transmit Overflow Error flag,write 1 to clear) 当发送缓存中的数据还未进入移位寄存器发送时，软件向发送缓存写入新数据，将触发TXOV标志置位。
18	**PERR**	奇偶校验错误中断标志，硬件置位，软件写 1 清零 (Parity Error flag,write 1 to clear)
17	**FERR**	帧格式错误中断标志，硬件置位，软件写 1 清零 (Frame Error flag,write 1 to clear)
16	**OERR**	接收缓存溢出错误中断标志，当接收缓存满的情况下，收到新的数据时置位；硬件置位，软件写 1 清零 (Receive Buffer Overflow Error flag,write 1 to clear)
15:9	**-**	未实现：读为0
8	**RXBF**	接收缓存满中断标志，硬件置位，软件写 1 或者读取 RXBUF 时清零 (Receive Buffer Full flag, write 1 to clear)
7:2	**-**	未实现：读为 0
1	**TXBE**	发送缓存空中断标志，硬件置位，写入 TXBUF 时清零 (Transmit Buffer Empty flag,write 1 to clear)
0	**TXSE**	发送缓存空且发送移位寄存器空中断标志，硬件置位，软件写 1 或者发送数据被载入移位寄存器时清零 (Transmit Shift register Empty flag, write 1 to clear)

22.7.4 LPUARTx 波特率调制寄存器（LPUARTx _BMR）

名称	LPUARTx _BMR(x=0,1,2)							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	MCTLEN	-		MCTL[12:8]				
位权限	R/W-1	U-0		R/W-00000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MCTL[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					BAUD		
位权限	U-0					R/W-000		

位号	助记符	功能描述
31	**MCTLEN**	波特率调制使能 (modulation control enable) 0：关闭波特率调制 1：使能波特率调制
30:29	**-**	未实现：读为0
28:16	**MCTL**	LPUART 每个 bit 的位宽调制控制信号
15:3	**-**	未实现：读为0

位号	助记符	功能描述
2:0	**BAUD**	波特率控制（bps） 000：9600 001：4800 010：2400 011：1200 100：600 101/110/111：300

22.7.5 LPUARTx 接收数据寄存器（LPUARTx _RXBUF）

名称	LPUARTx _RXBUF(x=0,1,2)							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							RXBUF[8]
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:9	**-**	**未实现：读为0**
8:0	**RXBUF**	接收数据缓存寄存器(Receive Buffer)

22.7.6 LPUARTx 发送数据寄存器（LPUARTx _TXBUF）

名称	LPUARTx _TXBUF(x=0,1,2)							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							TXBUF[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:9	-	未实现：读为0
8:0	**TXBUF**	发送数据缓存寄存器(Transmit Buffer)

22.7.7 LPUARTx 数据匹配寄存器（LPUARTx _DMR）

名称	LPUARTx _DMR(x=0,1,2)							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							MATD[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MATD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:9	-	**未实现：读为0**
8:0	**MATD**	第一帧接收比较数据，如果 RXEV=10，当接收到的第一帧数据与 MATD 相同时，触发 RXEVF 中断，可以用于休眠模式下的数据接收唤醒。(Matched Data)

23 串行外设接口（SPI/I2S）

23.1 概述

串行外设接口（Serial Peripheral Interface，SPI）是外部设备通过 4 线交换数据的串行同步通讯手段。SPI 接口模块可配置为主设备或从设备，实现与外部的 SPI 通信。

I2S 同样是一个同步串行通信外设，能够以主机或从机模式进行半双工通信，可以支持飞利浦 I2S 标准，MSB-LSB- justified，或 PCM 标准。

本模块可以配置为 SPI 或 I2S 功能。

SPI 特性：

- 全双工4线串行同步收发（SCLK, MOSI, MISO, SSN）
- MISO和MOSI可交换引脚顺序
- 半双工4线串行同步收发（SCLK, SDATA, SSN, DCN），用于TFT屏驱动
- 4路独立通道
- 主从模式
- 可编程时钟极性和相位
- 可编程比特速率
- 可编程数据字长（1~32bits）
- 最大波特率为$F_{APBCLK}/2$
- 传输结束中断标志
- 写冲突错标志
- 主模式错误检测、保护和中断标志
- 支持DMA

I2S特性：

- 支持主机和从机模式，半双工通信
- 8bit 时钟预分频电路
- 支持 16/24/32bit 数据格式
- 音频通道数据帧长度为 16bit 或 32bit
- 可编程时钟极性
- 支持 I2S 协议：
  - ✓ Philips I2S 标准

- ✓ MSB-justified 标准
- ✓ LSB-justified 标准
- ✓ PCM 标准

- 支持 DMA
- 支持主机时钟输出（256×Fs，Fs 为音频采样频率）

23.2 SPI 功能描述

23.2.1 SPI 结构框图

下图为 SPI 模块的结构示意图。

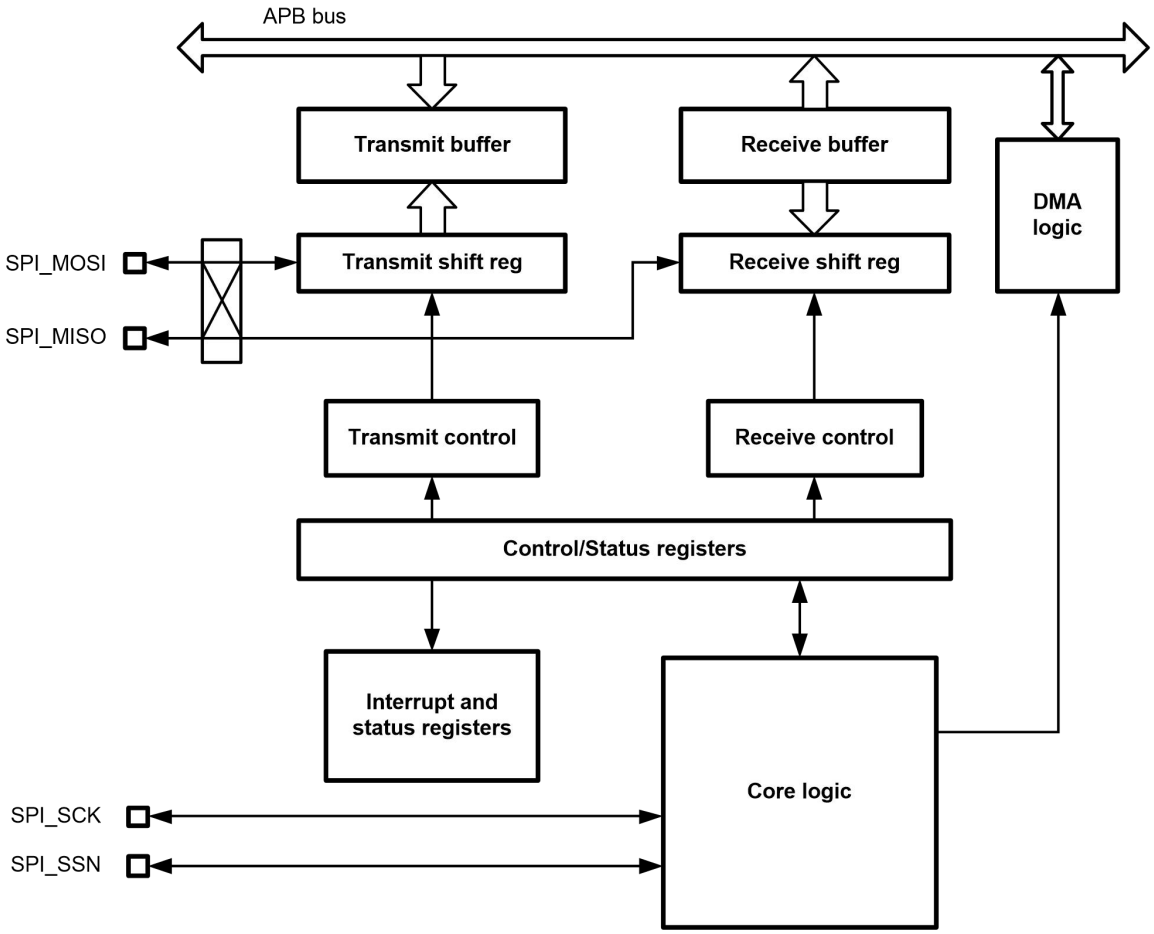


图 23-1 SPI 结构框图

23.2.2 SPI 接口时序

为了兼容不同的 SPI 外设，SPI 串行时钟的时序可以通过时钟相位选择位（CPHA）和时钟极性选择位(CPOL)设置产生 4 种不同组合。为保证数据正确传输，主从器件的时序配置必需一致。

当处于从器件模式或 SPI 系统使能位（SPE）位为 0 时，SPI 的 SCK 引脚无串行时钟输出。

23.2.2.1 CPHA=0

CPHA=0 时，SPI 模块在串行时钟的第一个跳变沿采样数据，即：

若 CPOL=1，总线 IDLE 时 SCK 停留在高电平，SPI 在串行时钟的下降沿采样数据，在串行时钟上升沿发送数据；

若 CPOL=0，总线 IDLE 时 SCK 停留在低电平，SPI 在串行时钟的上升沿采样数据，在串行时钟的下降沿发送数据。

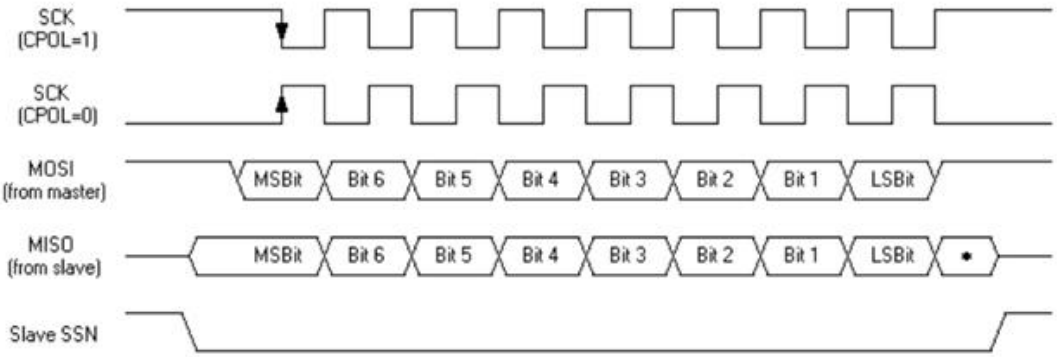


图 23-2 SPI 数据/时钟时序图（CPHA=0）

23.2.2.2 CPHA=1

CPHA=1 时，SPI 模块在串行时钟的第二个跳变沿采样数据，即：

若 CPOL=1，总线 IDLE 时 SCK 停留在高电平，在串行时钟的上升沿采样数据，在串行时钟的下降沿发送数据；

若 CPOL=0，总线 IDLE 时 SCK 停留在低电平，在串行时钟的下降沿采样数据，在串行时钟上升沿发送数据。

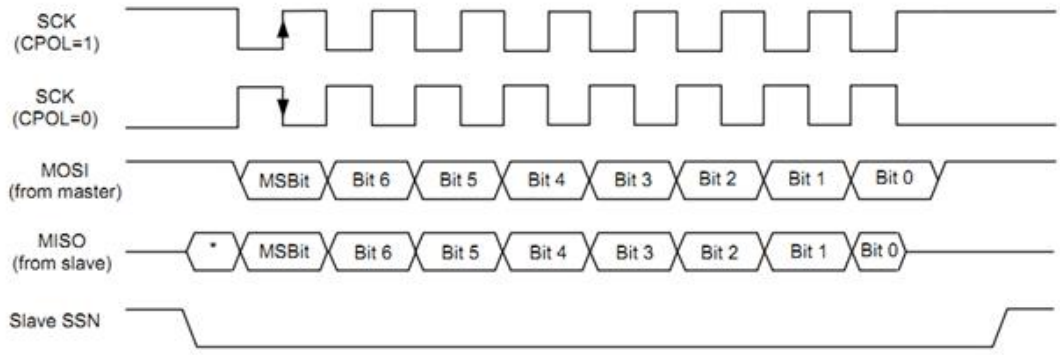


图 23-3 SPI 数据/时钟时序图（CPHA=1）

23.2.2.3 4 线半双工模式（主机）

4线半双工模式可以支持与点阵液晶或TFT屏的交互通信。在这种模式下，通过DCN信号的高低来区分当前发送的是命令帧还是数据帧。双向数据都通过SDATA（MOSI）引脚收发，由硬件自动完成数据方向切换。FM33FG0A的SPI仅支持4线半双工主机模式，不支持从机模式。

所有通信都由主机发起，主机首先发送命令帧，然后再进行数据帧传输。命令帧和数据帧通过DCN信号线区分。主机可以通过4线半双工接口向从机写入数据，或从从机读取数据。

4线半双工写操作

软件通过清零HD_RW寄存器，表示当前主机要发起写操作。

主机发起写操作前，首先发送写命令帧。当写命令帧发送完毕后，如果发送缓冲区为空，则硬件将拉高SSN并停止SCLK发送；如果发送缓冲区已经写入了新的数据，则硬件会连续发送后续的数据帧。

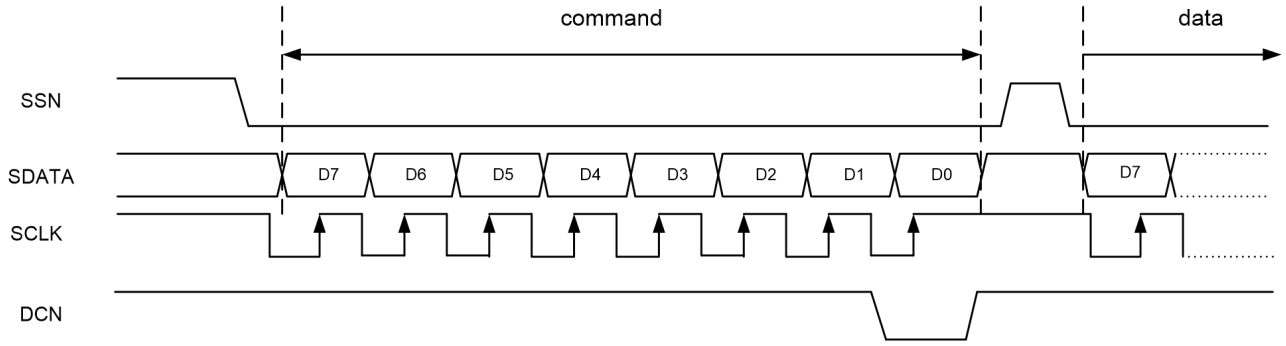


图 23-44 线半双工写操作

DCN在第8个时钟上升沿采样判决，如果为0，表示当前帧是命令帧。发送命令帧前，软件需要将DCN寄存器写0，命令帧发送完成后硬件自动将DCN寄存器置位。

4线半双工读操作

软件通过置位HD_RW寄存器，表示当前主机要发起读操作。

4线半双工读操作支持8位、24位和32位读取。主机发起读操作时，首先发送读命令帧。当读命令帧发送完毕后，可以根据寄存器配置发送1个dummy cycle，在dummy cycle期间，SCLK时钟正常发送，但是主机不驱动SDATA，也不接受SDATA输入。

完成命令帧和dummy cycle（可选）后，4线半双工SPI自动进入接收状态，SDATA信号改由从机驱动，主机收到的数据帧将被写入接收缓冲区。每个数据帧接收完成后，将置位RXBF中断标志寄存器。软件应及时读取接收缓冲区中的数据，如果接收缓冲区和接收移位寄存器都处于满状态，硬件会停止SCLK发送，暂停从从机读取数据，直到软件或DMA读取了接收缓冲区。

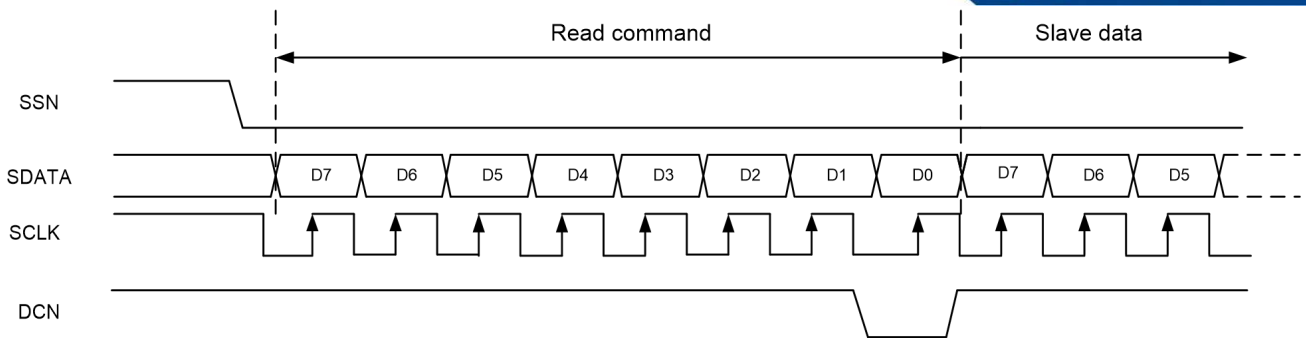


图 23-5 4 线半双工读操作（无 dummy cycle）

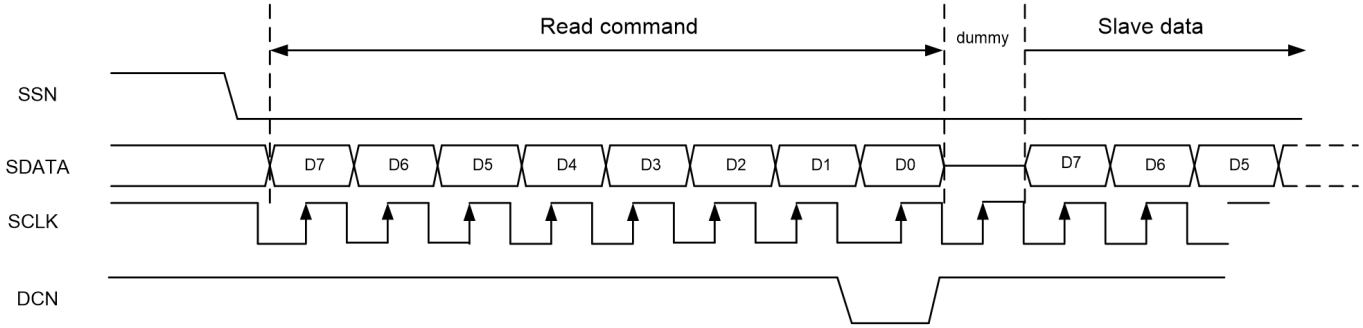


图 23-6 4 线半双工读操作（有 dummy cycle）

23.2.3 I/O 配置

主输出、从输入（MOSI）

主出从入（MOSI）引脚是主器件的输出和从器件的输入，用于主器件到从器件的串行数据传输。当 SPI 配置为主器件时，该引脚为输出，当 SPI 配置为从器件时，该引脚为输入。数据传输时 MSB 在前。

主输入、从输出（MISO）

主入从出（MISO）引脚是从器件的输出和主器件的输入，用于从器件到主器件的串行数据传输。当 SPI 配置为主器件时，该引脚为输入，当 SPI 配置为从器件时，该引脚为输出。数据传输时 MSB 在前。

串行时钟（SCK）

串行时钟（SCK）引脚是主器件的输出和从器件的输入，用于同步主器件和从器件之间在 MOSI 和 MISO 线上的串行数据传输。当 SPI 配置为主器件时，该引脚输出时钟，当 SPI 配置为从器件时，该引脚为输入。

从选择（SSN）

从选择（SSN）引脚用来控制从器件选中，如图 23-2 所示，当 SPI 配置为主器件时，SSN 引脚必

须接高电平，当 SPI 配置为从器件时，SSN 引脚必须接低电平。

SPI 主从器件的连接如下图所示：

主从器件的 MOSI、MISO 和 SCK 分别连在一起，主器件的 SSN 必须接高电平，从器件的 SSN 必须接低电平。主从器件通过 MOSI、MISO 连成一个环路，主器件输出时钟，数据传输时，主器件通过 MOSI 输出数据，从器件通过 MISO 输出数据。一字节数据传输完毕，主从器件将交换 8 位移位寄存器数值。

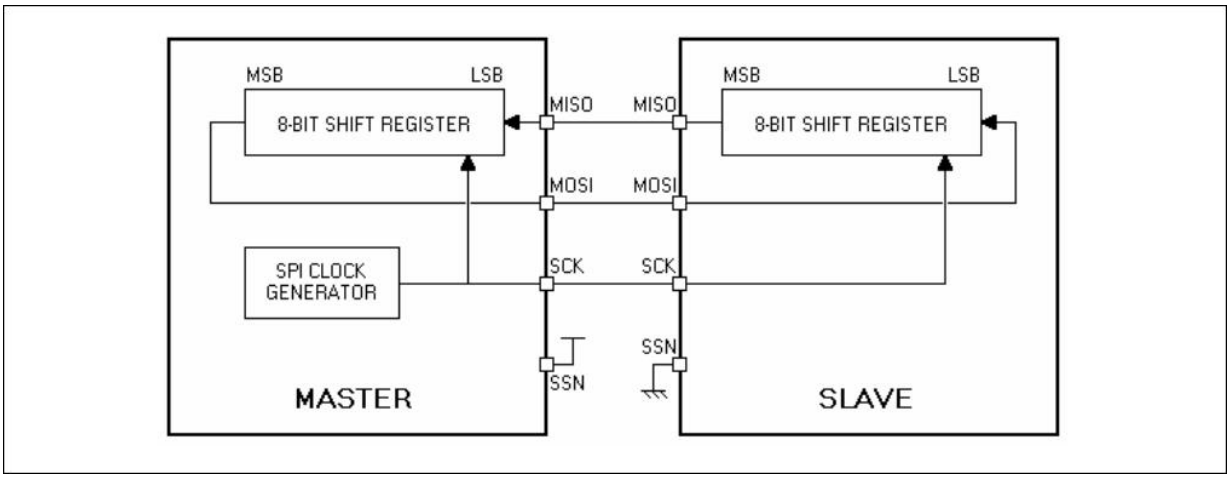


图 23-7 SPI Master/SPI Slave 互连

23.2.4 全双工数据通信

SPI模块默认为全双工通信，如果需要实现连续不间断的数据通信，软件需要确保TX BUFFER非空。即使软件只用SPI进行数据接收，由于SPI的全双工属性，软件仍需要对TX BUFFER进行写操作，此时写入的是无效数据，可根据MOSI无效状态配置写入全0或全F。

发送缓冲区

软件或DMA将待发送数据写入发送缓冲区（SPIxTXBUF寄存器），当发送开始时，硬件将数据从发送缓冲区拷贝到移位寄存器并开始发送。数据从发送缓冲区转移至移位寄存器后，发送缓存空标志（TXBE）被置位，表示可以向TXBUF写入新数据；如果TXIE寄存器置位，则产生中断。通过向TXBUF写入数据，可以清零TXBE寄存器。

如果在移位寄存器移位完成前，新的数据被写入发送缓冲区，则可以保证连续不断的数据发送。在TXBE为0的情况下写TXBUF，则会产生数据冲突，参见23.2.10数据冲突。

发送移位寄存器中的数据被完全发送出去后，发送移位寄存器空标志TXSE置位，当新的数据从发送缓冲区转移至发送移位寄存器后，TXSE标志自动清零。软件可以通过查询TXSE了解数据是否发送完毕。

接收缓冲区

当SPI完成一帧数据接收后，收到的数据将从移位寄存器拷贝到接收缓冲区（SPIxRXBUF寄存器），同时RXBF标志被置位，表示RXBUF中已有数据待处理。如果RXIE寄存器置位，则产生中断。通过读取RXBUF可以清零RXBF标志。

在RXBF没有置位的情况下读RXBUF，将返回上一次接收到的数据；如果应用没有及时处理RXBF，新的数据在RXBF置位的情况下完成接收，则产生数据冲突，参见23.2.10数据冲突。

BUSY标志

当SPI正在进行数据收发时，BUSY寄存器置位。此寄存器在某些场景下可以用来判断最后一帧数据是否传输完毕。比如TXBE只是表示数据已经进入移位发送，但是真正发送完成，需要等待BUSY标志清零。

如何启动SPI通信

主机模式下，建议遵循以下步骤启动SPI通信：

- 应用配置SPI模块
- 置位SPIEN
- 向TXBUF写入数据，SPI模块自动开始发送SCK并进行数据收发

从机模式下，建议应用在主机开始发送SCK之前完成配置和使能，并将第一帧待发送数据写入TXBUF，等待主机发送SCK开始通信。

如何结束SPI通信

主机模式下，建议遵循以下步骤结束SPI通信：

- 等待RXBF和TXBE标志置位，此时移位寄存器中还有最后一帧数据正在发送
- 查询BUSY标志，直到BUSY为0，最后一帧数据收发完成
- 关闭SPI模块，如果需要，读取最后一帧接收数据

从机模式下，应用可以在读取任意一帧数据后关闭SPI模块，关闭前已经被移入移位寄存器的数据将被忽略。

23.2.5 主机发送间隔

通过SPIx_CR1.WAIT寄存器可以配置主机发送间隔，0~3个SCK时钟。当WAIT=0时，由于字节之间没有间隔，即使SSNM=1，SSN也不会拉高。

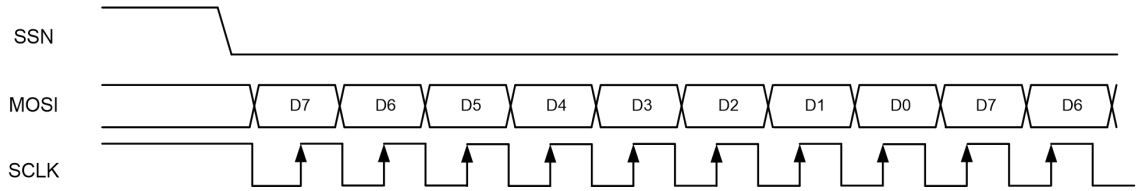


图 23-8 SPI Master 发送，WAIT=0

下图为WAIT=1的例子，SSNM=0，因此SSN不会拉高，但是SCK暂停一拍。

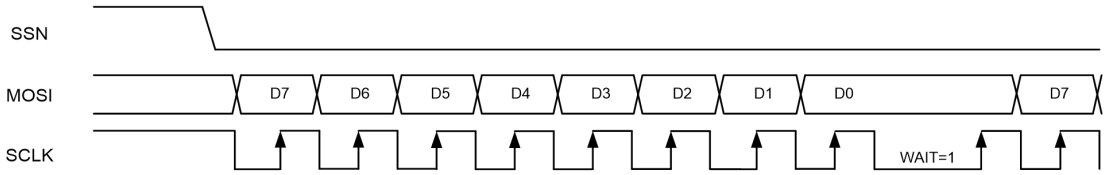


图 23-9 SPI Master 发送，WAIT=1

23.2.6 可编程数据字长

为了实现最大的协议灵活性，SPI支持1~32bit可编程数据字长，实际使用中常用的数据格式是8bit、16bit或32bit。

SPI发送缓冲和接收缓冲寄存器都是32bit的，当通信字长小于32bit时，数据在缓冲寄存器中以最低有效位保存；比如以8bit字长发送时，发送数据应写入TXBUF的bit7~0。数据缓冲寄存器不支持多帧缓冲，假设字长为8bit，不能一次性向TXBUF写入4帧数据，只能每次写入8bit，高于低8bit的数据将被丢弃。

23.2.7 TX-ONLY 模式

某些时候SPI通信是半双工的，在主机仅需进行发送的情况下，通过置位TXO寄存器进入TX-ONLY模式，此时MISO收到的数据不会被写入RX Buffer中，相应的也不会置位RXBF中断标志。

通过置位TXO_AC，可以实现TXO自动清零功能。在TX-ONLY模式下，如果TX buffer空（TXBE置位）并且发送移位寄存器空，则TXO寄存器自动清零，退出TX-ONLY状态。

23.2.8 RX-ONLY 模式

SPI主机仅需进行接收的情况下，通过置位RXO寄存器进入RX-ONLY模式，此时SPI模块无需软件对TX Buffer进行写操作，即可进行连续不断的数据接收，此时MOSI将保持IDLE电平，并且不会置位TXBE中断标志寄存器。

23.2.9 主机 SSN 控制

SPI模块主机支持硬件或软件控制SSN信号。

当SSNSEN寄存器清零时，SSN由硬件电路控制；如果SSNM寄存器置位，则SPI每发完一帧数据后，将拉高SSN，SSN高电平时间由WAIT寄存器配置（若干个SCK时钟周期）；

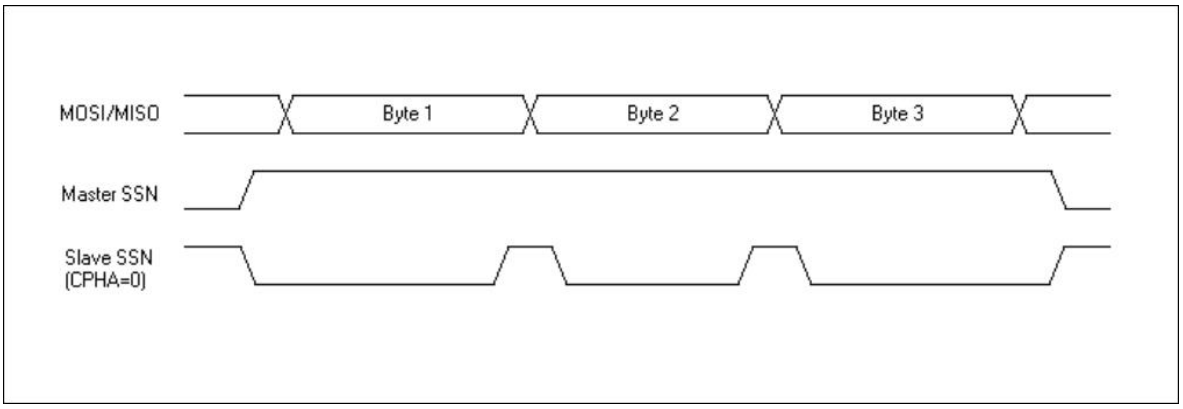


图 23-10 SPI SSN 时序图（SSNM=1，CPHA=0）

如果SSNM寄存器复位，则SPI每发完一帧数据后不会拉高SSN，而是直接进入下一帧数据发送。

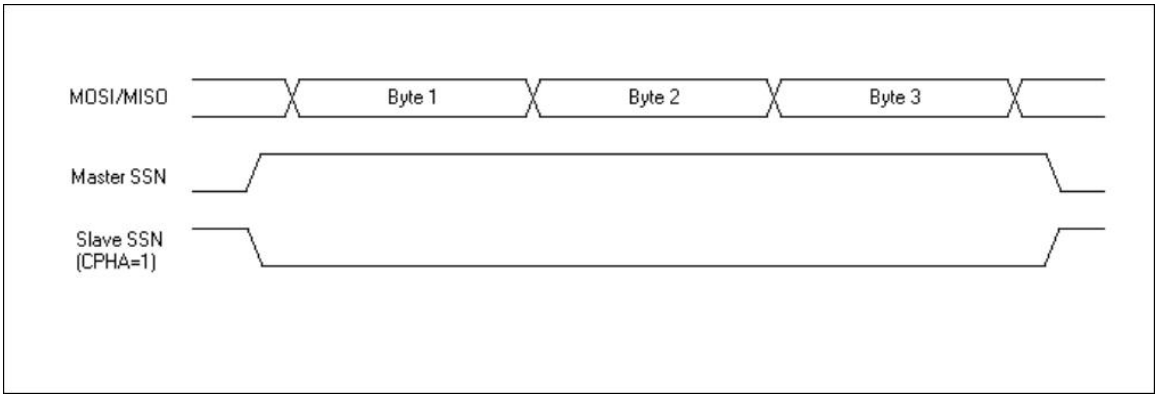


图 23-11 SPI SSN 时序图（SSNM=0）

当SSNSEN寄存器置位时，SSN直接由软件控制。软件通过写SPIxCR2.SSN寄存器位，可以直接操作SPI主机发送的SSN电平。

23.2.10 数据冲突

当 SPI 的 TX Buffer 数据尚未被读进移位寄存器，或者 SPI 的 RX Buffer 中的数据未被软件或 DMA 读取时，对 TX Buffer 或 RX Buffer 的写操作会产生对应的冲突错误，TXCOL/RXCOL 位会置起，产生中断。导致冲突的写入数据将被忽略。数据冲突错误在主从模式下都会产生。

对 TX Buffer 的写操作，由芯片内部的 Master 模块发起，包括 CPU、DMA 等等。对 RX Buffer 的写操作，则由外部 SPI 器件发起。

当数据冲突发生时，TX Buffer 和 RX Buffer 内原有数据不会被刷新，新写入的数据丢失。

23.2.11 使用 DMA 进行 SPI 收发

当 SPI 模块被使能后，SPI 模块在发送缓冲区空和接收缓冲区满时都会自动产生相应的 DMA 请求。应用软件需要事先配置 DMA 通道连接，将特定通道指向 SPI 外设，设置 RAM 访问的指针地址，并使能 DMA 通道。此后 DMA 会自动响应 SPI 请求，并完成 RAM 和 SPI 之间的数据搬运。

注意：如果使用 DMA 进行收发全双工通信，软件应先使能 DMA 发送通道，再使能 DMA 的接收通道；反之可能会导致 SPI 额外发送一个字节的 dummy 数据。

使用 DMA 进行 SPI 接收

- 将 DMA 通道 3 或 5 配置为 SPI_RX
- 设置 RAM 指针地址、地址递增递减、通道优先级、传输长度和中断设置等
- 使能对应 DMA 通道
- 配置 SPI 模块参数
- 使能 SPI 模块，等待数据接收
- 收到数据后 SPI 自动产生 DMA 请求
- DMA 响应请求，读取 SPI 接收缓存寄存器，写入指定 RAM 地址
- 当指定长度的 DMA 传输结束后，DMA 将忽略后续请求并产生传输完成中断，软件应处理中断并关闭 SPI
- 如果关闭 SPI 前又有数据被接收，软件可以通过写 RXBFC 清除 RXBUF

使用 DMA 进行 SPI 发送

DMA发送过程与上述接收过程类似，主要差别是，当指定长度的DMA传输结束后，软件不能立即关闭SPI，因为此时最后一帧数据还在移位发送中，因此软件需要查询BUSY标志直到移位发送结束，再关闭SPI模块。

数据帧长度与 RAM 数据组织方式

SPI 传输帧长度可以配置为 8、16、24、32bit。

当数据帧长度为 8bit 时，DMA 每次搬运 1byte，4 次搬运填满 RAM 一个地址，字内采用小端存储：

RAM word: { data3，data2，data1，data0 }

当数据帧长度为 16bit 时，DMA 每次搬运 2bytes，2 次搬运填满 RAM 一个地址，字内采用小端存储：

RAM word: { data1，data0 }

当数据帧长度为 24bit 时，DMA 每次搬运 1 word，1 次搬运填满 RAM 一个地址，但是有效数据仅占用 RAM 字内低 24bit：

RAM word: { 8'h0，data0 }

当数据帧长度为 32bit 时，DMA 每次搬运 1 word，1 次搬运填满 RAM 一个地址：

RAM word: { data0 }

23.3 I2S 功能描述

23.3.1 I2S 结构框图

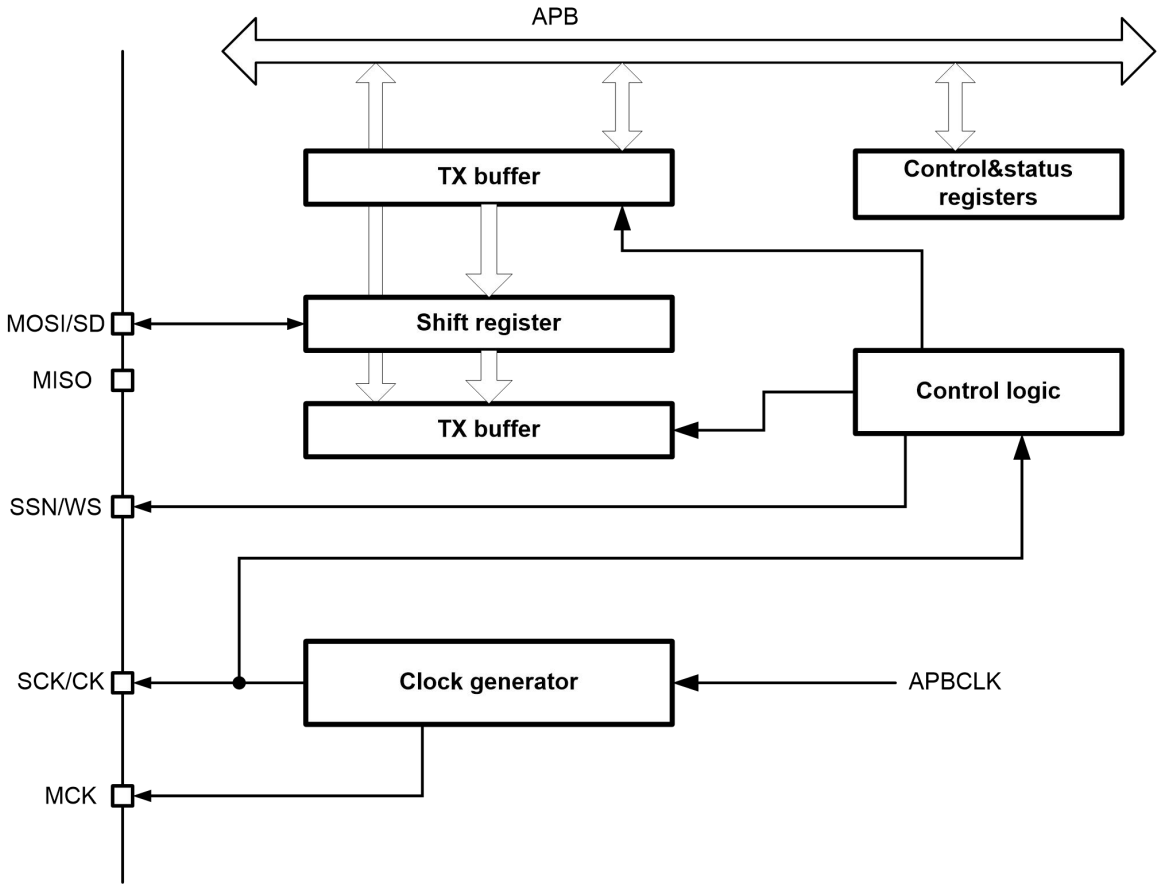


图 23-12I2S 结构框图

23.3.2 引脚定义

I2S与SPI模块共享3个引脚：

- SD：双向串行数据，与MOSI复用
- WS：声道选择，与SSN复用
- CK：串行时钟，与SCK复用

除了以上引脚，I2S还需要一个独立的主机时钟输出引脚：

- MCK：Master时钟输出，仅在主机模式下有效，输出频率为256*Fs，其中Fs为音频采样频率

23.3.3 协议支持

I2S 协议通过时分复用进行左右两个通道的数据收发。然而，I2S 模块只有 1 个 16bit 数据寄存器，软件需要通过 CHSIDE 寄存器来区分数据所属的声道。在通信过程中，左声道数据总是先于右声道传输。在 PCM 模式下，CHSIDE 寄存器没有意义。

I2S 模块支持如下数据帧格式：

- 16bit 帧中传输 16bit 数据
- 32bit 帧中传输 16、24、32bit 数据

当使用 32bit 帧传输 16 位数据时，仅高 16 位有效，低 16 位被硬件固定为全 0。

24bit 或 32bit 数据帧传输需要 CPU 或 DMA 的 2 次读写操作（每次读写 16bit）。当使用 32bit 帧传输 24 位数据时，低 8 位被固定成全 0。

针对所有通信协议标准，都是最高位先发（MSB first）。

I2S 模块支持 4 种协议标准，通过 I2SSTD 寄存器配置当前格式。

23.3.3.1 Philips standard

Philips 标准使用 WS 信号表示当前发送的是哪个声道。WS 在数据帧 1st bit 之前一个时钟周期被拉低，表示左声道开始传输；左声道传输完成后，开始右声道传输。

典型数据传输波形：

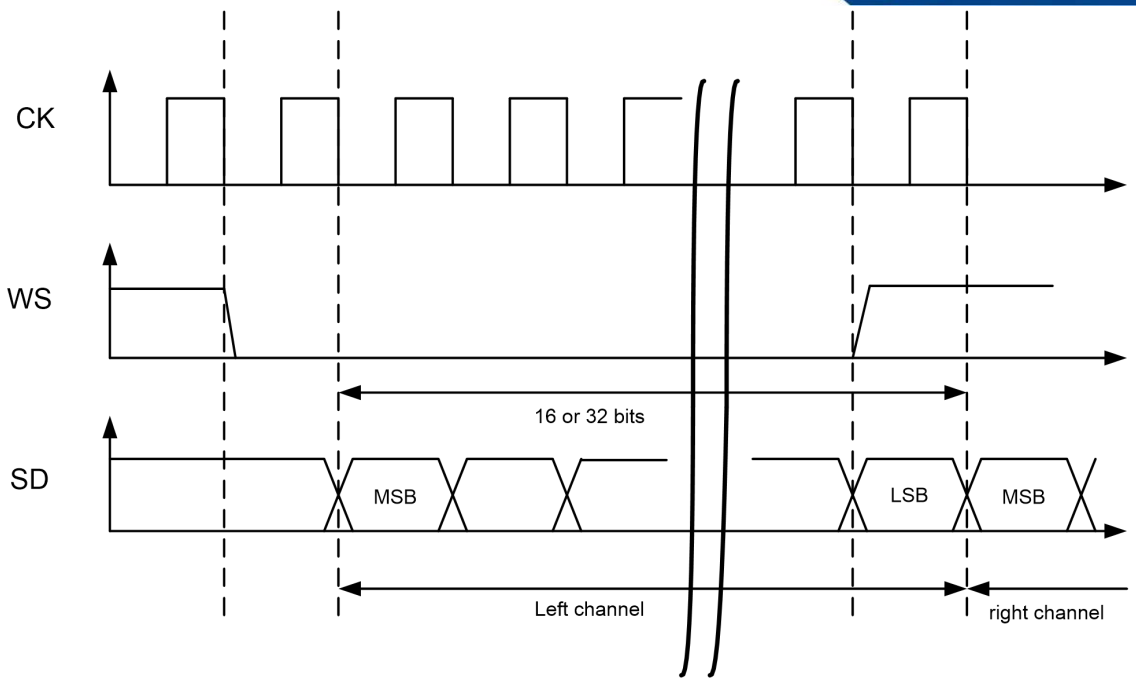


图 23- 13Philips 标准时序

23.3.3.2 MSB-justified standard

此标准与 Philips 的差异是，WS 信号与 1st bit 同步产生，并且 WS 高电平代表左声道。

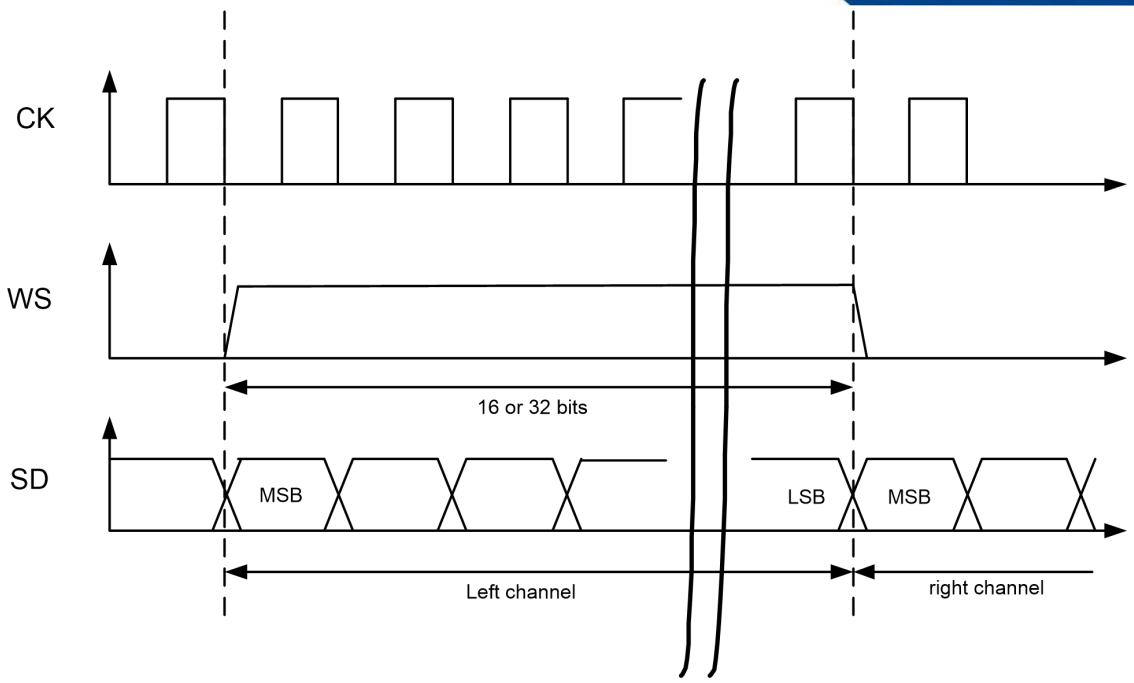


图 23-14 32/16-bits MSB-justified 时序

在 32 位数据帧中传输 24 位数据时，MSB justified 格式将有效数据先发，最后补 0：

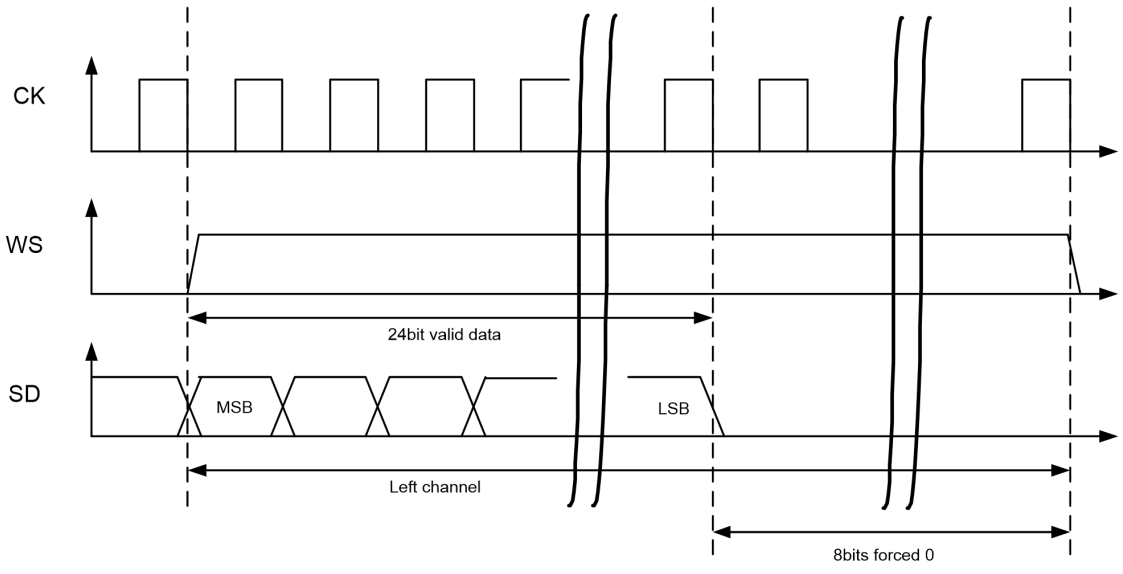


图 23-15 24-bits MSB-justified 时序

在 32 位数据帧中传输 16 位数据时，MSB justified 格式同样是将有效数据先发，最后补 0。

23.3.3.3 LSB-justified standard

LSB justified 标准与 MSB justified 类似，只是在有效数据长度不到数据帧长的情况下，将有效数据靠右对齐，先发填充的 0。

在 32 位数据帧中传输 24 位数据时的波形：

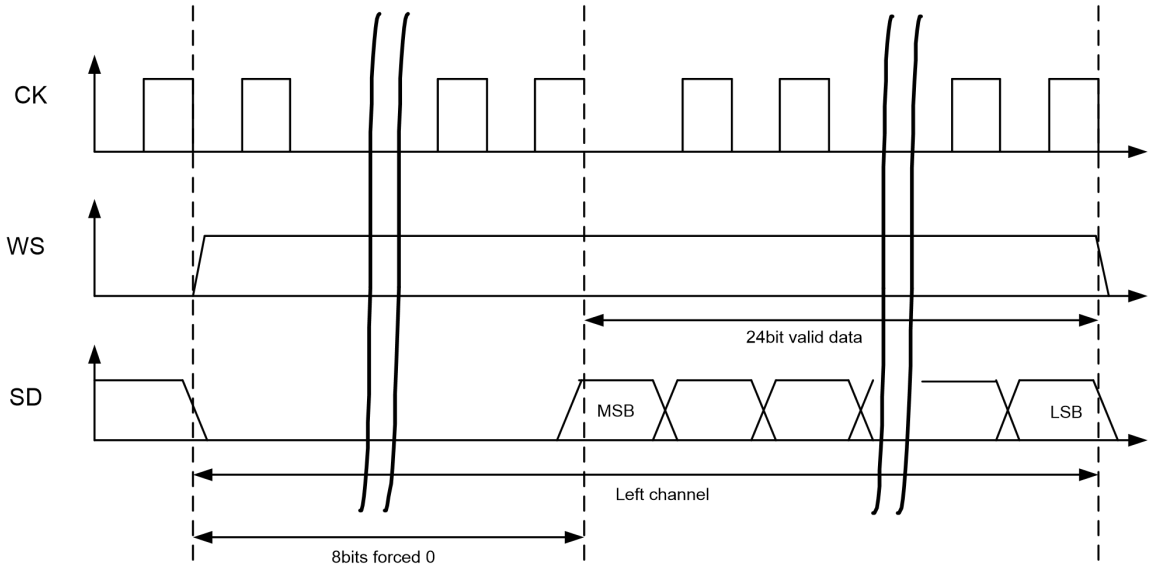


图 23-16LSB-justified 时序

23.3.3.4 PCM standard

针对 PCM 标准，无需使用声道信息。PCM 标准下支持长帧和短帧模式，通过 PCMSYNC 寄存器配置。WS 信号在 PCM 模式下用于数据帧起始同步，不代表左右声道识别。

长帧格式下，WS 信号固定置位 13bit 长度；短帧格式下，WS 信号只置位 1bit 长度。

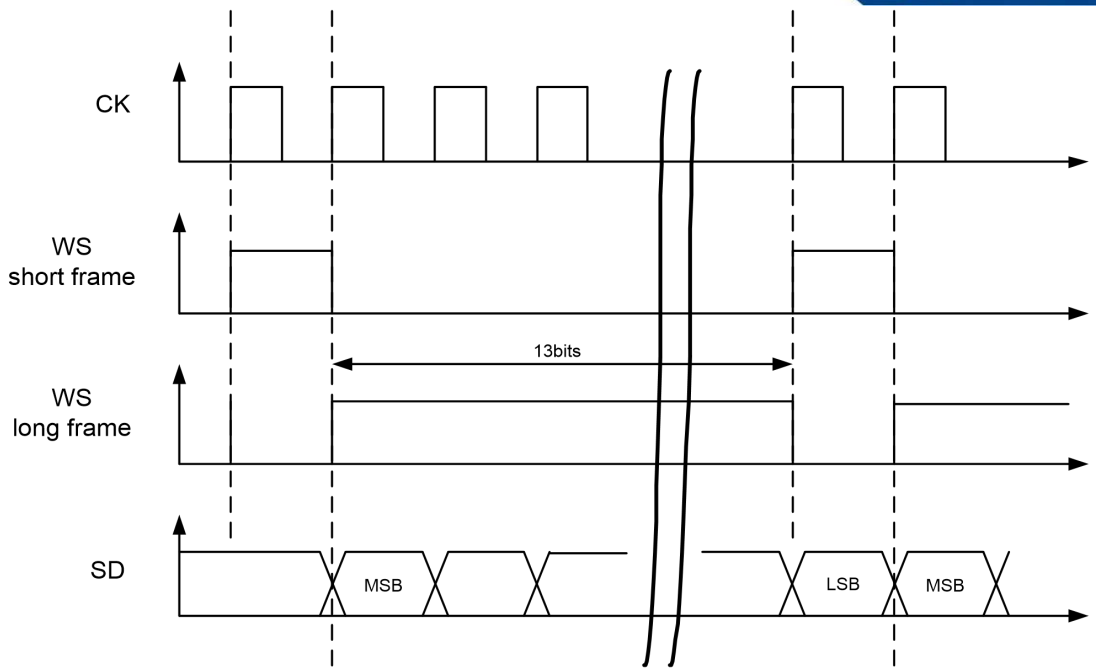


图 23-17PCM 标准时序

23.3.4 时钟产生

I2S 数据码率由下式决定：

I2S 码率 = 每声道数据帧位数×声道数×音频采样频率

假设采用 16 位音频帧格式，左右声道，音频采样率 F_S，则码率计算为：

I2S 码率 = 16×2×F_S

假设采用 32 位音频帧格式，左右声道，，音频采样率 F_S，则码率计算为：

I2S 码率 = 32×2×F_S

音频采样的频率包括：192Khz、96Khz、48Khz、44.1Khz、32Khz、22.05Khz、16Khz、11.025Khz、8Khz 等。为了获得以上目标频率，按照以下公式计算分频寄存器的值（CLKDIV 和 ODD 都是分频寄存器）：

如果输出 MCK：F_S = APBCLK / [(256)×((2×CLKDIV) + ODD)]，

如果不输出 MCK：F_S = APBCLK / [32×(CHLEN+1)×((2×CLKDIV) + ODD)]，

CHLEN=0：声道帧长 16bit

CHLEN=1：声道帧长 32bit

CLKDIV 寄存器定义了 50%占空比下时钟 CK 的高低电平宽度，ODD 寄存器用于调整高电平宽度，从而实现理论上的小数分频效果。

I2S 主机模式下可以通过 MCKOE 寄存器选择输出或者不输出 MCK，MCK 频率是 256×Fs

假设 APBCLK 为 48MHz，下表罗列了针对不同 Fs 的参考配置和实际采样频率误差。

APBCLK (MHz)	CHLEN	CLKDIV	ODD	MCK	Target Fs (Hz)	Actual Fs	Error
48	16	8	0	No	96000	93750	2.3438%
48	32	4	0	No	96000	93750	2.3438%
48	16	15	1	No	48000	48387.0968	0.8065%
48	32	8	0	No	48000	46875	2.3438%
48	16	17	0	No	44100	44117.6471	0.0400%
48	32	8	1	No	44100	44117.6471	0.0400%
48	16	23	1	No	32000	31914.8936	0.2660%
48	32	11	1	No	32000	32608.6957	1.9022%
48	16	34	0	No	22050	22058.8235	0.0400%
48	32	17	0	No	22050	22058.8235	0.0400%
48	16	47	0	No	16000	15957.4468	0.2660%
48	32	23	1	No	16000	15957.4468	0.2660%
48	16	68	0	No	11025	11029.4118	0.0400%
48	32	34	0	No	11025	11029.4118	0.0400%
48	16	94	0	No	8000	7978.7234	0.2660%
48	32	47	0	No	8000	7978.7234	0.2660%
48	16	2	0	Yes	48000	46875	2.3438%
48	32	2	0	Yes	48000	46875	2.3438%
48	16	2	0	Yes	44100	46875	6.2925%
48	32	2	0	Yes	44100	46875	6.2925%
48	16	3	0	Yes	32000	31250	2.3438%
48	32	3	0	Yes	32000	31250	2.3438%
48	16	4	1	Yes	22050	20833.3333	5.5178%
48	32	4	1	Yes	22050	20833.3333	5.5178%
48	16	6	0	Yes	16000	15625	2.3438%
48	32	6	0	Yes	16000	15625	2.3438%
48	16	8	1	Yes	11025	11029.4118	0.0400%
48	32	8	1	Yes	11025	11029.4118	0.0400%
48	16	11	1	Yes	8000	8152.17391	1.9022%
48	32	11	1	Yes	8000	8152.17391	1.9022%

表 23-1I2S 典型采样频率计算表

23.3.5 I2S 主机功能

主机模式下，模块负责产生 CK 和 WS 信号，MCK 可以配置为输出或不输出。

23.3.5.1 操作流程

I2S 模块通用操作流程如下：

- 配置 CLKDIV 和 ODD 寄存器，获得通信所需的串行时钟频率
- 通过 CKPOL 寄存器配置总线时钟在静态下的电平
- 配置 I2SCFG 寄存器，设置模块的工作模式、协议格式、帧长度等
- 使能需要的中断或 DMA
- 置位 I2SEN 使能模块通信

23.3.5.2 发送流程

在 I2S 使能的情况下，软件或 DMA 写 TX BUFFER 将启动总线发送。TX Buffer 的宽度是 16bit，因此 24 或 32 位音频数据发送需要 2 次写操作。数据发送时，硬件会自动根据当前发送的是左声道还是右声道数据，控制 WS 输出电平，并且置位或复位 CHSIDE 寄存器表征当前正在发送的声道。

当 I2S 总线空闲时，写入 TX buffer 的数据会被立即载入移位寄存器进行发送，同时 TXE 中断标志寄存器置位，表示发送缓冲区空，软件可以在 TXE 置位后写入下一笔数据，或者在使能 DMA 的情况下由 DMA 写入。

为了保证连续不间断的数据发送，软件必须保证在当前移位寄存器发送完成前，将后续 16 位数据填入 TX buffer。

软件在清零 I2SEN 寄存器关闭 I2S 模块前，必须等待 TXE 置位并且 BUSY 清零。

23.3.5.3 接收流程

无论配置的数据帧长是多少，实际数据接收缓冲区宽度是固定的 16bit。一旦移位寄存器完成 16 位接收，数据将自动载入 RX buffer，同时置位 RXNE 中断标志寄存器，表示接收缓冲区非空。软件可以在 RXNE 置位后读取接收缓冲寄存器，或者在使能 DMA 的情况下由 DMA 读取。

软件应确保在新的数据帧接收完成前，读取 RX buffer 中的数据。如果新的数据到来时 RX buffer 非空，则 OVR 错误标志寄存器将被置位。

23.3.6 I2S 从机功能

23.3.6.1 操作流程

在从机模式下，I2S 的 CK 和 WS 信号来自于外部主机，通信码率由主机决定，因此用户无需配置时钟分频寄存器。软件需要将 I2S 从机配置为发送或接收模式。

- 配置 I2S 为从机模式，配置协议格式、帧长度，选择 I2S 为发送或接收模式。
- 根据需要使能中断或 DMA
- 置位 I2SEN 寄存器

23.3.6.2 发送流程

当 I2S 从机配置为发送模式，主机在总线上发送时钟和 WS 时，从机自动开始发送数据。软件必须在通信开始前将待发数据写入 TX buffer。如果主机发送时钟和 WS 时，TX buffer 和移位寄存器都空，则 I2S 模块将置位 UDR 错误标志，表示数据下溢出。

CHSIDE 寄存器用于表征当前通信的声道，此寄存器在从机模式下根据主机发送的 WS 信号自动更新。

当第一个 I2S 时钟到来时，TX buffer 中的数据被载入移位寄存器开始发送，此时 TXE 中断标志寄存器置位，软件或 DMA 可以将下一笔发送数据写入 TX buffer。

为了保证连续不间断的数据发送，软件必须保证在当前移位寄存器发送完成前，将后续 16 位数据填入 TX buffer。

软件在清零 I2SEN 寄存器关闭 I2S 模块前，必须等待 TXE 置位并且 BUSY 清零。

23.3.6.3 接收流程

当 I2S 从机配置为发送模式，主机在总线上发送时钟和 WS 时，从机自动开始接收数据。无论配置的数据帧长是多少，实际数据接收缓冲区宽度是固定的 16bit。一旦移位寄存器完成 16 位接收，数据将自动载入 RX buffer，同时置位 RXNE 中断标志寄存器，表示接收缓冲区非空。软件可以在 RXNE 置位后读取接收缓冲寄存器，或者在使能 DMA 的情况下由 DMA 读取。

CHSIDE 寄存器用于表征当前通信的声道，此寄存器在从机模式下根据主机发送的 WS 信号自动更新。

软件应确保在新的数据帧接收完成前，读取 RX buffer 中的数据。如果新的数据到来时 RX buffer 非空，则 OVR 错误标志寄存器将被置位。

23.3.7 帧格式错误

I2S 从机模式下，如果主机发送的 WS 信号变化不符合协议规范，则 I2S 模块将置位 SERR 错误中断标志信号，表示帧格式错误。

23.3.8 DMA 支持

I2S 模块支持通过 DMA 进行数据收发。当 I2S 模块被使能后，I2S 模块在发送缓冲区空和接收缓冲区满时都会自动产生相应的 DMA 请求。应用软件需要事先配置 DMA 通道连接，将特定通道指向 I2S 外设，设置 RAM 访问的指针地址，并使能 DMA 通道。此后 DMA 会自动响应 I2S 请求，并完成 RAM 和 I2S 之间的数据搬运。

23.4 寄存器

SPI0模块基地址：0x40010400

SPI1模块基地址：0x40010800

SPI2模块基地址：0x40014800

SPI3模块基地址：0x4001A800

offset 地址	名称	符号
SPIx 寄存器 (x=0,1,2,3)		
0x00	SPIx 控制寄存器 1 （SPIx Control Register1）	SPIx_CR1
0x04	SPIx 控制寄存器 2 （SPIx Control Register2）	SPIx_CR2
0x08	SPIx 控制寄存器 3 （SPIx Control Register3）	SPIx_CR3
0x0C	SPIx 中断使能寄存器 （SPIx Interrupt Enable Register）	SPIx_IER
0x10	SPIx 中断状态寄存器 （SPIx Status Register）	SPIx_ISR
0x14	SPIx 发送数据缓冲寄存器 （SPIx Transmit Buffer）	SPIx_TXBUF
0x18	SPIx 接收数据缓冲寄存器 （SPIx Receive Buffer）	SPIx_RXBUF
0x20	I2S 配置寄存器 （SPIx I2S Control Register）	SPIx_I2SCR
0x24	I2S 预分频寄存器 （SPIx I2S Prescaler Register）	SPIx_I2SPR

23.4.1 SPI 控制寄存器 1（SPIx_CR1）

名称	SPIx_CR1 (x=0,1,2,3)							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				IOSWAP	MSPA	SSPA	MM
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WAIT		BAUD			LSBF	CPOL	CPHA
位权限	R/W-00		R/W-000			R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:11	--	RFU：未实现，读为 0
11	**IOSWAP**	MOSI 和 MISO 引脚交换 (IO swapping)

Bit	助记符	功能描述
		0：默认引脚顺序 1：交换引脚顺序
10	**MSPA**	Master Sampling Position Adjustment，Master 对 MISO 信号的采样位置调整，用于高速通信时补偿 PCB 走线延迟 1：采样点延迟半个 SCK 周期 0：不调整
9	**SSPA**	Slave Sending Position Adjustment，Slave MISO 发送位置调整 1：提前半个 SCK 周期发送 0：不调整
8	**MM**	Master/Slave 模式选择。 1：Master 模式 0：Slave 模式
7:6	**WAIT**	Master 模式下，每发送完一帧后加入 (WAIT)个 SCK cycle 等待时间，再传输下一帧的数据。如果 SSN 由硬件控制，并且 SSNM=1，则硬件会自动拉高 SSN。 当 WAIT=0 时，SPI 主机连续发送，字节之间没有间隔
5:3	**BAUD**	Master 模式波特率配置位: 000： $f_{APBCLK}/2$ 001： $f_{APBCLK}/4$ 010： $f_{APBCLK}/8$ 011： $f_{APBCLK}/16$ 100： $f_{APBCLK}/32$ 101： $f_{APBCLK}/64$ 110： $f_{APBCLK}/128$ 111： $f_{APBCLK}/256$ 当通信正在进行的时候，不能修改这些位。
2	**LSBF**	帧格式 (LSB First) 0：先发送 MSB 1：先发送 LSB 注：当通信在进行时不能改变该位的值。
1	**CPHOL**	时钟极性选择 (Clock Polarity) 1：串行时钟停止在高电平 0：串行时钟停止在低电平 注：当通信在进行时不能改变该位的值 注：当 SSN 为低时不能改变该位的值
0	**CPHA**	时钟相位选择 (Clock Phase) 1：第二个时钟边沿是第一个捕捉边沿 0：第一个时钟边沿是第一个捕捉边沿 注：当通信在进行时不能改变该位的值。

23.4.2 SPI 控制寄存器 2（SPIx_CR2）

名称	SPIx_CR2 (x=0,1,2,3)							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DUMMY_EN	DLEN2			RXO	DLEN		HALFDUPLEX
位权限	R/W-0	R/W-000			R/W-0	R/W-00		R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	HD_RW	CMD8b	SSNM	TXO_AC	TXO	SSN	SSNSEN	SPIEN
位权限	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-0

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	**DUMMY_EN**	4 线半双工协议下是否在读操作中插入 dummy cycle (Dummy cycle Enable) 0：不插入 dummy cycle 1：在读命令之后插入一个 dummy cycle
14:12	**DLEN2**	通信数据字长扩展配置寄存器，与 DLEN 组合起来形成最终配置（配置方式考虑既有产品的兼容性） {DLEN2, DLEN} 00000: 8bit 00001: 16bit 00010: 24bit 00011: 32bit 00100: 7bit 00101: 15bit 00110: 23bit 00111: 31bit 01000: 6bit 01001: 14bit 01010: 22bit 01011: 30bit 01100: 5bit 01101: 13bit 01110: 21bit 01111: 29bit 10000: 4bit 10001: 12bit 10010: 20bit 10011: 28bit 10100: 3bit 10101: 11bit 10110: 19bit 10111: 27bit 11000: 2bit 11001: 10bit 11010: 18bit 11011: 26bit 11100: 1bit 11101: 9bit 11110: 17bit 11111: 25bit
11	**RXO**	RXONLY 控制位，此寄存器置位时，SPI 可以连续接收，无需软件写 TXBUF (Receive Only mode) 1：启动 Master 的单接收模式 0：关闭单接收模式（收发全双工）

Bit	助记符	功能描述
10:9	**DLEN**	通信数据字长配置 (Data Length) 参见 DLEN2
8	**HALFDUPLEX**	通信模式选择 (Half-Duplex mode) 0：标准 SPI 模式，4 线全双工 1：DCN 模式，4 线半双工
7	**HD_RW**	半双工模式下主机读写操作配置 (Read/Write config for Half-Duplex mode) 0：4 线半双工协议下主机写入从机 1：4 线半双工协议下主机读取从机
6	**CMD8b**	半双工模式下定义 command 帧长度 (Command 8 bits) 1：command 帧固定为 8bit 0：command 帧长度由 DLEN 定义
5	**SSNM**	Master 模式下 SSN 控制模式选择 (SSN mode) 1：每发送完一帧后 Master 拉高 SSN，维持高电平时间由 WAIT 寄存器控制 0：每发送完一帧后 Master 保持 SSN 为低
4	**TXO_AC**	TXONLY 硬件自动清空的使能 (TXONLY auto-clear enable) 1：TXONLY 硬件自动清零有效，软件使能 TXO 后，等待发送完毕后，硬件清零 0：关闭 TXONLY 硬件自动清零
3	**TXO**	TXONLY 控制位 (Transmit Only mode enable) 1：启动 Master 的单发送模式 0：关闭单发送模式（收发全双工）
2	**SSN**	Master 模式下，如果 SSNSEN 为 1，软件可以通过此位控制 SSN 输出电平 1：SSN 输出高电平 0：SSN 输出低电平
1	**SSNSEN**	Master 模式下，软件控制 SSN 使能 (SSN Software Enable) 1：Master 模式下 SSN 输出由软件控制 0：Master 模式下 SSN 输出由硬件自动控制
0	**SPIEN**	SPI 使能(SPI enable) 1：使能 SPI 0：关闭 SPI，清空发送接收缓存

23.4.3 SPI 控制寄存器 3（SPIx_CR3）

名称	SPIx_CR3 (x=0,1,2,3)							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	TXBFC	RXBFC	MERRC	SERRC
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3	TXBFC	Transmit Buffer Clear，软件写 1 清除发送缓存，写 0 无效
2	RXBFC	Receive Buffer Clear，软件写 1 清除发送缓存，写 0 无效
1	MERRC	Master Error Clear，软件写 1 清除 HSPISTA.MERR 寄存器
0	SERRC	Slave Error Clear，软件写 1 清除 HSPISTA.SERR 寄存器

23.4.4 SPI/I2S 中断控制寄存器（SPIx_IER）

名称	SPIx_IER (x=0,1,2,3)							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		SSNRIE	SSNFIE	TXSEIE	ERRIE	TXBEIE	RXIE
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5	SSNRIE	从机 SSN 上升沿中断使能
4	SSNFIE	从机 SSN 下降沿中断使能
3	TXSEIE	发送移位寄存器空中断使能(Transmit shift register empty interrupt enable)
2	ERRIE	SPI 错误中断使能 (Error Interrupt Enable)
1	TXBEIE	发送缓冲空中断使能 (Transmit buffer empty Interrupt Enable)
0	RXIE	接收完成中断使能 (Receive Interrupt Enable)

23.4.5 SPI/I2S 中断标志寄存器（SPIx_ISR）

名称	SPIx_ISR (x=0,1,2,3)							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			DCN_TX	-	RXCOL	TXCOL	BUSY
位权限	U-0			R/W/Dy-1	U-0	R/W/Dy-0	R/W/Dy-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CHSIDE	MERR	SERR	SSNR	SSNF	TXSE	TXBE	RXBF
位权限	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-1	R-0

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12	**DCN_TX**	半双工模式下 (HALFDUPLEX=1)，配置在每个数据帧的最后 bit 发送的 DCN 信号电平 (Data/Command transmit config) 0：DCN=0，表示命令帧 1：DCN=1，表示数据帧 软件应在发送前设置 DCN_TX 寄存器，如果 DCN_TX=0，硬件在完成一帧发送后，自动将 DCN_TX 置 1，即默认只会发送一个命令帧，后续都是数据帧。
11	--	RFU：未实现，读为 0
10	**RXCOL**	接收缓存溢出，软件写 1 清零 (Receive Collision flag,write 1 to flag)
9	**TXCOL**	发送缓存溢出，软件写 1 清零 (Transmit Collision flag,write 1 to clear)
8	**BUSY**	SPI/I2S 空闲标志，只读(busy flag) 1：SPI/I2S 传输进行中 0：SPI/I2S 传输空闲 *注意：此标志在 SPI/I2S 波特率时钟下置位，当软件使能 SPI/I2S 发送后，需等待波特率时钟到来才会置位*
7	**CHSIDE**	声道标志寄存器，仅 I2S 模式有效，用于表示当前正在传输的数据属于哪个声道 0：左声道 1：右声道
6	**MERR**	Master Error 标志(Master Error flag)，硬件置位，软件写 1 清零 当 Master 下传输未满 8 位 SSN 就被拉高时，MERR 置位
5	**SERR**	Slave Error 标志(Slave Error flag)，硬件置位，软件写 1 清零 SPI 模式：当 Slave 下传输未满 8 位 SSN 就被拉高时，SERR 置位= I2S 模式：当主机在错误的位置改变 WS 信号电平时置位
4	**SSNR**	从机接收的 SSN 信号上升沿标志位 (slave SSN rising flag)，硬件置位，软件写 1 清零 1：从机接收到 SSN 上升沿 0：从机没有接收到 SSN 上升沿 此标志仅 SPI 从机模式下有效
3	**SSNF**	从机接收的 SSN 信号下降沿标志位 (slave SSN falling flag)，硬件置位，软件写 1 清零 1：从机接收到 SSN 下降沿 0：从机没有接收到 SSN 下降沿 此标志仅 SPI 从机模式下有效
2	**TXSE**	TX Shift Register Empty 标志位 (TX Shift Register Empty flag)

Bit	助记符	功能描述
		1：发送移位寄存器空 0：发送移位寄存器非空
1	**TXBE**	TX Buffer Empty 标志位(TX Buffer Empty flag) 1：发送缓存空，软件写 TXBUF 清零 0：发送缓存满
0	**RXBF**	RX Buffer Full 标志位(RX Buffer Full flag) 1：接收缓存满，软件读 RXBUF 清零 0：接收缓存空

23.4.6 SPI/I2S 发送缓存寄存器（SPIx_TXBUF）

名称	SPIx_TXBUF (x=0,1,2,3)							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TXBUF							
位权限	W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TXBUF							
位权限	W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TXBUF							
位权限	W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF							
位权限	W-00000000							

Bit	助记符	功能描述
31:0	**TXBUF**	SPI 发送缓存(Transmit Buffer)

23.4.7 SPI/I2S 发送缓存寄存器（SPIx_RXBUF）

名称	SPIx_RXBUF (x=0,1,2,3)							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RXBUF							
位权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RXBUF							
位权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RXBUF							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF							
位权限	R-00000000							

Bit	助记符	功能描述

Bit	助记符	功能描述
31:0	**RXBUF**	SPI 接收缓存(Receive Buffer)

23.4.8 I2S 配置寄存器（SPIx_I2SCR）

名称	SPIx_I2SCR (x=0,1,2,3)							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			RXOPT	I2SSTD		CHLEN	CKPOL
位权限	U-0			R/W-0	R/W-00		R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DLEN		PCMSYNC	I2SMM		-	I2SMOD	I2SEN
位权限	R/W-00		R/W-0	R/W-0		U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:13	**--**	RFU：未实现，读为 0
12	**RXOPT**	I2S 主机接收时的时钟输出选项，用于确保 i2s 输出时钟的完整性 0：一直连续输出（关闭 i2s_en 时马上停止输出时钟） 1：检测 rxbuf 有数据，停止时钟（建议最后一帧数据使用，否则输出时钟不连续）
11:10	**I2SSTD**	I2S 标准选择 00：Philips 标准 01：MSB-justified 10：LSB-justified 11：PCM 标准
9	**CHLEN**	声道长度（每个声道的 bit 数） 0：16bit 1：32bit 仅 DLEN=00 的情况下有效，否则 CHLEN 固定为 32bit
8	**CKPOL**	I2S 空闲时钟极性 0：CK 空闲时保持低电平 1：CK 空闲时保持高电平
7:6	**DLEN**	I2S 通信数据长度 00：16bit 01：24bit 10：32bit 11：RFU
5	**PCMSYNC**	PCM 帧同步 0：短帧同步 1：长帧同步 仅 I2SSTD=11 时有效

Bit	助记符	功能描述
4:3	**I2SMM**	I2S Master Mode 00：从机接收 01：从机发送 10：主机接收 11：主机发送
2	**--**	RFU：未实现，读为 0
1	**I2SMOD**	外设模式选择 0：SPI 模式 1：I2S 模式
0	**I2SEN**	I2S 使能 1：使能 I2S 模块 0：关闭 I2S 模块 此寄存器仅在 I2SMOD=1 的情况下有效

23.4.9 I2S 预分频寄存器（SPIx_I2SPR）

名称	SPIx_I2SPR (x=0,1,2,3)							
offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						MCKOE	ODD
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CLKDIV							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:10	**--**	RFU：未实现，读为 0
9	**MCKOE**	Master 时钟输出使能（仅主机模式有效） 0：禁止 MCK 输出 1：使能 MCK 输出
8	**ODD**	波特率预分频小数部分 0：实际分频系数是 CLKDIV*2 1：实际分频系数是 CLKDIV*2 + 1
7:0	**CLKDIV**	I2S 波特率分频寄存器，禁止使用 0 和 1 此寄存器定义 CK 高低电平宽度（单位是 APBCLK 周期），CK 频率是 F_{APBCLK}/(CLKDIV*2)

24 经典控制器区域网络（FSCAN）

24.1 概述

FSCAN模块用于经典CAN总线数据收发，支持CAN2.0A和2.0B协议。本芯片最大支持2路独立的FSCAN模块。

FSCAN模块主要特性如下：

- 符合ISO11898-1协议，支持CAN2.0A和CAN2.0B标准
- 支持standard（11bit ID）和extended（29bit ID）帧
- 支持最高波特率1Mbps
- 8 messages接收FIFO
- 3个独立的发送buffer
- 支持错误或仲裁失败条件下的自动重发
- 最大32个扩展ID接收滤波器
- 支持loopback模式
- 支持异步唤醒
- 发送和接收错误计数器
- 自动bus-off恢复或软件控制的bus-off恢复

24.2 结构框图

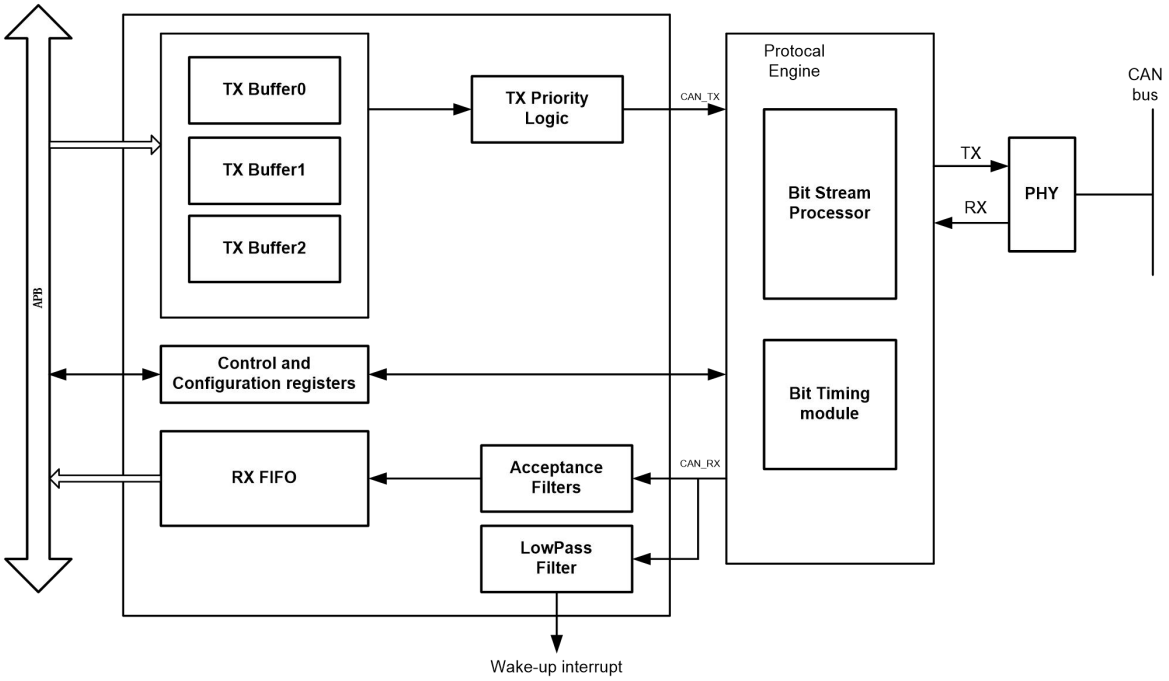


图 24-1FSCAN 模块结构框图

FSCAN模块需要外接一个CAN PHY芯片用于实现物理层数据收发。

24.3 功能描述

24.3.1 时钟和复位

CAN模块有两个时钟：CAN_CLK和SYS_CLK。

CAN总线的数据收发使用CAN_CLK进行，APB总线的寄存器访问使用SYS_CLK进行。CAN_CLK和SYS_CLK之间没有频率约束关系。系统上SYS_CLK连接到APBCLK。

- CAN_CLK频率范围：8~24Mhz
- SYS_CLK频率范围：8~64Mhz

CAN_CLK可以选择使用外部晶振、PLL，使用时钟的频率精度需满足ISO11898-1规定的容差范围。

CAN模块的复位包括系统复位、RMU软件复位、模块内部软件复位。其中，通过对模块内部寄存器SRST写1，可以实现模块内部的自复位操作，此时模块内所有配置和控制寄存器，包括SRST本身，会被复位成复位值。

24.3.2 Bit timing

CAN数据bit时序可以划分为4个部分：

- 同步段
- 传输段
- 相位段1
- 相位段2

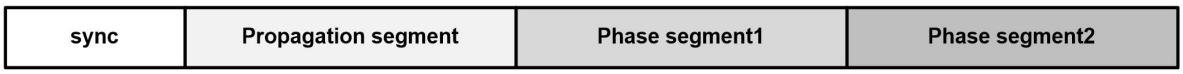


图 24-2 CAN bit timing

每个段都是由一定数量的时间片组成，称为time quanta（tq），量子时钟周期决定tq时间长度，而量子时钟是由CAN模块内部对CAN_CLK分频得到的。通过设置波特率预分频寄存器BRPR可以配置量子时钟相对于CAN_CLK的分频系数。

传输段和相位段1组成了时间段1（TS1），相位段2组成时间段2（TS2）；通过Bit-Timing寄存器BTR可以设置TS1和TS2中tq的数量。同步段总是1个tq长度。

24.3.3 Bit stream processor

BSP模块用于实现CAN总线协议中的MAC/LLC功能。在数据发送中BSP负载完成以下工作：

- 并行数据串行化
- 根据协议要求插入stuff bits、CRC、和其他协议数据域

在发送时BSP同时监控接收数据以执行总线仲裁，当仲裁失败时，将执行重发任务。

在接收时BSP移除stuff bits、CRC位和其他协议数据域，将有效负载数据存入RX FIFO，并进行数据错误监控。

根据总线错误状态，BSP控制CAN控制器进入对应的错误状态：Error Active，Error Passive，Bus Off。

当TX或RX数据信号上发现传输错误事件，BSP模块根据ISO11898-1标准规定的规则更新总线错误计数器（CAN_ECR），并根据ECR的值使CAN控制器进入各种错误状态。

24.3.4 控制器工作模式

CAN控制器模块支持以下工作模式：

- Configuration
- Normal
- Sleep
- Loop Back

Configuration mode

当以下任意事件发生时，CAN控制器将进入Configuration mode：

- CEN寄存器写0
- SRST寄存器写1
- CAN控制器复位后默认进入configuration mode

此模式下CAN控制器处于如下状态：

- 停止与总线同步，并驱动固定的隐性位输出
- ECR和ESR寄存器复位
- BTR和BRPR寄存器可以改写
- CAN控制器不再接收任何消息
- TX FIFO和TX HPB中的消息不会被发送，但是会被保留；当进入Normal模式后，这些消息会被发送
- 可以读取RX FIFO中的消息
- 可以向TX FIFO和TX HPB写入消息
- 所有配置寄存器可以访问

在configuration mode下，置位CEN寄存器将使CAN控制器在等待11个连续隐性位之后退出此模式。退出configuration mode后，根据LBACK和SLEEP寄存器状态，CAN控制器会进入Normal、LoopBack或Sleep模式。

Normal mode

在normal mode下，CAN控制器参与总线通信，正常收发消息。从normal模式控制器可以进入Confiuration或Sleep模式。

Sleep mode

FSCAN控制器可以从Configuration或者Normal模式进入Sleep模式。

在Normal模式下，如果SLEEP寄存器置1，TXbuffer中没有剩余消息，并且CAN总线IDLE，FSCAN控制器将进入Sleep模式。

在Configuration模式下，如果LPBACK寄存器为0，SLEEP寄存器为1，且CEN寄存器为1，CAN控制器将进入Sleep模式。

当总线上其他节点发送消息，或者软件对TX buffer写入数据，FSCAN控制器将退出Sleep模式。控制器进入或退出Sleep模式时都能够产生中断。

在Sleep模式下，FSCAN控制器可以进入Configuration或者Normal模式。

在Sleep模式下，如果使能了CAN_RX异步唤醒功能（WUPE=1），则CAN_RX上的下降沿信号将置位唤醒中断，能够将MCU从休眠模式唤醒，随后软件可以配置FSCAN控制器进入configuration或normal模式。

Loop Back mode

在loop back模式下，CAN控制器不参与总线通信，仅向总线持续发送隐性位，也不接收CAN总线上的任何消息。同时内部发送的bit流直接环回到接收端，实现自发自收，用于测试和诊断。

控制器仅能从configuration模式进入loop back模式，通过设置LPBACK寄存器为1，并且CEN寄存器为1。

24.3.5 寄存器写权限与控制器模式

以下寄存器仅能在configuration模式下改写：

- MSR寄存器中：

  TPRI（发送优先级配置），WUPE（唤醒使能），AREN（自动重发），ABOM（busoff恢复配置），LPBACK
- 位时序相关寄存器：

  CAN_BRPR, CAN_BTR

24.3.6 消息帧存储和消息帧结构（Message storage and structure）

消息存储

CAN总线通信以消息帧（Message Frame）为基本组织形式。待发送的消息保存在TX buffer中，模块共包含3个独立的发送buffer，并具有固定的优先级：buffer0 > buffer1 > buffer2，即编号小的发送buffer优先级高。

CAN控制器接收的消息首先经过acceptance filter过滤，符合滤波器规则的消息才会被存入RX FIFO，否则将被丢弃。RX FIFO中可以保存2条消息。

消息的发送和接收遵从以下规则：

- 编号越小的发送buffer优先级越高
- 发送过程中出现仲裁失败或错误，CAN控制器将尝试重发当前消息；只有当前消息成功发送完成后，才会发送后续消息，即使是HPB中的消息也不能打断当前消息的重发过程
- TX buffer、RX FIFO中的消息数据在CAN控制器处于Bus Off状态或Configuration模式时仍会保留

消息结构

CAN总线的每条消息长度为16字节，包含消息ID（4字节）、数据长度码DLC（4字节）、Data Word 1（4字节）和Data Word 2（4字节）。

消息ID（Identifier）

bit	31	30:13	12	11	10:0
field	RTR	ID[17:0]	IDE	SRR/RTR	ID[28:18]

数据长度码（Data Length Code – DLC）

bit	31:4	3:0
field	RFU	DLC

数据字1（Data Word 1）

bit	31:24	23:16	15:8	7:0
field	DB3[7:0]	DB2[7:0]	DB1[7:0]	DB0[7:0]

数据字2（Data Word 2）

bit	31:24	23:16	15:8	7:0
field	DB7[7:0]	DB6[7:0]	DB5[7:0]	DB4[7:0]

消息接收时，软件必须从RX FIFO读取完整16字节，即使接收的消息数据长度小于8字节；每个消息读取分4次32bit读操作完成，按照ID、DLC、DW1、DW2的顺序依次从FIFO读出。

消息发送时，软件必须向TX buffer写入完整的16字节，必须按照ID、DLC、DW1、DW2的顺序依次写入4个word，即使有效数据长度少于8字节，也必须完成4个word写入，DW中空余位应填充0。数据发送时，每个字节的MSb先发，数据字节按照DB0~DB7的顺序发送。

消息识别码（Identifier）

CAN控制器支持两种格式的消息识别码：

- 标准帧：ID长度为11bit，仅ID[28:18]、SRR/RTR和IDE为有效位，其中IDE为0，SRR/RTR根据数据帧和远程帧来选择。
- 扩展帧：ID长度为29bit，所有位都有效，SRR/RTR和IDE为1

Bits	Name	Descriptions
31	RTR	Remote Transmission Request 仅扩展帧有效 1：remote frame 0：data frame
30:13	ID[18:0]	Extended Message ID 仅扩展帧有效 标准帧应为全0
12	IDE	Identifier Extension

		1：使用扩展消息识别码，29位ID 0：使用标准消息识别码，11位ID
11	SRR/RTR	Substitute Remote Transmission Request 仅标准帧有效，扩展帧应保持1 1：表示当前消息帧是一个remote frame 0：表示当前消息帧是一个data frame
10:0	ID[28:18]	Standard Message ID 标准帧和扩展帧都有效

24.3.7 控制器初始化

芯片上电复位或系统复位后，在进行CAN总线通信前，需按照以下说明配置CAN总线寄存器。

- 选择操作模式

对于回环模式，置位MSR中的LPBACK位，清零MSR中的SLEEP位。

对于睡眠模式，置位MSR中的SLEEP位，清零MSR中的LPBACK位。

对于正常模式，清零MSR中的LPBACK和SLEEP位。

- 配置传输层配置寄存器

对波特率预分频寄存器BRPR和位时序寄存器BTR进行编程，使其与网络定时参数和系统的网络特性相对应。

- 配置接收滤波寄存器

清零CAN_AFR寄存器中的UAF位，并通过FSC和MSKM寄存器配置滤波器长度、个数和滤波策略

查询CAN_SR中的ACFBSY位直到返回0

将适当的掩码信息写入接收滤波掩码寄存器AFMRx

将适当的ID信息写入到接收滤波寄存器AFIRx

置位CAN_AFR寄存器中的UAF位

对每组需使能的接收滤波掩码寄存器AFMRx和接收滤波寄存器AFIRx对重复上述步骤

配置接收中断使能寄存器CAN_RX_IER以选择接收中断状态寄存器CAN_RX_ISR中可生成中断的标志位

置位CAN_CR寄存器中的CEN位来启用CAN

24.3.8 消息发送

消息发送采用队列形式，包含3个独立的发送buffer，每个发送buffer保存一个完整的消息帧。

软件可以直接操作所有的发送buffer，每个发送buffer有独立的发送请求（transmit request）、发送撤销请求（transmit abort request）、buffer空标志，并根据buffer编号具有固定优先级。

软件写buffer前需要根据TXE寄存器判断发送buffer是否空闲，对非空闲buffer写入的新数据将被丢弃。

24.3.8.1 发送优先级判决

如果多个发送buffer的发送请求同时有效，发送优先级判决将被执行。优先级判决可以基于以下两种方法：

- ID优先级（TPRI=0）
- Buffer编号优先级（TPRI=1）

当TPRI=0，发送优先级判决基于buffer中的消息ID。消息ID的优先级规则符合ISO11898-1规定的仲裁规则，经过内部仲裁后，ID值最小的消息帧将被首先发送。

当TPRI=1，发送优先级判决基于buffer编号。请求发送的多个buffer中编号最小的被最先发送。

在使能了自动重发功能后，如果出现发送错误或者仲裁失败，上一次赢得优先发送的buffer将被自动重发。

24.3.8.2 发送流程

- 查询TXE寄存器，确认当前发送buffer是否空闲
- 将待发送消息帧或远程帧写入目标发送buffer，写1清零对应的TXE寄存器，表示buffer中数据就绪可以发送
- 根据需要置位TXE中断使能
- FSCAN自动进行优先级判决并在总线空闲时启动发送
- 如果消息成功发送，TXE寄存器自动置位，TXSF[1:0]寄存器指示发送完成状态（成功发送或者abort），如果中断使能，则发起TXE中断

Buffer空寄存器TXE由软件清零，并在以下条件下自动置位：

- 发送成功完成
- 发送成功撤销
- 仲裁失败或发送出现错误，并且AREN=0（关闭自动重发）

24.3.8.3 发送撤销（transmit abort）

当某个发送buffer的发送请求寄存器被置位，软件可以通过置位TXAR寄存器来撤销发送。当发送撤销成功完成，TXE置位，TXSF指示当前发送被成功撤销。

正在发送中的消息不能被撤销。然而，如果一个被撤销但是已经在发送的消息输掉了仲裁或发生了错误，该消息不会被自动重发，而是直接结束发送并由TXSF指示当前发送被撤销。

不论发送成功完成还是成功撤销，TXAR都会被自动清零。

24.3.9 消息接收

接收到的消息如果成功通过了消息滤波器，将被存入深度为2的接收FIFO中，同时产生接收完成中断（RXOK）。消息长度为固定的16字节，软件必须从RX FIFO读取完整16字节，即使接收的消息数据长度小于8字节；每个消息读取分4次32bit读操作完成，按照ID、DLC、DW1、DW2的顺序依次从FIFO读出。

当接收FIFO已满的情况下收到新的消息，将导致FIFO溢出事件中断，新的消息帧将被丢弃。

每当接收到新消息并将其写入RX FIFO时，CAN_ISR中的RXNEMP位和RXOK位都会被置1。如果对空RX FIFO执行读取操作，则CAN_ISR中的RXUFLW位置1。

从RX FIFO读取消息：

1、轮询CAN_ISR中的RXOK或RXNEMP位。在中断模式下，读FIFO可以在CAN_ISR中的RXOK或RXNEMP位生成中断后发生执行。

-从接收 FIFO ID寄存器读取消息的ID

-从接收 FIFO DLC寄存器读取消息的数据长度

-从接收 FIFO DW1位置读取消息DW1

-从接收 FIFO DW2位置读取消息DW2

执行读取后，如果RX FIFO中有一条或多条消息，则置位CAN_ISR中的RXNEMP位。该位既可以被轮询，也可以生成中断。

重复此操作，直到FIFO为空。

24.3.10 消息滤波器（Acceptance Filter）

CAN控制器带有16组32bit消息滤波器，每组消息滤波器包含一个掩码寄存器（AFMR）和一个ID寄存器（AFIR）。

滤波器模式

每个消息滤波器组（AFIR+AFMR）支持两种模式，通过Filter Scale寄存器FSC[3:0]配置：

- 扩展模式（default）：包含11位base ID、18位ID extension、SRR、IDE、RTR
- 标准模式：包含11位base ID、RTR、IDE、ID extension的高3bit

16组消息滤波器由16bit使能寄存器UAF1~15使能或关闭。消息滤波器默认为扩展模式，当使用标准模式时，每组的2个16bit滤波器不能单独使能或关闭。掩码寄存器总是与ID寄存器一起使能和关闭。

掩码寄存器用作ID寄存器

每个Mask寄存器也可以用作ID寄存器，因此FSCAN最大可以支持32个32bit绝对ID滤波，由MSKM[3:0]寄存器配置。

例如，MSKM=1000，掩码寄存器CAN_AFMR4将被用作ID寄存器，格式定义与CAN_AFIR相同。

注意：由于32bit ID和16bit ID在消息滤波ID寄存器中的映射顺序不同，配置AFIR和AFMR寄存器之前，应先配置AFR寄存器中的MSKM和FSC位。

映射关系

下面以第一组消息滤波器为例说明消息滤波器寄存器和消息ID的映射关系

1） FSC[0]=0，MSKM=0，32bit滤波＋掩码

ID	AFIR0[31:24]	AFIR0[23:16]	AFIR0[15:8]	AFIR0[7:0]
MASK	AFMR0[31:24]	AFMR0[23:16]	AFMR0[15:8]	AFMR0[7:0]

mapping	RTR	ID[28:11] (EXID[17:0])	IDE	SRR	ID[10:0] (SID[10:0])

这种模式下，共支持16组32bit消息ID滤波。消息滤波寄存器数据组织格式请参见寄存器章节。

2） FSC[0]=0，MSKM=1，2组32bit滤波

ID	AFIR0[31:24]	AFIR0[23:16]	AFIR0[15:8]	AFIR0[7:0]
ID	AFMR0[31:24]	AFMR0[23:16]	AFMR0[15:8]	AFMR0[7:0]

mapping	RTR	ID[28:11] (EXID[17:0])	IDE	SRR	ID[10:0] (SID[10:0])

这种模式下，共支持32组32bit消息ID滤波。消息滤波寄存器数据组织格式请参见寄存器章节。

3） FSC[0]=1，MSKM=0，2组16bit滤波＋掩码

ID	AFIR0[31:24]	AFIR0[23:16]
MASK	AFMR0[31:24]	AFMR0[23:16]

ID	AFIR0[15:8]	AFIR0[7:0]
MASK	AFMR0[15:8]	AFMR0[7:0]

mapping	EXID[17:15]	IDE	RTR	SID[10:0]

这种模式下，共支持32组16bit消息ID滤波。消息滤波寄存器数据组织格式请参见寄存器章节。

注：上图中RTR在标准帧中代表RTR，在扩展帧中代表SRR；如果IDE=1，表示过滤扩展帧，低3bit作为扩展ID[17:15]参与比较，如果IDE=0，表示过滤标准帧，低3bit不参与比较。此时，32bit滤波器被展开成2组16bit滤波器使用。

注：如果AFMR中将AMIDE=0，则滤波器只比较SID[10:0]，EXID[17:15]不参与比较。如果AMIDE=1，滤波器根据IDE的值决定是否比较EXID[17:15]。

4） FSC[0]=1，MSKM=1，4组16bit滤波

ID	AFIR0[31:24]	AFIR0[23:16]
ID	AFMR0[31:24]	AFMR0[23:16]

ID	AFIR0[15:8]	AFIR0[7:0]
ID	AFMR0[15:8]	AFMR0[7:0]

mapping	EXID[17:15]	IDE	RTR	SID[10:0]

这种模式下，共支持64组16bit消息ID滤波。消息滤波寄存器数据组织格式请参见寄存器章节。

注：上图中RTR在标准帧中代表RTR，在扩展帧中代表SRR；如果IDE=1，表示过滤扩展帧，低3bit作为扩展ID[17:15]参与比较，如果IDE=0，表示过滤标准帧，低3bit不参与比较。此时，32bit滤波器被展开成4组16bit滤波器使用。

消息滤波工作流程

- 接收到的消息ID先与Mask寄存器进行逻辑与（如果掩码寄存器被用作ID滤波则跳过这个步骤）
- 滤波器ID寄存器内容同样与Mask寄存器进行逻辑与（如果掩码寄存器被用作ID滤波则跳过这个步骤）
- 两个逻辑与操作结果进行比较（如果掩码寄存器被用作ID滤波则直接比较输入ID和ID寄存器）
- 如果两个结果相等，本消息通过滤波器，被存入RX FIFO
- 多个滤波器可以同时工作，接收消息只要通过其中任意滤波器，就会被存入RX FIFO
- 无法通过滤波器的消息帧被丢弃

注意：软件可以选择使能任意多个消息滤波器，或者关闭所有消息滤波器。如果没有一个消息滤波被使能，则所有接收到的消息都会存入RX FIFO。

24.3.11 错误管理

24.3.11.1 节点策略

FSCAN 包含发送错误计数器和接收错误计数器。当消息帧被正确的发送或接收，相应的错误计数器应被递减，而当在发送或接收中遇到错误，相应的错误计数器应被递增。

下面罗列了错误计数的详细规则：

- 当接收节点检测到错误，接收错误计数器+1 （例外：发送主动错误标志或过载标志时检测到的 bit 错误不会导致接收错误计数器递增）
- 接收节点发送主动错误标志后，检测到后续第一个 bit 是显性位，则接收错误计数器+8
- 发送节点发送错误标志，发送错误计数器+8；以下例外情况下发送错误计数器保持不变
  - 发送节点处于被动错误状态，检测到 ACK 错误，并且被动错误标志中没有出现显性位
  - 仲裁阶段检测到填充位错误，并且发送的是隐性填充位，实际检测到显性填充位
- 发送节点在发送主动错误标志或过载标志时检测到 bit 错误，发送错误计数器+8
- 当检测到总线上连续 14 个显性位，或者被动错误标志后检测到连续 8 个显性位，发送节点应将发送错误计数器+8，接收节点应将接收错误计数器+8
- 当一个消息帧被正确发送后，发送错误计数器-1，减到 0 停止
- 当一个消息帧被正确接收后，接收错误计数器如果在 1~127 之间，则应-1，如果是 0 则保持不变，如果大于 127 则被设置一个 119 到 127 之间的数

24.3.11.2 错误状态转移

FSCAN 退出 configuration 模式后，进入主动错误状态（error active）。此状态下，CAN 控制器检测到错误后会主动发送错误帧。

当接收错误计数器或发送错误计数器的计数值大于 127，FSCAN 将进入被动错误状态（error passive）。此状态下，FSCAN 不再发送错误帧。

当被动错误节点的接收错误计数器和发送错误计数器的计数值都小于等于 127 时，FSCAN 回到主动错误状态。

当节点的发送错误计数器值大于 255，节点应进入总线关闭状态（bus-off）。

处于总线关闭状态的节点不能发送任何帧，也不能发送 ACK，对总线无影响。

节点错误状态转移可以符合下图所示的 ISO11898-1 协议规范。

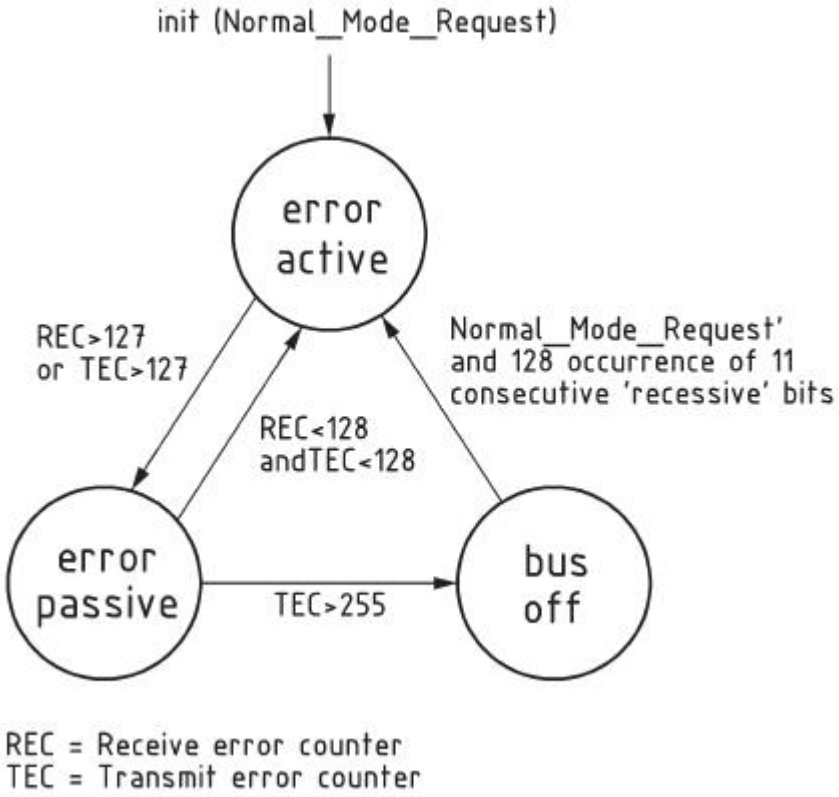


图 24-3 节点错误状态转移

FSCAN定义了以下错误状态：

- 正常接收、正常发送：TEC和REC都小于96
- 接收warning：REC大于等于96并且小于128
- 接收error：REC大于等于128
- 发送warning：TEC大于等于96并且小于128
- 发送error：TEC大于等于128并且小于256
- Bus-off：TEC大于等于256

当FSCAN内部状态机在以上状态之间发送切换时，ESCF中断标志寄存器置位，可以通知软件采取措施。同时，RSTS和TSTS寄存器用于指示当前CAN控制器所处的状态，软件也可以通过查询REC和TEC寄存器来确定控制器状态。

24.3.11.3 Bus-off recovery

FSCAN可以支持使用以下策略从总线关闭状态中恢复。

- ABOM=00/11：自动恢复模式

  这种模式下，FSCAN采用符合ISO11898协议规范的总线关闭恢复方式。当FSCAN连续检测到11bit隐性位超过128次，控制器自动回到error active状态。发送和接收错误计数器TEC和REC都自动清零，并且BORF中断标志置位。

- ABOM=01/10：软件恢复模式

  当ABOM=01，控制器退出bus-off由软件触发。软件置位RTBO后，控制器等待128次连续11位隐性位，然后退出bus-off state

  当ABOM=10，软件置位RTBO后，控制器立即退出bus-off state并进入integration state，在11个连续隐性位后，控制器重新加入CAN总线。

24.3.11.4 错误管理中断和标志

错误管理中断标志寄存器用于通知软件FSCAN内部错误状态发生了转换。

24.3.12 异步唤醒

FSCAN能够在没有CANCLK时钟的情况下监视CAN_RX信号的下降沿（隐性电平->显性电平），并产生异步唤醒中断，将处于休眠模式的MCU唤醒。下降沿检测带有一个模拟滤波电路，能够滤除小于100ns的毛刺信号，避免噪声干扰引起误唤醒。

借助此功能，可以在芯片休眠状态下实现CAN总线活动唤醒MCU。使用流程如下。

- 等待总线空闲
- 使能CAN_RX异步唤醒功能，并置位唤醒中断使能寄存器
- 配置FSCAN进入sleep mode
- 芯片进入休眠
- CAN总线下降沿置位唤醒中断，唤醒MCU
- MCU退出休眠，执行时钟初始化
- 软件配置FSCAN进入configuration或normal

注意：WUPF中断标志仅在FSCAN处于Sleepmode时才会置位。

24.4 寄存器

CAN0模块起始地址：0x40019400

CAN1模块起始地址：0x40018800

offset 地址	名称	符号
0x00	CAN 控制寄存器 (CAN Control Register)	CAN_CR
0x04	CAN 模式选择寄存器 (CAN mode select register)	CAN_MSR
0x08	波特率预分频寄存器 (CAN Baud rate Prescaler Register)	CAN_BRPR
0x0C	CAN 位时序寄存器 (CAN Bit Timing Register)	CAN_BTR
0x10	CAN 错误计数寄存器 (CAN Error Counter Register)	CAN_ECR
0x14	CAN 错误状态寄存器 (CAN Error Status Register)	CAN_ESR
0x18	CAN 状态标志寄存器 (CAN Status Register)	CAN_SR
0x1C	CAN 接收中断标志寄存器 (CAN Interrupt Status Register)	CAN_RXISR
0x20	CAN 接收中断使能寄存器 (CAN Interrupt Enable Register)	CAN_RXIER
0x24		
0x28	CAN 发送缓冲控制寄存器 (CAN Transmit Buffer Control Register)	CAN_TXBCR
0x2C	CAN 发送中断状态寄存器 (CAN Transmit Buffer Status Register)	CAN_TXBSR
0x30	CAN 发送缓冲 0ID 寄存器 (CAN Transmit Buffer 0 ID Register)	CAN_TXB0IDR
0x34	CAN 发送缓冲 0DLC 寄存器 (CAN Transmit Buffer 0 DLC Register)	CAN_TXB0DLCR
0x38	CAN 发送缓冲 0 数据字寄存器 1 (CAN Transmit Buffer 0 DataWord1 Register)	CAN_TXB0DW1R
0x3C	CAN 发送缓冲 0 数据字寄存器 2 (CAN Transmit Buffer 0 DataWord2 Register)	CAN_TXB0DW2R
0x40	CAN 发送缓冲 1 ID 寄存器 (CAN Transmit Buffer 1 ID Register)	CAN_TXB1IDR
0x44	CAN 发送缓冲 1 DLC 寄存器 (CAN Transmit Buffer 1 DLC Register)	CAN_TXB1DLCR
0x48	CAN 发送缓冲 1 数据字寄存器 1 (CAN Transmit Buffer 1 DataWord1 Register)	CAN_TXB1DW1R
0x4C	CAN 发送缓冲 1 数据字寄存器 2 (CAN Transmit Buffer 1 DataWord2 Register)	CAN_TXB1DW2R

0x50	CAN 发送缓冲 2ID 寄存器 (CAN Transmit Buffer 2 ID Register)	CAN_TXB2IDR
0x54	CAN 发送缓冲 2 DLC 寄存器 (CAN Transmit Buffer 2 DLC Register)	CAN_TXB2DLCR
0x58	CAN 发送缓冲 2 数据字寄存器 1 (CAN Transmit Buffer 2 DataWord1 Register)	CAN_TXB2DW1R
0x5C	CAN 发送缓冲 2 数据字寄存器 2 (CAN Transmit Buffer 2 DataWord2 Register)	CAN_TXB2DW2R
0x60	CAN 接收滤波控制寄存器 (Acceptance Filter Control Register)	CAN_AFCR
0x64	CAN 接受滤波配置寄存器 (Acceptance Filter Config Register)	CAN_AFCFGR
0x80	CAN 接收滤波掩码寄存器 0 (Acceptance Filter Mask Register0)	CAN_AFMR0
0x84	CAN 接收滤波 ID 寄存器 0 (Acceptance Filter ID Register0)	CAN_AFIR0
0x88	CAN 接收滤波掩码寄存器 1 (Acceptance Filter Mask Register1)	CAN_AFMR1
0x8C	CAN 接收滤波 ID 寄存器 1 (Acceptance Filter ID Register1)	CAN_AFIR1
…	…	…
0xF8	CAN 接收滤波掩码寄存器 15 (Acceptance Filter Mask Register15)	CAN_AFMR15
0xFC	CAN 接收滤波 ID 寄存器 15 (Acceptance Filter ID Register15)	CAN_AFIR15
0x100	接收 FIFO ID 寄存器 (CAN RX FIFO ID Register)	CAN_RXF_IDR
0x104	接收 FIFO DLC 寄存器 (CAN RX FIFO DLC Register)	CAN_RXF_DLCR
0x108	接收 FIFO Data Word1 寄存器 (CAN RX FIFO Data Word1 Register)	CAN_RXF_DW1R
0x10C	接收 FIFO Data Word2 寄存器 (CAN RX FIFO Data Word2 Register)	CAN_RXF_DW2R

24.4.1 CAN 控制寄存器（CAN_CR）

名称	CAN_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	CEN	SRST
位权限	U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	CEN	CAN 控制器使能 (CAN enable) 0：CAN 控制器处于 configuration mode 1：CAN 控制器根据 LPBACK 和 SLEEP 寄存器状态，进入 LoopBack、Sleep 或 Normal 模式
0	SRST	软件复位寄存器 (Software reset) 写 1 复位 CAN 控制器，读出总是 0

24.4.2 CAN 配置寄存器（CAN_MSR）

名称	CAN_MSR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		RX_PRESYNC_EN	TPRI	WUPE	AREN	ABOM	
位权限	U-0		R/W-1	R/W-1	R/W-0	R/W-1	R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RTBO	-					LPBACK	SLEEP
位权限	R/W-0	U-0					R/W-0	R/W-0

Bit	助记符	功能描述
31:14	--	RFU：未实现，读为 0
13	RX_PRESYNC_EN	CAN 接收数据预采样同步使能 0：不经过预采样 1：经过预采样
12	TPRI	发送优先级模式选择 1：采用发送 buffer 编号优先级方式 0：采用消息 ID 优先级方式
11	WUPE	唤醒使能 1：允许 MCU 休眠时使用 CAN_RX 信号下降沿唤醒 0：禁止唤醒功能
10	AREN	自动重发使能 1：使能自动重发功能 0：关闭自动重发
9:8	ABOM	自动总线关闭管理(Automatic Bus Off Management) 此寄存器控制 CAN 在退出 bus-off 状态时的行为

Bit	助记符	功能描述
		00/11：检测到 128 次连续 11 个隐性位的情况下，硬件自动退出 bus-off state（兼容 ISO11898） 01：软件退出 bus-off：对 RTBO 置位后，控制器等待检测到 128 次连续 11 个隐性位的情况下退出 bus-off state 10：软件退出 bus-off：对 RTBO 置位后，控制器立即退出 bus-off state 并进入 integration state
7	RTBO	软件强制从 bus-off 退出 此寄存器仅在 ABOM=01/10 的情况下有效且可写 软件置位后，控制器在退出 bus-off state 后自动清零
6:2	--	RFU：未实现，读为 0
1	LPBACK	Loop Back mode 1：CAN 控制器处于 Loop Back 模式 0：CAN 控制器处于 Normal、Configuration 或 Sleep 模式
0	SLEEP	Sleep mode 1：CAN 控制器处于 Sleep 模式 0：CAN 控制器处于 Normal、Configuration 或 Loop Back 模式

24.4.3 CAN 波特率预分频寄存器（CAN_BRPR）

名称	CAN_BRPR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BPR							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	BRP	波特率预分频 (Baud Rate Prescaler)，根据这个寄存器的值对 CAN_CLK 进行预分频得到量子时钟频率 实际分频值是 BPR+1 Tq=tCAN_CLK * (BPR+1) 注：此寄存器只能在 configuration 模式下改写

24.4.4 CAN 位时序寄存器（CAN_BTR）

名称	CAN_BTR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							SJW[1]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SJW[0]	TS2			TS1			
位权限	R/W-0	R/W-000			R/W-0000			

Bit	助记符	功能描述
31:9	--	RFU：未实现，读为 0
8:7	SJW	Synchronization Jump Width，参见 CAN2.0 协议 tSJW = tq * (SJW + 1)
6:4	TS2	Time Segment2，定义 Phase Segment2 的长度 tTS2 = tq * (TS2 + 1)
3:0	TS1	Time Segment1，定义 Propagation Segment + Phase Segment1 的长度 tTS1 = tq * (TS1 + 1)

24.4.5 CAN 错误计数寄存器（CAN_ECR）

名称	CAN_ECR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	REC							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TEC							
位权限	R-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:8	REC	接收错误计数器 (Receive Error Counter)

Bit	助记符	功能描述
		软件只读，每次发生接收错误时 REC 递增，正确接收后递减
7:0	TEC	发送错误计数器 (Transmit Error Counter) 软件只读，每次发生发送错误时 TEC 递增，正确发送后递减

注：递增递减规则参见“错误管理”-“节点策略”章节。

注：在以下情况下ECR寄存器清零

- SRST寄存器写1
- CEN寄存器清零
- CAN控制器进入Bus Off
- Bus off恢复

24.4.6 CAN 错误状态寄存器（CAN_ESR）

名称	CAN_ESR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							OVLF
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BLF	ADERR	ACKER	B0ERR	B1ERR	STER	FMER	CRCER
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:9	--	RFU：未实现，读为 0
8	OVLF	Overload flag，总线上接收到过载帧时置位，软件写 1 清零 1：检测到 overload 0：没有检测到 overload
7	BLF	Bus Lock Flag，当总线上连续检测到 32 个显性位时置位，软件写 1 清零 1：bus locked 0：no bus lock
6	ADERR	ACK delimiter error 1：检测到 ACK delimiter 错误 0：无 ACK delimiter 错误 硬件置位，软件写 1 清零
5	ACKER	Acknowledge Error 1：检测到 ACK 错误 0：无 ACK 错误

Bit	助记符	功能描述
		硬件置位，软件写 1 清零
4	B0ERR	Bit0 Error 1：接收到的 bit 不是正在发送的 dominant bit 0：无 bit 错误 硬件置位，软件写 1 清零
3	B1ERR	Bit1 Error 1：接收到的 bit 不是正在发送的 recessive bit 0：无 bit 错误 硬件置位，软件写 1 清零
2	STER	Stuffing Error 1：数据填充位错误 0：无填充错误 硬件置位，软件写 1 清零
1	FMER	Form Error 1：消息帧中固定格式域出现错误 0：无错误 硬件置位，软件写 1 清零
0	CRCER	CRC Error 1：CRC 校验错误 0：CRC 校验正确 硬件置位，软件写 1 清零

24.4.7 CAN 状态标志寄存器（CAN_SR）

名称	CAN_SR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				ACFBSY	RXFE	RSTS	
位权限	U-0				R-0	R-1	R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TSTS		BBSY	BIDLE	NORMAL	SLEEP	LPBACK	CONFIG
位权限	R-00		R-0	R-0	R-0	R-0	R-0	R-1

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11	ACFBSY	Acceptance Filter Busy 1：Acceptance Filter 工作，不能改写 Acceptance Filter Mask 和 ID 寄存器 0：Acceptance Filter 空闲，可以改写 Mask 和 ID 寄存器
10	RXFE	Receive FIFO empty 标志 1：接收 FIFO 空

Bit	助记符	功能描述
		0：接收 FIFO 非空
9:8	RSTS	FSCAN 内部接收错误状态指示 00：正常接收状态，REC<96 01：Warning 状态，96≦REC<128 10：Error 状态，128≦REC 11：Bus-off，这是一个冗余状态，因为接受错误并不会使 CAN 控制器进入 bus-off，这个状态与 TSTS=11 等价
7:6	TSTS	FSCAN 内部发送错误状态指示 00：正常发送状态，TEC<96 01：Warning 状态，96≦TEC<128 10：Error 状态，128≦TEC<256 11：Bus-off，256<TEC
5	BBSY	Bus Busy 标志，只读 1：CAN 控制器正在进行数据收发 0：CAN 控制器处于 configuration mode，或总线 IDLE
4	BIDLE	Bus IDLE 标志，只读 1：当前无总线通信 0:当前总线正在通信,或者 CAN 控制器处于 configuration mode
3	NORMAL	Normal mode 标志，只读 1：CAN 控制器处于 normal mode 0：CAN 控制器不处于 normal mode
2	SLEEP	Sleep mode 标志，只读 1：CAN 控制器处于 sleep mode 0：CAN 控制器不处于 sleep mode
1	LBACK	Loop Back mode 标志，只读 1：CAN 控制器处于 loop back mode 0：CAN 控制器不处于 loop back mode
0	CONFIG	Configuration mode 标志，只读 1：CAN 控制器处于 configuration mode 0：CAN 控制器不处于 configuration mode

24.4.8 CAN 接收中断标志寄存器（CAN_RXISR）

名称	CAN_RX_ISR							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				RXFIFO_MC			
位权限	U-0				R-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			BORF	WUPF	-	ESCF	ERROR
位权限	U-0			R-0	R-0	U-0	R-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXNEMP	RXOFLW	RXUFLW	RXOK	-			ARBLST

位权限	R-0	R-0	R-0	R-0	U-0	R-0

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
19:16	RXFIFO_MC	RXFIFO 中的 message count，有效数据为 0~8
15:13	--	RFU：未实现，读为 0
12	BORF	Bus-off Recovery Flag，硬件置位，软件写 1 清零 1：CAN 控制器从 bus-off 恢复
11	WUPF	Wakeup 中断标志，硬件置位，软件写 1 清零 1：CAN 控制器在 Sleep 模式下检测到 RX 信号上的下降沿
10	--	RFU：未实现，读为 0
9	ESCF	错误状态转换中断标志 (Error Status Change Flag)，硬件置位，软件写 1 清零 1：CAN 控制器内部错误状态发生转换 此中断标志置位后，软件应查询 RSTS 和 TSTS 寄存器确认当前的错误状态，也可以读取 REC 和 TEC 错误状态转换事件包含进入或者退出以下状态： ● 接收 warning ● 接收 error ● 发送 warning ● 发送 error ● Bus-off
8	ERROR	错误中断标志，硬件置位，软件写 1 清零 1：消息收发过程中发生了错误 *当 CAN_ESR 寄存器中任意 bit 置位时，此寄存器置位；软件响应错误中断时，需要去 CAN_ESR 中查询具体的错误标志*
7	RXNEMP	Receive FIFO Not Empty 中断标志，硬件置位，软件写 1 清零 1：接收 FIFO 非空
6	RXOFLW	Receive FIFO Overflow 中断标志，硬件置位，软件写 1 清零 1：接收 FIFO 上溢出，即在接收 FIFO 满的情况下收到新的消息
5	RXUFLW	Receive FIFO Underflow 中断标志，硬件置位，软件写 1 清零 1：接收 FIFO 下溢出，即在接收 FIFO 空的情况下执行读操作
4	RXOK	Receive OK 中断标志，硬件置位，软件写 1 清零 1：表示 1 个消息帧被成功接收
3	--	RFU：未实现，读为 0
2	--	RFU：未实现，读为 0
1	--	RFU：未实现，读为 0
0	ARBLST	Arbitration Lost 中断标志，硬件置位，软件写 1 清零 1：仲裁失败

24.4.9 CAN 接收中断使能寄存器（CAN_RXIER）

名称	CAN_RXIER							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			BORFIE	EWKUP	ESLP	EESCF	EERROR
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ERXNEMP	ERXOFLW	ERXUFLW	ERXOK	-			EARBLST
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0			R/W-0

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12	BORFIE	Bus-off recovery 中断使能，1 允许中断，0 禁止中断
11	EWKUP	Wakeup 中断使能，1 允许中断，0 禁止中断
10	--	RFU：未实现，读为 0
9	EESCF	错误转换中断使能，1 允许中断，0 禁止中断
8	EERROR	错误中断使能，1 允许中断，0 禁止中断
7	ERXNEMP	Receive FIFO Not Empty 中断使能，1 允许中断，0 禁止中断
6	ERXOFLW	Receive FIFO Overflow 中断使能，1 允许中断，0 禁止中断
5	ERXUFLW	Receive FIFO Underflow 中断使能，1 允许中断，0 禁止中断
4	ERXOK	Receive OK 中断使能，1 允许中断，0 禁止中断
3	--	RFU：未实现，读为 0
2	--	RFU：未实现，读为 0
1	--	RFU：未实现，读为 0
0	EARBLST	Arbitration Lost 中断使能，1 允许中断，0 禁止中断

24.4.10 CAN 发送缓冲控制寄存器（CAN_TXBCR）

名称	CAN_TXBCR							
offset	0x30							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					TXAR		
位权限	U-0					R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					TXIE		
位权限	U-0					R/W-000		

Bit	助记符	功能描述
31:11	--	RFU：未实现，读为 0

Bit	助记符	功能描述
10:8	TXAR	发送撤销请求（Transmit Abort request） 软件写 1 撤销发送请求，撤消后硬件自动清零。仅 TXR[y]=1 的情况下对应 TXAR[y]位可写 TXAR[2] – TxBuffer2 TXAR[1] – TxBuffer1 TXAR[0] – TxBuffer0
7:3	--	RFU：未实现，读为 0
2:0	TXIE	TXE 中断使能寄存器 1：允许 TXE 产生中断 0：禁止 TXE 中断

24.4.11 CAN 发送中断标志寄存器（CAN_TXBSR）

名称	CAN_TXBISR							
offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		TXRF2		TXRF1		TXRF0	
位权限	U-0		R-00		R-00		R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		TXS			TXE		
位权限	U-0		R -000			R/W-000		

Bit	助记符	功能描述
31:14	--	RFU：未实现，读为 0
13:12	TXRF2	发送结果标志（Transmit buffer 2 transmit result flag） 00：无发送请求或者还在发送中 01：发送撤销成功 10：发送成功（无撤销请求） 11：发送成功（有撤销请求，未能实现撤销）
11:10	TXRF1	发送结果标志（Transmit buffer 1 transmit result flag） 00：无发送请求或者还在发送中 01：发送撤销成功 10：发送成功（无撤销请求） 11：发送成功（有撤销请求，未能实现撤销）
9:8	TXRF0	发送结果标志（Transmit buffer 0 transmit result flag） 00：无发送请求或者还在发送中 01：发送撤销成功 10：发送成功（无撤销请求） 11：发送成功（有撤销请求，未能实现撤销）
7:6	--	RFU：未实现，读为 0
5:3	TXS	发送 buffer 状态标志 0：TxBuffer 不在发送中

Bit	助记符	功能描述
		1：TxBuffer 发送中 TXS[2] – TxBuffer2 TXS[1] – TxBuffer1 TXS[0] – TxBuffer0
2:0	TXE	发送 buffer 空标志，软件写 1 清零（写 0 无效），硬件置位 0：发送 buffer 非空，准备好发送 1：发送 buffer 空 TXE[2] – TxBuffer2 TXE[1] – TxBuffer1 TXE[0] – TxBuffer0 软件将待发消息写入 buffer 后，清零对应的 TXE 寄存器，通知硬件可以发送；发送完成后，控制器自动置位 TXE 并产生中断

24.4.12 CAN 发送缓冲 ID 寄存器（CAN_TXBxIDR, x=0,1,2）

名称	CAN_TXBxIDR, x=0,1,2							
offset	0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IDR[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	IDR[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IDR[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IDR[7:0]							
位权限	W-0000 0000							

Bit	助记符	功能描述
31:0	IDR	发送消息识别码 (Identifier Register) 数据格式参见 24.3.6 消息帧存储和消息帧结构（Message storage and structure）

24.4.13 CAN 发送缓冲 DLC 寄存器（CAN_TXBxDLCR, x=0,1,2）

名称	CAN_TXBxDLCR, x=0,1,2							
offset	0x44							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				DLC			
位权限	U-0				W-0000			

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	DLC	数据长度码 (Data Length Code) 定义消息帧中的数据字节长度

24.4.14 CAN 发送缓冲 DataWord1 寄存器（CAN_TXBxDW1R, x=0,1,2）

名称	CAN_TXBxDW1R, x=0,1,2							
offset	0x48							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DB3							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DB2							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DB1							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DB0							
位权限	W-0000 0000							

Bit	助记符	功能描述
31:24	DB3	Data Byte 3
23:16	DB2	Data Byte 2
15:8	DB1	Date Byte 1
7:0	DB0	Data Byte 0

24.4.15 CAN 发送缓冲 DataWord2 寄存器（CAN_TXBxDW2R, x=0,1,2）

名称	CAN_TXBxDW2R, x=0,1,2							
offset	0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DB7							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DB6							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DB5							

位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DB4							
位权限	W-0000 0000							

Bit	助记符	功能描述
31:24	DB7	Data Byte 7
23:16	DB6	Data Byte 6
15:8	DB5	Date Byte 5
7:0	DB4	Data Byte 4

24.4.16 CAN 接收滤波寄存器（CAN_AFR）

名称	CAN_AFCR							
offset	0x60							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	UAF[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	UAF[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	UAF	消息滤波器组使能 (Use Acceptance Filter) 1：使能滤波器 0：关闭滤波器 每个 bit 对应一组消息滤波器

24.4.17 CAN 接收滤波配置寄存器（CAN_AFCFGR）

名称	CAN_AFCFGR							
offset	0x64							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	MSKM[15:8]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MSKM[7:0]							
位权限	R/W-0000 0000							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	FSC[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FSC[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	MSKM	掩码寄存器使用模式，每个 bit 对应 1 个 AFMR 0：AFMR 用作掩码 1：AFMR 用作 ID 对应关系： MSKM[15] – AFMR15 … MSKM[3] – AFMR3 MSKM[2] – AFMR2 MSKM[1] – AFMR1 MSKM[0] – AFMR0
15:0	FSC	ID 寄存器扩展模式 0：32bit 扩展模式 1：16bit 标准模式 对应关系： FSC[15] – AFIR15 FSC[3] – AFIR3 FSC[2] – AFIR2 FSC[1] – AFIR1 FSC[0] – AFIR0

24.4.18 CAN 接收滤波掩码寄存器 x（CAN_AFMRx）

名称	CAN_AFMRx (x=0~15)							
offset	0x80 + (x)*0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	AMRTR	AMID[17:11]						
位权限	R/W-0	R/W-000 0000						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	AMID[10:3]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	AMID[2:0]			AMIDE	AMSRR	AMID[28:26]		
位权限	R/W-000			R/W-0	R/W-0	R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AMID[25:18]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31	AMRTR	RTR Mask 1：RTR 位参与滤波器比较 0：RTR 位不参与滤波器比较
30:13	AMID[17:0]	ID Mask bit 17-0 1：对应位参与滤波器比较 0：对应位不参与滤波器比较
12	AMIDE	IDE Mask 1：IDE 位参与滤波器比较 0：IDE 位不参与滤波器比较
11	AMSRR	SRR Mask 1：SRR 位参与滤波器比较 0：SRR 位不参与滤波器比较
10:0	AMID[28:18]	ID Mask bit 28-18 1：对应位参与滤波器比较 0：对应位不参与滤波器比较

24.4.19 CAN 接收滤波 ID 寄存器 x（CAN_AFIRx）

使用32bit ID滤波时：

名称	CAN_AFIRx (x=0~15)							
offset	0x84 + (x)*0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	AIRTR	AIID[17:11]						
位权限	R/W-0	R/W-000 0000						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	AIID[10:3]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	AIID[2:0]			AIIDE	AISRR	AIID[28:26]		
位权限	R/W-000			R/W-0	R/W-0	R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AIID[25:18]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31	AIRTR	滤波器 RTR (Acceptance ID RTR)
30:13	AIID[17:0]	滤波器 ID bit 17-0 (Acceptance ID IDR)
12	AIIDE	滤波器 IDE (Acceptance ID IDE)
11	AISRR	滤波器 SRR (Acceptance ID SRR)
10:0	AIID[28:18]	滤波器 ID bit 28-18 (Acceptance ID IDR)

使用16bit ID滤波时：

名称	CAN_AFIRx (x=0~15)							
offset	0x68 + (x-1)*0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	AIID[17:15]			AIIDE	AIRTR	AIID[10:8]		

位权限	R/W-000			R/W-0	R/W-0	R/W-000		
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	AIID[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	AIID[17:15]			AIIDE	AIRTR	AIID[10:8]		
位权限	R/W-000			R/W-0	R/W-0	R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AIID[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	同 bit[15:0]	同 bit[15:0]
15:13	AIID[17:15]	滤波器 ID bit 17-15 (Acceptance ID IDR)
12	AIIDE	滤波器 IDE (Acceptance ID IDE)
11	AIRTR	滤波器 RTR (Acceptance ID SRR)
10:0	AIID[7:0]	滤波器 ID bit 7-0 (Acceptance ID IDR)

24.4.20 CAN 接收 FIFO ID 寄存器（CAN_RXFIDR）

名称	CAN_RXFIDR							
offset	0x100							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IDR[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	IDR[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IDR[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IDR[7:0]							
位权限	W-0000 0000							

Bit	助记符	功能描述
31:0	IDR	发送消息识别码 (Identifier Register) 数据格式参见 24.3.6 消息帧存储和消息帧结构（Message storage and structure）

24.4.21 CAN 接收 FIFO DLC 寄存器（CAN_RXFDLCR）

名称	CAN_RXFDLCR							
offset	0x104							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				DLC			
位权限	U-0				W-0000			

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	DLC	数据长度码 (Data Length Code) 定义消息帧中的数据字节长度

24.4.22 CAN 接收 FIFO DataWord1 寄存器（CAN_RXFDW1R）

名称	CAN_RXFDW1R							
offset	0x108							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DB3							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DB2							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DB1							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DB0							
位权限	W-0000 0000							

Bit	助记符	功能描述
31:24	DB3	Data Byte 3
23:16	DB2	Data Byte 2
15:8	DB1	Date Byte 1
7:0	DB0	Data Byte 0

24.4.23 CAN 接收 FIFO DataWord2 寄存器（CAN_RXFDW2R）

名称	CAN_RXFDW2R							
offset	0x10C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DB7							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DB6							

位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DB5							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DB4							
位权限	W-0000 0000							

Bit	助记符	功能描述
31:24	DB7	Data Byte 7
23:16	DB6	Data Byte 6
15:8	DB5	Date Byte 5
7:0	DB4	Data Byte 4

25 灵活速率控制器区域网络（RS-CANFD-Lite）

25.1 概述

RS-CANFD-Lite是一个单通道CAN-FD控制器，符合ISO-11898-1(2015)协议规范，针对AUTOSAR软件架构优化，支持灵活数据速率（flexible data rate）。

- 兼容ISO 11898-1 (2015)
- 支持flexile data rate
- 支持11bit和29bit ID格式
- 单通道

25.2 主要特性

item	specification
protocal	ISO 11898-1
data rate	up to 1Mbps for arbitration phase up to 8Mbps for data phase
min APB clock	80MHz
frame ID type	11-bit or 29-bit
message buffer	up to 32 reception buffers
	4 transmit buffers
	1 transmit queue
FIFO	2 reception FIFO 1 common FIFO (configurable as reception or transmit)
enhanced reception filtering	11bit or 29bit ID
	acceptance filter mask
	programmable routing capability for each FIFO and reception buffer
	RTR and IDE masking
	DLC filter
	message buffer payload overload protection
	payload filter
Timer	TX and RX time stamp

表 25-1RS-CANFD-Lite Overview

25.3 模块框图

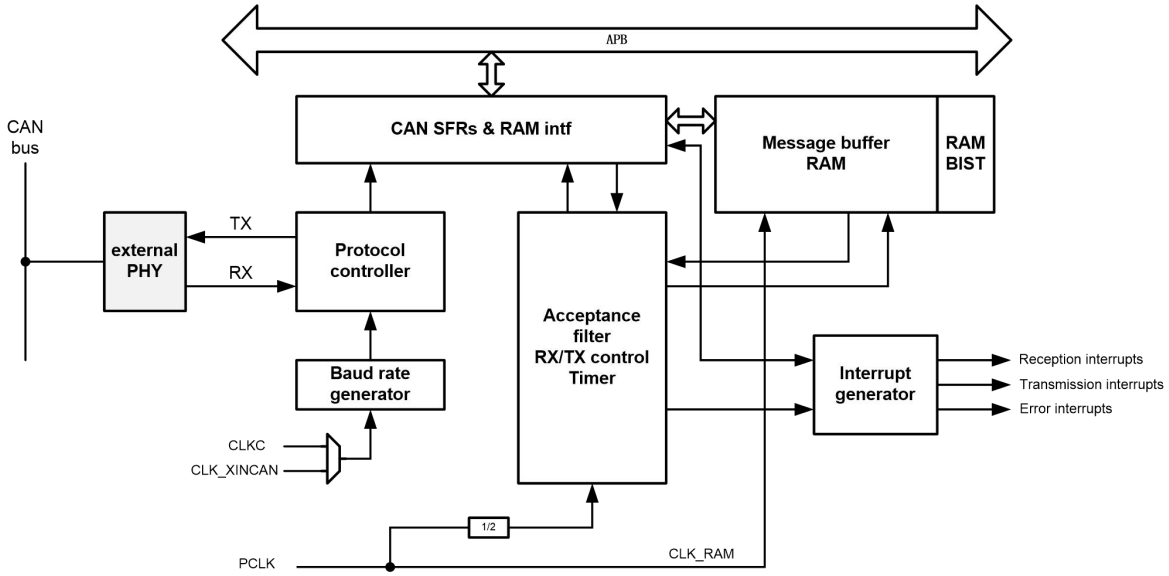


图 25- 1RS-CANFD-Lite 模块框图

RS-FDCAN-Lite包含以下主要模块：

- protocol controller：处理CAN物理层协议，包含总线仲裁、bit timing、stuffing、error handling等
- message buffer ram：发送和接收数据的消息缓存，可以用作buffer或者FIFO形式；每个消息包含独立ID、DLC (data length code)、数据域、消息指针和时间戳；ram同时用于消息滤波信息存储
- acceptance filter：对接收到的消息进行滤波处理，通过访问ram获得滤波配置
- timer：接收时间戳定时器，发送间隔定时器
- interrupt generator：产生各种全局或通道中断
- CAN SFRs：寄存器模块

25.4 时钟和复位

模块的工作时钟：

- PCLK：APB总线时钟
- CLK_RAM：RAM访问时钟
- CLKC或CLK_XINCAN（不使用）：CAN总线工作时钟，连接片内系统时钟或者直接连接晶振时钟

各个时钟之间需满足以下关系：

$$CLK_RAM/2 = PCLK \geq CLKC$$
$$CLK_RAM/2 = PCLK \geq CLK_XINCAN$$

即要求PCLK频率不低于CAN总线工作时钟，且RAM时钟频率为PCLK的2倍频。CLKC和另两个时钟之间是异步关系，而CLKRAM和PCLK为同步时钟。

- 当APBCLK=PLLx1且APBCLK=AHBCLK不分频：CLK_RAM=PLL_VCO，PCLK=APB2CLK
- 当APBCLK为其他时钟配置：CLK_RAM=AHBCLK，PCLK=APB2CLK=AHBCLK/2
- APB2CLK只支持AHBCLK的1分频或2分频

CANFD模块工作时钟组合：

CLKC	CLK_RAM	PCLK	说明
XTHF RCHF PLLx1	AHBCLK	APB2CLK=AHBCLK/2	需保证APBCLK >= CLKC
XTHF RCHF PLLx1	PLL_VCO	APB2CLK=AHBCLK	此时APBCLK选为PLL，并且APB不分频 需保证APBCLK >= CLKC
APB2CLK	AHBCLK	APB2CLK=AHBCLK/2	
APB2CLK	PLL_VCO	APB2CLK=AHBCLK	

模块的复位：

- RSTP_N：APBCLK时钟域复位，低电平用于复位CAN engine之外的所有电路，复位所有外设寄存器
- RSTC_N：CLKC时钟域复位，低电平用于复位CAN engine，不复位外设寄存器

25.5 端口

RS-CANFD-Lite模块与管脚连接的信号：

信号	功能描述	方向	连接
rxd_can	CAN总线接收数据	输入	连接到芯片管脚
can_txclk	CAN发送时钟（bit time clock）	输出	连接到芯片管脚（FOUT输出，测试目的）
can_tx_out	CAN总线发送数据	输出	连接到芯片管脚
can_tx_datarate_en	CAN-FD帧发送数据相位指示	输出	连接到芯片管脚
can_rx_datarate_en	CAN-FD帧接收数据相位指示	输出	连接到芯片管脚

表 25-2RS-CANFD-Lite 管脚连接

can_txclk 和can_tx_out之间的时序关系如下图：

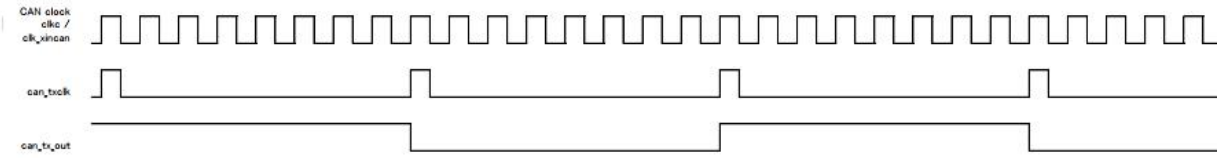


图 25- 2RS-CANFD-Lite 发送时序

25.6 工作模式

25.6.1 概述

RS-CANFD-Lite 主要有两大类工作模式：

- 全局模式
- 通道模式

25.6.2 全局模式（global modes）

全局模式包含：

- global sleep mode
- global reset mode
- global halt mode
- global operation mode

下图显示了全局模式之间的转移路径。

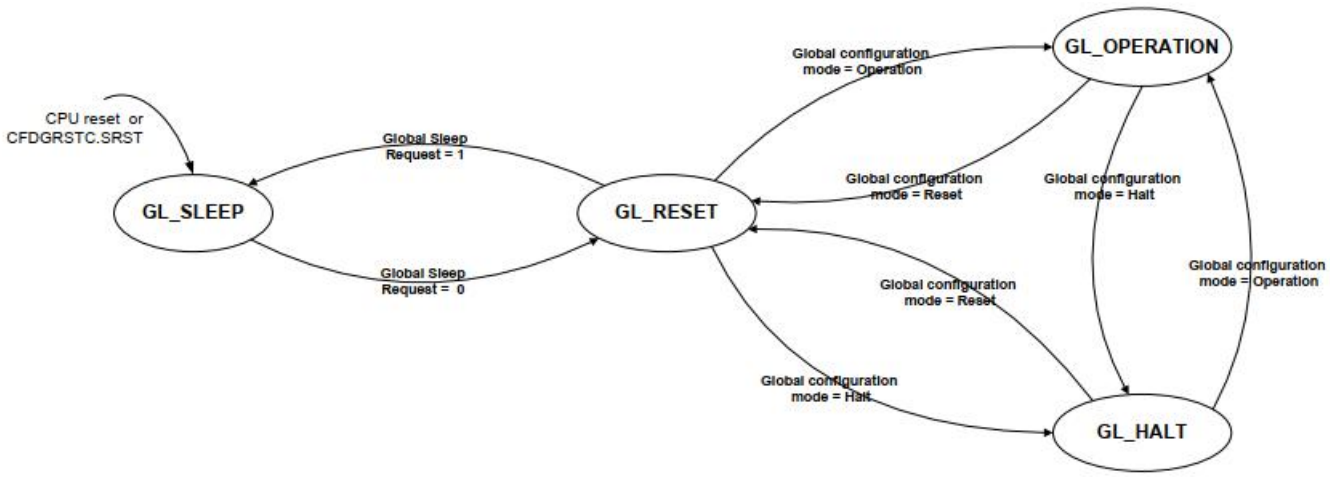


图 25-3 全局模式转移路径

24.6.2.1 Global Sleep Mode

当 RS-CANFD-Lite 模块退出复位后，自动进入 global sleep mode。

此模式下，CAN 工作时钟被门控，仅 PCLK 保持活跃，因此 CAN 模块无法通信，但是软件仍可以读写寄存器。

当 RS-CANFD-Lite 模块处于 gobal reset mode 时，软件置位 global sleep request 寄存器可以令模块进入 global sleep mode，而这个寄存器在 global halt mode 和 global operation mode 下是无效的。置位 global sleep request 寄存器使所有通道都进入 channel sleep mode。

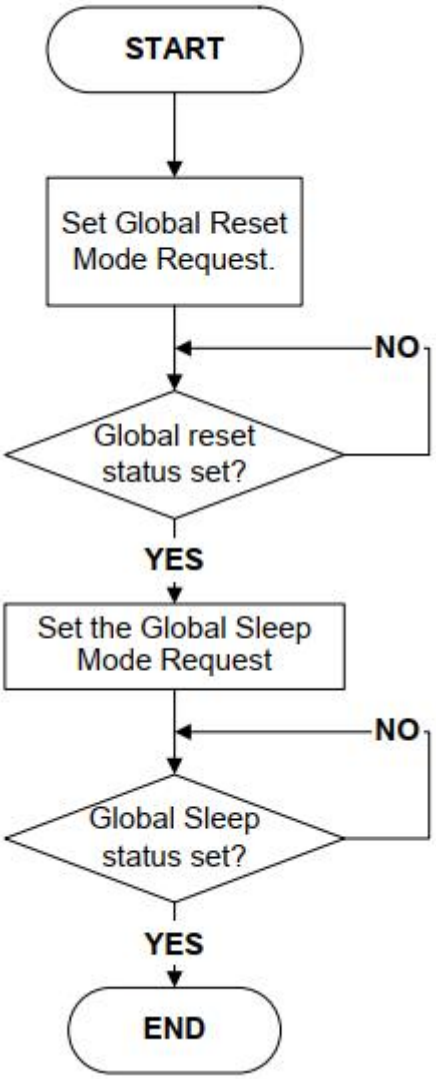


图 25-4 进入 global sleep mode 流程

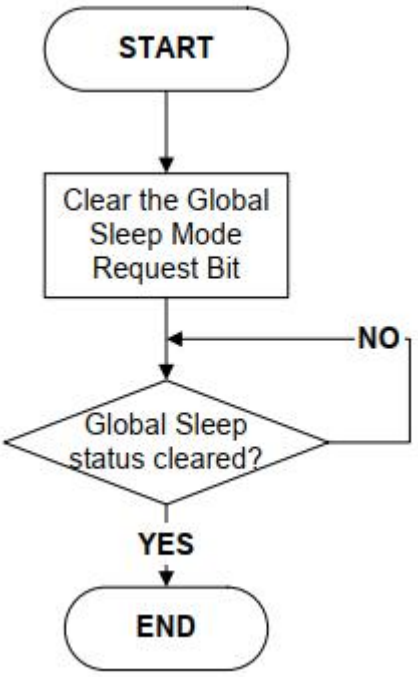


图 25-5 退出 global sleep mode 流程

24.6.2.2 Global Reset Mode

模块通过以下模式进入 global reset mode：

- 当模块处于 global halt/global operation mode 时，操作 CFDGCTR.GMDC 寄存器
- 当模块处于 global sleep mode 时，清零 sleep mode request 寄存器

global reset mode 下，所有功能被停止，并且左右状态和标志寄存器被初始化。所有 FIFO 和发送队列被禁止，发送控制寄存器被清零。设置 global reset mode 将强制使所有通道进入 channel reset mode。

设置 CFDGCTR.GMDC 为 reset mode 后，应检查 CFDGSTS.GRSTSTS 寄存器来确认模块成功进入了 global reset mode。

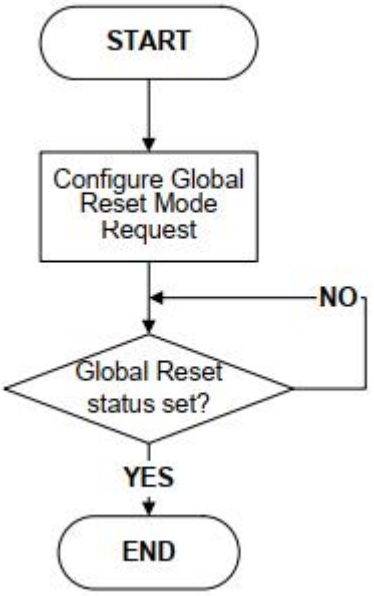


图 25-6 进入 global reset mode 流程

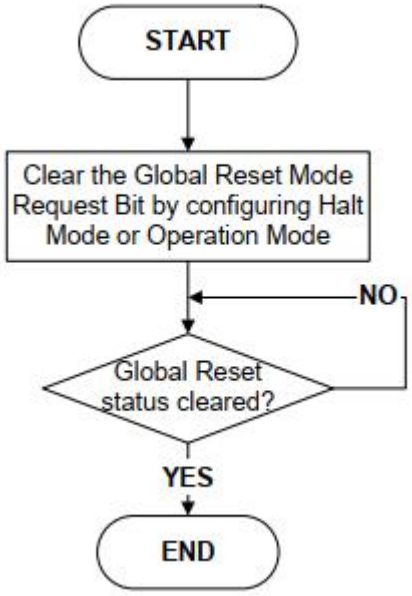


图 25-7 退出 global reset mode 流程

24.6.2.3 Global Halt Mode

模块通过操作 CFDGCTR.GMDC 寄存器进入 global halt mode。如果通道正处于 reset 或 sleep 模式则保持模式不变，如果通道正处于 operation mode 则转移到 channel halt mode。

如果当前通道正在进行通信，则模式转换延迟到通信完成再进行。如果通道正处于 bus-off 状态，则完整的 bus-off recovery 序列根据配置也可能被延迟。

进入 global halt mode 后所有通信停止，RS-CANFD-Lite 不会修改任何状态标志，如果通道处于 bus-off 则 halt mode 将会清零 REC 和 TEC 寄存器数据。

因此，global halt mode request 可以用于关闭所有 CAN 通道的通信，而不会丢失任何消息。

设置 CFDGCTR.GMDC 为 halt mode 后，应检查 CFDGSTS.GHLTSTS 寄存器来确认模块成功进入了 global halt mode。

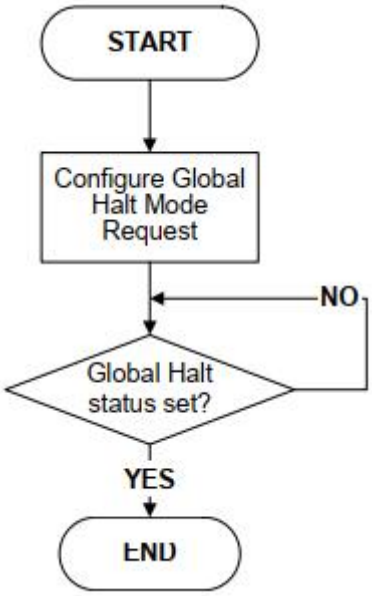


图 25-8 进入 global halt mode 流程

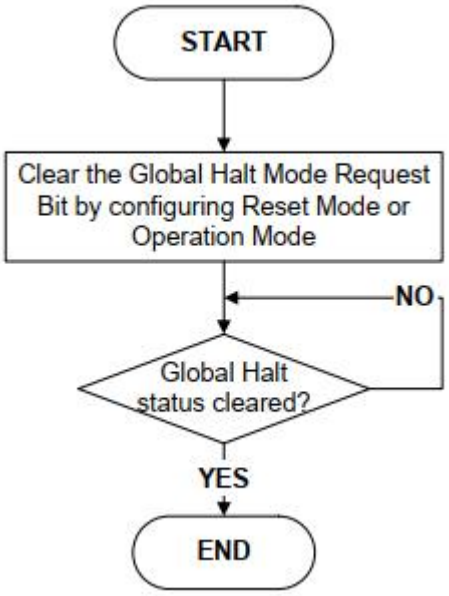


图 25-9 退出 global halt mode 流程

24.6.2.4 Global Operation Mode

所有通道的 CAN 总线通信仅能在 global operation mode 下进行。

设置 CFDGCTR.GMDC 为 operation mode 后，应检查 CFDGSTS.GHLTSTS 寄存器和

CFDGSTS.GRSTSTS 寄存器是否同时清零，来确认模块成功进入了 global operation mode。

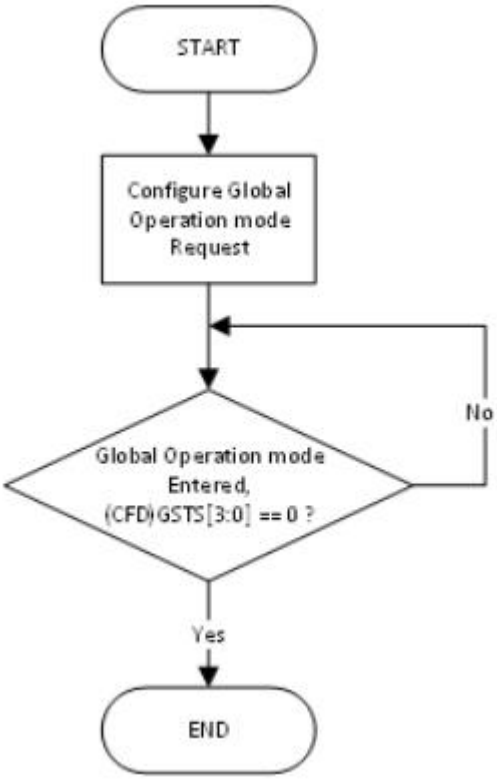


图 25-10 进入 global operation mode 流程

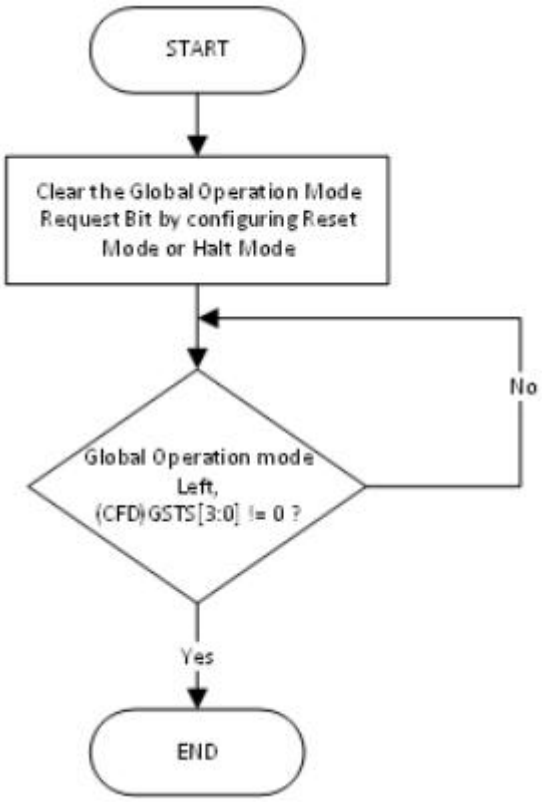


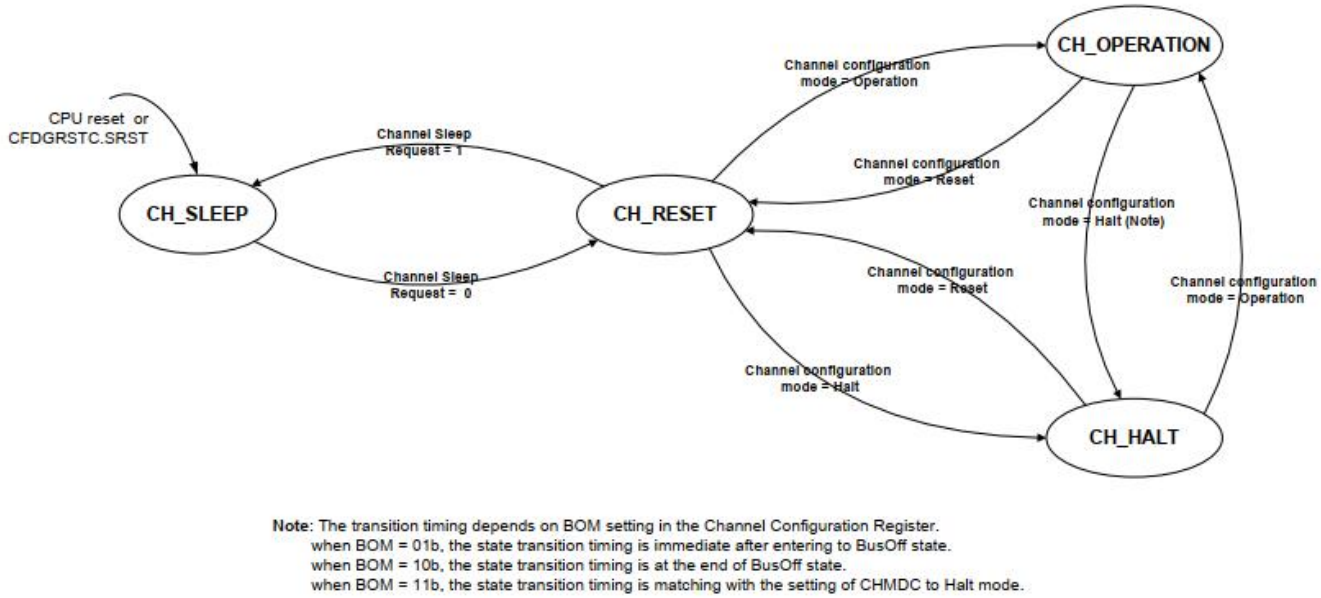
图 25-11 退出 global operation mode 流程

25.6.3 通道模式（channel modes）

通道模式包含：

- channel sleep mode
- channel reset mode
- channel halt mode
- channel operation mode

下图显示了全局模式之间的转移路径。



Note: The transition timing depends on BOM setting in the Channel Configuration Register.
when BOM = 01b, the state transition timing is immediate after entering to BusOff state.
when BOM = 10b, the state transition timing is at the end of BusOff state.
when BOM = 11b, the state transition timing is matching with the setting of CHMDC to Halt mode.

图 25-12 通道模式转移路径

24.6.3.1 channel sleep mode

当复位释放，或 RS-CANFD-Lite 进入 global sleep mode，所有通道都会进入 channel sleep mode。

当通道处于 reset mode，channel sleep mode request 寄存器被置位，对应的通道也会进入 channel sleep mode。

当通道处于 halt mode 或 operation mode 时，不应置位 channel sleep mode request 寄存器。

通道进入 sleep mode 后软件无法改写通道相关寄存器，但是仍可以读取。

24.6.3.2 channel reset mode

通过以下方式可以使通道进入 reset mode：

- 通道处于 halt 或 operation mode，配置 CFDC0CTR.CHMDC 为 reset mode
- 通道处于 sleep mode，清零 channel sleep mode request 寄存器
- 全局模式控制寄存器 CFDGCTR.GMDC 设置为 global reset mode，并且当前通道没有处于 sleep 或 reset mode

当通道进入 reset mode，所有通道相关的状态标志寄存器被硬件初始化、传输控制寄存器被复位并且通道发送队列被关闭。

通道配置寄存器不会被初始化。

24.6.3.3 channel halt mode

通过以下方式可以使通道进入 halt mode：

- 通道处于 reset 或 operation mode，配置 CFDC0CTR.CHMDC 为 halt mode

- 全局模式控制寄存器 CFDGCTR.GMDC 设置为 global halt mode，并且当前通道处于 operation mode

当通道进入 halt mode，通道相关的 CAN 通信都被暂停，但是所有状态标志寄存器保持不变，如果通道处于 bus-off 则 halt mode 将会清零 REC 和 TEC 寄存器数据。

下表罗列了当 CAN 通信正在进行的情况下，通道进入 reset 或 halt mode 时的影响：

state / mode	receiver	transmitter	bus-off
channel reset mode (CFDC0CTR.CHMDC=01)	CAN channel立即进入channel reset mode，不会等待当前接收完成	CAN channel立即进入channel reset mode，不会等待当前发送完成	CAN channel立即进入channel reset mode，不会等待bus-off recovery完成
channel halt mode (CFDC0CTR.CHMDC=10)	CAN channel等待当前接收完成后进入channel halt mode	CAN channel等待当前发送完成后进入channel halt mode	● CFDC0CTR.BOM=00：忽略halt request直到bus-off recovery完成 ● CFDC0CTR.BOM=10：等待bus-off recovery完成后进入channel halt mode ● CFDC0CTR.BOM=01：不等bus-off recovery完成，自动进入channel halt mode ● CFDC0CTR.BOM=11：当halt request置位后就进入channel halt mode，不等待bus-off recovery完成

表 25-3 进入 channel rest/halt mdoe 的行为

24.6.3.4 channel operation mode

通过配置 CFDC0CTR.CHMDC=00 激活 operation mode。进入 operation mode 后如果检测到连续 11bit 隐性位，则 CFDC0STS.COMSTS 标志被置位，通信功能被使能，RS-CANFD-Lite 成为一个活动节点，可以开始收发 CAN 消息。

当通道处于 operation mode，CAN 节点可以处于四个子状态：idle、transmit、receive、bus-off

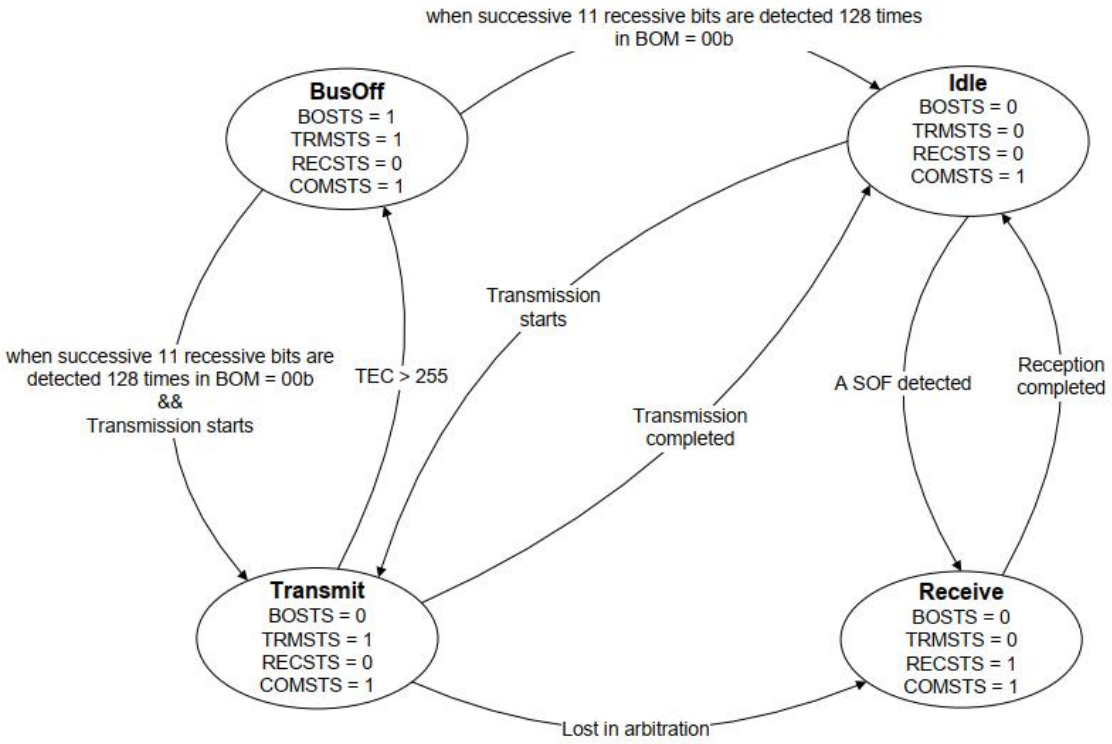


图 25-13CAN channel 收发子模式

24.6.3.5 channel bus-off

根据 ISO11898，当节点的发送错误计数器值大于 255，节点应进入总线关闭状态。

处于总线关闭状态的节点不能发送任何帧，也不能发送 ACK，对总线无影响。

总线关闭节点可以被 restart 请求重启，并且在检测到 128 个 bus idle 后返回主动错误状态，并清零发送错误计数器。

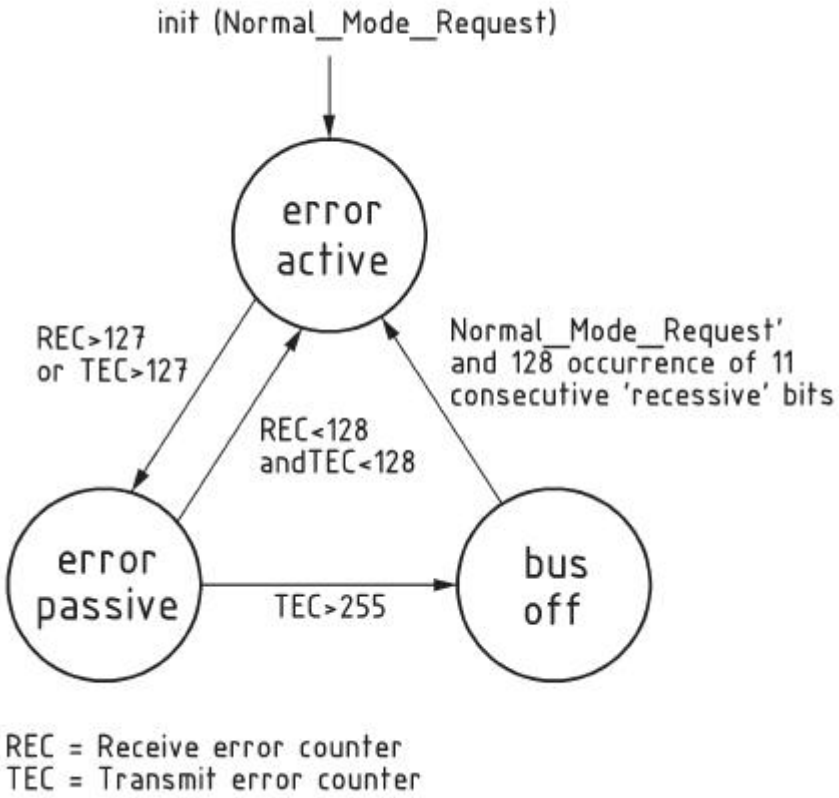


图 25-14 节点错误状态转移

根据寄存器配置，RS-CANFD-Lite 模块支持以下几种退出 bus-off 状态的方式：

- CFDC0CTR.BOM=00：

  bus-off recovery 符合 ISO11898-1，当连续检测到 128 次 11bit 隐性位，CAN 模块自动进入 error active state，TEC 和 REC 计数器自动清零，CFDC0ERFL.BORF 标志置位

- CFDC0CTR.BOM=01：

  bus-off 后立即进入 channel halt mode，CFDC0CTR.CHMDC 自动修改为 10，TEC 和 REC 计数器自动清零，CFDC0ERFL.BORF 标志不会置位

- CFDC0CTR.BOM=10：

  bus-off recovery 符合 ISO11898-1，当连续检测到 128 次 11bit 隐性位，CAN 通道自动进入 channel halt mode，TEC 和 REC 计数器自动清零，CFDC0ERFL.BORF 标志置位

- CFDC0CTR.BOM=11：

  bus-off recovery 流程可以被 channel halt mode request 打断，如果 halt request 置位则进入 channel halt mode，TEC 和 REC 计数器自动清零，CFDC0ERFL.BORF 标志不会置位；如果没有设置 CFDC0CTR.CHMDC=10 则 bus-off recovery 流程与 CFDC0CTR.BOM=00 相同

软件可以通过置位 CFDC0CTR.RTBO 寄存器，强制 CAN 通道从 bus-off 恢复，此时 CAN 节点错误状态从 bus-off 进入 integrating state，并且在检测到连续 11bit 隐性位后可以恢复正常通信，这种情

况下 BORF 标志不会置位，TEC 和 REC 会被清零。

注意在置位 RTBO 之前，所有发送 buffer、发送队列和发送模式下的 common FIFO 必须被禁止。RTBO 寄存器仅能在 CFDC0CTR.BOM=00 的配置下使用。如果 CAN 节点不处于 bus-off 状态，设置 RTBO 寄存器不会有任何作用，并且硬件将自动将其清零。

bus-off 进入标志（BOEF）和 bus-off 恢复标志（BORF）在不同 BOM 配置下的行为参见下表：

BOM	BOEF置位	BORF置位
00	always on entry to bus-off	always on exit from bus-off
00 CFDC0CTR.RTBO置位		仅当RTBO置位前退出bus-off才会置位BORF
01		从不
10		always on exit from bus-off
11		仅当软件置位halt request前退出bus-off才会置位BORF

表 25-4BOEF 和 BORF 置位条件

为了提高软件效率，RS-CANFD-Lite 可以在 bus-off 状态下进行发送初始化配置，参见下图的推荐流程：

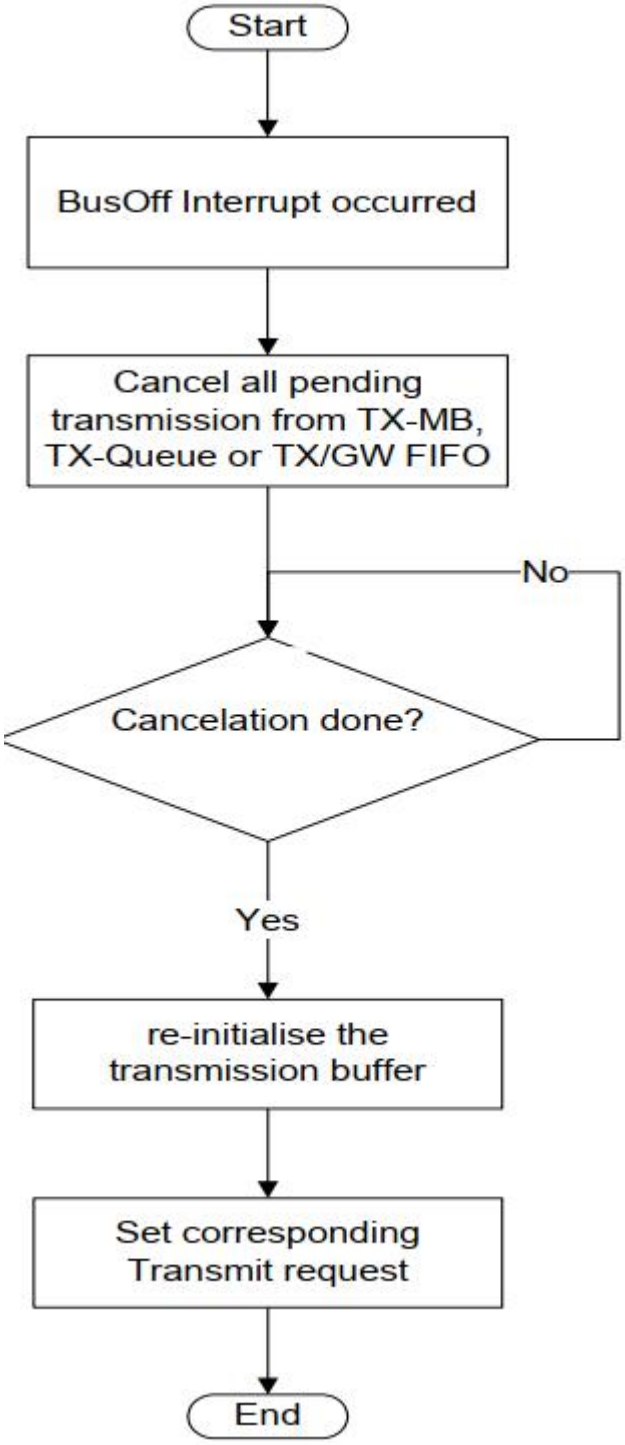


图 25- 15bus-off 状态下重新初始化发送

25.6.4 全局模式-通道模式转换交互

修改通道模式不会影响全局模式，而修改全局模式则会影响通道模式，全局模式对通道模式的影响参见下表：

Global Mode Change	Channel Mode	Channel Mode Transition Action
Sleep → Reset	Sleep	channel remains in Sleep Mode
Sleep → Halt	- (global mode change not possible)	
Sleep → Operation	- (global mode change not possible)	
Reset → Sleep	Sleep	channels remains in Sleep Mode
	Reset	channel Sleep request bit is set automatically, channel transits to Sleep Mode
Reset → Halt	Sleep	channel remains in Sleep Mode
	Reset	channel remains in Reset Mode
Reset → Operation	Sleep	channel remains in Sleep Mode
	Reset	channel remains in Reset Mode
Halt → Sleep	- (global mode change not possible)	
Halt → Reset	Sleep	channel remains in Sleep Mode
	Reset	channel remains in Reset Mode
	Halt	channel mode control is set to Reset Mode, channel transits to Reset Mode
Halt → Operation	Sleep	channel remains in Sleep Mode
	Reset	channel remains in Reset Mode
	Halt	channel remains in Halt Mode
Operation → Sleep	- (global mode change not possible)	
Operation → Reset	Sleep	channel remains in Sleep Mode
	Reset	channel remains in Reset Mode
	Halt	channel mode control is set to Reset Mode, channel transits to Reset Mode
	Operation	channel mode control is set to Reset Mode, channel transits to Reset Mode
Operation → Halt	Sleep	channel remains in Sleep Mode
	Reset	channel remains in Reset Mode
	Halt	channel remains in Halt Mode
	Operation	channel mode control is set to Halt Mode, channel transits to Halt Mode after communication finished

表 25- 5global mode – channel mode transition interactions

25.7 CAN 初始化

在 RS-CANFD-Lite 参与到总线通信之前，软件必须完成以下初始化工作：

- 时钟设置
- bit timing 设置（nominal and data rate）
- baud rate 设置（nominal and data rate）
- CAN-FD 设置
- 接收滤波器设置
- 接收、发送 buffer 和 FIFO 设置
- CAN 工作模式设置

25.7.1 时钟、bit timing 和 baud rate

24.7.1.1 bit timing

量子时间 (time quantum, tq) 定义了 CAN 总线的最小时间单位，CAN 数据 bit 长度是由 N 个量子时间定义的。CAN 模块通过对 clkc 时钟分频得到量子时间，CAN-FD 的数据位和名义位的 tq 长度可以通过 DBRP 和 NBRP 分别配置。

- 名义位 (nominal bit)

  $tq = (NBRP + 1) * t_{can_tq_clk}$

- 数据位 (data bit)

  $tq = (DBRP + 1) * t_{can_tq_clk}$

注意，只有 CAN-FD 需要支持两种 bit time，经典 CAN 只支持 nomial bit time。

Data bit time 仅用于 FD 帧的数据段，其起始点是 BRS 位采样点（如果 BRS 是隐性电平），到 CRC 分隔符结束，或者 CAN 模块发现总线错误并开始发送错误帧。

根据 ISO 11898-1，CAN bit 可以划分为 4 个段：同步段（Sync_Seg）、传输段（Prog_Seg）、相位段 1（Phase_Seg1）、相位段 2（Phase_Seg2）。

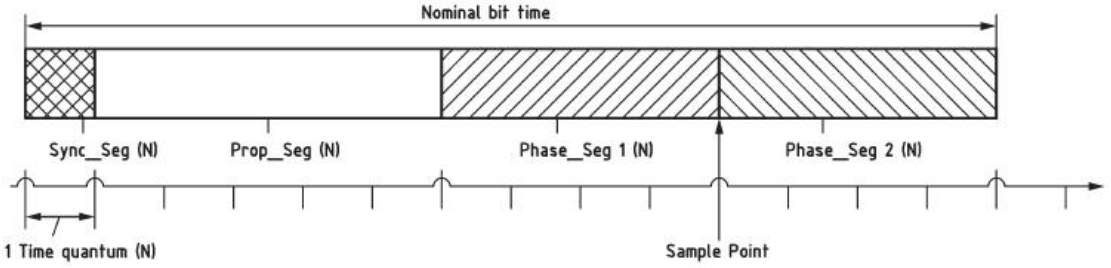


图 25-16 nominal bit timing

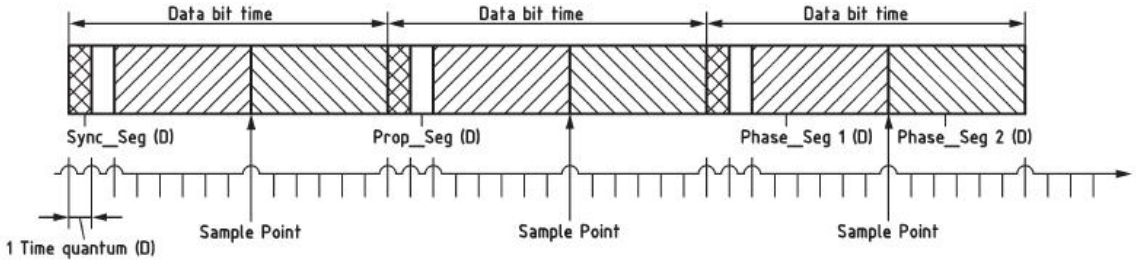


图 25-17 data bit timing

其中，同步段（SS）总是 1 个 tq 长度；而在 CAN 模块实现中，传输段和相位段 1 合并为时间段 1（TSEG1），相位段 2 构成时间段 2（TSEG2）；下图是一个 bit 的例子，采样点设置在 75%：

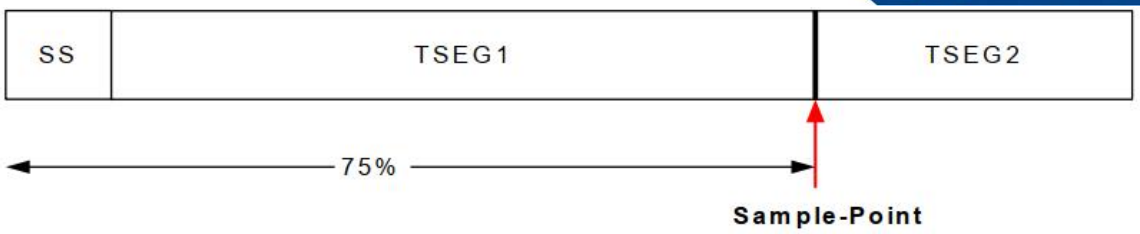


图 25-18 75%采样点设置

1) segment setting

SS = 1 Tq

TSEG1 = refer to (CFDC0NCFG) and (CFDC0DCFG)

TSEG2 = refer to (CFDC0NCFG) and (CFDC0DCFG)

SJW = refer to (CFDC0NCFG) and (CFDC0DCFG)

SS + TSEG1 + TSEG2 = 5~49 Tq for data bit, 8~385 Tq for nominal bit

2) TSEG1, TSEG2, SJW 之间的限制

TSEG1(N) > TSEG2(N) ≥ SJW(N)
TSEG1(D) ≥ TSEG2(D) ≥ SJW(D)

下表的例子说明了 bit timing 设置如何得到要求的采样点：

1 bit	Set value (TQ)				Sample point*1(%)
	SS	TSEG1	TSEG2	SJW	
5TQ	1	2	2	1	60.00
8TQ	1	4	3	1	62.50
	1	5	2	1	75.00
10TQ	1	6	3	1	70.00
	1	7	2	1	80.00
12TQ	1	8	3	1	75.00
	1	9	2	1	83.33
15TQ	1	10	4	1	73.33
	1	11	3	1	80.00
16TQ	1	10	5	1	68.75
	1	11	4	1	75.00
20TQ	1	12	7	1	65.00
	1	13	6	1	70.00
24TQ	1	15	8	1	66.66
	1	16	7	1	70.83
50TQ	1	39	10	4	80.00

表 25-6bit timing examples

采样点 (bit sample point)

bit 采样位置在 TSEG1 的结束点处，因此通过调整 TSEG1、TSEG2 的设置，可以调整 bit 内部采样点位置，以实现总线相位补偿。

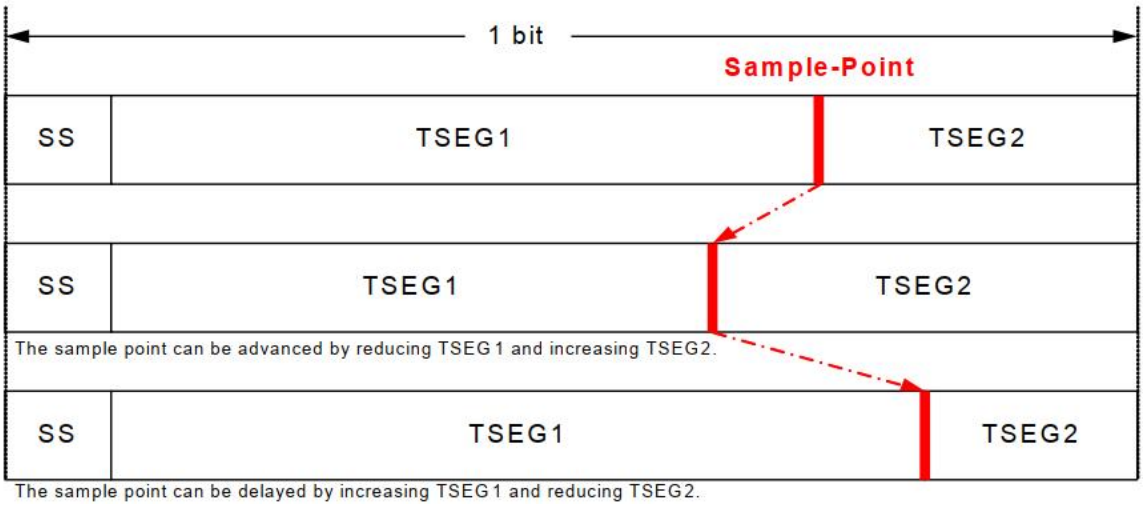


图 25-19 通过 TSEG 调整补偿采样点

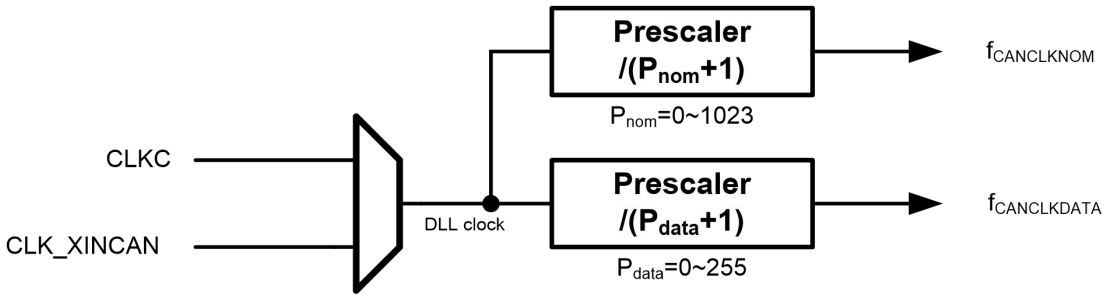
同步跳跃宽度 (Synchronization Jump Width, SJW)

受重同步的影响，bit 内的 TSEG1、TSEG2 长度可能发生变化。SJW 定义了每个 bit 内 SEG 长度改变的上限，格式是 tq 的个数。

24.7.1.2 baud rate

通信波特率由CAN工作时钟预分频、bit内部Tq数量，共同决定。

P_{Nom}和P_{Data}分别表示nominal和data bit的通信时钟预分频系数，通过CFDC0NCFG和CFDC0DCFG寄存器配置，对CAN工作时钟（CLKC或者CLK_XINCAN，二选一得到data link layer clock）分频后得到通信时钟：



下表是nominal波特率计算的例子，括号内表示分频系数 (P_{Nom} + 1)：

Baud rate calculation formula	(DLL Clock) (baud rate prescaler divide-by-N value[*1]) x (number of TQs in one bit)								
	80MHz	40MHz	32MHz	30MHz	24MHz	20MHz	16MHz	10MHz	8MHz[*3]
1Mbps	8TQ (10) 20TQ (4)	8TQ (5) 20TQ (2)	8TQ (4) 16TQ (2)	10TQ (3) 15TQ (2)	8TQ (3) 12TQ (2) 24TQ (1)	10TQ (2) 20TQ (1)	8TQ (2) 16TQ (1)	10TQ (1)	8TQ (1)
500Kbps	8TQ (20) 20TQ (8)	8TQ (10) 20TQ (4)	8TQ (8) 16TQ (4)	10TQ (6) 15TQ (4) 20TQ (3)	8TQ (6) 12TQ (4) 24TQ (2)	10TQ (4) 20TQ (2)	8TQ (4) 16TQ (2)	10TQ (2) 20TQ (1)	8TQ (2) 16TQ (1)
250Kbps	8TQ (40) 20TQ (16)	8TQ (20) 20TQ (8)	8TQ (16) 16TQ (8)	10TQ (12) 15TQ (8) 20TQ (6)	8TQ (12) 12TQ (8) 24TQ (4)	10TQ (8) 20TQ (4)	8TQ (8) 16TQ (4)	10TQ (4) 20TQ (2)	8TQ (4) 16TQ (2)
125Kbps	8TQ (80) 20TQ (32)	8TQ (40) 20TQ (16)	8TQ (32) 16TQ (16)	10TQ (24) 15TQ (16) 20TQ (12)	8TQ (24) 12TQ (16) 24TQ (8)	10TQ (16) 20TQ (8)	8TQ (16) 16TQ (8)	10TQ (8) 20TQ (4)	8TQ (8) 16TQ (4)
83.3Kbps	8TQ (120) 12TQ (80) 16TQ (60) 24TQ (40)	8TQ (60) 12TQ (40) 16TQ (30) 24TQ (20)	8TQ (48) 12TQ (32) 16TQ (24) 24TQ (16)	8TQ (45) 10TQ (36) 12TQ (30) 15TQ (24) 20TQ (18) 24TQ (15)	8TQ (36) 12TQ (24) 16TQ (18) 24TQ (12)	8TQ (30) 10TQ (24) 12TQ (20) 15TQ (16) 16TQ (15) 20TQ (12) 24TQ (10)	8TQ (24) 12TQ (16) 16TQ (12) 24TQ (8)	8TQ (15) 10TQ (12) 12TQ (10) 15TQ (8) 20TQ (6) 24TQ (5)	8TQ (12)
33.3Kbps	8TQ (300) 12TQ (200) 16TQ (150) 20TQ (120) 24TQ (100)	8TQ (150) 12TQ (100) 16TQ (75) 20TQ (60) 24TQ (50)	8TQ (120) 10TQ (96) 12TQ (80) 15TQ (64) 16TQ (60) 20TQ (48) 24TQ (40)	10TQ (90) 12TQ (75) 15TQ (60) 20TQ (45)	8TQ (90) 10TQ (72) 12TQ (60) 15TQ (48) 16TQ (45) 20TQ (36) 24TQ (30)	8TQ (75) 10TQ (60) 12TQ (50) 15TQ (40) 20TQ (30) 24TQ (25)	8TQ (60) 10TQ (48) 12TQ (40) 15TQ (32) 16TQ (30) 20TQ (24) 24TQ (20)	10TQ (30) 12TQ (25) 15TQ (20) 20TQ (15)	8TQ (30)

表 25-7nominal baud rate example

Baud rate calculation formula	(DLL Clock) (baud rate prescaler divide-by-N value[*1]) x (number of TQs in one bit)		
	80MHz	40MHz	20MHz
Nominal 1Mbps Data 8Mbps	80TQ (1) 10TQ (1)	40TQ (1) 5TQ (1)	20TQ (1) Not possible
Nominal 1Mbps Data 5Mbps	80TQ (1) 16TQ (1)	40TQ (1) 8TQ (1)	20TQ (1) Not possible
Nominal 500Kbps Data 2Mbps	160TQ (1) 40TQ (1)	80TQ (1) 20TQ (1)	40TQ (1) 10TQ (1)

表 25-8nominal and data rate example

根据上表计算可知，要支持5Mbps以上的FD数据传输，必须使用40MHz以上的CAN工作时钟。

24.7.2 发送延迟补偿（TDC）

发送延迟补偿仅适用于 CAN-FD，经典 CAN 不需要支持此特性。

为了提升通信可靠性，CAN 总线规定了发送节点必须支持总线回读机制，即 CAN 总线上的任意节点在发送数据的同时，必须同时从接收端监视总线数据。在报文帧的仲裁段，如果总线实时数据电平与所发送的数据不一致，则表示仲裁失败；在数据段，总线电平与发送电平不一致表示总线错误。

由于数据在总线上的传播不可避免的将引入发送数据和接收数据之间的延迟，对于 FD 应用，当通信波特率提高时，相位延迟可能导致发送节点数据比对错误，这种情况下的总线传播延迟将限制 FD 应用的最高波特率。

为了解决这个问题，CAN-FD 引入了发送延迟补偿机制，此功能仅用于 FD 帧的数据段。发送延迟补偿机制定义了第二采样点 SSP（Secondary Sample Point），SSP 滞后于标准采样点，发送节点

在 SSP 采样并比对数据，从而补偿了总线延迟带来的影响。

下图是 SSP 的示意图。

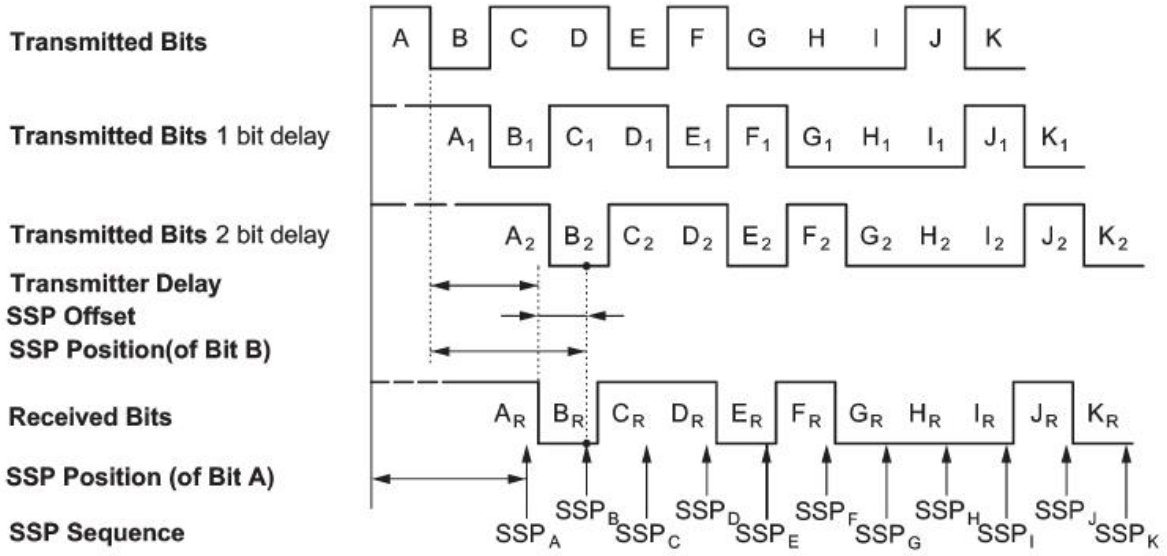


图 25-20 FD 帧数据段第二采样点 SSP

上图的例子中，发送节点看到的总线数据滞后于发送信号大约 2 个 bit 时间，直接在采样点比对数据将导致总线错误。CAN 模块首先应测量传播延迟，然后通过调整 SSP offset，获得理想的 SSP 位置，从而保证总线回读正确。

SSP 的位置由以下两部分组成：

- 总线传播延迟（Trv_Delay）
- SSP offset（TDCO）

CAN 模块通过测量 FDF 位和 res 位之间的下降沿，在 TX 引脚和 RX 引脚上出现的时间差，来计算发送传播延迟。这个延迟的测量单位是 mtq（最小 quantum 时间，即 1 个 DLL 时钟周期），Trv_Delay 最大计数值是 127。

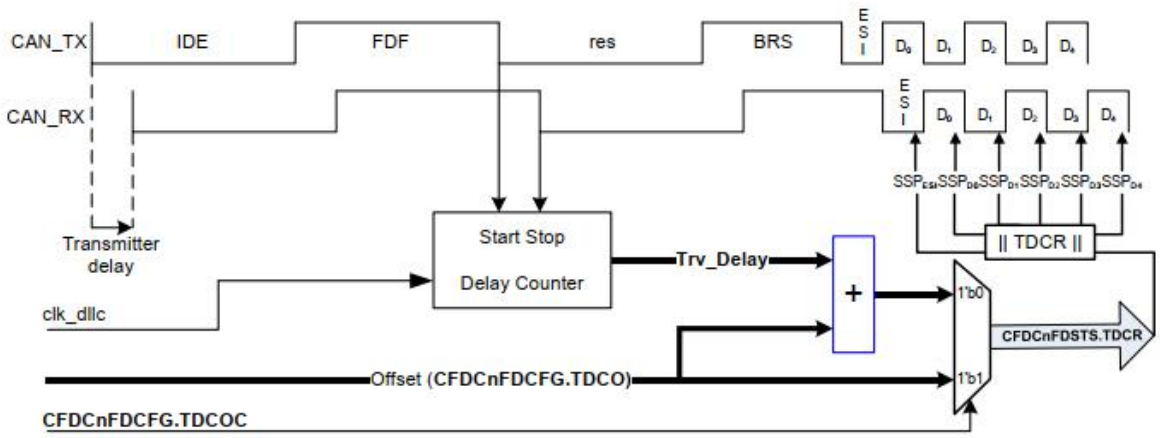


图 25-21 FD 帧发送延迟补偿

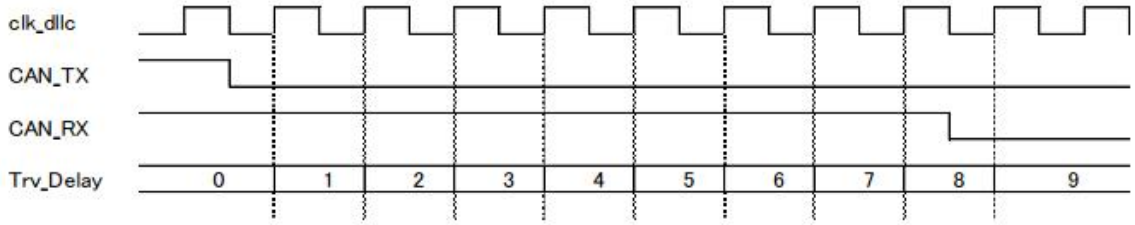


图 25-22 Trv_Delay 测量举例

CFDC0FDCFG.TDCO 寄存器用于定义 SSP offset，SSP 的具体位置定义：

- CFDC0FDCFG.TDCOC=0：SSP = Trv_Delay+CFDC0FDCFG.TDCO
- CFDC0FDCFG.TDCOC=1：SSP = CFDC0FDCFG.TDCO

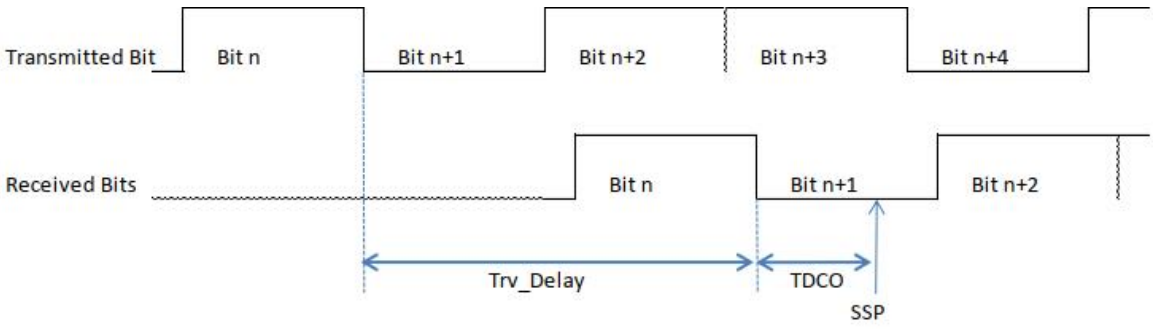


图 25-23 SSP 位置计算

RS-CANFD-Lite 能够补偿的最大相位延迟是（6 data bits – 2 clk_dlc）

24.7.3 复位后的 CAN 模块初始化

系统复位释放后，或者CFDGRSTC.SRST模块复位清零后，RS-CANFD-Lite模块自动进入global sleep mode。

复位释放后CAN模块自动执行RAM初始化，此时CFDGSTS.GRAMINIT标志置位，当初始化完成后此标志寄存器自动清零。软件应等待RAM初始化完成后再读写模块寄存器。注意CFDGRSTC.SRST软复位不会使CAN模块执行RAM初始化。

在开始通信前，全局消息滤波列表和消息FIFO buffer必须被配置，每个CAN通道需要配置bit timing和baud rate等时序信息。配置前需要退出global sleep mode和channel sleep mode，配置流程图参见下图

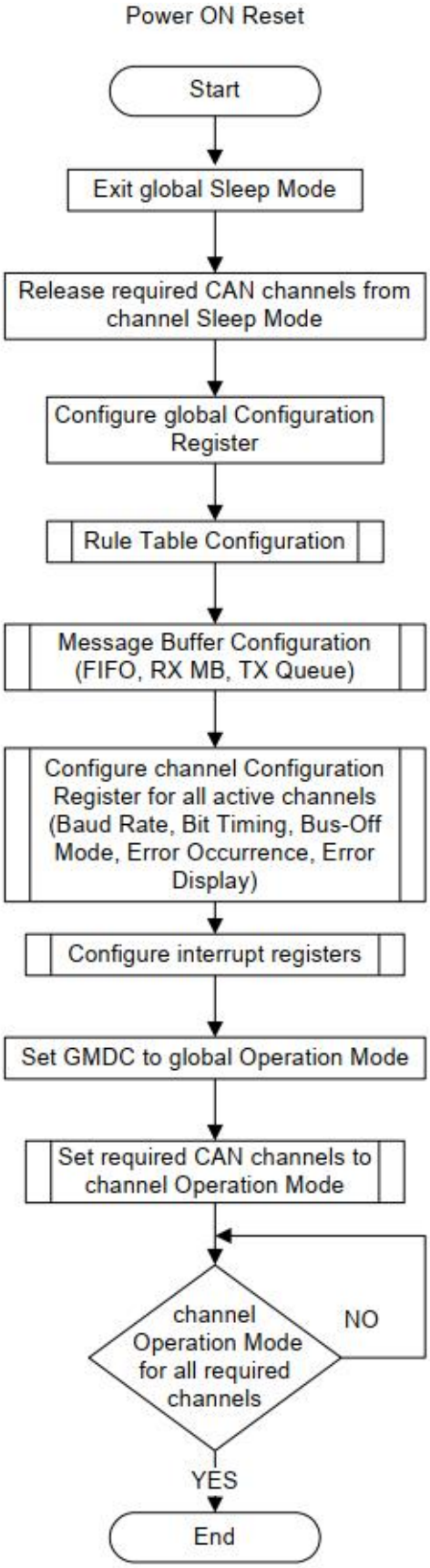


图 25- 24 硬件复位后的配置流程

25.8 消息滤波和全局滤波列表（Acceptance Filter List – AFL）

24.8.1 Overview

RS-CANFD-Lite通过AFL管理消息滤波器，AFL中的每个元素（element）定义一个滤波规则，能够定义以下行为：

- 基于ID和mask和接收滤波
- 基于DLC值的接收滤波
- 基于消息数据的接收滤波
- 根据AFL配置定义通过滤波的消息的存储地点
- 为保存的消息添加16bit指针，以支持AUTOSAR应用
- 为保存的消息添加2bit消息标签

RS-CANFD-Lite模块支持最大32个AFL元素。

当一帧消息被接收后，CAN模块自动执行AFL搜索，搜索从最小编号的元素开始，如果消息与某个AFL元素匹配，则搜索停止，如果所有的元素都不匹配，消息被丢弃；被丢弃的消息不会产生任何中断标志。

消息存储目标位置信息也保存在AFL中，用户可以配置通过滤波的消息保存在什么位置。每个消息可以存入最多两个目标位置，比如接收消息buffer和/或FIFO，配置更多的目标位置可能导致无法正常工作，应用程序需要确保配置正确。

24.8.2 DLC 滤波功能

如果DLC滤波使能：

- 如果接收消息的DLC值大于或等于AFL元素中配置的DLC，DLC滤波pass
- 如果接收消息的DLC值小于AFL中配置的DLC，则消息被丢弃，不会保存到RXMB或FIFO中
- DLC滤波失败将触发DLC error flag

如果DLC替换功能：

- DLC替换使能（CFDGCFG.DRE=1），AFL中的DLC>0且DLC滤波通过，此时接收消息中的DLC将被替换成AFL中配置的DLC并与消息一同存入RXMB或FIFO
- DLC替换使能（CFDGCFG.DRE=1），如果接收到的DLC大于AFL中的配置，则多余的数据字

节不会存入RXMB或FIFO，而代替存入0x00

- DLC替换禁止（CFDGCFG.DRE=0），如果接收到的DLC大于AFL中的配置，则多余的数据字节也存入RXMB或FIFO
- DLC替换使能（CFDGCFG.DRE=1），AFL中的DLC=0，则接收到的DLC将存入RXMB或FIFO

24.8.3 AFL 描述

每个AFL元素包含16字节，保存在RAM中，软件通过SFR寄存器映射来访问。最大支持32个AFL元素，因此最多占据512字节SRAM空间。

每个AFL element包含以下数据：

- 消息ID
- IDE bit
- RTR bit
- Loopback configuration bit
- ID mask
- IDE mask
- RTR mask
- 16bits pointer
- 2bits information label
- DLC value
- 目标存储buffer和使能位
- 目标FIFO指针

数据组织请参见寄存器描述。

AFL的访问通过以下寄存器实现：

- Global AFL ID Entry Register:Part 1 of the AFL entry
- Global AFL Mask Entry Register:Part 2 of the AFL entry
- Global AFL Pointer 0 Entry Register:Part 3 of the AFL entry
- Global AFL Pointer 1 Entry Register:Part 4 of the AFL entry

16个AFL元素组成一个页，全部AFL需要以分页形式访问，其中page0包含entry 0~15，page1包含entry16~31，通过CFDGAFLECTR寄存器选择访问页，并且使能或者禁止对AFL的访问。

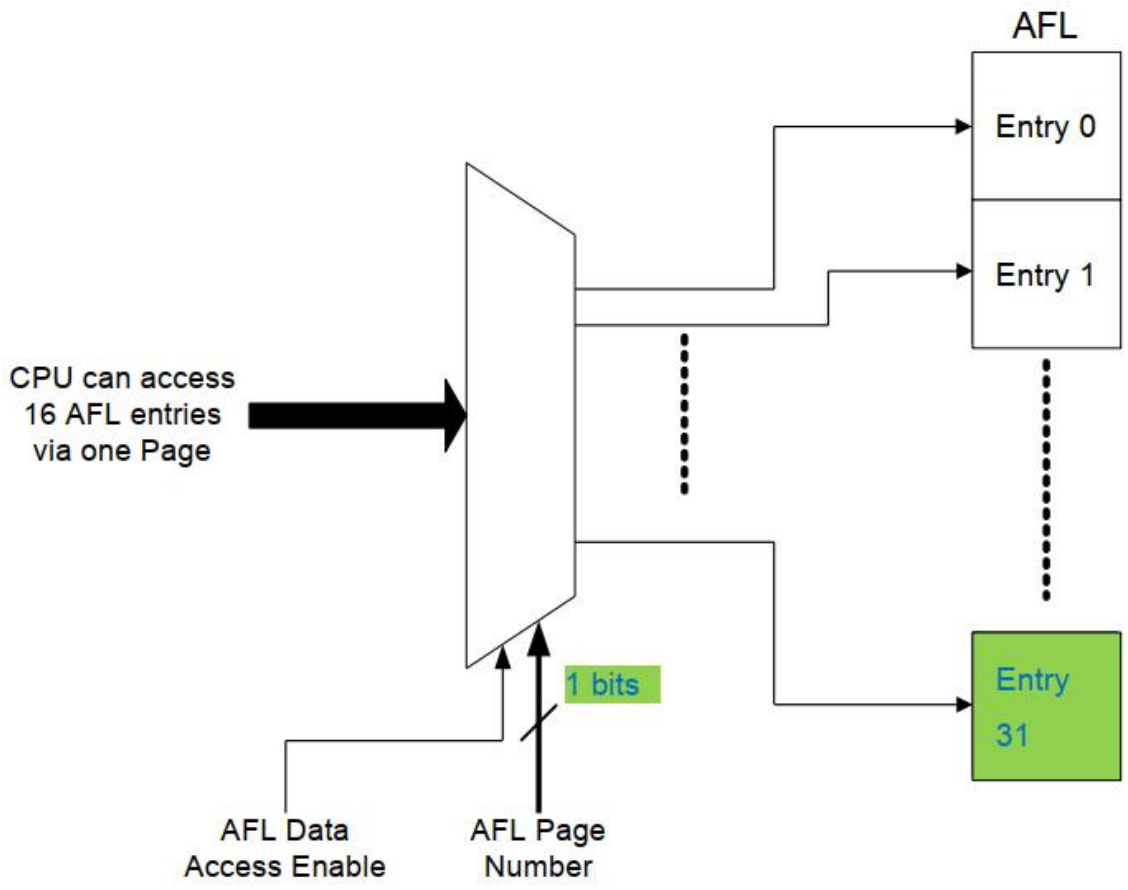


图 25-25 AFL 的分页访问

软件应按照下图流程输入AFL，当AFL编程完成，应关闭AFL访问使能，以避免非预期的改写AFL；禁止访问使能后，软件仍可以读取AFL用于实时数据检查，但是不能改写。注意AFL仅能在channel reset或者channel halt模式下改写。

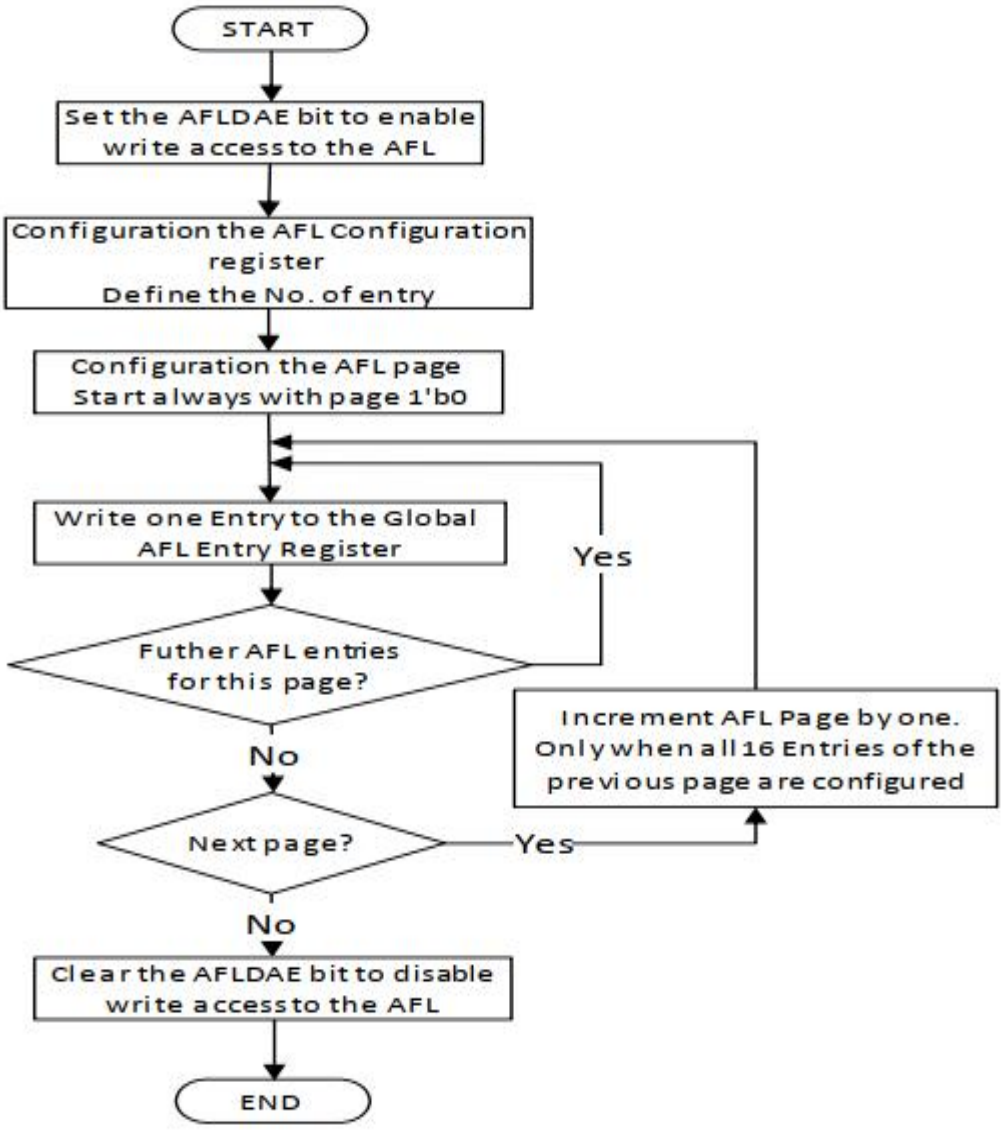


图 25- 26 AFL 配置流程

24.8.4 环回模式（Loopback mode）

当AFL元素中的loopback configuration bit置位，则当前AFL entry仅在环回测试模式、或mirror模式下有效，此时这个AFL entry对外部CAN节点发送的消息不起作用。

当CFDGCFG.MME=1，则使能mirror模式，此时RS-CANFD-Lite发送的消息也会被使能了loopback的AFL entry检查，如果匹配则发送消息同时被存入RXMB或FIFO。

Mirror mode enable (MME configuration bit)	Loopback in test mode (Selftest Mode 0 or Selftest Mode 1)	Channel Mode is	Loopback configuration bit in AFL Entry	AFL entry is
0	0	Receiver	0	valid
			1	invalid
		Transmitter	0	invalid
			1	invalid
	1	Receiver	0	valid
			1	invalid
		Transmitter	0	valid
			1	valid
1	0	Receiver	0	valid
			1	invalid
		Transmitter	0	invalid
			1	valid
	1	Receiver	0	valid
			1	invalid
		Transmitter	0	valid
			1	valid

表 25-9AFL 行为与 MME 和 loopback configuration 的关系

24.8.5 IDE masking

AFL entry中的GAFLIDEM用于配制AFL entry中的IDE bit是否用于ID匹配。当IDEM=0，根据接收到的消息中的IDE位来决定使用ID[10:0]或ID[28:0]做ID匹配。

举例来说，如果某个entry中的ID和mask配置为：

CFDGAFLID [x] = C0553A20h → IDE = 1, RTR = 1, LLB = 0, ID[10:0] = 220h / ID[28:0] = 0553A20h

CFDGAFLMr = 0000FFFFh → IDEM = 0. RTRM = 0, IDM[10:0] = 7FFh / IDM[28:0] = 0000FFFFh

则以下4个消息的滤波结果为：

- IDE=0 & ID=220，匹配成功
- IDE=0 & ID=320，匹配失败
- IDE=1 & ID=1FFF3A20，匹配成功
- IDE=1 & ID=08803220，匹配失败

24.8.6 通信过程中更新 AFL entry

用户可以在通信过程中更新AFL entry，通过entry ignore enable寄存器暂时忽略某个entry，然后更新这个entry的配置参数：

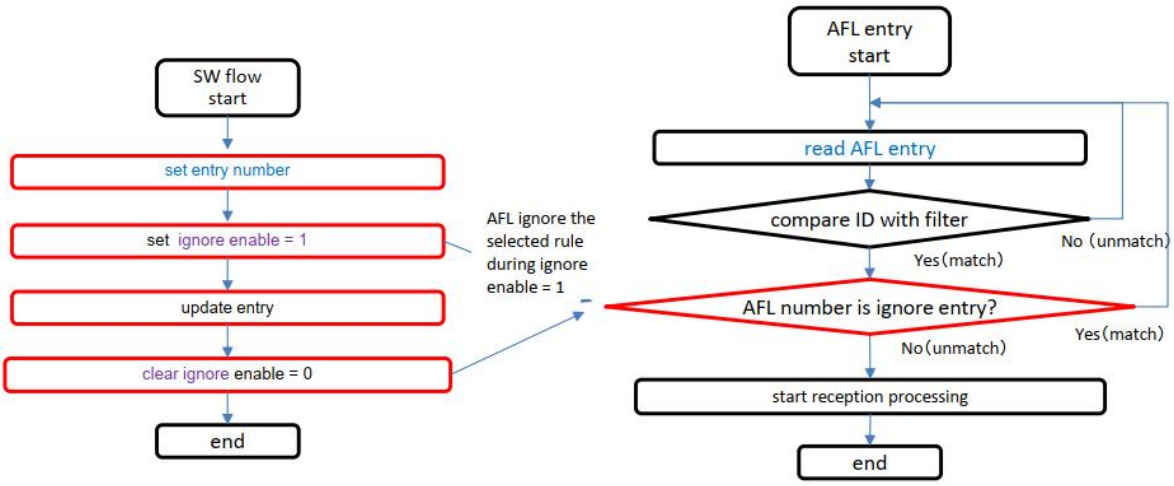
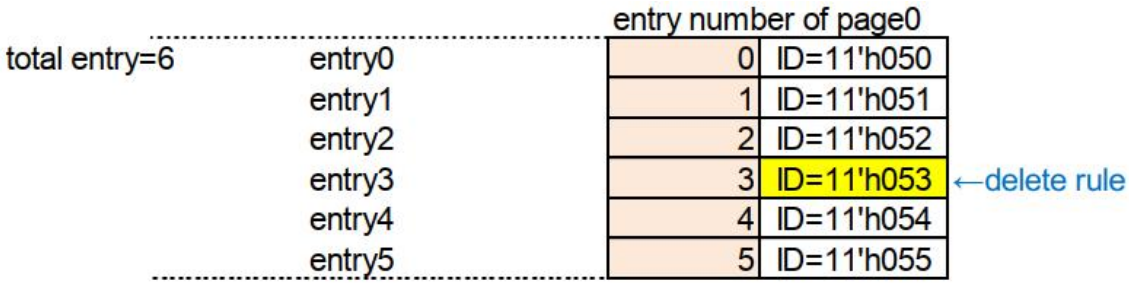


图 25- 27 AFL entry 更新流程

软件操作流程如下：

1) 配置CFDGAFLIGNENT寄存器，设置entry number
2) 向CFDGAFLIGNCTR寄存器写入16'hC401 (key code& enable bit)
3) 通过CFDGAFLECTR寄存器设置entry page
4) 通过CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r寄存器设置新的entry规则
5) 清零CFDGAFLECTR.AFLDAE
6) 向CFDGAFLIGNCTR寄存器写入16'hC400 (key code& enable bit)

应用举例（1）：删除某个entry



1. Set 32'h0000_0003 to CFDGAFLIGNENT register.
   (2) Set 32'h0000_c401 to CFDGAFLIGNCTR register
   (3) Set 32'h0000_0100 to CFDGAFLECTR register
   (4) Set same rule as previous rule by accessing to CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r (r=3, this is entry3)
   (5) Set 32'h0000_0000 to CFDGAFLECTR register
   (6) Set 32'h0000_c400 to CFDGAFLIGNCTR register

   操作完成后，当前AFL配置变成：

		entry number of page0	
total entry=5	entry0	0	ID=11'h050
entry2=entry3	entry1	1	ID=11'h051
	entry2	2	ID=11'h052
	entry3	3	ID=11'h052 ←set rule same as previous rule
	entry4	4	ID=11'h054
	entry5	5	ID=11'h055

应用举例（2）：增加1个entry

		entry number of page0	
total entry=5	entry0	0	ID=11'h050
entry2=entry3	entry1	1	ID=11'h051
	entry2	2	ID=11'h052
	entry3	3	ID=11'h052 ←add new rule in this position
	entry4	4	ID=11'h054
	entry5	5	ID=11'h055

2. Set 32'h0000_0003 to CFDGAFLIGNENT register.

(2) Set 32'h0000_c401 to CFDGAFLIGNCTR register

(3) Set 32'h0000_0100 to CFDGAFLECTR register

(4) Set new rule by accessing to CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r (r=3, this is entry3)

(5) Set 32'h0000_0000 to CFDGAFLECTR register

(6) Set 32'h0000_c400 to CFDGAFLIGNCTR register

操作完成后，当前AFL配置变成：

		entry number of page0	
total entry=6	entry0	0	ID=11'h050
	entry1	1	ID=11'h051
	entry2	2	ID=11'h052
	entry3	3	ID=11'h056 ←add new rule
	entry4	4	ID=11'h054
	entry5	5	ID=11'h055

25.9 FIFO 和 Message buffer 配置

24.9.1 Overview

RS-CANFD-Lite包含RX Message Buffer（RXMB），TX Message Buffer（TXMB），RX FIFO Buffer，common FIFO Buffer（可配置为接收或发送模式）。其中RXMB通过RXMB寄存器访问，TXMB通过TXMB寄存器访问，FIFO则通过FIFO access寄存器访问。如果common FIFO配置为发送模式，则用户只能通过FIFO access寄存器向FIFO写入数据，如果common FIFO配置为接收模式，则用户只能通过FIFO access寄存器从FIFO读取数据。

Message buffer和FIFO都通过SRAM实现，未使用的buffer地址读出为无效随机值。

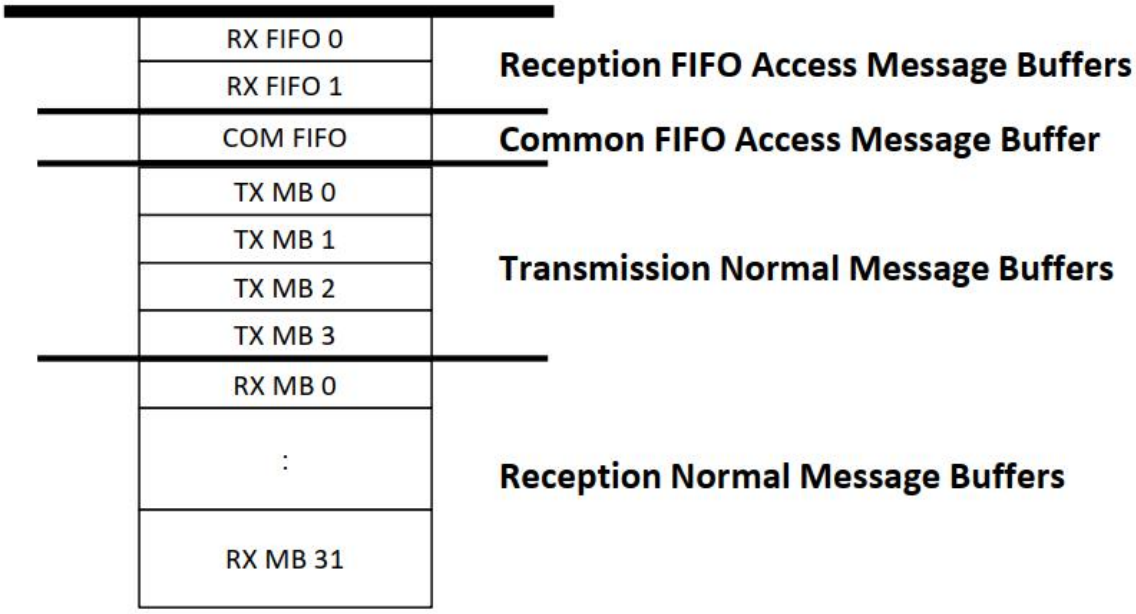


图 25-28 Message Buffer 和 FIFO mapping

24.9.2 RX Message Buffers (RXMB)

通过AFL entry设置，可以将接收到的消息存入指定的RXMB。

通过RXMB number寄存器可以配置RXMB的个数，最小值为0，最大值32，即支持最少0个RXMB和最大32个RXMB。

RXMB中的数据长度可以通过CFDRMNB.RMPLS寄存器配置，默认值为8字节，最大64字节。如果接收的消息数据长度大于CFDRMNB.RMPLS配置，则根据CFDGCFG.CMPOC寄存器配置可以拒收消息或者截取数据。

注意，AFL entry中的RXMB配置没有合法性检查机制，软件应保证AFL配置的正确性。

24.9.3 FIFO Buffers

RX FIFO个数固定为2，仅能用于消息接收；common FIFO固定为1个，可用于接收或者发送。

每个FIFO可以独立使能或禁止，FIFO的深度、地址、中断、overwrite机制也都可以通过寄存器配置。

24.9.3.1 FIFO 配置过程

软件通过RX FIFO configuration/control register和Common FIFO configuration/control register来配置FIFO，流程如下：

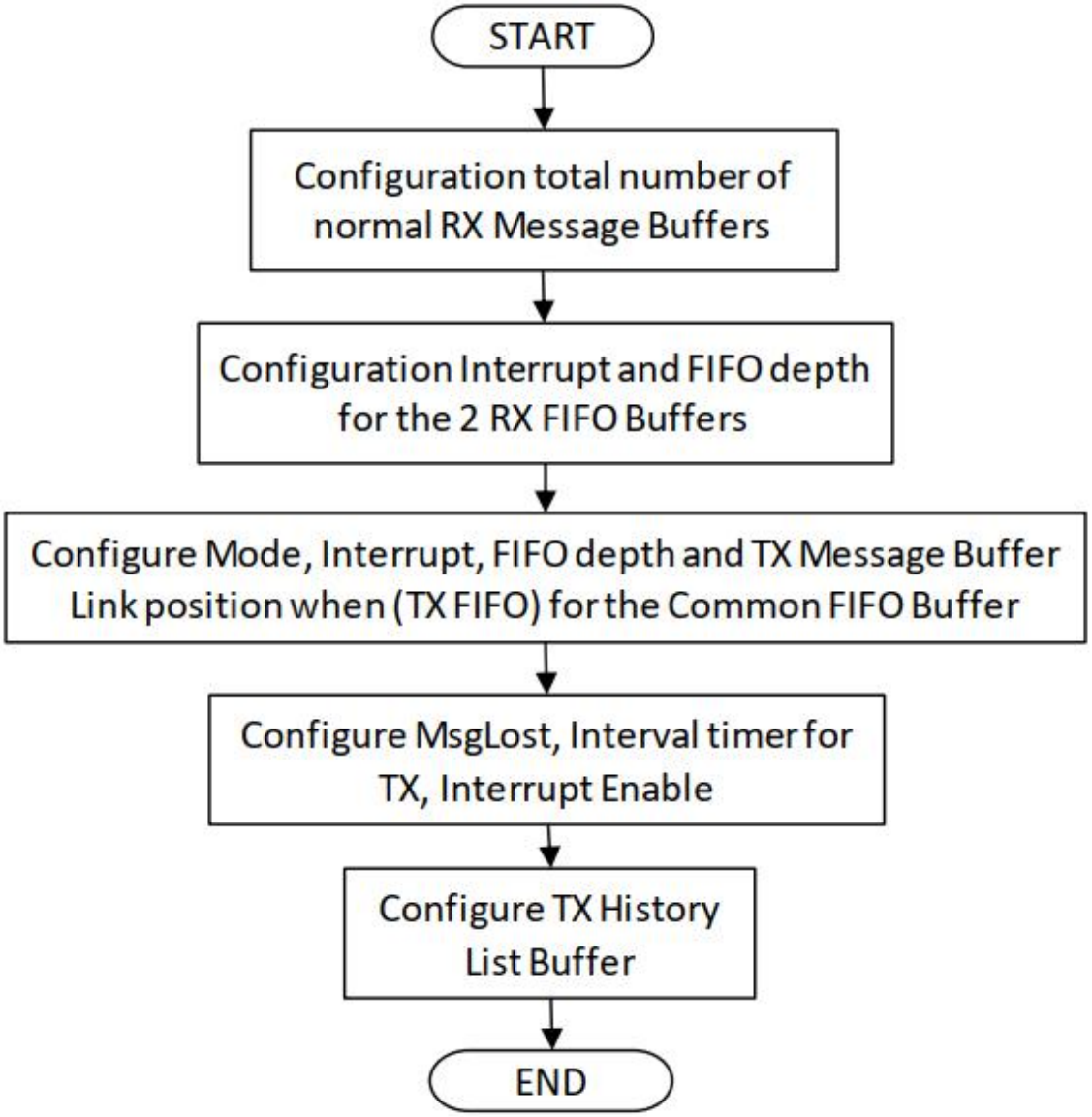


图 25-29 FIFO 配置流程

复位后common FIFO默认为接收模式。

24.9.3.2 FIFO TXMB link

的那个common FIFO被配置为发送模式，必须将FIFO链接到某个发送buffer上，这样才能使common FIFO参与到发送扫描过程中。

某个TXMB链接到Common FIFO后，用户不应再向这个TXMB写入数据。

同时，链接到FIFO的TXMB不应成为TX队列中的一部分。

CFDCFCC.CFTML[1:0]寄存器用于配制TXMB<> Common FIFO TX-mode链接：

00 –TXMB0

01 – TXMB1

10 – TXMB2

11 – TXMB3

24.9.3.3 FIFO 深度配制

RX FIFO 和 Common FIFO 深度都可以单独配置，通过 CFDRFCCa.RFDC[2:0] 和 CFDCFCC.CFDC[2:0]寄存器设置。

000 – 0 message

001 – 4 messages

010 – 8 messages

011 – 16 messages

100 – 32 messages

101 – 48 messages

110 – RFU

111 – RFU

注意RXMB和FIFO的RAM容量分配上限是16个64字节payload消息（1KB），配置超过16个消息的深度则不能支持64字节payload，软件应注意避免配置深度大于RAM上限。

24.9.3.4 FIFO payload size configuration

FIFO中保存的data payload size可以通过CFDRFCCa.RFPLS和CFDCFCC.CFPLS寄存器配置，可选的payload size有：8/12/16/20/24/32/48/64字节。

RXMB和FIFO能够支持的最大数据长度是16个64字节payload消息，软件不能配置大于这个上限。

24.9.3.5 FIFO 中断配置

RXFIFO和Common FIFO的中断策略可以通过CFDRFCCa.RFIM和CFDCFCC.CFIM（中断模式寄存器）分别配置。RFIM和CFIM都有两种模式可选：

- RFIM/CFIM=0

RX FIFO mode：FIFO指针等于CFDRFCCa.RFIGCV /CFDCFCC.CFIGCV时产生中断

TX FIFO mode：Common FIFO成功发送完最后一个消息后产生中断

- RFIM/CFIM=1

RX FIFO mode：FIFO每次保存接收消息后产生中断

TX FIFO mode：Common FIFO每次成功发送完一个消息后产生中断

当中断模式配置为0，RX FIFO的中断产生基于CFDRFCCa.RFIGCV，Common FIFO如果配置为RX模式则中断产生基于CFDCFCC.CFIGCV，可选的配置包括：

3’b000: Interrupt generated when FIFO is 1/8th Full

3’b001: Interrupt generated when FIFO is 1/4th Full

3’b010: Interrupt generated when FIFO is 3/8th Full

3’b011: Interrupt generated when FIFO is 1/2 Full

3’b100: Interrupt generated when FIFO is 5/8th Full

3’b101: Interrupt generated when FIFO is 3/4th Full

3’b110: Interrupt generated when FIFO is 7/8th Full

3’b111: Interrupt generated when FIFO is Full

需要注意的是，根据FIFO深度配置不同（FDC[2:0]寄存器），CFDRFCCa.RFIGCV/CFDCFCC.CFIGCV寄存器的配置有一定限制，如下表：

RFDC[2:0] (CFDC[2:0])	RFIGCV[2:0] (CFIGCV[2:0])							
	111	110	101	100	011	010	001	000
000	don’t care (FIFO can not be enabled)							
001	allowed	not allowed	allowed	not allowed	allowed	not allowed	allowed	not allowed
010	allowed							
011	allowed							
100	allowed							
101	allowed							

表 25-10FIFO 中断配置与 FIFO 深度

24.9.3.6 FIFO 控制

通过设置CFDRFCCa.RFIE来使能RXFIFO接收中断使能，通过设置CFDCFCC.CFRXIE or CFDCFCC.CFTXIE寄存器来使能CommonFIFO接收中断使能和发送中断使能。

所有配置完成后，通过置位CFDRFCCa.RFE andCFDCFCC.CFE寄存器来使能RXFIFO和CommonFIFO，此后FIFO可用于数据收发。

25.10 中断和 DMA

25.10.1 中断

RS-CANFD-Lite模块的中断分为两组：全局中断和通道中断。

- 全局中断

- 消息成功接收到2个RXFIFO中
- 消息成功接收到RXMB中
- 全局错误

- 通道中断

- 通道发送中断：发送成功，发送放弃，队列发送完成，Common FIFO发送完成
- Common FIFO接收成功
- 通道错误

全局中断结构框图：

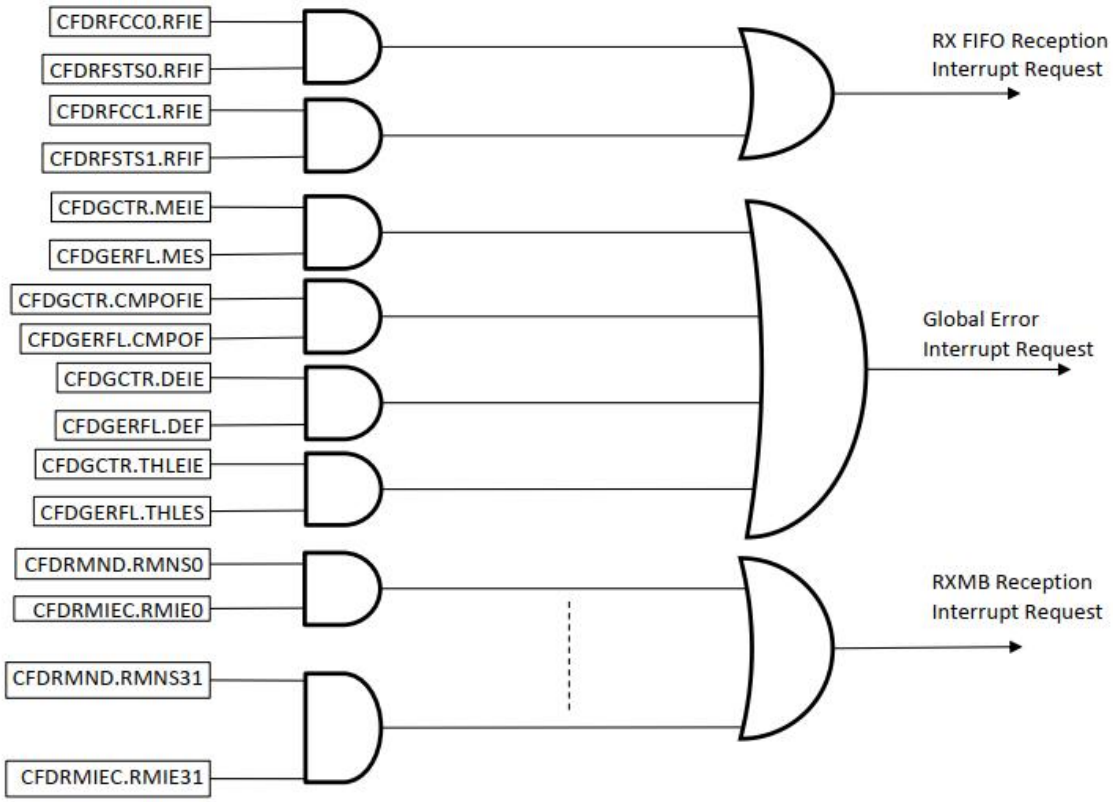


图 25-30 global interrupts block diagram

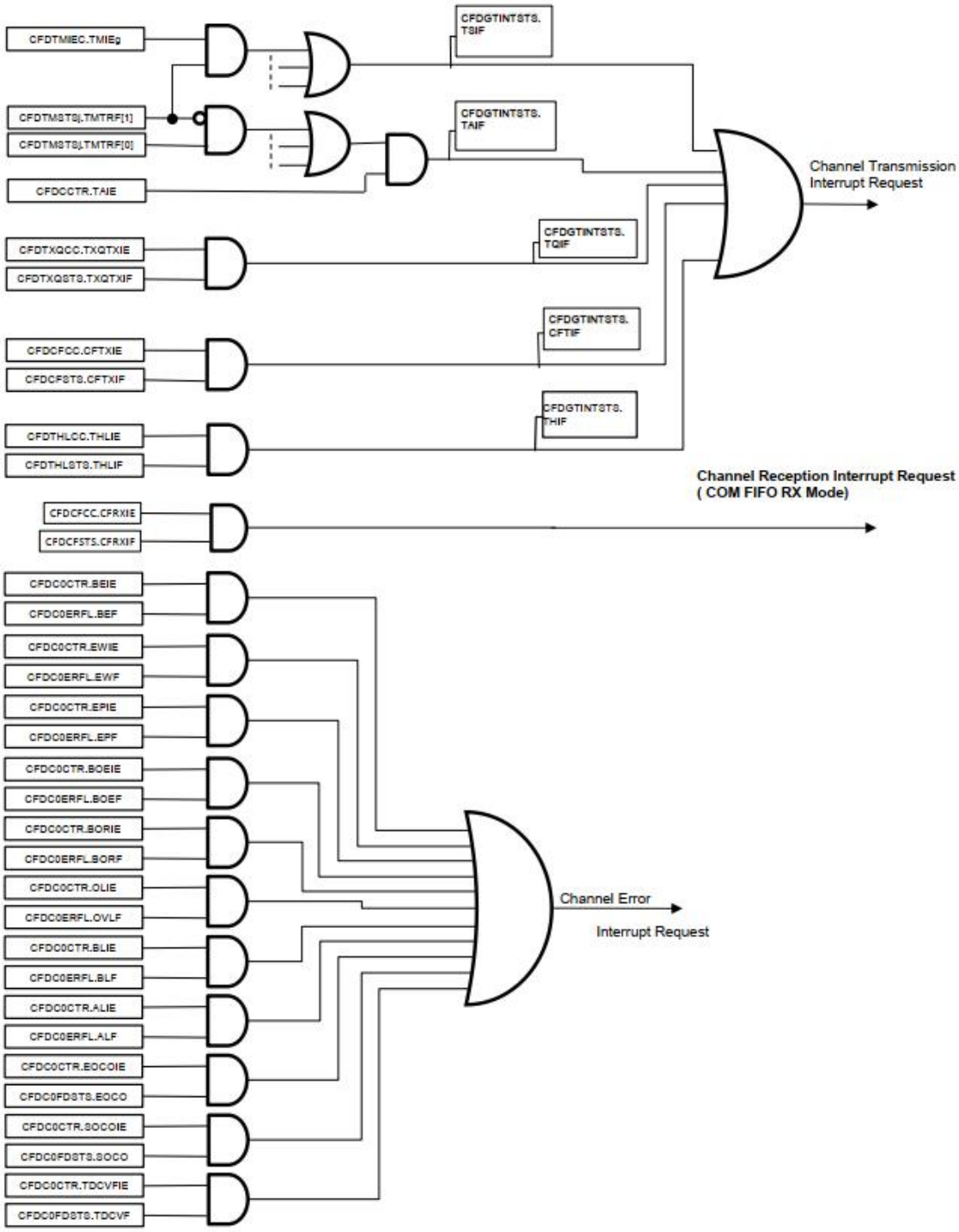


图 25-31 channel interrupts block diagram

25.10.2 DMA（不使用此功能，因为没有 DMA 通道分配）

25.11 接收和发送

25.11.1 消息接收

根据AFL entry配置，收到的并且通过消息滤波后的消息将被存入RXMB或者RXFIFO或者CommonFIFO中。

接收消息存入RXMB

当接收到的消息成功存入某个RXMB，此RXMB的newdata标志将自动置位。如果RXMB中旧的数据还未被读取，新的数据将会覆盖旧的数据。

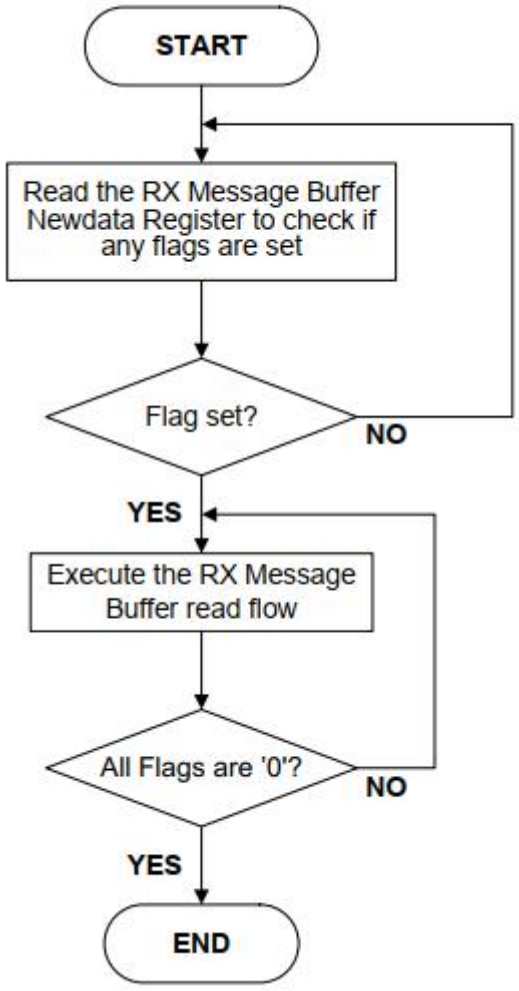


图 25-32RXMB 消息访问流程（轮询）

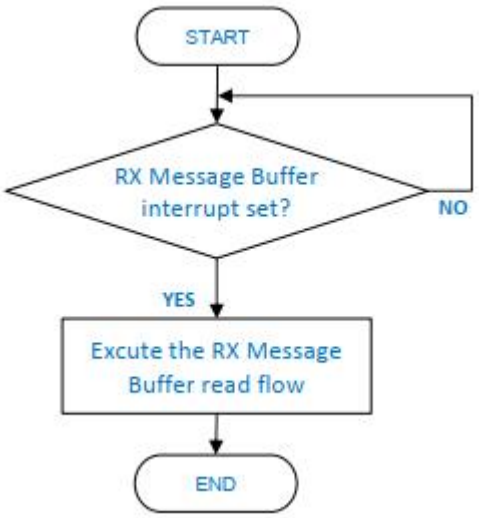


图 25-33RXMB 消息访问流程（中断）

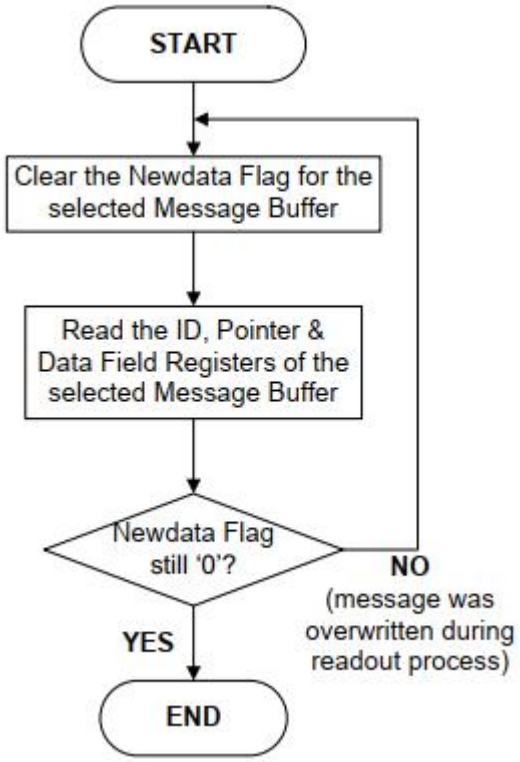


图 25-34RXMB 读取流程

接收消息存入 FIFO

当接收消息成功存入RX FIFO或者Common FIFO，FIFO对应的消息计数器自动递增，软件可以通过FIFO访问寄存器读取FIFO中的消息。

如果FIFO消息计数器等于FIFO深度，则FIFO满标志置位。如果软件对FIFO指针控制寄存器写FF，则FIFO消息计数器自动减1。用户应该在完整读取一个消息后，再对FIFO指针控制寄存器写FF。当FIFO中所有消息都被读取后，FIFO空标志置位。

在FIFO满的情况下收到新的消息，则FIFO message lost标志置位，新收到的消息将被丢弃（不会覆盖FIFO中的旧数据），为了避免数据丢失，应用可以合理配置FIFO告警深度，以提前预警FIFO将满。

RX FIFO和Common FIFO通过CFDRFCCa.RFE 和CFDCFCC.CFE寄存器来使能或禁止，当FIFO被禁止后，读写指针清零，FIFO内所有数据丢失，新收到的数据不会存入FIFO。

如果接收FIFO连接到DMA通道，则软件不应再通过FIFO访问寄存器来访问FIFO，也不可以对指针控制寄存器写FF，否则将导致无法预测的结果。

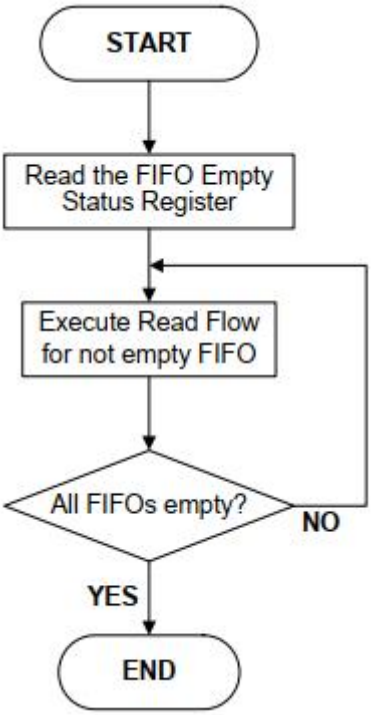


图 25-35FIFO 消息读取流程（轮询）

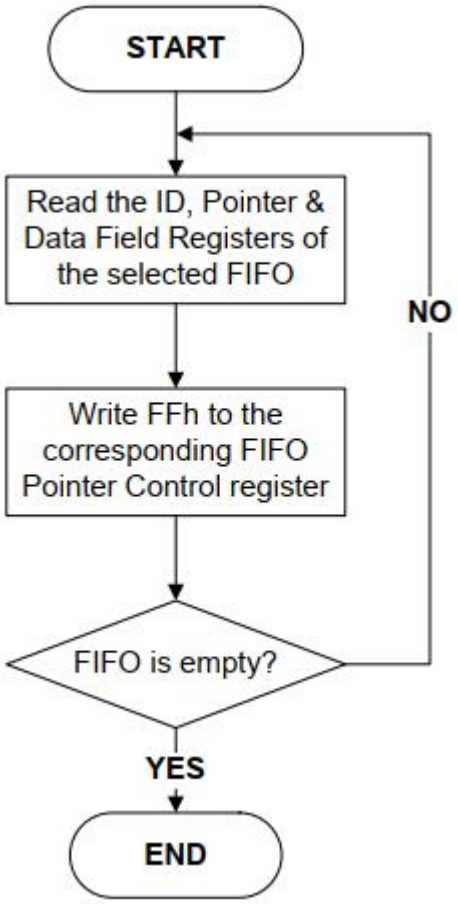


图 25- 36 接收 FIFO 读取流程（轮询）

25.11.2 时间戳 (Timestamp)

时间戳定时器是一个自由计数定时器，用于记录消息的接收时间和发送时间。根据CFDGFDCFG.TSCCFG[1:0]寄存器配置，timestamp可以在SoF采样点或者RES bit采样点记录。对于接收消息，时间戳与消息ID、数据一起存入RXMB或FIFO；对于发送消息，时间戳将被存入TX history list。

时间戳定时器的工作时钟可以选择APBCLK或CAN bit timing clock；使用bit timing clock时，要注意在CAN节点进入channel halt mode或channel reset mode时，定时器会被停止；使用APBCLK时定时器的工作不受通道模式影响。

CFDGCFG.TSP寄存器用于对定时器计数时钟预分频，CFDGCTR.TSRST寄存器用于复位时间戳定时器。

25.11.3 消息发送

RS-CANFD-Lite支持多种发送配置：

- Normal transmission

- FIFO transmission
- TX Queue transmission

最多4个TX buffer可以被组合成发送队列，发送队列使用TX buffer0 作为访问窗口。

Common FIFO配置为发送模式时，需要link到某个发送buffer作为访问窗口，Common FIFO不能link到已经配置为发送队列的TX buffer上。

发送配置示意图：

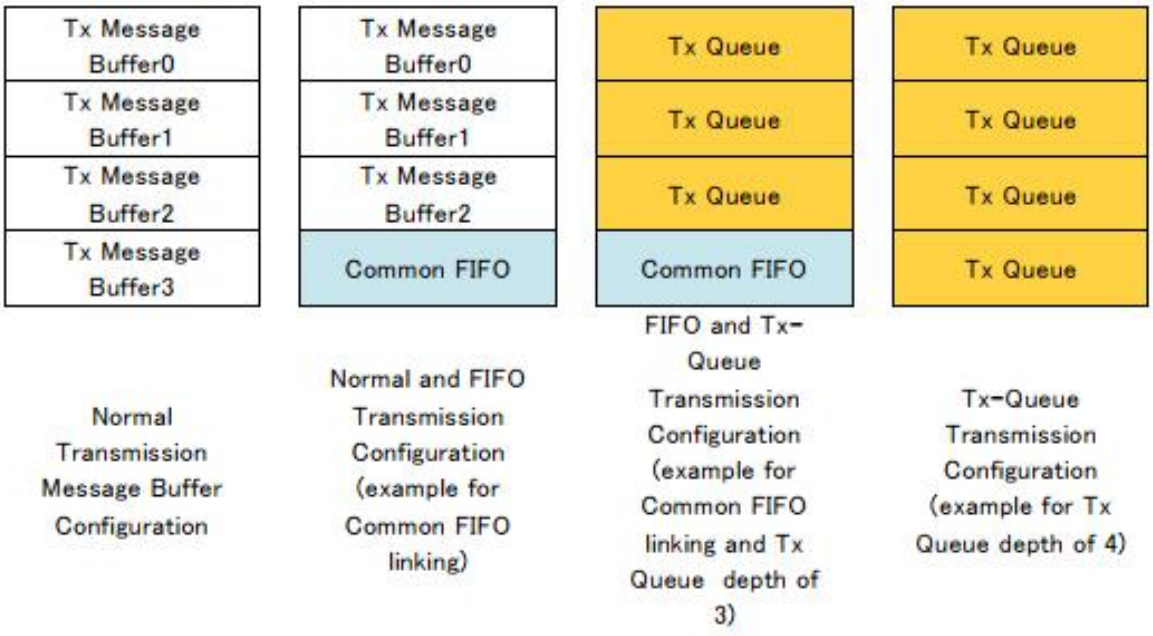


图 25-37 发送 buffer 配置

25.11.4 发送优先级

当多个buffer需要发送数据时，系统引入以下2种优先级策略：

- CAN ID优先级
- 消息buffer编号优先级

buffer编号优先级策略规定较小编号的buffer具备较高优先级，被link到特定buffer上的Common FIFO也遵循这一规则，此时FIFO的优先级由link的buffer编号决定。

注意，buffer编号优先级策略不能在发送队列使能的情况下使用。

ID优先级规则符合CAN总线仲裁规则，所有待发送消息，包括TXMB、Common FIFO和TX Queue中的消息，都会参与到ID优先级仲裁。

如果两个或多个消息的ID相同，则占据最小buffer编号的消息优先发送。

注意：发送队列中的所有buffer都会参与到发送仲裁中。

25.11.5 Normal transmission

每个发送buffer支持2种发送模式：

- 常规发送

  对于常规发送，如果buffer在仲裁中失败，或者发送中遇到错误，buffer会继续尝试发送消息，除非transmission abort request被置位

- one-shot发送

  对于one-shot发送，如果buffer在仲裁中失败，或者发送中遇到错误，buffer不会继续尝试发送消息

TXMB发送请求流程：

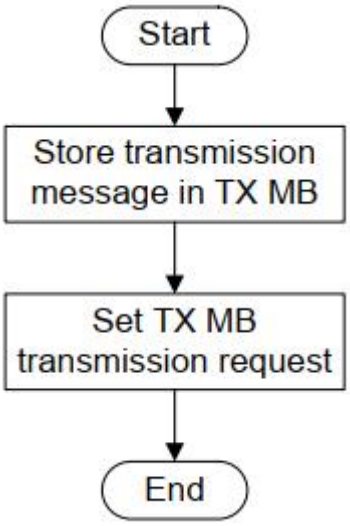


图 25-38 发送 buffer 请求

TXMB控制寄存器配置表：

Transmission Request CFDTMCi.TMTR	Transmission Abortion Request CFDTMCi.TMTAR	One Shot Enable CFDTMCi.TMOM	Communication activity
0	0	0	Message Buffer disabled
0	0	1	Message Buffer disabled
1	0	0	configured as a transmission Message Buffer for a data frame or a remote frame
1	0	1	configured as a one shot transmission Message Buffer for a data frame or a remote frame
1	1	0	Transmission abortion requested
1	1	1	One shot transmission abortion requested

表 25-11TX message buffer 发送模式配置

下图显示了2个TXMB成功发送的时序示意图：

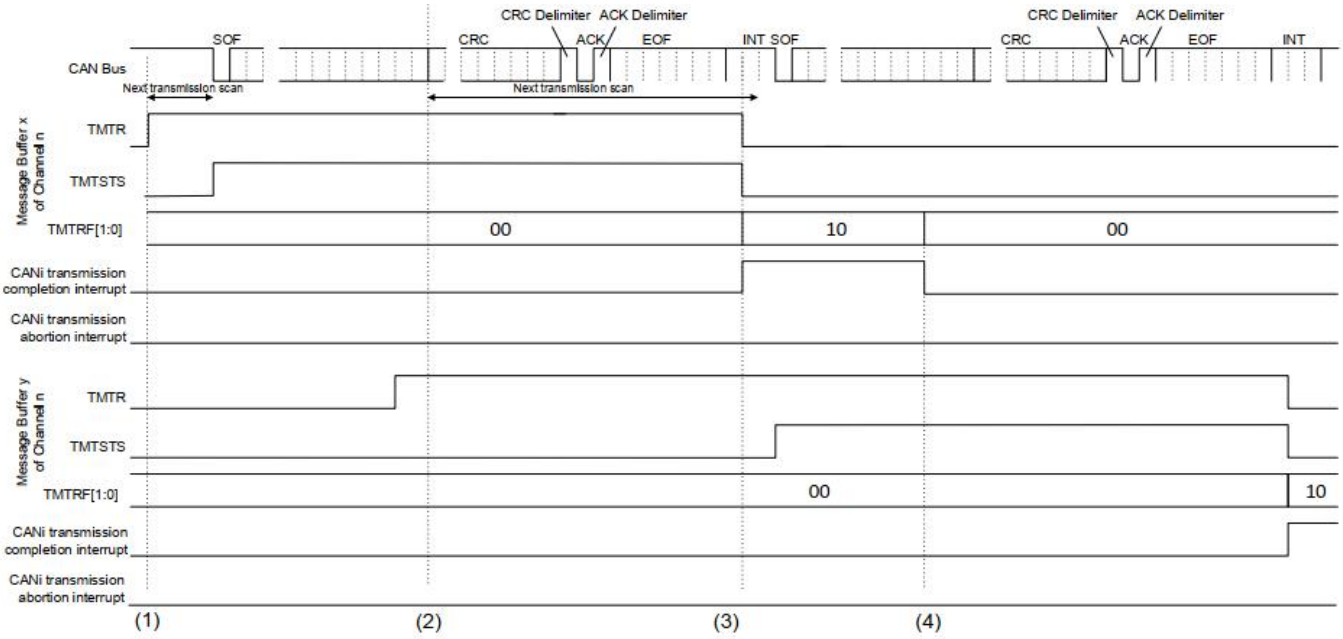


图 25-39 成功发送的请求和标志时序

（1） 在bus idle状态下置位发送buffer控制寄存器中的CFDTMCi.TMTR bit，message buffer扫描过程开始，决定最高优先级的发送buffer；当最高优先级的buffer被选中后，此buffer的状态寄存器中的CFDTMSTSj.TMTSTS bit被置位，表示此buffer正在发送，CAN通道开始发送

（2） 发送第一个CRC bit时，发送扫描逻辑开始判决下一个发送对象

（3） 当前消息被成功发送后，buffer状态寄存器中的CFDTMSTSj.TMTRF[1:0]被设置为10，

CFDTMSTSj.TMTSTS和CFDTMCi.TMTR被清零

（4） 在发送下一个消息之前，清零CFDTMSTSj.TMTRF，将下一个消息写入发送buffer，然后再置位CFDTMCi.TMTR请求发送；CFDTMSTSj.TMTRF被清零前，软件无法置位CFDTMCi.TMTR

下图显示了发送撤销的时序示意图：

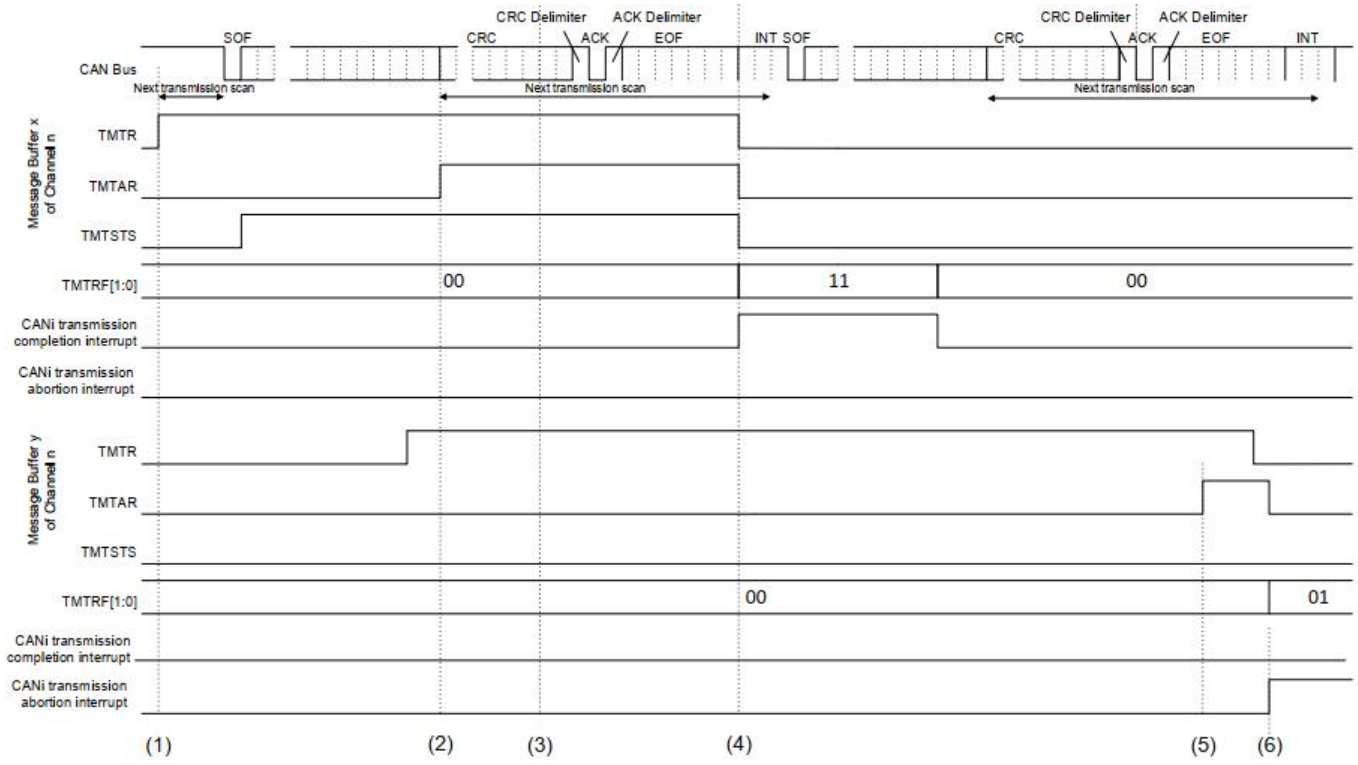


图 25-40 发送撤销的请求和标志时序

（1） 在bus idle状态下置位发送buffer控制寄存器中的CFDTMCi.TMTR bit，message buffer扫描过程开始，决定最高优先级的发送buffer；当最高优先级的buffer被选中后，此buffer的状态寄存器中的CFDTMSTSj.TMTSTS bit被置位，表示此buffer正在发送，CAN通道开始发送

（2） 如果撤销请求CFDTMCi.TMTAR置位时，当前buffer已经被选中发送，或者已经处于发送状态，则消息发送不会被撤销

（3） 在第一个CRC bit处，状态机开始下一轮发送扫描

（4） 如果消息发送成功，CFDTMSTSj.TMTRF标志被设置为11，CFDTMSTSj.TMTSTS和CFDTMCi.TMTR被清零

（5） 其他CAN节点正在发送，当前buffer没有启动发送，此时置位CFDTMCi.TMTAR撤销发送

（6） 经过内部处理延迟后，发送被取消，CFDTMSTSj.TMTRF标志被设置为01，CFDTMCi.TMTR和CFDTMCi.TMTAR被自动清零

25.11.6 FIFO transmission

使用common FIFO做消息发送前，必须通过CFDCFCC.CFTML寄存器将其链接到某个特定的发送buffer上，而被链接到common FIFO的TX buffer此后不应再被软件配置。

用户通过访问FIFO access寄存器将需要发送的消息写入common FIFO，然后通过对FIFO指针控制寄存器写FF，可以将FIFO的消息计数值加1，如果消息个数等于FIFO深度，则FIFO满标志置位。

FIFO中最早写入的消息会参与到当前的发送扫描流程中。当一个消息从common FIFO中发送出去后，FIFO消息计数值自动减1，当FIFO中所有消息都被成功发送后，FIFO空标志自动置位。

FIFO发送请求流程：

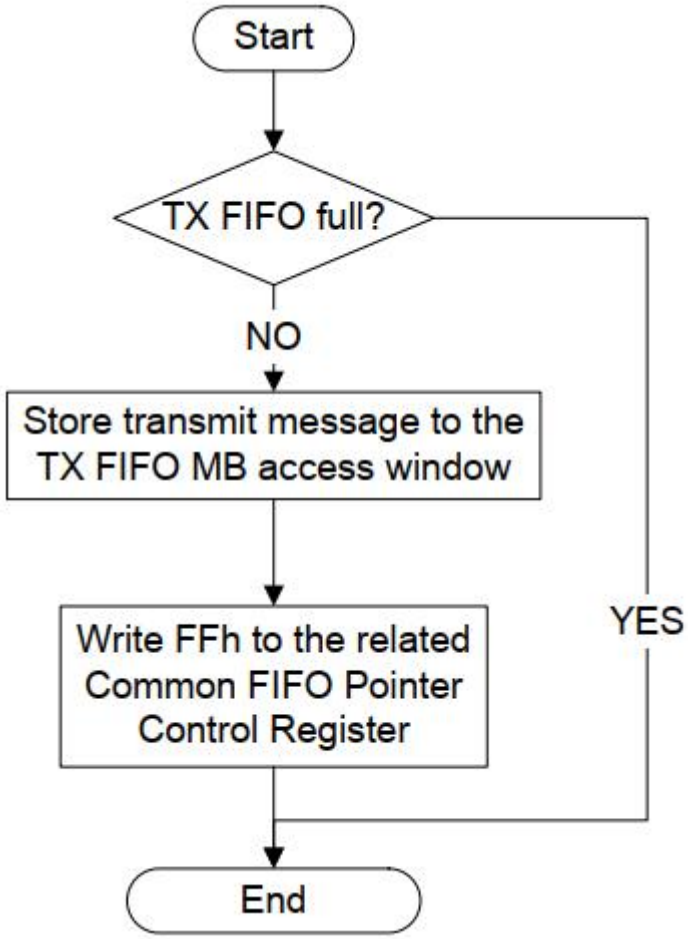


图 25-41 FIFO 发送请求流程

Common FIFO带有一个间隔定时器，这个定时器用于在连续消息发送之间插入特定延迟，定时器可以使用bit timing clock或者APBCLK工作，定时长度0~255个时钟周期。

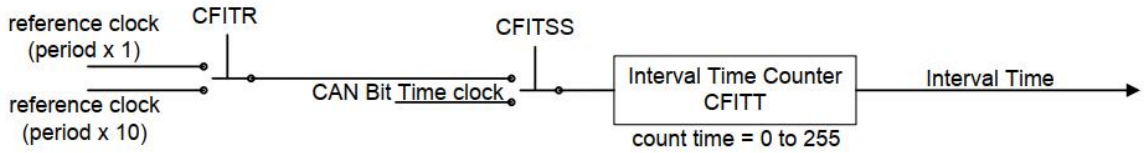


图 25-42 FIFO 间隔定时器结构框图

如果选择bit timing clock，则通道处于RESET、SLEEP、HALT模式下时，定时器停止工作。

如果选择APBCLK，则通道处于RESET或SLEEP模式下时，定时器停止工作。

Reference clock Peripheral clock	1μs	100μs	500μs
16MHz / 62.5 ns	16	1600	8000
20MHz / 50 ns	20	2000	10000
32MHz / 31.25 ns	32	3200	16000
50MHz / 20 ns	50	5000	25000

表 25-12FIFO 间隔定时例子

间隔定时器可以满足ISO15765-2定义的seperation time ，能够其要求的最大范围100us~127ms

当前消息发送完成后（EOF7）间隔定时器开始计数，计数完成后FIFO的发送请求置位，这样可以保证FIFO发送的连续2个消息之间的最小间隔。

下图是一个间隔发送的例子和相应的参数配置：

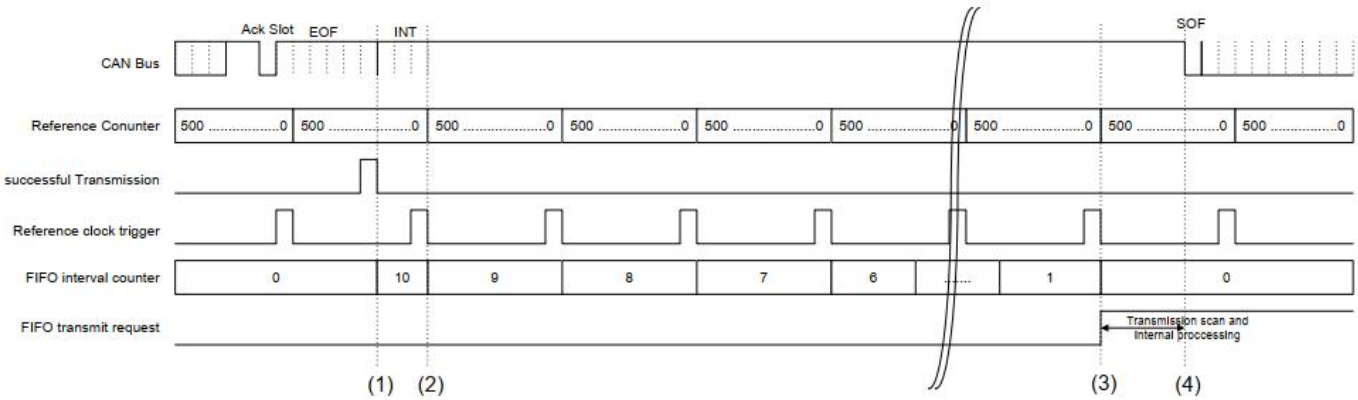


图 25-43 FIFO 间隔定时发送时序

Peripheral clock frequency = 50MHz

Interval Timer Reference clock (CFDGCFG.ITRCP)= 500 times

Reference clock due to the settings above = 10μs

Common FIFO interval Timer Source Selection (CFDCFCC.CFITSS)= 0

Common FIFO Interval Timer Resolution (CFDCFCC.CFITR)= 0

Common FIFO Interval Transmission time (CFDCFCC.CFITT)= 10 times

Theoretical Message separation interval = 100 μs

（1） 间隔定时器在消息发送完成后启动，注意启动过程不会同步于reference clock trigger，因此第一个间隔可能少于或等于一个reference clock interval

（2） 下一个reference clock trigger到来使得定时器递减

（3） 当定时器减到0，FIFO发送请求自动置位

（4） 由于内部处理延迟，3个CAN bit时钟后，开始发送

25.11.7 发送队列（TX Queue）

发送队列（TXQ）由3个或者4个发送buffer组成，其中TX buffer0作为访问窗口，用户应该只通过TXMB0来访问队列，而不能直接访问组成队列的TXMB。队列中所有的消息参与发送优先级比较，队列内部仅以ID优先级排序。

通过访问窗口写入的消息，会自动存入队列中空闲的buffer。如果队列已满，继续写入数据将会覆盖已有数据。

发送队列通过置位TXQE寄存器使能，当一个消息被存入TXQ，软件应向TXQ指针控制寄存器写FF，此时发送请求将被置位，TXQ内部指针将指向下一个空闲的消息buffer，以便后续消息可以写入空闲buffer。

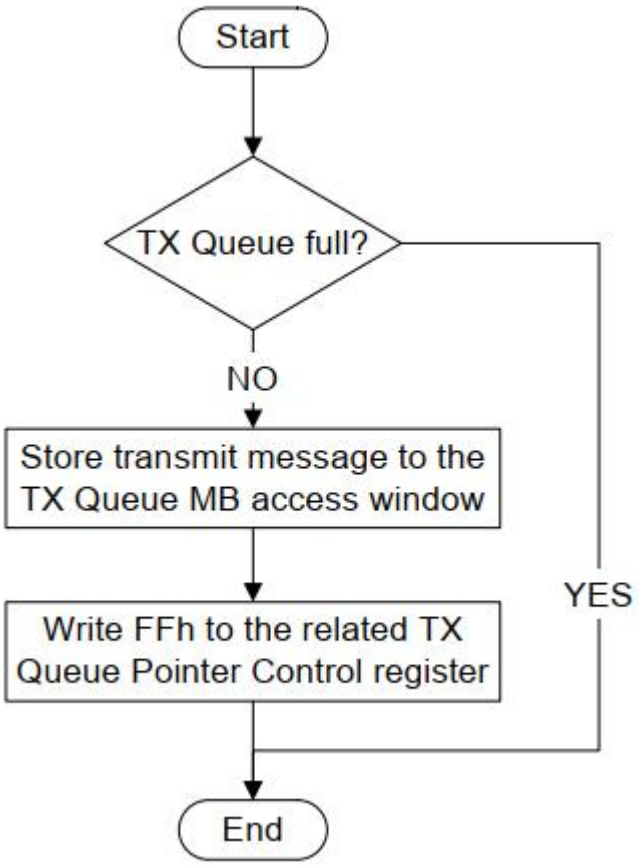


图 25-44 TX Queue 发送请求

25.11.8 发送历史列表（TX History List）

TX History List (THL) 功能用来记录成功发送的消息历史，模块提供2个THL buffer，每个buffer可以记录8个THL entry。

通过配置CFDTHLCC.THLDTE寄存器，软件可以选择记录TX FIFO/TXQ的发送历史，或者TX FIFO/TXQ/TXMB的发送历史。

每个消息也都可以通过CFDCFID.THLEN配置是否要被记录到THL中。THL成功记录一条发送历史后，THLIF标志置位，并且当THLIE=1时可以产生中断。

THL中会记录关于发送消息的以下信息：

- buffer type
  - 001: TXMB
  - 010: TX FIFO
  - 100: TX Queue
- buffer number，参见下表

BufferType / BufferNumber	001	010	100
00	Message Buffer 0	Number shown corresponds to the common FIFO TX Message Buffer Link CFTML of the related Common FIFO Configuration / Control Register	Number shown corresponds to the Message Buffer belonging to the TX Queue from which the frame was transmitted
01	Message Buffer 1		
10	Message Buffer 2		
11	Message Buffer 3		

表 25-13buffer number 和 buffer type

- transmission ID：保存在发送消息中的发送指针
- transmit timestamp：发送时捕获的消息时间戳
- transmission info label：发送消息里保存的信息标签

THL中保存的entry可以通过访问TX history list access register来读取，每读取一条entry后，需要向TX history list pointer register写入FF，然后可以读取下一条entry。

THL准备流程：

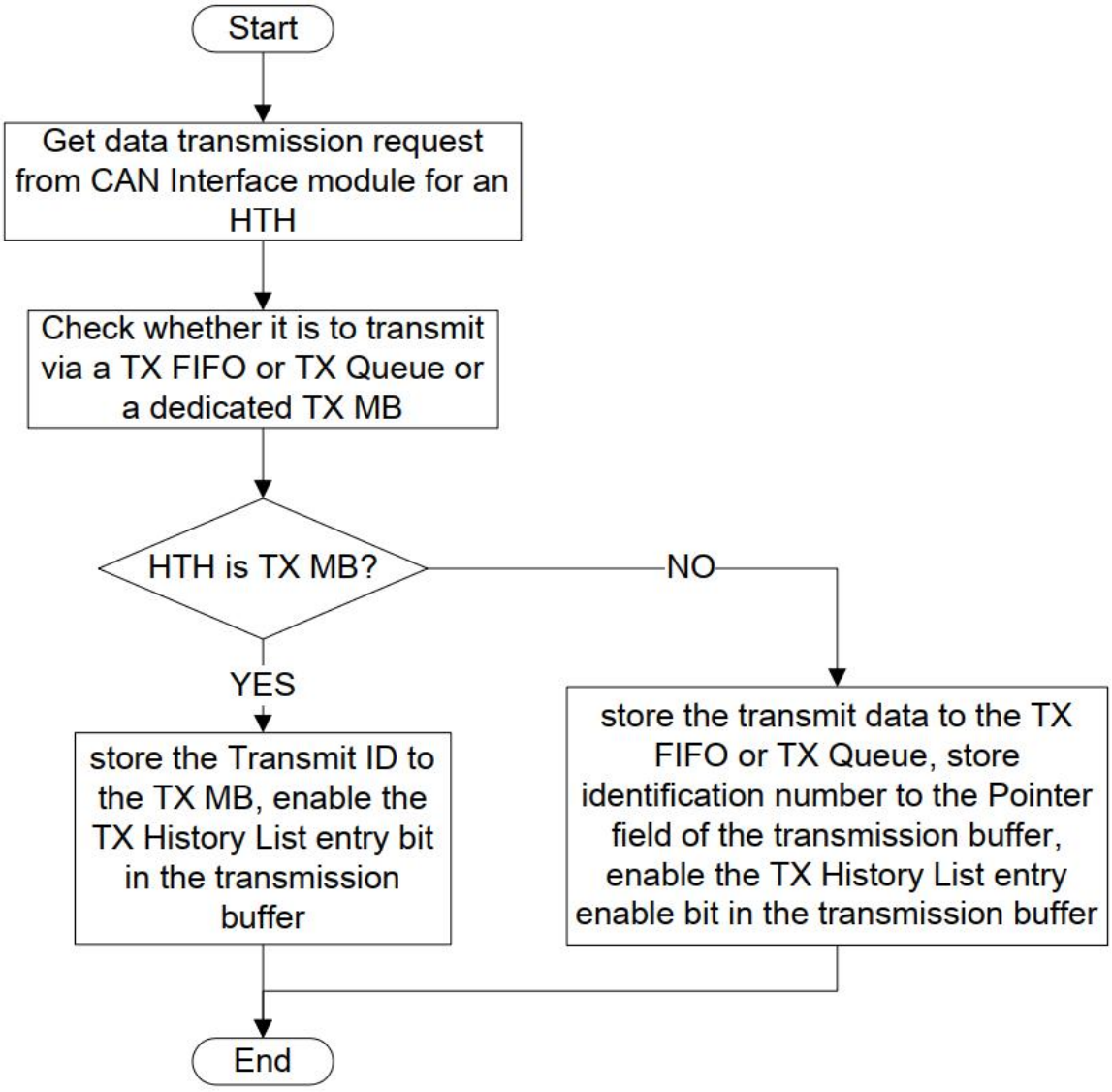


图 25-45 THL preparation flow

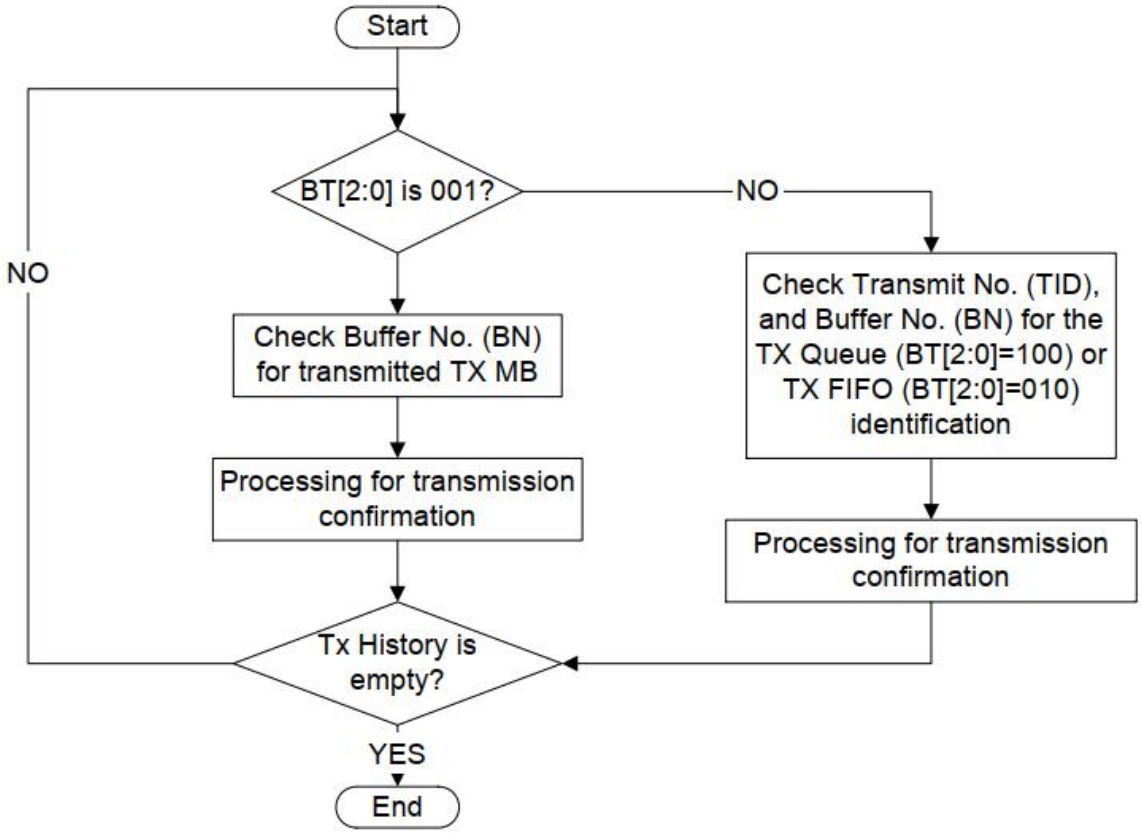


图 25-46 THL processing flow

25.11.9 发送数据填充（TX data padding）

如果消息中的DLC值大于发送buffer的data size，则发送的数据将会被填充CC。

25.12 测试模式

RS-CANFD-Lite包含多种测试模式，这些测试模式互相之间都是互斥的，即不可以同时使能多种测试模式。

25.12.1 通道测试模式

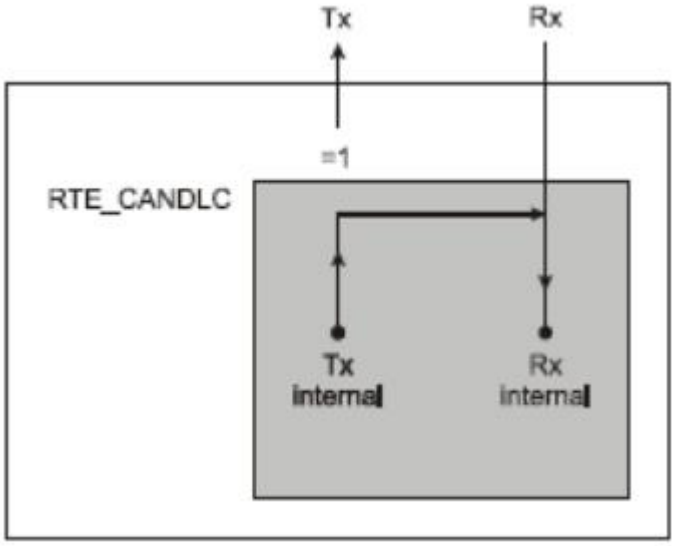
CAN通道包含以下测试模式：

- 基本测试
- Listen-only
- self-test mode 0 (external loopback)
- self-test mode 1 (internal loopback)
- restricted operation mode

25.12.1.1 Listen-only

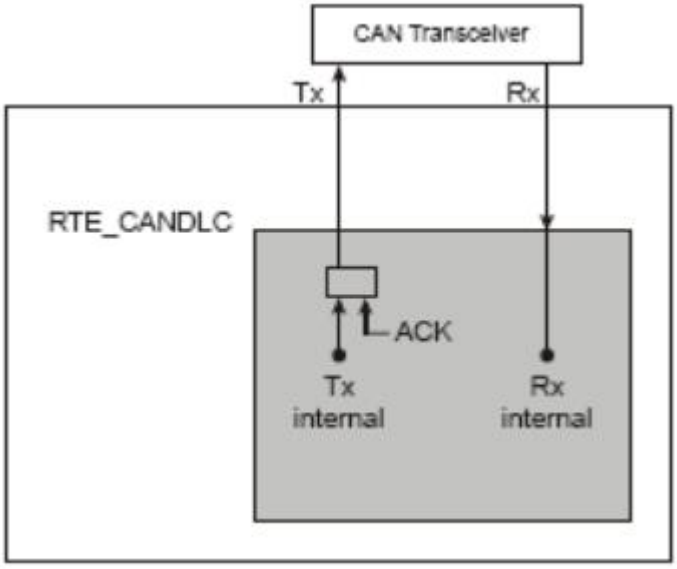
此模式下CAN节点仅监听总线信号，不会向总线发送任何数据。如果CAN节点需要发送显性位，这个电平会在内部连接，以使CAN节点能够接收到自己发送的显性电平。

Listen-only模式下，禁止任何发送请求。



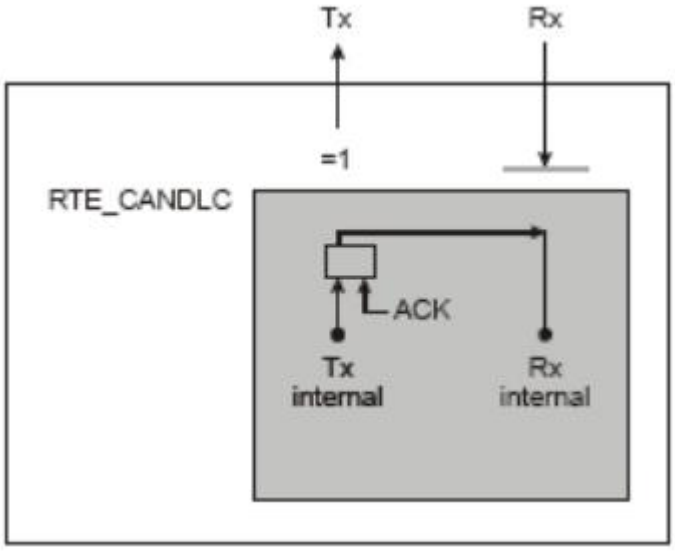
25.12.1.2 self test mode 0 (external loopback mode)

此模式下，CAN模块会接收自己发送的消息，并将收到的消息存入RXMB，此模式用于外部CAN transceiver测试，RX/TX引脚应连接到外部transceiver。



25.12.1.3 self test mode 1 (internal loopback mode)

此模式下，CAN模块会接收自己发送的消息，并将收到的消息存入RXMB，收发的消息不会经过外部transceiver，RX/TX引脚不需要连接到片外transceiver。



25.12.1.4 restricted operation mode

此模式下CAN节点可以正常接收数据和远程帧，并产生ack bit，active error和过载帧不会被发送，并且REC和TEC计数值冻结。

这个模式是由ISO11898-1定义的。

25.12.2 全局模式

RS-CANFD-Lite支持以下全局测试模式：

- RAM测试
- Bit flip 测试

全局测试模式由专用的key保护，需要unlock key才能使能：

Test Mode	Unlock key 1	Unlock key 2
RAM Test Mode	7575h	8A8Ah

解锁操作必须连续进行两次key写入，而不能中间插入对其他SFR的写操作，否则解锁序列必须重新开始。

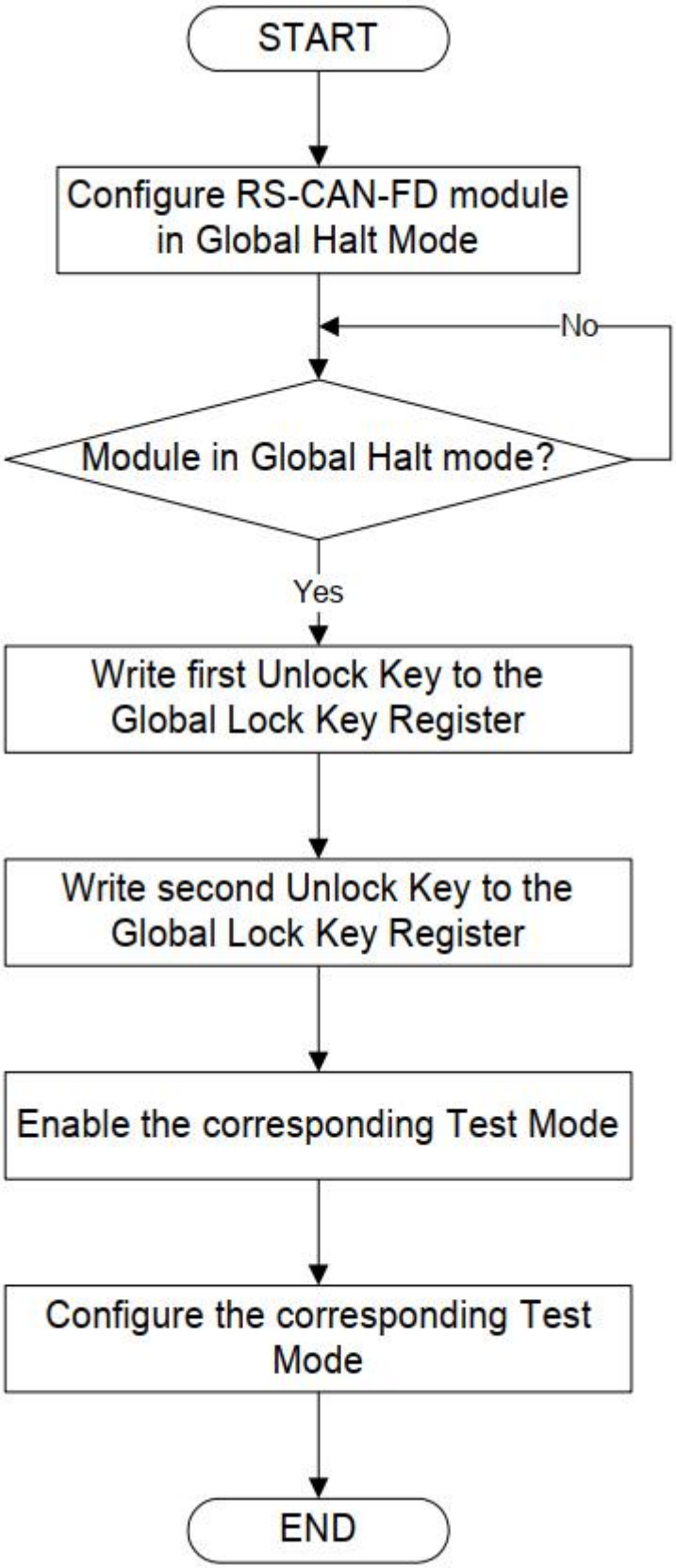
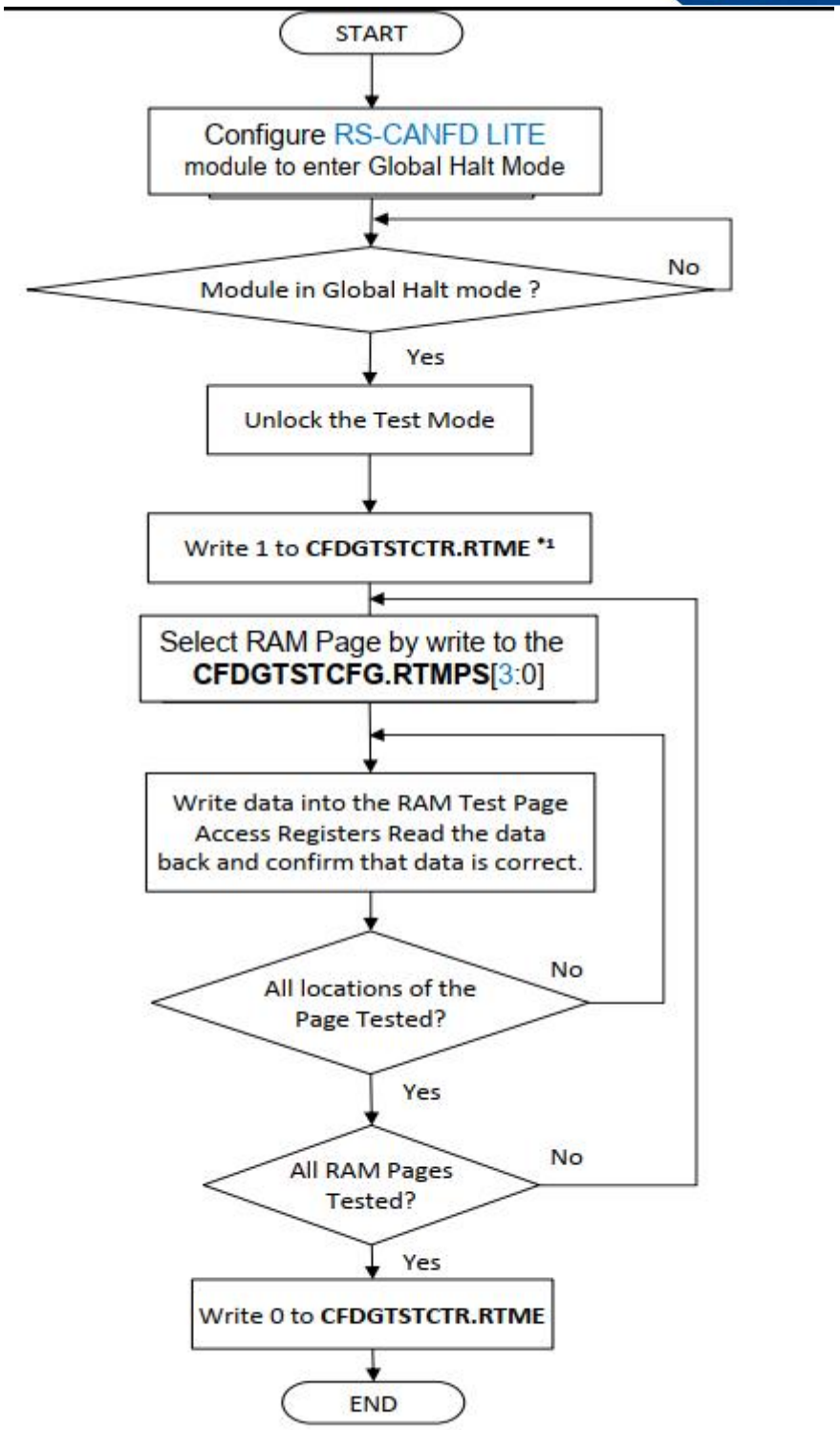


图 25-47 软件写保护解锁流程

25.12.2.1 RAM 测试

RAM测试模式使能后，整块RAM可以被软件访问。注意RAM空间以256字节为page组织，page选择通过CFDGTSTCFG.RTMPS[3:0]寄存器实现，然后通过访问ram test page access寄存器实现对ram当前page的读写。

在CANFD模块不用于通信的情况下，可以通过RAM测试功能，将message buffer RAM用作系统存储器使用。



*1 Change into the following status before changing to RAMTEST.
Cancel of a request of transmission.
Disable of all the FIFO and TXQ.
Clear of the receiving flag of a receiving buffer.

图 25-48 RAM 测试流程

25.12.2.2 Bit flip 测试

Bit flip测试能够翻转消息ID的第一个bit，对于发送节点，将触发bit错误或者仲裁丢失，对于接受节点，将触发CRC错误或者stuff错误。

25.13 RAM 配置

RS-CANFD-Lite模块例化的RAM用于保存以下数据：

- Rule table
- RXMB + FIFO buffer
- TXMB
- THL
- PFL Rule table

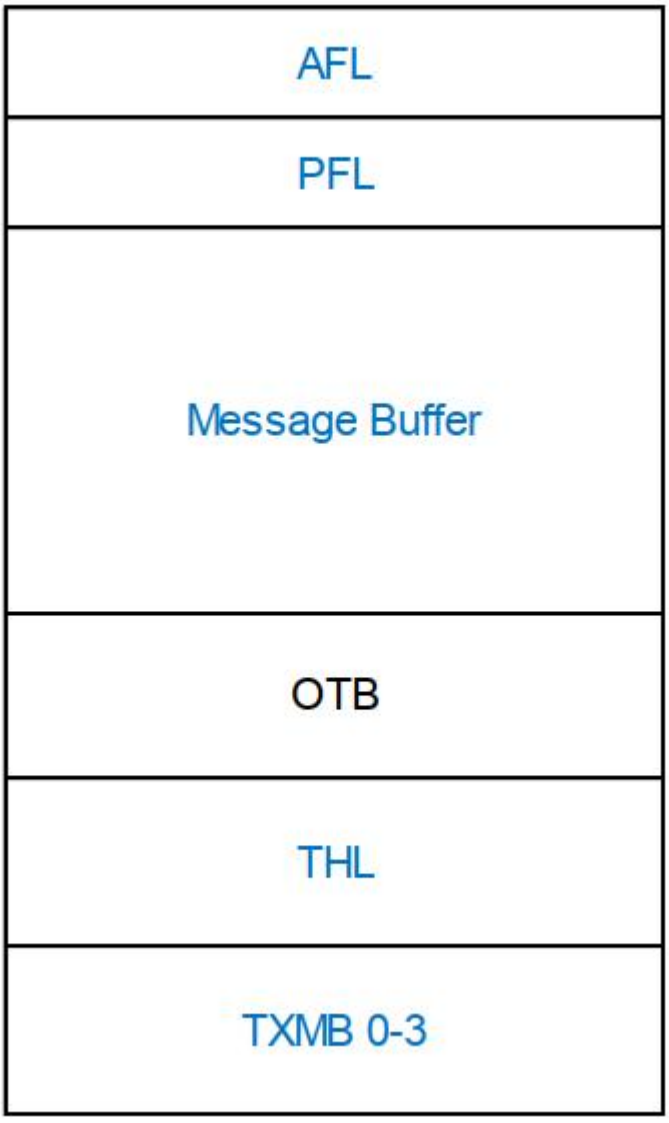


图 25-49 RAM 分组

MRAM_cfg = RXMB_MRAM_cfg + RXFIFO_MRAM_cfg + CFIFO_MRAM_cfg +TXMB_MRAM_cfg +THL_MRAM_cfg + OTB_MRAM_cfg +AFL_MRAM_cfg + PFL_MRAM_cfg

RXMB_MRAM_cfg = (12 Bytes + CFDRMND.RMPLS) * CFDRMNB.NRXMB

RXFIFO_MRAM_cfg = SUM((12 Bytes + CFDRFCCa.RFPLS) * CFDRFCCa.RFDC)

CFIFO_MRAM_cfg = (12 Bytes + CFDCFCC.CFPLS) * CFDCFCC.CFDC

TXMB_MRAM_cfg = 304 Bytes

THL_MRAM_cfg = 64 Bytes

OTB_MRAM_cfg = 160 Bytes

PFL_MRAM_cfg = 72 Bytes

AFL_MRAM_cfg = 512 Bytes

RAM Name	RAM Property	RAM Area Calculation Method	RAM Values
AFL	Avg. rule entries		32
	No. of Bytes in a rule entry	Fixed	16
	Number of Bytes in AFL area	Avg. rule entries * No. of Bytes in a rule entry	512
PFL	Avg. rule entries		2
	No. of Bytes in a rule entry	Fixed	36
	Number of Bytes in PFL area	Avg. rule entries * No. of Bytes in a rule entry	72
TX MB	No. of TX MBs	Fixed	4
	No. of Bytes needed for each TX MB	Fixed	76
	Number of Bytes in TX MB area	No. of TX MBs * No. of Bytes needed for each TX MB	304
THL	No. of entries in 1 THL buffer	Fixed	8
	No. of Bytes needed for each THL entry	Fixed	8
	Number of Bytes in THL area	No. of entries in 1 THL buffer * No. of Bytes needed for each THL entry	64
OTB	Avg. number of buffers		2
	No. of Bytes for OTB entry	Fixed	80
	Number of Bytes in OTB area	Avg. number of buffers * No. of Bytes for OTB entry	160
Message Buffer	No. of RX MBs	Fixed	32
	No. of RX FIFOs	Fixed	2
	No. of Common FIFO	Fixed	1
	Avg. number of messages for RXMB and FIFO buffers		16
	No. of Bytes for each stored message	Fixed	-
	Average size of a Message Buffer in Bytes		76
	Number of Bytes in Message Pool area	Avg. number of messages for RXMB and FIFO buffers * Average size of a Message Buffer in Bytes	1216
	Number of Bytes Message RAM	Number of Bytes in Message Pool area + Number of Bytes in OTB area + Number of Bytes in THL area + Number of Bytes in TXMB area + Number of Bytes in PFL area + Number of Bytes in AFL area	2328

表 25-14RAM 容量计算

上表共需要2328字节

配置举例：

		818h
	AFL area	
		718h
	PFL area	
		6D0h
	Unused area	
		640h
CFDCFCC.CFDC=1d (4 Message) **CFDCFCC.CFPLS**=0d (8byte) →20byte per Message	COM FIFO 0	
		5F0h
CFDRFCC1.RFDC=2d (8 Message) **CFDRFCC1.RFPLS**=0d (8byte) →20byte per Message	RX FIFO 1	
		550h
CFDRFCC0.RFDC=3d (16 Message) **CFDRFCC0.RFPLS**=5d (32byte) →44byte per Message	RX FIFO 0	
		290h
RXMB: **CFDRMNB.NRXMB**=4d (4 Message) **CFDRMNB.RMPLS**=3d(20byte) →32byte per RXMB	RX MB	
		210h
	OTB	
		170h
	THL	
		130h
	TXMB[3]	
	:	
	TXMB[0]	
		000h
		(unit : Byte)

表 25-15RAM 配置举例

25.14 PNF (Pretended Network Filter)

虚拟网络（pretended network, PN）功能用于降低系统功耗。PN模式下，RS-CANFD-Lite模块根据payload filter的结果产生CPU唤醒事件，因此可以减轻CPU检查收到的消息数据的工作负荷。

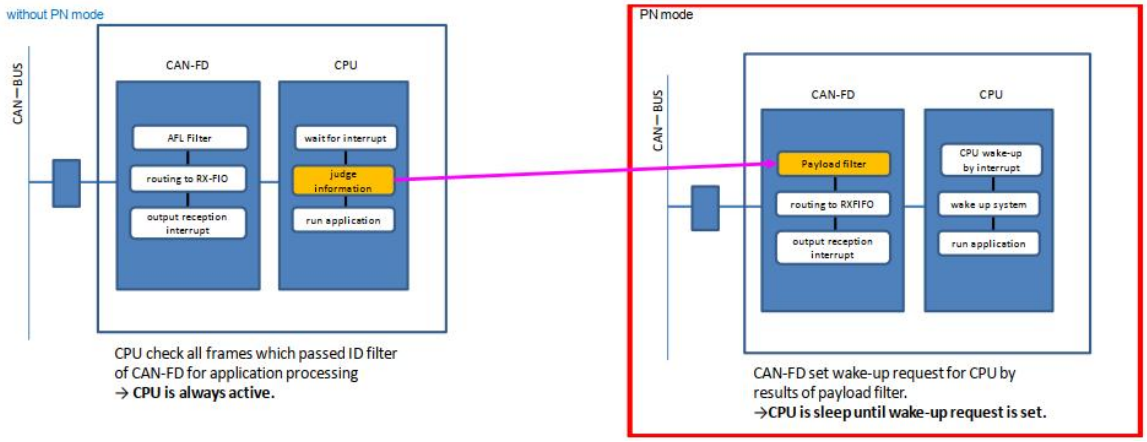


图 25-50 PN 模式的消息内容处理

PNF将接收到的消息ID和消息payload与滤波参数对比，PNF与AFL是独立工作的，因此会产生4种PNF和AFL配置组合，用户可以通过CFDC0FDCTR.PNMDC选择模式，并通过查询CFDC0FDSTS.PNSTS来确认当前模式。

Mode / Filter	Pretended Network Filter Mode (PN mode)	Normal Mode		
		Pretended Network Filter ID only and Acceptance Filter Mode	Acceptance Filter Mode	Pretended Network Filter and Acceptance Filter Mode
AFL Filter	OFF	ON	ON	ON
PN ID Filter	ON	ON	OFF	ON
PN Payload Filter	ON	OFF	OFF	ON
CFDC0FDCTR.PNMDC CFDC0FDSTS.PNSTS	11b	01b	00b	10b

表 25-16PNF 和 AFL 配置组合

25.14.1 Pretended Network Filter mode (PN mode)

PN mode下，接收到的消息ID和消息payload与PN filter比较，而AFL不起作用。

当匹配成功，RS-CANFD-Lite自动进入normal mode，并且可以唤醒CPU。下图是PN mode和normal mode之间切换的例子：

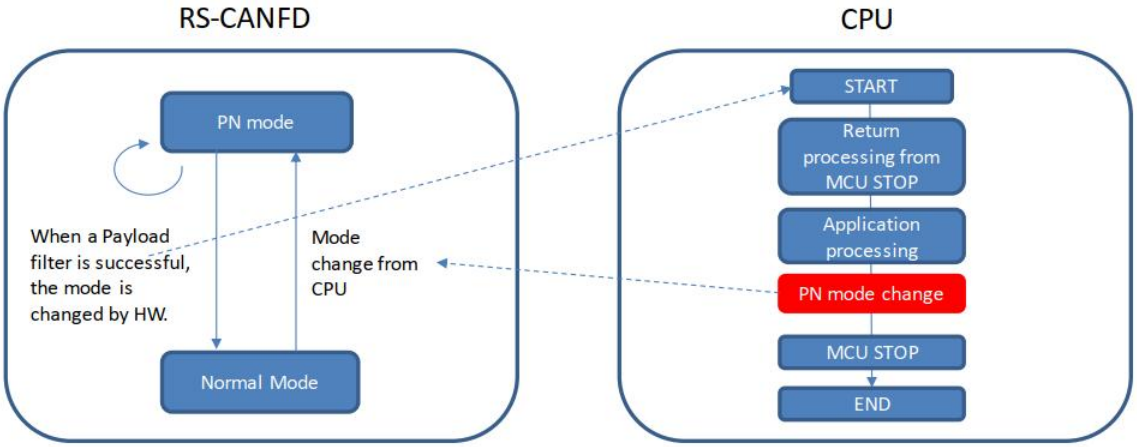


图 25-51 PN 模式处理流程

25.14.2 Pretended Network Filter ID only and Acceptance Filter mode

此模式下，接收消息的ID同时被PNF和AFL滤波，而消息payload不做滤波。

AFL	PNF		
ID Filter	ID Filter	Payload Filter	Routing
unmatch	unmatch	not perform	received message is rejected
not perform	match		use PNF information
match	unmatch		use AFL information

25.14.3 Acceptance Filter mode

这是不使能PNF时的正常工作模式。

AFL	PNF		
ID Filter	ID Filter	Payload Filter	Routing
unmatch	not perform	not perform	received message is rejected
match			use AFL information

25.14.4 Pretended Network Filter and Acceptance Filter mode

此模式下，接收消息的ID同时经过AFL和PNF滤波，接收消息的payload经过PNF滤波，当一个消息同时匹配AFL和PNF，PNF拥有优先级并进行payload比较。

AFL	PNF		
ID Filter	ID Filter	Payload Filter	Routing
unmatch	unmatch	not perform	received message is rejected
not perform	match	unmatch	received message is rejected
not perform	match	match	use PNF information
match	unmatch	not perform	use AFL information

25.14.5 滤波功能细节

25.14.5.1 PN ID filter

PN ID filter的功能与AFL的ID filter相同。

PNF行为与相关配置信号的关系：

Mirror mode enable (MME configuration bit)	Loopback in test mode (Selftest Mode 0 or Selftest Mode 1)	Channel Mode is	Loopback configuration bit in PNF Entry	PNF entry is
0	0	Receiver	0	valid
			1	invalid
		Transmitter	0	invalid
			1	invalid
	1	Receiver	0	valid
			1	invalid
		Transmitter	0	valid
			1	valid
1	0	Receiver	0	valid
			1	invalid
		Transmitter	0	invalid
			1	valid
	1	Receiver	0	valid
			1	invalid
		Transmitter	0	valid
			1	valid

表 25-17PNF 行为与配置信号关系

25.14.5.2 PN payload filter

payload filter在两个位置比较32bit数据内容，比较位置通过offset参数配置（offset参数以4字节为边界），DLC小于4的接收消息帧、远程帧都被拒绝。

如果offset设置超过了接受帧的payload长度，则当前帧被payload filter拒绝。

模块内有两个filter，每个filter支持两种比较方式：within range 和 out of range

within range: Lower limit <= payload value <= upper limit

out of range: Payload value < lower limit or upper limit < payload value

每个filter都可以选择比较方式，并且PNF可以选择比较通过的策略：同时满足两个filter，或者只满足其中一个filter。

25.14.5.3 PNF 配置

PNF entry配置保存在RAM中，共保存2个PNF entry：

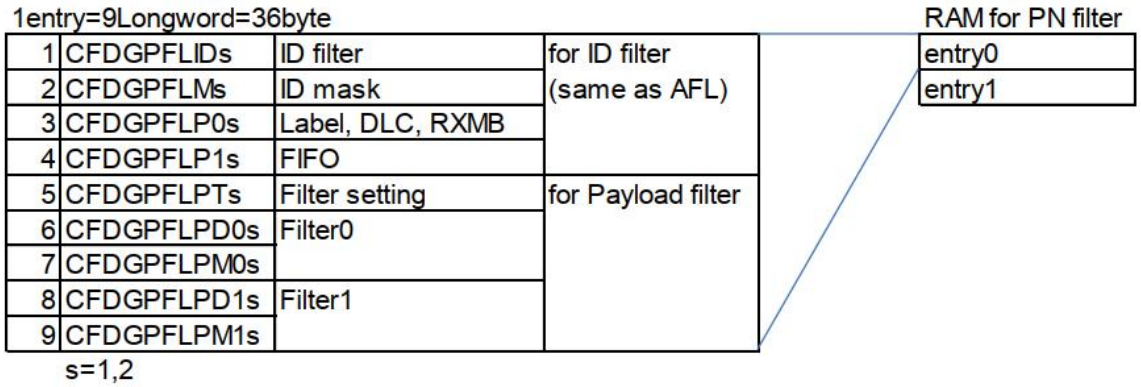


图 25-52 PNF 的 RAM 配置

25.14.5.4 payload filter 软件流程

配置payload filter的软件操作流程如下图所示：

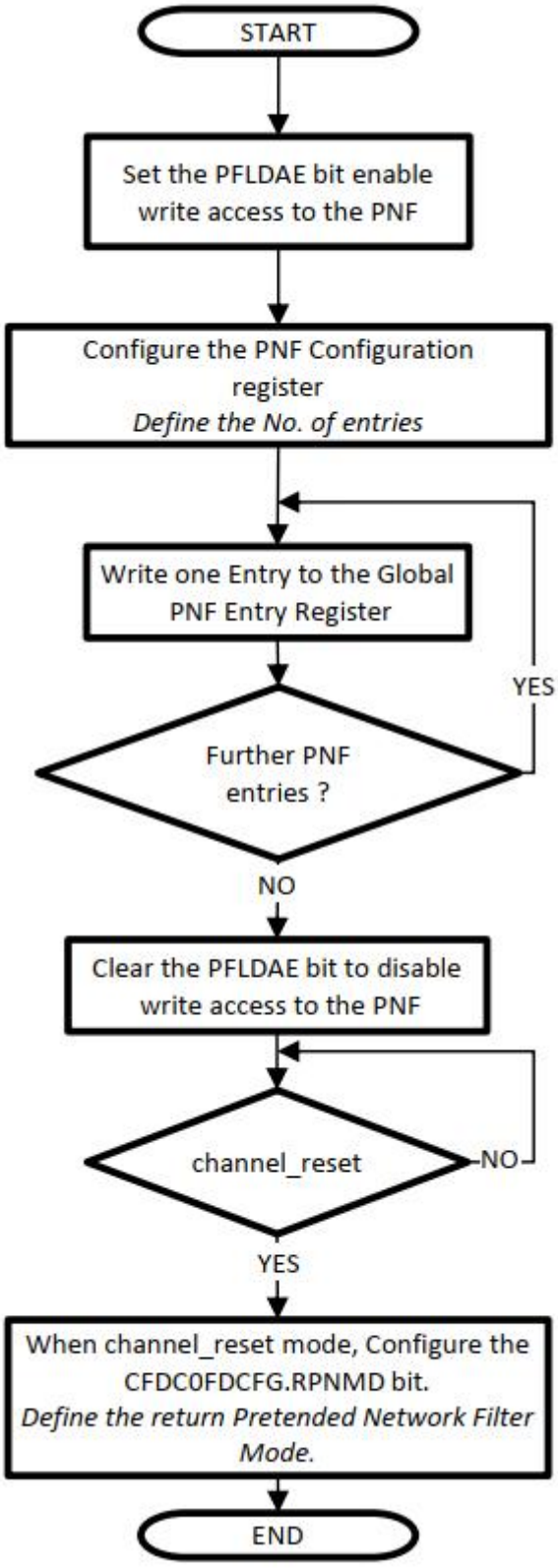


图 25-53 PNF 配置流程

在configuration mode下写入所有的entry之后，软件应锁定PNF访问，以保护PFL的意外改写。写保护开启后，软件仍然可以读取PFL以便于进行内容检查。

PN模式下的整体软件流程如下：

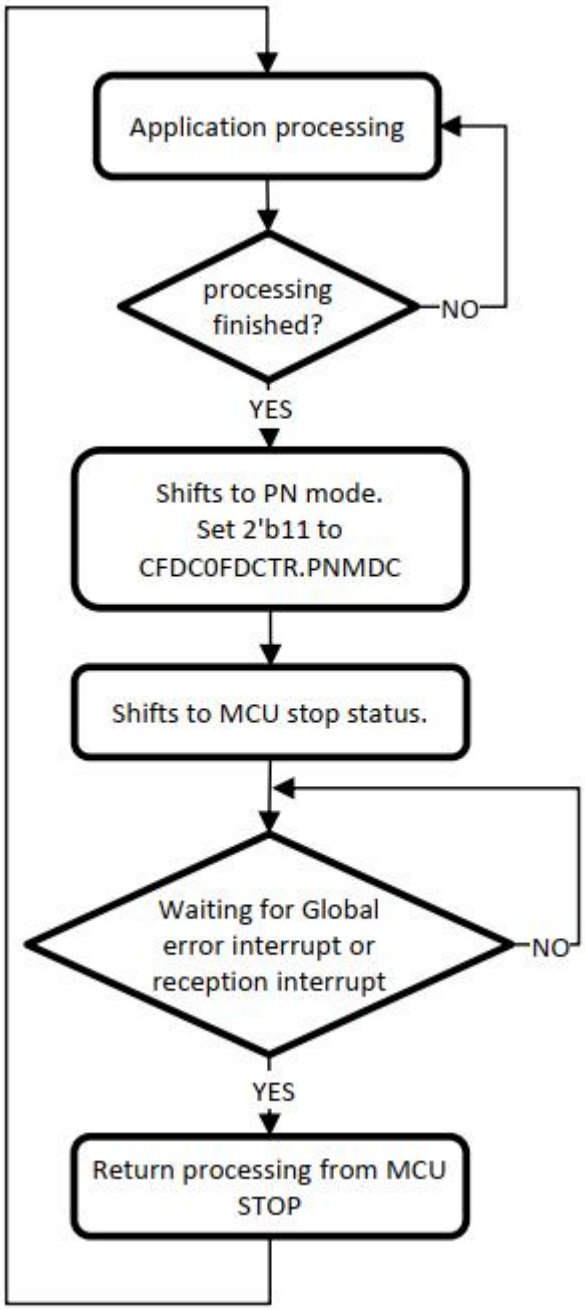


图 25-54 PNF 操作流程

从normal模式转移到PN模式的流程如下图：

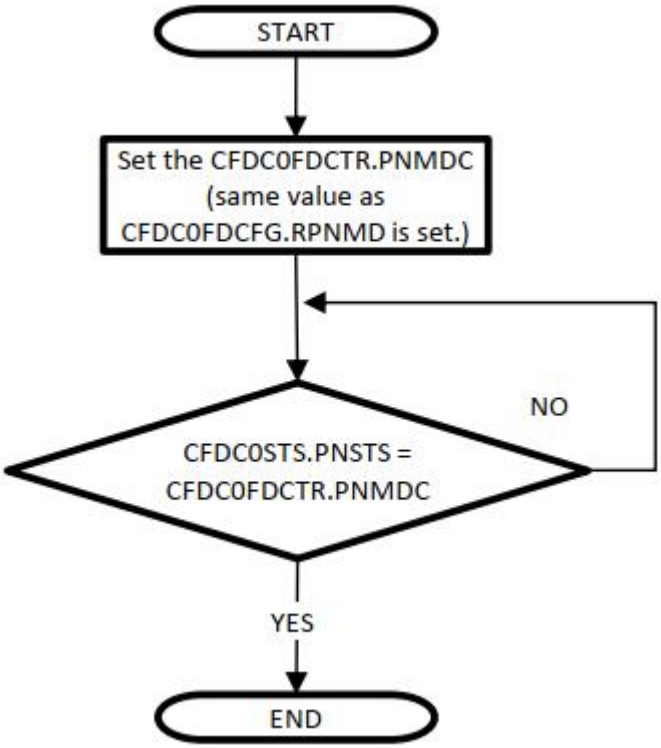


图 25-55 normal mode 转移

25.14.5.5 payload filter 字节对齐

CAN通信协议使用little endian格式，RS-CANFD-Lite模块也使用little endian格式，因此寄存器内的endian对齐如下图所示：

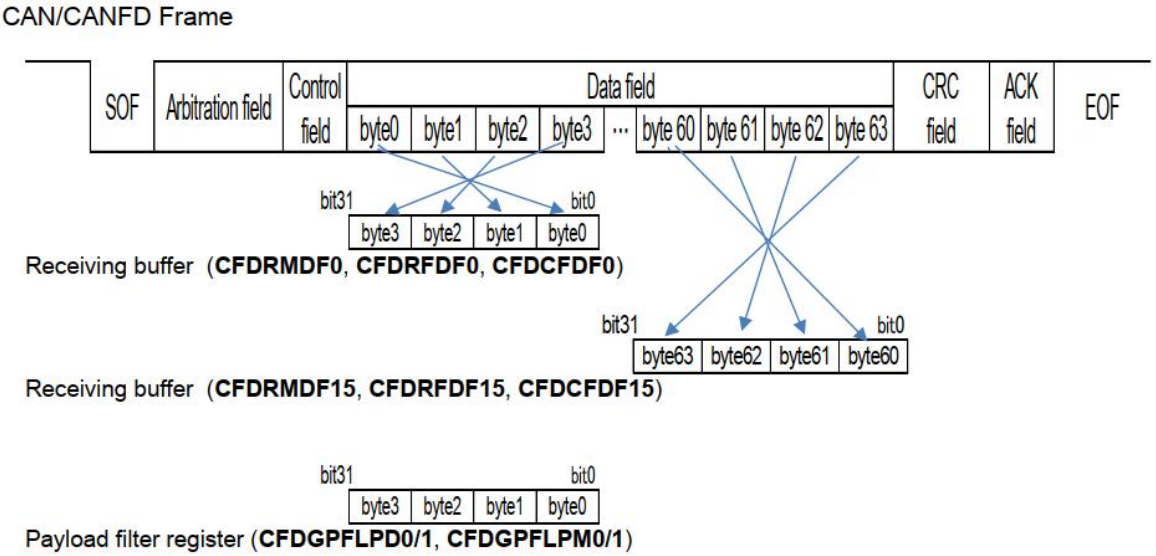


图 25-56 register endians

filter比较器使用以上格式进行内容比较。

25.15 异步唤醒

当芯片处于休眠状态下，系统没有CANCLK时钟，此时可以通过CAN_RX信号的下降沿（隐性电平->显性电平），并产生异步唤醒中断，将处于休眠模式的MCU唤醒。下降沿检测带有一个模拟滤波电路，能够滤除小于100ns的毛刺信号，避免噪声干扰引起误唤醒。实现方法是在CAN_RX引脚上曾WKUP引脚异步唤醒功能，并将唤醒条件配置为输入下降沿。

借助此功能，可以在芯片休眠状态下实现CAN总线活动唤醒MCU。使用流程如下。

- 等待总线空闲
- 使能CAN_RX引脚的异步唤醒功能，并置位唤醒中断使能寄存器（GPIO模块）
- 配置CANFD进入sleep, reset, halt mode
- 芯片进入休眠
- CAN总线下降沿置位唤醒中断，唤醒MCU
- MCU退出休眠，执行CANFD初始化
- 软件配置CANFD进入operation mode

25.16 寄存器

offset 地址	名称	符号
	RS-CANFD-Lite （模块基地址：）	
0000h	Channel 0 Nominal Bitrate Configuration Register	CFDC0NCFG
0004h	Channel 0 Control Register	CFDC0CTR
0008h	Channel 0 Status Register	CFDC0STS
000Ch	Channel 0 Error Flag Register	CFDC0ERFL
0010h	Global IP Version Register	CFDGIPV
0014h	Global Configuration Register	CFDGCFG
0018h	Global Control Register	CFDGCTR
001Ch	Global Status Register	CFDGSTS
0020h	Global Error Flag Register	CFDGERFL
0024h	Global Timestamp Counter Register	CFDGTSC
0028h	Global Acceptance Filter List Entry Control Register	CFDGAFLECTR
002Ch	Global Acceptance Filter List Configuration Register	CFDGAFLCFG
0030h	RX Message Buffer Number Register	CFDRMNB
0034h	RX Message Buffer New Data Register	CFDRMND
0038h	RX Message Buffer Interrupt Enable Configuration Register	CFDRMIEC
003Ch + a*0004h	RX FIFO Configuration / Control Registers a = [0:1]	CFDRFCCa
0044h + a*0004h	RX FIFO Status Registers a = [0:1]	CFDRFSTSa
004Ch + a*0004h	RX FIFO Pointer Control Registers a = [0:1]	CFDRFPCTRa
0054h	Common FIFO Configuration / Control Register	CFDCFCC
0058h	Common FIFO Status Register	CFDCFSTS
005Ch	Common FIFO Pointer Control Register	CFDCFPCTR
0060h	FIFO Empty Status Register	CFDFESTS
0064h	FIFO Full Status Register	CFDFFSTS
0068h	FIFO Message Lost Status Register	CFDFMSTS
006Ch	RX FIFO Interrupt Flag Status Register	CFDRFISTS
0070h + i*0001h	TX Message Buffer Control Registers I = [0:3]	CFDTMCi
0074h + j*0001h	TX Message Buffer Status Registers j = [0:3]	CFDTMSTSj
0078h	TX Message Buffer Transmission Request Status Register	CFDTMTRSTS
007Ch	TX Message Buffer Transmission Abort Request Status Register	CFDTMTARSTS
0080h	TX Message Buffer Transmission Completion Status Register	CFDTMTCSTS
0084h	TX Message Buffer Transmission Abort Status Register	CFDTMTASTS
0088h	TX Message Buffer Interrupt Enable Configuration Register	CFDTMIEC
008Ch	TX Queue Configuration / Control Register	CFDTXQCC

0090h	TX Queue Status Register	CFDTXQSTS
0094h	TX Queue Pointer Control Register	CFDTXQPCTR
0098h	TX History List Configuration / Control Register	CFDTHLCC
009Ch	TX History List Status Register	CFDTHLSTS
00A0h	TX History List Pointer Control Register	CFDTHLPCTR
00A4h	Global TX Interrupt Status Register	CFDGTINTSTS
00A8h	Global Test Configuration Register	CFDGTSTCFG
00ACh	Global Test Control Register	CFDGTSTCTR
00B0h	Global FD Configuration register	CFDGFDCFG
00B8h	Global Lock Key Register	CFDGLOCKK
00C0h	Global AFL Ignore Entry Register	CFDGAFLIGNENT
00C4h	Global AFL Ignore Control Register	CFDGAFLIGNCTR
00C8h	DMA Transfer Control Register	CFDCDTCT
00CCh	DMA Transfer Status Register	CFDCDTSTS
00D0h	Pretended Network Filter List Entry control Register	CFDGPFLECTR
00D4h	Pretended Network Filter List Entry Configuration Register	CFDGPFLCFG
00D8h	Global SW reset Register	CFDGRSTC
0100h	Channel 0 Data Bitrate Configuration Register	CFDC0DCFG
0104h	Channel 0 CAN-FD Configuration Register	CFDC0FDCFG
0108h	Channel 0 CAN-FD Control Register	CFDC0FDCTR
010Ch	Channel 0 CAN-FD Status Register	CFDC0FDSTS
0110h	Channel 0 CAN-FD CRC Register	CFDC0FDCRC
0120h + (r-1)*0010h	Global Acceptance Filter List ID Registers r = [1…10]h	CFDGAFLIDr
0124h + (r-1)*0010h	Global Acceptance Filter List Mask Registers r = [1…10]h	CFDGAFLMr
0128h + (r-1)*0010h	Global Acceptance Filter List Pointer 0 Registers r = [1…10]h	CFDGAFLP0r
012Ch + (r-1)*0010h	Global Acceptance Filter List Pointer 1 Registers r = [1…10]h	CFDGAFLP1r
0220h + (s-1) * 0024h	Global Pretended Network Filter List ID Registers s = [1…2]h	CFDGPFLIDs
0224h + (s-1) * 0024h	Global Pretended Network Filter List Mask Registers s = [1…2]h	CFDGPFLMs
0228h + (s-1) * 0024h	Global Pretended Network Filter List Pointer 0 Registers s = [1…2]h	CFDGPFLP0s
022Ch + (s-1) * 0024h	Global Pretended Network Filter List Pointer 1 Registers s = [1…2]h	CFDGPFLP1s
0230h + (s-1) * 0024h	Global Pretended Network Filter List Filter Payload Type Registers s = [1…2]h	CFDGPFLPTs
0234h + (s-1) * 0024h	Global Pretended Network Filter List Payload Data 0 Registers s = [1…2]	CFDGPFLPD0s
0238h + (s-1) * 0024h	Global Pretended Network Filter List Payload Mask 0 Registers s = [1…2]h	CFDGPFLPM0s

023Ch + (s-1) * 0024h	Global Pretended Network Filter List Payload Data 1 Registers s = [1…2]h	CFDGPFLPD1s
0240h + (s-1) * 0024h	Global Pretended Network Filter List Payload Mask 1 Registers s = [1…2]h	CFDGPFLPM1s
0280h + k * 0004h	RAM Test Page Access Registers k = [0…3F]h	CFDRPGACCk
0520h + b * 004Ch	RX FIFO Access ID Registers b = [0…1]h	CFDRFIDb
0524h + b * 004Ch	RX FIFO Access Pointer Registers b = [0…1]h	CFDRFPTRb
0528h + b * 004Ch	RX FIFO Access CAN-FD Status Registers b = [0…1]h	CFDRFFDSTSb
052Ch + p * 0004h + b * 004Ch	RX FIFO Access Data Field p Registers b = [0…1]h p = [0…F]h	CFDRFDFbp
05B8h	Common FIFO Access ID Register	CFDCFID
05BCh	Common FIFO Access Pointer Register	CFDCFPTR
05C0h	Common FIFO Access CAN-FD Control/Status Register	CFDCFFDCSTS
05C4h + p * 0004h	Common FIFO Access Data Field p Registers p = [0…F]h	CFDCFDFp
0604h + b * 004Ch	TX Message Buffer ID Registers b = [0…3]h	CFDTMIDb
0608h + b * 004Ch	TX Message Buffer Pointer Registers b = [0…3]h	CFDTMPTRb
060Ch + b * 004Ch	TX Message Buffer CAN-FD Control Registers b = [0…3]h	CFDTMFDCTRb
0610h + p * 0004h + b * 004Ch	TX Message Buffer Data Field p Registers b = [0…3]h p = [0…F]h	CFDTMDFbp
0740h	Channel 0 TX History List Access Registers 0	CFDTHLACC0
0744h	Channel 0 TX History List Access Registers 1	CFDTHLACC1
0920h + b * 004Ch	RX Message Buffer ID Registers b = [0…7]h	CFDRMIDb
0924h + b * 004Ch	RX Message Buffer Pointer Registers b = [0…7]h	CFDRMPTRb
0928h + b * 004Ch	RX Message Buffer CAN-FD Status Registers b = [0…7]h	CFDRMFDSTSb
092Ch + p * 0004h + b * 004Ch	RX Message Buffer Data Field p Registers b = [0…7]h p = [0…F]h	CFDRMDFbp
0D20h + (b – 8) * 004Ch	RX Message Buffer ID Registers b = [8…15]h	CFDRMIDb
0D24h + (b – 8) * 004Ch	RX Message Buffer Pointer Registers b = [8…15]h	CFDRMPTRb
0D28h + (b – 8) * 004Ch	RX Message Buffer CAN-FD Status Registers b = [8…15]h	CFDRMFDSTSb
0D2Ch + p * 0004h + (b 004Ch – 8) *	RX Message Buffer Data Field p Registers b = [8…15]h p = [0…F]h	CFDRMDFbp
1120h + (b – 16) * 004Ch	RX Message Buffer ID Registers b = [16…23]h (Only RX Max version)	CFDRMIDb
1124h + (b – 16) * 004Ch	RX Message Buffer Pointer Registers b = [16…23]h (Only RX Max version)	CFDRMPTRb
1128h + (b – 16) * 004Ch	RX Message Buffer CAN-FD Status Registers b = [16…23]h (Only RX Max version)	CFDRMFDSTSb
112Ch + p * 0004h + (b 004Ch – 16) *	RX Message Buffer Data Field p Registers b = [16…23]h p = [0…F]h (Only RX Max version)	CFDRMDFbp
1520h + (b – 24) * 004Ch	RX Message Buffer ID Registers b = [24…31]h (Only RX Max version)	CFDRMIDb
1524h + (b – 24) * 004Ch	RX Message Buffer Pointer Registers b = [24…31]h (Only RX Max version)	CFDRMPTRb
1528h + (b – 24) *	RX Message Buffer CAN-FD Status Registers b =	CFDRMFDSTSb

004Ch	[24…31]h (Only RX Max version)	
152Ch + p * 0004h + (b 004Ch – 24) *	RX Message Buffer Data Field p Registers b = [24…31]h p = [0…F]h (Only RX Max version)	CFDRMDFbp

25.16.1 Channel 0 nominal bitrate configuration register (CFDC0NCFG)

名称	CFDC0NCFG							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	NTSEG2[6:0]							NTSEG1[7]
位权限	R/W-000 0000							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	NTSEG1[6:0]							NSJW[6]
位权限	R/W-000 0000							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	NSJW[5:0]						NBRP[9:8]	
位权限	R/W-000 000						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	NBRP[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:25	NTSEG2	Timing segment2 00h: RFU 01h: 2Tq … 7Eh: 127Tq 7Fh: 128Tq 此寄存器不能在 CH_OPERATION 和 CH_SLEEP 模式下改写，用户只能在 CH_RESET 和 CH_HALT 模式下改写
24:17	NTSEG1	Timing segment1 00h: RFU 01h: 2Tq 02h: 3Tq 03h: 4Tq ... FEh: 255Tq FFh: 256Tq 此寄存器不能在 CH_OPERATION 和 CH_SLEEP 模式下改写，用户只能在 CH_RESET 和 CH_HALT 模式下改写
16:10	NSJW	Resynchronization jump width 00h: 1Tq 01h: 2Tq ... 7Eh: 127Tq 7Fh: 128Tq 此寄存器不能在 CH_OPERATION 和 CH_SLEEP 模式下改写，用户只能在 CH_RESET 和 CH_HALT 模式下改写
9:0	NBRP	nominal baud rate prescaler division ration 分频系数为 NBRP+1

位号	位名	说明
		此寄存器不能在 CH_OPERATION 和 CH_SLEEP 模式下改写，用户只能在 CH_RESET 和 CH_HALT 模式下改写

25.16.2 Channel 0 Control Register (CFDC0CTR)

名称	CFDC0CTR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ROM	BFT	-			CTMS		CTME
位权限	R/W-0	R/W-0	U-0			R/W-0		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ERRD	BOM		-	TDCVFIE	SOCOIE	EOCOIE	TAIE
位权限	R/W-0	R/W-00		U-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				RTBO	CSLPR	CHMDC	
位权限	U-0				R/W-0	R/W-1	R/W-01	

位号	位名	说明
31	ROM	Restricted operation mode 0: Restricted operation mode disabled 1: Restricted operation mode enabled 软件只能在 CH_HALT 模式下改写，控制器进入 CH_RESET 模式时自动清零
30	BFT	Bit Flip Test 0：接收码流的第一个数据 bit 不取反 1：接收码流的第一个数据 bit 取反 软件只能在 CH_HALT 模式下改写，控制器进入 CH_RESET 模式时自动清零
29:27	--	RFU：未实现，读为 0
26:25	CTMS	Channel test mode select 00: basic test mode 01: listen-only mode 10: self test mode 0 (external loopback) 10: self test mode 1 (internal loopback) 软件只能在 CH_HALT 模式下改写，控制器进入 CH_RESET 模式时自动清零
24	CTME	channel test mode enable 0: channel test mode disabled 1: channel test mode enabled 软件只能在 CH_HALT 模式下改写，控制器进入 CH_RESET 模式时自动清零
23	ERRD	channel error display 0：只显示第一组 error code 1：显示累积 error code 软件只能在 CH_RESET 或 CH_HALT 模式下改写
22:21	BOM	channel bus-off mode

位号	位名	说明
		00：normal mode (ISO11898-1) 01：bus-off 开始后自动进入 halt mode 10：bus-off 结束后自动进入 halt mode 11：bus-off 恢复阶段内可以由软件控制进入 halt mode 软件只能在 CH_RESET 模式下改写
20	--	RFU：未实现，读为 0
19	TDCVFIE	Transmission delay compensation violation interrupt enable 0：禁止中断 1：使能中断 软件只能在 CH_RESET 模式下改写
18	SOCOIE	Successful Occurrence Counter Overflow Interrupt enable 1’b0: Successful occurrence counter overflow interrupt disabled 1’b1: Successful occurrence counter overflow interrupt enabled 软件只能在 CH_RESET 模式下改写
17	EOCOIE	Error occurrence counter overflow Interrupt enable 1’b0: Error occurrence counter overflow Interrupt disabled 1’b1: Error occurrence counter overflow Interrupt enabled 软件只能在 CH_RESET 模式下改写
16	TAIE	Transmission abort Interrupt Enable 1’b0: TX abort Interrupt disabled 1’b1: TX abort Interrupt enabled 软件只能在 CH_RESET 模式下改写
15	ALIE	Arbitration Lost Interrupt Enable 1’b0: Arbitration Lost Interrupt disabled 1’b1: Arbitration Lost Interrupt enabled 软件只能在 CH_RESET 模式下改写
14	BLIE	Bus Lock Interrupt Enable 1’b0: Bus Lock Interrupt disabled 1’b1: Bus Lock Interrupt enabled 软件只能在 CH_RESET 模式下改写
13	OLIE	Overload Interrupt Enable 1’b0: Overload Interrupt disabled 1’b1: Overload Interrupt enabled 软件只能在 CH_RESET 模式下改写
12	BORIE	Bus-Off Recovery Interrupt Enable 1’b0: Bus-Off Recovery Interrupt disabled 1’b1: Bus-Off Recovery Interrupt enabled 软件只能在 CH_RESET 模式下改写
11	BOEIE	Bus-Off Entry Interrupt Enable 1’b0: Bus-Off Entry Interrupt disabled 1’b1: Bus-Off Entry Interrupt enabled 软件只能在 CH_RESET 模式下改写
10	EPIE	Error Passive Interrupt Enable 1’b0: Error Passive Interrupt disabled 1’b1: Error Passive Interrupt enabled 软件只能在 CH_RESET 模式下改写
9	EWIE	Error Warning Interrupt Enable 1’b0: Error Warning Interrupt disabled 1’b1: Error Warning Interrupt enable 软件只能在 CH_RESET 模式下改写
8	BEIE	Bus Error Interrupt Enable 1’b0: Bus Error Interrupt disabled 1’b1: Bus Error Interrupt enabled

位号	位名	说明
		软件只能在 CH_RESET 模式下改写
7:4	--	RFU：未实现，读为 0
3	RTBO	Return from Bus-Off 1'b0: Channel is not forced to return from Bus-Off 1'b1: Channel is forced to return from Bus-Off 软件置位此寄存器将使 TEC 和 REC 寄存器被初始化，Bus-off 恢复中断不会产生，此寄存器软件读回总是 0 RTBO 功能只能在 BOM=00 的情况下使用 此寄存器只能在 CH_OPERATION 模式下置位，在非 bus-off 状态下置位这个寄存器无效，并且会被立即自动清零
2	CSLPR	Channel Sleep Request 1'b0: Channel Sleep Request disabled 1'b1: Channel Sleep Request enabled 用户只能在 CH_RESET 或 CH_SLEEP 模式下改写此寄存器
1:0	CHMDC	channel mode control 2'b00: Channel Operation Mode request 2'b01: Channel Reset request 2'b10: Channel Halt request 2'b11: Keep current value 此寄存器在 CAN 控制器进入 halt mode 时会被自动改写

25.16.3 Channel 0 Status Register(CFDC0STS)

名称	CFDC0STS							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TEC							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	REC							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							ESIF
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
位权限	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

位号	位名	说明
31:24	TEC	Transmission Error Count This register increments or decrements the counter value I error status of the CAN channel during Transmission.
23:16	REC	Reception Error Count This register increments or decrements the counter value I error status of the CAN channel during Reception.
15:9	--	RFU：未实现，读为 0
8	ESIF	Error State Indication Flag 1'b0: No CAN-FD message has been received with the ESI flag was set 1'b1: At least 1 CAN-FD message was received where the ESI flag

		was set
7	COMSTS	Channel Communication Status 1'b0: Channel is not ready for communication 1'b1: Channel is ready for communication
6	RECSTS	Channel Receive Status 1'b0: Channel is not receiving 1'b1: Channel is receiving
5	TRMSTS	Channel Transmit Status 1'b0: Channel is not transmitting 1'b1: Channel is transmitting
4	BOSTS	Channel Bus-Off Status 1'b0: Channel not in Bus-Off state 1'b1: Channel in Bus-Off state
3	EPSTS	Channel Error Passive Status 1'b0: Channel not in Error Passive state. 1'b1: Channel in Error Passive state.
2	CSLPSTS	Channel SLEEP Status 1'b0: Channel not in Sleep Mode 1'b1: Channel in Sleep Mode
1	CHLTSTS	Channel HALT Status 1'b0: Channel not in Halt Mode 1'b1: Channel in Halt Mode
0	CRSTSTS	Channel RESET Status 1'b0: Channel not in Reset Mode 1'b1: Channel in Reset Mode

25.16.4 Channel 0 Error Flag Register (CFDC0ERFL)

名称	CFDC0ERFL							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	CRCREG[14:8]						
位权限	U-0	R/W-000 0000						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CRCREG[7:0]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31	--	RFU：未实现，读为 0
30:16	CRCREG	CRC Register value 当 CFDC0CTR.CTME=1 时，这些寄存器保存 CRC 计算结果 当 CFDC0CTR.CTME=0 时，这些寄存器读出总是 0 当通道进入 CH_RESET 模式后此寄存器自动清零
15	--	RFU：未实现，读为 0
14	ADERR	Acknowledge Delimiter Error 1'b0: Channel Ack Del Error not detected

		1'b1: Channel Ack Del Error detected 此寄存器置位代表在 ack 分割符处检测到格式错误 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零 清零此寄存器需要遵循以下流程： 1. 对寄存器写 0 2. 读出寄存器查询是否清零成功 3. 如果没有清零，则返回 step1
13	B0ERR	Bit 0 Error 1'b0: Channel Bit 0 Error not detected 1'b1: Channel Bit 0 Error detected 此寄存器置位代表在预期接收显性位时收到了隐性位 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零 清零此寄存器需要遵循以下流程： 4. 对寄存器写 0 5. 读出寄存器查询是否清零成功 6. 如果没有清零，则返回 step1
12	B1ERR	Bit 1 Error 1'b0: Channel Bit 1 Error not detected 1'b1: Channel Bit 1 Error detected 此寄存器置位代表在预期接收隐性位时收到了显性位 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零 清零此寄存器需要遵循以下流程： 1. 对寄存器写 0 2. 读出寄存器查询是否清零成功 3. 如果没有清零，则返回 step1
11	CERR	CRC Error 1'b0: Channel CRC Error not detected 1'b1: Channel CRC Error detected 此寄存器置位代表检测到总线 CRC error 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零 清零此寄存器需要遵循以下流程： 1. 对寄存器写 0 2. 读出寄存器查询是否清零成功 3. 如果没有清零，则返回 step1
10	AERR	Acknowledge Error 1'b0: Channel Ack Error not detected 1'b1: Channel Ack Error detected 此寄存器置位代表检测到总线 acknowledge error 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零 清零此寄存器需要遵循以下流程： 1. 对寄存器写 0 2. 读出寄存器查询是否清零成功 3. 如果没有清零，则返回 step1
9	FERR	Form Error 1'b0: Channel Form Error not detected 1'b1: Channel Form Error detected 此寄存器置位代表检测到总线 form error

		硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零 清零此寄存器需要遵循以下流程： 1. 对寄存器写 0 2. 读出寄存器查询是否清零成功 3. 如果没有清零，则返回 step1
8	SERR	Stuff Error 1'b0: Channel stuff Error not detected 1'b1: Channel stuff Error detected 此寄存器置位代表检测到总线 stuffing error 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零
7	ALF	Arbitration Lost Flag 1'b0: Channel Arbitration Lost not detected 1'b1: Channel Arbitration Lost detected 此寄存器置位代表检测到总线仲裁失败 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零
6	BLF	Bus Lock Flag 1'b0: Channel Bus Lock not detected 1'b1: Channel Bus Lock detected 此寄存器置位代表总线进入 bus lock condition（检测到总线上连续 32 个显性位） 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零
5	OVLF	Overload Flag 1'b0: Channel Overload not detected 1'b1: Channel Overload detected 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零
4	BORF	Bus-Off Recovery Flag 1'b0: Channel Bus-Off Recovery not detected 1'b1: Channel Bus-Off Recovery detected 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零 当 BOM=00/10/11，此寄存器在 CAN 通道从 bus-off 中恢复时自动置位；在以下条件下，此寄存器在 CAN 通道从 bus-off 中恢复时不会自动置位： ● CFDC0CTR.RTBO 软件置位 ● CFDC0CTR.BOM=01 ● CFDC0CTR.BOM=11 且 halt request 在 bus-off 结束前置位
3	BOEF	Bus-Off Entry Flag 1'b0: Channel Bus-Off Entry not detected 1'b1: Channel Bus-Off Entry detected 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零
2	EPF	Error Passive Flag 1'b0: Channel Error Passive not detected 1'b1: Channel Error Passive detected 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零 当 TEC 或 REC 值从小于等于 127 跨越到大于 127 时，此寄存器自动置位

1	EWF	Error Warning Flag 1'b0: Channel Error Warning not detected 1'b1: Channel Error Warning detected 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零 当 TEC 或 REC 值从小于等于 95 跨越到大于 95 时，此寄存器自动置位
0	BEF	Bus Error Flag 1'b0: Channel Bus Error not detected 1'b1: Channel Bus Error detected 硬件置位，软件写 0 清零，当通道进入 CH_RESET 模式后此寄存器自动清零

25.16.5 Channel 0 Data Bitrate Configuration Register (CFDC0DCFG)

名称	CFDC0DCFG							
Offset	0x0100							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				DSJW			
位权限	U-0				R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				DTSEG2			
位权限	U-0				R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			DTSEG1				
位权限	U-0			R/W-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DBRP							
位权限	R/W-0000 0000							

位号	位名	说明
31:28	--	RFU：未实现，读为 0
27:24	DSJW	Resynchronization Jump Width 4'b0000: 1 Tq 4'b0001: 2 Tq … 4'b1111: 16 Tq 此寄存器只能在 CH_RESET 或 CH_HALT 模式下改写
23:20	--	RFU：未实现，读为 0
19:16	DTSEG2	Timing Segment 2 4'b0000: Reserved 4'b0001: 2 Tq … 4'b1110: 15 Tq 4'b1111: 16 Tq 此寄存器只能在 CH_RESET 或 CH_HALT 模式下改写
15:13	--	RFU：未实现，读为 0
12:8	DTSEG1	Timing Segment 1 5'b00000: Reserved 5'b00001: 2Tq 5'b00010: 3Tq 5'b00011: 4Tq

		… 5'b11110: 31 Tq 5'b11111: 32 Tq 此寄存器只能在 CH_RESET 或 CH_HALT 模式下改写
7:0	DBRP	Channel Data Baud Rate Prescaler 数据波特率预分频系数

25.16.6 Channel 0 CAN-FD Configuration Register (CFDC0FDCFG)

名称	CFDC0FDCFG							
Offset	0x0104							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	CLOE	REFE	FDOE	-			
位权限	U-0	R/W-0	R/W-0	R/W-0	U-0			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TDCO							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		RPNMD		-	ESIC	TDCE	TDCOC
位权限	U-0		R/W-00		U-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					EOCCFG		
位权限	U-0					R/W-000		

位号	位名	说明
31	--	RFU：未实现，读为 0
30	CLOE	Classical CAN only enable 1'b0: Classical Classical only mode disabled 1'b1: Classical Classical only mode enabled 此寄存器置位后，CAN 控制器只能发送经典帧，收到 FD 帧将被认为 CRC 错误或者格式错误 用户要避免同时置位 CLOE 和 FDOE 此寄存器不能在 CH_OPERATION, CH_HALT, CH_SLEEP 模式下改写
29	REFE	RX edge filter enable 1'b0: RX edge filter disabled 1'b1: RX edge filter enabled 接收信号边沿滤波仅用于 IDLE 检测（总线集成阶段） 此寄存器不能在 CH_OPERATION, CH_HALT, CH_SLEEP 模式下改写
28	FDOE	FD only enable 1'b0: FD only mode disabled 1'b1: FD only mode enabled 此寄存器置位后经典 CAN 帧传输被禁止，此时如果收到经典 CAN frame，控制器将其视为无效帧并回复错误帧；如果在 FDOE=1 的情况下发送经典帧，则 FDF 位总是发送隐性位，如果 DLC 配置大于 8 则数据填充 CC 此寄存器不能在 CH_OPERATION, CH_HALT, CH_SLEEP 模式下改写
27:24	--	RFU：未实现，读为 0
23:16	TDCO	Transceiver Delay Compensation Offset

		Transceiver delay compensation offset value 实际的 offset = (TDCO+1) * T_{DLL_CLK} 此寄存器只能在 CH_RESET 或者 CH_HALT 模式下改写
15:14	--	RFU：未实现，读为 0
13:12	RPNMD	Return Pretended Network Filter Mode 2'b00: Return to Acceptance Filter Mode 2'b01: Return to Pretended Network Filter ID only and Acceptance Filter Mode 2'b10: Return to Pretended Network Filter and Acceptance Filter Mode 2'b11: Return to Pretended Network Filter Mode (Not return)
11	--	RFU：未实现，读为 0
10	ESIC	Error State Indication Configuration 1'b0: The ESI bit in the frame will be representing the Error state of the node itself 1'b1: The ESI bit in the frame will be representing the Error state of message buffer if the node itself is not in error passive. If the node is in Error Passive then the ESI bit will be driven by the node itself
9	TDCE	Transceiver Delay Compensation Enable 1'b0: Transceiver Delay Compensation disabled 1'b1: Transceiver Delay Compensation enabled
8	TDCOC	TDCOC Transceiver Delay Compensation Offset Configuration 1'b0: Measured + offset 1'b1: offset only
7:3	--	RFU：未实现，读为 0
2:0	EOCCFG	Error Occurrence Counter Configuration 3'b000: All Transmitter or Receiver CAN Frames 3'b001: All Transmitter CAN Frames 3'b010: All Receiver CAN Frames 3'b011: Reserved 3'b100: Only Transmitter or Receiver CAN-FD Data-Phase (fast bits) 3'b101: Only Transmitter CAN-FD Data-Phase (fast bits) 3'b110: Only Receiver CAN-FD Data-Phase (fast bits) 3'b111: Reserved

25.16.7 Channel 0 CAN-FD Control Register (CFDC0FDCTR)

名称	CFDC0FDCTR							
Offset	0x0108							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	KEY							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						PNMDC	
位权限	U-0						R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						SOCCLR	EOCCLR
位权限	U-0						R/W-0	R/W-0

位号	位名	说明
31:24	KEY	Key code These bits control the right or wrong of rewriting of PNMDC bit. 对此寄存器写入 C4h 后，PNMDC 才可写，读回总是 00 软件应对 KEY 和 PNMDC 同时进行写操作
23:18	--	RFU：未实现，读为 0
17:16	PNMDC	Pretended Network Filter Mode Control 2'b00: Acceptance Filter Mode request 2'b01: Pretended Network Filter ID only and Acceptance Filter Mode request 2'b10: Pretended Network Filter and Acceptance Filter Mode request 2'b11: Pretended Network Filter Mode request 用户应只对此寄存器写 11 或者 CFDC0FDCFG.RPNMD 的值 在 PN 模式下，如果接收帧通过 PNF，则硬件自动将此寄存器设置为 CFDC0FDCFG.RPNMD 的值
15:2	--	RFU：未实现，读为 0
1	SOCCLR	Successful Occurrence Counter Clear 1'b0: No Successful Occurrence Counter clear 1'b1: Clear Successful Occurrence Counter 软件置 1 清除 SOC，读回总是 0, 此寄存器不能在 CH_SLEEP 或 CH_RESET 模式下置位
0	EOCCLR	Error Occurrence Counter Clear 1'b0: No Error Occurrence Counter clear 1'b1: Clear Error Occurrence Counter 软件置 1 清除 EOC，读回总是 0, 此寄存器不能在 CH_SLEEP 或 CH_RESET 模式下置位

25.16.8 Channel 0 CAN-FD Status Register (CFDC0FDSTS)

名称	CFDC0FDSTS							
Offset	0x010C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SOC							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EOC							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TDCVF	-	PNSTS		-		SOCO	EOCO
位权限	R/W-0	U-0	R-00		U-0		R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TDCR							
位权限	R-0000 0000							

位号	位名	说明
31:24	SOC	Successful occurrence counter register These bits show the successful occurrence counter value. These bits are updated (incremented) when the occurrence of any

位号	位名	说明
		error-free messages on the bus isdetected. Any means (received or transmitted). When the counter reaches the value of 8'hFF then updatingis stopped. Note in case of Loopback mode the counter would be incremented twice. These bits are cleared by writing 1'b1 to CFDC0FDCTR.SOCCLR These bits are cleared automatically when the related RS-CANFD LITE channel is in CH_RESET mode.
23:16	EOC	Error occurrence counter register These bits show the error occurrence counter value. This counter is used together with the CFDC0FDSTS.SOC counter to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing reduced payload bit lengthexperience significantly higher error rates compared to other messages. These bits are updated (incremented) when an error occurs, according to the configuration of theCFDC0FDCFG.EOCCFG bits. When the counter reaches the value of 8'hFF then updating is stopped. These bits are cleared by writing 1'b1 to CFDC0FDCTR.EOCCLR These bits are cleared automatically when the related RS-CANFD LITE channel is in CH_RESET mode
15	TDCVF	Transceiver Delay Compensation Violation Flag 1'b0: Transceiver Delay Compensation Violation has not occurred 1'b1: Transceiver Delay Compensation Violation has occurred This bit is set automatically when the Transceiver Delay Compensation is greater than the max. delaycompensation (6 data bit times – 2clk_dlc) and the internal Bit is overrun. This bit is cleared by writing a 1'b0 to it. This bit is cleared automatically when the related RS-CANFD LITE channel is in CH_RESET mode.
14	--	RFU：未实现，读为 0
13:12	PNSTS	Pretended Network Filter State 2'b00: Acceptance Filter Mode 2'b01: Pretended Network Filter ID only and Acceptance Filter Mode 2'b10: Pretended Network Filter and Acceptance Filter Mode 2'b11: Pretended Network Filter Mode CH_RESET 模式下此寄存器自动清零
11:10	--	RFU：未实现，读为 0
9	SOCO	Successful occurrence counter overflow 1'b0: Successful occurrence counter has not overflowed 1'b1: Successful occurrence counter has overflowed 当 SOC=8'hFF 时，又完成一次成功的帧接收或发送，SOCO 自动置位；软件写 0 清零 CH_RESET 模式下此寄存器自动清零
8	EOCO	Error occurrence counter overflow 1'b0: Error occurrence counter has not overflowed 1'b1: Error occurrence counter has overflowed 当 EOC=8'hFF 时，又检测到 bus error，EOCO 自动置位；软件写 0 清零 CH_RESET 模式下此寄存器自动清零
7:0	TDCR	Transceiver Delay Compensation Result

位号	位名	说明
		TDC 测量的延迟，用 DLL 时钟的整数倍表示 当 TDC 功能使能后，此寄存器在 FDF 和 res 位之间的下降沿更新 当通道进入 CH_RESET 后此寄存器自动清零

25.16.9 Channel 0 CAN-FD CRC Register

名称	CFDC0FDCRC							
Offset	0x0110							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				SCNT			
位权限	U-0				R-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-			CRCREG[20:16]				
位权限	U-0			R-00000				
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CRCREG[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CRCREG[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:28	--	RFU：未实现，读为 0
27:24	SCNT	Stuff bit count These bits shows the stuff bit count (mod 8) for the CAN FD frame It shows the number of inserted stuff bits (modulo 8, Gray-coded) for a CAN FD frame if the CFDC0CTR.CTME bit is enabled on CFDC0FDCRC.SCNT[3:1]. And the corresponding Parity bit to this counter value on CFDC0FDCRC.SCNT[0].
23:21	--	RFU：未实现，读为 0
20:0	CRCREG	The calculated CRC value can be read from these bits, only when CFDC0CTR.CTME is 1’b1 for the channel. If CFDC0CTR.CTME bit is 1’b0, then these bits are always read as 0. If CRC_17 (17 bit CRC) is used, then bits [20:17] are always read as 0. These bits are cleared automatically when the related RS-CANFD LITE channel is in CH_RESET mode.

25.16.10 Global IP Version Register (CFDGIPV)

名称	CFDGIPV							
Offset	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PSI[15:8]							
位权限	R-x							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PSI[7:0]							

位权限	R-x							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CPUBUS	-					IPT	
位权限	R-x	U-0					R-x	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IPV							
位权限	R-0100 0011							

位号	位名	说明
31:16	PSI	Parameter Status Information These bits shows the status of a parameter. Reserved.
15	CPUBUS	CPU bus Information These bits shows the status of a parameter. Reserved.
14:10	--	RFU：未实现，读为 0
9:8	IPT	IP Type Release Number These bits show the IP type used in the Product 01: RS-CANFD
7:0	IPV	IP Version Release Number These bits show the IP version used in the Product 8'h43: Lite, single channel, ISO11898-1(2015)

25.16.11 Global Configuration Register (CFDGCFG)

名称	**CFDGCFG**							
Offset	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ITRCP[15:8]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ITRCP[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			TSSS	TSP			
位权限	U-0			R/W-0	R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		CMPOC	DCS	MME	DRE	DCE	TPRI
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:16	ITRCP	Interval Timer Reference Clock Prescaler FIFO Interval timer prescaler value When the CFDGCFG.ITRCP[15:0] bits are 16'h0000, then the timer is disabled. 用户应只在 GL_RESET 模式下改写此寄存器
15:13	--	RFU：未实现，读为 0
12	TSSS	Timestamp Source Select 1'b0: Source clock for Timestamp counter is peripheral clock 1'b1: Source clock for Timestamp counter is bit time clock

		用户应只在 GL_RESET 模式下改写此寄存器
11:8	TSP	Timestamp Prescaler 4'b0000: Timestamp Prescaler = 1 4'b0001: Timestamp Prescaler = 2 4'b0010: Timestamp Prescaler = 4 4'b0011: Timestamp Prescaler = 8 … 4'b1101: Timestamp Prescaler = 8192 4'b1110: Timestamp Prescaler = 16384 4'b1111: Timestamp Prescaler = 32768 The value configured in these bits defines the period of the clock source used for the Timestamp counter 用户应只在 GL_RESET 模式下改写此寄存器
7:6	--	RFU：未实现，读为 0
5	CMPOC	CAN-FD message Payload overflow configuration 1'b0: Message is rejected 1'b1: Message payload is cut to fit to configured message size The received message payload is always compared with the available message payload size in the MessageBuffer. This bit controls the message payload acceptance mechanism in the case when the received payloadis higher than the Message Buffer payload size CFDRMNB.RMPLS, CFDRFCCA.RFPLS,CFDCFCCd.CFPLS. When this bit is set and payload overflow occurs, DLC value is stored in a RXMB or FIFO without changing. 用户应只在 GL_RESET 模式下改写此寄存器
4	DCS	Data Link Controller Clock Select 1'b0: Internal clean clock 1'b1: External Clock source connected to clk_xincan pin 请勿设置为 1
3	MME	Mirror Mode Enable 1'b0: Mirror Mode disabled 1'b1: Mirror Mode enabled 用户应只在 GL_RESET 模式下改写此寄存器
2	DRE	DLC Replacement Enable 1'b0: DLC replacement disabled 1'b1: DLC replacement enabled If this bit is 1'b1 and CFDGCFG.DCE is 1'b1, then RS-CANFD LITE will store the configured value(CFDGAFLP0r.GAFLDLC) of DLC in the destination RX Message Buffer or FIFO buffer if the DLC check passes. Otherwise the DLC value in the destination RX Message Buffer or FIFO buffer is unchanged. 用户应只在 GL_RESET 模式下改写此寄存器
1	DCE	DLC Check Enable 1'b0: DLC check disabled 1'b1: DLC check enabled 用户应只在 GL_RESET 模式下改写此寄存器
0	TPRI	Transmission Priority 1'b0: ID Priority

		1’b1: Message Buffer Number Priority 用户应只在 GL_RESET 模式下改写此寄存器

25.16.12 Global Control Register (CFDGCTR)

名称	CFDGCTR							
Offset	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							TSRST
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				CMPOFIE	THLEIE	MEIE	DEIE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					GSLPR	GMDC	
位权限	U-0					R/W-1	R/W-01	

位号	位名	说明
31:17	--	RFU：未实现，读为 0
16	TSRST	Timestamp Reset 1’b0: Timestamp not reset 1’b1: Timestamp reset 置位此寄存器使 global timestamp register 复位成 16’h0000 置位后硬件自动清零，读回总是 0
15:12	--	RFU：未实现，读为 0
11	CMPOFIE	CAN-FD message payload overflow Flag Interrupt enable 1’b0: CAN-FD message payload overflow Flag Interrupt Disabled 1’b1: CAN-FD message payload overflow Flag Interrupt Enabled
10	THLEIE	TX History List Entry Lost Interrupt Enable 1’b0: TX History List Entry Lost Interrupt Disabled 1’b1: TX History List Entry Lost Interrupt Enabled
9	MEIE	Message lost Error Interrupt Enable 1’b0: Message Lost Error Interrupt Disabled 1’b1: Message Lost Error Interrupt Enabled
8	DEIE	DLC check Interrupt Enable 1’b0: DLC check Interrupt Disabled 1’b1: DLC check Interrupt Enabled
7:3	--	RFU：未实现，读为 0
2	GSLPR	Global Sleep Request 1’b0: Global Sleep Request Disabled 1’b1: Global Sleep Request Enabled 此寄存器只能在 GL_RESET 模式下置位，置位后 CAN 模块进入 GL_SLEEP 模式
1:0	GMDC	Global Mode Control 2’b00: Global Operation Mode Request 2’b01: Global Reset Mode Request 2’b10: Global Halt Mode Request 2’b11: Keep Current Value

25.16.13 Global Status Register (CFDGSTS)

名称	CFDGSTS							
Offset	0x001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				GRAMINIT	GSLPSTS	GHLTSTS	GRSTSTS
位权限	U-0				R-1	R-1	R-0	R-1

位号	位名	说明
31:4	--	RFU：未实现，读为 0
3	GRAMINIT	Global RAM Initialisation 1'b0: RAM initialisation is finished 1'b1: RAM initialisation ongoing
2	GSLPSTS	Global Sleep Status 1'b0: Not in Sleep Mode 1'b1: In Sleep Mode
1	GHLTSTS	Global Halt Status 1'b0: Not in Halt Mode 1'b1: In Halt Mode
0	GRSTSTS	Global Reset Status 1'b0: Not in Reset Mode 1'b1: In Reset Mode

25.16.14 Global Error Flag Register (CFDGERFL)

名称	CFDGERFL							
Offset	0x0020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							EEF
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				CMPOF	THLES	MES	DEF

位权限	U-0	R/W-0	R-0	R-0	R/W-0

位号	位名	说明
31:17	--	RFU：未实现，读为 0
16	EEF	ECC Error Flag 1’b0: ECC Error not detected during TX-SCAN 1’b1: ECC Error detected during TX-SCAN 硬件置位，软件写 0 清零，GL_RESET 模式下自动清零
15:4	--	RFU：未实现，读为 0
3	CMPOF	CAN-FD message payload overflow Flag 1’b0: CAN-FD message payload overflow not detected 1’b1: CAN-FD message payload overflow detected 硬件置位，软件写 0 清零，GL_RESET 模式下自动清零
2	THLES	TX History List Entry Lost Error Status 1’b0: TX History List Entry Lost Error not detected 1’b1: TX History List Entry Lost Error detected 检测到 TX history List Entry lost error 时置位，所有 TX history List Entry lost flags 被清零后自动清零 GL_RESET 模式下自动清零
1	MES	Message Lost Error Status 1’b0: Message lost Error not detected 1’b1: Message lost Error detected 检测到 FIFO message lost error 时置位，所有 FIFO message lost flags 被清零后自动清零 GL_RESET 模式下自动清零
0	DEF	DLC Error Flag 1’b0: DLC Error not detected 1’b1: DLC Error detected 硬件置位，软件写 0 清零，GL_RESET 模式下自动清零

25.16.15 Global TX Interrupt Status Register(CFDGTINSTS)

名称	CFDGTINTSTS							
Offset	0x00A4							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			THIF	CFTIF	TQIF	TAIF	TSIF
位权限	U-0			R-0	R-0	R-0	R-0	R-0

位号	位名	说明
31:5	--	RFU：未实现，读为 0
4	THIF	TX History List Interrupt 1’b0: TX History List Interrupt flag not set

		1’b1: TX History List Interrupt flag set This bit is set automatically when the related TX History List Interrupt Flag (CFDTHLSTS.THLIF) is set when the Interrupt is enabled. This bit is cleared automatically when related TX History List Interrupt Flag (CFDTHLSTS.THLIF) is cleared or the Interrupt enable is disabled. This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.
3	CFTIF	COM FIFO TX Mode Interrupt Flag 1’b0: COM FIFO TX mode Interrupt flag not set 1’b1: COM FIFO TX mode Interrupt flag set This bit is set automatically when the related Common TX FIFO Interrupt Flag (CFDCFSTS.CFTXIF) is set when the Interrupt is enabled. This bit is cleared automatically when related Common TX FIFO Interrupt Flag is cleared or the Interrupt enable is disabled. This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.
2	TQIF	TX Queue Interrupt Flag 1’b0: TX Queue Interrupt flag not set 1’b1: TX Queue Interrupt flag set This bit is set automatically when the related TX Queue Interrupt flag of the related channel is set when the Interrupt is enabled. This bit is cleared automatically when related TX Queue Interrupt flag is cleared or the Interrupt enable is disabled. This bit will be cleared automatically in GL_RESET mode or CH_RESET mode
1	TAIF	TX Abort Interrupt Flag 1’b0: TX abort Interrupt flag not set 1’b1: TX abort Interrupt flag set This bit is set automatically when abort Successful flag of the related channel is set when the Interrupt is enabled. This bit is cleared automatically when the related TX MB Result status bits are cleared or the Interrupt enable is disabled. This bit will be cleared automatically in GL_RESET mode or CH_RESET mode
0	TSIF	TX Successful Interrupt Flag 1’b0: TX Successful completion Interrupt flag not set 1’b1: TX Successful completion Interrupt flag set This bit is set automatically when the Transmission Successful flag of the related channel is set when the Interrupt is enabled. This bit is cleared automatically when related TX MB Result status bits are cleared or the Interrupt enable is disabled. This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

25.16.16 Global Timestamp Counter Register (CFDGTSC)

名称	CFDGTSC
Offset	0x0024

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TS[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TS[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:16	--	RFU：未实现，读为 0
15:0	TS	Timestamp value The Timestamp value is stored in this register based on the configuration of TSSS, TSBTCS and TSP. These bits will be cleared automatically in GL_RESET mode.

25.16.17 Global Acceptance Filter List Entry Control Register (CFDGAFLECTR)

名称	CFDGAFLECTR							
Offset	0x0028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							AFLDAE
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							AFLPN
位权限	U-0							R/W-0

位号	位名	说明
31:9	--	RFU：未实现，读为 0
8	AFLDAE	Acceptance Filter List Data Access Enable 1’b0: Acceptance Filter List Data access disabled 1’b1: Acceptance Filter List Data access enabled This bit prevents Acceptance Filter List write access if cleared after configuration of the Acceptance Filter List. Users can read data from Acceptance Filter List independent of the status of this bit. Users cannot write to this bit when RS-CANFD-LITE module is in GL_SLEEP mode.

		This bit should be set to enable write access to Acceptance Filter List
7:1	--	RFU：未实现，读为 0
0	AFLPN	Acceptance Filter List Page Number These bits select the Page Number to access the desired RAM area of the Acceptance Filter List. One Acceptance Filter List page consists of 16 Acceptance Filter List entries. Read/Write accesses to the Acceptance Filter List can only be performed via a fixed window. Users cannot write to this bit when RS-CANFD-LITE module is in GL_SLEEP mode.

25.16.18 Global Acceptance Filter List Configuration Register (CFDGAFLCFG)

名称	CFDGAFLCFG							
Offset	0x002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		RNC					
位权限	U-0		R/W-00 0000					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							
位权限	U-0							

位号	位名	说明
31:22	--	RFU：未实现，读为 0
21:16	RNC	Rule Number These bits define the number of rules in the Acceptance Filter List. Users can only write to this bit when RS-CANFD LITE module is in GL_RESET mode.
15:0	--	RFU：未实现，读为 0

25.16.19 Global Acceptance Filter List ID Registers (CFDGAFLIDr)

名称	CFDGAFLIDr, r=[1…10]h							
Offset	0x0120h + (r-1)*0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:24]				
位权限	R/W-0	R/W-0	R/W-0	R/W-0 0000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	GAFLID[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	GAFLID[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	GAFLID[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31	GAFLIDE	Global Acceptance Filter List Entry IDE Field 1'b0: Standard Identifier of Rule entry ID is valid for acceptance filtering 1'b1: Extended Identifier of Rule entry ID is valid for acceptance filtering Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.
30	GAFLRTR	Global Acceptance Filter List Entry RTR Field 1'b0: Data Frame 1'b1: Remote Frame Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.
29	GAFLLB	Global Acceptance Filter List Entry Loopback Configuration 1'b0: Global Acceptance Filter List entry ID for acceptance filtering has attribute 'RX' 1'b1: Global Acceptance Filter List entry ID for acceptance filtering has attribute 'TX' Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.
28:0	GAFLID	Global Acceptance Filter List Entry ID Field These bits represent the CAN Identifier (ID) field of each of the Global Acceptance Filter List entry. Acceptance filter process compares this field against the ID of a received CAN message. Users cannot write to these bits when CFDGAFLECTR.AFLDAE bit is 1'b0. Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.

25.16.20 Global Acceptance Filter List Mask Registers (CFDGAFLMr)

名称	CFDGAFLMr, r=[1…10]h							
Offset	0x0124h + (r-1)*0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	GAFLIDEM	GAFLRTRM	GAFLIFL1	GAFLIDM[28:24]				
位权限	R/W-0	R/W-0	R/W-0	R/W-0 0000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	GAFLIDM[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	GAFLIDM[15:8]							

位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	GAFLIDM[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31	GAFLIDEM	Global Acceptance Filter List IDE Mask 1'b0: IDE bit is not considered for ID matching 1'b1: IDE bit is considered for ID matching When IDE mask bit is 1'b0, then the ID comparison depends upon the received IDE bit. If received IDE bit is 1'b0, then STD-ID comparison takes place. If received IDE bit is 1'b1, then EXT-ID comparison takes place. Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.
30	GAFLRTRM	Global Acceptance Filter List Entry RTR Mask 1'b0: RTR bit is not considered for ID matching R/W 1'b1: RTR bit is considered for ID matching Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.
29	GAFLIFL1	Global Acceptance Filter List Information Label 1 Global Acceptance Filter List Information Label bit1 These bits allow the configuration of a 2-bit Information label that will be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is a MSB bit of an information label. Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.
28:0	GAFLIDM	Global Acceptance Filter List ID Mask Field Global Acceptance Filter List Mask field bits for ID field bits 0: Corresponding STD-ID / EXT-ID bit is not considered for ID matching 1: Corresponding STD-ID / EXT-ID bit is considered for ID matching Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.

25.16.21 Global Acceptance Filter List Pointer0 Registers (CFDGAFLP0r)

名称	CFDGAFLP0r, r=[1…10]h							
Offset	0x0128h + (r-1)*0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	GAFLPTR[15:8]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	GAFLPTR[7:0]							
位权限	R/W-0000 0000							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	GAFLRMV	-		GAFLRMDP				
位权限	R/W-0	U-0		R/W-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	GAFLIFL0	-			GAFLDLC			
位权限	R/W-0	U-0			R/W-0000			

位号	位名	说明
31:16	GAFLPTR	Global Acceptance Filter List Pointer Field Global Acceptance Filter List Pointer These bits allow the configuration of a 16-bit pointer that will be attached to a received message accepted by the related Global Acceptance Filter List entry. The Pointer will be added during message storage in the Message Buffer area and can be used by the application as support function. The pointer information could be used for example to support PDU Identifier allocation for the received message in AUTOSAR systems. Users cannot write to these bits when CFDGAFLECTR.AFLDAE bit is 1'b0. Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.
15	GAFLRMV	Global Acceptance Filter List RX Message Buffer Valid 1'b0: Global Acceptance Filter List Single Message Buffer Direction Pointer is invalid 1'b1: Global Acceptance Filter List Single Message Buffer Direction Pointer is valid This bit allows the enabling/disabling of a single reception Message Buffer as the target for a received message that is passing the acceptance check of the related Global Acceptance Filter List entry. Users cannot write to these bits when CFDGAFLECTR.AFLDAE bit is 1'b0. Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.
14:13	--	RFU：未实现，读为 0
12:8	GAFLRMDP	Global Acceptance Filter List RX Message Buffer Direction Pointer RX Message Buffer number for storage of received messages These bits allow the configuration of a single reception Message Buffer as the destination target for a received message that is passing the acceptance check of the related Global Acceptance Filter List entry. The value entered is the single destination Message Buffer number
7	GAFLIFL0	Global Acceptance Filter List Information Label 0 Global Acceptance Filter List Information Label bit0 These bits allow the configuration of a 2-bit Information label that will be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is a LSB bit of an information label
6:4	--	RFU：未实现，读为 0

3:0	GAFLDLC	Global Acceptance Filter List DLC Field Minimum no. of Data Bytes in a Data Frame required for its acceptance These bits allow the configuration of the minimum DLC (Data Length Code) value for a message to be accepted by the related Global Acceptance Filter List entry (automatic DLC filter function). DLC filter process is only passed if the DLC value of the message accepted by a Global Acceptance Filter List entry is equal or higher than the DLC value configured for this related Global Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding Rule Entry when this field is set to 4'h0. 0000: DLC of received message = 0 or more 0001: DLC of received message = 1 or more 0010: DLC of received message = 2 or more 0011: DLC of received message = 3 or more 0100: DLC of received message = 4 or more 0101: DLC of received message = 5 or more 0110: DLC of received message = 6 or more 0111: DLC of received message = 7 or more 1000: DLC of received message = 8 or more 1001: DLC of received message = 12 or more 1010: DLC of received message = 16 or more 1011: DLC of received message = 20 or more 1100: DLC of received message = 24 or more 1101: DLC of received message = 32 or more 1110: DLC of received message = 48 or more 1111: DLC of received message = 64 Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.

25.16.22 Global Acceptance Filter List Pointer1 Registers (CFDGAFLP1r)

名称	CFDGAFLP1r, r=[1…10]h							
Offset	0x012Ch + (r-1)*0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名		-						GAFLFDP[8]
位权限		U-0						R/W0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						GAFLFDP[1:0]	
位权限	U-0						R/W-00	

位号	位名	说明

31:9	-	RFU：未实现，读为 0
8	GAFLFDP[8]	Global Accetance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage These bits allow the configuration of FIFO Buffers as the target for a received message passing theacceptance check of the related Global Acceptance Filter List entry. Each bit of theCFDGAFLP1r.GAFLFDP[8] is configuring a dedicated FIFO: bit8:0：Disable Common FIFO as target for reception 1：Enable Common FIFO as target for reception Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET orCH_HALT mode. Users should only configure up to 2 destination FIFO Buffers or 1 destination FIFO Buffers plus one RXMessage Buffer.
7:2	--	RFU：未实现，读为 0
1:0	GAFLFDP[1:0]	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage These bits allow the configuration of FIFO Buffers as the target for a received message passing theacceptance check of the related Global Acceptance Filter List entry. Each bit of theCFDGAFLP1r.GAFLFDP[8,1:0] is configuring a dedicated FIFO: bit0:0：Disable RX FIFO 0 as target for reception 1：Enable RX FIFO 0 as target for reception bit1:0：Disable RX FIFO 1 as target for reception 1：Enable RX FIFO 1 as target for reception Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET orCH_HALT mode. Users should only configure up to 2 destination FIFO Buffers or 1 destination FIFO Buffers plus one RXMessage Buffer.

25.16.23 RX Message Buffer Number Registers (CFDRMNB)

名称	CFDRMNB							
Offset	0x30							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					RMPLS		
位权限	U-0					R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		NRXMB					
位权限	U-0		R/W-000000					

位号	位名	说明
31:11	--	RFU：未实现，读为 0
10:8	RMPLS	Reception Message BufferPayload Data Size 3'b000: 8 Bytes 3'b001: 12 Bytes 3'b010: 16 Bytes 3'b011: 20 Bytes 3'b100: 24 Bytes 3'b101: 32 Bytes 3'b110: 48 Bytes 3'b111: 64 Bytes
7:6	--	RFU：未实现，读为 0
5:0	NRXMB	Number of RX message buffers, 0 – 32

25.16.24 RX Message Buffer New Data Registers (CFDRMND)

名称	CFDRMND							
Offset	0x34							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RMNSu[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RMNSu[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RMNSu[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RMNSu[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	RMNSu	RX message buffer new data status 0: new data not stored in corresponding RX message buffer 1: new data stored in corresponding RX message buffer 每个 bit 对应一个接收消息 buffer 当开始向 buffer 存入消息时，此寄存器自动置位，软件写 0 清零 对于 8 字载荷，消息保存时间是 6 个 PCLK 周期，对于最长 64 字节载荷，消息保存时间是 20 个 PCLK 周期 This bit is cleared automatically when RS-CANFD LITE module enters GL_RESET mode

25.16.25 RX FIFO Configuration/Control Registers a (CFDRFCCa)

名称	CFDRFCCa (a=0,1)							
Offset	0x3Ch+a*0x4h							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFIGCV			RFIM	-	RFDC		
位权限	R/W-000			R/W-0	U-0	R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	RFPLS			-		RFIE	RFE
位权限	U-0	R/W-000			U-0		R/W-0	R/W-0

位号	位名	说明
31:16	--	RFU：未实现，读为 0
15:13	RFIGCV	RX FIFO interrupt generation counter value Interrupt is generated when: 000: FIFO is 1/8 full 001: FIFO is 1/4 full 010: FIFO is 3/8 full 011: FIFO is 1/2 full 100: FIFO is 5/8 full 101: FIFO is 3/4 full 110: FIFO is 7/8 full 111: FIFO is full 此寄存器只能在 GL_RESET 模式下设置
12	RFIM	RX FIFO interrupt mode 0: RX FIFO 水位达到 RFIGCV 时产生中断 1: 每次完成接收帧保存后产生中断 此寄存器只能在 GL_RESET 模式下设置
11	--	RFU：未实现，读为 0
10:8	RFDC	RX FIFO Depth Configuration 配置 RX FIFO 深度 000: 0 message 001: 4 messages 010: 8 messages 011: 16 messages 100: 32 messages 101: 48 messages 110: RFU 111: RFU 此寄存器只能在 GL_RESET 模式下设置
7	--	RFU：未实现，读为 0
6:4	RFPLS	RX FIFO payload data size configuration 000: 8 bytes 001: 12 bytes 010: 16 bytes 011: 20 bytes 100: 24 bytes 101: 32 bytes 110: 48 bytes 111: 64 bytes 此寄存器只能在 GL_RESET 模式下设置
3:2	--	RFU：未实现，读为 0

1	RFIE	RX FIFO interrupt enable 0: disabled 1: enabled GL_SLEEP 模式下不能置位
0	RFE	RX FIFO enable 0: disabled 1: enabled 用户只能在 GL_HALT 或者 GL_OPERATION 模式下写这个寄存器，并且只能在 FIFO 深度大于 0 小于 6 的情况下置位 RFE 应在 CFDRFCCa 中其他位已经配置后，再置位 当 CANFD 进入 GL_RESET 模式后此寄存器自动清零

25.16.26 RX FIFO Status Registers a (CFDRFSTSa)

名称	CFDRFSTSa (a=0,1)							
Offset	0x44h + a*0x04h							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		RFMC					
位权限	U-0		R-000000					
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				RFIF	RFMLT	RFFLL	RFEMP
位权限	U-0				R/W-0	R/W-0	R-0	R-1

位号	位名	说明
31:14	--	RFU：未实现，读为 0
13:8	RFMC	RX FIFO message count 只读，用于查询保存在 FIFO 中的消息个数
7:4	--	RFU：未实现，读为 0
3	RFIF	RX FIFO interrupt flag 硬件置位，软件写 0 清零
2	RFMLT	RF FIFO message lost 当 FIFO 已满的情况下，新的接收消息将丢失，此时 RFMLT 自动置位，软件写 0 清零
1	RFFLL	RX FIFO Full 0: not full 1: full
0	RFEMP	RX FIFO empty 0: not empty 1: empty

25.16.27 RX FIFO Pointer Control Registers (CFDRFPCTRa)

名称	CFDRFPCTRa (a=0,1)

Offset	0x4Ch + a*0x04h							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFPC							
位权限	W-0000 0000							

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7:0	RFPC	RX FIFO Pointer Control 当对此寄存器写 FF，FIFO 指针将指向下一条 FIFO entry 读此寄存器永远返回 00 用户只能在 GL_HALT 或 GL_OPERATION 模式下操作此寄存器

25.16.28 CommonFIFO Configuration/Control Register (CFDCFCC)

名称	CFDCFCC							
Offset	0x54							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CFITT							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CFDC			-			CFTML	
位权限	R/W-000			U-0			R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CFIGCV			CFIM	CFITR	CFITSS	-	CFM
位权限	R/W-000			R/W-0	R/W-0	R/W-0	U-0	R/W_0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	CFPLS			-	CFTXIE	CFRXIE	CFE
位权限	U-0	R/W-000			U-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:24	CFITT	Common FIFO interval transmission time CFIFO 配置为发送模式时，数据发送的延迟，单位是参考时钟 x1，或者参考时钟 x10，或者 bit time clock
23:21	CFDC	Common FIFO Depth Configuration 000: 0 message 001: 4 messages 010: 8 messages 011: 16 messages 100: 32 messages

		101: 48 messages 110: RFU 111: RFU
20:18	--	RFU：未实现，读为 0
17:16	CFTML	Common FIFO TX message buffer link 发送扫描链接位置
15:13	CFIGCV	Common FIFO interrupt generation counter value Interrupt is generated when: 000: FIFO is 1/8 full 001: FIFO is 1/4 full 010: FIFO is 3/8 full 011: FIFO is 1/2 full 100: FIFO is 5/8 full 101: FIFO is 3/4 full 110: FIFO is 7/8 full 111: FIFO is full 此寄存器只能在 GL_RESET 模式下设置
12	CFIM	RX FIFO interrupt mode 0: RX FIFO 水位达到 RFIGCV 时产生中断，TX FIFO 成功发送了最后一个消息 1: RX FIFO 每次完成接收帧保存后产生中断，TX FIFO 每次发送成功后产生中断 此寄存器只能在 GL_RESET 模式下设置
11	CFITR	Common FIFO interval timer resolution 0: reference clock period x1 1: reference clock period x10
10	CFITSS	Common FIFO interval timer source selection 0: reference clock 1: bit time clock
9	--	RFU：未实现，读为 0
8	CFM	Common FIFO mode 0: RX FIFO mode 1: TX FIFO mode
7	--	RFU：未实现，读为 0
6:4	CFPLS	RX FIFO payload data size configuration 000: 8 bytes 001: 12 bytes 010: 16 bytes 011: 20 bytes 100: 24 bytes 101: 32 bytes 110: 48 bytes 111: 64 bytes 此寄存器只能在 GL_RESET 模式下设置
3	--	RFU：未实现，读为 0
2	CFTXIE	Common FIFO TX interrupt enable 0: TX interrupt disabled 1: TX interrupt enabled
1	CFRXIE	Common FIFO RX interrupt enable 0: RX interrupt disabled 1: RX interrupt enabled
0	CFE	Common FIFO enable 0: disabled 1: enabled 用户只能在 GL_HALT 或者 GL_OPERATION 模式下写这个寄存器，

		并且只能在 FIFO 深度大于 0 小于 6 的情况下置位 CFE 应在 CFDCFCC 中其他位已经配置后，再置位 当 CANFD 进入 GL_RESET 模式后此寄存器自动清零

25.16.29 Common FIFO Status Register (CFDCFSTS)

名称	CFDCFSTS							
Offset	0x58							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		CFMC					
位权限	U-0		R-000000					
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
位权限	U-0			R/W-0	R/W-0	R/W-0	R-0	R-1

位号	位名	说明
31:14	--	RFU：未实现，读为 0
13:8	CFMC	Common FIFO message count 只读，用于查询保存在 FIFO 中的消息个数
7:5	--	RFU：未实现，读为 0
4	CFTXIF	Common FIFO TX interrupt flag 硬件置位，软件写 0 清零
3	CFRXIF	Common FIFO RX interrupt flag 硬件置位，软件写 0 清零
2	CFMLT	Common FIFO message lost RX mode 下，当 FIFO 已满的情况下，新的接收消息将丢失，此时 CFMLT 自动置位，软件写 0 清零
1	CFFLL	Common FIFO Full 0: not full 1: full
0	CFEMP	Common FIFO empty 0: not empty 1: empty

25.16.30 CommonFIFO Pointer Control Registers (CFDCFPCTR)

名称	CFDCFPCTR							
Offset	0x5C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CFPC							
位权限	R/W-0000 0000							

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7:0	CFPC	Common FIFO Pointer Control 当对此寄存器写 FF，FIFO 指针将指向下一条 FIFO entry 读此寄存器永远返回 00 用户只能在 GL_HALT 或 GL_OPERATION 模式下操作此寄存器 RX mode 下，用户应该仅在 Common FIFO 非空情况下写此寄存器 TX mode 下，用户应该仅在 Common FIFO 非满情况下写此寄存器

25.16.31 FIFO Empty Status Register (CFDFESTS)

名称	CFDFESTS							
Offset	0x60							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							CFEMP
位权限	U-0							R-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						RFxEMP	
位权限	U-0						R-11	

位号	位名	说明
31:9	--	RFU：未实现，读为 0
8	CFEMP	Common FIFO empty 0: not empty 1: empty
7:2	--	RFU：未实现，读为 0
1:0	RFxEMP	RX FIFO empty 0: not empty 1: empty Bit0 表示 RX FIFO 0 Bit1 表示 RX FIFO 1

25.16.32 FIFO Full Status Register (CFDFFSTS)

名称	CFDFFSTS							
Offset	0x64							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							CFFLL
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						RFxFLL	
位权限	U-0						R-00	

位号	位名	说明
31:9	--	RFU：未实现，读为 0
8	CFFLL	Common FIFO full 0: not full 1: full
7:2	--	RFU：未实现，读为 0
1:0	RFxFLL	RX FIFO full 0: not full 1: full Bit0 表示 RX FIFO 0 Bit1 表示 RX FIFO 1

25.16.33 FIFO Message Lost Status Register (CFDFMSTS)

名称	CFDFMSTS							
Offset	0x68							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							CFMLT
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						RFxMLT	
位权限	U-0						R-00	

位号	位名	说明
31:9	--	RFU：未实现，读为 0

8	CFMLT	Common FIFO message lost 0: not lost 1: lost
7:2	--	RFU：未实现，读为 0
1:0	RFxMLT	RX FIFO message lost 0: not lost 1: lost Bit0 表示 RX FIFO 0 Bit1 表示 RX FIFO 1

25.16.34 RX FIFO Interrupt Status Register (CFDRFISTS)

名称	CFDRFISTS							
Offset	0x6C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						RFxIF	
位权限	U-0						R-00	

位号	位名	说明
31:2	--	RFU：未实现，读为 0
1:0	RFxIF	RX FIFO interrupt status flag 0: interrupt not set 1: interrupt set Bit0 表示 RX FIFO 0 Bit1 表示 RX FIFO 1

25.16.35 DMA Transfer Control Register (CFDCDTCT) （不使用）

25.16.36 DMA Transfer Status Register (CFDCDTSTS) （不使用）

25.16.37 TX Message Buffer Control Registers I (CFDTMCi)

名称	CFDTMCi (i=0,1,2,3)							
Offset	0x70h + i*0x01h							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					TMOM	TMTAR	TMTR
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	位名	说明
7:3	--	RFU：未实现，读为 0
2	TMOM	TX Message Buffer One-Shot mode 0: not one-shot mode 1: one-shot mode TMOM 置位时，模块仅尝试一次发送，如果发送时出现总线错误或仲裁丢失，CANFD 不会自动重发 用户仅能在 GL_HALT 或 GL_OPERATION 模式下写此寄存器，GL_RESET 或 CH_RESET 模式下自动清零 如果消息已经被请求发送，则用户不应再置位 TMOM
1	TMTAR	TX message buffer transmission abort request 0: TX abort not requested 1: TX abort requested 用户仅能在 GL_HALT 或 GL_OPERATION 模式下写此寄存器，GL_RESET 或 CH_RESET 模式下自动清零 软件置位此寄存器来执行发送撤销，当发送成功、发送被撤销或发送总线错误或仲裁丢失时，自动清零
0	TMTR	TX message buffer transmission request 0: TX not requested 1: TX requested 用户仅能在 GL_HALT 或 GL_OPERATION 模式下写此寄存器，GL_RESET 或 CH_RESET 模式下自动清零 如果 TX buffer 链接到 common FIFO，或者是发送队列的一部分，用户不应置位此寄存器 软件置位后，此寄存器在发送成功、发送撤销、或 TMOM=1 时发生总线错误或仲裁丢失时，自动清零

25.16.38 TX Message Buffer Status Registers j (CFDTMSTSj)

名称	CFDTMSTSj (j=0,1,2,3)							
Offset	0x74h + i*0x01h							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			TMTARM	TMTRM	TMTRF		TMTSTS
位权限	U-0			R-0	R-0	R/W-00		R-0

位号	位名	说明
7:5	--	RFU：未实现，读为 0
4	TMTARM	TX message buffer transmission abort request mirrored 0: TX abort not requested 1: TX abort requested 只读的镜像位，镜像 CFDTMCi.TMTAR
3	TMTRM	TX message buffer transmission request mirrored 0: TX not requested 1: TX requested 只读的镜像位，镜像 CFDTMCi.TMTR

2:1	TMTRF	TX message buffer transmission result flag 00: No result 01: TX aborted from TXMB 10: TX successful from TXMB, and abort not requested 11: TX successful from TXMB, and abort requested
0	TMTSTS	TX message buffer transmission status 0: no transmission on-going 1: transmission on-going

25.16.39 TX Message Buffer Transmission Request Status Register (CFDTMTRSTS)

名称	CFDTMTRSTS							
Offset	0x78							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				CFDTMTRSTS			
位权限	U-0				R-0000			

位号	位名	说明
31:4	--	RFU：未实现，读为 0
3:0	CFDTMTRSTS	TX message buffer transmission request status 0: TX not requested 1: TX requested 表示 CFDTMCi.TMTR 的状态，只读，每个 bit 对应一个 TX buffer

25.16.40 TX Message Buffer Transmission Abort Request Status Register (CFDTMTARSTS)

名称	CFDTMTARSTS							
Offset	0x7C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							

位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				CFDTMTARSTS			
位权限	U-0				R-0000			

位号	位名	说明
31:4	--	RFU：未实现，读为 0
3:0	CFDTMTARSTS	TX message buffer transmission abort request status 0: TX abort not requested 1: TX abort requested 表示 CFDTMCi.TMTAR 的状态，只读，每个 bit 对应一个 TX buffer

25.16.41 TX Message Buffer Transmission Completion Status Register (CFDTMTCSTS)

名称	CFDTMTCSTS							
Offset	0x80							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				CFDTMTCSTS			
位权限	U-0				R-0000			

位号	位名	说明
31:4	--	RFU：未实现，读为 0
3:0	CFDTMTCSTS	TX message buffer transmission completion status 0: TX buffer transmission not completed 1: TX buffer transmission completed

25.16.42 TX Message Buffer Transmission Abort Status Register (CFDTMTASTS)

名称	CFDTMTASTS							
Offset	0x84							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				CFDTMTASTS			
位权限	U-0				R-0000			

位号	位名	说明
31:4	--	RFU：未实现，读为 0
3:0	CFDTMTASTS	TX message buffer transmission abort status 0: TX buffer transmission not aborted 1: TX buffer transmission aborted 表示 CFDTMSTSj.TMTRF 的状态，只读，当 TMTRF=01 时自动置位 每个 bit 对应一个 TX buffer

25.16.43 TX Message Buffer Interrupt Enable Register (CFDTMIEC)

名称	CFDTMIEC							
Offset	0x88							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TMIE			
位权限	U-0				R/W-0000			

位号	位名	说明
31:4	--	RFU：未实现，读为 0
3:0	TMIE	TX message buffer interrupt enable 0: interrupt disabled 1: interrupt enabled 当对应的 TX buffer 是发送队列的一部分、或链接到 Common FIFO 时，用户不应写此寄存器

25.16.44 TX Queue Configuration/Control Register (CFDTXQCC)

名称	CFDTXQCC							
Offset	0x8C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						TXQDC	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXQIM	-	TXQTXIE	-				TXQE
位权限	R/W-0	U-0	R/W-0	U-0				R/W-0

位号	位名	说明
31:10	--	RFU：未实现，读为 0
9:8	TXQDC	TX Queue Depth Configuration 00: 0 message 01: RFU 10: 3 messages 11: 4 messages
7	TXQIM	TX Queue Interrupt Mode 0: 队列中最后一个消息帧成功发送后产生中断 1: 每个消息帧成功发送后产生中断
6	--	RFU：未实现，读为 0
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX interrupt disabled 1: TX interrupt enabled
4:1	--	RFU：未实现，读为 0
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled

25.16.45 TX Queue Status Register (CFDTXQSTS)

名称	CFDTXQSTS							
Offset	0x90							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					TXQMC		
位权限	U-0					R-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					TXQTXIF	TXQFLL	TXQEMP
位权限	U-0					R/W-0	R-0	R-1

位号	位名	说明
31:11	--	RFU：未实现，读为 0
10:8	TXQMC	TX Queue message count, number of messages in the TX Queue
7:3	--	RFU：未实现，读为 0
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue interrupt not set 1: TX Queue interrupt set
1	TXQFLL	TX Queue Full 0: TX Queue not full 1: TX Queue full
0	TXQEMP	TX Queue empty 0: TX Queue not empty 1: TX Queue empty

25.16.46 TX Queue Pointer Control Registers (CFDTXQPCTR)

名称	CFDTXQPCTR							
Offset	0x94							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXQPC							
位权限	W-0000 0000							

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7:0	TXQPC	TX Queue Pointer Control 当对此寄存器写 FF，发送队列的写指针将被更新，并且对当前消息触发发送请求 读此寄存器永远返回 00 用户只能在发送队列使能，并且队列非满的情况下写 FF

25.16.47 TX History List Configuration/Control Register (CFDTHLCC)

名称	CFDTHLCC							
Offset	0x98							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					THLDTE	THLIM	THLIE
位权限	U-0					R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							THLE
位权限	U-0							R/W-0

位号	位名	说明
31:11	--	RFU：未实现，读为 0
10	THLDTE	TX history list dedicated TX enable 0: TX FIFO + TX Queue 1: Flat TX MB + TX FIFO + TX Queue
9	THLIM	TX history list interrupt mode 0: interrupt generated if TX history list level reaches 3/4 of history list depth 1: interrupt generated for every successfully stored entry
8	THLIE	TX history list interrupt enable 0: TX history list interrupt diabled 1: TX history list interrupt enabled
7:1	--	RFU：未实现，读为 0
0	THLE	TX history list enable 0: TX history list disabled 1: TX history list enabled

25.16.48 TX History List Status Register (CFDTHLSTS)

名称	CFDTHLSTS							
Offset	0x9C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				THLMC			
位权限	U-0				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				THLIF	THLELT	THLFLL	THLEMP
位权限	U-0				R/W-0	R/W-0	R-0	R-1

位号	位名	说明
31:12	--	RFU：未实现，读为 0
11:8	THLMC	TX History List Message Count
7:4	--	RFU：未实现，读为 0
3	THLIF	TX History List Interrupt flag 0: TX History List Interrupt not set 1: TX History List Interrupt set

2	THLELT	TX History List Entry Lost 0: no entry lost in TX history list 1: TX history list entry lost
1	THLFLL	TX History List Full 0: TX history list not full 1: TX history list full
0	THLEMP	TX History List Empty 0: TX history list not empty 1: TX history list empty

25.16.49 TX History List Access Register0 (CFDTHLACC0)

名称	CFDTHLACC0							
Offset	0x740							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TMTS[15:8]							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TMTS[7:0]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			BN		BT		
位权限	U-0			R-00		R-000		

位号	位名	说明
31:16	TMTS	Transmit Timestamp 只读，用于查询内部时间戳定时器
15:5	--	RFU：未实现，读为 0
4:3	BN	Number of TX message buffer 用于查询发送成功的消息来自哪个 MB
2:0	BT	Buffer type 001: flat TX message buffer 010: TX FIFO MB 100: TX Queue MB 用于指示待发送消息所处的存储位置

25.16.50 TX History List Access Register1 (CFDTHLACC1)

名称	CFDTHLACC1							
Offset	0x744							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						TIFL	
位权限	U-0						R-00	

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TID[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TID[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:18	--	RFU：未实现，读为 0
17:16	TIFL	Transmit information label 指示 MB information label (CFDTMFDCTR.TMIFL) 或 TX FIFO information label (CFDCFFDCSTS.CFIFL)
15:0	TID	Transmit ID 指示 MB 参考 ID (CFDTMFDCTR.TMPTR) 或 TX FIFO 参考 ID (CFDCFFDCSTS.CFPTR)

25.16.51 TX History List Pointer Control Register (CFDTHLPCTR)

名称	CFDTHLPCTR							
Offset	0xA0							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	THLPC[7:0]							
位权限	W-0000 0000							

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7:0	THLPC	TX history list pointer control 对此寄存器写 FF，将使读指针指向发送历史列表中的下一条 entry 用户只能在 CH_HALT 或 CH_OPERATION 模式下写 THLPC 用户应仅在发送历史列表使能，并且列表非空的情况下写 FF

25.16.52 Global Reset Control Register (CFDGRSTC)

名称	CFDGRSTC							
Offset	0xD8							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	KEY							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							SRST
位权限	U-0							R/W-0

位号	位名	说明
31:16	--	RFU：未实现，读为 0
15:8	KEY	写 SRST 之前必须向 KEY 写入 0xC4，以解锁 SRST 写保护
7:1	--	RFU：未实现，读为 0
0	SRST	Software Reset 软件置位 SRST 将使 RS-CANFD 模块全局复位，软件清零撤销复位

25.16.53 Global Test Configuration Register (CFDGTSTCFG)

名称	CFDGTSTCFG							
Offset	0xA8							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				RTMPS			
位权限	U-0				R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							
位权限	U-0							

位号	位名	说明
31:20	--	RFU：未实现，读为 0
19:16	RTMPS	RAM test mode page select 用于选择 CPU 读写访问的 RAM page，有效值 0~9 用户应仅在 GL_HALT 模式下改写此寄存器
15:0	--	RFU：未实现，读为 0

25.16.54 Global Test Control Register (CFDGTSTCTR)

名称	CFDGTSTCTR							
Offset	0xAC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					RTME	-	
位权限	U-0					R/W-0	U-0	

位号	位名	说明
31:3	--	RFU：未实现，读为 0
2	RTME	Ram test mode enable 0: disable ram test mode 1: enable ram test mode 用户只能在 GL_HALT 模式下写此寄存器
1:0	--	RFU：未实现，读为 0

25.16.55 Global FD Configuration Register (CFDGFDCFG)

名称	CFDGFDCFG							
Offset	0xB0							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						TSCCFG	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							RPED
位权限	U-0							R/W-0

位号	位名	说明
31:10	--	RFU：未实现，读为 0
9:8	TSCCFG	Timestamp capture config 00: timestamp capture at SOF 01: timestamp capture at frame valid indication 10: timestamp capture at RES bit for FD frames, and at SOF for classic frames 11: RFU
7:1	--	RFU：未实现，读为 0
0	RPED	RES bit protocol exception disable 0: protocol exception event dectect enabled 1: protocol exception event dectect disabled 用户只能在 GL_RESET 模式下设置此寄存器 置位时，模块在检测到 RES 隐性位时发送错误帧

25.16.56 Global Lock Key Register (CFDGLOCKK)

名称	CFDGLOCKK							
Offset	0xB8							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LOCK[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCK[7:0]							
位权限	W-0000 0000							

位号	位名	说明
31:16	--	RFU：未实现，读为 0
15:0	LOCK	Lock key 参见测试模式章节

25.16.57 RAM Test Page Access Register (CFDRPGACCk)

名称	CFDRPGACCk (k=0~63)							
Offset	0x280h + k * 0x04h							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RDTA[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RDTA[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RDTA[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RDTA[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	RDTA	Ram Data Test Access RAM 测试模式下可通过读写此寄存器访问 RAM

25.16.58 Global Pretended Network Filter List Entry Control Register (CFDGPFLECTR)

名称	CFDGPFLECTR							
Offset	0xD0							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							PFLDAE
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							
位权限	U-0							

位号	位名	说明
31:9	--	RFU：未实现，读为 0
8	PFLDAE	Pretended Network Filter List Data Access Enable 1'b0: Pretended Network Filter List Data access disabled 1'b1: Pretended Network Filter List Data access enabled
1:0	--	RFU：未实现，读为 0

25.16.59 Global Pretended Network Filter List Configuration Register (CFDGPFLCFG)

名称	CFDGPFLCFG							
Offset	0xD4							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						RNC	
位权限	U-0						R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							
位权限	U-0							

位号	位名	说明
31:26	--	RFU：未实现，读为 0
25:24	RNC	Rule number 定义 pretended network filter list 中的 rule 个数 用户只能在 GL_RESET 模式下改写
23:0	--	RFU：未实现，读为 0

25.16.60 Global Pretended Network Filter ID Registers (CFDGPFLIDs)

名称	CFDGPFLIDs (s=1,2)							
Offset	0x220h + (s-1) * 0x24h							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	GPFLIDE	GPFLRTR	GPFLLB	GPFLID[28:24]				
位权限	R/W-0	R/W-0	R/W-0	R/W-00000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	GPFLID[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	GPFLID[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	GPFLID[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31	GPFLIDE	Global Pretended NetworkFilter List IDE Field 1'b0: Standard Identifier of Rule entry ID is valid for acceptance filtering 1'b1: Extended Identifier of Rule entry ID is valid for acceptance filtering
30	GPFLRTR	Global Pretended NetworkFilter List Entry RTR Field 1'b0: Data Frame 1'b1: Remote Frame
29	GPFLLB	Global Pretended NetworkFilter List Entry LoopbackConfiguration 1'b0: Global Pretended Network Filter List entry ID for acceptance filtering has attribute 'RX' 1'b1: Global Pretended Network Filter List entry ID for acceptance filtering has attribute 'TX'
26:0	GPFLID	Global Pretended NetworkFilter List ID Field PNF 比较此 ID 与接收消息 ID，位对齐格式参见“Identifier bits alignment”章节 当 CFDGPFLECTR.PFLDAE=0 时，用户不能改写此寄存器 用户应该仅在 CH_RESET 或 CH_HALT 模式下写此寄存器

25.16.61 Global Pretended Network Filter List Mask Registers (CFDGPFLMs)

名称	CFDGPFLMs (s=1,2)							
Offset	0x224h + (s-1) * 0x24h							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	GPFLIDEM	GPFLRTRM	GPFLIFL1	GPFLIDM[28:24]				
位权限	R/W-0	R/W-0	R/W-0	R/W-00000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	GPFLIDM[23:16]							
位权限	R/W-0000 0000							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	GPFLIDM[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	GPFLIDM[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31	GPFLIDEM	Global Pretended NetworkFilter List IDE Mask 1'b0: IDE bit is not considered for ID matching 1'b1: IDE bit is considered for ID matching
30	GPFLRTRM	Global Pretended NetworkFilter List Entry RTR Mask 1'b0: RTR bit is not considered for ID matching 1'b1: RTR bit is considered for ID matching
29	GPFLIFL1	Global Pretended Network Filter List Information Label 1 Global Pretended Network Filter List Information Label bit1
26:0	GPFLIDM	Global Pretended NetworkFilter List ID Mask 0: Corresponding STD-ID/EXT-ID bit is not considered for ID matching 1: Corresponding STD-ID/EXT-ID bit is considered for ID matching

25.16.62 Global Pretended Network Filter List Pointer0 Registers (CFDGPFLP0s)

名称	CFDGPFLP0s (s=1,2)							
Offset	0x228h + (s-1) * 0x24h							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	GPFLPTR[15:8]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	GPFLPTR[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	GPFLRMV	-		GPFLRMDP				
位权限	R/W-0	U-0		R/W-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	GPFLIFL0	-		GPFLDLC				
位权限	R/W-0	U-0		R/W-0000				

位号	位名	说明
31:16	GPFLPTR	Global Pretended Network Filter List Pointer
15	GPFLRMV	Global Pretended Network Filter List RX Message Buffer Valid 1'b0: Global Pretended Network Filter List Single Message Buffer Direction Pointer is invalid 1'b1: Global Pretended Network Filter List Single Message Buffer Direction Pointer is valid
14:13	--	RFU：未实现，读为 0

12:8	GPFLRMDP	Global Pretended NetworkFilter List RX MessageBuffer Direction Pointer 用于指向接收到的消息所存入的 RX buffer
7	GPFLIFL0	Global Pretended Network Filter List Information Label bit0
6:4	--	RFU：未实现，读为 0
3:0	GPFLDLC	Global Pretended NetworkFilter List DLC Field Minimum no. of Data Bytes in a Data Frame required for itsacceptance

GPFLDLC定义PNF list可以接收的最小DLC长度：

Format	DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
CAN and CAN-FD	0	0	0	0	DLC of received message = 0 or more (DLC Filter check is disabled)
CAN and CAN-FD	0	0	0	1	DLC of received message = 1 or more
CAN and CAN-FD	0	0	1	0	DLC of received message = 2 or more
CAN and CAN-FD	0	0	1	1	DLC of received message = 3 or more
CAN and CAN-FD	0	1	0	0	DLC of received message = 4 or more
CAN and CAN-FD	0	1	0	1	DLC of received message = 5 or more
CAN and CAN-FD	0	1	1	0	DLC of received message = 6 or more
CAN and CAN-FD	0	1	1	1	DLC of received message = 7 or more
CAN	1	x	x	x	DLC of received message = 8 or more
CAN-FD	1	0	0	0	DLC of received message = 8 or more
CAN-FD	1	0	0	1	DLC of received message = 12 or more
CAN-FD	1	0	1	0	DLC of received message = 16 or more
CAN-FD	1	0	1	1	DLC of received message = 20 or more
CAN-FD	1	1	0	0	DLC of received message = 24 or more
CAN-FD	1	1	0	1	DLC of received message = 32 or more
CAN-FD	1	1	1	0	DLC of received message = 48 or more
CAN-FD	1	1	1	1	DLC of received message = 64

25.16.63 Global Pretended Network Filter List Pointer1 Registers (CFDGPFLP1s)

名称	CFDGPFLP1s (s=1,2)							
Offset	0x22Ch + (s-1) * 0x24h							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							GPFLFDP[8]
位权限	U-0							R/W-0

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						GPFLFDP[1:0]	
位权限	U-0						R/W-00	

位号	位名	说明
31:9	--	RFU：未实现，读为 0
8	GPFLFDP[8]	Global Pretended Network Filter List FIFO Direction Pointer (GPFLFDP[8])
7:2	--	RFU：未实现，读为 0
1:0	GPFLFDP[1:0]	Global Pretended Network Filter List FIFO Direction Pointer (GPFLFDP[1:0])

Global Pretended Network Filter List FIFO Direction Pointer (GPFLFDP[8,1:0])
These bits allow the configuration of FIFO Buffers as the target for a received message passing the acceptance check of the related Global Pretended Network Filter List entry. Each bit of the CFDGPFLP1s.GPFLFDP[8,1:0] is configuring a dedicated FIFO:

Bit	**Value (Binary)**	**Function**
b0	0	Disable RX FIFO 0 as target for reception
	1	Enable RX FIFO 0 as target for reception
b1	0	Disable RX FIFO 1 as target for reception
	1	Enable RX FIFO 1 as target for reception
b8	0	Disable Common FIFO as target for reception
	1	Enable Common FIFO as target for reception

25.16.64 Global Pretended Network Filter List Type Registers (CFDGPFLPTs)

名称	**CFDGPFLPTs (s=1,2)**							
Offset	0x230h + (s-1) * 0x24h							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	GPFLANDOR	GPFLRANG0	GPFLOUT0	-				
位权限	R/W-0	R/W-0	R/W-0	U-0				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				GPFLOFFSET0			
位权限	U-0				R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	GPFLRANG1	GPFLOUT1	-				
位权限	U-0	R/W-0	R/W-0	U-0				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				GPFLOFFSET1			
位权限	U-0				R/W-0000			

位号	位名	说明
31	GPFLANDOR	Global Pretended Network filter conditions of the filters 0 and 1 1’b0: Both of filters 0 and 1 are successful. 1’b1: One of the filter 0 or 1 is successful
30	GPFLRANG0	Global Pretended Network filter comparison conditions of the filter0 1’b0: payload data match filter

		1'b1: upper / lower filter
29	GPFLOUT0	Global Pretended Network filter conditions of upper / lower filter of the filter0 1'b0: Within the range of upper limit and lower limit 1'b1: Outside of the range of upper limit and lower limit
28:20	--	RFU：未实现，读为 0
19:16	GPFLOFFSET0	Global Pretended Network filter offset value of the filter0
15	--	RFU：未实现，读为 0
14	GPFLRANG1	Global Pretended Network filter comparison conditions of the filter1 1'b0: payload data match filter 1'b1: upper / lower filter
13	GPFLOUT1	Global Pretended Network filter conditions of upper / lower filter of the filter1 1'b0: Within the range of upper limit and lower limit 1'b1: Outside of the range of upper limit and lower limit
12:4	--	RFU：未实现，读为 0
3:0	GPFLOFFSET1	Global Pretended Network filter offset value of the filter1

25.16.65 Global Pretended Network Filter List Payload Data 0 Registers (CFDGPFLPD0s)

名称	CFDGPFLPD0s (s=1,2)							
Offset	0x234h + (s-1) * 0x24h							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	FDATA[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	FDATA[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	FDATA[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FDATA[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	FDATA	Pretended Network Filter List Filter data When CFDGPFLPTs.GPFLRANG0 is 1'b0, these bits set up the payload match data of the filter0. When CFDGPFLPTs.GPFLRANG0 is 1'b1, these bits set up the upper-limit filter value of the filter0. Users cannot write to these bits when CFDGPFLECTR.PFLDAE bit is 1'b0. Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.

25.16.66 Global Pretended Network Filter List Payload Data 1 Registers (CFDGPFLPD1s)

名称	CFDGPFLPD1s (s=1,2)							
Offset	0x23Ch + (s-1) * 0x24h							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	FDATA[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	FDATA[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	FDATA[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FDATA[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	FDATA	Pretended Network Filter List Filter data When CFDGPFLPTs.GPFLRANG1 is 1'b0, these bits set up the payload match data of the filter0. When CFDGPFLPTs.GPFLRANG1 is 1'b1, these bits set up the upper-limit filter value of the filter0. Users cannot write to these bits when CFDGPFLECTR.PFLDAE bit is 1'b0. Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.

25.16.67 Global Pretended Network Filter List Payload Mask 0 Registers (CFDGPFLPM0s)

名称	CFDGPFLPM0s (s=1,2)							
Offset	0x238h + (s-1) * 0x24h							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	FMASK[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	FMASK[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	FMASK[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FMASK[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	FMASK	Pretended Network Filter List Filter data mask

		When CFDGPFLPTs.GPFLRANG0 is 1'b0, these bits set up the payload data mask of the filter0. When CFDGPFLPTs.GPFLRANG0 is 1'b1, these bits set up the lower-limit filter value of the filter0. Users cannot write to these bits when CFDGPFLECTR.PFLDAE bit is 1'b0. Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.

25.16.68 Global Pretended Network Filter List Payload Mask 1 Registers (CFDGPFLPM1s)

名称	CFDGPFLPM1s (s=1,2)							
Offset	0x240h + (s-1) * 0x24h							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	FMASK[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	FMASK[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	FMASK[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FMASK[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	FMASK	Pretended Network Filter List Filter data mask When CFDGPFLPTs.GPFLRANG1 is 1'b0, these bits set up the payload data mask of the filter1. When CFDGPFLPTs.GPFLRANG1 is 1'b1, these bits set up the lower-limit filter value of the filter1. Users cannot write to these bits when CFDGPFLECTR.PFLDAE bit is 1'b0. Users should only write to these bits when the related RS-CANFD LITE channel is in CH_RESET or CH_HALT mode.

25.16.69 Global AFL Ignore Entry Register (CFDGAFLIGNENT)

名称	CFDGAFLIGNENT							
Offset	0xC0							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			IRN				
位权限	U-0			R/W-00000				

位号	位名	说明
31:5	--	RFU：未实现，读为 0
4:0	IRN	Ignore Rule Number Define rule number which ignores an AFL entry Users should enter, only the values between 0 and 31 inclusive. Users should only write to these bits when CFDGAFLIGNCTR.IREN bit is 1'b0. Users cannot write to this bit when the RS-CANFD LITE module is in GL_SLEEP mode

25.16.70 Global AFL Ignore Control Register (CFDGAFLIGNCTR)

名称	CFDGAFLIGNCTR							
Offset	0xC4							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	KEY							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							IREN
位权限	U-0							R/W-0

位号	位名	说明
31:16	--	RFU：未实现，读为 0
15:8	KEY	When C4h is written in these bits, the write of a CFDGAFLIGNCTR.IREN bit becomes available. Read value from these bits is always 8'h0. Users should write a CFDGAFLIGNCTR.IREN bit and the CFDGAFLIGNCTR.KEY bit simultaneously
7:1	--	RFU：未实现，读为 0
0	IREN	Ignore Rule Enable When this bit is set, the entry number (selected by CFDGAFLIGNENT register) is ignored. This bit is cleared automatically when RS-CANFD LITE module enters GL_RESET mode

25.16.71 RX Message Buffer Interrupt Enable Configuration Register g (CFDRMIEC)

名称	CFDRMIEC

Offset	0x38							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RMIE							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RMIE							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RMIE							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RMIE							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	RMIEg	RX Message Buffer Interrupt Enable If this bit is set, then an interrupt will be generated at the end of a successful reception from the corresponding Message Buffer. Users cannot write to this bit when the RS-CANFD LITE module is in GL_SLEEP mode.

25.17 ID 位对齐格式

标准ID（11bit）格式：ID28 – ID18对齐到bit10 – bit0

扩展ID（29bit）格式：ID28 – ID0 对齐到bit28 – bit0

对于标准ID，bit28 – bit11应为18'h0

Standard Identifier (11 bit) format

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE=0	RTR	-	0	0	0	0	0	0	0	0	0	0	0	0	0
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18

Extended Identifier (29 bit) format

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE=1	RTR	-	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

25.18 Message Buffer 结构

RS-CANFD-Lite包含一个通道，其中包括32个接收MB，2个接收FIFO，1个common FIFO，4个发送MB。这些消息buffer都是通过寄存器映射访问，寄存器的起始地址参见下表。

RMBCP – RX message buffer component

RFMBCP – RX FIFO message buffer component

CFMBCP – Common FIFO message buffer component

TMBCP – TX message buffer component

b = Message Buffer Component Index	message buffer component	Register	p	Regular Start Address
[0…no_of_RMBCPs-1] b = [0…7]	RMBCP**b[i]**	RMID	x	0920h + b*004Ch
		RMPTR	x	0924h + b*004Ch
		RMFDSTS	x	0928h + b*004Ch
		RMDF**p**	[0…15]	092Ch + p*0004h + b*004Ch
[0…no_of_RMBCPs-1] b = [8…15]	RMBCPb[i]	RMID	x	0D20h + (b-8)*004Ch
		RMPTR	x	0D24h + (b-8)*004Ch
		RMFDSTS	x	0D28h + (b-8)*004Ch
		RMDF**p**	[0…15]	0D2Ch + p*0004h + (b-8)*004Ch
[0…no_of_RMBCPs-1] b = [16…23] (Only RX max version)	RMBCPb[i]	RMID	x	1120h + (b-16)*004Ch
		RMPTR	x	1124h + (b-16)*004Ch
		RMFDSTS	x	1128h + (b-16)*004Ch
		RMDF**p**	[0…15]	112Ch + p*0004h + (b-16)*004Ch
[0…no_of_RMBCPs-1] b = [24…31] (Only RX max version)	RMBCPb[i]	RMID	x	1520h + (b-24)*004Ch
		RMPTR	x	1524h + (b-24)*004Ch
		RMFDSTS	x	1528h + (b-24)*004Ch
		RMDF**p**	[0…15]	152Ch + p*0004h + (b-24)b*004Ch
[0…no_of_RFMBCPs-1]	RFMBCPb[i]	RFID	x	0520h + b*004Ch
		RFPTR	x	0524h + b*004Ch
		RFFDSTS	x	0528h + b*004Ch
		RFDF**p**	[0…15]	052Ch + p*0004h + b*004Ch
[0…no_of_CFMBCPs-1]	CFMBCPb[i]	CFID	x	05B8h
		CFPTR	x	05BCh
		CFFDCSTS	x	05C0h
		CFDF**p**	[0…15]	05C4h + p*0004h
[0…no_of_TMBCPs-1]	TMBCPb[i]	TMID	x	0604h + b*004Ch
		TMPTR	x	0608h + b*004Ch
		TMFDCTR	x	060Ch + b*004Ch
		TMDF**p**	[0…15]	0610h + p*0004h + b*004Ch

Message Buffer有四种类型：RX Message Buffer, RX FIFO Access Message Buffer, Common FIFO Access Message Buffer, TX Message Buffer

每个MB有以下寄存器对应：ID，PTR（pointer），DFp（data field）

下文详细说明MB相关寄存器组。

25.18.1 RX Message Buffer Component b (CFDRMBCPb)

名称	CFDRMBCPb (b=0~31)							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	Rc							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	Rc							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	Rc							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	Rc							
位权限	R-0000 0000							

位号	位名	说明
31:0	Rc	RX Message BufferComponent c The RX Message Buffer Component is made up of the following registers: CFDRMID, CFDRMPTR,CFDRMFDSTS, and CFDRMDFp.

每个RX message buffer component包含19个寄存器：c=0~18，分别是CFDRMID，CFDRMPTR，CFDRMFTSTS，CFDRMDFp

RX Message Buffer Component (RMBCP)	
Rc	
R0	RX Message Buffer (b) ID Registers
R1	RX Message Buffer (b) Pointer Registers
R2	RX Message Buffer (b) CAN-FD Status Registers
R3	RX Message Buffer (b) Data Field 0 Registers
R4	RX Message Buffer (b) Data Field 1 Registers
R5	RX Message Buffer (b) Data Field 2 Registers
R6	RX Message Buffer (b) Data Field 3 Registers
R7	RX Message Buffer (b) Data Field 4 Registers
R8	RX Message Buffer (b) Data Field 5 Registers
R9	RX Message Buffer (b) Data Field 6 Registers
R10	RX Message Buffer (b) Data Field 7 Registers
R11	RX Message Buffer (b) Data Field 8 Registers
R12	RX Message Buffer (b) Data Field 9 Registers
R13	RX Message Buffer (b) Data Field 10 Registers
R14	RX Message Buffer (b) Data Field 11 Registers
R15	RX Message Buffer (b) Data Field 12 Registers

R16	RX Message Buffer (b) Data Field 13 Registers
R17	RX Message Buffer (b) Data Field 14 Registers
R18	RX Message Buffer (b) Data Field 15 Registers

每个寄存器的内部 bit 组织如下：

RX Message Buffer Component (RMBCP)																														
Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	x	CFDRMID	RMIDE	RMRTR	-	RMID																												
R1	x	CFDRMPTR	RMDLC				-	-	-	-	-	-	-	-	-	-	-	-	RMTS															
R2	x	CFDRMFDSTS	RMPTR																-	-	-	-	-	-	RMIFL	-	-	-	-	-	RMFDF	RMBRS	RMESI
R3	0	CFDRMDFp	RMDB((p*q)+(q-1))								RMDB((p*q)+(q-2))								RMDB((p*q)+(q-3))								RMDB((p*q)+(q-4))						
R[4…18]	[1…15]	CFDRMDFp	RMDB((p*q)+(q-1))								RMDB((p*q)+(q-2))								RMDB((p*q)+(q-3))								RMDB((p*q)+(q-4))						

25.18.2 RX Message Buffer ID Registers (CFDRMID)

名称	CFDRMID							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RMIDE	RMRTR	-	RMID				
位权限	R-0	R-0	U-0	R-0 0000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RMID							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RMID							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RMID							
位权限	R-0000 0000							

位号	位名	说明
31	RMIDE	RX Message Buffer IDE Bit 1'b0: STD-ID is stored 1'b1: EXT-ID is stored
30	RMRTR	RX Message Buffer RTRBit 1'b0: Data Frame 1'b1: Remote Frame
29	--	RFU：未实现，读为 0
28:0	RMID	RX Message Buffer ID Field STD-ID / EXT-ID fields

25.18.3 RX Message Buffer Pointer Registers (CFDRMPTR)

名称	CFDRMPTR							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RMDLC				-			
位权限	R-0000				U-0			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RMTS							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RMTS							
位权限	R-0000 0000							

位号	位名	说明
31:28	RMDLC	RX Message Buffer DLCField No. of Data Bytes received in a CAN Frame
27:16	--	RFU：未实现，读为 0
15:0	RMTS	RX Message BufferTimestamp Field Timestamp value stored for the message stored in the RX Message Buffer

25.18.4 RX Message Buffer CAN-FD Status Registers (CFDRMFDSTS)

名称	CFDRMFDSTS							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RMPTR[15:8]							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RMPTR[7:0]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						RMIFL[1:0]	
位权限	U-0						R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					RMFDF	RMBRS	RMESI
位权限	U-0					R-0	R-0	R-0

位号	位名	说明
31:16	RMPTR	RX Message Buffer Pointer
15:10	--	RFU：未实现，读为 0
9:8	RMIFL	RX Message BufferInformation label Field
7:3	--	RFU：未实现，读为 0
2	RMFDF	CAN FD Format bit

		1'b0: Non CAN-FD frame received 1'b1: CAN-FD frame received
1	RMBRS	Bit Rate Switch bit 1'b0: CAN-FD frame received with no bit rate switch 1'b1: CAN-FD frame received with bit rate switch
0	RMESI	Error State Indicator bit 1'b0: CAN-FD frame received from error active node 1'b1: CAN-FD frame received from error passive node

25.18.5 RX Message Buffer Data Field p Registers (CFDRMDFp)

名称	CFDRMDFp (p=0~15)							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RMDB							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RMDB							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RMDB							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RMDB							
位权限	R-0000 0000							

位号	位名	说明
31:0	RMDB	RX Message Buffer Data Byte

25.18.6 RX FIFO Access Message Buffer Component b (CFDRFMBCPb)

名称	CFDRFMBCPb (b=0,1)							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	Rc[31:24]							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	Rc[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	Rc[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	Rc[7:0]							
位权限	R-0000 0000							

位号	位名	说明

31:0	Rc	RX FIFO Access Message BufferComponent c

每个RX message buffer component包含19个寄存器：c=0~18，分别是CFDRFID，CFDRFPTR，CFDRFFDSTS，CFDRFDFp

RX FIFO Access Message Buffer Component (RFMBCP)	
Rc	
R0	RX FIFO Access ID Registers
R1	RX FIFO Access Pointer Register
R2	RX FIFO Access CAN-FD Status Registers
R3	RX FIFO Access Data Field 0 Registers
R4	RX FIFO Access Data Field 1 Registers
R5	RX FIFO Access Data Field 2 Registers
R6	RX FIFO Access Data Field 3 Registers
R7	RX FIFO Access Data Field 4 Registers
R8	RX FIFO Access Data Field 5 Registers
R9	RX FIFO Access Data Field 6 Registers
R10	RX FIFO Access Data Field 7 Registers
R11	RX FIFO Access Data Field 8 Registers
R12	RX FIFO Access Data Field 9 Registers
R13	RX FIFO Access Data Field 10 Registers
R14	RX FIFO Access Data Field 11 Registers
R15	RX FIFO Access Data Field 12 Registers
R16	RX FIFO Access Data Field 13 Registers
R17	RX FIFO Access Data Field 14 Registers
R18	RX FIFO Access Data Field 15 Registers

每个寄存器的内部 bit 组织如下：

RX FIFO Access Message Buffer Component (RFMBCP)																																
Rc	**p**	**Symbol**	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R0	x	CFDRFID	RFIDE	RFRTR	-	RFID																												
R1	x	CFDRFPTR	RFDLC			-	-	-	-	-	-	-	-	-	-	-	-	-	RFTS															
R2	x	CFDRFFDSTS	RFPTR																-	-	-	-	-	-	RFIFL	-	-	-	-	-	RFFDF	RFBRS	RFESI	
R3	0	CFDRFDFp	RFDB((p*q)+(q-1))								RFDB((p*q)+(q-2))								RFDB((p*q)+(q-3))								RFDB((p*q)+(q-4))							
R[4…18]	[1…15]	CFDRFDFp	RFDB((p*q)+(q-1))								RFDB((p*q)+(q-2))								RFDB((p*q)+(q-3))								RFDB((p*q)+(q-4))							

25.18.7 RX FIFO Access ID Registers (CFDRFID)

名称	CFDRFID							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFIDE	RFRTR	-	RFID				
位权限	R-0	R-0	U-0	R-0 0000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFID							

位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFID							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFID							
位权限	R-0000 0000							

位号	位名	说明
31	RFIDE	RX FIFO Buffer IDE Bit 1'b0: STD-ID has been received 1'b1: EXT-ID has been received
30	RFRTR	RX FIFO Buffer RTRBit 1'b0: Data Frame 1'b1: Remote Frame
29	--	RFU：未实现，读为 0
28:0	RFID	RX FIFO Buffer IDField STD-ID / EXT-ID fields

25.18.8 RX FIFO Access Pointer Register (CFDRFPTR)

名称	CFDRFPTR							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFDLC				-			
位权限	R-0000				U-0			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFTS							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFTS							
位权限	R-0000 0000							

位号	位名	说明
31:28	RFDLC	RX FIFO Buffer DLCField No. of Data Bytes received in a CAN Frame
27:16	--	RFU：未实现，读为 0
15:0	RFTS	RX FIFO Timestamp Field Timestamp value of the received CAN frame

25.18.9 RX FIFO Access CAN-FD Status Register (CFDRFFDSTS)

名称	CFDRFFDSTS							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	RFPTR[15:8]							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFPTR[7:0]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						RFIFL	
位权限	U-0						R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					RFFDF	RFBRS	RFESI
位权限	U-0					R-0	R-0	R-0

位号	位名	说明
31:16	RFPTR	RX FIFO Buffer Pointer
15:10	--	RFU：未实现，读为 0
9:8	RFIFL	RX FIFO BufferInformation label Field
7:3	--	RFU：未实现，读为 0
2	RFFDF	CAN FD Format bit 1'b0: Non CAN-FD frame received 1'b1: CAN-FD frame received
1	RFBRS	Bit Rate Switch bit 1'b0: CAN-FD frame received with no bit rate switch 1'b1: CAN-FD frame received with bit rate switch
0	RFESI	Error State Indicator bit 1'b0: CAN-FD frame received from error active node 1'b1: CAN-FD frame received from error passive node

25.18.10 RX FIFO Access Data Field p Registers (CFDRFDFp)

名称	CFDRFDFp (p=0~15)							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFDB							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFDB							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFDB							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFDB							
位权限	R-0000 0000							

位号	位名	说明
31:0	RFDB	RX FIFO Buffer Data Byte

25.18.11 Common FIFO Access Message Buffer Component b (CFDCFMBCPb)

名称	CFDCFMBCPb (b=0)							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	Rc[31:24]							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	Rc[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	Rc[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	Rc[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:0	Rc	Common FIFO Access Message BufferComponent c

每个RX message buffer component包含19个寄存器：c=0~18，分别是CFDRFID，CFDRFPTR，CFDRFFDSTS，CFDRFDFp

Common FIFO Access Message Buffer Component (CFMBCP)	
Rc	
R0	Common FIFO Access ID Registers
R1	Common FIFO Access Pointer Register
R2	Common FIFO Access CAN-FD Status Registers
R3	Common FIFO Access Data Field 0 Registers
R4	Common FIFO Access Data Field 1 Registers
R5	Common FIFO Access Data Field 2 Registers
R6	Common FIFO Access Data Field 3 Registers
R7	Common FIFO Access Data Field 4 Registers
R8	Common FIFO Access Data Field 5 Registers
R9	Common FIFO Access Data Field 6 Registers
R10	Common FIFO Access Data Field 7 Registers
R11	Common FIFO Access Data Field 8 Registers
R12	Common FIFO Access Data Field 9 Registers
R13	Common FIFO Access Data Field 10 Registers
R14	Common FIFO Access Data Field 11 Registers
R15	Common FIFO Access Data Field 12 Registers
R16	Common FIFO Access Data Field 13 Registers
R17	Common FIFO Access Data Field 14 Registers
R18	Common FIFO Access Data Field 15 Registers

每个寄存器的内部 bit 组织如下：

| Com mon FIFO Access Message Buffer Com ponent (CFMBCP) | | |

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CFTS							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CFTS							
位权限	R/W-0000 0000							

位号	位名	说明
31:28	CFDLC	Common FIFO Buffer DLCField No. of Data Bytes received in a CAN Frame
27:16	--	RFU：未实现，读为 0
15:0	CFTS	CommonFIFO Timestamp Field Timestamp value of the received CAN frame

25.18.14 Common FIFO Access CAN-FD Status Register (CFDCFFDSTS)

名称	CFDCFFDSTS							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CFPTR							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CFPTR							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CFIFL	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					CFFDF	CFBRS	CFESI
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	位名	说明
31:16	CFPTR	Common FIFO Buffer Pointer
15:10	--	RFU：未实现，读为 0
9:8	CFIFL	CommonFIFO BufferInformation label Field
7:3	--	RFU：未实现，读为 0
2	CFFDF	CAN FD Format bit 1’b0: Non CAN-FD frame received 1’b1: CAN-FD frame received
1	CFBRS	Bit Rate Switch bit 1’b0: CAN-FD frame received with no bit rate switch 1’b1: CAN-FD frame received with bit rate switch
0	CFESI	Error State Indicator bit 1’b0: CAN-FD frame received from error active node 1’b1: CAN-FD frame received from error passive node

25.18.15 Common FIFO Access Data Field p Registers (CFDCFDFp)

名称	CFDCFDFp (p=0~15)							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CFDB							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CFDB							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CFDB							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CFDB							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	CFDB	Common FIFO Buffer Data Byte

25.18.16 TX Message Buffer Component b (CFDTMBCPb)

名称	CFDTMBCPb (b=0~3)							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	Rc[31:24]							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	Rc[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	Rc[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	Rc[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:0	Rc	TX Message BufferComponent c The TX Message Buffer Component is made up of the following registers: CFDTMID, CFDTMPTR,CFDTMFDSTS, and CFDTMDFp.

每个RX message buffer component包含19个寄存器：c=0~18，分别是CFDTMID，CFDTMPTR，CFDTMFTSTS，CFDTMDFp

TX Message Buffer Component (TMBCP)	
Rc	
R0	TX Message Buffer (b) ID Registers CHn
R1	TX Message Buffer (b) Pointer Registers CHn
R2	TX Message Buffer (b) CAN-FD Status Registers CHn
R3	TX Message Buffer (b) Data Field 0 Registers CHn
R4	TX Message Buffer (b) Data Field 1 Registers CHn
R5	TX Message Buffer (b) Data Field 2 Registers CHn
R6	TX Message Buffer (b) Data Field 3 Registers CHn
R7	TX Message Buffer (b) Data Field 4 Registers CHn
R8	TX Message Buffer (b) Data Field 5 Registers CHn
R9	TX Message Buffer (b) Data Field 6 Registers CHn
R10	TX Message Buffer (b) Data Field 7 Registers CHn
R11	TX Message Buffer (b) Data Field 8 Registers CHn
R12	TX Message Buffer (b) Data Field 9 Registers CHn
R13	TX Message Buffer (b) Data Field 10 Registers CHn
R14	TX Message Buffer (b) Data Field 11 Registers CHn
R15	TX Message Buffer (b) Data Field 12 Registers CHn
R16	TX Message Buffer (b) Data Field 13 Registers CHn
R17	TX Message Buffer (b) Data Field 14 Registers CHn
R18	TX Message Buffer (b) Data Field 15 Registers CHn

每个寄存器的内部 bit 组织如下：

TX Message Buffer Component (TMBCP)																															
Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	x	CFDTMID	TMIDE	TMRTR	THLEN	TMID																												
R1	x	CFDTMPTR	TMDLC																															
R2	x	CFDTMFDCTR	TMPTR																						TMIFL							TMFDF	TMBRS	TMESI
R3	0	CFDTMDFp	TMDB((p*q)+(q-1))												TMDB((p*q)+(q-2))										TMDB((p*q)+(q-3))									
R[4…18]	[1…15]	CFDTMDFp	TMDB((p*q)+(q-1))												TMDB((p*q)+(q-2))										TMDB((p*q)+(q-3))									

25.18.17 TX Message Buffer ID Registers (CFDTMID)

名称	CFDTMID							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TMIDE	TMRTR	THLEN	TMID[28:24]				
位权限	R/W-0	R/W-0	R/W-0	R/W-0 0000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TMID[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TMID[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TMID[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31	TMIDE	TX Message Buffer IDE Bit 1'b0: STD-ID will be transmitted 1'b1: EXT-ID will be transmitted
30	TMRTR	TX Message Buffer RTRBit 1'b0: Data Frame 1'b1: Remote Frame
29	THLEN	Tx History List Entry 1'b0: Entry not stored in THL after successful TX 1'b1: Entry stored in THL after successful TX
28:0	TMID	TX Message Buffer ID Field STD-ID / EXT-ID fields

25.18.18 TX Message Buffer Pointer Registers (CFDTMPTR)

名称	CFDTMPTR							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TMDLC				-			
位权限	R/W-0000				U-0			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							
位权限	U-0							

位号	位名	说明
31:28	TMDLC	TX Message Buffer DLCField No. of Data Bytes to be transmitted in a CAN Frame
27:16	--	RFU：未实现，读为 0
15:0	--	RFU：未实现，读为 0

25.18.19 TX Message Buffer CAN-FD Control Register (CFDTMFDCTR)

名称	CFDTMFDCTR							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TMPTR							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TMPTR							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	-						TMIFL	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					TMFDF	TMBRS	TMESI
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	位名	说明
31:16	TMPTR	TX Message Buffer Pointer
15:10	--	RFU：未实现，读为 0
9:8	TMIFL	TX Message BufferInformation label Field
7:3	--	RFU：未实现，读为 0
2	TMFDF	CAN FD Format bit 1'b0: Non CAN-FD frame to transmit 1'b1: CAN-FD frame to transmit
1	TMBRS	Bit Rate Switch bit 1'b0: CAN-FD frame to transmit with no bit rate switch 1'b1: CAN-FD frame to transmit with bit rate switch
0	TMESI	Error State Indicator bit 1'b0: CAN-FD frame to transmit by error active node 1'b1: CAN-FD frame to transmit by error passive node

25.18.20 TX Message Buffer Data Field p Registers (CFDTMDFp)

名称	CFDTMDFp (p=0~15)							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TMDB							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TMDB							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TMDB							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TMDB							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	TMDB	TX Message Buffer Data Byte

26 单边沿半字节传输模块（SENT）

26.1 概述

SENT（Single Edge Nibble Transmission）模块是基于SAE J2716协议的单向、单线串行通信接口，

本接口主要用于车载传感器与ECU之间的数据通信。

SENT模块主要特性如下：

- 符合SAE J2716协议
- 可配置为发送或接收模式
- 自动数据波特率同步
- CRC计算和CRC检测功能
- 1个消息帧的数据缓存
- 可选数据长度
- 可选的暂停脉冲
- 帧格式错误检测

SENT通信时序基于预定义的时间单元tick，接收方和发送方都要在通信开始前预先配置相同的tick周期，支持的tick周期范围是3us到90us。

一个SENT消息帧从同步脉冲开始，同步脉冲用来让接收方计算发送方的数据波特率，SENT协议允许tick周期出现20%的变化，从而降低对通信双方的时钟精度要求。

26.2 消息格式

26.2.1 Data Nibble

SENT数据传输基于nibble形式，每个nibble总是用来编码4bit数据，编码形式是nibble的长度。每个nibble总是以4 tick长度的低电平开始，nibble最小长度为12 tick，代表数据0000，最大长度为27 tick，代表数据1111：

- Data = 4’b0000: nibble is 12 ticks
- Data = 4’b0001: nibble is 12+1=13 ticks
- ...
- Data = 4’b1111: nibble is 12+15=27 ticks

Nibble波形示意图如下：

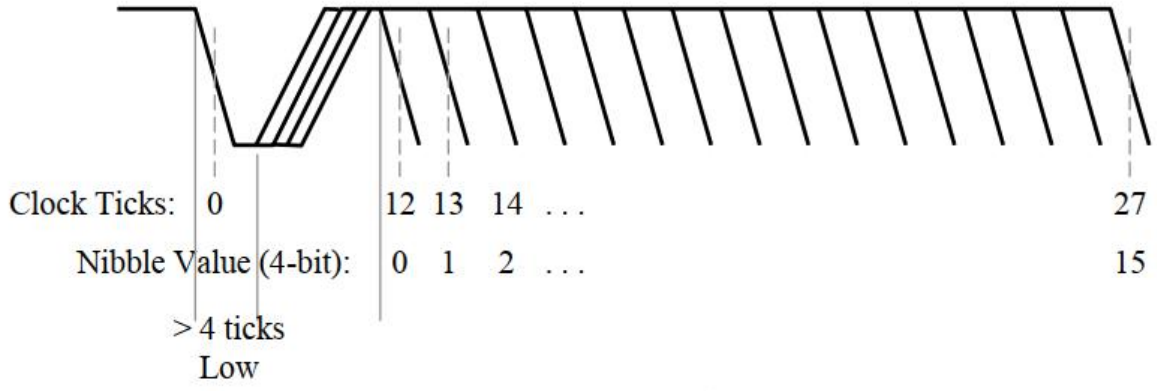


图 26-1nibble 波形示意图

发送数据0和发送数据7时的nibble波形如下所示：

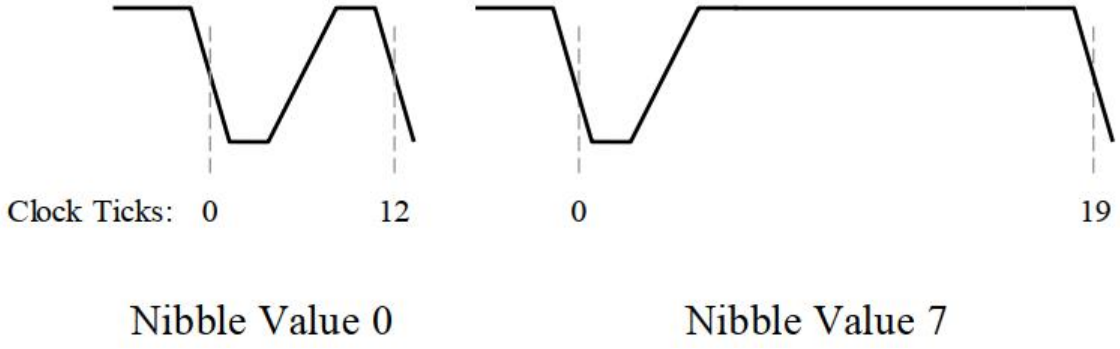


图 26-2 数据 0 和数据 7 的 nibble 波形

26.2.2 同步脉冲（Sync Pulse）

同步脉冲用于接收方波特率计算，名义长度是56 ticks，以4 ticks低电平为起始。

通信双方虽然事先约定了tick长度，但是考虑到时钟偏差，实际tick长度可能与理想值不同，通过检测sync pulse的实际长度，接收方可以计算实际的tick长度，并校准采样时钟。

26.2.3 Status and communication nibble

这个nibble在同步脉冲之后、数据nibble之前发送，用于传输特定辅助信息，其中最低2bit为保留位。注意status and communication nibble不包含在帧CRC计算中。

Bit2和bit3可作为可选的消息通道使用，支持短消息格式（short serial message format）或加强消息格式（enhanced serial message format）。

注意：使用status and communication nibble用作option channel时，收发的数据和识别pattern注入、CRC计算都是由软件完成的。

26.2.3.1 Short serial message format

短消息格式中，使用status and communication nibble的bit 2来传输额外的16bit串行数据，由于每个帧用来传输数据的仅1bit，所以需要连续16帧完成短消息传输，并且连续16帧必须满足无数据格式

错误和CRC错误的条件。

作为短消息格式的识别，第一帧的status and communication nibble的bit 3为1，而后续15帧bit 3连续为0；16bit短消息中包含4bit消息ID、8bit数据字节、4bit CRC校验和。CRC校验包含消息ID和数据字节，校验算法与data nibble相同。

短消息发送格式如下图所示：

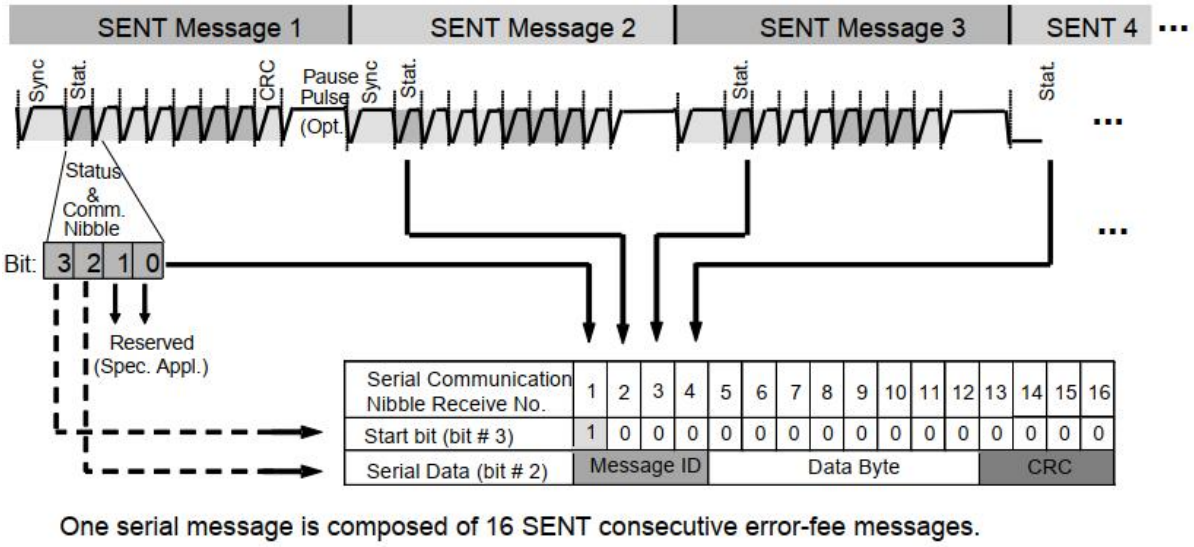


图 26-3 短消息格式 option channel

26.2.3.2 Enhanced serial message format

另一种option channel格式为加强消息格式。加强消息格式中，需要连续18个帧，并且同时使用status and communication nibble的bit 2和bit 3来做通信。连续18帧必须同时满足无数据格式错误和CRC错误的条件。

加强消息帧通过特殊的bit3 pattern来识别：“01111110”，这个pattern在消息帧的#18、#1~#7中分别发送。

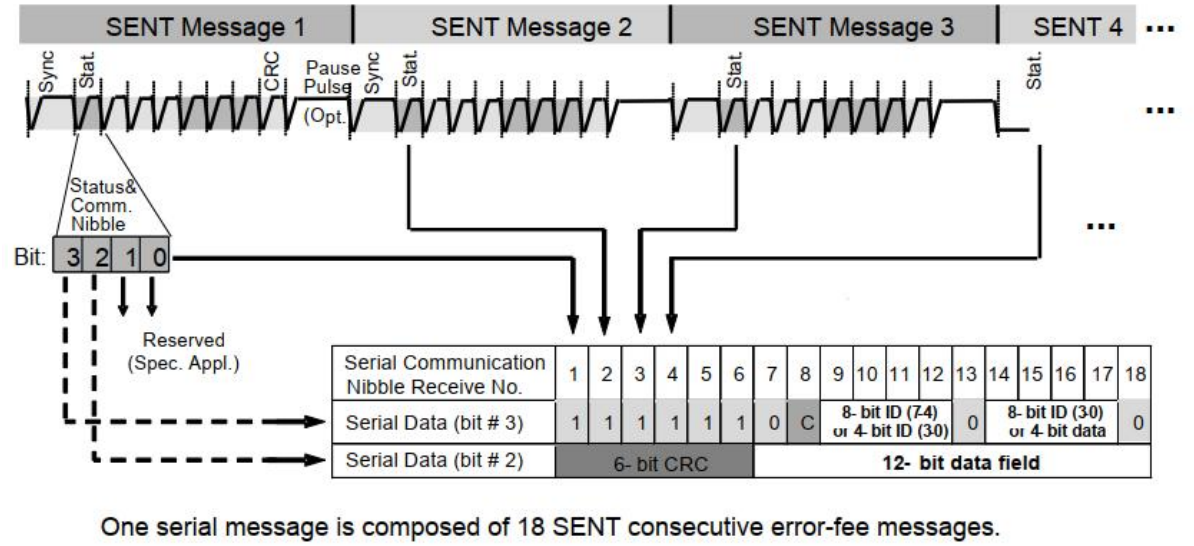


图 26-4 加强消息格式 option channel

使用加强消息格式时，option channel中传输数据的CRC算法与data nibble不同，具体请参见SAE J2716协议。

26.2.4 消息帧格式

一个完整的消息帧包含了同步脉冲，状态和通信nibble，数据nibble，CRC nibble，可选的暂停脉冲，如下图所示：

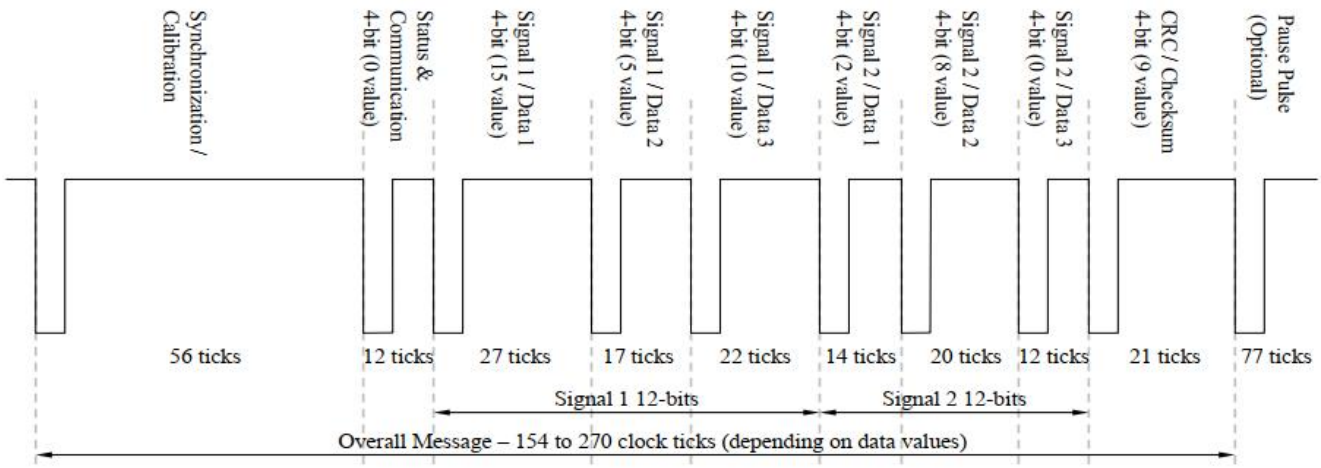


图 26-5 消息帧格式

消息帧中数据nibble个数是由应用决定的，不同应用使用预先定义的、固定的的nibble个数。比如传感器需要发送24bit数据给ECU，则nibble个数为6。本模块支持1~6 data nibbles。

26.2.5 Optional Pause Pulse

Pause pulse用于填充消息帧，使得所有的消息帧都可以有相同的长度（tick个数）。FRAMETIME寄存器用于定义整个消息帧的长度，虽然这是一个12bit寄存器，SAE J2716协议规定了Pause脉冲最长不超过768 ticks，因此整个消息帧长度不会超过1038 ticks（270+768），配置FRAMETIME大于1038是没有意义的。

$$FRAMETIME = T_{FRAME} / T_{tick}$$

26.2.6 CRC 校验

SENT通信仅针对data nibble使用4bit CRC，status&communication nibble不参与CRC计算。

CRC多项式：x^4+x^3+x^2+1，初始化种子0101

注意：新版标准建议发送时计算CRC需要在数据后加上一个0000的输入增强：参见SAE J2716 Appendix B

Table B.1-3 – Example 4-bit CRC calculation by polynomial division

		seed	nibble1	nibble2	nibble3	nibble4	nibble5	nibble6	augmentation
Seed/Aug	5/0	5							0
data (hex)	4AC4AC		4	A	C	4	A	C	
data (bin)		0 1 0 1	0 1 0 0	1 0 1 0	1 1 0 0	0 1 0 0	1 0 1 0	1 1 0 0	0 0 0 0

为了与legacy标准保持兼容，可以通过CRCM寄存器配置是否在发送模式计算CRC时加入0000增强。

26.3 结构框图

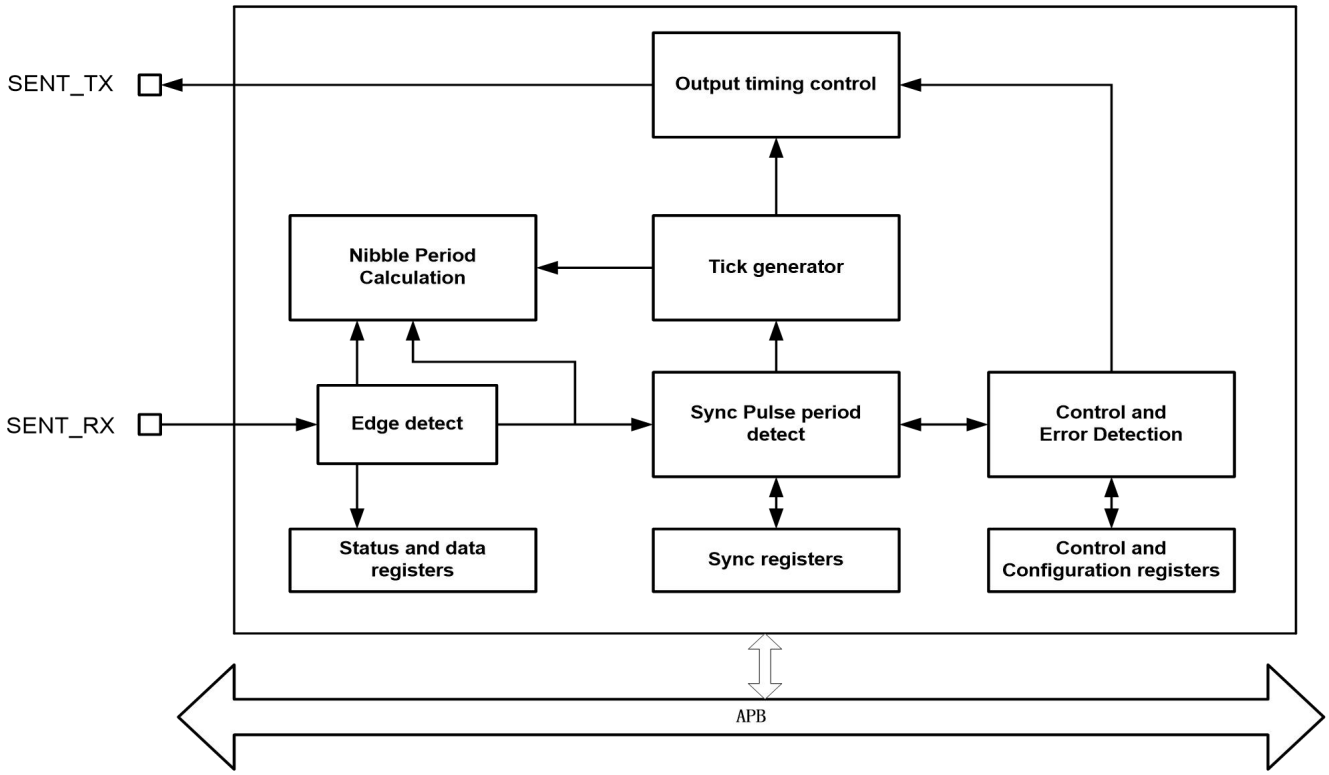


图 26- 6SENT 模块结构框图

26.4 功能描述

26.4.1 发送 Tick 时间

SENT模块使用APBCLK工作。由于SENT协议规定的tick长度是3~90us，即tick时钟频率11.1KHz~333KHz，根据Prescaler寄存器设置，对APBCLK分频后得到工作时钟F_{SENT}，这个时钟固定为Tick频率的8倍。

$$F_{SENT} = F_{APBCLK}/(Prescaler + 1)$$

$$T_{tick} = \frac{8}{F_{SENT}}$$

模块工作前，需要在CMU和RMU模块中配置相关寄存器，使能模块工作时钟并释放模块复位。

26.4.2 发送模式

SENT模块配置为发送模式时，可以支持连续发送或软件触发发送模式。

- 连续发送：软件使能发送后，SENT模块连续不断发送寄存器中的数据，软件应负责及时填充新的数据；如果数据更新不及时，SENT模块将重复发送已经发送过的数据
- 软件触发：SENT模块完成一帧发送后，会等待软件重新使能发送

发送初始化流程：

3. 清零MOD，进入发送模式
4. 配置TXM选择连续发送或软件触发模式
5. 配置Tick长度
6. 配置数据nibble个数
7. 配置CRCEN和CRCM，选择硬件计算CRC或软件计算CRC，以及是否进行CRC增强
8. 配置PPEN选择是否要发送pause pulse
9. 如果PPEN=1，向FRAMETIME寄存器写入帧长度
10. 向数据寄存器写入待发送的数据和Status&Communication nibble；如果CRCEN=0则同时需写入软件计算的CRC值，否则CRC由硬件计算并自动更新。
11. 根据需要使能中断
12. 如果CRCEN=0，软件计算CRC并写入CRC寄存器
13. 如果CRCEN=1，软件通过置位CCRC寄存器触发硬件计算CRC
14. 置位发送使能，开始发送（CCRC和TXEN可以同时置位）

注意：在连续发送的模式下，每个data nibble发完都可以及时更新data寄存器，所有datanibble更新完毕后，通过置位CCRC更新CRCnibble；以上操作必须在下一个sync发送前完成。

注意：参与CRC计算的data nibble个数由NIBCNT指定。

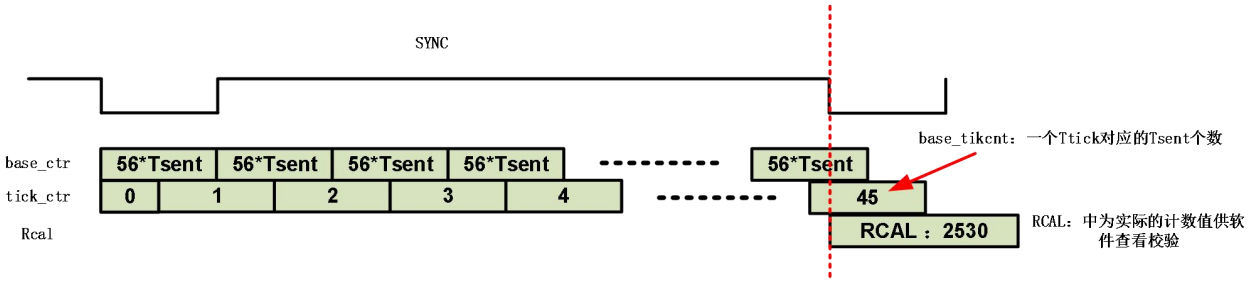
26.4.3 接收模式

接收模式下，SENT模块自动计算sync pulse的宽度，采样到的sync pulse宽度处于SYNCMIN和

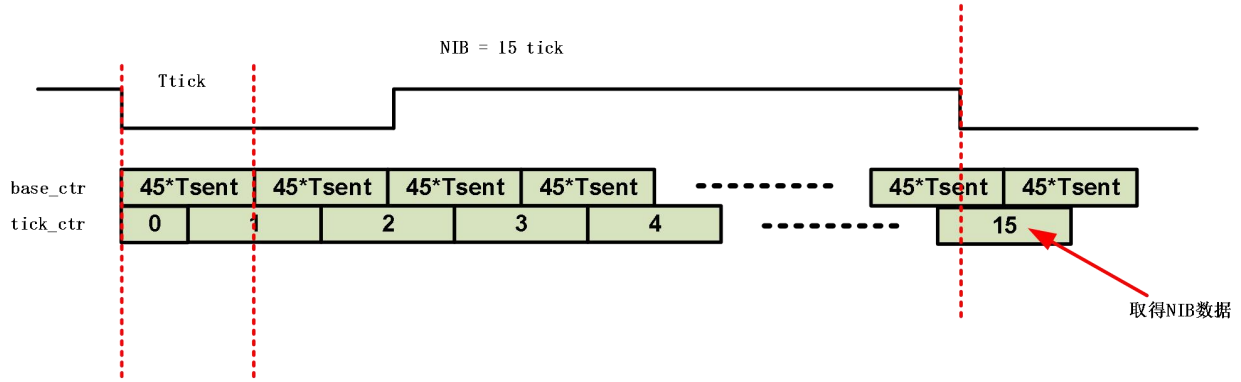
SYNCMAX寄存器定义的范围之内时，模块开始正常数据接收。Sync脉冲宽度被自动写入RCAL寄存器中，用作后续数据接收时的校准参数。

在接收SYNC阶段，会在sent_rx的下降沿开始以 $F_{SENT} = F_{APBCLK} / (prescaler + 1)$ 的时钟频率计数按照每56计数加1一直到下一个下降沿，期间计算多少个56*T_{SENT}即可以得到当前的T_{TICK}，此处会进行round操作，每一轮计数过半才能加1，需要注意的是，为了减小误差到小于1/64，需要根据使用的Ttick调节pres分频选取合适的F_{SENT}操作，让tick_ctr的值大于32，避免短的Ttick使用较低频率。

例如下图 sync中得到的base_tickcnt=45，则表示1*T_{TICK} =45*T_{SENT}



在接收数据阶段，base_ctr则按照SYNC过程中得到的对应于Tick的计数值进行计数，得到最终的NIB长度。如下图



SYNCMIN和SYNCMAX用于接收诊断，因为协议允许+/-20%的误差容限，而当误差超过+/-20%时，应产生错误事件；

计算这两个值需要根据通信Ttick的时间，当前Fsent的频率，计算理论SYNC的周期数，SYNC = Ttick*56*Fsent

举例：Ttick = 6us，Fsent=8M，SYNC = 6*8*56 = 2688

SYNCMIN = 0.8* SYNC=2150

SYNCMAX = 1.2* SYNC=3225

超出这个范围的将会报错并继续接收新的SYNC过程。

接收初始化流程：

1. 根据Ttick配置合适的模块频率与预分频值
2. 置位MOD，进入接收模式
3. 配置数据nibble个数
4. 配置CRCEN，选择硬件计算CRC或软件计算CRC
5. 配置PPEN选择是否要接收pause pulse
6. 配置SYNCMAX和SYNCMIN
7. 根据需要使能中断
8. 置位接收使能，开始接收

SENT模块在接收data nibble过程中CRC模块自动计算校验和，并将结果与接收到的CRC nibble进行比较，如果校验和错误将置位CRCERR标志，并可以触发中断。

26.4.4 接收消息诊断

接收模式下，当以下情况发生时，认为遇到了错误，模块产生错误标志和中断。

- Sync pulse大于SYNCMAX或小于SYNCMIN
- 两个sync pulse之间的下降沿个数与预期不符（消息帧长度是预定义的，比如data nibble个数为6，PPEN=1，则两个syncpulse之间应出现9个下降沿，否则就是格式错误）
- CRC错误
- Data nibble计算结果<0（小于12ticks）或>15（大于27ticks）
- 连续两个sync pulse测量值误差超过了+/-1.5625%（1/64）

注意：即使检测到错误事件（CRC错误，和连续2个syncpulse测量误差超过了1/64），SENT模块仍会将解码的数据写入寄存器，软件应自行判断如何处理数据。（其他错误会导致数据无法写入，无法解码）

26.5 寄存器

SENT0模块基地址：0x4001B000

SENT1模块基地址：0x4001B400

offset 地址	名称	符号
0x00	SENT 控制寄存器	SENT_CR
0x04	SENT 配置寄存器	SENT_CFGR
0x08	SENT 预分频寄存器	SENT_PSR
0x0C	SENT 帧长度寄存器	SENT_FTR
0x10	SENT 状态和标志寄存器	SENT_SR
0x14	SENT 数据寄存器	SENT_DR
0x18	SENT 同步脉冲诊断寄存器	SENT_SYNC
0x1C	SENT 同步校准寄存器	SENT_RCAL

26.5.1 SENT 控制寄存器（SENT_CR）

名称	SENT_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					-	RXEN	TXEN
位权限	U-0					U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	RXEN	SENT 接收使能，仅当 MOD=0 时可以置位 1：开始接收数据 0：关闭接收功能
0	TXEN	SENT 发送使能，软件置位启动发送，仅当 MOD=1 时可以置位 当 TXM=1，即连续发送模式时，TXEN 置位后保持为 1，直到软件清零，SENT 模块会不断发送数据，软件应保证及时更新数据寄存器 当 TXM=0，即软件触发模式时，TXEN 在当前帧发送完成后自动清零，SENT 模块发送完一帧后停止，等待下次软件置位 TXEN

26.5.2 SENT 配置寄存器（SENT_CFGR）

名称	SENT_CFGR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名			-					
位权限			U-0					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		FTXIE	NTXIE	SYNCIE	ERRIE	FRXIE	NRXIE
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					NIBCNT		
位权限	U-0					R/W-110		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		CRCM	PPEN	CRCEN	TXPOL	TXM	MOD
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:22	--	RFU：未实现，读为 0
21	FTXIE	Frame 发送完成中断使能
20	NTXIE	Nibble 发送完成中断使能
19	SYNCIE	Sync pulse 中断使能 1：syncpulse 正确接收（宽度在 SYNCMIN 和 SYNCMAX 之间）后产生中断 0：syncpulse 不产生中断
18	ERRIE	错误事件中断使能 1：错误事件发生时产生中断 0：不产生错误中断（标志寄存器仍会置位）
17	FRXIE	Frame 接收完成中断使能
16	NRXIE	Nibble 接收完成中断使能
10:8	NIBCNT	帧中 Data Nibbles 个数 000: forbidden 001: 1 nibble 010: 2 nibbles … 110: 6 nibbles 111: forbidden
7:6	--	RFU：未实现，读为 0
5	CRCM	CRC 模式选择，仅发送模式有效 0:recommended mode, augmented with 0000 1: legacy mode, no augment
4	PPEN	暂停脉冲（pause pulse）使能 1：帧发送完成后发送 PP，PP 结束后才能启动下一帧发送 0：帧发送完成后不发送 PP，直接启动下一帧发送
3	CRCEN	硬件 CRC 使能 1：使能硬件 CRC 计算 0：禁止硬件 CRC 计算
2	TXPOL	发送空闲状态下的输出电平 0：发送空闲时 SENT_TX 输出高电平 1：发送空闲时 SENT_TX 输出低电平

Bit	助记符	功能描述
1	TXM	发送模式选择 0：软件触发模式 1：连续发送模式
0	MOD	SENT 工作模式 0：接收模式 1：发送模式

26.5.3 SENT 预分频寄存器（SENT_PSR）

名称	SENT_PSR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PRESCALER							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PRESCALER							
位权限	R/W-0000 0001							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PRESCALER	SENT 时钟预分频，根据这个寄存器的值对 APBCLK 进行预分频得到 SENT 工作时钟频率 实际分频值是 PRESCALER+1 $F_{SENT}=F_{APBCLK}/(PRESCALER+1)$

26.5.4 SENT 帧长度寄存器（SENT_FTR）

名称	SENT_FTR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名					FRAMETIME			
位权限					R/W-000			

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FRAMETIME							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11:0	FRAMETIME	发送帧长度，单位是 tick，包含 sync、status、data、CRC 和 pause 在内的总长度；仅发送模式且 PPEN=1 时有效，据此可以计算出 pause 脉冲的长度

26.5.5 SENT 状态和标志寄存器（SENT_SR）

名称	SENT_SR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		NIB			RXIDLE	PAUSE	SYNC
位权限	U-0		R-000			R-0	R-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		TCOERR	SSERR	NBERR	CRCERR	NCNTERR	SYNCERR
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		FTXF	NTXF	SYNCF	-	FRXF	NRXF
位权限	U-0		R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:22	--	RFU：未实现，读为 0
21:19	NIB	Nibble 收发状态指示 000：正在收发 status nibble 001：正在收发 1st data nibble 010：正在收发 2nd data nibble … 110：正在收发 6nddatanibble 111：正在收发 CRCnibble
18	RXIDLE	接收空闲状态 1：接收到高电平的时间已经超过了 SYNCMAX 0：采样到低电平，处于接收过程中
17	PAUSE	Pause 状态指示 1：正在接收/发送 pause 0：未在接收/发送 pause
16	SYNC	Sync 状态指示 1：正在接收/发送 sync 0：未在接收/发送 sync
15:14	--	RFU：未实现，读为 0

Bit	助记符	功能描述
13	TCOERR	Tick counter overflow error tick 计数器溢出错误，此错误可能会在接收 SYNC 的时候出现，如果模块时钟和 pres 配置与 Tick 的长度不匹配，比如过长的 Tick 和过慢的时钟，则会出现溢出错误，推荐计算参考 1.6.3 接收 Tick 时间章节，软件写 1 清零 仅配置为接收工作时有效
12	SSERR	连续 Sync（Successive Sync）长度误差超范围，硬件置位，软件写 1 清零 当连续两个 sync 长度测量值误差超过 1/64 时置位
11	NBERR	Nibble 测量错误（<12ticks 或>27ticks）标志，硬件置位，软件写 1 清零
10	CRCERR	CRC 校验错误标志，硬件置位，软件写 1 清零
9	NCNTERR	两个 sync 脉冲之间的 nibble 个数与设定的帧格式不符 硬件置位，软件写 1 清零
8	SYNCERR	Sync 脉冲宽度小于 SYNCMIN 或大于 SYNCMAX，硬件置位，软件写 1 清零
7:6	--	RFU：未实现，读为 0
5	FTXF	Frame 发送完成标志，硬件置位，软件写 1 清零
4	NTXF	Nibble 发送完成标志，硬件置位，软件写 1 清零
3	SYNCF	Sync 脉冲接收完成标志，硬件置位，软件写 1 清零 此标志置位的同时更新 RCAL
2	--	RFU：未实现，读为 0
1	FRXF	Frame 接收完成标志，硬件置位，软件写 1 清零
0	NRXF	Nibble 接收完成标志，硬件置位，软件写 1 清零

26.5.6 SENT 数据寄存器（SENT_DR）

名称	SENT_DR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CRC				DATA6			
位权限	R/W-0000				R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DATA5				DATA4			
位权限	R/W-0000				R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DATA3				DATA2			
位权限	R/W-0000				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DATA1				STATUS			
位权限	R/W-0000				R/W-0000			

Bit	助记符	功能描述
31:28	CRC	CRC nibble 发送模式： 当 CRCEN=1 时，此数据由硬件在发送过程中自动计算产生，软件只可读、不可写 当 CRCEN=0 时，此数据由软件在发送前写入

Bit	助记符	功能描述
		接收模式： 此数据为收到的 CRCnibble 值；当 CRCEN=1 时，硬件在接收过程中自动计算 CRC，并将结果与此数据比较；当 CRCEN=0，硬件不做 CRC 计算和比较，由软件自行校验
27:24	DATA6	Data nibble 6
23:20	DATA5	Data nibble 5
19:16	DATA4	Data nibble 4
15:12	DATA3	Data nibble 3
11:8	DATA2	Data nibble 2
7:4	DATA1	Data nibble 1 发送时由软件写入，接收时由硬件更新，软件读取
3:0	STATUS	Status and communication nibble 发送时由软件写入，接收时由硬件更新，软件读取

26.5.7 SENT 同步脉冲诊断寄存器（SENT_SYNC）

名称	SENT_SYNC							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				SYNCMAX			
位权限	U-0				R/W-1111			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SYNCMAX							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				SYNCMIN			
位权限	U-0				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SYNCMIN							
位权限	R/W-0000 0001							

Bit	助记符	功能描述
31:28	--	RFU：未实现，读为 0
27:16	SYNCMAX	Sync 长度上限，检测到 sync 长度大于此值时产生错误事件 单位是 F_{SENT} 时钟周期，即 ticks/8
15:12	--	RFU：未实现，读为 0
11:0	SYNCMIN	Sync 长度下限，检测到 sync 长度小于此值时产生错误事件 单位是 F_{SENT} 时钟周期，即 ticks/8

26.5.8 SENT 同步校准寄存器（SENT_RCAL）

名称	SENT_RCAL							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-		BASECNT					
位权限	U-0		R-000000					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				RCAL			
位权限	U-0				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RCAL							
位权限	R-0000 0000							

Bit	助记符	功能描述
31:22	--	RFU：未实现，读为 0
21:16	BASECNT	保存同步过程中记录下的 Tick 时间对应的技术值，结合当前时钟可计算出模块得到的 tick 宽度 仅配置为接收工作时有效 **单位是 F_{SENT} 时钟周期**
15:12	--	RFU：未实现，读为 0
11:0	RCAL	SyncCalibration，只读，保存 syncpulse 测量结果，单位 ticks/8

27 直接存储访问控制器（DMA）

27.1 概述

- 7通道外设PDMA，支持Peripherals<>RAM传输
- 1通道TSI电容触摸PDMA，支持15个通道指针
- 1通道存储器MDMA，支持Flash<>RAM传输
- 外设DMA传输由外设请求触发，DMA工作期间不影响CPU运行
- 外设通道最大传输长度8192次，支持byte/half-word/word传输
- Flash->RAM通道最大传输长度8192次，只支持word传输
- 支持Flash连续编程（RAM->Flash），需要预先进行擦除，一次编程固定为256字节
- RAM指针递增、递减
- 可产生半程中断和全程中断
- 通道优先级可配置（4级优先级）

27.2 工作原理

外设 DMA 为 Peripheral<>RAM 通道，采用外设请求触发方式进行数据传输，每个外设通道都可以支持外设->RAM 或者 RAM->外设的数据传输，并且根据目标外设类型的不同，自适应选择 byte/half-word/word 传输方式。DMA 作为 Master，在收到 request 后将发起 AHB transactions 进行数据操作，外设目标地址根据通道接入选择自动定位，RAM 目标地址则根据寄存器配置定位。

TSI DMA 为电容按键触摸控制器与 RAM 之间的专用通道，可以支持最大 15 个扫描通道，并将扫描数据分散到 RAM 不同地址。

每个 channel 可以从多个外设中选择一个作为 source 或 destination，同时软件可以设置通道优先级，当两个通道同时要访问 RAM 时，由优先级决定谁先访问，另一个通道将被挂起，直到优先通道访问完毕。

外设请求可以是准备发送（RAM/Flash->Peripheral）或接收完成（Peripheral->RAM），数据传输通过 AHB 总线完成，当 DMA 访问外设时，CPU 对同一个外设的访问将引起冲突，哪个 Master 访问被挂起取决于 BusMatrix 设置的仲裁优先级。这里需要注意的是，由于大部分外设都被挂在 APB 总线上，APB 映射到 AHB 仅为一个 slave，因此当 DMA 访问 APB 中任意外设时，CPU 即使访问 APB 下的其他外设，也同样会引起总线仲裁。通过 DIR 寄存器可以配置每个通道的传输方向，软件必须保证传输方向配置与实际挂载到这个通道上的外设请求相一致。比如通道 1 当前挂载的外设请求是 UART0 接收，则必须将 DIR 寄存器配置为 0（数据从外设读出，写入 RAM），每次 UART0 接收完一帧数据，将发送 RXD0 请求给 DMA，DMA 响应请求后，从 UART0 接收缓存寄存器读取数据，如果 DIR 被错误的配置为 1，则 DMA 对 UART0 接收缓存寄存器的写操作将被 UART0 忽略。

软件可设置 DMA 的存储器指针，用于配置 DMA 传输的起始地址，可以选择指针递增或递减方式。另有 TRFLEN 寄存器配置传输次数，根据起始地址和传输次数，计算得到终止地址，当存储器指针指向终止地址时，本次传输结束，关闭通道。

当 channel 被使能后，DMA 就准备好接受通道所选中的外设请求。当配置传输长度一半的字节被传输后，一个 HTIF（Half transfer interrupt flag）中断置位；当配置传输长度全部完成后，TCIF（Transfer complete interrupt flag）中断置位。上述中断都可以被相应的中断使能寄存器屏蔽。

在 DMA 一个完整 transfer block 完成之前，软件随时可以关闭 channel 使能，此时 DMA 将被挂起，如果软件此后重新使能通道，则 DMA 继续执行之前挂起的操作。

27.3 结构框图

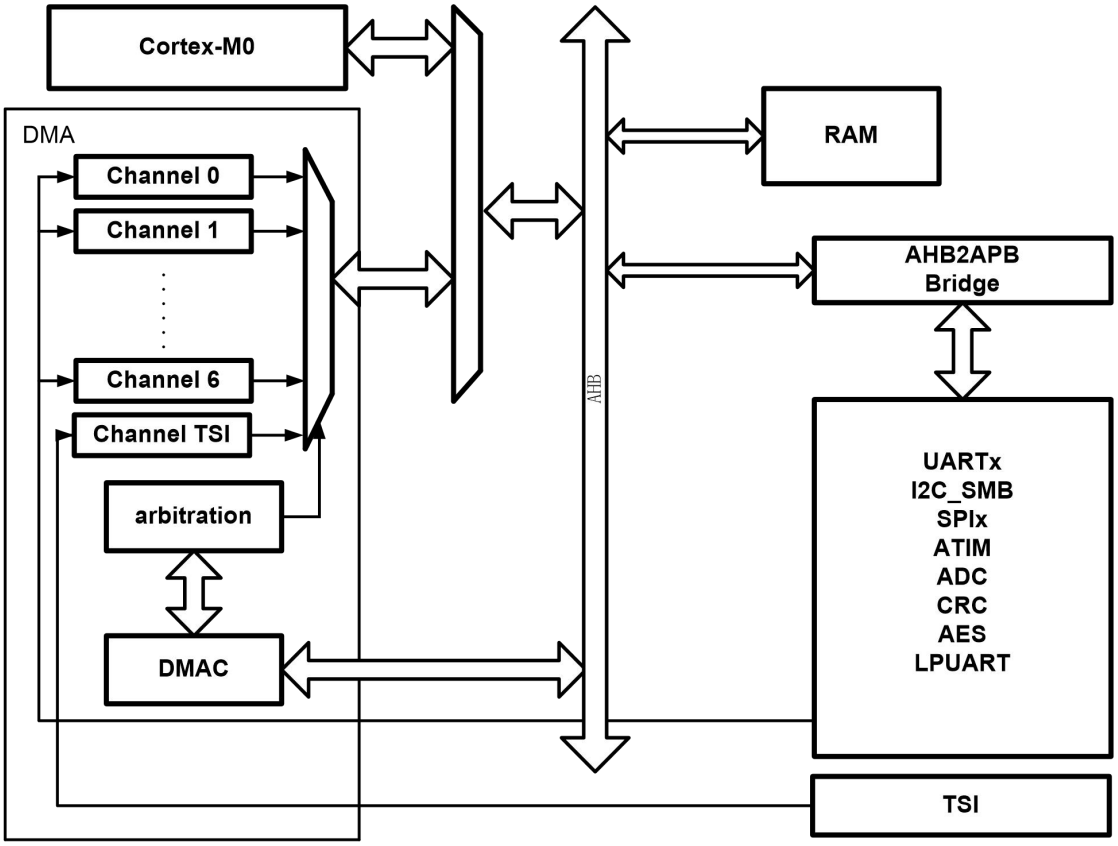


图 27-1 DMA 结构框图

27.4 工作流程

DMA 寄存器配置：



图 27-2 DMA 寄存器配置

DMA 对请求响应分成两部分处理：通道请求处理过程和数据搬运过程。

- 通道请求处理

  a) DMA 接受到请求，跳到步骤 b

  b) 判断是否有其他通道正在搬运数，若有，则停留在步骤 b 直至其他通道当次搬运完成；若无，进一步判断是否有其他同时置起的请求信号，若有，则判断当前通道优先级是否高于其他通道，若是，则跳到步骤 c 并向数据搬运过程发起请求，若否，则停留在步骤 b 直至其他通道当次搬运完成

  c) 并等待数据搬运完成响应信号，得到响应则，跳到步骤 d，否则停在步骤 c

  d) 数据搬运长度+1，判断是否达到设定长度，若是则产生通道使能关闭脉冲；判断请求是否释放，若是，则跳到步骤 a，若否，则停留在步骤 d 判断数据传输达到设定长度，否则跳到步骤 a

- 数据搬运

  a) 等待通道请求处理过程发起请求

  b) 向 HADDR 写源地址

  c) 向 HADDR 写目的地址，同时读取 HRDATA

d) 将读到的 HRDATA 数据写到 HWDATA

e) 向通道请求处理过程发出搬运完成响应，并跳到步骤 a

DMA 工作的流程如下图所示：

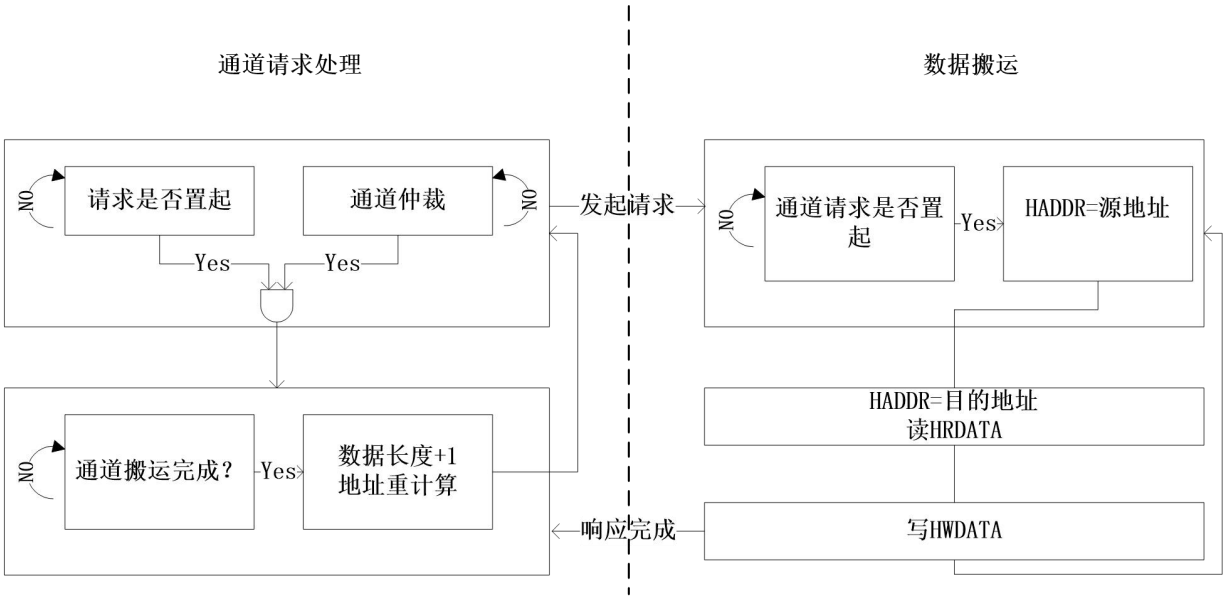


图 27-3 DMA 工作流程

27.5 访问带宽

DMA 外设通道支持字节/半字/字访问，每个通道都可以通过通道控制寄存器中的 BDW 位来配置传输带宽。

27.6 通道控制

27.6.1 DMA 请求映射

DMA 共有 7 个优先级可配的外设通道，每个通道可接受 8 个请求响应，根据每个通道的配置寄存器选择其中一个请求送入通道控制器，通道控制器根据各个通道的 busy 状态和优先级选择其中一个通道请求进行响应处理，外设请求映射如下。

编号	外设	通道0	通道1	通道2	通道3	通道4	通道5	通道6
0	***ADC***	ADC				ADC		ADC
1	***SPI0***		SPI0_RX	SPI0_TX			SPI0_RX	SPI0_TX
2	***SPI1***				SPI1_RX	SPI1_TX		
3	***SPI2***						SPI2_RX	SPI2_TX
4	***SPI3***	SPI3_RX	SPI3_TX					
5	***UART0***		RXD0	TXD0	RXD0	TXD0		
6	***UART1***				RXD1	TXD1	RXD1	TXD1
7	***UART2***	RXD2	TXD2					
8	***UART3***			RXD3		TXD3	RXD3	TXD3
9	***UART4***			RXD4	TXD4			
10	***UART5***					RXD5		TXD5
11	***ATIM***	ATIM_CH1 (Burst not supported)	ATIM_CH2	ATIM_CH3	ATIM_CH4	ATIM_TRIG ATIM_COM ATIM_UEV		
12	***I2C_SMB0***			I2C0_TX			I2C0_RX	
13	***I2C_SMB1***					I2C1_RX		I2C1_TX
14	***AES***	AES_IN	AES_OUT					
15	***CRC***	注[1]					CRC	
16	***GPTIM0***	GPTIM0_CH1	GPTIM0_CH2	GPTIM0_CH3	GPTIM0_CH4		GPTIM0_TRIG GPTIM0_UEV	
17	***GPTIM1***	GPTIM1_CH1	GPTIM1_CH2	GPTIM1_CH3	GPTIM1_CH4			GPTIM1_TRIG GPTIM1_UEV
18	***DAC***				DAC		DAC	

表 27-1PDMA 通道映射列表

注[1]：通道0的CH0SSEL=0x5不可用

外设请求映射通过CHxSSEL寄存器配置，上表中从上到下分别表示CHxSSEL=0~7情况下有效的外设请求信号。比如针对通道0，当CH0SSEL=2时，被选中的外设请求是RXD2，即UART2的数据接收DMA请求被连接到DMA通道0的请求输入。

27.6.2 通道优先级

DMA 总共有 7 个外设通道，每个通道的优先级别可以通过寄存器配置为：very high,high,low,very low。当多个通道配置为相同优先级别时，通道序号越大，优先级别越低。

DMA 每搬运完一次数据都会重新进行通道请求选择，假设通道 0 传输长度为 3，通道 1 传输长度为

2。当通道 0 完成第二次传输准备进行第三次数据搬运时，通道 1 请求响应置起，这时通道控制器根据通道优先级切换至通道 1 数据搬运，直至通道 1 数据全部搬运完成，通道寄存器再切换回通道 0 完成剩下的数据搬运。

27.6.3 传输方向定义

在 DMA 通道定义规则中，_RX 表示 DMA 从外设读取数据，写入 RAM，_TX 表示 DMA 从 RAM/Flash 读取数据，写入外设。

软件在配置每个通道的外设分配之后，还需要配置 CHx_DIR 寄存器设定通道传输方向，错误的方向设置会导致 DMA 无法正常工作。

27.6.4 循环模式

外设 DMA 通道支持循环模式（Circular mode）。循环模式下，当 CHxTSIZE 寄存器定义的传输长度完成后，DMA 不会自动停止，而是返回 RAM 指针寄存器定义的起始地址，继续传输。DMA 的半程中断和全程中断还是会正常置起，DMA 不会终止传输，直到软件关闭通道。

通过置位 CHxCTRL.CIRC 寄存器使能循环模式。

存储 DMA 通道不支持循环模式。

27.7 寄存器

模块起始地址：0x4000_0400

offset 地址	名称	符号
0x00	DMA 全局控制寄存器 （DMA Global Control Register）	DMA_GCR
0x04	通道 0 控制寄存器 （Channel 0 Control Register）	DMA_CH0CR
0x08	通道 0 存储器指针寄存器 （Channel 0 Memory Address Register）	DMA_CH0MAR
0x0C	通道 1 控制寄存器 （Channel 1 Control Register）	DMA_CH1CR
0x10	通道 1 存储器指针寄存器 （Channel 1 Memory Address Register）	DMA_CH1MAR
0x14	通道 2 控制寄存器 （Channel 2 Control Register）	DMA_CH2CR
0x18	通道 2 存储器指针寄存器 （Channel 2 Memory Address Register）	DMA_CH2MAR
0x1C	通道 3 控制寄存器 （Channel 3 Control Register）	DMA_CH3CR
0x20	通道 3 存储器指针寄存器 （Channel 3 Memory Address Register）	DMA_CH3MAR
0x24	通道 4 控制寄存器 （Channel 4 Control Register）	DMA_CH4CR
0x28	通道 4 存储器指针寄存器 （Channel 4 Memory Address Register）	DMA_CH4MAR
0x2C	通道 5 控制寄存器 （Channel 5 Control Register）	DMA_CH5CR
0x30	通道 5 存储器指针寄存器 （Channel 5 Memory Address Register）	DMA_CH5MAR
0x34	通道 6 控制寄存器 （Channel 6 Control Register）	DMA_CH6CR
0x38	通道 6 存储器指针寄存器 （Channel 6 Memory Address Register）	DMA_CH6MAR
0x3C	通道 7 控制寄存器 （Channel 11 Control Register）	DMA_CH7CR
0x40	通道 7 Flash 指针寄存器 （Channel 11 Flash Address Register）	DMA_CH7FAR
0x44	通道 7 RAM 指针寄存器 （Channel 11 RAM Address Register）	DMA_CH7RAR
0x48	DMA 中断状态标志寄存器 （DMA Interrupt Status Register）	DMA_ISR

27.7.1 DMA 全局控制寄存器（DMA_GCR）

名称	DMA_GCR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

名称	DMA_GCR							
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						ADDRERR_EN	EN
位权限	U-0						R/W-0	R/W-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	**ADDRERR_EN**	DMA 错误地址中断使能 (DMA address error interrupt enable) 1：允许错误地址中断 0：禁止错误地址中断
0	**EN**	DMA 全局使能 (DMA enable) 1：DMA 使能 0：DMA 关闭

27.7.2 通道 x 控制寄存器（DMA_CHxCR）

名称	DMA_CHxCR(x=0~6)							
Offset	0x00000004 + x*0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TSIZE[15:8]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TSIZE[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		PRI		INC	SSEL		
位权限	U-0		R/W-00		R/W-0	R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SUSP	DIR	BDW		CIRC	FTIE	HTIE	EN
位权限	R/W-0	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:16	**TSIZE**	Channelx 传输长度，实际传输长度是 TSIZE+1 *注意：禁止将 TSIZE 配置为 0*
15:14	-	RFU：未实现，读为 0
13:12	**PRI**	Channelx 优先级(Channels Priority) 00：Low 01：Medium 10：High

Bit	助记符	功能描述
		11：Very High
11	**INC**	RAM 地址增减设置（Channelx Ram address Incremental) 1：RAM 地址递增 0：RAM 地址递减
10:8	**SSEL**	Channelx 外设请求映射（Channelx request Source Select) 每个通道可以接受 8 个外设请求，外设请求的映射参见通道控制章节**错误!未找到引用源。**
7	**SUSP**	通道暂停控制(Suspend) 1：暂停本通道传输，内部传输状态不复位 0：恢复通道传输 当软件置位 SUSP 寄存器时，数字电路在完成当前传输后暂停 DMA 通道工作，仅在 EN=1 的情况下有效 SUSP 清零后 DMA 通道恢复传输
6	**DIR**	通道传输方向（Direction) 0：从外设读取数据写入 RAM 1：从 RAM 读取数据写入外设
5:4	**BDW**	传输带宽设置(Bandwidth) 00：字节，8bit 01：半字，16bit 10：字，32bit 11：RFU
3	**CIRC**	循环缓冲模式（Circular mode enable) 0：关闭循环模式 1：使能循环模式
2	**FTIE**	Channelx 传输完成中断使能（Channelx Finished-Transfer Interrupt Enable) 1：使能传输完成中断 0：关闭传输完成中断
1	**HTIE**	Channelx 半程传输完成中断使能（Channelx Half-Transfer Interrupt Enable) 1：使能半程中断 0：关闭半程中断
0	**EN**	Channelx 使能（Channelx Enable) 1：启动通道 0 0：关闭通道 0

27.7.3 通道 x 存储器指针寄存器（DMA_CHxMAD）

名称	DMA_CHxMAD(x=0~6)							
Offset	0x00000008 + x*0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	MEMAD[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MEMAD[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MEMAD[15:8]							
位权限	R/W-0000 0000							

名称	DMA_CHxMAD(x=0~6)							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MEMAD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**MEMAD**	ChannelX 存储器指针地址，DMA 传输启动前软件向此寄存器写入存储器目标地址。(Channel x Memory Address pointer) 当指针指向空地址时，DMA 访问将触发 hardfault 当指针指向 Flash 时，禁止向 Flash 写入数据。 软件可以查询当前 DMA 传输的目标存储器地址。

27.7.4 通道 7 控制寄存器（DMA_CH7CR）

名称	DMA_CH7CR							
Offset	0x0000003C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-		TSIZE[13:8]					
位权限	U-0		R/W-000000					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TSIZE[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		PRI		-	DIR	RI	FI
位权限	U-0		R/W-00		U-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					FTIE	HTIE	EN
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:30	**-**	RFU：未实现，读为 0
29:16	**TSIZE**	Channel7 传输长度 (Channel 7 Transfer Size)，仅在 Flash->RAM 传输时有效，RAM->Flash 传输为固定长度 64 次传输
15:14	**-**	RFU：未实现，读为 0
13:12	**PRI**	Channel7 优先级 (Channel 7 Priority) 00：Low 01：Medium 10：High 11：Very High
11	**-**	RFU：未实现，读为 0
10	**DIR**	Channel7 传输方向 (Channel 7 Direction) 1：Flash->RAM 传输 0：RAM->Flash 传输
9	**RI**	Channel7 RAM 地址增减设置，仅在 Flash->RAM 传输中有效 (Channel 7 Ram Incremental) 1：RAM 地址递增 0：RAM 地址递减

位号	助记符	功能描述
8	**FI**	Channel7 Flash 地址增减设置，仅在 Flash->RAM 传输中有效 (Channel 7 Flash Incremental) 1：Flash 地址递增 0：Flash 地址递减
7:3	**-**	RFU：未实现，读为 0
2	**FTIE**	Channel7 传输完成中断使能 (Channel 7 Finished-Transfer Interupt Enable) 1：使能传输完成中断 0：关闭传输完成中断
1	**HTIE**	Channel7 半程传输完成中断使能 (Channel 7 Half-Transfer Interrupt Enable) 1：使能半程中断 0：关闭半程中断
0	**EN**	Channel7 使能 (Channle 7 Enable) 1：启动通道 0 0：关闭通道 0

27.7.5 通道 7 Flash 指针寄存器（DMA_CH7FLSAD）

名称	DMA_CH7FLSAD							
Offset	0x00000040							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							FLSAD[16]
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	FLSAD[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FLSAD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:17	**-**	RFU：未实现，读为 0
16:0	**FLSAD**	Channel7 Flash 指针地址（word 地址），DMA 传输启动前软件向此寄存器写入 Flash 目标地址，DMA 启动后此寄存器随 DMA 传输自增或自减 (Channel 7 Flash Address pointer) 软件可以查询当前 DMA 传输的目标 Flash 地址 此寄存器低位（bit5-0）仅在 Flash->RAM 传输中有效，RAM->Flash 传输中默认对齐 Flash 的 half-sector 起始地址

27.7.6 通道 7 RAM 指针寄存器（DMA_CH7RAMAD）

名称	DMA_CH7RAMAD
Offset	0x00000044

名称	DMA_CH7RAMAD							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RAMAD[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RAMAD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	**RAMAD**	Channel7 RAM 字指针地址，DMA 传输启动前软件向此寄存器写入 RAM 目标地址（word 地址），DMA 启动后此寄存器随 DMA 传输自增或自减 (Channel 7 RAM Address pointer) 软件可以查询当前 DMA 传输的目标 RAM 地址

27.7.7 DMA 中断状态标志寄存器（DMA_ISR）

名称	DMA_ISR							
Offset	0x00000048							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							ADDRERR
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CHFT[7:0]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CHHT[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:17	-	RFU：未实现，读为 0
16	**ADDRERR**	DMA 传输地址错误标志，硬件置位，软件写 1 清零 (DMA address error flag,write 1 to clear)
15:8	**CHFT**	DMA 通道 x 传输完成标志(DMA channel Finished-Transfer Flag,write 1 to clear)，硬件置位，软件写 1 清零 1：对应通道传输完成 0：对应通道传输未完成
7:0	**CHHT**	DMA 通道 x 传输半程标志 (DMA channel Half-Transfer

位号	助记符	功能描述
		Flag,write 1 to clear)，硬件置位，软件写 1 清零

28 循环冗余校验（CRC）

28.1 概述

循环冗余校验(Cyclic Redundancy Check，CRC)是最为常用的计算机和仪表数据通信的校验方法，FM33FG0A中CRC计算单元为完全独立模块，通过软件控制可进行7816、I2C、UART和SPI模块有串行数据流接口的收发CRC计算和校验。

此外，CRC也可进行Flash内容的完整性校验。通过结合DMA，可以实时计算Flash中程序内容的CRC结果，并生成一个完整性签名，与程序一同保存在Flash中。通过校验这个CRC签名，可以验证Flash内容是否正确、完整。

- 支持7/8/16/32位CRC，支持任意多项式
- 初值可设置
- 支持输入输出数据顺序自动调整（以字节、半字、或全字为单位）
- 支持对输出结果异或

28.2 软件配置过程

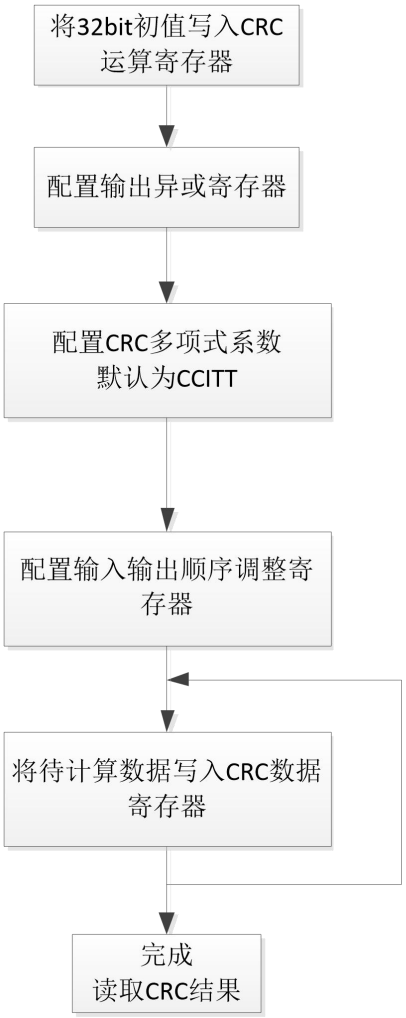


图 28-1CRC 运算流程图

CRC 配置及计算流程如下：

- CRC开始计算的时候，配置运算移位寄存器中的初始值，范围是0x0000_0000~0xFFFF_FFFF。
- 配置输出异或寄存器CRC_XOR
- 软件需配置好输入REFLECTIN处理使能；输出REFLECTOUT和XOROUT处理使能
- 软件将需要计算CRC码的数据放入数据寄存器(CRC_DR)，然后自动开始计算逐次移位。
- 计算完毕后，结果数据回写到数据寄存器，软件根据当前计算状态BUSY位来判断是否能取结果：
- 若多项式为7bit多项式则结果为CRC_DR[6:0]，若多项式为8bit多项式则结果为CRC[7:0]，若多项式为16bit多项式则结果为CRC_DR[15:0]，若多项式为32bit多项式则结果为CRC_DR[31:0]；
- 计算完前一次CRC后，数据寄存器中会保留前一次结果，作为后续数据的移位寄存器初始值。在多次连续触发CRC计算后，软件最终读取的是累积计算的完整数据的CRC值。

28.3 Golden 数据

提供 Golden 数据表格供应用中测试及校验使用。

多项式	输入序列	初始值(16 进制)		
		全 0	全 F	6363
		CRC 计算结果（16 进制）		
CRC-8	5A5A	0F	D8	C5
	1223344	F9	28	96
CRC-16	5A5A	5DD9	DDD4	9696
	11223344	7D35	7D11	4698
CRC-CCITT	5A5A	1ACB	07C4	1877
	11223344	DD33	59F3	DD06

28.4 DMA 接口

CRC与DMA之间通道为单向的（RAM->CRC）。CRC模块可以通过DMA模块读取并校验RAM数据，其工作流程如图所示。CRC向DMA发起请求，DMA接收请求后，读取RAM并将数据写入CRC模块的CRCDR寄存器中。CRC模块接收到数据后，撤销DMA请求并开始计算校验值，校验完成后，CRC模块重新置起DMA请求。

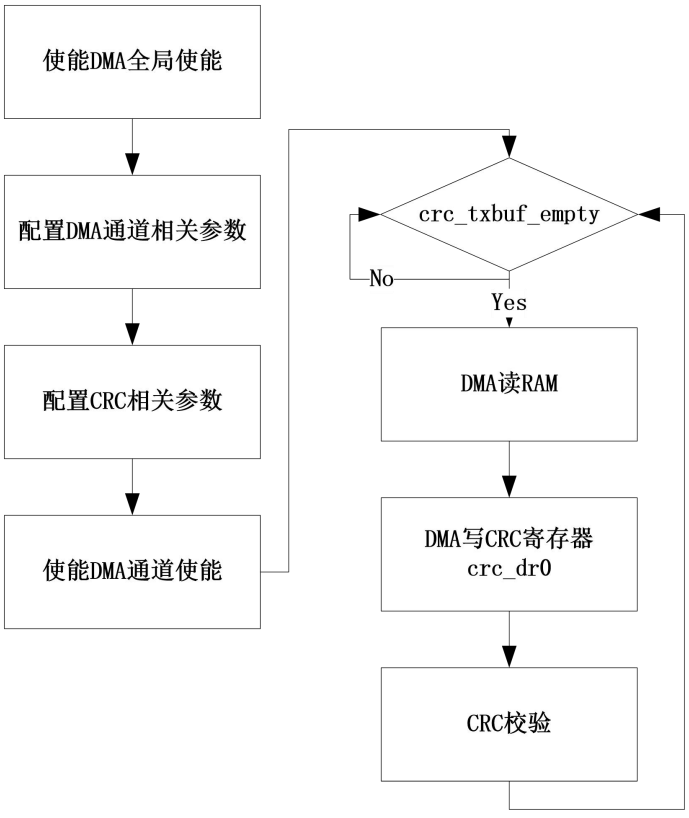


图 28-2 使用 DMA 对 RAM 中的数据进行 CRC 运算

28.5 Flash 数据完整性校验

为提升系统主频，降低critical path delay，删除这个局部总线功能。

如果需要CRC进行Flash内容的完整性校验，解决方案是使用DMA，将存储器指针直接指向flash，DMA可以直接将DMA数据搬运到CRC模块进行校验。

28.6 寄存器

模块起始地址：0x4001_0000

offset 地址	名称	符号
0x00000000	CRC 数据寄存器 （CRC Data Register）	CRC_DR
0x00000004	CRC 控制状态寄存器 （CRC Control Register）	CRC_CR
0x00000008	CRC LFSR 寄存器 （CRC Linear Feedback Shift Register）	CRC_LFSR
0x0000000C	CRC 输出异或寄存器 （CRC output XOR Register）	CRC_XOR
0x0000001C	CRC 多项式寄存器 （CRC Polynominal Register）	CRC_POLY

28.6.1 CRC 数据寄存器（CRC_DR）

名称	CRC_DR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DR[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DR[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	**DR**	CRCDR 用于作为数据输入寄存器，并且在运算结束后保存 CRC 计算结果。(CRC Data Register) 作为输入时：若 word 操作使能，则对 CRCDR[31:0]进行计算，共 4 次 byte 运算（由低到高）；否则对 CRCDR[7:0]进行计算，共 1 次 byte 运算。 保存结果时：若为 7 位多项式结果保存在 CRCDR[6:0]，若为 8 位多项式结果保存在 CRCDR[7:0]，若为 16 位多项式结果保存在 CRCDR[15:0]，若为 32 位多项式结果保存在 CRCDR[31:0]。

28.6.2 CRC 控制状态寄存器（CRC_CR）

名称	CRC_CR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						OPWD	PARL
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFLTIN		RFLTO	RES	BUSY	XOR	SEL	
位权限	R/W-00		R/W-0	R-0	R-0	R/W-0	R/W-01	

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9	**OPWD**	WORD 操作使能 (Operation by Word) 0：字节操作，CRC 计算仅针对 CRCDR 最低字节进行 1：字操作，CRC 计算针对 CRCDR 全部 4 字节进行
8	**PARL**	并行操作使能 0：并行操作使能关闭 1：并行操作使能打开
7:6	**RFLTIN**	CRC 输入反转控制 (Reflected Input) 00：输入不反转 01：输入按字节反转 10：输入按半字反转 11：输入按字反转 例如：计算数据为 0x11223344， 如果 RFLTIN==01，则将数据变为 0x8844CC22，再进行计算 如果 RFLTIN==10，则将数据变为 0x448822CC，再进行计算 如果 RFLTIN==11，则将数据变为 0x22CC4488，再进行计算
5	**RFLTO**	CRC 输出反转控制 (Reflected Output) 0：输入不反转 1：输入按字节反转 例如： 如果 RFLTO==1，若当前计算的 CRC 结果为 0x1234，则输出的结果为 0x2C48 如果 RFLTO==0，则直接输出 0x1234 注意：此结果不一定为最终输出结果，还需要看 XOR 是否为 1，详见本寄存器 bit2 说明
4	**RES**	CRC 结果标志位，只读 (Result Flag) 0：CRC 结果为 0 1：CRC 结果非全 0
3	**BUSY**	CRC 运算标志位，只读 (Busy) 0：CRC 运算结束 1：CRC 运算进行中
2	**XOR**	输出异或使能 (Output XORed with CRC_XOR register enable) 0：输出不异或 CRC_XOR 寄存器 1：输出异或 CRC_XOR 寄存器
1:0	**SEL**	CRC 多项式位宽选择 (Polynomial width Selection) 00：32 位 01：16 位

位号	助记符	功能描述
		10：8 位 11：7 位

28.6.3 CRC LFSR 寄存器（CRC_LFSR）

名称	CRC_LFSR							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LFSR[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LFSR[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LFSR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LFSR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	**LFSR**	CRC 线性反馈移位寄存器（Linear Feedback Shift Register） 运算开始前可以由软件写入 CRC 初始值

28.6.4 CRC 输出异或寄存器（CRC_XOR）

名称	CRC_XOR							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	XOR[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	XOR[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	XOR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	XOR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**XOR**	CRC 运算结果异或寄存器（eXclusive OR） 当 CRC_CR.XOR 为 1 时，CRC 结果输出前将异或此寄存器的数据。

28.6.5 CRC 多项式寄存器（CRC_POLY）

名称	CRC_POLY							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	POLY[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	POLY[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	POLY[15:8]							
位权限	R/W-0001 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	POLY[7:0]							
位权限	R/W-0010 0001							

位号	助记符	功能描述
31:0	**POLY**	CRC 运算多项式系数 (CRC Polynominals)

29 高级定时器（ATIM）

29.1 概述

FM33FG0A包含一个高级定时器。

高级定时器包含一个16bit自动重载计数器及一个可编程预分频器。

高级定时器可以支持多种应用，包括如捕捉、输出比较、PWM、带死区插入的互补PWM。

29.2 主要特性

- 16bit向上、向下、双向自动重载计数器
- 16bit可编程预分频器，支持实时调整计数时钟分频
- 4个独立通道可用于输入捕捉、输出比较、PWM、单脉冲输出
- 可编程死区插入的互补输出
- 重复计数器，支持定时器多个循环后更新状态
- 两路刹车引脚输入、比较器刹车、SVD刹车，刹车信号滤波和极性选择，刹车信号组合配置
- 支持在以下事件发生时产生中断或DMA事件
  - 计数器上/下溢出，计数器初始化（软件或硬件 trigger）
  - Trigger 事件（计数器启动、停止、初始化、内外部触发）
  - 输入捕捉
  - 输出比较
  - 刹车输入
- 支持增量正交编码器和霍尔传感器

术语：

- OCxREF信号为高称为有效电平（active），为低称为无效电平（inactive）
- 空闲模式（IDLE mode）：相对运行模式而言，指发生刹车事件时MOE=0
- 输出禁止（output disables）：GPIO输出使能关闭，不受TIMER驱动
- Off-state：GPIO输出使能打开，但是TIMER输出无效状态
- 无效状态（inactive state）：互补通道中的一路或两路输出无效电平时的状态

29.3 结构框图

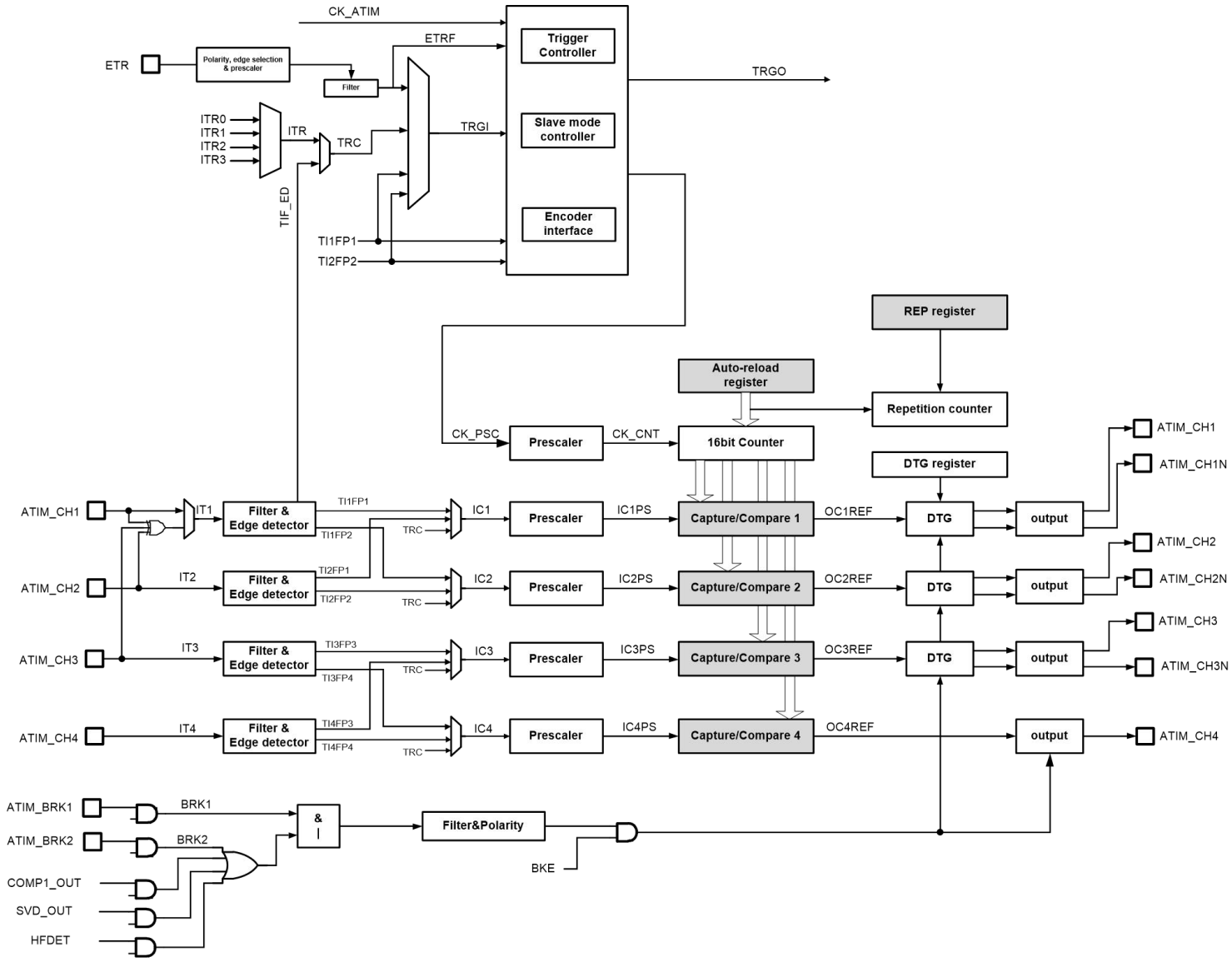


图 29-1 高级定时器结构框图

29.4 功能描述

29.4.1 定时单元

高级定时器的定时单元由一个16位计数器和自动重载寄存器组成。计数器可以向上、向下或双向计数。计数时钟可以通过16位预分频器对APBCLK进行分频后得到。

计数器、自动重载寄存器预分频寄存器都可以由软件改写或读取，即使在计数器正在运行时也是如此。

定时单元包含如下寄存器：

- 计数器（ATIM_CNT）
- 预分频寄存器（ATIM_PSC）
- 自动重载寄存器（ATIM_ARR）
- 重复计数寄存器（ATIM_RCR）

ARR包含预装载功能，该功能通过ARPE（Auto Reload Preload Enable）寄存器控制。当ARPE=0时，对ARR寄存器执行写入，写入数据将直接传入到影子寄存器；当ARPE=1时，对ARR寄存器执行写入的数据在update event（ATIM_CNT上溢出或者下溢出）发生时，传送到影子寄存器。软件也可以通过寄存器操作主动触发ARR更新（UEV）。

ATIM_CNT工作时钟由ATIM_PSC产生的分频时钟驱动，只有在计数器使能寄存器（CEN）置位时，CNT才开始计数。当CNT=ARR时，本轮计数结束，发送update event。

ATIM_PSC是一个同步预分频器，能够对APBCLK进行1~65536分频。PSC寄存器同样被缓存，改写PSC实际不改写影子寄存器，只有当新的update event到来时，才会从PSC更新至影子寄存器。因此在CNT计数过程中，软件可以实时改写PSC，而新的预分频比将在下一更新事件发生时被采用。

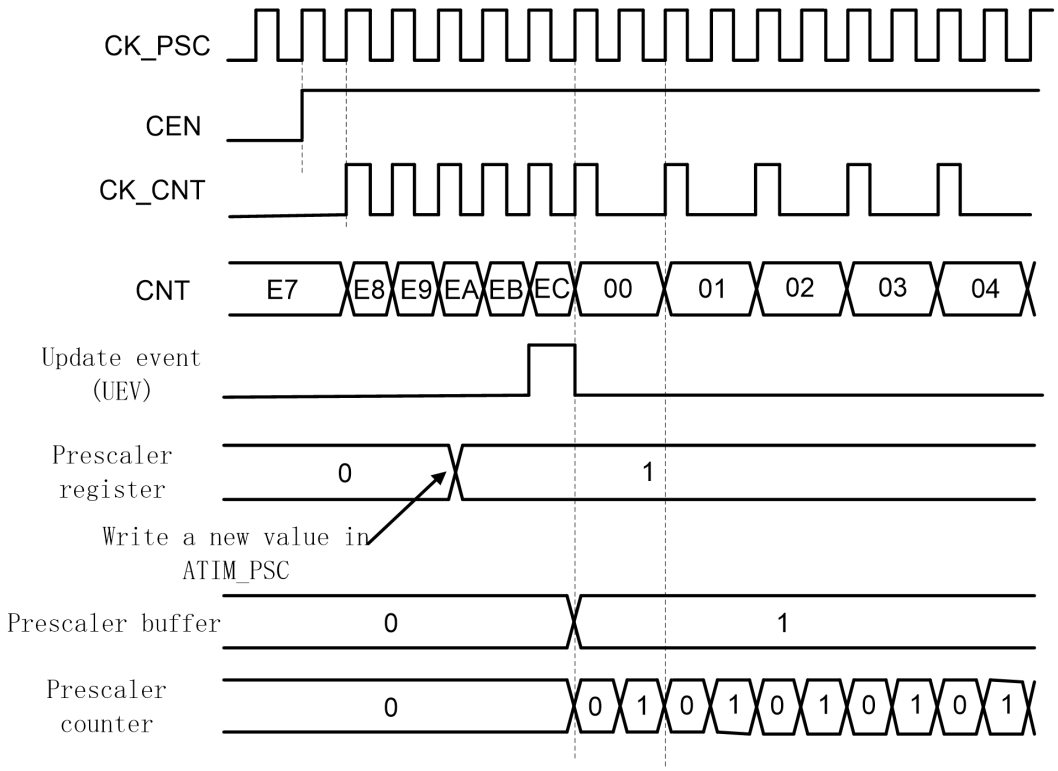


图 29-2 预分频从 1 变为 2 的波形

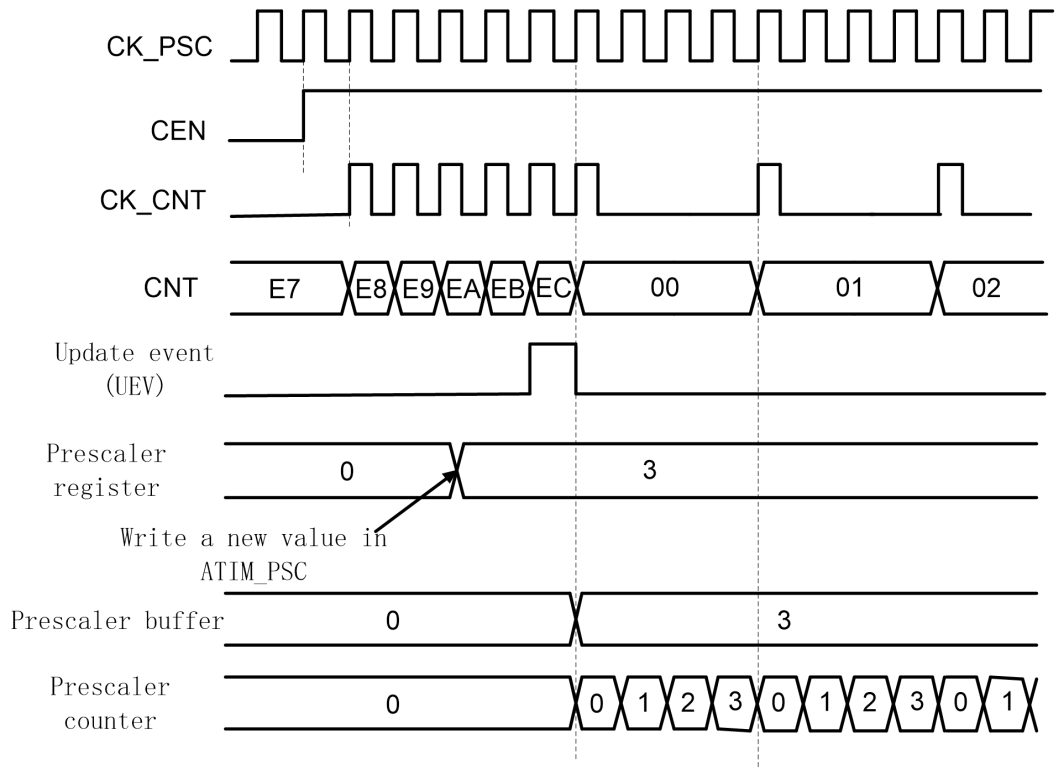


图 29-3 预分频从 1 变为 4 的波形

29.4.2 定时器工作模式

定时器支持向上计数、向下计数和中心计数模式。

向上计数

此模式中，计数器使能后从0开始计数，直到CNT=ARR，产生溢出事件，然后重新从0开始计数。

如果使能了重复计数功能，则计数器按照RCR的定义重复上述过程若干次（RCR+1），才会产生溢出事件。

软件可以通过设置UG寄存器直接触发update event，此时CNT和预分频计数器自动清零。设置UG寄存器是否触发UIF（Update Interrupt Flag）中断标志置位由URS寄存器的设置决定。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- RCR影子寄存器被更新为ATIM_RCR内容
- ARR影子寄存器被更新为ATIM_ARR内容
- PSC影子寄存器被更新为ATIM_PSC内容

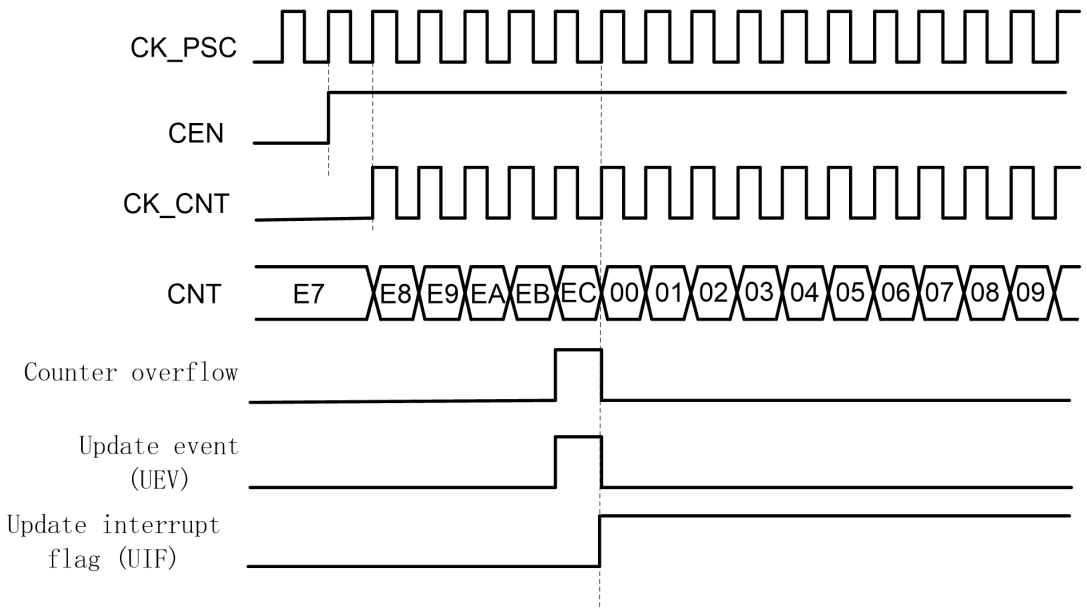


图 29-4 向上计数波形，内部时钟不分频

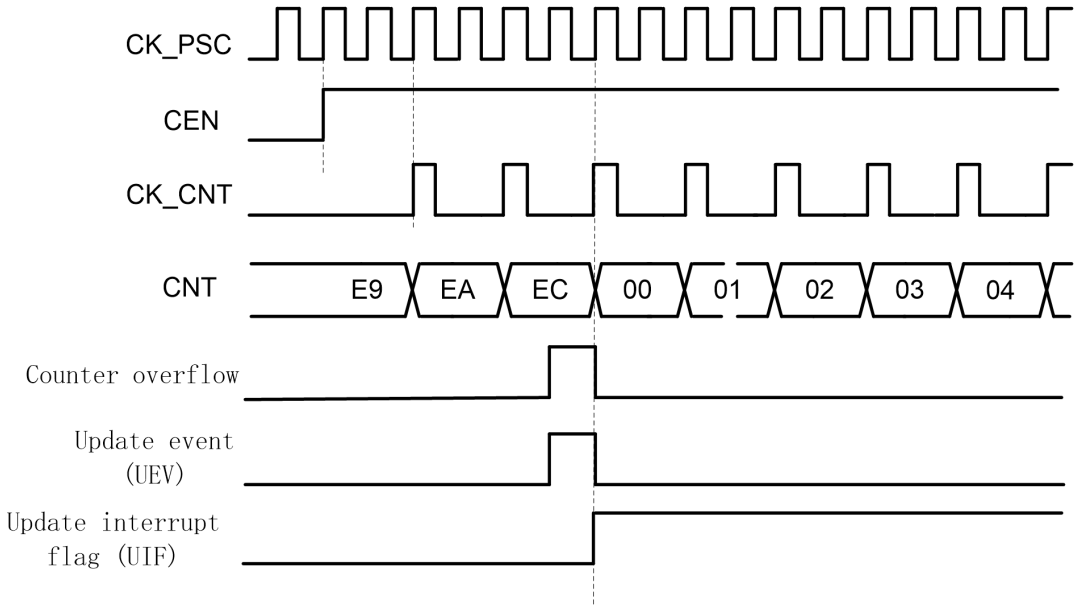


图 29-5 向上计数波形，内部时钟 2 分频

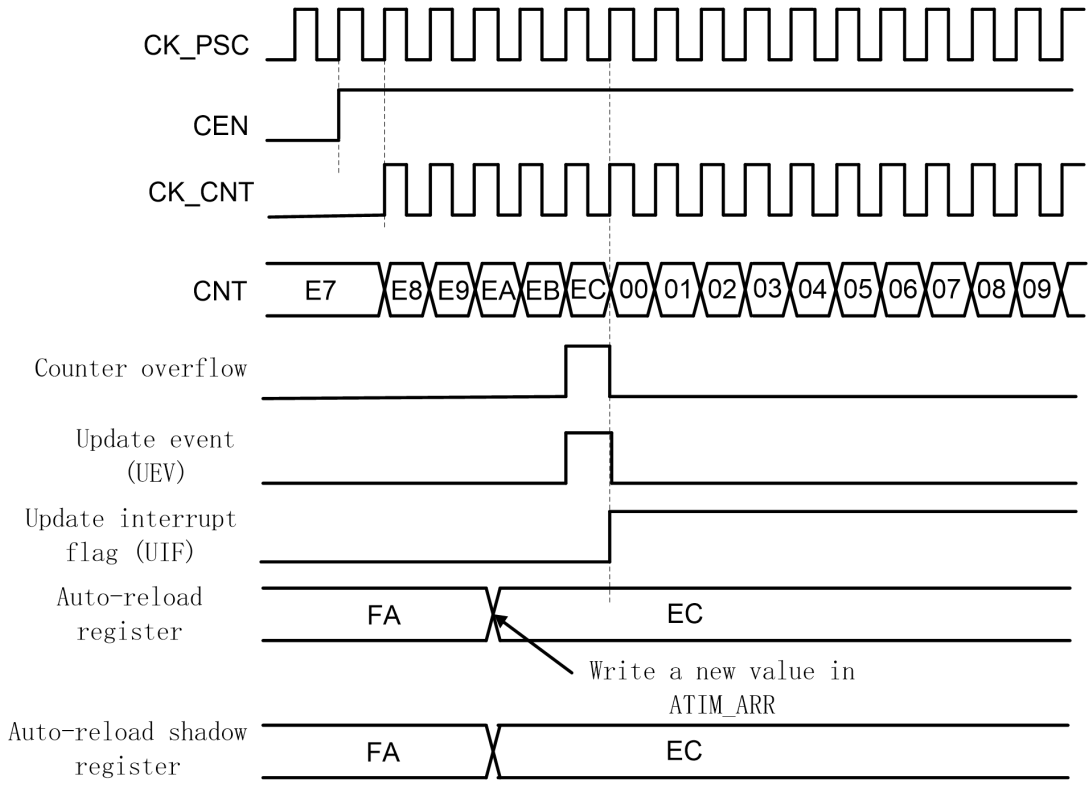


图 29-6ARPE=0（ATIM_ARR 没有预装载）时的更新事件

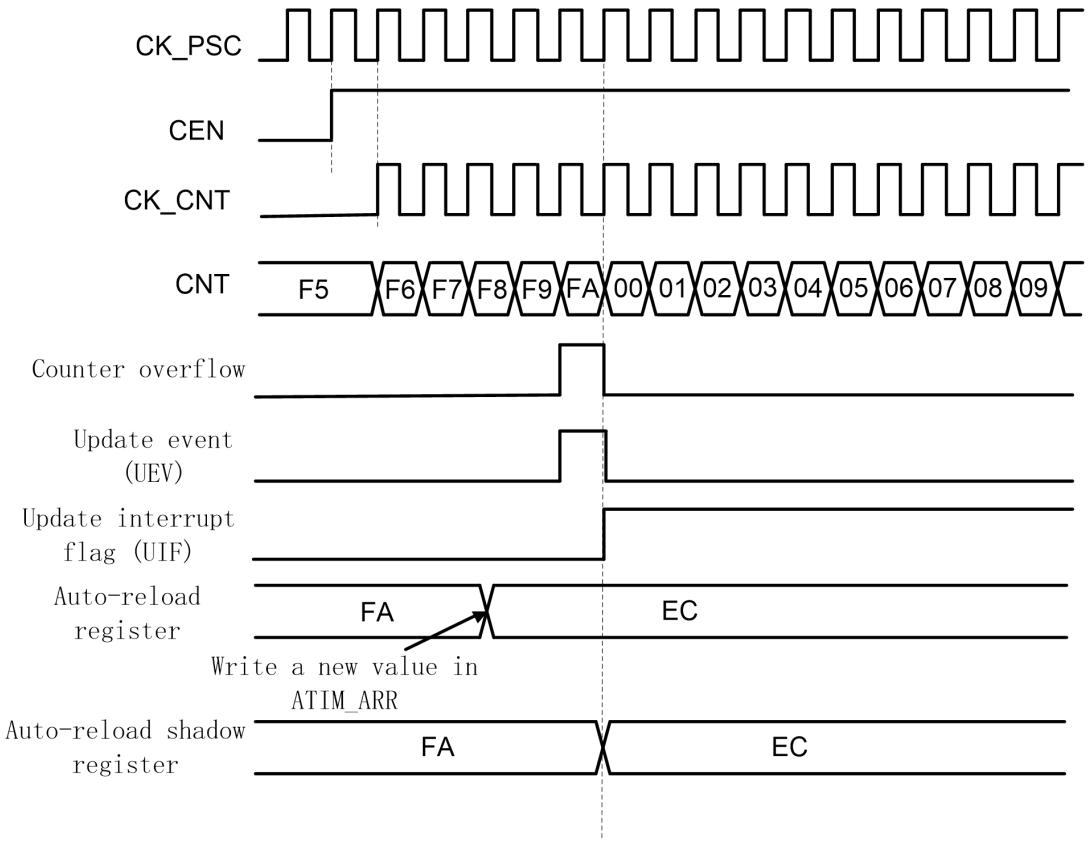


图 29-7ARPE=1（ATIM_ARR 预装载）时的更新事件

向下计数

向下计数模式中，计数器从ARR值开始递减，到0后产生下溢出事件，并且重新从ARR开始计数。

如果使能了重复计数功能，则计数器按照RCR的定义重复上述过程若干次（RCR+1），才会产生溢出事件。

软件可以通过设置UG寄存器直接触发update event，此时CNT和预分频计数器自动清零。设置UG寄存器是否触发UIF（Update Interrupt Flag）中断标志置位由URS寄存器的设置决定。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- RCR影子寄存器被更新为ATIM_RCR内容
- ARR影子寄存器被更新为ATIM_ARR内容
- PSC影子寄存器被更新为ATIM_PSC内容

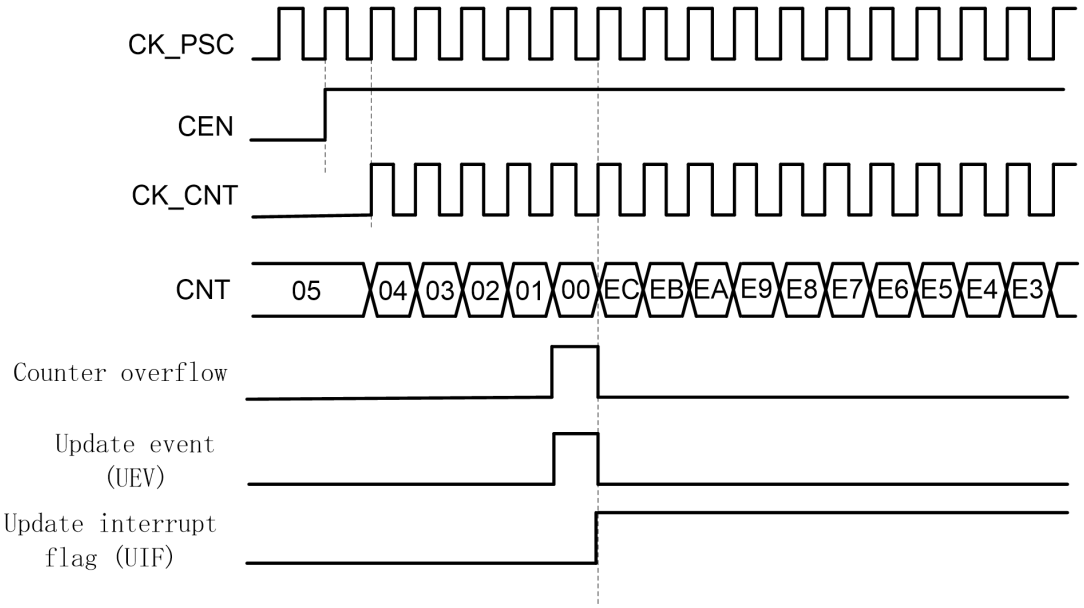


图 29-8 向下计数，内部时钟不分频

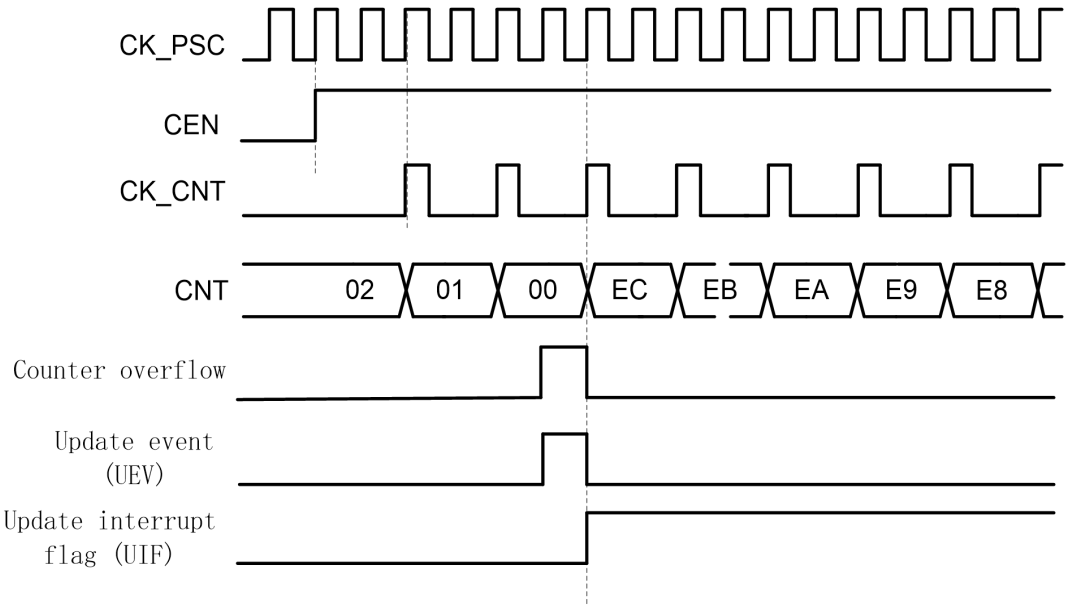


图 29-9 向下计数，内部时钟 2 分频

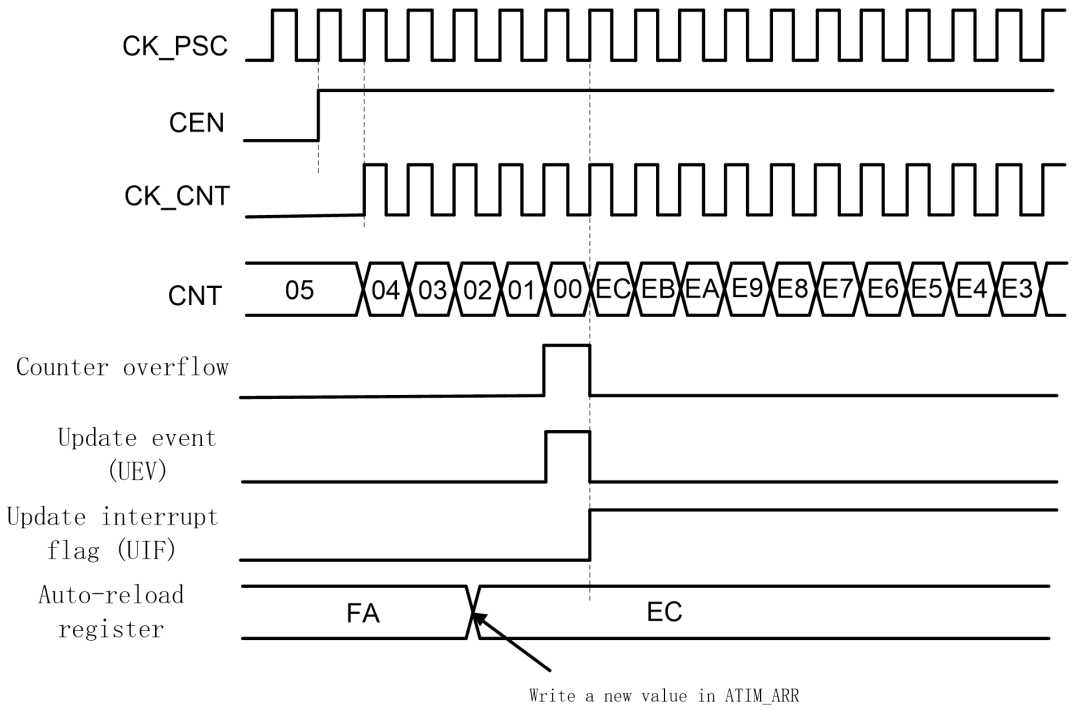


图 29-10 向下计数，内部时钟 2 分频

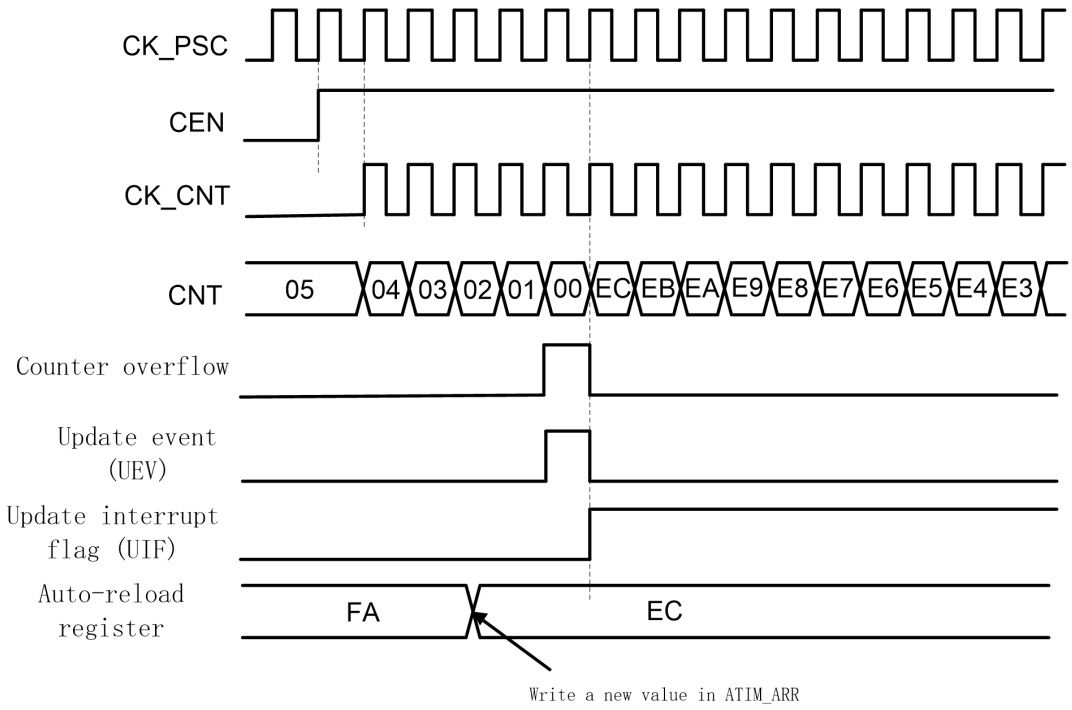


图 29-11 向下计数，不使用重复计数时的更新事件

中心对齐计数

在中心对齐模式下，计数器从0开始向上计数，到ARR-1产生上溢出事件，然后从ARR开始向下计数到1，产生下溢出事件，再从0重新开始向上计数。

CMS[1:0]寄存器用于使能中心对齐模式，并选择中心对齐模式下的输出比较工作方式。当CMS!=00时为中心对齐计数，当CMS=01时，输出比较功能仅在向下计数时有效，当CMS=10时，输出比较功能仅在向上计数时有效，当CMS=11时，输出比较功能在上下计数时都有效。

中心对齐模式下，DIR寄存器无法由软件改写，而是随着计数方向变化硬件自动更新，表示当前计数方向。

计数器在overflow和underflow的事件上都会更新 ARR、PSC和RCR的影子寄存器。

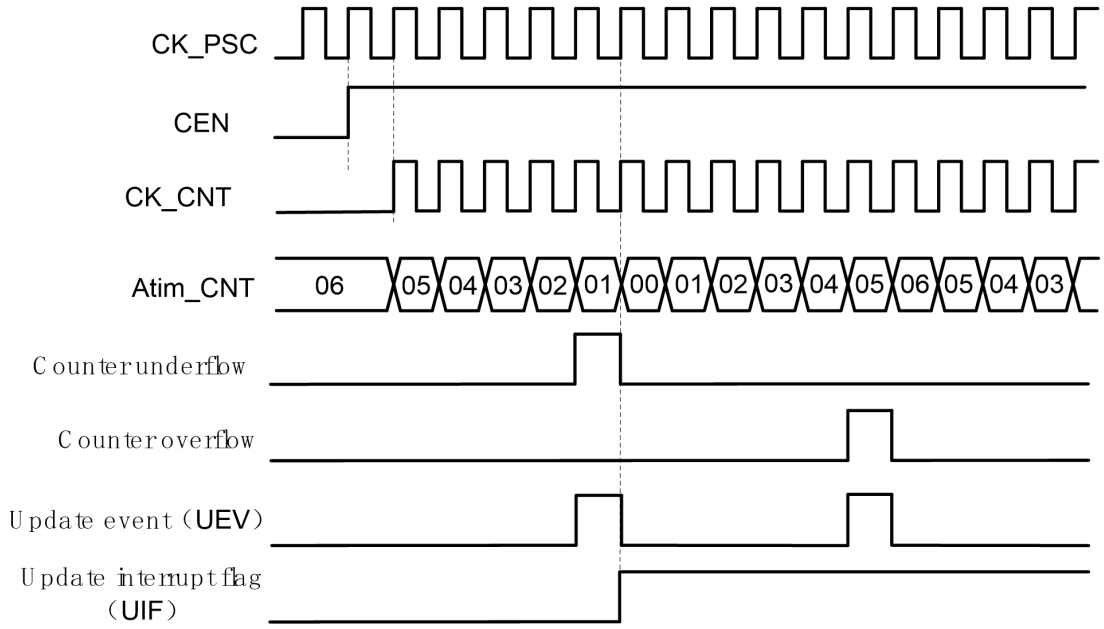


图 29-12 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6

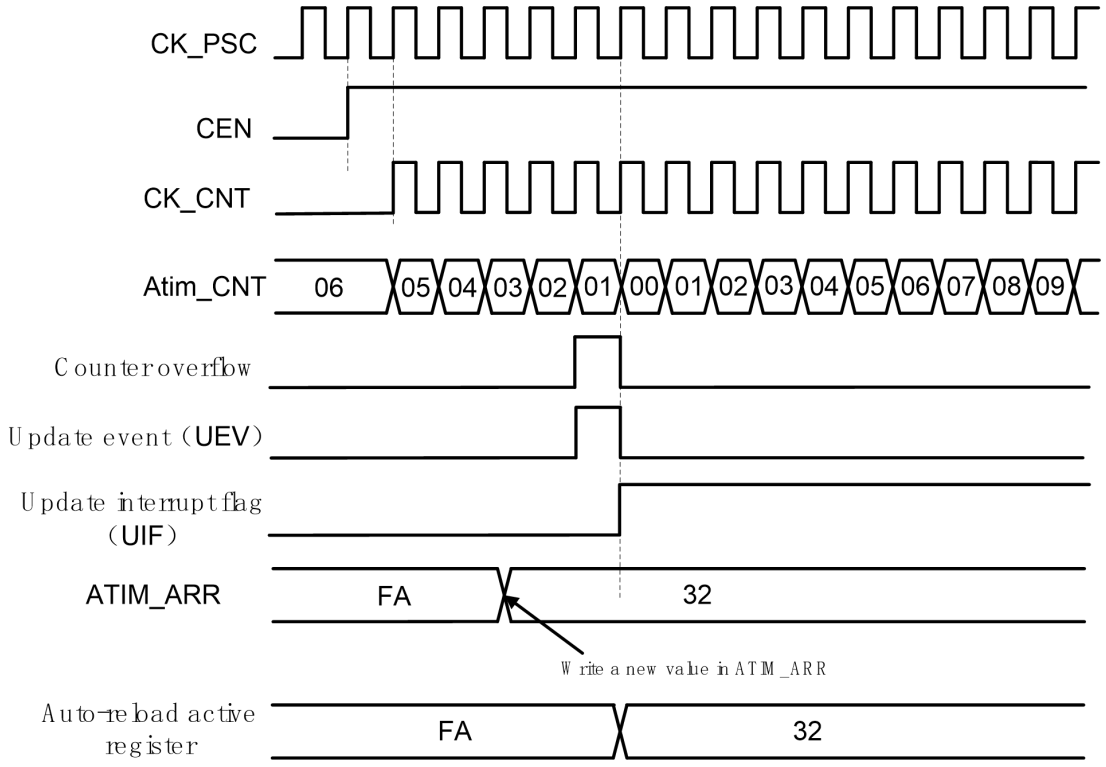


图 29-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

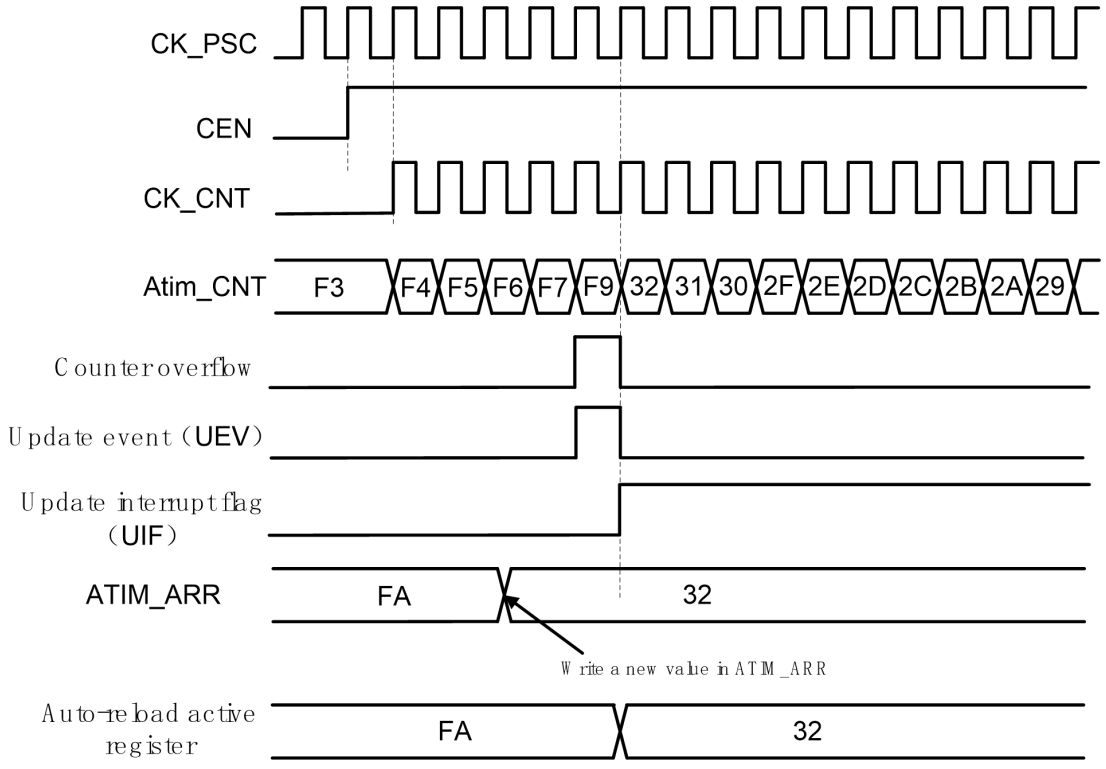


图 29-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

29.4.3 重复计数器

Update event在计数器overflow或underflow，并且重复计数器为0 的情况下产生。这意味着ARR、PSC、CCR（比较/捕捉寄存器，输出比较模式下）的preload寄存器会在N+1次overflow或underflow之后，才将数据传输给影子寄存器，其中N是RCR寄存器值。

重复计数器在以下情况下递减：

- 向上计数模式下发生上溢出
- 向下计数模式下发生下溢出
- 中心计数模式下每次上溢出或者下溢出

注意，当update event由软件或slave mode controller触发时，更新事件会立即发生，而不管当前RCR是什么值，同时重复计数器也会被立即更新为RCR的值。

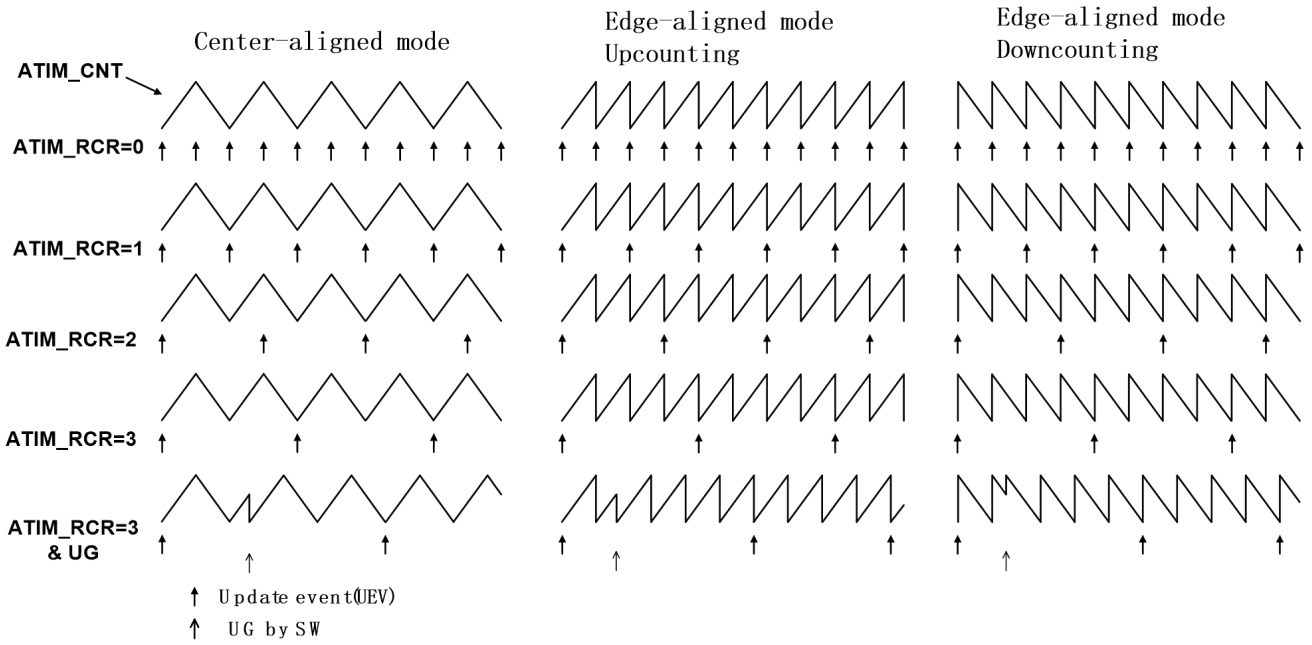


图 29-15 不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置

29.4.4 Preload 寄存器

以下功能寄存器支持preload功能：

- 自动重载寄存器ARR
- 重复计数寄存器RCR
- 预分频寄存器PSC（不可关闭preload功能）
- 通道控制寄存器CCR
- CcxE和CcxNE控制寄存器
- OcxM控制寄存器

以上寄存器，除了PSC之外，都可以由软件选择使能或者禁止preload功能。

具备preload功能的寄存器，包含两组物理实体：

- Shadow register（影子寄存器）：实际定时器正在使用的寄存器
- Preload register（预装载寄存器）：软件可以访问的寄存器

当禁止preload时，具备preload功能的寄存器特性如下：

- Preload寄存器可以实时由软件访问、改写
- Shadow寄存器与Preload寄存器同步更新

如果使能了preload，则：

- 所有软件操作访问的是preload寄存器
- 当update event发生时，所有preload寄存器内容将同步被转移到对应的shadow寄存器

29.4.5 计数器工作时钟

计数器可以使用如下时钟工作：

- APBCLK——内部时钟模式
- 外部引脚输入时钟（Tix）——外部时钟模式1
- 外部引脚触发输入（ETR）——外部时钟模式2
- 内部触发（ITRx）——使用一个timer的触发输出（TGO）作为计数时钟

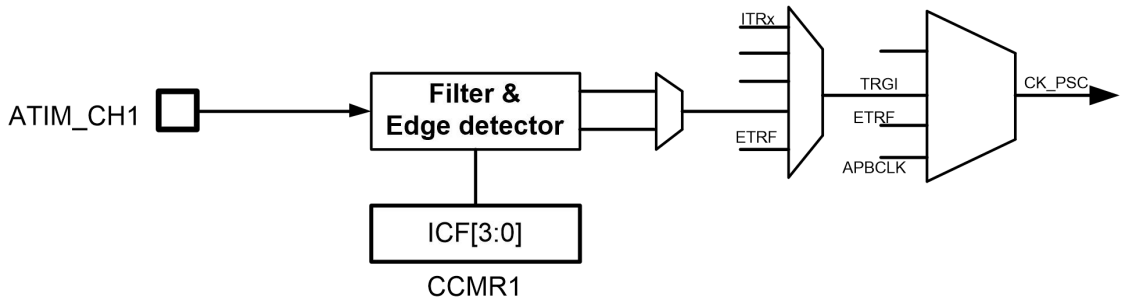


图 29-16 ATIM 时钟源框图

29.4.5.1 内部时钟模式

内部时钟模式下，禁止从机模式（SMS=000），CEN、DIR、UG等寄存器位都是软件控制

软件操作UG寄存器后，update信号经过CLK_PSC同步后，计数器值将被重新初始化。

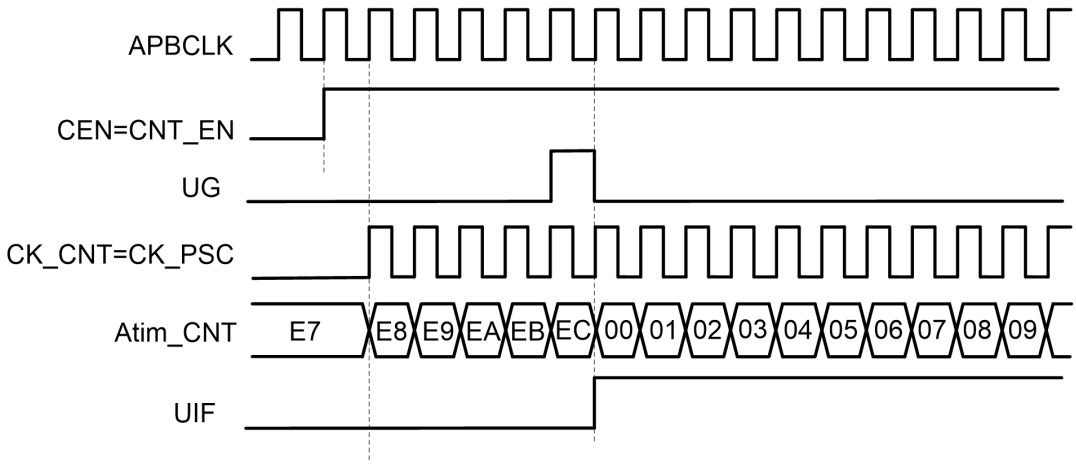


图 29-17 内部时钟源模式，时钟分频因子为 1

29.4.5.2 外部时钟模式 1

此模式下直接使用外部引脚输入信号作为计数时钟，配置SMS=111，计数边沿可以配置为上升或下降沿。

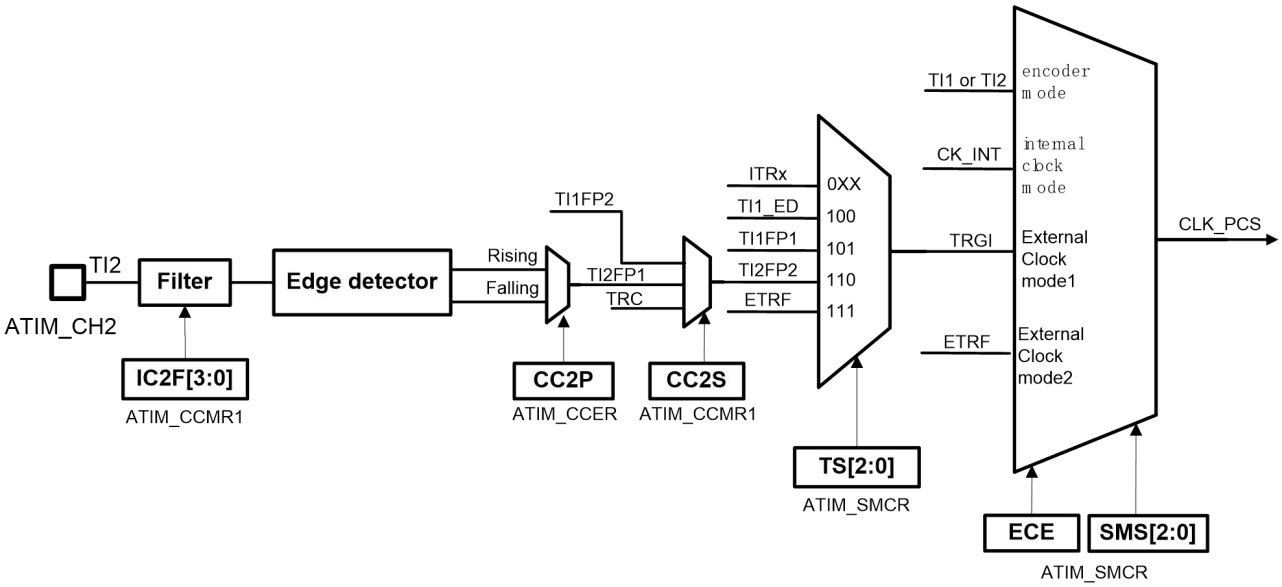


图 29-18TI2 外部时钟连接例子

外部输入信号在触发计数器计数前，会先经过内部时钟的同步过程，同时输入信号的有效沿会触发TIF标志

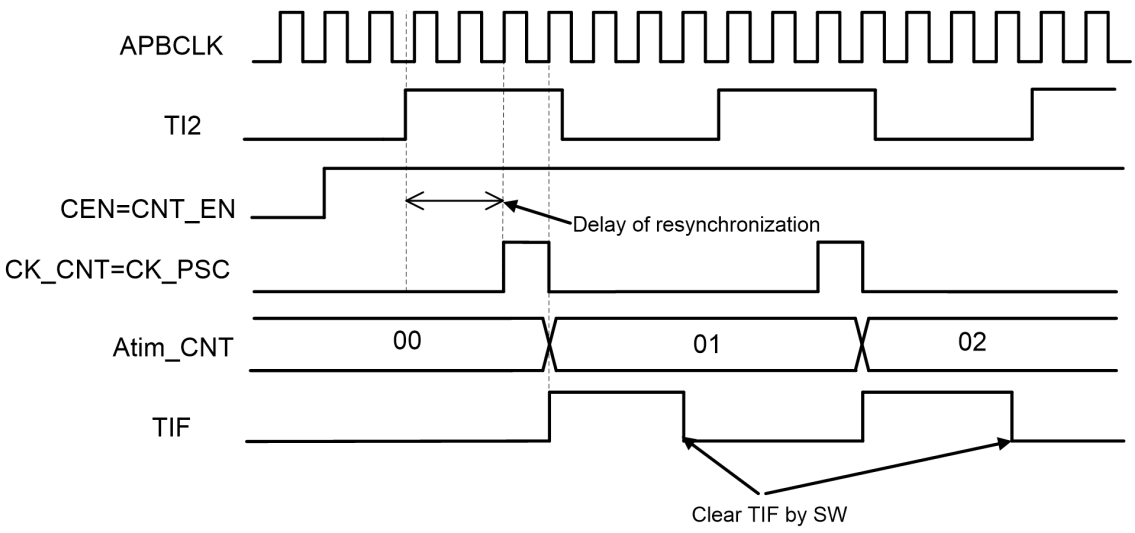


图 29-19 外部时钟模式 1 下的时序

使用外部时钟计数时，仍然要使能ATIM的内部时钟（APBCLK），因为ATIM要使用APB_CLK来对

外部输入时钟进行同步和滤波。在外部时钟模式1下，外部输入时钟首先经过滤波和边沿选择，得到有效的计数沿，作为有效工作时钟（CLK_PSC）输入给预分频模块。

外部时钟同步采用简单的2级触发器结构，因此为了避免亚稳态，要求外部输入时钟宽度至少大于2个APB_CLK周期。

此模式下只有通道1和2的输入可以用做时钟输入，所需配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH2功能
- 关闭通道使能，配置ATIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC2S=01, IC2映射到TI2
- 选择计数有效沿，配置ATIM_CCER.CC2P=0，选择上沿或者下沿
- 配置输入滤波时间，配置ATIM_CCMR1.IC2F[3:0](IC2F=0000，不进行输入滤波)
- 使能外部时钟模式1，配置ATIM_SMCR.SMCR=111
- 选择触发输入源，配置ATIM_SMCR.TS=110,选定TI2作为触发输入源
- 打开通道使能，配置ATIM_CCER.CC2E=1
- 使能计数器，配置ATIM_CR1.CEN=1

下图是一个典型的外部时钟计数模式1的示例：

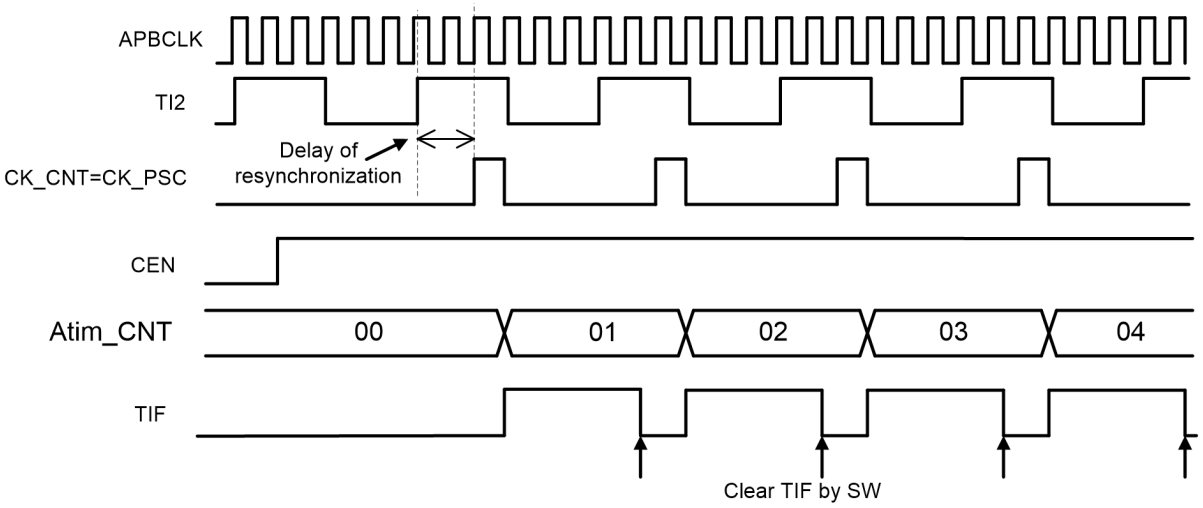


图 29- 20 外部时钟模式 1 下的时序

29.4.5.3 外部时钟模式 2

此模式下使用ATIM_ETR管脚输入信号的上升沿或下降沿（不支持双沿）来计数。

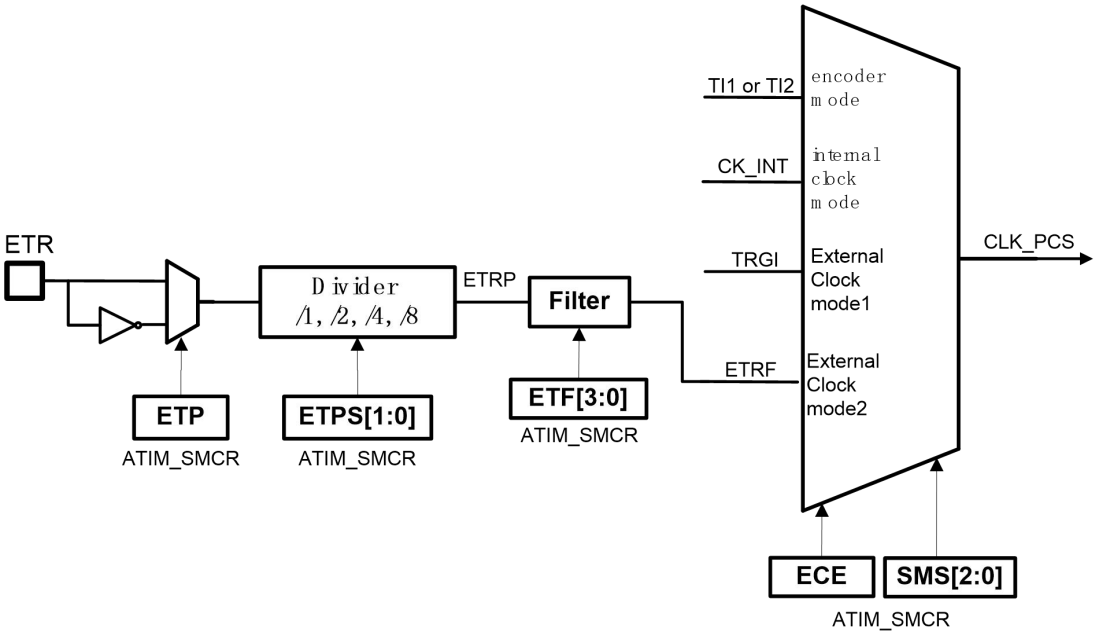


图 29-21 外部触发输入框图

下图是使用ETR二分频后的上升沿进行计数，其中实际计数发生时间因为内部时钟的同步过程而延迟于ETR输入上升沿。

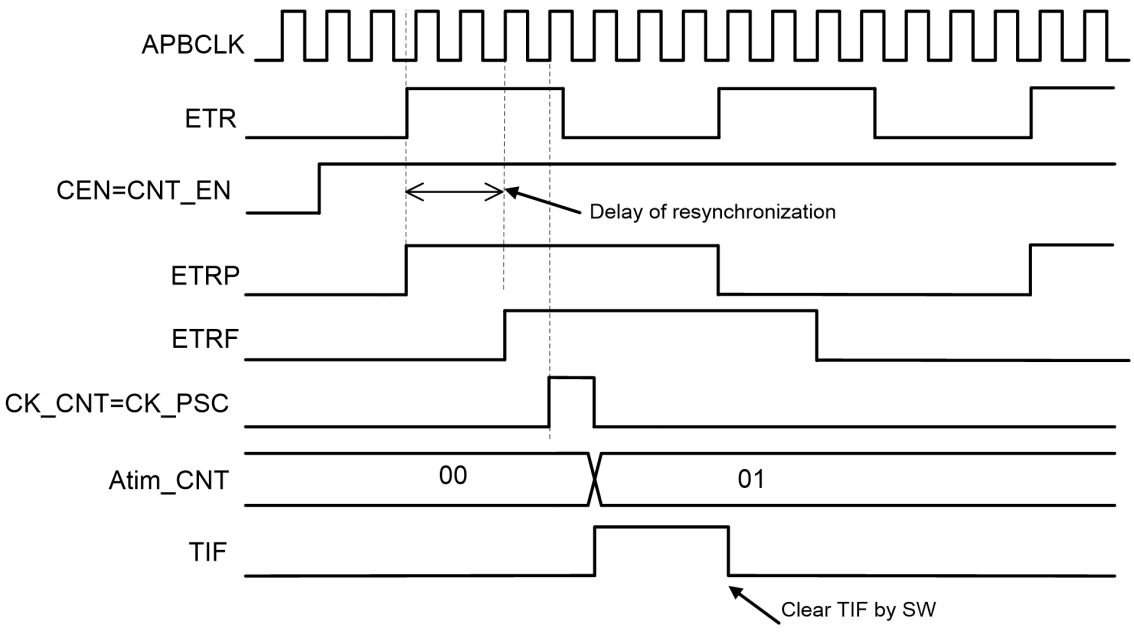


图 29-22 外部时钟模式 2 下的时序 1

与外部时钟模式1的主要差别是，ETR输入直接被分频后再进行滤波，产生CK_PSC时钟，这意味着可以支持ETR输入频率高于APB_CLK的应用场景，这种情况下，需要首先对ETR输入进行预分频，再用于驱动计数器。

此模式所需配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_ETR功能
- 设置ETP进行沿选择，ATIM_SMCR.ETP=0
- 设置ETR分频比，配置ATIM_SMCR.ETPS[1:0]=01
- 配置输入滤波时间，ATIM_SMCR.ETF[3:0]=0000
- 置位ECE寄存器，使能外部时钟模式2,ATIM_SMCR.ECE=1，ATIM_SMCR.SMS=000
- 使能计数器，配置ATIM_CR1.CEN=1

下图是一个典型的外部时钟模式2的示例：

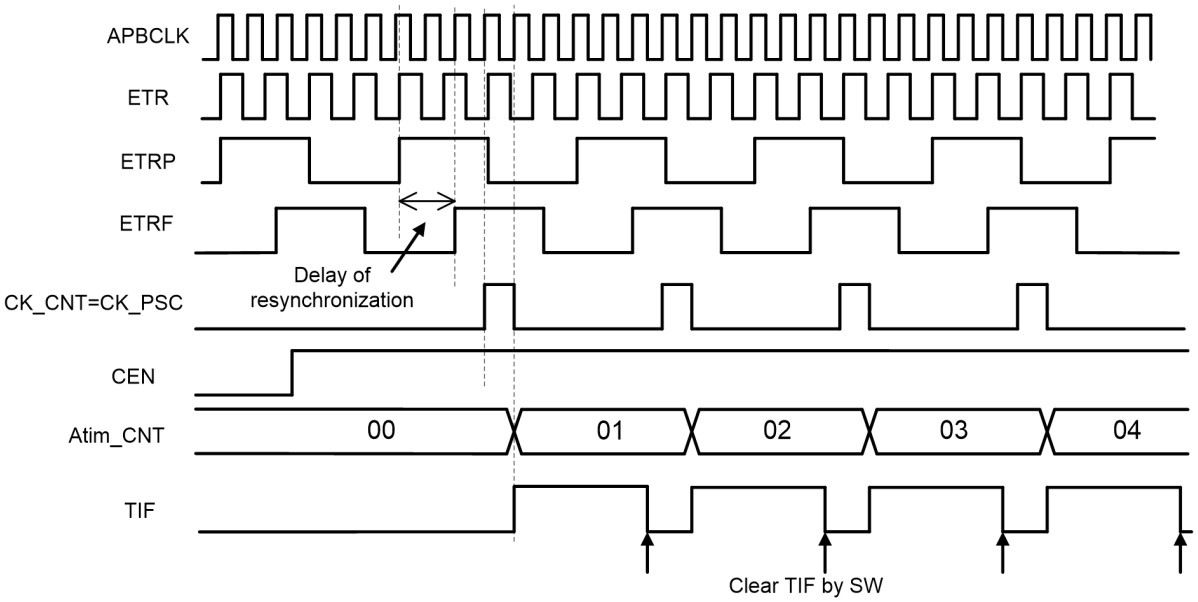


图 29-23 外部时钟模式 2 下的时序 2

在使用外部时钟模式2时，仍可以将ATIM配置为slave模式：比如使用ETR输入计数，同时使用另一个Timer的TRGO作为触发信号，当触发事件到来时，复位计数器重新开始计数。

29.4.6 内部触发信号（ITRx）

ATIM支持4个ITR输入，可用于计数触发或者内部信号捕捉。当用于内部信号捕捉时，需要将TS配置为000~011用于选择ITR0~ITR3，并将CCxS配置为11，即将TRC选为捕捉信号。

每个ITR输入支持4个内部信号扩展，由ITRxSEL寄存器配置。输入信号源参考下表：

Slave	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
ATIM	TAU10_TRGO	TAU11_TRGO	COMP1	COMP2

29.4.7 捕捉/比较通道

ATIM包含4个捕捉/比较通道，每个通道由一个捕捉比较寄存器（CCR）（包含影子寄存器）、一个捕捉输入级、一个比较输出级组成。

输入级电路会采样Tix输入并产生滤波后的信号TixF，然后边沿检测和极性选择产生对应的TixFPx信号，此信号可作为计数触发或者待捕捉信号，并且在被捕捉前经过预分频。

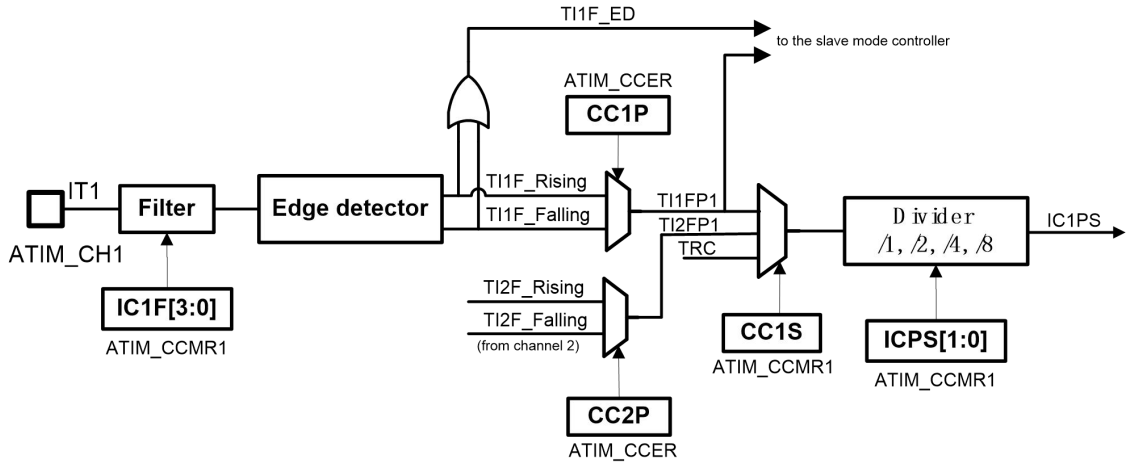


图 29-24 捕获/比较通道(通道 1 输入部分)

输出级电路会产生一个输出基准信号OCxREF，此信号固定为高电平有效，作为最终输出电路的参考输入。其中通道1~3支持互补输出和死区插入，通道4则比较简单，不支持互补输出。

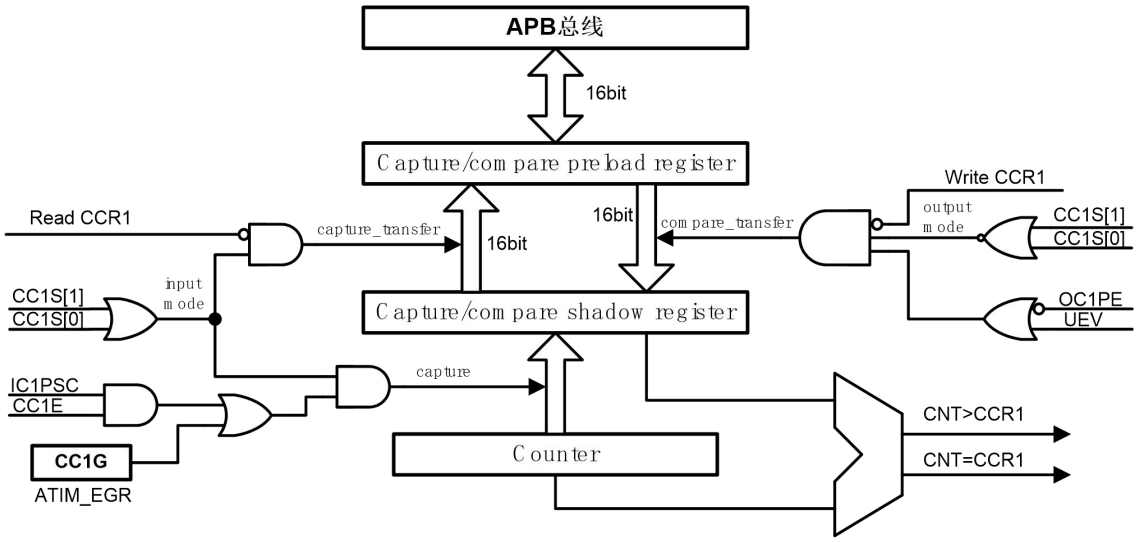


图 29-25 捕获/比较通道 1 的主电路

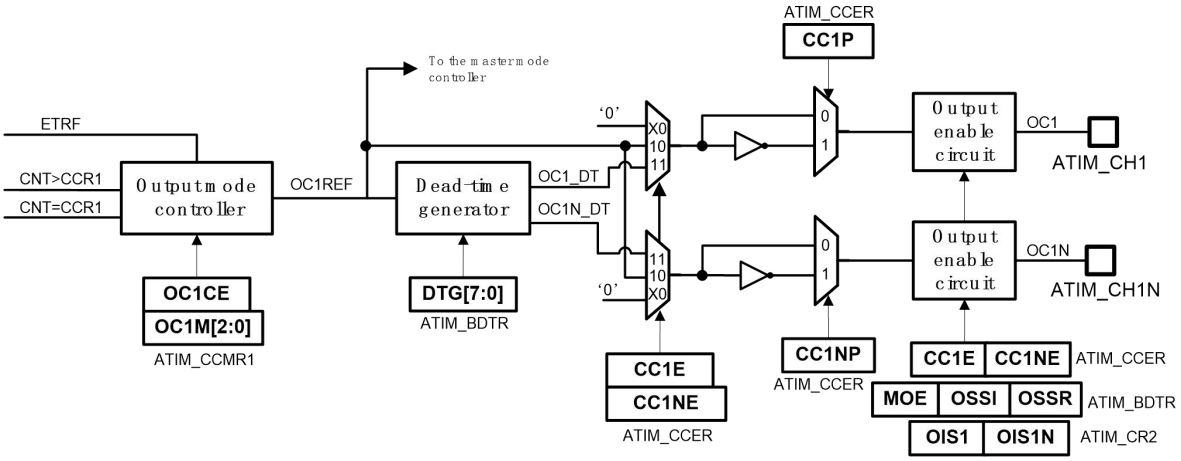


图 29-26 捕获/比较通道的输出部分(通道 1 至 3)

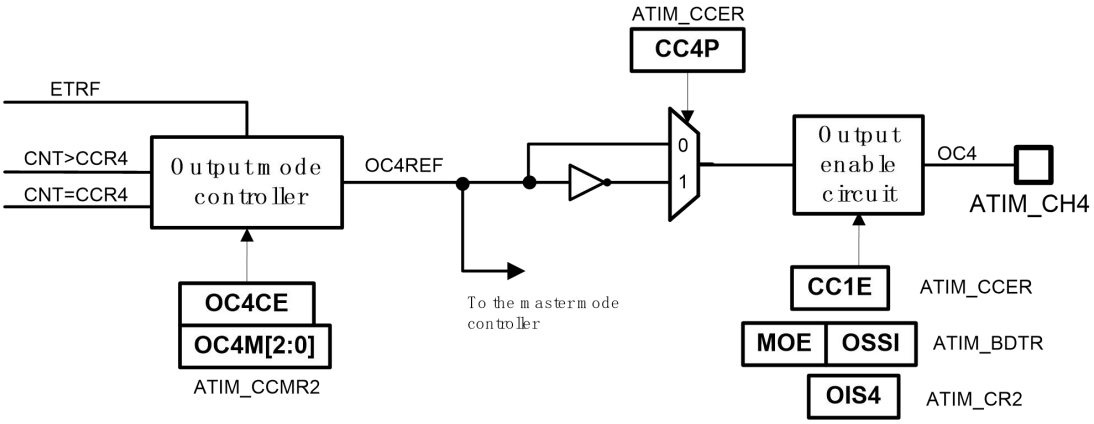


图 29-27 捕获/比较通道的输出部分(通道 4)

捕捉/比较寄存器（CCR）包含preload寄存器和shadow寄存器，软件读写总是访问preload寄存器。在捕捉模式下，捕捉值保存在shadow寄存器中并复制到preload寄存器。在比较模式下，preload寄存器的值被拷贝到shadow寄存器用来与计数器比较。

29.4.8 输入捕捉模式

当Icx信号上出现预期的电平变换，将触发一次capture，当前计数器值被锁存进CCR，与此同时，CcxIF中断标志置位，并且可以触发对应的中断或者DMA请求。如果一个捕捉事件在CcxIF为高的情况下出现，则捕捉数据冲突标志（CcxOF, Over-Capture）置位（CCR中上次捕捉值被覆盖）。CcxIF可以由软件清零，或者通过读取CCR寄存器自动清零。CcxOF标志通过软件写1清零。

通过两个或更多通道配合，可以实现PWM信号的输入捕捉。比如要计算一个输入信号的周期和占空比，可以将此信号从TI1引脚输入，芯片内部将滤波后的信号取上升沿得到TI1FP1，将滤波后的信号取下降沿得到TI1FP2，将TI1FP1输入给捕捉通道1，将TI1FP2输入给捕捉通道2，即可实现通道1对输入信号上升沿捕捉，同时通道2对输入信号下降沿捕捉；捕捉中断定期发生后，软件通过CCR1和CCR2寄存器的值，即可计算输入信号的周期和占空比。

实现在TI1输入的上升沿捕获计数器的值到ATIM_CCR1寄存器，配置步骤如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0，确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01, IC1映射到TI1
- 选择计数有效沿，配置ATIM_CCER.CC1P，选择上沿或者下沿
- 配置输入滤波时间，配置ATIM_CCMR1.IC1F[3:0]
- 配置输入预分频器，配置ATIM_CCMR1.IC1PS[1:0]
- 打开通道使能，配置ATIM_CCER.CC1E=1

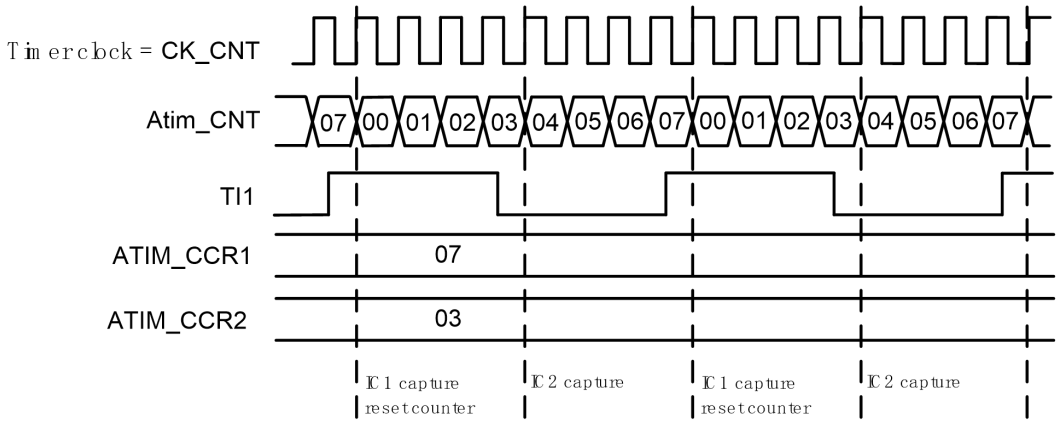


图 29-28 PWM 输入捕获模式时序

若想实现PWM输入捕获功能，需进行如下设置：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0，ATIM_CCER.CC2E=0确保之后通道配置成功
- 选择输入通道，两个通道IC1,IC2被映射到同一个TI1输入口，配置ATIM_CCMR1.CC1S=01,

ATIM_CCMR1.CC2S=10

- 选择计数有效沿，两个通道IC1,IC2有效沿极性相反，配置ATIM_CCER.CC1P=0，ATIM_CCER.CC2P=1
- 配置输入滤波时间，配置ATIM_CCMR1.IC1F[3:0]，ATIM_CCMR1.IC2F[3:0]
- 配置输入预分频器，配置ATIM_CCMR1.IC1PS[1:0] ，ATIM_CCMR1.IC2PS[1:0]
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101
- 设定从模式控制器为复位模式，配置ATIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置ATIM_CCER.CC1E=1，ATIM_CCER.CC2E=1

29.4.9 软件 Force 输出

在比较输出模式下，软件可以直接将OCxREF force成特定电平，而独立于CCR和计数器的比较结果。

软件通过写OcxM=101寄存器，可以直接将OCxREF强制为有效（OCxREF固定为高有效），通过写OcxM=100可以直接将OCxREF强制为无效（低电平）。但是软件force操作不会取消比较过程，CCR和计数器的比较还会一直进行。

29.4.10 输出比较模式

输出比较模式下，当CCR与计数器值相等，OCxREF可以被置位成有效、无效、或电平翻转。同时，中断标志也会置位，DMA请求可以发送。

输出比较也可以被用于输出一个特定宽度的脉冲信号（单次输出）。

使用步骤：

- 选择计数时钟（内部、外部、预分频等）
- 向ARR和CCR寄存器写入期望数据
- 根据需要设置中断使能和DMA使能
- 选择输出模式
- 使能计数器

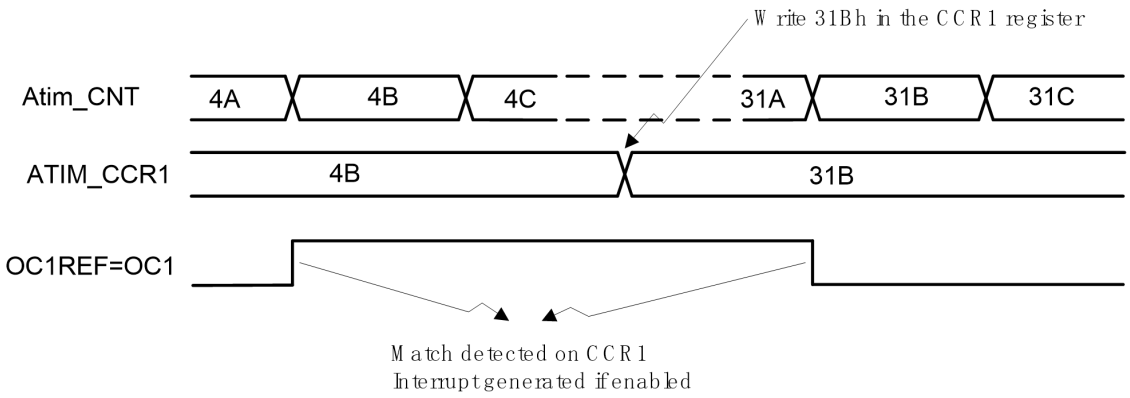


图 29-29 输出比较模式，翻转 OC1

在不使能preload的情况下，软件可以随时改写CCR寄存器实现对输出波形的实时控制。如果使能了preload，则CCR shadow寄存器仅在下一次update event发生时更新为preload寄存器的内容。

29.4.11 PWM 输出

PWM模式可以输出脉宽调制信号，其周期由ARR寄存器决定，占空比由CCR寄存器决定。

输出信号的极性可以由CCxP寄存器配置。PWM模式工作中，CNT和CCR实时比较。由于计数器支持边缘对齐和中央对齐计数模式，PWM输出也支持边缘对齐和中央对齐模式。

PWM边缘对齐模式

在向上计数的情况下，配置为PWM模式1时，OCxREF信号在CNT<CCR时为高电平，否则为低电平。如果CCR值大于ARR值，则OCxREF被固定为1；如果CCR为0则OCxREF被固定为0.

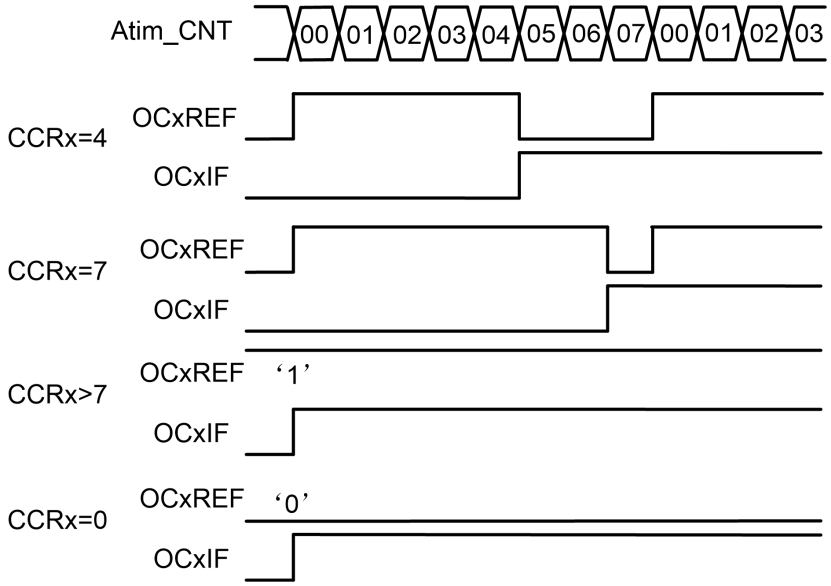


图 29-30 边沿对齐的 PWM 波形(ARR=7)

在向下计数时，OCxREF电平高低定义与向上计数时相同。

PWM中央对齐模式

OCxREF电平定义与边缘对齐模式相同。下图是一个示例：

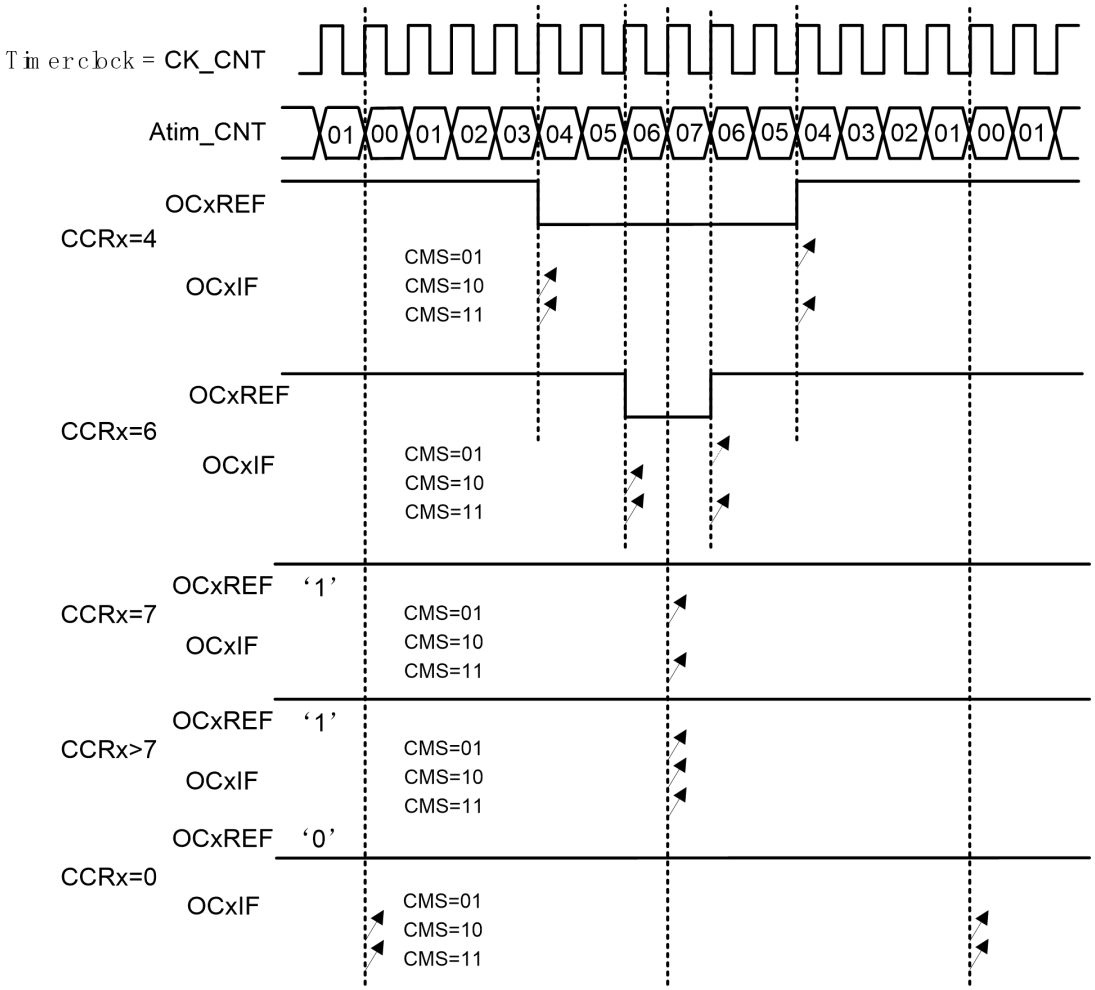


图 29-31 中央对齐的 PWM 波形(APR=7)

当启动中央对齐计数时，一开始的计数方向是由DIR寄存器决定的；随后在计数过程中，DIR寄存器的状态由硬件直接控制。安全起见，建议用户程序在启动计数器之前，通过UG寄存器做一次update，并且在计数过程中不要改写计数器。

29.4.12 互补输出和死区插入

ATIM的通道1~3支持互补输出和死区插入。DTG[7:0]寄存器用于设置死区时间（对所有通道同时有效）。输出信号Ocx与参考信号OCxREF同相，OcxN与参考信号反相；Ocx的上升沿是OCxREF上升沿的delay，OcxN的上升沿是OCxREF下降沿的delay。

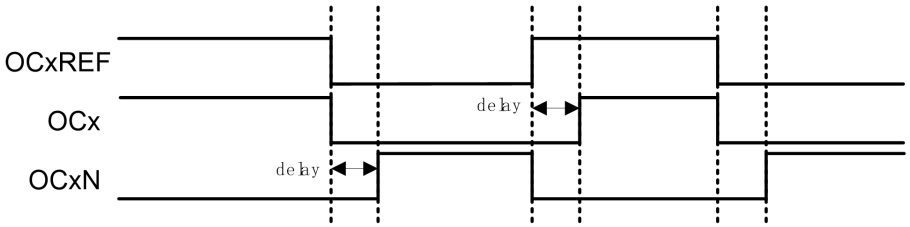


图 29-32 带死区插入的互补输出

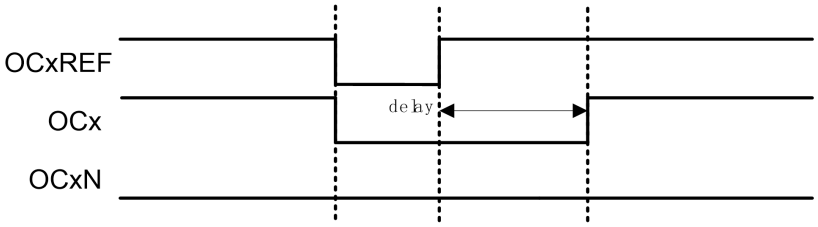


图 29-33 死区波形延迟大于负脉冲

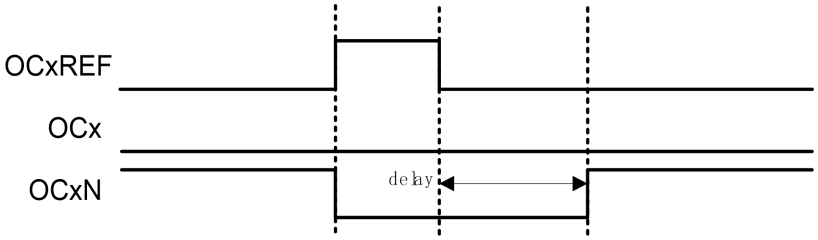
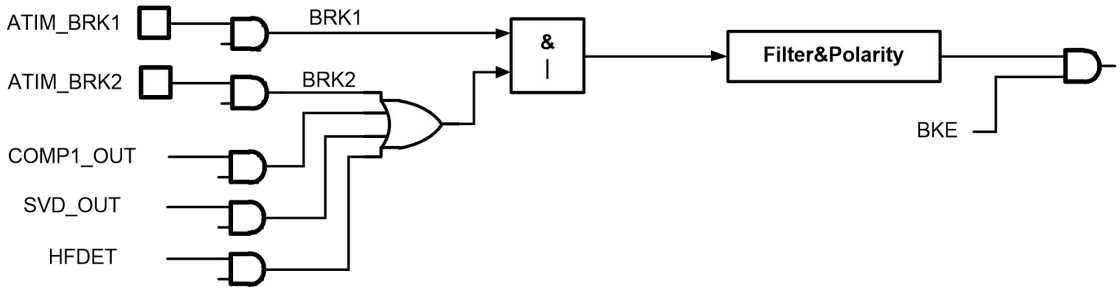


图 29-34 死区波形延迟大于正脉冲

29.4.13 刹车功能

刹车功能可以使用外部BRK引脚输入的2路刹车信号，或者比较器、SVD、XTHF停振检测产生的有效输出；上电复位后刹车电路被禁止，用户通过置位BKE寄存器使能刹车功能；2路刹车输入可以配置为相与或者相或操作。组合后的刹车信号可以配置有效极性，以及数字滤波。

刹车输入控制逻辑如下图所示：



ATIM_BRKx复用GPIO功能，当GPIO设置为数字外设功能时，其输入信号直接连接到ATIM的刹车输入上；当GPIO设置为其他功能时，ATIM的刹车输入端口被固定成1。通过BRKxGATE寄存器，可以控制门控后的BRKx信号的实际电平，软件能够灵活的将不使用的BRKx设置为0或者1电平，以适应后级逻辑电路的需要。

当一个刹车事件发生时：

- 输出使能寄存器被异步清零，可以通过OSSI寄存器选择输出被强制为inactive/idle/reset状态
- 每个输出通道被驱动为OISx寄存器定义的电平
- 当互补输出使能时，输出被异步置位成inactive和reset状态，死区插入电路开始工作，在死区时间后驱动输出为OISx和OISxN定义的电平
- 刹车标志寄存器置位，根据配置可以触发中断或DMA
- 如果使能了自动输出（AOE=1），输出使能位（MOE）将在下一个update event发生时被自动置位；否则MOE将保持为0直到被软件重新置位。

注意BRK信号是电平有效的，因此在BRK保持有效的情况下，无法使能MOE，同时刹车标志BIF也无法清除。

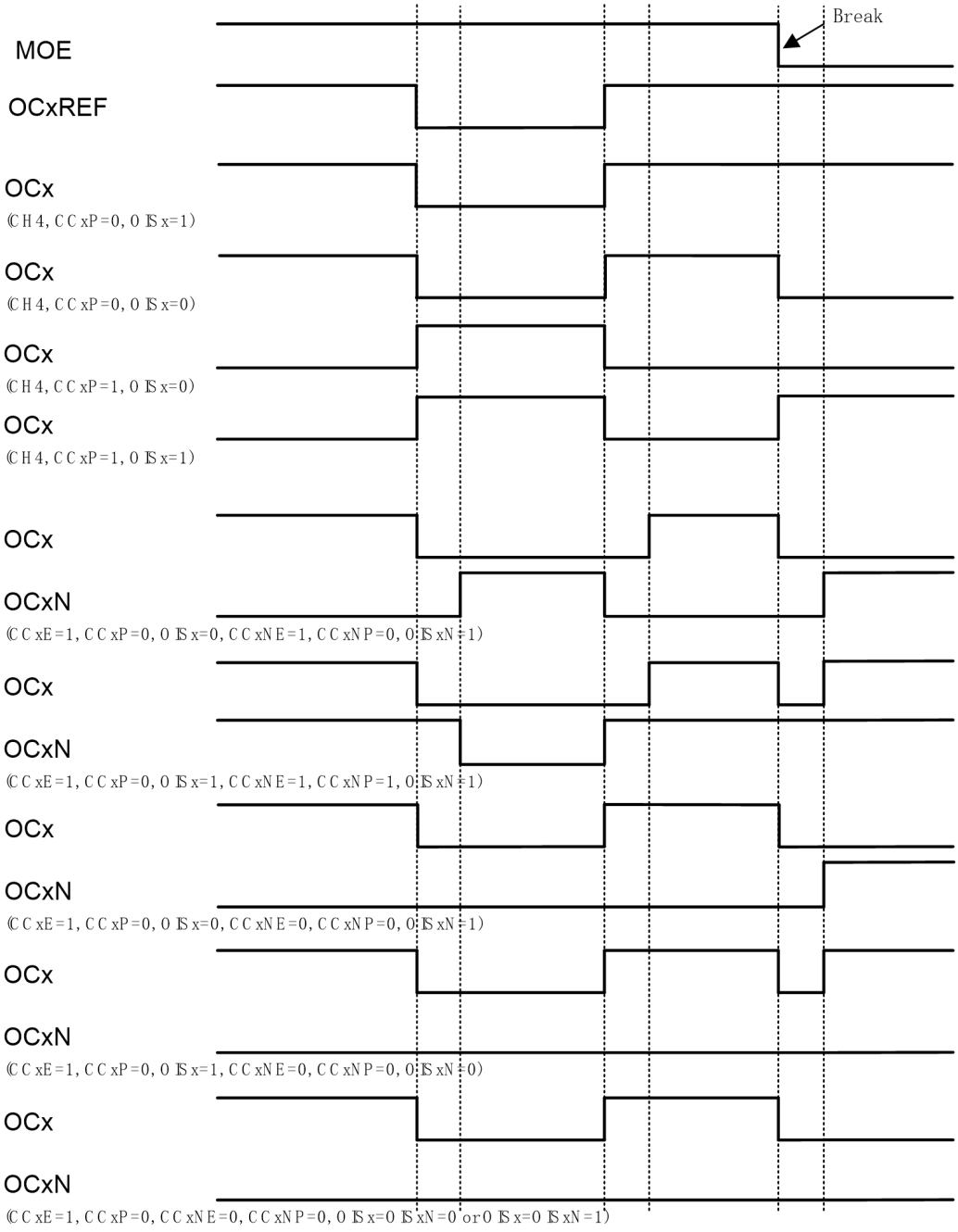


图 29-35 响应刹车的输出

29.4.14 互补输出通道信号状态逻辑表

以下是控制寄存器和互补输出通道的状态对应表，其中 MOE 为定时器总输出使能位，OSSI 定义 IDLE 状态（MOE=0）下是关闭 IO 输出还是进入 off state，OSSR 定义 RUN 状态（MOE=1）下的是关闭 IO 输出还是进入 off state。

控制寄存器					输出状态	
MOE	**OSSI**	**OSSR**	**CcxE**	**CcxNE**	**Ocx 输出状态**	**OcxN 输出状态**
1	X	0	0	0	输出关闭（不由ATIM驱动）,Ocx=0,Ocx_EN=0	输出关闭（不由ATIM驱动）,OcxN=0,OcxN_EN=0
		0	0	1	输出关闭（不由ATIM驱动）,Ocx=0,Ocx_EN=0	OCxREF + Polarity OcxN=OCxREF xor CCxNP, OcxN_EN=1
		0	1	0	OCxREF + Polarity Ocx=OCxREF xor CCxP, Ocx_EN=1	Output Disabled (not driven by the timer) OcxN=0, OcxN_EN=0
		0	1	1	OCREF + Polarity + dead-time Ocx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OcxN_EN=1
		1	0	0	Output Disabled (not driven by the timer) Ocx=CCxP, Ocx_EN=0	Output Disabled (not driven by the timer) OcxN=CCxNP, OcxN_EN=0
		1	0	1	Off-State (output enabled with inactive state) Ocx=CCxP, Ocx_EN=1	OCxREF + Polarity OcxN=OCxREF xor CCxNP, OcxN_EN=1
		1	1	0	OCxREF + Polarity Ocx=OCxREF xor CCxP, Ocx_EN=1	Off-State (output enabled with inactive state) OcxN=CCxNP, OcxN_EN=1
		1	1	1	OCREF + Polarity + dead-time Ocx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OcxN_EN=1
0	0	X	0	0	输出关闭（不由ATIM驱动） Ocx=CCxP, Ocx_EN=0	输出关闭（不由ATIM驱动） OcxN=CCxNP, OcxN_EN=0
	0		0	1	输出关闭（不由ATIM驱动） 如果无时钟：Ocx=CCxP, Ocx_EN=0, OcxN=CCxNP, OcxN_EN=0 如果有时钟：经过死区时间后Ocx=OISx, OcxN=OISxN	
	0		1	0		
	0		1	1		
	1		0	0	输出关闭（不由ATIM驱动） Ocx=CCxP, Ocx_EN=0	输出关闭（不由ATIM驱动） OcxN=CCxNP, OcxN_EN=0
	1		0	1	Off-state（输出使能，inactive输出） 如果无时钟：Ocx=CCxP, Ocx_EN=1, OcxN=CCxNP, OcxN_EN=1 如果有时钟：经过死区时间后Ocx=OISx, OcxN=OISxN	
	1		1	0		
	1		1	1		

29.4.15 6-step PWM 输出

当某个通道使用互补输出时，OcxM, CcxE, CcxNE寄存器支持preload功能，preload寄存器的值在换相（COM）事件发生时被装载到shadow寄存器中。用户因此可以预先设置下一步配置，并在COM事件发生时同步更新所有通道。COM事件可以由软件写ATIM_EGR中的COM位触发，或者由TRGI上升沿硬件触发。

当COM事件发生时，换相标志寄存器置位，并且可以产生中断或DMA请求。

下图是一个6步换相控制的例子，当COM事件发生时，三个例子显示不同配置下的输出变化。

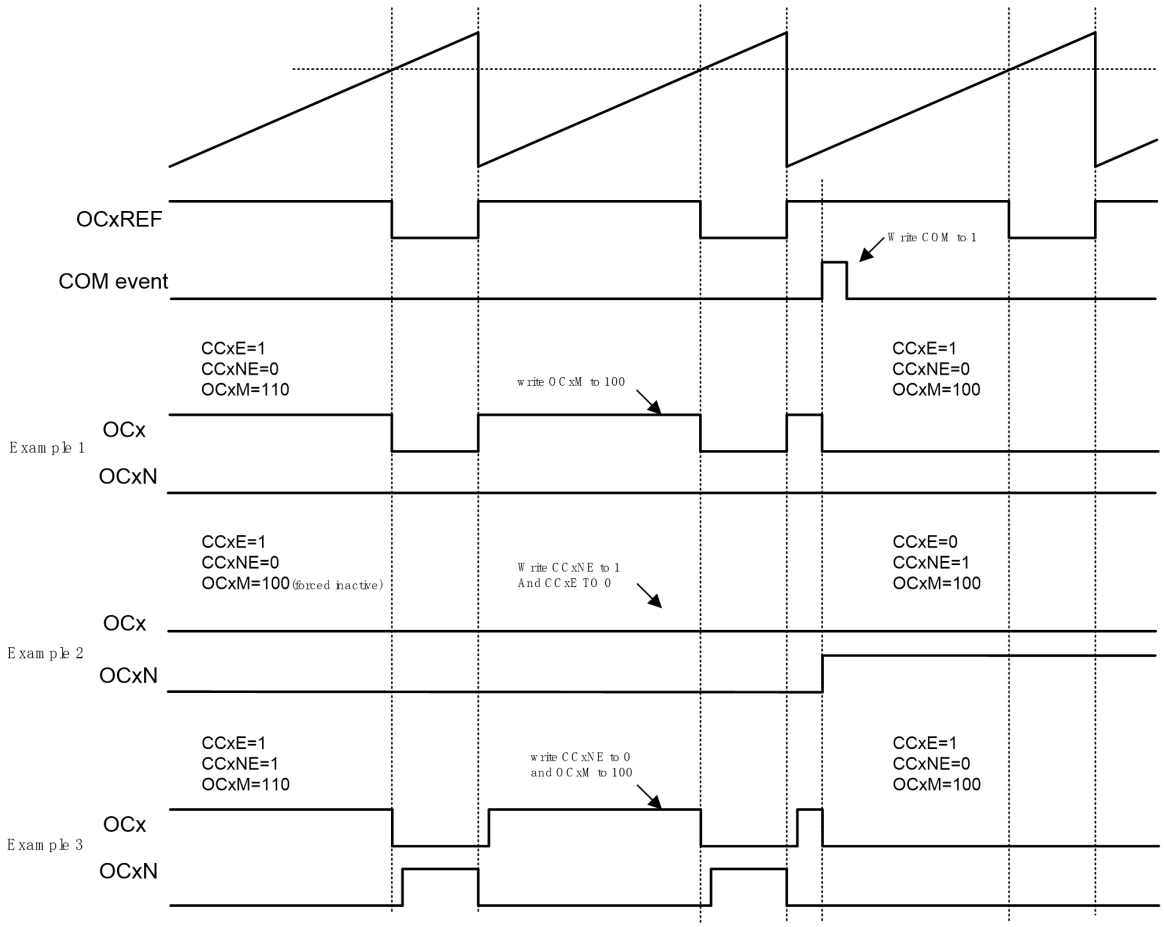


图 29-36 产生六步 PWM，使用 COM 的例子(OSSR=1)

29.4.16 单脉冲输出

单脉冲输出是比较输出模式的特殊情况，允许用户在某个事件发生后，经过可编程的延迟，输出一个可编程宽度的脉冲信号。

与其他输出模式不同的是，在下一次update event到来时，计数器会自动停止。只有当CCR和计数器初值不同时，脉冲才有可能正确输出。在向上计数时，要求CNT<CCR<=ARR，在向下计数时，要求CNT>CCR

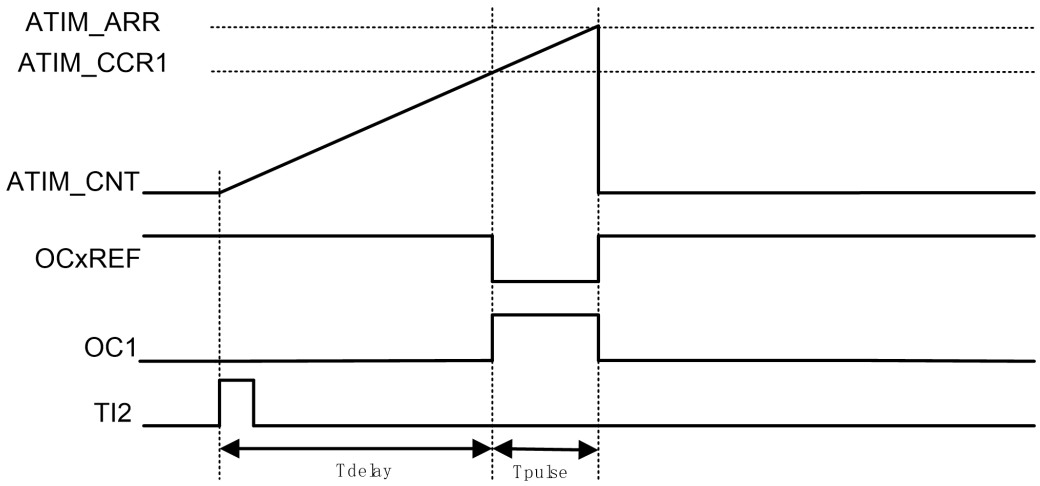


图 29-37 单脉冲模式的例子

上图是以TI2输入为计数器触发信号，计数值等于CCR后OCxREF输出低电平，计数到ARR后OCxREF回到高电平，并且计数器回滚到0，停止计数。

实现上述功能TI2作为输入触发的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH2功能
- 关闭通道使能，配置ATIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC2S=01
- 选择计数有效沿，配置ATIM_CCER.CC2P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=110，TI2FP2作为TRGI
- 设定从模式控制器为触发模式，配置ATIM_SMCR.SMS[2:0]=110,TI2FP2用来启动计数器
- 打开通道使能，配置ATIM_CCER.CC2E=1

实现上述功能OC1作为输出的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0，确保之后通道配置成功
- 输出通道，配置ATIM_CCMR1.CC1S=00

- 选择计数有效沿，配置ATIM_ CCMR1.OC1M=111,PWM模式2
- 打开通道使能，配置ATIM_CCER.CC1E=1

OPM波形产生时基的特殊设置：

- ATIM_CCR1的值决定了Tdelay
- ATIM_ARR和ATIM_CCR1的差值决定了Tpulse（ATIM_ARR-ATIM_CCR1）
- 设置为单脉冲模式，配置ATIM_CR1.OPM=1

29.4.17 外部事件清除 OCxREF

OCxREF的有效状态未高电平，通过对外部ETR引脚施加高电平，可以直接拉低OCxREF，直到下一次update event。此功能仅在输出比较和PWM模式下有效。使能此功能需要将OcxCE置1。

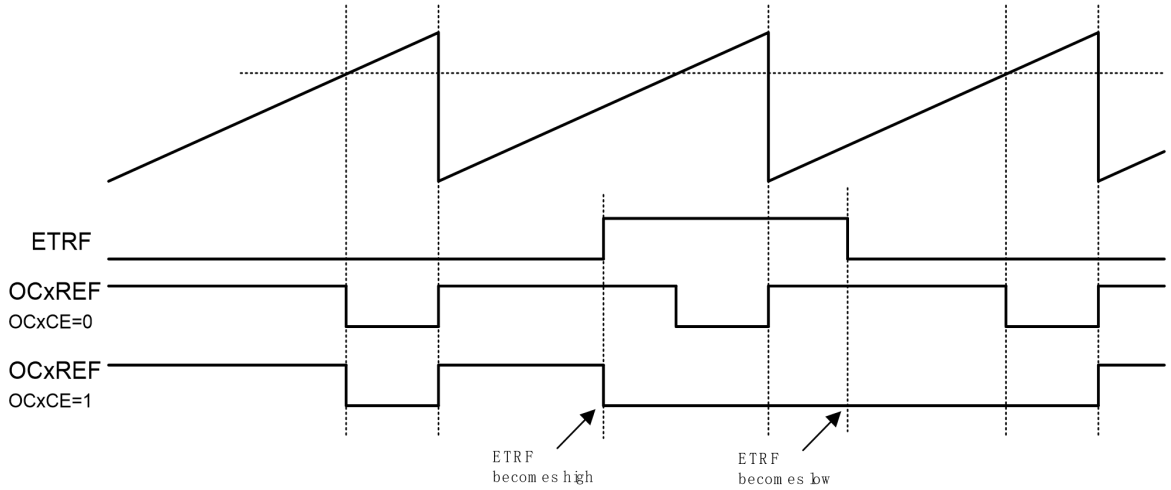


图 29-38 ETR 信号清除 ATIM 的 OCxREF

29.4.18 编码器接口模式（encoder interface）

编码器接口模式涉及到两个外部输入信号，ATIM根据其中一个信号的边沿相对于另一个信号的电平来决定递增还是递减计数值。下表是计数方式与两路输入信号之间的关系：

有效沿	对应信号的电平（TI1 对应TI2，TI2对应TI1）	TI1信号		TI2信号	
		上升	下降	上升	下降
仅在TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在TI2处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在TI1 和TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

表29-1encoder interface计数方式

比如在计数器以TI1信号为时钟计数时，如果TI1上升沿采样到TI2为高电平，则计数器递减；如果TI1下降沿采样到TI2为高电平，则计数器递增。

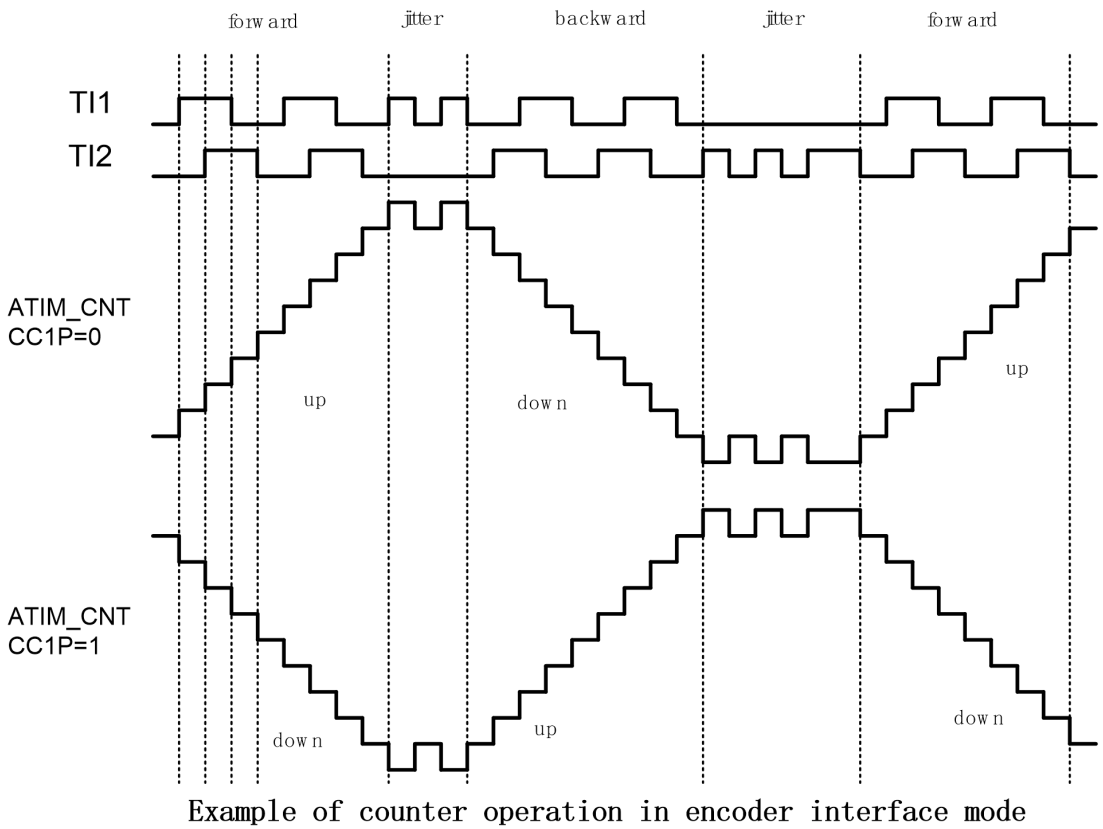


图 29-39 编码器模式下的计数器操作实例

编码模式输入通道需进行如下设置：

- 在GPIO模块中，配置相应管脚为ATIM_CH1，ATIM_CH2功能

- 关闭通道使能，配置ATIM_CCER.CC1E=0，ATIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01，ATIM_CCMR1.CC2S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0，ATIM_CCER.CC2P=0
- 设定从模式控制器为编码模式3，配置ATIM_SMCR.SMS[2:0]=011
- 打开通道使能，配置ATIM_CCER.CC1E=1，ATIM_CCER.CC2E=1

29.4.19 TIM 从机模式

ATIM作为slave时（外部事件触发），可配置为三种工作模式：复位模式、门控模式、触发模式。

复位模式

此模式下，外部输入的事件将导致TIM内部所有preload寄存器重新初始化，CNT回到0开始计数。以下图为例，计数器正常计数，外部TI1输入上升沿时，触发计数器清零，重新开始计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为复位模式，配置ATIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置ATIM_CCER.CC1E=1
- 使能计数器，配置ATIM_CR1.CEN=1

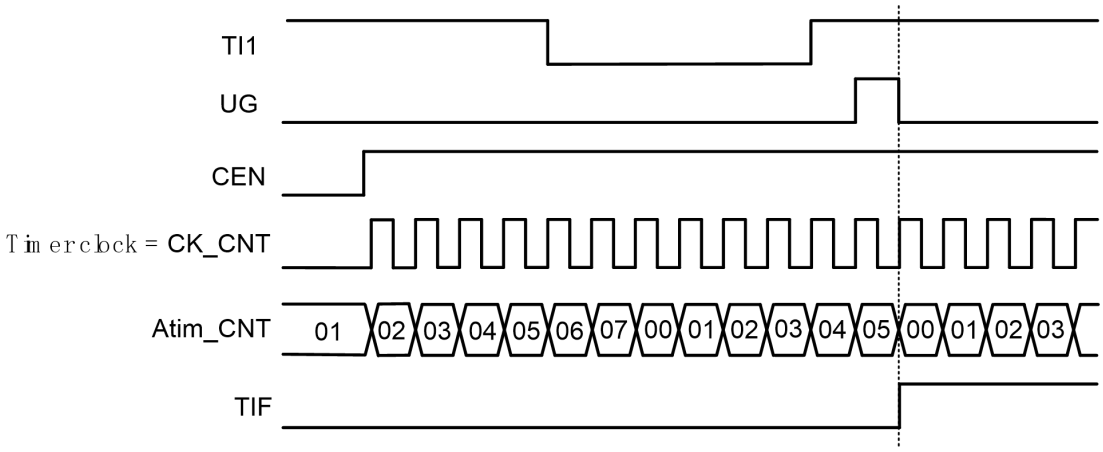


图 29-40 复位模式下的时序

门控模式

此模式下，计数器仅在输入信号为特定电平时工作。电平变换导致计数器开始或停止计数时，都会触发中断标志。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能

- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为门控模式，配置ATIM_SMCR.SMS[2:0]=101
- 打开通道使能，配置ATIM_CCER.CC1E=1
- 使能计数器，配置ATIM_CR1.CEN=1

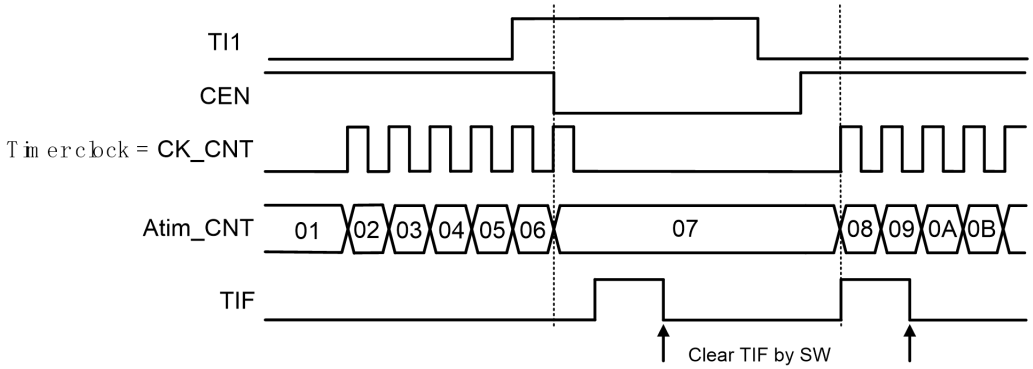


图 29-41 门控模式下的时序

触发模式

计数器在外部输入的某个事件到来后才开始计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为触发模式，配置ATIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置ATIM_CCER.CC1E=1

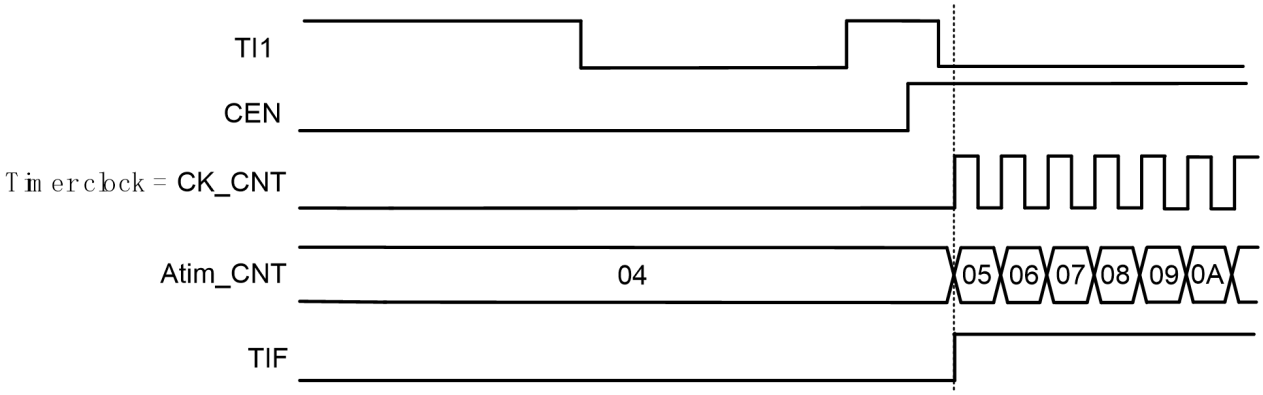


图 29-42 触发器模式下的时序

外部事件触发的外部时钟计数模式

可以将ETR设置为计数时钟，同时使用另一个外部输入作为计数器启动触发信号。比如在检测到TI1的上升沿之后，计数器开始以ETR输入的上升沿计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1，ATIM_ETR功能
- 设置ETP进行沿选择，ATIM_SMCR.ETP=0
- 设置ETR分频比，配置ATIM_SMCR.ETPS[1:0]=01
- 配置输入滤波时间，ATIM_SMCR.ETF[3:0]=0000
- 置位ECE寄存器，使能外部时钟模式2,ATIM_SMCR.ECE=1
- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为触发模式，配置ATIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置ATIM_CCER.CC1E=1

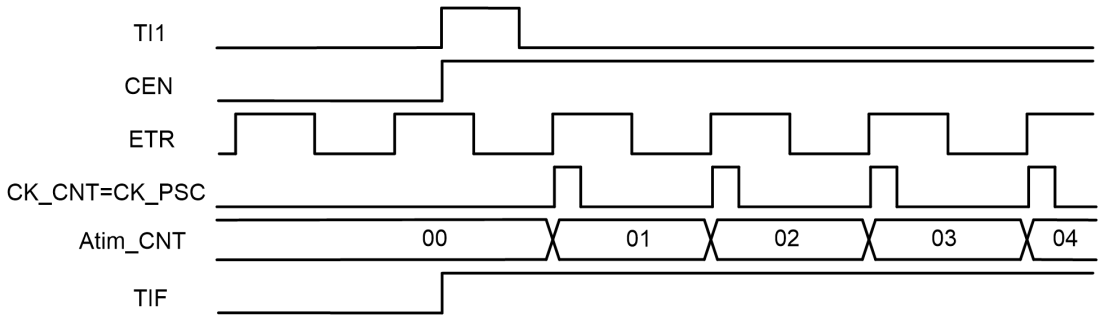


图 29-43 外部时钟模式 2＋触发模式下的时序

29.4.20 DMA 访问

ATIM支持7种DMA请求，分别为4个CC通道请求、外部触发请求、用户软件触发请求和COM触发请求。

其中每个CC通道各自产生一个DMA请求，在捕捉模式下用于将CCRx中的内容传输给RAM，在比较模式下则用于将RAM中的数据写入CCRx；CC通道的DMA请求可以配置为单次传输或Burst传输（CCxBURSTEN），单次传输仅访问CCRx寄存器，Burst传输则根据DCR寄存器配置对特定的一组寄存器进行访问。

此外，外部触发事件、软件触发事件和COM事件也可以产生DMA请求，当这些请求发生时，会启动DMA Burst传输，向ATIM内部1个或多个寄存器写入数据，或者从ATIM读取1个或多个寄存器值。

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
ATIM_CH1	0	0	Read CCR1	1
		1	Write CCR1	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH2	0	0	Read CCR2	1
		1	Write CCR2	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH3	0	0	Read CCR3	1
		1	Write CCR3	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH4	0	0	Read CCR4	1
		1	Write CCR4	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_TRIG	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_UEV	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_COM	N/A	0	Read DMAR	DBL
		1	Write DMAR	

29.4.21 DMA Burst

ATIM支持DMA和DMA-Burst访问，可以配置ATIM在特定事件发生时触发DMA请求，可以将CCR中的捕捉结果写入RAM，或者从RAM中将一个或多个寄存器内容写入ATIM的preload寄存器中。

DMA-Burst支持一个事件触发连续多次DMA请求，主要作用是在事件发生后连续更新多个寄存器的内容，因此可以实现动态实时调整输出波形等功能。

DMA控制器需将外设目标地址指向一个虚拟寄存器ATIM_DMAR。在特定的定时器事件发生时，ATIM会连续发射多个DMA请求。每个DMA对ATIM_DMAR的写操作都会被ATIM重新定向到实际的功能寄存器上。

DBL寄存器用于设置DMA burst长度，DBA寄存器用于设置DMA访问ATIM内部的基地址（相对于ATIM_CR的offset）。

29.4.22 输入异或功能

通道1~3的输入信号可以被异或起来之后，接入到通道1的滤波和边沿电路输入，用于通道1的输入捕捉或者触发。

ATIM_CR2寄存器的TI1S位用于选择通道1的输入是否来自于三个通道输入的异或。

29.4.23 Debug 模式

当Cortex-M0进入debug模式后，定时器可以停止或继续工作，其行为由DCU模块的DBG_TIMx_STOP寄存器定义。

Debug时当定时器被停止后，其输出会被禁止（MOE清零），根据寄存器配置，此时的输出信号可以被force成inactive或由GPIO模块控制。DMA-Burst模式下，DMA所有访问都要指向DMAR虚拟寄存器，由ATIM自动根据访问来累加内部offset地址。DBA寄存器用于指定ATIM内部首次DMA传输的目标地址，而DBL用于指定Burst长度

29.5 寄存器

ATIM模块基地址：0x40013000

Offset 地址	名称	符号
0x00	ATIM 控制寄存器 1	ATIM_CR1
0x04	ATIM 控制寄存器 2	ATIM_CR2
0x08	ATIM 从机模式控制寄存器	ATIM_SMCR
0x0C	ATIM DMA 和中断使能寄存器	ATIM_DIER
0x10	ATIM 状态寄存器	ATIM_SR
0x14	ATIM 事件产生寄存器	ATIM_EGR
0x18	ATIM 捕捉/比较模式寄存器 1	ATIM_CCMR1
0x1C	ATIM 捕捉/比较模式寄存器 2	ATIM_CCMR2
0x20	ATIM 捕捉/比较使能寄存器	ATIM_CCER
0x24	ATIM 计数器寄存器	ATIM_CNT
0x28	ATIM 预分频寄存器	ATIM_PSC
0x2C	ATIM 自动重载寄存器	ATIM_ARR
0x30	ATIM 重复计数寄存器	ATIM_RCR
0x34	ATIM 捕捉/比较寄存器 1	ATIM_CCR1
0x38	ATIM 捕捉/比较寄存器 2	ATIM_CCR2
0x3C	ATIM 捕捉/比较寄存器 3	ATIM_CCR3
0x40	ATIM 捕捉/比较寄存器 4	ATIM_CCR4
0x44	ATIM 刹车和死区控制寄存器	ATIM_BDTR
0x48	ATIM DMA 控制寄存器	ATIM_DCR
0x4C	ATIM DMA 访问寄存器	ATIM_DMAR
0x60	ATIM 刹车输入控制寄存器	ATIM_BKCTL

29.5.1 ATIM 控制寄存器 1

名称	ATIM_CR1							
地址	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CKD	
位权限	U-0						RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	CMS		DIR	OPM	URS	UDIS	CEN
位权限	RW-0	RW-00		RW-0	RW-0	RW-0	RW-0	RW-0

Bit	助记符	功能描述
31:10	--	RFU：未实现，读为 0
9:8	CKD	Dead time 和数字滤波时钟频率分频寄存器（相对 CK_INT 的分频比）

Bit	助记符	功能描述
		00：$t_{DTS}=t_{CK_INT}$ 01：$t_{DTS}=2*t_{CK_INT}$ 10：$t_{DTS}=4*t_{CK_INT}$ 11：RFU，禁止使用
7	**ARPE**	Auto-reload 预装载使能 0：ARR 寄存器不使能 preload 1：ARR 寄存器使能 preload
6:5	**CMS**	计数器对齐模式选择 00：边沿对齐模式 01：中央对齐模式 1，输出比较中断标志仅在计数器向下计数的过程中置位 10：中央对齐模式 2，输出比较中断标志仅在计数器向上计数的过程中置位 11：中央对齐模式 3，输出比较中断标志在计数器向上向下计数的过程中都会置位
4	**DIR**	计数方向寄存器 0：向上计数 1：向下计数 注意：当定时器配置为中央计数模式或编码器模式时，此寄存器只读
3	**OPM**	单脉冲输出模式 0：Update Event 发生时计数器不停止 1：Update Event 发生时计数器停止（自动清零 CEN）
2	**URS**	更新请求选择 0：以下事件能够产生 update 中断或 DMA 请求 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1：仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求
1	**UDIS**	禁止 update 0：使能 update 事件；以下事件发生时产生 update 事件 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1：禁止 update 事件，不更新 shadow 寄存器。当 UG 置位或从机控制器收到硬件 reset 时重新初始化计数器和预分频器。
0	**CEN**	计数器使能 0：计数器关闭 1：计数器使能 注意：外部触发模式可以自动置位 CEN

29.5.2 ATIM 控制寄存器 2

名称	ATIM_CR2							
地址	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1
位权限	U-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TI1S	MMS			CCDS	CCUS	-	CCPC
位权限	RW-0	RW-000			RW-0	RW-0	U-0	RW-0

Bit	助记符	功能描述
31:15	--	RFU：未实现，读为 0
14	**OIS4**	参考 OIS1
13	**OIS3N**	参考 OIS1N
12	**OIS3**	参考 OIS1
11	**OIS2N**	参考 OIS1N
10	**OIS2**	参考 OIS1
9	**OIS1N**	定义 OC1N 的输出 IDLE 状态 0：当 MOE=0 时，经过 dead time 后，OC1N=0 1：当 MOE=0 时，经过 dead time 后，OC1N=1
8	**OIS1**	定义 OC1 的输出 IDLE 状态 0：当 MOE=0 时（如果使能了互补输出，需经过 dead time 后），OC1=0 1：当 MOE=0 时（如果使能了互补输出，需经过 dead time 后），OC1=1
7	**TI1S**	ATIM 输入 TI1 选择 0：ATIM_CH1 引脚连接到 TI1 输入 1：ATIM_CH1、CH2、CH3 引脚 XOR 后连接到 TI1 输入
6:4	**MMS**	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源 000：ATIM_EGR 的 UG 寄存器被用作 TRGO 001：计数器使能信号 CNT_EN 被用作 TRGO，可用于同时启动多个定时器 010：UE（update event）信号被用作 TRGO 011：比较脉冲，如果 CC1IF 标志将要置位，TRGO 输出一个正脉冲 100：OC1REF 用作 TRGO 101：OC2REF 用作 TRGO 110：OC3REF 用作 TRGO 111：OC4REF 用作 TRGO 注意：从机定时器或 ADC 必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
3	**CCDS**	捕捉/比较 DMA 选择 0：捕捉/比较事件发生时发送 DMA 请求 1：Update Event 发生时发送 DMA 请求
2	**CCUS**	捕捉/比较控制寄存器更新选择 0：当捕捉/比较控制寄存器使能了 preload（CCPC=1），他们仅在置位 COMG 寄存器时更新 1：当捕捉/比较控制寄存器使能了 preload（CCPC=1），他们在置位 COMG 寄存器或者 TRGI 上升沿时更新

Bit	助记符	功能描述
1	--	RFU：未实现，读为 0
0	**CCPC**	捕捉/比较预装载控制 0：CcxE, CcxNE, OcxM 寄存器不使能 preload 1：CcxE, CcxNE, OcxM 寄存器使能 preload 注意：此寄存器仅在拥有互补输出功能的通道上有效

29.5.3 ATIM 从机模式控制寄存器

名称	ATIM_SMCR							
地址	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ETP	ECE	ETPS		ETF			
位权限	RW-0	RW-0	RW-00		RW-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSM	TS			-	SMS		
位权限	RW-0	RW-000			U-0	RW-000		

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	**ETP**	外部触发信号极性配置 0：高电平或上升沿有效 1：低电平或下降沿有效
14	**ECE**	外部时钟使能 0：关闭外部时钟模式 2 1：使能外部时钟模式 2，计数器时钟为 ETRF 有效沿
13:12	**ETPS**	外部触发信号预分频寄存器 外部触发信号 ETRP 的频率最多只能是 ATIM 工作时钟的 1/4，当输入信号频率较高时，可以使用预分频。 00：不分频 01：2 分频 10：4 分频 11：8 分频
11:8	**ETF**	外部触发信号滤波时钟和长度选择 0000：无滤波 0001：$f_{SAMPLING}=f_{CK_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6

Bit	助记符	功能描述
		1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
7	**MSM**	主机从机模式选 0：无动作 1：TRGI 触发的动作被延迟，以便于通过 TRGO 将当前定时器和从机定时器完美同步起来
6:4	**TS**	触发选择，用于选择同步计数器的触发源 000：内部触发信号（ITR0） 001：内部触发信号（ITR1） 010：内部触发信号（ITR2） 011：内部触发信号（ITR3） 100：TI1 边沿检测（TI1F_ED） 101：滤波后 TI1（TI1FP1） 110：滤波后 TI2（TI2FP2） 111：外部触发输入（ETRF） 注意：仅当 SMS=000 即禁止从机模式的情况下，可以改写 TS 寄存器
3	**--**	RFU：未实现，读为 0
2:0	**SMS**	从机模式选择 000：从机模式禁止；CEN 使能后预分频电路时钟源来自内部时钟 001：Encoder 模式 1；计数器使用 TI2FP1 边沿，根据 TI1FP2 电平高低来计数 010：Encoder 模式 2；计数器使用 TI1FP2 边沿，根据 TI2FP1 电平高低来计数 011：Encoder 模式 3；计数器同时使用 TI1FP1 和 TI2FP2 边沿，根据其他输入信号电平来计数 100：复位模式；TRGI 上升沿初始化计数器，并触发寄存器 update 101：闸门模式；TRGI 为高电平时，计数时钟使能，TRGI 为低电平时，计数时钟停止 110：触发模式；TRGI 上升沿触发计数器开始计数（不会复位计数器） 111：外部时钟模式 1；TRGI 上升沿直接驱动计数器

29.5.4 ATIM DMA 和中断使能寄存器

名称	ATIM_DIER							
地址	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-				CC4BURSTEN	CC3BURSTEN	CC2BURSTEN	CC1BURSTEN
位权限	U-0				RW-0	RW-0	RW-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE
位权限	U-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19	**CC4BURSTEN**	捕捉比较通道 4 的 DMA 模式配置 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
18	**CC3BURSTEN**	捕捉比较通道 3 的 DMA 模式配置 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
17	**CC2BURSTEN**	捕捉比较通道 2 的 DMA 模式配置 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
16	**CC1BURSTEN**	捕捉比较通道 1 的 DMA 模式配置 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
15	--	RFU：未实现，读为 0
14	**TDE**	外部触发 DMA 请求使能 0：从机模式下，禁止外部触发事件产生 DMA 请求 1：从机模式下，允许外部触发事件产生 DMA 请求（可用于自动更新 preload 寄存器）
13	**COMDE**	COM 事件 DMA 请求使能 0：COM 事件发生时，禁止产生 DMA 请求 1：COM 事件发生时，允许产生 DMA 请求
12	**CC4DE**	捕捉比较通道 4 的 DMA 请求使能 0：禁止 CC4 DMA 请求 1：允许 CC4 DMA 请求
11	**CC3DE**	捕捉比较通道 3 的 DMA 请求使能 0：禁止 CC3 DMA 请求 1：允许 CC3 DMA 请求
10	**CC2DE**	捕捉比较通道 2 的 DMA 请求使能 0：禁止 CC2 DMA 请求 1：允许 CC2 DMA 请求
9	**CC1DE**	捕捉比较通道 1 的 DMA 请求使能 0：禁止 CC1 DMA 请求 1：允许 CC1 DMA 请求
8	**UDE**	Update Event DMA 请求使能 0：Update Event 发生时，禁止产生 DMA 请求 1：Update Event 发生时，允许产生 DMA 请求
7	**BIE**	刹车事件中断使能 0：禁止刹车事件中断 1：允许刹车事件中断

Bit	助记符	功能描述
6	**TIE**	触发事件中断使能 0：禁止触发事件中断 1：允许触发事件中断
5	**COMIE**	COM 事件中断使能 0：禁止 COM 事件中断 1：允许 COM 事件中断
4	**CC4IE**	捕捉/比较通道 4 中断使能 0：禁止捕捉/比较 4 中断 1：允许捕捉/比较 4 中断
3	**CC3IE**	捕捉/比较通道 3 中断使能 0：禁止捕捉/比较 3 中断 1：允许捕捉/比较 3 中断
2	**CC2IE**	捕捉/比较通道 2 中断使能 0：禁止捕捉/比较 2 中断 1：允许捕捉/比较 2 中断
1	**CC1IE**	捕捉/比较通道 1 中断使能 0：禁止捕捉/比较 1 中断 1：允许捕捉/比较 1 中断
0	**UIE**	Update 事件中断使能 0：禁止 Update 事件中断 1：允许 Update 事件中断

29.5.5 ATIM 状态寄存器

名称	ATIM_SR							
地址	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			CC4OF	CC3OF	CC2OF	CC1OF	-
位权限	U-0			RW-0	RW-0	RW-0	RW-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Bit	助记符	功能描述
31:13	**--**	RFU：未实现，读为 0
12	**CC4OF**	捕捉/比较通道 4 的 Overcapture 中断 参考 CC1OF
11	**CC3OF**	捕捉/比较通道 3 的 Overcapture 中断 参考 CC1OF
10	**CC2OF**	捕捉/比较通道 2 的 Overcapture 中断 参考 CC1OF
9	**CC1OF**	捕捉/比较通道 1 的 Overcapture 中断

Bit	助记符	功能描述
		此寄存器仅在对应通道设置为输入捕捉模式的情况下有效。硬件置位，软件写 1 清零。 0：无 overcapture 事件 1：在 CC1IF 标志为 1 的情况下发生新的捕捉
8	--	RFU：未实现，读为 0
7	BIF	刹车事件中断标志，硬件置位，软件写 1 清零
6	TIF	触发事件中断标志，硬件置位，软件写 1 清零
5	COMIF	COM 事件中断标志，硬件置位，软件写 1 清零
4	CC4IF	捕捉/比较通道 4 中断标志 参考 CC1IF
3	CC3IF	捕捉/比较通道 3 中断标志 参考 CC3IF
2	CC2IF	捕捉/比较通道 2 中断标志 参考 CC2IF
1	CC1IF	捕捉/比较通道 1 中断标志 如果 CC1 通道配置为输出：CC1IF 在计数值等于比较值时置位，软件写 1 清零。 如果 CC1 通道配置为输入：发生捕捉事件时置位，软件写 1 清零，或者软件读 ATIM_CCR1 自动清零。
0	UIF	Update 事件中断标志，硬件置位，软件写 1 清零。 当以下事件发生时，UIF 置位，并更新 shadow 寄存器 -重复计数器=0，并且 UDIS=0 的情况下，计数器发生溢出 -URS=0 且 UDIS=0 的情况下，软件置位 UG 寄存器初始化计数器 -URS=0 且 UDIS=0 的情况下，触发事件初始化计数器

29.5.6 ATIM 事件产生寄存器

名称	ATIM_EGR							
地址	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
位权限	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7	BG	软件刹车，软件置位此寄存器产生刹车事件，硬件自动清零
6	TG	软件触发，软件置位此寄存器产生触发事件，硬件自动清零
5	COMG	软件 COM 事件，硬件置位，软件写 1 清零

Bit	助记符	功能描述
4	**CC4G**	捕捉/比较通道 4 软件触发，参考 CC1G
3	**CC3G**	捕捉/比较通道 3 软件触发，参考 CC1G
2	**CC2G**	捕捉/比较通道 2 软件触发，参考 CC1G
1	**CC1G**	捕捉/比较通道 1 软件触发 如果 CC1 通道配置为输出：CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求 如果 CC1 通道配置为输入：当前计数值被捕捉到 ATIM_CCR1 寄存器，CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求
0	**UG**	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零 软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。

29.5.7 ATIM 捕捉/比较模式寄存器 1

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能

名称	ATIM_CCMR1							
地址	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC2CE	OC2M			OC2PE	OC2FE	CC2S	
	IC2F				IC2PSC			
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC1CE	OC1M			OC1PE	OC1FE	CC1S	
	IC1F				IC1PSC			
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

输出比较模式

Bit	助记符	功能描述
31:16	**--**	RFU：未实现，读为 0
15	**OC2CE**	输出比较 2 清零使能，参考 OC1CE
14:12	**OC2M**	输出比较 2 模式配置，参考 OC1M
11	**OC2PE**	输出比较 2 预装载使能，参考 OC1PE
10	**OC2FE**	输出比较 2 快速使能，参考 OC1FE
9:8	**CC2S**	捕捉/比较 2 通道选择 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 10：CC2 通道配置为输入，IC2 映射到 TI1 11：CC2 通道配置为输入，IC2 映射到 TRC 注意：CC2S 仅在通道关闭时（CC2E=0）可以写

Bit	助记符	功能描述
7	**OC1CE**	输出比较 1 清零使能 0：OC1REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	**OC1M**	输出比较 1 模式配置，此寄存器定义 OC1REF 信号的行为 000：输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出 001：CCR1=CNT 时，将 OC1REF 置高 010：CCR1=CNT 时，将 OC1REF 置低 011：CCR1=CNT 时，翻转 OC1REF 100：OC1REF 固定为低（inactive） 101：OC1REF 固定为高（active） 110：PWM 模式 1 –在向上计数时，OC1REF 在 CNT<CCR1 时置高，否则置低；在向下计数时，OC1REF 在 CNT>CCR1 时置低，否则置高 111：PWM 模式 2 –在向上计数时，OC1REF 在 CNT<CCR1 时置低，否则置高；在向下计数时，OC1REF 在 CNT>CCR1 时置高，否则置低
3	**OC1PE**	输出比较 1 预装载使能 0：CCR1 preload 寄存器无效，CCR1 可以直接写入 1：CCR1 preload 寄存器有效，针对 CCR1 的读写操作都是访问 preload 寄存器，当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	**OC1FE**	输出比较 1 快速使能 0：关闭快速使能，trigger 输入不会影响比较输出 1：打开快速使能，trigger 输入会立即将 OC1REF 改变为比较值匹配时的输出，而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效
1:0	**CC1S**	捕捉/比较 1 通道选择 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 10：CC1 通道配置为输入，IC1 映射到 TI2 11：CC1 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

输入捕捉模式

Bit	助记符	功能描述
31:16	**--**	RFU：未实现，读为 0
15:12	**IC2F**	输入捕捉 2 滤波
11:10	**IC2PSC**	输入捕捉 2 预分频
9:8	**CC2S**	捕捉/比较 2 通道选择 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC3 映射到 TI2 10：CC2 通道配置为输入，IC3 映射到 TI1 11：CC2 通道配置为输入，IC3 映射到 TRC 注意：CC2S 仅在通道关闭时（CC2E=0）可以写
7:4	**IC1F**	输入捕捉 1 滤波 此寄存器定义 TI1 的采样频率和滤波长度 0000：无滤波，使用 f_{DTS} 采样 0001：$f_{SAMPLING}=f_{CK_INT}$, N=2

Bit	助记符	功能描述
		0010：$f_{SAMPLING}=f_{CK_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC1PSC**	输入捕捉 1 预分频 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC1PSC 寄存器在 CC1E=0 时复位
1:0	**CC1S**	捕捉/比较 1 通道选择 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 10：CC1 通道配置为输入，IC1 映射到 TI2 11：CC1 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

29.5.8 ATIM 捕捉/比较模式寄存器 2

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能

名称	ATIM_CCMR2							
地址	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC4CE	OC4M			OC4PE	OC4FE	CC4S	
	IC4F				IC4PSC			
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC3CE	OC3M			OC3PE	OC3FE	CC3S	
	IC3F				IC3PSC			
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

输出比较模式

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	**OC4CE**	输出比较 4 清零使能，参考 OC1CE
14:12	**OC4M**	输出比较 4 模式配置，参考 OC1M
11	**OC4PE**	输出比较 4 预装载使能，参考 OC1PE
10	**OC4FE**	输出比较 4 快速使能，参考 OC1FE
9:8	**CC4S**	捕捉/比较 4 通道选择 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 10：CC4 通道配置为输入，IC4 映射到 TI3 11：CC4 通道配置为输入，IC4 映射到 TRC 注意：CC4S 仅在通道关闭时（CC4E=0）可以写
7	**OC3CE**	输出比较 3 清零使能 0：OC3REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC3REF
6:4	**OC3M**	输出比较 3 模式配置，此寄存器定义 OC3REF 信号的行为 000：输出比较寄存器 CCR3 和计数器 CNT 的比较结果不会影响输出 001：CCR3=CNT 时，将 OC1REF 置高 010：CCR3=CNT 时，将 OC1REF 置低 011：CCR3=CNT 时，翻转 OC1REF 100：OC3REF 固定为低（inactive） 101：OC3REF 固定为高（active） 110：PWM 模式 1 –在向上计数时，OC3REF 在 CNT<CCR3 时置高，否则置低；在向下计数时，OC3REF 在 CNT>CCR3 时置低，否则置高 111：PWM 模式 2 –在向上计数时，OC3REF 在 CNT<CCR3 时置低，否则置高；在向下计数时，OC3REF 在 CNT>CCR3 时置高，否则置低
3	**OC3PE**	输出比较 3 预装载使能 0：CCR3 preload 寄存器无效，CCR3 可以直接写入 1：CCR3 preload 寄存器有效，针对 CCR3 的读写操作都是访问 preload 寄存器，当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	**OC3FE**	输出比较 3 快速使能 0：关闭快速使能，trigger 输入不会影响比较输出 1：打开快速使能，trigger 输入会立即将 OC3REF 改变为比较值匹配时的输出，而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效
1:0	**CC3S**	捕捉/比较 3 通道选择 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC1 映射到 TI3 10：CC3 通道配置为输入，IC1 映射到 TI4 11：CC3 通道配置为输入，IC1 映射到 TRC 注意：CC3S 仅在通道关闭时（CC3E=0）可以写

输入捕捉模式

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:12	**IC4F**	输入捕捉 4 滤波

Bit	助记符	功能描述
11:10	**IC4PSC**	输入捕捉 4 预分频
9:8	**CC4S**	捕捉/比较 4 通道选择 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 10：CC4 通道配置为输入，IC4 映射到 TI3 11：CC4 通道配置为输入，IC4 映射到 TRC 注意：CC4S 仅在通道关闭时（CC4E=0）可以写
7:4	**IC3F**	输入捕捉 1 滤波 此寄存器定义 TI3 的采样频率和滤波长度 0000：无滤波，使用 f_{DTS} 采样 0001：$f_{SAMPLING}=f_{CK_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC3PSC**	输入捕捉 3 预分频 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC1PSC 寄存器在 CC1E=0 时复位
1:0	**CC3S**	捕捉/比较 3 通道选择 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC1 映射到 TI3 10：CC3 通道配置为输入，IC1 映射到 TI4 11：CC3 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

29.5.9 ATIM 捕捉/比较使能寄存器

名称	ATIM_CCER							
地址	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E
位权限	U-0		RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Bit	助记符	功能描述
31:14	--	RFU：未实现，读为 0
13	**CC4P**	捕捉/比较 4 输出极性，参考 CC1P
12	**CC4E**	捕捉/比较 4 输出使能，参考 CC1E
11	**CC3NP**	捕捉/比较 3 互补输出极性，参考 CC1NP
10	**CC3NE**	捕捉/比较 3 互补输出使能，参考 CC1NE
9	**CC3P**	捕捉/比较 3 输出极性，参考 CC1P
8	**CC3E**	捕捉/比较 3 输出使能，参考 CC1E
7	**CC2NP**	捕捉/比较 2 互补输出极性，参考 CC1NP
6	**CC2NE**	捕捉/比较 2 互补输出使能，参考 CC1NE
5	**CC2P**	捕捉/比较 2 输出极性，参考 CC1P
4	**CC2E**	捕捉/比较 2 输出使能，参考 CC1E
3	**CC1NP**	捕捉/比较 1 互补输出极性 0：OC1N 高电平为 active 1：OC1N 低电平为 active
2	**CC1NE**	捕捉/比较 1 互补输出使能 0：OC1N 无效，OC1N 电平由 MOE, OSSI, OSSR, OIS1, OIS1N, CC1E 寄存器决定 1：OC1N 有效
1	**CC1P**	捕捉/比较 1 输出极性 CC1 通道配置为输出时 0：OC1 高电平 active 1：OC1 低电平 active CC1 通道配置为输入时 0：非取反模式–捕捉在 IC1 的上升沿进行 1：取反模式–捕捉在 IC1 的下降沿进行
0	**CC1E**	捕捉/比较 1 输出使能 CC1 通道配置为输出时 0：OC1 使能 1：OC1 关闭 CC1 通道配置为输入时 0：关闭捕捉功能 1：使能捕捉功能

29.5.10 ATIM 计数器寄存器

名称	ATIM_CNT							
地址	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CNT**	计数器值

29.5.11 ATIM 预分频寄存器

名称	ATIM_PSC							
地址	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PSC[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**PSC**	计数器时钟（CK_CNT）预分频值 $f_{CK_CNT}=f_{CK_PSC}/(PSC[15:0]+1)$ 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

29.5.12 ATIM 自动重载（auto-reload）寄存器

名称	ATIM_ARR							
地址	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	RW-11111111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	RW-11111111							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**ARR**	计数溢出时的自动重载值 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

29.5.13 ATIM 重复计数寄存器

名称	ATIM_RCR							
地址	0x30							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	REP[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	**REP**	重复计数值。REP 不为 0 时，每次 update 条件发生时 REP 递减，当 REP=0 时触发 update 事件

29.5.14 ATIM 捕捉/比较寄存器 1

名称	ATIM_CCR1							
地址	0x34							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1[15:8]							
位权限	RW-00000000							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CCR1**	捕捉/比较通道 1 寄存器 **如果通道 1 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC1 输出 **如果通道 1 配置为输入：** CCR1 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR1 为只读

29.5.15 ATIM 捕捉/比较寄存器 2

名称	ATIM_CCR2							
地址	0x38							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CCR2**	捕捉/比较通道 2 寄存器 **如果通道 2 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC2 输出 **如果通道 2 配置为输入：** CCR2 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR2 为只读

29.5.16 ATIM 捕捉/比较寄存器 3

名称	ATIM_CCR3							
地址	0x3C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR3[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR3[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CCR3**	捕捉/比较通道 3 寄存器 **如果通道 3 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC3 输出 **如果通道 3 配置为输入：** CCR3 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR3 为只读

29.5.17 ATIM 捕捉/比较寄存器 4

名称	ATIM_CCR4							
地址	0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR4[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR4[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CCR4**	捕捉/比较通道 4 寄存器 **如果通道 4 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC4 输出 **如果通道 4 配置为输入：** CCR4 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR4 为只读

29.5.18 ATIM 刹车和死区控制寄存器

名称	ATIM_BDTR							
地址	0x44							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DTG							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	**MOE**	输出使能总控制位 此寄存器控制所有通道的输出使能，每个通道独立的输出使能还需要 CcxE 和 CcxNE 来控制。MOE 由软件置位，或者在 AOE=1 的情况下硬件触发自动置位。当刹车输入有效时，MOE 被硬件异步清零。 0：关闭 OC 和 OCN 输出，具体 IO 输出状态由 OSSI 决定 1：使能 OC 和 OCN 输出（仍需各个通道的 CcxE 和 CcxNE 状态来决定是否输出）
14	**AOE**	自动输出使能 0：MOE 仅能由软件置位 1：MOE 可以软件置位，或者由 update 事件自动置位
13	**BKP**	刹车极性 0：刹车输入为低电平有效 1：刹车输入为高电平有效
12	**BKE**	刹车使能 0：禁止刹车输入 1：允许刹车输入
11	**OSSR**	运行状态下的输出关闭状态选择 仅在 MOE=1 的情况下，针对使能了互补输出的通道有效。 0：输出通道不使能时，OC 和 OCN 不驱动 GPIO 1：输出通道不使能时，OC 和 OCN 驱动 GPIO 为无效状态
10	**OSSI**	IDLE 状态下的输出关闭状态选择 仅在 MOE=0 的情况下，针对输出通道有效。 0：输出通道不使能时，OC 和 OCN 不驱动 GPIO 1：输出通道不使能时，OC 和 OCN 先驱动无效电平，待死区时间结束后，OC=OISx 并且 OCN=OISxN
9:8	**LOCK**	寄存器写保护配置 00：无写保护 01：保护等级 1 – DTG, OISx, OISxN, BKE, BKP, AOE 不能改写 10：保护等级 2 –在等级 1 基础上，CCxP, CCxNP, OSSR,

Bit	助记符	功能描述
		OSSI 不能改写 11：保护等级 3 –在等级 2 基础上，OcxM, OcxPE 在相应通道配置为输出时不能改写 注意：LOCK 寄存器在复位后仅能改写 1 次，写保护后的寄存器只有在下一次芯片复位后才能重新写入。
7:0	**DTG**	死区时间插入，用于配置互补输出插入的死区时间长度 DTG[7:5]=0xx：DT=DTG[7:0] * t_{DTS} DTG[7:5]=10x：DT=(64+DTG[5:0]) * 2 * t_{DTS} DTG[7:5]=110：DT=(32+DTG[4:0]) * 8 * t_{DTS} DTG[7:5]=111：DT=(32+DTG[4:0]) * 16 * t_{DTS}

29.5.19 ATIM DMA 控制寄存器

名称	ATIM_DCR							
地址	0x48							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			DBL				
位权限	U-0			RW-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			DBA				
位权限	U-0			RW-00000				

Bit	助记符	功能描述
31:13	**--**	RFU：未实现，读为 0
12:8	**DBL**	DMA Burst 长度 对 ATIM_DMAR 寄存器的读写将触发 burst DMA 操作，burst 长度为 1~18 00000：长度=1 00001：长度=2 …… 10001：长度=18 其他：无效值，禁止写入
7:5	**--**	RFU：未实现，读为 0
4:0	**DBA**	DMA 基地址，定义指向寄存器的偏移地址 00000：ATIM_CR1 00001：ATIM_CR2 00010：ATIM_SMCR …… 注意：当 DBA+DBL 超出了 ATIM 寄存器地址范围，则实际 burst 传输到 ATIM 最高寄存器地址后自动停止，即 burst 长度会缩短。

29.5.20 ATIM DMA 访问寄存器

名称	ATIM_DMAR							
地址	0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DMAR[31:24]							
位权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DMAR[23:16]							
位权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DMAR[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DMAR[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:0	**DMAR**	DMA burst 访问寄存器 在使用 DMA burst 传输时，将 DMA 通道外设地址设置为 ATIM_DMAR，ATIM 会根据 DBL 的值产生多次 DMA 请求

29.5.21 ATIM 刹车输入控制寄存器

名称	ATIM_BKCTL							
地址	0x60							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						BRK2GATE	BRK1GATE
位权限	U-0						RW-1	RW-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BRKF				BRKCOMB	HFDET_BRKEN	SVD_BRKEN	COMP_BRKEN
位权限	RW-0000				RW-0	RW-0	RW-0	RW-0

Bit	助记符	功能描述
31:10	-	RFU，未实现，读为 0
9	**BRK2GATE**	ATIM_BRK2 引脚输入门控信号 0：将 ATIM_BRK2 的输入门控成 0 1：不门控
8	**BRK1GATE**	ATIM_BRK1 引脚输入门控信号 0：将 ATIM_BRK2 的输入门控成 0

Bit	助记符	功能描述
		1：不门控
7:4	**BRKF**	刹车信号的滤波时钟和长度选择 0000：无滤波 0001：$f_{SAMPLING}=f_{CK_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3	**BRKCOMB**	刹车组合控制 0：两路刹车信号相或 1：两路刹车信号相与
2	**HFDET_BRKEN**	XTHF 停振检测刹车信号使能 0：禁止 HFDET 刹车信号 1：使能 HFDET 刹车信号
1	**SVD_BRKEN**	SVD 刹车信号使能 0：禁止 SVD 刹车信号 1：使能 SVD 刹车信号
0	**COMP_BRKEN**	比较器输出刹车信号使能 0：禁止比较器刹车信号 1：使能比较器刹车信号

30 通用定时器（GPTIM）

30.1 概述

芯片包含3个通用定时器。

通用定时器包含一个16bit自动重载计数器及一个可编程预分频器。

通用定时器可以支持多种应用，包括如捕捉、输出比较、PWM。

30.2 主要特性

- 16bit向上、向下、双向计数自动重载计数器
- 16bit可编程预分频器，支持实时调整计数时钟分频
- 4个独立通道可用于输入捕捉、输出比较、PWM（边缘或中心对齐模式）、单脉冲输出
- 支持与其他定时器级联
- 支持在以下事件发生时产生中断
  - 计数器溢出，计数器初始化（软件或硬件 trigger）
  - Trigger 事件（计数器启动、停止、初始化、内外部触发）
  - 输入捕捉
  - 输出比较

30.3 结构框图

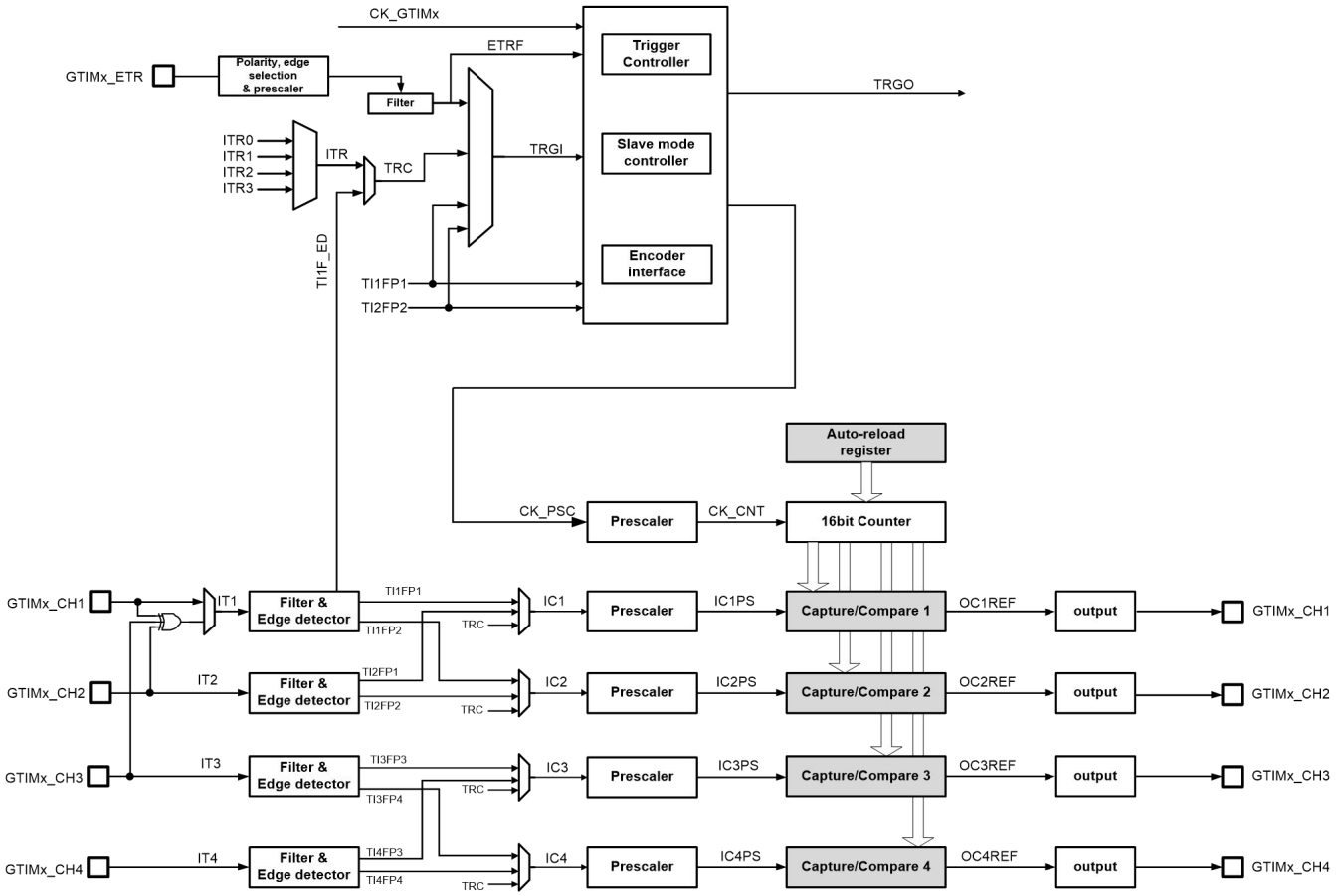


图30-1通用定时器架构示意图

30.4 功能描述

30.4.1 定时单元

高级定时器的定时单元由一个16位计数器和自动重载寄存器组成。计数器可以向上、向下或双向计数。计数时钟可以通过16位预分频器对APBCLK进行分频后得到。

计数器、自动重载寄存器预分频寄存器都可以由软件改写或读取，即使在计数器正在运行时也是如此。

定时单元包含如下寄存器：

- 计数器（GPTIM_CNT）
- 预分频寄存器（GPTIM_PSC）
- 自动重载寄存器（GPTIM_ARR）

ARR包含预装载功能，该功能通过ARPE（Auto Reload Preload Enable）寄存器控制。当ARPE=0时，对ARR寄存器执行写入，写入数据将直接传入到影子寄存器；当ARPE=1时，对ARR寄存器执行写入的数据在update event（GPTIM_CNT上溢出或者下溢出）发生时，传送到影子寄存器。软件也可以通过寄存器操作主动触发ARR更新（UEV）。

GPTIM_CNT工作时钟由GPTIM_PSC产生的分频时钟驱动，只有在计数器使能寄存器（CEN）置位时，CNT才开始计数。当CNT=ARR时，本轮计数结束，发送update event。

GPTIM_PSC是一个同步预分频器，能够对APBCLK进行1~65536分频。PSC寄存器同样被缓存，改写PSC实际不改写影子寄存器，只有当新的update event到来时，才会从PSC更新至影子寄存器。因此在CNT计数过程中，软件可以实时改写PSC，而新的预分频比将在下一更新事件发生时被采用。

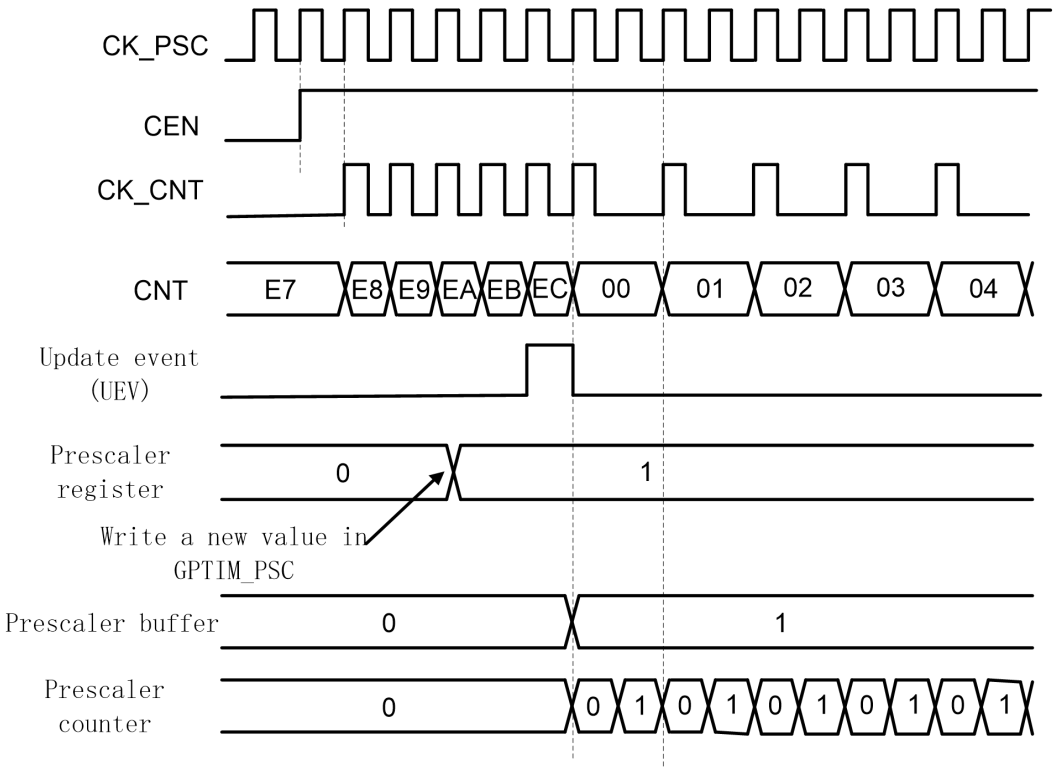


图 30-2 预分频从 1 变为 2 的波形

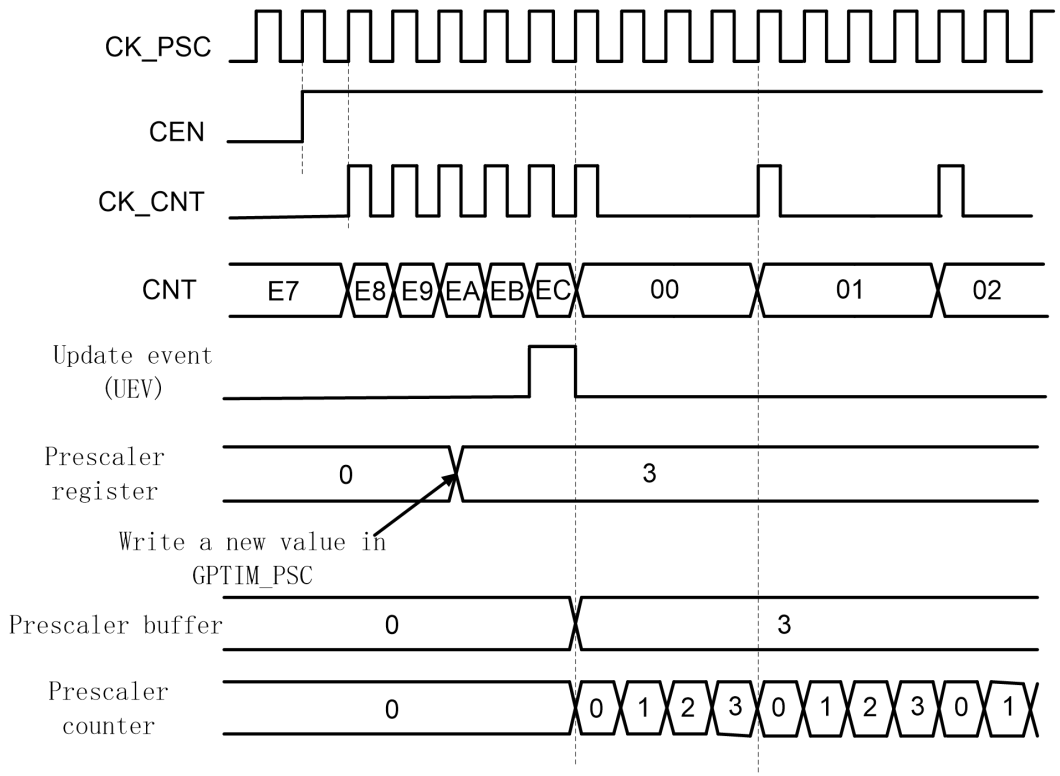


图 30-3 预分频从 1 变为 4 的波形

30.4.2 定时器工作模式

定时器支持向上计数、向下计数和中心计数模式。

向上计数

此模式中，计数器使能后从0开始计数，直到CNT=ARR，产生溢出事件，然后重新从0开始计数。

软件可以通过设置UG寄存器直接触发update event，此时CNT和预分频计数器自动清零。设置UG寄存器是否触发UIF（Update Interrupt Flag）中断标志置位由URS寄存器的设置决定。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- ARR影子寄存器被更新为GPTIM_ARR内容
- PSC影子寄存器被更新为GPTIM_PSC内容

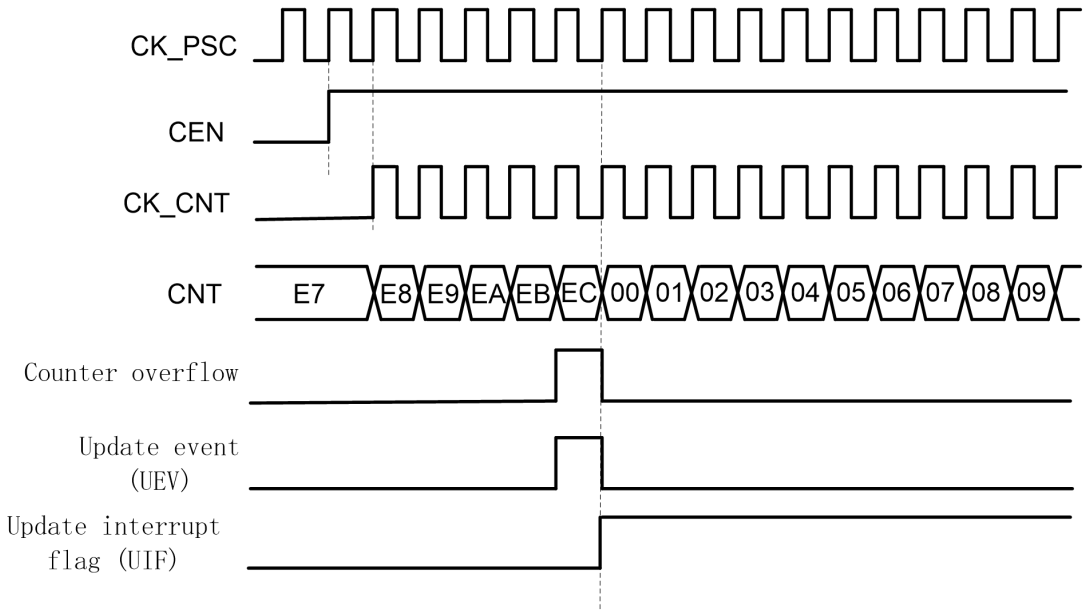


图 30-4 向上计数波形，内部时钟不分频

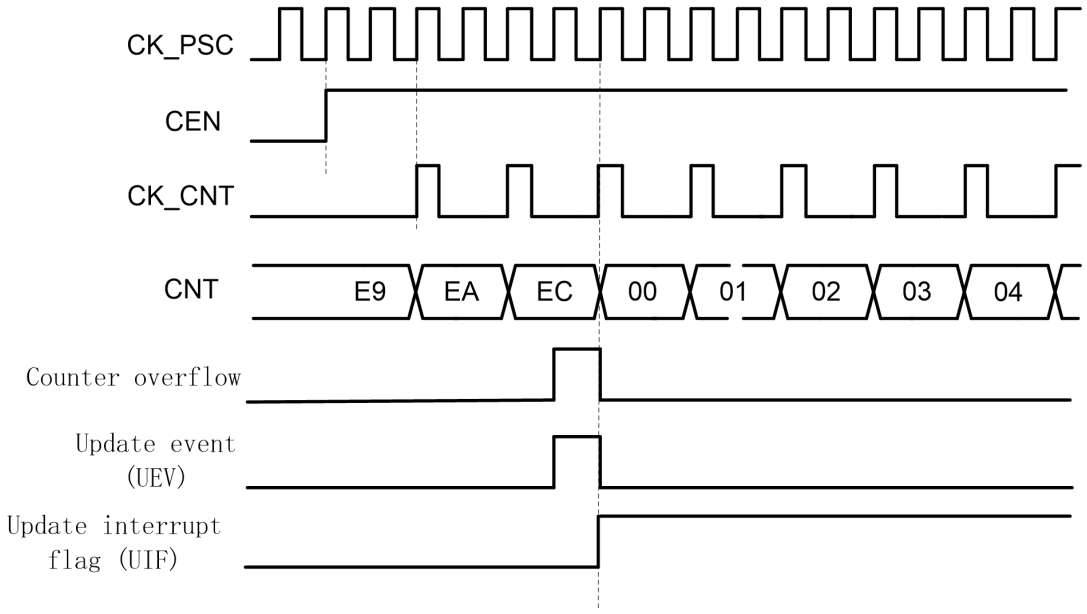


图 30-5 向上计数波形，内部时钟 2 分频

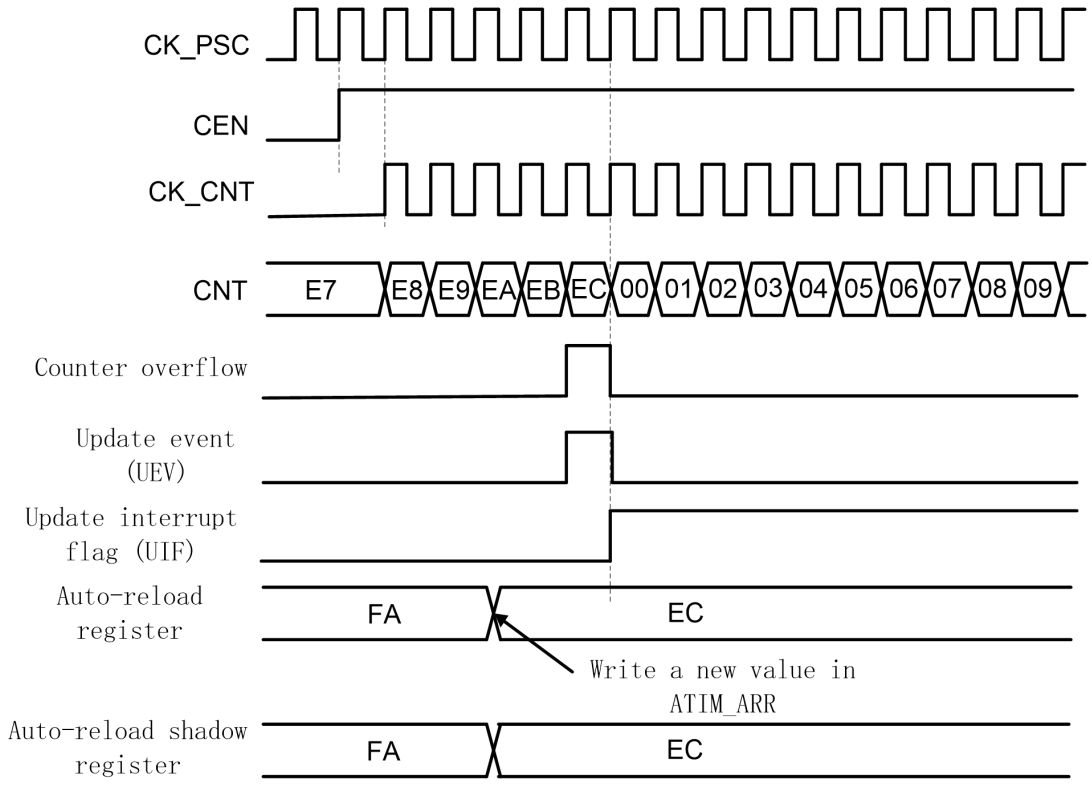


图 30-6 ARPE=0（GPTIM_ARR 没有预装载）时的更新事件

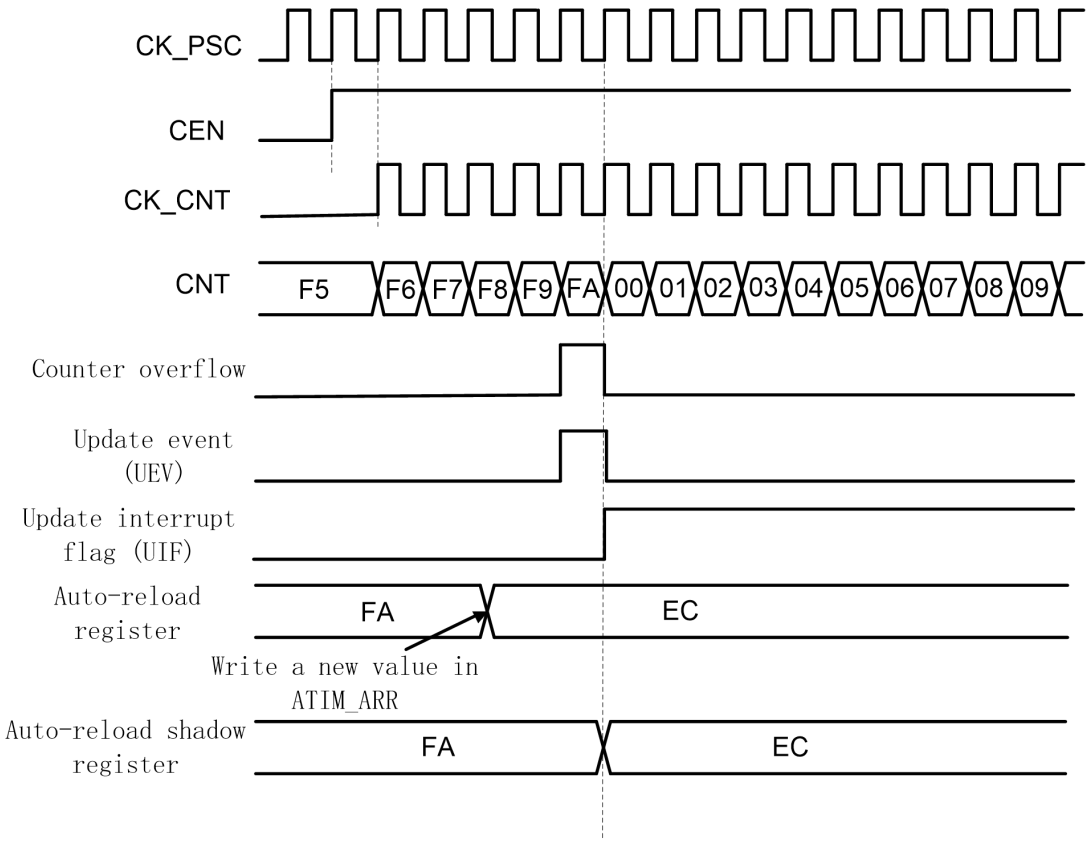


图 30-7 ARPE=1（GPTIM_ARR 预装载）时的更新事件

向下计数

向下计数模式中，计数器从ARR值开始递减，到0后产生下溢出事件，并且重新从ARR开始计数。

软件可以通过设置UG寄存器直接触发update event，此时CNT和预分频计数器自动清零。设置UG寄存器是否触发UIF（Update Interrupt Flag）中断标志置位由URS寄存器的设置决定。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- ARR影子寄存器被更新为GPTIM_ARR内容
- PSC影子寄存器被更新为GPTIM_PSC内容

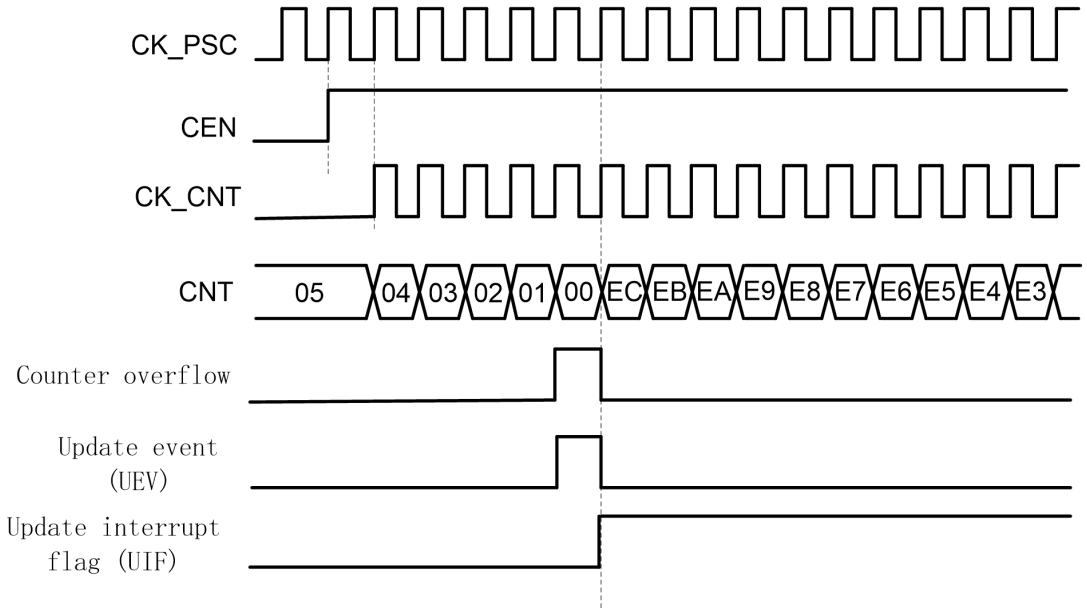


图 30-8 向下计数，内部时钟不分频

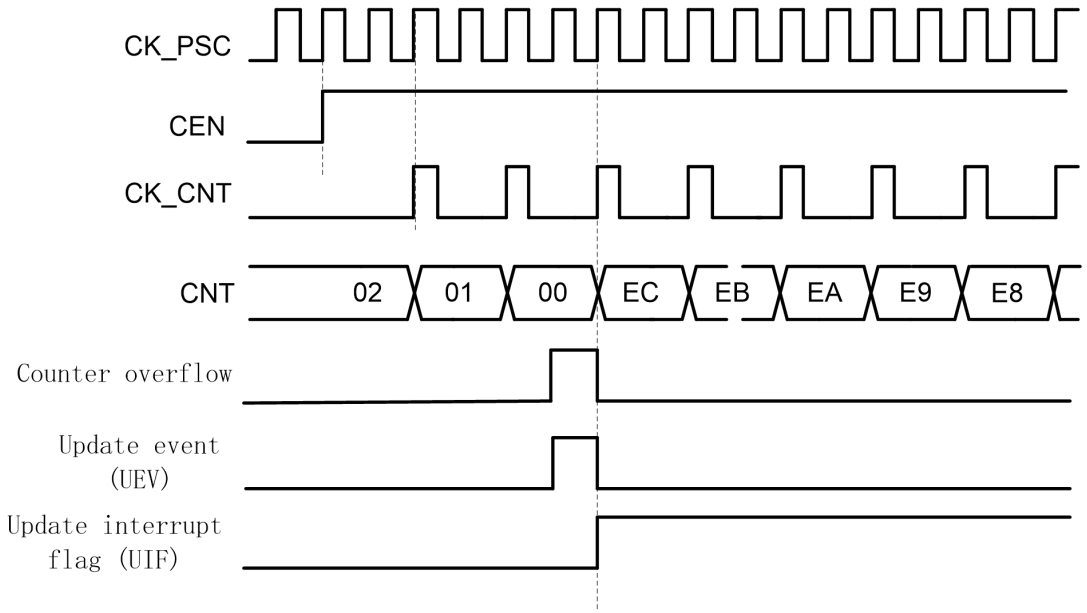


图 30-9 向下计数，内部时钟 2 分频

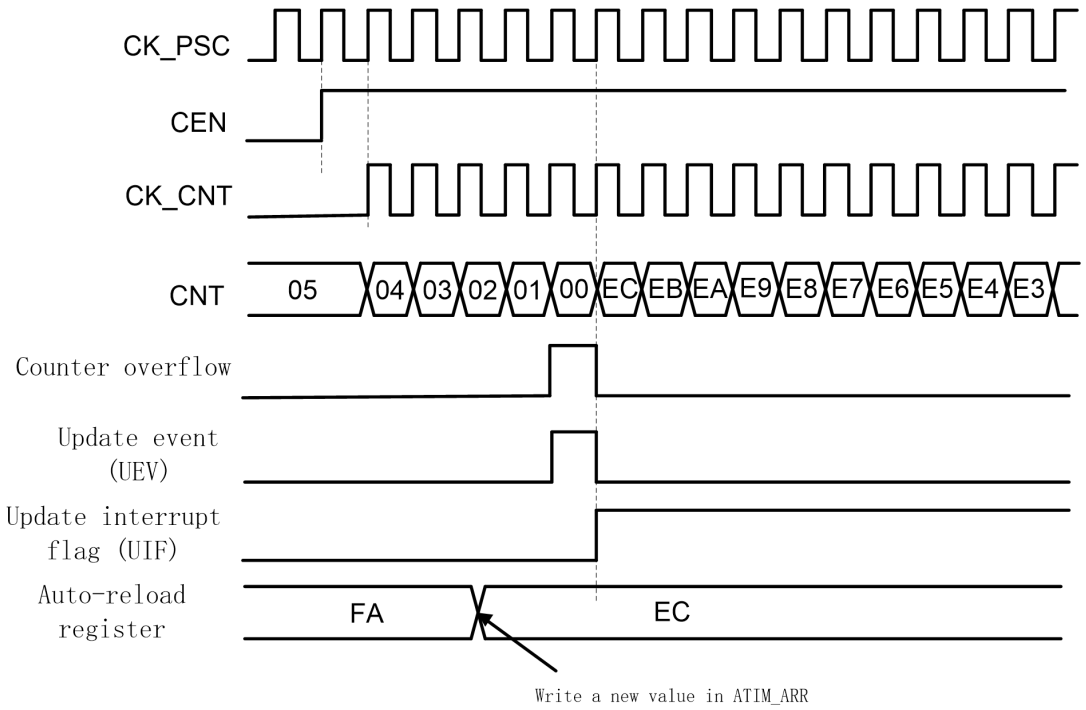


图 30-10 向下计数，内部时钟 2 分频

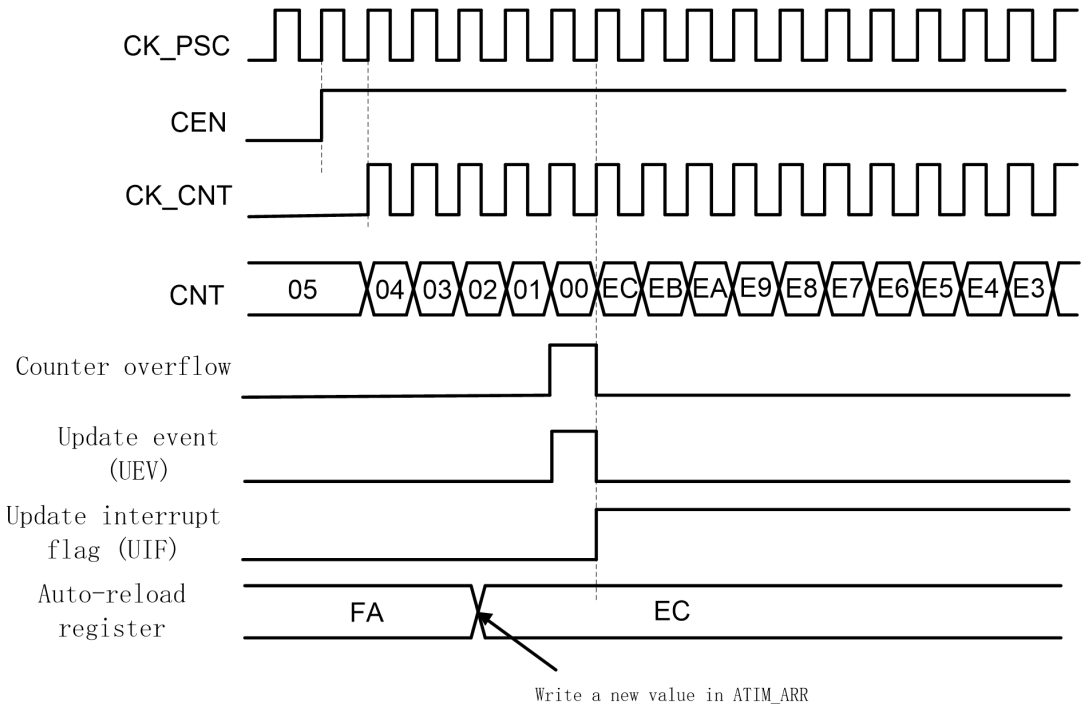


图 30-11 向下计数，不使用重复计数时的更新事件

中心对齐计数

在中心对齐模式下，计数器从0开始向上计数，到ARR-1产生上溢出事件，然后从ARR开始向下计数到1，产生下溢出事件，再从0重新开始向上计数。

CMS[1:0]寄存器用于使能中心对齐模式，并选择中心对齐模式下的输出比较工作方式。当CMS!=00时为中心对齐计数，当CMS=01时，输出比较功能仅在向下计数时有效，当CMS=10时，输出比较功能仅在向上计数时有效，当CMS=11时，输出比较功能在上下计数时都有效。

中心对齐模式下，DIR寄存器无法由软件改写，而是随着计数方向变化硬件自动更新，表示当前计数方向。

计数器在overflow和underflow的事件上都会更新 ARR、PSC的影子寄存器。

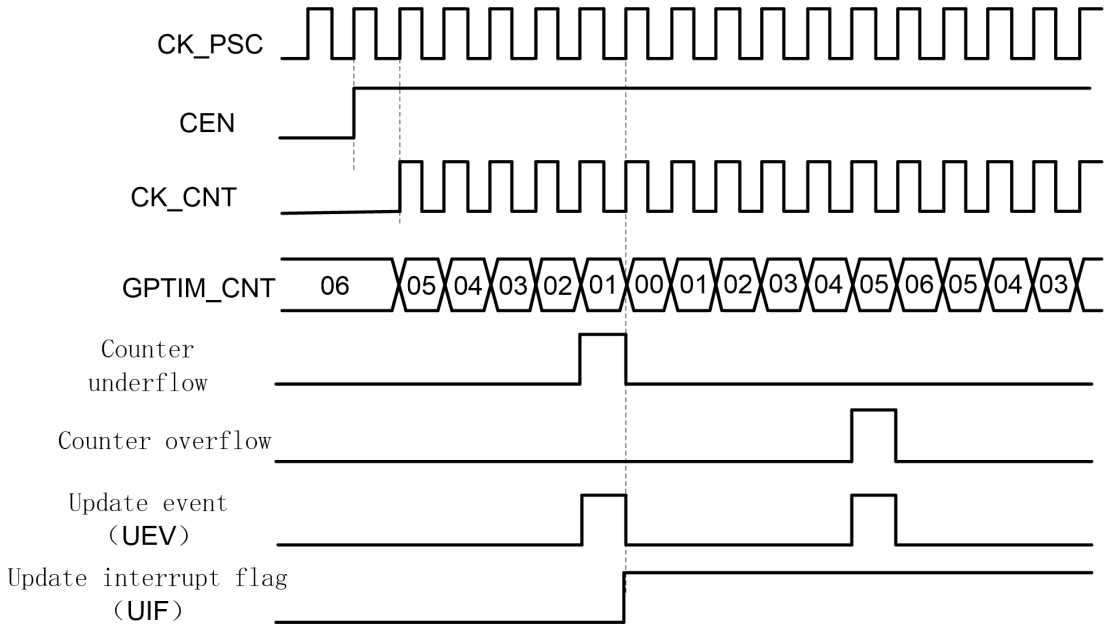


图 30-12 中心对齐计数器时序图，GPTIM_PCS=0，GPTIM_ARR=0x6

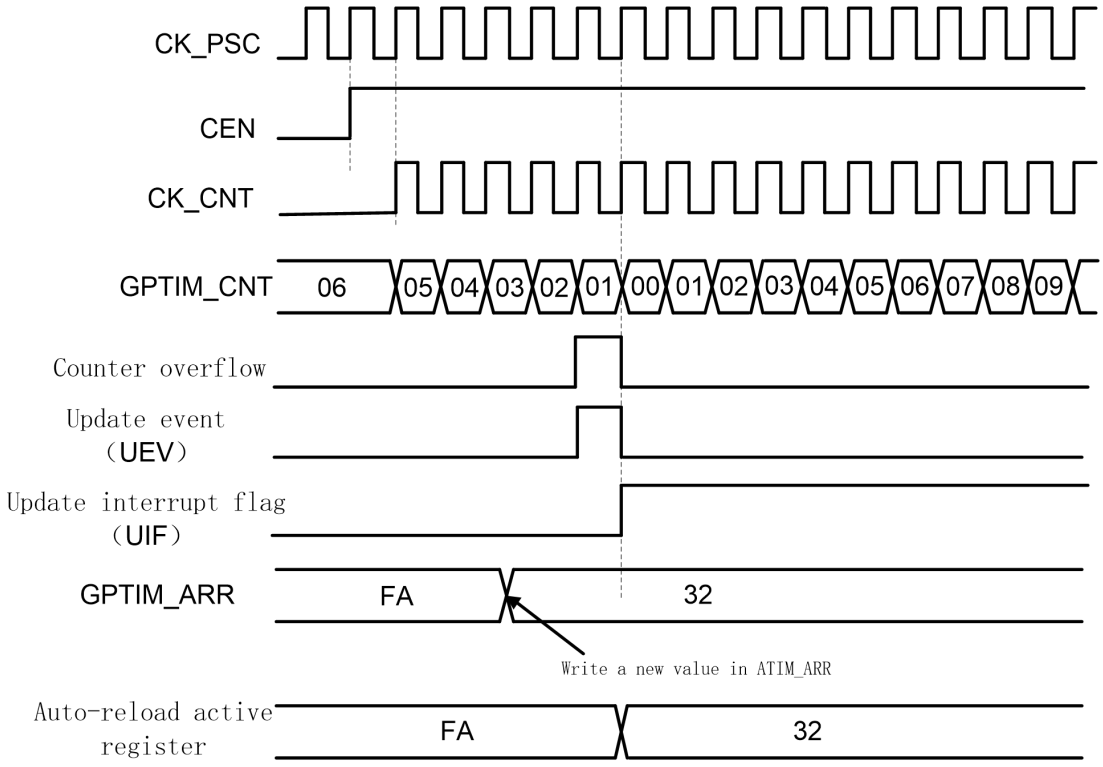


图 30-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

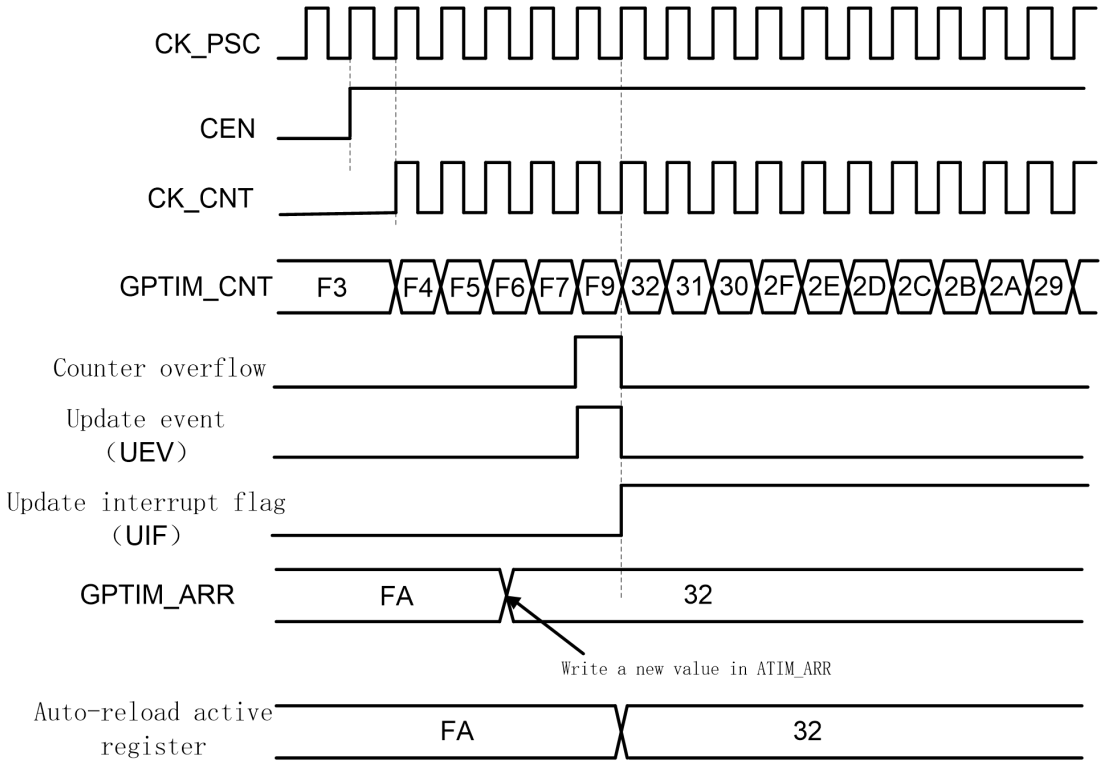


图 30-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

30.4.3 计数器工作时钟

计数器可以使用如下时钟工作：

- APBCLK——内部时钟模式
- 外部引脚输入时钟（Tix）——外部时钟模式1
- 外部引脚触发输入（ETR）——外部时钟模式2
- 内部触发（ITRx）——使用一个timer的触发输出（TGO）作为计数时钟

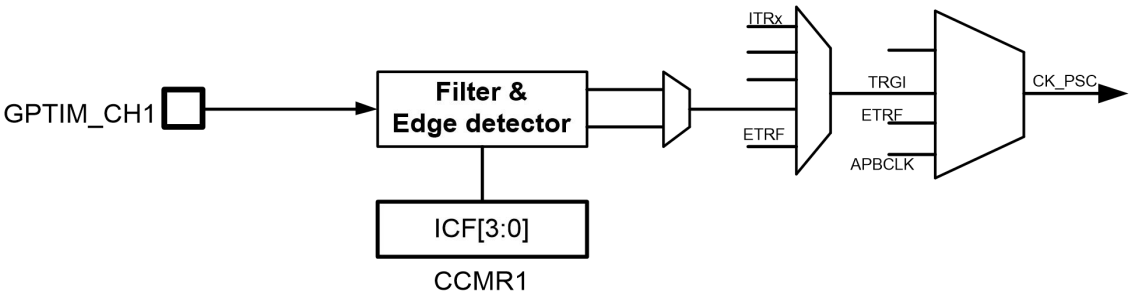


图 30-15 GPTIM 时钟源框图

30.4.3.1 内部时钟模式

内部时钟模式下，禁止从机模式（SMS=000），CEN、DIR、UG等寄存器位都是软件控制

软件操作UG寄存器后，update信号经过CLK_PSC同步后，计数器值将被重新初始化。

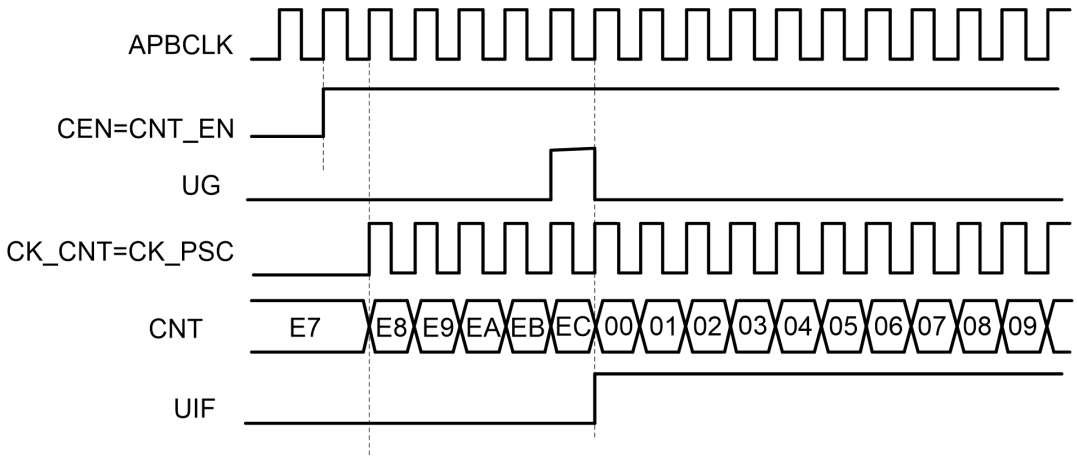


图 30-16 内部时钟源模式，时钟分频因子为 1

30.4.3.2 外部时钟模式 1

此模式下直接使用外部引脚输入信号作为计数时钟，配置SMS=111，计数边沿可以配置为上升或下

降沿。

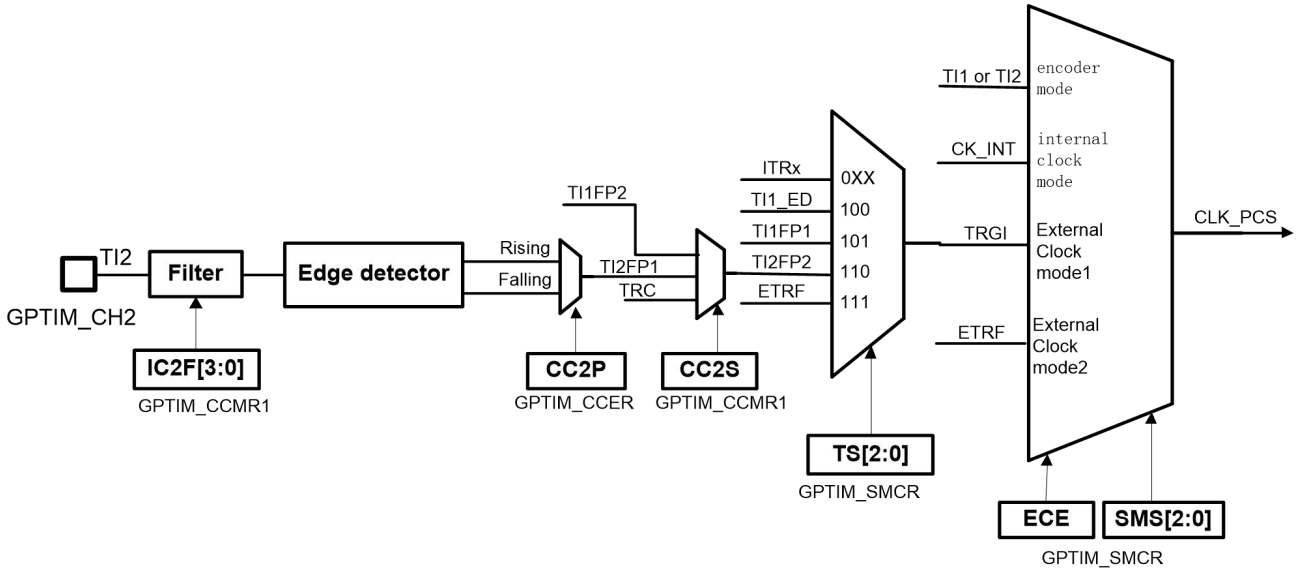


图 30-17TI2 外部时钟连接例子

外部输入信号在触发计数器计数前，会先经过内部时钟的同步过程，同时输入信号的有效沿会触发TIF标志

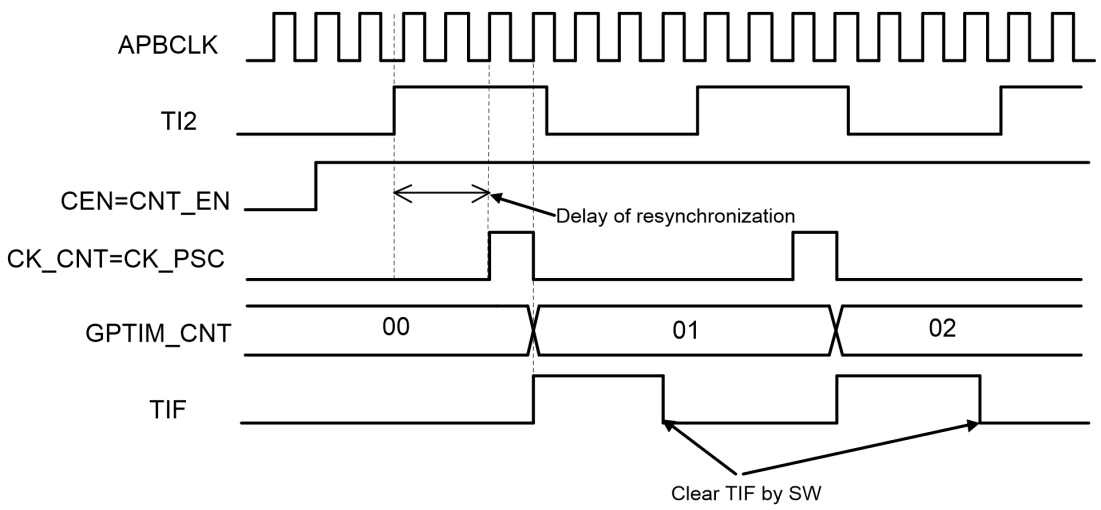


图 30-18 外部时钟模式 1 下的时序

使用外部时钟计数时，仍然要使能GPTIM的内部时钟（APBCLK），因为GPTIM要使用APB_CLK来对外部输入时钟进行同步和滤波。在外部时钟模式1下，外部输入时钟首先经过滤波和边沿选择，得到有效的计数沿，作为有效工作时钟（CLK_PSC）输入给预分频模块。

外部时钟同步采用简单的2级触发器结构，因此为了避免亚稳态，要求外部输入时钟宽度至少大于2个APB_CLK周期。

此模式下只有通道1和2的输入可以用做时钟输入，所需配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH2功能
- 关闭通道使能，配置GPTIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC2S=01, IC2映射到TI2
- 选择计数有效沿，配置GPTIM_CCER.CC2P=0，选择上沿或者下沿
- 配置输入滤波时间，配置GPTIM_CCMR1.IC2F[3:0](IC2F=0000，不进行输入滤波)
- 使能外部时钟模式1，配置GPTIM_SMCR.SMCR=111
- 选择触发输入源，配置GPTIM_SMCR.TS=110,选定TI2作为触发输入源
- 打开通道使能，配置GPTIM_CCER.CC2E=1
- 使能计数器，配置GPTIM_CR1.CEN=1

下图是一个典型的外部时钟计数模式1的示例：

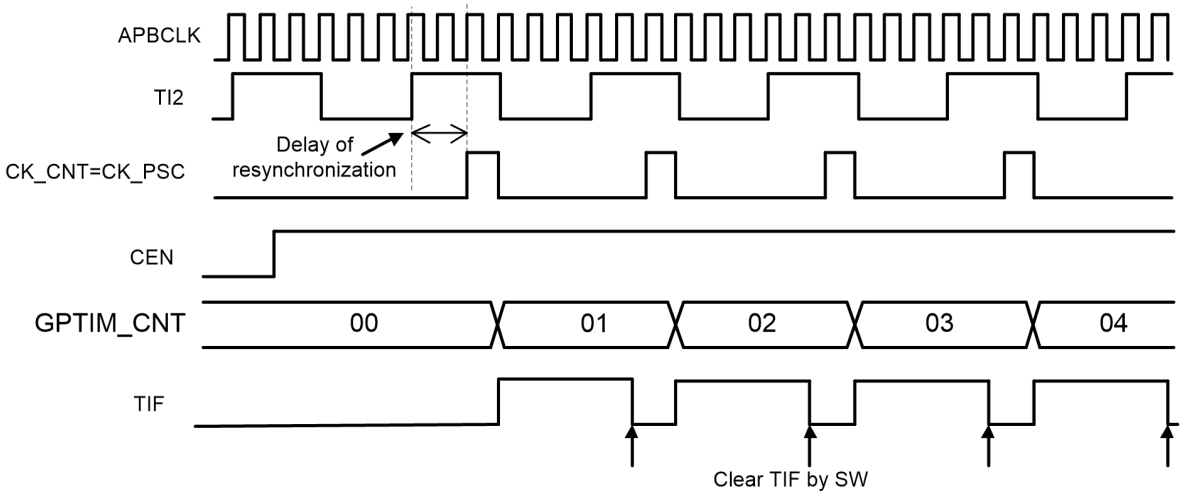


图 30-19 外部时钟模式 1 下的时序

30.4.3.3 外部时钟模式 2

此模式下使用GPTIM_ETR管脚输入信号的上升沿或下降沿（不支持双沿）来计数。

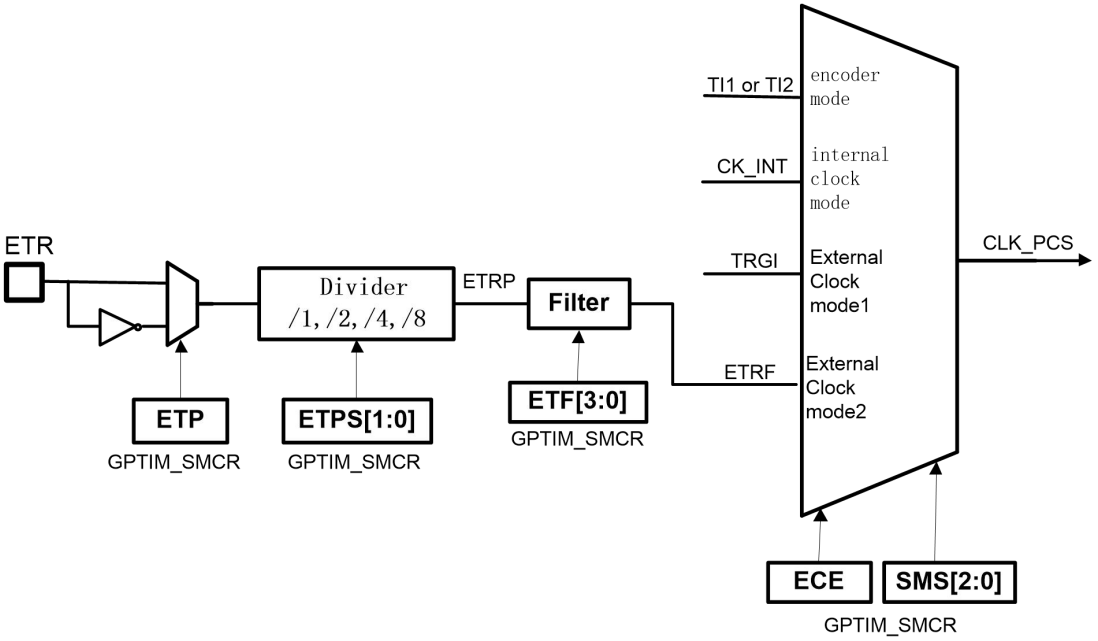


图 30-20 外部触发输入框图

下图是使用ETR二分频后的上升沿进行计数，其中实际计数发生时间因为内部时钟的同步过程而延迟于ETR输入上升沿。

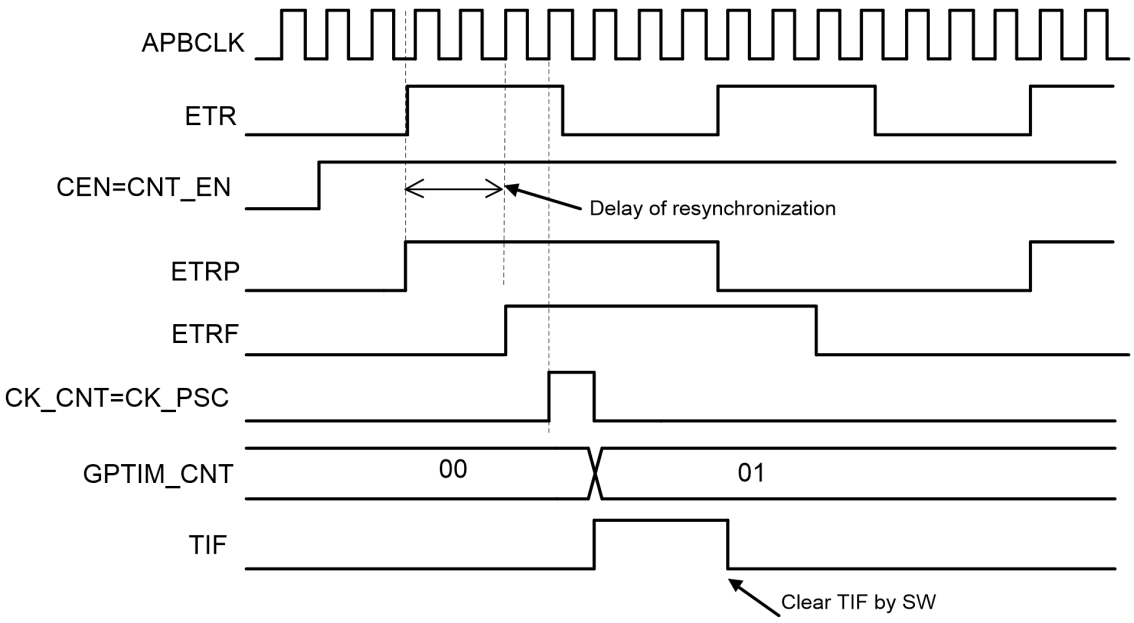


图 30-21 外部时钟模式 2 下的时序 1

与外部时钟模式1的主要差别是，ETR输入直接被分频后再进行滤波，产生CK_PSC时钟，这意味着可以支持ETR输入频率高于APB_CLK的应用场景，这种情况下，需要首先对ETR输入进行预分频，再用于驱动计数器。

此模式所需配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_ETR功能
- 设置ETP进行沿选择，GPTIM_SMCR.ETP=0
- 设置ETR分频比，配置GPTIM_SMCR.ETPS[1:0]=01
- 配置输入滤波时间，GPTIM_SMCR.ETF[3:0]=0000
- 置位ECE寄存器，使能外部时钟模式2,GPTIM_SMCR.ECE=1，GPTIM_SMCR.SMS=000
- 使能计数器，配置GPTIM_CR1.CEN=1

下图是一个典型的外部时钟模式2的示例：

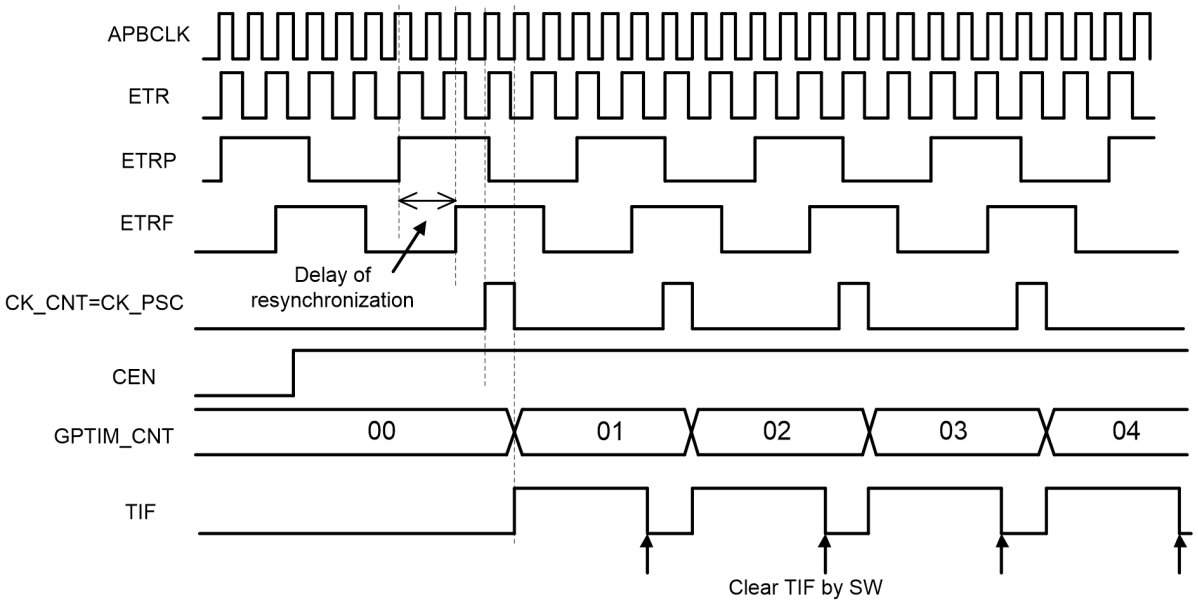
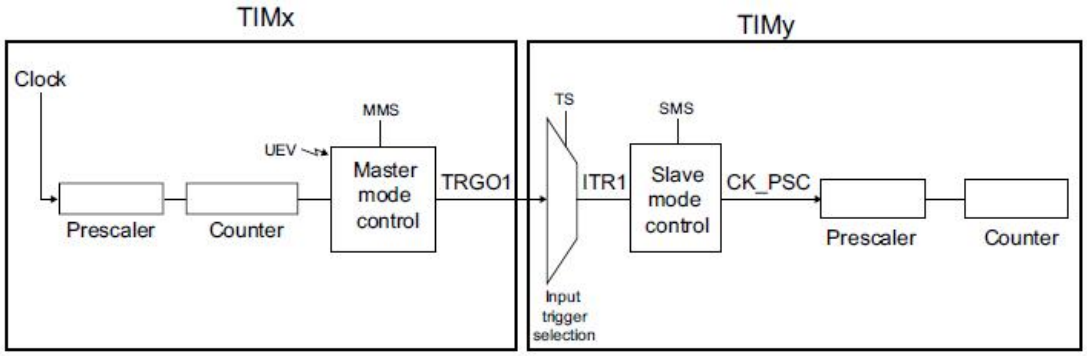


图30-22外部时钟模式2下的时序2

在使用外部时钟模式2时，仍可以将GPTIM配置为slave模式：比如使用ETR输入计数，同时使用另一个Timer的TRGO作为触发信号，当触发事件到来时，复位计数器重新开始计数。

30.4.3.4 内部触发模式

每个GPTIM支持4个ITR输入，可用于计数触发或者内部信号捕捉。当ITR选择为计数触发信号时，GPTIM计数器将在每个ITR信号的高电平计数。通过内部触发模式可以实现Timer级联，下图是一个例子：



配置TIMx为master mode并周期性输出TRGO脉冲信号，TIMy配置为Slave mode并将TIMx的TRGO设置为ITR；当TIMx.TRGO脉冲到来时，TIMy计数一次。

基于内部触发模式的timer级联有如下要求：

- TRGO信号设计为APBCLK单周期脉冲
- TIMx和TIMy都工作在APBCLK时钟域
- TRGO对于接收方来说是一个同步脉冲
- Master和Slave的工作时钟都必须使能

内部触发模式可以使用的触发信号除了其他定时器输出外，还可以是ADC_EOC或者比较器输出，为了满足以上要求，需要设计时将ADC和COMP输出的trigger信号处理成APBCLK同步脉冲。

30.4.4 内部触发信号（ITRx）的捕捉

每个GPTIM支持4个ITR输入，可用于计数触发或者内部信号捕捉。当用于内部信号捕捉时，需要将TS配置为000~011用于选择ITR0~ITR3，并将CCxS配置为11，即将TRC选为捕捉信号。通过这个方法，Timer可以捕捉各种芯片内部信号的周期或电平宽度。

每个ITR输入支持4个内部信号扩展，由ITRxSEL寄存器配置。输入信号源参考下表：

GPTIM0			Function
ITR0SEL	00	ATIM_TRGO	计数触发
	01	UART0_RX	宽度捕捉
	10	UART1_RX	宽度捕捉
	11	UART3_RX	宽度捕捉
ITR1SEL	00	GPTIM2_TRGO	计数触发
	01	XTHF	周期捕捉
	10	RCHF	周期捕捉
	11	LPUART1_RX	周期捕捉
ITR2SEL	00	BSTIM32_TRGO	计数触发
	01	LPUART2_RX	宽度捕捉
	10	RCLP	周期捕捉
	11	XTLF	周期捕捉
ITR3SEL	00	COMP1_TRGO	计数触发
	01	RCLF	周期捕捉
	10	COMP2_TRGO	计数触发
	11	LPT32_TRGO	计数触发
GPTIM1			**Function**
ITR0SEL	00	ATIM_TRGO	计数触发
	01	UART0_RX	宽度捕捉
	10	UART1_RX	宽度捕捉
	11	UART3_RX	宽度捕捉
ITR1SEL	00	GPTIM0_TRGO	计数触发
	01	LUT1_TRGO	周期捕捉
	10	RCHF	周期捕捉
	11	ADC_EOC_TRGO	计数触发
ITR2SEL	00	BSTIM32_TRGO	计数触发
	01	LSCLK	周期捕捉
	10	RCLP	周期捕捉
	11	XTLF	周期捕捉
ITR3SEL	00	COMP1_TRGO	计数触发
	01	LUT3_TRGO	周期捕捉
	10	COMP2_TRGO	计数触发
	11	LPT32_TRGO	计数触发
GPTIM2			**Function**
ITR0SEL	00	ATIM_TRGO	计数触发
	01	UART3_RX	宽度捕捉
	10	UART4_RX	宽度捕捉
	11	LUT0_TRGO	宽度捕捉
ITR1SEL	00	GPTIM1_TRGO	计数触发
	01	XTHF	周期捕捉

	10	RCHF	周期捕捉
	11	ADC_EOC_TRGO	计数触发
ITR2SEL	00	BSTIM16_TRGO	计数触发
	01	LSCLK	周期捕捉
	10	RCLP	周期捕捉
	11	XTLF	周期捕捉
ITR3SEL	00	COMP1_TRGO	计数触发
	01	LUT2_TRGO	周期捕捉
	10	COMP2_TRGO	计数触发
	11	LPT16_TRGO	计数触发

软件应保证选择正确的信号用于正确的功能，错误的配置将导致完全错误的结果。比如将ATIM_TRGO用于宽度捕捉，则结果没有意义。

30.4.5 捕捉/比较通道

GPTIM包含4个捕捉/比较通道，每个通道由一个捕捉比较寄存器（CCR）（包含影子寄存器）、一个捕捉输入级、一个比较输出级组成。

输入级电路会采样Tix输入并产生滤波后的信号TixF，然后边沿检测和极性选择产生对应的TixFPx信号，此信号可作为计数触发或者待捕捉信号，并且在被捕捉前经过预分频。

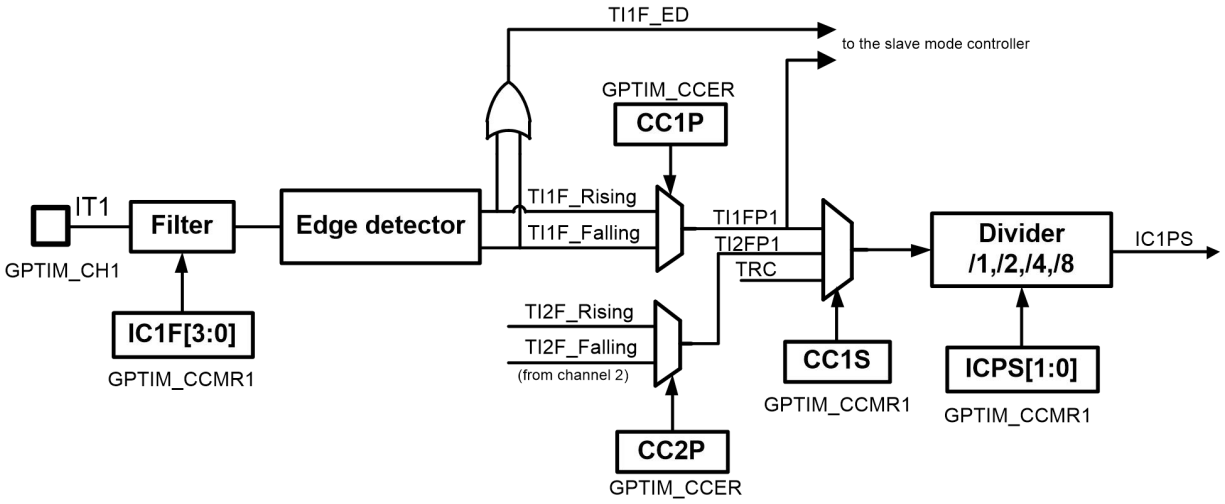


图30-23捕获/比较通道(通道1输入部分)

输出级电路会产生一个输出基准信号OCxREF，此信号固定为高电平有效，作为最终输出电路的参考输入。GPTIM输出通道不支持互补输出。

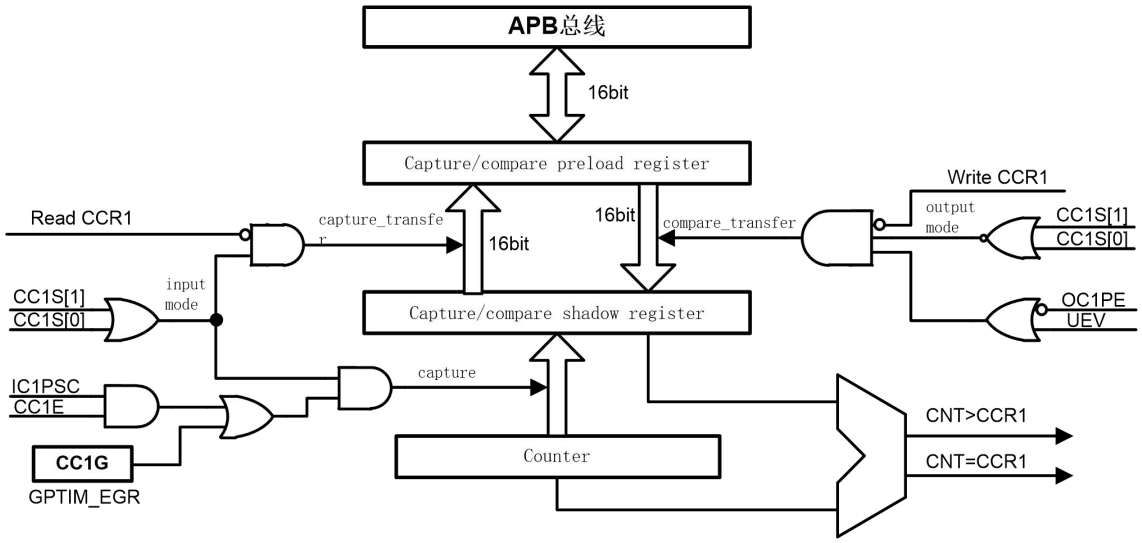


图30-24捕获/比较通道1的主电路

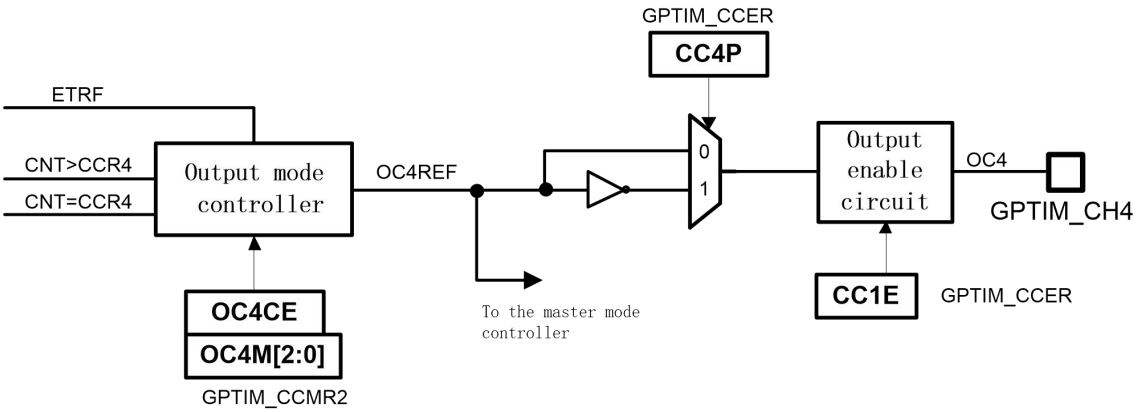


图30-25捕获/比较通道的输出部分

捕捉/比较寄存器（CCR）包含preload寄存器和shadow寄存器，软件读写总是访问preload寄存器。在捕捉模式下，捕捉值保存在shadow寄存器中并复制到preload寄存器。在比较模式下，preload寄存器的值被拷贝到shadow寄存器用来与计数器比较。

30.4.6 输入捕捉模式

当Icx信号上出现预期的电平变换，将触发一次capture，当前计数器值被锁存进CCR，与此同时，CcxIF中断标志置位，并且可以触发对应的中断或者DMA请求。如果一个捕捉事件在CcxIF为高的情况下出现，则捕捉数据冲突标志（CcxOF, Over-Capture）置位（CCR中上次捕捉值被覆盖）。CcxIF可以由软件清零，或者通过读取CCR寄存器自动清零。CcxOF标志通过软件写1清零。

通过两个或更多通道配合，可以实现PWM信号的输入捕捉。比如要计算一个输入信号的周期和占空比，可以将此信号从TI1引脚输入，芯片内部将滤波后的信号取上升沿得到TI1FP1，将滤波后的信

号取下降沿得到TI1FP2，将TI1FP1输入给捕捉通道1，将TI1FP2输入给捕捉通道2，即可实现通道1对输入信号上升沿捕捉，同时通道2对输入信号下降沿捕捉；捕捉中断定期发生后，软件通过CCR1和CCR2寄存器的值，即可计算输入信号的周期和占空比。

实现在TI1输入的上升沿捕获计数器的值到GPTIM_CCR1寄存器，配置步骤如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0，确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01, IC1映射到TI1
- 选择计数有效沿，配置GPTIM_CCER.CC1P，选择上沿或者下沿
- 配置输入滤波时间，配置GPTIM_CCMR1.IC1F[3:0]
- 配置输入预分频器，配置GPTIM_CCMR1.IC1PS[1:0]
- 打开通道使能，配置GPTIM_CCER.CC1E=1

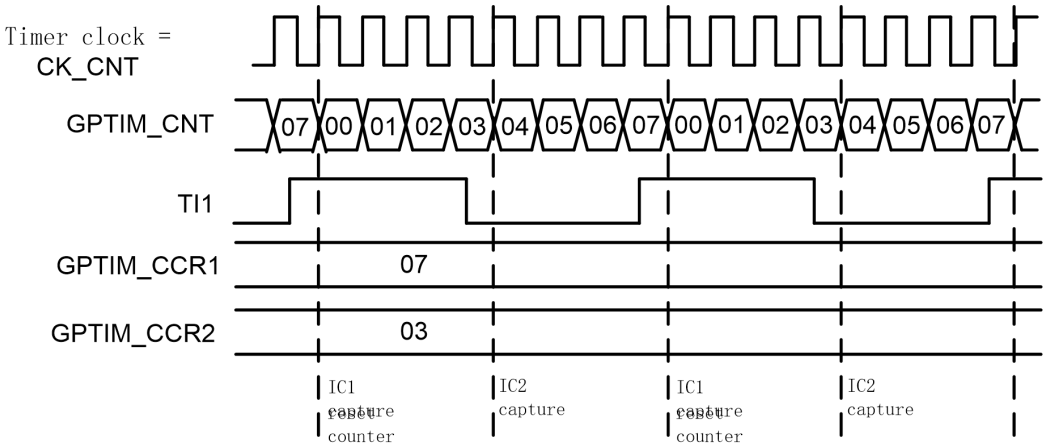


图30-26PWM输入捕获模式时序

若想实现PWM输入捕获功能，需进行如下设置：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0，GPTIM_CCER.CC2E=0确保之后通道配置成功
- 选择输入通道，两个通道IC1,IC2被映射到同一个TI1输入口，配置GPTIM_CCMR1.CC1S=01, GPTIM_CCMR1.CC2S=10
- 选择计数有效沿，两个通道IC1,IC2有效沿极性相反，配置GPTIM_CCER.CC1P=0，GPTIM_CCER.CC2P=1
- 配置输入滤波时间，配置GPTIM_CCMR1.IC1F[3:0]，GPTIM_CCMR1.IC2F[3:0]
- 配置输入预分频器，配置GPTIM_CCMR1.IC1PS[1:0] ，GPTIM_CCMR1.IC2PS[1:0]
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=101
- 设定从模式控制器为复位模式，配置GPTIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置GPTIM_CCER.CC1E=1，GPTIM_CCER.CC2E=1

30.4.7 软件 Force 输出

在比较输出模式下，软件可以直接将OCxREF force成特定电平，而独立于CCR和计数器的比较结果。

软件通过写OcxM=101寄存器，可以直接将OCxREF强制为有效（OCxREF固定为高有效），通过写OcxM=100可以直接将OCxREF强制为无效（低电平）。但是软件force操作不会取消比较过程，CCR和计数器的比较还会一直进行。

30.4.8 输出比较模式

输出比较模式下，当CCR与计数器值相等，OCxREF可以被置位成有效、无效、或电平翻转。同时，中断标志也会置位，DMA请求可以发送。

输出比较也可以被用于输出一个特定宽度的脉冲信号（单次输出）。

使用步骤：

1、选择计数时钟（内部、外部、预分频等）

2、向ARR和CCR寄存器写入期望数据

3、根据需要设置中断使能和DMA使能

4、选择输出模式

5、使能计数器

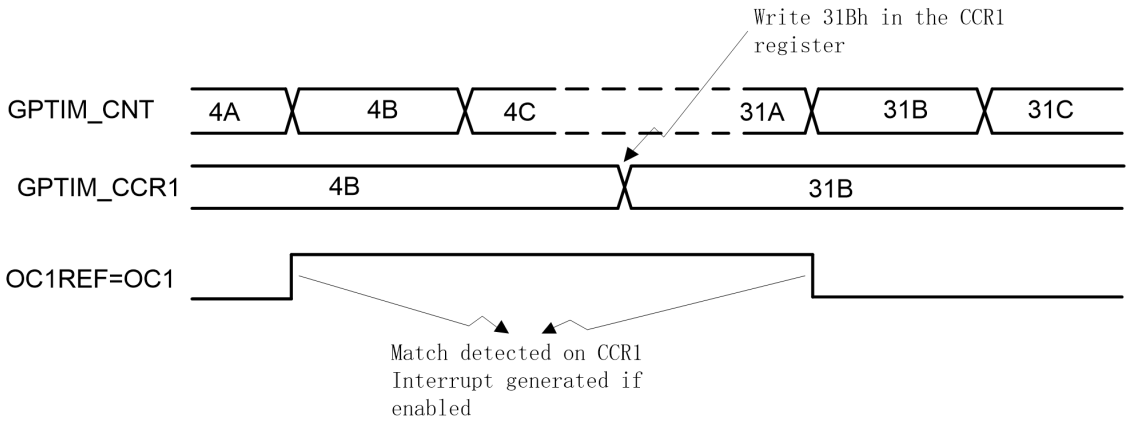


图30-27输出比较模式，翻转OC1

在不使能preload的情况下，软件可以随时改写CCR寄存器实现对输出波形的实时控制。如果使能了preload，则CCR shadow寄存器仅在下一次update event发生时更新为preload寄存器的内容。

30.4.9 PWM 输入模式

PWM模式可以输出脉宽调制信号，其周期由ARR寄存器决定，占空比由CCR寄存器决定。

输出信号的极性可以由CCxP寄存器配置。PWM模式工作中，CNT和CCR实时比较。由于计数器支持边缘对齐和中央对齐计数模式，PWM输出也支持边缘对齐和中央对齐模式。

PWM边缘对齐模式

在向上计数的情况下，配置为PWM模式1时，OCxREF信号在CNT<CCR时为高电平，否则为低电平。如果CCR值大于ARR值，则OCxREF被固定为1；如果CCR为0则OCxREF被固定为0.

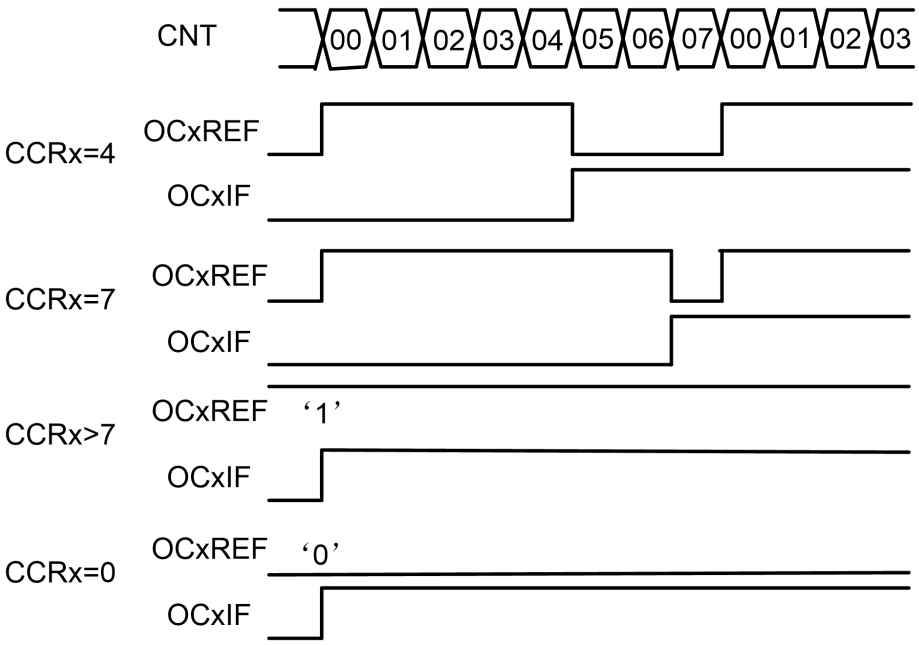


图30-28边沿对齐的PWM波形(ARR=7)

PWM中央对齐模式

OCxREF电平定义与边缘对齐模式相同。下图是一个示例：

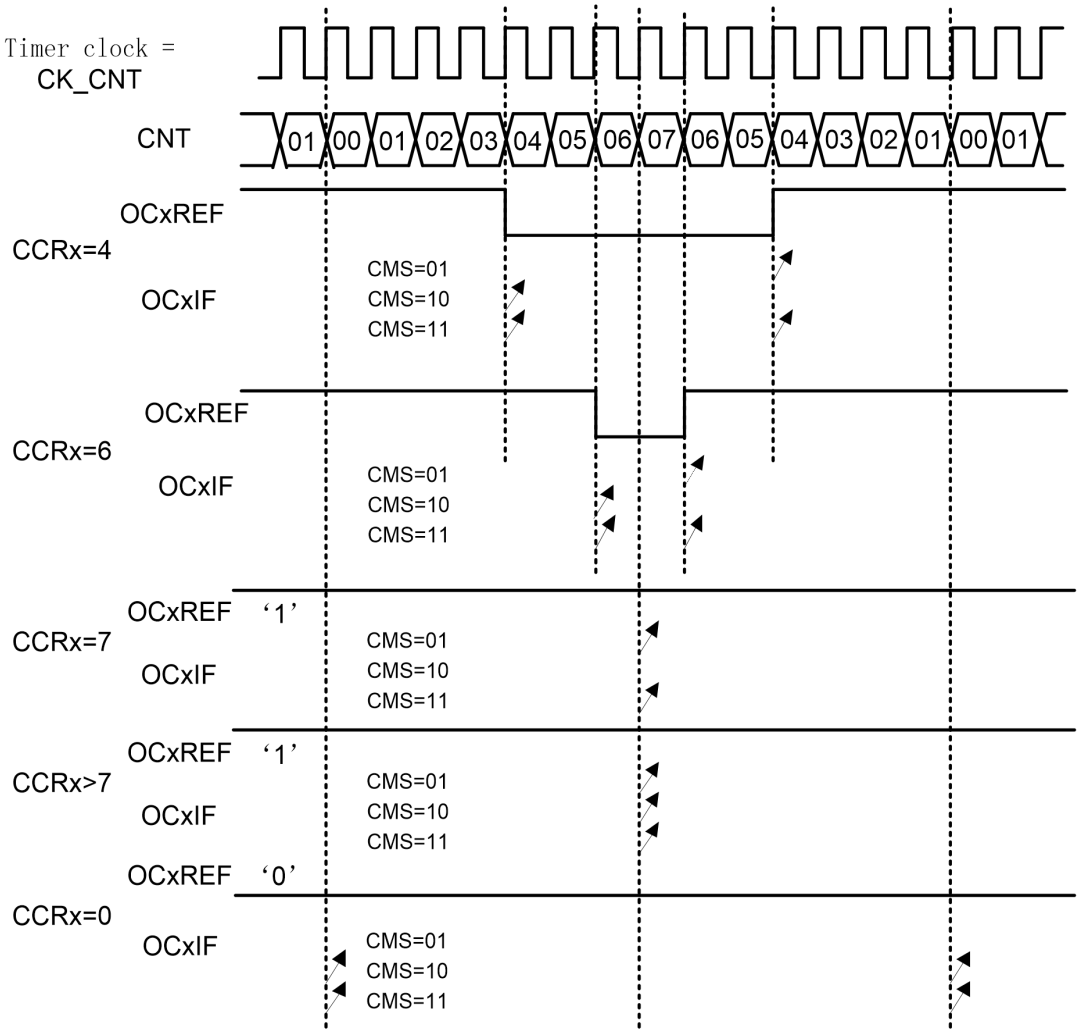


图30-29中央对齐的PWM波形(APR=7)

当启动中央对齐计数时，一开始的计数方向是由DIR寄存器决定的；随后在计数过程中，DIR寄存器的状态由硬件直接控制。安全起见，建议用户程序在启动计数器之前，通过UG寄存器做一次update，并且在计数过程中不要改写计数器。

30.4.10 单脉冲输出

单脉冲输出是比较输出模式的特殊情况，允许用户在某个事件发生后，经过可编程的延迟，输出一个可编程宽度的脉冲信号。

与其他输出模式不同的是，在下一次update event到来时，计数器会自动停止。只有当CCR和计数器初值不同时，脉冲才有可能正确输出。在向上计数时，要求CNT<CCR<=ARR，在向下计数时，要求CNT>CCR

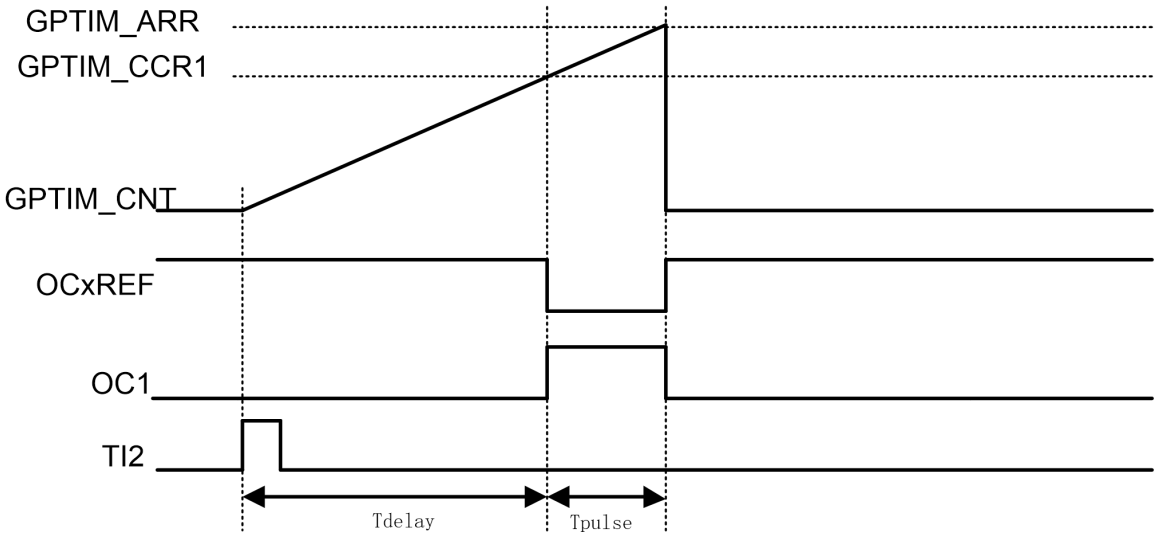


图30-30单脉冲模式的例子

上图是以TI2输入为计数器触发信号，计数值等于CCR后OCxREF输出低电平，计数到ARR后OCxREF回到高电平，并且计数器回滚到0，停止计数。

实现上述功能TI2作为输入触发的配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH2功能
- 关闭通道使能，配置GPTIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC2S=01
- 选择计数有效沿，配置GPTIM_CCER.CC2P=0
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=110，TI2FP2作为TRGI
- 设定从模式控制器为触发模式，配置GPTIM_SMCR.SMS[2:0]=110,TI2FP2用来启动计数器
- 打开通道使能，配置GPTIM_CCER.CC2E=1

实现上述功能OC1作为输出的配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0，确保之后通道配置成功
- 输出通道，配置GPTIM_CCMR1.CC1S=00
- 选择计数有效沿，配置GPTIM_ CCMR1.OC1M=111,PWM模式2
- 打开通道使能，配置GPTIM_CCER.CC1E=1

OPM波形产生时基的特殊设置：

- GPTIM_CCR1的值决定了Tdelay
- GPTIM_ARR和GPTIM_CCR1的差值决定了Tpulse（GPTIM_ARR-GPTIM_CCR1）

- 设置为单脉冲模式，配置GPTIM_CR1.OPM=1

30.4.11 外部事件清除 OCxREF

OCxREF的有效状态未高电平，通过对外部ETR引脚施加高电平，可以直接拉低OCxREF，直到下一次update event。此功能仅在输出比较和PWM模式下有效，无法在软件force模式下起作用。使能此功能需要将OcxCE置1.

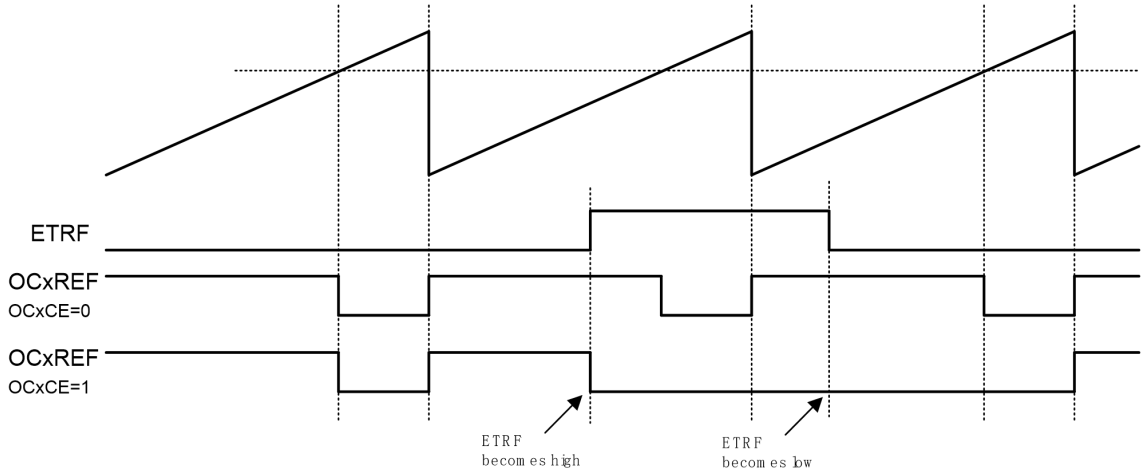


图30-31 ETR信号清除GPTIM的OCxREF

30.4.12 编码器接口模式（encoder interface）

编码器接口模式涉及到两个外部输入信号，GPTIM根据其中一个信号的边沿相对于另一个信号的电平来决定递增还是递减计数值。下表是计数方式与两路输入信号之间的关系：

有效沿	对应信号的电平（TI1 对应TI2，TI2对应TI1）	TI1信号		TI2信号	
		上升	下降	上升	下降
仅在TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在TI2处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在TI1 和TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

表30-1encoder interface计数方式

比如在计数器以TI1信号为时钟计数时，如果TI1上升沿采样到TI2为高电平，则计数器递减；如果TI1下降沿采样到TI2为高电平，则计数器递增。

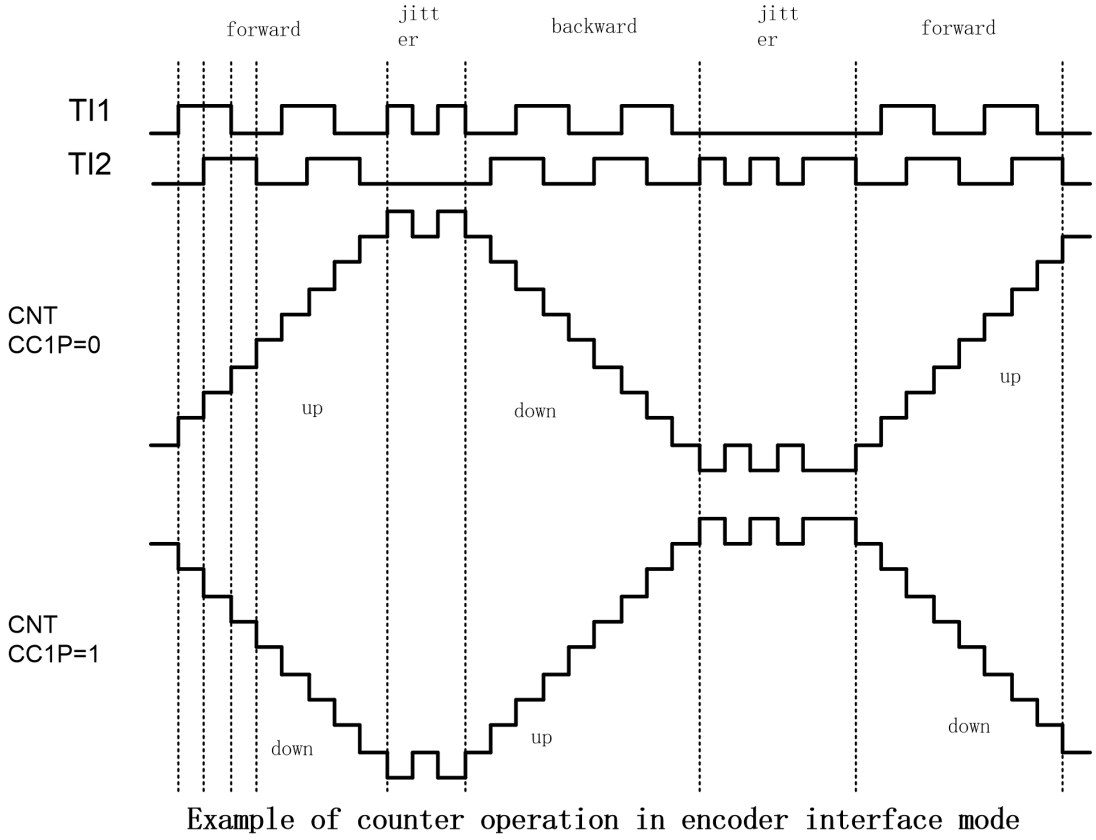


图30-32编码器模式下的计数器操作实例

编码模式输入通道需进行如下设置：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1，GPTIM_CH2功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0，GPTIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01，GPTIM_CCMR1.CC2S=01
- 选择计数有效沿，配置GPTIM_CCER.CC1P=0，GPTIM_CCER.CC2P=0
- 设定从模式控制器为编码模式3，配置GPTIM_SMCR.SMS[2:0]=011
- 打开通道使能，配置GPTIM_CCER.CC1E=1，GPTIM_CCER.CC2E=1

30.4.13 GPTIM 从机模式

GPTIM作为slave时（外部事件触发），可配置为三种工作模式：复位模式、门控模式、触发模式。

复位模式

此模式下，外部输入的事件将导致TIM内部所有preload寄存器重新初始化，CNT回到0开始计数。以下图为例，计数器正常计数，外部TI1输入上升沿时，触发计数器清零，重新开始计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置GPTIM_CCER.CC1P=0
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为复位模式，配置GPTIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置GPTIM_CCER.CC1E=1
- 使能计数器，配置GPTIM_CR1.CEN=1

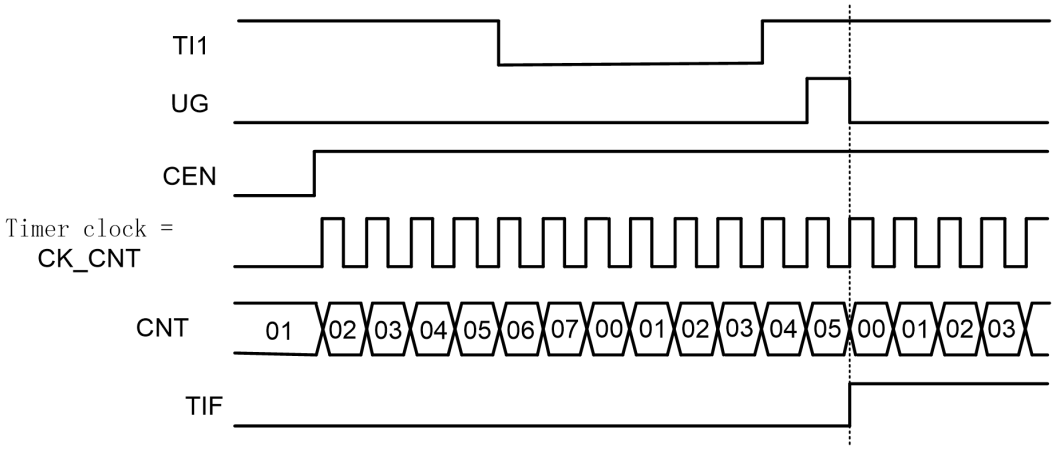


图30-33复位模式下的时序

门控模式

此模式下，计数器仅在输入信号为特定电平时工作。电平变换导致计数器开始或停止计数时，都会触发中断标志。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置GPTIM_CCER.CC1P=0
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为门控模式，配置GPTIM_SMCR.SMS[2:0]=101
- 打开通道使能，配置GPTIM_CCER.CC1E=1
- 使能计数器，配置GPTIM_CR1.CEN=1

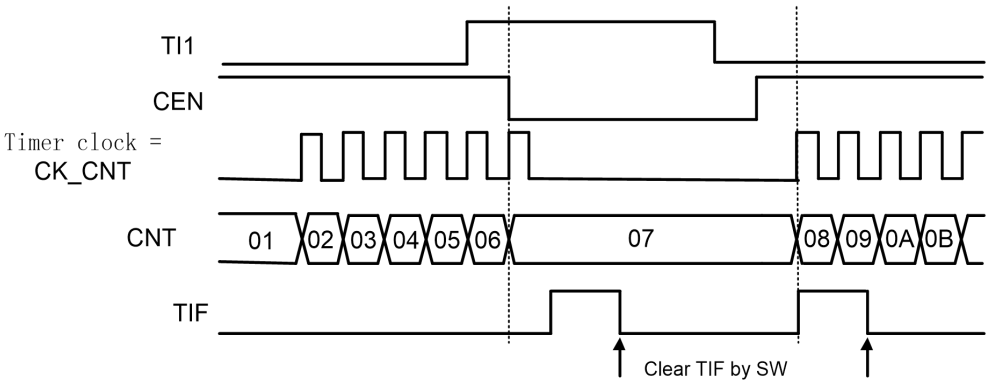


图30-34门控模式下的时序

触发模式

计数器在外部输入的某个事件到来后才开始计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置GPTIM_CCER.CC1P=0
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为触发模式，配置GPTIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置GPTIM_CCER.CC1E=1

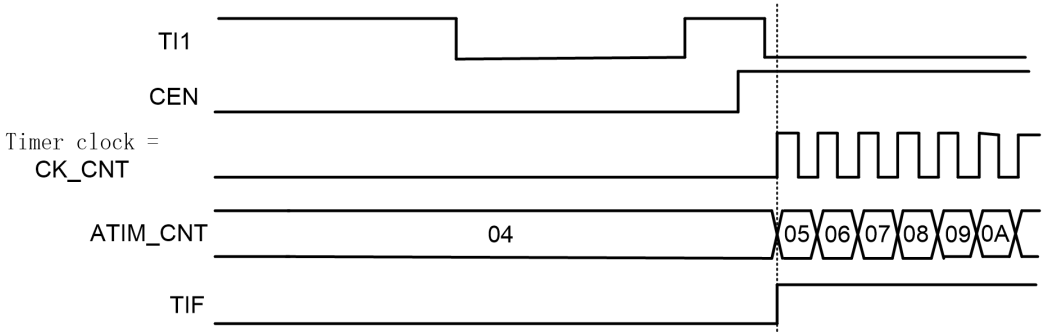


图30-35触发器模式下的时序

外部事件触发的外部时钟计数模式

可以将ETR设置为计数时钟，同时使用另一个外部输入作为计数器启动触发信号。比如在检测到TI1的上升沿之后，计数器开始以ETR输入的上升沿计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1，ATIM_ETR功能
- 设置ETP进行沿选择，GPTIM_SMCR.ETP=0
- 设置ETR分频比，配置GPTIM_SMCR.ETPS[1:0]=01
- 配置输入滤波时间，GPTIM_SMCR.ETF[3:0]=0000
- 置位ECE寄存器，使能外部时钟模式2，GPTIM_SMCR.ECE=1
- 关闭通道使能，配置GPTIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置GPTIM_CCER.CC1P=0
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为触发模式，配置GPTIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置GPTIM_CCER.CC1E=1

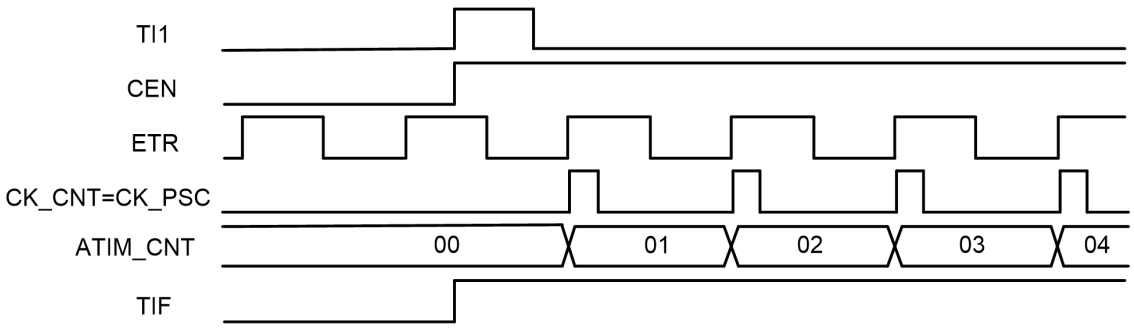


图30-36外部时钟模式2＋触发模式下的时序

30.4.14 DMA 访问

GPTIM支持6种DMA请求，分别为4个CC通道请求、外部触发请求和用户软件触发请求。

其中每个CC通道各自产生一个DMA请求，在捕捉模式下用于将CCRx中的内容传输给RAM，在比较模式下则用于将RAM中的数据写入CCRx；CC通道的DMA请求可以配置为单次传输或Burst传输（CCxBURSTEN），单次传输仅访问CCRx寄存器，Burst传输则根据DCR寄存器配置对特定的一组寄存器进行访问。

此外，外部触发事件和软件触发事件也可以产生DMA请求，当这两种请求发生时，会启动DMA Burst传输，向GPTIM内部1个或多个寄存器写入数据，或者从GPTIM读取1个或多个寄存器值。

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
GTIMx_CH1	0	0	Read CCR1	1
		1	Write CCR1	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH2	0	0	Read CCR2	1
		1	Write CCR2	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH3	0	0	Read CCR3	1
		1	Write CCR3	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH4	0	0	Read CCR4	1
		1	Write CCR4	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_TRIG	N/A	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_UEV	N/A	0	Read DMAR	DBL
		1	Write DMAR	

30.4.15 DMA Burst

DMA-Burst支持一个事件触发连续多次DMA请求，主要作用是在事件发生后连续更新多个寄存器的内容，因此可以实现动态实时调整输出波形等功能。

DMA控制器需将外设目标地址指向一个虚拟寄存器GPTIM_DMAR。在特定的定时器事件发生时，GPTIM会连续发射多个DMA请求。每个DMA对GPTIM_DMAR的写操作都会被GPTIM重新定向到实际的功能寄存器上。

DBL寄存器用于设置DMA burst长度，DBA寄存器用于设置DMA访问GPTIM内部的基地址（相对于GPTIM_CR的offset）。

30.4.16 输入异或功能

通道1~3的输入信号可以被异或起来之后，接入到通道1的滤波和边沿电路输入，用于通道1的输入捕捉或者触发。

GPTIM_CR2寄存器的TI1S位用于选择通道1的输入是否来自于三个通道输入的异或。

30.4.17 Debug 模式

当Cortex-M0进入debug模式后，定时器可以停止或继续工作，其行为由DCU模块的DBG_TIMx_STOP寄存器定义。

30.5 寄存器

GPTIM0模块基地址：0x40014C00

GPTIM1模块基地址：0x40016400

GPTIM2模块基地址：0x40018000

地址	名称	符号
GPTIM		
0x00	GPTIM 控制寄存器 1	GPTIMx_CR1
0x04	GPTIM 控制寄存器 2	GPTIMx_CR2
0x08	GPTIM 从机模式控制寄存器	GPTIMx_SMCR
0x0C	GPTIMDMA 和中断使能寄存器	GPTIMx_DIER
0x10	GPTIM 状态寄存器	GPTIMx_SR
0x14	GPTIM 事件产生寄存器	GPTIMx_EGR
0x18	GPTIM 捕捉/比较模式寄存器 1	GPTIMx_CCMR1
0x1C	GPTIM 捕捉/比较模式寄存器 2	GPTIMx_CCMR2
0x20	GPTIM 捕捉/比较使能寄存器	GPTIMx_CCER
0x24	GPTIM 计数器寄存器	GPTIMx_CNT
0x28	GPTIM 预分频寄存器	GPTIMx_PSC
0x2C	GPTIM 自动重载寄存器	GPTIMx_ARR
0x34	GPTIM 捕捉/比较寄存器 1	GPTIMx_CCR1
0x38	GPTIM 捕捉/比较寄存器 2	GPTIMx_CCR2
0x3C	GPTIM 捕捉/比较寄存器 3	GPTIMx_CCR3
0x40	GPTIM 捕捉/比较寄存器 4	GPTIMx_CCR4
0x48	GPTIMDMA 控制寄存器	GPTIMx_DCR
0x4C	GPTIMDMA 访问寄存器	GPTIMx_DMAR
0x60	GPTIMITR 选择寄存器	GPTIMx_ITRSEL

30.5.1 GPTIMx 控制寄存器 1

名称	GPTIMx_CR1, x=0,1,2							
地址	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CKD	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	CMS		DIR	OPM	URS	UDIS	CEN
位权限	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述

Bit	助记符	功能描述
31:10	-	RFU，未实现，读为 0
9:8	CKD	Dead time 和数字滤波时钟频率分频寄存器（相对 CK_INT 的分频比） 00：tDTS=tCK_INT 01：tDTS=2*tCK_INT 10：tDTS=4*tCK_INT 11：RFU，禁止使用
7	ARPE	Auto-reload 预装载使能 0：ARR 寄存器不使能 preload 1：ARR 寄存器使能 preload
6:5	CMS	计数器对齐模式选择 00：边沿对齐模式 01：中央对齐模式 1，输出比较中断标志仅在计数器向下计数的过程中置位 10：中央对齐模式 2，输出比较中断标志仅在计数器向上计数的过程中置位 11：中央对齐模式 3，输出比较中断标志在计数器向上向下计数的过程中都会置位
4	DIR	计数方向寄存器 0：向上计数 1：向下计数 注意：当定时器配置为中央计数模式或编码器模式时，此寄存器只读
3	OPM	单脉冲输出模式 0：Update Event 发生时计数器不停止 1：Update Event 发生时计数器停止（自动清零 CEN）
2	URS	更新请求选择 0：以下事件能够产生 update 中断 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1：仅计数器上溢出或下溢出会产生 update 中断
1	UDIS	禁止 update 0：使能 update 事件；以下事件发生时产生 update 事件 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1：禁止 update 事件，不更新 shadow 寄存器。当 UG 置位或从机控制器收到硬件 reset 时重新初始化计数器和预分频器。
0	CEN	计数器使能 0：计数器关闭 1：计数器使能 注意：外部触发模式可以自动置位 CEN

30.5.2 GPTIMx 控制寄存器 2

名称	GPTIMx_CR2							
地址	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TI1S	MMS			CCDS	-		
位权限	R/W-0	R/W-000			R/W-0	U-0		

Bit	助记符	功能描述
31:8	-	RFU，未实现，读为 0
7	TI1S	通道 1 输入源选择 0：GPTIMx_CH1 输入通道 1 1：GPTIMx_CH1, CH2, CH3 异或后输入通道 1
6:4	MMS	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源 000：GPTIM_EGR 的 UG 寄存器被用作 TRGO 001：计数器使能信号 CNT_EN 被用作 TRGO，可用于同时启动多个定时器 010：UE（update event）信号被用作 TRGO 011：比较脉冲，如果 CC1IF 标志将要置位，TRGO 输出一个正脉冲 100：OC1REF 用作 TRGO 101：OC2REF 用作 TRGO 110：OC3REF 用作 TRGO 111：OC4REF 用作 TRGO 注意：从机定时器或 ADC 必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
3	CCDS	捕捉/比较 DMA 选择 0：捕捉/比较事件发生时发送 DMA 请求 1：Update Event 发生时发送 DMA 请求
2:0	-	RFU，未实现，读为 0

30.5.3 GPTIMx 从机模式控制寄存器

名称	GPTIMx_SMCR							
地址	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ETP	ECE	ETPS		ETF			

位权限	RW-0	RW-0	RW-00		RW-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSM	TS			-	SMS		
位权限	RW-0	RW-000			U-0	RW-000		

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	**ETP**	外部触发信号极性配置 0：高电平或上升沿有效 1：低电平或下降沿有效
14	**ECE**	外部时钟使能 0：关闭外部时钟模式 2 1：使能外部时钟模式 2，计数器时钟为 ETRF 有效沿
13:12	**ETPS**	外部触发信号预分频寄存器 外部触发信号 ETRP 的频率最多只能是 GPTIM 工作时钟的 1/4，当输入信号频率较高时，可以使用预分频。 00：不分频 01：2 分频 10：4 分频 11：8 分频
11:8	**ETF**	外部触发信号滤波时钟和长度选择 0000：无滤波 0001：$f_{SAMPLING}=f_{CK_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
7	**MSM**	主/从模式 0：无动作 1：TRGI 触发的动作被延迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于单个外部事件对多个定时器进行同步的情况。
6:4	**TS**	触发选择，用于选择同步计数器的触发源 000：内部触发信号（ITR0） 001：内部触发信号（ITR1） 010：内部触发信号（ITR2） 011：内部触发信号（ITR3） 100：TI1 边沿检测（TI1F_ED） 101：滤波后 TI1（TI1FP1） 110：滤波后 TI2（TI2FP2）

Bit	助记符	功能描述
		111：外部触发输入（ETRF） 注意：仅当 SMS=000 即禁止从机模式的情况下，可以改写 TS 寄存器
3	--	RFU：未实现，读为 0
2:0	**SMS**	从机模式选择 000：从机模式禁止；CEN 使能后预分频电路时钟源来自内部时钟 001：Encoder 模式 1；计数器使用 TI2FP1 边沿，根据 TI1FP2 电平高低来计数 010：Encoder 模式 2；计数器使用 TI1FP2 边沿，根据 TI2FP1 电平高低来计数 011：Encoder 模式 3；计数器同时使用 TI1FP1 和 TI2FP2 边沿，根据其他输入信号电平来计数 100：复位模式；TRGI 上升沿初始化计数器，并触发寄存器 update 101：闸门模式；TRGI 为高电平时，计数时钟使能，TRGI 为低电平时，计数时钟停止 110：触发模式；TRGI 上升沿触发计数器开始计数（不会复位计数器） 111：外部时钟模式 1；TRGI 上升沿直接驱动计数器

30.5.4 GPTIMx DMA 和中断使能寄存器

名称	GPTIMx_DIER							
地址	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				CC4BURSTEN	CC3BURSTEN	CC2BURSTEN	CC1BURSTEN
位权限	U-0				RW-0	RW-0	RW-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	TDE	QDCIE	CC4DE	CC3DE	CC2DE	CC1DE	UDE
位权限	U-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	TIE	-	CC4IE	CC3IE	CC2IE	CC1IE	UIE
位权限	U-0	RW-0	U-0	RW-0	RW-0	RW-0	RW-0	RW-0

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19	**CC4BURSTEN**	捕捉比较通道 4 的 DMA 模式配置 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
18	**CC3BURSTEN**	捕捉比较通道 3 的 DMA 模式配置 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
17	**CC2BURSTEN**	捕捉比较通道 2 的 DMA 模式配置

Bit	助记符	功能描述
		0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
16	**CC1BURSTEN**	捕捉比较通道 1 的 DMA 模式配置 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
15	**--**	RFU：未实现，读为 0
14	**TDE**	外部触发 DMA 请求使能 0：从机模式下，禁止外部触发事件产生 DMA 请求 1：从机模式下，允许外部触发事件产生 DMA 请求（可用于自动更新 preload 寄存器）
13	**QDCIE**	Quadrature Decoder 方向改变中断使能 (Quad-decoder Interrupt Enable) 1：使能 QDCIF 中断 0：禁止 QDCIF 中断
12	**CC4DE**	捕捉比较通道 4 的 DMA 请求使能 0：禁止 CC4 DMA 请求 1：允许 CC4 DMA 请求
11	**CC3DE**	捕捉比较通道 3 的 DMA 请求使能 0：禁止 CC3 DMA 请求 1：允许 CC3 DMA 请求
10	**CC2DE**	捕捉比较通道 2 的 DMA 请求使能 0：禁止 CC2 DMA 请求 1：允许 CC2 DMA 请求
9	**CC1DE**	捕捉比较通道 1 的 DMA 请求使能 0：禁止 CC1 DMA 请求 1：允许 CC1 DMA 请求
8	**UDE**	Update Event DMA 请求使能 0：Update Event 发生时，禁止产生 DMA 请求 1：Update Event 发生时，允许产生 DMA 请求
7	**--**	RFU：未实现，读为 0
6	**TIE**	触发事件中断使能 0：禁止触发事件中断 1：允许触发事件中断
5	**--**	RFU：未实现，读为 0
4	**CC4IE**	捕捉/比较通道 4 中断使能 0：禁止捕捉/比较 4 中断 1：允许捕捉/比较 4 中断
3	**CC3IE**	捕捉/比较通道 3 中断使能 0：禁止捕捉/比较 3 中断 1：允许捕捉/比较 3 中断
2	**CC2IE**	捕捉/比较通道 2 中断使能 0：禁止捕捉/比较 2 中断 1：允许捕捉/比较 2 中断
1	**CC1IE**	捕捉/比较通道 1 中断使能 0：禁止捕捉/比较 1 中断 1：允许捕捉/比较 1 中断
0	**UIE**	Update 事件中断使能 0：禁止 Update 事件中断 1：允许 Update 事件中断

30.5.5 GPTIMx 状态寄存器

名称	GPTIMx_SR							
地址	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		QDIR	CC4OF	CC3OF	CC2OF	CC1OF	-
位权限	U-0		R-0	RW-0	RW-0	RC/W-0	RC/W-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	TIF	QDCIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
位权限	U-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Bit	助记符	功能描述
31:14	--	RFU：未实现，读为 0
13	QDIR	Quadrature Decoder 方向方向寄存器 (Quad-decoder Direction) 0：CH1 领先于 CH2 1：CH1 滞后于 CH2
12	CC4OF	捕捉/比较通道 4 的 Overcapture 中断 参考 CC1OF
11	CC3OF	捕捉/比较通道 3 的 Overcapture 中断 参考 CC1OF
10	CC2OF	捕捉/比较通道 2 的 Overcapture 中断 参考 CC1OF
9	CC1OF	捕捉/比较通道 1 的 Overcapture 中断 此寄存器仅在对应通道设置为输入捕捉模式的情况下有效。硬件置位，软件写 1 清零。 0：无 overcapture 事件 1：在 CC1IF 标志为 1 的情况下发生新的捕捉
8:7	--	RFU：未实现，读为 0
6	TIF	触发事件中断标志，硬件置位，软件写 1 清零
5	QDCIF	Quadrature Decoder 方向改变中断，硬件置位，软件写 1 清零(Quad-decoder Direction Change Interrupt Flag) 0：没有改变方向 1：方向改变
4	CC4IF	捕捉/比较通道 4 中断标志 参考 CC1IF
3	CC3IF	捕捉/比较通道 3 中断标志 参考 CC3IF
2	CC2IF	捕捉/比较通道 2 中断标志 参考 CC2IF
1	CC1IF	捕捉/比较通道 1 中断标志 如果 CC1 通道配置为输出：CC1IF 在计数值等于比较值时置

Bit	助记符	功能描述
		位，软件写 1 清零。 如果 CC1 通道配置为输入：发生捕捉事件时置位，软件写 1 清零，或者软件读 ATIM_CCR1 自动清零。
0	**UIF**	Update 事件中断标志，硬件置位，软件写 1 清零。 当以下事件发生时，UIF 置位，并更新 shadow 寄存器 -重复计数器=0，并且 UDIS=0 的情况下，计数器发生溢出 -URS=0 且 UDIS=0 的情况下，软件置位 UG 寄存器初始化计数器 -URS=0 且 UDIS=0 的情况下，触发事件初始化计数器

30.5.6 GPTIMx 事件产生寄存器

名称	GPTIMx_EGR							
地址	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	TG	-	CC4G	CC3G	CC2G	CC1G	UG
位权限	U-0	W-0	U-0	W-0	W-0	W-0	W-0	W-0

Bit	助记符	功能描述
31:7	**--**	RFU：未实现，读为 0
6	**TG**	软件触发，软件置位此寄存器产生触发事件，硬件自动清零
5	**--**	RFU：未实现，读为 0
4	**CC4G**	捕捉/比较通道 4 软件触发，参考 CC1G
3	**CC3G**	捕捉/比较通道 3 软件触发，参考 CC1G
2	**CC2G**	捕捉/比较通道 2 软件触发，参考 CC1G
1	**CC1G**	捕捉/比较通道 1 软件触发 如果 CC1 通道配置为输出：CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求 如果 CC1 通道配置为输入：当前计数值被捕捉到 ATIM_CCR1 寄存器，CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求
0	**UG**	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零 软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。

30.5.7 GPTIMx 捕捉/比较模式寄存器 1

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能

名称	GPTIMx_CCMR1							
地址	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC2CE	OC2M			OC2PE	OC2FE	CC2S	
	IC2F				IC2PSC			
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC1CE	OC1M			OC1PE	OC1FE	CC1S	
	IC1F				IC1PSC			
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

输出比较模式

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	**OC2CE**	输出比较 2 清零使能，参考 OC1CE
14:12	**OC2M**	输出比较 2 模式配置，参考 OC1M
11	**OC2PE**	输出比较 2 预装载使能，参考 OC1PE
10	**OC2FE**	输出比较 2 快速使能，参考 OC1FE
9:8	**CC2S**	捕捉/比较 2 通道选择 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 10：CC2 通道配置为输入，IC2 映射到 TI1 11：CC2 通道配置为输入，IC2 映射到 TRC 注意：CC2S 仅在通道关闭时（CC2E=0）可以写
7	**OC1CE**	输出比较 1 清零使能 0：OC1REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	**OC1M**	输出比较 1 模式配置，此寄存器定义 OC1REF 信号的行为 000：输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出 001：CCR1=CNT 时，将 OC1REF 置高 010：CCR1=CNT 时，将 OC1REF 置低 011：CCR1=CNT 时，翻转 OC1REF 100：OC1REF 固定为低（inactive） 101：OC1REF 固定为高（active） 110：PWM 模式 1 –在向上计数时，OC1REF 在 CNT<CCR1 时置高，否则置低；在向下计数时，OC1REF 在 CNT>CCR1 时置低，否则置高 111：PWM 模式 2 –在向上计数时，OC1REF 在 CNT<CCR1 时置低，否则置高；在向下计数时，OC1REF 在 CNT>CCR1 时置高，否则置低
3	**OC1PE**	输出比较 1 预装载使能 0：CCR1 preload 寄存器无效，CCR1 可以直接写入

Bit	助记符	功能描述
		1：CCR1 preload 寄存器有效，针对 CCR1 的读写操作都是访问 preload 寄存器，当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	**OC1FE**	输出比较 1 快速使能 0：关闭快速使能，trigger 输入不会影响比较输出 1：打开快速使能，trigger 输入会立即将 OC1REF 改变为比较值匹配时的输出，而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效
1:0	**CC1S**	捕捉/比较 1 通道选择 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 10：CC1 通道配置为输入，IC1 映射到 TI2 11：CC1 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

输入捕捉模式

Bit	助记符	功能描述
31:16	**--**	RFU：未实现，读为 0
15:12	**IC2F**	输入捕捉 2 滤波
11:10	**IC2PSC**	输入捕捉 2 预分频
9:8	**CC2S**	捕捉/比较 2 通道选择 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC3 映射到 TI2 10：CC2 通道配置为输入，IC3 映射到 TI1 11：CC2 通道配置为输入，IC3 映射到 TRC 注意：CC2S 仅在通道关闭时（CC2E=0）可以写
7:4	**IC1F**	输入捕捉 1 滤波 此寄存器定义 TI1 的采样频率和滤波长度 0000：无滤波，使用 f_{DTS} 采样 0001：$f_{SAMPLING}=f_{CK_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC1PSC**	输入捕捉 1 预分频 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉

Bit	助记符	功能描述
		IC1PSC 寄存器在 CC1E=0 时复位
1:0	**CC1S**	捕捉/比较 1 通道选择 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 10：CC1 通道配置为输入，IC1 映射到 TI2 11：CC1 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

30.5.8 GPTIMx 捕捉/比较模式寄存器 2

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能

名称	GPTIMx_CCMR2							
地址	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC4CE	OC4M			OC4PE	OC4FE	CC4S	
	IC2F				IC2PSC			
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC3CE	OC3M			OC3PE	OC3FE	CC3S	
	IC3F				IC3PSC			
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

输出比较模式

Bit	助记符	功能描述
31:16	**--**	RFU：未实现，读为 0
15	**OC4CE**	输出比较 4 清零使能，参考 OC1CE
14:12	**OC4M**	输出比较 4 模式配置，参考 OC1M
11	**OC4PE**	输出比较 4 预装载使能，参考 OC1PE
10	**OC4FE**	输出比较 4 快速使能，参考 OC1FE
9:8	**CC4S**	捕捉/比较 4 通道选择 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 10：CC4 通道配置为输入，IC4 映射到 TI3 11：CC4 通道配置为输入，IC4 映射到 TRC 注意：CC4S 仅在通道关闭时（CC4E=0）可以写
7	**OC3CE**	输出比较 1 清零使能 0：OC1REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	**OC3M**	输出比较 3 模式配置，此寄存器定义 OC3REF 信号的行为 000：输出比较寄存器 CCR3 和计数器 CNT 的比较结果不会影响输出

Bit	助记符	功能描述
		001：CCR3=CNT 时，将 OC1REF 置高 010：CCR3=CNT 时，将 OC1REF 置低 011：CCR3=CNT 时，翻转 OC1REF 100：OC3REF 固定为低（inactive） 101：OC3REF 固定为高（active） 110：PWM 模式 1 –在向上计数时，OC3REF 在 CNT<CCR3 时置高，否则置低；在向下计数时，OC3REF 在 CNT>CCR3 时置低，否则置高 111：PWM 模式 2 –在向上计数时，OC3REF 在 CNT<CCR3 时置低，否则置高；在向下计数时，OC3REF 在 CNT>CCR3 时置高，否则置低
3	**OC3PE**	输出比较 3 预装载使能 0：CCR3 preload 寄存器无效，CCR3 可以直接写入 1：CCR3 preload 寄存器有效，针对 CCR3 的读写操作都是访问 preload 寄存器，当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	**OC3FE**	输出比较 3 快速使能 0：关闭快速使能，trigger 输入不会影响比较输出 1：打开快速使能，trigger 输入会立即将 OC3REF 改变为比较值匹配时的输出，而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效
1:0	**CC3S**	捕捉/比较 3 通道选择 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC1 映射到 TI3 10：CC3 通道配置为输入，IC1 映射到 TI4 11：CC3 通道配置为输入，IC1 映射到 TRC 注意：CC3S 仅在通道关闭时（CC3E=0）可以写

输入捕捉模式

Bit	助记符	功能描述
31:16	**--**	RFU：未实现，读为 0
15:12	**IC4F**	输入捕捉 4 滤波
11:10	**IC4PSC**	输入捕捉 4 预分频
9:8	**CC4S**	捕捉/比较 4 通道选择 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 10：CC4 通道配置为输入，IC4 映射到 TI3 11：CC4 通道配置为输入，IC4 映射到 TRC 注意：CC4S 仅在通道关闭时（CC4E=0）可以写
7:4	**IC3F**	输入捕捉 1 滤波 此寄存器定义 TI3 的采样频率和滤波长度 0000：无滤波，使用 f_{DTS} 采样 0001：$f_{SAMPLING}=f_{CK_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8

Bit	助记符	功能描述
		1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC3PSC**	输入捕捉 3 预分频 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC1PSC 寄存器在 CC1E=0 时复位
1:0	**CC3S**	捕捉/比较 3 通道选择 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC1 映射到 TI3 10：CC3 通道配置为输入，IC1 映射到 TI4 11：CC3 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

30.5.9 GPTIMx 捕捉/比较使能寄存器

名称	GPTIMx_CCER							
地址	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		CC4P	CC4E	-		CC3P	CC3E
位权限	U-0		R/W-0	R/W-0	U-0		R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		CC2P	CC2E	-		CC1P	CC1E
位权限	U-0		RW-0	RW-0	U-0		RW-0	RW-0

Bit	助记符	功能描述
31:14	**--**	RFU：未实现，读为 0
13	**CC4P**	捕捉/比较 4 输出极性，参考 CC1P
12	**CC4E**	捕捉/比较 4 输出使能，参考 CC1E
11:10	**--**	RFU：未实现，读为 0
9	**CC3P**	捕捉/比较 3 输出极性，参考 CC1P
8	**CC3E**	捕捉/比较 3 输出使能，参考 CC1E
7:6	**--**	RFU：未实现，读为 0
5	**CC2P**	捕捉/比较 2 输出极性，参考 CC1P
4	**CC2E**	捕捉/比较 2 输出使能，参考 CC1E

Bit	助记符	功能描述
3:2	--	RFU：未实现，读为 0
1	**CC1P**	捕捉/比较 1 输出极性 **CC1 通道配置为输出时：** 0：OC1 高有效 1：OC1 低有效 **CC1 通道配置为输入时：** CC1NP/CC1P 用于选择 TI1FP1 和 TI2FP1 的极性 00：非取反/上升沿 01：取反/下降沿 10：保留，不要使用 11：非取反，上下沿都有效
0	**CC1E**	捕捉/比较 1 输出使能 **CC1 通道配置为输出时** 0：OC1 输出关闭，Ocx=0，Ocx_EN=0 1：Ocx=OCxREF+极性选择，Ocx_EN=1 **CC1 通道配置为输入时** 0：关闭捕捉功能 1：使能捕捉功能

标准 Ocx 通道的输出控制位

CcxE 位	Ocx 输出状态
0	禁止输出（Ocx=0，Ocx_EN=0）
1	Ocx=OCxREF + 极性，Ocx_EN=1

30.5.10 GPTIMx 计数器寄存器

名称	GPTIMx_CNT							
地址	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CNT**	计数器值

30.5.11 GPTIMx 预分频寄存器

名称	GPTIMx_PSC							
地址	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PSC[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**PSC**	计数器时钟（CK_CNT）预分频值 $f_{CK_CNT}=f_{CK_PSC}/(PSC[15:0]+1)$ 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

30.5.12 GPTIMx 自动重载（auto-reload）寄存器

名称	GPTIMx_ARR							
地址	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	RW-11111111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	RW-11111111							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**ARR**	计数溢出时的自动重载值 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

30.5.13 GPTIMx 捕捉/比较寄存器 1

名称	GPTIMx_CCR1							
地址	0x34							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CCR1**	捕捉/比较通道 1 寄存器 **如果通道 1 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC1 输出 **如果通道 1 配置为输入：** CCR1 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR1 为只读

30.5.14 GPTIMx 捕捉/比较寄存器 2

名称	GPTIMx_CCR2							
地址	0x38							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CCR2**	捕捉/比较通道 2 寄存器 **如果通道 2 配置为输出：**

Bit	助记符	功能描述
		这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC2 输出 **如果通道 2 配置为输入：** CCR2 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR2 为只读

30.5.15 GPTIMx 捕捉/比较寄存器 3

名称	GPTIMx_CCR3							
地址	0x3C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR3[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR3[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CCR3**	捕捉/比较通道 3 寄存器 **如果通道 3 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC3 输出 **如果通道 3 配置为输入：** CCR3 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR3 为只读

30.5.16 GPTIMx 捕捉/比较寄存器 4

名称	GPTIMx_CCR4							
地址	0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR4[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	CCR4[7:0]
位权限	RW-00000000

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CCR4**	捕捉/比较通道 4 寄存器 **如果通道 4 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC4 输出 **如果通道 4 配置为输入：** CCR4 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR4 为只读

30.5.17 GPTIMx DMA 控制寄存器

名称	GPTIMx_DCR							
地址	0x48							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			DBL				
位权限	U-0			RW-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			DBA				
位权限	U-0			RW-00000				

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12:8	**DBL**	DMA Burst 长度 对 GPTIM_DMAR 寄存器的读写将触发 burst DMA 操作，burst 长度为 1~18 00000：长度=1 00001：长度=2 …… 10001：长度=18 其他：无效值，禁止写入
7:5	--	RFU：未实现，读为 0
4:0	**DBA**	DMA 基地址，定义指向寄存器的偏移地址 00000：GPTIM_CR1 00001：GPTIM_CR2 00010：GPTIM_SMCR …… 注意：当 DBA+DBL 超出了 GPTIM 寄存器地址范围，则实际

Bit	助记符	功能描述
		burst 传输到 GPTIM 最高寄存器地址后自动停止，即 burst 长度会缩短。

30.5.18 GPTIMx DMA 访问寄存器

名称	GPTIMx_DMAR							
地址	0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DMAR[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DMAR[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**DMAR**	DMA burst 访问寄存器 在使用 DMA burst 传输时，将 DMA 通道外设地址设置为 GPTIM_DMAR，GPTIM 会根据 DBL 的值产生多次 DMA 请求

30.5.19 GPTIMx ITR 选择寄存器

名称	GPTIMx_ITRSEL							
地址	0x60							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ITR3SEL		ITR2SEL		ITR1SEL		ITR0SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	

Bit	助记符	功能描述
31:10	-	RFU，未实现，读为 0

Bit	助记符	功能描述
7:6	ITR3SEL	ITR 输入信号选择，详情参见 0 内部触发信号（ITRx）的捕捉
5:4	ITR2SEL	
3:2	ITR1SEL	
1:0	ITR0SEL	

31 定时器阵列单元（TAU）

31.1 概述

TAU是16bit定时器组成的阵列，支持以下功能

- 2组共16个独立的16bit向上计数器（TAU00~TAU07，TAU10~TAU17）
- Auto-reload定时
- 外部事件计数
- 输入边沿捕捉（事件触发、自由计数）
- 输入预分频
- 脉冲宽度或周期捕捉（PWC模式）
- 脉冲宽度调制（PWM模式）
- 可级联成32bit定时器

8个独立定时器结构相同，仅是输入输出通道不同；每个独立定时器包含计数器、自动重载寄存器、捕捉/比较寄存器。定时器启动后从0开始计数，计数至自动重载寄存器的值溢出信号，同时计数器回零重新开始计数。定时器允许在计数过程中改变计数初值，完成动态的定时时间更新；

支持2种边沿捕捉模式：

✓ 事件触发捕捉：使能后计数器保持0，捕捉到第一个有效沿之后定时器才开始计数；

✓ 自由计数捕捉：定时器使能后立即由0开始计数，在有效捕捉信号沿到来时锁存当前计数值，同时产生捕捉中断，在下一有效捕捉沿到来时再次锁存当前计数值并产生捕捉中断，在计数溢出后产生溢出中断。

脉冲宽度测量模式（PWC）在第一个有效沿到来后开始计数，第二个有效沿到来后停止计数，两个有效沿可以独立配置上升或下降，即可以实现正脉冲宽度捕捉、负脉冲宽度捕捉、正沿周期捕捉、负沿周期捕捉。

每个定时器支持外部引脚输入和外部引脚输出。其输入信号也可以重定向到内部信号，用于内部信号的测量和捕捉功能。

31.2 结构框图

TAU整体结构示意图如下，包含TAU0和TAU1两个阵列。

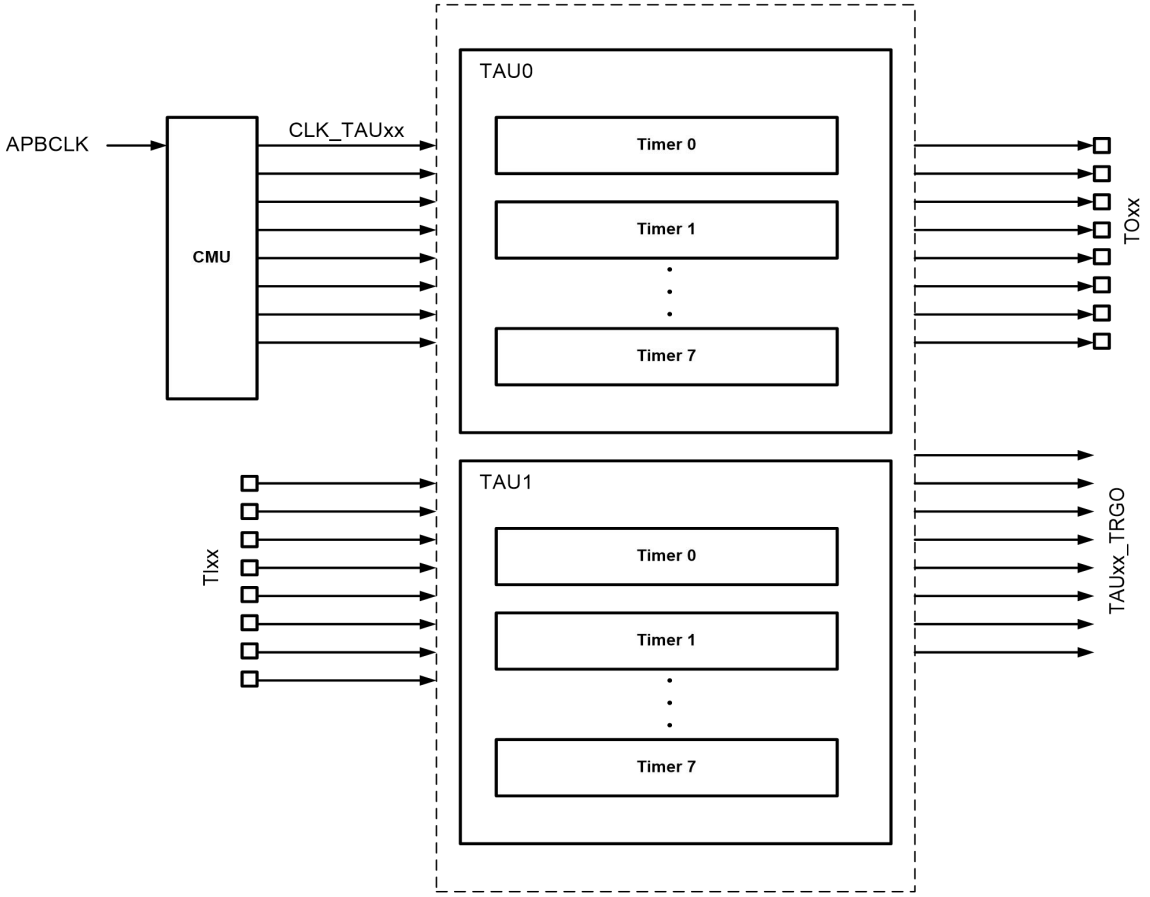


图 31-1TAU 结构框图

独立定时器结构框图如下：

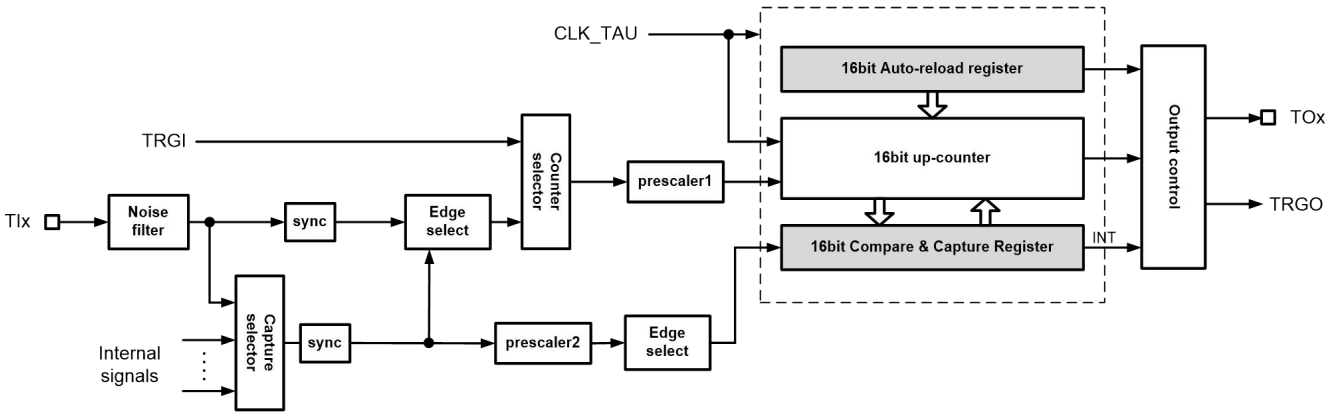


图 31-2TimerX 结构框图

定时器TAU0支持级联模式，此时一个16位定时器作为主机，一个或者多个16位定时器作为从机，从而实现32位级联定时器的功能。级联模式下，以master输出的溢出中断作为slave的计数触发信号，同时将输入到master的捕捉信号连接到slave的捕捉信号输入。

31.3 功能描述

31.3.1 时钟和复位

TAU使用APBCLK工作，在使能TAU开始工作前，必须使能其总线时钟，即置位TAUxx_PCE寄存器，参见CMU章节。比如想要使能TAU0中的Timer0工作，应置位TAU00_PEC寄存器。

通过TAUxx_RST寄存器可以实现每个定时器的独立复位，参见RMU章节。如果想要使能TAU0中的Timer3工作，应清零TAU03_RST寄存器。

31.3.2 工作模式

定时器支持以下工作模式：

- 计数模式
  - 单次计数和连续计数
  - 内部时钟计数
  - 外部信号计数
  - 内部信号计数
  - 定时器级联计数
- 捕捉模式
  - 周期捕捉
  - 电平宽度捕捉
  - 单次捕捉和连续捕捉
- 比较模式
  - PWM输出
  - 单次脉冲输出
  - 外部信号分频

31.3.3 输入输出通道

每个定时器包含多个输入通道：

- 外部引脚输入ETIx：用于计数或者捕捉
- 内部信号输入：用于计数或者捕捉

- 其他定时器输入：用于级联计数或级联捕捉

每个定时器包含2个输出通道：

- 外部引脚输出ETOx：用于波形输出
- 触发信号输出：用于级联或其他外设定时触发

31.3.4 16 位计数模式

定时器计数模式启动和计数溢出时将初值寄存器的值加载到计数器中。计数值达到0xFFFF时产生计数溢出中断。

定时器计数模式支持单次计数和连续计数功能。单次计数（one-shot）在定时器被触发后

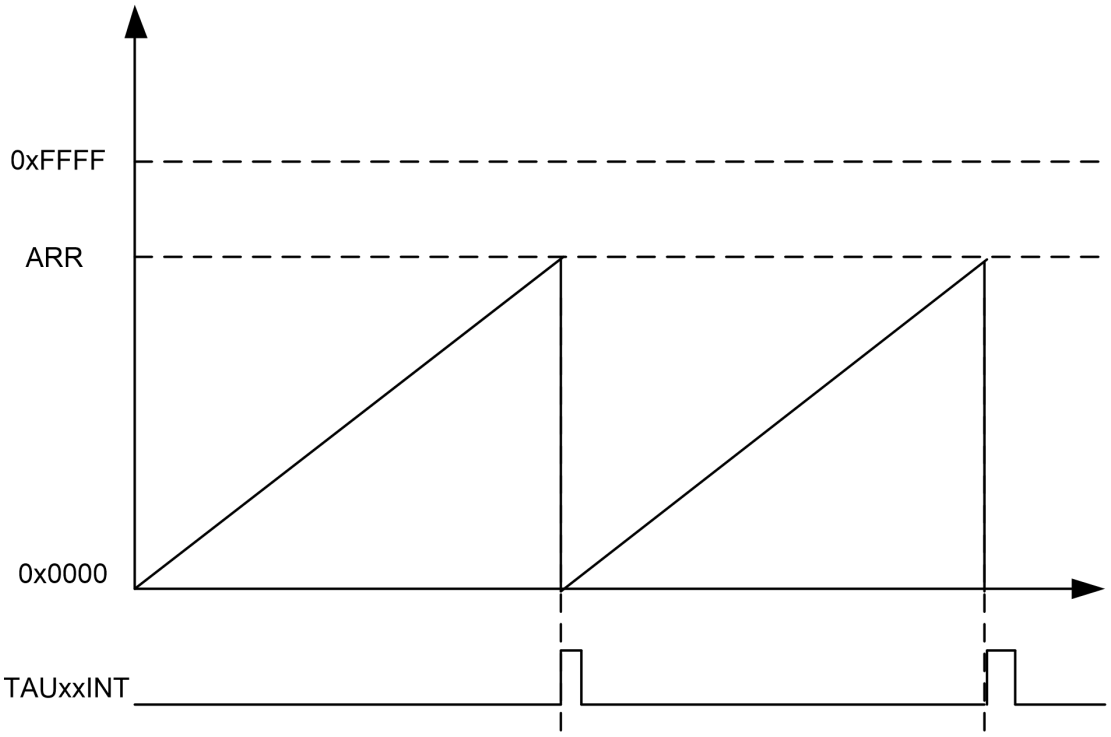


图 31-3 16bit 连续定时功能

定时器计数源可以基于工作时钟（及其分频）、外部输入信号、内部输入信号、Master定时器触发信号。

内部时钟计数

计数源通过CNTSEL寄存器选择为APBCLK时，定时器在每个APBCLK上升沿递增。此时如果配置了Prescale1（预分频寄存器1），则定时器以分频后的频率递增，如下图所示。

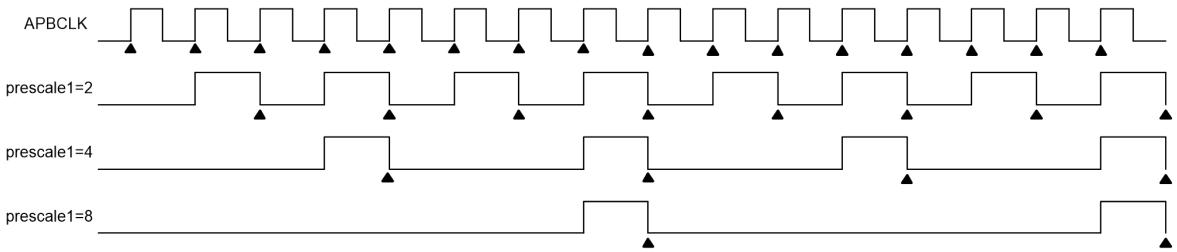


图 31-4 定时器内部时钟预分频计数

上图显示了基本定时功能，分别使用APBCLK不分频、2分频、4分频、8分频作为计数源，黑色三角表示实际计数器递增的位置，由于定时器是同步工作的，计数器递增都发生在APBCLK的上升沿。

外部输入信号计数

定时器可以基于外部输入信号计数，可以选择使用外部信号的上升沿、下降沿、或者上升下降沿计数，并且同样可以实现分频计数。

下图为ETIx上升沿事件计数的例子，黑色三角表示计数器递增位置。为了实现同步计数，输入信号在进行边沿检测前首先被APBCLK采样同步。

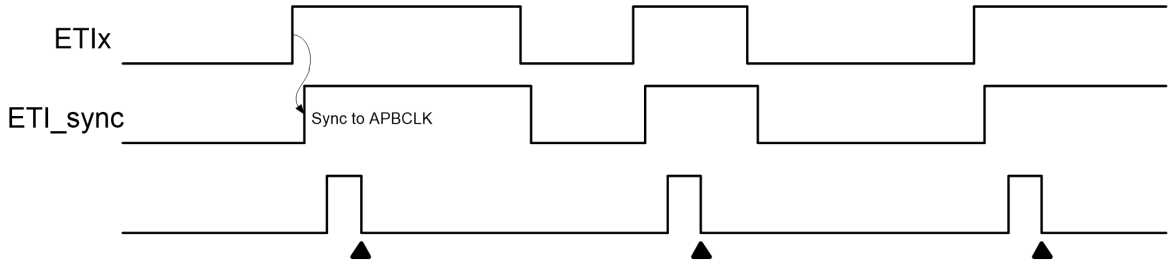


图 31-5 定时器外部事件上升沿递增时序

下图为ETIx上升下降沿事件计数，并且分频系数为2的例子。

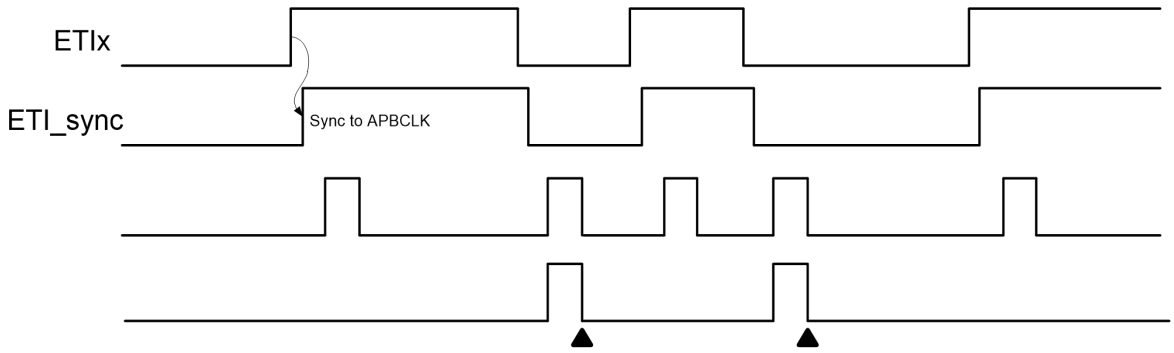


图 31-6 定时器外部事件上升下降沿递增，2 分频时序

内部输入信号计数

通过配置CNTSEL寄存器也可以实现对内部信号特定边沿的计数，比如内部时钟、比较器输出、UART

接收数据等。

下图为定时器在比较器输出下降沿计数，可实现累计模拟输入信号向下翻越比较器比较基准的次数。

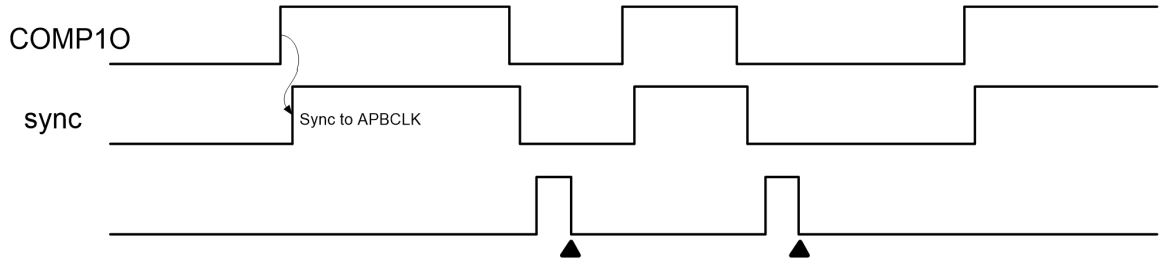


图 31-7 定时器内部信号下降沿递增

31.3.5 触发信号输出（TRGO）

TAU的每个16bit独立定时器都可以输出触发信号，当定时器计数值匹配CCR时，产生一个TRGO信号，宽度为1个APBCLK周期。此信号可以连接到TAU的其他定时器用于形成32bit级联定时器，或者输出到其他外设模块，用于定时触发其他外设工作。

TAU的16个独立定时器共可以输出16个TRGO：TAUxx_TRGO

TAU输出的TO信号除了可以连接搭配GPIO之外，也可以连接到其他外设作为异步触发信号使用。

31.3.6 32 位级联模式

级联计数模式下，作为master的定时器在计数器匹配CCR时发送TRGO触发信号，其他定时器如果使能slave模式，会在接收到TRGO信号后进行计数器递增。

从机模式有两种工作方式：触发递增（trigger to increment）和触发启动（trigger to start），通过SLVMD寄存器配置。

- 触发递增：每个触发信号导致从机计数器+1
- 触发启动：触发信号导致从机计数器从0开始计数，直到计数值等于ARR停止

下图是触发递增模式示例，以TAU0x为master，TAU0y为slave，当slave计数值到达ARR后产生从机OVIF，并回到0x0000重新计数。

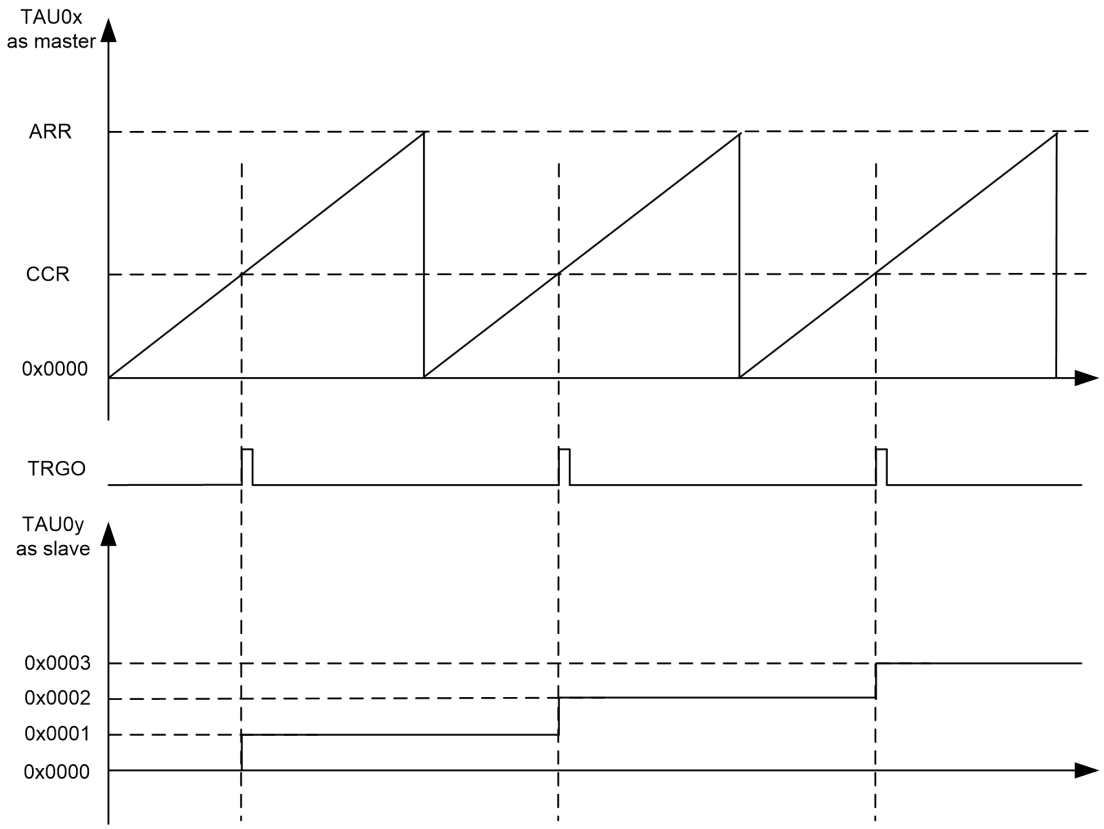


图 31-8 定时器级联计数（trigger to increment）

如果从机配置了计数源预分频，则主机多次触发才会导致从机递增一次。

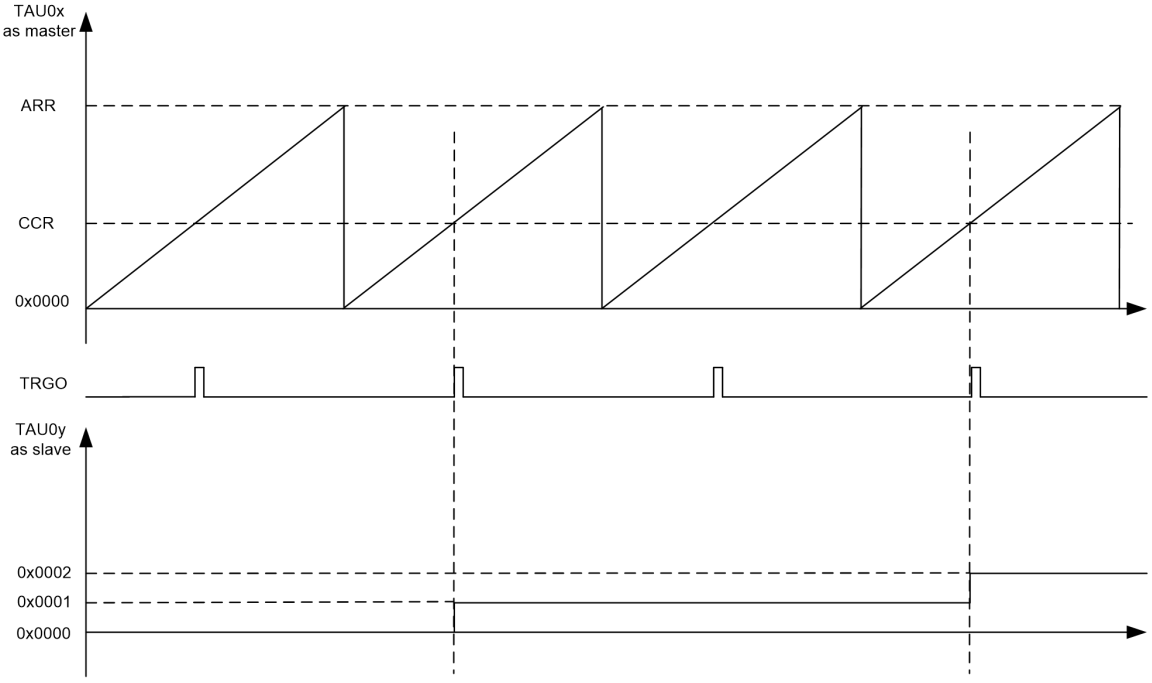


图 31-9 定时器级联计数（trigger to increment），从机 Prescale1 = 1

下图是触发启动模式示例，以TAU0x为master，TAU0y为slave，当slave计数值到达ARR后产生从机OVIF，并回到0x0000等待下一个触发信号。

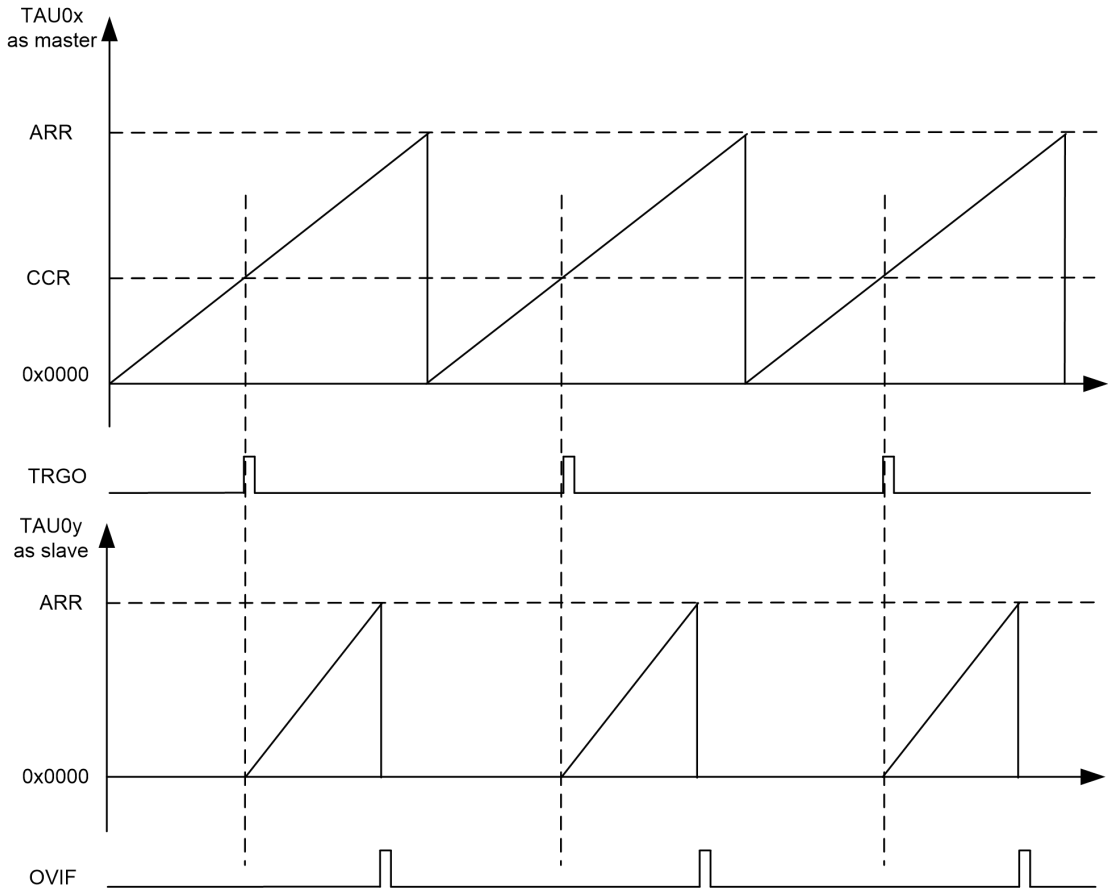


图 31-10 定时器级联计数（trigger to start）

注意，在触发启动模式下，软件应该合理配置主机从机的ARR参数，否则可能导致从机永远无法达到ARR，则从机OVIF永远不会置起，如下图所示。

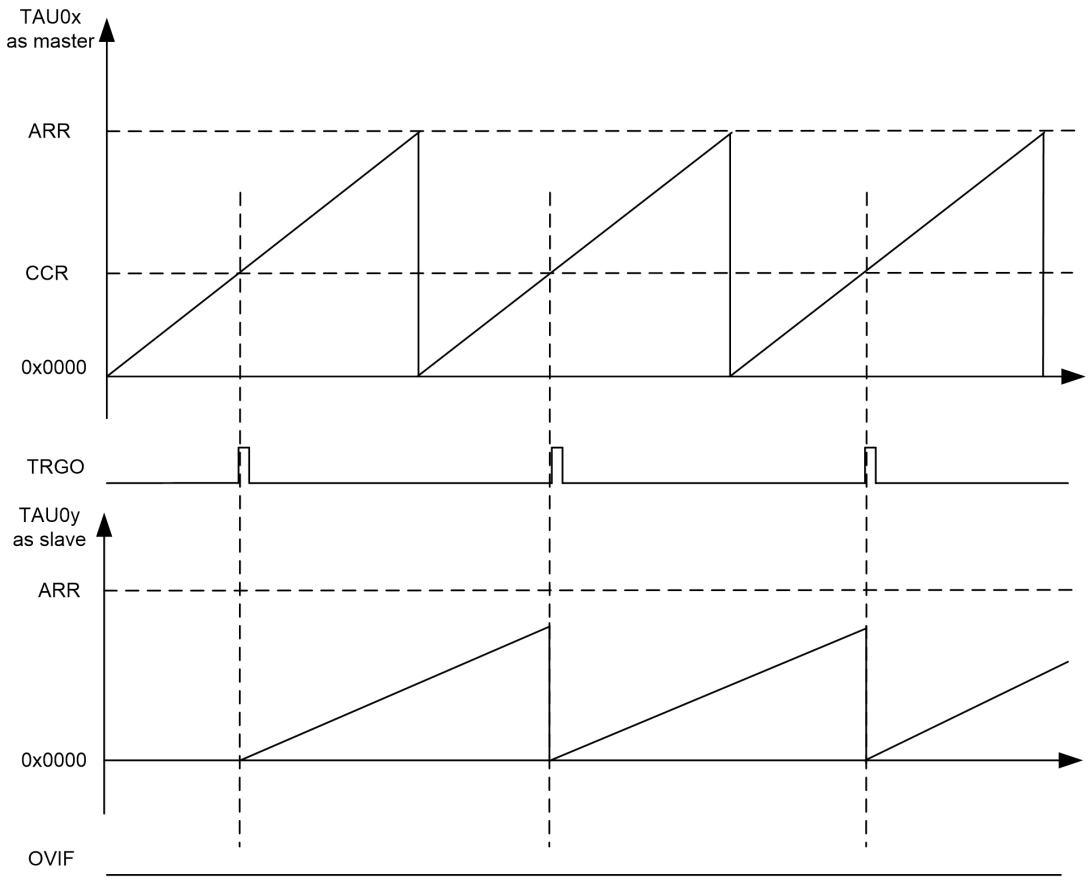


图 31-11 定时器级联计数（trigger to start），从机无法产生溢出

触发启动（trigger to start）模式下，主机和从机也可以配置不同的计数源预分频值，此时从机被触发后，计数器递增频率与主机不同。

下图为主机计数频率较高，而从机计数频率较低的例子。

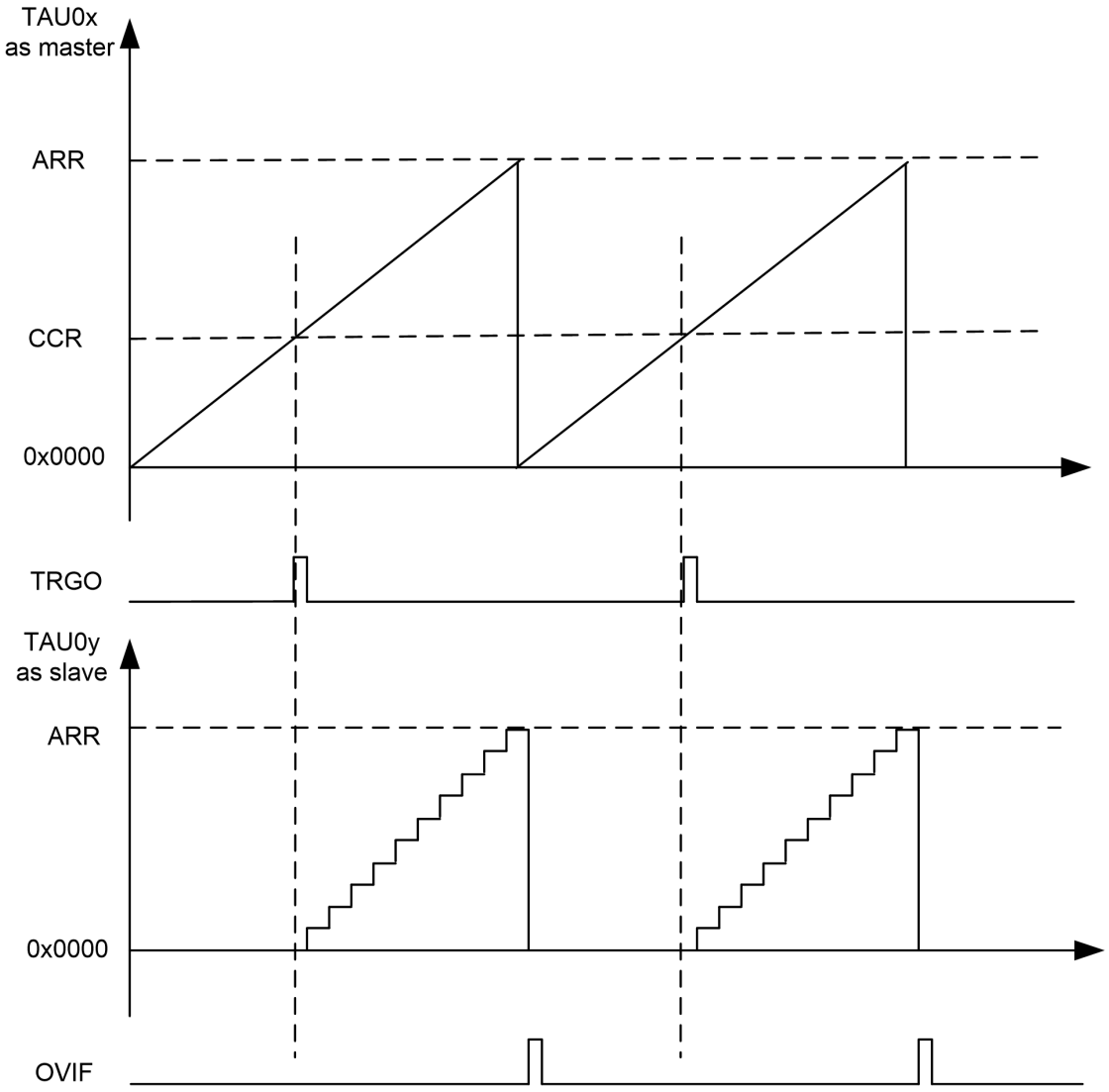


图 31-12 定时器级联计数（trigger to start），从机计数预分频

注意：只有TAU0支持级联计数模式，TAU1不支持级联计数。

TS寄存器（Trigger Select）用于选择从机模式下的主机触发源，每个定时器有2bit TS，可以选择4个主机触发输入。定时器0不支持从机模式，定时器1~7的主机触发输入选择参照以下表格。

TAU0 定时器#	TS			
	00	01	10	11
0	-	-	-	-
1	Timer0	-	-	-
2	Timer0	Timer1	-	-
3	Timer0	Timer1	Timer2	-
4	Timer0	Timer1	Timer2	Timer3
5	Timer0	Timer2	Timer3	Timer4

TAU0 定时器#	TS			
	00	01	10	11
6	Timer0	Timer3	Timer4	Timer5
7	Timer0	Timer4	Timer5	Timer6

表 31-1 定时器级联触发选择

注意：如果需要在级联模式下进行捕捉，则必须将输入到master的被捕捉信号同时连接到slave的捕捉信号输入。

31.3.7 捕捉功能

定时器支持两种捕捉模式：

- ✓ 清零捕捉：使能后计数器保持0，捕捉到第一个有效沿之后定时器才开始计数；并且每次捕捉事件将自动清零计数器。
- ✓ 自由计数捕捉：定时器使能后立即由0开始计数，在有效捕捉信号沿到来时锁存当前计数值，同时产生捕捉中断，在下一有效捕捉沿到来时再次锁存当前计数值并产生捕捉中断，在计数溢出后产生溢出中断。

捕捉事件发生时，当前计数值被保存在TAUx_CCRy寄存器中供软件读取。

定时器使用APBCLK采样其输入信号进行计数，所以应用中应避免被采样的输入信号频率高于APBCLK时钟频率。

软件可以配置信号捕捉采用上升沿、下降沿、还是同时捕捉上升下降沿。对被捕捉的信号边沿还可以进行预分频。

周期捕捉模式（MD=10）用于捕捉输入信号的周期，即相同边沿之间的间隔。

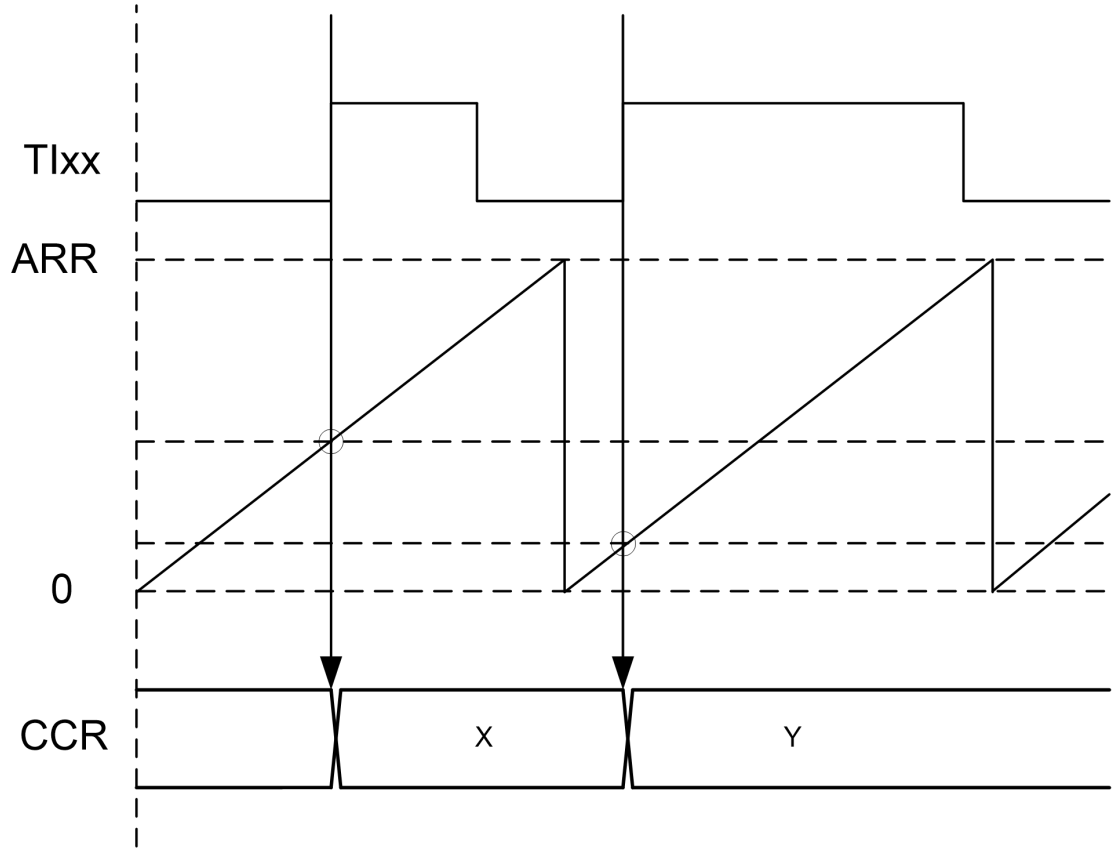


图 31-13 定时器捕捉外部输入信号上升沿

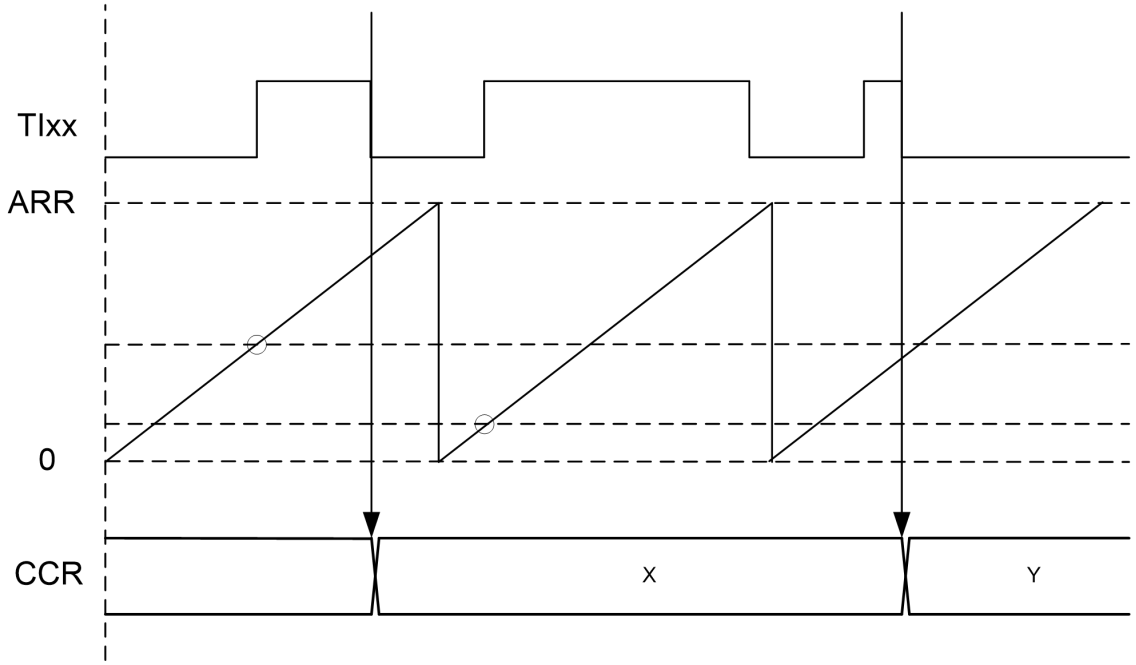


图 31-14 定时器 2 分频捕捉外部输入信号下降沿

脉冲宽度捕捉（PWC）功能可以用于测量输入信号的高电平宽度或者低电平宽度，需要配置MD寄存器为11。PWC模式下，如果捕捉沿配置为上升沿，则定时器在捕捉到第一个上升沿时开始计数，紧接着在捕捉到第一个下降沿时停止计数，此时定时器计数值就表示输入信号高电平的宽度；如果捕捉沿配置为下降沿，则定时器在捕捉到第一个下降沿时开始计数，紧接着在捕捉到第一个上升沿时停止计数，此时定时器计数值就表示输入信号低电平的宽度；如果捕捉沿配置为上升下降沿，则定时器在捕捉到第一个任意边沿时开始计数，紧接着在捕捉到下一个边沿时停止计数，此时定时器计数值就表示输入信号电平的宽度。

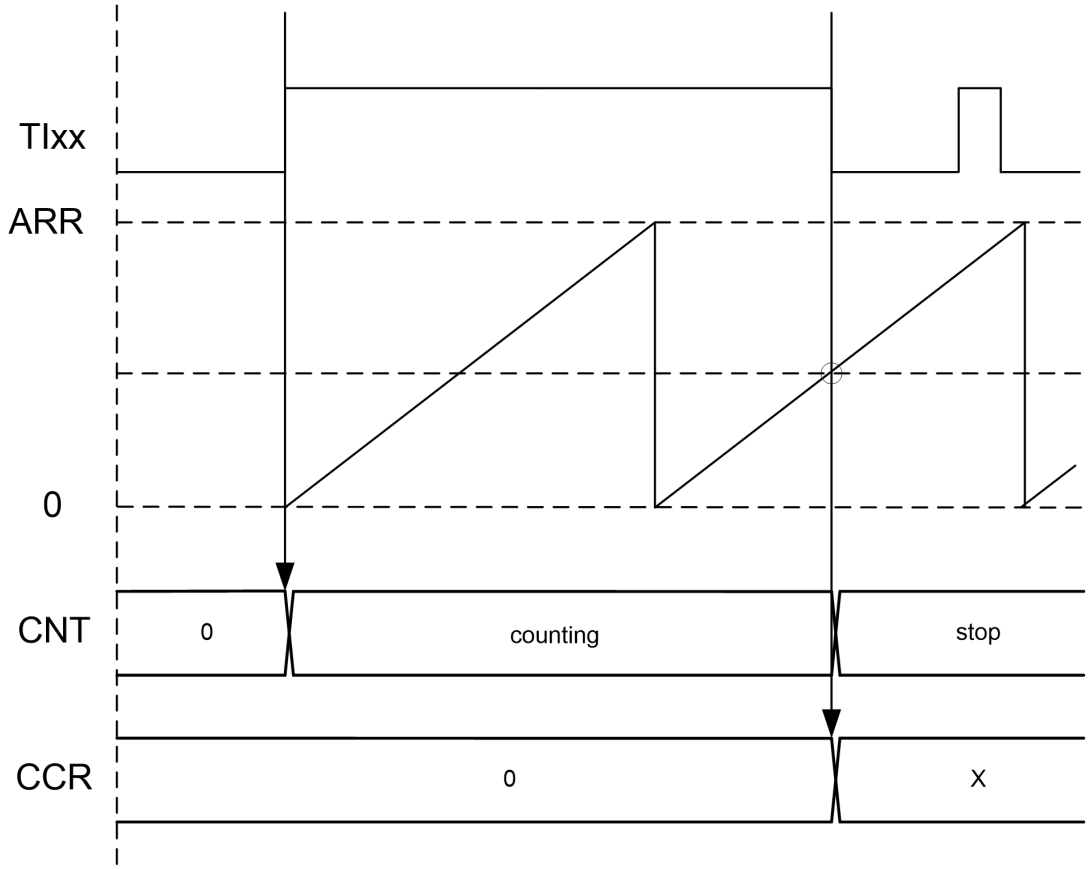


图 31-15 定时器脉冲宽度高电平捕捉

捕捉功能支持对内部信号的捕捉，通过TAU配置寄存器中的CAPSEL寄存器，可以选择每个定时器的捕捉源。

TAU0定时器	CAPSEL	被捕捉信号
Timer0	000	TI00
	001	UART0_RXD
	010	UART1_RXD
	011	XTLF
	100	UART2_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP

Timer1	000	TI01
	001	UART3_RXD
	010	UART4_RXD
	011	XTLF
	100	UART5_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer2	000	TI02
	001	UART0_RXD
	010	UART1_RXD
	011	XTLF
	100	UART2_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer3	000	TI03
	001	UART3_RXD
	010	UART4_RXD
	011	XTLF
	100	UART5_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer4	000	TI04
	001	LPUART1_RXD
	010	XTLF
	011	RCHF
	100	RCLP
	101	-
	110	-
	111	-
Timer5	000	TI05
	001	LPUART1_RXD
	010	XTLF
	011	RCHF
	100	RCLP
	101	-
	110	-
	111	-
Timer6	000	TI06
	001	-
	010	-
	011	-
	100	-
	101	-
	110	-
	111	-
Timer7	000	TI07
	001	-
	010	-
	011	-
	100	-
	101	-
	110	-

	111	-

表 31-2 定时器捕捉源选择

TAU1定时器	CAPSEL	被捕捉信号
Timer0	000	TI10
	001	UART0_RXD
	010	UART1_RXD
	011	XTLF
	100	UART2_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer1	000	TI11
	001	UART3_RXD
	010	UART4_RXD
	011	XTLF
	100	UART5_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer2	000	TI12
	001	UART0_RXD
	010	UART1_RXD
	011	XTLF
	100	UART2_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer3	000	TI13
	001	UART3_RXD
	010	UART4_RXD
	011	XTLF
	100	UART5_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP

表 31-3 定时器捕捉源选择

单次捕捉和连续捕捉

捕捉模式下CAPONCE寄存器可以选择单次捕捉模式或连续捕捉模式。当CAPONCE=1时，定时器完成一个捕捉周期后（周期捕捉完成一个周期捕捉，脉宽捕捉完成电平宽度捕捉），定时器自动关闭。当CAPONCE=0，定时器连续工作，直到软件关闭。

31.3.8 预分频

定时器可以对计数源和捕捉源进行预分频。

当对计数源进行预分频时，每隔特定数量的计数事件，才会进行一次计数值递增。下图以TI00上升沿计数、2分频为例：

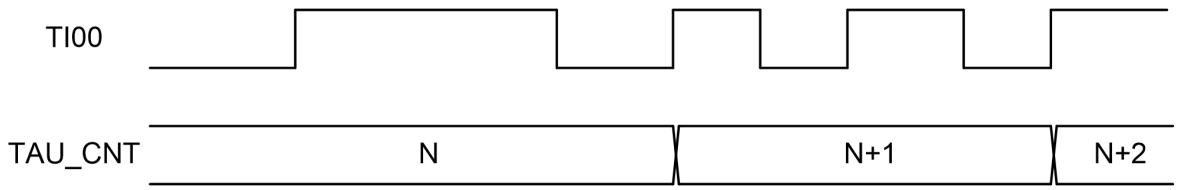


图 31-16TI00 二分频计数

当对捕捉源进行预分频时，每隔特定数量的捕捉事件，才会进行一次计数值捕捉。下图是对TI00上升沿进行捕捉、2分频的例子：

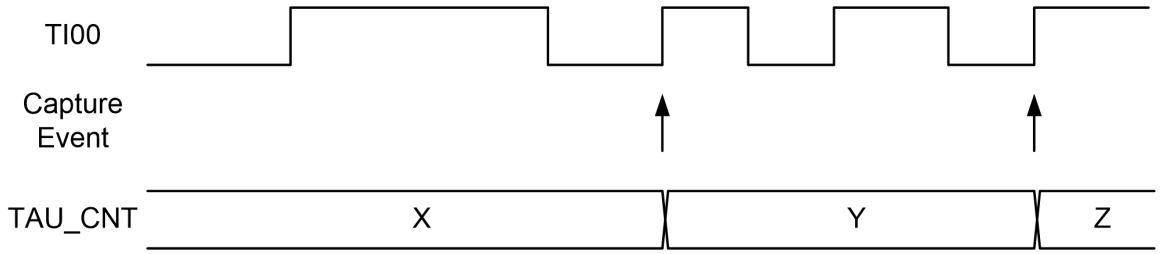


图 31-17TI00 二分频捕捉

31.3.9 16 位 PWM 输出

单个TAU支持独立16bitPWM输出，启动PWM之后定时器从初值开始计数，当计数值等于比较寄存器时，PWM输出置1，当计数值溢出时PWM输出置0，16bit PWM如下图所示：

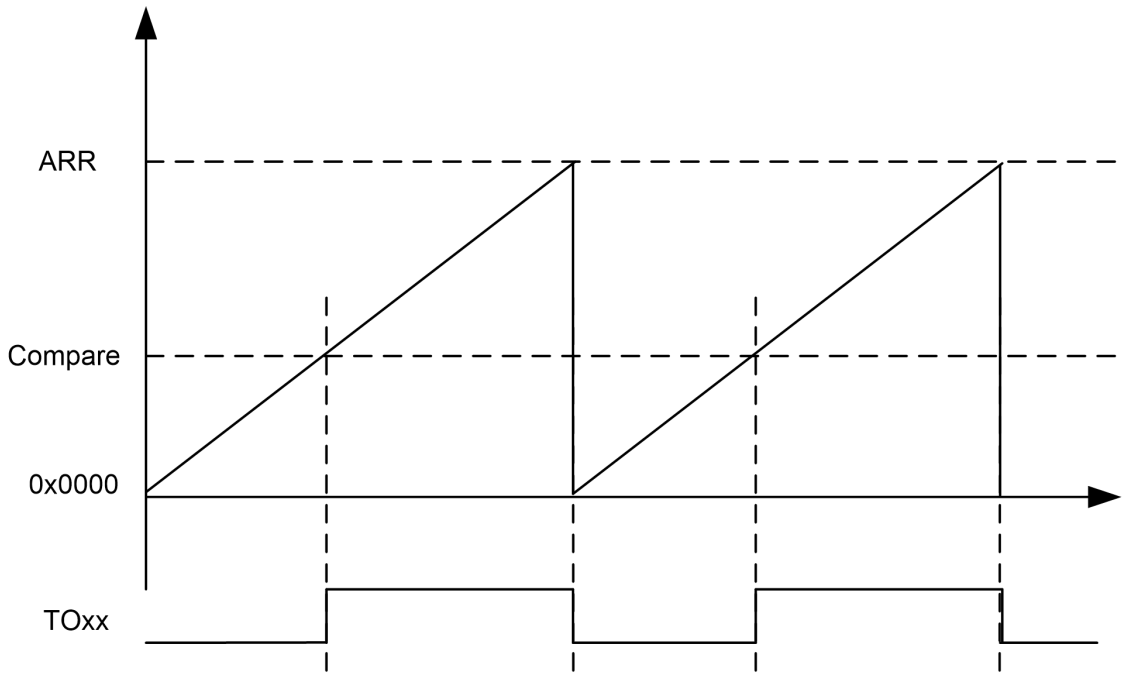


图 31-1816 位 PWM 输出示意图

PWM周期由ARR寄存器决定：

$$Period_{PWM} = T_{CNT} \times (\mathrm{ARR}+1)$$

PWM占空比由初值寄存器和比较寄存器共同决定：

$$Duty_{PWM} = T_{CNT} \times \frac{(ARR+1)-CCR}{ARR+1}$$

其中T_{CNT}为计数时钟周期。输出PWM信号极性可以取反，由OPOL寄存器控制。以上公式为输出极性不取反的情况，如果输出极性取反，则占空比变为1-$Duty_{PWM}$。

31.3.10 级联 PWM 输出

通过级联模式，两个独立定时器可以组合输出更为复杂的PWM波形。

比如将TAU0x和TAU0y级联，以TAU0x为主机，TAU0y采用触发启动模式，并且配置为PWM输出。从机接收到触发信号后开始计数，计数到ARR后自动清零并停止计数，直到下一次触发信号到来。下图中TAU0y的TO能够输出以ARR0x为周期，高电平宽度为ARR0y-CCR0y，前半部分低电平宽度为CCR0x+CCR0y，后半部分低电平宽度为ARR0x-ARR0y-CCR0x的特殊非对称PWM波形。

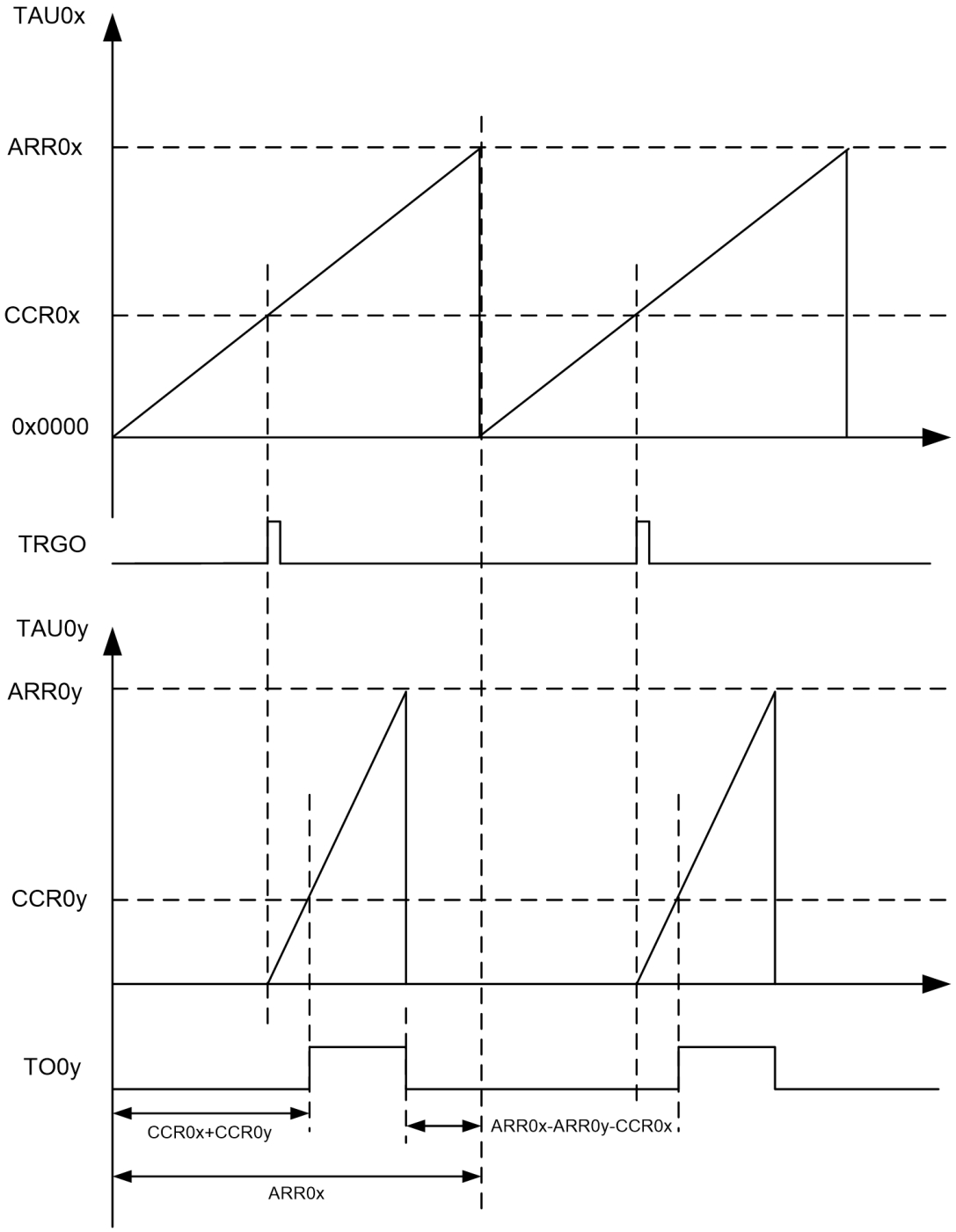


图 31-19 非对称级联 PWM 输出示意图

使用级联PWM输出时，软件必须合理配置主机和从机的ARR值，以获得符合期望的波形输出。下图为从机ARR大于主机ARR的例子。

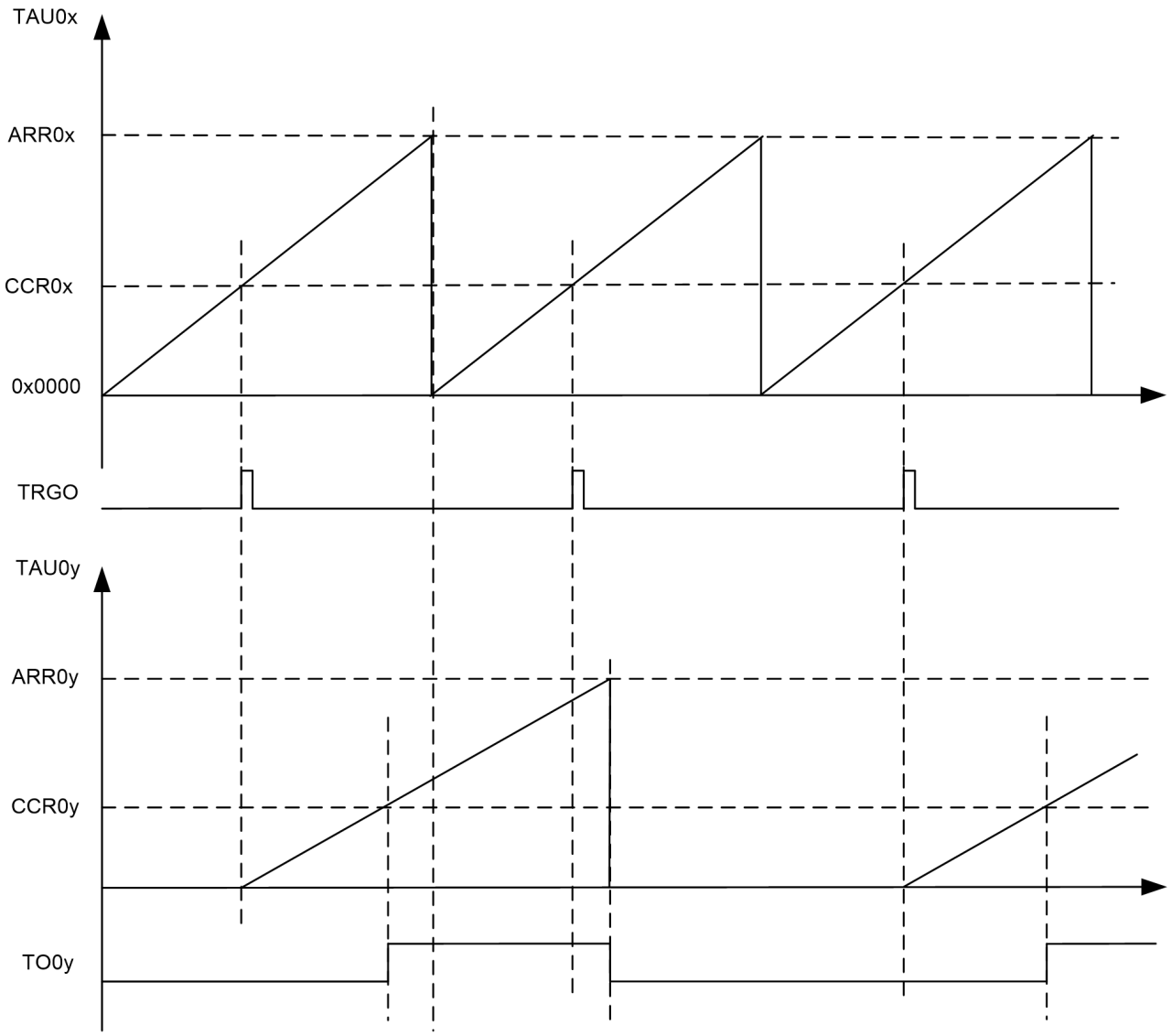


图 31-20 从机 ARR 大于主机的情况

级联定时器再搭配第三个独立定时器，可以输出两路非交叠PWM，如下图所示。

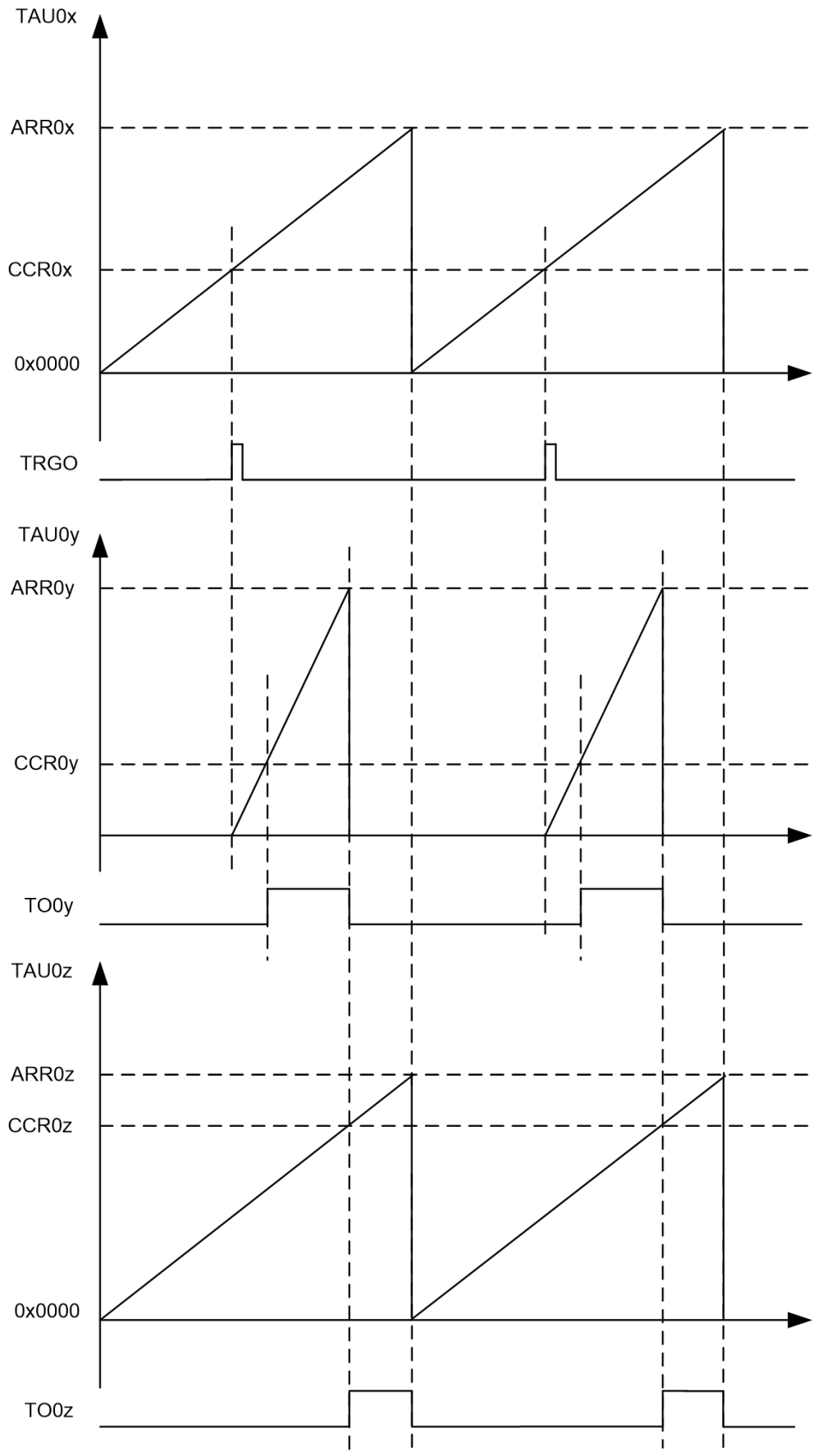


图 31-21 非对称非交叠 PWM 输出示意图

31.3.11 外部引脚输入数字滤波

扩展定时器的外部引脚输入TIxx可以选择是否使能数字滤波，以抑制信号噪声。使能数字滤波的情况下，使用APBCLK连续3次采样相同才认为是输入变化有效；不使能数字滤波时，扩展定时器直接对管脚输入进行采样。

数字滤波示意图如下：

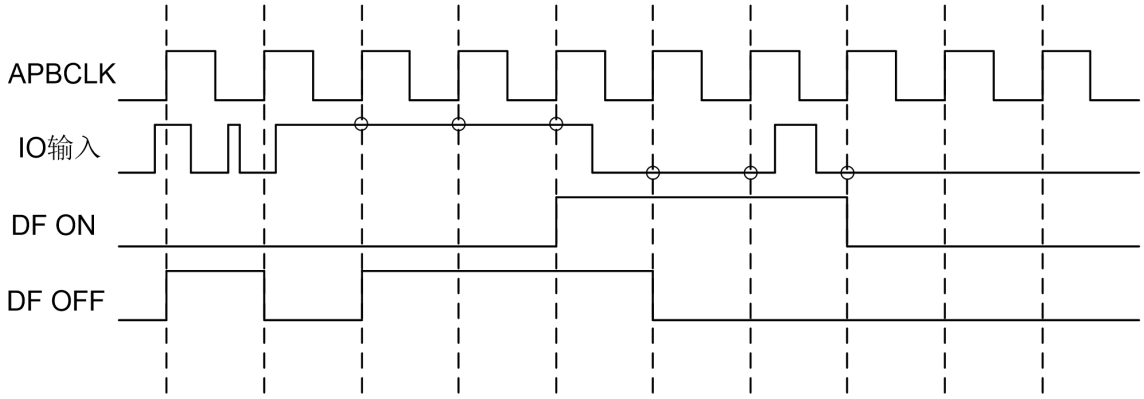


图 31-22TIxx 输入数字滤波示意图

31.4 寄存器

TAU模块基地址：0x40015800

offset 地址	名称	符号
0x00	TAU0 控制寄存器	TAU0_CR
0x04	TAU1 控制寄存器	TAU1_CR
0x10	TAU0 通道 x 配置寄存器	TAU0_TxCFGR
0x14		
0x18		
0x1C		
0x20		
0x24		
0x28		
0x2C		
0x30	TAU0 通道 x 模式寄存器	TAU0_TxMDR
0x34		
0x38		
0x3C		
0x40		
0x44		
0x48		
0x4C		
0x50	TAU0 重载寄存器	TAU0_TxARR
0x54		
0x58		
0x5C		
0x60		
0x64		
0x68		
0x6C		
0x70	TAU0 捕捉比较寄存器	TAU0_TxCCR
0x74		
0x78		
0x7C		
0x80		
0x84		
0x88		
0x8C		
0x90	TAU0 中断使能寄存器	TAU0_TxIER
0x94		
0x98		
0x9C		
0xA0		
0xA4		

0xA8		
0xAC		
0xB0	TAU0 中断标志寄存器	TAU0_TxISR
0xB4		
0xB8		
0xBC		
0xC0		
0xC4		
0xC8		
0xCC		
0xD0	TAU0 中断标志寄存器	TAU0_TxCNT
0xD4		
0xD8		
0xDC		
0xE0		
0xE4		
0xE8		
0xEC		
0x110	TAU1 通道 x 配置寄存器	TAU1_TxCFGR
0x114		
0x118		
0x11C		
0x120		
0x124		
0x128		
0x12C		
0x130	TAU1 通道 x 模式寄存器	TAU1_TxMDR
0x134		
0x138		
0x13C		
0x140		
0x144		
0x148		
0x14C		
0x150	TAU1 重载寄存器	TAU1_TxARR
0x154		
0x158		
0x15C		
0x160		
0x164		
0x168		
0x16C		
0x170	TAU1 捕捉比较寄存器	TAU1_TxCCR
0x174		
0x178		
0x17C		
0x180		
0x184		

0x188		
0x18C		
0x190	TAU1 中断使能寄存器	TAU1_TxIER
0x194		
0x198		
0x19C		
0x1A0		
0x1A4		
0x1A8		
0x1AC		
0x1B0	TAU1 中断标志寄存器	TAU1_TxISR
0x1B4		
0x1B8		
0x1BC		
0x1C0		
0x1C4		
0x1C8		
0x1CC		
0x1D0	TAU1 计数值寄存器	TAU1_TxCNT
0x1D4		
0x1D8		
0x1DC		
0x1E0		
0x1E4		
0x1E8		
0x1EC		

31.4.1 TAU0 控制寄存器（TAU0_CR）

名称	TAU0_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EN[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:8	--	未实现：读为0

位号	位名	说明
7:0	EN	定时器使能，每个 bit 单独控制一路定时器，1 使能 Bit0：Timer0 Bit1：Timer1 …… Bit7：Timer7 *注意，应在完成所有配置寄存器写入后，最后置位 EN 寄存器。EN 置位后，定时器立即开始工作。*

31.4.2 TAU0 通道 x 配置寄存器（TAU0_TxCFGR）

名称	TAU0_TxCFGR (x=0~7)							
offset	0x10~0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	CAPCLR	CAPONCE	NF	CAPEDGE		CNTEDGE	
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PRESCALE1							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PRESCALE2							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OPOL	TS		CAPSEL[2:0]			CNTSEL[1:0]	
位权限	R/W-0	R/W-00		R/W-000			R/W-00	

位号	位名	说明
31	--	未实现，读为0
30	CAPCLR	清零捕捉模式 0：自由计数捕捉模式 1：清零捕捉模式
29	CAPONCE	单次捕捉控制 1：单次捕捉有效，在捕捉到一次脉冲周期后计数器停止，若需要再次捕捉需重新启动 0：连续捕捉
28	NF	噪声滤波 0：关闭数字滤波 1：打开数字滤波
27:26	CAPEDGE	捕捉源边沿选择 00/11：上升沿和下降沿 01：上升沿 10：下降沿
25:24	CNTEDGE	计数源边沿选择 00/11：上升沿和下降沿 01：上升沿 10：下降沿

位号	位名	说明
23:16	PRESCALE1	计数源预分频，分频值=PRESCALE1 + 1
15:8	PRESCALE2	捕捉源预分频，分频值=PRESCALE2 + 1
7	OPOL	输出极性选择 0：输出不取反 1：输出取反
6:5	TS	从机触发源选择（Trigger Select） 仅在从机模式下（slave）有效，每个定时器详细的触发选择参见31.3.632位级联模式
4:2	CAPSEL	捕捉源选择寄存器 000：外部引脚输入（Tixx） 其他：参见31.3.7捕捉功能
1:0	CNTSEL	计数源选择寄存器 00/11：APBCLK 01：外部引脚输入（Tixx） 10：捕捉信号输入（CAPSEL选择后的输出）

31.4.3 TAU0 通道 x 模式寄存器（TAU0_TxMDR）

名称	TAU0_TxMDR (x=0~7)							
offset	0x30~0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			SLVMD	TOEN	SLV	MD	
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-00	

位号	位名	说明
31:5	--	未实现，读为0
4	SLVMD	Slave模式选择，仅SLV=1的情况下有效 0：触发递增模式（trigger to increment），每个触发信号导致从机计数器加1 1：触发启动模式（trigger to start），每个触发信号导致从机从0开始计数，直到ARR停止
3	TOEN	TO输出使能 0：关闭TO输出 1：使能TO输出
2	SLV	Slave模式使能 0：主机模式 1：从机模式，可以接收其他Timer发送的TRGO *注意：Timer0无从机模式，此bit无效。*

位号	位名	说明
1:0	MD	定时器工作模式 00：计数模式 01：比较输出模式（PWM） 10：输入捕捉模式（周期捕捉） 11：输入捕捉模式（脉宽捕捉）

31.4.4 TAU0 重载寄存器（TAU0_TxARR）

名称	TAU0_TxARR (x=0~7)							
地址	0x50~0x6C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-00000000							

位号	位名	说明
31:16	---	未实现，读为0
15:0	ARR	自动重载寄存器 计数器计数值等于 ARR 时产生溢出信号并回到 0 开始重新计数。

31.4.5 TAU0 捕捉比较寄存器（TAU0_TxCCR）

名称	TAU0_TxCCR(x=0~7)							
地址	0x70~0x8C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR[7:0]							
位权限	R/W-00000000							

位号	位名	说明

位号	位名	说明
31:16	---	未实现，读为0
15:0	CCR	捕捉比较寄存器 捕捉模式下，当捕捉

31.4.6 TAU0 中断使能寄存器（TAU0_TxIER）

名称	TAU0_TxIER(x=0~7)							
地址	0x90~0xAC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	---					CMPIE	CAPIE	OVIE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	位名	说明
31:3	---	未实现：读为 0
2	CMPIE	定时器比较中断使能 1 = 使能 0 = 禁止
1	CAPIE	定时器捕捉中断使能 1 = 使能 0 = 禁止
0	OVIE	定时器溢出中断使能 1 = 使能 0 = 禁止

31.4.7 TAU0 中断标志寄存器（TAU0_TxISR）

名称	TAU0_TxISR(x=0~7)							
地址	0xB0~0xCC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	EDGESTA	CMPIF	CAPIF	OVIF
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:4	---	未实现：读为 0
3	EDGESTA	捕捉沿状态标志，软件写 1 清 0 1 = 脉冲宽度捕捉模式时表示捕捉到下沿 0 = 脉冲宽度捕捉模式时表示捕捉到上沿
2	CMPIF	比较状态，软件写 1 清 0 1 =当前计数器的值大于等于比较寄存器的值 0 =当前计数器的值小于比较寄存器的值
1	CAPIF	定时器捕捉产生信号，软件写1清0 1 = 捕捉到指定的沿 0 = 未捕捉到指定的沿
0	OVIF	定时器溢出信号，当计数器的值达到ARR时将置位，软件写1清0 1 = 产生计数溢出 0 = 未产生溢出

31.4.8 TAU0 计数值寄存器（TAU0_TxCNTR）

名称	TAU0_TxCNTR (x=0~7)							
地址	0xD0~0xEC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:16	---	未实现：读为 0
15:0	CNT	计数器当前计数值，只读

31.4.9 TAU1 控制寄存器（TAU1_CR）

名称	TAU1_CR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

名称	TAU1_CR							
offset	0x04							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EN[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:8	--	未实现：读为0
7:0	EN	定时器使能，每个 bit 单独控制一路定时器，1 使能 Bit0：Timer0 Bit1：Timer1 …… Bit3：Timer3 *注意，应在完成所有配置寄存器写入后，最后置位 EN 寄存器。EN 置位后，定时器立即开始工作。*

31.4.10 TAU1 通道 x 配置寄存器（TAU1_TxCFGR）

名称	TAU1_TxCFGR (x=0~7)							
offset	0x110~0x12C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	CAPCLR	CAPONCE	NF	CAPEDGE		CNTEDGE	
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PRESCALE1							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PRESCALE2							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OPOL	RFUI		CAPSEL[2:0]			CNTSEL[1:0]	
位权限	R/W-0	R/W-00		R/W-000			R/W-00	

位号	位名	说明
31	--	未实现，读为0
30	CAPCLR	清零捕捉模式 0：自由计数捕捉模式 1：清零捕捉模式

位号	位名	说明
29	CAPONCE	单次捕捉控制 1：单次捕捉有效，在捕捉到一次脉冲周期后计数器停止，若需要再次捕捉需重新启动 0：连续捕捉
28	NF	噪声滤波 0：关闭数字滤波 1：打开数字滤波
27:26	CAPEDGE	捕捉源边沿选择 00/11：上升沿和下降沿 01：上升沿 10：下降沿
25:24	CNTEDGE	计数源边沿选择 00/11：上升沿和下降沿 01：上升沿 10：下降沿
23:16	PRESCALE1	计数源预分频，分频值=PRESCALE1 + 1
15:8	PRESCALE2	捕捉源预分频，分频值=PRESCALE2 + 1
7	OPOL	输出极性选择 0：输出不取反 1：输出取反
6:5	RFUI	保留位
4:2	CAPSEL	捕捉源选择寄存器 000：外部引脚输入（Tixx） 其他：参见31.3.7捕捉功能
1:0	CNTSEL	计数源选择寄存器 00/11：APBCLK 01：外部引脚输入（Tixx） 10：捕捉信号输入（CAPSEL选择后的输出）

31.4.11 TAU1 通道 x 模式寄存器（TAU1_TxMDR）

名称	TAU1_TxMDR (x=0~7)							
offset	0x130~0x14C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			RFUI	TOEN	SLV	MD	
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-00	

位号	位名	说明
31:5	--	未实现，读为0

位号	位名	说明
4	RFUI	保留位
3	TOEN	TO输出使能 0：关闭TO输出 1：使能TO输出
2	SLV	Slave模式使能 0：主机模式 1：从机模式，可以接收其他Timer发送的TRGO *注意：TAU1无从机模式，此bit无效。*
1:0	MD	定时器工作模式 00：计数模式 01：比较输出模式（PWM） 10：输入捕捉模式 11：RFU

31.4.12 TAU1 重载寄存器（TAU1_TxARR）

名称	TAU1_TxARR (x=0~7)							
地址	0x150~0x16C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-00000000							

位号	位名	说明
31:16	---	未实现，读为0
15:0	ARR	自动重载寄存器 计数器计数值等于 ARR 时产生溢出信号并回到 0 开始重新计数。

31.4.13 TAU1 捕捉比较寄存器（TAU1_TxCCR）

名称	TAU1_TxCCR (x=0~7)							
地址	0x170~0x18C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR[7:0]							
位权限	R/W-00000000							

位号	位名	说明
31:16	---	未实现，读为0
15:0	CCR	捕捉比较寄存器 捕捉模式下，当捕捉

31.4.14 TAU1 中断使能寄存器（TAU1_TxIER）

名称	TAU1_TxIER (x=0~7)							
地址	0x190~0x1AC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	---					CMPIE	CAPIE	OVIE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	位名	说明
31:3	---	未实现：读为 0
2	CMPIE	定时器比较中断使能 1 = 使能 0 = 禁止
1	CAPIE	定时器捕捉中断使能 1 = 使能 0 = 禁止
0	OVIE	定时器溢出中断使能 1 = 使能 0 = 禁止

31.4.15 TAU1 中断标志寄存器（TAU1_TxISR）

名称	TAU1_TxISR (x=0~7)							
地址	0x1B0~0x1CC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	---				EDGESTA	CMPIF	CAPIF	OVIF
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:4	---	未实现：读为 0
3	CMPIF	比较状态，软件写 1 清 0 1 =当前计数器的值大于等于比较寄存器的值 0 =当前计数器的值小于比较寄存器的值
2	EDGESTA	捕捉沿状态标志，软件写 1 清 0 1 = 脉冲宽度捕捉模式时表示捕捉到下沿 0 = 脉冲宽度捕捉模式时表示捕捉到上沿
1	CAPIF	定时器捕捉产生信号，软件写1清0 1 = 捕捉到指定的沿 0 = 未捕捉到指定的沿
0	OVIF	定时器溢出信号，当计数器的值达到ARR时将置位，软件写1清0 1 = 产生计数溢出 0 = 未产生溢出

31.4.16 TAU1 计数值寄存器（TAU1_TxCNTR）

名称	TAU1_TxCNTR (x=0~7)							
地址	0x1D0~0x1EC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:16	---	未实现：读为 0
15:0	CNT	计数器当前计数值，只读

32 32 位基本定时器（BSTIM32）

32.1 概述

FM33LG0A包含1个32位基本定时器。

基本定时器包含一个32bit自动重载计数器及一个可编程预分频器。

基本定时器主要用来产生系统时基，也可以产生触发事件来驱动ADC采样。

32.2 主要特性

- 32bit向上计数自动重载计数器
- 32bit可编程预分频器，支持实时调整计数时钟分频
- ADC定时触发功能
- 计数器溢出时产生中断

32.3 结构框图

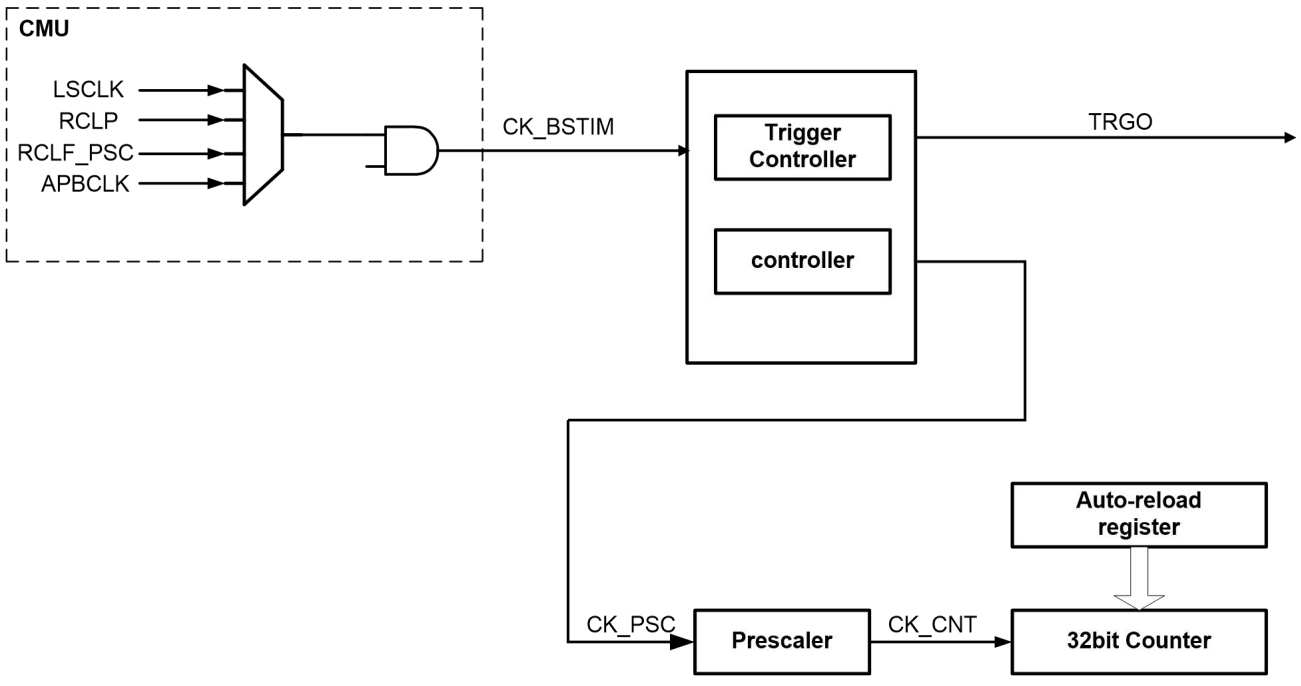


图 32-1 32 位基本定时器结构框图

32.4 功能描述

32.4.1 定时单元

基本定时器的定时单元由一个32位计数器和自动重载寄存器组成。计数器向上计数。计数时钟可以通过32位预分频器对APBCLK、LSCLK、RCLP进行分频后得到。

计数器、自动重载寄存器预分频寄存器都可以由软件改写或读取，即使在计数器正在运行时也是如此。

定时单元包含如下寄存器：

- 计数器（BSTIM32_CNT）
- 预分频寄存器（BSTIM32_PSC）
- 自动重载寄存器（BSTIM32_ARR）

ARR包含preload功能，软件读写ARR可以直接起效，或者只是访问其缓存，通过ARPE（Auto Reload Preload Enable）寄存器控制。当ARPE=1时，软件读写ARR都是访问其缓存寄存器，当update event（BSTIM32_CNT上溢出或者下溢出）发生时，会将缓存寄存器内的数据更新到ARR中。软件也可以通过寄存器操作主动触发ARR更新。

BSTIM_CNT工作时钟由BSTIM_PSC产生的分频时钟驱动，只有在计数器使能寄存器（CEN）置位时，CNT才开始计数。当CNT=ARR时，本轮计数结束，发送update event。

BSTIM_PSC是一个同步预分频器，能够对APBCLK、LSCLK、RCLP进行1~2^{32}分频。PSC寄存器同样被缓存，改写PSC实际是改写缓存寄存器，只有当新的update event到来时，才会从缓存寄存器更新PSC。因此在CNT计数过程中，软件可以实时改写PSC。

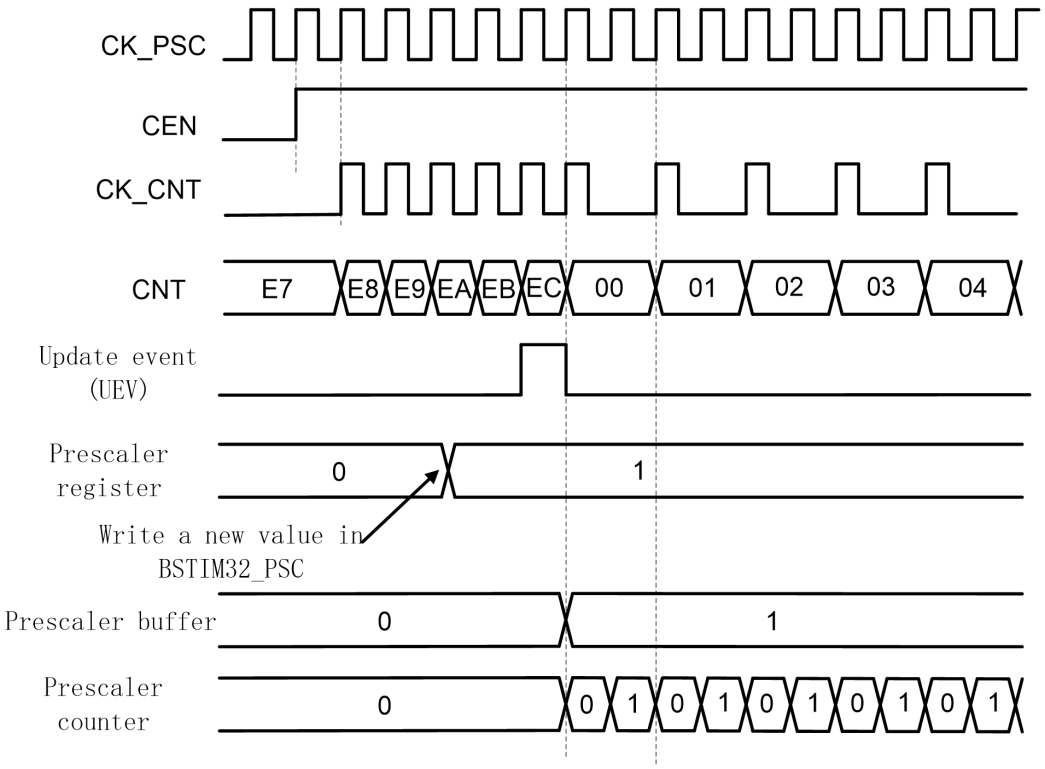


图 32-2 预分频从 1 变为 2 的波形

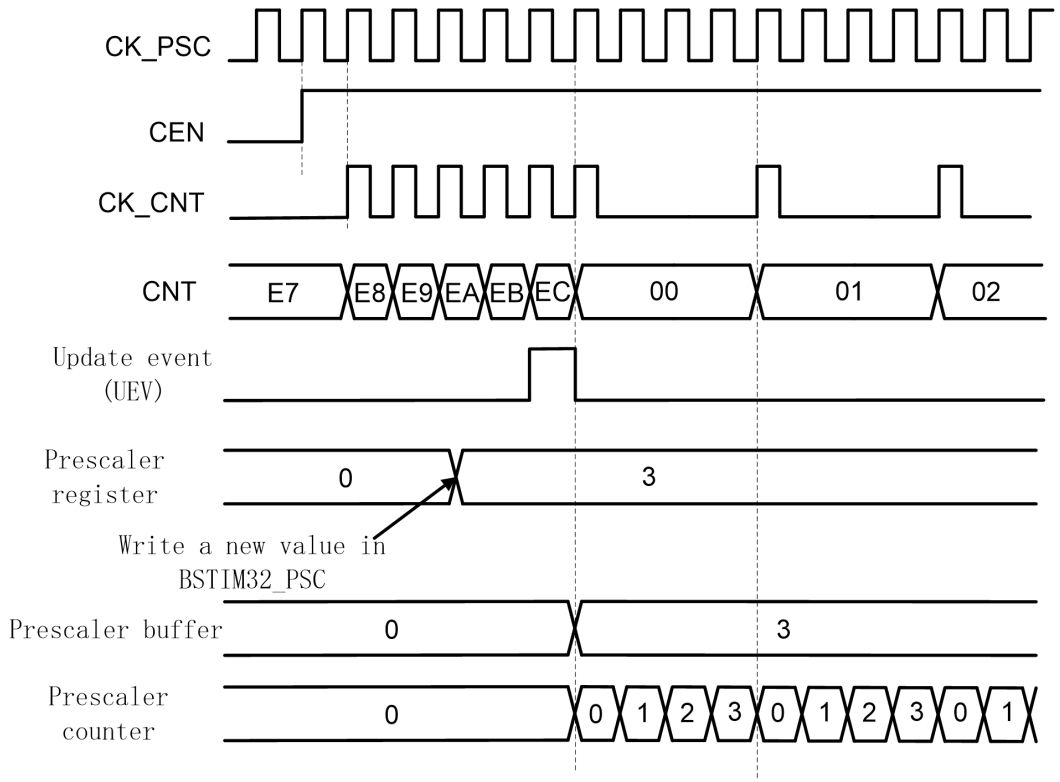


图 32-3 预分频从 1 变为 4 的波形

32.4.2 定时器工作模式

通用定时器只支持向上计数模式。

向上计数

此模式中，计数器使能后从0开始计数，直到CNT=ARR，产生溢出事件，然后重新从0开始计数。

软件可以通过设置UG寄存器直接触发update event，此时CNT和预分频计数器自动清零。设置UG寄存器不会触发UIF（Update Interrupt Flag）中断标志置位。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- BSTIM_ARR更新为缓存中的值
- BSTIM_PSC更新为缓存中的值

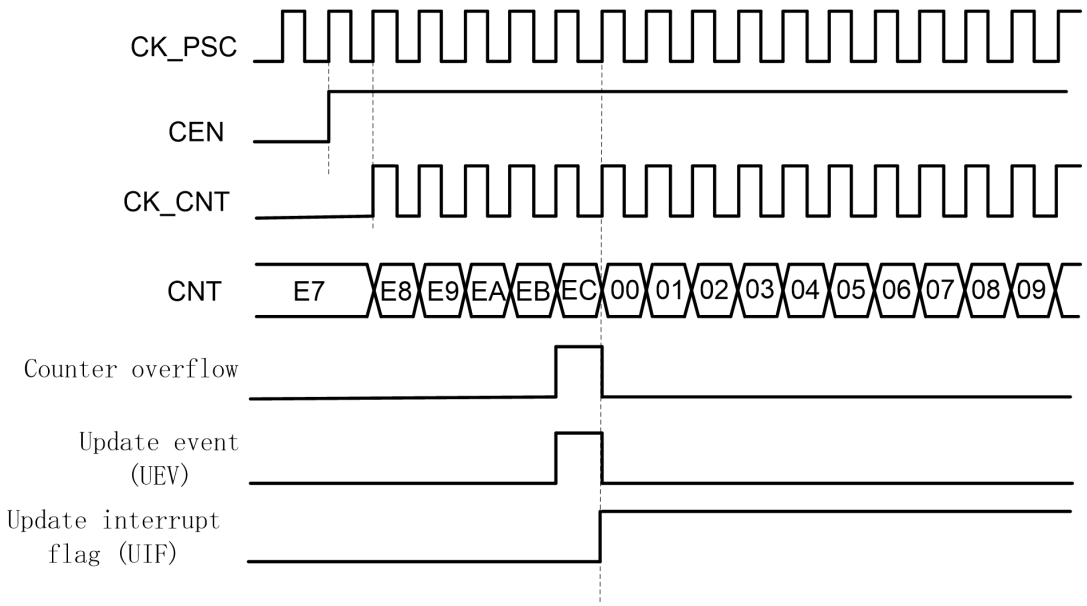


图 32-4 向上计数波形，内部时钟不分频

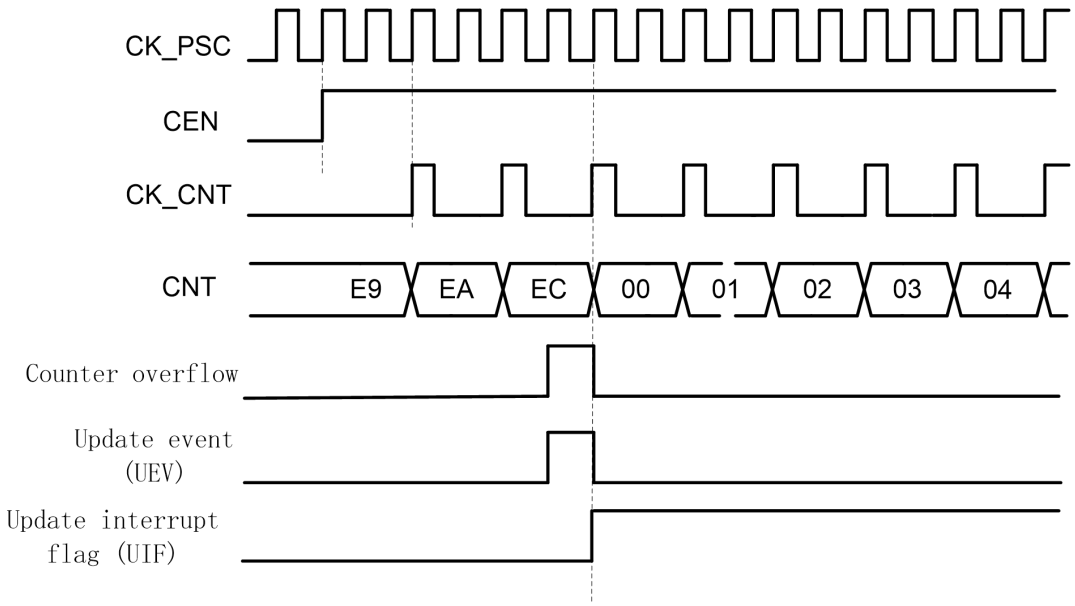


图 32-5 向上计数波形，内部时钟 2 分频

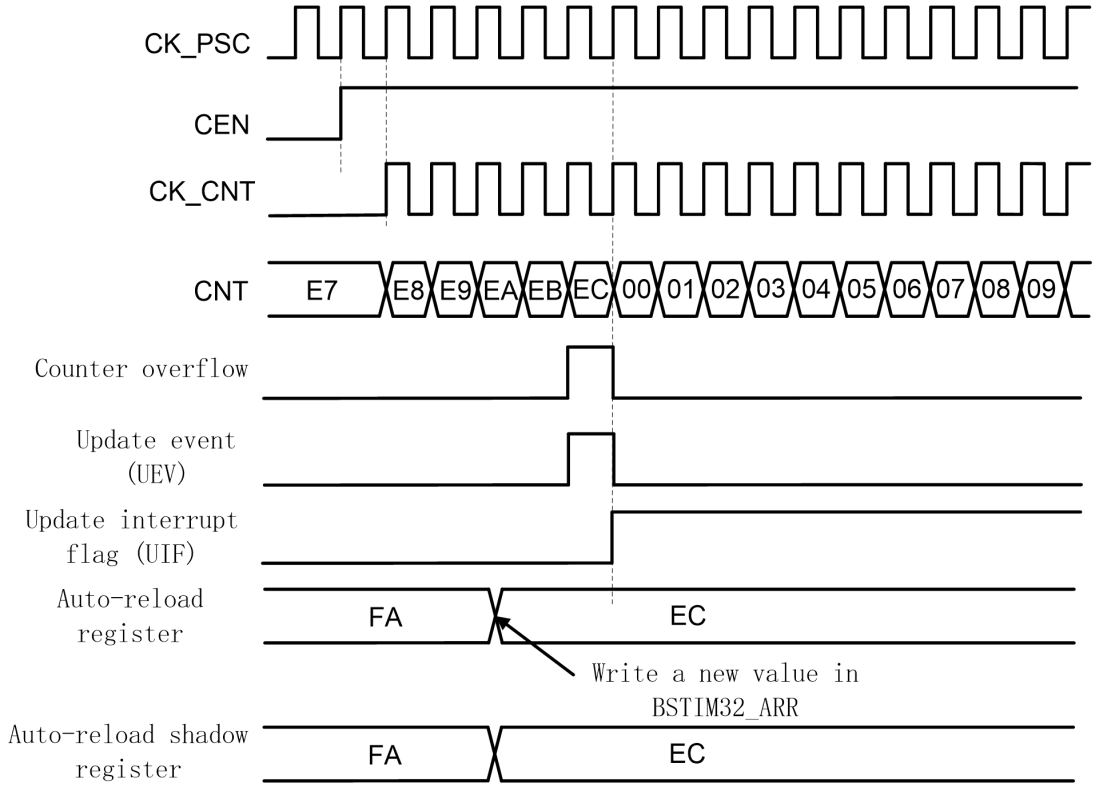


图 32-6ARPE=0（ARR 没有预装载）时的更新事件

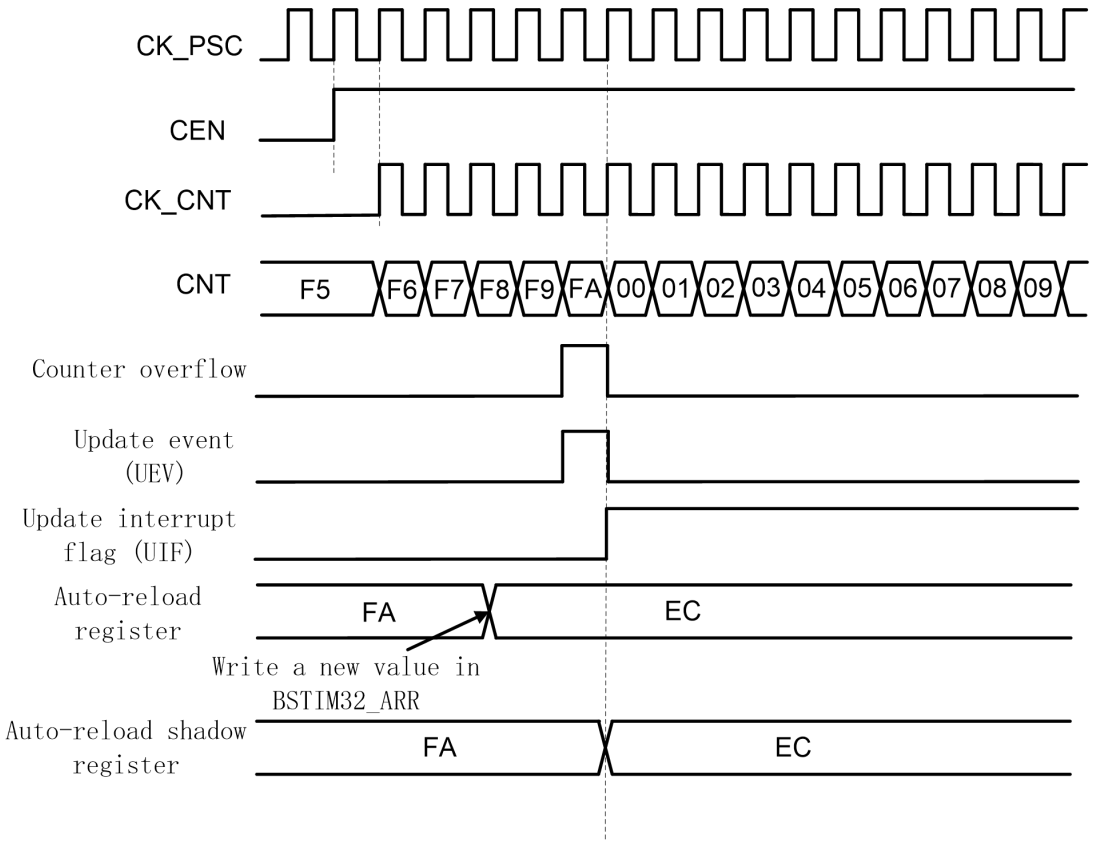


图 32-7 ARPE=1（ARR 预装载）时的更新事件

32.4.3 计数器工作时钟

BSTIM使用内部时钟工作，CEN、UG等寄存器位都是软件控制

软件操作UG寄存器后，update信号经过CLK_PSC同步后，计数器值将被重新初始化。

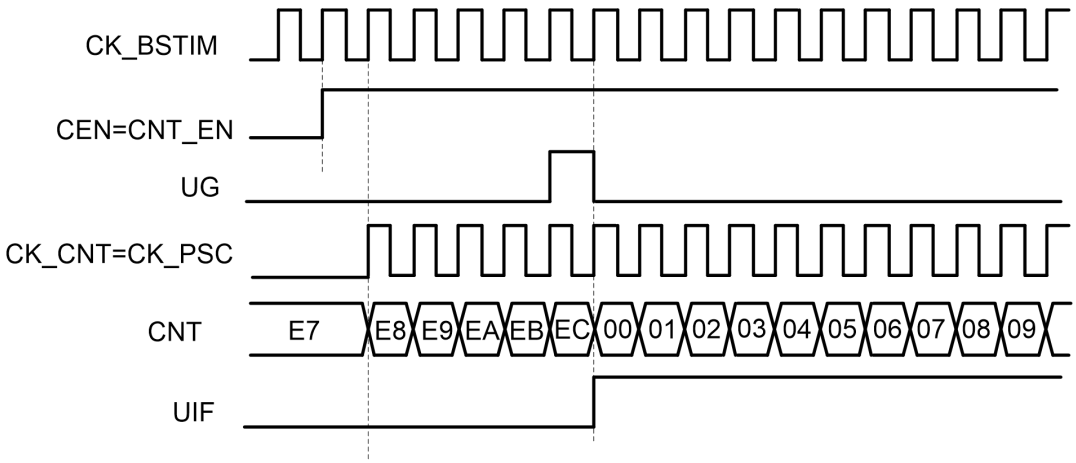


图 32-8 内部时钟源模式，时钟分频因子为 1

32.4.4 Debug 模式

当Cortex-M0进入debug模式后，定时器可以停止或继续工作，其行为由DBG模块的DBG_BT32_STOP寄存器定义。

32.5 寄存器

BSTIM32模块基地址：0x40016000

offset 地址	名称	符号
BSTIM32(模块基地址：0x40016000)		
0x00000000	BSTIM32 控制寄存器 1 （BSTIM32 Control Register1）	BSTIM32_CR1
0x00000004	BSTIM32 控制寄存器 2 （BSTIM32 Control Register2）	BSTIM32_CR2
0x0000000C	BSTIM32 中断使能寄存器 （BSTIM 32Interrupt Enable Register）	BSTIM32_IER
0x00000010	BSTIM32 中断标志寄存器 （BSTIM32 Interrupt Status Register）	BSTIM32_ISR
0x00000014	BSTIM32 事件产生寄存器 （BSTIM32 Event Generation Register）	BSTIM32_EGR
0x00000024	BSTIM32 计数器寄存器 （BSTIM32 Counter Register）	BSTIM32_CNT
0x00000028	BSTIM32 预分频寄存器 （BSTIM32 Prescaler Register）	BSTIM32_PSC
0x0000002C	BSTIM32 自动重载寄存器 （BSTIM32 Auto-Reload Register）	BSTIM32_ARR

32.5.1 BSTIM32 控制寄存器 1（BSTIM32_CR1）

名称	BSTIM32_CR1							
Offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	-			OPM	URS	UDIS	CEN
位权限	R/W-0	U-0			R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:8	-	RFU，未实现，读为 0
7	ARPE	Auto-reload 预装载使能 (Auto-Reload Preload Enable) 0：ARR 寄存器不使能 preload 1：ARR 寄存器使能 preload
6:4	-	RFU，未实现，读为 0
3	OPM	单脉冲输出模式 (One Pulse Mode) 0：Update Event 发生时计数器不停止

位号	助记符	功能描述
		1：Update Event 发生时计数器停止（自动清零 CEN）
2	URS	更新请求选择 (Update Request Select) 0：以下事件能够产生 update 中断 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 1：仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求
1	UDIS	禁止 update (Update Disable) 0：使能 update 事件；以下事件发生时产生 update 事件 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 1：禁止 update 事件，不更新 shadow 寄存器。当 UG 置位或从机控制器收到硬件 reset 时重新初始化计数器和预分频器。
0	CEN	计数器使能 (Counter Enable) 0：计数器关闭 1：计数器使能 注意：外部触发模式可以自动置位 CEN

32.5.2 BSTIM32 控制寄存器 2（BSTIM32_CR2）

名称	BSTIM32_CR2							
Offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	MMS			-			
位权限	U-0	R/W-000			U-0			

位号	助记符	功能描述
31:7	-	RFU，未实现，读为 0
6:4	MMS	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源(Master Mode Select) 000：BSTIM_EGR 的 UG 寄存器被用作 TRGO 001：计数器使能信号 CNT_EN 被用作 TRGO，可用于同时启动多个定时器 010：UE（update event）信号被用作 TRGO 011/100/111：RFU 注意：从机定时器或 ADC 必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
3:0	-	RFU，未实现，读为 0

32.5.3 BSTIM32 中断使能寄存器（BSTIM32_IER）

名称	BSTIM32_IER							
Offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							UIE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	UIE	Update 事件中断使能(Update event Interrupt Enable) 0：禁止 Update 事件中断 1：允许 Update 事件中断

32.5.4 BSTIM32 中断标志寄存器（BSTIM32_ISR）

名称	BSTIM32_ISR							
Offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							UIF
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	UIF	Update 事件中断标志，硬件置位，软件写 1 清零。(Update event Interrupt Flag,write 1 to flag) 当以下事件发生时，UIF 置位，并更新 shadow 寄存器 -重复计数器=0，并且 UDIS=0 的情况下，计数器发生溢出 -URS=0 且 UDIS=0 的情况下，软件置位 UG 寄存器初始化计数器

位号	助记符	功能描述
		-URS=0 且 UDIS=0 的情况下，触发事件初始化计数器

32.5.5 BSTIM32 事件产生寄存器（BSTIM32_EGR）

名称	BSTIM32_EGR							
Offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							UG
位权限	U-0							W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	UG	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零(User Generate) 软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。 *注意：由于 BSTIM 计数器工作时钟与系统总线时钟 APBCLK 是互相独立的，当软件置位 UG 寄存器后，UIF 并不会立即置位，而是需要经过 CK_BSTIM 同步后才置位，两者之间的延迟与 APBCLK 和 CK_BSTIM 之间的频率及相位关系有关，并非确定值。*

32.5.6 BSTIM32 计数器寄存器（BSTIM32_CNT）

名称	BSTIM32_CNT							
Offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CNT[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CNT[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**CNT**	计数器值(Counter)

32.5.7 BSTIM32 预分频寄存器（BSTIM_PSC）

名称	BSTIM32_PSC							
Offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PSC[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PSC[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PSC[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**PSC**	计数器时钟（CK_CNT）预分频值(Counter Clock Prescaler) $f_{CK_CNT}=f_{CK_PSC}/(PSC[31:0]+1)$ 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

32.5.8 BSTIM32 自动重载寄存器（BSTIM32_ARR）

名称	BSTIM32_ARR							
Offset	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ARR[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ARR[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**ARR**	计数溢出时的自动重载值(Auto-Reload Register) 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

33 16 位基本定时器（BSTIM16）

33.1 概述

FM33FG0A包含1个16位基本定时器。

基本定时器包含一个16bit自动重载计数器及一个可编程预分频器。

基本定时器主要用来产生系统时基，也可以产生触发事件来驱动ADC采样。

33.2 主要特性

- 16bit向上计数自动重载计数器
- 16bit可编程预分频器，支持实时调整计数时钟分频
- ADC定时触发功能
- 计数器溢出时产生中断

33.3 结构框图

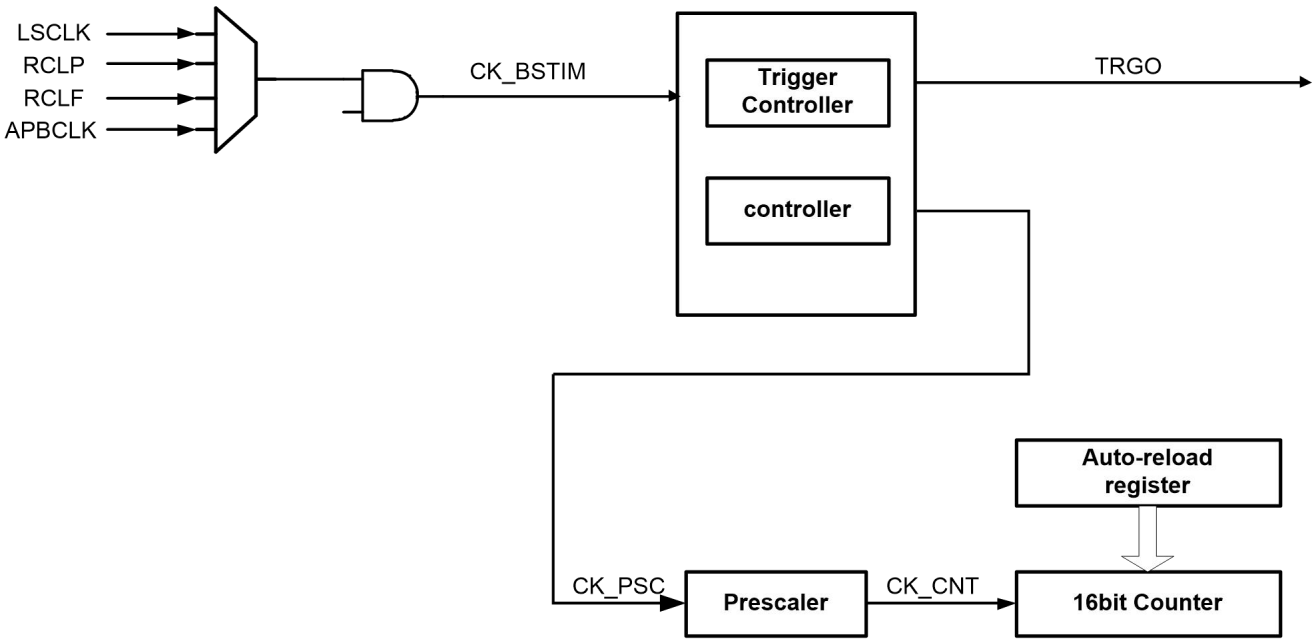


图 33-1 16 位基本定时器结构框图

33.4 功能描述

33.4.1 定时单元

基本定时器的定时单元由一个16位计数器和自动重载寄存器组成。计数器向上计数。计数时钟可以通过16位预分频器对APBCLK进行分频后得到。

计数器、自动重载寄存器预分频寄存器都可以由软件改写或读取，即使在计数器正在运行时也是如此。

定时单元包含如下寄存器：

- 计数器（BSTIM_CNT）
- 预分频寄存器（BSTIM_PSC）
- 自动重载寄存器（BSTIM_ARR）

ARR包含preload功能，软件读写ARR可以直接起效，或者只是访问其缓存，通过ARPE（Auto Reload Preload Enable）寄存器控制。当ARPE=1时，软件读写ARR都是访问其缓存寄存器，当update event（ATIM_CNT上溢出或者下溢出）发生时，会将缓存寄存器内的数据更新到ARR中。软件也可以通过寄存器操作主动触发ARR更新。

BSTIM_CNT工作时钟由BSTIM_PSC产生的分频时钟驱动，只有在计数器使能寄存器（CEN）置位时，CNT才开始计数。当CNT=ARR时，本轮计数结束，发送update event。

BSTIM_PSC是一个同步预分频器，能够对APBCLK进行1~65536分频。PSC寄存器同样被缓存，改写PSC实际是改写缓存寄存器，只有当新的update event到来时，才会从缓存寄存器更新PSC。因此在CNT计数过程中，软件可以实时改写PSC。

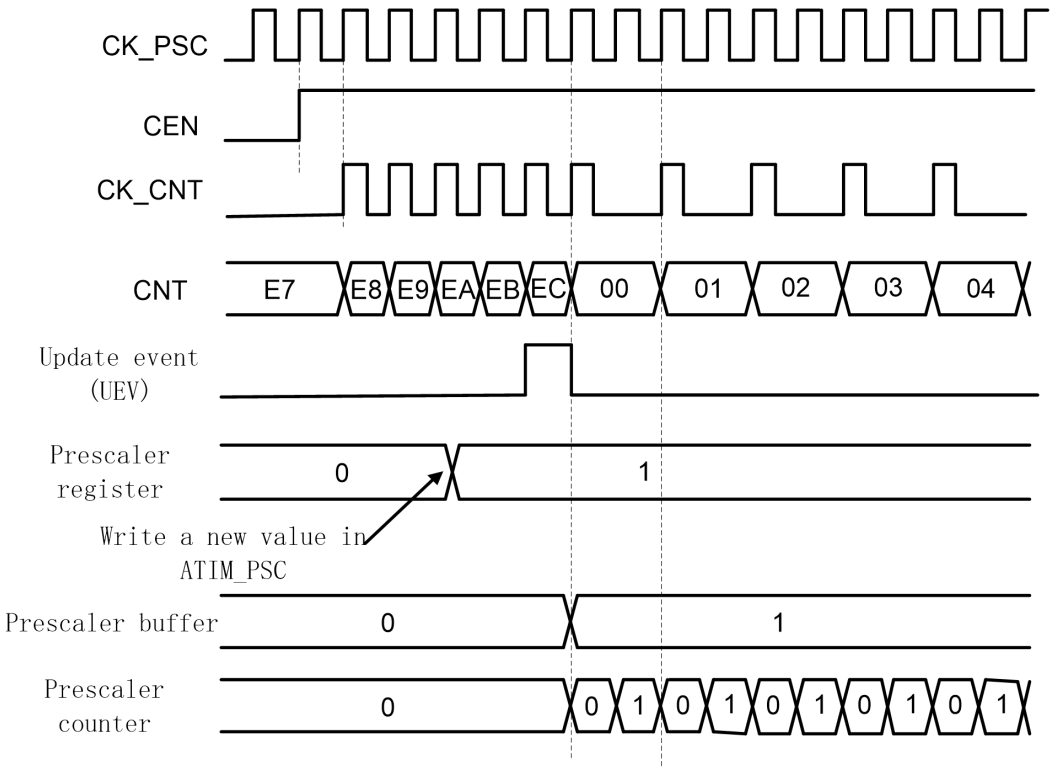


图 33-2 预分频从 1 变为 2 的波形

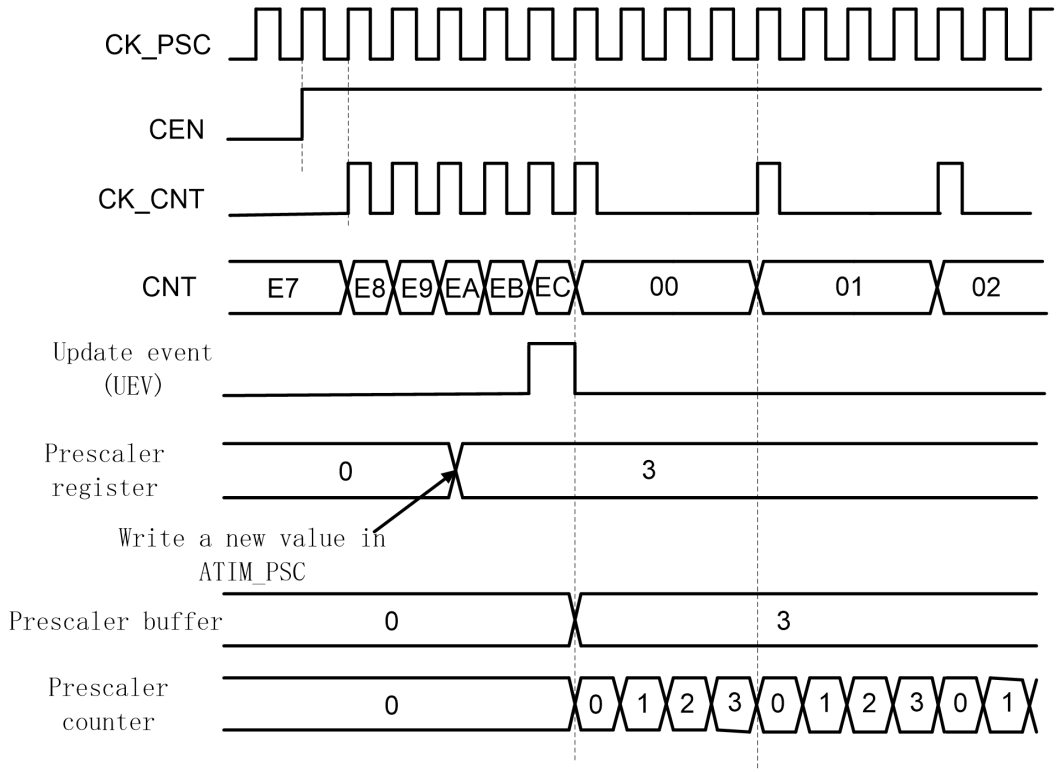


图 33-3 预分频从 1 变为 4 的波形

33.4.2 定时器工作模式

通用定时器只支持向上计数模式。

向上计数

此模式中，计数器使能后从0开始计数，直到CNT=ARR，产生溢出事件，然后重新从0开始计数。

软件可以通过设置UG寄存器直接触发update event，此时CNT和预分频计数器自动清零。设置UG寄存器不会触发UIF（Update Interrupt Flag）中断标志置位。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- BSTIM_ARR更新为缓存中的值
- BSTIM_PSC更新为缓存中的值

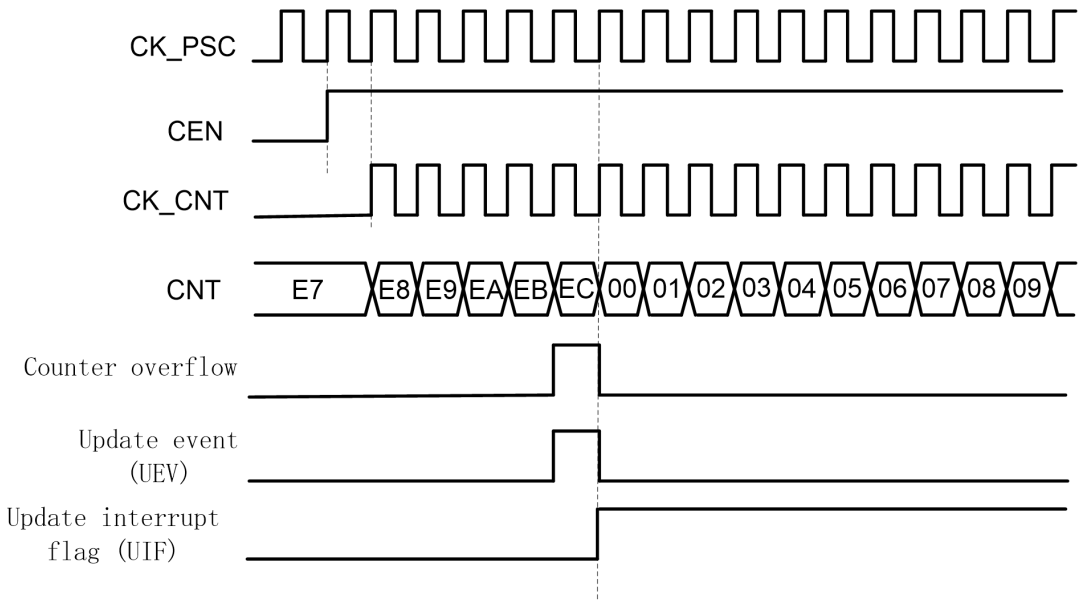


图 33-4 向上计数波形，内部时钟不分频

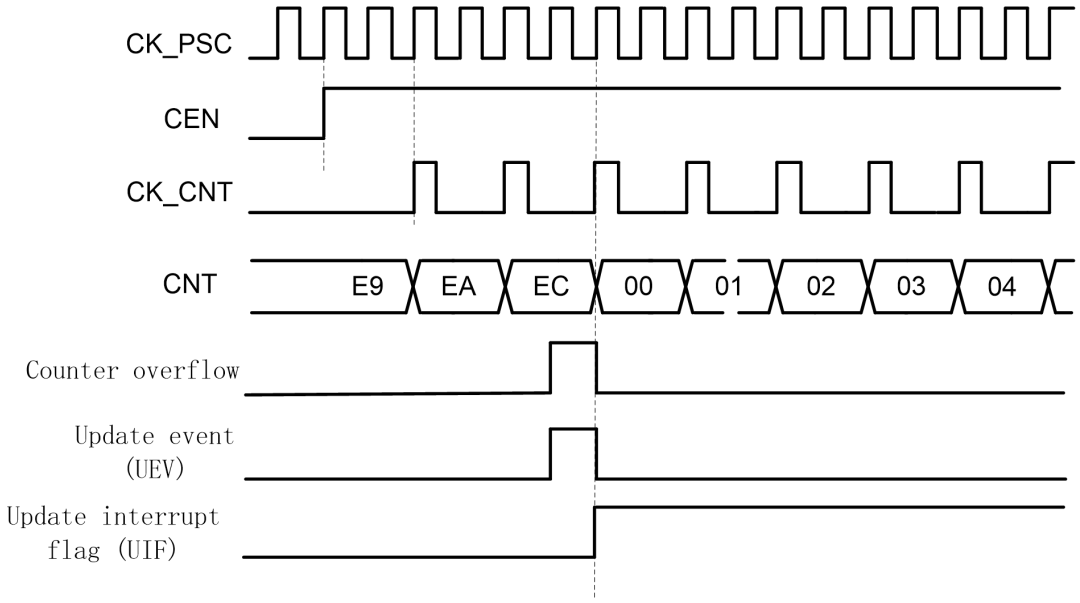


图 33-5 向上计数波形，内部时钟 2 分频

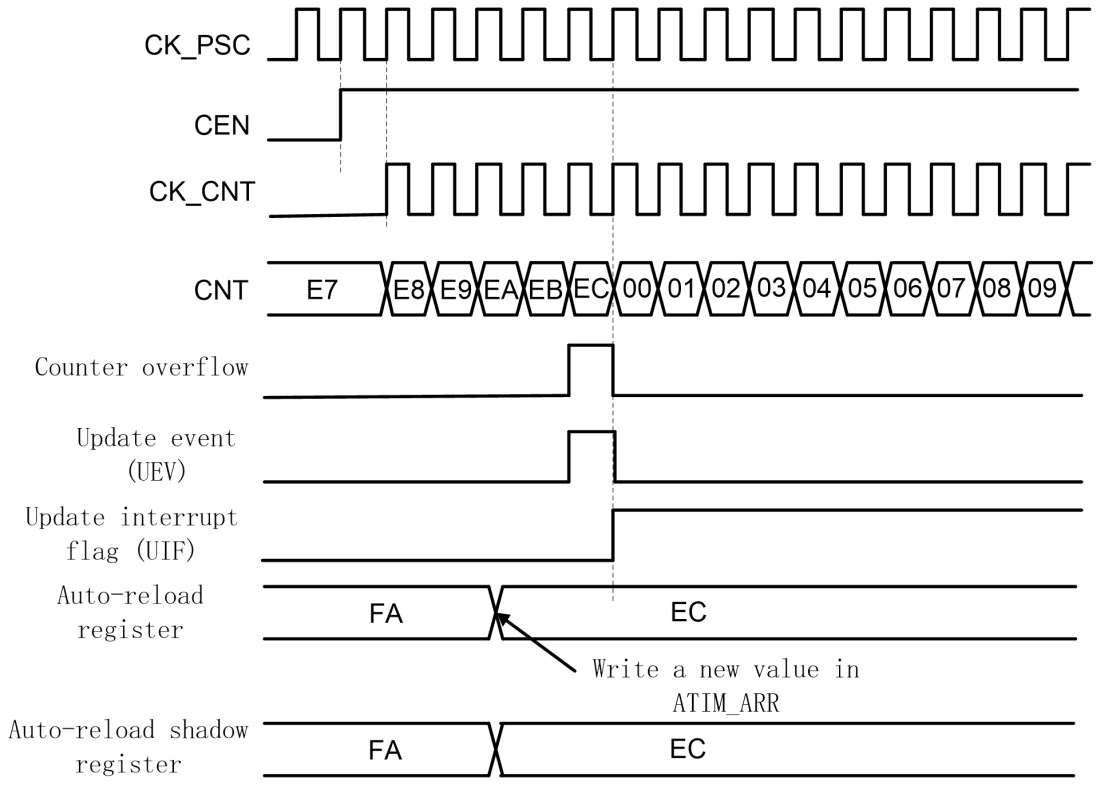


图 33-6 ARPE=0（ARR 没有预装载）时的更新事件

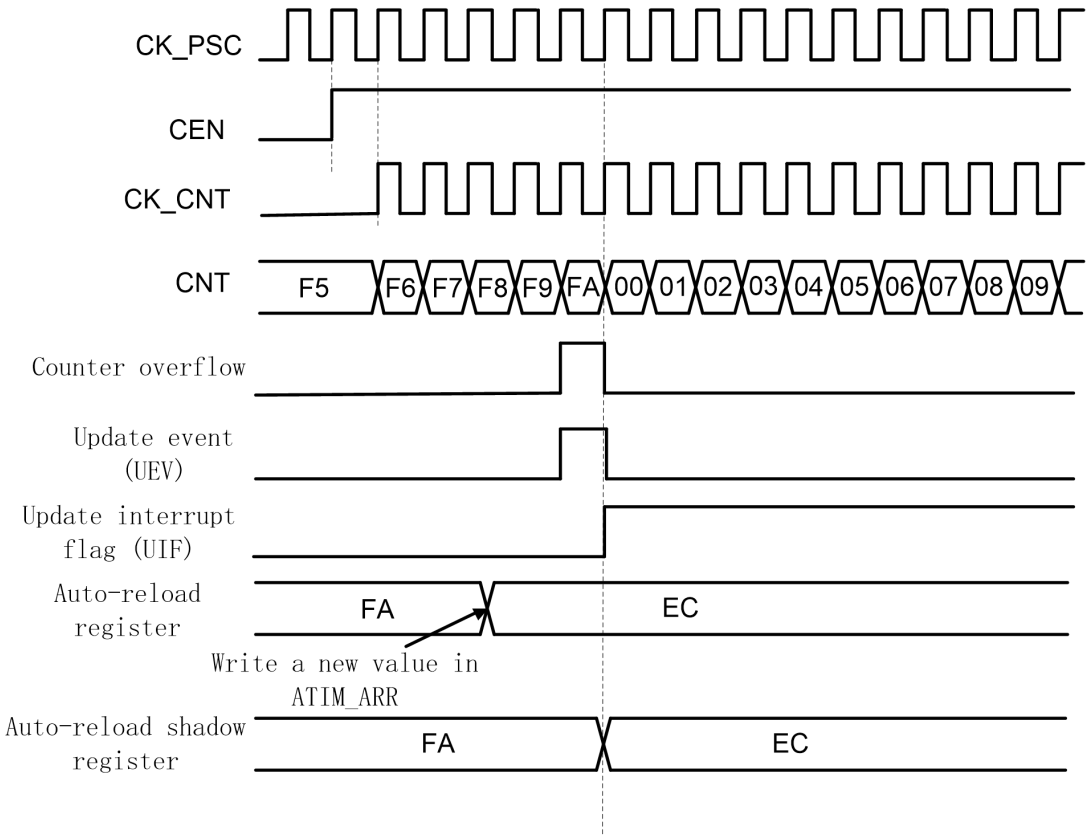


图 33-7 ARPE=1（ARR 预装载）时的更新事件

33.4.3 计数器工作时钟

BSTIM使用内部时钟工作，CEN、UG等寄存器位都是软件控制

软件操作UG寄存器后，update信号经过CLK_PSC同步后，计数器值将被重新初始化。

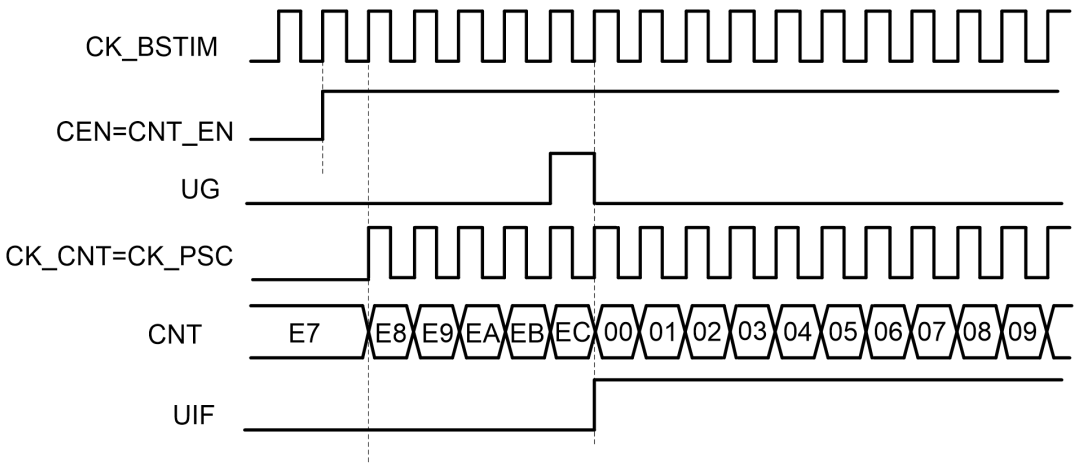


图 33-8 内部时钟源模式，时钟分频因子为 1

33.4.1 Debug 模式

当Cortex-M0进入debug模式后，定时器可以停止或继续工作，其行为由DCU模块的DBG_TIMx_STOP寄存器定义。

33.5 寄存器

BSTIM16模块基地址：0x4001_8400

offset 地址	名称	符号
0x00000000	BSTIM 控制寄存器 1 （BSTIM Control Register1）	BSTIM16_CR1
0x00000004	BSTIM 控制寄存器 2 （BSTIM Control Register2）	BSTIM16_CR2
0x0000000C	BSTIM 中断使能寄存器 （BSTIM Interrupt Enable Register）	BSTIM16_IER
0x00000010	BSTIM 中断标志寄存器 （BSTIM Interrupt Status Register）	BSTIM16_ISR
0x00000014	BSTIM 事件产生寄存器 （BSTIM Event Generation Register）	BSTIM16_EGR
0x00000024	BSTIM 计数器寄存器 （BSTIM Counter Register）	BSTIM16_CNT
0x00000028	BSTIM 预分频寄存器 （BSTIM Prescaler Register）	BSTIM16_PSC
0x0000002C	BSTIM 自动重载寄存器 （BSTIM Auto-Reload Register）	BSTIM16_ARR

33.5.1 BSTIM 控制寄存器 1

名称	BSTIM16_CR1							
地址	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	-			OPM	URS	UDIS	CEN
位权限	R/W-0	U-0			R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:8	-	RFU，未实现，读为 0
7	ARPE	Auto-reload 预装载使能 0：ARR 寄存器不使能 preload 1：ARR 寄存器使能 preload
6:4	-	RFU，未实现，读为 0
3	OPM	单脉冲输出模式 0：Update Event 发生时计数器不停止

Bit	助记符	功能描述
		1：Update Event 发生时计数器停止（自动清零 CEN）
2	URS	更新请求选择 0：以下事件能够产生 update 中断 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1：仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求
1	UDIS	禁止 update 0：使能 update 事件；以下事件发生时产生 update 事件 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1：禁止 update 事件，不更新 shadow 寄存器。当 UG 置位或从机控制器收到硬件 reset 时重新初始化计数器和预分频器。
0	CEN	计数器使能 0：计数器关闭 1：计数器使能 注意：外部触发模式可以自动置位 CEN

33.5.2 BSTIM 控制寄存器 2

名称	BSTIM16_CR2							
地址	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	MMS			-			
位权限	U-0	R/W-000			U-0			

Bit	助记符	功能描述
31:7	-	RFU，未实现，读为 0
6:4	MMS	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源 000：BSTIM_EGR 的 UG 寄存器被用作 TRGO 001：计数器使能信号 CNT_EN 被用作 TRGO，可用于同时启动多个定时器 010：UE（update event）信号被用作 TRGO Others：RFU 注意：从机定时器或 ADC 必须事先使能工作时钟，才能接收主机定时器发送的 TRGO

Bit	助记符	功能描述
3:0	-	RFU，未实现，读为 0

33.5.3 BSTIM 中断使能寄存器

名称	BSTIM16_IER							
地址	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							UIE
位权限	U-0							RW-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	UIE	Update 事件中断使能 0：禁止 Update 事件中断 1：允许 Update 事件中断

33.5.4 BSTIM 状态寄存器

名称	BSTIM16_ISR							
地址	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							UIF
位权限	U-0							RW-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	UIF	Update 事件中断标志，硬件置位，软件写 1 清零。 当以下事件发生时，UIF 置位，并更新 shadow 寄存器 -重复计数器=0，并且 UDIS=0 的情况下，计数器发生溢出

Bit	助记符	功能描述
		-URS=0 且 UDIS=0 的情况下，软件置位 UG 寄存器初始化计数器 -URS=0 且 UDIS=0 的情况下，触发事件初始化计数器

33.5.5 BSTIM 事件产生寄存器

名称	BSTIM16_EGR							
地址	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							UG
位权限	U-0							W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	**UG**	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零 软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。 *注意：由于 BSTIM 计数器工作时钟与系统总线时钟 APBCLK 是互相独立的，当软件置位 UG 寄存器后，UIF 并不会立即置位，而是需要经过 CK_BSTIM 同步后才置位，两者之间的延迟与 APBCLK 和 CK_BSTIM 之间的频率及相位关系有关，并非确定值。*

33.5.6 BSTIM 计数器寄存器

名称	BSTIM16_CNT							
地址	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	CNT[7:0]
位权限	RW-00000000

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	CNT	计数器值

33.5.7 BSTIM 预分频寄存器

名称	BSTIM16_PSC							
地址	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PSC[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PSC	计数器时钟（CK_CNT）预分频值 $f_{CK_CNT}=f_{CK_PSC}/(PSC[15:0]+1)$ 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

33.5.8 BSTIM 自动重载（auto-reload）寄存器

名称	BSTIM16_ARR							
地址	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**ARR**	计数溢出时的自动重载值 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

34 32 位低功耗定时器（LPTIM32）

34.1 概述

LPTIM32是32bits低功耗定时/计数器模块。通过选择合适的工作时钟，LPTIM32在各种低功耗模式下保持运行，并且只消耗很低的功耗。LPTIM32甚至可以在没有内部时钟的条件下工作，因此可实现休眠模式下的外部脉冲计数功能。此外，与外部输入的触发信号结合，可以实现低功耗超时唤醒功能。LPTIM32的主要特性有：

- 1 个独立的 32bit 向上计数器
- 3bit 异步时钟预分频器，8 种分频系数（1、2、4、8、16、32、64、128）
- 可选工作时钟：
  - 内部时钟源：LSCLK、RCLP、APBCLK
  - 外部时钟源：LPT32_ETR（带有模拟滤波）
- 四通道 32bit 捕捉/比较寄存器
- 32bit 自动重载寄存器
- 输入极性选择
- 无时钟外部脉冲计数
- 外部触发的休眠超时唤醒
- 32bit PWM 输出
- 32bit 输入信号捕捉
- 触发信号输出

34.2 结构框图

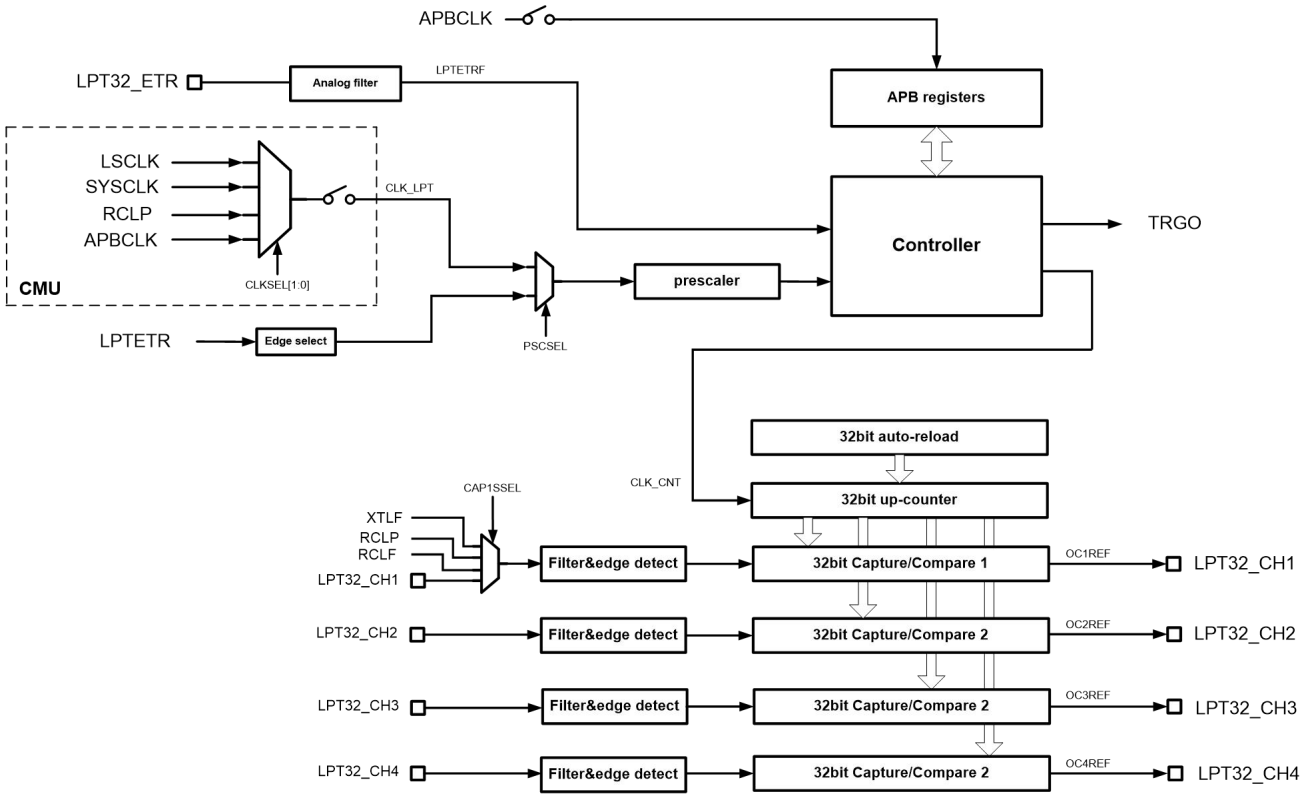


图 34-1 LPTIM32 结构框图

34.3 时钟和复位

LPTIM32的控制和状态寄存器都位于APB总线，因此软件访问寄存器前必须先使能外设总线时钟，详情参见11时钟管理单元（CMU）。

LPTIM32定时器的工作时钟独立于系统总线时钟，可以从多个独立时钟源中选择。通过配置CMU寄存器可以选择LPTIM32的计数时钟源。为了保证定时器工作稳定可靠，禁止在EN为1的情况下修改计数时钟。

在选择了合适的计数时钟之后，还可以通过DIVSEL寄存器对其进行预分频，以获得更低的工作时钟频率。同样，必须在EN为0的情况下修改DIVSEL。

LPTIM32模块可以通过操作LPT32RST寄存器来复位和撤销复位，详情参见RMU章节。

34.4 相关引脚

功能	引脚映射
	LQFP80
LPT32_ETR	PA10
LPT32_CH1	PA8
LPT32_CH2	PA9
LPT32_CH3	PC15
LPT32_CH4	PE5

表 34-1 LPTIM32 引脚映射

34.5 定时器功能

LPTIM32支持4种定时器工作模式：普通定时器、外部脉冲触发计数、外部异步脉冲计数、Timeout模式。

34.5.1 普通定时器

当LPTIM32_CFGR.TMODE=00时，LPTIM32为普通定时器工作模式

- 使用多路选择后的CLK_LPT时钟计数
- 需要配置CMU模块中的OPCCR1.LPT32CKS寄存器，选择合适的计数时钟
- LPTIM32_CR.EN使能置位后有两个计数时钟的同步过程
- 使能后定时器即开始向上计数，直到计数值等于ARR

单次计数和连续计数

LPTIM32有两种计数模式——单次计数和连续计数。

连续计数模式：计数器启动后保持运行，直到被关闭为止。计数器达到目标值（ARR）后回到0重新开始计数，并产生溢出中断OVIF。

单次计数模式：计数器被触发后计数到目标值（ARR）后回到0，并自动停止，产生溢出中断OVIF，同时硬件自动清除LPTIM32_CR.EN。

注意：由于LPTIM32使用的计数时钟异步于APBCLK，当CPU清零OVIF寄存器时，清零动作被同步到LPTIM32计数时钟上，需要2个cycle。当ARR配置为0或者1时，同步过程会导致1个OVIF事件丢失。因此不建议将ARR设置为0或者1。

34.5.2 外部脉冲触发计数

外部脉冲触发计数模式（LPTIM32_CFGR.TMODE=01）下，LPTIM32将LPT32_ETR引脚输入的信号作为触发信号使用。LPT32_ETR信号首先经过LPTIM32工作时钟采样、同步后，可以在其上升沿、下降沿或上升下降沿触发定时器递增。由于需要使用CLK_LPT采样并识别LPT32_ETR信号的变化

沿，这里要求ETR输入信号有效电平宽度必须大于CLK_LPT周期的2倍。软件可以通过LPTIM32_CFG.TRIGCFG寄存器设置LPTIM32对LPT32_ETR的哪个边沿计数。

下图举例说明了LPT32_ETR触发计数，上升沿有效的情况。

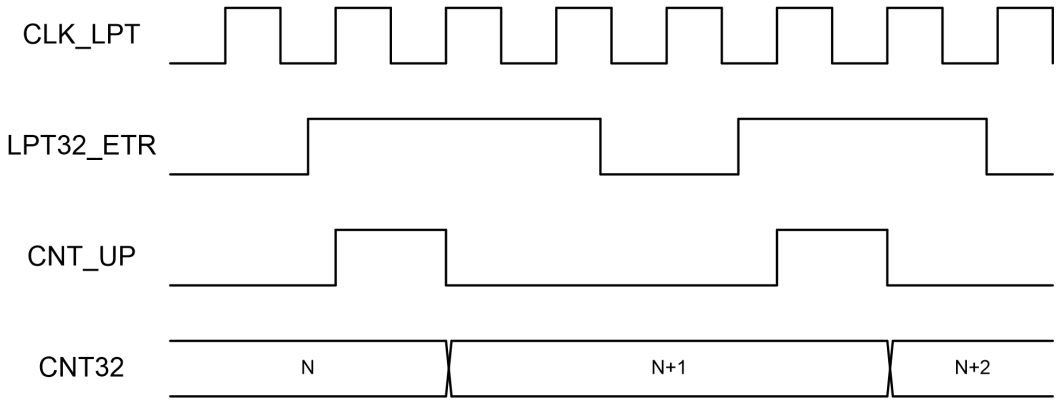


图 34-2 外部 ETR 脉冲上升沿触发计数

34.5.3 外部异步脉冲计数

外部异步脉冲计数模式（LPTIM32_CFG.TMODE=10）下，LPTIM32将LPT32_ETR引脚输入的信号直接作为计数时钟使用。这种情况下，LPTIM32全异步工作，不需要使能任何内部时钟。软件可以通过LPTIM32_CFG.EDGESEL来选择定时器使用ETR上升沿还是下降沿计数。由于这种模式下LPT32_ETR引脚上的任何干扰信号都有可能引起定时器误动作，因此推荐使能ETR输入模拟滤波功能，能够滤除大约100ns以内的glitch信号。

下图举例说明了外部异步脉冲计数，下降沿有效的情况。

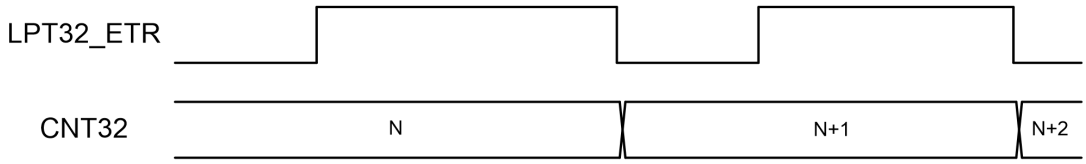


图 34-3 外部 ETR 脉冲异步计数（下降沿）

34.5.4 Timeout 模式

Timeout模式（LPTIM32_CFG.TMODE=11）下，LPTIM32将LPT32_ETR引脚输入的信号作为触发信号使用，定时器使用内部时钟CLK_LPT工作。Timeout模式下定时器启动后，不会立即开始计数，而是等待第一个LPT32_ETR信号的有效沿到来。当第一个有效沿到来后，触发定时器开始自由计数，此后每个新的ETR有效沿都会清零计数器，并重新开始计数。根据外部输入ETR信号的实际频率，合理配置计数器工作时钟和溢出上限（ARR），可以保持定时器不会溢出。如果定时器出现溢出，则表示在规定时间间隔内没有预期的ETR事件到来，则定时器产生溢出中断，计数值回到0，并自动清除LPTIM_CR.EN结束计数过程。

LPT32_ETR信号首先经过LPTIM32工作时钟采样、同步后，可以在其上升沿、下降沿或上升下降沿触发计数器清零重新计数。由于需要使用CLK_LPT采样并识别LPT32_ETR信号的变化沿，这里要求ETR输入信号有效电平宽度必须大于CLK_LPT周期的2倍。软件可以通过LPTIM32_CFG.TRIGCFG寄存器设置LPTIM32对LPT32_ETR的哪个边沿计数。

下图是timeout模式下使用LPT32_ETR上升沿清零，并最终溢出的例子。

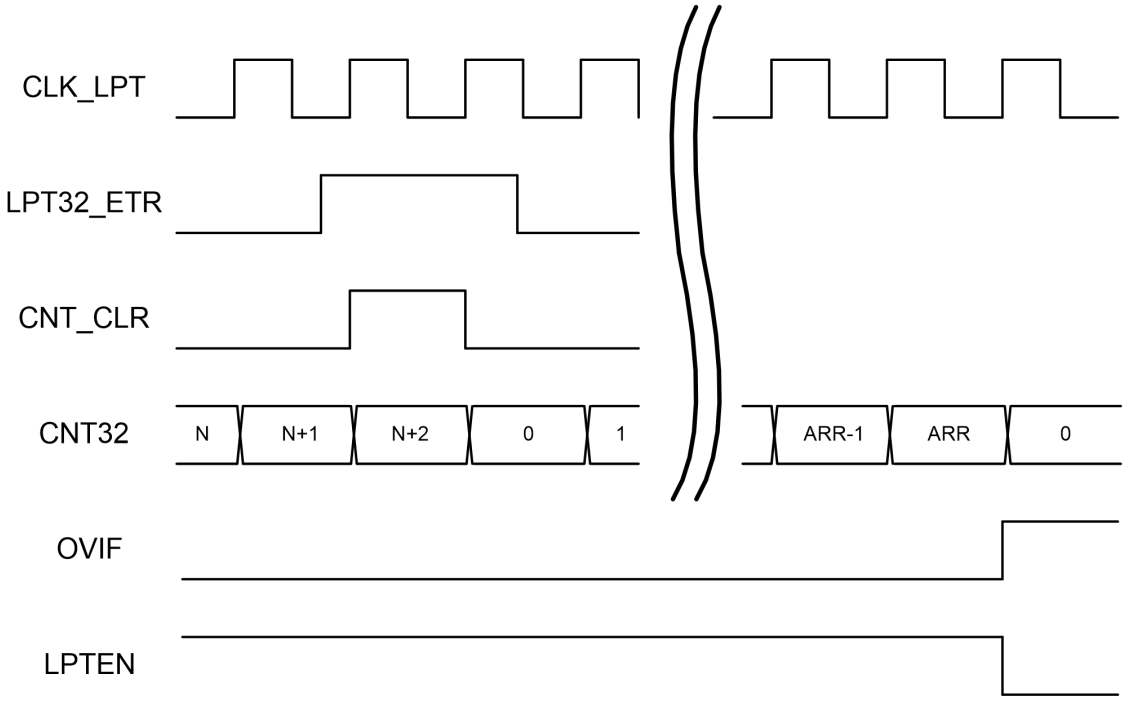


图 34-4TimeOut 模式

使用TimeOut模式，并使能LPTIM32中断，在芯片休眠时可以实现外部信号触发的超时唤醒功能。此时只要LPT32_ETR管脚上有周期性信号输入，就能使芯片保持休眠，而一旦超过规定时间内没有新的触发信号到来，LPTIM32超时溢出中断将唤醒芯片。

34.6 捕捉比较功能

LPTIM32带有四个独立的32bit捕捉比较通道，以32bit定时器为时基，结合CCRx寄存器，可以实现四路32bit PWM输出，或32bit输入捕捉功能。

34.6.1 32bit PWM

LPTIM32的两个独立捕捉/比较通道都可以输出32bit PWM波形。PWM功能需要将捕捉/比较通道配置为比较输出。

使能PWM功能后LPTIM32从0x0000_0000开始计数，以正极性波形为例，计数值等于比较值（CCRx）

时输出置高，计数值等于目标值寄存器（ARR）时输出变低；PWM周期由ARR寄存器决定，占空比由CCRx寄存器决定。LPTIM32_CCSR.POLAR寄存器可以配置输出波形的极性。

实现PWM输出功能，需要将LPTIM32_CCSR.CCxS配置为10，此时LPT_CHx成为输出通道，相应的GPIO自动使能输出功能（软件需将GPIO配置为数字外设功能）。

下图是PWM输出，POLAR=1的例子。

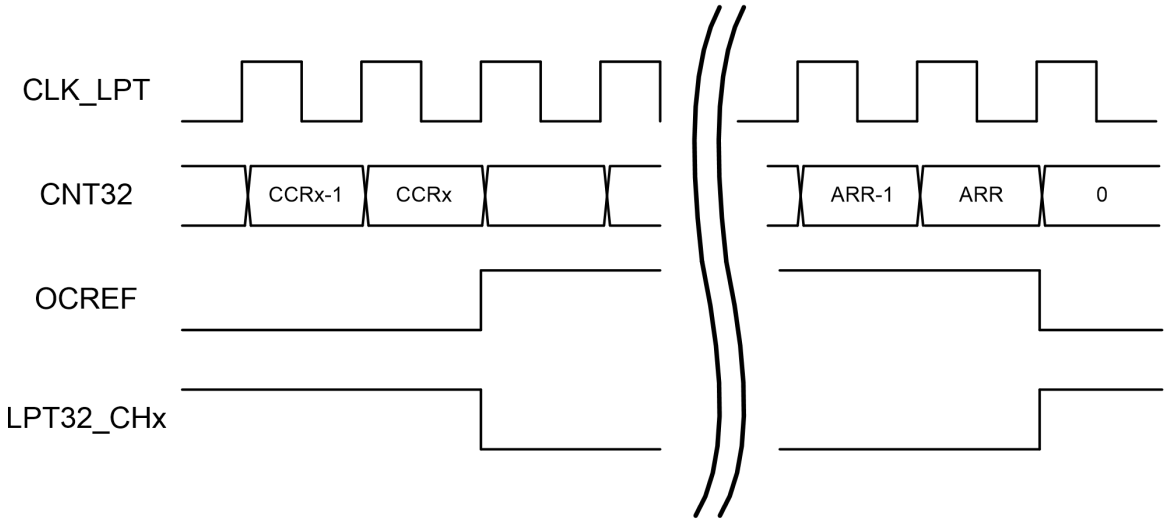


图 34-5PWM 输出

34.6.2 输入捕捉

LPTIM32的四个捕捉/比较通道可以实现两路独立的输入信号周期或电平宽度捕捉功能。

输入捕捉可以配置为针对输入信号的上升沿、下降沿或上升下降沿进行捕捉。每次捕捉发生时，CAPxEDGE寄存器会指示当前捕捉到的是上升沿还是下降沿。

LPTIM32的通道1可以对外部引脚输入或者芯片内部时钟信号（XTLF、RCLP、RCLF）进行捕捉，对内部时钟信号的周期捕捉可以用于软件配合的时钟频率校准；而通道2、3、4只能对外部引脚输入信号进行捕捉。

使能输入模式后，32bit计数器作为时基自由计数，当被捕捉信号的有效边沿到来后，当前计数值被锁存入CCRx寄存器，并产生捕捉中断；软件读取CCRx寄存器时，硬件都会自动清除捕捉中断，此外捕捉中断也可以由软件写1清零。当捕捉中断未被清除时，又有新的捕捉事件到来，会置位捕捉冲突中断标志（CAPxOVR）。

下图是对输入信号上升下降沿进行捕捉的例子。

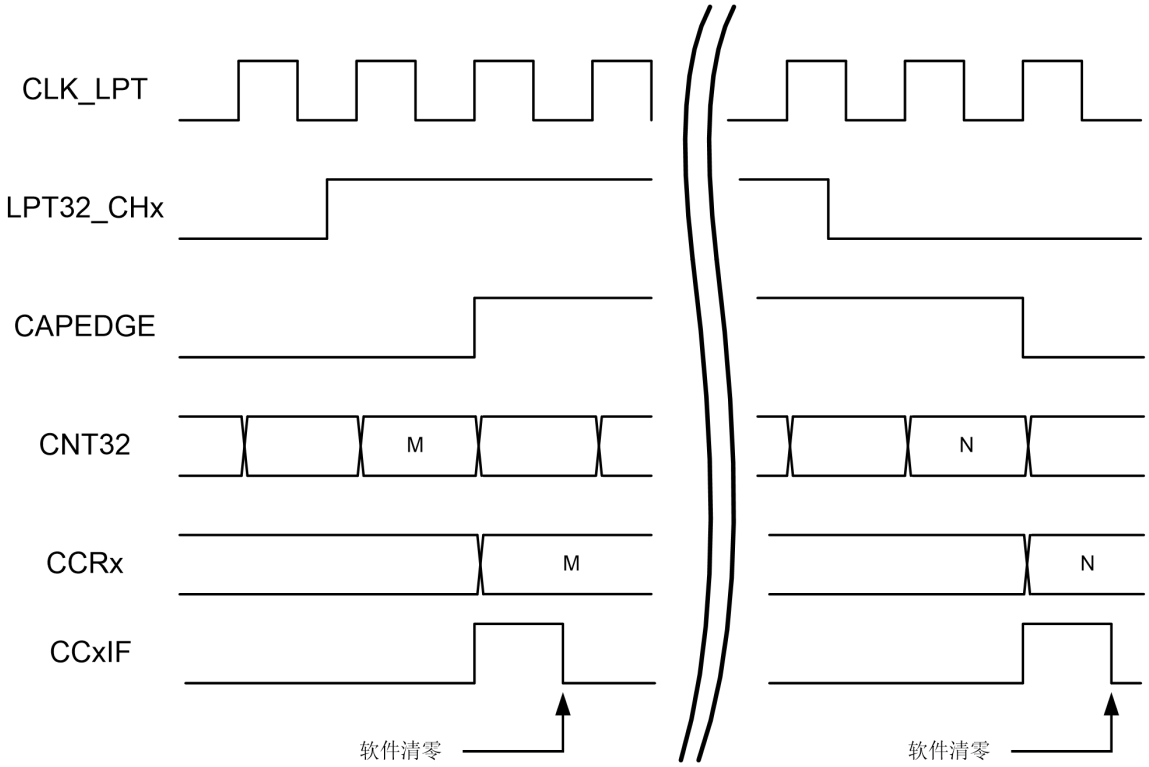


图 34-6 输入信号边沿捕捉

34.7 触发信号输出

LPTIM32可以在特定条件下向其他外设输出触发信号。触发信号源包括：

- LPTIM32使能
- 更新事件update event：计数器溢出、计数值等于CCR1或CCR2或CCR3或CCR4
- CC1通道比较脉冲：计数值等于CCR1
- CC1通道捕捉事件
- CC2通道捕捉事件
- CC3通道捕捉事件
- CC4通道捕捉事件

触发信号同步于APBCLK，因此在使用这个功能前必须打开LPTIM32的总线时钟，即置位LPT32_PCE寄存器。

34.8 寄存器

LPTIM32模块基地址：0x4001_3400

offset 地址	名称	符号
LPTIM32		
0x00	LPTIM32 配置寄存器 (LPTIM32 Config Register)	LPTIM32_CFGR
0x04	LPTIM32 计数寄存器 （LPTIM32 Counter Register）	LPTIM32_CNT
0x08	LPTIM32 捕捉比较控制和状态寄存器 （LPTIM32 Capture/Compare Control and Status Register）	LPTIM32_CCSR
0x0C	LPTIM32 目标值寄存器 （LPTIM32 Auto-Reload Register）	LPTIM32_ARR
0x10	LPTIM32 中断使能寄存器 （LPTIM32 Interrupt Enable Register）	LPTIM32_IER
0x14	LPTIM32 中断标志寄存器 （LPTIM32 Interrupt Status Register）	LPTIM32_ISR
0x18	LPTIM32 控制寄存器 （LPTIM32 Control Register）	LPTIM32_CR
0x20	LPTIM32 捕捉比较寄存器 1 （LPTIM32 Capture/Compare Register1）	LPTIM32_CCR1
0x24	LPTIM32 捕捉比较寄存器 2 （LPTIM32 Capture/Compare Register2）	LPTIM32_CCR2
0x28	LPTIM32 捕捉比较寄存器 3 （LPTIM32 Capture/Compare Register3）	LPTIM32_CCR3
0x2C	LPTIM32 捕捉比较寄存器 4 （LPTIM32 Capture/Compare Register4）	LPTIM32_CCR4

34.8.1 LPTIM32 配置寄存器（LPTIM32_CFGR）

名称	LPTIM32_CFGR							
Offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							ETR_AFEN
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					MMS		
位权限	U-0					R/W-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	PSCSEL	-	DIVSEL			-	
位权限	U-0	R/W-0	U-0	R/W-000			U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EDGESEL	TRIGCFG		-		ONST	TMOD	
位权限	R/W-0	R/W-00		U-0		R/W-0	R/W-00	

位号	位名	说明

位号	位名	说明
31:25	--	**未实现：读为0**
24	**ETR_AFEN**	LPT32_ETR 输入模拟滤波使能(External Trigger input Analog Filter Enable) 0：关闭模拟滤波 1：使能模拟滤波，滤波宽度约 100ns
23:19	--	未实现：读为0
18:16	**MMS**	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源 000：RFU 001：计数器使能信号 EN 被用作 TRGO 010：UE（update event）信号被用作 TRGO 011：CC1 比较脉冲，如果 CC1IF 标志将要置位，TRGO 输出一个正脉冲 100：CC1 捕捉事件用作 TRGO 101：CC2 捕捉事件用作 TRGO 110：CC3 捕捉事件用作 TRGO 111：CC4 捕捉事件用作 TRGO 注意：从机必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
15	--	未实现：读为0
14	**PSCSEL**	时钟预分频输入选择(Prescaler input Select) 0：CLKSEL 选择的时钟 1：LPTETR
13	--	未实现：读为0
12:10	**DIVSEL**	计数时钟分频选择(Counter Clock Divider Select) 000：1 分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110：64 分频 111：128 分频
9:8	--	未实现：读为0
7	**EDGESEL**	ETR 输入边沿选择(ETR Clock Edge Select) 0：LPT_ETR 的上升沿计数 1：LPT_ETR 的下降沿计数
6:5	**TRIGCFG**	外部触发边沿选择（需使用内部时钟同步采样 LPT_ETR）(ETR trigger Configuration) 00: LPT_ETR 输入信号上升沿触发 01: LPT_ETR 输入信号下降沿触发 10/11: LPT_ETR 输入信号上升下降沿触发
4:3	--	未实现：读为0
2	**ONST**	单次计数模式使能(One State Timer) 0：连续计数模式：计数器被触发后保持运行，直到被关闭为止。计数器达到目标值后回到 0 重新开始计数，并产生溢出中断。 1：单次计数模式：计数器被触发后计数到目标值后回到 0，并自动停止，产生溢出中断。

位号	位名	说明
1:0	**TMODE**	工作模式选择(Timer operation Mode) 00：普通定时器模式 01：Trigger 脉冲触发计数模式 10：外部异步脉冲计数模式 11：Timeout 模式

34.8.2 LPTIM32 计数值寄存器（LPTIM32_CNT）

名称	LPTIM32_CNT							
Offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CNT32[31:24]							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CNT32[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT32[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT32[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:0	**CNT32**	32bit 计数器当前计数值(Counter 32bits-wide)

34.8.3 LPTIM32 捕捉比较控制和状态寄存器（LPTIM32_CCSR）

名称	LPTIM32_CCSR							
Offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						CAP1SSEL	
位权限	U-0						R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CAP4EDGE	CAP3EDGE	CAP2EDGE	CAP1EDGE	POLAR4	POLAR3	POLAR2	POLAR1
位权限	R-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CAPCFG4		CAPCFG3		CAPCFG2		CAPCFG1	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CC4S		CC3S		CC2S		CC1S	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	

位号	位名	说明
31:26	--	未实现：读为0

位号	位名	说明
25:24	**CAP1SSEL**	通道 1 捕捉源选择(Capture channel 1 source select)，仅在 CH1 通道配置为输入捕捉时有效 00：LPT32_CH1 输入 01：XTLF 10：RCLP 11：RCLF
23	**CAP4EDGE**	通道 4 当前被捕捉的边沿，在 CC4IF 置位时更新(Channel 4 Captured Edge) 0：下降沿 1：上升沿
22	**CAP3EDGE**	通道 3 当前被捕捉的边沿，在 CC3IF 置位时更新(Channel 3 Captured Edge) 0：下降沿 1：上升沿
21	**CAP2EDGE**	通道 2 当前被捕捉的边沿，在 CC2IF 置位时更新(Channel2 Captured Edge) 0：下降沿 1：上升沿
20	**CAP1EDGE**	通道 1 当前被捕捉的边沿，在 CC1IF 置位时更新(Channel 1 Captured Edge) 0：下降沿 1：上升沿
19	**POLAR4**	通道4比较输出波形极性选择 (Channel 4 compare output Polarity) 0：正极性波形，起始为低，计数值==比较值时置高，计数值==ARR 时恢复为低 1：负极性波形，正极性波形取反
18	**POLAR3**	通道3比较输出波形极性选择 (Channel 3 compare output Polarity) 0：正极性波形，起始为低，计数值==比较值时置高，计数值==ARR 时恢复为低 1：负极性波形，正极性波形取反
17	**POLAR2**	通道2比较输出波形极性选择 (Channel 2 compare output Polarity) 0：正极性波形，起始为低，计数值==比较值时置高，计数值==ARR 时恢复为低 1：负极性波形，正极性波形取反
16	**POLAR1**	通道1比较输出波形极性选择 (Channel 1 compare output Polarity) 0：正极性波形，起始为低，计数值==比较值时置高，计数值==ARR 时恢复为低 1：负极性波形，正极性波形取反
15:14	**CAPCFG4**	通道 4 捕捉边沿选择(Channel 4 Capture edge Config) 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：RFU
13:12	**CAPCFG3**	通道 3 捕捉边沿选择(Channel 3 Capture edge Config) 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：RFU

位号	位名	说明
11:10	**CAPCFG2**	通道 2 捕捉边沿选择(Channel 2 Capture edge Config) 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：RFU
9:8	**CAPCFG1**	通道 1 捕捉边沿选择(Channel 1 Capture edge Config) 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：RFU
7:6	**CC4S**	通道 4 捕捉/比较功能使能(Channel 4 Capture/Compare Select) 00,11：禁止通道 4 捕捉/比较功能 01：使能通道 4 捕捉功能（LPT32_CH4 为输入） 10：使能通道4比较功能（LPT32_CH4为输出）
5:4	**CC3S**	通道 3 捕捉/比较功能使能(Channel 3 Capture/Compare Select) 00,11：禁止通道 3 捕捉/比较功能 01：使能通道 3 捕捉功能（LPT32_CH3 为输入） 10：使能通道 3 比较功能（LPT32_CH3 为输出）
3:2	**CC2S**	通道 2 捕捉/比较功能使能(Channel 2 Capture/Compare Select) 00,11：禁止通道 2 捕捉/比较功能 01：使能通道 2 捕捉功能（LPT32_CH2 为输入） 10：使能通道2比较功能（LPT32_CH2为输出）
1:0	**CC1S**	通道 1 捕捉/比较功能使能(Channel 1 Capture/Compare Select) 00,11：禁止通道 1 捕捉/比较功能 01：使能通道 1 捕捉功能（LPT32_CH1 为输入） 10：使能通道 1 比较功能（LPT32_CH1 为输出）

34.8.4 LPTIM32 目标值寄存器（LPTIM32_ARR）

名称	LPTIM32_ARR							
Offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ARR[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ARR[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	**ARR**	自动重载目标寄存器(Auto-Reload Register) 当计数器计数值等于 ARR 时，计数器回到初值重新开始向上计数

34.8.5 LPTIM32 中断使能寄存器（LPTIM32_IER）

名称	LPTIM32_IER							
Offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				OVR4IE	OVR3IE	OVR2IE	OVR1IE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGIE	OVIE	-		CC4IE	CC3IE	CC2IE	CC1IE
位权限	R/W-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:12	--	**未实现：读为0**
11	**OVR4IE**	通道 4 捕捉溢出中断使能(Channel 4 Over-Capture Interrupt Enable) 1：允许中断 0：禁止中断
10	**OVR3IE**	通道 3 捕捉溢出中断使能(Channel 3 Over-Capture Interrupt Enable) 1：允许中断 0：禁止中断
9	**OVR2IE**	通道 2 捕捉溢出中断使能(Channel 2 Over-Capture Interrupt Enable) 1：允许中断 0：禁止中断
8	**OVR1IE**	通道 1 捕捉溢出中断使能(Channel 1 Over-Capture Interrupt Enable) 1：允许中断 0：禁止中断
7	**TRIGIE**	外部触发到来中断使能位(External Trigger Interrupt Enable) 1：外部触发到来中断使能 0：外部触发到来中断禁止
6	**OVIE**	计数器溢出中断使能位(Counter Over-Flow Interrupt Enable) 1：计数器溢出中断使能 0：计数器溢出中断禁止
5:4	--	**未实现：读为0**
3	**CC4IE**	捕捉/比较通道 4 中断使能位(Capture/Compare channel 4 Interrupt Enable) 1：捕捉/比较通道 4 中断使能 0：捕捉/比较通道 4 中断禁止

位号	位名	说明
2	**CC3IE**	捕捉/比较通道 3 中断使能位(Capture/Compare channel 3 Interrupt Enable) 1：捕捉/比较通道 3 中断使能 0：捕捉/比较通道 3 中断禁止
1	**CC2IE**	捕捉/比较通道 2 中断使能位(Capture/Compare channel 2 Interrupt Enable) 1：捕捉/比较通道 2 中断使能 0：捕捉/比较通道 2 中断禁止
0	**CC1IE**	捕捉/比较通道 1 中断使能位(Capture/Compare channel 1 Interrupt Enable) 1：捕捉/比较通道 1 中断使能 0：捕捉/比较通道 1 中断禁止

34.8.6 LPTIM32 中断标志寄存器（LPTIM32_ISR）

名称	LPTIM32_ISR							
Offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				CAP4OVR	CAP3OVR	CAP2OVR	CAP1OVR
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGIF	OVIF	-		CC4IF	CC3IF	CC2IF	CC1IF
位权限	R/W-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:12	**--**	**未实现：读为0**
11	**CAP4OVR**	通道 4 捕捉溢出，硬件置位，软件写 1 清零(Channel 4 Over-Capture Interrupt Flag) 1：输入捕捉模式下，CC4IF 为 1 时出现新的捕捉，发生 overrun 0：没有发生 overrun
10	**CAP3OVR**	通道 3 捕捉溢出，硬件置位，软件写 1 清零(Channel 3 Over-Capture Interrupt Flag) 1：输入捕捉模式下，CC3IF 为 1 时出现新的捕捉，发生 overrun 0：没有发生 overrun
9	**CAP2OVR**	通道 2 捕捉溢出，硬件置位，软件写 1 清零(Channel 2 Over-Capture Interrupt Flag) 1：输入捕捉模式下，CC2IF 为 1 时出现新的捕捉，发生 overrun 0：没有发生 overrun
8	**CAP1OVR**	通道 1 捕捉溢出，硬件置位，软件写 1 清零(Channel 1 Over-Capture Interrupt Flag) 1：输入捕捉模式下，CC1IF 为 1 时出现新的捕捉，发生 overrun 0：没有发生 overrun

位号	位名	说明
7	**TRIGIF**	外部触发到来中断标志位，写 1 清零(External Trigger Interrupt Flag) 1：外部触发到来中断产生 0：无中断产生
6	**OVIF**	计数器溢出中断使能位，写 1 清零(Counter Over-Flow Interrupt Flag) 1：计数器溢出中断产生 0：无中断产生
5:4	**--**	**未实现：读为0**
3	**CC4IF**	捕捉/比较通道 4 中断标志，硬件置位，软件写 1 清零(Capture/Compare channel 4 Interrupt Flag) 1：计数器值和比较值 4 匹配，或者发生捕捉事件 0：无中断产生
2	**CC3IF**	捕捉/比较通道 3 中断标志，硬件置位，软件写 1 清零(Capture/Compare channel 3 Interrupt Flag) 1：计数器值和比较值 3 匹配，或者发生捕捉事件 0：无中断产生
1	**CC2IF**	捕捉/比较通道 2 中断标志，硬件置位，软件写 1 清零(Capture/Compare channel 2 Interrupt Flag) 1：计数器值和比较值 2 匹配，或者发生捕捉事件 0：无中断产生 *注：捕捉模式下，此标志写 1 清零或者读 CCR2 清零。*
0	**CC1IF**	捕捉/比较通道 1 中断标志，硬件置位，软件写 1 清零(Capture/Compare channel 1 Interrupt Flag) 1：计数器值和比较值 1 匹配，或者发生捕捉事件 0：无中断产生 *注：捕捉模式下，此标志写 1 清零或者读 CCR1 清零。*

34.8.7 LPTIM32 控制寄存器（LPTIM32_CR）

名称	LPTIM32_CR							
Offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

位号	位名	说明
31:1	**--**	**未实现：读为0**

位号	位名	说明
0	**EN**	LPTIM 使能位(LPTIM Enable) 1：使能计数器计数 0：禁止计数器计数

34.8.8 LPTIM32 捕捉比较寄存器 1（LPTIM32_CCR1）

名称	LPTIM32_CCR1							
Offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CCR1[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CCR1[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	**CCR1**	捕捉/比较值寄存器 1 (Channel1 Capture/Compare Register)

34.8.9 LPTIM32 捕捉比较寄存器 2（LPTIM32_CCR2）

名称	LPTIM32_CCR2							
offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CCR2[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CCR2[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	**CCR2**	捕捉/比较值寄存器 2 (Channel2 Capture/Compare Register)

34.8.10 LPTIM32 捕捉比较寄存器 3（LPTIM32_CCR3）

名称	LPTIM32_CCR3							
offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CCR3[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CCR3[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR3[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR3[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	**CCR3**	捕捉/比较值寄存器 3 (Channel3 Capture/Compare Register)

34.8.11 LPTIM32 捕捉比较寄存器 4（LPTIM32_CCR4）

名称	LPTIM32_CCR4							
offset	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CCR4[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CCR4[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR4[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR4[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	**CCR4**	捕捉/比较值寄存器 4 (Channel4 Capture/Compare Register)

35 16 位低功耗定时器（LPTIM16）

35.1 概述

LPTIM16是16bits低功耗定时/计数器模块。通过选择合适的工作时钟，LPTIM16在在各种低功耗模式下保持运行，并且只消耗很低的功耗。LPTIM16甚至可以在没有内部时钟的条件下工作，因此可实现休眠模式下的外部脉冲计数功能。此外，与外部输入的触发信号结合，可以实现低功耗超时唤醒功能。LPTIM16的主要特性有：

- 1 个独立的 16bit 向上计数器
- 3bit 异步时钟预分频器，8 种分频系数（1、2、4、8、16、32、64、128）
- 可选工作时钟：
  - 内部时钟源：LSCLK、RCLP、APBCLK、SYSCLK
  - 外部时钟源：LPT16_ETR（带有模拟滤波）
- 2 通道 16bit 捕捉/比较寄存器
- 16bit 自动重载寄存器
- 输入极性选择
- 无时钟外部脉冲计数
- 外部触发的休眠超时唤醒
- 16bit PWM 输出
- 16bit 输入信号捕捉
- 触发信号输出
- 双通道正交编码器

35.2 结构框图

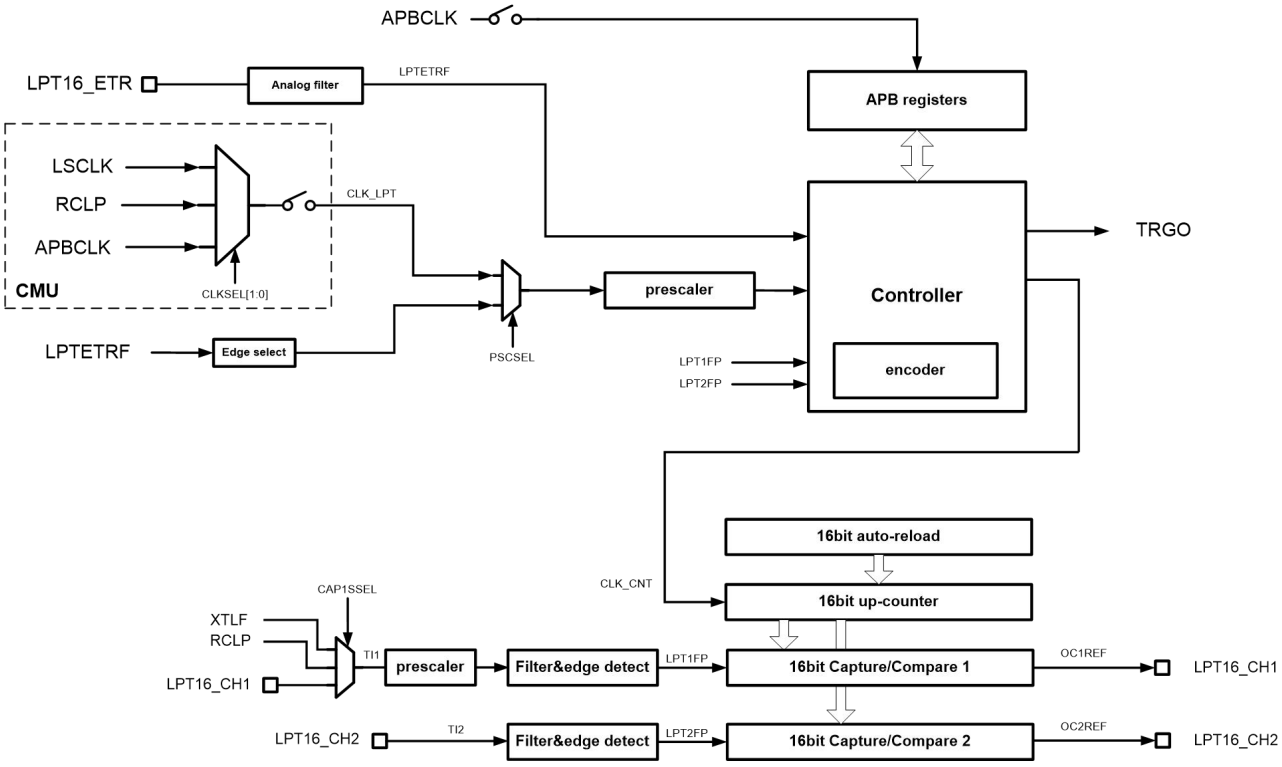


图 35-1 LPTIM16 结构框图

35.3 时钟和复位

LPTIM16的控制和状态寄存器都位于APB总线，因此软件访问寄存器前必须先使能外设总线时钟，详情参见11时钟管理单元（CMU）。

LPTIM16定时器的工作时钟独立于系统总线时钟，可以从多个独立时钟源中选择。通过配置CMU寄存器可以选择LPTIM16的计数时钟源。为了保证定时器工作稳定可靠，禁止在EN为1的情况下修改计数时钟。

在选择了合适的计数时钟之后，还可以通过DIVSEL寄存器对其进行预分频，以获得更低的工作时钟频率。同样，必须在EN为0的情况下修改DIVSEL。

LPTIM16模块可以通过操作LPT16RST寄存器来复位和撤销复位，详情参见RMU章节。

35.4 定时器功能

LPTIM16支持4种定时器工作模式：普通定时器、外部脉冲触发计数、外部异步脉冲计数、Timeout模式。

35.4.1 普通定时器

当LPTCFG.TMODE=00时，LPTIM16为普通定时器工作模式

- 使用多路选择后的CLK_LPT时钟计数
- 需要配置CMU模块中的OPCCON2.LPTCKS寄存器，选择合适的计数时钟
- LPTEN使能置位后有两个计数时钟的同步过程
- 使能后定时器即开始向上计数，直到计数值等于LPTARR

单次计数和连续计数

LPTIM16有两种计数模式——单次计数和连续计数。

连续计数模式：计数器启动后保持运行，直到被关闭为止。计数器达到目标值（LPTARR）后回到0重新开始计数，并产生溢出中断。

单次计数模式：计数器被触发后计数到目标值（LPTARR）后回到0，并自动停止，产生溢出中断，同时硬件自动清除LPTEN。

35.4.2 外部脉冲触发计数

外部脉冲触发计数模式（LPTCFG.TMODE=01）下，LPTIM16将LPT16_ETR引脚输入的信号作为触发信号使用。LPT16_ETR信号首先经过LPTIM16工作时钟采样、同步后，可以在其上升沿、下降沿或上升下降沿触发定时器递增。由于需要使用CLK_LPT采样并识别LPT16_ETR信号的变化沿，这里要求ETR输入信号有效电平宽度必须大于CLK_LPT周期的2倍。软件可以通过LPTCFG.TRIGCFG寄存器设置LPTIM16对LPT16_ETR的哪个边沿计数。

下图举例说明了LPT16_ETR触发计数，上升沿有效的情况。

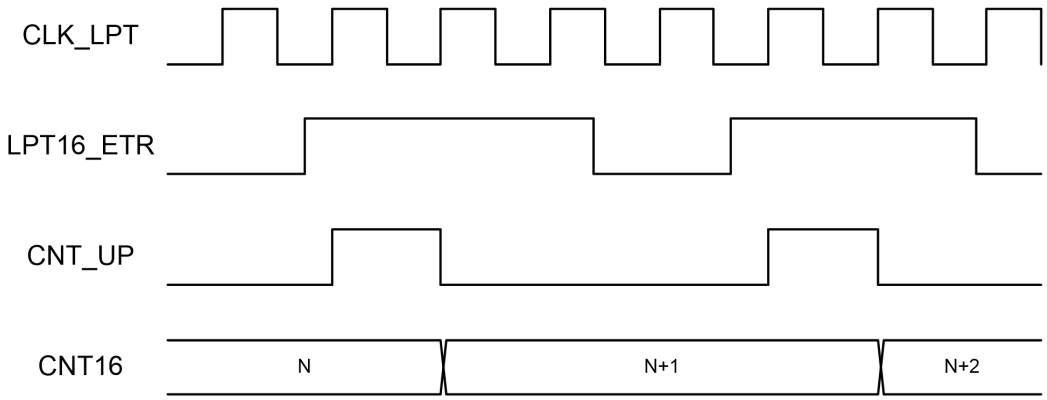


图 35-2 外部 ETR 脉冲上升沿触发计数

35.4.3 外部异步脉冲计数

外部异步脉冲计数模式（LPTCFG.TMODE=10）下，LPTIM16将LPT16_ETR引脚输入的信号直接作为计数时钟使用。这种情况下，LPTIM16全异步工作，不需要使能任何内部时钟。软件可以通过LPTCFG.EDGESEL来选择定时器使用ETR上升沿还是下降沿计数。由于这种模式下LPT16_ETR引脚上的任何干扰信号都有可能引起定时器误动作，因此推荐使能ETR输入模拟滤波功能，能够滤除大约100ns以内的glitch信号。

下图举例说明了外部异步脉冲计数，下降沿有效的情况。

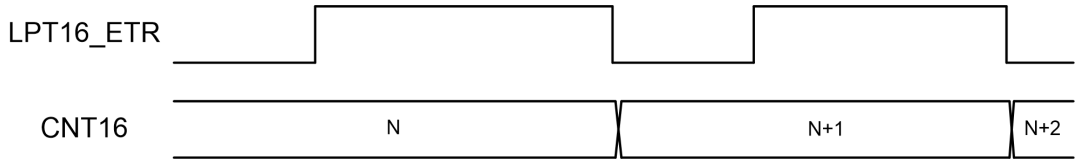


图 35-3 外部 ETR 脉冲异步计数（下降沿）

35.4.4 Timeout 模式

Timeout模式（LPTCFG.TMODE=11）下，LPTIM16将LPT16_ETR引脚输入的信号作为触发信号使用，定时器使用内部时钟CLK_LPT工作。Timeout模式下定时器启动后，不会立即开始计数，而是等待第一个LPT16_ETR信号的有效沿到来。当第一个有效沿到来后，触发定时器开始自由计数，此后每个新的ETR有效沿都会清零计数器，并重新开始计数。根据外部输入ETR信号的实际频率，合理配置计数器工作时钟和溢出上限（LPTARR），可以保持定时器不会溢出。如果定时器出现溢出，则表示在规定时间间隔内没有预期的ETR事件到来，则定时器产生溢出中断，计数值回到0，并自动清除LPTEN结束计数过程。

LPT16_ETR信号首先经过LPTIM16工作时钟采样、同步后，可以在其上升沿、下降沿或上升下降沿触发计数器清零重新计数。由于需要使用CLK_LPT采样并识别LPT16_ETR信号的变化沿，这里要求ETR输入信号有效电平宽度必须大于CLK_LPT周期的2倍。软件可以通过LPTCFG.TRIGCFG寄存器设置LPTIM16对LPT16_ETR的哪个边沿计数。

下图是timeout模式下使用LPT16_ETR上升沿清零，并最终溢出的例子。

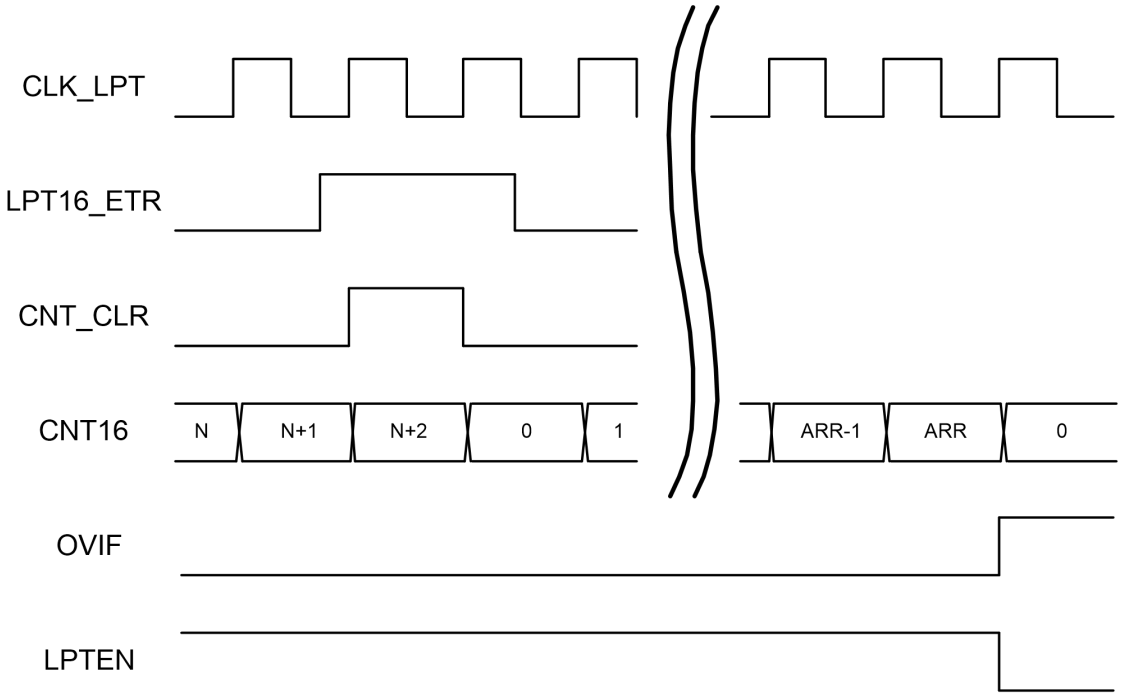


图 35-4TimeOut 模式

使用TimeOut模式，并使能LPTIM中断，在芯片休眠时可以实现外部信号触发的超时唤醒功能。此时只要LPT16_ETR管脚上有周期性信号输入，就能使芯片保持休眠，而一旦超过规定时间内没有新的触发信号到来，LPTIM超时溢出中断将唤醒芯片。

35.5 捕捉比较功能

LPTIM16带有两个独立的16bit捕捉比较通道，以16bit定时器为时基，结合CCRx寄存器，可以实现两路16bit PWM输出，或16bit输入捕捉功能。

35.5.1 16bit PWM

LPTIM16的两个独立捕捉/比较通道都可以输出16bit PWM波形。PWM功能需要将捕捉/比较通道配置为比较输出。

使能PWM功能后LPTIM16从0x0000_0000开始计数，计数值等于比较值（CCRx）时输出置高，计数值等于目标值寄存器（LPTARR）时输出变低；PWM周期由ARR寄存器决定，占空比由CCRx寄存器决定。LPTCFG.POLARITY寄存器可以配置输出波形的极性。

实现PWM输出功能，需要将LPTCFG.CCxS配置为10，此时LPT_CHx成为输出通道，相应的GPIO自动使能输出功能（软件需将GPIO配置为数字外设功能）。

下图是PWM输出，POLARITY=1的例子。

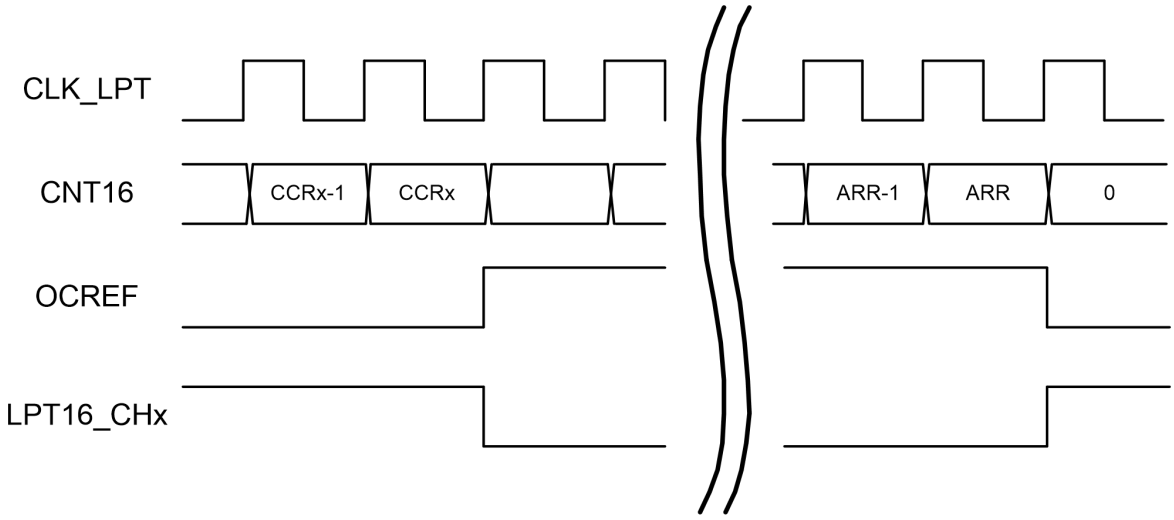


图 35-5PWM 输出

35.5.2 输入捕捉

LPTIM16的两个捕捉/比较通道可以实现两路独立的输入信号周期或电平宽度捕捉功能。输入信号捕捉功能可以配合DMA使用，实现多次连续捕捉结果的自动搬运。

输入捕捉可以配置为针对输入信号的上升沿、下降沿或上升下降沿进行捕捉。每次捕捉发生时，CAPxEDGE寄存器会指示当前捕捉到的是上升沿还是下降沿。

LPTIM16的通道1可以对外部引脚输入或者芯片内部时钟信号（XTLF、RCLP）进行捕捉，对内部时钟信号的周期捕捉可以用于软件配合的时钟频率校准；而通道2只能对外部引脚输入信号进行捕捉。通道1内置预分频器，经过对输入信号分频，可以对高频时钟进行分频后的周期捕捉和校准。

使能输入模式后，16bit计数器作为时基自由计数，当被捕捉信号的有效边沿到来后，当前计数值被锁存入CCRx寄存器，并产生捕捉中断；软件读取CCRx寄存器时，硬件都会自动清除捕捉中断，此外捕捉中断也可以由软件写1清零。当捕捉中断未被清除时，又有新的捕捉事件到来，会置位捕捉冲突中断标志（CAPxOVR）。

下图是对输入信号上升下降沿进行捕捉的例子。

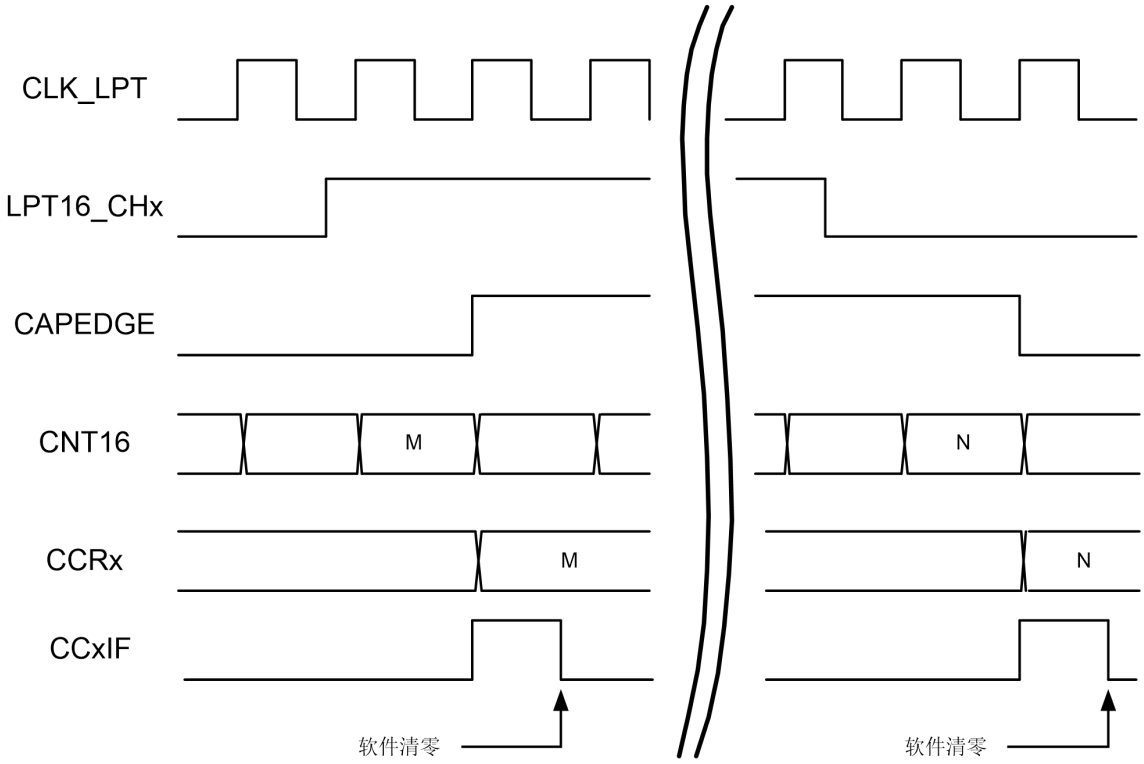


图 35-6 输入信号边沿捕捉

35.5.1 输入数字滤波

两个输入通道可以独立使能或关闭输入数字滤波功能。使能数字滤波的情况下，使用CLK_LPT计数时钟连续采样输入信号三次，相同电平认为有效。

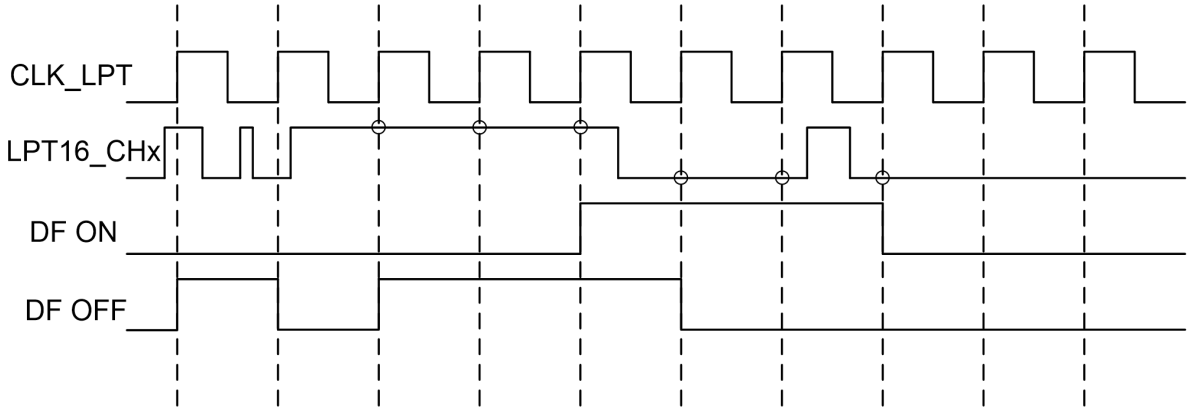


图 35-7 通道输入数字滤波

35.6 正交编码器

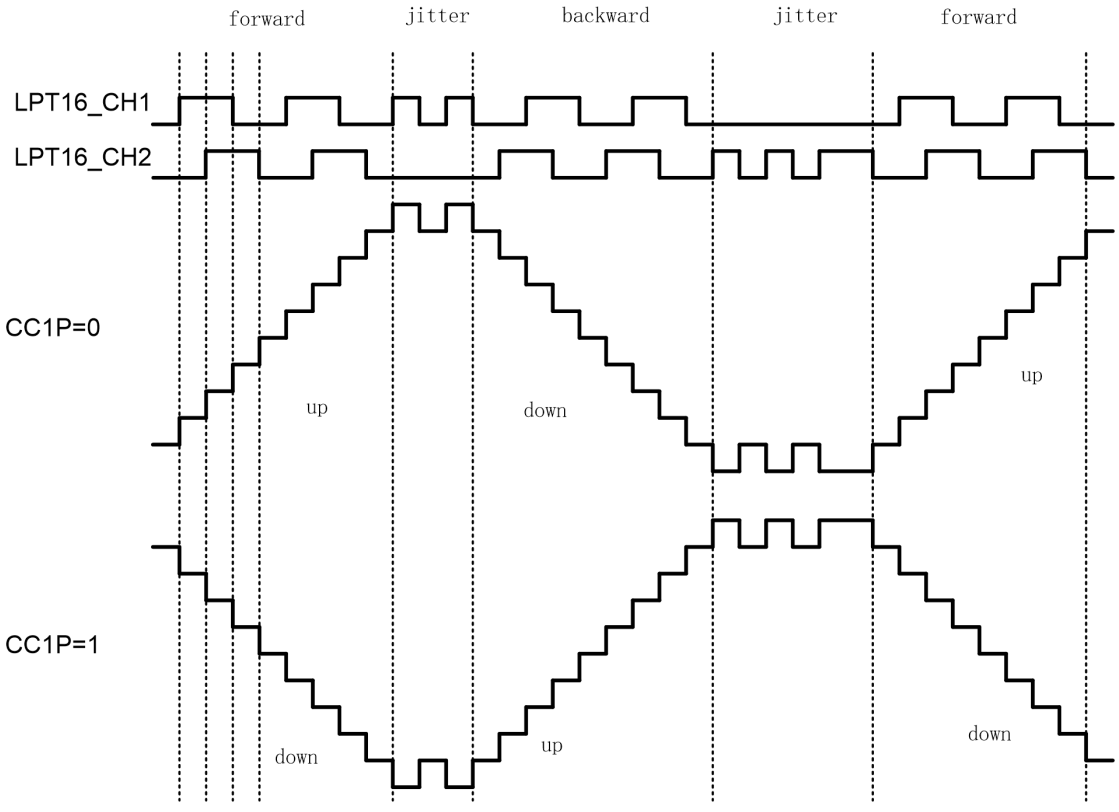
编码器接口模式涉及到两个外部输入信号，LPTIM16根据其中一个信号的边沿相对于另一个信号的电平来决定递增还是递减计数值。下表是计数方式与两路输入信号之间的关系：

有效沿	对应信号的电平（LPT1FP对应LPT2FP，LPT2FP对应LPT1FP）	LPT1FP信号		LPT2FP信号	
		上升	下降	上升	下降
仅在LPT1FP处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在LPT2FP处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在LPT1FP和LPT2FP处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

表 35-1 encoder interface 计数方式

比如在计数器以LPT1FP信号为时钟计数时，如果LPT1FP上升沿采样到LPT2FP为高电平，则计数器递减；如果LPT1FP下降沿采样到LPT2FP为高电平，则计数器递增。（同步计数，CLK_LPT采样取沿后进行计数，参考ATIM设计）

下图为编码器在LPT1FP和LPT2FP边沿同时计数的例子。由于CC1P能够控制输入信号LPT16_CH1的极性，所以当CC1P为1时输入取反，相应的计数器增减方向也会反向。



Example of counter operation in encoder interface mode

图 35-8 编码器模式下的计数器操作实例

正交编码信号使用LPT16_CHx输入，可以选择对输入信号是否进行数字滤波。注意LPT16_CH1通道上还有预分频电路，在进行正交编码计数时应关闭预分频。

QDIR寄存器用于表示当前输入信号相位关系，当CH1相位领先于CH2时，QDIR=0，反之QDIR=1，此寄存器只读。

当CH1和CH2发生相位翻转时，QDCIF中断标志置位，表示出现换向，如果QDCIE寄存器置位，则产生中断事件。

编码模式输入通道需进行如下设置：

- 在GPIO模块中，配置相应管脚为LPT16_CH1，LPT16_CH2功能
- 选择输入通道，配置LPTIM_CCSR.CC1S=01，LPTIM_CCSR.CC2S=01
- 选择计数有效沿（输入信号极性），配置LPTIM_CCSR.CC1P和CC2P
- 设定编码器模式，配置LPTIM_CFGR.QEMD寄存器
- 使能LPTIM16，置位LPTIM_CR.EN

35.7 寄存器

模块基地址：0x4001_8800

offset 地址	名称	符号
0x00	LPTIM 配置寄存器 (LPTIM Config Register)	LPTIM16_CFGR
0x04	LPTIM 计数寄存器 （LPTIM Counter Register）	LPTIM16_CNT
0x08	LPTIM 捕捉比较控制和状态寄存器 （ LPTIM Capture/Compare Control and Status Register）	LPTIM16_CCSR
0x0C	LPTIM 目标值寄存器 （LPTIM Auto-Reload Register）	LPTIM16_ARR
0x10	LPTIM 中断使能寄存器 （LPTIM Interrupt Enable Register）	LPTIM16_IER
0x14	LPTIM 中断标志寄存器 （LPTIM Interrupt Status Register）	LPTIM16_ISR
0x18	LPTIM 控制寄存器 （LPTIM Control Register）	LPTIM16_CR
0x20	LPTIM 捕捉比较寄存器 1 （LPTIM Capture/Compare Register1）	LPTIM16_CCR1
0x24	LPTIM 捕捉比较寄存器 2 （LPTIM Capture/Compare Register2）	LPTIM16_CCR2

35.7.1 LPTIM 配置寄存器（LPTIM16_CFGR）

名称	LPTIM16_CFGR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							ETR_AFEN
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					MMS		
位权限	U-0					R/W-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	PSCSEL	-	DIVSEL			-	
位权限	U-0	R/W-0	U-0	R/W-000			U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EDGESEL	TRIGCFG		QEMD		ONST	TMOD	
位权限	R/W-0	R/W-00		R/W-00		R/W-0	R/W-00	

位号	位名	说明
31:25	--	**未实现：读为0**
24	**ETR_AFEN**	LPT16_ETR 输入模拟滤波使能(External Trigger input Analog Filter Enable) 0：关闭模拟滤波 1：使能模拟滤波，滤波宽度约 100ns

位号	位名	说明
23:19	--	未实现：读为0
18:16	**MMS**	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源 000：RFU 001：计数器使能信号 EN 被用作 TRGO 010：UE（update event）信号被用作 TRGO 011：比较脉冲，如果 CC1IF 标志将要置位，TRGO 输出一个正脉冲 100：OC1REF 用作 TRGO 101：OC2REF 用作 TRGO 110：RFU 111：RFU 注意：从机必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
15	--	未实现：读为0
14	**PSCSEL**	时钟预分频输入选择(Prescaler input Select) 0：CLKSEL 选择的时钟 1：LPTETRF
13	--	未实现：读为0
12:10	**DIVSEL**	计数时钟分频选择(Counter Clock Divider Select) 000：1 分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110：64 分频 111：128 分频
9:8	--	未实现：读为0
7	**EDGESEL**	ETR 输入边沿选择(ETR Clock Edge Select) 0：LPT_ETR 的上升沿计数 1：LPT_ETR 的下降沿计数
6:5	**TRIGCFG**	外部触发边沿选择（需使用内部时钟同步采样 LPT_ETR）(ETR trigger Configuration) 00: LPT_ETR 输入信号上升沿触发 01: LPT_ETR 输入信号下降沿触发 10/11: 外部输入信号上升下降沿触发
4:3	**QEMD**	正交编码器模式 (Quad encoder mode)，仅TMODE=00时有效 00：关闭正交编码器 01：Encoder 模式 1；计数器使用 LPT2FP 边沿，根据 LPT1FP 电平高低来计数 10：Encoder 模式 2；计数器使用 LPT1FP 边沿，根据 LPT2FP 电平高低来计数 11：Encoder 模式 3；计数器同时使用 LPT1FP 和 LPT2FP 边沿，根据其他输入信号电平来计数

位号	位名	说明
2	**ONST**	单次计数模式使能(One State Timer) 0：连续计数模式：计数器被触发后保持运行，直到被关闭为止。计数器达到目标值后回到 0 重新开始计数，并产生溢出中断。 1：单次计数模式：计数器被触发后计数到目标值后回到 0，并自动停止，产生溢出中断。
1:0	**TMODE**	工作模式选择(Timer operation Mode) 00：普通定时器模式 01：Trigger 脉冲触发计数模式 10：外部异步脉冲计数模式 11：Timeout 模式

35.7.2 LPTIM 计数值寄存器（LPTIM16_CNT）

名称	LPTIM16_CNT							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:16	**--**	未实现：读为0
15:0	**CNT**	16bit 计数器当前计数值(Counter 16bits-wide)

35.7.3 LPTIM 捕捉比较控制和状态寄存器（LPTIM16_CCSR）

名称	LPTIM16_CCSR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CAP1PSC						CAP1SSEL	
位权限	R/W-00 0000						R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		CAP2EDGE	CAP1EDGE	-	-	CC2P	CC1P
位权限	U-0		R-0	R-0	U-0	U-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		CC2DF	CC1DF	CAPCFG2		CAPCFG1	
位权限	U-0		R/W-0	R/W-0	R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	CC2S	CC1S
位权限	U-0	R/W-00	R/W-00

位号	位名	说明
31:26	**CAP1PSC**	通道1输入预分频 (Capture channel 1 prescaler) 0x00：不分频 0x3F：128分频
25:24	**CAP1SSEL**	通道 1 捕捉源选择(Capture channel 1 source select)，仅在 CH1 通道配置为输入捕捉时有效 00：LPT16_CH1 输入 01：XTLF 10：RCLP 11：RFU
23	--	未实现：读为0
22	--	未实现：读为0
21	**CAP2EDGE**	通道 2 当前被捕捉的边沿，在 CC2IF 置位时更新(Channel2 Captured Edge) 0：下降沿 1：上升沿
20	**CAP1EDGE**	通道 1 当前被捕捉的边沿，在 CC1IF 置位时更新(Channel 1 Captured Edge) 0：下降沿 1：上升沿
19	--	未实现：读为0
18	--	未实现：读为0
17	**CC2P**	通道2配置为比较功能时：通道2比较输出波形极性选择 (Channel 2 compare output Polarity) 0：正极性波形，起始为低，计数值==比较值时置高，计数值==ARR时恢复为低 1：负极性波形，正极性波形取反
		通道2配置为输入捕捉功能时：通道2输入波形极性选择 (Channel 2 input polarity) 0：输入不取反 1：输入取反
16	**CC1P**	通道1比较输出波形极性选择 (Channel 1 compare output Polarity) 0：正极性波形，起始为低，计数值==比较值时置高，计数值==ARR时恢复为低 1：负极性波形，正极性波形取反
		通道1配置为输入捕捉功能时：通道1输入波形极性选择 (Channel 1 input polarity) 0：输入不取反 1：输入取反
15:14	--	未实现：读为0
13	**CC2DF**	通道2输入数字滤波 (Channel 2 input digital filter) 1：使能数字滤波 0：关闭数字滤波
12	**CC1DF**	通道1输入数字滤波 (Channel 1 input digital filter) 1：使能数字滤波 0：关闭数字滤波

位号	位名	说明
11:10	**CAPCFG2**	通道 2 捕捉边沿选择(Channel 2 Capture edge Config) 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：RFU
9:8	**CAPCFG1**	通道 1 捕捉边沿选择(Channel 1 Capture edge Config) 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：RFU
7:6	--	未实现：读为0
5:4	--	未实现：读为0
3:2	**CC2S**	通道 2 捕捉/比较功能使能(Channel 2 Capture/Compare Select) 00,11：禁止通道 2 捕捉/比较功能 01：使能通道 2 捕捉功能（LPT16_CH2 为输入） 10：使能通道2比较功能（LPT16_CH2为输出）
1:0	**CC1S**	通道 1 捕捉/比较功能使能(Channel 1 Capture/Compare Select) 00,11：禁止通道 1 捕捉/比较功能 01：使能通道 1 捕捉功能（LPT16_CH1 为输入） 10：使能通道 1 比较功能（LPT16_CH1 为输出）

35.7.4 LPTIM 目标值寄存器（LPTIM16_ARR）

名称	LPTIM16_ARR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:16	--	未实现：读为0
15:0	**ARR**	自动重载目标寄存器(Auto-Reload Register) 当计数器计数值等于 ARR 时，计数器回到初值重新开始向上计数

35.7.5 LPTIM 中断使能寄存器（LPTIM16_IER）

名称	LPTIM16_IER
offset	0x10

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						OVR2IE	OVR1IE
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGIE	OVIE	QDCIE	-			CC2IE	CC1IE
位权限	R/W-0	R/W-0	R/W-0	U-0			R/W-0	R/W-0

位号	位名	说明
31:10	--	**未实现：读为0**
9	**OVR2IE**	通道 2 捕捉溢出中断使能(Channel 2 Over-Capture Interrupt Enable) 1：允许中断 0：禁止中断
8	**OVR1IE**	通道 1 捕捉溢出中断使能(Channel 1 Over-Capture Interrupt Enable) 1：允许中断 0：禁止中断
7	**TRIGIE**	外部触发到来中断使能位(External Trigger Interrupt Enable) 1：外部触发到来中断使能 0：外部触发到来中断禁止
6	**OVIE**	计数器溢出中断使能位(Counter Over-Flow Interrupt Enable) 1：计数器溢出中断使能 0：计数器溢出中断禁止
5	**QDCIE**	Quadrature Decoder 方向改变中断使能 (Quad-decoder Interrupt Enable) 1：使能 QDCIF 中断 0：禁止 QDCIF 中断
4:2	--	**未实现：读为0**
1	**CC2IE**	捕捉/比较通道 2 中断使能位(Capture/Compare channel 2 Interrupt Enable) 1：捕捉/比较通道 2 中断使能 0：捕捉/比较通道 2 中断禁止
0	**CC1IE**	捕捉/比较通道 1 中断使能位(Capture/Compare channel 1 Interrupt Enable) 1：捕捉/比较通道 1 中断使能 0：捕捉/比较通道 1 中断禁止

35.7.6 LPTIM 中断标志寄存器（LPTIM16_ISR）

名称	**LPTIM16_ISR**							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名			QDIR	-			CAP2OVR	CAP1OVR
位权限			R-0	U-0			R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGIF	OVIF	QDCIF	-			CC2IF	CC1IF
位权限	R/W-0	R/W-0	R/W-0	U-0			R/W-0	R/W-0

位号	位名	说明
31:14	--	**未实现：读为0**
13	**QDIR**	Quadrature Decoder 方向寄存器 (Quad-decoder Direction) 0：CH1 领先于 CH2 1：CH1 滞后于 CH2
12:10	--	**未实现：读为0**
9	**CAP2OVR**	通道 2 捕捉溢出，硬件置位，软件写 1 清零(Channel 2 Over-Capture Interrupt Flag) 1：输入捕捉模式下，CC2IF 为 1 时出现新的捕捉，发生 overrun 0：没有发生 overrun
8	**CAP1OVR**	通道 1 捕捉溢出，硬件置位，软件写 1 清零(Channel 1 Over-Capture Interrupt Flag) 1：输入捕捉模式下，CC1IF 为 1 时出现新的捕捉，发生 overrun 0：没有发生 overrun
7	**TRIGIF**	外部触发到来中断标志位，写 1 清零(External Trigger Interrupt Flag) 1：外部触发到来中断产生 0：无中断产生
6	**OVIF**	计数器溢出中断使能位，写 1 清零(Counter Over-Flow Interrupt Flag) 1：计数器溢出中断产生 0：无中断产生
5	**QDCIF**	Quadrature Decoder 方向改变中断，硬件置位，软件写 1 清零(Quad-decoder Direction Change Interrupt Flag) 0：没有改变方向 1：方向改变
4:2	--	**未实现：读为0**
1	**CC2IF**	捕捉/比较通道 2 中断标志，硬件置位，软件写 1 清零(Capture/Compare channel 2 Interrupt Flag) 1：计数器值和比较值 2 匹配，或者发生捕捉事件 0：无中断产生
0	**CC1IF**	捕捉/比较通道 1 中断标志，硬件置位，软件写 1 清零(Capture/Compare channel 1 Interrupt Flag) 1：计数器值和比较值 1 匹配，或者发生捕捉事件 0：无中断产生

35.7.7 LPTIM 控制寄存器（LPTIM16_CR）

名称	LPTIM16_CR
offset	0x18

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

位号	位名	说明
31:1	--	**未实现：读为0**
0	**EN**	LPTIM 使能位(LPTIM Enable) 1：使能计数器计数 0：禁止计数器计数

35.7.8 LPTIM 捕捉比较寄存器 1（LPTIM16_CCR1）

名称	LPTIM16_CCR1							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:16	--	**未实现：读为0**
15:0	**CCR1**	捕捉/比较值寄存器 1 (Channel1 Capture/Compare Register)

35.7.9 LPTIM 捕捉比较寄存器 2（LPTIM16_CCR2）

名称	LPTIM16_CCR2							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:16	**--**	**未实现：读为0**
15:0	**CCR2**	捕捉/比较值寄存器 2 (Channel2 Capture/Compare Register)

36 实时时钟（RTC）

36.1 概述

实时时钟(RTC)模块可长时间维持精确计时，功耗极低，在所有功耗模式下都可以工作。

主要特性如下：

- BCD 时间，完整万年历（00~99 年）
- 周期唤醒中断
- 闹钟功能
- 可配置周期定时信号输出
- 数字调校，精度+/-0.477ppm
- 反馈电阻集成
- RTC 计时器部分不复位

36.2 结构框图

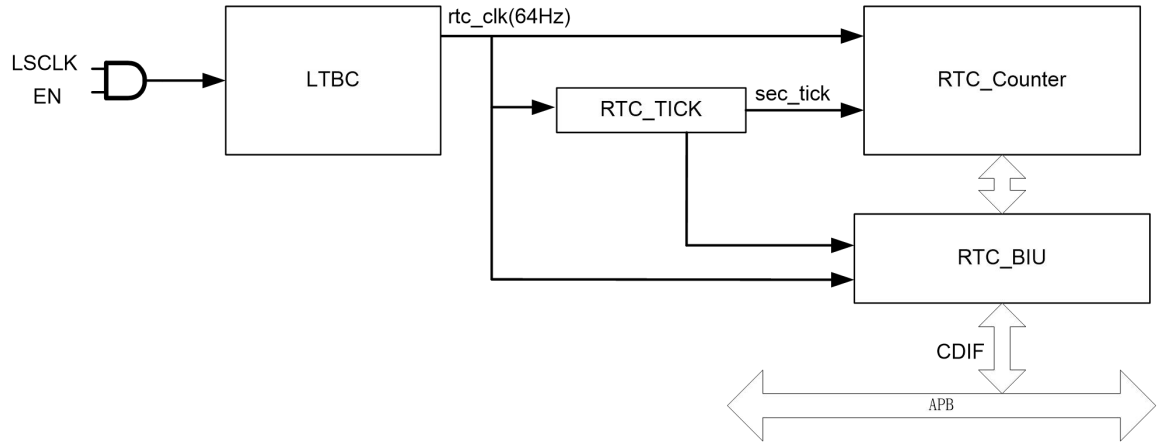


图 36-1RTC 结构框图

LTBC 模块为低功耗时基计数器模块，用于产生系统所需的低速工作时钟，具体描述见 29.3.1 节 LTBC 功能介绍。

RTC_TICK 模块主要用来产生每秒跳变的秒冲信号 sec_tick 以及产生中断需要的 2Hz，4Hz，8Hz，16Hz 等信号。Sec_tick 信号输出到 RTC_Counter 模块用来实现万年历的计数器同步。

RTC_Counter 模块是 RTC 的万年历实现模块，包括秒计数器，分钟计数器，小时计数器，天计数器，周天计数器，月计数器以及年计数器。模块可以实现闰年的自动识别。

36.3 工作原理

RTC 上电后不复位，因此正常工作前需要软件置入当前时间。走时时钟使用 32.768KHz 晶体振荡器。由于晶体振荡器有可能停振，为了保证可靠性，停振检测电路使能后不断检测 32.768KHz 振荡器输出，一旦发现停振，则产生报警中断。同时，软件可以配置 XTLF 停振时是否自动将 RTC 时钟切换到 RCLP，如果使能这一功能，则 RTC 走时有一定误差，但是并不会停止；如果未使能自动切换，也可以由软件响应停振中断后进行相应处理。

36.3.1 时基计数器（LTBC）

低功耗时基计数器(LTBC)模块用于产生系统所需的低速工作时钟，功能包括：

- 通过对 LSCLK 的预分频得到 64Hz 的 RTC 与 WDT 工作时钟
- 可通过调整计数周期实现 RTC 时钟的数字调校，每 32s 调校一次可实现最小步长为 0.952ppm，调校后理论精度+/-0.477ppm
- PLL 虚拟调校可得到精确秒时标
- 可产生 1KHz、256Hz、64Hz、16Hz、4Hz、1Hz 周期中断，其中 1K 和 256Hz 是未经调校的，其他是经过数字调校的（如果使能了数字调校）
- 64Hz 预分频电路不受芯片复位影响
- 1/256s 精度授时

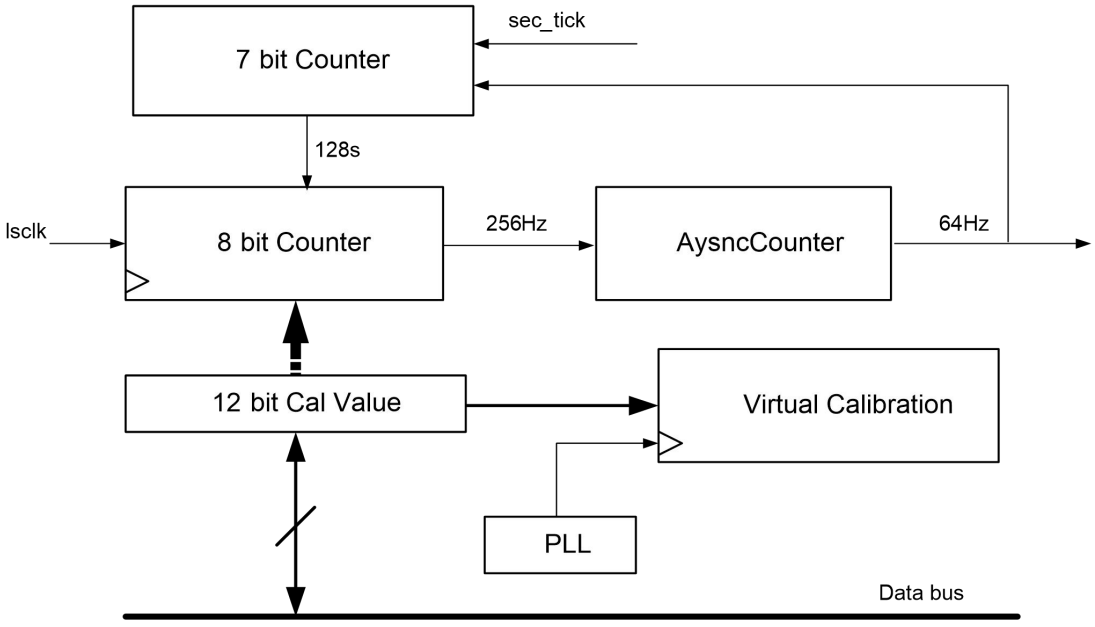


图 36-2LTBC 结构框图

36.3.2 LTBC 数字调校

LTBC主要由同步预分频计数器、异步分频计数器、时钟调校值寄存器、虚拟调校电路和控制寄存器组成。

数字调校的目的是使RTC能够在较长周期内获得平均准确的计时。由于RTC的时钟源是32768Hz，因此数字调校的最小步长是30.5us，如果在1秒内调整一次，则最高精度只能达到30.517ppm。为了得到更高精度，必须在更长时间周期内进行调整。FM33FG0A以32s为一个调校周期，每个周期内可以调整0~+/-511个32768Hz时钟周期，因此最高精度为30.5us/32s=0.952ppm，最大调校范围为+/-(511*30.517us/32s)=+/-487ppm，调校后平均最小时钟误差为+/-0.476ppm。

调校值由10bit寄存器组成，其中最高位为符号位，表示计数值增减，其余9bit表示增减的绝对值。为了提高每秒的平均精度，避免较大的秒间跃变，采取将32s调校值平均分配到每秒内的做法，其实现方法如下：

除了最高符号位，其余9bit可分为高4bit的公共值和5bit私有值，其中公共值表示32s内每秒都要调整的值，私有值表示32s内部分秒需要加减1。

Bit9	Bit[8:5]	Bit[4:0]
Sign	Common Value I	Differential Value (D)

调校值公式可表示为：Correction(ppm) = (C*32 + D)*30.517/32000000

假设只使时钟增加0.953ppm，即32s周期只增加一个30.5us，调校值写为0_0000_00001，所以公共值为0，私有值为1，只需要对32s内的一个秒周期加1即可；假设增加487ppm，即32s周期内增加511个30.5us，调校值写为0_1111_11111，公共值为15，私有值为31，表示32s中每秒都要加15，同时其中还有31s需要额外加1。

调校值举例：

ppm	ADJUST[1]	Common	Differential	Expression
0.953	0_0000_00001	0	1	1*30.517/32000000
-125.88	1_0100_00100	4	4	(4*32+4)*30.517/32000000
32.42	0_0001_00010	1	2	(1*32+2)*30.517/32000000
487.32	0_1111_11111	15	31	(15*32+31)*30.517/32000000

注：

[1] ADJUST: Clock Error Adjustment Register

通过以上方式，可以得到平滑调整的秒周期，秒和秒之间最大只差30.5ppm，可以避免集中式调整

引入的秒周期抖动。

此方案的另一个好处是，由于每个秒周期内最多只会加减15+1=16个32768Hz时钟，这表示同步计数预分频器的长度可以做的很短，理论上只要8bit（正常计数范围0~127，调校时扩展计数范围0~143，终值下限111，上限143），得到典型值256Hz同步分频输出，随后使用异步分频得到64Hz RTC走时时钟即可，较少的同步时钟负载能够降低功耗、增强EMC抗干扰性能。

为避免时序冲突，软件应在秒中断后更新ADJUST并启动时钟调校。

以ADJUST=0_0001_0000000为例，在每一秒的最后添加1个32768周期。

36.3.3 BCD 时间

秒计时

秒计时仅需7bit，从0计数到59，其中bit[3:0]为1秒单位，计数范围0-9；bit[6:4]为10秒单位，计数范围0-5。当计数满60s后触发秒进位信号使分钟计数器加1。

Bit6-4	Bit3-0
0-5	0-9

分钟计时

分计时也仅需7bit，计数范围与秒相同，因此实现方法也相同。

Bit6-4	Bit3-0
0-5	0-9

小时计时

小时计数范围为 0-24，仅需 6bit：

Bit5-4	Bit3-0
0-2	0-9

天计时

天计数范围为 1-31，仅需 6bit，从 1 开始计数，根据月份以及闰年计数到 28/29/30/31，计满后触发天进位信号使月计数器加 1。

Bit5-4	Bit3-0
0-3	0-9

星期计时

星期计数范围为 0-6，仅需 3bit，从 0 到 6 循环计数。

Bit2-0
0-6

月计时

月计数范围为 1-12，仅需 5bit，从 1 开始计数到 12，计满后触发月进位信号使年计数器加 1。

Bit4	Bit3-0
0-1	0-9

年计时

年计数范围为 0-99，需 8bit，从 0 到 99 循环计数。

Bit7-4	Bit3-0
0-9	0-9

36.3.4 RTC 使能与停止

RTC 可以由软件使能或关闭。上电后 RTC 默认关闭，软件在 XTLF 起振后通过操作 EN 寄存器使能 RTC 走时。

当 EN 寄存器清零时，RTC 内部走时时钟被门控，以节省功耗。

36.3.5 RTC 时间设置

软件可以在任意时刻直接设置 RTC 时间寄存器，通常建议在 XTLF 完成起振后再设置时间；由于设置时间寄存器的操作与 RTC 走时为异步操作关系，建议软件在秒中断事件之后进行时间设置，并且在置时后读出时间值校验。

注意，硬件并不检查时间合法性，软件须保证写入的 BCD 时间正确。

同时 FM33FG0A 支持 ms 级授时，即可以设置时间到 3.9ms 级别精度（1/256s）。此外，当软件写入秒时间时，硬件自动清零 64Hz->1Hz 的秒内计数器，以便实现秒对齐。

为了提高抗干扰能力，FM33FG0A 提供时间写保护功能，必须先对写保护寄存器写入 0xACACACAC，才能改写时间寄存器，置时完成后软件可以通过写入任意其他值来禁止时间寄存器的写入，恢复写保护。

36.3.6 RTC 时间读取

时间读取方式 1：

- 读当前时间寄存器值
- 再次读当前时间寄存器值

- 如果 2 次读取内容一致，则为正确的当前时间；如果两次读取内容不一致，则重复前两个步骤。

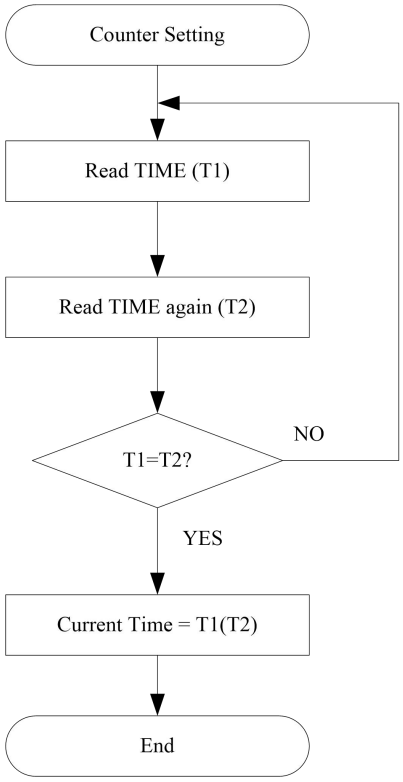


图 36-3RTC 时间读取流程图

时间读取方式 2：

软件在 1s 中断发生后立即读取时间寄存器，能保证读到正确的当前时间值。

36.3.7 闰年判断

FM33FG0A 的 RTC 模块会自动判断闰年。

闰年的条件：(mod 400 == 0) or (mod 4 ==0 and mod 100 <> 0)

这里只考虑 2000-2099 范围。

36.4 寄存器

模块基地址：0x4001_1000

offset 地址	名称	符号
0x00000000	RTC 写使能寄存器 （RTC Write Enable Register）	RTC_WER
0x00000004	RTC 中断使能寄存器 （RTC Interrupt Enable Register）	RTC_IER
0x00000008	RTC 中断标志寄存器 （RTC Interrupt Status Register）	RTC_ISR
0x0000000C	BCD 时间秒寄存器 （BCD format time second registers）	RTC_BCDSEC
0x00000010	BCD 时间分钟寄存器 （BCD format time minute registers）	RTC_BCDMIN
0x00000014	BCD 时间小时寄存器 （BCD format time hour registers）	RTC_BCDHOUR
0x00000018	BCD 时间天寄存器 （BCD format time day registers）	RTC_BCDDAY
0x0000001C	BCD 时间星期寄存器 （BCD format time week registers）	RTC_BCDWEEK
0x00000020	BCD 时间月寄存器 （BCD format time month registers）	RTC_BCDMONTH
0x00000024	BCD 时间年寄存器 （BCD format time year registers）	RTC_BCDYEAR
0x00000028	闹钟寄存器 （RTC Alarm Register）	RTC_ALARM
0x0000002C	RTC 时间信号输出寄存器 （RTC Time Mark Select）	RTC_TMSEL
0x00000030	LTBC 数值调整寄存器 （RTC time Adjust Register）	RTC_ADJUST
0x0000003C	亚秒计数值寄存器 （RTC Sub-Second Counter）	RTC_SBSCNT
0x00000040	RTC 控制寄存器 （RTC Control Register）	RTC_CR

36.4.1 RTC 写使能寄存器（RTC_WER）

名称	RTC_WER							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							WE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	WE	RTC 写使能寄存器 (RTC Write Enable) 当 CPU 向 RTCWE 写入 0xACACACAC 时，允许 CPU 向 RTC 的 BCD 时间寄存器写入初值，这时 RTCWE 置 1；当 CPU 向 RTCWE 写入不为 0xACACACAC 的任意值时恢复写保护,这时 RTCWE 清 0。

36.4.2 RTC 中断使能寄存器（RTC_IER）

名称	RTC_IER							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			ADJ_IE	ALARM_IE	1KHZ_IE	256HZ_IE	64HZ_IE
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
位名	16HZ_IE	8HZ_IE	4HZ_IE	2HZ_IE	SEC_IE	MIN_IE	HOUR_IE	DAY_IE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12	ADJ_IE	调校周期中断使能 (time Adjust Interrupt Enable) 1：中断使能打开 0：中断使能禁止
11	ALARM_IE	闹钟中断使能 (Alarm Interrupt Enable) 1：中断使能打开 0：中断使能禁止
10	1KHZ_IE	1khz 中断使能 (1Khz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
9	256HZ_IE	256hz 中断使能 (256hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
8	64HZ_IE	64hz 中断使能 (64hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
7	16HZ_IE	16hz 中断使能 (16hz periodic Interrupt Enable)

位号	助记符	功能描述
		1：中断使能打开 0：中断使能禁止
6	8HZ_IE	8hz 中断使能 (8hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
5	4HZ_IE	4hz 中断使能 (4hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
4	2HZ_IE	2hz 中断使能 (2hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
3	SEC_IE	秒中断使能 (1hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
2	MIN_IE	分中断使能 (Minute Interrupt Enable) 1：中断使能打开 0：中断使能禁止
1	HOUR_IE	小时中断使能 (Hour Interrupt Enable) 1：中断使能打开 0：中断使能禁止
0	DAY_IE	天中断使能 (Day Interrupt Enable) 1：中断使能打开 0：中断使能禁止

36.4.3 RTC 中断标志寄存器（RTC_ISR）

名称	RTC_ISR							
offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			ADJ_IF	ALARM_IF	1KHZ_IF	256HZ_IF	64HZ_IF
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
位名	16HZ_IF	8HZ_IF	4HZ_IF	2HZ_IF	SEC_IF	MIN_IF	HOUR_IF	DAY_IF
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12	ADJ_IF	调校周期中断标志。写 1 清零 (time Adjust Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生

位号	助记符	功能描述
11	ALARM_IF	闹钟中断标志。写 1 清零 (Alarm Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
10	1KHZ_IF	1khz 中断标志。写 1 清零 (1Khz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
9	256HZ_IF	256hz 中断标志。写 1 清零 (256hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
8	64HZ_IF	64hz 中断标志。写 1 清零 (64hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
7	16HZ_IF	16hz 中断标志。写 1 清零 (16hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
6	8HZ_IF	8hz 中断标志。写 1 清零 (8hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
5	4HZ_IF	4hz 中断标志。写 1 清零 (4hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
4	2HZ_IF	2hz 中断标志。写 1 清零 (2hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
3	SEC_IF	秒中断标志。写 1 清零 (1hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
2	MIN_IF	分中断标志。写 1 清零 (Minute Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
1	HOUR_IF	小时中断标志。写 1 清零 (Hour Interrupt Flag,write 1 to clear) 1：中断置位 0：无中断产生
0	DAY_IF	天中断标志。写 1 清零 (Day Interrupt Flag,write 1 to clear) 1：中断置位 0：无中断产生

36.4.4 BCD 时间秒寄存器（RTC_BCDSEC）

名称	RTC_BCDSEC

Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	SEC						
位权限	U-0	R/W-xxx xxxx						

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6:0	SEC	秒时间数值，BCD 格式。(Binary-Coded Decimal format Seconds Register)

36.4.5 BCD 时间分钟寄存器（RTC_BCDMIN）

名称	RTC_BCDMIN							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	MIN						
位权限	U-0	R/W-xxx xxxx						

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6:0	MIN	分钟时间数值，BCD 格式。(Binary-Coded Decimal format Minutes Register)

36.4.6 BCD 时间小时寄存器（RTC_BCDHOUR）

名称	RTC_BCDHOUR							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		HOUR					
位权限	U-0		R/W-xx xxxx					

位号	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5:0	HOUR	小时数值，BCD 格式。(Binary-Coded Decimal format Hours Register)

36.4.7 BCD 时间天寄存器（RTC_BCDDAY）

名称	RTC_BCDDAY							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		DAY					
位权限	U-0		R/W-xx xxxx					

位号	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5:0	DAY	天数值，BCD 格式。(Binary-Coded Decimal format Date Register)

36.4.8 BCD 时间星期寄存器（RTC_BCDWEEK）

名称	RTC_BCDWEEK							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					WEEK		
位权限	U-0					R/W-xxx		

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2:0	WEEK	周数值，BCD 格式。(Binary-Coded Decimal format Week Register)

36.4.9 BCD 时间月寄存器（RTC_BCDMONTH）

名称	RTC_BCDMONTH							
Offset	0x00000020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			MONTH				
位权限	U-0			R/W-x xxxx				

位号	助记符	功能描述
31:5	-	RFU：未实现，读为 0
4:0	MONTH	月数值，BCD 格式。(Binary-Coded Decimal format Month Register)

36.4.10 BCD 时间年寄存器（RTC_BCDYEAR）

名称	RTC_BCDYEAR							
Offset	0x00000024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	YEAR							
位权限	R/W-xxxx xxxx							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	YEAR	年数值，BCD 格式。(Binary-Coded Decimal format Year Register)

36.4.11 闹钟寄存器（RTC_ALARM）

名称	RTC_ALARM							
Offset	0x00000028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		HOUR					
位权限	U-0		R/W-00 0000					
位	Bit15	Bit14	BIT13	BIT12	BIT11	BIT10	Bit9	Bit8
位名	-	MIN						
位权限	U-0	R/W-000 0000						
位	Bit7	Bit6	BIT5	BIT4	BIT3	BIT2	Bit1	Bit0
位名	-	SEC						
位权限	U-0	R/W-000 0000						

位号	助记符	功能描述
31:22	-	RFU：未实现，读为 0
21:16	HOUR	闹钟的小时数值。(Alarm Hour Register)
15	-	RFU：未实现，读为 0
14:8	MIN	闹钟的分数值。(Alarm Minute Register)
7	-	RFU：未实现，读为 0
6:0	SEC	闹钟的秒数值。(Alarm Second Register)

36.4.12 RTC 时间信号输出寄存器（RTC_TMSEL）

名称	RTC_TMSEL							
offset	0x0000002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TMSEL			
位权限	U-0				R/W-0000			

位号	助记符	功能描述
31:4	-	RFU：未实现，读为 0
3:0	TMSEL	频率输出选择信号：(Time Mark Select) 0000：RFU 0001：RFU 0010：输出秒计数器进位信号，高电平宽度 1s 0011：输出分计数器进位信号，高电平宽度 1s 0100：输出小时计数器进位信号，高电平宽度 1s 0101：输出天计数器进位信号，高电平宽度 1s 0110：输出闹钟匹配信号 0111：输出 32 秒方波信号 1000：RFU 1001：反向输出秒计数器进位信号 1010：反向输出分计数器进位信号 1011：反向输出小时计数器进位信号 1100：反向输出天计数器进位信号 1101：反向输出闹钟匹配信号 1110：RFU 1111：输出 RTC 内部秒时标方波

36.4.13 LTBC 数值调整寄存器（RTC_ADJUST）

名称	RTC_ADJUST							
Offset	0x00000030							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						ADSIGN	ADJUST [8]
位权限	U-0						R/W-x	R/W-x
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADJUST[7:0]							
位权限	R/W-xxxx xxxx							

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0

位号	助记符	功能描述
9	ADSIGN	LTBC 补偿方向（Adjust Sign） 0：表示增加计数初值 1：表示减少计数初值
8:0	ADJUST	LTBC 补偿调整数值（Time Adjust）

36.4.14 毫秒计数值寄存器（RTC_SBSCNT）

名称	RTC_SBSCNT							
offset	0x0000003C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSCNT							
位权限	R/W-xxxx xxxx							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	MSCNT	毫秒计数器值，有效位 8bit，精度 3.9ms。(Milli-Second Counter)

36.4.15 RTC 控制寄存器（RTC_CR）

名称	RTC_CR							
offset	0x00000040							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	EN	RTC 使能寄存器（RTC enable） 0：关闭 RTC（内部走时时钟关闭）

Bit	助记符	功能描述
		1：使能 RTC 注：此寄存器只受上下电复位影响

37 模数转换器（ADC）

37.1 概述

FM33FG0A 带有 2Msps 12bit SAR-ADC，可实现温度、电池电压或其他直流信号的测量功能。主要特点为：

- 工作电压 2.5～5.5V
- 输入信号幅度 0~VREF+
- 可灵活选择基准源
- 最高采样率 2Msps（F_{ADC}=32Mhz）
- 最大 31 个外部输入通道，单端输入
- 内部采样通道
  - 温度传感器
  - 内部基准测量
  - DAC 输出测量
  - VREFP 和 VREFN 自检通道
- 可配置的采样保持时间
- 支持单次转换和连续转换
- 支持 DMA
- 支持过采样硬件平均，最高 16bit 输出（256 次平均）
- 超低功耗架构

37.2 结构框图

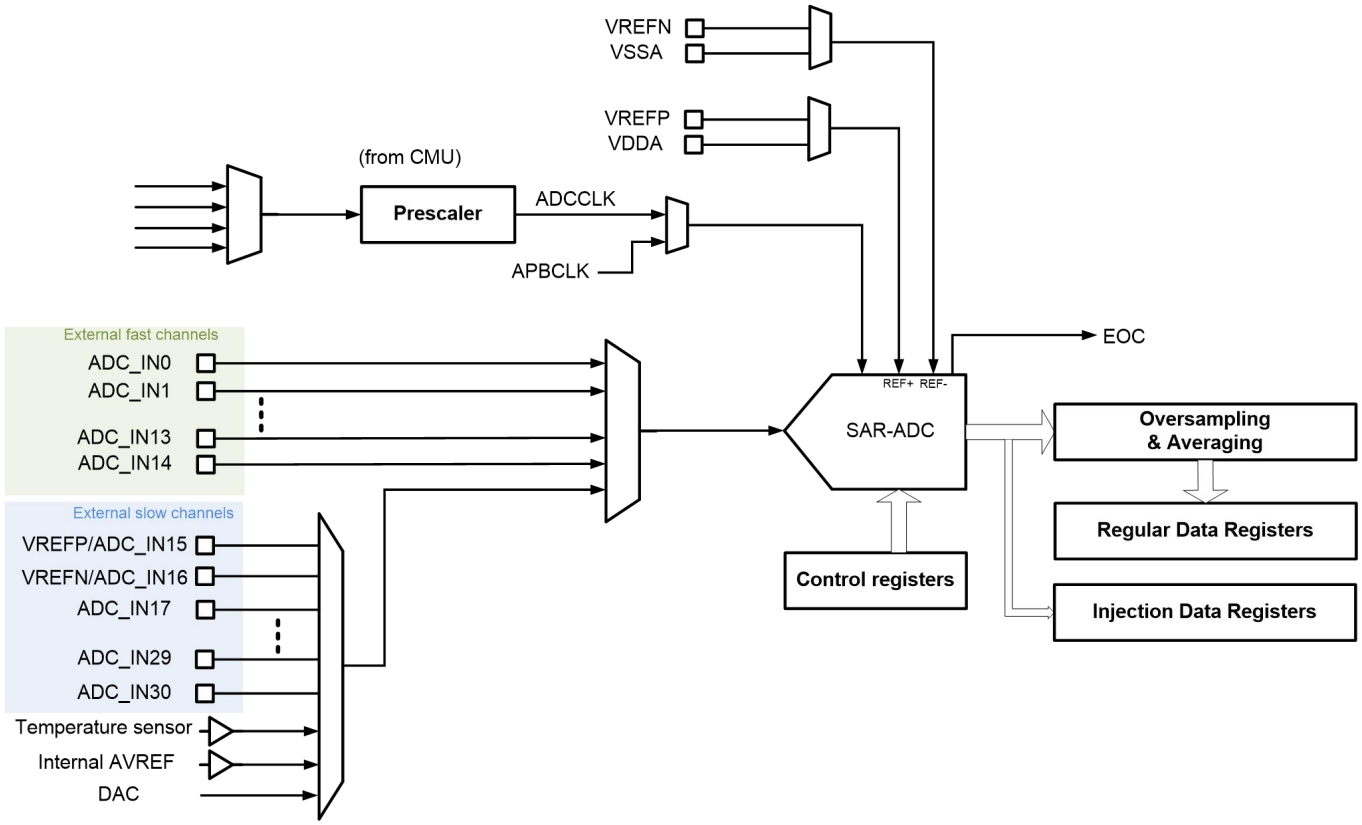


图 37-1 ADC 结构框图

ADC 的正负基准可以在 VREFP/VREFN 和 VDDA/VSSA 之间灵活选择，上电后默认基准源是 VDDA/VSSA，此时 VREFP 和 VREFN 管脚可以作为模拟输入通道使用。

温度传感器和内部 AVREF 输出阻抗较大，因此在 ADC 采样之前应使能内部 buffer。

37.3 输入通道

ADC 最大支持 3 个内部通道和 31 个外部通道。

通道	IO	说明
ADC_IN0	PD9	外部快速通道，单端输入
ADC_IN1	PD10	
ADC_IN2	PD11	
ADC_IN3	PD12	
ADC_IN4	PD13	
ADC_IN5	PD14	
ADC_IN6	PD15	
ADC_IN7	PF6	
ADC_IN8	PF7	

通道	IO	说明	
ADC_IN9	PF8		
ADC_IN10	PD6		
ADC_IN11	PD0		
ADC_IN12	PD1		
ADC_IN13	PD2		
ADC_IN14	PD3		
ADC_IN15	VREFP	外部慢速通道，单端输入 ***注意，当使用 ADC_IN15 和 ADC_IN16 通道时，必须将 ADC 基准源选择为 VDDA***	
ADC_IN16	VREFN		
ADC_IN17	PD4		
ADC_IN18	PD5		
ADC_IN19	PH15		
ADC_IN20	PE9		
ADC_IN21	PA13		
ADC_IN22	PA14		
ADC_IN23	PA0		
ADC_IN24	PA1		
ADC_IN25	PC7		
ADC_IN26	PC8		
ADC_IN27	PC9		
ADC_IN28	PC10		
ADC_IN29	PC11		
ADC_IN30	PC12		
DAC	N/A	DAC 输出采样通道	内部慢速通道
TempSensor		温度传感器采样通道	
AVREF		内部 1.0V 基准源采样通道	
VREFINT		内部产生的基准电压	

表 37-1 ADC 输入通道

37.4 单端输入

ADC 仅支持单端输入模式。

单端模式下 ADC 转换的是单个输入引脚对地的电压值，输入幅度范围是 0~VREF+(VDDA)。为了避免可能的波形削顶风险导致输入失真，一般建议输入信号最大幅值不超过 0.95*VREF+。

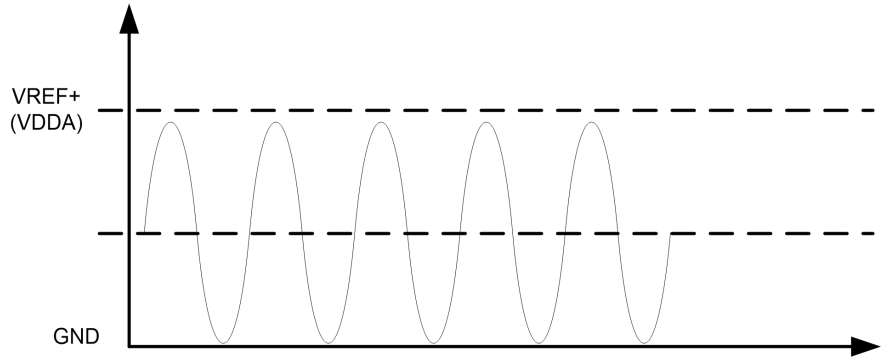


图 37-2 单端输入

37.5 工作时序

ADC 有 offset 校准和正常工作两种时序。

上电后建议先进行一次前台校准，以获得更好的性能。校准后无需重新校准，除非芯片发生全局复位，或者电源电压、温度发生较大变化。Offset 校准通过软件置位 CALEN 启动，校准过程包含多个采样转换周期，周期数由 OSCAL_CYCLE 寄存器配置，一般推荐配置为 8，即完成校准需要 128 个 ADCCLK 周期。校准完成后置位 EOCAL 标志寄存器。

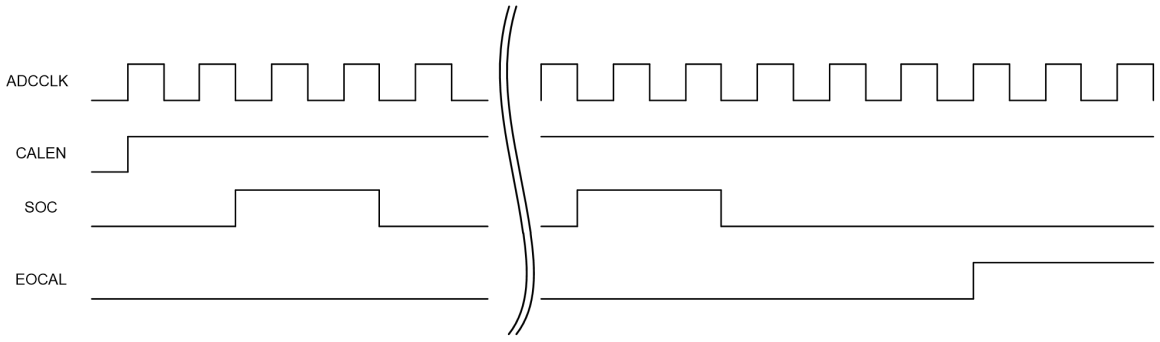


图 37-3 ADC 校准时序

采样转换时，通过 SOC 信号启动 ADC 采样，SOC 高电平宽度控制了 ADC 采样时间，SOC 变低后启动转换，转换周期为 14 个 ADC_CLK 周期，采样时间可配置，最短 2 个 ADC_CLK，最长 512 个 ADC_CLK。

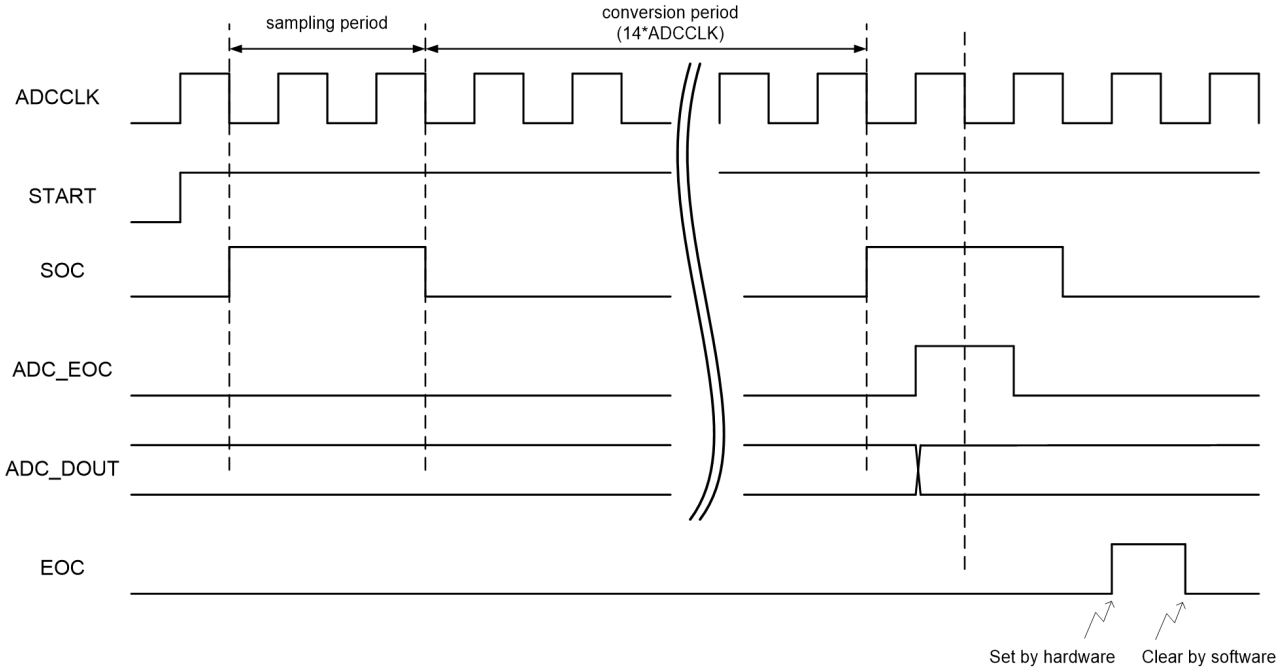


图 37-4 ADC 采样转换时序

多通道转换序列的时序示意图如下：

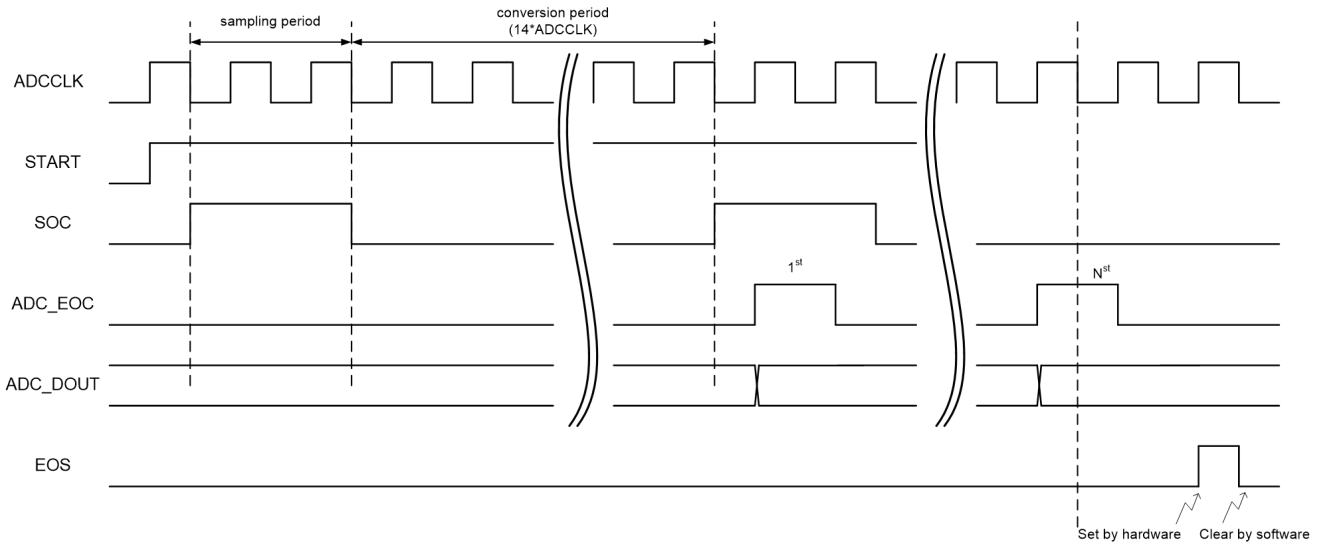


图 37-5 ADC 采样序列时序

SOC 信号对齐 ADC_CLK 的下降沿，ADC 产生的 EOC 信号对齐 ADC_CLK 的上升沿。

当 ADC_CLK 为 16MHz，采样时间配置为 2*ADCCLK 时，转换速率为 1Msps。

当 ADC_CLK 为 32MHz，采样时间配置为 2*ADCCLK 时，转换速率为 2Msps。

37.6 功能描述

37.6.1 使用 VDDA 作为基准

ADC 一般使用电源电压作为基准电压，在电源电压发生变化时，特定输入信号电平对应的转换值也会发生变化。为了能够得到准确的绝对电压，可以由两种解决方案。

使用快速内部基准作为参考信号

芯片内部有一个快速建立的内部基准源，其输出电压是 1.0V，并且每颗芯片出厂时都经过校准，常温下精度为 1.0V+/-0.5%。

- ADC 采样并转换快速基准输出，并通过以下公式换算 VDDA 电压（假设 ADC 采样快速基准源的 12 位转换结果是 AVREF_DATA）：

$$VDDA = \frac{4095}{AVREF_DATA} \times 1.0V$$

- 假设 ADC 对某个输入通道的采样值为 ADC_DATA，通过以下公式可以得到当前某个输入通道的实际电压（12bit 输出）

$$V_{CHANNEL} = \frac{ADC_DATA}{AVREF_DATA} \times 1.0V$$

37.6.2 温度传感器

根据温度传感器采样值计算当前温度

ADC 使用内部通道测量 PTAT 输出电压，得到转换数据 TS_DATA，当 ADC 工作参考电压是 VDDA 时，根据以下步骤可以计算当前实际温度：

1) 由以下公式计算当前温度传感器输出的绝对电压值

$$\mathrm{VPTAT} = \frac{VREFIN_CAL \times TS_DATA}{VREFINT_DATA \times 4095} \times 3V$$

2) 由以下公式计算温度标定时（30°C）温度传感器输出的绝对电压值

$$\mathrm{VPTAT_30C} = \frac{TS_CAL1}{4095} \times 3V$$

3) 根据温度传感器输出斜率计算当前绝对温度

$$\mathrm{Temperature} = \frac{VPTAT - VPTAT_30C}{3.06mV/C} + 30C$$

其中，TS_DATA 是 ADC 采样当前温度传感器输出的转换值；由于不知道当前 VDDA 的准确电平，因此这个转换值需要根据 VREFINT 的转换结果进行比例缩放；TS_CAL1 是芯片生产时在 30C+/-1C、VDDA=3.0V 的条件下进行温度定标的转换结果，这个数据保存在 flash 中。

如果 ADC 当前工作参考电压是 VREFP，则当前温度可以通过下式计算：

$$Temperature = \frac{TS_DATA \times \frac{VREFP}{3.0V} - TS_CAL30}{slope} + 30C$$

根据温度传感器采样值进行 RTC 温度补偿

如果温度采样值只是用来做 RTC 温度补偿，则并不需要计算实际的温度值（-40~85 的十进制数），仅需要根据 $TS_DATA \times \frac{VREFINT_CAL}{VREFINT_DATA}$ 的 12bit 结果（从物理原理来看这个结果就是 12bit），以 TS_CAL1 为中心点进行地址查表即可。因为上式计算结果代表的是折算到 VDDA=3V 情况下温度传感器的 12 位输出结果，它与 TS_CAL1 的差值，即为偏离 30°C 多少个 LSB，用这个信息作为地址对温度补偿校正表格查表，即可得到相应温度下的校正值。

使用温度传感器功能时，需要使能内部基准源的温度传感器输出，即同时置位 VREF_EN 和 PTAT_EN 寄存器，并且置位 VPTAT_BUFFER_EN 寄存器使能 PTAT 缓冲器，等待 5us 建立时间后，使能 ADC 对温度传感器通道进行采样。

37.6.3 温度传感器的斜率和标定

温度传感器工作电压范围为 2.5~5.5V，温度测量范围-40~+125C。PTAT 输出特性参见下表。

电源电压	VPTAT slope	ADC mV/LSB	LSB/°C
2.5~5.5V			

表 37-2 温度传感器斜率

37.6.4 可编程采样时间

通过调整采样时间，可以适应不同输入信号源的内阻。通过 SMTS1 和 SMTS2 寄存器可以选择采样时间：

SMTSx	Sampling cycles (T_{ADCCLK})
0000	2
0001	4
0010	8

SMTSx	Sampling cycles (T_{ADCCLK})
0011	12
0100	16
0101	32
0110	64
0111	80
1000	96
1001	128
1010	160
1011	192
1100	256
1101	320
1110	384
1111	512

表 37-3 ADC 采样时间

实际 ADC 的采样转换时间：

T_{CONV} = (Sampling Cycles + 14) * T_{ADCCLK}

ADC 采样时间主要由采样电容、被采样信号的输出阻抗、芯片内部输入通道阻抗和所需达到的采样精度共同决定。

下图是单端输入通道的电路结构示意：

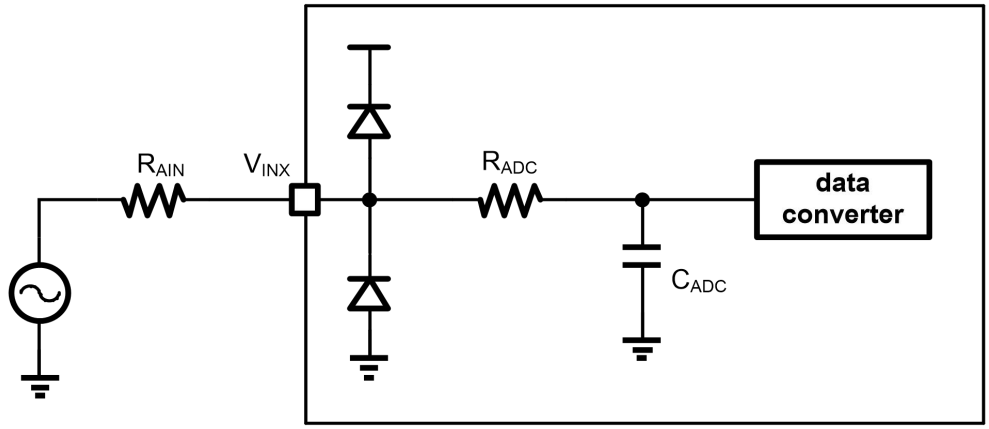


图 37-6 ADC 输入通道示意图

要求的采样时间可以根据下式估算：

$$T_{samp} = \ln\left(\frac{2^n}{SA}\right) \times \left(R_{AIN} + R_{ADC}\right) \times C_{ADC}$$

其中，n=12，SA 表示容许的采样误差，比如 0.25 代表 1/4 LSB

应用中应根据芯片手册中的相关参数、以及系统参数，计算并确定可以接受的采样时间，并根据这

个结果来配置 ADC 的工作时钟、采样周期等。

37.6.5 转换模式

ADC 支持以下转换模式：

- 单次转换
  - 半自动触发（SEMI-AUTOMATIC）
  - 全自动触发（AUTOMATIC）
- 连续转换

转换启动可以由软件或事件触发，通过寄存器选择多个事件触发源。

37.6.5.1 单次转换模式

支持半自动触发和全自动触发两种模式。

全自动触发模式：软件或硬件触发事件启动 ADC 转换后，ADC 会顺序采样所有被使能的通道，单个通道采样完成后，EOC（End of Conversion）标志置位，所有通道采样完成后，EOS（End of Sequence）标志置位，本次转换结束。假设通道 0、3、5 被使能

- 1st 触发事件：通道 0、3、5 被顺序采样，过程中产生三次 EOC，最终产生 EOS
- 2nd 触发事件：重复上述过程

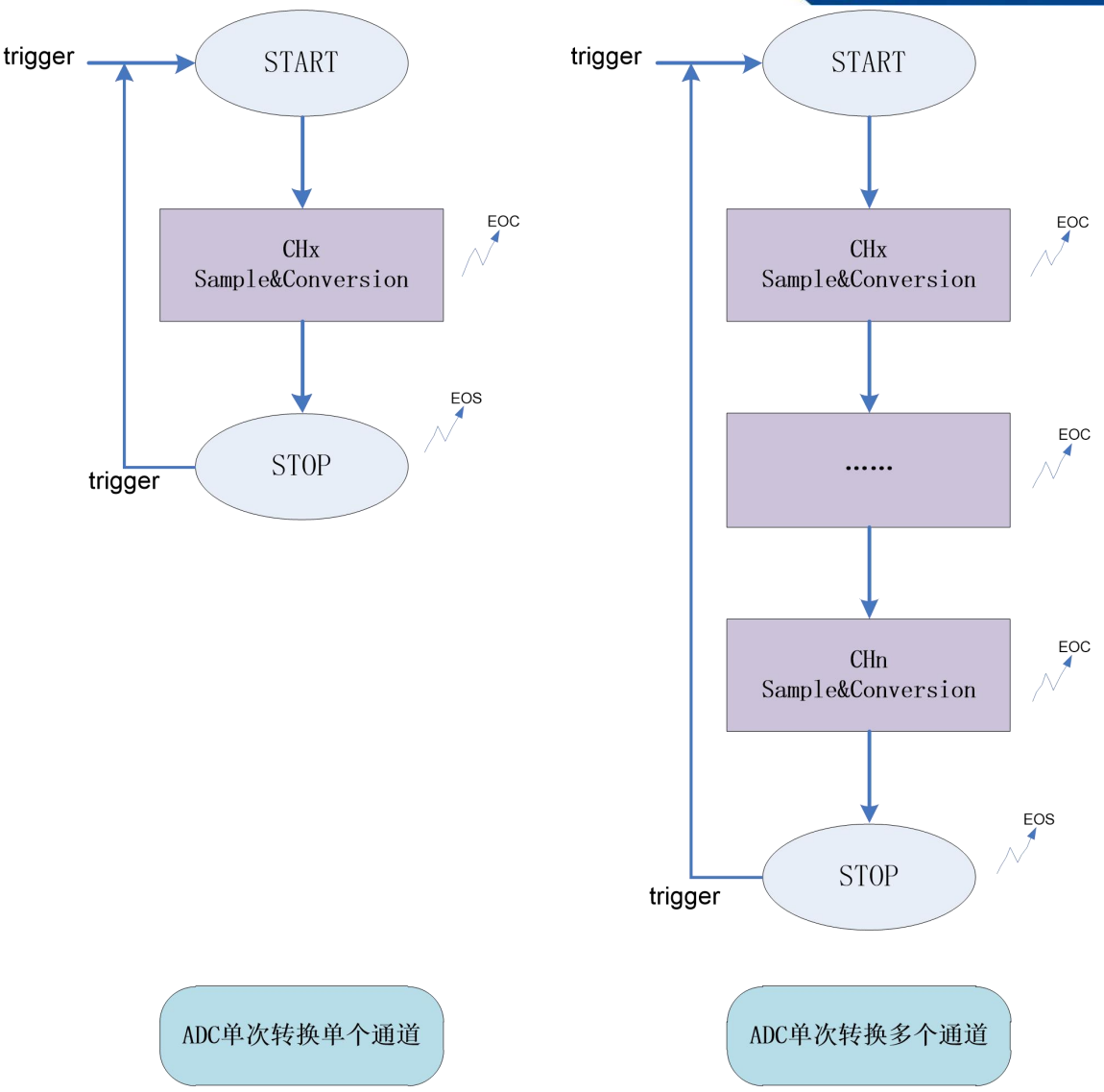


图 37-7 ADC 单次转换全自动触发模式

半自动触发模式：软件或硬件触发事件只会让 ADC 启动一次，转换一个使能通道。比如通道 0、3、5 被使能

- 1st 触发事件：通道 0 被采样，产生 EOC
- 2nd 触发事件：通道 3 被采样，产生 EOC
- 3rd 触发事件：通道 5 被采样，产生 EOC 和 EOS
- 4th 触发事件：通道 0 被采样，产生 EOC
- 5th 触发事件：通道 3 被采样，产生 EOC

……

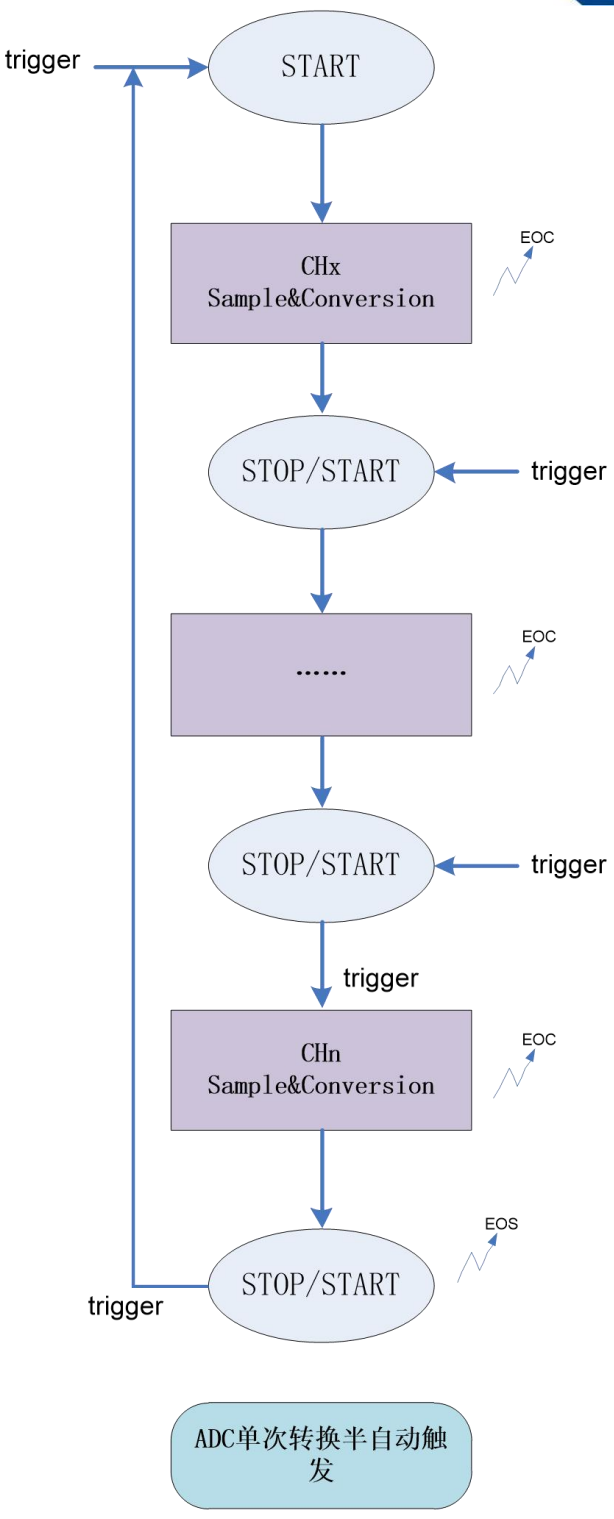


图 37-8 ADC 单次转换半自动触发模式

37.6.5.2 连续转换模式

触发事件到来后，所有使能通道被采样，并且 ADC 不会自动停止，而是循环采样，直到软件停止 ADC。

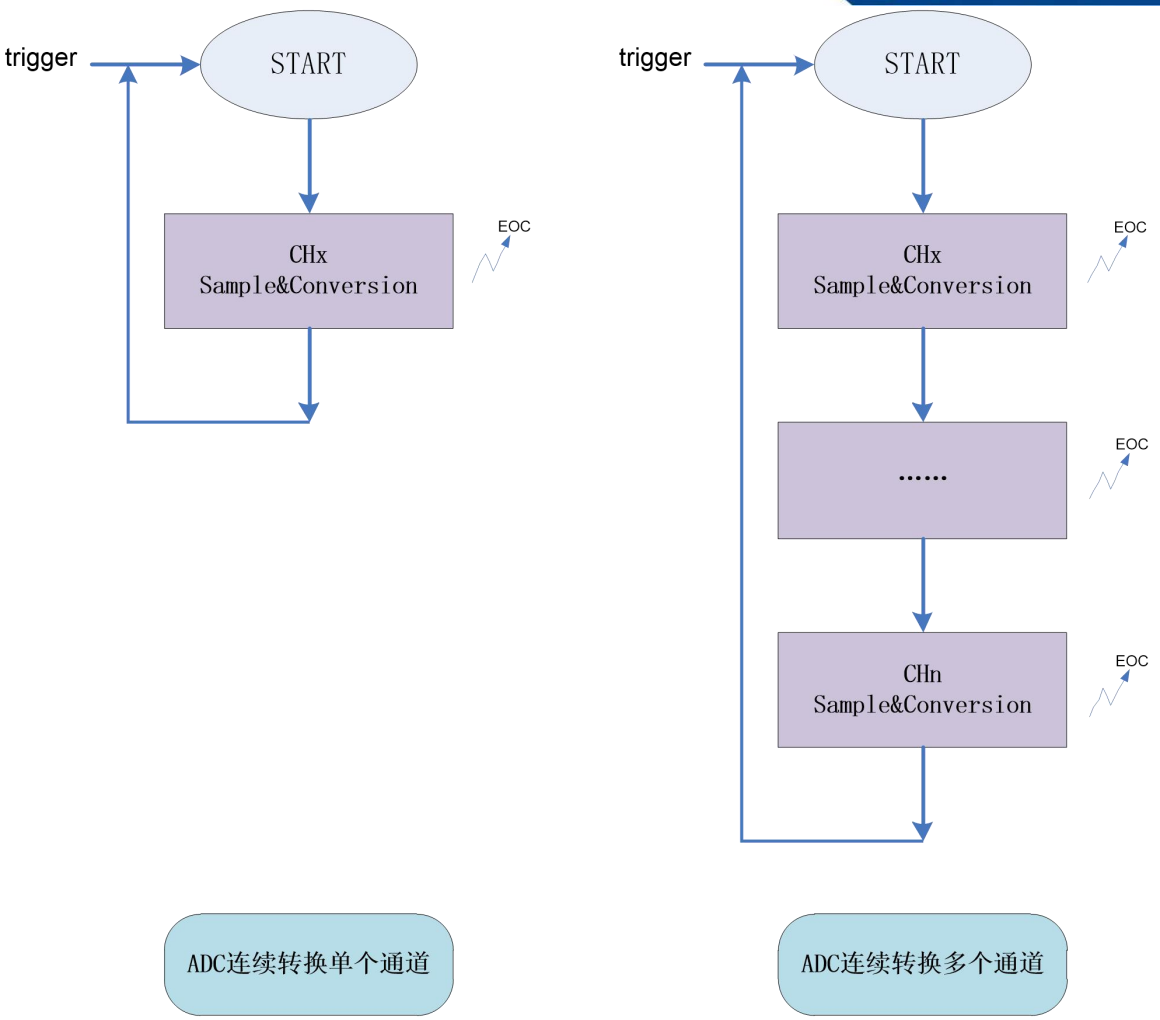


图 37-9 ADC 连续转换模式

每个通道被采样后，数据保存在 ADC_DATA 寄存器中，软件要在下次转换前及时读走数据，或者通过 DMA 进行数据搬移。如果不能及时取走数据，将引起 Overrun，置位 overrun 标志，并可以发出中断。

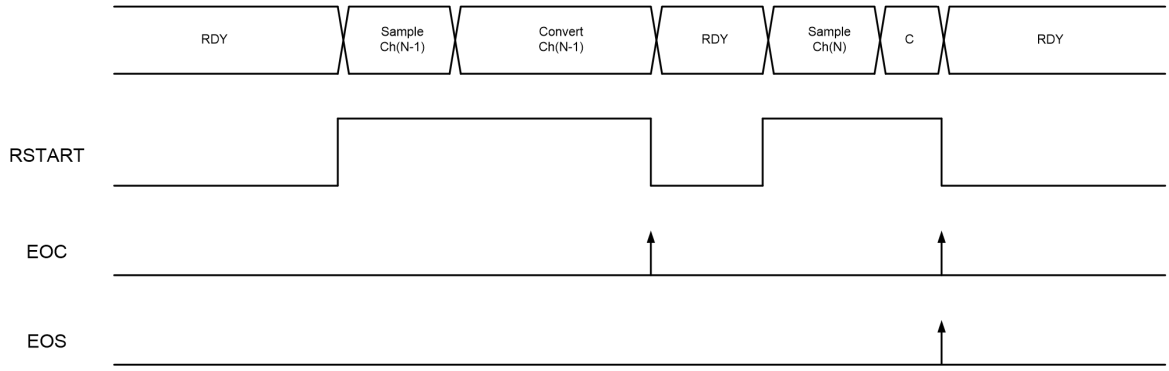
37.6.6 转换触发

ADC 使能后，转换触发支持软件或硬件事件触发。

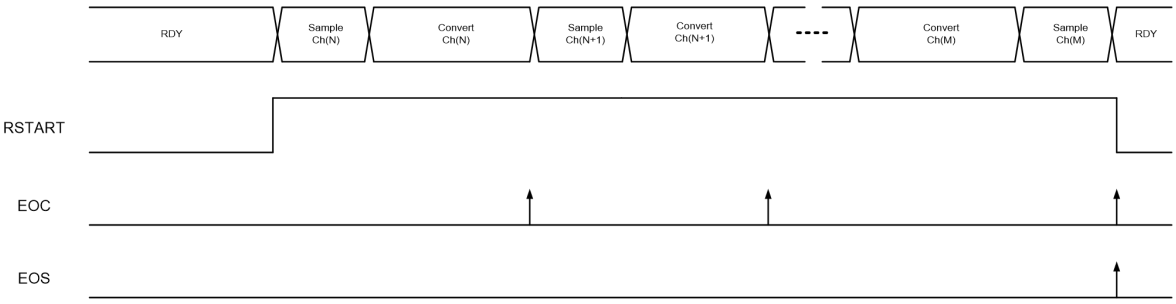
软件触发

软件通过直接置位 RSTART 寄存器启动转换。

下图是软件半自动触发的单次转换，软件每次置位 RSTART 启动一次转换：



下图是软件全自动触发的单次转换，软件置位 RSTART 后完成全部使能通道转换：

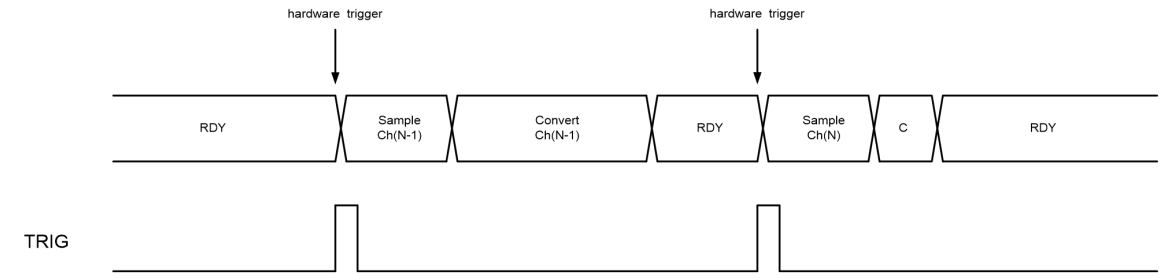


硬件触发

此时 ADC 转换由硬件触发源启动，可以配置在系统触发信号的上升沿、下降沿触发、或者同时在上升沿下降沿触发转换。

如果 ADC 正处于转换过程中，此时到来的触发信号会被忽略。

下图为硬件半自动触发：



硬件触发源示意图：

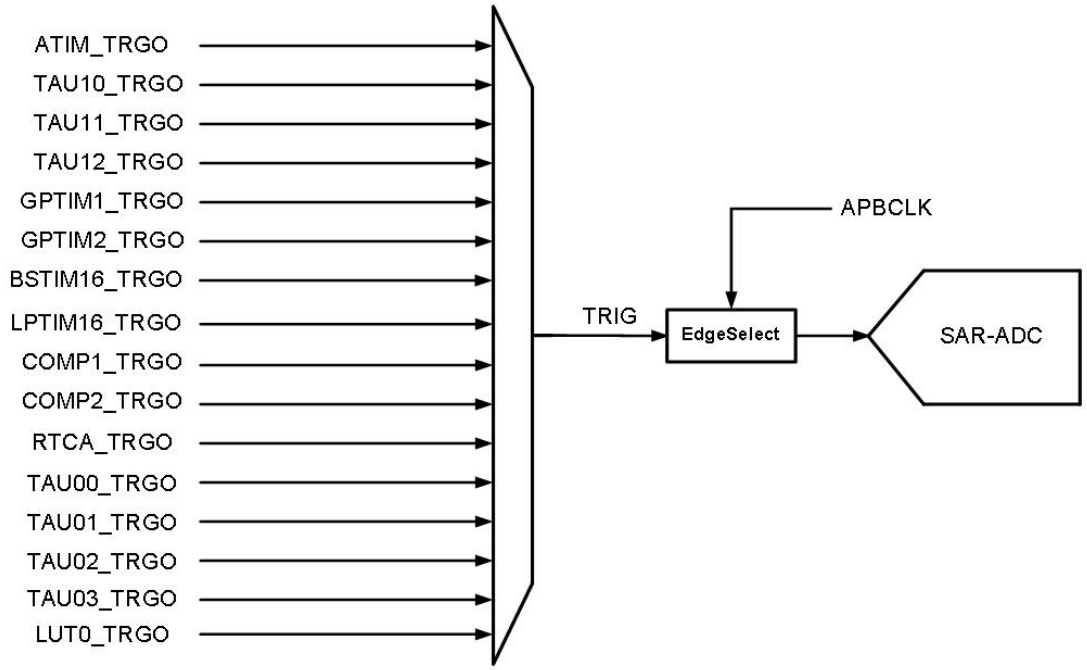


图 37-10 ADC 硬件触发源

注意：硬件触发源输入后经过 APBCLK 同步采样，因此使用硬件触发功能时必须使能 ADC 总线时钟（置位 CMU 模块中的 ADC_PCE 寄存器）。

ADC 注入组硬件触发源如下表。

JEXTS	触发信号	说明
0000	ATIM_TRGO	
0001	TAU10_TRGO	
0010	TAU11_TRGO	
0011	TAU12_TRGO	
0100	GPTIM1_TRGO	
0101	GPTIM2_TRGO	
0110	BSTIM16_TRGO	
0111	LPTIM16_TRGO	
1000	COMP1_TRGO	
1001	COMP2_TRGO	
1010	RTC_TRGO	
1011	TAU00_TRGO	
1100	TAU01_TRGO	
1101	TAU02_TRGO	
1110	TAU03_TRGO	
1111	LUT0_TRGO	

图 37-11 ADC 注入组硬件触发源

触发源时序特征分类。

触发源	时钟域	当 ADC 工作时钟选择为 APBCLK 时的触发延迟	说明

		(T_{APBCLK})	
软件触发	APBCLK	3.5	可实现同步触发，确定延迟
LUTx_TRGO		4.5	
GPTIMx_TRGO		4.5	
COMPx_TRGO		4.5	
ATIM_TRGO	其他	2~3	异步时钟触发
BSTIM16_TRGO			
LPTIM16_TRGO			
RTC_TRGO			

37.6.7 转换停止

软件通过置位 RSTOP 和 JSTOP 寄存器，可以停止当前正在进行的常规组和注入组转换。常规组停止和注入组停止互不影响：

- 在注入组转换时置位 JSTOP，ADC 立即停止注入组转换并返回常规组转换序列
- 在注入组转换时置位 RSTOP，ADC 注入组转换仍继续执行，注入组转换完成后，由于常规组已经停止，不会再回到常规组转换序列
- RSTOP 会停止当前常规组序列，此后如果新的触发到来，则常规组转换序列将从头开始

下图示例显示了在常规组半自动硬件触发模式下，软件置位 RSTOP：

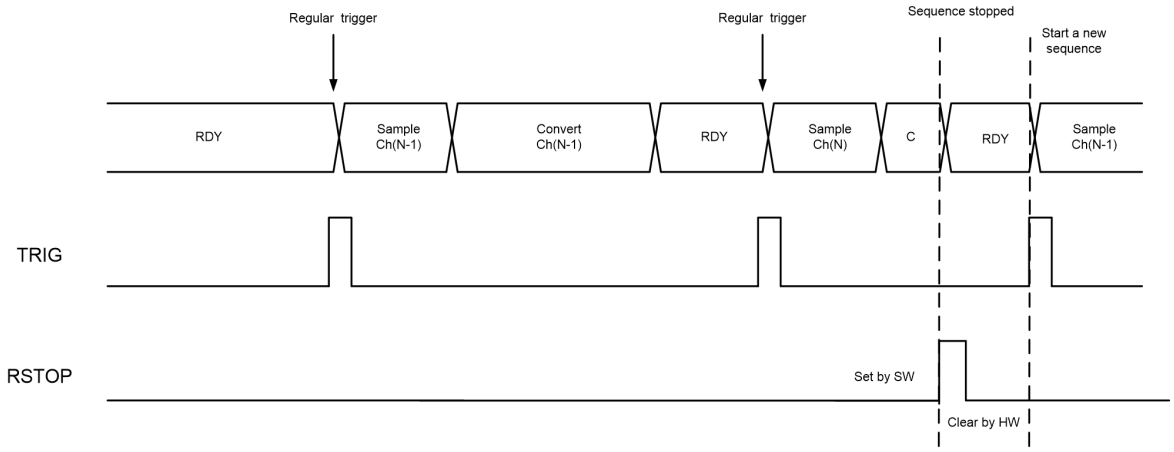


图 37-12 常规组转换 STOP

下图示例显示了常规组和注入组分别 STOP 的情况：常规组由软件触发后连续转换，其中插入软件触发的注入转换，注入转换被 JSTOP 终止，状态机回到常规组，随后常规组被 RSTOP 终止。

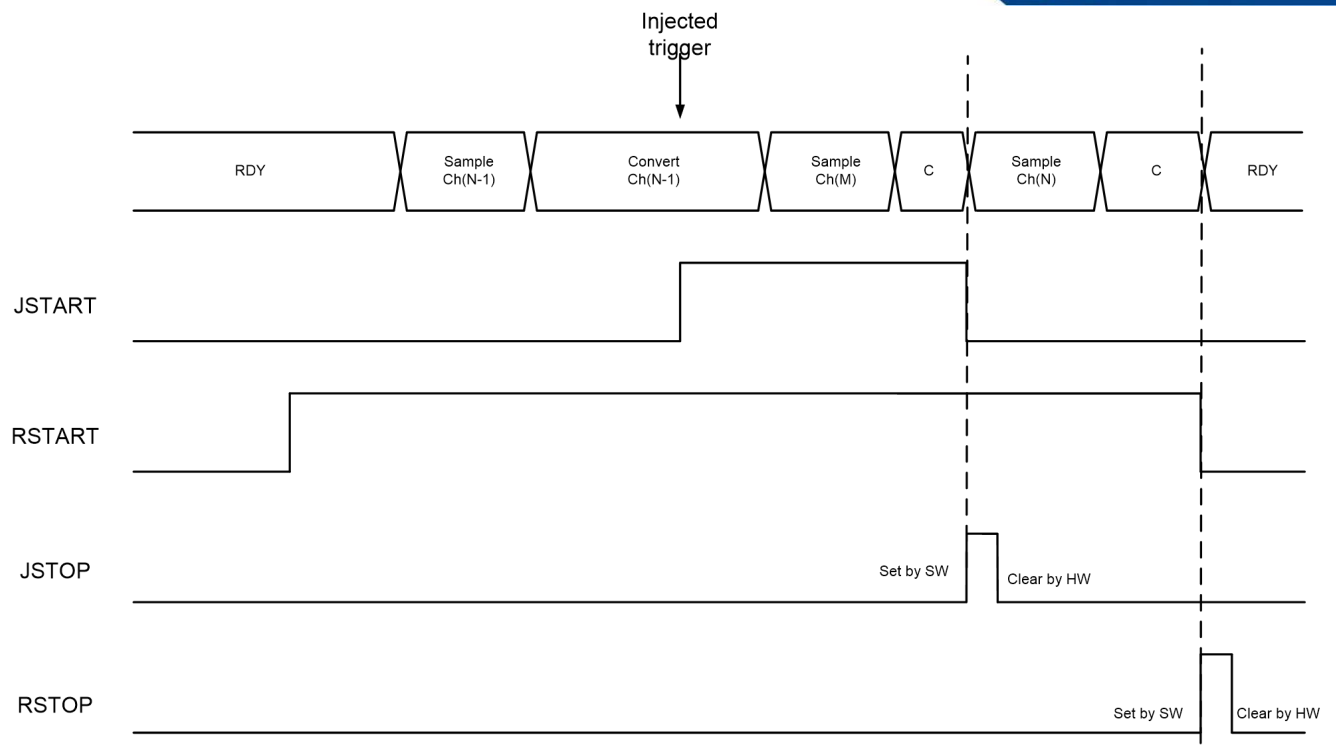


图 37-13 常规组和注入组转换 STOP

37.6.8 过采样和硬件平均

ADC 支持硬件过采样平均，可以在一定程度上提高分辨率。原理是对于低速输入信号，可以通过连续多次采样后求平均的方法提高 ENOB，过采样公式如下：

$$result = \frac{\sum_{n=1}^{N} CONVERSION_n}{M}$$

其中 N 是过采样倍数，可配置为 2/4/8/16/32/64/128/256，M 为结果右移位数，最大右移 8bit；由于每次转换结果为 12bit，最大 256 次累加得到的结果为 20bit，经过移位后可以得到 12~16bit 最终结果。ADC 输出结果最多只有 16bit，如果右移后结果超过 16bit，高位也会被丢弃。

在使能过采样的情况下，EOC 信号在 N 次连续采样后才置位，对于应用程序和 DMA 来说，感觉就好像只经过一次采样转换。

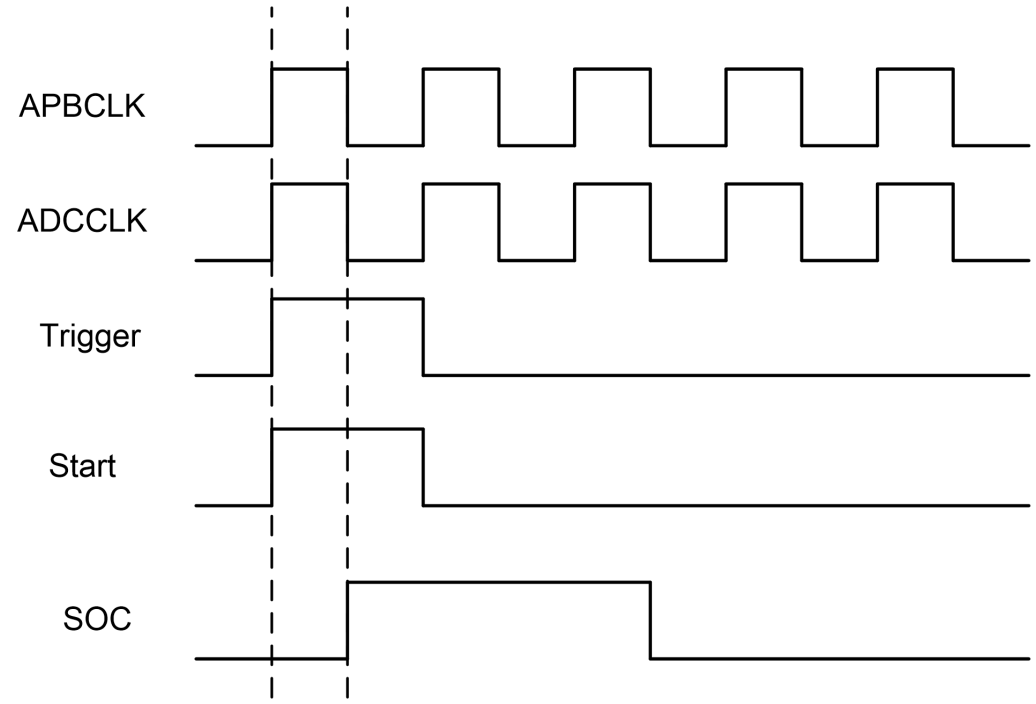
37.6.9 ADC 工作时钟

ADC 模块采用双时钟工作，其中 APBCLK 用于总线寄存器访问，并且也可以作为 ADC 工作时钟使用；ADCCLK 为 ADC 工作时钟，可以是 APBCLK 或者来自 CMU 的独立时钟。

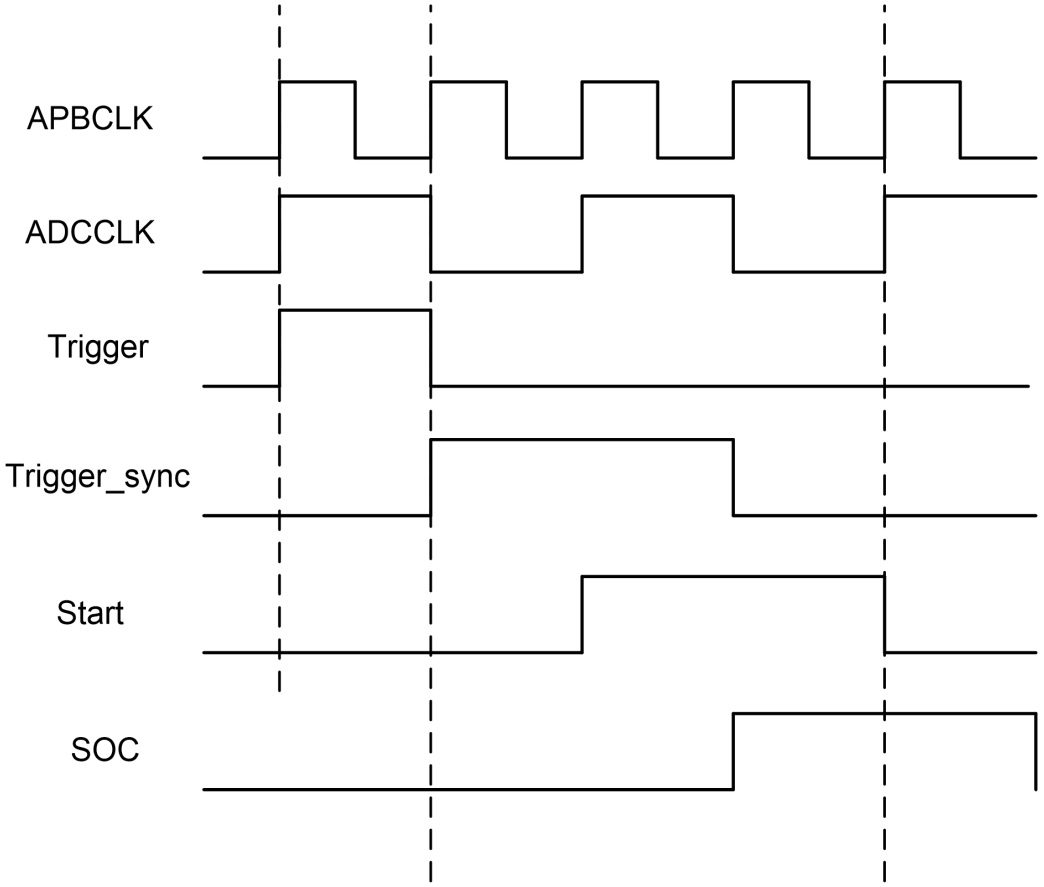
当 ADC 的采样转换工作在 APBCLK 下时，所有 APBCLK 下外设所产生的触发事件都是同步于 ADCCLK 的，则采样启动时间相对于触发事件的延迟是确定的，可用于时延敏感型应用场合。

注意，并不是所有触发源都支持确定延迟触发，只有触发信号输出为 APBCLK 时钟域的触发源才支持这一特性。参见“转换触发”章节。当 ADCCLK 选择为 APBCLK 分频时，触发事件到来与 ADC 采样启动之间的时间间隔在一定范围内有不确定性。

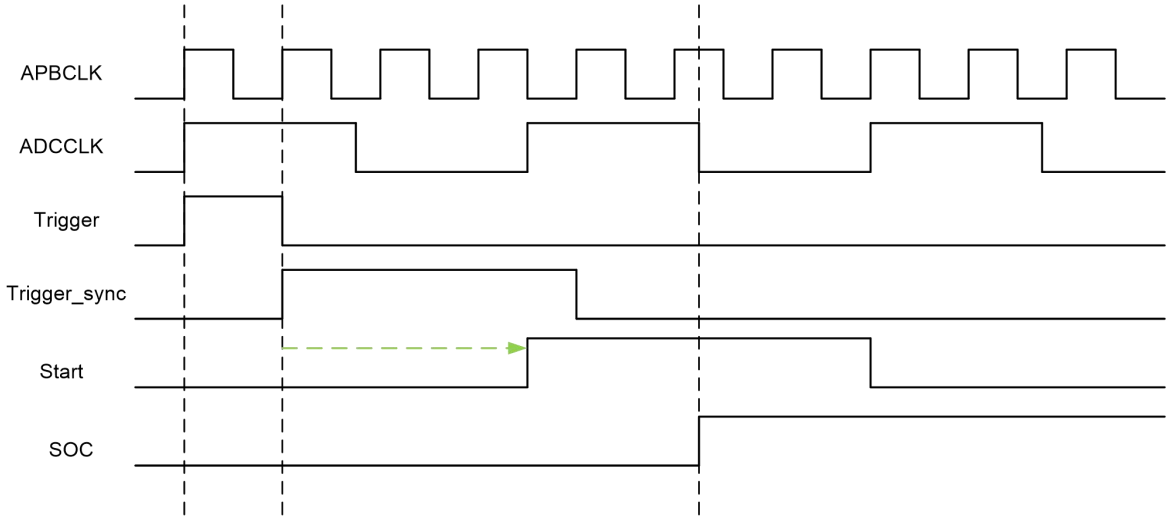
举例来说，下图显示了 ADCCLK=APBCLK 的情况（同步触发），触发延迟是固定的 2 个 APBCLK。



下图显示了 ADCCLK=APBCLK/2 的情况（作为异步时钟处理），触发延迟可能是 3 个或者 4 个 APBCLK。



下图显示了当 ADCCLK 完全异步于 APBCLK 时的触发时序。



当 ADC 的采样转换工作在独立时钟下时，ADC 工作频率可以与系统时钟频率解耦，比如 CPU 和总线可以工作在较低频率下，同时保持 ADC 工作在较高采样率，实现更高的应用灵活性。但是这种情况下，触发事件与 ADC 采样启动之间存在一个跨时钟域的重新同步延迟，这个延迟的大小与当前 APBCLK 和 ADCCLK 之间的频率及相位关系相关，有一定的变化范围。

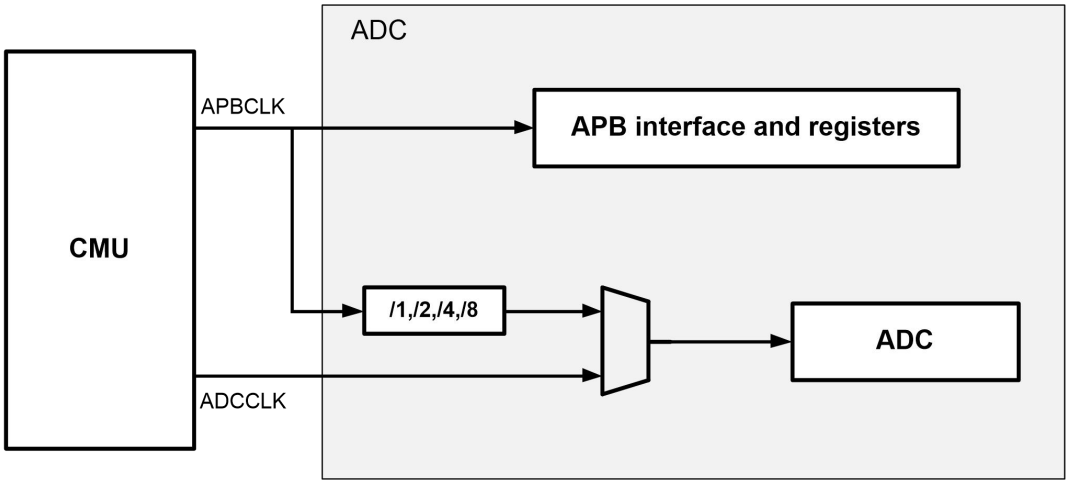


图 37-14 ADC 时钟示意图

37.6.10 数据冲突和自动等待

每次转换完成后 EOC 标志会置位，软件或 DMA 读取 ADC_DATA 寄存器后会自动清除 EOC，也可以由软件写 1 清除。当 EOC 标志没有被清除的情况下，新的转换数据到来，就会导致 data overrun；有两种 overrun 模式：

OVRMOD=0：保持旧的数据，新数据丢弃

OVRMOD=1：写数据写入覆盖旧数据

当使用 DMA 时出现 overrun，则不会发起新的 DMA request，直到 OVR 标志被软件清零

ADC 控制器还支持自动等待，如果 WAIT 寄存器被软件置位，那么在 ADC_DATA 寄存器被读取之前，ADC 控制器不会发起新的转换；在等待状态中到来的硬件触发事件也会被忽略。

下图是软件触发连续模式的情况下，使能了自动等待的示意图：

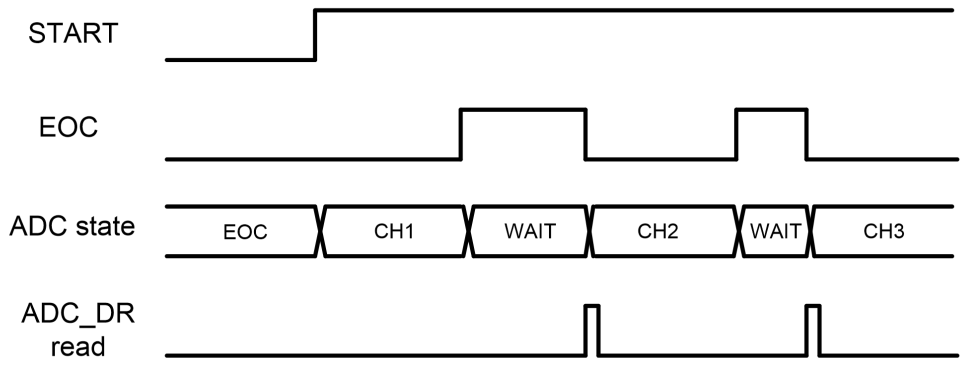


图 37-15 ADC 自动等待

注意：DMA 循环模式下请保持 WAIT=0

37.6.11 DMA

在多通道转换或连续转换时，使用 DMA 进行转换结果搬移是高效的解决方案。在使能了 DMAEN 的情况下，当每次转换完成后（EOC），ADC controller 模块会产生一个 DMA 请求，通知 DMA 将数据寄存器中的结果搬运到指定的 SRAM 地址。ADC 转换由触发事件启动，触发可以来自于软件触发或硬件触发。DMA 模式下支持一次触发后连续执行多次转换（自动模式），或者每次触发执行一次转换的方式（半自动模式）。

在自动模式下（ADC_CFGR2.SEMI=0）和半自动模式下（ADC_CFGR2.SEMI=1），ADC 的 DMA 接口都可以支持单次模式和循环模式：

单次模式

转换完成后发起数据搬运，此过程会一直重复，直到软件配置的 DMA 传输长度完成，然后 ADC 控制器会自动停止转换（通过接收 DMA 的传输完成中断标志信号），关闭 ADC，不再向 DMA 发起请求。此模式主要用于对特定模拟信号进行一定长度的采样。

循环模式

与 DMA 的循环模式相配合，ADC 不断循环转换并发起 DMA 请求，直到软件停止转换。此模式可以用于处理连续不断模拟信号采样。

通过 ADC_CFGR2.SEMI 寄存器可以配置 DMA 模式下的自动或半自动转换，通过 DMA 模块的 CHxCIRC 配置循环模式或单次模式，通过 DMA 模块的 CHxTSIZE 配置传输长度。寄存器配置组合实现的效果如下表所示。

ADC_CFGR2.CONT	ADC_CFGR2.SEMI	DMA CHxCIRC	DMA CHxTSIZE	说明
0	0	0	N	单次全自动模式，触发事件到来后，依次对使能输入通道转换 N 次并将数据搬运到 SRAM；所有使能输入通道都被采样过，ADC 自动停止并等待下一次触发；如果数据搬运次数达到 N，DMA 自动关闭。
0	0	1	N	单次全自动模式（循环存储），触发事件到来后，依次对使能输入通道不断转换并搬运数据，直到所有使能输入通道都采样过，ADC 自动停止并等待下一次触发；RAM 数据空间长度为 N，当搬运数据长度超过 N 之后回到 DMA 通道指针指向的

				起始地址，并覆盖原数据。
0	1	0	N	单次半自动模式，每个触发事件启动一次转换，依次采样所有被使能的输入通道；数据搬运次数达到 N，DMA 自动关闭。
0	1	1	N	单次半自动模式（循环存储），每个触发事件启动一次转换，依次采样所有被使能的输入通道；RAM 数据空间长度为 N，当搬运数据长度超过 N 之后回到 DMA 通道指针指向的起始地址，并覆盖原数据。
1	x	0	N	连续模式，触发后 ADC 不断采样使能通道，DMA 不断搬运数据，直到完成 N 次数据搬运，DMA 自动关闭。
1	x	1	N	连续模式（循环存储），触发后 ADC 不断采样使能通道，DMA 不断搬运数据；RAM 数据空间长度为 N，当搬运数据长度超过 N 之后回到起始地址覆盖原数据；直到软件关闭 DMA。

表 37-4 DMA 配置与功能

在 DMA 使能情况下，如果发生 overrun，则 ADC 控制器不再发送 DMA 请求，直到 OVR 标志被清除。

注意，在单次和连续转换模式下，都可以支持 DMA 传输；DMA 传输长度以 EOC 的次数定义，而不是 EOS，即 DMA 只关心搬运多少次 ADC_DATA。
DMA 循环模式下应保持 WAIT=0

注意，使能 DMA 后，注入组转换也会触发 DMA 搬运，DMA 无法区分常规组数据和注入组数据。为了避免 RAM 中数据混淆，一般不建议使用注入通道时使能 DMA 功能。

37.6.11.1 DMA 使用案例

使用案例 1：
ADC 配置为单次全自动模式，使能 3 个 ADC 输入通道，DMA TSIZE 配置为 6。第一次触发后，ADC 依次采样 3 个通道，DMA 搬运 3 次数据，然后 ADC 和 DMA 暂停工作，等待第二次触发。第二次触发后，ADC 依次采样 3 个通道，DMA 搬运 3 次数据。6 次 DMA 传输完成，DMA 通道关闭。

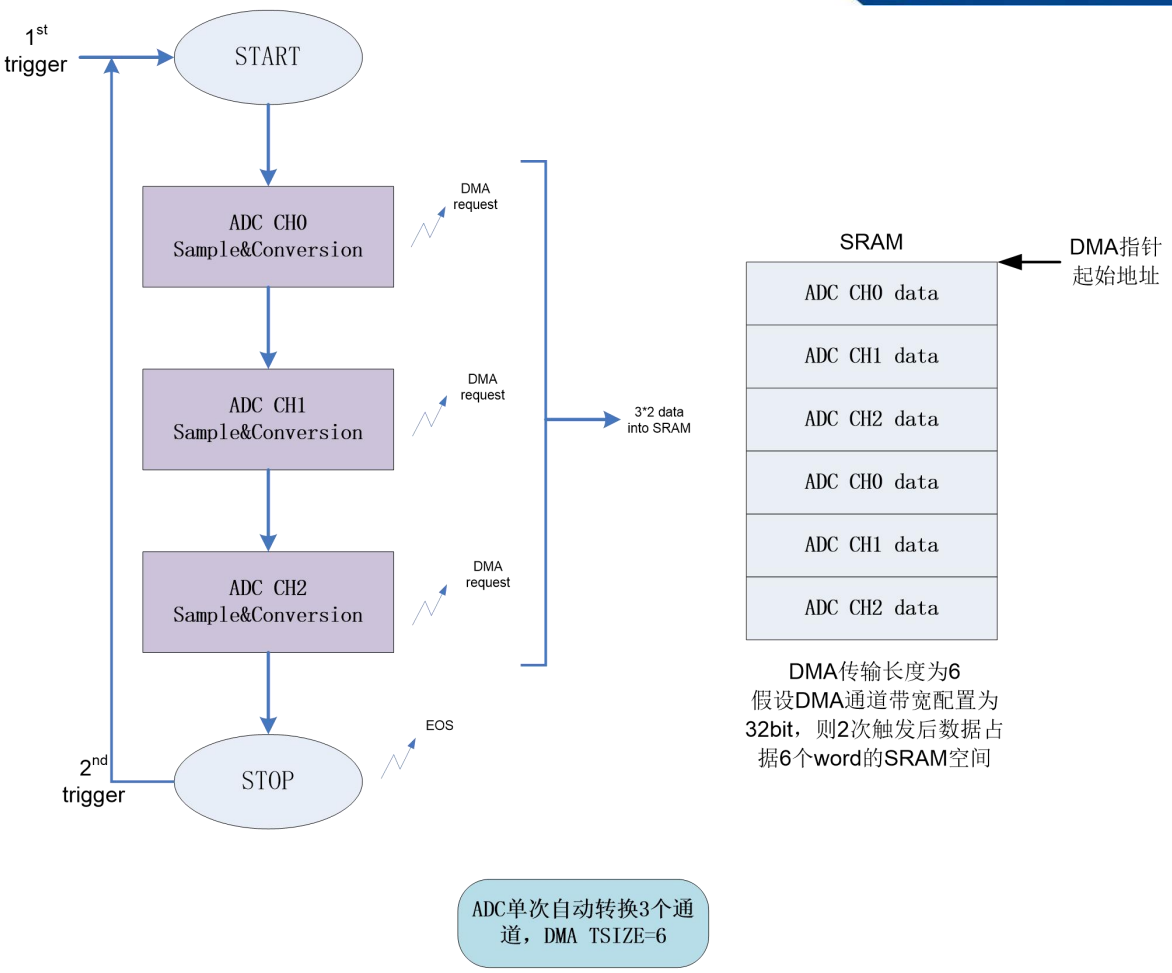


图 37-16 ADC 单次全自动触发+DMA 案例 1

使用案例 2：

ADC 配置为单次全自动模式，使能 3 个 ADC 输入通道，DMA TSIZE 配置为 2。第一次触发后，ADC 依次采样 3 个通道，DMA 搬运 2 次数据后 DMA 和 ADC 自动关闭，不会对第三个输入通道进行采样。

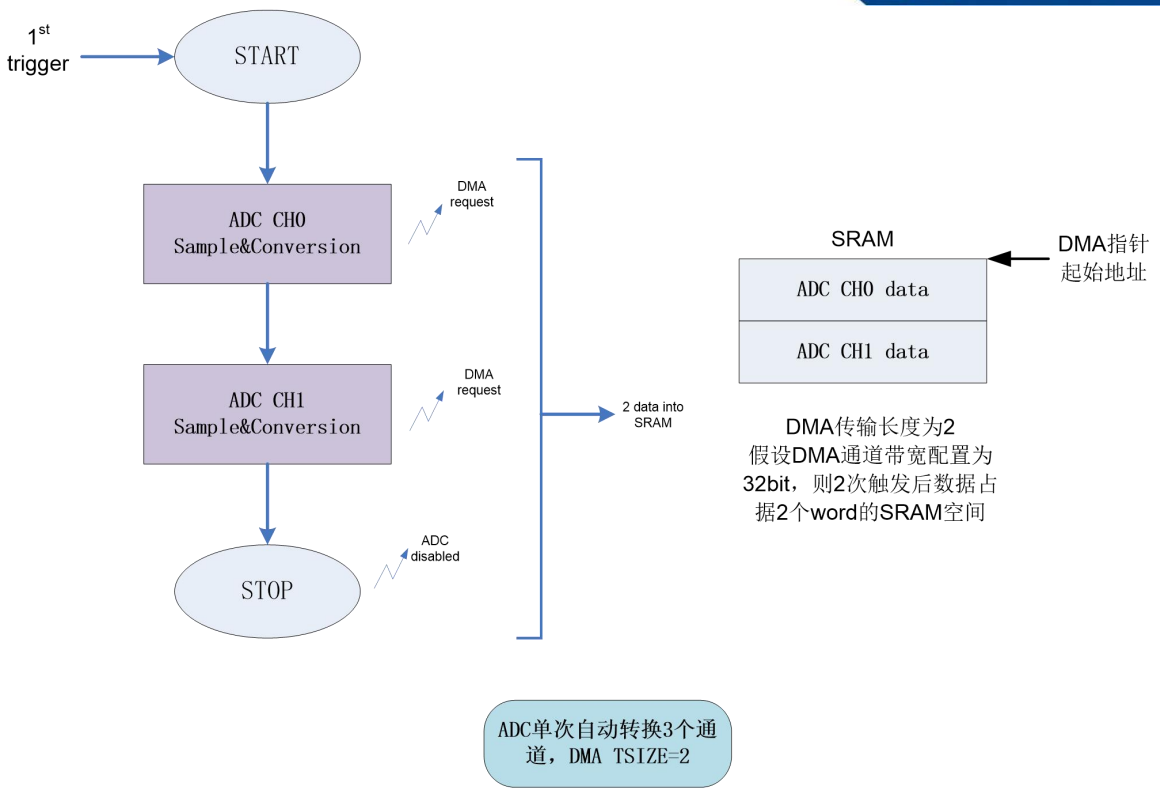


图 37-17 ADC 单次全自动触发+DMA 案例 2

使用案例 3：

ADC 配置为单次半自动模式，使能 3 个 ADC 输入通道，DMA TSIZE 配置为 6。每次触发后，DMA 搬运一个转换数据，总共 6 个数据搬运完毕后，自动关闭 ADC 和 DMA。

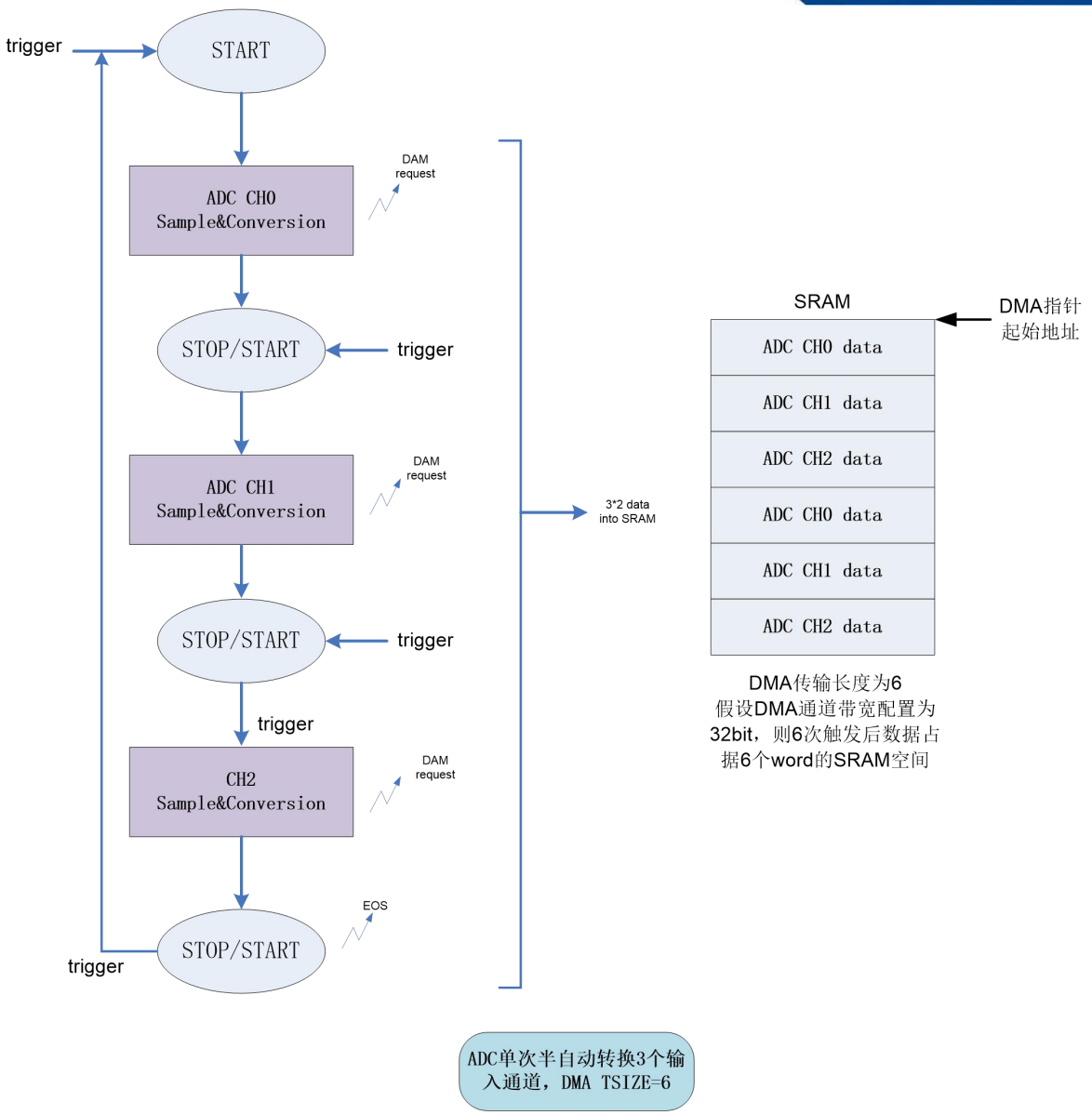


图 37-18 ADC 单次半自动触发+DMA

使用案例 4：

ADC 配置为全自动模式（SEMI=0），使能 3 个 ADC 输入通道（ADC_IN0/1/2），DMA TSIZE 配置为 6，DMA 指针指向 RAM 地址 x，DMA 配置为循环模式（CIRC=1）。第一次触发后，ADC 执行 3 次采样转换，DMA 搬运 3 次数据到 RAM 的 x/x+1/x+2 地址；第二次触发后，ADC 执行 3 次采样，DMA 搬运 3 次数据到 RAM 的 x+3/x+4/x+5 地址；第三次触发后，ADC 执行 3 次采样，DMA 搬运 3 个数据到 RAM 的 x/x+1/x+2 地址，循环往复直到软件关闭 DMA。

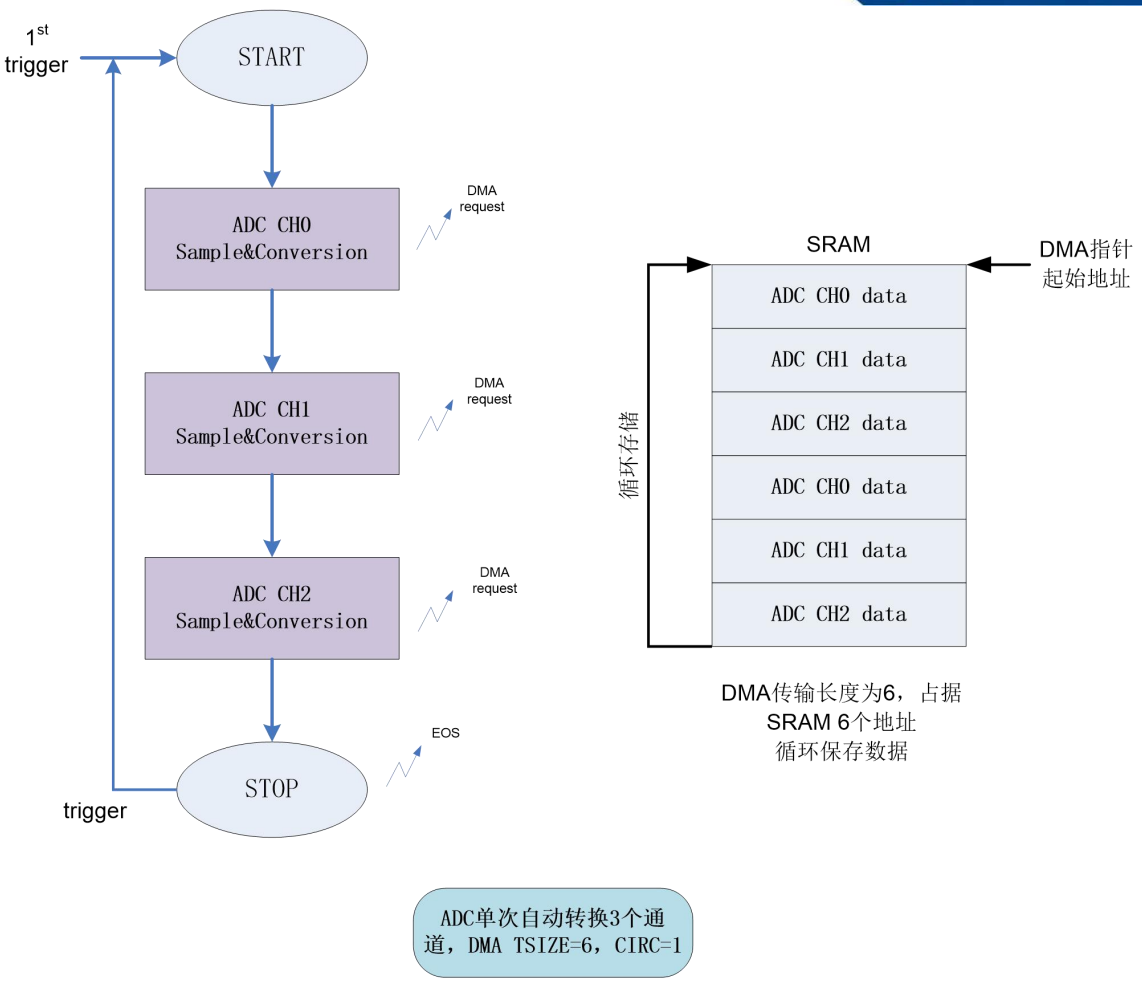


图 37-19 ADC 全自动触发+DMA 循环模式

使用案例 5：

ADC 配置为连续模式（CONT=1），使能 3 个 ADC 输入通道（ADC_IN0/1/2），DMA TSIZE 配置为 6，DMA 指针指向 RAM 地址 x，DMA 配置为循环模式（CIRC=1）。第一次触发后，ADC 对使能通道连续不间断采样转换，DMA 连续搬运数据到 RAM 的 x/x+1/x+2/x+3/x+4/x+5 地址；循环往复直到软件关闭 DMA。

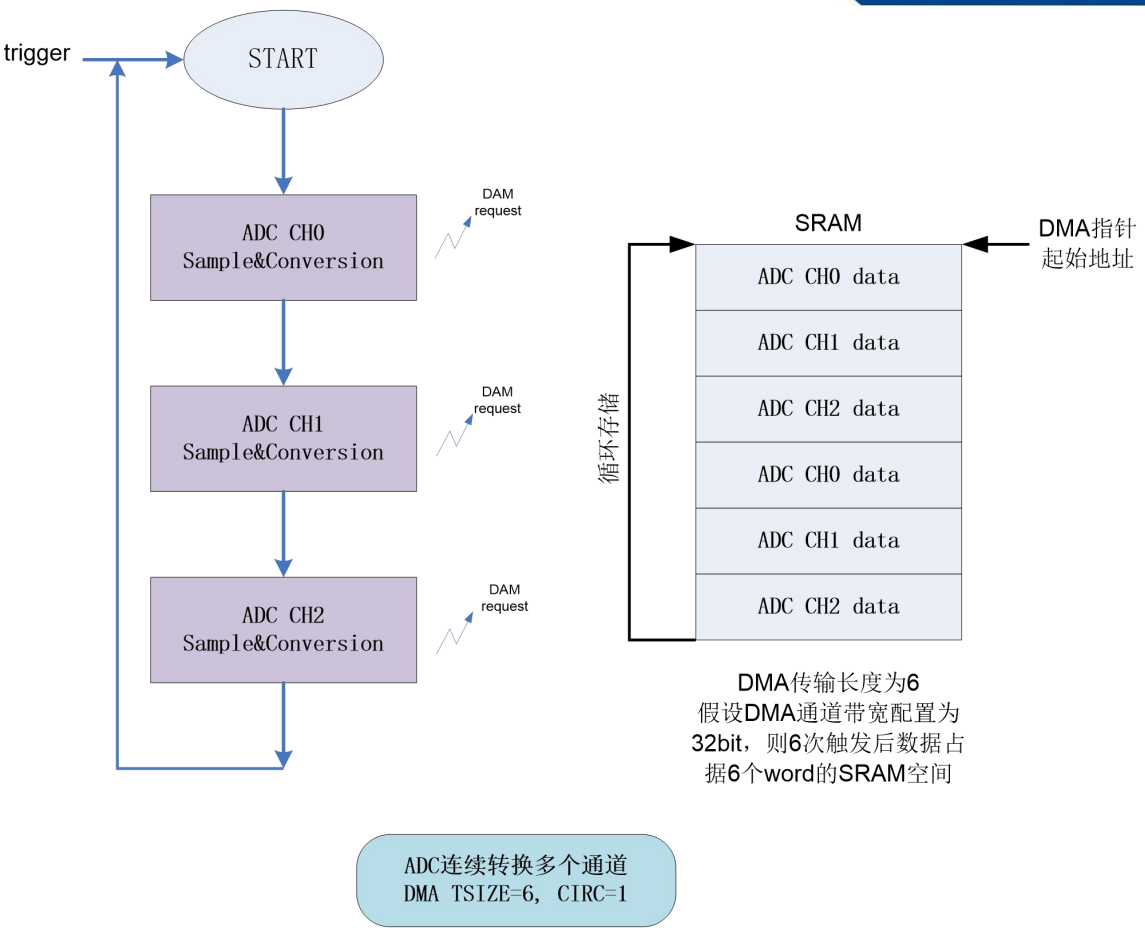


图 37-20 ADC 连续模式+DMA 循环模式

37.6.12 模拟窗口看门狗（AWD）

AWD 功能用于监视某个模拟输入通道或所有输入通道的输入信号电平是否处于寄存器设置的幅值范围之内。当 ADC 转换值高于 AWD_HT 或者低于 AWD_LT 时，都会置位中断标志寄存器。标志寄存器由软件写 1 清零。

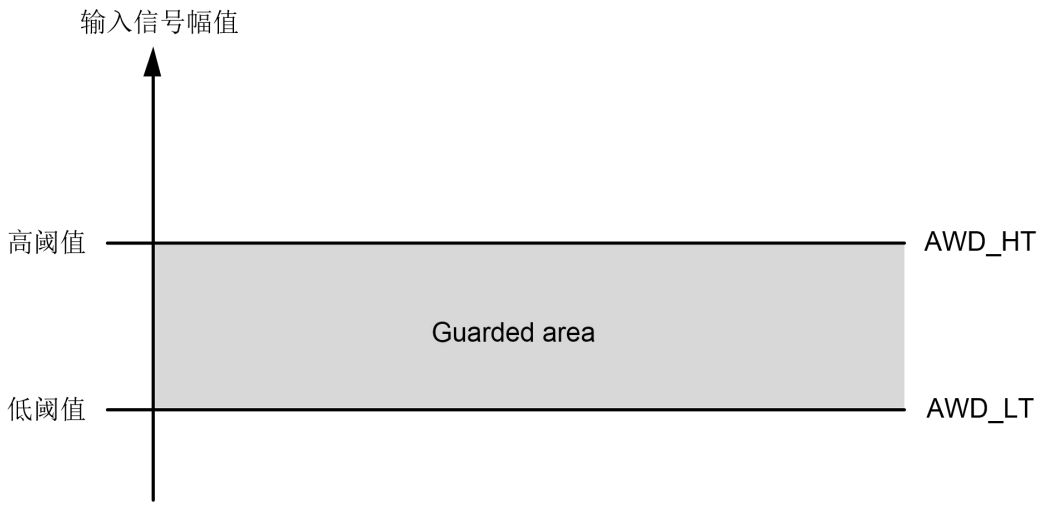


图 37-21 ADC 模拟看门狗阈值示意图

通过 AWDEN 寄存器使能模拟窗口看门狗功能，通过 AWDSC 寄存器配置单通道监视或全部通道监视。

AWD 阈值设置寄存器的位宽是 16bit。

37.6.13 ADC 校准

ADC 支持 offset 自校准，建议在芯片上电后，首先进行一次校准操作，以获得更好的精度。

Offset 校准包含两部分：比较器校准和通道（单端）校准，分别用于消除比较器 offset 和采样通道的 offset。两种校准由寄存器分别控制和使能，推荐用户先进行比较器校准，再进行通道（单端）校准（如果需要）。

当 CALSEL=0 时，软件置位 CALEN 启动比较器校准，硬件完成 ADC 自校准操作后会自动清零 CALEN 寄存器，校准操作结束后 EOCAL 中断标志置位，如果中断使能 EOCALIE 为 1，则产生中断通知 CPU。

注：ADC 复位后，在第一次转换之前，建议启动一次校准操作。校准完成后，重新使能 ADC 不需要再次进行校准即可进行转换。因此建议用户在芯片上电复位完成后先进行一次校准操作。当 ADC 工作环境（温度、电压）发生显著变化时，也推荐进行校准以获得更高精度。

进行校准操作后，校准参数被保存在 ADC 内部寄存器，ADC 模块复位或 ADC 控制器复位都不会清除校准寄存器内容，只有上下电复位会清除校准寄存器。

37.7 低功耗模式

在低功耗模式下，芯片自动关闭了所有高速时钟源，所以 ADC 无法工作。

37.8 寄存器

模块起始地址：0x4001_5C00

offset 地址	名称	符号
0x00	ADC 中断和状态寄存器 （ADC Interrupt and Status Register）	ADC_ISR
0x04	ADC 中断使能寄存器 （ADC Interrupt Enable Register）	ADC_IER
0x08	ADC 控制寄存器 1 （ADC Control Register1）	ADC_CR1
0x0C	ADC 控制寄存器 2 （ADC Control Register2）	ADC_CR2
0x10	ADC 校准控制寄存器 （ADC Calibration Register）	ADC_CALR
0x14	ADC 配置寄存器 1 （ADC Config Register1）	ADC_CFGR1
0x18	ADC 配置寄存器 2 （ADC Config Register2）	ADC_CFGR2
0x1C	ADC 采样时间控制寄存器 （ADC Sampling Time Register）	ADC_SMTR
0x20	ADC 外部通道控制寄存器 （ADC ExternalChannel Enable Register）	ADC_ECHER
0x24	ADC 内部通道控制寄存器 （ADCInternal Channel Control Register）	ADC_ICHER
0x28	ADC 常规组数据寄存器 （ADC Data Register）	ADC_DR
0x2C	ADC 模拟看门狗阈值寄存器 （ADC analog watchdog Threshold Register）	ADC_HLTR
0x30	ADC 注入组全局配置寄存器 （ADC Injection Global Config Register）	ADC_JGCFGR
0x34	ADC 注入组配置寄存器 （ADC Injection Group Config Register）	ADC_JCFGR
0x38	ADC 注入组数据寄存器 0 （ADC Injection Group Data Register0）	ADC_JDR0
0x3C	ADC 注入组数据寄存器 1 （ADC Injection Group Data Register1）	ADC_JDR1
0x40	ADC 注入组数据寄存器 2 （ADC Injection Group Data Register2）	ADC_JDR2
0x44	ADC 注入组数据寄存器 3 （ADC Injection Group Data Register3）	ADC_JDR3

0x4C	ADC 原始数据寄存器 （ADC Raw Data Register）	ADC_RAWR

37.8.1 ADC 中断和状态寄存器（ADC_ISR）

名称	ADC_ISR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	AWD_AH	AWD_UL	EOCAL	BUSY	OVR	EOS	EOC
位权限	U-0	R/W/Dy-0	R/W/Dy-0	R/W/Dy-0	R-0	R/W/Dy-0	R/W/Dy-0	R/W/Dy-0

Bit	助记符	功能描述
31:7	--	RFU：未实现，读为 0
6	AWD_AH	模拟看门狗超出上限中断标志 (Analog Watchdog Above High)，当采样值高于 AWD_HT 时，硬件置位，软件写 1 清零
5	AWD_UL	模拟看门狗低于下限中断标志 (Analog Watchdog Under Low)，当采样值低于 AWD_LT 时，硬件置位，软件写 1 清零
4	EOCAL	校准结束（End Of Calibration），硬件置位，软件写 1 清零 1：校准过程结束 0：无校准过程
3	BUSY	ADC 忙标志 (ADC busy) 1：ADC 正在校准、采样或转换过程中 0：ADC 空闲
2	OVR	数据冲突标志，硬件置位，软件写 1 清零。(Over-Run) 当 ADC_DATA 寄存器中的上一次转换结果还未被读取，新的转换结果又到来时，硬件置位 OVR 标志。 0：没有数据冲突 1：出现数据冲突
1	EOS	转换序列结束（End Of Sequence） 所有使能通道都转换完成后，置位 EOS，软件写 1 清零。
0	EOC	单次转换结束（End Of Conversion） 每个通道转换完成后，置位 EOC，软件写 1 清零。

37.8.2 ADC 中断使能寄存器（ ADC_IER）

名称	ADC_IER							
Offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	AWD_AHIE	AWD_ULIE	EOCALIE	-	OVRIE	EOSIE	EOCIE
位权限	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:7	--	RFU：未实现，读为 0
6	AWD_AHIE	模拟看门狗采样值高于上限中断使能，1 有效
5	AWD_ULIE	模拟看门狗采样值低于上限中断使能，1 有效
4	EOCALIE	校准结束中断使能寄存器 0：禁止 EOCAL 中断 1：允许 EOCAL 中断
3	--	RFU：未实现，读为 0
2	OVRIE	数据冲突中断使能寄存器 0：禁止数据冲突中断 1：允许数据冲突中断
1	EOSIE	转换序列结束（End Of Sequence）中断使能寄存器 0：禁止 EOS 中断 1：允许 EOS 中断
0	EOCIE	单次转换结束（End Of Conversion）中断使能寄存器 0：禁止 EOC 中断 1：允许 EOC 中断

37.8.3 ADC 控制寄存器 1（ADC_CR1）

名称	ADC_CR1							
Offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			-	RSTOP	-	RSTART	ADEN
位权限	U-0			U-0	R/W-0	U-0	R/W-0	R/W-0

Bit	助记符	功能描述

Bit	助记符	功能描述
31:5	--	RFU：未实现，读为 0
4	--	RFU：未实现，读为 0
3	RSTOP	ADC 停止转换寄存器，软件写 1 停止，硬件自动清零
2	--	RFU：未实现，读为 0
1	RSTART	ADC 启动转换寄存器（软件触发），软件写 1 启动，转换结束（EOC）或者软件置位 RSTOP 时，硬件自动清零。
0	ADEN	ADC 使能寄存器。在启动转换前先要置位 ADEN 0：关闭 ADC 1：使能 ADC

37.8.4 ADC 控制寄存器 2（ADC_CR2）

名称	ADC_CR2							
Offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				-		RTRGCFG	
位权限	U-0				U-0		R/W-00	

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:2	--	RFU：未实现，读为 0
1:0	RTRGCFG	触发信号使能和极性选择。(Regular Group Trigger Config)。 00：软件触发 01：硬件上升沿触发 10：硬件下降沿触发 11：硬件上升、下降沿都触发

37.8.5 ADC 校准控制寄存器（ADC_CALR）

名称	ADC_CALR							
Offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	OSCAL_CYCLE							
位权限	R/W-0000 0111							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			RFUI				
位权限	U-0			R/W-0				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	VCM_CTRL	VCM_MODE	CMPRDY_DELAY		CMP_MODE	CKDIG_DELAY		CALEN
位权限	R/W-0	R/W-0	R/W-00		R/W-0	R/W-00		R/W-0

位号	助记符	功能描述
31:24	-	RFU：未实现，读为 0
23:16	OSCAL_CYCLE	Offset 校准周期数配置
15:13	-	RFU：未实现，读为 0
12	RFUI	保留位
11:8	-	RFU：未实现，读为 0
7	VCM_CTRL	共模电流配置
6	VCM_MODE	VCM 控制模式 0：正常 1：长通
5:4	CMPRDY_DELAY	比较器延迟控制 00：延迟最小 11：验延迟最大
3	CMP_MODE	比较器模式选择 0：正常模式 1：快速模式
2:1	CKDIG_DELAY	数字时钟延迟选择 00：延迟最小 11：延迟最大
0	CALEN	Offset Calibration 使能 (Offset Calibration Enable) 软件写 1 启动校准周期，校准结束后自动清零并置位 EOCAL 寄存器。 Offset 校准分为比较器校准和通道校准，由 CALSEL 寄存器选择校准项目。两种校准都是由 CALEN 寄存器启动的。 *注意：CALEN 软件无法清零，只能硬件自动清零或者通过模块复位清零*

37.8.6 ADC 配置寄存器 1（ADC_CFGR1）

名称	ADC_CFGR1							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							REFNSEL
位权限	U-0							R/W-0

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	APBCLK_PSC		-	RFUI		CLKSEL	-	REFPSEL
位权限	R/W-00		U-0	R/W-00		R/W-0	U-0	R/W-0

Bit	助记符	功能描述
31:11	--	RFU：未实现，读为 0
8	REFNSEL	ADC 负基准选择(Negative Reference Select) 0：VSSA 1：VREFN
7:6	APBCLK_PSC	APBCLK 用作 ADC 工作时钟前的预分频 (APBCLK prescaler) 00：不分频 01：2 分频 10：4 分频 11：8 分频
5	--	RFU：未实现，读为 0
4:3	RFUI	RFUI：保留位，可读写
2	CLKSEL	ADC 工作时钟选择 (Clock Select) 0：使用 ADCCLK 工作 1：使用 APBCLK 工作
1	--	RFU：未实现，读为 0
0	REFPSEL	ADC 正基准选择 (Positive Reference Select) 0：使用 VDDA 作为基准 1：使用 VREFP 作为基准 *注意：当使用 VREFP 和 VREFN 引脚作为输入通道时，必须将 ADC 基准源选择为 VDDA*

37.8.7 ADC 配置寄存器 2（ADC_CFGR2）

名称	ADC_CFGR2							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	AWDCH					AWDSC	AWDEN
位权限	U-0	R/W-00000					R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	OVSS				OVSR			OVSEN
位权限	R/W-0000				R/W-000			R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	IOTRFEN	-		SEMI	WAIT	CONT	OVRM
位权限	U-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTS				-	SCANDIR	-	DMAEN
位权限	R/W-0000				U-0	R/W-0	U-0	R/W-0

Bit	助记符	功能描述
31	--	RFU：未实现，读为 0
30:26	AWDCH	模拟窗口看门狗监视通道选择，仅在 AWDSC=1 时有效

Bit	助记符	功能描述
		00000：AWD 监视 ADC_IN0 00001：AWD 监视 ADC_IN1 00010：AWD 监视 ADC_IN2 00011：AWD 监视 ADC_IN3 00100：AWD 监视 ADC_IN4 00101：AWD 监视 ADC_IN5 00110：AWD 监视 ADC_IN6 00111：AWD 监视 ADC_IN7 01000：AWD 监视 ADC_IN8 01001：AWD 监视 ADC_IN9 01010：AWD 监视 ADC_IN10 01011：AWD 监视 ADC_IN11 01100：AWD 监视 ADC_IN12 01101：AWD 监视 ADC_IN13 01110：AWD 监视 ADC_IN14 01111：AWD 监视 ADC_IN15 10000：AWD 监视 ADC_IN16 10001：AWD 监视 ADC_IN17 10010：AWD 监视 ADC_IN18 10011：AWD 监视 ADC_IN19 10100：AWD 监视 ADC_IN20 10101：AWD 监视 ADC_IN21 10110：AWD 监视 ADC_IN22 10111：AWD 监视 ADC_IN23 11000：AWD 监视 ADC_IN24 11001：AWD 监视 ADC_IN25 11010：AWD 监视 ADC_IN26 11011：AWD 监视 ADC_IN27 11100：AWD 监视 ADC_IN28 11101：AWD 监视 ADC_IN29 11110：AWD 监视 ADC_IN30 11111：RFU
25	AWDSC	模拟窗口看门狗单通道或全通道选择 0：AWD 监视所有被使能的外部输入通道 1：AWD 监视 AWDCH 指定的单个通道
24	AWDEN	模拟窗口看门狗使能寄存器 0：关闭 AWD 1：使能 AWD 仅能在 START=0 的情况下配置此寄存器
23:20	OVSS	过采样移位控制寄存器 0000：不移位 0001：右移 1bit 0010：右移 2bit 0011：右移 3bit 0100：右移 4bit 0101：右移 5bit 0110：右移 6bit 0111：右移 7bit 1000：右移 8bit

Bit	助记符	功能描述
		Others：RFU
19:17	OVSR	过采样率控制 000：2x 001：4x 010：8x 011：16x 100：32x 101：64x 110：128x 111：256x
16	OVSEN	过采样使能 0：禁止过采样 1：使能过采样
15	--	RFU：未实现，读为 0
14	IOTRFEN	引脚触发信号数字滤波使能 0：禁止数字滤波 1：使能数字滤波
13:12	--	RFU：未实现，读为 0
11	SEMI	单次转换半自动模式，仅在单次转换（CONT=0）时有效，参见“转换模式”章节 0：自动模式 1：半自动模式
10	WAIT	等待模式控制 0:无等待,如果上次转换数据没有及时读取,则可能出现 Overrun 1：等待模式，在上次转换数据被读取前，不会启动下一次转换
9	CONT	连续转换模式使能 0：单次转换 1：连续转换
8	OVRM	Overrun 模式控制 0：当 overrun 发生时，保持上次数据，丢弃本次转换值 1：当 overrun 发生时，覆盖上次数据
7:4	EXTS	硬件触发源选择 0000：ATIM_TRGO 0001：TAU10_TRGO 0010：TAU11_TRGO 0011：TAU12_TRGO 0100：GPTIM1_TRGO 0101：GPTIM2_TRGO 0110：BSTIM16_TRGO 0111：LPTIM16_TRGO 1000：COMP1_TRGO 1001：COMP2_TRGO 1010：RTCA_TRGO 1011：TAU00_TRGO 1100：TAU01_TRGO 1101：TAU02_TRGO 1110：TAU03_TRGO 1111：LUT0_TRGO *注意：必须在 TRGCFG=00 的情况下修改 EXTS 寄存器*

Bit	助记符	功能描述
3	--	RFU：未实现，读为 0
2	SCANDIR	外部通道扫描顺序控制（共 26 个通道，实际只会采样被使能的通道） 0：前向扫描，ADC_IN0->ADC_IN18->内部通道 1：反向扫描，内部通道->ADC_IN18->ADC_IN0
1	--	RFU：未实现，读为 0
0	DMAEN	DMA 使能 0：禁止 DMA 1：使能 DMA

37.8.8 ADC 采样时间控制寄存器（ADC_SMTR）

名称	ADC_SMTR							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SMTS2				SMTS1			
位权限	R/W-0000				R/W-0000			

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:4	SMTS2	快速采样时间控制 2（*ADC 工作时钟周期），适用于所有快速通道

SMTSx	Sampling cycles
0000	2
0001	4
0010	8
0011	12
0100	16
0101	32
0110	64
0111	80
1000	96
1001	128
1010	160
1011	192
1100	256
1101	320
1110	384

Bit	助记符	功能描述
		1111 512
3:0	SMTS1	慢速采样时间控制 1（*ADC 工作时钟周期），适用于所有慢速通道 SMTSx / Sampling cycles 0000 2 0001 4 0010 8 0011 12 0100 16 0101 32 0110 64 0111 80 1000 96 1001 128 1010 160 1011 192 1100 256 1101 320 1110 384 1111 512

37.8.9 ADC 通道控制寄存器（ADC_CHER）

名称	ADC_ECHER							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	ECH30	ECH29	ECH28	ECH27	ECH26	ECH25	ECH24
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ECH23	ECH22	ECH21	ECH20	ECH19	ECH18	ECH17	ECH16
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ECH15	ECH14	ECH13	ECH12	ECH11	ECH10	ECH9	ECH8
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1	ECH0
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31	--	RFU：未实现，读为 0
30	ECH30	ADC_IN0~30 测量通道，写 1 使能 其中 ADC_IN16 和 ADC_IN15 与 VREFP 和 VREFN 引脚复用
29	ECH29	

Bit	助记符	功能描述
28	ECH28	
27	ECH27	
26	ECH26	
25	ECH25	
24	ECH24	
23	ECH23	
22	ECH22	
21	ECH21	
20	ECH20	
19	ECH19	
18	ECH18	
17	ECH17	
16	ECH16	
15	ECH15	
14	ECH14	
13	ECH13	
12	ECH12	
11	ECH11	
10	ECH10	
9	ECH9	
8	ECH8	
7	ECH7	
6	ECH6	
5	ECH5	
4	ECH4	
3	ECH3	
2	ECH2	
1	ECH1	
0	ECH0	

37.8.10 ADC 内部通道控制寄存器（ADC_ICHER）

名称	ADC_ICHER							
offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					DACEN	VREFEN	TSEN
位权限	U-0					R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	DACEN	DAC 采样通道使能，写 1 使能
1	VREFEN	AVREF 采样通道使能，写 1 使能
0	TSEN	温度传感器采样通道使能，写 1 使能

37.8.11 ADC 常规组数据寄存器（ADC_RDR）

名称	ADC_RDR							
offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-		CHID					
位权限	U-0		R-00 0000					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DATA[15:8]							
位权限	R-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DATA[7:0]							
位权限	R-0							

Bit	助记符	功能描述
31:30	--	RFU：未实现，读为 0
29:24	CHID	通道 ID，记录转换结果对应的通道编号 0~30 表示外部输入通道 31：温度传感器 32：AVREF 33：DAC 其他：RFU *注意：当注入转换发生时，注入组通道 ID 也会被写入 CHID 寄存器，但是注入组转换结果不会写入 ADC_RDR.DATA*
23:16	--	RFU：未实现，读为 0
15:0	DATA	ADC 转换结果 在没有使能过采样平均的情况下，结果为低 12bit；在使能过采样平均的情况下，结果为 12~16bit

37.8.12 AWD 阈值寄存器（ADC_HLTR）

名称	ADC_HLTR							
offset	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	AWD_HT[15:8]							
位权限	R/W-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	AWD_HT[7:0]							

位权限	R/W-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	AWD_LT[15:8]							
位权限	R/W-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AWD_LT[7:0]							
位权限	R/W-0							

Bit	助记符	功能描述
31:16	AWD_HT	AWD 监视高阈值，最长 16bit，软件根据实际所需位数设置，高位空余位数保持为 0
15:0	AWD_LT	AWD 监视低阈值，最长 16bit，软件根据实际所需位数设置，高位空余位数保持为 0

37.8.13 ADC 原始数据寄存器（ADC_RAWR）

名称	ADC_RAWR							
offset	0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名				RAWD[12:8]				
位权限					R-0			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RAWD[7:0]							
位权限	R-0							

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
12:0	RAWD	经过权重计算后的 13bit 数据

38 数模转换器（DAC）

38.1 概述

本芯片集成1个10位电压输出型数模转换器。DAC使用VDDA引脚电压作为参考基准，基准电压范围0~VDDA。

DAC可以为高速比较器提供比较基准。

DAC的主要特性如下：

- 工作电压 1.8～5.5V
- 输出信号幅度 VSSA~VDDA
- 最高输出转换率 1Msps（F_{DAC}=16Mhz）
- DAC 输 40 出可以连接到比较器输入和 ADC 输入

38.2 结构框图

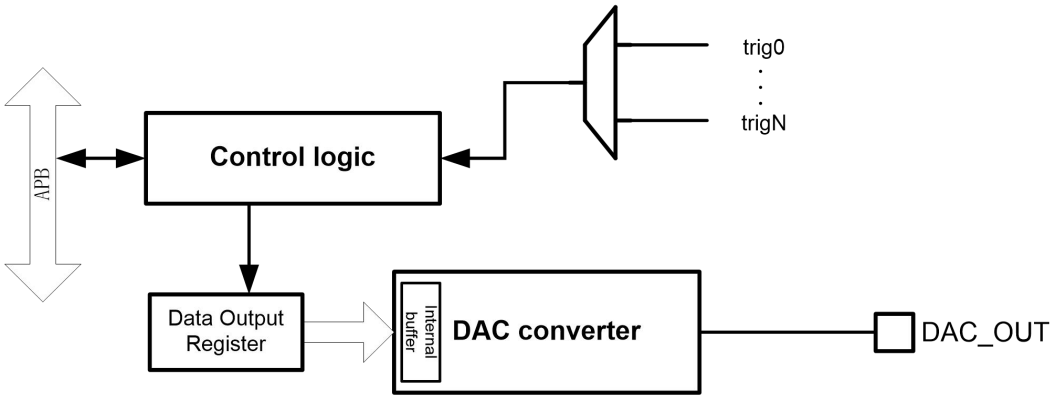


图 38-1 DAC 结构框图

38.3 引脚定义

DAC输出引脚与普通GPIO复用，使用DAC输出功能前需要将相关引脚配置为模拟功能。

引脚	类型	说明
VDDA	模拟电源	DAC 工作电源
VSSA	模拟地	DAC 的参考地
DAC_OUT	模拟输出	DAC 输出信号

表 38-1 DAC 相关引脚

38.4 功能描述

38.4.1 工作时钟与信号时序

DAC转换器使用APB时钟工作，数字码更新频率最高不能超过1MHz，否则无法保证良好的输出动态性能。DAC工作时钟来自于CMU模块，使用DAC之前必须正确配置时钟。

当DAC使能后，DAC有一个启动时间，经过t_{WAKEUP}时间之后，DAC输出电压才会开始建立，t_{WAKEUP}的具体指标参见3电参数。

软件或DMA将需要输出的数字码写入DHR寄存器。在连续输出模式下，DHR寄存器中的数字码将直接复制到DAC内部缓存中，并驱动DAC输出电平。在触发输出模式下，DHR寄存器中的数字码等待触发事件到来后，再更新到DAC内部缓存中。一旦数字码更新完成，DAC工作不再需要时钟。

DAC工作时序示意图如下：

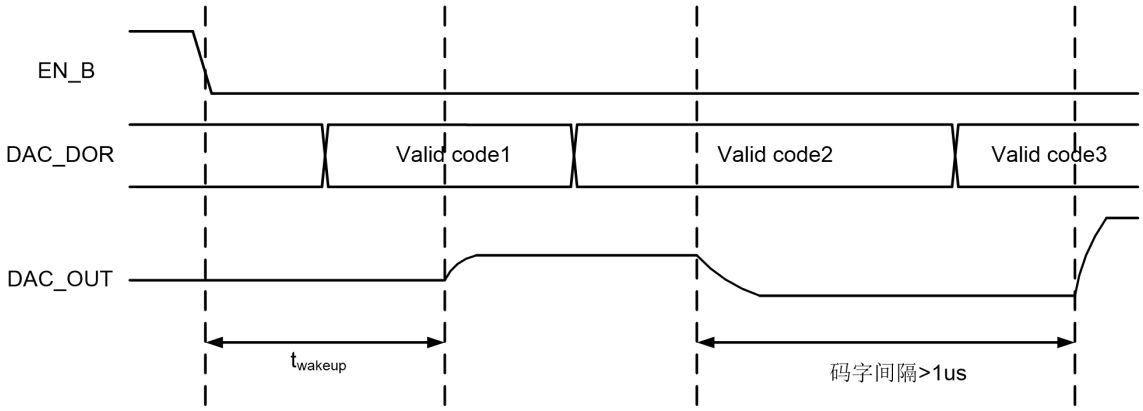


图 38-2 DAC 工作时序

38.4.2 DAC 输出模式

DAC输出可以采用触发方式或连续方式。

在TRGEN（Trigger Enable）为1的情况下，当trigger信号到来时，DAC_DHR寄存器中的数据被传输到DAC内部缓存中，DAC才会更新DAC_OUTx输出电压。

如果TRGEN=0，则软件对DAC_DHR寄存器的写操作将同时被映射到DAC内部缓存中，即DAC_OUTx的输出电平将被实时改变。

下图为DAC连续输出模式。在连续模式下，软件更新DAC_DHR寄存器的频率不能高于1Mhz，否则输出无法保证正确建立。

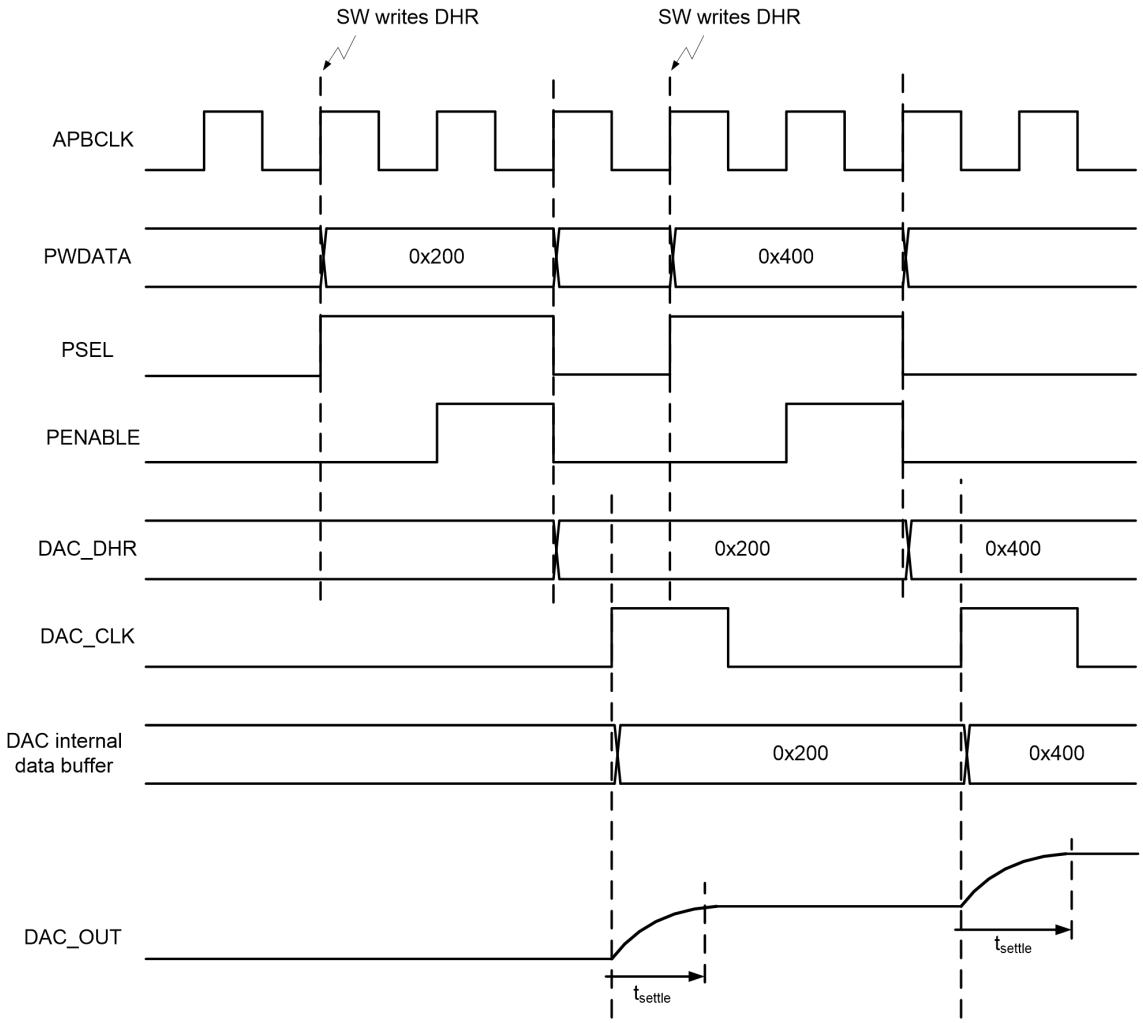


图 38-3 DAC 连续输出模式

下图为DAC触发输出模式。触发输出模式下，触发事件到来的频率不能超过1Mhz，否则输出无法保证正确建立。

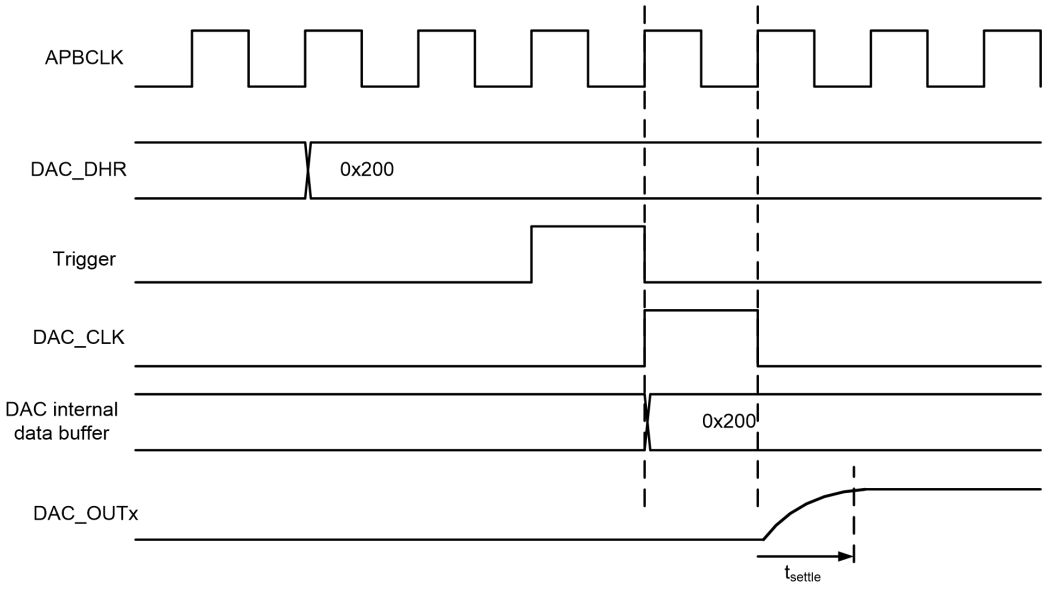


图 38-4 DAC 触发输出模式

DAC内部数据buffer被更新后，到DAC_OUTx输出电压稳定，有一个settling time，这个时间与电源电压、输出负载和DAC输出驱动能力有关。

38.4.3 DAC 触发源选择

DAC触发模式支持硬件触发或者软件触发。

通过TRGSEL寄存器，可以选择DAC使用的触发源，默认使用软件触发，软件可以通过对SWTRG寄存器写1实现DAC触发更新。

当选择其他触发信号源时，DAC将检测输入触发信号的上升沿，并将DAC_DHR中的数据更新到DAC_DOR寄存器中。

下表定义了TRGSEL不同配置对应的触发源。

触发源	类型	TRGSEL[3:0]
SWTRG	软件触发	0000
ATIM_TRGO	内部定时器触发信号	0001
GPTIM1_TRGO		0010
GPTIM2_TRGO		0011
BSTIM_TRGO		0100
LPTIM_TRGO		0101
TAU00_TRGO		0110
TAU03_TRGO		0111
RFU	-	1000
		1001
		1010
		1011
GPIO_EXTIFAB[13]	引脚中断触发	1100
GPIO_EXTIFAB[29]		1101
GPIO_EXTIFCD[4]		1110
GPIO_EXTIFCD[30]		1111

表 38-2 DAC 触发源

38.4.4 DAC 输出电压

DAC输出电压是在0~VREF+范围内对输入数字码的线性转换。

DAC输出电压论值由下式决定：

$$DAC_OUT = VDDA \times \frac{DAC_DOR}{1024}$$

38.4.5 DAC DMA 传输

DAC支持DMA功能。DMA功能仅能在触发模式下使用。

当DMAEN寄存器置位时，外部硬件触发信号（不含软件触发）将使DAC内部缓存被更新为DAC_DHR的内容，同时DAC会产生一个DMA请求。DMA控制器接收到DAC请求后，将自动更新DAC_DHR寄存器。

在当前的DMA请求完成前，新来的硬件触发信号将被忽略，同时DMAERR标志寄存器将被置位。出现这种情况说明触发信号过于密集，系统来不及刷新DAC输出，此时软件应降低触发信号的出现频率。

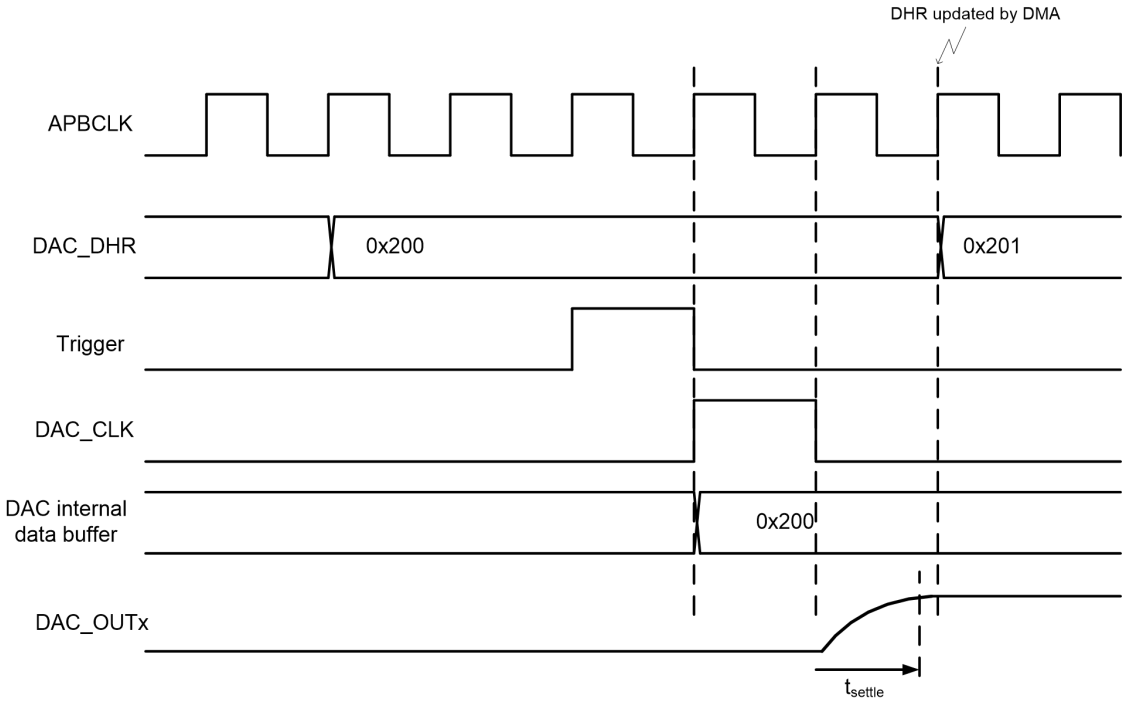


图 38-5 触发模式下通过 DMA 更新数据

38.4.6 DAC 输出缓冲

DAC本身输出无驱动能力，不能驱动阻性负载。在需要驱动能力的场合下，请在片外连接buffer。

38.5 寄存器

模块起始地址：0x40019000

offset 地址	名称	符号
0x00	DAC 控制寄存器 1 （DAC Control Register）	DAC_CR1
0x04	DAC 控制寄存器 2 （DAC Control Register）	DAC_CR2
0x08	DAC 配置寄存器	DAC_CFGR

	（DAC Config Register）	
0x0C	DAC 软件触发寄存器 （DAC Software Trigger Register）	DAC_SWTRGR
0x10	DAC 数据保持寄存器 （DAC Data Holding Register）	DAC_DHR
0x14	DAC 状态标志寄存器 （DAC Interrupt Status Register）	DAC_ISR
0x18	DAC 中断使能寄存器 （DAC Interrupt Enable Register）	DAC_IER

38.5.1 DAC 控制寄存器 1（DAC_CR1）

名称	DAC_CR1							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	EN	DAC 使能（enable） 0：关闭 DAC 1：使能 DAC

38.5.2 DAC 控制寄存器 2（DAC_CR2）

名称	DAC_CR2							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						DMAEN	TRGEN
位权限	U-0						R/W-0	R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
1	DMAEN	触发模式使能 (Trigger enable) 0：禁止 DMA 功能 1：使能 DMA 功能
0	TRGEN	触发模式使能 (Trigger enable) 0：禁止触发模式 1：使能触发模式

38.5.3 DAC 配置寄存器（DAC_CFGR）

名称	DAC_CFGR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		TRGSEL				-	SWIEN
位权限	U-0		R/W-0000				U-0	R/W-0

Bit	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5:2	TRGSEL	触发源选择 (Trigger select)，必须在 TRGEN 为 0 的情况下修改触发源选择 参见 38.4.3DAC 触发源选择
1	--	RFU：未实现，读为 0
0	SWIEN	DAC 反馈开关使能 (Switch Enable) 0：断开开关 1：使能开关

38.5.4 DAC 软件触发寄存器（DAC_SWTRGR）

名称	DAC_SWTRGR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							SWTRIG
位权限	U-0							W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	SWTRIG	DAC 软件触发寄存器 (Software Trigger)，软件置位，当 DAC_DHR 中数据载入 DAC_DOR 寄存器后硬件自动清零。

38.5.5 DAC 数据保持寄存器（DAC_DHR）

名称	DAC_DHR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						DHR	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DHR							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:10	--	RFU：未实现，读为 0
9:0	DHR	DAC 数据保持寄存器 (Data Holding Register)

38.5.6 DAC 状态标志寄存器（DAC_ISR）

名称	DAC_ISR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				DMAERR	-		DOU
位权限	U-0				R/W-0	U-0		R/W-0

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3	DMAERR	DMA 错误标志（DMA Error） 仅 DMAEN=1 时有效，软件写 1 清零
2:1	--	RFU：未实现，读为 0
0	DOU	数据输出寄存器更新标志 (Data Output Updated) 当 DHR 寄存器内容被写入 DOR 时，此标志置位，软件写 1 清零。 仅在触发方式下（TRGEN=1）有效。

38.5.7 DAC 中断使能寄存器（DAC_IER）

名称	DAC_IER							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				DMAE_IE	-		DOU_IE
位权限	U-0				R/W-0	U-0		R/W-0

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3	DMAE_IE	DMA 错误中断使能 (DMA Error Interrupt Enable) 1：允许 DMAERR 中断 0：禁止 DMAERR 中断
2:1	--	RFU：未实现，读为 0
0	DOU_IE	数据输出寄存器更新标志 (Data Output Updated Interrupt Enable) 1：允许 DOU 中断 0：禁止 DOU 中断

39 可编程胶合逻辑（PGL）

39.1 概述

可编程胶合逻辑（Programmable Glue Logic）是基于查找表（LUT）的简单可编程逻辑，其输入输出可以连接到芯片引脚、内部信号，实现一些简单的胶合逻辑，在一些应用中可以帮助系统设计减少PCB上的逻辑器件。

每个LUT包含4个输入，1个输出，1个真值表，可选的同步/滤波电路。用户通过对真值表编程，可以获得期望的组合逻辑输出表达式。每个输入信号都可以被单独屏蔽。

PGL的基本特性如下：

- 实现简单胶合逻辑，简化PCB设计
- 4个4输入查找表
- 可以通过真值表编程实现 AND,NAND,OR,NOR,XOR,NOT等逻辑表达式
- 时序同步或滤波
- 灵活的LUT输入选择：IO，内部信号，其他LUT输出
- 输出可以被连接到IO或其他外设触发

39.2 结构框图

LUT的结构如下图所示。

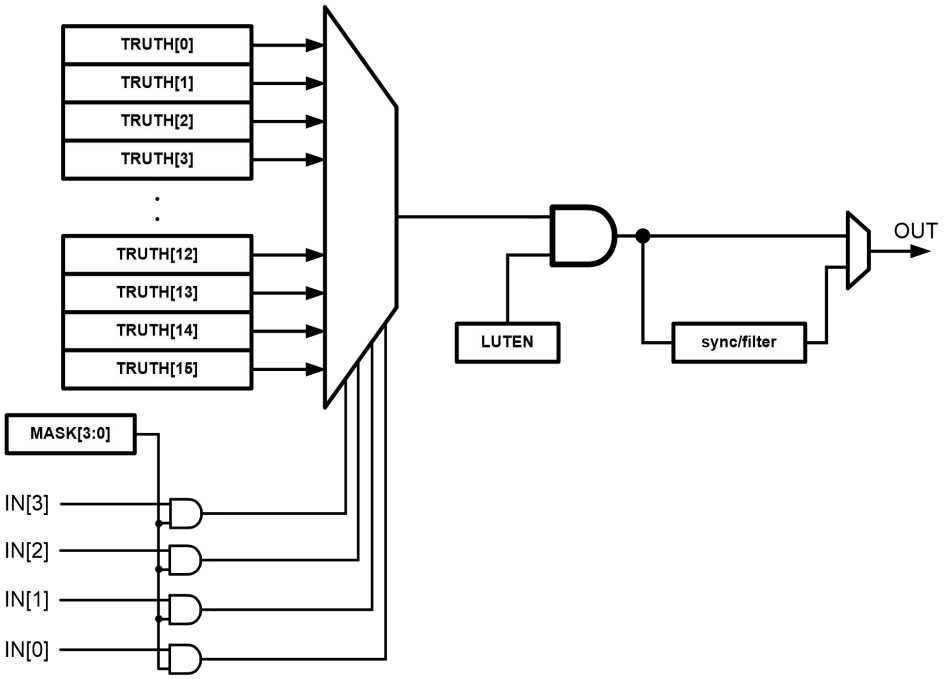


图 39-1LUT 结构框图

PGL模块共包含了4个LUT。

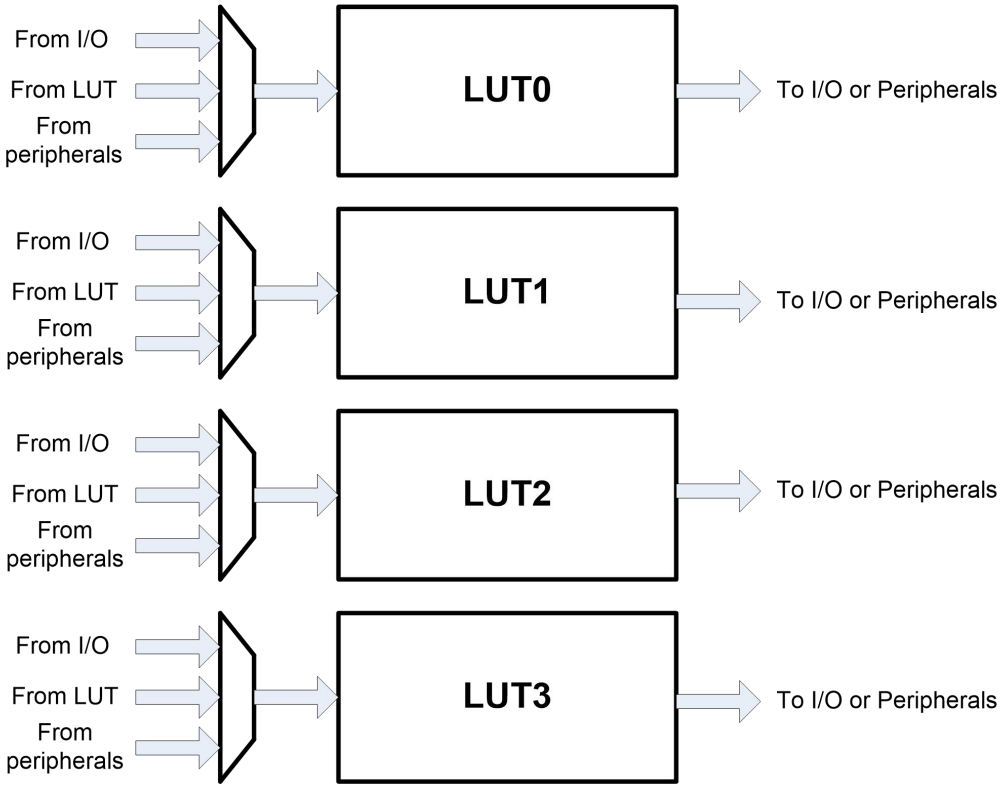


图 39-2 PGL 结构框图

39.3 引脚定义

PGL的输入输出引脚与普通GPIO复用，使用PGL功能需要将对应IO配置为GPIO输入或者输入。

引脚	类型	说明
PA0~PA7	GPIO 输入	LUT0 引脚输入通道
PB0~PB7	GPIO 输入	LUT1 引脚输入通道
PC0~PC7	GPIO 输入	LUT2 引脚输入通道
PD0~PD7	GPIO 输入	LUT3 引脚输入通道
PB8, PD2, PA2, PC0, PC8	额外数字功能	LUT0 输出
PA3, PB9, PC1, PC9, PD3		LUT1 输出
PA1, PA4, PB10, PC10, PD4		LUT2 输出
PA0, PA5, PB11, PC11, PD5		LUT3 输出

表 39-1 PGL 相关引脚

当使用LUT引脚输入时，需将对应引脚配置为GPIO输入，清除输入的MASK寄存器，并使能LUT。

当使用LUT引脚输出时，需将对应引脚配置为数字功能，并通过数字功能选择寄存器将引脚配置为LUT输出，然后使能LUT。

39.4 功能描述

39.4.1 LUT 真值表

通过输入状态组合查询LUT真值表中的数据，可以获得想要的组合逻辑功能。

举例来说，希望通过LUT0实现一个2输入NAND逻辑功能，可以采用以下配置：

- 将LUT0的输入2和输入3屏蔽为0
- LUT0的输入0和输入1连接到GPIO
- LUT0真值表寄存器低4bit写入0111

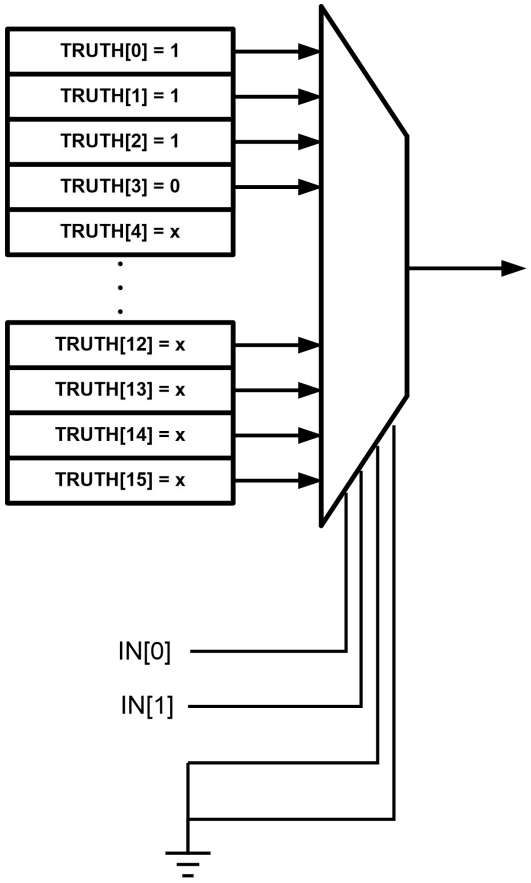


图 39-3 LUT 实现 2 输入 NAND 示意图

39.4.2 LUT 输入

每个LUT有4个输入，每个输入都可以来自于引脚、芯片外设、或其他LUT输出。

通过INSEL寄存器，可以选择LUT输入信号源，详情请参见寄存器章节。

39.4.3 LUT 输出

LUT的输出可以连接到引脚、或其他LUT输入。如下表所示。

LUT 输出	引脚	LUT 输入
LUT0 输出	PC0 PA2 PB8 PC8 PD2	LUT1,LUT2,LUT3
LUT1 输出	PC1 PA3 PB9 PC9 PD3	LUT2,LUT3
LUT2 输出	PA1 PA4 PB10 PC10 PD4	LUT3
LUT3 输出	PA0 PA5 PB11 PC11 PD5	-

表 39-2 LUT 输出连接

39.4.4 滤波和采样

LUT支持对输出的滤波和同步采样，其结构示意图如下。

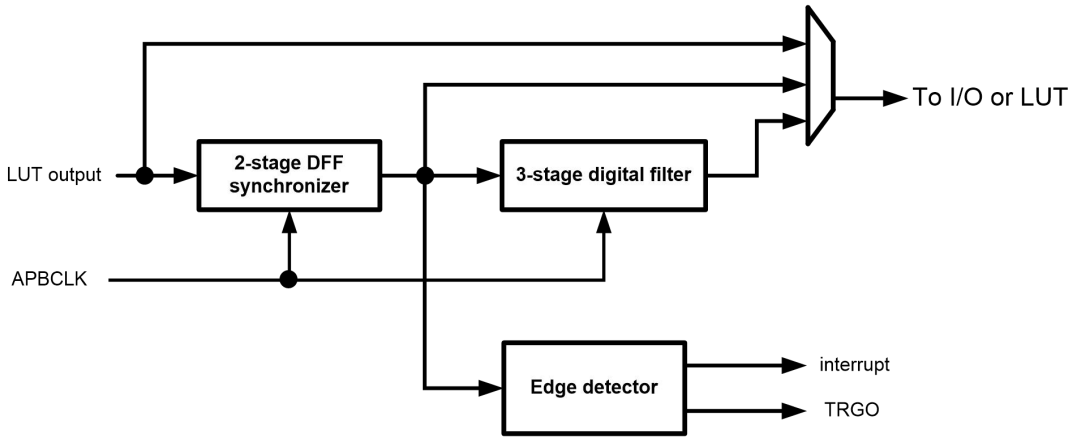


图 39-4 输出滤波和采样

数字滤波采用以下方法，当APBCLK连续采样到3个相同电平，则认为合法电平，否则滤波输出不会改变。

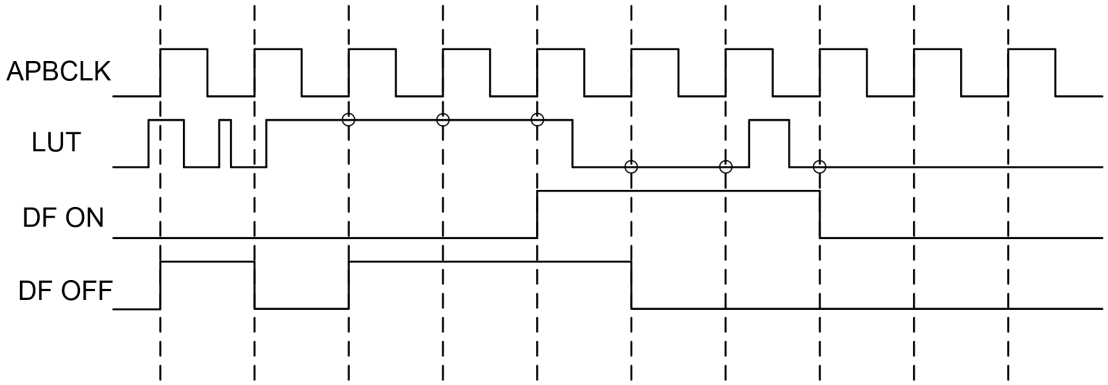


图 39-5 数字滤波

根据寄存器配置，应用可以选择LUT的输出为组合逻辑输出、APBCLK同步采样后的输出、或者为采样后经过数字滤波的输出。

经过同步采样后的信号，经过边沿产生电路，可以产生1个APBCLK周期宽度的上升沿或下降沿脉冲信号。这个同步脉冲信号可用于其它外设的trigger信号，或者产生中断。

应用上可以使用PGL对外部引脚输入信号进行数字滤波后，再提供给通信外设输入。

比如，使用LUT0给UART0输入做数字滤波：

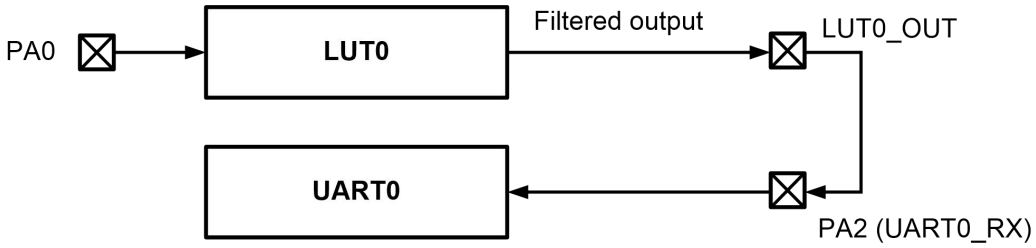


图 39-6 使用 LUT0 为 UART 输入做数字滤波

39.4.5 中断和触发

LUT输出经过采样和边沿选择后，可以产生中断标志或者输出触发信号。

必须使能LUT外设时钟，才能产生中断或触发。

39.4.6 低功耗模式

PGL可以在低功耗模式下使用。但是要注意滤波和采样功能需要APB时钟，因此无法在休眠模式下使用。

如果只是使用LUT的组合逻辑输出功能，则应用可以在配置完寄存器之后关闭PGL外设模块时钟，以节省功耗。

39.5 寄存器

模块起始地址：0x4001_6C00

offset 地址	名称	符号
0x00	PGL 控制寄存器	PGL_CR
0x04	PGL 配置寄存器 0	PGL_CFGR0
0x08	PGL 配置寄存器 1	PGL_CFGR1
0x0C	PGL 配置寄存器 2	PGL_CFGR2
0x10	PGL 配置寄存器 3	PGL_CFGR3
0x14	PGL 中断使能寄存器	PGL_IER
0x18	PGL 中断标志寄存器	PGL_ISR
0x1C	LUT0 真值表寄存器	PGL_LUT0
0x20	LUT1 真值表寄存器	PGL_LUT1
0x24	LUT2 真值表寄存器	PGL_LUT2
0x28	LUT3 真值表寄存器	PGL_LUT3

39.5.1 PGL 控制寄存器（PGL_CR）

名称	PGL_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				LUTEN[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	LUTEN	LUT 使能寄存器，分别控制 LUT0~3 1：使能 LUT[x] 0：关闭 LUT[x]

39.5.2 PGL 配置寄存器 0（PGL_CFGR0）

名称	PGL_CFGR0
offset	0x04

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				EDGESEL		OUTSEL	
位权限	U-0				R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				MASK[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19:18	EDGESEL	LUT0 有效边沿选择 00：上升沿产生中断和 TRGO 01：下降沿产生中断和 TRGO 10：上升下降沿都产生中断和 TRGO 11：不产生中断和 TRGO
17:16	OUTSEL	LUT0 输出选择 00：组合逻辑输出 01：同步采样输出 10：数字滤波输出 11：RFU
15:14	IN3SEL	LUT0 输入 3 选择寄存器 00：GPIO 01：LPT16_CH1 10：GPTIM1.OC1REF 11：ATIM.OC1REF
13:12	IN2SEL	LUT0 输入 2 选择寄存器 00：GPIO PA4 01：GPIO PA5 10：GPIO PA6 11：RFU
11:10	IN1SEL	LUT0 输入 1 选择寄存器 00：GPIO PA2 01：GPIO PA3 10：COMP2_OUT 11：GPTIM1.OC1REF
9:8	IN0SEL	LUT0 输入 0 选择寄存器 00：GPIO PA0 01：GPIO PA1 10：COMP1_OUT 11：GPTIM0.OC1REF
7:4	--	RFU：未实现，读为 0
3:0	MASK	LUT0 输入屏蔽寄存器 1：屏蔽对应的输入（即固定为 0） 0：不屏蔽输入

39.5.3 PGL 配置寄存器 1（PGL_CFGR1）

名称	PGL_CFGR1							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				EDGESEL		OUTSEL	
位权限	U-0				R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				MASK[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19:18	EDGESEL	LUT1 有效边沿选择 00：上升沿产生中断和 TRGO 01：下降沿产生中断和 TRGO 10：上升下降沿都产生中断和 TRGO 11：不产生中断和 TRGO
17:16	OUTSEL	LUT1 输出选择 00：组合逻辑输出 01：同步采样输出 10：数字滤波输出 11：RFU
15:14	IN3SEL	LUT1 输入 3 选择寄存器 00：GPIO PB7 01：LPT16_CH1 10：COMP3_OUT 11：ATIM.OC1REF
13:12	IN2SEL	LUT1 输入 2 选择寄存器 00：GPIO PB3 01：GPIO PB4 10：GPIO PB5 11：GPIO PB6
11:10	IN1SEL	LUT1 输入 1 选择寄存器 00：GPIO PB1 01：GPIO PB2 10：COMP2_OUT 11：GPTIM1.OC1REF
9:8	IN0SEL	LUT1 输入 0 选择寄存器 00：GPIO PB0 01：LUT0 output 10：COMP1_OUT 11：GPTIM0.OC1REF
7:4	--	RFU：未实现，读为 0

Bit	助记符	功能描述
3:0	MASK	LUT1 输入屏蔽寄存器 1：屏蔽对应的输入（即固定为 0） 0：不屏蔽输入

39.5.4 PGL 配置寄存器 2（PGL_CFGR2）

名称	PGL_CFGR2							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				EDGESEL		OUTSEL	
位权限	U-0				R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				MASK[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19:18	EDGESEL	LUT2 有效边沿选择 00：上升沿产生中断和 TRGO 01：下降沿产生中断和 TRGO 10：上升下降沿都产生中断和 TRGO 11：不产生中断和 TRGO
17:16	OUTSEL	LUT2 输出选择 00：组合逻辑输出 01：同步采样输出 10：数字滤波输出 11：RFU
15:14	IN3SEL	LUT2 输入 3 选择寄存器 00：GPIO PC6 01：LPT16_CH1 10：GPIO PC7 11：ATIM.OC1REF
13:12	IN2SEL	LUT2 输入 2 选择寄存器 00：GPIO PC5 01：GPIO PC4 10：GPIO PC3 11：GPIO PC2
11:10	IN1SEL	LUT2 输入 1 选择寄存器 00：GPIO PC1 01：LUT1 output 10：COMP2_OUT

Bit	助记符	功能描述
		11：GPTIM1.OC1REF
9:8	IN0SEL	LUT2 输入 0 选择寄存器 00：GPIO PC0 01：LUT0 output 10：COMP1_OUT 11：GPTIM0.OC1REF
7:4	--	RFU：未实现，读为 0
3:0	MASK	LUT2 输入屏蔽寄存器 1：屏蔽对应的输入（即固定为 0） 0：不屏蔽输入

39.5.5 PGL 配置寄存器 3（PGL_CFGR3）

名称	PGL_CFGR3							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				EDGESEL		OUTSEL	
位权限	U-0				R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				MASK[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19:18	EDGESEL	LUT3 有效边沿选择 00：上升沿产生中断和 TRGO 01：下降沿产生中断和 TRGO 10：上升下降沿都产生中断和 TRGO 11：不产生中断和 TRGO
17:16	OUTSEL	LUT3 输出选择 00：组合逻辑输出 01：同步采样输出 10：数字滤波输出 11：RFU
15:14	IN3SEL	LUT3 输入 3 选择寄存器 00：GPIO PD7 01：GPIO PD6 10：LPT16_CH1 11：GPIO PD5
13:12	IN2SEL	LUT3 输入 2 选择寄存器 00：GPIO PD2

Bit	助记符	功能描述
		01：LUT2 output 10：GPIO PD3 11：GPIO PD4
11:10	IN1SEL	LUT3 输入 1 选择寄存器 00：GPIO PD1 01：LUT1 output 10：COMP2_OUT 11：GPTIM1.OC1REF
9:8	IN0SEL	LUT3 输入 0 选择寄存器 00：GPIO PD0 01：LUT0 output 10：COMP1_OUT 11：GPTIM0.OC1REF
7:4	--	RFU：未实现，读为 0
3:0	MASK	LUT3 输入屏蔽寄存器 1：屏蔽对应的输入（即固定为 0） 0：不屏蔽输入

39.5.6 PGL 中断使能寄存器（PGL_IER）

名称	PGL_IER							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				LUTIE[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	LUTIE	LUT 中断使能寄存器，分别控制 LUT0~3 1：使能 LUT[x]中断 0：禁止 LUT[x]中断

39.5.7 PGL 中断标志寄存器（PGL_ISR）

名称	PGL_ISR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				LUTIF[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	LUTIF	LUT 中断标志寄存器，硬件置位，软件写 1 清零

39.5.8 LUT0 真值表寄存器（PGL_LUT0）

名称	PGL_LUT0							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TRUTH[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRUTH[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	TRUTH	LUT0 真值表，软件可以写入任意值以获得需要的逻辑组合

39.5.9 LUT1 真值表寄存器（PGL_LUT1）

名称	PGL_LUT1							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	TRUTH[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRUTH[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	TRUTH	LUT1 真值表，软件可以写入任意值以获得需要的逻辑组合

39.5.10 LUT2 真值表寄存器（PGL_LUT2）

名称	PGL_LUT2							
offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TRUTH[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRUTH[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	TRUTH	LUT2 真值表，软件可以写入任意值以获得需要的逻辑组合

39.5.11 LUT3 真值表寄存器（PGL_LUT3）

名称	PGL_LUT3							
offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TRUTH[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRUTH[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	TRUTH	LUT3 真值表，软件可以写入任意值以获得需要的逻辑组合

40 I/O 端口（GPIO）

40.1 概述

I/O 端口的主要功能特性：

- GPIO 数字输入具有施密特特性
- 部分 GPIO 输入支持模拟滤波
- 部分 GPIO 输入支持数字滤波
- GPIO 可配置为上拉、下拉、开漏输出
- Sleep/DeepSleep 模式下保持状态
- 上下电复位期间，所有 IO 控制信号被钳位到 disable 状态
- 可配置输出驱动能力（x1, x2）

40.2 引脚类型

FM33FG0A 主要有三种类型的 GPIO 引脚，其中大部分引脚支持输入输出、数字外设功能、模拟外设通道、可控上拉电阻、可控开漏输出功能；强驱动引脚除了以上功能外，具有增强的推挽输出驱动能力；

40.2.1 GPIO，输入输出使能，可控上下拉电阻，可控开漏输出，可控驱动能力

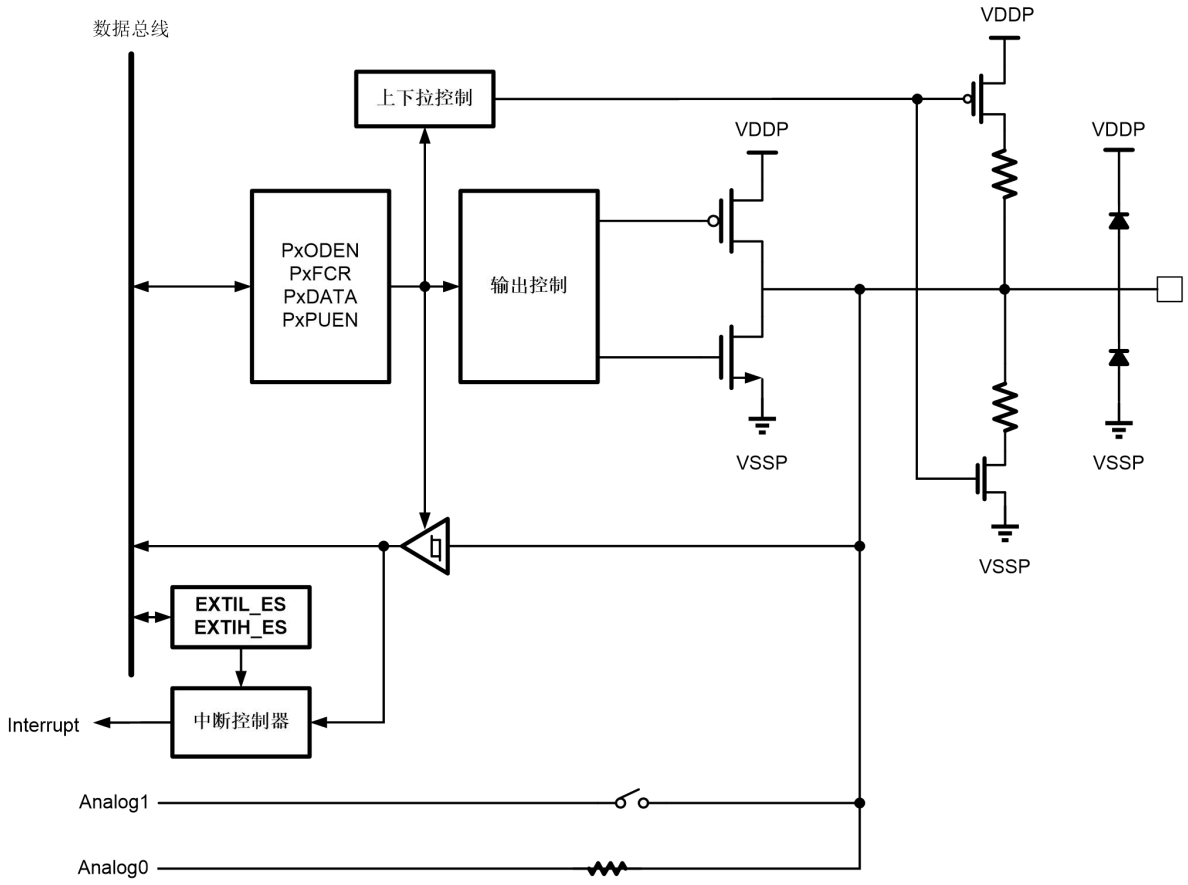


图 40-1 普通 GPIO 结构框图

控制逻辑定义如下：

Registers					PAD Interface		
FCR	INEN	ODEN	PUEN	DATA	INPUT_EN	OUTPUT_EN	PUEN
00	0	x	0/1	x	0	0	0/1
	1				1		
01	x	0	0/1	x	0	1	0/1
	x	1		0	0	1	
				1	0	0	
10	x	x	0/1	外设输入功能	1	0	0/1
	x	0		外设推挽输出功能	0	1	
	x	1		外设开漏输出 0	0	1	
				外设开漏输出 1	0	0	
11	x	x	x	x	0	0	0

表 40-1GPIO 功能逻辑定义表

上下拉电阻各50K欧姆

2级驱动能力：

X1 – 4mA

X2 – 8mA

40.3 IO 端口功能定义

芯片大部分引脚为数模混合IO，每个通用GPIO都有4bit控制寄存器：FCR[1:0]、PUEN、ODEN，其中FCR用于选择IO引脚功能，定义如下：

FCR: Function Control Register	PAD function
00	GPIO input
01	GPIO output
10	Digital Function（数字外设功能）
11	Analog

表 40-2FCR 定义表

40.3.1 GPIO 输入

当某个 GPIO 被配置成输入功能，并且对应的输入使能寄存器被置位时：

- 输出驱动缓冲器被关闭
- 施密特触发器使能
- 上拉电阻由 PxPUEN 寄存器控制使能或关闭
- 下拉电阻由 PxPDEN 寄存器控制使能或关闭
- PxDIN 寄存器直接反应 IO 上的电平状态

40.3.2 GPIO 输出

当某个 GPIO 被配置成输出功能，并且对应的输出使能寄存器被置位时：

- 输出驱动缓冲器使能
  - 开漏输出模式（PxODEN=1）：输出 0 时 IO 驱动低电平，输出 1 时 IO 关闭驱动缓冲器
  - 推挽输出模式（PxODEN=0）：输出 0 时 IO 驱动低电平，输出 1 时 IO 驱动高电平
- 上拉电阻由 PxPUEN 寄存器控制使能或关闭

- 软件读取 PxDIN 寄存器能够获得 IO 上的电平状态
- 软件读取 PxDO 寄存器获得上次写入的值
- 支持 4 级驱动能力可配置，解决驱动能力和 EMI 之间的平衡

40.3.3 数字外设功能

当某个 GPIO 被配置成数字外设功能：

- IO 的输入或输出方向由所连接的外设功能决定
- 由 PxODEN 控制输出时是开漏输出还是推挽输出
- 上拉电阻由 PxPUEN 寄存器控制使能或关闭
- 软件读取 PxDIN 寄存器能够获得 IO 上的电平状态
- 每个 IO 支持 2bit 外设功能选择，可以自由选择 4 种外设功能接入

额外的外设功能选择通过 AFSELx 寄存器实现。具体引脚功能列表参见 2 引脚和封装。

当存在多个 GPIO 连接到某个数字外设的输入时，为了避免数据冲突，按照 A->E、0->15 的优先级判决进行数据选择。比如，PA0 和 PB0 都连接某个外设输入，则 PA0 数据被送到外设端，PB0 输入信号无效。

40.3.4 模拟功能

当某个 GPIO 被配置成模拟功能：

- 输出缓冲器关闭
- 数字输入功能关闭
- 上拉电阻关闭
- 软件读取 PxDIN 返回 0
- IO 模拟通道被连接到特定的模拟外设上
- 如果一个 IO 同时连接到多个模拟外设，则多个模拟外设在同一时刻只能使能其中一个

40.3.5 使用外部晶体引脚

FM33FG0A 支持外接 32768Hz 晶体和 4~16MHz 高频晶体。

其中 PC2 和 PC3 默认为 GPIO，配置为模拟功能之后作为 XTHFIN 和 XTHFOUT 外接高频晶体。

PI7 和 PI8 默认为模拟功能，接 32768Hz 晶体，软件可以配置为 GPIO 使用。

如果要使用外部 32K 时钟输入，可从 XT32KI 灌入 32K 时钟。

40.4 SWD 引脚

ARM SWD 引脚复用 PD7(SWCLK)和 PD8(SWIO)，这两个 GPIO 上电复位后默认为 SWD 功能，并且默认使能内部上拉电阻(50K)，以省去外部上拉。

40.5 WKUPx 引脚

FM33FG0A 有 8 个 WKUP 引脚，能够将芯片从 Sleep/DeepSleep 模式下唤醒，即使片上振荡器都停止工作，WKUP 仍能唤醒芯片。

WKUPx 引脚输入上升沿、下降沿或上升下降沿（软件配置）能够将芯片从休眠模式下唤醒。为了使能此功能，需将对应引脚配置为 GPIO 输入功能，并且相应的 PINWKEN 置位，注意 PAD 内部带有上拉电阻，如果配置为上升沿唤醒，则必须关闭上拉电阻。

每个支持 WKUP 功能的 IO 都带有大约 100ns 的片内模拟滤波，能够滤除输入信号上的毛刺，避免误触发。

Sleep/DeepSleep 模式下，使能了的 WKUPx 引脚上任何大于 100ns 的脉冲都会触发芯片唤醒。

9 路 WKUPx 电路结构完全独立，下图显示了其中一路 WKUP 功能的结构框图

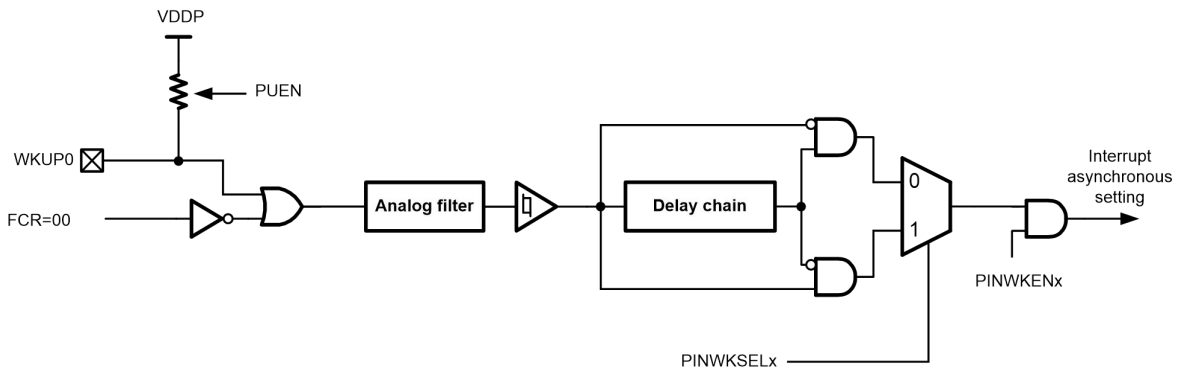


图 40-2WKUPx 功能结构框图

WKUPx 功能使用时需要注意外部引脚输入的初始状态。在使能 WKUP 时，可能由于初态的关系导致虚假的唤醒事件，软件应注意识别并处理。

使用 WKUP 功能时，必须将对应引脚的 FCR 寄存器配置为 00（GPIO 输入），按需要设置唤醒边沿（PINWKSELx）并使能 PINWKENx 寄存器。当某个 WKUPx 引脚上产生唤醒事件后，PMU 模块内部的唤醒源标志查询寄存器内对应的 bit 位将会自动置位。

40.6 外部引脚中断（EXTI）

40.6.1 功能说明

芯片的 GPIO 组 A~H 中，每个引脚都可以独立产生 EXTI 中断，最终所有的 EXTI 中断汇总到 NVIC 的中断入口。

总共可以产生多达 128 个引脚中断事件。软件可以单独配置每个管脚的中断事件为上升沿、下降沿、上升下降沿、或不产生中断。

EXTI_xSEL 寄存器用于选择某个 IO 接入 EXTI 通道，EXTI 模块可以配置是否对输入信号进行数字滤波。

数字滤波的实现方法是由 IO 采样时钟连续采样到 3 次相同电平才认为是合法电平输入，如下图所示。

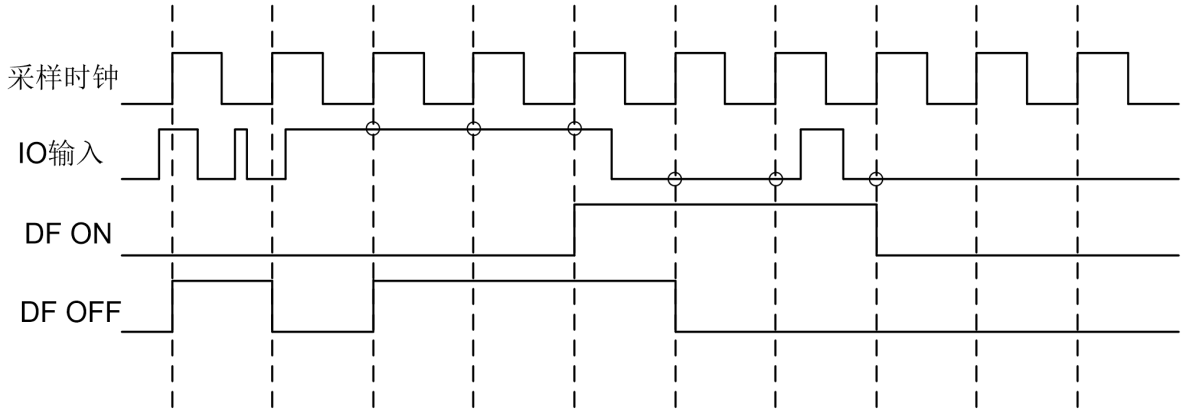


图 40-3 引脚输入数字滤波

当软件可以选择数字滤波的采样时钟为 APBCLK 或者 LSCLK。

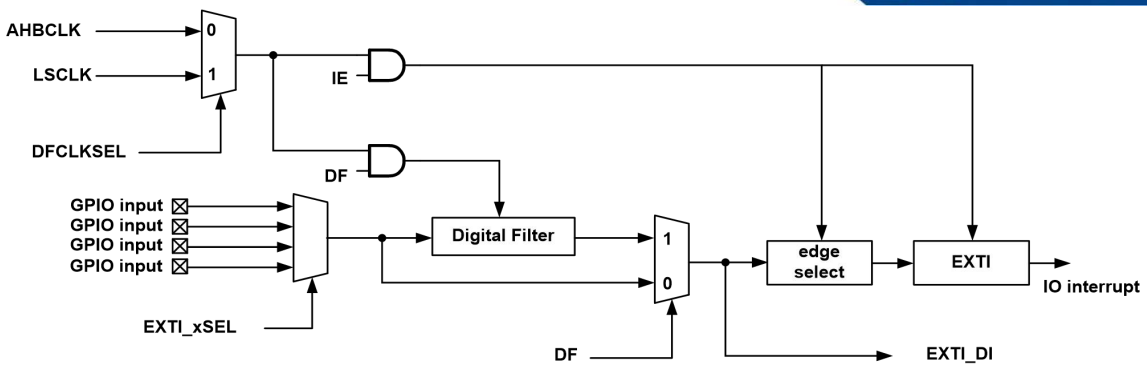


图 40-4 EXTI 信号输入示意图

用户应根据引脚功能需要使能或禁止数字滤波功能，使能数字滤波后，将根据 AHBCLK 频率不同，对 IO 输入信号引入不同的采样延迟。经过数字滤波后的输出信号，软件也可以在 EXTI_DI 寄存器读到。

EXTI还可以配置输入信号的有效边沿，支持上升沿、下降沿、上升下降沿触发中断，或者禁止EXTI中断触发，由EXTI_EDS寄存器配置。

40.6.2 应用指南

如需在 Sleep/DeepSleep 模式下启动 EXTI 中断唤醒功能，推荐按照如下步骤进行操作：

- 关闭所有 EXTI 使能
- 配置 SYSCLKSEL 寄存器（0x4000020C）的 EXTICKSEL 位为 1，选择 LSCLK 进行 EXTI 采样
- 根据需要打开或关闭 EXTI 数字滤波使能
- 配置相应 GPIO 为输入
- 配置 EXTI_SEL 寄存器选择对应的 IO
- 置位 EXTICKE，打开 EXTI 工作时钟使能
- 等待至少 4 个 LSCLK 周期
- 配置 EXTI_EDS 触发边沿选择，使能所需的 EXTI 中断
- 正常进入 Sleep 模式

芯片上电后默认关闭所有 EXTI，同时默认的引脚中断采样时钟是系统时钟 APBCLK。如果用户使用系统时钟产生 EXTI，推荐流程如下：

- 打开数字滤波使能（如果需要）
- 配置 GPIO 为输入
- 置位 EXTICKE，打开 EXTI 工作时钟使能
- 等待至少 4 个 APBCLK 周期

- 配置 EXTI_EDS 触发边沿选择，使能所需的 EXTI 中断

如果希望使用低速的 LSCLK 来产生 EXTI，推荐流程如下：

- 将 EXTI 采样时钟配置为 LSCLK
- 打开数字滤波使能（如果需要）
- 配置 GPIO 为输入
- 置位 EXTICKE，打开 EXTI 采样时钟使能
- 等待至少 4 个 LSCLK 时钟周期
- 配置 EXTI_EDS 触发边沿，使能所需的 EXTI 中断

40.7 快速 GPIO 输出

FM33FG0A 可以通过 set-reset 功能快速改变每个 GPIO 的输出数据（bitwise operation），从而提高 IO 输出效率，特别是可以提高 read-modify-write 操作的效率和可靠性（atomic）。方法是每个 GPIO 组的输出数据寄存器都有 2 组 set-reset 映射虚拟地址，对 set 寄存器特定 bit 写 1 可以置位对应的数据寄存器的 bit 位，对 reset 寄存器特定地址写 1 可以清除对应的数据寄存器的 bit 位。

40.8 寄存器

GPIO模块基地址：0x4000_0C00

offset 地址	名称	符号
0x00	GPIOA 输入使能寄存器 （GPIOA Input Enable Register）	GPIOA_INEN
0x04	GPIOA 上下拉使能寄存器 （GPIOA Pull-Up Pull-Down Enable Register）	GPIOA_PUDEN
0x08	GPIOA 开漏使能寄存器 （GPIOA Open-Drain Enable Register）	GPIOA_ODEN
0x0C	GPIOA 功能选择寄存器 （GPIOA Function Control Register）	GPIOA_FCR
0x10	GPIOA 输出数据寄存器 （GPIOA Data Output Register）	GPIOA_DO
0x14	GPIOA 输出数据置位寄存器 （GPIOA Data Set Register）	GPIOA_DSET
0x18	GPIOA 输出数据复位寄存器 （GPIOA Data Reset Register）	GPIOA_DRST
0x1C	GPIOA 输入数据寄存器 （GPIOA Data Input Register）	GPIOA_DI
0x20	GPIOA 额外数字功能寄存器 （GPIOA Digital Function Select）	GPIOA_DFS
0x30	GPIOA 驱动能力配置寄存器 （GPIOA Driver Strength Register）	GPIOA_DSR
0x40	GPIOB 输入使能寄存器 （GPIOB Input Enable Register）	GPIOB_INEN
0x44	GPIOB 上拉使能寄存器 （GPIOB Pull-Up Enable Register）	GPIOB_PUDEN
0x48	GPIOB 开漏使能寄存器 （GPIOB Opeb-Drain Enable Register）	GPIOB_ODEN
0x4C	GPIOB 功能选择寄存器 （GPIOB Function Control Register）	GPIOB_FCR
0x50	GPIOB 输出数据寄存器 （GPIOB Data Output Register）	GPIOB_DO
0x54	GPIOB 输出数据置位寄存器 （GPIOB Data Set Register）	GPIOB_DSET
0x58	GPIOB 输出数据复位寄存器 （GPIOB Data Reset Register）	GPIOB_DRST
0x5C	GPIOB 输入数据寄存器 （GPIOBData Input Register）	GPIOB_DIN
0x60	GPIOB 额外数字功能寄存器 （GPIOB Digital Function Select）	GPIOB_DFS
0x70	GPIOB 驱动能力配置寄存器 （GPIOB Driver Strength Register）	GPIOB_DSR
0x80	GPIOC 输入使能寄存器	GPIOC_INEN

offset 地址	名称	符号
	（GPIOC Input Enable Register）	
0x84	GPIOC 上拉使能寄存器 （GPIOC Pull-Up Enable Register）	GPIOC_PUDEN
0x88	GPIOC 开漏使能寄存器 （GPIOC Opeb-Drain Enable Register）	GPIOC_ODEN
0x8C	GPIOC 功能选择寄存器 （GPIOC Function Control Register）	GPIOC_FCR
0x90	GPIOC 输出数据寄存器 （GPIOC Data Output Register）	GPIOC_DO
0x94	GPIOC 输出数据置位寄存器 （GPIOC Data Set Register）	GPIOC_DSET
0x98	GPIOC 输出数据复位寄存器 （GPIOC Data Reset Register）	GPIOC_DRST
0x9C	GPIOC 输入数据寄存器 （GPIOC Data Input Register）	GPIOC_DIN
0xA0	GPIOC 额外数字功能寄存器 （GPIOC Digital Function Select）	GPIOC_DFS
	-	-
0xB0	GPIOC 驱动能力配置寄存器 （GPIOC Driver Strength Register）	GPIOC_DSR
0xC0	GPIOD 输入使能寄存器 （GPIOD Input Enable Register）	GPIOD_INEN
0xC4	GPIOD 上拉使能寄存器 （GPIOD Pull-Up Enable Register）	GPIOD_PUDEN
0xC8	GPIOD 开漏使能寄存器 （GPIOD Opeb-Drain Enable Register）	GPIOD_ODEN
0xCC	GPIOD 功能选择寄存器 （GPIOD Function Control Register）	GPIOD_FCR
0xD0	GPIOD 输出数据寄存器 （GPIOD Data Output Register）	GPIOD_DO
0xD4	GPIOD 输出数据置位寄存器 （GPIOD Data Set Register）	GPIOD_DSET
0xD8	GPIOD 输出数据复位寄存器 （GPIOD Data Reset Register）	GPIOD_DRST
0xDC	GPIOD 输入数据寄存器 （GPIOD Data Input Register）	GPIOD_DIN
0xE0	GPIOD 额外数字功能寄存器 （GPIOD Digital Function Select）	GPIOD_DFS
0xF0	GPIOD 驱动能力配置寄存器 （GPIOD Driver Strength Register）	GPIOD_DSR
0x100	GPIOE 输入使能寄存器 （GPIOE Input Enable Register）	GPIOE_INEN
0x104	GPIOE 上拉使能寄存器 （GPIOE Pull-Up Enable Register）	GPIOE_PUDEN
0x108	GPIOE 开漏使能寄存器 （GPIOE Opeb-Drain Enable Register）	GPIOE_ODEN

offset 地址	名称	符号
0x10C	GPIOE 功能选择寄存器 （GPIOE Function Control Register）	GPIOE_FCR
0x110	GPIOE 输出数据寄存器 （GPIOE Data Output Register）	GPIOE_DO
0x114	GPIOE 输出数据置位寄存器 （GPIOE Data Set Register）	GPIOE_DSET
0x118	GPIOE 输出数据复位寄存器 （GPIOE Data Reset Register）	GPIOE_DRST
0x11C	GPIOE 输入数据寄存器 （GPIOE Data Input Register）	GPIOE_DIN
0x120	GPIOE 额外数字功能寄存器 （GPIOE Digital Function Select）	GPIOE_DFS
0x130	GPIOE 驱动能力配置寄存器 （GPIOE Driver Strength Register）	GPIOE_DSR
0x140	GPIOF 输入使能寄存器 （GPIOF Input Enable Register）	GPIOF_INEN
0x144	GPIOF 上拉使能寄存器 （GPIOF Pull-Up Enable Register）	GPIOF_PUDEN
0x148	GPIOF 开漏使能寄存器 （GPIOF Opeb-Drain Enable Register）	GPIOF_ODEN
0x14C	GPIOF 功能选择寄存器 （GPIOF Function Control Register）	GPIOF_FCR
0x150	GPIOF 输出数据寄存器 （GPIOF Data Output Register）	GPIOF_DO
0x154	GPIOF 输出数据置位寄存器 （GPIOF Data Set Register）	GPIOF_DSET
0x158	GPIOF 输出数据复位寄存器 （GPIOF Data Reset Register）	GPIOF_DRST
0x15C	GPIOF 输入数据寄存器 （GPIOF Data Input Register）	GPIOF_DIN
0x160	GPIOF 额外数字功能寄存器 （GPIOF Digital Function Select）	GPIOF_DFS
0x170	GPIOF 驱动能力配置寄存器 （GPIOF Driver Strength Register）	GPIOF_DSR
0x180	GPIOG 输入使能寄存器 （GPIOG Input Enable Register）	GPIOG_INEN
0x184	GPIOG 上拉使能寄存器 （GPIOG Pull-Up Enable Register）	GPIOG_PUDEN
0x188	GPIOG 开漏使能寄存器 （GPIOG Opeb-Drain Enable Register）	GPIOG_ODEN
0x18C	GPIOG 功能选择寄存器 （GPIOG Function Control Register）	GPIOG_FCR
0x190	GPIOG 输出数据寄存器 （GPIOG Data Output Register）	GPIOG_DO
0x194	GPIOG 输出数据置位寄存器	GPIOG_DSET

offset 地址	名称	符号
	（GPIOG Data Set Register）	
0x198	GPIOG 输出数据复位寄存器 （GPIOG Data Reset Register）	GPIOG_DRST
0x19C	GPIOG 输入数据寄存器 （GPIOG Data Input Register）	GPIOG_DIN
0x1A0	GPIOG 额外数字功能寄存器 （GPIOG Digital Function Select）	GPIOG_DFS
0x1B0	GPIOG 驱动能力配置寄存器 （GPIOG Driver Strength Register）	GPIOG_DSR
0x1C0	GPIOH 输入使能寄存器 （GPIOH Input Enable Register）	GPIOH_INEN
0x184	GPIOH 上拉使能寄存器 （GPIOH Pull-Up Enable Register）	GPIOH_PUDEN
0x188	GPIOH 开漏使能寄存器 （GPIOH Opeb-Drain Enable Register）	GPIOH_ODEN
0x18C	GPIOH 功能选择寄存器 （GPIOH Function Control Register）	GPIOH_FCR
0x190	GPIOH 输出数据寄存器 （GPIOH Data Output Register）	GPIOH_DO
0x194	GPIOH 输出数据置位寄存器 （GPIOH Data Set Register）	GPIOH_DSET
0x198	GPIOH 输出数据复位寄存器 （GPIOH Data Reset Register）	GPIOH_DRST
0x19C	GPIOH 输入数据寄存器 （GPIOH Data Input Register）	GPIOH_DIN
0x1F0	GPIOH 额外数字功能寄存器 （GPIOH Digital Function Select）	GPIOH_DFS
0x200	GPIOH 驱动能力配置寄存器 （GPIOH Driver Strength Register）	GPIOH_DSR
0x204	GPIOI 输入使能寄存器 （GPIOI Input Enable Register）	GPIOI_INEN
0x208	GPIOI 上拉使能寄存器 （GPIOI Pull-Up Enable Register）	GPIOI_PUDEN
0x20C	GPIOI 开漏使能寄存器 （GPIOI Opeb-Drain Enable Register）	GPIOI_ODEN
0x210	GPIOI 功能选择寄存器 （GPIOI Function Control Register）	GPIOI_FCR
0x214	GPIOI 输出数据寄存器 （GPIOI Data Output Register）	GPIOI_DO
0x218	GPIOI 输出数据置位寄存器 （GPIOI Data Set Register）	GPIOI_DSET
0x21C	GPIOI 输出数据复位寄存器 （GPIOI Data Reset Register）	GPIOI_DRST
0x220	GPIOI 输入数据寄存器 （GPIOI Data Input Register）	GPIOI_DIN

offset 地址	名称	符号
	GPIOI 额外数字功能寄存器 （GPIOI Digital Function Select）	GPIOI_DFS
0x230	GPIOI 驱动能力配置寄存器） （GPIOI Driver Strength Register）	GPIOI_DSR
0x240	EXTI 边沿选择和使能寄存器 A （External Interrupt Edge Select and Enable RegisterA）	GPIO_EXTIA
0x244	EXTI 边沿选择和使能寄存器 B （External Interrupt Edge Select and Enable RegisterB）	GPIO_EXTIB
0x248	EXTI 边沿选择和使能寄存器 C （External Interrupt Edge Select and Enable RegisterC）	GPIO_EXTIC
0x24C	EXTI 边沿选择和使能寄存器 D （External Interrupt Edge Select and Enable RegisterD）	GPIO_EXTID
0x250	EXTI 边沿选择和使能寄存器 E （External Interrupt Edge Select and Enable RegisterE）	GPIO_EXTIE
0x254	EXTI 边沿选择和使能寄存器 F （External Interrupt Edge Select and Enable RegisterF）	GPIO_EXTIF
0x258	EXTI 边沿选择和使能寄存器 G （External Interrupt Edge Select and Enable RegisterG）	GPIO_EXTIG
0x25C	EXTI 边沿选择和使能寄存器 H （External Interrupt Edge Select and Enable RegisterH）	GPIO_EXTIH
0x260	EXTI 边沿选择和使能寄存器 I （External Interrupt Edge Select and Enable RegisterI）	GPIO_EXTII
0x264	EXTI 数字滤波控制寄存器 AB （External Interrupt Digital Filter Register AB）	GPIO_EXTIDFAB
0x268	EXTI 数字滤波控制寄存器 CD （External Interrupt Digital Filter Register CD）	GPIO_EXTIDFCD
0x26C	EXTI 数字滤波控制寄存器 EF （External Interrupt Digital Filter Register EF）	GPIO_EXTIDFEF
0x270	EXTI 数字滤波控制寄存器 GH （External Interrupt Digital Filter Register GH）	GPIO_EXTIDFGH
0x274	EXTI 数字滤波控制寄存器 I （External Interrupt Digital Filter Register I）	GPIO_EXTIDFI
0x278	EXTI 中断标志寄存器 AB （External Digital Filter Control Register AB）	GPIO_EXTIFAB
0x27C	EXTI 中断标志寄存器寄存器 CD （External Digital Filter Control Register CD）	GPIO_EXTIFCD
0x280	EXTI 中断标志寄存器存器 EF （External Digital Filter Control Register EF）	GPIO_EXTIFEF
0x284	EXTI 中断标志寄存器寄存器 GH	GPIO_EXTIFGH

offset 地址	名称	符号
	（External Digital Filter Control Register GH）	
0x288	EXTI 中断标志寄存器寄存器 I （External Digital Filter Control Register I）	GPIO_EXTIFI
0x28C	EXTI 输入信号寄存器 AB （External Data Input Register AB）	GPIO_EXTIDIAB
0x290	EXTI 输入信号寄存器 CD （External Data Input Register CD）	GPIO_EXTIDICD
0x294	EXTI 输入信号寄存器 EF （External Data Input Register EF）	GPIO_EXTIDIEF
0x298	EXTI 输入信号寄存器 GH （External Data Input Register GH）	GPIO_EXTIDIGH
0x29C	EXTI 输入信号寄存器 I （External Data Input Register I）	GPIO_EXTIDII
0x2C0	FOUT 配置寄存器 （Frequency Output Select Register）	GPIO_FOUTSEL
0x31C	WKUP 控制寄存器 （Wakeup Enable Register）	GPIO_PINWKEN

40.8.1 GPIO 输入使能寄存器（GPIOx_INEN）

名称	GPIOx_INEN(x=A~I)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 PH, y=7 PI, y=8 0x00 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxINEN15	PxINEN14	PxINEN13	PxINEN12	PxINEN11	PxINEN10	PxINEN9	PxINEN8
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxINEN7	PxINEN6	PxINEN5	PxINEN4	PxINEN3	PxINEN2	PxINEN1	PxINEN0
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxINEN	GPIO 输入使能控制 0：关闭输入使能 1：打开输入使能

40.8.2 GPIO 上下拉使能寄存器（GPIOx_PUDEN）

名称	GPIOx_PUDEN (x=A~I)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG,y=6 PH, y=7 PI, y=8 0x04 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PxPDEN[15:8]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PxPDEN[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxPUEN [15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxPUEN [7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	PxPDEN	GPIO 下拉使能开关 0：关闭下拉使能 1：开启下拉使能
15:0	PxPUEN	GPIO 上拉使能开关 0：关闭上拉使能 1：开启上拉使能 弱上拉和弱下拉为 50K *注：PDPUEN[7]和 PDPUEN[8]复位值为 1，默认弱上拉（SWD）*

40.8.3 GPIO 开漏使能寄存器（GPIOx_ODEN）

名称	GPIOx_ODEN (x=A~I)
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG,y=6 PH, y=7 PI, y=8 0x08 + y*0x40

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PAODEN15	PAODEN14	PAODEN13	PAODEN12	PAODEN11	PAODEN10	PAODEN9	PAODEN8
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PAODEN7	PAODEN6	PAODEN5	PAODEN4	PAODEN3	PAODEN2	PAODEN1	PAODEN0
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxODEN	GPIO 开漏输出使能 0：关闭开漏输出 1：使能开漏输出

40.8.4 GPIO 功能选择寄存器（GPIOx_FCR）

名称	GPIOx_FCR (x=A~I)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG,y=6 PH, y=7 PI, y=8 0x0C + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	Px15FCR		Px14FCR		Px13FCR		Px12FCR	
位权限	R/W-0		R/W-0		R/W-0		R/W-0	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	Px11FCR		Px10FCR		Px9FCR		Px8FCR	
位权限	R/W-0		R/W-0		R/W-0		R/W-0	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	Px7FCR		Px6FCR		Px5FCR		Px4FCR	
位权限	R/W-0		R/W-0		R/W-0		R/W-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	Px3FCR		Px2FCR		Px1FCR		Px0FCR	
位权限	R/W-0		R/W-0		R/W-0		R/W-0	

Bit	助记符	功能描述
31:30	Px15FCR	Px[15]引脚功能选择 00：GPIO 输入

Bit	助记符	功能描述
		01：GPIO 输出 10：Digital function 11：Analog function
29:28	Px14FCR	Px[14]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
27:26	Px13FCR	Px[13]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
25:24	Px12FCR	Px[12]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
23:22	Px11FCR	Px[11]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
21:20	Px10FCR	Px[10]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
19:18	Px9FCR	Px[9]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
17:16	Px8FCR	Px[8]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function *注：PD8FCR 复位值为 10(SWD)*
15:14	Px7FCR	Px[7]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function *注：PD7FCR 复位值为 10(SWD)*
13:12	Px6FCR	Px[6]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function

Bit	助记符	功能描述
		11：Analog function
11:10	Px5FCR	Px[5]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
9:8	Px4FCR	Px[4]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
7:6	Px3FCR	Px[3]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
5:4	Px2FCR	Px[2]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
3:2	Px1FCR	Px[1]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
1:0	Px0FCR	Px[0]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function

40.8.5 GPIO 输出数据寄存器（GPIOx_DO）

名称	GPIOx_DO (x=A~I)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG,y=6 PH, y=7 PI, y=8 0x10 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxDO15	PxDO14	PxDO13	PxDO12	PxDO11	PxDO10	PxDO9	PxDO8
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxDO7	PxDO6	PxDO5	PxDO4	PxDO3	PxDO2	PxDO1	PxDO0
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxDO	GPIO output data register

40.8.6 GPIO 输出数据置位寄存器（GPIOx_DSET）

名称	GPIOx_DSET (x=A~I)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG,y=6 PH, y=7 PI, y=8 0x14 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxSET15	PxSET14	PxSET13	PxSET12	PxSET11	PxSET10	PxSET9	PxSET8
位权限	W	W	W	W	W	W	W	W
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxSET7	PxSET6	PxSET5	PxSET4	PxSET3	PxSET2	PxSET1	PxSET0
位权限	W	W	W	W	W	W	W	W

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxSET	GPIO output data set register 举例：向 PADSET 写 0x0000_8000，则 PADO[15]置位，其余位保持不变。

40.8.7 GPIO 输出数据复位寄存器（GPIOx_DRST）

名称	GPIOx_DRST (x=A~I)

offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG,y=6 PH, y=7 PI, y=8 0x18 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxRESET15	PxRESET14	PxRESET13	PxRESET12	PxRESET11	PxRESET10	PxRESET9	PxRESET8
位权限	W	W	W	W	W	W	W	W
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxRESET7	PxRESET6	PxRESET5	PxRESET4	PxRESET3	PxRESET2	PxRESET1	PxRESET0
位权限	W	W	W	W	W	W	W	W

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxRESET	GPIO output data reset register 举例：向 PADRST 写 0x0000_8000，则 PADO[15]清零，其余位保持不变

40.8.8 GPIO 输入数据寄存器（GPIOx_DIN）

名称	GPIOx_DIN (x=A~I)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG,y=6 PH, y=7 PI, y=8 0x1C + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxDIN15	PxDIN14	PxDIN13	PxDIN12	PxDIN11	PxDIN10	PxDIN9	PxDIN8

位权限	R	R	R	R	R	R	R	R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxDIN7	PxDIN6	PxDIN5	PxDIN4	PxDIN3	PxDIN2	PxDIN1	PxDIN0
位权限	R	R	R	R	R	R	R	R

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxDIN	Portx input data register 此寄存器仅占用地址空间，无物理实现。软件读此寄存器直接返回引脚输入信号，芯片并不对引脚输入进行锁存

40.8.9 GPIO 额外数字功能选择寄存器（GPIOx_DFS）

名称	GPIOx_DFS (x=A~I)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG,y=6 PH, y=7 PI, y=8 0x20 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PxDFS[31:24]							
位权限	R/W-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PxDFS[23:16]							
位权限	R/W-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxDFS[15:8]							
位权限	R/W-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxDFS[7:0]							
位权限	R/W-0							

Bit	助记符	功能描述
31:0	PxDFS	Portx Digital Function Select 对于具有多个数字外设功能的引脚，通过 PxDFS 寄存器可以选择使用哪个外设功能。 注意，对于不同的 IO 分组，有效的寄存器位置是不一样的，详细定义请参考**错误!未找到引用源。错误!未找到引用源。错误!未找到引用源。错误!未找到引用源。**

40.8.10 GPIO 驱动强度配置寄存器（GPIOx_DSR）

名称	GPIOx_DSR (x=A~I)

offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG,y=6 PH, y=7 PI, y=8 0x30 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PxDS[31:24]							
位权限	R/W-1010 1010							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PxDS[15:8]							
位权限	R/W-1010 1010							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxDS[15:8]							
位权限	R/W-1010 1010							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxDS[7:0]							
位权限	R/W-1010 1010							

Bit	助记符	功能描述
31:0	PxDS	GPIO 驱动强度配置，2bit 为一组控制 1 个 GPIO 比如 PxDS[31:30]用于控制 Px15 引脚，PxDS[7:6]用于控制 Px3 引脚 00：驱动关闭 01：驱动最弱 X1 10：驱动次强 X3 11：驱动最强 X4 上电后默认所有引脚驱动为（X3）

40.8.11 EXTI 边沿选择和使能寄存器 A~I（GPIO_EXTIA/B/…/I）

名称	GPIO_EXTIA/B/…/I							
offset	0x240~0x260							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTI15_EDS		EXTI14_EDS		EXTI13_EDS		EXTI12_EDS	
位权限	R/W-11		R/W-11		R/W-11		R/W-11	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EXTI11_EDS		EXTI10_EDS		EXTI9_EDS		EXTI8_EDS	
位权限	R/W-11		R/W-11		R/W-11		R/W-11	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI7_EDS		EXTI6_EDS		EXTI5_EDS		EXTI4_EDS	
位权限	R/W-11		R/W-11		R/W-11		R/W-11	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI3_EDS		EXTI2_EDS		EXTI1_EDS		EXTI0_EDS	

位权限	R/W-11	R/W-11	R/W-11	R/W-11

Bit	助记符	功能描述
31:30	EXTI15_EDS	EXTI[15]边缘触发选择 00: rising 01: falling 10: both 11: disable
…	…	
1:0	EXTI0_EDS	EXTI1[0] 边缘触发选择 00: rising 01: falling 10: both 11: disable

40.8.12 EXTI 数字滤波控制寄存器 AB/CD/EF/GH/I（GPIO_EXTIDFAB/CD/EF/GH/I）

名称	GPIO_EXTIDFAB/CD/EF/GH/I							
offset	0x264~0x274							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTI_DF[31:24]							
位权限	R/W -0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EXTI_DF[23:16]							
位权限	R/W -0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_DF[15:8]							
位权限	R/W -0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI_DF[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	EXTI_DF	EXTI 输入数字滤波功能使能 0：关闭 EXTI 数字滤波 1：使能 EXTI 数字滤波 GPIO_EXTIDFAB[31:16]对应 GPIOB 组 GPIO_EXTIDFAB[15:0]对应 GPIOA 组 GPIO_EXTIDFCD[31:16]对应 GPIOD 组 GPIO_EXTIDFCD[15:0]对应 GPIOC 组 GPIO_EXTIDFEF[31:16]对应 GPIOF 组 GPIO_EXTIDFEF[15:0]对应 GPIOE 组 GPIO_EXTIDFGH[31:16]对应 GPIOH 组 GPIO_EXTIDFGH[15:0]对应 GPIOG 组 GPIO_EXTIDFI[31:16]无效 GPIO_EXTIDFI[15:0]对应 GPIOI 组

40.8.13 EXTI 中断标志寄存器（GPIO_EXTIFAB/CD/EF/GH/I）

名称	GPIO_EXTIFAB/CD/EF/GH/I							
offset	0x278~0x288							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTI[31:24]							
位权限	R/W-00 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EXTI[23:16]							
位权限	R/W-00 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	EXTI	外部引脚中断标志寄存器，每个引脚对应 1 个引脚中断 硬件置位，软件写 1 清零 GPIO_EXTIAB[31:16]对应 GPIOB 组 GPIO_EXTIAB[15:0]对应 GPIOA 组 GPIO_EXTICD[31:16]对应 GPIOD 组 GPIO_EXTICD[15:0]对应 GPIOC 组 GPIO_EXTIEF[31:16]对应 GPIOF 组 GPIO_EXTIEF[15:0]对应 GPIOE 组 GPIO_EXTIGH[31:16]对应 GPIOH 组 GPIO_EXTIGH[15:0]对应 GPIOG 组 GPIO_EXTII[31:16]无效 GPIO_EXTII[15:0]对应 GPIOI 组

40.8.14 EXTI 输入信号寄存器（GPIO_EXTIDIAB/CD/EF/GH/I）

名称	GPIO_EXTIDIAB/CD/EF/GH/I							
offset	0x28C~0x29C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTI_DI[31:24]							
位权限	R/W-00 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EXTI_DI[23:16]							
位权限	R/W-00 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_DI[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI_DI[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	EXTI_DI	输入数据寄存器 GPIO_EXTIDIAB[31:16]对应 GPIOB 组 GPIO_EXTIDIAB[15:0]对应 GPIOA 组 GPIO_EXTIDICD[31:16]对应 GPIOD 组 GPIO_EXTIDICD[15:0]对应 GPIOC 组 GPIO_EXTIDIEF[31:16]对应 GPIOF 组 GPIO_EXTIDIEF[15:0]对应 GPIOE 组 GPIO_EXTIDIGH[31:16]对应 GPIOH 组 GPIO_EXTIDIGH[15:0]对应 GPIOG 组 GPIO_EXTIDII[31:16]无效 GPIO_EXTIDII[15:0]对应 GPIOI 组

40.8.15 FOUT 配置寄存器（GPIO_FOUTSEL）

名称	GPIO_FOUTSEL							
offset	0x2C0							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	FOUT1SEL							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FOUT0SEL							
位权限	R/W-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:8	FOUT1SEL	PB12 输出选择 8'h00 : XTLF 8'h01 : RCLP 8'h02 : ADC_CLK 8'h03 : LSCLK 8'h04 : ADC_EOC 8'h05 : RTCTM 8'h06 : PLL1X/64 8'h07 : ADC_EOCCAL 8'h08 : APBCLK1/64 8'h09 : PLL1X 8'h0A : CAN_TXCLK 8'h0B : RCHF 8'h0C : XTHF/64 8'h0D : ADC_CLK/64 8'h0E : CLK8K 8'h0F : XTHF

Bit	助记符	功能描述
		8'h10 : APBCLK2/64 8'h11 : PLLO_VCO/64 others : RFU
7:0	FOUT0SEL	PE7/PD11 输出选择 8'h00 : XTLF 8'h01 : RCLP 8'h02 : RCHF/64 8'h03 : LSCLK 8'h04 : AHBCLK/64 8'h05 : RTCTM 8'h06 : PLL1X/64 8'h07 : RTCCLK64Hz 8'h08 : APBCLK1/64 8'h09 : PLL1X 8'h0A : CAN_TXCLK 8'h0B : RCHF 8'h0C : XTHF/64 8'h0D : PLL_VCO/64 8'h0E : CLK8K 8'h0F : ADC_CLK 8'h10 : APBCLK2/64 Others :RFU

40.8.16 WKUP 控制寄存器（GPIO_PINWKEN）

名称	GPIO_PINWKEN							
offset	0x31C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	WKSEL		PINWKSEL					
位权限	R/W-0		R/W-000000					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PINWKSEL							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PINWKSEL						PINWKEN	
位权限	R/W-0000 0000						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PINWKEN							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31	WKSEL	WKUPx 唤醒中断入口选择 0：WKUPx 中断为 NMI 中断 1：WKUPx 中断为 39 号中断 【注】如果需要实现唤醒后不进入 ISR，直接运行主程序，可以将 WKSEL 设为 1，并置位 PRIMASK 寄存器后进入休眠；当 WKUPx 事件到来后，芯片会退出休眠模式，但是由于 PRIMASK=1，不会进入中断服务程序。
30	--	RFU：未实现，读为 0
29:10	PINWKSEL	WKUP 边沿选择

Bit	助记符	功能描述
		00：对应的 WKUP 引脚为下降沿唤醒 01：对应的 WKUP 引脚为上升沿唤醒 10/11：对应的 WKUP 引脚为上升、下降沿都能唤醒 寄存器对应顺序： Bit[11:10]对应 WKUP0 Bit[13:12]对应 WKUP1 Bit[15:14]对应 WKUP2 Bit[17:16]对应 WKUP3 Bit[19:18]对应 WKUP4 Bit[21:20]对应 WKUP5 Bit[23:22]对应 WKUP6 Bit[25:24]对应 WKUP7 Bit[27:26]对应 WKUP8 Bit[29:28]对应 WKUP9
9:0	PINWKEN	WKUP 引脚使能信号 1：对应的 WKUP 引脚功能有效 0：对应的 WKUP 引脚功能无效 PINWKEN[x]控制 WKUPx 引脚的使能

41 安全功能（SFU）

41.1 概述

本芯片内置了以下安全功能，目的是通过芯片自诊断设计，在故障和异常发生时，及时的停止工作。

- 闪存内容 CRC 检查（通过 DMA 实现，检查时 CPU 可以在 RAM 中运行程序代码）
- RAM 自检
- 代码闪存 ECC，SEC-DED
- 数据闪存 ECC，SEC-DED
- RAM ECC 校验，SEC-DED
- MPU
- RAM 数据写保护
- 关键寄存器写保护
- 非法地址访问检测
- 时钟频率检测
- 晶振停振检测
- 电源电压检测
- IO 输出状态回读
- LIN 自测试
- CAN 自测试
- ADC 自测试
- DAC 自测试
- 比较器自测试
- 温度检测

41.2 闪存 CRC 检查

使用 DMA 实现对闪存内容的自检。为了不影响程序执行，启动闪存自检前应在 RAM 中执行代码，然后将 DMA 通道触发配置为 CRC，将源地址指向代码闪存，DMA 启动后，硬件自动将闪存内的数据搬运至 CRC 模块进行校验，完成全部闪存检验后，程序读取 CRC 结果并与预期值比较，以判断闪存内容是否存在问题。闪存检查的长度由 DMA 的 CHxTSIZE 寄存器控制，最大支持 512KB 闪存的全空间校验，用于检查程序完整性。

41.3 RAM 自检

系统 RAM 可以通过 RAMBIST 电路实现自检。

CAN-FD 的 messageRAM 在上电后执行硬件自检，参见 CAN-FD 章节。

41.4 Flash ECC

程序 flash 和数据 flash 都支持 ECC（SEC-DED），参见总线与存储章节。

41.5 RAM ECC

系统 RAM 支持 ECC，可以实现 1bit 纠错和 2bit 检错（SEC-DED）。详情参见总线与存储章节。

41.6 MPU

芯片处理器内核支持 MPU，最大 8 个 region，可以实现进程之间的数据隔离和访问保护，参见 CPU 章节中的 MPU 部分。

41.7 RAM 写保护

为了满足 IEC61508 的技术要求，RAM 写保护功能用于防止在 CPU 失控时改写 RAM 中的关键数据。RAM 写保护最多覆盖 4KB RAM 空间，以 1KB 为颗粒度，被保护的空间不允许写入，只允许读出。

注意：RAM 写保护使能后，CPU、DMA、RAMBIST 都不能改写被保护地址。

41.8 关键寄存器写保护

为了保证运行过程中的安全，IEC61508 标准要求即使 CPU 失控也需要保护关键的控制寄存器，使其免遭改写。寄存器写保护用于保护 GPIO 控制寄存器、时钟控制寄存器、下电复位和电源检测控制寄存器、RAM 校验控制寄存器。

当写保护使能后，被保护的寄存器不能改写，但是仍可以正常读取。

支持写保护的模块寄存器：

- 比较器控制寄存器
- SVD 控制寄存器和配置寄存器
- CMU 系统时钟配置寄存器、外设时钟配置寄存器
- RMU 配置寄存器
- GPIO 控制寄存器
- IWDT 配置寄存器和窗口寄存器
- WWDT 配置寄存器和预分频寄存器
- Flash 控制寄存器

参见存储访问保护控制寄存器（SFU_MAPCR）

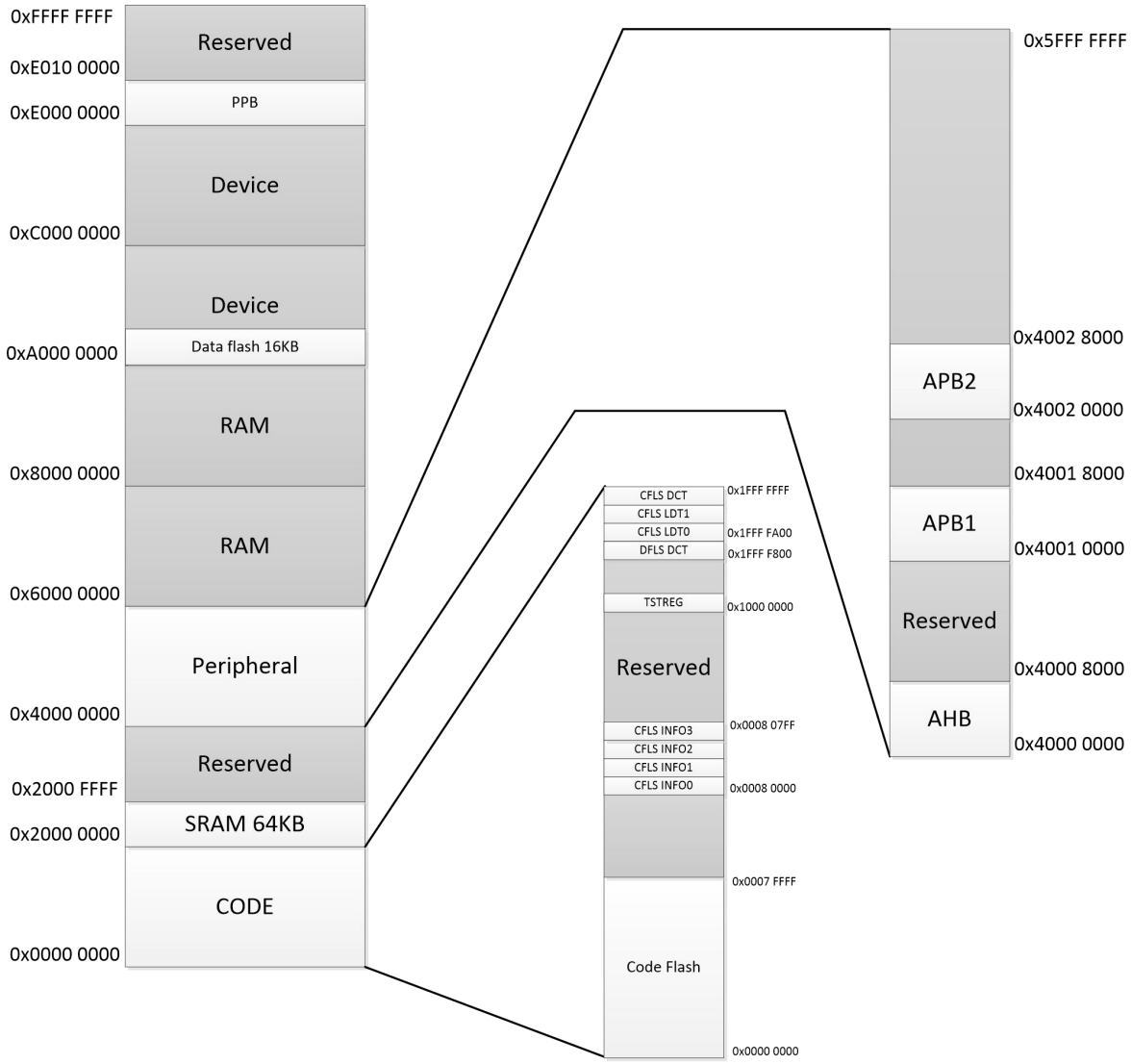
保护控制寄存器位	受保护模块	受保护寄存器
CMP1P	模拟比较器	COMP1_CR COMP2_CR COMP3_CR COMP4_CR COMP_ICR COMP_BUFCR
WWP	窗口看门狗	WWDT_CFGR
IWP	独立看门狗	IWDT_CR IWDT_WIN
NVMP	Flash 控制寄存器	FLS_RDCR
RCP	复位管理单元	RMU_PDRCFG RMU_BORCFG
SVDP	电源电压监测	SVD_CFGR SVD_CR SVD_IER SVD_VSR
CCP	时钟管理单元	CMU_SCCR CMU_RCHFCR CMU_RCHTR CMU_PLLCR CMU_RCLPTR CMU_XTLFCR CMU_XTHFCR CMU_IER
CLM0P	时钟监控模块 0	CLM0_CR CLM0_CFGR CLM0_CMPH CLM0_CMPL
CLM1P	时钟监控模块 1	CLM1_CR CLM1_CFGR CLM1_CMPH CLM1_CMPL
GPIOAP	GPIOA 组	GPIOA_INEN GPIOA_PUDEN

		GPIOA_ODEN GPIOA_FCR GPIOA_DFS GPIOA_DSR
GPIOBP	GPIOB 组	GPIOB_INEN GPIOB_PUDEN GPIOB_ODEN GPIOB_FCR GPIOB_DFS GPIOB_DSR
GPIOCP	GPIOC 组	GPIOC_INEN GPIOC_PUDEN GPIOC_ODEN GPIOC_FCR GPIOC_DFS GPIOC_DSR
GPIODP	GPIOD 组	GPIOD_INEN GPIOD_PUDEN GPIOD_ODEN GPIOD_FCR GPIOD_DFS GPIOD_DSR
GPIOEP	GPIOE 组	GPIOE_INEN GPIOE_PUDEN GPIOE_ODEN GPIOE_FCR GPIOE_DFS GPIOE_DSR
GPIOFP	GPIOF 组	GPIOF_INEN GPIOF_PUDEN GPIOF_ODEN GPIOF_FCR GPIOF_DFS GPIOF_DSR
GPIOGP	GPIOG 组	GPIOG_INEN GPIOG_PUDEN GPIOG_ODEN GPIOG_FCR GPIOG_DFS GPIOG_DSR
GPIOHP	GPIOH 组	GPIOH_INEN GPIOH_PUDEN GPIOH_ODEN GPIOH_FCR GPIOH_DFS GPIOH_DSR
GPIOIP	GPIOI 组	GPIOI_INEN GPIOI_PUDEN GPIOI_ODEN GPIOI_FCR GPIOI_DFS GPIOI_DSR
EXTIP	EXTI 配置寄存器	GPIO_EXTIA/B/…/H GPIO_EXTIDF
WKP	WKUP 控制寄存器	GPIO_PINWKEN

41.9 非法地址访问检测

IEC60730 标准强制要求检查 CPU 和中断功能的运行是否正常，如果发现程序访问了系统中的空白地址，则认为程序执行异常，可以根据 MAP_SEL 寄存器配置产生异常访问中断（NMI 中断）或者复位。非法地址访问功能通过 MAP_EN 寄存器来使能或关闭。当软件对 MAP_EN 置位后，无法再清零，直到下一次芯片复位。

下图中，灰色地址区域禁止任何形式的访问（禁止读、写、取指），白色地址区域则具有不同的访问权限控制，详情参见后续表格。注意，此非法地址访问检测不区分 CPU 或 DMA 发起的访问。



地址访问权限表（FM33FG06x）

属性	地址范围	Read	Write	Fetch

	min	Max			
Code Flash	0x00000000	0x0007FFFF	O	O	O
Data Flash	0xA0000000	0xA003FFFF	O	O	X
Info Flash	0x00080000	0x000807FF	O	O	X
Vendor Reserved Flash	0x1FFFFA00	0x1FFFFFFF	O	X	X
SRAM	0x20000000	0x2000FFFF	O	O	O
Peripheral AHB	0x40000000	0x40007FFF	O	O	X
Peripheral APB1	0x40010000	0x4001FFFF	O	O	X
Peripheral APB2	0x40020000	0x4002FFFF	O	O	X
Others	-	-	X	X	X

表 41-1512K flash 版本总线访问控制表

地址访问权限表（FM33FG04x）

属性	地址范围		Read	Write	Fetch
	min	Max			
Code Flash	0x00000000	0x0003FFFF	O	O	O
Data Flash	0xA0000000	0xA001FFFF	O	O	X
Info Flash	0x00080000	0x000807FF	O	O	X
Vendor Reserved Flash	0x1FFFFA00	0x1FFFFFFF	O	X	X
SRAM	0x20000000	0x20007FFF	O	O	O
Peripheral AHB	0x40000000	0x40007FFF	O	O	X
Peripheral APB1	0x40010000	0x4001FFFF	O	O	X
Peripheral APB2	0x40020000	0x4002FFFF	O	O	X
Others	-	-	X	X	X

表 41-2 256K flash 版本总线访问控制表

41.10 时钟频率检测

IEC60730 标准强制要求检查系统时钟振荡频率是否正常。

CLM 模块用于实现时钟频率互检，参见 CLM 章节。

通过多个定时器也可以实现类似功能。比如使用 LPTIM16，将其工作时钟配置为 APBCLK，将通道 1 输入捕获目标设置为 RCLP（32Khz），通过捕捉 RCLP 的周期，可以测量系统时钟频率是否正常。

41.11 晶振停振检测

芯片的高频晶振和低频晶振各自带有停振检测电路，当晶振停振时，可根据配置产生报警中断或者复位。

同时，如果芯片正在使用晶振作为系统时钟，停振检测电路发现晶体停振时，芯片会自动将系统时钟切换为内部环振，避免系统死机。如果使能了停振（Lost Of Clock）复位功能，则直接复位芯片。

41.12 PLL 失锁检测

芯片的 PLL 带有锁定检测电路，当 PLL 失锁时，可根据配置产生报警中断或者复位。

同时，如果芯片正在使用 PLL 作为系统时钟，锁定检测电路发现 PLL 失锁时，芯片会自动将系统时钟切换为内部环振，避免系统死机。如果使能了失锁（Lost Of Lock）复位功能，则直接复位芯片。

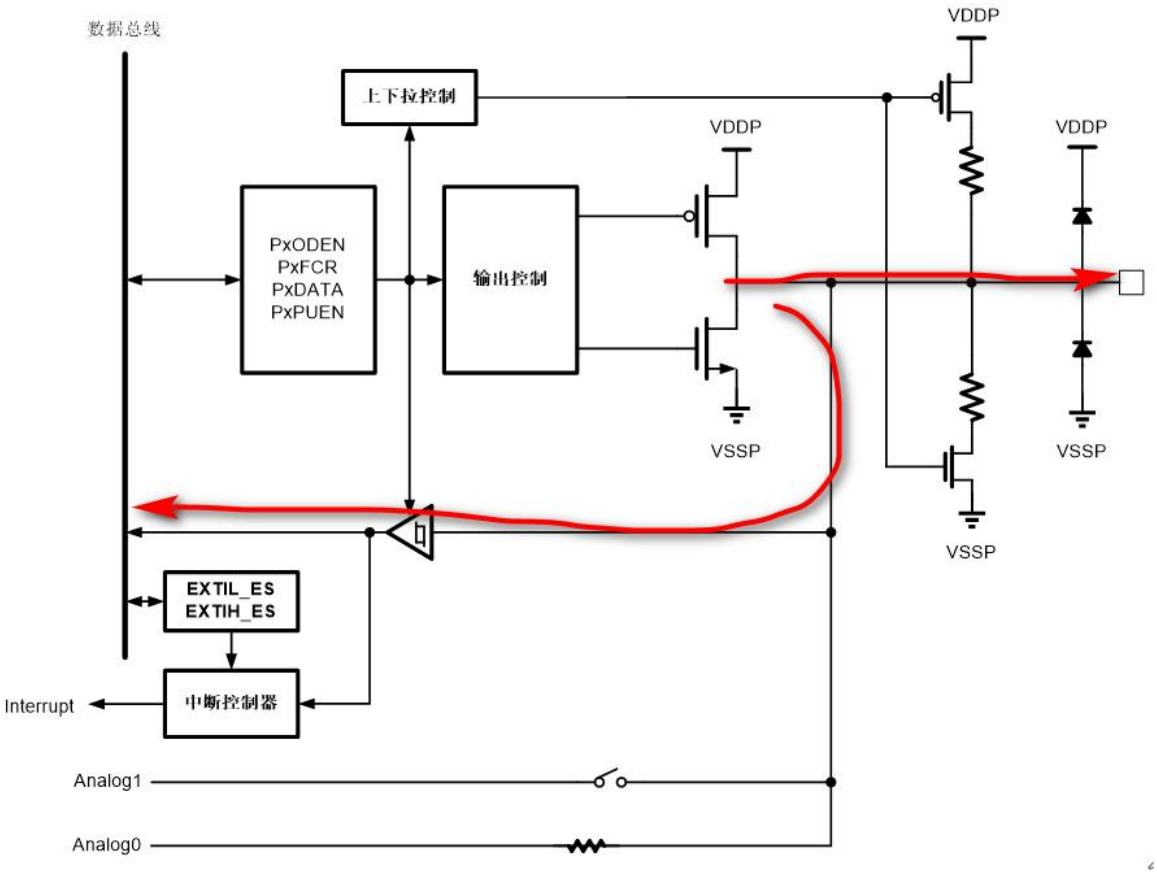
41.13 电源电压检测

芯片可以通过 SVD 电路、BOR 电路和 PDR 电路实现多重电源检测和欠压复位保护。

请参见 SVD、RMU 章节。

41.14 IO 输出状态回读

GPIO 在输出信号时（外设输出或者 IO 输出），软件可以从输入数据寄存器回读 IO 状态，如下图所示。



41.15 LIN 自测试

参见 UART 章节中的 LIN 子章节

41.16 CAN 自测试

FSCAN 和 CAN-FD 都支持环回测试（loopback），参见 FSCAN 和 CAN-FD 章节。

41.17 ADC 测试

IEC60730 标准强制要求进行 A/D 转换器的测试。此 A/D 测试功能通过对 A/D 转换器的正（+）基准电压、负（–）基准电压、模拟输入通道（ADC_Inx）、温度传感器输出电压以及内部基准电压进行 A/D 转换来确认 A/D 转换器的正常运行。

通过以下步骤确认模拟多路转换器：

1.选择 ADC_INx 引脚作为 A/D 转换对象

2.对 ADC_INx 引脚进行 A/D 转换（转换结果 1-1）

3.选择 A/D 转换器的负（– ）基准电压作为 A/D 转换对象

4.对 A/D 转换器的负（–）基准电压进行 A/D 转换（转换结果 2-1）

5.选择 ADC_INx 引脚作为 A/D 转换对象

6.对 ADC_INx 引脚进行 A/D 转换（转换结果 1-2）

7.选择 A/D 转换器的正（+）基准电压作为 A/D 转换对象

8.对 A/D 转换器的正（+）基准电压进行 A/D 转换（转换结果 2-2）

9.选择 ADC_INx 引脚作为 A/D 转换对象

10.对 ADC_INx 引脚进行 A/D 转换（转换结果 1-3）

11.确认“转换结果 1-1”=“转换结果 1-2”=“转换结果 1-3”

12.确认“转换结果 2-1”的 A/D 转换结果全部为“0”，“转换结果 2-2”的 A/D 转换结果全部为“1”。

通过以上步骤，能确认已选择模拟多路转换器以及布线没有断线。

41.18 DAC 自测试

DAC 输出可以连接到 ADC 输入，通过 ADC 采样 DAC 输出，可以判断 DAC 是否工作正常。

41.19 比较器自测试

通过以下方法实现比较器自测试：

- 比较器正端连接 DAC 输出
- 比较器负端连接内部基准电压
- 软件配置 DAC 输出一个低于基准的电压，此时比较器输出为 0
- 软件配置 DAC 输出一个高于基准的电压，此时比较器输出为 1
- 由于 DAC 的参考电压是 VDD，所以执行测试前，需要明确当前芯片的 VDD 电压，才能正确计算 DAC 的合理码值

41.20 温度检测

芯片内置温度传感器，通过 ADC 采样温度传感器输出，可以得到芯片当前的工作结温，以此大致判断芯片所处的环境温度。

当结温临界，甚至高于芯片允许的工作温度范围时，软件应及时进入安全状态，措施包括但不限于：降低 CPU 工作频率，关闭非必须外设，关闭引脚输出避免抽灌电流，芯片主动停止工作等。

41.21 寄存器

安全功能模块基地址：0x4000_1C00

offset 地址	名称	符号
0x00	-	-
0x04	-	-
0x08	存储器访问保护控制寄存器 (Memory Access Protection Control Register)	SFU_MAPCR
0x0C	存储器访问保护状态寄存器 (Memory Access Protection Control Register)	SFU_MAPSR

41.21.1 存储访问保护控制寄存器（SFU_MAPCR）

名称	SFU_MAP_CR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-					WKP	EXTIP	GPIOIP
位权限	U-0					R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	GPIOHP	GPIOGP	GPIOFP	GPIOEP	GPIODP	GPIOCP	GPIOBP	GPIOAP
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		CLM1P	CLM0P	CMPP	WWP	IWP	NVMP
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MAP_SEL	RCP	SVDP	CCP	-	RAMWP		MAP_EN
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-00		R/W-0

Bit	助记符	功能描述
31:27	--	RFU：未实现，读为 0
26	WKP	WKUP 配置寄存器写保护
25	EXTIP	EXTI 配置寄存器写保护
24	GPIOIP	GPIO I 组寄存器写保护
23	GPIOHP	GPIO H 组寄存器写保护
22	GPIOGP	GPIO G 组寄存器写保护
21	GPIOFP	GPIO F 组寄存器写保护
20	GPIOEP	GPIO E 组寄存器写保护
19	GPIODP	GPIO D 组寄存器写保护
18	GPIOCP	GPIO C 组寄存器写保护
17	GPIOBP	GPIO B 组寄存器写保护
16	GPIOAP	GPIO A 组寄存器写保护
15:14	--	RFU：未实现，读为 0
13	CLM1P	CLM1 寄存器写保护 1：CLM1 控制寄存器不可改写

Bit	助记符	功能描述
		0：无保护
12	CLM0P	CLM0 寄存器写保护 1：CLM0 控制寄存器不可改写 0：无保护
11	CMPP	比较器寄存器保护 1：比较器控制寄存器不可改写 0：无保护
10	WWP	WWDT 看门狗寄存器保护 1：WWDT 配置寄存器不可改写 0：无保护
9	IWP	IWDT 看门狗寄存器保护 1：IWDT 配置寄存器不可改写 0：无保护
8	NVMP	Flash 控制寄存器保护 1：Flash 控制寄存器不可改写 0：无保护
7	MAP_SEL	非法地址访问发生时的系统行为配置 (Memory Access Protection Select) 1：非法地址访问发生时，复位芯片 0：非法地址访问发生时，产生 NMI
6	RCP	RMU 控制寄存器保护(Reset Control Protection) 1：RMU 模块的配置寄存器不可改写 0：无保护
5	SVDP	SVD 控制寄存器保护(SVD and BOR Protection) 1：SVD 控制寄存器不可改写 0：无保护
4	CCP	CMU 控制寄存器保护(Clock Control Protection) 1：CMU 模块的停振检测控制、系统时钟配置、时钟调校寄存器不可改写 0：无保护
3	--	RFU：未实现，读为 0
2:1	RAMWP	RAM 写保护(RAM Write Protection) 00：无写保护 01：RAM 起始地址开始的 1K 字节禁止写入 10：RAM 起始地址开始的 2K 字节禁止写入 11：RAM 起始地址开始的 4K 字节禁止写入
0	MAP_EN	存储器访问保护使能(Memory Access Protection Enable) 1：使能非法地址访问检测功能 0：不使能非法地址访问检测 *注意：此寄存器一旦置位，软件不能对其清零。*

41.21.2 存储访问保护状态寄存器（SFU_MAPSR）

名称	SFU_MAPSR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						-	MAP_IF
位权限	U-0						U-0	R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	MAP_IF	存储器访问保护中断标志 (Memory Access Protection Interrupt flag)，硬件置位，软件写 1 清零。此标志置位后会触发 NMI 中断。 1：发生了非法地址访问 0：未发生非法地址访问

42 专用编程接口

42.1 概述

FM33FG0A芯片可使用复旦微电子所提供的专用编程器，或者通过Bootloader下载用户程序。编程器通过专用编程接口(SWD)与芯片通信，完成程序下载，并可对Flash全空间内容进行Checksum校验。

42.2 编程器使用

编程器的使用方法请参考应用手册，或联系复旦微电子公司。

43 调试支持

43.1 概述

FM33FG0A芯片基于ARM Cortex-M0处理器构建，并支持相应的debug特性。通过硬件断点（breakpoint）和数据观察点（watchpoint），调试器可以在特定指令取指和数据访问时停止CPU内核运行，检视内核寄存器和系统外设状态，并根据需要恢复内核运行。

仿真调试主机通过SWD接口与FM33FG0A芯片互联，并实现仿真调试。

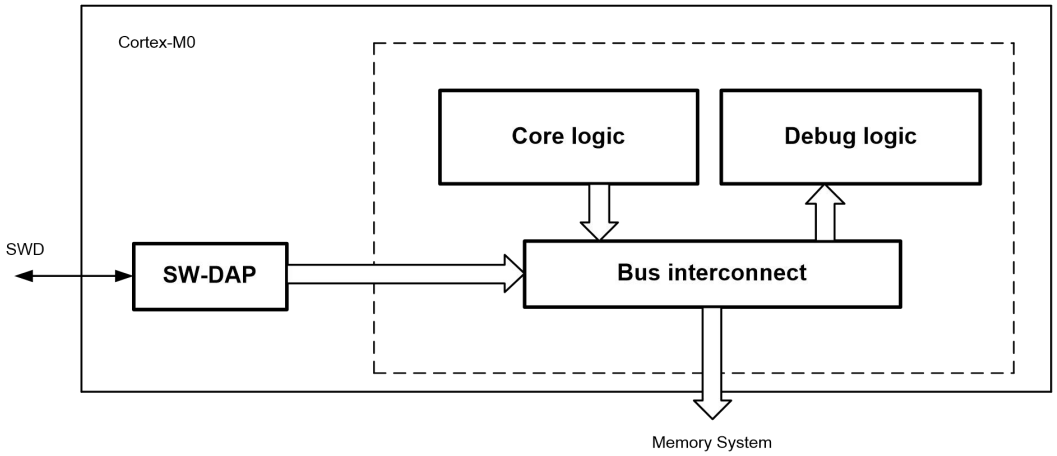


图 43-1 Cortex-M0 调试系统示意图

关于Cortex-M0内核的debug特性，请参考ARM公司的Cortex-M0技术参考手册。

43.2 Debug 引脚

43.2.1 SWD 引脚

FM33FG0A系列MCU的SWD引脚位置如下表：

SWD pins	Debug功能	引脚定义
SWDIO	SWD数据输入/输出	PD8
SWCLK	SWD时钟输入	PD7

注意：芯片复位后PD8和PD7都默认为输入状态，与大部分GPIO不同。

43.2.2 上拉电阻

芯片复位后，SWDIO和SWCLK引脚默认使能内部上拉（约50K欧姆），用户无需在PCB上外接上拉电阻。

43.3 SWD 接口协议

43.3.1 协议简介

SWD协议采用LSB-first进行数据收发。通过SWD接口，调试主机可以读写DPACC和APACC寄存器组。

SWIO每次切换数据方向时，总线上需插入turn-around时间，这段时间内主机和从机都不会驱动SWIO。在两次传输之间，主机必须将线驱动为低电平进入idle状态，或继续发送一次新传输的起始位继续传输，在一次数据包传输之后，主机也可以空闲，使线保持为高电平或由上拉电阻上拉。SWD协议没有明确的复位信号，在没有看到预期的信号时，主机或目标机将对复位进行检测。通过保持线为高电平持续50个时钟周期之后跟随一个读ID的请求，可以确保在检测到错误或复位之后重新同步成功。

43.3.2 传输序列

SWD每个通信传输序列包含三个部分：

1、包请求（8bits），由主机发送

2、ACK响应（3bits），由从机回发

3、数据传输（33bits），由主机或从机发送

其中包请求字节定义如下：

Bit	Name	描述
0	Start	起始位，必须是1
1	ApnDP	AP/DP选择 0：DP访问 1：AP访问
2	RnW	读写选择 0：写请求 1：读请求
4:3	A[3:2]	DP/AP寄存器的地址域
5	Parity	Bit0~Bit4数据的校验位
6	Stop	0
7	Park	主机不驱动，通过总线上拉，从机读为1

包请求发送后，总线上总是有1bit的turn-around时间。

ACK响应定义如下：

Bit	Name	描述
0:2	ACK	001：FAULT 010：WAIT 100：OK

如果主机发起读操作，或者ACK为WAIT或FAULT，则ACK之后必须插入turn-around时间。

数据传输格式如下：

Bit	Name	描述
0:31	Data	读出或写入的数据
32	Parity	32bit 数据的校验位

43.3.3 SW-DP ID code

Cortex-M0的SW-DP有一个固定的ID code：0x0BB11477

SW-DP处于非活跃状态，直到主机读取ID code。

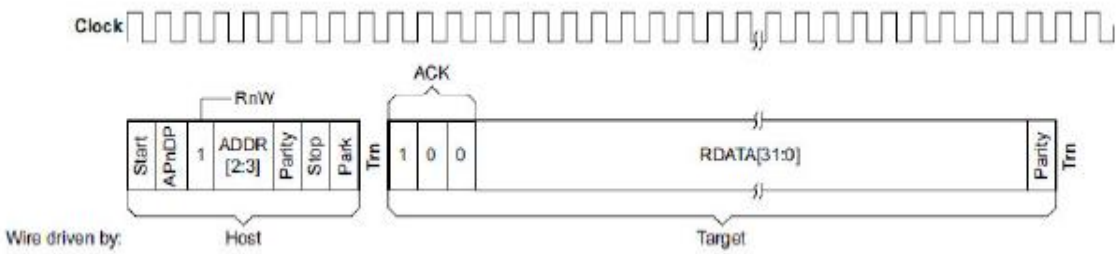
- 芯片复位，或者SWIO拉高50个SWCLK周期后，SW-DP处于RESET状态
- 拉低SWIO至少2个SWCLK周期后，SW-DP进入IDLE状态
- 当SW-DP处于RESET，主机必须先使其进入IDLE，然后对ID code寄存器进行读操作，才能激活SW-DP。否则从机会对主机的通信回应FAULT响应。

43.3.4 主机读操作

一次成功的读操作由以下三个阶段组成

- 一个8位的读数据包请求（request），从主机到目标。
- 一个3位的应答（ack），从目标到主机。成功的OK响应为100，WAIT响应为010，FAULT响应为001.
- 一个33位的数据读阶段（payload），从主机到目标。

默认情况下，在第一和第二阶段之间以及第三阶段之后有一个时钟的掉转周期，一次成功的读操作如下图。

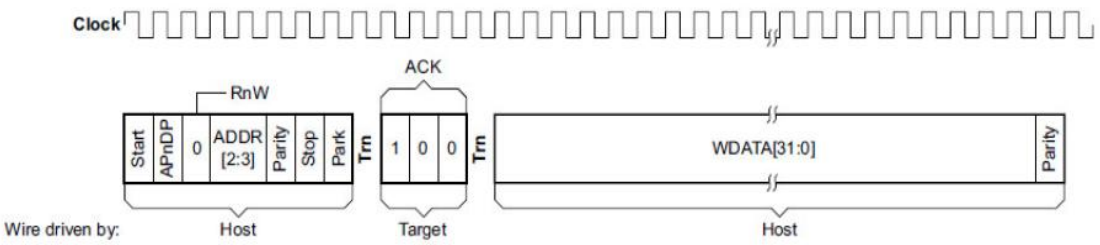


43.3.5 主机写操作

一次写操作由以下三个阶段组成

- 一个8位的写数据包请求（header），从主机到目标。
- 一个3位的应答（ack），从目标到主机。成功的OK响应为100，FAULT响应为001.
- 一个33位的数据写阶段（payload），从主机到目标。

默认情况下，每两个阶段之间都有一个时钟的掉转周期，一次成功的写操作如下图。



43.4 SWD-DP 寄存器

43.4.1 寄存器列表

Address (A[3:2])	DPBANKSEL	Name	Access
00	x	DHCSR	RO
		ABORT	WO
01	0x0	CTRL/STAT	RW
	0x1	DLCR	RW
10	x	RESEND	RO
		SELECT	WO
11	x	RDBUFF	RO

关于寄存器的详细说明，请参考Cortex-M0 Technical Reference Manual.

43.5 Core debug 寄存器

通过操作core debug寄存器可以实现内核调试。主机通过SW-DP访问以下内核调试寄存器。

Address	Name	Type	Function
0xE000EDF0	DHCSR	RW	Debug Halting Control and Status Register
0xE000EDF4	DCRSR	WO	Debug Core Register Selector Register
0xE000EDF8	DCRDR	RW	Debug Core Register Data Register
0xE000EDFC	DEMCR	RW	Debug Exception and Monitor Control Register
0xE000EE00 to 0xE000EEFF	-	-	Reserved for Debug Extension

上述debug寄存器不被系统复位影响，仅受上电复位影响。通过以下方式可以实现CPU复位后立即halt：

- 置位DEMCR寄存器的bit0 (VC_CORRESET)
- 置位DHCSR寄存器的bit0 (C_DEBUGEN)
- 执行系统复位

43.6 低功耗调试支持

通常情况下，当芯片进入Sleep/DeepSleep模式时，CPU的FCLK和HCLK都会被关闭，这样会导致host无法保持与芯片的debugger连接。为了支持低功耗模式下的debug调试，当CPU连接debugger时，FCLK和HCLK必须保持运行，即此时芯片不会真正进入休眠，调试中应注意这一点。

43.7 Debug 相关的配置项

通过配置DBG_CR寄存器，可以设置在调试状态下，芯片内部的定时器、看门狗电路是否继续工作。详情请参见MCU DEBUG配置寄存器。

43.8 寄存器

地址	名称	符号
0x40000004	MCU Debug 配置寄存器	DBG_CR
0x40000008	HARDFAULT 查询寄存器	HDFR

43.8.1 MCU DEBUG 配置寄存器

FM33FG0A 扩展了 MCUDBGCR 寄存器，用于配置 Debug 状态下的看门狗和定时器。MCUDBGCR 寄存器可以由 SWD 接口或软件改写。

名称	DBG_CR							
地址	0x40000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						DBG_RCEN	DBG_SLEEP
位权限	U-0						R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DBG_LPT32_STOP	DBG_LPT16_STOP	-	-	-	-	DBG_BT32_STOP	DBG_BT16_STOP
位权限	R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DBG_GPT2_STOP	DBG_GPT1_STOP	DBG_GPT0_STOP	DBG_ATIM_STOP	DBG_TAU1_STOP	DBG_TAU0_STOP	DBG_WWDT_STOP	DBG_IWDT_STOP
位权限	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

Bit	助记符	功能描述
31:18	--	RFU：未实现，读为 0
17	DBG_RCEN	Debug 模式下是否支持标志寄存器读清零操作 0：debug 禁止标志读清零操作 1：debug 允许标志读清零操作
16	DBG_SLEEP	休眠模式下的 debug 配置 0：正常进入休眠，关闭 HCLK 和 FCLK，debugger 无法保持连接 1：不关闭 HCLK 和 FCLK，debugger 能够在休眠模式下保持连接
15	DBG_LPT3	Debug 状态下 LPTIM32 使能控制位

Bit	助记符	功能描述
	2_STOP	1：Debug 时关闭 LPTIM32 0：Debug 时保持 LPTIM32 原来状态
14	DBG_LPT16_STOP	Debug 状态下 LPTIM16 使能控制位 1：Debug 时关闭 LPTIM16 0：Debug 时保持 LPTIM16 原来状态
13:10	--	RFU：未实现，读为 0
9	DBG_BT32_STOP	Debug 状态下 BSTIM32 使能控制位 1：Debug 时关闭 BSTIM32 0：Debug 时保持 BSTIM32 原来状态
8	DBG_BT16_STOP	Debug 状态下 BSTIM16 使能控制位 1：Debug 时关闭 BSTIM16 0：Debug 时保持 BSTIM16 原来状态
7	DBG_GPT2_STOP	Debug 状态下 GPTIM2 使能控制位 1：Debug 时关闭 GPTIM2 0：Debug 时保持 GPTIM2 原来状态
6	DBG_GPT1_STOP	Debug 状态下 GPTIM1 使能控制位 1：Debug 时关闭 GPTIM1 0：Debug 时保持 GPTIM1 原来状态
5	DBG_GPT0_STOP	Debug 状态下 GPTIM0 使能控制位 1：Debug 时关闭 GPTIM0 0：Debug 时保持 GPTIM0 原来状态
4	DBG_ATIM_STOP	Debug 状态下 ATIM 使能控制位 1：Debug 时关闭 ATIM 0：Debug 时保持 ATIM 原来状态
3	DBG_TAU1_STOP	Debug 状态下 TAU1 使能控制位 1：Debug 时关闭 TAU1 0：Debug 时保持 TAU1 原来状态
2	DBG_TAU0_STOP	Debug 状态下 TAU0 使能控制位 1：Debug 时关闭 TAU0 0：Debug 时保持 TAU0 原来状态
1	DBG_WWDT_STOP	Debug 状态下 WWDT 使能控制位 1：Debug 时关闭 WWDT 0：Debug 时保持 WWDT 原来状态
0	DBG_IWDT_STOP	Debug 状态下 IWDT 使能控制位 1：Debug 时关闭 IWDT 0：Debug 时保持 IWDT 开启

43.8.2 HardFault 查询寄存器

名称	HDFR							
地址	0x40000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

名称	HDFR							
地址	0x40000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	DABORT_ADDR_FLAG	DABORT_RESP_FLAG	-	BKPT_FLAG	TBIT_FLAG	DFLASH_FLAG	HDF_REQUEST_FLAG
位权限	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:7	--	RFU：未实现，读为 0
6	DABORT_ADDR_FLAG	地址非对齐访问错误标志，写 1 清零 1：地址非对齐访问错误 0：未进行地址非对齐访问
5	DABORT_RESP_FLAG	非法地址访问错误标志，写 1 清零 1：总线传输中访问了非法地址导致 HRESP 为高产生错误 0：未访问非法地址
4	--	RFU：未实现，读为 0
3	BKPT_FLAG	执行 BKPT 指令标志，写 1 清零 1：执行了 BKPT 指令 0：未执行 BKPT 指令
2	TBIT_FLAG	Thumb-State 标志，写 1 清零 1：切换到 ARM 状态 0：处于 Thumb-State
1	DFLS_FLAG	DFLASH 取指标志，写 1 清零 1：执行了特殊指令代码，如试图在 DFLASH 区域内取指 0：无指令代码试图在 DFLASH 被执行
0	HDF_REQUEST_FLAG	hardfault 标志位，任何类型的 hardfault 都会导致该位置位，写 1 清零 1：hardfault 请求 0：无 hardfault 请求

44 器件签名信息

每一颗FM33FG0A系列MCU都有自己的器件签名，包括存储器容量信息和唯一器件ID号。

44.1 唯一器件 ID

唯一器件ID保存在Flash中，包含4个word，分别保存了芯片的Lot ID、Wafer ID、芯片在wafer上的坐标位置。通过这个数据，保证每颗芯片都有独立不重复的识别码。

具体使用方法请咨询复旦微电子。

版本列表

版本号	发布日期	页数	章节或图表	更改说明
0.1	2023.3			preliminary release
0.2	2023.4			更新LQFP64封装图
0.3	2023.6			更新LIN功能部分描述
0.4	2023.6			更新电参数 更新比较器内建基准缓冲器的结构图
0.5	2023.8			更新PC引脚组数字外设功能表格 更新I2C主机配置寄存器 更新OPTBYTES选项字节

上海复旦微电子集团股份有限公司销售及服务网点

上海复旦微电子集团股份有限公司
地址：上海市国泰路 127 号 4 号楼
邮编：200433
电话：(86-021) 6565 5050
传真：(86-021) 6565 9115

上海复旦微电子（香港）股份有限公司
地址：香港九龙尖沙咀东嘉连威老道 98 号东海商业中心 5 楼 506 室
电话：(852) 2116 3288　2116 3338
传真：(852) 2116 0882

北京办事处
地址：北京市东城区东直门北小街青龙胡同 1 号歌华大厦 B 座 423 室
邮编：100007
电话：（86-10）8418 6608
传真：（86-10）8418 6211

深圳办事处
地址：深圳市华强北路 4002 号圣廷苑酒店世纪楼 1301 室
邮编：518028
电话：(86-0755) 8335 0911　8335 1011　8335 2011 8335 0611
传真：(86-0755) 8335 9011

台湾办事处
地址：台北市 114 内湖区内湖路一段 252 号 12 楼 1225 室
电话：(886-2) 7721 1889
传真：(886-2) 7722 3888

新加坡办事处
地址：237, Alexandra Road, #07-01, The Alexcier, Singapore 159929
电话：(65) 6472 3688
传真：(65) 6472 3669

北美办事处
地址：2490 W. Ray Road Suite#2 Chandler, AZ 85224 USA
电话：(480) 857-6500 ext 18

公司网址：http://www.fmsh.com/