Shanghai Fudan Microelectronics Group Company Limited

复旦微电子

FM33LG0
低功耗 MCU 芯片

产品说明书

2020. 06

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 23

图目录 26

1 产品综述 33

1.1 概述 33
1.2 芯片结构框图 36
1.3 产品型号列表 37
1.4 产品特性对照表 38

2 引脚和封装 39

2.1 封装和引脚排列 39
2.1.1 LQFP80 封装图（FM33LG0x8） 39
2.1.2 LQFP64 封装图（FM33LG0x6） 40
2.1.3 LQFP48 封装图（FM33LG0x5） 41
2.1.4 QFN32 封装图（FM33LG0x3） 42
2.1.5 引脚功能定义（FM33LG0xx） 43
2.1.6 封装尺寸图 50
2.2 焊接安装说明 56
2.3 MSL 等级 57

3 电参数 59

3.1 参数说明 59
3.2 参数测试条件 59
3.2.1 供电方案 59
3.3 极限参数 61
3.4 性能参数 62
3.4.1 典型工作条件 62
3.4.2 功耗参数 63
3.4.3 复位和电源监控 65
3.4.4 高精度基准源 66
3.4.5 低功耗模式唤醒时间 68
3.4.6 外部时钟源特性 69
3.4.7 内部时钟源特性 71
3.4.8 PLL 特性 72
3.4.9 ADC 特性 73
3.4.10 DAC 特性 78
3.4.11 温度传感器 81
3.4.12 运算放大器特性 83
3.4.13 模拟比较器特性 85
3.4.14 Flash 存储器特性 86
3.4.15 GPIO 特性 87
3.4.16 LCD 特性 89
3.4.17 VBAT 测量特性 90

4 总线与存储 91

4.1 系统总线 91
4.2 存储空间分配 92
4.2.1 概述 92

4.2.2 外设模块寄存器地址分配 95
4.3 RAM 97
4.3.1 概述 97
4.4 FLASH 98
4.4.1 概述 98
4.4.2 特殊信息扇区说明 98
4.4.3 指令 Prefetch 103
4.4.4 Flash 编程 103
4.4.5 Data Flash 108
4.4.6 Flash 的内容保护 109
4.5 寄存器 112
4.5.1 Flash 读取控制寄存器（FLS_RDCR） 112
4.5.2 预取指控制寄存器（FLS_PFCR） 113
4.5.3 用户配置字寄存器（FLS_OPTBR） 113
4.5.4 ACLOCK 寄存器 1（FLS_ACLOCK1） 114
4.5.5 ACLOCK 寄存器 2（FLS_ACLOCK2） 115
4.5.6 Flash 擦写控制寄存器（FLS_EPCR） 115
4.5.7 Flash Key 输入寄存器（FLS_KEY） 116
4.5.8 Flash 中断使能寄存器（FLS_IER） 116
4.5.9 Flash 标志寄存器（FLS_ISR） 117

5 电源管理单元（PMU） 119

5.1 芯片工作电源 119
5.1.1 电源域划分 119
5.1.2 电源结构图 120
5.1.3 ADC 和基准电压的独立供电 120
5.1.4 片内快速基准源（AVREF） 121
5.1.5 片内高精度基准源（VREF1p2） 121
5.1.6 VREFP 产生（VREFP_VREG） 121
5.2 功耗模式 122
5.2.1 概述 122
5.2.2 功耗模式与系统频率 123
5.2.3 Active 模式 124
5.2.4 LP Active 模式 124
5.2.5 LP Run 模式 125
5.2.6 SLEEP 模式 125
5.2.7 DEEPSLEEP 模式 126
5.2.8 VBAT 模式 127
5.3 唤醒源 128
5.3.1 VREF1p2 延迟唤醒功能 129
5.4 休眠唤醒后的时钟控制 130
5.5 寄存器 131
5.5.1 低功耗控制寄存器（PMU_CR） 131
5.5.2 唤醒时间控制寄存器（PMU_WKTR） 132
5.5.3 唤醒源标志查询寄存器（PMU_WKFR） 133
5.5.4 PMU 中断使能寄存器（PMU_IER） 134
5.5.5 PMU 中断标志寄存器（PMU_ISR） 135
5.5.6 ULPBG 调校寄存器（PMU_ULPB_TR） 135
5.5.7 VREFP 控制寄存器（PMU_VREFP_CR） 136
5.5.8 VREFP 配置寄存器（PMU_VREFP_CFGR） 136
5.5.9 VREFP 状态标志寄存器（PMU_VREFP_ISR） 137
5.5.10 VREFP 调校寄存器（PMU_VREFP_TR） 138

6 高精度基准源（VREF1P2） 139

6.1 概述 139
6.2 基准电压的应用 139
6.3 温度传感器 139
6.4 输出缓冲器 140
6.5 芯片休眠 140
6.6 寄存器 141
6.6.1 VREF1p2 控制寄存器（VREF_CR） 141
6.6.2 VREF1p2 配置寄存器（VREF_CFGR） 141
6.6.3 VREF1p2 标志寄存器（VREF_ISR） 142
6.6.4 VREF1p2 中断使能寄存器（VREF_IER） 143
6.6.5 模拟 BUFFER 控制寄存器（VREF_BUFCR） 143

7 VREFP 基准电压（VREFP_VREG） 145

7.1 概述 145
7.2 功能描述 145
7.2.1 概述 145
7.2.2 时钟和复位 150
7.2.3 参考电压 150
7.2.4 输出电压 150
7.2.5 工作模式 151
7.2.6 中断和标志 152
7.3 寄存器 153

8 备份电源域（VAO DOMAIN） 154

8.1 概述 154
8.2 结构框图 155
8.3 电源切换 155
8.4 复位 155
8.5 低频晶体振荡电路 (XTLF) 156
8.5.1 概述 156
8.5.2 工作方式 156
8.5.3 停振检测 156
8.6 实时时钟 (RTC_B) 157
8.7 IO 157
8.7.1 使用 PH15 输入实现 tamper 检测 157
8.8 寄存器 159
8.8.1 VAO 复位控制寄存器（VAO_RSTCR） 159
8.8.2 XTLF 控制寄存器（VAO_XTLFCR） 160
8.8.3 XTLF 振荡强度寄存器（VAO_XTLFPR） 160
8.8.4 停振检测中断使能寄存器（VAO_FDIER） 161
8.8.5 停振检测中断标志寄存器（VAO_FDISR） 162
8.8.6 VAO IO 输入使能寄存器（VAO_INEN） 162
8.8.7 VAO IO 上拉使能寄存器（VAO_PUEN） 163
8.8.8 VAO IO 开漏使能寄存器（VAO_ODEN） 163
8.8.9 VAO IO 功能选择寄存器（VAO_FCR） 164
8.8.10 VAO IO 输出数据寄存器（VAO_DOR） 164
8.8.11 VAO IO 输入数据寄存器（VAO_DIR） 165
8.8.12 VAO IO 输入低阈值配置寄存器（VAO_VILR） 165

9 跨电源域接口（CDIF） 167

9.1 概述 167
9.2 总线地址 167
9.3 时钟和复位 167
9.4 寄存器（CPU DOMAIN） 168

9.4.1 接口控制寄存器（CDIF_CR）.....168
9.4.2 接口预分频寄存器（CDIF_PRSC）.....168

10 处理器（CPU）.....170

10.1 概述.....170
10.1.1 处理器配置.....170
10.2 内核寄存器.....171
10.3 异常和中断.....172
10.3.1 中断向量表.....172
10.3.2 中断优先级.....173
10.3.3 错误处理.....173
10.3.4 锁定（Lockup）.....174
10.4 MPU.....174
10.4.1 MPU 寄存器.....175
10.5 调试特性.....179
10.5.1 调试功能引脚.....179
10.5.2 调试状态下的看门狗控制.....179
10.5.3 DEBUG 的复位.....179

11 复位管理单元（RMU）.....180

11.1 概述.....180
11.2 模块框图.....181
11.3 VDD 上下电复位.....182
11.4 软件复位.....183
11.5 NRST 引脚复位.....183
11.6 寄存器.....184
11.6.1 PDR 配置寄存器（RMU_PDRCR）.....184
11.6.2 BOR 配置寄存器（RMU_BORCR）.....185
11.6.3 复位配置寄存器（RMU_LKPCR）.....185
11.6.4 软件复位寄存器（RMU_SOFTRST）.....186
11.6.5 复位标志寄存器（RMU_RSTFR）.....186
11.6.6 外设复位使能寄存器（RMU_PRSTEN）.....187
11.6.7 AHB 外设复位寄存器（RMU_AHBRSTCR）.....187
11.6.8 APB1 外设复位寄存器（RMU_APBRSTCR1）.....188
11.6.9 APB2 外设复位寄存器（RMU_APBRSTCR2）.....190

12 独立看门狗（IWDT）.....192

12.1 概述.....192
12.2 结构框图.....192
12.3 IWDT 功能描述.....192
12.4 IWDT 窗口功能.....193
12.5 IWDT 冻结.....194
12.6 寄存器.....195
12.6.1 IWDT 清除寄存器（IWDT_SERV）.....195
12.6.2 IWDT 配置寄存器（IWDT_CR）.....195
12.6.3 IWDT 计数值寄存器（IWDT_CNT）.....196
12.6.4 IWDT 窗口寄存器（IWDT_WIN）.....197
12.6.5 IWDT 中断使能寄存器（IWDT_IER）.....197
12.6.6 IWDT 中断标志寄存器（IWDT_ISR）.....197

13 窗口看门狗（WWDT）.....199

13.1 功能描述.....199
13.2 结构框图.....199
13.3 WWDT 工作方式.....200

13.4 寄存器 202
13.4.1 WWDT 控制寄存器（WWDT_CR） 202
13.4.2 WWDT 配置寄存器（WWDT_CFGR） 202
13.4.3 WWDT 计数寄存器（WWDT_CNT） 203
13.4.4 WWDT 中断使能寄存器（WWDT_IER） 203
13.4.5 WWDT 中断标志寄存器（WWDT_ISR） 204
13.4.6 WWDT 预分频寄存器（WWDT_PSC） 204
14 时钟管理单元（CMU） 206
14.1 概述 206
14.2 时钟架构 207
14.2.1 时钟树 207
14.2.2 SYSCLK 切换说明 208
14.2.3 时钟安全性 208
14.2.4 主要时钟说明 208
14.2.5 外设模块的总线时钟和工作时钟 209
14.2.6 休眠模式下的外设时钟 210
14.2.7 LSCLK 切换逻辑 211
14.3 高频 RC 振荡器(RCHF) 212
14.3.1 概述 212
14.3.2 软件使用指南 212
14.4 低频 RC 振荡器(RCLF) 213
14.4.1 概述 213
14.5 低功耗 RC 振荡器(RCLP) 214
14.5.1 概述 214
14.5.2 软件使用指南 214
14.6 高频晶体振荡电路(XTHF) 215
14.6.1 概述 215
14.6.2 工作方式 215
14.6.3 停振检测（HFDET） 215
14.7 锁相环(PLL) 216
14.7.1 概述 216
14.7.2 软件应用指南 216
14.8 时钟校准 217
14.9 低功耗模式下的时钟源 218
14.10 休眠唤醒的时钟处理 218
14.11 寄存器 219
14.11.1 系统时钟配置寄存器（CMU_SYSCLKCR） 220
14.11.2 RCHF 时钟控制寄存器（CMU_RCHCR） 221
14.11.3 RCHF 调校寄存器（CMU_RCHTR） 222
14.11.4 PLL 控制寄存器（CMU_PLLCR） 222
14.11.5 RCLP 控制寄存器（CMU_RCLPCR） 223
14.11.6 RCLP 调校寄存器（CMU_RCLPTR） 224
14.11.7 LSCLK 选择寄存器（CMU_LSCLKSEL） 224
14.11.8 XTHF 控制寄存器（CMU_XTHCR） 225
14.11.9 RCLF 控制寄存器（CMU_RCLFCR） 225
14.11.10 RCLF 调校寄存器（CMU_RCLFTR） 226
14.11.11 CMU 中断使能寄存器（CMU_IER） 226
14.11.12 CMU 中断标志寄存器（CMU_ISR） 227
14.11.13 外设总线时钟控制寄存器 1（CMU_PCLKCR1） 228
14.11.14 外设总线时钟控制寄存器 2（CMU_PCLKCR2） 228
14.11.15 外设总线时钟控制寄存器 3（CMU_PCLKCR3） 229
14.11.16 外设总线时钟控制寄存器 4（CMU_PCLKCR4） 230

14.11.17 外设工作时钟配置寄存器1（CMU_OPCCR1）.....231
14.11.18 外设工作时钟配置寄存器2（CMU_OPCCR2）.....232
14.11.19 外设工作时钟配置寄存器3（CMU_OPCCR3）.....233
14.11.20 时钟校准控制寄存器（CMU_CCCR）.....234
14.11.21 时钟校准配置寄存器（CMU_CCFR）.....235
14.11.22 时钟校准计数值寄存器（CMU_CCNR）.....236
14.11.23 时钟校准中断标志寄存器（CMU_CCISR）.....236

15 电源电压监测（SVD）.....237

15.1 概述.....237
15.2 结构框图.....237
15.3 引脚定义.....238
15.4 功能描述.....238
15.5 间歇使能模式.....240
15.6 外部电源检测.....240
15.7 电源检测阈值.....241
15.8 寄存器.....245
15.8.1 SVD 配置寄存器（SVD_CFGR）.....245
15.8.2 SVD 控制寄存器（SVD_CR）.....246
15.8.3 SVD 中断使能寄存器（SVD_IER）.....246
15.8.4 SVD 状态和标志寄存器（SVD_ISR）.....247
15.8.5 SVD 参考电压选择寄存器（SVD_VSR）.....247

16 AES 硬件运算单元（AES）.....249

16.1 功能描述.....249
16.2 工作模式.....249
16.3 AES 数据流处理模式.....250
16.3.1 ECB 模式.....250
16.3.2 CBC 模式.....251
16.3.3 暂停模式.....253
16.3.4 CTR 模式.....254
16.3.5 CTR 模式下的暂停模式.....255
16.3.6 GCM 模式.....255
16.3.7 MultH 模块.....258
16.3.8 推荐的 GCM 流程.....259
16.4 数据类型.....260
16.5 工作流程.....261
16.5.1 模式1：加密.....261
16.5.2 模式2：密钥扩展.....262
16.5.3 模式3：解密.....263
16.5.4 模式4：密钥扩展+解密.....263
16.5.5 使用 MultH 模块.....264
16.6 DMA 接口.....265
16.6.1 MultH 模块与 DMA 间接口.....266
16.7 错误标志.....266
16.8 寄存器.....267
16.8.1 AES 控制寄存器（AES_CR）.....267
16.8.2 AES 中断使能寄存器（AES_IER）.....269
16.8.3 AES 中断标志寄存器（AES_ISR）.....270
16.8.4 AES 数据输入寄存器（AES_DIR）.....270
16.8.5 AES 数据输出寄存器（AES_DOR）.....271
16.8.6 AES 秘钥寄存器（AES_KEYx）.....271
16.8.7 AES 初始向量寄存器（AES_IVRx）.....272

16.8.8 AES MultH 参数寄存器（AES_Hx）.....272

17 随机数发生器（TRNG）.....273

17.1 概述.....273
17.2 功能描述.....274
17.2.1 随机数产生.....274
17.2.2 工作时钟.....274
17.2.3 随机数读取.....275
17.2.4 CRC 运算.....275
17.3 寄存器.....276
17.3.1 随机数控制寄存器（RNG_CR）.....276
17.3.2 随机数/CRC 结果输出寄存器（RNG_DOR）.....276
17.3.3 RNG 标志寄存器（RNG_SR）.....277
17.3.4 CRC 控制寄存器（RNG_CRCCR）.....277
17.3.5 CRC 输入数据寄存器（RNG_CRCDIR）.....278
17.3.6 CRC 标志寄存器（RNG_CRCSR）.....278

18 运算放大器（OPA1）.....280

18.1 概述.....280
18.2 结构框图.....281
18.3 引脚定义.....282
18.4 功能描述.....282
18.4.1 时钟和复位.....282
18.4.2 Standalone 模式（non-inverting amplifier）.....283
18.4.3 Standalone 模式（inverting amplifier）.....283
18.4.4 Buffer 模式.....284
18.4.5 同相 PGA 模式.....285
18.4.6 反相 PGA 模式.....287
18.4.7 Offset 校准.....289
18.4.8 低功耗模式.....290
18.4.9 休眠模式下的 OPA.....291
18.5 寄存器.....292
18.5.1 OPA1 控制寄存器（OPA1_CR）.....292
18.5.2 OPA1 校准寄存器（OPA1_CALR）.....293
18.5.3 OPA1 校准输出寄存器（OPA1_COR）.....294

19 模拟比较器（COMPARATOR）.....296

19.1 概述.....296
19.2 结构框图.....297
19.3 功能描述.....298
19.3.1 基本功能.....298
19.3.2 内部比较基准选择和产生.....298
19.3.3 时钟和复位.....299
19.3.4 引脚和内部信号连接.....299
19.3.5 窗口功能.....300
19.3.6 功耗和速度模式.....301
19.3.1 比较器中断.....302
19.3.2 比较器输出和触发输出.....302
19.3.3 输出数字滤波.....304
19.4 寄存器.....305
19.4.1 COMP1 控制寄存器（COMP1_CR）.....305
19.4.2 COMP2 控制寄存器（COMP2_CR）.....306
19.4.3 COMP3 控制寄存器（COMP3_CR）.....308
19.4.4 COMP 中断配置寄存器（COMP_ICR）.....309

19.4.5 COMP 中断标志寄存器（COMP_IF）.....310
19.4.6 COMP 缓冲器控制寄存器（COMP_BUFCR）.....310

20 除法/开方运算加速器（DIVAS）.....312

20.1 概述.....312
20.2 时钟和复位.....312
20.3 硬件除法工作流程.....312
20.4 硬件开方工作流程.....313
20.5 寄存器.....314
20.5.1 操作数寄存器.....314
20.5.2 除数寄存器.....314
20.5.3 商寄存器.....315
20.5.4 余数寄存器.....315
20.5.5 根寄存器.....316
20.5.6 状态标志寄存器.....316
20.5.7 控制寄存器.....317

21 双线串行总线（I²C）.....318

21.1 概述.....318
21.2 结构框图.....318
21.3 引脚定义和上拉电阻范围.....319
21.4 时钟和复位.....323
21.5 接口时序.....324
21.5.1 接口时序图.....324
21.5.2 接口时序描述.....325
21.6 I²C 工作模式.....327
21.7 I²C 从机地址格式.....328
21.8 I²C 初始化.....329
21.8.1 IO 配置.....329
21.8.2 主机波特率配置.....329
21.8.3 从机的输入模拟滤波和输出延迟.....330
21.9 I²C 主机功能.....331
21.9.1 7bit 寻址.....331
21.9.2 10bit 寻址.....336
21.9.3 DMA.....339
21.9.4 SCL 延展（Slave Clock Stretching）.....343
21.9.5 超时机制.....343
21.9.6 可编程时序.....343
21.10 I²C 从机功能.....345
21.10.1 从机寻址.....345
21.10.2 从机发送数据.....345
21.10.3 从机接收数据.....346
21.10.4 从机低功耗接收唤醒.....348
21.10.5 DMA.....348
21.10.6 从机时序.....352
21.11 寄存器.....353
21.11.1 I2C 主机配置寄存器（I2C_MSPCFGR）.....353
21.11.2 I2C 主机控制寄存器（I2C_MSPCR）.....354
21.11.3 I2C 主机中断使能寄存器（I2C_MSPIER）.....355
21.11.4 I2C 主机中断标志寄存器（I2C_MSPISR）.....356
21.11.5 I2C 主机状态寄存器（I2C_MSPSR）.....356
21.11.6 I2C 主机波特率设置寄存器（I2C_MSPBGR）.....357
21.11.7 I2C 主机收发缓冲寄存器（I2C_MSPBUF）.....358
21.11.8 I2C 主机时序控制寄存器（I2C_MSPTCR）.....358

21.11.9 I2C 主机超时寄存器（I2C_MSPTOR）.....359
21.11.10 I2C 从机控制寄存器（I2C_SSPCR）.....359
21.11.11 I2C 从机中断使能寄存器（I2C_SSPIER）.....360
21.11.12 I2C 从机中断标志寄存器（I2C_SSPISR）.....361
21.11.13 I2C 从机状态寄存器（I2C_SSPSR）.....362
21.11.14 I2C 从机收发缓冲寄存器（I2C_SSPBUF）.....363
21.11.15 I2C 从机地址寄存器（I2C_SSPADR）.....363

22 通用异步收发传输器（UART）.....365

22.1 概述.....365
22.2 结构框图.....366
22.3 引脚定义.....367
22.4 UART 类型区分.....369
22.5 UART 字符描述.....369
22.6 功能描述.....371
22.6.1 时钟和复位.....371
22.6.2 位接收采样.....371
22.6.3 数据发送.....372
22.6.4 数据接收.....374
22.6.5 低功耗休眠唤醒（UART0/1）.....375
22.6.6 使用 DMA 进行 UART 收发.....375
22.6.7 DMA 模式下的发送完成中断.....376
22.7 波特率发生.....376
22.7.1 波特率发生.....376
22.7.1 波特率自适应.....377
22.8 红外调制.....378
22.9 接收超时.....379
22.10 发送延迟.....379
22.11 寄存器.....380
22.11.1 红外调制寄存器（UART_IRCR）.....381
22.11.2 UARTx 控制状态寄存器（UARTx_CSR）.....382
22.11.3 UARTx 中断使能寄存器（UARTx_IER）.....383
22.11.4 UARTx 中断标志寄存器（UARTx_ISR）.....384
22.11.5 UARTx 超时和延迟寄存器（UARTx_TODR）.....385
22.11.6 UARTx 接收缓冲寄存器（UARTx_RXBUF）.....385
22.11.7 UARTx 发送缓冲寄存器（UARTx_TXBUF）.....386
22.11.8 UATRx 波特率产生寄存器（UARTx_BGR）.....386

23 低功耗 UART（LPUART）.....388

23.1 概述.....388
23.2 结构框图.....389
23.3 引脚定义.....390
23.4 时钟和复位.....390
23.5 字符描述.....391
23.6 功能描述.....393
23.6.1 位接收采样和发送.....393
23.6.2 接收流程.....394
23.6.3 发送流程.....394
23.6.4 使用 DMA 进行 LPUART 收发.....394
23.6.5 休眠模式下的数据接收唤醒.....395
23.6.6 LPRUN 模式下的数据 DMA 收发.....395
23.6.7 DMA 模式下的发送完成中断.....395
23.7 寄存器.....397

23.7.1 LPUARTx 控制状态寄存器（LPUARTx_CSR）.....398
23.7.2 LPUARTx 中断使能寄存器（LPUARTx_IER）.....399
23.7.3 LPUARTx 中断标志寄存器（LPUARTx_ISR）.....400
23.7.4 LPUARTx 波特率调制寄存器（LPUARTx_BMR）.....400
23.7.5 LPUARTx 接收数据寄存器（LPUARTx_RXBUF）.....401
23.7.6 LPUARTx 发送数据寄存器（LPUARTx_TXBUF）.....402
23.7.7 LPUARTx 数据匹配寄存器（LPUARTx_DMR）.....402

24 串行外设接口（SPI）.....404

24.1 概述.....404
24.2 结构框图.....404
24.3 引脚定义.....406
24.4 时钟和复位.....406
24.5 接口时序.....406
24.5.1 CPHA=0.....406
24.5.2 CPHA=1.....407
24.5.1 4 线半双工模式（主机）.....407
24.6 功能描述.....409
24.6.1 I/O 配置.....409
24.6.2 全双工数据通信.....410
24.6.3 TX-ONLY 模式.....411
24.6.4 RX-ONLY 模式.....411
24.6.5 主机 SSN 控制.....411
24.6.6 数据冲突.....412
24.6.7 使用 DMA 进行 SPI 收发.....413
24.7 寄存器.....415
24.7.1 SPI 控制寄存器 1（SPIx_CR1）.....416
24.7.2 SPI 控制寄存器 2（SPIx_CR2）.....417
24.7.3 SPI 控制寄存器 3（SPIx_CR3）.....418
24.7.4 SPI 中断控制寄存器（SPIx_IER）.....419
24.7.5 SPI 中断标志寄存器（SPIx_ISR）.....419
24.7.6 SPI 发送缓存寄存器（SPIx_TXBUF）.....420
24.7.7 SPI 发送缓存寄存器（SPIx_RXBUF）.....420

25 智能卡接口（ISO7816）.....422

25.1 概述.....422
25.2 结构框图.....422
25.3 时钟和复位.....423
25.4 接口时序.....423
25.5 功能描述.....424
25.5.1 数据接收.....424
25.5.2 数据发送.....424
25.5.3 使用 DMA 进行 7816 收发.....426
25.6 寄存器.....427
25.6.1 U7816 控制寄存器（U7816_CR）.....427
25.6.2 U7816 帧格式控制寄存器（U7816_FFR）.....428
25.6.3 U7816 额外保护时间寄存器（U7816_EGTR）.....429
25.6.4 U7816 工作时钟分频寄存器（U7816_PSC）.....429
25.6.5 U7816 波特率寄存器（U7816_BGR）.....430
25.6.6 U7816 数据接收缓冲寄存器（U7816_RXBUF）.....430
25.6.7 U7816 数据发送缓冲寄存器（U7816_TXBUF）.....431
25.6.8 U7816 中断使能寄存器（U7816_IER）.....431
25.6.9 U7816 状态标志寄存器（U7816_ISR）.....432

26 控制器区域网络（CAN） 434

26.1 概述 434
26.2 结构框图 434
26.3 引脚定义 435
26.4 功能描述 435
26.4.1 时钟和复位 435
26.4.2 Bit timing 435
26.4.3 Bit stream processor 436
26.4.4 控制器工作模式 436
26.4.5 消息帧存储和消息帧结构（Message storage and structure） 438
26.4.6 消息滤波器（Acceptance Filter） 439
26.5 编程模型 441
26.5.1 寄存器配置 441
26.5.2 消息传输 441
26.6 寄存器 443
26.6.1 CAN 控制寄存器（CAN_CR） 444
26.6.2 CAN 模式选择寄存器（CAN_MSR） 445
26.6.3 CAN 波特率预分频寄存器（CAN_BRPR） 445
26.6.4 CAN 位时序寄存器（CAN_BTR） 446
26.6.5 CAN 错误计数寄存器（CAN_ECR） 446
26.6.6 CAN 错误状态寄存器（CAN_ESR） 447
26.6.7 CAN 状态标志寄存器（CAN_SR） 448
26.6.8 CAN 中断标志寄存器（CAN_ISR） 449
26.6.9 CAN 中断使能寄存器（CAN_IER） 450
26.6.10 CAN 中断清除寄存器（CAN_ICR） 451
26.6.11 CAN 发送 FIFO ID 寄存器（CAN_TXFIDR） 452
26.6.12 CAN 发送 FIFO DLC 寄存器（CAN_TXFDLCR） 452
26.6.13 CAN 发送 FIFO DataWord1 寄存器（CAN_TXFDW1R） 453
26.6.14 CAN 发送 FIFO DataWord2 寄存器（CAN_TXFDW2R） 453
26.6.15 CAN 高优先级发送缓冲 ID 寄存器（CAN_HPBIDR） 454
26.6.16 CAN 高优先级发送缓冲 DLC 寄存器（CAN_HPBDLCR） 454
26.6.17 CAN 高优先级发送缓冲 DataWord1 寄存器（CAN_HPBDW1R） 454
26.6.18 CAN 高优先级发送缓冲 DataWord2 寄存器（CAN_HPBDW2R） 455
26.6.19 CAN 接收 FIFO ID 寄存器（CAN_RXF_IDR） 455
26.6.20 CAN 接收 FIFO DLC 寄存器（CAN_RXFDLCR） 456
26.6.21 CAN 接收 FIFO DataWord1 寄存器（CAN_RXFDW1R） 456
26.6.22 CAN 接收 FIFO DataWord2 寄存器（CAN_RXFDW2R） 457
26.6.23 CAN 接收滤波寄存器（CAN_AFR） 457
26.6.24 CAN 接收滤波掩码寄存器 x（CAN_AFMRx） 458
26.6.25 CAN 接收滤波 ID 寄存器 x（CAN_AFIRx） 459

27 直接存储访问控制器（DMA） 460

27.1 概述 460
27.2 工作原理 461
27.3 结构框图 462
27.4 工作流程 462
27.5 访问带宽 464
27.6 通道控制 465
27.6.1 DMA 请求映射 465
27.6.2 通道优先级 466
27.6.3 传输方向定义 466
27.6.4 循环模式 466
27.7 寄存器 467

27.7.1 DMA 全局控制寄存器（DMA_GCR）.....467
27.7.2 通道 x 控制寄存器（DMA_CHxCR）.....468
27.7.3 通道 x 存储器指针寄存器（DMA_CHxMAR）.....469
27.7.4 通道 7 控制寄存器（DMA_CH7CR）.....470
27.7.5 通道 7 Flash 指针寄存器（DMA_CH7FLSAD）.....471
27.7.6 通道 7 RAM 指针寄存器（DMA_CH7RAMAD）.....471
27.7.7 DMA 状态标志寄存器（DMA_ISR）.....472

28 循环冗余校验（CRC）.....473

28.1 概述.....473
28.2 软件配置过程.....474
28.3 GOLDEN 数据.....475
28.4 DMA 接口.....475
28.5 FLASH 数据完整性校验.....476
28.6 寄存器.....477
28.6.1 CRC 数据寄存器（CRC_DR）.....477
28.6.2 CRC 控制状态寄存器（CRC_CR）.....477
28.6.3 CRC LFSR 寄存器（CRC_LFSR）.....479
28.6.4 CRC 输出异或寄存器（CRC_XOR）.....479
28.6.5 CRC 多项式寄存器（CRC_POLY）.....479

29 高级定时器（ATIM）.....481

29.1 概述.....481
29.2 主要特性.....481
29.3 结构框图.....482
29.4 功能描述.....483
29.4.1 定时单元.....483
29.4.2 定时器工作模式.....485
29.4.3 重复计数器.....492
29.4.4 Preload 寄存器.....493
29.4.5 计数器工作时钟.....494
29.4.6 内部触发信号（ITRx）.....499
29.4.7 捕捉/比较通道.....500
29.4.8 输入捕捉模式.....502
29.4.9 软件 Force 输出.....504
29.4.10 输出比较模式.....505
29.4.11 PWM 输出.....506
29.4.12 互补输出和死区插入.....508
29.4.13 刹车功能.....509
29.4.14 互补输出通道信号状态逻辑表.....511
29.4.15 6-step PWM 输出.....512
29.4.16 单脉冲输出.....513
29.4.17 外部事件清除 OCxREF.....515
29.4.18 编码器接口模式（encoder interface）.....516
29.4.19 TIM 从机模式.....518
29.4.20 DMA 访问.....521
29.4.21 DMA Burst.....522
29.4.22 输入异或功能.....523
29.4.23 Debug 模式.....523
29.5 寄存器.....524
29.5.1 ATIM 控制寄存器 1（ATIM_CR1）.....524
29.5.2 ATIM 控制寄存器 2（ATIM_CR2）.....526
29.5.3 ATIM 从机模式控制寄存器（ATIM_SMCR）.....527
29.5.4 ATIM DMA 和中断使能寄存器（ATIM_DIER）.....529

29.5.5 ATIM 状态寄存器（ATIM_ISR）.....530
29.5.6 ATIM 事件产生寄存器（ATIM_EGR）.....532
29.5.7 ATIM 捕捉/比较模式寄存器 1（ATIM_CCMR1）.....532
29.5.8 ATIM 捕捉/比较模式寄存器 2（ATIM_CCMR2）.....535
29.5.9 ATIM 捕捉/比较使能寄存器（ATIM_CCER）.....537
29.5.10 ATIM 计数器寄存器（ATIM_CNT）.....538
29.5.11 ATIM 预分频寄存器（ATIM_PSC）.....538
29.5.12 ATIM 自动重载寄存器（ATIM_ARR）.....539
29.5.13 ATIM 重复计数寄存器（ATIM_RCR）.....539
29.5.14 ATIM 捕捉/比较寄存器 1（ATIM_CCR1）.....540
29.5.15 ATIM 捕捉/比较寄存器 2（ATIM_CCR2）.....541
29.5.16 ATIM 捕捉/比较寄存器 3（ATIM_CCR3）.....541
29.5.17 ATIM 捕捉/比较寄存器 4（ATIM_CCR4）.....542
29.5.18 ATIM 刹车和死区控制寄存器（ATIM_BDTR）.....542
29.5.19 ATIM DMA 控制寄存器（ATIM_DCR）.....544
29.5.20 ATIM DMA 访问寄存器（ATIM_DMAR）.....545
29.5.21 ATIM 刹车输入控制寄存器（ATIM_BKCR）.....545

30 通用定时器（GPTIM0,1,2）.....547

30.1 概述.....547
30.2 主要特性.....547
30.3 结构框图.....548
30.4 功能描述.....549
30.4.1 定时单元.....549
30.4.2 定时器工作模式.....551
30.4.3 计数器工作时钟.....558
30.4.4 内部触发信号（ITRx）的捕捉.....564
30.4.5 捕捉/比较通道.....565
30.4.6 输入捕捉模式.....566
30.4.7 软件 Force 输出.....568
30.4.8 输出比较模式.....568
30.4.9 PWM 输入模式.....569
30.4.10 单脉冲输出.....570
30.4.11 外部事件清除 OCxREF.....572
30.4.12 编码器接口模式（encoder interface）.....572
30.4.13 GPTIM 从机模式.....573
30.4.14 DMA 访问.....576
30.4.15 DMA Burst.....577
30.4.16 输入异或功能.....577
30.4.17 Debug 模式.....577
30.5 寄存器.....578
30.5.1 GPTIMx 控制寄存器 1（GPTIMx_CR1）.....580
30.5.2 GPTIMx 控制寄存器 2（GPTIMx_CR2）.....581
30.5.3 GPTIMx 从机模式控制寄存器（GPTIMx_SMCR）.....582
30.5.4 GPTIMx DMA 和中断使能寄存器（GPTIMx_DIER）.....584
30.5.5 GPTIMx 状态寄存器（GPTIMx_ISR）.....585
30.5.6 GPTIMx 事件产生寄存器（GPTIMx_EGR）.....586
30.5.7 GPTIMx 捕捉/比较模式寄存器 1（GPTIMx_CCMR1）.....587
30.5.8 GPTIMx 捕捉/比较模式寄存器 2（GPTIMx_CCMR2）.....589
30.5.9 GPTIMx 捕捉/比较使能寄存器（GPTIMx_CCER）.....592
30.5.10 GPTIMx 计数器寄存器（GPTIMx_CNT）.....593
30.5.11 GPTIMx 预分频寄存器（GPTIMx_PSC）.....593
30.5.12 GPTIMx 自动重载寄存器（GPTIMx_ARR）.....594

30.5.13 GPTIMx 捕捉/比较寄存器 1（GPTIMx_CCR1）.....594
30.5.14 GPTIMx 捕捉/比较寄存器 2（GPTIMx_CCR2）.....595
30.5.15 GPTIMx 捕捉/比较寄存器 3（GPTIMx_CCR3）.....595
30.5.16 GPTIMx 捕捉/比较寄存器 4（GPTIMx_CCR4）.....596
30.5.17 GPTIMx DMA 控制寄存器（GPTIMx_DCR）.....596
30.5.18 GPTIMx DMA 访问寄存器（GPTIMx_DMAR）.....598
30.5.19 GPTIMx ITR 选择寄存器（GPTIMx_ITRSEL）.....598

31 32 位基本定时器（BSTIM32）.....600

31.1 概述.....600
31.2 主要特性.....600
31.3 结构框图.....600
31.4 功能描述.....601
31.4.1 定时单元.....601
31.4.2 定时器工作模式.....603
31.4.3 计数器工作时钟.....605
31.4.1 Debug 模式.....606
31.5 寄存器.....607
31.5.1 BSTIM 控制寄存器 1（BSTIM_CR1）.....607
31.5.2 BSTIM 控制寄存器 2（BSTIM_CR2）.....608
31.5.3 BSTIM 中断使能寄存器（BSTIM_IER）.....608
31.5.4 BSTIM 中断标志寄存器（BSTIM_ISR）.....609
31.5.5 BSTIM 事件产生寄存器（BSTIM_EGR）.....610
31.5.6 BSTIM 计数器寄存器（BSTIM_CNT）.....610
31.5.7 BSTIM 预分频寄存器（BSTIM_PSC）.....611
31.5.8 BSTIM 自动重载寄存器（BSTIM_ARR）.....611

32 16 位基本定时器（BSTIM16）.....612

32.1 概述.....612
32.2 主要特性.....612
32.3 结构框图.....612
32.4 功能描述.....613
32.4.1 定时单元.....613
32.4.2 定时器工作模式.....615
32.4.3 计数器工作时钟.....617
32.4.4 Debug 模式.....618
32.5 寄存器.....619
32.5.1 BSTIM 控制寄存器 1（BSTIM_CR1）.....619
32.5.2 BSTIM 控制寄存器 2（BSTIM_CR2）.....620
32.5.3 BSTIM 中断使能寄存器（BSTIM_IER）.....621
32.5.4 BSTIM 中断标志寄存器（BSTIM_ISR）.....621
32.5.5 BSTIM 事件产生寄存器（BSTIM_EGR）.....622
32.5.1 BSTIM 计数器寄存器（BSTIM_CNT）.....622
32.5.1 BSTIM 预分频寄存器（BSTIM_PSC）.....623
32.5.1 BSTIM 自动重载寄存器（BSTIM_ARR）.....623

33 32 位低功耗定时器（LPTIM32）.....625

33.1 概述.....625
33.2 结构框图.....626
33.3 时钟和复位.....626
33.4 相关引脚.....627
33.5 定时器功能.....627
33.5.1 普通定时器.....627
33.5.2 外部脉冲触发计数.....627

33.5.3 外部异步脉冲计数 628
33.5.4 Timeout 模式 628
33.6 捕捉比较功能 629
33.6.1 32bit PWM 629
33.6.2 输入捕捉 630
33.7 触发信号输出 631
33.8 寄存器 632
33.8.1 LPTIM 配置寄存器（LPTIM_CFGR） 632
33.8.2 LPTIM 计数值寄存器（LPTIM_CNT） 634
33.8.3 LPTIM 捕捉比较控制和状态寄存器（LPTIM_CCSR） 634
33.8.4 LPTIM 目标值寄存器（LPTIM_ARR） 636
33.8.5 LPTIM 中断使能寄存器（LPTIM_IER） 637
33.8.6 LPTIM 中断标志寄存器（LPTIM_ISR） 638
33.8.7 LPTIM 控制寄存器（LPTIM_CR） 639
33.8.8 LPTIM 捕捉比较寄存器 1（LPTIM_CCR1） 640
33.8.9 LPTIM 捕捉比较寄存器 2（LPTIM_CCR2） 640
33.8.10 LPTIM 捕捉比较寄存器 3（LPTIM_CCR3） 641
33.8.11 LPTIM 捕捉比较寄存器 4（LPTIM_CCR4） 641

34 16 位低功耗定时器（LPTIM16） 642

34.1 概述 642
34.2 结构框图 643
34.3 时钟和复位 643
34.4 相关引脚 644
34.5 定时器功能 644
34.5.1 普通定时器 644
34.5.2 外部脉冲触发计数 644
34.5.3 外部异步脉冲计数 645
34.5.4 Timeout 模式 645
34.6 捕捉比较功能 646
34.6.1 16bit PWM 646
34.6.2 输入捕捉 647
34.6.3 输入数字滤波 648
34.7 正交编码器 649
34.8 触发信号输出 650
34.9 寄存器 651
34.9.1 LPTIM 配置寄存器（LPTIM_CFGR） 651
34.9.2 LPTIM 计数值寄存器（LPTIM_CNT） 653
34.9.3 LPTIM 捕捉比较控制和状态寄存器（LPTIM_CCSR） 653
34.9.4 LPTIM 目标值寄存器（LPTIM_ARR） 655
34.9.5 LPTIM 中断使能寄存器（LPTIM_IER） 656
34.9.6 LPTIM 中断标志寄存器（LPTIM_ISR） 656
34.9.7 LPTIM 控制寄存器（LPTIM_CR） 657
34.9.8 LPTIM 捕捉比较寄存器 1（LPTIM_CCR1） 658
34.9.9 LPTIM 捕捉比较寄存器 2（LPTIM_CCR2） 658

35 实时时钟（RTCA） 660

35.1 概述 660
35.2 结构框图 660
35.3 工作原理 661
35.3.1 时基计数器（LTBC） 661
35.3.2 LTBC 数字调校 662
35.3.3 BCD 时间 663
35.3.4 RTC 使能与停止 664

35.3.5 RTC 时间设置 664
35.3.6 RTC 时间读取 664
35.3.7 闰年判断 665
35.4 寄存器 666
35.4.1 RTCA 写使能寄存器（RTCA_WER） 666
35.4.2 RTCA 中断使能寄存器（RTCA_IER） 667
35.4.3 RTCA 中断标志寄存器（RTCA_ISR） 668
35.4.4 BCD 时间秒寄存器（RTCA_BCDSEC） 669
35.4.5 BCD 时间分钟寄存器（RTCA_BCDMIN） 670
35.4.6 BCD 时间小时寄存器（RTCA_BCDHOUR） 670
35.4.7 BCD 时间天寄存器（RTCA_BCDDAY） 671
35.4.8 BCD 时间星期寄存器（RTCA_BCDWEEK） 671
35.4.9 BCD 时间月寄存器（RTCA_BCDMONTH） 672
35.4.10 BCD 时间年寄存器（RTCA_BCDYEAR） 672
35.4.11 闹钟寄存器（RTCA_ALARM） 673
35.4.12 RTC 时间信号输出寄存器（RTCA_TMSEL） 673
35.4.13 LTBC 数值调整寄存器（RTCA_ADJUST） 674
35.4.14 毫秒计数值寄存器（RTCA_SBSCNT） 675
35.4.15 RTCA 控制寄存器 RTCA_CR 675

36 实时时钟（RTCB） 676

36.1 概述 676
36.2 结构框图 676
36.3 工作原理 677
36.3.1 工作时钟 677
36.3.2 时基计数器（LTBC） 677
36.3.3 LTBC 数字调校 677
36.3.4 BCD 时间 679
36.3.5 RTC 使能与停止 679
36.3.6 RTC 时间设置 680
36.3.7 RTC 时间读取 680
36.3.8 闰年判断 681
36.3.9 RTC 时间戳 681
36.3.10 备份寄存器 682
36.4 寄存器 683
36.4.1 RTCB 写使能寄存器（RTCB_WER） 684
36.4.2 RTCB 中断使能寄存器（RTCB_IER） 684
36.4.3 RTCB 中断标志寄存器（RTCB_ISR） 685
36.4.4 BCD 时间秒寄存器（RTCB_BCDSEC） 686
36.4.5 BCD 时间分钟寄存器（RTCB_BCDMIN） 686
36.4.6 BCD 时间小时寄存器（RTCB_BCDHOUR） 687
36.4.7 BCD 时间天寄存器（RTCB_BCDDAY） 687
36.4.8 BCD 时间星期寄存器（RTCB_BCDWEEK） 688
36.4.9 BCD 时间月寄存器（RTCB_BCDMONTH） 688
36.4.10 BCD 时间年寄存器（RTCB_BCDYEAR） 689
36.4.11 RTCB 时间信号输出寄存器（RTCB_TMSEL） 689
36.4.12 RTCB 时钟调校寄存器（RTCB_ADJR） 690
36.4.13 RTCB 控制寄存器（RTCB_CR） 690
36.4.14 RTCB 时间戳控制寄存器（RTCB_STPCR） 691
36.4.15 RTCB 时间戳时钟寄存器（RTCB_STPCLKRR） 692
36.4.16 RTCB 时间戳日历寄存器（RTCB_STPCALRR） 692
36.4.17 RTCB 备份寄存器组（RTCB_BKRx） 693

37 段码液晶驱动（LCD） 694

37.1 概述 694
37.2 结构框图 694
37.3 IO 配置 696
37.4 功能说明 696
37.4.1 工作时钟和显示帧频率 696
37.4.2 LCD Type A 扫描波形 696
37.4.3 LCD Type B 扫描波形 701
37.4.4 片内 buffer 驱动模式 705
37.4.5 片外电容驱动模式 705
37.4.6 显示闪烁功能 706
37.4.7 偏置电压调整 706
37.5 低功耗模式 707
37.6 寄存器 708
37.6.1 显示控制寄存器（LCD_CR） 708
37.6.2 显示测试控制寄存器（LCD_TEST） 710
37.6.3 显示频率控制寄存器（LCD_FCR） 711
37.6.4 闪烁时间寄存器（LCD_FLKT） 712
37.6.5 显示中断使能寄存器（LCD_IER） 712
37.6.6 显示中断标志寄存器（LCD_ISR） 713
37.6.7 显示数据寄存器（LCD_DATAx） 713
37.6.8 LCD COM 使能控制寄存器（LCD_COMEN） 720
37.6.9 LCD SEG 使能控制寄存器 0（LCD_SEGEN0） 720
37.6.10 LCD SEG 使能控制寄存器 1（LCD_SEGEN1） 721

38 模数转换器（ADC） 722

38.1 概述 722
38.2 结构框图 723
38.3 输入通道 724
38.4 单端和差分输入 725
38.5 工作时序 727
38.6 功能描述 729
38.6.1 使用 VDDA 作为基准 729
38.6.2 使用 VREFP 作为基准 730
38.6.3 温度传感器 730
38.6.4 温度传感器的斜率和标定 732
38.6.5 可编程采样时间 732
38.6.6 输出位宽选择 733
38.6.7 输入缓冲器 734
38.6.8 VBAT 和 VDD 电源电压采样 735
38.6.9 内部信号采样 735
38.6.10 转换模式 736
38.6.11 转换触发 739
38.6.12 过采样和硬件平均 740
38.6.13 ADC 工作时钟 741
38.6.14 数据冲突和自动等待 741
38.6.15 DMA 742
38.6.16 模拟窗口看门狗（AWD） 748
38.6.17 ADC 校准 749
38.7 低功耗模式 750
38.8 寄存器 751
38.8.1 ADC 中断和状态寄存器（ADC_ISR） 751
38.8.2 ADC 中断使能寄存器（ ADC_IER） 752
38.8.3 ADC 控制寄存器 1（ADC_CR1） 753

38.8.4 ADC 控制寄存器 2（ADC_CR2）.....753
38.8.5 ADC 校准控制寄存器（ADC_CALR）.....754
38.8.6 ADC 配置寄存器 1（ADC_CFGR1）.....755
38.8.7 ADC 配置寄存器 2（ADC_CFGR2）.....756
38.8.8 ADC 采样时间控制寄存器（ADC_SMTR）.....758
38.8.9 ADC 通道控制寄存器（ADC_CHER）.....760
38.8.10 ADC 通道差分控制寄存器（ADC_DCR）.....761
38.8.11 ADC 数据寄存器（ADC_DR）.....761
38.8.12 AWD 阈值寄存器（ADC_HLTR）.....762

39 数模转换器（DAC）.....763

39.1 概述.....763
39.2 结构框图.....763
39.3 引脚定义.....764
39.4 DAC 输出与引脚和其他模块的互联.....764
39.5 功能描述.....765
39.5.1 工作时钟与信号时序.....765
39.5.2 DAC 输出模式.....765
39.5.3 DAC 触发源选择.....767
39.5.4 DAC 输出电压.....767
39.5.5 DMA.....767
39.5.6 采样保持.....768
39.5.7 DAC 输出 Buffer.....770
39.5.8 低功耗模式下的 DAC.....771
39.6 寄存器.....772
39.6.1 DAC 控制寄存器 1（DAC_CR1）.....772
39.6.2 DAC 控制寄存器 2（DAC_CR2）.....772
39.6.1 DAC 配置寄存器（DAC_CFGR）.....773
39.6.2 DAC 软件触发寄存器（DAC_SWTRGR）.....774
39.6.3 DAC 数据保持寄存器（DAC_DHR）.....774
39.6.4 DAC 状态标志寄存器（DAC_ISR）.....775
39.6.5 DAC 中断使能寄存器（DAC_IER）.....775
39.6.1 DAC 采样保持时间寄存器（DAC_SHTR）.....776

40 可编程胶合逻辑（PGL）.....778

40.1 概述.....778
40.2 结构框图.....778
40.3 引脚定义.....780
40.4 功能描述.....781
40.4.1 LUT 真值表.....781
40.4.2 LUT 输入.....782
40.4.3 LUT 输出.....782
40.4.4 滤波和采样.....782
40.4.5 中断和触发.....783
40.4.6 低功耗模式.....784
40.5 寄存器.....785
40.5.1 PGL 控制寄存器（PGL_CR）.....785
40.5.2 PGL 配置寄存器 0（PGL_CFGR0）.....786
40.5.3 PGL 配置寄存器 1（PGL_CFGR1）.....787
40.5.4 PGL 配置寄存器 2（PGL_CFGR2）.....788
40.5.5 PGL 配置寄存器 3（PGL_CFGR3）.....789
40.5.6 PGL 中断使能寄存器（PGL_IER）.....790
40.5.7 PGL 中断标志寄存器（PGL_ISR）.....791
40.5.8 LUT0 真值表寄存器（PGL_LUT0）.....791

40.5.9 LUT1 真值表寄存器（PGL_LUT1）.....792
40.5.10 LUT2 真值表寄存器（PGL_LUT2）.....792
40.5.11 LUT3 真值表寄存器（PGL_LUT3）.....793

41 I/O 端口（GPIO）.....794

41.1 概述.....794
41.2 引脚类型.....794
41.2.1 GPIO，输入输出使能，可控上拉电阻，可控开漏输出.....795
41.2.2 GPIO，输入输出使能，2 个可控上拉电阻，可控开漏输出（仅 PC12-7816 数据口）.....796
41.3 IO 端口功能定义.....796
41.3.1 GPIO 输入.....796
41.3.2 GPIO 输出.....797
41.3.3 数字外设功能.....797
41.3.4 模拟功能.....799
41.3.5 使用外部晶体引脚.....800
41.4 VBAT 供电引脚.....800
41.5 SWD 引脚.....801
41.6 WKUPx 引脚.....801
41.7 外部引脚中断（EXTI）.....802
41.7.1 功能说明.....802
41.7.2 应用指南.....803
41.8 快速 GPIO 输出.....804
41.9 寄存器.....805
41.9.1 GPIO 输入使能寄存器（GPIOx_INEN）.....808
41.9.2 GPIO 上拉使能寄存器（GPIOx_PUEN）.....808
41.9.3 GPIO 开漏使能寄存器（GPIOx_ODEN）.....809
41.9.4 GPIO 功能选择寄存器（GPIOx_FCR）.....809
41.9.5 GPIO 输出数据寄存器（GPIOx_DO）.....811
41.9.6 GPIO 输出数据置位寄存器（GPIOx_DSET）.....812
41.9.7 GPIO 输出数据复位寄存器（GPIOx_DRST）.....813
41.9.8 GPIO 输入数据寄存器（GPIOx_DIN）.....813
41.9.9 GPIO 额外数字功能选择寄存器（GPIOx_DFS）.....814
41.9.10 GPIO 模拟开关使能寄存器（GPIOx_ANEN）.....814
41.9.11 GPIO 输入低阈值配置寄存器（GPIOx_VILR）.....815
41.9.12 EXTI 输入选择寄存器 0（GPIO_EXTISEL0）.....816
41.9.13 EXTI 输入选择寄存器 1（GPIO_EXTISEL1）.....818
41.9.14 EXTI 边沿选择和使能寄存器 0（GPIO_EXTIEDS0）.....819
41.9.1 EXTI 边沿选择和使能寄存器 1（GPIO_EXTIEDS1）.....819
41.9.2 EXTI 数字滤波控制寄存器（GPIO_EXTIDF）.....820
41.9.3 EXTI 中断标志（GPIO_EXTIISR）.....820
41.9.4 EXTI 输入信号寄存器（GPIO_EXTIDI）.....821
41.9.5 FOUT 配置寄存器（GPIO_FOUTSEL）.....821
41.9.6 WKUP 控制寄存器（GPIO_PINWKEN）.....822

42 专用编程接口.....824

42.1 概述.....824
42.2 编程器使用.....824

43 调试支持.....825

43.1 概述.....825
43.2 DEBUG 引脚.....826
43.2.1 SWD 引脚.....826
43.2.2 上拉电阻.....826
43.3 SWD 接口协议.....827

43.3.1 协议简介 827
43.3.2 传输序列 827
43.3.3 SW-DP ID code 828
43.3.4 主机读操作 828
43.3.5 主机写操作 829
43.4 SWD-DP 寄存器 830
43.4.1 寄存器列表 830
43.5 CORE DEBUG 寄存器 830
43.6 低功耗调试支持 830
43.7 DEBUG 相关的配置项 831
43.8 寄存器 831
43.8.1 系统模式配置寄存器 831
43.8.2 MCU DEBUG 配置寄存器 832
43.8.3 HardFault 查询寄存器 833
44 器件签名信息 835
44.1 存储器容量查询 835
44.2 器件 UID 836
版本列表 837
上海复旦微电子集团股份有限公司销售及服务网点 838

表目录

表 1-1FM33LG0 型号列表 37
表 1-2 FM33LG0xx 特性列表 38
表 2-1 引脚列表 49
表 3-1FM33LG0 极限参数 61
表 3-2FM33LG0 典型工作条件 62
表 3-3ACTIVE 电流参数 63
表 3-4LP ACTIVE 电流参数 63
表 3-5LP RUN 电流参数 64
表 3-6SLEEP 电流参数 64
表 3-7DEEPSLEEP 电流参数 64
表 3-8VBAT 电流参数 64
表 3-9 复位和电源检测参数 66
表 3-10 高精度基准源参数 67
表 3-11 唤醒时间参数 68
表 3-12 低频晶体振荡器参数 69
表 3-13 高频晶体振荡器参数 69
表 3-14 内部高频 RC 振荡器参数 71
表 3-15 内部中频 RC 振荡器参数 71
表 3-16 内部低频 RC 振荡器参数 72
表 3-17 PLL 参数 72
表 3-18ADC 参数 74
表 3-19ADC 输入阻抗 77
表 3-20 DAC 参数 79
表 3-21 温度传感器参数 82
表 3-22 OPA 参数 84
表 3-23 模拟比较器参数 85
表 3-24 FLASH 参数 86
表 3-25 普通 I/O 参数 87
表 3-26 NRST 引脚参数 88
表 3-27 引脚 AC 参数 88
表 3-28 LCD 片内电阻分压 89
表 3-29 LCD 片外电容驱动 89
表 3-30VBAT 测量特性 90
表 4-1 外设模块总线地址列表 96
表 4-2FLASH 特殊信息区 98
表 4-3FLASHLDT0 扇区 99
表 4-4LDT0 数据格式 99
表 4-5FLASHLDT1 扇区 100
表 4-6FLASH 选项字节 100
表 4-7FLASH LOCK 配制 101
表 4-8LOCK 位和地址对应表 101
表 4-9DATAFLASH 配置 108
表 4-10LOCK 位权限控制 110
表 4-11FLASH 权限控制 111
表 5-1 功耗模式表 123
表 5-2 功耗模式与频率对照表 124
表 5-3 休眠模式唤醒源列表 128
表 5-4VREF1P2 延迟唤醒可应用的唤醒源 129
表 10-1 FM33LGx0xx CPU 配置简表 170

表 10-2 CORTEX-M0 内核寄存器简表.....171
表 10-3 FM33LX0XX 中断向量表.....173
表 12-1 IWDT 溢出周期表.....193
表 13-1 WWDT 溢出周期表.....201
表 14-1 系统时钟切换控制.....208
表 14-2 主要时钟说明.....209
表 14-3 外设时钟说明.....210
表 14-4RCLP 状态说明.....214
表 14-5 低功耗模式下的时钟源状态说明.....218
表 18-1OPA 引脚列表.....282
表 19-1 比较器 1 引脚列表.....299
表 19-2 比较器 2 引脚列表.....299
表 19-3 比较器 3 引脚列表.....299
表 19-4 比较器 1/2 工作模式.....301
表 19-5 比较器 3 工作模式.....302
表 21-1 I2C 引脚列表.....319
表 21-2I²C 接口时序要求.....326
表 21-3 I2C 从机保留地址定义.....328
表 22-1 UART 引脚列表.....367
表 22-2 UART 类型列表.....369
表 22-3 UART 数据帧格式.....370
表 22-4 DMA 发送中断.....376
表 22-5 常用时钟频率下波特率计算.....377
表 23-1 LPUART 引脚对应表.....390
表 23-2LPUART 数据帧格式.....392
表 23-3 LPUART 数据位调制系数.....393
表 23-4 LPUART DMA 中断说明.....396
表 24-1 SPI 引脚对应表.....406
表 26-1 CAN 引脚列表.....435
表 29-1ENCODER INTERFACE 计数方式.....516
表 30-1ENCODER INTERFACE 计数方式.....572
表 33-1 LPTIM32 引脚映射.....627
表 34-1 LPTIM16 引脚映射.....644
表 34-2 ENCODER INTERFACE 计数方式.....649
表 37-1 帧频率计算公式.....696
表 37-2 典型帧频率和 DF 的关系.....696
表 37-3 LCD 与低功耗模式.....707
表 38-1 ADC 输入通道.....724
表 38-2 温度传感器斜率.....732
表 38-3 ADC 采样时间.....733
表 38-4 ADC 输出位宽与速度.....734
表 38-5 DMA 配置与功能.....744
表 38-6 ADC 与低功耗模式.....750
表 39-1 DAC 相关引脚.....764
表 39-2 DAC 触发源.....767
表 39-3 DAC 与低功耗模式.....771
表 40-1 PGL 相关引脚.....780
表 40-2LUT 查找表.....781
表 40-3 LUT 输出连接.....782
表 40-4 LUT 输出连接.....784
表 41-1GPIO 功能逻辑定义表.....795
表 41-2FCR 定义表.....796
表 41-3 多个数字外设功能选择表.....799

表 41-4 模拟功能通道选择表.....800

图目录

图 1-1 芯片结构框图.....36
图 2-1 FM33LG0x8 LQFP80 封装图.....39
图 2-2 FM33LG0x6 LQFP64 封装图.....40
图 2-3 FM33LG0x5 LQFP48 封装图.....41
图 2-4 FM33LG0x3 QFN32 封装图.....42
图 2-5LQFP80 封装尺寸图.....50
图 2-6LQFP64 封装尺寸图.....52
图 2-7 LQFP48 封装尺寸图.....54
图 3-1FM33LG0xx 测试条件.....60
图 3-2 ADC 差分输入典型 DNL 和 INL.....75
图 3-3 ADC 单端输入典型 DNL 和 INL.....75
图 3-4 3.3V ADC 差分（左）和单端输入（右）典型信噪比.....75
图 3-5 1.8V ADC 差分（左）和单端输入（右）典型信噪比.....76
图 3-6 ADC 通道输入阻抗.....76
图 3-7 DAC 典型输出曲线.....79
图 3-8 DAC 典型静态特性（5V）.....80
图 3-9 DAC 典型静态特性（3.3V）.....80
图 3-10 DAC 典型静态特性（1.8V）.....81
图 3-11 温度传感器输出曲线.....82
图 3-12LCD 片外电容驱动模式电容连接.....90
图 4-1 系统总线示意图.....92
图 4-2 FM33LG04x 总线地址.....93
图 4-3 FM33LG02x 总线地址.....94
图 4-4 FM33LG01x 总线地址.....95
图 4-5Bootswap 示意图.....102
图 4-6Flash 擦写 Key 认证.....104
图 5-1 芯片电源结构图.....120
图 5-2 功耗模式与系统主频.....123
图 5-3VREF1p2 延迟唤醒时序图.....129
图 6-1 温度传感器输出与标定.....140
图 7-1 ADC 和 DAC 基准源.....146
图 7-2 外部电源基准方案 1.....147
图 7-3 外部独立基准方案.....148
图 7-4ADC 和 DAC 都是用内部基准方案的系统连接.....149
图 7-5 ADC 使用外部基准，DAC 使用内部基准.....150
图 7-6 VREFP 间歇使能波形示意图.....151
图 7-7 VREF1p2 关闭情况下的间歇使能波形示意图.....152
图 7-8 中断标志示意图.....152
图 8-1 备份电源域结构框图.....155
图 8-2 VAO 上下电复位示意图.....156
图 11-1 芯片复位源框图.....181
图 11-2 上下电复位示意图.....182
图 12-1 IWDT 结构框图.....192
图 12-2 IWDT 窗口示意图.....194
图 13-1WWDT 结构框图.....199
图 13-2WWDT 窗口示意图.....201
图 14-1 芯片时钟框图.....207
图 14-2 时钟校准电路框图.....217
图 15-1 低压检测电路框图.....237

图 15-2 低压检测电路工作时序.....238
图 15-3 电源检测电路间歇工作模式.....239
图 16-1 ECB 模式加密流程.....250
图 16-2 ECB 模式解密流程.....251
图 16-3 CBC 加密过程.....252
图 16-4 CBC 解密过程.....253
图 16-5 暂停模式流程.....253
图 16-6 CTR 加密流程.....254
图 16-7 CTR 解密流程.....255
图 16-832 位计数器和随机数的存储方式.....255
图 16-9 GCM 加密流程.....257
图 16-10GCM 解密流程.....258
图 16-11 MULTH 模块框图.....259
图 16-12 根据数据类型存储数据的示意图.....261
图 16-13 模式 1：加密流程.....262
图 16-14 模式 2 示意图.....262
图 16-15 模式 3 示意图.....263
图 16-16 模式 4 示意图.....264
图 16-17 MULTH 模块使用流程示意图.....265
图 16-18 输入时 DMA 请求和数据传输示意图.....265
图 16-19 输出时 DMA 请求和数据传输示意图.....266
图 17-1 真随机数模块框图.....274
图 17-2 真随机数模块工作时钟.....274
图 18-1OPA1 电路框图.....281
图 18-2OPA 用作 ADC 前端放大.....282
图 18-3OPA 非反相放大.....283
图 18-4 OPA 反相放大.....284
图 18-5OPA 缓冲器模式.....285
图 18-6 同相 PGA 模式.....286
图 18-7OPA 环路滤波.....287
图 18-8 反相 PGA 带 DAC 偏置.....288
图 18-9 反相 PGA 带外部偏置.....289
图 19-1 比较器电路框图.....297
图 19-2 比较器内建基准缓冲器.....298
图 19-3 窗口比较器框图.....300
图 19-4 窗口比较器波形示意图.....301
图 19-5 比较器中断产生.....302
图 19-6 比较器输出逻辑.....303
图 19-7 比较器输出上升沿产生触发输出，无滤波.....303
图 19-8 数字滤波（DFLEN=3）波形示意图.....304
图 21-1 I2C 模块框图.....318
图 21-2I²C 总线时序.....324
图 21-3 数据有效时序.....324
图 21-4 起始（START）与停止(STOP)命令定义.....324
图 21-5 输出应答(ACK).....325
图 21-6 从机信号滤波.....330
图 21-7 主机向 7 位地址从机写入数据时的帧格式.....331
图 21-8I2C 软件发送数据流图.....332
图 21-9 I2C 主机对 7 位地址从机发送数据流图.....333
图 21-10 主机从 7 位地址从机读取数据时的帧格式.....333
图 21-11I2C 软件发送数据流图.....334
图 21-12 I2C 从 7 位地址从机读取数据流图.....335
图 21-13 双向数据通信帧格式.....335

图 21-14 10BIT 寻址，主机向从机写入数据 336
图 21-15I2C 软件发送数据流图 337
图 21-16 10BIT 寻址，主机从从机读取数据 337
图 21-17I2C 软件发送数据流图 338
图 21-18 I2C 软件发送数据流图 339
图 21-19 I2C 主机 DMA 发送流程图 340
图 21-20 I2C 主机 DMA 接收流程图 342
图 21-21 主机时序控制 344
图 21-22 从机数据发送波形 346
图 21-23 从机数据接收波形 347
图 21-24 从机数据接收波形（SCLSEN=0，接收溢出） 348
图 21-25 I2C 从机 DMA 接收流程图 350
图 21-26 I2C 从机 DMA 发送流程图 351
图 21-27 SDA 输出延迟波形 352
图 22-1 UART 接口时序 366
图 22-2 UART 字符描述 369
图 22-3 位接收 16 倍采样 371
图 22-4 位接收 8 倍采样 372
图 22-5 UART 异步发送波形 1 373
图 22-6 UART 异步发送波形 2 374
图 22-7 UART 异步发送波形 3 374
图 22-8 红外调制波形 378
图 22-9 UART 发送延迟 379
图 23-1 LPUART 结构框图 389
图 23-2 LPUART 工作时钟 391
图 23-3 字符描述 391
图 24-1 SPI 结构框图 405
图 24-2 SPI 数据/时钟时序图（CPHA=0） 407
图 24-3 SPI 数据/时钟时序图（CPHA=1） 407
图 24-44 线半双工写操作 408
图 24-5 4 线半双工读操作（无 DUMMY CYCLE） 408
图 24-6 4 线半双工读操作（有 DUMMY CYCLE） 409
图 24-7 SPI MASTER/SPI SLAVE 互连 410
图 24-8 SPI SSN 时序图（SSNM=1，CPHA=0） 412
图 24-9 SPI SSN 时序图（SSNM=0） 412
图 25-1 7816 结构框图 422
图 25-2 7816 帧结构图 423
图 25-3 7816 接收流程图 424
图 25-4 7816 发送流程图 425
图 26-1 CAN 模块结构框图 434
图 26-2 CAN BIT TIMING 436
图 27-1 DMA 结构框图 462
图 27-2 DMA 寄存器配置 463
图 27-3 DMA 工作流程 464
图 28-1CRC 运算流程图 474
图 28-2 使用 DMA 对 RAM 中的数据进行 CRC 运算 475
图 29-1 高级定时器结构框图 482
图 29-2 预分频从 1 变为 2 的波形 484
图 29-3 预分频从 1 变为 4 的波形 484
图 29-4 向上计数波形，内部时钟不分频 485
图 29-5 向上计数波形，内部时钟 2 分频 486
图 29-6ARPE=0（ATIM_ARR 没有预装载）时的更新事件 486
图 29-7ARPE=1（ATIM_ARR 预装载）时的更新事件 487

图 29-8 向下计数，内部时钟不分频.....488
图 29-9 向下计数，内部时钟 2 分频.....488
图 29-10 向下计数，内部时钟 2 分频.....489
图 29-11 向下计数，不使用重复计数时的更新事件.....489
图 29-12 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6.....490
图 29-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢).....491
图 29-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出).....491
图 29-15 不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置.....492
图 29-16 ATIM 时钟源框图.....494
图 29-17 内部时钟源模式，时钟分频因子为 1.....494
图 29-18TI2 外部时钟连接例子.....495
图 29-19 外部时钟模式 1 下的时序.....495
图 29-20 外部时钟模式 1 下的时序.....496
图 29-21 外部触发输入框图.....497
图 29-22 外部时钟模式 2 下的时序 1.....497
图 29-23 外部时钟模式 2 下的时序 2.....498
图 29-24 捕获/比较通道(通道 1 输入部分).....500
图 29-25 捕获/比较通道 1 的主电路.....500
图 29-26 捕获/比较通道的输出部分(通道 1 至 3).....501
图 29-27 捕获/比较通道的输出部分(通道 4).....501
图 29-28PWM 输入捕获模式时序.....502
图 29-29 输出比较模式，翻转 OC1.....505
图 29-30 边沿对齐的 PWM 波形(ARR=7).....506
图 29-31 中央对齐的 PWM 波形(APR=7).....507
图 29-32 带死区插入的互补输出.....508
图 29-33 死区波形延迟大于负脉冲.....508
图 29-34 死区波形延迟大于正脉冲.....508
图 29-35 响应刹车的输出.....510
图 29-36 产生六步 PWM，使用 COM 的例子(OSSR=1).....512
图 29-37 单脉冲模式的例子.....513
图 29-38 ETR 信号清除 ATIM 的 OCxREF.....515
图 29-39 编码器模式下的计数器操作实例.....516
图 29-40 复位模式下的时序.....518
图 29-41 门控模式下的时序.....519
图 29-42 触发器模式下的时序.....520
图 29-43 外部时钟模式 2＋触发模式下的时序.....520
图 30-1 通用定时器架构示意图.....548
图 30-2 预分频从 1 变为 2 的波形.....550
图 30-3 预分频从 1 变为 4 的波形.....550
图 30-4 向上计数波形，内部时钟不分频.....551
图 30-5 向上计数波形，内部时钟 2 分频.....552
图 30-6ARPE=0（GPTIM_ARR 没有预装载）时的更新事件.....552
图 30-7ARPE=1（GPTIM_ARR 预装载）时的更新事件.....553
图 30-8 向下计数，内部时钟不分频.....554
图 30-9 向下计数，内部时钟 2 分频.....554
图 30-10 向下计数，内部时钟 2 分频.....555
图 30-11 向下计数，不使用重复计数时的更新事件.....555
图 30-12 中心对齐计数器时序图，GPTIM_PCS=0，GPTIM_ARR=0x6.....556
图 30-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢).....557
图 30-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出).....557
图 30-15GPTIM 时钟源框图.....558
图 30-16 内部时钟源模式，时钟分频因子为 1.....558
图 30-17TI2 外部时钟连接例子.....559

图 30-18 外部时钟模式 1 下的时序.....559
图 30-19 外部时钟模式 1 下的时序.....560
图 30-20 外部触发输入框图.....561
图 30-21 外部时钟模式 2 下的时序 1.....561
图 30-22 外部时钟模式 2 下的时序 2.....562
图 30-23 捕获/比较通道(通道 1 输入部分).....565
图 30-24 捕获/比较通道 1 的主电路.....566
图 30-25 捕获/比较通道的输出部分.....566
图 30-26PWM 输入捕获模式时序.....567
图 30-27 输出比较模式，翻转 OC1.....568
图 30-28 边沿对齐的 PWM 波形(ARR=7).....569
图 30-29 中央对齐的 PWM 波形(APR=7).....570
图 30-30 单脉冲模式的例子.....571
图 30-31 ETR 信号清除 GPTIM 的 OCxREF.....572
图 30-32 编码器模式下的计数器操作实例.....573
图 30-33 复位模式下的时序.....574
图 30-34 门控模式下的时序.....575
图 30-35 触发器模式下的时序.....575
图 30-36 外部时钟模式 2＋触发模式下的时序.....576
图 31-1 32 位基本定时器结构框图.....600
图 31-2 预分频从 1 变为 2 的波形.....602
图 31-3 预分频从 1 变为 4 的波形.....602
图 31-4 向上计数波形，内部时钟不分频.....603
图 31-5 向上计数波形，内部时钟 2 分频.....604
图 31-6ARPE=0（ARR 没有预装载）时的更新事件.....604
图 31-7 ARPE=1（ARR 预装载）时的更新事件.....605
图 31-8 内部时钟源模式，时钟分频因子为 1.....605
图 32-1 16 位基本定时器结构框图.....612
图 32-2 预分频从 1 变为 2 的波形.....614
图 32-3 预分频从 1 变为 4 的波形.....614
图 32-4 向上计数波形，内部时钟不分频.....615
图 32-5 向上计数波形，内部时钟 2 分频.....616
图 32-6 ARPE=0（ARR 没有预装载）时的更新事件.....616
图 32-7 ARPE=1（ARR 预装载）时的更新事件.....617
图 32-8 内部时钟源模式，时钟分频因子为 1.....617
图 33-1 LPTIM32 结构框图.....626
图 33-2 外部 ETR 脉冲上升沿触发计数.....628
图 33-3 外部 ETR 脉冲异步计数（下降沿）.....628
图 33-4TIMEOUT 模式.....629
图 33-5PWM 输出.....630
图 33-6 输入信号边沿捕捉.....631
图 34-1 LPTIM16 结构框图.....643
图 34-2 外部 ETR 脉冲上升沿触发计数.....645
图 34-3 外部 ETR 脉冲异步计数（下降沿）.....645
图 34-4TIMEOUT 模式.....646
图 34-5PWM 输出.....647
图 34-6 输入信号边沿捕捉.....648
图 34-7 通道输入数字滤波.....648
图 34-8 编码器模式下的计数器操作实例.....649
图 35-1RTC 结构框图.....660
图 35-2LTBC 结构框图.....661
图 35-3RTC 时间读取流程图.....665
图 36-1RTC_B 结构框图.....676

图 36-2LTBC 结构框图 677
图 36-3RTC 时间读取流程图 680
图 36-4 时间戳 681
图 36-5 TAMPER 输入数字滤波 682
图 37-1 LCD 显示控制模块结构框图 695
图 37-2LCD 驱动波形(1/4 DUTY, 1/3 BIAS, TYPE A) 697
图 37-3LCD 驱动波形(1/6 DUTY, 1/4 BIAS, TYPE A) 698
图 37-4LCD 驱动波形(1/6 DUTY, 1/3 BIAS, TYPE A) 699
图 37-5LCD 驱动波形(1/8 DUTY, 1/4 BIAS, TYPE A) 700
图 37-6 LCD 驱动波形(1/4 DUTY, 1/3 BIAS, TYPE B) 701
图 37-7 LCD 驱动波形(1/6 DUTY, 1/4 BIAS, TYPE B) 702
图 37-8 LCD 驱动波形(1/6 DUTY, 1/3 BIAS, TYPE B) 703
图 37-9 LCD 驱动波形(1/8 DUTY, 1/4 BIAS, TYPE B) 704
图 37-10LCD 片内电阻 BUFFER 型驱动电路 705
图 38-1 ADC 结构框图 723
图 38-2 单端输入 725
图 38-3 差分输入 725
图 38-4 ADC 校准时序 727
图 38-5 ADC 采样转换时序 727
图 38-6 ADC 采样序列时序 728
图 38-7 ADC 输入通道示意图 733
图 38-8 VBAT/VDD 分压电路示意图 735
图 38-9 ADC 单次转换全自动触发模式 737
图 38-10 ADC 单次转换半自动触发模式 738
图 38-11 ADC 连续转换模式 739
图 38-12 ADC 硬件触发源 740
图 38-13 ADC 时钟示意图 741
图 38-14 ADC 自动等待 742
图 38-15 ADC 单次全自动触发+DMA 案例 1 744
图 38-16 ADC 单次全自动触发+DMA 案例 2 745
图 38-17 ADC 单次半自动触发+DMA 746
图 38-18 ADC 全自动触发+DMA 循环模式 747
图 38-19 ADC 连续模式+DMA 循环模式 748
图 38-20 ADC 模拟看门狗阈值示意图 749
图 39-1 DAC 结构框图 763
图 39-2 DAC 结构框图 764
图 39-3 DAC 工作时序 765
图 39-4 DAC 连续输出模式 766
图 39-5 DAC 触发输出模式 767
图 39-6 触发模式下通过 DMA 更新数据 768
图 39-7 DAC 采样保持输出 769
图 39-8 DAC 采样保持输出过程中更新 DHR（TRGEN=0） 770
图 39-9 DAC 采样保持输出过程中更新 DHR（TRGEN=1） 770
图 40-1LUT 结构框图 779
图 40-2 PGL 结构框图 779
图 40-3 LUT 实现 2 输入 NAND 示意图 782
图 40-4 输出滤波和采样 783
图 40-5 数字滤波 783
图 41-1 普通 GPIO 结构框图 795
图 41-2 普通 GPIO（两路上拉）结构框图 796
图 41-3WKUPx 功能结构框图 801
图 41-4 引脚输入数字滤波 802
图 41-5 EXTI 信号输入示意图 803

图 43-1 CORTEX-M0 调试系统示意图.....825

1 产品综述

1.1 概述

FM33LG0的主要特性如下：

- 宽电压范围：1.65~5.5V
- 工作温度范围（T_A）：-40℃~+85℃
- 处理器内核
  - ARM Cortex-M0+
  - 支持MPU
  - 支持用户/特权模式
  - 最高主频64Mhz
  - SWD调试接口
  - 24bit Systick定时器
- 低功耗技术平台
  - 典型运行功耗130uA/MHz@48MHz
  - VBAT备份电源切换
  - Sleep模式：5uA typ
  - DeepSleep模式，RTC走时+全部RAM保持+CPU内核保持：1.5uA typ
  - VBAT模式，RTC走时+备份寄存器：0.8uA typ
- 存储器
  - 64/128/256KB Flash空间
  - Flash擦写寿命：100,000次
  - Flash数据保存时间：10年@85℃
  - 用户代码保护
  - 16/32/32KB RAM空间
- 丰富的模拟外设
  - 高可靠、可配置BOR电路（支持4级可编程下电复位阈值）
  - 超低功耗PDR电路（支持4级可编程下电复位阈值）
  - 可编程电源监测模块（SVD）
  - 3x低功耗模拟比较器
  - 12bit 2Msps SAR-ADC
  - 12bit 1Msps DAC
  - 内置基准电压产生电路

- 高精度温度传感器，精度+/-2℃

- 通用通信接口
  - UART*5
  - LPUART*3
  - 7816主机*1
  - SPI*3，主从模式
  - I2C*1，主从模式
  - CAN2.0B*1
  - 7通道外设DMA
  - 可编程CRC校验模块
- 定时资源
  - 16bit高级定时器*1，最高PWM分辨率120MHz
  - 16bit通用定时器*3
  - 32bit基本定时器*1，16bit基本定时器*1
  - 24-bit Systick*1
  - 32-bit低功耗定时器*1，16bit低功耗定时器*1
  - 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
- LCD显示控制电路
  - 最大支持4COM×44SEG / 6COM×42SEG / 8COM×40SEG
  - 1/3 bias、1/4bias
  - 片内电阻分压
  - 支持休眠显示
- 安全算法
  - 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%，8MHz全温区精度小于+/-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，4~32MHz
  - PLL，最高输出64MHz，非分频输出128MHz

- 封装：LQFP80/64/48，QFN32，TSSOP20

1.2 芯片结构框图

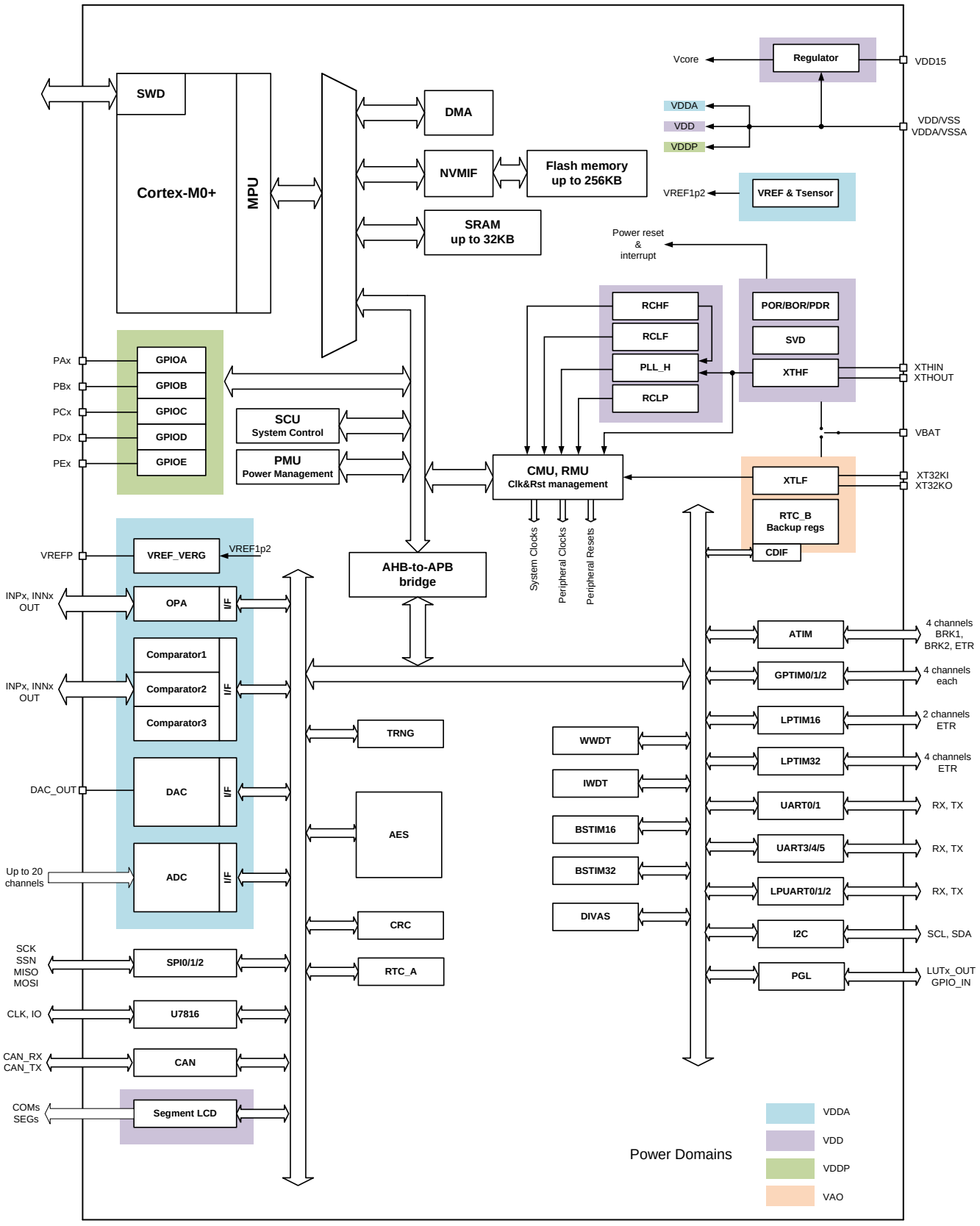


图 1-1 芯片结构框图

1.3 产品型号列表

型号	Flash 容量（KBytes）	RAM 容量（KBytes）	封装
FM33LG048	256	32	LQFP80
FM33LG046	256	32	LQFP64
FM33LG045	256	32	LQFP48
FM33LG043	256	32	QFN32
FM33LG026	128	32	LQFP64
FM33LG025	128	32	LQFP48
FM33LG023	128	32	QFN32
FM33LG016	64	16	LQFP64
FM33LG013	64	16	QFN32
FM33LG012	64	16	TSSOP20

表 1-1FM33LG0 型号列表

1.4 产品特性对照表

型号		FM33LG048	FM33LG046	FM33LG045	FM33LG043	FM33LG028	FM33LG026	FM33LG025	FM33LG023	FM33LG016	FM33LG015	FM33LG013
CPU		Cortex-M0+										
MPU		Y										
Max Freq.		64MHz										
Flash		256KB				128KB				64KB		
RAM		32KB				32KB				16KB		
AES		1										
RNG		1										
Timer	ATIM	1										
	GTIM	3										
	BSTIM32	1										
	BSTIM16	1										
	LPTIM32	1										
	LPTIM16	1										
	systick	1										
RTC/WWDT/IWDT		2/1/1										
SPI		3	3	2	2	3	3	2	2	3	2	2
I2C		1	1	1	1	1	1	1	1	1	1	1
UART		5	5	5	4	5	5	5	4	5	5	4
LPUART		3	3	3	2	3	3	3	2	3	3	2
ISO7816		1	1	-	-	1	1	-	-	1	-	-
GPIO		71	56	40	26	71	56	40	26	56	40	26
LCD		4*44 6*42 8*40	4*36 6*34 8*32	4*25 6*23 8*21	-	4*44 6*42 8*40	4*36 6*34 8*32	4*25 6*23 8*21	-	4*36 6*34 8*32	4*25 6*23 8*21	-
OPA		1	1	1	1	1	1	1	1	1	1	1
12bit SAR-ADC		20ch	18ch	9ch	8ch	20ch	18ch	9ch	8ch	18ch	9ch	8ch
TempSensor		1										

表 1-2 FM33LG0xx 特性列表

2 引脚和封装

2.1 封装和引脚排列

2.1.1 LQFP80 封装图（FM33LG0x8）

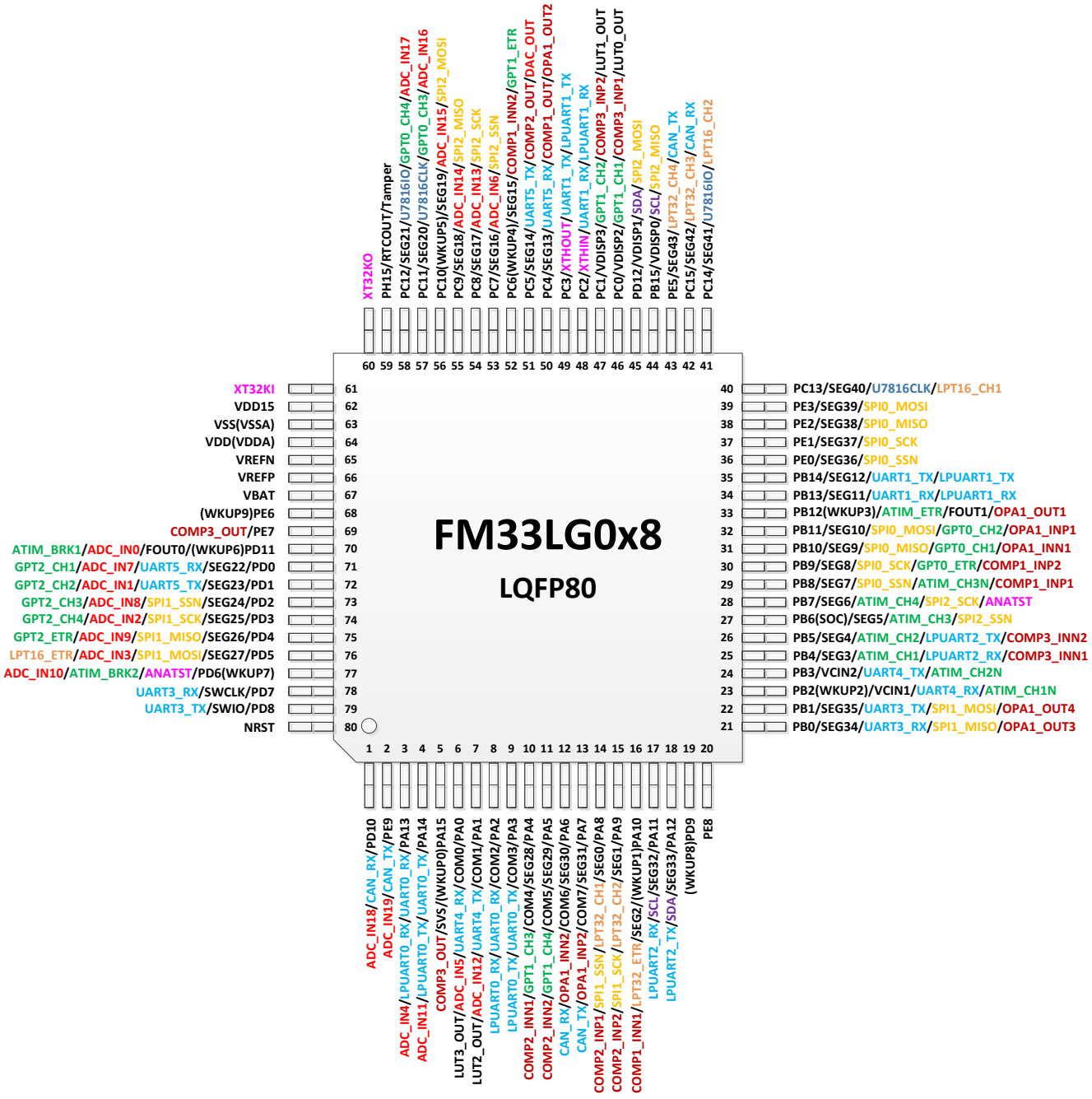


图 2-1 FM33LG0x8 LQFP80 封装图

典型资源：

LCD 4*44

ADC 21个外部通道

支持VBAT，有Tamper引脚（PH15）

2.1.2 LQFP64 封装图（FM33LG0x6）

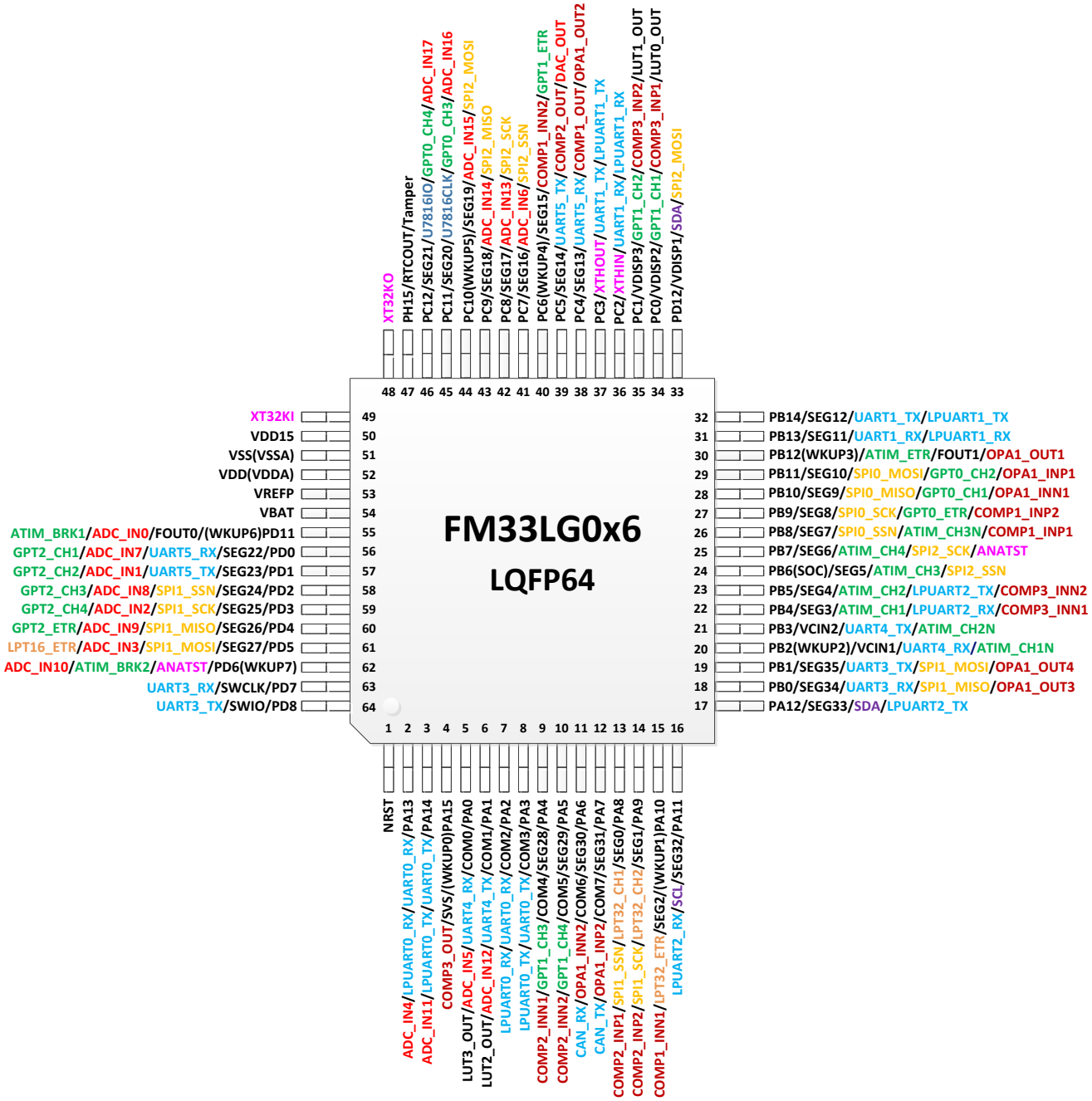


图 2-2 FM33LG0x6 LQFP64 封装图

典型资源：

LCD 4*36

ADC 18个外部通道

支持VBAT，有Tamper引脚（PH15）

2.1.3 LQFP48 封装图（FM33LG0x5）

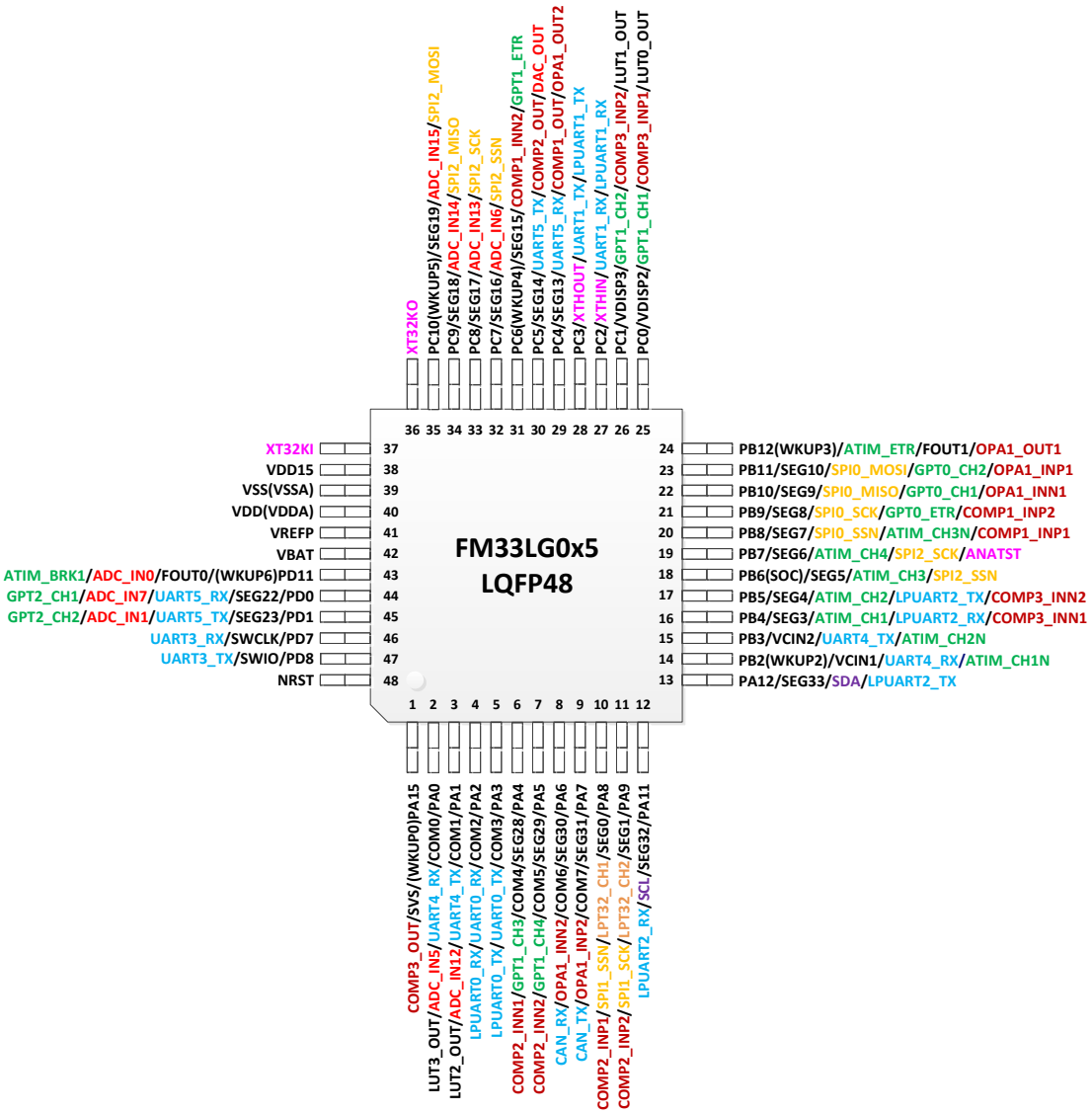


图 2-3 FM33LG0x5 LQFP48 封装图

典型资源：

LCD 4*25, 6*23, 8*21

ADC 9个外部通道

支持VBAT

2.1.4 QFN32 封装图（FM33LG0x3）

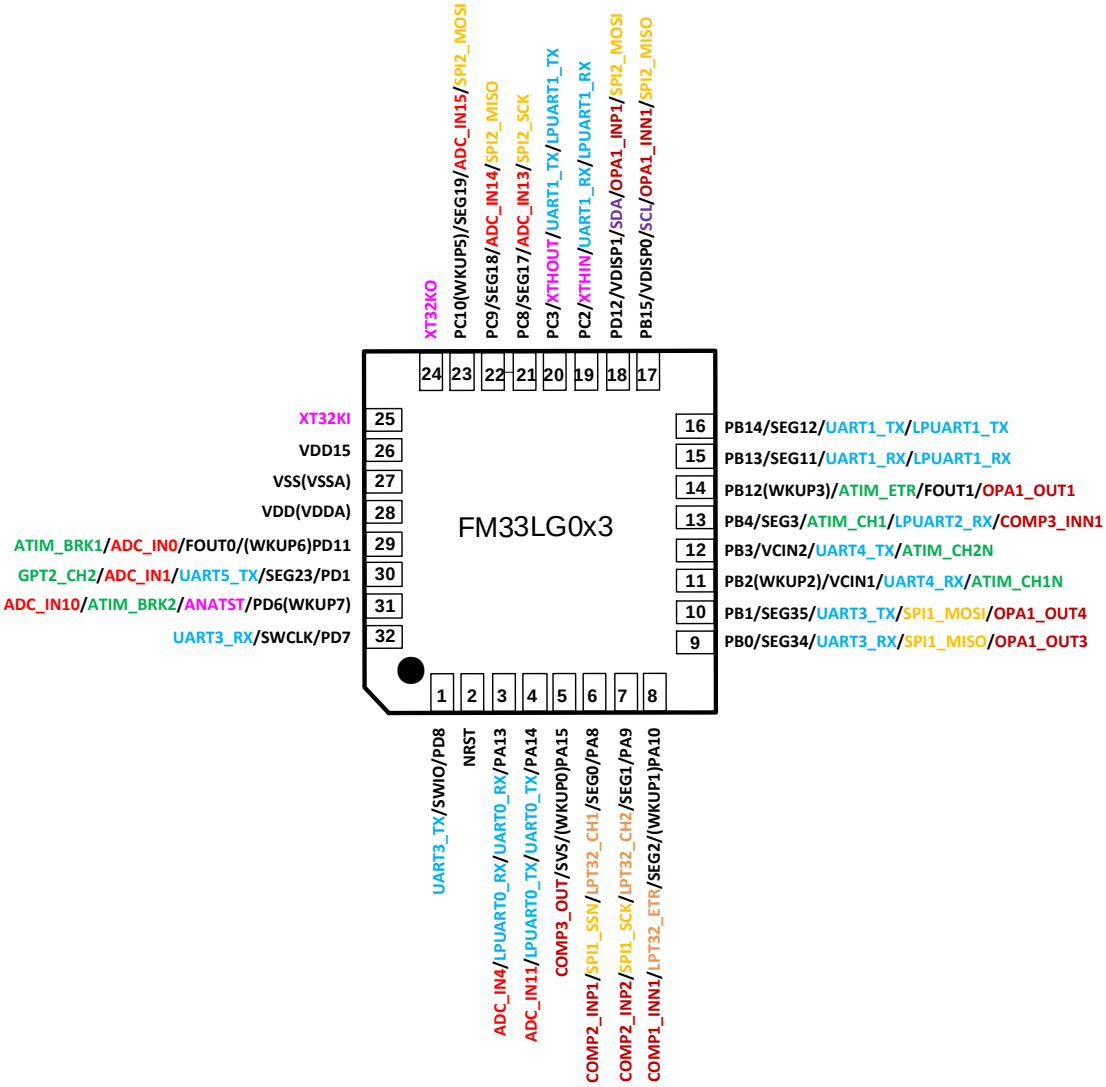


图 2-4 FM33LG0x3 QFN32 封装图

2.1.5 引脚功能定义（FM33LG0xx）

Pin Number			Pin Function	Descriptions
LQFP80	LQFP64	LQFP48		
1	-	-	PD10	GPIO
			CAN_RX	CAN 接收
			ADC_IN18	ADC 输入通道
2	-	-	PE9	GPIO
			CAN_TX	CAN 发送
			ADC_IN19	ADC 输入通道
3	2	-	PA13	GPIO
			UART0_RX	UART 接收
			LPUART0_RX	低功耗 UART 接收
			ADC_IN4	ADC 输入通道
4	3	-	PA14	GPIO
			UART0_TX	UART 发送
			LPUART0_TX	低功耗 UART 发送
			ADC_IN11	ADC 输入通道
5	4	1	PA15	GPIO
			WKUP0	外部唤醒引脚
			SVS	外部电源检测
			COMP3_OUT	比较器输出
6	5	2	PA0	GPIO
			COM0	LCD 驱动 COM 端
			UART4_RX	UART 接收
			ADC_IN5	ADC 输入通道
			LUT3_OUT	PGL 输出
7	6	3	PA1	GPIO
			COM1	LCD 驱动 COM 端
			UART4_TX	UART 发送
			ADC_IN12	ADC 输入通道
			LUT2_OUT	PGL 输出
8	7	4	PA2	GPIO
			COM2	LCD 驱动 COM 端
			UART0_RX	UART 接收
			LPUART0_RX	低功耗 UART 接收
9	8	5	PA3	GPIO
			COM3	LCD 驱动 COM 端
			UART0_TX	UART 发送
			LPUART0_TX	低功耗 UART 发送
10	9	6	PA4	GPIO
			COM4/SEG28	LCD 驱动 COM/SEG 端
			GPT1_CH3	通用定时器外部通道
			COMP2_INN1	比较器输入
11	10	7	PA5	GPIO
			COM5/SEG29	LCD 驱动 COM/SEG 端
			GPT1_CH4	通用定时器外部通道
			COMP2_INN2	比较器输入

Pin Number			Pin Function	Descriptions
LQFP80	LQFP64	LQFP48		
12	11	8	PA6	GPIO
			COM6/SEG30	LCD 驱动 COM/SEG 端
			OPA1_INN2	OPA 输入通道
			CAN_RX	CAN 接收
13	12	9	PA7	GPIO
			COM7/SEG31	LCD 驱动 COM/SEG 端
			OPA1_INP2	OPA 输入通道
14	13	10	PA8	GPIO
			SEG0	LCD 驱动 SEG 端
			LPT32_CH1	低功耗定时器外部通道
			COMP2_INP1	比较器输入
15	14	11	PA9	GPIO
			SEG1	LCD 驱动 SEG 端
			LPT32_CH2	低功耗定时器外部通道
			COMP2_INP2	比较器输入
16	15	-	PA10	GPIO
			WKUP1	外部唤醒引脚
			SEG2	LCD 驱动 SEG 端
			LPT32_ETR	低功耗定时器外部触发输入
			COMP1_INN1	比较器输入
17	16	12	PA11	GPIO
			SEG32	LCD 驱动 SEG 端
			SCL	I2C 时钟
			LPUART2_RX	LPUART 接收
18	17	13	PA12	GPIO
			SEG33	LCD 驱动 SEG 端
			SDA	I2C 数据
			LPUART2_TX	LPUART 发送
19	-	-	PD9	GPIO
			WKUP8	外部唤醒引脚
20	-	-	PE8	GPIO
21	18	-	PB0	GPIO
			SEG34	LCD 驱动 SEG 端
			UART3_RX	UART 接收
			SPI1_MISO	SPI 数据
			OPA1_OUT3	OPA 输出
22	19	-	PB1	外部唤醒引脚
			SEG35	LCD 驱动 SEG 端
			UART3_TX	UART 发送
			SPI1_MOSI	SPI 数据
			OPA1_OUT4	OPA 输出
23	20	14	PB2	GPIO
			WKUP2	外部唤醒引脚
			VCIN1	LCD 电容引脚（电容驱动模式）
			UART4_RX	UART 接收

Pin Number			Pin Function	Descriptions
LQFP80	LQFP64	LQFP48		
			ATIM_CH1N	高级定时器外部通道
24	21	15	PB3	GPIO
			VCIN2	LCD 电容引脚（电容驱动模式）
			UART4_TX	UART 接收
			ATIM_CH2N	高级定时器外部通道
25	22	16	PB4	GPIO
			SEG3	LCD 驱动 SEG 端
			ATIM_CH1	高级定时器外部通道
			LPUART2_RX	LPUART 接收
			COMP3_INN1	比较器输入
26	23	17	PB5	GPIO
			SEG4	LCD 驱动 SEG 端
			ATIM_CH2	高级定时器外部通道
			LPUART2_TX	LPUART 发送
			COMP3_INN2	比较器输入
27	24	18	PB6	GPIO
			SEG5	LCD 驱动 SEG 端
			ATIM_CH3	高级定时器外部通道
			SPI2_SSN	SPI 片选
28	25	19	PB7	GPIO
			SEG6	LCD 驱动 SEG 端
			ATIM_CH4	高级定时器外部通道
			SPI2_SCK	SPI 时钟
			ANATST	原厂测试通道
29	26	20	PB8	GPIO
			SEG7	LCD 驱动 SEG 端
			SPI0_SSN	SPI 片选
			ATIM_CH3N	高级定时器外部通道
			COMP1_INN1	比较器输入
30	27	21	PB9	GPIO
			SEG8	LCD 驱动 SEG 端
			COMP1_INP1	比较器输入
			SPI0_SCK	SPI 时钟
			GPT0_ETR	通用定时器外部触发输入
31	28	22	PB10	GPIO
			SEG9	LCD 驱动 SEG 端
			OPA1_INN1	OPA 输入
			SPI0_MISO	SPI 数据线
			GPT0_CH1	通用定时器外部通道
32	29	23	PB11	GPIO
			SEG10	LCD 驱动 SEG 端
			OPA1_INP1	OPA 输入
			SPI1_MOSI	SPI 数据线
			GPT0_CH2	通用定时器外部通道
33	30	24	PB12	GPIO
			WKUP3	外部唤醒引脚

Pin Number			Pin Function	Descriptions
LQFP80	LQFP64	LQFP48		
			FOUT1	时钟频率输出
			ATIM_ETR	高级定时器外部触发输入
			OPA1_OUT1	OPA 输出
34	31	-	PB13	GPIO
			SEG11	LCD 驱动 SEG 端
			UART1_RX	UART 接收
			LPUART1_RX	LPUART 接收
35	32	-	PB14	GPIO
			SEG12	LCD 驱动 SEG 端
			UART1_TX	UART 发送
			LPUART1_TX	LPUART 发送
36	-	-	PE0	GPIO
			SEG36	LCD 驱动 SEG 端
			SPI0_SSN	SPI 片选
37	-	-	PE1	GPIO
			SEG37	LCD 驱动 SEG 端
			SPI0_SCK	SPI 时钟
38	-	-	PE2	GPIO
			SEG38	LCD 驱动 SEG 端
			SPI0_MISO	SPI 数据
39	-	-	PE3	GPIO
			SEG39	LCD 驱动 SEG 端
			SPI0_MOSI	SPI 数据
40	-	-	PC13	GPIO
			SEG40	LCD 驱动 SEG 端
			U7816CLK	7816 接口时钟
			LPT16_CH1	LPTIM16 通道
41	-	-	PC14	GPIO
			SEG41	LCD 驱动 SEG 端
			U7816IO	7816 接口数据（100K 上拉）
			LPT16_CH2	LPTIM16 通道
42	-	-	PC15	GPIO
			SEG42	LCD 驱动 SEG 端
			LPT32_CH3	LPTIM32 通道
			CAN_RX	CAN 接口接收
43	-	-	PE5	GPIO
			SEG43	LCD 驱动 SEG 端
			LPT32_CH4	LPTIM32 通道
			CAN_TX	CAN 接口发送
44	-	-	PB15	GPIO
			VDISP0	LCD 片外电容（电容驱动模式）
			SCL	I2C 时钟
			SPI2_MISO	SPI 数据
45	33	-	PD12	GPIO
			VDISP1	LCD 片外电容（电容驱动模式）
			SDA	I2C 数据

Pin Number			Pin Function	Descriptions
LQFP80	LQFP64	LQFP48		
			SPI2_MOSI	SPI 数据
46	34	25	PC0	GPIO
			VDISP2	LCD 片外电容（电容驱动模式）
			COMP3_INP1	模拟比较器输入
			LUT0_OUT	PGL 输出
			GPT1_CH1	通用定时器外部通道
47	35	26	PC1	GPIO
			VDISP3	LCD 片外电容（电容驱动模式）
			COMP3_INP2	模拟比较器输入
			LUT1_OUT	PGL 输出
			GPT1_CH2	通用定时器外部通道
48	36	27	PC2	GPIO
			XTHIN	高频晶振输入
			UART1_RX	UART 接收
			LPUART1_RX	低功耗 UART 接收
49	37	28	PC3	GPIO
			XTHOUT	高频晶振输出
			UART1_TX	UART 发送
			LPUART1_TX	低功耗 UART 发送
50	38	29	PC4	GPIO
			SEG13	LCD 驱动 SEG 端
			OPA1_OUT2	OPA 输出
			COMP1_OUT	比较器输出
			UART5_RX	UART 接收
51	39	30	PC5	GPIO
			SEG14	LCD 驱动 SEG 端
			DAC_OUT	DAC 输出
			COMP2_OUT	比较器输出
			UART5_TX	UART 发送
52	40	31	PC6	GPIO
			WKUP4	外部唤醒引脚
			SEG15	LCD 驱动 SEG 端
			GPT1_ETR	通用定时器外部触发输入
			COMP1_INN2	比较器输入
53	41	32	PC7	GPIO
			SEG16	LCD 驱动 SEG 端
			SPI2_SSN	SPI 片选
			ADC_IN6	ADC 输入通道
54	42	33	PC8	GPIO
			SEG17	LCD 驱动 SEG 端
			SPI2_SCK	SPI 时钟
			ADC_IN13	ADC 输入通道
55	43	34	PC9	GPIO
			SEG18	LCD 驱动 SEG 端
			SPI2_MISO	SPI 数据线
			ADC_IN14	ADC 输入通道

Pin Number			Pin Function	Descriptions
LQFP80	LQFP64	LQFP48		
56	44	35	PC10	GPIO
			WKUP5	外部唤醒引脚
			SEG19	LCD 驱动 SEG 端
			SPI2_MOSI	SPI 数据线
			ADC_IN15	ADC 输入通道
57	45	-	PC11	GPIO
			SEG20	LCD 驱动 SEG 端
			U7816CLK	7816 接口时钟
			GPT0_CH3	通用定时器外部通道
			ADC_IN16	ADC 输入通道
58	46	-	PC12	GPIO
			SEG21	LCD 驱动 SEG 端
			U7816IO	7816 接口数据（10K 上拉）
			GPT0_CH4	通用定时器外部通道
			ADC_IN17	ADC 输入通道
59	47	-	PH15	GPIO
			RTCOUT	RTCB 输出信号
			Tamper	篡改检测信号输入
60	48	36	XT32KO	32768Hz 晶振输出脚
61	49	37	XT32KI	32768Hz 晶振输入脚
62	50	38	VDD15	LDO 输出，外接 100nF 电容到地
63	51	39	VSS	地
64	52	40	VDD	电源
65	-	-	VREFN	基准地
66	53	41	VREFP	基准源
67	54	42	VBAT	备份电源
68	-	-	PE6	GPIO
			WKUP9	唤醒引脚
69	-	-	PE7	GPIO
			COMP3_OUT	比较器输出
70	55	43	PD11	GPIO
			WKUP6	外部唤醒引脚
			FOUT0	时钟频率输出
			ATIM_BRK1	高级定时器刹车输入
			ADC_IN0	ADC 输入通道
71	56	44	PD0	GPIO
			SEG22	LCD 驱动 SEG 端
			UART5_RX	UART 接收
			GPT2_CH1	通用定时器通道
			ADC_IN7	ADC 输入通道
72	57	45	PD1	GPIO
			SEG23	LCD 驱动 SEG 端
			UART5_TX	UART 发送
			GPT2_CH2	通用定时器通道
			ADC_IN1	ADC 输入通道
73	58	-	PD2	GPIO

Pin Number			Pin Function	Descriptions
LQFP80	LQFP64	LQFP48		
			SEG24	LCD 驱动 SEG 端
			SPI1_SSN	SPI 片选
			GPT2_CH3	通用定时器通道
			ADC_IN8	ADC 输入通道
74	59	-	PD3	GPIO
			SEG25	LCD 驱动 SEG 端
			SPI1_SCK	SPI 时钟
			GPT2_CH4	通用定时器通道
			ADC_IN2	ADC 输入通道
75	60	-	PD4	GPIO
			SEG26	LCD 驱动 SEG 端
			SPI1_MISO	SPI 数据
			GPT2_ETR	通用定时器外部触发
			ADC_IN9	ADC 输入通道
76	61	-	PD5	GPIO
			SEG27	LCD 驱动 SEG 端
			SPI1_MOSI	SPI 数据
			LPT16_ETR	LPTIM16 外部触发
			ADC_IN3	ADC 输入通道
77	62	-	PD6	GPIO
			WKUP7	外部唤醒引脚
			ANATST	模拟测试通道
			ATIM_BRK2	高级定时器刹车输入
			ADC_IN10	ADC 输入通道
78	63	46	PD7	GPIO
			UART3_RX	UART 接收
			SWCLK	SWD 接口时钟
79	64	47	PD8	GPIO
			UART3_TX	UART 发送
			SWIO	SWD 接口数据
80	1	48	NRST	复位引脚

表 2-1 引脚列表

2.1.6 封装尺寸图

2.1.6.1 LQFP80

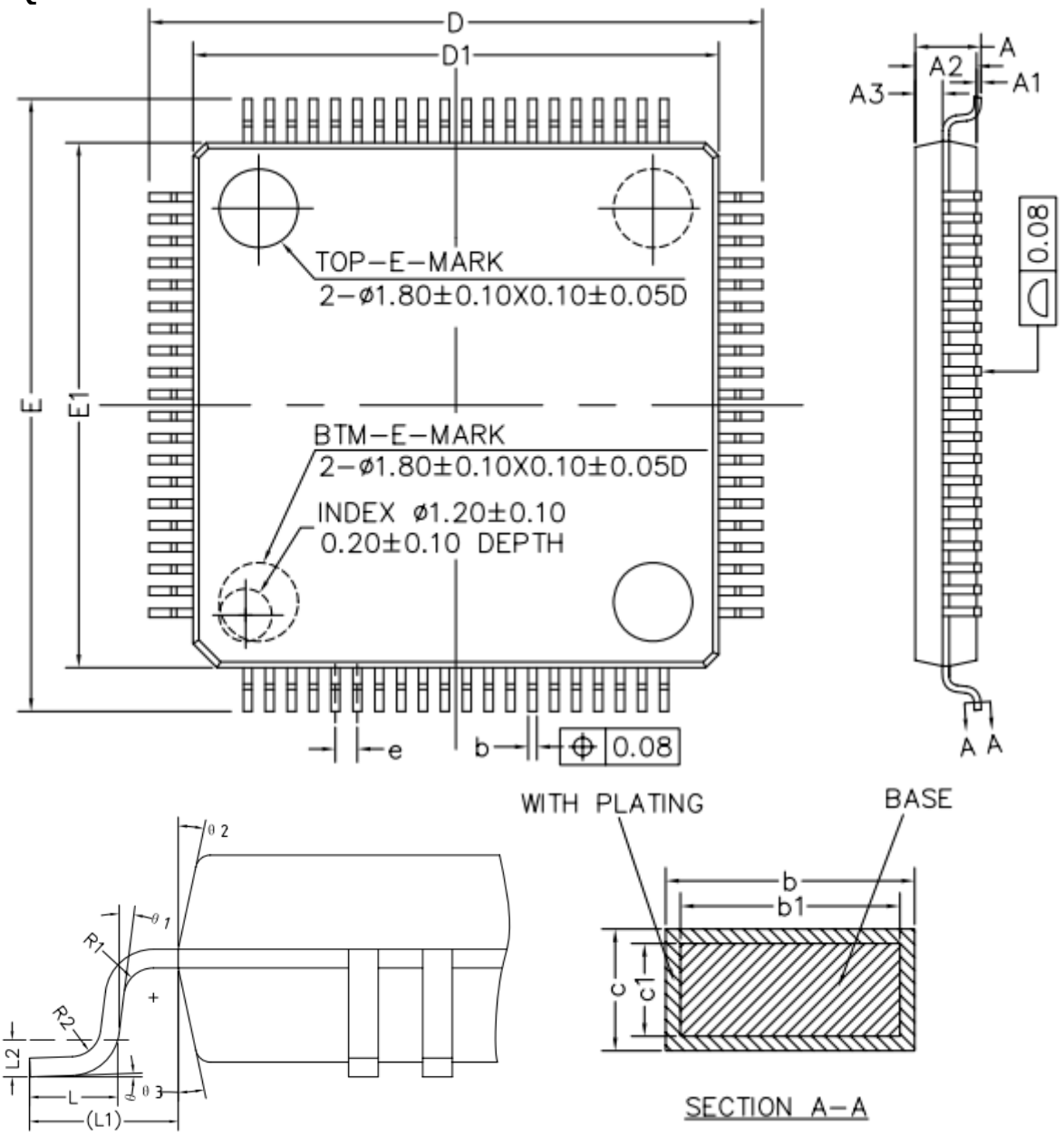


图 2-5LQFP80 封装尺寸图

Symbol	MIN	NOM	MA
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	13.80	14.00	14.20

Symbol	MIN	NOM	MA
D1	11.90	12.00	12.10
E	13.80	14.00	14.20
E1	11.90	12.00	12.10
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 MO-220 WMMD-4.

2.1.6.2 LQFP64

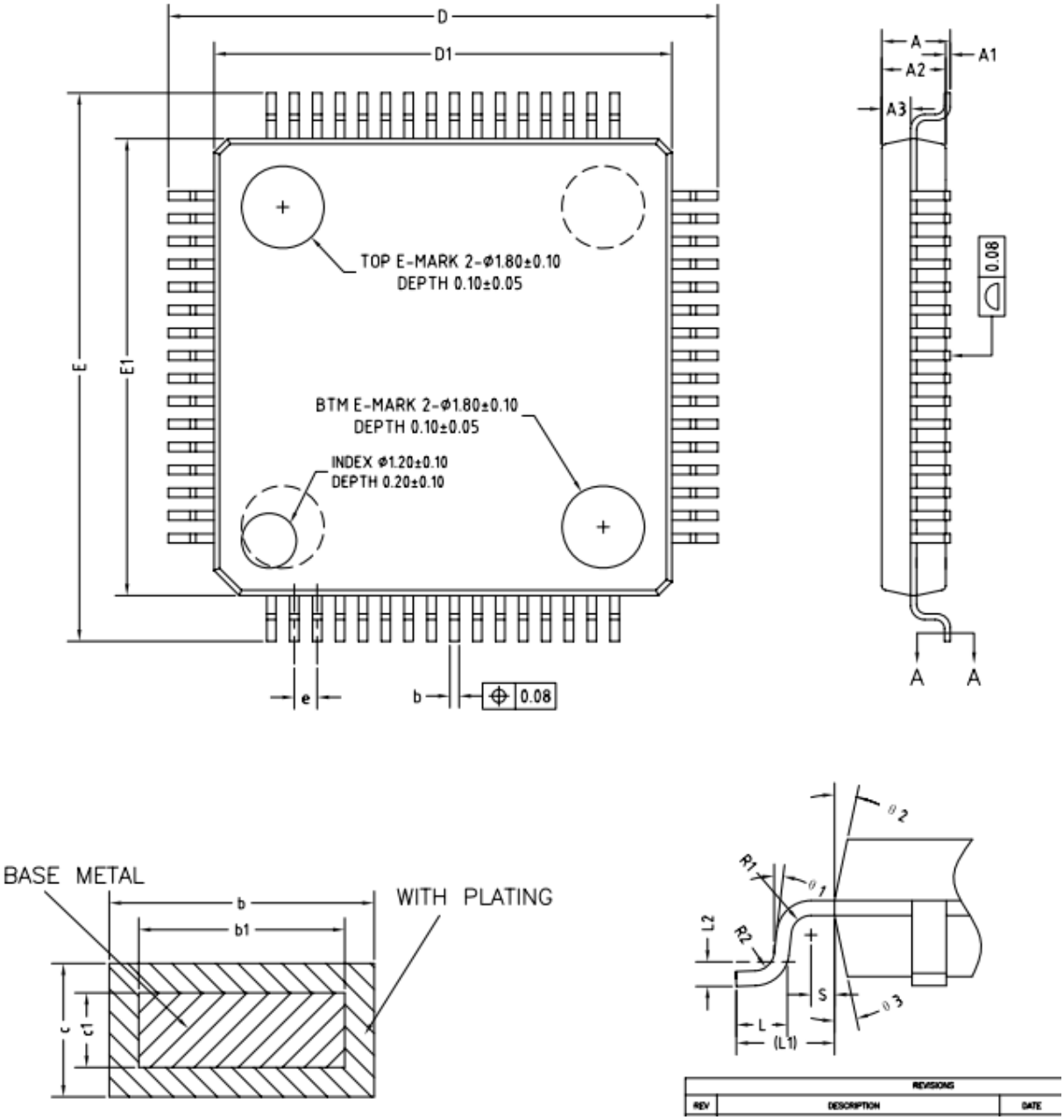


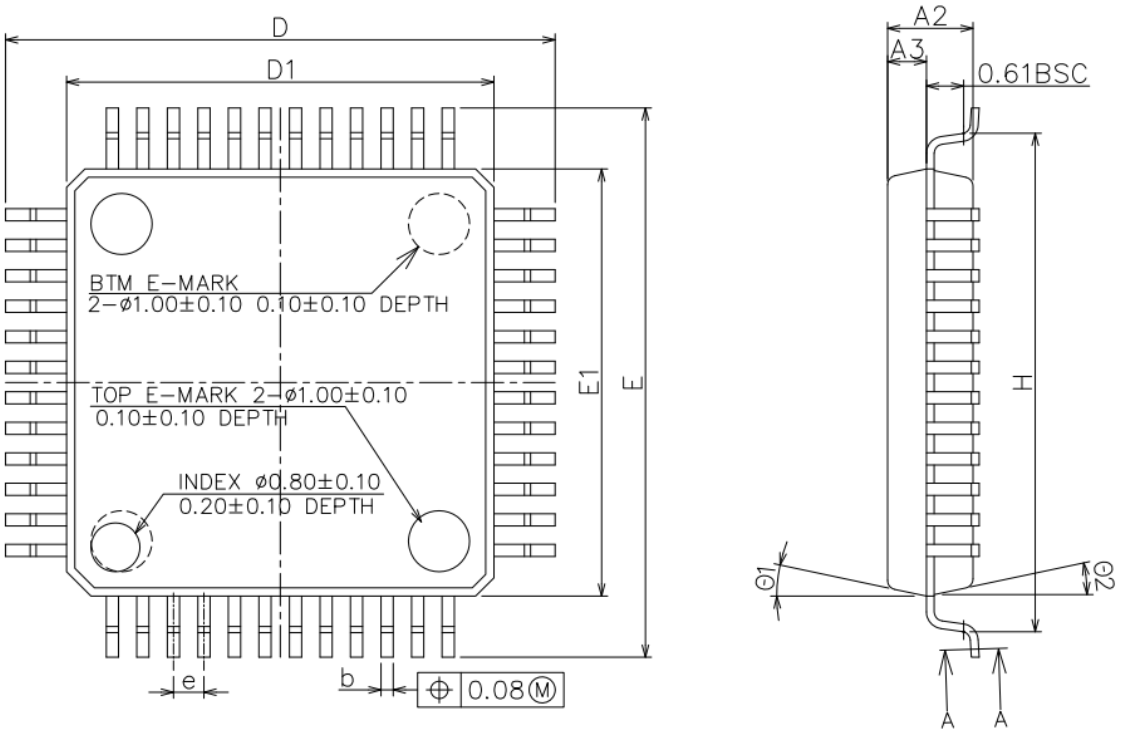
图 2-6LQFP64 封装尺寸图

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.

2.1.6.3 LQFP48



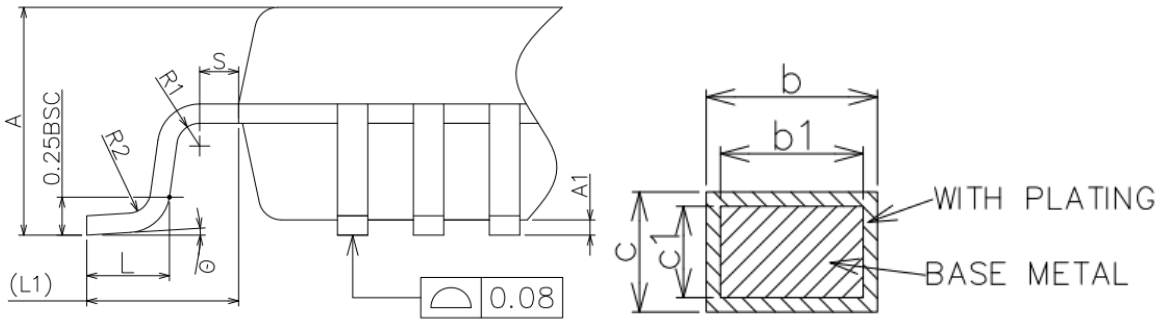


图 2-7 LQFP48 封装尺寸图

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	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
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 MS-026 BDD.

2.1.6.4 QFN32

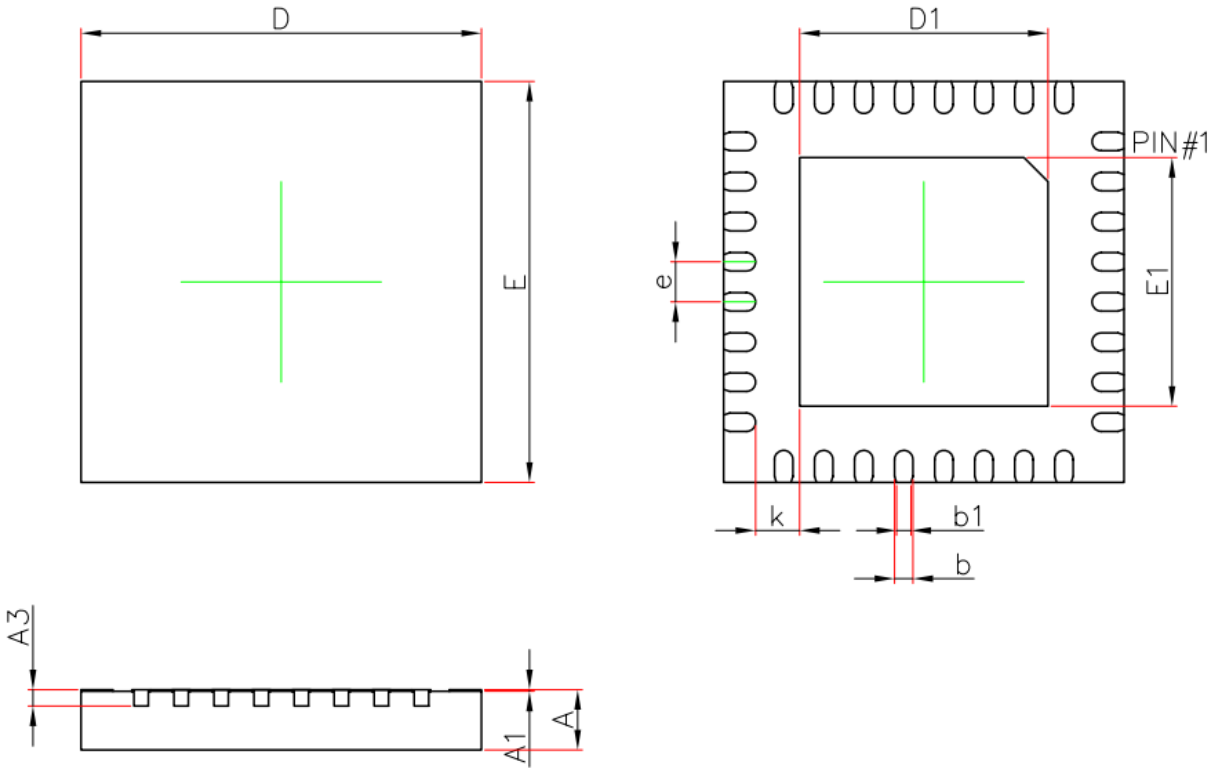


图 1-8 QFN32 封装尺寸图

Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min.	Max.	Min.	Max.
A	0.700	0.800	0.028	0.031
A1	0.000	0.050	0.000	0.002
A3	0.203 REF.		0.008 REF.	
b	0.180	0.300	0.007	0.012
b1	0.130	0.230	0.005	0.009
D	4.900	5.100	0.193	0.201
D1	3.000	3.200	0.118	0.126
E	4.900	5.100	0.193	0.201
E1	3.000	3.200	0.118	0.126
e	0.500 BSC.		0.020 BSC.	
k	0.550 REF.		0.022 REF.	
L	0.324	0.476	0.013	0.019

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

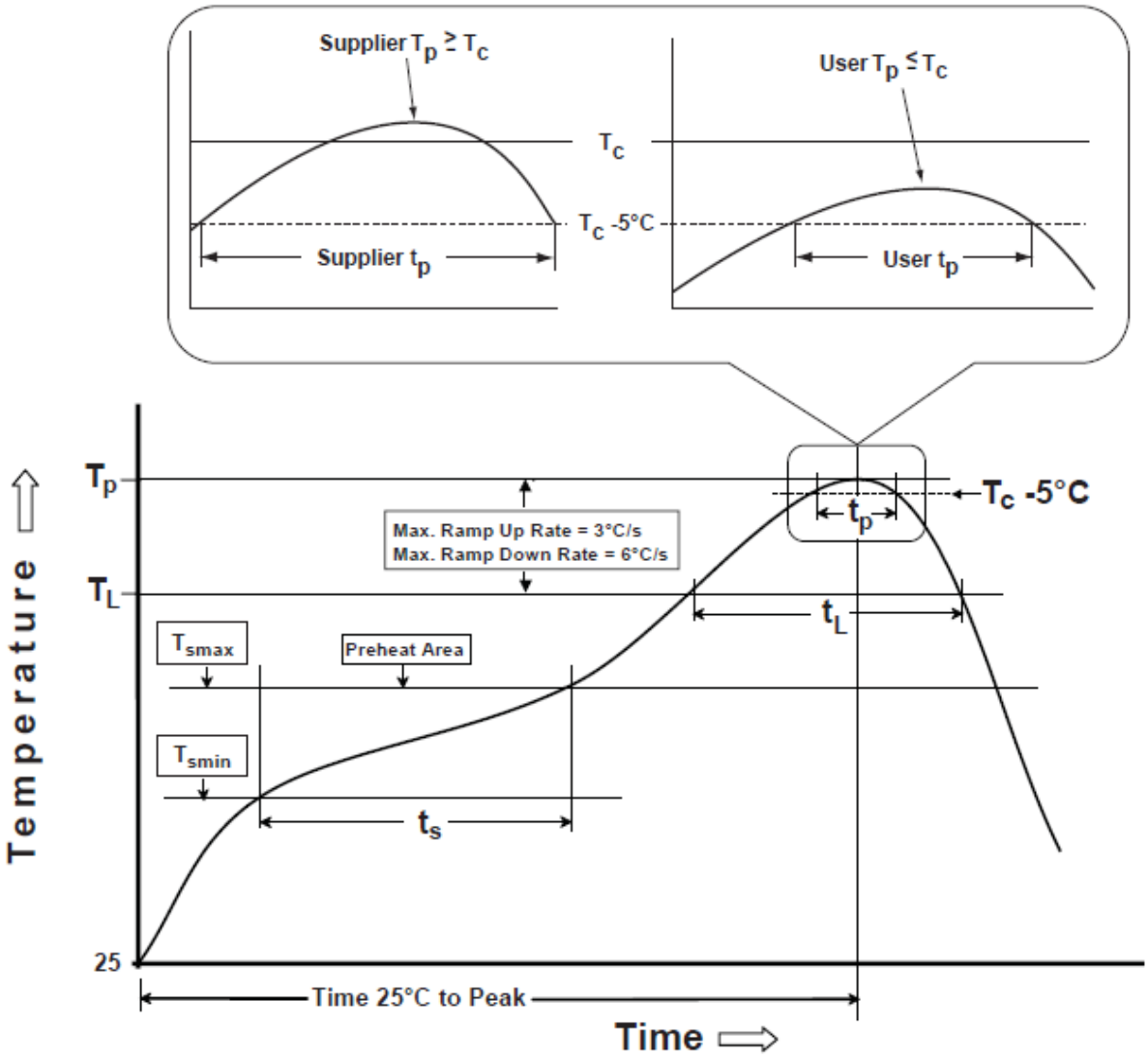


图 1-7 JEDEC 标准的耐热回流温度曲线

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

FM33LG0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。

2.4 热阻特性

芯片结温（T_J）可以通过以下公式计算。

$$T_J = T_A + P_D \times \Theta_{JA}$$

其中：

- T_A 为工作环境温度，单位°C
- Θ_{JA} 是封装热阻系数，单位°C/W
- P_D 是芯片功耗，包含内核功耗和 IO 功耗，单位 W；其中 IO 功耗由下式计算：

$$P_{IO} = \Sigma(V_{OL} \times I_{OL}) + \Sigma((V_{DDIO} - V_{OH}) \times I_{OH})$$

不同封装形式的热阻系数可以参考下表：

封装形式	封装尺寸	参考热阻 Θ_{JA}（℃/W）
LQFP80	12x12x1.4mm	50
LQFP100	14x14x1.4mm	45
LQFP64	10x10x1.4mm	48
LQFP48	7x7x1.4mm	55
QFN32	5x5x0.75mm	35

举例：

- 假设平均工作环境温度 T_A=55°C
- 芯片内核电流 IDD=5mA，VDD=3.6V
- 10 个 IO 同时输出低电平，每个 IO sink 5mA，VOL=0.3V
- 10 个 IO 同时输出高电平，每个 IO source 5mA，VOH=2.9V

计算芯片功耗：

$$P_D = P_{INT} + P_{IO} = 5mA \times 3.6V + 10 \times 5mA \times 0.3V + 10 \times (3.6V - 2.9V) \times 5mA = 68mW$$

对于 LQFP64 封装的产品，Θ_{JA}=48°C/W，因此可以计算芯片结温：

$$T_J = T_A + P_D \times \Theta_{JA} = 55^oC + 0.068W \times 48^oC/W = 58.264^oC$$

3 电参数

3.1 参数说明

电参数章节中所罗列的典型值是大量样本数据分布的中心值，常温下的最大值是由芯片量产测试所保证的。高低温下的电参数通常基于特征参数提取，由大量样本数据分布的平均值加减 3sigma 得到。

3.2 参数测试条件

3.2.1 供电方案

芯片测试时采用下图所示的电源供电方案。

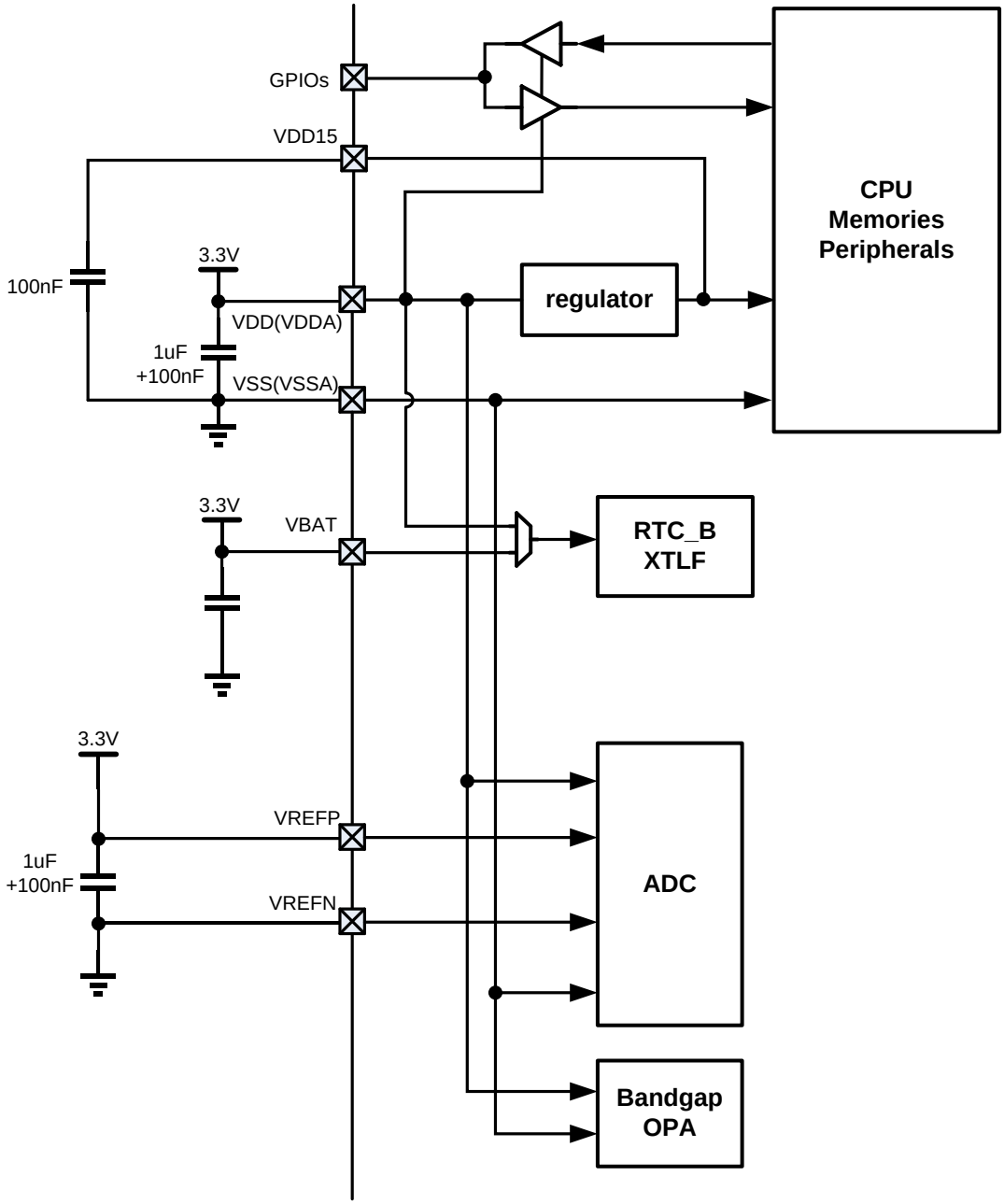


图 3-1FM33LG0xx 测试条件

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	工作温度		-40	85	℃
T_{STG}	存储温度		-55	150	℃
HBM	ESD HBM 模式 TA=25℃ 测试标准符合 JEDEC JS-001	PC5(DAC_OUT)		+/-2000	V
		All others		+/-4000	V
CDM	ESD CDM 模式 TA=25℃ 测试标准符合 JEDEC JS-002			+/-1000	V
LU	IO Latchup -(0.5VDD) < VI < (1.5VDD) TA=25℃ 测试标准符合 JESD78E			+/-210	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 3-1FM33LG0 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	-	0	64	MHz
f_{PCLK}	APB 时钟频率	-	0	64	
VDD	典型工作电压范围	BOR 使能	1.65	5.5	V
		BOR 关闭	1.45	5.5	
VBAT	备份电源工作电压范围		1.5	5.5	V
T_J	结温		-40	125	°C

表 3-2FM33LG0 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=3.3V 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}=16MHz (RCHF) PLL off Flash 0 wait	TA=25℃	-	2.32	-	mA
			TA=85℃	-	2.31	-	
		f_{AHB}=24MHz (RCHF) PLL off Flash 0 wait	TA=25℃	-	3.43	-	mA
			TA=85℃		3.40		
		f_{AHB}=48MHz PLL on Flash 1 wait	TA=25℃	-	6.16	-	mA
			TA=85℃		6.15		
		f_{AHB}=614KHz (RCLF) PLL off Flash 0 wait	TA=25℃		150		uA
			TA=85℃		150		

表 3-3ACTIVE 电流参数

注：上表参数基于特征参数提取，不包含在量产测试中

3.4.2.2 LP 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}	LP Active 模式下的功耗，CPU 从 Flash 取指，Coremark	f_{AHB}=614KHz (RCLF) PLL, RCHF off Flash 0 wait	TA=25℃	-	125	-	uA
			TA=85℃	-	150	-	
IDD_{RUN}	LP Active 模式下的功耗，CPU 从 Flash 取指，while(1)	f_{AHB}=614KHz (RCLF) PLL, RCHF off Flash 0 wait	TA=25℃	-	120	-	uA
			TA=85℃	-	140	-	

表 3-4LP ACTIVE 电流参数

3.4.2.3 LP RUN 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_{LPRUN}	LP RUN 模式下的功耗，CPU 从 Flash 取指，Coremark	f_{AHB}=32768Hz (XTLF) PLL, RCHF, RCLF off Flash 0 wait	TA=25℃	-	29	-	uA
			TA=85℃	-	32	-	
IDD_{LPRUN}	LP RUN 模式下的功耗，CPU 从 Flash 取指，while(1)	f_{AHB}=32768Hz (XTLF) PLL, RCHF, RCLF off Flash 0 wait	TA=25℃	-	28	-	uA
			TA=85℃	-	30	-	

表 3-5LP RUN 电流参数

3.4.2.4 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_{sleep1}	Sleep 模式电流	BOR、SVD 关闭 RTC 使用 XTLF 走时 CPU、RAM、外设数据保持 LCD 显示使能，空载	TA=25℃	-	3	5	uA
			TA=85℃	-	7.5	-	

表 3-6SLEEP 电流参数

3.4.2.5 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_{sleep1}	Sleep 模式电流	BOR、SVD 关闭 RTC 使用 XTLF 走时 CPU、RAM、外设数据保持 LCD 显示关闭	TA=25℃	-	1.2	2	uA
			TA=85℃	-	5.5	-	

表 3-7DEEPSLEEP 电流参数

3.4.2.6 VBAT 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_{VBAT}	VBAT 电流	VDD 掉电 VBAT=3.3V XTLF 驱动电流 250nA RTCB 走时	TA=25℃	-	0.8	-	uA
			TA=85℃	-	1	-	

表 3-8VBAT 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≤85℃		1.45	1.55	1.65	V
V_{BOR}	下电复位电压（预先校准 ULPBG）-40℃≤T_A≤85℃[1]	BORCFG==2'b00		1.65	1.75	1.85	V
		BORCFG==2'b01		1.85	1.95	2.05	
		BORCFG==2'b10		2.05	2.15	2.25	
		BORCFG==2'b11		2.25	2.35	2.45	
V_{PDR}	低功耗下电复位电压（预先校准 ULPBG）-40℃≤T_A≤85℃[1]	PDRCFG==2'b00		1.23	1.33	1.4	V
		PDRCFG==2'b01		1.28	1.38	1.45	
		PDRCFG==2'b10		1.32	1.42	1.48	
		PDRCFG==2'b11		1.4	1.48	1.55	
I_{BOR}	BOR 功耗 VDD=3.3V	BORCFG==2'b00			1		uA
I_{PDR}	PDR 功耗 VDD=3.3V				45		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

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
			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 3-9 复位和电源检测参数

注：

[1] 基于特征参数提取

3.4.4 高精度基准源

芯片内建高精度基准电压源，为 ADC 和 OPA 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 Flash 中，用户应用中可以将这个转换值作为参考基准使用。详细使用方法请参考复旦微电子驱动库函数。

符号	参数说明
REF_CAL	ADC 对 VREF 输出的转换值 测试条件： T_A=30±1℃ VDDA=3V±10mV
REF_RAW	VREF 输出的电压值 测试条件： T_A=30±1℃ VDDA=3V±10mV

高精度基准源主要参数：

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
V_{REF}	基准源输出电压[1]	-40℃≤T_A≤85℃		1.192	1.208	1.223	V
T_{setup}	内部基准源建立时间	典型时间：VDD=3V 最大时间：VDD=1.6V		-	0.5	2	ms
V_{VREF_MEAS}	出厂时测量转换 VREF 的 VDDA 电压	-		2.99	3	3.01	V
T_{coeff}	内部基准源温度系数	-40℃≤T_A≤85℃			25	85	ppm/℃
T_{S_VREF}	ADC 测量 VREF 时的采样时间	预先使能 VREF Buffer		10			us
T_{ADC_BUF}	驱动 ADC 输入的 VREF Buffer 的建立时间[1]	VDD=3V ADC 采样值稳定到 1LSB				100	us
I_{REF}	基准源工作电流[2]	T_A = 25℃	PTAT_EN=0		1.8		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	
		VDD=3V	PTAT_EN=1		2.6		

表 3-10 高精度基准源参数

注：

[1] 基于特征参数提取，不包含在量产测试中

[2] 基于电路仿真

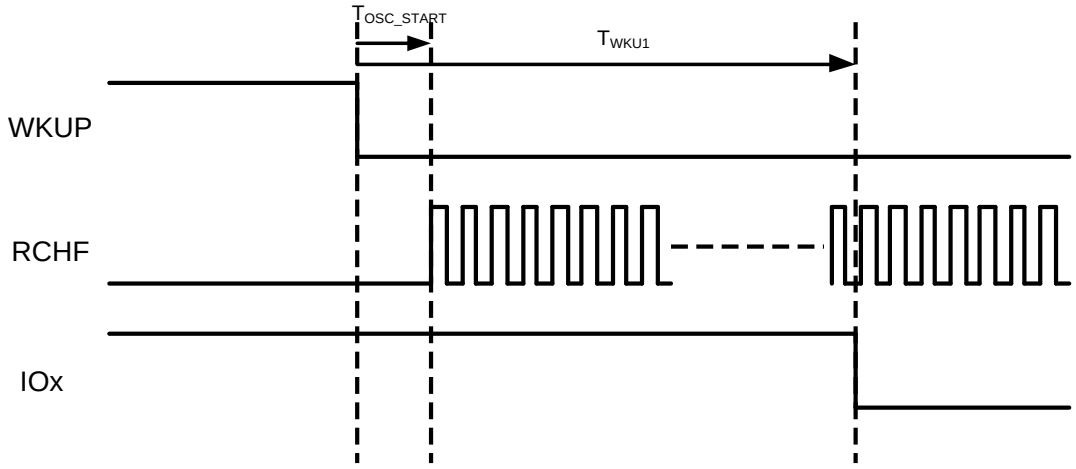
3.4.5 低功耗模式唤醒时间

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	-	6.8	-	us
T_{WKU2}	LPRUN 模式唤醒时间		-	0	-	us

表 3-11 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 3.5us

T_{WKU1} 为唤醒事件到来，到程序运行后翻转 IO 的时间，典型值 6.8us。

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 3-12 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	VDD=3.3V		4	-	24	MHz
R_{fb}	反馈电阻	-		-	200	-	KΩ
gm	跨导增益	VDD=5V	HF_CFG =00000	-	1.75	-	mA/V
			HF_CFG =00001	-	3.5	-	
		VDD=3V	HF_CFG =00000	-	1.2	-	
			HF_CFG =00001	-	2.4	-	
			HF_CFG =00010	-	3.48	-	
		VDD=1.6V	HF_CFG =00000	-	0.32	-	
			HF_CFG =00100	-	1.59	-	
			HF_CFG =00111	-	2.55	-	
VDD_{rise}	XTHF 最低工作电压 HF_CFG=00000	4MHz		1.08	-	-	V
		8MHz,		1.6	-	-	
		24MHz		1.8	-	-	
		32MHz		1.88	-	-	
IDD	XTHF 工作电流 HF_CFG=00000	4MHz		-	170	-	uA
		8MHz		-	200	-	
		32MHz		-	360	-	
	XTHF 工作电流 HF_CFG=11111	4MHz		-	1050	-	uA
		8MHz		-	1100	-	
		32MHz		-	1350	-	
T_{start1}	XTHF 8M 起振时间	VDD=3.3V HF_CFG=00000		-	1.3	-	ms
		VDD=3.3V, HF_CFG=11111		-	0.3	-	ms
T_{start2}	XTHF 16M 起振时间	VDD=3.3V, HF_CFG=00000		-	0.9	-	ms
		VDD=3.3V, HF_CFG=11111		-	0.1	-	ms
C_L	负载电容	-		5	-	25	pF

表 3-13 高频晶体振荡器参数

注：

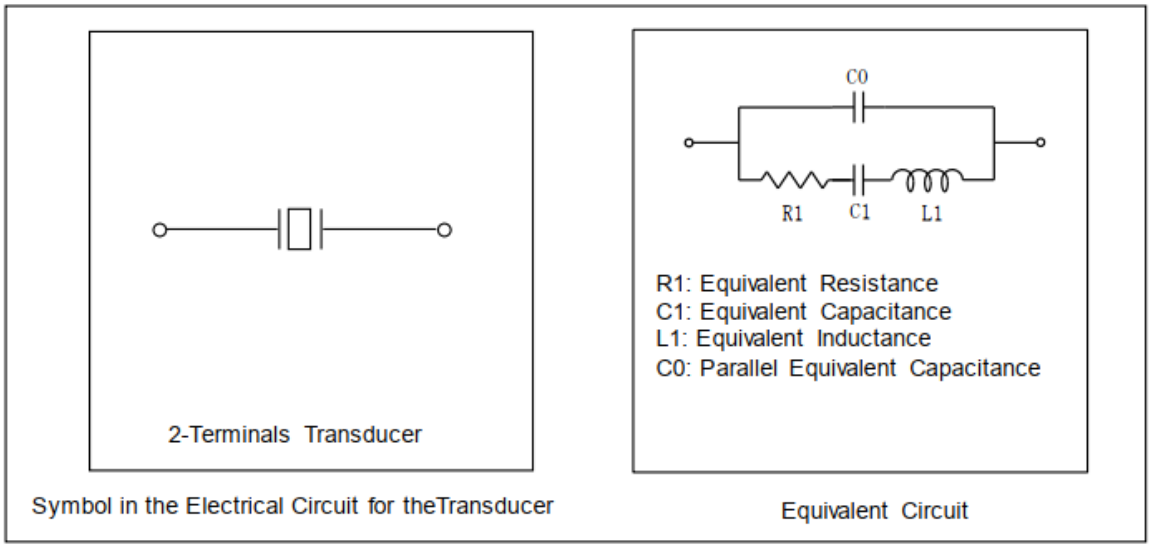
[1]以上指标基于特征参数提取

[2]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

XTHF 典型 gm 参数（仅供设计参考）：

HF_CFG	Gm(mA/V)		
	VDD=5V	VDD=3V	VDD=1.6V
00000	1.75	1.2	0.32
00001	3.5	2.4	0.64
00010	4.91	3.48	0.95
00011	6.67	4.68	1.28
00100	8.07	5.77	1.59
00101	9.83	6.96	1.91
00110	11.2	8.05	2.23
00111	13	9.25	2.55
01000	14.4	10.3	2.86
01001	16.2	11.5	3.18
01010	17.6	12.6	3.5

晶振或陶振的等效电路模型参见下图



振荡器起振的最小环路增益可以通过下式计算：

$$g_{mcrit} = 4 \times ESR \times (2\pi F)^2 \times (C_0 + C_L)^2$$

其中 C_L 是晶振要求的负载电容或陶振内建负载电容，ESR 可以通过下式计算：

$$ESR = R_1 \times \left(1 + \frac{C_0}{C_L}\right)^2$$

为了保证足够安全的振荡余量，通常要求 g_m 大于 5 倍 g_{mcrit}，据此可以根据晶振或陶振手册，选择合适的 XTHF 振荡强度配置。

下表以 muRata 公司的晶振和陶振为例给出计算示例：

型号	类型	频率 (MHz)	ESR(typ) (Ohm)	C0 (pF)	CL (pF)	g_{mcrit} (mA/V)	最小 gm (mA/V)
XRCGB24M000F2P00R0	Crystal	24	49.06	0.526	6pF	0.19	0.95
CSTNE8M00G550000R0	Ceramic	8	19.95	13.15	33	0.4294	2.147
CSTNE12M0G550000R0	Ceramic	12	12.81	14.44	33	0.6557	3.279

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}[1]	RCHF 振荡频率	VDD=1.8~5.5V	FSEL==2'b00	7.96	8	8.04	MHz
			FSEL==2'b01	15.92	16	16.08	
			FSEL==2'b10	23.88	24	24.12	
			FSEL==2'b11	31.84	32	32.16	
ACC_{RCHF}[2]	全温区 RCHF 变化范围	VDD=1.8~5.5V	FSEL==2'b00 T=-40~+85℃	-1	-	1.5	%
			FSEL==2'b01 T=-40~+85℃	-2.5	-	3	%
			FSEL==2'b10 T=-40~+85℃	-3	-	4	%
ACC_{RCHF}[2]	部分温区 RCHF 变化范围	VDD=1.8~5.5V	FSEL==2'b00 T=-10~+70℃	TBD	-	TBD	%
			FSEL==2'b01 T=-10~+70℃	TBD	-	TBD	%
			FSEL==2'b10 T=-10~+70℃	TBD	-	TBD	%

表 3-14 内部高频 RC 振荡器参数

[注1]：此项指标由量产测试保证
[注2]：此项指标基于特征参数提取

内部中频 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_{RCLF}	RCLF 低功耗振荡频率	VDD=1.65~5.5V T=25℃	612	615	618	KHz
I_{DD_RCLF}	RCLF 功耗	VDD=1.65~5.5V T=25℃		1		uA
t_{START}	RCLF 启动时间	VDD=1.65~5.5V T=25℃		120		us

表 3-15 内部中频 RC 振荡器参数

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	VDD=1.65~5.5V	32.4	32.8	33.2	KHz

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
	耗振荡频率	T=25℃				
I_{DD_RCLP}	RCLP 功耗	VDD=1.65~5.5V T=25℃		350		nA
t_{START}	RCLP 启动时间	VDD=1.65~5.5V T=25℃		380		us

表 3-16 内部低频 RC 振荡器参数

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	VDD=1.65~5.5V T=-40~85℃	32	-	64	MHz
I_{DD_PLL}	PLL 功耗	输入频率 1Mhz，输出频率 32Mhz	-	450	-	uA
		输入频率 1Mhz，输出频率 64Mhz	-	450	-	
t_{LOCK}	PLL 锁定时间		-	65	-	us

表 3-17 PLL 参数

3.4.9 ADC 特性

3.4.9.1 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	-		1.65		5.5	V
VREF+	正参考电压	-		1.5		VDDA	V
VREF-	负参考电压	-		0		0.5	V
T_J	工作结温范围	-		-40		125	°C
V_{AIN}	输入电压范围	-		VREF-		VREF+	V
Vcm	差分信号共模输入范围	-		0.2		VREF-0.2	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=2.0~5.5V				2	Msps
		VDDA=1.6~2.0V				1.5	
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	-	250	-	uA
			Fs=2Msps	-		-	
			Fs=250Ksps	-		-	
		VDDA=5V	Fs=1Msps	-	380	-	uA
			Fs=2Msps	-		-	
			Fs=250Ksps	-		-	
ADC 动态性能[1]							
ENOB	有效位数与输入信号频率的关系 VDDA=3.3V VREF+=VDDA F_S=2Msps T_A=25℃ ADCCLK=XTHF	单端模式 F_{AIN}=29KHz		-	10.8	-	bits
		差分模式 F_{AIN}=29KHz		-	11.3	-	bits
		单端模式 F_{AIN}=499KHz		-	10.6	-	bits
		差分模式 F_{AIN}=499KHz		-	11.3	-	bits
	有效位数与工作电压的关系 VREF+=VDDA F_{AIN}=29KHz T_A=25℃ ADCCLK=XTHF	VDDA=5.0V F_S=1Msps	差分模式	-	11.4	-	bits
			单端模式	-	10.9	-	
		VDDA=3.3V F_S=1Msps	差分模式	-	11.4	-	bits
			单端模式	-	10.8	-	
		VDDA=1.8V F_S=1Msps	差分模式	-	11.1	-	bits
			单端模式	-	10.3	-	
SNDR	信噪失真比 VDDA=3.3V	单端模式 F_{AIN}=29KHz		-	65	-	dB

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
	VREF+=VDDA F_S=1Msps -40℃≤T_A≤85℃	单端模式 F_{AIN}=499KHz	-	63	-	dB
SFDR	无杂散动态范围 VDDA=3.3V VREF+=VDDA F_S=1Msps F_{AIN}=29KHz -40℃≤T_A≤85℃	单端模式	-	78	-	dB
ADC 静态性能						
DNL	差分非线性 VDDA=3.3V Fs=1Msps	单端模式	-1	-	1	LSB
		差分模式	-1	-	1	
INL	积分非线性 VDDA=3.3V Fs=1Msps	单端模式	-1	-	2	LSB
		差分模式	-1	-	1	
OffsetError	失调误差 校准后	单端模式		-0.5		LSB
GainError	增益误差 校准后	单端模式		-0.1		%

表 3-18ADC 参数

注：

[1]基于特征参数提取

ADC 典型静态性能参数图表

- **差分输入模式**

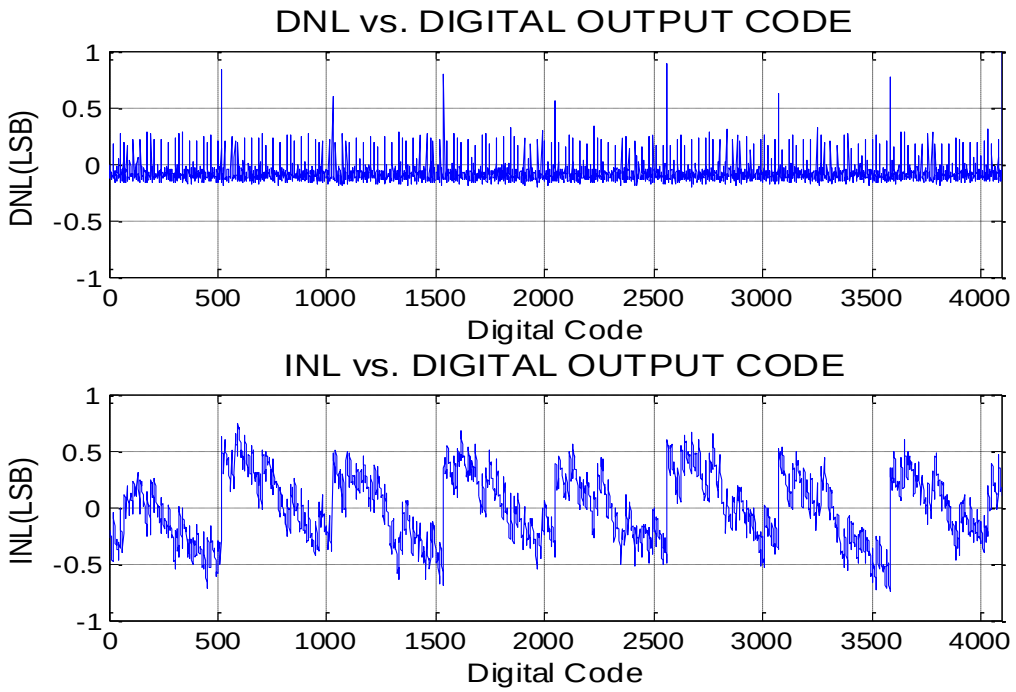


图 3-2 ADC 差分输入典型 DNL 和 INL

- 单端输入模式

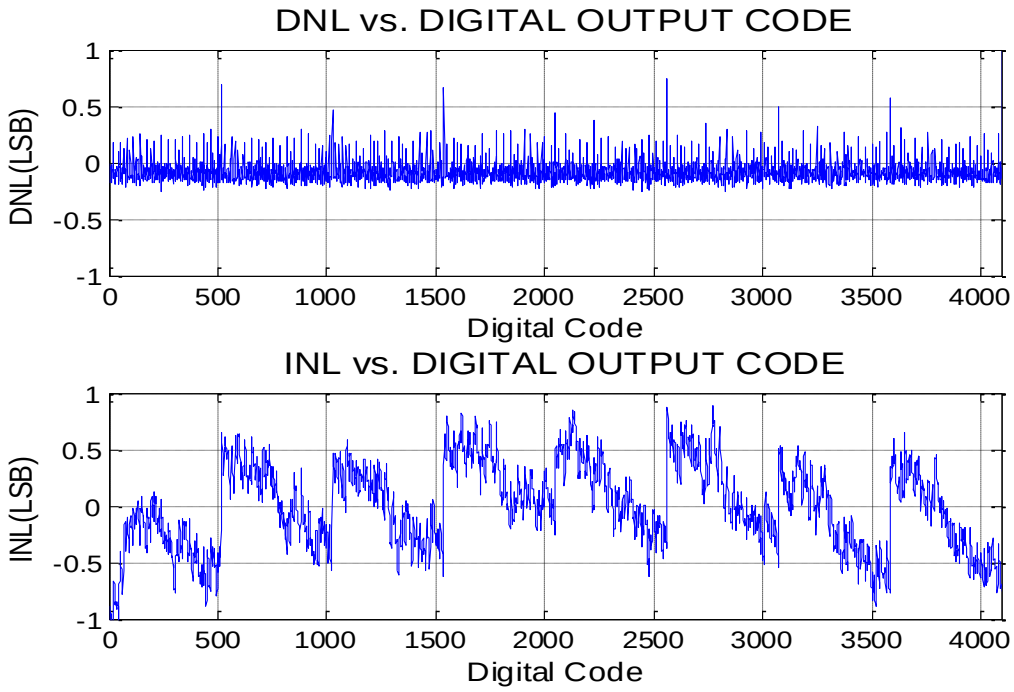


图 3-3 ADC 单端输入典型 DNL 和 INL

ADC 典型信噪比频谱图

- **VDD=VREFP=3.3V, F_S=2Msps,ADCCLK=XTHF32M, FAIN=29KHz**

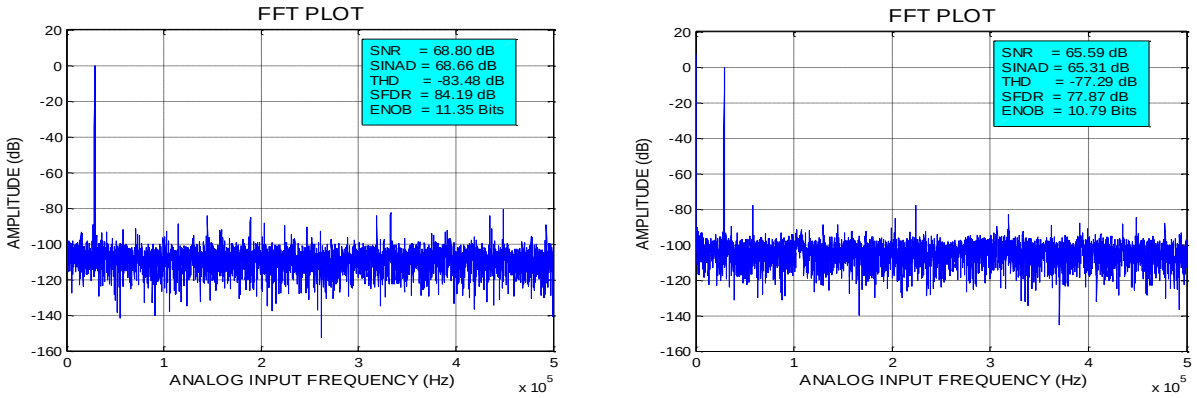


图 3-4 3.3V ADC 差分（左）和单端输入（右）典型信噪比

- **VDD=VREFP=1.8V, F_S=2Msps,ADCCLK=XTHF32M, FAIN=29KHz**

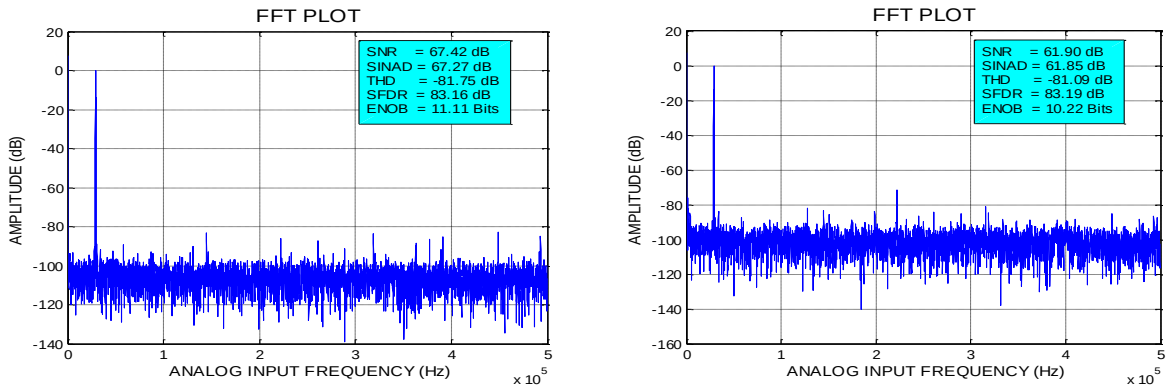


图 3-5 1.8V ADC 差分（左）和单端输入（右）典型信噪比

3.4.9.2 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

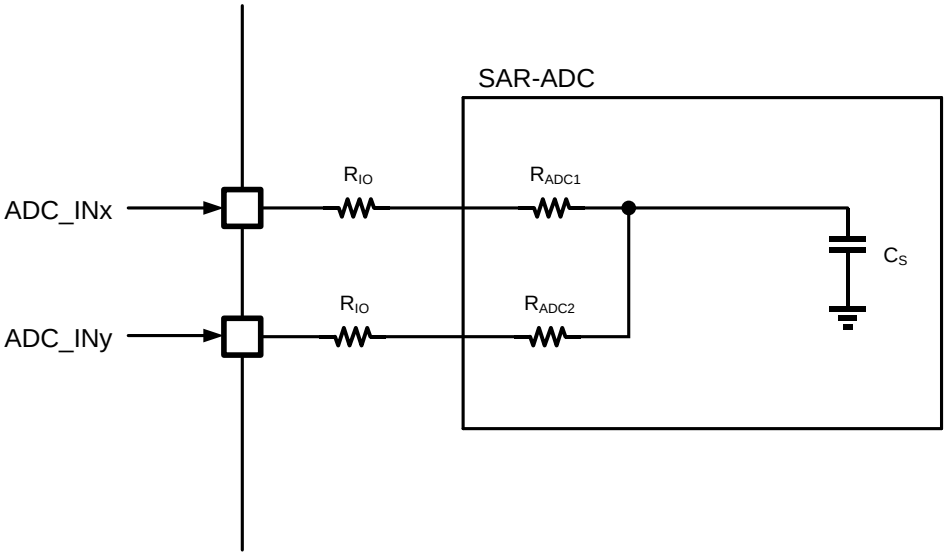


图 3-6 ADC 通道输入阻抗

3.4.9.3 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	
快速通道，差分输入						
R_{ADC}	5V	25C	-	182	-	Ω
		85C	-	240	303	
		-40C	-	126	159	
	3.3V	25C	-	280	-	
		85C	-	360	467	
		-40C	-	200	259	
	1.6V	25C	-	972	-	
		85C	-	1100	1612	
		-40C	-	819	1285	
快速通道，单端输入						
R_{ADC}	5V	25C	-	2222	-	Ω
		85C	-	2391	2724	
		-40C	-	2172	2471	
	3.3V	25C	-	2320	-	
		85C	-	2513	2887	
		-40C	-	2244	2569	
	1.6V	25C	-	2978	-	
		85C	-	3230	3973	
		-40C	-	2823	3511	
慢速通道，单端输入，输入信号电平 VDDA/2						
R_{ADC}	5V	25C	-	1285	-	Ω
		85C	-	-	1704	
		-40C	-	-	1437	
	3.3V	25C	-	1568	-	
		85C	-	-	2169	
		-40C	-	-	1795	
	1.6V	25C	-	5123	-	
		85C	-	-	8088	
		-40C	-	-	16970	

表 3-19ADC 输入阻抗

采样时间计算示例：

1）假设采用快速通道单端采样一个信号源，信号源内阻为 1KΩ，工作电源 VDDA=3.3V，工作温度 25C，则根据 Tsamp 公式，建议的最小采样时间为 107ns。如果 ADC 工作时钟是 16Mhz，相当于采样时间配置应大于 2 个 ADC 时钟。

2）假设采用慢速通道采样一个信号源，信号源内阻为 100KΩ，工作电源 VDDA=1.6V，工作温度 -40C~85C，则根据 Tsamp 公式，建议的最小采样时间为 3.4us。如果 ADC 工作时钟是 8Mhz，相当于采样时间配置应大于 28 个 ADC 时钟。

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	-	1.6	-	5.5	V
VREF	参考电压范围	-	1.6	-	VDDA	V
IDD_{VREF}	VREF 功耗[1] VDDA=VREF=3.3V Buffer ON，无负载	Middle code (0x800)		120		uA
		Worst code (0x000)		200		
	VREF 功耗[1] VDDA=VREF=3.3V Buffer OFF，无负载	Middle code (0x800)		100		uA
		Worst code (0x000)		50		
IDD_{VDDA}	VDDA 功耗[1] VDDA=VREF=3.3V Buffer ON，无负载	Middle code (0x800)		300		uA
		Worst code (0xF5C)		360		
	VDDA 功耗[1] VDDA=VREF=3.3V Buffer OFF，无负载	Middle code (0x800)		0		uA
		Worst code (0xF5C)		0		
V_{DAC}	输出电压范围[1]	Buffer ON	0.15	-	VREF-0.15	V
		Buffer OFF	0	-	VREF	
R_O	DAC 输出内阻	Buffer OFF[1]		16.8		KΩ
		Buffer ON[1]		5		Ω
R_L	输出阻性负载[1]	Buffer ON R_L connected to VSSA	5			KΩ
C_L	输出容性负载[1]	Buffer ON			50	pF
DNL	差分非线性[1]	Buffer ON	-1		+1	LSB
		Buffer OFF	-1		+1	LSB
INL	积分非线性[1]	Buffer ON VDDA=VREF=3.3V	-2		+2	LSB
		Buffer OFF VDDA=VREF=3.3V	-1		+2	LSB
OffsetError	失调误差[1] VDDA=VREF=3.3V	Buffer OFF Code=0x000		0.1		mV
GainError	增益误差[1] VDDA=VREF=3.3V	Buffer OFF Code=0xFFF		6		mV
t_{setup}	满幅输出建立时间	Code=0xFFF Buffer OFF		3		us
		Code=0xFFF		1		

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
		Buffer ON				

表 3-20 DAC 参数

[1]基于特征参数提取

DAC 典型输出曲线（VDD=3.3V, BUFFER ON, R_L=5KΩ）

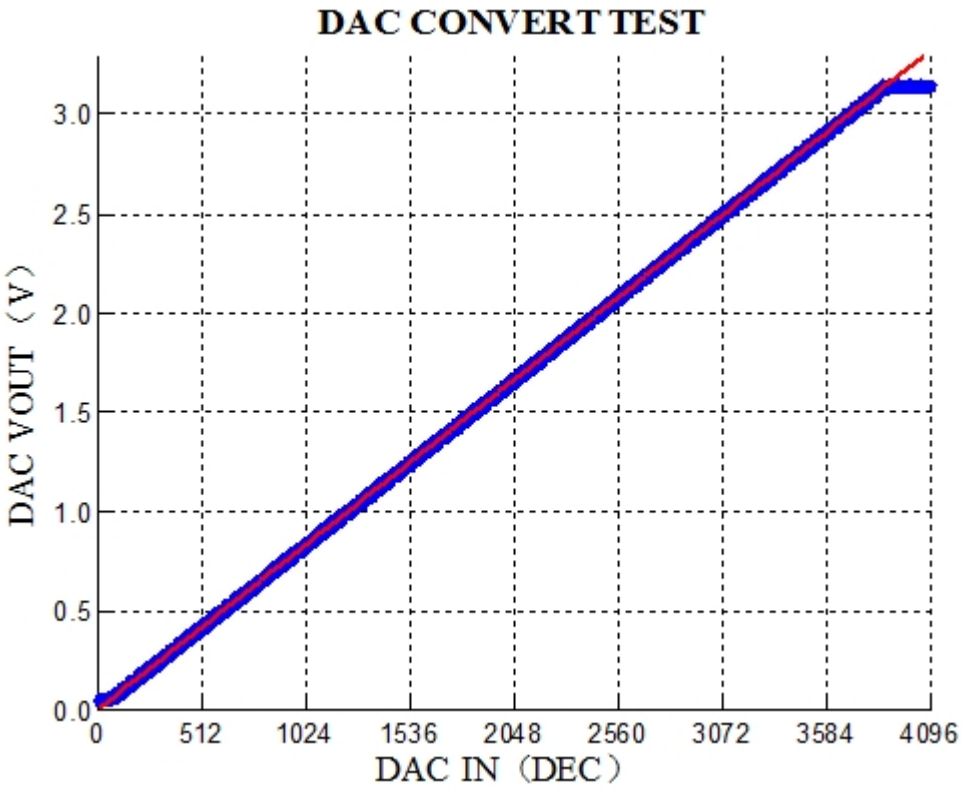


图 3-7 DAC 典型输出曲线

DAC 典型 DNL 和 INL 曲线

- **VDD=5V, BUFFER OFF,** 无负载

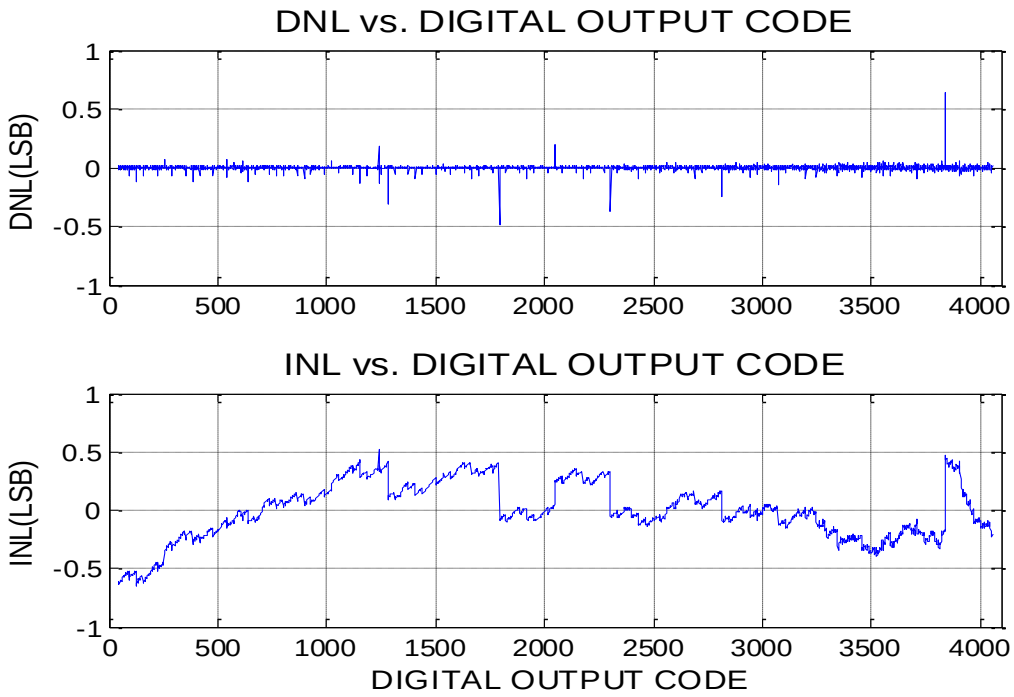


图 3-8 DAC 典型静态特性（5V）

- **VDD=3.3V, BUFFER OFF,** 无负载

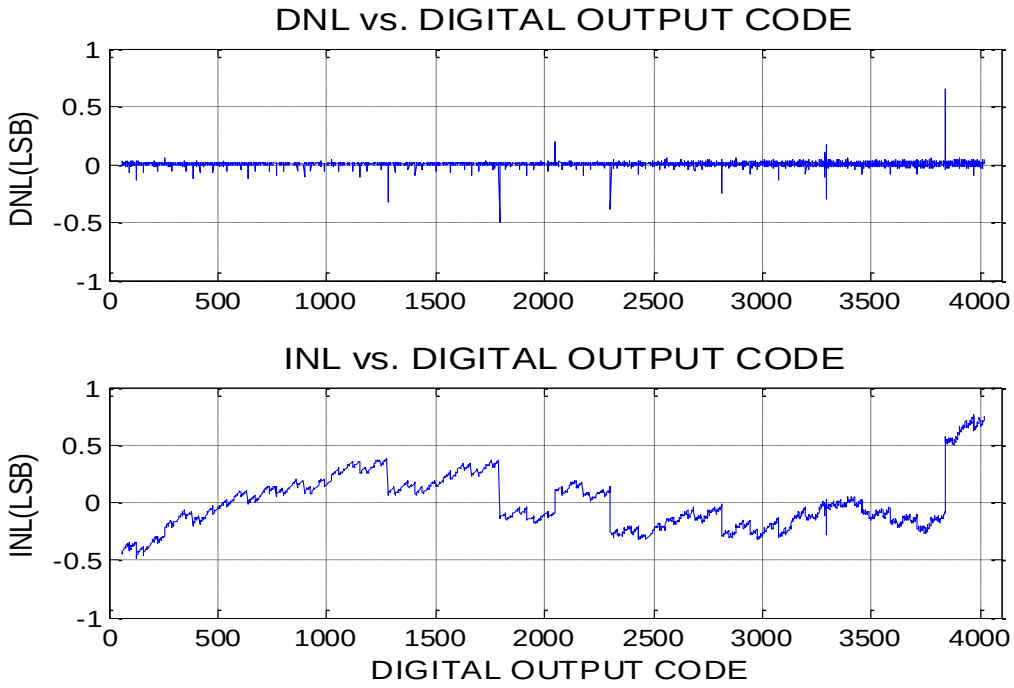


图 3-9 DAC 典型静态特性（3.3V）

- **VDD=1.8V, BUFFER OFF,** 无负载

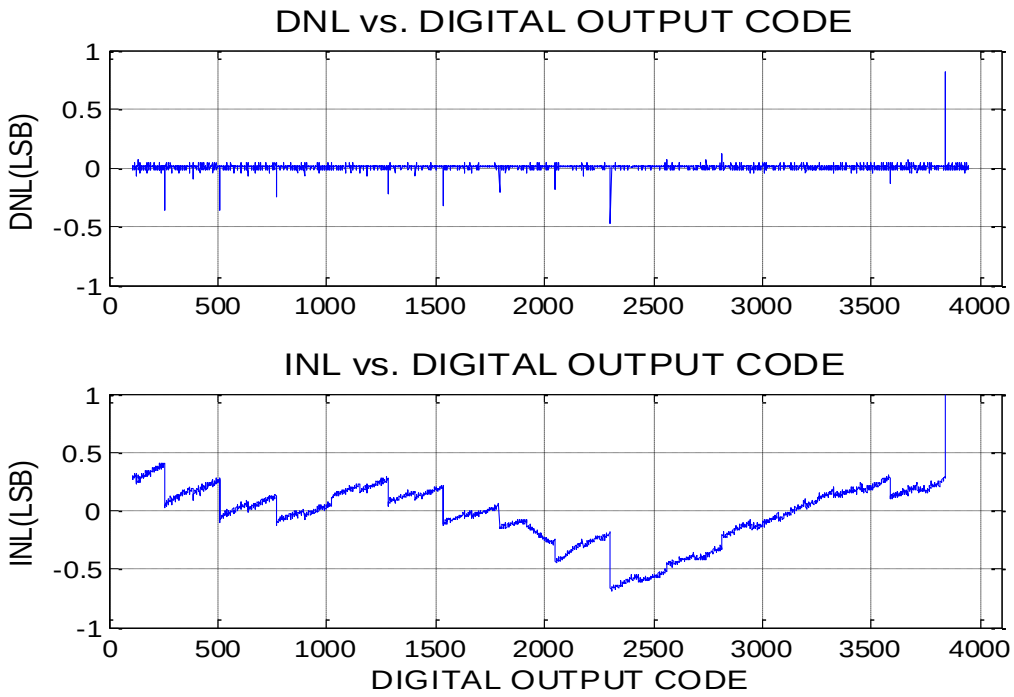


图 3-10 DAC 典型静态特性（1.8V）

3.4.11 温度传感器

芯片出厂时经过温度定标，定标条件是 VDD=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。详细使用方法请参考复旦微电子提供的库函数。

符号	参数说明	测试条件
TS_CAL1	温度传感器标定值 1	VDD=3.0V，T_A=30+/-1°C

注：根据 TS_CAL1 的数值，可以计算温度定标时温度传感器在 30° C 下的输出电压绝对值。

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		2.04		LSB/°C
		VDDA=VREF+=3V		3.14		
Slope	输出斜率[1]	T_A=-40~+85° C VDDA=1.8~5.5V		2.53		mV/°C
Linerity	全温区线性度[1]		-	+/-1	+/-2	°C
I_{DDA}	温度传感器功耗（不含 ADC）[2]	VDDA=3.3V		0.8		uA
t_{START}	温度传感器启动时间，包含输出 buffer 建立时间[2]	VREF1p2 已经使能，置位 PTAT_EN 寄存器、VPTATBUFFER_OUTEN、VPTATBUFFER_EN			50	us
		VREF1p2 未使能			1.4	ms

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_{SAMPLE}	ADC 采样温度传感器输出时要求的采样时间[2]		10	-	-	us

表 3-21 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。

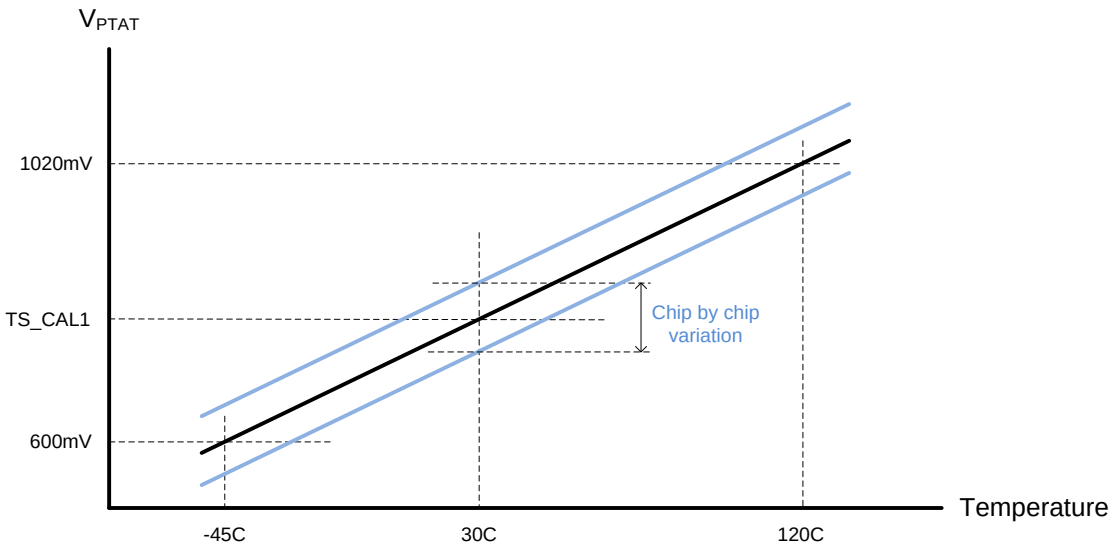


图 3-11 温度传感器输出曲线

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 运算放大器特性

除非特别声明，以下参数基于特征参数提取。

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			1.8		5.5	V
Vos	输入 offset 电压	校准前 VDDA=1.8~5.5V VCM=VDDA/2 T_A=25C			±1.5		mV
Ibias	输入偏置电流	VDDA=3.3V VCM=VDDA/2 T_A=25C				±100	pA
Ios	输入失调电流					±100	pA
TRIMSTEP_P	低共模输入电压 offset trim 步长（0.1xVDDA）	VDDA=3.3V VCM=VDDA/2 T_A=25C			0.04		mV
TRIMSTEP_N	高共模输入电压 offset trim 步长（0.9xVDDA）				0.05		
VOHsat	输出高饱和电压与 VDDA 的差值	VCM=VDDA/2 R_L=4KOhm	VDDA=2.0V	-	200	-	mV
			VDDA=3.3V	-	300	-	
			VDDA=5.5V	-	400	-	
		VCM=VDDA/2 R_L=50KOhm	VDDA=2.0V	-	40	-	mV
			VDDA=3.3V	-	60	-	
			VDDA=5.5V	-	100	-	
VOLsat	输出低饱和电压与 VSSA 的差值	VCM=VDDA/2 R_L=4KΩ	VDDA=2.0V	-	160	-	mV
			VDDA=3.3V	-	190	-	
			VDDA=5.5V	-	300	-	
		VCM=VDDA/2 R_L=50KΩ	VDDA=2.0V	-	20	-	mV
			VDDA=3.3V	-	20	-	
			VDDA=5.5V	-	30	-	
I_{LOAD}	输出饱和驱动电流	R_L=4KΩ	VDDA=2.0V	-	400	-	uA
			VDDA=3.3V	-	680	-	
			VDDA=5.5V	-	1150	-	
		R_L=50KΩ	VDDA=2.0V	-	40	-	uA
			VDDA=3.3V	-	65	-	
			VDDA=5.5V	-	110	-	
C_{LOAD}	容性负载					50	pF

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
CMIR	共模输入范围	VDDA=3.3V Voltage follower or PGA configuration		0.01		VDDA-0.01	V
CMRR	共模抑制比	VDDA=3.3V VCM=0.3~3V			66		dB
		VDDA=5V VCM=0.5~4.8V			70		dB
PSRR	电源抑制比	VDDA=1.8~5.5V VCM=VDDA/2		80			dB
GBW	-3dB 带宽	VDDA=3.3V	VCM=1.0~2.5V		3500		KHz
			VCM=0.2V		2200		
			VCM=0.015V		1200		
			VCM=3V		1800		
SR	上升压摆率 （输出电压变化范围是 10%到 90%）	VDDA=2.0V			1.1		V/us
		VDDA=3.3V			1.25		
		VDDA=5V			1.45		
	下降压摆率 （输出电压变化范围是 90%到 10%）	VDDA=2.0V			1.3		V/us
		VDDA=3.3V			1.4		
		VDDA=5V			1.6		
AO	开环增益	VDDA=3.3V			100		dB
Phi	相位裕度[1]	C_{LOAD}=50pF VDDA=5V			80		°
GM	增益裕度[1]	C_{LOAD}=50pF VDDA=5V			11		dB
t_{START}	启动时间	Buffer 模式 VDDA=3.3V			2.2		us
PGA gain	PGA 增益	VDDA=3.3V 同相放大	Gain=2		2		-
			Gain=4		3.99		
			Gain=8		7.96		
			Gain=16		15.85		
		VDDA=3.3V 反相放大	Gain=1		0.99		-
			Gain=3		2.95		
			Gain=7		6.79		
			Gain=15		14.1		
PGA BW	PGA 带宽	Gain=2			GBW/2		
		Gain=4			GBW/4		
		Gain=8			GBW/8		
		Gain=16			GBW/16		
I_{DDA}	功耗	Standalone，正常模式，空载 VDDA=3.3V VCM=VDDA/2			150		uA
		Buffer，低功耗模式，空载 VDDA=3.3V VCM=VDDA/2			1.5		

表 3-22 OPA 参数

[1] 基于电路设计仿真

3.4.13 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	-		1.8	-	5.5	V
VI_{comp1}	比较器1输入电压范围	-		0	-	VDDA	V
VI_{comp2}	比较器2输入电压范围	-		0	-	VDDA	V
VI_{comp3}	比较器3输入电压范围	-		0	-	VDDA-0.7	V
I_{comp12}	比较器 1/2 工作电流	VDDA=3.3V 50Khz 方波，±100mV overdrive	低功耗模式	-	9	-	uA
			中速模式	-	9.2	-	
			高速模式	-	32	-	
		VDDA=3.3V DC 输入	低功耗模式	-	1	-	
			中速模式	-	1.5	-	
			高速模式	-	25	-	
$T_{propagation12}$	比较器 1/2 传播延迟	VDDA=3.3V 200mV step 100mV overdrive	低功耗模式	-	0.8	-	us
			中速模式	-	0.6	-	
			高速模式	-	150	-	ns
$T_{setup12}$	比较器 1/2 建立时间	VDDA=3.3V	低功耗模式	-	20	-	us
			中速模式	-	15	-	
			高速模式	-	5	-	
I_{comp3}	比较器3工作电流	VDDA=3.3V VDDA=3.3V 50Khz 方波，±100mV overdrive		-	8	-	uA
		VDDA=3.3V DC 输入		-	150	-	nA
$T_{propagation3}$	比较器3传播延迟	VDDA=3.3V		-	1.5	-	us
T_{setup3}	比较器3建立时间	VDDA=3.3V		-	20	-	us

表 3-23 模拟比较器参数

3.4.14 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	
	Flash size		64K	-	256K	bytes
T_{PROG}	Byte Program Time		6	-	7.5	μs
T_{ERASE}	Sector/Block Erase		4	-	5	ms
	Chip Erase		30	-	40	ms
N_{ED}	Sector Endurance		100,000			Erase/Write cycles
T_{DR}	Data Retention	T=85℃ After 20K cycling	10			yrs

表 3-24 Flash 参数

3.4.15 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	PA11,PA12		1.1		V
		V_{DD}=5V I_{SINK}=5mA	PC12		1		
			PH15		0.3		
		V_{DD}=5V I_{SINK}=10mA	其他		1.05		
V_{OH}	输出高电平	V_{DD}=5V I_{SOURCE}=20mA	PA11,PA12		3.75		V
		V_{DD}=5V I_{SOURCE} =5mA	PC12		3.95		
			PH15		3.9		
		V_{DD}=5V I_{SOURCE} =10mA	其他		3.8		
V_{OL}	输出低电平	V_{DD}=3.3V I_{SINK}=10mA	PA11,PA12		0.6		V
		V_{DD}=3.3V I_{SINK}=5mA	PC12		1.1		
			PH15		0.4		
		V_{DD}=3.3V I_{SINK}=5mA	其他		0.6		
V_{OH}	输出高电平	V_{DD}=3.3V I_{SOURCE}=10mA	PA11,PA12		2.5		V
		V_{DD}=3.3V I_{SOURCE} =1.5mA	PC12		2.9		
			PH15		2.8		
		V_{DD}=3.3V I_{SOURCE} =5mA	其他		2.6		
R_{PU}	弱上拉电阻				100		KΩ

表 3-25 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
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			5		KΩ
$T_{AFILTER}$	模拟滤波长度[1]	VDD=5V		100		ns
		VDD=3V				ns
$T_{DFILTER}$	数字滤波长度[1]	VDD=1.65~5.5V -40℃≤T_A≤85℃	50		100	us

表 3-26 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 3-27 引脚 AC 参数

注：

[1] 依据电路仿真，不在量产测试中测试

3.4.16 LCD 特性

片内电阻分压模式

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_{LCD}	片内电阻分压模式下的 LCD 工作电流（空载）[1]	VDD=5V		2		uA
		VDD=3V				
V_{LCD}	LCD 偏置电压	-	0.547×VDD		VDD	V

表 3-28 LCD 片内电阻分压

[1]基于特征参数提取

片外电容模式

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_{LCD}	片内电阻分压模式下的 LCD 工作电流（空载）[1]	VDD=5V				uA
		VDD=3V				
V_{LCD}	LCD 偏置电压	-	0.547×VDD		VDD	V
C1	VCIN1 和 VCIN2 之间的去耦电容	-		0.1		uF
C20	VDISP0 对地去耦电容	-		0.1		uF
C21	VDISP1 对地去耦电容	-		0.1		uF
C22	VDISP2 对地去耦电容	-		0.1		uF
C23	VDISP3 对地去耦电容	-		0.1		uF

表 3-29 LCD 片外电容驱动

[1]基于特征参数提取

LCD 电容驱动模式的片外连接如下图所示：

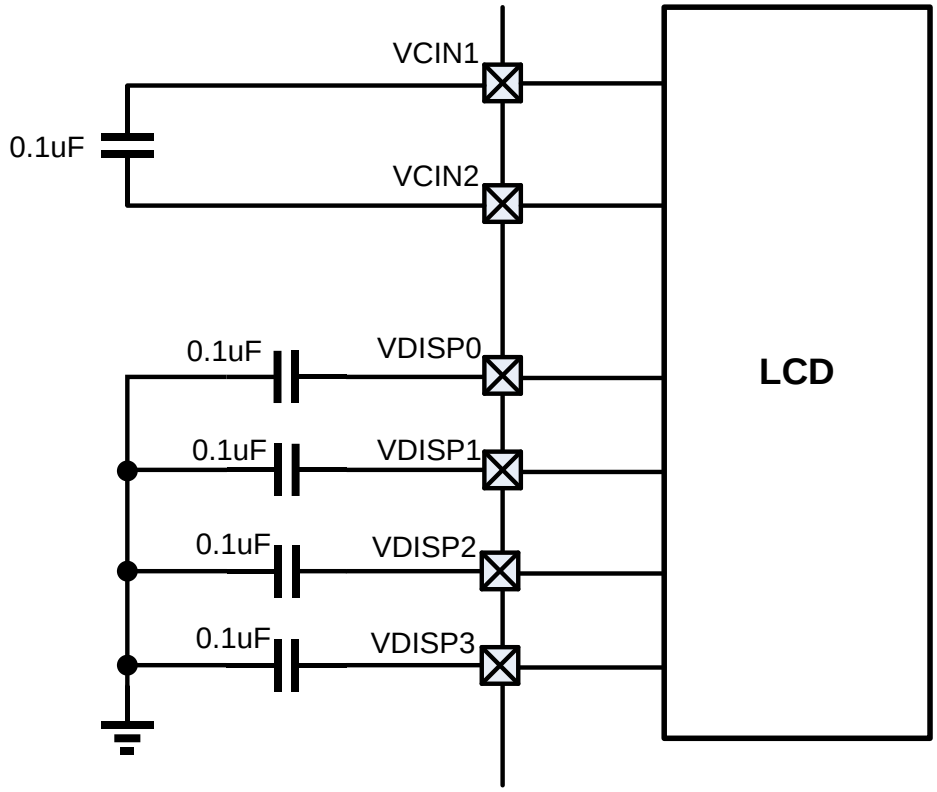


图 3-12LCD 片外电容驱动模式电容连接

3.4.17 VBAT 测量特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
R	VBAT 测量电路分压电阻			100		KΩ
Q	VBAT 分压比例	VBAT_MEASURE=VBAT/Q		3		
Er	分压比例误差[1]		-1	-	1	%
t_{S_VBAT}	测量 VBAT 时的 ADC 采样时间[1]	ADC 内部 Buffer 使能				μs
		ADC 内部 Buffer 不使能				

表 3-30VBAT 测量特性

[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

FM33LG0 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

FM33LG0 的系统总线示意图如下，包含一条 AHB-Lite 总线、一条 APB 总线。

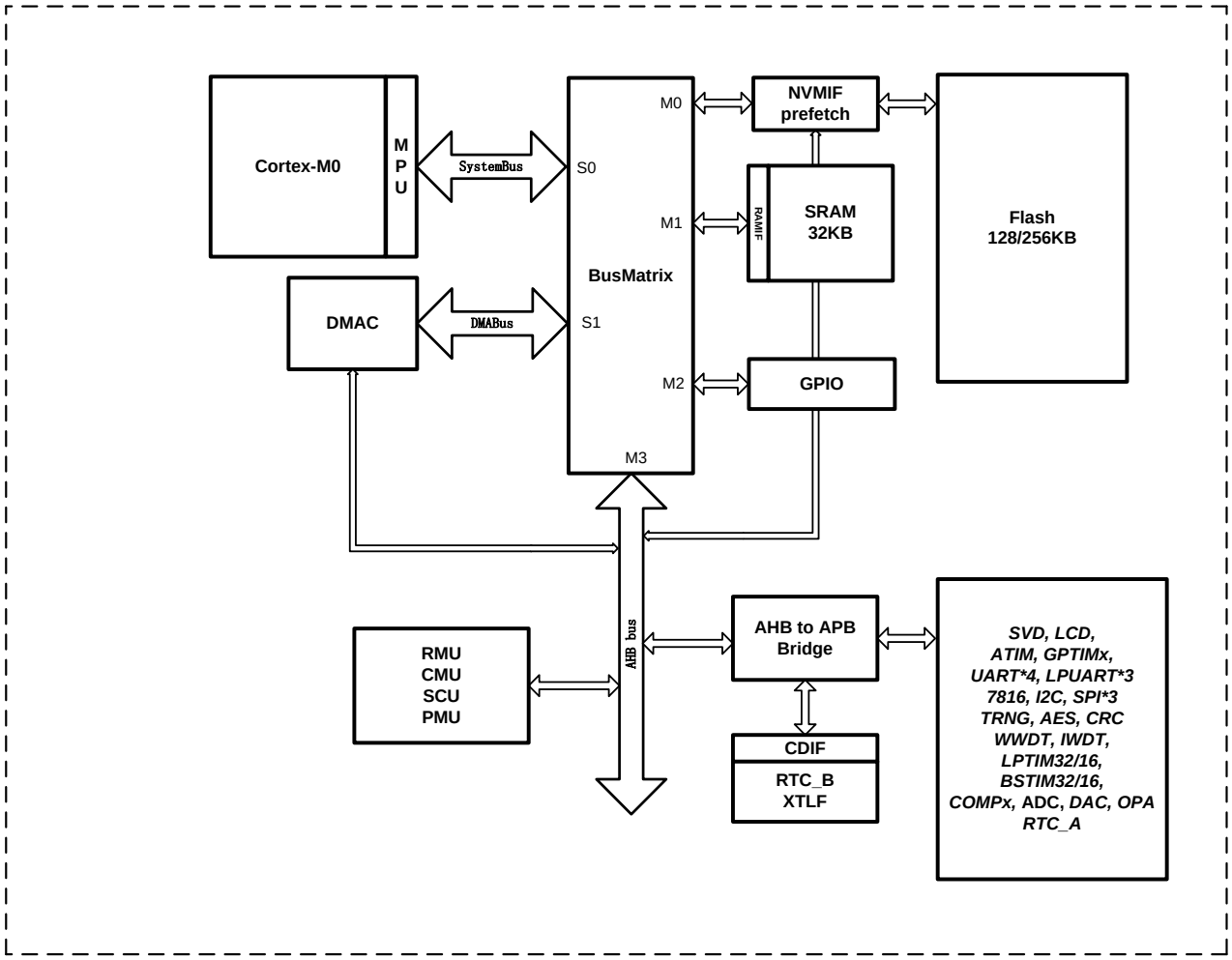


图 4-1 系统总线示意图

4.2 存储空间分配

4.2.1 概述

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

Flash 包含 4 个 information 页，2 个 LDT 页，1 个冗余页，1 个 DCT 页。其中 DCT 和 LDT 为芯片原厂保留页，不对用户开放。Information 为用户配置页，用于保存用户配置信息。所有 option 页在地址上与 Flash 主区域互相隔离。

当芯片从 Flash 启动时，FM33LG0 的地址空间分配如下图（256KB Flash，32KB RAM）：

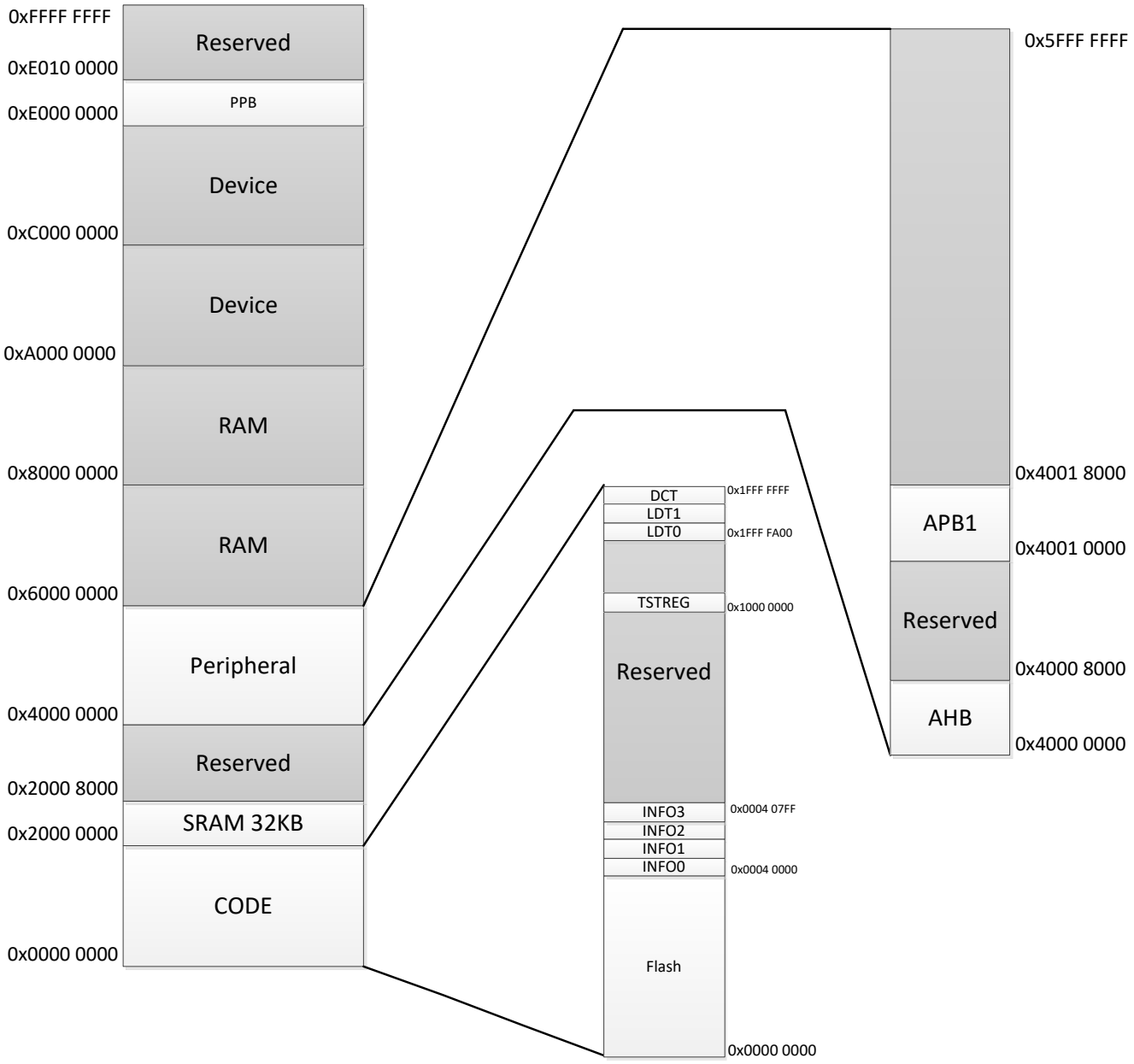


图 4-2 FM33LG04x 总线地址

128KB flash+32KB RAM：

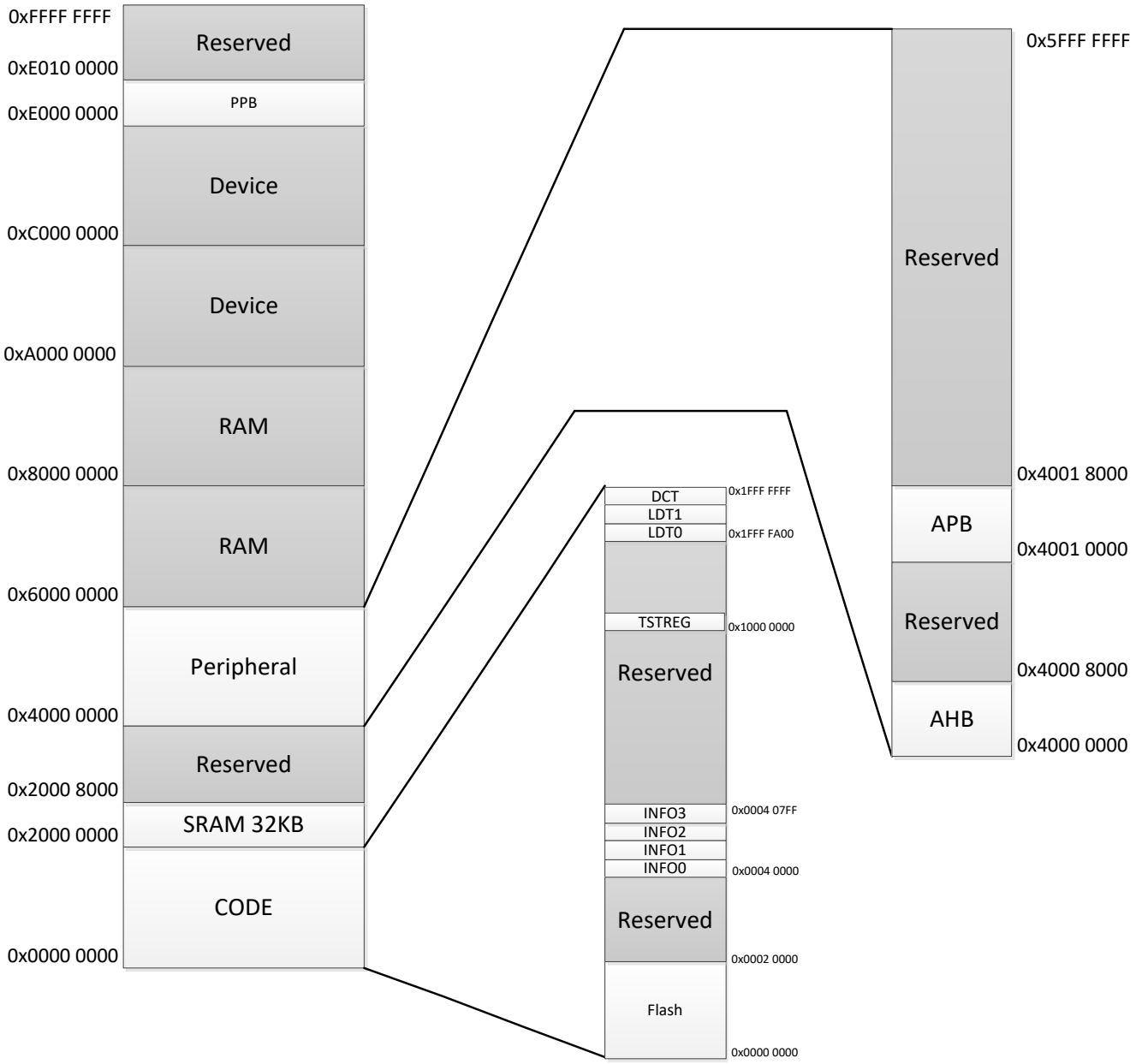


图 4-3 FM33LG02x 总线地址

64KB flash+16KB RAM：

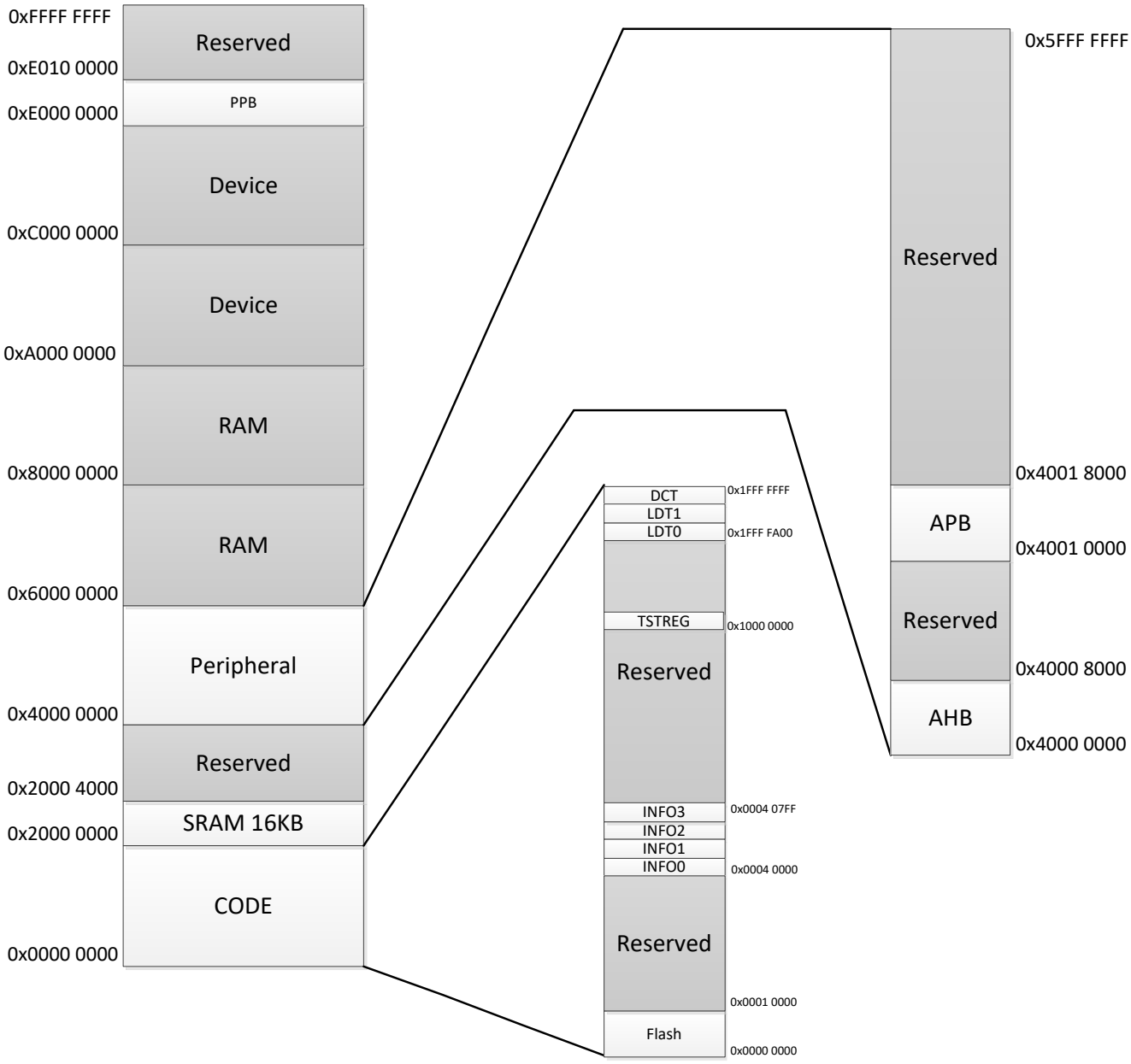


图 4-4 FM33LG01x 总线地址

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	地址边界	空间	外设
AHB	0x0000_0000~0x0007_FFFF	512KB	Flash main array
	0x1FFF_F000~0x1FFF_FFFF	4KB	Flash Option cell array
	0x2000_0000~0x2000_7FFF	32KB	SRAM
	0x2000_8000~0x2001_3FFF	-	-
	0x4000_0000~0x4000_03FF	1KB	SCU
	0x4000_0400~0x4000_07FF	1KB	DMA
	0x4000_0800~0x4000_0CFF	-	-
	0x4000_0C00~0x4000_0FFF	1KB	GPIO
	0x4000_1000~0x4000_13FF	1KB	NVMIF Registers

	0x4000_1400~0x4000_17FF	-	-
	0x4000_1800~0x4000_1BFF	-	-
	0x4000_1C00~0x4000_1FFF	-	-
	0x4000_2000~0x4000_23FF	1KB	PMU
	0x4000_2400~0x4000_27FF	1KB	CMU
	0x4000_2800~0x4000_2BFF	1KB	RMU
	0x4001_0000~0x4001_FFFF	64KB	APB
APB	0x4001_0000~0x4001_03FF	1KB	CRC
	0x4001_0400~0x4001_07FF	1KB	SPI0
	0x4001_0800~0x4001_0BFF	1KB	SPI1
	0x4001_0C00~0x4001_0FFF	1KB	LCD
	0x4001_1000~0x4001_13FF	1KB	RTC_A
	0x4001_1400~0x4001_17FF	1KB	IWDT
	0x4001_1800~0x4001_1BFF	1KB	WWDT
	0x4001_1C00~0x4001_1FFF	1KB	U7816
	0x4001_2000~0x4001_23FF	1KB	UART0
	0x4001_2400~0x4001_27FF	1KB	I2C
	0x4001_2800~0x4001_2BFF	1KB	SVD
	0x4001_2C00~0x4001_2FFF	1KB	RAMBIST
	0x4001_3000~0x4001_33FF	1KB	ATIM
	0x4001_3400~0x4001_37FF	1KB	LPTIM32
	0x4001_3800~0x4001_3BFF	1KB	AES
	0x4001_3C00~0x4001_3FFF	1KB	TRNG
	0x4001_4000~0x4001_43FF	1KB	LPUART0
	0x4001_4400~0x4001_47FF	1KB	LPUART1
	0x4001_4800~0x4001_4BFF	1KB	SPI2
	0x4001_4C00~0x4001_4FFF	1KB	GPTIM0
	0x4001_5000~0x4001_53FF	1KB	LPUART2
	0x4001_5400~0x4001_57FF	1KB	COMPx
	0x4001_5800~0x4001_5BFF	1KB	AutoTrim
	0x4001_5C00~0x4001_5FFF	1KB	ADC
	0x4001_6000~0x4001_63FF	1KB	BSTIM32
	0x4001_6400~0x4001_67FF	1KB	GPTIM1
	0x4001_6800~0x4001_6BFF	1KB	UART1
	0x4001_6C00~0x4001_6FFF	1KB	PGL
	0x4001_7000~0x4001_73FF	1KB	UART3
	0x4001_7400~0x4001_77FF	1KB	UART4
	0x4001_7800~0x4001_7BFF	1KB	UART5
	0x4001_7C00~0x4001_7FFF	1KB	UARTIR
	0x4001_8000~0x4001_83FF	1KB	GPTIM2
	0x4001_8400~0x4001_87FF	1KB	-
	0x4001_8800~0x4001_8BFF	1KB	LPTIM16
	0x4001_9000~0x4001_93FF	1KB	ANTEST(BUF4TST)
	0x4001_9800~0x4001_9BFF	1KB	DAC
	0x4001_9C00~0x4001_9FFF	1KB	DIVAS
	0x4001_A000~0x4001_A3FF	1KB	OPA
	0x4001_A400~0x4001_A7FF	1KB	VREF1p2
	0x4001_E000~0x4001_E3FF	1KB	CDIF controller
	0x4001_F000~0x4001_FFFF	4KB	CDIF(RTC_B, XTLF, VAO)

表 4-1 外设模块总线地址列表

4.3 RAM

4.3.1 概述

FM33LG0 含有一块 32KB RAM（8K*32）。

SRAM 地址空间范围是 0x2000_0000~0x2000_7FFF，软件可以对 SRAM 进行字节、半字、字访问，CPU 和 DMA 都可以以最大系统频率对 SRAM 实现无等待的单周期读写。CPU 也可以从 SRAM 取指执行程序，因此在对程序效率要求高的场合，可以将部分代码导入 SRAM 中，实现最高频率下无等待的执行。

4.4 Flash

4.4.1 概述

FM33LG0 使用的 Flash 容量为 64K*32，即 256KB；阵列组织格式包含 page(512B)、sector(2KB)、mat(256KB)

main array 共包含 512 个页，另有 4 个信息页。Flash 支持页擦、扇区擦和全擦。

4.4.2 特殊信息扇区说明

Flash 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）
RED	冗余信息	保存用于失效扇区替换的信息
IF	Information 区	4 个 page 共 2KB，供用户使用；软件可以读写

表 4-2Flash 特殊信息区

4.4.2.1 LDT0 page

LDT0 中保存原厂写入的芯片参数，软件只可读不可改写。

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 addr	Bit[31:16]	Bit[15:0]	Description
0x1FFF_FA84	~VREFREG45_TRIM	VREFREG45_TRIM	VREFP_REGU 4.5V trim值
0x1FFF_FA88	~VREFREG30_TRIM	VREFREG30_TRIM	VREFP_REGU 3.0V trim值
0x1FFF_FA8C	~VREFREG25_TRIM	VREFREG25_TRIM	VREFP_REGU 2.5V trim值
0x1FFF_FA90	~VREFREG20_TRIM	VREFREG20_TRIM	VREFP_REGU 2.0V trim值
0x1FFF_FA94	~VREFREG15_TRIM	VREFREG15_TRIM	VREFP_REGU 1.5V trim值
0x1FFF_FA98	~ULPBG_TRIM	ULPBG_TRIM	ULPBGtrim值
0x1FFF_FB08	~VREFCAL	VREFCAL	3V,30C 下ADC对VREF1p2的转换值
0x1FFF_FB0C	VREFRAW		VREF1p2实际电压值
0x1FFF_FB10	~TS_CAL	TS_CAL	3V, 30C下ADC对PTAT的转换值
0x1FFF_FB20	~RCLP_TRIM	RCLP_TRIM	RCLP调校值
0x1FFF_FB38	~RCHF24TRIM	RCHF24TRIM	RCHF 24MHz调校值
0x1FFF_FB3C	~RCHF16TRIM	RCHF16TRIM	RCHF 16MHz调校值
0x1FFF_FB40	~RCHF8TRIM	RCHF8TRIM	RCHF 8MHz调校值(auto-load)

0x1FFF_FB44	~RCLFTRIM	RCLFTRIM	RCLF调校值

表 4-3FlashLDT0 扇区

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	Bit[31:16]	Bit[15:0]	Description
VREFCAL	{4'h0, ~VREFCAL}	{4'hF, VREFCAL}	VDDA=VREFP=3V+/-10mV, 30C+/-1C 下 ADC 对 VREF1p2的转换值
VREFRAW			VREF1p2输出电压的实际测量值，数据格式TBD
RCLP_TRIM	{8'h00, ~trim}	{8'hFF, trim}	trim[7:0]表示8bit调校值
RCHF24TRIM RCHF16TRIM RCHF8TRIM	{8'b0000_0000, ~RCHFtrim[7:0]}	{8'b1111_1111,RCHFtrim[7:0]}	RCHFtrim[7:0]表示8bit调校值
VREFREG45_TRIM	{8'b0000_0000, ~VREFREGU_trim}	{8'b1111_1111, VREFREGU_trim}	VREFP_VREG 调校值 4.5V
VREFREG30_TRIM	{8'b0000_0000, ~VREFREGU_trim}	{8'b1111_1111, VREFREGU_trim}	VREFP_VREG 调校值 3.0V
VREFREG25_TRIM	{8'b0000_0000, ~VREFREGU_trim}	{8'b1111_1111, VREFREGU_trim}	VREFP_VREG 调校值 2.5V
VREFREG20_TRIM	{8'b0000_0000, ~VREFREGU_trim}	{8'b1111_1111, VREFREGU_trim}	VREFP_VREG 调校值 2.0V
VREFREG15_TRIM	{8'b0000_0000, ~VREFREGU_trim}	{8'b1111_1111, VREFREGU_trim}	VREFP_VREG 调校值 1.5V
ULPBG_TRIM	{11'b0000_0000_000, ~ULPBG_trim}	{11'b1111_1111_111, ULPBG_trim}	ULPBG_VDD 调校值 1.2V

表 4-4LDT0 数据格式

4.4.2.2 LDT1 page

LDT1为用户配置信息区，主要用于保存用户选项字节和Flash锁定信息。LDT1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，执行flash全擦后，才能擦除LDT1。

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，控制低 16 blocks
0x1FFF_FC0C	LOCK2		ACLOCK 配置字，控制

		高 16 blocks

表 4-5FlashLDT1 扇区

OPTBYTES 选项字节定义如下：

Bitfield	助记符	功能描述	出厂默认
31:24	BTSWPEN	启动区交换使能 0x55：允许启动区交换功能 其他：禁止启动区交换	0xFF
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 IWDT 计数 其他：任何模式下禁止应用停止 IWDT	0xA
19:16	DFLSEN	Data flash 使能 0x5：使能数据 flash，main array 的最高 16KB 地址被定义为 data flash 其他：禁止数据 flash	0xF
15:8	ACLKEN	应用代码保护使能 0x33：禁止 ACLOCK 其他：使能 ACLOCK	0x33
7:0	DBRDPEN	调试接口访问保护使能 0xAA：关闭调试接口保护 其他：使能调试接口保护	0xAA

表 4-6Flash 选项字节

【注】在出厂时，用户软件或者 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。

LOCK 信息用于配制 Flash 内容保护，以 8KB block 为单位进行访问权限保护。详情参见“Flash 内容保护”章节。

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（Flash 地址空间	0xFFFFFFFF

Bitfield	助记符	功能描述	出厂默认
		128~136KB），LOCK2[31:30]对应 Block31（Flash 地址空间 248~256KB），其他以此类推	

表 4-7Flash LOCK 配制

LOCK bit 和 Flash 权限锁定地址的对应关系如下表：

Address	LOCK bits
0x0000_0000 ~ 0x0000_1FFF	LOCK1[1:0]
0x0000_2000 ~ 0x0000_3FFF	LOCK1[3:2]
0x0000_4000 ~ 0x0000_5FFF	LOCK1[5:4]
0x0000_6000 ~ 0x0000_7FFF	LOCK1[7:6]
0x0000_8000 ~ 0x0000_9FFF	LOCK1[9:8]
0x0000_A000 ~ 0x0000_BFFF	LOCK1[11:10]
0x0000_C000 ~ 0x0000_DFFF	LOCK1[13:12]
0x0000_E000 ~ 0x0000_FFFF	LOCK1[15:14]
0x0001_0000 ~ 0x0001_1FFF	LOCK1[17:16]
0x0001_2000 ~ 0x0001_3FFF	LOCK1[19:18]
0x0001_4000 ~ 0x0001_5FFF	LOCK1[21:20]
0x0001_6000 ~ 0x0001_7FFF	LOCK1[23:22]
0x0001_8000 ~ 0x0001_9FFF	LOCK1[25:24]
0x0001_A000 ~ 0x0001_BFFF	LOCK1[27:26]
0x0001_C000 ~ 0x0001_DFFF	LOCK1[29:28]
0x0001_E000 ~ 0x0001_FFFF	LOCK1[31:30]
0x0002_0000 ~ 0x0002_1FFF	LOCK2[1:0]
0x0002_2000 ~ 0x0002_3FFF	LOCK2[3:2]
0x0002_4000 ~ 0x0002_5FFF	LOCK2[5:4]
0x0002_6000 ~ 0x0002_7FFF	LOCK2[7:6]
0x0002_8000 ~ 0x0002_9FFF	LOCK2[9:8]
0x0002_A000 ~ 0x0002_BFFF	LOCK2[11:10]
0x0002_C000 ~ 0x0002_DFFF	LOCK2[13:12]
0x0002_E000 ~ 0x0002_FFFF	LOCK2[15:14]
0x0003_0000 ~ 0x0003_1FFF	LOCK2[17:16]
0x0003_2000 ~ 0x0003_3FFF	LOCK2[19:18]
0x0003_4000 ~ 0x0003_5FFF	LOCK2[21:20]
0x0003_6000 ~ 0x0003_7FFF	LOCK2[23:22]
0x0003_8000 ~ 0x0003_9FFF	LOCK2[25:24]
0x0003_A000 ~ 0x0003_BFFF	LOCK2[27:26]
0x0003_C000 ~ 0x0003_DFFF	LOCK2[29:28]
0x0003_E000 ~ 0x0003_FFFF	LOCK2[31:30]

表 4-8LOCK 位和地址对应表

4.4.2.3 Information3 page

Flash还包含4个information页，其中information3用于控制BootSwap功能。Information3总线地址是0x0004_0600~0x0004_07FF；

在LDT1中BOOTSWAPEN=0x55的前提下，可以通过INFO3中最低地址数据内容来进行BootSwap操作。当数据为0x5454_ABAB时，芯片将Flash最低两个8KB空间的逻辑地址互换（注意，ACLOCK只按照逻辑地址处理，不考虑实际物理地址），从而实现启动代码无风险升级。

BootSwap示意图如下：

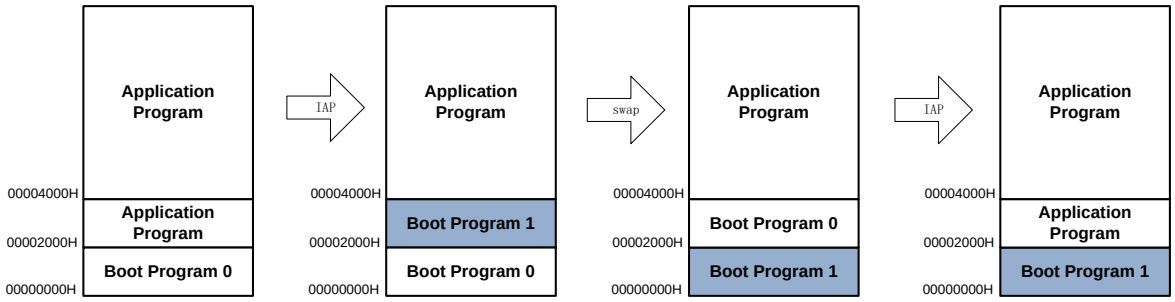


图 4-5Bootswap 示意图

其主要目的是，防止在系统更新启动代码时出现意外中断（停电、异常复位等），如果此时原来的启动代码已经被擦除，将导致芯片重启后无法正常运行。

实现BootSwap功能后，假设启动代码占据0000~1FFF共8KB空间，系统升级时应先将新的启动代码写入2000~3FFF地址，然后使能BootSwap。此时有几种可能性：

- 芯片擦写2000~3FFF地址时掉电，由于原来的启动代码不会被擦除，不会影响重启
- 芯片成功写入boot program1，然后使能bootswap并执行软复位，芯片重启后将执行boot program1
- 芯片擦写boot program0时掉电，由于boot program1已经写入，将不影响后续运行

推荐应用按照如下步骤升级：

- 更新application program
- 如需升级boot，先将新的boot程序写入第二个8KB空间
- 配置INFO3，使能BootSwap
- 执行软复位，重启后执行新的boot程序
- 将第二个8KB空间改写为新的应用程序

注意：IF3 的 BootSwap 功能实际只用到本页的最低一个 word，其余地址空间开放读写（无特定功能），软件或 Debugger 都可以任意写入数据。

4.4.2.4 Information1~2 page（Debugger only, lockable）

这2个information页开放给用户使用，仅SWD可以擦写，软件可读。

总线地址是0x0004_0200~0x0004_05FF，低地址为IF1，高地址为IF2，总共1KB。

IF2~1这2个页各自的最高地址字节为页锁定标志，如果SWD将最高地址字节改写为0x55，则芯片复位后当前页被禁止编程，SWD进行页擦后，可以重新编程。

不论是否有锁定标志，这几个页都是SWD和软件可读的。

4.4.2.5 Information0 page（OTP）

IF0是一个OTP页，这个页用户只能编程一次，编程后不能擦除或改写。IF0的总线地址是0x0004_0000~0x0004_01FF，共512字节。

对IF0 page的读取没有任何限制。

4.4.3 指令 Prefetch

FM33LG0 最高主频是 64MHz，当系统主频高于 24Mhz 时，需要在访问 flash 时插入 wait。Wait cycle 对程序执行效率造成一定影响，为改善 CPU 在高频下的运行效率，FM33LG0 实现了指令预取指以在高主频下降低 flash 等待带来的影响。

指令预取功能通过置位 PFTBUF_EN 和 PFTPHS_EN 寄存器使能。

4.4.4 Flash 编程

4.4.4.1 概述

FM33LG0 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，禁止对未经过擦除的 flash 地址进行重复编程。Flash 支持三种擦除操作：全擦、扇区擦、页擦

4.4.4.2 Flash 擦写时钟

执行 Flash 擦写时使用 RCHF 时钟，但是系统时钟可以是任意时钟。为了保证 flash 擦写定时的精度，建议在擦写 flash 之前对 RCHF 进行频率校准。Flash 擦写支持的 RCHF 频率为 8M、16M 和 24M。

4.4.4.3 Flash 擦写方法

FM33LG0 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：

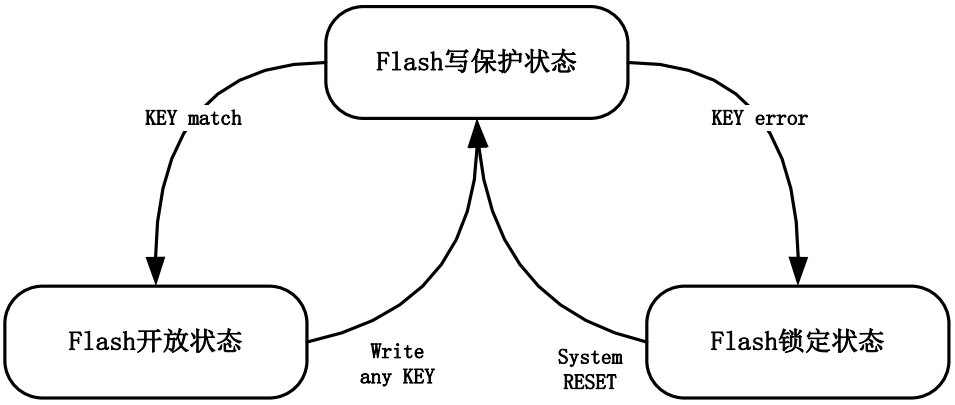


图 4-6Flash 擦写 Key 认证

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

4.4.4.4 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 main array，不会擦除特殊信息区。SWD 可以在制造商或用户模式下启动全擦，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
- 芯片启动对 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 main array 全部擦除，任何对 main array 的编程将清除此标志）
- 在全擦标志有效的情况下，SWD 可以任意擦写 LDT1，否则擦写 LDT1 被禁止并触发错误中断
- 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，不会影响特殊信息扇区

4.4.4.5 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 01
- 清除 PREQ 寄存器，置位 EREQ 寄存器
- 写入 Flash 块擦 Key：0x9696_9696 和 0x3C3C_3C3C
- 向需要擦除的扇区内任意地址写擦除请求 0x1234_ABCD
- 芯片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

4.4.4.6 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 Page 内任意地址写擦除请求 0x1234_ABCD
- 芯片检查目标 Page 是否属于被 ACLOCK 锁定的 Block，如果没有锁定则启动对目标 Page 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

4.4.4.7 单次编程

单次编程由软件发起，通过总线直接写 Flash，每次操作编程 32bits，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
- 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

4.4.4.8 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：

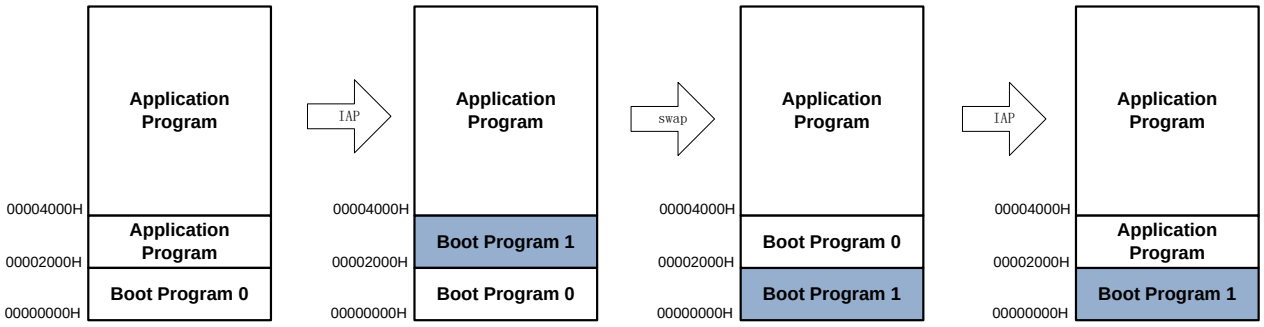
- 清除 EREQ 寄存器，置位 PREQ 寄存器
- 置位连续编程使能寄存器（DMA 模式使能）
- 向 RAM 中写入 256 字节待编程数据
- 配置 DMA 存储器通道，设定传输方向、读地址和写地址
- 使能 DMA 存储器通道
- 写入 Flash 编程 Key：0xA5A5_A5A5 和 0xF1F1_F1F1
- 软件触发 DMA 存储器通道，DMA 连续 64 次读取 RAM 并对 Flash 编程
- 芯片检查被编程扇区是否被 ACLOCK 锁定，如果锁定则触发错误中断并通知 DMA 停止编程
- 256 字节完全编程结束后产生中断，释放 Flash 总线
- 软件确认编程结束后向 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，应通过 VTOR 将中断向量表重新映射到 RAM 中。

4.4.4.9 启动区交换（BootSwap）

BootSwap主要目的是，防止在系统更新启动代码时出现意外中断（停电、异常复位等），如果此时原来的启动代码已经被擦除，将导致芯片重启后无法正常运行。BootSwap功能通过编程information page3最低地址word实现。

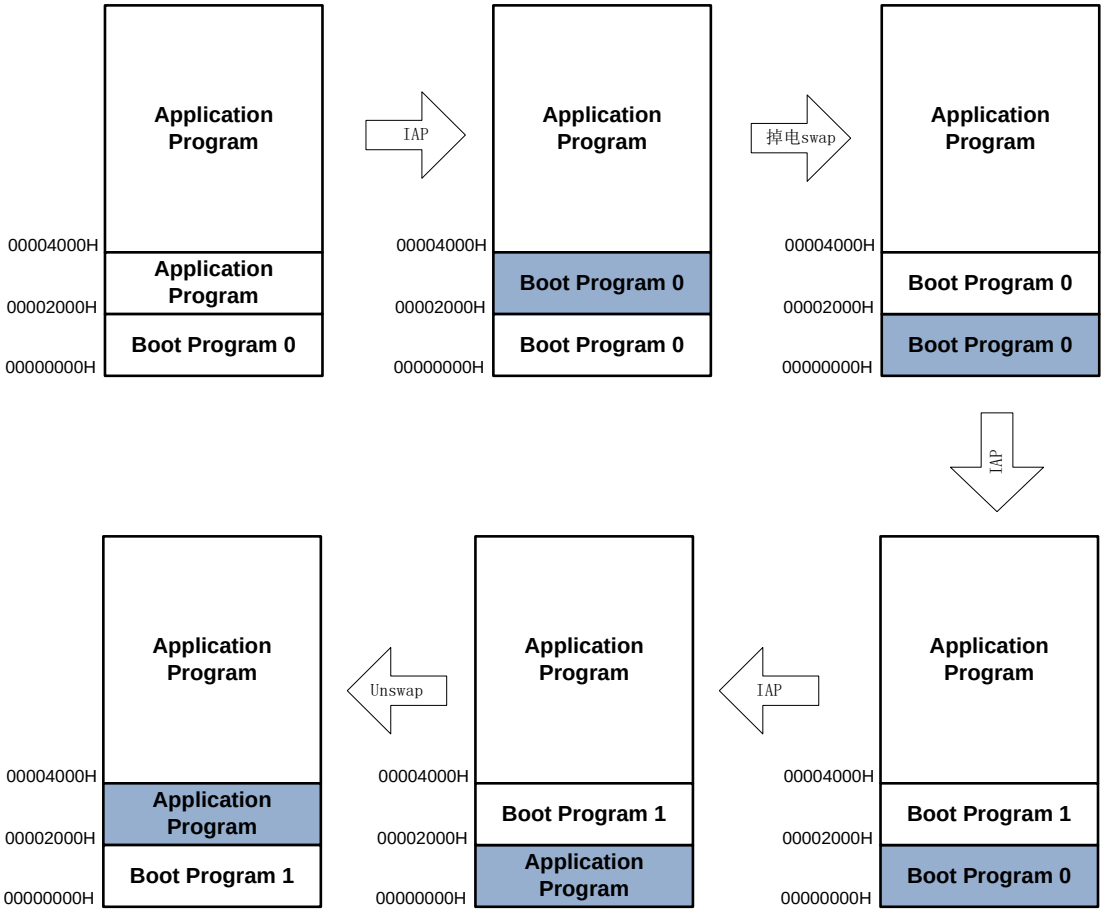
BootSwap示意图如下：



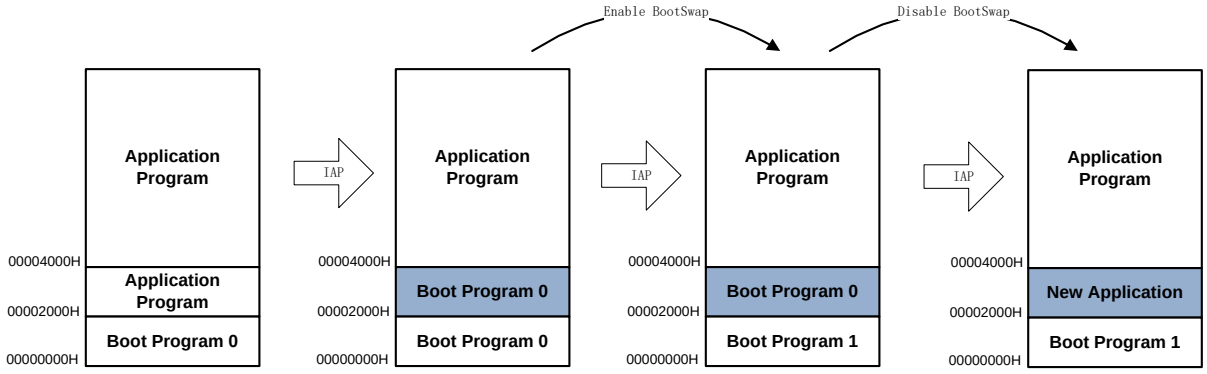
实现BootSwap功能后，假设启动代码占据0000~1FFF共8KB空间，系统升级时应先将新的启动代码写入2000~3FFF地址，然后使能BootSwap。此时有几种可能性：

- 芯片擦写2000~3FFF地址时掉电，由于原来的启动代码还在，不会影响重启
- 芯片成功写入boot program1，然后使能bootswap并执行软复位，芯片重启后将执行boot program1
- 芯片擦写boot program0时掉电，由于boot program1已经写入，将不影响后续运行

另一种BootSwap应用方法如下图，为了保证可靠的更新启动代码，使用2nd 8KB物理空间作为原来Boot程序的备份，如果编程期间发生异常掉电，则触发BootSwap：



如果启动程序更新期间没有发生异常掉电，则可以不执行软复位，无需真正Swap，仅需在更新原来的Boot程序前使能BootSwap，成功更新后撤销BootSwap即可：



推荐应用按照如下步骤升级：

- 更新application program
- 如需升级boot，先将新的boot程序写入第二个8KB空间
- 配置information block，使能BootSwap
- 执行软复位，重启后执行新的boot程序
- 将第二个8KB空间改写为新的应用程序

寄存器标志（FLS_ISR.BTSF）来表示当前的Boot区是1st 8KB物理地址、还是2nd 8KB物理地址，用于给用户代码查询当前启动情况。

4.4.5 Data Flash

当用户配置 OPTBYTES 使能了 DFLSEN 之后，FM33LG0 将开放 16KB 数据 flash 给用户用于数据存储。在使能了 data flash 之后，不同型号产品的 flash 容量划分如下表所示：

型号	Data flash size	Data flash address	Program flash size
FM33L04x	16KB	0x0003_C000~0x0003_FFFF	240KB
FM33L02x	16KB	0x0001_C000~0x0001_FFFF	112KB
FM33L01x	N/A	N/A	64KB

表 4-9dataflash 配置

注意：64KB flash 版本不支持 data flash

当使能了 data flash 之后，相应的程序 flash 空间会减少 16KB。Data flash 总是位于 flash 逻辑地址空间的最高 16KB。

Data flash 与 program flash 在访问权限上没有差别，同样受 DBRDP 和 ACLOCK 控制。但是当芯

片进行全擦操作时，data flash 不会被擦除。实际上，当使能了 data flash 之后，芯片不再支持真正的 mat erase（8ms），进行全擦操作时，芯片会按扇区逐一进行伪全擦，直到擦除完除了 data flash 以外的全部扇区（2ms/page）。而 data flash 不使能时，芯片支持 mat erase，执行全擦会擦除 main array 所有数据。

注：在使能 data flash 的情况下，芯片全擦时间显著增加（因为是扇区擦除），对于 256KB 容量型号，全擦时间从 8ms 增加到 240ms，对于 128KB 容量型号，全擦时间从 8ms 增加到 112ms。

4.4.6 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。

4.4.6.1 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 LDT1 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（mat erase 或者伪全擦），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。

4.4.6.2 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

表 4-10LOCK 位权限控制

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：不能使用 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 禁止读取将导致 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（mat erase 或伪全擦），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。

4.4.6.3 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
LDT1	ON	x	x	R[2]	R
	OFF	x	x	R/E/W	R
IF3	x	x	x	R/E/W	R/E/W
IF2,1,0	x	x	55	R/E	R
			others	R/E/W	R

表 4-11Flash 权限控制

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。

4.5 寄存器

offset 地址	名称	符号
FLASH(模块起始地址：0x40001000)		
0x00	Flash 读取控制寄存器 （Flash Read Control Register）	FLS_RDCR
0x04	预取指控制寄存器 （Flash Prefetch Control Register）	FLS_PFCR
0x08	用户配置字寄存器 （Flash Option Bytes Register）	FLS_OPTBR
0x0C	ACLOCK 寄存器 1 （Flash Application Code Lock Register1）	FLS_ACLOCK1
0x10	ACLOCK 寄存器 2 （Flash Application Code Lock Register2）	FLS_ACLOCK2
0x14	Flash 擦写控制寄存器 （Flash Erase/Program Control Register）	FLS_EPCR
0x18	Flash Key 输入寄存器 （Flash Key Register）	FLS_KEY
0x1C	Flash 中断使能寄存器 （Flash Interrupt Enable Register）	FLS_IER
0x20	Flash 标志寄存器 （Flash Interrupt Status Register）	FLS_ISR

4.5.1 Flash 读取控制寄存器（FLS_RDCR）

名称	FLS_RDCR							
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	-						WAIT	
位权限	U-0						R/W-00	

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1:0	WAIT	Flash 读等待周期配置 (Wait Cycles) 00/11：0 wait cycle 01：1 wait cycle 10：2 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

4.5.2 预取指控制寄存器（FLS_PFCR）

名称	FLS_PFCR							
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	-						PFTBUF_EN	PFTPHS_EN
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1	PFTBUF_EN	指令预取指缓冲区使能，在 WAIT==00 的情况下写 1 无效 (Prefetch Buffer Enable) 1：使能 Prefetch Buffer 0：禁止 Prefetch Buffer
0	PFTPHS_EN	指令预取使能，在 WAIT==00 的情况下写 1 无效 (Prefetch Phase Enable) 1：使能 Phase Prefetch 0：禁止 Phase Prefetch

4.5.3 用户配置字寄存器（FLS_OPTBR）

名称	FLS_OPTBR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IWDTSLP	-						
位权限	R-0	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					IF2LOCK	IF1LOCK	-
位权限	U-0					R-0	R-0	U-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					DFLSEN	BTSEN	
位权限	U-0					R-0	R-01	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				ACLOCKEN		DBRDPEN	
位权限	U-0				R-01		R-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)

位号	助记符	功能描述
		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
30:19	-	RFU：未实现，读为 0
18	IF2LOCK	Information2 区锁定标志 (IF2 Lock) 0：未锁定 1：已锁定，锁定后软件不可改写本扇区
17	IF1LOCK	Information1 区锁定标志 (IF1 Lock) 0：未锁定 1：已锁定，锁定后软件不可改写本扇区
16:11	-	RFU：未实现，读为 0
10	DFLSEN	DataFlash 使能 (DataFlash Enable) 0：无 data flash 1：有 data flash
9:8	BTSEN	BootSwap 功能使能 (BootSwap Enable) 00/01/11：禁止 BootSwap 功能 10：允许 BootSwap
7:4	-	RFU：未实现，读为 0
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
1:0	DBRDPEN	Debug Port 读取保护使能 (Debug Read Protection Enable) 00/01/11：DBRDP 不使能 10：DBRDP 使能

4.5.4 ACLOCK 寄存器 1（FLS_ACLOCK1）

名称	FLS_ACLOCK1							
offset	0x0C							
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 配置寄存器低 32bit，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

位号	助记符	功能描述
		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。

4.5.5 ACLOCK 寄存器 2（FLS_ACLOCK2）

名称	FLS_ACLOCK2							
offset	0x10							
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 配置寄存器高 32bit，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) 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。

4.5.6 Flash 擦写控制寄存器（FLS_EPCR）

名称	FLS_EPCR							
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	-						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	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
0	EREQ	Erase Request 软件置位，硬件完成擦除后自动清零

4.5.7 Flash Key 输入寄存器（FLS_KEY）

名称	FLS_KEY							
offset	0x18							
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	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)

4.5.8 Flash 中断使能寄存器（FLS_IER）

名称	FLS_IER							
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	-				OTPIE	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	OTPIE	OTP 编程错误中断使能，1 有效 (OTP program error Interrupt Enable)
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 Enable)
0	ERDIE	擦写完成标志中断使能，1 有效 (Erase Done Interrupt Enable)

4.5.9 Flash 标志寄存器（FLS_ISR）

名称	FLS_ISR							
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	-				KEYSTA			BTSF
位权限	U-0				R-000			R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				OTPERR	AUTHERR	KEYERR	CKERR
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	-						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	BTSF	BootSwap 标志寄存器 (BootSwap Flag) 0：启动程序区为 Flash 物理地址 0000H~1FFFH 1：启动程序区为 Flash 物理地址 2000H~3FFFH

位号	助记符	功能描述
15:12	-	RFU：未实现，读为 0
11	OTPERR	OTP page 编程权限错误，硬件置位，软件写 1 清零 (OTP program Error) 1：尝试对已编程的 OTP 字节进行编程 0：无 OTP 编程错误
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

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(VDDA)

芯片的主电源（VDD）的典型工作电压范围是 1.65~5.5V。

芯片上电时，复位释放阈值主要由 BOR 电路决定，其典型复位释放电压是 1.65V。

芯片下电时，如果使能了 BOR，下电复位阈值由 BOR 电路决定，可以由软件配置 BORCR.BOR_PDRCFG 获得 4 个阈值档位，默认值为 1.8V。如果没有使能 BOR，使能了 PDR，下电复位阈值由 PDR 电路决定，软件可以通过 PDRCR.CFG 配置 3 个档位，默认值 1.5V。

综上，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。

- VREFP

仅有少部分封装形式中有独立的 VREFP 引脚，VREFP 是 ADC 的基准电压输入，在 ADC 工作时，会从 VREFP 引脚抽取几十到上百 uA 的电流。大部分封装中，VREFP 都是和 VDDA 封装在一起的。

- VBAT

部分型号支持 VBAT 电源引脚，可以在 VDD 主电源失电时，将 RTC 电路自动切换到 VBAT 供电。VBAT 引脚应连接合适的备份电源，通常是备份电池或者超级电容。如果不使用备份电源，则系统上应将 VBAT 和 VDD 短接。

- VDD15

VDD15 是芯片内核电源，由一个线性电源稳压器产生 1.5V 电源输出。所有的数字电路、Flash、SRAM 和部分模拟电路工作在这个电源下。VDD15 引脚需要外挂 0.1~1uF 稳压电容。当主电源 VDD 跌落至 1.5V 以下时，稳压器输出将跟随 VDD 变化。

5.1.2 电源结构图

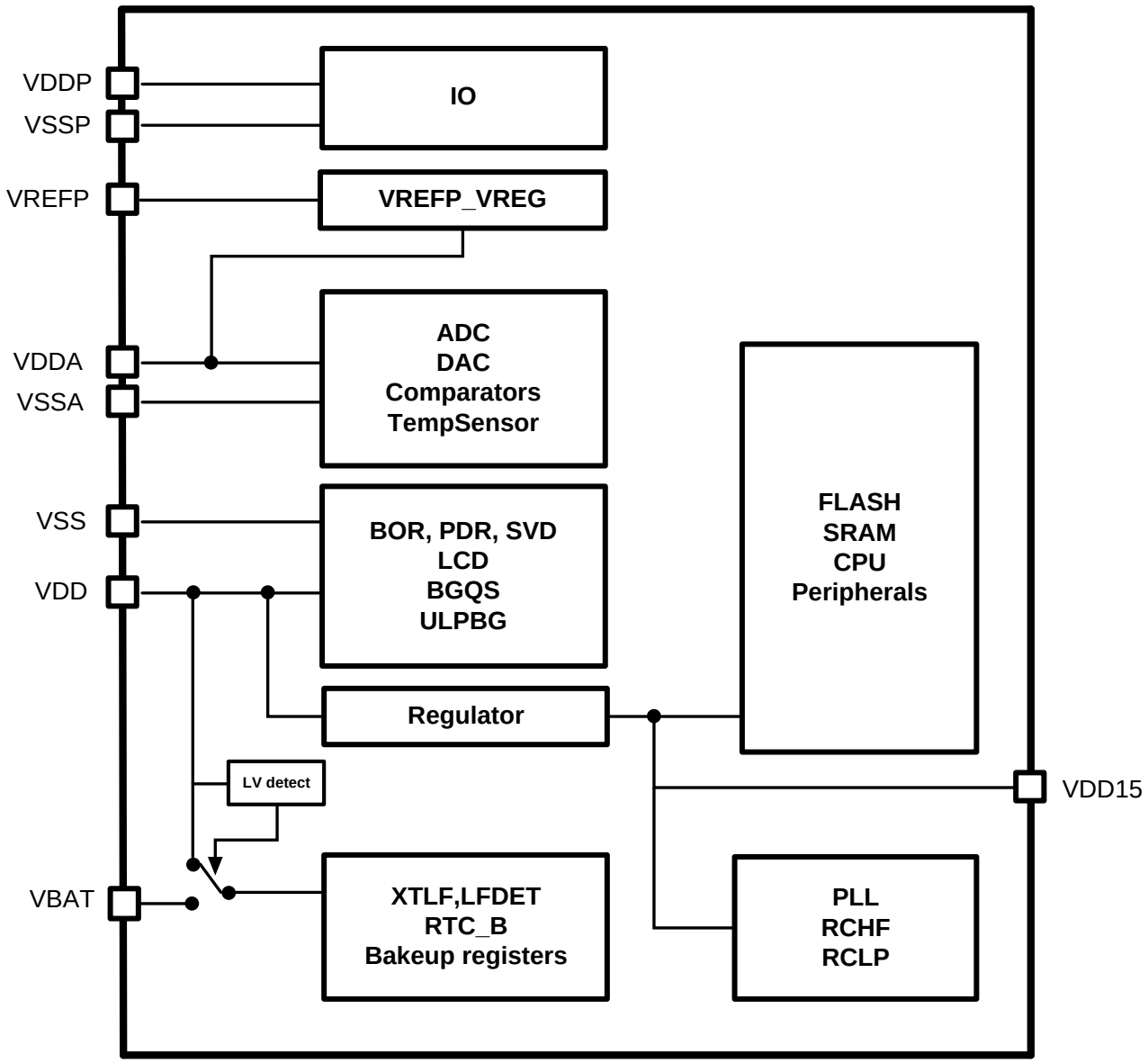


图 5-1 芯片电源结构图

5.1.3 ADC 和基准电压的独立供电

ADC 使用的电源地为 VDDA 和 VSSP，其参考电压是 VREFP。在部分封装形式中，VREFP 会被单独引出到芯片管脚，其中 VREFP 可以由系统提供，也可以在片内产生，或者直接在 PCB 上与 VDDA 短接在一起。

VREFP 与 VDDA 独立时，其输入基准可以与 VDDA 不同，允许的输入范围是：

1.6V ≤ VREFP ≤ VDDA

5.1.4 片内快速基准源（AVREF）

AVREF 电压为 1.0V，启动速度很快，使能后仅需 3us 左右即可建立 1V 输出电压，典型功耗小于 3uA。当芯片休眠时，AVREF 被自动关闭以节省功耗。

5.1.5 片内高精度基准源（VREF1p2）

FM33LGx0 集成了一个高精度基准源，典型输出电压为 1.2V 左右，在 1.6V≤VDDA≤5.5V 范围内都可以稳定工作。这个基准电压经过 Buffer 输出后，可以被 ADC 采样，也用于比较器的参考电压输入。这个基准电压经过升压驱动后，可以得到 2.0V、2.5V、3.0V、4.5V 的内部基准（不能高于电源电压），可以用于 ADC、DAC 和比较器的稳压基准源。

在整个工作温度范围内，此基准源的典型温度系数小于 25ppm/°C，同时内建了温度传感器输出，供 ADC 采样并测量当前芯片的基底温度。

软件可以开启或者关闭此基准源，打开基准源后，VREF1p2 输出建立时间小于 1ms，典型功耗为 1.5uA 左右。当打开温度传感器时，VREF1p2 功耗小于 2uA。

软件使能VREF1p2后，芯片内部在等待足够时间保证VREF输出完全建立后，置位VREF_DRY状态标志寄存器，并置位VREF_IF中断标志。软件可以自行定时或根据VREF_RDY寄存器来确认VREF1p2有效建立。

当软件关闭VREF1p2后，VREF_RDY寄存器被自动清零，VREF_IF由软件写1清零。

温度传感器最大支持的测温范围为-55~125°C，温度传感器输出电压随温度变化表现为一条正温度系数的直线，典型斜率为 5.1mV/°C。在芯片出厂前，温度传感器会在 30°C +/- 1°C 的条件下进行标定，在此条件下，-40~+85°C 范围内的温度测量误差在+/-2°C 以内。

5.1.6 VREFP 产生（VREFP_VREG）

参见“VREFP 基准电压”章节。

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
- LP Active 模式：LDO 进入低功耗模式，CPU 主频不超过 4MHz，所有外设可以运行
- LP Run 模式：LDO 工作在超低功耗模式下，CPU 和外设只能运行在较低频率下
- 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
- VBAT 模式：VDD 掉电，VBAT 通过备份电源供电

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	-
LP Active	500uA@4Mhz	软件主动退出	LDO 进入低功耗模式 关闭 XTHF、PLL CPU 运行频率不高于 600Khz	-
LP Run	30uA@32KHz	软件主动退出	低速工作	-
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、XTHF 等 保持 BGQS 开启，关闭 LDO15，RTC 走时 VREF1p2 由软件配置决定是否开启，开启的话增加 1.5uA 功耗	3us
DEEPSLEEP	1.5uA		CPU 休眠[2] 关闭 RCHF、PLL、XTHF 等 关闭 BG_QS，关闭 LDO15，RTC 走时	5us

			VREF1p22 由软件配置决定是否开启，开启的话增加 1.5uA 功耗	
VBAT	0.8uA	VDD 重新上电	VDD 掉电，仅 VBAT 供电	-

表 5-1 功耗模式表

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

在不同功耗模式下，CPU 主频的限制如下图所示：

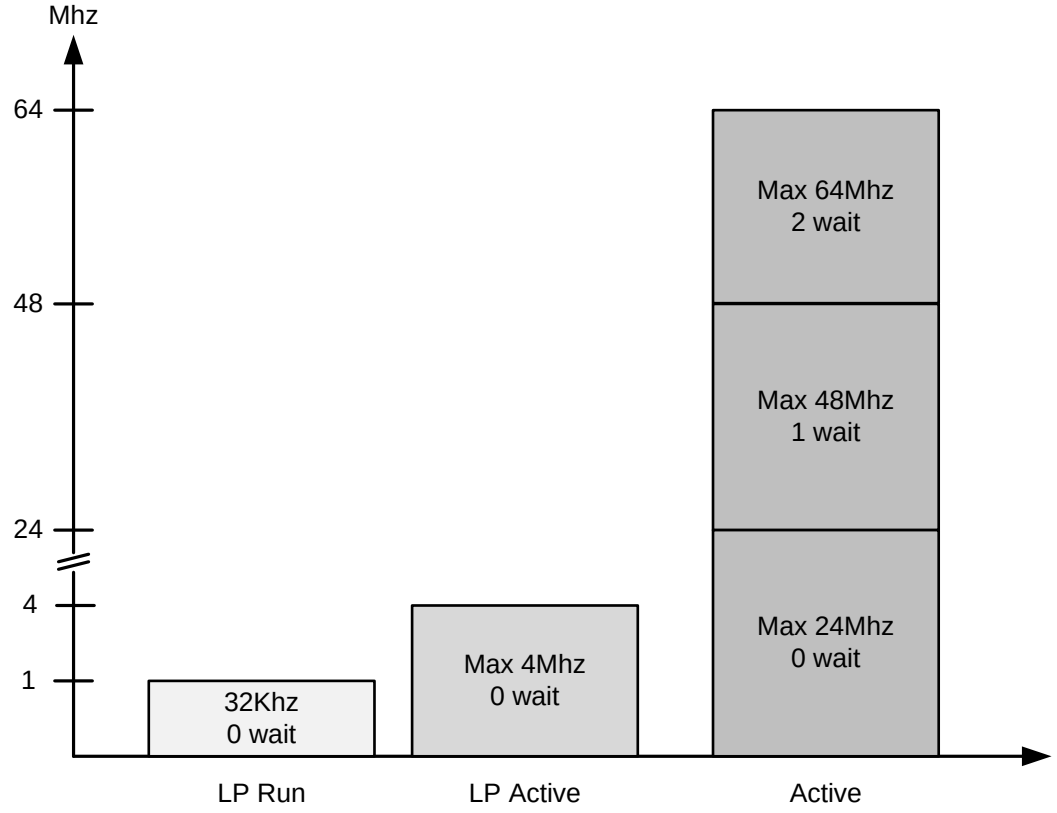


图 5-2 功耗模式与系统主频

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	

LP Active	4Mhz	RCHF, RCLF, XTLF, RCLP	0	RCHF, RCLF, XTLF, RCLP
LP Run	32Khz	RCLF, XTLF, RCLP	0	RCLF, XTLF, RCLP

表 5-2 功耗模式与频率对照表

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 64MHz。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，必须使能 Flash 等待功能，此时建议软件开启预取指功能来提升指令执行效率。

5.2.4 LP Active 模式

软件通过置位 LPMCFG.LDO_LPM 寄存器，可以进入 LP Active 模式。此时 LDO 被置于低功耗模式下，本身功耗下降的同时，驱动能力也有所下降。因此在 LP Active 模式下，CPU 主频不能超过 4MHz。同时，外设模块仍可以使用 RCHF、RCLF、XTLF、RCLP 工作，但是 PLL 和 XTHF 被硬件强制关闭，无法使用。

LP Active 模式的典型应用场景是，在对 CPU 处理能力要求不高的场景下，令 CPU 待机或者低速运行时，保持 1~2 个外设（如 UART、Timer）长时间正常运行，为一些特殊的低功耗场景提供最优的能效比。

进入 LP Active 模式

- 将系统时钟配置为 4MHz 或更低
- 确保没有外设正在使用 XTHF 或 PLL 时钟
- 置位 LDO_LPM 寄存器

LP Active 模式下的硬件行为

进入 LP Active 之后，硬件自动关闭 XTHF、PLL，随后使 LDO 进入低功耗模式。所有模拟和数字外设都可以工作。

退出 LP Active 模式

- 软件清零 LDO_LPM 寄存器
- 等待几条 NOP 指令
- 根据需要配置系统时钟，恢复正常的 Active 模式运行

LP Active 模式下 CPU 改写 PMOD 寄存器可以直接进入 LP RUN/SLEEP/DEEPSLEEP 模式。

5.2.5 LP Run 模式

当芯片需要低功耗低速运行时，可进入 LP RUN 模式，此时 LDO 进入低功耗模式，内核使用 XTLF、RCLP 或 RCLF 运行，典型频率 32KHz。在需要高速运行时，软件可主动退出 LP RUN 进入 ACTIVE 模式，然后再将系统时钟切换到较高频率。

进入 LPRUN 模式

进入 LPRUN 的操作步骤：

- 软件将系统时钟（SYSCLK）配置为 XTLF 或者 RCLP
- 配置 PMOD 寄存器为 01
- 如果系统时钟配置不满足以上寄存器条件，则置位异常中断并且禁止进入 LPRUN

LPRUN 模式下的硬件行为

进入 LP Run 之后，硬件自动关闭 RCHF、XTHF、PLL、TRNG，随后关闭 LDO15。SVD、比较器、ADC、DAC、OPA 仍可以在 LPRUN 模式下工作（VDDA 供电）。由于高速时钟都被关闭，ADC 工作时钟最高只有 RCLF，相当于最快 38Ksps 采样率。

如果软件在 LPRUN 模式下执行 WFI/WFE 指令，CPU 和 Flash 将停止活动，但是外设仍可以继续工作。

退出 LPRUN 模式

按照以下步骤退出 LPRUN 模式：

- 软件将 PMOD 寄存器配置为 00
- 软件根据需要使能 RCHF、XTHF 或 PLL
- 等待时钟建立后配置系统时钟为 RCHF、XTHF 或 PLL

LP Run 模式下 CPU 改写 PMOD 寄存器可以返回 ACTIVE，或者进入 SLEEP/DEEPSLEEP 模式。如果返回 ACTIVE，硬件自动将 LDO 置于正常模式，并解除对高速时钟模块的限制。

5.2.6 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、OPA、ADC、DAC、比较器仍可以在 SLEEP 模式下工作。其中由于高速时钟都被关闭，ADC 工作时钟最高只有 RCLF，相当于最快 38Ksps 采样率。

数字外设模块可以使用 RCLF、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

5.2.7 DEEPSLEEP 模式

DEEPSLEEP 是芯片 VDD 供电时的最低功耗模式，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。

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、OPA、ADC、DAC、比较器仍可以在 DEEPSLEEP 模式下工作。其中由于高速时钟都被关闭，ADC 工作时钟最高只有 RCLF，相当于最快 38Ksps 采样率。DeepSleep 模式下可以有选择的使能或关闭 VREF1p2，根据外设功能的需要由软件灵活设置。

数字外设模块可以使用 RCLF、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

5.2.8 VBAT 模式

对于封装管脚中包含 VBAT 引脚的芯片，当 VDD 掉电时，片内电源切换电路会自动将备份电源域的供电切换到 VBAT 引脚。此时芯片仅能维持 RTC_B 的走时和备份寄存器内容。详情请参见 8 备份电源域。

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	√	√
VREF	可屏蔽，在 VREF1p22 建立后产生中断唤醒芯片	√	√
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	√	√
ADC	可屏蔽，ADC 的各种中断均可用于唤醒	√	√
RTC_A RTC_B	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	√	√
UART0/1_RXD	可屏蔽，下降沿唤醒	√	√
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	√	√
BSTIM32	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	√	√
I2C 从机	可屏蔽，用于从机接收唤醒	√	√

表 5-3 休眠模式唤醒源列表

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。

5.3.1 VREF1p2 延迟唤醒功能

在周期性定时唤醒的应用场景下，可能希望在Vref1p2建立后再唤醒MCU，延迟唤醒功能可以实现这一目标。

在使能了延迟唤醒功能的情况下，当周期性定时唤醒事件发生时，MCU不会被立刻唤醒，而是自动使能Vref1p2，并等待可配置延迟时间后，再唤醒MCU。如果Vref1p2本来就是使能的，则此功能无效。并且延迟唤醒仅针对定时唤醒事件有效（RTC和定时器，即必须有工作时钟），对于WKUPx、引脚、模拟唤醒源等唤醒事件是无效的；不论延迟唤醒是否使能，当非定时事件到来时，MCU都是立即唤醒。

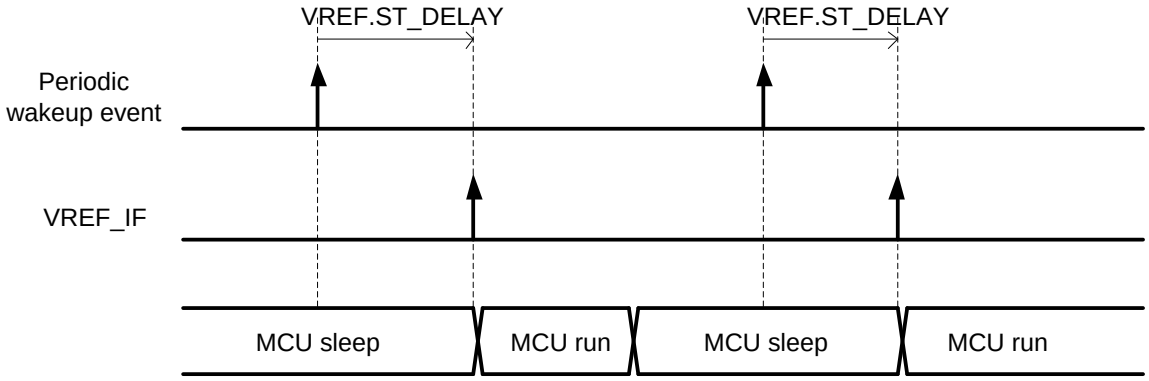


图 5-3VREF1p2 延迟唤醒时序图

唤醒源	VREF 延迟唤醒
停振检测	X
VREF	X
SVD	X
比较器	X
ADC	X
RTC	O
IO 引脚中断	X
Debug	X
LPUART	X
WKUPx 引脚	X
NRST	X
LPTIM16/32	O
BSTIM16/32	O
I2C 从机	X

表 5-4VREF1p2 延迟唤醒可应用的唤醒源

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）。最快情况下芯片唤醒后将以24MHz时钟启动。

休眠时AHBPRES寄存器不会复位，但是SYSCLKSEL寄存器将复位成00（选择RCHF）。因此，如果休眠前系统时钟不是RCHF，则唤醒后将默认使用RCHF，是否经过分频由休眠前的AHBPRES寄存器决定。

5.5 寄存器

地址	名称	符号
PMU(模块起始地址：0x40002000)		
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
0x38	ULPBG 调校寄存器 （ULPBG trim Register）	PMU_ULPB_TR
0x3C	VREFP 控制寄存器 （VREFP Control Register）	PMU_VREFP_CR
0x40	VREFP 配置寄存器 （VREFP Config Register）	PMU_VREFP_CFGR
0x44	VREFP 状态标志寄存器 （VREFP Interrupt Status Register）	PMU_VREFP_ISR
0x48	VREFP 调校寄存器 （VREFP Trim Register）	PMU_VREP_TR

5.5.1 低功耗控制寄存器（PMU_CR）

名称	PMU_CR							
offset	0x00							
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	-				LDO_LPM		LDO15EN	LDO15EN_B
位权限	U-0				R/W-01		R/Dy-1	R/Dy-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				WKFSEL		SLPDP	CVS
位权限	U-0				R/W-00		R/W-0	R/W-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	

位号	助记符	功能描述
31	**RFUI**	Dummy 寄存器
30:20	**--**	未实现，读为 0
19:18	**LDO_LPM**	LDO 低功耗模式配置 00/01/11：正常模式 10：LDO 进入低功耗模式

位号	助记符	功能描述
17	**LDO15EN**	LDO15 使能标志位 1：LDO15 处于工作状态 0：LDO15 被关闭
16	**LDO15EN_B**	LDO15 使能标志反码校验位
15:12	**--**	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	**CVS**	CoreVoltageScaling 配置 0：低功耗模式下不使能内核电压调整 1：低功耗模式下降低内核电压 该位仅在 Sleep/DeepSleep 模式下起作用
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 / LP Active mode 01：LPRUN mode 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	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	-				VREFDLY	STPCLR	T1a	
位权限	U-0				R/W-0	R/W-0	R/W-01	

位号	助记符	功能描述
31:4	**--**	未实现，读为 0
3	**VREFDLY**	VREF 唤醒延迟功能

位号	助记符	功能描述
		0：唤醒事件到来后立即唤醒 MCU 1：唤醒事件到来后启动 VREF1p2，并等待 VREF1p2 建立后再唤醒 MCU *注：* *1）此功能仅针对定时唤醒有效* *2） 如果 VREF1p2 休眠时没有关闭，则此功能无效*
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 11：8us

5.5.3 唤醒源标志查询寄存器（PMU_WKFR）

名称	PMU_WKFR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ADCWKF	UART1WKF	UART0WKF	RTCWKF	SVDWKF	LFDETWKF	VREFWKF	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	I2CWKF	LPU2WKF	LPU1WKF	LPU0WKF	--	COMP3WKF	COMP2WKF	COMP1WKF
位权限	R-0	R-0	R-0	R-0	U-0	R-0	R-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	--	LPT32WKF	LPT16WKF	BST32WKF	BST16WKF	DBGWKF	WKPxF[9:8]	
位权限	U-0	R-0	R-0	R-0	R-0	R/W-0	R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WKPxF[7:0]							
位权限	R/W-00000000							

位号	助记符	功能描述
31	**ADCWKF**	ADC 中断唤醒标志，中断撤销时硬件自动清零
30	**UART1WKF**	UART1 接收下降沿异步唤醒标志，中断撤销时硬件自动清零
29	**UART0WKF**	UART0 接收下降沿异步唤醒标志，中断撤销时硬件自动清零
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	**VREFWKF**	VREF1P22 基准源建立中断唤醒标志，中断撤销时硬件自动清零

位号	助记符	功能描述
24	**IOWKF**	IO 中断唤醒标志，中断撤销时硬件自动清零
23	**I2CWKF**	I2C 中断唤醒标志，中断撤销时硬件自动清零
22	**LPU2WKF**	LPUART2 中断唤醒标志，中断撤销时硬件自动清零
21	**LPU1WKF**	LPUART1 中断唤醒标志，中断撤销时硬件自动清零
20	**LPU0WKF**	LPUART0 中断唤醒标志，中断撤销时硬件自动清零
19	**--**	未实现，读为 0
18	**COMP3WKF**	比较器 3 中断唤醒标志，中断撤销时硬件自动清零
17	**COMP2WKF**	比较器 2 中断唤醒标志，中断撤销时硬件自动清零
16	**COMP1WKF**	比较器 1 中断唤醒标志，中断撤销时硬件自动清零
15	**--**	未实现，读为 0
14	**LPT32WKF**	LPTIM32 中断唤醒标志，中断撤销时硬件自动清零
13	**LPT16WKF**	LPTIM16 中断唤醒标志，中断撤销时硬件自动清零
12	**BST32WKF**	BSTIM32 中断唤醒标志，中断撤销时硬件自动清零
11	**BST16WKF**	BSTIM16 中断唤醒标志，中断撤销时硬件自动清零
10	**DBGWKF**	CPU Debugger 唤醒标志，软件写 1 清零
9:0	**WKPxF**	NWKUPx 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	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	-					LPACTIE	SLPEIE	LPREIE
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
2	**LPACTIE**	LPACTIVE 错误中断使能 1：使能 LPACTIVE 错误中断 0：禁止 LPACTIVE 错误中断
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	**LPREIE**	LPRUN 错误中断使能 1：使能 LPRUN 错误中断 0：禁止 LPRUN 错误中断

5.5.5 PMU 中断标志寄存器（PMU_ISR）

名称	PMU_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	-					LPACTIF	SLPEIF	LPREIF
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
2	**LPACTIF**	LPACT 错误中断标志，硬件置位，软件写 1 清零 1：置位 LDO15LPM 寄存器时，系统工作时钟为 RCHF 且大于 4Mhz，或系统时钟为 PLL、XTHF 0：正常进入 LPACTIVE
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	**LPREIF**	LPRUN 错误中断标志，硬件置位，软件写 1 清零；软件进入 LPRUN 模式时如果触发了 LPREIF，则芯片仍将停留在 ACTIVE 模式 1：LPRUN Condition Error，即进入 LPRUN 时满足如下情况： 1） 系统时钟不是 LSCLK 或 RCLF，或 2） RCHF 使能未关闭 0：LPRUN 正常进入

5.5.6 ULPBG 调校寄存器（PMU_ULPB_TR）

名称	PMU_ULPB_TR							
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	-							
位权限	U-0							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			ULPBG_TRIM				
位权限	U-0			R/W-10000				

位号	助记符	功能描述
31:5	--	RFU：未实现，读为 0
4:0	ULPBG_TRIM	ULPBG 输出的 1.16V 基准电压调校寄存器

5.5.7 VREFP 控制寄存器（PMU_VREFP_CR）

名称	PMU_VREFP_CR							
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	-					DEND_IE	POV_IE	EN
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	DEND_IE	驱动结束中断使能 (Driving end interrupt enable) 1：允许 DEND 中断 0：禁止 DEND 中断
1	POV_IE	间歇使能周期结束中断使能 (Periodic overflow interrupt enable) 1：允许 POV 中断 0：禁止 POV 中断
0	EN	VREFP_VREG 使能寄存器 (enable) 0：关闭并 bypass VREFP_VREG 1：使能 VREFP_VREG

5.5.8 VREFP 配置寄存器（PMU_VREFP_CFGR）

名称	PMU_VREFP_CFGR							
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	-					VRS		
位权限	U-0					R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TPERIOD			TDRV			LPM	-
位权限	R/W-000			R/W-000			R/W-0	U-0

位号	助记符	功能描述
31:10	--	RFU：未实现，读为 0
10:8	VRS	输出电压选择 (Voltage regulation select) 000：VREFP 输出 2.0V 001：VREFP 输出 2.5V 010：VREFP 输出 3.0V 011：VREFP 输出 4.5V 1xx：VREFP 输出 1.5V
7:5	TPERIOD	间歇使能模式下的使能周期 (Time of Period) 000：1ms 001：4ms 010：16ms 011：32ms 100：64ms 101：256ms 110：1s 111：4s
4:2	TDRV	间歇使能模式下的单次驱动时间 (Time of Driving) 000：4*T_{LSCLK} 001：8*T_{LSCLK} 010：16*T_{LSCLK} 011：32*T_{LSCLK} 100：64*T_{LSCLK} 101：128*T_{LSCLK} 110：256*T_{LSCLK} 111：512*T_{LSCLK} *注 1：如果 TDRV 大于 TPERIOD，则 VREFP_VREG 将保持常使能。* *注 2：如果 VREF1p2 被关闭，则单次驱动时间不包含 VREF1p2 启动后的建立时间*
1	LPM	间歇使能寄存器 (Low power mode) 0：常使能模式 1：间歇使能模式
0	--	RFU：未实现，读为 0

5.5.9 VREFP 状态标志寄存器（PMU_VREFP_ISR）

名称	PMU_VREFP_ISR							
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	-					BUSY	DEND	POV
位权限	U-0					R-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	BUSY	驱动标志，只读 (Busy) 0：VREG 当前没有在驱动 1：VREG 正在驱动
1	DEND	驱动结束，仅间歇使能模式下有效 (Driving End) 每次驱动周期结束后硬件置位，软件写 1 清零 当 DEND_IE 置位时，DEND 置位会触发中断。
0	POV	间歇使能周期溢出，仅间歇使能模式下有效 (Periodic Overflow) 每个间歇使能周期结束后硬件置位，软件写 1 清零 当 POV_IE 置位时，POV 置位会触发中断。

5.5.10 VREFP 调校寄存器（PMU_VREFP_TR）

名称	PMU_VREFP_TR							
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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIM							
位权限	R/W-1000 0000							

位号	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	TRIM	VREFP 输出电压调校 (trimming) 每 LSB 对应步长 0.1%，最多可以加 127steps 或者减 127steps，对应调校范围-12.7% ~ +12.7%

6 高精度基准源（VREF1p2）

6.1 概述

FM33LG0 集成了一个高精度基准源，典型输出电压为 1.2V 左右，在芯片整个工作电源范围内都可以稳定工作。这个基准电压经过 Buffer 输出后，可以被 ADC 采样，也用于比较器的参考电压输入。在整个工作温度范围内，此基准源的典型温度系数小于 25ppm/°C，同时内建了温度传感器输出，供 ADC 采样并测量当前芯片的基底温度。

软件可以开启或者关闭此基准源，打开基准源后，VREF1p2 输出建立时间小于 1ms，典型功耗为 1.5uA 左右。当打开温度传感器时，VREF1p2 功耗小于 2uA。

软件使能VREF1p2后，芯片内部有一个硬件延迟电路，在等待足够时间保证VREF输出完全建立后，置位VREF_DRY状态标志寄存器，并置位VREF_IF中断标志。软件可以自行定时或根据VREF_RDY寄存器来确认VREF1p2有效建立。

当软件关闭VREF1p2后，VREF_RDY寄存器被自动清零，VREF_IF由软件写1清零。

温度传感器最大支持的测温范围为-55~125°C，温度传感器输出电压随温度变化表现为一条正温度系数的直线，典型斜率为 5.1mV/°C。在芯片出厂前，温度传感器会在 30°C +/- 1°C 的条件下进行标定，在此条件下，-40~+85°C 范围内的温度测量误差在+/-2°C 以内。

6.2 基准电压的应用

VREF1p2输出可用于以下用途：

- VREFP_VREG的参考输入
- ADC测量输入（VREF1p2和PTAT）
- 比较器输入参考电压

6.3 温度传感器

VREF1p2内建的PTAT电路能够输出正温度系数的电压信号，由于其输出电压与温度变化的斜率是确定的，通过ADC测量PTAT电压可以实现对当前芯片基底温度的测量。为了保证绝对温度测量精度，芯片出厂前都会在VDDA=3V+/-10mV，T_A=30C+/-1C的条件下进行温度传感器输出定标，并将定标数据（TS_CAL1）写入Flash。应用中，根据温度定标数据，以及电压-温度拟合线的斜率，可以计算当前芯片基底的绝对温度。在-40~+85C的全温区范围内，绝对温度测量误差小于+/-2C。

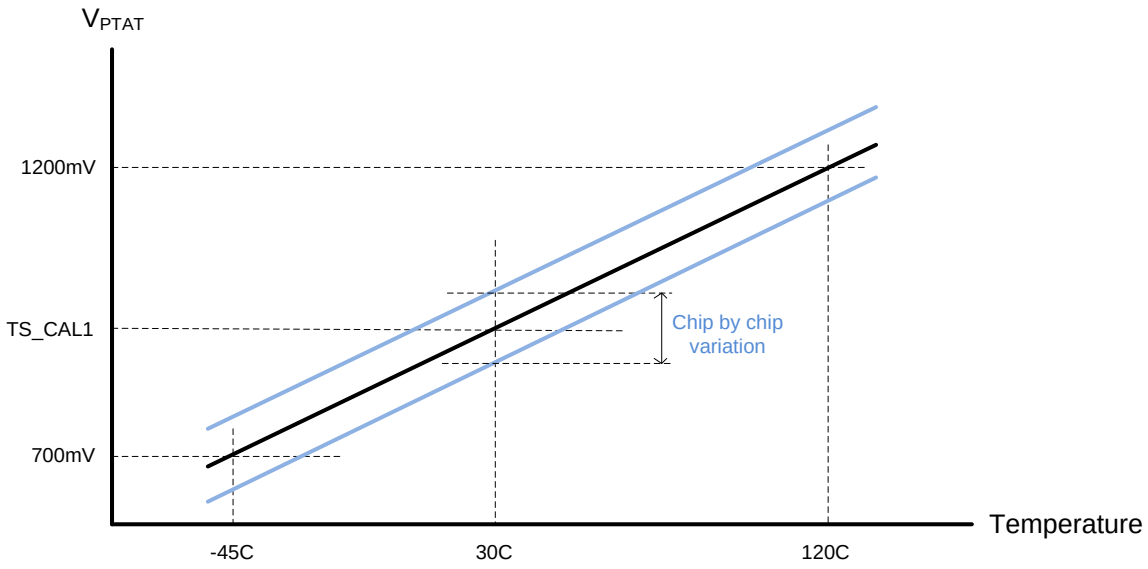


图 6-1 温度传感器输出与标定

在-40~+85C范围内，VDD=1.8~5.5V范围，PTAT输出相对温度变化的典型斜率是5.1mV/C，PTAT输出电压范围大致为700mv~1100mV。

6.4 输出缓冲器

VREF1p2输出和PTAT输出都内建了输出Buffer，以加强信号驱动。在使用ADC采样VREF1p2和PTAT输出时，建议使能内建Buffer，缩短采样保持时间。如果不使能VREF内部Buffer，则应使能ADC慢速采样通道的内建Buffer。使用VREF1p2内建Buffer时，建议采样时间不小于10us。

6.5 芯片休眠

当芯片休眠时，VREF1p2是否保持工作由软件决定。

如果软件关闭了VREF1p2，则VREFP_VREG将无法保持输出。软件可以使用VREF_VREG的间歇使能模式，在这种模式下，VREF_VREG定时开启，如果VREF1p2被关闭，硬件也会自动打开VREF1p2。详细参见7VREFP基准电压（VREFP_VREG）。

6.6 寄存器

模块起始地址：0x4001A400

offset 地址	名称	符号
VREF1p2(模块起始地址：0x4001A400)		
0x00	VREF1p2 控制寄存器 （VREF Control Register）	VREF_CR
0x04	VREF1p2 配置寄存器 （VREF Config Register）	VREF_CFGR
0x08	VREF1p2 标志寄存器 （VREF Status Register）	VREF_ISR
0x0C	VREF1p2 中断使能寄存器 （VREF Interrupt Enable Register）	VREF_IER
0x10	模拟 BUFFER 控制寄存器 （Buffer Control Register）	VREF_BUFCR

6.6.1 VREF1p2 控制寄存器（VREF_CR）

名称	VREF_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	-						PTAT_EN	VREF_EN
位权限	U-0						R/W-0	R/W-1

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1	PTAT_EN	Bandgap 温度传感器使能 (Temperature sensor enable) 0：关闭温度传感器输出 1：使能温度传感器输出
0	VREF_EN	VREF1p2 使能寄存器 (Voltage reference enable) 0：关闭 VREF1p2 1：使能 VREF1p2

6.6.2 VREF1p2 配置寄存器（VREF_CFGR）

名称	VREF_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	-						ST_DELAY	
位权限	U-0						R/W-00	

位号	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1:0	ST_DELAY	启动延迟时间 (Start Delay)，VREF_EN 寄存器置位使能 VREF1p2 输出后，等待可配置时间，硬件自动置位 VREF_RDY 寄存器，表示基准电压已经建立完成。 00：1.5ms 01：1ms 10：750us 11：500us

6.6.3 VREF1p2 标志寄存器（VREF_ISR）

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

位号	助记符	功能描述
31:9	-	RFU：未实现，读为 0
8	FLAG_B	模拟输出的 vref1p2 寄存器电压建立标志 (VREF setable Flag from analog, auto to clear)
7:2	-	RFU：未实现，读为 0
1	RDY	VREF1p2 基准电压建立标志 (VREF Ready Flag, auto to clear) VREF1p2 使能后，通过数字电路延迟置位，软件只读。关闭 VREF1p2 模块后，此寄存器自动清零。 数字电路延迟时间 1.5ms

位号	助记符	功能描述
0	IF	VREF1p2 基准电压建立中断 (VREF Ready Interrupt Flag, write 1 to clear) 0：VREF1p2 没有建立 1：VREF1p2 建立完成 此标志在 VREF1p2 使能后，通过数字电路延迟置位，硬件置位，软件写 1 清零

6.6.4 VREF1p2 中断使能寄存器（VREF_IER）

名称	VREF_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	VREF1p2 基准电压建立中断使能 (VREF Ready Interrupt Enable) 0：禁止产生 VREF 建立完成中断 1：允许产生 VREF 建立完成中断 此寄存器为 1 的情况下，当 VREF1p2 建立完成后，将输出中断给 CPU

6.6.5 模拟 BUFFER 控制寄存器（VREF_BUFCR）

名称	VREF_BUFCR							
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	-	AVREFBUF_OUTEN	AVREFBUF_EN	VPTATBUFFER_OUTEN	VPTATBUFFER_EN	VREFBUFFER_OUTEN	VREFBUFFER_EN
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	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	VREFBUFFER_OUTEN	Vref Buffer 模块开关通道输出使能信号，高电平使能有效。(VREF Buffer Output Enable)
0	VREFBUFFER_EN	Vref Buffer 模块使能信号，高电平使能有效。(VREF Buffer Enable)

7 VREFP 基准电压（VREFP_VREG）

7.1 概述

FM33LG0可以从VREFP引脚获取外部基准电压，或者由片内电路产生多档可配置基准电压。

内部基准电压由VREFP_VREG电路产生，VREFP_VREG使用VREF1p2为参考电压，获得4.5V、3.0V、2.5V、2.0V、1.5V输出，需要在VREFP引脚上外挂1uF稳压电容。

使用外部基准时，外部基准电压直接外加到VREFP引脚，与此同时必须关闭VREFP_VREG电路，以避免电平冲突。

VREFP_VREG产生的基准电压，除了可以给内部电路使用（如ADC、DAC、比较器等），还可以供片外使用。VREFP_VREG驱动能力为10mA。

为降低芯片整体功耗，VREFP_VREG还支持间歇启动模式，通过间歇使能的方式维持片外电容上的电压，这样可以大幅降低休眠功耗并仍然将VREFP基准电压维持在可接受的范围内。

7.2 功能描述

7.2.1 概述

VREFP_VREG电路可以用于产生恒定的内部基准电压，此时可以使ADC、DAC的基准电源不依赖于外部电源，因此也就不受外部电源波动的影响。

ADC的基准源可以在VREFP、VDDA之间切换，而DAC的基准源固定接VREFP。

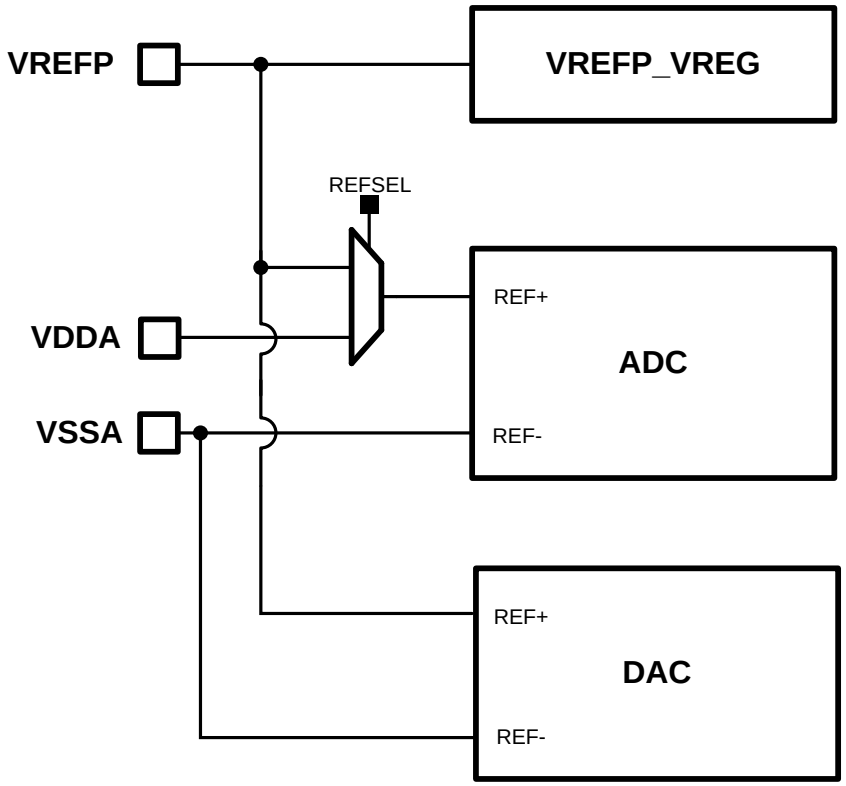


图 7-1 ADC 和 DAC 基准源

系统方案上可以同时支持外部基准和内部基准。

7.2.1.1 外部电源为基准

当使用外部电源作为基准时，应用应保持VREFP_VREG关闭，并在系统上短接VREFP和VDDA引脚，此时ADC和DAC的基准源在系统上是等价的。

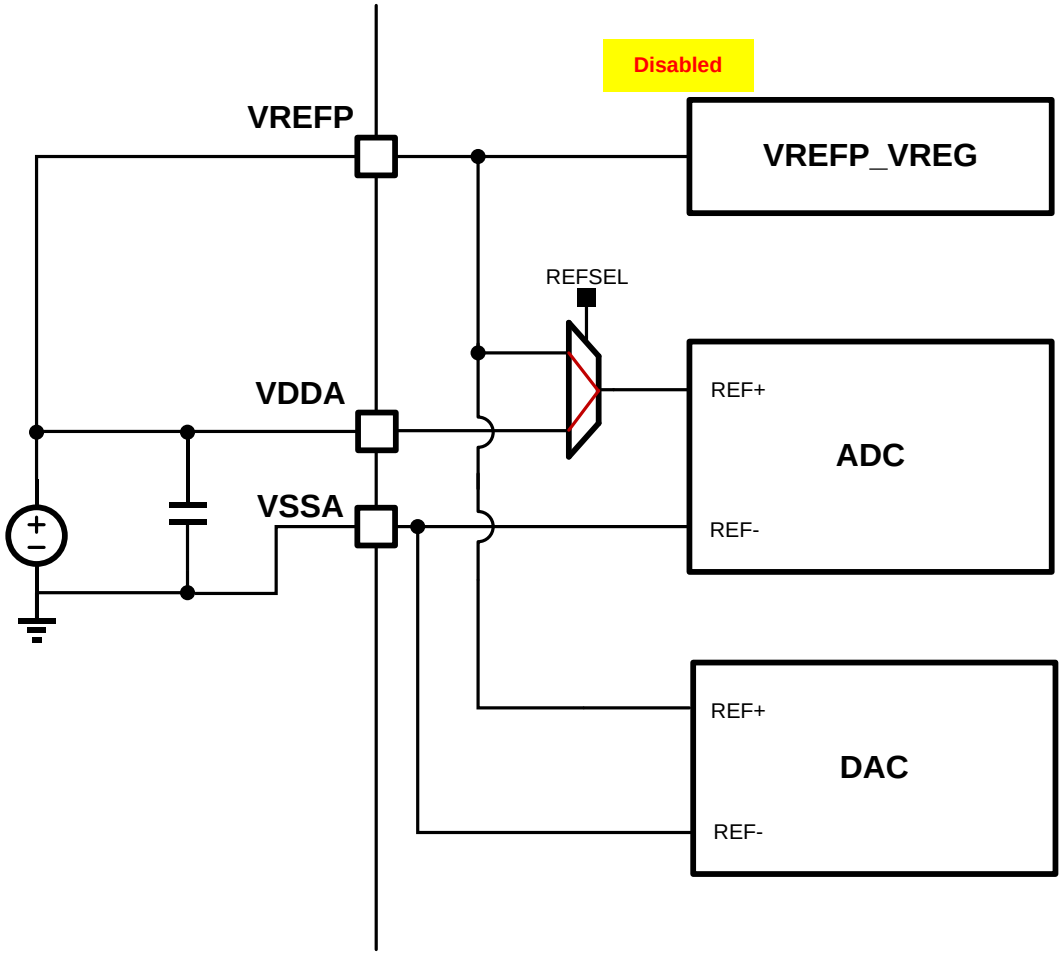


图 7-2 外部电源基准方案 1

7.2.1.2 外部输入独立基准

也可以支持系统上提供一个独立于电源的基准电压，此时VREFP_VREG保持关闭，ADC和DAC的基准源选择从VREFP提供。这个方案里PCB上不能短接VREFP和VDDA，对应的封装形式必须为独立VREFP引脚。

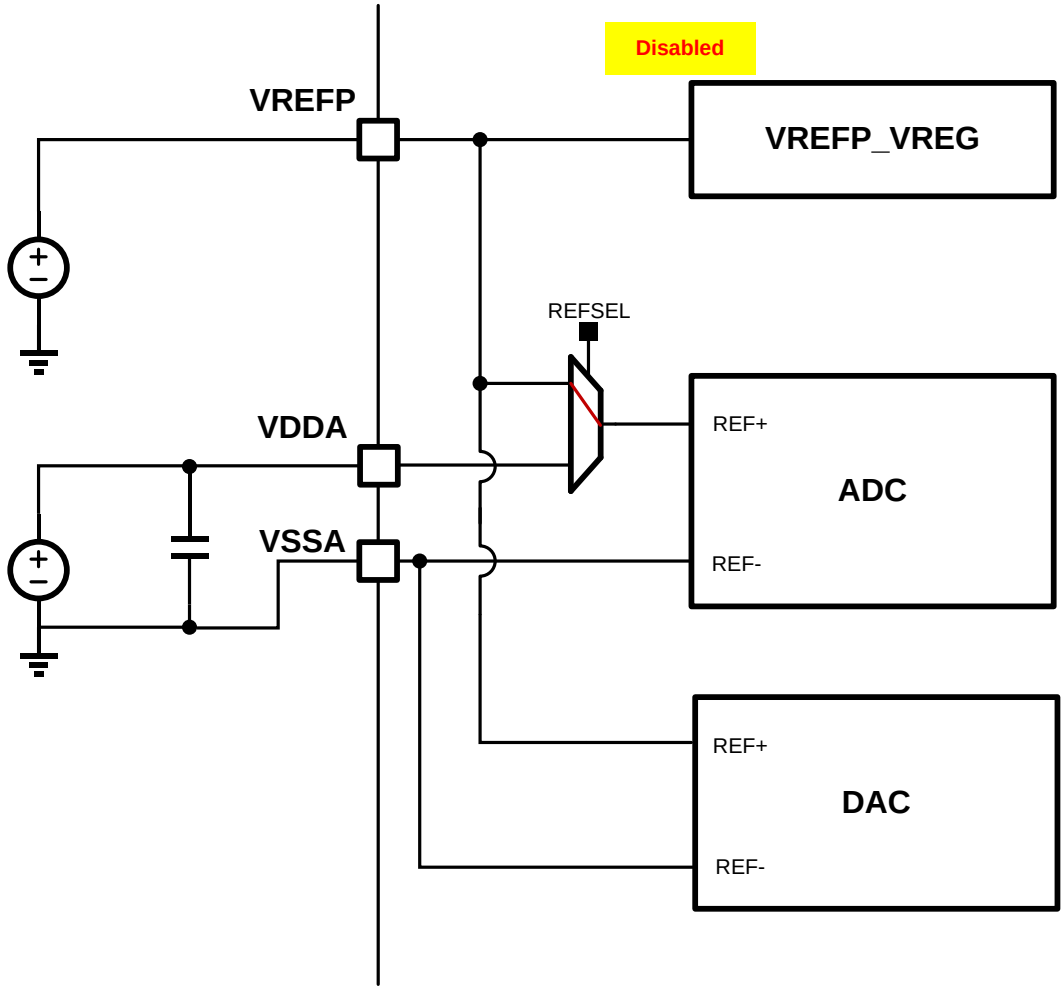


图 7-3 外部独立基准方案

7.2.1.3 内部基准

当使能内部基准时，VREFP_REG表现为一个稳压器，必须在VREFP引脚上外挂一个1uF电容；下图显示了ADC和DAC都使用内部基准源的方案。

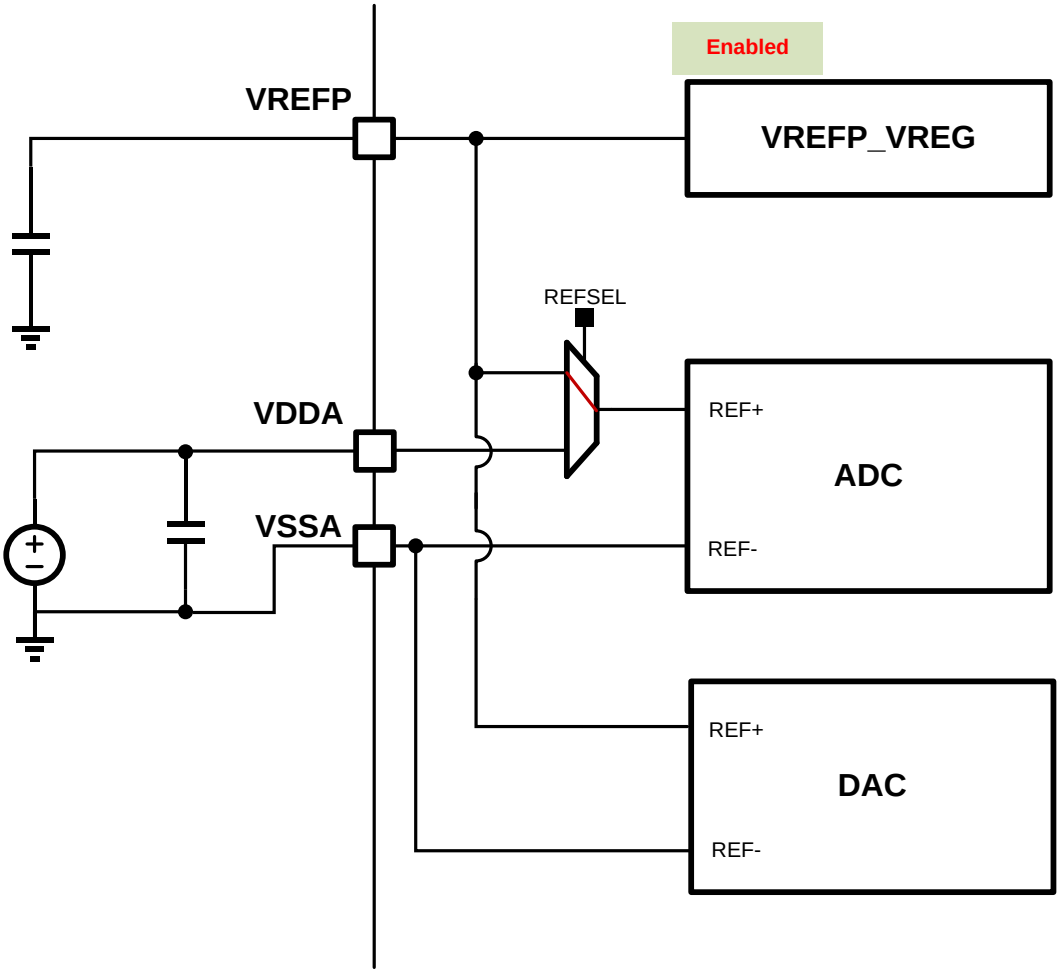


图 7-4ADC 和 DAC 都是用内部基准方案的系统连接

下图中ADC使用VDDA作为基准，而DAC使用内部基准。

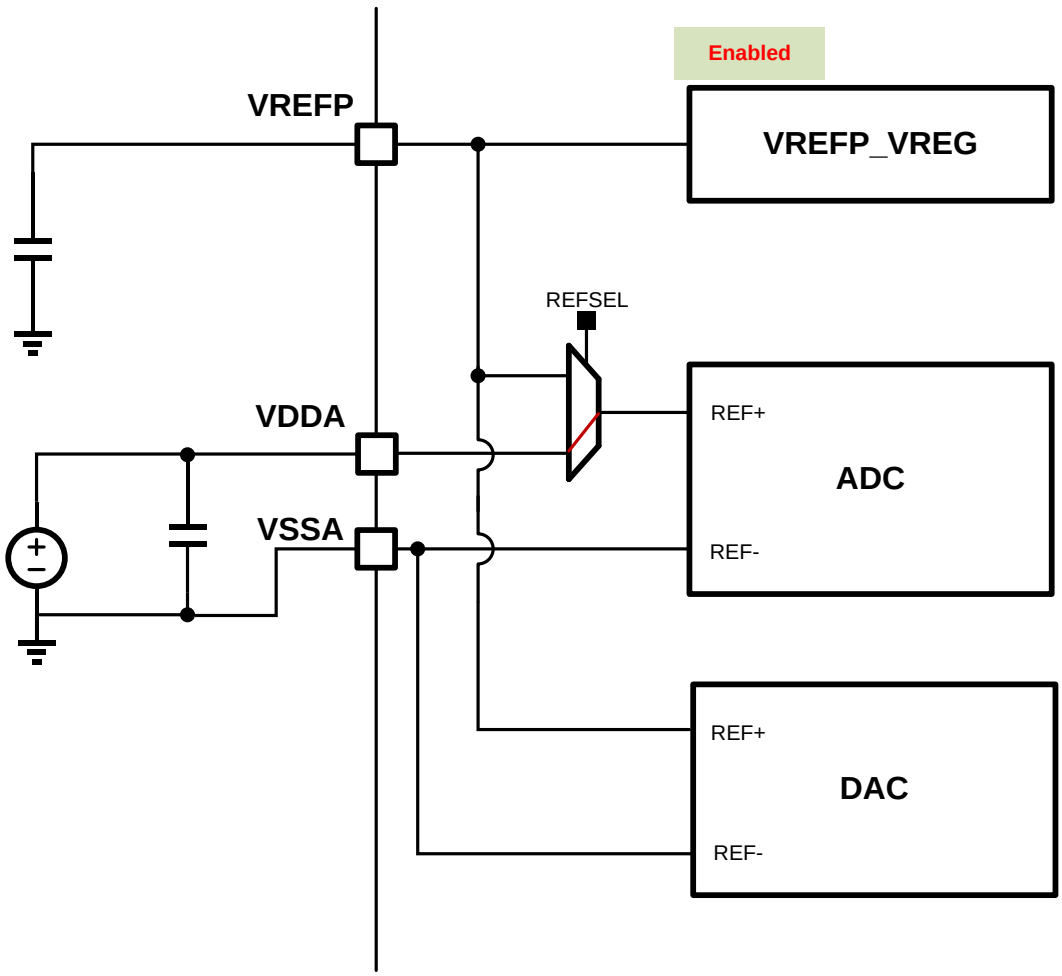


图 7-5 ADC 使用外部基准，DAC 使用内部基准

只有带有独立VREFP引脚封装的芯片才能支持内部基准方案。

注意：本芯片无法支持ADC使用内部基准的同时，DAC使用外部基准。

7.2.2 时钟和复位

VREFP_VREG控制器采用两个工作时钟：APBCLK和LSCLK。

寄存器的总线访问只用APBCLK，而间歇启动控制采用LSCLK进行。

VREFP_VREG模块没有单独的外设复位功能。

7.2.3 参考电压

VREFP_VREG的参考电压来自于VREF1p2，因此使用内部产生的VREFP时必须使能VREF1p2。

7.2.4 输出电压

VREFP_VREG支持5个输出档位，并且支持至少+/-10%的输出调校范围，调校步长为0.1%，即trim

后可以实现+/-0.05%的输出精度。

由于ADC使用VREFP作为基准电压，应用可以通过测量基准源输出的转换值来确认VREFP是否已经建立到正常范围内。

同时VREFP也作为DAC的基准电源，并可作为比较器的参考电压输入。

7.2.5 工作模式

VREFP_VREG支持常使能和间歇使能两种工作模式，通过LPM寄存器控制。

当LPM=0时，VREFP_VREG保持常使能，除非软件通过清零EN寄存器关闭这个模块。注意常使能模式下软件不能关闭VREF1p2.

当LPM=1时，VREFP_VREG工作在间歇使能模式下，即定时间歇性开启，驱动一段可编程时间后再自动关闭。定时开启的周期可以软件编程，以平衡功耗和VREFP纹波幅度。

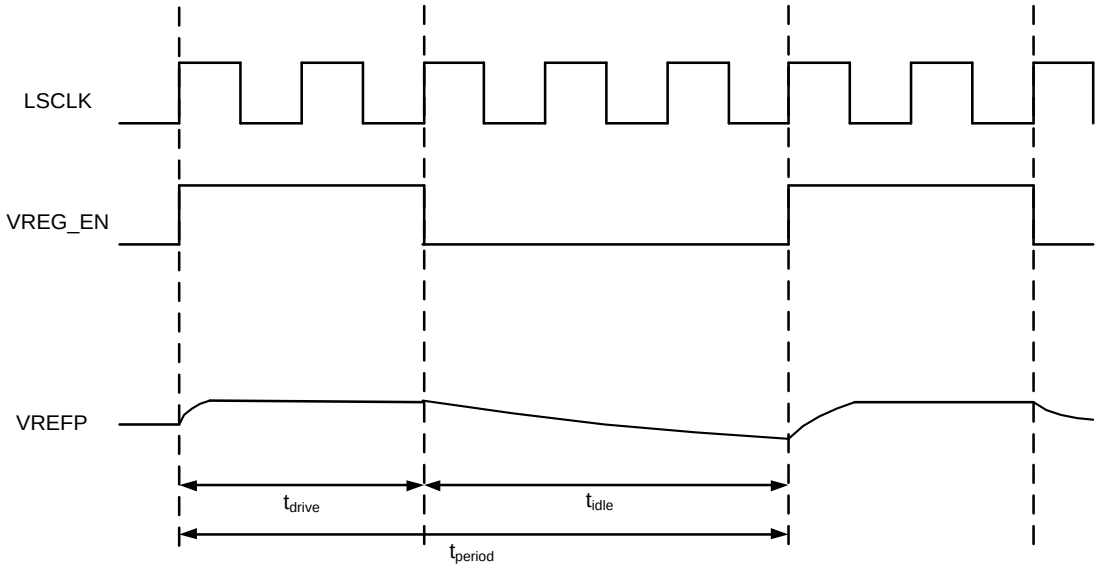


图 7-6 VREFP 间歇使能波形示意图

间歇使能模式在芯片休眠时可以以极低的功耗维持VREFP电压。如果软件关闭了VREF1p2，则控制电路在每次间歇使能开始时先自动使能VREF1p2，等待其建立完成标志，然后再使能VREFP_VREG开始输出驱动。这种情况下，TDRV寄存器配置的驱动时间不包含VREF1p2的建立时间。如果VREF1p2没有关闭，则间歇使能时直接开启VREFP_VREG。

注意VREF1p2关闭的情况下，间歇启动周期配置必须大于VREF1p2建立时间（1.5ms）以及驱动时间之和。

$$t_{period} > t_{VREF_setup} + t_{drive}$$

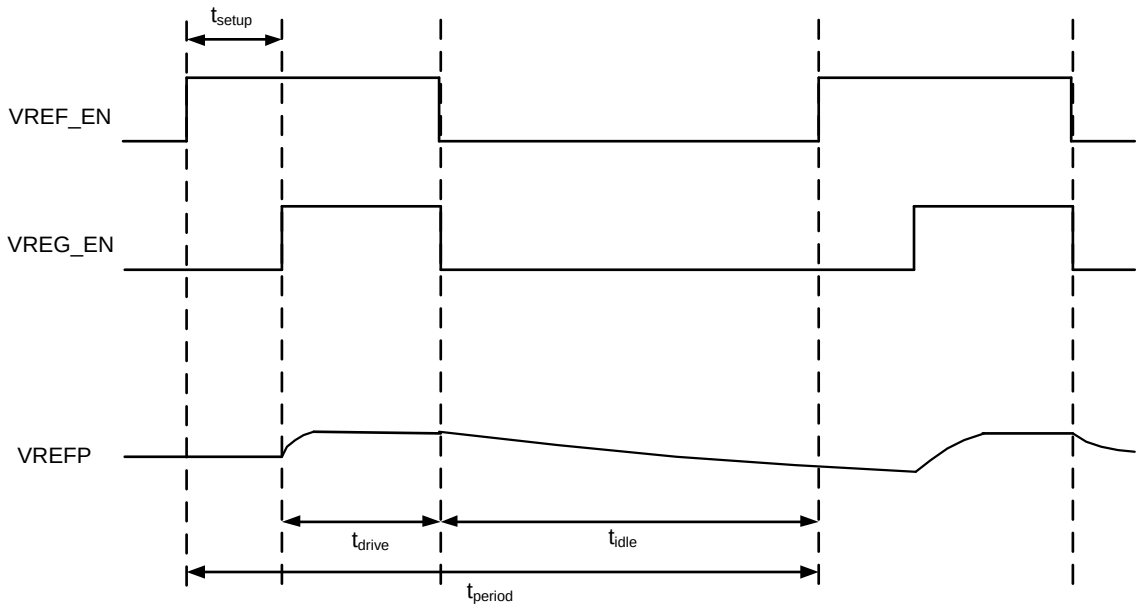


图 7-7 VREF1p2 关闭情况下的间歇使能波形示意图

7.2.6 中断和标志

VREFP_VREG在使能时输出BUSY标志，间歇使能状态下，软件可以通过查询BUSY寄存器确认当前VREFP_VREG是否启动。

间歇使能模式下，VREFP_VREG还会产生两种中断标志；当一次启动完成后，产生DEND标志，当一个间歇周期结束后，产生POV标志。

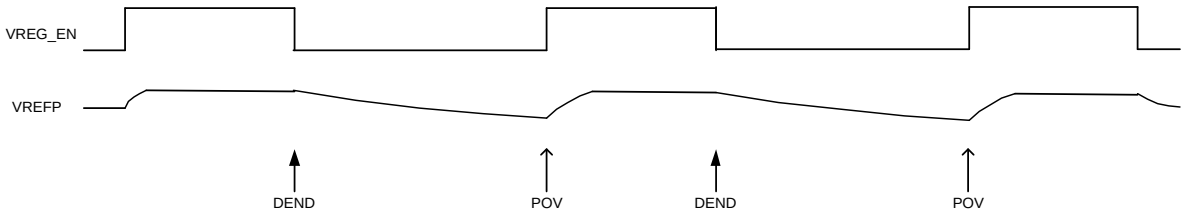


图 7-8 中断标志示意图

7.3 寄存器

参见PMU章节寄存器。

8 备份电源域（VAO domain）

8.1 概述

VAO 电源域采用主备份电源切换后供电。当 VDD 电源跌落导致 PDR 下电复位发生时，芯片自动将 VAO 电源切换到 VBAT；当 VDD 重新上电，恢复到 PDR 上电复位阈值以上时，芯片自动将 VAO 电源切换到 VDD。

VAO 电源域下包含以下电路：

- 电源切换开关
- XTLF 和 LFDET
- RTC_B
- 备份寄存器
- 32768 晶振引脚，PH15 为 GPIO+Tamper
- 上电复位电路
- 模拟测试 Buffer

8.2 结构框图

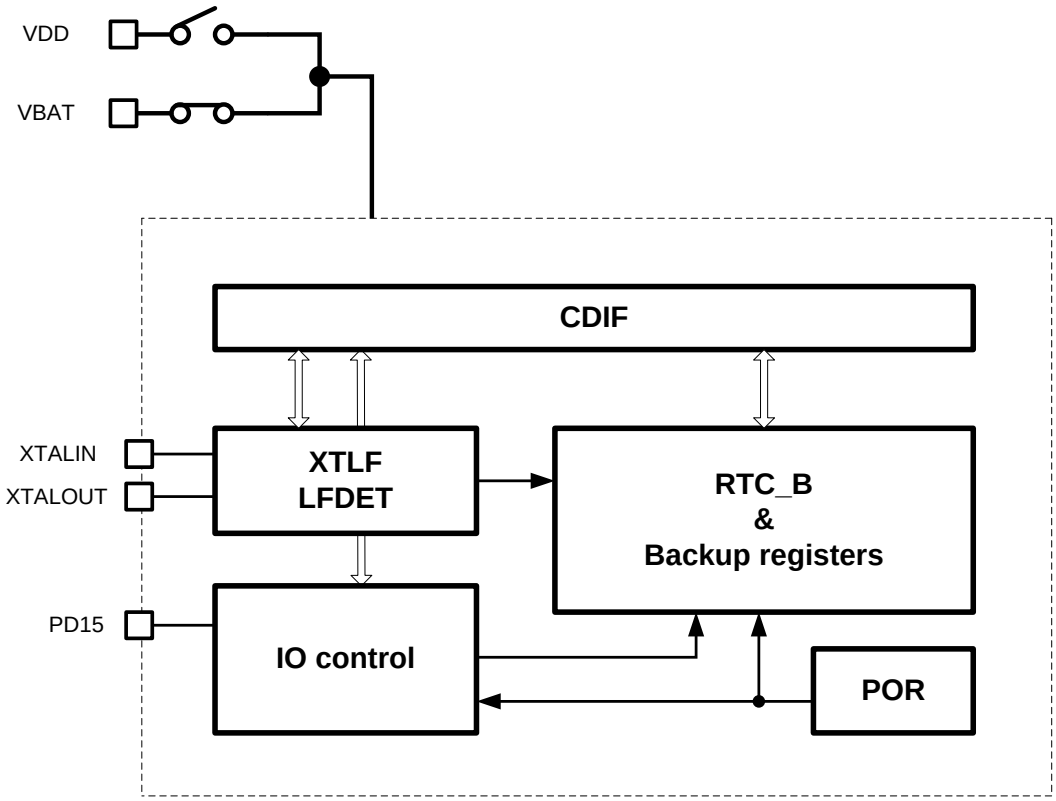


图 8-1 备份电源域结构框图

8.3 电源切换

Power Switch 电路使用 PDR 产生的下电复位信号进行备份电源切换。

应用电源切换的方案中必须保证 PDR 下电阈值高于 XTLF 的最低工作电压。

档位	典型阈值	min	max
POR	1.6		
PDR	1.5	1.4	1.55

8.4 复位

VAO 电源下的上电复位电路，用于在 VDD 或者 VBAT 上电时为 VAO 下的控制寄存器提供复位，确保 PH15 引脚在上电后具有确定的初始状态，XTLF 和部分 RTC_B 控制寄存器有确定的初始值。

在 VAO 电源建立后，上电复位电路自动关闭以节省功耗。此时软件仍可以通过 CDIF 操作 VBAT_RST 寄存器来手工复位。

VAO 电源域有以下两个复位方式：

- 通过置位 VBAT_RST 寄存器实现软件复位
- 在 VDD 和 VBAT 都掉电的情况下，VDD 或者 VBAT 上电都会触发 VAO 上电复位

VAO 上下电复位示意图如下。

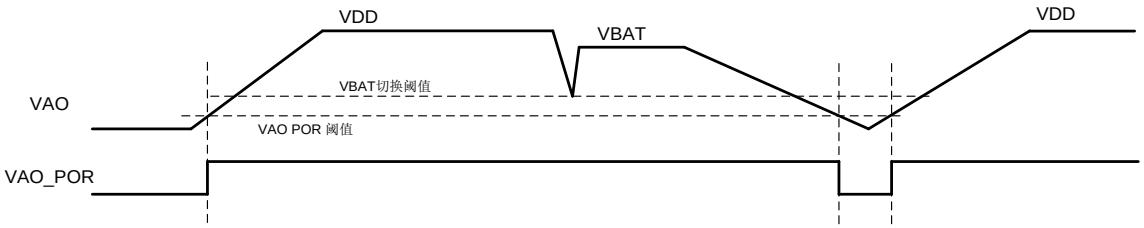


图 8-2 VAO 上下电复位示意图

在 VAO 上电复位后，XTLF 默认打开，RTC_B 上电后默认使能，PH15 控制寄存器默认为高阻态，关闭输入输出。

8.5 低频晶体振荡电路 (XTLF)

8.5.1 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，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。

8.5.2 工作方式

XTLF上电后默认使能，软件可以关闭，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。

8.5.3 停振检测

FM33LG0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RCLP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。

8.6 实时时钟 (RTC_B)

详细参见 36 实时时钟（RTCB）

8.7 IO

VAO 电源下的 GPIO 只有 PH15。

VAO 下还有两个 XTLF 振荡引脚，外接 32768hz 晶体；不能配置为 GPIO 使用。

PH15 主要是 RTCOUT 和 Tamper 功能引脚。

以上引脚供电在电源切换开关之后，因此驱动能力有限，最大 source 电流不能超过 1mA。

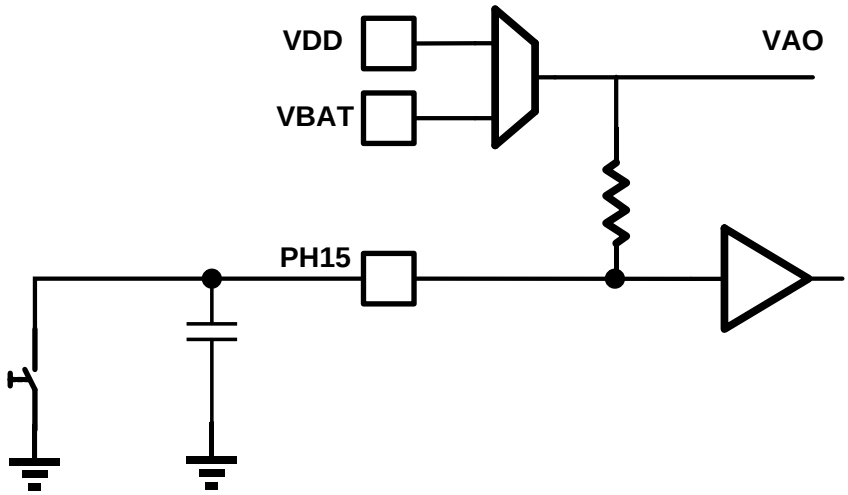
控制寄存器都位于 VAO 电源下，CPU 需要通过 CDIF 接口进行访问。

PH15 是一个内建上拉电阻的 5V tolerant IO，由切换后的 VAO 电源供电。当芯片由 VDD 供电时，PH15 同样由 VDD 供电，因此其推挽输出的高电平是 VDD；当 VDD 掉电时芯片备份电源域由 VBAT 供电，此时 PH15 也是由 VBAT 供电，推挽输出高电平是 VBAT。

8.7.1 使用 PH15 输入实现 tamper 检测

在 VDD 供电和 VDD 掉电的场景下，都需要实现 tamper 检测功能。

下图是使用 PH15 做 tamper 检测的一个示例，通过 PAD 内部上拉电阻上拉到 VAO，当 tamper 事件发生时会输入一个逻辑 0。



在 VDD=5V 的系统中，不建议将 PH15 通过外部上拉连接到 VBAT，否则由于 VBAT 低于 VDD，在芯片由 VDD 供电时可能会导致 PAD 漏电。因此此方案中只能通过 PH15 内部上拉电阻实现上拉。而在 VDD=3V 的系统中，如果 VDD 与 VBAT 相接近，则可以外部上拉到 VBAT。

PH15 为 5V tolerant 引脚，可以接受外部输入电平高于 VAO。因此当 VBAT 高于 VDD 时，可以将 PH15 通过外部上拉电阻上拉到 VBAT。

8.8 寄存器

offset 地址	名称	符号
VAO(模块起始地址：0x4001F000)		
0x800	VAO 复位控制寄存器 (VBAT Reset Control Register)	VAO_RSTCR
0x804	XTLF 控制寄存器 (XTLF Control Register)	VAO_XTLFCR
0x808	XTLF 振荡强度寄存器 (XTLF Power Register)	VAO_XTLFPR
0x80C	低频停振检测中断使能寄存器 (XTLF Oscillation Fail Detection Interrupt Enable Register)	VAO_FDIER
0x810	低频停振检测中断标志寄存器 (XTLF Oscillation Fail Detection Interrupt Status Register)	VAO_FDISR
0xC00	VAO IO 输入使能寄存器 (VAO IO Input Enable Register)	VAO_INEN
0xC04	VAO IO 上拉使能寄存器 (VAO IO Pull-up Enable Register)	VAO_PUEN
0xC08	VAO IO 开漏使能寄存器 (VAO IO Open Drain Enable Register)	VAO_ODEN
0xC0C	VAO IO 功能选择寄存器 (VAO IO Function Control Register)	VAO_FCR
0xC10	VAO IO 输出数据寄存器 (VAO IO Data Output Register)	VAO_DOR
0xC14	VAO IO 输入数据寄存器 (VAO IO Data Input Register)	VAO_DIR
0xC18	VAO IO 输入低阈值配置寄存器 (VAO IO Voltage Input Low Register)	VAO_VILR

8.8.1 VAO 复位控制寄存器（VAO_RSTCR）

名称	VAO_RSTCR							
offset	0x800							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	-							VBAT_RST
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	VBAT_RST	VBAT 电源域寄存器复位控制 1：复位 VBAT 电源域下的寄存器（包含 RTC_B 内部部分寄存器） 0：撤销复位

8.8.2 XTLF 控制寄存器（VAO_XTLFCR）

名称	VAO_XTLFCR							
offset	0x804							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	-				XTLFEN			
位权限	U-0				R/W-xxxx			

位号	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	XTLFEN	XTLF 使能寄存器，上电默认 XTLF 使能 1010：关闭 XTLF 和 FDET 0101：使能 XTLF 和 FDET 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 物理上只有 1 个 bit

8.8.3 XTLF 振荡强度寄存器（VAO_XTLFPR）

名称	VAO_XTLFPR							
offset	0x808							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	-	DRVCFG			XTLFIPW			
位权限	U-0	R/W-001			R/W-1000			

位号	助记符	功能描述
31:7	--	RFU：未实现，读为 0
6:4	DRVCFG	输出级驱动能力配置 000：无驱动 001：最弱驱动 …… 111：最强驱动
3:0	XTLFIPW	XTLF 工作电流选择，电流越大表示振荡强度越高，上电复位后使用 000 档位起振，正常工作时推荐使用 100 或 011 档位，实际应根据适配晶体的实测负阻特性选择合适的电流大小 0000 : 850nA 0001 : 800nA 0010 : 750nA 0011 : 700nA 0100 : 650nA 0101 : 600nA 0110 : 550nA 0111 : 500nA 1000 : 450nA (default) 1001 : 400nA 1010 : 350nA 1011 : 300nA 1100 : 250nA 1101 : 200nA 1110 : 150nA 1111 : 100nA

8.8.4 停振检测中断使能寄存器（VAO_FDIER）

名称	VAO_FDIER							
Offset	0x80C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	-							LFDET_IE
位权限	U-0							R/W-0

位号	助记符	功能描述

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	LFDET_IE	XTLF 低频检测报警中断使能，1 有效 (XTLF fail detect interrupt enable)

8.8.5 停振检测中断标志寄存器（VAO_FDISR）

名称	VAO_FDISR							
Offset	0x810							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	-						LFDETO	LFDETIF
位权限	U-0						R-0	R/W-0

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1	LFDETO	低频晶体停振检测模块输出 (XTLF fail detect output) 1：XTLF 未停振 0：XTLF 停振
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 LFDETO 不为 0 的情况下才能够清除此寄存器 (XTLF fail detect interrupt flag,write 1 to clear)

8.8.6 VAO IO 输入使能寄存器（VAO_INEN）

名称	VAO_INEN							
offset	0xC00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PHINEN15	-						
位权限	R/W-0	U-0						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							

位权限	U-0

位号	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	PHINEN	PH15 引脚输入使能寄存 0：关闭输入 1：使能输入
14:13	--	RFU：未实现，读为 0
12:0	--	RFU：未实现，读为 0

8.8.7 VAO IO 上拉使能寄存器（VAO_PUEN）

名称	VAO_PUEN							
offset	0xC04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PHPUEN15	-						
位权限	R/W-0	U-0						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							
位权限	U-0							

位号	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	PHPUEN	GPIO PH15 上拉控制 0：关闭上拉 1：使能上拉
14:13	--	RFU：未实现，读为 0
12:0	--	RFU：未实现，读为 0

8.8.8 VAO IO 开漏使能寄存器（VAO_ODEN）

名称	VAO_ODEN							
offset	0xC08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	PHODEN15	-						
位权限	R/W-0	U-0						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							
位权限	U-0							

位号	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	PHODEN	GPIO PH15 开漏输出使能 0：关闭开漏输出 1：使能开漏输出
14:13	--	RFU：未实现，读为 0
12:0	--	RFU：未实现，读为 0

8.8.9 VAO IO 功能选择寄存器（VAO_FCR）

名称	VAO_FCR							
offset	0xC0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	Px15FCR		-					
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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							
位权限	U-0							

位号	助记符	功能描述
31:30	PH15FCR	PH[15]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：RTCOUT 11：RFU
29:0	--	RFU：未实现，读为 0

8.8.10 VAO IO 输出数据寄存器（VAO_DOR）

名称	VAO_DOR							
offset	0xC10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PHDO15	-						
位权限	R/W-0	U-0						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							
位权限	U-0							

位号	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	PHDO	GPIO output data register PH15
14:13	--	RFU：未实现，读为 0
12:0	--	RFU：未实现，读为 0

8.8.11 VAO IO 输入数据寄存器（VAO_DIR）

名称	VAO_DIR							
offset	0xC14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	-						
位权限	R	U-0						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							
位权限	U-0							

位号	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	PHDIN	Portx input data register PH15 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
14:13	--	RFU：未实现，读为 0
12:0	--	RFU：未实现，读为 0

8.8.12 VAO IO 输入低阈值配置寄存器（VAO_VILR）

名称	VAO_VILR							
offset	0xC18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PHVIL15	-						
位权限	R/W-0	U-0						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							
位权限	U-0							

位号	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	PHVIL15	PH15 输入低阈值配置 0：输入低阈值正常 1：输入低阈值降低
14:0	--	RFU：未实现，读为 0

9 跨电源域接口（CDIF）

9.1 概述

CPU 通过 CDIF 接口访问 VAO 电源域下的模块

CDIF 仅支持 word 访问，不支持 DMA 访问。访问 VAO 下的寄存器之前，必须置位 INTF_OEN 和 INTF_IEN 寄存器并配置接口分频寄存器。CDIF 接口访问的最高频率建议不要超过 8Mhz，比如当 APBCLK 为 32Mhz 时，建议配置 PRSC 寄存器实现 4 分频。

寄存器访问结束后，建议清零 INTF_OEN，否则会带来十几 uA 的额外电流。

9.2 总线地址

在 APB 上分配 4KB 地址空间，用于容纳所有 VAO 下的模块寄存器，包括 RTC_B、XTLF 等，参见“备份电源域”章节。

这个地址段不支持 DMA 操作。

9.3 时钟和复位

CDIF 模块使用 APBCLK 工作，操作前需要通过配置 CMU 模块寄存器（CDIF_PCE）打开模块时钟。

CDIF 模块没有独立的 RMU 复位寄存器。

9.4 寄存器（CPU domain）

offset 地址	名称	符号
CDIF(模块起始地址：0x4001E000)		
0x00	接口控制寄存器	CDIF_CR
0x04	接口预分频寄存器	CDIF_PRSC

9.4.1 接口控制寄存器（CDIF_CR）

名称	CDIF_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	-						INTF_IEN	INTF_OEN
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	--	未实现，读为 0
1	INTF_IEN	跨电源域接口 VAO->CPU 方向使能 1：使能接口（CPU 端输入使能） 0：关闭接口（CPU 端输入关闭）
0	INTF_OEN	跨电源域接口 CPU->VAO 方向使能 1：使能接口 0：关闭接口

9.4.2 接口预分频寄存器（CDIF_PRSC）

名称	INTF_PRSC							
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	-					PRSC		
位权限	U-0					R/W-000		

位号	助记符	功能描述
31:4	--	未实现，读为 0
3:0	**PRSC**	跨电源域时序控制，配置相对于 APBCLK 的定时比例 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

10 处理器（CPU）

10.1 概述

FM33LG0 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

10.1.1 处理器配置

Feature	Options	FM33LG0Config
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	Present

表 10-1 FM33LGx0xx CPU 配置简表

10.2 内核寄存器

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 10-2 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。

10.3 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。

10.3.1 中断向量表

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 低功耗模式错误中断	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_A/RTC_B	实时时钟中断	0x0000_0048
19	10	settable	FLASH	NVMIF 中断	0x0000_004C
20	11	settable	FDET	XTLF 或 XTHF 停振检测中断 系统时钟选择错误中断	0x0000_0050
21	12	settable	ADC	ADC 转换完成中断	0x0000_0054
22	13	settable	DAC	DAC 中断	0x0000_0058

Position	Priority	Priority type	Acronym	Description	Address
23	14	settable	SPI0	SPI 中断	0x0000_005C
24	15	settable	SPI1		0x0000_0060
25	16	settable	SPI2		0x0000_0064
26	17	settable	UART0	UART 中断	0x0000_0068
27	18	settable	UART1		0x0000_006C
28	19	settable	UART3		0x0000_0070
29	20	settable	UART4		0x0000_0074
30	21	settable	UART5		0x0000_0078
31	22	settable	U7816	U7816 中断	0x0000_007C
32	23	settable	LPUARTx	LPUART0/1/2 中断	0x0000_0080
33	24	settable	I2C	I2C 中断	0x0000_0084
34	25	settable	CCL	时钟校准中断	0x0000_0088
35	26	settable	AES	AES 中断	0x0000_008C
36	27	settable	LPTIM	LPTIM16 或 LPTIM32 中断	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	LUT	LUT 中断	0x0000_009C
40	31	settable	BSTIM	BSTIM16 或 BSTIM32 中断	0x0000_00A0
41	32	settable	COMPx	COMPx 中断	0x0000_00A4
42	33	settable	GPT0,1	通用定时器 0,1 中断	0x0000_00A8
43	34	settable	GPT2	通用定时器 2 中断	0x0000_00AC
44	35	settable	ATIM	高级定时器中断	0x0000_00B0
45	36	settable	VREF1p2 /VREF_V REG	1.2V 内部基准电压建立中断 VREF_VREG 中断	0x0000_00B4
46	37	settable	EXTI	外部引脚中断	0x0000_00B8
47	38	settable	CAN	CAN2.0 中断	0x0000_00BC

表 10-3 FM33Lx0xx 中断向量表

其中WKUPx中断可以接到NMI（兼容FM385）或者38#入口。通过PADCFG模块的WKUPEN.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。

10.3.2 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。

10.3.3 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

FM33LG0 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。

10.3.4 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。

10.4 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

10.4.1 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

10.4.1.1 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

10.4.1.2 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	0	R/W

	HardFault		
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

10.4.1.3 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

10.4.1.4 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

10.4.1.5 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 访问控制	000	R/W
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

10.5 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》

10.5.1 调试功能引脚

FM33LG0 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。

10.5.2 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。当 TEST_N 引脚为低电平超过 8ms 后，看门狗自动关闭。当调试器不使用 TEST_N 引脚时，软件或 Debugger 可以通过 MCUDBGCR 寄存器配置看门狗打开或关闭。

10.5.3 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)。

11 复位管理单元（RMU）

11.1 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 1.65V
- BOR 下电复位产生电压软件可配置为 1.8/2.0/2.2/2.4V，可关闭。
- 低功耗下电复位电路（PDR），下电复位电压可配置为 1.4/1.45/1.5/1.55V，可关闭
- 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。

11.2 模块框图

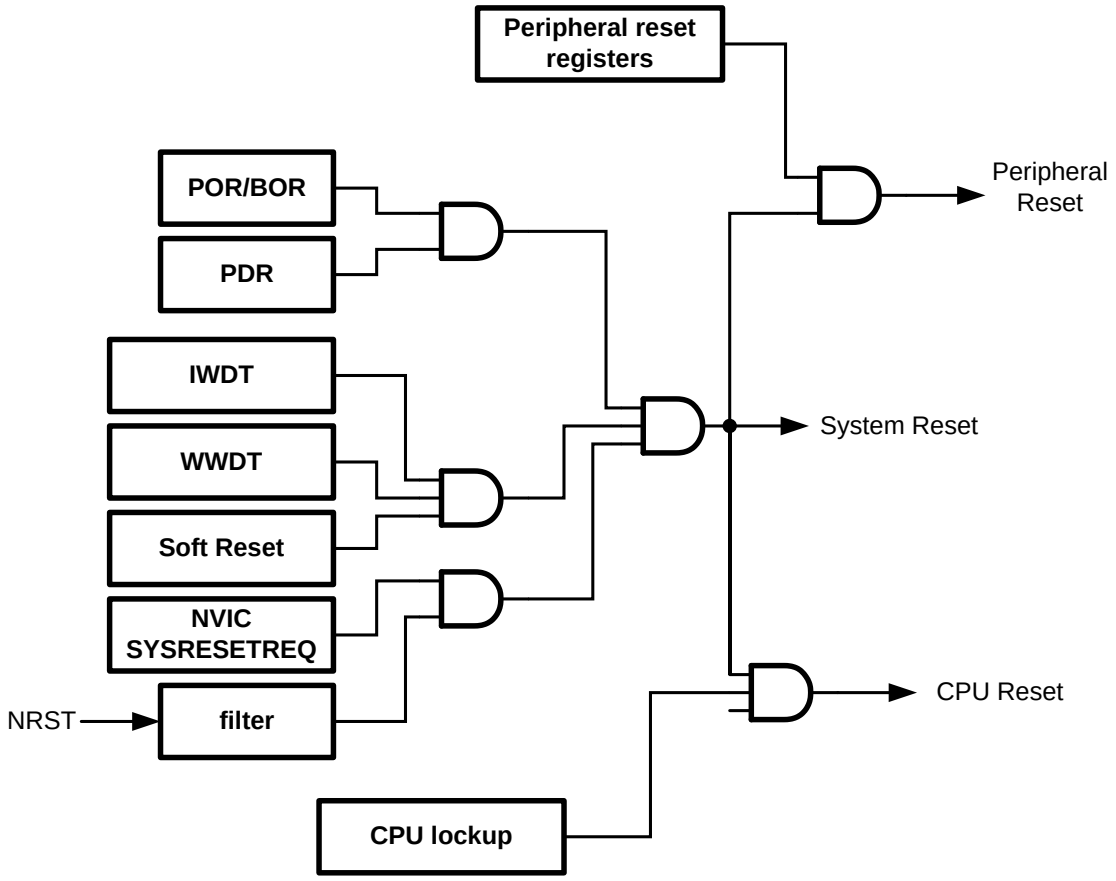


图 11-1 芯片复位源框图

11.3 VDD 上下电复位

上下电复位电路监控 VDD 电源，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。

V_{POR} 阈值固定为 1.6V（典型值）

V_{PDR} 默认阈值设定为 1.5V（典型值）

VBOR 下电复位阈值软件可配置，共 4 档：1.8V、2.0V、2.2V、2.4V

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。

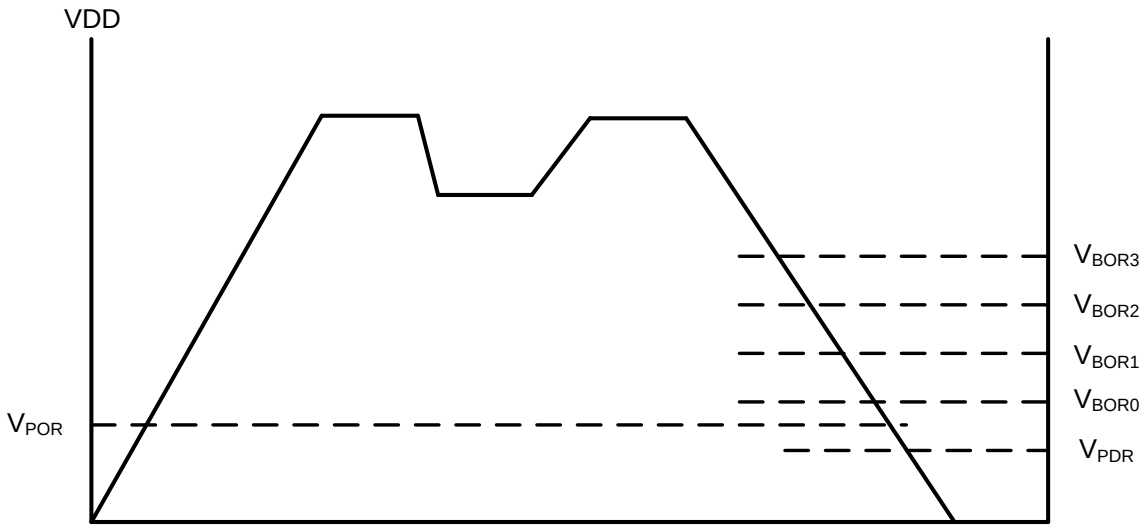


图 11-2 上下电复位示意图

为了保证安全的上电复位，建议上电速度不能快于2us/V，即电源从0上升到5V，用时不能短于10us。

11.4 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。

11.5 NRST 引脚复位

NRST 引脚拉低能够产生全局复位，为了增强抗干扰能力，NRST 引脚有数字滤波功能，为了确保可靠的复位，建议拉低 NRST 并保持低电平大于 1ms。

11.6 寄存器

offset 地址	名称	符号
RMU(模块起始地址：0x40002800)		
0x00	PDR 控制寄存器 （PDR Control Register）	RMU_PDRCR
0x04	BOR 控制寄存器 （BOR Control Register）	RMU_BORCR
0x08	复位配置寄存器 （Lockup reset Control Register）	RMU_LKPCR
0x0C	软件复位寄存器 （Software Reset Register）	RMU_SOFTRST
0x10	复位标志寄存器 （Reset Flag Register）	RMU_RSTFR
0x14	外设复位使能寄存器 （Peripheral Reset Enable Register）	RMU_PRSTEN
0x18	AHB 外设复位寄存器 （AHB Peripherals Reset Register）	RMU_AHBRSTCR
0x1C	APB 外设复位寄存器 1 （APB Peripherals Reset Register1）	RMU_APBRSTCR1
0x20	APB 外设复位寄存器 2 （APB Peripherals Reset Register2）	RMU_APBRSTCR2

11.6.1 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

位号	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

位号	助记符	功能描述
		0：关闭下电复位 1：使能下电复位

11.6.2 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	-				BOR_PDRCFG		-	BOR_ENB
位权限	U-0				RW-00		U-0	R/W-1

位号	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:2	BOR_PDRCFG	下电复位电压配置 00—1.8V 01—2.0V 10—2.2V 11—2.4V
1	--	RFU：未实现，读为 0
0	BOR_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 注意，使能 OFF_BOR_1P2 时必须确保 VREF_EN 置位

11.6.3 复位配置寄存器（RMU_LKPCR）

名称	RMU_LKPCR							
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							

名称	RMU_LKPCR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						EN	-
位权限	U-0						R/W-0	U-0

位号	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	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

11.6.4 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							

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

11.6.5 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	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			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
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		SOFTN_FLAG	IWDTN_FLAG	-	WWDTN_FLAG	LKUPN_FLAG	NVICN_FLAG
位权限	U-0		R/W-0	R/W-0	U-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	**MDF_FLAG**	模式诊断超时复位标志，高有效，软件写 1 清零
11	**NRSTN_FLAG**	RST_TST_N 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（POR 和 BOR），高有效，软件写 1 清零
8	**PDRN_FLAG**	下电复位标志（PDR），高有效，软件写 1 清零
7:6	--	RFU：未实现，读为 0
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

11.6.6 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							

位号	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

11.6.7 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

位号	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

11.6.8 APB1 外设复位寄存器（RMU_APBRSTCR1）

名称	RMU_APBRSTCR1							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	UART5RST	UART4RST	UART3RST	-	UART1RST	UART0RST	UARTIRRST	U7816RST
位权限	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	GPT2RST	GPT1RST	GPT0RST	ATIMRST	BT32RST	BT16RST	-	-
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SPI2RST	SPI1RST	SPI0RST	-	I2CRST	LPUART2RST	LPUART1RST	LPUART0RST
位权限	R/W-0	R/W-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	-	VREFRST	PGLRST	LCDRST	DACRST	OPARST	LPT16RST	LPT32RST
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	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

位号	助记符	功能描述
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	--	RFU：未实现，读为 0
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	U7816RST	U7816 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
23	GPT2RST	GPTIM2 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
22	GPT1RST	GPTIM1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
21	GPT0RST	GPTIM0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
20	ATIMRST	ATIM 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
19	BT32RST	BSTIM32 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
18	BT16RST	BSTIM16 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
17:16	--	RFU：未实现，读为 0
15	SPI2RST	SPI2 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
14	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
13	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
12	--	RFU：未实现，读为 0
11	I2CRST	I2C 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
10	LPUART2RST	LPUART2 模块复位，软件写 1 复位，写 0 撤销复位

位号	助记符	功能描述
		0：不复位 1：复位
9	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
8	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
7	--	RFU：未实现，读为 0
6	VREFRST	VREF1p2 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	LCDRST	LCD 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
3	DACRST	DAC 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
2	OPARST	OPA 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	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

11.6.9 APB2 外设复位寄存器（RMU_APBRSTCR2）

名称	RMU_APBRSTCR2							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							ADCCRST
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ADCRST	-				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	-				DIVASRST	CANRST	SVDRST	COMPRST

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
31:25	--	RFU：未实现，读为 0
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
22:19	--	RFU：未实现，读为 0
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	DIVASRST	DIVAS 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
2	CANRST	CAN 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
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

12 独立看门狗（IWDT）

12.1 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。独立看门狗在芯片上电后由软件启动，启动后无法关闭，直到芯片发生复位。

为了便于调试，在以下情况下 IWDT 会停止运行：

- 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。

IWDT 使用 LSCLK 工作，结合停振检测电路，可以确保在 XTLF 低频晶振停振时也不会停止运行。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。

12.2 结构框图

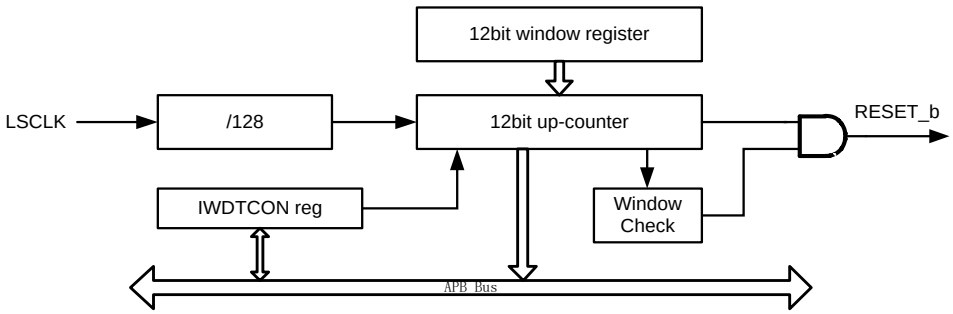


图 12-1 IWDT 结构框图

12.3 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 LSCLK 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：

$tWWDT = T_{LSCLK} * 128 * IWDTOVP$

LSCLK 频率	溢出长度配置	溢出时间（ms）
32768Hz	32	125
	64	250
	128	500
	256	1000
	512	2000
	1024	4000
	2048	8000
	4096	16000

表 12-1 IWDT 溢出周期表

12.4 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。

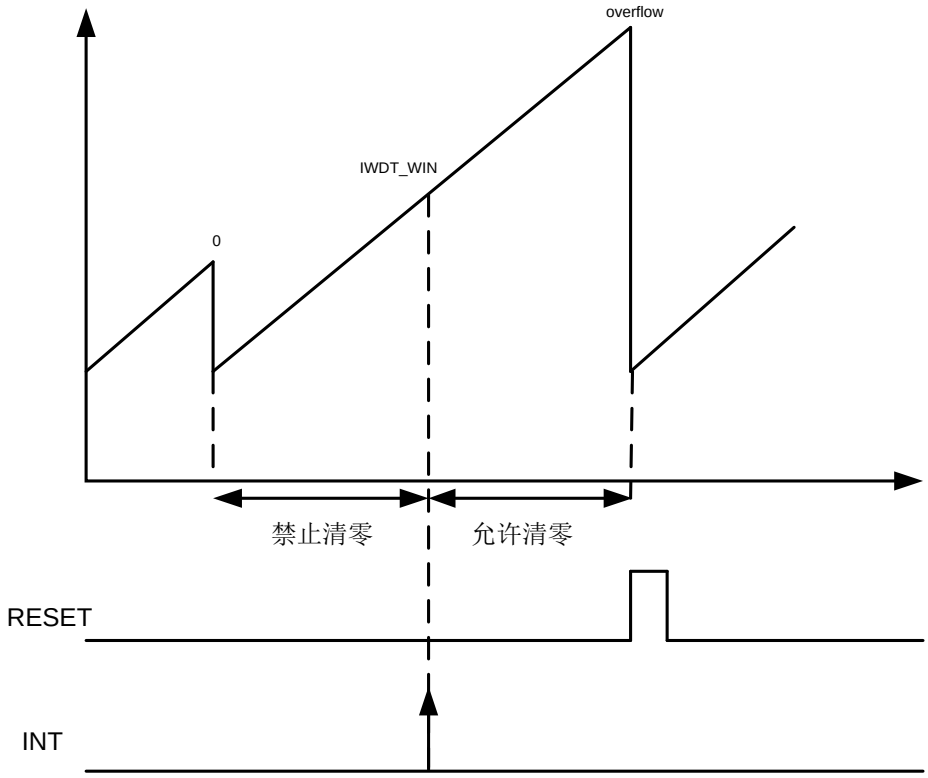


图 12-2 IWDT 窗口示意图

12.5 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）。

12.6 寄存器

offset 地址	名称	符号
IWDT(模块起始地址：0x40011400)		
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

12.6.1 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**	上电复位后 IWDT 默认关闭，软件向此寄存器写入 0x1234_5A5A 后启动 IWDT，此后 IWDT 不可关闭直到下一次芯片复位。 IWDT 启动后，软件向此地址写入 0x1234_5A5A 时清狗 (IWDT Service Register,write only)

12.6.2 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

12.6.3 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

12.6.4 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)

12.6.5 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

12.6.6 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

13 窗口看门狗（WWDT）

13.1 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。

13.2 结构框图

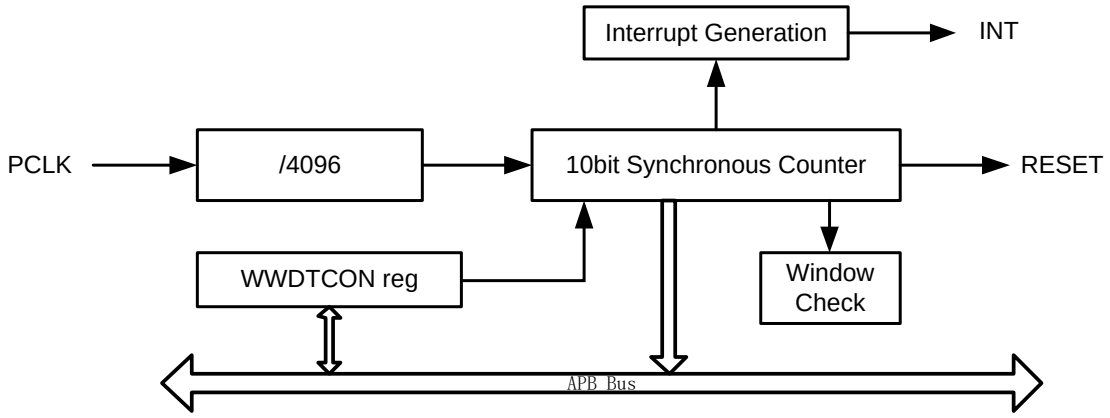


图 13-1WWDT 结构框图

13.3 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

APBCLK 频率	溢出长度配置	溢出时间（ms）
	1024	524.288

表 13-1 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。

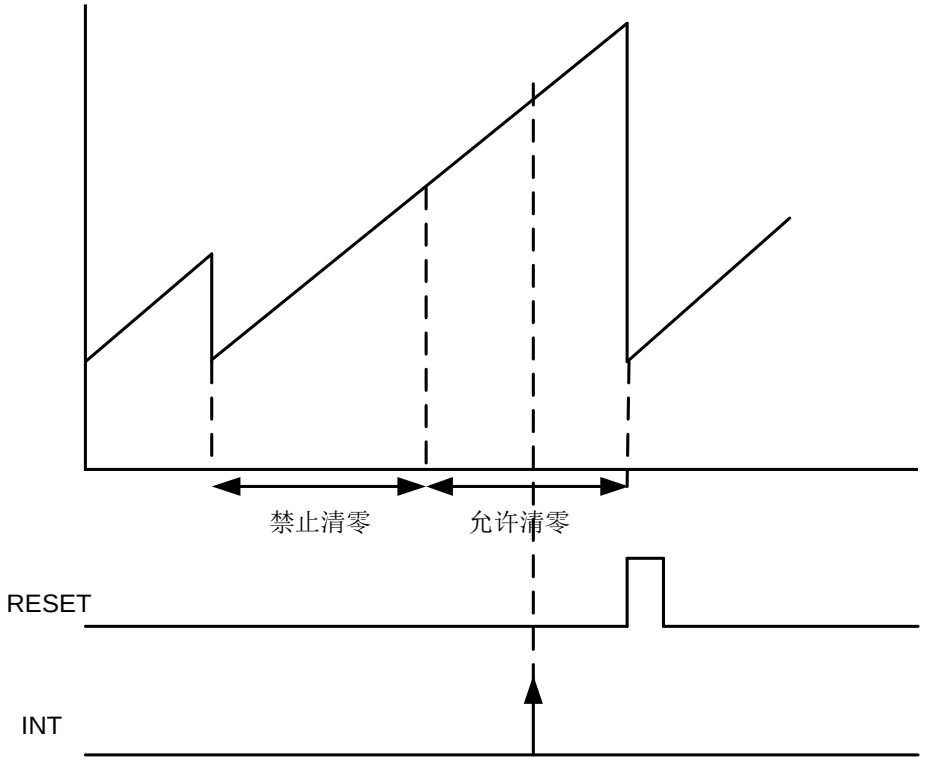


图 13-2WWDT 窗口示意图

13.4 寄存器

offset 地址	名称	符号
WWDT(模块起始地址：0x40011800)		
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

13.4.1 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

13.4.2 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

13.4.3 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)

13.4.4 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

13.4.5 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

13.4.6 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)

14 时钟管理单元（CMU）

14.1 概述

芯片内包含32.768KHz低频晶体振荡电路(XTLF)、4~24MHz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)、614KHz低功耗环振(RCLF)、高频晶体振荡器(XTHF)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

14.2 时钟架构

14.2.1 时钟树

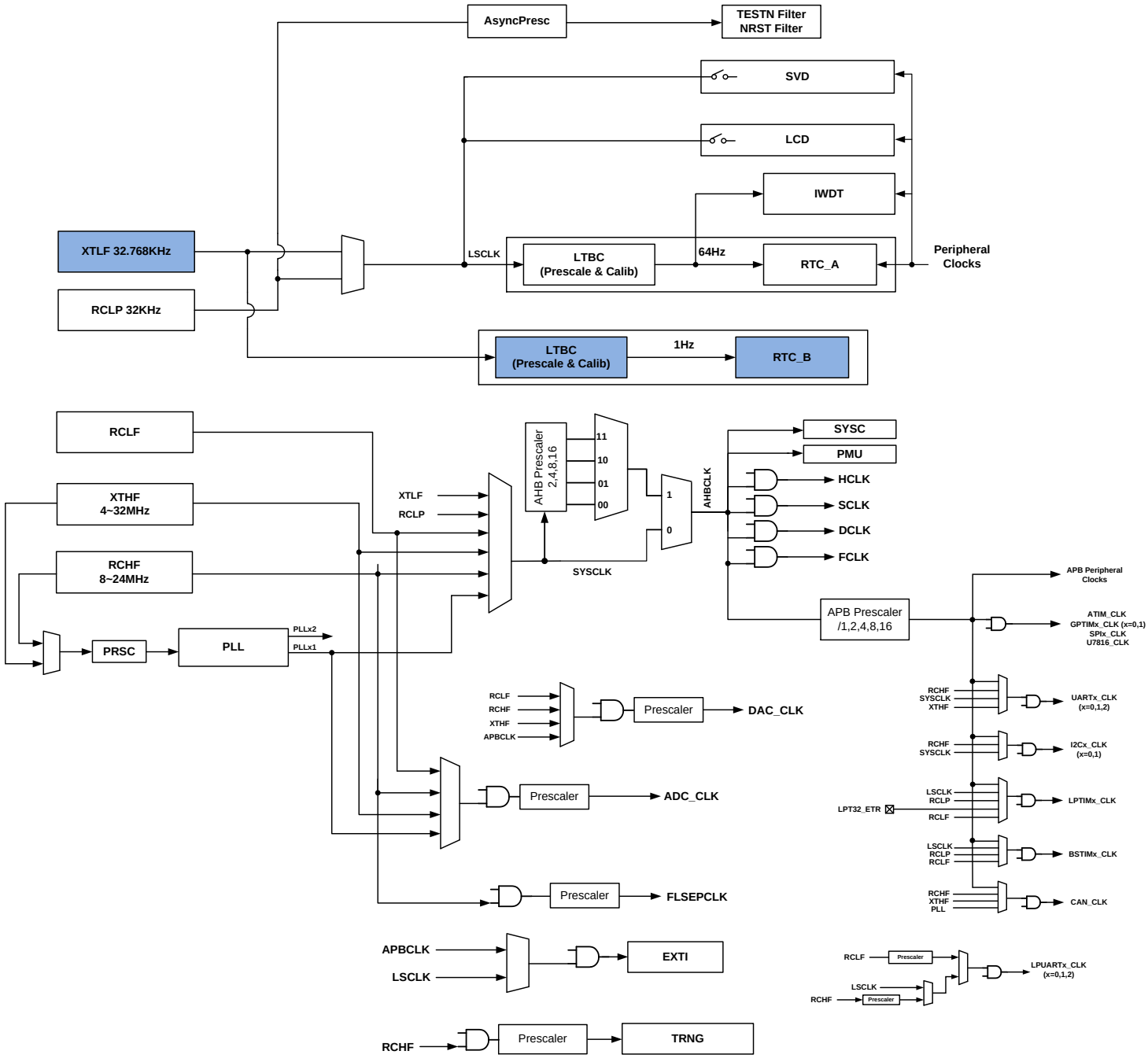


图 14-1 芯片时钟框图

系统主时钟(SYSCLK)可由XTLF、RCHF、XTHF、PLL、RCLP、RCLF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。

14.2.2 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
RCLF	RCLF 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	RCLP 使能

表 14-1 系统时钟切换控制

14.2.3 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。

- 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。

14.2.4 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

时钟	源头	说明
		到 RCLP 主要用于 RTC、IWDT、SVD、LCD
SYSCLK	RCHF, PLL, RCLP, XTLF, XTHF, RCLF	32K~64MHz，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

表 14-2 主要时钟说明

14.2.5 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)	APBCLK	APBCLK
		RCHF
		SYSCLK
LPUARTx (x=0,1,2)	APBCLK	LSCLK
		RCHF
		RC1M
I2Cx (x=0,1)	APBCLK	APBCLK
		RCHF
		SYSCLK
ATIM	APBCLK	APBCLK
		PLLx2_CLK
LPTIM32 LPTIM16	APBCLK	APBCLK
		LSCLK
		RCLP
		LPT32_ETR, LPT16_ETR

模块	总线时钟	工作时钟
		RC1M
BSTIM32 BSTIM16	APBCLK	APBCLK
		LSCLK
		RCLP
		RC1M
ADC	APBCLK	RC1M
		XTHF
		RCHF
		PLL
NVMIF (Flash erase/program)	AHBCLK	RCHF
EXTI (PADCFG)	AHBCLK	AHBCLK
		LSCLK
TRNG	APBCLK	RCHF
IWDT	APBCLK	LSCLK
LCD	APBCLK	LSCLK
RTC	APBCLK	LSCLK
非独立工作时钟外设		
PMU	AHBCLK	
DMA	AHBCLK	
GPTIMx (x=0,1,2)	APBCLK	
UARTy (y=3,4,5)	APBCLK	
SPIx (x=0,1,2)	APBCLK	
7816	APBCLK	
AES	APBCLK	
CRC	APBCLK	
WWDT	APBCLK	
OPAx	APBCLK	
COMPx (x=0,1,2)	APBCLK	

表 14-3 外设时钟说明

注意，外设工作前，需要先使能总线时钟和工作时钟。在使能外设模块工作时钟时，应先配置时钟源和预分频寄存器，最后再通过CMU_PCER寄存器使能工作时钟输出。

14.2.6 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。

14.2.7 LSCLK 切换逻辑

LSCLK是供RTC、IWDT、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。

当LSCLK用作系统时钟（SYSCLK）时，建议软件打开停振自动切换功能。

14.3 高频 RC 振荡器(RCHF)

14.3.1 概述

高频RC振荡器典型振荡频率为8/16/24MHz，可用作系统主时钟，当系统需要更高主频时，可以使用PLL将RCHF倍频到最高64Mhz。RCHF输出频率可以进行调校，出厂前调校到目标频率的+/-0.5%以内，8MHz输出全温区（-40~+85°C）频率变化小于+/-1%，16MHz输出全温区（-40~+85°C）频率变化小于+/-2%。

14.3.2 软件使用指南

芯片上电后默认使用RCHF 8MHz时钟工作，硬件电路会自动从LDT0读取8MHz校准值，保证8MHz频率误差小于+/-0.5%。

如果软件需要使用其他频率，则按照以下步骤操作：

- 改写CMU_RCHFCR.FSEL
- 从LDT0读取对应频率的调校值（常温校准值）
- 将频率调校值写入CMU_RCHTR寄存器，即可得到常温下目标频率误差小于+/-0.5%的时钟

如果希望在高低温环境下获得精确的环振时钟，可以使用定时器和XTLF时钟，对RCHF进行实时校准。在没有32768Hz晶体的情况下，则可以通过温度传感器结合环振高低温频率经验曲线进行大致的调校，以改善高低温下的频率偏移。详情请联系上海复旦微电子公司获取技术支持。

14.4 低频 RC 振荡器(RCLF)

14.4.1 概述

RCLF是一个低功耗中频环振，典型频率614.4KHz，用于CPU低功耗低速运行，以及为LPUART提供工作时钟，以实现准确的9600波特率收发。

14.5 低功耗 RC 振荡器(RCLP)

14.5.1 概述

RCLP是一个极地功耗的低频环振，典型频率32Khz，典型功耗仅300nA。此环振可用于XTLF的备份时钟，也可用于CPU低速运行和部分外设工作。

14.5.2 软件使用指南

RCLP功耗极低，因此在ACTIVE和LPRUN模式下默认开启。

如果使能了XTLF停振自动切换备份时钟的功能，当XTLF停振时RCLP被强制开启。

当软件关闭XTLF时，LFDET不会输出停振信号，如果此时软件也关闭了RCLP，则IWDT和RTC都停止运行；如果软件希望在休眠模式下保持RCLP运行，则需要保证在休眠前使能RCLP。

功耗模式	RCLP 控制状态说明
Active/LPRun	默认使能，软件可以控制使能或者关闭
Sleep/DeepSleep	休眠时状态由使能寄存器决定 如果 RCLP 关闭，则 XTLF 停振时根据寄存器配置决定是否自动启动 RCLP 并输出 32KHz

表 14-4RCLP 状态说明

14.6 高频晶体振荡电路(XTHF)

14.6.1 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。晶体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4~24MHz晶体。软件可以通过XTHFEN寄存器使能或关闭XTHF时钟。

14.6.2 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PC2和PC3引脚配置为模拟功能。

14.6.3 停振检测（HFDET）

FM33LG0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死机。

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。

14.7 锁相环(PLL)

14.7.1 概述

锁相环输入参考时钟可以是RCHF或XTHF的分频，需要根据RCHF和XTHF频率，配置预分频寄存器，得到1MHz的输入参考时钟，最高输出频率可达64MHz（占空比50%）以及其2倍频。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。

14.7.2 软件应用指南

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

配置PLL输出64MHz，并使系统以64MHz主频运行：

- 配置PLLCON寄存器，选择输入时钟源和输出时钟频率
- 设置Flash wait cycle为2
- 使能PREFETCH（可选）
- 将AHB时钟选择为PLL输出

14.8 时钟校准

时钟校准电路是一个同步计数器，用于不同时钟之间互相校准，而不需要占用定时器资源。

时钟校准电路的核心是16位同步计数器，预分频器和输入信号源选择。如下图。

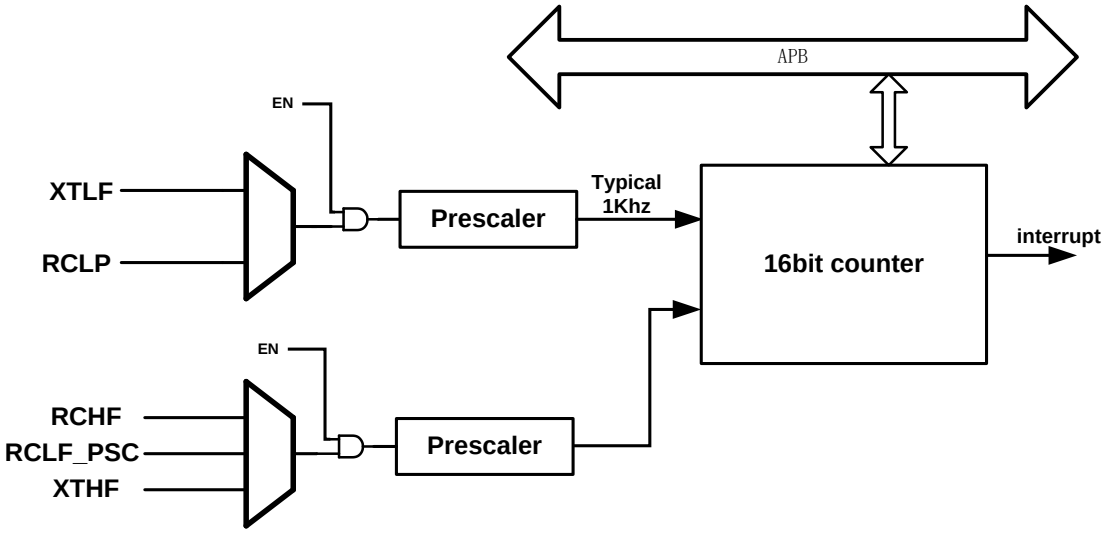


图 14-2 时钟校准电路框图

时钟校准的主要基准是XTLF或RCLP，经过预分频后得到周期性参考信号，典型频率1Khz左右。参考信号上升沿触发计数器计数，并在下一个上升沿停止计数器，计数器中的计数值反映了参考信号和被测时钟之间的频率关系。

比如，参考信号选择为32768Hz的32分频，即1024Hz；被测时钟为RCHF 8Mhz，如果频率为准确的8Mhz，则参考信号的一个周期内定时器计数值应该在7812附近，如果计数值偏离较多，则可以根据计数值计算频率误差并校准RCHF输出。

如果被测信号频率较高，比如RCHF 32Mhz，在校准前应对被测时钟进行预分频，确保计数值不会溢出。

通常情况下使用XTLF来对RCHF和RCLF进行校准，在RCHF完成校准的情况下，可以反过来对RCLP进行校准，使用XTHF也可以实现对RCLP的校准。

14.9 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	VBAT	说明
RCHF	X	硬件强制关闭	X	掉电
PLL	X		X	
XTHF	X		X	
RCLF	O	软件配置使能或关闭	X	
RCLP	O		X	
XTLF	O		O	软件配置使能或关闭

表 14-5 低功耗模式下的时钟源状态说明

14.10 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。

14.11 寄存器

offset 地址	名称	符号
CMU(模块起始地址：0x40002400)		
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_RCHCR
0x08	RCHF 调校寄存器（RCHF Trim Register）	CMU_RCHTR
0x0C	PLL 控制寄存器（PLL Control Register）	CMU_PLLCR
0x10	RCLP 控制寄存器（RCLP Control Register）	CMU_RCLPCR
0x14	RCLP 调校寄存器（RCLP Trim Register）	CMU_RCLPTR
0x1C	LSCLK 选择寄存器（LSCLK Select Register）	CMU_LSCLKSEL
0x20	XTHF 控制寄存器（XTHF Control Register）	CMU_XTHCR
0x24	RCLF 控制寄存器（RCLF Control Register）	CMU_RCLFCR
0x28	RCLF 调校寄存器（RCLF Trim Register）	CMU_RCLFTR
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_PCLKCR3
0x40	外设总线时钟控制寄存器 4（Peripheral bus Clock Control Register4）	CMU_PCLKCR4
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
0x54	时钟校准控制寄存器（Clock Calibration Control Register）	CMU_CCCR

offset 地址	名称	符号
0x58	时钟校准配置寄存器（Clock Calibration Config Register）	CMU_CCFR
0x5C	时钟校准计数值寄存器（Clock Calibration Counter Register）	CMU_CCNR
0x60	时钟校准中断标志寄存器（Clock Calibration Interrupt Status Register）	CMU_CCISR

14.11.1 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	-					APBPRES		
位权限	U-0					R/W-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFUI		-			AHBPRES		
位权限	R/W-00		U-0			R/W-011		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	STCLKSEL		-			SYSCLKSEL		
位权限	R/W-00		U-0			R/W-000		

位号	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）
24:19	--	RFU：未实现，读为 0
18:16	APBPRES	APB 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
15:11	--	RFU：未实现，读为 0

位号	助记符	功能描述
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：RCLF 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：RCLF 101：XTLF 110：RCLP 111：RCHF

14.11.2 RCHF 时钟控制寄存器（CMU_RCHCR）

名称	CMU_RCHCR							
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	-							RCHFEN
位权限	U-0							R/W-1

位号	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 0111：40MHz 1111：48MHz

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

14.11.3 RCHF 调校寄存器（CMU_RCHTR）

名称	CMU_RCHTR							
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	RCHFTRIM							
位权限	R/W-1000 0000							

位号	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	RCHFTRIM	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 0.25% 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。

14.11.4 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	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-	PLLDB						
位权限	U-0	R/W-0011111						
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			PLLOSEL	-	PLLINSEL	PLLEN

位权限	R/Dy-0	R/W-000	R/W-0	U-0	R/W-0	R/W-0

位号	助记符	功能描述
31:23	--	RFU：未实现，读为 0
22:16	PLLDB	PLL 倍频比，输出时钟频率为 1M*(PLLDB+1) 0011111：输出 32 倍频 0101111：输出 48 倍频 0111111：输出 64 倍频 *注：请勿使用高于 64 倍频的设置。*
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	PLLOSEL	PLL 输出选择寄存器 0：选择 PLL 一倍输出作为数字电路内的 PLL 时钟 1：选择 PLL 两倍输出作为数字电路内的 PLL 时钟
2	--	RFU：未实现，读为 0
1	PLLINSEL	PLL 输入选择寄存器 0：RCHF 1：XTHF
0	PLLEN	PLL 使能寄存器 1：使能 PLL 0：关闭 PLL

14.11.5 RCLP 控制寄存器（CMU_RCLPCR）

名称	CMU_RCLPCR							
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	-							ENB

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	ENB	RCLP 使能寄存器 0：使能 RCLP 1：关闭 RCLP

14.11.6 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	RCLP_TRIM							
位权限	R/W-1000 0000							

位号	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	RCLP_TRIM	RCLP 调校值寄存器 0000 0000：频率最低 1111 1111：频率最高

14.11.7 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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LSCLKSEL							
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	LSCLKSEL	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

14.11.8 XTHF 控制寄存器（CMU_XTHCR）

名称	CMU_XTHCR							
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	-			HF_CFG				
位权限	U-0			R/W-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							XTHFEN
位权限	U-0							R/W-0

位号	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12:8	HF_CFG	XTHF 振荡强度配置 00000：最弱 11111：最强
0	XTHFEN	XTHF 使能寄存器 0：关闭 XTHF 1：使能 XTHF

14.11.9 RCLF 控制寄存器（CMU_RCLFCR）

名称	CMU_RCLFCR							
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	-						RCLF_PSC	

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	-							RCLF_EN
位权限	U-0							R/W-0

位号	助记符	功能描述
31:18	--	RFU：未实现，读为 0
17:16	RCLF_PSC	RCLF 输出预分频 00：不分频 01：4 分频 10：8 分频 11：16 分频
15:1	--	RFU：未实现，读为 0
0	RCLF_EN	RCLF 使能寄存器 0：关闭 RCLF 1：打开 RCLF

14.11.10 RCLF 调校寄存器（CMU_RCLFTR）

名称	CMU_RCLFTR							
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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RCLF_TRIM							
位权限	R/W -1000_0000							

位号	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	RCLF_TRIM	RCLF 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 1%

14.11.11 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	-
位权限	U-0					R/W-0	R/W-0	U-0

位号	助记符	功能描述
31:3	--	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	--	RFU：未实现，读为 0

14.11.12 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	-
位权限	U-0						R-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					SYSCSE_IF	HFDETIF	-
位权限	U-0					R/W-0	R/W -0	U-0

位号	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
8:3	--	RFU：未实现，读为 0
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

位号	助记符	功能描述
0	--	RFU：未实现，读为 0

14.11.13 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	-			VREF1p2_PCE	OPA_PCE	ATT_PCE	COMP_PCE	SVD_PCE
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	PAD_PCE	ANAC_PCE	IWDT_PCE	SCU_PCE	PMU_PCE	RTCA_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

位号	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12	VREF1p2_PCE	VREF1p2 模块总线时钟使能，高使能
11	OPA_PCE	OPA 总线时钟使能，高使能
10	ATT_PCE	AUTOTRIM 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	ANAC_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

14.11.14 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	DIVAS_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	FLASH_PCE	DMA_PCE	LCD_PCE	AES_PCE	TRNG_PCE	CRC_PCE
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
31:12	--	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	DIVAS_PCE	硬件除法器总线时钟使能，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	FLASH_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	LCD_PCE	LCD 总线时钟使能，高使能
2	AES_PCE	AES 总线时钟使能，高使能
1	TRNG_PCE	RNG 总线时钟使能，高使能
0	CRC_PCE	CRC 总线时钟使能，高使能

14.11.15 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	-							I2C_PCE
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				CAN_PCE	LPUART2_PCE	LPUART1_PCE	LPUART0_PCE
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	U7816_PCE	UARTIR_PCE	UART5_PCE	UART4_PCE	UART3_PCE	-	UART1_PCE	UART0_PCE
位权限	R/W-0	R/W-0	R/W-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	-					SPI2_PCE	SPI1_PCE	SPI0_PCE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:25	-	RFU：未实现，读为 0
24	I2C_PCE	I2C 总线时钟使能，高有效

位号	助记符	功能描述
23:20	-	RFU：未实现，读为 0
19	CAN_PCE	CAN 总线时钟使能，高有效
18	LPUART2_PCE	LPUART1 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	LPUART 总线时钟使能，高有效
15	U7816_PCE	7816 总线时钟使能，高有效
14	UARTIR_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	-	RFU：未实现，读为 0
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
7:3	-	RFU：未实现，读为 0
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

14.11.16 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	GT2_PCE	GT1_PCE	GT0_PCE	BT32_PCE
位权限	U-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
7:5	-	RFU：未实现，读为 0
4	AT_PCE	高级定时器总线时钟使能，高有效
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

14.11.17 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	-		LPUART1CKS		LPUART0CKS	
位权限	U-0	R/W-0	U-0		R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LPUART2CKS		-				I2CCKS	
位权限	W-00		U-0				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	-	RFU：未实现，读为 0
28	-	RFU：未实现，读为 0
27:26	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：RCLF 分频（16 分频得到 38.4Khz） 11：RFU
25:24	LPUART0CKS	LPUART0 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：RCLF 分频（16 分频得到 38.4Khz） 11：RFU
23:22	LPUART2CKS	LPUART2 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：RCLF 分频（16 分频得到 38.4Khz） 11：RFU
21:18	-	RFU：未实现，读为 0
17:16	I2CCKS	I2C 主机工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：RCLF_PSC

位号	助记符	功能描述
15:14	BT16CKS	BSTIM16 工作时钟选择 00：APBCLK 01：LSCLK 10：RCLP 11：RCLF_PSC
13:12	BT32CKS	BSTIM32 工作时钟选择 00：APBCLK 01：LSCLK 10：RCLP 11：RCLF_PSC
11:10	LPT16CKS	LPTIM16 工作时钟选择 00：APBCLK 01：LSCLK 10：RCLP 11：RCLF_PSC
9:8	LPT32CKS	LPTIM32 工作时钟选择 00：APBCLK 01：LSCLK 10：RCLP 11：RCLF_PSC
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	CAN 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 01：RCHF 10：SYSCLK 11：XTHF
1:0	UART0CKS	UART0 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：XTHF

14.11.18 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	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			RNGPSC			-	
位权限	U-0			R/W-000			U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			ADCPSC			ADCCKS	
位权限	U-0			R/W-000			R/W-00	

位号	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12:10	RNGPSC	随机数发生器时钟分频 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	ADCPSC	ADC 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
1:0	ADCCKS	ADC 工作时钟选择 00：RCLF_PSC 01：RCHF 10：XTHF 11：PLL

14.11.19 外设工作时钟配置寄存器 3（CMU_OPCCR3）

名称	CMU_OPCCR3							
offset	0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTICKE	FLASHCKE	LPU1CKE	LPU0CKE	-			RNGCKE
位权限	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	-		LPU2CKE	I2CCKE	-			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	-				UART1CKE	UART0CKE

位权限	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	-				BT16CKE	BT32CKE	LPT16CKE	LPT32CKE
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	EXTICKE	EXTI 工作时钟使能，高有效
30	FLASHCKE	Flash 擦写时钟使能，高有效
29	LPU1CKE	LPUART1 工作时钟使能，高有效
28	LPU0CKE	LPUART0 工作时钟使能，高有效
27:25	--	RFU：未实现，读为 0
24	RNGCKE	随机数发生器工作时钟使能，高有效
23:22	--	RFU：未实现，读为 0
21	LPU2CKE	LPUART2 工作时钟使能，高有效
20	I2CCKE	I2C 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	CANCKE	CAN 总线模块工作时钟使能，高有效
13:10	--	RFU：未实现，读为 0
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

14.11.20 时钟校准控制寄存器（CMU_CCCR）

名称	CMU_CCCR							
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	-						CCLIE	EN
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述

位号	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	CCLIE	时钟校准中断使能 (Clock Calibration interrupt enable) 0：禁止中断 1：允许中断
0	EN	时钟校准使能 1：启动校准 0：关闭校准，校准电路复位

14.11.21 时钟校准配置寄存器（CMU_CCFR）

名称	CMU_CCFR							
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	-				CALPSC		REFPSC	
位权限	U-0				R/W-00		R/W-10	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					CALSEL		REFSEL
位权限	U-0					R/W-00		R/W-0

位号	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11:10	CALPSC	校准时钟预分频 (Calibration clock prescaler) 00：不分频 01：2 分频 10：4 分频 11：8 分频
9:8	REFPSC	参考时钟预分频 (Reference clock prescaler) 00：8 分频 01：16 分频 10：32 分频 11：64 分频
7:3	--	RFU：未实现，读为 0
2:1	CALSEL	校准时钟选择 (Calibration clock select) 00：RFU 01：RCHF 10：RCLF_PSC 11：XTHF
0	REFSEL	参考时钟选择 (Reference clock select) 0：XTLF 1：RCLP

14.11.22 时钟校准计数值寄存器（CMU_CCNR）

名称	CMU_CCNR							
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	CCL_CNT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCL_CNT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	CCNT	时钟校准计数器计数值 (Clock calibration counter) 校准周期结束后软件通过读取计数值来计算目标时钟频率

14.11.23 时钟校准中断标志寄存器（CMU_CCISR）

名称	CMU_CCISR							
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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							CCLIF
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	CCLIF	时钟校准中断标志 (Clock calibration interrupt flag) 校准周期完成后硬件置位，软件写 1 清零

15 电源电压监测（SVD）

15.1 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

15.2 结构框图

下图是电源检测电路的模块框图。

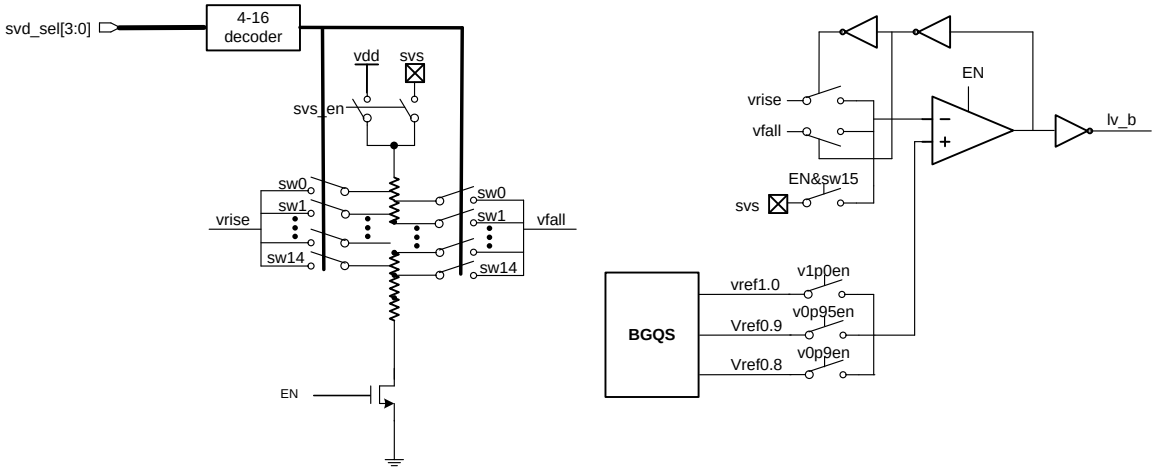


图 15-1 低压检测电路框图

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：

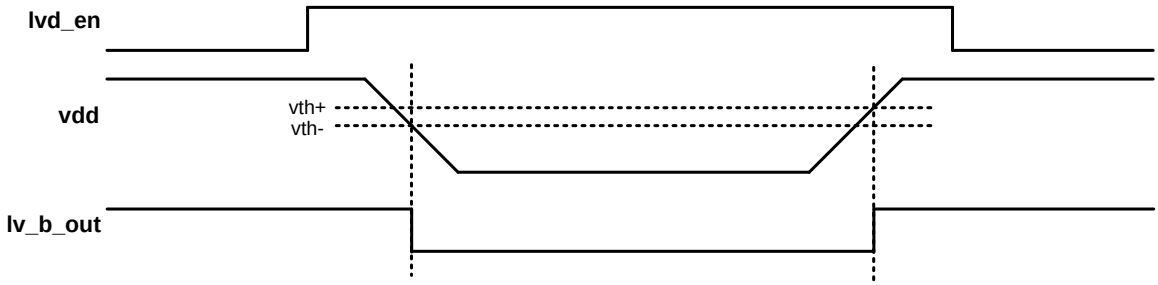


图 15-2 低压检测电路工作时序

15.3 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）。

15.4 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，比较基准电压有1.0V、0.9V、0.8V三个档位供软件选择。

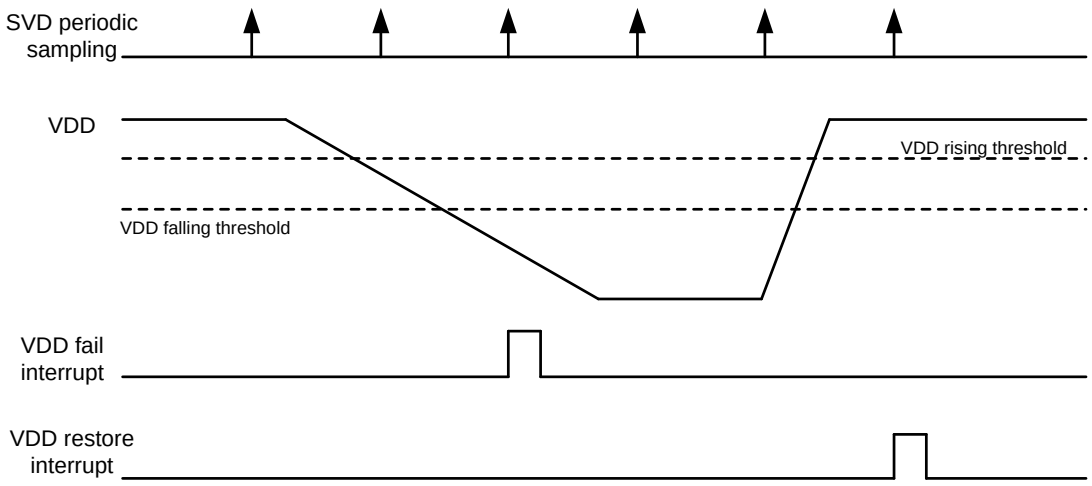


图 15-3 电源检测电路间歇工作模式

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。

15.5 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。

SVD开启窗口中，比较器加基准电压总功耗小于5uA。每次开启时间约100us，如果开启间隔设置为1s，则平均电流小于5nA。

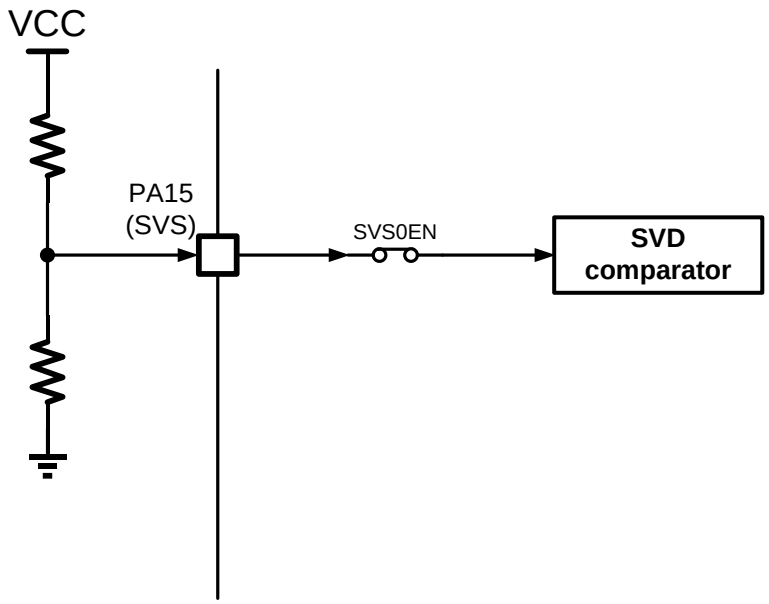
通过间歇使能模式，芯片可以保持对VDD电源供电情况的监视，同时几乎不会带来额外的休眠功耗增加。

15.6 外部电源检测

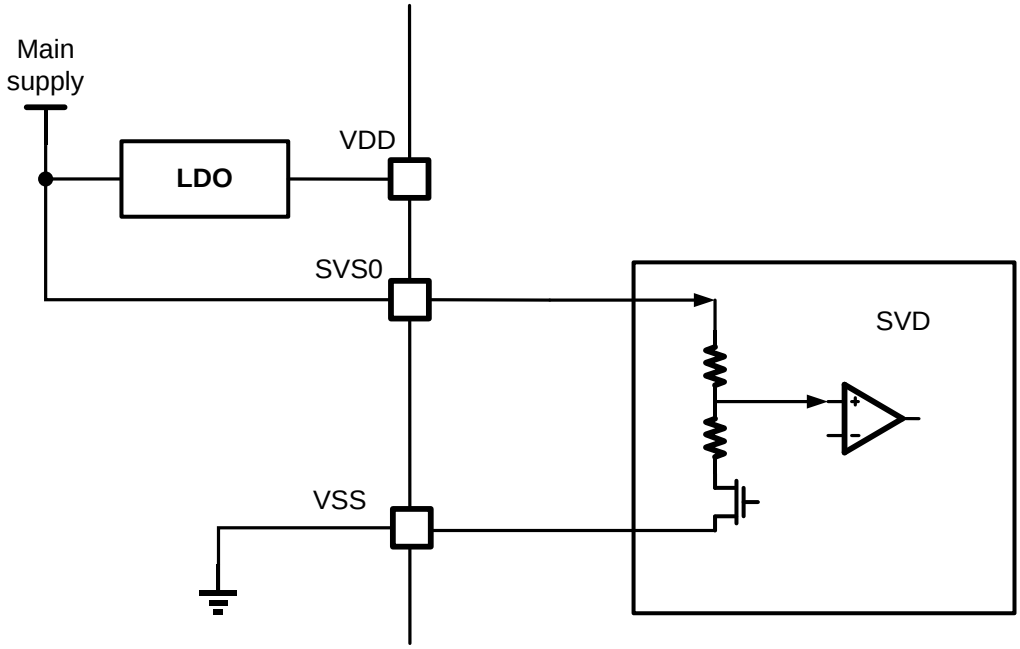
SVD除了可以检测芯片电源，也可以对外部电压信号进行掉电或上电检测。

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。注意输入到SVS引脚的被检测电压不得高于芯片电源电压。

下图为外部电阻分压的外部电源检测。此时软件应置位EN寄存器，但是要清零LDEN关闭内部电阻串，以免对外部电阻形成并联。



下图为内部电阻分压的外部电源检测：



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 *注意：此时外部输入电压不能高于电源电压*
	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

15.7 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

15.8 寄存器

地址	名称	符号
SVD(模块偏移地址：0x40012800)		
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

15.8.1 SVD 配置寄存器（SVD_CFGR）

名称	SVD_CFGR							
offset	**错误!未找到引用源。**							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	

位号	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 15.7 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

15.8.2 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						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	TE	SVD 测试使能，避免写 1 (SVD test enable)
7:2	--	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

15.8.3 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

位号	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

15.8.4 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

位号	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

15.8.5 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	V0P95EN	V0P9EN
位权限	U-0					R/W-1	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	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	V0P95EN	0.95V 基准输入使能信号 (0.95V reference enable) 1：使能 0.95V 基准输入 0：关闭 0.95V 基准输入
0	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

16 AES 硬件运算单元（AES）

16.1 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

16.2 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。

16.3 AES 数据流处理模式

AES有4种数据流处理模式：ECB，CBC，CTR，GCM。

16.3.1 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16-1和图16-2所示。

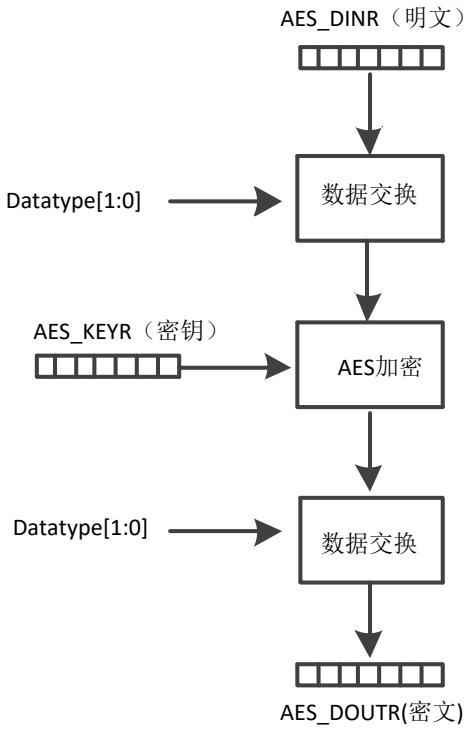


图 16-1 ECB 模式加密流程

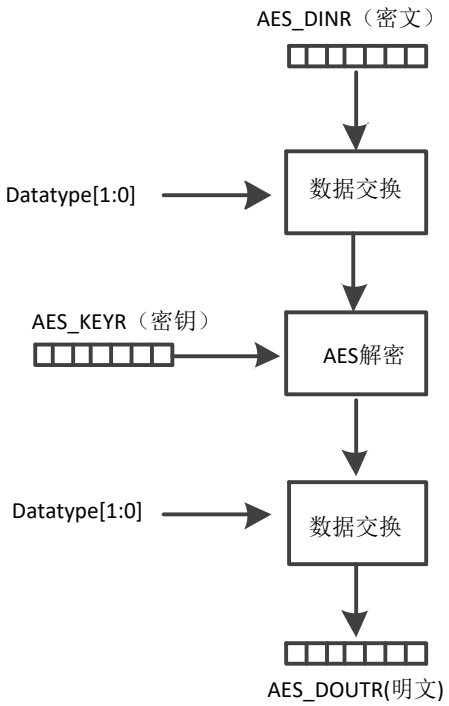


图 16-2 ECB 模式解密流程

16.3.2 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16-3和图16-4所示。

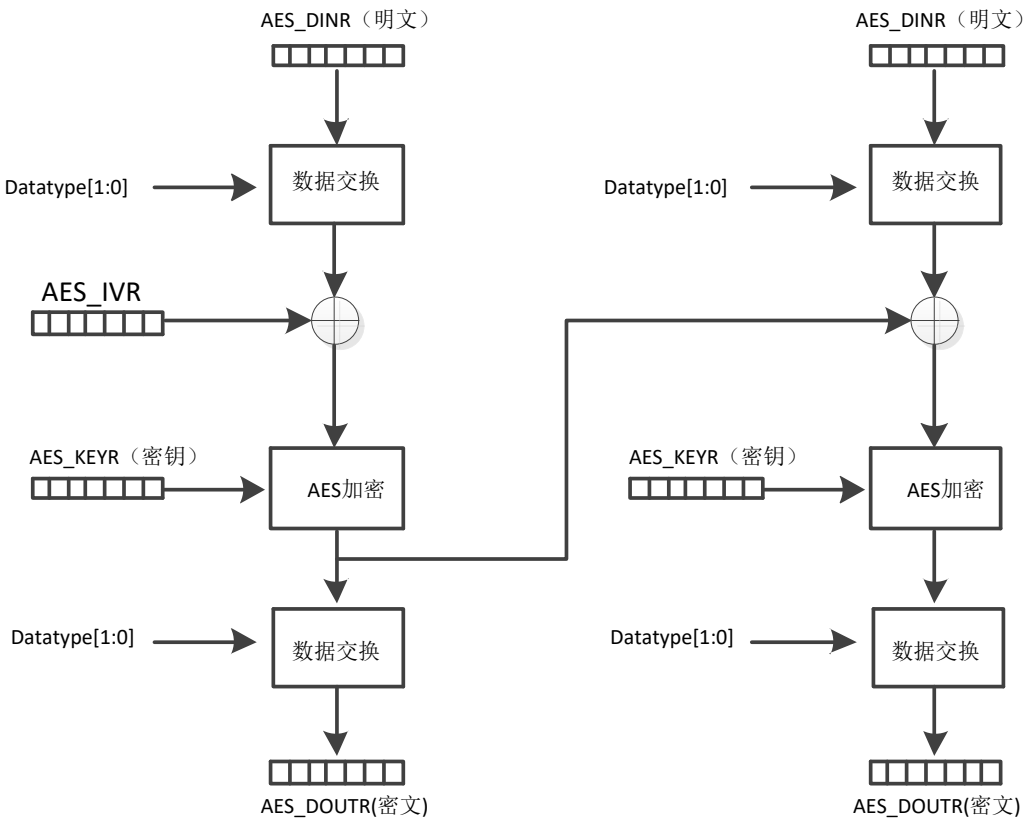


图 16-3 CBC 加密过程

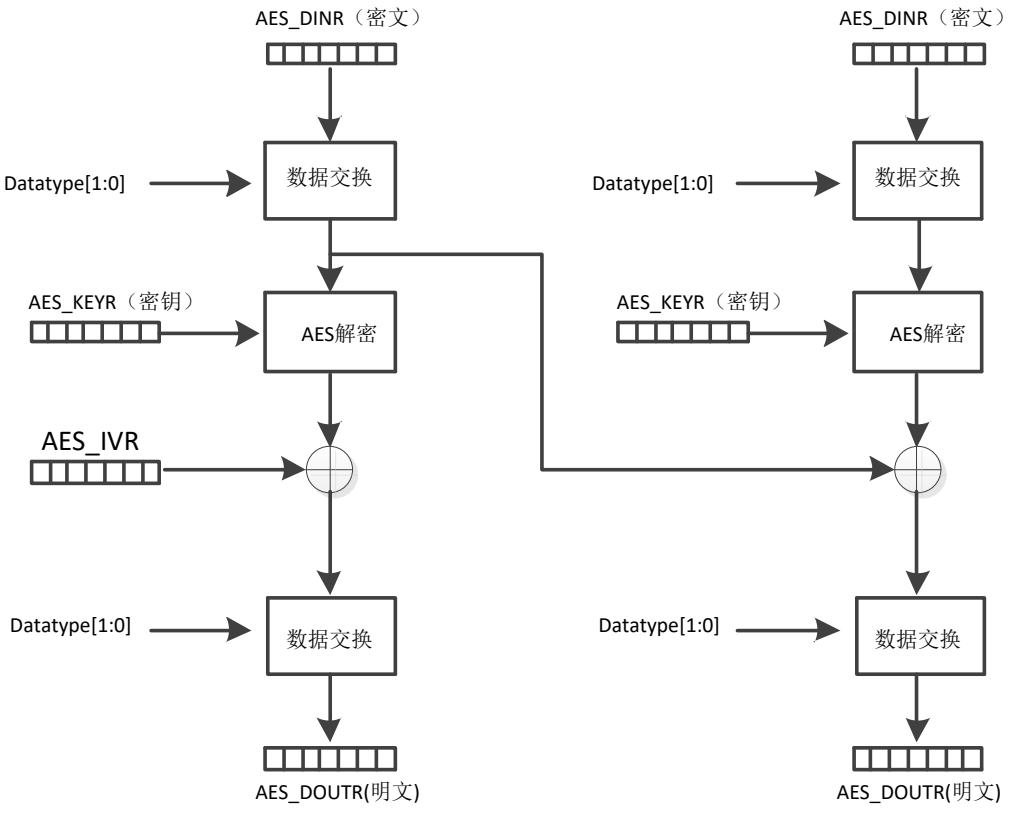


图 16-4 CBC 解密过程

注：在AES工作时读取AES_IVR寄存器的值为0x00000000

16.3.3 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16-5所示

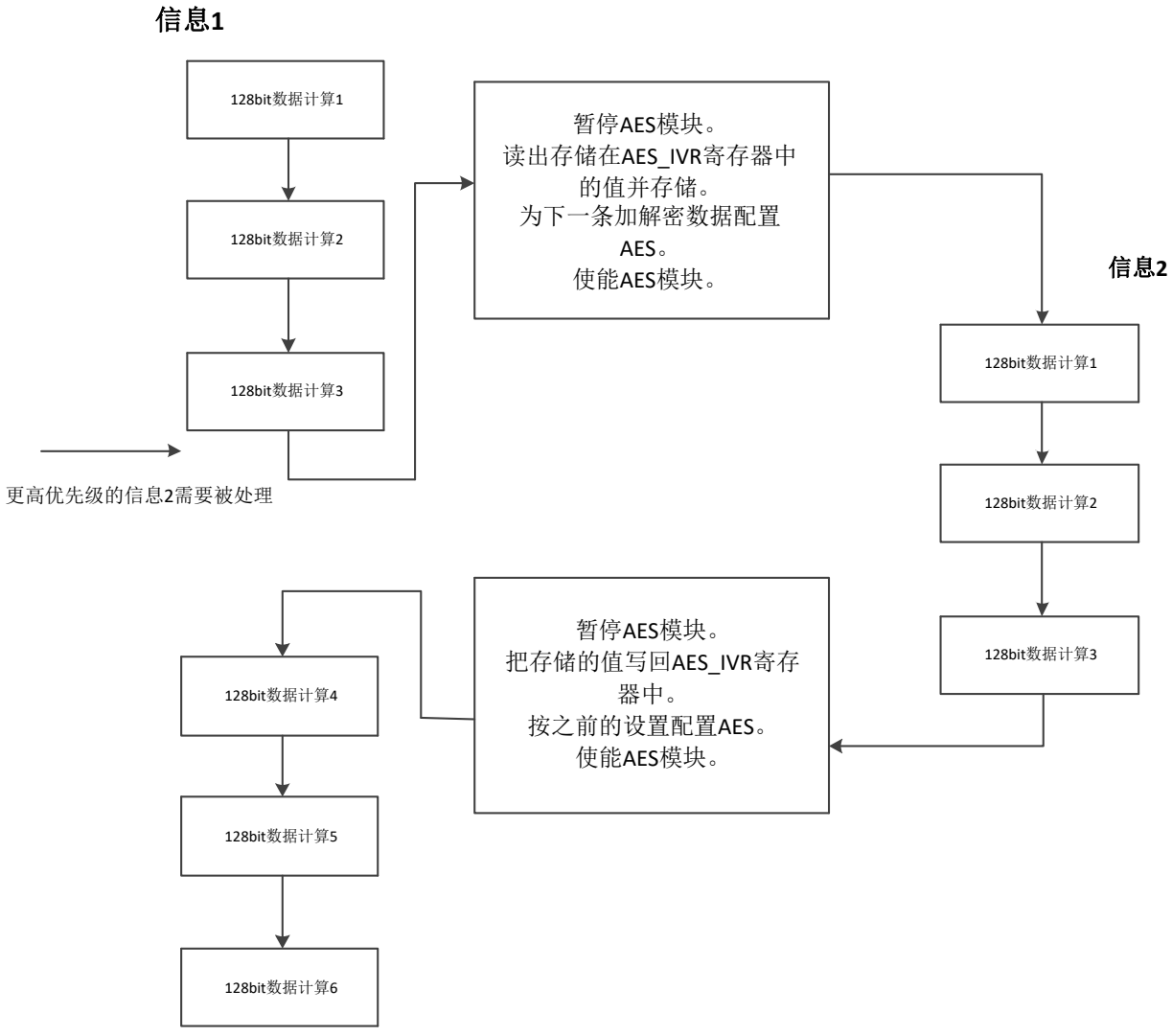


图 16-5 暂停模式流程

16.3.4 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16-6和图16-7所示。

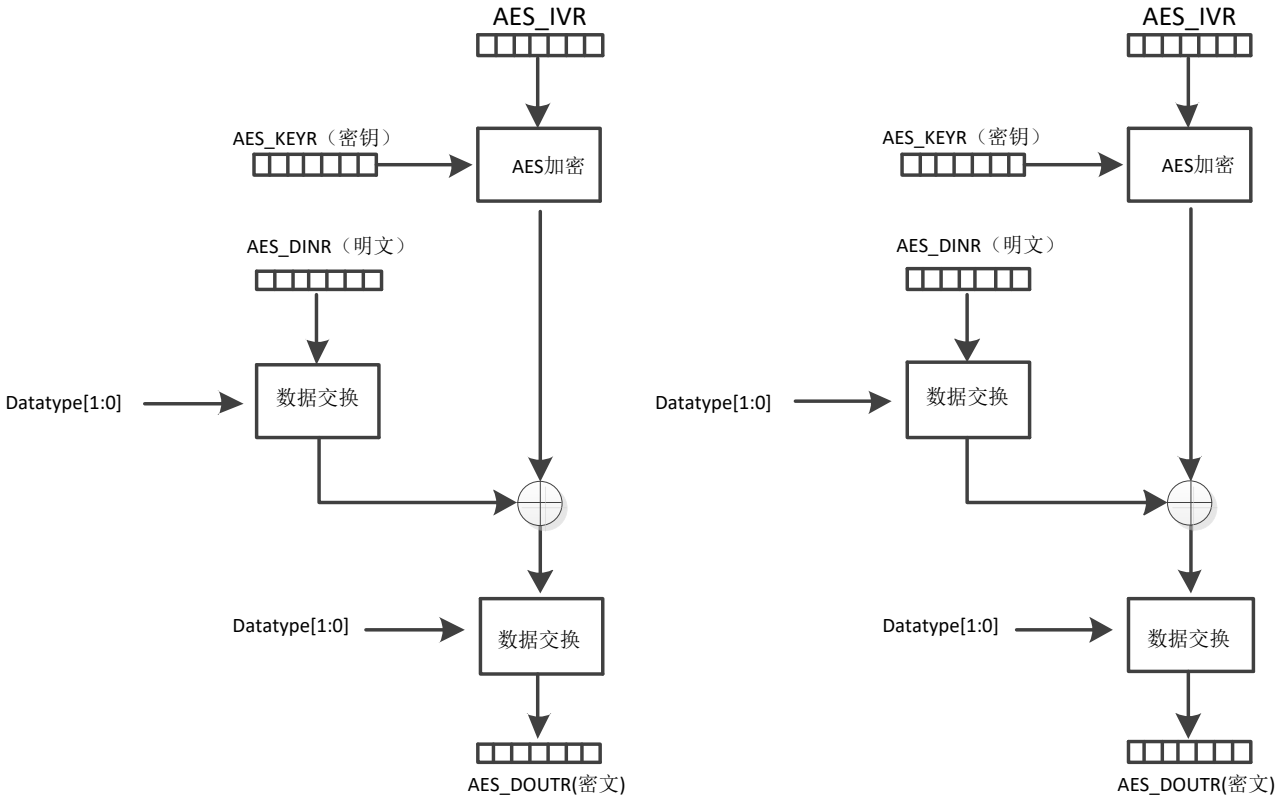


图 16-6 CTR 加密流程

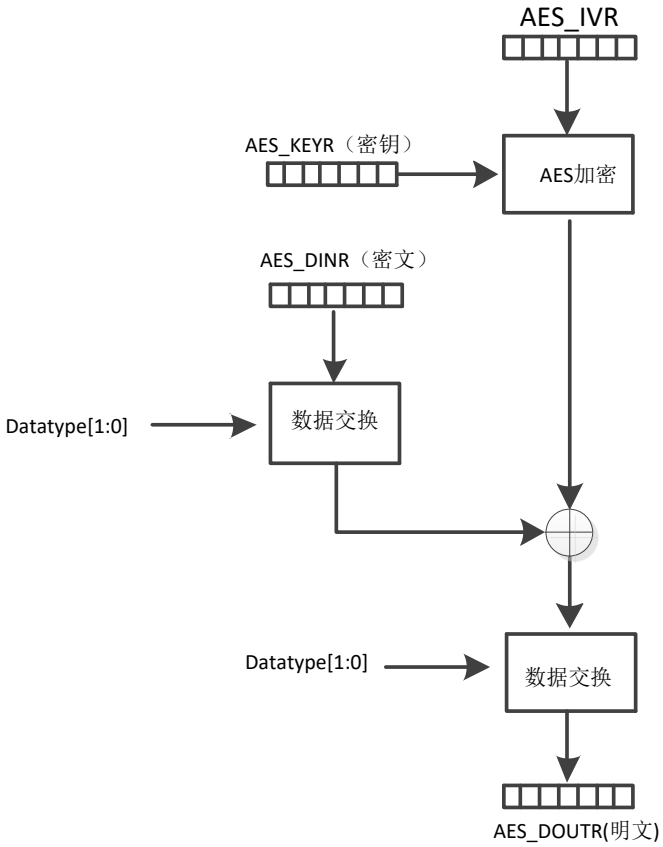


图 16-7 CTR 解密流程

随机数（nonce）和32位计数器存储在IV寄存器中，如图16-8所示

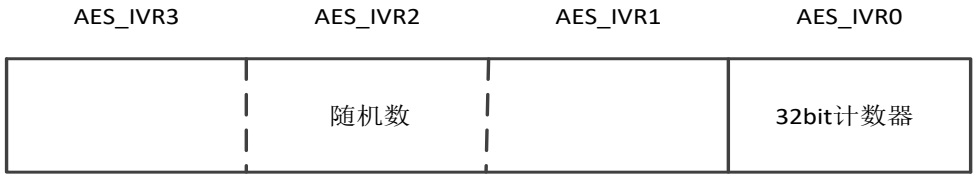


图 16-832 位计数器和随机数的存储方式

CTR模式下密钥扩展和解密模式没有意义。

16.3.5 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。

16.3.6 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16-9，图16-10所示。

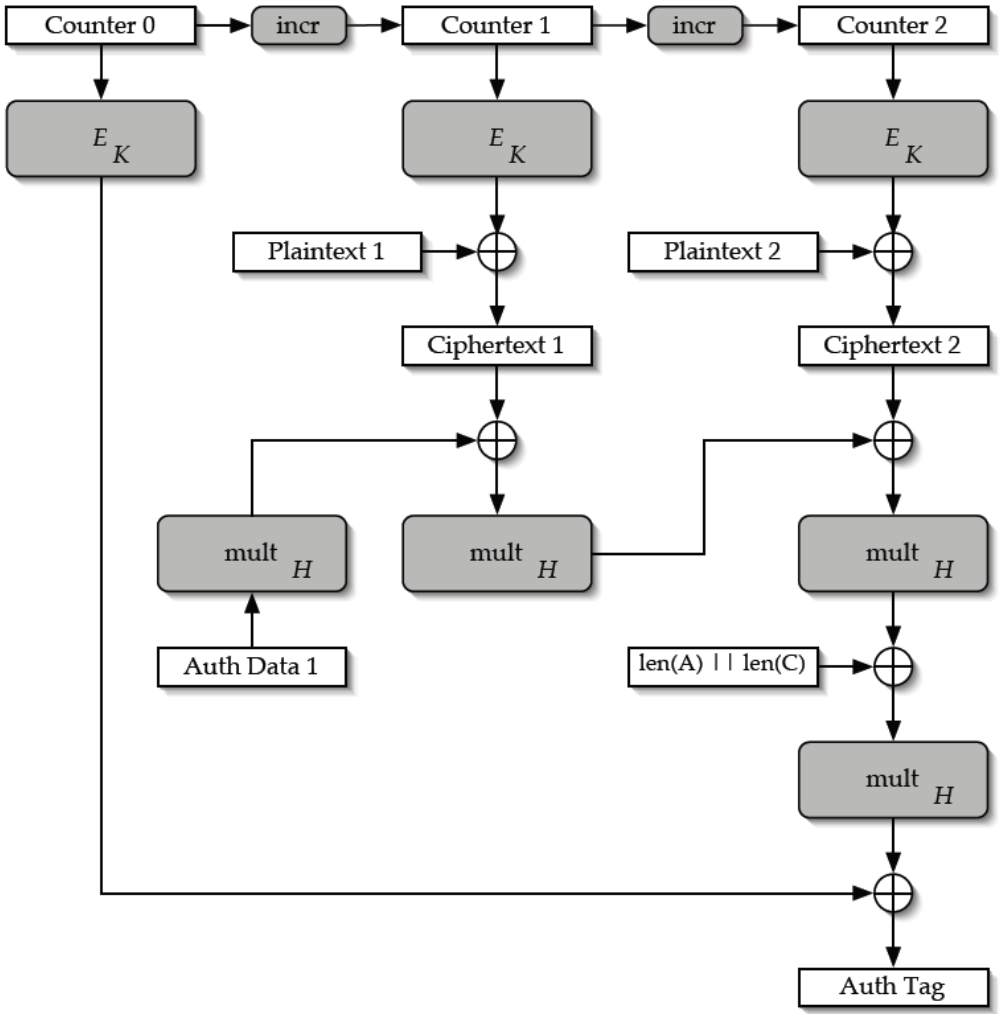


图 16-9 GCM 加密流程

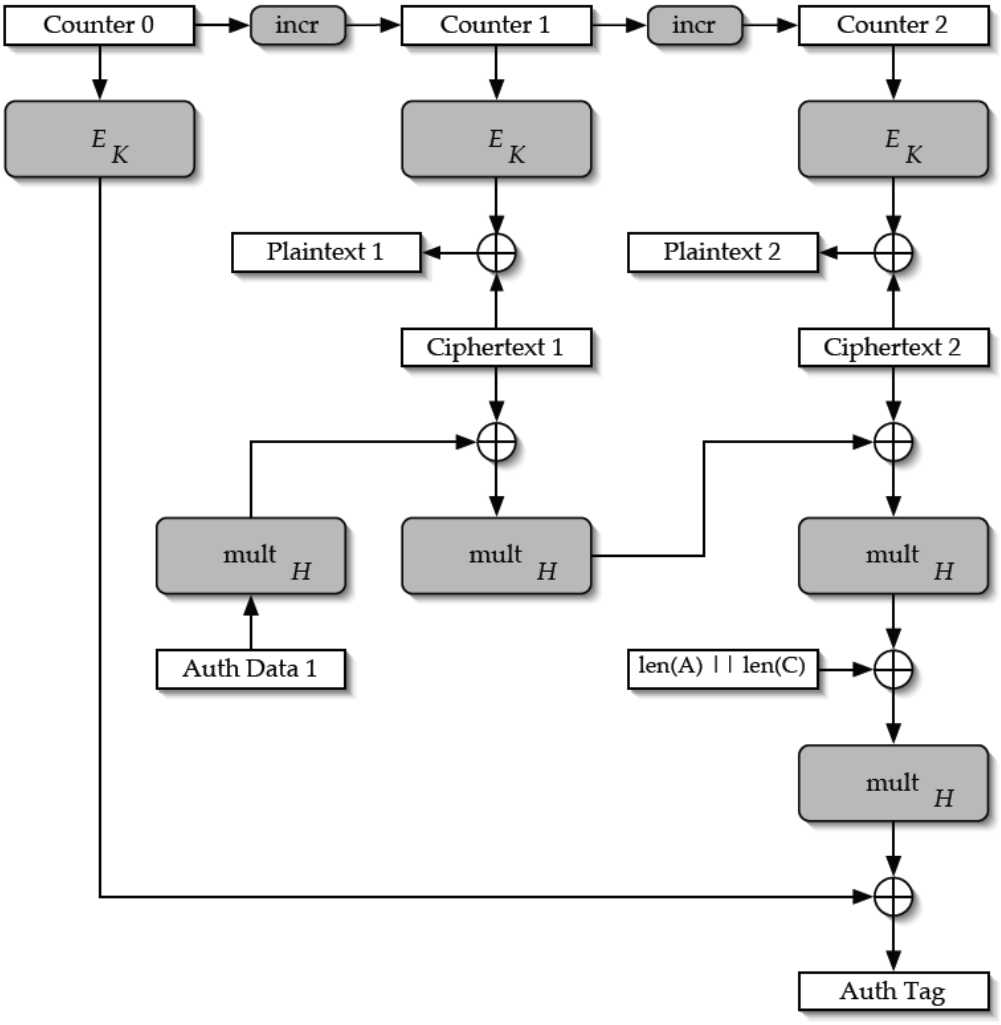


图 16-10GCM 解密流程

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。

16.3.7 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16-11所示。

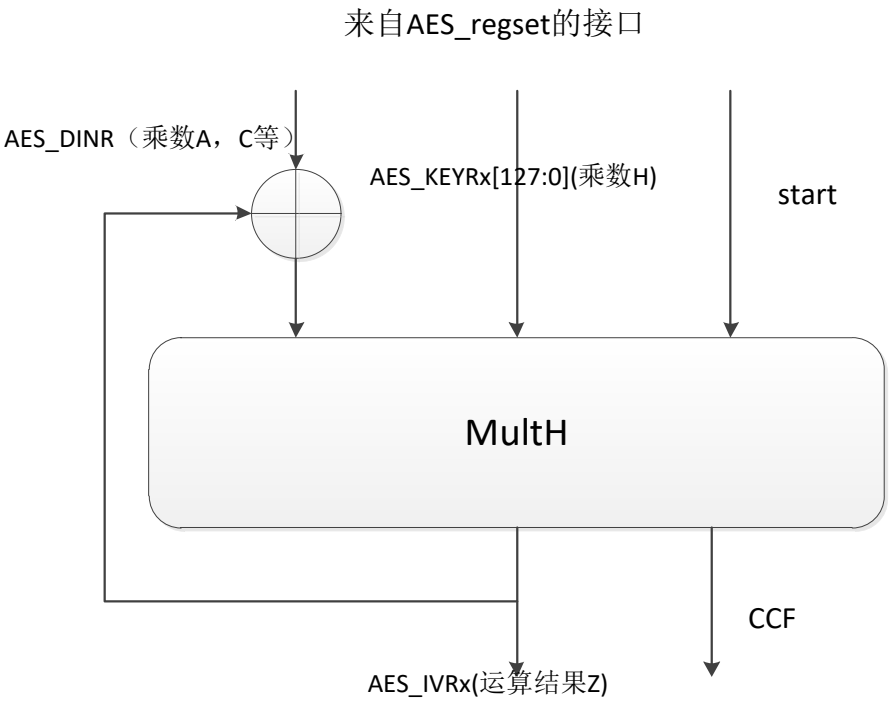


图 16-11 multH 模块框图

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。

16.3.8 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。

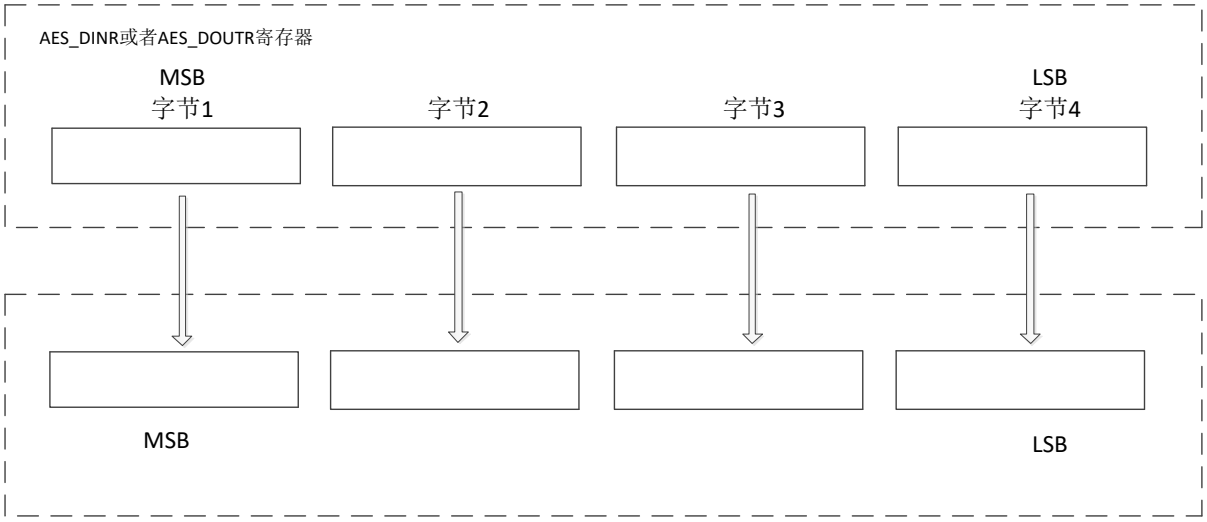
GCM模式的加解密过程和CTR模式相同。认证过程时仅使用MultH模块而不用AES加解密。

- 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。

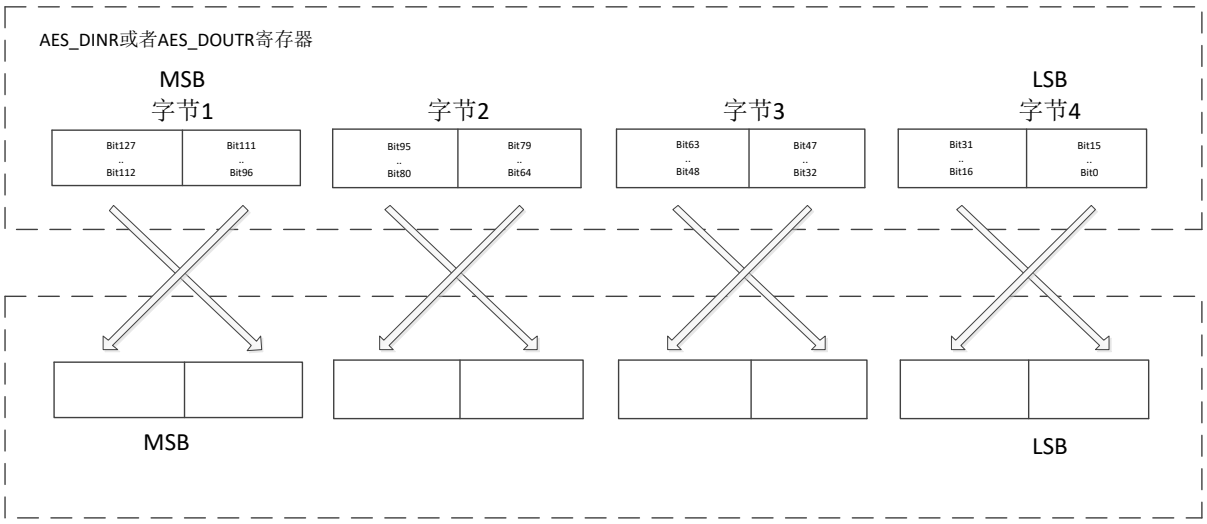
16.4 数据类型

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16-12所示。

Datatpye 2'b00 : 不交换



Datatpye 2'b01 : 半字交换



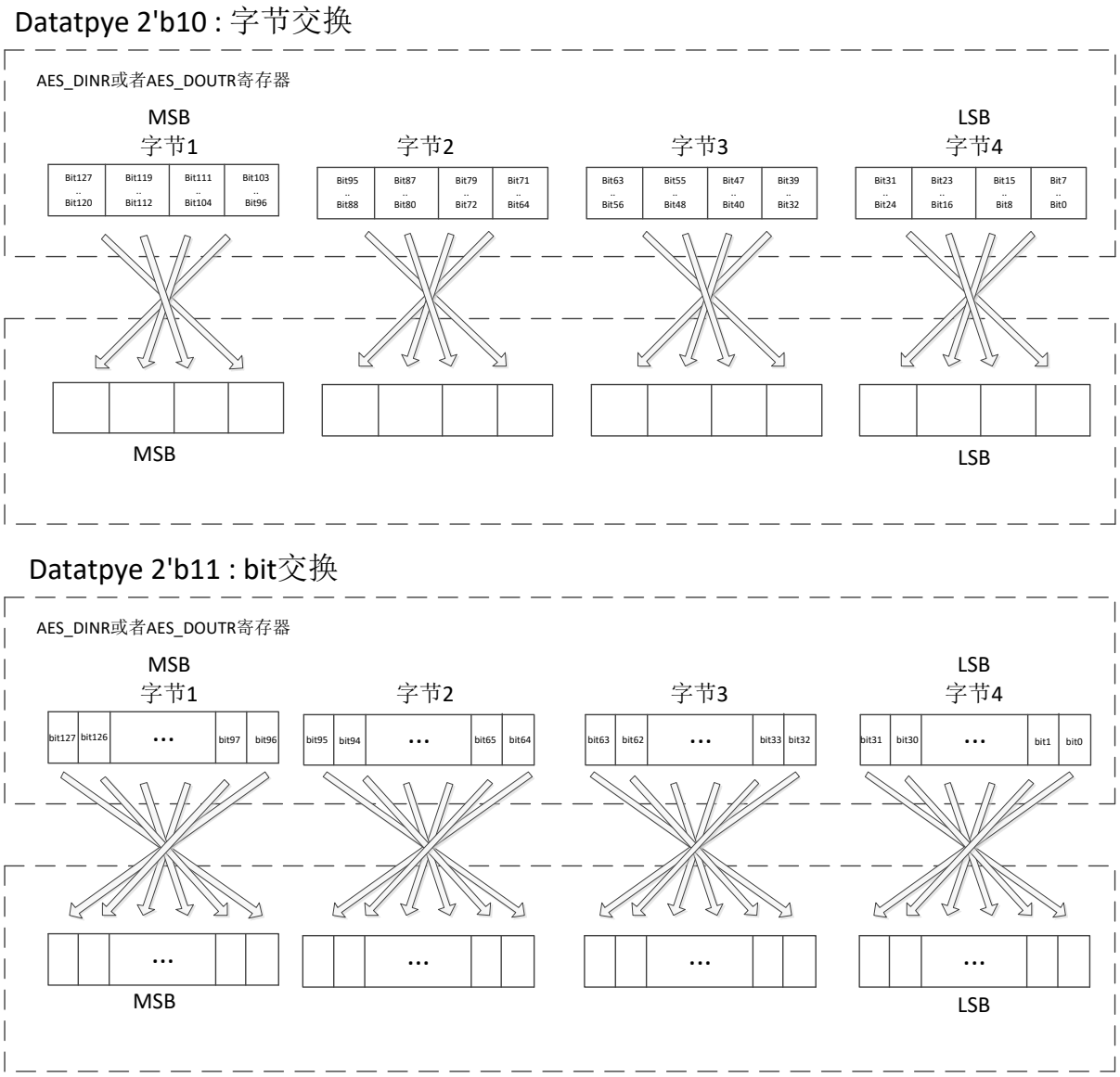


图 16-12 根据数据类型存储数据的示意图

16.5 工作流程

16.5.1 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。

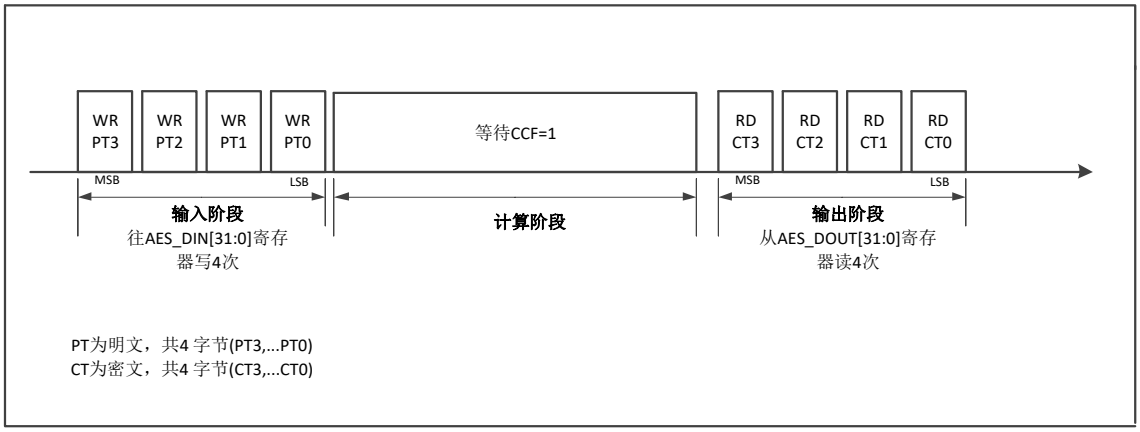


图 16-13 模式 1：加密流程

16.5.2 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。

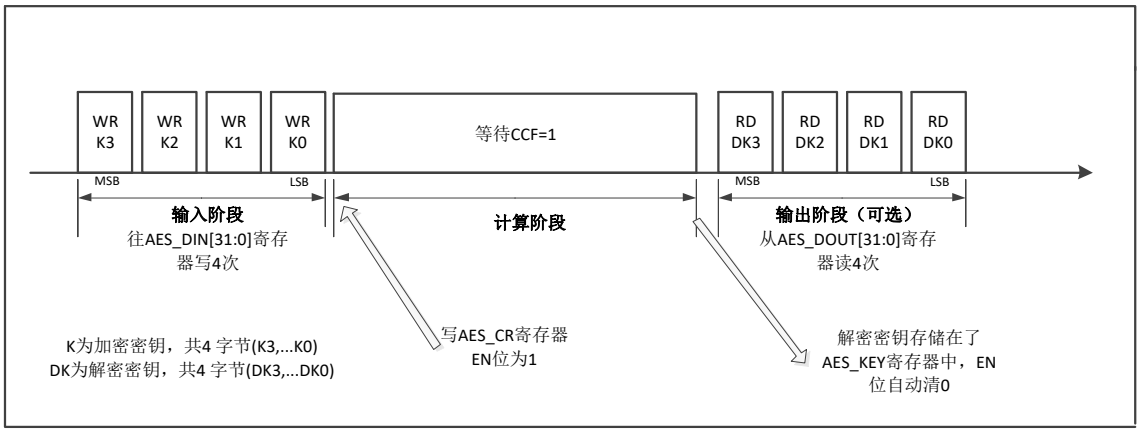


图 16-14 模式 2 示意图

16.5.3 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

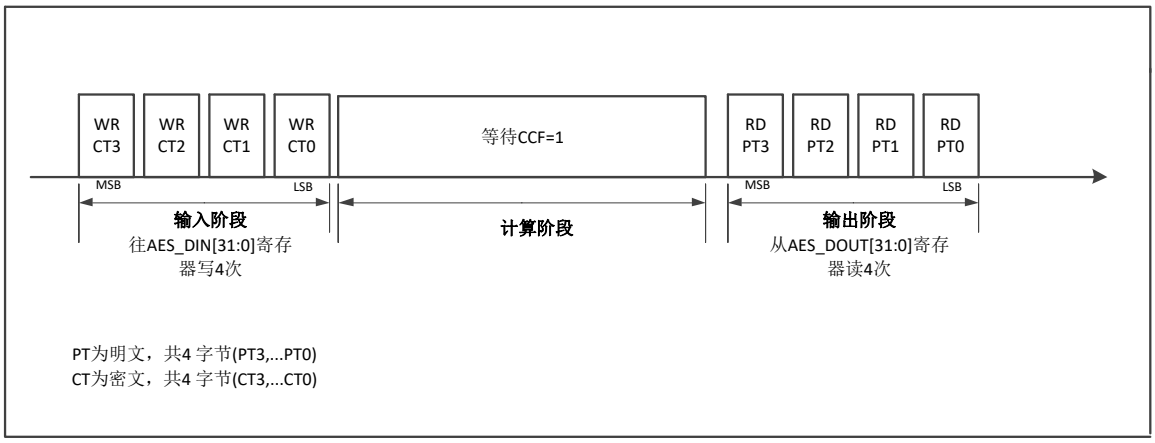


图 16-15 模式 3 示意图

16.5.4 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。

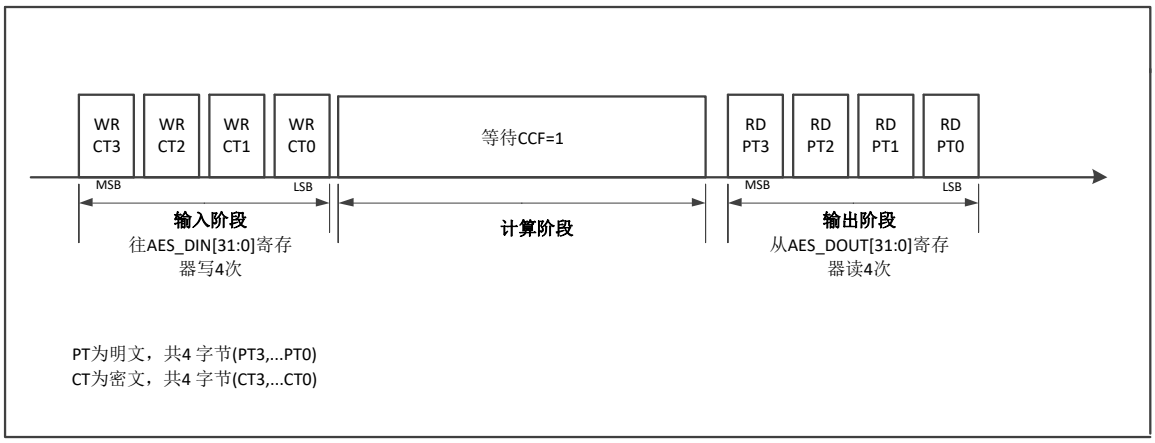


图 16-16 模式 4 示意图

16.5.5 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。

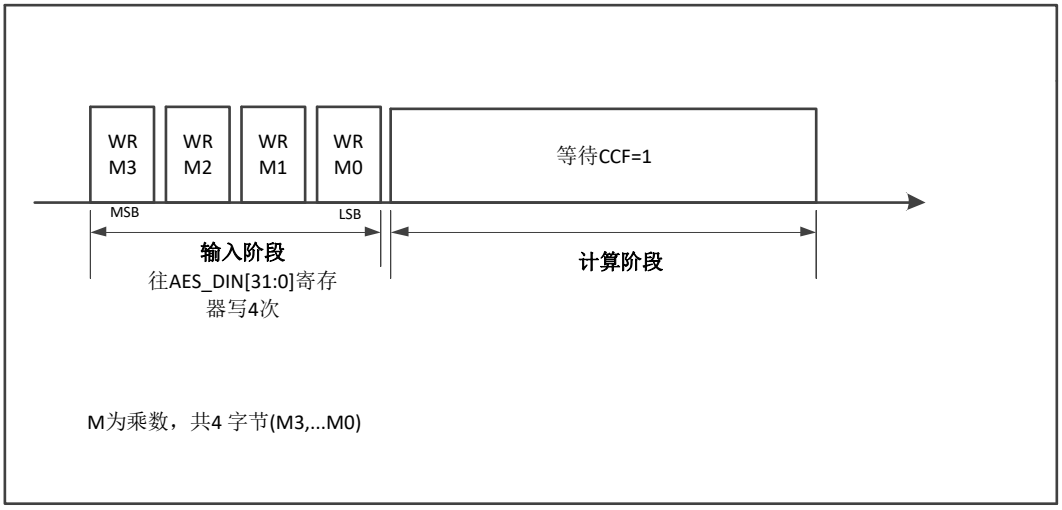


图 16-17 multH 模块使用流程示意图

16.6 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。

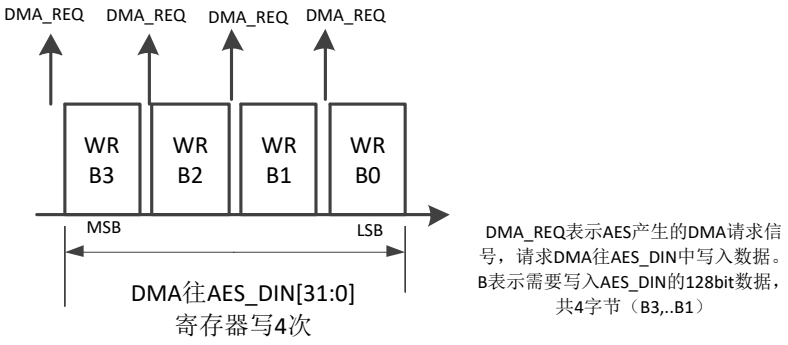


图 16-18 输入时 DMA 请求和数据传输示意图

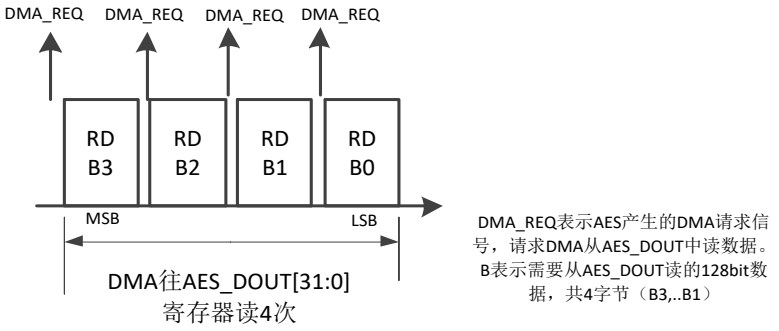


图 16-19 输出时 DMA 请求和数据传输示意图

16.6.1 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。

16.7 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。

16.8 寄存器

offset 地址	名称	符号
AES(模块基地址：0x40013800)		
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

16.8.1 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

位号	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。 (operation MODE)

位号	助记符	功能描述
		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

16.8.2 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)

16.8.3 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

位号	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

16.8.4 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							

位号	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。

位号	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。

16.8.5 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							

位号	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。

16.8.6 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							

位号	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，

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

16.8.7 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							

位号	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)

16.8.8 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							

位号	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]

17 随机数发生器（TRNG）

17.1 概述

FM33LG0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%。

17.2 功能描述

17.2.1 随机数产生

下图为真随机数发生器结构框图。

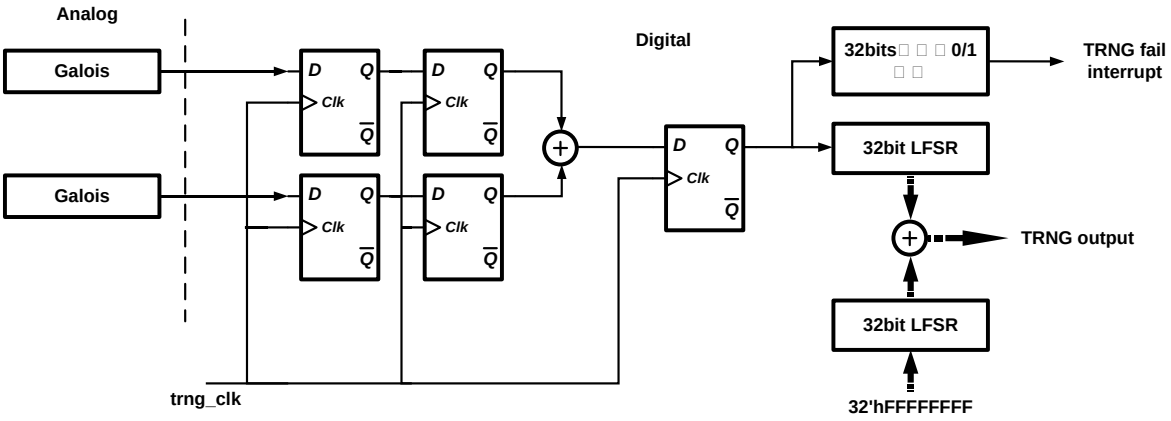


图 17-1 真随机数模块框图

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。

17.2.2 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）。

TRNG的寄存器配置使用APBCLK。在使用TRNG模块前需要置位总线时钟使能寄存器TRNG_PCE和模块工作时钟使能寄存器RNGCKE。参见CMU模块寄存器说明。

工作时钟示意图如下：

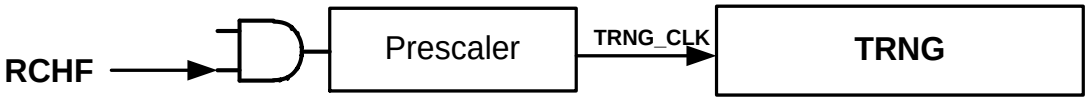


图 17-2 真随机数模块工作时钟

17.2.3 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。

17.2.4 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

17.3 寄存器

模块基地址：0x40013C00

offset 地址	名称	符号
0x00	随机数控制寄存器（ Random Number Generator Control Register）	RNG_CR
0x04	随机数/CRC 结果输出寄存器（Random Number Generator Data Output Register）	RNG_DOR
0x10	RNG 标志寄存器（ Random Number Generator Status Register）	RNG_SR
0x14	CRC 控制寄存器（CRC Control Register）	RNG_CRCCR
0x18	CRC 输入数据寄存器（CRC Data input Register）	RNG_CRCDIR
0x1C	CRC 标志寄存器（CRC Status Register）	RNG_CRCSR

17.3.1 随机数控制寄存器（RNG_CR）

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	-							RNGEN
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	RNGEN	RNG 使能寄存器，软件写 1 启动 (RNG enable) 1：启动 RNG 0：关闭 RNG

17.3.2 随机数/CRC 结果输出寄存器（RNG_DOR）

名称	RNG_DOR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	RNGOUT[31:24]							
位权限	R							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RNGOUT[23:16]							
位权限	R							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RNGOUT[15:8]							
位权限	R							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RNGOUT[7:0]							
位权限	R							

Bit	助记符	功能描述
31:0	RNGOUT	随机数生成结果或 CRC 运算结果寄存器，只读 (RNG output)

17.3.3 RNG 标志寄存器（RNG_SR）

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

17.3.4 CRC 控制寄存器（RNG_CRCCR）

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	-							CRCEN
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	CRCEN	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 (CRC enable) 1：CRC 使能 0：CRC 关闭

17.3.5 CRC 输入数据寄存器（RNG_CRCDIR）

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

Bit	助记符	功能描述
31:0	CRCIN	CRC 运算数据输入寄存器 (CRC data input)

17.3.6 CRC 标志寄存器（RNG_CRCSR）

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	-							CRCDONE
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	CRCDONE	CRC 计算完成标志，软件写 0 清零 (CRC calculation done) 1：CRC 计算完成 0：CRC 计算未完成

18 运算放大器（OPA1）

18.1 概述

FM33LG0集成1个运算放大器，可用于放大微弱输入信号，或用于弱驱动信号阻抗匹配。

基本特性如下：

- 输入电压范围rail-to-rail
- 典型GBW 2MHz
- 典型功耗150uA（正常模式），2uA（低功耗模式）
- 最大驱动电流500uA
- 支持standalone模式、buffer模式、PGA模式（x2, x4, x8, x16）
- 典型输入offset +/-3mv，支持用户校准
- OPA输出可连接ADC，用于输入信号预放大和阻抗匹配

18.2 结构框图

下图是OPA1的结构框图：

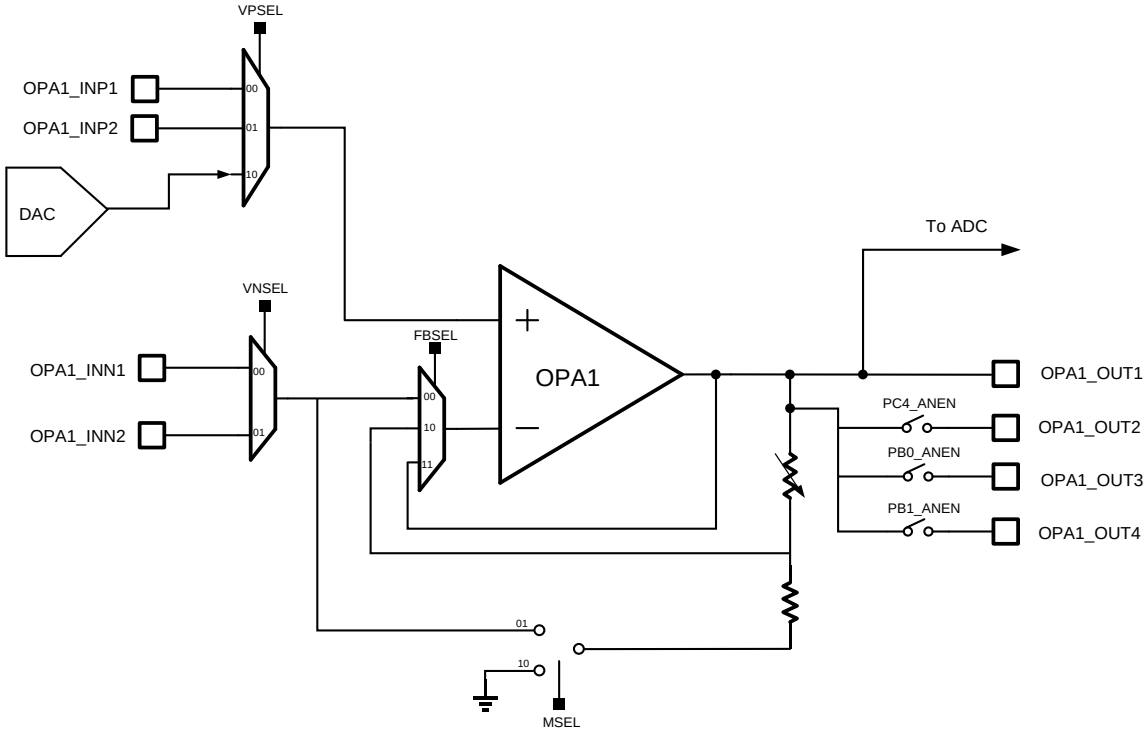


图 18-1OPA1 电路框图

OPA输出连接到4个GPIO上，其中OPA1_OUT1不经过模拟开关，以优化输出阻抗；其他3个输出通路经过模拟开关，默认关闭，如果希望使用OPA1_OUT2/3/4，则必须在GPIO模块置位对应管脚的ANEN寄存器。

根据寄存器配置选择AMUX不同通路，可以实现不同的闭环应用，比如buffer、PGA（内置反馈电阻）、独立运放。输出可以从IO引出，或接给ADC，也可以产生数字信号或中断输出。

下图是OPA作为ADC前端放大应用时的连接关系示意图：

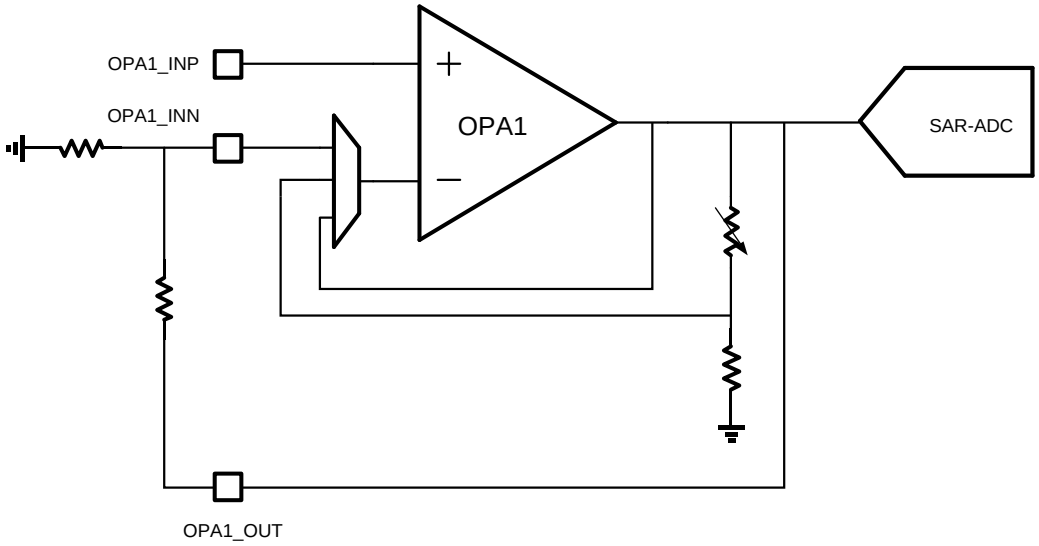


图 18-2OPA 用作 ADC 前端放大

18.3 引脚定义

OPA模块有多个模拟输入输出端口，被复用到多个GPIO上。

引脚	OPAx	符号	功能
PB10	OPA1	OPA1_INN1	运放负端输入
PB15			
PB11		OPA1_INP1	运放正端输入
PD12			
PA6		OPA1_INN2	运放负端输入
PA7		OPA1_INP2	运放正端输入
PB12		OPA1_OUT	运放输出
PB0		OPA1_OUT	运放输出

表 18-1OPA 引脚列表

18.4 功能描述

OPA支持standalone模式、buffer模式和PGA模式（x2, x4, x8, x16）

18.4.1 时钟和复位

OPA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OPA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。

OPA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OPA的寄存器配置。

18.4.2 Standalone 模式（non-inverting amplifier）

此模式下OPA的输入和输出都直接连接到芯片GPIO的电阻通道上，通过在片外连接反馈电阻，用户可以灵活调节运放的负反馈增益，如下图所示：

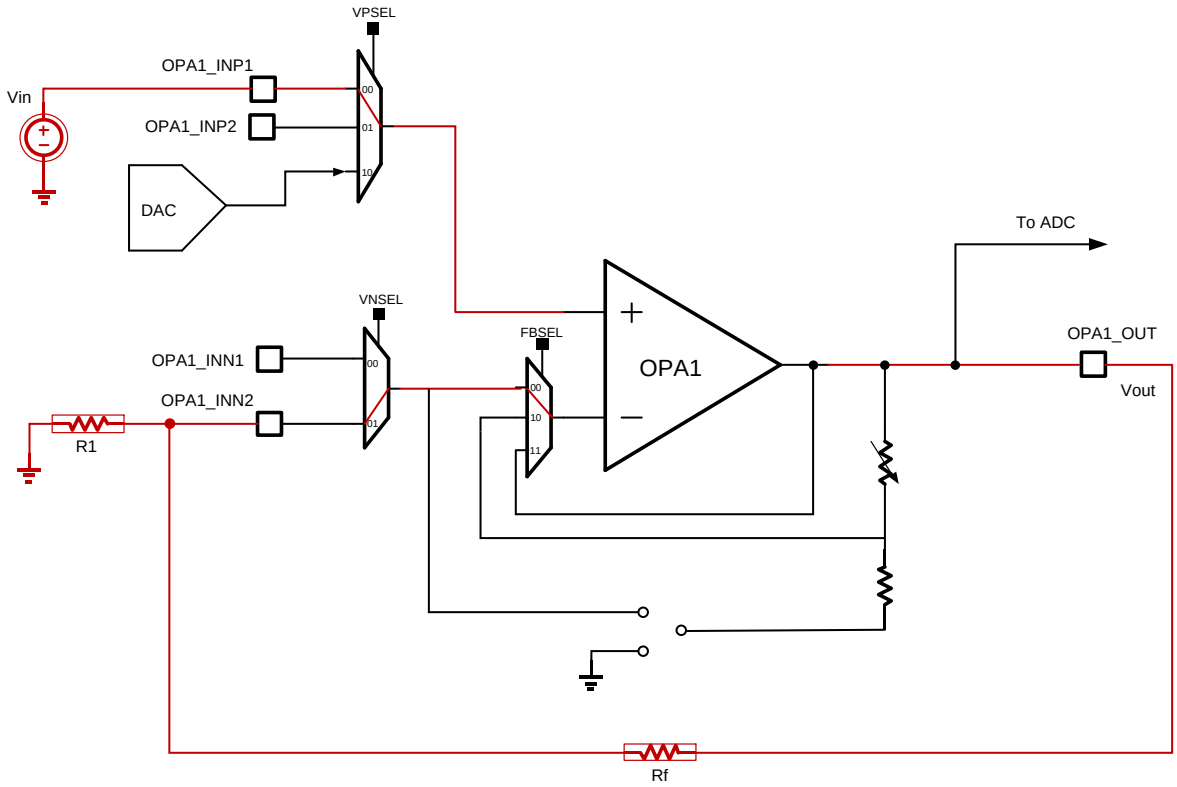


图 18-3OPA 非反相放大

输出信号Vout由下式定义：

$$V_{out} = (1 + \frac{Rf}{R1}) \times V_{in}$$

软件配置方法：

- 配置OPAxCR.VPSEL和VNSEL选择输入IO
- 配置OPAxCR.OPAxMOD为00，即standalone模式
- 使能OPAx

18.4.3 Standalone 模式（inverting amplifier）

此模式下OPA的输入和输出都直接连接到芯片GPIO的电阻通道上，通过在片外连接反馈电阻，用户

可以灵活调节运放的负反馈增益，如下图所示：

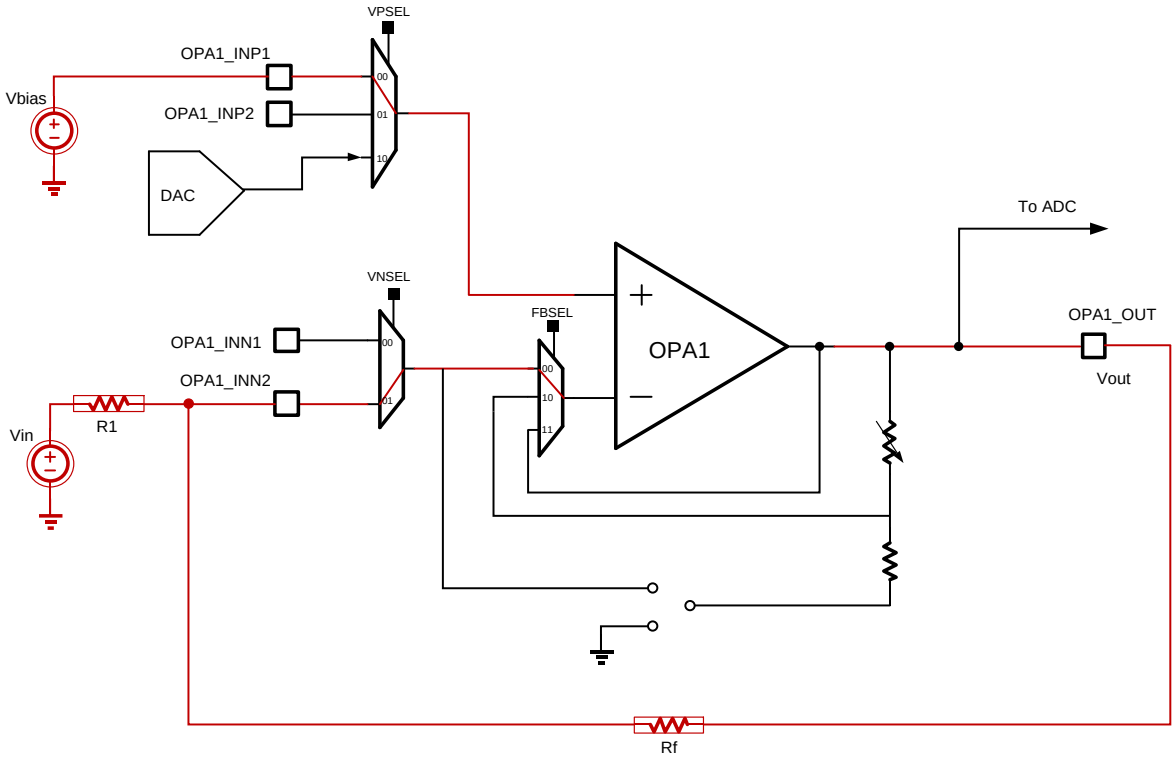


图 18-4 OPA 反相放大

输出信号Vout由下式定义：

$$V_{out} = V_{bias} + (V_{bias} - V_{in}) \times \frac{Rf}{R1}$$

软件配置方法：

- 配置OPAxCR.VPSEL和VNSEL选择输入IO
- 配置OPAxCR.OPAxMOD为00，即standalone模式
- 使能OPAx

18.4.4 Buffer 模式

buffer模式下OPA可用于为ADC输入提供阻抗调整，当输入信号频率与OPA的GBW相适应时，配置为buffer模式的OPA可以增强ADC输入信号的驱动能力。

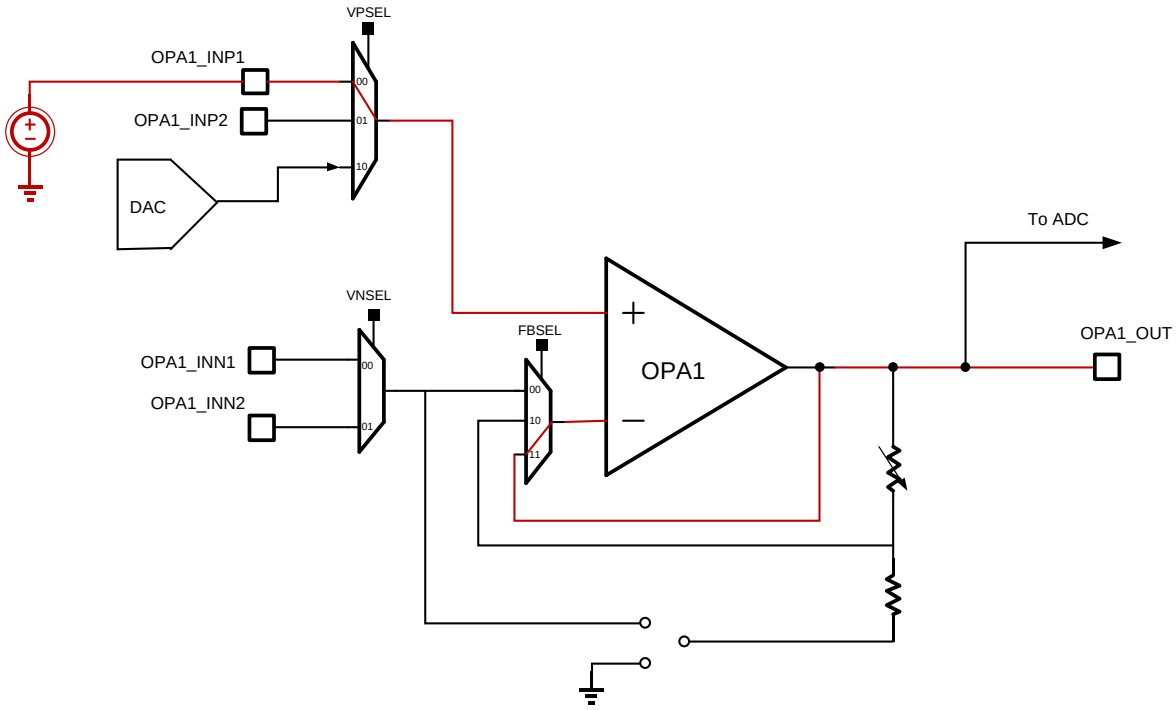


图 18-5OPA 缓冲器模式

软件配置方法：

- 配置OPAxCR.VPSEL和VNSEL选择输入IO
- 配置OPAxCR.OPAxMOD为11，即buffer模式
- 使能OPAx

18.4.5 同相 PGA 模式

PGA模式下，通过调整片内电阻阻值，可以实现固定增益的放大效果，无需连接片外反馈电阻。仅OPA1支持PGA模式，其支持的增益为x2, x4, x8, x16

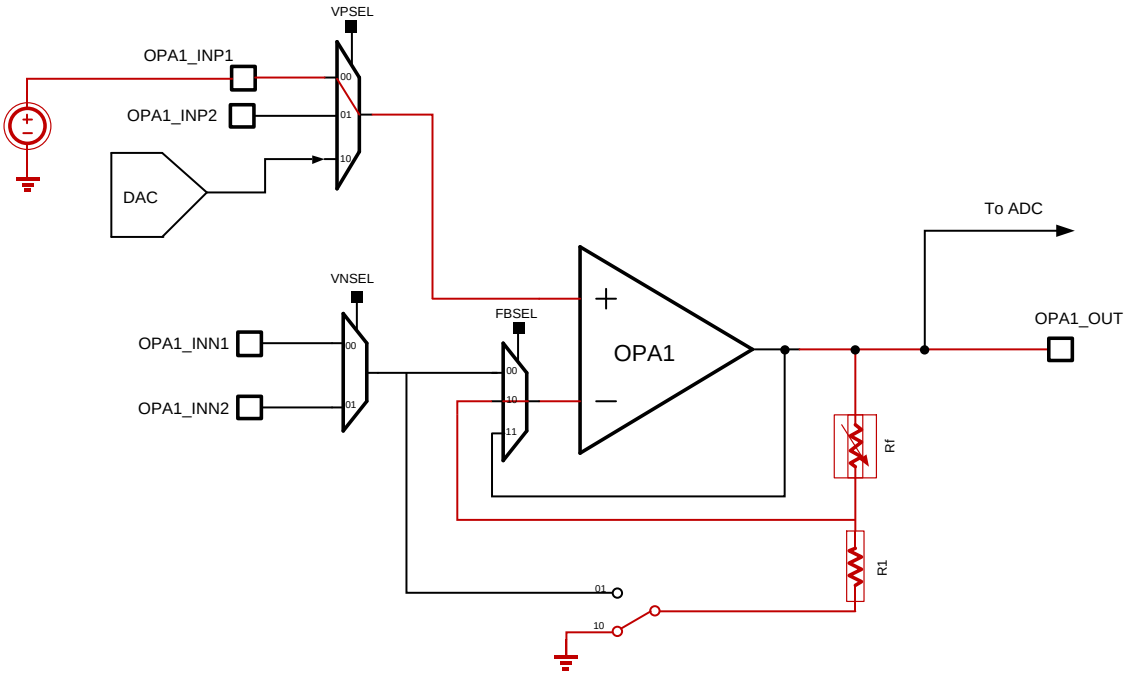


图 18-6 同相 PGA 模式

根据PGA_GAIN寄存器的配置，Rf阻值为10K、30K、70K、150Kohm，R1阻值固定为10Kohm，则PGA同相增益计算公式如下：

$$Gain_{non-inverting} = \frac{R_f + R_1}{R_1}$$

因此可以得到x2, x4, x8, x16的同相放大倍数。

通过配置VN_SEL寄存器，并在OPA1_OUT和OPA1_INN之间连接片外电容，可以实现环路滤波，如下图所示：

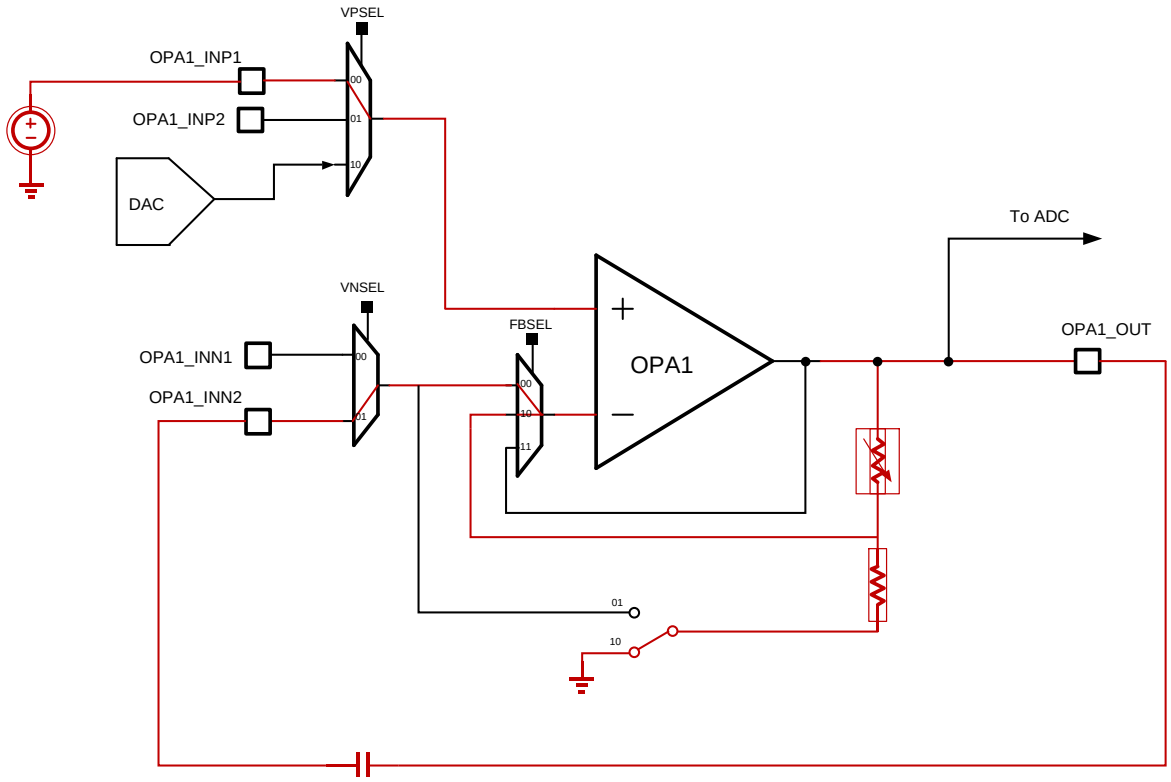


图 18-7OPA 环路滤波

此时，PGA增益可以表示为：

$$Gain_{non-invering} = 1 + \frac{Rf \parallel \frac{1}{2\pi fC}}{R1}$$

软件配置方法：

- 配置OPA1CR.VPSEL和VNSEL选择输入IO
- 配置OPA1CR.OPA1MOD为10，即PGA模式
- 配置OPA1CR.PGA_GAIN选择增益倍数
- 如果需要片外环路滤波，置位OPA1CR.VN_EXC
- 使能OPA1

18.4.6 反相 PGA 模式

在反相PGA模式下可以通过DAC在OPA正端施加一个直流偏置，如下图所示。

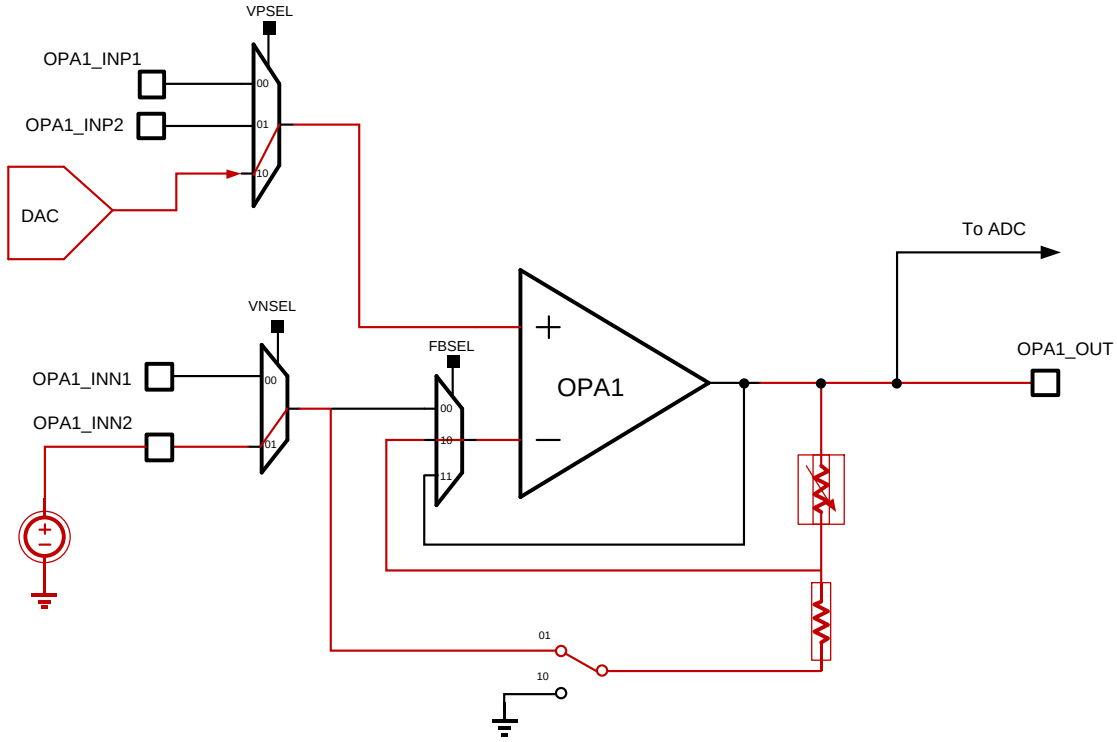


图 18-8 反相 PGA 带 DAC 偏置

根据PGA_GAIN寄存器的配置，Rf阻值为10K、30K、70K、150Kohm，R1阻值固定为10Kohm，则PGA反相增益计算公式如下：

$$V_{out} = V_{DAC} + (V_{DAC} - V_{in}) \times \frac{Rf}{R1}$$

因此可以得到x1, x3, x7, x15的反相放大倍数。

或者可以通过外部引脚输入直流偏置，如下图所示。

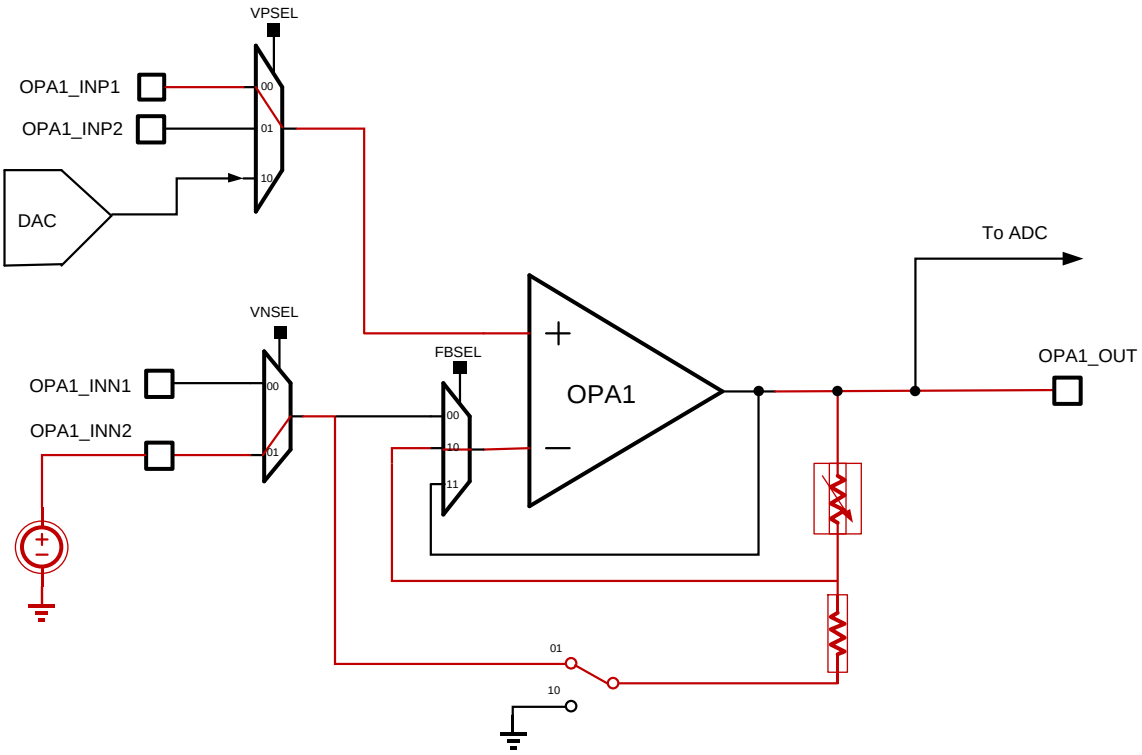


图 18-9 反相 PGA 带外部偏置

注意：使用反相PGA模式时，需要注意内部开关的阻抗和GPIO通道阻抗。这个阻抗典型值为90欧姆左右。当存在电流从OPA1_OUT流经开关输入信号源时，开关上的压降会对实际放大倍数造成影响。反向输入通道中，只有INN2通道支持反相PGA模式。

举例来说，选择反相放大倍数x15，则Rf=150Kohm，R1=10Kohm，开关阻抗90Ohm，IO通道阻抗100Ohm，则：

$$V_{out} = V_{DAC} + (V_{DAC} - V_{in}) \times \frac{Rf}{(R1 + R_{sw} + R_{IO})} \approx V_{DAC} + (V_{DAC} - V_{in}) \times 14.72$$

18.4.7 Offset 校准

校准功能用于抵消运放输入offset电压。为了避免封装和reflow应力影响，在对运放offset要求较高的场合，建议用户在reflow焊接后进行板级校准。校准完全由软件操作实现，软件通过调整输入差分对镜像电流的大小来补偿固有输入offset电压，并通过读取OPA输出来实现校准。

Offset校准支持自动校准和外部校准两种模式，其中自动校准模式下由芯片内部自行产生输入偏置电压，外部校准模式下需要用户提供外部偏置输入。

自动校准模式

用户需按照以下步骤，分别对运放的N输入差分对和P输入差分对进行校准。

N差分输入校准步骤：

- 配置VPSEL=11，VNSEL=10
- 置位TRIM_MODE
- 置位NCAL_EN
- 置位EN
- 软件读取OUT寄存器
  - 如果OUT=1，将NCAL寄存器写为000000并逐步递增，直到OUT从1变成0
  - 如果OUT=0，将NCAL寄存器写为100000并逐步递增，直到OUT从0变成1
- 校准完成，并保存NCAL数据

P差分输入校准步骤：

- 配置VPSEL=11，VNSEL=10
- 置位TRIM_MODE
- 置位PCAL_EN
- 置位EN
- 软件读取OUT寄存器
  - 如果OUT=1，将PCAL寄存器写为000000并逐步递增，直到OUT从1变成0
  - 如果OUT=0，将PCAL寄存器写为100000并逐步递增，直到OUT从0变成1
- 校准完成，并保存PCAL数据

注意：校准电路的典型调校步长是120uV，校准后可以实现小于+/-100uV的输入offset电压。

外部校准模式

校准原理与自动校准模式类似，仅需清零TRIM_MODE，并在OPA外部输入管脚上提供所需的偏置电压。偏置电压可以根据实际应用中的共模电压选择。

当应用共模电压小于GND+500mV时，将INN和INP端短接并输入共模电平，此时仅校准PCAL。

当应用共模电压大于VDDA-500mV时，将INN和INP端短接并输入共模电平，此时仅校准NCAL。

当应用共模电压介于GND+500mV和VDDA-500mV之间时，将INN和INP端短接并输入共模电平，分两次校准PCAL和NCAL。

18.4.8 低功耗模式

OPA1可以进入低功耗模式以显著降低运行功耗，此时运放的带宽大幅减小。

通过置位LPM寄存器使OPA1进入低功耗模式，此时运放单位增益带宽降低到50Khz，功耗降低到

2uA左右。

18.4.9 休眠模式下的 OPA

由于OPA电源由VDDA供电，没有低功耗限制，可以在任何休眠模式下保持工作，由软件配置决定。注意，OPA工作需要VREF1p2提供偏置电流，因此当OPA使能的情况下进入DeepSleep时，软件不能关闭VREF1p2。

18.5 寄存器

offset 地址	名称	符号
OPA(模块起始地址：0x4001A000)		
0x00	OPA1 控制寄存器 （OPA1 Control Register）	OPA1_CR
0x04	OPA1 校准寄存器 （OPA1 Calibration Register）	OPA1_CALR
0x08	OPA1 校准输出寄存器 （OPA1 Calibration Output Register）	OPA1_COR

18.5.1 OPA1 控制寄存器（OPA1_CR）

名称	OPA1_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	MSEL		-		VNSEL		VPSEL	
位权限	R/W-00		U-0		R/W-111		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	VN_EXC	PGA_GAIN		FBSEL		LPM	EN
位权限	U-0	R/W-0	R/W-00		R/W-00		R/W-0	R/W-0

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:14	MSEL	PGA 模式选择 (PGA Mode Select) 01：反馈电阻到 N 端输入开关导通 10：反馈电阻到地开关导通 00,11：RFU
13:12	-	RFU：未实现，读为 0
11:10	VNSEL	OPA1 负端输入选择 (OPA1 Negtive Input Select) 00：OPA1_INN1 01：OPA1_INN2 10,11：RFU *注意：使用反相 PGA 功能时，VNSEL 必须选择 01 通道*
9:8	VPSEL	OPA1 正端输入选择 (OPA1 Positive Input Select) 00：OPA1_INP1 01：OPA1_INP2 10：DAC 11：RFU
7	-	RFU：未实现，读为 0

位号	助记符	功能描述
6	VN_EXC	OPA1 负端连接 GPIO，仅 FBSEL=10 时有效 (OPA1 Negtive Input Connected to GPIO enable) 0：PGA 模式下 OPA1 负端不连接 GPIO 1：PGA 模式下 OPA1 负端同时连接到 GPIO
5:4	PGA_GAIN	PGA 增益选择 (PGA gain select) 同相 PGA 配置下，放大倍数定义如下 00：PGA 增益 x2 01：PGA 增益 x4 10：PGA 增益 x8 11：PGA 增益 x16 反相 PGA 配置下，放大倍数定义如下： 00：PGA 增益 x1 01：PGA 增益 x3 10：PGA 增益 x7 11：PGA 增益 x15
3:2	FBSEL	OPA1 工作模式 (Feedback Select) 00：standalone 模式 01：RFU 10：PGA 模式 11：buffer 模式
1	LPM	OPA1 低功耗控制寄存器 (OPA1 low power mode) 0：正常模式 1：低功耗模式
0	EN	OPA1 使能寄存器 (OPA1 enable) 0：关闭 OPA1 1：使能 OPA1

18.5.2 OPA1 校准寄存器（OPA1_CALR）

名称	OPA1_CALR							
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	NCAL							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PCAL							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					NCAL_EN	PCAL_EN	TRIM_MODE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:24	-	RFU：未实现，读为 0

位号	助记符	功能描述
23:16	NCAL	OPA1 负输入端校准 trim 信号，最高位为符号位 (OPA1 negative input calibration) 01111111：输出电压减小最大 00000001：输出电压减小最小 00000000：输出电压不变 10000000：输出电压不变 10000001：输出电压增加最小 11111111：输出电压增加最大
15:8	PCAL	OPA1 正输入端校准 trim 信号，最高位为符号位 (OPA1 positive input calibration) 01111111：输出电压减小最大 00000001：输出电压减小最小 00000000：输出电压不变 10000000：输出电压不变 10000001：输出电压增加最小 11111111：输出电压增加最大
7:3	-	RFU：未实现，读为 0
2	NCAL_EN	N 差分对校准使能 0：关闭 N 校准 1：使能 N 校准
1	PCAL_EN	P 差分对校准使能 0：关闭 P 校准 1：使能 P 校准
0	TRIM_MODE	校准模式选择 0：外部校准模式 1：自动校准模式

18.5.3 OPA1 校准输出寄存器（OPA1_COR）

名称	OPA1_COR							
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	-							OUT
位权限	U-0							R-x

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	OUT	校准模式下 OPA 的输出。软件通过轮询这个信号来判断 offset 校准是否成功。

19 模拟比较器（Comparator）

19.1 概述

芯片集成3个比较器，支持以下特性：

- 2个rail-to-rail比较器，支持轨到轨输入，多种功耗模式
- 1个低功耗比较器，典型功耗200nA
- 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

19.2 结构框图

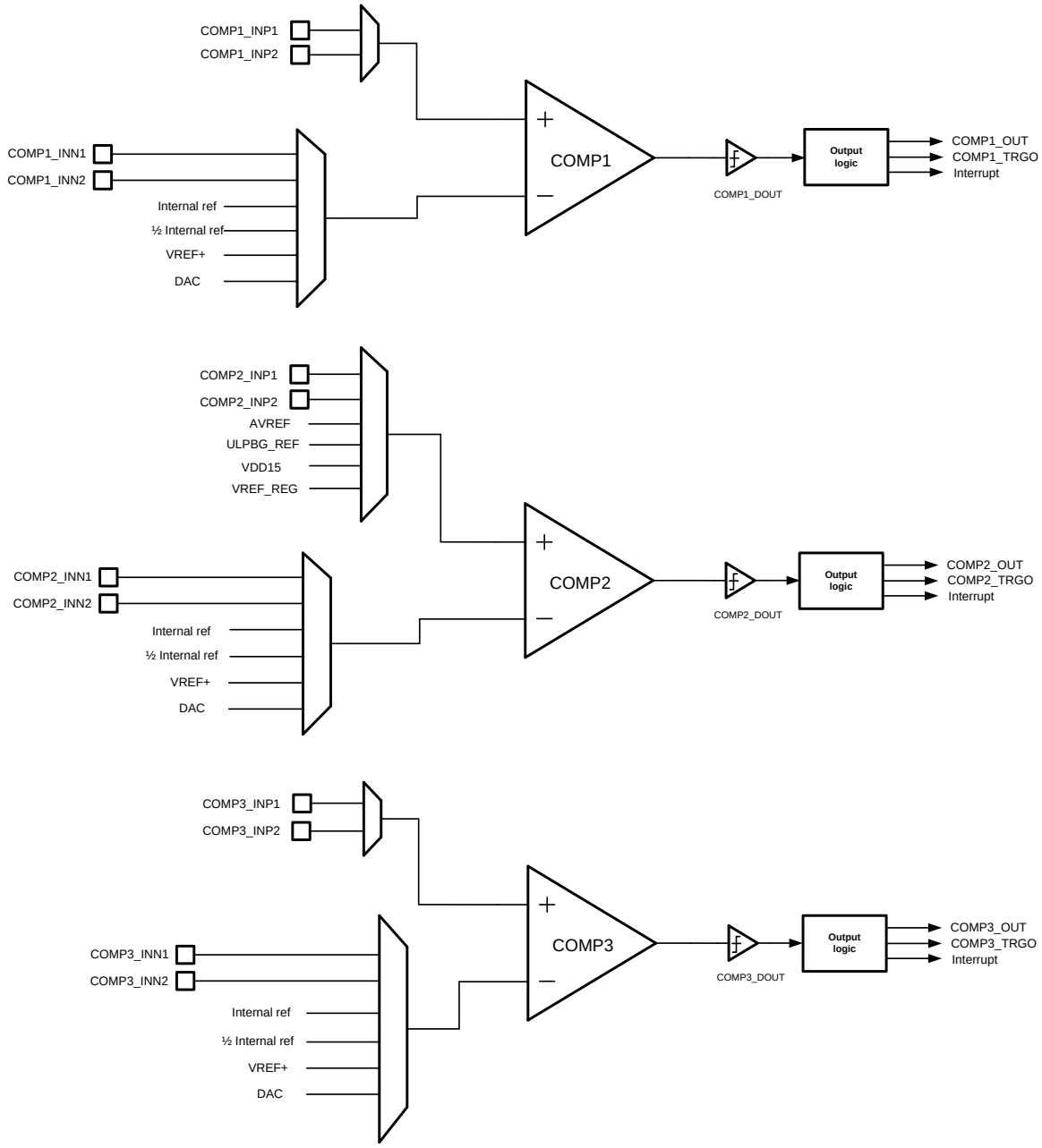


图 19-1 比较器电路框图

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，internal ref来自于AVREF或VREF1p2，软件可以选择BUFFER的输入源。如果比较基准为AVREF或VREF1p2本身，则可以关闭并bypass BUFFER节省功耗。比较器1、2完全相同，输入端连接方式如结构图所示，两个比较器产生输出信号输出至数字电路。

比较器1、2输入电压范围0~VDD，建立时间小于15us，典型传输延迟小于1us，典型功耗2uA。

比较器3输入电压范围0~VDD-0.7V，建立时间小于10us，传输延迟小于5us，典型功耗200nA。

19.3 功能描述

19.3.1 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。

19.3.2 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VREF1p2及其分压、AVREF及其分压、也可以选择使用DAC输出电压作为比较基准。

当使用VREF1p2和AVREF分压时，必须使能分压buffer并选择buffer输入源。通过BUFSEL和BUFENB寄存器来使能比较器模块内建的基准源buffer，并选择输入基准。

如果置位BUFENB并且置位BUFBYP，则基准电压bypass BUFFER后直接输出到internal ref；注意此时不能使用1/2 internal ref作为比较器输入。

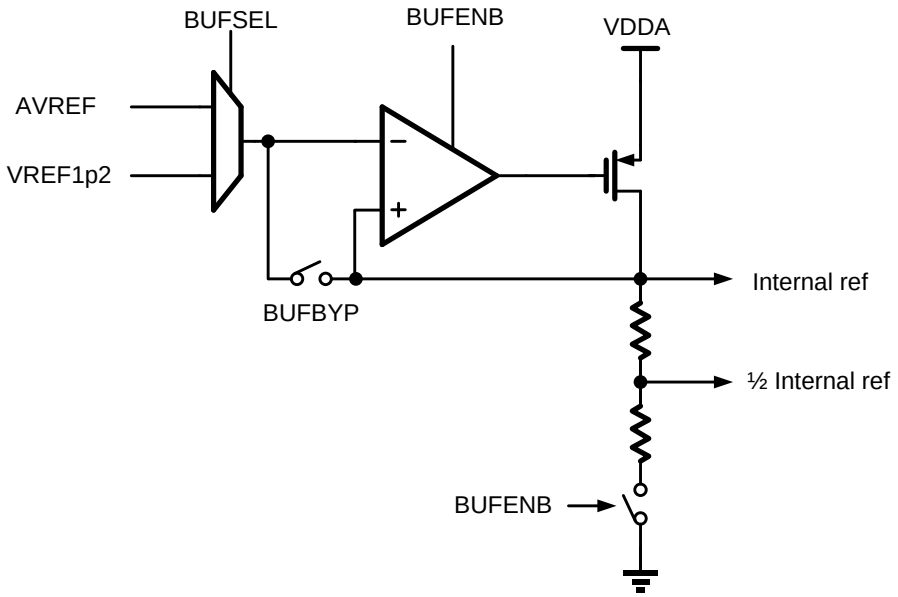


图 19-2 比较器内建基准缓冲器

19.3.3 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。

19.3.4 引脚和内部信号连接

比较器输入信号可以映射到以下IO引脚上，使用外部引脚输入时需要将对应的GPIO配置为模拟通道功能。

COMP1	引脚	寄存器
COMP1_INP1	PB8	COMP1_CR.V1PSEL=00
COMP1_INP2	PB9	COMP1_CR.V1PSEL=01
COMP1_INN1	PA10	COMP1_CR.V1NSEL=000
COMP1_INN2	PC6	COMP1_CR.V1NSEL=001
COMP1_OUT	PC4	-

表 19-1 比较器 1 引脚列表

COMP2	引脚	寄存器
COMP2_INP1	PA8	COMP2_CR.V2PSEL=000
COMP2_INP2	PA9	COMP2_CR.V2PSEL=001
COMP2_INN1	PA4	COMP2_CR.V2NSEL=000
COMP2_INN2	PA5	COMP2_CR.V2NSEL=001
COMP2_OUT	PC5	-

表 19-2 比较器 2 引脚列表

COMP3	引脚	寄存器
COMP3_INP1	PC0	COMP3_CR.V3PSEL=00
COMP3_INP2	PC1	COMP3_CR.V1PSEL=01
COMP3_INN1	PB4	COMP3_CR.V3NSEL=000
COMP3_INN2	PB5	COMP3_CR.V3NSEL=001
COMP3_OUT	PE7	-

表 19-3 比较器 3 引脚列表

19.3.5 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。

当WINMODE寄存器置位后，下图中的开关将COMP2的正端输入切换到COMP1的正端输入信号上。

窗口比较器结构如下图所示：

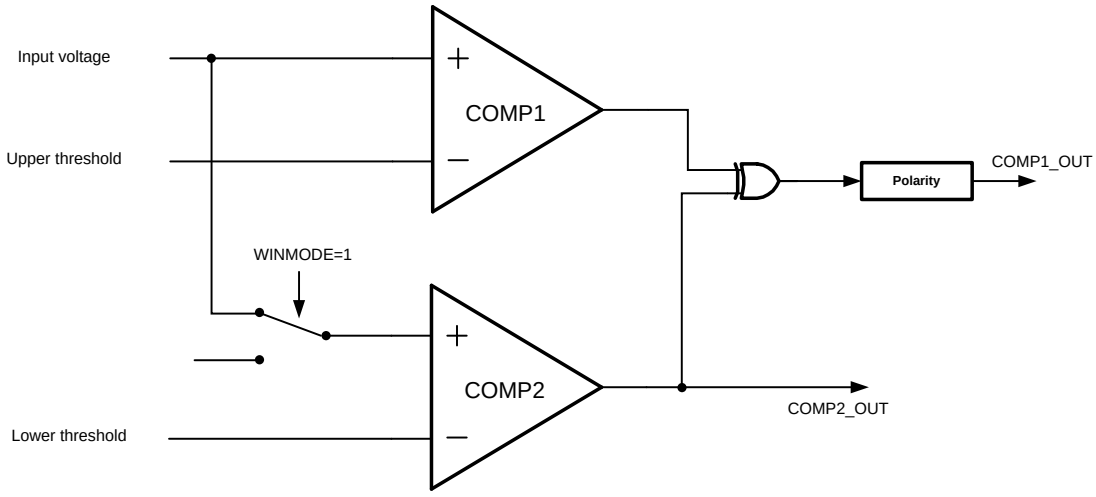


图 19-3 窗口比较器框图

窗口比较器工作时的波形示意图如下。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。

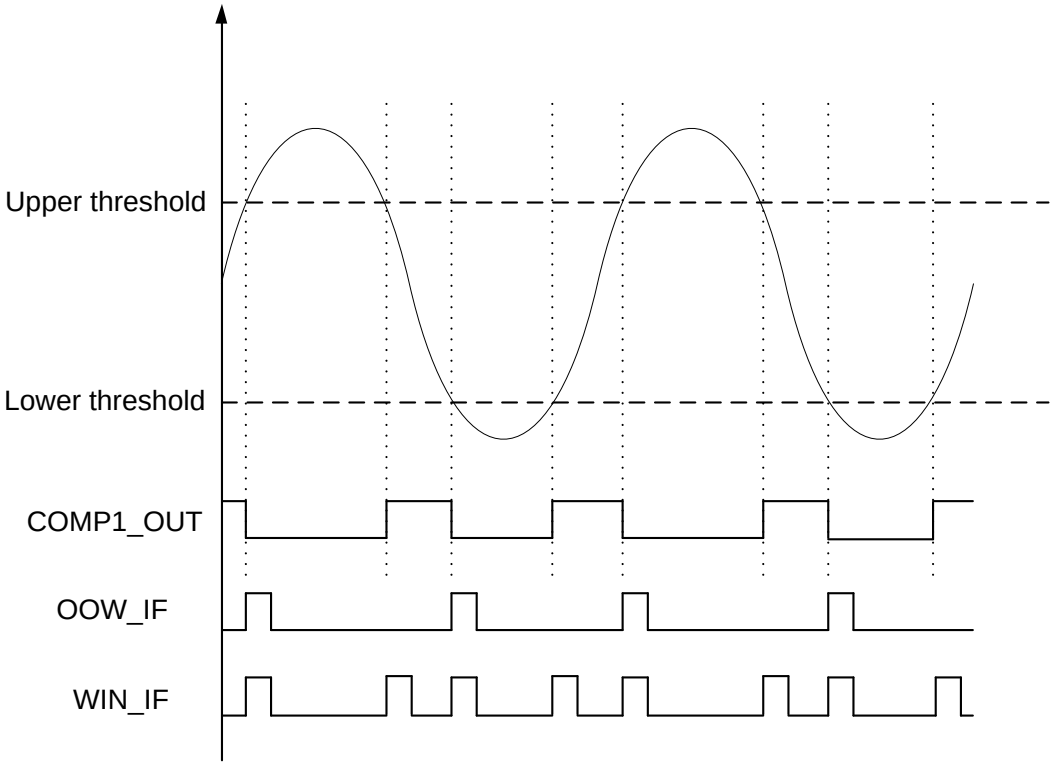


图 19-4 窗口比较器波形示意图

窗口模式中断标志的置位可以选择使用经过数字滤波或不经过数字滤波的COMP1输出信号。当输入信号变化非常缓慢时，在比较阈值附近比较器输出可能频繁翻转，通过数字滤波可以避免OOW_IF和WIN_IF寄存器反复置位。

如果在休眠模式下仍希望使用窗口功能，需要关闭比较器输出的数字滤波功能。

19.3.6 功耗和速度模式

比较器1和2支持轨到轨输入，输入电压范围0~VDD，支持低功耗模式、中速模式和高速模式，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		50ns		所有芯片功耗模式下都可以使用
中速模式	<2uA	0.6us	5us	
低功耗模式	<1uA	2us	15us	

表 19-4 比较器 1/2 工作模式

比较器3为低功耗比较器，输入电压范围0~VDD-0.7V，建立时间小于10us，传输延迟小于5us，典型功耗200nA。

模式	典型功耗	典型传输延迟	典型建立时间	使用条件
低功耗模式	0.2uA	5us	10us	所有芯片功耗模式下都可以使用

表 19-5 比较器 3 工作模式

19.3.1 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。

在窗口比较器模式下，根据输入模拟信号的电压范围，可以产生窗口阈值中断和出窗口中断。

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。

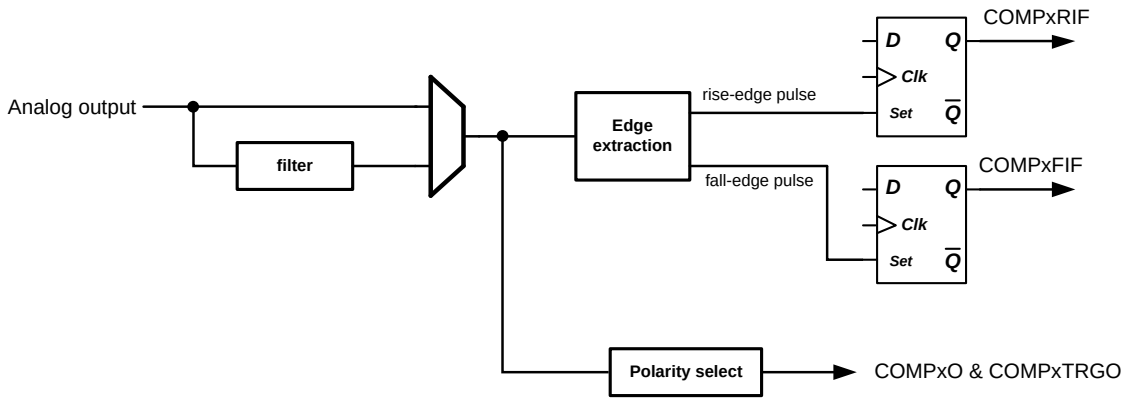


图 19-5 比较器中断产生

19.3.2 比较器输出和触发输出

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。

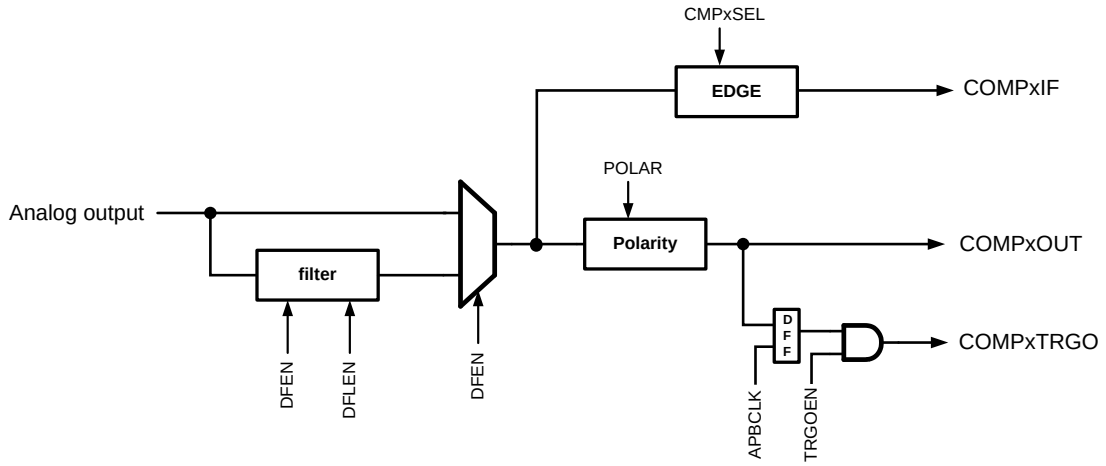


图 19-6 比较器输出逻辑

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。

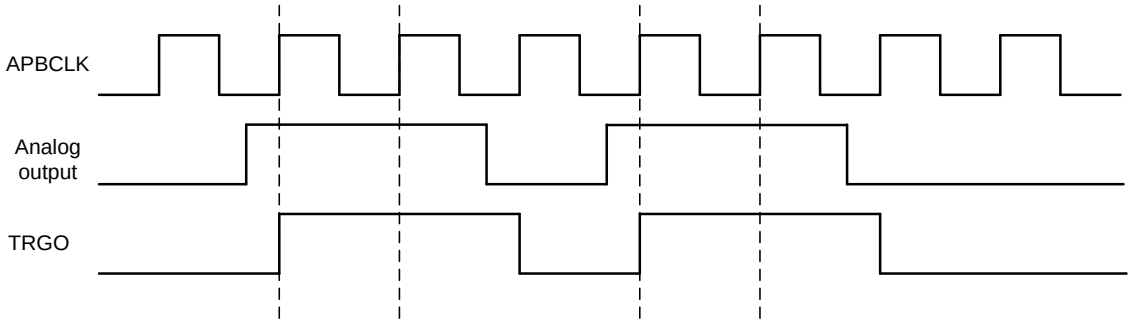


图 19-7 比较器输出上升沿产生触发输出，无滤波

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。具体连接关系请参考30.4.4内部触发信号（ITRx）的捕捉。

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38.6.11转换触发。

19.3.3 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：

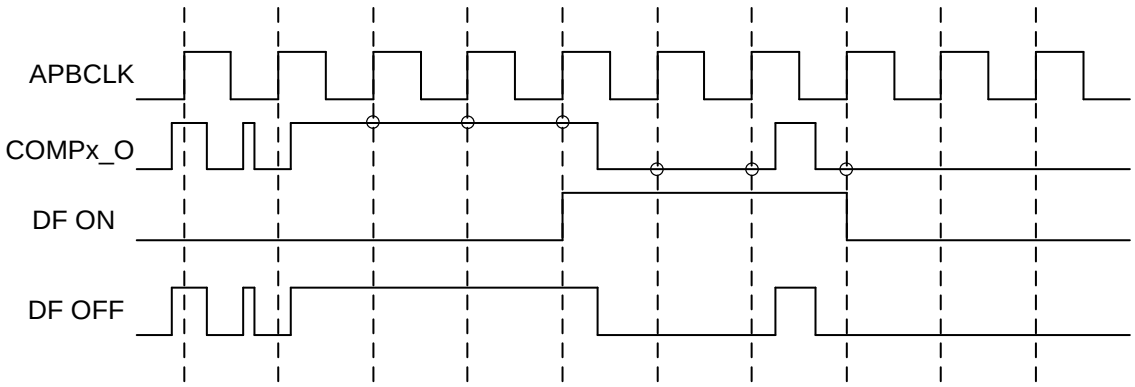


图 19-8 数字滤波（DFLEN=3）波形示意图

19.4 寄存器

offset 地址	名称	符号
COMP(模块基地址：0x40015400)		
0x00	COMP1 控制寄存器 （ComparatorControl Register 1）	COMP1_CR
0x04	COMP2 控制寄存器 （Comparator Control Register 2）	COMP2_CR
0x08	COMP3 控制寄存器 （Comparator Control Register 3）	COMP3_CR
0x0C	COMP 中断配置寄存器 （Comparator Interrupt Config Register）	COMP_ICR
0x10	COMP 中断标志寄存器 （Comparator Interrupt Status Register）	COMP_ISR
0x14	COMP 缓冲器控制寄存器 （Comparator Buffer Control Register）	COMP_BUFCR

19.4.1 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	-							TRGOEN
位权限	U-0							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		-					CMP1O
位权限	R/W-00		U-0					R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		V1PSEL		V1NSEL			CMP1EN
位权限	U-0		R/W-00		R/W-00			R/W-0

位号	助记符	功能描述
31:25	--	RFU：未实现，读为 0
24	TRGOEN	比较器 1 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 …… 11111：32 次 APBCLK 采样

位号	助记符	功能描述
18	WINMODE	比较器 1&2 窗口模式控制寄存器 0：禁止窗口模式 1：使能窗口模式
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：中速模式 10：高速模式
13:9	--	RFU：未实现，读为 0
8	CMP1O	比较器 1 输出，软件只读
7:6	--	RFU：未实现，读为 0
5:4	V1PSEL	比较器 1 正极输入选择 00：COMP1_INP1 01：COMP1_INP2 10：RFU 11：RFU
3:1	V1NSEL	比较器 1 负极输入选择 000：COMP1_INN1 001：COMP1_INN2 010：internal reference 011：1/2 (internal reference) 100：VREF+ 101：DAC 110：RFU 111：RFU
0	CMP1EN	比较器 1 使能位 0：关闭比较器 1 1：使能比较器 1

19.4.2 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	-							TRGOEN
位权限	U-0							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		-					CMP2O
位权限	R/W-00		U-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

位号	助记符	功能描述
31:25	--	RFU：未实现，读为 0
24	TRGOEN	比较器 2 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 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：中速模式 10：高速模式
13:9	--	RFU：未实现，读为 0
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：AVREF 011：ULPBG_VREF(1.16V) 100：VDD15 101：VREF_VREG
3:1	V2NSEL	比较器 2 负极输入选择 000：COMP2_INN1 001：COMP2_INN2 010：internal reference 011：1/2 (internal reference) 100：VREF+ 101：DAC 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

19.4.3 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	-							TRGOEN
位权限	U-0							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	-							CMP3O
位权限	U-0							R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		V3PSEL		V3NSEL			CMP3EN
位权限	U-0		R/W-00		R/W-000			R/W-0

位号	助记符	功能描述
31:25	--	RFU：未实现，读为 0
24	TRGOEN	比较器 3 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 3 输出数字滤波长度配置寄存器。滤波长度周期是 DFLEN+1 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 3 输出极性控制 0：正向输出 1：取反输出
16	DFEN	比较器 3 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
15:9	--	RFU：未实现，读为 0
8	CMP3O	比较器 3 输出，软件只读
7:6	--	RFU：未实现，读为 0
5:4	V3PSEL	比较器 3 正极输入选择 00：COMP3_INP1 01：COMP3_INP2 10,11：RFU
3:1	V3NSEL	比较器 3 负极输入选择 000：COMP3_INN1 001：COMP3_INN2 010：internal reference

位号	助记符	功能描述
		011：1/2 (internal reference) 100：VREFBUF 101：DAC 110：RFU 111：RFU
0	CMP3EN	比较器 3 使能位 0：关闭比较器 1 1：使能比较器 1

19.4.4 COMP 中断配置寄存器（COMP_ICR）

名称	COMP_ICR							
offset	0x0C							
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	-				CMP3SEL		-	CMP3IE
位权限	U-0				R/W-00		U-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				CMP2SEL		-	CMP2IE
位权限	U-0				R/W-00		U-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				CMP1SEL		-	CMP1IE
位权限	U-0				R/W-00		U-0	R/W-0

位号	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
23:20	--	RFU：未实现，读为 0
19:18	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
17	--	RFU：未实现，读为 0
16	CMP3IE	比较器 3 中断使能，1 有效
15:12	--	RFU：未实现，读为 0
11:10	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
9	--	RFU：未实现，读为 0
8	CMP2IE	比较器 2 中断使能，1 有效
7:4	--	RFU：未实现，读为 0
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

位号	助记符	功能描述
		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。

19.4.5 COMP 中断标志寄存器（COMP_IF）

名称	COMP_IF							
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	-			OOW_IF	WIN_IF	CMP3IF	CMP2IF	CMP1IF
位权限	U-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
31:5	--	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	CMP3IF	比较器 3 中断标志，硬件置位，软件写 1 清零
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

19.4.6 COMP 缓冲器控制寄存器（COMP_BUFCR）

名称	COMP_BUFCR							
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	-					BUFBYP	BUFSEL	BUFENB

位权限	U-0	R/W-0	R/W-0	R/W-1

位号	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	BUFBYP	缓冲器 bypass 使能 0：不 bypass 1：bypass
1	BUFSEL	缓冲器输入基准选择 0：AVREF 1：VREF1p2
0	BUFENB	缓冲器使能 0：使能基准缓冲器 1：关闭基准缓冲器

20 除法/开方运算加速器（DIVAS）

20.1 概述

DIVAS模块用于帮助软件加速除法和开方运算。该硬件电路包含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16bit余数；以及一个无符号整数开方电路，输入32bit无符号数，输出16bit开方运算结果。

基本特性：

- 有符号整数除法运算（二进制补码格式）
- 32bit被除数、16bit除数
- 输出32bit商和16bit余数
- 除以0警告
- 一次除法计算需要16个时钟周期
- 无符号整数开方运算
- 32bit被开方数，16bit结果
- 一次开方运算需要16个时钟周期

20.2 时钟和复位

DIVAS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DIVAS模块寄存器之前必须在RMU模块中清除DIVAS复位（DIVASRST），并在CMU中使能DIVAS时钟（DIVAS_PCE）。

20.3 硬件除法工作流程

软件按照如下步骤调用硬件除法器。

- 清零MODE寄存器
- 向OPRD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）
- 写入DIVSOR后硬件除法器自动开始工作，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

20.4 硬件开方工作流程

软件按照如下步骤调用硬件开方器。

- 置位MODE寄存器
- 向OPRD寄存器写入32bit被开方数
- 硬件开方器自动开始工作，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
- 读取ROOT寄存器中的根

20.5 寄存器

模块基地址：0x40019C00

offset 地址	名称	符号
DIVAS(模块基地址：0x40019C00)		
0x00	操作数寄存器 （Operand Register）	DIVAS_OPRD
0x04	除数寄存器 （Divisor Regsiter）	DIVAS_DIVSOR
0x08	商寄存器 （Quotient Register）	DIVAS_QUOT
0x0C	余数寄存器 （Reminder Register）	DIVAS_REMD
0x10	根寄存器 （Root Register）	DIVAS_ROOT
0x14	状态标志寄存器 （Status Register）	DIVAS_SR
0x18	控制寄存器 （Control Register）	DIVAS_CR

20.5.1 操作数寄存器

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

位号	位名	说明
31:0	OPRD	操作数寄存器 在除法运算时保存 32bit 有符号被除数 在开方运算时保存 32bit 无符号被开方数

20.5.2 除数寄存器

名称	DIVAS_DIVSOR

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	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

20.5.3 商寄存器

名称	DIVAS_QUOT							
offset	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 DW_div 模块输出）

20.5.4 余数寄存器

名称	DIVAS_REMD							
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	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）

20.5.5 根寄存器

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

位号	位名	说明
31:16	--	RFU：未实现，读为 0
15:0	ROOT	16bit 无符号开方根 （仅地址，无实际寄存器，读取时直接返回 DW_sqrt 模块输出）

20.5.6 状态标志寄存器

名称	DIVAS_SR							
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	-						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 = DIVAS 在计算过程中，结果未就绪 0 = 计算完毕，结果就绪 软件在写入操作数或除数后，DIVAS 开始计算，软件应查询 BUSY 为低后再读取商、余数或根寄存器 单次计算需要 16 个时钟周期

20.5.7 控制寄存器

名称	DIVAS_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	-							MODE
位权限	U-0							R/W-0

位号	位名	说明
31:1	--	RFU：未实现，读为 0
0	MODE	工作模式控制 0：硬件除法器 1：硬件开方器

21 双线串行总线（I²C）

21.1 概述

I²C 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 的主机和从机模式，不支持多主机模式。

特点：

- 1 路独立 I2C 接口
- 支持主机和从机模式，不支持多主机模式
- 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)
- 支持 DMA，主机和从机独立 DMA 通道
- 低功耗从机设计，可以在没有系统时钟的情况下收发数据
- 支持异步从机地址匹配唤醒、数据帧接收完成唤醒或 START 检测唤醒

21.2 结构框图

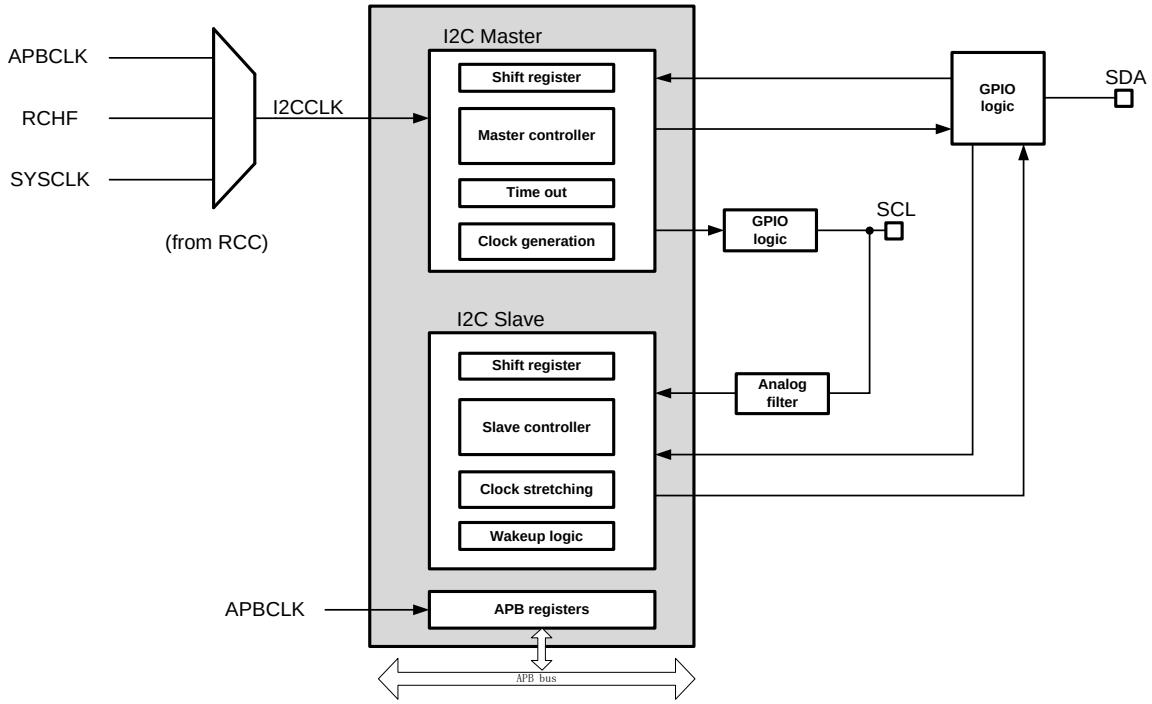


图 21-1 I2C 模块框图

21.3 引脚定义和上拉电阻范围

I2C 模块使用 2 个真开漏引脚与外部器件通信，工作时需要外接上拉电阻：

引脚	I2Cx	符号	功能	引脚类型
PA11	I2C	SCL0	I2C 时钟	High-sink(20mA)
PA12		SDA0	I2C 数据	
PB15		SCL1	I2C 时钟	Normal
PD12		SDA1	I2C 数据	

表 21-1 I2C 引脚列表

I2C 总线协议规定了 standard-mode、fast-mode 和 fast-mode plus 的信号最大上升时间，以及 IO 能够支持的最小 sink 电流。参见下表。

Symbol	Parameter	Conditions	Standard-mode		Fast-mode		Fast-mode Plus		Unit
			Min	Max	Min	Max	Min	Max	
V_{IL}	LOW-level input voltage[1]		0.5	$0.3V_{DD}$	0.5	$0.3V_{DD}$	0.5	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage[1]		$0.7V_{DD}$	[2]	$0.7V_{DD}$	[2]	$0.7V_{DD}$[1]	[2]	V
V_{hys}	hysteresis of Schmitt trigger inputs		-	-	$0.05V_{DD}$	-	$0.05V_{DD}$	-	V
V_{OL1}	LOW-level output voltage 1	(open-drain or open-collector) at 3 mA sink current; V_{DD}>2V	0	0.4	0	0.4	0	0.4	V
V_{OL2}	LOW-level output voltage 2	(open-drain or open-collector) at 2 mA sink current[3]; V_{DD} 2V	-	-	0	$0.2V_{DD}$	0	$0.2V_{DD}$	V
I_{OL}	LOW-level output current	V_{OL} = 0.4 V	3	-	3	-	20	-	mA
		V_{OL} = 0.6 V[4]	-	-	6	-	-	-	mA
t_{of}	output fall time from $V_{Ihmin}$to V_{ILmax}		-	250[5]	20 (V_{DD} / 5.5 V)[6]	250[5]	20 (V_{DD} / 5.5 V)[6]	120[7]	ns

t_{SP}	pulse width of spikes that must be suppressed by the input filter		-	-	0	50[8]	0	50[8]	ns
I_i	input current each I/O pin	$0.1V_{DD} < V_I < 0.9V_{Ddmax}$	10	+10	10[9]	+10[9]	10[9]	+10[9]	A
C_i	capacitance for each I/O pin[10]		-	10	-	10	-	10	pF

下表则定义了总线信号允许的最大上升时间和下降时间。

Symbol	Parameter	Conditions	Standard-mode		Fast-mode		Fast-mode Plus		Unit
			Min	Max	Min	Max	Min	Max	
f_{SCL}	SCL clock frequency		0	100	0	400	0	1000	kHz
$t_{HD;STA}$	hold time (repeated) START condition	After this period, the first clock pulse is generated.	4.0	-	0.6	-	0.26	-	s
t_{LOW}	LOW period of the SCL clock		4.7	-	1.3	-	0.5	-	s
t_{HIGH}	HIGH period of the SCL clock		4.0	-	0.6	-	0.26	-	s
$t_{SU;STA}$	set-up time for a repeated START condition		4.7	-	0.6	-	0.26	-	s
$t_{HD;DAT}$	data hold time[2]	CBUS compatible masters (see Remark in Section 4.1)	5.0	-	-	-	-	-	s
		I²C-bus devices	0[3]	-[4]	0[3]	-[4]	0	-	s
$t_{SU;DAT}$	data set-up time		250	-	100[5]	-	50	-	ns
t_r	rise time of both SDA and SCL signals		-	1000	20	300	-	120	ns
t_f	fall time of both SDA and SCL signals[3][6][7][8]		-	300	20 (V_{DD} / 5.5 V)	300	20 (V_{DD} / 5.5 V)[9]	120[8]	ns
$t_{SU;STO}$	set-up time for STOP condition		4.0	-	0.6	-	0.26	-	s
t_{BUF}	bus free time between a STOP and START condition		4.7	-	1.3	-	0.5	-	s
C_b	capacitive load for each bus line[10]		-	400	-	400	-	550	pF
$t_{VD;DAT}$	data valid time[11]		-	3.45[4]	-	0.9[4]	-	0.45[4]	s
$t_{VD;ACK}$	data valid acknowledge time[12]		-	3.45[4]	-	0.9[4]	-	0.45[4]	s
V_{nL}	noise margin at the LOW level	for each connected device (including hysteresis)	$0.1V_{DD}$	-	$0.1V_{DD}$	-	$0.1V_{DD}$	-	V
V_{nH}	noise margin at the HIGH level	for each connected device (including hysteresis)	$0.2V_{DD}$	-	$0.2V_{DD}$	-	$0.2V_{DD}$	-	V

根据以上协议规范，我们可以计算外部上拉电阻的合理范围。

假设总线信号从 VIL=0.3VDD 上升到 VIH=0.7VDD，则充电时间可以计算为：

V(t1) = 0.3 V_{DD} = V_{DD} (1 e $^{t1 / RC}$); t1 = 0.3566749 RC

V(t2) = 0.7 V_{DD} = V_{DD} (1 e $^{t2 / RC}$); t2 = 1.2039729 RC

T = t2 t1 = 0.8473 RC

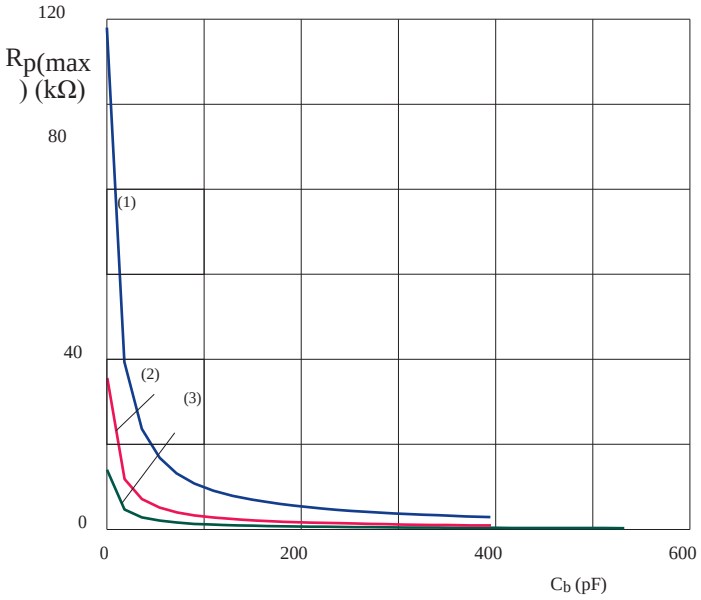
根据总线容性负载大小和协议对信号最大上升时间的要求，我们可以计算上拉电阻的最大值：

$$R_{p(\max)} = \frac{t_r}{0.8473 \times C_b}$$

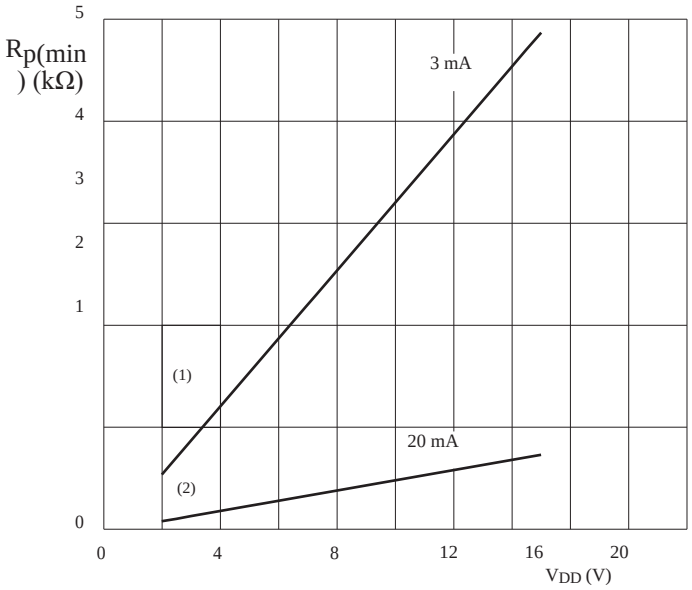
而上拉电阻的最小值则由总线电源电压 VDD 和 IO 电流 sink 能力决定，FM33LG0 的 I2C 引脚 sink 能力是 20mA，而协议要求的最小 sink 能力是标准/快速模式 3mA，Fm+模式 20mA。

$$R_{p(\min)} = \frac{VDD - V_{OL(\max)}}{I_{OL}}$$

根据以上计算可以得到推荐的上拉电阻阻值范围，参见下图。



(1) Standard-mode
(2) Fast-mode
(3) Fast-modePlus



(1) Fast-mode andStandard-mode
(2) Fast-modePlus

21.4 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。参见 14.11.15 外设总线时钟控制寄存器 3。
- 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、RCLF，能够独立于 APBCLK 工作。必须使能 I2CCLK 才能进行数据收发。
- 从机的数据收发时钟使用 SCL 总线时钟输入，因此无需系统时钟就可以进行数据收发

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。

模块工作前需先清除 RMU 模块中的 I2C 复位寄存器（I2CRST）。

21.5 接口时序

21.5.1 接口时序图

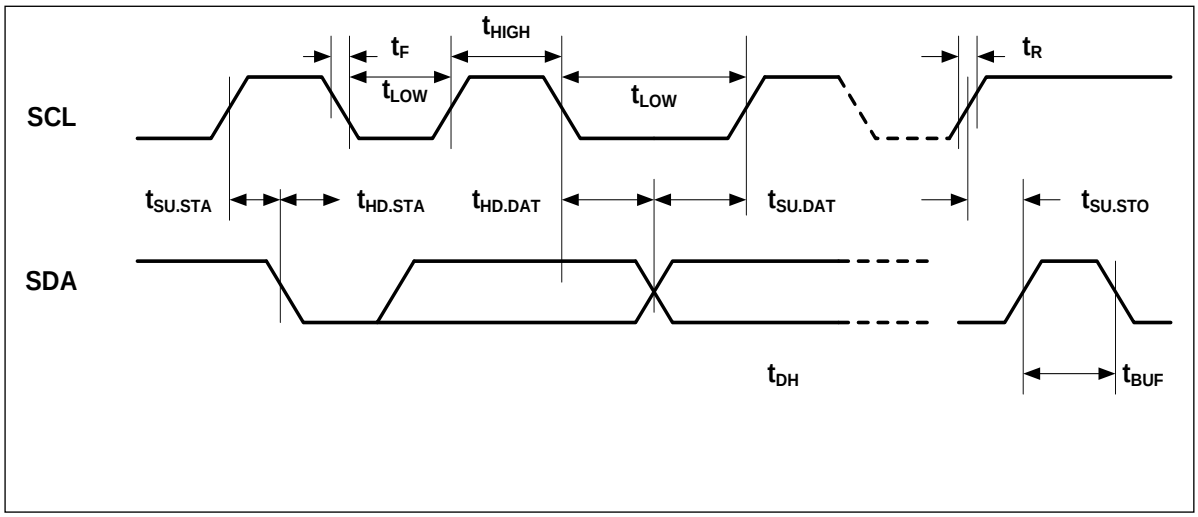


图 21-2I²C 总线时序

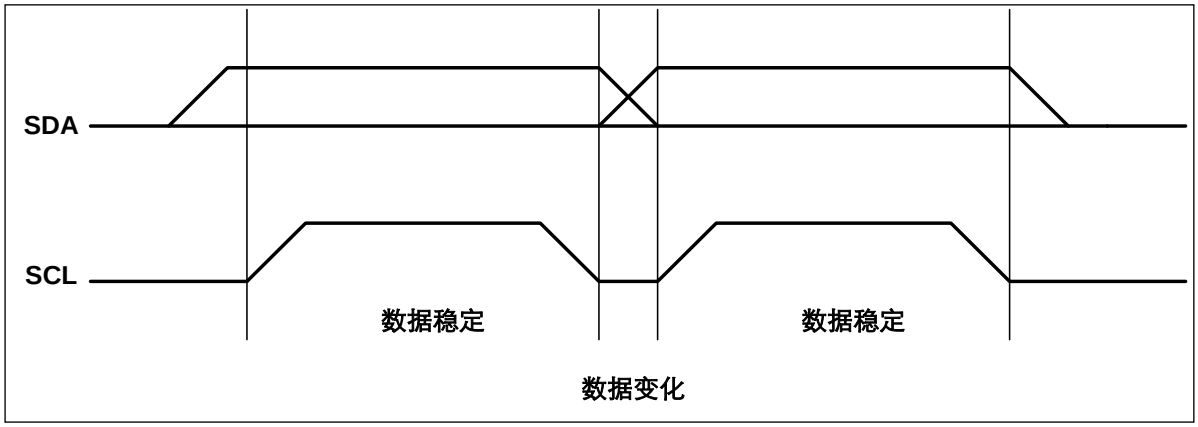


图 21-3 数据有效时序

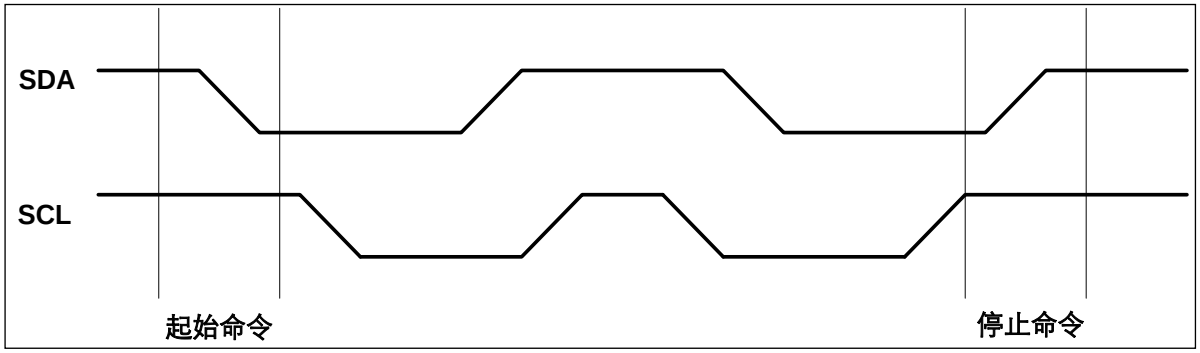


图 21-4 起始（Start）与停止(Stop)命令定义

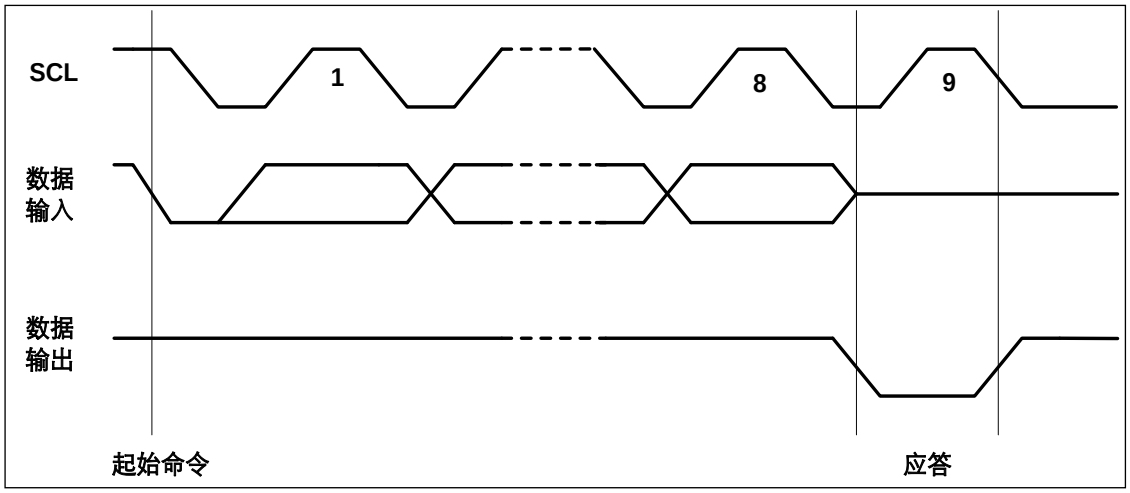


图 21-5 输出应答(ACK)

21.5.2 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 21-3）；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 21-4）。

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 EEPROM 进入等待态低功耗模式（参见图 21-4）。

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 21-5)。

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

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	
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 21-2I²C 接口时序要求

21.6 I²C 工作模式

I2C模块支持以下工作模式：

- 主机接收
- 主机发送
- 从机接收
- 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，或设置SSPEN来使能从机通信。

主机和从机不能同时工作，因为他们复用相同的IO引脚作为SCL和SDA，原则上禁止软件同时将MSPEN和SSPEN置1。

21.7 I²C 从机地址格式

I2C总线协议定义了以下保留地址，对其中多数保留地址，I2C从机硬件不做合法性判断，软件可以根据收到的地址进行自定义的处理。

但是对于10bit从机地址应用，即SSPCON.A10EN=1的情况下，要求1st字节必须以11110开头，否则将触发ADDR_ERROR错误标志。而在SSPCON.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
1111 1XX	X	Reserved for future purpose
1111 0XX	X	10bit slave addressing

表 21-3 I2C 从机保留地址定义

21.8 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
- 配置RCC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）
- 根据需要配置模拟滤波使能（SCL和SDA输入模拟滤波，>50ns）

21.8.1 IO 配置

FM33LG0 最多有两组引脚用于数据传输，开始 I2C 通信前需将对应引脚的 FCR 寄存器设置为 AF：

SDA：PA12/PD12

SCL：PA11/PB15

注意，如果 PA11 和 PB15 同时配置为 SDA 功能，则 PA11 被连接到 I2C 模块，PB15 无效；如果 PA12 和 PD12 同时配置为 SCL 功能，则主机模式下两个引脚都会输出 SCL 信号，从机模式下只有 PA12 被连接到 I2C 从机的 SCL 输入。

PA11 和 PA12 是强驱动真 OD 引脚，必须搭配外部总线上拉电阻使用，并且具备 20mA sink 电流能力，能够支持 Fm+模式。

21.8.2 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。

MSPBRGH和MSPBRGL波特率配置寄存器用于产生通信波特率。MSPBRGH和MSPBRGL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MSPBRGH定义SCL高电平宽度，MSPBRGL定义SCL低电平宽度

T_{BRGH}＝T_{I2CCLK}×(MSPBRGH+1)

T_{BRGL}＝T_{I2CCLK}×(MSPBRGL+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MSPBRGH=MSPBRGL=39

21.8.3 从机的输入模拟滤波和输出延迟

模拟滤波功能仅针对 SCL 引脚，并且只有从机的 SCLi 输入信号上可以使能模拟滤波功能。

同时，从机的 SDA 输出延迟，通过在 SDAo 上增加大于 300ns 的模拟延迟，来确保 SDA 相对于 SCL 下降沿的输出保持时间。

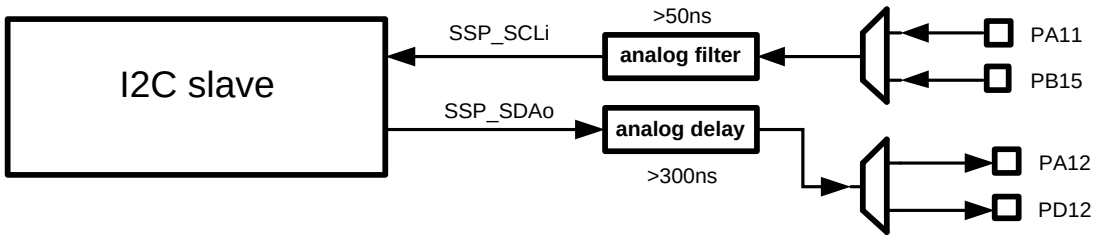


图 21-6 从机信号滤波

21.9 I²C 主机功能

FM33LG0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。

21.9.1 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。

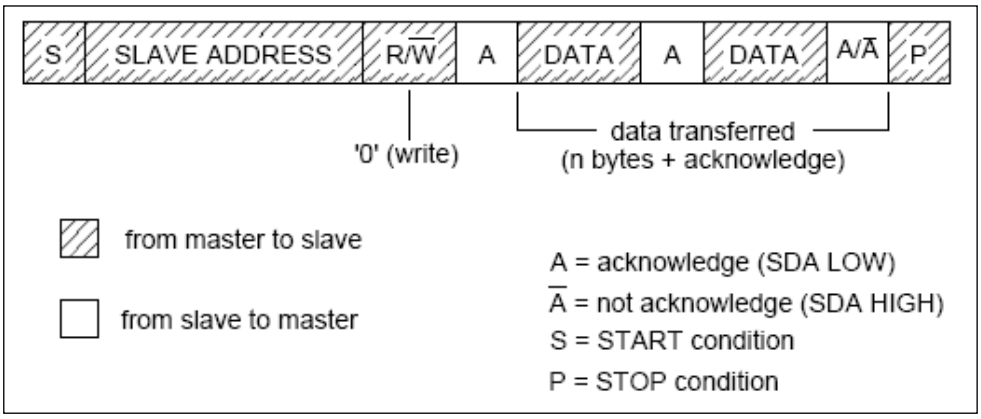


图 21-7 主机向 7 位地址从机写入数据时的帧格式

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：

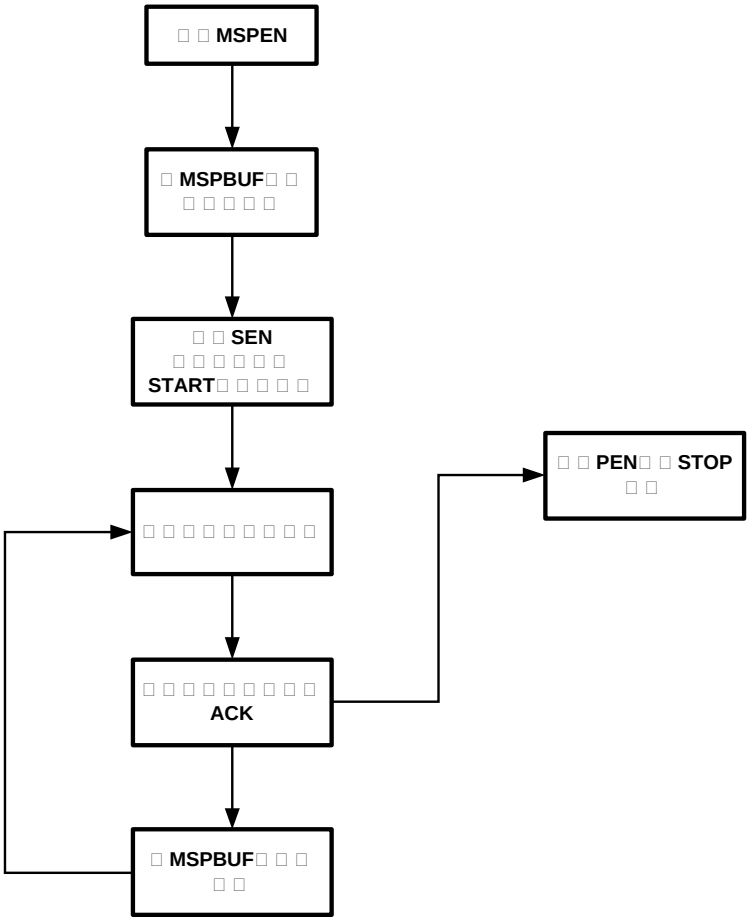


图 21-8I2C 软件发送数据流图

I2C主机对7位地址从机写入数据的波形示意图如下：

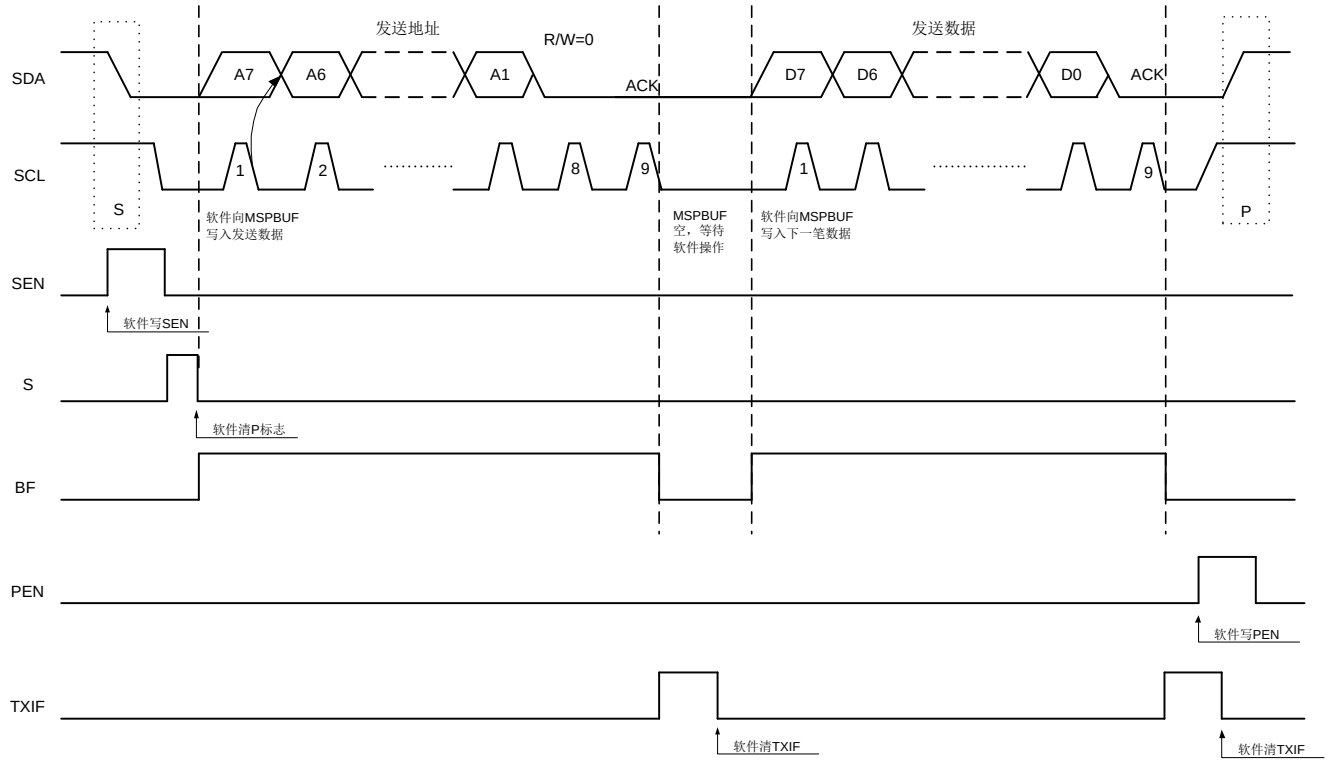


图 21-9 I2C 主机对 7 位地址从机发送数据流图

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。

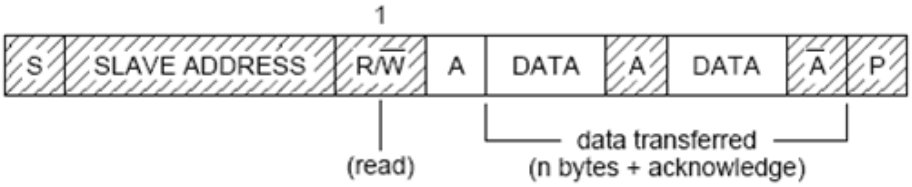


图 21-10 主机从 7 位地址从机读取数据时的帧格式

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：

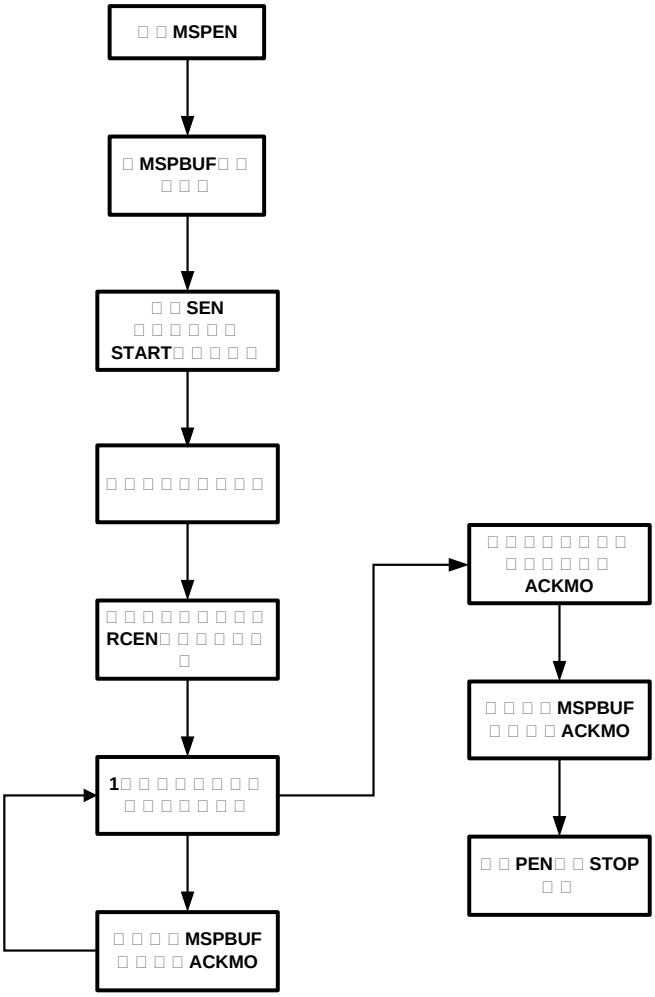


图 21-11I2C 软件发送数据流图

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：

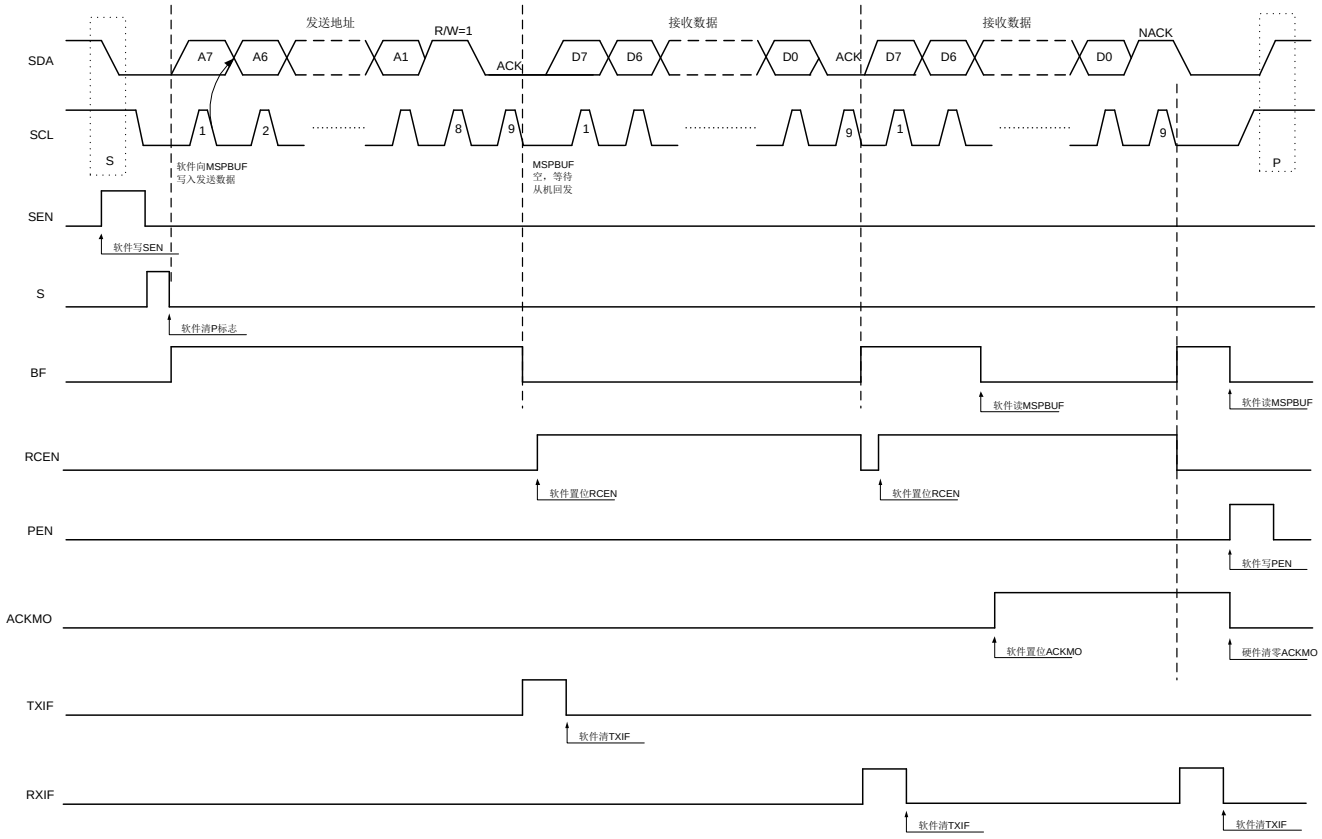


图 21-12 I2C 从 7 位地址从机读取数据流图

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。

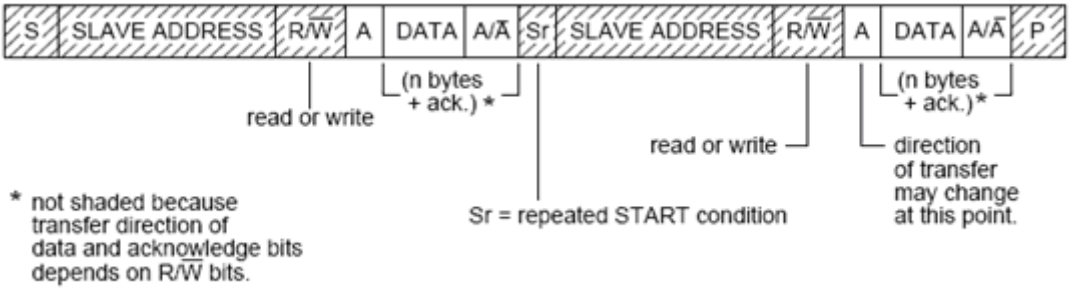


图 21-13 双向数据通信帧格式

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。

21.9.2 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。

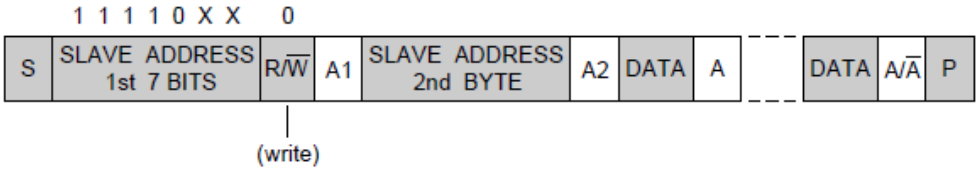


图 21-14 10bit 寻址，主机向从机写入数据

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：

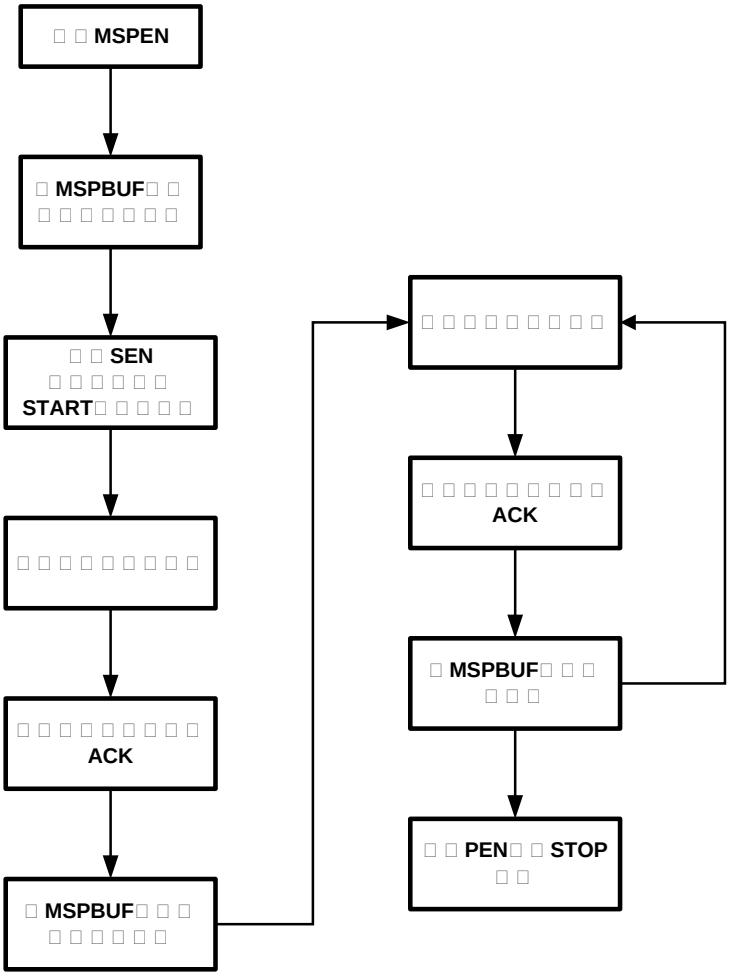


图 21-15I2C 软件发送数据流图

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。

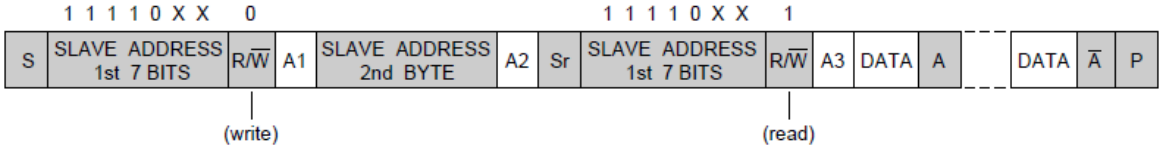


图 21-16 10bit 寻址，主机从从机读取数据

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：

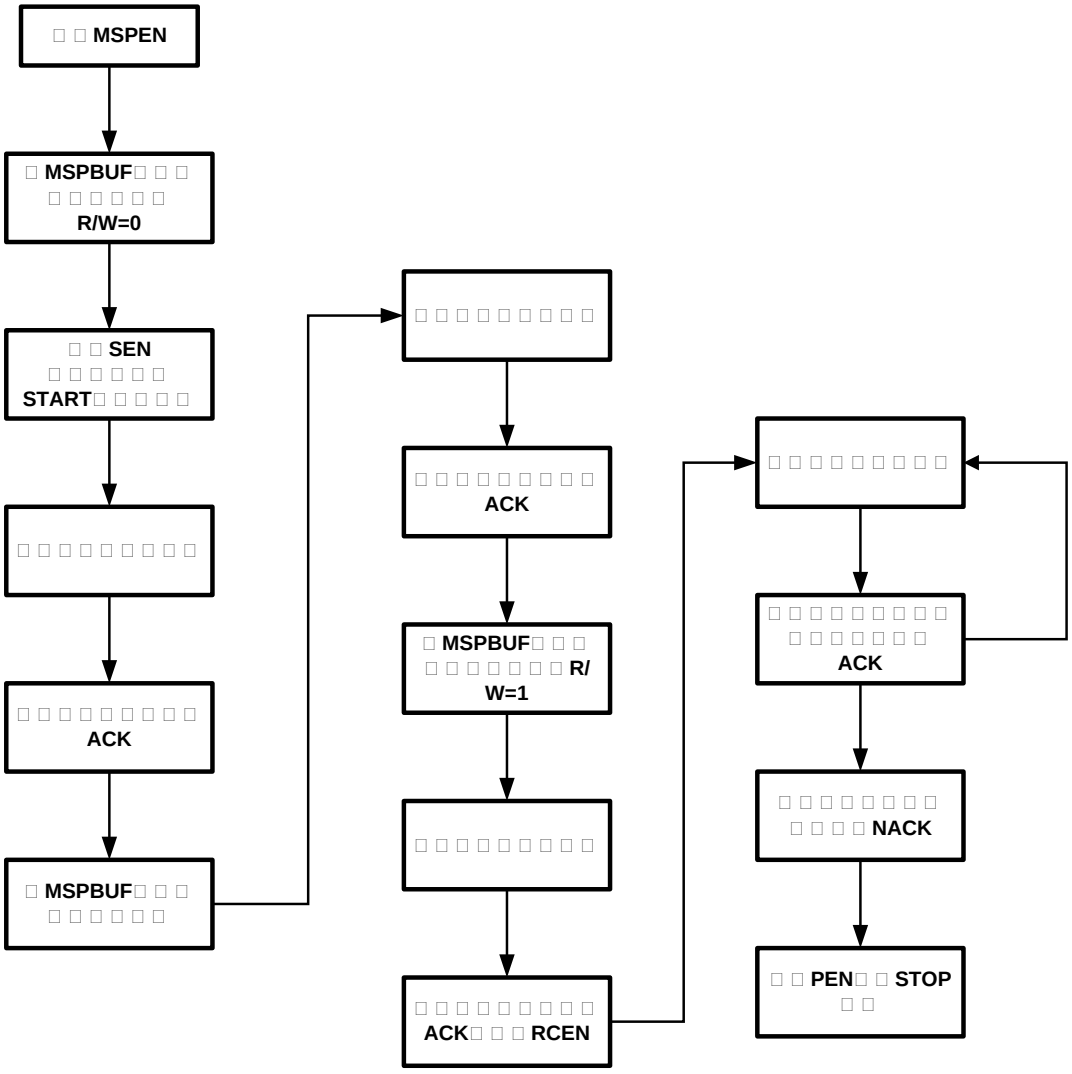


图 21-17I2C 软件发送数据流图

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。

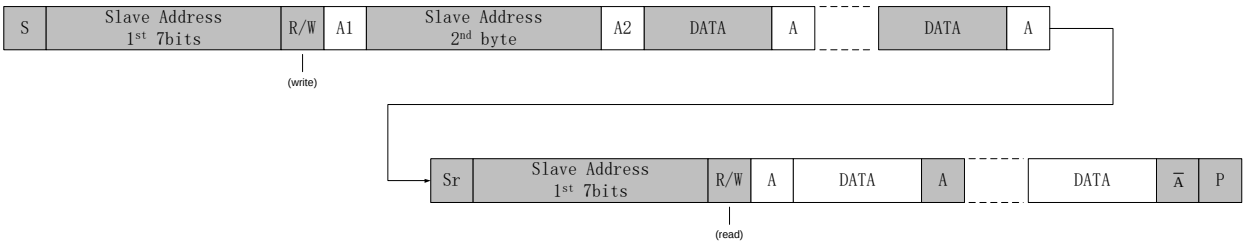


图 21-18 I2C 软件发送数据流图

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。

21.9.3 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，MSPEN 置位，如果数据缓存寄存器 MSPBUF 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 MSPBUF，同时 I2C 模块自动置位 SEN 产生 START 时序，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，此时可以又软件置位PEN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 PEN 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：

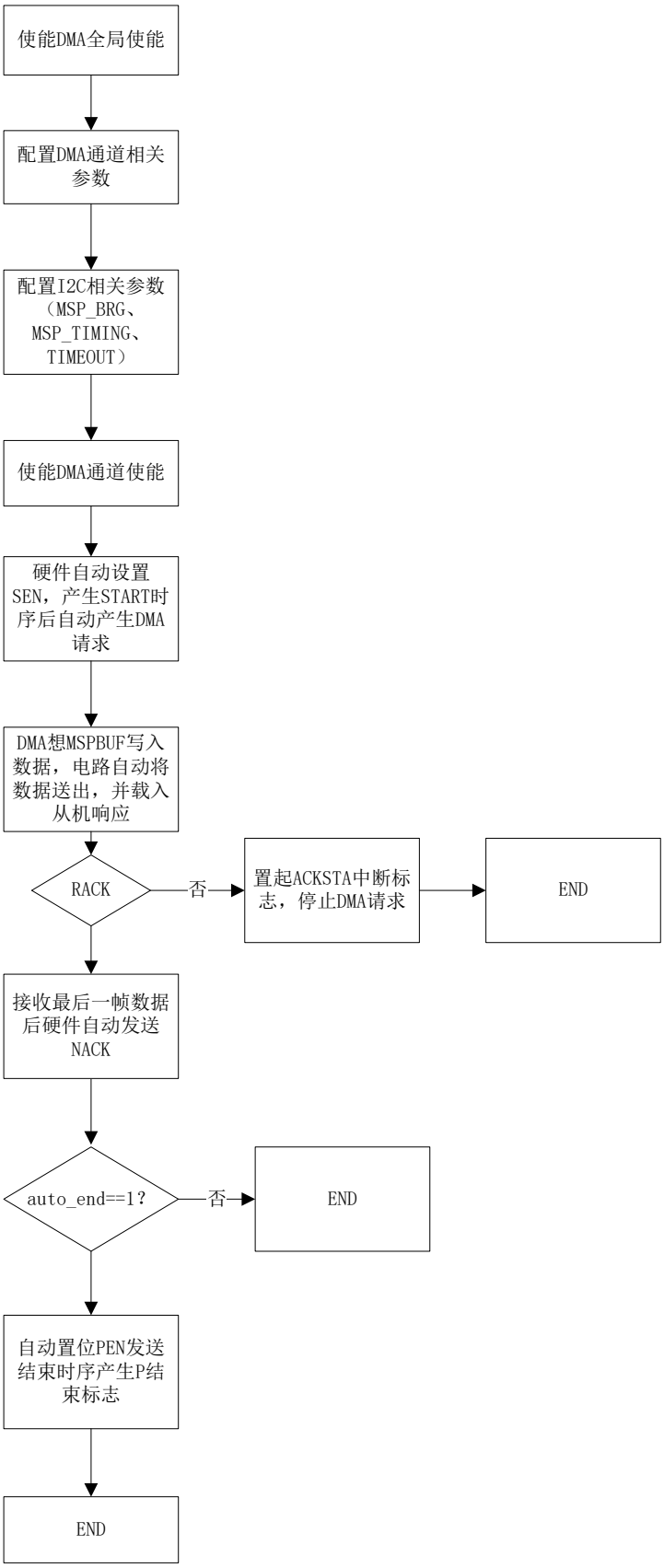


图 21-19 I2C 主机 DMA 发送流程图

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 MSP_DMAEN=1，然后使能对应的 DMA 通道，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 MSPBUF 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 PEN 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：

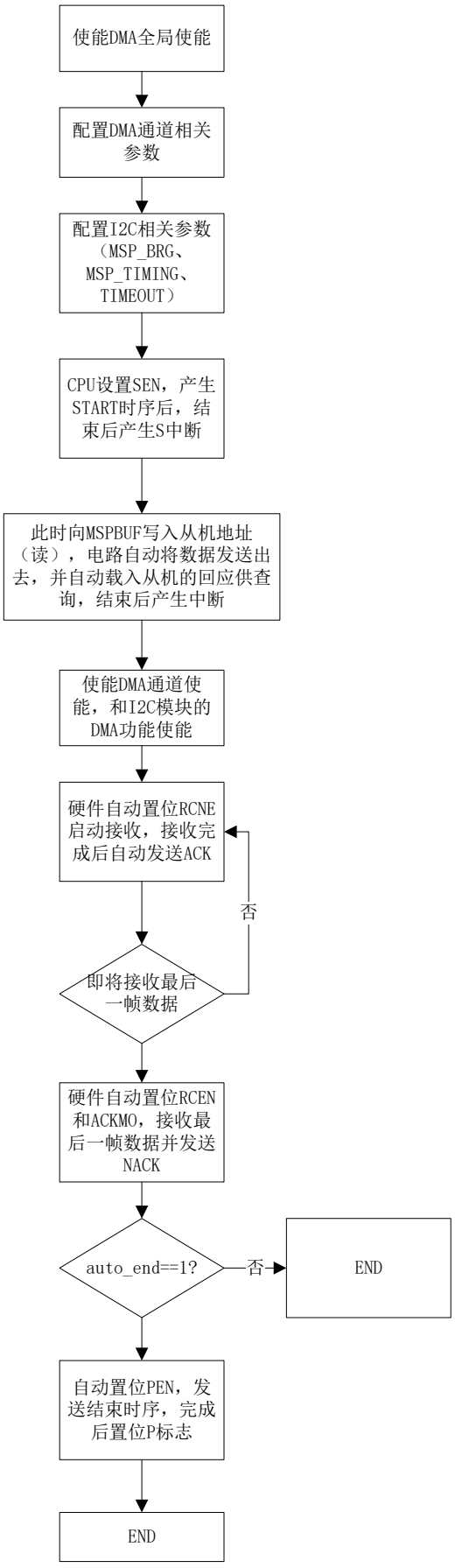


图 21-20 I2C 主机 DMA 接收流程图

21.9.4 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。

21.9.5 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。

21.9.6 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。

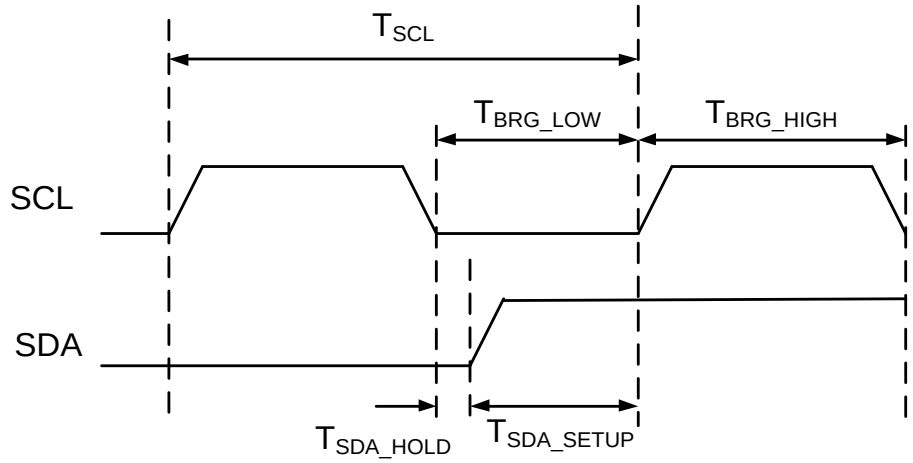


图 21-21 主机时序控制

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

21.10 I²C 从机功能

I2C 从机的工作不需要系统时钟，因此可以在芯片休眠的状态下进行数据收发和唤醒。

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（软件控制使能），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。

21.10.1 从机寻址

根据 SSPCON.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 SSPCON.A10EN=1 的情况下，要求 1st 字节必须以 11110 开头，否则将触发 ADDR_ERROR 错误标志。而在 SSPCON.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。

21.10.2 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SSPBUF
- 硬件自动释放SCL
- 新的SCL到来，SSPBUF移位输出到SDA总线
- 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：

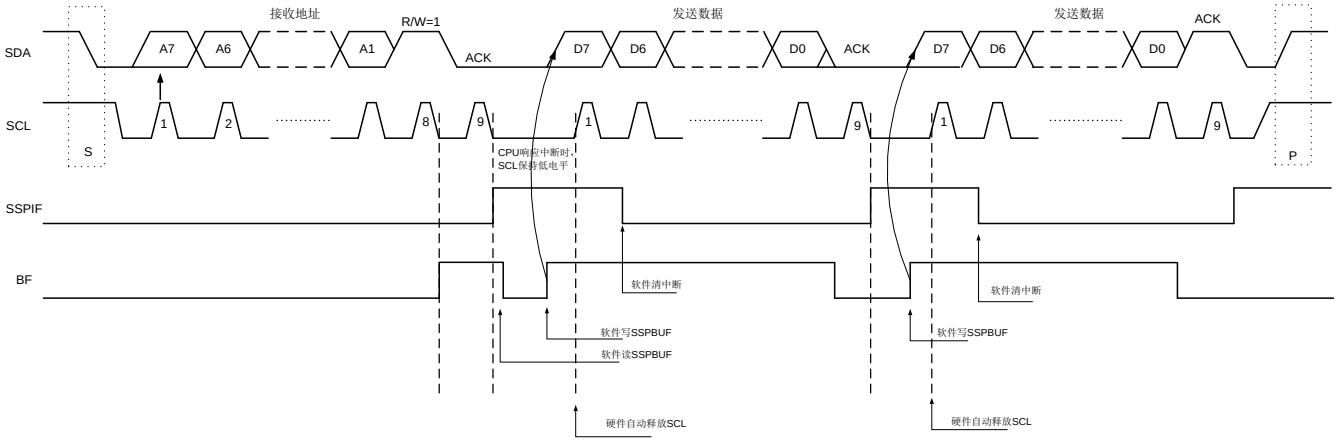


图 21-22 从机数据发送波形

在从机发送流程中，当从机收到正确地址时，ADM标志置位，地址字节将被写入SSPBUF并置位BF标志，然后硬件拉低SCL，等待软件写入SSPBUF，当软件读取SSPBUF后，BF标志自动清零，软件将待发送数据写入SSPBUF后BF置位，同时硬件释放SCL。

21.10.3 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SSPBUF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BF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SSPBUF，硬件自动清零BF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）：

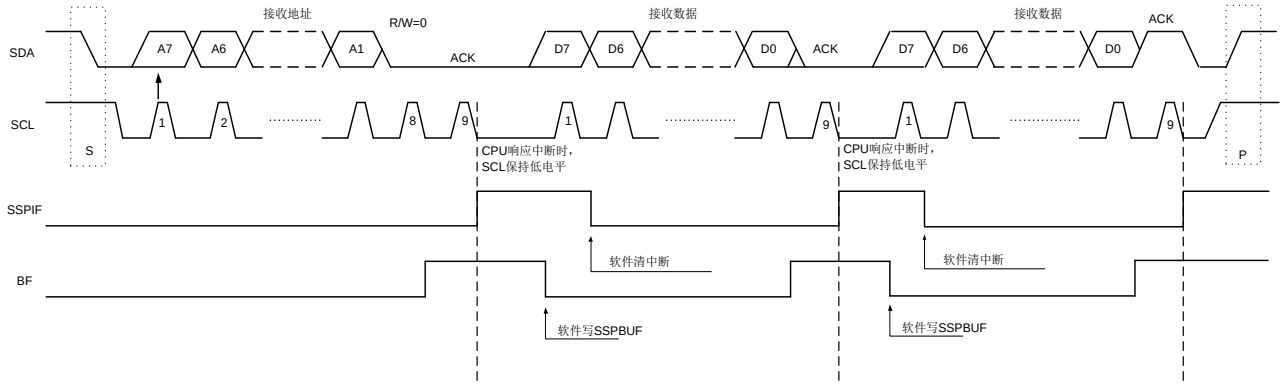


图21-23从机数据接收波形

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SSPBUF并置位BF标志，然后硬件拉低SCL。当软件读取SSPBUF后，BF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SSPBUF并导致BF置位，软件需要读取SSPBUF来清零BF并释放SCL。而从机发送流程中地址字节不会被写入SSPBUF因此也不会置位BF标志。

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BF标志。如果接收中出现了数据溢出，SSPOV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；SSPOV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，SSPOV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：

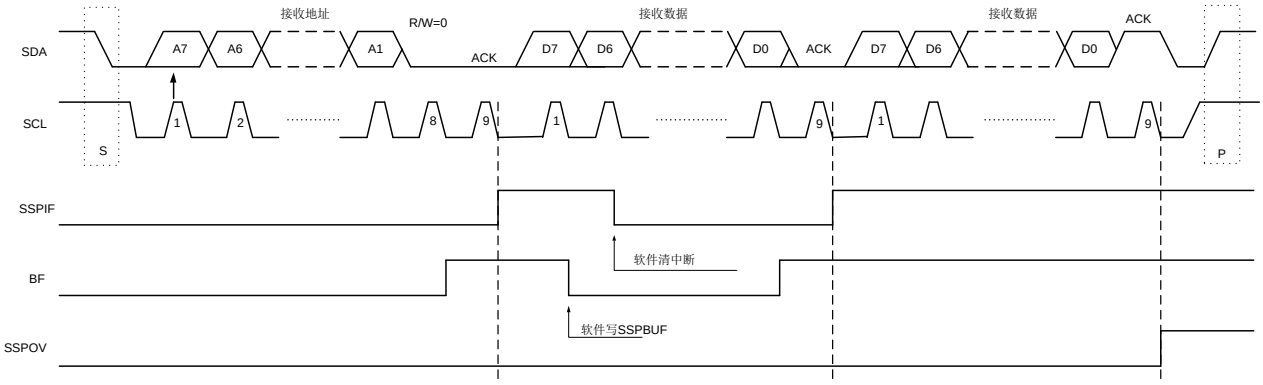


图21-24从机数据接收波形（SCLSEN=0，接收溢出）

21.10.4 从机低功耗接收唤醒

由于I2C从机不需要系统时钟即可工作，因此可以在休眠模式下接收数据并唤醒MCU。

I2C从机支持START时序唤醒、地址匹配唤醒和数据接收完成唤醒。

软件设置流程：

- 关闭I2C主机
- 设置从机地址
- 根据所需的唤醒事件，设置SE、ADME或BFE中断使能
- 设置对应的GPIO为I2C功能
- 置位SSPEN，启动I2C从机
- 进入休眠模式等待数据接收
- 当唤醒事件到来后，软件查询唤醒源，处理I2C数据传输

21.10.5 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 SSPBUF。

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：

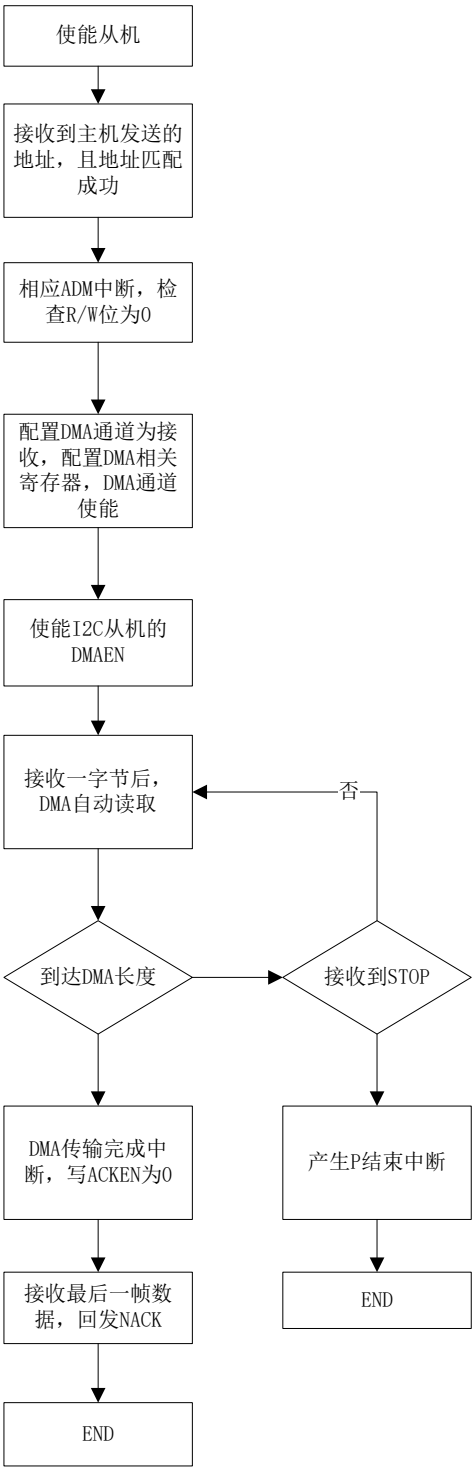


图 21-25 I2C 从机 DMA 接收流程图

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。此时软件需要先读取 SSPBUF 清除 BF 标志，然后配置特定 DMA 通道为 I2C_TX，并使能 I2C 从机的 DMAEN；随后当从机数据缓存 SSPBUF 为空时，将

产生 DMA 请求，通知 DMA 写入 SSPBUF。

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：

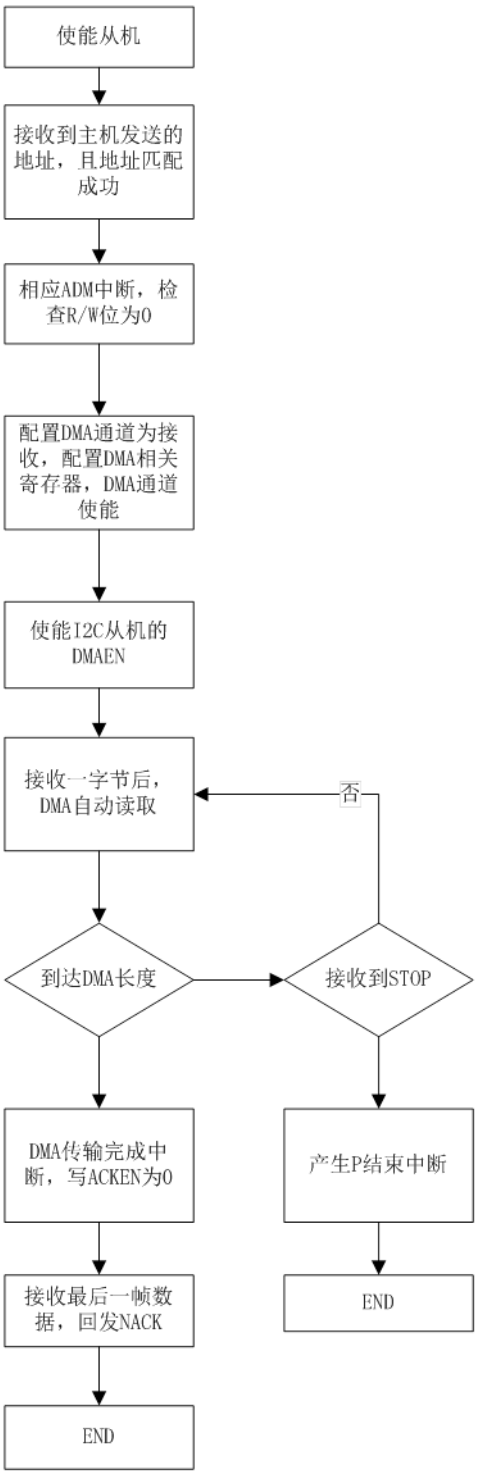


图 21-26 I2C 从机 DMA 发送流程图

21.10.6 从机时序

由于从机的数据收发只使用 SCL 进行，因此需要一些模拟延迟来实现 SDA 的数据建立和保持时间控制，而 SCL 的时序完全由主机控制。

从机时序控制如下图所示。根据 I2C 协议要求，SDA 相对于 SCL 下降沿的数据保持时间最小为 0ns，即从机使用 SCL 下降沿发送数据即可满足要求。但是考虑到总线上 SCL 波形的实际下降时间，为了更好的覆盖保持时间要求，在 SDA 输出上额外加入大于 300ns 的 RC 延迟。这个延迟仅需施加在 I2C 从机的 SDA 输出上（SSP_SDAO）

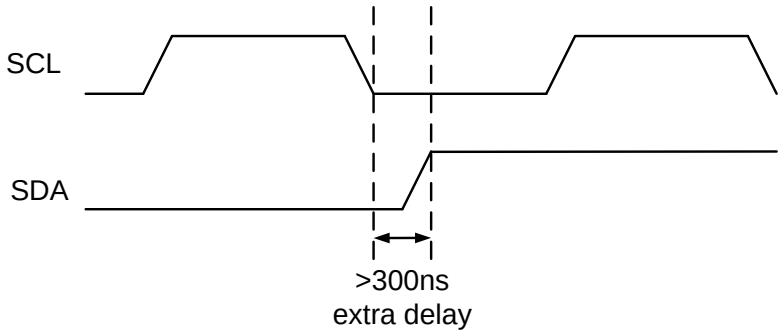


图 21-27 SDA 输出延迟波形

21.11 寄存器

offset 地址	名称	符号
I2C(模块基地址：0x40012400)		
0x00000000	I2C 主机配置寄存器（I2C Master Config Register）	I2C_MSPCFGR
0x00000004	I2C 主机控制寄存器（I2C Master Control Register）	I2C_MSPCR
0x00000008	I2C 主机中断使能寄存器（ I2C Master Intterupt Enable Register）	I2C_MSPIER
0x0000000C	I2C 主机中断标志寄存器（ I2C Master Interrupt Status Register）	I2C_MSPISR
0x00000010	I2C 主机状态寄存器（I2C Master Status Register）	I2C_MSPSR
0x00000014	I2C 主机波特率寄存器（I2C Master Baud rate Generator Register）	I2C_MSPBGR
0x00000018	I2C 主机收发缓存寄存器（I2C Master transfer Buffer）	I2C_MSPBUF
0x0000001C	I2C 主机时序控制寄存器（ I2C Master Timing Control Register）	I2C_MSPTCR
0x00000020	I2C 主机超时寄存器（I2C Master Time-Out Register）	I2C_MSPTOR
0x00000024	I2C 从机控制寄存器（I2C Slave Control Register）	I2C_SSPCR
0x00000028	I2C 从机中断使能寄存器（ I2C Slave Interrupt Enable Register）	I2C_SSPIER
0x0000002C	I2C 从机中断标志寄存器（ I2C Slave Interrupt Status Register）	I2C_SSPISR
0x00000030	I2C 从机状态寄存器（I2C Slave Status Register）	I2C_SSPSR
0x00000034	I2C 从机收发缓存寄存器（I2C Slave transfer Buffer）	I2C_SSPBUF
0x00000038	I2C 从机地址寄存器（I2C Slave Address Register）	I2C_SSPADR

21.11.1 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	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
31:15	--	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	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

21.11.2 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

位号	位名	说明
		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)

21.11.3 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	-	WCOLIE	OVTIE	SIE	PIE	NACKIE	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	WCOLIE	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	TOIE	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	SIE	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	PIE	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	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
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

21.11.4 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	-	-	ACKSTA	TXIF	RXIF
位权限	U-0	R/W-0	R/W-0	U-0	U-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；硬件置位，软件写 1 清零(SCL OverTime Interrupt Flag) 1：发生 SCL 超时 0：没有发生 SCL 超时
4	--	RFU：未实现，读为 0
3	--	RFU：未实现，读为 0
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。

21.11.5 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***

21.11.6 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)

21.11.7 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)

21.11.8 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

21.11.9 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 $T_{SCL_STRETCHING_TIMEOUT}$=TIMEOUT[11:0] * T_{SCL}

21.11.10 I2C 从机控制寄存器（I2C_SSPCR）

名称	I2C_SSPCR							
Offset	0x00000024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-00000000							

名称	I2C_SSPCR							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						SCLSEN	DMAEN
位权限	U-00000000						R/W-1	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			ACKEN	-	ANFEN	A10EN	SSPEN
位权限	U-0			RW-1	U-0	RW-0	RW-0	RW-0

位号	位名	功能描述
31:10	RFU	RFU：未实现，读为 0
9	SCLSEN	I2C 从机时钟延展使能 (SCL Stretching Enable) 0：禁止 slave clock stretching 1：使能 slave clock stretching *注意：当从机使用 DMA 通信时，必须将 SCLCEN 置 1*
8	DMAEN	I2C 从机 DMA 使能 (Slave DMA enable) 1：使能 DMA 功能 0：关闭 DMA 功能
7:5	-	RFU：未实现，读为 0
4	ACKEN	ACK 使能位：(Slave Ack Enable) 1 = slave 接收完成后将回发 ACK 0 = slave 不回发 ACK
3	SDAO_DLYEN	SDA 从机输出延迟使能 (SDA output delay enable) 0：bypass 从机 SDA 输出延迟 1：使能从机 SDA 输出延迟
2	SCLI_ANFEN	SCL 从机输入模拟滤波使能 (SCL input analog filter enable) 0：bypass 模拟滤波 1：使能模拟滤波
1	A10EN	10 位地址使能：(10bit Slave address enable) 1 = slave 使用 10bit address 0 = slave 使用 7bit address
0	SSPEN	I2C 从机使能位 (Slave enable) 1：使能 I2C 从机 0：关闭 I2C 从机

21.11.11 I2C 从机中断使能寄存器（I2C_SSPIER）

名称	I2C_SSPIER							
Offset	0x00000028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	--							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	--							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	--							

名称	I2C_SSPIER							
位权限	U-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADEIE	SIE	PIE	WCOLIE	SSPOVIE	ADMIE	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	功能描述
31:8	--	RFU：未实现，读为 0
7	ADEIE	从机地址错误中断使能，1 有效 (Address Error Enable)
6	SIE	Start 中断使能，1 有效 (Start interrupt enable)
5	PIE	Stop 中断使能，1 有效 (Stop interrupt enable)
4	WCOLIE	WCOL 中断使能，1 有效 (Write Collision Enable)
3	SSPOVIE	SSPOV 中断使能，1 有效 (Slave Buffer Overflow Enable)
2	ADMIE	从机地址匹配中断使能，1 有效 (Address Match Enable)
1	TXIE	发送完成中断使能，1 有效 (Transmit interrupt enable)
0	RXIE	接收完成中断使能，1 有效 (Receive interrupt enable)

21.11.12 I2C 从机中断标志寄存器（I2C_SSPISR）

名称	I2C_SSPISR							
Offset	0x0000002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	--							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	--							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	--							
位权限	U-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADE	S	P	WCOL	SSPOV	ADM	TXIF	RXIF
位权限	R/W1C-0	R-0	R-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0

位号	位名	功能描述
31:8	--	RFU：未实现，读为 0
7	ADE	从机地址格式错误，硬件置位，软件写 1 清零 (Address Error flag,write 1 to clear) 在 7 位地址情况下收到 11110 开头的地址字节，或者在 10 位地址情况下第一个字节不以 11110 开头时，触发 ADEE
6	S	检测到 start 时序，硬件置位，软件读取后自动清零 (Start flag)
5	P	检测到 stop 时序，硬件置位，软件读取后自动清零 (Stop flag)
4	WCOL	写冲突标志，硬件置位，软件写 1 清零 (Write Collision flag, write 1 to clear) 1：在 BF=1 的情况下，软件向 SSPBUF 写入新的数据

位号	位名	功能描述
		0：无写入冲突 当 WCOL 发生时，新的数据将被丢弃
3	SSPOV	SSPBUF 溢出标志，硬件置位，软件写 1 清零 (Slave buffer overflow flag,write 1 to clear) 1：在 BF=1 的情况下，从机又接收到新的数据 0：没有接收溢出 如果从机使能 SCL 延展，不会出现接收数据溢出的情况；因此 SSPOV 只可能在 SCLSEN=0 的情况下被置位。
2	ADM	从机地址匹配标志，硬件置位，软件写 1 清零 (Address Matched flag ,write 1 to clear) 1：接收到的 7bit 或 10bit 地址与 SLAVE_ADDR 寄存器内容一致 0：接收到的地址与 SLAVE_ADDR 不一致
1	TXIF	I2C 从机发送完成标志，硬件置位，软件写 1 清零 (Transmit interrupt flag,write 1 to clear)
0	RXIF	I2C 从机接收完成标志，硬件置位，软件写 1 清零 (Receive interrupt flag，write 1 to clear)

21.11.13 I2C 从机状态寄存器（I2C_SSPSR）

名称	I2C_SSPSR							
Offset	0x00000030							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
位权限	U-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				BUSY	RW_	DA	BF
位权限	U-0				R-0	R-0	R-0	R-0

位号	位名	功能描述
31:4	RFU	RFU：未实现，读为 0
3	BUSY	从机通信标志 (Busy) 1：从机数据收发中 0：从机空闲
2	RW_	读写方向状态寄存器 (Read/Write) 1：slave 收到 R/W=1，slave 需要发送数据给 master 0：slave 处于接收数据状态
1	DA	data/address 帧指示 1：上一字节收到的是数据 0：上一字节收到的是地址
0	BF	从机数据缓冲区满标志 (Buffer Full flag) 1：SSPBUF 满

位号	位名	功能描述
		0：SSPBUF 空

21.11.14 I2C 从机收发缓冲寄存器（I2C_SSPBUF）

名称	I2C_SSPBUF							
Offset	0x00000034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
位权限	U-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SSPBUF							
位权限	RW-00000000							

位号	位名	功能描述
31:8	RFU	RFU：未实现，读为 0
7:0	SSPBUF	SSPBUF[7:0]：数据的读写通过对 SSPBUF 的操作完成。发送时，对 SSPBUF 执行写操作，同时也载入数据收发移位寄存器(SSPSR)；接收时，SSPBUF 与 SSPSR 组成双缓冲结构，读出数据为 SSPBUF 的数据。接收完一个字节的数据，SSPSR 将数据载入 SSPBUF，同时置位 I2CIF。SSPSR 不是直接寄存器，没有物理地址 (Slave Buffer)

21.11.15 I2C 从机地址寄存器（I2C_SSPADR）

名称	I2C_SSPADR							
Offset	0x00000038							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU						SSPADDR	
位权限	U-000000						RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SSPADDR							
位权限	RW-00000000							

位号	位名	功能描述
31:10	RFU	RFU：未实现，读为 0
9:0	SSPADDR	从机地址寄存器 (Slave Address) A10EN = 1 10 位地址都有效 A10EN = 0 仅低 7 位有效

22 通用异步收发传输器（UART）

22.1 概述

UART串行通信模块特点如下

- 波特率软件可配置
- 5路独立通道（UART0, UART1, UART3, UART4, UART5）
- 全双工通信口
- 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

22.2 结构框图

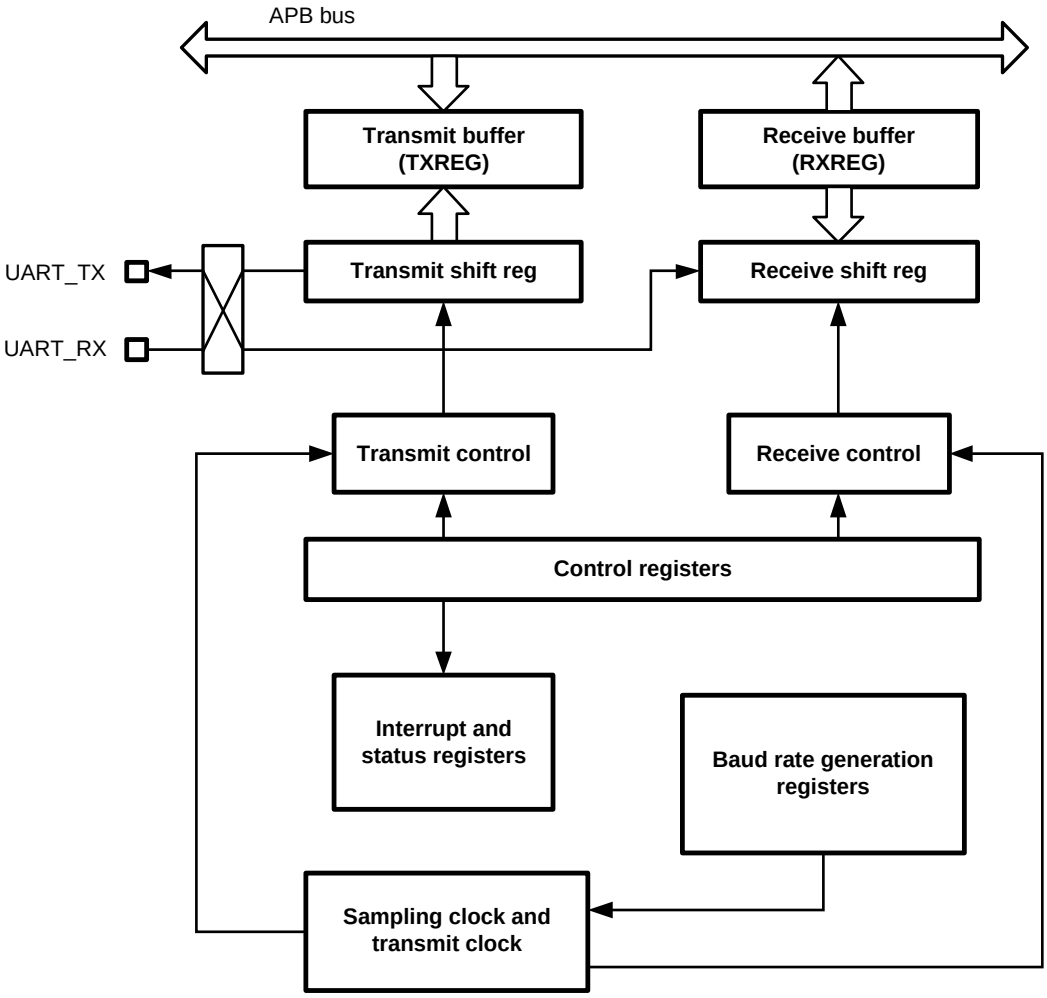


图 22-1 UART 接口时序

22.3 引脚定义

UART 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。

下表为 FM33LG0x8 的引脚映射关系

引脚		UARTx	符号	功能
PA2	PA13	UART0	UART0_RX	数据接收
PA3	PA14		UART0_TX	数据发送
PC2	PB13	UART1	UART1_RX	数据接收
PC3	PB14		UART1_TX	数据发送
PB0	PD7	UART3	UART3_RX	数据接收
PB1	PD8		UART3_TX	数据发送
PA0	PB2	UART4	UART4_RX	数据接收
PA1	PB3		UART4_TX	数据发送
PC4	PD0	UART5	UART5_RX	数据接收
PC5	PD1		UART5_TX	数据发送

下表为 FM33LG0x6 的引脚映射关系

引脚		UARTx	符号	功能
PA2	PA13	UART0	UART0_RX	数据接收
PA3	PA14		UART0_TX	数据发送
PC2	PB13	UART1	UART1_RX	数据接收
PC3	PB14		UART1_TX	数据发送
PB0	PD7	UART3	UART3_RX	数据接收
PB1	PD8		UART3_TX	数据发送
PA0	PB2	UART4	UART4_RX	数据接收
PA1	PB3		UART4_TX	数据发送
PC4	PD0	UART5	UART5_RX	数据接收
PC5	PD1		UART5_TX	数据发送

表 22-1 UART 引脚列表

当 UART 功能被同时映射到多个引脚上时：

- PA2 和 PA13 同时配置为数字外设功能
  - 只有 PA2 上的 RX 信号会输入到模块内部
- PC2 和 PB13 同时配置为数字外设功能
  - 只有 PA2 上的 RX 信号会输入到模块内部
- PC4 和 PD0 同时配置为数字外设功能
  - 只有 PC4 上的 RX 信号会输入到模块内部
- PA0 和 PB2 同时配置为数字外设功能

- 只有 PA0 上的 RX 信号会输入到模块内部

- UART 发送功能被同时映射到多个 GPIO 上时，这些引脚会同时发送数据

22.4 UART 类型区分

FM33LG0xx集成了多种不同类型的UART（LPUART），其差异如下表所示：

UART 特性	UART0/1	UART3/4/5	LPUART0/1/2
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
休眠唤醒	Y	-	Y
接收超时	Y	-	-
发送延迟	Y	-	-
数据长度	6、7、8、9bits		

表 22-2 UART 类型列表

22.5 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。

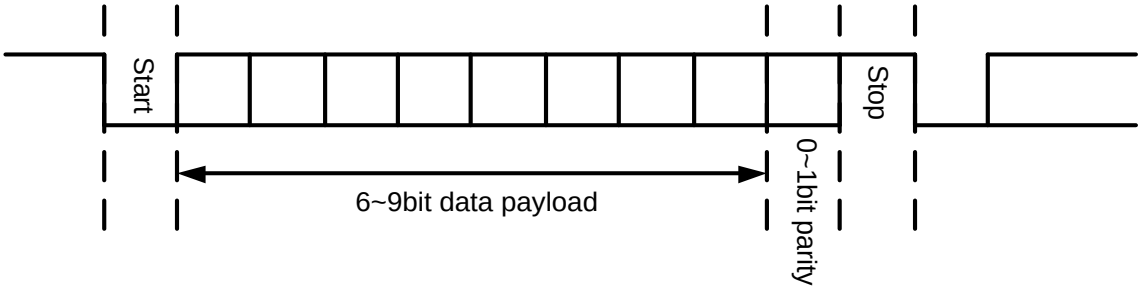


图 22-2 UART 字符描述

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 22-3 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】。

22.6 功能描述

22.6.1 时钟和复位

UART0 和 UART1 采用了双时钟结构：

- 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、RCLF，能够独立于 APBCLK 工作。必须使能 UCLK 才能进行数据收发。

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，UART4 和 UART5 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。

UART 模块工作前必须清除 RMU 中的复位寄存器（UARTxRST）。

22.6.2 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。

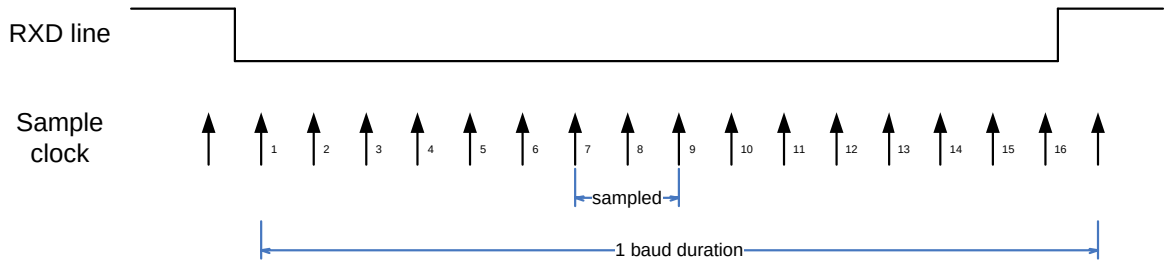


图 22-3 位接收 16 倍采样

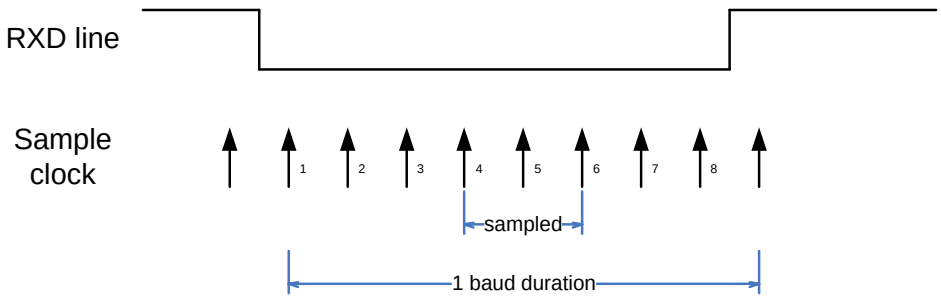


图 22-4 位接收 8 倍采样

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。

22.6.3 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。

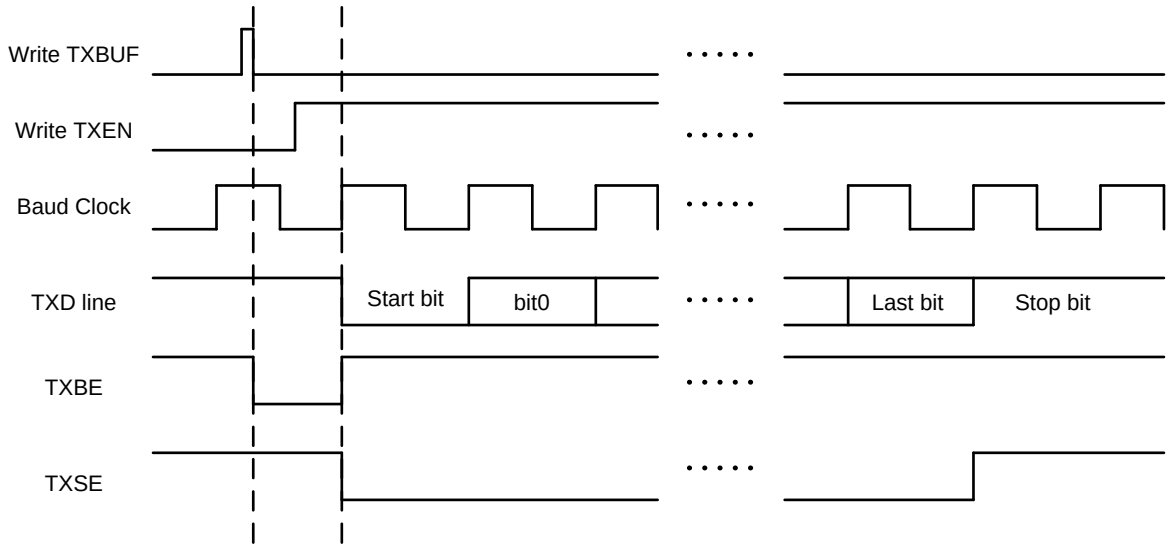


图 22-5 UART 异步发送波形 1

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。

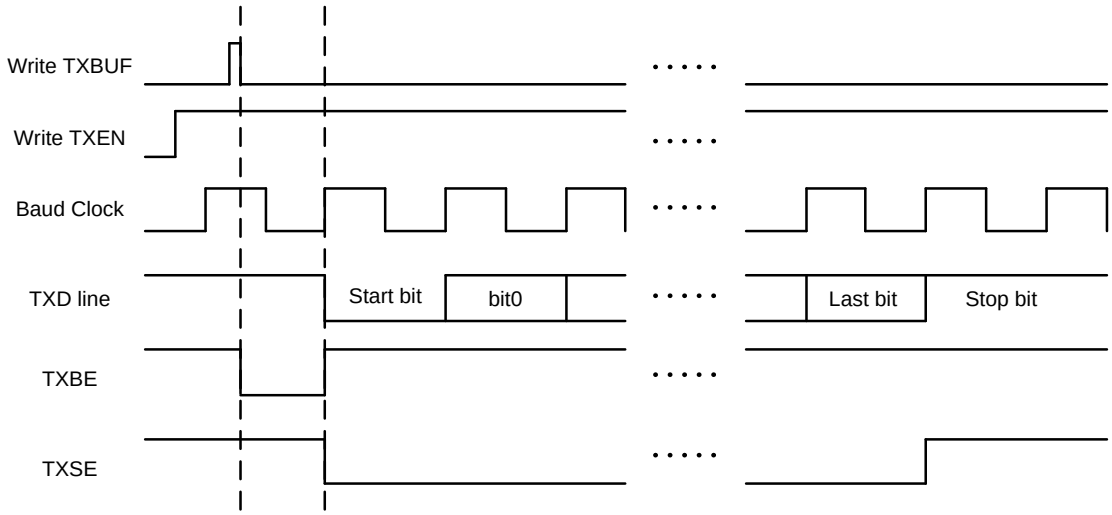


图 22-6 UART 异步发送波形 2

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。

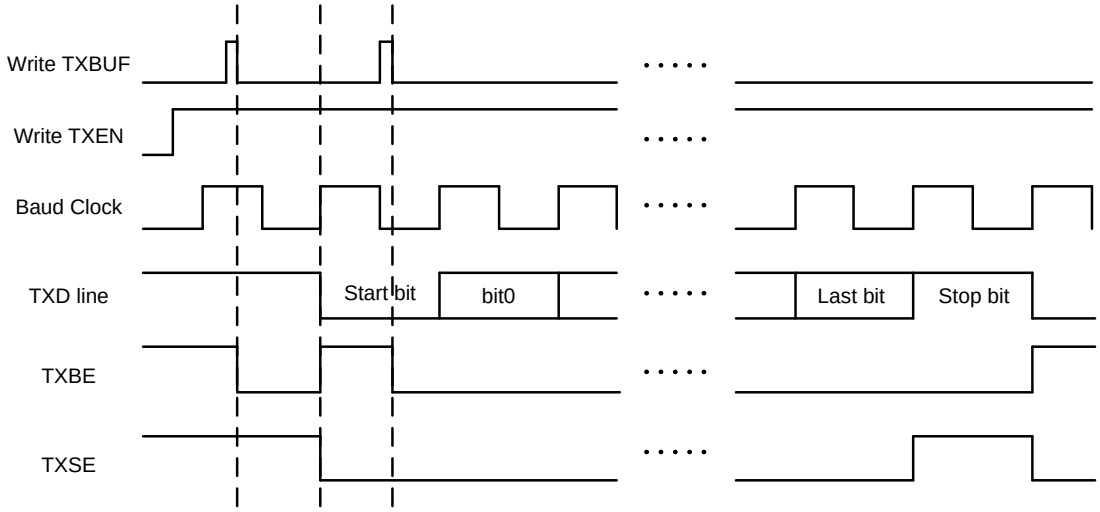


图 22-7 UART 异步发送波形 3

22.6.4 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

22.6.5 低功耗休眠唤醒（UART0/1）

UART0和UART1支持RXD下降沿触发的芯片休眠唤醒。当置位了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UART0/1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

22.6.6 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

22.6.7 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 22-4 DMA 发送中断

22.7 波特率发生

22.7.1 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。

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 22-5 常用时钟频率下波特率计算

22.7.1 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。

22.8 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：

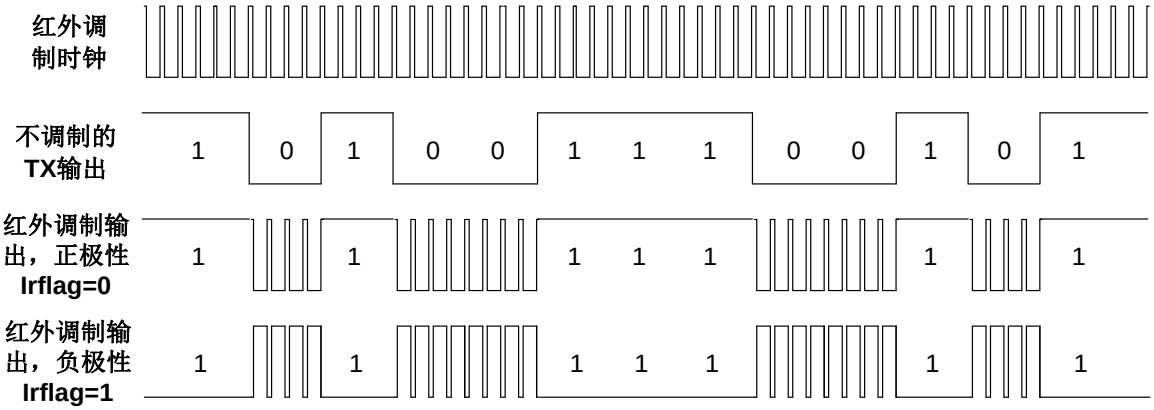


图 22-8 红外调制波形

无论有效电平是 0 还是 1，占空比定义为高电平长度/周期。

22.9 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。

注：UART4 和 UART5 不支持接收超时功能。

22.10 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。

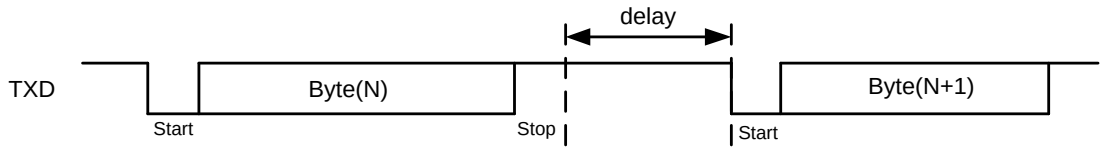


图 22-9 UART 发送延迟

注：UART4 和 UART5 不支持发送延迟功能。

22.11 寄存器

offset 地址	名称	符号
UARTIR寄存器(模块基地址：0x40017C00)		
0x00	红外调制寄存器 （Infrared modulation Control Register）	UART_IRCR
UART0 寄存器(模块基地址：0x40012000)		
0x00	UART0 控制状态寄存器 （UART0 Control Status Register）	UART0_CSR
0x04	UART0 中断使能寄存器 （UART0 Interrupt Enable Register）	UART0_IER
0x08	UART0 中断标志寄存器 （UART0 Interrupt Status Register）	UART0_ISR
0x0C	UART0 超时和延迟寄存器 （UART0 Time-Out and Delay Register）	UART0_TODR
0x10	UART0 接收缓冲寄存器 （UART0 Receive Buffer）	UART0_RXBUF
0x14	UART0 发送缓冲寄存器 （UART0 Transmit Buffer）	UART0_TXBUF
0x18	UATR0 波特率产生寄存器 （UART0 Baud rate Generator Register）	UART0_BGR
UART1 寄存器(模块基地址：0x40016800)		
0x00	UART1 控制状态寄存器 （UART1 Control Status Register）	UART1_CSR
0x04	UART1 中断使能寄存器 （UART1 Interrupt Enable Register）	UART1_IER
0x08	UART1 中断标志寄存器 （UART1 Interrupt Status Register）	UART1_ISR
0x0C	UART1 超时和延迟寄存器 （UART1 Time-Out and Delay Register）	UART1_TODR
0x10	UART1 接收缓冲寄存器 （UART1 Receive Buffer）	UART1_RXBUF
0x14	UART1 发送缓冲寄存器 （UART1 Transmit Buffer）	UART1_TXBUF
0x18	UATR1 波特率产生寄存器 （UART1 Baud rate Generator Register）	UART1_BGR
UART3 寄存器(模块基地址：0x40017000)		
0x00	UART3 控制状态寄存器 （UART3 Control Status Register）	UART3_CSR
0x04	UART3 中断使能寄存器 （UART3 Interrupt Enable Register）	UART3_IER
0x08	UART3 中断标志寄存器 （UART3 Interrupt Status Register）	UART3_ISR
0x10	UART3 接收缓冲寄存器 （UART3 Receive Buffer）	UART3_RXBUF
0x14	UART3 发送缓冲寄存器 （UART3Transmit Buffer）	UART3_TXBUF
0x18	UATR3 波特率产生寄存器 （UART3 Baud rate Generator Register）	UART3_BGR

offset 地址	名称	符号
UART4 寄存器(模块基地址：0x40017400)		
0x00	UART4 控制状态寄存器 （UART4 Control Status Register）	UART4_CSR
0x04	UART4 中断使能寄存器 （UART4 Interrupt Enable Register）	UART4_IER
0x08	UART4 中断标志寄存器 （UART4 Interrupt Status Register）	UART4_ISR
0x10	UART4 接收缓冲寄存器 （UART4 Receive Buffer）	UART4_RXBUF
0x14	UART4 发送缓冲寄存器 （UART4 Transmit Buffer）	UART4_TXBUF
0x18	UATR4 波特率产生寄存器 （UART4 Baud rate Generator Register）	UART4_BGR
UART5 寄存器(模块基地址：0x40017800)		
0x00	UART5 控制状态寄存器 （UART5 Control Status Register）	UART5_CSR
0x04	UART5 中断使能寄存器 （UART5 Interrupt Enable Register）	UART5_IER
0x08	UART5 中断标志寄存器 （UART5 Interrupt Status Register）	UART5_ISR
0x10	UART5 接收缓冲寄存器 （UART5 Receive Buffer）	UART5_RXBUF
0x14	UART5 发送缓冲寄存器 （UART5 Transmit Buffer）	UART5_TXBUF
0x18	UATR5 波特率产生寄存器 （UART5 Baud rate Generator Register）	UART5_BGR

22.11.1 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
14:11	**TH**	红外占空比调制参数

位号	助记符	功能描述
10:0	**TZBRG**	红外调制频率

22.11.2 UARTx 控制状态寄存器（UARTx_CSR）

名称	UARTx_CSR(x=0,1,2,3,4,5)							
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	-						TXIREN	RXTOEN
位权限	U-0						R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		OVSM	IOSWAP	NEWUP	DMATXIFCFG	BITORD	STOPCFG
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	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**	UART 通信标志，只读 1：UART 正在通信中 0：UART 空闲
23:18	**-**	未实现：读为0
17	**TXIREN**	发送红外调制使能 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 *注：仅 UART0/1 具有此功能，其他 UART 模块请保持为 0*
15:14	**-**	未实现：读为0
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
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） 1：使能RX下降沿唤醒 0：禁止RX下降沿唤醒

位号	助记符	功能描述
10	**DMATXIFCFG**	DMA发送完成中断使能，仅在UART通过DMA进行发送时有效 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 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 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 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 0：正向 1：取反
2	**TXPOL**	发送数据极性配置 0：正向 1：取反
1	**RXEN**	接收使能，1 有效
0	**TXEN**	发送使能，1 有效

22.11.3 UARTx 中断使能寄存器（UARTx_IER）

名称	UARTx_IER(x=0,1,2,3,4,5)							
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	-				RXTO_IE	RXERR_IE	-	RXBF_IE
位权限	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
31:12	-	**未实现：读为0**
11	**RXTO_IE**	接收超时中断使能，1 有效 （仅 UART0 和 UART1 有效）
10	**RXERR_IE**	接收错误中断使能，1 有效 此 bit 用于控制 PERR、FERR、OERR 三个接收错误标志是否产生中断事件。
9	-	**未实现：读为0**
8	**RXBF_IE**	接收缓存满中断使能，1 有效
7	**NEWUP_IE**	RX 下降沿异步检测中断使能，1 有效 （仅 UART0 和 UART1 有效）
6:2	-	**未实现：读为 0**
1	**TXBE_IE**	发送缓存空中断使能，1 有效
0	**TXSE_IE**	发送缓存空且发送移位寄存器空中断使能，1 有效

22.11.4 UARTx 中断标志寄存器（UARTx_ISR）

名称	UARTx_ISR(x=0,1,2,3,4,5)							
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	-				RXTO	-		RXBF
位权限	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-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
17	**FERR**	帧格式错误中断标志，硬件置位，软件写 1 清零
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。
15:12	-	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 （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
7	**NEWKF**	RX 下降沿异步检测中断标志，硬件置位，软件写 1 清零 （仅 UART0 和 UART1 有效）
6:3	-	未实现：读为 0

位号	助记符	功能描述
2	**TX_OERR**	发送缓存不为空时，对发送缓存的写操作将置位此标志，软件写 1 清零。注意此 bit 不会产生中断事件，仅供查询。
1	**TXBE**	发送缓存空中断标志，硬件置位，软件写入 TXBUF 时清零
0	**TXSE**	发送缓存空且移位寄存器发送完成中断标志，硬件置位，软件写 1 或者软件写发送缓存时清零

22.11.5 UARTx 超时和延迟寄存器（UARTx_TODR）

名称	UARTx_TODR(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	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
7:0	**RXTO_LEN**	接收超时溢出长度，最大 255baud

22.11.6 UARTx 接收缓冲寄存器（UARTx_RXBUF）

名称	UARTx_RXBUF(x=0,1,2,3,4,5)							
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

7位收发时，接收的7bits数据存入RXBUF[6:0]

22.11.7 UARTx 发送缓冲寄存器（UARTx_TXBUF）

名称	UARTx_TXBUF(x=0,1,2,3,4,5)							
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

7位收发时，发送的7bits数据写入TXBUF[6:0]

22.11.8 UATRx 波特率产生寄存器（UARTx_BGR）

名称	UARTx_BGR(x=0,1,2,3,4,5)							
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

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;

23 低功耗 UART（LPUART）

23.1 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）、低功耗低频环振时钟（RCLF）。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（LPUART0, LPUART1, LPUART2）
- 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或RCLF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
- 支持DMA（Sleep/DeepSleep模式下不支持）

23.2 结构框图

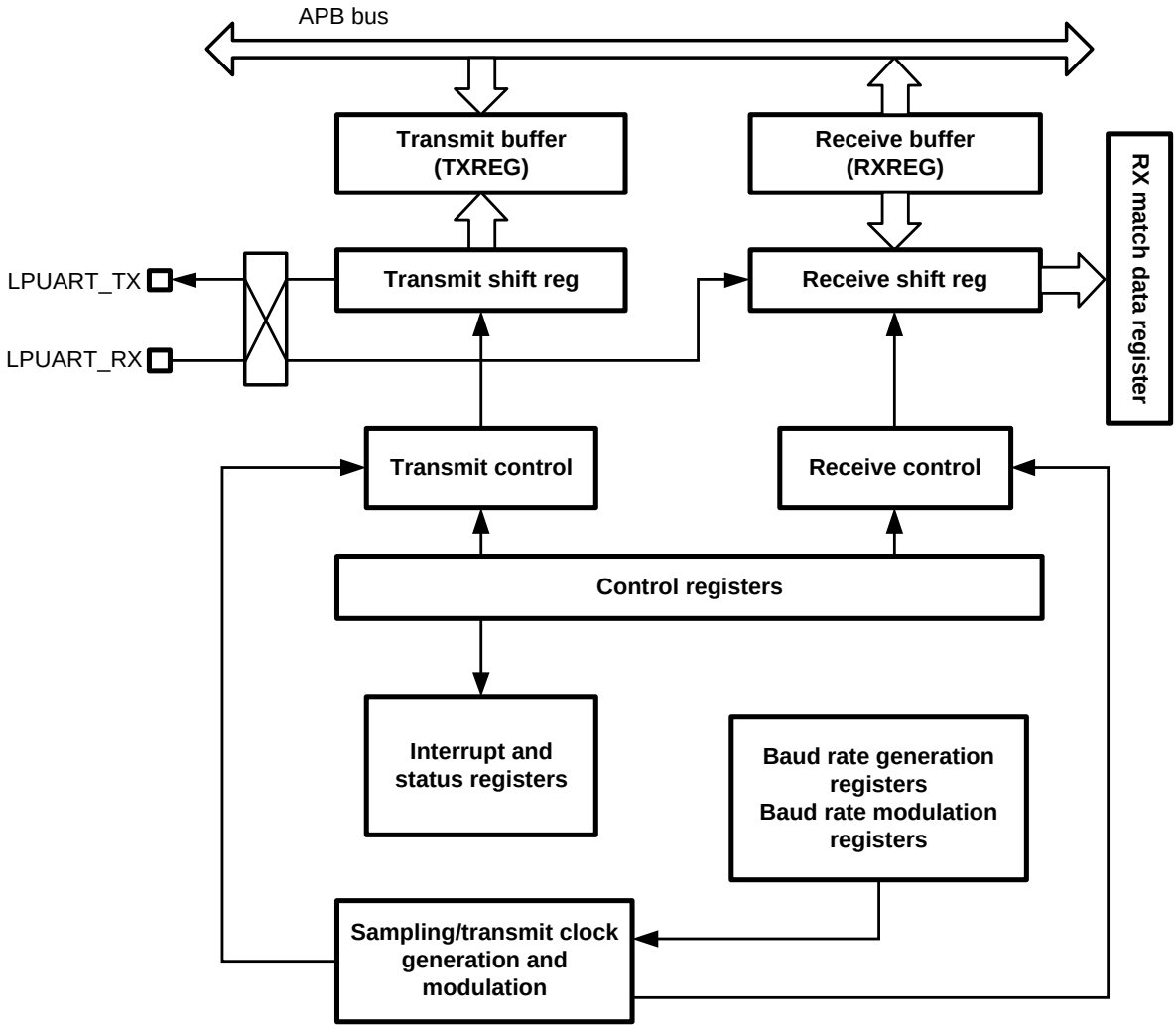


图 23-1 LPUART 结构框图

23.3 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
PA13	PA2	LPUART0	LPUART0_RX	数据接收
PA14	PA3		LPUART0_TX	数据发送
PC2	PB13[1]	LPUART1	LPUART1_RX	数据接收
PC3	PB14[1]		LPUART1_TX	数据发送
PA11	PB4	LPUART2	LPUART2_RX	数据接收
PA12	PB5		LPUART2_TX	数据发送

表 23-1 LPUART 引脚对应表

[1]仅FM33LG0x6型号有

当 LPUART 功能被同时映射到多个引脚上时：

- PA2 和 PA13 同时配置为数字外设功能
  - 只有 PA2 上的 RX 信号会输入到模块内部
- PC2 和 PB13 同时配置为数字外设功能
  - 只有 PC2 上的 RX 信号会输入到模块内部
- PA11 和 PB4 同时配置为数字外设功能
  - 只有 PA11 上的 RX 信号会输入到模块内部
- LPUART 发送功能被同时映射到多个 GPIO 上时，这些引脚会同时发送数据

23.4 时钟和复位

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 可以使用 XTLF 或者 RCLF 工作。由于 RCLF 精度不高，在使用 RCLF 进行 LPUART 通信前建议先进行时钟校准，将 RCLF 校准到+/-1%以内。

在 ACTIVE 模式下，LPUART 也可以使用 RCHF 工作，此时时钟精度会高于 RCLF，以获得更好的时序容错性能。使用 RCHF 工作时，prescaler 电路对 RCHF 进行预分频，获得与 32768Hz 相近的时钟频率，比如 RCHF 为 8M/16M/24M 时，prescaler 分频系数应为分别 244/488/732。

LPUART 工作时钟结构参见下图，这部分功能和寄存器在 CMU 模块实现，通过置位 LPUARTx_PCE 寄存器使能总线时钟以操作寄存器，通过配置 LPUARTxCKS 选择工作时钟源，通过置位

LPUARTxCKE 使能工作时钟。

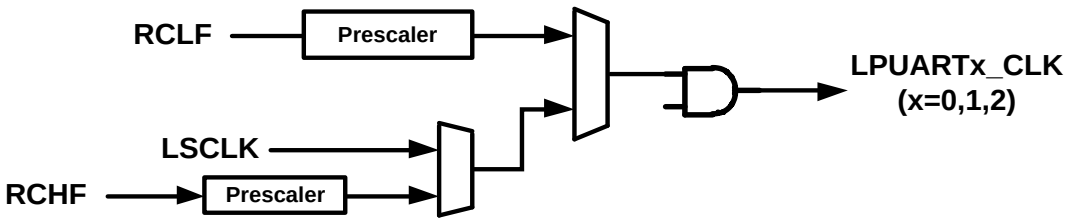


图 23-2 LPUART 工作时钟

RCLF 频率为 614.4Khz，经过 CMU 模块内部预分频，输出给 LPUART 的工作时钟为 38.4Khz，即 9600 的 4 倍频。通过这个时钟，可以实现更为准确的 LPUART 数据收发，特别是保证输出 baud 长度的稳定性。

23.5 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。

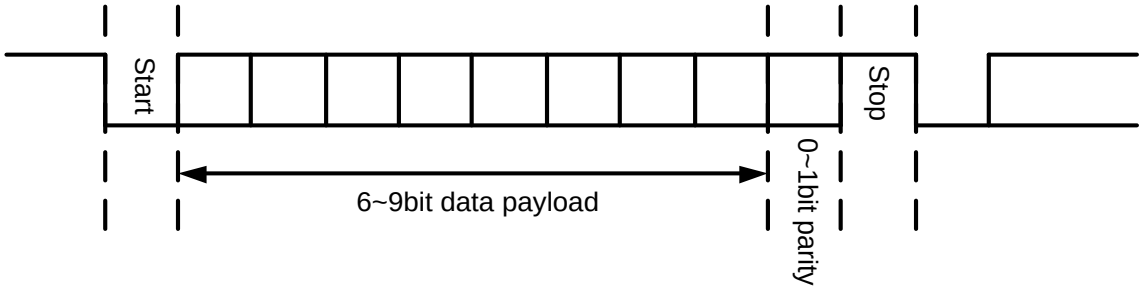


图 23-3 字符描述

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 23-2LPUART 数据帧格式

[1]：Stop 位可能是 1bit 或者 2bit，根据 STOPCFG 寄存器决定

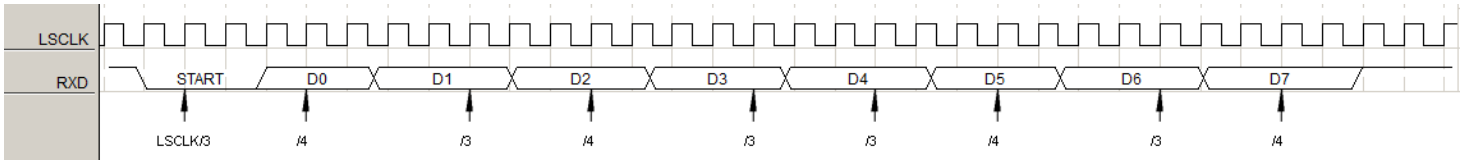
注意 PDSEL 寄存器用于配置帧的数据长度，通信帧长为【起始位+数据位+校验位+停止位】。

23.6 功能描述

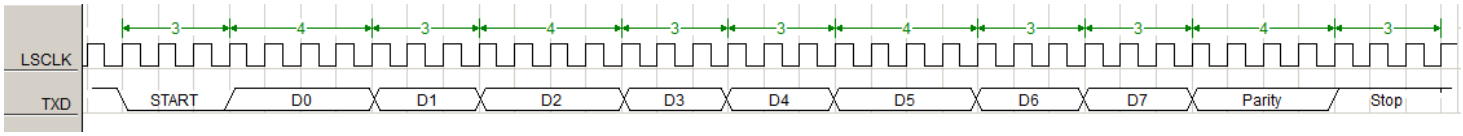
23.6.1 位接收采样和发送

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 23-3 LPUART 数据位调制系数

当工作时钟选择为 RCLF 分频时，经过校准后的 RCLF 频率为 614.4Khz 左右，对应 9600 的 64 倍频，经过预分频后得到 38.4Khz 工作时钟，此时 LPUART 通信不需要进行数据位调制，软件应将

MCTL_EN 清零。

23.6.2 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

23.6.3 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

23.6.4 使用 DMA 进行 LPUART 收发

当 LPUART 模块被使能后，LPUART 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 LPUART 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 LPUART 请求，并完成 RAM 和 LPUART 之间的数据搬运。

应用举例：使用 DMA 进行 LPUART1 接收

- 将 DMA 通道 0 或 3 配置为 LPUART1_RX
- 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 LPUART1 模块参数
- 使能 LPUART1 模块接收使能 LPUEN.RXEN=1，等待数据接收
- 收到数据后 LPUART1 自动产生 DMA 请求
- DMA 响应请求，读取 LPUART1 接收缓存寄存器，写入指定 RAM 地址

23.6.5 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

23.6.6 LPRUN 模式下的数据 DMA 收发

通过 LPUART 和 DMA，软件可以实现 LPRUN 模式下一定数据量的 LPUART 自动收发，而无需 CPU 干预，同时保证典型条件下全芯片功耗小于 10uA。

- 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DMA通道控制寄存器，选择LPUART收发
- 如果需要发送数据，将待发数据写入RAM中指定位置
- 配置DMA数据收发长度和RAM指针
- 将系统主时钟选为LSCLK
- 软件进入LPRUN
- 配置LPUEN寄存器打开发送接收使能
- 如CPU无额外工作，可以主动进入WFI/WFE，等待中断唤醒

23.6.7 DMA 模式下的发送完成中断

当 LPUART 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 LPUART 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LPUART发送

- 关闭DMA通道中断使能
- 置位LPUART TXBE_IE或TXSE_IE寄存器，允许中断产生
- 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
- LPUART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，LPUART产生TXBE或TXSE中断

下表假设 LPUART 通过 DMA 发送 N 个帧：

TXBE_IE TXSE_IE	DMATXIFCFG	Frame No.	TXBE TXSE	LPUART 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 23-4 LPUART DMA 中断说明

23.7 寄存器

offset 地址	名称	符号
LPUART0 寄存器(模块基地址：0x40014000)		
0x00	LPUART0 控制状态寄存器（LPUART0 Control Status Register）	LPUART0_CSR
0x04	LPUART0 中断使能寄存器（LPUART0 Interrupt Enable Register）	LPUART0_IER
0x08	LPUART0 中断标志寄存器（LPUART0 Interrupt Status Register）	LPUART0_ISR
0x0C	LPUART0 波特率调制寄存器（ LPUART0 Baud rate Modulation Register）	LPUART0_BMR
0x10	LPUART0 接收缓冲寄存器（LPUART0 Receive Buffer Register）	LPUART0_RXBUF
0x14	LPUART0 发送缓冲寄存器（LPUART0 Transmit Buffer Register）	LPUART0_TXBUF
0x18	LPUART0 数据匹配寄存器（LPUART0 data Matching Register）	LPUART0_DMR
LPUART1 寄存器(模块基地址：0x40014400)		
0x00	LPUART1 控制状态寄存器（LPUART1 Control Status Register）	LPUART1_CSR
0x04	LPUART1 中断使能寄存器（LPUART1 Interrupt Enable Register）	LPUART1_IER
0x08	LPUART1 中断标志寄存器（LPUART1 Interrupt Status Register）	LPUART1_ISR
0x0C	LPUART1 波特率调制寄存器（ LPUART1 Baud rate Modulation Register）	LPUART1_BMR
0x10	LPUART1 接收缓冲寄存器（LPUART1 Receive Data Register）	LPUART1_RXBUF
0x14	LPUART1 发送缓冲寄存器（LPUART1 Transmit Data Register）	LPUART1_TXBUF
0x18	LPUART1 数据匹配寄存器（LPUART1 data Matching Register）	LPUART1_DMR
LPUART2 寄存器(模块基地址：0x40015000)		
0x00	LPUART2 控制状态寄存器（LPUART2 Control Status Register）	LPUART2_CSR
0x04	LPUART2 中断使能寄存器（LPUART2 Interrupt Enable Register）	LPUART2_IER
0x08	LPUART2 中断标志寄存器（LPUART2 Interrupt Status Register）	LPUART2_ISR
0x0C	LPUART2 波特率调制寄存器（ LPUART2 Baud rate Modulation Register）	LPUART2_BMR
0x10	LPUART2 接收缓冲寄存器（LPUART1 Receive Data Register）	LPUART2_RXBUF
0x14	LPUART2 发送缓冲寄存器（LPUART2 Transmit Data Register）	LPUART2_TXBUF
0x18	LPUART2 数据匹配寄存器（LPUART2 data Matching Register）	LPUART2_DMR

23.7.1 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	DMATXIFCFG	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 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 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
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 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 0：默认引脚顺序（与封装图一致） 1：交换引脚顺序
10	**DMATXIFCFG**	DMA发送完成中断使能，仅在LPUART通过DMA进行发送时有效 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 0：LSB first 1：MSB first

位号	助记符	功能描述
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 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 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 0：正向 1：取反
2	**TXPOL**	发送数据极性配置 0：正向 1：取反
1	**RXEN**	接收使能，1 有效
0	**TXEN**	发送使能，1 有效

23.7.2 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
11	**-**	未实现：读为0
10	**RXERR_IE**	接收错误中断使能，1 有效
9	**-**	未实现：读为0
8	**RXBF_IE**	接收缓存满中断使能，1 有效

位号	助记符	功能描述
7:2	-	未实现：读为 0
1	TXBE_IE	发送缓存空中断使能，1 有效
0	TXSE_IE	发送缓存空且发送移位寄存器空中断使能，1 有效

23.7.3 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 中断标志触发源由 LPUxCR.RXEV 寄存器配置。
23:20	-	未实现：读为0
19	TXOV	发送缓存溢出错误，硬件置位，软件写1清零 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
17	FERR	帧格式错误中断标志，硬件置位，软件写 1 清零
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
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
7:2	-	未实现：读为 0
1	TXBE	发送缓存空中断标志，硬件置位，软件写 1 或者写入 TXBUF 时清零
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

23.7.4 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	MCTL_EN	-			MCTL[11:8]			
位权限	R/W-1	U-0			R/W-0000			
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	**MCTL_EN**	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 当LPUART工作时钟为RCLF时，不需要波特率调制，软件应清零MCTL_EN。
30:28	**-**	未实现：读为0
27:16	**MCTL**	LPUART 每个 bit 的位宽调制控制信号
15:3	**-**	未实现：读为0
2:0	**BAUD**	波特率控制（bps） 000：9600 001：4800 010：2400 011：1200 100：600 101/110/111：300

23.7.5 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

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

23.7.6 LPUARTx 发送数据寄存器（LPUARTx _TXBUF）

名称	LPUARTx _TXBUF(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	-							
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

23.7.7 LPUARTx 数据匹配寄存器（LPUARTx _DMR）

名称	LPUARTx _DMR(x=0,1)							
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**

位号	助记符	功能描述
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。

24 串行外设接口（SPI）

24.1 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。芯片提供了 3 个 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。

特点：

- 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
- 3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（8/16/24/32bits）
- 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

24.2 结构框图

下图为 SPI 模块的结构示意图。

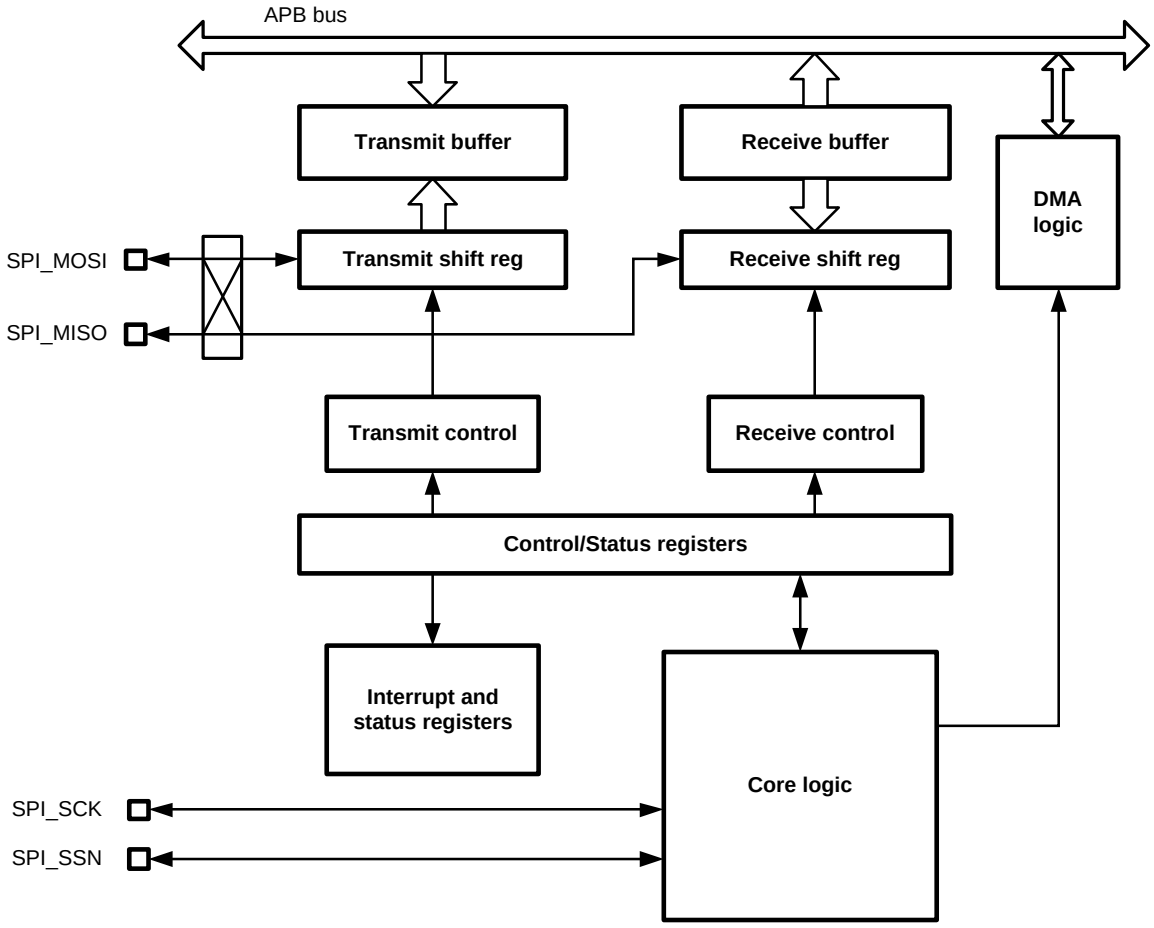


图 24-1 SPI 结构框图

24.3 引脚定义

SPI 模块使用 4 个引脚与外部器件通信，在全双工和半双工模式下，这四个引脚的功能定义有所不同，如下表所示：

引脚	SPIx	全双工	功能	半双工	功能
PB8/PE0	SPI0	SSN	片选信号	SSN	片选信号
PB9/PE1		SCLK	时钟	SCLK	时钟
PB10/PE2		MISO	主机输入从机输出	DCN	命令/数据标识
PB11/PE3		MOSI	主机输出从机输入	SDATA	双向数据
PA8/PD2	SPI1	SSN	片选信号	SSN	片选信号
PA9/PD3		SCLK	时钟	SCLK	时钟
PB0/PD4		MISO	主机输入从机输出	DCN	命令/数据标识
PB1/PD5		MOSI	主机输出从机输入	SDATA	双向数据
PB6/PC7	SPI2	SSN	片选信号	SSN	片选信号
PB7/PC8		SCLK	时钟	SCLK	时钟
PB15/PC9		MISO	主机输入从机输出	DCN	命令/数据标识
PD12/PC10		MOSI	主机输出从机输入	SDATA	双向数据

表 24-1 SPI 引脚对应表

24.4 时钟和复位

SPI 模块工作时钟和总线寄存器时钟都是 APBCLK。

使用 SPI 模块前，必须先清除 RMU 中的 SPIxRST 寄存器以释放复位，并置位 CMU 中的 SPIx_PCE 打开时钟。

24.5 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。

24.5.1 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。

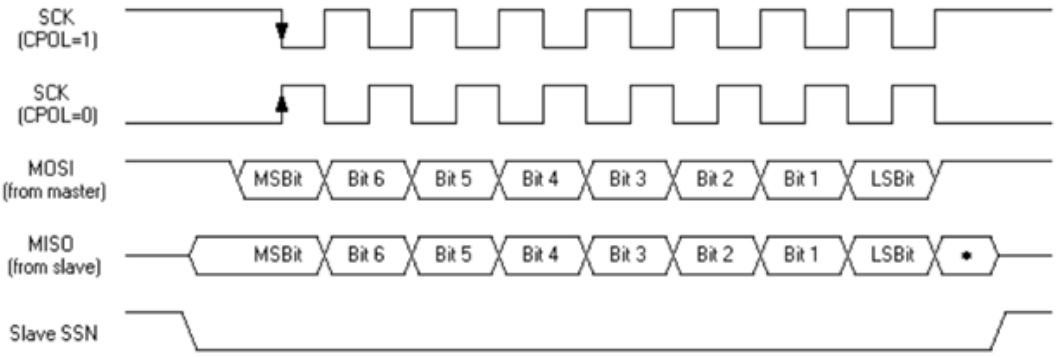


图 24-2 SPI 数据/时钟时序图（CPHA=0）

24.5.2 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。

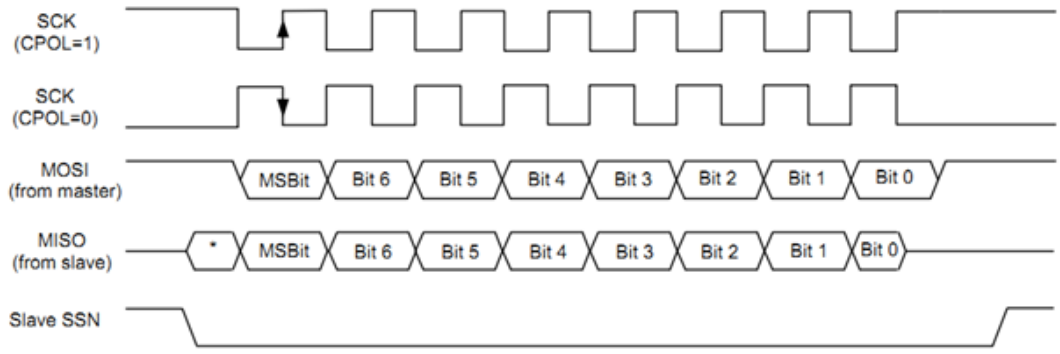


图 24-3 SPI 数据/时钟时序图（CPHA=1）

24.5.1 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。FM33LG0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。

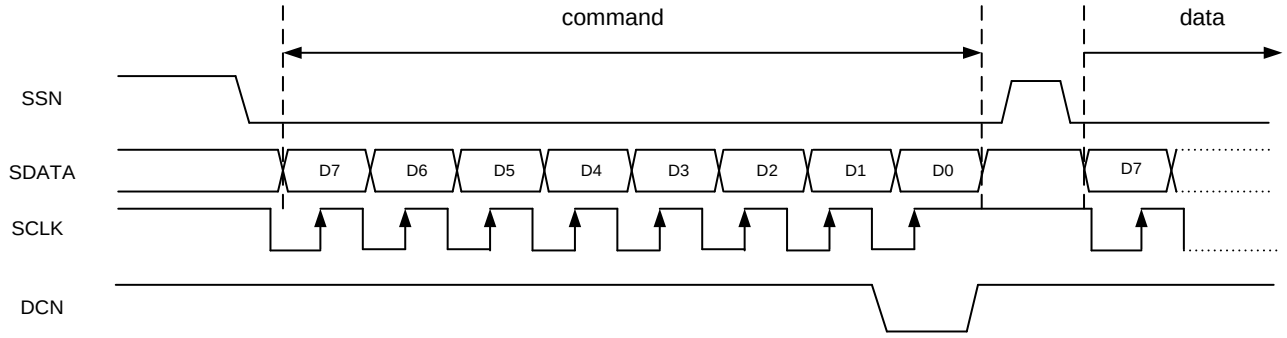


图 24-44 线半双工写操作

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。

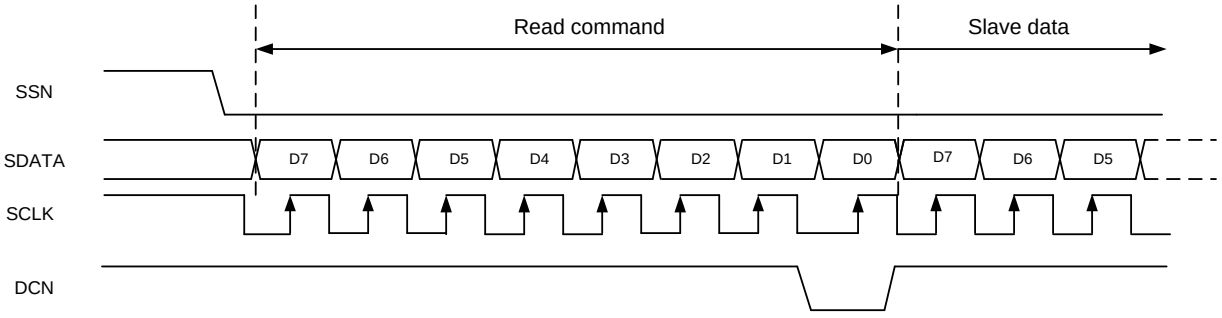


图 24-5 4 线半双工读操作（无 dummy cycle）

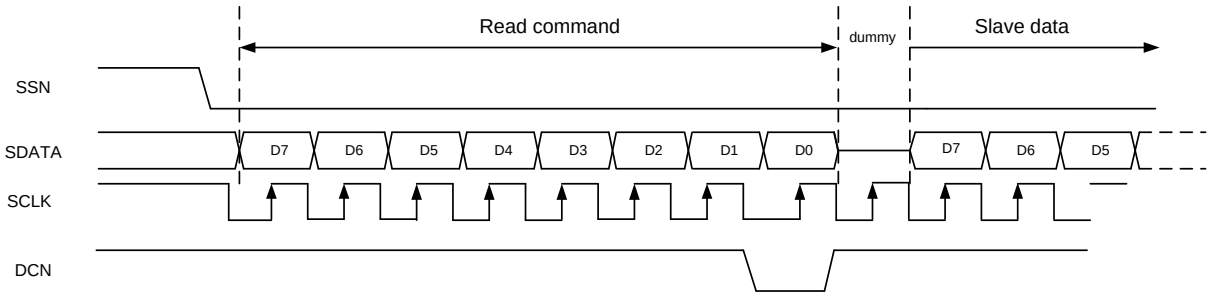


图 24-6 4 线半双工读操作（有 dummy cycle）

24.6 功能描述

24.6.1 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 24-2 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 主从器件的连接如图 24-7 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。

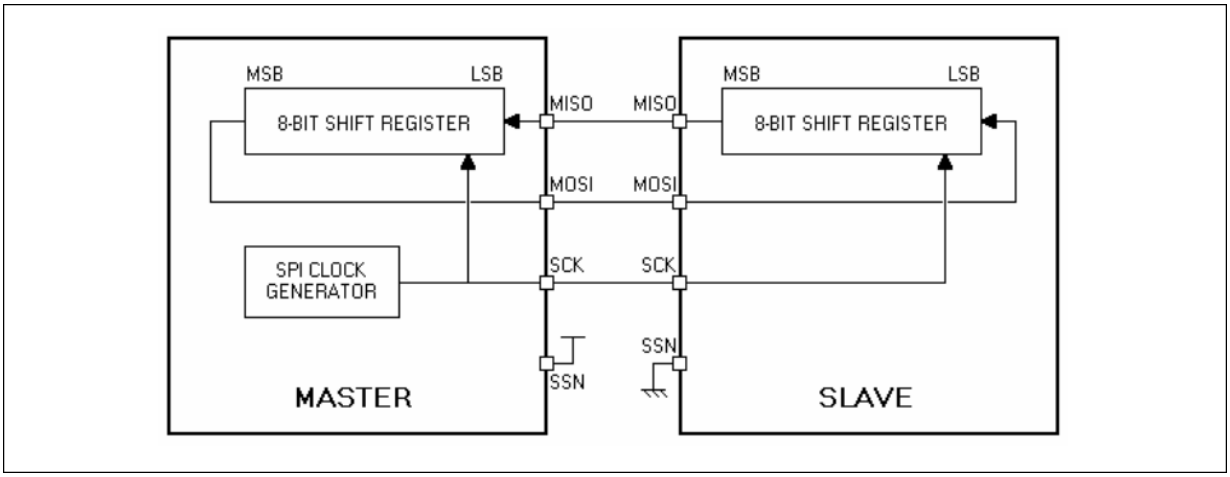


图 24-7 SPI Master/SPI Slave 互连

24.6.2 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24.6.6数据冲突。

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24.6.6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。

24.6.3 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。

24.6.4 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。

24.6.5 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）；

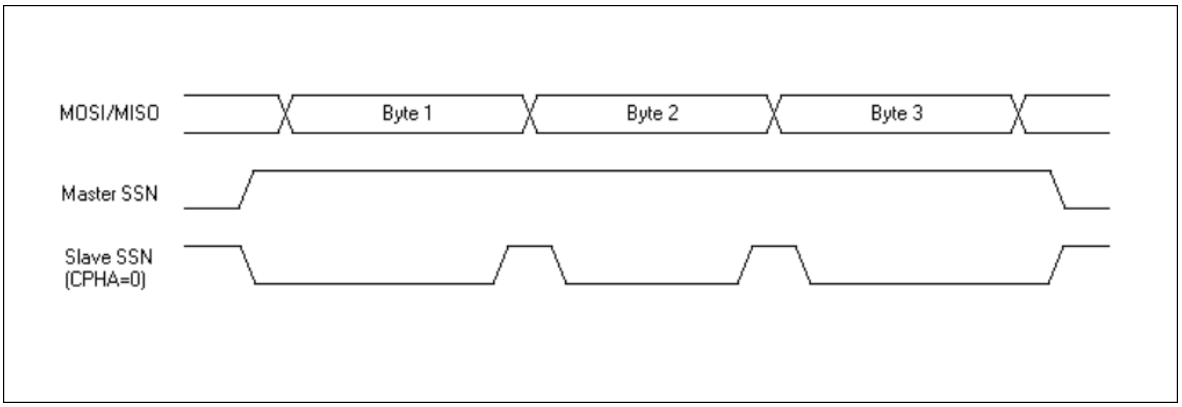


图 24-8 SPI SSN 时序图（SSNM=1，CPHA=0）

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。

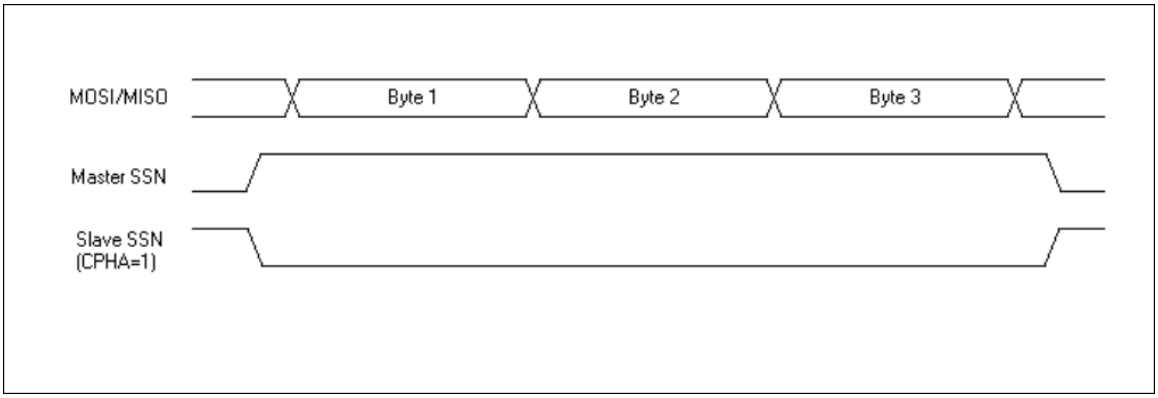


图 24-9 SPI SSN 时序图（SSNM=0）

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。

24.6.6 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。

24.6.7 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 }

24.7 寄存器

offset 地址	名称	符号
SPI0寄存器 (模块基地址：0x40010400)		
0x00	SPI0 控制寄存器 1 （SPI0Control Register1）	SPI0_CR1
0x04	SPI0 控制寄存器 2 （SPI0Control Register2）	SPI0_CR2
0x08	SPI0 控制寄存器 3 （SPI0Control Register3）	SPI0_CR3
0x0C	SPI0 中断使能寄存器 （SPI0Interrupt Enable Register）	SPI0_IER
0x10	SPI0 中断状态寄存器 （SPI0Status Register）	SPI0_ISR
0x14	SPI0 发送数据缓冲寄存器 （SPI0Transmit Buffer）	SPI0_TXBUF
0x18	SPI0 接收数据缓冲寄存器 （SPI0Receive Buffer）	SPI0_RXBUF
SPI1 寄存器 (模块基地址：0x40010800)		
0x00	SPI1 控制寄存器 1 （SPI1Control Register1）	SPI1_CR1
0x04	SPI1 控制寄存器 2 （SPI1Control Register2）	SPI1_CR2
0x08	SPI1 控制寄存器 3 （SPI1Control Register3）	SPI1_CR3
0x0C	SPI1 中断使能寄存器 （SPI1Interrupt Enable Register）	SPI1_IER
0x10	SPI1 中断状态寄存器 （SPI1Status Register）	SPI1_ISR
0x14	SPI1 发送数据缓冲寄存器 （SPI1Transmit Buffer）	SPI1_TXBUF
0x18	SPI1 接收数据缓冲寄存器 （SPI1Receive Buffer）	SPI1_RXBUF
SPI2 寄存器 (模块基地址：0x40014800)		
0x00	SPI2 控制寄存器 1 （SPI2Control Register1）	SPI2_CR1
0x04	SPI2 控制寄存器 2 （SPI2Control Register2）	SPI2_CR2
0x08	SPI2 控制寄存器 3 （SPI2Control Register3）	SPI2_CR3
0x0C	SPI2 中断使能寄存器 （SPI2Interrupt Enable Register）	SPI2_IER
0x10	SPI2 中断状态寄存器 （SPI2Status Register）	SPI2_ISR
0x14	SPI2 发送数据缓冲寄存器 （SPI2Transmit Buffer）	SPI2_TXBUF
0x18	SPI2 接收数据缓冲寄存器 （SPI2Receive Buffer）	SPI2_RXBUF

24.7.1 SPI 控制寄存器 1（SPIx_CR1）

名称	SPIx_CR1 (x=0,1,2)							
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

位号	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 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，每发完 8Bit 后加入至少(1+WAIT)个 SCK cycle 等待时间再传输下一个 8Bit 的数据
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 (Frame format) 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。 1：串行时钟停止在高电平

位号	助记符	功能描述
		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： 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。

24.7.2 SPI 控制寄存器 2（SPIx_CR2）

名称	SPIx_CR2 (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	DUMMY_EN	-			RXO	DLEN		HALFDUPLEX
位权限	R/W-0	U-0			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

位号	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 0：不插入 dummy cycle 1：在读命令之后插入一个 dummy cycle
14:12	**--**	RFU：未实现，读为 0
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 1：启动 Master 的单接收模式 0：关闭单接收模式（收发全双工）
10:9	**DLEN**	通信数据字长配置 00：8bit 01：16bit 10：24bit 11：32bit
8	**HALFDUPLEX**	通信模式选择 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 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)

位号	助记符	功能描述
		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 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 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 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 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。采用关闭时钟的方式来关闭使能。 1：使能 SPI 0：关闭 SPI，清空发送接收缓存

24.7.3 SPI 控制寄存器 3（SPIx_CR3）

名称	SPIx_CR3 (x=0,1,2)							
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

位号	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

24.7.4 SPI 中断控制寄存器（SPIx_IER）

名称	SPIx_IER (x=0,1,2)							
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	-					ERRIE	TXIE	RXIE
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	ERRIE	SPI 错误中断使能
1	TXIE	发送完成中断使能
0	RXIE	接收完成中断使能

24.7.5 SPI 中断标志寄存器（SPIx_ISR）

名称	SPIx_ISR (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	-			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	-	MERR	SERR	-			TXBE	RXBF
位权限	U-0	R-0	R-0	U-0			R-1	R-0

位号	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 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

位号	助记符	功能描述
		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
9	**TXCOL**	发送缓存溢出，软件写 1 清零
8	**BUSY**	SPI 空闲标志，只读 1：SPI 传输进行中 0：SPI 传输空闲 *注意：此标志在 SPI 波特率时钟下置位，当软件使能 SPI 发送后，需等待波特率时钟到来才会置位*
7	--	RFU：未实现，读为 0
6	**MERR**	Master Error 标志 当 Master 下传输未满 8 位 SSN 就被拉高时，MERR 置位
5	**SERR**	Slave Error 标志 当 Slave 下传输未满 8 位 SSN 就被拉高时，SERR 置位
4:2	--	RFU：未实现，读为 0
1	**TXBE**	TX Buffer Empty 标志位 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 1：接收缓存满，软件读 RXBUF 清零 0：接收缓存空

24.7.6 SPI 发送缓存寄存器（SPIx_TXBUF）

名称	SPIx_TXBUF (x=0,1,2)							
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							

位号	助记符	功能描述
31:0	**TXBUF**	SPI 发送缓存

24.7.7 SPI 发送缓存寄存器（SPIx_RXBUF）

名称	SPIx_RXBUF (x=0,1,2)
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							

位号	助记符	功能描述
31:0	**RXBUF**	SPI 接收缓存

25 智能卡接口（ISO7816）

25.1 概述

智能卡接口（7816）是外部智能卡通过 2 线交换 8 位数据的串行同步通讯手段。芯片提供了 2 个 7816 主机接口模块。

- 1路7816接口
- 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

25.2 结构框图

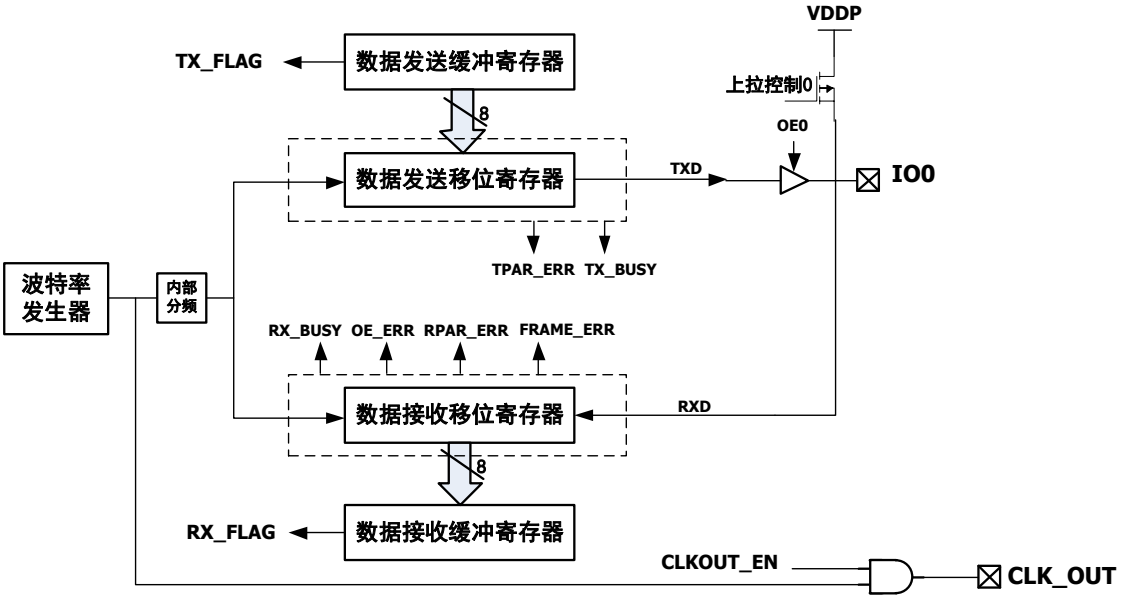


图 25-1 7816 结构框图

25.3 时钟和复位

7816 模块工作时钟和总线寄存器时钟都是 APBCLK。

使用 7816 模块前，必须先清除 RMU 中的 U7816RST 寄存器以释放复位，并置位 CMU 中的 U7816_PCE 打开时钟。

25.4 接口时序

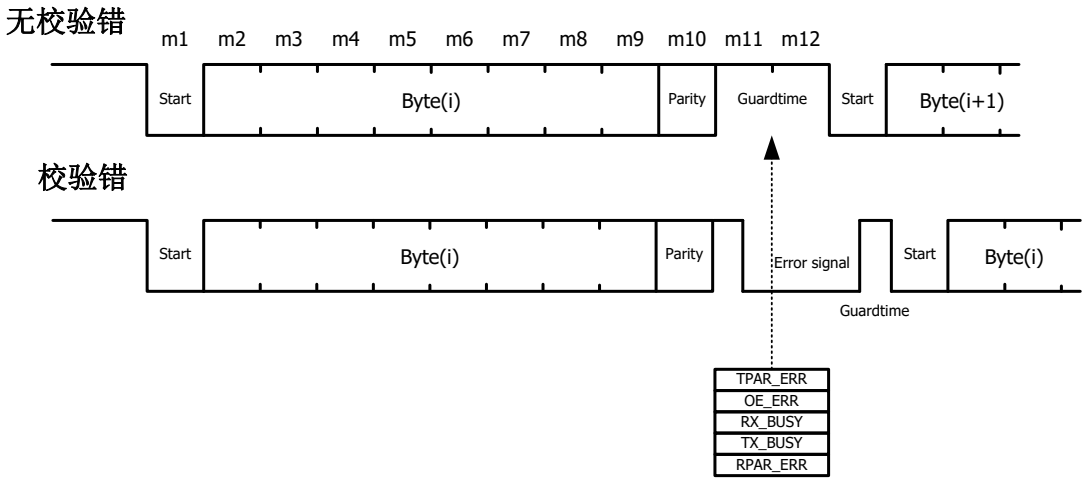


图 25-2 7816 帧结构图

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1ETU或2ETU的GUARDTIME结束。
- 单字节数据长度最小为11ETU或12ETU。
- 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12ETU，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10.5ETU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。
- 所有中断标志尽可能都在同一时刻产生，使得MCU可以正确及时处理中断。

25.5 功能描述

25.5.1 数据接收

7816 数据接收过程：

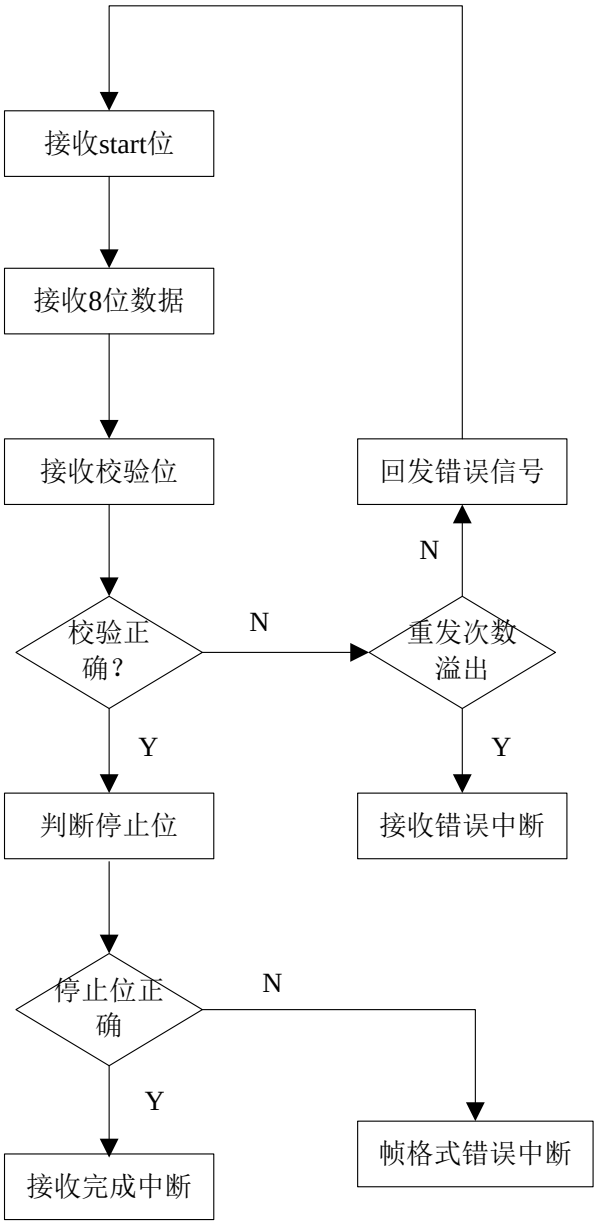


图 25-3 7816 接收流程图

25.5.2 数据发送

在 TXEN 开启的时候，软件只要向 TXBUF 写入数据，硬件在相应 IO 口空闲的条件下会自动发送数据，软件可以在发送过程中向 TXBUF 写入数据，硬件会在前一帧发送结束后继续发送下一帧。

当进行数据发送时，内部输入端口自动关闭，即电路正常应用模式下不能收到自己发出的数据。要注意的是，由于本设计中只有一级缓存，软件两次写 TXBUF 的间隔不能太短，如果在状态机把数据装入移位寄存器开始发送之前又写 TXBUF，会把前面的数据冲掉。注意在发送时，软件至少要等硬件将前一笔数据移入移位寄存器以后才能写下一笔数据，软件可以监视 TX_FLAG，TX_FLAG 为 1 表示发送缓存寄存器空，数据已经进入移位寄存器发送，可以向 TXBUF 写入下一笔数据。

7816 数据发送流程：

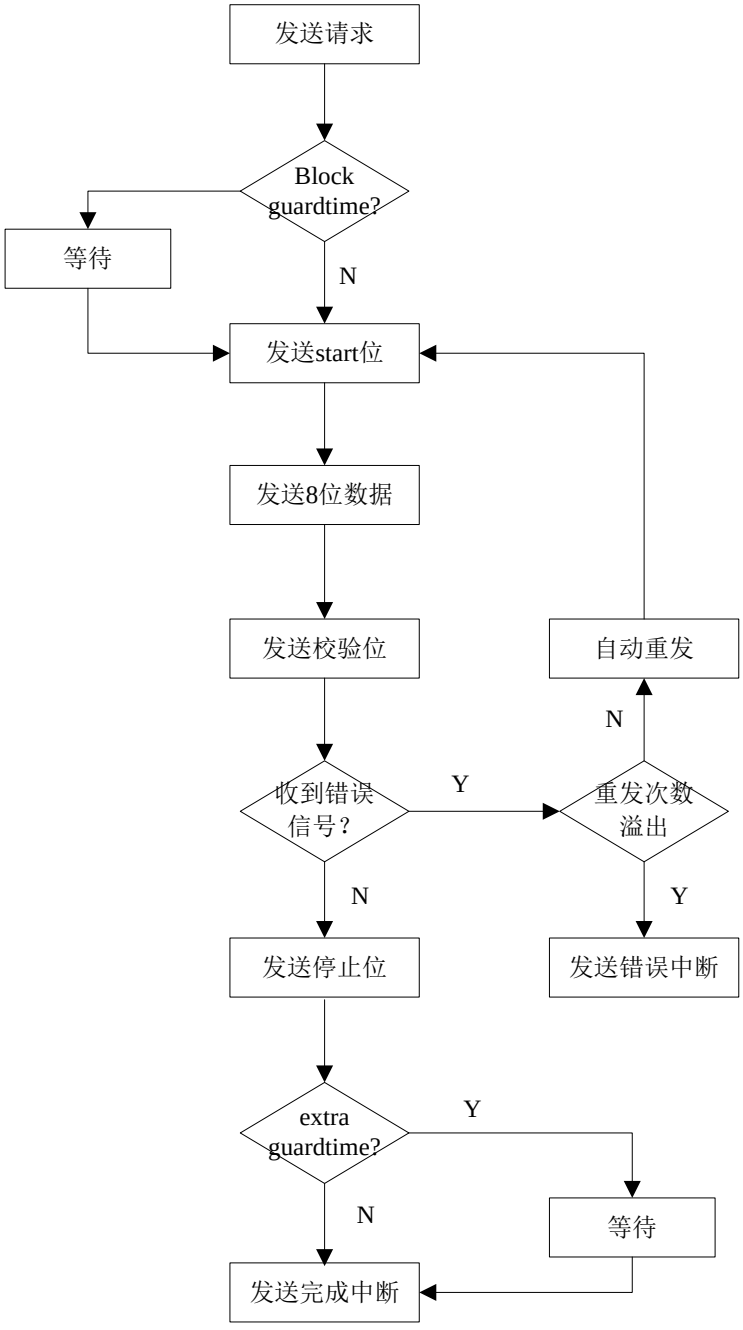


图 25-4 7816 发送流程图

25.5.3 使用 DMA 进行 7816 收发

当 7816 模块被使能后，7816 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 7816 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 7816 请求，并完成 RAM 和 SPI 之间的数据搬运。

应用举例：使用 DMA 进行 7816 接收

- 将 DMA 通道 0 配置为 U7816_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 7816 模块参数
- 使能 7816 模块，等待数据接收
- 收到数据后 7816 自动产生 DMA 请求
- DMA 响应请求，读取 7816 接收缓存寄存器，写入指定 RAM 地址

25.6 寄存器

offset 地址	名称	符号
U7816(模块基地址：0x40011C00)		
0x00	7816 控制寄存器 (7816 Control Register)	U7816_CR
0x04	7816 帧格式控制寄存器 (7816 Frame Format Control Register)	U7816_FFR
0x08	7816 额外保护时间寄存器 (7816 Extra Guard Time Register)	U7816_EGTR
0x0C	7816 工作时钟分频寄存器 (7816 Prescaler Register)	U7816_PSC
0x10	7816 波特率寄存器 (7816 Baud rate Generator Register)	U7816_BGR
0x14	7816 数据接收缓冲寄存器 (7816 data RX Buffer)	U7816_RXBUF
0x18	7816 发送数据缓冲寄存器 (7816 data TX Buffer)	U7816_TXBUF
0x1C	7816 中断使能寄存器 (7816 Interrupt Enable Register)	U7816_IER
0x20	7816 状态标志寄存器 (7816 Interrupt Status Register)	U7816_ISR

25.6.1 U7816 控制寄存器（U7816_CR）

名称	U7816_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	-		TXEN	RXEN	CKOEN	-	-	RFUI
位权限	U-0		R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0

位号	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5	TXEN	U7816 通道发送使能控制位 1 = 通道发送使能，可发送数据 0 = 通道发送禁止，不可发送数据，并关断输出端口
4	RXEN	U7816 通道接收使能控制位。 1 = 通道接收使能，可接收数据 0 = 通道接收禁止，不可接收数据，并关断输入端口
3	CKOEN	U7816 时钟 CLK 输出使能控制位

位号	助记符	功能描述
		1 = 7816 时钟输出使能 0 = 7816 时钟输出禁止
2	--	RFU：未实现，读为 0
1	--	RFU：未实现，读为 0
0	RFUI	保留位

25.6.2 U7816 帧格式控制寄存器（U7816_FFR）

名称	U7816_FFR							
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	-					ERSW		ERSGD
位权限	U-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

位号	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12:11	DUMMY	可读写，无功能
10:9	ERSW	ERROR SIGNAL 宽度选择 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） 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（block guard time）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 1 = BGT 使能，插入 Block guard time(22 etu); 0 = BGT 禁止，不插入 Block guard time(22 etu);
6	REP_T	控制接收数据奇偶校验出错时自动重发次数 1 = 3 次 0 = 1 次
5:4	PAR	奇偶校验类型选择 00：Even 01：Odd

位号	助记符	功能描述
		10：Always 1 11：不校验，处理
3	FREN	Guard Time 长度控制位（发送时严格按照协议 2etu） 1 = Guard time 为 1 etu 0 = Guard time 为 2 etu
2	TREPEN	缓发送数据奇偶校验错的处理方式选择 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 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 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

25.6.3 U7816 额外保护时间寄存器（U7816_EGTR）

名称	U7816_EGTR							
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	TXEGT							
位权限	R/W-00000000							

位号	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）

25.6.4 U7816 工作时钟分频寄存器（U7816_PSC）

名称	U7816_PSC							
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	-			CLKDIV				
位权限	U-0			R/W-00011				

位号	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，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。

25.6.5 U7816 波特率寄存器（U7816_BGR）

名称	U7816_BGR							
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	-				PDIV			
位权限	U-0				R/W-0001			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PDIV							
位权限	R/W-01110011							

位号	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11:0	PDIV	U7816 预分频控制寄存器，控制 7816 通信分频比（波特率） Baud ＝ F_{7816}/(PDIV ＋ 1)

25.6.6 U7816 数据接收缓冲寄存器（U7816_RXBUF）

名称	U7816_RXBUF							
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	RXBUF							
位权限	R/W-00000000							

位号	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	RXBUF	U7816 数据接收缓存寄存器

25.6.7 U7816 数据发送缓冲寄存器（U7816_TXBUF）

名称	U7816_TXBUF							
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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF							
位权限	R/W-00000000							

位号	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	TXBUF	U7816 数据发送缓存寄存器

25.6.8 U7816 中断使能寄存器（U7816_IER）

名称	U7816_IER							
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						RXIE	TXIE	LSIE
位权限						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	RXIE	数据接收中断使能位。对应 RX_FLAG 中断标志位 1 = 数据接收中断使能位 0 = 数据接收中断使能位
1	TXIE	数据发送中断使能位。对应 TX_FLAG 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 ERROR_FLAG 中断标志位 1 = 线路状态中断使能 0 = 线路状态中断禁止

25.6.9 U7816 状态标志寄存器（U7816_ISR）

名称	U7816_ISR							
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	-					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	-					RXIF	TXIF	ERRIF
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	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; 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; 软件可以查询此位，但是不能改写;
17	TXBUSY	发送数据忙标志。（发送完成后自动清零） 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） 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

位号	助记符	功能描述
		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
10	RPARERR	接收数据奇偶校验错误标志位。硬件置位，写 1 清零
9	FRERR	接收帧格式错误标志位。硬件置位，写 1 清零 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 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
7:3	--	RFU：未实现，读为 0
2	RXIF	接收完成标志，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，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，寄存器配置出错或传输过程中出错。硬件置位，清 U7816ERR 中相应错误标志清零 1 = 出错，错误类型见 U7816ERR 寄存器 0 = 无出错情况

26 控制器区域网络（CAN）

26.1 概述

CAN模块用于CAN总线数据收发，支持CAN2.0A和2.0B协议

CAN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
- 2 messages发送FIFO（32字节），2 messages接收FIFO（32字节）
- 1个高优先级发送buffer
- 支持错误或仲裁失败条件下的自动重发
- 4个接收滤波器
- 支持loopback模式
- 支持自动唤醒
- 错误计数器

26.2 结构框图

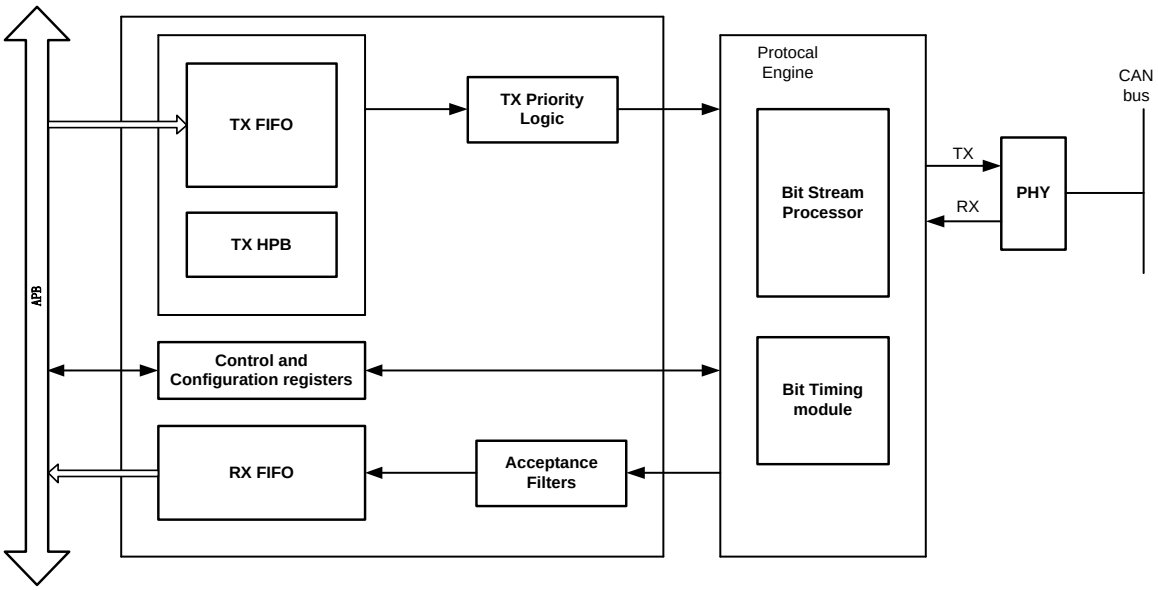


图 26-1 CAN 模块结构框图

CAN模块需要外接一个CAN PHY芯片用于实现物理层数据收发。

26.3 引脚定义

CAN 模块使用 2 个引脚与外部器件通信，收发信号可能被映射到不同的 GPIO 上。

下表为 FM33LG0x8 的引脚映射关系

引脚			符号	功能
PA6	PC15	PD10	CAN_RX	CAN 接收引脚
PA7	PE5	PE9	CAN_TX	CAN 发送引脚

表 26-1 CAN 引脚列表

注意：PA16，PC15，PD10都可以作为CAN输入引脚。如果这三个引脚都被配置为CAN功能，则优先级PD10>PC15>PA6

26.4 功能描述

26.4.1 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可以选择使用芯片内部高频环振、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。

26.4.2 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 26-2 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。

26.4.3 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。

26.4.4 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

CAN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，TX FIFO和TX HPB中没有剩余消息，并且CAN总线IDLE，CAN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 FIFO或TX HPB写入数据，CAN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，CAN控制器可以进入Configuration或者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。

26.4.5 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 FIFO或HPB中，其中HPB为高优先级Buffer，HPB中的消息帧总是优先于TX FIFO中的消息帧被发送。HPB只能保存一条消息，而TX FIFO可以保存2条消息。

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：

- TX HPB中的消息优先级高于TX FIFO
- 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 FIFO、HPB、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 FIFO或HPB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。

注意：CAN_TXF_IDR, CAN_TXF_DLCR, CAN_TXF_DW1R, CAN_TXF_DW2R寄存器为FIFO形式，对这四个地址的写都会被按顺序映射到FIFO push操作；举例来说，软件对CAN_TXF_DLCR连续进行4次写，硬件电路也会依次对ID、DLC、DW1、DW2执行4次写入。为避免混淆，要求软件严格按照顺序对发送FIFO执行连续写入。

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

26.4.6 消息滤波器（Acceptance Filter）

CAN控制器带有4个消息滤波器，每个消息滤波器包含一组Mask寄存器和一组ID寄存器。

消息滤波工作步骤如下：

- 接收到的消息ID先与Mask寄存器进行逻辑与
- 滤波器ID寄存器内容同样与Mask寄存器进行逻辑与
- 两个逻辑与操作结果进行比较
- 如果两个结果相等，本消息通过滤波器，被存入RX FIFO

- 4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。

26.5 编程模型

CAN总线模块编程模型用于指导软件编程和应用开发，包含寄存器配置方法和消息传输。

26.5.1 寄存器配置

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

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

配置中断使能寄存器IER以选择中断状态寄存器ISR中可生成中断的位

置位CAN_SRR寄存器中的CEN位来启用CAN

26.5.2 消息传输

要发送的消息可以写入TX FIFO或TX HPB。TX HPB中的消息优先于TX FIFO中的消息。CAN_ISR中的TXOK位是在CAN成功传输消息后自动置位的。

- 向TX FIFO写入消息

所有写入TX FIFO的消息应遵循之前 “消息帧存储和消息帧结构”中定义的格式。

执行消息写入：

1、轮询CAN_SR中的TXFLL位。当TXFLL位为“0”时，可以将消息写入TX FIFO。

2、将消息的ID写入发送FIFO ID寄存器内

3、将消息的数据长度写入发送 FIFO DLC寄存器内

4、将消息DW1写入发送FIFO DataWord1寄存器内

5、将消息DW2写入发送FIFO DataWord2寄存器内

消息可以连续写入TX FIFO直到TX FIFO满。当TX FIFO满时，硬件置位CAN_ISR中的TXFLL位和CAN_SR中的TXFLL位。如果软件采用轮询方式，则应在每次写入后轮询状态寄存器中的TXFLL位。如果软件使用中断方式，可以连续写入，直到在ISR中的TXFLL位产生中断。

- 向TX HPB写入消息

所有写入TX FIFO的消息应遵循之前 “消息帧存储和消息帧结构”中定义的格式。

向TX HPB写入消息：

1、轮询CAN_SR中的TXBFLL位。当TXBFLL位为“0”时，可以将消息写入TX HPB。

2、将消息的ID写入高优先级发送缓冲 ID寄存器内

3、将消息的数据长度写入高优先级发送缓冲DLC寄存器内

4、将消息DW1写入高优先级发送缓冲DataWord1寄存器内

5、将消息DW2写入高优先级发送缓冲DataWord2寄存器内

每次写入TX HPB后，状态寄存器CAN_SR中的TXBFLL位和中断状态寄存器CAN_ISR中的TXBFLL位置1。

- 接收消息

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。

26.6 寄存器

offset 地址	名称	符号
CAN(模块起始地址：0x40019400)		
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	位时序寄存器 (CAN Bit Timing Register)	CAN_BTR
0x10	错误计数寄存器 (CAN Error Counter Register)	CAN_ECR
0x14	错误状态寄存器 (CAN Error Status Register)	CAN_ESR
0x18	状态标志寄存器 (CAN Status Register)	CAN_SR
0x1C	中断标志寄存器 (CAN Interrupt Status Register)	CAN_ISR
0x20	中断使能寄存器 (CAN Interrupt Enable Register)	CAN_IER
0x24	中断清除寄存器 (CAN Interrupt Clear Register)	CAN_ICR
0x28	-	-
0x2C	-	-
0x30	发送 FIFO ID 寄存器 (CAN TX FIFO ID Register)	CAN_TXF_IDR
0x34	发送 FIFO DLC 寄存器 (CAN TX FIFO DLC Register)	CAN_TXF_DLCR
0x38	发送 FIFO Data Word1 寄存器 (CAN TX FIFO Data Word1 Register)	CAN_TXF_DW1R
0x3C	发送 FIFO Data Word2 寄存器 (CAN TX FIFO Data Word2 Register)	CAN_TXF_DW2R
0x40	高优先级发送缓冲 ID 寄存器 (CAN TX HPB ID Register)	CAN_HPB_IDR
0x44	高优先级发送缓冲 DLC 寄存器 (CAN TX HPB DLC Register)	CAN_HPB_DLCR
0x48	高优先级发送缓冲 Data Word1 寄存器 (CAN TX HPB Data Word1 Register)	CAN_HPB_DW1R
0x4C	高优先级发送缓冲 Data Word2 寄存器 (CAN TX HPB Data Word2 Register)	CAN_HPB_DW2R
0x50	接收 FIFO ID 寄存器 (CAN RX FIFO ID Register)	CAN_RXF_IDR
0x54	接收 FIFO DLC 寄存器 (CAN RX FIFO DLC Register)	CAN_RXF_DLCR
0x58	接收 FIFO Data Word1 寄存器 (CAN RX FIFO Data Word1 Register)	CAN_RXF_DW1R
0x5C	接收 FIFO Data Word2 寄存器 (CAN RX FIFO Data Word2 Register)	CAN_RXF_DW2R

0x60	接收滤波寄存器 (Acceptance Filter Register)	CAN_AFR
0x64	接收滤波掩码寄存器 1 (Acceptance Filter Mask Register1)	CAN_AFMR1
0x68	接收滤波 ID 寄存器 1 (Acceptance Filter ID Register1)	CAN_AFIR1
0x6C	接收滤波掩码寄存器 2 (Acceptance Filter Mask Register2)	CAN_AFMR2
0x70	接收滤波 ID 寄存器 2 (Acceptance Filter ID Register2)	CAN_AFIR2
0x74	接收滤波掩码寄存器 3 (Acceptance Filter Mask Register3)	CAN_AFMR3
0x78	接收滤波 ID 寄存器 3 (Acceptance Filter ID Register3)	CAN_AFIR3
0x7C	接收滤波掩码寄存器 4 (Acceptance Filter Mask Register4)	CAN_AFMR4
0x80	接收滤波 ID 寄存器 4 (Acceptance Filter ID Register4)	CAN_AFIR4

26.6.1 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

位号	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

26.6.2 CAN 模式选择寄存器（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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						LPBACK	SLEEP
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	--	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

26.6.3 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							

位号	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

位号	助记符	功能描述
		实际分频值是 BPR+1 Tq=tCAN_CLK * (BPR+1)

26.6.4 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			

位号	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)

26.6.5 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

位号	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) 软件只读，每次发生接收错误时 REC 加 1
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 加 1

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恢复阶段中，CAN控制器收到128个L1隐性位后进入Error Active

26.6.6 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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			ACKER	BERR	STER	FMER	CRCER
位权限	U-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
31:5	--	RFU：未实现，读为 0
4	ACKER	Acknowledge Error 1：检测到 ACK 错误 0：无 ACK 错误 硬件置位，软件写 1 清零
3	BERR	Bit Error 1：接收到的 bit 不同于正在发送的 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

位号	助记符	功能描述
		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

26.6.7 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	TXFLL	TXBFLL	ESTAT[1]
位权限	U-0				R-0	R-0	R-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ESTAT[0]	ERRWRN	BBSY	BIDLE	NORMAL	SLEEP	LPBACK	CONFIG
位权限	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-1

位号	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	TXFLL	发送 FIFO 满 (Transmit FIFO is FULL) 1：TX FIFO 满 0：TX FIFO 不满
9	TXBFLL	高优先级发送缓冲区满 (High Priority Transmit Buffer is FULL) 1：HPB 满 0：HPB 不满
8:7	ESTAT	Error Status，只读 00：CAN 控制器处于 Configuration mode 01：Error Active state 10：Bus off state 11：Error Passive state
6	ERRWRN	Error Warning，只读

位号	助记符	功能描述
		1：TEC 或 REC 计数值大于等于 96 0：TEC 和 REC 计数值都小于 96
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

26.6.8 CAN 中断标志寄存器（CAN_ISR）

名称	CAN_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	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				WKUPIF	SLEEPIF	BSOFFIF	ERRORIF
位权限	U-0				R-0	R-0	R-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXNEMPIF	RXOFLWIF	RXUFLWIF	RXOKIF	TXBFLLIF	TXFLLIF	TXOKIF	ARBLSTIF
位权限	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

位号	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11	WKUPIF	Wakeup 中断标志，硬件置位，软件写 ICR 寄存器清零 1：CAN 控制器从 SLEEP 模式进入 Normal 模式
10	SLPIF	SLEEP 中断标志，硬件置位，软件写 ICR 寄存器清零 1：CAN 控制器进入 SLEEP 模式
9	BSOFFIF	Bus Off 中断标志，硬件置位，软件写 ICR 寄存器清零

位号	助记符	功能描述
		1：CAN 控制器进入 Bus Off 状态
8	ERRORIF	错误中断标志，硬件置位，软件写 ICR 寄存器清零 1：消息收发过程中发生了错误
7	RXNEMPIF	Receive FIFO Not Empty 中断标志，硬件置位，软件写 ICR 寄存器清零 1：接收 FIFO 非空
6	RXOFLWIF	Receive FIFO Overflow 中断标志，硬件置位，软件写 ICR 寄存器清零 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 ICR 寄存器清零 1：接收 FIFO 下溢出，即在接收 FIFO 空的情况下执行读操作
4	RXOKIF	Receive OK 中断标志，硬件置位，软件写 ICR 寄存器清零 1：表示 1 个消息帧被成功接收
3	TXBFLLIF	High Priority Transmit Buffer FULL 中断标志，硬件置位，软件写 ICR 寄存器清零 1：表示高优先级发送 BUFFER 满
2	TXFLLIF	Transmit FIFO FULL 中断标志，硬件置位，软件写 ICR 寄存器清零 1：表示发送 FIFO 满
1	TXOKIF	Transmission OK 中断标志，硬件置位，软件写 ICR 寄存器清零 1：表示 1 个消息帧被成功发送
0	ARBLSTIF	Arbitration Lost 中断标志，硬件置位，软件写 ICR 寄存器清零 1：仲裁失败

26.6.9 CAN 中断使能寄存器（CAN_IER）

名称	CAN_IER							
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	-				EWKUPIE	ESLEEPIE	EBSOFFIE	EERRORIE
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	ERXNEMPIE	ERXOFLWIE	ERXUFLWIE	ERXOKIE	ETXBFLLIE	ETXFLLIE	ETXOKIE	EARBLSTIE
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
31:12	--	RFU：未实现，读为 0
11	WKUPIE	Wakeup 中断使能，1 允许中断，0 禁止中断

位号	助记符	功能描述
10	SLPIE	SLEEP 中断使能，1 允许中断，0 禁止中断
9	BSOFFIE	Bus Off 中断使能，1 允许中断，0 禁止中断
8	ERRORIE	错误中断使能，1 允许中断，0 禁止中断
7	RXNEMPIE	Receive FIFO Not Empty 中断使能，1 允许中断，0 禁止中断
6	RXOFLWIE	Receive FIFO Overflow 中断使能，1 允许中断，0 禁止中断
5	RXUFLWIE	Receive FIFO Underflow 中断使能，1 允许中断，0 禁止中断
4	RXOKIE	Receive OK 中断使能，1 允许中断，0 禁止中断
3	TXBFLLIE	High Priority Transmit Buffer FULL 中断是能，1 允许中断，0 禁止中断
2	TXFLLIE	Transmit FIFO FULL 中断使能，1 允许中断，0 禁止中断
1	TXOKIE	Transmission OK 中断使能，1 允许中断，0 禁止中断
0	ARBLSTIE	Arbitration Lost 中断使能，1 允许中断，0 禁止中断

26.6.10 CAN 中断清除寄存器（CAN_ICR）

名称	CAN_ICR							
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	-				CWKUP	CSLEEP	CBSOFF	CERROR
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	CRXNEMP	CRXOFLW	CRXUFLW	CRXOK	CTXBFLL	CTXFLL	CTXOK	CARBLST
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
31:12	--	RFU：未实现，读为 0
11	CWKUP	Clear Wakeup，写 1 清零 WKUP
10	CSLP	Clear SLEEP，写 1 清零 SLP
9	CBSOFF	Clear Bus Off，写 1 清零 BSOFF
8	CERROR	Clear ERROR，写 1 清零 ERROR
7	CRXNEMP	Clear Receive FIFO Not Empty，写 1 清零 RXNEMP
6	CRXOFLW	Clear Receive FIFO Overflow，写 1 清零 RXOFLW
5	CRXUFLW	Clear Receive FIFO Underflow，写 1 清零 RXUFLW
4	CRXOK	Clear Receive OK，写 1 清零 RXOK
3	CTXBFLL	Clear High Priority Transmit Buffer FULL，写 1 清零 TXBFLL
2	CTXFLL	Clear Transmit FIFO FULL，写 1 清零 TXFLL
1	CTXOK	Clear Transmission OK，写 1 清零 TXOK
0	CARBLST	Clear Arbitration Lost，写 1 清零 ARBLST

26.6.11 CAN 发送 FIFO ID 寄存器（CAN_TXFIDR）

名称	CAN_TXFIDR							
offset	0x30							
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							

位号	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 26.4.5 消息帧存储和消息帧结构（Message storage and structure）

26.6.12 CAN 发送 FIFO DLC 寄存器（CAN_TXFDLCR）

名称	CAN_TXFDLCR							
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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				DLC			
位权限	U-0				W-0000			

位号	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

26.6.13 CAN 发送 FIFO DataWord1 寄存器（CAN_TXFDW1R）

名称	CAN_TXFDW1R							
offset	0x38							
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							

位号	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

26.6.14 CAN 发送 FIFO DataWord2 寄存器（CAN_TXFDW2R）

名称	CAN_TXFDW2R							
offset	0x3C							
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							

位号	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

26.6.15 CAN 高优先级发送缓冲 ID 寄存器（CAN_HPBIDR）

名称	CAN_HPBIDR							
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							

位号	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 26.4.5 消息帧存储和消息帧结构（Message storage and structure）

26.6.16 CAN 高优先级发送缓冲 DLC 寄存器（CAN_HPBDLCR）

名称	CAN_HPBDLCR							
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			

位号	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

26.6.17 CAN 高优先级发送缓冲 DataWord1 寄存器（CAN_HPBDW1R）

名称	CAN_HPBDW1R

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							

位号	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

26.6.18 CAN 高优先级发送缓冲 DataWord2 寄存器（CAN_HPBDW2R）

名称	CAN_HPBDW2R							
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							

位号	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

26.6.19 CAN 接收 FIFO ID 寄存器（CAN_RXF_IDR）

名称	CAN_RXFIDR
offset	0x50

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							

位号	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 26.4.5 消息帧存储和消息帧结构（Message storage and structure）

26.6.20 CAN 接收 FIFO DLC 寄存器（CAN_RXFDLCR）

名称	CAN_RXFDLCR							
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	-				DLC			
位权限	U-0				W-0000			

位号	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

26.6.21 CAN 接收 FIFO DataWord1 寄存器（CAN_RXFDW1R）

名称	CAN_RXFDW1R							
offset	0x58							
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							

位号	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

26.6.22 CAN 接收 FIFO DataWord2 寄存器（CAN_RXFDW2R）

名称	CAN_RXFDW2R							
offset	0x5C							
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							

位号	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

26.6.23 CAN 接收滤波寄存器（CAN_AFR）

名称	CAN_AFR							
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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				UAF4	UAF3	UAF2	UAF1
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	UAF4	使用 4 号消息滤波器 (Use Acceptance Filter 4) 1：使能 4 号滤波器 0：关闭 4 号滤波器
2	UAF3	使用 3 号消息滤波器 (Use Acceptance Filter 3) 1：使能 3 号滤波器 0：关闭 3 号滤波器
1	UAF2	使用 2 号消息滤波器 (Use Acceptance Filter 2) 1：使能 2 号滤波器 0：关闭 2 号滤波器
0	UAF1	使用 1 号消息滤波器 (Use Acceptance Filter 1) 1：使能 1 号滤波器 0：关闭 1 号滤波器

26.6.24 CAN 接收滤波掩码寄存器 x（CAN_AFMRx）

名称	CAN_AFMRx (x=1,2,3,4)							
offset	0x64 + (x-1)*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							

位号	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

位号	助记符	功能描述
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

26.6.25 CAN 接收滤波 ID 寄存器 x（CAN_AFIRx）

名称	CAN_AFIRx (x=1,2,3,4)							
offset	0x68 + (x-1)*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							

位号	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)

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65536字节（64KB），支持byte/half-word/word传输
- Flash->RAM通道最大传输长度8192字节，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。

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

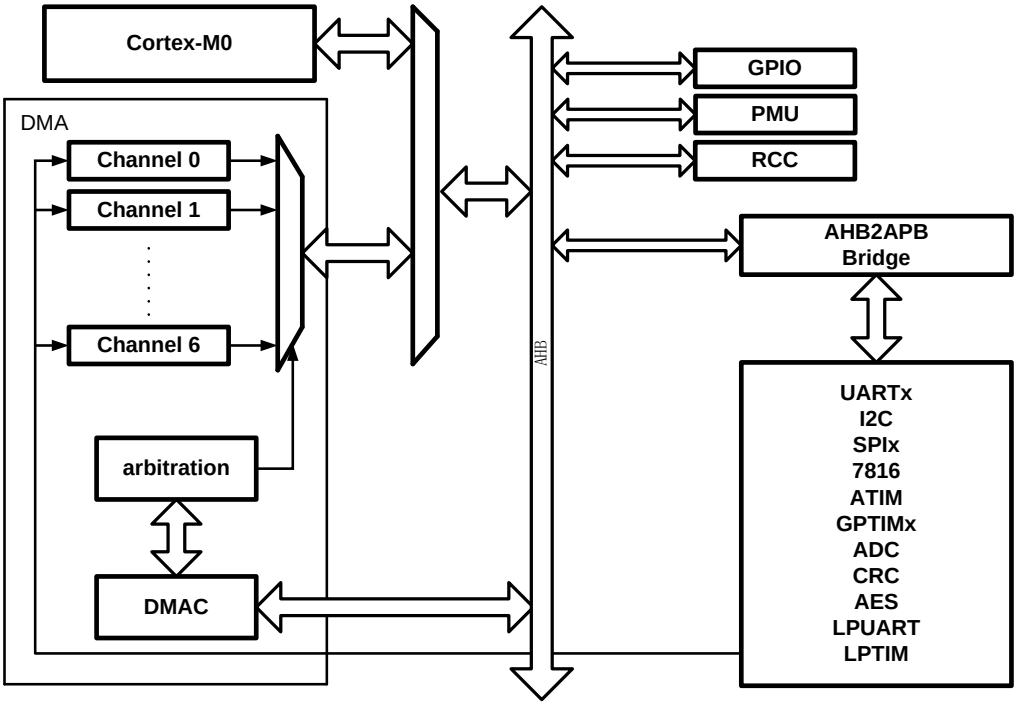


图 27-1 DMA 结构框图

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，若是则关闭通道使能；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：

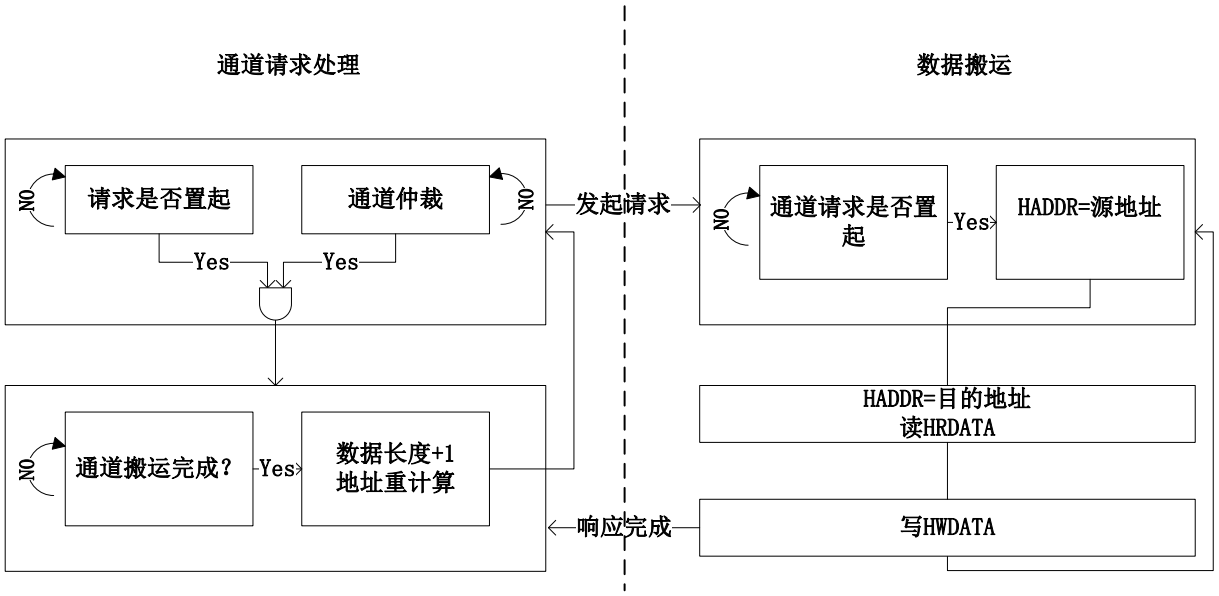


图 27-3 DMA 工作流程

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		
1	***SPI0***		SPI0_RX	SPI0_TX				
2	***SPI1***				SPI1_RX	SPI1_TX	SPI1_RX	SPI1_TX
3	***SPI2***			SPI2_RX		SPI2_TX	SPI2_RX	SPI2_TX
4	***UART0***		RXD0	TXD0	RXD0	TXD0		
5	***UART1***				RXD1	TXD1	RXD1	TXD1
6	***UART3***						RXD3	TXD3
7	***UART4***			RXD4	TXD4			
8	***UART5***					RXD5		TXD5
9	***LPUART0***	LPUART0_RX	LPUART0_TX					
10	***LPUART1***	LPUART1_TX		LPUART1_RX				
11	***LPUART2***				LPUART2_RX		LPUART2_TX	
12	***U7816***						U7816RX	U7816TX
13	***I2C***					I2C_RX		I2C_TX
14	***AES***	AES_IN	AES_OUT					
15	***CRC***	CRC						
16	***ATIM***	ATIM_CH1	ATIM_CH2	ATIM_CH3	ATIM_CH4	ATIM_TRIG ATIM_COM ATIM_UEV		
17	***GTIM1***	GTIM1_CH1	GTIM1_CH2	GTIM1_CH3	GTIM1_CH4			GTIM_TRIG GTIM1_UEV
18	***GTIM2***	GTIM2_CH1	GTIM2_CH2	GTIM2_CH3	GTIM2_CH4		GTIM2_TRIG GTIM2_UEV	
19	***DAC***		DAC				DAC	
		8	8	8	8	8	8	8

注意，ATIM_TRIG、ATIM_COM和ATIM_UEV请求，都仅针对高级定时器的DMA Burst模式，即这些请求到来时，DMA都是要访问ATIM的DMAR寄存器，因此这三个请求可以合并到一个通道上完成；同理，通用定时器的GTIMx_TRIG和GTIMx_UEV请求也可以合并到一个通道上。

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UART1_TX，即UART1的数据发送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

offset 地址	名称	符号
DMA(模块基地址：0x40000400)		
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_CH6MAD
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_CH7FLSAD
0x44	通道 7 RAM 指针寄存器 （Channel 11 RAM Address Register）	DMA_CH7RAMAD
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	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

名称	DMA_GCR							
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_IE	DMAEN
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	**ADDRERR_IE**	DMA 错误地址中断使能 1：允许错误地址中断 0：禁止错误地址中断
0	**DMAEN**	DMA 全局使能 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	x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CHxTSIZE							
位权限	R/W-00000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CHxTSIZE							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		CHxPRI		CHxINC	CHxSSEL		
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	-	CHxDIR	CHxBDW		CHxCIRC	CHxFTIE	CHxHTIE	ChxEN
位权限	U-0	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	**CHxTSIZE**	Channelx 传输长度，1-65536 次传输
15:14	--	RFU：未实现，读为 0
13:12	**CHxPRI**	Channelx 优先级 00：Low 01：Medium 10：High 11：Very High
11	**CHxINC**	RAM 地址增减设置 1：RAM 地址递增 0：RAM 地址递减

位号	助记符	功能描述
10:8	**CHxSSEL**	Channelx 外设请求映射 每个通道可以接受 8 个外设请求，外设请求的映射参见 27.6.1DMA 请求映射
7	**--**	RFU：未实现，读为 0
6	**CHxDIR**	通道传输方向 0：从外设读取数据写入 RAM 1：从 RAM 读取数据写入外设
5:4	**CHxBDW**	传输带宽设置 00：字节，8bit 01：半字，16bit 10：字，32bit 11：RFU
3	**CHxCIRC**	循环缓冲模式 0：关闭循环模式 1：使能循环模式
2	**CHxFTIE**	Channelx 传输完成中断使能 1：使能传输完成中断 0：关闭传输完成中断
1	**CHxHTIE**	Channelx 半程传输完成中断使能 1：使能半程中断 0：关闭半程中断
0	**ChxEN**	Channelx 使能 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_CHxMAR）

名称	DMA_CHxMAR (x=0~6)							
offset	x*0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CHxMEMAD[31:24]							
位权限	R/W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CHxMEMAD[23:16]							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CHxMEMAD[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CHxMEMAD[7:0]							
位权限	R/W-00000000							

位号	助记符	功能描述
31:0	**CHxMEMAD**	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。 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	0x3C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				MCHTSIZE			
位权限	U-0				R/W-00000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MCHTSIZE							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		MCHPRI		-	MCHDIR	MCHRI	MCHFI
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	-					MCHFTIE	MCHHTIE	MCHEN
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:28	--	RFU：未实现，读为 0
27:16	MCHTSIZE	Channel7 传输长度，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	MCHPRI	Channel7 优先级 00：Low 01：Medium 10：High 11：Very High
11	--	RFU：未实现，读为 0
10	MCHDIR	Channel7 传输方向 1：Flash->RAM 传输 0：RAM->Flash 传输
9	MCHRI	Channel7 RAM 地址增减设置，仅在 Flash->RAM 传输中有效 1：RAM 地址递增 0：RAM 地址递减
8	MCHFI	Channel7 Flash 地址增减设置，仅在 Flash->RAM 传输中有效 1：Flash 地址递增 0：Flash 地址递减
7:3	--	RFU：未实现，读为 0
2	MCHFTIE	Channel7 传输完成中断使能 1：使能传输完成中断 0：关闭传输完成中断
1	MCHHTIE	Channel7 半程传输完成中断使能 1：使能半程中断 0：关闭半程中断
0	MCHEN	Channel7 使能 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	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	-	MCHFLSAD[14:8]						
位权限	U-0	R/W-00000000						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MCHFLSAD[7:0]							
位权限	R/W-00000000							

位号	助记符	功能描述
31:15	--	RFU：未实现，读为 0
14:0	**MCHFLSAD**	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 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	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	-				MCHRAMAD[13:8]			
位权限	U-0				R/W-00000000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MCHRAMAD[7:0]							
位权限	R/W-00000000							

位号	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11:0	**MCHRAMAD**	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

位号	助记符	功能描述
		增或自减 软件可以查询当前 DMA 传输的目标 RAM 地址

27.7.7 DMA 状态标志寄存器（DMA_ISR）

名称	DMA_ISR							
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	-							DMA_ADDRERR
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DMACHFT[7:0]							
位权限	R/W1C-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DMACHHT[7:0]							
位权限	R/W1C-00000000							

位号	助记符	功能描述
31:17	--	RFU：未实现，读为 0
16	**DMA_ADDRERR**	DMA 传输地址错误标志
15:8	**DMACHFT[7:0]**	DMA 通道 x 传输完成标志，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	**DMACHHT[7:0]**	DMA 通道 x 传输半程标志，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，FM33LG0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
- CRC快速算法，1个时钟周期完成8bit CRC运算，4个时钟周期完成32bit CRC运算
- 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

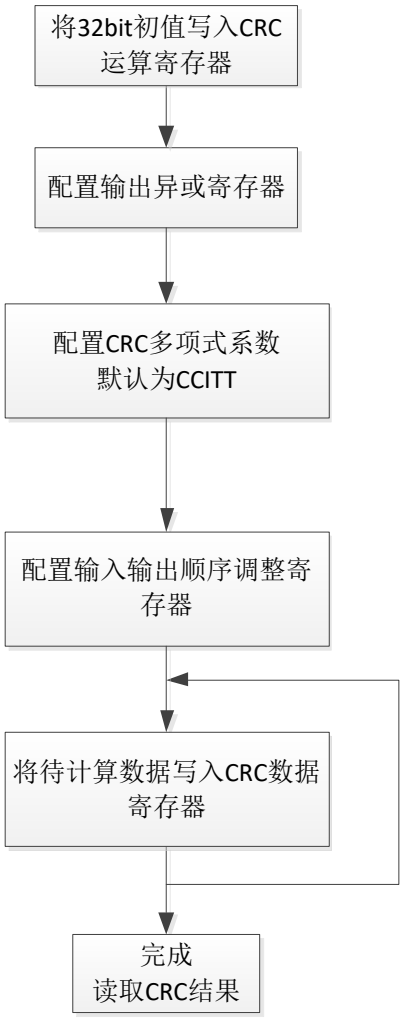


图 28-1CRC 运算流程图

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。

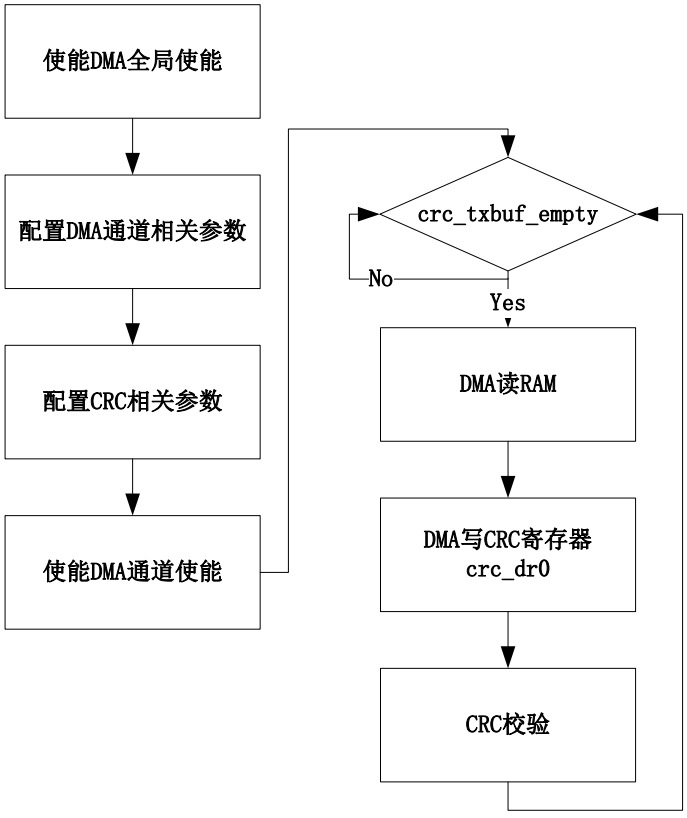


图28-2使用DMA对RAM中的数据进行CRC运算

28.5 Flash 数据完整性校验

通过使用DMA，可以实现CRC进行Flash内容的完整性校验，选中DMA的CRC通道，并将DMA地址指针指向Flash地址，配置DMA长度，启动DMA即可实现将Flash内容连续搬运到CRC模块进行CRC计算，DMA搬运结束后软件可以读取CRC结果，与预期值比对，以确认Flash内容是否正确、完整。

28.6 寄存器

offset 地址	名称	符号
CRC(模块基地址：0x40010000)		
0x00	CRC 数据寄存器 （CRC Data Register）	CRC_DR
0x04	CRC 控制状态寄存器 （CRC Control Register）	CRC_CR
0x08	CRC LFSR 寄存器 （CRC Linear Feedback Shift Register）	CRC_LFSR
0x0C	CRC 输出异或寄存器 （CRC output XOR Register）	CRC_XOR
0x1C	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	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CRCDR[31:24]							
位权限	R/W-11111111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CRCDR[23:16]							
位权限	R/W-11111111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CRCDR[15:8]							
位权限	R/W-11111111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CRCDR[7:0]							
位权限	R/W-11111111							

位号	助记符	功能描述
31:0	**CRCDR**	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。 作为输入时：若为 Flash CRC 计算或 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	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	-						OPWD	PARA
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	CRCSEL	
位权限	R/W-00		R/W-0	R-0	R-0	R/W-0	R/W-10	

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 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	**PARA**	CRC 快速计算使能 0：串行运算，计算 1 个字节需要 8 个时钟周期
7:6	**RFLTIN**	CRC 输入反转控制 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 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 0：CRC 结果为 0 1：CRC 结果非全 0
3	**BUSY**	CRC 运算标志位，只读 0：CRC 运算结束 1：CRC 运算进行中
2	**XOR**	输出异或使能 0：输出不异或 CRC_XOR 寄存器 1：输出异或 CRC_XOR 寄存器
1:0	**CRCSEL**	CRC 多项式位宽选择 00：32 位 01：16 位 10：8 位 11：7 位

28.6.3 CRC LFSR 寄存器（CRC_LFSR）

名称	CRC_LFSR							
Offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LFSR[31:24]							
位权限	R/W-11111111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LFSR[23:16]							
位权限	R/W-11111111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LFSR[15:8]							
位权限	R/W-11111111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LFSR[7:0]							
位权限	R/W-11111111							

位号	助记符	功能描述
31:0	**LFSR**	CRC LFSR 寄存器，运算开始前可以由软件写入初始值

28.6.4 CRC 输出异或寄存器（CRC_XOR）

名称	CRC_XOR							
Offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CRC_XOR[31:24]							
位权限	R/W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CRC_XOR[23:16]							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CRC_XOR[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CRC_XOR[7:0]							
位权限	R/W-00000000							

位号	助记符	功能描述
31:0	**CRC_XOR**	CRC 运算结果异或寄存器

28.6.5 CRC 多项式寄存器（CRC_POLY）

名称	CRC_POLY							
Offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CRCPOLY[31:24]							

位权限	R/W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CRCPOLY[23:16]							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CRCPOLY[15:8]							
位权限	R/W-00010000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CRCPOLY[7:0]							
位权限	R/W-00100001							

位号	助记符	功能描述
31:0	**CRCPOLY**	CRC 运算多项式系数

29 高级定时器（ATIM）

29.1 概述

FM33LG0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

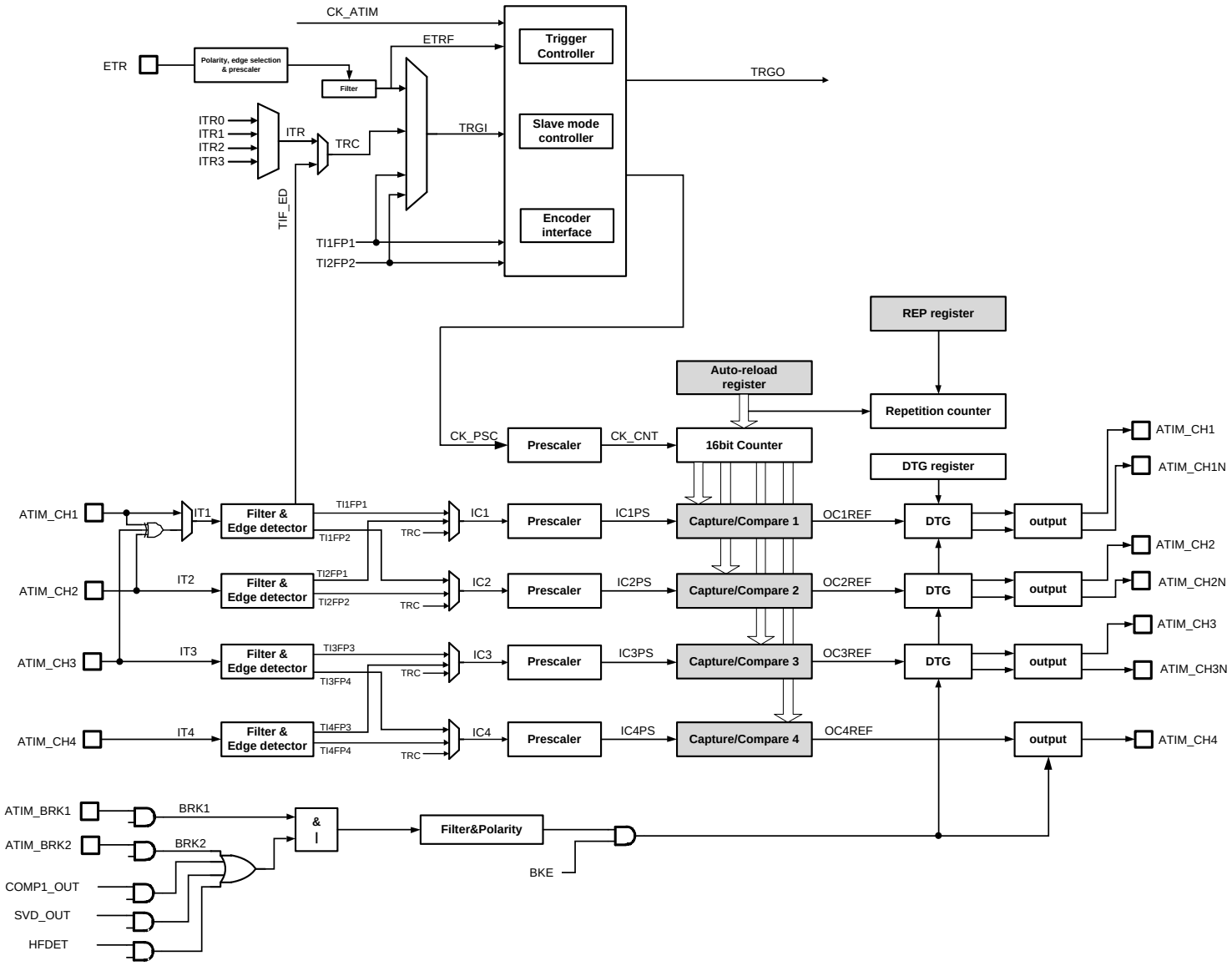


图 29-1 高级定时器结构框图

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。

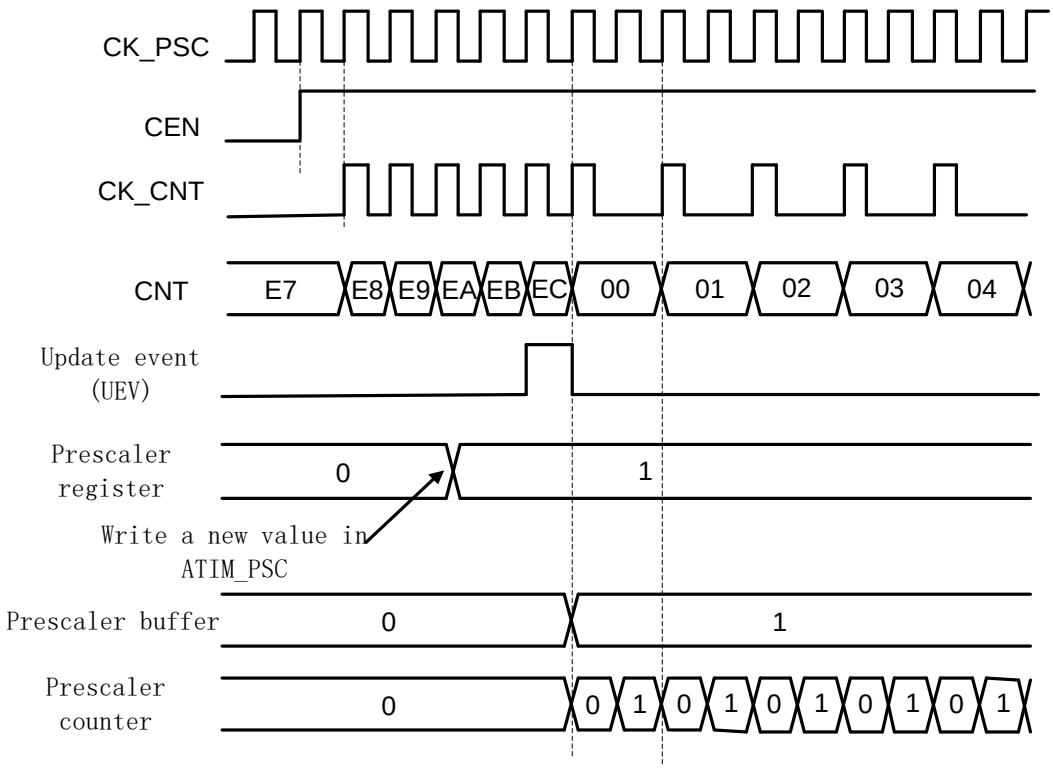


图 29-2 预分频从 1 变为 2 的波形

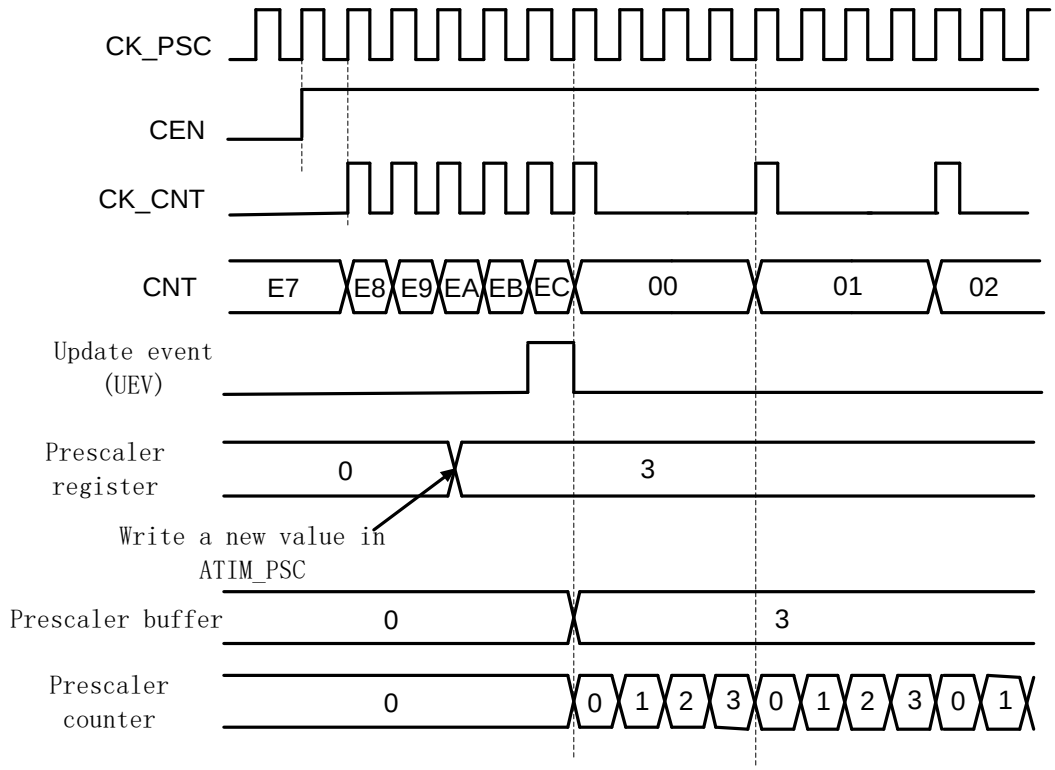


图 29-3 预分频从 1 变为 4 的波形

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

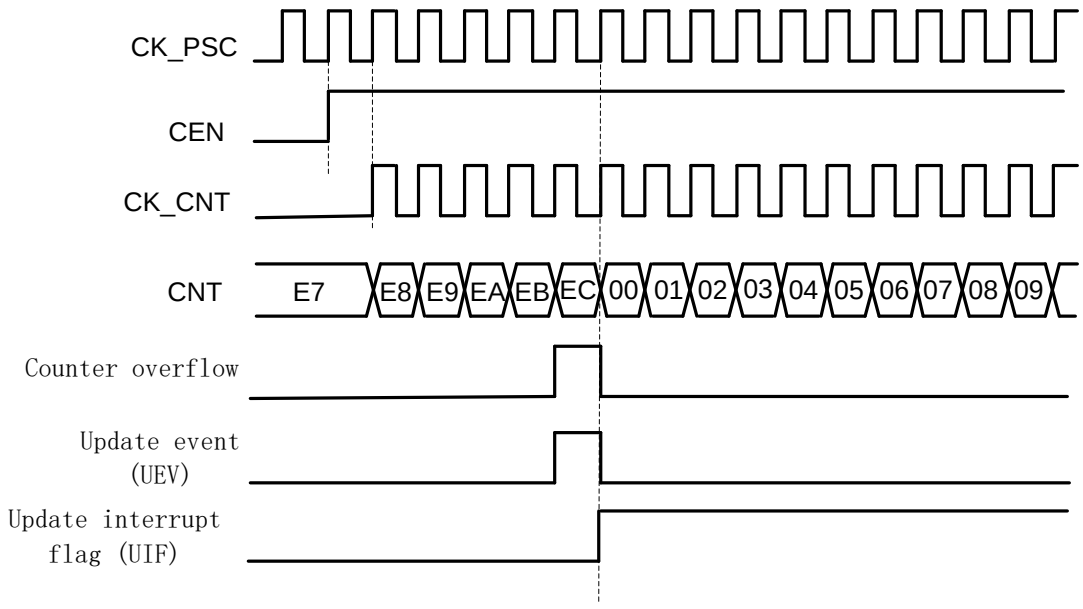


图 29-4 向上计数波形，内部时钟不分频

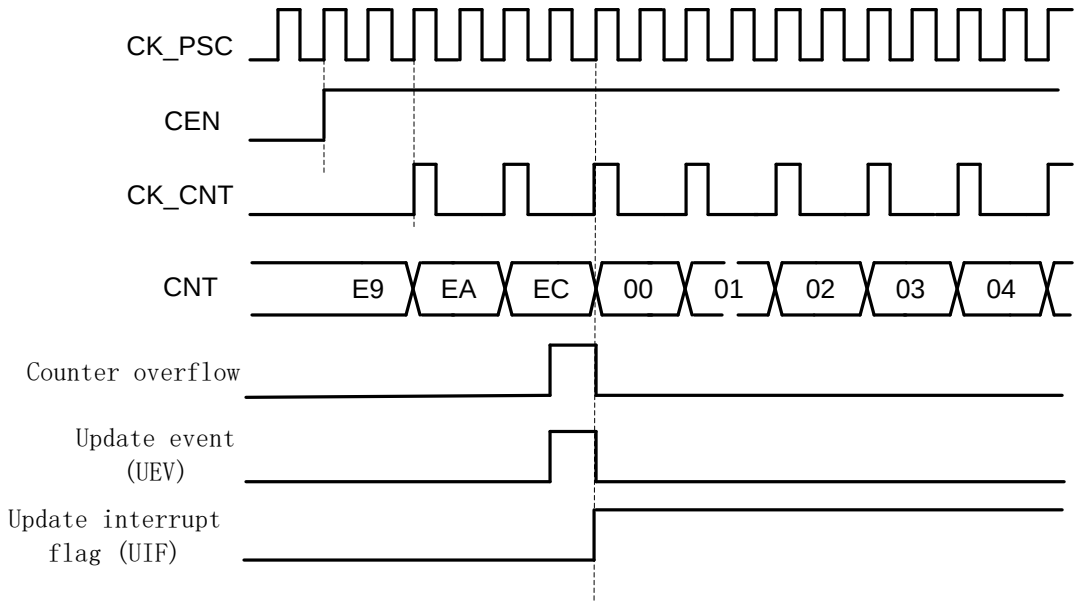


图 29-5 向上计数波形，内部时钟 2 分频

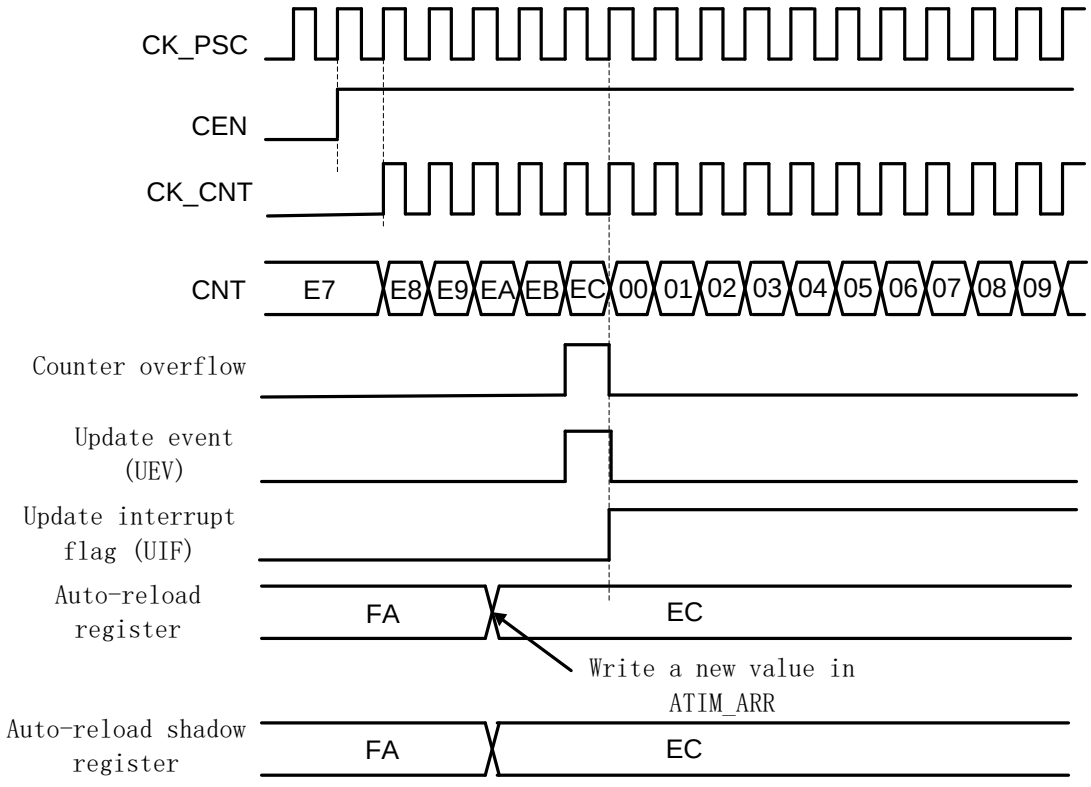


图 29-6ARPE=0（ATIM_ARR 没有预装载）时的更新事件

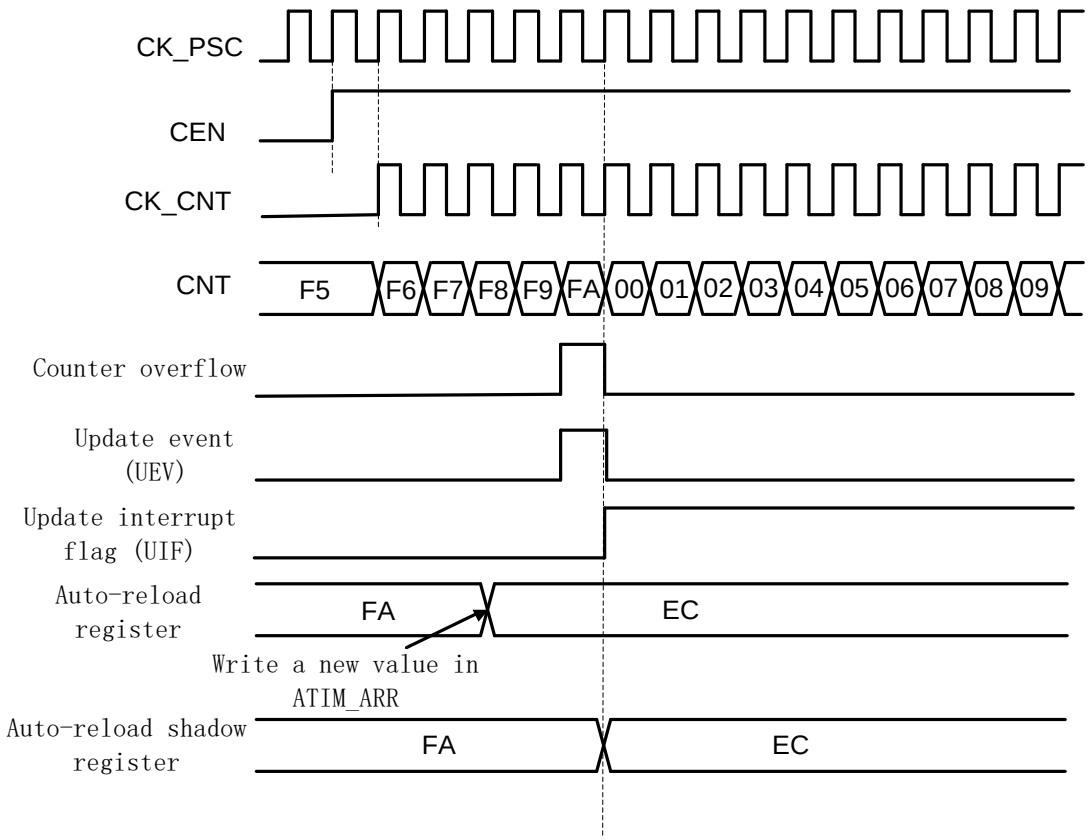


图 29-7ARPE=1（ATIM_ARR 预装载）时的更新事件

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

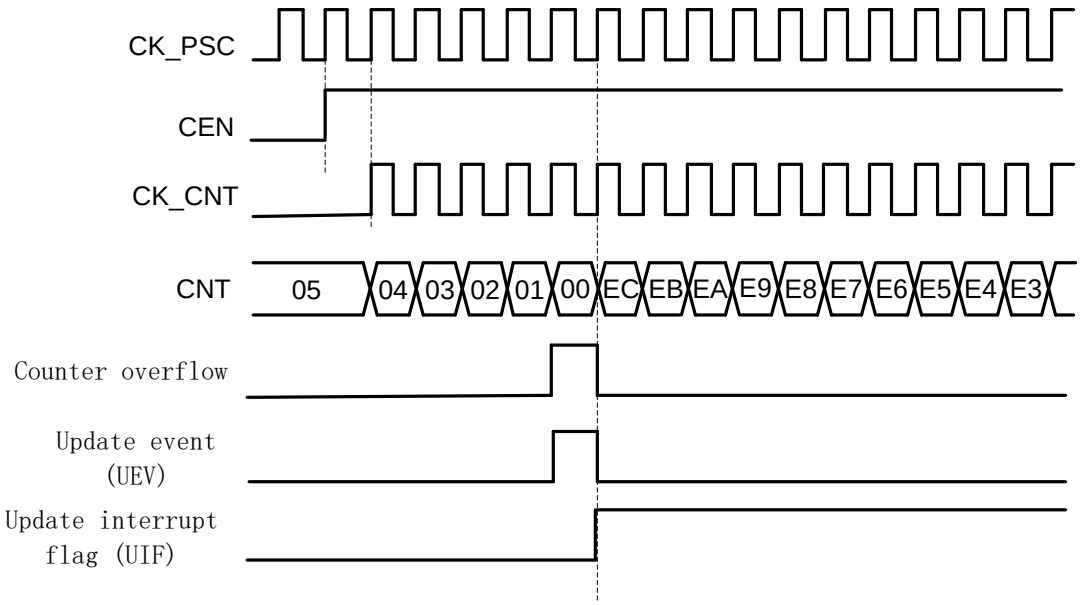


图 29-8 向下计数，内部时钟不分频

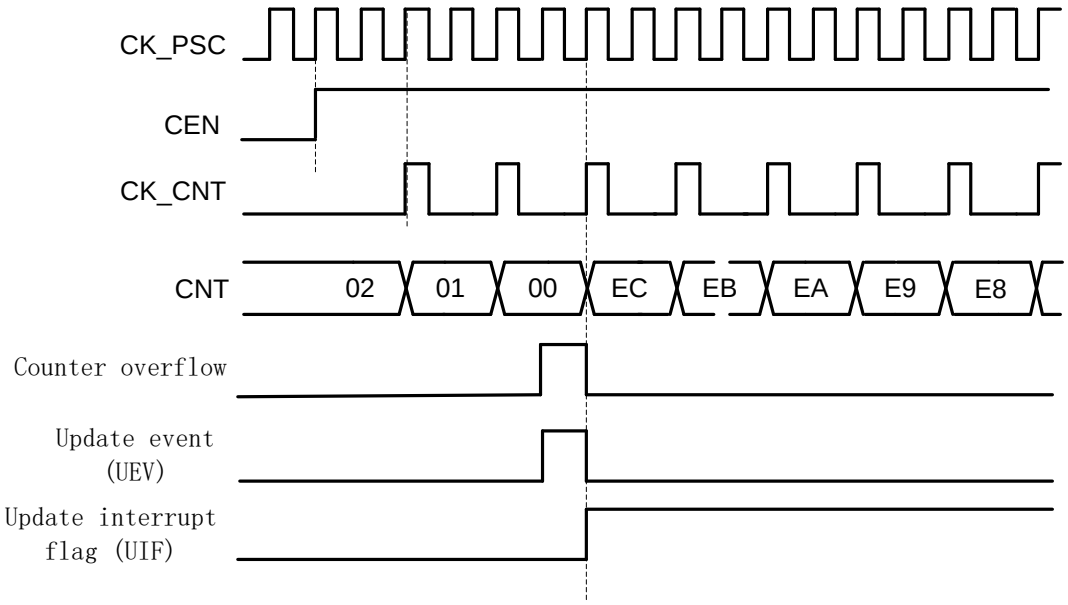


图 29-9 向下计数，内部时钟 2 分频

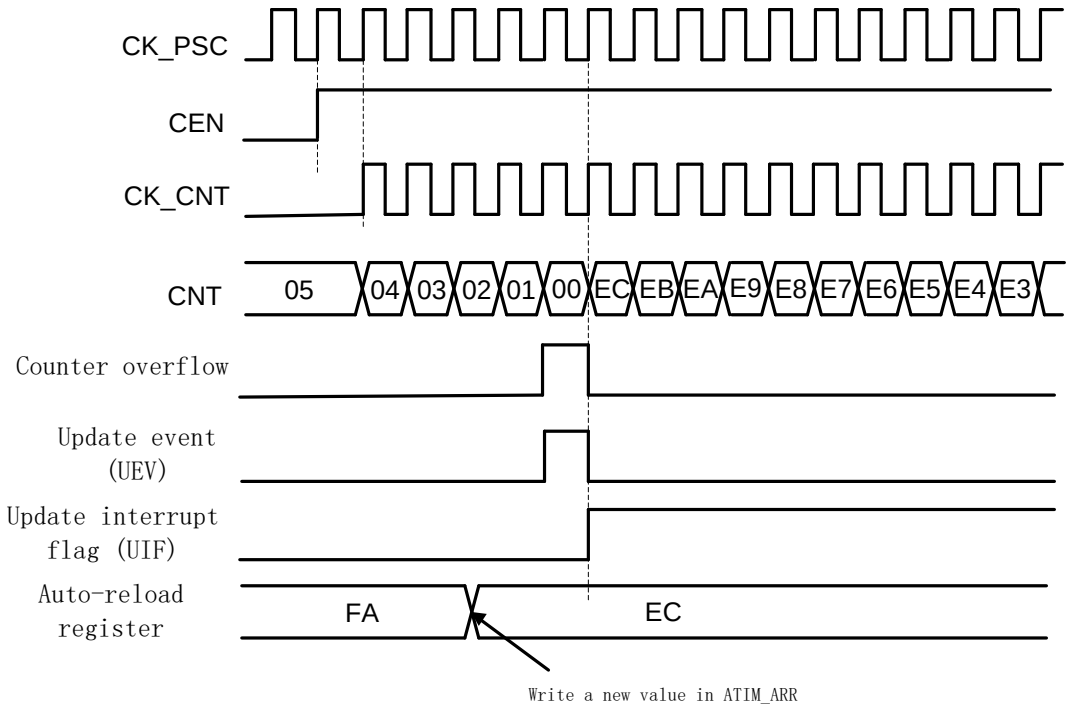


图 29-10 向下计数，内部时钟 2 分频

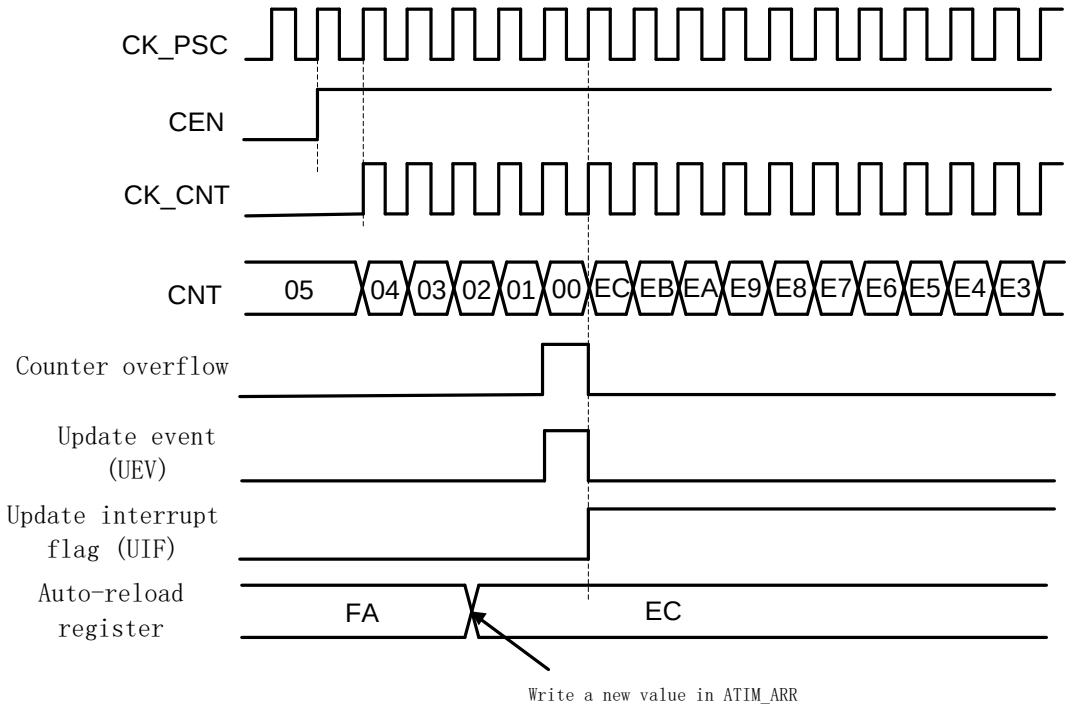


图 29-11 向下计数，不使用重复计数时的更新事件

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。

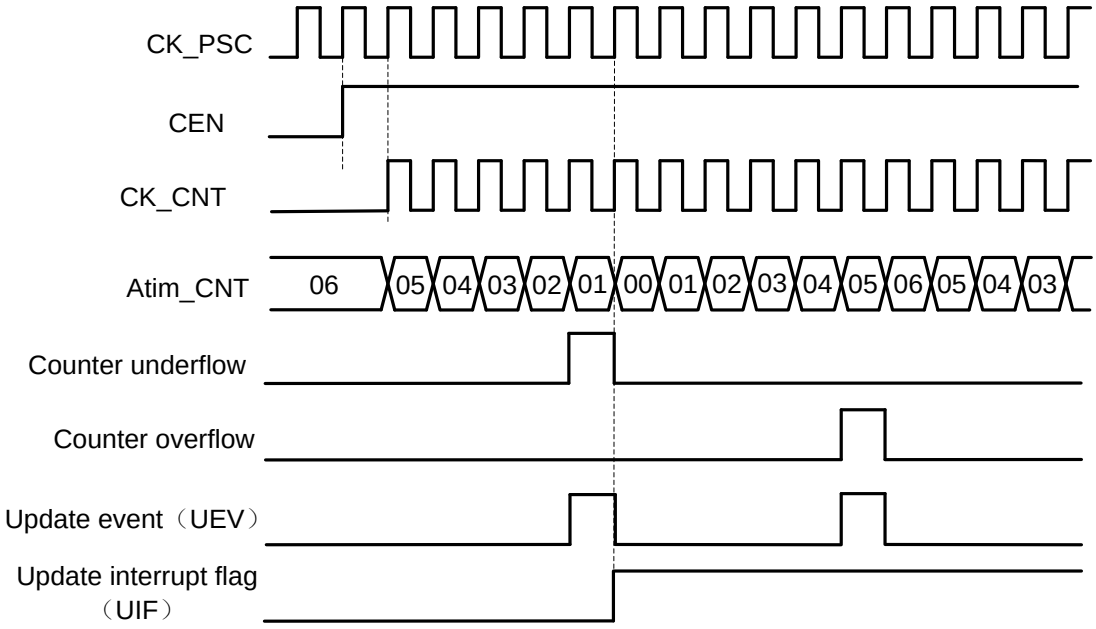


图 29-12 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6

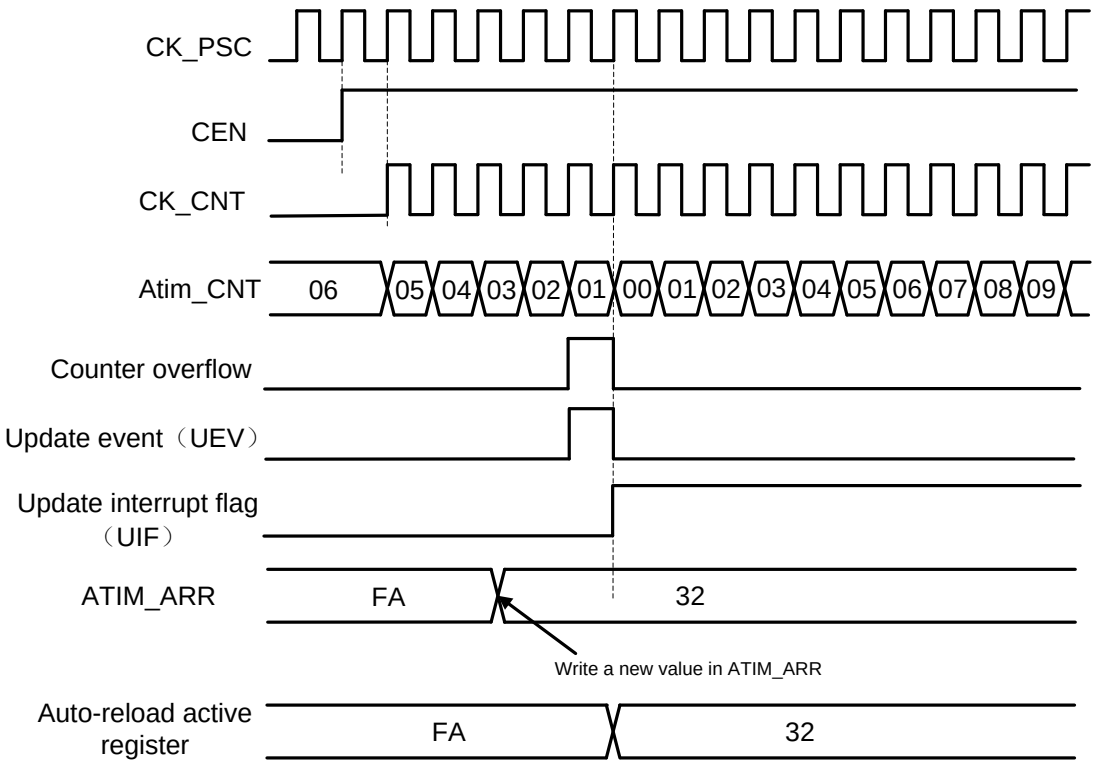


图 29-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

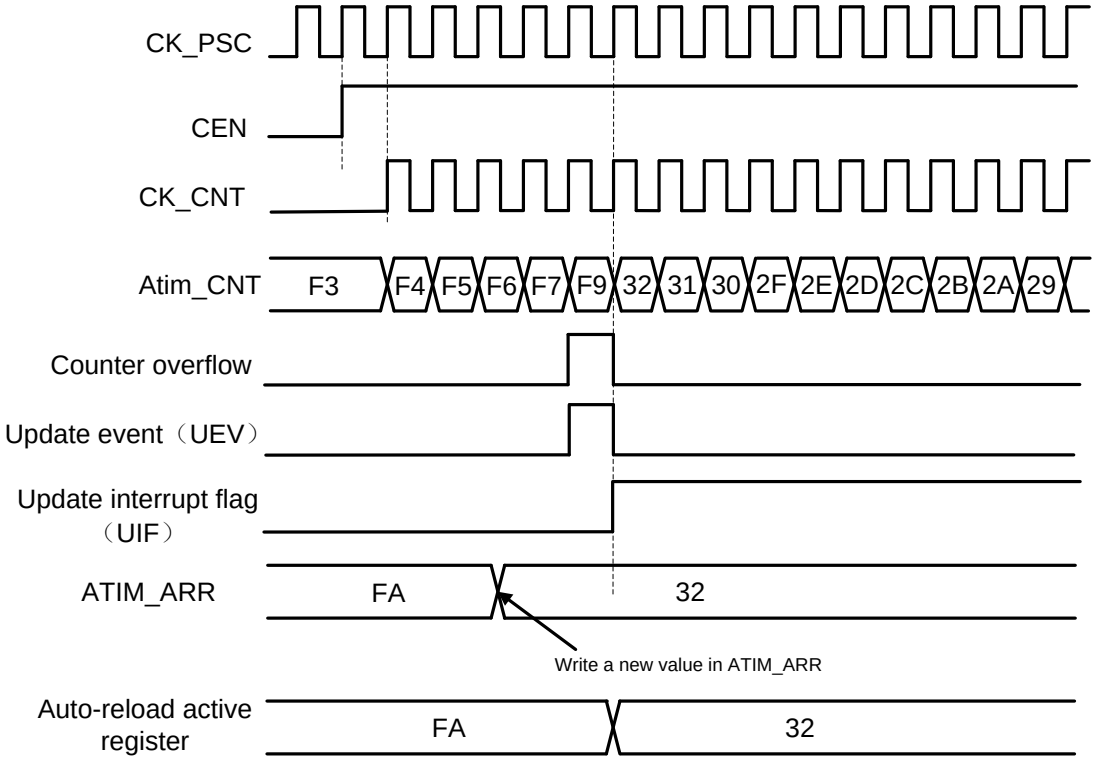


图 29-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

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。

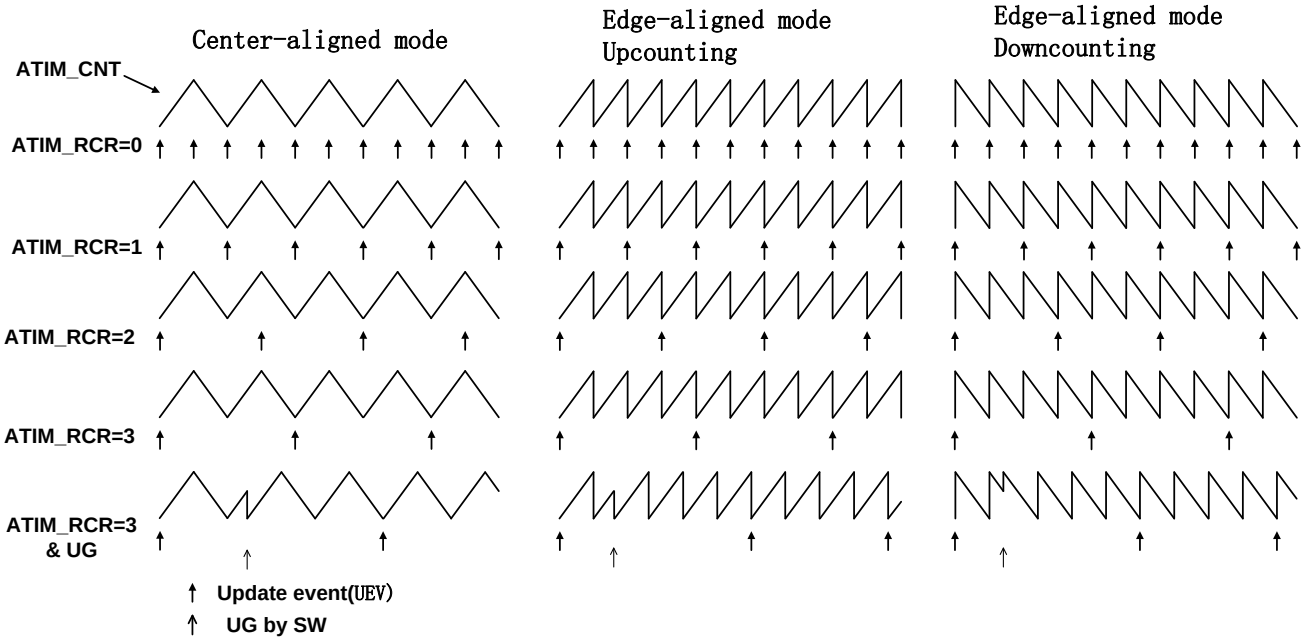


图 29-15 不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置

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

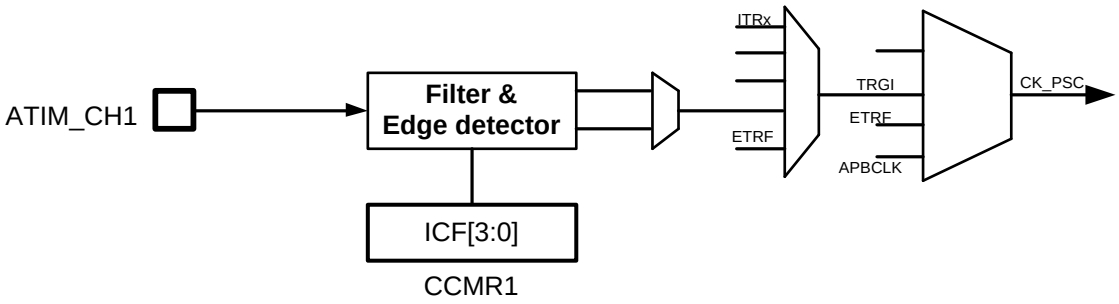


图 29-16 ATIM 时钟源框图

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。

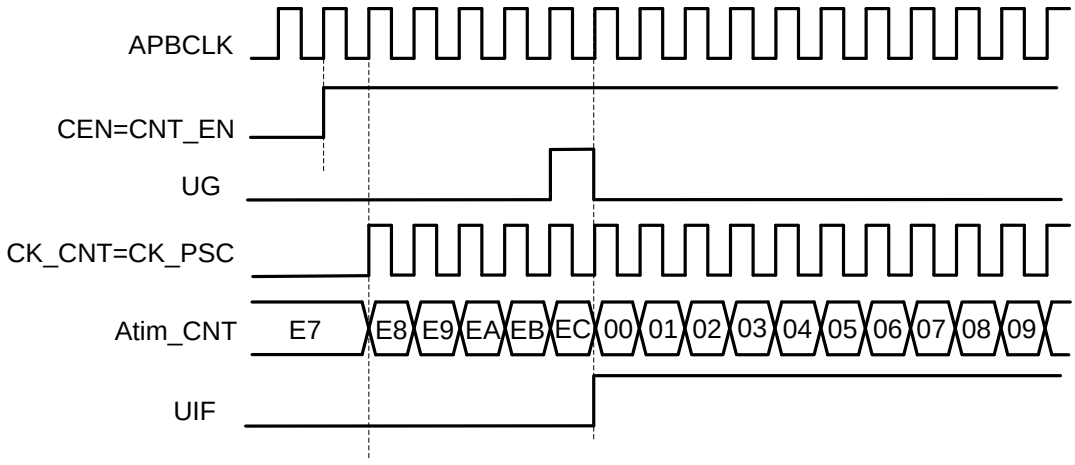


图 29-17 内部时钟源模式，时钟分频因子为 1

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。

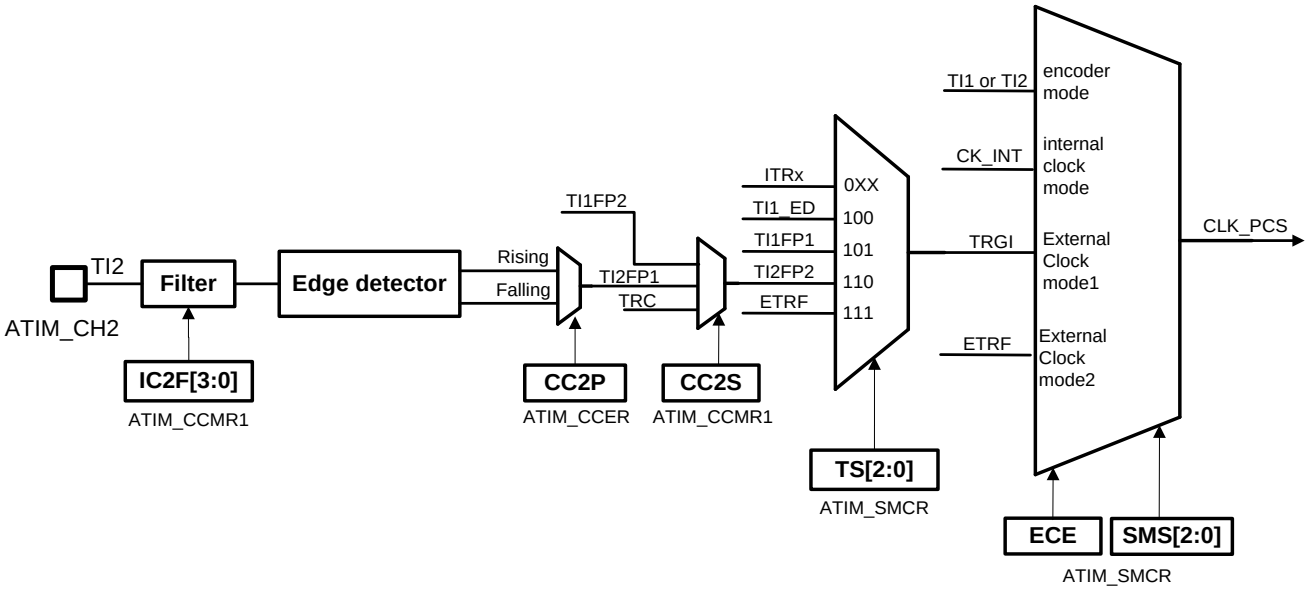


图 29-18TI2 外部时钟连接例子

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

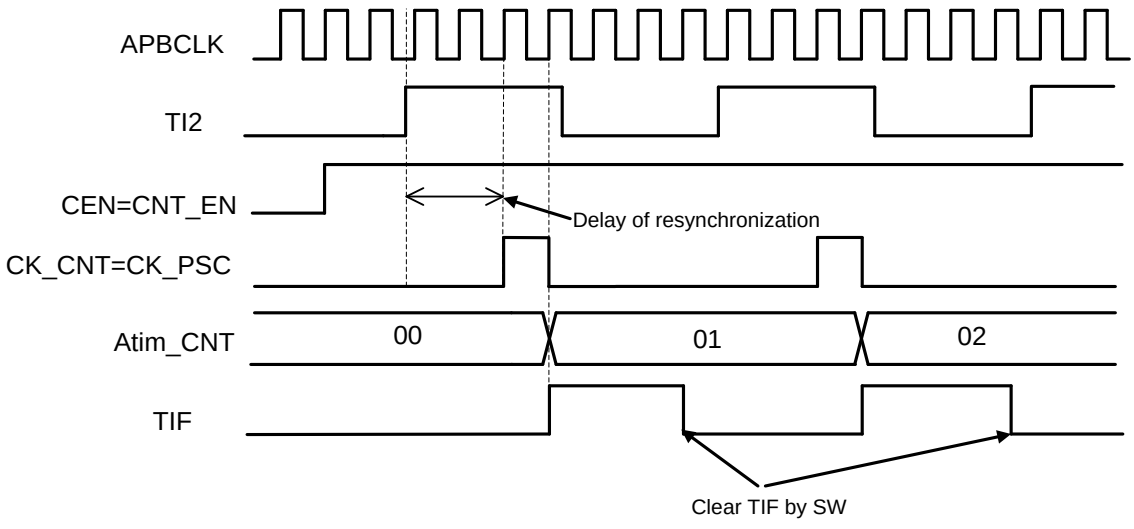


图 29-19 外部时钟模式 1 下的时序

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：

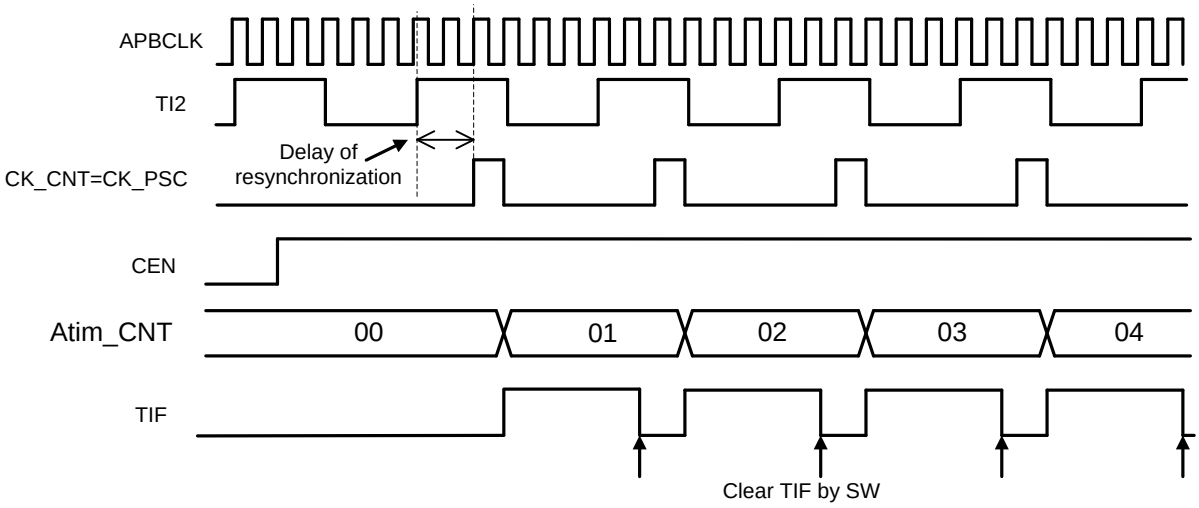


图 29-20 外部时钟模式 1 下的时序

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。

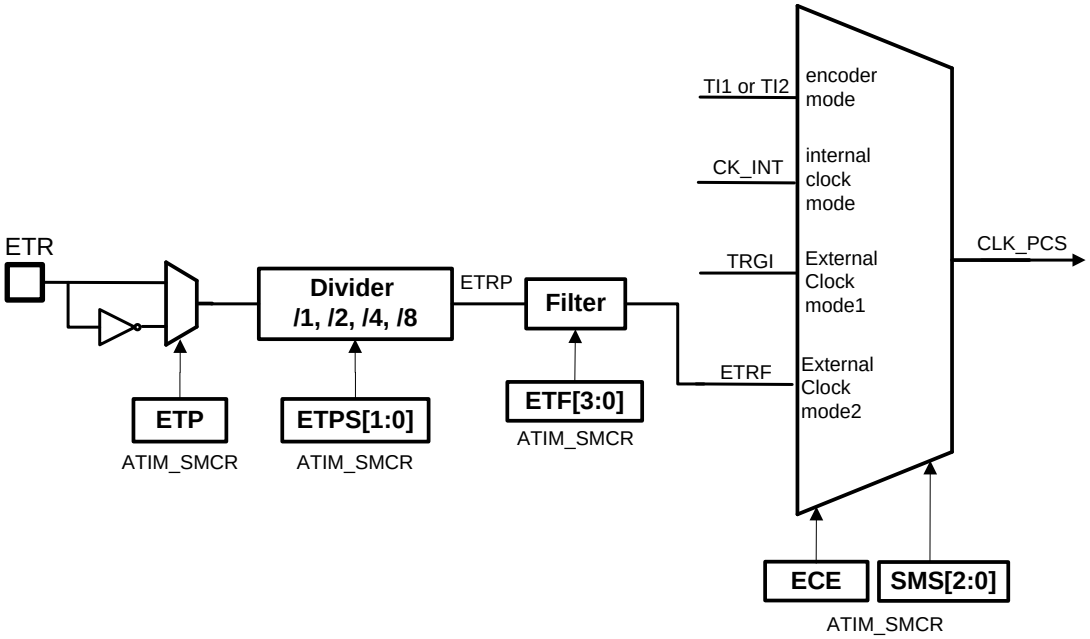


图 29-21 外部触发输入框图

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。

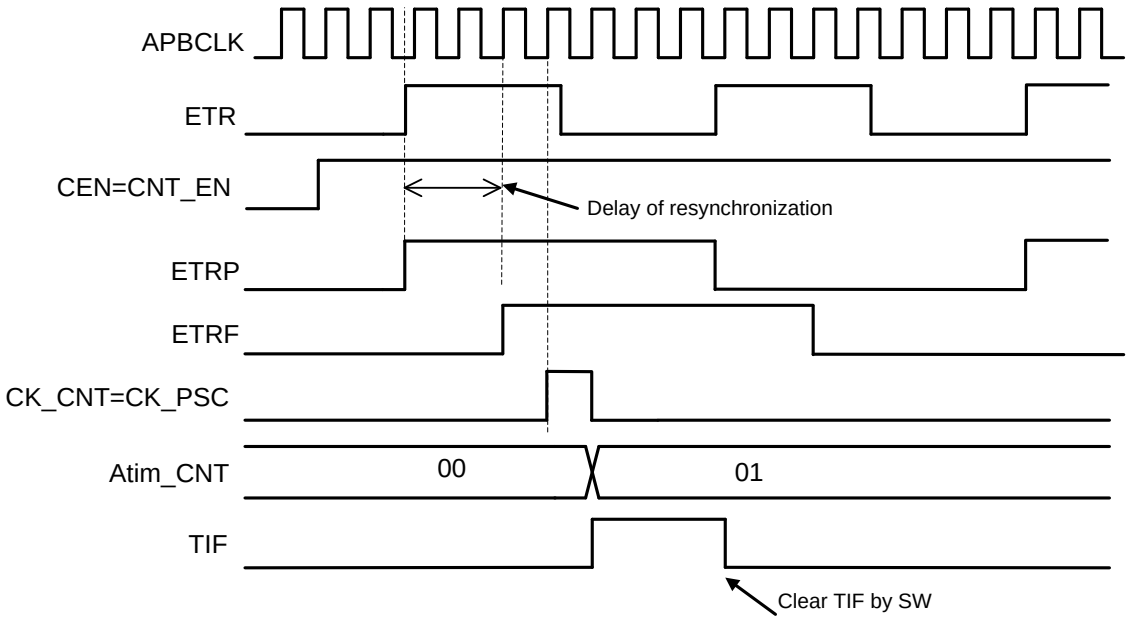


图 29-22 外部时钟模式 2 下的时序 1

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：

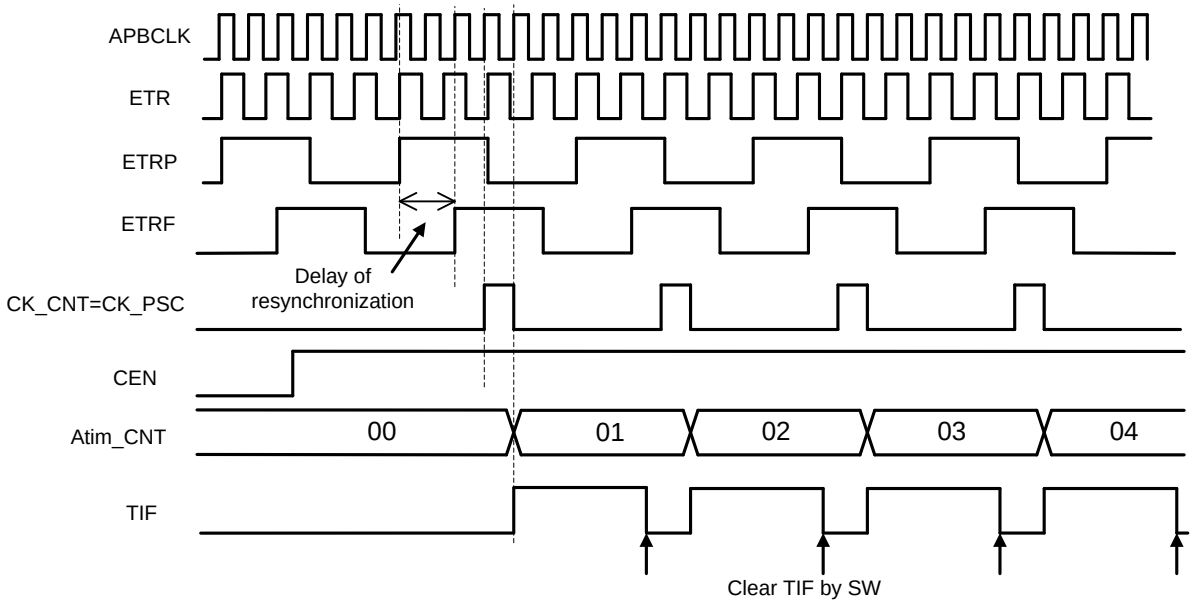


图29-23外部时钟模式2下的时序2

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	GPTIM1_TRGO	GPTIM2_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。

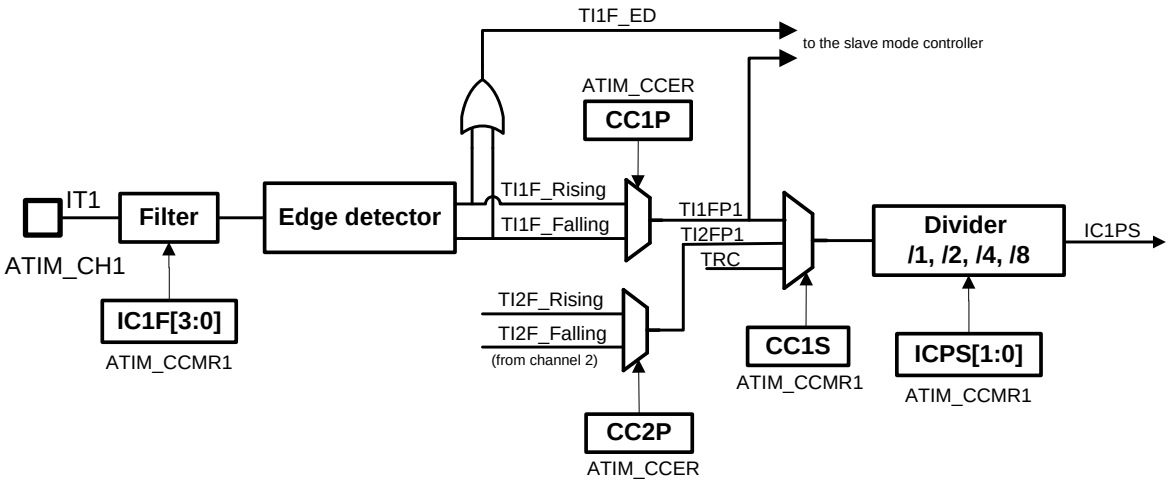


图29-24捕获/比较通道(通道1输入部分)

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。

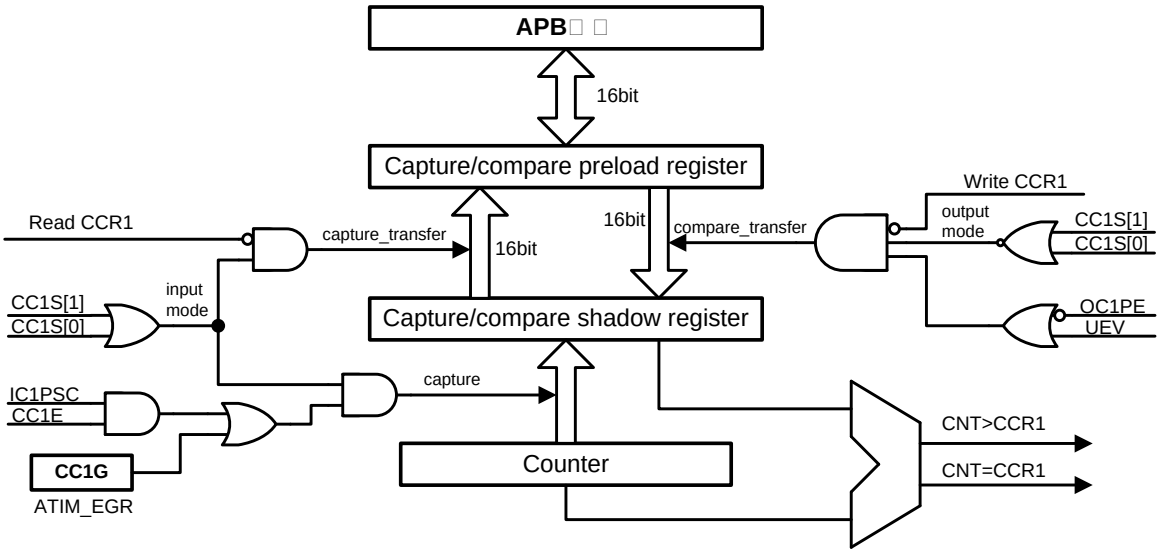


图29-25捕获/比较通道1的主电路

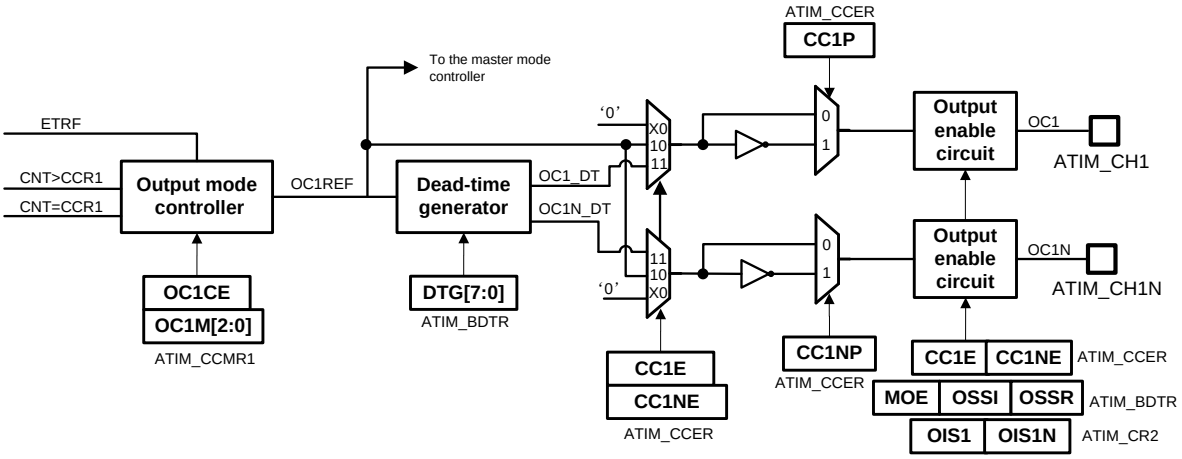


图29-26捕获/比较通道的输出部分(通道1至3)

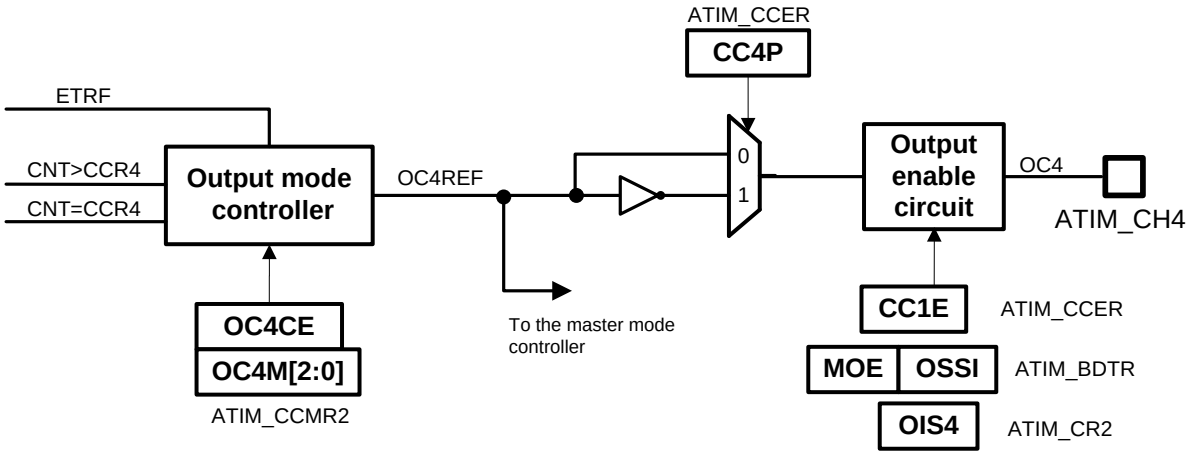


图29-27捕获/比较通道的输出部分(通道4)

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

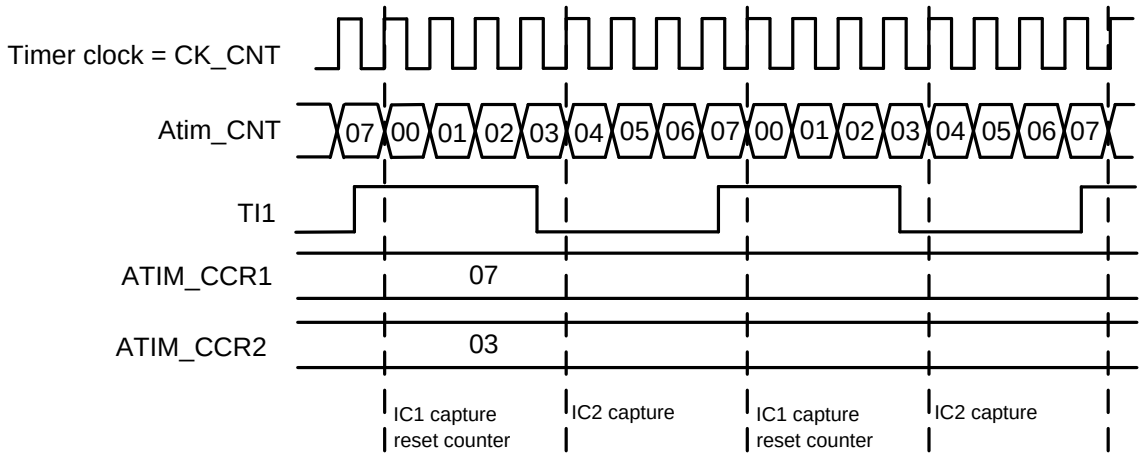


图29-28PWM输入捕获模式时序

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

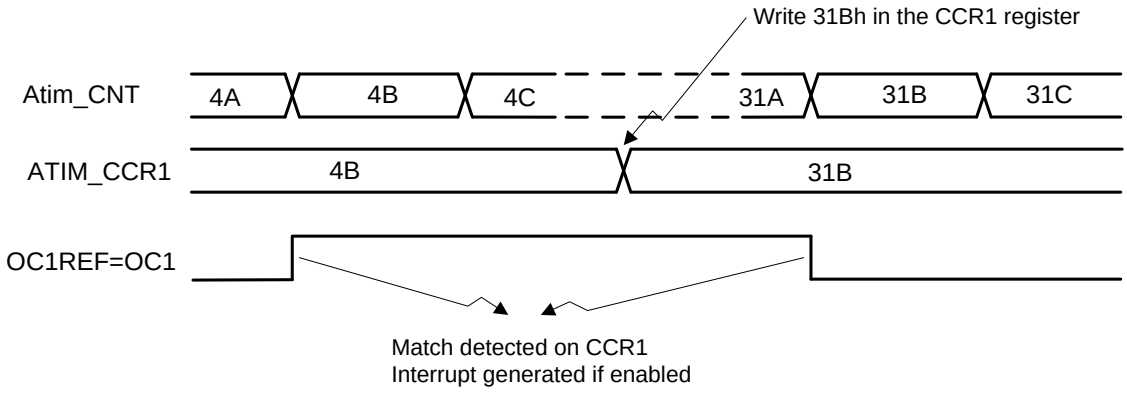


图29-29输出比较模式，翻转OC1

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.

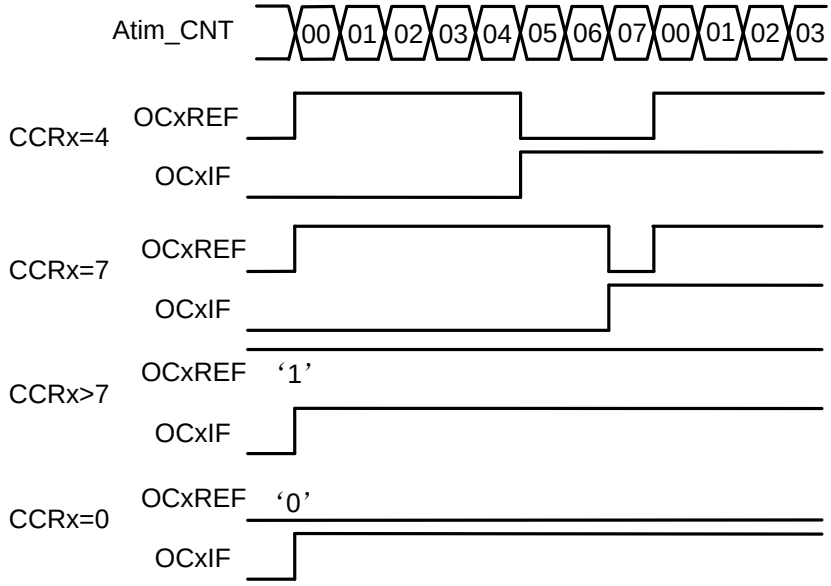


图29-30边沿对齐的PWM波形(ARR=7)

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：

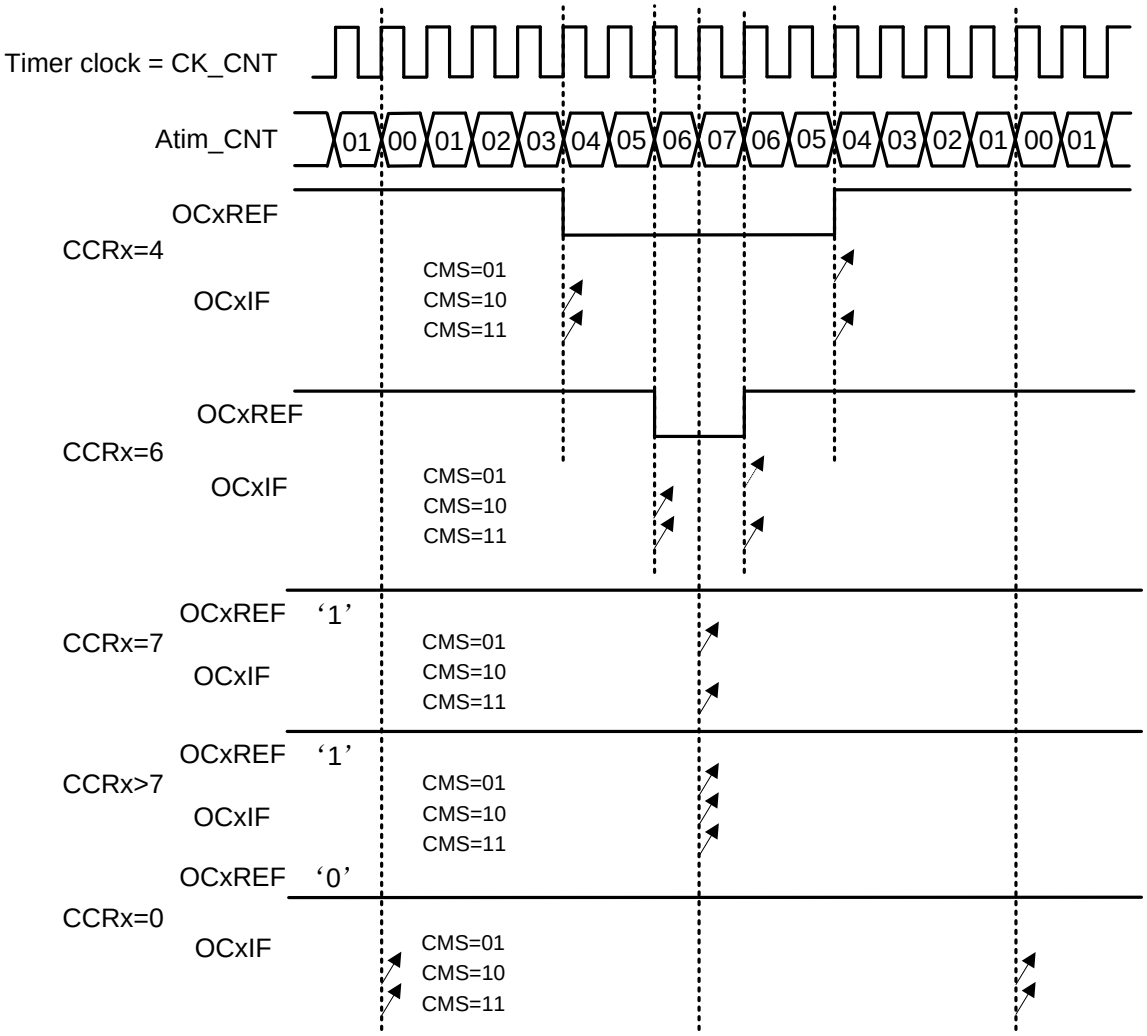


图29-31中央对齐的PWM波形(APR=7)

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。

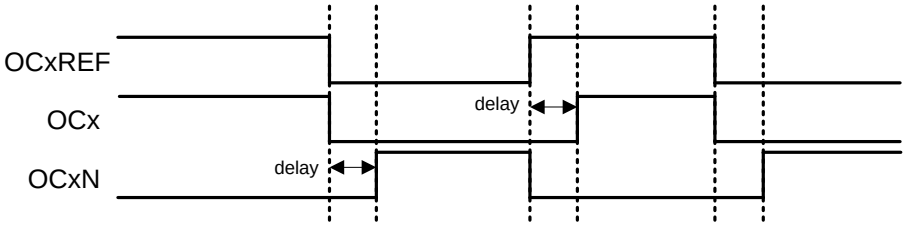


图29-32带死区插入的互补输出

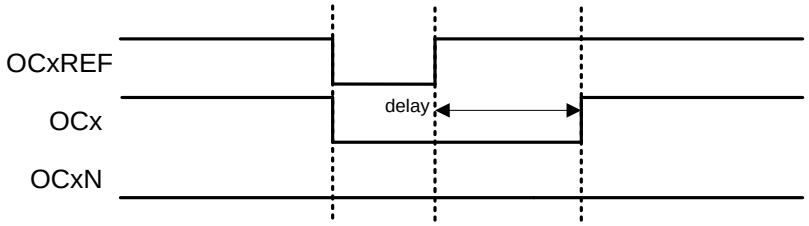


图29-33死区波形延迟大于负脉冲

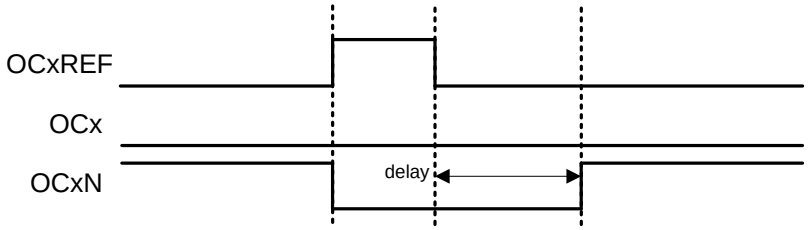
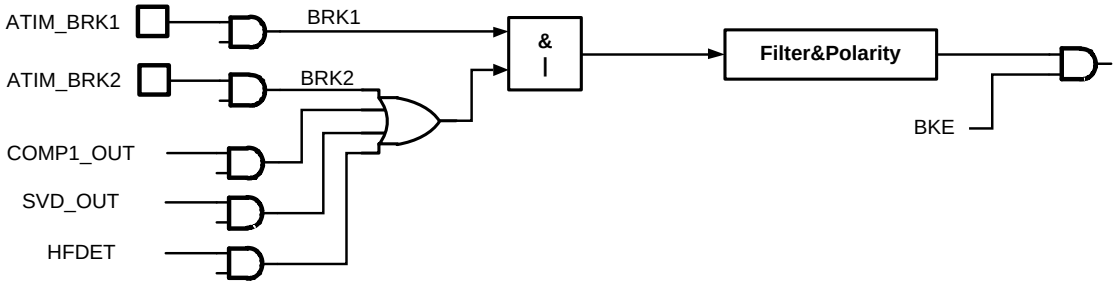


图29-34死区波形延迟大于正脉冲

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。

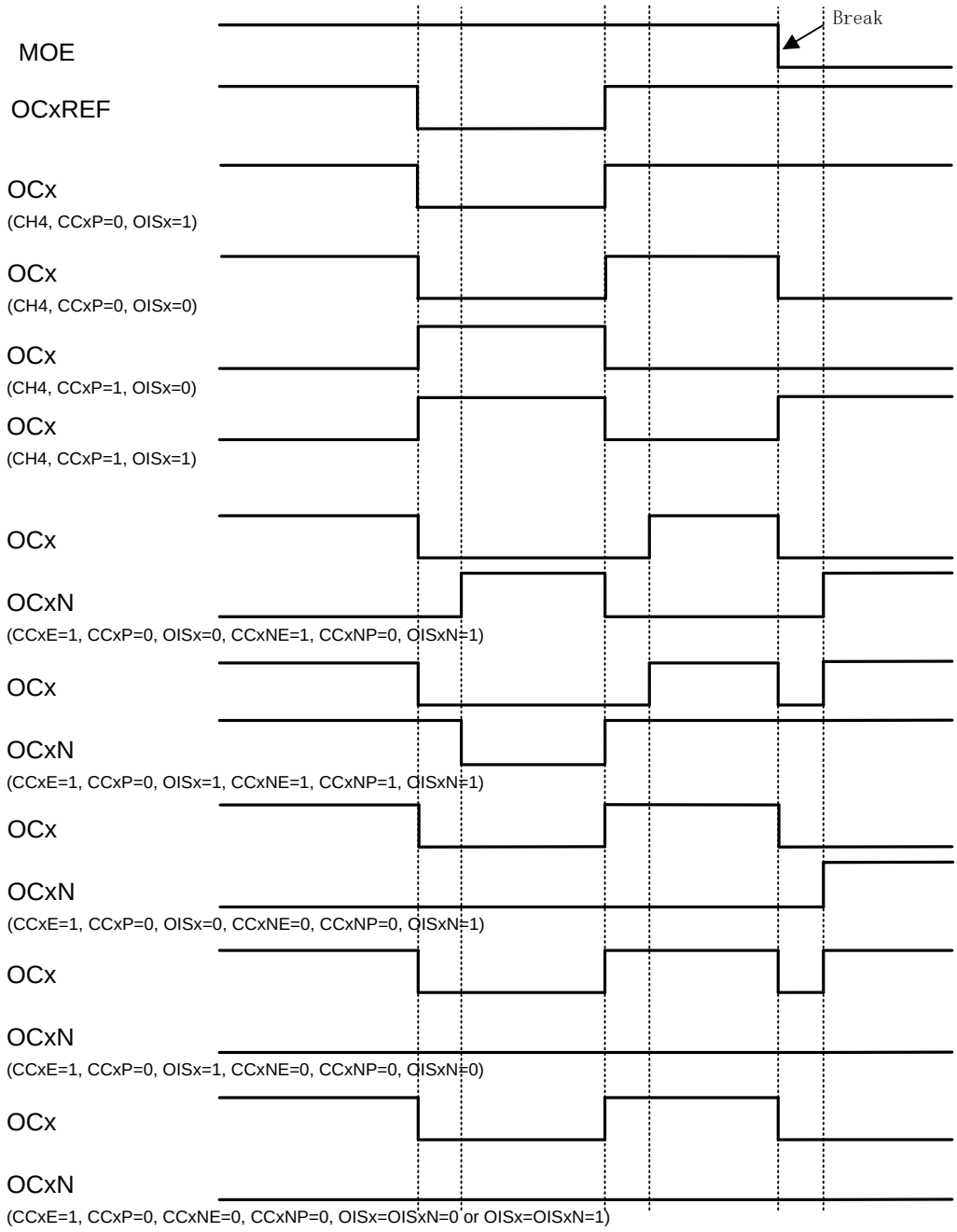


图29-35响应刹车的输出

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。

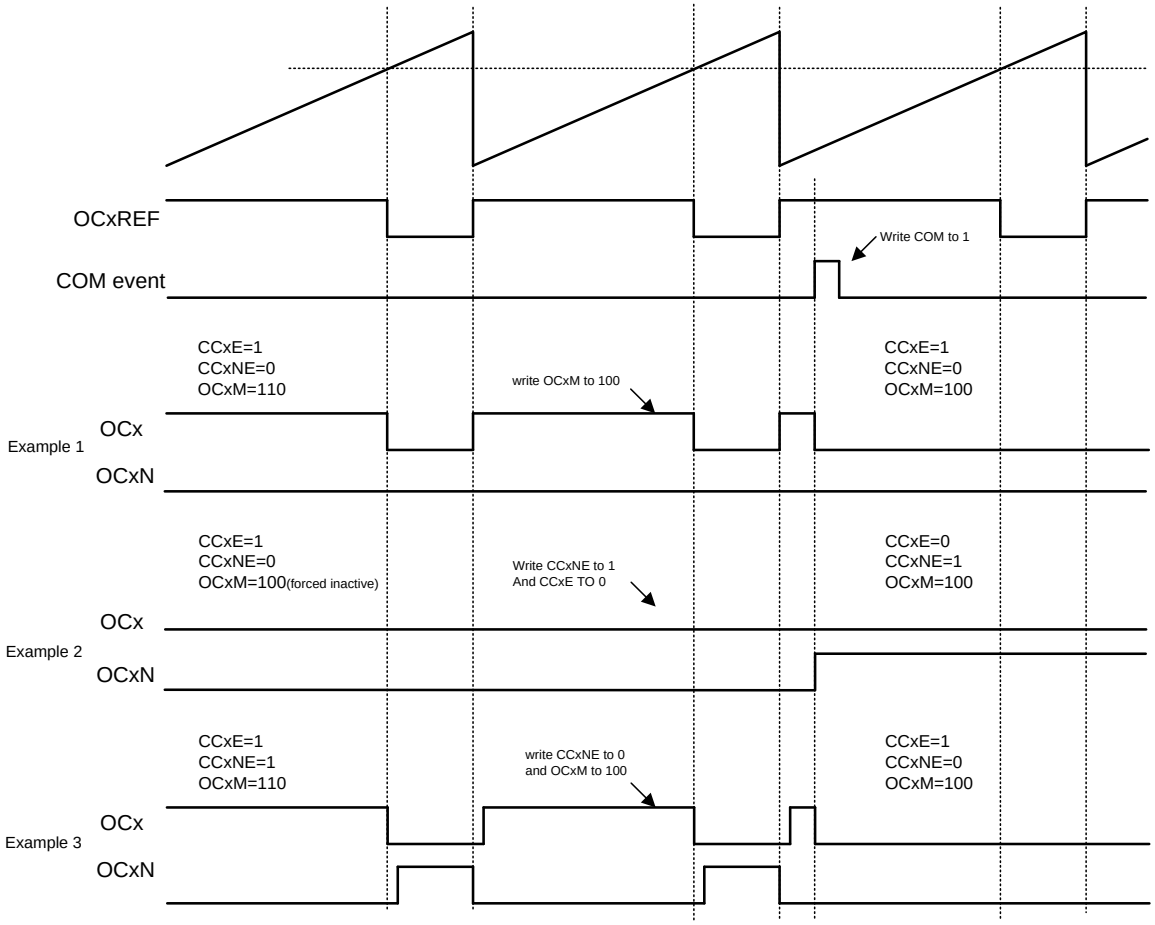


图29-36产生六步PWM，使用COM的例子(OSSR=1)

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

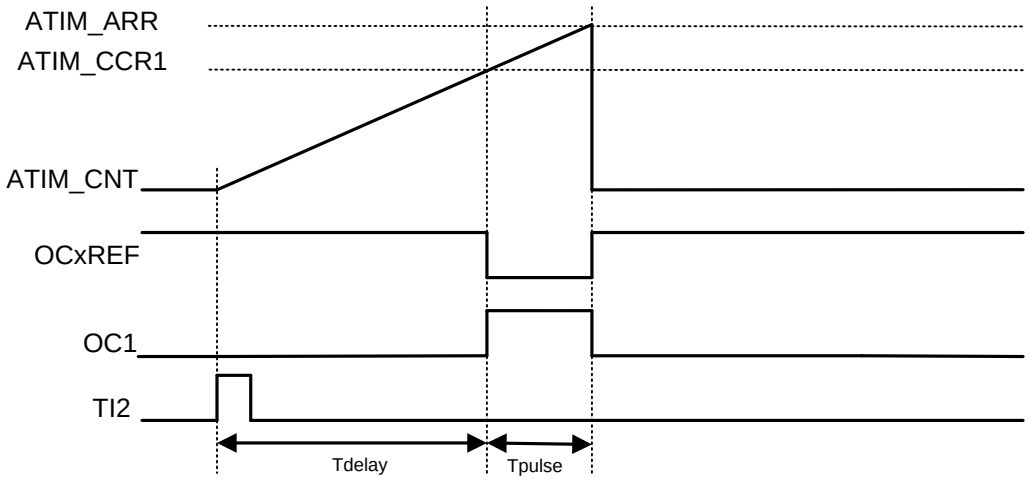


图29-37单脉冲模式的例子

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。

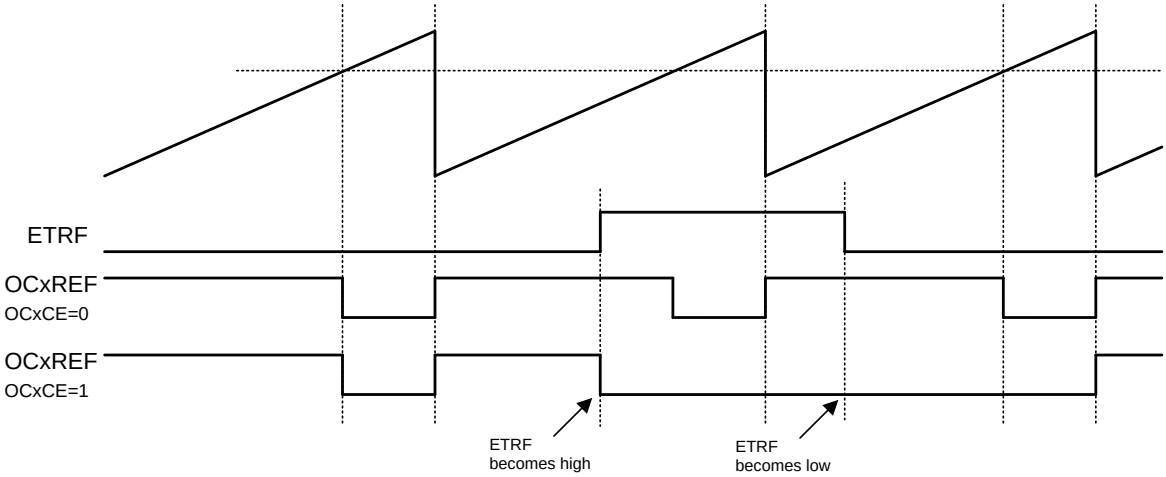


图29-38 ETR信号清除ATIM的OCxREF

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。

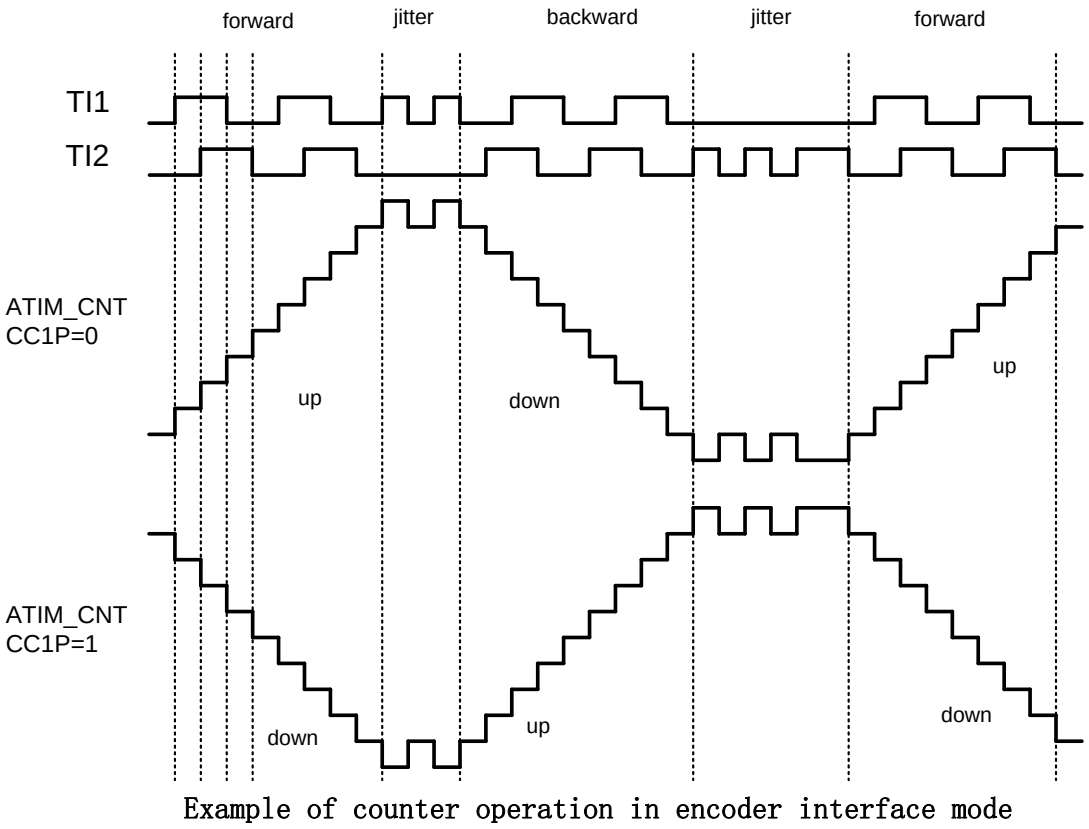


图29-39编码器模式下的计数器操作实例

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

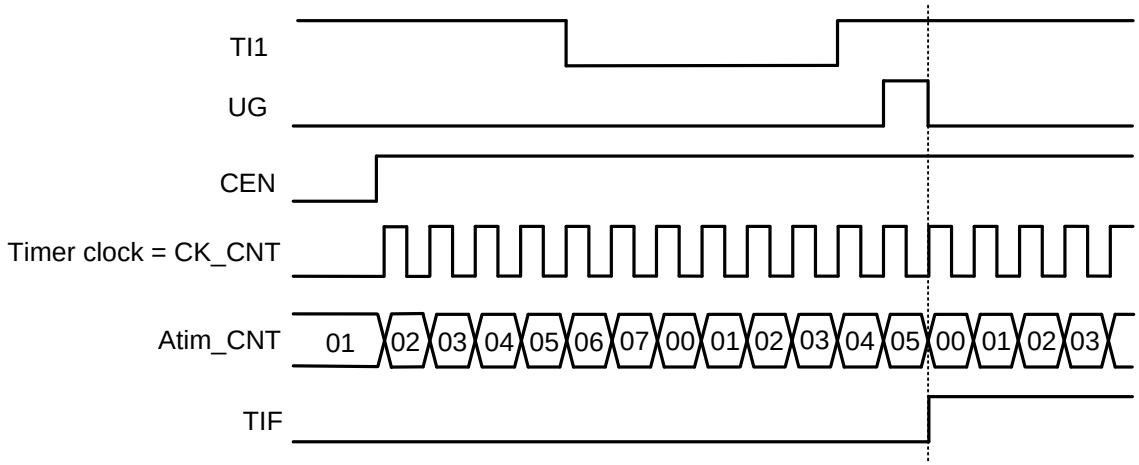


图29-40复位模式下的时序

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

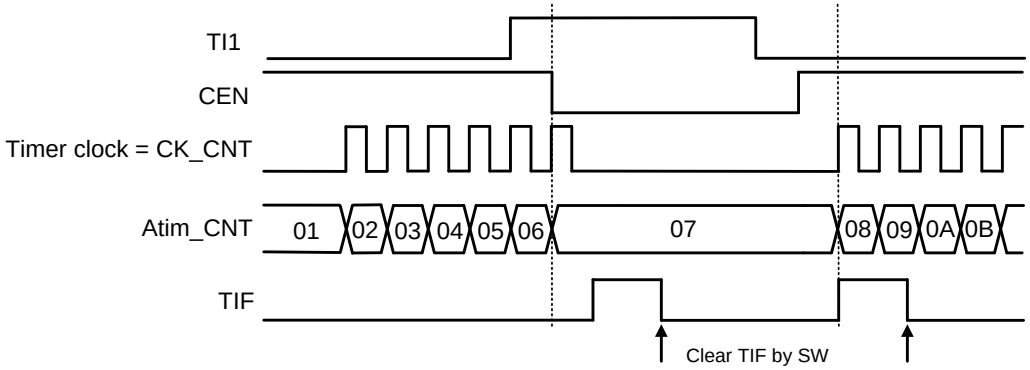


图29-41门控模式下的时序

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

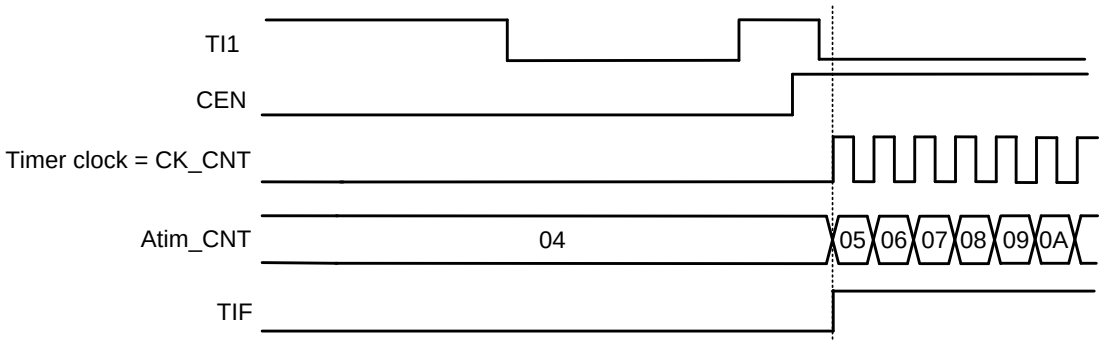


图29-42触发器模式下的时序

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

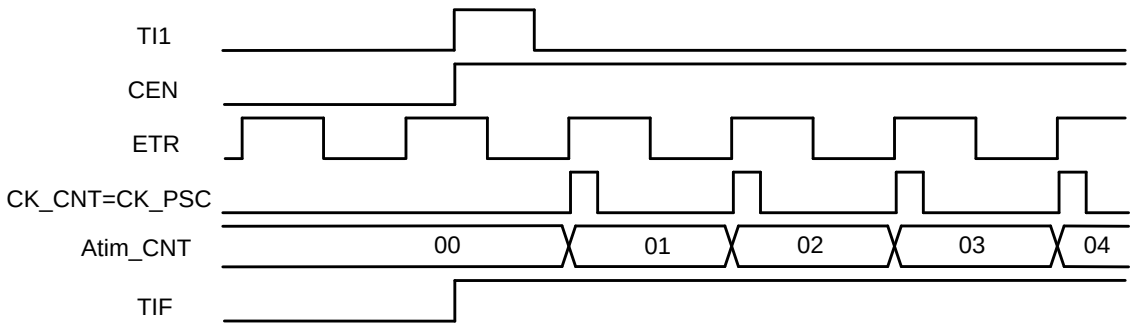


图29-43外部时钟模式2＋触发模式下的时序

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

offset 地址	名称	符号
ATIM(模块基地址：0x40013000)		
0x00000000	ATIM 控制寄存器 1 （ATIM Control Register1）	ATIM_CR1
0x00000004	ATIM 控制寄存器 2 （ATIM Control Register2）	ATIM_CR2
0x00000008	ATIM 从机模式控制寄存器 （ATIM Slave Mode Control Register）	ATIM_SMCR
0x0000000C	ATIM DMA 和中断使能寄存器 （ATIM DMA and Interrupt Enable Register）	ATIM_DIER
0x00000010	ATIM 状态寄存器 （ATIM Interrupt Status Register）	ATIM_ISR
0x00000014	ATIM 事件产生寄存器 （ATIM Event Generation Register）	ATIM_EGR
0x00000018	ATIM 捕捉/比较模式寄存器 1 （ATIM Capture/Compare Mode Register1）	ATIM_CCMR1
0x0000001C	ATIM 捕捉/比较模式寄存器 2 （ATIM Capture/Compare Mode Register2）	ATIM_CCMR2
0x00000020	ATIM 捕捉/比较使能寄存器 （ATIM Capture/Compare Enable Register）	ATIM_CCER
0x00000024	ATIM 计数器寄存器 （ATIM Counter Register）	ATIM_CNT
0x00000028	ATIM 预分频寄存器 （ATIM Prescaler Register）	ATIM_PSC
0x0000002C	ATIM 自动重载寄存器 （ATIM Auto-Reload Register）	ATIM_ARR
0x00000030	ATIM 重复计数寄存器 （ATIM Repetition Counter Register）	ATIM_RCR
0x00000034	ATIM 捕捉/比较寄存器 1 （ATIM Capture/Compare Register1）	ATIM_CCR1
0x00000038	ATIM 捕捉/比较寄存器 2 （ATIM Capture/Compare Register2）	ATIM_CCR2
0x0000003C	ATIM 捕捉/比较寄存器 3 （ATIM Capture/Compare Register3）	ATIM_CCR3
0x00000040	ATIM 捕捉/比较寄存器 4 （ATIM Capture/Compare Register4）	ATIM_CCR4
0x00000044	ATIM 刹车和死区控制寄存器 （ATIM Break and Deadtime Register）	ATIM_BDTR
0x00000048	ATIM DMA 控制寄存器 （ATIM DMA Control Register）	ATIM_DCR
0x0000004C	ATIM DMA 访问寄存器 （ATIM DMA Access Register）	ATIM_DMAR
0x00000060	ATIM 刹车输入控制寄存器 （ATIM Break Control Register）	ATIM_BKCR

29.5.1 ATIM 控制寄存器 1（ATIM_CR1）

名称	ATIM_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	-							
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

位号	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）(Counter cloc Divider) 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 (Auto-Reload Preload Enable) 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 (Counter Mode Selection) 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 (counter Direction) 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 (One Pulse Mode) 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 (Update Request Selection) 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 (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

位号	助记符	功能描述
		- 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 (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

29.5.2 ATIM 控制寄存器 2（ATIM_CR2）

名称	ATIM_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	-	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1
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	TI1S	MMS			CCDS	CCUS	-	CCPC
位权限	R/W-0	R/W-000			R/W-0	R/W-0	U-0	R/W-0

位号	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 (Output Idle State for OC1N) 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 (Output Idle State for OC1) 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 (Timer Input 1 Selection) 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 (Master Mode Selection) 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

位号	助记符	功能描述
		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 (Capture/Compare DMA Selection) 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 (Capture/Compare Update Selection) 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
1	**-**	RFU：未实现，读为 0
0	**CCPC**	捕捉/比较预装载控制 (Capture/Compare Preload Control enable) 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（ATIM_SMCR）

名称	ATIM_SMCR							
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	ETP	ECE	ETPS		ETF			
位权限	R/W-0	R/W-0	R/W-00		R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSM	TS			-	SMS		
位权限	R/W-0	R/W-000			U-0	R/W-000		

位号	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 (External Trigger Polarity) 0：高电平或上升沿有效 1：低电平或下降沿有效

位号	助记符	功能描述
14	**ECE**	外部时钟使能 (External Clock Enable) 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 (External Trigger Prescaler) 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 (External Trigger Filter) 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**	主机从机模式选 (Master Slave Mode) 0：无动作 1：触发模式下，TRGI 触发的动作被延迟，以便于通过 TRGO 实现当前定时器和从机定时器同步
6:4	**TS**	触发选择，用于选择同步计数器的触发源 (Trigger Source) 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 (Slave Mode Selection) 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 TI2FP2 边沿，根据 TI1 电平高低来计数 010：Encoder 模式 2；计数器使用 TI1FP1 边沿，根据 TI2 电平

位号	助记符	功能描述
		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（ATIM_DIER）

名称	ATIM_DIER							
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	-				CC4BURSTEN	CC3BURSTEN	CC2BURSTEN	CC1BURSTEN
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	-	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE
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	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
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
31:20	-	RFU：未实现，读为 0
19	**CC4BURSTEN**	捕捉比较通道 4 的 DMA 模式配置 (CC4 Burst Enable) 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 (CC3 Burst Enable) 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 (CC2 Burst Enable) 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 (CC1 Burst Enable) 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 (Triggered DMA Enable) 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）

位号	助记符	功能描述
13	**COMDE**	COM 事件 DMA 请求使能 (COM event DMA Enable) 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 (CC4 DMA Enable) 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 (CC3 DMA Enable) 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 (CC2 DMA Enable) 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 (CC1 DMA Enable) 0：禁止 CC1 DMA 请求 1：允许 CC1 DMA 请求
8	**UDE**	更新事件 DMA 请求使能 (Update Event DMA Enable) 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 (Break event Interrupt Enable) 0：禁止刹车事件中断 1：允许刹车事件中断
6	**TIE**	触发事件中断使能 (Trigger event Interrupt Enable) 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 (COM event Interrupt Enable) 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 (CC4 Interrupt Enable) 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 (CC3 Interrupt Enable) 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 (CC2 Interrupt Enable) 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 (CC1 Interrupt Enable) 0：禁止捕捉/比较 1 中断 1：允许捕捉/比较 1 中断
0	**UIE**	更新事件中断使能 (Update event Interrupt Enable) 0：禁止 Update 事件中断 1：允许 Update 事件中断

29.5.5 ATIM 状态寄存器（ATIM_ISR）

名称	**ATIM_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	-			CC4OF	CC3OF	CC2OF	CC1OF	-
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-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	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	**CC4OF**	捕捉/比较通道 4 的 Overcapture 中断 (Over-Capture Interrupt Flag for CC4, write 1 to clear) 参考 CC1OF
11	**CC3OF**	捕捉/比较通道 3 的 Overcapture 中断 (Over-Capture Interrupt Flag for CC3, write 1 to clear) 参考 CC1OF
10	**CC2OF**	捕捉/比较通道 2 的 Overcapture 中断 (Over-Capture Interrupt Flag for CC2, write 1 to clear) 参考 CC1OF
9	**CC1OF**	捕捉/比较通道 1 的 Overcapture 中断 (Over-Capture Interrupt Flag for CC1, write 1 to clear) 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 (Break Interrupt Flag, write 1 to clear)
6	**TIF**	触发事件中断标志，硬件置位，软件写 1 清零 (Trigger Interrupt Flag, write 1 to clear)
5	**COMIF**	COM 事件中断标志，硬件置位，软件写 1 清零 (COM Interrupt Flag, write 1 to clear)
4	**CC4IF**	捕捉/比较通道 4 中断标志 (CC4 Interrupt Flag, write 1 to clear) 参考 CC1IF
3	**CC3IF**	捕捉/比较通道 3 中断标志 (CC3 Interrupt Flag, write 1 to clear) 参考 CC3IF
2	**CC2IF**	捕捉/比较通道 2 中断标志 (CC2 Interrupt Flag, write 1 to clear) 参考 CC2IF
1	**CC1IF**	捕捉/比较通道 1 中断标志 (CC1 Interrupt Flag, write 1 to clear) 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**	更新事件中断标志，硬件置位，软件写 1 清零。(Update event Interrupt Flag,write 1 to clear) 当以下事件发生时，UIF 置位，并更新 shadow 寄存器 -重复结束，并且 UDIS=0 的情况下，计数器发生溢出

位号	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

29.5.6 ATIM 事件产生寄存器（ATIM_EGR）

名称	ATIM_EGR							
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	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

位号	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 (Break Generate)
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 (Trigger Interrupt Flag)
5	**COMG**	软件 COM 事件，硬件置位，软件写 1 清零 (COMG Generate)
4	**CC4G**	捕捉/比较通道 4 软件触发，参考 CC1G (CC4 Generate)
3	**CC3G**	捕捉/比较通道 3 软件触发，参考 CC1G (CC3 Generate)
2	**CC2G**	捕捉/比较通道 2 软件触发，参考 CC1G (CC2 Generate)
1	**CC1G**	捕捉/比较通道 1 软件触发 (CC1 Generate) 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 (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。

29.5.7 ATIM 捕捉/比较模式寄存器 1（ATIM_CCMR1）

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							
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	OC2CE	OC2M			OC2PE	OC2FE	CC2S	
	IC2F				IC2PSC		CC2S	
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	OC1CE	OC1M			OC1PE	OC1FE	CC1S	
	IC1F				IC1PSC		CC1S	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	

输出比较模式

位号	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 (OC2 Clear Enable)
14:12	**OC2M**	输出比较 2 模式配置，参考 OC1M (OC2 Mode)
11	**OC2PE**	输出比较 2 预装载使能，参考 OC1PE (OC2 Preload Enable)
10	**OC2FE**	输出比较 2 快速使能，参考 OC1FE (OC2 Fast Enable)
9:8	**CC2S**	捕捉/比较 2 通道选择 (CC2 Channel Selection) 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 (OC1 Clear Enable) 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 (OC1 Mode) 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 (OC1 Preload Enable) 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

位号	助记符	功能描述
		的内容转移到 shadow 寄存器中
2	**OC1FE**	输出比较 1 快速使能 (OC1 Fast Enable) 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（只用于触发模式） 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 (CC1 Channel Selection) 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

位号	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 (IC2 Filter)
11:10	**IC2PSC**	输入捕捉 2 预分频 (IC2 Prescaler)
9:8	**CC2S**	捕捉/比较 2 通道选择 (CC2 Channel Selection) 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 (IC1 Filter) 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 (IC1 Prescaler) 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 (CC1 Channel Selection)

位号	助记符	功能描述
		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（ATIM_CCMR2）

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							
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	OC4CE	OC4M			OC4PE	OC4FE	CC4S	
	IC4F				IC4PSC		CC4S	
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	OC3CE	OC3M			OC3PE	OC3FE	CC3S	
	IC3F				IC3PSC		CC3S	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	

输出比较模式

位号	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 (OC4 Clear Enable)
14:12	**OC4M**	输出比较 4 模式配置，参考 OC1M (OC4 Mode)
11	**OC4PE**	输出比较 4 预装载使能，参考 OC1PE (OC4 Preload Enable)
10	**OC4FE**	输出比较 4 快速使能，参考 OC1FE (OC4 Fast Enable)
9:8	**CC4S**	捕捉/比较 4 通道选择 (CC4 Channel Selection) 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(OC3 Clear Enable) 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 (OC3 Mode) 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

位号	助记符	功能描述
		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 (OC3 Preload Enable) 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 (OC3 Fast Enable) 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 (CC4 Channel Selection) 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

位号	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 (IC4 Filter)
11:10	**IC4PSC**	输入捕捉 4 预分频 (IC4 Prescaler)
9:8	**CC4S**	捕捉/比较 4 通道选择 (CC4 channel Selection) 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 3 滤波 (IC3 Filter) 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

位号	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
3:2	**IC3PSC**	输入捕捉 3 预分频 (IC3 Prescaler) 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 (CC3 channel Selection) 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（ATIM_CCER）

名称	ATIM_CCER							
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	-		CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E
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	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
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
31:14	**-**	RFU：未实现，读为 0
13	**CC4P**	捕捉/比较 4 输出极性，参考 CC1P (CC4 Polarity)
12	**CC4E**	捕捉/比较 4 输出使能，参考 CC1E (CC4 Enable)
11	**CC3NP**	捕捉/比较 3 互补输出极性，参考 CC1NP (CC3N Polarity)
10	**CC3NE**	捕捉/比较 3 互补输出使能，参考 CC1NE (CC3N Enable)
9	**CC3P**	捕捉/比较 3 输出极性，参考 CC1P (CC3 Polarity)
8	**CC3E**	捕捉/比较 3 输出使能，参考 CC1E (CC3 Enable)
7	**CC2NP**	捕捉/比较 2 互补输出极性，参考 CC1NP (CC2N Polarity)
6	**CC2NE**	捕捉/比较 2 互补输出使能，参考 CC1NE (CC2N Enable)

位号	助记符	功能描述
5	**CC2P**	捕捉/比较 2 输出极性，参考 CC1P (CC2 Polarity)
4	**CC2E**	捕捉/比较 2 输出使能，参考 CC1E (CC2 Enable)
3	**CC1NP**	捕捉/比较 1 互补输出极性 (CC1N Polarity) 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 (CC1N Enable) 0：OC1N 无效，OC1N 电平由 MOE, OSSI, OSSR, OIS1, OIS1N, CC1E 寄存器决定
1	**CC1P**	捕捉/比较 1 输出极性 (CC1 Polarity) 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 Enable) CC1 通道配置为输出时 0：OC1 不 active 1：OC1 active 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（ATIM_CNT）

名称	ATIM_CNT							
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	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
31:16	**-**	RFU：未实现，读为 0
15:0	**CNT**	计数器值 (Counter)

29.5.11 ATIM 预分频寄存器（ATIM_PSC）

名称	ATIM_PSC

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	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
31:16	-	RFU：未实现，读为 0
15:0	**PSC**	计数器时钟（CK_CNT）预分频值 (Prescaler) $f_{CK_CNT}=f_{CK_PSC}/(PSC[15:0]+1)$ 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 自动重载寄存器（ATIM_ARR）

名称	ATIM_ARR							
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	ARR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-1111 1111							

位号	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 (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

29.5.13 ATIM 重复计数寄存器（ATIM_RCR）

名称	ATIM_RCR
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	REP[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	**REP**	重复计数值。(Repetition) 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（ATIM_CCR1）

名称	ATIM_CCR1							
Offset	0x00000034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	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 (Capture/Compare channel 1 Register) **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（ATIM_CCR2）

名称	ATIM_CCR2							
Offset	0x00000038							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	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 (Capture/Compare channel 2 Register) **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（ATIM_CCR3）

名称	ATIM_CCR3							
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	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	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 (Capture/Compare channel 3 Register) **如果通道 3 配置为输出：**

位号	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 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（ATIM_CCR4）

名称	ATIM_CCR4							
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	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	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 (Capture/Compare channel 4 Register) **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（ATIM_BDTR）

名称	ATIM_BDTR							
Offset	0x00000044							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	DTG
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	**MOE**	输出使能主控 (Master Output Enable) 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 (Automatic Output Enable) 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 (Break Polarity) 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 (Break Enable) 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 (Off-State Select in Run mode) 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 (Off-State Select in IDLE mode) 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 先驱动空闲状态，待死区时间结束后，启动无效状态
9:8	**LOCK**	寄存器写保护配置 (register write LOCK) 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, 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 寄存器在被写入非 00 值之后无法再改写，写保护后的寄存器只有在 ATIM 模块被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 (Dead Time Generation) 000/001/010/011：DT=DTG[7:0] * t_{DTS} 100/101：DT=(64+DTG[5:0]) * 2 * t_{DTS} 110：DT=(32+DTG[4:0]) * 8 * t_{DTS} 111：DT=(32+DTG[4:0]) * 16 * t_{DTS}

29.5.19 ATIM DMA 控制寄存器（ATIM_DCR）

名称	ATIM_DCR							
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	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			DBL				
位权限	U-0			R/W-0 0000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			DBA				
位权限	U-0			R/W-0 0000				

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12:8	**DBL**	DMA Burst 长度 (DMA Burst Length) 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 00010：长度=3 00011：长度=4 00100：长度=5 00101：长度=6 00110：长度=7 00111：长度=8 01000：长度=9 01001：长度=10 01010：长度=11 01011：长度=12 01100：长度=13 01101：长度=14 01110：长度=15 01111：长度=16 10000：长度=17 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 (DMA Burst Address) 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（ATIM_DMAR）

名称	ATIM_DMAR							
Offset	0x0000004C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DMAR[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DMAR[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DMAR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DMAR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**DMAR**	DMA burst 访问寄存器 (DMA burst access Register) 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（ATIM_BKCR）

名称	ATIM_BKCR							
Offset	0x00000060							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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						R/W-1	R/W-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BRKF				BRKCOMB	HFDET_BRKEN	SVD_BRKEN	COMP_BRKEN
位权限	R/W-0000				R/W-0	R/W-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	**BRK2GATE**	ATIM_BRK2 引脚输入门控信号 (Break 2 Gate) 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 (Break 1 Gate) 0：将 ATIM_BRK1 的输入门控成 0

位号	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 (Break Filter) 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 (Break Combination) 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 (HFDET Break Enable) 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 (SVD Break Enable) 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 (Comparator Break Enable) 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（GPTIM0,1,2）

30.1 概述

FM33LG0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

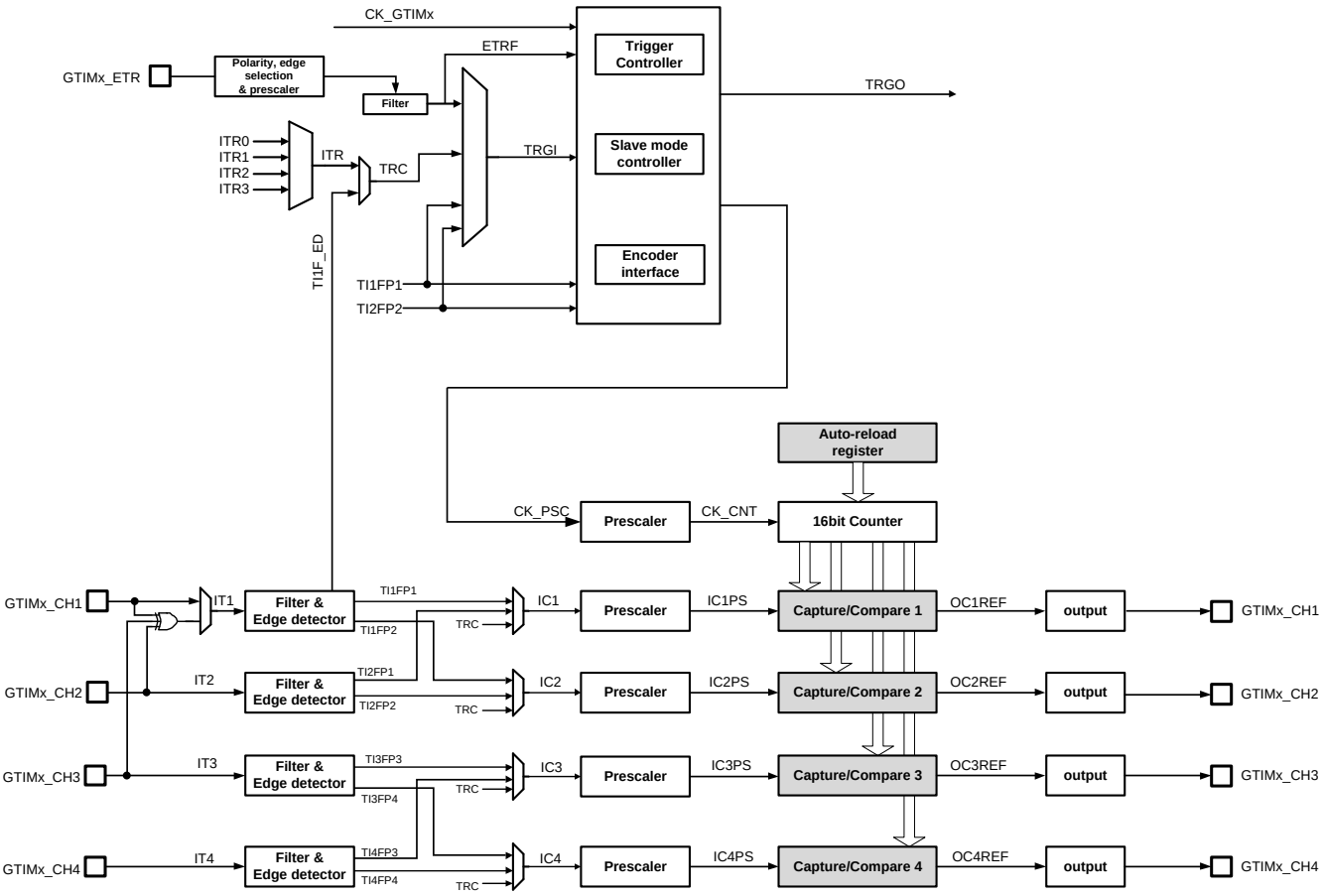


图30-1通用定时器架构示意图

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。

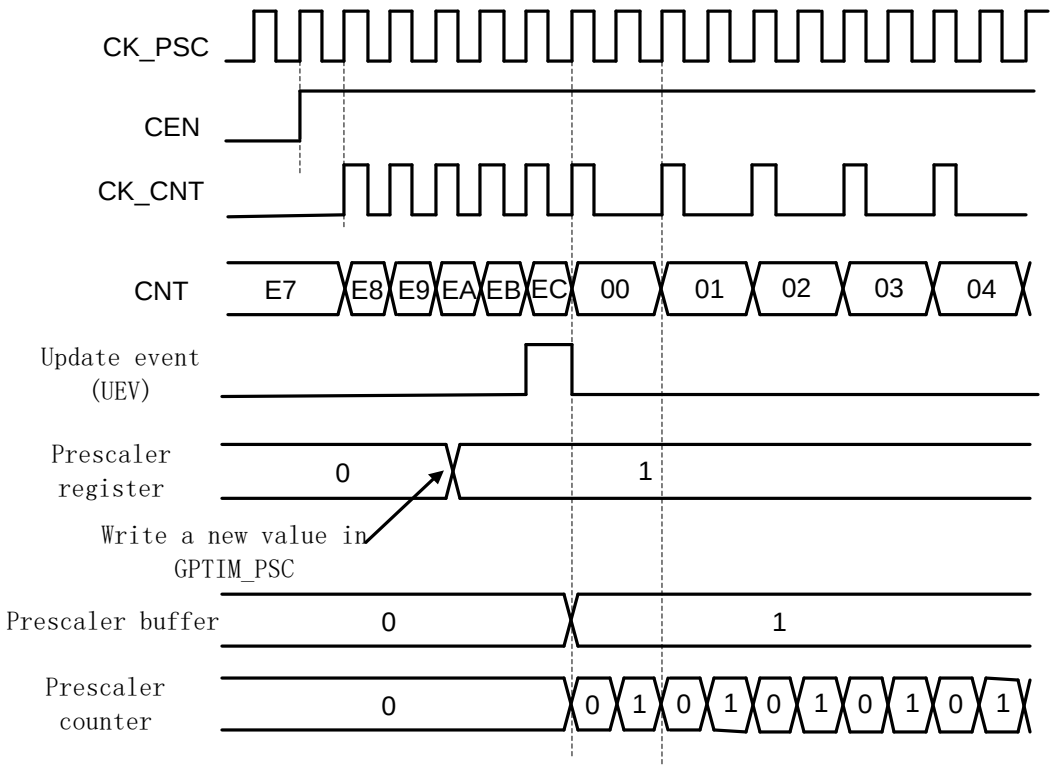


图 30-2 预分频从 1 变为 2 的波形

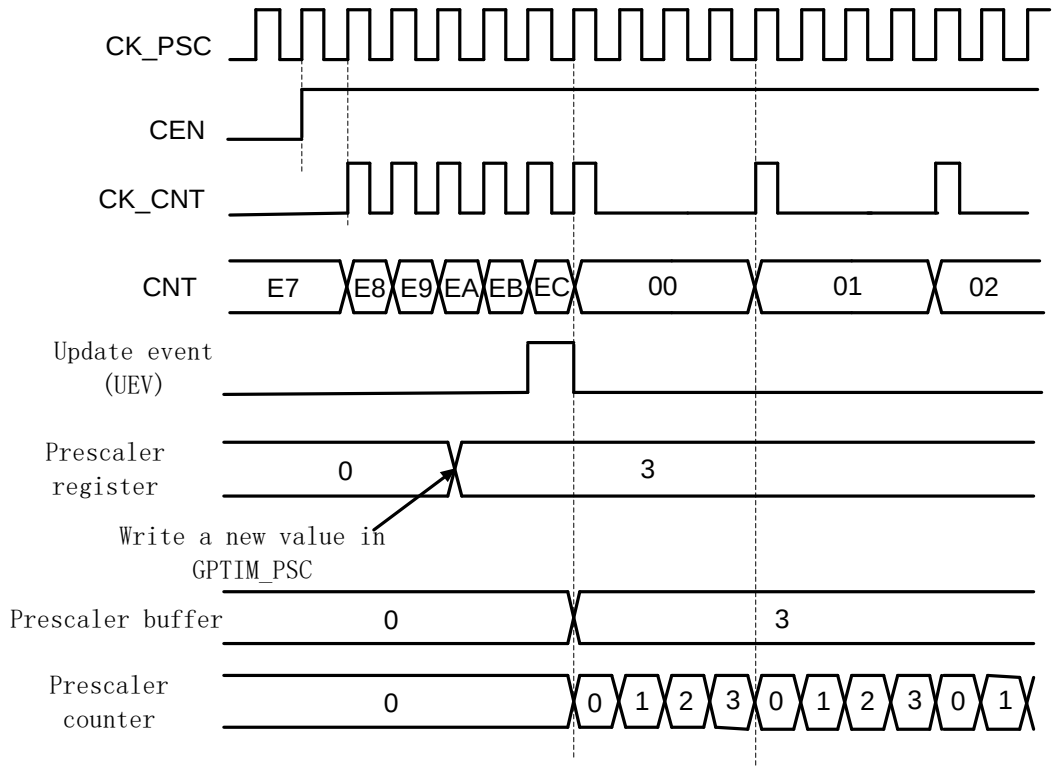


图 30-3 预分频从 1 变为 4 的波形

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

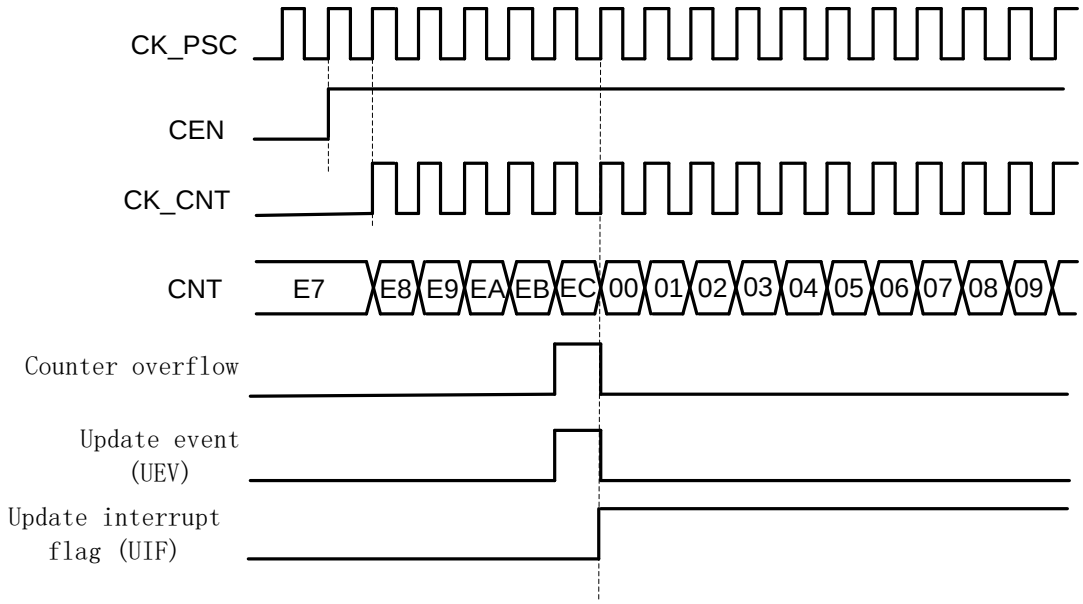


图 30-4 向上计数波形，内部时钟不分频

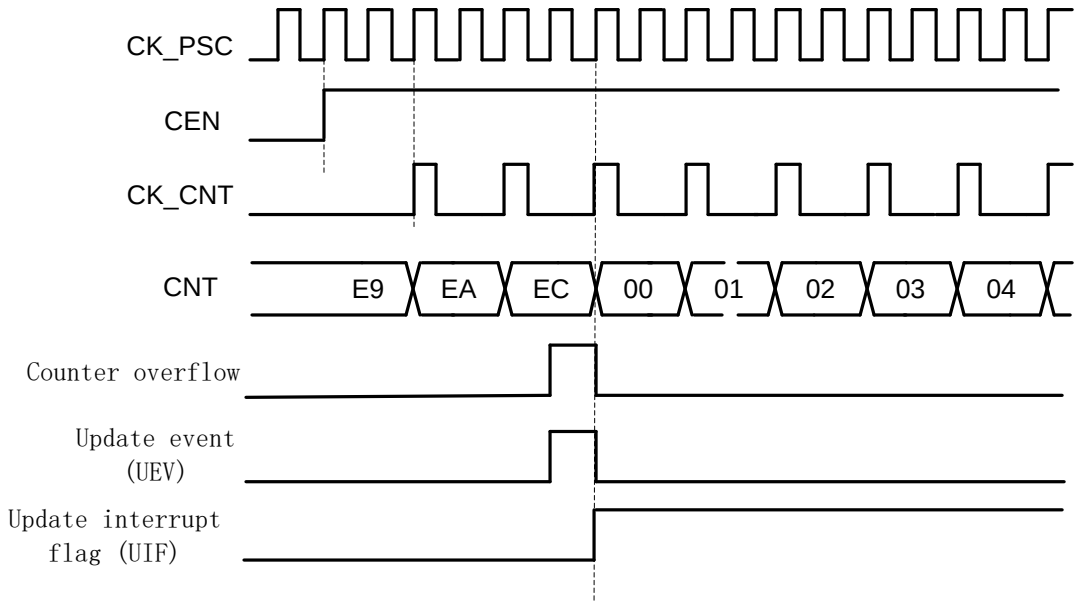


图 30-5 向上计数波形，内部时钟 2 分频

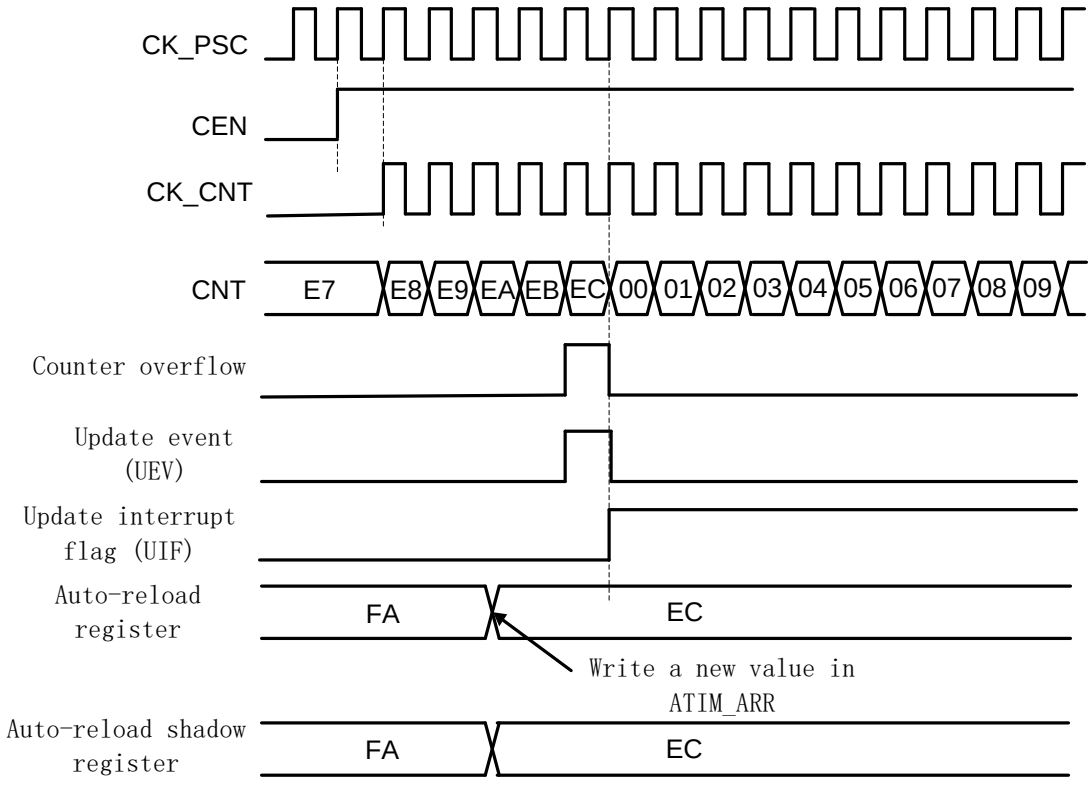


图 30-6ARPE=0（GPTIM_ARR 没有预装载）时的更新事件

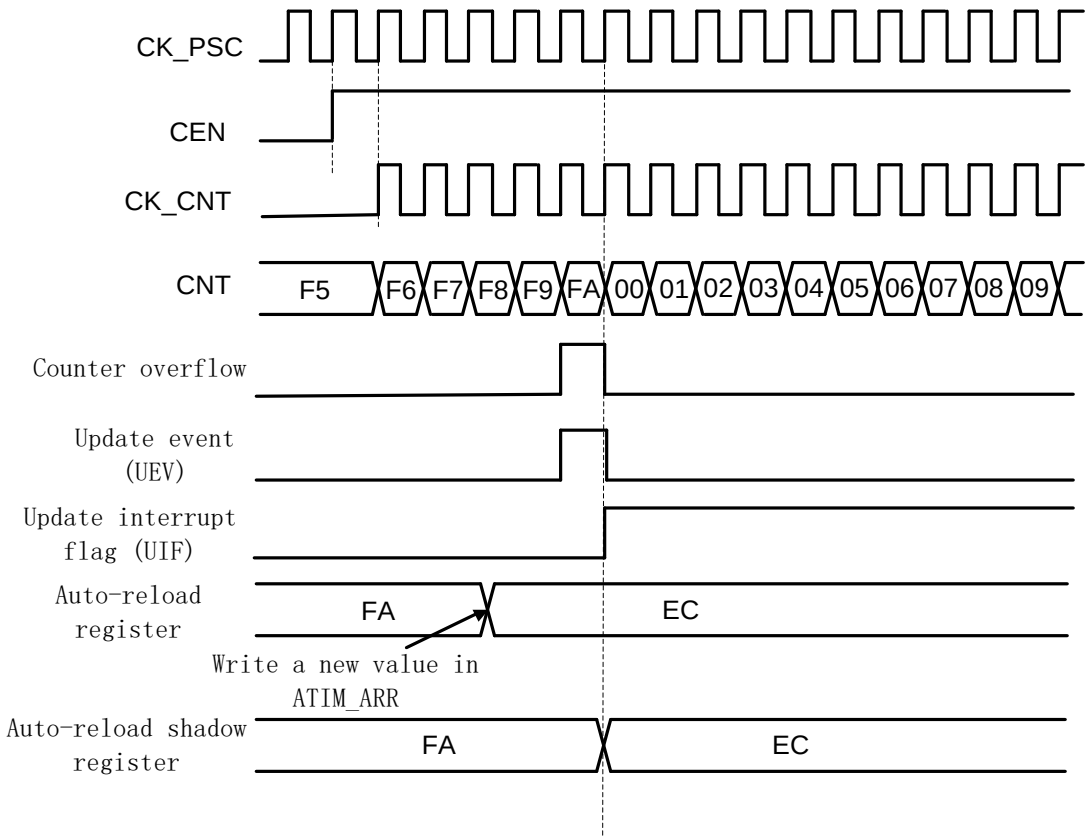


图 30-7ARPE=1（GPTIM_ARR 预装载）时的更新事件

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

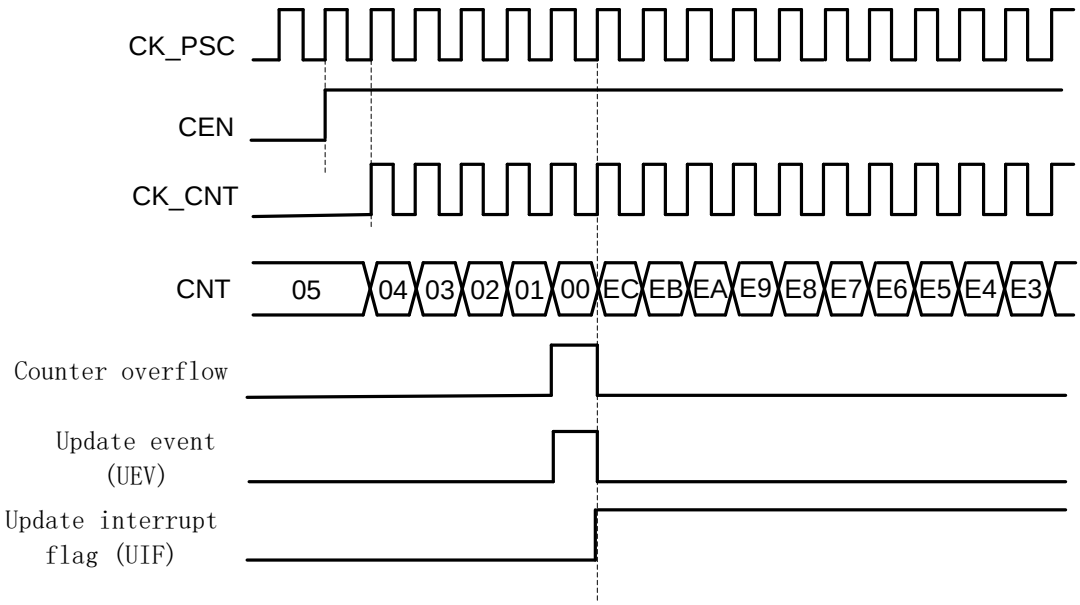


图 30-8 向下计数，内部时钟不分频

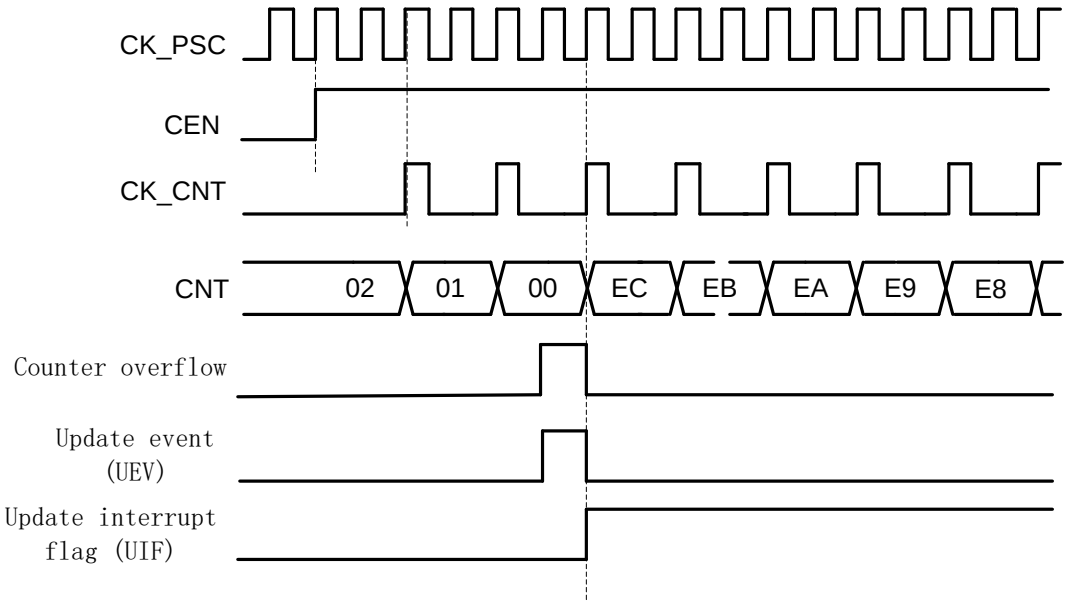


图 30-9 向下计数，内部时钟 2 分频

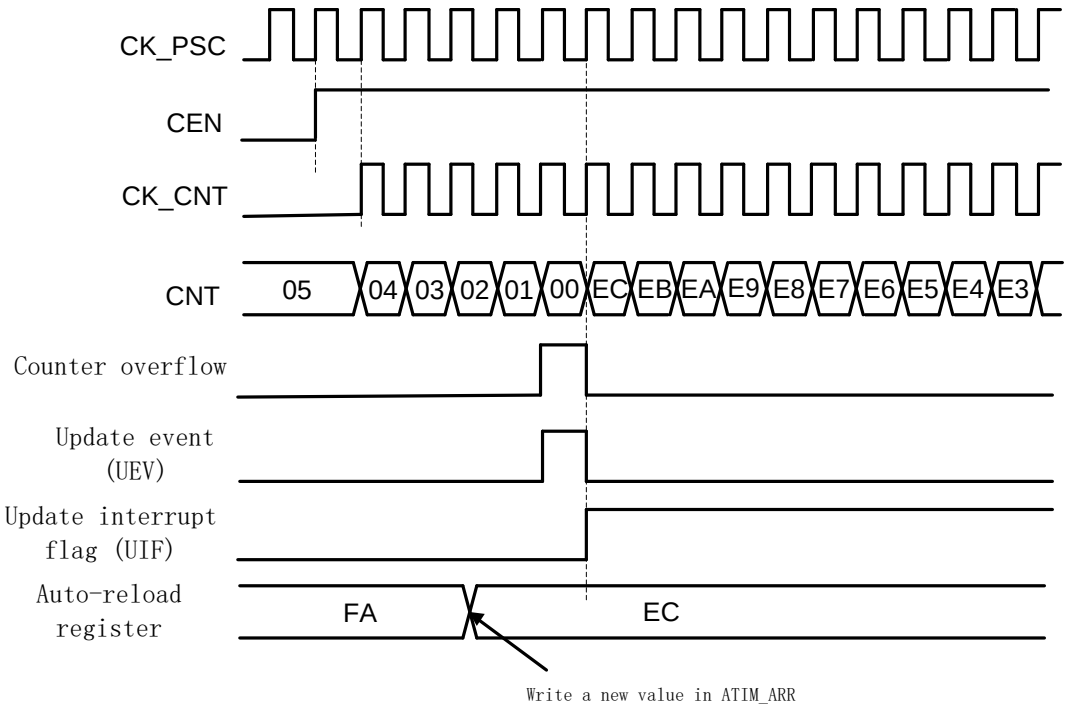


图 30-10 向下计数，内部时钟 2 分频

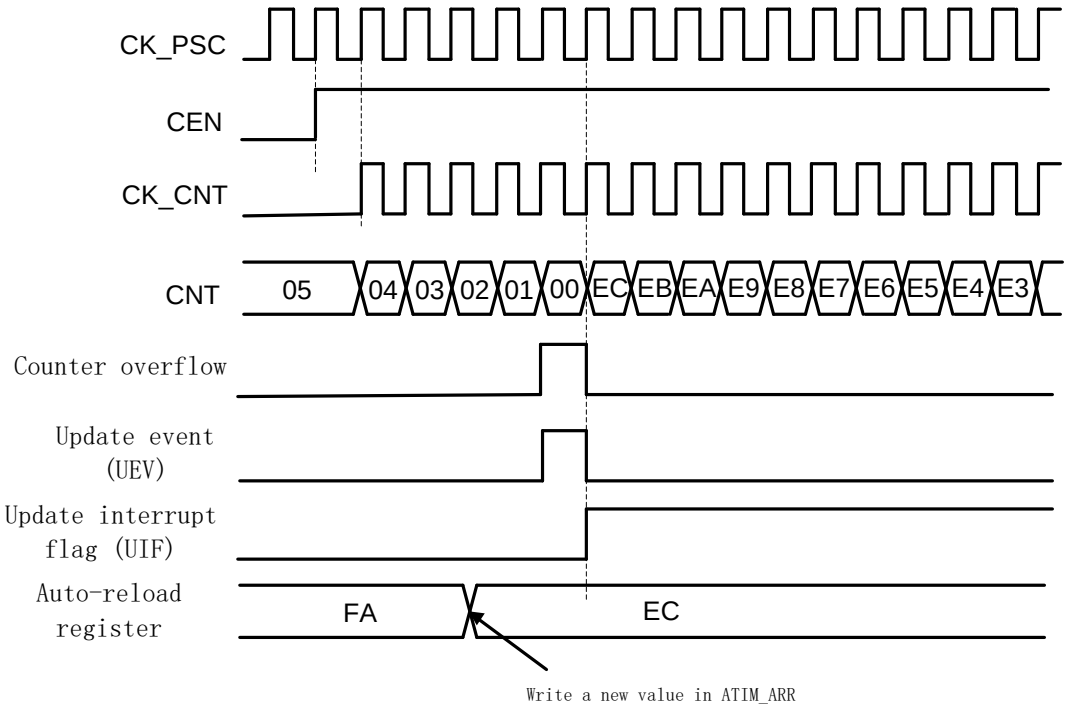


图 30-11 向下计数，不使用重复计数时的更新事件

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。

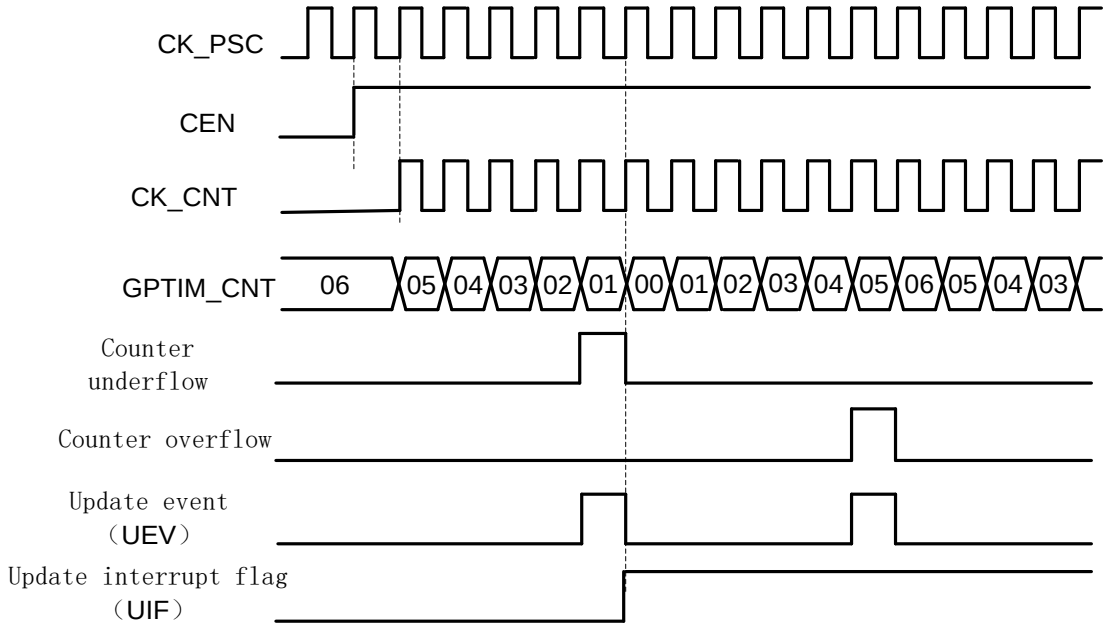


图 30-12 中心对齐计数器时序图，GPTIM_PCS=0，GPTIM_ARR=0x6

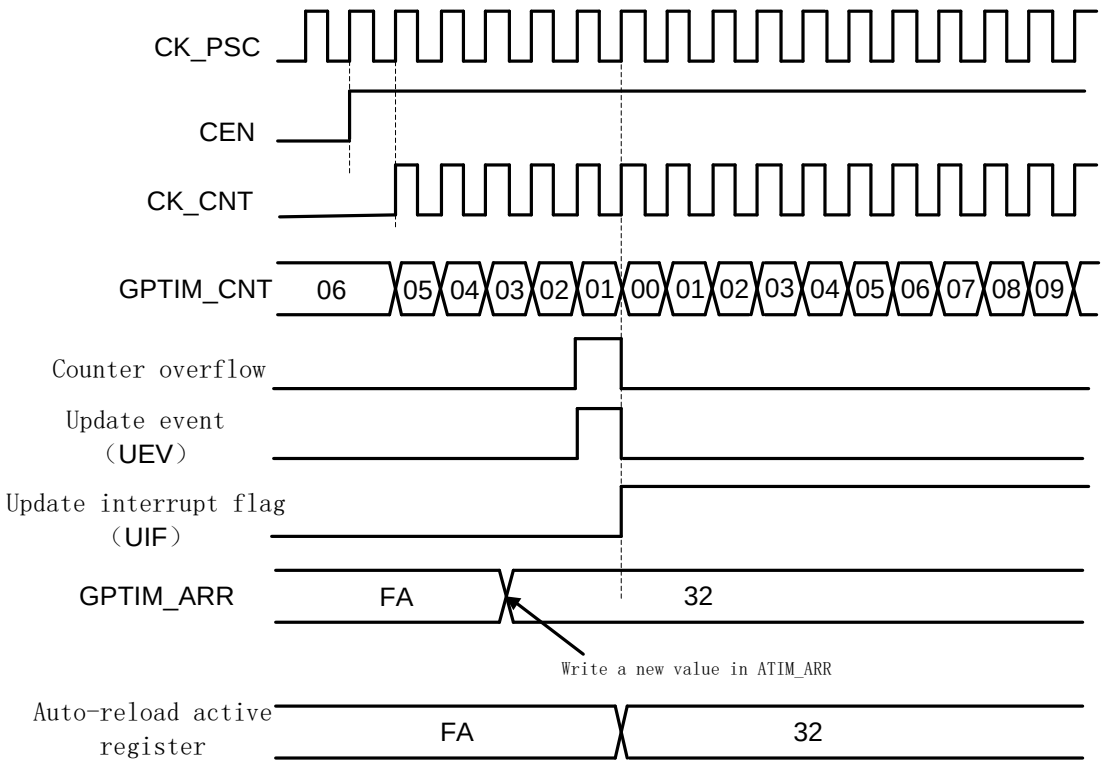


图 30-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

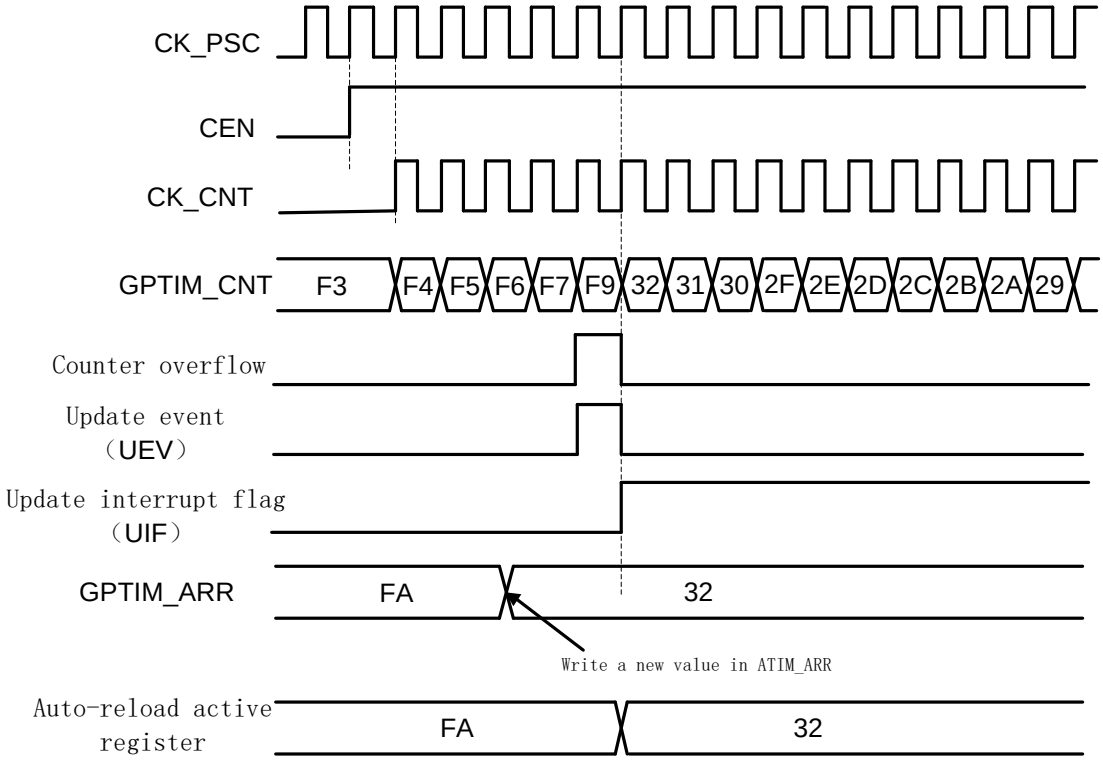


图 30-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

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

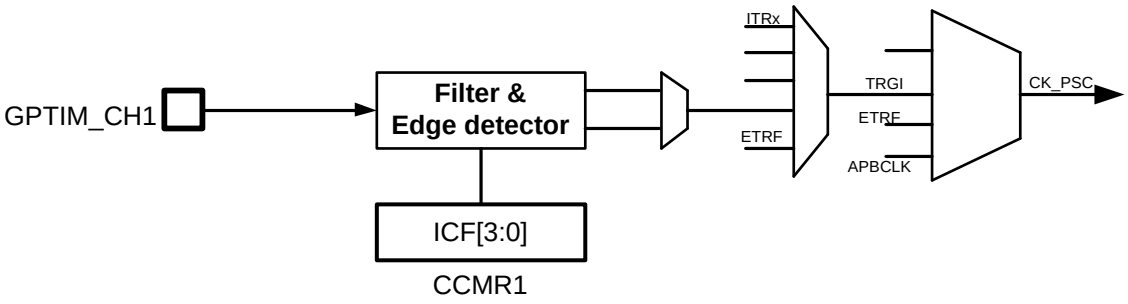


图 30-15GPTIM 时钟源框图

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。

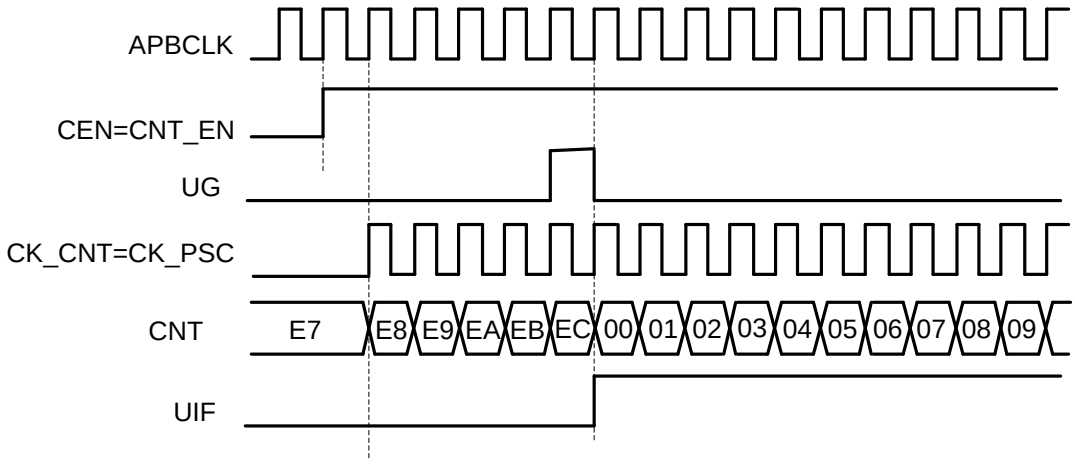


图 30-16 内部时钟源模式，时钟分频因子为 1

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。

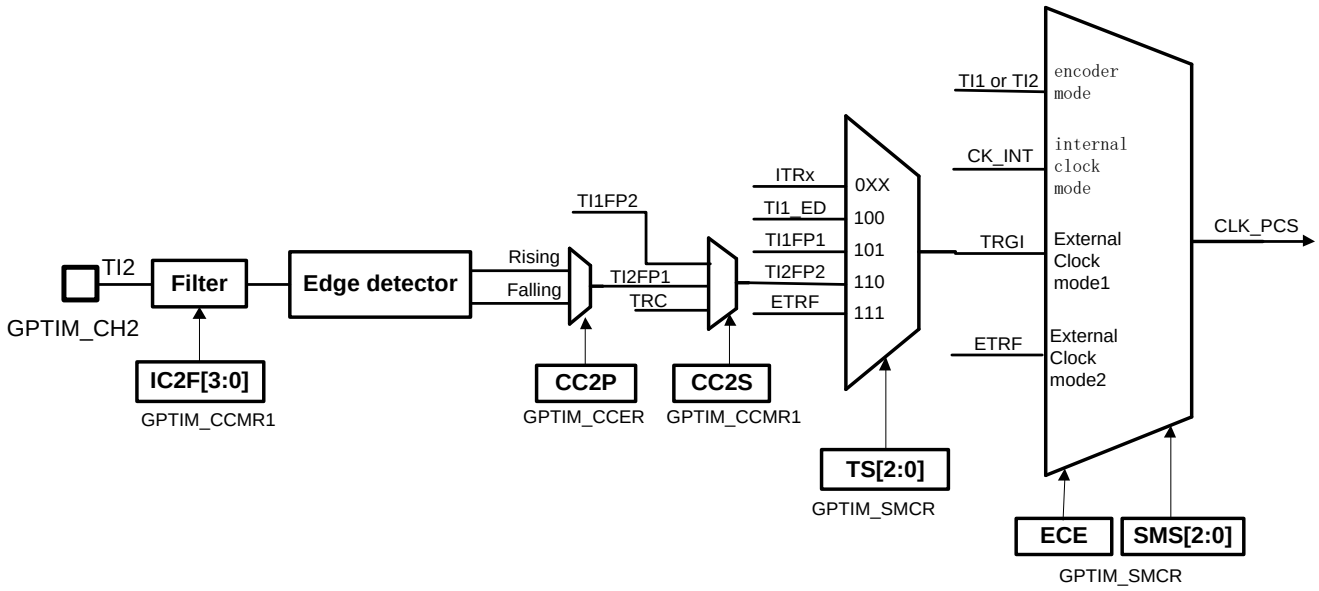


图 30-17TI2 外部时钟连接例子

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

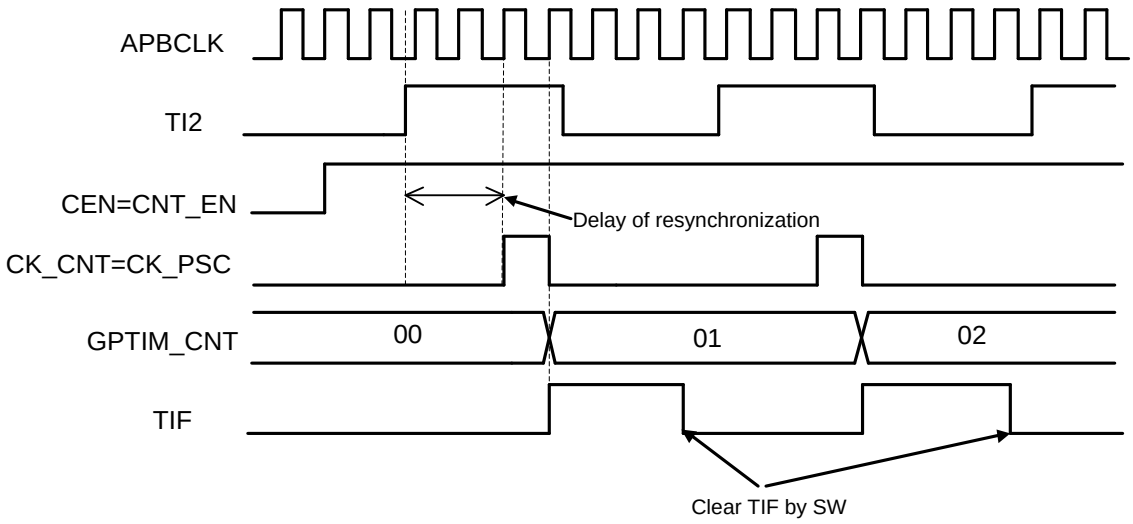


图 30-18 外部时钟模式 1 下的时序

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：

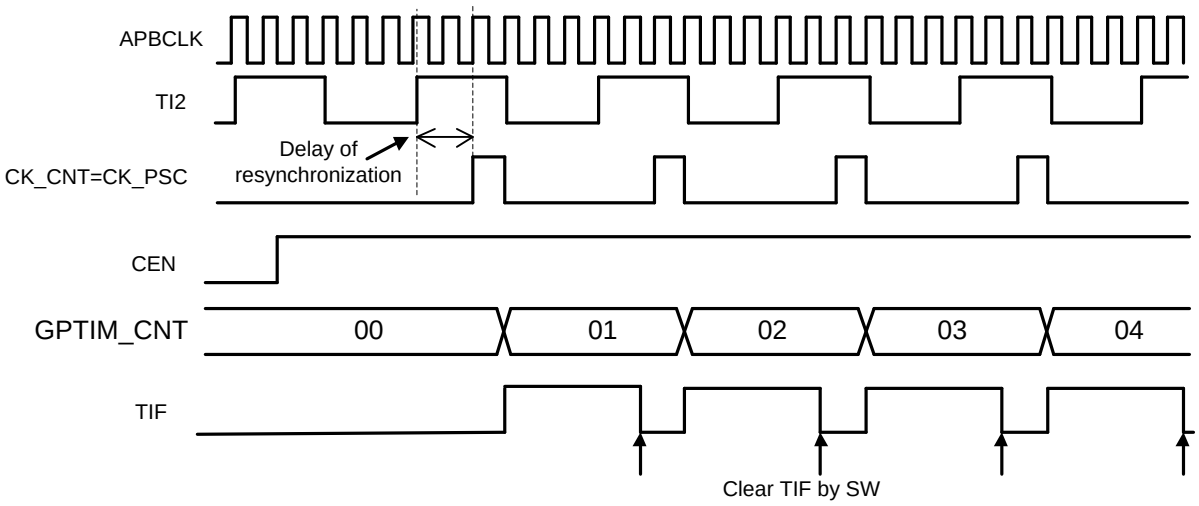


图 30-19 外部时钟模式 1 下的时序

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。

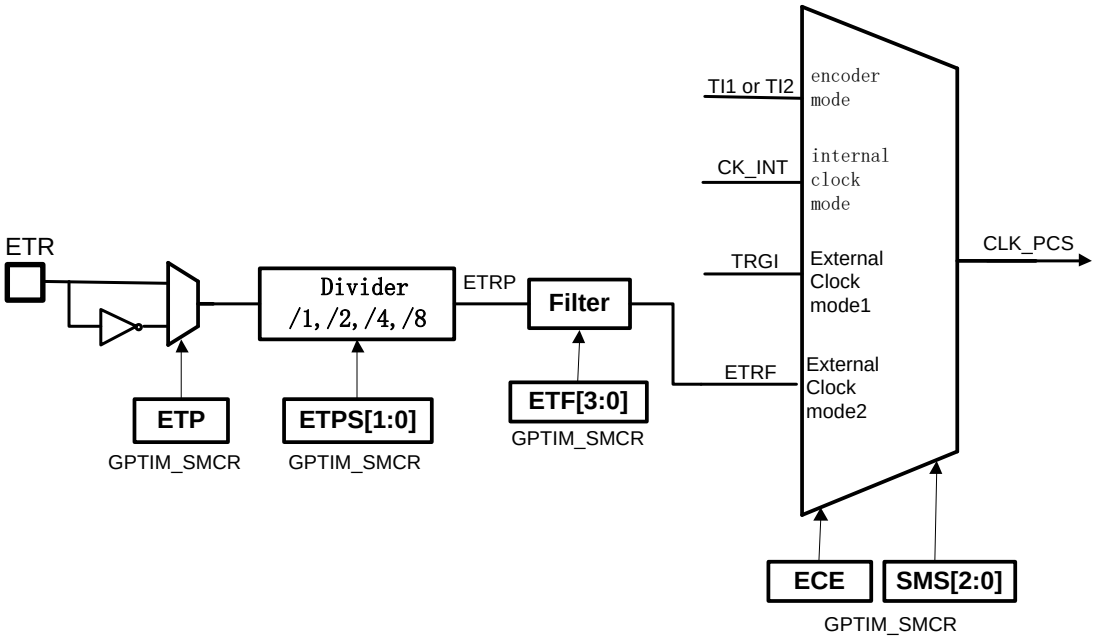


图 30-20 外部触发输入框图

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。

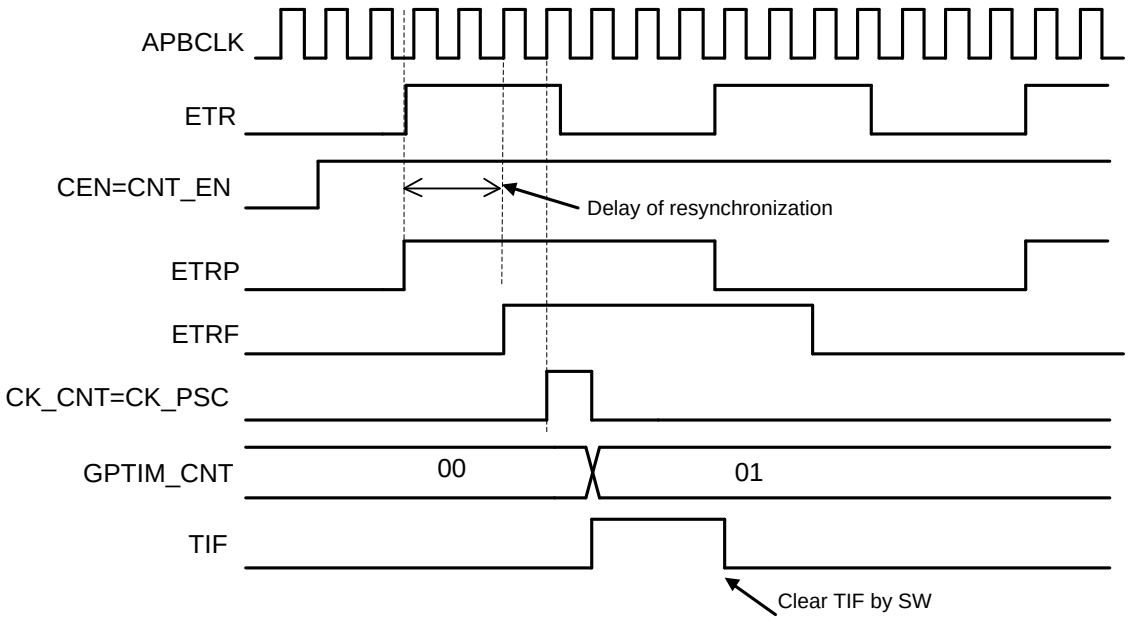


图 30-21 外部时钟模式 2 下的时序 1

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：

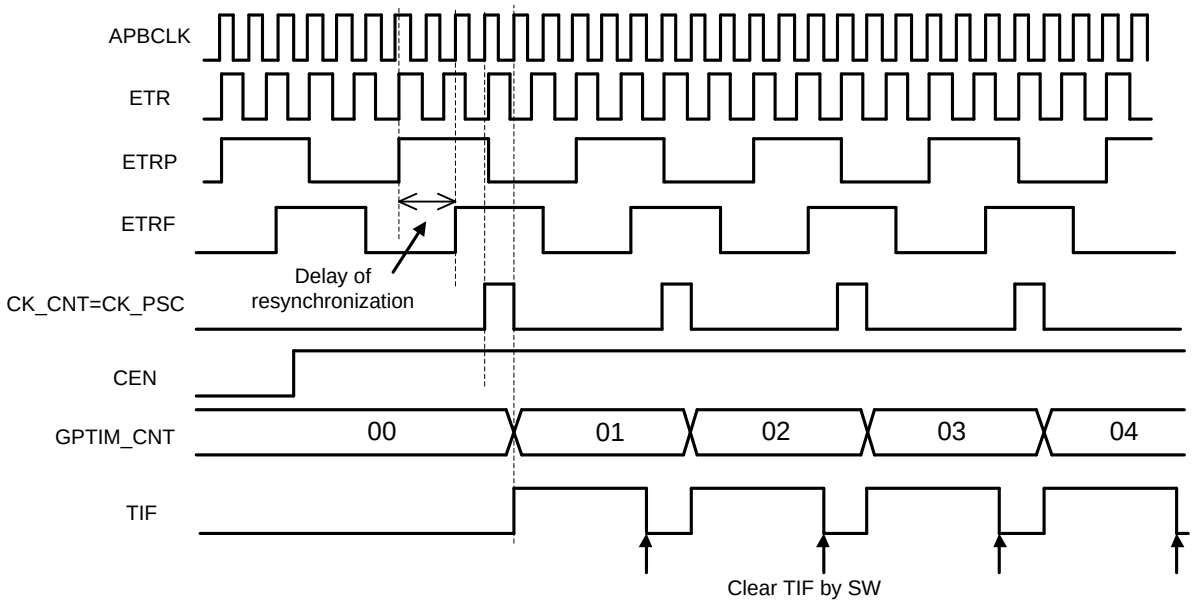
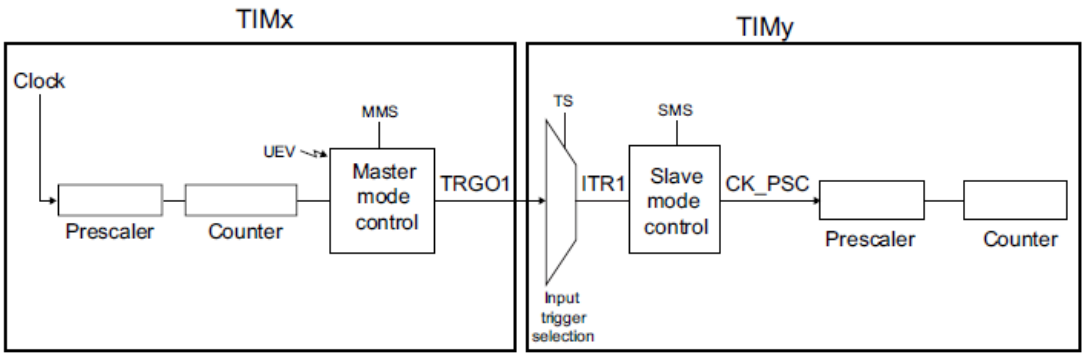


图30-22外部时钟模式2下的时序2

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_OUT	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

其中，作为计数触发的TRGO信号，在到达GPTIM的ITRx之前都被处理成一个APBCLK宽度的同步使能信号。而用作周期或宽度捕捉的信号，则无需处理，直接由GPTIM进行捕捉。

上表中红色字体部分表示需要master处理成APBCLK同步脉冲的信号。

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。

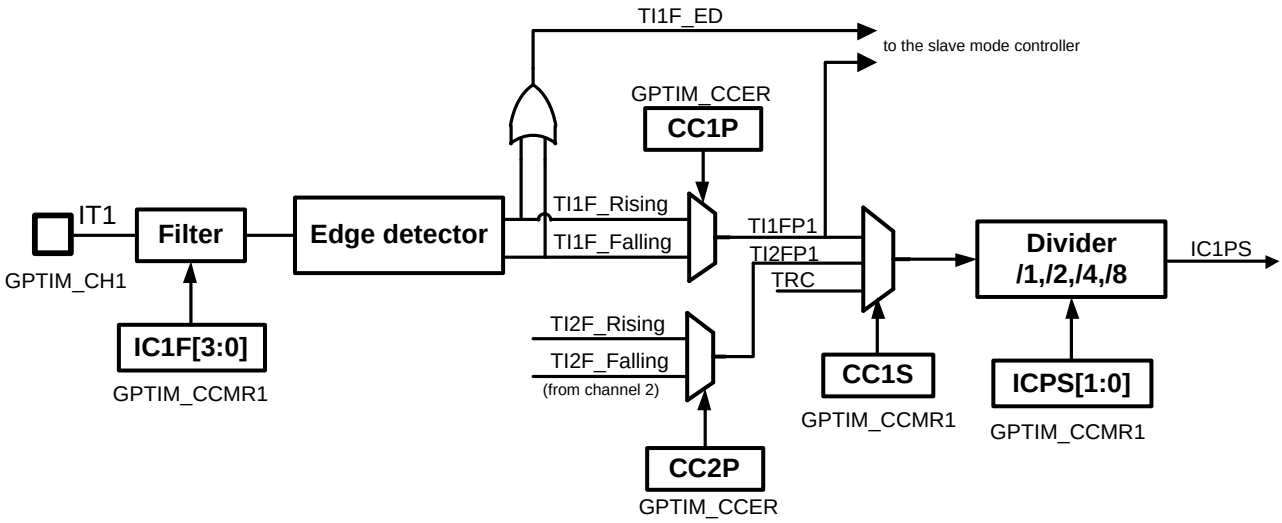


图30-23捕获/比较通道(通道1输入部分)

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。

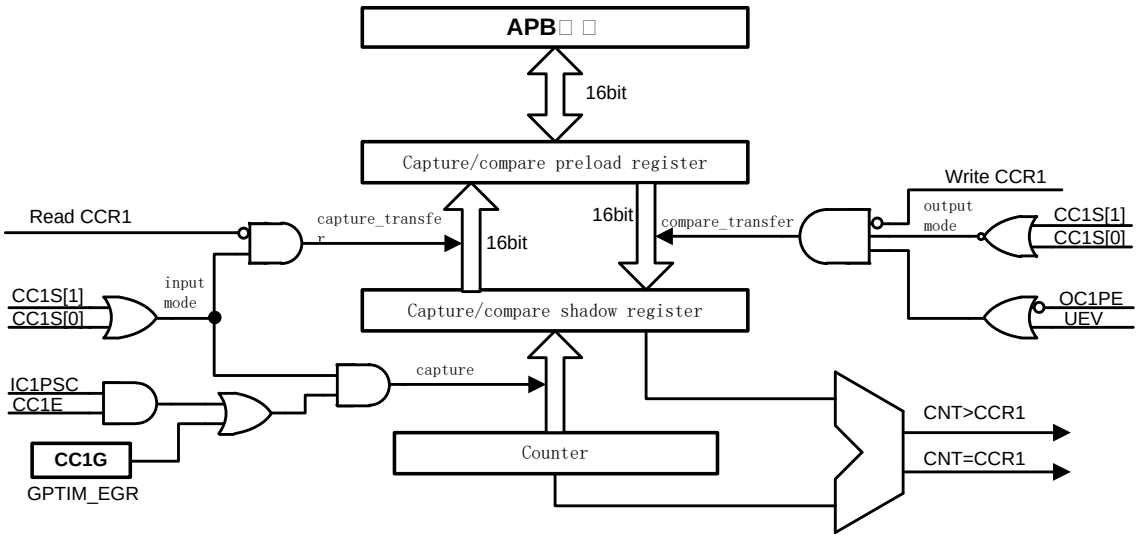


图30-24捕获/比较通道1的主电路

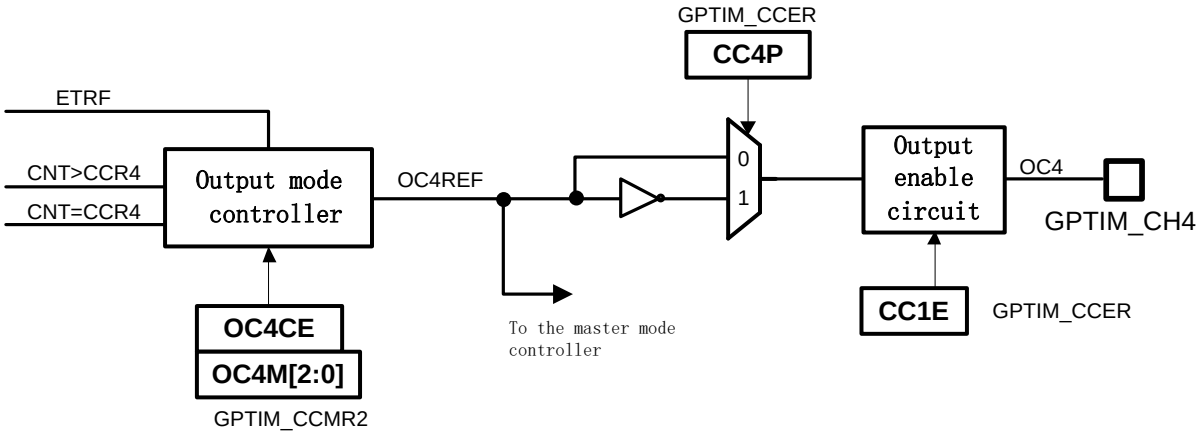


图30-25捕获/比较通道的输出部分

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

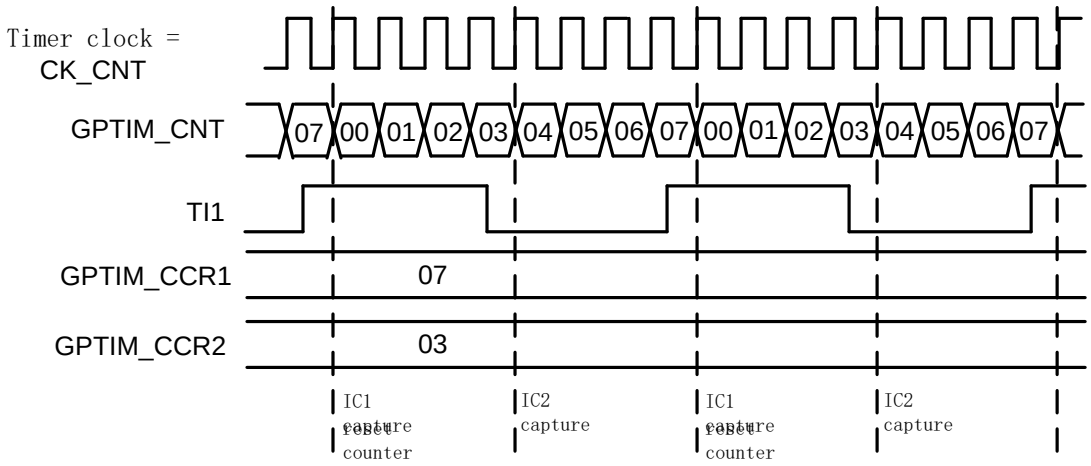


图30-26PWM输入捕获模式时序

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

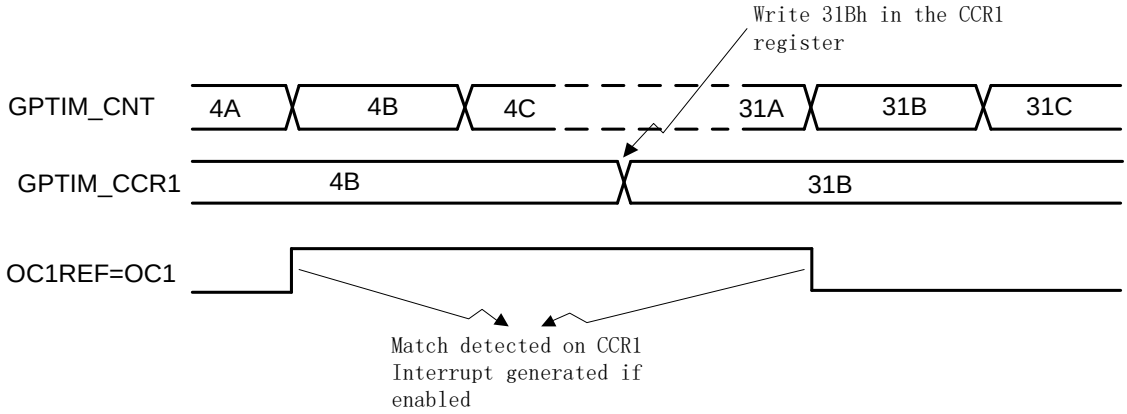


图30-27输出比较模式，翻转OC1

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.

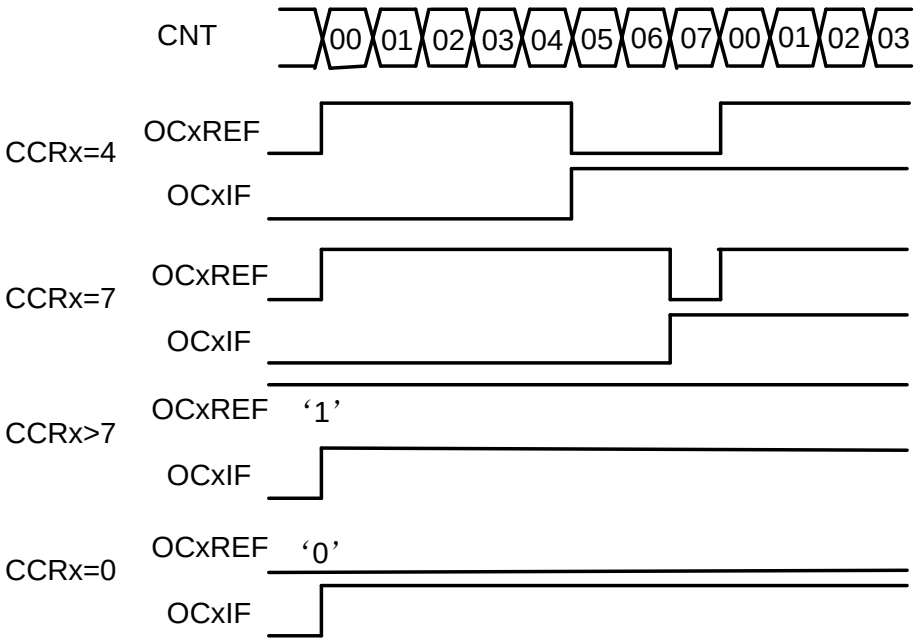


图30-28边沿对齐的PWM波形(ARR=7)

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：

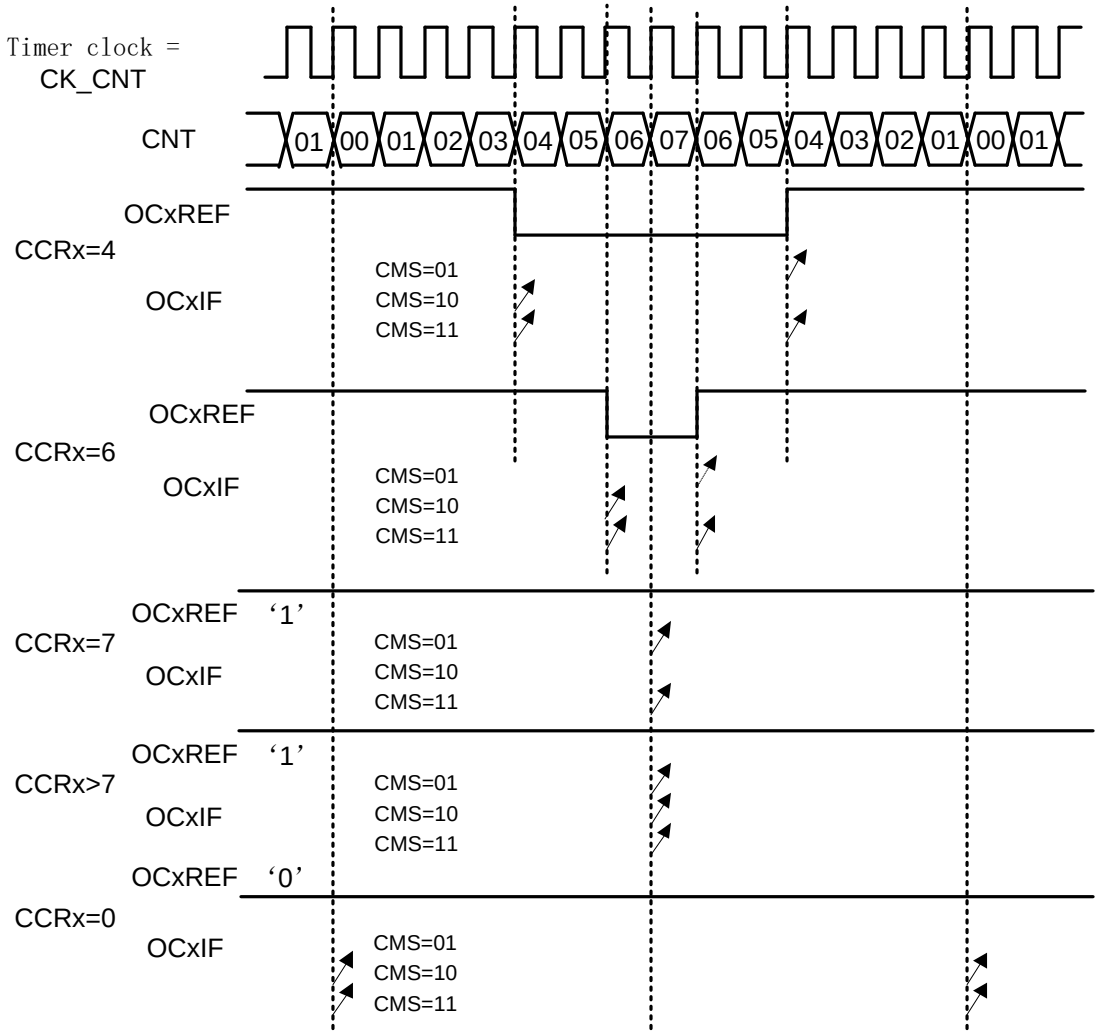


图30-29中央对齐的PWM波形(APR=7)

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

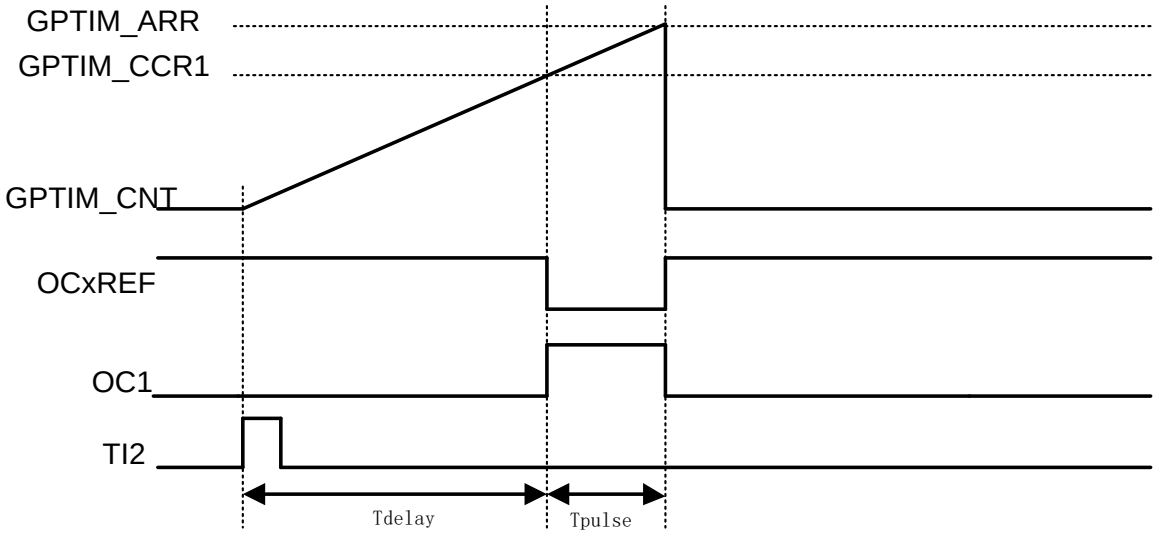


图30-30单脉冲模式的例子

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.

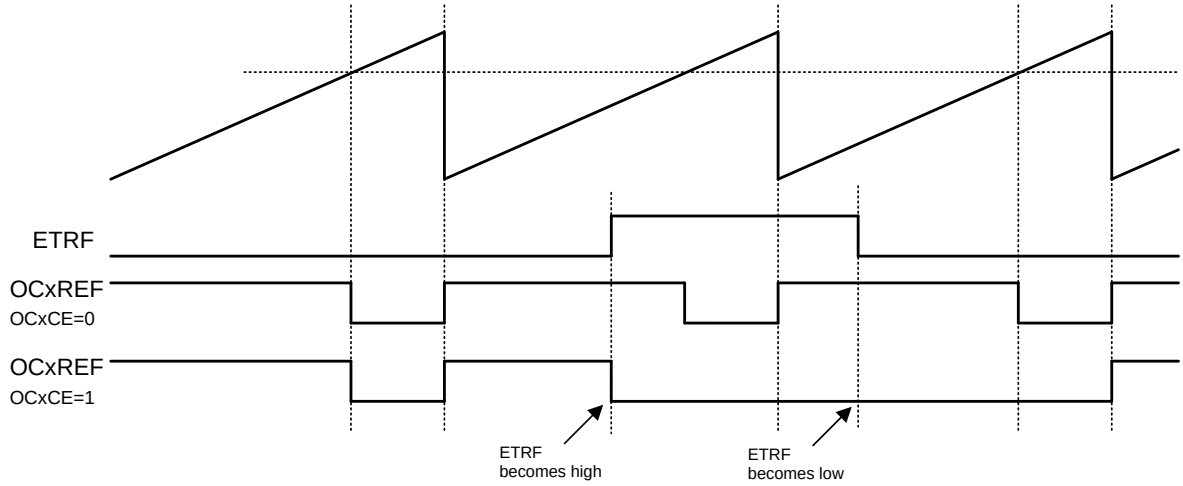


图30-31 ETR信号清除GPTIM的OCxREF

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。

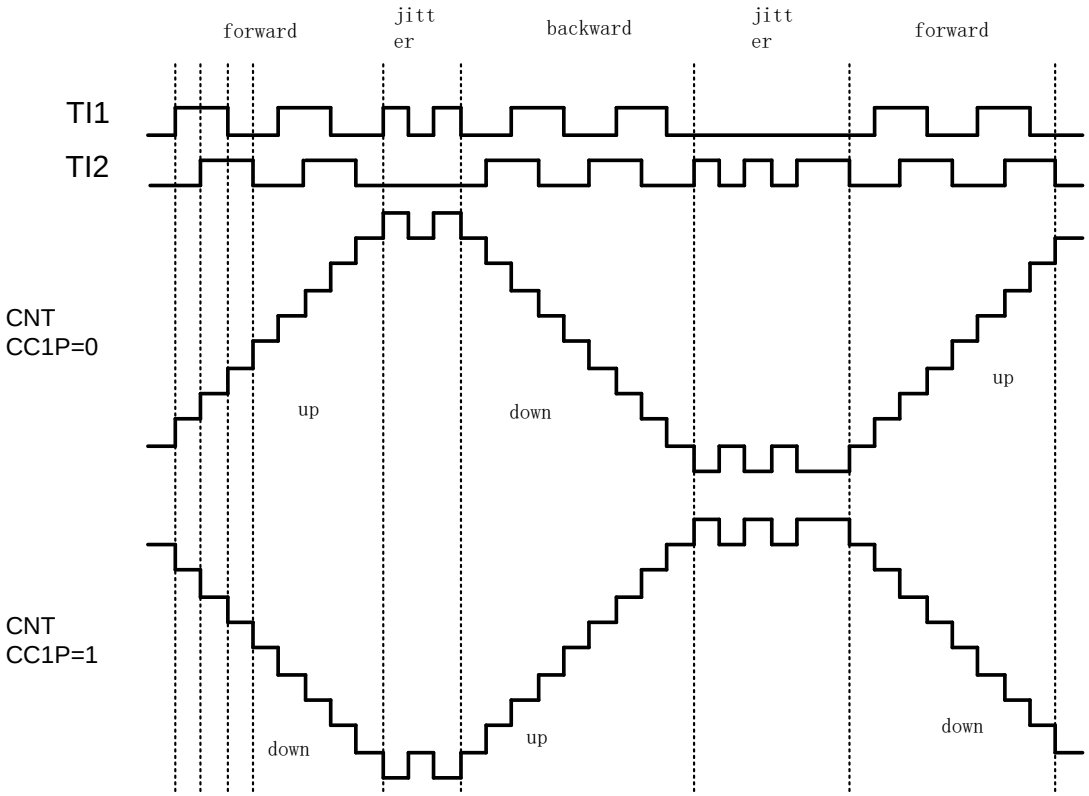


图30-32编码器模式下的计数器操作实例

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

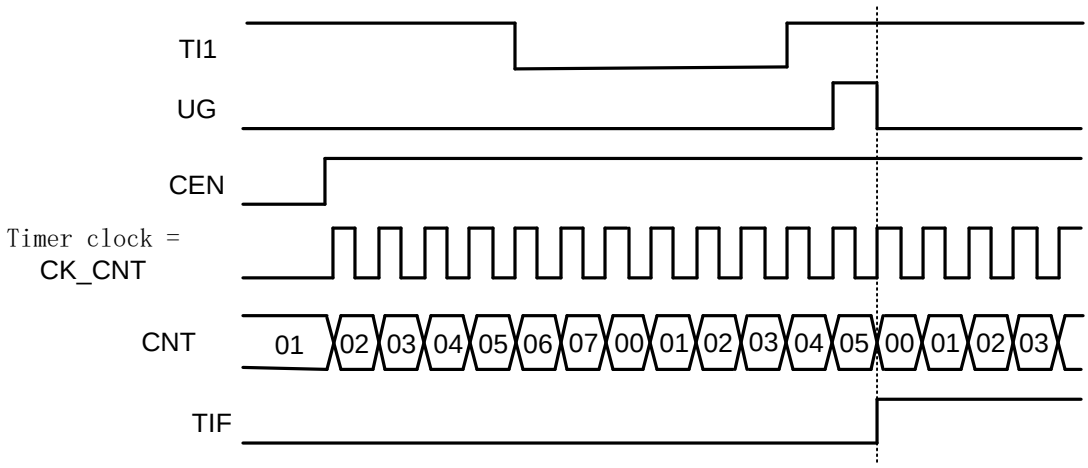


图30-33复位模式下的时序

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

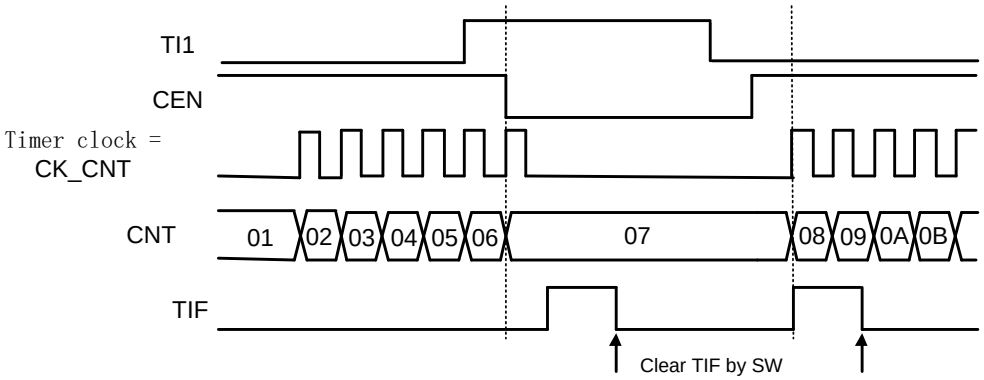


图30-34门控模式下的时序

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

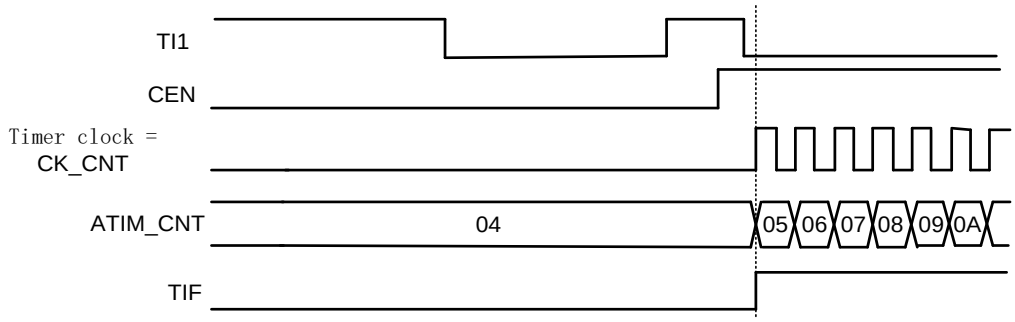


图30-35触发器模式下的时序

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

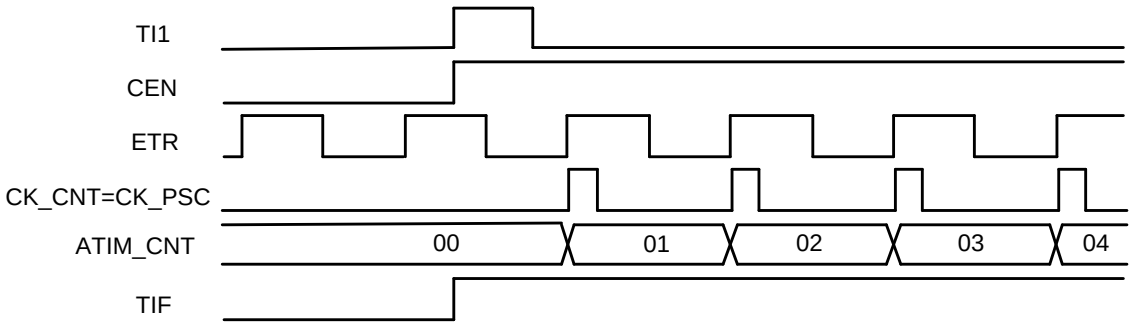


图30-36外部时钟模式2＋触发模式下的时序

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
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

地址	名称	符号
GPTIM0(模块基地址：0x40014C00)		
0x00000000	GPTIM0 控制寄存器 1 （GPTIM0 Control Register1）	GPTIM0_CR1
0x00000004	GPTIM0 控制寄存器 2 （GPTIM0 Control Register2）	GPTIM0_CR2
0x00000008	GPTIM0 从机模式控制寄存器 （GPTIM0 Slave Mode Control Register）	GPTIM0_SMCR
0x0000000C	GPTIM0 DMA 和中断使能寄存器 （GPTIM0 DMA and Interrupt Enable Register）	GPTIM0_DIER
0x00000010	GPTIM0 状态寄存器 （GPTIM0 Interrupt Status Register）	GPTIM0_SR
0x00000014	GPTIM0 事件产生寄存器 （GPTIM0 Event Generation Register）	GPTIM0_EGR
0x00000018	GPTIM0 捕捉/比较模式寄存器 1 （GPTIM0 Capture/Compare Mode Register1）	GPTIM0_CCMR1
0x0000001C	GPTIM0 捕捉/比较模式寄存器 2 （GPTIM0 Capture/Compare Mode Register2）	GPTIM0_CCMR2
0x00000020	GPTIM0 捕捉/比较使能寄存器 （GPTIM0 Capture/Compare Enable Register）	GPTIM0_CCER
0x00000024	GPTIM0 计数器寄存器 （GPTIM0 Counter Register）	GPTIM0_CNT
0x00000028	GPTIM0 预分频寄存器 （GPTIM0 Prescaler Register）	GPTIM0_PSC
0x0000002C	GPTIM0 自动重载寄存器 （GPTIM0 Auto-Reload Register）	GPTIM0_ARR
0x00000034	GPTIM0 捕捉/比较寄存器 1 （GPTIM0 Capture/Compare Register1）	GPTIM0_CCR1
0x00000038	GPTIM0 捕捉/比较寄存器 2 （GPTIM0 Capture/Compare Register2）	GPTIM0_CCR2
0x0000003C	GPTIM0 捕捉/比较寄存器 3 （GPTIM0 Capture/Compare Register3）	GPTIM0_CCR3
0x00000040	GPTIM0 捕捉/比较寄存器 4 （GPTIM0 Capture/Compare Register4）	GPTIM0_CCR4
0x00000048	GPTIM0 DMA 控制寄存器 （GPTIM0 DMA Control Register）	GPTIM0_DCR
0x0000004C	GPTIM0 DMA 访问寄存器 （GPTIM0 DMA access Register）	GPTIM0_DMAR
0x00000060	GPTIM0 ITR 选择寄存器 （GPTIM0 Internal Trigger Select Register）	GPTIM0_ITRSEL
GPTIM1(模块基地址：0x40016400)		
0x00000000	GPTIM1 控制寄存器 1 （GPTIM1 Control Register1）	GPTIM1_CR1
0x00000004	GPTIM1 控制寄存器 2 （GPTIM1 Control Register2）	GPTIM1_CR2
0x00000008	GPTIM1 从机模式控制寄存器 （GPTIM1 Slave Mode Control Register）	GPTIM1_SMCR
0x0000000C	GPTIM1 DMA 和中断使能寄存器 （GPTIM1 DMA and Interrupt Enable Register）	GPTIM1_DIER

地址	名称	符号
0x00000010	GPTIM1 状态寄存器 （GPTIM1 Interrupt Status Register）	GPTIM1_SR
0x00000014	GPTIM1 事件产生寄存器 （GPTIM1 Event Generation Register）	GPTIM1_EGR
0x00000018	GPTIM1 捕捉/比较模式寄存器 1 （GPTIM1 Capture/Compare Mode Register1）	GPTIM1_CCMR1
0x0000001C	GPTIM1 捕捉/比较模式寄存器 2 （GPTIM1 Capture/Compare Mode Register2）	GPTIM1_CCMR2
0x00000020	GPTIM1 捕捉/比较使能寄存器 （GPTIM1 Capture/Compare Enable Register）	GPTIM1_CCER
0x00000024	GPTIM1 计数器寄存器 （GPTIM1 Counter Register）	GPTIM1_CNT
0x00000028	GPTIM1 预分频寄存器 （GPTIM1 Prescaler Register）	GPTIM1_PSC
0x0000002C	GPTIM1 自动重载寄存器 （GPTIM1 Auto-Reload Register）	GPTIM1_ARR
0x00000034	GPTIM1 捕捉/比较寄存器 1 （GPTIM1 Capture/Compare Register1）	GPTIM1_CCR1
0x00000038	GPTIM1 捕捉/比较寄存器 2 （GPTIM1 Capture/Compare Register2）	GPTIM1_CCR2
0x0000003C	GPTIM1 捕捉/比较寄存器 3 （GPTIM1 Capture/Compare Register3）	GPTIM1_CCR3
0x00000040	GPTIM1 捕捉/比较寄存器 4 （GPTIM1 Capture/Compare Register4）	GPTIM1_CCR4
0x00000048	GPTIM1 DMA 控制寄存器 （GPTIM1 DMA Control Register）	GPTIM1_DCR
0x0000004C	GPTIM1 DMA 访问寄存器 （GPTIM1 DMA access Register）	GPTIM1_DMAR
0x00000060	GPTIM1 ITR 选择寄存器 （GPTIM1 Internal Trigger Select Register）	GPTIM1_ITRSEL
GPTIM2(模块基地址：0x40018000)		
0x00000000	GPTIM2 控制寄存器 1 （GPTIM2 Control Register1）	GPTIM2_CR1
0x00000004	GPTIM2 控制寄存器 2 （GPTIM2 Control Register2）	GPTIM2_CR2
0x00000008	GPTIM2 从机模式控制寄存器 （GPTIM2 Slave Mode Control Register）	GPTIM2_SMCR
0x0000000C	GPTIM2 DMA 和中断使能寄存器 （GPTIM2 DMA and Interrupt Enable Register）	GPTIM2_DIER
0x00000010	GPTIM2 状态寄存器 （GPTIM2 Interrupt Status Register）	GPTIM2_SR
0x00000014	GPTIM2 事件产生寄存器 （GPTIM2 Event Generation Register）	GPTIM2_EGR
0x00000018	GPTIM2 捕捉/比较模式寄存器 1 （GPTIM2 Capture/Compare Mode Register1）	GPTIM2_CCMR1
0x0000001C	GPTIM2 捕捉/比较模式寄存器 2 （GPTIM2 Capture/Compare Mode Register2）	GPTIM2_CCMR2
0x00000020	GPTIM2 捕捉/比较使能寄存器 （GPTIM2 Capture/Compare Enable Register）	GPTIM2_CCER
0x00000024	GPTIM2 计数器寄存器 （GPTIM2 Counter Register）	GPTIM2_CNT

地址	名称	符号
0x00000028	GPTIM2 预分频寄存器 （GPTIM2 Prescaler Register）	GPTIM2_PSC
0x0000002C	GPTIM2 自动重载寄存器 （GPTIM2 Auto-Reload Register）	GPTIM2_ARR
0x00000034	GPTIM2 捕捉/比较寄存器 1 （GPTIM2 Capture/Compare Register1）	GPTIM2_CCR1
0x00000038	GPTIM2 捕捉/比较寄存器 2 （GPTIM2 Capture/Compare Register2）	GPTIM2_CCR2
0x0000003C	GPTIM2 捕捉/比较寄存器 3 （GPTIM2 Capture/Compare Register3）	GPTIM2_CCR3
0x00000040	GPTIM2 捕捉/比较寄存器 4 （GPTIM2 Capture/Compare Register4）	GPTIM2_CCR4
0x00000048	GPTIM2 DMA 控制寄存器 （GPTIM2 DMA Control Register）	GPTIM2_DCR
0x0000004C	GPTIM2 DMA 访问寄存器 （GPTIM2 DMA access Register）	GPTIM2_DMAR
0x00000060	GPTIM2 ITR 选择寄存器 （GPTIM2 Internal Trigger Select Register）	GPTIM2_ITRSEL

30.5.1 GPTIMx 控制寄存器 1（GPTIMx_CR1）

名称	GPTIMx_CR1(x=0,1)							
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	-						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

位号	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）(Counter 580lock Divider) 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(Auto-Reload Preload Enable) 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(Counter Mode Selection) 00：边沿对齐模式

位号	助记符	功能描述
		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(counter Direction) 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(One Pulse Mode) 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(Update Request Selection) 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(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 - 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(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

30.5.2 GPTIMx 控制寄存器 2（GPTIMx_CR2）

名称	GPTIMx_CR2(x=0,1)							
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	TI1S	MMS			CCDS	-		
位权限	R/W-0	R/W-000			R/W-0	U-0		

位号	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(Timer Input 1 Selection) 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(Master Mode Selection) 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(Capture/Compare DMA Selection) 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（GPTIMx_SMCR）

名称	GPTIMx_SMCR(x=0,1)							
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	ETP	ECE	ETPS		ETF			
位权限	R/W-0	R/W-0	R/W-00		R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSM	TS			-	SMS		
位权限	R/W-0	R/W-000			U-0	R/W-000		

位号	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(External Trigger Polarity) 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(External Clock Enable)

位号	助记符	功能描述
		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(External Trigger Prescaler) 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(External Trigger Filter) 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(Master Slave Mode) 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(Trigger Source) 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(Slave Mode Selection) 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

位号	助记符	功能描述
		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（GPTIMx_DIER）

名称	GPTIMx_DIER(x=0,1)							
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	-				CC4BURSTEN	CC3BURSTEN	CC2BURSTEN	CC1BURSTEN
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	-	TDE	-	CC4DE	CC3DE	CC2DE	CC1DE	UDE
位权限	U-0	R/W-0	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	-	TIE	-			CC2IE	CC1IE	UIE
位权限	U-0	R/W-0	U-0			R/W-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	**CC4BURSTEN**	捕捉比较通道 4 的 DMA 模式配置 (CC4 Burst Enable) 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 (CC3 Burst Enable) 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 (CC2 Burst Enable) 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 (CC1 Burst Enable) 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(Triggered DMA Enable) 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）

位号	助记符	功能描述
13	-	RFU：未实现，读为 0
12	**CC4DE**	捕捉比较通道 4 的 DMA 请求使能 (CC4 DMA Enable) 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 (CC3 DMA Enable) 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 (CC2 DMA Enable) 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 (CC1 DMA Enable) 0：禁止 CC1 DMA 请求 1：允许 CC1 DMA 请求
8	**UDE**	Update Event DMA 请求使能 (Update event DMA Enable) 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 (Trigger event Interrupt Enable) 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	**CC3IE**	捕捉/比较通道 4 中断使能 (CC4 Interrupt Enable) 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 (CC3 Interrupt Enable) 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 (CC2 Interrupt Enable) 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 (CC1 Interrupt Enable) 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 (Update event Interrupt Enable) 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（GPTIMx_ISR）

名称	GPTIMx_ISR(x=0,1)							
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	-			CC4OF	CC3OF	CC2OF	CC1OF	-
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	TIF	-	CC4IF	CC3IF	CC2IF	CC1IF	UIF
位权限	U-0	R/W-0	U-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	**CC4OF**	捕捉/比较通道 4 的 Overcapture 中断(Over-Capture Interrupt Flag for CC4, write 1 to clear) 参考 CC1OF
11	**CC3OF**	捕捉/比较通道 3 的 Overcapture 中断(Over-Capture Interrupt Flag for CC3, write 1 to clear) 参考 CC1OF
10	**CC2OF**	捕捉/比较通道 2 的 Overcapture 中断(Over-Capture Interrupt Flag for CC2, write 1 to clear) 参考 CC1OF
9	**CC1OF**	捕捉/比较通道 1 的 Overcapture 中断(Over-Capture Interrupt Flag for CC1, write 1 to clear) 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 (Trigger event Interrupt Flag, write 1 to clear)
5	**-**	RFU：未实现，读为 0
4	**CC4IF**	捕捉/比较通道 4 中断标志 (CC4 Interrupt Flag, write 1 to clear) 参考 CC1IF
3	**CC3IF**	捕捉/比较通道 3 中断标志 (CC3 Interrupt Flag, write 1 to clear) 参考 CC3IF
2	**CC2IF**	捕捉/比较通道 2 中断标志 (CC2 Interrupt Flag, write 1 to clear) 参考 CC2IF
1	**CC1IF**	捕捉/比较通道 1 中断标志 (CC1 Interrupt Flag, write 1 to clear) 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。(Update event Interrupt Flag,write 1 to clear) 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（GPTIMx_EGR）

名称	GPTIMx_EGR(x=0,1)
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	-	TG	-			CC2G	CC1G	UG
位权限	U-0	W-0	U-0			W-0	W-0	W-0

位号	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 (Trigger Generate)
5:3	-	RFU：未实现，读为 0
2	**CC2G**	捕捉/比较通道 2 软件触发，参考 CC1G (CC2 Generate)
1	**CC1G**	捕捉/比较通道 1 软件触发 (CC1 Generate) 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 (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。

30.5.7 GPTIMx 捕捉/比较模式寄存器 1（GPTIMx_CCMR1）

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(x=0,1)							
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	OC2CE	OC2M			OC2PE	OC2FE	CC2S	
	IC2F				IC2PSC		CC2S	
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	OC1CE	OC1M			OC1PE	OC1FE	CC1S	
	IC1F				IC1PSC		CC1S	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	

输出比较模式

位号	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 (OC2 Clear Enable)
14:12	**OC2M**	输出比较 2 模式配置，参考 OC1M (OC2 Mode)
11	**OC2PE**	输出比较 2 预装载使能，参考 OC1PE (OC2 Preload Enable)
10	**OC2FE**	输出比较 2 快速使能，参考 OC1FE (OC2 Fast Enable)
9:8	**CC2S**	捕捉/比较 2 通道选择 (CC2 channel Selection) 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(OC2 Clear Enable) 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 (OC1 Mode) 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 (OC1 Preload Enable) 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 (OC1 Fast Enable) 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 (CC1 channel Selection) 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

位号	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 (IC2 Filter)
11:10	**IC2PSC**	输入捕捉 2 预分频 (IC2 Prescaler)
9:8	**CC2S**	捕捉/比较 2 通道选择 (Capture/Compare2 channel Selection) 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 (IC1 Filter) 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 (IC1 Prescaler) 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 (Capture/Compare1 channel Selection) 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（GPTIMx_CCMR2）

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(x=0,1)							
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	OC4CE	OC4M			OC4PE	OC4FE	CC4S	
	IC2F				IC2PSC		CC4S	
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		CC3S	
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

位号	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 OC3CE (OC4 Clear Enable)
14:12	**OC4M**	输出比较 4 模式配置，参考 OC3M (OC4 Mode)
11	**OC4PE**	输出比较 4 预装载使能，参考 OC3PE (OC4 Preload Enable)
10	**OC4FE**	输出比较 4 快速使能，参考 OC3FE (OC4 Fast Enable)
9:8	**CC4S**	捕捉/比较 4 通道选择 (CC4 channel Selection) 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(OC3 Clear Enable) 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 (OC3 Mode) 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 (OC3 Preload Enable) 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 (OC3 Fast Enable) 0：关闭快速使能，trigger 输入不会影响比较输出

位号	助记符	功能描述
		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 (CC3 channel Selection) 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

位号	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 (IC4 Filter)
11:10	**IC4PSC**	输入捕捉 4 预分频 (IC4 Prescaler)
9:8	**CC4S**	捕捉/比较 4 通道选择 (CC4 channel Selection) 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 3 滤波 (IC3 Filter) 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(IC3 Prescaler) 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 (CC3 channel Selection) 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

位号	助记符	功能描述
		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（GPTIMx_CCER）

名称	GPTIMx_CCER(x=0,1)							
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	-		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		R/W-0	R/W-0	U-0		R/W-0	R/W-0

位号	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 (CC4 Polarity)
12	**CC4E**	捕捉/比较 4 输出使能，参考 CC1E (CC4 output Enable)
11:10	**-**	RFU：未实现，读为 0
9	**CC3P**	捕捉/比较 3 输出极性，参考 CC1P (CC3 Polarity)
8	**CC3E**	捕捉/比较 3 输出使能，参考 CC1E (CC3 output Enable)
7:6	**-**	RFU：未实现，读为 0
5	**CC2P**	捕捉/比较 2 输出极性，参考 CC1P (CC2 Polarity)
4	**CC2E**	捕捉/比较 2 输出使能，参考 CC1E (CC2 output Enable)
3:2	**-**	RFU：未实现，读为 0
1	**CC1P**	捕捉/比较 1 输出极性 (CC1 Polarity) **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 output Enable) **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

位号	助记符	功能描述
		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（GPTIMx_CNT）

名称	GPTIMx_CNT(x=0,1)							
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	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
31:16	-	RFU：未实现，读为 0
15:0	CNT	计数器值(Counter)

30.5.11 GPTIMx 预分频寄存器（GPTIMx_PSC）

名称	GPTIMx_PSC(x=0,1)							
offset	0x00000028 + x*0x400							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	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
31:16	-	RFU：未实现，读为 0
15:0	**PSC**	计数器时钟（CK_CNT）预分频值(Counter Clock Prescaler) $f_{CK_CNT}=f_{CK_PSC}/(PSC[15:0]+1)$ 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 自动重载寄存器（GPTIMx_ARR）

名称	GPTIMx_ARR(x=0,1)							
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	ARR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-1111 1111							

位号	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(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

30.5.13 GPTIMx 捕捉/比较寄存器 1（GPTIMx_CCR1）

名称	GPTIMx_CCR1(x=0,1)							
Offset	0x00000034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	助记符	功能描述

位号	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(Capture/Compare channel 1 Register) **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（GPTIMx_CCR2）

名称	GPTIMx_CCR2(x=0,1)							
Offset	0x00000038							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	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(Capture/Compare channel 2 Register) **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

30.5.15 GPTIMx 捕捉/比较寄存器 3（GPTIMx_CCR3）

名称	GPTIMx_CCR3(x=0,1)							
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	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	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(Capture/Compare channel 3 Register) **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（GPTIMx_CCR4）

名称	**GPTIMx_CCR4(x=0,1)**							
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	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	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(Capture/Compare channel 4 Register) **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（GPTIMx_DCR）

名称	**GPTIMx_DCR(x=0,1)**
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	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			DBL				
位权限	U-0			R/W-0 0000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			DBA				
位权限	U-0			R/W-0 0000				

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12:8	**DBL**	DMA Burst 长度 (DMA Burst Length) 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 00010：长度=3 00011：长度=4 00100：长度=5 00101：长度=6 00110：长度=7 00111：长度=8 01000：长度=9 01001：长度=10 01010：长度=11 01011：长度=12 01100：长度=13 01101：长度=14 01110：长度=15 01111：长度=16 10000：长度=17 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 (DMA Burst offset Address) 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 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（GPTIMx_DMAR）

名称	GPTIMx_DMAR(x=0,1)							
Offset	0x0000004C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DMAR[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	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（GPTIMx_ITRSEL）

名称	GPTIMx_ITRSEL(x=0,1)							
Offset	0x00000060							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	

位号	助记符	功能描述
31:8	-	RFU，未实现，读为 0
7:6	ITR3SEL	ITR 输入信号选择(Internal Trigger Source Selection) 内部触发信号（ITRx）的捕捉 详情参见 30.4.4 内部触发信号（ITRx）的捕捉
5:4	ITR2SEL	
3:2	ITR1SEL	
1:0	ITR0SEL	

31 32 位基本定时器（BSTIM32）

31.1 概述

FM33LG0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。

31.2 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

31.3 结构框图

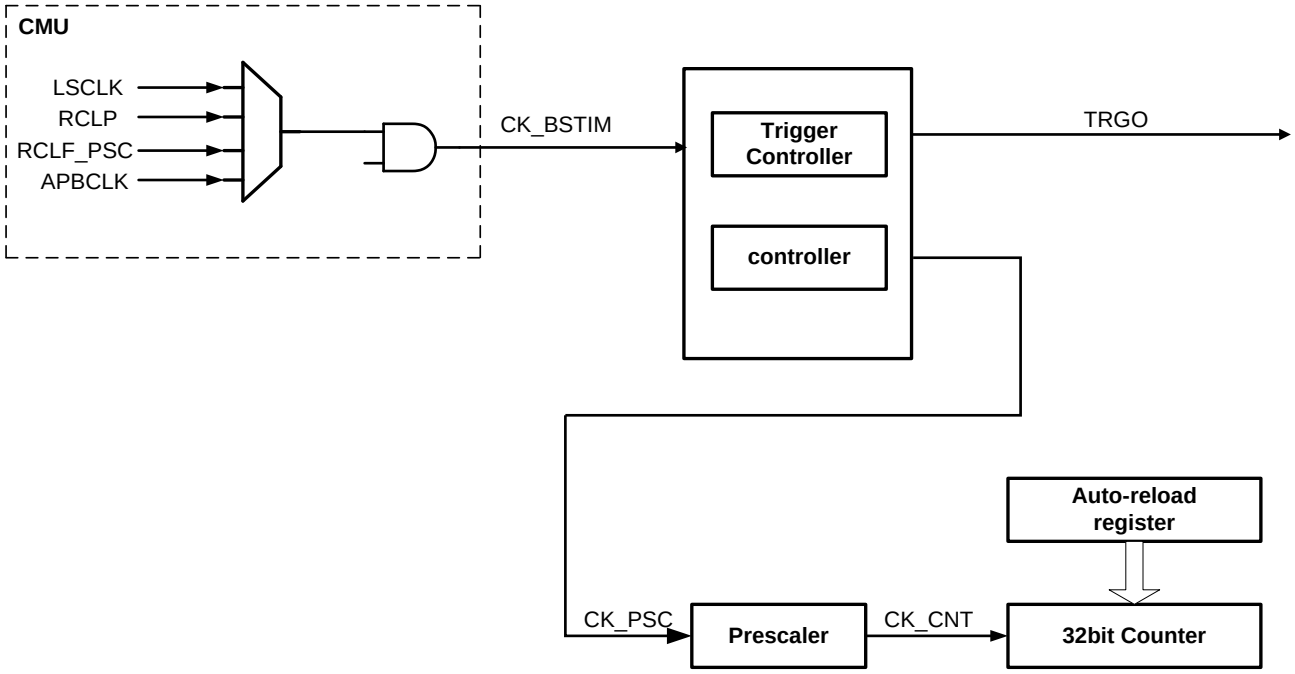


图 31-1 32 位基本定时器结构框图

31.4 功能描述

31.4.1 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。

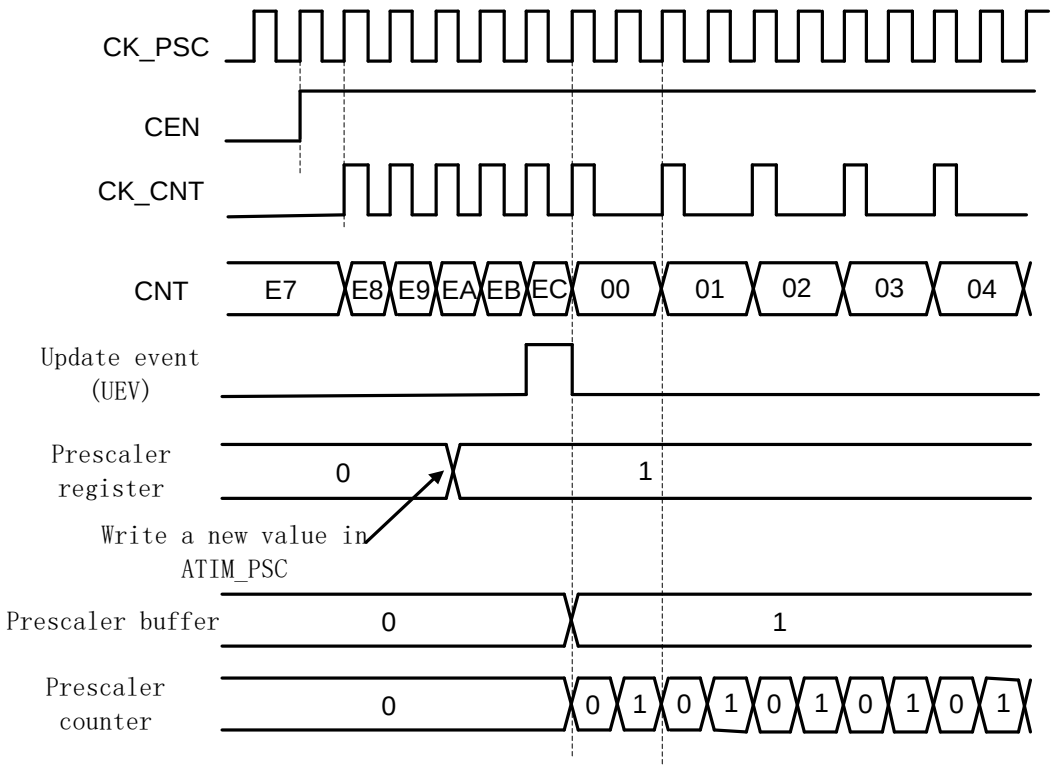


图 31-2 预分频从 1 变为 2 的波形

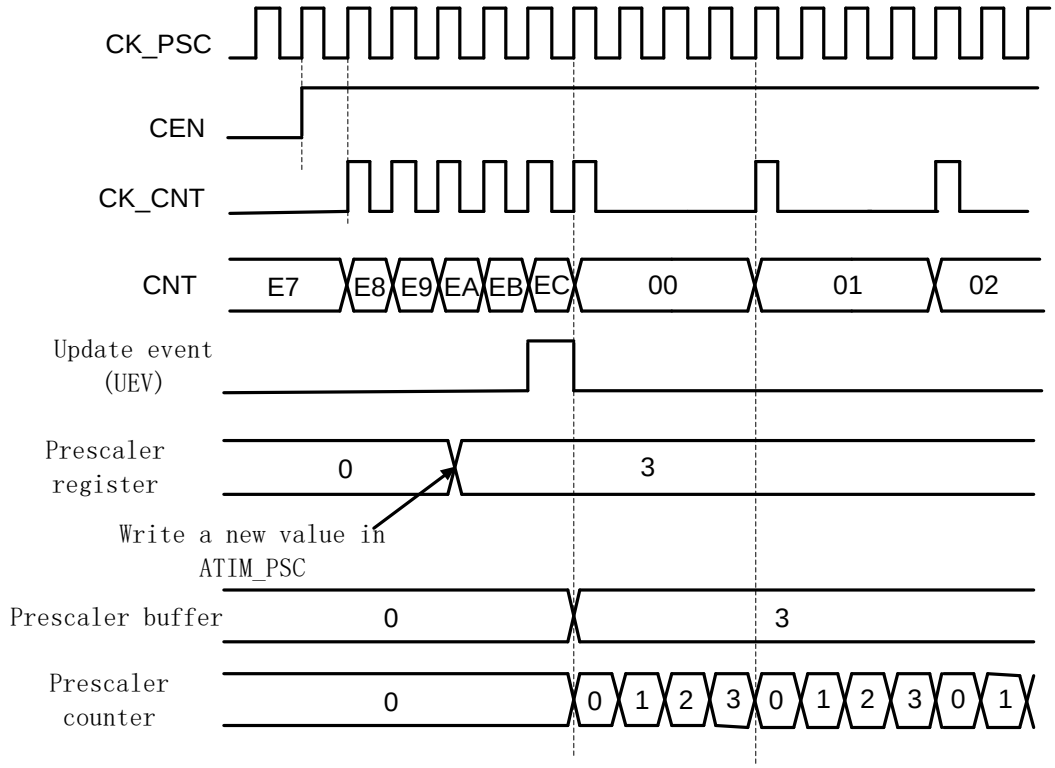


图 31-3 预分频从 1 变为 4 的波形

31.4.2 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

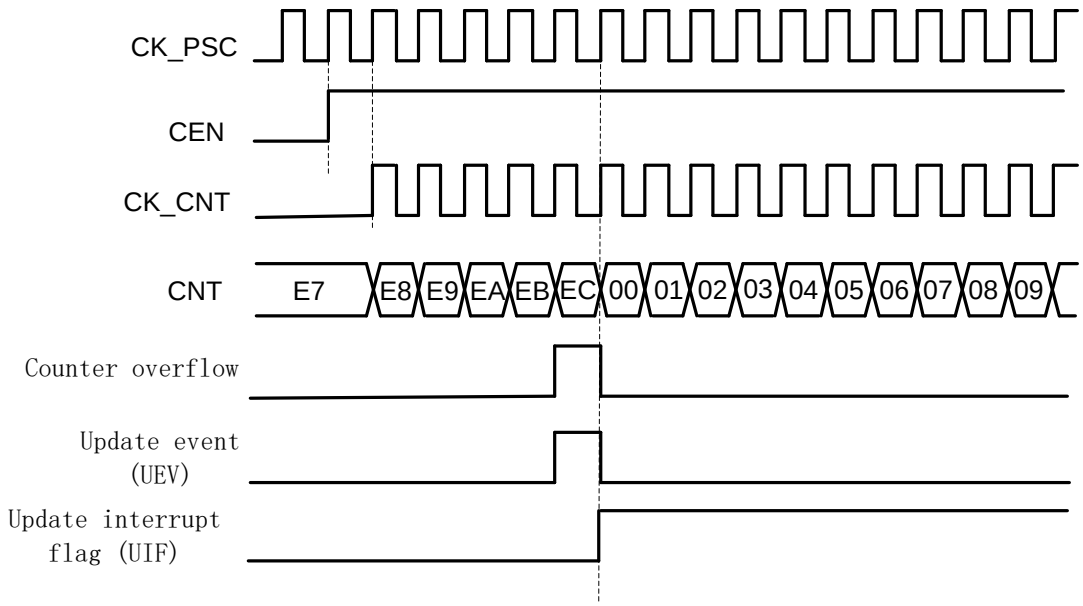


图 31-4 向上计数波形，内部时钟不分频

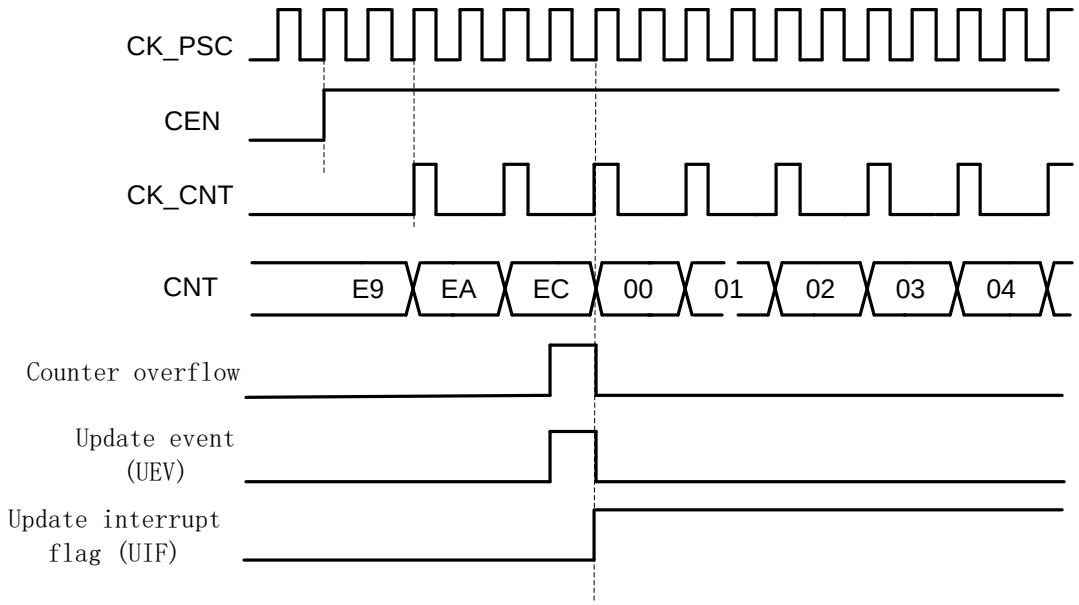


图 31-5 向上计数波形，内部时钟 2 分频

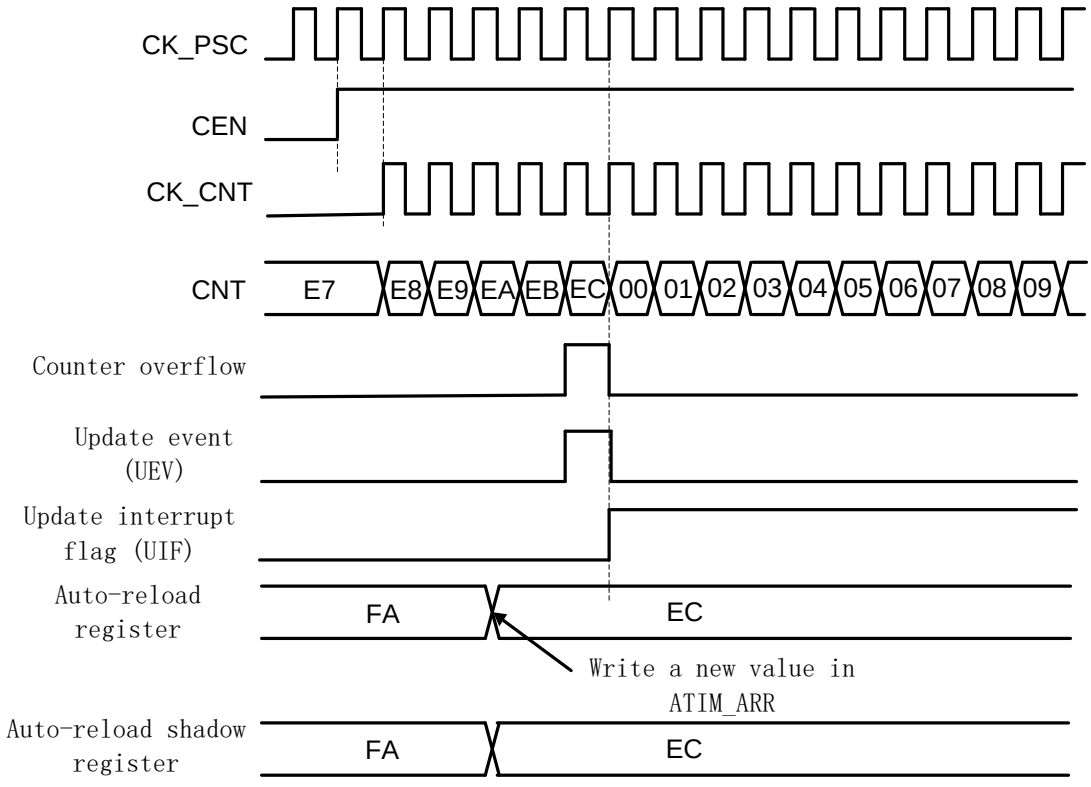


图 31-6ARPE=0（ARR 没有预装载）时的更新事件

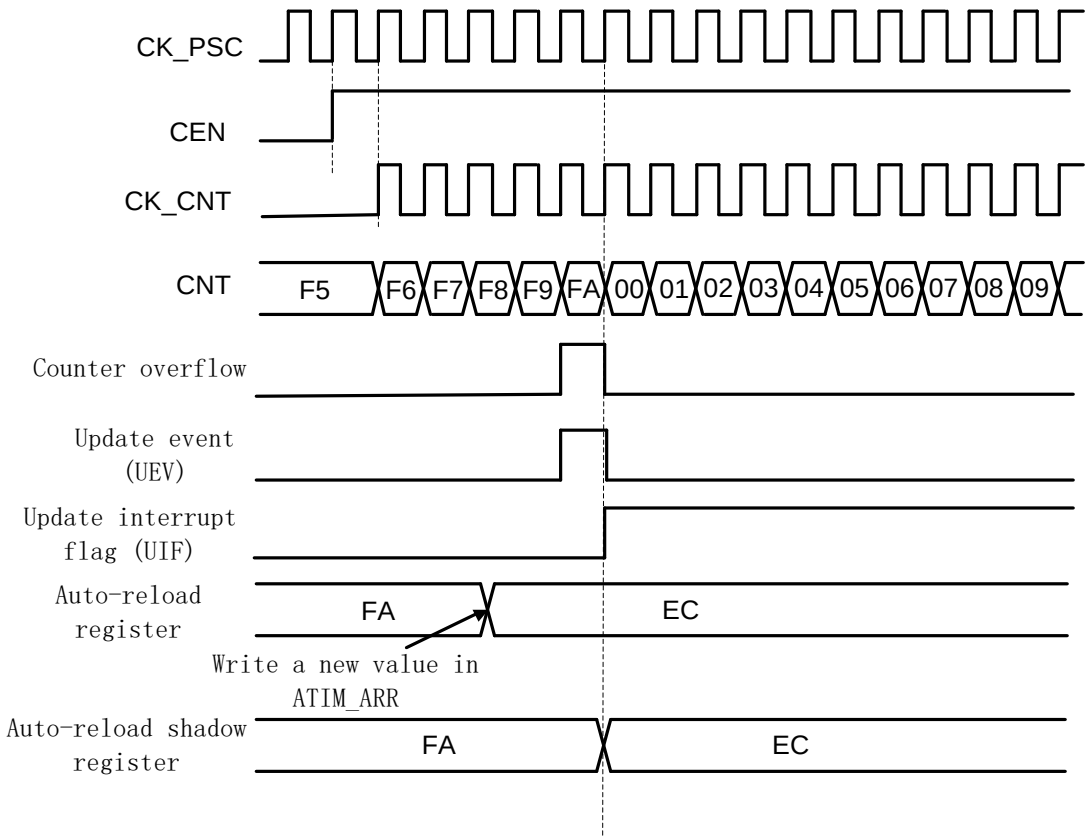


图 31-7 ARPE=1（ARR 预装载）时的更新事件

31.4.3 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。

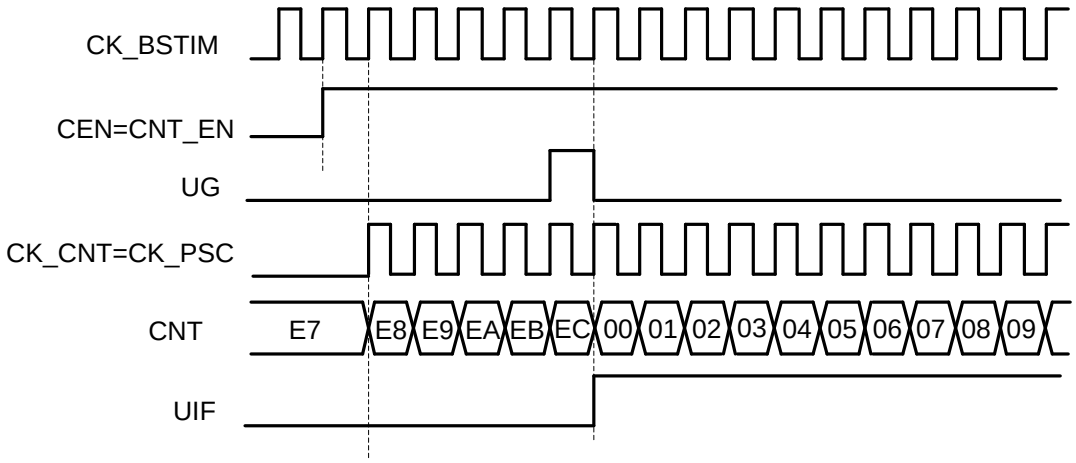


图 31-8 内部时钟源模式，时钟分频因子为 1

31.4.1 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。

31.5 寄存器

offset 地址	名称	符号
BSTIM32(模块基地址：0x40016000)		
0x00000000	BSTIM 控制寄存器 1 （BSTIM Control Register1）	BSTIM_CR1
0x00000004	BSTIM 控制寄存器 2 （BSTIM Control Register2）	BSTIM_CR2
0x0000000C	BSTIM 中断使能寄存器 （BSTIM Interrupt Enable Register）	BSTIM_IER
0x00000010	BSTIM 中断标志寄存器 （BSTIM Interrupt Status Register）	BSTIM_ISR
0x00000014	BSTIM 事件产生寄存器 （BSTIM Event Generation Register）	BSTIM_EGR
0x00000024	BSTIM 计数器寄存器 （BSTIM Counter Register）	BSTIM_CNT
0x00000028	BSTIM 预分频寄存器 （BSTIM Prescaler Register）	BSTIM_PSC
0x0000002C	BSTIM 自动重载寄存器 （BSTIM Auto-Reload Register）	BSTIM_ARR

31.5.1 BSTIM 控制寄存器 1（BSTIM_CR1）

名称	BSTIM_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	-							
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 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

位号	助记符	功能描述
		- 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

31.5.2 BSTIM 控制寄存器 2（BSTIM_CR2）

名称	BSTIM_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	-	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

31.5.3 BSTIM 中断使能寄存器（BSTIM_IER）

名称	BSTIM_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	-							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

31.5.4 BSTIM 中断标志寄存器（BSTIM_ISR）

名称	BSTIM_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	-							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 -URS=0 且 UDIS=0 的情况下，触发事件初始化计数器

31.5.5 BSTIM 事件产生寄存器（BSTIM_EGR）

名称	BSTIM_EGR							
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	-							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。*

31.5.6 BSTIM 计数器寄存器（BSTIM_CNT）

名称	BSTIM_CNT							
Offset	0x00000024							
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)

31.5.7 BSTIM 预分频寄存器（BSTIM_PSC）

名称	BSTIM_PSC							
Offset	0x00000028							
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

31.5.8 BSTIM 自动重载寄存器（BSTIM_ARR）

名称	BSTIM_ARR							
Offset	0x0000002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ARR[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ARR[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-1111 1111							

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

32 16 位基本定时器（BSTIM16）

32.1 概述

FM33LG0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。

32.2 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

32.3 结构框图

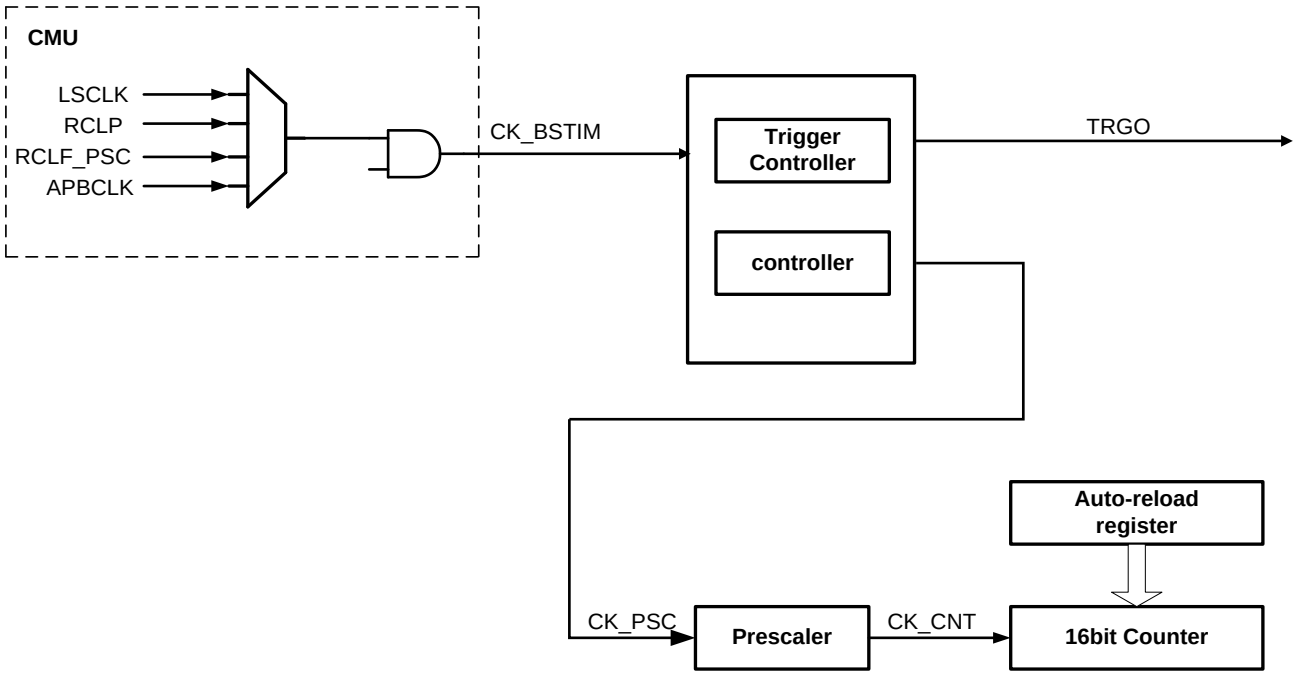


图 32-1 16 位基本定时器结构框图

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。

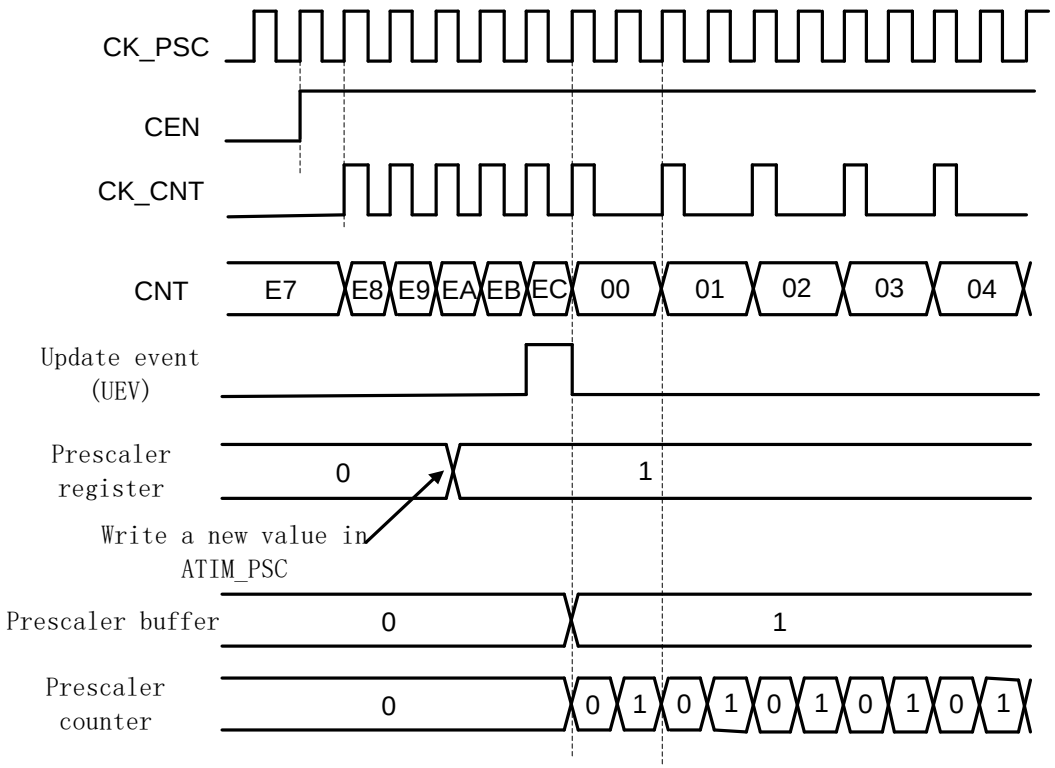


图 32-2 预分频从 1 变为 2 的波形

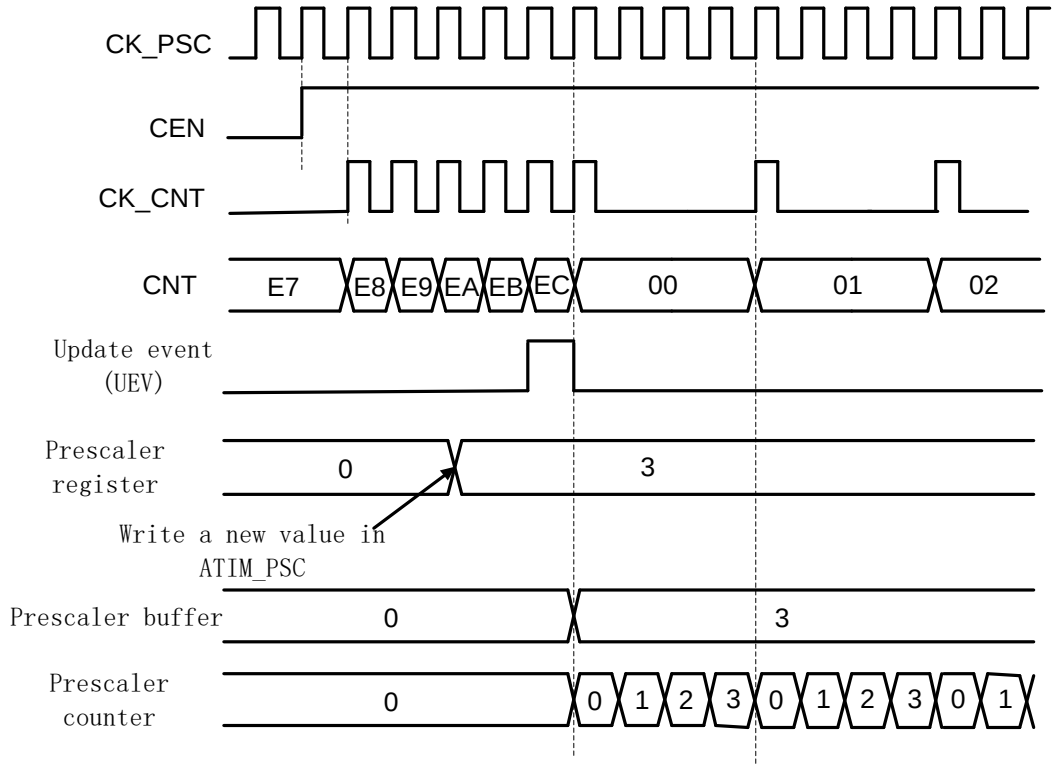


图 32-3 预分频从 1 变为 4 的波形

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

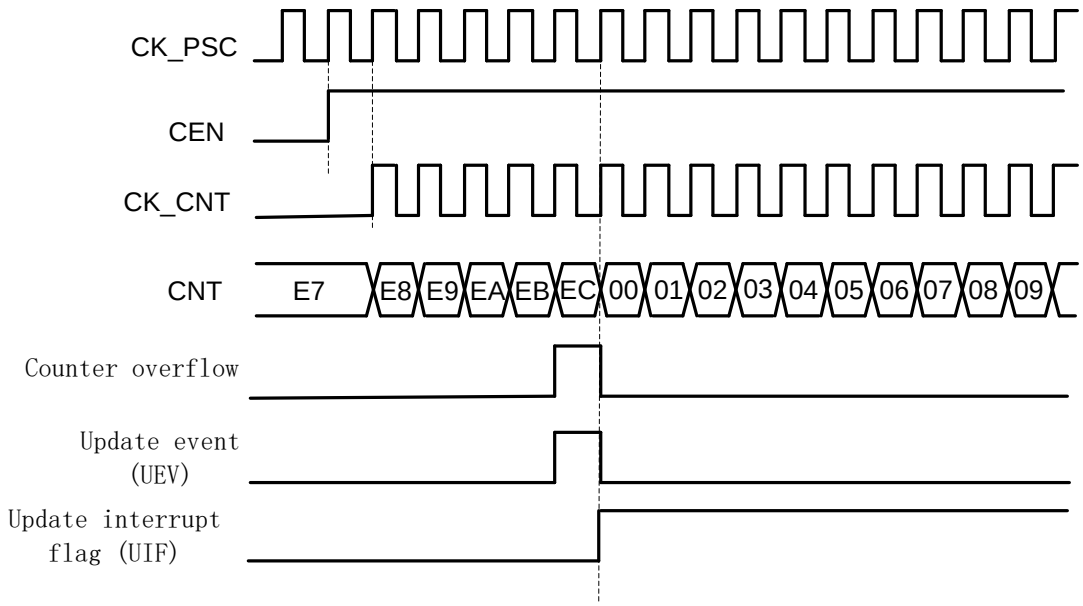


图 32-4 向上计数波形，内部时钟不分频

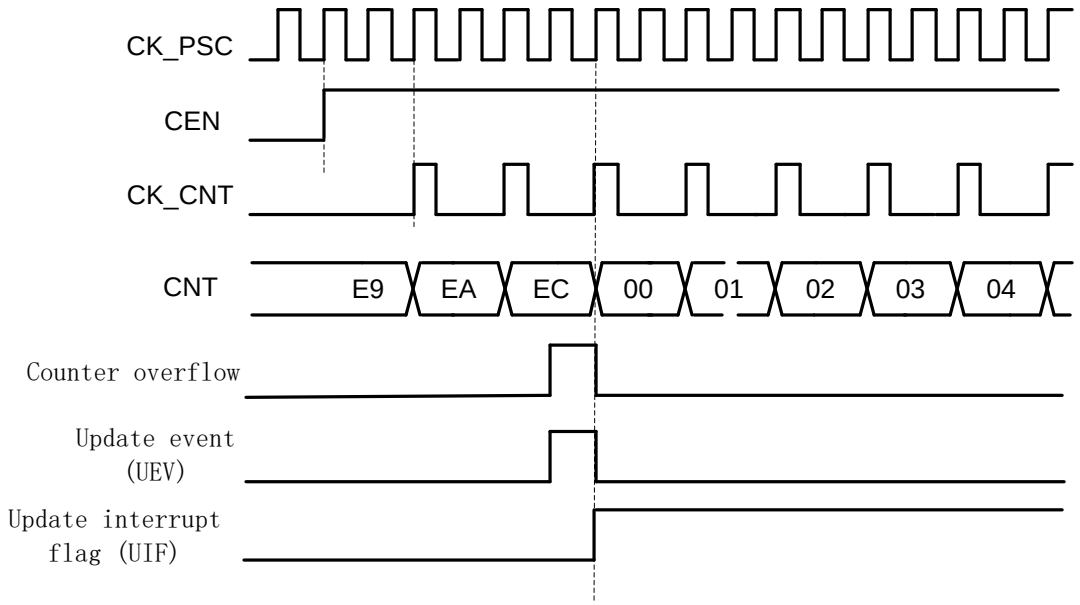


图 32-5 向上计数波形，内部时钟 2 分频

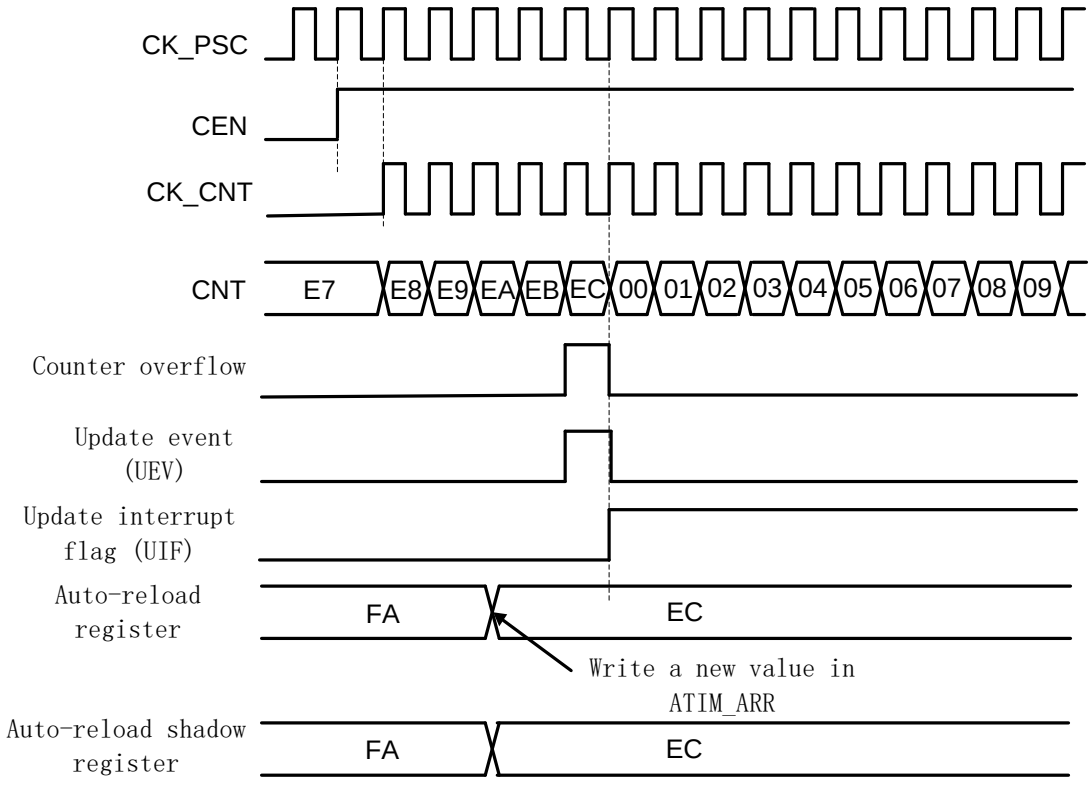


图 32-6 ARPE=0（ARR 没有预装载）时的更新事件

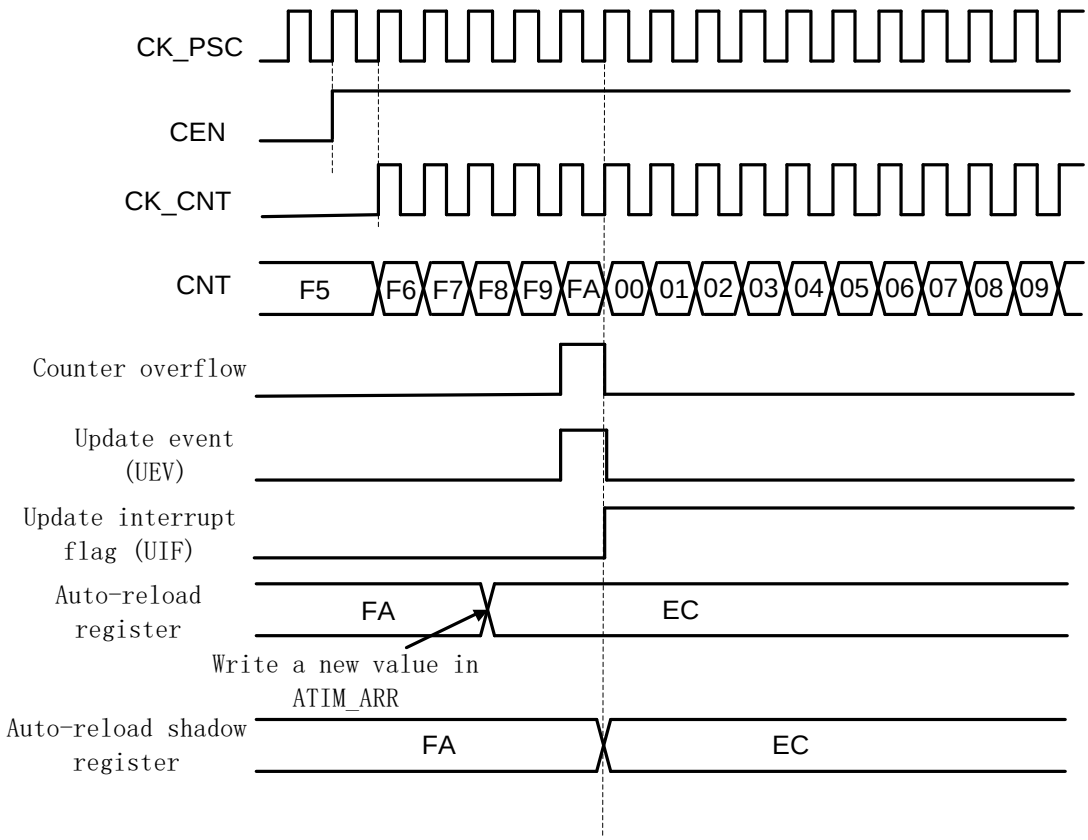


图 32-7 ARPE=1（ARR 预装载）时的更新事件

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。

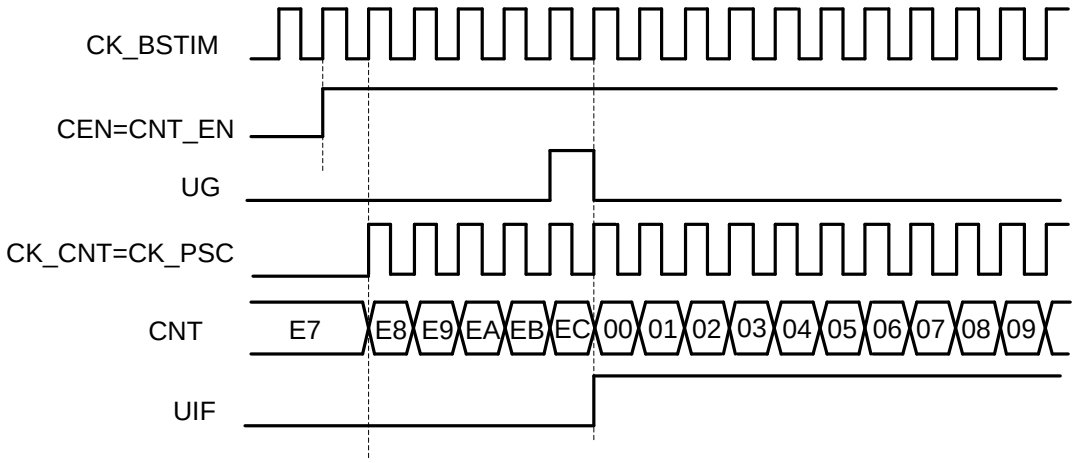


图 32-8 内部时钟源模式，时钟分频因子为 1

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 DCU 模块的 DBG_TIMx_STOP寄存器定义。

32.5 寄存器

offset 地址	名称	符号
BSTIM16(模块基地址：0x40018C00)		
0x00000000	BSTIM 控制寄存器 1 （BSTIM Control Register1）	BSTIM_CR1
0x00000004	BSTIM 控制寄存器 2 （BSTIM Control Register2）	BSTIM_CR2
0x0000000C	BSTIM 中断使能寄存器 （BSTIM Interrupt Enable Register）	BSTIM_IER
0x00000010	BSTIM 中断标志寄存器 （BSTIM Interrupt Status Register）	BSTIM_ISR
0x00000014	BSTIM 事件产生寄存器 （BSTIM Event Generation Register）	BSTIM_EGR
0x00000024	BSTIM 计数器寄存器 （BSTIM Counter Register）	BSTIM_CNT
0x00000028	BSTIM 预分频寄存器 （BSTIM Prescaler Register）	BSTIM_PSC
0x0000002C	BSTIM 自动重载寄存器 （BSTIM Auto-Reload Register）	BSTIM_ARR

32.5.1 BSTIM 控制寄存器 1（BSTIM_CR1）

名称	BSTIM_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	-							
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 1：Update Event 发生时计数器停止（自动清零 CEN）

位号	助记符	功能描述
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 BSTIM 控制寄存器 2（BSTIM_CR2）

名称	BSTIM_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	-	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 BSTIM 中断使能寄存器（BSTIM_IER）

名称	BSTIM_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	-							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 BSTIM 中断标志寄存器（BSTIM_ISR）

名称	BSTIM_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	-							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 BSTIM 事件产生寄存器（BSTIM_EGR）

名称	BSTIM_EGR							
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	-							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。*

32.5.1 BSTIM 计数器寄存器（BSTIM_CNT）

名称	BSTIM_CNT							
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	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
31:16	-	RFU：未实现，读为 0
15:0	**CNT**	计数器值(Counter)

32.5.1 BSTIM 预分频寄存器（BSTIM_PSC）

名称	BSTIM_PSC							
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	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
31:16	-	RFU：未实现，读为 0
15:0	**PSC**	计数器时钟（CK_CNT）预分频值(Counter Clock Prescaler) $f_{CK_CNT}=f_{CK_PSC}/(PSC[15:0]+1)$ 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

32.5.1 BSTIM 自动重载寄存器（BSTIM_ARR）

名称	BSTIM_ARR							
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	ARR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述

位号	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(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 32 位低功耗定时器（LPTIM32）

33.1 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、RCLF_PSC
  - 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

33.2 结构框图

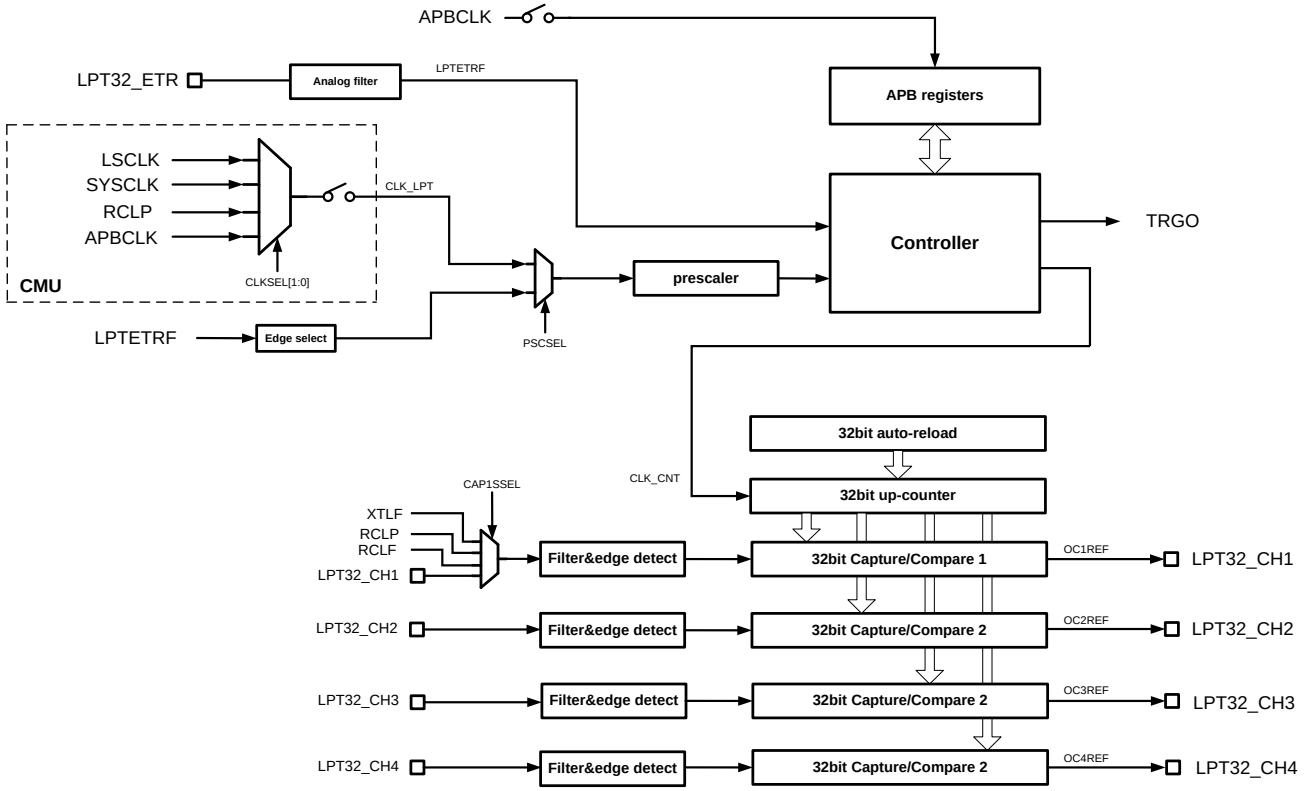


图 33-1 LPTIM32 结构框图

33.3 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14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。

33.4 相关引脚

功能	引脚映射	
	LQFP100	LQFP80
LPT32_ETR	PA13	
LPT32_CH1	PA11	
LPT32_CH2	PA12	
LPT32_CH3		
LPT32_CH4		

表 33-1 LPTIM32 引脚映射

33.5 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。

33.5.1 普通定时器

当LPTCFG.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

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（LPTARR）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（LPTARR）后回到0，并自动停止，产生溢出中断OVIF，同时硬件自动清除LPTEN。

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。

33.5.2 外部脉冲触发计数

外部脉冲触发计数模式（LPTCFG.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LPTCFG.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。

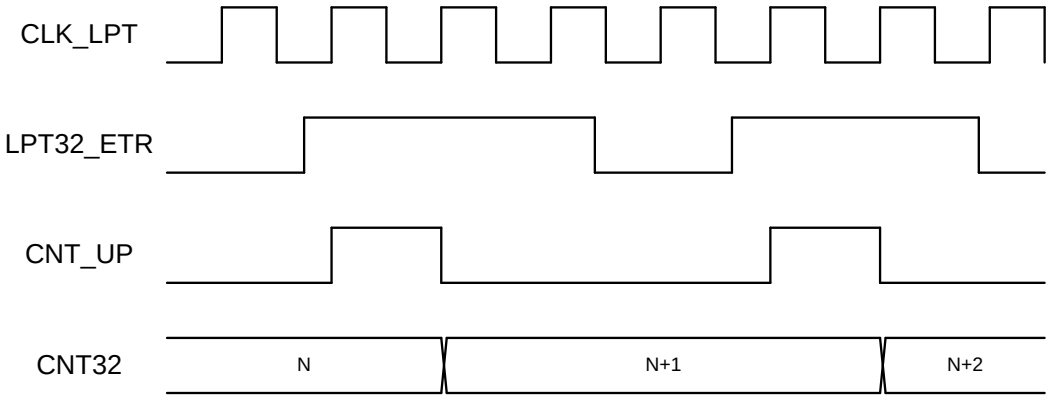


图 33-2 外部 ETR 脉冲上升沿触发计数

33.5.3 外部异步脉冲计数

外部异步脉冲计数模式（LPTCFG.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。

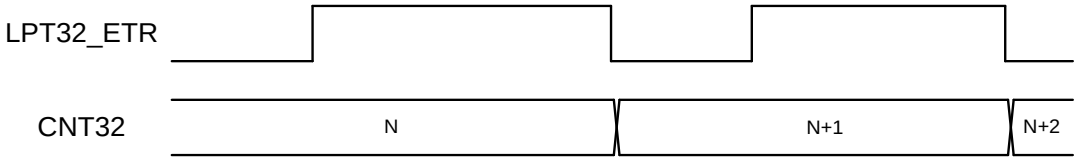


图 33-3 外部 ETR 脉冲异步计数（下降沿）

33.5.4 Timeout 模式

Timeout模式（LPTCFG.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（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。

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LPTCFG.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。

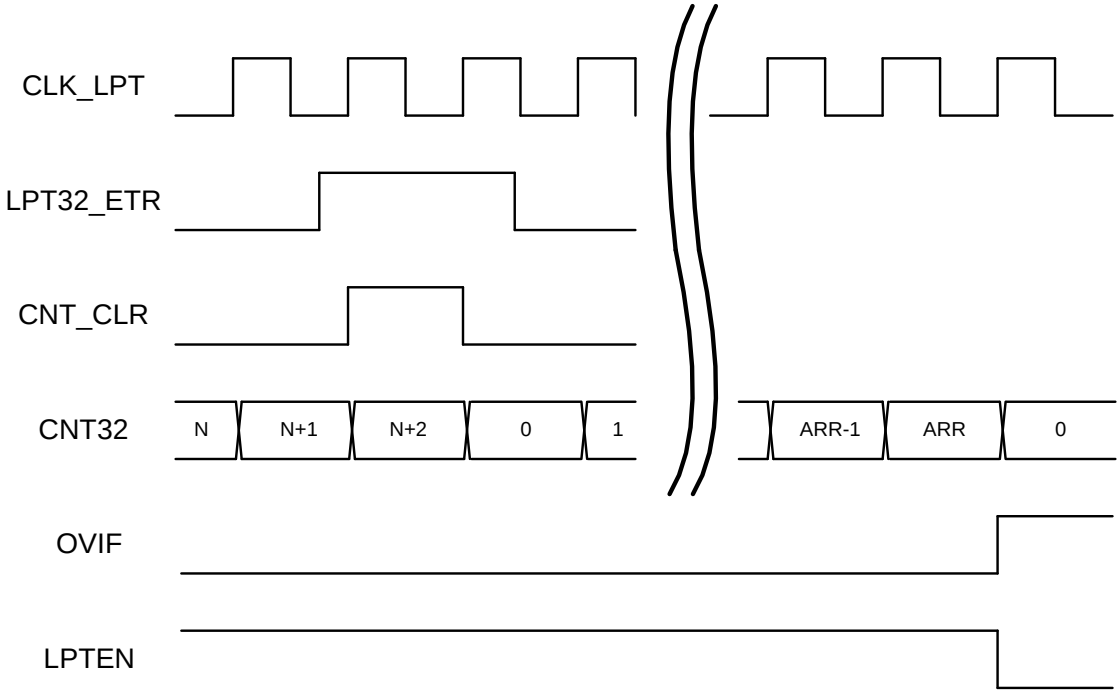


图 33-4TimeOut 模式

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，LPTIM超时溢出中断将唤醒芯片。

33.6 捕捉比较功能

LPTIM32带有两个独立的32bit捕捉比较通道，以32bit定时器为时基，结合CCRx寄存器，可以实现两路32bit PWM输出，或32bit输入捕捉功能。

33.6.1 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，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。

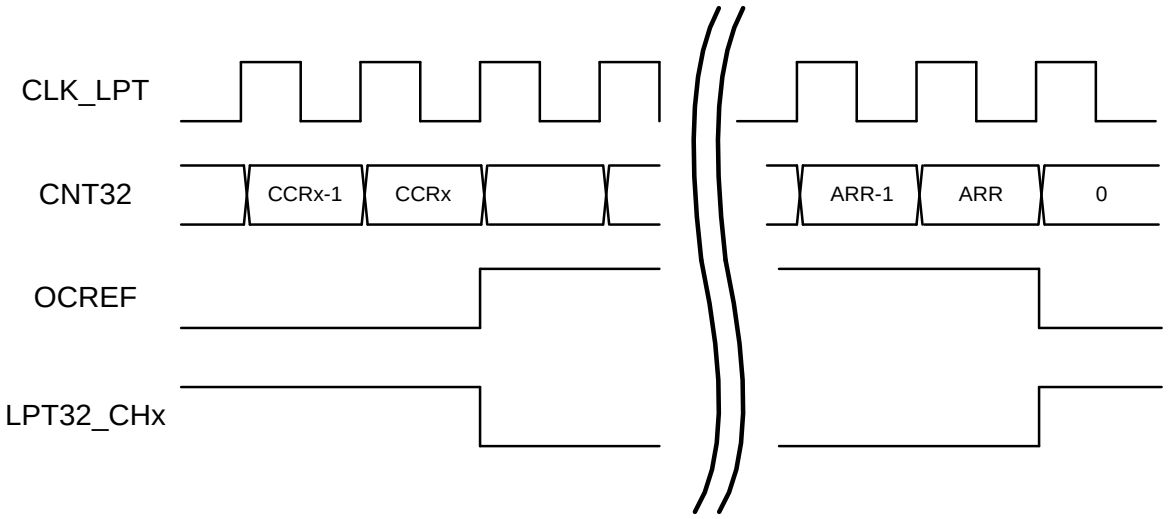


图 33-5PWM 输出

33.6.2 输入捕捉

LPTIM32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）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。

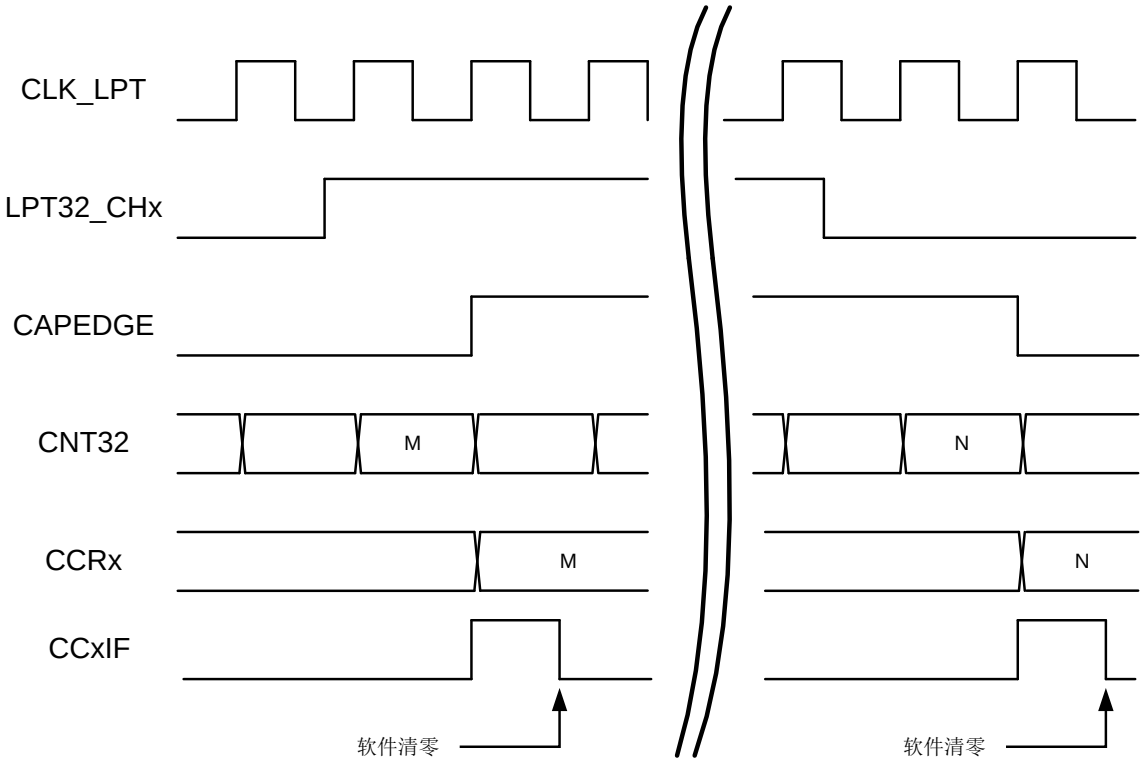


图 33-6 输入信号边沿捕捉

33.7 触发信号输出

LPTIM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：

- LPTIM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LPTIM的总线时钟，即置位LPT32_PCE寄存器。

33.8 寄存器

offset 地址	名称	符号
LPTIM32(模块基地址：0x40013400)		
0x00000000	LPTIM 配置寄存器 (LPTIM Config Register)	LPTIM_CFGR
0x00000004	LPTIM 计数寄存器 （LPTIM Counter Register）	LPTIM_CNT
0x00000008	LPTIM 捕捉比较控制和状态寄存器 （LPTIM Capture/Compare Control and Status Register）	LPTIM_CCSR
0x0000000C	LPTIM 目标值寄存器 （LPTIM Auto-Reload Register）	LPTIM_ARR
0x00000010	LPTIM 中断使能寄存器 （LPTIM Interrupt Enable Register）	LPTIM_IER
0x00000014	LPTIM 中断标志寄存器 （LPTIM Interrupt Status Register）	LPTIM_ISR
0x00000018	LPTIM 控制寄存器 （LPTIM Control Register）	LPTIM_CR
0x00000020	LPTIM 捕捉比较寄存器 1 （LPTIM Capture/Compare Register1）	LPTIM_CCR1
0x00000024	LPTIM 捕捉比较寄存器 2 （LPTIM Capture/Compare Register2）	LPTIM_CCR2
0x00000028	LPTIM 捕捉比较寄存器 3 （LPTIM Capture/Compare Register3）	LPTIM_CCR3
0x0000002C	LPTIM 捕捉比较寄存器 4 （LPTIM Capture/Compare Register4）	LPTIM_CCR4

33.8.1 LPTIM 配置寄存器（LPTIM_CFGR）

名称	LPTIM_CFGR							
Offset	0x00000000							
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
31:25	--	**未实现：读为0**

位号	位名	说明
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：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	**--**	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

33.8.2 LPTIM 计数值寄存器（LPTIM_CNT）

名称	LPTIM_CNT							
Offset	0x00000004							
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)

33.8.3 LPTIM 捕捉比较控制和状态寄存器（LPTIM_CCSR）

名称	LPTIM_CCSR							
Offset	0x00000008							
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：上升沿 1：下降沿
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：上升沿 1：下降沿
21	**CAP2EDGE**	通道 2 当前被捕捉的边沿，在 CC2IF 置位时更新(Channel2 Captured Edge) 0：上升沿 1：下降沿
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：上升沿 1：下降沿
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）

33.8.4 LPTIM 目标值寄存器（LPTIM_ARR）

名称	LPTIM_ARR							
Offset	0x0000000C							
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

33.8.5 LPTIM 中断使能寄存器（LPTIM_IER）

名称	LPTIM_IER							
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	-				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

33.8.6 LPTIM 中断标志寄存器（LPTIM_ISR）

名称	LPTIM_ISR							
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	-				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。*

33.8.7 LPTIM 控制寄存器（LPTIM_CR）

名称	LPTIM_CR							
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	-							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

33.8.8 LPTIM 捕捉比较寄存器 1（LPTIM_CCR1）

名称	**LPTIM_CCR1**							
Offset	0x00000020							
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)

33.8.9 LPTIM 捕捉比较寄存器 2（LPTIM_CCR2）

名称	**LPTIN_CCR2**							
offset	0x00000024							
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)

33.8.10 LPTIM 捕捉比较寄存器 3（LPTIM_CCR3）

名称	LPTIM_CCR3							
offset	0x00000028							
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)

33.8.11 LPTIM 捕捉比较寄存器 4（LPTIM_CCR4）

名称	LPTIM_CCR4							
offset	0x0000002C							
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)

34 16 位低功耗定时器（LPTIM16）

34.1 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、RCLF
  - 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

34.2 结构框图

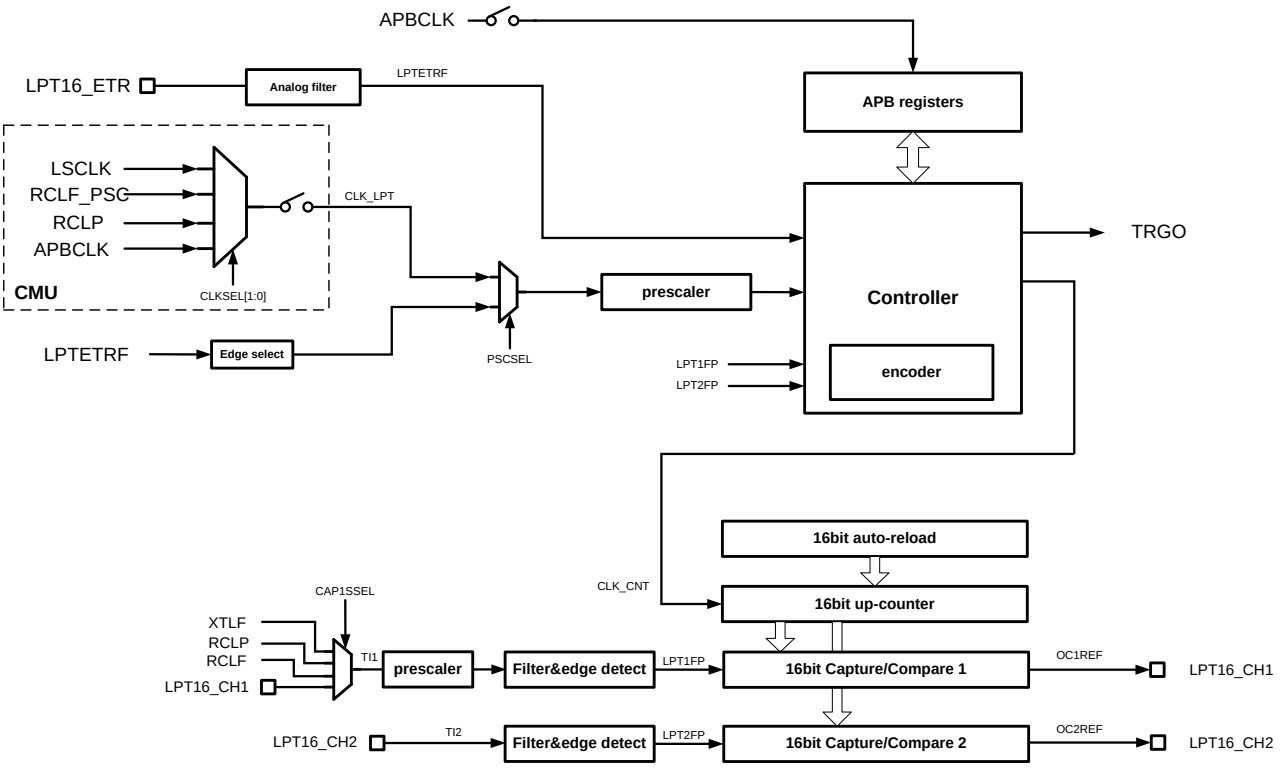


图 34-1 LPTIM16 结构框图

34.3 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14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中的外设独立工作时钟可以选择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外设复位寄存器。

34.4 相关引脚

功能	引脚映射			
	LQFP80	LQFP64	LQFP48	QFN32
LPT16_ETR	PD5	PD5	-	-
LPT16_CH1	PC13	-	-	-
LPT16_CH2	PC14	-	-	-

表 34-1 LPTIM16 引脚映射

34.5 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。

34.5.1 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OVIF。

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OVIF，同时硬件自动清除LPTEN。

注意：由于LPTIM16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LPTIM16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（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。

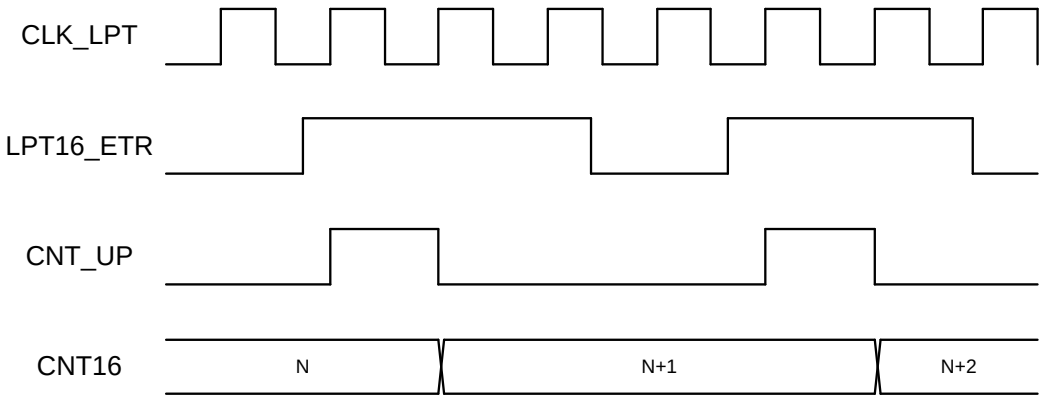


图 34-2 外部 ETR 脉冲上升沿触发计数

34.5.3 外部异步脉冲计数

外部异步脉冲计数模式（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。

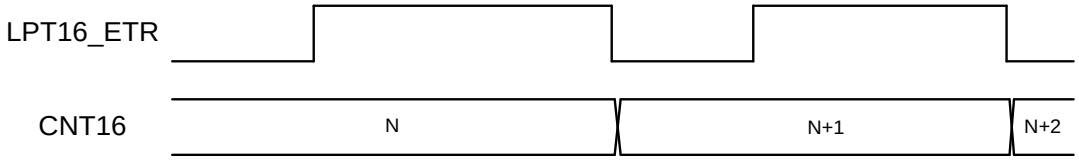


图 34-3 外部 ETR 脉冲异步计数（下降沿）

34.5.4 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。

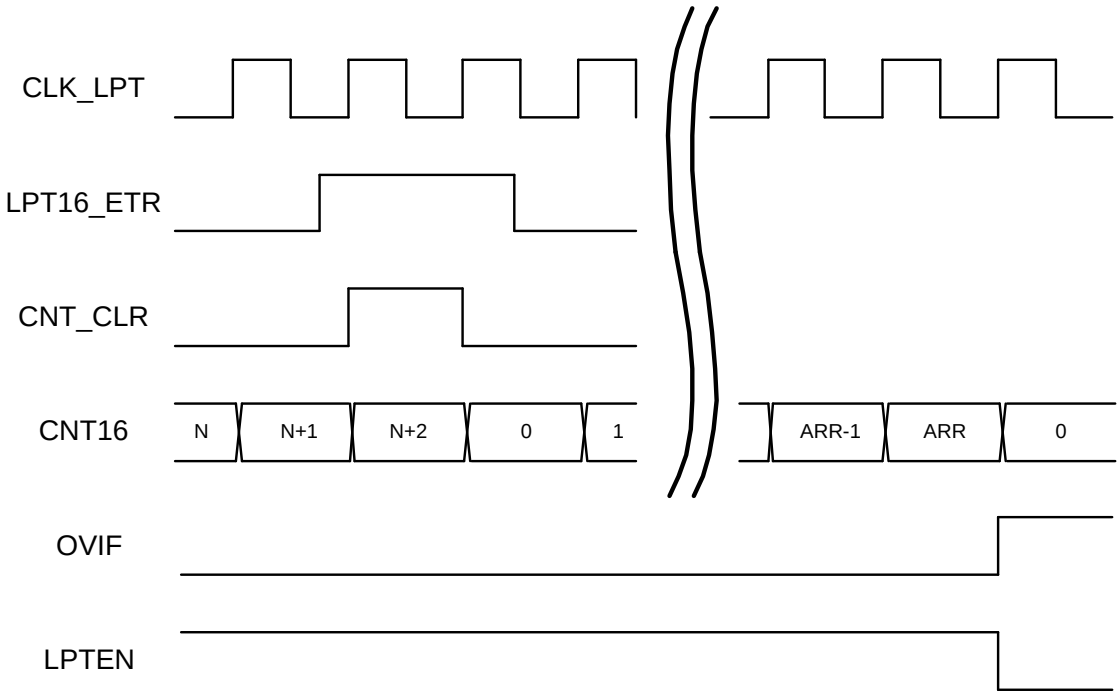


图 34-4TimeOut 模式

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。

34.6 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。

34.6.1 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。

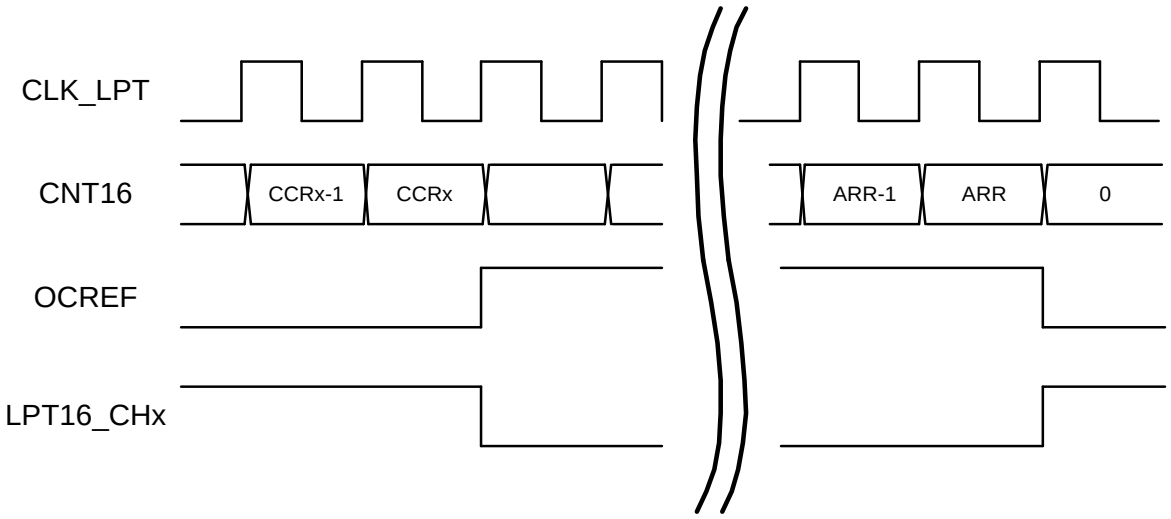


图 34-5PWM 输出

34.6.2 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、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。

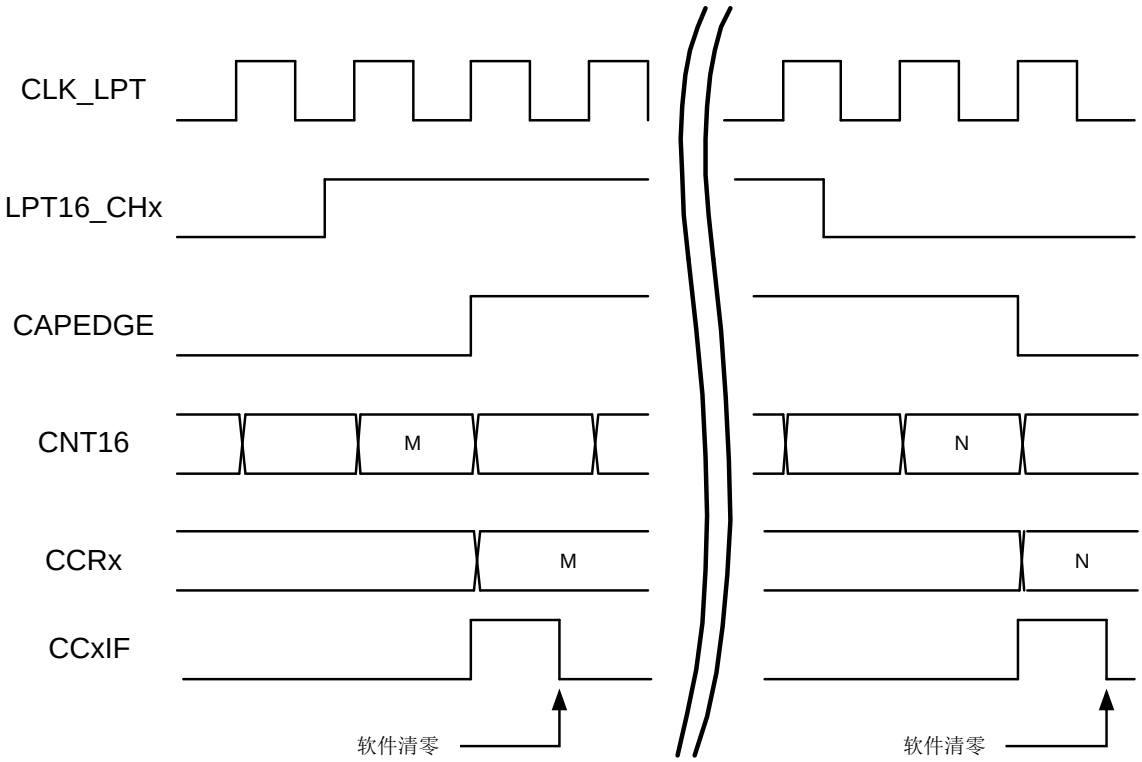


图 34-6 输入信号边沿捕捉

34.6.3 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。

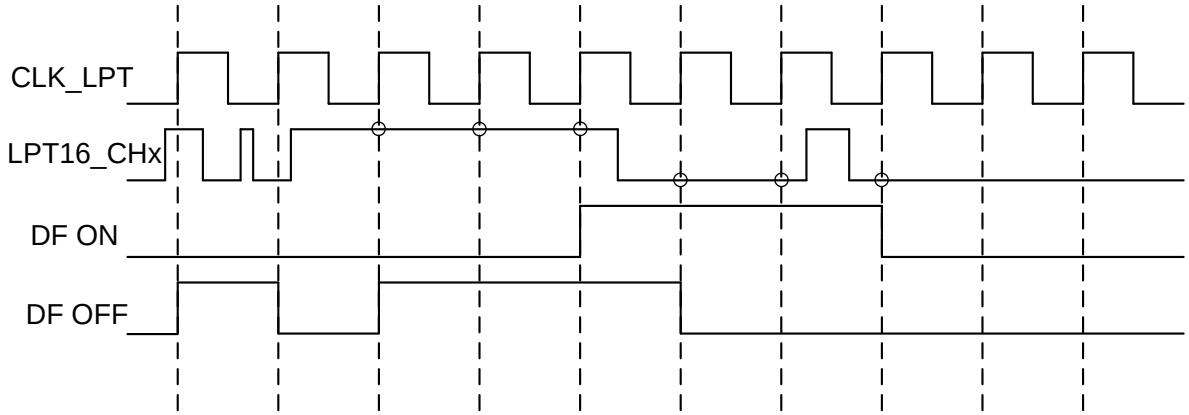


图 34-7 通道输入数字滤波

34.7 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34-2 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。

下图为编码器在LPT1FP和LPT2FP边沿同时计数的例子。

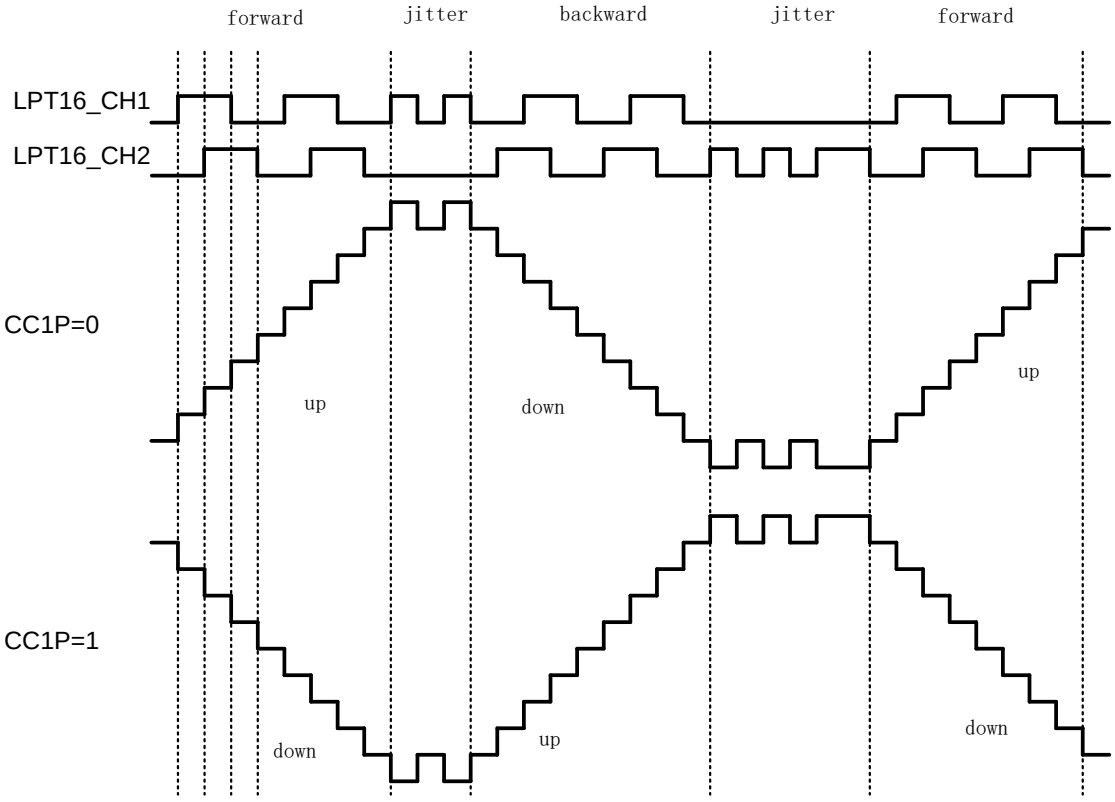


图34-8编码器模式下的计数器操作实例

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。

QECD寄存器用于配置正交编码计数的方向。

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

34.8 触发信号输出

LPTIM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：

- LPTIM使能
- 更新事件update event：计数器溢出、计数值等于CCR1或计数值等于CCR2
- 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

触发信号同步于APBCLK，因此在使用这个功能前必须打开LPTIM的总线时钟，即置位LPT16_PCE寄存器。

34.9 寄存器

offset 地址	名称	符号
LPTIM(模块基地址：0x40018800)		
0x00000000	LPTIM 配置寄存器 (LPTIM Config Register)	LPTIM_CFGR
0x00000004	LPTIM 计数寄存器 （LPTIM Counter Register）	LPTIM_CNT
0x00000008	LPTIM 捕捉比较控制和状态寄存器 （LPTIM Capture/Compare Control and Status Register）	LPTIM_CCSR
0x0000000C	LPTIM 目标值寄存器 （LPTIM Auto-Reload Register）	LPTIM_ARR
0x00000010	LPTIM 中断使能寄存器 （LPTIM Interrupt Enable Register）	LPTIM_IER
0x00000014	LPTIM 中断标志寄存器 （LPTIM Interrupt Status Register）	LPTIM_ISR
0x00000018	LPTIM 控制寄存器 （LPTIM Control Register）	LPTIM_CR
0x00000020	LPTIM 捕捉比较寄存器 1 （LPTIM Capture/Compare Register1）	LPTIM_CCR1
0x00000024	LPTIM 捕捉比较寄存器 2 （LPTIM Capture/Compare Register2）	LPTIM_CCR2

34.9.1 LPTIM 配置寄存器（LPTIM_CFGR）

名称	LPTIM_CFGR							
Offset	0x00000000							
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：CC1 比较脉冲，如果 CC1IF 标志将要置位，TRGO 输出一个正脉冲 100：CC1 通道捕捉事件 101：CC2 通道捕捉事件 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

34.9.2 LPTIM 计数值寄存器（LPTIM_CNT）

名称	LPTIM_CNT							
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	CNT16[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT16[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:16	**--**	未实现：读为0
15:0	**CNT16**	16bit 计数器当前计数值(Counter 16bits-wide)

34.9.3 LPTIM 捕捉比较控制和状态寄存器（LPTIM_CCSR）

名称	LPTIM_CCSR							
Offset	0x00000008							
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 0x01：2分频 0x02：3分频 …… 0x3F：64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：RCLF
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：上升沿 1：下降沿
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：上升沿 1：下降沿
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

位号	位名	说明
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
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）

34.9.4 LPTIM 目标值寄存器（LPTIM_ARR）

名称	LPTIM_ARR							
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	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

34.9.5 LPTIM 中断使能寄存器（LPTIM_IER）

名称	LPTIM_IER							
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	-						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	-				CC2IE	CC1IE
位权限	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
5:2	--	**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

34.9.6 LPTIM 中断标志寄存器（LPTIM_ISR）

名称	LPTIM_ISR							
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	-						CAP2OVR	CAP1OVR
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGIF	OVIF	-				CC2IF	CC1IF
位权限	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	**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：非编码器模式 – 计数器等于 ARR 编码器模式 – 计数器上溢出（等于 ARR），或者下溢出 0：无中断产生
5:2	--	**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 *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。*

34.9.7 LPTIM 控制寄存器（LPTIM_CR）

名称	LPTIM_CR							
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	-							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

34.9.8 LPTIM 捕捉比较寄存器 1（LPTIM_CCR1）

名称	LPTIM_CCR1							
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	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)

34.9.9 LPTIM 捕捉比较寄存器 2（LPTIM_CCR2）

名称	LPTIM_CCR2							
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	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)

35 实时时钟（RTCA）

35.1 概述

实时时钟(RTCA)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

35.2 结构框图

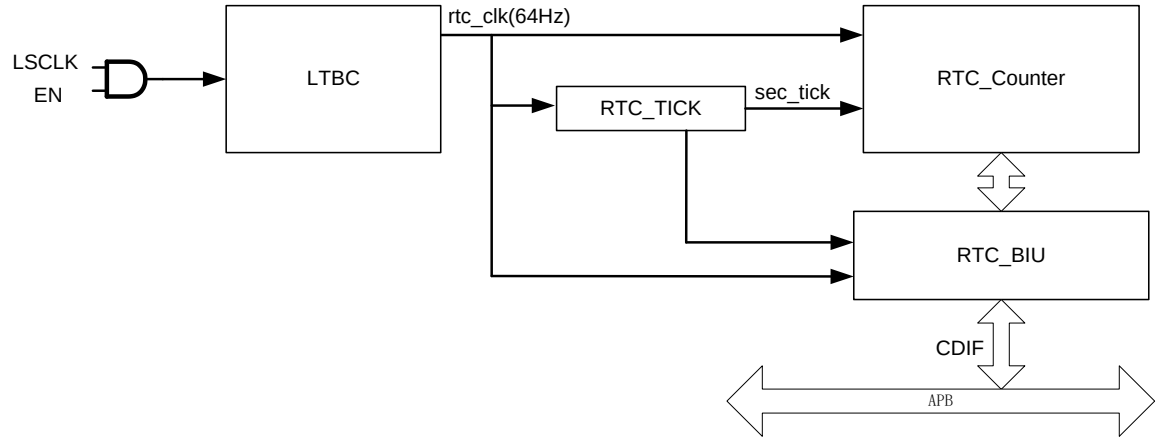


图 35-1RTC 结构框图

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。

35.3 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。

35.3.1 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

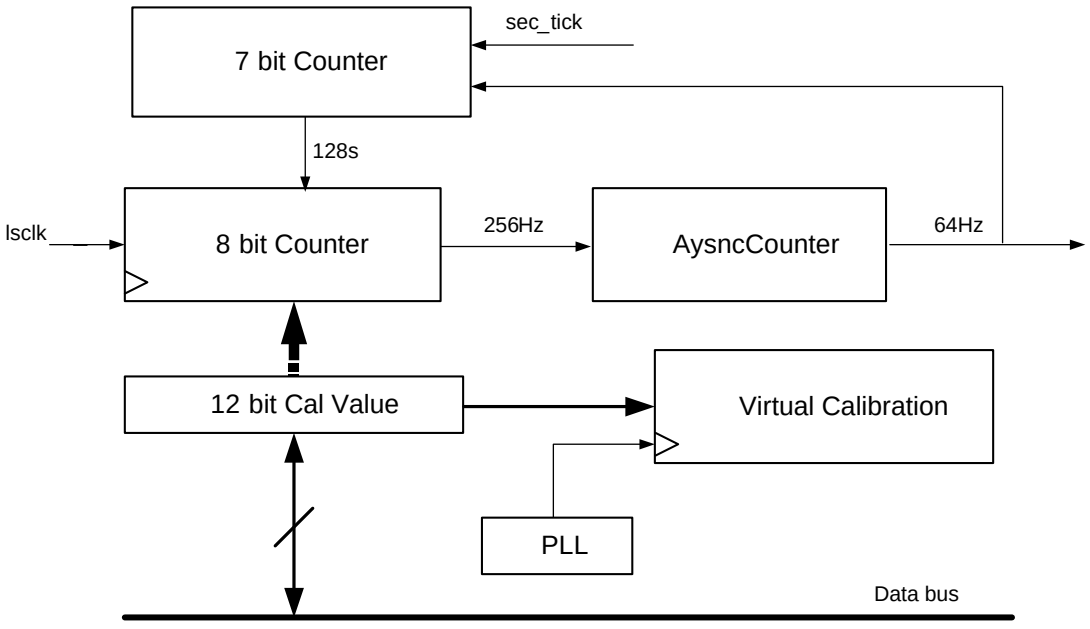


图 35-2LTBC 结构框图

35.3.2 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。FM33LG0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。

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。

35.3.3 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

35.3.4 RTC 使能与停止

RTC_A 可以由软件使能或关闭。上电后 RTC_A 默认关闭，软件在 XTLF 起振后通过操作 EN 寄存器使能 RTC_A 走时。

当 EN 寄存器清零时，RTC_A 内部走时时钟被门控，以节省功耗。

35.3.5 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 FM33LG0 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，FM33LG0 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。

35.3.6 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。

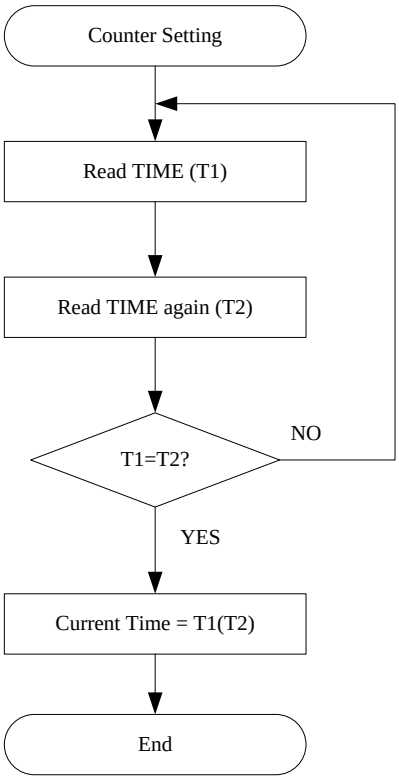


图 35-3RTC 时间读取流程图

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。

35.3.7 闰年判断

RTC_B 模块会在 2000-2099 范围内自动判断闰年，并处理 BCD 时间。

35.4 寄存器

offset 地址	名称	符号
RTCA(模块基地址：0x40011000)		
0x00000000	RTCA 写使能寄存器 （RTC Write Enable Register）	RTCA_WER
0x00000004	RTCA 中断使能寄存器 （RTC Interrupt Enable Register）	RTCA _IER
0x00000008	RTCA 中断标志寄存器 （RTC Interrupt Status Register）	RTCA _ISR
0x0000000C	BCD 时间秒寄存器 （BCD format time second registers）	RTCA _BCDSEC
0x00000010	BCD 时间分钟寄存器 （BCD format time minute registers）	RTCA _BCDMIN
0x00000014	BCD 时间小时寄存器 （BCD format time hour registers）	RTCA _BCDHOUR
0x00000018	BCD 时间天寄存器 （BCD format time day registers）	RTCA _BCDDAY
0x0000001C	BCD 时间星期寄存器 （BCD format time week registers）	RTCA _BCDWEEK
0x00000020	BCD 时间月寄存器 （BCD format time month registers）	RTCA _BCDMONTH
0x00000024	BCD 时间年寄存器 （BCD format time year registers）	RTCA _BCDYEAR
0x00000028	闹钟寄存器 （RTCA Alarm Register）	RTCA _ALARM
0x0000002C	RTC 时间信号输出寄存器 （RTCA Time Mark Select）	RTCA _TMSEL
0x00000030	LTBC 数值调整寄存器 （RTCA time Adjust Register）	RTCA _ADJUST
0x00000034	LTBC 数值调整方向寄存器 （RTCA time Adjust Sign Register）	RTCA _ADSIGN
0x0000003C	毫秒计数值寄存器 （RTCA Sub-Second Counter）	RTCA _SBSCNT
0x00000040	控制寄存器 （RTCA Control Register）	RTCA _CR

35.4.1 RTCA 写使能寄存器（RTCA_WER）

名称	RTCA_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。

35.4.2 RTCA 中断使能寄存器（RTCA_IER）

名称	RTCA_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

35.4.3 RTCA 中断标志寄存器（RTCA_ISR）

名称	RTCA_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

35.4.4 BCD 时间秒寄存器（RTCA_BCDSEC）

名称	RTCA_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)

35.4.5 BCD 时间分钟寄存器（RTCA_BCDMIN）

名称	RTCA_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)

35.4.6 BCD 时间小时寄存器（RTCA_BCDHOUR）

名称	RTCA_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)

35.4.7 BCD 时间天寄存器（RTCA_BCDDAY）

名称	RTCA_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)

35.4.8 BCD 时间星期寄存器（RTCA_BCDWEEK）

名称	RTCA_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)

35.4.9 BCD 时间月寄存器（RTCA_BCDMONTH）

名称	RTCA_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)

35.4.10 BCD 时间年寄存器（RTCA_BCDYEAR）

名称	RTCA_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)

35.4.11 闹钟寄存器（RTCA_ALARM）

名称	RTCA_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)

35.4.12 RTC 时间信号输出寄存器（RTCA_TMSEL）

名称	RTCA_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

35.4.13 LTBC 数值调整寄存器（RTCA_ADJUST）

名称	RTCA_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
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)

35.4.14 毫秒计数值寄存器（RTCA_SBSCNT）

名称	RTCA_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)

35.4.15 RTCA 控制寄存器 RTCA_CR

名称	RTCA_CR							
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	-							
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_A（内部走时时钟关闭） 1：使能 RTC_A

36 实时时钟（RTCB）

36.1 概述

实时时钟(RTCB)模块可长时间维持精确计时，处于 VAO 电源域，在 VDD 掉电时仍可以使用 VBAT 电源维持走时。

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
- RTC 计时器部分不复位
- 5 个 32bit 备份寄存器

36.2 结构框图

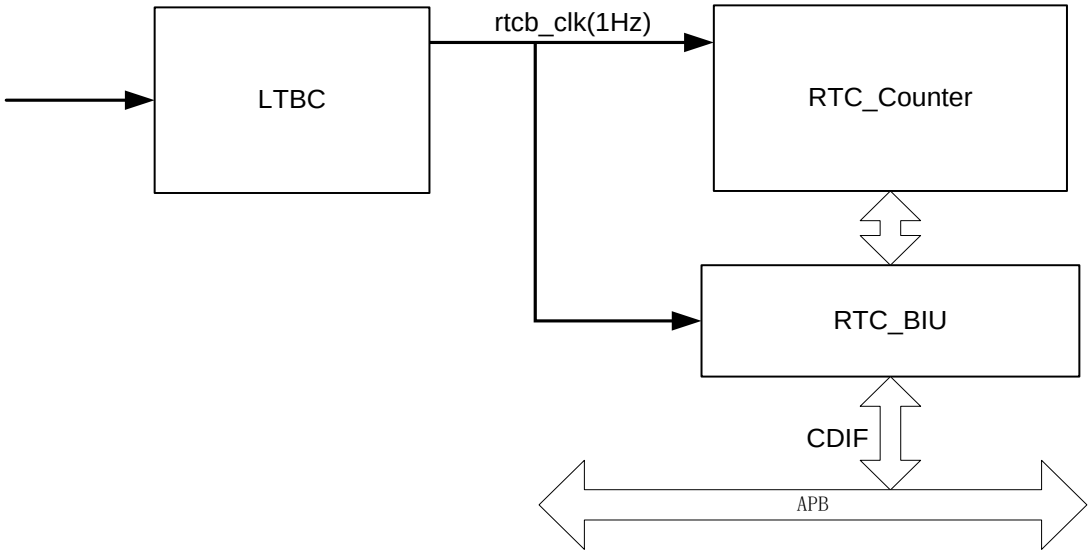


图 36-1RTC_B 结构框图

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 LTBC 功能介绍。

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。

36.3.1 工作时钟

RTC_B 只使用 XTLF 工作，如果 XTLF 停振，则 RTC_B 停止走时。

36.3.2 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 XTLF 的预分频得到 1Hz 的 RTC 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
- 可产生 1Hz 周期中断和闹钟定时中断
- 分频电路和 BCD 时间不受芯片复位影响

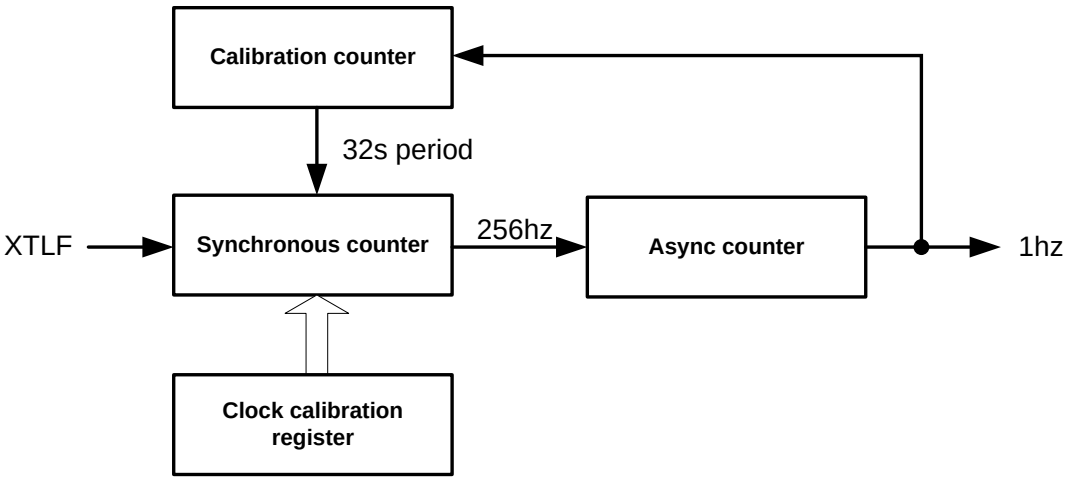


图 36-2LTBC 结构框图

36.3.3 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。FM33LG0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。

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。

36.3.4 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

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

36.3.5 RTC 使能与停止

RTC_B 可以由软件使能或关闭。上电后 RTC_B 默认使能。

36.3.6 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。

为了提高抗干扰能力，RTC_B 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。

36.3.7 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。

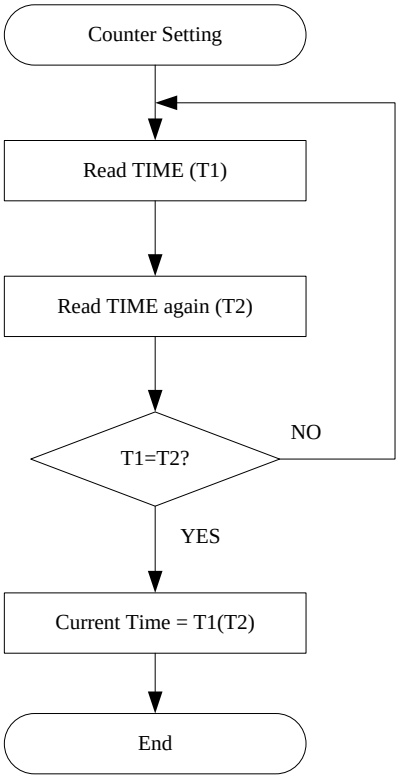


图 36-3RTC 时间读取流程图

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。

36.3.8 闰年判断

RTC_B 模块会在 2000-2099 范围内自动判断闰年，并处理 BCD 时间。

36.3.9 RTC 时间戳

为了支持 Tamper Detection，RTC_B 支持外部 IO 事件触发的时间戳功能。外部 IO 触发源为 PH15 的输入电平变化，为了确保输入检测的可靠性，建议使能 Tamper 数字滤波。使用此功能时，将 PH15 配置为 GPIO 功能，打开 RTCSTAMPEN 寄存器，当 PH15 上出现任何滤波后的上升沿或下降沿时，RTC 会自动记录当前时间到 STAMP 寄存器组中，同时产生相应的标志，可用于产生中断或者供软件查询。

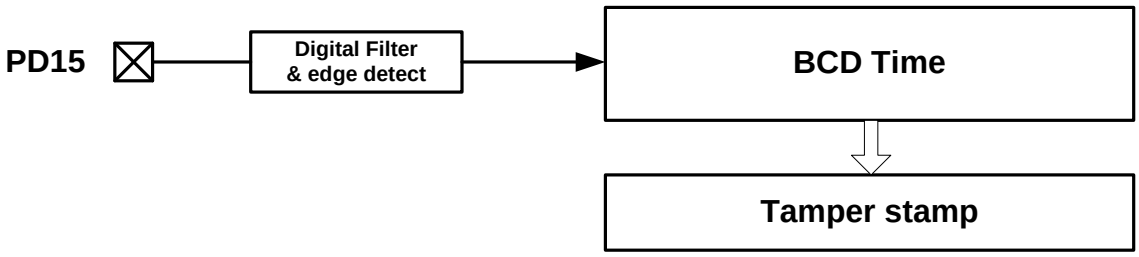


图 36-4 时间戳

时间戳仅在相应标志寄存器为 0 的情况下记录事件发生时间，如果对应标志已经为 1，则忽略相应事件。因此如果有多次事件发生，时间戳仅记录第一次事件发生的时间，除非软件在事件发生后清除了标志寄存器。

Tamper 输入信号在 RTC_B 模块内进行数字滤波。滤波方法是使用 XTLF 连续采样 3 拍相同才认为合法。数字滤波功能可以打开或关闭。

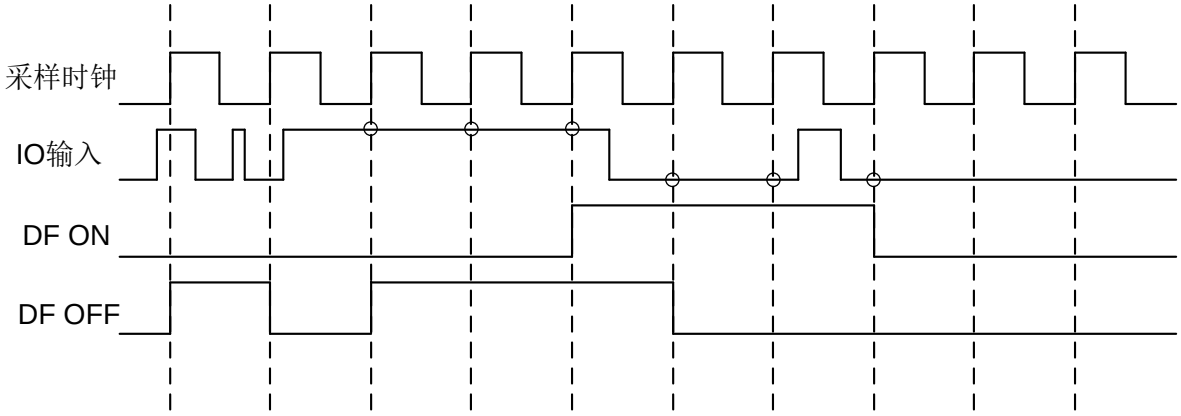


图 36-5 Tamper 输入数字滤波

36.3.10 备份寄存器

RTC_B 包含了 5 个 32bit 备份寄存器，共 20 字节。在 VDD 掉电的情况下，备份寄存器由 VBAT 供电，仍可以用于保持关键数据。

当检测到 Tamper 事件发生时，芯片可以自动擦除备份寄存器中的内容。此功能可以通过 TAMPEN 寄存器配置使能或关闭。

备份寄存器支持 tamper 事件清零功能，当特定事件发生时，硬件自动清除备份寄存器中的内容。支持的清零事件包括：

- Tamper 引脚事件（与时间戳事件相同）
- XTLF 停振事件

以上事件发生时是否清除寄存器，可以由 TAMPEN 寄存器使能，并通过 TAMPSEL 寄存器选择触发备份寄存器清除的事件源。

36.4 寄存器

offset 地址	名称	符号
RTCB(模块基地址：0x4001F000)		
0x00000000	RTCB 写使能寄存器 （RTCBWrite Enable Register）	RTCB_WER
0x00000004	RTCB 中断使能寄存器 （RTCBInterrupt Enable Register）	RTCB_IER
0x00000008	RTCB 中断标志寄存器 （RTCBInterrupt Status Register）	RTCB_ISR
0x0000000C	BCD 时间秒寄存器 （BCD format time second registers）	RTCB_BCDSEC
0x00000010	BCD 时间分钟寄存器 （BCD format time minute registers）	RTCB_BCDMIN
0x00000014	BCD 时间小时寄存器 （BCD format time hour registers）	RTCB_BCDHOUR
0x00000018	BCD 时间天寄存器 （BCD format time day registers）	RTCB_BCDDAY
0x0000001C	BCD 时间星期寄存器 （BCD format time week registers）	RTCB_BCDWEEK
0x00000020	BCD 时间月寄存器 （BCD format time month registers）	RTCB_BCDMONTH
0x00000024	BCD 时间年寄存器 （BCD format time year registers）	RTCB_BCDYEAR
0x00000028	闹钟寄存器 （RTCB Alarm Register）	RTCB_ALARM
0x0000002C	RTCB 时间信号输出寄存器 （RTCB Time Mark Select）	RTCB_TMSEL
0x00000030	RTCB 时钟调校寄存器 （RTCB time Adjust Register）	RTCB_ADJR
0x00000040	RTCB 控制寄存器 （RTCB Control Register）	RTCB_CR
0x00000048	RTCB 时间戳控制寄存器 （RTCB Time Stamp Control Register）	RTCB_STPCR
0x0000004C	RTC 时间戳时钟寄存器 （RTC Time Stamp Clock Record Register）	RTCB_STPCLKRR
0x00000050	RTCB 时间戳日历寄存器 （ RTCB Time Stamp Calendar Record Register）	RTCB_STPCALRR
0x00000070	RTCB 备份寄存器 0 （RTCBBackup Register0）	RTC_BKR0
0x00000074	RTCB 备份寄存器 1 （RTCB Backup Register1）	RTCB_BKR1
0x00000078	RTCB 备份寄存器 2 （RTCB Backup Register2）	RTCB_BKR2
0x0000007C	RTCB 备份寄存器 3 （RTCB Backup Register3）	RTCB_BKR3
0x00000080	RTCB 备份寄存器 4 （RTCB Backup Register4）	RTCB_BKR4

36.4.1 RTCB 写使能寄存器（RTCB_WER）

名称	RTCB_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 RTCB 中断使能寄存器（RTCB_IER）

名称	RTCB_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	-		STP0IE	ADJ_IE	-			
位权限	U-0		R/W-0	R/W-0	U-0			
位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
位名	-				SEC_IE	MIN_IE	HOUR_IE	DAY_IE
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
31:14	-	RFU：未实现，读为 0
13	STP0IE	时间戳中断使能 (time stamp interrupt enable) 1：中断使能打开 0：中断使能禁止

位号	助记符	功能描述
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
11:4	-	RFU：未实现，读为 0
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 RTCB 中断标志寄存器（RTCB_ISR）

名称	RTCB_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	-		STP0_IF	ADJ_IF	-			
位权限	U-0		R/W-0	R/W-0	U-0			
位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
位名	-				SEC_IF	MIN_IF	HOUR_IF	DAY_IF
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
31:14	-	RFU：未实现，读为 0
13	STP0_IF	时间戳中断标志。写 1 清零 (time stamp Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
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
11:4	-	RFU：未实现，读为 0
3	SEC_IF	秒中断标志。写 1 清零 (1hz periodic Interrupt Flag, write 1 to clear)

位号	助记符	功能描述
		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（RTCB_BCDSEC）

名称	RTCB_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（RTCB_BCDMIN）

名称	RTCB_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（RTCB_BCDHOUR）

名称	RTCB_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（RTCB_BCDDAY）

名称	RTCB_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（RTCB_BCDWEEK）

名称	RTCB_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（RTCB_BCDMONTH）

名称	RTCB_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（RTCB_BCDYEAR）

名称	RTCB_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 RTCB 时间信号输出寄存器（RTCB_TMSEL）

名称	RTCB_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	此寄存器配置从 RTCOUT（PH15）引脚输出的 RTCB 内部频率信号。 频率输出选择信号： 4’b0000：关闭输出 4’b0001：RFU

位号	助记符	功能描述
		4’b0010：输出秒计数器进位信号，高电平宽度 1s 4’b0011：输出分计数器进位信号，高电平宽度 1s 4’b0100：输出小时计数器进位信号，高电平宽度 1s 4’b0101：输出天计数器进位信号，高电平宽度 1s 4’b0110：RFU 4’b0111：输出 32 秒方波信号 4’b1000：秒时标信号 其他：关闭输出

36.4.12 RTCB 时钟调校寄存器（RTCB_ADJR）

名称	RTCB_ADJR							
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	-						ADJUST[9:8]	
位权限	U-0						R/W-xxx	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADJUST[7:0]							
位权限	R/W-xxxxxxxx							

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
9:0	ADJUST	LTBC 补偿调整数值，其中最高位是符号位，表示调整方向；低 9 位为调整值。

36.4.13 RTCB 控制寄存器（RTCB_CR）

名称	RTCB_CR							
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	-							

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
31:1	--	RFU：未实现，读为 0
0	EN	RTC 使能寄存器 (RTC enable)，默认使能 0：关闭 RTC_B 1：使能 RTC_B

36.4.14 RTCB 时间戳控制寄存器（RTCB_STPCR）

名称	RTCB_STPCR							
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	-						TAMPSEL	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DF	-				TSEDGE	TAMPEN	TSEN
位权限	R/W-0	U-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
9:8	TAMPSEL	Tamper 清除备份寄存器事件源选择 00：无 01：Tamper pin 事件，由 TSEDGE 选择有效沿 10：XTLF 停振事假 11：Tamper pin 和 XTLF 停振同时有效 *注意：如果选择 XTLF 停振，则当停振发生时备份寄存器一直被异步复位，软件无法再写入数据，直到 XTLF 恢复*
7	DF	Tamper Pin 数字滤波使能 (Tamper digital filter enable) 0：关闭数字滤波 1：使能数字滤波
6:3	--	RFU：未实现，读为 0
2	TSEDGE	时间戳事件边沿选择 (Time Stamp Edge select) 0：输入信号上沿产生时间戳事件和中断 1：输入信号下沿产生时间戳事件和中断
1	TAMPEN	Tamper 事件清除备份寄存器使能 (Tamper Enable) 1：当 tamper 事件发生时，自动清除备份寄存器

Bit	助记符	功能描述
		0：当 tamper 事件发生时，保持备份寄存器
0	TSEN	PH15 触发的时间戳功能使能位 (Time Stamp Enable) 1：打开时间戳 0：关闭时间戳

36.4.15 RTCB 时间戳时钟寄存器（RTCB_STPCLKRR）

名称	RTCB_STPCLKRR							
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	-		HRSTP0R					
位权限	U-0		RW-X					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	MINSTP0R						
位权限	U-0	RW-X						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	SECSTP0R						
位权限	U-0	RW-X						

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
21:16	HRSTP0R	检测到 PH15 上升沿后存储 BCD 小时寄存器的值。
15	--	RFU：未实现，读为 0
14:8	MINSTP0R	检测到 PH15 上升沿后存储 BCD 分寄存器的值。
7	--	RFU：未实现，读为 0
6:0	SECSTP0R	检测到 PH15 上升沿后存储 BCD 秒寄存器的值。

36.4.16 RTCB 时间戳日历寄存器（RTCB_STPCALRR）

名称	RTCB_STPCALRR							
offset	0x50							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	YRSTP0R							
位权限	RW-X							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名				MONSTP0R				
位权限				RW-X				
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	-		DAYSTP0R					
位权限	U-0		RW-X					

Bit	助记符	功能描述
31:24	YRSTP0R	检测到 PH15 上升沿后存储 BCD 年寄存器的值。
23:21	--	RFU：未实现，读为 0
20:16	MONSTP0R	检测到 PH15 上升沿后存储 BCD 月寄存器的值。
15:6	--	RFU：未实现，读为 0
5:0	DAYSTP0R	检测到 PH15 上升沿后存储 BCD 天寄存器的值。

36.4.17 RTCB 备份寄存器组（RTCB_BKRx）

总共20字节，可以读写，无复位值。

名称	RTCB_BKRx(x=0,1,2,3,4)							
offset	0x70~80							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	BKP[31:24]							
位权限	R/W-xxxx xxxx							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	BKP[23:16]							
位权限	R/W-xxxx xxxx							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BKP[15:8]							
位权限	R/W-xxxx xxxx							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BKP[7:0]							
位权限	R/W-xxxx xxxx							

位号	助记符	功能描述
31:0	BKP	备份寄存器，可读写，无复位值 (RTC Backup Registers)

37 段码液晶驱动（LCD）

37.1 概述

LCD 显示驱动模块用于驱动段码式液晶屏，能够支持 4、6、8COM，最大显示段数分别为 128 段（4COM）、180 段（6COM）和 224 段（8COM）。

主要特点：

- 最大支持 8×40、6×42、4×44 的显示段数
- 1/3bias、1/4bias
- 16 级灰度可调
- LCD 驱动支持片内电阻型、片外电容型 2 种模式
- 支持闪烁功能，且闪烁频率可调
- 支持间歇式点亮功能，点亮、熄灭时间可配置
- 支持全亮、全灭功能
- 低功耗，LCD 驱动可以在 Active 模式、Sleep 模式和 DeepSleep 模式下工作
- 支持 Type A 和 Type B 两种 LCD 驱动波形（可配置）
- 典型帧刷新频率 64Hz

37.2 结构框图

LCD 显示驱动模块结构框图如图 37-1 所示：

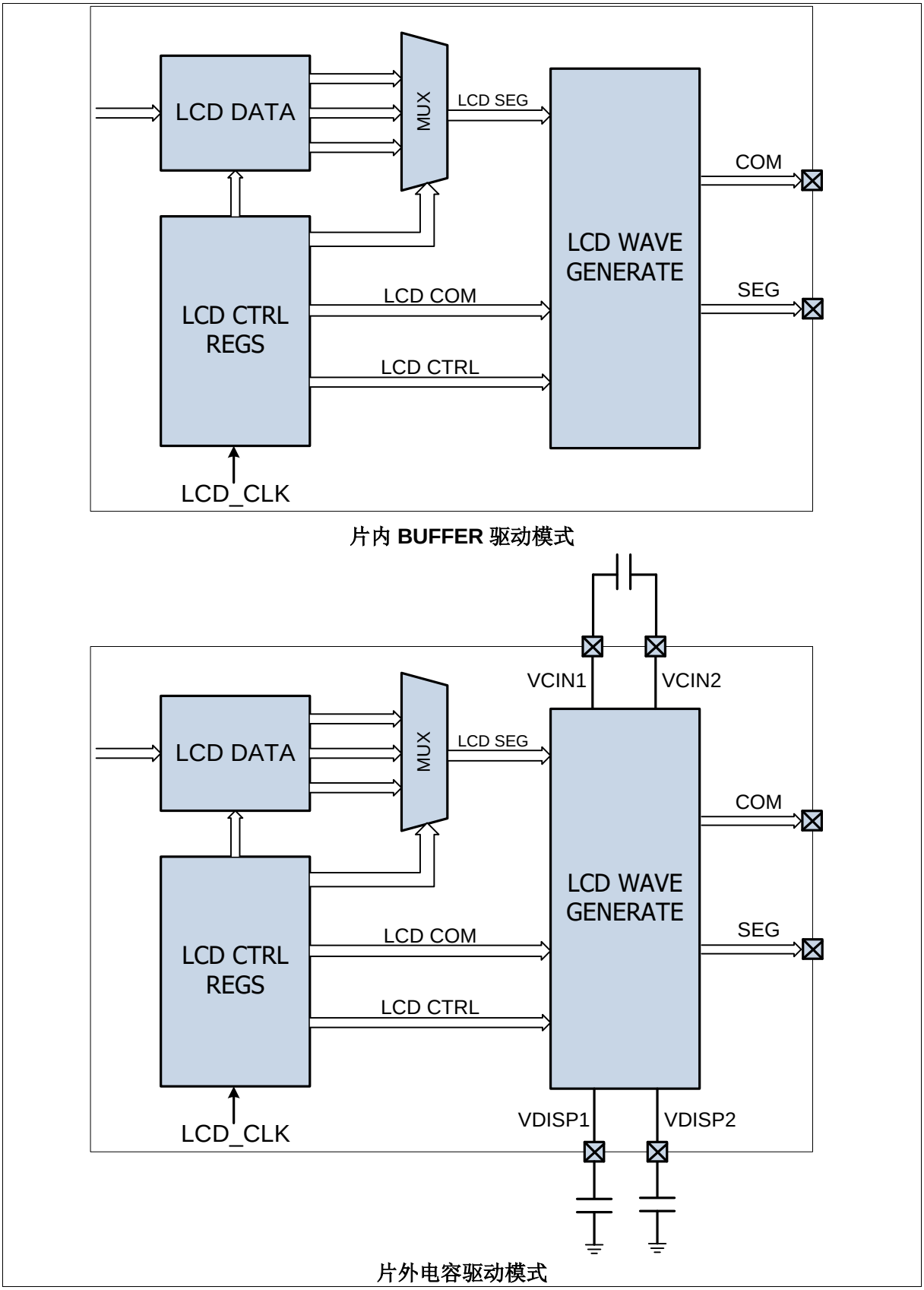


图 37-1 LCD 显示控制模块结构框图

37.3 IO 配置

LCD驱动电路工作时最多会占用36个GPIO，在使用LCD前，需要将用到的引脚设置为模拟功能（PxyFCR=11），并将相应的COMEN或SEGEN打开。如果在同一个引脚上复用了LCD之外的其他模拟功能，则必须保证其他模式外设不会使用这个引脚通道。

37.4 功能说明

37.4.1 工作时钟和显示帧频率

LCD驱动电路使用LSCLK工作，其典型频率在32KHz左右。通过配置DF寄存器，可以设置LCD显示的帧频率。帧频率的计算公式如下（注意DF不能为0）：

COM 数量	帧频率 Hz	
	A 类波形	B 类波形
4	显示电路工作频率 / (4 ×DF[7:0] × 2)	显示电路工作频率 / (4 ×DF[7:0] × 4)
6	显示电路工作频率 / (6 ×DF[7:0] × 2)	显示电路工作频率 / (6 ×DF[7:0] × 4)
8	显示电路工作频率 / (8 ×DF[7:0] × 2)	显示电路工作频率 / (8 ×DF[7:0] × 4)

表37-1帧频率计算公式

下表举例说明了DF寄存器的取值与帧频率之间的关系。通常情况下将帧频率设置为60Hz左右。

帧频率（Hz）	工作时钟（Hz）	4 公共端		6 公共端		8 公共端	
		A 类	B 类	A 类	B 类	A 类	B 类
50	32768	82	41	54	27	41	20
58	32768	70	35	47	24	35	17
64	32768	64	32	42	21	32	16
70	32768	58	29	39	20	29	14
75	32768	54	27	36	18	27	13

表37-2典型帧频率和DF的关系

37.4.2 LCD Type A 扫描波形

1/3 bias表示LCD驱动电路能够输出4种驱动电平，对偏置电压平均分配得到。1/4 bias表示LCD驱动电路能够输出5种驱动电平，对偏置电压平均分配得到。

通过LCDBIAS寄存器可以配置VLCD偏置电压大小，最大不超过电源电压。

下面的图例显示了4COM 1/3 bias、6COM 1/3 bias和1/4 bias、8COM 1/4 bias的A类驱动波形。

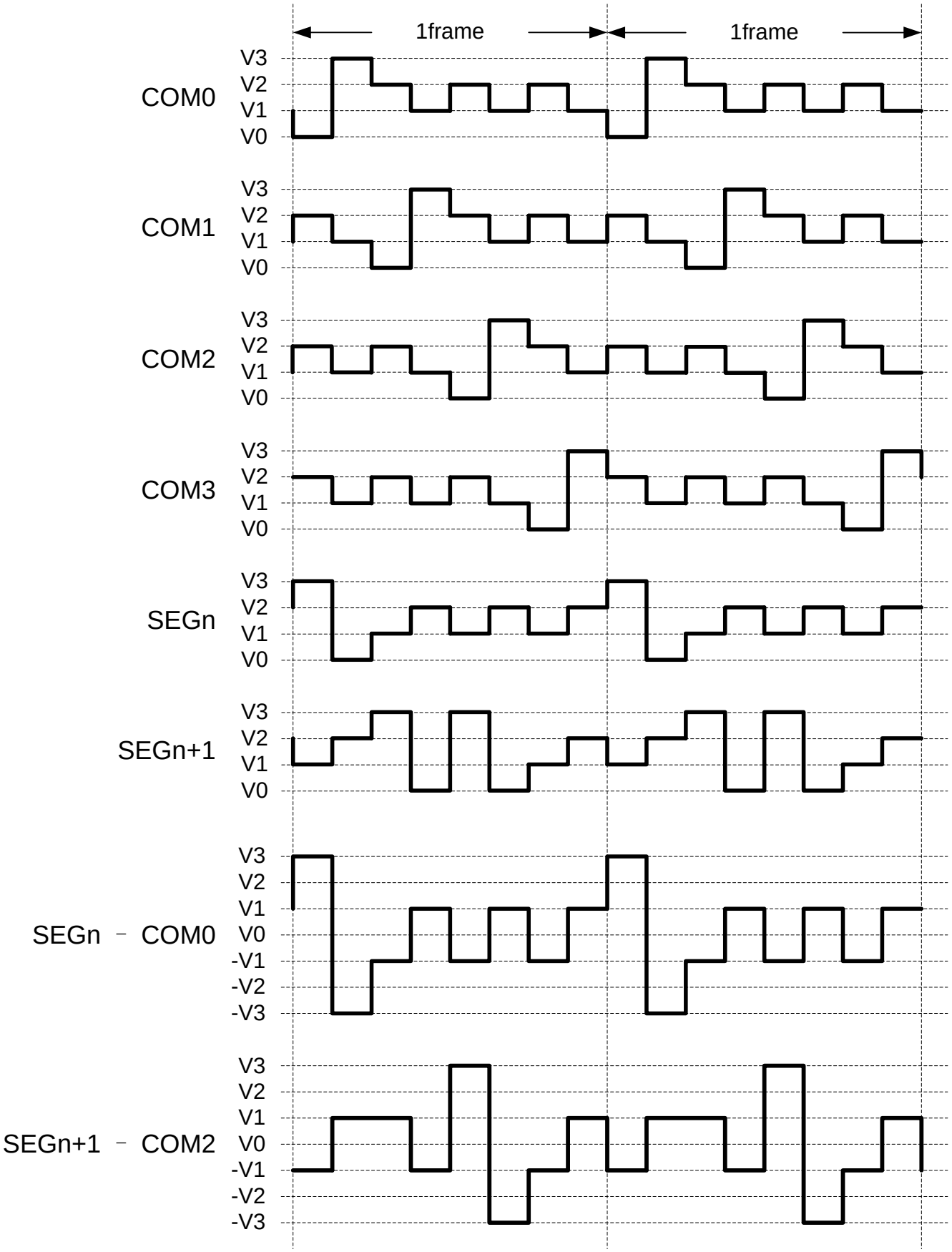


图 37-2LCD 驱动波形(1/4 duty, 1/3 bias, type A)

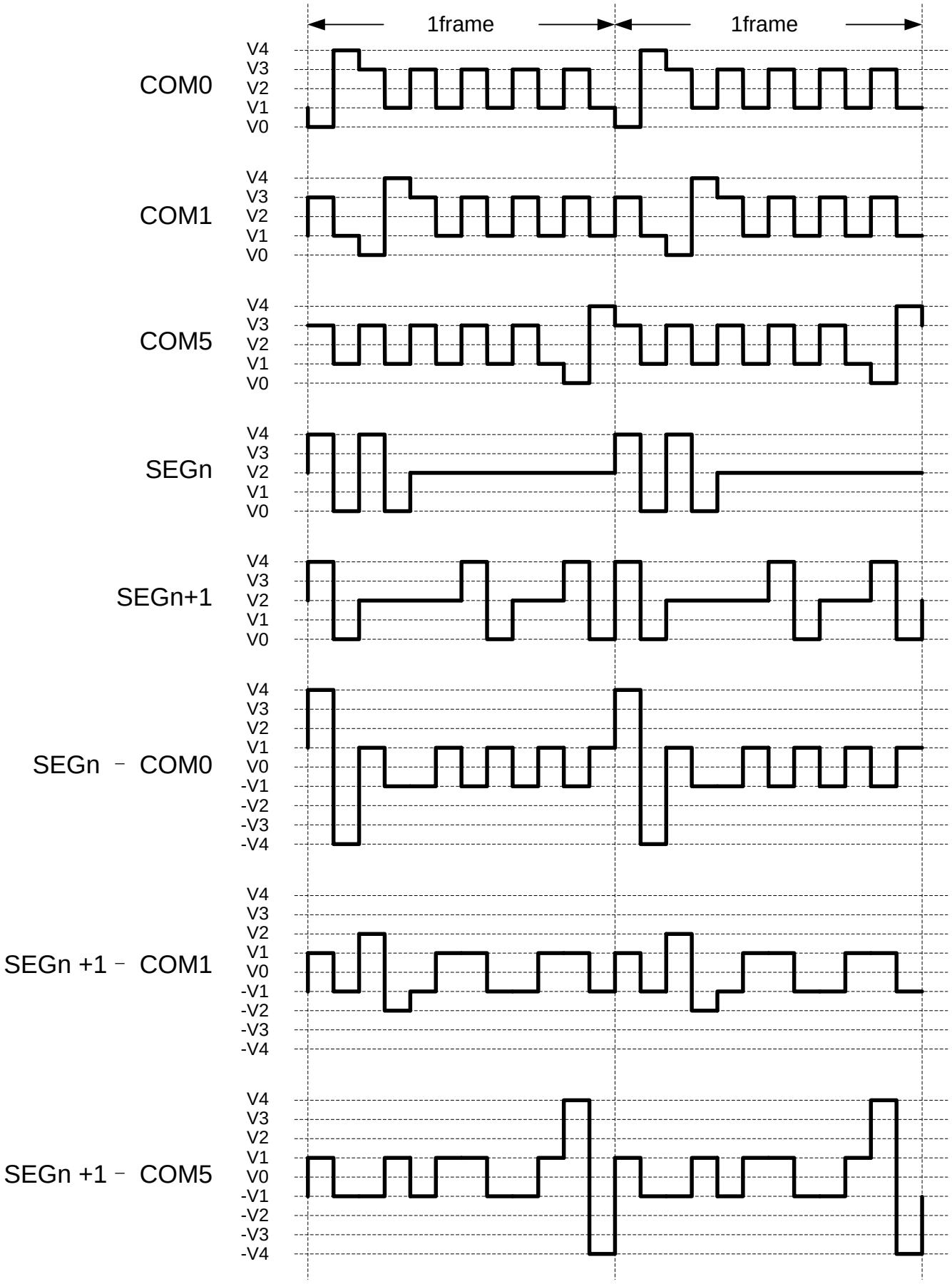


图 37-3LCD 驱动波形(1/6 duty, 1/4 bias, type A)

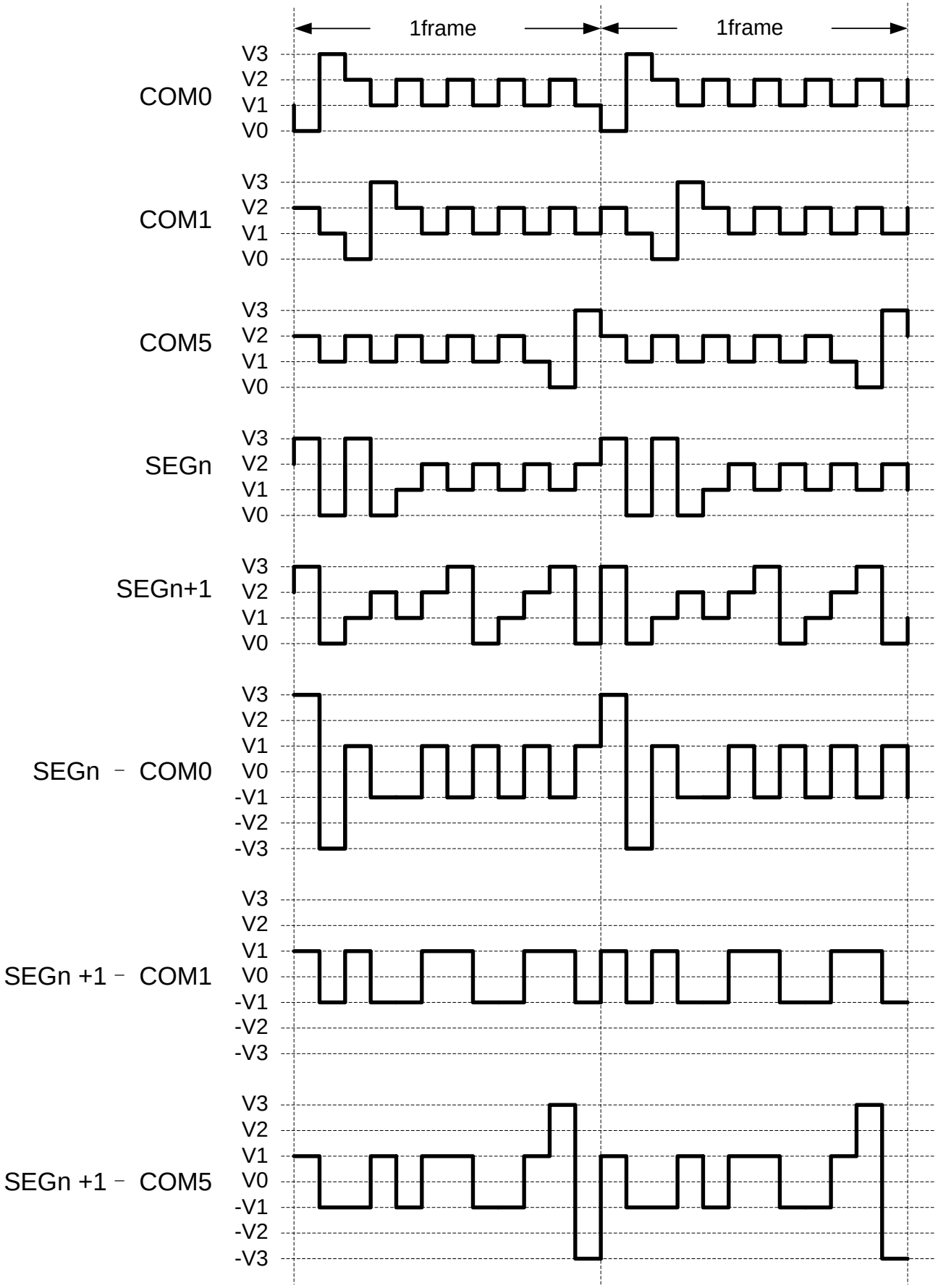


图 37-4LCD 驱动波形(1/6 duty, 1/3 bias, type A)

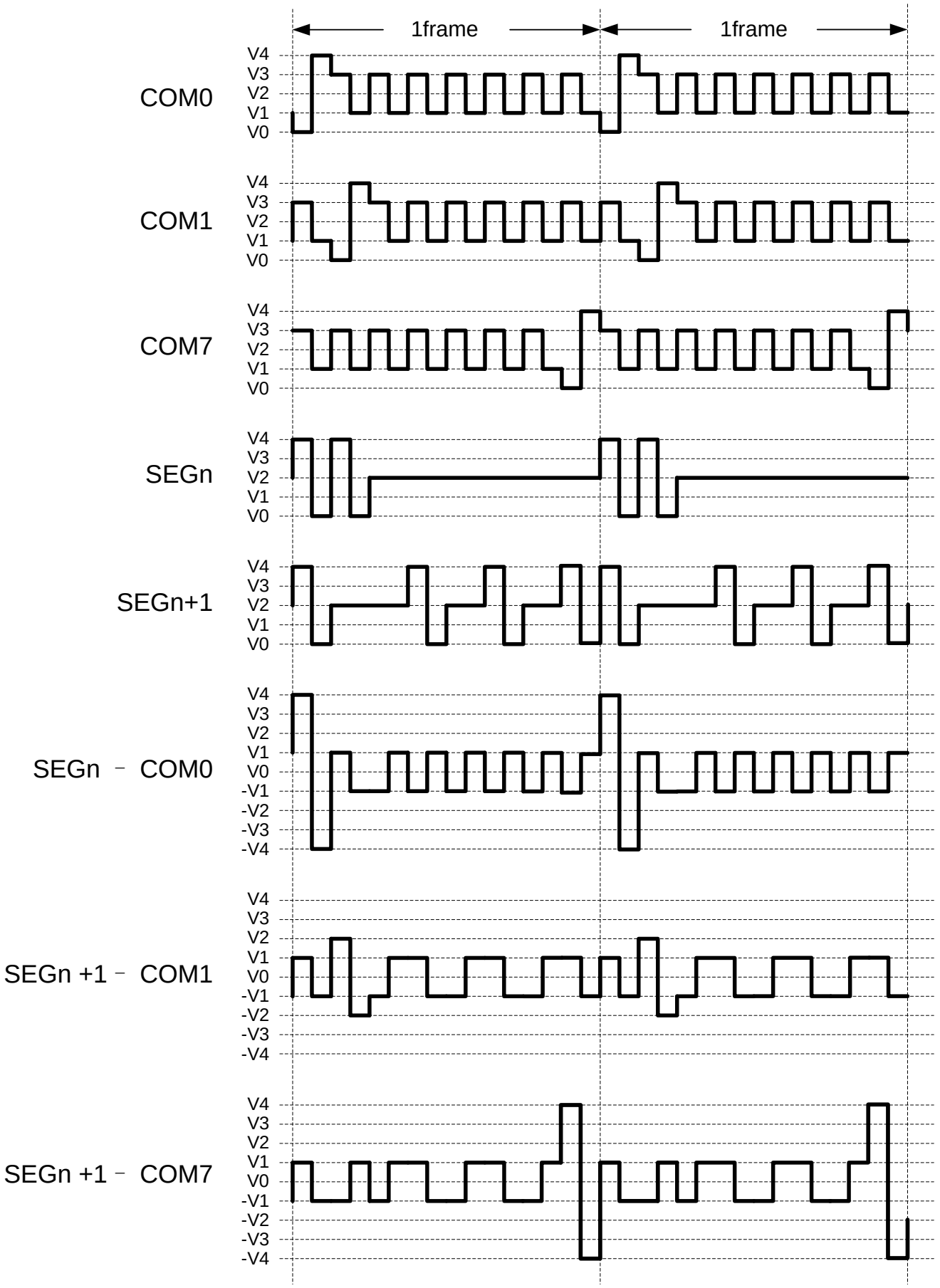


图 37-5LCD 驱动波形(1/8 duty, 1/4 bias, type A)

37.4.3 LCD Type B 扫描波形

下面的图例显示了4COM 1/3 bias、6COM 1/3 bias和1/4 bias、8COM 1/4 bias的B类驱动波形。

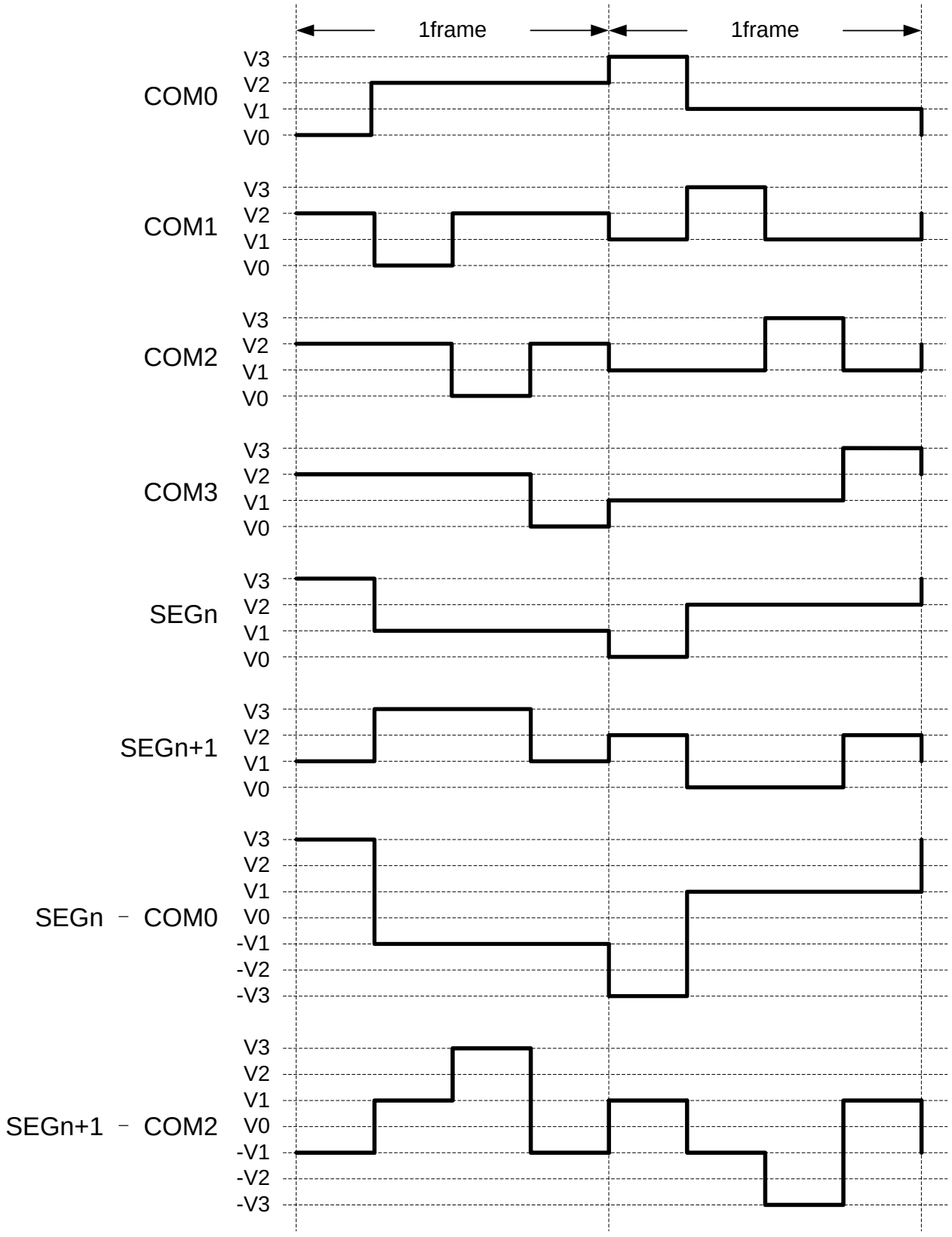


图 37-6 LCD 驱动波形(1/4 duty, 1/3 bias, type B)

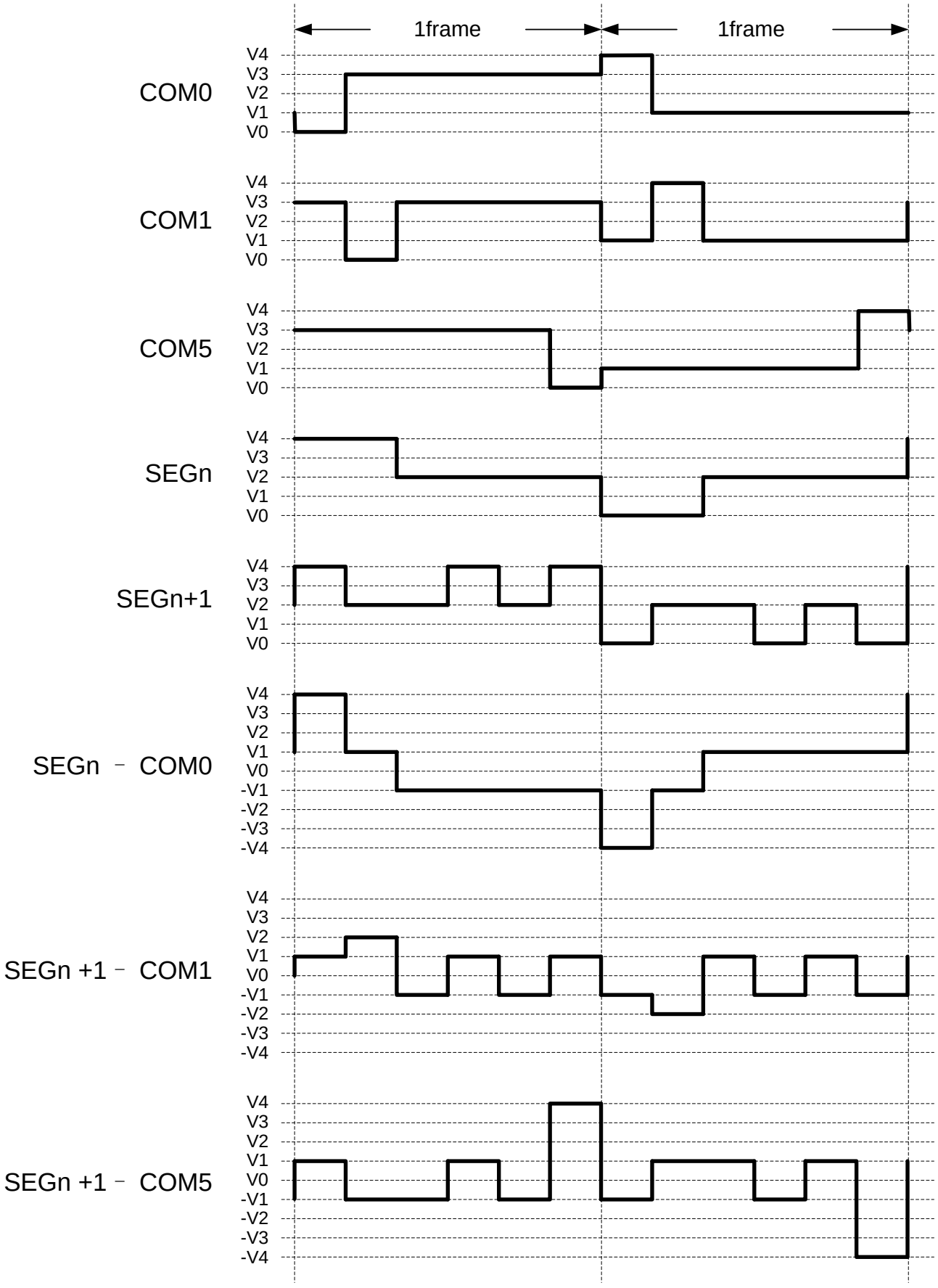


图 37-7 LCD 驱动波形(1/6 duty, 1/4 bias, type B)

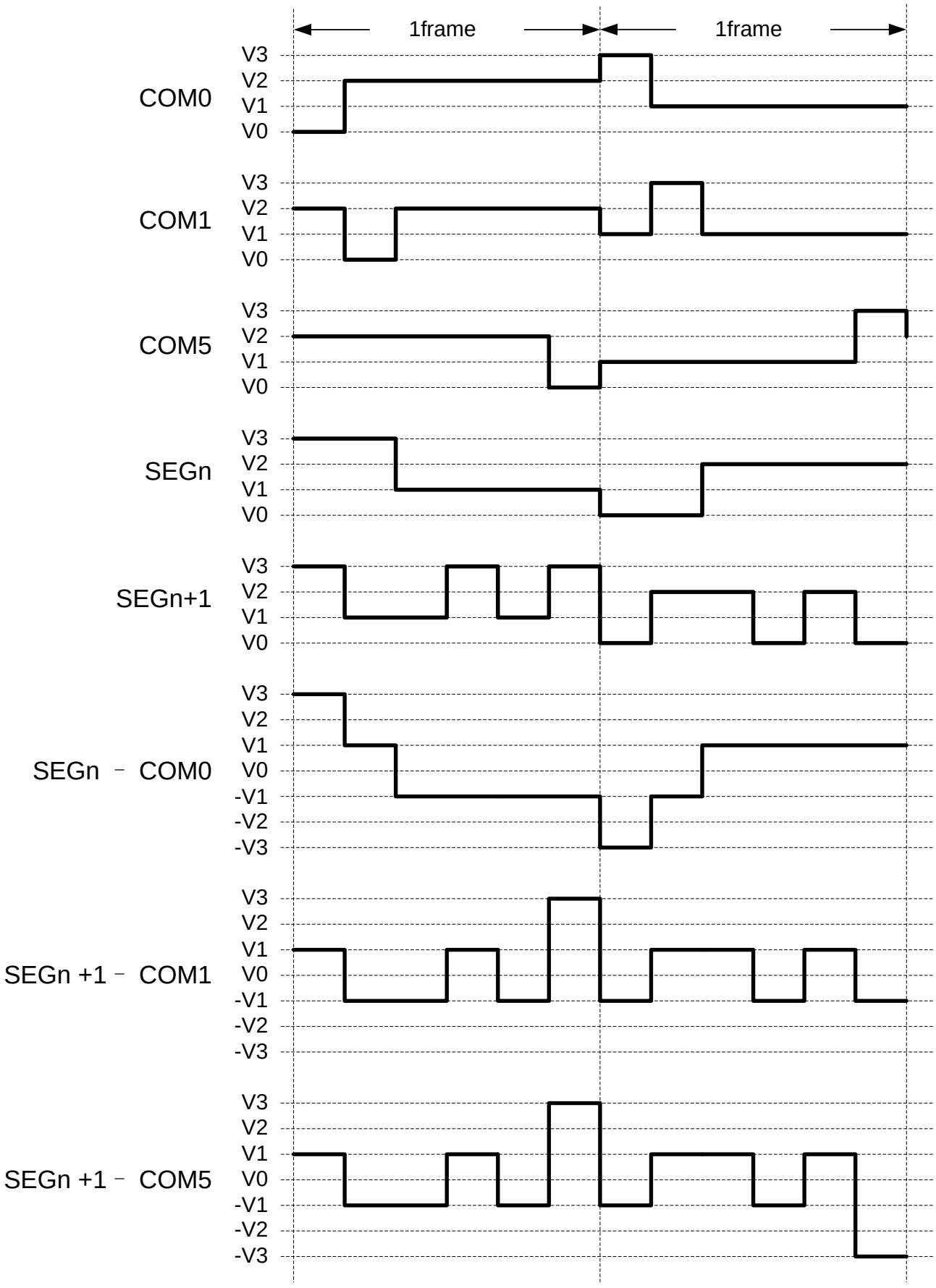


图 37-8 LCD 驱动波形(1/6 duty, 1/3 bias, type B)

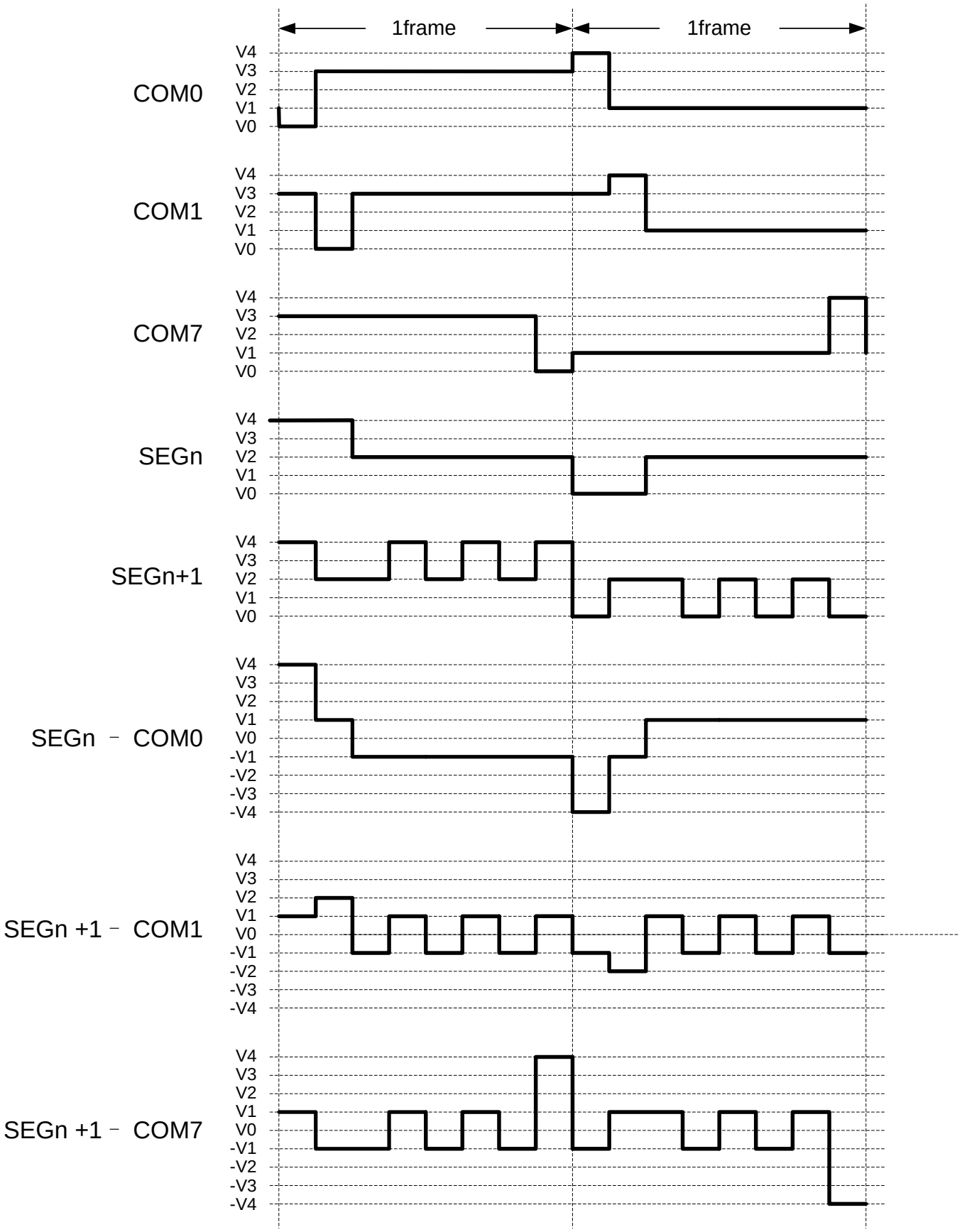


图 37-9 LCD 驱动波形(1/8 duty, 1/4 bias, type B)

37.4.4 片内 buffer 驱动模式

片内buffer驱动模式由电源电压通过分压电阻产生等分电压，分压输入到低功耗buffer以增强驱动能力，buffer输出连接至波形产生模块后产生COM和SEG信号，此模式无需片外器件，功耗较低。

其结构示意图如下：

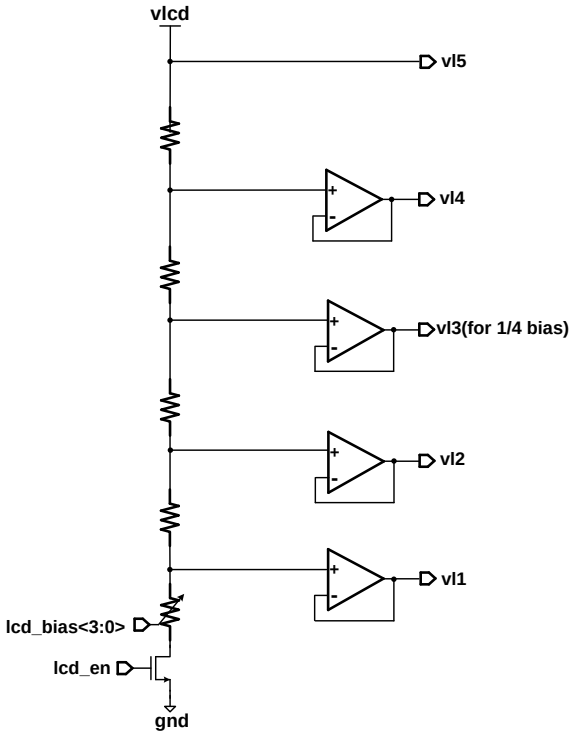


图 37-10LCD 片内电阻 buffer 型驱动电路

驱动Buffer的输出阻抗大约为5Kohm。

通过配置LCDBIAS寄存器，可以实现对LCD驱动电压的调整，具体参见37.4.7“偏置电压调整”章节。

37.4.5 片外电容驱动模式

片外电容驱动模式由开关配合电容产生驱动电平，电容由片外器件提供，需要在VCIN1和VCIN2引脚之间、VDISP0/1/2/3引脚到地之间分别接入0.1uF电容，通过片外电容和片内开关切换实现等分电源电压的输出。片外电容驱动模式具有比buffer驱动模式更低的功耗，但是需要片外电容配合使用。

使用片外电容驱动模式时，需要将LCD驱动模式寄存器LCDDRV中的ENMODE设置为1，并通过SC_CTRL和SCFSEL配置驱动次数和驱动频率。一般而言，选择较高的驱动频率和较多的驱动次数，将获得更好的显示效果，但是相应的功耗也会增加。

注意：使用片外电容模式时，必须将VDISPx和VCINx所在引脚配置为模拟功能（FCR=11），并且VDISPx所在引脚的PxANEN寄存器必须置位，参见41.9.10GPIO模拟开关使能寄存器。

电容驱动模式的详细使用和配置方法，请咨询复旦微电子的应用笔记和例程。

37.4.6 显示闪烁功能

软件可以设置显示控制寄存器 DISPCTRL 中的 FLICKER 位为 1，来使能显示闪烁。FLICKER 使能后，根据 TON 和 TOFF 寄存器值确定闪烁频率。在使能 FLICKER 功能之前应先设置 TON/TOFF 并设置 MD 打开显示，若不设置 TON/TOFF，则其复位值为 0，显示会以 64Hz 闪烁。若不先打开显示，FLICKER 设置无效，不会有显示。

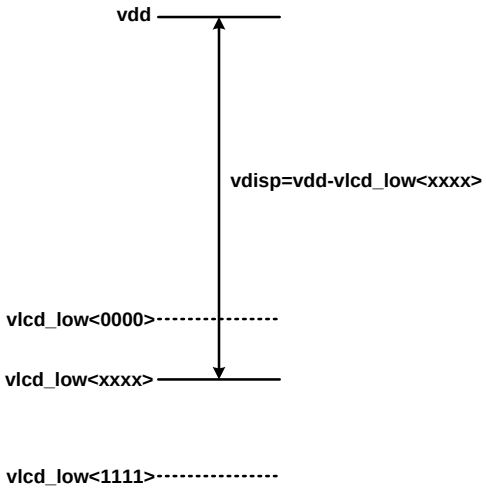
TON/TOFF 最小步长为 T_{step} = COM*DF[7:0]*2*16/32768Hz，实际 ON/OFF 时间为 TON/TOFF * T_{step}。显示和熄灭与帧扫描同步，即在一帧扫描完后熄灭，或在一帧开始时点亮，熄灭或点亮后给出相应中断。由于帧结束信号是 64HZ 的，因此 TON/TOFF 的计数值应为寄存器设置值 x16。

37.4.7 偏置电压调整

LCD 输出的显示电压范围可以调节以适应不同规格的液晶面板，输出电压范围可以表示为：

VDISP=VDD-VLCD_LOW

其中 VLCD_LOW 可以由 LCDBIAS[3:0]调节，LCDBIAS=0000 对应的 VLCD_LOW 电压最高，输出电压范围 VDISP=VDD-VLCD_LOW 最小；LCDBIAS=1111 对应的 VLCD_LOW 电压最低，输出电压范围 VDISP=VDD-VLCD_LOW 最大，如下图所示：



应用中应根据实际 LCD 面板特性，选择合适的 VDISP 电压，可参考 37.6.12 中的表格进行配置。

37.5 低功耗模式

当芯片进入低功耗模式时，LCD 仍然可以正常工作。当主电源掉电时，LCD 停止工作。

模式	说明
LPRUN	正常显示
Sleep/DeepSleep	正常显示
VBAT	不能工作

表 37-3 LCD 与低功耗模式

37.6 寄存器

offset 地址	名称	符号
LCD(模块基地址：0x40010C00)		
0x00000000	显示控制寄存器 （LCD Control Register）	LCD_CR
0x00000004	显示测试控制寄存器 （LCD test Register）	LCD_TEST
0x00000008	显示频率控制寄存器 （LCD Frequency Control Register）	LCD_FCR
0x0000000C	闪烁时间寄存器 （LCD Flick Time Register）	LCD_FLKT
0x00000014	显示中断使能寄存器 （LCD Interrupt Enable Register）	LCD_IER
0x00000018	显示中断标志寄存器 （LCD Interrupt Status Register）	LCD_ISR
0x00000024	显示数据缓存寄存器 0 （LCD data buffer registers 0）	LCD_DATA0
0x00000028	显示数据缓存寄存器 1 （LCD data buffer registers 1）	LCD_DATA1
0x0000002C	显示数据缓存寄存器 2 （LCD data buffer registers 2）	LCD_DATA2
0x00000030	显示数据缓存寄存器 3 （LCD data buffer registers 3）	LCD_DATA3
0x00000034	显示数据缓存寄存器 4 （LCD data buffer registers 4）	LCD_DATA4
0x00000038	显示数据缓存寄存器 5 （LCD data buffer registers 5）	LCD_DATA5
0x0000003C	显示数据缓存寄存器 6 （LCD data buffer registers 6）	LCD_DATA6
0x00000040	显示数据缓存寄存器 7 （LCD data buffer registers 7）	LCD_DATA7
0x00000044	显示数据缓存寄存器 8 （LCD data buffer registers 8）	LCD_DATA8
0x00000048	显示数据缓存寄存器 9 （LCD data buffer registers 9）	LCD_DATA9
0x00000050	COM 使能控制寄存器 （LCD COM Enable Register）	LCD_COMEN
0x00000054	SEG 使能控制寄存器 0 （LCD SEG Enable Register0）	LCD_SEGEN0
0x00000058	SEG 使能控制寄存器 1 （LCD SEG Enable Register 1）	LCD_SEGEN1

37.6.1 显示控制寄存器（LCD_CR）

名称	LCD_CR							
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		SCFSEL			SC_CTRL		IC_CTRL	
位权限		R/W-000			R/W-00		R/W-01	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ENMODE	FLICK	-		LCDBIAS			
位权限	R/W-1	R/W-0	U-0		R/W-1110			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		BIASMD	ANTIPOLAR	WFT	LMUX		LCDEN
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-00		R/W-0

位号	助记符	功能描述
30:23	--	RFU：未实现，读为 0
22:20	**SCFSEL**	电容驱动频率选择 000 = 频率为帧频*COM 数 001 =频率为 LSCLK/8 010 =频率为 LSCLK/16 011 =频率为 LSCLK/32 100 =频率为 LSCLK/64 101 =频率为 LSCLK/128 110 =频率为 LSCLK/256 111=频率为 LSCLK /512 注意：当选择 110 或 111 档位时，如果频率低于帧频*2*COM，则输出结果与 000 档位相同；当帧频率设置较高时，使用 000 档位可能导致驱动脉冲频率太高无法正常输出。
19:18	**SC_CTRL**	片外电容驱动模式下，驱动方式控制 00 = 单次驱动 01 = 连续驱动 2 次 10 = 连续驱动 4 次，当 SC 频率大于等于 4KHz 时，此选项也为多次驱动 11 = 多次驱动
17:16	**IC_CTRL**	偏置电路输入电流源大小控制 (Input bias Current Control) 00 = 电流最大 01 = 电流次大 10 = 电流次小 11 = 电流最小
15	**ENMODE**	驱动模式选择 (LCD Enabling Mode) 0= 片外电容驱动 1 = 片内电阻型驱动
14	**FLICK**	显示闪烁使能位 (LCD Flick Enable) 1：显示闪烁，闪烁频率由 TON 和 TOFF 寄存器设置 0：关闭闪烁
13:12	**--**	RFU：未实现，读为 0
11:8	**LCDBIAS**	LCD 偏置电平选择位，用于显示灰度控制 (LCD Bias Voltage Select)
7:6	**--**	RFU：未实现，读为 0
5	**BIASMD**	偏置类型选择 (Bias Mode) 1：1/3 Bias

位号	助记符	功能描述
		0：1/4 Bias
4	**ANTIPOLAR**	防极化使能 (Anti-Polarization) 1：COM 和 SEG 在 LCD 关闭情况下接地 0：COM 和 SEG 在 LCD 关闭情况下浮空
3	**WFT**	驱动波形选择 (Waveform Format) 1：B类波形 0：A类波形
2:1	**LMUX**	COM 数量选择 (Segment Line Mux) 00：4COM 01：6COM 10/11：8COM
0	**LCDEN**	LCD 显示使能位 (LCD Enable) 1：启动 LCD 显示 0：关闭 LCD 显示

LCDBIAS[3:0]	**不同 VDD 下内部偏置电压(V)**			
	5	**4.5**	**3.6**	**3.0**
0000	2.74	2.47	1.97	1.64
0001	2.83	2.54	2.03	1.69
0010	2.92	2.62	2.10	1.75
0011	3.01	2.71	2.17	1.81
0100	3.12	2.80	2.24	1.87
0101	3.23	2.90	2.32	1.94
0110	3.35	3.01	2.41	2.01
0111	3.47	3.13	2.50	2.08
1000	3.61	3.25	2.60	2.17
1001	3.76	3.39	2.71	2.26
1010	3.93	3.53	2.83	2.35
1011	4.10	3.69	2.95	2.46
1100	4.30	3.87	3.09	2.58
1101	4.51	4.06	3.25	2.71
1110	4.75	4.27	3.42	2.85
1111	5.00	4.50	3.60	3.00

37.6.2 显示测试控制寄存器（LCD_TEST）

名称	**LCD_TEST**							
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	LCCTRL	-						TESTEN
位权限	R/W-0	U-0						R/W-0

位号	位名	说明
31:8	**---**	未实现，读为0
7	**LCCTRL**	LCD测试控制位，仅在测试模式下有效 (Line Constant Control) COM、SEG 输出电平由测试模式下的引脚输出数据寄存器决定。不同设置下 SEG 或 COM 输出的结果参见后文表格。
6:0	**---**	未实现，读为0
0	**TESTEN**	测试模式使能位 (Test mode Enable) 1 = LCD 测试模式使能。在 LCD 测试模式下，LCD 引脚静态输出模拟直流电平，所有与动态扫描时间以及扫描波形相关寄存器设置无效 0= 正常工作模式，测试模式无效，相关测试寄存器控制无效

测试模式下引脚输出电平：

LCCTRL	DISPDATA	COM 引脚输出电平	SEG 引脚输出电平
		1/3bias	1/3bias
0	0	V3	V2
0	1	V1	V4
1	0	V2	V3
1	1	V4	V1

37.6.3 显示频率控制寄存器（LCD_FCR）

名称	LCD_FCR							
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	DF[7:0]							
位权限	R/W-00000000							

位号	位名	说明
31:8	**---**	未实现，读为0
7:0	**DF**	显示预分频寄存器 (Display Frequency)

37.6.4 闪烁时间寄存器（LCD_FLKT）

名称	LCD_FLKT							
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	TOFF[7:0]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TON[7:0]							
位权限	R/W-00000000							

位号	位名	说明
31:16	---	**未实现，读为0**
15:8	**TOFF**	闪烁显示熄灭时间寄存器 (Display-Off Time) TOFF最小步长为T_{step} = COM*DF[7:0]*2*16/32768Hz，实际OFF时间为TOFF * T_{step}
7:0	**TON**	闪烁显示点亮时间寄存器 (Display-On Time) TON最小步长为T_{step} = COM*DF[7:0]*2*16/32768Hz，实际ON时间为TON * T_{step}

37.6.5 显示中断使能寄存器（LCD_IER）

名称	LCD_IER							
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	-						DONIE	DOFFIE
位权限	U-0						R/W-0	R/W-0

位号	位名	说明
31:2	-	未实现，读为0
1	**DONIE**	显示点亮中断使能位 (Display-On Interrupt Enable) 1：显示点亮中断使能 0：显示点亮中断禁止

位号	位名	说明
0	**DOFFIE**	显示熄灭中断使能位 (Display-OFF Interrupt Enable) 1：显示熄灭中断使能 0：显示熄灭中断禁止

37.6.6 显示中断标志寄存器（LCD_ISR）

名称	LCD_ISR							
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	-						DONF	DOFFIF
位权限	U-0						R/W-0	R/W-0

位号	位名	说明
31:2	**-**	未实现，读为0
1	**DONIF**	显示点亮中断标志 (Display-On Interrupt Flag,write 1 to clear) 显示由灭变亮时硬件产生中断标志，硬件置位，软件清零
0	**DOFFIF**	显示熄灭中断标志 (Display-OFF Interrupt Flag,write 1 to clear) 显示由亮变灭时硬件产生中断标志，硬件置位，软件清零

37.6.7 显示数据寄存器（LCD_DATAx）

LCD 显示模块内有 8 个 32 bit 的显示数据寄存器。均为可读可写，复位值为 0。

名称	LCD_DATAx(x=0,1,2,3,4,5,6,7,8,9)							
Offset	0x00000024 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DSDA[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DSDA[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DSDA[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DSDA[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	DSDA	LCD 显示数据 (Display Data)

37.6.7.1 4COM 显示数据寄存器

名称	4com 显示数据寄存器							
LCD_DATA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM0	SEG30 COM0	SEG29 COM0	SEG28 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0
LCD_DATA 1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM1	SEG30 COM1	SEG29 COM1	SEG28 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1
LCD_DATA 2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM2	SEG30 COM2	SEG29 COM2	SEG28 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
LCD_DATA 3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3

名称	4com 显示数据寄存器							
LCD_DATA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM3	SEG30 COM3	SEG29 COM3	SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
LCD_DATA 4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG39 COM0	SEG38 COM0	SEG37 COM0	SEG36 COM0	SEG35 COM0	SEG34 COM0	SEG33 COM0	SEG32 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG35 COM1	SEG34 COM1	SEG33 COM1	SEG32 COM1	SEG43 COM0	SEG42 COM0	SEG41 COM0	SEG40 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG43 COM1	SEG42 COM1	SEG41 COM1	SEG40 COM1	SEG39 COM1	SEG38 COM1	SEG37 COM1	SEG36 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG39 COM2	SEG38 COM2	SEG37 COM2	SEG36 COM2	SEG35 COM2	SEG34 COM2	SEG33 COM2	SEG32 COM2
LCD_DATA 5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG35 COM3	SEG34 COM3	SEG33 COM3	SEG32 COM3	SEG43 COM2	SEG42 COM2	SEG41 COM2	SEG40 COM2
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG43 COM3	SEG42 COM3	SEG41 COM3	SEG40 COM3	SEG39 COM3	SEG38 COM3	SEG37 COM3	SEG36 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

37.6.7.2 6COM 显示数据寄存器

名称	6com 显示数据寄存器							
LCD_DATA 0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM0	SEG30 COM0	SEG43 COM0	SEG42 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0
LCD_DATA 1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

名称	6com 显示数据寄存器							
LCD_DATA 0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM1	SEG30 COM1	SEG43 COM1	SEG42 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1
LCD_DATA 2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM2	SEG30 COM2	SEG43 COM2	SEG42 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
LCD_DATA 3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM3	SEG30 COM3	SEG43 COM3	SEG42 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
LCD_DATA 4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM4	SEG6 COM4	SEG5 COM4	SEG4 COM4	SEG3 COM4	SEG2 COM4	SEG1 COM4	SEG0 COM4
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM4	SEG14 COM4	SEG13 COM4	SEG12 COM4	SEG11 COM4	SEG10 COM4	SEG9 COM4	SEG8 COM4
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM4	SEG22 COM4	SEG21 COM4	SEG20 COM4	SEG19 COM4	SEG18 COM4	SEG17 COM4	SEG16 COM4
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM4	SEG30 COM4	SEG43 COM4	SEG42 COM4	SEG27 COM4	SEG26 COM4	SEG25 COM4	SEG24 COM4
LCD_DATA 5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM5	SEG6 COM5	SEG5 COM5	SEG4 COM5	SEG3 COM5	SEG2 COM5	SEG1 COM5	SEG0 COM5
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM5	SEG14 COM5	SEG13 COM5	SEG12 COM5	SEG11 COM5	SEG10 COM5	SEG9 COM5	SEG8 COM5
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM5	SEG22 COM5	SEG21 COM5	SEG20 COM5	SEG19 COM5	SEG18 COM5	SEG17 COM5	SEG16 COM5
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

名称	6com 显示数据寄存器							
LCD_DATA 0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG31 COM5	SEG30 COM5	SEG43 COM5	SEG42 COM5	SEG27 COM5	SEG26 COM5	SEG25 COM5	SEG24 COM5
LCD_DATA 6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG39 COM0	SEG38 COM0	SEG37 COM0	SEG36 COM0	SEG35 COM0	SEG34 COM0	SEG33 COM0	SEG32 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG37 COM1	SEG36 COM1	SEG35 COM1	SEG34 COM1	SEG33 COM1	SEG32 COM1	SEG41 COM0	SEG40 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG35 COM2	SEG34 COM2	SEG33 COM2	SEG32 COM2	SEG41 COM1	SEG40 COM1	SEG39 COM1	SEG38 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG33 COM3	SEG32 COM3	SEG41 COM2	SEG40 COM2	SEG39 COM2	SEG38 COM2	SEG37 COM2	SEG36 COM2
LCD_DATA 7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG41 COM3	SEG40 COM3	SEG39 COM3	SEG38 COM3	SEG37 COM3	SEG36 COM3	SEG35 COM3	SEG34 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG39 COM4	SEG38 COM4	SEG37 COM4	SEG36 COM4	SEG35 COM4	SEG34 COM4	SEG33 COM4	SEG32 COM4
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG37 COM5	SEG36 COM5	SEG35 COM5	SEG34 COM5	SEG33 COM5	SEG32 COM5	SEG41 COM4	SEG40 COM4
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
					SEG41 COM5	SEG40 COM5	SEG39 COM5	SEG38 COM5

37.6.7.3 8COM 显示数据寄存器

名称	8com 显示数据寄存器							
LCD_DATA 0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM0	SEG42 COM0	SEG41 COM0	SEG40 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0
LCD_DATA 1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

名称	8com 显示数据寄存器							
LCD_DATA 0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM1	SEG42 COM1	SEG41 COM1	SEG40 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1
LCD_DATA 2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM2	SEG42 COM2	SEG41 COM2	SEG40 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
LCD_DATA 3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM3	SEG42 COM3	SEG41 COM3	SEG40 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
LCD_DATA 4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM4	SEG6 COM4	SEG5 COM4	SEG4 COM4	SEG3 COM4	SEG2 COM4	SEG1 COM4	SEG0 COM4
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM4	SEG14 COM4	SEG13 COM4	SEG12 COM4	SEG11 COM4	SEG10 COM4	SEG9 COM4	SEG8 COM4
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM4	SEG22 COM4	SEG21 COM4	SEG20 COM4	SEG19 COM4	SEG18 COM4	SEG17 COM4	SEG16 COM4
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM4	SEG42 COM4	SEG41 COM4	SEG40 COM4	SEG27 COM4	SEG26 COM4	SEG25 COM4	SEG24 COM4
LCD_DATA 5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM5	SEG6 COM5	SEG5 COM5	SEG4 COM5	SEG3 COM5	SEG2 COM5	SEG1 COM5	SEG0 COM5
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM5	SEG14 COM5	SEG13 COM5	SEG12 COM5	SEG11 COM5	SEG10 COM5	SEG9 COM5	SEG8 COM5
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM5	SEG22 COM5	SEG21 COM5	SEG20 COM5	SEG19 COM5	SEG18 COM5	SEG17 COM5	SEG16 COM5
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

名称	8com 显示数据寄存器							
LCD_DATA 0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG43 COM5	SEG42 COM5	SEG41 COM5	SEG40 COM5	SEG27 COM5	SEG26 COM5	SEG25 COM5	SEG24 COM5
LCD_DATA 6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM6	SEG6 COM6	SEG5 COM6	SEG4 COM6	SEG3 COM6	SEG2 COM6	SEG1 COM6	SEG0 COM6
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM6	SEG14 COM6	SEG13 COM6	SEG12 COM6	SEG11 COM6	SEG10 COM6	SEG9 COM6	SEG8 COM6
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM6	SEG22 COM6	SEG21 COM6	SEG20 COM6	SEG19 COM6	SEG18 COM6	SEG17 COM6	SEG16 COM6
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM6	SEG42 COM6	SEG41 COM6	SEG40 COM6	SEG27 COM6	SEG26 COM6	SEG25 COM6	SEG24 COM6
LCD_DATA 7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM7	SEG6 COM7	SEG5 COM7	SEG4 COM7	SEG3 COM7	SEG2 COM7	SEG1 COM7	SEG0 COM7
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM7	SEG14 COM7	SEG13 COM7	SEG12 COM7	SEG11 COM7	SEG10 COM7	SEG9 COM7	SEG8 COM7
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM7	SEG22 COM7	SEG21 COM7	SEG20 COM7	SEG19 COM7	SEG18 COM7	SEG17 COM7	SEG16 COM7
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM7	SEG30 COM7	SEG29 COM7	SEG28 COM7	SEG27 COM7	SEG26 COM7	SEG25 COM7	SEG24 COM7
LCD_DATA 8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG39 COM0	SEG38 COM0	SEG37 COM0	SEG36 COM0	SEG35 COM0	SEG34 COM0	SEG33 COM0	SEG32 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG39 COM1	SEG38 COM1	SEG37 COM1	SEG36 COM1	SEG35 COM1	SEG34 COM1	SEG33 COM1	SEG32 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG39 COM2	SEG38 COM2	SEG37 COM2	SEG36 COM2	SEG35 COM2	SEG34 COM2	SEG33 COM2	SEG32 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG39 COM3	SEG38 COM3	SEG37 COM3	SEG36 COM3	SEG35 COM3	SEG34 COM3	SEG33 COM3	SEG32 COM3
LCD_DATA 9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG39 COM4	SEG38 COM4	SEG37 COM4	SEG36 COM4	SEG35 COM4	SEG34 COM4	SEG33 COM4	SEG32 COM4
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG39 COM5	SEG38 COM5	SEG37 COM5	SEG36 COM5	SEG35 COM5	SEG34 COM5	SEG33 COM5	SEG32 COM5
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG39 COM6	SEG38 COM6	SEG37 COM6	SEG36 COM6	SEG35 COM6	SEG34 COM6	SEG33 COM6	SEG32 COM6
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG39 COM7	SEG38 COM7	SEG37 COM7	SEG36 COM7	SEG35 COM7	SEG34 COM7	SEG33 COM7	SEG32 COM7

37.6.8 LCD COM 使能控制寄存器（LCD_COMEN）

名称	LCD_COMEN							
Offset	0x00000050							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	-				COMEN[3:0]			
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	COMEN	LCD COM 输出使能控制 (COM Enable) 1：COM 输出使能 0：COM 输出禁止

37.6.9 LCD SEG 使能控制寄存器 0（LCD_SEGEN0）

名称	LCD_SEGEN0							
Offset	0x00000054							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SEGEN31	SEGEN30	SEGEN29	SEGEN28	SEGEN27	SEGEN26	SEGEN25	SEGEN24
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	SEGEN23	SEGEN22	SEGEN21	SEGEN20	SEGEN19	SEGEN18	SEGEN17	SEGEN16
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	SEGEN15	SEGEN14	SEGEN13	SEGEN12	SEGEN11	SEGEN10	SEGEN9	SEGEN8
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	SEGEN7	SEGEN6	SEGEN5	SEGEN4	SEGEN3	SEGEN2	SEGEN1	SEGEN0
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
31:0	SEGENx	LCD SEG 输出使能控制， 1: SEG 输出使能 0: SEG 输出禁止 *注：SEGEN28~31 同时控制 COM4~7 使能*

37.6.10 LCD SEG 使能控制寄存器 1（LCD_SEGEN1）

名称	LCD_SEGEN1							
Offset	0x00000058							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	-	-	-	SEGEN 43	SEGEN 42	SEGEN 41	SEGEN 40
位权限	U-0	U-0	U-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	SEGEN39	SEGEN 38	SEGEN 37	SEGEN 36	SEGEN 35	SEGEN 34	SEGEN 33	SEGEN 32
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
31:12	--	RFU：未实现，读为 0
11:0	SEGENx	LCD SEG 输出使能控制， 1: SEG 输出使能 0: SEG 输出禁止

38 模数转换器（ADC）

38.1 概述

FM33LG0 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 1.6～5.5V
- 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 20 个外部输入通道：
  - 14 个快速通道，可以组成 7 个差分输入对
  - 6 个慢速通道，仅支持单端输入
  - 慢速通道内建缓冲器，可用于弱驱动信号测量
- 7 个内部采样通道
  - 温度传感器
  - 内部基准测量
  - VBAT/3
  - VDD/3
  - DAC 输出
  - OPA 输出
- 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

38.2 结构框图

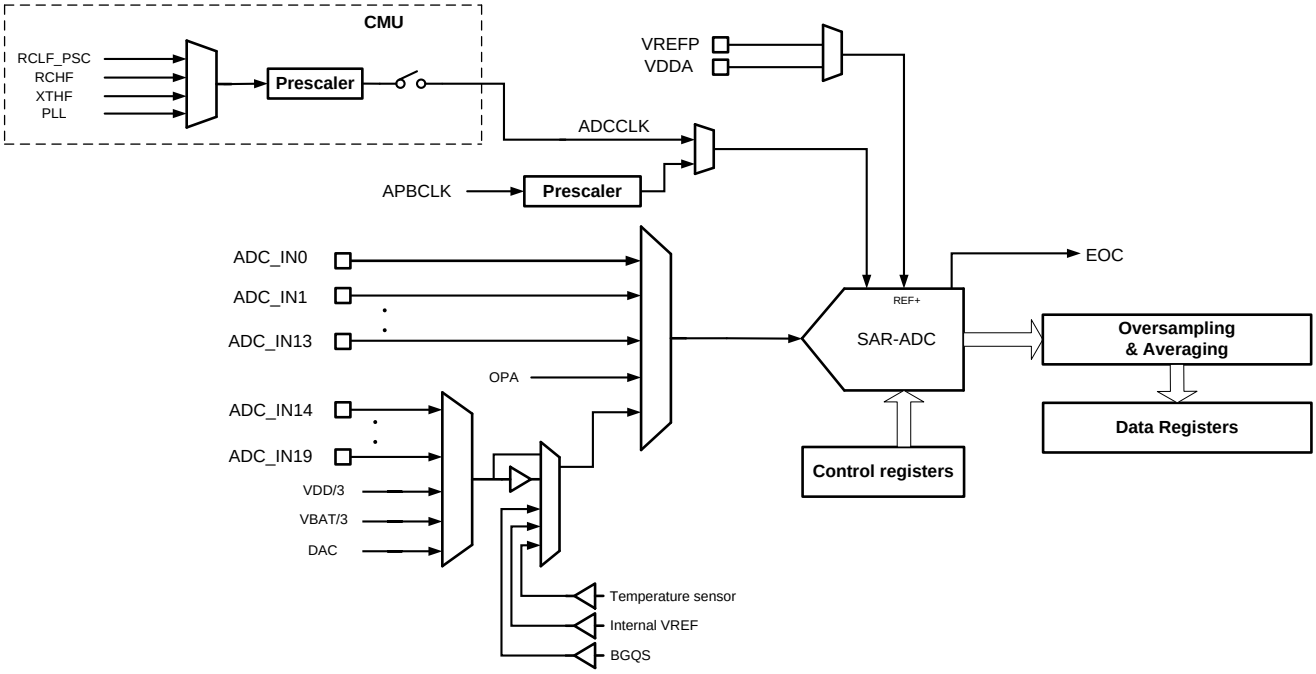


图 38-1 ADC 结构框图

38.3 输入通道

ADC 最大支持 7 个内部通道和 20 个外部通道。

通道	IO	说明
ADC_IN0	PD11	外部快速通道，单端或差分输入 其中差分对组合关系如下： ADC_IN0 – ADC_IN7 ADC_IN1 – ADC_IN8 ADC_IN2 – ADC_IN9 ADC_IN3 – ADC_IN10 ADC_IN4 – ADC_IN11 ADC_IN5 – ADC_IN12 ADC_IN6 – ADC_IN13
ADC_IN1	PD1	
ADC_IN2	PD3	
ADC_IN3	PD5	
ADC_IN4	PA13	
ADC_IN5	PA0	
ADC_IN6	PC7	
ADC_IN7	PD0	
ADC_IN8	PD2	
ADC_IN9	PD4	
ADC_IN10	PD6	
ADC_IN11	PA14	
ADC_IN12	PA1	
ADC_IN13	PC8	
ADC_IN14	PC9	外部慢速通道，单端输入
ADC_IN15	PC10	
ADC_IN16	PC11	
ADC_IN17	PC12	
ADC_IN18	PD10	
ADC_IN19	PE9	
VBAT/3	N/A	VBAT 采样通道
VDD/3		VDD 采样通道
VREF1p2		内部 1.2V 基准源采样通道
TS		温度传感器采样通道
AVREF		内部 1.0V 基准源采样通道
DAC		DAC 输出采样通道
OPA		运放输出采样通道

表 38-1 ADC 输入通道

注意并非所有封装形式都能支持到最大通道数量，产品型号与 ADC 通道数量的对应关系参见下表。

产品型号	封装	外部通道	内部通道
FM33LG0x8	LQFP80	20	7
FM33LG0x6V	LQFP64	18	7
FM33LG0x6R	LQFP64	18	7

38.4 单端和差分输入

ADC 支持单端输入和差分输入模式。

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+。

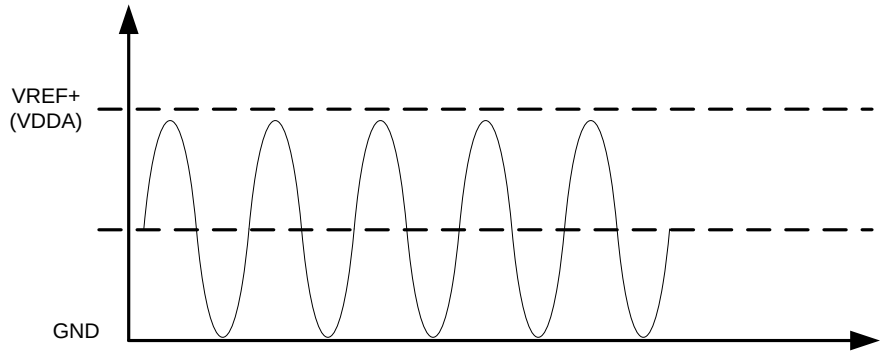


图 38-2 单端输入

差分输入模式下，ADC 转换的是差分输入引脚对 VIN+和 VIN-之间的差值，输入信号为(VIN+) – (VIN-)，输入范围是-VREF~+VREF。使用差分输入方式，可以获得更好的共模噪声抑制效果，因此当被采样信号源远离 ADC 输入引脚时，推荐使用差分对方式提高信噪比。

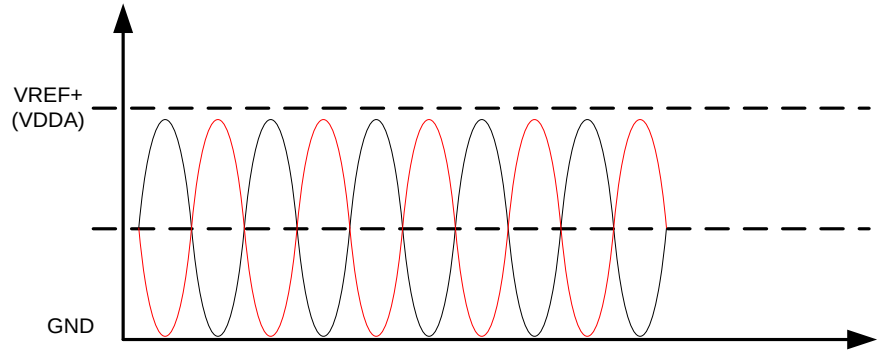


图 38-3 差分输入

自动转换多个通道时，可以实现差分通道和单端通道的交织。举个简单的例子，如果希望连续采样 ADC_IN0 和 ADC_IN7 组成的差分通道，以及 ADC_IN2 的单端通道，则软件应采取如下配置方法：

- 配置 ADC_CHER 寄存器，ECH0, ECH2 置 1
- 配置 ADC_DCR 寄存器，AINS[0]置 1
- 选择单次转换或连续转换
- 启动 ADC 开始转换

ADC 启动后，ADC_IN0 和 ADC_IN7 通道同时使能，ADC 对差分输入信号进行采样转换；完成后关闭 ADC_IN0 和 ADC_IN7 通道，使能 ADC_IN2 通道，ADC 对单端输入信号进行采样转换。

注意：差分对工作时仅需使能低位编号的通道即可，比如上面的例子中差分对 IN0 和 IN7 仅需使能 ADC_IN0 通道，并置位 AINS[0]

38.5 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 64 个 ADCCLK 周期。校准完成后置位 EOCAL 标志寄存器。

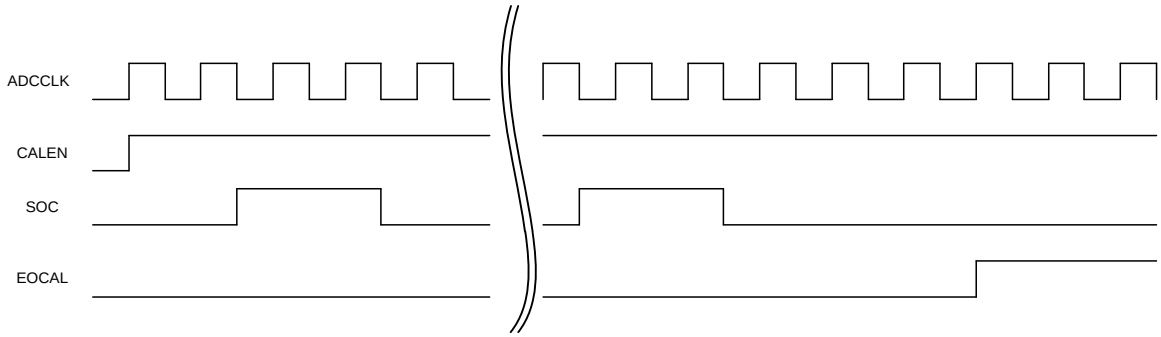


图 38-4 ADC 校准时序

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。ADC 采样转换完成后产生 EOCONV（End of Conversion）信号，ADC 控制器在采样 EOCONV 后置位 EOC 标志寄存器，并且可以根据配置产生中断事件或 DMA 请求。

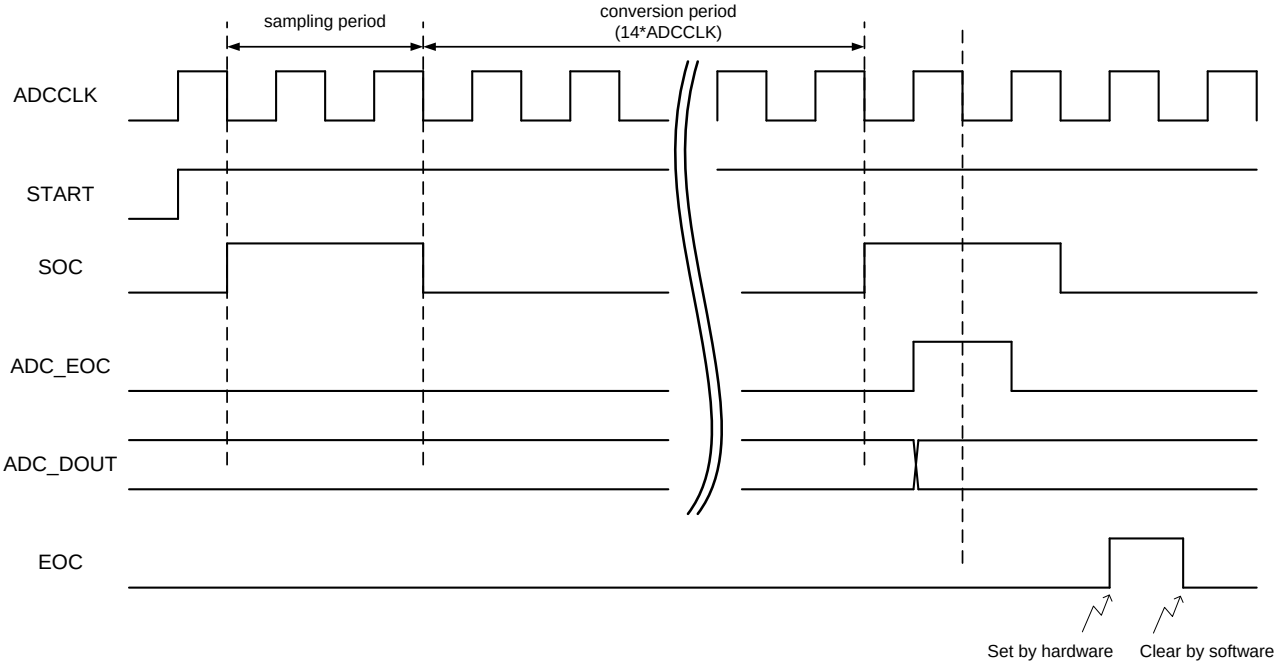


图 38-5 ADC 采样转换时序

多通道转换序列的时序示意图如下。ADC 对所有别使能的输入通道依次完成一轮采样转换，成为一个转换序列。当所有使能的输入通道都被依次采样转换过后，ADC 控制器置位 EOS（End of Sequence）标志寄存器，并可以根据配置产生中断事件。

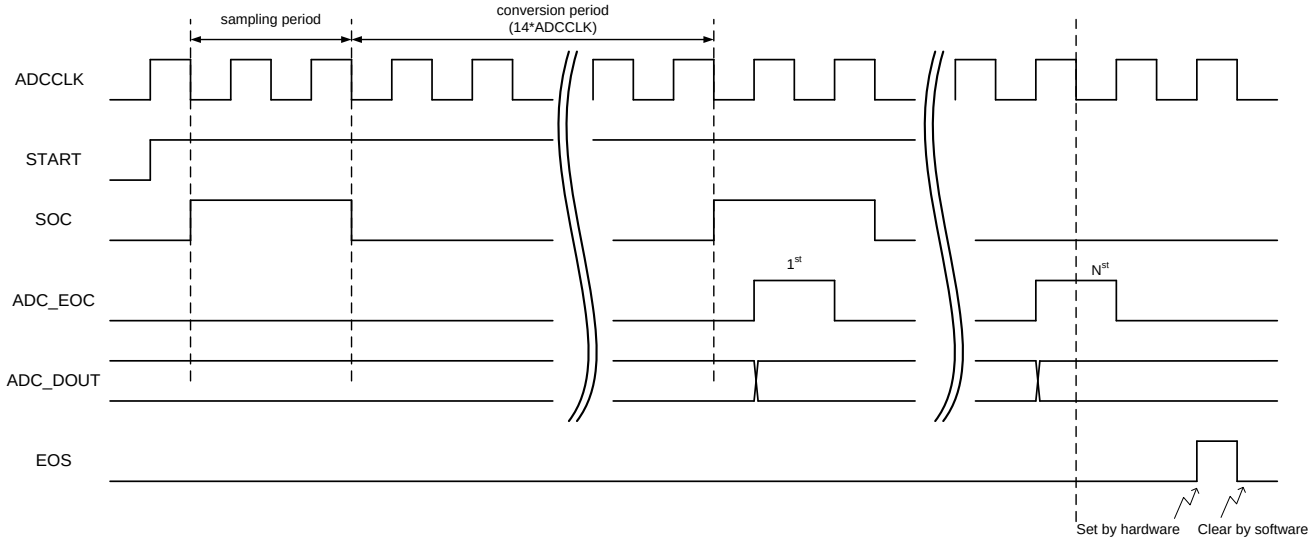


图 38-6 ADC 采样序列时序

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。

38.6 功能描述

38.6.1 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。

使用高精度内部基准作为参考信号

- 首先芯片 Flash 中保存了 30C 环境温度下 VDDA=3.0V 条件下测量 VREF1p2 电压得到的转换值
- 芯片实际应用中，由于不知道当前 VDDA 电压，ADC 先测量 VREF 得到转换值 VREFINT_DATA；通过以下公式可以得到当前实际的 VDDA：

$$VDDA = \frac{VREFINT_CAL}{VREFINT_DATA} \times 3V$$

- 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）

$$V_{CHANNEL} = \frac{VREFINT_CAL \times ADC_DATA}{VREFINT_DATA \times 4095} \times 3V$$

- 采用这个方式，不需要知道每颗芯片 VREF 的实际电压值，仅需计算当前 VREF1p2 采样值和出厂测试值的比例；但是需要保证测试时 VDDA 的电压精度和温度精度，以尽可能减小误差

注意到 VREFIN_CAL 是 3V 参考电压下 1.22V 的转换值，那么这个值应该在 1665 附近，即 11bit 无符号有效值，ADC_DATA 是 12bit 无符号数，VREFINT_CAL*ADC_DATA*3 最大为 25bit；VREFINT_DATA 最大转换值为 1.8V 下的 1.22V 即 2775 左右，相当于 12bit 无符号数，所以 VREFINT_DATA*4095 最大为 24bit，通过 32bit 除法可以实现计算。

使用快速内部基准作为参考信号

芯片内部还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%。相对 VREF1p2，快速基准源的启动速度很快，典型条件下可以在 5us 内完成启动和输出电压建立，而 VREF1p2 需要 1ms 以上。但是快速基准源的精度受温度影响相对较大，全温区变化可以达到+/-4%，因此适用于对绝对精度要求不高的场合。

- 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 BGQS_DATA）：

$$VDDA = \frac{4095}{BGQS_DATA} \times 1.0V$$

- 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）

$$V_{CHANNEL} = \frac{ADC_DATA}{BGQS_DATA} \times 1.0V$$

38.6.2 使用 VREFP 作为基准

如果使用 VREFP 作为基准源，并且由内部升压 Buffer 驱动 VREFP 引脚，则 ADC 的工作参考电压是确定的，无需再通过采样内部基准信号来反推 ADC 参考电压。

此时采样外部通道时，输入电压的实际幅度可以通过下式确定：

$$V_{CHANNEL} = \frac{ADC_DATA}{4095} \times VREFP$$

38.6.3 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

$$\text{VPTAT} = \frac{VREFIN_CAL \times TS_DATA}{VREFINT_DATA \times 4095} \times 3V$$

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

$$\text{VPTAT_30C} = \frac{TS_CAL1}{4095} \times 3V$$

3) 根据温度传感器输出斜率计算当前绝对温度

$$\text{Temperature} = \frac{VPTAT - VPTAT_30C}{slope} + 30C$$

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。

Slope 表示温度传感器输出斜率，典型值为 2.5mV/°C，温度传感器输出的详细特性参见 6.3 温度传感器。

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。

温度计算例程：

```
1    int32_t ComputeTemperature(uint32_t measure)
{

      int32_t temperature;
      GPIO_SetBits(GPIOB, GPIO_Pin_1);
      temperature = ((measure * VDD_APPLI / VDD_CALIB)- (int32_t) *(int16_t*) (0x00008000) );
      temperature = temperature * (int32_t)(130 – 30);
      temperature = temperature / (int32_t)(*(int16_t*) ( 0x00008002) -*(int16_t*) (0x00008000));
      GPIO_ResetBits(GPIOB, GPIO_Pin_1);
      return(temperature);
}
```

以上程序温度分辨率为 1℃，主频 8M 是 22us， 16M 是 11.6us，24M 为 7.6us

②int32_t ComputeTemperature(uint32_t measure)

```
{

    int32_t temperature;
    GPIO_SetBits(GPIOB, GPIO_Pin_1);
    temperature = ((measure * VDD_APPLI / VDD_CALIB)- (int32_t) *(int16_t*) (0x00008000) );
    temperature = temperature * (int32_t)(130 – 30);
     temperature = 10*temperature / (int32_t)(*(int16_t*) ( 0x00008002) -*(int16_t*) (0x00008000));
    GPIO_ResetBits(GPIOB, GPIO_Pin_1);
    return(temperature);
}
```

以上程序温度分辨率为 0.1℃，主频 8M 是 26us， 16M 是 13.2us，24M 为 8.5us

38.6.4 温度传感器的斜率和标定

温度传感器工作电压范围为 1.8~5.5V，温度测量范围不小于-40~+85C，跨度 125C。PTAT 输出斜率为 3.06mV/C，在 VDDA=5V 的情况下，分辨率为 1.22mV/LSB，即 2.51LSB/C；在 VDDA=3V 情况下，分辨率为 0.73mV/LSB，即 4.19 LSB/C。

电源电压	VPTAT slope	ADC mV/LSB	LSB/°C
1.8~5.5V	2.5mV/C	1.22@5V	2.05
		0.73@3V	3.42

表 38-2 温度传感器斜率

38.6.5 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
0011	12
0100	16
0101	32
0110	64
0111	80
1000	96
1001	128
1010	160

SMTSx	Sampling cycles (T_{ADCCLK})
1011	192
1100	256
1101	320
1110	384
1111	512

表 38-3 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：

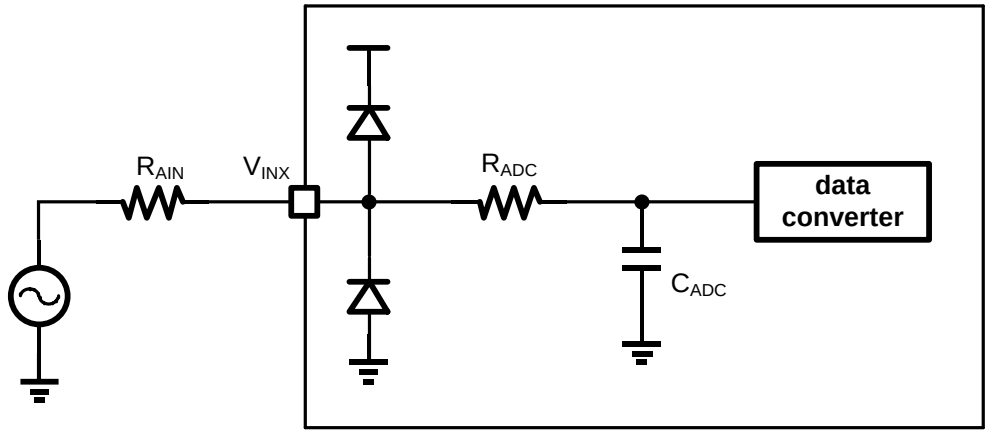


图 38-7 ADC 输入通道示意图

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。

38.6.6 输出位宽选择

SAR-ADC 支持最高 12 位，最低 6 位的输出位宽设置。通过 BITSEL 寄存器可以配置输出数据位宽，以获得精度和速度的折中。

选择不同的输出位宽时，ADC 逐次逼近的转换周期数不同，而与 ADC 采样时间无关。

假设 ADC 采样时间设置为 2*T_{ADCCLK}，在 12bit 精度下，转换时间为 14*T_{ADCCLK}，总的采样转换时间为 16*T_{ADCCLK}；在 10bit 精度下，转换时间相应变为 12*T_{ADCCLK}。

BITSEL	resolution	Conversion time (T_{ADCCLK})	Max speed @ADCCLK=16Mhz
00	12	14	1Msps
01	10	12	1.14Msps
10	8	10	1.33Msps
11	6	8	1.6Msps
BITSEL	**resolution**	**Conversion time (T_{ADCCLK})**	**Max speed @ADCCLK=32Mhz**
00	12	14	2Msps
01	10	12	2.28Msps
10	8	10	2.66Msps
11	6	8	3.2Msps

表 38-4 ADC 输出位宽与速度

38.6.7 输入缓冲器

ADC 低速通道内建了一个模拟输入缓冲器，用于阻抗匹配。当采样驱动很弱的外部信号或内部信号时，建议使能输入缓冲器（置位 BUFEN 寄存器），这样可以减少 ADC 采样时间，获得更为准确的采样结果。

输入缓冲器无法跟随高频交流信号，因此仅用于对直流输入信号的驱动跟随。如果采样交流信号，或者输入信号驱动能力足够强，则必须 bypass 输入缓冲器。

输入缓冲器无法实现轨到轨输入输出，推荐的输入信号范围是 0.1V ~ VREF – 0.1V，工作电压范围是 VDDA=2.0~5.5V。缓冲器支持低功耗模式，正常模式下典型功耗约 40uA，低功耗模式下为 20uA。正常模式下驱动能力可以满足最大 1us 的 ADC 采样时间，而低功耗模式下需要将采样时间延长到 10us。

注意：

1） ADC 高速通道没有内建缓冲器，以确保对高频交流信号的采样能力。因此，不建议使用高速通道采样驱动很弱的直流信号。

2） ADC 测量内部基准 VREF1p2 和温度传感器时，应使用 VREF1p2 和温度传感器自带的输出 buffer，并 bypass ADC 输入 buffer。

38.6.7.1 Buffer offset 的消除

输入缓冲器的 offset 会影响对小信号的采样，为了尽可能消除 offset 的影响，buffer 实现了斩波功能。当置位 BUFCHP_EN 寄存器后，配合斩波功能必须使能硬件过采样以平均掉 offset 误差，过采样次数大于等于 2 次即可。因此使用 buffer 并启动斩波的情况下，采样率相比不启动 chopper 的情况至

少会下降一半。

38.6.8 VBAT 和 VDD 电源电压采样

ADC 可以用于 VBAT 和 VDD 采样，采样前 VBAT 经过电阻分压，得到 VBAT/3 或 VDD/3 电平送入 ADC 进行转换。

当 VBAT/3 或 VDD/3 通道被使能时，分压电阻串自动使能。分压电阻串阻值约 100Kohm，在输入 3V 的情况下，电阻串消耗的电流约 30uA。

分压电阻串输出阻抗较大，ADC 采样时需要使能内部 buffer 以加强驱动。

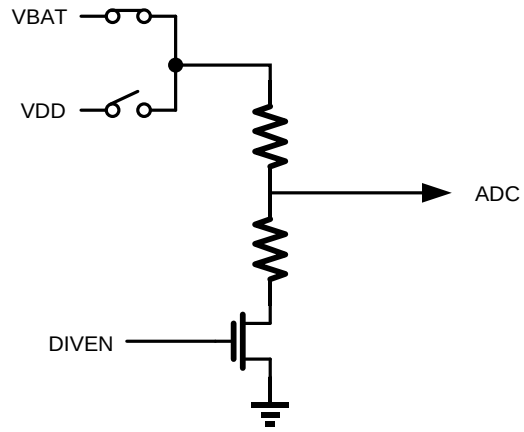


图 38-8 VBAT/VDD 分压电路示意图

38.6.9 内部信号采样

ADC 可以采样以下芯片内部信号：

- DAC 输出
- 温度传感器输出
- 1.2V 内部基准源
- 1.0V 内部基准源
- OPA 输出

其中，1.2V 基准、1.0V 基准、温度传感器都有独立的输出缓冲器，ADC 采样前需要使能这些模拟 buffer 以加强驱动。

芯片内部连接的 DAC 输出没有经过 DAC 内部 buffer 驱动，因此需要在采样时使能 ADC 内部 buffer。

38.6.10 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。

38.6.10.1 单次转换模式

单次转换是指 ADC 完成一个转换序列后自动停止采样（但是 ADC 仍然是保持使能的），等待新的触发事件到来。

单次转换支持半自动触发和全自动触发两种方式。

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

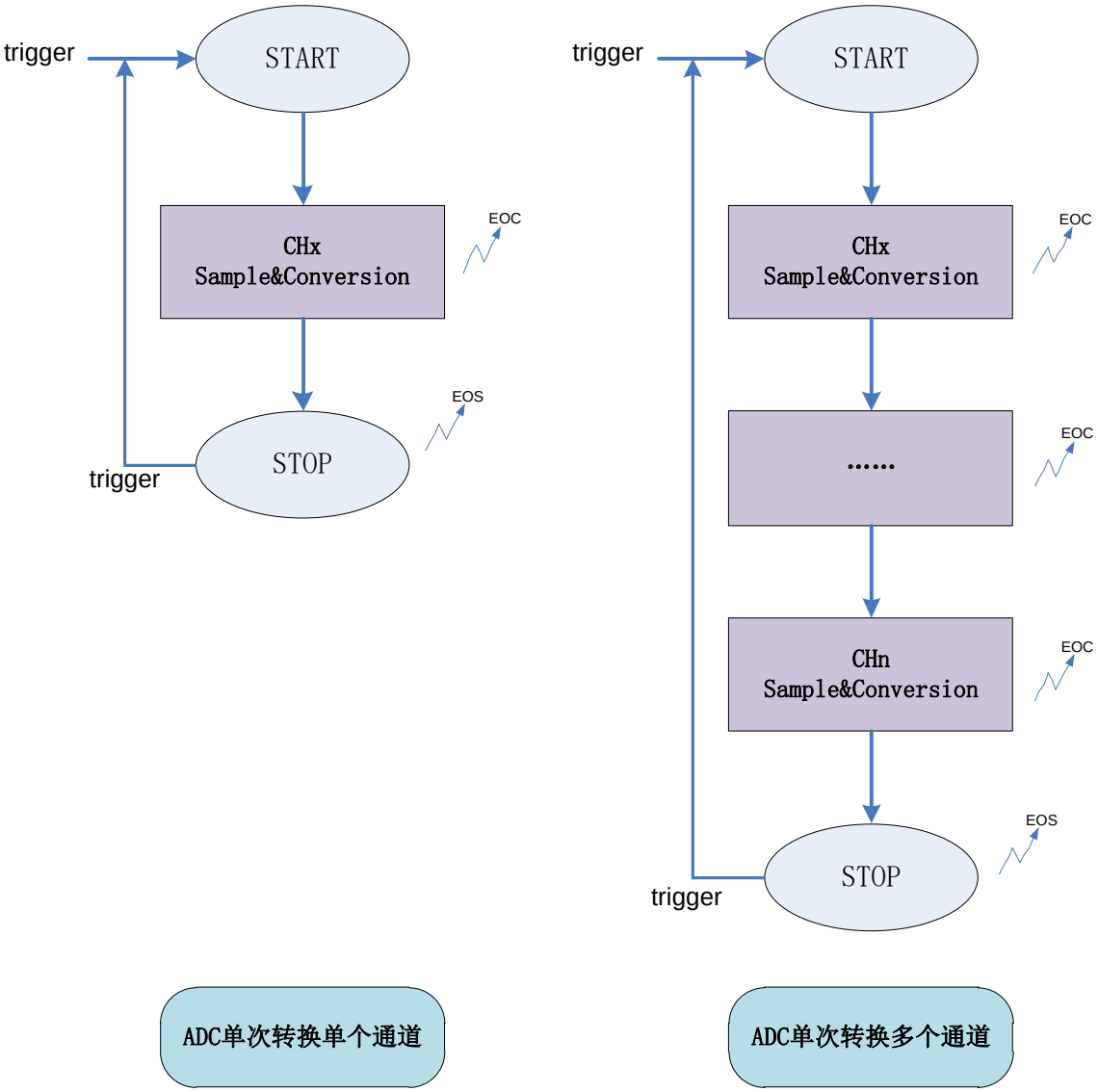


图 38-9 ADC 单次转换全自动触发模式

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

……

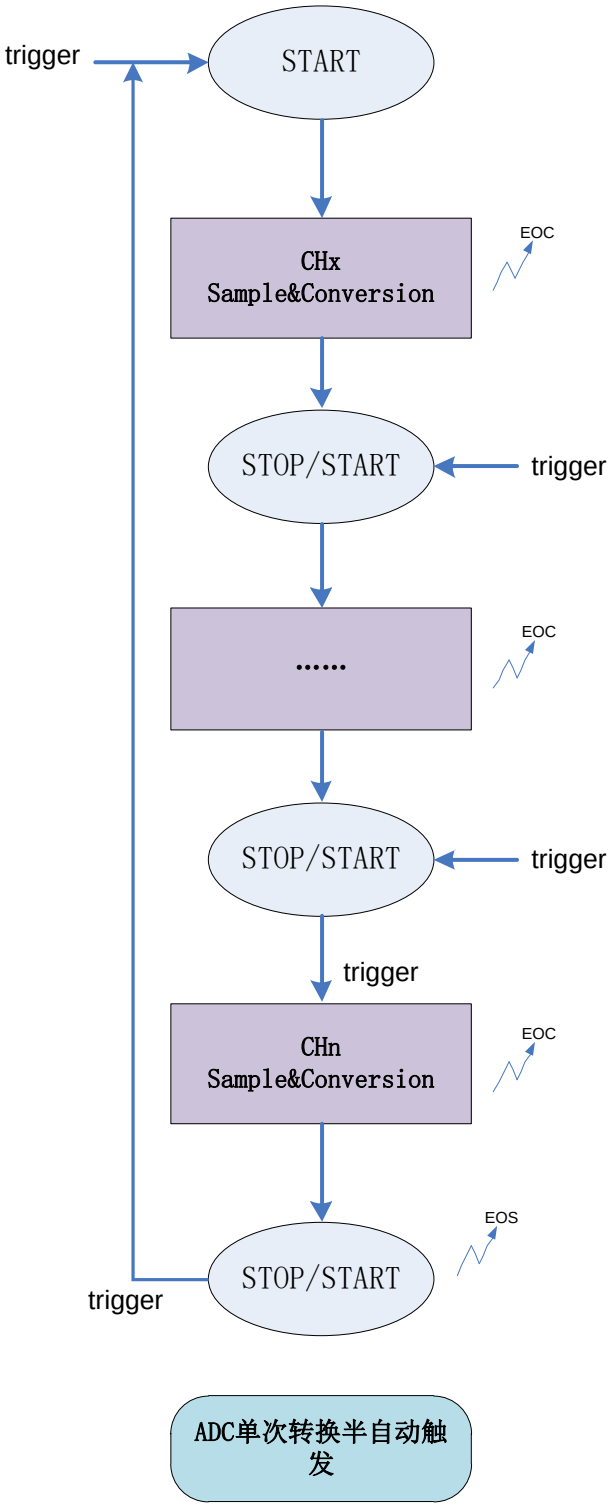


图 38-10 ADC 单次转换半自动触发模式

38.6.10.2 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。

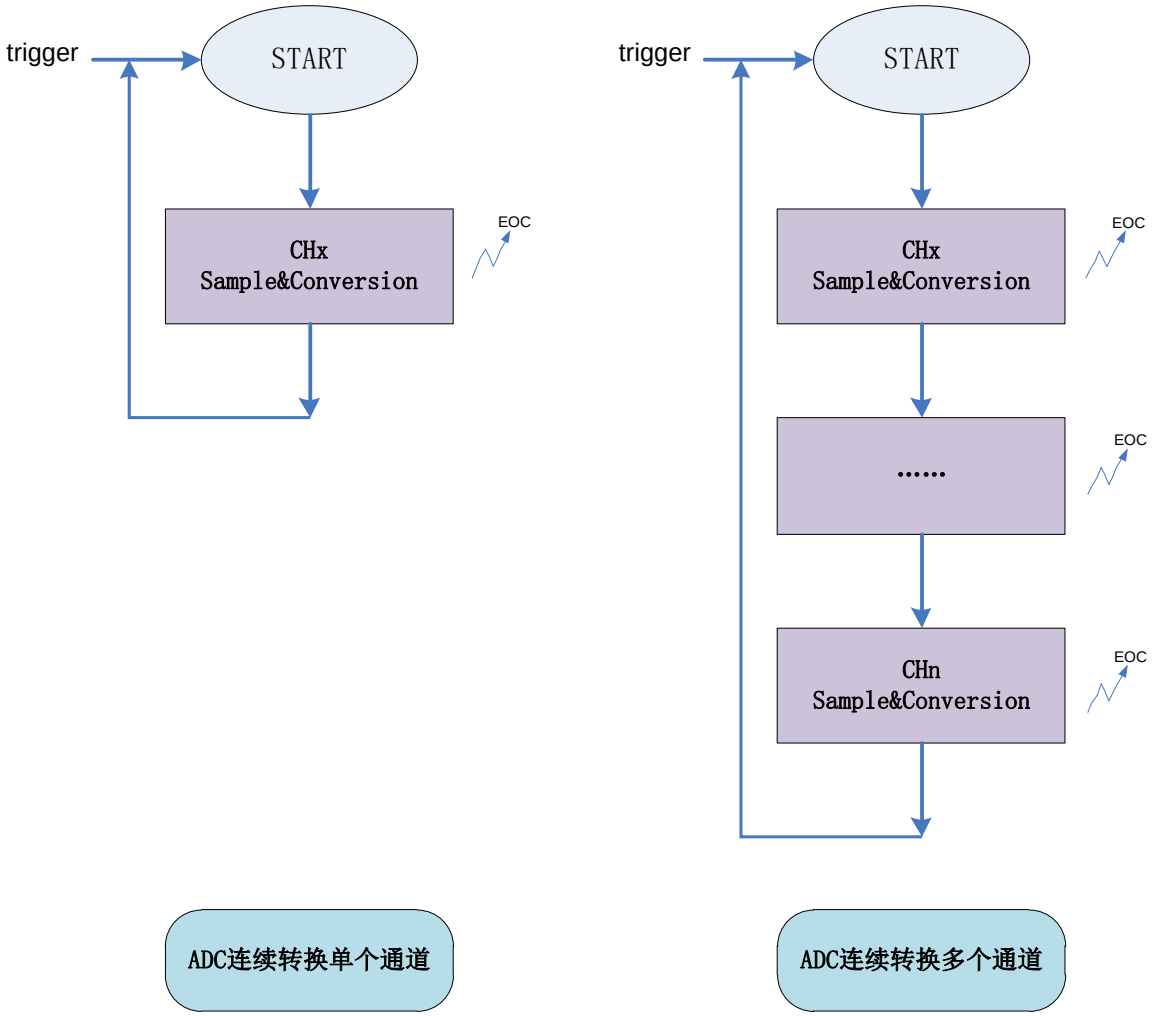


图 38-11 ADC 连续转换模式

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。

38.6.11 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

软件通过置位 SWTRIG 寄存器启动转换。

硬件触发

ADC 共有如下硬件触发源：LUTx_TRGO、RTCA_TRGO、ATIM_TRGO、GPTIM0_TRGO、

GPTIM1_TRGO、GPTIM2_TRGO、BSTIM16_TRGO、LPTIM16_TRGO、比较器输出；如果 ADC 正处于转换过程中，此时到来的触发信号会被忽略。

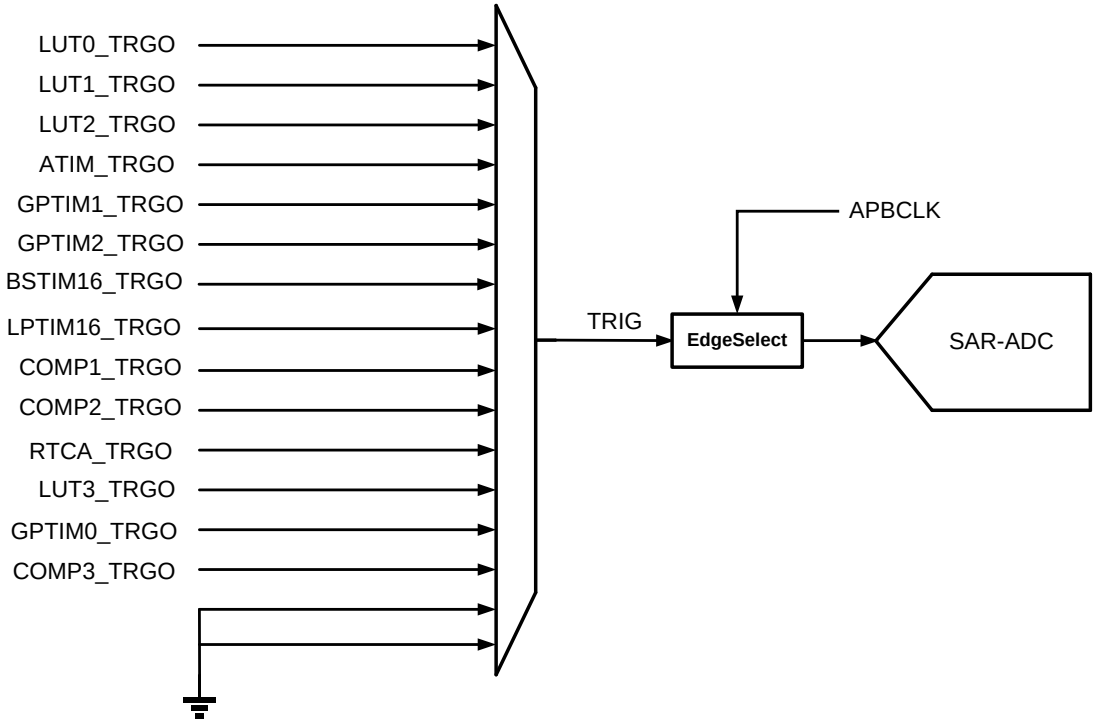


图 38-12 ADC 硬件触发源

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）。

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 (T_{APBCLK})	说明
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			
RTCA_TRGO			

38.6.12 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。

38.6.13 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；ADCCLK 为 ADC 工作时钟，可以是 APBCLK 或者其他独立时钟。

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。

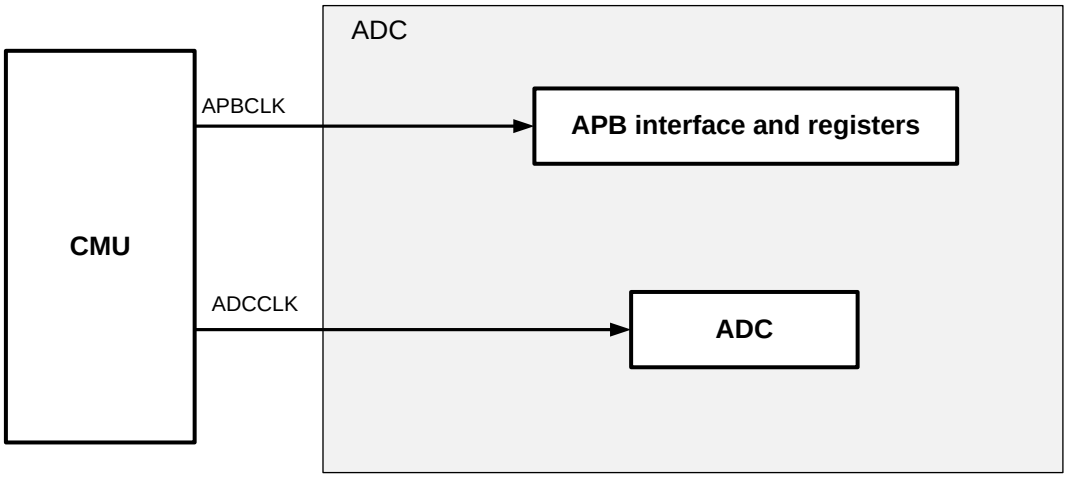


图 38-13 ADC 时钟示意图

38.6.14 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：

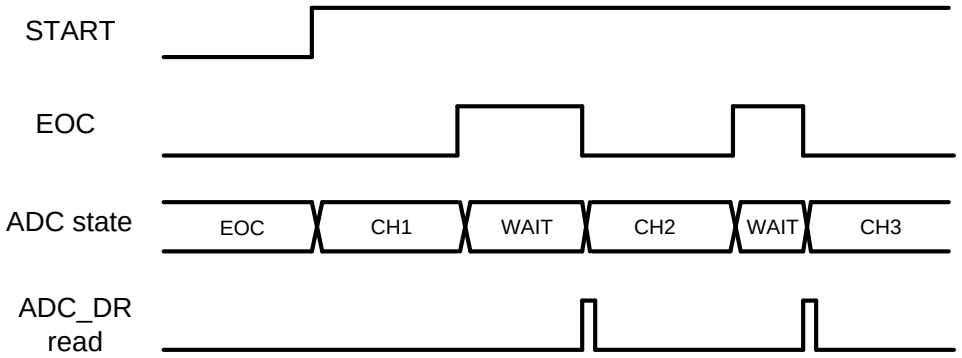


图 38-14 ADC 自动等待

38.6.15 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 控制器会产生一个 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 38-5 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。

38.6.15.1 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。

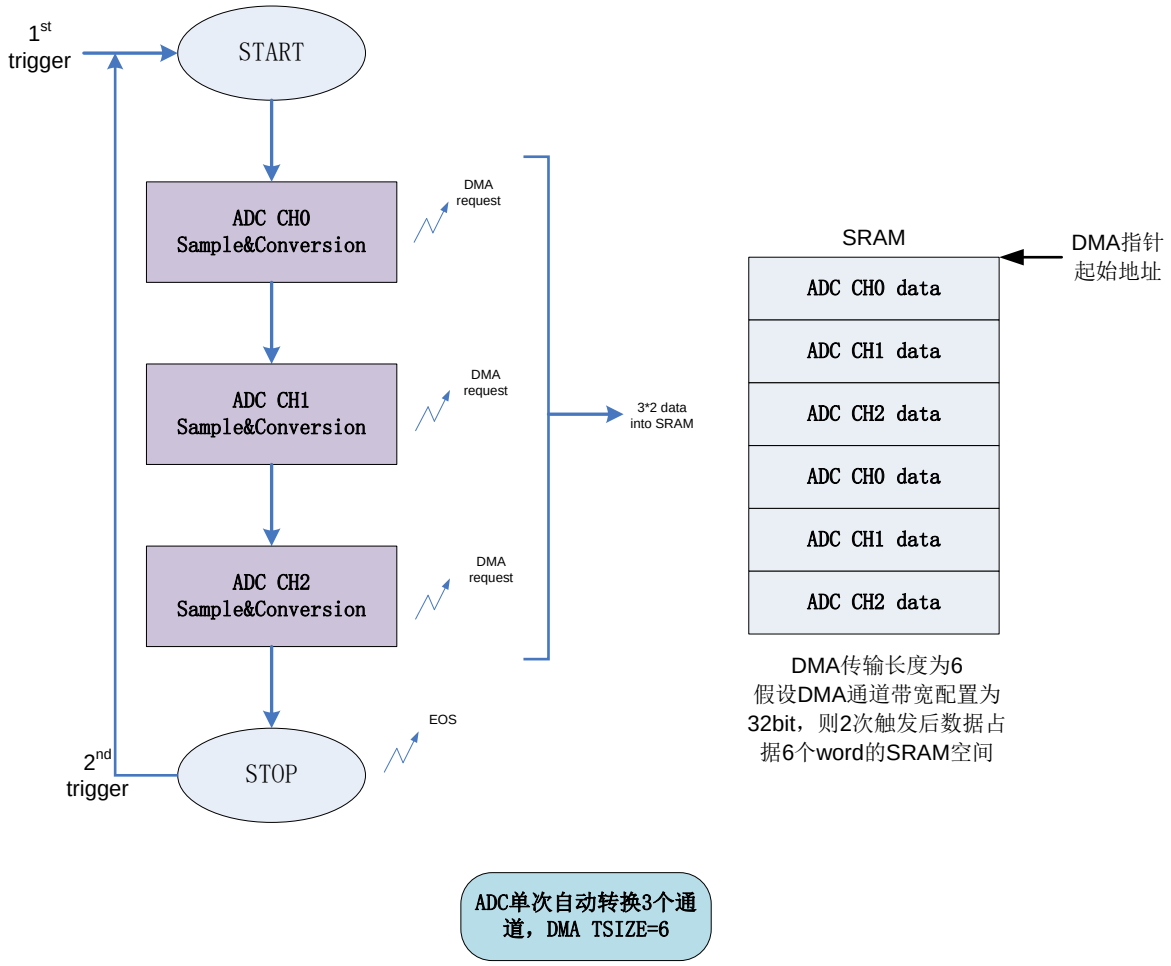


图 38-15 ADC 单次全自动触发+DMA 案例 1

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。

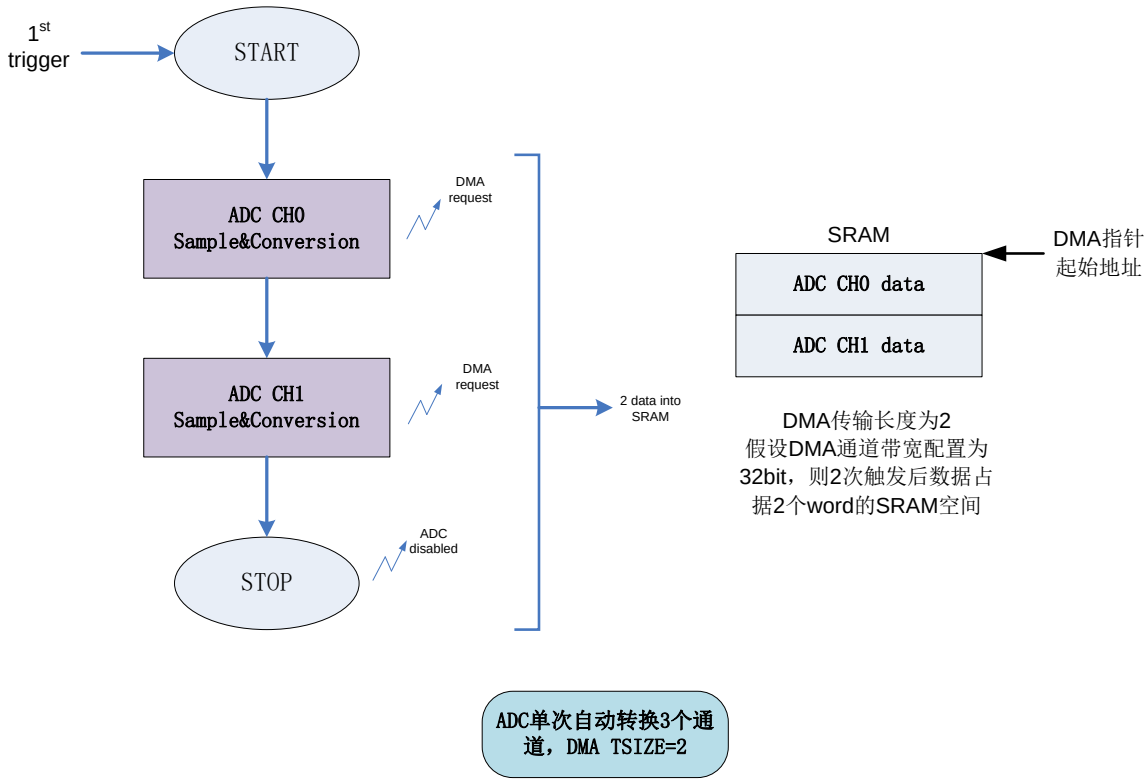


图 38-16 ADC 单次全自动触发+DMA 案例 2

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。

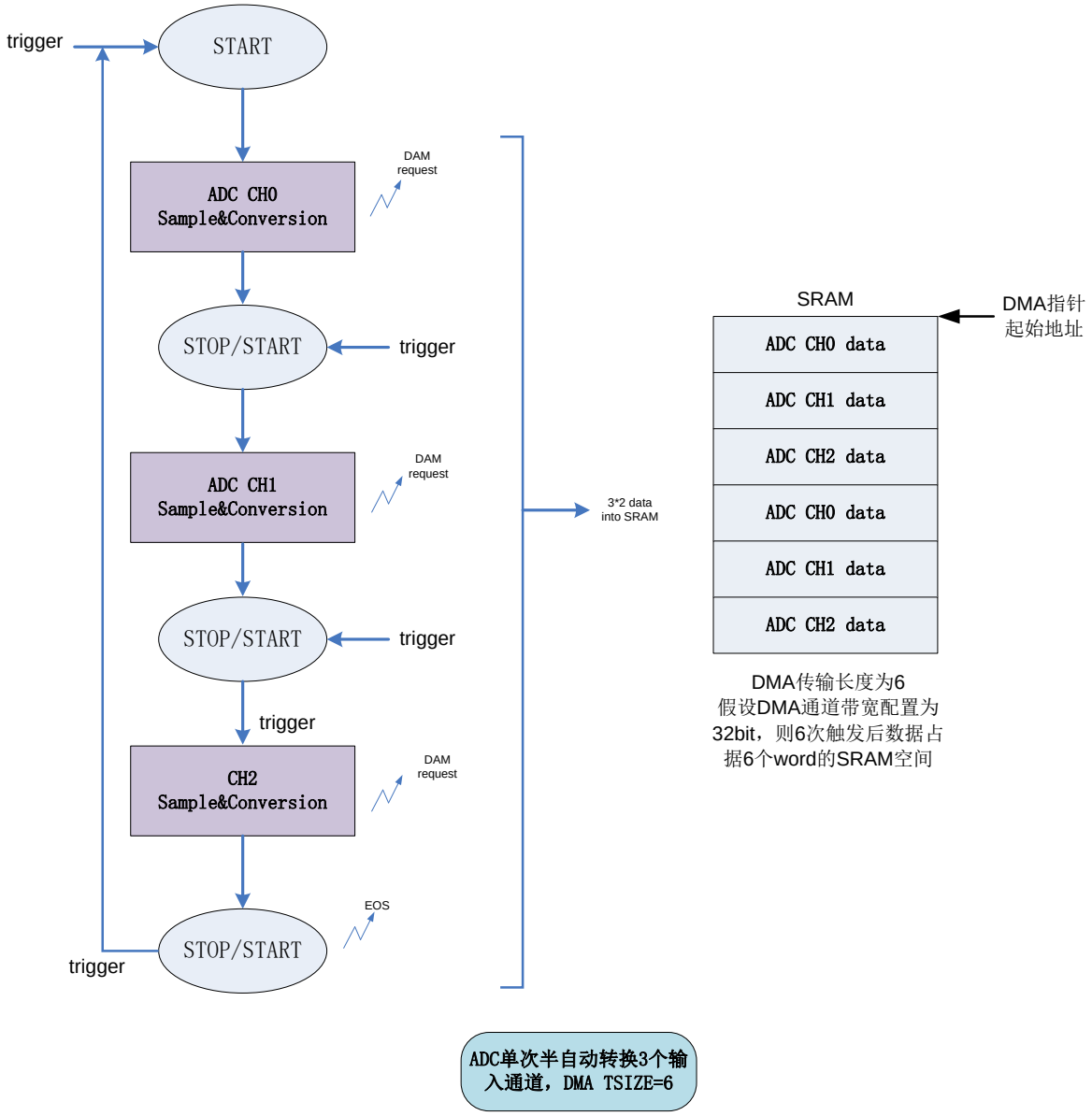


图 38-17 ADC 单次半自动触发+DMA

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。

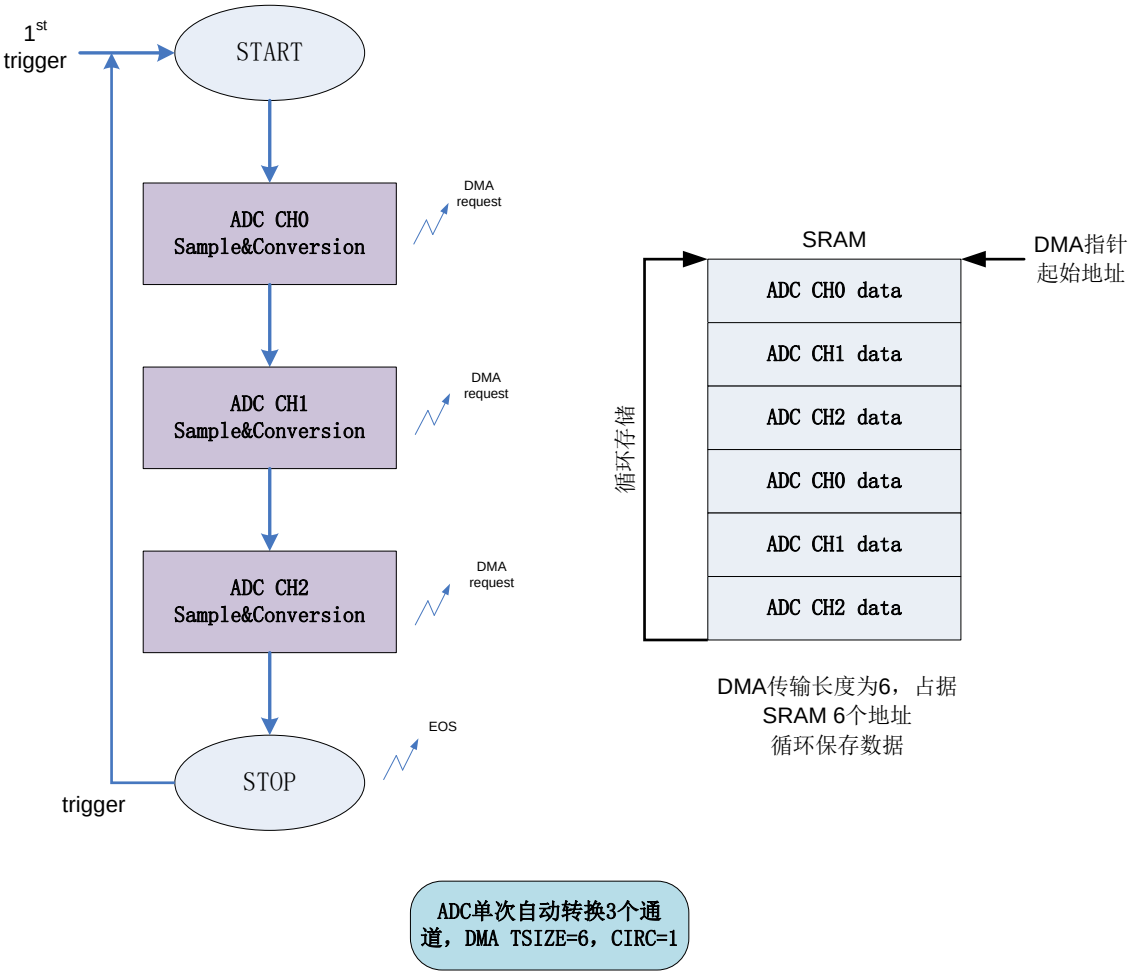


图 38-18 ADC 全自动触发+DMA 循环模式

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。

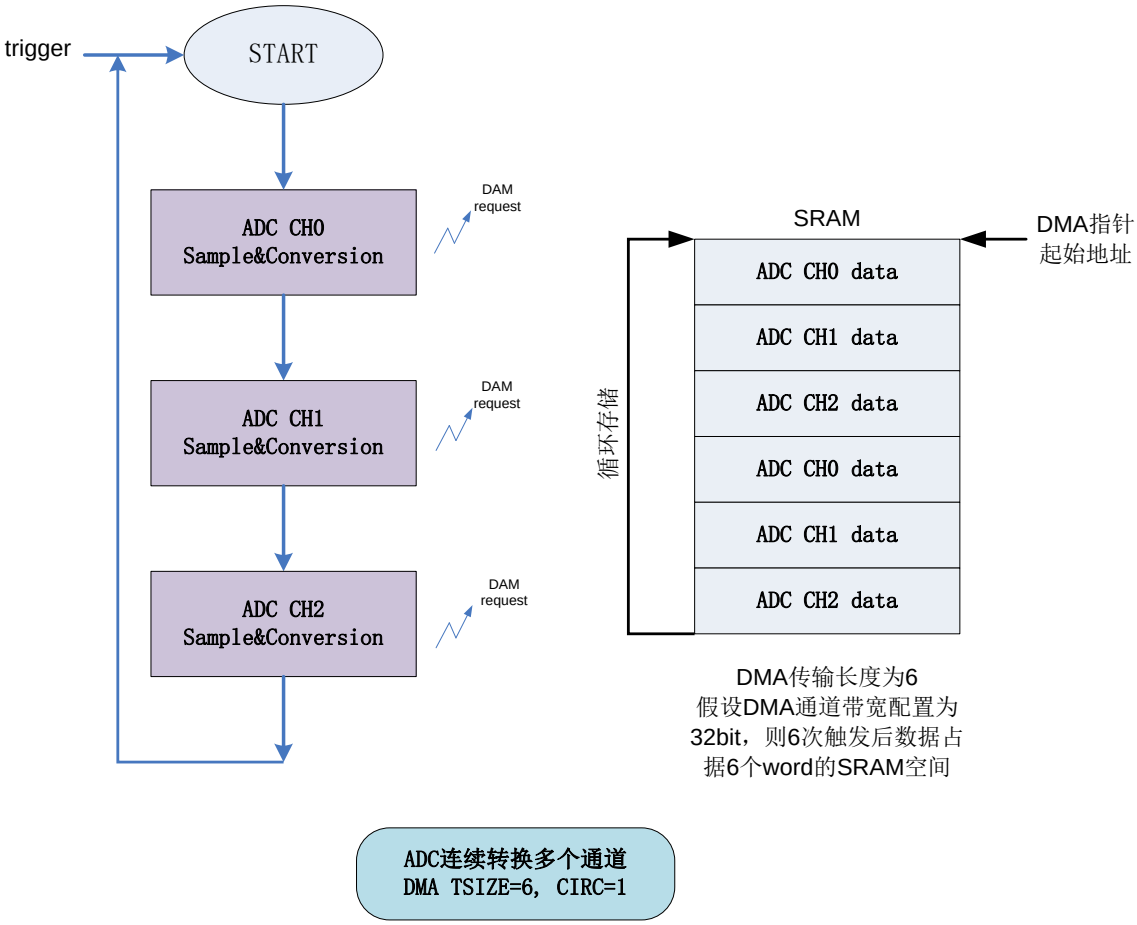


图 38-19 ADC 连续模式+DMA 循环模式

38.6.16 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。

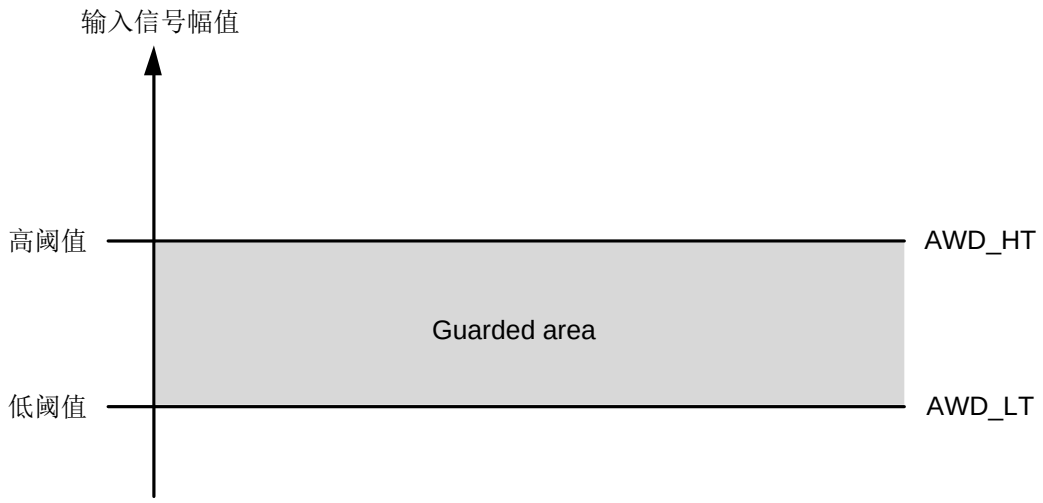


图 38-20 ADC 模拟看门狗阈值示意图

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

38.6.17 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。

38.7 低功耗模式

当芯片进入低功耗模式时，ADC 仍然允许工作。但是在低功耗模式下，芯片自动关闭了所有高速时钟源，所以 ADC 最高工作时钟仅为 RCLF，对应的最高采样率是 38.4Ksps。

模式	说明
LPRUN	可以工作，可以使用 DMA
Sleep/DeepSleep	可以工作，只能使用 RCLF 作为 ADC 时钟；并且可以唤醒 MCU
VBAT	不能工作

表 38-6 ADC 与低功耗模式

注意：由于外设触发需要 APBCLK，Sleep/DeepSleep 休眠模式下 APBCLK 关闭，因此无法在休眠模式下通过外设触发启动 ADC 转换。但是如果 ADC 使用 RCLF 时钟，并且软件在进入休眠前启动了 ADC 转换，则芯片休眠后 ADC 仍会工作。

38.8 寄存器

offset 地址	名称	符号
ADC(模块起始地址：0x40015C00)		
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 Register）	ADC_CR1
0x0C	ADC 控制寄存器 2 （ADC Control Register）	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 通道控制寄存器 （ADC Channel Enable Register）	ADC_CHER
0x24	ADC 差分通道控制寄存器 （ADC Differential Channel Control Register）	ADC_DCR
0x28	ADC 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

38.8.1 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-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:7	--	RFU：未实现，读为 0

位号	助记符	功能描述
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 比较器校准和通道校准结束后都会置位 EOCAL 标志。
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。

38.8.2 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

位号	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下限中断使能，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

位号	助记符	功能描述
		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

38.8.3 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	-						SWTRIG	ADEN
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	SWTRIG	ADC 启动转换寄存器（软件触发），软件写 1 启动，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

38.8.4 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	-						TRGCFG	
位权限	U-0						R/W-00	

位号	助记符	功能描述
31:1	--	RFU：未实现，读为 0
1:0	TRGCFG	触发信号使能和极性选择。(Trigger Config) 当配置为非软件触发时，禁止软件写 START 寄存器。 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

38.8.5 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	-			OFFL_EN				
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		-			CALEN
位权限	R/W-0	R/W-0	R/W-00		U-0			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	OFFL_EN	离线校准使能 (offline calibration enable) 0：自动校准 1：离线校准
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
3:1	-	RFU：未实现，读为 0

位号	助记符	功能描述
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：启动校准时必须保证 OVSEN=0（不能使能过采样）*

38.8.6 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	-				BUFCHP_EN	BUFLPF	BUFMOD	BUFEN
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	APBCLK_PSC		EXSOC	BITSEL		-	REFSEL	
位权限	R/W-00		R/W-0	R/W-00		U-0	R/W-00	

位号	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11	BUFCHP_EN	缓冲器 chopper 使能 (Buffer chopper enable) 0：关闭 chopper 功能 1：使能 chopper 功能 *注：使能 chopper 的时候必须使能硬件过采样。*
10	BUFLPF	慢速通道缓冲器低通滤波模式 (Buffer low-pass-filter mode) 0：正常模式 1：低通滤波模式
9	BUFMOD	慢速通道缓冲器功耗模式 (Buffer Mode) 0：正常模式 1：低功耗模式
8	BUFEN	慢速通道缓冲器使能 (Buffer Enable) 0：关闭并 bypass buffer 1：enable buffer
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	EXSOC	外部输入 SOC 使能 (External Start-of-conversion) 0：禁止外部输入 SOC 1：使能外部输入 SOC

位号	助记符	功能描述
4:3	BITSEL	ADC 输出位宽选择 (Bit-width) 00：12 位 01：10 位 10：8 位 11：6 位
2	CLKSEL	ADC 工作时钟选择 (Clock Select) 0：使用 ADCCLK 工作 1：使用 APBCLK 工作
1:0	REFSEL	ADC 基准源选择 (Reference Select) 00,11：使用 VDDA 作为基准 01：使用 VREFP 作为基准 10：使用 VDD15 作为基准

38.8.7 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

位号	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 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 00111：AWD 监视 ADC_IN6 01000：AWD 监视 ADC_IN7 01001：AWD 监视 ADC_IN8 01010：AWD 监视 ADC_IN9 01011：AWD 监视 ADC_IN10 01100：AWD 监视 ADC_IN11 01101：AWD 监视 ADC_IN12

位号	助记符	功能描述
		01110：AWD 监视 ADC_IN13 01111：AWD 监视 ADC_IN14 10000：AWD 监视 ADC_IN15 10001：AWD 监视 ADC_IN16 10010：AWD 监视 ADC_IN17 10011：AWD 监视 ADC_IN18 10100：AWD 监视 ADC_IN19 10101：AWD 监视 ADC_IN20 其他：保留
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 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

位号	助记符	功能描述
		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：LUT0_TRGO 0001：LUT1_TRGO 0010：LUT2_TRGO 0011：ATIM_TRGO 0100：GPTIM1_TRGO 0101：GPTIM2_TRGO 0110：BSTIM16_TRGO 0111：LPTIM16_TRGO 1000：COMP1_TRGO 1001：COMP2_TRGO 1010：RTCA_TRGO 1011：LUT3_TRGO 1100：GPTIM0_TRGO 1101：COMP3_TRGO 1110：RFU 1111：RFU *注意：必须在 TRGCFG=00 的情况下修改 EXTS 寄存器*
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_IN19->内部通道 1：反向扫描，内部通道->ADC_IN19->ADC_IN0
1	--	RFU：未实现，读为 0
0	DMAEN	DMA 使能 0：禁止 DMA 1：使能 DMA

38.8.8 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			

位号	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 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

38.8.9 ADC 通道控制寄存器（ADC_CHER）

名称	ADC_CHER							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	OPA	DAC	VDD/3	VBAT/3	AVREF	TS	REFINT
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	-				ECH19	ECH18	ECH17	ECH16
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	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

位号	助记符	功能描述
31	--	RFU：未实现，读为 0
30	OPA	OPA 输出测量，写 1 使能
29	DAC	DAC 输出测量，写 1 使能
28	VDD/3	VDD 分压测量，写 1 使能
27	VBAT/3	VBAT 分压测量，写 1 使能 *注意：VBAT 测量通道使能时，VBAT 分压电阻串自动使能，VBAT 电源下会增加一路功耗*
26	AVREF	快速基准源输出测量，写 1 使能
25	TS	温度传感器测量通道，写 1 使能
24	VREF1p2	内部基准电压测量通道，写 1 使能
23:20	--	RFU：未实现，读为 0
19	ECH19	ADC_IN0~19 测量通道，写 1 使能 其中，ADC_IN0~13 可以组成 7 对差分输入对，也可以作为 14 个单端输入使用 ADC_IN14~19 只能用作单端输入
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	

位号	助记符	功能描述
1	ECH1	
0	ECH0	

38.8.10 ADC 通道差分控制寄存器（ADC_DCR）

名称	ADC_DCR							
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	-	AINS						
位权限	U-0	R/W-000 0000						

位号	助记符	功能描述
31:7	--	RFU：未实现，读为 0
6:0	AINS	差分输入对使能寄存器，用于配置 ADC_IN0~ADC_IN13 是否组成 7 组差分对。0 表示单端输入，1 表示差分输入。(Analog Input Select) AINS[0]：1 表示配置 ADC_IN0 和 ADC_IN7 为差分输入 AINS[1]：1 表示配置 ADC_IN1 和 ADC_IN8 为差分输入 AINS[2]：1 表示配置 ADC_IN2 和 ADC_IN9 为差分输入 AINS[3]：1 表示配置 ADC_IN3 和 ADC_IN10 为差分输入 AINS[4]：1 表示配置 ADC_IN4 和 ADC_IN11 为差分输入 AINS[5]：1 表示配置 ADC_IN5 和 ADC_IN12 为差分输入 AINS[6]：1 表示配置 ADC_IN6 和 ADC_IN13 为差分输入

38.8.11 ADC 数据寄存器（ADC_DR）

名称	ADC_DR							
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	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							

位号	助记符	功能描述
31:16	--	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

38.8.12 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							

位号	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

39 数模转换器（DAC）

39.1 概述

FM33LG0集成一个12位电压输出型数模转换器，最高输出速率为1Msps，支持DMA传输。DAC使用VREF+引脚电压作为参考基准，引脚电压可以来自于内部基准源或者片外输入。

DAC带有一个输出驱动Buffer，使能Buffer的情况下可以提供更大的驱动能力；DAC输出支持采样保持模式，以降低平均工作电流。

DAC的主要特性如下：

- 工作电压 1.8～5.5V
- 输出信号幅度 0~VDDA
- 最高输出转换率 1Msps（F_{DAC}=16Mhz）
- 支持采样保持模式
- 支持 DMA
- DAC 输出可以连接到比较器输入

39.2 结构框图

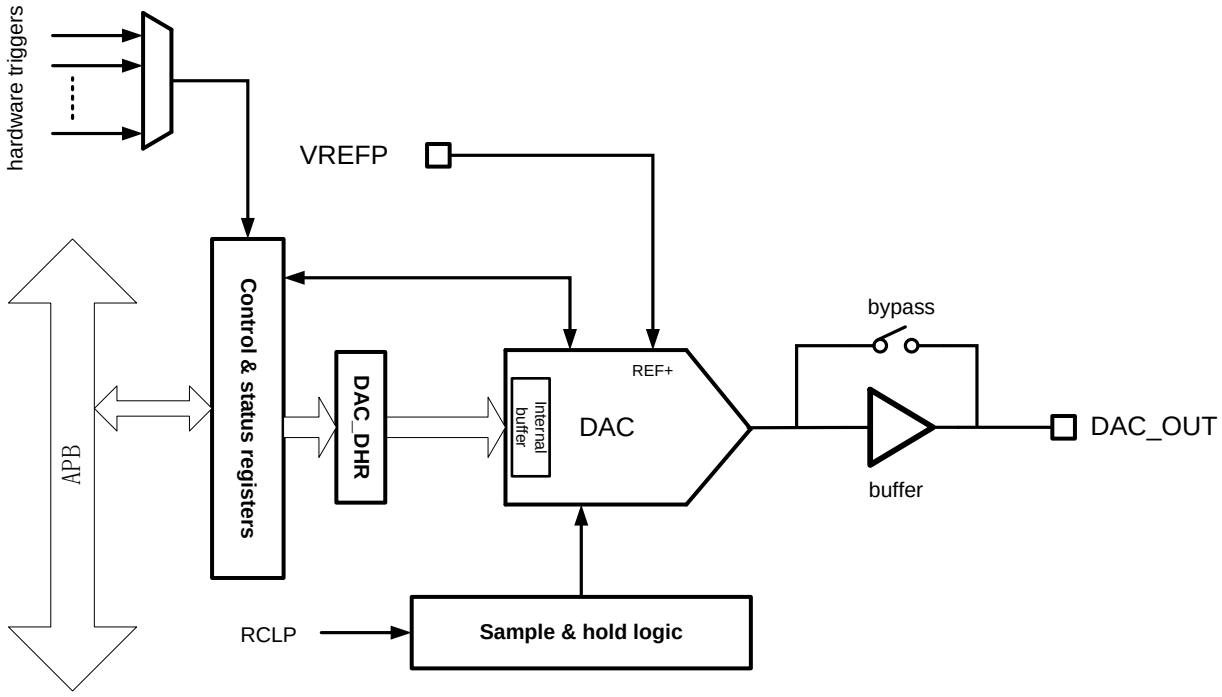


图 39-1 DAC 结构框图

39.3 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
VREF+	输入基准电压	DAC 工作基准，等于或低于 VDDA
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 39-1 DAC 相关引脚

39.4 DAC 输出与引脚和其他模块的互联

DAC输出既可以从PC5引脚直接输出，也可以输出到内部模块。为了避免输出给其他模块时影响PC5引脚功能，DAC buffer的反馈通路经过PAD的开关通道控制，如下图。

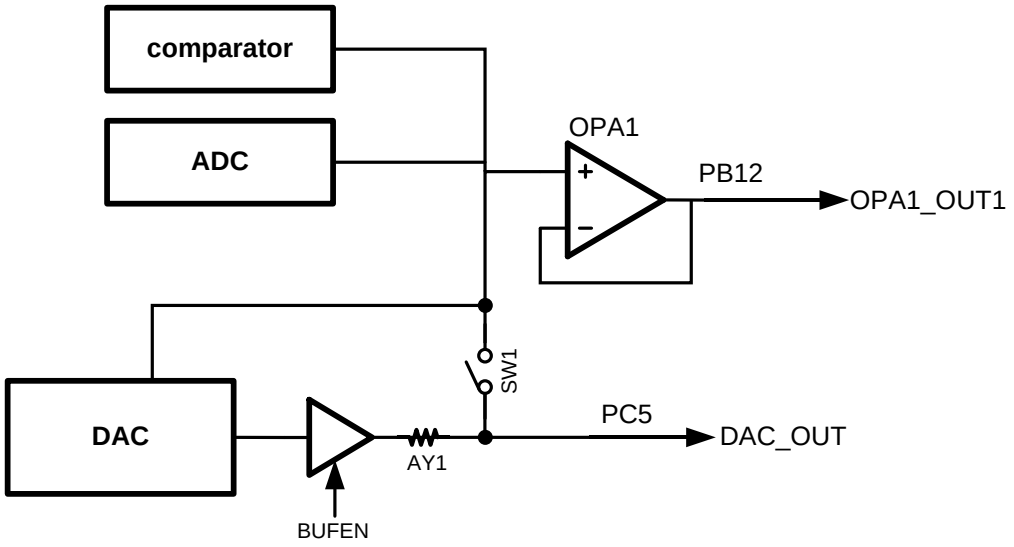


图 39-2 DAC 结构框图

当DAC_OUT输出连接片外负载时，BUFEN=1，SW1闭合（SWIEN=1）。

当DAC输出只连接到片内模块，不从PC5输出时，BUFEN=0，SW1断开（SWIEN=0）。

39.5 功能描述

39.5.1 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：

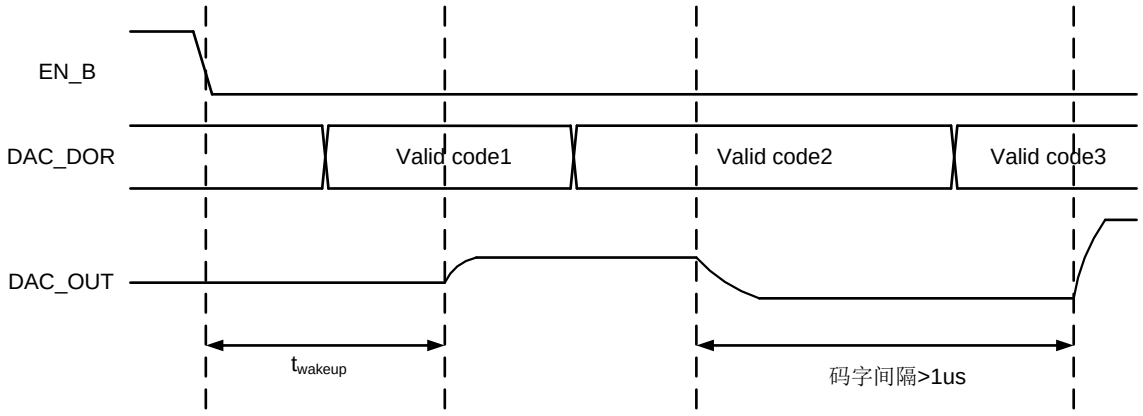


图 39-3 DAC 工作时序

39.5.2 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。

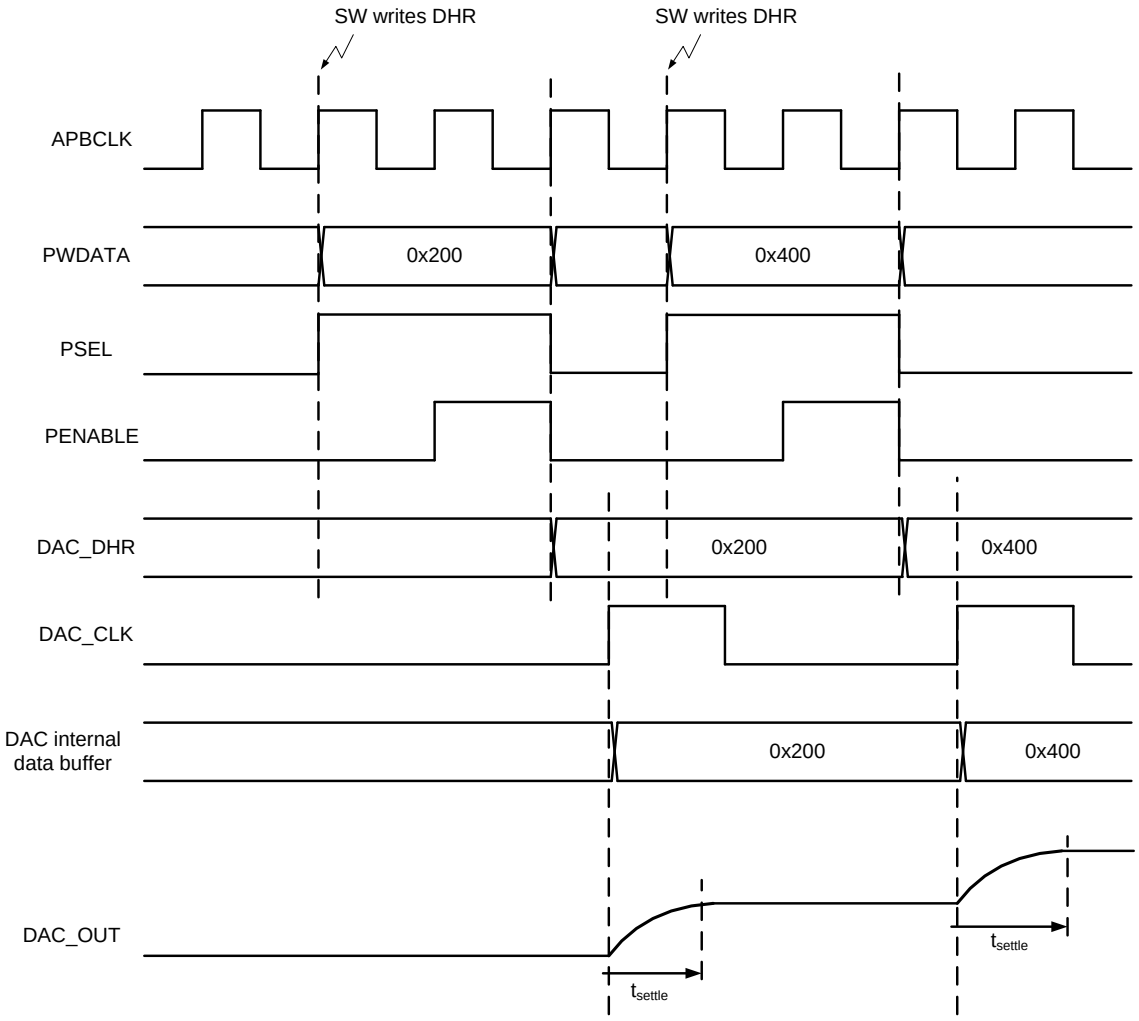


图 39-4 DAC 连续输出模式

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。

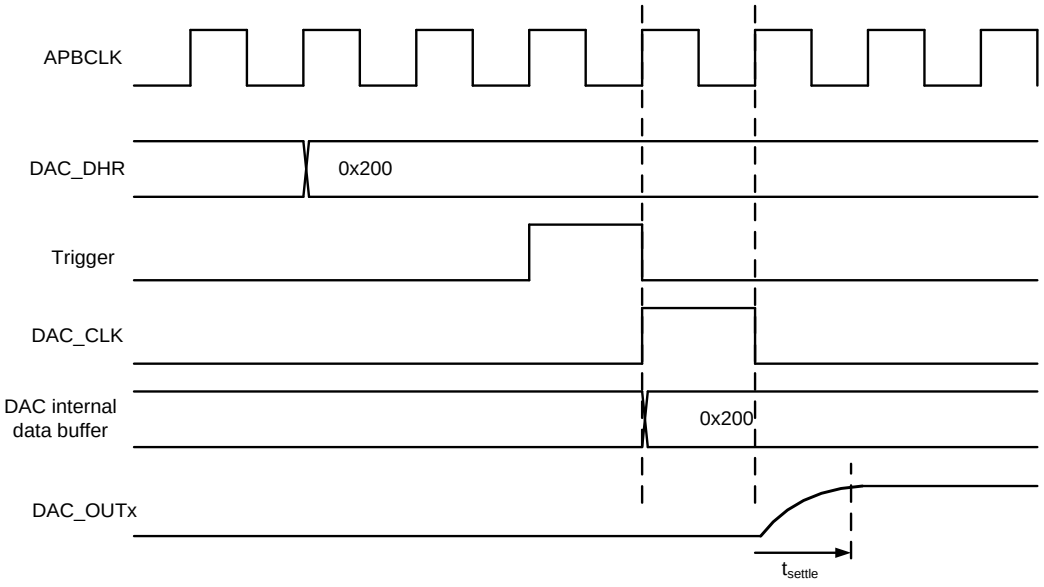


图 39-5 DAC 触发输出模式

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。

39.5.3 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
BSTIM16_TRGO		0100
LPTIM16_TRGO		0101
RFU	-	0110
		0111
		1000
		1001
		1010
		1011
EXTI[0]	引脚中断触发	1100
EXTI[4]		1101
EXTI[8]		1110
EXTI[12]		1111

表 39-2 DAC 触发源

39.5.4 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输出电压理论值由下式决定：

$$DAC_OUT = VREF \times \frac{DAC_DOR}{4096}$$

39.5.5 DMA

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。

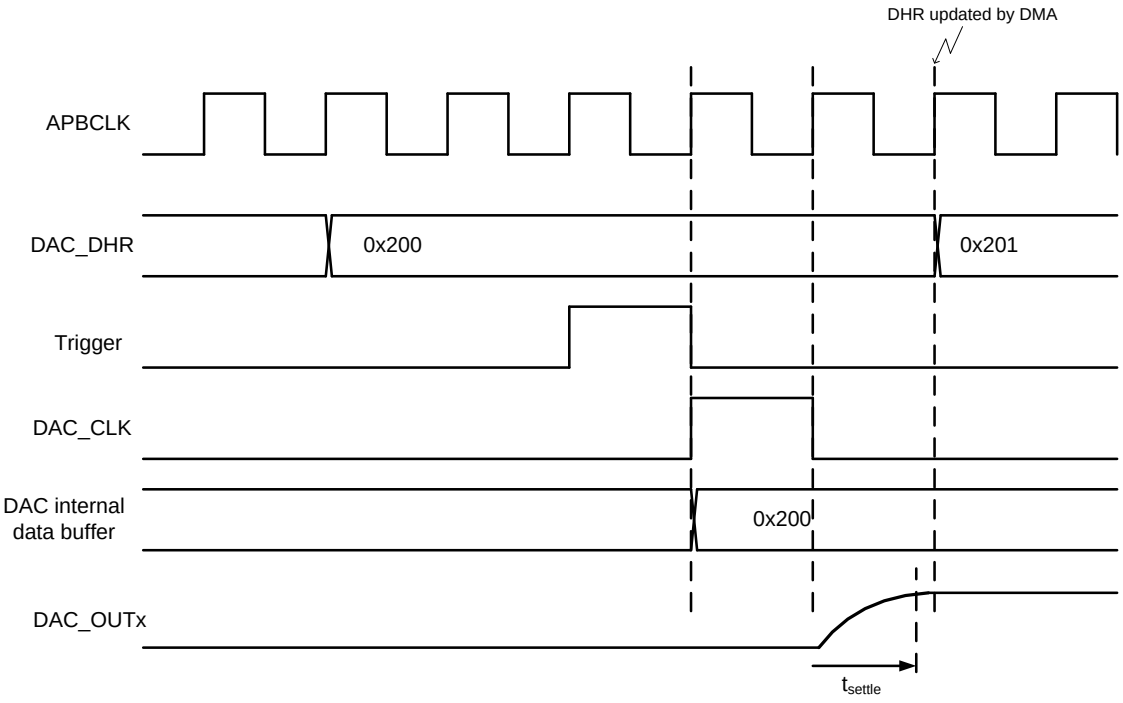


图 39-6 触发模式下通过 DMA 更新数据

39.5.6 采样保持

DAC输出支持正常模式和采样保持模式。通过使能DAC输出Buffer可以获得较强的输出驱动能力。

正常模式下，DAC保持持续工作，即DAC_OUTx信号会被DAC持续驱动；这是最常见的工作方式，在DAC工作期间会有稳定的电流消耗。

在采样保持模式下，DAC在接收到触发信号后，只会对DAC_OUTx输出驱动一段时间，然后自动关闭DAC，DAC_OUTx信号变成高阻态，需要片外电容来保持之前的电压状态；内部定时器会在间隔一段可编程的时间后，重新使能DAC并驱动DAC_OUTx；通过这种间歇驱动的方式，配合片外采样保持电容，可以大幅降低DAC工作的平均电流。

采样保持模式可以被划分为2个阶段：

- 采样阶段：DAC输出驱动使能，外部采样电容被充电至目标电平。充电时间主要取决于DAC驱动能力，以及采样电容的容值。通过TSMPL寄存器，可以配置采样阶段的时间长度（基于RCLP

时钟定时）。

- 保持阶段：这个阶段中DAC被关闭，输出高阻态，采样电容保持电压状态。保持阶段的时间长度由THLD寄存器配置（基于RCLP时钟定时）。

进入采样保持模式的步骤：

- 配置T_{SMPL}和T_{HLD}寄存器
- 使能DAC
- 置位SHEN，启动采样保持模式
- 内部计数器开始计数，当计数值等于T_{SMPL}，硬件自动关闭DAC使能
- 计数器继续计数，当计数值等于T_{SMPL} + T_{HLD}，硬件重新使能DAC，并清零计数器重新计数

退出采样保持模式：

- 清零SHEN寄存器（关闭采样保持，回到常使能模式）
- 清零EN寄存器（关闭DAC，可选）

SHEN控制计数器逻辑，当SHEN=0时，periodic_en=1，即DAC由DAC_CR.EN寄存器直接控制。
当SHEN=1时，periodic_en信号可以周期性关闭DAC，而DAC_CR.EN寄存器保持置位。
所以软件先置位DAC_CR.EN，在需要时置位SHEN进入采样保持。
此后如果软件清零SHEN，则直接回到常使能状态。
如果软件清零DAC_CR.EN，则彻底关闭DAC。

采样保持控制电路使用RCLP时钟工作，以降低运行功耗，使能采样保持模式前，软件必须确保RCLP时钟使能。

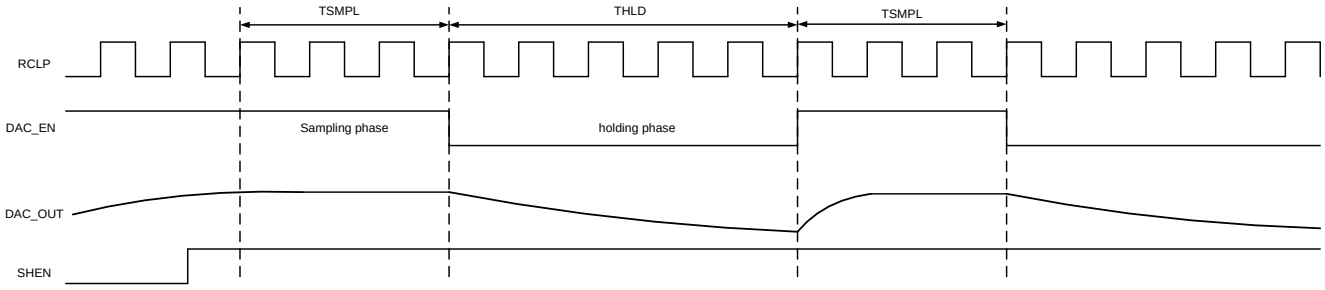


图 39-7 DAC 采样保持输出

采样保持模式的目的是使用很小的功耗维持外部采样电容上的一个基本恒定的电压。在连续输出模式下，当发生针对DAC_DHR寄存器的任何写操作时，DAC_DOR数据也会被实时更新，此时不论当前采样保持控制电路处于哪个阶段，都会立即触发一次新的采样。（即清零采样保持计数器，重新开始采样阶段）

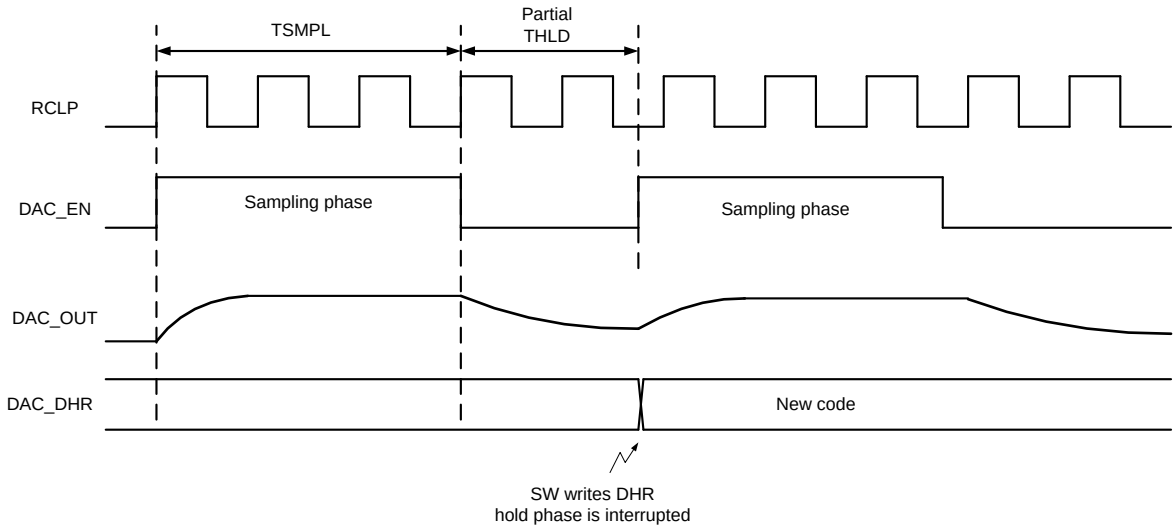


图 39-8 DAC 采样保持输出过程中更新 DHR（TRGEN=0）

在触发输出模式下，软件更新DHR寄存器并不立即更新DAC输出，而是要等待触发事件到来，才清零计数器并将DHR数据更新到DAC输出buffer中。如果应用希望即时更新输出电平，则应在更新DHR寄存器后写一次SWTRIG寄存器。

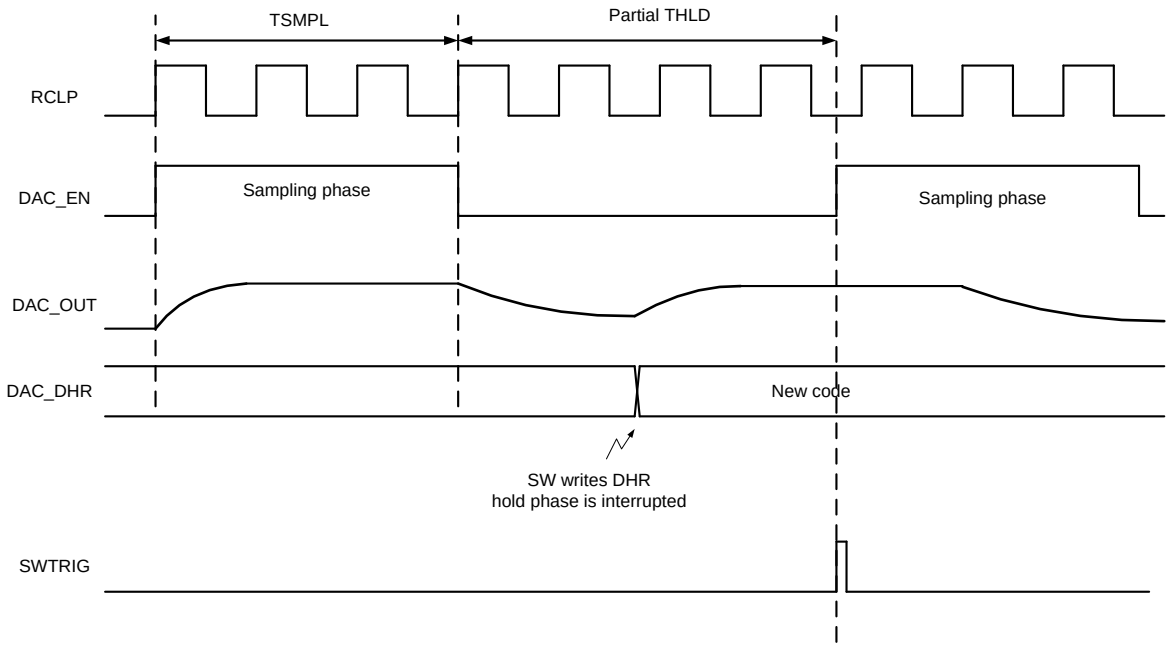


图 39-9 DAC 采样保持输出过程中更新 DHR（TRGEN=1）

39.5.7 DAC 输出 Buffer

DAC输出Buffer可以用来增强DAC输出驱动能力，具体驱动能力参数参见3电参数。

通过Buffer输出时，无法实现0~VREF+的轨到轨输出范围，最大输出范围仅能支持0.2V~VREF-0.2V。

39.5.8 低功耗模式下的 DAC

模式	说明
LPRUN	可以工作，可以使用 DMA
Sleep/DeepSleep	可以工作，但是只能维持输出固定电平
VBAT	不能工作

表 39-3 DAC 与低功耗模式

39.6 寄存器

offset 地址	名称	符号
DAC(模块起始地址：0x40019800)		
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 Config Register）	DAC_CFGR
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
0x1C	DAC 采样保持时间寄存器 （DAC Sample Hold Time Register）	DAC_SHTR

39.6.1 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

位号	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

39.6.2 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

位号	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	DMAEN	DMA 使能 (DMA 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

39.6.1 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	-							SHEN
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BUFEN	-	TRGSEL				-	SWIEN
位权限	R/W-0	U-0	R/W-0000				U-0	R/W-0

位号	助记符	功能描述
31:9	--	RFU：未实现，读为 0
8	SHEN	采样保持模式使能 (Sample Hold Enable) 1：采用采样保持输出模式 0：正常连续输出模式
7	BUFEN	DAC 输出驱动 Buffer 使能 (Buffer enable) 0：关闭输出 buffer 1：使能输出 buffer

位号	助记符	功能描述
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 39.5.3DAC 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

39.6.2 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

位号	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。

39.6.3 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-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DHR							

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	DHR	DAC 数据保持寄存器 (Data Holding Register)

39.6.4 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	EOH	EOS	DOU
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	DMAERR	DMA 错误标志 (DMA Error) 仅 DMAEN=1 时有效，软件写 1 清零
2	EOH	保持阶段结束标志 (End of Holding phase) 仅采样保持模式下（SHEN=1）有效，当保持阶段结束后置位，软件写 1 清零。
1	EOS	采样阶段结束标志 (End of Sampling phase) 仅采样保持模式下（SHEN=1）有效，当采样阶段结束后置位，软件写 1 清零。
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。

39.6.5 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	EOH_IE	EOS_IE	DOU_IE
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	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	EOH_IE	保持阶段结束中断使能 (End of Holding phase Interrupt Enable) 1：允许 EOH 中断 0：禁止 EOH 中断
1	EOS_IE	采样阶段结束标志 (End of Sampling phase Interrupt Enable) 1：允许 EOS 中断 0：禁止 EOS 中断
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

39.6.1 DAC 采样保持时间寄存器（DAC_SHTR）

名称	DAC_SHTR							
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	THLD[15:8]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	THLD[7:0]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TSMPL							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:24	--	RFU：未实现，读为 0
23:8	THLD	采样保持模式下的保持时间 (Time holding)，单位是 RCLP 时钟周期 0x0：1T

位号	助记符	功能描述
		0x1：2T …… 0xFFFF：65536T
7:0	TSMPL	采样保持模式下的采样时间 (Time Sampling)，单位是 RCLP 时钟周期 0x0：1T 0x1：2T …… 0xFF：256T

注：禁止在SHEN=1的情况下修改TSMPL和THLD寄存器。

40 可编程胶合逻辑（PGL）

40.1 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

40.2 结构框图

LUT的结构如下图所示。

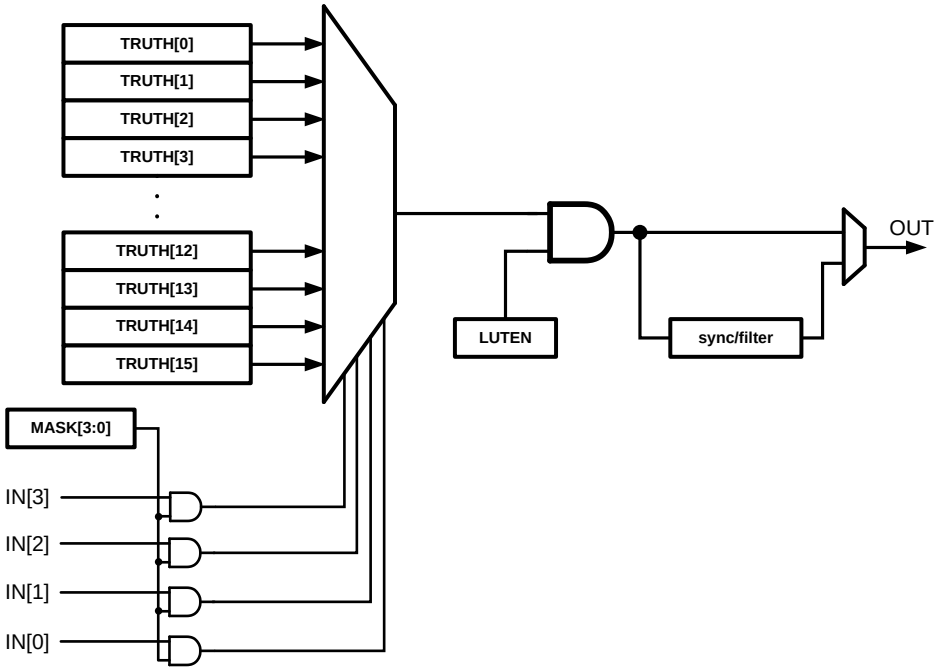


图 40-1LUT 结构框图

PGL模块共包含了4个LUT。

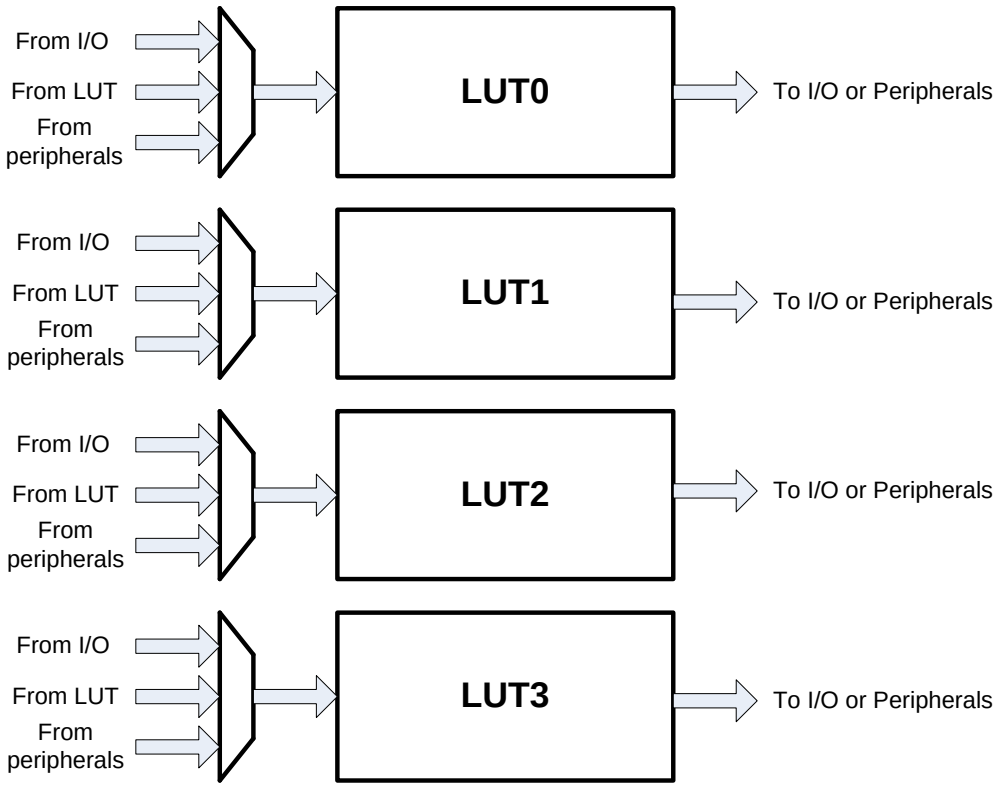


图 40-2 PGL 结构框图

40.3 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
PC2	GPIO 输入	LUT0 输入 0
PC3		LUT0 输入 1
PC4		LUT0 输入 2
PC5		LUT0 输入 3
PC6		LUT1 输入 0
PC7		LUT1 输入 1
PC8		LUT1 输入 2
PC9		LUT1 输入 3
PB7		LUT2 输入 0
PB8		LUT2 输入 1
PB9		LUT2 输入 2
PB10		LUT2 输入 3
PB11		LUT3 输入 0
PB12		LUT3 输入 1
PB13		LUT3 输入 2
PB14		LUT3 输入 3
PC0	额外数字功能	LUT0 输出
PC1		LUT1 输出
PA1		LUT2 输出
PA0		LUT3 输出

表 40-1 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。

40.4 功能描述

40.4.1 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。

每个LUT支持4输入，输入作为地址可以查询到16个输出，LUT本身就是一个4bit地址寻址的存储空间。寄存器TRUTH[15:0]保存了16个输出数据，地址对应关系如下表所示。

输入 3~0	查询地址	输出数据
0000	0	TRUTH[0]
0001	1	TRUTH[1]
0010	2	TRUTH[2]
0011	3	TRUTH[3]
0100	4	TRUTH[4]
0101	5	TRUTH[5]
0110	6	TRUTH[6]
0111	7	TRUTH[7]
1000	8	TRUTH[8]
1001	9	TRUTH[9]
1010	10	TRUTH[10]
1011	11	TRUTH[11]
1100	12	TRUTH[12]
1101	13	TRUTH[13]
1110	14	TRUTH[14]
1111	15	TRUTH[15]

表 40-2LUT 查找表

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

输入 1~0	查询地址	输出数据
00	0	TRUTH[0]=1
01	1	TRUTH[1]=1
10	2	TRUTH[2]=1
11	3	TRUTH[3]=0

可以看到以上真值表对应一个2输入的NAND。

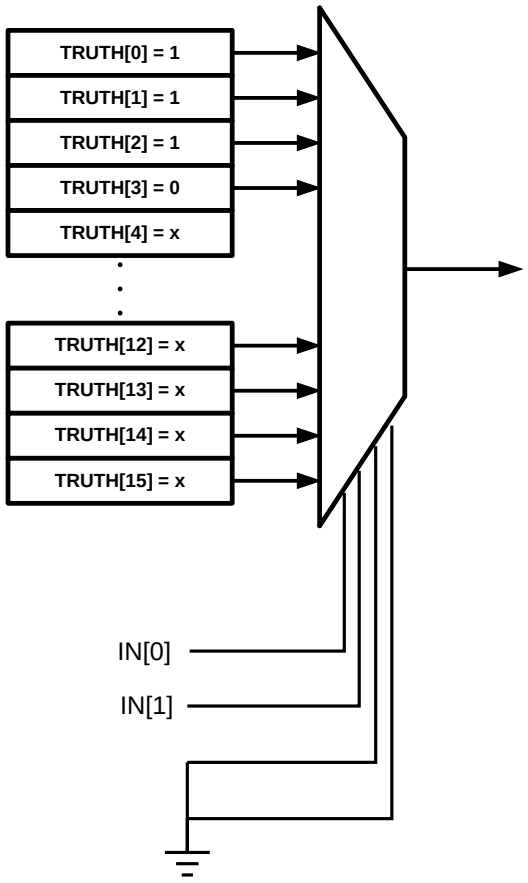


图 40-3 LUT 实现 2 输入 NAND 示意图

40.4.2 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。

40.4.3 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	LUT1,LUT2,LUT3
LUT1 输出	PC1	LUT2,LUT3
LUT2 输出	PA1	LUT3
LUT3 输出	PA0	-

表 40-3 LUT 输出连接

40.4.4 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。

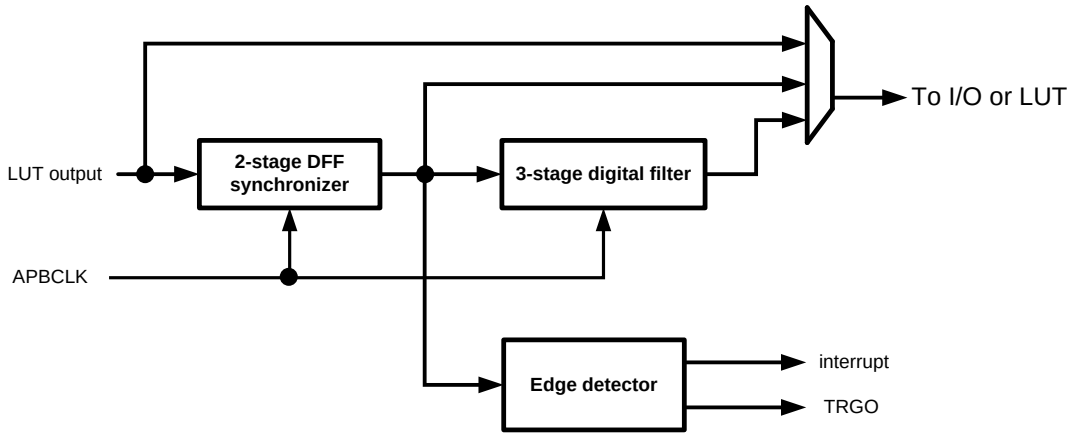


图 40-4 输出滤波和采样

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。

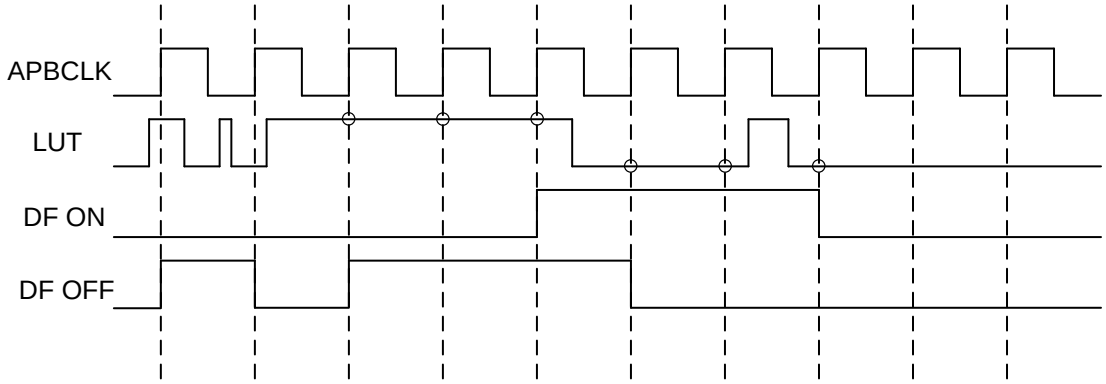


图 40-5 数字滤波

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。

40.4.5 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。

PGL的4个LUT可以产生4个触发信号输出，其扇出关系如下：

Trigger 输出	接收模块
LUT0_TRGO	ADC, GPTIM2
LUT1_TRGO	ADC, GPTIM1
LUT2_TRGO	ADC, GPTIM2
LUT3_TRGO	ADC, GPTIM1

表 40-4 LUT 输出连接

40.4.6 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。

40.5 寄存器

offset 地址	名称	符号
PGL(模块起始地址：0x40016C00)		
0x00	PGL 控制寄存器 （PGL Control Register）	PGL_CR
0x04	PGL 配置寄存器 0 （PGL Config Register0）	PGL_CFGR0
0x08	PGL 配置寄存器 1 （PGL Config Register1）	PGL_CFGR1
0x0C	PGL 配置寄存器 2 （PGL Config Register2）	PGL_CFGR2
0x10	PGL 配置寄存器 3 （PGL Config Register3）	PGL_CFGR3
0x14	PGL 中断使能寄存器 （PGL Interrupt Enable Register）	PGL_IER
0x18	PGL 中断标志寄存器 （PGL Interrupt Status Register）	PGL_ISR
0x1C	LUT0 真值表寄存器 （Look Up Table0）	PGL_LUT0
0x20	LUT1 真值表寄存器 （Look Up Table1）	PGL_LUT1
0x24	LUT2 真值表寄存器 （Look Up Table2）	PGL_LUT2
0x28	LUT3 真值表寄存器 （Look Up Table3）	PGL_LUT3

40.5.1 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			

位号	助记符	功能描述
31:4	--	RFU：未实现，读为 0

位号	助记符	功能描述
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]

40.5.2 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			

位号	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：LPTIM16_CH1 10：LPTIM32_CH1 11：ATIM_CH1
13:12	IN2SEL	LUT0 输入 2 选择寄存器 00：GPIO 01：RFU 10：COMP3_OUT 11：GPTIM2_CH1
11:10	IN1SEL	LUT0 输入 1 选择寄存器 00：GPIO 01：RFU

位号	助记符	功能描述
		10：COMP2_OUT 11：GPTIM1_CH1
9:8	IN0SEL	LUT0 输入 0 选择寄存器 00：GPIO 01：RFU 10：COMP1_OUT 11：GPTIM0_CH1
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

40.5.3 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			

位号	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 01：LPTIM16_CH1 10：LPTIM32_CH1 11：ATIM_OC1REF
13:12	IN2SEL	LUT1 输入 2 选择寄存器

位号	助记符	功能描述
		00：GPIO 01：RFU 10：COMP3_OUT 11：GPTIM2_CH1
11:10	IN1SEL	LUT1 输入 1 选择寄存器 00：GPIO 01：RFU 10：COMP2_OUT 11：GPTIM1_CH1
9:8	IN0SEL	LUT1 输入 0 选择寄存器 00：GPIO 01：LUT0 output 10：COMP1_OUT 11：GPTIM0_CH1
7:4	--	RFU：未实现，读为 0
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

40.5.4 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			

位号	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

位号	助记符	功能描述
		11：RFU
15:14	IN3SEL	LUT2 输入 3 选择寄存器 00：GPIO 01：LPTIM16_CH1 10：LPTIM32_CH1 11：ATIM_CH1
13:12	IN2SEL	LUT2 输入 2 选择寄存器 00：GPIO 01：RFU 10：COMP3_OUT 11：GPTIM2_CH1
11:10	IN1SEL	LUT2 输入 1 选择寄存器 00：GPIO 01：LUT1 output 10：COMP2_OUT 11：GPTIM1.OC1REF
9:8	IN0SEL	LUT2 输入 0 选择寄存器 00：GPIO 01：LUT0 output 10：COMP1_OUT 11：GPTIM0_CH1
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

40.5.5 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			

位号	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

位号	助记符	功能描述
		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 01：LPTIM16_CH1 10：LPTIM32_CH1 11：ATIM_CH1
13:12	IN2SEL	LUT3 输入 2 选择寄存器 00：GPIO 01：LUT2 output 10：COMP3_OUT 11：GPTIM2_CH1
11:10	IN1SEL	LUT3 输入 1 选择寄存器 00：GPIO 01：LUT1 output 10：COMP2_OUT 11：GPTIM1_CH1
9:8	IN0SEL	LUT3 输入 0 选择寄存器 00：GPIO 01：LUT0 output 10：COMP1_OUT 11：GPTIM0_CH1
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

40.5.6 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

位号	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

40.5.7 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			

位号	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

40.5.8 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							

位号	助记符	功能描述

位号	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

40.5.9 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							

位号	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

40.5.10 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							

位号	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

40.5.11 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							

位号	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

41 I/O 端口（GPIO）

41.1 概述

I/O 端口的主要功能特性：

- GPIO 引脚输入电压不能高于电源电压
- 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、开漏输出
- Sleep/DeepSleep 模式下保持状态
- 3 个 IO 由切换后的 VAO 电源供电

41.2 引脚类型

FM33LG0 的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；

41.2.1 GPIO，输入输出使能，可控上拉电阻，可控开漏输出

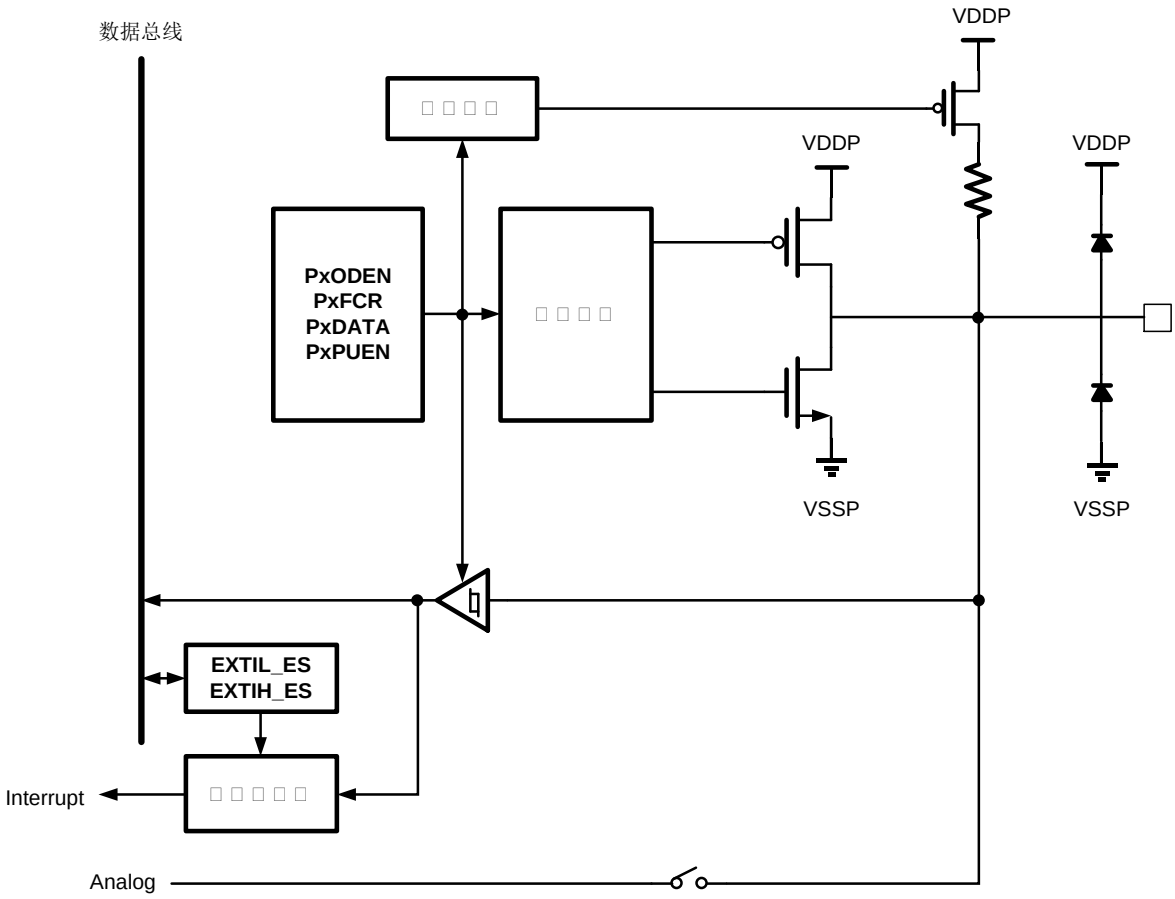


图 41-1 普通 GPIO 结构框图

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
	0	0		外设推挽输出功能	0	1	
	1				1	1	
	0	1		外设开漏输出 0	0	1	
	1				1	1	
	0			外设开漏输出 1	0	0	
	1				1	0	
11	x	x	x	x	0	0	0

表41-1GPIO功能逻辑定义表

41.2.2 GPIO，输入输出使能，2 个可控上拉电阻，可控开漏输出（仅 PC12-7816 数据口）

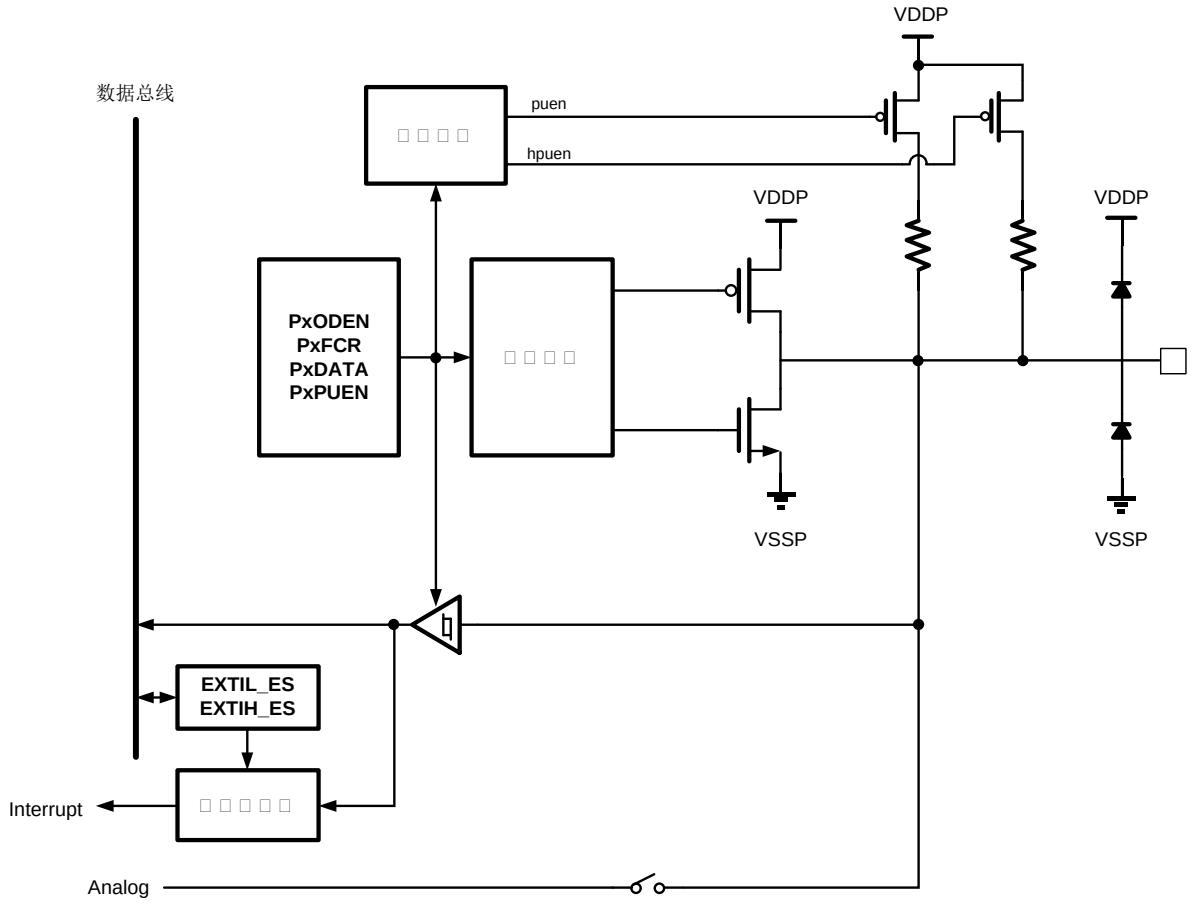


图 41-2 普通 GPIO（两路上拉）结构框图

上述IO的寄存器控制逻辑与其他GPIO相同，并联上拉控制（强上拉）仅由7816模块自动控制。

41.3 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41-2FCR定义表

41.3.1 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
- PxDIN 寄存器直接反应 IO 上的电平状态

41.3.2 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 PxDO 寄存器获得上次写入的值

41.3.3 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
- 在 GPIOx_INEN 寄存器置位的情况下，软件读取 PxDIN 寄存器能够获得 IO 上的电平状态

部分引脚支持多个数字外设功能，则还需要额外的控制寄存器（ADT_AFSELx）来区分。

支持多个数字外设功能的引脚有：

GPIO	数字功能 1 PxDFS[x]=0	数字功能 2 PxDFS[x]=1	Additional AFSEL
PA0	UART4_RX	LUT3_OUT	PADFS[0]
PA1	UART4_TX	LUT2_OUT	PADFS[1]
PA2	UART0_RX	LPUART0_RX	PADFS[2]
PA3	UART0_TX	LPUART0_TX	PADFS[3]
PA4	GPT1_CH3	-	-
PA5	GPT1_CH4	-	-

GPIO	数字功能 1 PxDFS[x]=0	数字功能 2 PxDFS[x]=1	Additional AFSEL
PA6	CAN_RX	-	-
PA7	CAN_TX	-	-
PA8	SPI1_SSN	LPT32_CH1	PADFS[8]
PA9	SPI1_SCK	LPT32_CH2	PADFS[9]
PA10	LPT32_ETR	-	-
PA11	SCL	LPUART2_RX	PADFS[11]
PA12	SDA	LPUART2_TX	PADFS[12]
PA13	UART0_RX	LPUART0_RX	PADFS[13]
PA14	UART0_TX	LPUART0_TX	PADFS[14]
PA15	COMP3_OUT	-	-
PB0	SPI1_MISO	UART3_RX	PBDFS[0]
PB1	SPI1_MOSI	UART3_TX	PBDFS[1]
PB2	UART4_RX	ATIM_CH1N	PBDFS[2]
PB3	UART4_TX	ATIM_CH2N	PBDFS[3]
PB4	LPUART2_RX	ATIM_CH1	PBDFS[4]
PB5	LPUART2_TX	ATIM_CH2	PBDFS[5]
PB6	SPI2_SSN	ATIM_CH3	PBDFS[6]
PB7	SPI2_SCK	ATIM_CH4	PBDFS[7]
PB8	SPI0_SSN	ATIM_CH3N	PBDFS[8]
PB9	SPI0_SCK	GPT0_ETR	PBDFS[9]
PB10	SPI0_MISO	GPT0_CH1	PBDFS[10]
PB11	SPI0_MOSI	GPT0_CH2	PBDFS[11]
PB12	FOUT1	ATIM_ETR	PBDFS[12]
PB13	UART1_RX	LPUART1_RX	PBDFS[13]
PB14	UART1_TX	LPUART1_TX	PBDFS[14]
PB15	SCL	SPI2_MISO	PBDFS[15]
PC0	GPT1_CH1	LUT0_OUT	PCDFS[0]
PC1	GPT1_CH2	LUT1_OUT	PCDFS[1]
PC2	UART1_RX	LPUART1_RX	PCDFS[2]
PC3	UART1_TX	LPUART1_TX	PCDFS[3]
PC4	UART5_RX	COMP1_OUT	PCDFS[4]
PC5	UART5_TX	COMP2_OUT	PCDFS[5]
PC6	GPT1_ETR	-	-
PC7	SPI2_SSN	-	-
PC8	SPI2_SCK	-	-
PC9	SPI2_MISO	-	-
PC10	SPI2_MOSI	-	-
PC11	U7816_CLK	GPT0_CH3	PCDFS[11]
PC12	U7816_IO	GPT0_CH4	PCDFS[12]
PC13	U7816_CLK	LPT16_CH1	PCDFS[13]
PC14	U7816_IO	LPT16_CH2	PCDFS[14]
PC15	LPT32_CH3	CAN_RX	PCDFS[15]
PD0	UART5_RX	GPT2_CH1	PDDFS[0]

GPIO	数字功能 1 PxDFS[x]=0	数字功能 2 PxDFS[x]=1	Additional AFSEL
PD1	UART5_TX	GPT2_CH2	PDDFS[1]
PD2	SPI1_SSN	GPT2_CH3	PDDFS[2]
PD3	SPI1_SCK	GPT2_CH4	PDDFS[3]
PD4	SPI1_MISO	GPT2_ETR	PDDFS[4]
PD5	SPI1_MOSI	LPT16_ETR	PDDFS[5]
PD6	ATIM_BRK2	-	-
PD7	SWCLK	UART3_RX	PDDFS[7]
PD8	SWIO	UART3_TX	PDDFS[8]
PD9	-	-	-
PD10	CAN_RX	-	-
PD11	FOUT0	ATIM_BKR	PDDFS[11]
PD12	SDA	SPI2_MOSI	PDDFS[12]
PD13	-	-	-
PD14	-	-	-
PH15	RTCOUT	-	-
PE0	SPI0_SSN	-	-
PE1	SPI0_SCK	-	-
PE2	SPI0_MISO	-	-
PE3	SPI0_MOSI	-	-
PE4	-	-	-
PE5	LPT32_CH4	CAN_TX	PEDFS[5]
PE6	-	-	-
PE7	COMP3_OUT	-	-
PE8	-	-	-
PE9	CAN_TX	-	-

表 41-3 多个数字外设功能选择表

41.3.4 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

每个 GPIO 有两个模拟通道，分别为电阻通道和开关通道。电阻通道无法关闭，引脚上的任何信号都会传输到芯片内部，开关通道可以通过 ANAEN 信号使能或关闭。

芯片绝大部分模拟信号，比如 COM/SEG，都是连接到 GPIO 的电阻通道上，其输出电平由 DISP 模块内部关闭或使能。其他模拟信号部分走电阻通道，部分走开关通道，参见下表：

模拟功能	PAD 通道	引脚	说明
COM	电阻		模块内部有开关
SEG	电阻		
XTHF	电阻		模块内部有开关
XTLF	电阻		
ADC_Inx	电阻		模块内部采用 T 型开关隔离 crosstalk
OPAx_INN/INP	电阻		
OPAx_OUT	开关	PB0, PB1, PC4	与 LCD 显示不能同时使用
ANATST	开关	PB7,PD6	FCR=11 并且 ANEN=1 时时使能开关
VCINx	开关	PB2,PB3	FCR=11 并且 ANEN=1 时时使能开关
VDISPx	开关	PB15,PC0,PC1,PD12	FCR=11 并且 ANEN=1 时使能开关

表 41-4 模拟功能通道选择表

41.3.5 使用外部晶体引脚

FM33LG0 支持外接 32768Hz 晶体和 4~24MHz 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。

XT32KI 和 XT32KO 为模拟功能，外接 32768Hz 晶体，无法配置为 GPIO 使用。

如果要使用外部 32K 时钟输入，可从 XT32KI 灌入 32K 时钟。

41.4 VBAT 供电引脚

PD13, PD14, PH15 为 VBAT 供电引脚，其电源在 VDD 和 VBAT 之间切换。受限于电源切换开关的导通能力，这 3 个 IO 不能 source 大电流，输出电流应限制在 1mA 以下。

当芯片 VDD 掉电时，除了上述 3 个 IO 仍有供电之外，所有其他 IO 都没有供电。

41.5 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，以省去外部上拉。

41.6 WKUPx 引脚

FM33LG0 有 8 个 WKUP 引脚，能够将芯片从 Sleep/DeepSleep/RTCBKP 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。

8 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

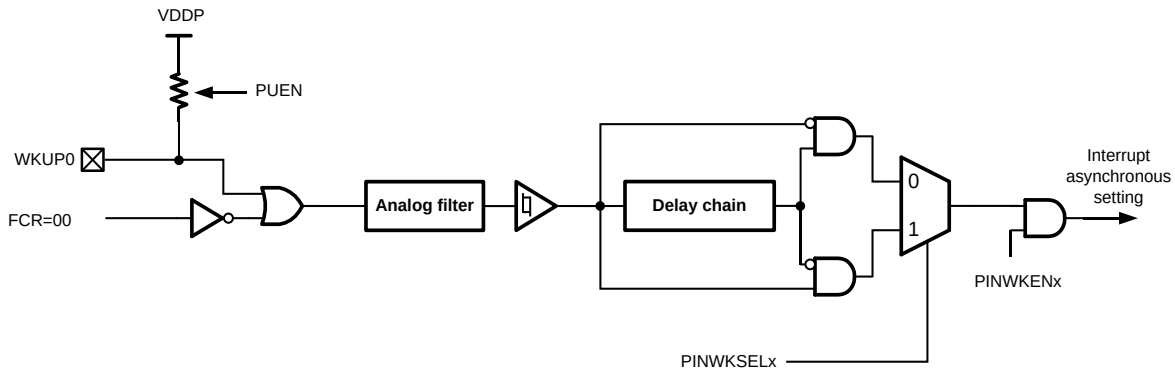


图 41-3WKUPx 功能结构框图

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。

41.7 外部引脚中断（EXTI）

41.7.1 功能说明

FM33LG0 的 5 组 GPIO（A~E）最多可以产生 24 个 EXTI 中断，每组 GPIO 分别可以产生 4 个 EXTI 中断标志，最终所有的 EXTI 中断汇总到 NVIC 的#46 入口。

中断标志和引脚对应关系如下表：

GPIO	EXTI输入选择	EXTI
PA0~PA3	EXTI_ASEL[1:0]	EXTI[0]
PA4~PA7	EXTI_ASEL[3:2]	EXTI[1]
PA8~PA11	EXTI_ASEL[5:4]	EXTI[2]
PA12~PA15	EXTI_ASEL[7:6]	EXTI[3]
PB0~PB3	EXTI_BSEL[1:0]	EXTI[4]
PB4~PB7	EXTI_BSEL[3:2]	EXTI[5]
PB8~PB11	EXTI_BSEL[5:4]	EXTI[6]
PB12~PB15	EXTI_BSEL[7:6]	EXTI[7]
PC0~PC3	EXTI_CSEL[1:0]	EXTI[8]
PC4~PC7	EXTI_CSEL[3:2]	EXTI[9]
PC8~PC11	EXTI_CSEL[5:4]	EXTI[10]
PC12~PC15	EXTI_CSEL[7:6]	EXTI[11]
PD0~PD3	EXTI_DSEL[1:0]	EXTI[12]
PD4~PD7	EXTI_DSEL[3:2]	EXTI[13]
PD8~PD11	EXTI_DSEL[5:4]	EXTI[14]
PD12	EXTI_DSEL[7:6]	EXTI[15]
PE0~PE3	EXTI_ESEL[1:0]	EXTI[16]
PE5~PE7	EXTI_ESEL[3:2]	EXTI[17]
PE8~PE9	EXTI_ESEL[5:4]	EXTI[18]
-	EXTI_ESEL[7:6]	EXTI[19]

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。

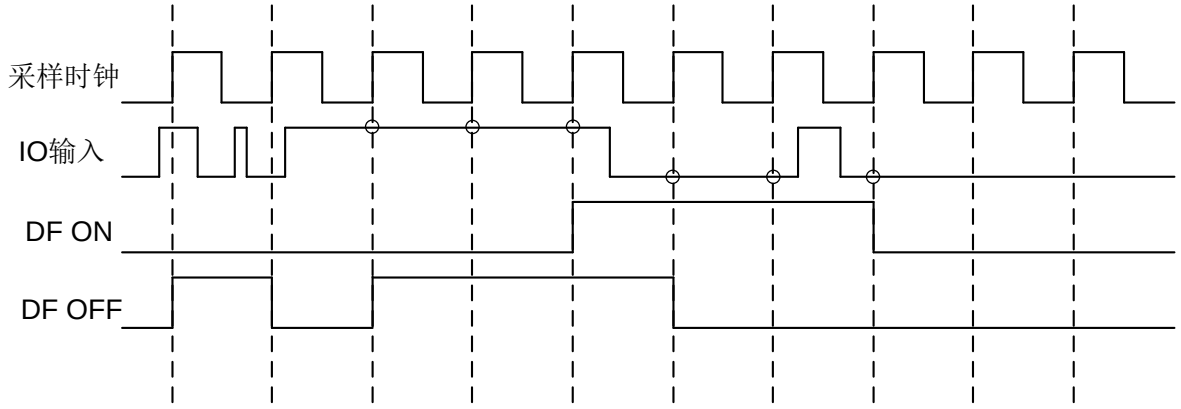


图 41-4 引脚输入数字滤波

当软件可以选择数字滤波的采样时钟为 APBCLK 或者 LSCLK。

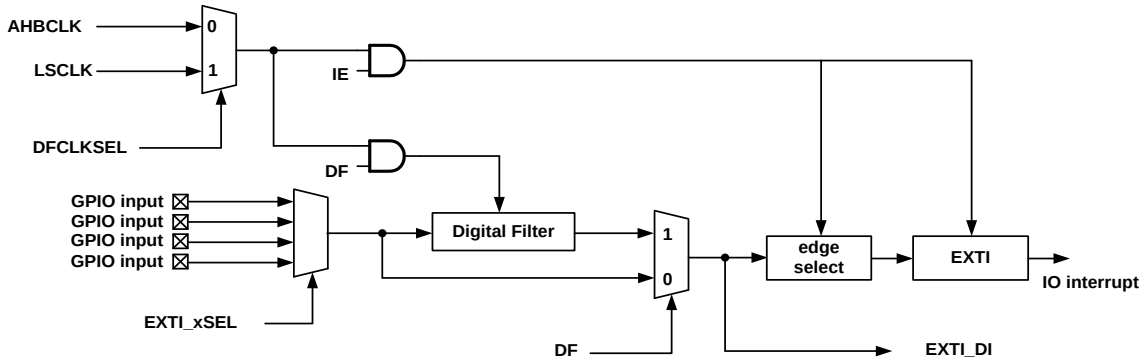


图 41-5 EXTI 信号输入示意图

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。

41.7.2 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（0x0x4000020C）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 OPCCON1.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 OPCCON1.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 OPCCON1.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

41.8 快速 GPIO 输出

FM33LG0 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。

41.9 寄存器

offset 地址	名称	符号
GPIO(模块基地址：0x40000C00)		
0x00	GPIOA 输入使能寄存器 （GPIOA Input Enable Register）	GPIOA_INEN
0x04	GPIOA 上拉使能寄存器 （GPIOA Pull-Up Enable Register）	GPIOA_PUEN
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
0x24	-	-
0x28	GPIOA 模拟开关使能寄存器 （GPIOA Analog channel Enable Register）	GPIOA_ANEN
0x2C	GPIOA 输入低阈值配置寄存器 （GPIOA Voltage Input Low Register）	GPIOA_VILR
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_PUEN
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
0x64	-	-
0x68	GPIOB 模拟开关使能寄存器 （GPIOB Analog channel Enable Register）	GPIOB_ANEN

offset 地址	名称	符号
0x6C	GPIOB 输入低阈值配置寄存器 （GPIOB Voltage Input Low Register）	GPIOB_VILR
0x80	GPIOC 输入使能寄存器 （GPIOC Input Enable Register）	GPIOC_INEN
0x84	GPIOC 上拉使能寄存器 （GPIOC Pull-Up Enable Register）	GPIOC_PUEN
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
0xA4	-	-
0xA8	GPIOC 模拟开关使能寄存器 （GPIOC Analog channel Enable Register）	GPIOC_ANEN
0xAC	GPIOC 输入低阈值配置寄存器 （GPIOC Voltage Input Low Register）	GPIOC_VILR
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_PUEN
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
0xE4	-	-
0xE8	GPIOD 模拟开关使能寄存器 （GPIOD Analog channel Enable Register）	GPIOD_ANEN
0xEC	GPIOD 输入低阈值配置寄存器	GPIOD_VILR

offset 地址	名称	符号
	（GPIOD Voltage Input Low Register）	
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_PUEN
0x108	GPIOE 开漏使能寄存器 （GPIOE Opeb-Drain Enable Register）	GPIOE_ODEN
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
0x100	GPIOE 额外数字功能寄存器 （GPIOE Digital Function Select）	GPIOE_DFS
0x104	-	-
0x108	GPIOE 模拟开关使能寄存器 （GPIOD Analog channel Enable Register）	GPIOE_ANEN
0x10C	GPIOE 输入低阈值配置寄存器 （GPIOD Voltage Input Low Register）	GPIOE_VILR
0x1C0	EXTI 输入选择寄存器 0 （External Interrupt Input Select Register0）	GPIO_EXTISEL0
0x1C4	EXTI 输入选择寄存器 1 （External Interrupt Input Select Register1）	GPIO_EXTISEL1
0x1C8	EXTI 边沿选择和使能寄存器 0 （External Interrupt Edge Select and Enable Register0）	GPIO_EXTIEDS0
0x1CC	EXTI 边沿选择和使能寄存器 1 （External Interrupt Edge Select and Enable Register1）	GPIO_EXTIEDS1
0x1D0	EXTI 数字滤波控制寄存器 （External Interrupt Digital Filter Register）	GPIO_EXTIDF
0x1D4	EXTI 中断标志寄存器 （External Interrupt and Status Register）	GPIO_EXTIISR
0x1D8	EXTI 输入信号寄存器 （External Interrupt Data Input Register）	GPIO_EXTIDI
0x200	FOUT 配置寄存器 （Frequency Output Select Register）	GPIO_FOUTSEL
0x300	WKUP 使能寄存器 （Wakeup Enable Register）	GPIO_PINWKEN

41.9.1 GPIO 输入使能寄存器（GPIOx_INEN）

名称	GPIOx_INEN(x=A,B,C,D,E)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 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

位号	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

41.9.2 GPIO 上拉使能寄存器（GPIOx_PUEN）

名称	GPIOx_PUEN(x=A,B,C,D,E)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x04 + 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	PxPUEN15	PxPUEN14	PxPUEN13	PxPUEN12	PxPUEN11	PxPUEN10	PxPUEN9	PxPUEN8
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	PxPUEN7	PxPUEN6	PxPUEN5	PxPUEN4	PxPUEN3	PxPUEN2	PxPUEN1	PxPUEN0
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
31:16	--	RFU：未实现，读为 0
15:0	PxPUEN	GPIO 上拉控制 0：关闭上拉 1：使能上拉

41.9.3 GPIO 开漏使能寄存器（GPIOx_ODEN）

名称	GPIOx_ODEN(x=A,B,C,D,E)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 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

位号	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

41.9.4 GPIO 功能选择寄存器（GPIOx_FCR）

名称	GPIOx_FCR(x=A,B,C,D,E)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 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	

位号	助记符	功能描述
31:30	Px15FCR	Px[15]引脚功能选择 00：GPIO 输入 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

位号	助记符	功能描述
		10：Digital function 11：Analog function
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
13:12	Px6FCR	Px[6]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 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

41.9.5 GPIO 输出数据寄存器（GPIOx_DO）

名称	GPIOx_DO(x=A,B,C,D,E)

offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 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

位号	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxDO	GPIO output data register

41.9.6 GPIO 输出数据置位寄存器（GPIOx_DSET）

名称	GPIOx_DSET(x=A,B,C,D,E)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 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

位号	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxSET	GPIO output data set register

位号	助记符	功能描述
		举例：向 PADSET 写 0x0000_8000，则 PADO[15]置位，其余位保持不变。

41.9.7 GPIO 输出数据复位寄存器（GPIOx_DRST）

名称	GPIOx_DRST(x=A,B,C,D,E)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 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

位号	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

41.9.8 GPIO 输入数据寄存器（GPIOx_DIN）

名称	GPIOx_DIN(x=A,B,C,D,E)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 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

位号	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

41.9.9 GPIO 额外数字功能选择寄存器（GPIOx_DFS）

名称	GPIOx_DFS(x=A,B,C,D,E)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x20 + 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	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							

位号	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	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**表 41-3**

41.9.10 GPIO 模拟开关使能寄存器（GPIOx_ANEN）

名称	GPIOx_ANEN(x=A,B,C,D,E)

offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x28 + 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	PxANEN[15:8]							
位权限	R/W-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxANEN[7:0]							
位权限	R/W-0							

位号	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxANEN	PortX 模拟开关使能 1：使能 IO 模拟开关 0：关闭 IO 模拟开关 *注：支持模拟开关的 IO 有 PB0, PB1, PB15, PD12, PC0, PC1, PC4* *对应以上 IO 的 PxANEN 寄存器有效；其余寄存器无意义。*

41.9.11 GPIO 输入低阈值配置寄存器（GPIOx_VILR）

名称	GPIOx_VILR (x=A,B,C,D,E)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x2C + 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	PxVIL15	PxVIL14	PxVIL13	PxVIL12	PxVIL11	PxVIL10	PxVIL9	PxVIL8
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	PxVIL7	PxVIL6	PxVIL5	PxVIL4	PxVIL3	PxVIL2	PxVIL1	PxVIL0
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
31:16	--	RFU：未实现，读为 0
15:0	PxVIL	GPIO 输入低阈值控制 0：输入低阈值正常 1：输入低阈值降低

41.9.12 EXTI 输入选择寄存器 0（GPIO_EXTISEL0）

名称	GPIO_EXTISEL0							
offset	0x1C0							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTI_DSEL							
位权限	R/W-00 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EXTI_CSEL							
位权限	R/W-00 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_BSEL							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI_ASEL							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:24	EXTI_DSEL[5:0]	PortD EXTI 中断输入选择 EXTI_DSEL[7:6] – 00：PD12 01：RFU 10：RFU 11：RFU EXTI_DSEL[5:4] – 00：PD8 01：PD9 10：PD10 11：PD11 EXTI_DSEL[3:2] – 00：PD4 01：PD5 10：PD6 11：PD7 EXTI_DSEL[1:0] – 00：PD0 01：PD1 10：PD2 11：PD3
23:16	EXTI_CSEL[5:0]	PortC EXTI 中断输入选择

位号	助记符	功能描述
		EXTI_CSEL[5:4] – 00：PC8 01：PC9 10：PC10 11：PC11 EXTI_CSEL[3:2] – 00：PC4 01：PC5 10：PC6 11：PC7 EXTI_CSEL[1:0] – 00：PC0 01：PC1 10：PC2 11：PC3
15:8	EXTI_BSEL[7:0]	PortB EXTI 中断输入选择 EXTI_BSEL[7:6] – 00：PB12 01：PB13 10：PB14 11：PB15 EXTI_BSEL[5:4] – 00：PB8 01：PB9 10：PB10 11：PB11 EXTI_BSEL[3:2] – 00：PB4 01：PB5 10：PB6 11：PB7 EXTI_BSEL[1:0] – 00：PB0 01：PB1 10：PB2 11：PB3
7:0	EXTI_ASEL	PortA EXTI 中断输入选择 EXTI_ASEL[7:6] – 00：PA12 01：PA13 10：PA14 11：PA15 EXTI_ASEL[5:4] – 00：PA8 01：PA9 10：PA10 11：PA11 EXTI_ASEL[3:2] – 00：PA4 01：PA5

位号	助记符	功能描述
		10：PA6 11：PA7 EXTI_ASEL[1:0] – 00：PA0 01：PA1 10：PA2 11：PA3

41.9.13 EXTI 输入选择寄存器 1（GPIO_EXTISEL1）

名称	GPIO_EXTISEL1							
offset	0x1C4							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	EXTI_ESEL							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:24	--	RFU：未实现，读为 0
7:0	EXTI_ESEL	PortE EXTI 中断输入选择 EXTI_ESEL[7:6] – 00：RFU 01：RFU 10：RFU 11：RFU EXTI_ESEL[5:4] – 00：PE8 01：PE9 10：RFU 11：RFU EXTI_ESEL[3:2] – 00：RFU 01：PE5 10：PE6 11：PE7 EXTI_ESEL[1:0] – 00：PE0 01：PE1

位号	助记符	功能描述
		10：PE2 11：PE3

41.9.14 EXTI 边沿选择和使能寄存器 0（GPIO_EXTIEDS0）

名称	GPIO_EXTIEDS0							
offset	0x1C8							
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	

位号	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

41.9.1 EXTI 边沿选择和使能寄存器 1（GPIO_EXTIEDS1）

名称	GPIO_EXTIEDS1							
offset	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	-	EXTI18_EDS	EXTI17_EDS	EXTI16_EDS
位权限	U-0	R/W-11	R/W-11	R/W-11

位号	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5:4	EXTI18_EDS	EXTI[18]边沿触发选择 00: rising 01: falling 10: both 11: disable
3:2	EXTI17_EDS	EXTI[17]边沿触发选择 00: rising 01: falling 10: both 11: disable
1:0	EXTI16_EDS	EXTI1[16]边沿触发选择 00: rising 01: falling 10: both 11: disable

41.9.2 EXTI 数字滤波控制寄存器（GPIO_EXTIDF）

名称	GPIO_EXTIDF							
offset	0x1D0							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					EXTI_DF[18:16]		
位权限	U-0					R/W -000		
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							

位号	助记符	功能描述
31:19	--	RFU：未实现，读为 0
18:0	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

41.9.3 EXTI 中断标志（GPIO_EXTIISR）

名称	GPIO_EXTIISR							
offset	0x1D4							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					EXTI[18:16]		
位权限	U-0					R/W-000		
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							

位号	助记符	功能描述
31:19	--	RFU：未实现，读为 0
18:0	EXTI	外部引脚中断标志寄存器，共可以产生 19 个引脚中断 硬件置位，软件写 1 清零

41.9.4 EXTI 输入信号寄存器（GPIO_EXTIDI）

名称	GPIO_EXTIDI							
offset	0x1D8							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					EXTI_DI[18:16]		
位权限	U-0					R-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_DI[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI_DI[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:19	--	RFU：未实现，读为 0
18:0	EXTI_DI	EXTI 输入信号只读寄存器，软件读取此寄存器可以观察 EXTI 的 19 个输入信号的当前状态 *注：当使能了数字滤波后，软件可以从这个寄存器读取到某个 IO 输入信号滤波后的状态。*

41.9.5 FOUT 配置寄存器（GPIO_FOUTSEL）

名称	GPIO_FOUTSEL							
offset	0x200							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	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	FOUT1SEL				FOUT0SEL			
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	FOUT1SEL	PB12 输出选择 0000: XTLF 0001: RCLP 0010: ADCCLK 0011: LSCLK 0100: EOC 0101: RTCTM 0110: PLLO/64 0111: EOCAL 1000: APBCLK/64 1001: ROSC_TDLV 1010: RCLF 1011: RCHF 1100: XTHF/64 1101: ADCCLK/64 1110: CLK8K 1111: ROSC_TDHV
3:0	FOUT0SEL	PD11 输出选择 0000: XTLF 0001: RCLP 0010: RCHF/64 0011: LSCLK 0100: AHBCLK/64 0101: RTCTM 0110: PLLO/64 0111: RTCCLK64Hz 1000: APBCLK/64 1001: PLLO 1010: RCLF 1011: RCHF 1100: XTHF/64 1101: COMP1O 1110: CLK8K 1111: ADC_CLK

41.9.6 WKUP 控制寄存器（GPIO_PINWKEN）

名称	GPIO_PINWKEN							
offset	0x300							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	WKSEL	-						

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	PINWKSEL[7:0]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PINWKEN[7:0]							
位权限	R/W-00000000							

位号	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 38 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 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[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

42 专用编程接口

42.1 概述

FM33LG0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

FM33LC0XX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FM33LC0xx芯片互联，并实现仿真调试。

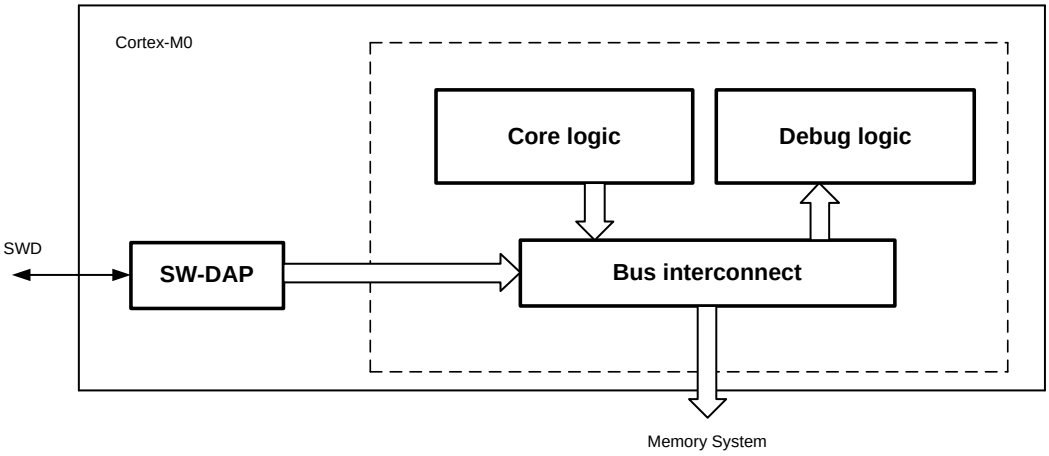


图 43-1 Cortex-M0 调试系统示意图

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

FM33LG0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PD7和PD8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引脚默认使能内部上拉（约100K欧姆），SWCLK引脚默认不使能内部上拉电阻，推荐用户在PCB上外接100K下拉电阻，以防止输入引脚浮空导致漏电增加。

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。

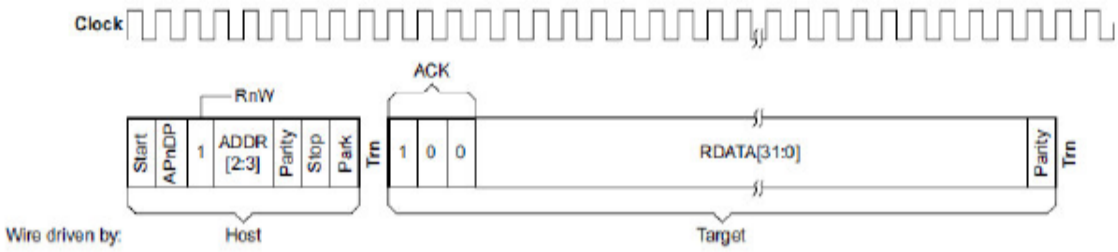
- 芯片复位，或者SWIO拉高50个SWCLK周期后，SW-DP处于RESET状态
- 拉低SWIO至少2个SWCLK周期后，SW-DP进入IDLE状态
- 当SW-DP处于RESET，主机必须先使其进入IDLE，然后对ID code寄存器进行读操作，才能激活SW-DP。否则从机会对主机的通信回应FAULT响应。

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。

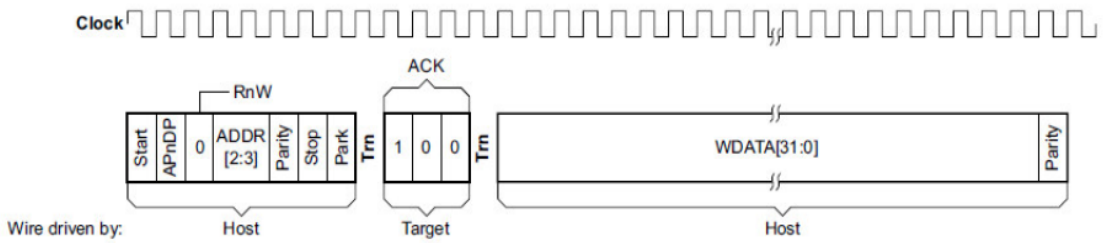


43.3.5 主机写操作

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
DBG(模块基地址：0x40000000)		
0x00	系统配置寄存器	SYSCFG
0x04	MCU Debug 配置寄存器	CR
0x08	HARDFAULT 查询寄存器	HDFR

43.8.1 系统模式配置寄存器

*SYSCFG 位于 AO domain，由 NVMIF 读取后通过局部总线写入；AHB/APB 总线不能改写，只能读取。

名称	SYSCFG							
地址	0x40000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LDT0FAIL	LDT1FAIL	DCTFAIL	-				
位权限	R-0	R-0	R-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	RED_INFO							
位权限	R-xx							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			RAMCFG	FLSCFG		MODE	
位权限	U-0			R-x	R-xx		R-00	

位号	助记符	功能描述
31	LDT0FAIL	Flash 配置信息，即 LDT0 区，校验错误标志 1：某项配置信息校验出现错误，错误项将保持默认值 0：校验正确
30	LDT1FAIL	OPTBYTE/ACLOCK，即 LDT1 区，校验错误标志 1：校验失败，Debug 接口及 ACLOCK 保护相关寄存器置 1 0：校验通过
29	DCTFAIL	DCT page trim-tag 校验错误标志 1：校验失败，非 5A5A5A5A 0：校验通过
28:16	--	RFU：未实现，读为 0
15:8	RED_INFO	Flash 冗余信息，redundancy 页最低 8bit
7:5	--	RFU：未实现，读为 0

位号	助记符	功能描述
4	RAMCFG	RAM 版本配置 0：32KB 1：16KB
3:2	FLSCFG	Flash 大小配置 00/11：256KB 01：128KB 10：64KB
1:0	MODE	芯片权限模式 00/10：User 01：Manufacture 11：Original

43.8.2 MCU DEBUG 配置寄存器

FM33LG0 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							
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	-							DBG_SLEEP
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DBG_LPT16_STOP	DBG_BT16_STOP	DBG_AT_STOP	DBG_LPT32_STOP	DBG_GT2_STOP	DBG_GT1_STOP	DBG_GT0_STOP	DBG_BT32_STOP
位权限	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						DBG_WWDT_STOP	DBG_IWDT_STOP
位权限	U-0						R/W-1	R/W-1

位号	助记符	功能描述
31:17	--	RFU：未实现，读为 0
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_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
14	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

位号	助记符	功能描述
13	DBG_AT_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
12	DBG_LPT32_STOP	Debug 状态下 LPTIM32 使能控制位 1：Debug 时关闭 LPTIM32 0：Debug 时保持 LPTIM32 原来状态
11	DBG_GT2_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
10	DBG_GT1_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
9	DBG_GT0_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
8	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
7:2	--	RFU：未实现，读为 0
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

43.8.3 HardFault 查询寄存器

名称	HDFR							
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	-	DABORT_ADDR_FLAG	DABORT_RESP_FLAG	SVCUNDEF_FLAG	BKPT_FLAG	TBIT_FLAG	SPECIAL_OP_FLAG	HDF_REQUEST_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

位号	助记符	功能描述
31:7	--	RFU：未实现，读为 0
6	DABORT_A	地址非对齐访问错误标志，写 1 清零

位号	助记符	功能描述
	DDR_FLAG	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	SVCUNDEF_FLAG	SVC instructions 未定义标志，写 1 清零 if the SVCall priority is lower than the currently activelevel, or if HardFault or NMI is active, or PRIMASK is set, the core should treat SVC instructions as though theywere UNDEFINED。
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	SPECIAL_OP_FLAG	特殊指令标志，写 1 清零 1：执行了特殊指令代码，如试图在 XN 区域内取指 0：无特殊指令代码被执行
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FM33LG0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 存储器容量查询

通过查询SYSCON寄存器，可以获得器件Flash和RAM容量信息。

名称	SYSCFG							
地址	0x40000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	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	-			RAMCFG	FLSCFG		MODE	
位权限	U-0			R-x	R-xx		R-00	

位号	助记符	功能描述
31:5	--	RFU：未实现，读为 0
4	RAMCFG	RAM 版本配置 0：32KB 1：16KB
3:2	FLSCFG	Flash 大小配置 00/11：256KB 01：128KB 10：64KB
1:0	--	RFU：未实现，读为 0

44.2 器件 UID

FM33LG0系列每颗MCU的器件UID都是全球唯一的，由原厂写入，出厂后不可改写。

UID共128 bits，保存在Flash特殊扇区，软件运行时可以读取此UID，用于实现代码保护或安全启动类应用。

UID访问地址是0x1FFF_FE80~0x1FFF_FE8F，详细请联系上海复旦微电子公司。

版本列表

版本号	发布日期	页数	章节或图表	更改说明
1.0	2020-6	825		Initial release
1.1	2020-9	833		更新电参数
1.2	2020-9	831		更新引脚Pin Number
1.3	2020-12	836		修订描述错误
1.4	2021-3	838		补充XTHF和OPA参数
1.5	2021-3	838	2.4	补充封装热阻特性

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/