Shanghai Fudan Microelectronics Group Company Limited

复旦微电子

FM33A0xxEV

512KB Cortex-M0 智能电表 MCU

产品说明书

2024. 8

产品说明书版本列表

版本号	发布日期	页数	章节或图表	更改说明
1.0	2020.02			首次发布
1.1	2020.02		1.4	增加芯片框图
1.2	2020.04			更新电参数和 SVD 章节
1.3	2020.05		1.5.4	增加 LQFP48 封装
1.4	2020.08			优化 IO 参数描述
1.5	2020.09		1.5.3	LQFP64 封装中 PB6 改为 PB5
1.6	2020.10		29.5.11	更新 LCD boost 电压档位
1.7	2021.02		19.7.1	LPUART 增加 IOSWAP 寄存器
1.8	2021.07			删除 18.6.5 低功耗休眠唤醒(UART0/1)
1.9	2022.07		31.4	修订 SWCLK 上电复位后没有上拉
2.0	2022.08		5.5.1 9.125	1 明确 48M 时 FLASH wait 为 1 2 PLLH 的范围为 32M~64M
2.1	2023.06		29.2	BSTIM32 结构框图删除 CK_BSTIM 后的分频器
2.2	2023.08		1.1	更新最低工作电压
2.3	2023.10		2.4.5	更新时钟电路描述
2.4	2023.10		16.3.5	更新参数条件描述
2.5	2024.3		11.3	更新 SVD 功能描述
2.6	2024.8		1.5.5 34.2	修正 VDD15 管脚描述:管脚串 10 欧电阻再接 1uF 对地电容; 修正器件 UID 访问地址描述错误;

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，不保证其中不含任何瑕疵，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

产品说明书版本列表 **2**

章节列表 **4**

表目录 **18**

图目录 **20**

1 产品综述 **24**

1.1 概述 24
1.2 产品型号列表 26
1.3 产品特性对照表 26
1.4 芯片结构框图 27
1.5 引脚和封装定义 28
1.5.1 LQFP100 封装图 *28*
1.5.2 LQFP80 封装图 *29*
1.5.3 LQFP64 封装图 *30*
1.5.4 LQFP48 封装图 *31*
1.5.5 引脚功能定义 *32*
1.5.6 封装尺寸图 *39*

2 电参数 **45**

2.1 参数说明 45
2.2 参数测试条件 45
2.2.1 供电方案 *45*
2.3 极限参数 46
2.4 性能参数 47
2.4.1 典型工作条件 *47*
2.4.2 功耗参数 *48*
2.4.3 复位和电源监控 *52*
2.4.4 低功耗模式唤醒时间 *53*
2.4.5 外部时钟源特性 *54*
2.4.6 内部时钟源特性 *55*
2.4.7 PLL 特性 *59*
2.4.8 ADC 特性 *60*
2.4.9 温度传感器 *60*
2.4.10 Flash 存储器特性 *61*
2.4.11 GPIO 特性 *62*
2.4.12 LCD 特性 *64*

3 电源管理单元（PMU） **66**

3.1 电源方案 66
3.1.1 电源域划分 *66*
3.1.2 电源结构图 *67*
3.2 功耗模式 68
3.2.1 概述 *68*
3.2.1 功耗模式与系统频率 *69*
3.2.2 Active 模式 *70*
3.2.3 LP Run 模式 *70*
3.2.4 SLEEP 模式 *71*
3.2.5 DEEPSLEEP 模式 *72*

3.3 唤醒源 73
3.4 休眠唤醒后的时钟控制 73
3.5 寄存器 74
3.5.1 低功耗控制寄存器（PMU_CR） 74
3.5.2 唤醒时间控制寄存器（PMU_WKTR） 75
3.5.3 休眠唤醒源标志寄存器（PMU_WKFR） 75
3.5.4 PMU 中断使能寄存器（PMU_IER） 76
3.5.5 PMU 中断标志寄存器（PMU_ISR） 77

4 CPU 80

4.1 概述 80
4.1.1 处理器配置 80
4.2 寄存器 81
4.3 异常和中断 82
4.3.1 中断向量表 82
4.3.2 中断优先级 83
4.3.3 错误处理 83
4.3.4 锁定（Lockup） 84
4.4 MPU 85
4.4.1 MPU 寄存器 85
4.5 调试特性 90
4.5.1 调试功能引脚 90
4.5.2 调试状态下的看门狗控制 90
4.5.3 DEBUG 的复位 90
4.6 寄存器 91
4.6.1 MCU DEBUG 配置寄存器（DBG_CR） 91
4.6.2 HardFault 查询寄存器（DBG_HDFR） 92

5 总线与存储（BUS AND MEMORIES） 94

5.1 系统总线 94
5.2 存储空间分配 96
5.2.1 概述 96
5.2.2 外设模块寄存器地址分配 98
5.3 RAM 100
5.3.1 系统 RAM 100
5.3.2 算法 RAM 100
5.3.3 RTC RAM 100
5.4 FLASH 101
5.4.1 概述 101
5.4.2 Flash 擦除与编程 107
5.4.3 Flash 的内容保护 112
5.5 寄存器 115
5.5.1 Flash 读取控制寄存器（FLS_RDCR） 115
5.5.1 预取指控制寄存器（FLS_PFCR） 116
5.5.1 用户配置字寄存器（FLS_OPTBR） 116
5.5.1 Flash 擦写控制寄存器（FLS_EPCR） 117
5.5.1 Flash Key 输入寄存器（FLS_KEY） 118
5.5.1 Flash 中断使能寄存器（FLS_IER） 118
5.5.2 Flash 标志寄存器（FLS_ISR） 119
5.5.3 ACLOCK 寄存器 1（FLS_ACLOCK1） 120
5.5.4 ACLOCK 寄存器 2（FLS_ACLOCK2） 120
5.5.5 ACLOCK 寄存器 3（FLS_ACLOCK3） 121
5.5.6 ACLOCK 寄存器 4（FLS_ACLOCK4） 122

6 复位管理单元（RMU） 123

6.1 概述 123
6.2 模块框图 124
6.3 上下电复位（BOR+PDR） 125
6.4 软件复位 125
6.5 NRST 引脚复位 125
6.6 寄存器 126
6.6.1 PDR 配置寄存器（RMU_PDRCR） 126
6.6.2 BOR 配置寄存器（RMU_BORCR） 127
6.6.3 复位配置寄存器（RMU_RSTCFGR） 127
6.6.4 软件复位寄存器（RMU_SOFTRST） 128
6.6.5 复位标志寄存器（RMU_RSR） 128
6.6.6 外设复位使能寄存器（RMU_PRSTEN） 129
6.6.7 AHB 外设复位寄存器（RMU_AHBRST） 129
6.6.8 APB 外设复位寄存器 1（RMU_APBRST1） 130
6.6.9 APB 外设复位寄存器 2（RMU_APBRST2） 132

7 独立看门狗（IWDT） 134

7.1 概述 134
7.2 结构框图 134
7.3 IWDT 功能描述 134
7.4 寄存器 135
7.4.1 IWDT 清除寄存器（IWDT_SERV） 135
7.4.2 IWDT 配置寄存器（IWDT_CFGR） 135
7.4.3 IWDT 计数值寄存器（IWDT_CNTR） 136

8 窗口看门狗（WWDT）复位 137

8.1 功能描述 137
8.2 WWDT 工作方式 137
8.3 寄存器 139
8.3.1 WWDT 控制寄存器（WWDT_CR） 139
8.3.2 WWDT 配置寄存器（WWDT_CFGR） 140
8.3.3 WWDT 计数寄存器（WWDT_CNTR） 141
8.3.4 WWDT 中断使能寄存器（WWDT_IER） 141
8.3.5 WWDT 中断标志寄存器（WWDT_ISR） 141
8.3.6 WWDT 预分频寄存器（WWDT_PSCR） 142

9 时钟管理单元（CMU） 143

9.1 概述 143
9.2 时钟架构 144
9.2.1 主要时钟说明 145
9.2.2 外设模块的总线时钟和工作时钟 145
9.2.3 休眠模式下的外设时钟 146
9.3 高频 RC 振荡器(RCHF) 147
9.3.1 概述 147
9.3.1 软件使用说明 147
9.4 高频晶体振荡电路(XTHF) 148
9.4.1 概述 148
9.4.2 工作方式 148
9.4.3 停振检测（HFDET） 148
9.4.4 引脚 148
9.5 锁相环(PLL_H) 149
9.5.1 概述 149

9.5.2 软件使用说明 149
9.6 锁相环(PLL_L) 150
9.6.1 概述 150
9.6.2 软件使用说明 150
9.7 中频 RC 振荡器(RCMF) 151
9.7.1 概述 151
9.8 低频晶体振荡电路(XTLF) 152
9.8.1 概述 152
9.8.2 工作方式 152
9.8.3 停振检测 152
9.9 低功耗 RC 振荡器(RCLP) 153
9.9.1 概述 153
9.9.2 软件使用说明 153
9.10 低功耗模式下的时钟源 154
9.11 休眠唤醒的时钟处理 154
9.12 寄存器 155
9.12.1 系统时钟配置寄存器（CMU_SYSCLKCR） 155
9.12.2 RCHF 时钟控制寄存器（CMU_RCHFCR） 156
9.12.3 RCHF 调校寄存器（CMU_RCHFTR） 157
9.12.4 PLL_L 控制寄存器（CMU_PLLLCR） 157
9.12.5 PLL_H 控制寄存器（CMU_PLLHCR） 158
9.12.6 XTHF 控制寄存器（CMU_XTHFCR） 159
9.12.7 时钟控制中断使能寄存器（CMU_IER） 160
9.12.8 时钟控制中断标志寄存器（CMU_ISR） 160
9.12.9 外设总线时钟控制寄存器 1（CMU_PCLKCR1） 161
9.12.10 外设总线时钟控制寄存器 2（CMU_PCLKCR2） 161
9.12.11 外设总线时钟控制寄存器 3（CMU_PCLKCR3） 162
9.12.12 外设总线时钟控制寄存器 4（CMU_PCLKCR4） 163
9.12.13 外设独立工作时钟控制寄存器 1（CMU_OPCCR1） 164
9.12.14 外设独立工作时钟控制寄存器 2（CMU_OPCCR2） 165
10 其他模拟控制寄存器（CDIF） 167
10.1 概述 167
10.2 访问方式 167
10.3 CDIF 寄存器 167
10.3.1 接口控制寄存器（CDIF_CR） 167
10.3.2 接口预分频寄存器（CDIF_PSCR） 168
10.4 其他模拟控制寄存器 168
10.4.1 RCMF 控制寄存器（VRTC_RCMFCR） 169
10.4.2 RCLP 控制寄存器（VRTC_RCLPCR） 170
10.4.3 RCLP 调校寄存器（VRTC_RCLPTR） 170
10.4.4 XTLF 控制寄存器（VRTC_XTLFCR） 171
10.4.5 ADC 时钟控制寄存器（VRTC_ADCCR） 172
10.4.6 低频检测中断使能寄存器（VRTC_LFDIER） 172
10.4.7 低频检测中断标志寄存器（VRTC_LFDISR） 173
11 电源电压监测（SVD） 174
11.1 概述 174
11.2 模块框图 174
11.3 功能描述 175
11.3.1 电源检测 175
11.3.2 外部电压检测 176
11.3.3 检测阈值 177

11.4 寄存器 181
11.4.1 SVD 配置寄存器（SVD_CFGR） 181
11.4.2 SVD 控制寄存器（SVD_CR） 182
11.4.3 SVD 中断使能寄存器（SVD_IER） 182
11.4.4 SVD 状态和标志寄存器（SVD_ISR） 183
11.4.5 SVD 参考电压选择寄存器（SVD_VSR） 183

12 AES 运算单元（AES） 185

12.1 功能描述 185
12.2 工作模式 185
12.3 AES 数据流处理模式 186
12.3.1 ECB 模式 186
12.3.2 CBC 模式 187
12.3.3 暂停模式 189
12.3.4 CTR 模式 190
12.3.5 CTR 模式下的暂停模式 191
12.3.6 GCM 模式 191
12.3.7 MultH 模块 194
12.3.8 推荐的 GCM 流程 195
12.4 数据类型 196
12.5 工作流程 197
12.5.1 模式 1：加密 197
12.5.2 模式 2：密钥扩展 198
12.5.3 模式 3：解密 199
12.5.4 模式 4：密钥扩展+解密 199
12.5.5 使用 MultH 模块 200
12.6 DMA 接口 201
12.6.1 MultH 模块与 DMA 间接口 202
12.7 错误标志 202
12.8 寄存器 203
12.8.1 AES 控制寄存器（AES_CR） 204
12.8.1 AES 中断使能寄存器（AES_IER） 205
12.8.2 AES 中断标志寄存器（AES_ISR） 205
12.8.3 AES 数据输入寄存器（AES_DIR） 206
12.8.4 AES 数据输出寄存器（AES_DOR） 206
12.8.5 AES 秘钥寄存器（AES_KEYx） 207
12.8.6 AES 初始向量寄存器（AES_IVRx） 208
12.8.7 AES MultH 参数寄存器（AES_Hx） 208

13 公钥算法加速器（PAE） 209

13.1 概述 209
13.2 软件使用方法 210
13.2.1 蒙哥马利模加 210
13.2.2 蒙哥马利模加 210
13.2.3 蒙哥马利模乘 211
13.2.4 Jacobin 坐标倍点 211
13.2.5 混合坐标点加 212
13.2.6 蒙哥马利点乘倍点运算（ECDBL） 212
13.2.7 蒙哥马利点乘倍点点加运算（ECADDDBL） 213
13.2.8 蒙哥马利点乘 Y 坐标恢复运算（ECYRecover） 214
13.2.9 连续 RSA 214
13.2.10 连续 ECC 215
13.3 寄存器 216

13.3.1 PAE 控制状态寄存器（PAE_CSR）.....216
13.3.2 PAE 模长寄存器（PAE_MLR）.....217
13.3.3 PAE 模参寄存器（PAE_MPR）.....217
13.3.4 模式 0（单次模运算）配置寄存器（PAE_M0CFG）.....218
13.3.5 模式 1（单次点运算）配置寄存器（PAE_M1CFG）.....219
13.3.6 模式 2（32bit 连续 RSA）配置寄存器（PAE_M2CFG）.....219
13.3.7 模式 3（32bit 连续 ECC）配置寄存器（PAE_M3CFG）.....220
13.3.8 PAE 输入数据寄存器（PAE_WORD）.....221

14 散列算法加速器（HASH）.....222

14.1 概述.....222
14.2 结构框图.....222
14.3 使用说明.....223
14.4 寄存器.....225
14.4.1 HASH 控制状态寄存器（HASH_CSR）.....225
14.4.2 HASH 数据类型寄存器（HASH_DTR）.....225

15 随机数发生器（TRNG）.....227

15.1 概述.....227
15.2 功能描述.....227
15.2.1 随机数产生.....227
15.2.2 工作时钟.....228
15.2.3 随机数读取.....228
15.2.4 CRC 运算.....228
15.3 寄存器.....229
15.3.1 随机数控制寄存器（RNG_CR）.....229
15.3.2 随机数/CRC 结果输出寄存器（RNG_DOR）.....230
15.3.3 RNG 标志寄存器（RNG_SR）.....230
15.3.4 CRC 控制寄存器（RNG_CRC_CR）.....231
15.3.5 CRC 输入数据寄存器（RNG_CRC_DIR）.....231
15.3.6 CRC 标志寄存器（RNG_CRC_SR）.....231

16 模拟比较器（COMPARATOR）.....233

16.1 概述.....233
16.2 结构框图.....234
16.3 功能描述.....235
16.3.1 基本功能.....235
16.3.2 时钟和复位.....235
16.3.3 引脚和内部信号连接.....235
16.3.4 输出数字滤波.....236
16.3.5 功耗和速度模式.....236
16.4 寄存器.....237
16.4.1 COMP1 控制寄存器（COMP_CR1）.....237
16.4.2 COMP2 控制寄存器（COMP_CR2）.....238
16.4.3 COMP 中断配置寄存器（COMP_ICR）.....239
16.4.4 COMP 中断标志寄存器（COMP_ISR）.....240
16.4.5 COMP 功耗控制寄存器(COMP_PCR).....241

17 双线串行总线（I²C）.....242

17.1 概述.....242
17.2 结构框图.....242
17.3 引脚定义.....243
17.4 时钟结构.....244
17.5 接口时序.....245

17.5.1 接口时序图 245
17.5.2 接口时序描述 246
17.6 I²C 初始化 248
17.6.1 IO 配置 248
17.6.2 主机波特率配置 248
17.7 I²C 主机功能 249
17.7.1 7bit 寻址 249
17.7.2 10bit 寻址 252
17.7.3 DMA 255
17.7.4 SCL 延展（Slave Clock Stretching） 259
17.7.5 超时机制 259
17.7.6 可编程时序和波特率发生 259
17.8 寄存器 261
17.8.1 I2C 主机配置寄存器 (I2Cx_CFGR) 261
17.8.2 I2C 主机控制寄存器 (I2Cx_CR) 262
17.8.3 I2C 主机中断使能寄存器 (I2Cx_IER) 263
17.8.4 I2C 主机中断标志寄存器 (I2Cx_ISR) 264
17.8.5 I2C 主机状态寄存器 (I2Cx_SR) 264
17.8.6 I2C 主机波特率设置寄存器 (I2Cx_BRG) 265
17.8.7 I2C 主机收发缓冲寄存器 (I2Cx_BUF) 266
17.8.8 I2C 主机时序控制寄存器 (I2Cx_TIMING) 266
17.8.9 I2C 主机超时寄存器 (I2Cx_TO) 267
18 通用异步收发传输器（UART） 268
18.1 概述 268
18.2 结构框图 269
18.3 引脚定义 270
18.4 UART 类型区分 271
18.5 UART 字符描述 271
18.6 功能描述 273
18.6.1 时钟结构 273
18.6.2 位接收采样 273
18.6.3 数据发送 274
18.6.4 数据接收 276
18.6.5 使用 DMA 进行 UART 收发 276
18.6.6 DMA 模式下的发送完成中断 277
18.7 波特率发生 278
18.7.1 波特率发生 278
18.7.1 波特率自适应 279
18.8 红外调制 280
18.9 接收超时 281
18.10 发送延迟 281
18.11 寄存器 282
18.11.1 红外调制寄存器（UART_IRCR） 284
18.11.1 UARTx 控制状态寄存器（UARTx_CSR） 284
18.11.2 UARTx 中断使能寄存器（UARTx_IER） 285
18.11.3 UARTx 中断标志寄存器（UARTx_ISR） 286
18.11.4 UARTx 超时和延迟寄存器（UARTx_TODR） 287
18.11.5 UARTx 接收缓冲寄存器（UARTx_RXBUF） 287
18.11.6 UARTx 发送缓冲寄存器（UARTx_TXBUF） 288
18.11.7 UATRx 波特率产生寄存器（UARTx_BGR） 288
19 低功耗 UART（LPUART） 290
19.1 概述 290

19.2 结构框图 291
19.3 工作时钟 292
19.4 引脚定义 293
19.5 字符描述 294
19.6 功能描述 295
19.6.1 位接收采样 295
19.6.2 接收流程 295
19.6.3 发送流程 295
19.6.4 使用 DMA 进行 LPUART 收发 295
19.6.5 休眠模式下的数据接收唤醒 296
19.6.6 LPRUN 模式下的数据 DMA 收发 296
19.6.7 DMA 模式下的发送完成中断 297
19.7 寄存器 298
19.7.1 LPUARTx 控制状态寄存器（LPUARTx_CSR） 298
19.7.2 LPUARTx 中断使能寄存器（LPUARTx_IER） 300
19.7.3 LPUARTx 中断标志寄存器（LPUARTx_ISR） 301
19.7.4 LPUARTx 波特率调制寄存器（LPUARTx_BMR） 301
19.7.5 LPUARTx 接收缓冲寄存器（LPUARTx_RXBUF） 302
19.7.6 LPUARTx 发送缓冲寄存器（LPUARTx_TXBUF） 303
19.7.7 LPUARTx 数据匹配寄存器（LPUARTx_DMR） 303

20 串行外设接口（SPI） 304

20.1 概述 304
20.2 结构框图 305
20.3 引脚定义 305
20.4 接口时序 307
20.4.1 CPHA=0 307
20.4.2 CPHA=1 307
20.4.1 4 线半双工模式（主机） 308
20.5 功能描述 310
20.5.1 I/O 配置 310
20.5.2 全双工数据通信 311
20.5.3 TX-ONLY 模式 312
20.5.4 RX-ONLY 模式 312
20.5.5 主机 SSN 控制 312
20.5.6 数据冲突 313
20.5.7 使用 DMA 进行 SPI 收发 314
20.6 寄存器 316
20.6.1 SPIx 控制寄存器（SPIx_CR1） 317
20.6.2 SPIx 控制寄存器 2（SPIx_CR2） 318
20.6.3 SPIx 控制寄存器 3（SPIx_CR3） 320
20.6.4 SPIx 中断控制寄存器（SPIx_IER） 320
20.6.5 SPI 中断标志寄存器（SPIx_ISR） 321
20.6.6 SPIx 发送缓存寄存器（SPIx_TXBUF） 322
20.6.7 SPIx 接收缓存寄存器（SPIx_RXBUF） 322

21 智能卡接口（ISO7816） 324

21.1 概述 324
21.2 结构框图 324
21.3 引脚定义 325
21.4 接口时序 325
21.5 功能描述 326
21.5.1 数据接收 326
21.5.2 数据发送 327

21.5.3 使用 DMA 进行 7816 收发 328
21.6 寄存器 330
21.6.1 U7816 控制寄存器（U7816_CR） 330
21.6.2 U7816 帧格式寄存器（U7816_FFR） 331
21.6.3 U7816 额外保护时间寄存器（U7816_EGTR） 332
21.6.4 U7816 工作时钟分频寄存器（U7816_PSC） 333
21.6.5 U7816 波特率寄存器（U7816_BGR） 333
21.6.6 U7816 数据接收缓存寄存器（U7816_RXBUF） 334
21.6.7 U7816 数据发送缓存寄存器（U7816_TXBUF） 334
21.6.8 U7816 中断使能寄存器（U7816_IER） 335
21.6.9 U7816 中断状态标志寄存器（U7816_ISR） 335

22 QUADSPI（QSPI） 338

22.1 概述 338
22.2 结构框图 338
22.3 引脚定义 339
22.4 功能描述 339
22.4.1 工作时钟 339
22.4.2 工作模式 339
22.4.3 总线配置 344
22.4.4 帧格式 346
22.4.5 延迟数据采样 349
22.4.6 BUSY 位和 ABORT 功能 349
22.4.7 nCS 控制 349
22.5 高效的 XIP 实现 351
22.5.1 应用举例 351
22.6 寄存器 354
22.6.1 QuadSPI 控制寄存器（QSPI_CR） 354
22.6.2 QuadSPI 配置寄存器（QSPI_CFGR） 356
22.6.3 QuadSPI 状态寄存器（QSPI_SR） 356
22.6.4 QuadSPI 数据长度寄存器（QSPI_DATALEN） 357
22.6.5 QuadSPI 通信配置寄存器（QSPI_CCR） 358
22.6.6 QuadSPI 地址寄存器（QSPI_ADDR） 359
22.6.7 QuadSPI alternate bytes 寄存器（QSPI_ABR） 359
22.6.8 QuadSPI 数据寄存器（QSPI_DR） 360
22.6.9 QuadSPI 状态 MASK 寄存器（QSPI_SMSK） 360
22.6.10 QuadSPI 状态匹配寄存器（QSPI_SMAT） 361
22.6.11 QuadSPI 轮询间隔寄存器（QSPI_PITV） 361
22.6.12 QuadSPI 超时寄存器（QSPI_TO） 362

23 直接存储访问控制器 1（DMA1） 363

23.1 概述 363
23.2 工作原理 363
23.3 结构框图 365
23.4 工作流程 366
23.5 访问带宽 367
23.6 通道控制 368
23.6.1 DMA 请求映射 368
23.6.2 通道优先级 370
23.6.3 传输方向定义 370
23.6.4 循环模式 370
23.6.5 影子寄存器 370
23.7 寄存器 372

23.7.1 DMA 全局控制寄存器（DMA_GCR）.....374
23.7.2 通道 x 控制寄存器（DMA_CHxCR）.....374
23.7.3 通道 x 存储器指针寄存器（DMA_CHxMAR）.....376
23.7.4 通道 11 控制寄存器（DMA_CH11CR）.....376
23.7.5 通道 11 Flash 指针寄存器（DMA_CH11FAR）.....377
23.7.6 通道 11 RAM 指针寄存器（DMA_CH11RAR）.....378
23.7.7 DMA 中断状态标志寄存器（DMA_ISR）.....378
23.7.8 通道 x 控制影子寄存器（DMA_CHxCSR）.....379
23.7.9 通道 x 存储器指针影子寄存器（DMA_CHxMASR）.....380

24 循环冗余校验（CRC）.....381

24.1 概述.....381
24.2 软件配置过程.....382
24.3 GOLDEN 数据.....383
24.4 DMA 接口.....383
24.5 FLASH 数据完整性校验.....384
24.6 寄存器.....385
24.6.1 CRC 数据寄存器（CRC_DR）.....385
24.6.2 CRC 控制状态寄存器（CRC_CR）.....385
24.6.3 CRC LFSR 寄存器（CRC_LFSR）.....387
24.6.4 CRC 输出异或寄存器（CRC_XOR）.....387
24.6.5 CRC 多项式寄存器（CRC_POLY）.....388

25 通用定时器组（TIMER ARRAY）.....389

25.1 功能描述.....389
25.2 BASIC TIMER 模块.....391
25.2.1 概述.....391
25.2.2 结构框图.....391
25.2.3 工作模式.....393
25.2.4 功能说明.....399
25.2.5 寄存器.....400
25.3 EXTENDED TIMER 模块.....414
25.3.1 概述.....414
25.3.2 结构框图.....415
25.3.3 输入信号.....416
25.3.4 外部引脚输入数字滤波.....416
25.3.5 功能说明.....416
25.3.6 寄存器.....418

26 基本定时器（BSTIM32）.....427

26.1 概述.....427
26.2 结构框图.....427
26.3 时钟和复位.....428
26.4 功能描述.....429
26.4.1 定时单元.....429
26.4.2 定时器工作模式.....431
26.4.3 计数器工作时钟.....433
26.4.4 Debug 模式.....434
26.5 寄存器.....435
26.5.1 BSTIM 控制寄存器 1（BSTIM_CR1）.....435
26.5.2 BSTIM 控制寄存器 2（BSTIM_CR2）.....436
26.5.3 BSTIM 中断使能寄存器（BSTIM_IER）.....437
26.5.4 BSTIM 中断标志寄存器（BSTIM_ISR）.....437
26.5.5 BSTIM 事件产生寄存器（BSTIM_EGR）.....438

26.5.6 BSTIM 计数器寄存器（BSTIM_CNTR）.....438
26.5.7 BSTIM 预分频寄存器（BSTIM_PSCR）.....439
26.5.8 BSTIM 自动重载寄存器（BSTIM_ARR）.....439

27 低功耗定时器（LPTIM32）.....440

27.1 概述.....440
27.2 结构框图.....441
27.3 时钟和复位.....441
27.4 相关引脚.....442
27.5 定时器功能.....442
27.5.1 普通定时器.....442
27.5.2 外部脉冲触发计数.....442
27.5.3 外部异步脉冲计数.....443
27.5.4 Timeout 模式.....443
27.6 捕捉比较功能.....444
27.6.1 32bit PWM.....444
27.6.2 输入捕捉.....445
27.7 寄存器.....447
27.7.1 LPTIM 配置寄存器（LPTIM_CFGR）.....447
27.7.2 LPTIM 计数值寄存器（LPTIM_CNTR）.....448
27.7.3 LPTIM 捕捉比较控制和状态寄存器（LPTIM_CCSR）.....449
27.7.4 LPTIM 目标值寄存器（LPTIM_ARR）.....451
27.7.5 LPTIM 中断使能寄存器（LPTIM_IER）.....451
27.7.6 LPTIM 中断标志寄存器（LPTIM_ISR）.....452
27.7.7 LPTIM 控制寄存器（LPTIM_CR）.....454
27.7.8 LPTIM 捕捉比较寄存器 1（LPTIM_CCR1）.....454
27.7.9 LPTIM 捕捉比较寄存器 2（LPTIM_CCR2）.....455
27.7.10 LPTIM 捕捉比较寄存器 3（LPTIM_CCR3）.....455
27.7.11 LPTIM 捕捉比较寄存器 4（LPTIM_CCR4）.....455

28 实时时钟（RTC）.....457

28.1 概述.....457
28.2 结构框图.....457
28.3 功能说明.....458
28.3.1 时基计数器（LTBC）.....458
28.3.2 LTBC 数字调校.....458
28.3.3 LTBC 虚拟调校.....460
28.3.4 亚秒时间计数器.....460
28.3.5 BCD 时间.....460
28.3.6 RTC 使能与停止.....461
28.3.7 RTC 时间设置.....462
28.3.8 RTC 时间读取.....462
28.3.9 闰年判断.....463
28.4 寄存器.....464
28.4.1 RTC 写使能寄存器（RTC_WER）.....465
28.4.2 RTC 中断使能寄存器（RTC_IER）.....465
28.4.3 RTC 中断标志寄存器（RTC_ISR）.....466
28.4.4 RTC BCD 时间秒寄存器（RTC_BCDSEC）.....468
28.4.5 RTC BCD 时间分钟寄存器（RTC_BCDMIN）.....468
28.4.6 RTC_BCD 时间小时寄存器（RTC_BCDHOUR）.....468
28.4.7 RTC BCD 时间天寄存器（RTC_BCDDATE）.....469
28.4.8 RTC BCD 时间星期寄存器（RTC_BCDWEEK）.....469
28.4.9 RTC BCD 时间月寄存器（RTC_BCDMONTH）.....470
28.4.10 RTC BCD 时间年寄存器（RTC_BCDYEAR）.....470

28.4.11 RTC 闹钟寄存器（RTC_ALARM）.....471
28.4.12 RTC 时钟信号输出控制寄存器（RTC_TMSEL）.....471
28.4.13 RTC LTBC 数值调整寄存器（RTC_ADJUST）.....472
28.4.14 RTC LTBC 数值调整方向寄存器（RTC_ADSIGN）.....473
28.4.15 RTC 虚拟调校寄存器（RTC_VCAL）.....473
28.4.16 RTC 调校步长寄存器（RTC_CALSTEP）.....474
28.4.17 RTC 补偿周期寄存器（RTC_ADCNT）.....474
28.4.18 RTC 亚秒计数寄存器（RTC_SSR）.....475
28.4.19 RTC 秒时标占空比寄存器（RTC_DTR）.....475
28.4.20 RTC 控制寄存器（RTC_CR）.....476

29 LCD 显示.....477

29.1 概述.....477
29.2 结构框图.....478
29.3 IO 配置.....478
29.4 功能说明.....479
29.4.1 工作时钟和显示帧频率.....479
29.4.2 LCD Type A 扫描波形.....479
29.4.3 LCD Type B 扫描波形.....480
29.4.4 片内 buffer 驱动模式.....481
29.4.5 显示闪烁功能.....482
29.4.6 偏置电压调整.....482
29.5 寄存器.....484
29.5.1 显示控制寄存器（LCD_CR）.....484
29.5.2 显示测试控制寄存器（LCD_TEST）.....486
29.5.3 显示频率控制寄存器（LCD_FCR）.....487
29.5.4 闪烁时间寄存器（LCD_FLKT）.....487
29.5.5 显示中断使能寄存器（LCD_IER）.....488
29.5.6 显示中断标志寄存器（LCD_ISR）.....489
29.5.7 显示数据寄存器（LCD_DATAx）.....489
29.5.8 LCD COM 使能控制寄存器（LCD_COMEN）.....495
29.5.9 LCD SEG 使能控制寄存器 0（LCD_SEGEN0）.....496
29.5.10 LCD SEG 使能控制寄存器 1（LCD_SEGEN1）.....496
29.5.11 LCD Boost 控制寄存器（LCD_BSTCR）.....497

30 ADC 和温度传感器.....499

30.1 概述.....499
30.2 模块框图.....500
30.3 工作模式.....501
30.4 电压测量（TBD）.....501
30.4.1 测量电源电压.....501
30.4.2 测量外部通道输入.....501
30.5 温度传感器.....502
30.5.1 温度调校.....502
30.5.2 使用方式.....502
30.6 CIC 模式.....503
30.7 中断.....504
30.8 DMA.....504
30.9 ADC 外部输入通道.....505
30.10 寄存器.....506
30.10.1 ADC 控制寄存器(ADC_CR).....506
30.10.2 ADC 调校寄存器(ADC_TRIM).....507
30.10.3 ADC 输出数据寄存器(ADC_DR).....507
30.10.4 ADC 中断标志寄存器(ADC_ISR).....508

30.10.5 ADC 配置寄存器(ADC_CFGR) 509
30.10.6 CIC 输出数据寄存器(CIC_DR) 510
30.10.7 CIC 输出偏移寄存器(CIC_OS) 510
30.10.8 CIC 无符号输出结果寄存器(CIC_USDR) 511
30.10.9 CIC 滤波器控制寄存器(CIC_CR) 511
30.10.10 CIC 中断标志寄存器(CIC_ISR) 512

31 I/O 端口 514

31.1 概述 514
31.2 引脚类型 514
31.2.1 GPIO，输入输出使能，可控上拉电阻，可控开漏输出 515
31.2.2 GPIO，输入输出使能，2 个可控上拉电阻，可控开漏输出（仅 7816 数据口） 516
31.2.3 GPIO，输入输出使能，可控上拉电阻，可控开漏输出，5V tolerant 517
31.3 IO 端口功能定义 518
31.3.1 GPIO 输入 518
31.3.2 GPIO 输出 518
31.3.3 数字外设功能 518
31.3.4 数字外设功能优先级 520
31.3.5 模拟功能 521
31.3.6 使用外部晶体引脚 522
31.3.7 开漏输出 522
31.4 SWD 引脚 523
31.5 NRST 引脚 523
31.6 WKUPx 引脚 523
31.7 外部引脚中断（EXTI） 524
31.7.1 功能说明 524
31.7.2 应用指南 525
31.8 快速 GPIO 输出 526
31.9 IO MUX 输出 526
31.10 寄存器 527
31.10.1 GPIO 输入使能寄存器（GPIOx_INEN） 530
31.10.2 GPIO 上拉使能寄存器（GPIOx_PUEN） 531
31.10.3 GPIO 开漏使能寄存器（GPIOx_ODEN） 531
31.10.4 GPIO 功能选择寄存器（GPIOx_FCR） 532
31.10.5 GPIO 输出数据寄存器（GPIOx_DO） 534
31.10.6 GPIO 输出数据置位寄存器（GPIOx_DSET） 535
31.10.7 GPIO 输出数据复位寄存器（GPIOx_DRST） 535
31.10.8 GPIO 输入数据寄存器（GPIOx_DIN） 536
31.10.9 GPIO 额外数字功能选择寄存器（GPIOx_DFS） 537
31.10.10 GPIO 模拟开关使能寄存器（GPIOx_ANEN） 537
31.10.1 EXTI 输入选择寄存器 0（GPIO_EXTISEL0） 538
31.10.2 EXTI 输入选择寄存器 1（GPIO_EXTISEL1） 540
31.10.3 EXTI 边沿选择和使能寄存器 0（GPIO_EXTIEDS0） 541
31.10.4 EXTI 边沿选择和使能寄存器 1（GPIO_EXTIEDS1） 543
31.10.5 EXTI 数字滤波控制寄存器（GPIO_EXTIDF） 544
31.10.6 EXTI 中断标志寄存器（GPIO_EXTIISR） 545
31.10.7 EXTI 输入信号寄存器（GPIO_EXTIDI） 545
31.10.8 FOUT 配置寄存器（GPIO_FOUTSEL） 546
31.10.9 IO MUX 寄存器（GPIO_IOMCR） 547
31.10.10 WKUP 使能寄存器（GPIO_PINWKEN） 548
31.11 GPIOH 寄存器 549
31.11.1 GPIO 输入使能寄存器（GPIOH_INEN） 549
31.11.2 GPIO 上拉使能寄存器（GPIOH_PUEN） 549

31.11.3 GPIO 功能选择寄存器（GPIOH_FCR）.....550
31.11.4 GPIO 输出数据寄存器（GPIOH_DO）.....551
31.11.5 GPIO 输入数据寄存器（GPIOH_DIN）.....551
32 专用编程接口.....553
32.1 概述.....553
32.2 编程器使用.....553
33 调试支持.....554
33.1 概述.....554
33.2 DEBUG 引脚.....554
33.2.1 SWD 引脚.....554
33.2.2 上拉电阻.....554
33.3 SWD 接口协议.....555
33.3.1 协议简介.....555
33.3.2 传输序列.....555
33.3.3 SW-DP ID code.....556
33.3.4 主机读操作.....556
33.3.5 主机写操作.....557
33.4 SWD-DP 寄存器.....557
33.4.1 寄存器列表.....557
33.5 CORE DEBUG 寄存器.....558
33.6 DEBUG 相关的配置项.....558
34 器件签名信息.....559
34.1 存储器容量查询.....559
34.2 器件 UID.....559
34.3 芯片版本信息（ECO）.....560
上海复旦微电子集团股份有限公司销售及服务网点.....561

表目录

表 1-1 FM33A0xxEV 型号列表.....26
表 1-2 FM33A0xxEV 特性列表.....27
表 1-3 引脚列表.....38
表 2-1 FM33A0xxEV 极限参数.....46
表 2-2 FM33A0xxEV 典型工作条件.....47
表 2-3 ACTIVE 电流参数.....48
表 2-4 LP RUN 电流参数.....49
表 2-5 SLEEP 电流参数.....49
表 2-6 DEEPSLEEP 电流参数.....49
表 2-7 复位和电源监控参数.....52
表 2-8 唤醒时间参数.....53
表 2-9 低频晶体振荡器参数.....54
表 2-10 高频晶体振荡器参数.....54
表 2-11 内部高频 RC 振荡器参数.....55
表 2-12 内部中频 RC 振荡器参数.....57
表 2-13 内部 RCLP 振荡器参数.....58
表 2-14 PLL_H 参数.....59
表 2-15 PLL_L 参数.....59
表 2-16 ADC 参数.....60
表 2-17 温度传感器参数.....60
表 2-18 FLASH 参数.....61
表 2-19 普通 I/O 参数.....62
表 2-20 高驱动 I/O 参数.....63
表 2-21 NRST 引脚参数.....63
表 2-22 LCD 片内电阻分压.....64
表 2-23 LCD 片外电容驱动.....64
表 2-24 LCD BOOSTER 升压驱动.....65
表 4-1 FM33A0xxEV 处理器配置表.....80
表 4-2 CORTEX-M0 内核寄存器简表.....81
表 4-3 CORTEX-M0 中断向量表.....83
表 4-4 CORTEX-M0 HARDFAULT 类型.....84
表 5-1 总线地址分配表.....99
表 5-2 CPIF 接口内部地址分配.....99
表 5-3 用户配置信息区.....101
表 5-4 用户选项字节.....102
表 5-5 LOCK 配置字.....103
表 5-6 LOCK 位与 FLASH 地址对应表.....104
表 5-7 SYSCFG 配置字.....105
表 5-8 AES 密钥保存格式.....106
表 8-1 WWDT 时钟频率和溢出时间.....138
表 9-1 主要工作时钟及其源头.....145
表 9-2 外设模块工作时钟和总线时钟.....146
表 11-1 外部电压检测寄存器配置表.....177
表 11-2 内部电源检测阈值（基准选择 100）.....178
表 11-3 内部电源检测阈值（基准选择 010）.....178
表 11-4 内部电源检测阈值（基准选择 001）.....179
表 11-5 外部电源检测阈值（基准选择 100）.....179
表 11-6 外部电源检测阈值（基准选择 010）.....179
表 11-7 外部电源检测阈值（基准选择 001）.....180

表 16-1 比较器 1 正端输入列表.....235
表 16-2 比较器 1 负端输入列表.....235
表 16-3 比较器 2 负端输入列表.....236
表 16-4 比较器 2 负端输入列表.....236
表 16-5 比较器工作模式（常温 TA=25℃）.....236
表 17-1 I2C 引脚定义.....243
表 17-2 I^2C 接口时序要求.....247
表 18-1 UART 引脚定义.....270
表 18-2 UART 接收引脚优先级.....270
表 18-3 UART 类型区分.....271
表 18-4 UART 数据帧格式.....272
表 18-5 UART 使用 DMA 发送时的中断标志置位.....277
表 18-6 常用时钟频率下波特率计算.....279
表 19-1 LPUART 引脚定义.....293
表 19-2 LPUART 接收引脚优先级.....293
表 19-3 LPUART 数据帧格式.....294
表 19-4 LPUART 位调制系数.....295
表 19-5 LPUART 通过 DMA 发送时的中断标志产生.....297
表 20-1 SPI 引脚定义.....306
表 20-2 SPI 引脚定义.....306
表 21-1 ISO7816 引脚定义.....325
表 22-1 QUADSPI 引脚定义.....339
表 23-1 DMA1 通道映射.....368
表 24-1 CRC GOLDEN 参考数据.....383
表 27-1 LPTIM32 引脚映射.....442
表 29-1 帧频率计算公式.....479
表 29-2 典型帧频率和 DF 的关系.....479
表 30-1 ADC 相关中断和标志.....504
表 30-2 ADC 输入通道选择.....505
表 31-1GPIO 功能逻辑定义表.....515
表 31-2GPIO 功能逻辑定义表.....518
表 31-3 FCR 定义表.....518
表 31-4 多个数字外设功能选择表.....520
表 31-5 相同数字外设功能引脚优先级.....521
表 31-6 模拟信号通道.....522
详细请参考表 31-7.....538

图目录

图 1-1 FM33A0xxEV 结构框图.....28
图 1-2 FM33A0x10EV LQFP100 封装图.....28
图 1-3 FM33A0x8EV LQFP80 封装图.....29
图 1-4FM33A0x6EV LQFP64 封装图.....30
图 1-5 FM33A0x5EV LQFP48 封装图.....31
图 1-6 LQFP100 封装尺寸图.....39
图 1-7 LQFP80 封装尺寸图.....41
图 1-8LQFP64 封装尺寸图.....42
图 1-9LQFP48 封装尺寸图.....44
图 2-1 FM33A0xxEV 供电方案.....45
图 2-2 LCD 片外电容驱动模式电容连接.....65
图 3-1 芯片电源结构图.....67
图 3-2 芯片功耗模式转换.....69
图 3-3 功耗模式与系统主频.....70
图 5-1 芯片总线架构图.....95
图 5-2 总线地址分配（512KB FLASH）.....96
图 5-3 总线地址分配（256KB FLASH）.....97
图 5-4 启动区交换示意图.....105
图 6-1 芯片复位源框图.....124
图 6-2 上下电复位示意图.....125
图 7-1 IWDT 结构框图.....134
图 8-1 WWDT 结构框图.....137
图 8-2 WWDT 窗口示意图.....139
图 9-1 芯片时钟框图.....144
图 11-1 低压检测电路框图.....174
图 11-2 低压检测电路工作时序.....175
图 11-3 电源检测电路间歇工作模式.....176
图 11-4 使用外部电阻分压的 SVS 检测.....177
图 12-1 ECB 模式加密流程.....186
图 12-2 ECB 模式解密流程.....187
图 12-3 CBC 加密过程.....188
图 12-4 CBC 解密过程.....189
图 12-5 暂停模式流程.....189
图 12-6 CTR 加密流程.....190
图 12-7 CTR 解密流程.....191
图 12-8 32 位计数器和随机数的存储方式.....191
图 12-9 GCM 加密流程.....193
图 12-10 GCM 解密流程.....194
图 12-11 MULTH 模块框图.....195
图 12-12 根据数据类型存储数据的示意图.....197
图 12-13 模式 1：加密流程.....198
图 12-14 模式 2 示意图.....198
图 12-15 模式 3 示意图.....199
图 12-16 模式 4 示意图.....200
图 12-17 MULTH 模块使用流程示意图.....201
图 12-18 输入时 DMA 请求和数据传输示意图.....201
图 12-19 输出时 DMA 请求和数据传输示意图.....202
图 13-1 PAE 结构框图.....209
图 14-1 HASH 结构框图.....222

图 15-1 真随机数模块框图.....227
图 15-2 真随机数模块工作时钟.....228
图 16-1 比较器电路框图.....234
图 16-2 数字滤波（DFLEN=3）波形示意图.....236
图 17-1 I2C 模块框图.....242
图 17-2I²C 总线时序.....245
图 17-3 数据有效时序.....245
图 17-4 起始（START）与停止(STOP)命令定义.....245
图 17-5 输出应答(ACK).....246
图 17-6 主机向 7 位地址从机写入数据时的帧格式.....249
图 17-7 I2C 软件发送数据流图.....250
图 17-8 主机从 7 位地址从机读取数据时的帧格式.....251
图 17-9 I2C 软件发送数据流图.....252
图 17-10 双向数据通信帧格式.....252
图 17-11 I2C 软件发送数据流图.....253
图 17-12 I2C 软件发送数据流图.....254
图 17-13 I2C 主机 DMA 发送流程图.....256
图 17-14 I2C 主机 DMA 接收流程图.....258
图 17-15 主机时序控制.....260
图 18-1 UART 结构框图.....269
图 18-2 UART 字符描述.....271
图 18-3 位接收采样.....273
图 18-4 UART 异步发送波形 1.....274
图 18-5 UART 异步发送波形 2.....275
图 18-6 UART 异步发送波形 3.....275
图 18-7 红外调制波形.....280
图 18-8 UART 发送延迟.....281
图 19-1 LPUART 结构框图.....291
图 19-2 LPUART 工作时钟.....292
图 19-3 字符描述.....294
图 20-1 SPI 结构框图.....305
图 20-2 SPI 数据/时钟时序图（CPHA=0）.....307
图 20-3 SPI 数据/时钟时序图（CPHA=1）.....308
图 20-4 4 线半双工写操作.....308
图 20-5 4 线半双工读操作（无 DUMMY CYCLE）.....309
图 20-6 4 线半双工读操作（有 DUMMY CYCLE）.....309
图 20-7 SPI MASTER/SPI SLAVE 互连.....311
图 20-8 SPI SSN 时序图（SSNM=1，CPHA=0）.....313
图 20-9 SPI SSN 时序图（SSNM=0）.....313
图 21-1 ISO7816 结构框图.....324
图 21-2 ISO7816 数据帧结构.....325
图 21-3 ISO7816 数据接收过程.....326
图 21-4 ISO7816 数据发送过程.....328
图 22-1 QSPI 结构框图.....338
图 22-2 QPI 模式下的 PAGE PROGRAM.....343
图 22-3 QSPI XIP 示意图.....344
图 22-4 SINGLE-SPI CONFIGURATION.....345
图 22-5 SINGLE-SPI MODE 0 AND MODE 3.....345
图 22-6 DUAL-SPI CONFIGURATION.....345
图 22-7 QUAD-SPI CONFIGURATION.....346
图 22-8 单线发送指令.....347
图 22-9 双线发送指令.....347
图 22-10 四线发送指令.....348

图 22-11 延迟数据采样.....349
图 22-12 MODE 0 NCS 时序.....350
图 22-13 MODE 3 NCS 时序.....350
图 22-14 ABORT 时序.....350
图 22-15 XIP 跳转.....351
图 22-16 STANDARD SPI 模式下发送 WRITE ENABLE 指令.....352
图 22-17 STANDARD SPI 模式下发送 WRITE STATUS REGISTER2 指令.....352
图 22-18 STANDARD SPI 模式下发送 ENABLE QPI 指令.....352
图 22-19 QPI 模式下 FAST READ QUAD I/O 操作（前一条指令 M5-4=10）.....353
图 23-1 DMA1 结构框图.....365
图 23-2 DMA 寄存器配置.....366
图 23-3 DMA 工作流程.....367
图 23-4 DMA 循环模式下自动更新传输参数.....371
图 24-1CRC 运算流程图.....382
图 24-2 使用 DMA 对 RAM 数据进行 CRC.....384
图 25-1 通用定时器组.....389
图 25-2 BT1 结构图.....391
图 25-3 BT2 结构图.....392
图 25-4 BT1/2 输入控制逻辑.....392
图 25-5 BT1/2 输出控制逻辑.....393
图 25-6 BT 16BIT 级联计数.....394
图 25-7 BT 8BIT 独立计数.....395
图 25-8 BT 捕捉模式.....396
图 25-9 8BIT PWM.....397
图 25-10 输出脉冲.....398
图 25-11 输出翻转.....398
图 25-12 计数源预分频.....399
图 25-13 ET1 结构框图.....415
图 25-14 32BIT 级联结构框图.....415
图 25-15 IO 输入数字滤波示意图.....416
图 25-16 ETX PWM 输出示意图.....417
图 26-1 BSTIM32 结构框图.....427
图 26-2 预分频从 1 变为 2 的波形.....430
图 26-3 预分频从 1 变为 4 的波形.....430
图 26-4 向上计数波形，内部时钟不分频.....431
图 26-5 向上计数波形，内部时钟 2 分频.....432
图 26-6 ARPE=0（ARR 没有预装载）时的更新事件.....432
图 26-7 ARPE=1（ARR 预装载）时的更新事件.....433
图 26-8 内部时钟源模式，时钟分频因子为 1.....433
图 27-1 LPTIM32 结构框图.....441
图 27-2 外部 ETR 脉冲上升沿触发计数.....443
图 27-3 外部 ETR 脉冲异步计数（下降沿）.....443
图 27-4TIMEOUT 模式.....444
图 27-5PWM 输出.....445
图 27-6 输入信号边沿捕捉.....446
图 28-1 RTC 结构框图.....457
图 28-2 RTC 时间读取流程图.....462
图 29-1 LCD 显示控制模块结构框图.....478
图 29-2 LCD 驱动波形(1/4 DUTY, 1/3 BIAS, TYPE A).....480
图 29-3 LCD 驱动波形(1/4 DUTY, 1/3 BIAS, TYPE B).....481
图 29-4 LCD 片内电阻 BUFFER 型驱动电路.....482
图 29-5 偏置电压调整.....483
图 30-1 ADC 结构框图.....500

图 30-2 SDM 抽取滤波示意图 503
图 30-3 CIC 初始输出采样点丢弃 503
图 31-1 普通 GPIO 结构框图 515
图 31-2 普通 GPIO（两路上拉）结构框图 516
图 31-3 5V-TOLERANT GPIO 结构框图 517
图 31-4 与不同电源供电芯片的连接 523
图 31-5 引脚输入数字滤波 525
图 31-6 EXTI 信号输入示意图 525
图 33-1 CORTEX-M0 调试系统示意图 554

1 产品综述

1.1 概述

FM33A0xxEV是针对国网、南网单三相智能电表和海外智能电表的主控MCU，基于ARM Cortex-M0内核设计，集成最大512KB嵌入式闪存和最大80KB SRAM。

主要特性如下：

- 宽电压范围：2.0~5.5V
- 模拟电路工作电压范围：2.5~5.5V
- 工作温度范围：-40℃~+85℃
- 处理器内核
  - 32位 ARM Cortex-M0
  - 支持用户/特权模式
  - 支持MPU
  - 支持中断向量表重映射
  - 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135uA/MHz(测试条件f=32MHZ)
  - 32KHz下LPRUN功耗：典型值30uA
  - Sleep模式：典型值3.6uA
  - DeepSleep模式，RTC走时+全部RAM保持+CPU内核保持：典型值1.5uA
- 存储器
  - 256/512KB 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
  - 48/80KB 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）
  - 2x低功耗模拟比较器

- 高精度温度传感器，精度+/-1℃

- 通信接口
  - UART*6
  - LPUART*2
  - 7816主机*1
  - SPI*5，主从模式，QSPI*1
  - I2C*2，400KHz，主机模式
  - 11通道外设DMA，1通道MEMORY DMA
  - 可编程CRC校验模块
- 定时资源
  - 8bit通用定时器*4
  - 16bit通用定时器*4
  - 32bit基本定时器*1
  - 24-bit Systick*1
  - 32-bit低功耗定时器*1
  - 带窗口的CPU看门狗定时器*1
  - 系统看门狗定时器*1
  - 低功耗实时时钟日历（RTCC），带有数字调校功能，调校精度+/-0.03ppm
- LCD显示控制电路
  - 最大支持4COM×44SEG / 6COM×42SEG / 8COM×40SEG
  - 1/3 bias、1/4bias
  - 支持片内电阻和片外电容分压
  - 支持Booster升压
  - 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
  - 公钥密码算法加速引擎：支持ECC192/256/384/512、RSA2048
  - HASH硬件加速器：支持SHA-1、SHA-256
  - 真随机数发生器
- 时钟发生电路
  - 片上可配置高速RC振荡器，可配置频率输出8/16/24/32MHz，出厂调校误差+/-0.5%，8/16MHz全温区变化小于+/-2%
  - 低功耗32768Hz晶体振荡器，带有停振检测电路

- 8~24Mhz高频晶体振荡器
- PLL_H，最高输出64MHz
- PLL_L，输入32768Hz，输出16.384MHz

1.2 产品型号列表

型号	Flash 容量（KBytes）	RAM 容量（KBytes）	封装
FM33A0610EV	512	80	LQFP100
FM33A068EV	512	80	LQFP80
FM33A066EV	512	80	LQFP64
FM33A065EV	512	80	LQFP48
FM33A0410EV	256	48	LQFP100
FM33A048EV	256	48	LQFP80
FM33A046EV	256	48	LQFP64
FM33A045EV	256	48	LQFP48

表 1-1 FM33A0xxEV 型号列表

1.3 产品特性对照表

型号		FM33A0x10EV	FM33A0x8EV	FM33A0x6EV	FM33A0x5EV
CPU		Cortex-M0			
MPU		√			
Max Freq.		64MHz			
Flash		512/256KB			
RAM		80/48KB			
AES		√			
ECC/RSA		√			
SHA		√			
RNG		√			
Timers	16bitTimer	6			
	BSTIM32	1			
	LPTIM32	1			
	systick	1			
RTC/WWDT/IWDT		1/1/1			
SPI		5	5	3	2
QSPI		1	1	-	-
I2C		2	1	1	1
UART		6	6	6	5
LPUART		2	2	1	1
7816		1	1	1	1
GPIO		91	73	58	42
LCD		4*44/6*42/8*40	4*42/6*40/8*38	4*32/6*30	-
LCD booster		√	√	-	-
Comparator		2	2	2	1
ADC		12ch	8ch	7ch	4ch
TempSensor		√			

表 1-2 FM33A0xxEV 特性列表

1.4 芯片结构框图

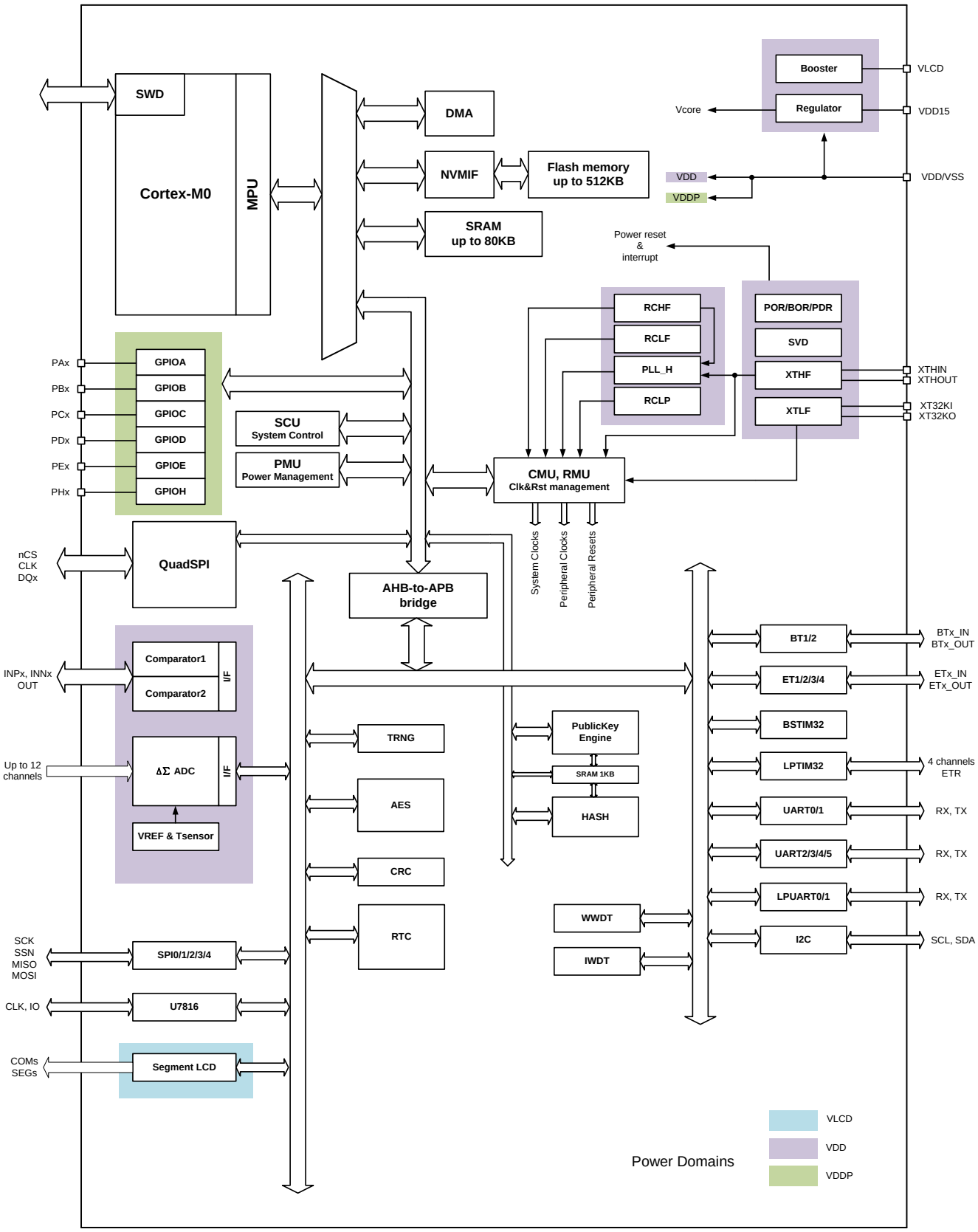


图 1-1 FM33A0xxEV 结构框图

1.5 引脚和封装定义

1.5.1 LQFP100 封装图

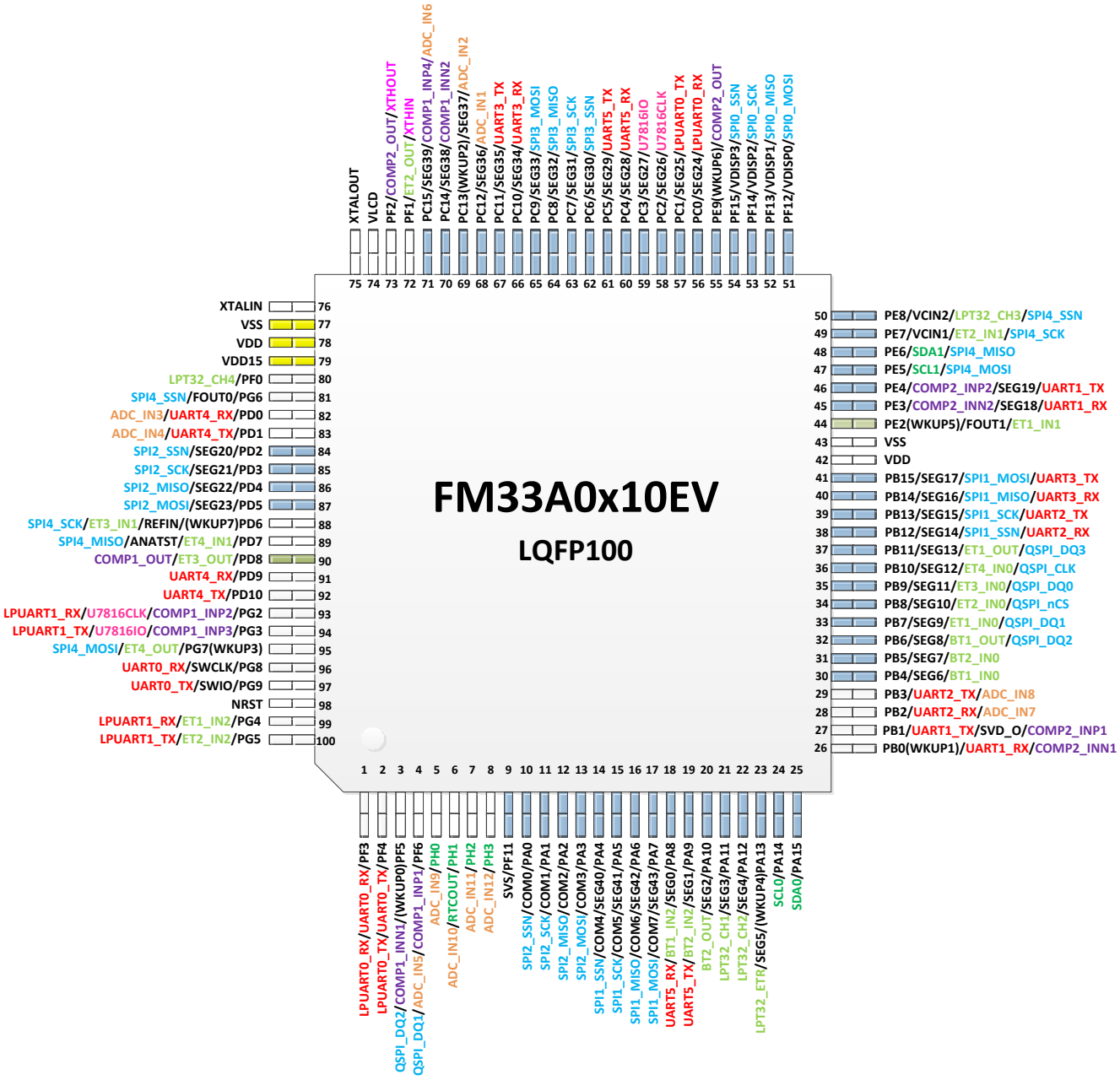


图 1-2 FM33A0x10EV LQFP100 封装图

【注】

1、蓝色管脚为5V tolerant (FT)，可以承受高于电源电压的输入电平而不会漏电

2、灰绿色管脚是强驱动引脚，其中PD8增强PMOS可以source 20mA，PE2增强NMOS可以sink 20mA

1.5.2 LQFP80 封装图

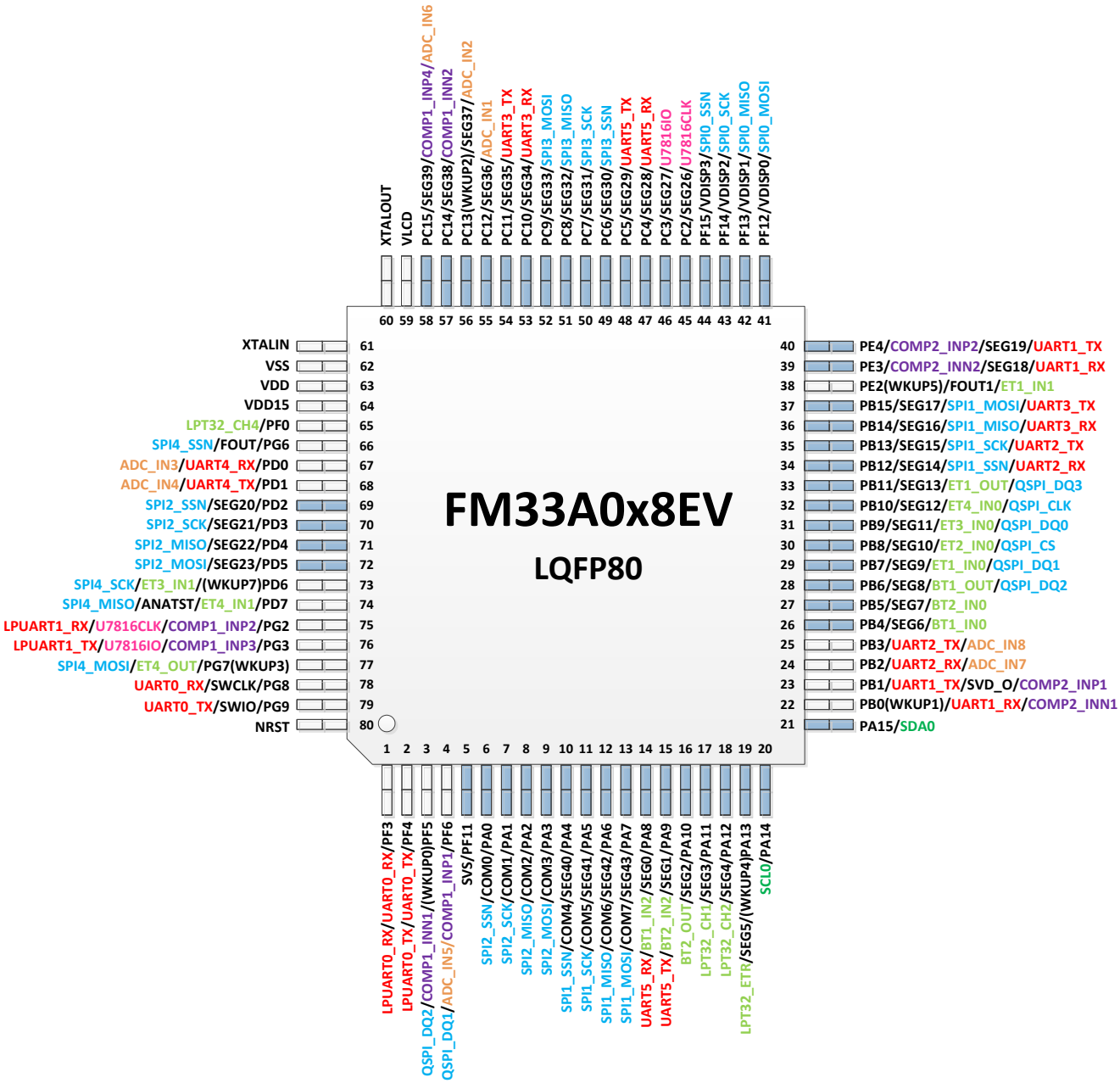


图 1-3 FM33A0x8EV LQFP80 封装图

【注】蓝色管脚为5V tolerant (FT)，可以承受高于电源电压的输入电平而不会漏电

1.5.3 LQFP64 封装图

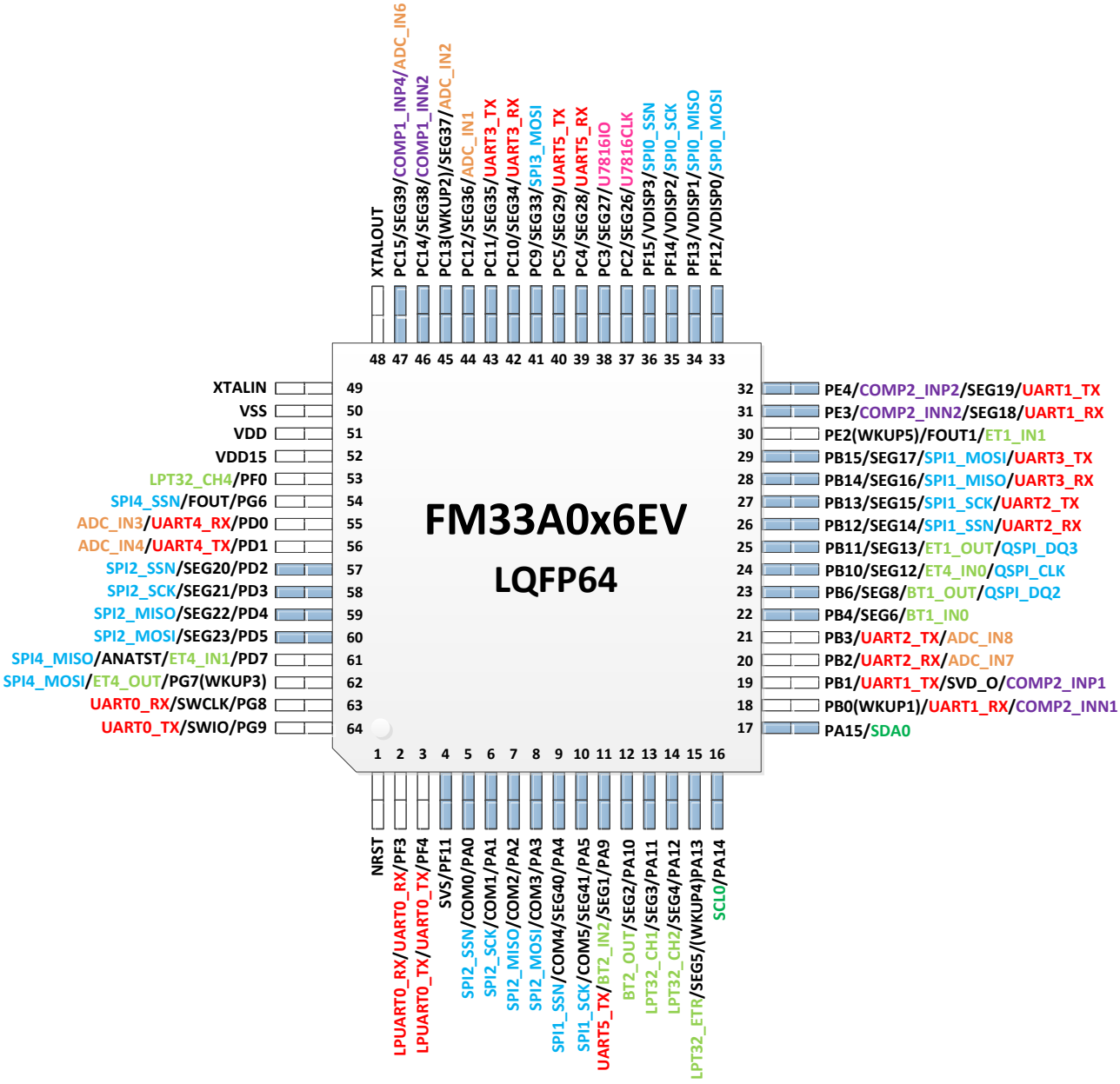


图 1-4FM33A0x6EV LQFP64 封装图

【注】蓝色管脚为5V tolerant (FT)，可以承受高于电源电压的输入电平而不会漏电

1.5.4 LQFP48 封装图

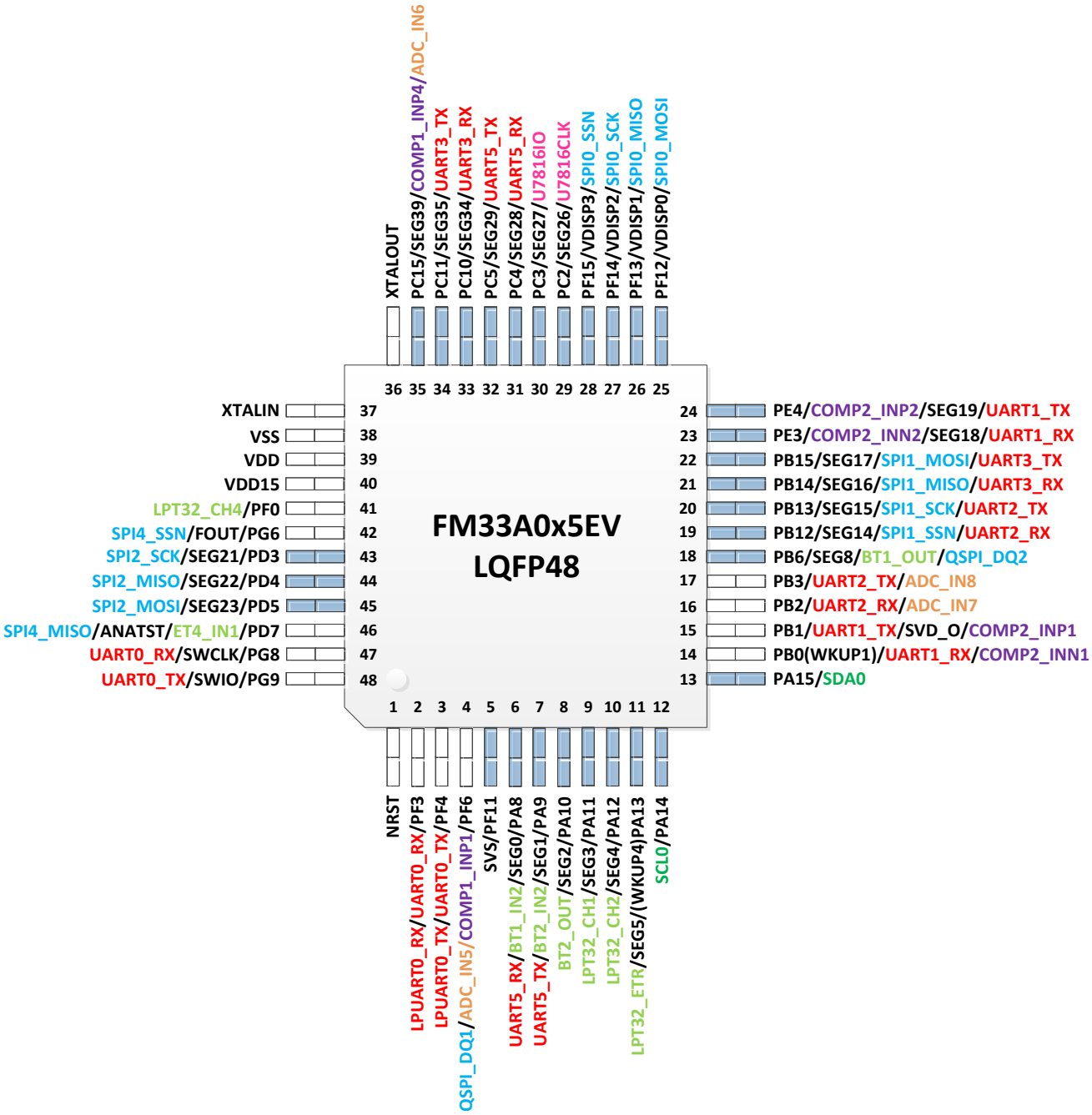


图 1-5 FM33A0x5EV LQFP48 封装图

【注】蓝色管脚为5V tolerant (FT)，可以承受高于电源电压的输入电平而不会漏电

1.5.5 引脚功能定义

Pin Number				Pin Function	Type	Descriptions
LQFP100	LQFP80	LQFP64	LQFP48			
1	1	2	2	PF3	FT	GPIO
				UART0_RX		UART 接收
				LPUART0_RX		LPUART 发送
2	2	3	3	PF4	FT	GPIO
				UART0_TX		UART 发送
				LPUART0_TX		LPUART 发送
3	3	-	-	PF5(WKUP0)		GPIO
				COMP1_INN1		比较器输入
				QSPI_DQ2		QSPI 数据
4	4	-	4	PF6		GPIO
				ADC_IN5		ADC 输入
				COMP1_INP1		比较器输入
				QSPI_DQ1		QSPI 数据
5	-	-	-	PH0		GPIO
				ADC_IN9		ADC 输入
6	-	-	-	PH1		GPIO
				ADC_IN10		ADC 输入
				RTCOUT		RTC 输出
7	-	-	-	PH2		GPIO
				ADC_IN11		ADC 输入
8	-	-	-	PH3		GPIO
				ADC_IN12		ADC 输入
9	5	4	5	PF11	FT	GPIO
				SVS		外部低压检测
10	6	5	-	PA0	FT	GPIO
				SPI2_SSN		SPI 片选
				COM0		LCD 驱动
11	7	6	-	PA1	FT	GPIO
				SPI2_SCK		SPI 时钟
				COM1		LCD 驱动
12	8	7	-	PA2	FT	GPIO
				SPI2_MISO		SPI 数据
				COM2		LCD 驱动
13	9	8	-	PA3	FT	GPIO
				SPI2_MOSI		SPI 数据
				COM3		LCD 驱动
14	10	9	-	PA4	FT	GPIO
				SPI1_SSN		SPI 片选
				COM4/SEG40		LCD 驱动
15	11	10	-	PA5	FT	GPIO
				SPI1_SCK		SPI 时钟
				COM5/SEG41		LCD 驱动
16	12			PA6	FT	GPIO

Pin Number				Pin Function	Type	Descriptions
LQFP100	LQFP80	LQFP64	LQFP48			
				SPI1_MISO		SPI 数据
				COM6/SEG42		LCD 驱动
17	13	-	-	PA7	FT	GPIO
				SPI1_MOSI		SPI 数据
				COM7/SEG43		LCD 驱动
18	14	-	6	PA8	FT	GPIO
				UART5_RX		UART 接收
				BT1_IN2		定时器输入
				SEG0		LCD 驱动
19	15	11	7	PA9	FT	GPIO
				UART5_TX		UART 发送
				BT2_IN2		定时器输入
				SEG1		LCD 驱动
20	16	12	8	PA10	FT	GPIO
				BT2_OUT		定时器输出
				SEG2		LCD 驱动
21	17	13	9	PA11	FT	GPIO
				LPT32_CH1		低功耗定时器通道
				SEG3		LCD 驱动
22	18	14	10	PA12	FT	GPIO
				LPT32_CH2		低功耗定时器通道
				SEG4		LCD 驱动
23	19	15	11	PA13(WKUP4)	FT	GPIO
				LPT32_ETR		低功耗定时器外部触发
				SEG5		LCD 驱动
24	20	16	12	PA14	FT	GPIO
				SCL0		I2C 时钟
				-		
25	21	17	13	PA15	FT	GPIO
				SDA0		I2C 数据
				-		
26	22	18	14	PB0(WKUP1)	FT	GPIO
				UART1_RX		UART 接收
				COMP2_INN1		比较器输入
27	23	19	15	PB1	FT	GPIO
				UART1_TX		UART 发送
				SVD_O		低压检测输出
				COMP2_INP1		比较器输入
28	24	20	16	PB2	FT	GPIO
				UART2_RX		UART 接收
				ADC_IN7		ADC 输入
29	25	21	17	PB3	FT	GPIO
				UART2_TX		UART 发送
				ADC_IN8		ADC 输入
30	26	22	-	PB4	FT	GPIO
				BT1_IN0		定时器输入

Pin Number				Pin Function	Type	Descriptions
LQFP100	LQFP80	LQFP64	LQFP48			
				SEG6		LCD 驱动
31	27	-	-	PB5	FT	GPIO
				BT2_IN0		定时器输入
				SEG7		LCD 驱动
32	28	23	18	PB6	FT	GPIO
				BT1_OUT		定时器输出
				QSPI_DQ2		QSPI 数据
				SEG8		LCD 驱动
33	29	-	-	PB7	FT	GPIO
				ET1_IN0		定时器输入
				QSPI_DQ1		QSPI 数据
				SEG9		LCD 驱动
34	30	-	-	PB8	FT	GPIO
				ET2_IN0		定时器输入
				QSPI_nCS		QSPI 片选
				SEG10		LCD 驱动
35	31	-	-	PB9	FT	GPIO
				ET3_IN0		定时器输入
				QSPI_DQ0		QSPI 数据
				SEG11		LCD 驱动
36	32	24	-	PB10	FT	GPIO
				ET4_IN0		定时器输入
				QSPI_CLK		QSPI 时钟
				SEG12		LCD 驱动
37	33	25	-	PB11	FT	GPIO
				ET1_OUT		定时器输出
				QSPI_DQ3		QSPI 数据
				SEG13		LCD 驱动
38	34	26	19	PB12	FT	GPIO
				SPI1_SSN		SPI 片选
				UART2_RX		UART 接收
				SEG14		LCD 驱动
39	35	27	20	PB13	FT	GPIO
				SPI1_SCK		SPI 时钟
				UART2_TX		UART 发送
				SEG15		LCD 驱动
40	36	28	21	PB14	FT	GPIO
				SPI1_MISO		SPI 数据
				UART3_RX		UART 接收
				SEG16		LCD 驱动
41	37	29	22	PB15	FT	GPIO
				SPI1_MOSI		SPI 数据
				UART3_TX		UART 发送
				SEG17		LCD 驱动
42	-			VDD		电源
43	-			VSS		地

Pin Number				Pin Function	Type	Descriptions
LQFP100	LQFP80	LQFP64	LQFP48			
44	38	30	-	PE2(WKUP5)		GPIO
				FOUT1		测试信号输出
				ET1_IN1		定时器输入
45	39	31	23	PE3	FT	GPIO
				UART1_RX		UART 接收
				COMP2_INN2		比较器输入
				SEG18		LCD 驱动
46	40	32	24	PE4	FT	GPIO
				UART1_TX		UART 发送
				COMP2_INP2		比较器输入
				SEG19		LCD 驱动
47	-	-	-	PE5	FT	GPIO
				SCL1		I2C 时钟
				SPI4_MOSI		SPI 数据
48	-	-	-	PE6	FT	GPIO
				SDA1		I2C 数据
				SPI4_MISO		SPI 数据
49	-	-	-	PE7	FT	GPIO
				ET2_IN1		定时器输入
				SPI4_SCK		SPI 时钟
				VCIN1		LCD 外部电容（电容模式）
50	-	-	-	PE8	FT	GPIO
				LPT32_CH3		低功耗定时器通道
				SPI4_SSN		SPI 片选
				VCIN2		LCD 外部电容（电容模式）
51	41	33	25	PF12	FT	GPIO
				SPI0_MOSI		SPI 数据
				VDISP0		LCD 外部电容（电容模式）
52	42	34	26	PF13	FT	GPIO
				SPI0_MISO		SPI 数据
				VDISP1		LCD 外部电容（电容模式）
53	43	35	27	PF14	FT	GPIO
				SPI0_SCK		SPI 时钟
				VDISP2		LCD 外部电容（电容模式）
54	44	36	28	PF15	FT	GPIO
				SPI0_SSN		SPI 片选
				VDISP3		LCD 外部电容（电容模式）
55	-	-	-	PE9(WKUP6)		GPIO
				COMP2_OUT		比较器输出
56	-	-	-	PC0	FT	GPIO

Pin Number				Pin Function	Type	Descriptions
LQFP100	LQFP80	LQFP64	LQFP48			
				LPUART0_RX		LPUART 接收
				SEG24		LCD 驱动
57	-	-	-	PC1	FT	GPIO
				LPUART0_TX		LPUART 发送
				SEG25		LCD 驱动
58	45	37	29	PC2	FT	GPIO
				U7816CLK		7816 时钟
				SEG26		LCD 驱动
59	46	38	30	PC3	FT	GPIO
				U7816IO		7816 数据
				SEG27		LCD 驱动
60	47	39	31	PC4	FT	GPIO
				UART5_RX		UART 接收
				SEG28		LCD 驱动
61	48	40	32	PC5	FT	GPIO
				UART5_TX		UART 发送
				SEG29		LCD 驱动
62	49	-	-	PC6	FT	GPIO
				SPI3_SSN		SPI 片选
				SEG30		LCD 驱动
63	50	-	-	PC7	FT	GPIO
				SPI3_SCK		SPI 时钟
				SEG31		LCD 驱动
64	51	-	-	PC8	FT	GPIO
				SPI3_MISO		SPI 数据
				SEG32		LCD 驱动
65	52	41	-	PC9	FT	GPIO
				SPI3_MOSI		SPI 数据
				SEG33		LCD 驱动
66	53	42	33	PC10	FT	GPIO
				UART3_RX		UART 接收
				SEG34		LCD 驱动
67	54	43	34	PC11	FT	GPIO
				UART3_TX		UART 发送
				SEG35		LCD 驱动
68	55	44	-	PC12	FT	GPIO
				ADC_IN1		ADC 输入
				SEG36		LCD 驱动
69	56	45	-	PC13(WKUP2)	FT	GPIO
				ADC_IN2		ADC 输入
				SEG37		LCD 驱动
70	57	46	-	PC14	FT	GPIO
				COMP1_INN2		比较器输入
				SEG38		LCD 驱动
71	58	47	35	PC15	FT	GPIO
				COMP1_INP4		比较器输入

Pin Number				Pin Function	Type	Descriptions
LQFP100	LQFP80	LQFP64	LQFP48			
				ADC_IN6		ADC 输入
				SEG39		LCD 驱动
72	-	-	-	PF1		GPIO
				ET2_OUT		定时器输出
				XTHIN		高频晶振输入
73	-	-	-	PF2		GPIO
				COMP2_OUT		比较器输出
				XTHOUT		高频晶振输出
74	59	-	-	VLCD		LCD 升压输出
75	60	48	36	XTALOUT		低频晶振输出
76	61	49	37	XTALIN		低频晶振输入
77	62	50	38	VSS		地
78	63	51	39	VDD		电源
79	64	52	40	VDD15		内部 LDO 输出，管脚串 10 欧电阻再接 1uF 对地电容
80	65	53	41	PF0		GPIO
				LPT32_CH4		低功耗定时器通道
81	66	54	42	PG6		GPIO
				FOUT0		测试信号输出
				SPI4_SSN		SPI 片选
82	67	55	-	PD0		GPIO
				UART4_RX		UART 接收
				ADC_IN3		ADC 输入
83	68	56	-	PD1		GPIO
				UART4_TX		UART 发送
				ADC_IN4		ADC 输入
84	69	57	-	PD2		GPIO
				SPI2_SSN	FT	SPI 片选
				SEG20		LCD 驱动
85	70	58	43	PD3		GPIO
				SPI2_SCK	FT	SPI 时钟
				SEG21		LCD 驱动
86	71	59	44	PD4		GPIO
				SPI2_MISO	FT	SPI 数据
				SEG22		LCD 驱动
87	72	60	45	PD5		GPIO
				SPI2_MOSI	FT	SPI 数据
				SEG23		LCD 驱动
88	73	-	-	PD6(WKUP7)		GPIO
				SPI4_SCK		SPI 时钟
				ET3_IN1		定时器输入
				REFIN		参考信号输入
89	74	61	46	PD7		GPIO
				SPI4_MISO		SPI 数据
				ANATST		模拟测试输出

Pin Number				Pin Function	Type	Descriptions
LQFP100	LQFP80	LQFP64	LQFP48			
				ET4_IN1		定时器输入
90	-	-	-	PD8		GPIO
				ET3_OUT		定时器输出
				COMP1_OUT		比较器输出
91	-	-	-	PD9	FT	GPIO
				UART4_RX		UART 接收
92	-	-	-	PD10	FT	GPIO
				UART4_TX		UART 发送
93	75	-	-	PG2	FT	GPIO
				LPUART1_RX		LPUART 接收
				U7816CLK		7816 时钟
				COMP1_INP2		比较器输入
94	76	-	-	PG3	FT	GPIO
				LPUART1_TX		LPUART 发送
				U7816IO		7816 数据
				COMP1_INP3		比较器输入
95	77	62	-	PG7(WKUP3)		GPIO
				SPI4_MOSI		SPI 数据
				ET4_OUT		定时器输出
96	78	63	47	PG8		GPIO
				SWCLK		SWD 时钟
				UART0_RX		UART 接收
97	79	64	48	PG9		GPIO
				SWIO		SWD 数据
				UART0_TX		UART 发送
98	80	1	1	NRST		复位输入
99	-	-	-	PG4	FT	GPIO
				LPUART1_RX		LPUART 接收
				ET1_IN2		定时器输入
100	-	-	-	PG5	FT	GPIO
				LPUART1_TX		LPUART 发送
				ET2_IN2		定时器输入

表 1-3 引脚列表

1.5.6 封装尺寸图

1.5.6.1 LQFP100

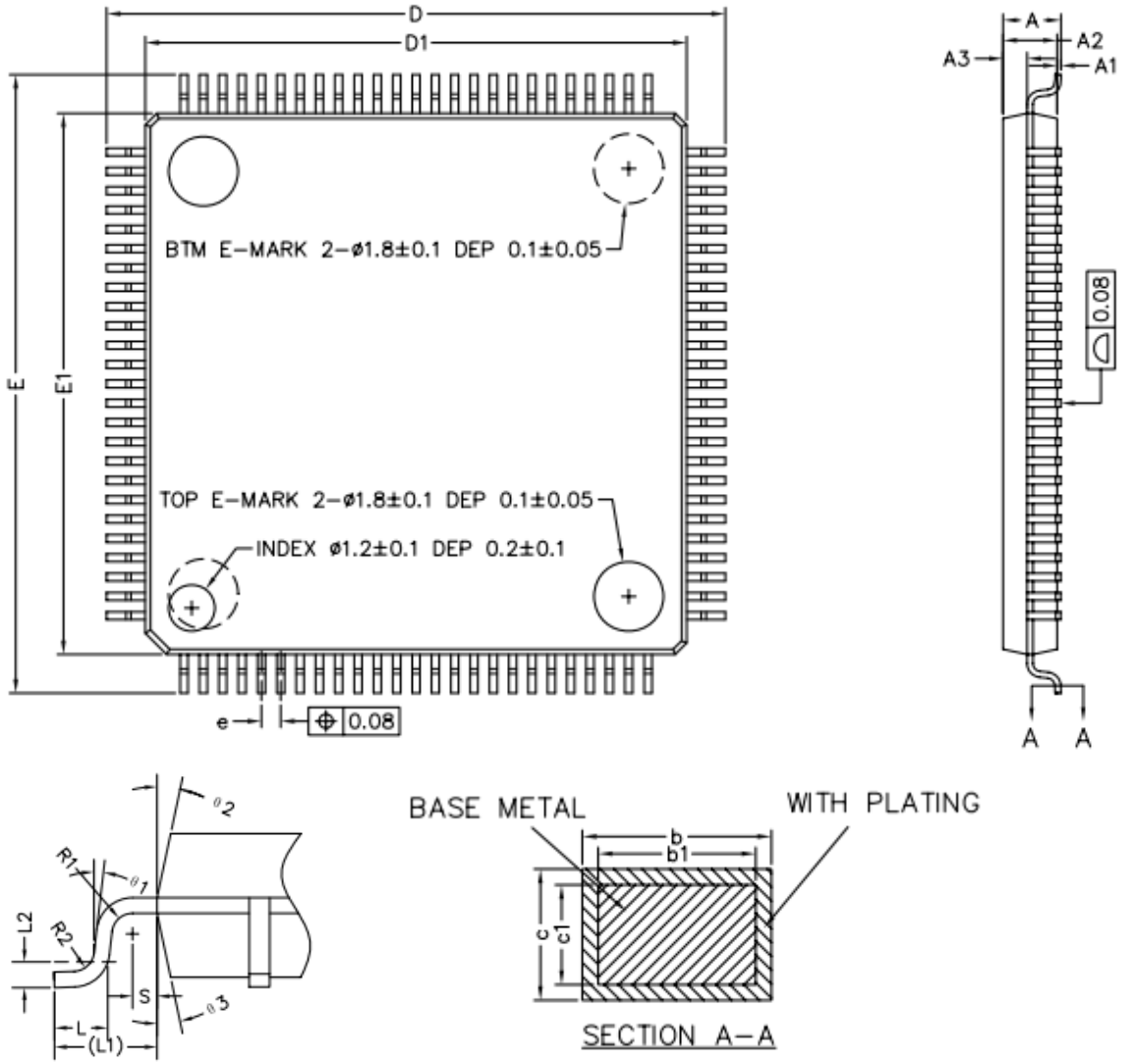


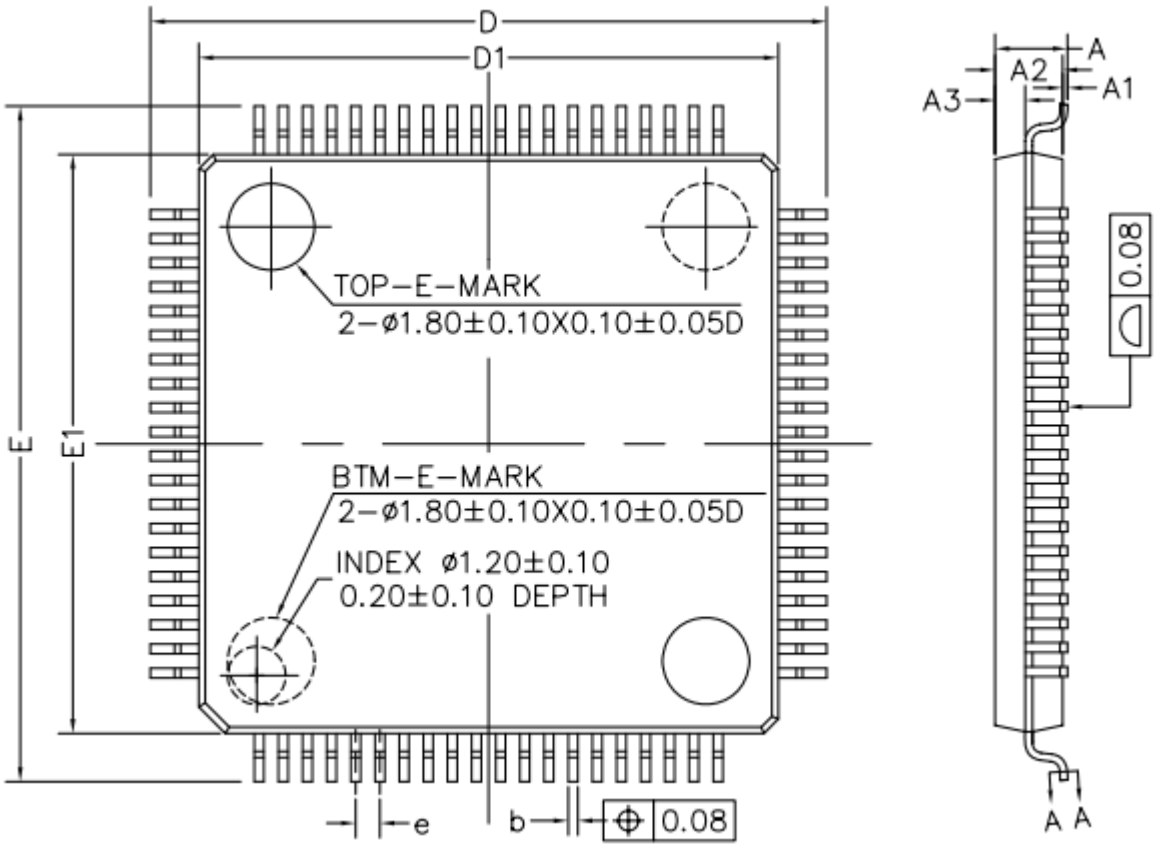
图 1-6 LQFP100 封装尺寸图

Symbol	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	15.80	16.00	16.20
D1	13.90	14.00	14.10
E	15.80	16.00	16.20

Symbol	MIN	NOM	MAX
E1	13.90	14.00	14.10
e	0.40	0.50	0.60
L	0.45	0.60	0.75
L1		1.00REF	
L2		0.25BSC	
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

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

1.5.6.2 LQFP80



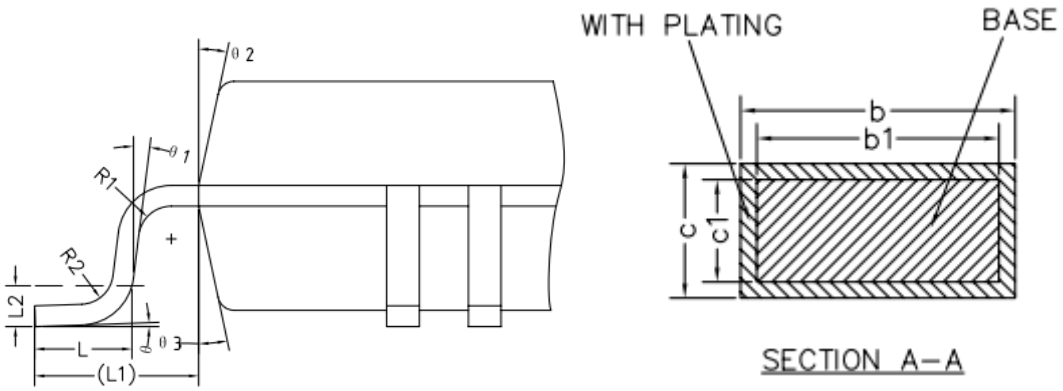


图 1-7 LQFP80 封装尺寸图

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	13.80	14.00	14.20
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 MS-026 BDD.

1.5.6.3 LQFP64

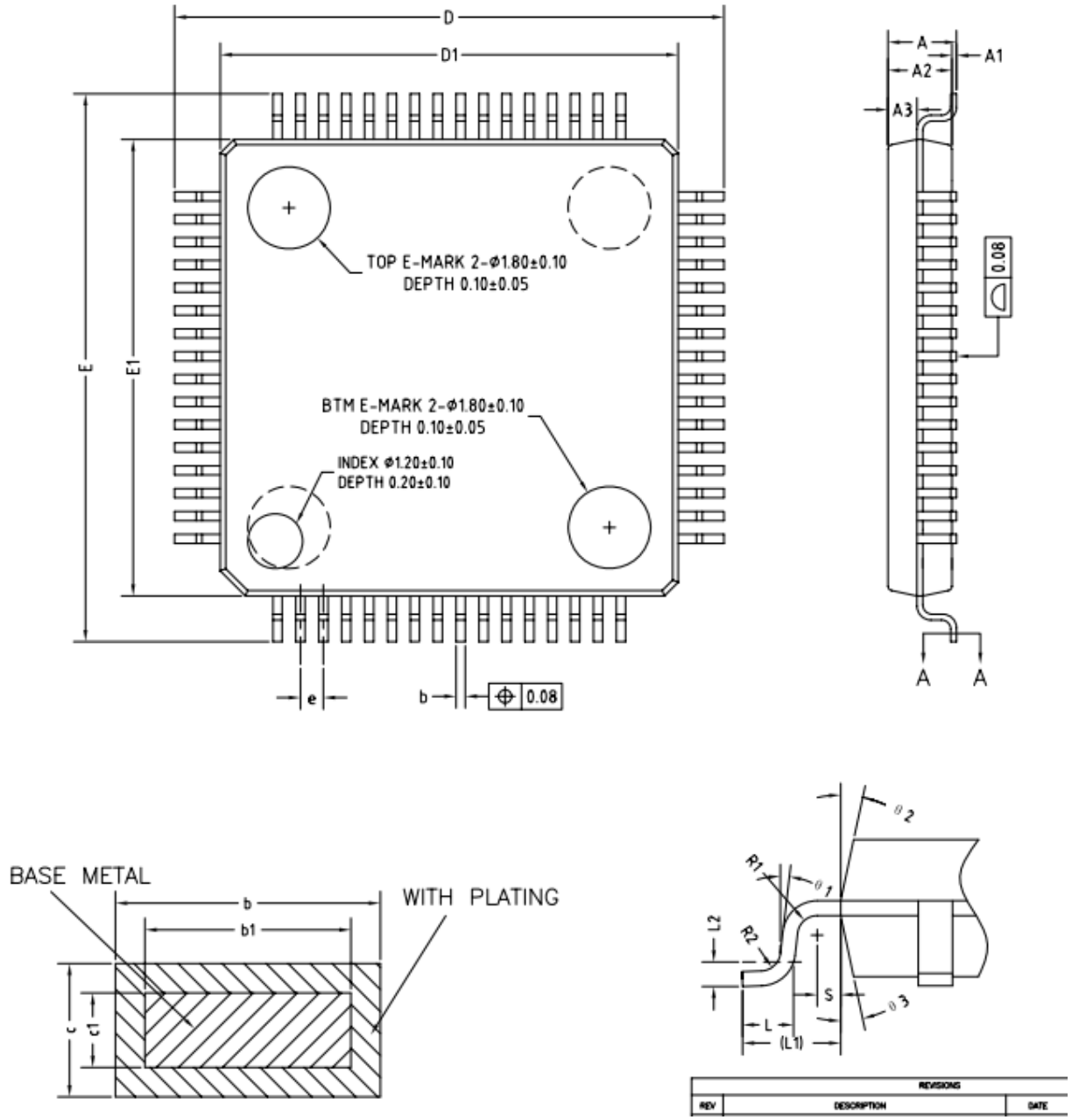


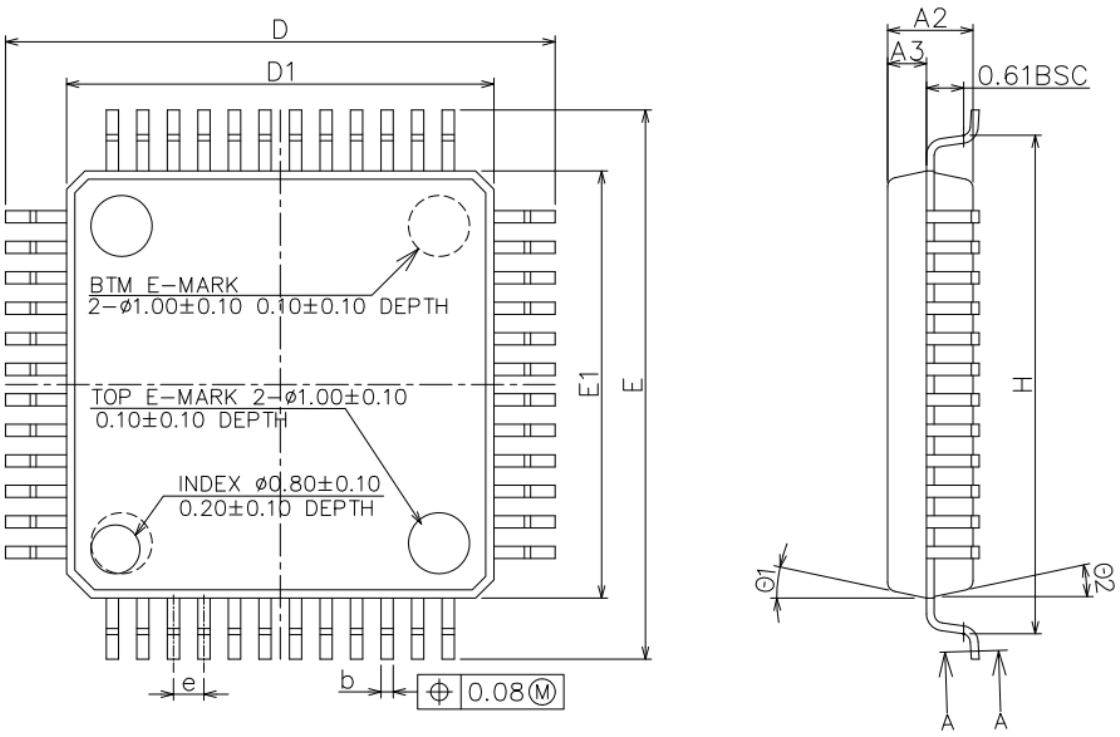
图 1-8LQFP64 封装尺寸图

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
D1	9.90	10.00	10.10

Symbol	MIN	NOM	MAX
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 MS-026 BDD.

1.5.6.4 LQFP48



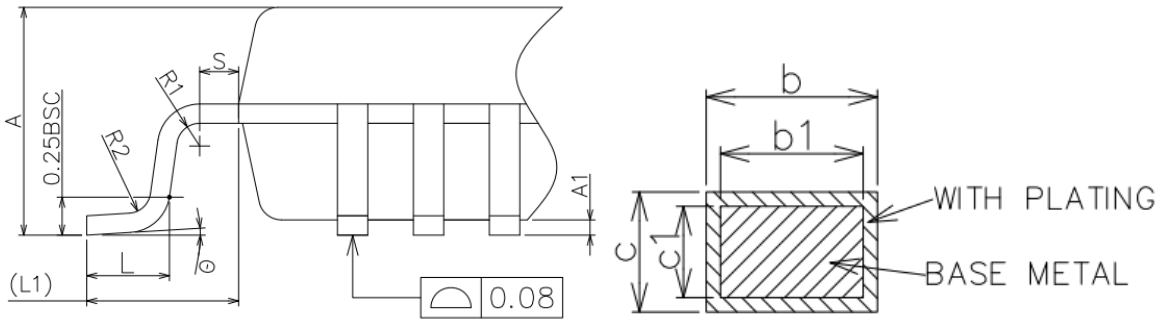


图 1-9LQFP48 封装尺寸图

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-026BDD.

2 电参数

2.1 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，由一定数量样本数据分布的平均值加减 3sigma 得到。

2.2 参数测试条件

2.2.1 供电方案

芯片测试时采用下图所示的电源供电方案。

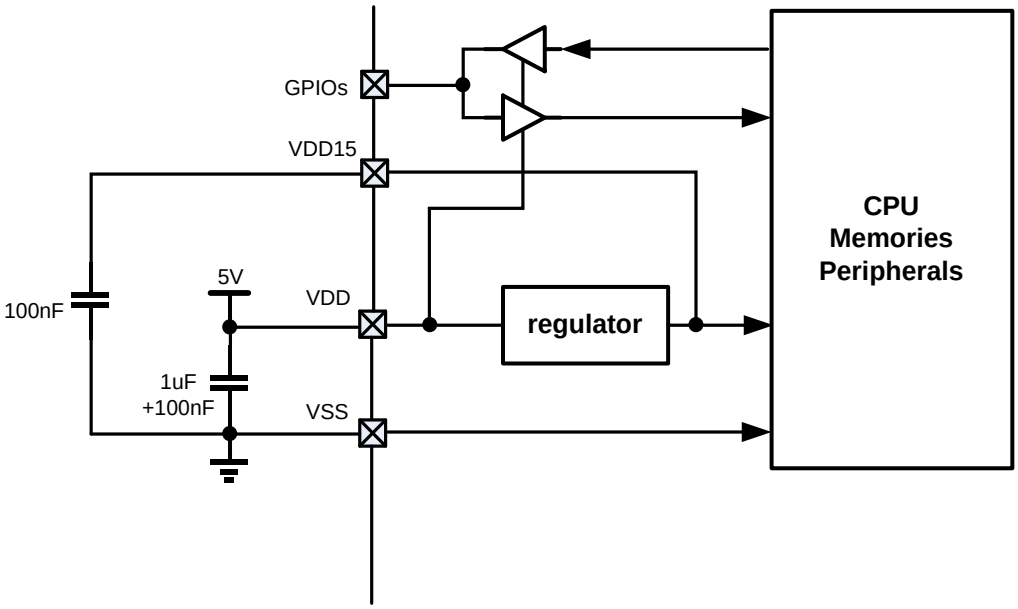


图 2-1 FM33A0xxEV 供电方案

2.3 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	-0.3	6	V
V_{PIN}	管脚电压	V_{SS}-0.3	6	V
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		+/-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)		90	mA
$\sum I_{VSS}$	从芯片 VSS 流出的最大电流 (sink)		70	mA
$\sum I_{IO}$	所有 IO sink 的最大总和电流		70	mA
	所有 IO source 的最大总和电流		90	mA

表 2-1 FM33A0xxEV 极限参数

2.4 性能参数

2.4.1 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	1.8	5.5	V
		BOR 关闭	1.8	5.5	
VRTC	RTC 典型工作电压范围	-	1.5	5.5	V
T_J	结温	T_A=-40~+85C	-40	105	° C

表 2-2 FM33A0xxEV 典型工作条件

2.4.2 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=VDDA=3.3V 的条件下测试获得

Active 模式功耗

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
IDD_{RUN}	运行模式下的功耗，CPU 从 Flash 取指，Coremark	f_{AHB}=8MHz (RCHF) PLL off Flash 0 wait	TA=25℃	-	1.59	-	mA
			TA=85℃	-	1.6	-	
		f_{AHB}=16MHz (RCHF) PLL off Flash 0 wait	TA=25℃	-	3.1	-	mA
			TA=85℃	-	3.2	-	
		f_{AHB}=24MHz (RCHF) PLL off Flash 0 wait	TA=25℃	-	4.5	-	mA
			TA=85℃		4.5		
		f_{AHB}=48MHz PLL on Flash 1 wait	TA=25℃	-	6.5	-	mA
			TA=85℃				
IDD_{RUN}	运行模式下的功耗，CPU 从 Flash 取指，while(1)	f_{AHB}=8MHz (RCHF) PLL off Flash 0 wait	TA=25℃	-	1.52	-	mA
			TA=85℃	-	1.55	-	
		f_{AHB}=16MHz (RCHF) PLL off Flash 0 wait	TA=25℃	-	2.9	-	mA
			TA=85℃				
		f_{AHB}=24MHz (RCHF) PLL off Flash 0 wait	TA=25℃	-	4.2	-	mA
			TA=85℃				
		f_{AHB}=48MHz PLL on Flash 1 wait	TA=25℃	-	6.1	-	mA
			TA=85℃				

表 2-3 ACTIVE 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

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	VDD=5V f_{AHB}=32768Hz (XTLF) PLL, RCHF, RC4M off Flash 0 wait	TA=25℃		28		uA
			TA=85℃		33		

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
	取指，While(1)						

表 2-4 LP RUN 电流参数

SLEEP 模式功耗

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
I_{sleep}	Sleep 模式电流	BOR、SVD 关闭 RTC 使用 XTLF 走时 CPU、RAM、外设数据保持 LCD 显示使能，空载	TA=25℃		4		uA
			TA=85℃		10		

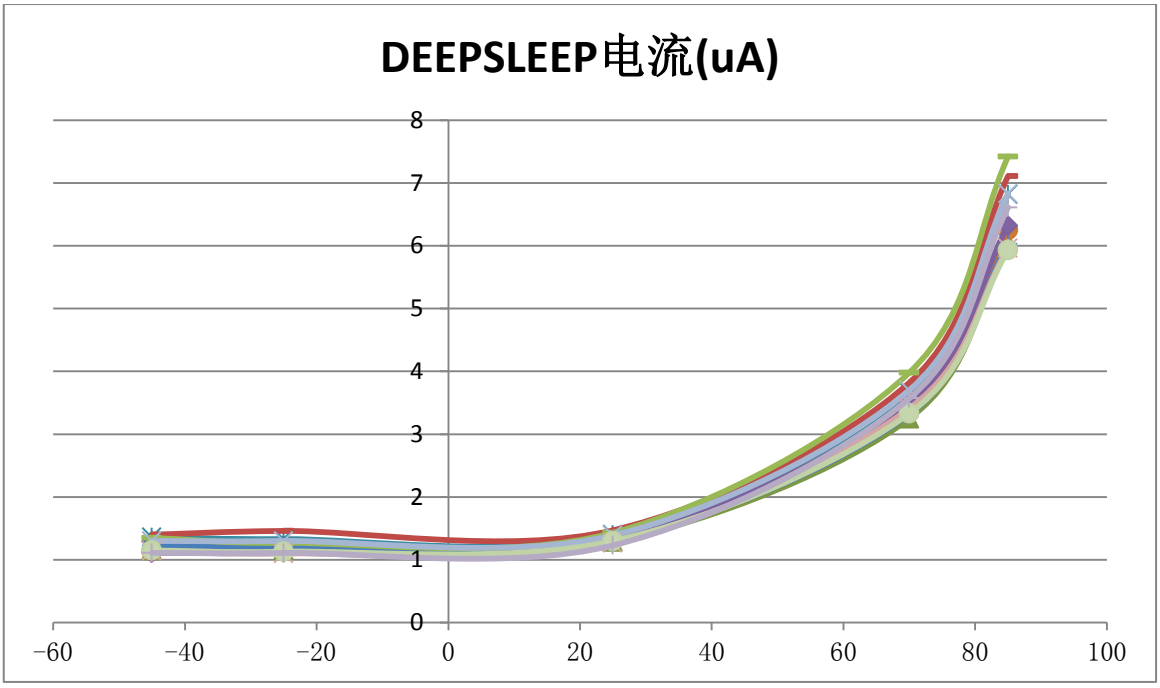
表 2-5 SLEEP 电流参数

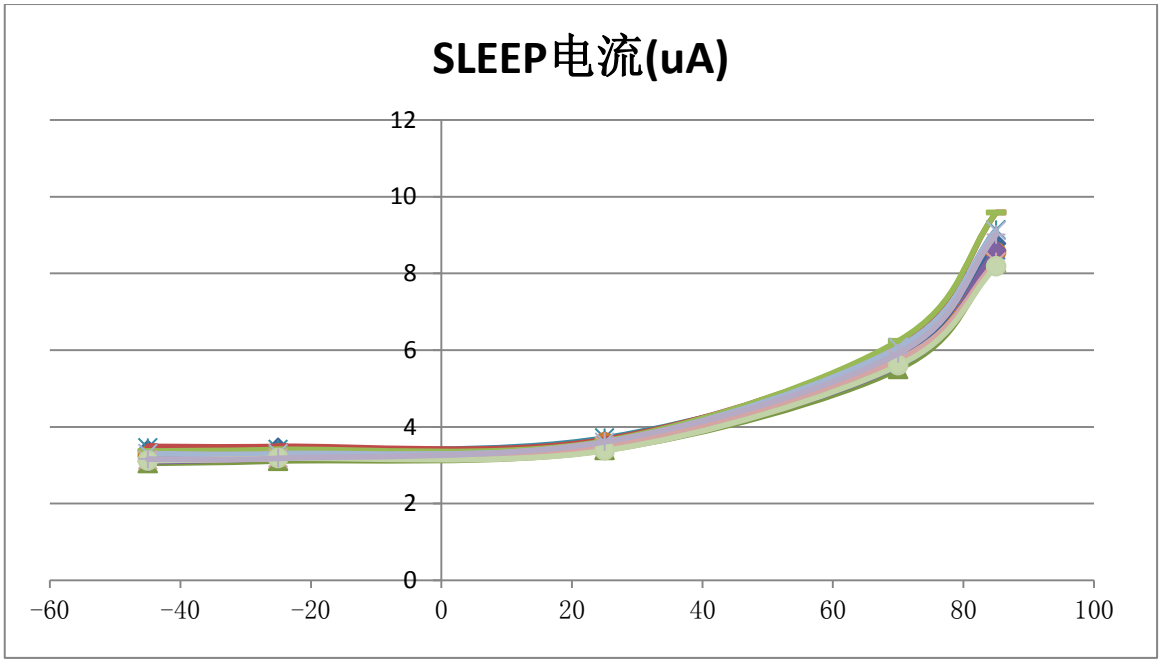
DEEPSLEEP 模式功耗

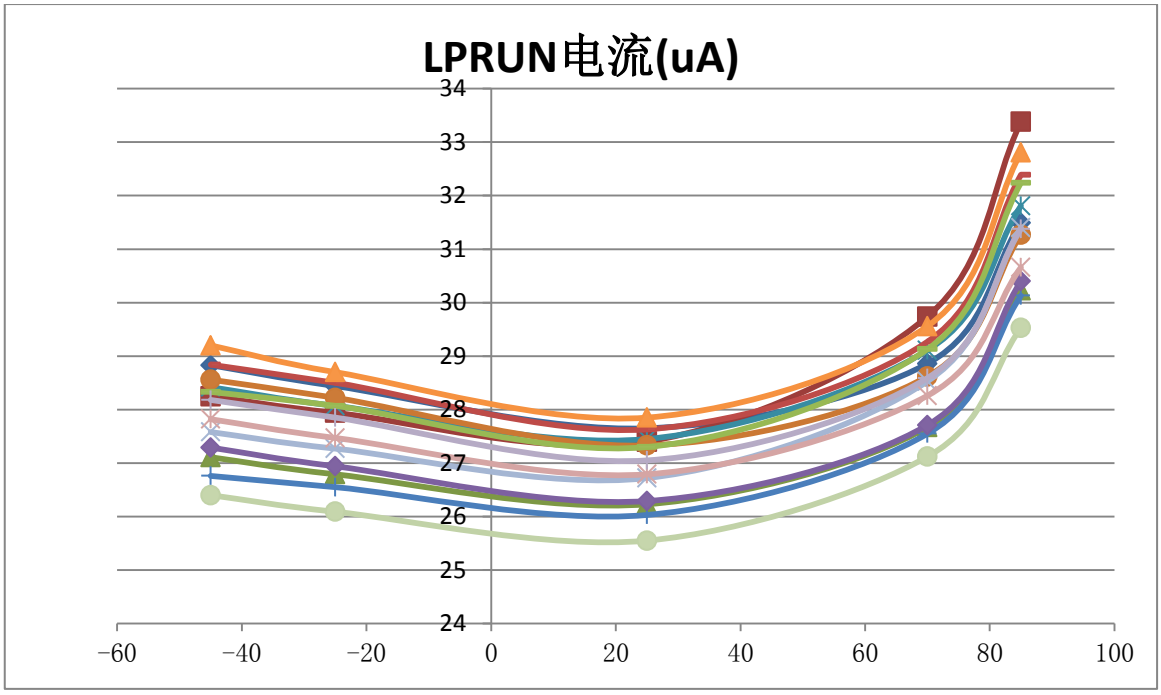
符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
$I_{deepsleep}$	DeepSleep 模式电流	BOR、SVD 关闭 RTC 使用 XTLF 走时 XTLF 驱动电流 250nA CVS=1 CPU、RAM、外设数据保持 LCD 显示关闭	TA=25℃		1.5	3	uA
			TA=85℃		7		

表 2-6 DEEPSLEEP 电流参数

以下为电流参数与温度的测试曲线，仅供设计参考。


DEEPSLEEP电流(uA)
8
7
6
5
4
3
2
1
0
-60
-40
-20
0
20
40
60
80
100


SLEEP电流(uA)
12
10
8
6
4
2
0
-60
-40
-20
0
20
40
60
80
100


LPRUN电流(uA)
34
33
32
31
30
29
28
27
26
25
24
-60
-40
-20
0
20
40
60
80
100

2.4.3 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		200		∞	us/V
		BOR		600		∞	us/V
T_{reset_delay}	上电复位延迟时间				2		ms
T_{pdr_filter}	下电复位滤波时间				32		us
V_{POR}	上电复位电压			1.71	1.8	2.0[1] (-40℃)	V
V_{BOR}	下电复位电压	BORCFG==2'b01		1.52	1.6	1.69	V
V_{PDR}	低功耗下电复位电压	PDRCFG==2'b11		1.34[1] (85℃)	1.52	1.72[1] (-40℃)	V
I_{BOR}	BOR 功耗				1.7		uA
I_{PDR}	PDR 功耗				55		nA
V_{SVD}	电压监测阈值电平	SVD[3:0]=0000	Fall		1.800		V
			Rise		1.900		
		SVD[3:0]=0001	Fall		2.014		V
			Rise		2.114		
		SVD[3:0]=0010	Fall		2.229		V
			Rise		2.329		
		SVD[3:0]=0011	Fall		2.443		V
			Rise		2.543		
		SVD[3:0]=0100	Fall		2.657		V
			Rise		2.757		
		SVD[3:0]=0101	Fall		2.871		V
			Rise		2.971		
		SVD[3:0]=0110	Fall		3.086		V
			Rise		3.186		
		SVD[3:0]=0111	Fall		3.300		V
			Rise		3.400		
		SVD[3:0]=1000	Fall		3.514		V
			Rise		3.614		
		SVD[3:0]=1001	Fall		3.729		V
			Rise		3.829		
		SVD[3:0]=1010	Fall		3.943		V
			Rise		4.043		
		SVD[3:0]=1011	Fall		4.157		V
			Rise		4.257		
		SVD[3:0]=1100	Fall		4.371		V
			Rise		4.471		
		SVD[3:0]=1101	Fall		4.586		V
			Rise		4.686		
		SVD[3:0]=1110	Fall		4.800		V
			Rise		4.900		
		SVD[3:0]=1111	Fall		-		V
			Rise		-		

表 2-7 复位和电源监控参数

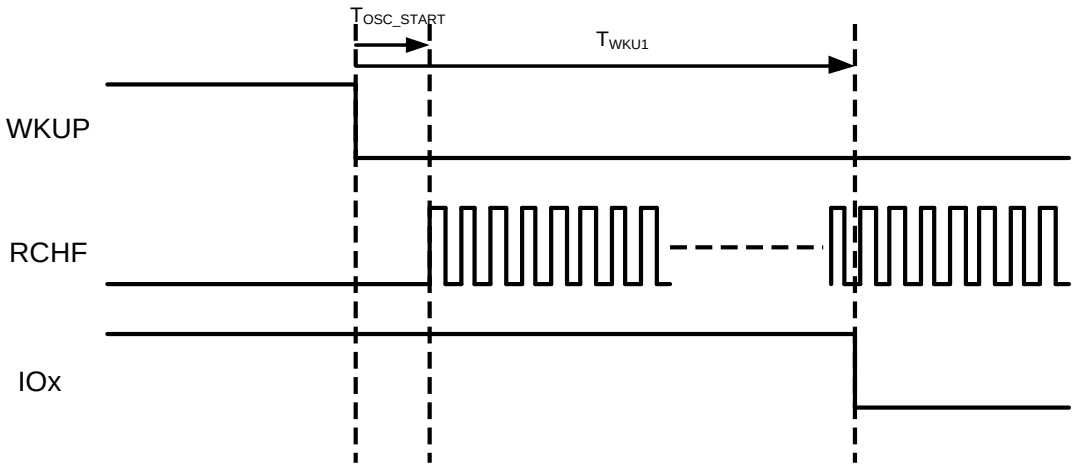
2.4.4 低功耗模式唤醒时间

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 2-8 唤醒时间参数

[1] 基于特征参数提取

典型唤醒事件波形图，仅供设计参考



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

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

2.4.5 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==4'b1000		1	3	s
	时钟频率偏差	外接 32768Hz 晶体，常温，调校后			±5	ppm
	时钟电路功耗	常温，3.3V			1.3	uA

表 2-9 低频晶体振荡器参数

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
F_{XTHF}	XTHF 振荡频率	-	8	-	24	MHz
R_{fb}	反馈电阻	-	-	200	-	KΩ
VDD_{rise}	XTHF 最低工作电压（上电）	8Mhz, CFG=000 C_{load}=10pF	1.9	-	-	V
		8Mhz, CFG=111 C_{load}=10pF	1.9	-	-	
		24Mhz, CFG=000 C_{load}=10pF	2.75	-	-	
		24Mhz, CFG=111 C_{load}=10pF	2.45	-	-	
VDD_{fall}	XTHF 最低工作电压（下电）	8Mhz, CFG=000 C_{load}=10pF	1.5	-	-	V
		8Mhz, CFG=111 C_{load}=10pF	1.35	-	-	
		24Mhz, CFG=000 C_{load}=10pF	2.7	-	-	
		24Mhz, CFG=111 C_{load}=10pF	2.2	-	-	
T_{start}	XTHF 起振时间	VDD=3.3V, CFG=000 外接 8Mhz 晶体	-	2.5	-	ms
C_L	负载电容	-	4	-	20	pF
I_{XTHF}	XTHF 振荡电流	VDD=3.3V, CFG=000 外接 8Mhz 晶体		180		uA
		VDD=5V, CFG=000 外接 8Mhz 晶体		400		
		VDD=5V, CFG=000 外接 24Mhz 晶体		1200		

表 2-10 高频晶体振荡器参数

2.4.6 内部时钟源特性

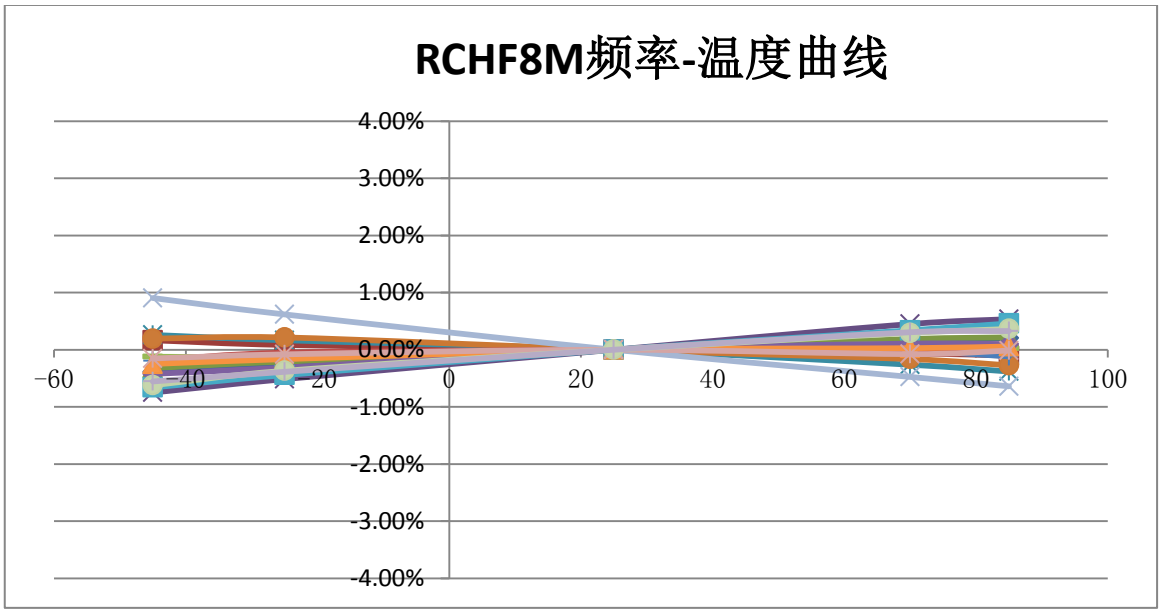
内部高频 RC 振荡器（RCHF）

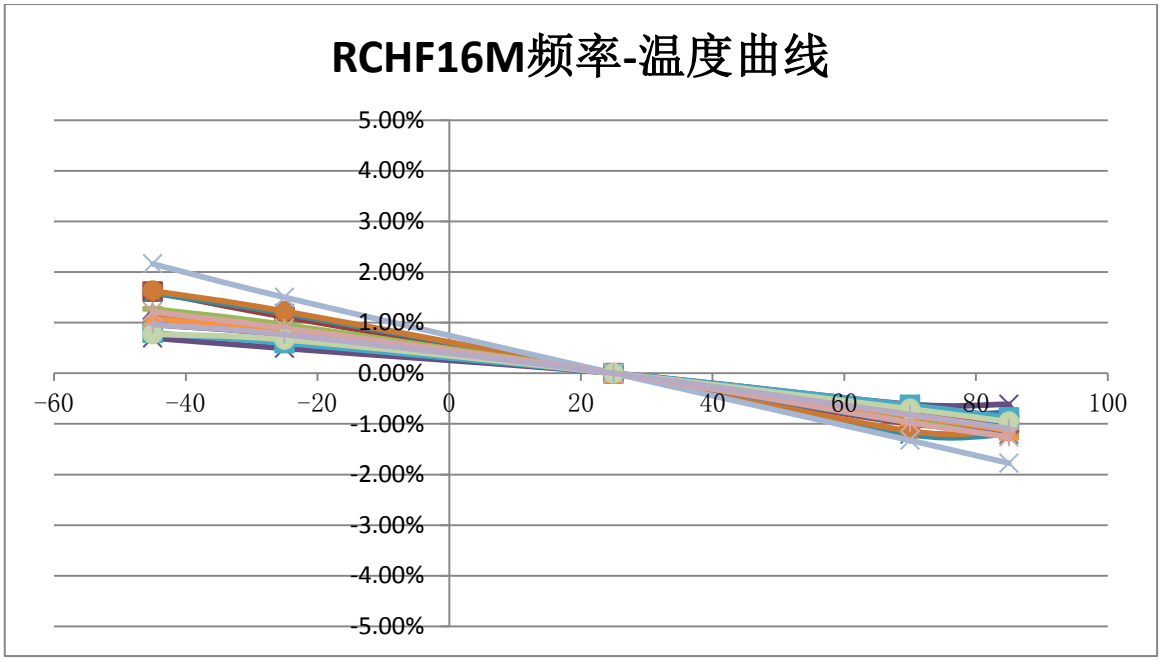
符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
f_{RCHF}[1]	RCHF 振荡频率	VDD=1.8~5.5V	FSEL==0000	7.92	8	8.08	MHz
			FSEL==0001	15.84	16	16.16	
			FSEL==0010	23.76	24	24.24	
			FSEL==0011	31.68	32	32.32	
ACC_{RCHF}[2]	全温区 RCHF 变化范围	VDD=1.8~5.5V	FSEL==0000 T=-40~+85℃	-1	-	1	%
			FSEL==0001 T=-40~+85℃	-2	-	2	%
			FSEL==0010 T=-40~+85℃	-3	-	3	%
			FSEL==0011 T=-40~+85℃	-3.5	-	3.5	%

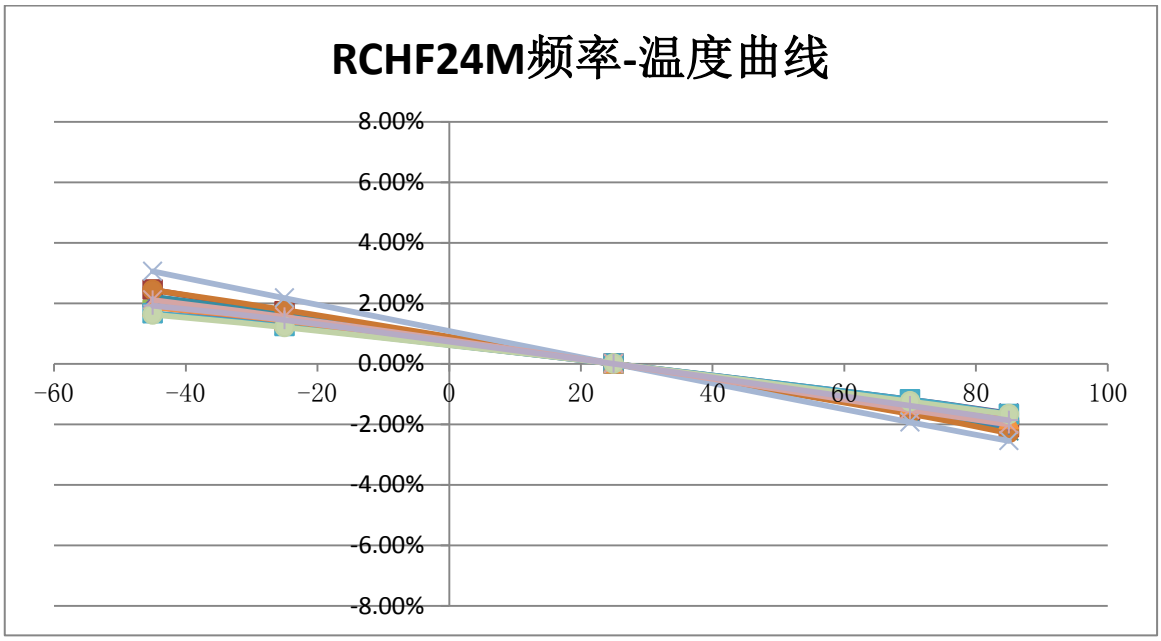
表 2-11 内部高频 RC 振荡器参数

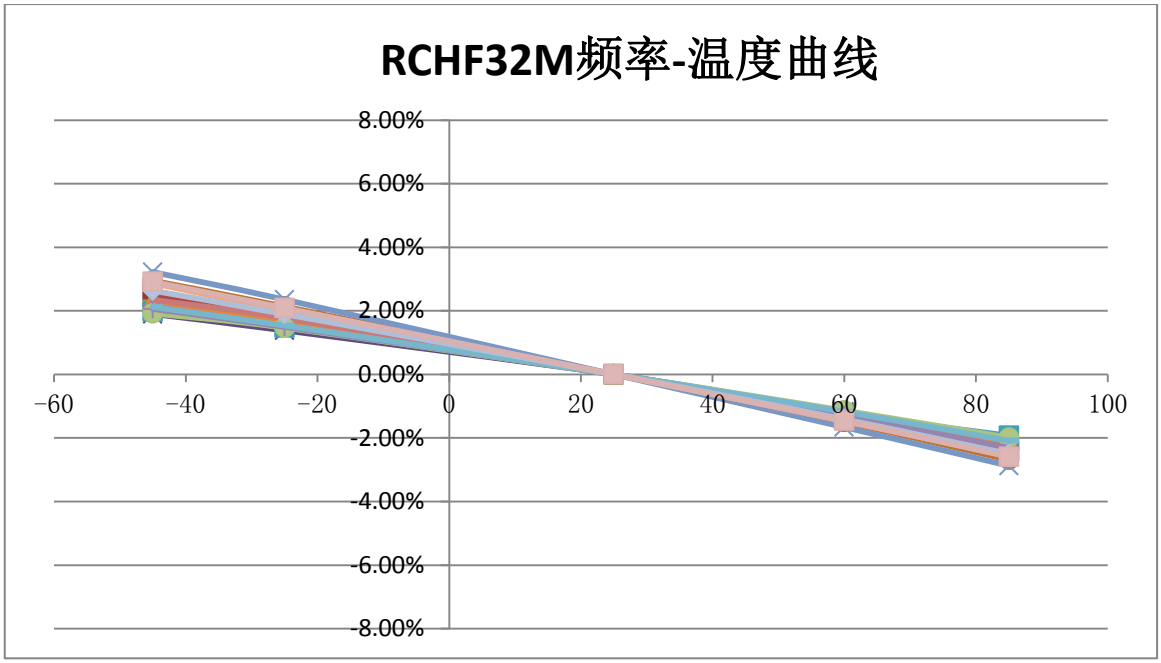
[注1]：此项指标由量产测试保证
[注2]：此项指标基于特征参数提取

以下为RCHF的频率-温度测试曲线，仅供测试参考：




RCHF16M频率-温度曲线
5.00%
4.00%
3.00%
2.00%
1.00%
0.00%
-1.00%
-2.00%
-3.00%
-4.00%
-5.00%
-60
-40
-20
0
20
40
60
80
100


RCHF24M频率-温度曲线
8.00%
6.00%
4.00%
2.00%
0.00%
-2.00%
-4.00%
-6.00%
-8.00%
-60
-40
-20
0
20
40
60
80
100



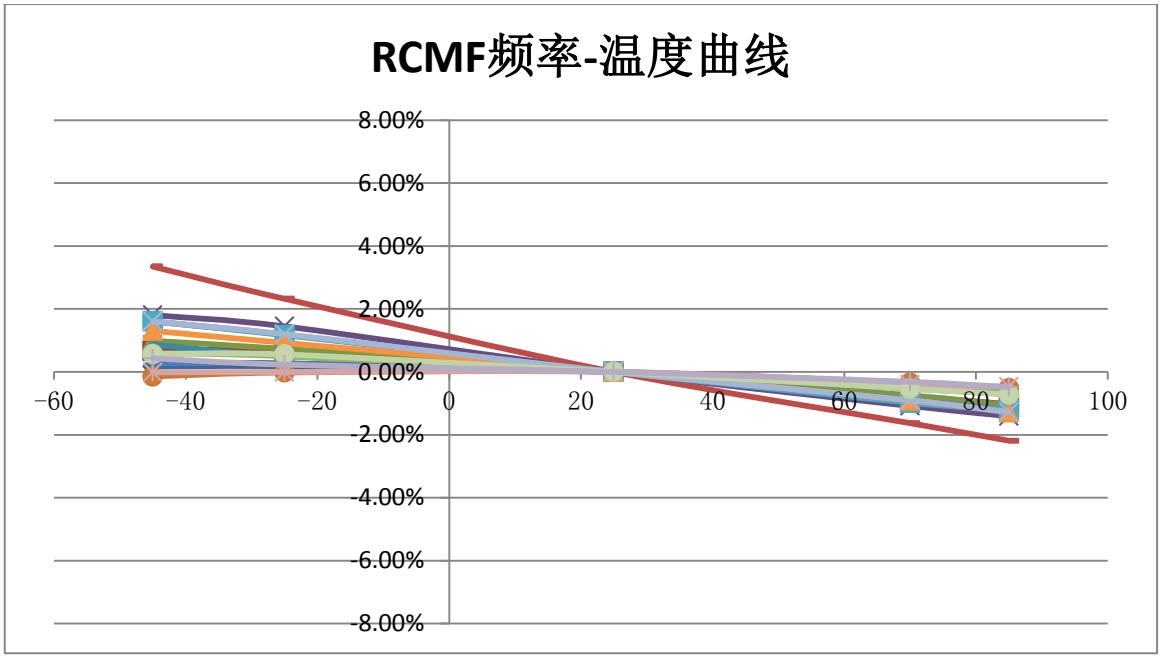
内部中频 RC 振荡器（RCMF）

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_{RCMF}[1]	RCMF 低功耗振荡频率	VDD=2.0~5.5V T=25℃	1.98	2	2.02	MHz
I_{DD_RCMF}	RCMF 功耗	VDD=2.0~5.5V TA=25℃		20		uA
t_{START}	RCMF 启动时间	VDD=5V TA=25℃		3	6	us
ACC_{RCMF}[2]	全温区 RCMF 变化范围	VDD=2.0~5.5V TA=-40~+85℃	-3	-	4	%

表 2-12 内部中频 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

以下为RCMF的频率-温度测试曲线，仅供设计参考：



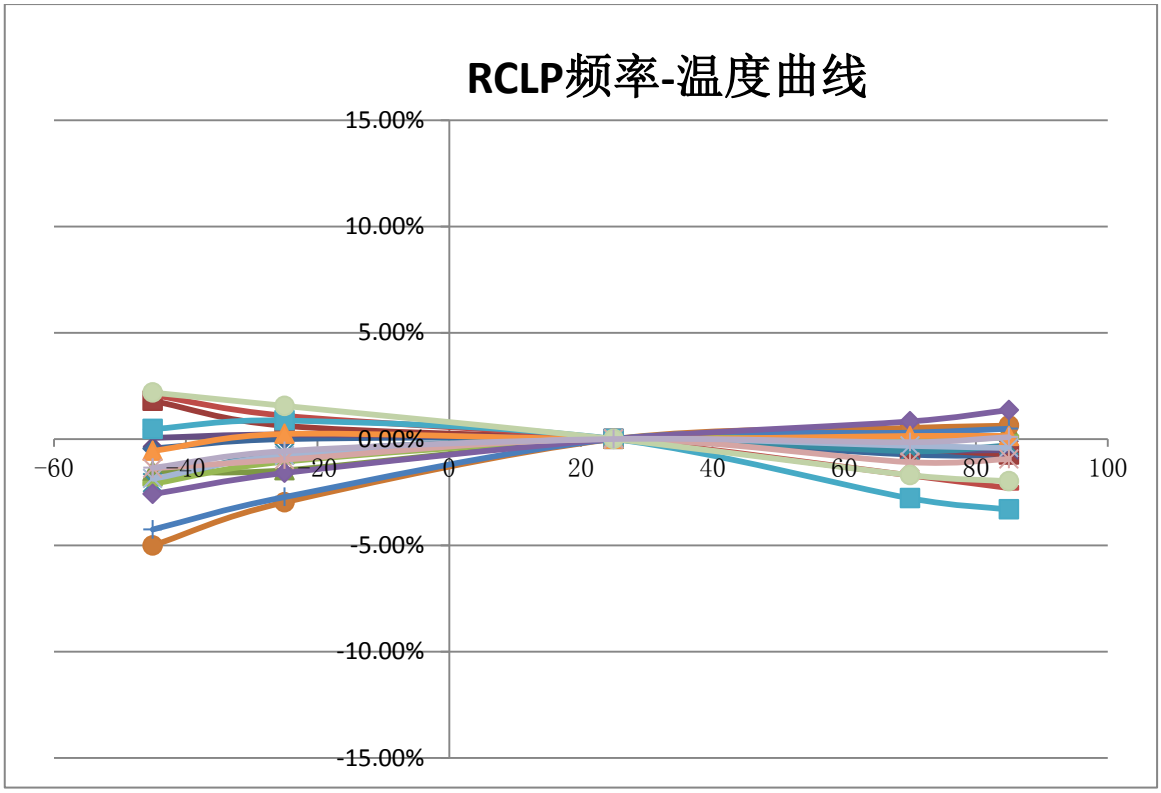
内部低频 RC 振荡器（RCLP）

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}[1]	RCLP 低功耗振荡频率	VDD=2.0~5.5V T=25℃		32		KHz
I_{DD_RCLP}	RCLP 功耗	VDD=2.0~5.5V T=25℃		200		nA
t_{START}	RCLP 启动时间			400		us
ACC_{RCLP}[2]	全温区 RCLP 变化范围	VDD=2.0~5.5V T_A=-40~85℃	-5	-	3	%

表 2-13 内部 RCLP 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

以下为RCMF的频率-温度测试曲线，仅供设计参考：



2.4.7 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=1.8~5.5V T=25℃	32		64	MHz
I_{DD_PLL}	PLL 功耗	F_{PLLO} = 32MHz		400		uA
		F_{PLLO} = 64MHz		600		
t_{LOCK}	PLL 锁定时间			65		us

表 2-14 PLL_H 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_{PLL}	PLL 输出频率	VDD=1.8~5.5V T=25℃		16.384		MHz
I_{DD_PLL}	PLL 功耗	F_{REF} = 32768Hz F_{PLLO} = 16.384MHz				uA
t_{LOCK}	PLL 锁定时间					us

表 2-15 PLL_L 参数

2.4.8 ADC 特性

2.4.8.1 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	
VDD	工作电压范围		2.2	-	5.5	V
T_J	工作结温范围		-40	-	125	°C
V_{AIN}	输入电压范围	输入 Buffer 旁路	0	-	4.8	V
		输入 Buffer 使能	0.2	-	4.8	
R_{AIN}	外部输入阻抗	输入 Buffer 旁路	10	-	-	MΩ
		输入 Buffer 使能	100	-	-	
R_{ADC}	采样开关阻抗	输入 Buffer 旁路	-	5	-	KΩ
		输入 Buffer 使能	-	2	-	
F_{CLK}	ADC 工作时钟频率		-	1	2	MHz
F_S	ADC 采样频率 F_{CLK}=1.024Mhz	OSR=1024		1000		sps
		OSR=2048		500		
		OSR=4096		250		
		OSR=8192		125		
		OSR=16384		62.5		
DNL[1]	差分非线性	OSR=16384	-0.3	-	0.3	LSB
		OSR=1024	-0.05	-	0.05	LSB
INL[1]	积分非线性	OSR=16384	-8		4	LSB
E_O[1]	失调误差	OSR=2048	-	7	-	LSB
E_G[1]	增益误差	OSR=2048		-2		%

表 2-16 ADC 参数

[注 1]：此项参数基于特征参数提取

2.4.9 温度传感器

芯片出厂时经过温度定标，定标条件是 VDD=5.0V，T_A=30+/-1°C。在此条件下，使用 ADC 采样并转换温度传感器输出电压，将转换结果保存在 Flash 指定地址。

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
Reso	分辨率	外部累加模式，OSR=16384		22.5		LSB/°C
		内部累加模式，trim=0x7FF		5.6		
Linerity	全温区线性度[1]		-	+/-1	+/-2	°C
I_{DDA}	温度传感器功耗（包含 ADC）[2]			170		uA
T_{acc}	温度传感器转换时间[2]	F_{ADC} = 1Mhz 累加器模式, OSR=2048		2		ms
		F_{ADC} = 1Mhz 累加器模式，OSR=16384		16		

表 2-17 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

2.4.10 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		-	-	512K	bytes
	Page size		-	512	-	bytes
	Sector size		-	2K	-	bytes
T_{PROG}	Word Program Time		-	25	-	μs
T_{ERASE}	Sector/Page Erase		-	2	-	ms
	Chip Erase		-	8	-	ms
N_{ED}	Sector Endurance		-	100,000	-	cycles
T_{DR}	Data Retention	T=85℃ After 10K cycling	10	-	-	yrs

表 2-18 Flash 参数

2.4.11 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	VDD=5V, T_A=25C	-	2.25	2.35	V
		VDD=3V, T_A=25C	-	1.52	1.55	
V_{IH}	输入高电平	VDD=5V, T_A=25C	-	2.55	2.6	V
		VDD=3V, T_A=25C	-	1.7	1.75	
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		1.0		V
		V_{DD}=5V I_{SINK}=10mA		0.45		
		V_{DD}=3V I_{SINK}=10mA		0.95		V
		V_{DD}=3V I_{SINK}=5mA		0.32		
V_{OH}	输出高电平	V_{DD}=5V I_{SOURCE}=20mA		3.25		V
		V_{DD}=5V I_{SOURCE}=10mA		4.3		
		V_{DD}=3V I_{SOURCE}=10mA		2.15		
		V_{DD}=3V I_{SOURCE}=5mA		2.9		
R_{PU}	弱上拉电阻			100		KΩ

表 2-19 普通 I/O 参数

[注]以上参数基于特征参数提取

高推挽驱动 IO（PD8, PE2）

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	VDD=5V, T_A=25C	-	2.25	2.35	V
		VDD=3V, T_A=25C	-	1.52	1.55	
V_{IH}	输入高电平	VDD=5V, T_A=25C	-	2.55	2.6	V
		VDD=3V, T_A=25C	-	1.7	1.75	
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}=45mA		0.45		V
		V_{DD}=5V I_{SINK}=20mA		0.2		
		V_{DD}=3V I_{SINK}=20mA		0.3		
		V_{DD}=3V I_{SINK}=10mA		0.15		
V_{OH}	输出高电平	V_{DD}=5V I_{SOURCE}=20mA		4.32		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	
		V_{DD}=5V I_{SOURCE}=10mA		4.75		
		V_{DD}=3V I_{SOURCE}=20mA		2.25		
		V_{DD}=3V I_{SOURCE}=10mA		2.8		
R_{PU}	弱上拉电阻			100		KΩ

表 2-20 高驱动 I/O 参数

[注]以上参数基于特征参数提取

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
$T_{DFILTER}$	数字滤波长度[1]	VDD=1.8~5.5V -40℃≤T_A≤85℃		8		ms

表 2-21 NRST 引脚参数

注：[1] 此项参数基于特征参数提取

2.4.12 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				uA
		VDD=3V				
V_{LCD}	LCD 偏置电压	-	0.547×VDD		VDD	V

表 2-22 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 2-23 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：

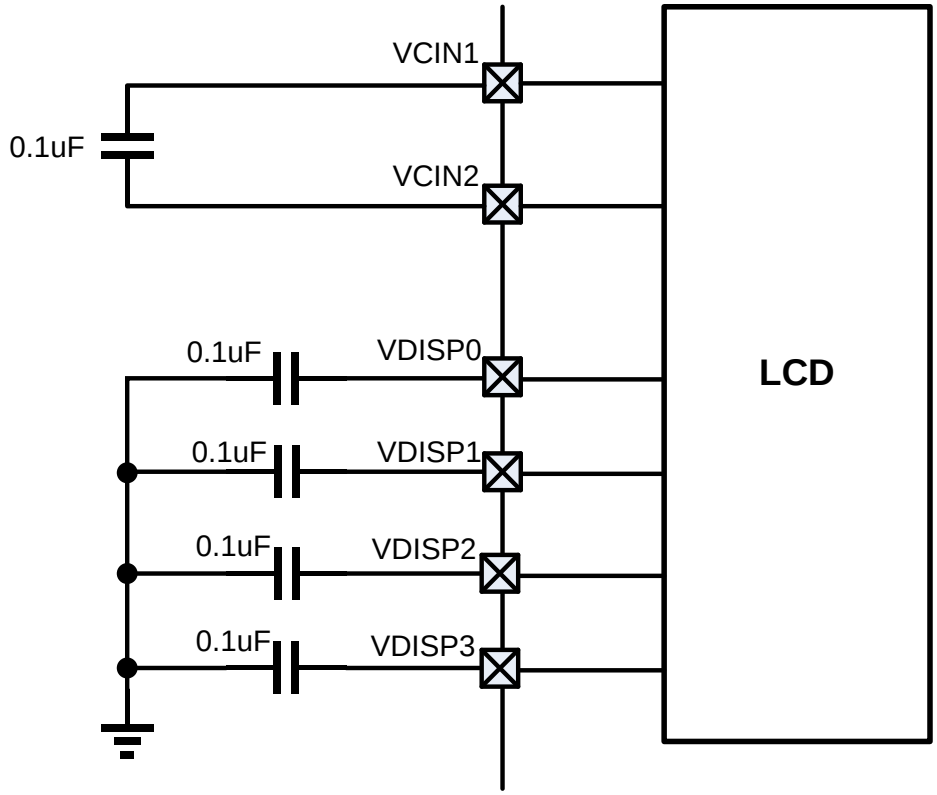


图 2-2 LCD 片外电容驱动模式电容连接

Booster 升压模式

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}	Booster 模式下的 LCD 工作电流(空载) [1]	VDD=3V		5		uA
V_{LCD}	内部产生的 LCD 偏置电压	-	2.5		5.5	V
C_{ext}	VLCD 引脚外接稳压电容	-		0.1		uF

表 2-24 LCD Booster 升压驱动

3 电源管理单元（PMU）

3.1 电源方案

3.1.1 电源域划分

- VDD

芯片的主电源（VDD）的典型工作电压范围是 1.8~5.5V。

其中，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 1.8V。如果芯片电源上升时间很短（小于几个 ms），则上电复位释放电压主要由 RC 延迟决定，典型情况下将略低于 1.8V，低温下可能略高于 1.8V。

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 BOR_PDRCFG 获得 4 个阈值档位，默认值为 1.6V。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 PDRCFG 配置 4 个档位，默认值 1.4V 左右。

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。

- 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。

- VLCD

VLCD 是由芯片内部电荷泵产生的 LCD 驱动电源，通过电荷泵可以实现 VLCD 高于电源电压 VDD，当芯片供电电压较低时仍可以获得较好的段码 LCD 显示效果。VLCD 的输出范围是 2.5V 到 5.5V。

3.1.2 电源结构图

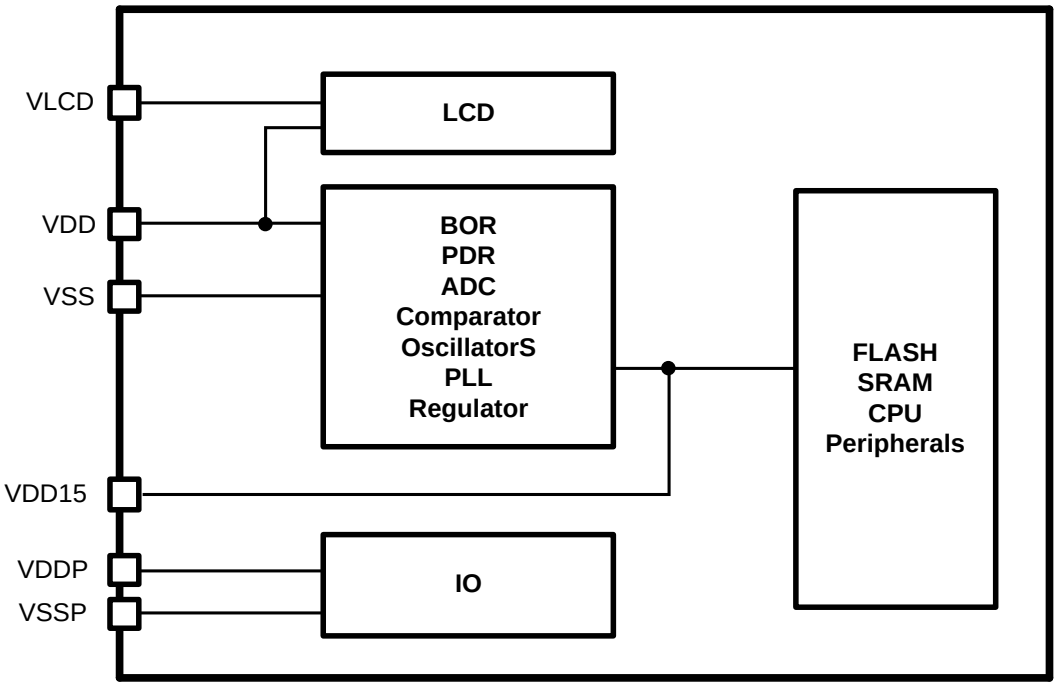


图 3-1 芯片电源结构图

3.2 功耗模式

3.2.1 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 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 工作在低功耗模式下，仅部分外设可以运行

此外，ACTIVE 模式下的运行功耗也可以通过以下手段降低：

- 降低系统时钟频率
- 关闭不使用的外设的总线时钟和工作时钟

功耗模式	典型功耗	唤醒条件	芯片状态	典型唤醒时间[1]
ACTIVE	126uA/MHz		正常工作	-
LP Run	30uA@32KHz	软件主动退出	低速工作	-
SLEEP	3.6uA	电源检测中断 比较器中断 RTC 定时中断 LPTIM 定时中断 BSTIM 定时中断 LPUART 唤醒 IO 引脚中断 WKUPx 唤醒 32K 晶振停振 看门狗复位 NRST 引脚复位	CPU 休眠 关闭 RCHF、PLL、ADC 等 保持 BG_QS 开启，关闭 LDO15，RTC 走时	8us
DEEPSLEEP	1.5uA		CPU 休眠[2] 关闭 RCHF、PLL、ADC 等 关闭 BG_QS，关闭 LDO15，RTC 走时	8us

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。

不同功耗模式之间的转换如下图所示：

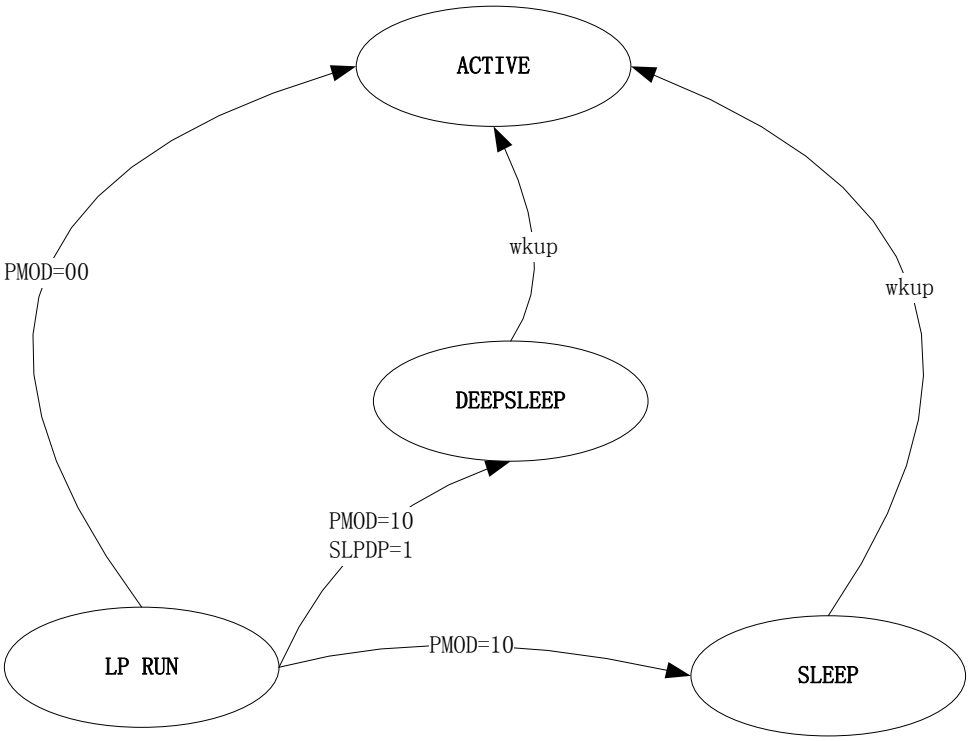


图 3-2 芯片功耗模式转换

3.2.1 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：

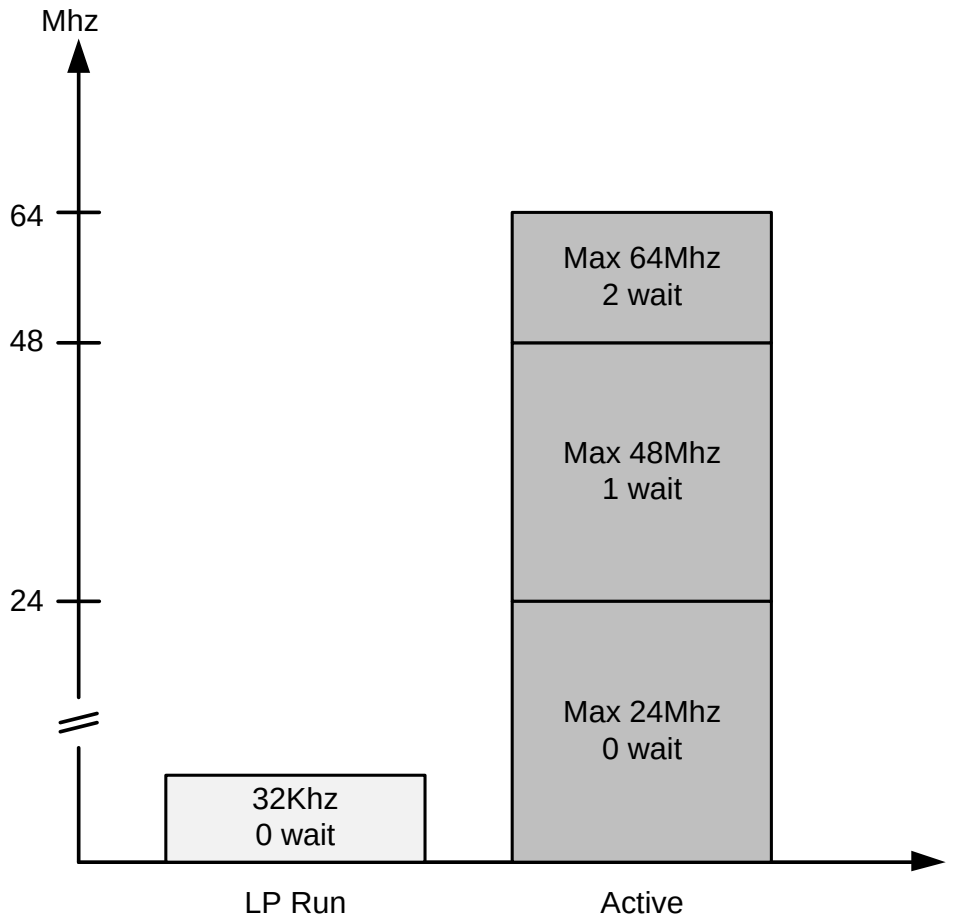


图 3-3 功耗模式与系统主频

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 Run	32Khz	XTLF, RCLP	0	XTLF, RCLP

3.2.2 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，建议软件开启 Flash Prefetch 来提升指令执行效率。

3.2.3 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 LSCLK

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 LSCLK
- 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，随后使 LDO 进入低功耗模式。SVD、比较器、ADC 仍可以在 LPRUN 模式下工作。

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 或 PLL
- 等待时钟建立后配置系统时钟为 RCHF 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。

3.2.4 SLEEP 模式

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

进入 SLEEP 模式

软件按如下步骤进入 SLEEP 模式：

- 配置 PMOD 寄存器为 10
- 执行 WFI 或 WFE 指令

SLEEP 模式下的硬件行为

进入 SLEEP 模式后芯片关闭 CPU 时钟，Flash 进入 STOP 模式，硬件自动关闭 RCHF、PLL、XTHF、

TRNG，SVD、比较器、ADC 仍可以在 SLEEP 模式下工作。

数字外设模块可以使用 RCMF、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

3.2.5 DEEPSLEEP 模式

DEEPSLEEP 是芯片最低功耗模式，此模式下由于关闭了内部基准源，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 SLPDP 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、ADC、比较器仍可以在 DEEPSLEEP 模式下工作。

数字外设模块可以使用 RCMF、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

3.3 唤醒源

唤醒源	应用	可唤醒模式	
		Sleep	DeepSleep
停振检测	可屏蔽，32786Hz 晶振停振时唤醒芯片	√	√
SVD	可屏蔽，在电源电压跌落至阈值以下或升高至阈值以上时唤醒芯片	√	√
比较器	可屏蔽，用于外部事件唤醒	√	√
ADC	可屏蔽，ADC 中断 ADC_IF 可用于唤醒	√	√
RTC	可屏蔽，根据需要的唤醒周期设置	√	√
IO 引脚中断	可屏蔽，用于外部事件唤醒	√	√
Debug	不可屏蔽，用于 debug 唤醒	√	√
LPUART	可屏蔽，接收数据唤醒	√	√
WKUPx 引脚	可屏蔽，用于外部输入唤醒	√	√
NRST	不可屏蔽，用于全局复位	√	√
LPTIM32	可屏蔽，用于定时唤醒	√	√
BSTIM32	可屏蔽，用于定时唤醒	√	√

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模块WKFR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。

3.4 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模块CR.WKFSEL）。最快情况下芯片唤醒后将以24MHz时钟启动。

休眠时AHBPRES寄存器复位将复位成00，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。

3.5 寄存器

模块起始地址：0x4000_2000

offset 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

3.5.1 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	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						LDO15EN	LDO15EN_B
位权限	U-0						R-1	R-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	-						PMOD	
位权限	U-0						R/W-00	

位号	助记符	功能描述
31:18	-	未实现，读为 0
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：RCHF-32MHz
9	**SLPDP**	DeepSleep 控制寄存器

位号	助记符	功能描述
		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
7:2	**-**	未实现，读为 0
1:0	**PMOD**	低功耗模式配置寄存器 00：Active mode 01：LPRUN mode 10：Sleep mode / DeepSleep mode 11：RFU

3.5.2 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	-					STPCLR	T1A	
位权限	U-0					R/W-0	R/W-01	

位号	助记符	功能描述
31:3	**-**	未实现，读为 0
2	**STPCLR**	Flash Stop 唤醒控制 0：Stop 信号等待时钟建立后同步清零 1：Stop 信号异步清零
1:0	**T1A**	可编程额外唤醒延迟 在 Sleep/DeepSleep 模式下，RCHF 时钟到来后，根据此寄存器配置等待额外延迟时间后，再读取 Flash 校验字 00：0us 01：2us 10：4us 11：8us

3.5.3 休眠唤醒源标志寄存器（PMU_WKFR）

名称	PMU_WKFR

名称	PMU_WKFR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ADCWKF	-		RTCWKF	SVDWKF	LFDETWKF	-	IOWKF
位权限	R-0	U-0		R-0	R-0	R-0	U-0	R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		LPU1WKF	LPU0WKF	COMP_OOWF	COMP_WINF	COMP2WKF	COMP1WKF
位权限	U-0		R-0	R-0	R-0	R-0	R-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					LPTWKF	BSTWKF	DBGWKF
位权限	U-0					R-0	R-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WKPxF							
位权限	R/W-00000000							

位号	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
30:29	-	未实现，读为 0
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	-	未实现，读为 0
24	IOWKF	IO 中断唤醒标志，中断撤销时硬件自动清零
23:22	-	未实现，读为 0
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	COMP_OOWF	比较器 out-of-window 唤醒标志，中断撤销时硬件自动清零
18	COMP_WINF	比较器 window 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
15:11	-	未实现，读为 0
10	LPTWKF	LPTIM32 中断唤醒标志，中断撤销时硬件自动清零
9	BSTWKF	BSTIM32 中断唤醒标志，中断撤销时硬件自动清零
8	DBGWKF	CPU Debugger 唤醒标志，软件写 1 清零
7:0	WKPxF	WKUPx Pin 唤醒标志，软件写 1 清零

3.5.4 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	-						SLPEIE	LPREIE
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	**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

3.5.5 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	-						SLPEIF	LPREIF
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	**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

4 CPU

4.1 概述

FM33A0xxEV 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
- MPU
- 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

4.1.1 处理器配置

Feature	Options	FM33A0xxEV Config
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 4-1 FM33A0xxEV 处理器配置表

4.2 寄存器

主要内核寄存器列表

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

表 4-2 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。

4.3 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。

4.3.1 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	WWDT	窗口看门狗中断	0x0000_0040
17	8	settable	SVD	电源监测报警中断	0x0000_0044
18	9	settable	RTC	RTC 中断	0x0000_0048
19	10	settable	FLASH	NVMIF 中断	0x0000_004C
20	11	settable	CMU	XTLF 停振检测中断 XTHF 停振检测中断 系统时钟切换错误中断	0x0000_0050
21	12	settable	ADC	ADC 或 CIC 中断	0x0000_0054
22	13	settable	SPI0	SPI 中断	0x0000_0058
23	14	settable	SPI1		0x0000_005C
24	15	settable	SPI2		0x0000_0060

Position	Priority	Priority type	Acronym	Description	Address
25	16	settable	UART0	UART 中断	0x0000_0064
26	17	settable	UART1		0x0000_0068
27	18	settable	UART2		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	LPUART0	低功耗 UART 中断	0x0000_0080
33	24	settable	I2Cx	I2C0&1 中断	0x0000_0084
34	25	settable	-	-	0x0000_0088
35	26	settable	Crypto	加密算法或 TRNG 中断 整合 AES/PAE/HASH	0x0000_008C
36	27	settable	LPTIM	低功耗定时器中断	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	Comp	模拟比较器中断	0x0000_009C
40	31	settable	BTx	Basic Timer x 中断	0x0000_00A0
41	32	settable	QSPI	QuadSPI 中断	0x0000_00A4
42	33	settable	ETx	Extended Timer x 中断	0x0000_00A8
43	34	settable	BSTIM	基本定时器中断	0x0000_00AC
44	35	settable	SPI3	SPI 中断	0x0000_00B0
45	36	settable	SPI4		0x0000_00B4
46	37	settable	GPIO	外部引脚中断	0x0000_00B8
47	38	settable	LPUART1	低功耗 UART 中断	0x0000_00BC

表 4-3 Cortex-M0 中断向量表

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

4.3.2 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。

4.3.3 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

表 4-4 Cortex-M0 hardfault 类型

FM33A0xxEV 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。

4.3.4 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。

4.4 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

4.4.1 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

4.4.1.1 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	SEPERATE	0	R

4.4.1.2 MPU 控制寄存器

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

4.4.1.3 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

4.4.1.4 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

4.4.1.5 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	-	R/W

	可缓冲		
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

4.5 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》

4.5.1 调试功能引脚

FM33A0xxEV 使用 SWD 调试接口，用户模式下最少仅需 2 线（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。

4.5.2 调试状态下的看门狗控制

看门狗在调试模式下可以保持使能或关闭。软件或 Debugger 可以通过 DBG_CR 寄存器配置看门狗在调试状态下是否运行。

4.5.3 DEBUG 的复位

内核的 DEBUG 部分受上下电复位和独立看门狗影响，其他系统复位源如引脚复位、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)。

4.6 寄存器

模块起始地址：0x4000_0000

offset 地址	名称	符号
DBG(模块起始地址: 0x40000000)		
0x04	DEBUG 配置寄存器 （Debug Configuration Register）	DBG_CR
0x08	HardFault 查询寄存器 （HardFault Flag Register）	DBG_HDFR

4.6.1 MCU DEBUG 配置寄存器（DBG_CR）

FM33A0xxEV 扩展了 DBG_CR 寄存器，用于配置 Debug 状态下的看门狗和定时器。DBG_CR 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	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DBG_LPT_STOP	DBG_BST_STOP	DBG_ET4_STOP	DBG_ET3_STOP	DBG_ET2_STOP	DBG_ET1_STOP	DBG_BT2_STOP	DBG_BT1_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
31:16	-	RFU：未实现，读为 0
15	DBG_LPT_STOP	Debug 状态下 LPTIM 使能控制位 1：Debug 时关闭 LPTIM 0：Debug 时保持 LPTIM 原来状态
14	DBG_BST_STOP	Debug 状态下 BSTIM 使能控制位 1：Debug 时关闭 BSTIM 0：Debug 时保持 BSTIM 原来状态
13	DBG_ET4_STOP	Debug 状态下 ET4 使能控制位 1：Debug 时关闭 ET4 0：Debug 时保持 ET4 原来状态
12	DBG_ET3_STOP	Debug 状态下 ET3 使能控制位 1：Debug 时关闭 ET3 0：Debug 时保持 ET3 原来状态
11	DBG_ET2_	Debug 状态下 ET2 使能控制位

位号	助记符	功能描述
	STOP	1：Debug 时关闭 ET2 0：Debug 时保持 ET2 原来状态
10	DBG_ET1_STOP	Debug 状态下 ET1 使能控制位 1：Debug 时关闭 ET1 0：Debug 时保持 ET1 原来状态
9	DBG_BT2_STOP	R Debug 状态下 BT2 使能控制位 1：Debug 时关闭 BT2 0：Debug 时保持 BT2 原来状态
8	DBG_BT1_STOP	Debug 状态下 BT1 使能控制位 1：Debug 时关闭 BT1 0：Debug 时保持 BT1 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

4.6.2 HardFault 查询寄存器（DBG_HDFR）

名称	DBG_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_ADDR_FLAG	地址非对齐访问错误标志，写 1 清零 1：地址非对齐访问错误 0：未进行地址非对齐访问
5	DABORT_RESP_FLAG	非法地址访问错误标志，写 1 清零 1：总线传输中访问了非法地址导致 HRESP 为高产生错误 0：未访问非法地址
4	SVCUNDEF_FLAG	SVC instructions 未定义标志，写 1 清零 if the SVCall priority is lower than the currently activelevel,

位号	助记符	功能描述
		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

5 总线与存储（Bus and Memories）

5.1 系统总线

FM33A0xxEV 总线架构包含以下主要部件：

- 2 个 Master
  - Cortex-M0 内核
  - DMA 控制器
- 5 个 Slave
  - Flash 存储器
  - SRAM1
  - SRAM2
  - AHB 外设和 AHB-to-APB Bridge
  - QSPI

FM33A0xxEV 的系统总线示意图如下。

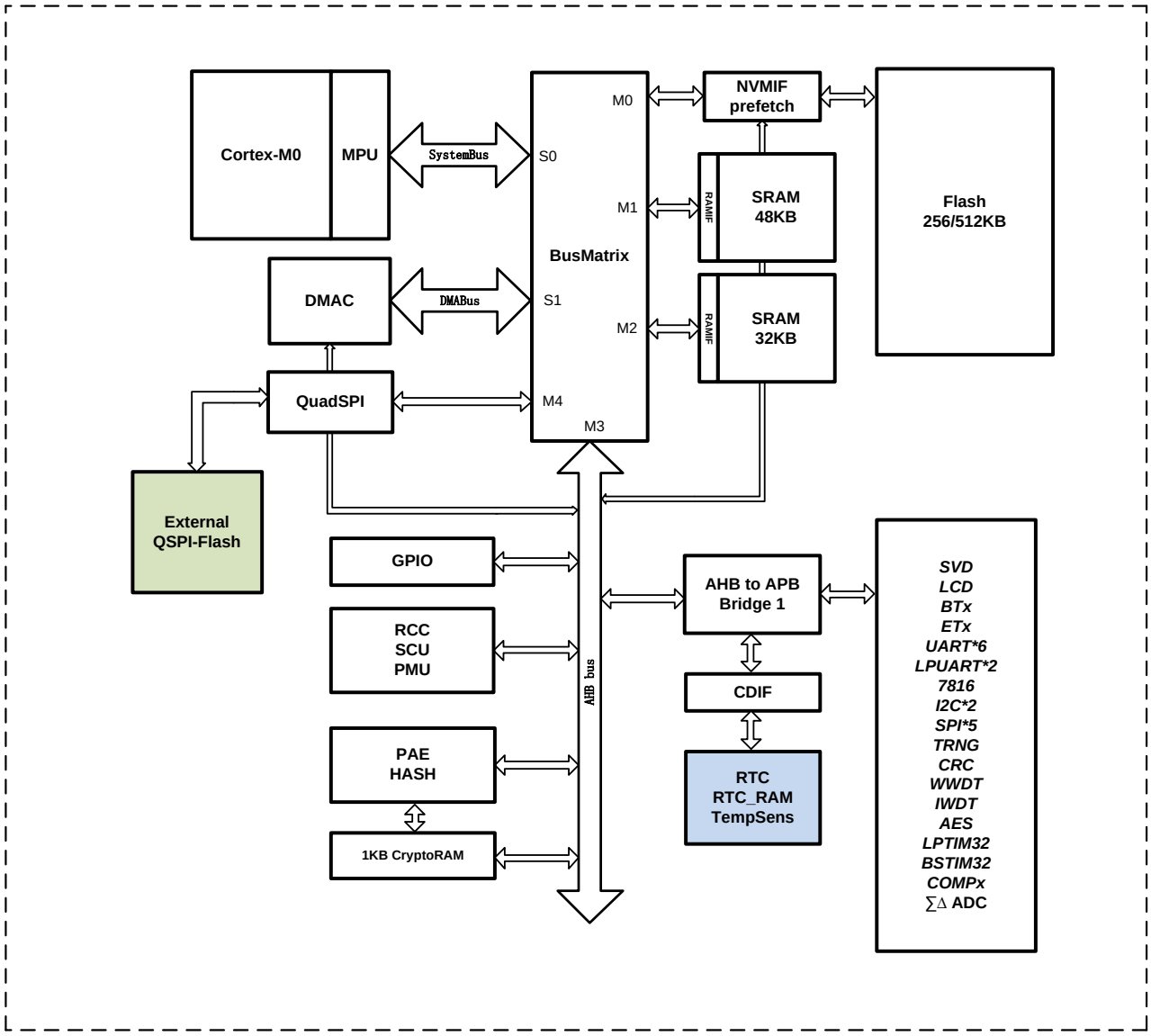

Cortex-M0
MPU
SystemBus
S0
M0
NVMIF
prefetch
Flash
256/512KB
M1
RAMIF
SRAM
48KB
BusMatrix
DMAC
DMABus
S1
M2
RAMIF
SRAM
32KB
QuadSPI
M4
M3
External
QSPI-Flash
GPIO
AHB to APB
Bridge 1
SVD
LCD
BTx
ETx
UART*6
LPUART*2
7816
I2C*2
SPI*5
TRNG
CRC
WWDT
IWDT
AES
LPTIM32
BSTIM32
COMPx
∑Δ ADC
RCC
SCU
PMU
AHB bus
CDIF
PAE
HASH
RTC
RTC_RAM
TempSens
1KB CryptoRAM

图 5-1 芯片总线架构图

5.2 存储空间分配

5.2.1 概述

Flash 页（Page）大小为 512 字节，扇区（sector）大小为 2KB。

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

当芯片从 Flash 启动时，FM33A0xxEV 的地址空间分配如下图（512KB Flash，80KB RAM）：

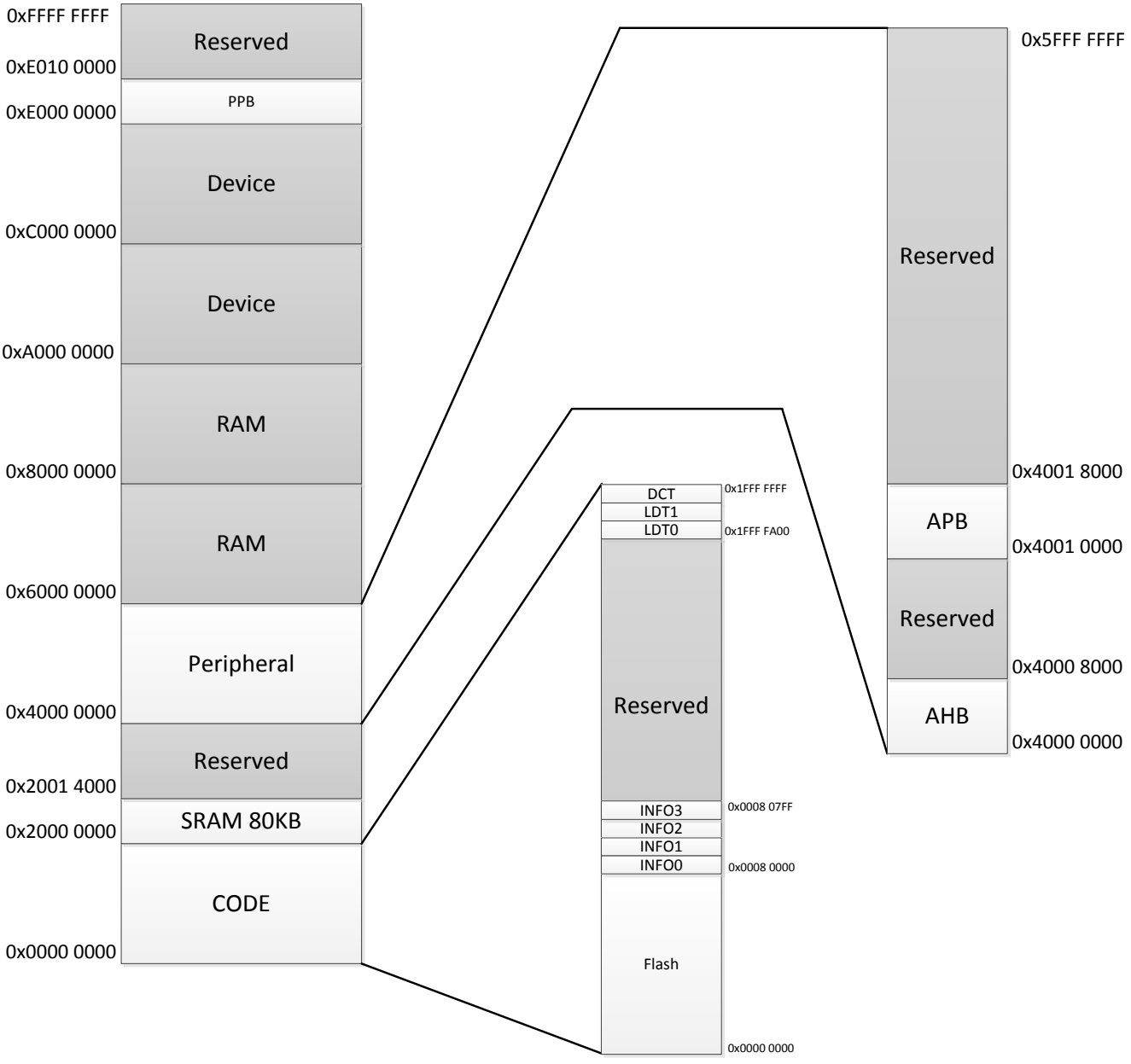


图 5-2 总线地址分配（512KB flash）

256KB flash+48KB RAM：

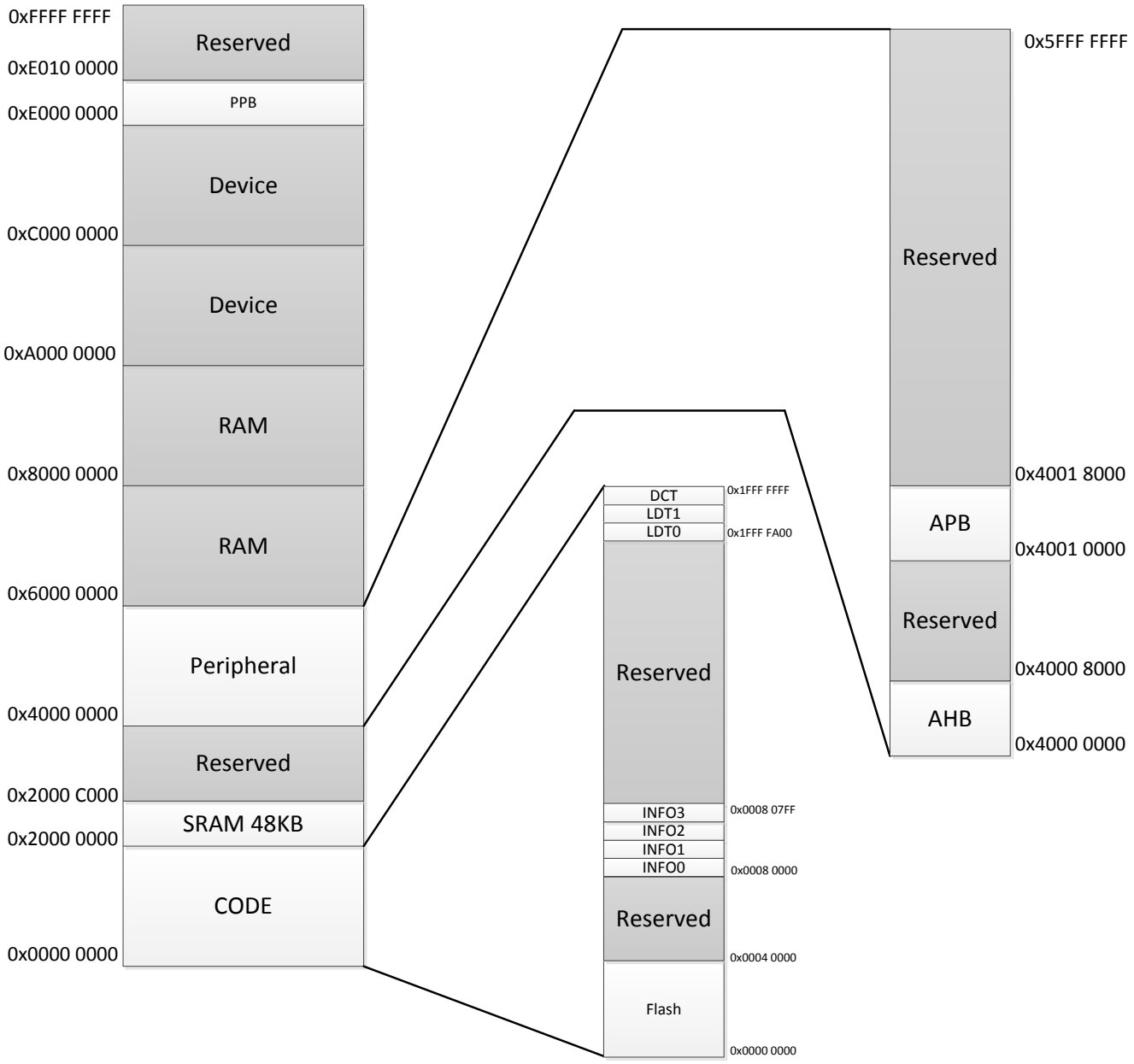


图 5-3 总线地址分配（256KB flash）

5.2.2 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_BFFF	48KB	SRAM0
	0x2000_C000~0x2001_3FFF	32KB	SRAM1
	0x4000_0000~0x4000_03FF	1KB	SCU
	0x4000_0400~0x4000_07FF	1KB	DMA
	0x4000_0800~0x4000_0CFF	1KB	QSPI Registers
	0x4000_0C00~0x4000_0FFF	1KB	GPIO
	0x4000_1000~0x4000_13FF	1KB	NVMIF Registers
	0x4000_1400~0x4000_17FF	1KB	PAE
	0x4000_1800~0x4000_1BFF	1KB	HASH
	0x4000_1C00~0x4000_1FFF	1KB	Crypto-RAM
	0x4000_2000~0x4000_23FF	1KB	PMU
	0x4000_2400~0x4000_27FF	1KB	CMU
	0x4000_2800~0x4000_2BFF	1KB	RMU
	0x4001_0000~0x4001_FFFF	64KB	APB
	0x9000_0000~0x9FFF_FFFF	256MB	QSPI-XIP space
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
	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	I2C0
	0x4001_2800~0x4001_2BFF	1KB	SVD
	0x4001_2C00~0x4001_2FFF	1KB	RAMBIST
	0x4001_3000~0x4001_33FF	1KB	TIMERS
	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	SPI3
	0x4001_5000~0x4001_53FF	1KB	I2C1
	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	SPI4
	0x4001_6800~0x4001_6BFF	1KB	UART1
	0x4001_6C00~0x4001_6FFF	1KB	UART2
	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	ANTEST(BUF4TST)

	0x4001_E000~0x4001_E3FF	1KB	CPIF controller
	0x4001_F000~0x4001_FFFF	4KB	CPIF (RTC, RTC_RAM, misc)

表 5-1 总线地址分配表

其中 CPIF 内部地址划分如下：

总线	地址边界	空间	外设
CPIF	0x4001_F000~0x4001_F3FF	1KB	RTC
	0x4001_F400~0x4001_F7FF	1KB	RTC_RAM
	0x4001_F800~0x4001_FBFF	1KB	Power, clock, ADC
	0x4001_FC00~0x4001_FFFF	1KB	IO

表 5-2 CPIF 接口内部地址分配

5.3 RAM

5.3.1 系统 RAM

FM33A0xxEV 总共包含最大 80KB SRAM

SRAM 地址空间范围是 0x2000_0000~0x2001_3FFF，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。

5.3.2 算法 RAM

芯片还包含了 1 块 1KB 算法 RAM，算法 RAM 位于总线地址：0x4000_1C00~0x4000_1FFF。

这块 RAM 主要供公钥加密算法加速引擎（PAE）和散列算法加速器（HASH）使用，当 PAE 或 HASH 在工作时，软件无法访问算法 RAM；当 PAE 和 HASH 不工作时，软件可以将算法 RAM 作为普通 RAM 使用。

5.3.3 RTC RAM

RTC 模块带有一块 512bytes RAM，可用于关键数据保存或者休眠模式下的自动温度补偿。

RTC RAM 的总线地址为 0x4001_F400~0x4001_F5FF。

5.4 Flash

5.4.1 概述

FM33A0xxEV 使用的 Flash 容量为 512KB；阵列组织格式包含 page(512B)、sector(2KB)、matrix(512KB)。

Main array 共包含 256 个扇区、1024 个页，另有 4 个信息页，2 个用户配置页，1 个芯片原厂保留信息页。Flash 支持页擦、扇区擦和全擦。

5.4.1.1 LDT0 page

LDT0为FMSH保留的原厂配置信息区，用户只读。

LDT0的总线地址是0x1FFF_FA00~0x1FFF_FBFF，共512字节。

以下地址保存的时钟校准数据，用户可以读取使用。

AHB地址	[31:16]	[15:0]	说明
0x1FFF_FB20	~RCLP_TRIM	RCLP_TRIM	RCLP 调校值
0x1FFF_FB34	~RCHF32TRIM	RCHF32TRIM	RCHF 32MHz 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（上电自动装载）

数据定义表如下：

助记符	[31:16]	[15:0]	功能描述
RCLP_TRIM	{8'h00, ~trim}	{8'hFF, trim}	trim[7:0]表示 8bit 调校值
RCHF32TRIM	{9'b0000_0000_0, RCHFtrim[6:0]}	{9'b1111_1111_1,RCHFtrim[6:0]}	RCHFtrim[6:0]表示 7bit 调校值
RCHF24TRIM			
RCHF16TRIM			
RCHF8TRIM			

5.4.1.2 LDT1 page

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

LDT1的总线地址是0x1FFF_FC00~0x1FFF_FDFF。

AHB addr	Bit[31:16]	Bit[15:0]	Description
0x1FFF_FC00	~OPTBYTES[15:0]	OPTBYTES[15:0]	用户选项字节低半字
0x1FFF_FC04	~OPTBYTES[31:16]	OPTBYTES[31:16]	用户选项字节高半字
0x1FFF_FC08	LOCK1		ACLOCK 配置字 1
0x1FFF_FC0C	LOCK2		ACLOCK 配置字 2
0x1FFF_FC10	LOCK3		ACLOCK 配置字 3
0x1FFF_FC14	LOCK4		ACLOCK 配置字 4
0x1FFF_FC18	~SYSCFG	SYSCFG	

表 5-3 用户配置信息区

OPTBYTES 用户选项字节定义如下：

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
23:16	DBGCFG	使能或禁止 MCUDBGCR 寄存器 0xFF：使能 MCUDBGCR 寄存器 其他：禁止 MCUDBGCR 寄存器	0xFF
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 0xCC：调试接口熔断 其他：使能调试接口保护	0xAA

表 5-4 用户选项字节

【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 位

总计128bit LOCK位用于进行ACLOCK控制。每2bit控制8KB，整个Flash共有64个8KB块区，所以占用128bit LOCK数据。

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 128~136KB），LOCK2[31:30]对应 Block31（Flash 地址空间 248~256KB），其他以此类推	0xFFFFFFFF
31:0	LOCK3	Block Lock 字 3，每 2it 对应 8KB Block 11：无保护	0xFFFFFFFF

Bitfield	助记符	功能描述	出厂默认
		01、10：软件读写保护，仅取指 00：软件读写保护，仅取指；SWD 读写保护 LOCK3[1:0] 对应 Block32（Flash 地址空间 256~264KB），LOCK3[31:30]对应 Block47（Flash 地址空间 376~384KB），其他以此类推	
31:0	LOCK4	Block Lock 字 4，每 2it 对应 8KB Block 11：无保护 01、10：软件读写保护，仅取指 00：软件读写保护，仅取指；SWD 读写保护 LOCK4[1:0] 对应 Block48（Flash 地址空间 384~392KB），LOCK4[31:30]对应 Block63（Flash 地址空间 504~512KB），其他以此类推	0xFFFFFFFF

表 5-5 LOCK 配置字

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]

Address	LOCK bits
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]
0x0004_0000 ~ 0x0004_1FFF	LOCK3[1:0]
0x0004_2000 ~ 0x0004_3FFF	LOCK3[3:2]
0x0004_4000 ~ 0x0004_5FFF	LOCK3[5:4]
0x0004_6000 ~ 0x0004_7FFF	LOCK3[7:6]
0x0004_8000 ~ 0x0004_9FFF	LOCK3[9:8]
0x0004_A000 ~ 0x0004_BFFF	LOCK3[11:10]
0x0004_C000 ~ 0x0004_DFFF	LOCK3[13:12]
0x0004_E000 ~ 0x0004_FFFF	LOCK3[15:14]
0x0005_0000 ~ 0x0005_1FFF	LOCK3[17:16]
0x0005_2000 ~ 0x0005_3FFF	LOCK3[19:18]
0x0005_4000 ~ 0x0005_5FFF	LOCK3[21:20]
0x0005_6000 ~ 0x0005_7FFF	LOCK3[23:22]
0x0005_8000 ~ 0x0005_9FFF	LOCK3[25:24]
0x0005_A000 ~ 0x0005_BFFF	LOCK3[27:26]
0x0005_C000 ~ 0x0005_DFFF	LOCK3[29:28]
0x0005_E000 ~ 0x0005_FFFF	LOCK3[31:30]
0x0006_0000 ~ 0x0006_1FFF	LOCK4[1:0]
0x0006_2000 ~ 0x0006_3FFF	LOCK4[3:2]
0x0006_4000 ~ 0x0006_5FFF	LOCK4[5:4]
0x0006_6000 ~ 0x0006_7FFF	LOCK4[7:6]
0x0006_8000 ~ 0x0006_9FFF	LOCK4[9:8]
0x0006_A000 ~ 0x0006_BFFF	LOCK4[11:10]
0x0006_C000 ~ 0x0006_DFFF	LOCK4[13:12]
0x0006_E000 ~ 0x0006_FFFF	LOCK4[15:14]
0x0007_0000 ~ 0x0007_1FFF	LOCK4[17:16]
0x0007_2000 ~ 0x0007_3FFF	LOCK4[19:18]
0x0007_4000 ~ 0x0007_5FFF	LOCK4[21:20]
0x0007_6000 ~ 0x0007_7FFF	LOCK4[23:22]
0x0007_8000 ~ 0x0007_9FFF	LOCK4[25:24]
0x0007_A000 ~ 0x0007_BFFF	LOCK4[27:26]
0x0007_C000 ~ 0x0007_DFFF	LOCK4[29:28]
0x0007_E000 ~ 0x0007_FFFF	LOCK4[31:30]

表 5-6 Lock 位与 flash 地址对应表

注意制造商模式下，ACLOCKEN 和 DBRDPEN 不起作用，不会保护 Flash 内容。

SYSCFG 配置字定义如下。如果用户系统方案中不需要外部 32768hz 晶体，建议配置 SYSCFG 将芯片配置为无晶体模式，可以降低休眠功耗。

Bit	助记符	功能描述	出厂默认
31:24	--	RFU，无效数据	0xFF
23:16	~CRYSLESS	无晶体配置校验信息 0xAA：无晶体配置，关闭 XTLF 和 FDET 其他：有晶体	0xFF
15:8	--	RFU，无效数据	0xFF
7:0	CRYSLESS	无晶体配置信息 0x55：无晶体配置，关闭 XTLF 和 FDET 其他：有晶体	0xFF

表 5-7 SYSCFG 配置字

5.4.1.3 Information3 page

Flash还包含4个information扇区，其中information3用于控制BootSwap功能。Information3总线地址是0x0008_0600~0x0008_07FF。

在OPTBYTES中BTSWPEN=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：

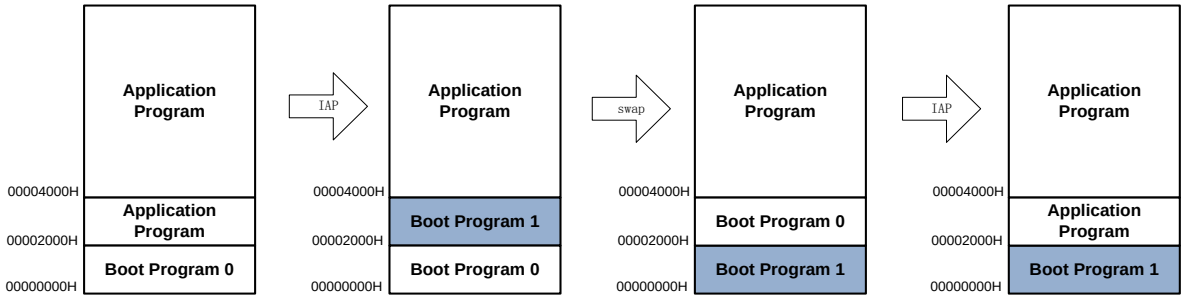


图 5-4 启动区交换示意图

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

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

逻辑地址对Flash物理地址的重映射由NVMIF模块完成，不论程序还是DEBUG都以逻辑地址进行访问。

5.4.1.4 Information2 page（Debugger only, lockable）

这个information扇区开放给用户使用，仅SWD可以擦写，软件可读。

总线地址是0x0008_0400~0x0008_05FF，共512字节。

IF2这个扇区的最高地址字节为扇区锁定标志，如果SWD将最高地址字节改写为0x55，则芯片复位后当前扇区被禁止编程，SWD进行扇区擦后，可以重新编程。

不论是否有锁定标志，这几个扇区都是SWD和软件可读的。

5.4.1.5 Information1 page（Secure Page）

IF1用于保存AES密钥等安全数据，支持密钥长度为128、192、256bit。

总线地址是0x0008_0200~0x0008_03FF，共512字节。

此扇区的最高地址为扇区读锁定标志，如果SWD将最高地址字节改写为0x55，则芯片复位后此扇区禁止读出（SWD、软件、DMA都不能读取），但是软件和SWD都可以进行擦写，经过协商后的对称加密密钥可以由软件写入此扇区。通过以上方法，保护密钥被第三方读出。

AES密钥保存必须遵循固定格式：

扇区偏移地址	数据	说明
0x00	AESKEY0	AES密钥，最多256bit，即8个word 当AES长度为128bit时，AESKEY3~0组成密钥；当AES长度为192bit时，AESKEY5~0组成密钥；当AES长度为256bit时，AESKEY7~0组成密钥
0x01	AESKEY1	
0x02	AESKEY2	
0x03	AESKEY3	
0x04	AESKEY4	
0x05	AESKEY5	
0x06	AESKEY6	
0x07	AESKEY7	
0x7F	Page Lock	Bit[31:24]最高字节为page lock

表 5-8 AES 密钥保存格式

全局复位后，芯片自动从IF1读取AES密钥并写入AES_KEYx寄存器。

5.4.1.6 Information0 page（OTP）

IF0是一个OTP page，这个page在用户模式下只能编程一次，编程后不能擦除或改写。OTP编程以word为单位，即只要32位中任意bit被改写为非1数据，这个word就不能重新编程，也不能擦除。

IF0的总线地址是0x0008_0000~0x0008_01FF，共512字节。

对IF0 page的读取没有任何限制。

5.4.2 Flash 擦除与编程

5.4.2.1 概述

FM33A0xxEV 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

5.4.2.2 Flash 擦写时钟

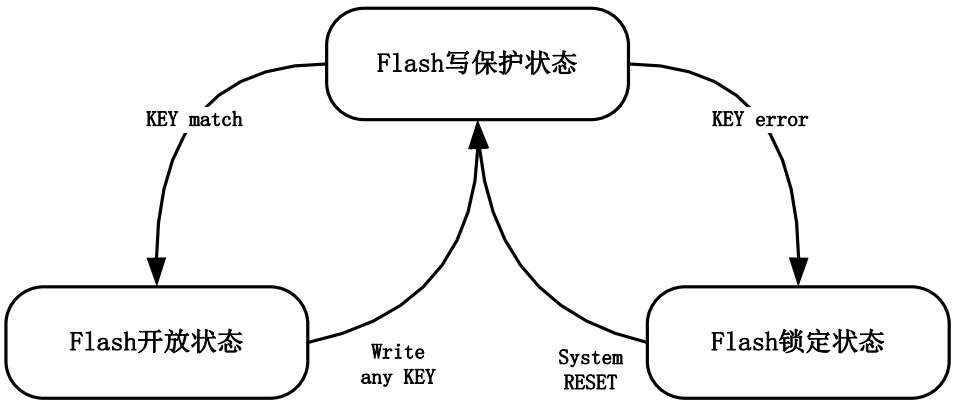
执行 Flash 擦写时使用校准后的 RCHF 时钟，但是系统时钟可以是任意时钟。NVMIF 根据当前 RCHF 实际频率设置，来产生相应的计时长度。需要支持的 RCHF 频率为 8M、16M 和 24M。

擦写时钟独立于 CPU 工作时钟。

5.4.2.3 Flash 擦写方法

FM33A0xxEV 支持 Flash 擦除操作，以及单次编程和连续编程。

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



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

5.4.2.4 全擦操作（Chip 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 INFO 扇区。SWD 启动全擦操作流程如下：

- 编程器通过 SWD 配置 ERTYPE 寄存器为 10
- 编程器通过 SWD 清除 PREQ 寄存器，置位 EREQ 寄存器
- 编程器通过 SWD 写入 Flash 全擦 Key：0x9696_9696 和 0x7D7D_7D7D
- SWD 向 Flash 任意地址写擦除请求 0x1234_ABCD
- NVMIF 启动对 Flash 的全擦，并暂停任何 Master 对 Flash 的访问
- 全擦完成后置位中断标志和全擦标志（全擦标志表示 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

5.4.2.5 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
- NVMIF 检查目标扇区是否属于被 ACLOCK 锁定的 Block，如果没有锁定则启动对目标扇区的擦除，如果被锁定则触发错误标志
- 扇区擦完成后置位中断标志
- 软件确认擦除结束后向 FlashKEY 寄存器写任意值恢复写保护

5.4.2.6 页擦操作（Page Erase）

SWD 和应用代码都可以执行页擦。操作流程如下：

- 配置 ERTYPE 寄存器为 00/11
- 清除 PREQ 寄存器，置位 EREQ 寄存器
- 写入 Flash 块擦 Key：0x9696_9696 和 0xEAEA_EAEA
- 向需要擦除的扇区内任意地址写擦除请求 0x1234_ABCD
- NVMIF 检查目标页是否属于被 ACLOCK 锁定的 Block，如果没有锁定则启动对目标页的擦除，

如果被锁定则触发错误标志

- 页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

5.4.2.7 单次编程

单次编程由软件发起，通过总线直接写 Flash，编程以 word 为单位（32bit），操作流程如下：

- 清除 EREQ 寄存器，置位 PREQ 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

5.4.2.8 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
- 向 RAM 中写入 256 字节待编程数据
- 配置 DMA 存储器通道，设定传输方向、读地址和写地址
- 使能 DMA 存储器通道
- 写入 Flash 编程 Key：0xA5A5_A5A5 和 0xF1F1_F1F1
- 软件触发 DMA 存储器通道，DMA 连续 64 次读取 RAM 并对 Flash 编程
- NVMIF 检查被编程扇区是否被 ACLOCK 锁定，如果锁定则触发错误中断并通知 DMA 停止编程
- 256 字节完全编程结束后产生中断，释放 Flash 总线
- 软件确认编程结束后向 FlashKEY 寄存器写任意值恢复写保护

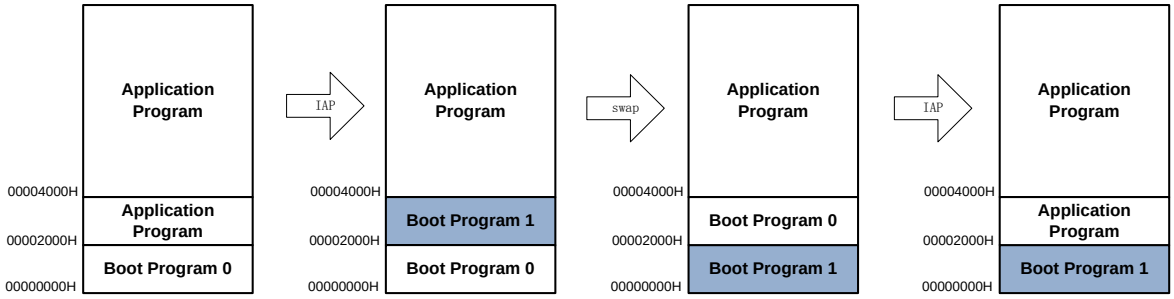
注意：如果 CPU 在 Flash 中取指时进行 Flash 擦写，则 CPU 取指将被暂停，直到擦写操作完成。如果 CPU 跳转到 RAM 中取指运行，则 Flash 擦写不会暂停 CPU 的执行。Flash 擦写过程中，若用

户希望在 RAM 中执行代码时仍然能够实时响应中断，应将中断向量表重新映射到 RAM 中。

5.4.2.9 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。

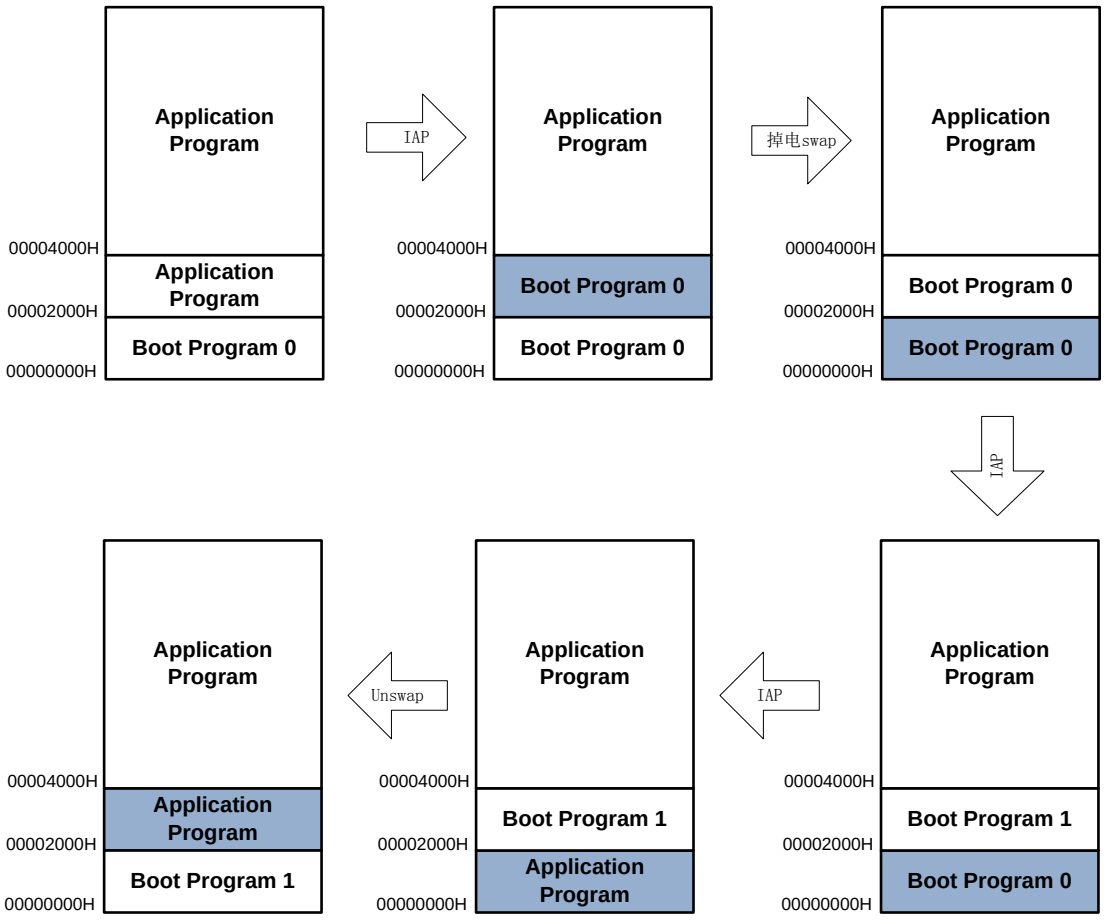
BootSwap示意图如下：



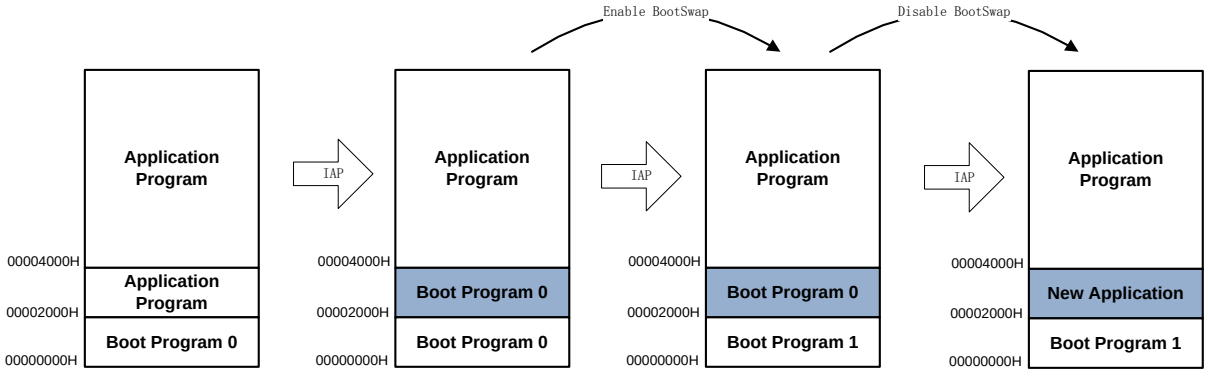
实现BootSwap功能后，假设启动代码占据0000~1FFF共8KB空间，系统升级时应先将新的启动代码写入2000~3FFF地址，然后使能BootSwap。此时有几种可能性：

- 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

逻辑地址对Flash物理地址的重映射由NVMIF模块完成，不论程序还是DEBUG都以逻辑地址进行访问。

NVMIF模块中有专门的寄存器标志（FLSIF.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。

5.4.3 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。

5.4.3.1 Debug 接口保护（DBRDP – DeBug ReaD Protection）

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 OPTBYTES.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）。

DBRDP 使能后有两个等级：基本保护等级和熔断等级

禁止 DBRDP：无保护

通过 SWD 接口可以读、擦除、编程 flash。

基本保护等级：

无法通过 SWD 接口读取或擦写 Flash main array。

退出这个等级的方法：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。

熔断保护：

SWD 无法再使用，且不可恢复。进入这个保护模式后，原厂也无法对芯片进行任何失效分析。

熔断保护的情况下，在 NRST 为低的时候，Debugger 也无法与 CPU 建立 connection。

（NRST 不复位 CFGLOADER，不重启模式诊断）

5.4.3.2 应用代码保护（ACLOCK – Application Code Lock）

ACLOCK 的主要目的是防止 hacking code 读取或篡改 Flash 中的 application code。通过 ACLOCK

功能，可以设置 CPU 对 Flash 的某些区域只能进行取指操作，不能 read-as-data，也不能擦写。

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

ACLOCK 的功能：

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

LOCK 位与 Block 访问权限的关系可以参照下表：

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

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

ACLOCK 不使能时，LOCK 寄存器内容无效。

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

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

5.4.3.3 用户模式 Flash 访问权限说明

Flash 空间访问权限分配：

Flash area	DBRDP	LOCK bits (per Block)[3]	Last byte in page	SWD	Application
Main array	ON	00	x	-	R/E/W/F
		01	x	-	对应 Block 只能取指
		10/11	x	-	对应 Block 只能取指
	OFF	00	x	R/E/W	R/E/W/F
		01	x		对应 Block 只能取指
		10/11	x	对应 Block 无法访问	对应 Block 只能取指
DCT LDT0	x	x	x	R	R
LDT1	ON	x	x	R[2]	R
	OFF	x	x	R/E/W	R
IF3	x	x	x	R/E/W	R/E/W
IF2,1,0	x	x	55	R/E	R
			others	R/E/W	R

注：

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

[2] 进行 flash 全擦后可以擦写 LDT1

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

5.5 寄存器

模块起始地址：0x4000_1000

offset 地址	名称	符号
FLS(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
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
0x48	ACLOCK 寄存器 1 （Flash Application Code Lock Register1）	FLS_ACLOCK1
0x4C	ACLOCK 寄存器 2 （Flash Application Code Lock Register2）	FLS_ACLOCK2
0x50	ACLOCK 寄存器 3 （Flash Application Code Lock Register1）	FLS_ACLOCK3
0x54	ACLOCK 寄存器 4 （Flash Application Code Lock Register2）	FLS_ACLOCK4

5.5.1 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 Config)

位号	助记符	功能描述
		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 小于等于 48Mhz 时使能 1 wait，主频大于 48Mhz 时使能 2 wait

5.5.1 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	-							PRFTEN
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	PRFTEN	指令预取指使能，在 WAIT==00 的情况下写 1 无效 (Prefetch Enable) 1：使能 Prefetch 0：禁止 Prefetch

5.5.1 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	DBGCFGEN	-						
位权限	R-1	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-0101			

位号	助记符	功能描述
31	DBGCFGEN	DBG 控制寄存器使能，若校验错误，则 ldt1fail 置 1 0：屏蔽 DBGCFG 寄存器功能 1：DBGCFG 寄存器使能
30:19	-	RFU：未实现，读为 0
18	IF2LOCK	Information2 区锁定标志 (IF2 Lock Enable) 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 Enable) 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
7:6	-	RFU：未实现，读为 0
5:4	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
3:0	DBRDPEN	Debug Port 读取保护使能 1111：基本保护 1010：熔断保护 其他：DBRDP 不使能，无保护 注意：一旦进入熔断保护状态，无法退出。

5.5.1 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)

位号	助记符	功能描述
		00/11：Page Erase 01：Sector Erase 10：Chip Erase (SWD only)
7:2	-	RFU：未实现，读为 0
1	PREQ	Program Request 软件置位，硬件完成编程后自动清零，软件无法清零
0	EREQ	Erase Request 软件置位，硬件完成擦除后自动清零，软件无法清零

注：PREQ和EREQ同时处于使能状态时，FLASH执行擦或写会将使CPU一直挂起，需要避免这种情况

5.5.1 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)

5.5.1 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)

5.5.2 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：扇区擦解锁状态 011：编程解锁状态 100：KEY 错误锁定状态，需要复位才能解锁 101/110/111：RFU
16	BTSF	BootSwap 标志寄存器 (BootSwap) 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
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 Flag. Write 1 to clear) 1：尝试对已编程的 OTP 字节进行编程 0：无 OTP 编程错误

位号	助记符	功能描述
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 Flag, write 1 to clear) 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 Flag, write 1 to clear)
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 Flag, write 1 to clear)
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 (Program Done Flag,write 1 to clear)
0	ERD	Erase Done，擦写完成标志，硬件置位，软件写 1 清零(Erase Done Flag,write 1 to clear)

5.5.3 ACLOCK 寄存器 1（FLS_ACLOCK1）

名称	FLS_ACLOCK1							
offset	0x48							
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。 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。

5.5.4 ACLOCK 寄存器 2（FLS_ACLOCK2）

名称	FLS_ACLOCK2
offset	0x4C

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。 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。

5.5.5 ACLOCK 寄存器 3（FLS_ACLOCK3）

名称	FLS_ACLOCK3							
offset	0x50							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK3[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LOCK3[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LOCK3[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCK3[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	LOCK3	ACLOCK 配置寄存器高 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。 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。

5.5.6 ACLOCK 寄存器 4（FLS_ACLOCK4）

名称	FLS_ACLOCK4							
offset	0x54							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK4[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LOCK4[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LOCK4[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCK4[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	LOCK4	ACLOCK 配置寄存器高 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。 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。

6 复位管理单元（RMU）

6.1 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 1.8V
- BOR 下电复位产生电压软件可配置为 1.75/1.7/1.65/1.6V，可关闭。
- 低功耗下电复位电路（PDR），下电复位电压可配置为 1.25/1.35/1.4/1.5V，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。

6.2 模块框图

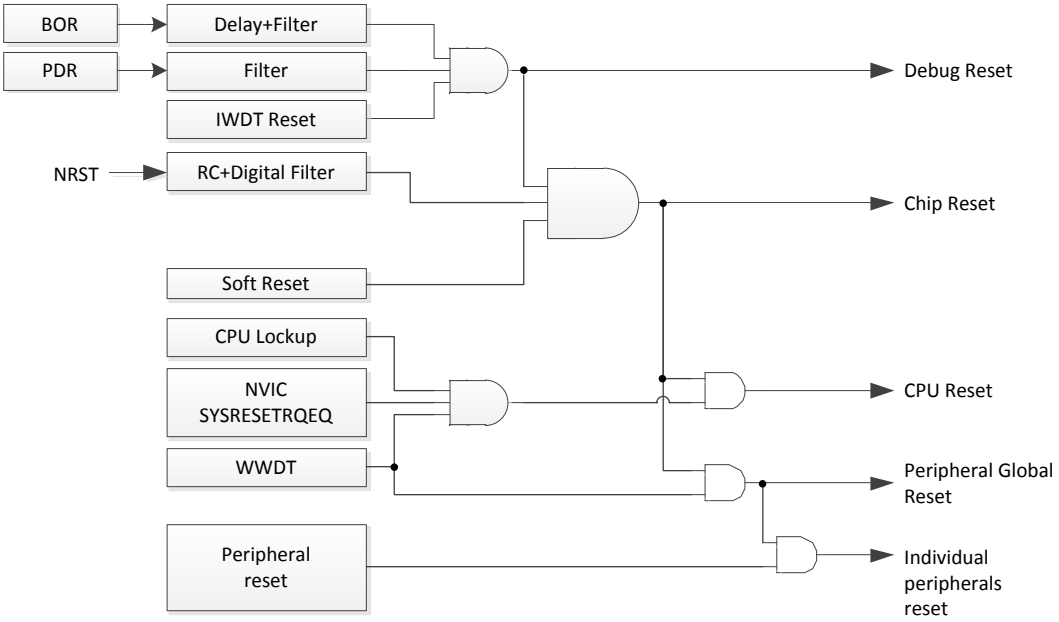


图 6-1 芯片复位源框图

6.3 上下电复位（BOR+PDR）

上下电复位电路监控 VDD 电源，由 BOR 和低功耗 PDR 组成。BOR 复位阈值电压精确度高，但是工作电流较大，因此在不需要精确下电复位阈值的场合，推荐关闭 BOR，仅保留 PDR。

VDD 电源上电期间上电复位信号有效，当 VDD 电压超过 V_{BOR} 时，上电复位放开；VDD 跌落到 V_{PDR} 时下电复位有效。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_{BOR} 上电阈值固定为 1.8V，BOR 下电复位阈值和低功耗 PDR 下电复位阈值软件可设置。

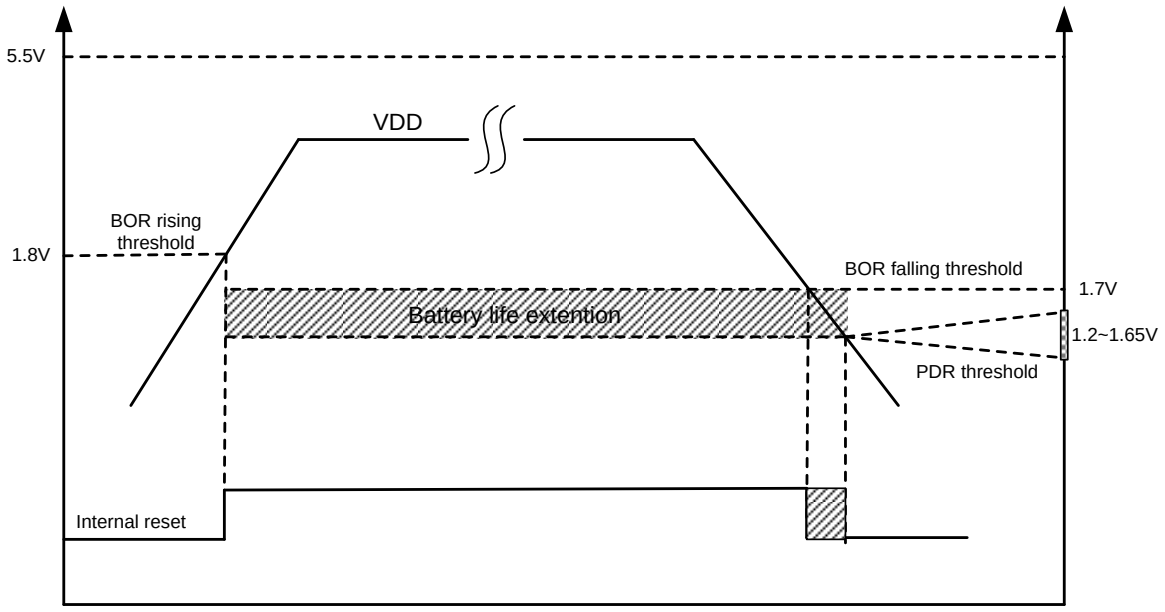


图 6-2 上下电复位示意图

6.4 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。

6.5 NRST 引脚复位

NRST 是芯片专用复位引脚，NRST 保持低电平超过 8ms 后，芯片将进入系统复位，但是并不会复位 DEBUG 逻辑。如果芯片处于低功耗模式，NRST 有效也会使芯片复位退出低功耗模式。

6.6 寄存器

模块起始地址：0x4000_2800

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 （Reset Config Register）	RMU_RSTCFGR
0x0C	软件复位寄存器 （Soft Reset Register）	RMU_SOFTRST
0x10	复位标志寄存器 （Reset Status Register）	RMU_RSR
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 Peripheral Reset Register）	RMU_AHBRST
0x1C	APB 外设复位寄存器 1 （APB Peripheral Reset Register1）	RMU_APBRST1
0x20	APB 外设复位寄存器 2 （APB Peripheral Reset Register2）	RMU_APBRST2

6.6.1 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	-					PDRCFG		PDREN
位权限	U-0					R/W-11		R/W-1

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2:1	PDRCFG	下电复位电压配置 00：1.5V 01：1.25V 10：1.35V

位号	助记符	功能描述
		11：1.4V（默认）
0	**PDREN**	下电复位使能 0：关闭下电复位 1：使能下电复位

6.6.2 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		OFF_BOR
位权限	U-0					RW-01		R/W-0

位号	助记符	功能描述
31:3	**-**	RFU：未实现，读为 0
2:1	**BOR_PDRCFG**	下电复位电压配置 00：1.7V 01：1.6V（默认） 10：1.65V 11：1.75V
0	**OFF_BOR**	BOR 使能控制寄存器 0：使能 BOR 1：关闭 BOR

6.6.3 复位配置寄存器（RMU_RSTCFGR）

名称	RMU_RSTCFGR							
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_RSTCFGR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						LKUPRST_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	LKUPRST_EN	LOCKUP 复位使能 1：使能 SC000 LOCKUP 复位 0：屏蔽 SC000 LOCKUP 复位
0	-	RFU：未实现，读为 0

6.6.4 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[31:24]							
位权限	W							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SOFTRST[23:16]							
位权限	W							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SOFTRST[15:8]							
位权限	W							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SOFTRST[7:0]							
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

6.6.5 复位标志寄存器（RMU_RSR）

名称	RMU_RSR							
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	-				NRSTN_FLAG	TESTN_FLAG	PORN_FLAG	PDRN_FLAG

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	-		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
31:12	-	RFU：未实现，读为 0
11	**NRSTN_FLAG**	NRST 引脚复位标志，高有效，写 1 清零
10	**TESTN_FLAG**	TESTN 引脚复位标志，高有效，写 1 清零
9	**PORN_FLAG**	上电复位标志，高有效，写 1 清零
8	**PDRN_FLAG**	下电复位标志，高有效，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，写 1 清零
4	**IWDTN_FLAG**	IWDT 复位标志，高有效，写 1 清零
3	-	RFU：未实现，读为 0
2	**WWDTN_FLAG**	WWDT 复位标志，高有效，写 1 清零
1	**LKUPN_FLAG**	LOOKUP 复位标志，高有效，写 1 清零
0	**NVICN_FLAG**	NVIC 复位标志，高有效，写 1 清零

6.6.6 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

6.6.7 AHB 外设复位寄存器（RMU_AHBRST）

名称	RMU_AHBRST							
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	-					HASHRST	PAERST	DMARST
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	HASHRST	HASH 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
1	PAERST	PAE 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
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

6.6.8 APB 外设复位寄存器 1（RMU_APBRST1）

名称	RMU_APBRST1							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	UART5RST	UART4RST	UART3RST	UART2RST	-			
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TIMARST	-						LCDRST
位权限	R/W-0	U-0						R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	U7816RST	-	SPI4RST	SPI3RST	SPI2RST	-	
位权限	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	LPUART0RST	-		I2C1RST	-		LPTRST
位权限	U-0	R/W-0	U-0		R/W-0	U-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

位号	助记符	功能描述
30	UART4RST	UART4 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
29	UART3RST	UART3 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
28	UART2RST	UART2 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
27:24	-	RFU：未实现，读为 0
23	TIMARST	Timer array 复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
22:17	-	RFU：未实现，读为 0
16	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
15	-	RFU：未实现，读为 0
14	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
13	-	RFU：未实现，读为 0
12	SPI4RST	SPI3 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
11	SPI3RST	SPI2 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
10	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
9:7	-	RFU：未实现，读为 0
6	LPUART0RST	EUART0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
5:4	-	RFU：未实现，读为 0
3	I2C1RST	I2C1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
2:1	-	RFU：未实现，读为 0
0	LPTRST	LPTIM 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

6.6.9 APB 外设复位寄存器 2（RMU_APBRST2）

名称	RMU_APBRST2							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	UART1RST	UART0RST	UARTIRRST	BSTRST	-			CICRST
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	-					AESRST	CRCRST	RNGRST
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	-						SPI1RST	SPI0RST
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LPUART1RST	-		I2C0RST	-		SVDRST	COMPRST
位权限	R/W-0	U-0		R/W-0	U-0		R/W-0	R/W-0

位号	助记符	功能描述
31	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
30	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
29	UARTIRRST	UART 红外模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
28	BSTRST	BSTIM 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
27:25	-	RFU：未实现，读为 0
24	CICRST	CIC 滤波模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
23:19	--	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
15:10	-	RFU：未实现，读为 0
9	SPI1RST	SPI1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位

位号	助记符	功能描述
		1：复位
8	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
7	LPUART1RST	EUART1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
6:5	-	RFU：未实现，读为 0
4	I2C0RST	I2C0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
3:2	-	RFU：未实现，读为 0
1	SVDRST	SVD 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
0	COMPRST	比较器模块复位 ，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

7 独立看门狗（IWDT）

7.1 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自动启动，且无法关闭，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

IWDT 核心是一个 20bit 向上计数器，复位后从 0 开始递增，计数到 0xFFF 后触发 IWDT 复位。IWDT 复位是一个全局复位，效果等同于上下电复位。

IWDT 使用 LSCLK 工作，结合停振检测电路，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 20bit。

溢出周期可配置为 125ms、500ms、2s、8s、4096s，其中 4096s 档位仅在休眠模式下可以使用。芯片一旦从休眠模式唤醒则自动切换回 4 个正常周期之一，并且唤醒完成后硬件会自动清狗，重新开始计数。

7.2 结构框图

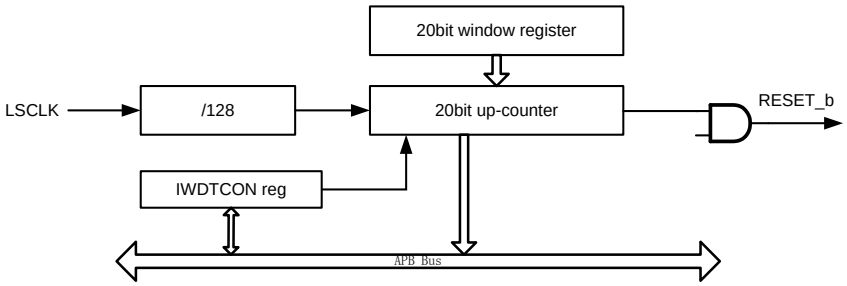


图 7-1 IWDT 结构框图

7.3 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 IWDT_CFGR 寄存器，选择合适的溢出周期
- 读 IWDT_CFGR，确保写入正确
- 溢出周期更新完毕，CPU 正常运行

7.4 寄存器

模块起始地址：0x4001_1400

offset 地址	名称	符号
IWDT(模块起始地址: 0x40011400)		
0x00	IWDT 清除寄存器 (IWDT Serve Register)	IWDT_SERV
0x04	IWDT 配置寄存器 (IWDT Config Register)	IWDT_CFGR
0x08	IWDT 计数值寄存器 (IWDT Counter Register)	IWDT_CNTR

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

位号	助记符	功能描述
31:0	**IWDTSERV**	IWDT 启动后，软件向此地址写入 0x1234_5A5A 时清狗

7.4.2 IWDT 配置寄存器（IWDT_CFGR）

名称	IWDT_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	-					IWDTOVP		
位权限	U-0					R/W-001		

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2:0	IWDTOVP	IWDT 溢出周期设置 x00：125ms x01：500ms x10：2s x11：8s 说明：当 bit2 为 0 时，休眠后仍使用短周期，而 bit2 为 1 时，休眠后自动使用 4096s；非休眠状态下仅支持 125ms/500ms/2s/8s 四种周期。

7.4.3 IWDT 计数值寄存器（IWDT_CNTR）

名称	IWDT_CNTR							
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	-				IWDTCNT			
位权限	U-0				R-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IWDTCNT							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IWDTCNT							
位权限	R-0000 0000							

位号	助记符	功能描述
31:20	-	RFU：未实现，读为 0
19:0	IWDTCNT	IWDT 当前计数值

8 窗口看门狗（WWDT）复位

8.1 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 CPU 和外设，避免不可预计的后果。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。

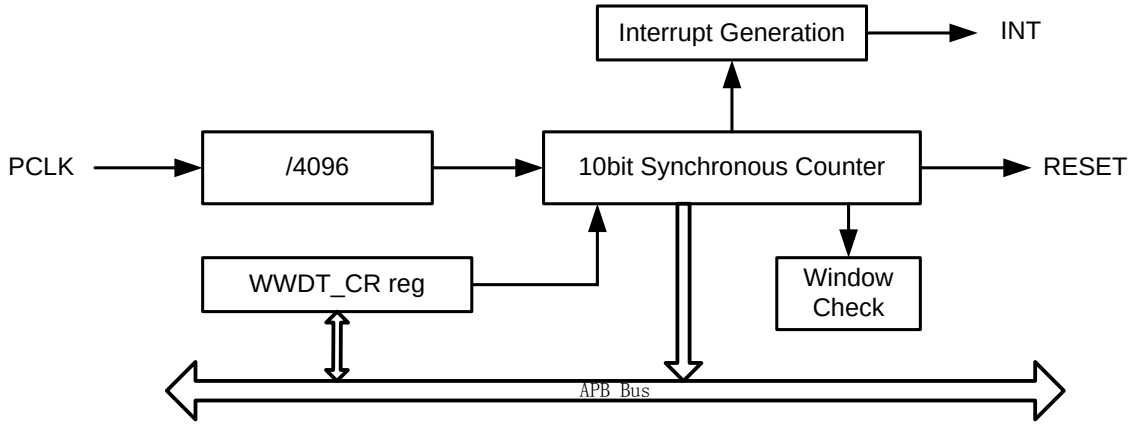


图 8-1 WWDT 结构框图

8.2 WWDT 工作方式

WWDT 在芯片复位后默认关闭，软件需对 WWDT_CR 寄存器写入 0x5A 来启动 WWDT。WWDT 启动后，如果软件在 open window 内对 WWDT_CR 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	65.536
	1024	131.072
16MHz	1	0.256
	4	1.024
	16	4.096
	64	16.384
	128	32.768
	256	65.536
	512	131.072
	1024	262.144
8MHz	1	0.512
	4	2.048
	16	8.192
	64	32.768
	128	65.536
	256	131.072
	512	262.144
	1024	524.288

表 8-1 WWDT 时钟频率和溢出时间

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。

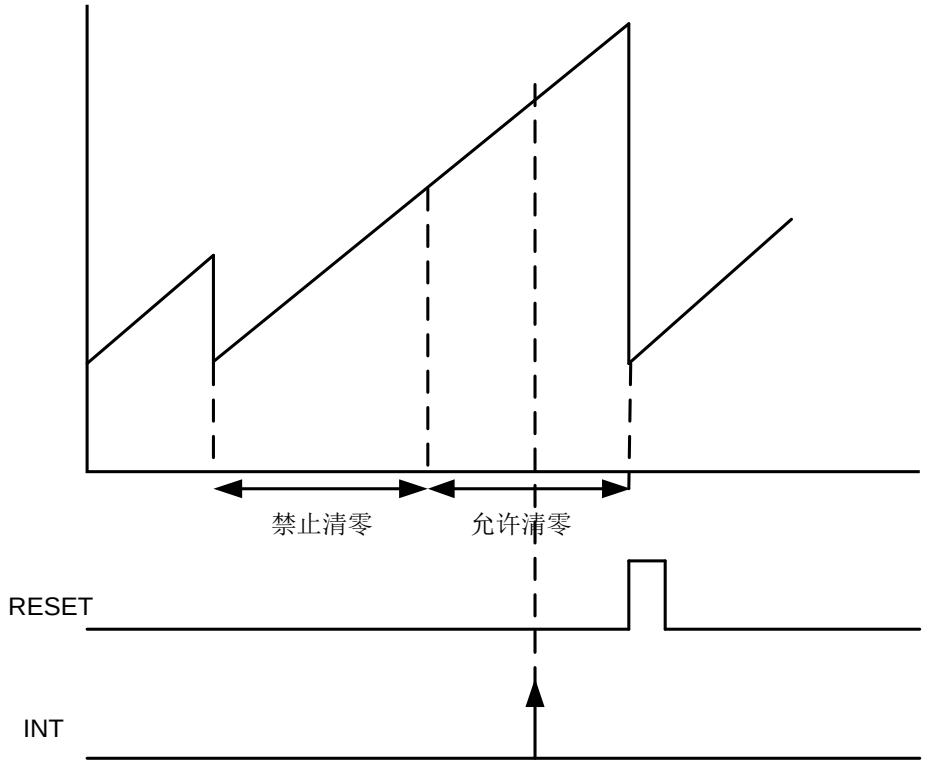


图 8-2 WWDT 窗口示意图

8.3 寄存器

模块起始地址：0x4001_1800

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_CNTR
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_PSCR

8.3.1 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

8.3.2 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

8.3.3 WWDT 计数寄存器（WWDT_CNTR）

名称	WWDT_CNTR							
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)

8.3.4 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

8.3.5 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

8.3.6 WWDT 预分频寄存器（WWDT_PSCR）

名称	WWDT_PSCR							
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	-				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)

9 时钟管理单元（CMU）

9.1 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（XTHF）、最高32MHz高频RC振荡器(RCHF)、32KHz低功耗内部环振（RCLP）、两个独立锁相环（PLL_L和PLL_H）。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
- XTLF 和 XTHF 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

9.2 时钟架构

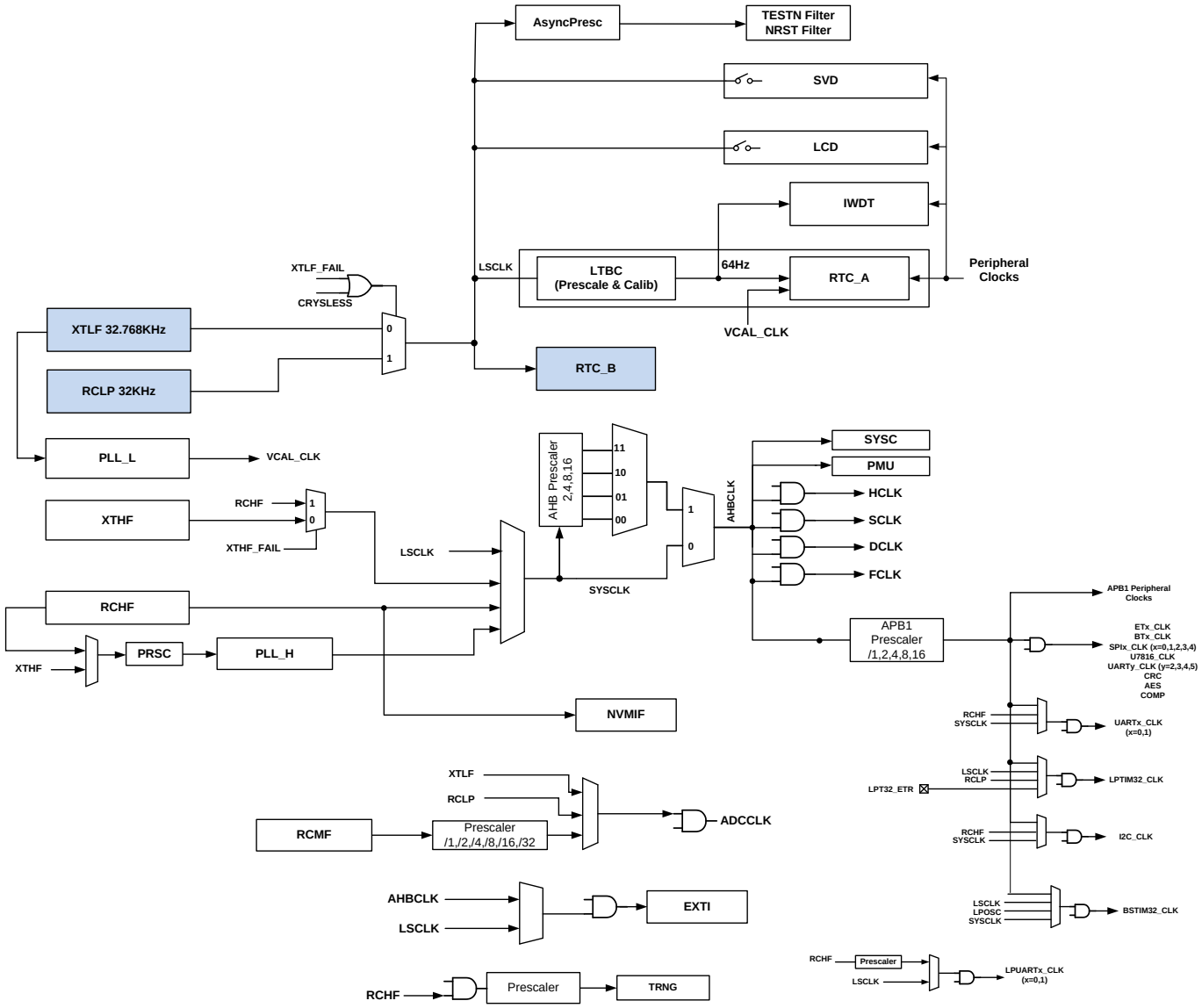


图 9-1 芯片时钟框图

系统主时钟(AHBCLK)可由LSCLK、RCHF、XTHF、PLL_H及它们的分频时钟产生。上电默认使用8MHz RCHF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。

注意：软件选择SYSCLK时钟源时，必须保证被选中的时钟源已经使能，如果目标时钟源没有被使能，则硬件禁止时钟切换并产生时钟错误中断标志。

9.2.1 主要时钟说明

时钟	源头	说明
LSCLK	XTLF, RCLP	32KHz 低频系统时钟，晶体停振时可以切换到 RCLP 主要用于 RTC、IWDT、引脚滤波、SVD、LCD
SYSCLK	RCHF, PLL_H, XTHF, LSCLK	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 9-1 主要工作时钟及其源头

9.2.2 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
LPTIM32	APBCLK	APBCLK
		LSCLK
		RCLP
		PLL_L
BSTIM32	APBCLK	APBCLK
		LSCLK
		RCLP
		SYSCLK
I2C	APBCLK	APBCLK
		RCHF
		SYSCLK
		APBCLK
∑△ADC	APBCLK	RCMF
		RCLP

模块	总线时钟	工作时钟
		XTLF
NVMIF (Flash erase/program)	AHBCLK	RCHF
EXTI (PADCFG)	AHBCLK	AHBCLK
		LSCLK
TRNG	APBCLK	RCHF
IWDT	APBCLK	LSCLK
LCD	APBCLK	LSCLK
非独立工作时钟外设		
PMU	AHBCLK	
DMA	AHBCLK	
BT/ET	APBCLK	
SPIx	APBCLK	
UARTy (y=2,3,4,5)	APBCLK	
7816	APBCLK	
AES	APBCLK	
PAE	AHBCLK	
HASH	AHBCLK	
CRC	APBCLK	
WWDT	APBCLK	

表 9-2 外设模块工作时钟和总线时钟

9.2.3 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LPUART、LPTIM32。

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。

9.3 高频 RC 振荡器(RCHF)

9.3.1 概述

高频RC振荡器典型振荡频率为8/16/24/32MHz，可用作系统主时钟，MCU在此频率下工作，可达到性能与功耗的平衡。为满足不同应用对MCU执行速度的需求，高频RC振荡器的输出频率可调，最高能达到32MHz。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%。

9.3.1 软件使用说明

芯片上电后默认使用RCHF 8MHz时钟工作，硬件电路会自动从LDT1读取8MHz校准值，保证8MHz常温频率误差小于+/-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RCHFCR.FSEL
- 从LDT0（0x1FFF_FB34、0x1FFF_FB38、0x1FFF_FB3C）读取频率调校值（分别对应32/24/16MHz）
- 将频率调校值写入RCHFTR寄存器，即可得到目标常温频率误差小于+/-0.5%的时钟

9.4 高频晶体振荡电路(XTHF)

9.4.1 概述

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

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

9.4.2 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PF1和PF2引脚配置为模拟功能。

9.4.3 停振检测（HFDET）

FM33A0xxEV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；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。

9.4.4 引脚

FM33A0xxEV复用PF1和PF2作为高频晶体引脚，使用外接高频晶振时，需将PF1和PF2配置成模拟功能。

9.5 锁相环(PLL_H)

9.5.1 概述

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

9.5.2 软件使用说明

出于可靠性考虑，软件需注意以下几点：

- 软件选择PLL_H输入时必须保证RCHF或XTHF为使能状态
- PLL_H输出选为SYSCLK时不能关闭PLL_H
- 软件应等待PLL_H锁定后（查询LOCKED寄存器）再将SYSCLK配置为PLL_H输出

如果配置PLL_H输出64Mhz，并作为系统时钟使用，则必须先将flash wait配置为2，再配置SYSCLK为PLL_H。

9.6 锁相环(PLL_L)

9.6.1 概述

锁相环输入参考时钟XTLF，输出频率16.384MHz，主要为RTC提供虚拟调校时钟。

频率输出满足关系：

$$F_{CKO} = \frac{F_{VCO}}{2} = F_{REFIN} \times (DB<9:0>+1)$$

其中FREFIN为32.768KHz，FCKO=16.384MHz，则DB<9:0>=499=111110011

9.6.2 软件使用说明

在开启锁相环之前，需保证XTLF正常起振，软件可以通过查询XTLF停振检测电路输出来确认XTLF振荡正常。通常情况下，上电后XTLF起振需要等待至少1s。开启锁相环后，软件还需要等待PLL_L锁定，在XTLF振荡正常的情况下，PLL_L典型锁定时间为2ms，建议等待4ms。

9.7 中频 RC 振荡器(RCMF)

9.7.1 概述

RCMF是一个中频低功耗环振，典型振荡频率为2Mhz；可以进行软件校准，校准步长0.5%，调校范围+/-30%。

RCMF主要用于ADC和温度传感器工作时钟，因此在使用ADC和温度传感器之前，必须使能RCMF。

RCMF控制寄存器参见CDIF章节。

9.8 低频晶体振荡电路(XTLF)

9.8.1 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。

XTLF默认上电后就开始运行，软件不可关闭。通过LDT1中的SYSCFG配置字，可以在无外接32768hz晶体的方案中关闭XTLF。

XTLF控制寄存器参见CDIF章节。

9.8.2 工作方式

XTLF上电后开始起振，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。

9.8.3 停振检测

芯片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，并自动将LSCLK切换到RCLP。

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。

当芯片配置为无晶体模式，LFDET 停止工作，输出固定为“不停振”。

9.9 低功耗 RC 振荡器(RCLP)

9.9.1 概述

RCLP是一个极低功耗的RC振荡器，主要用作XTLF的备份时钟，当XTLF意外停振时，RCLP可以保证RTC和IWDT不会停止。

RCLP典型频率为32Khz，可以软件校准，调校范围不小于±30%，典型调校步长小于0.35%。

RCLP控制寄存器参见CDIF章节。

9.9.2 软件使用说明

在没有配置为无晶体模式的情况下，当XTLF停振时RCLP被强制开启并将LSCLK自动切换为RCLP。当配置为无晶体模式时，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，NRST引脚滤波也无效，无法进行引脚复位；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	控制说明
Active/LPRun	软件控制
Sleep/DeepSleep	有 32768hz 晶体： RCLP_OFF 寄存器直接控制是否使能 XTLF 一旦停振，自动使能 RCLP 并切换 LSCLK 源
	无晶体模式： 软件控制 RCLP_OFF 决定是否使能 可以关闭
Sleep/DeepSleep 唤醒	软件控制

注意：NRST引脚输入滤波使用LSCLK。在休眠模式下，如果关闭了XTLF和RCLP， NRST就没有了滤波时钟，外部引脚复位就不起作用。

9.10 低功耗模式下的时钟源

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

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

9.11 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并恢复到休眠前的设置的PMU_CR.WKFSEL频率输出；同时将SYSCLKSEL寄存器复位成00，将系统时钟选为RCHF，AHBPRES寄存器被复位为0；因此芯片唤醒后默认将使用RCHF时钟工作。

9.12 寄存器

模块基地址：0x4000_2400

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_RCHFCR
0x08	RCHF 调校寄存器（RCHF Trim Register）	CMU_RCHFTR
0x0C	PLL_L 控制寄存器（PLL_L Control Register）	CMU_PLLLCR
0x10	PLL_H 控制寄存器（PLL_H Control Register）	CMU_PLLHCR
0x14	XTHF 控制寄存器（XTHF Control Register）	CMU_XTHFCR
0x18	时钟控制中断使能寄存器（CMU Interrupt Enable Register）	CMU_IER
0x1C	时钟控制中断标志寄存器（CMU Interrupt Status Register）	CMU_ISR
0x20	外设总线时钟控制寄存器 1（Peripheral bus Clock Control Register1）	CMU_PCLKCR1
0x24	外设总线时钟控制寄存器 2（Peripheral bus Clock Control Register2）	CMU_PCLKCR2
0x28	外设总线时钟控制寄存器 3（Peripheral bus Clock Control Register3）	CMU_PCLKCR3
0x2C	外设总线时钟控制寄存器 4（Peripheral bus Clock Control Register4）	CMU_PCLKCR4
0x30	外设工作时钟控制寄存器 1（ Peripheral Operation Clock Control Register1）	CMU_OPCCR1
0x34	外设工作时钟控制寄存器 2（ Peripheral Operation Clock Control Register2）	CMU_OPCCR2

9.12.1 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	-						SLP_EN EXTI	-
位权限	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	-					AHBPRES		

位权限	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
31:26	-	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 总线时钟分频选择 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
10:8	AHBPRES	AHB 时钟分频选择 0xx：不分频 100：2 分频 101：4 分频 110：8 分频 111：16 分频
7:6	STCLKSEL	CPU 内核 systick 工作时钟选择 00：SCLK 01：LSCLK 10：RFU 11：SYSCLK
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_H 011：LSCLK Others：RCHF

9.12.2 RCHF 时钟控制寄存器（CMU_RCHFCR）

名称	CMU_RCHFCR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				FSEL			
位权限	U-0				R/W-0000			

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

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 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

9.12.3 RCHF 调校寄存器（CMU_RCHFTR）

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

9.12.4 PLL_L 控制寄存器（CMU_PLLLCR）

名称	CMU_PLLLCR

offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						PLLDB[9:8]	
位权限	U-0						R/W-01	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PLLDB[7:0]							
位权限	R/W-11110011							
位	Bit15	Bit14	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	-						PLLEN
位权限	R-0	U-0						R/W-0

位号	助记符	功能描述
31:26	-	RFU：未实现，读为 0
25:16	PLLDB	PLL 倍频比 输入为 32768Hz，输出为 16.384MHz，则 PLLDB=0111110011
15:8	-	RFU：未实现，读为 0
7	LOCKED	PLL_L 锁定标志 1：PLL_L 已锁定 0：PLL_L 未锁定
6:1	-	RFU：未实现，读为 0
0	PLLEN	PLL 使能寄存器 1：使能 PLL 0：关闭 PLL 【注】当 sysclk 选择 PLL 作为时钟源时，无法关闭 PLL

9.12.5 PLL_H 控制寄存器（CMU_PLLHCR）

名称	CMU_PLLHCR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						PLLHDB[25:24]	
位权限	U-0						R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PLLHDB[23:16]							
位权限	R/W-0001 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCKED	REFPRSC[6:4]			PLLH_OSEL	-	PLLH_INSEL	PLLH_EN
位权限	R-0	R/W-000			R/W-0	U-0	R/W-0	R/W-0

位号	助记符	功能描述
31:26	-	RFU：未实现，读为 0

位号	助记符	功能描述
25:16	PLLHDB	PLLH 倍频比，参考时钟 1Mhz 0000011111：输出 32 倍频 0111111：输出 64 倍频 PLL 输出范围为 32MHz~64MHz
15:8	-	RFU：未实现，读为 0
7	LOCKED	PLLH 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：PLLH 已锁定 0：PLLH 未锁定
6:4	REFPRSC	PLLH 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	PLLH_OSEL	PLLH 输出选择寄存器 0：选择 PLLH 一倍输出作为数字电路内的 PLL 时钟 1：选择 PLLH 两倍输出作为数字电路内的 PLL 时钟
2	-	RFU：未实现，读为 0
1	PLLH_INSEL	PLLH 输入选择寄存器 0：RCHF 1：XTHF
0	PLLH_EN	PLLH 使能寄存器 1：使能 PLLH 0：关闭 PLLH

9.12.6 XTHF 控制寄存器（CMU_XTHFCR）

名称	CMU_XTHFCR							
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	-					XTHF_CFG		
位权限	U-0					R/W-000		
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
31:11	-	RFU：未实现，读为 0

位号	助记符	功能描述
10:8	XTHF_CFG	XTHF 振荡强度配置 000：最弱 111：最强
7:1	-	RFU：未实现，读为 0
0	XTHFEN	XTHF 使能寄存器 0：关闭 XTHF 1：使能 XTHF

9.12.7 时钟控制中断使能寄存器（CMU_IER）

名称	CMU_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	-						SYSCSE_IE	HFDET_IE
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	SYSCSE_IE	SYSCLK 选择错误中断使能，1 有效
0	HFDET_IE	XTHF 高频检测报警中断使能，1 有效

9.12.8 时钟控制中断标志寄存器（CMU_ISR）

名称	CMU_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	-							HFDETO
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						SYSCSE_IF	HFDET_IF

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	HFDETO	高频晶体停振检测模块输出 1：XTHF 未停振 0：XTHF 停振
7:2	-	RFU：未实现，读为 0
1	SYSCSE_IF	SYSCLK 时钟选择错误中断标志寄存器；当被选中的时钟源没有使能时，此中断标志置位，软件写 1 清零。
0	HFDET_IF	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

9.12.9 外设总线时钟控制寄存器 1（CMU_PCLKCR1）

名称	CMU_PCLKCR1							
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	-						COMP_PCE	SVD_PCE
位权限	U-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	RTC_PCE	-	LPT_PCE
位权限	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	U-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9	COMP_PCE	COMP 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	ANAC 总线时钟使能，高使能（仅用于测试功能）
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	RTC_PCE	RTC 总线时钟使能，高使能
1	-	RFU：未实现，读为 0
0	LPT_PCE	LPTIM32 总线时钟使能，高使能

9.12.10 外设总线时钟控制寄存器 2（CMU_PCLKCR2）

名称	CMU_PCLKCR2

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	-						PAE_PCE	SHA_PCE
位权限	U-0						R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							CIC_PCE
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WWDT_PCE	RAMBIST_PCE	NVM_PCE	DMA_PCE	LCD_PCE	AES_PCE	TRNG_PCE	CRC_PCE
位权限	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:18	-	RFU：未实现，读为 0
17	PAE_PCE	PAE 总线时钟使能，高使能
16	SHA_PCE	SHA 总线时钟使能，高使能
15:9	-	RFU：未实现，读为 0
8	CIC_PCE	CIC 总线时钟使能，高使能
7	WWDT_PCE	WWDT 总线时钟使能，高使能
6	RAMBIST_PCE	RAMBIST 总线时钟使能，高使能
5	NVM_PCE	NVMIF（Flash 擦写控制器）总线时钟使能，高使能
4	DMA_PCE	DMA 总线时钟使能，高使能
3	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

9.12.11 外设总线时钟控制寄存器 3（CMU_PCLKCR3）

名称	CMU_PCLKCR3							
offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						I2C1_PCE	I2C0_PCE
位权限	U-0						R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					LPUART1_PCE	-	U7816_PCE
位权限	U-0					R/W-0	U-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LPUART0_PCE	UICR_PCE	UART5_PCE	UART4_PCE	UART3_PCE	UART2_PCE	UART1_PCE	UART0_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	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	QSPI_PCE	-	SPI4_PCE	SPI3_PCE	SPI2_PCE	SPI1_PCE	SPI0_PCE
位权限	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
31:26	-	RFU：未实现，读为 0
25	I2C1_PCE	I2C1 总线时钟使能，高有效
24	I2C0_PCE	I2C0 总线时钟使能，高有效
23:19	-	RFU：未实现，读为 0
18	LPUART1_PCE	LPUART1 总线时钟使能，高有效
17	-	RFU：未实现，读为 0
16	U7816_PCE	7816 总线时钟使能，高有效
15	LPUART0_PCE	LPUART0 总线时钟使能，高有效
14	UICR_PCE	UART 红外调制总线使能，高有效
13	UART5_PCE	UART5 总线时钟使能，高有效
12	UART4_PCE	UART4 总线时钟使能，高有效
11	UART3_PCE	UART3 总线时钟使能，高有效
10	UART2_PCE	UART2 总线时钟使能，高有效
9	UART1_PCE	UART1 总线时钟使能，高有效
8	UART0_PCE	UART0 总线时钟使能，高有效
7	QSPI_PCE	QuadSPI 总线时钟使能，高有效
6:5	-	RFU：未实现，读为 0
4	SPI4_PCE	SPI4 总线时钟使能，高有效
3	SPI3_PCE	SPI3 总线时钟使能，高有效
2	SPI2_PCE	SPI2 总线时钟使能，高有效
1	SPI1_PCE	SPI1 总线时钟使能，高有效
0	SPI0_PCE	SPI0 总线时钟使能，高有效

9.12.12 外设总线时钟控制寄存器 4（CMU_PCLKCR4）

名称	CMU_PCLKCR4							
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	-	ET4CKEN	ET3CKEN	ET2CKEN	ET1CKEN	BT2CKEN	BT1CKEN	BSTIM_PCE
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6	ET4CKEN	Etimer4 时钟使能，高使能
5	ET3CKEN	Etimer3 时钟使能，高使能
4	ET2CKEN	Etimer2 时钟使能，高使能
3	ET1CKEN	Etimer1 时钟使能，高使能
2	BT2CKEN	BaseTimer2 时钟使能，高使能
1	BT1CKEN	BaseTimer1 时钟使能，高使能
0	BSTIM_PCE	基本定时器 BSTIM32 总线时钟使能，高使能

9.12.13 外设独立工作时钟控制寄存器 1（CMU_OPCCR1）

名称	CMU_OPCCR1							
offset	0x30							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTICKE	EXTICKSEL	LPUART1CKE	LPUART0CKE	LPUART1CKS		LPUART0CKS	
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		I2C1CKE	I2C0CKE	I2C1CKS		I2C0CKS	
位权限	U-0		R/W-0	R/W-0	R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						UART1CKE	UART0CKE
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	-			UART1CKS		UART0CKS	
位权限	U-0	U-0			R/W-00		R/W-00	

位号	助记符	功能描述
31	EXTICKE	EXTI 工作时钟使能，高有效
30	EXTICKSEL	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 HCLK 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	LPUART1CKE	LPUART1 工作时钟使能，高有效
28	LPUART0CKE	LPUART0 工作时钟使能，高有效
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：RFU 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：RFU 11：RFU

位号	助记符	功能描述
23:22	-	RFU：未实现，读为 0
21	I2C1CKE	I2C1 工作时钟使能
20	I2C0CKE	I2C0 工作时钟使能
19:18	I2C1CKS	I2C1 主机工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：APBCLK
17:16	I2C0CKS	I2C 主机工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：APBCLK
15:10	-	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
3:2	UART1CKS	UART1 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：RFU
1:0	UART0CKS	UART0 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：RFU

9.12.14 外设独立工作时钟控制寄存器 2（CMU_OPCCR2）

名称	CMU_OPCCR2							
offset	0x34							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	RNGPRSC			-			
位权限	U-0	R/W-000			U-0			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-	NVMCKE	RNGCKE	-				
位权限	U-0	R/W-0	R/W-0	U-0				
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			LPTCKE	-		LPTCKS	
位权限	U-0			R/W-0	U-0		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			BSTCKE	-		BSTCKS	
位权限	U-0			R/W-0	U-0		R/W-00	

位号	助记符	功能描述

位号	助记符	功能描述
31	-	RFU：未实现，读为 0
30:28	RNGPRSC	随机数发生器工作时钟分频 000：不分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110, 111：RFU
27:23	-	RFU：未实现，读为 0
22	NVMCKE	Flash 擦写时钟使能，高有效
21	RNGCKE	随机数发生器工作时钟使能，高有效
20:13	-	RFU：未实现，读为 0
12	LPTCKE	LPTIM 工作时钟使能，高有效
11:10	-	RFU：未实现，读为 0
9:8	LPTCKS	LPTIM 工作时钟选择 00：APBCLK 01：LSCLK 10：RCLP 11：PLL_L
7:5	-	RFU：未实现，读为 0
4	BSTCKE	BSTIM 工作时钟使能，高有效
3:2	-	RFU：未实现，读为 0
1:0	BSTCKS	BSTIM 工作时钟源选择 00：APBCLK 01：LSCLK 10：RCLP 11：SYSCLK

10 其他模拟控制寄存器（CDIF）

10.1 概述

以下模块相关寄存器采用独立总线编址，逻辑地址上全部属于CDIF模块。

- 512bytes RAM
- ADC
- RCMF
- XTLF
- RCLP

CDIF是一个特殊的控制接口，通过这个接口软件可以配置RCMF、RCLP、ADC、XTLF相关的寄存器。

10.2 访问方式

CDIF 仅支持 CPU 发起 word 访问，不支持 DMA 访问。

访问CDIF地址段下的模拟控制寄存器时，必须先置位CDIF_CR.INTF_EN寄存器，并在访问结束后清零这个寄存器。如果不清零INTF_EN，会额外引入十几uA的功耗。

10.3 CDIF 寄存器

CDIF控制模块起始地址：0x4001_E000

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

10.3.1 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_EN			
位权限	U-0				R/W-0101			

位号	助记符	功能描述
31:4	-	未实现，读为 0
3:0	**INTF_EN**	跨电源域接口使能 1010：使能接口 其他：关闭接口

10.3.2 接口预分频寄存器（CDIF_PSCR）

名称	CDIF_PSCR							
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-0000		

位号	助记符	功能描述
31:4	-	未实现，读为 0
2:0	**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

10.4 其他模拟控制寄存器

模块起始地址：0x4001_F000

总线	地址边界	空间	外设
CDIF	0x4001_F000~0x4001_F3FF	1KB	-
	0x4001_F400~0x4001_F7FF	1KB	RAM
	0x4001_F800~0x4001_FBFF	1KB	模拟模块控制

ADC相关寄存器，请参见ADC章节。

其他模拟控制寄存器起始地址：0x4001_F800

offset 地址	名称	符号
VRTC(模块起始地址:0x4001F800)		
0x14	RCMF 控制寄存器 (RCMF Control Register)	VRTC_RCMFCR
0x18	RCLP 控制寄存器 (RCLP Control Register)	VRTC_RCLPCR
0x1C	RCLP 调校寄存器 (RCLP Control Register)	VRTC_RCLPTR
0x20	XTLF 控制寄存器 (XTLF Control Register)	VRTC_XTLFCR
0x24	ADC 时钟控制寄存器 (ADC Clock Control Register)	VRTC_ADCCR
0x28	LFDET 中断使能寄存器 (LFDET Interrupt Enable Register)	VRTC_LFDIER
0x2C	LFDET 中断标志寄存器 (LFDET Interrupt Status Register)	VRTC_LFDISR

10.4.1 RCMF 控制寄存器（VRTC_RCMFCR）

名称	VRTC_RCMFCR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EN	-	TRIM					
位权限	R/W-0	U-0	R/W-100000					

位号	助记符	功能描述
31:8	-	未实现，读为 0
7	EN	RCMF 使能寄存器 0：关闭 RCMF 1：使能 RCMF

位号	助记符	功能描述
		【注】当使能了 RTC 自动温补后，RCMF 受硬件控制自行启动和关闭
6	-	RFU：未实现，读为 0
5:0	**TRIM**	RCMF 频率调校寄存器

10.4.2 RCLP 控制寄存器（VRTC_RCLPCR）

名称	VRTC_RCLPCR							
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	-							RCLP_OFF
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	RCLP_OFF	RCLP OFF 使能信号 0：使能 RCLP 1：关闭 RCLP 【注】XTLF 异常停振时，自动使能 RCLP

10.4.3 RCLP 调校寄存器（VRTC_RCLPTR）

名称	VRTC_RCLPTR							
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	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

10.4.4 XTLF 控制寄存器（VRTC_XTLFCR）

名称	VRTC_XTLFCR							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				XTLFIPW			
位权限	U-0				R/W-1000			

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
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 1000 档位起振，起振后可以选用电流较小的档位以降低功耗，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

10.4.5 ADC 时钟控制寄存器（VRTC_ADCCR）

名称	VRTC_ADCCR							
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	-				CKS			CKE
位权限	U-0				R/W-010			R/W-0

位号	助记符	功能描述
31:4	-	RFU：未实现，读为 0
3:1	CKS	ADC 工作时钟选择寄存器 000：RCMF 001：RCMF/2 010：RCMF/4 011：RCMF/8 100：RCMF/16 101：RCMF/32 110：RCLP 111：XTLF
0	CKE	ADC 工作时钟使能 0：关闭 ADC 工作时钟 1：使能 ADC 工作时钟

10.4.6 低频检测中断使能寄存器（VRTC_LFDIER）

名称	VRTC_LFDIER							
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	-							LFDET_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	LFDET_IE	XTLF 低频检测报警中断使能，1 有效

10.4.7 低频检测中断标志寄存器（VRTC_LFDISR）

名称	VRTC_LFDISR							
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	-						LFDETO	LFDET_IF
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	低频停振检测状态输出 1：XTLF 正常 0：XTLF 停振
0	LFDET_IF	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

11 电源电压监测（SVD）

11.1 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
- 外部通道通过设置参考电压支持 100mV 窗口

11.2 模块框图

下图是电源检测电路的模块框图。

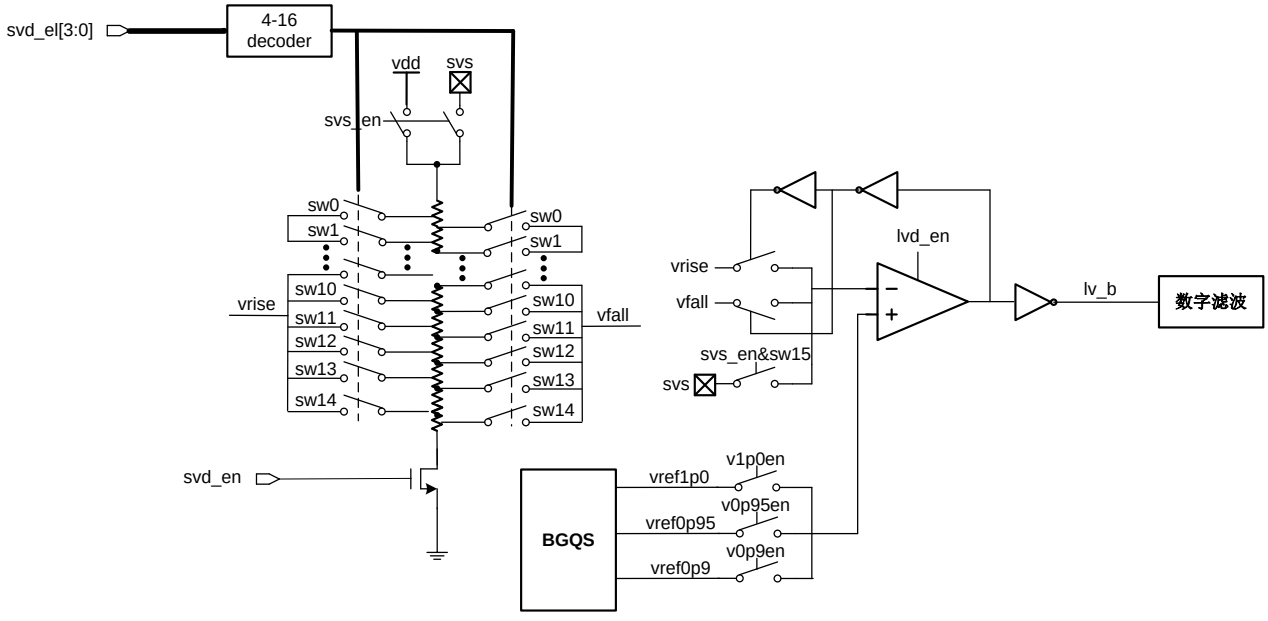


图 11-1 低压检测电路框图

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：

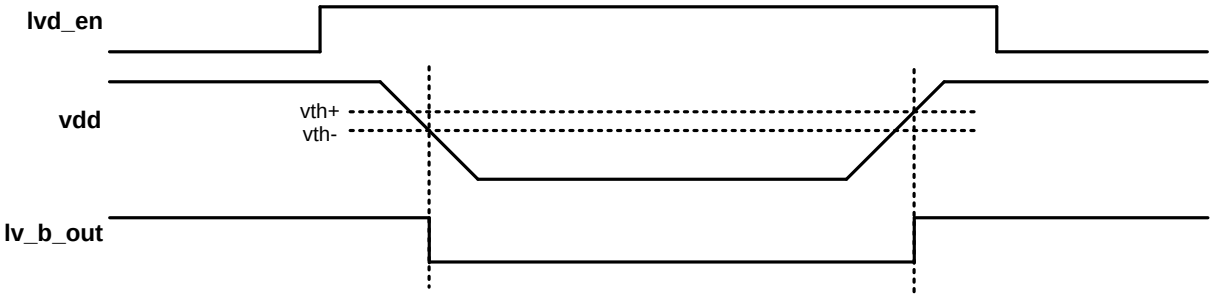


图 11-2 低压检测电路工作时序

11.3 功能描述

11.3.1 电源检测

电源检测电路可以用来检测主电源电压及外部电压（SVS引脚输入）。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1路外部输入检测电平，共16级检测电平。通过16选1 MUX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SVD_CFGR.ITVL寄存器设置开启时间间隔。

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

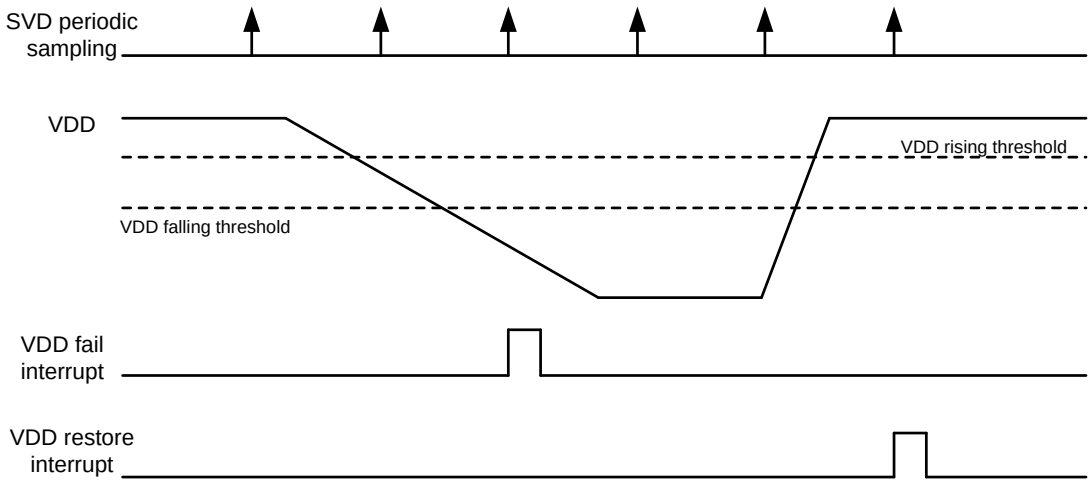


图 11-3 电源检测电路间歇工作模式

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。

阈值窗口：

- 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，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，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。

11.3.2 外部电压检测

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（PF11）实现，SVS的输入可以采用外部电阻分压后，再输入到比较器进行检测。使用SVS前需要将PF11引脚设置为模拟功能。

下图为外部电阻分压的外部电源检测：

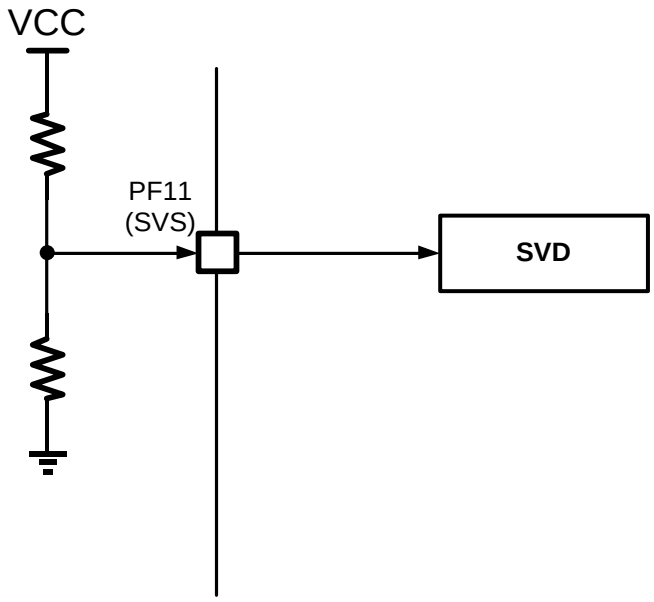


图 11-4 使用外部电阻分压的 SVS 检测

寄存器配置方法如下表：

SVSEN	SVDLVL	说明
0	1111	无检测功能，SVD 模块关闭
	0000~1110	内部电源检测，检测阈值参见 LVL 寄存器定义
1	1111	外部电压输入不做内部分压，直接输入到比较器与内部基准电压比较
	0000~1110	无检测功能

表 11-1 外部电压检测寄存器配置表

注意，使用SVS监视外部电压时，必须将PF11配置为模拟功能。

11.3.3 检测阈值

通过SVSEN和LVL寄存器可以选择电压检测对象和检测阈值。

内部电源检测：SVSEN = 0，{V1P0EN, V0P95EN, V0P9EN} = 100

SVDLVL	下降阈值（V）	上升阈值（V）
0000	1.799	1.912
0001	2.013	2.128
0010	2.227	2.343
0011	2.442	2.558
0100	2.656	2.774
0101	2.871	2.989
0110	3.085	3.204
0111	3.300	3.420
1000	3.515	3.635
1001	3.73	3.851

1010	3.945	4.064
1011	4.16	4.284
1100	4.376	4.500
1101	4.592	4.716
1110	4.807	4.932
1111	N/A(无效)	N/A(无效)

表 11-2 内部电源检测阈值（基准选择 100）

*内部电源检测：**SVSEN = 0，{V1P0EN, V0P95EN, V0P9EN } = 010***

SVDLVL	下降阈值（V）	上升阈值（V）
0000	1.704	1.812
0001	1.907	2.016
0010	2.110	2.220
0011	2.313	2.424
0100	2.516	2.628
0101	2.719	2.832
0110	2.923	3.036
0111	3.126	3.240
1000	3.329	3.444
1001	3.533	3.649
1010	3.737	3.853
1011	3.941	4.058
1100	4.146	4.263
1101	4.350	4.468
1110	4.554	4.672
1111	N/A(无效)	N/A(无效)

表 11-3 内部电源检测阈值（基准选择 010）

*内部电源检测：**SVSEN = 0，{V1P0EN, V0P95EN, V0P9EN } = 001***

SVDLVL	下降阈值（V）	上升阈值（V）
0000	1.609	1.712
0001	1.801	1.904
0010	1.992	2.097
0011	2.184	2.289
0100	2.376	2.482
0101	2.568	2.675
0110	2.760	2.867
0111	2.952	3.060
1000	3.144	3.253
1001	3.336	3.446
1010	3.529	3.639
1011	3.722	3.833
1100	3.915	4.026
1101	4.108	4.220

1110	4.300	4.413
1111	N/A(无效)	N/A(无效)

表 11-4 内部电源检测阈值（基准选择 001）

外部电压检测：SVSEN =1，{V1P0EN, V0P95EN, V0P9EN } = 100

SVDLVL	下降阈值（V）	上升阈值（V）
0000	N/A(无效)	N/A(无效)
0001		
0010		
0011		
0100		
0101		
0110		
0111		
1000		
1001		
1010		
1011		
1100		
1101		
1110		
1111	1.0	1.0

表 11-5 外部电源检测阈值（基准选择 100）

外部电压检测：SVSEN =1，{V1P0EN, V0P95EN, V0P9EN } = 010

SVDLVL	下降阈值（V）	上升阈值（V）
0000	N/A(无效)	N/A(无效)
0001		
0010		
0011		
0100		
0101		
0110		
0111		
1000		
1001		
1010		
1011		
1100		
1101		
1110		
1111	0.95	0.95

表 11-6 外部电源检测阈值（基准选择 010）

外部电压检测：SVSEN =1，{V1P0EN, V0P95EN, V0P9EN } = 001

SVDLVL	下降阈值（V）	上升阈值（V）
0000	N/A(无效)	N/A(无效)
0001		
0010		
0011		
0100		
0101		
0110		
0111		
1000		
1001		
1010		
1011		
1100		
1101		
1110		
1111	0.9	0.9

表 11-7 外部电源检测阈值（基准选择 001）

11.4 寄存器

模块基地址：0x4001_2800

offset 地址	名称	符号
SVD(模块起始地址: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

11.4.1 SVD 配置寄存器（SVD_CFGR）

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

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:4	LVL	SVD 报警阈值档位设置，参见 11.3.3 检测阈值 (SVD threshold level)
3	DFEN	数字滤波使能（MOD=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 SVD_CR.EN 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 间歇使能间隔 (SVD interval enable period)

位号	助记符	功能描述
		00：62.5ms 01：256ms 10：1s 11：4s

11.4.2 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	-							SVDTE
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						SVSEN	SVDEN
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	SVDTE	SVD 测试使能，避免写 1 (SVD test enable)
7:2	-	RFU：未实现，读为 0
1	SVSEN	SVS 外部电源检测通道控制信号 (SVS external monitor channel enable) 0：SVS 通道关闭 1：SVS 通道使能 当 EN=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 LVL=1111，则 SVS 输入不做分压，如果 LVL != 1111，则 SVS 输入经过内部电阻分压。
0	SVDEN	SVD 使能 (SVD enable) 1：启动 SVD 0：关闭 SVD

11.4.3 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，1 有效 (Power Fall interrupt enable)
0	PRIE	电源恢复中断使能寄存器，1 有效 (Power Rise interrupt enable)

11.4.4 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	R/W

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 (SVD output) 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 (SVD registered output)
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 (Power fall flag)
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 (Power rise flag)

11.4.5 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

12 AES 运算单元（AES）

12.1 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

12.2 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。

12.3 AES 数据流处理模式

AES有4种数据流处理模式：ECB，CBC，CTR，GCM。

12.3.1 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12-1和图12-2所示。

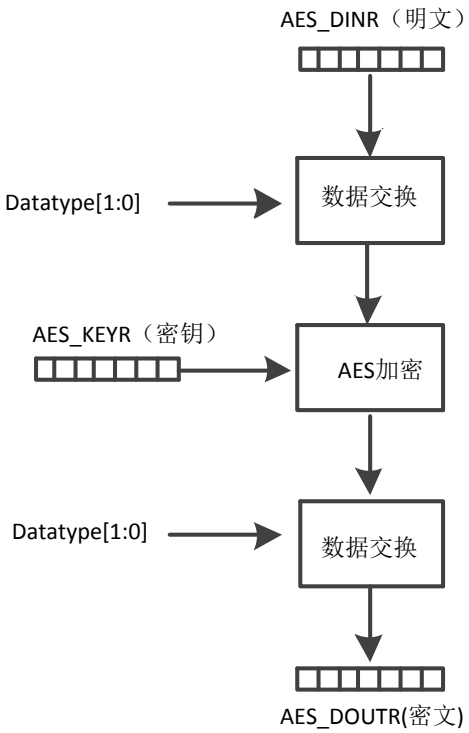


图 12-1 ECB 模式加密流程

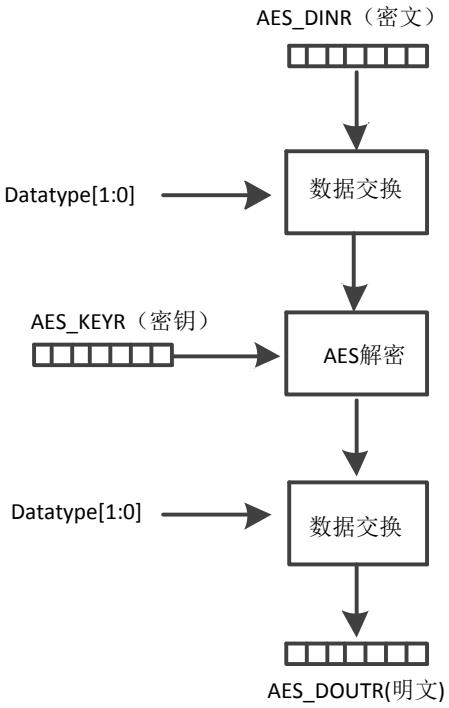


图 12-2 ECB 模式解密流程

12.3.2 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12-3和图12-4所示。

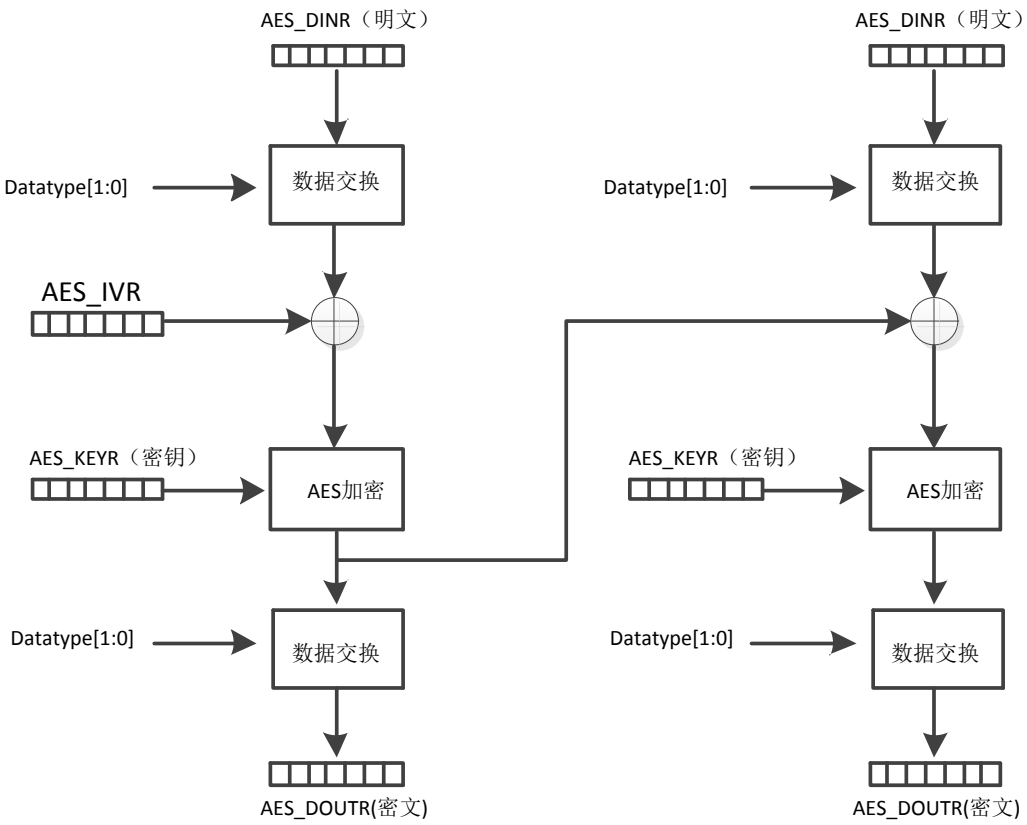


图 12-3 CBC 加密过程

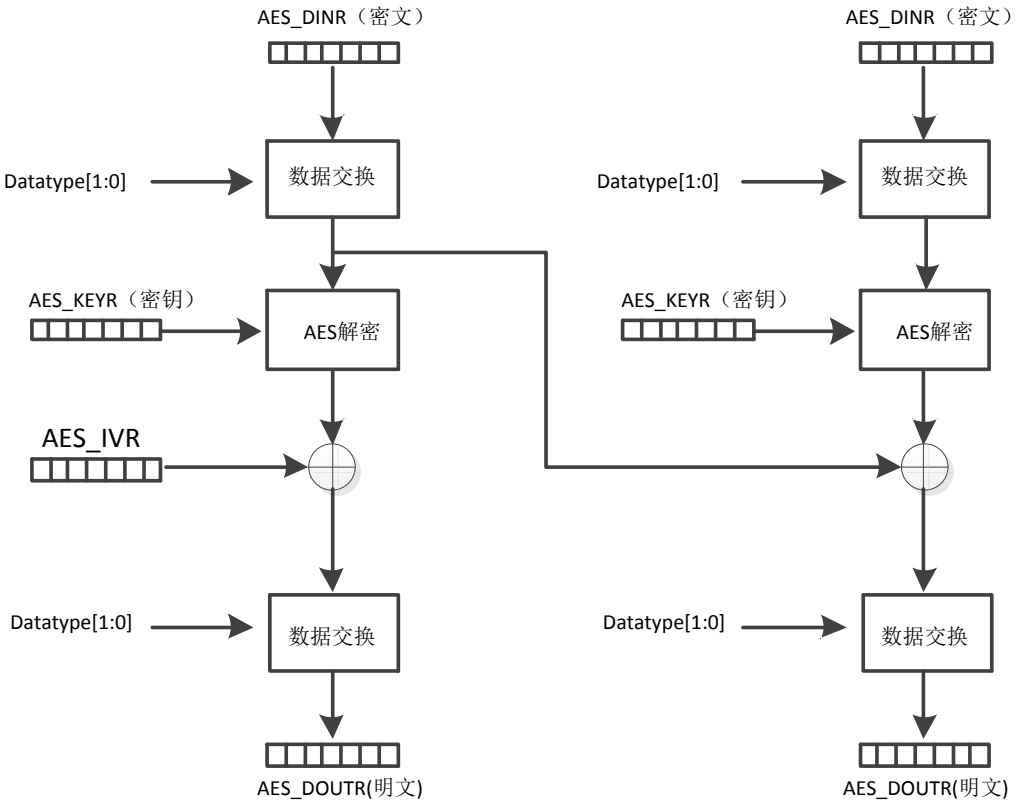


图 12-4 CBC 解密过程

注：在AES工作时读取AES_IVR寄存器的值为0x00000000

12.3.3 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12-5所示



图 12-5 暂停模式流程

12.3.4 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12-6和图12-7所示。

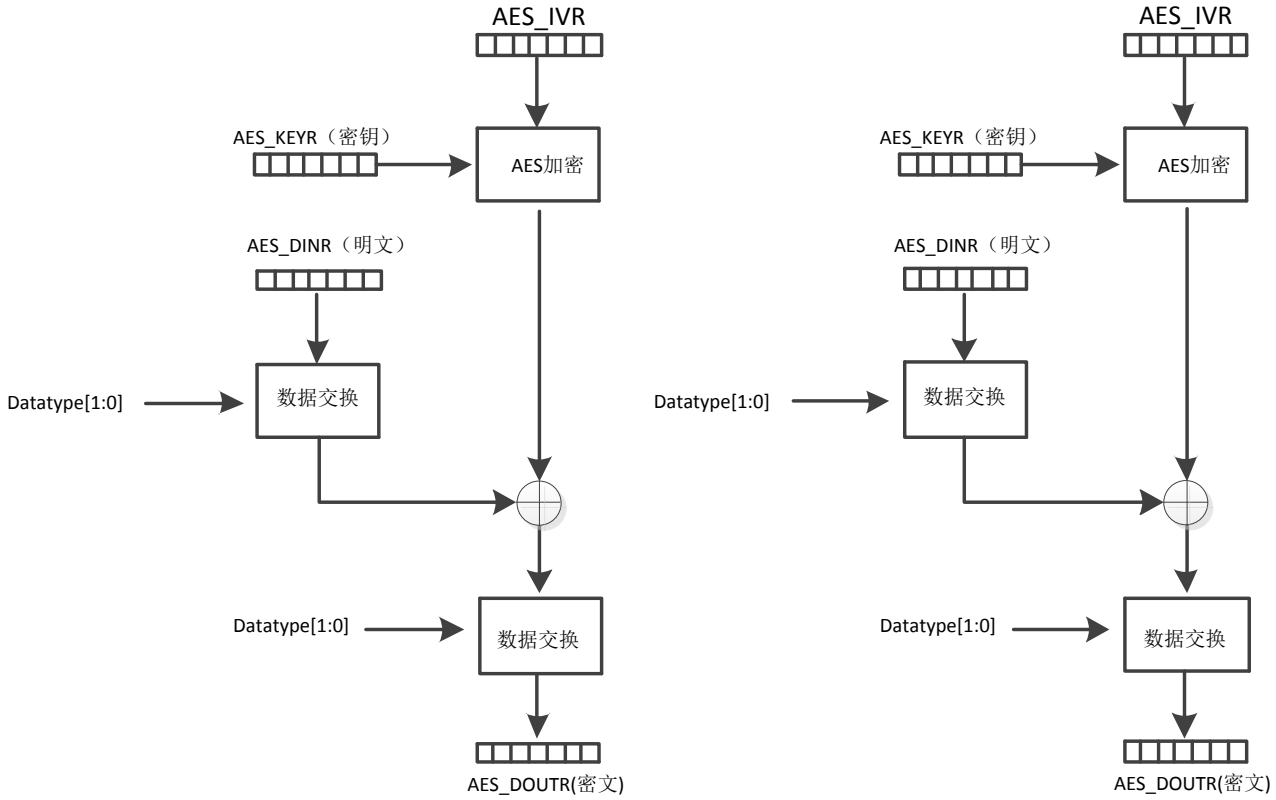


图 12-6 CTR 加密流程

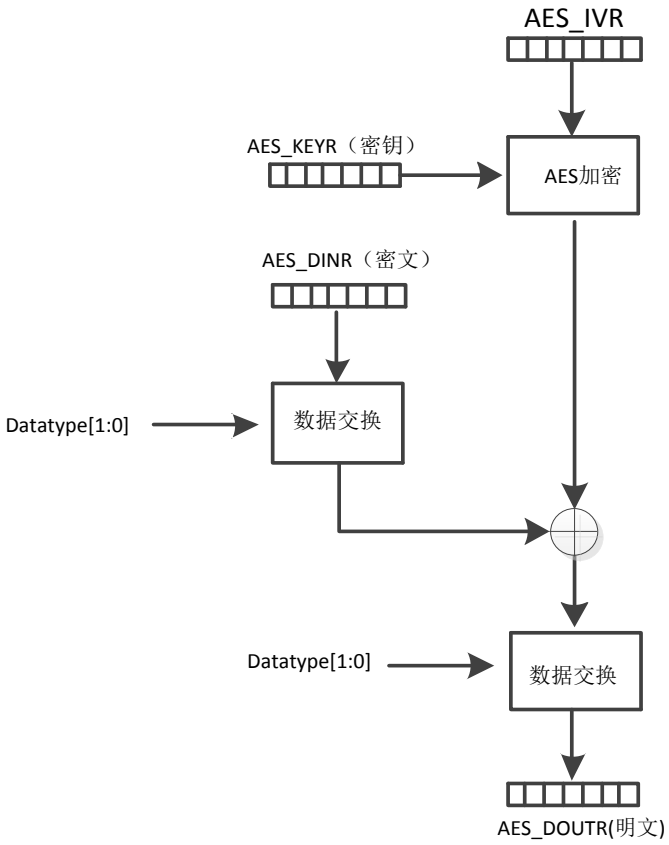


图 12-7 CTR 解密流程

随机数（nonce）和32位计数器存储在IV寄存器中，如图12-8所示

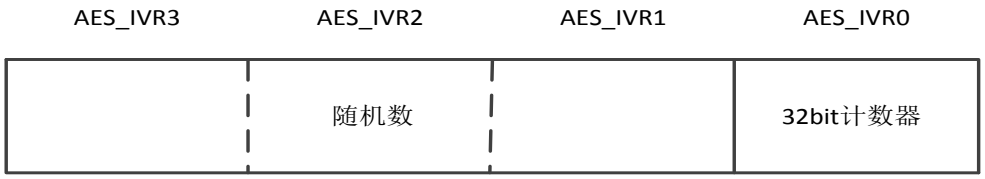


图 12-8 32 位计数器和随机数的存储方式

CTR模式下密钥扩展和解密模式没有意义。

12.3.5 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。

12.3.6 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：

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

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

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

GCM模式的加解密流程如图12-9，图12-10所示。

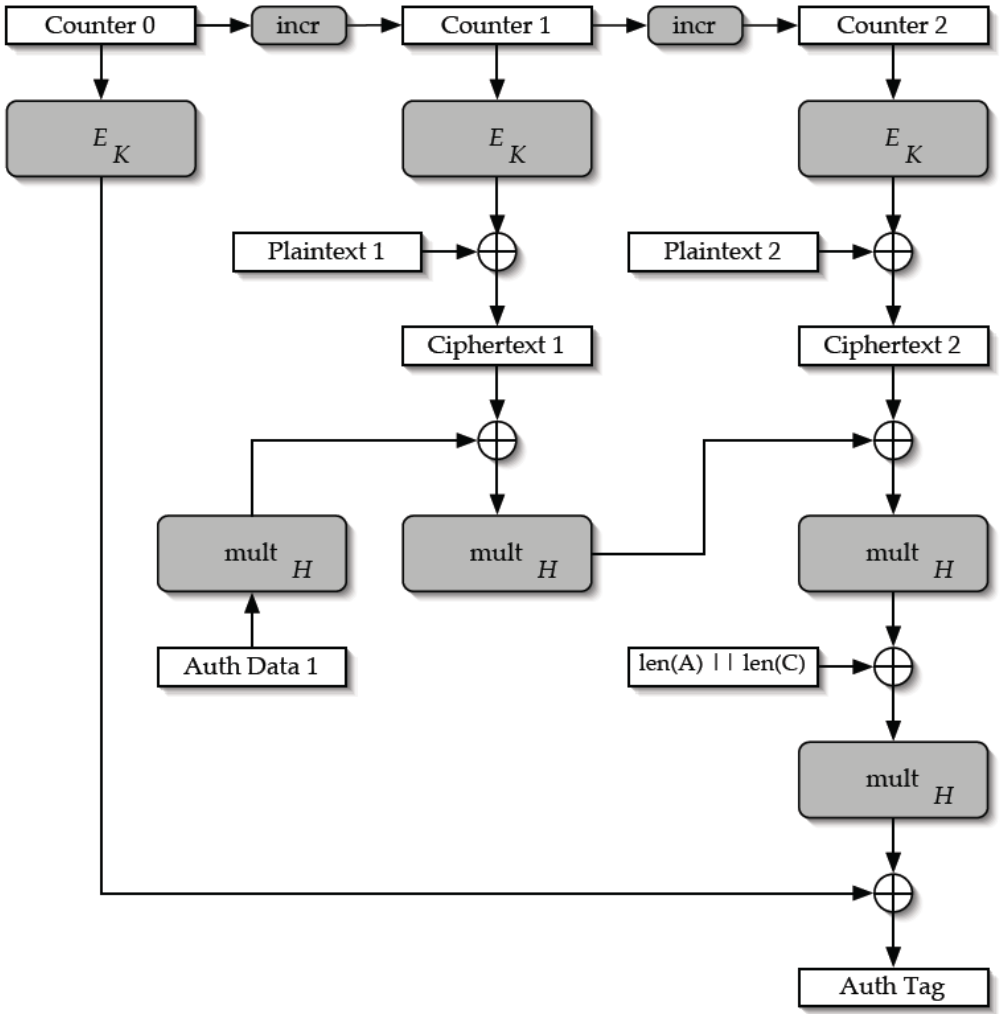


图 12-9 GCM 加密流程

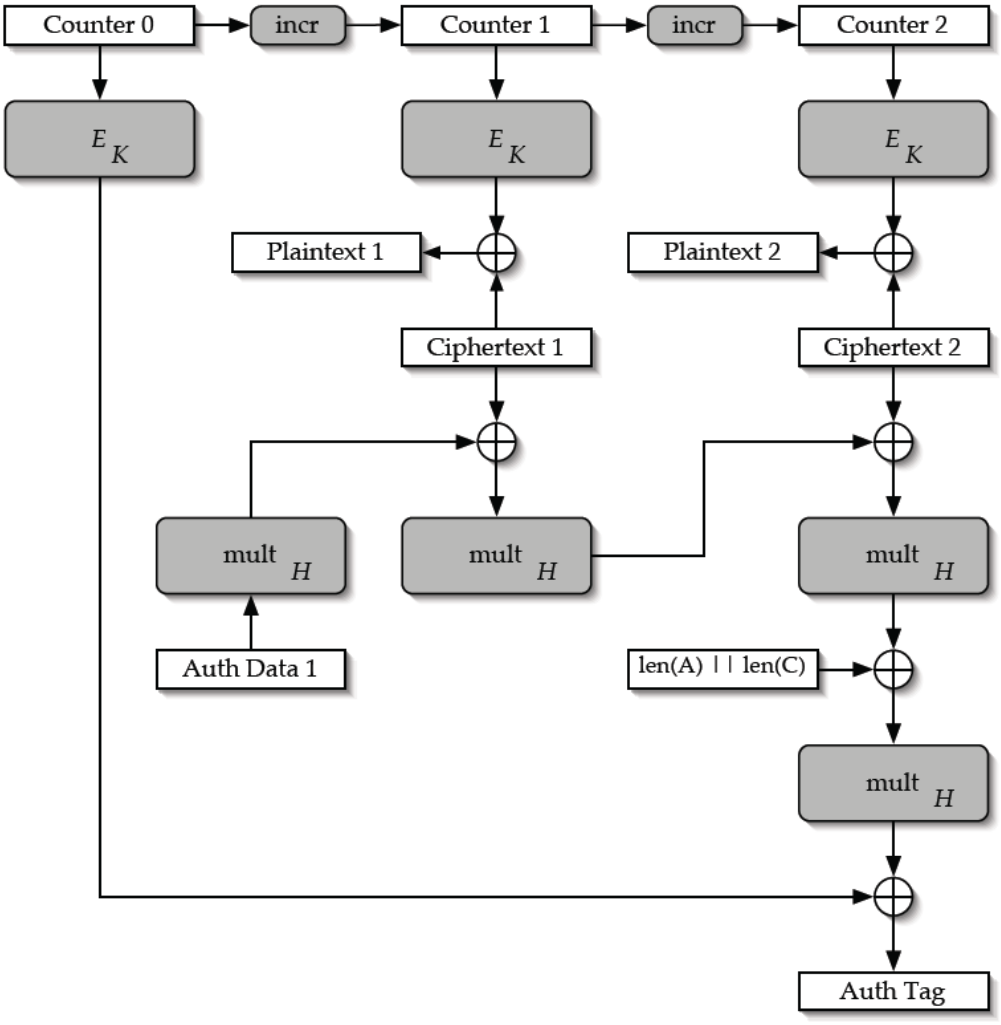


图 12-10 GCM 解密流程

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

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

12.3.7 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12-11所示。

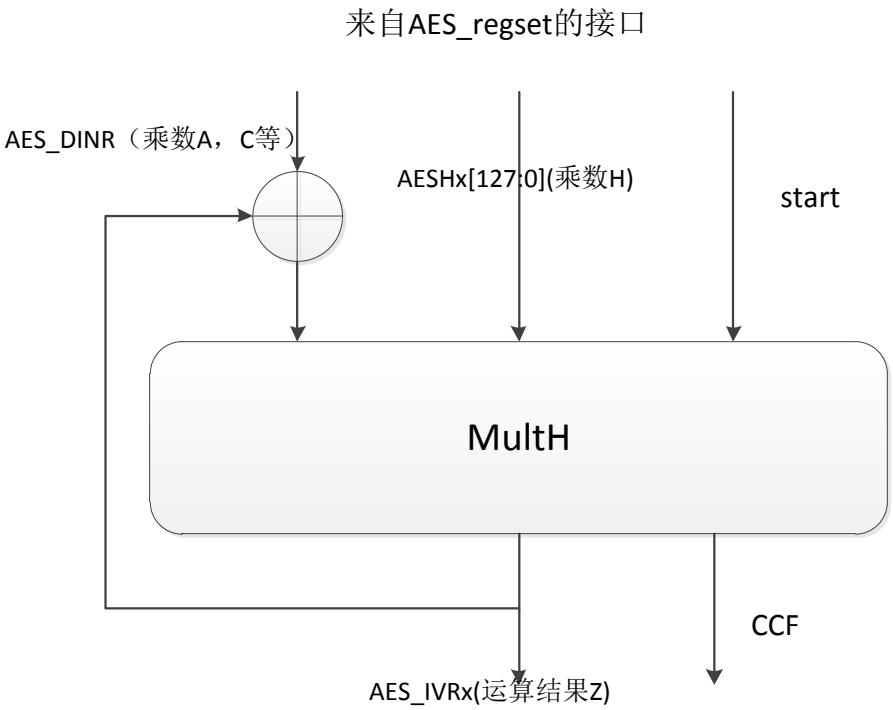


图 12-11 multH 模块框图

multH模块的输入寄存器为AES的输入寄存器AES_DINR和AESHx寄存器。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AESHx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。

12.3.8 推荐的 GCM 流程

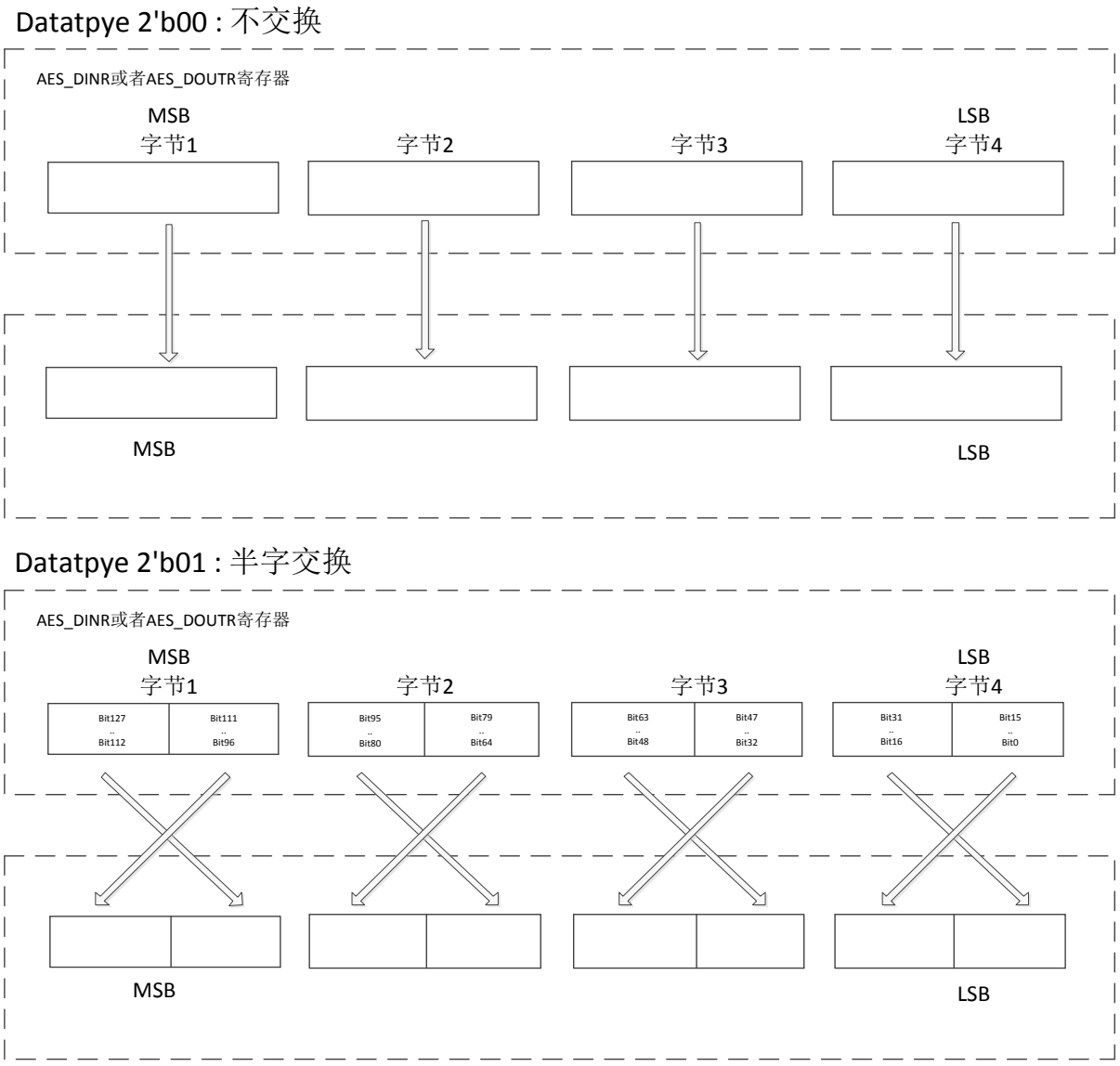
GCM模式的实现需要软硬件配合，本文档提供一种推荐的使用方法。

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。

12.4 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12-12所示。



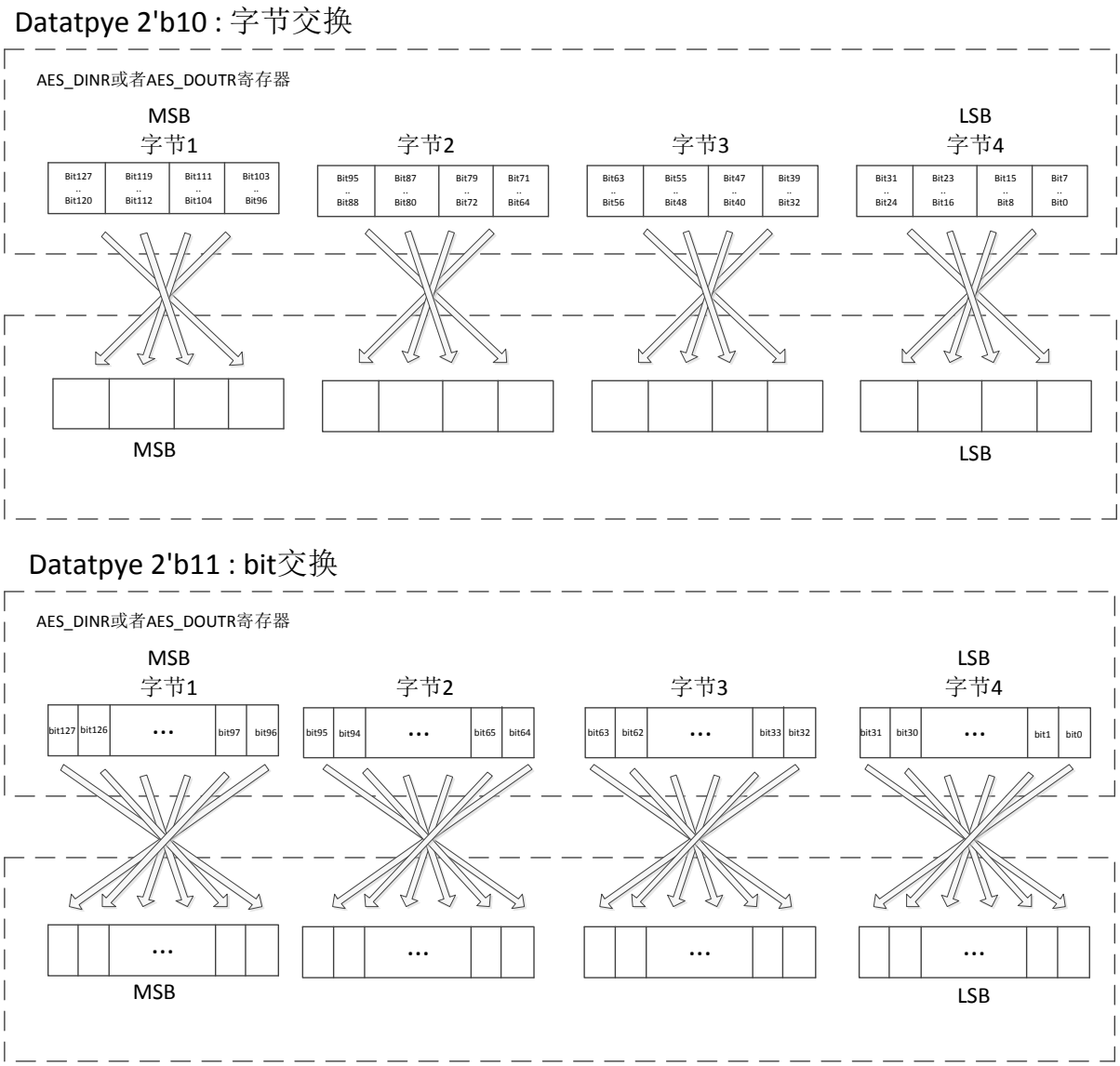


图 12-12 根据数据类型存储数据的示意图

12.5 工作流程

12.5.1 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。

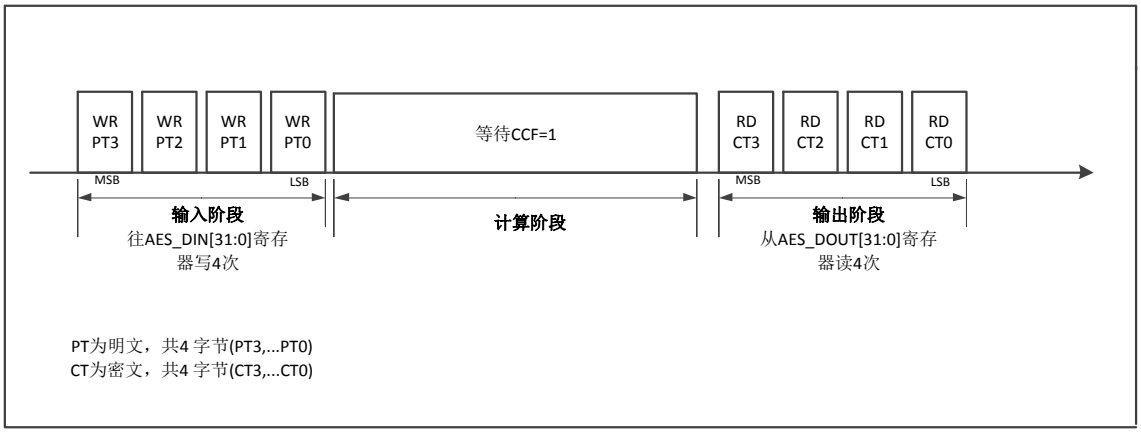


图 12-13 模式 1：加密流程

12.5.2 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。

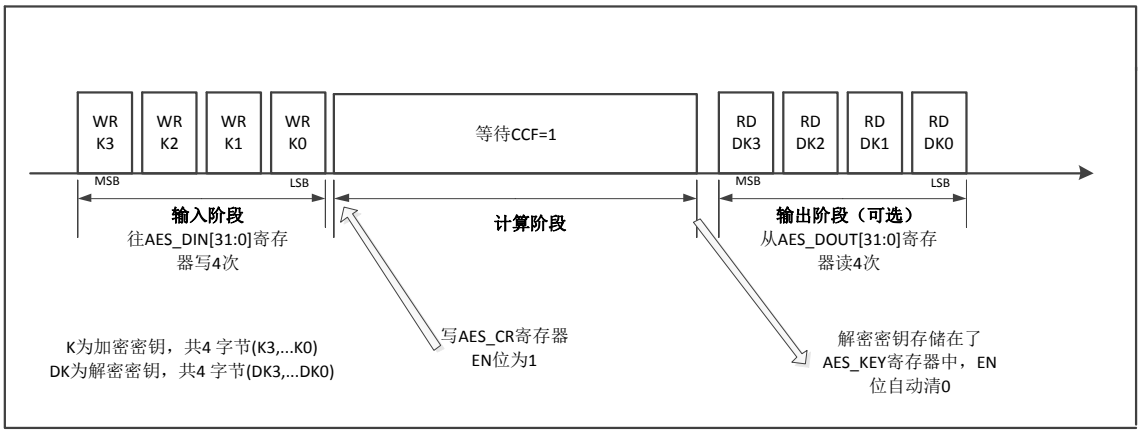


图 12-14 模式 2 示意图

12.5.3 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

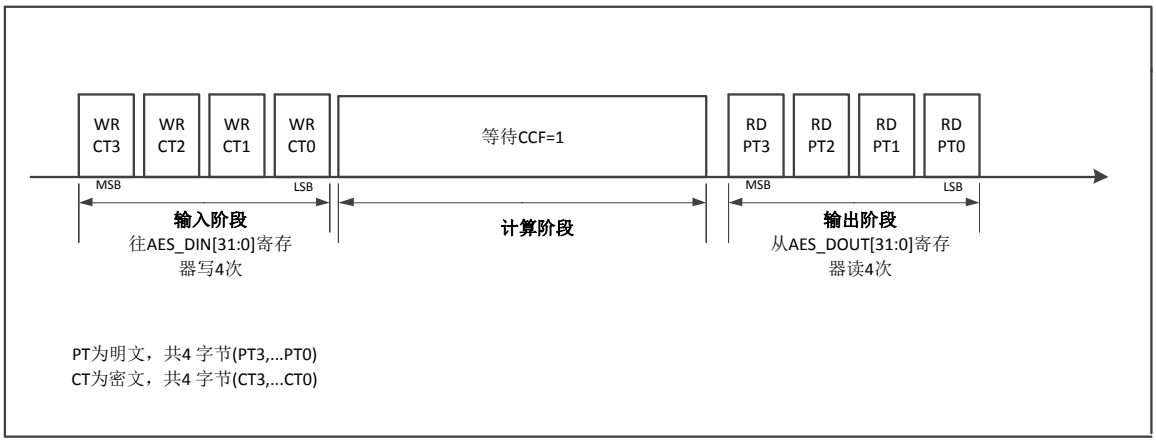


图 12-15 模式 3 示意图

12.5.4 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。

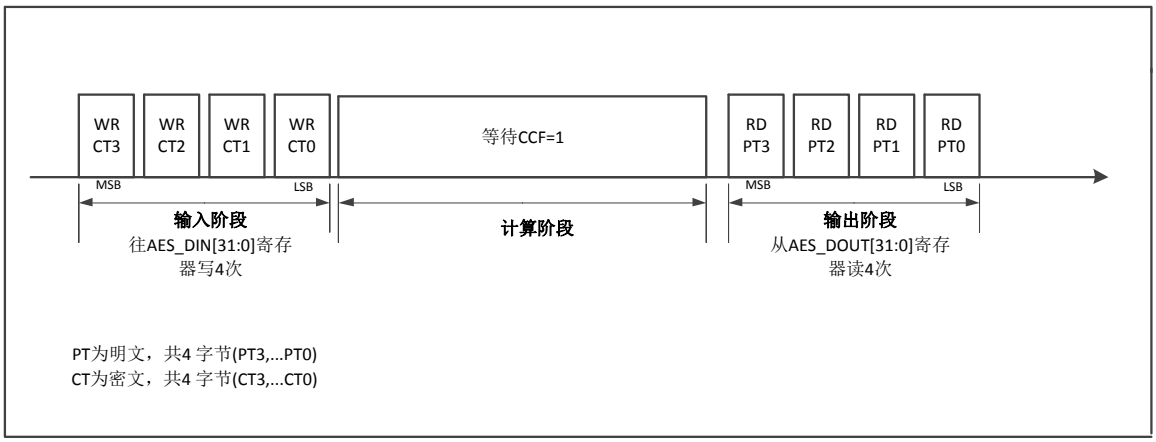


图 12-16 模式 4 示意图

12.5.5 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。
- 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。

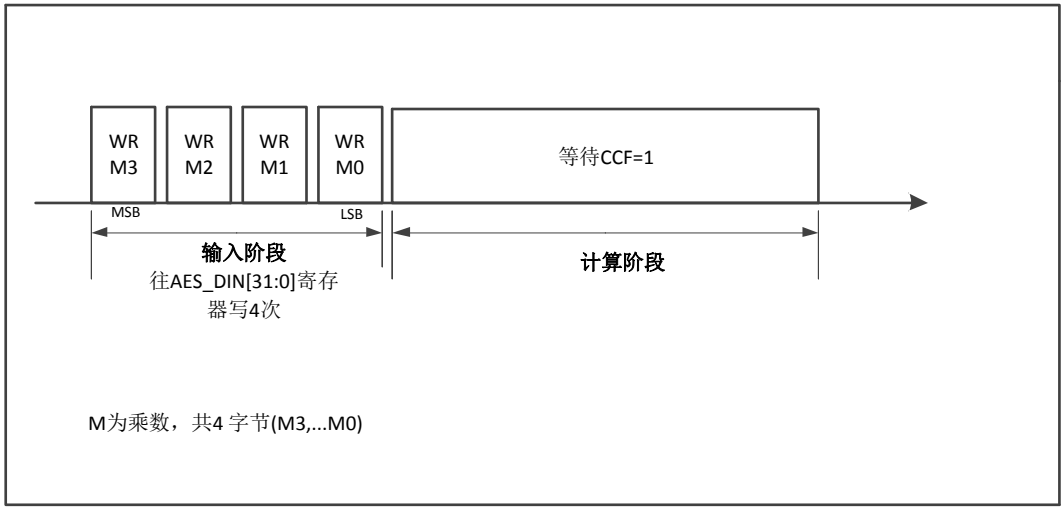


图 12-17 multH 模块使用流程示意图

12.6 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。

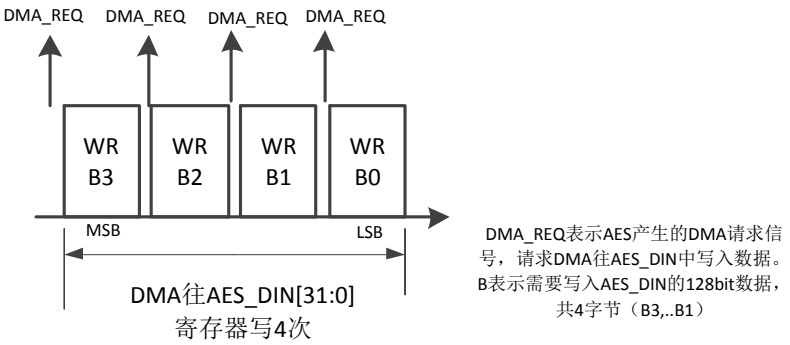


图 12-18 输入时 DMA 请求和数据传输示意图

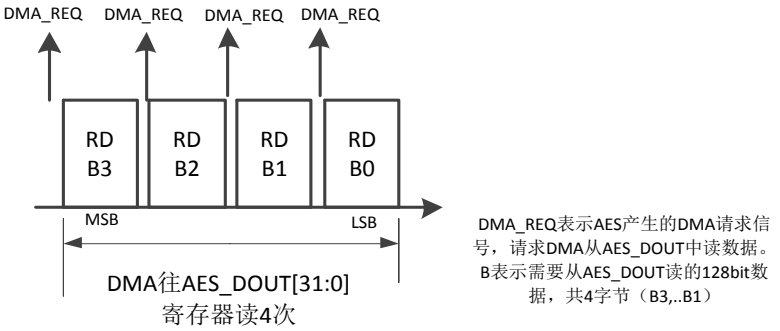


图 12-19 输出时 DMA 请求和数据传输示意图

12.6.1 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。

12.7 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。

12.8 寄存器

模块基地址：0x4001_3800

offset 地址	名称	符号
AES(模块起始地址: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

12.8.1 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	-		
位权限	U-0	R/W-00		R/W-0	R/W-0	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。
10:7	-	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) 00：模式 1：加密 01：模式 2：密钥扩展 10：模式 3：解密

位号	助记符	功能描述
		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

12.8.1 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)

12.8.2 AES 中断标志寄存器（AES_ISR）

名称	AES_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	-							
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/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	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

12.8.3 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。 MultH 模式：把乘数 A 或 C 从 MSB 到 LSB 分 4 次写入。

12.8.4 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。

12.8.5 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	W-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	KEYx[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	KEYx[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	KEYx[7:0]							
位权限	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，AESLKEY7 存放秘钥最高 32bit。(AES Key) *注意：此寄存器软件只可写，不可读；此寄存器内容不受 WWDT 复位影响*

12.8.6 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)

12.8.7 AES MultH 参数寄存器（AES_Hx）

名称	AES_Hx(x=0,1,2,3)							
offset	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]

13 公钥算法加速器（PAE）

13.1 概述

PAE（Public-Key Algorithm Engine）公钥算法引擎，是为公钥算法，如RSA、ECC(p)、SM2(p)，提供硬件加速的模块。

本模块为公钥算法提供了4种模式的加速功能：

(1) 单次模运算：支持字长从32bit至2048bit的模加、模减运算，支持字长从128bit至2048bit的蒙哥马利模乘运算；

(2) 单次点运算：支持字长从32bit至512bit的倍点、点加、蒙哥马利点乘三种基本运算；

(3) 连续RSA运算：支持字长从32bit至2048bit的32bit密钥连续RSA模幂运算；

(4) 连续ECC运算：支持字长从32bit至512bit的32bit密钥连续ECC点乘运算；

其中，单次模运算、梳状算法密钥预处理、模逆运算为模块的基本运算，其余5种模式下的计算是通过硬件调用模加、模减和蒙哥马利模乘来实现的。

PAE的结构框图如下：

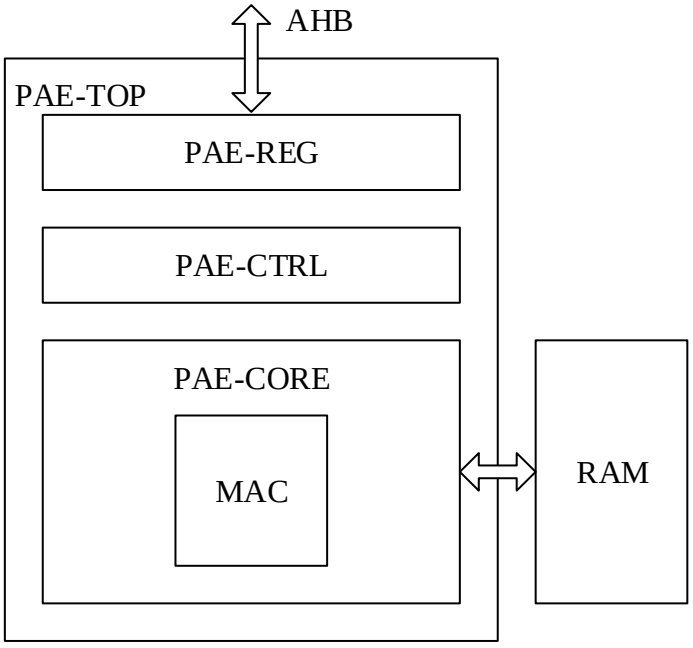


图 13-1 PAE 结构框图

PAE运算时需要占用1KB的SRAM，算法RAM位于总线地址：0x4000_1C00~0x4000_1FFF，运算时将无法被软件访问；

13.2 软件使用方法

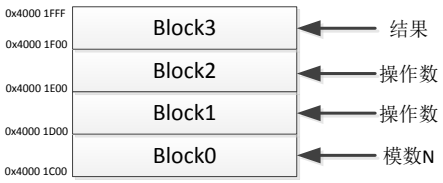
软件通过调用PAE模块实现基本模运算和点运算的硬件加速。本章节介绍软件如何调用PAE实现其所支持的各种基本运算。

注：复旦微电子为客户提供基于PAE硬件加速的公钥算法库函数，便于客户快速实现例如ECDSA等算法应用，并可以根据用户需要提供不同安全等级的实现方案。

13.2.1 蒙哥马利模加

蒙哥马利模加按照如下步骤实现：

- 配置PAE_MLR寄存器，设置模长；对于RSA，支持的模长是32~2048bit；对于ECC，支持的模长是32~512bit
- 根据模长配置RAM Block数量，对于512bit以下的模长，配置BLKCFG=1（16个Block），否则配置BLKCFG=0（4个Block）
- 将模数N、两个操作数写入RAM，并通过配置OP1_BLK、OP2_BLK、RES_BLK寄存器设置操作数和运算结果所在的Block；其中模数必须存放在Block0，操作数和结果可以存放在其他任意Block。下图以BLKCFG=0为例，说明一种典型的数据组织格式：



- 配置PAE_M0CFG寄存器，将INS_M0设置为模加运算
- 配置PAECSR寄存器，将RUN_MODE设置为单次模运算，置位START启动运算
- 等待BUSY信号清零，或中断标志置位
- 从特定Block中读取计算结果

13.2.2 蒙哥马利模加

调用方法与蒙哥马利模加类似，只是PAE_M0CFG.INS_M0寄存器设置为模减运算。

13.2.3 蒙哥马利模乘

蒙哥马利模乘按照如下步骤实现：

- 配置PAEMLR寄存器，设置模长，最小96bit
- 配置PAEMPR寄存器，设置模参
- 将模数N、两个操作数写入RAM，并通过配置OP1_BLK、OP2_BLK、RES_BLK寄存器设置操作数和运算结果所在的Block；模数应满足最低bit为1且最高bit为1；模数必须存放在Block0，操作数和结果不能设置为Block0和最高Block（Block3或BlockF）
- 配置PAE_M0CFG寄存器，将INS_M0设置为模乘运算
- 配置PAECSR寄存器，将RUN_MODE设置为单次模运算，置位START启动运算
- 等待BUSY信号清零，或中断标志置位
- 从特定Block中读取计算结果

13.2.4 Jacobin 坐标倍点

- 配置PAEMLR寄存器，设置模长为0x07
- 配置PAEMPR寄存器，设置模参
- 将模数N、操作数写入RAM，模数应满足最低bit为1且最高bit为1；模数和操作数需要按如下格式存放

	描述	RAM 16 BLK
N_BLK	模数	00
X1_BLK	点的雅克比坐标 X1	01
Y1_BLK	点的雅克比坐标 Y1	02
Z1_BLK	点的雅克比坐标 Z1	03
A_BLK	曲线方程系数 a	06
0_BLK	全 0	0E

- 配置PAE_M1CFG寄存器，INS_M1配置为000，AEN3_M1和BIT_VALUE_M1根据曲线方程设置
- 配置PAECSR寄存器，将RUN_MODE设为单次点运算，置位START启动运算
- 等待BUSY信号清零，或中断标志置位
- 从特定Block中读取计算结果，结果存放格式如下：

描述	RAM 16 BLK
倍点坐标 X3	01
倍点坐标 Y3	02
倍点坐标 Z3	03

13.2.5 混合坐标点加

- 配置PAEMLR寄存器，设置模长为0x07
- 配置PAEMPR寄存器，设置模参
- 将模数N、操作数写入RAM，模数应满足最低bit为1且最高bit为1；模数和操作数需要按如下格式存放

	描述	RAM 16 BLK
N_BLK	模数	00
X1_BLK	点 1 的雅克比坐标 X1	01
Y1_BLK	点 1 的雅克比坐标 Y1	02
Z1_BLK	点 1 的雅克比坐标 Z1	03
X2_BLK	点 2 的仿射坐标 X2	04
Y2_BLK	点 2 的仿射坐标 Y2	05
0_BLK	全 0	0E

- 配置PAE_M1CFG寄存器，INS_M1配置为001，AEN3_M1和BIT_VALUE_M1根据曲线方程设置
- 配置PAECSR寄存器，将RUN_MODE设为单次点运算，置位START启动运算
- 等待BUSY信号清零，或中断标志置位
- 从特定Block中读取计算结果，结果存放格式如下：

描述	RAM 16 BLK
点加坐标 X3	01
点加坐标 Y3	02
点加坐标 Z3	03

13.2.6 蒙哥马利点乘倍点运算（ECDBL）

- 配置PAEMLR寄存器，设置模长为0x07
- 配置PAEMPR寄存器，设置模参
- 将模数N、操作数写入RAM，模数应满足最低bit为1且最高bit为1；模数和操作数需要按如下格式存放

	描述	RAM 16 BLK
N_BLK	模数	00
X1_BLK	Q0 的射影坐标 X1	01
Y1_BLK	Q0 的射影坐标 Y1	02

Z1_BLK	Q0 的射影坐标 Z1	03
A_BLK	系数 a	06
0_BLK	全 0	0E

- 配置PAE_M1CFG寄存器，INS_M1配置为011，AEN3_M1和BIT_VALUE_M1根据曲线方程设置
- 配置PAECSR寄存器，将RUN_MODE设为单次点运算，置位START启动运算
- 等待BUSY信号清零，或中断标志置位
- 从特定Block中读取计算结果，结果存放格式如下：

描述	RAM 16 BLK
Q0 的射影坐标 X1	08
Q0 的射影坐标 Z1	09
Q1 的射影坐标 X2	0A
Q1 的射影坐标 Z2	07

13.2.7 蒙哥马利点乘倍点点加运算（ECADDDBL）

- 配置PAEMLR寄存器，设置模长为0x07
- 配置PAEMPR寄存器，设置模参
- 将模数N、操作数写入RAM，模数应满足最低bit为1且最高bit为1；模数和操作数需要按如下格式存放

	描述	RAM 16 BLK
N_BLK	模数	00
X1_BLK	Q0 的射影坐标 X1	08
Z1_BLK	Q0 的射影坐标 Z1	09
X2_BLK	Q1 的射影坐标 X2	0A
Z2_BLK	Q1 的射影坐标 Z2	07
X_BLK	基点 X 坐标	0B
A_BLK	ECC 系数 a	0C
B_BLK	ECC 系数 b	0D
0_BLK	全 0	0E

- 配置PAE_M1CFG寄存器，INS_M1配置为100，AEN3_M1和BIT_VALUE_M1根据曲线方程设置
- 配置PAECSR寄存器，将RUN_MODE设为单次点运算，置位START启动运算
- 等待BUSY信号清零，或中断标志置位
- 从特定Block中读取计算结果，结果存放格式如下：

描述	RAM 16 BLK
Q0 的射影坐标 X1	08
Q0 的射影坐标 Z1	09
Q1 的射影坐标 X2	0A
Q1 的射影坐标 Z2	07

13.2.8 蒙哥马利点乘 Y 坐标恢复运算（ECYRecover）

- 配置PAEMLR寄存器，设置模长为0x07
- 配置PAEMPR寄存器，设置模参
- 将模数N、操作数写入RAM，模数应满足最低bit为1且最高bit为1；模数和操作数需要按如下格式存放

	描述	RAM 16 BLK
N_BLK	模数	00
X1_BLK	Q0 的射影坐标 X1	01
Z1_BLK	Q0 的射影坐标 Z1	02
X2_BLK	Q1 的射影坐标 X2	03
Z2_BLK	Q1 的射影坐标 Z2	04
Y_BLK	基点 Y 坐标	0A
X_BLK	基点 X 坐标	0B
A_BLK	ECC 系数 a	0C
B_BLK	ECC 系数 b	0D
0_BLK	全 0	0E

- 配置PAE_M1CFG寄存器，INS_M1配置为101，AEN3_M1和BIT_VALUE_M1根据曲线方程设置
- 配置PAECSR寄存器，将RUN_MODE设为单次点运算，置位START启动运算
- 等待BUSY信号清零，或中断标志置位
- 从特定Block中读取计算结果，结果存放格式如下：

描述	RAM 16 BLK
Q0 的射影坐标 X1	01
Q0 的射影坐标 Y1	02
Q0 的射影坐标 Z1	03

13.2.9 连续 RSA

- 配置PAEMLR寄存器，设置模长为0x03~0x7F
- 配置PAEMPR寄存器，设置模参
- 配置连续运算输入寄存器PAE_WORD
- 将模数N、操作数写入RAM，模数应满足最低bit为1且最高bit为1；模数和操作数需要按如下格式存放

	描述	RAM 4 BLK
N_BLK	模数	00
C_BLK	初始密文	01
M_BLK	明文	02

- 配置PAE_M2CFG，根据安全性需求将ALWAYS_MULT置为0或1，FIND_BIT_ONE表示是否找到第一个为1的计算位，初始值置0，当模块找到第一个1后硬件置位
- 写状态控制寄存器PAECSR，RUN_MODE设置为连续RSA运算，START位置1启动计算

- 等待BUSY信号清零，或中断标志置位
- 从RAM读计算结果，RSA加密运算后的密文存储在Block1

13.2.10 连续 ECC

单次调用计算32bit密钥，在24Mhz频率下用时约6.67ms；对于256bit的签名运算，需要9~10次调用，因此总运算时间大约小于70ms。

- 配置PAEMLR寄存器，设置模长为0x07
- 配置PAEMPR寄存器，设置模参
- 配置连续运算输入寄存器PAE_WORD
- 将模数N、操作数写入RAM，模数应满足最低bit为1且最高bit为1；模数和操作数需要按如下格式存放

	描述	RAM 16 BLK
N_BLK	模数	00
X1_BLK	Q0 的射影坐标 X1	08
Z1_BLK	Q0 的射影坐标 Z1	09
X2_BLK	Q1 的射影坐标 X2	0A
Z2_BLK	Q1 的射影坐标 Z2	07
X_BLK	基点 X 坐标	0B
A_BLK	ECC 系数 a	0C
B_BLK	ECC 系数 b	0D
0_BLK	全 0	0E

- 配置PAE_M3CFG，设置AEN3_M3，FIND_BIT_ONE表示是否找到第一个为1的计算位，初始值置0，当模块找到第一个1后硬件置位
- 写状态控制寄存器PAECSR，RUN_MODE设置为连续ECC运算，START位置1启动计算
- 等待BUSY信号清零，或中断标志置位
- 从RAM读计算结果，结果存放地址如下

描述	RAM 16 BLK
Q0 的射影坐标 X1	08
Q0 的射影坐标 Z1	09
Q1 的射影坐标 X2	0A
Q1 的射影坐标 Z2	07

13.3 寄存器

模块基地址：0x4000_1400

offset 地址	名称	符号
PAE(模块起始地址:0x40001400)		
0x00	PAE 控制状态寄存器	PAE_CSR
0x04	PAE 模长寄存器	PAE_MLR
0x08	PAE 模参寄存器	PAE_MPR
0x0C	PAE 模式 0 配置寄存器	PAE_M0CFG
0x10	PAE 模式 1 配置寄存器	PAE_M1CFG
0x14	PAE 模式 2 配置寄存器	PAE_M2CFG
0x18	PAE 模式 3 配置寄存器	PAE_M3CFG
0x1C	PAE 输入数据寄存器	PAE_WORD

13.3.1 PAE 控制状态寄存器（PAE_CSR）

名称	PAE_CSR							
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	-						DATA_TYPE	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	START	-	RUN_MODE		SOFTRST	PAEIE	PAEIF	BUSY
位权限	W-0	U-0	R/W-00		W-0	R/W-0	R/W-0	R-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9:8	DATA_TYPE	输入数据顺序交换控制 00：不交换 01：half-word 交换 10：byte 交换 11：bit 交换 此寄存器用于对输入 32bit 数据的位序进行调整；比如输入为 0x1234_5678，half-word 交换后为 0x5678_1234，byte 交换后为 0x7856_3412，bit 交换后为 0x1E6A_2C48
7	START	运算启动位，软件写 1 启动计算，计算结束后硬件自动清零
6	-	RFU：未实现，读为 0
5:4	RUN_MODE	模块工作模式： 00：模式 0，单次模运算，可配置 4Block/16Block 01：模式 1，单次点运算，固定 16Block 10：模式 2，32bit 密钥连续 RSA 运算，固定 4Block

位号	助记符	功能描述
		11：模式 3，32bit 密钥连续 ECC 运算，固定 16Block
3	SOFTRST	PAE 软件复位，软件写 1 复位 PAE 内部主要寄存器（模长寄存器 PAE_MLR 和模参寄存器 PAE_MPR 不受影响），复位结束后硬件自动清零 注意：软件写 SOFTRST 之后，需插入至少一条 NOP 指令后，再写入 PAE 其他寄存器；在 SOFTRST 操作后立即写 PAE 寄存器可能写入失败。
2	PAEIE	PAE 中断使能 1：使能中断输出 0：禁止中断输出
1	PAEIF	PAE 中断标志寄存器
0	BUSY	PAE 工作标志 1：PAE 正在运算中 0：PAE 空闲

13.3.2 PAE 模长寄存器（PAE_MLR）

名称	PAE_MLR							
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	-		PAE_MLR					
位权限	U-0		R/W-000000					

位号	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5:0	PAE_MLR	在 RAM 配置为 4 Block 时，模长最大为 2048bit： 模长=（MLR[5:0]+1）*32 位。 在 RAM 配置为 16 Block 时，模长最大为 512bit，MLR 只有低 4bit 有效： 模长=（MLR[3:0]+1）*32 位。

13.3.3 PAE 模参寄存器（PAE_MPR）

名称	PAE_MPR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PAE_MPR[31:24]							
位权限	W-00000000							

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

位号	助记符	功能描述
31:0	PAE_MPR	32bit 模参寄存器，只可写，不可读

13.3.4 模式 0（单次模运算）配置寄存器（PAE_M0CFG）

名称	PAE_M0CFG							
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	BLKCFG	-	INS_M0		RES_BLK			
位权限	R/W-0	U-0	R/W-00		R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OP1_BLK				OP2_BLK			
位权限	R/W-0000				R/W-0000			

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15	BLKCFG	RAM Block 配置位 1：算法 RAM 配置为 16 Block 0：算法 RAM 配置位 4 Block
14	-	RFU：未实现，读为 0
13:12	INS_M0	单次模运算指令： 00：蒙哥马利模乘 01：蒙哥马利模加 10：RFU 11：蒙哥马利模减
11:8	RES_BLK	配置运算结果所在 Block 当 BLK_CFG==1，4bit 有效，表示 Block0~Block15 当 BLK_CFG==0，低 2bit 有效，表示 Block0~Block3
7:4	OP1_BLK	操作数 1 地址： 当 BLK_CFG==1，4bit 有效，表示 Block0~Block15 当 BLK_CFG==0，低 2bit 有效，表示 Block0~Block3
3:0	OP2_BLK	操作数 2 地址：

位号	助记符	功能描述
		当 BLK_CFG==1，4bit 有效，表示 Block0~Block15 当 BLK_CFG==0，低 2bit 有效，表示 Block0~Block3

13.3.5 模式 1（单次点运算）配置寄存器（PAE_M1CFG）

名称	PAE_M1CFG							
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	-	INS_M1			-		AEN3_M1	BIT_VALUE_M1
位权限	U-0	R/W-000			U-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
6:4	INS_M1	单次点运算指令： 000：Jacobin 坐标倍点运算 001：混合坐标点加运算 010：RFU 011：蒙哥马利点乘倍点运算（ECDBL） 100：蒙哥马利点乘倍点点加运算（ECADDDBL） 101：蒙哥马利点乘 Y 坐标恢复运算（ECYRecover） 110：RFU 111：RFU
3:2	-	RFU：未实现，读为 0
1	AEN3_M1	参数 A 指示信号 1：A 等于-3 0：A 不等于-3
0	BIT_VALUE_M1	当前计算密钥位指示标志 1：当前位为 1 0：当前位为 0

13.3.6 模式 2（32bit 连续 RSA）配置寄存器（PAE_M2CFG）

名称	PAE_M2CFG							
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	-			ALWAYS_MULT	-			FBO_M2
位权限	U-0			R/W-0	U-0			R/W-0

位号	助记符	功能描述
31:5	-	RFU：未实现，读为 0
4	ALWAYS_MULT	虚模乘控制位 1：当前密钥位为 0 时同样进行模乘操作 0：当前密钥位为 0 时不进行模乘操作
3:1	-	RFU：未实现，读为 0
0	FBO_M2	密钥位指示信号，硬件自动置位，软件清零 1：已找到为 1 的 bit 0：未找到为 1 的 bit

13.3.7 模式 3（32bit 连续 ECC）配置寄存器（PAE_M3CFG）

名称	PAE_M3CFG							
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	-			AEN3_M3	-			FBO_M3
位权限	U-0			R/W-0	U-0			R/W-0

位号	助记符	功能描述
31:5	-	RFU：未实现，读为 0
4	AEN3_M3	参数 A 指示信号 1：A 等于-3 0：A 不等于-3
3:1	-	RFU：未实现，读为 0
0	FBO_M3	密钥位指示信号，硬件自动置位，软件清零 1：已找到为 1 的 bit 0：未找到为 1 的 bit

13.3.8 PAE 输入数据寄存器（PAE_WORD）

名称	PAE_WORD							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PAE_WORD[31:24]							
位权限	W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PAE_WORD[23:16]							
位权限	W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PAE_WORD[15:8]							
位权限	W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PAE_WORD[7:0]							
位权限	W-00000000							

位号	助记符	功能描述
31:0	PAE_WORD	32bit 数据寄存器，只可写，不可读；仅在连续 RSA 运算和连续 ECC 运算模式下有效。

14 散列算法加速器（HASH）

14.1 概述

HASH模块主要实现SHA1算法和SHA256算法。将需要进行数据摘要的消息存入RAM特定地址，然后写控制状态寄存器启动相应的运算，待运算结束后，通过读取RAM特定地址得到摘要结果。

对于最长可达（2^{64}-1）位的消息，散列处理器可以计算其消息摘要，其中SHA-1算法得到的摘要为160位，SHA-256得到的摘要为256位。

散列处理器符合以下标准：

- FIPS PUB 180-1 安全散列标准规范（SHA-1）
- FIPS PUB 180-2 安全散列标准规范（SHA-256）

散列计算性能：

- 对512bits输入消息，计算SHA-1需要大约1600个AHBCLK时钟周期
- 对于512bits输入消息，计算SHA-256需要大约1500个AHBCLK时钟周期。

HASH加速器没有中断输出，软件需要通过查询BUSY标志来确认运算完成。

14.2 结构框图

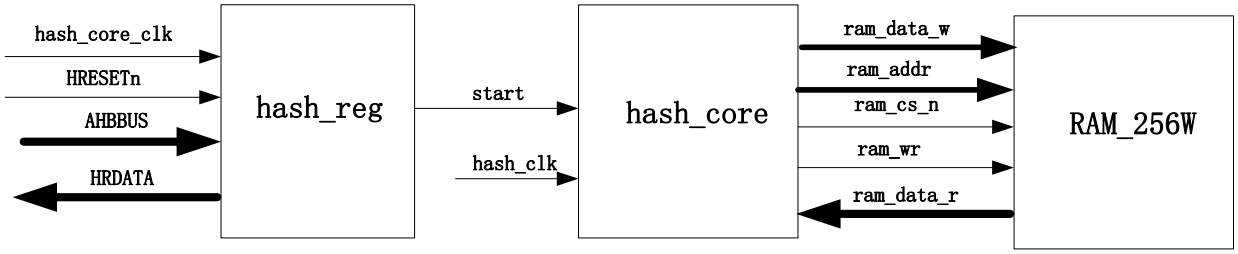


图 14-1 HASH 结构框图

HASH_REG模块为两算法模块共用的控制寄存器，用于控制运算启动以及输出运算模块状态信息。

HASH_CORE模块支持SHA1和SHA256运算，进行运算的消息数据、摘要结果和运算的中间数据都保存在RAM中。

注意：散列算法处理器与公钥算法加速器共用1KB算法RAM，应用中要注意避免数据冲突。

14.3 使用说明

每次HASH运算输入消息分组数据长度为64字节，软件将消息存储于0x4000_1C00~0x4000_1C3F的RAM中，将初始向量IV(0)存储于RAM地址0x4000_1C60~0x4000_1C7F（SHA256）或者0x4000_1C60~0x4000_1C73（SHA1），写控制寄存器启动HASH算法。

HASH算法模块从RAM中读出运算所需要的消息和运算所需要的初始向量，经过一轮运算之后，将运算结果存储回RAM地址0x4000_1C60~0x4000_1C7F（SHA256）或0x4000_1C60~0x4000_1C73（SHA1）。运算过程中，中间运算结果暂存在RAM地址0x4000_1C40~0x4000_1C5F（SHA256）或0x4000_1C40~0x4000_1C53（SHA1）中，同时算法模块输出BUSY信号。运算结束后，算法模块自动把START信号和BUSY信号拉低。

一个消息分组运算结束后，若还有剩余组消息需要运算，则软件将下一组消息写入RAM的0x4000_1C00~0x4000_1C3F地址。写控制寄存器启动HASH算法。HASH算法模块重复以上运算过程。

整个消息（多个分组）运算结束后，软件从RAM的地址0x4000_1C60~0x4000_1C7F（SHA256）或者0x4000_1C60~0x4000_1C73（SHA1）读出最终的摘要结果。

RAM内数据组织格式可以参考下表：

算法	数据	RAM 地址
SHA256	64 字节消息分组	0x4000_1C00~0x4000_1C3F
	32 字节初始向量 IV(0)	0x4000_1C60~0x4000_1C7F
	32 字节临时数据	0x4000_1C40~0x4000_1C5F
	32 字节摘要结果	0x4000_1C60~0x4000_1C7F
SHA1	64 字节消息分组	0x4000_1C00~0x4000_1C3F
	20 字节初始向量 IV(0)	0x4000_1C60~0x4000_1C73
	20 字节临时数据	0x4000_1C40~0x4000_1C53
	20 字节摘要结果	0x4000_1C60~0x4000_1C73

消息和摘要数据顺序参考下表，数据为小端格式：

数据类型	数据排列	Word 地址
Digest	H0	0x4000_1C60
	H1	0x4000_1C64
	H2	0x4000_1C68
	H3	0x4000_1C6C
	H4	0x4000_1C70
	H5（SHA1 时全 0）	0x4000_1C74
	H6（SHA1 时全 0）	0x4000_1C78
	H7（SHA1 时全 0）	0x4000_1C7C
Message	M0	0x4000_1C00

	M1	0x4000_1C04
	M2	0x4000_1C08
	M3	0x4000_1C0C
	M4	0x4000_1C10
	M5	0x4000_1C14
	M6	0x4000_1C18
	M7	0x4000_1C1C
	M8	0x4000_1C20
	M9	0x4000_1C24
	M10	0x4000_1C28
	M11	0x4000_1C2C
	M12	0x4000_1C30
	M13	0x4000_1C34
	M14	0x4000_1C38
	M15	0x4000_1C3C

14.4 寄存器

模块基地址：0x4000_1800

offset 地址	名称	符号
HASH(模块起始地址:0x40001800)		
0x00	HASH 控制和状态寄存器 (Hash Control and Status Register)	HASH_CSR
0x04	HASH 数据类型寄存器 (Hash Data Type Register)	HASH_DTR

14.4.1 HASH 控制状态寄存器（HASH_CSR）

名称	HASH_CSR							
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	START	-					HASHSEL	-
位权限	R/W-0	U-0					R/W-0	U-0

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7	START	HASH 运算启动寄存器，软件写 1 启动运算，运算结束后硬件自动清零
6:2	-	RFU：未实现，读为 0
1	HASHSEL	SHA 算法选择 0：SHA256 1：SHA1
0	-	RFU：未实现，读为 0

14.4.2 HASH 数据类型寄存器（HASH_DTR）

名称	HASH_DTR							
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							

名称	HASH_DTR							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						DATA_TYPE	
位权限	U-0						RW-00	

位号	助记符	功能描述
31:2	-	未实现：读为 0
1:0	DATA_TYPE	输入或输出数据顺序交换控制 00：不交换 01：half-word 交换 10：byte 交换 11：bit 交换 此寄存器用于对输入 32bit 数据的位序进行调整；比如输入为 0x1234_5678，half-word 交换后为 0x5678_1234，byte 交换后为 0x7856_3412，bit 交换后为 0x1E6A_2C48

15 随机数发生器（TRNG）

15.1 概述

FM33A0xxEV使用2个Galois真随机噪声源作为真随机数种子，配合简单在线检测（32位全0全1检测）、LFSR后处理、伪随机LFSR共同组成芯片的随机数发生器。

TRNG的启动测试和完整的在线测试功能需要固件实现。

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

真随机数发生器通过了FIPS PUB140-2测试，成功率99.9%。

15.2 功能描述

15.2.1 随机数产生

下图为真随机数发生器结构框图。

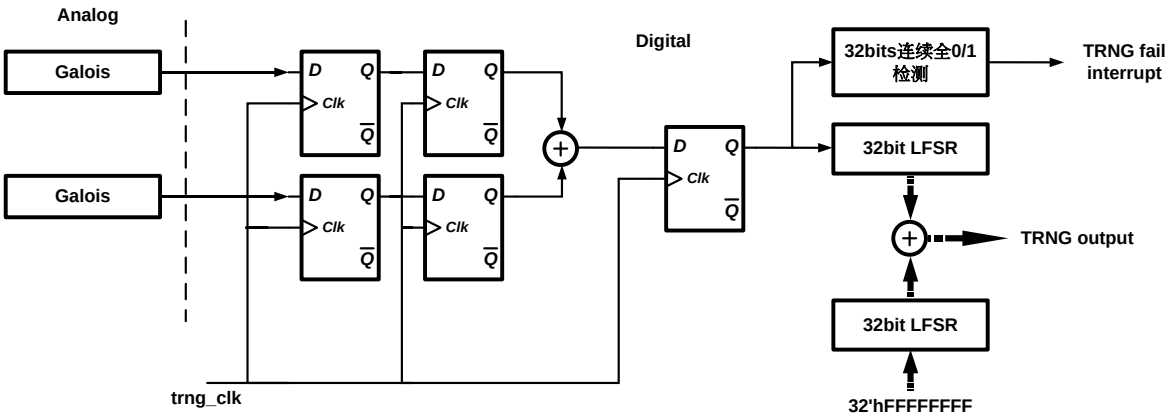


图 15-1 真随机数模块框图

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

15.2.2 工作时钟

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

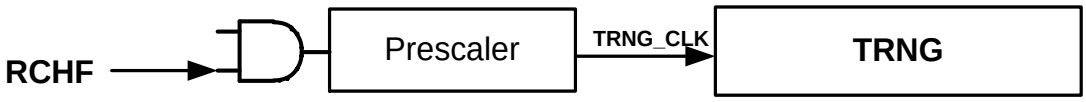


图 15-2 真随机数模块工作时钟

15.2.3 随机数读取

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

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

15.2.4 CRC 运算

用作随机数后处理的LFSR也可用于进行CRC计算。

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

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

CRC多项式：

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

软件操作流程：

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

15.3 寄存器

模块基地址：0x4001_3C00

offset 地址	名称	符号
RNG(模块起始地址:0x40013C00)		
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_CRC_CR
0x18	CRC 输入数据寄存器（CRC Data input Register）	RNG_CRC_DIR
0x1C	CRC 标志寄存器（CRC Status Register）	RNG_CRC_SR

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

15.3.2 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	OUT[31:24]							
位权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	OUT[23:16]							
位权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OUT[15:8]							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OUT[7:0]							
位权限	R-00000000							

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

15.3.3 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	-						LFSREN	RNF
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	LFSREN	LFSR 状态标志，只读 (LFSR enable) 1：LFSR 在运行中，不可进行 CRC 验证 0：LFSR 不在运行中，可进行 CRC 验证 注：本寄存器不会引起模块中断，仅供查询
0	RNF	随机数生成失败标志，软件写 1 清零 (Random Number Fail) 1：随机数未能通过质量检测 0：随机数通过质量检测

15.3.4 CRC 控制寄存器（RNG_CRC_CR）

名称	RNG_CRC_CR							
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

位号	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

15.3.5 CRC 输入数据寄存器（RNG_CRC_DIR）

名称	RNG_CRC_DIR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CRCIN[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CRCIN[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CRCIN[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CRCIN[7:0]							
位权限	R/W-1111 1111							

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

15.3.6 CRC 标志寄存器（RNG_CRC_SR）

名称	RNG_CRC_SR
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

16 模拟比较器（Comparator）

16.1 概述

芯片集成2个比较器，支持以下特性：

- 2个比较器，轨到轨输入，低功耗模式和快速模式
- 灵活的输入选择
  - IO引脚输入
  - 内部基准电压及其分压
- 可编程的功耗和速度选项
- 灵活的输出连接
  - 引脚输出
  - 连接到内部定时器输入
- 可编程的输出数字滤波功能
- 中断事件可唤醒MCU

16.2 结构框图

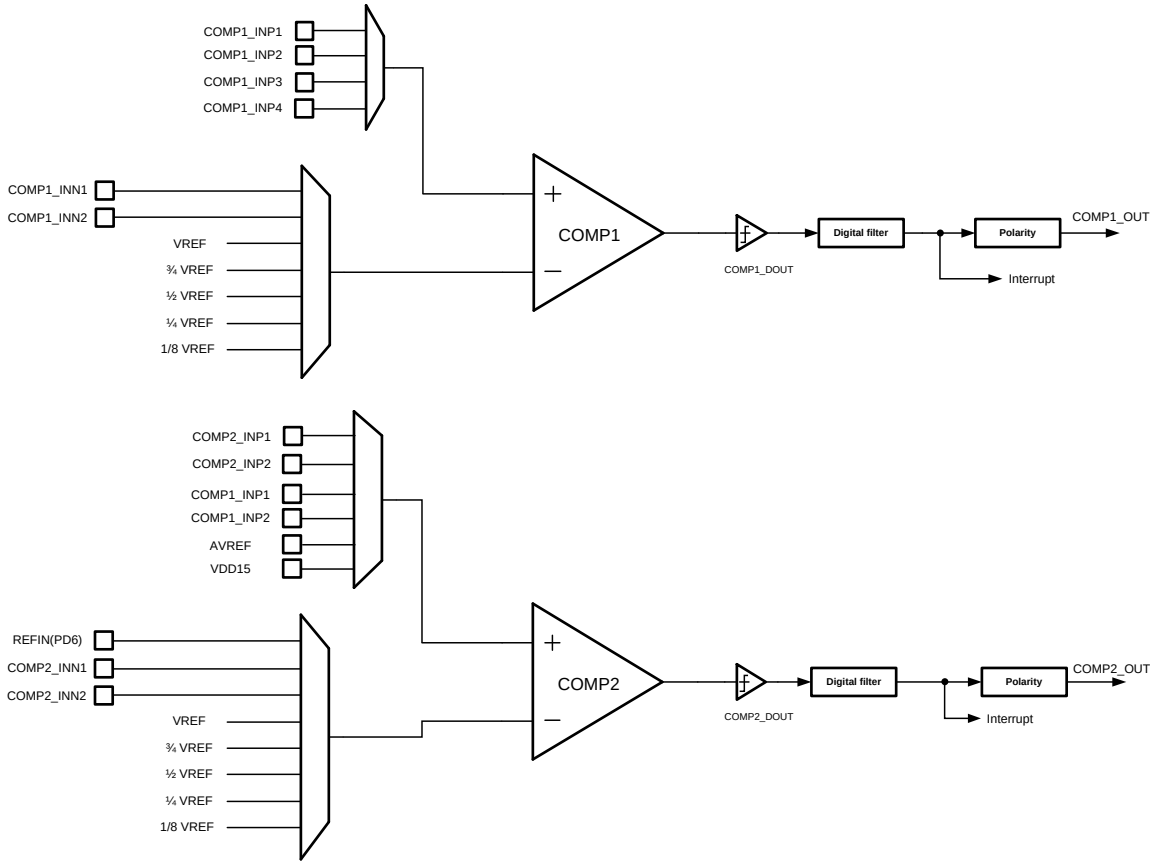


图 16-1 比较器电路框图

比较器结构如上图所示，基准电压VREF输入经过BUFFER模块后输出VREF和VREF分压，BUFFER输出可以配置为BYPASS，在配置成BYPASS使能后，BUFFER功能关闭，VREF直接输出，无法输出VREF分压。两个比较器完全相同，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，也可以输出到GPIO。

16.3 功能描述

16.3.1 基本功能

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

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

比较器输出的逻辑信号可以经过数字滤波和极性控制后输出，或产生中断信号。

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

16.3.2 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中使能比较器总线时钟。

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

16.3.3 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 正端输入	GPIO	V1PSEL 寄存器
COMP1_INP1	PF6	00
COMP1_INP2	PG2	01
COMP1_INP3	PG3	10
COMP1_INP4	PC15	11

表 16-1 比较器 1 正端输入列表

COMP1 负端输入	GPIO	V1NSEL 寄存器
COMP1_INN1	PF5	000
COMP1_INN2	PC14	001

表 16-2 比较器 1 负端输入列表

COMP2 正端输入	GPIO	V2PSEL 寄存器

COMP2_INP1	PB1	000
COMP2_INP2	PE4	001
COMP1_INP1	PF6	010
COMP1_INP2	PG2	011

表 16-3 比较器 2 负端输入列表

COMP2 负端输入	GPIO	V2NSEL 寄存器
COMP2_INN1	PB0	000
COMP2_INN2	PE3	001

表 16-4 比较器 2 负端输入列表

16.3.4 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。

下图是DFLEN=3的示意图：

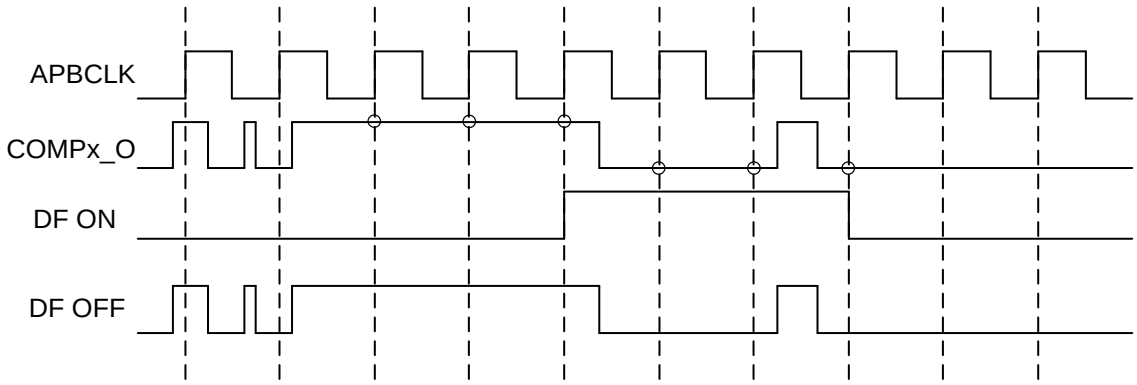


图 16-2 数字滤波（DFLEN=3）波形示意图

16.3.5 功耗和速度模式

两个比较器都支持轨到轨输入，输入电压范围0~VDD，在常温环境的低功耗模式下建立时间小于15us，传输延迟小于2us，功耗小于2uA；在常温环境的快速模式下建立时间小于5us，传输延迟小于0.5us，功耗5uA。

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
快速模式	5uA	0.3us	5us
低功耗模式	2uA	2us	15us

表 16-5 比较器工作模式（常温 TA=25℃）

16.4 寄存器

模块基地址：0x4001_5400

offset 地址	名称	符号
COMP(模块起始地址:0x40015400)		
0x00	COMP1 控制寄存器	COMP_CR1
0x04	COMP2 控制寄存器	COMP_CR2
0x08	COMP 中断配置寄存器	COMP_ICR
0x0C	COMP 中断标志寄存器	COMP_ISR
0x10	COMP 中断标志寄存器	COMP_PCR

16.4.1 COMP1 控制寄存器（COMP_CR1）

名称	COMP_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	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	-						CMP1OR	CMP1OF
位权限	U-0						R-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	V1NSEL			-	V1PSEL		CMP1EN
位权限	U-0	R/W-000			U-0	R/W-00		R/W-0

位号	助记符	功能描述
31:24	-	RFU：未实现，读为 0
23:19	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 00000：1 次 APBCLK 采样 00001：2 次 APBCLK 采样 …… 11111：32 次 APBCLK 采样
18	WINMODE	比较器 1 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
15:10	-	RFU：未实现，读为 0

位号	助记符	功能描述
9	CMP1OR	比较器 1 输出（数字滤波前），软件只读
8	CMP1OF	比较器 1 输出（数字滤波后），软件只读
7	-	RFU：未实现，读为 0
6:4	V1NSEL	比较器 1 负极输入选择 000：COMP1_INN1 001：COMP1_INN2 010：VREF 011：3/4 VREF 100：1/2 VREF 101：1/4 VREF 110：1/8 VREF 111：RFU
3	-	RFU：未实现，读为 0
2:1	V1PSEL	比较器 1 正极输入选择 00：COMP1_INP1 01：COMP1_INP2 10：COMP1_INP3 11：COMP1_INP4
0	CMP1EN	比较器 1 使能位 0：关闭比较器 1 1：使能比较器 1

16.4.2 COMP2 控制寄存器（COMP_CR2）

名称	COMP_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	DFLEN					WINMODE	POLAR	DFEN
位权限	R/W-0 0000					R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CMP2OR	CMP2OF
位权限	U-0						R-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	V2NSEL			V2PSEL			CMP2EN
位权限	U-0	R/W-000			R/W-000			R/W-0

位号	助记符	功能描述
31:24	-	RFU：未实现，读为 0
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 00000：1 次 APBCLK 采样 00001：2 次 APBCLK 采样 ……

位号	助记符	功能描述
		11111：32 次 APBCLK 采样
18	WINMODE	比较器 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 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
15:10	-	RFU：未实现，读为 0
9	CMP2OR	比较器 2 输出（数字滤波前），软件只读
8	CMP2OF	比较器 2 输出（数字滤波后），软件只读
7	-	RFU：未实现，读为 0
6:4	V2NSEL	比较器 2 负极输入选择 000：COMP2_INN1 001：COMP2_INN2 010：VREF 011：3/4 VREF 100：1/2 VREF 101：1/4 VREF 110：1/8 VREF 111：REFIN (PD6)
3:1	V2PSEL	比较器 2 正极输入选择 000：COMP2_INP1 001：COMP2_INP2 010：COMP1_INP1 011：COMP1_INP2 100：AVREF 101：VDD15 110, 111：RFU
0	CMP2EN	比较器 2 使能位 0：关闭比较器 2 1：使能比较器 2

16.4.3 COMP 中断配置寄存器（COMP_ICR）

名称	COMP_ICR							
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	-						OOW_IE	WIN_IE

位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BUFBYP	BUF_OFF	CMP2SEL		CMP1SEL		CMP2IE	CMP1IE
位权限	RW-0	RW-1	R/W-00		R/W-00		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	OOW_IE	Out-Of-Window 中断使能
8	WIN_IE	Window 中断使能
7	BUFBYP	比较器 Buffer Bypass 0：不 Bypass 比较器 Buffer 1：Bypass 比较器 Buffer
6	BUF_OFF	比较器 Buffer 使能 0：使能比较器 Buffer 1：禁止比较器 Buffer
5:4	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
3:2	CMP1SEL	比较器 1 中断源选择 00/11：比较器 1 输出上升或下降沿产生中断 01：比较器 1 输出上升沿产生中断 10：比较器 1 输出下降沿产生中断
1	CMP2IE	比较器 2 中断使能
0	CMP1IE	比较器 1 中断使能

*注：为了避免误触发中断，应在关闭中断使能的情况下设置中断源选择寄存器。

16.4.4 COMP 中断标志寄存器（COMP_ISR）

名称	COMP_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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				OOW_IF	WIN_IF	CMP2IF	CMP1IF
位权限	U-0				R/W-0	R/W-0	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	OOW_IF	Out-of-Window 中断标志，硬件置位，软件写 1 清零
2	WIN_IF	Window 中断标志，硬件置位，软件写 1 清零

位号	助记符	功能描述
1	CMP2IF	比较器 2 中断标志，硬件置位，软件写 1 清零
0	CMP1IF	比较器 1 中断标志，硬件置位，软件写 1 清零

16.4.5 COMP 功耗控制寄存器(COMP_PCR)

名称	COMP_PCR							
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	-					HSCMP BUF_EN	HSCMP2 _EN	HSCMP1 _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	HSCMPBUF_EN	比较器 BUFFER 高功耗模式使能 0：关闭 1：使能
1	HSCMP2_EN	比较器 2 高功耗模式使能 0：关闭 1：使能
0	HSCMP1_EN	比较器 1 高功耗模式使能 0：关闭 1：使能

17 双线串行总线（I²C）

17.1 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 的主机模式，不支持多主机模式。

特点：

- 2 路独立 I2C 接口
- 支持主机模式，不支持多主机模式
- 传输速度支持 standard mode(100Kbps), fast mode(400Kbps)
- 支持 DMA

17.2 结构框图

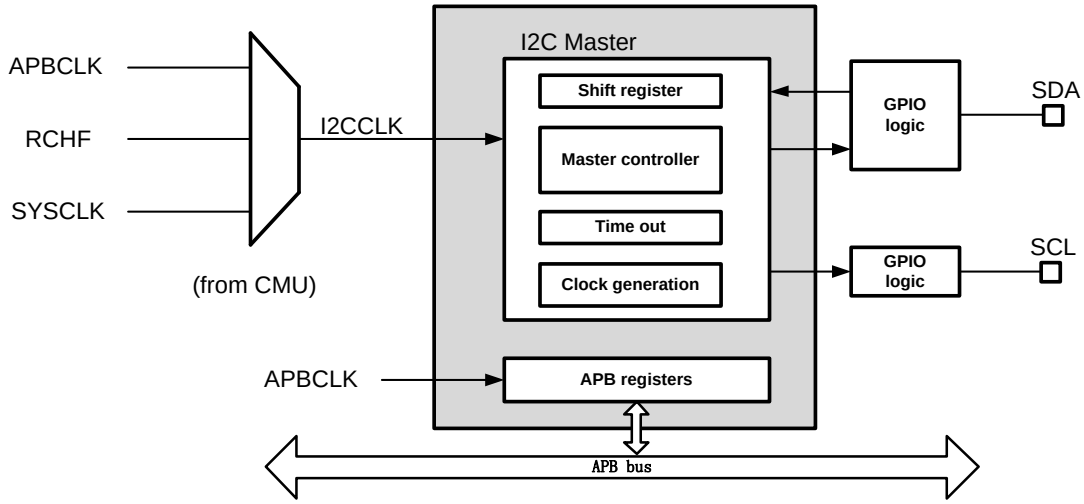


图 17-1 I2C 模块框图

17.3 引脚定义

I2C 模块使用 2 个引脚与外部器件通信，可以复用到 2 组 GPIO 上：

引脚	符号	I2Cx	功能
PA14	SCL0	I2C0	I2C 时钟
PA15	SDA0		I2C 数据
PE5	SCL1	I2C1	I2C 时钟
PE6	SDA1		I2C 数据

表 17-1 I2C 引脚定义

注意，当 I2C 与外部不同电源供电的外设通信时，需要将管脚的 ODEN 置位，配置为开漏模式，并外加上拉电阻，实现电平匹配。

17.4 时钟结构

I2C 主机采用了双时钟结构：

- 主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，必须通过置位 I2C_PCE 寄存器使能 PCLK。参见 9.12.11 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，当时钟选择为 RCHF 和 SYSCLK 时，I2C 能够独立于 APBCLK 工作。必须通过置位 I2CxCKE 寄存器使能 I2CCLK 才能进行数据收发。参见 9.12.13 外设独立工作时钟控制寄存器 1

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。

17.5 接口时序

17.5.1 接口时序图

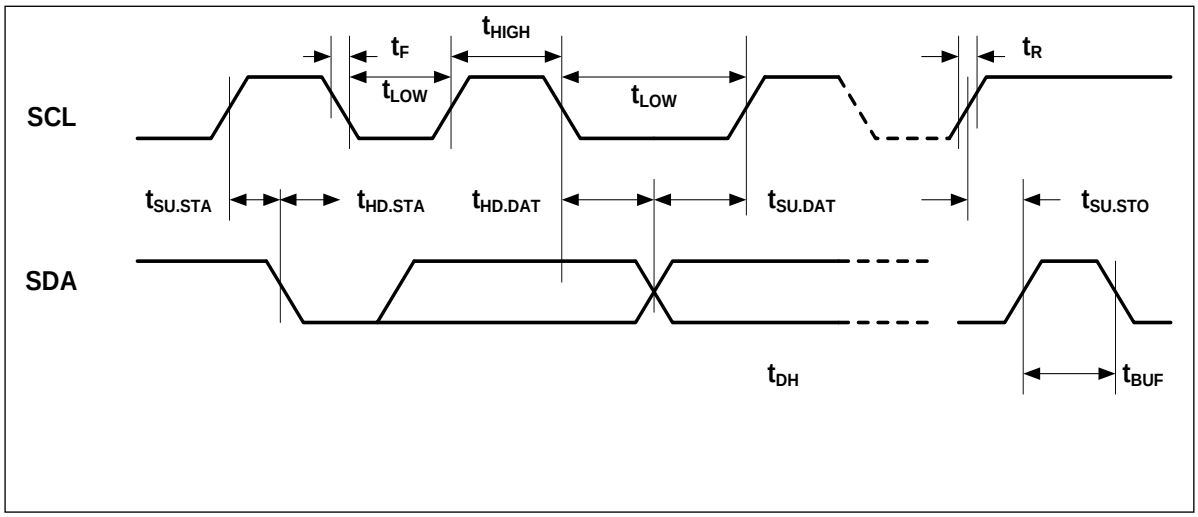


图 17-2I²C 总线时序

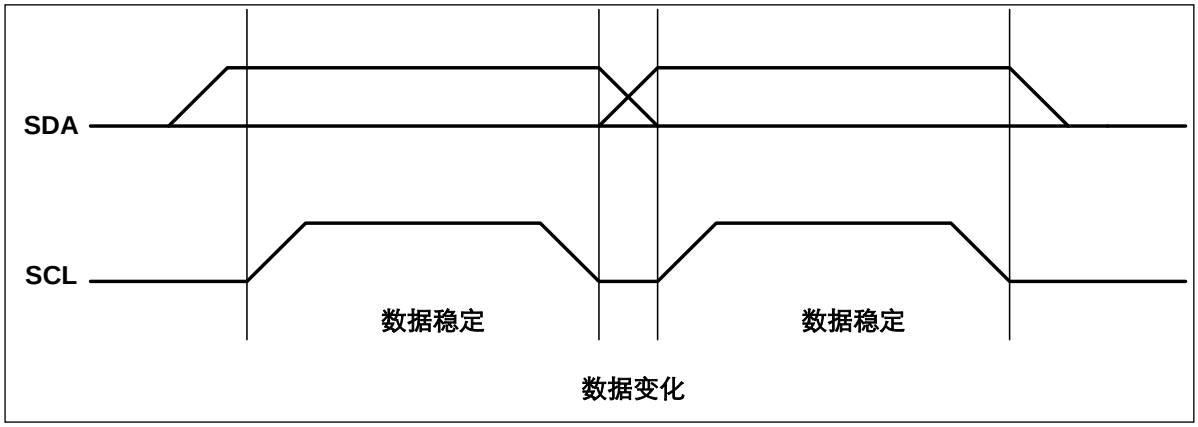


图 17-3 数据有效时序

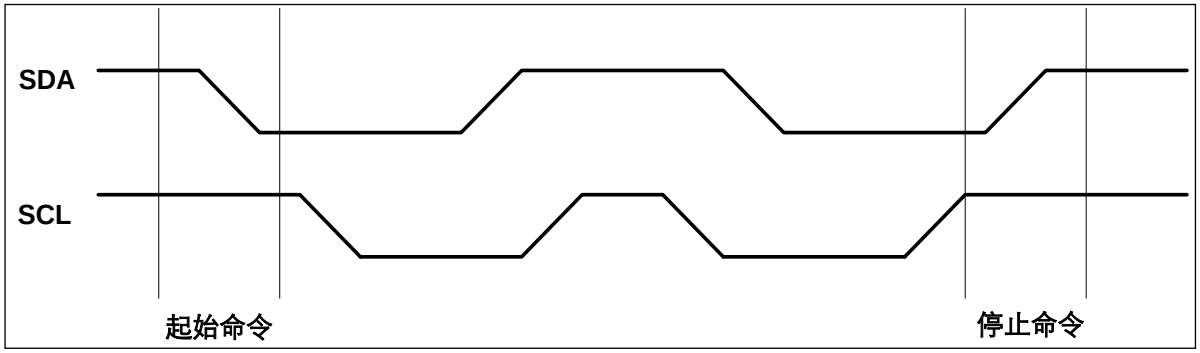


图 17-4 起始（Start）与停止(Stop)命令定义

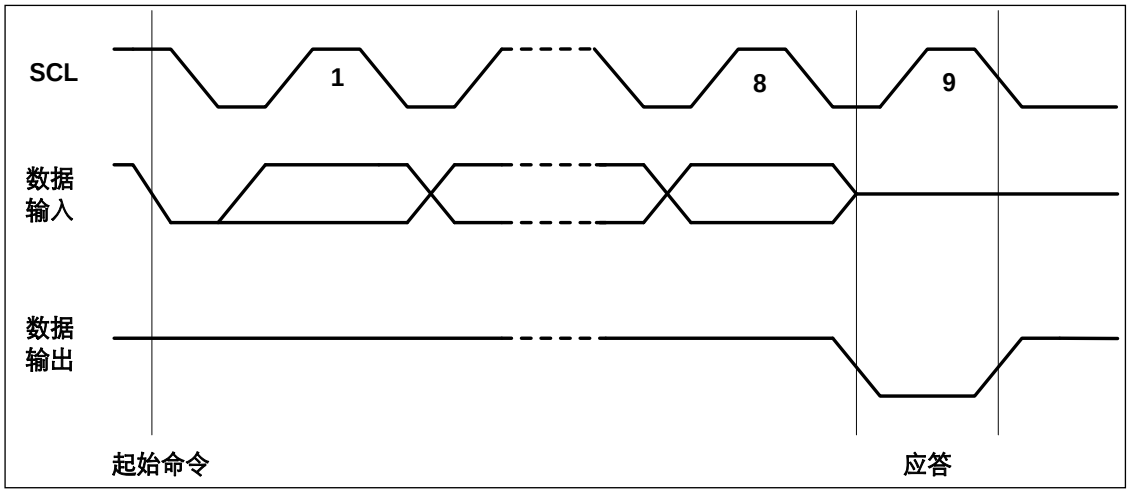


图 17-5 输出应答(ACK)

17.5.2 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 17-3）；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 17-4）。

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，停止命令会使外设进入等待态低功耗模式（参见图 17-4）。

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 17-5)。

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 17-2 I²C 接口时序要求

17.6 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RMU模块的I2CRST寄存器，确保I2C模块不处于复位状态
- 置位CMU模块的I2Cx_PCE寄存器，使能I2C模块寄存器总线接口时钟
- 配置CMU模块的I2CxCKS和I2CxCKE寄存器，选择并使能I2C工作时钟

17.6.1 IO 配置

FM33A0xxEV 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：PA15/PE6

SCL：PA14/PE5

17.6.2 主机波特率配置

I2C 主机需要在使能前配置通信波特率。

CPU 通过配置 MSPBRG 寄存器确定波特率时钟 SCL 对系统时钟的分频值，计算公式如下：

TBRGH＝TOSC×(MSPBGRH[8:0]+1)；TOSC为主时钟周期

TBRGL＝TOSC×(MSPBGRL[8:0]+1)；TOSC为主时钟周期

TSCL＝TBRGH+ TBRGL

17.7 I²C 主机功能

FM33A0xxEV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。

17.7.1 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。

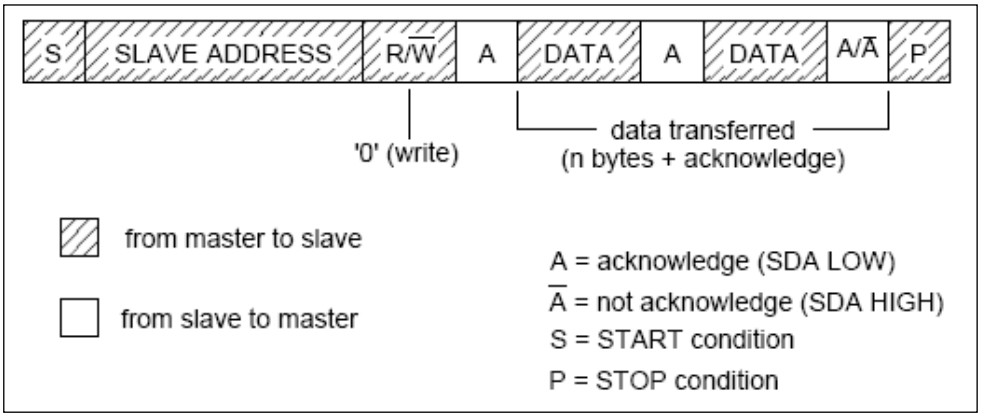


图 17-6 主机向 7 位地址从机写入数据时的帧格式

1、主机发起 START 时序

2、主机发送从机地址，从机地址包含 7 位从机地址和 1 位 R/W 标志位，发送数据时 R/W 位为 0

4、主机发送第一帧 8 位数据

5、主机在每次发送完 8 位数据后，会在第 9 个 SCL 判断是否检测到有效的 ACK，如果主机检测到 ACK 成功后，会继续输出下一字节数据

6、若从机无法响应 ACK，主机检测到 NACK 后应发送 STOP 时序终止发送

7、主机完成所有数据发送后，发送 STOP 时序

软件启动 I2C 主机发送的操作流程如下图：

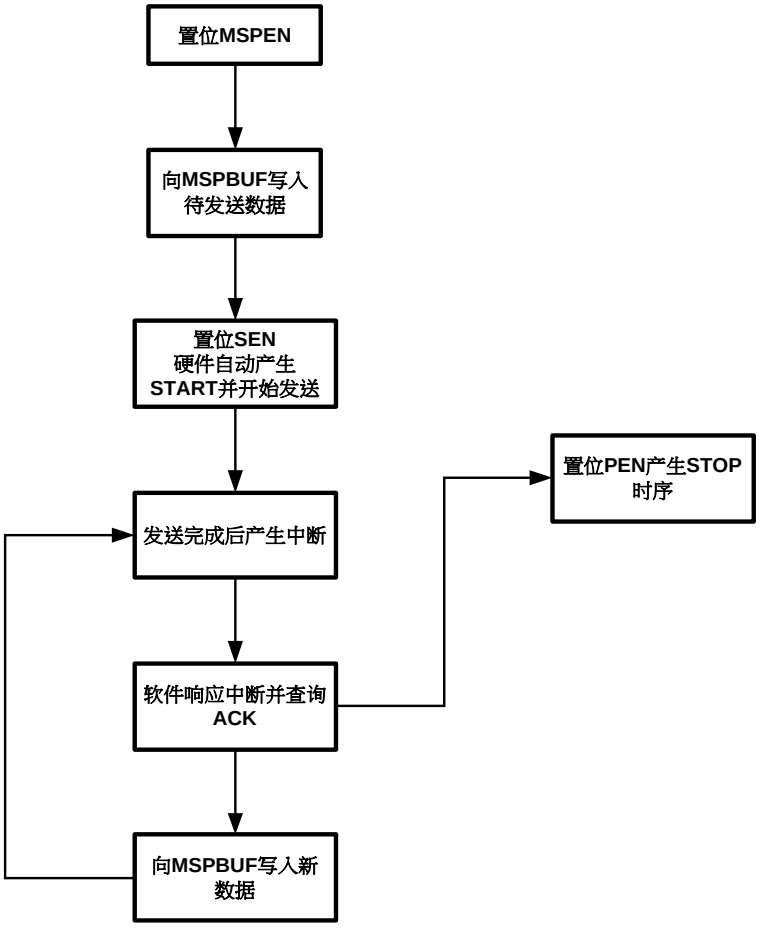


图 17-7 I2C 软件发送数据流图

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。

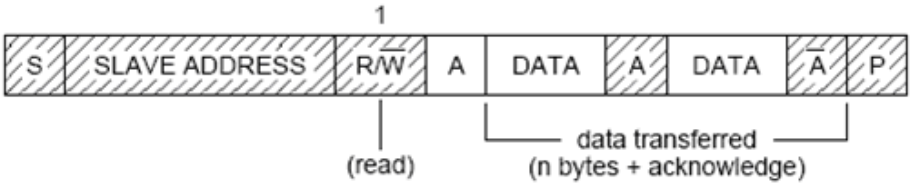


图 17-8 主机从 7 位地址从机读取数据时的帧格式

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：

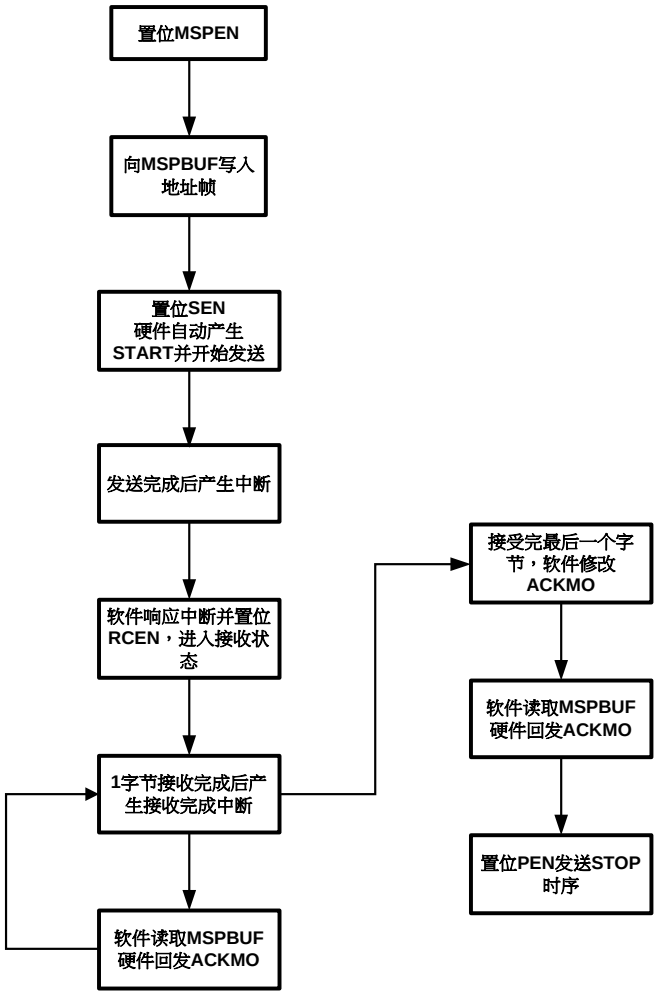


图 17-9 I2C 软件发送数据流图

主机每次接收完从机发送的数据后，根据ACKMO寄存器回发响应。ACKMO复位值为0，即默认状态下主机回发ACK。如果软件希望主机在接收完成后回发NACK，则需要在前一个字节接收完成中断中将ACKMO寄存器改写为1。ACKMO为1的情况下，主机在发送完响应后会自动清零ACKMO。

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。

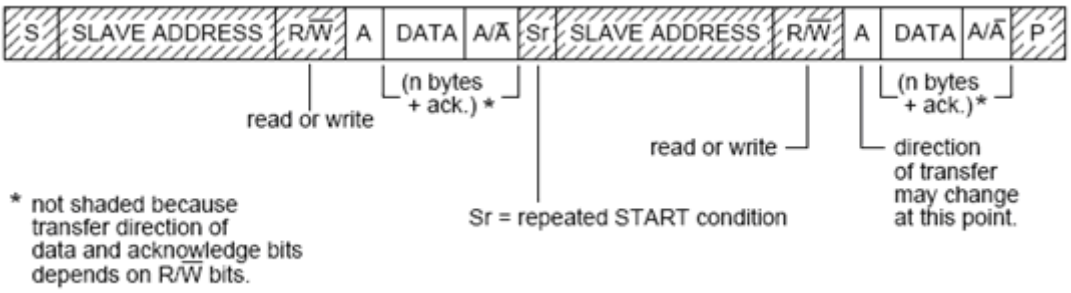


图 17-10 双向数据通信帧格式

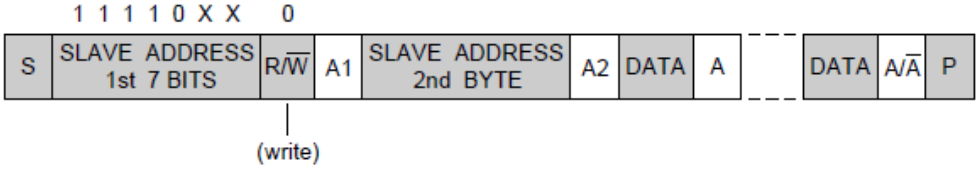
组合传输的软件操作流程与单向传输类似，只是在某个字节收发完成后，通过发送ReSTART时序和从机地址字节来修改传输方向。

17.7.2 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。



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：

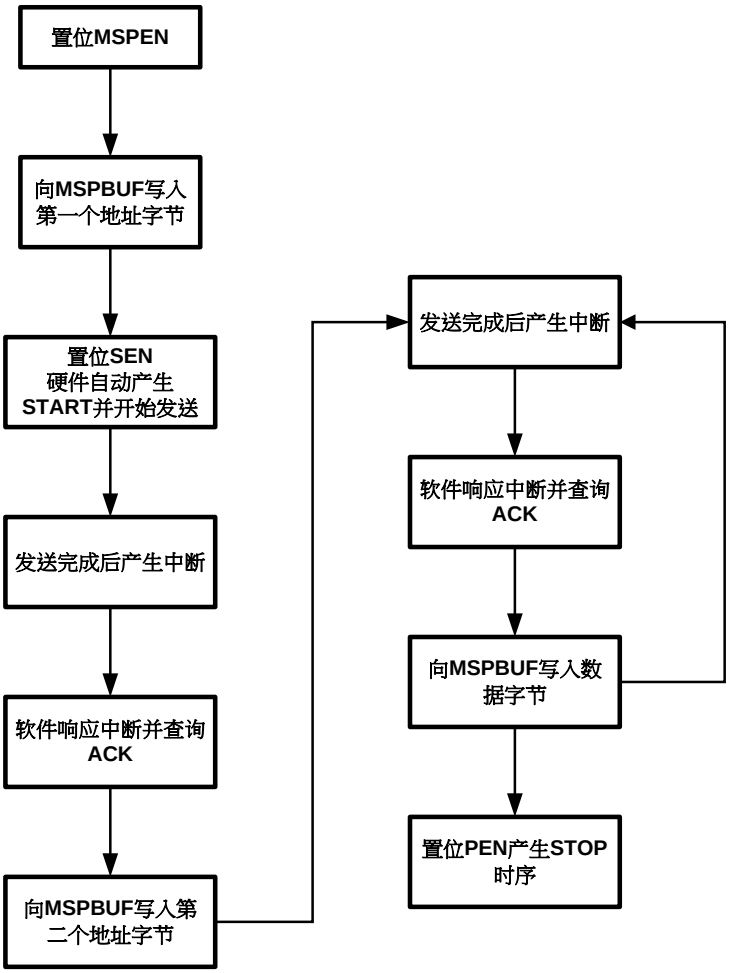
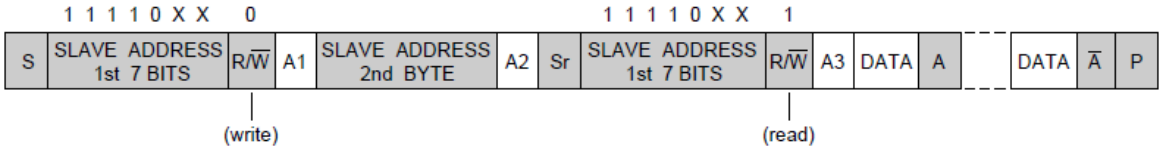


图 17-11 I2C 软件发送数据流图

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。



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：

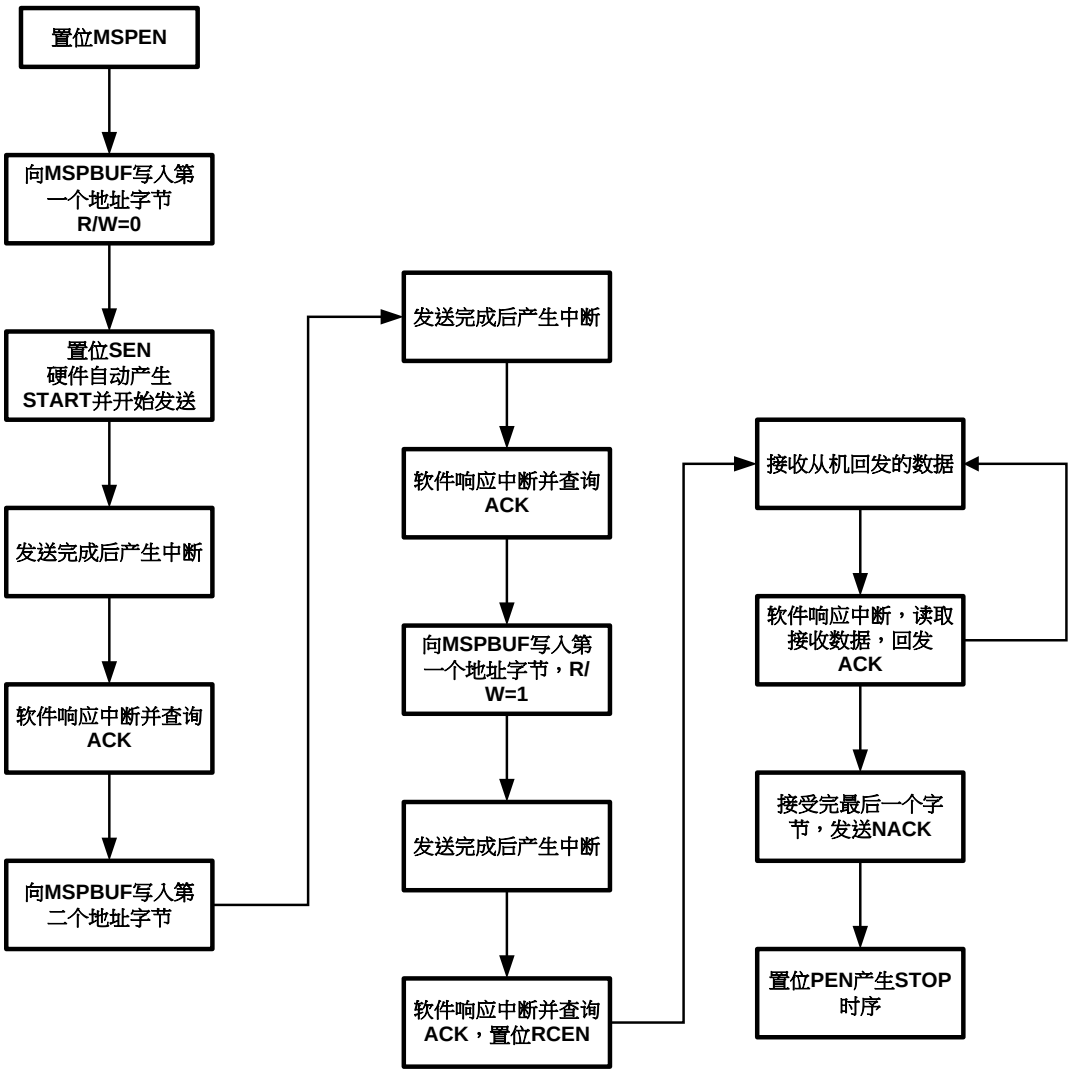
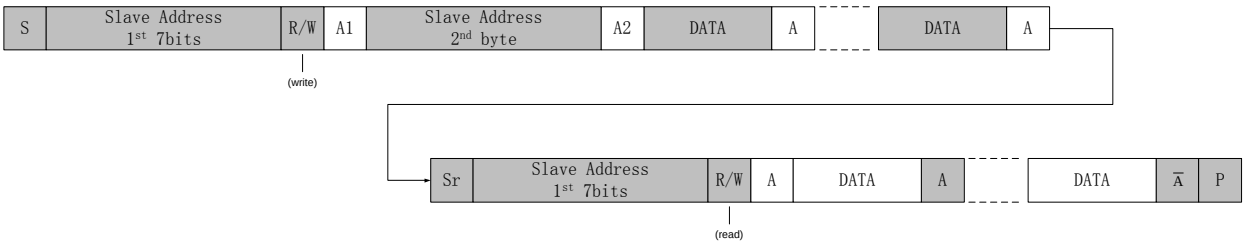


图 17-12 I2C 软件发送数据流图

双向数据传输（组合模式）

典型的双向数据读写流图如下图所示。在主机发送或读取数据过程中，主机可以通过发送 Repeated Start 时序来重新启动一次新的发送或读取通信，所以主机在一次通信中，即可以有数据发送也可以有数据读取。



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。

17.7.3 DMA

I2C 主机支持 DMA，需要注意的是，必须在 I2C 模块的总线时钟（PCLK）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：

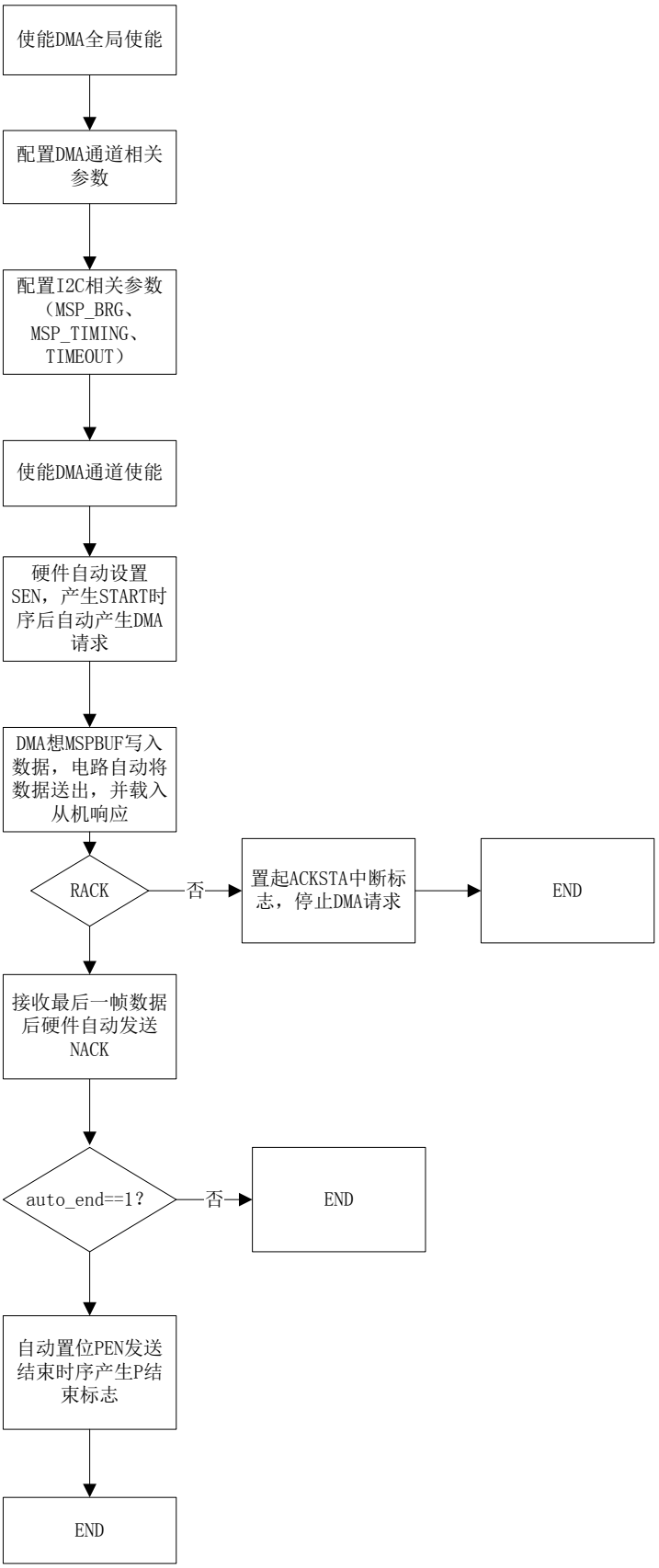


图 17-13 I2C 主机 DMA 发送流程图

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，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。

主机使用DMA进行接收的流程如下图：

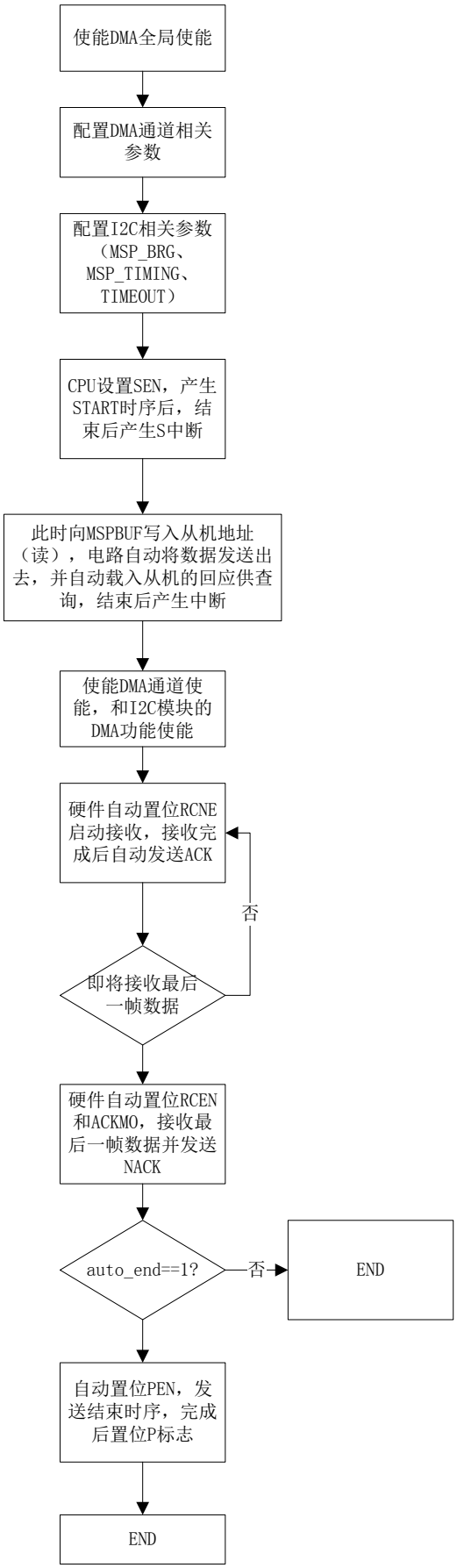


图 17-14 I2C 主机 DMA 接收流程图

17.7.4 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 主机通过支持这一特性来提高对低速外设的兼容性。在每个字节收发起始位置处，主机在尝试发送 SCL 高电平后，会自动检查总线上 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。

17.7.5 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。

17.7.6 可编程时序和波特率发生

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。

下图是主机关键时序示意图，其中 SCL 高电平宽度由 MSPBRGH 寄存器定义，SCL 低电平宽度由 MSPBRGL 寄存器定义，SDA 相对于 SCL 下降沿的保持时间由 SDAHD 寄存器定义：

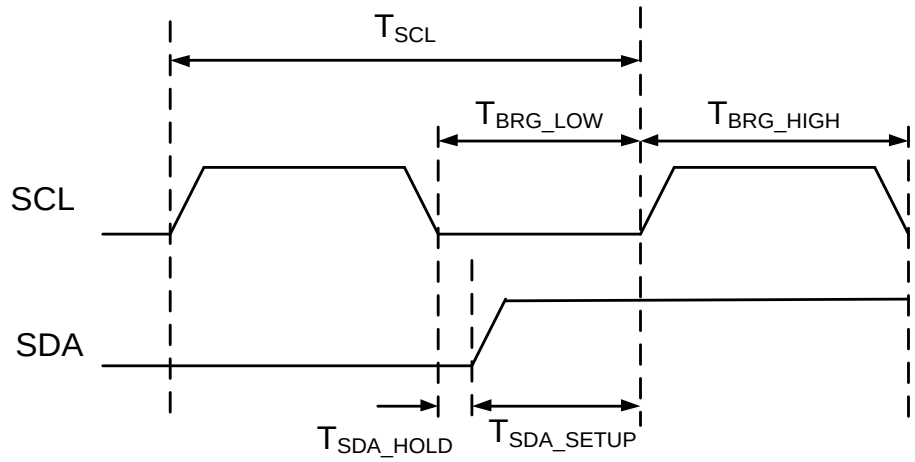


图 17-15 主机时序控制

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

17.8 寄存器

I2C0模块起始地址：0x4001_2400

I2C1模块起始地址：0x4001_5000

offset 地址	名称	符号
I2C0 寄存器(模块起始地址：0x40012400)		
0x00	I2C0 主机配置寄存器 (I2C0 Master Config Register)	I2C0_CFGR
0x04	I2C0 主机控制寄存器 (I2C0 Master Control Register)	I2C0_CR
0x08	I2C0 主机中断使能寄存器 (I2C0 Master Interrupt Enable Register)	I2C0_IER
0x0C	I2C0 主机中断标志寄存器 (I2C0 Master Interrupt Status Register)	I2C0_ISR
0x10	I2C0 主机状态寄存器 (I2C0 Master Status Register)	I2C0_SR
0x14	I2C0 主机波特率寄存器 (I2C0 Master Baud Rate Generation Register)	I2C0_BRG
0x18	I2C0 主机收发缓存寄存器 (I2C0 Master Buffer Register)	I2C0_BUF
0x1C	I2C0 主机时序控制寄存器 (I2C0 Master Timing Register)	I2C0_TIMING
0x20	I2C0 主机超时寄存器 (I2C0 Master Time-Out Register)	I2C0_TO
I2C1 寄存器(模块起始地址：0x40015000)		
0x00	I2C1 主机配置寄存器 (I2C1 Master Config Register)	I2C1_CFGR
0x04	I2C1 主机控制寄存器 (I2C1 Master Control Register)	I2C1_CR
0x08	I2C1 主机中断使能寄存器 (I2C1 Master Interrupt Enable Register)	I2C1_IER
0x0C	I2C1 主机中断标志寄存器 (I2C1 Master Interrupt Status Register)	I2C1_ISR
0x10	I2C1 主机状态寄存器 (I2C1 Master Status Register)	I2C1_SR
0x14	I2C1 主机波特率寄存器 (I2C1 Master Baud Rate Generation Register)	I2C1_BRG
0x18	I2C1 主机收发缓存寄存器 (I2C1 Master Buffer Register)	I2C1_BUF
0x1C	I2C1 主机时序控制寄存器 (I2C1 Master Timing Register)	I2C1_TIMING
0x20	I2C1 主机超时寄存器 (I2C1 Master Time-Out Register)	I2C1_TO

17.8.1 I2C 主机配置寄存器 (I2Cx_CFGR)

名称	I2Cx_CFGR(x=0,1)
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	-						AUTOEND	MSP_DMAEN
位权限	U-0						R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						TOEN	MSPEN
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:18	-	RFU：未实现，读为 0
17	AUTOEND	主机 DMA 自动终止 (Automatic Ending) 1：DMA 指定长度传输完成后，自动发送 STOP 时序 0：DMA 指定长度传输完成后，等待软件接管
16	MSP_DMAEN	主机 DMA 使能 (Master DMA Enable) 0：关闭 DMA 功能 1：使能 DMA 功能
15:2	-	RFU：未实现，读为 0
1	TOEN	SCL 拉低超时使能（TimeOut） 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

17.8.2 I2C 主机控制寄存器 (I2Cx_CR)

名称	I2Cx_CR(x=0,1)							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
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-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

位号	助记符	功能描述
3	RCEN	主控接收模式下，接收使能位 (Receive Enable) 1：主机接收使能 0：接收禁止
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 (ReStart 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)

17.8.3 I2C 主机中断使能寄存器 (I2Cx_IER)

名称	I2Cx_IER(x=0,1)							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	WCOLE	TOE	SE	PE	NACKE	TXIE	RXIE
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6	WCOLE	WCOL 中断使能寄存器 (Write Collide Enable) 1：允许写冲突中断 0：禁止写冲突中断
5	TOE	SCL 超时中断使能寄存器 (Time-Out Enable) 1：允许超时中断 0：禁止超时中断
4	SE	START 时序中断使能寄存器 (START interrupt Enable) 1：允许 START 时序中断 0：禁止 START 时序中断
3	PE	STOP 时序中断使能寄存器 (STOP interrupt Enable) 1：允许 STOP 时序中断 0：禁止 STOP 时序中断
2	NACKE	主机发送模式下 NACK 中断使能寄存器 (Non-ACK interrupt Enable) 1：允许收到 NACK 产生中断 0：禁止产生 NACK 中断
1	TXIE	I2C 主机发送完成中断使能 (Transmit 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)

位号	助记符	功能描述
		1：允许接收完成中断 0：禁止接收完成中断

17.8.4 I2C 主机中断标志寄存器 (I2Cx_ISR)

名称	I2Cx_ISR(x=0,1)							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	WCOL	TO	S	P	ACKSTA	TXIF	RXIF
位权限	U-0	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
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 SSPBUF，否则发生写冲突；硬件置位，软件写 1 清零 (Write Collide) 1 = 发送写冲突 0 = 未发生冲突
5	TO	SCL 超时中断标志，仅在 TOEN 为 1 时工作 (Time Out)，软件写 1 清零 1：发生 SCL 超时 0：没有发生 SCL 超时
4	S	START 时序发送完成中断标志，硬件置位，软件读取后清零 (START done)
3	P	STOP 时序发送完成中断标志，硬件置位，软件读取后清零 (STOP done)
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) 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 (Transmit done interrupt flag)
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)

17.8.5 I2C 主机状态寄存器 (I2Cx_SR)

名称	I2Cx_SR(x=0,1)
offset	0x10

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	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	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5	BUSY	I2C 通信状态位 (I2C is 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 or Write Bar) 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，SSPBUF 满 0：接收未完成，SSPBUF 空 发送： 1：正在发送，SSPBUF 满 0：发送完成，SSPBUF 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 (Acknowledge mode) 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***

17.8.6 I2C 主机波特率设置寄存器 (I2Cx_BRG)

名称	I2Cx_BRG(x=0,1)							
offset	0x14							
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	助记符	功能描述
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 (Baud rate generation HIGH level)
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 (Baud rate generation LOW level)

17.8.7 I2C 主机收发缓冲寄存器 (I2Cx_BUF)

名称	I2Cx_BUF(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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSPBUF							
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	MSPBUF	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。 (Master Buffer)

17.8.8 I2C 主机时序控制寄存器 (I2Cx_TIMING)

名称	I2Cx_TIMING(x=0,1)							
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	-							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	助记符	功能描述
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 time)

17.8.9 I2C 主机超时寄存器 (I2Cx_TO)

名称	I2Cx_TO(x=0,1)							
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					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	助记符	功能描述
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}

18 通用异步收发传输器（UART）

18.1 概述

UART串行通信模块特点如下

- 波特率软件可配置
- 6路独立通道（UART0, UART1, UART2, UART3, UART4, UART5）
- 全双工通信口
- 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
- 可交换的RX和TX引脚
- 支持DMA
- 支持接收超时机制

18.2 结构框图

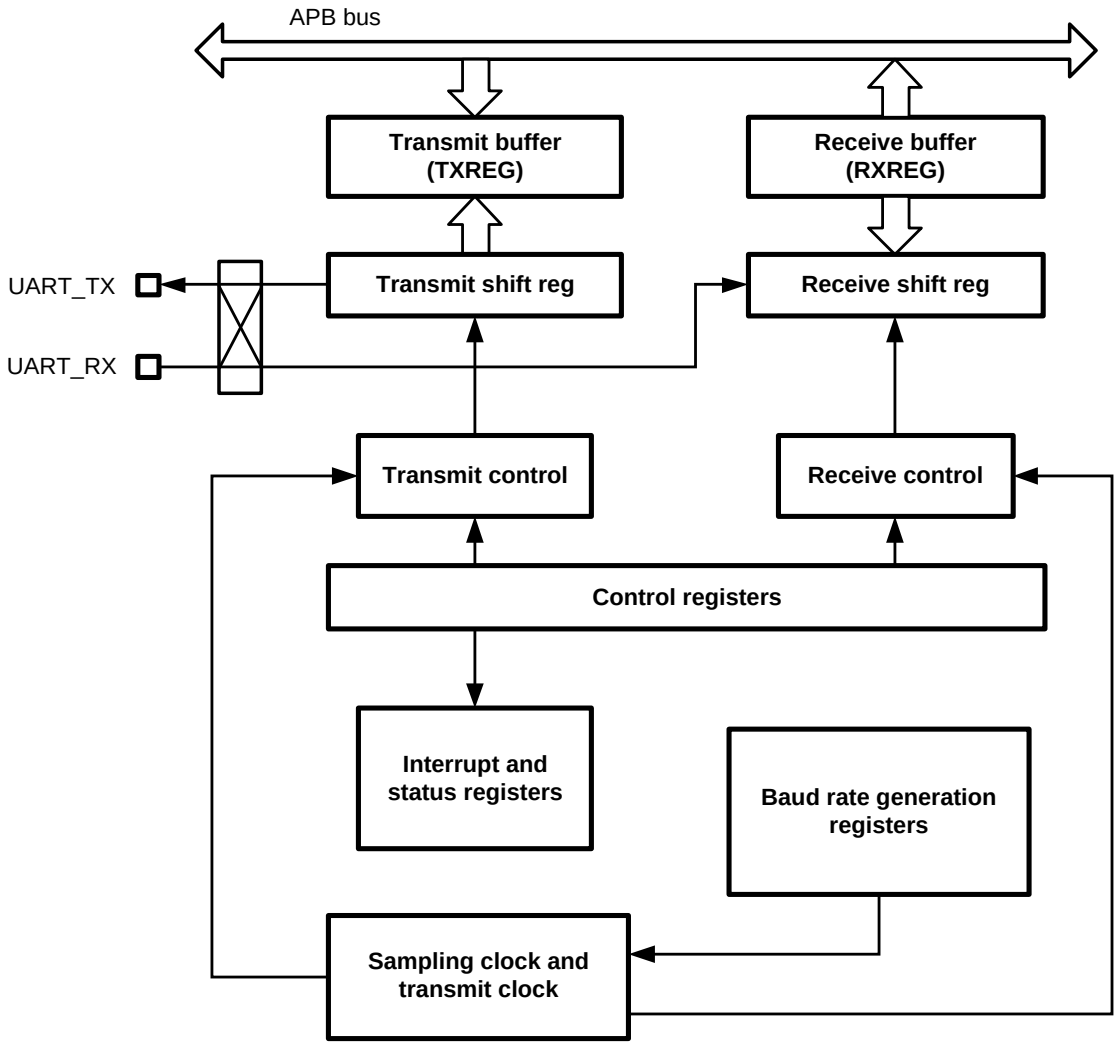


图 18-1 UART 结构框图

18.3 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 UART 引脚映射关系

引脚		UARTx	符号	功能
PF3	PC2	UART0	UART0_RX	数据接收
PF4	PC3		UART0_TX	数据发送
PB0	PE3	UART1	UART1_RX	数据接收
PB1	PE4		UART1_TX	数据发送
PB2	PB12	UART2	UART2_RX	数据接收
PB3	PB13		UART2_TX	数据发送
PB14	PC10	UART3	UART3_RX	数据接收
PB15	PC11		UART3_TX	数据发送
PD0	PD9	UART4	UART4_RX	数据接收
PD1	PD10		UART4_TX	数据发送
PA8	PC4	UART5	UART5_RX	数据接收
PA9	PC5		UART5_TX	数据发送

表 18-1 UART 引脚定义

接收数据的管脚优先级：

外设功能		高优先级->低优先级	
UART5_RX	IOSWAP=0	PC4	PA8
	IOSWAP=1	PC5	PA9
UART4_RX	IOSWAP=0	PD9	PD0
	IOSWAP=1	PD10	PD1
UART3_RX	IOSWAP=0	PC10	PB14
	IOSWAP=1	PC11	PB15
UART2_RX	IOSWAP=0	PB12	PB2
	IOSWAP=1	PB13	PB3
UART1_RX	IOSWAP=0	PE3	PB0
	IOSWAP=1	PE4	PB1
UART0_RX	IOSWAP=0	PG8	PF3
	IOSWAP=1	PG9	PF4

表 18-2 UART 接收引脚优先级

18.4 UART 类型区分

FM33A0xxEV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	UART2/3/4/5	LPUART0/1
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 18-3 UART 类型区分

18.5 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。

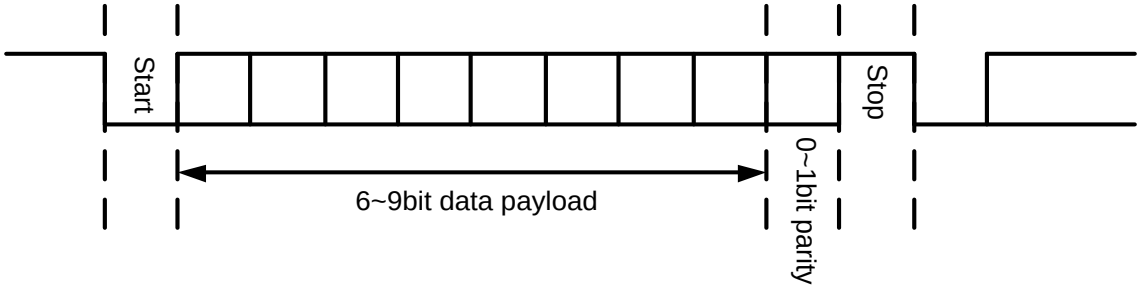


图 18-2 UART 字符描述

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 18-4 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】。

18.6 功能描述

18.6.1 时钟结构

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、RC4M，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，UATR2、UART3、UART4、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。

18.6.2 位接收采样

UART 对接收数据进行波特率的 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。

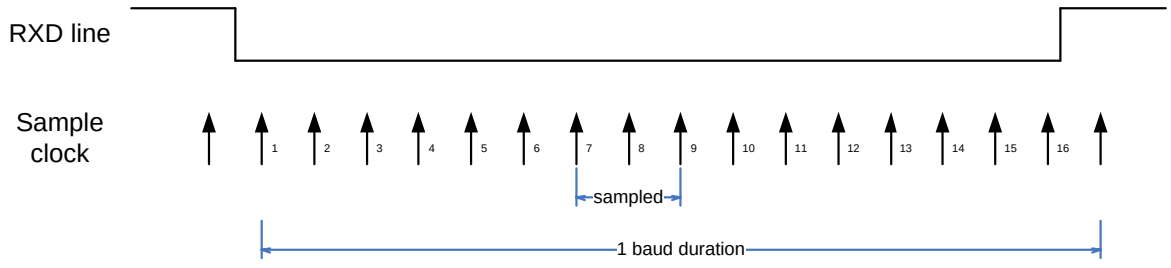


图 18-3 位接收采样

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。

由于 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。

18.6.3 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。

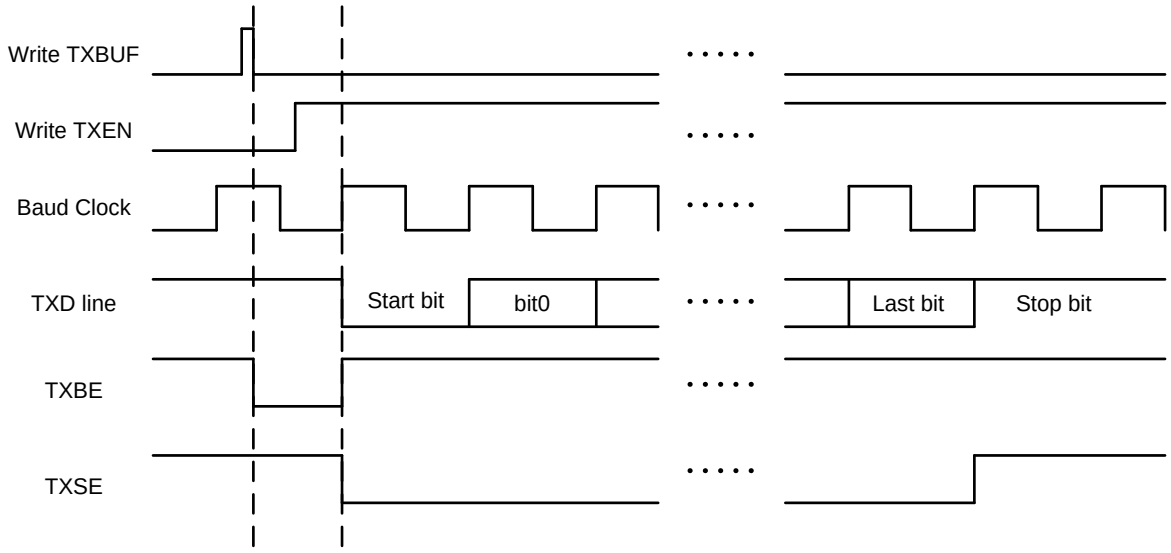


图 18-4 UART 异步发送波形 1

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。

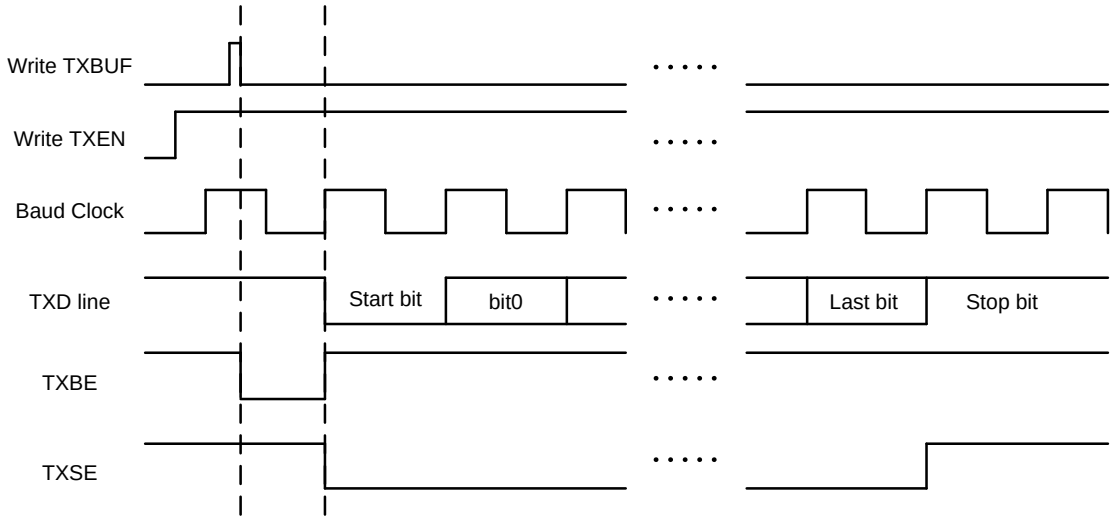


图 18-5 UART 异步发送波形 2

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。

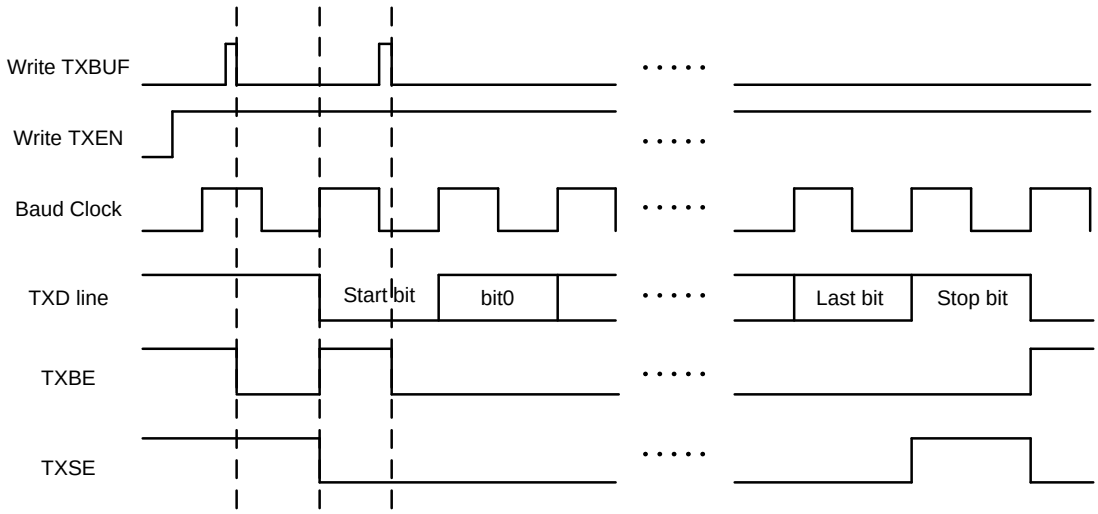


图 18-6 UART 异步发送波形 3

18.6.4 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

18.6.5 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

18.6.6 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UART 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
- 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 18-5 UART 使用 DMA 发送时的中断标志置位

18.7 波特率发生

18.7.1 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 18-6 常用时钟频率下波特率计算

18.7.1 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。

18.8 红外调制

UART_IRCR 寄存器保存一个 11 位的分频系数 TZBRG，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：

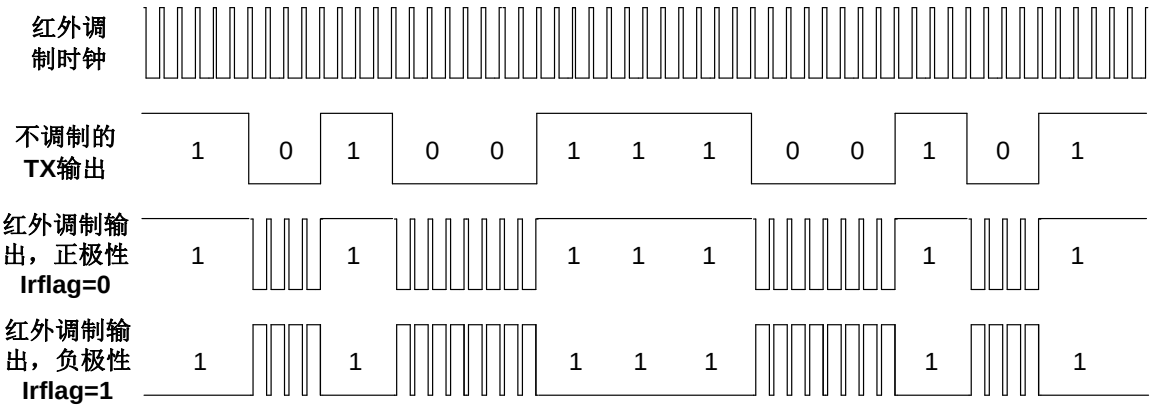


图 18-7 红外调制波形

无论有效电平是 0 还是 1，占空比定义为高电平长度/周期。

18.9 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：UART2/3/4/5 不支持接收超时功能。

18.10 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。

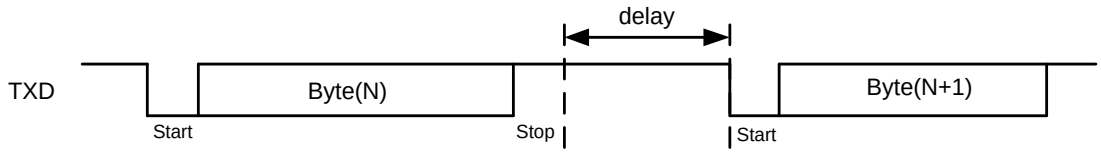


图 18-8 UART 发送延迟

注：UART2/3/4/5 不支持发送延迟功能。

18.11 寄存器

UART0模块起始地址：0x4001_2000

UART1模块起始地址：0x4001_6800

UART2模块起始地址：0x4001_6C00

UART3模块起始地址：0x4001_7000

UART4模块起始地址：0x4001_7400

UART5模块起始地址：0x4001_7800

UARTIR模块起始地址：0x4001_7C00

offset 地址	名称	符号
UARTIR 寄存器(模块起始地址：0x40017C00)		
0x00	红外调制寄存器 （Infrared modulation Control Register）	UART_IRCR
UART0 寄存器(模块起始地址：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(模块起始地址：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
UART2 寄存器(模块起始地址：0x40016C00)		
0x00	UART2 控制状态寄存器 （UART2 Control Status Register）	UART2_CSR
0x04	UART2 中断使能寄存器 （UART2 Interrupt Enable Register）	UART2_IER

offset 地址	名称	符号
0x08	UART2 中断标志寄存器 （UART2 Interrupt Status Register）	UART2_ISR
0x10	UART4 接收缓冲寄存器 （UART2 Receive Buffer）	UART2_RXBUF
0x14	UART2 发送缓冲寄存器 （UART2 Transmit Buffer）	UART2_TXBUF
0x18	UATR2 波特率产生寄存器 （UART2 Baud rate Generator Register）	UART2_BGR
UART3 寄存器(模块起始地址：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
UART4 寄存器(模块起始地址：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(模块起始地址：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

18.11.1 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-1101 0010							

位号	助记符	功能描述
31:16	**-**	**未实现：读为0**
15	**IRFLAG**	控制红外调制发送数据时的默认输出极性 (Infra Red Flag) 0：正极性 1：负极性
14:11	**TH**	红外占空比调制参数 (Tranmission High Duty)
10:0	**TZBRG**	红外调制频率 (Transmission Baud Rate)

18.11.1 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	-			IOSWAP	-	DMATXIFCFG	BITORD	STOPCFG
位权限	U-0			R/W-0	U-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PDSEL		PARITY		RXPOL	TXPOL	RXEN	TXEN
位权限	R/W-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

位号	助记符	功能描述
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 1：使能接收超时功能 0：关闭接收超时功能
15:13	**-**	未实现：读为0
12	**IOSWAP**	RX 和 TX 引脚交换 0：默认引脚顺序（与封装图一致） 1：交换引脚顺序
11	**-**	未实现：读为0
10	**DMATXIFCFG**	DMA发送完成中断使能，仅在UART通过DMA进行发送时有效 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

18.11.2 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	-						TXBE_IE	TXSE_IE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:12	**-**	**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
9	**-**	**未实现：读为0**
8	**RXBF_IE**	接收缓存满中断使能，1 有效
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

18.11.3 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	-						TXBE	TXSE
位权限	U-0						R/W-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

位号	助记符	功能描述
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
7:2	**-**	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

18.11.4 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

18.11.5 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]

18.11.6 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							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	发送数据缓冲寄存器数据

7位收发时，发送的7bits数据写入TXBUF[6:0]

18.11.7 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18.7.1波特率发生

注：当SPBRG <= 0x000F时，UARTDIV=16’H000F;

当SPBRG >0x000F时，UARTDIV=SPBRG;

19 低功耗 UART（LPUART）

19.1 概述

LPUART 是低功耗异步串行通信接口，当工作时钟选择为 XTLF 或者 RCLP 时，可以支持到最高 9600 波特率的数据收发，LPUART 功耗极低，可以在 Sleep/DeepSleep 模式下工作。

特点：

- 异步数据收发
- 2路独立LPUART（LPUART0, LPUART1）
- 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RCLP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）

19.2 结构框图

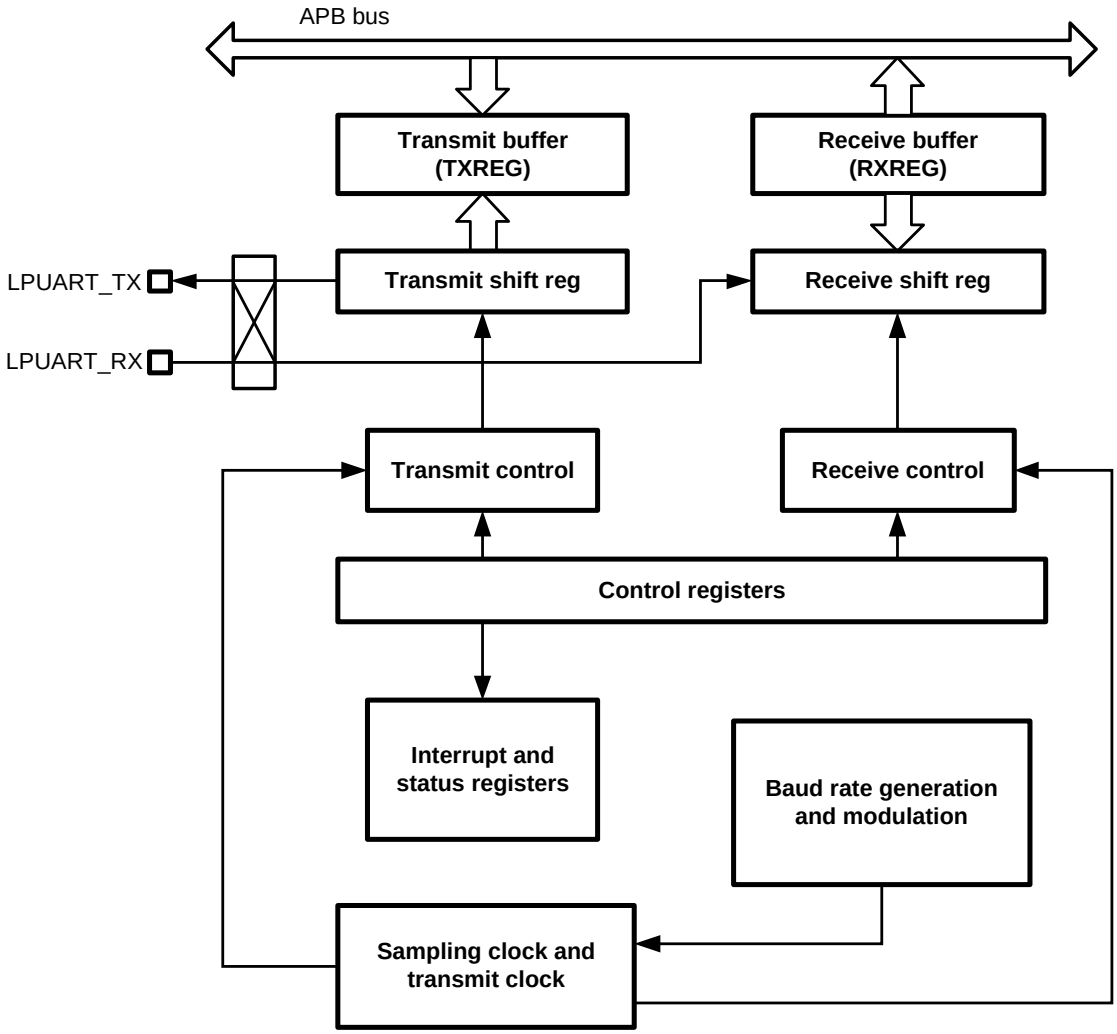


图 19-1 LPUART 结构框图

19.3 工作时钟

LPUART 使用独立于 APBCLK 的时钟进行数据收发，工作前需要在 CMU 模块中配置相关寄存器。

LPUART 可以使用 XTLF 或者 RCLP 工作。当系统上有 32768Hz 晶体时，LSCLK 为 XTLF 时钟，当系统上没有 32768Hz 晶体时，LSCLK 可以使用 RCLP 时钟。

在 ACTIVE 模式下，LPUART 也可以使用 RCHF 工作。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，具体请参考 9 时钟管理单元（CMU）。

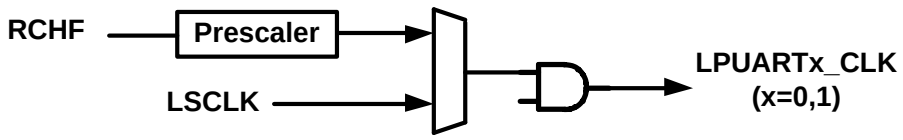


图 19-2 LPUART 工作时钟

19.4 引脚定义

LPUART 模块使用 2 个引脚与外部器件通信，引脚映射关系如下。

引脚		UARTx	符号	功能
PF3	PC0	LPUART0	LPUART0_RX	数据接收
PF4	PC1		LPUART0_TX	数据发送
PG2	PG4	LPUART1	LPUART1_RX	数据接收
PG3	PG5		LPUART1_TX	数据发送

表 19-1 LPUART 引脚定义

接收数据的管脚优先级：

外设功能		高优先级->低优先级	
LPUART0_RX	IOSWAP=0	PF3	PC0
	IOSWAP=1	PF4	PC1
LPUART1_RX	IOSWAP=0	PG4	PG2
	IOSWAP=1	PG5	PG3

表 19-2 LPUART 接收引脚优先级

19.5 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。

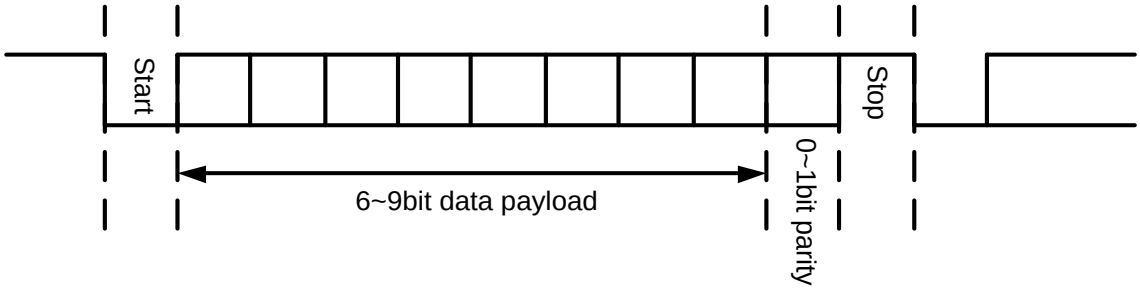


图 19-3 字符描述

LPUART 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 19-3 LPUART 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】。

19.6 功能描述

19.6.1 位接收采样

低功耗串口工作时钟频率仅为 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。

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，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
9600	0	1	0	0	1	0	1	0	1	0	0	1
4800	1	1	0	1	1	1	1	1	0	1	1	1
2400	1	1	0	1	1	0	1	1	0	1	1	0
1200	0	1	0	0	1	0	0	1	0	0	1	0
600	0	1	1	0	1	0	1	1	0	1	1	0
300	0	1	0	0	0	0	1	0	0	0	0	1

表 19-4 LPUART 位调制系数

19.6.2 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

19.6.3 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

19.6.4 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 LPUART0 接收

- 将 DMA 通道 x 配置为 LPUART0_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 LPUART0 模块参数
- 使能 LPUART0 模块接收使能 LPUEN.RXEN=1，等待数据接收
- 收到数据后 LPUART0 自动产生 DMA 请求
- DMA 响应请求，读取 LPUART0 接收缓存寄存器，写入指定 RAM 地址

19.6.5 休眠模式下的数据接收唤醒

LPUART 在低功耗串口模式下支持在 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CMU将EUART工作时钟选择为LSCLK
- 配置波特率
- 根据波特率选择合适的调制参数，配置MCTL寄存器
- 配置帧格式、极性，通过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RXEN寄存器打开接收使能
- 软件进入Sleep/DeepSleep

19.6.6 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，保证芯片处于极低公好状态下。

- 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

19.6.7 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 19-5 LPUART 通过 DMA 发送时的中断标志产生

19.7 寄存器

LPUART0模块起始地址：0x4001_4000

LPUART1模块起始地址：0x4001_4400

offset 地址	名称	符号
LPUART0 寄存器(模块起始地址：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(模块起始地址：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

19.7.1 LPUARTx 控制状态寄存器（LPUARTx_CSR）

名称	LPUARTx_CSR(x=0,1)							
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	-				WKBYTECFG	-	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 (Busy) 1：LPUART 正在通信中 0：LPUART 空闲
23:20	-	未实现：读为0
19	**WKBYTECFG**	数据接收唤醒条件配置 (Wakeup Byte Config) 1：接收完1字节，并且奇偶校验和STOP位都正确，才触发唤醒中断 0：接收完1字节，不检查校验位和STOP位，直接触发唤醒中断
18	-	未实现：读为0
17:16	**RXEV**	唤醒中断事件配置，用于控制何种事件下向 CPU 提供唤醒中断 (Receive Wakeup Event) 00：START 位检测唤醒 01：1byte 数据接收完成 10：接收数据匹配成功 11：RXD 下降沿检测
15:12	-	未实现：读为0
11	**IOSWAP**	RX 和 TX 引脚交换 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 (DMA Transmit Interrupt Config) 1：IE=1的情况下，DMA模式下发送完最后一帧后，允许中断信号输出；最后一帧之前的数据帧发送完成后不允许中断信号输出 0：是否允许中断信号输出仅由IE决定
9	**BITORD**	数据发送/接收时的位顺序 (Bit Order) 0：LSB first 1：MSB first
8	**STOPCFG**	停止位宽度配置，仅对发送帧格式有效，接收时不判断停止位个数 (Stop bit Config) 0：1位停止位 1：2位停止位
7:6	**PDSEL**	每帧数据长度选择；此寄存器对数据发送和接收同时有效 (Payload Data length Select) 00：7 位数据 01：8 位数据 10：9 位数据 11：6 位数据

位号	助记符	功能描述
5:4	**PARITY**	校验位配置；此寄存器对数据发送和接收同时有效 (Parity) 00：无校验位 01：偶校验 10：奇校验 11：RFU
3	**RXPOL**	接收数据极性配置 (Receive Polarity) 0：正向 1：取反
2	**TXPOL**	发送数据极性配置 (Transmit Polarity) 0：正向 1：取反
1	**RXEN**	接收使能，1 有效 (Receive Enable)
0	**TXEN**	发送使能，1 有效 (Transmit Enable)

19.7.2 LPUARTx 中断使能寄存器（LPUARTx_IER）

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

位号	助记符	功能描述
31:13	**-**	未实现：读为0
12	**RXEV_IE**	接收唤醒事件中断使能，1 有效 (Receive Event Interrupt Enable)
11	**-**	未实现：读为0
10	**RXERR_IE**	接收错误中断使能，1 有效 (Receive Error Interrupt Enable)
9	**-**	未实现：读为0
8	**RXBF_IE**	接收缓存满中断使能，1 有效 (Receive Buffer Full Interrupt Enable)
7:2	**-**	未实现：读为 0
1	**TXBE_IE**	发送缓存空中断使能，1 有效 (Transmit Buffer Empty Interrupt Enable)
0	**TXSE_IE**	发送缓存空且发送移位寄存器空中断使能，1 有效 (Transmit Shift register Interrupt Enable)

19.7.3 LPUARTx 中断标志寄存器（LPUARTx_ISR）

名称	LPUARTx_ISR(x=0,1)							
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-1	R/W-0

位号	助记符	功能描述
31:25	-	未实现：读为0
24	RXEVF	接收唤醒事件中断标志，硬件置位，软件写 1 清零 (Receive Event Interrupt Flag) 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 (Transmit Overflow Error) 当发送缓存中的数据还未进入移位寄存器发送时，软件向发送缓存写入新数据，将触发TXOV标志置位。
18	PERR	奇偶校验错误中断标志，硬件置位，软件写 1 清零 (Parity Error)
17	FERR	帧格式错误中断标志，硬件置位，软件写 1 清零 (Frame Error)
16	OERR	接收缓存溢出错误中断标志，当接收缓存满的情况下，收到新的数据时置位；硬件置位，软件写 1 清零 (Receive Buffer Overflow Error)
15:9	-	未实现：读为0
8	RXBF	接收缓存满中断标志，硬件置位，软件写 1 或者读取 RXBUF 时清零 (Receive Buffer Full)
7:2	-	未实现：读为 0
1	TXBE	发送缓存空中断标志，硬件置位，写入 TXBUF 时清零 (Transmit Buffer Empty)
0	TXSE	发送缓存空且发送移位寄存器空中断标志，硬件置位，软件写 1 或者发送数据被载入移位寄存器时清零 (Transmit Shift register Empty)

19.7.4 LPUARTx 波特率调制寄存器（LPUARTx_BMR）

名称	LPUARTx_BMR(x=0,1)							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-			MCTL[12:8]				
位权限	U-0			R/W-0 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
31:29	-	未实现：读为0
28:16	**MCTL**	LPUART 每个 bit 的位宽调制控制信号 9600: MCTL= 0x552 4800: MCTL= 0x1EFB 2400: MCTL= 0x16DB 1200: MCTL= 0x492 600: MCTL= 0x16D6 300: MCTL= 0x842
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

19.7.5 LPUARTx 接收缓冲寄存器（LPUARTx_RXBUF）

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

位号	助记符	功能描述
31:9	-	**未实现：读为0**
8:0	**RXBUF**	接收数据缓存寄存器 (Receive Buffer)

19.7.6 LPUARTx 发送缓冲寄存器（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	W-0000 0000							

位号	助记符	功能描述
31:9	**-**	未实现：读为0
8:0	**TXBUF**	发送数据缓存寄存器（Transmit Buffer）

19.7.7 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**
8:0	**MATD**	第一帧接收比较数据，如果 RXEV=10，当接收到的第一帧数据与 MATD 相同时，触发 RXEVF 中断，可以用于休眠模式下的数据接收唤醒。 (Matched Data)

20 串行外设接口（SPI）

20.1 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 5 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）
- 5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

20.2 结构框图

下图为 SPI 模块的结构示意图。

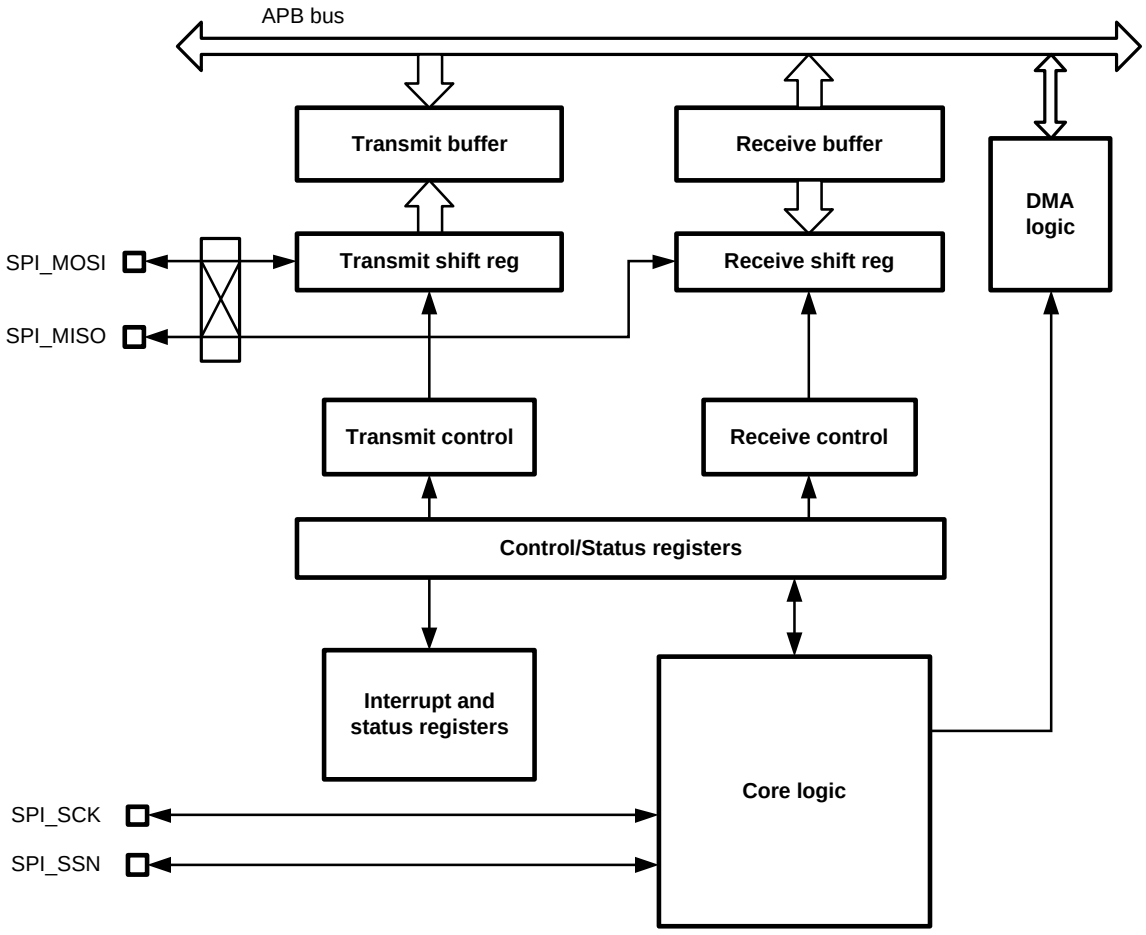


图 20-1 SPI 结构框图

20.3 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。5 组 SPI 中，3 组（SPI1/2/4）可以分别映射到两组 IO 使用，而 SPI0 和 SPI3 只能映射到一组 IO 使用。

引脚		SPIx	全双工	功能	半双工	功能
PA4	PB12	SPI1	SSN	片选信号	SSN	片选信号
PA5	PB13		SCLK	时钟	SCLK	时钟
PA6	PB14		MISO	主机输入从机输出	DCN	命令/数据标识
PA7	PB15		MOSI	主机输出从机输入	SDATA	双向数据
PA0	PD2	SPI2	SSN	片选信号	SSN	片选信号
PA1	PD3		SCLK	时钟	SCLK	时钟
PA2	PD4		MISO	主机输入从机输	DCN	命令/数据标识

				出		
PA3	PD5		MOSI	主机输出从机输入	SDATA	双向数据
PE8	PG6	SPI4	SSN	片选信号	SSN	片选信号
PE7	PD6		SCLK	时钟	SCLK	时钟
PE6	PD7		MISO	主机输入从机输出	DCN	命令/数据标识
PE5	PG7		MOSI	主机输出从机输入	SDATA	双向数据

引脚		SPIx	全双工	功能	半双工	功能
PF12	-	SPI0	SSN	片选信号	SSN	片选信号
PF13	-		SCLK	时钟	SCLK	时钟
PF14	-		MISO	主机输入从机输出	DCN	命令/数据标识
PF15	-		MOSI	主机输出从机输入	SDATA	双向数据
PC6	-	SPI3	SSN	片选信号	SSN	片选信号
PC7	-		SCLK	时钟	SCLK	时钟
PC8	-		MISO	主机输入从机输出	DCN	命令/数据标识
PC9	-		MOSI	主机输出从机输入	SDATA	双向数据

表 20-1 SPI 引脚定义

SPI 接收引脚优先级：

外设功能		高优先级->低优先级	
SPI0_MOSI	IOSWAP=0	PF12	-
	IOSWAP=1	PF13	-
SPI0_MISO	IOSWAP=0	PF13	-
	IOSWAP=1	PF12	-
SPI1_MOSI	IOSWAP=0	PB15	PA7
	IOSWAP=1	PB14	PA6
SPI1_MISO	IOSWAP=0	PB14	PA6
	IOSWAP=1	PB15	PA7
SPI2_MOSI	IOSWAP=0	PD5	PA3
	IOSWAP=1	PD4	PA2
SPI2_MISO	IOSWAP=0	PD4	PA2
	IOSWAP=1	PD5	PA3
SPI3_MOSI	IOSWAP=0	PC9	-
	IOSWAP=1	PC8	-
SPI3_MISO	IOSWAP=0	PC8	-
	IOSWAP=1	PC9	-
SPI4_MISO	IOSWAP=0	PE6	PD7
	IOSWAP=1	PG7	PE5
SPI4_MOSI	IOSWAP=0	PG7	PE5
	IOSWAP=1	PE6	PD7

表 20-2 SPI 引脚定义

20.4 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。

20.4.1 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。

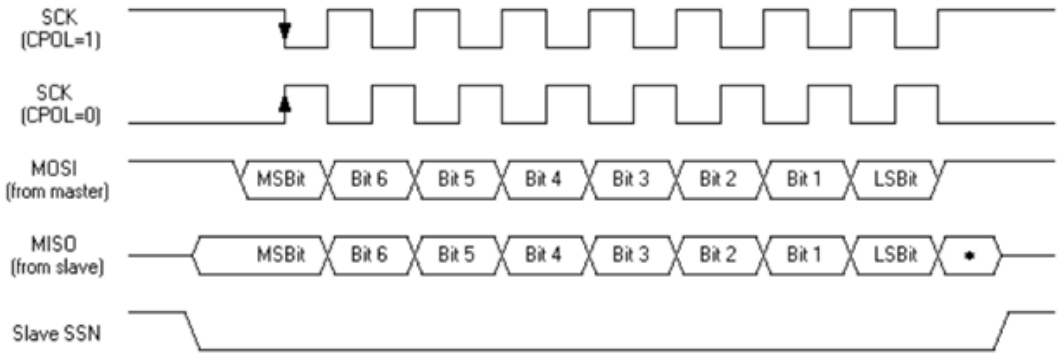


图 20-2 SPI 数据/时钟时序图（CPHA=0）

20.4.2 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。

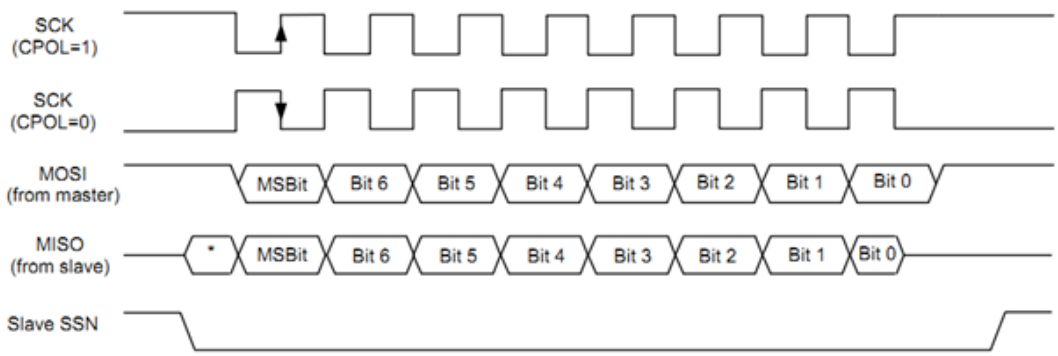


图 20-3 SPI 数据/时钟时序图（CPHA=1）

20.4.1 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。FM33A0xxEV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。

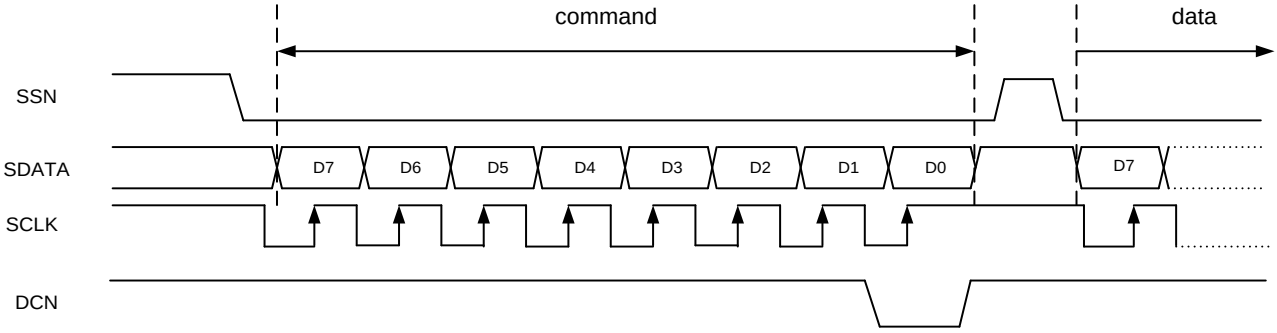


图 20-4 4 线半双工写操作

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。

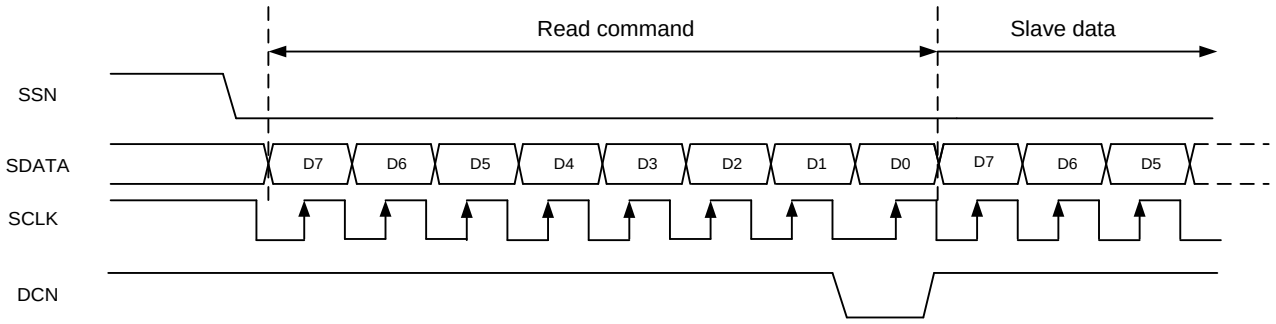


图 20-5 4 线半双工读操作（无 dummy cycle）

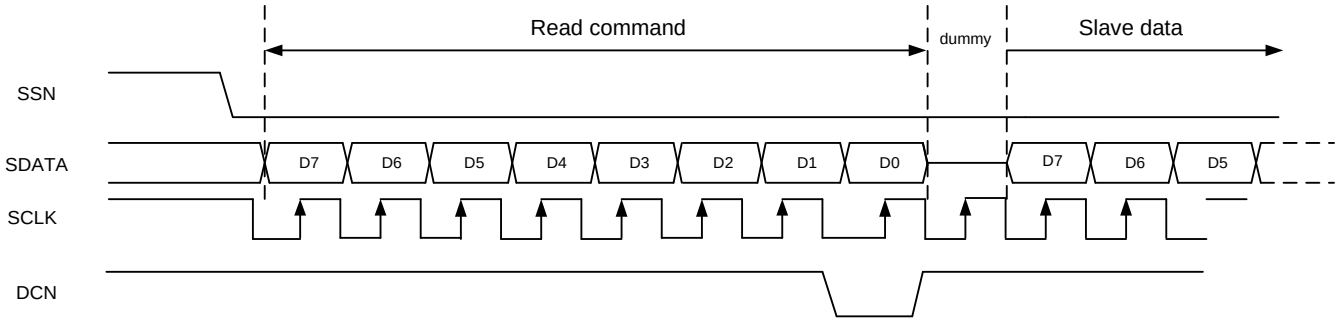


图 20-6 4 线半双工读操作（有 dummy cycle）

20.5 功能描述

20.5.1 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 20-2 所示，当 SPI 配置为主器件时，SSN 引脚必须接高电平，当 SPI 配置为从器件时，SSN 引脚必须接低电平。

SPI 主从器件的连接如下图所示。

主从器件的 MOSI、MISO 和 SCK 分别连在一起，主器件的 SSN 必须接高电平，从器件的 SSN 必须接低电平。主从器件通过 MOSI、MISO 连成一个环路，主器件输出时钟，数据传输时，主器件通过 MOSI 输出数据，从器件通过 MISO 输出数据。一字节数据传输完毕，主从器件将交换 8 位移位寄存器数值。

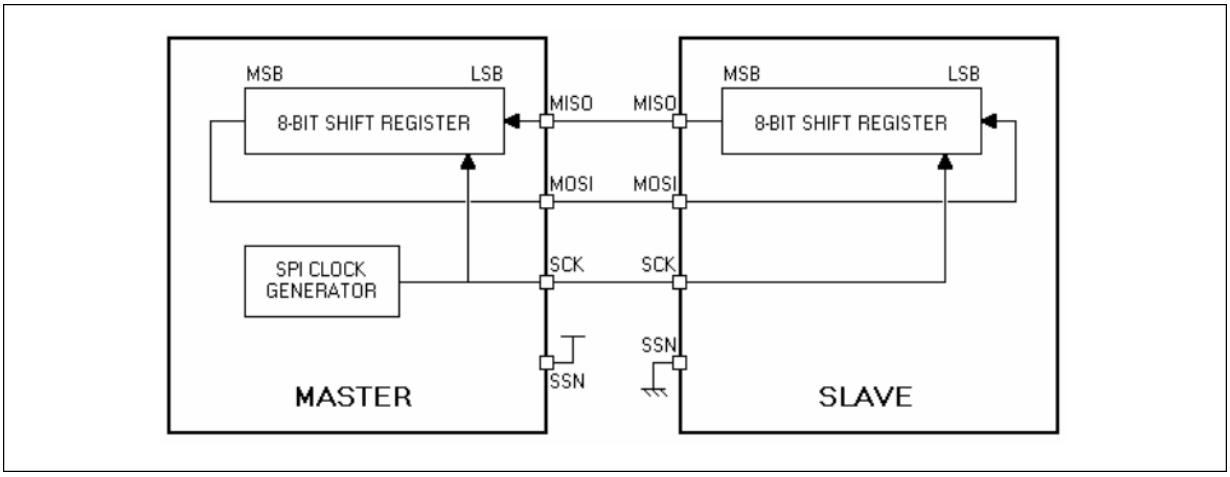


图 20-7 SPI Master/SPI Slave 互连

20.5.2 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。

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。

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。

20.5.3 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。

20.5.4 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。

20.5.5 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）；

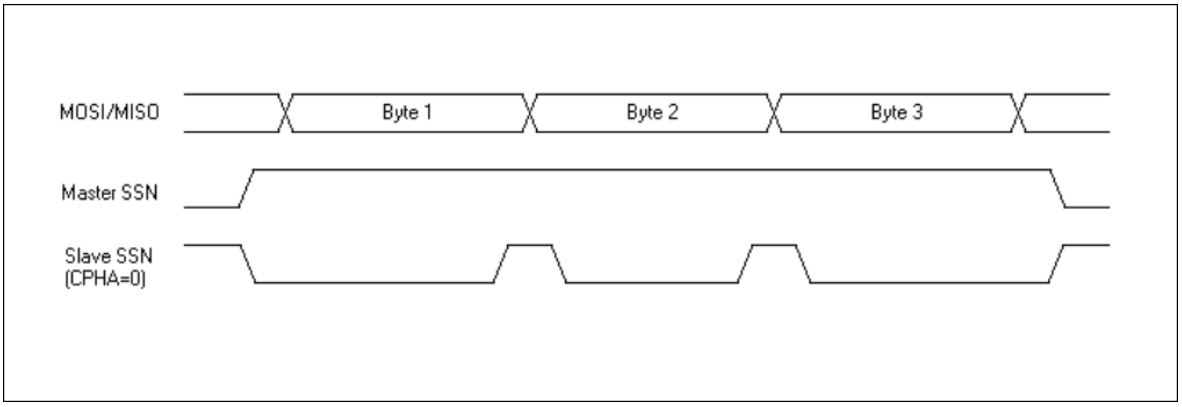


图 20-8 SPI SSN 时序图（SSNM=1，CPHA=0）

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。

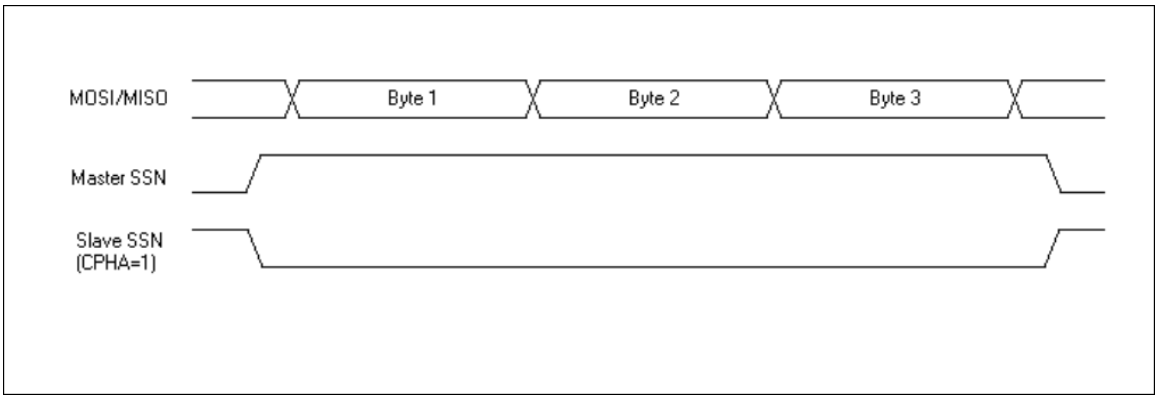


图 20-9 SPI SSN 时序图（SSNM=0）

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。

20.5.6 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。

20.5.7 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 }

20.6 寄存器

SPI0模块起始地址：0x4001_0400

SPI1模块起始地址：0x4001_0800

SPI2模块起始地址：0x4001_4800

SPI3模块起始地址：0x4001_4C00

SPI4模块起始地址：0x4001_6400

offset 地址	名称	符号
SPI0 寄存器(模块起始地址：0x40010400)		
0x00	SPI0 控制寄存器 1 （SPI0 Control Register1）	SPI0_CR1
0x04	SPI0 控制寄存器 2 （SPI0 Control Register2）	SPI0_CR2
0x08	SPI0 控制寄存器 3 （SPI0 Control Register3）	SPI0_CR3
0x0C	SPI0 中断使能寄存器 （SPI0 Interrupt Enable Register）	SPI0_IER
0x10	SPI0 中断状态寄存器 （SPI0 Status Register）	SPI0_ISR
0x14	SPI0 发送数据缓冲寄存器 （SPI0 Transmit Buffer）	SPI0_TXBUF
0x18	SPI0 接收数据缓冲寄存器 （SPI0 Receive Buffer）	SPI0_RXBUF
SPI1 寄存器(模块起始地址：0x40010800)		
0x00	SPI1 控制寄存器 1 （SPI1 Control Register1）	SPI1_CR1
0x04	SPI1 控制寄存器 2 （SPI1 Control Register2）	SPI1_CR2
0x08	SPI1 控制寄存器 3 （SPI1 Control Register3）	SPI1_CR3
0x0C	SPI1 中断使能寄存器 （SPI1 Interrupt Enable Register）	SPI1_IER
0x10	SPI1 中断状态寄存器 （SPI1 Status Register）	SPI1_ISR
0x14	SPI1 发送数据缓冲寄存器 （SPI1 Transmit Buffer）	SPI1_TXBUF
0x18	SPI1 接收数据缓冲寄存器 （SPI1 Receive Buffer）	SPI1_RXBUF
SPI2 寄存器(模块起始地址：0x40014800)		
0x00	SPI2 控制寄存器 1 （SPI2 Control Register1）	SPI2_CR1
0x04	SPI2 控制寄存器 2 （SPI2 Control Register2）	SPI2_CR2
0x08	SPI2 控制寄存器 3 （SPI2 Control Register3）	SPI2_CR3
0x0C	SPI2 中断使能寄存器 （SPI2 Interrupt Enable Register）	SPI2_IER
0x10	SPI2 中断状态寄存器	SPI2_ISR

offset 地址	名称	符号
	（SPI2 Status Register）	
0x14	SPI2 发送数据缓冲寄存器 （SPI2 Transmit Buffer）	SPI2_TXBUF
0x18	SPI2 接收数据缓冲寄存器 （SPI2 Receive Buffer）	SPI2_RXBUF
SPI3 寄存器(模块起始地址：0x40014C00)		
0x00	SPI3 控制寄存器 1 （SPI3 Control Register1）	SPI3_CR1
0x04	SPI3 控制寄存器 2 （SPI3 Control Register2）	SPI3_CR2
0x08	SPI3 控制寄存器 3 （SPI3 Control Register3）	SPI3_CR3
0x0C	SPI3 中断使能寄存器 （SPI3 Interrupt Enable Register）	SPI3_IER
0x10	SPI3 中断状态寄存器 （SPI3 Status Register）	SPI3_ISR
0x14	SPI3 发送数据缓冲寄存器 （SPI3 Transmit Buffer）	SPI3_TXBUF
0x18	SPI3 接收数据缓冲寄存器 （SPI3 Receive Buffer）	SPI3_RXBUF
SPI4 寄存器(模块起始地址：0x40016400)		
0x00	SPI4 控制寄存器 1 （SPI4 Control Register1）	SPI4_CR1
0x04	SPI4 控制寄存器 2 （SPI4 Control Register2）	SPI4_CR2
0x08	SPI4 控制寄存器 3 （SPI4 Control Register3）	SPI4_CR3
0x0C	SPI4 中断使能寄存器 （SPI4 Interrupt Enable Register）	SPI4_IER
0x10	SPI4 中断状态寄存器 （SPI4 Status Register）	SPI4_ISR
0x14	SPI4 发送数据缓冲寄存器 （SPI4 Transmit Buffer）	SPI4_TXBUF
0x18	SPI4 接收数据缓冲寄存器 （SPI4 Receive Buffer）	SPI4_RXBUF

20.6.1 SPIx 控制寄存器（SPIx_CR1）

名称	SPIx_CR1(x=0,1,2,3,4)							
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	CPHOL	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
31:12	-	RFU：未实现，读为 0
11	**IOSWAP**	MOSI 和 MISO 引脚交换 (IO swapping) 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。(Master Mode) 1：Master 模式 0：Slave 模式
7:6	**WAIT**	Master 模式下，每发送完一帧后加入至少(1+WAIT)个 SCK cycle 等待时间，再传输下一帧的数据。如果 SSN 由硬件控制，并且 SSNM=1，则硬件会自动拉高 SSN。
5:3	**BAUD**	Master 模式波特率配置位: (Baud rate) 000： $f_{APBCLK}/2$ 001： $f_{APBCLK}/4$ 010： $f_{APBCLK}/8$ 011： $f_{APBCLK}/16$ 100： $f_{APBCLK}/32$ 101： $f_{APBCLK}/64$ 110： $f_{APBCLK}/128$ 111： $f_{APBCLK}/256$ 当通信正在进行的时候，不能修改这些位。
2	**LSBF**	帧格式 (LSB First) 0：先发送 MSB 1：先发送 LSB 注：当通信在进行时不能改变该位的值。
1	**CPHOL**	时钟极性选择 (Clock Polarity) 1：串行时钟停止在高电平 0：串行时钟停止在低电平 注：当通信在进行时不能改变该位的值 注：当 SSN 为低时不能改变该位的值
0	**CPHA**	时钟相位选择 (Clock Phase) 1：第二个时钟边沿是第一个捕捉边沿 0：第一个时钟边沿是第一个捕捉边沿 注：当通信在进行时不能改变该位的值。

20.6.2 SPIx 控制寄存器 2（SPIx_CR2）

名称	SPIx_CR2(x=0,1,2,3,4)
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 (Dummy cycle Enable) 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 (Receive Only mode) 1：启动 Master 的单接收模式 0：关闭单接收模式（收发全双工）
10:9	**DLEN**	通信数据字长配置 (Data Length) 00：8bit 01：16bit 10：24bit 11：32bit
8	**HALFDUPLEX**	通信模式选择 (Half-Duplex mode) 0：标准 SPI 模式，4 线全双工 1：DCN 模式，4 线半双工
7	**HD_RW**	半双工模式下主机读写操作配置 (Read/Write config for Half-Duplex mode) 0：4 线半双工协议下主机写入从机 1：4 线半双工协议下主机读取从机
6	**CMD8b**	半双工模式下定义 command 帧长度 (Command 8 bits) 1：command 帧固定为 8bit 0：command 帧长度由 DLEN 定义
5	**SSNM**	Master 模式下 SSN 控制模式选择 (SSN mode) 1：每发送完一帧后 Master 拉高 SSN，维持高电平时间由 WAIT 寄存器控制 0：每发送完一帧后 Master 保持 SSN 为低
4	**TXO_AC**	TXONLY 硬件自动清空的使能 (TXONLY auto-clear enable) 1：TXONLY 硬件自动清零有效，软件使能 TXO 后，等待发送完毕后，硬件清零 0：关闭 TXONLY 硬件自动清零
3	**TXO**	TXONLY 控制位 (Transmit Only mode enable) 1：启动 Master 的单发送模式

位号	助记符	功能描述
		0：关闭单发送模式（收发全双工）
2	**SSN**	Master 模式下，如果 SSNSEN 为 1，软件可以通过此位控制 SSN 输出电平 1：SSN 输出高电平 0：SSN 输出低电平
1	**SSNSEN**	Master 模式下，软件控制 SSN 使能 (SSN Software Enable) 1：Master 模式下 SSN 输出由软件控制 0：Master 模式下 SSN 输出由硬件自动控制
0	**SPIEN**	SPI 使能 (SPI enable) 1：使能 SPI 0：关闭 SPI，清空发送接收缓存

20.6.3 SPIx 控制寄存器 3（SPIx_CR3）

名称	SPIx_CR3(x=0,1,2,3,4)							
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				W-0	W-0	W-0	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 1 清零
2	**RXBFC**	Receive Buffer Clear，软件写 1 清除发送缓存，写 1 清零
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

20.6.4 SPIx 中断控制寄存器（SPIx_IER）

名称	SPIx_IER(x=0,1,2,3,4)							
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 (Error Interrupt Enable)
1	**TXIE**	发送完成中断使能 (Transmit Interrupt Enable)
0	**RXIE**	接收完成中断使能 (Receive Interrupt Enable)

20.6.5 SPI 中断标志寄存器（SPIx_ISR）

名称	**SPIx_ISR(x=0,1,2,3,4)**							
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-1	U-0	R/W-0	R/W-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 (Data/Command transmit config) 0：DCN=0，表示命令帧 1：DCN=1，表示数据帧 软件应在发送前设置 DCN_TX 寄存器，如果 DCN_TX=0，硬件在完成一帧发送后，自动将 DCN_TX 置 1，即默认只会发送一个命令帧，后续都是数据帧。
11	**-**	RFU：未实现，读为 0
10	**RXCOL**	接收缓存溢出，软件写 1 清零 (Receive Collision flag,write 1 to flag)
9	**TXCOL**	发送缓存溢出，软件写 1 清零 (Transmit Collision flag,write 1 to clear)
8	**BUSY**	SPI 空闲标志，只读 (busy flag) 1：SPI 传输进行中 0：SPI 传输空闲
7	**-**	RFU：未实现，读为 0
6	**MERR**	Master Error 标志(Master Error flag)

位号	助记符	功能描述
		当 Master 下传输未满 8 位 SSN 就被拉高时，MERR 置位
5	**SERR**	Slave Error 标志(Slave Error flag) 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(TX Buffer Empty flag) 1：发送缓存空，软件写 TXBUF 清零 0：发送缓存满
0	**RXBF**	RX Buffer Full 标志位(RX Buffer Full flag) 1：接收缓存满，软件读 RXBUF 清零 0：接收缓存空

20.6.6 SPIx 发送缓存寄存器（SPIx_TXBUF）

名称	SPIx_TXBUF(x=0,1,2,3,4)							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TXBUF[31:24]							
位权限	W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TXBUF[23:16]							
位权限	W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TXBUF[15:8]							
位权限	W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF[7:0]							
位权限	W-00000000							

位号	助记符	功能描述
31:0	**TXBUF**	SPI 发送缓存 (Transmit Buffer)

20.6.7 SPIx 接收缓存寄存器（SPIx_RXBUF）

名称	SPIx_RXBUF(x=0,1,2,3,4)							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RXBUF[31:24]							
位权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RXBUF[23:16]							
位权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RXBUF[15:8]							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF[7:0]							
位权限	R-00000000							

位号	助记符	功能描述
31:0	**RXBUF**	SPI 接收缓存 (Receive Buffer)

21 智能卡接口（ISO7816）

21.1 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 1 个 7816 主机接口模块。

- 1路7816接口（映射到两组GPIO）
- 具备卡时钟输出端口，输出频率在1MHz~5MHz之间可设
- 位传输方向可配置，支持MSB First或LSB First
- 错误信号宽度可配置为1/1.5/2个ETU
- 发送数据支持传输错误重发机制，重发次数可配置为0~3次
- 支持EGT可设0~256，并支持多种超时中断
- 具有数据接收完成/接收错误中断，并提示错误类型
- 发送中断产生条件可配置为缓冲区空或移位寄存器空
- 支持DMA接口

21.2 结构框图

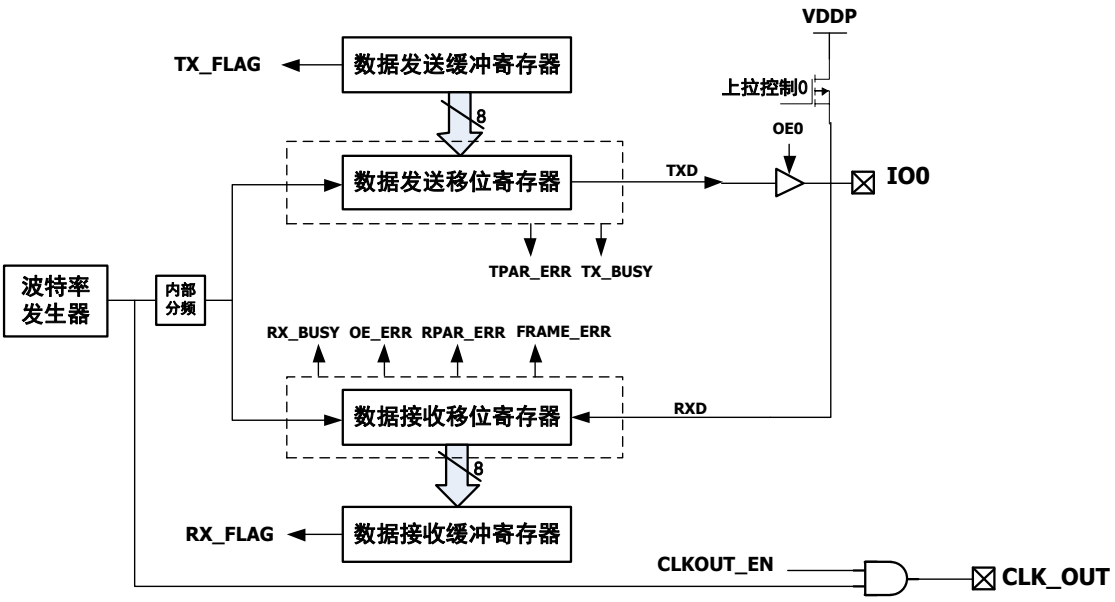


图 21-1 ISO7816 结构框图

21.3 引脚定义

ISO7816 模块使用 2 个引脚与外部器件通信，引脚映射关系如下。

引脚		ISO7816	符号	功能
PC2	PG2	U7816	U7816CLK	通信时钟
PC3	PG3		U7816IO	双向数据收发

表 21-1 ISO7816 引脚定义

21.4 接口时序

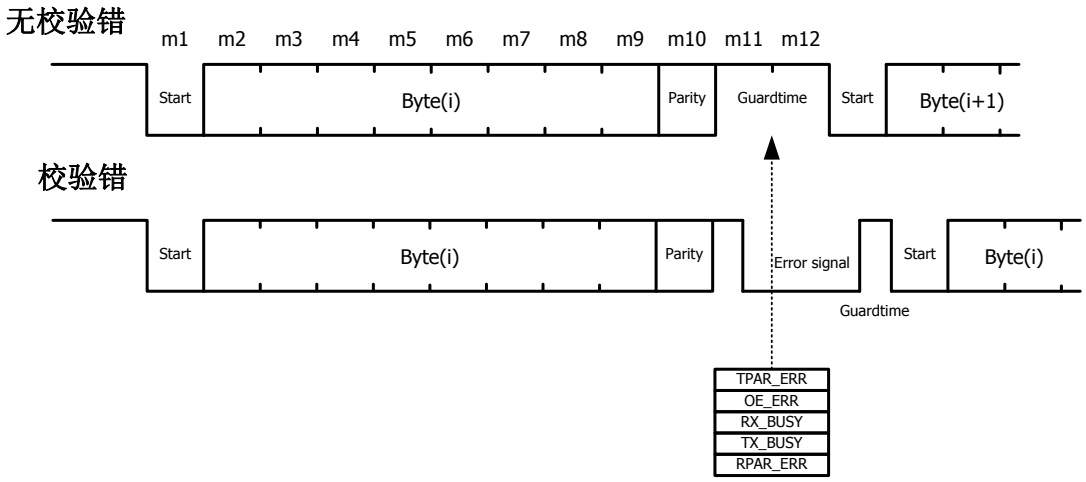


图 21-2 ISO7816 数据帧结构

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。

21.5 功能描述

21.5.1 数据接收

7816 数据接收过程：

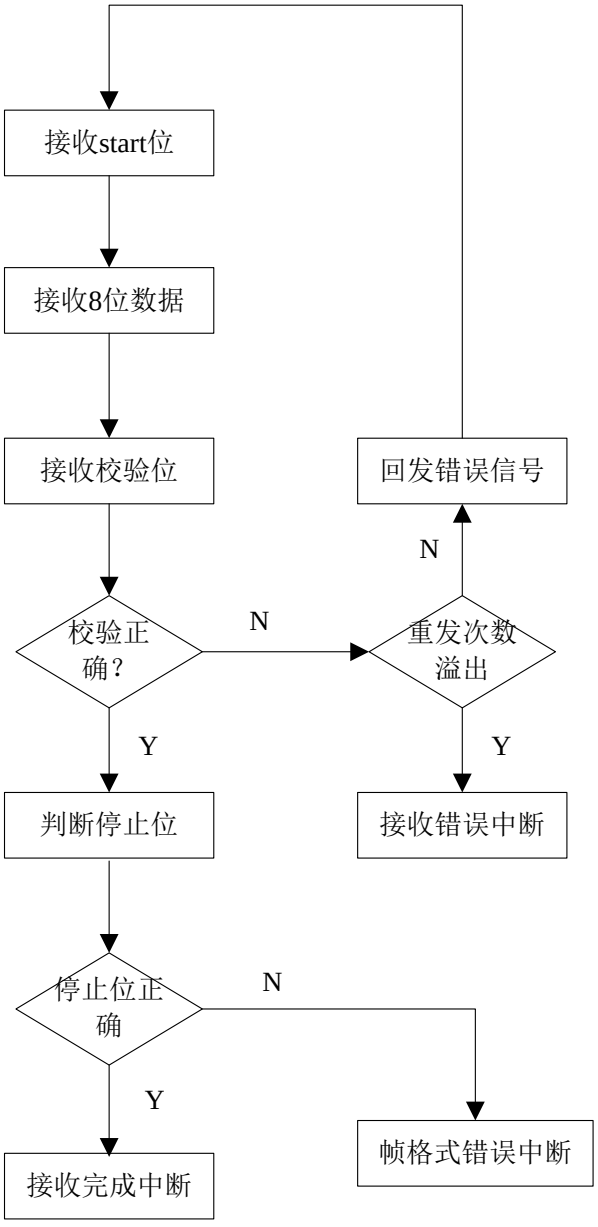


图 21-3 ISO7816 数据接收过程

21.5.2 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：

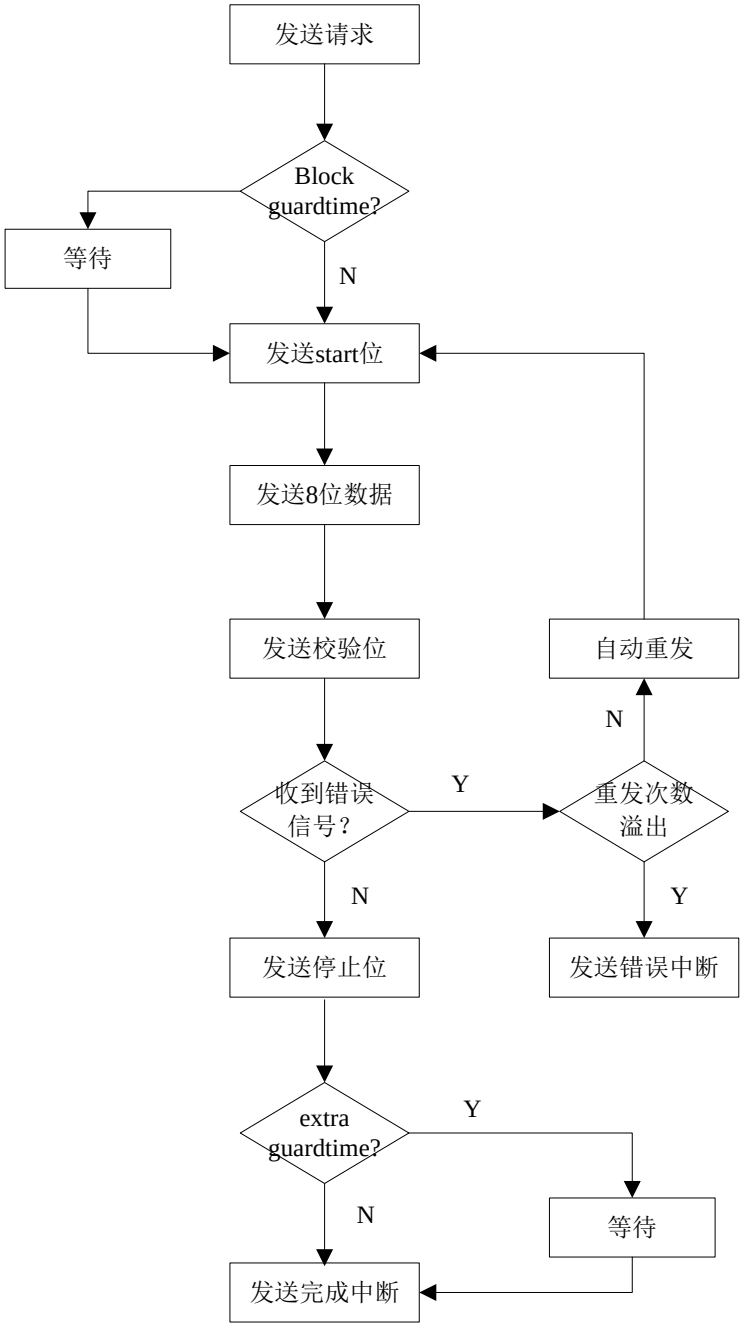


图 21-4 ISO7816 数据发送过程

21.5.3 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

21.6 寄存器

7816模块起始地址：0x4001_1C00

offset 地址	名称	符号
U7816(模块起始地址:0x40011C00)		
0x00	U7816 控制寄存器 （U7816 Control Register）	U7816_CR
0x04	U7816 帧格式寄存器 （U7816 Frame Format Register）	U7816_FFR
0x08	U7816 额外保护时间寄存器 （U7816 Extra Guard Time Register）	U7816_EGTR
0x0C	U7816 工作时钟分频寄存器 （U7816 Prescaler Register）	U7816_PSC
0x10	U7816 波特率寄存器 （U7816 Baud rate Generator Register）	U7816_BGR
0x14	U7816 数据接收缓存寄存器 （U7816 Receive Buffer）	U7816_RXBUF
0x18	U7816 数据发送缓存寄存器 （U7816 Transmit Buffer）	U7816_TXBUF
0x1C	U7816 中断使能寄存器 （U7816 Interrupt Enable Register）	U7816_IER
0x20	U7816 中断状态标志寄存器 （U7816 Interrupt Status Register）	U7816_ISR

21.6.1 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	HPUAT	HPUEN	-
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-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 (Transmit Enable) 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，将 SCL 信号转化为低电平
4	RXEN	U7816 通道接收使能控制位 (Receive Enable) 1：通道接收使能，可接收数据

位号	助记符	功能描述
		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 (Clock output Enable) 1：7816 时钟输出使能 0：7816 时钟输出禁止
2	HPUAT	U7816 通道数据发送强上拉电阻自动有效控制位 (High-Pullup Automatically) 1：数据发送时上拉电阻自动有效，接收态上拉电阻无效 0:数据发送时上拉电阻自动有效功能禁止,上拉电阻由 HPUEN，LPUEN 控制
1	HPUEN	U7816 通道强上拉使能控制位 (High-Pullup Enable) 1：强上拉有效 0：强上拉无效
0	-	保留位

21.6.2 U7816 帧格式寄存器（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	-				SFREN	ERSW		ERSGD
位权限	U-0				R/W-0	R/W-00		R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BGTEN	REP_T	PAR		RFREN	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
31:12	-	RFU：未实现，读为 0
11	SFREN	Guard Time 发送长度控制位 (Send long Frame Enable) 1：Guard time 为 3 etu 0：Guard time 为 2 etu
10:9	ERSW	ERROR SIGNAL 宽度选择 (Error Signal Width) 11：ERROR SIGNAL 宽度为 1ETU; 10：ERROR SIGNAL 宽度为 1.5ETU; 01：ERROR SIGNAL 宽度为 2ETU; 00：ERROR SIGNAL 宽度为 2ETU;
8	ERSGD	ERROR SIGNAL 后 GUARDTIME 宽度选择（仅在发送时有效）(Error Signal Guard Time) 1：ERROR SIGNAL 后 GUARDTIME 为 1~1.5ETU。 0：ERROR SIGNAL 后 GUARDTIME 为 2~2.5ETU。 ERROR SIGNAL 宽度为整数 ETU 时 GUARDTIME 为 1.5 或 2.5ETU；ERROR SIGNAL 宽度为 1.5ETU 时 GUARDTIME 为 1 或 2ETU

位号	助记符	功能描述
7	BGTEN	BGT 控制位。控制接收->发送之间是否插入 BGT。BGT 是接收->发送之间需要的最小时间（block guard time enable） 1：BGT 使能，插入 Block guard time(12 etu); 0：BGT 禁止，不插入 Block guard time(12 etu);
6	REP_T	控制接收数据奇偶校验出错时自动重发次数 (Repeated Times) 1：3 次 0：1 次
5:4	PAR	奇偶校验类型选择 (Parity) 00：Even 01：Odd 10：Always 1 11：不校验，处理
3	RFREN	Guard Time 接收长度控制位 (Receive short Frame) 1：Guard time 为 1 etu 0：Guard time 为 2 etu
2	TREPEN	发送数据奇偶校验错的处理方式选择 (Transmit Repeat Enable) 1：收到奇偶校验出错标志（error signal），根据 T＝0 协议自动进行回发。在单一 byte 重复发送次数超过 REP_T 后，置 tx_parity_err 标志，进行中断 0：收到 Error signal 时不进行自动回发，置 tx_parity_err 标志，直接中断
1	RREPEN	接收数据奇偶校验错的处理方式选择 (Receive Repeat Enable) 1：奇偶校验错，根据 T=0 协议自动回发 ERROR SIGNAL。单一 BYTE 连续接收次数超过 REP_T 后，置 RX_PARITY_ERR 标志，进行中断 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 (bit Direction Conversion) 1：反向编码，先收发 MSB；(收发数据+校验位)反逻辑电平 0：正向编码，先收发 LSB ； (收发数据+校验位)正逻辑电平

21.6.3 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-0000 0000							

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 Extra Guard Time 时间（以 ETU 为单位） (Transmit Extra Guard Time)

21.6.4 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-0 0011				

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(Clock Divider)，控制 7816 工作时钟分频数。 U7816 工作时钟与 APBCLK 的分频关系： $F_{7816}=F_{APBCLK}/(CLKDIV+1)$ 特殊情况：CLK_DIV 设置成 0 或 1 时，$F_{7816}=F_{APBCLK}/2$ 注：7816 协议规定的工作时钟范围是 1~5MHZ。

21.6.5 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[11:8]			
位权限	U-0				R/W-0001			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PDIV[7:0]							
位权限	R/W-0111 0011							

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(Pre-Divider)，控制 7816 通信分频比（波特率） Baud ＝ F_{7816}/(PDIV ＋ 1) *注意：PDIV 最小可用值是 0x1，应用禁止配置 0x0*

21.6.6 U7816 数据接收缓存寄存器（U7816_RXBUF）

名称	U7816_RXBUF							
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	RXBUF							
位权限	R-0000 0000							

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 (Receive Buffer)

21.6.7 U7816 数据发送缓存寄存器（U7816_TXBUF）

名称	U7816_TXBUF							
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	TXBUF							
位权限	W-0000 0000							

位号	助记符	功能描述

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 (Transmit Buffer)

21.6.8 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	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	RXIE	数据接收中断使能位。对应 RXIF 中断标志位 (Receive Interrupt Enable) 1：当 RXIF 寄存器置位时产生接收完成中断 0：禁止接收完成中断
1	TXIE	数据发送中断使能位。对应 TXIF 中断标志位(Transmit Interrupt Enable) 1：当 TXIF 寄存器置位时产生发送完成中断 0：禁止发送完成中断
0	LSIE	线路状态中断使能位。对应 ERRIF 中断标志位(Line Status Interrupt Enable) 1：当 ERRIF 寄存器置位时产生线路错误中断 0：禁止线路错误中断

21.6.9 U7816 中断状态标志寄存器（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-0	R-0	R-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-0	R-1	R-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；(Waiting for Repeat flag) 状态机进入发送错误信号状态时置位，收到数据起始位或者进入发送状态时硬件清零； 软件只读。
17	TXBUSY	发送数据忙标志。（发送完成后自动清零）(Transmission busy flag) 1：处于数据发送状态，发送移位寄存器正在发送数据。（开始发送起始位置 1，停止位中间清零） 0：数据发送空闲
16	RXBUSY	接收数据忙标志。（接收完成后自动清零）(Receiving busy flag) 1：处于数据接收状态，接收移位寄存器正在接收数据。（收到起始位置 1，收到停止位清零，若接收数据出错需重发，则回发 error signal 时清零。即数据及校验位接收之后，无论是否需要重发，都需要及时清除该标志） 0：数据接收空闲
15:12	-	RFU：未实现，读为 0
11	TPARERR	发送数据奇偶校验错误标志位。硬件置位，写 1 清零 (Transmit Parity Error,write 1 to clear)
10	RPARERR	接收数据奇偶校验错误标志位。硬件置位，写 1 清零 (Receive Parity Error flag,write 1 to clear)
9	FRERR	接收帧格式错误标志位。硬件置位，写 1 清零 (Frame Error flag,write 1 to clear) 1：帧格式有错误，接收到的 frame 字节长度有误或接收到的 frame 或者 stop 位有误 0：接收数据时无奇偶校验错误
8	OVERR	接收溢出错误标志位。硬件置位，写 1 清零 (Receive Overflow Error,write 1 to clear) 1：接收缓冲寄存器未被读出，又接收到新的数据，溢出错误标志有效。原接收缓冲寄存器内数据被新覆盖 0：无溢出错误
7:3	-	RFU：未实现，读为 0
2	RXIF	接收完成标志(Receive interrupt flag)，U7816 接口控制器每收到 1byte 数据，根据接收的通道相应发出一次中断。硬件置位，读数据接收缓冲寄存器清零 1：接收到 1byte 数据，数据接收缓冲器满 0：未接收到数据，数据接收缓冲器空
1	TXIF	发送缓冲区空标志(Transmit interrupt flag)，上电复位后此标志就自动置位，表示缓冲区空，可以写入数据。软件写入数据后标志自动清除，数据从发送缓存移入移位寄存器后置 1 1：数据发送缓冲器空 0：数据发送缓冲器内有数据待发送

位号	助记符	功能描述
0	ERRIF	错误标志(Error interrupt flag)，寄存器配置出错或传输过程中出错。此 bit 是 TPARERR、RPARERR、FRERR、OVERR 的或。软件通过清除以上错误标志寄存器来清除此 bit。

22 QuadSPI（QSPI）

22.1 概述

QuadSPI是一个高速串行接口，可以使用最多6根信号线与外部QSPI存储器通信。QuadSPI接口支持传统SPI模式（4线半双工，不支持全双工）、Dual-SPI模式和Quad-SPI模式。通过QuadSPI接口，MCU可以有效扩展片内存储器，并且能够通过eXecution-In-Place（XIP）方式，直接从片外存储器取指令运行。

QuadSPI模块主要特点如下：

- QSPI主机
- 支持三种工作模式：外设模式、自动查询模式、存储器映射模式
- 灵活可编程的命令字节（instruction）
- 灵活可编程的帧格式
- 集成数据16字节FIFO
- 外设模式支持DMA访问

22.2 结构框图

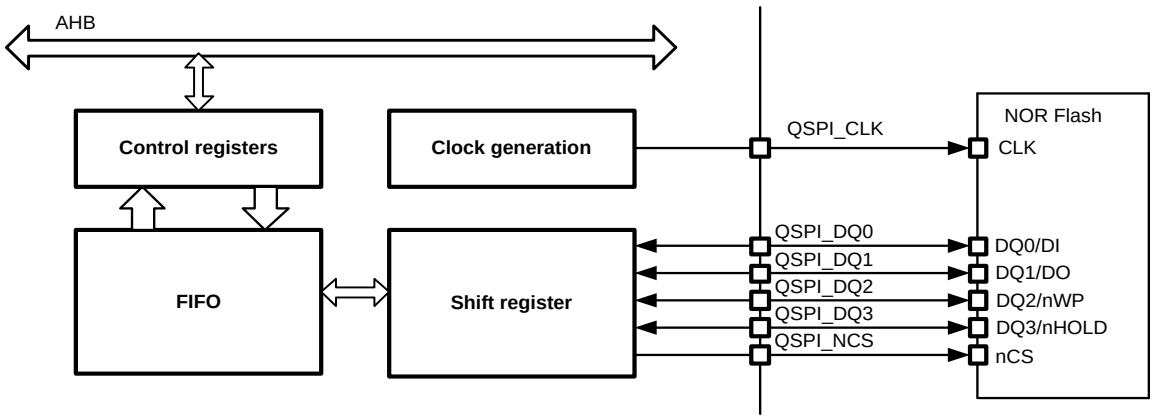


图 22-1 QSPI 结构框图

22.3 引脚定义

QuadSPI 模块最多使用 6 个引脚与外部器件通信，实现片外 NOR flash 连接。

引脚		QSPI	符号	功能
PB6		QSPI	QSPI_DQ2	双向数据
PB7			QSPI_DQ1	双向数据
PB8			QSPI_nCS	片选输出
PB9			QSPI_DQ0	双向数据
PB10			QSPI_CLK	时钟输出
PB11			QSPI_DQ3	双向数据

表 22-1 QuadSPI 引脚定义

22.4 功能描述

22.4.1 工作时钟

QuadSPI模块工作时钟来自于AHBCLK，其对外输出的总线时钟可以是AHBCLK及其分频。用8bit的PRESCALER寄存器可以定义QSPI总线时钟相对于AHBCLK的分频比，支持2~256之间的任意分频，奇数分频时，占空比不是50%。

QSPI输出总线时钟频率由下式定义（最小分频比2）：

$F_{QSPI_CLK} = F_{AHB} / (PRESCALER + 1)$

22.4.2 工作模式

QuadSPI模块支持以下三种工作模式：

- 外设模式
- 自动查询模式
- 存储器映射模式

22.4.2.1 外设模式

外设模式是QuadSPI最基本的工作模式。这个模式下，QuadSPI模块与其他通信外设类似，CPU和DMA都可以通过访问模块寄存器的方式，来实现与外部QSPI从机的通信。通过OPMODE寄存器，可以配置外设读模式和外设写模式，分别用于对外部QSPI存储器读取数据和写入数据。

通过设置DATALEN寄存器，可以配置QSPI的通信字节长度，当通信开始后，QSPI_nCS引脚被拉低，只有DATALEN指定的数据长度传输完成后，nCS才会回到高电平，结束本次通信过程。

当OPMODE=00，QuadSPI处于外设写模式，即应用可以对外部QSPI存储器写入数据，此时FIFO用作写缓存，AHB总线master对QSPI_DATA寄存器的写操作都会触发对FIFO的PUSH，与此同时，FIFO非空会触发QuadSPI接口的指令和数据发送，从而完成对外部QSPI存储器的写操作。写操作过程中，如果FIFO空，而DATALEN定义的字节数还未发送完成，则QSPI总线时钟被暂停，直到软件或DMA向FIFO写入新数据；如果FIFO满，或者AHB准备写入的数据字节数大于FIFO空余字节数（比如FIFO剩余3字节空间时，AHB发起一个32bit写），则写操作被stall，直到FIFO有足够空间容纳本次需要写入的字节。

当OPMODE=01，QuadSPI处于外设读模式，即应用可以从外部QSPI存储器读取数据，此时FIFO作为读缓存，AHB总线master对QSPI_DATA寄存器的读操作都会触发对FIFO的POP，与此同时，FIFO非满的情况下QuadSPI接口会发送指令和地址并不断接收QSPI存储器返回的数据。读操作过程中，如果FIFO空，或者FIFO中剩余字节数小于当前AHB读所请求的字节数，则AHB总线读出数据将被填充0；如果FIFO满，则QSPI总线读取外部存储器暂停，直到AHB完成一次FIFO读。

使用外设模式可以实现对外部Flash存储器的如下访问：

- 配置控制寄存器和查询状态寄存器
- 读取外部Flash存储器
- 编程或擦除外部Flash存储器

注意，只有外设模式下可以对片外Flash进行编程和擦除。

命令发送的触发条件

QuadSPI接口在合适的条件下自动开始发送指令，这里包含以下三种情况：

- 应用不需要提供地址（AMODE=00）和数据（DMODE=00或OPMODE=01），即只发送指令，应用写入INSTRUCTION寄存器后，QuadSPI自动开始发送指令
- 应用需要提供地址（AMODE != 00），但是不需要提供数据（DMODE=00或OPMODE=01），即只发送指令和地址，应用在写入QSPI_ADDR寄存器后，QuadSPI自动开始发送指令
- 应用在写入QSPI_DATA寄存器后，QuadSPI开始自动发送指令

当指令开始发送后，BUSY标志寄存器自动置位。

FIFO控制

外设模式下QuadSPI数据收发都是通过FIFO进行，FIFOLVL寄存器用于指示当前FIFO中保存的数据字节数。

在外设写模式下，AHB向FIFO PUSH数据，QuadSPI接口POP数据；在外设读模式下，QuadSPI

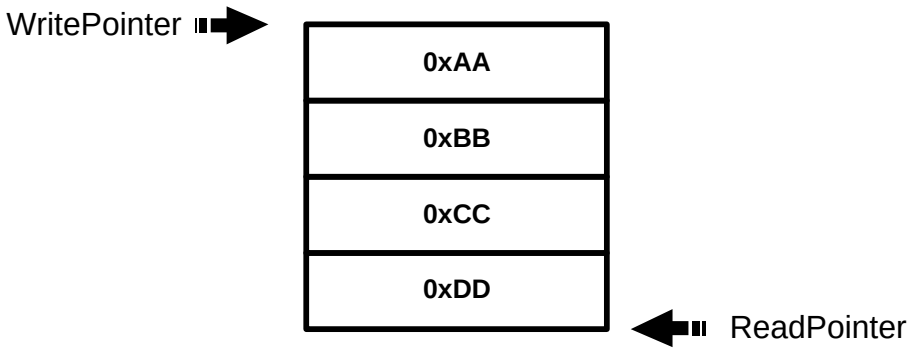
向FIFO PUSH数据，AHB POP数据。AHB的字节、半字访问必须对齐QSPI_DATA寄存器的低地址。FIFOTHR寄存器用于定义FIFO threshold。在外设读模式下，当FIFO中有效数据字节数大于FIFOTHR，FTF（FIFO threshold flag）置位，当DATALEN指定的数据长度读取完成后，TCF（Transfer Complete Flag）标志置位，软件应查询FIFO中剩余字节数；在外设写模式下，当FIFO中空字节数大于FIFOTHR时，FTF标志置位。当FIFO数据字节数不满足FTF置位标志时，FTF将自动清零。当FTIE=1时，FTF标志置位将触发中断。

在外设模式下，当FIFO满时，QuadSPI接口暂停读取QSPI存储器，并且直到FIFO空字节数大于等于4（FIFOLVL<=12）时才会重新开始读取QSPI存储器。

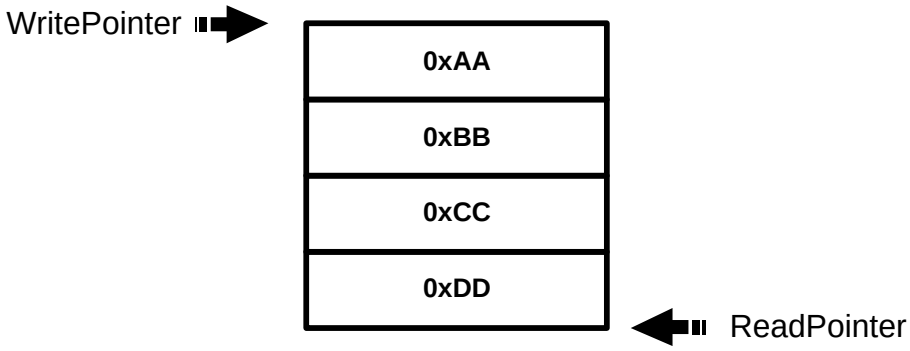
FIFO数据的字节顺序遵循little endian规则，即bit[7:0]是首先被PUSH或POP的。

总线读FIFO时，首先POP出来的字节被对齐到总线32bit数据的低8位[7:0]，第二次POP的字节对齐总线32位数据的[15:8]，以此类推。

假设FIFO中有下图所示4字节数据，则软件或DMA发起word读时，总线上返回的数据是{0xAA,0xBB,0xCC,0xDD}；发起byte读时，总线返回{0x00,0x00,0x00,0xDD}；发起half-word读时，总线返回{0x00,0x00, 0xCC,0xDD}

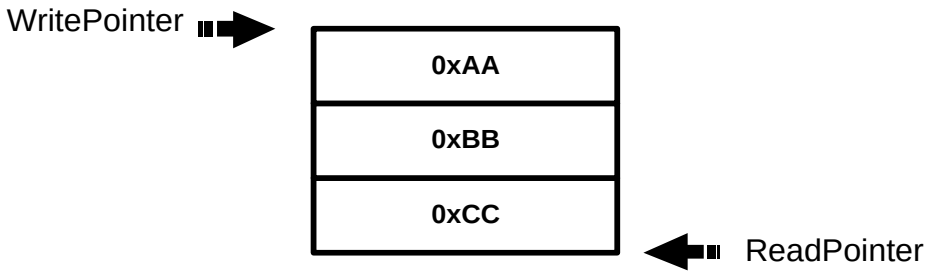


同理，当总线对FIFO写入数据时，bit[7:0]被首先PUSH，接下来是bit[15:8]，以此类推；假设总线发起32bit写操作，写入数据0xAABBCCDD，则经过4次PUSH后FIFO内数据为：



注意：当FIFO内字节数小于AHB总线读取位宽时，AHB返回数据高位将被补0。例如，FIFO中剩余

3字节（参见下图），AHB总线发起一次word读，则QSPI向总线返回{0x00,0xAA,0xBB,0xCC}



DMA支持

使能DMA功能后，AHB总线读写由DMA发起，外设读写模式都可以通过DMA进行。

外设读模式下，软件首先配置指令、地址寄存器，将DMA传输长度和QuadSPI的DATALEN配置为相同的值。QuadSPI开始读取QSPI存储器后，只要FIFO的FTF标志置位，QuadSPI就发送DMA请求，DMA从FIFO中取走当前字节。

外设写模式下，软件首先配置指令、地址寄存器，将DMA传输长度和QuadSPI的DATALEN配置为相同的值。使能DMA后DMA开始向FIFO写入数据，同时QuadSPI开始发送，只要FIFO的FTF标志置位，QuadSPI就发送DMA请求，DMA继续向FIFO写入数据。

应用举例：外设写模式

下面以复旦微电子公司的8M bits SPI串行NOR Flash芯片FM25Q08A为例，说明如何通过QuadSPI接口实现256字节的page program。

- 为了提升编程效率，首先使能QPI模式（详细方法参见FM25Q08A产品手册）
- 配置AMODE、DMODE、OPMODE
- 向INSTRUCTION寄存器写入”Page Program”指令代码02h
- 向地址寄存器写入编程目标地址
- 将DATALEN寄存器设置为256
- 根据QSPI总线时钟频率设置合适的dummy cycles
- Flush FIFO
- 向QSPI_DATA寄存器写入1st word，4字节数据被PUSH进入FIFO，QuadSPI接口自动拉低nCS发送指令、地址，以及后续4字节数据
- 软件继续向FIFO写入数据，同时注意FIFO水位和中断标志
- 全部256字节发送结束后，QuadSPI自动拉高nCS信号，完成本次传输，并置位传输完成标志

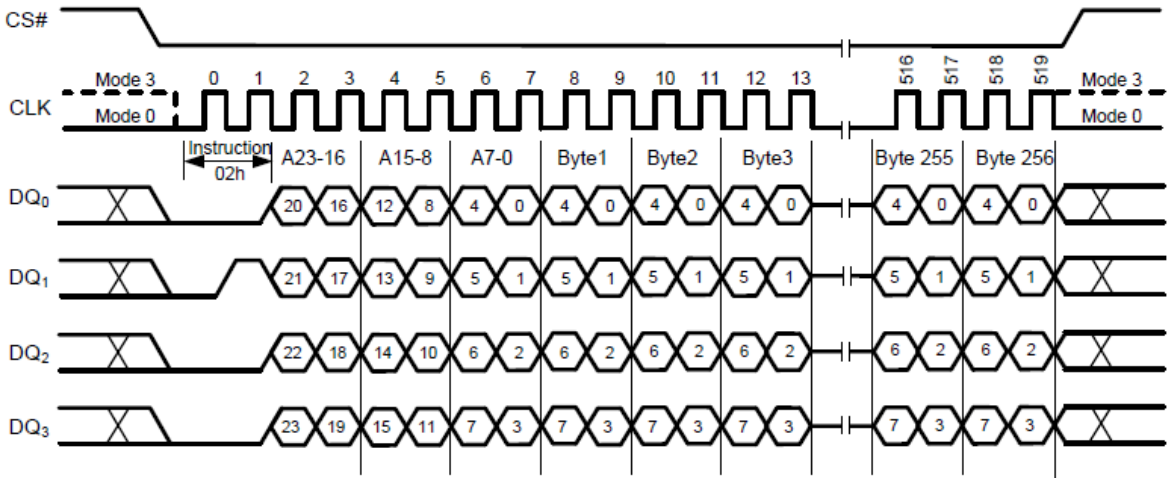


图 22-2 QPI 模式下的 page program

22.4.2.2 自动查询模式

在自动查询模式下（OPMODE=10），QuadSPI模块能够自动轮询外部Flash存储器的状态标志寄存器，而无需CPU参与；QuadSPI以配置的频率周期性读取Flash状态标志，并将返回的数据中的特定bit与期望值自动比较，当符合条件的状态匹配出现时，QuadSPI模块自动停止轮询，并产生中断，通知CPU状态匹配完成。

此模式在等待Flash完成某项操作时非常有效，比如在QuadSPI发送擦除命令后，以自动轮询方式等待Flash擦除完成，在这个过程中CPU可以继续执行其他任务，直到QuadSPI中断通知CPU轮询结束。

自动查询模式一次可以从QSPI存储器读取1~4字节数据，数据被写入QSPI_DATA寄存器，FIFO在这个模式下不起作用。读取的数据经过MASK寄存器屏蔽无关位，然后与MATCH寄存器比较，如果匹配成功，则产生中断标志，BUSY标志清零，QuadSPI停止工作。数据比较支持AND或OR模式，AND模式表示所有未mask位都要匹配，OR模式表示未mask位只要有1bit匹配即可。

22.4.2.3 存储器映射模式

存储器映射模式下，片外Flash被映射到AHB总线上，CPU可以将其视为片内存储器进行直接访问。通过存储映射，可以实现CPU在片外Flash上直接运行代码，即XIP功能。借助XIP，MCU能够有效扩展代码存储空间，而不完全受限于片内Flash容量大小，为应用提供更大的灵活性和扩展性。

在系统总线上，片外Flash将被映射到以下逻辑地址：0x9000_0000 ~ 0x9FFF_FFFF，总共256MB。因此存储器映射模式只支持最大256MB的片外Flash空间访问。

注意：存储器映射模式下只支持对片外Flash的读操作，不支持擦除和编程。

为了提高代码执行效率，QuadSPI采用了Prefetch策略，一旦CPU对片外Flash进行映射读取，QuadSPI会在总线空闲时连续读取后续地址数据，并将数据保存在QuadSPI内部的16字节FIFO中，这在逻辑上相当于实现了4个words的预取指buffer。指令预取会在以下两种情况下停止：

- FIFO满
- CPU取指发生了跳转

当CPU连续地址取指时，AHB总线接口会不断的从FIFO POP数据，每次总线32bit读操作会连续POP 4次，即返回4个字节。当FIFO中有效数据少于4字节时，AHB读访问将被stall，总线被暂停，直到FIFO中有了足够数据。

eXecution In Place (XIP)

XIP模式下CPU直接从外部Flash取指令运行，AHB总线直接访问外部Flash被映射的逻辑地址空间，QuadSPI自动将AHB总线访问转换成QSPI总线访问。指令预取通过FIFO实现，CPU在连续地址取指时，AHB不断从FIFO POP数据直到FIFO空，与此同时QuadSPI不断向FIFO PUSH数据直到FIFO满。一旦取指出现跳转，则需要flush掉FIFO，重新开始新一轮读操作。

在连续取指过程中，如果CPU取指较慢，FIFO被QuadSPI填满的话，QSPI总线上将暂停时钟，但是nCS保持拉低，片外Flash处于等待访问的状态。

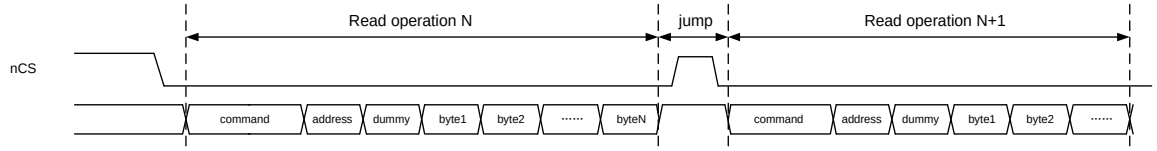


图 22-3 QSPI XIP 示意图

22.4.3 总线配置

QuadSPI支持如下总线配置形式，仅支持SDR格式，不支持DDR：

- Single-SPI
- Dual-SPI
- Quad-SPI

22.4.3.1 Single-SPI (classic SPI)

Single-SPI类似经典SPI总线，仅用到nCS、CLK、DQ0、DQ1四根信号线，其中DQ0作为单向数据输出（SO），DQ1作为单向数据输入（SI），不支持全双工通信。

在Single-SPI配置下，DQ2、DQ3信号是不需要的，可以用做其他功能。如果被连接到片外Flash的nWP和nHOLD接口上，则软件应将对应的GPIO输出固定为1。

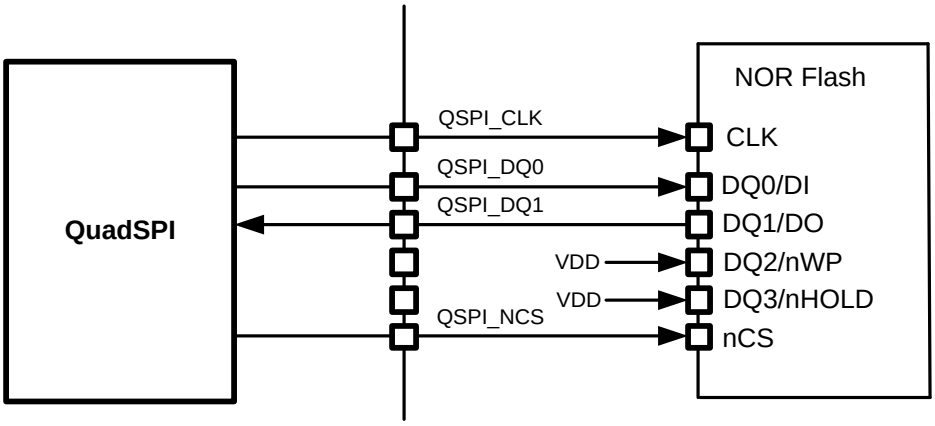


图 22-4 Single-SPI configuration

总线上的数据在CLK下降沿发送，在CLK上升沿被采样，支持SPI Mode0和Mode3，两种模式的差别仅在于总线IDLE时CLK信号的默认电平，参见下图：

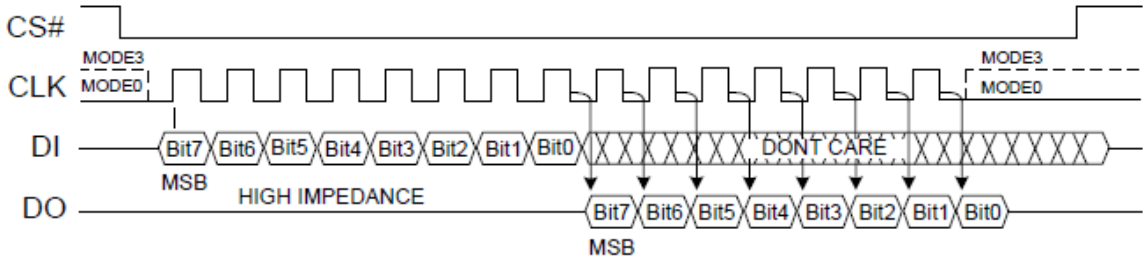


图 22-5 Single-SPI Mode 0 and Mode 3

QuadSPI模块上电后默认为single-SPI模式。

22.4.3.2 Dual-SPI

Dual-SPI配置下同样只用到4根信号线进行通信，与Single-SPI的主要区别是DQ0和DQ1同时参与数据的发送和接收，即DQ0和DQ1变成双向传输。Dual-SPI配置下DQ2和DQ3仍然是不需要的。

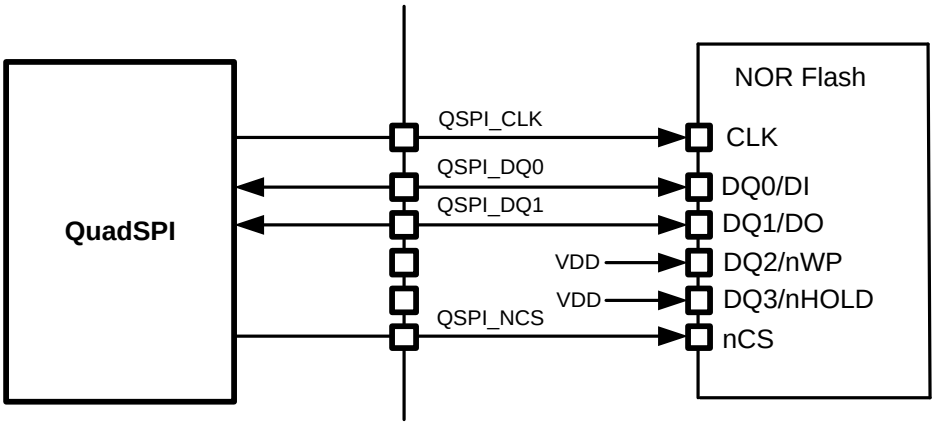


图 22-6 Dual-SPI configuration

22.4.3.3 Quad-SPI

Quad-SPI配置下需要用到6根信号线进行数据收发，DQ0/DQ1/DQ2/DQ3全部用于数据通信，用户可以灵活选择通信数据帧中的命令、地址、数据以单线、双线或四线方式收发。

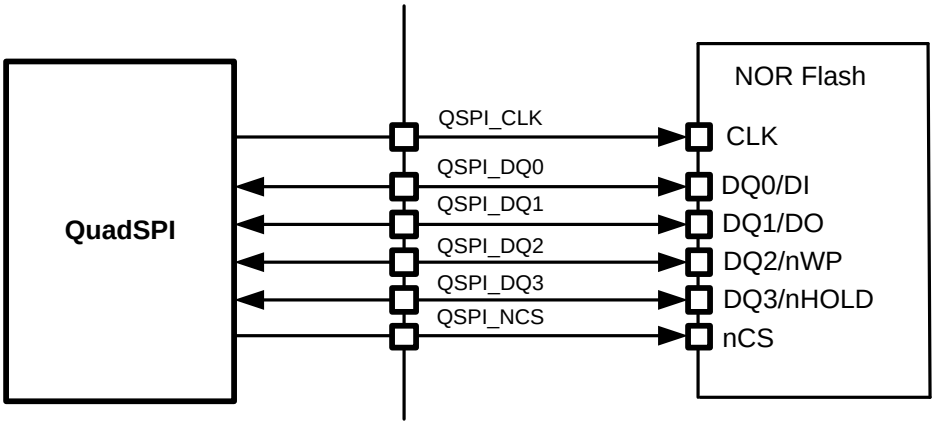


图 22-7 Quad-SPI configuration

22.4.4 帧格式

QSPI通信中，数据帧可以划分为5个部分：instruction, address, alternate, dummy, data
这5个部分根据配置可以存在或跳过，也可以选择以单线、双线或四线方式收发，QuadSPI模块在帧格式上提供了高度的灵活性。

22.4.4.1 Instruction phase

QSPI首次通信都以主机发送特定的指令开始，8bit指令由指令寄存器定义，可以发送任意内容。根据寄存器配置，指令可以以单线、双线或四线发送，某些场景下也可以跳过指令发送阶段。
通过IMODE寄存器，可以配置指令发送方式。

IMODE=00，跳过指令发送

IMODE=01，指令以单线方式发送

以下图为例，QuadSPI对片外Flash发送指令，仅使用DQ0：

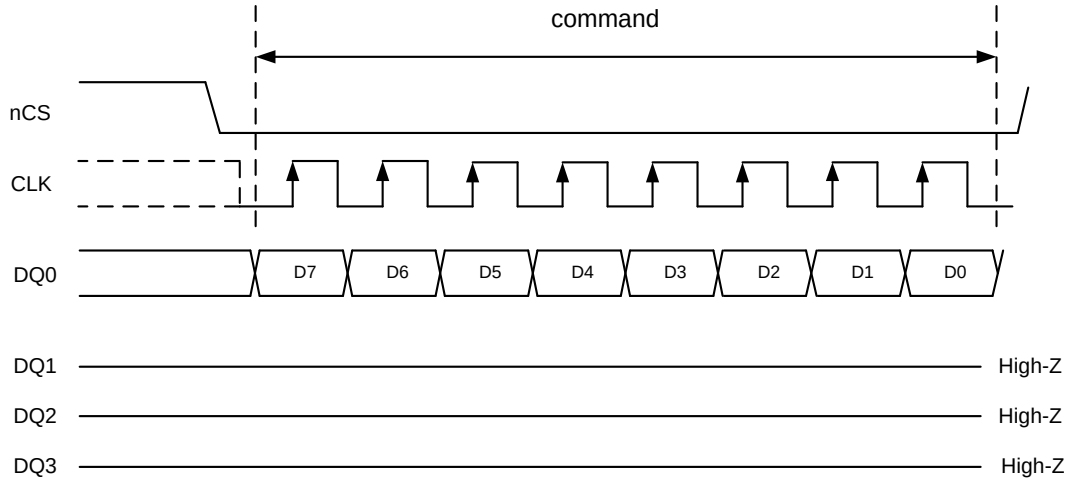


图 22-8 单线发送指令

IMODE=10，指令以双线方式发送

以下图为例，QuadSPI对片外Flash发送指令，使用DQ0和DQ1：

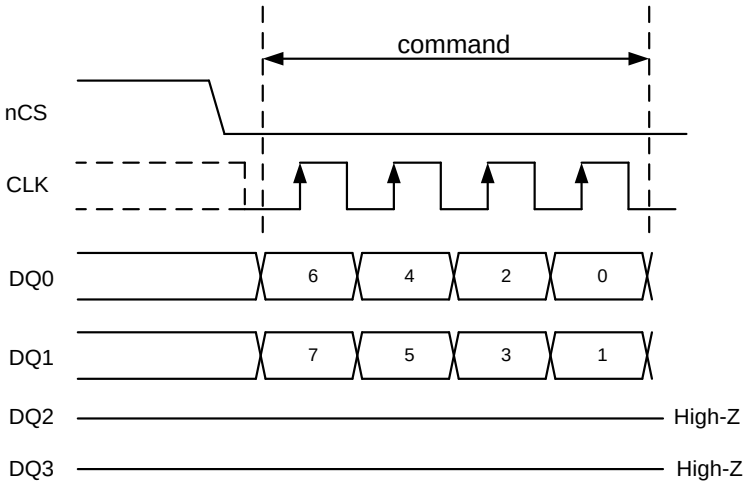


图 22-9 双线发送指令

IMODE=11，指令以四线方式发送

以下图为例，QuadSPI对片外Flash发送指令，使用DQ0、DQ1、DQ2、DQ3：

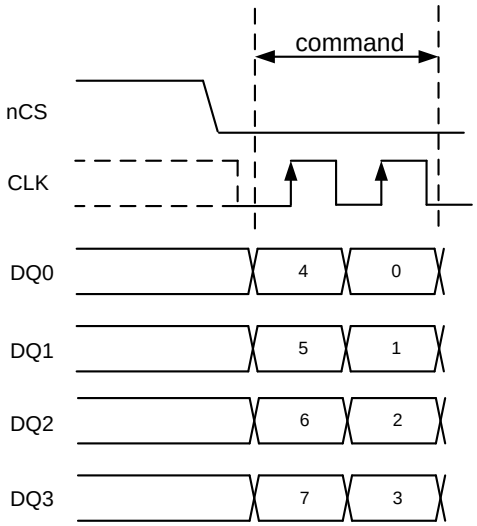


图 22-10 四线发送指令

22.4.4.2 address phase

QuadSPI在address phase向片外存储器发送访问地址，软件可以通过ADSIZE寄存器配置QuadSPI发送1、2、3、4字节地址，通过ADMODE寄存器配置地址以单线、双线、四线方式发送或跳过地址阶段。

在外设模式和自动查询模式下，QuadSPI发送地址由地址寄存器提供；在存储器映射模式下，发送地址直接由AHB总线提供。

22.4.4.3 alternate phase

alternate phase用于发送alternate bytes，某些NOR flash芯片可以使用alternate bytes来启动连续传输模式（比如FM25Q08A的”continous read mode”）。通过ABSIZE寄存器，软件可以配置QuadSPI发送1、2、3、4字节的alternate bytes；通过ABMODE则可以配置以单线、双线、四线方式发送alternate bytes或跳过alternate阶段。QuadSPI发送的alternate bytes内容由寄存器提供。

22.4.4.4 dummy phase

某些QSPI存储器需要在数据帧中插入dummy cycle以提供双向数据口线的turn-around时间。通过DUMCYC寄存器可以设置dummy cycle个数，选择范围是0~31个CLK周期。在dummy phase中，DQ0、DQ1、DQ2、DQ3输出高阻态。

22.4.4.5 data phase

data phase中QuadSPI模块与外部QSPI存储器进行数据收发。

在外设模式和自动查询模式下，通过DATALEN寄存器配置传输数据的长度（bytes），当指定长度的

数据被发送和接收完成后，硬件拉高nCS结束本次通信。

在存储器映射模式下，AHB直接通过QuadSPI的FIFO读取数据，DATALEN寄存器不起作用。此时QuadSPI只支持读操作，不支持写操作。AHB可以进行字节、半字和字读取，分别对应FIFO的POP指针+1、+2和+4。

所有读写操作都是通过QSPI_DR寄存器的访问来实现。

通过DMODE寄存器，可以配置数据收发以单线、双线、四线方式进行或者跳过数据阶段。

22.4.5 延迟数据采样

当PCB布线引入较大的时钟数据相位差时，有可能通过数据延迟采样来弥补。正常情况下数据再CLK下降沿发送，接收数据在CLK上升沿被采样；使能了延迟采样后，采样点被人为延迟半个时钟周期，即在下一个CLK的下降沿被采样。

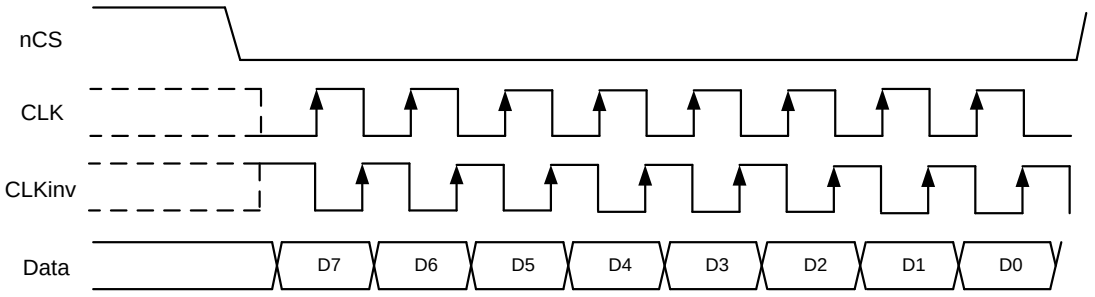


图 22-11 延迟数据采样

22.4.6 BUSY 位和 ABORT 功能

当QuadSPI开始访问QSPI存储器时，BUSY标志寄存器自动置位。

外设模式下，BUSY标志在数据传输完成并且FIFO空的情况下清零。

自动查询模式下，最近一次周期性查询结束并且匹配成功，或者应用执行了ABORT操作后，BUSY标志清零。

存储器映射模式下，BUSY保持高直到timeout或者ABORT。

软件通过置位ABORT寄存器可以立即结束当前通信过程，ABORT置位后nCS被拉高，BUSY清零，FIFO被Flush，但是QuadSPI配置寄存器数据不会被清除，ABORT操作后软件可以随时重新发起新的通信。

22.4.7 nCS 控制

QuadSPI通信中，nCS信号完全由硬件控制。默认情况下nCS为高电平，当QSPI操作开始时nCS拉低，通信结束后nCS拉高。

MODE 0

在mode 0配置下，QSPI_CLK与nCS的时序关系如下图：

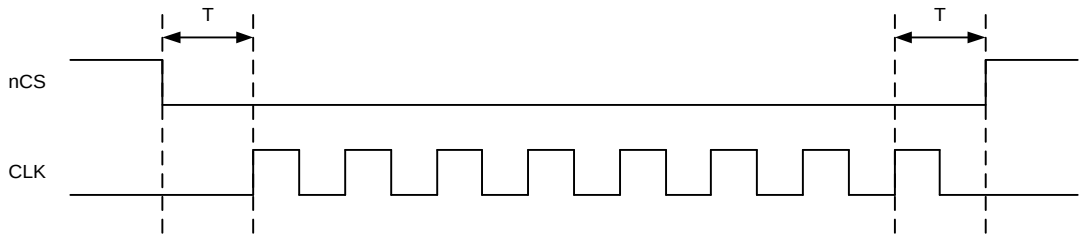


图 22-12 mode 0 nCS 时序

MODE 3

Mode3配置下的nCS与QSPI_CLK时钟的时序关系如下图：

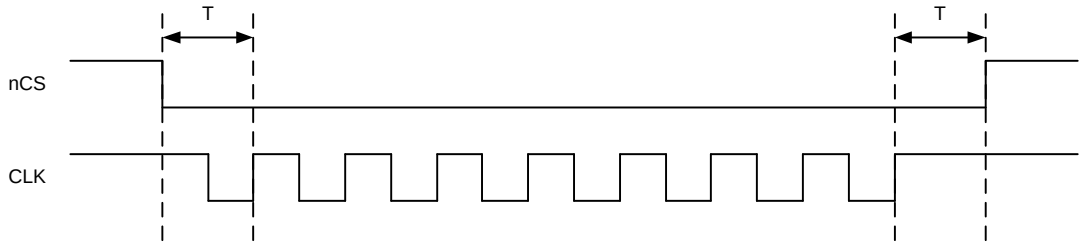


图 22-13 mode 3 nCS 时序

如果在写操作中遇到FIFO空，或者读操作中遇到FIFO满，QSPI_CLK将被暂停，直到应用更新了FIFO状态。ABORT操作会导致nCS和QSPI_CLK回到IDLE状态。

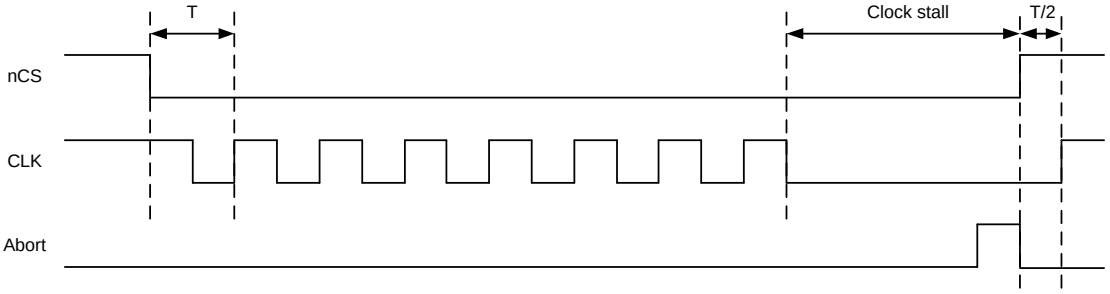


图 22-14 Abort 时序

存储器映射模式下，FIFO满后QuadSPI停止总线操作，此时如果使能了超时功能，超时计数器以QSPI_CLK频率开始计数，计数值达到QSPI_TO设定值之后，认为总线超时，拉高nCS。

22.5 高效的 XIP 实现

QuadSPI通过16字节的FIFO提升XIP效率，一旦AHB对0x9000_0000 ~ 0x9FFF_FFFF地址发起读操作，QuadSPI将自动发起对QSPI存储器的读取，并不断将返回数据PUSH到FIFO里。

AHB触发XIP后，QuadSPI会在当前地址基础上不断递增读取后续地址数据，直到FIFO满才会停止。如果AHB发生任何jump，原有FIFO数据被Flush掉，重新从新地址开始读取。

最高效率的XIP通过与支持"QPI"和"Continuous Read Mode"的NOR Flash配合实现，通过置位CRM寄存器，QuadSPI仅在第一次读取时发送指令，后续读取新的地址时，仅需重新发送目标地址和alternate bytes，无需再发送指令；而支持"QPI"模式的Flash允许主机以四线方式发送指令、地址、alternate bytes和dummy bytes，从而实现最小的读取延迟。

以下图为例，使能了CRM后，发生地址跳转时，从新的地址读取不需要再次发送command。

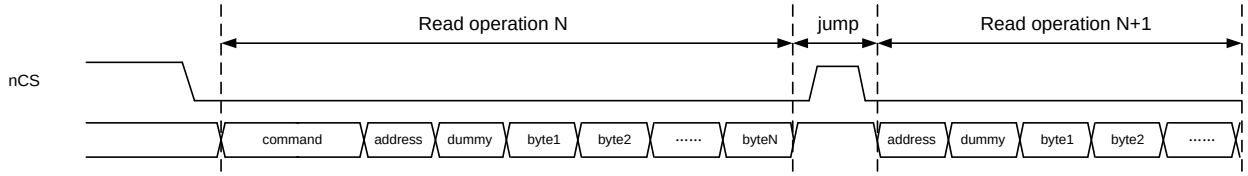


图 22-15 XIP 跳转

22.5.1 应用举例

下面以复旦微电子公司的8M bits SPI串行NOR Flash芯片FM25Q08A为例，说明如何通过QuadSPI接口实现高效的XIP方案。

使能QPI模式

FM25Q08A上电后默认为标志SPI模式，首先要使能其QPI模式。

- 将FM25Q08A的Status Register 2中QE位置为1

上电后Status Register Protect (SRP1, SRP0) = 00，状态寄存器处于软件保护状态；在改写QE寄存器之前，需要首先发送Write Enable (06h)指令，解除软件保护。

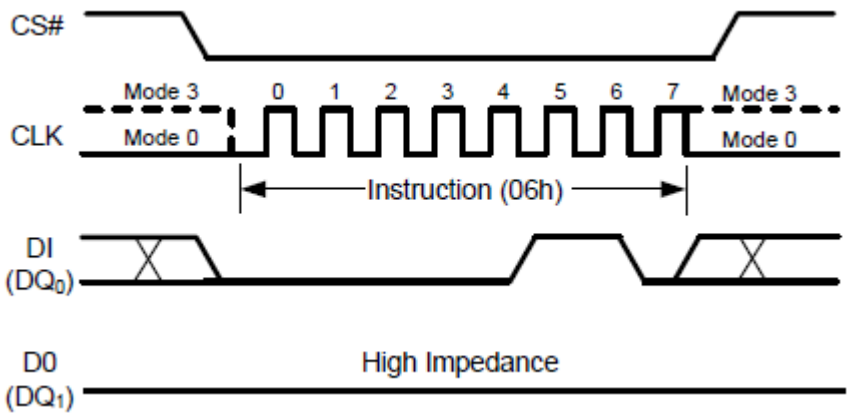


图 22-16 Standard SPI 模式下发送 Write Enable 指令

收到Write Enable指令之后，FM25Q08A的WEL寄存器将被置位，表示已解除写保护。此时，QuadSPI通过发送”Write Status Register2 (31h)”指令将QE位置为1。

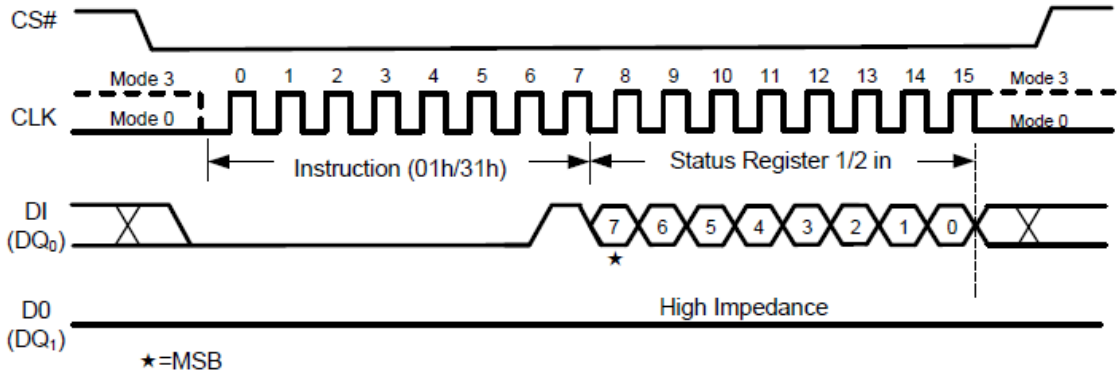


图 22-17 Standard SPI 模式下发送 Write Status Register2 指令

- 发送”Enable QPI (38h)”指令，使能QPI模式

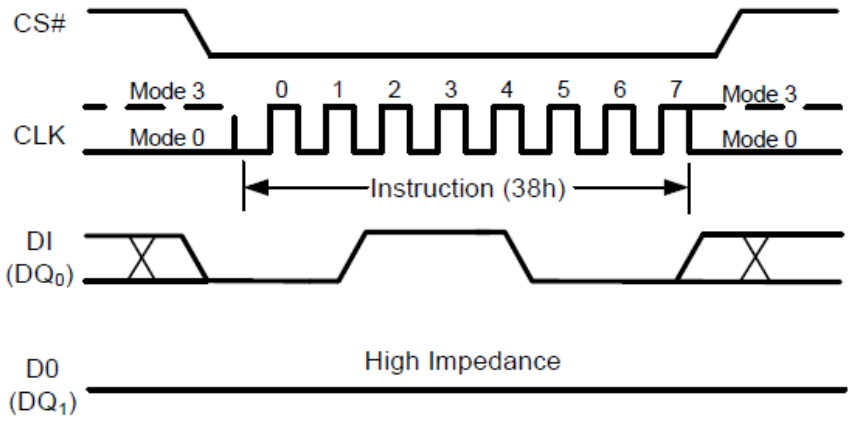


图 22-18 Standard SPI 模式下发送 Enable QPI 指令

完成以上操作后，FM25Q08A被置于QPI模式，后续通信可以采用四线方式进行。

QPI模式读取Flash

推荐使用”Fast Read Quad I/O (Ebh)”指令读取Flash，FM25Q08A的本指令可以支持”Continuous Read Mode”，只需要在第一次读操作时发送指令字节，后续读操作仅需发送地址。

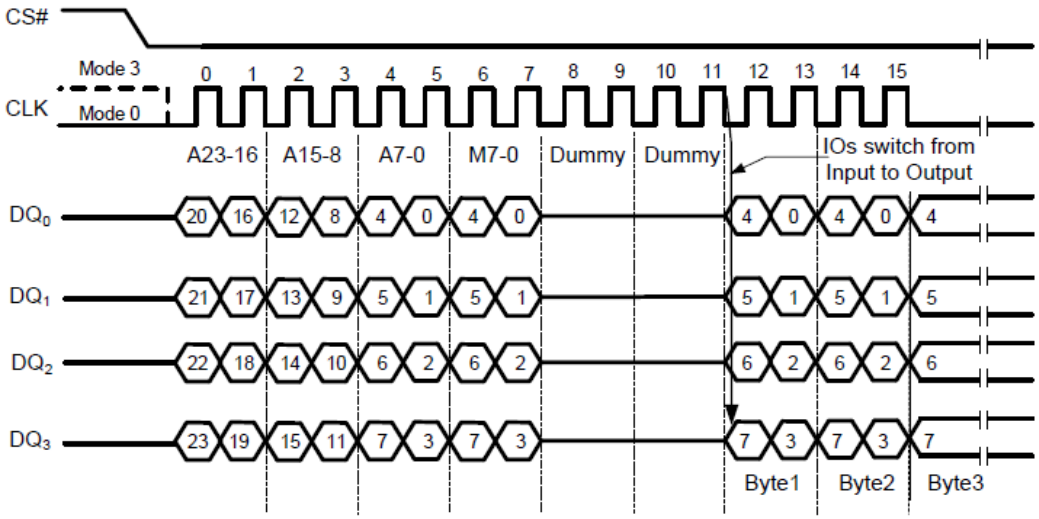


图 22-19 QPI 模式下 Fast Read Quad I/O 操作（前一条指令 M5-4=10）

软件操作方法如下：

- 指令寄存器写入Ebh
- Alternate byte的bit[5:4]写入10，ABSIZE配置为1
- 根据CLK频率配置合适的dummy cycles
- 写入XIP起始地址
- IMODE、ADMODE、ABMODE都配置成11（四线方式）
- 软件跳转到XIP映射地址开始取指

22.6 寄存器

模块基地址：0x4000_0800

offset 地址	名称	符号
QSPI(模块起始地址:0x40000800)		
0x00	QSPI 控制寄存器 (QSPI Control Register)	QSPI_CR
0x04	QSPI 配置寄存器 (QSPI Config Register)	QSPI_CFGR
0x08	QSPI 状态寄存器 (QSPI Status Register)	QSPI_SR
0x0C	QSPI 数据长度寄存器 (QSPI Data Length Register)	QSPI_DATALEN
0x10	QSPI 通信配置寄存器 (QSPI Communication Config Register)	QSPI_CCR
0x14	QSPI 地址寄存器 (QSPI Address Register)	QSPI_ADDR
0x18	QSPI alternate bytes 寄存器 (QSPI Alternate Bytes Register)	QSPI_ABR
0x1C	QSPI 数据寄存器 (QSPI Data Register)	QSPI_DR
0x20	QSPI 状态 MASK 寄存器 (QSPI Status Mask Register)	QSPI_SMSK
0x24	QSPI 状态匹配寄存器 (QSPI Status Match Register)	QSPI_SMAT
0x28	QSPI 轮询间隔寄存器 (QSPI Polling Interval Register)	QSPI_PITV
0x2C	QSPI 超时寄存器 (QSPI Time Out Register)	QSPI_TO

22.6.1 QuadSPI 控制寄存器（QSPI_CR）

名称	QSPI_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PRESCALER							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PMM	-		TOIE	SMIE	FTIE	TCIE	TEIE
位权限	R/W-0	U-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	-				FIFOTHR			
位权限	U-0				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			SSHFT	TCEN	DMAEN	ABORT	EN
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
31:24	**PRESCALER**	QSPI 时钟预分频系数

位号	助记符	功能描述
		00000000：$F_{QSPI_CLK} = F_{AHB}/2$ 00000001：$F_{QSPI_CLK} = F_{AHB}/2$ 00000010：$F_{QSPI_CLK} = F_{AHB}/3$ …… 11111111：$F_{QSPI_CLK} = F_{AHB}/256$ 注意：对于奇数分频，时钟占空比不是 50%，高电平比低电平长一个 AHBCLK 周期
23	**PMM**	轮询匹配模式 0：AND 模式，所有 bit 都匹配才置位 SMF 1：OR 模式，至少 1bit 匹配就会置位 SMF
22:21	**-**	RFU：未实现，读为 0
20	**TOIE**	超时中断使能 0：禁止超时中断 1：允许超时中断
19	**SMIE**	状态匹配中断使能 0：禁止状态匹配中断 1：允许状态匹配中断
18	**FTIE**	FIFO 水位中断使能 0：禁止 FIFO 水位中断 1：允许 FIFO 水位中断
17	**TCIE**	传输完成中断使能 0：禁止传输完成中断 1：允许传输完成中断
16	**TEIE**	传输错误中断使能 0：禁止传输错误中断 1：允许传输错误中断
15:12	**-**	RFU：未实现，读为 0
11:8	**FIFOTHR**	FIFO 水位寄存器，仅外设模式下起效 外设写模式： 0000：FTF 在 FIFO 空字节大于等于 1 时置位 0001：FTF 在 FIFO 空字节大于等于 2 时置位 …… 1111：FTF 在 FIFO 空字节等于 16 时置位 外设读模式： 0000：FTF 在 FIFO 中字节数大于等于 1 时置位 0001：FTF 在 FIFO 中字节数大于等于 2 时置位 …… 1111：FTF 在 FIFO 中字节数等于 16 时置位
7:5	**-**	RFU：未实现，读为 0
4	**SSHFT**	延迟采样使能 0：关闭延迟采样功能 1：使能延迟采样功能
3	**TCEN**	总线超时使能，此寄存器仅在存储器映射模式下有效 当 BUSY 置位后，如果 QuadSPI 不发起对 QSPI 存储器的访问，超时寄存器开始计数，计数溢出长度由 TIMEOUT 寄存器定义。当 QSPI 总线长时间无动作，计数器溢出，nCS 被自动拉高，强制结束当前传输过程。 0：关闭超时功能 1：使能超时功能

位号	助记符	功能描述
2	**DMAEN**	DMA 使能 0：DMA 功能关闭，QuadSPI 不会发送 DMA 请求 1：DMA 功能开启，QuadSPI 在满足条件时发送 DMA 请求
1	**ABORT**	当前传输终止寄存器，软件写 1 终止传输，硬件拉高 nCS 后自动清零
0	**EN**	QuadSPI 模块使能 0：关闭 QuadSPI 1：使能 QuadSPI

22.6.2 QuadSPI 配置寄存器（QSPI_CFGR）

名称	QSPI_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	-					CSHT		
位权限	U-0					R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							CKMODE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:11	**-**	RFU：未实现，读为 0
10:8	**CSHT**	nCS 最小高电平时间，定义了连续两个帧之间 nCS 所需保持高电平的最短时间，以 QSPI_CLK 周期计数 0：至少 1 cycle 1：至少 2 cycles …… 7：至少 8 cycles
7:1	**-**	RFU：未实现，读为 0
0	**CKMODE**	SPI Clock Mode 寄存器 0：mode 0 1：mode 3

22.6.3 QuadSPI 状态寄存器（QSPI_SR）

名称	QSPI_SR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

名称	QSPI_SR							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			FIFOLVL				
位权限	U-0			R-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		BUSY	TOF	SMF	FTF	TCF	-
位权限	U-0		R-0	R/W-0	R/W-0	R-0	R/W-0	U-0

位号	助记符	功能描述
31:13	**-**	RFU：未实现，读为 0
12:8	**FIFOLVL**	FIFO 水位标志 此寄存器表示当前 FIFO 中保存的数据字节数，0 表示 FIFO 空，16 表示 FIFO 满 自动查询模式下此寄存器保持 0
7:6	**-**	RFU：未实现，读为 0
5	**BUSY**	1 表示 QuadSPI 传输进行中，通信结束后自动清零
4	**TOF**	超时标志，硬件置位，软件写 1 清零
3	**SMF**	自动查询模式下表征状态寄存器匹配成功，硬件置位，软件写 1 清零
2	**FTF**	FIFO threshold 标志，FIFO 水位高于设定阈值时自动置位，低于阈值时自动清零 自动查询模式下，每次读回一组状态值后都会自动置位 FTF，如果软件读取 QSPI_DATA 寄存器则 FTF 清零
1	**TCF**	传输完成标志，硬件置位，软件写 1 清零
0	**-**	RFU：未实现，读为 0

22.6.4 QuadSPI 数据长度寄存器（QSPI_DATALEN）

名称	QSPI_DATALEN							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QSPI_DATALEN[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QSPI_DATALEN[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QSPI_DATALEN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_DATALEN[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
31:0	**QSPI_DATALEN**	传输数据长度为 DATALEN+1（bytes）

22.6.5 QuadSPI 通信配置寄存器（QSPI_CCR）

名称	QSPI_CCR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-			CRM	OPMODE		DMODE	
位权限	U-0			R/W-0	R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-	DUMCYC					ABSIZE	
位权限	U-0	R/W-00000					R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ABMODE		ADSIZE		ADMODE		IMODE	
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	INSTRUCTION							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:29	**-**	RFU：未实现，读为 0
28	**CRM**	Continuous Read Mode 0：每次通信发起时都要发送指令 1：只在第一次通信时发送指令
27:26	**OPMODE**	操作模式 00：外设写模式 01：外设读模式 10：自动查询模式 11：存储器映射模式
25:24	**DMODE**	数据通信模式（data phase） 00：无数据 01：单线 10：双线 11：四线
23	**-**	RFU：未实现，读为 0
22:18	**DUMCYC**	Dummy cycle 个数配置（以 QSPI_CLK 周期计算），0~31
17:16	**ABSIZE**	Alternate bytes 个数 00：8bits alternate bytes 01：16bits alternate bytes 10：24bits alternate bytes 11：32bits alternate bytes
15:14	**ABMODE**	Alternate bytes 发送模式 00：无 alternate bytes 01：单线 10：双线 11：四线
13:12	**ADSIZE**	地址字节长度

位号	助记符	功能描述
		00：8bits 地址 01：16bits 地址 10：24bits 地址 11：32bits 地址
11:10	**ADMODE**	地址字节发送模式 00：无地址字节 01：单线 10：双线 11：四线
9:8	**IMODE**	指令发送模式 00：无指令字节 01：单线 10：双线 11：四线
7:0	**INSTRUCTION**	QuadSPI 发送的指令字节

22.6.6 QuadSPI 地址寄存器（QSPI_ADDR）

名称	QSPI_ADDR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QSPI_ADDR[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QSPI_ ADDR[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QSPI_ ADDR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_ ADDR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**QSPI_ADDR**	发送给 QSPI 存储器的地址，在存储器映射模式下无效

22.6.7 QuadSPI alternate bytes 寄存器（QSPI_ABR）

名称	QSPI_ABR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QSPI_ABR[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QSPI_ ABR[23:16]							
位权限	R/W-0000 0000							

名称	QSPI_ABR							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QSPI_ ABR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_ ABR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**QSPI_ABR**	发送给 QSPI 存储器的 alternate bytes

22.6.8 QuadSPI 数据寄存器（QSPI_DR）

名称	QSPI_DR							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QSPI_DATA[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QSPI_DATA[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QSPI_DATA[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_DATA[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**QSPI_DATA**	QSPI 数据寄存器 外设模式写操作时，对 QSPI_DR 寄存器写入的数据将被 push FIFO，支持字节、半字、字写入，分别对 FIFO 压入 1、2、4 字节；如果写入字节数大于 FIFO 中空字节数，当前写操作被挂起，直到 FIFO 中有足够空间容纳当前写入数据。 外设模式读操作时，读取 QSPI_DR 寄存器将 pop FIFO，支持字节、半字、字读取，分别从 FIFO 弹出 1、2、4 字节；如果读取字节数大于 FIFO 中有效字节数，当前读操作被挂起，直到 FIFO 中有足够字节可以被读取，或者传输完成，后一种情况下只弹出最后几个实际有效字节。 对 QSPI_DR 的访问必须对齐低地址，即字节访问必须对齐 QSPI_DR[7:0]，半字访问必须对齐 QSPI_DR[15:0]

22.6.9 QuadSPI 状态 MASK 寄存器（QSPI_SMSK）

名称	QSPI_SMSK							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

名称	QSPI_SMSK							
位名	QSPI_SMSK[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QSPI_SMSK[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QSPI_SMSK[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_SMSK[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**QSPI_SMSK**	自动状态查询模式下的状态 mask 寄存器，对应 bit 写 0 屏蔽相应状态位

22.6.10 QuadSPI 状态匹配寄存器（QSPI_SMAT）

名称	QSPI_SMAT							
offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QSPI_SMAT[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QSPI_SMAT[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QSPI_SMAT[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_SMAT[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**QSPI_SMAT**	自动状态查询模式下的状态匹配寄存器 比较对象是 QSPI_DATA & QSPI_SMSK

22.6.11 QuadSPI 轮询间隔寄存器（QSPI_PITV）

名称	QSPI_PITV							
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	-							

名称	QSPI_PITV							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QSPI_PITV[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_PITV[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	**QSPI_PITV**	自动状态查询模式下的轮询间隔（polling interval），定义为 QSPI_CLK 周期数

22.6.12 QuadSPI 超时寄存器（QSPI_TO）

名称	QSPI_TO							
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	QSPI_TO[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_TO[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	**QSPI_TO**	超时周期设置，定义为 QSPI_CLK 周期数，仅在存储器映射模式下有效 当 FIFO 满之后，QSPI 总线行为停止，超时计数器开始计数，计数值到达 QSPI_TO 设定值之后，拉高 nCS

23 直接存储访问控制器 1（DMA1）

23.1 概述

- 11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8192字节（8KB），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）
- 接入外设可选择为SPIx、UARTx、I2C(Master/Slave)、LPUARTx、ADC、AES、CRC等

23.2 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。

23.3 结构框图

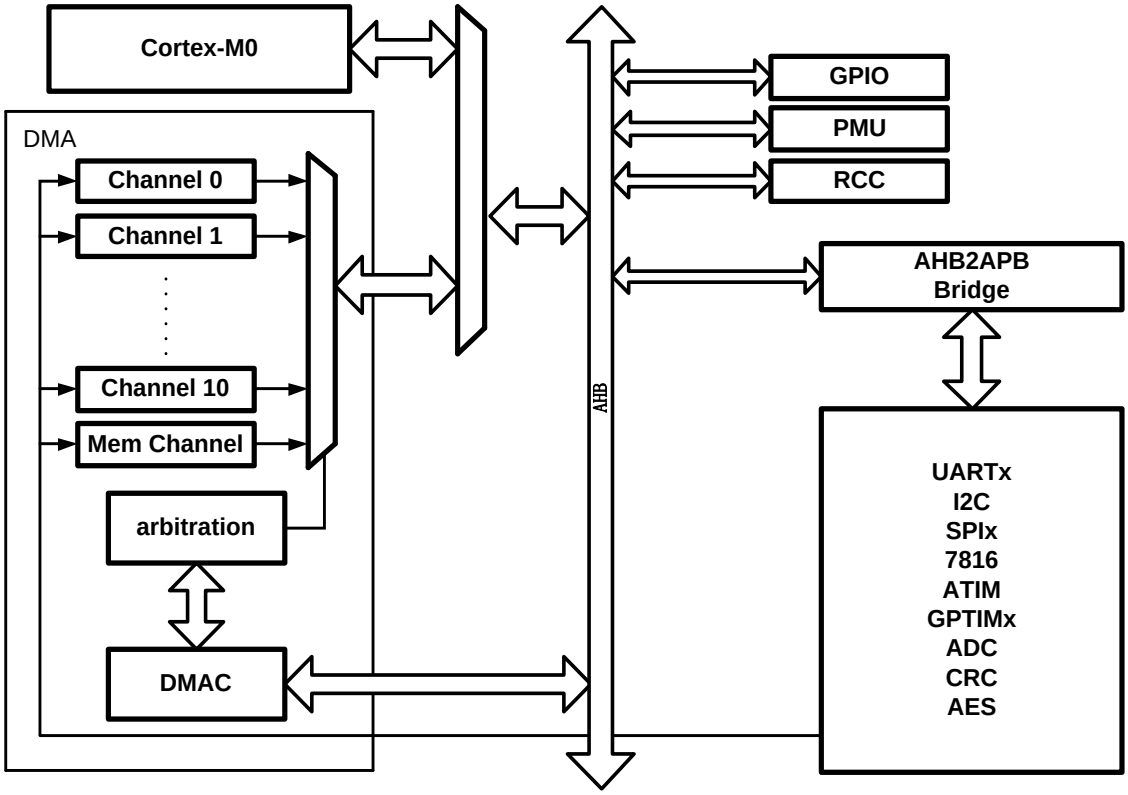


图 23-1 DMA1 结构框图

23.4 工作流程

DMA 寄存器配置：



图 23-2 DMA 寄存器配置

DMA 对请求响应分成两部分处理：通道请求处理过程和数据搬运过程。

- 通道请求处理

  a) DMA 接受到请求，跳到步骤 b

  b) 判断是否有其他通道正在搬运数，若有，则停留在步骤 b 直至其他通道当次搬运完成；若无，进一步判断是否有其他同时置起的请求信号，若有，则判断当前通道优先级是否高于其他通道，若是，则跳到步骤 c 并向数据搬运过程发起请求，若否，则停留在步骤 b 直至其他通道当次搬运完成

  c) 并等待数据搬运完成响应信号，得到响应则，跳到步骤 d，否则停在步骤 c

  d) 数据搬运长度+1，判断是否达到设定长度，若是则产生通道使能关闭脉冲；判断请求是否释放，若是，则跳到步骤 a，若否，则停留在步骤 d 判断数据传输达到设定长度，否则跳到步骤 a

- 数据搬运

  a) 等待通道请求处理过程发起请求

  b) 向 HADDR 写源地址

  c) 向 HADDR 写目的地址，同时读取 HRDATA

  d) 将读到的 HRDATA 数据写到 HWDATA

e) 向通道请求处理过程发出搬运完成响应，并跳到步骤 a

DMA 工作的流程如下图所示：

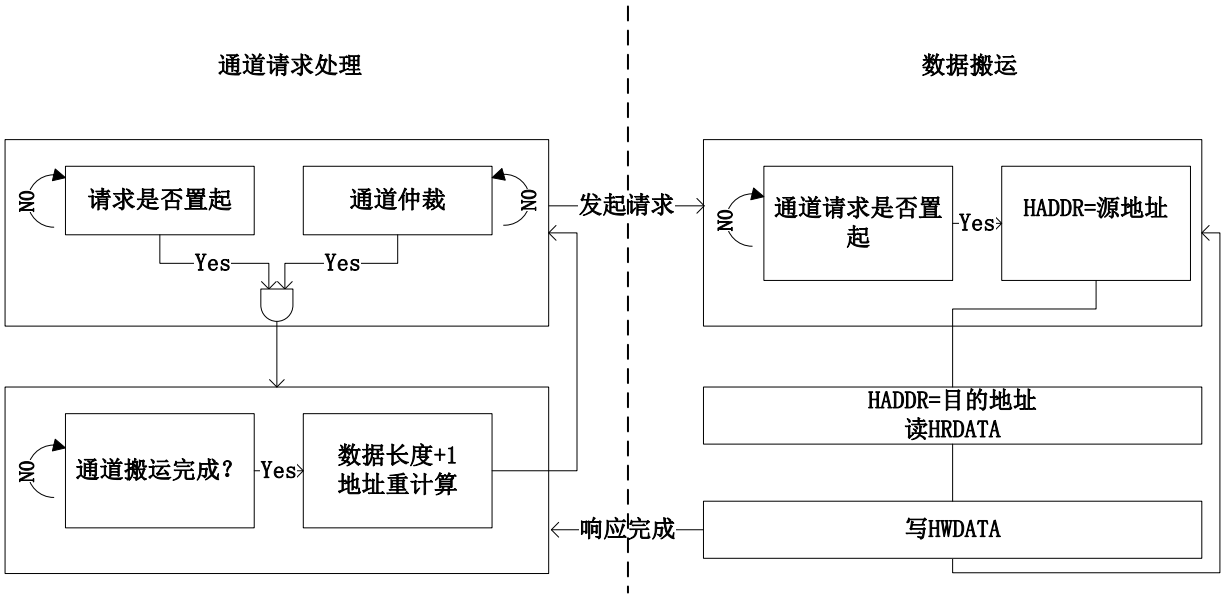


图 23-3 DMA 工作流程

23.5 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。

外设使用DMA进行数据通信时，需正确配置传输带宽。比如UART配置接收数据长度为8bit时，需采用字节传输，而如果接收数据长度为9bit，则应当使用半字传输。

23.6 通道控制

23.6.1 DMA 请求映射

DMA 共有 10 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	外设请求	通道0	通道1	通道2	通道3	通道4	通道5	通道6	通道7	通道8	通道9	通道10
0	***ADC***	ADC	ADC						ADC	ADC		
1	***SPI0***		SPI0_RX	SPI0_TX			SPI0_RX	SPI0_TX			SPI0_RX	SPI0_TX
2	***SPI1***				SPI1_RX	SPI1_TX			SPI1_RX	SPI1_TX		
3	***SPI2***		SPI2_RX	SPI2_TX			SPI2_RX	SPI2_TX		SPI2_RX	SPI2_TX	
4	***SPI3***	SPI3_RX	SPI3_TX			SPI3_RX			SPI3_RX	SPI3_TX		
5	***SPI4***			SPI4_RX	SPI4_TX						SPI4_RX	SPI4_TX
6	***LPUART0***	LPUART0_RX			LPUART0_RX	LPUART0_TX			LPUART0_RX	LPUART0_TX		
7	***LPUART1***	LPUART1_RX					LPUART1_RX	LPUART1_TX			LPUART1_RX	LPUART1_TX
8	***UART0***		RXD0	TXD0					RXD0	TXD0		
9	***UART1***				RXD1	TXD1	RXD1	TXD1			RXD1	TXD1
10	***UART2***	RXD2	TXD2			RXD2	TXD2		RXD2	TXD2		
11	***UART3***		RXD3	TXD3			RXD3	TXD3			RXD3	TXD3
12	***UART4***	RXD4			RXD4	TXD4			RXD4	TXD4		
13	***UART5***			RXD5	TXD5		RXD5	TXD5			RXD5	TXD5
14	***U7816***		U7816RX	U7816TX								
15	***I2C0***				I2C0_TX	I2C0_RX					I2C0_TX	I2C0_RX
16	***I2C1***						I2C1_TX	I2C1_RX				
17	***AES***	AES_IN		AES_OUT	AES_IN	AES_OUT						
18	***CRC***							CRC				CRC
19	***QSPI***	QSPI							QSPI			

表 23-1 DMA1 通道映射

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=3时，被选中的外设请求是LPUART1_RX，即LPUART1的数据接收DMA请求被连接到DMA通道0的请求输入。

23.6.2 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, medium, 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。

23.6.3 传输方向定义

DMA 传输方向由通道控制寄存器的 DIR 位控制，可以选择数据传输方向为外设->RAM，或者 RAM->外设。当某个通道所使能的触发信号到来时，DMA 引擎根据 DIR 寄存器设定的传输方向进行数据搬运。

比如希望使用 DMA 进行 UART 数据接收，则需要将对应通道的 DIR 寄存器设置为 0，即从 UART 读取接收数据写入到 RAM 中。假设希望使用 DMA 进行 UART 发送，则应将对应通道的 DIR 寄存器设置为 1。

23.6.4 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。

23.6.5 影子寄存器

针对部分通道控制寄存器和存储器指针寄存器（通道 0~6），拥有影子寄存器；在循环模式下，如果使能了自动更新功能（CIRC_UPD），当本轮传输结束后，硬件会自动将影子寄存器中的内容拷贝到控制寄存器和存储器指针寄存器中，这样在新的一轮传输中，会自动使用新的传输控制参数。

软件可以在半程中断中更新影子寄存器，在循环模式下实现实时调整下一轮传输参数的目的。

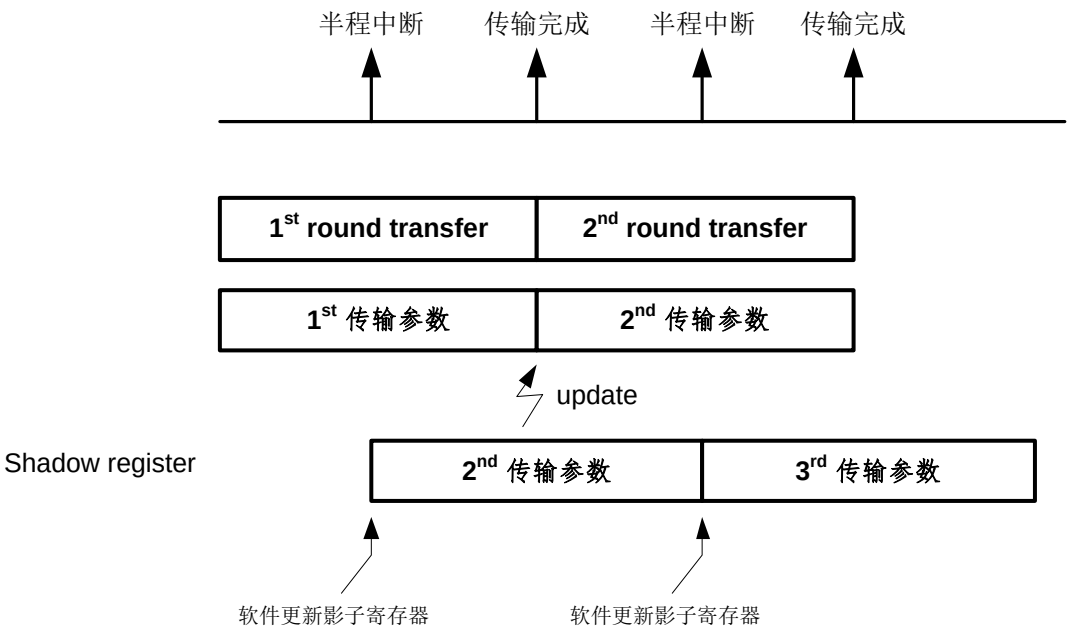


图 23-4 DMA 循环模式下自动更新传输参数

23.7 寄存器

模块基地址：0x4000_0400

offset 地址	名称	符号
DMA(模块起始地址:0x40000400)		
0x0000	DMA 全局控制寄存器（DMA Global Control Register）	DMA_GCR
0x0004	通道 0 控制寄存器（Channel 0 Control Register）	DMA_CH0CR
0x0008	通道 0 存储器指针寄存器（Channel 0 Memory Address Register）	DMA_CH0MAR
0x000C	通道 1 控制寄存器（Channel 1 Control Register）	DMA_CH1CR
0x0010	通道 1 存储器指针寄存器（Channel 1 Memory Address Register）	DMA_CH1MAR
0x0014	通道 2 控制寄存器（Channel 2 Control Register）	DMA_CH2CR
0x0018	通道 2 存储器指针寄存器（Channel 2 Memory Address Register）	DMA_CH2MAR
0x001C	通道 3 控制寄存器（Channel 3 Control Register）	DMA_CH3CR
0x0020	通道 3 存储器指针寄存器（Channel 3 Memory Address Register）	DMA_CH3MAR
0x0024	通道 4 控制寄存器（Channel 4 Control Register）	DMA_CH4CR
0x0028	通道 4 存储器指针寄存器（Channel 4 Memory Address Register）	DMA_CH4MAR
0x002C	通道 5 控制寄存器（Channel 5 Control Register）	DMA_CH5CR
0x0030	通道 5 存储器指针寄存器（Channel 5 Memory Address Register）	DMA_CH5MAR
0x0034	通道 6 控制寄存器（Channel 6 Control Register）	DMA_CH6CR
0x0038	通道 6 存储器指针寄存器（Channel 6 Memory Address Register）	DMA_CH6MAD
0x003C	通道 7 控制寄存器（Channel 7 Control Register）	DMA_CH7CR
0x0040	通道 7 存储器指针寄存器（Channel 7 Memory Address Register）	DMA_CH7MAR
0x0044	通道 8 控制寄存器（Channel 8 Control Register）	DMA_CH8CR
0x0048	通道 8 存储器指针寄存器（Channel 8 Memory Address	DMA_CH8MAR

offset 地址	名称	符号
DMA(模块起始地址:0x40000400)		
	Register）	
0x004C	通道 9 控制寄存器 （Channel 9 Control Register）	DMA_CH9CR
0x0050	通道 9 存储器指针寄存器 （Channel 9 Memory Address Register）	DMA_CH9MAR
0x0054	通道 10 控制寄存器 （Channel 10 Control Register）	DMA_CH10CR
0x0058	通道 10 存储器指针寄存器 （Channel 10 Memory Address Register）	DMA_CH10MAR
0x005C	通道 11 控制寄存器 （Channel 11 Control Register）	DMA_CH11CR
0x0060	通道 11 Flash 指针寄存器 （Channel 11 Flash Address Register）	DMA_CH11FAR
0x0064	通道 11 RAM 指针寄存器 （Channel 11 RAM Address Register）	DMA_CH11RAR
0x0068	DMA 中断状态标志寄存器 （DMA Interrupt Status Register）	DMA_ISR
0x0100	通道 0 控制影子寄存器 （Channel 0 Control Shadow Register）	DMA_CH0CSR
0x0104	通道 0 存储器指针影子寄存器 （Channel 0 Memory Address Shadow Register）	DMA_CH0MASR
0x0108	通道 1 控制影子寄存器 （Channel 0 Control Shadow Register）	DMA_CH1CSR
0x010C	通道 1 存储器指针影子寄存器 （Channel 0 Memory Address Shadow Register）	DMA_CH1MASR
0x0110	通道 2 控制影子寄存器 （Channel 0 Control Shadow Register）	DMA_CH2CSR
0x0114	通道 2 存储器指针影子寄存器 （Channel 0 Memory Address Shadow Register）	DMA_CH2MASR
0x0118	通道 3 控制影子寄存器 （Channel 0 Control Shadow Register）	DMA_CH3CSR
0x011C	通道 3 存储器指针影子寄存器 （Channel 0 Memory Address Shadow Register）	DMA_CH3MASR
0x0120	通道 4 控制影子寄存器 （Channel 0 Control Shadow Register）	DMA_CH4CSR
0x0124	通道 4 存储器指针影子寄存器 （Channel 0 Memory Address	DMA_CH4MASR

offset 地址	名称	符号
DMA(模块起始地址:0x40000400)		
	Shadow Register）	
0x0128	通道 5 控制影子寄存器（Channel 0 Control Shadow Register）	DMA_CH5CSR
0x012C	通道 5 存储器指针影子寄存器（Channel 0 Memory Address Shadow Register）	DMA_CH5MASR
0x0130	通道 6 控制影子寄存器（Channel 0 Control Shadow Register）	DMA_CH6CSR
0x0134	通道 6 存储器指针影子寄存器（Channel 0 Memory Address Shadow Register）	DMA_CH6MASR

23.7.1 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
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						DMA_ADDRERR_EN	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	**DMA_ADDRERR_EN**	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

23.7.2 通道 x 控制寄存器（DMA_CHxCR）

名称	DMA_CHxCR(x=0,1,2,3,4,5,6,7,8,9,10)							
offset	0x04+x*0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

名称	DMA_CHxCR(x=0,1,2,3,4,5,6,7,8,9,10)							
位名	-			CHxTSIZE[12:8]				
位权限	U-0			R/W-00000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CHxTSIZE[7:0]							
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	CIRC_UPD	DIR	BDW		CIRC	CHxFTIE	CHxHTIE	CHxEN
位权限	R/W-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
31:29	**-**	RFU：未实现，读为 0
28:16	**CHxTSIZE**	Channelx 传输长度，1-8192 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
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 23.6.1DMA 请求映射
7	**CIRC_UPD**	循环缓冲模式下自动更新传输参数，本轮传输完成后将 shadow 寄存器内容复制到控制寄存器 0：禁止自动更新 1：使能自动更新 注意：仅通道 0~6 支持此功能，通道 7~10 无此寄存器
6	**DIR**	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	**BDW**	传输带宽设置 00：字节，8bit 01：半字，16bit 10：字，32bit 11：RFU
3	**CIRC**	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

位号	助记符	功能描述
		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

23.7.3 通道 x 存储器指针寄存器（DMA_CHxMAR）

名称	DMA_CHxMAR(x=0,1,2,3,4,5,6,7,8,9,10)							
offset	0x08+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。 *注意：此指针禁止指向 0x00080000~0x1FFFFFFF 地址，这段地址为 flash 保留信息区，指向这段地址可能在 DMA 访问中导致不可预计的结果*

23.7.4 通道 11 控制寄存器（DMA_CH11CR）

名称	DMA_CH11CR							
offset	0x5C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-			CH11TSIZE[12:8]				
位权限	U-0			R/W-00000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CH11TSIZE[7:0]							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		CH11PRI		-	CH11DIR	CH11RI	CH11FI
位权限	U-0		R/W-00		U-0	R/W-0	R/W-0	R/W-0

名称	DMA_CH11CR							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					CH11FTIE	CH11HTIE	CH11EN
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:29	-	RFU：未实现，读为 0
28:16	**CH11TSIZE**	Channel11 传输长度，1-8192 次传输，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	**CH11PRI**	Channel11 优先级 00：Low 01：Medium 10：High 11：Very High
11	-	RFU：未实现，读为 0
10	**CH11DIR**	Channel11 传输方向 1：Flash->RAM 传输 0：RAM->Flash 传输
9	**CH11RI**	Channel11 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	**CH11FI**	Channel11 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	**CH11FTIE**	Channel11 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	**CH11HTIE**	Channel11 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	**CH11EN**	Channel11 使能 1：启动通道 0 0：关闭通道 0

23.7.5 通道 11 Flash 指针寄存器（DMA_CH11FAR）

名称	DMA_CH11FAR							
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	-	CH11FLSAD[14:8]						

名称	DMA_CH11FAR							
位权限	U-0	R/W-0000000						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CH11FLSAD[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	**CH11FLSAD**	Channel11 Flash 指针地址，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

23.7.6 通道 11 RAM 指针寄存器（DMA_CH11RAR）

名称	DMA_CH11RAR							
offset	0x64							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	CH7RAMAD[14:8]						
位权限	U-0	R/W-0000000						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CH7RAMAD[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	**CH7RAMAD**	Channel7 RAM 字指针地址，DMA 传输启动前软件向此寄存器写入 RAM 目标地址（word 地址），DMA 启动后此寄存器随 DMA 传输自增或自减 软件可以查询当前 DMA 传输的目标 RAM 地址

23.7.7 DMA 中断状态标志寄存器（DMA_ISR）

名称	DMA_ISR							
offset	0x68							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-			DMA_ADDRERR	DMACHFT[11:8]			
位权限	U-0			R/W-0	R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

名称	DMA_ISR							
位名	DMACHFT[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				DMACHHT[11:8]			
位权限	U-0				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DMACHHT[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:29	-	RFU：未实现，读为 0
28	**DMA_ADDRERR**	DMA 传输地址错误标志，当存储器指针超过 RAM 和 Flash 合法地址范围时置位
27:16	**DMACHFT**	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
15:12	-	RFU：未实现，读为 0
11:0	**DMACHHT**	DMA 通道 x 传输半程标志，硬件置位，软件写 1 清零

23.7.8 通道 x 控制影子寄存器（DMA_CHxCSR）

名称	DMA_CHxCSR(x=0,1,2,3,4,5,6)							
offset	0x100 + x*0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-			CHxTSIZE_SDW[12:8]				
位权限	U-0			R/W-00000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CHxTSIZE_SDW[7:0]							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				CHxINC_SDW	-		
位权限	U-0				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
31:29	-	RFU：未实现，读为 0
28:16	**CHxTSIZE_SDW**	Channelx 传输长度 shadow 寄存器；循环模式下如果使能了 CIRC_UPD 寄存器，则在本轮传输完成后将 shadow 寄存器值复制到 CHxTSIZE 中。
15:12	-	RFU：未实现，读为 0
11	**CHxINC_SDW**	RAM 地址增减设置 shadow 寄存器，循环模式下如果使能了 CIRC_UPD 寄存器，则在本轮传输完成后将 shadow 寄存器值复制到 CHxTSIZE 中。
10:0	-	RFU：未实现，读为 0

23.7.9 通道 x 存储器指针影子寄存器（DMA_CHxMASR）

名称	DMA_CHxMASR(x=0,1,2,3,4,5,6)							
offset	0x104 + x*0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CHxMAD_SDW[31:24]							
位权限	R/W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CHxMAD_SDW[23:16]							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CHxMAD_SDW[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CHxMAD_SDW[7:0]							
位权限	R/W-00000000							

位号	助记符	功能描述
31:0	**CHxMAD_SDW**	Channelx 存储器指针地址影子寄存器。 循环模式下如果使能了 CIRC_UPD 寄存器，则在本轮传输完成后将 shadow 寄存器值复制到 CHxMEMAD 中。

24 循环冗余校验（CRC）

24.1 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，FM33A0xxEV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

24.2 软件配置过程

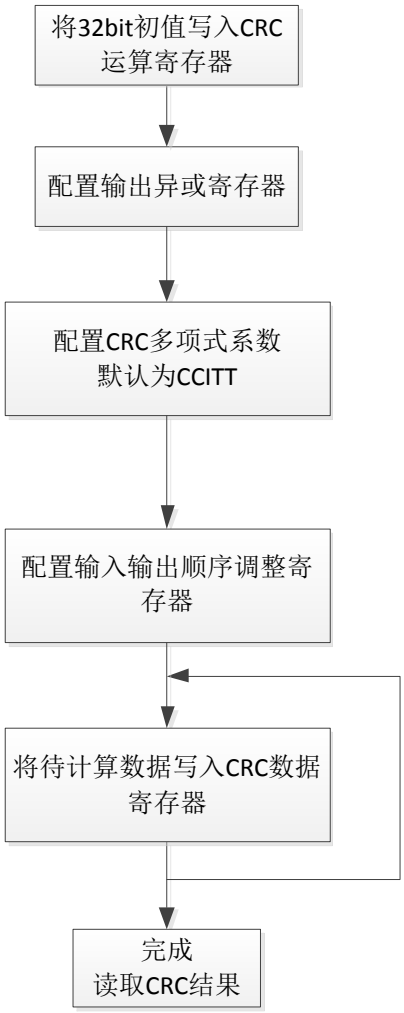


图 24-1CRC 运算流程图

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。

24.3 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

表 24-1 CRC golden 参考数据

24.4 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。

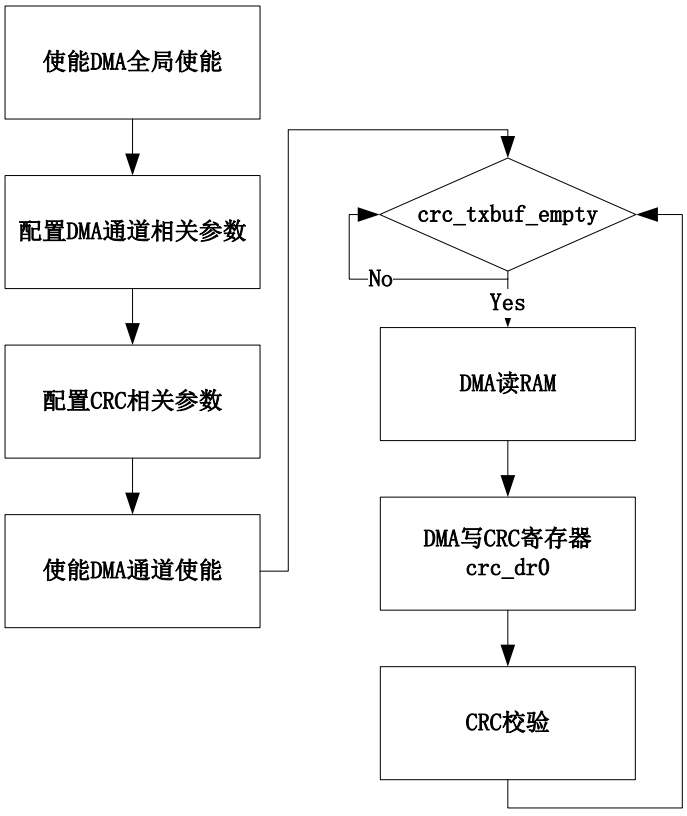


图 24-2 使用 DMA 对 RAM 数据进行 CRC

24.5 Flash 数据完整性校验

Flash内容的CRC校验可以在上电后进行，以保证内容正确性。CRC模块读取Flash时起始地址必须对齐word边界，每读一次flash，需要运算4次8bit CRC。

Flash内容校验必须通过DMA进行，即DMA先将待校验数据搬运到RAM，然后再从RAM搬运到CRC。

24.6 寄存器

模块起始地址：0x4001_0000

offset 地址	名称	符号
CRC(模块起始地址:0x40010000)		
0x0000	CRC 数据寄存器 （CRC Data Register）	CRC_DR
0x0004	CRC 控制状态寄存器 （CRC Control Register）	CRC_CR
0x0008	CRC LFSR 寄存器 （CRC Linear Feedback Shift Register）	CRC_LFSR
0x000C	CRC 输出异或寄存器 （CRC output XOR Register）	CRC_XOR
0x001C	CRC 多项式寄存器 （CRC Polynominal Register）	CRC_POLY

24.6.1 CRC 数据寄存器（CRC_DR）

名称	CRC_DR							
offset	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DR[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DR[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	**DR**	CRCDR 用于作为数据输入寄存器，并且在运算结束后保存 CRC 计算结果。(CRC Data Register) 作为输入时：若 word 操作使能，则对 CRCDR[31:0]进行计算，共 4 次 byte 运算（由低到高）；否则对 CRCDR[7:0]进行计算，共 1 次 byte 运算。 保存结果时：若为 7 位多项式结果保存在 CRCDR[6:0]，若为 8 位多项式结果保存在 CRCDR[7:0]，若为 16 位多项式结果保存在 CRCDR[15:0]，若为 32 位多项式结果保存在 CRCDR[31:0]。

24.6.2 CRC 控制状态寄存器（CRC_CR）

名称	CRC_CR							
offset	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	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	SEL	
位权限	R/W-00		R/W-0	R-0	R-0	R/W-0	R/W-01	

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9	**OPWD**	WORD 操作使能 (Operation by Word) 0：字节操作，CRC 计算仅针对 CRCDR 最低字节进行 1：字操作，CRC 计算针对 CRCDR 全部 4 字节进行
8	**PARA**	CRC 快速计算使能 (Parallel Calculation) 0：串行运算，计算 1 个字节需要 8 个时钟周期 1：并行计算，计算 1 个字节需要 1 个时钟周期
7:6	**RFLTIN**	CRC 输入反转控制 (Reflected Input) 00：输入不反转 01：输入按字节反转 10：输入按半字反转 11：输入按字反转 例如：计算数据为 0x11223344， 如果 RFLTIN==01，则将数据变为 0x8844CC22，再进行计算 如果 RFLTIN==10，则将数据变为 0x448822CC，再进行计算 如果 RFLTIN==11，则将数据变为 0x22CC4488，再进行计算
5	**RFLTO**	CRC 输出反转控制 (Reflected Output) 0：输出不反转 1：输出按字节反转 例如： 如果 RFLTO==1，若当前计算的 CRC 结果为 0x1234，则输出的结果为 0x2C48 如果 RFLTO==0，则直接输出 0x1234 注意：此结果不一定为最终输出结果，还需要看 XOR 是否为 1，详见本寄存器 bit2 说明
4	**RES**	CRC 结果标志位，只读 (Result Flag) 0：CRC 结果为 0 1：CRC 结果非全 0
3	**BUSY**	CRC 运算标志位，只读 (Busy) 0：CRC 运算结束 1：CRC 运算进行中
2	**XOR**	输出异或使能 (Output XORed with CRC_XOR register enable) 0：输出不异或 CRC_XOR 寄存器 1：输出异或 CRC_XOR 寄存器
1:0	**SEL**	CRC 多项式位宽选择 (Polynomial width Selection) 00：32 位 01：16 位

位号	助记符	功能描述
		10：8 位 11：7 位

24.6.3 CRC LFSR 寄存器（CRC_LFSR）

名称	CRC_LFSR							
offset	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LFSR[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LFSR[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LFSR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LFSR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	**LFSR**	CRC 线性反馈移位寄存器 (Linear Feedback Shift Register) 运算开始前可以由软件写入 CRC 初始值

24.6.4 CRC 输出异或寄存器（CRC_XOR）

名称	CRC_XOR							
offset	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	XOR[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	XOR[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	XOR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	XOR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**XOR**	CRC 运算结果异或寄存器 (eXclusive OR) 当 CRC_CR.XOR 为 1 时，CRC 结果输出前将异或此寄存器的数据。

24.6.5 CRC 多项式寄存器（CRC_POLY）

名称	CRC_POLY							
offset	0x001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	POLY[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	POLY[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	POLY[15:8]							
位权限	R/W-0001 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	POLY[7:0]							
位权限	R/W-0010 0001							

位号	助记符	功能描述
31:0	**POLY**	CRC 运算多项式系数 (CRC Polynominals)

25 通用定时器组（Timer Array）

25.1 功能描述

定时器阵列TMRA共包括4个8bit Basic Timer（可级联成2个16bit Timer），4个16bit Extended Timer。如下图所示：

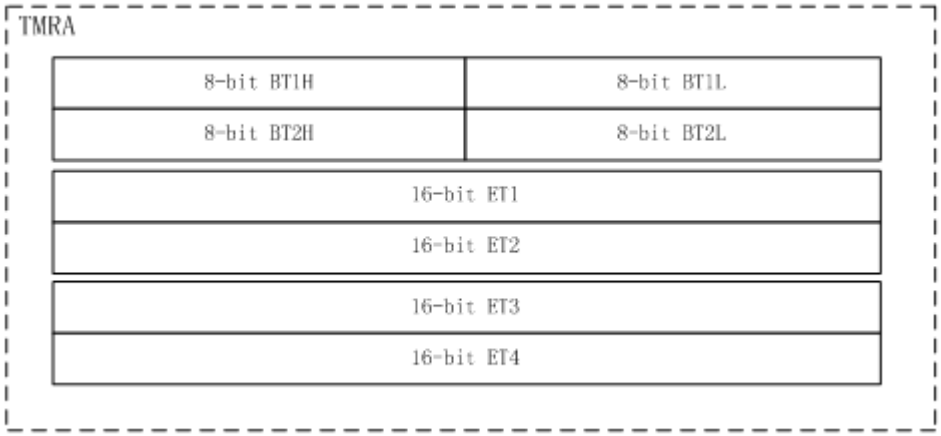


图 25-1 通用定时器组

通用定时器特点：

- 丰富的计数源选择
  - ✓ 在选择内部信号作为定时器的计数源时可选择丰富的计数源信号
- 输入控制
  - ✓ 可选择外部输入信号作为定时器的输入信号
  - ✓ 外部输入信号计数沿可设
  - ✓ 可实现电能表专用的双输入信号相加并实现方向检测
- Basic Timer
  - ✓ 4个8位定时/计数器，时钟源/计数源可分别选择，均支持预置数、重加载、软件加载等功能，可级联成2个16位定时/计数器；
  - ✓ 包含8位初值寄存器和比较寄存器，支持计数、输入捕捉、输出比较和PWM
  - ✓ 可配置成支持电能表专用的电量脉冲计数器，计数匹配信号可由输出引脚输出，输出信号可选择电平和脉冲，电平和脉冲的极性可设，脉冲宽度可设置；
  - ✓ 捕捉功能仅支持16位捕捉模式，捕捉模式支持多周期捕捉（由预分频器配合）、周期捕捉、脉宽捕捉等，均支持清零和不清零模式，支持单次捕捉；
  - ✓ 捕捉模式下支持DMA接口（Half-word传输，Timer->RAM）
- Extended Timer

- ✓ 4个16bit扩展定时器
- ✓ 每个扩展定时器包含一个16位计数器、一个16位初值寄存器和一个16位比较寄存器，初值寄存器同时作为捕捉寄存器（存放捕捉值）；
- ✓ 允许在计数过程中改变计数初值，完成动态的定时时间更新；
- ✓ 捕捉模式下有两种模式，带清零的捕捉模式和不带清零的捕捉模式
- ✓ 支持PWM
- ✓ 可级联为32位定时器使用

- 中断系统
  - ✓ 提供丰富的中断源和中断使能信号供MCU控制和查询，提高系统响应的实时性

25.2 Basic Timer 模块

25.2.1 概述

Basic Timer是4个相同的8bit计数器，可级联成2个16bit定时器（BT1和BT2）。每个16bit定时器各带有一个8bit预分频器。

作为8位定时/计数功能时两个8位定时/计数器的时钟源/计数源可分别选择，均支持预置数、重加载、软件加载等功能；计数功能可配置成支持电能表专用的电量脉冲计数器，计数匹配信号可由输出引脚输出，输出信号可选择电平和脉冲，电平和脉冲的极性可设，脉冲宽度可设置。

捕捉功能仅支持16位捕捉模式，捕捉模式支持多周期捕捉、周期捕捉、脉宽捕捉等，均支持带清零和不清零清零模式；支持单次捕捉，即捕捉一次后停止捕捉功能，方便软件处理。

Timer输入信号支持8bit边沿预分频，可对输入信号的上升沿或下降沿进行预分频，以获得更灵活的计数源和捕捉源。

25.2.2 结构框图

BT1的结构框图，其中CNTL和CNTH代表2个8bit计数器：

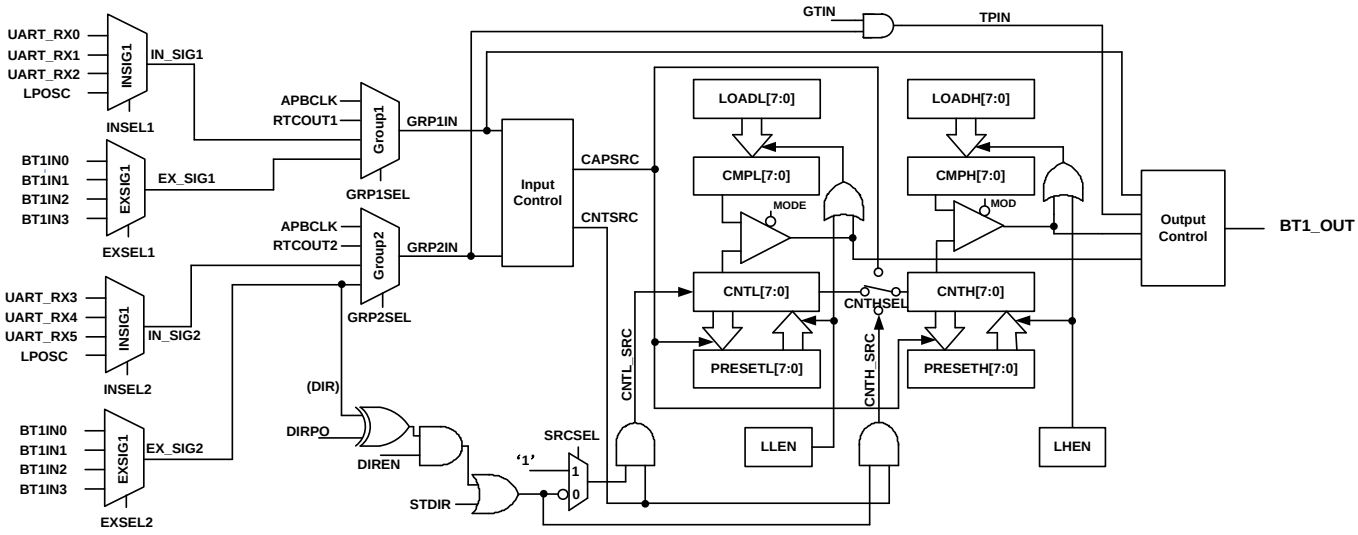


图 25-2 BT1 结构图

BT1支持使用外部输入的信号（DIR）来分别控制高低位计数器计数。

BT2的结构框图，与BT1类似，只是输入信号源有所不同：

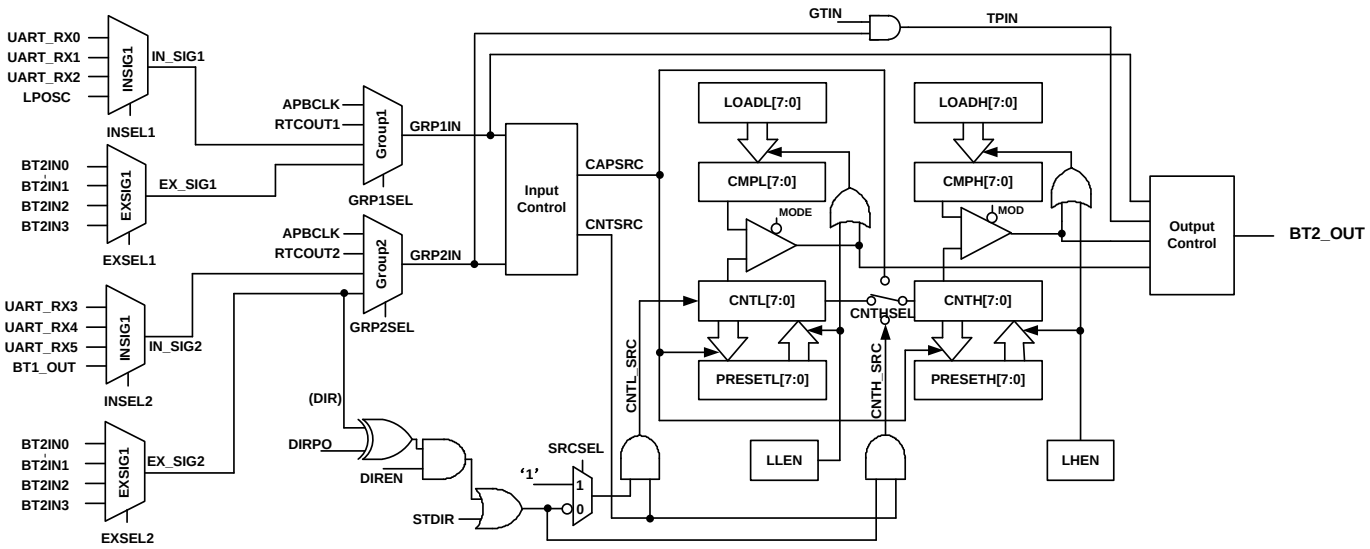


图 25-3 BT2 结构图

BT1和BT2的输入控制逻辑示意图如下，可以实现对计数源的预分频，以及计数、捕捉的边沿选择。

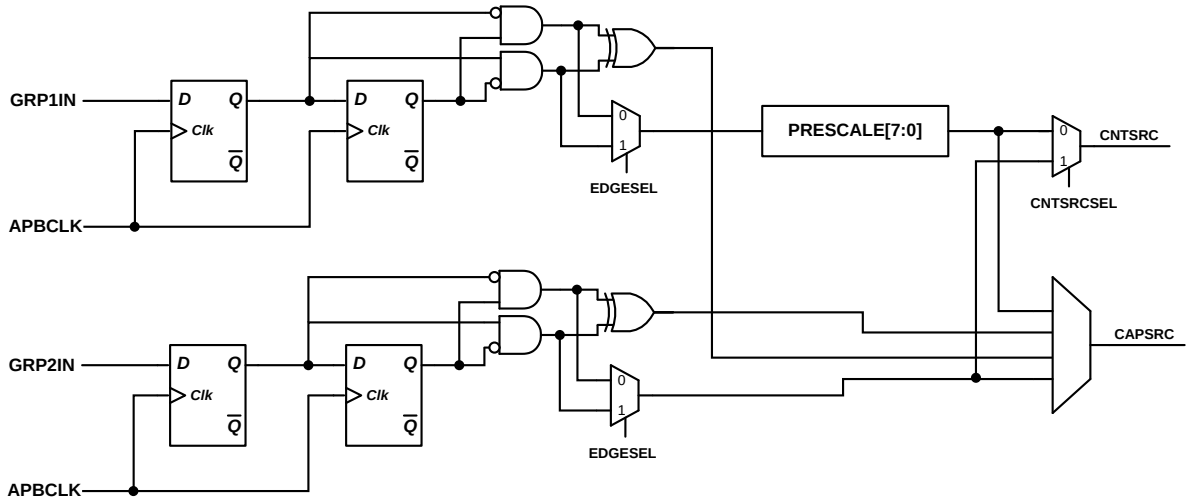


图 25-4 BT1/2 输入控制逻辑

BT1和BT2的输出控制逻辑示意图如下，比较值匹配后，可以输出电平或脉冲信号，脉冲信号宽度可以设置，输出极性可配置。

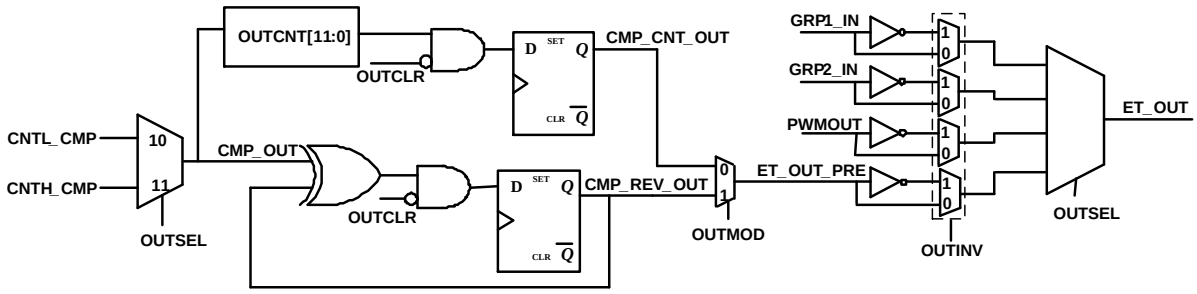


图 25-5 BT1/2 输出控制逻辑

25.2.3 工作模式

25.2.3.1 计数模式

- 16bit级联计数模式：

16bit计数值等于16bit比较寄存器时，比较中断状态寄存器只有CMPHIF置起。下图的参考波形中，CHTHSEL=00/11表示级联模式；LOADL和LOADH为比较值加载寄存器；LLEN和LHEN是加载使能寄存器，16位级联计数模式下软件对LHEN写1将使BasicTimer自动加载比较值和计数器预设值，随后自动启动计数；CMPH和CMPL为高低位比较寄存器；CLEN和CHEN是计数器使能信号；CNTL和CNTH是两组8bit计数器；COMPHIF和COMPHIF为高低计数器比较中断标志位；OVHIF和OVLIF为高低计数器溢出标志位。

当16bit级联计数器的计数值等于比较寄存器{CMPH,CMPL}时，CMPHIF中断标志置位，同时计数器自动归零，重新开始计数。

在计数器启动后，软件可以通过改写LOADL和LOADH寄存器动态调整下一个计数周期使用的比较值；当计数值匹配比较值，并重启计数器时，Basic Timer也会同时重新从LOAD寄存器加载比较值，这样下一个计数周期的长度也就相应的改变了。

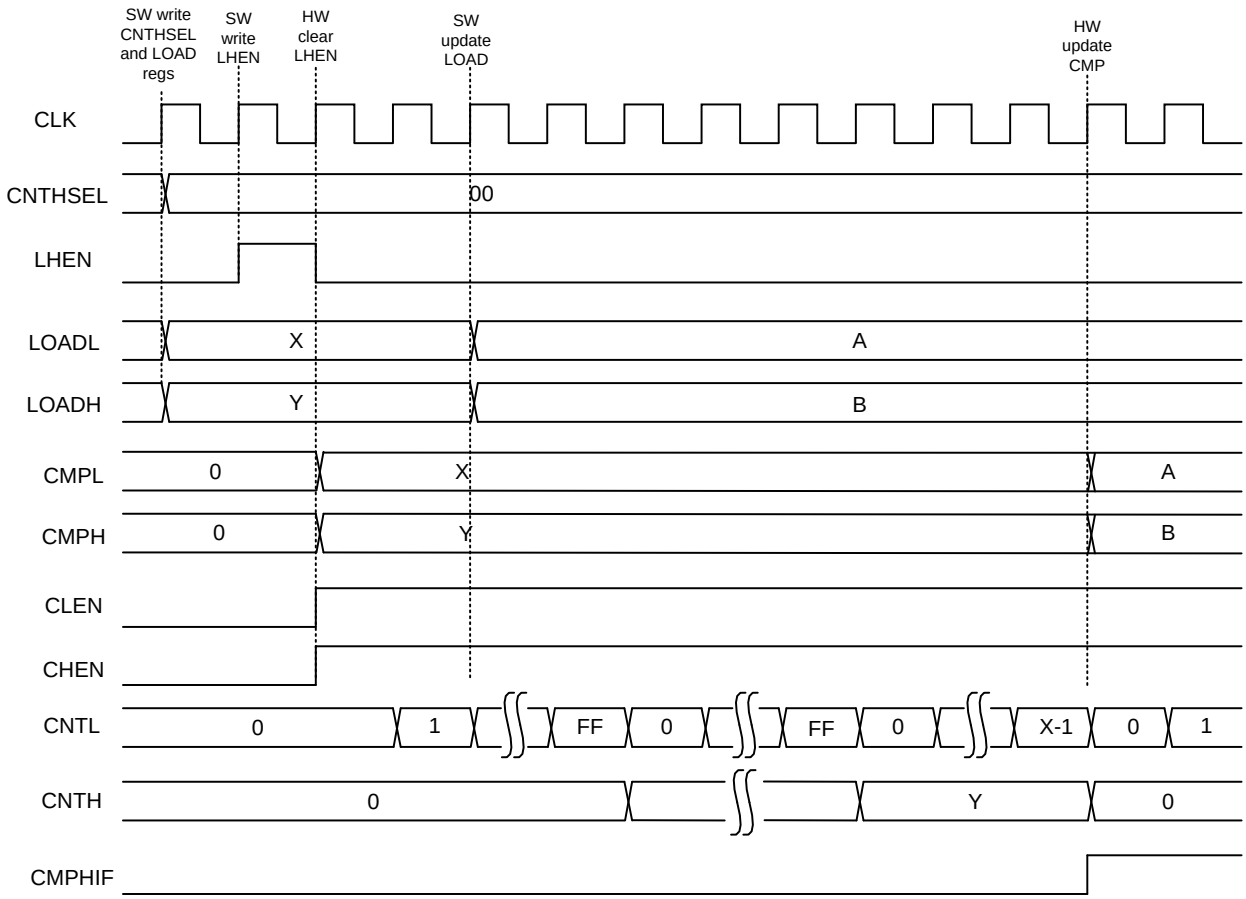


图 25-6 BT 16bit 级联计数

- 8bit独立计数模式

8bit独立计数模式下，每个8bit计数器独立工作，波形示意图如下。

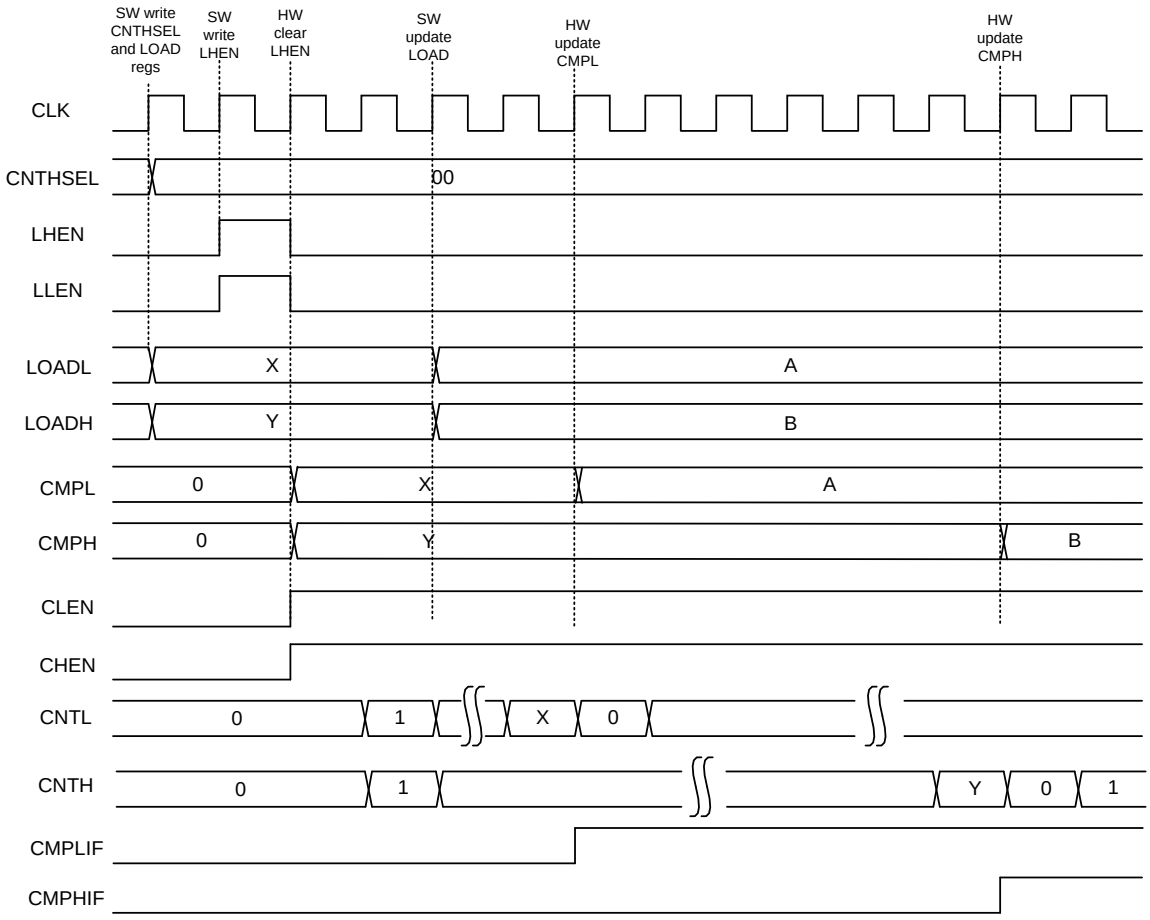


图 25-7 BT 8bit 独立计数

25.2.3.2 捕捉模式

捕捉模式只能工作在16bit级联模式下，从0开始计数。当选择上升沿捕捉周期模式时，电路在检测到捕捉源CAPSRC的有效沿时把计数器CNTL和CNTH的值捕捉到寄存器PRESETL和PRESETH中，并置位中断标志位CAPIF。

下图以捕捉CAPSRC信号上升沿为例。

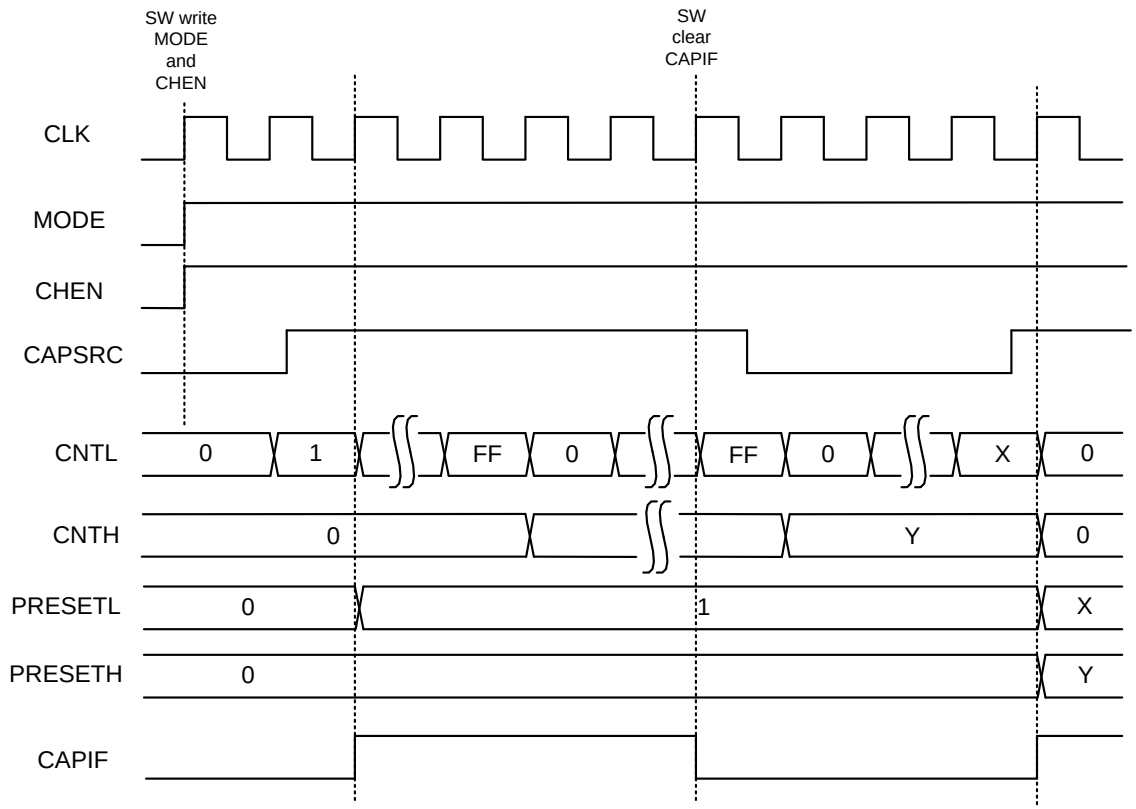


图 25-8 BT 捕捉模式

捕捉模式支持清零捕捉和单次捕捉

- 清零捕捉

不论在脉冲宽度还是周期捕捉情况下，捕捉到第一个沿后将计数器清零产生中断，捕捉到第二个沿后把计数值锁存到高低位预置数寄存器，同时清零计数器。

- 单次捕捉

单次捕捉是指捕捉到一个有效的脉冲宽度（宽度捕捉）或者一个有效的周期（周期捕捉）后停止计数。以带清零捕捉模式下周期捕捉为例见下图：CAPSRC为捕捉源信号；PRESETL和PRESETH为预置数寄存器；CLEN和CHEN为计数器使能信号；CNTL和CNTH为计数器；CAPIF为捕捉中断标志位。

25.2.3.3 8bit PWM 模式

PWM模式开启时，使用CNTL计数，加载控制寄存器中LLEN不能配置为1。PWM启动后CNTL首先从PRESETL值开始计数，并连续与LOADL寄存器比较，当CNTL < LOADL时，输出波形为低，当匹配发生时（CNTL=LOADL），输出波形为高电平，并保持直到CNTL计数值等于0xFF，随后计数

器自动重载为PRESETL值，并将输出波形恢复为低电平。

示意图如下：

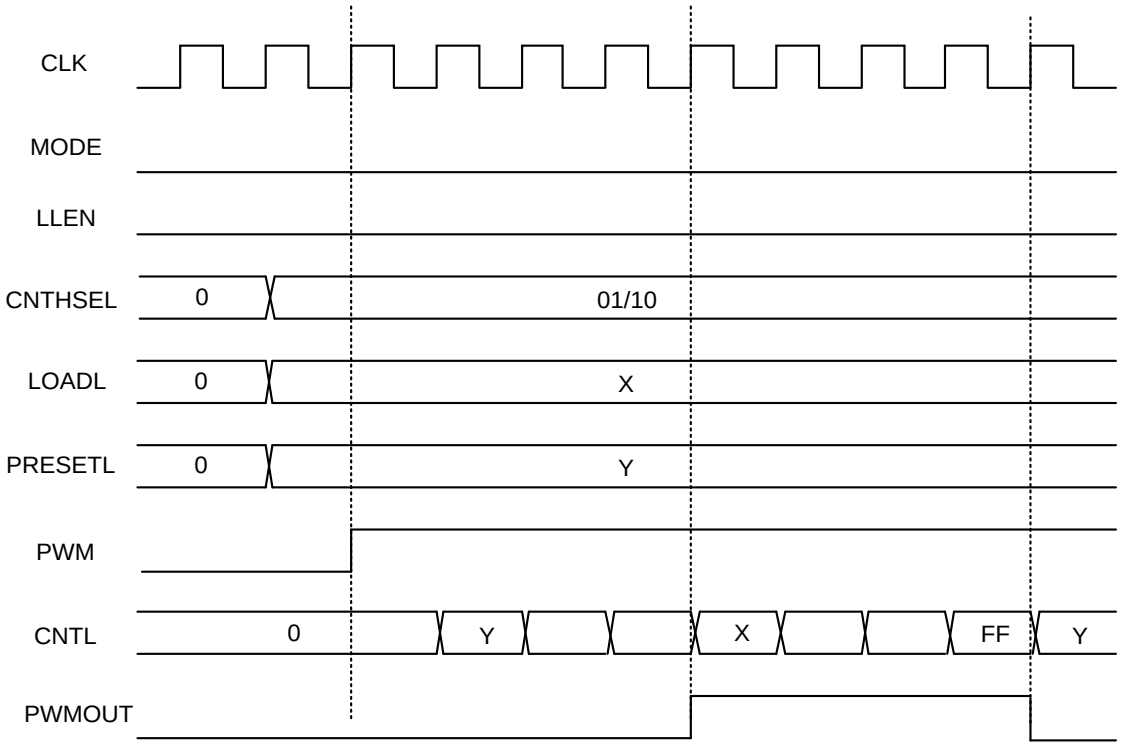


图 25-9 8bit PWM

25.2.3.4 脉冲输出模式

当设置了BT_OUT输出脉冲宽度寄存器后，模块可以输出相应的脉冲宽度。当计数器匹配到来后的第一个32768时钟上升沿开始触发脉冲输出，计数时钟为32768Hz。输出脉冲宽度=([OUTCNTH,OUTCNTL]+1)/32768秒。以高8BIT计数器匹配为例，设[OUTCNTH,OUTCNTL]=1，波形如下：LOADH为高位加载寄存器；LHEN是高位加载使能寄存器，用来把高位加载寄存器的值加载进高位比较寄存器里；CMPH为高位比较寄存器；CHEN高位是计数器使能信号；CNTH是高位计数器；XTLF是32768HZ时钟信号；CMP_CNT是以32768HZ为计数周期的脉冲宽度内部计数器，当它计数到等于[OUTCNTH,OUTCNTL]的值时回到0；BT_OUT为输出脉冲信号。

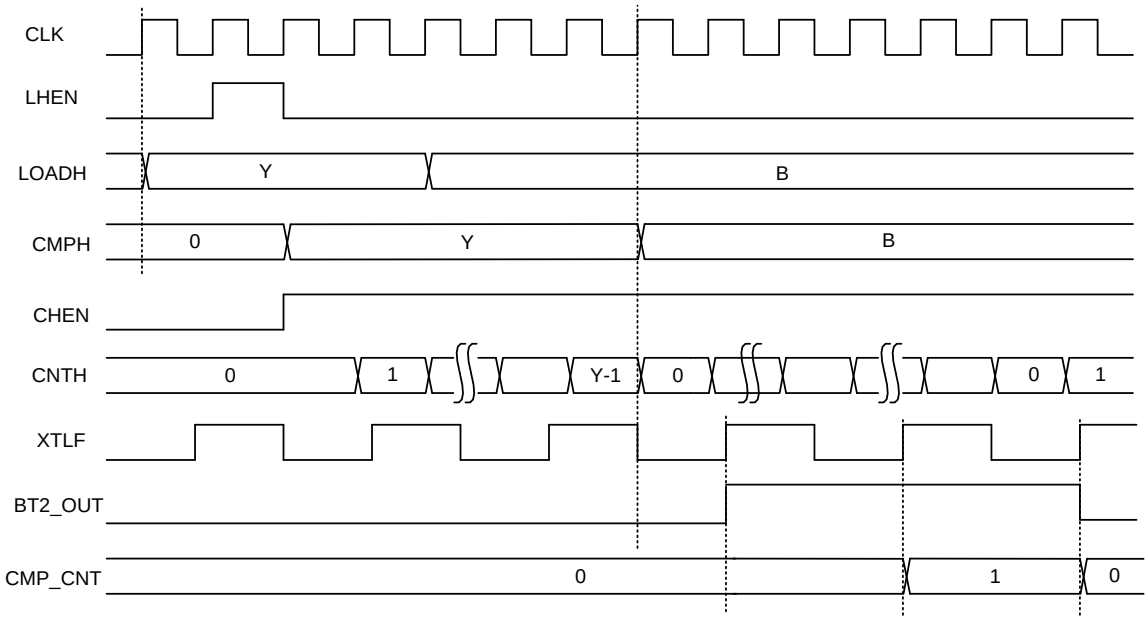


图 25-10 输出脉冲

25.2.3.5 Toggle 输出模式

当设置了输出BT_OUT控制寄存器的OUTMOD位为1后，当比较匹配后模块可以输出与之前相反的电平（Toggle）。以高8位的计数器为例，示意波形如下图。

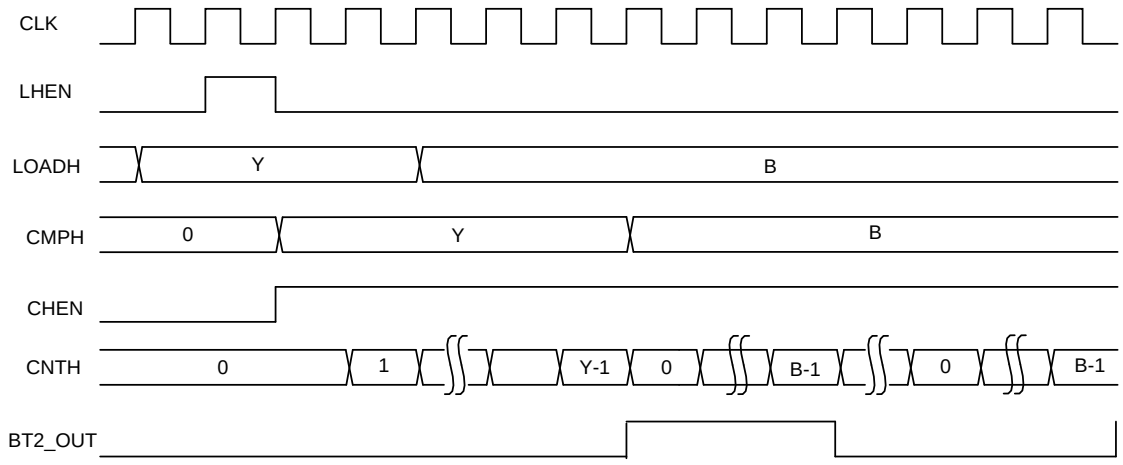


图 25-11 输出翻转

25.2.4 功能说明

25.2.4.1 计数源预分频功能

通过写计数源预分频寄存器，分频数=（X+1），即0x00表示1分频，0xFF表示256分频。预分频后的信号为单周期脉冲同步计数使能信号。假设X=2，计数源预分频示意图如下：PRESCALE为分频寄存器；CNTSRC为分频后的计数脉冲；CNTL为计数器；这里假设BT1对GRP1IN输入信号进行计数。

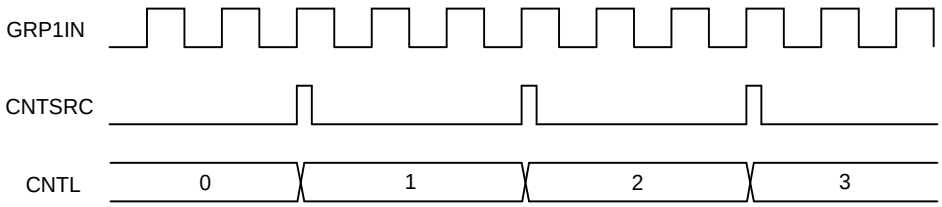


图 25-12 计数源预分频

25.2.4.2 计数器加载功能

预置数功能（PRESET）：加载控制寄存器（LHEN/LLEN）有效时将预置数寄存器（PRESET）的值加载到计数器中作为初值。

比较寄存器（CMP）：存放计数目标值，计数模式下有效；当计数器当前值与比较寄存器匹配、或加载控制有效时，硬件自动将LOAD寄存器的值加载到比较寄存器中。

25.2.5 寄存器

模块起始地址：0x4001_3000

offset 地址	名称	符号
BT1(模块起始地址:0x40013000)		
0x00	Basic Timer1 控制寄存器 1 (Basic Timer 1 Control Register 1)	BT1_CR1
0x04	Basic Timer1 控制寄存器 2 (Basic Timer 1 Control Register 2)	BT1_CR2
0x08	Basic Timer1 配置寄存器 1 (Basic Timer 1 Config Register 1)	BT1_CFGR1
0x0C	Basic Timer1 配置寄存器 2 (Basic Timer 1 Config Register 2)	BT1_CFGR2
0x10	Basic Timer1 预分频寄存器 (Basic Timer 1 Prescaler Register)	BT1_PRES
0x14	Basic Timer1 加载控制寄存器 (Basic Timer 1 Load Control Register)	BT1_LOADCR
0x18	Basic Timer1 低位计数器寄存器 (Basic Timer 1 Counter Low)	BT1_CNTL
0x1C	Basic Timer1 高位计数器寄存器 (Basic Timer 1 Counter High)	BT1_CNTH
0x20	Basic Timer1 预置数寄存器 (Basic Timer 1 Preset Register)	BT1_PRESET
0x24	Basic Timer1 低位加载寄存器 (Basic Timer 1 Load Register Low)	BT1_LOADL
0x28	Basic Timer1 高位加载寄存器 (Basic Timer 1 Load Register High)	BT1_LOADH
0x2C	Basic Timer1 低位比较寄存器 (Basic Timer 1 Compare Register Low)	BT1_CMPL
0x30	Basic Timer1 高位比较寄存器 (Basic Timer 1 Compare Register High)	BT1_CMPH
0x34	Basic Timer1 输出脉冲宽度寄存器 (Basic Timer 1 Output Counter)	BT1_OUTCNT
0x38	Basic Timer1 输出控制寄存器 (Basic Timer 1 Output Control Register)	BT1_OCR
0x3C	Basic Timer1 中断使能寄存器 (Basic Timer 1 Interrupt Enable Register)	BT1_IER
0x40	Basic Timer1 中断标志寄存器 (Basic Timer 1 Interrupt Status Register)	BT1_ISR
BT2(模块起始地址:0x40013044)		
0x00	Basic Timer2 控制寄存器 1 (Basic Timer2 Control Register 1)	BT2_CR1
0x04	Basic Timer2 控制寄存器 2 (Basic Timer2 Control Register 2)	BT2_CR2
0x08	Basic Timer2 配置寄存器 1 (Basic Timer2 Config Register 1)	BT2_CFGR1
0x0C	Basic Timer2 配置寄存器 2 (Basic Timer2 Config Register 2)	BT2_CFGR2
0x10	Basic Timer2 预分频寄存器 (Basic Timer2 Prescaler Register)	BT2_PRES
0x14	Basic Timer2 加载控制寄存器 (Basic Timer2 Load Control Register)	BT2_LOADCR

offset 地址	名称	符号
BT1(模块起始地址:0x40013000)		
0x18	Basic Timer2 低位计数器寄存器 (Basic Timer2 Counter Low)	BT2_CNTL
0x1C	Basic Timer2 高位计数器寄存器 (Basic Timer2 Counter High)	BT2_CNTH
0x20	Basic Timer2 预置数寄存器 (Basic Timer2 Preset Register)	BT2_PRESET
0x24	Basic Timer2 低位加载寄存器 (Basic Timer2 Load Register Low)	BT2_LOADL
0x28	Basic Timer2 高位加载寄存器 (Basic Timer2 Load Register High)	BT2_LOADH
0x2C	Basic Timer2 低位比较寄存器 (Basic Timer2 Compare Register Low)	BT2_CMPL
0x30	Basic Timer2 高位比较寄存器 (Basic Timer2 Compare Register High)	BT2_CMPH
0x34	Basic Timer2 输出脉冲宽度寄存器 (Basic Timer2 Output Counter)	BT2_OUTCNT
0x38	Basic Timer2 输出控制寄存器 (Basic Timer2 Output Control Register)	BT2_OCR
0x3C	Basic Timer2 中断使能寄存器 (Basic Timer2 Interrupt Enable Register)	BT2_IER
0x40	Basic Timer2 中断标志寄存器 (Basic Timer2 Interrupt Status Register)	BT2_ISR

25.2.5.1 BasicTimer x 控制寄存器 1 (BTx_CR1)

*两个 8bit 定时器 BT1H 和 BT1L 组合成 BT1；两个 8bit 定时器 BT2H 和 BT2L 组合成 BT2

名称	BTx_CR1(x=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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CHEN	CLEN	MODE	EDGESEL	CAPMOD	CAPCLR	CAPONCE	PWM
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
31:8	-	RFU，未实现，读为 0
7	CHEN	高 8 位计数器（BT1H 或 BT2H）启动控制 (Counter-Highend enable) 1：启动高 8bit 计数器，在计数器模式下启动时将预置数值和加载

位号	助记符	功能描述
		值分别加载至计数器和比较寄存器；捕捉模式下启动时计数器由零开始自由计数，计数到 0xFFFF 后产生溢出信号然后由零开始重新计数，捕捉功能只工作在 16 位模式；在 16 位的定时/计数和捕捉模式下 CHEN 作为 16 位计数器的启动控制，CLEN 自动失效 0：停止高 8bit 计数器
6	CLEN	低 8 位计数器（BT1L 或 BT2L）启动控制 (Counter-Lowend enable) 1：启动低 8bit 计数器，在计数器模式下启动时将预置数值和加载值分别加载至计数器和比较寄存器；捕捉模式下启动时计数器由零开始自由计数，计数到 0xFFFF 后产生溢出信号然后由零开始重新计数，捕捉功能只工作在 16 位模式；在 16 位的定时/计数和捕捉模式下 CHEN 作为计数器的启动控制，CLEN 自动失效 0：停止低 8bit 计数器计数
5	MODE	工作模式选择 (work mode) 1：16 位捕捉模式 0：8 位定时/计数模式，若高位计数器计数源选择为低位计数器的溢出信号，则可实现 16 位定时/计数模式
4	EDGESEL	计数模式下的计数沿和周期捕捉时的捕捉沿选择位 (edge select) 1：计数模式采样计数源下降沿，周期捕捉模式时下沿捕捉 0：计数模式采样计数源上升沿，周期捕捉模式时上沿捕捉 注：不支持系统时钟的下降沿计数，捕捉源和计数源为系统时钟时选择下降沿将不会有效计数。
3	CAPMOD	捕捉模式控制（只在捕捉模式下有效）(capture mode) 1：脉冲宽度捕捉 0：脉冲周期捕捉
2	CAPCLR	带清零捕捉模式控制 (capture clear) 1：不论在脉冲宽度还是周期捕捉情况下，捕捉到第一个沿后将计数器清零产生中断，捕捉到第二个沿后锁存（锁存到高低位预置数寄存器）计数值并同时清零计数器 0：捕捉不清零，计数器一直自由计数
1	CAPONCE	单次捕捉控制 (capture once) 1：单次捕捉有效，在捕捉到一次脉冲宽度或脉冲周期后计数器停止，若需要再次捕捉需重新启动 0：连续捕捉
0	PWM	PWM 模式输出 (pulse width modulation) 1：PWM 输出使能 0：PWM 输出不使能

25.2.5.2 BasicTimer x 控制寄存器 2 (BTx_CR2)

名称	BTx_CR2(x=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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SIG2SEL	SIG1SEL	CNTHSEL		DIREN	STDIR	SRCSEL	DIRPO
位权限	R/W-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
31:8	-	RFU，未实现，读为 0
7	SIG2SEL	计数器内部计数源信号选择 (signal group2 select) 1：内部计数源信号选择 Group2 0：内部计数源信号选择 Group1
6	SIG1SEL	计数器内部捕捉源信号选择 (signal group1 select) 1：内部捕捉源信号选择 Group1 0：内部捕捉源信号选择 Group2
5:4	CNTHSEL	高 8 位计数器计数源选择 (Counter Highend source select) 00/11：选择 ET1 的低位计数器的溢出信号，与低位计数器组成 16 位计数器 01：选择内部捕捉源信号 10：选择内部计数源信号或外部 DIR 输入组合信号
3	DIREN	外部输入 DIR 控制使能。通常电量脉冲输出时同时会输出一个由高低电平指示正反向的方向信号 DIR。电路将通过 DIR 信号电平的高低，分别控制高位计数器和低位计数器计数使能，以实现针对正向、反向脉冲的各种计数功能 (direction bit enable) 1：外部输入的 DIR 信号有效，此时高低位计数器是否计数可由外部输入的 DIR 信号控制。 0：外部输入的 DIR 信号无效，此时高低位计数器是否计数将由内部控制信号控制。
2	STDIR	内部 DIR 控制信号，当 DIREN 为 0，即外部输入 DIR 控制无效时，可由该信号代替 DIR 输入，直接控制内部计数器的计数。当需要外部 DIR 输入，即 DIREN 为 1 时，该位应设置为 0 (set direction) 1：内部 DIR 信号为高电平，则高 8 位计数器计数
1	SRCSEL	低位计数器计数使能控制选择信号 (source select) 1 低位计数器计数使能端选则常使能。此时低位计数器计数不受 DIR 控制，可将正反向所有脉冲一并计数 0：低位计数器计数使能端选则由寄存器 STDIR 或外部 EX_SIG2 输入控制。
0	DIRPO	输入信号 2 极性选择 (direction polarity) 1：对外部输入 DIR 信号 EX_SIG2 反向 0：对外部输入 DIR 信号 EX_SIG2 不反向

外部DIR信号（EX_SIG2）、内部控制寄存器与计数器计数的关系如下表：

DIR（EX_SIG2）	DIRPO	DIREN	STDIR	SRCSEL	CNTL	CNTH
X	X	0	0	1	√	x
X	X	0	1	1	√	√
0	0	1	0	1	√	x
1	0	1	0	1	√	√
0	1	1	0	1	√	√
1	1	1	0	1	√	x

DIR（EX_SIG2）	DIRPO	DIREN	STDIR	SRCSEL	CNTL	CNTH
X	X	0	0	0	√	x
X	X	0	1	0	x	√
0	0	1	0	0	√	x
1	0	1	0	0	x	√
0	1	1	0	0	x	√
1	1	1	0	0	√	x

25.2.5.3 BasicTimer x 配置寄存器 1 (BTx_CFGR1)

名称	BTx_CFGR1(x=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	RTCSEL2		RTCSEL1		GRP2SEL		GRP1SEL	
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	RTCSEL2	RTCOUT2 信号选择控制 2 (RTCOUT2 source select) 00：32768Hz，XTLF 时钟输出 01：RTCSec，由 RTC 模块输出的秒信号 10：RTCMin，由 RTC 模块输出的分钟信号 11：RFU
5:4	RTCSEL1	RTCOUT1 信号选择控制 1 (RTCOUT1 source select) 00：32768Hz，XTLF 时钟输出 01：RTCSec，由 RTC 模块输出的秒信号 10：RTCMin，由 RTC 模块输出的分钟信号 11：RFU
3:2	GRP2SEL	Group2 信号选择控制 (Group2 source select) 00：APBCLK 01：RTCOUT2 10：IN_SIG2，内部输入信号 2 11：EX_SIG2，外部输入信号 2 注：不支持 APBCLK 的下降沿捕捉和计数，捕捉源和计数源为 APBCLK 时选择下降沿将不会有效捕捉和计数。
1:0	GRP1SEL	Group1 信号选择控制（可作为捕捉模式下采样时钟选择，同时可作为信号捕捉源）(Group1 source select) 00：APBCLK 01：RTCOUT1

位号	助记符	功能描述
		10：IN_SIG1，内部输入信号 1 11：EX_SIG1，外部输入信号 1 注：不支持 APBCLK 的下降沿捕捉和计数，捕捉源和计数源为 APBCLK 时选择下降沿将不会有效捕捉和计数。

25.2.5.4 BasicTimer x 配置寄存器 2 (BTx_CFGR2)

名称	BTx_CFGR2(x=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	EXSEL2		EXSEL1		INSEL2		INSEL1	
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	EXSEL2	外部输入信号选择控制 2 (external source select2) 00：BT1_IN0/BT2_IN0 01：BT1_IN1/BT2_IN1 10：BT1_IN2/BT2_IN2 11：BT1_IN3/BT2_IN3
5:4	EXSEL1	外部输入信号选择控制 1 (external source select1) 00：BT1_IN0/BT2_IN0 01：BT1_IN1/BT2_IN1 10：BT1_IN2/BT2_IN2 11：BT1_IN3/BT2_IN3
3:2	INSEL2	内部输入信号选择控制 2 (internal source select 2) 00：UART_RX3/UART_RX3 01：UART_RX4/UART_RX4 10：UART_RX5/UART_RX5 11：RCLP/BT1_OUT
1:0	INSEL1	内部输入信号选择控制 1 (internal source select 1) 00：UART_RX0/UART_RX0 01：UART_RX1/UART_RX1 10：UART_RX2/UART_RX2 11：RCLP/RCLP

25.2.5.5 BasicTimer x 预分频寄存器 (BTx_PRES)

名称	BTx_PRES(x=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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PRESCALE							
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	PRESCALE	输入 Group1 的预分频寄存器 (Group1 input signal prescaler) 分频数=（X+1），即 00 表示 1 分频，FF 表示 256 分频。预分频后的信号都为单周期脉冲的形式，占空比 1:X

25.2.5.6 BasicTimer x 加载控制寄存器 (BTx_LOADCR)

名称	BTx_LOADCR(x=1,2)							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			LHEN	-			LLEN
位权限	U-0			R/W-0	U-0			R/W-0

位号	助记符	功能描述
31:5	-	RFU，未实现，读为 0
4	LHEN	高位加载控制 (Counter highend load enable) 写 1 将预置数寄存器 ET1PRESETH 和加载寄存器 ET1LOADH 分别加载到计数值寄存器 ET1CNTH 和比较寄存器 ET1CMPH，写 0 无效，该位硬件自动清 0。在 16 位的定时/计数下 LHEN 作为计数器的加载控制，LLEN 自动失效

位号	助记符	功能描述
3:1	-	RFU，未实现，读为 0
0	LLEN	低位加载控制 (Counter lowend load enable) 写 1 将预置数寄存器 PRESETL 和加载寄存器 LOADL 分别加载到计数值寄存器 ET1CNTL 和比较寄存器 ET1CMPL，写 0 无效，该位硬件自动清 0。在 16 位的定时/计数下 LHEN 作为计数器的加载控制，LLEN 自动失效

25.2.5.7 BasicTimer x 低位计数器寄存器 (BTx_CNTL)

名称	BTx_CNTL(x=1,2)							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNTL							
位权限	R-00000000							

位号	助记符	功能描述
31:8	**-**	**未实现：读为 0**
7:0	CNTL	计数器低位计数值寄存器 (counter lowend) LLEN 有效时加载预置数到该寄存器。

25.2.5.8 BasicTimer x 高位计数器寄存器 (BTx_CNTH)

名称	BTx_CNTH(x=1,2)							
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	CNTH							
位权限	R-00000000							

位号	助记符	功能描述
31:8	-	未实现：读为 0
7:0	CNTH	计数器高位计数值寄存器 (counter highend) LHEN 有效时加载预置数到该寄存器。

25.2.5.9 BasicTimer x 预置数寄存器 (BTx_PRESET)

名称	BTx_PRESET(x=1,2)							
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	PRESETH							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PRESETL							
位权限	R/W-00000000							

位号	助记符	功能描述
31:16	-	未实现：读为 0
15:8	PRESETH	计数器高位预置数寄存器 (preset highend) 用于保存高位计数器初值，或保存捕捉结果高 8bit
7:0	PRESETL	计数器低位预置数寄存器 (preset lowend) 用于保存低位计数器初值，或保存捕捉结果低 8bit

25.2.5.10 BasicTimer x 低位加载寄存器 (BTx_LOADL)

名称	BTx_LOADL(x=1,2)							
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	LOADL							
位权限	R/W-00000000							

位号	助记符	功能描述
31:8	-	**未实现：读为 0**
7:0	LOADL	计数器低位加载寄存器 (load lowend counter) 在计数匹配或执行加载命令时将加载寄存器的值加载至比较工作寄存器。

25.2.5.11 BasicTimer x 高位加载寄存器 (BTx_LOADH)

名称	BTx_LOADH(x=1,2)							
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	LOADH							
位权限	R/W-00000000							

位号	助记符	功能描述
31:8	-	**未实现：读为 0**
7:0	LOADH	计数器高位加载寄存器 (load highend counter) 在计数匹配或执行加载命令时将加载寄存器的值加载至比较工作寄存器。当工作在 8 位定时/计数器模式时，该高位加载寄存器不支持加载值为 0x00 的设置。

25.2.5.12 BasicTimer x 低位比较寄存器 (BTx_CMPL)

名称	BTx_CMPL(x=1,2)							
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	CMPL							

位权限	R/W-00000000

位号	助记符	功能描述
31:8	-	**未实现：读为 0**
7:0	CMPL	计数器低位比较寄存器 (compare lowend) 加载寄存器的值加载后将写入该寄存器，该寄存器与计数器比较，若计数值大于等于该寄存器的值，则产生计数匹配信号至输出控制模块，并产生相应中断。

25.2.5.13 BasicTimer x 高位比较寄存器 (BTx_CMPH)

名称	BTx_CMPH(x=1,2)							
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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CMPH							
位权限	R/W-00000000							

位号	助记符	功能描述
31:8	-	**未实现：读为 0**
7:0	CMPH	计数器高位比较寄存器 (compare highend) 加载寄存器的值加载后将写入该寄存器，该寄存器与计数器比较，若计数值大于等于该寄存器的值，则产生计数匹配信号至输出控制模块，并产生相应中断。

25.2.5.14 BasicTimer x 输出脉冲宽度寄存器 (BTx_OUTCNT)

名称	BTx_OUTCNT(x=1,2)							
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	-				OUTCNT[11:8]			
位权限	U-0				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	OUTCNT[7:0]
位权限	R/W-00000000

位号	助记符	功能描述
31:12	-	**未实现：读为 0**
11:0	OUTCNT	计数器输出脉冲宽度计数器 (output pulse width counter) 该寄存器用于调整输出脉冲宽度。计数时钟为 32768Hz，对应的输出脉冲宽度范围为 30.5uS~125mS 。输出脉冲宽度=(OUTCNT+1)/32768 秒

25.2.5.15 BasicTimer x 输出控制寄存器 (BTx_OCR)

名称	BTx_OCR(x=1,2)							
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	-		OUTCLR	OUTINV	OUTMOD	OUTSEL		
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-000		

位号	助记符	功能描述
31:6	-	未实现：读为 0
5	OUTCLR	输出清零，该位硬件自动清0，清0后输出由原信号驱动 (output clear) 1：输出清零，若设置了输出反向则清零后输出为高电平，反之为低电平（只适用于 OUTMOD 输出模式=1 输出之前反向电平的情况，OUTMOD 输出模式=0 输出脉冲模式无效） 0：写入无效
4	OUTINV	输出电平反向选择 (output inverted) 1：输出电平取反（CHEN/CLEN 有效后，ETOUT 信号输出本信号在 CHEN/CLEN 有效之前的状态） 0：输出电平不取反（CHEN/CLEN 有效后，ETOUT 信号输出本信号在 CHEN/CLEN 有效之前状态的反态）
3	OUTMOD	输出模式选择 (output mode) 1：输出之前的反向电平 0：输出脉冲，脉冲宽度可调

位号	助记符	功能描述
2:0	OUTSEL	输出信号选择 (output select) 000：输出高位计数器的比较信号，只在计数模式有效。 001：输出低位计数器的比较信号，只在计数模式有效。 010：直接输出 Group1 的输入信号，计数模式、捕捉模式有效。 011：直接输出Group2的输入信号，计数模式、捕捉模式有效。 其他：PWM输出 注：输出信号不能配置为APBCLK的同频信号，实际得到的结果为正向输出高电平，反向输出低电平

25.2.5.16 BasicTimer x 中断使能寄存器 (BTx_IER)

名称	BTx_IER(x=1,2)							
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	-			CMPHIE	CMPLIE	OVHIE	OVLIE	CAPIE
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	-	**未实现：读为 0**
4	CMPHIE	扩展定时器高位比较发生信号 (compare highend interrupt enable) 1：中断使能 0：中断禁止
3	CMPLIE	扩展定时器低位比较发生信号 (compare lowend interrupt enable) 1：中断使能 0：中断禁止
2	OVHIE	定时器高位溢出信号 (highend overflow interrupt enable) 1：中断使能 0：中断禁止
1	OVLIE	定时器低位溢出信号 (lowend overflow interrupt enable) 1：中断使能 0：中断禁止
0	CAPIE	定时器捕捉产生信号 (capture interrupt enable) 1：中断使能 0：中断禁止

25.2.5.17 BasicTimer x 中断标志寄存器 (BTx_ISR)

名称	BTx_ISR(x=1,2)							
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	-		EDGESTA	CMPHIF	CMPLIF	OVHIF	OVLIF	CAPIF
位权限	U-0		R-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	-	**未实现：读为 0**
5	EDGESTA	捕捉沿状态 (edge status) 1：脉冲宽度捕捉模式时表示捕捉到下沿 0：脉冲宽度捕捉模式时表示捕捉到上沿
4	CMPHIF	定时器高位比较发生信号 (compare highend interrupt flag) 1：当前计数器的值大于等于比较寄存器的值，该信号将重新加载新的加载寄存器的值到工作寄存器。 0：当前计数器的值小于比较寄存器的值
3	CMPLIF	定时器低位比较发生信号 (compare lowend interrupt flag) 1：当前计数器的值大于等于比较寄存器的值，该信号将重新加载新的加载寄存器的值到工作寄存器。 0：当前计数器的值小于比较寄存器的值
2	OVHIF	定时器高位溢出信号 (highend overflow interrupt flag) 1：产生计数溢出 0：未产生溢出
1	OVLIF	定时器低位溢出信号 (lowend overflow interrupt flag) 1：产生计数溢出 0：未产生溢出
0	CAPIF	定时器捕捉产生信号 (capture interrupt flag) 1：捕捉到指定的沿 0：未捕捉到指定的沿

25.3 Extended Timer 模块

25.3.1 概述

扩展定时器支持以下功能

- 4个独立的16bit定时器
- Auto-reload定时
- 外部事件计数
- 输入边沿捕捉（事件触发、自由计数）
- 脉冲宽度或周期捕捉（PWC模式）
- 脉冲宽度调制（PWM模式）
- 32bit级联模式

4个扩展定时器结构相同，仅是计数源和捕捉源信号不同而已；每个扩展定时器包含一个计数器和一个初值寄存器，初值寄存器同时作为捕捉寄存器使用。扩展定时器使能后自动将初值寄存器的值加载至计数器，然后计数器以这个初值开始计数，计数至溢出后产生溢出信号，同时将初值寄存器的值重新加载至计数器。扩展定时器允许在计数过程中改变计数初值，完成动态的定时时间更新；

支持2种边沿捕捉模式：

✓ 事件触发捕捉：使能后计数器保持0，捕捉到第一个有效沿之后定时器才开始计数；

✓ 自由计数捕捉：定时器使能后立即由0开始计数，在有效捕捉信号沿到来时锁存当前计数值，同时产生捕捉中断，在下一有效捕捉沿到来时再次锁存当前计数值并产生捕捉中断，在计数溢出后产生溢出中断。

PWC模式与事件触发捕捉类似，只不过在第一个有效沿到来后开始计数，第二个有效沿到来后停止计数，两个有效沿可以独立配置上升或下降，即可以实现正脉冲宽度捕捉、负脉冲宽度捕捉、正沿周期捕捉、负沿周期捕捉。

每个扩展定时器支持多个外部引脚输入，和一个外部引脚输出。

25.3.2 结构框图

ET1结构框图如下：

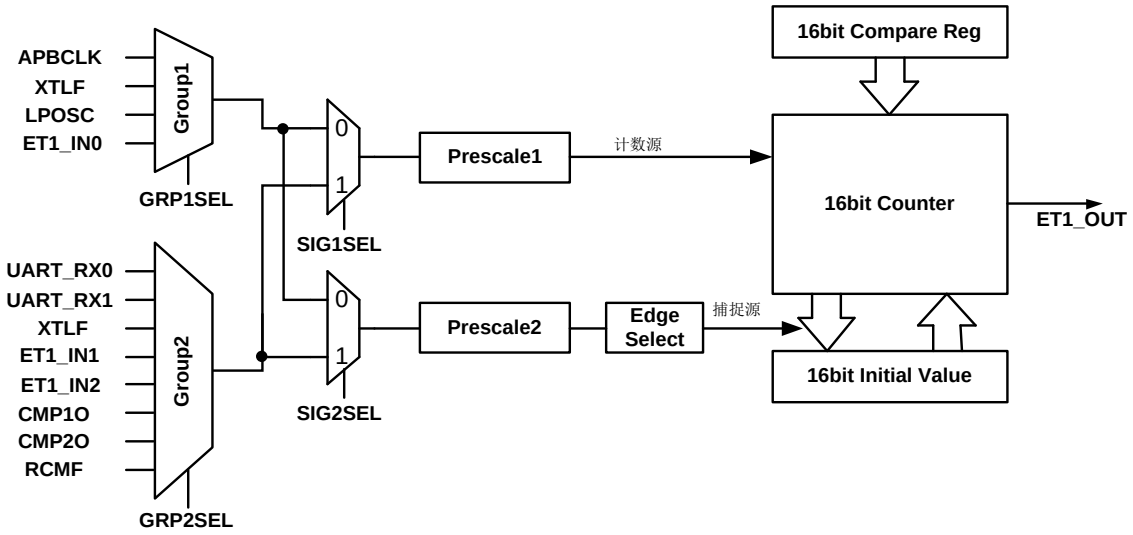


图 25-13 ET1 结构框图

ET2/3/4结构与ET1相同，只是输入信号源不一样。参见输入源选择寄存器定义。

级联模式下，32bit定时器的输入为ET1和ET3的输入，以ET1+ET2级联为例，结构框图如下：

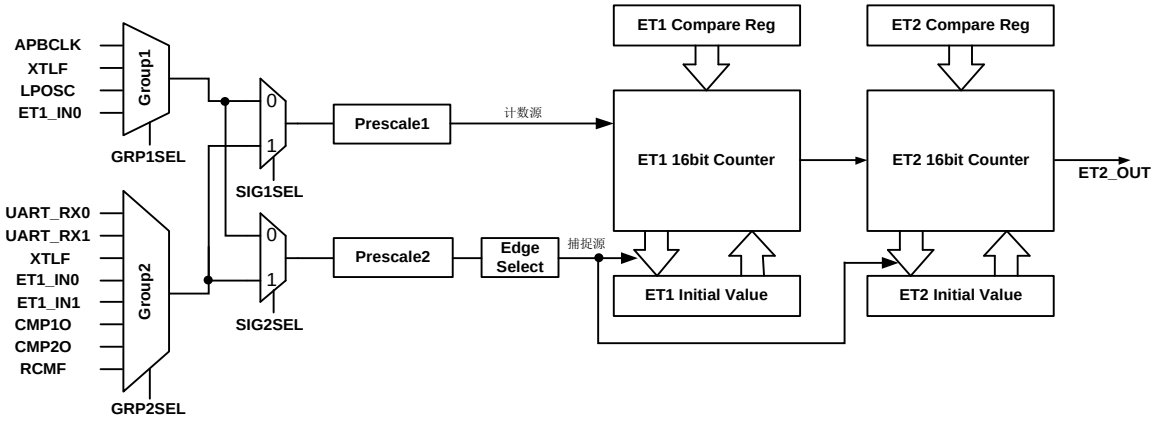


图 25-14 32bit 级联结构框图

级联后的ET1+ET2，仅产生ET2中断标志和ET2输出。

级联后的ET3+ET4，仅产生ET4中断标志和ET4输出。

25.3.3 输入信号

每个扩展定时器的输入分为两组，Group1为计数器计数源选择，Group2为捕捉源选择。根据寄存器设置，Group1和Group2可以互换。

输入信号包括APB总线时钟、XTLF、RCLP、外部GPIO输入、UART接收数据、比较器输出信号、LPTIM输出信号。

每个扩展定时器的计数源和捕捉源具体定义参见寄存器说明。

25.3.4 外部引脚输入数字滤波

Etx的外部引脚输入可以选择是否使能数字滤波。使能数字滤波的情况下，APBCLK连续3次采样相同才认为是输入变化有效；不使能数字滤波时，扩展定时器直接对管脚输入进行采样。

数字滤波示意图如下：

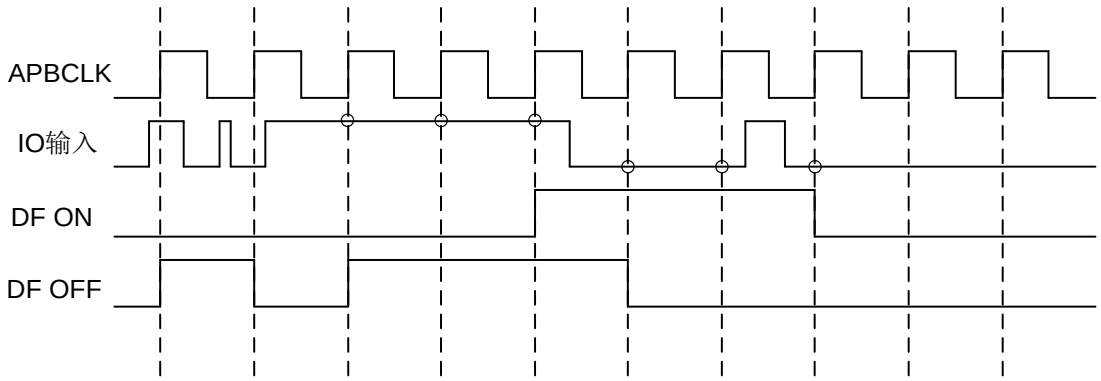


图 25-15 IO 输入数字滤波示意图

25.3.5 功能说明

- 计数器模式

支持16bit计数或级联的32bit计数。当CNTSEL=0，计数器为16bit；当CNTSEL=1，用ET1/ET3的16bit计数溢出信号作为ET2/ET4的计数指示信号。计数模式启动和计数溢出时将初值寄存器的值加载到计数器中。计数值达到0xFFFF时产生计数溢出中断。32bit级联模式下，ET2/ET4的计数值达到0xFFFF时产生级联溢出中断。

- 捕捉功能

ET使能后从0x0000开始自动计数，当捕捉源有效沿到来时，将当前计数值写入初值寄存器，并产生捕捉中断标志。ET保持计数和信号沿捕捉直到被关闭为止。

Etx用APBCLK采样其输入信号进行计数，所以应用中应避免被采样的输入信号频率高于APBCLK时

钟频率。

- PWM

扩展定时器支持16bit或32bit的PWM输出，启动PWM之后定时器从初值开始计数，当计数值等于比较寄存器时，PWM输出置1，当计数值溢出时PWM输出置0，16bit PWM如下图所示：

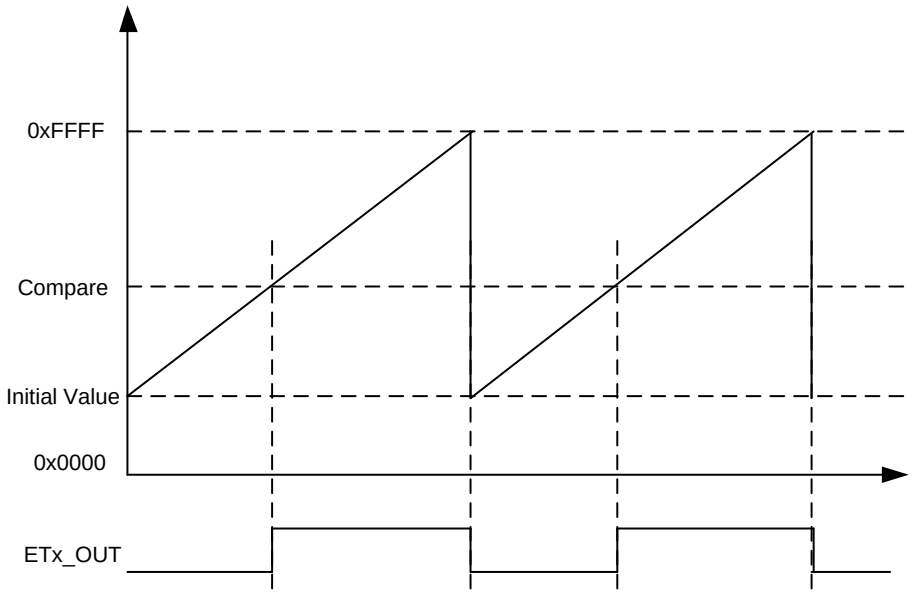


图 25-16 Etx PWM 输出示意图

25.3.6 寄存器

模块起始地址：0x4001_3000

offset 地址	名称	符号
ET1(模块起始地址:0x40013090)		
0x00	ET1 控制寄存器 (ET1 Control Register)	ET1_CR
0x04	ET1 输入源选择寄存器 (ET1 Input Select Register)	ET1_INSR
0x08	ET1 预分频寄存器 1 (ET1 Prescaler Register1)	ET1_PSCR1
0x0C	ET1 预分频寄存器 2 (ET1 Prescaler Register2)	ET1_PSCR2
0x10	ET1 初值寄存器 (ET1 Initial Value Register)	ET1_IVR
0x14	ET1 比较寄存器 (ET1 Compare Register)	ET1_CMPR
0x18	ET1 中断使能寄存器 (ET1 Interrupt Enable Register)	ET1_IER
0x1C	ET1 中断标志寄存器 (ET1 Interrupt Status Register)	ET1_ISR
ET2(模块起始地址:0x400130B0)		
0x00	ET2 控制寄存器 (ET2 Control Register)	ET2_CR
0x04	ET2 输入源选择寄存器 (ET2 Input Select Register)	ET2_INSR
0x08	ET2 预分频寄存器 1 (ET2 Prescaler Register1)	ET2_PSCR1
0x0C	ET2 预分频寄存器 2 (ET2 Prescaler Register2)	ET2_PSCR2
0x10	ET2 初值寄存器 (ET2 Initial Value Register)	ET2_IVR
0x14	ET2 比较寄存器 (ET2 Compare Register)	ET2_CMPR
0x18	ET2 中断使能寄存器 (ET2 Interrupt Enable Register)	ET2_IER
0x1C	ET2 中断标志寄存器 (ET2 Interrupt Status Register)	ET2_ISR
ET3(模块起始地址:0x400130D0)		
0x00	ET3 控制寄存器 (ET3 Control Register)	ET3_CR
0x04	ET3 输入源选择寄存器 (ET3 Input Select Register)	ET3_INSR
0x08	ET3 预分频寄存器 1 (ET3 Prescaler Register1)	ET3_PSCR1
0x0C	ET3 预分频寄存器 2 (ET3 Prescaler Register2)	ET3_PSCR2
0x10	ET3 初值寄存器 (ET3 Initial Value Register)	ET3_IVR
0x14	ET3 比较寄存器 (ET3 Compare Register)	ET3_CMPR
0x18	ET3 中断使能寄存器	ET3_IER

	(ET3 Interrupt Enable Register)	
0x1C	ET3 中断标志寄存器 (ET3 Interrupt Status Register)	ET3_ISR
ET4(模块起始地址:0x400130F0)		
0x00	ET4 控制寄存器 (ET4 Control Register)	ET4_CR
0x04	ET4 输入源选择寄存器 (ET4 Input Select Register)	ET4_INSR
0x08	ET4 预分频寄存器 1 (ET4 Prescaler Register1)	ET4_PSCR1
0x0C	ET4 预分频寄存器 2 (ET4 Prescaler Register2)	ET4_PSCR2
0x10	ET4 初值寄存器 (ET4 Initial Value Register)	ET4_IVR
0x14	ET4 比较寄存器 (ET4 Compare Register)	ET4_CMPR
0x18	ET4 中断使能寄存器 (ET4 Interrupt Enable Register)	ET4_IER
0x1C	ET4 中断标志寄存器 (ET4 Interrupt Status Register)	ET4_ISR
ETCNT(模块起始地址:0x40013110)		
0x00	ET1 计数值寄存器 (ET1 Counter Register)	ETCNT_CNT1
0x04	ET2 计数值寄存器 (ET1 Counter Register)	ETCNT_CNT2
0x08	ET3 计数值寄存器 (ET1 Counter Register)	ETCNT_CNT3
0x0C	ET4 计数值寄存器 (ET1 Counter Register)	ETCNT_CNT4

25.3.6.1 Etx 控制寄存器（ETx_CR）

名称	ETx_CR(x=1,2,3,4)							
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	-					OPOL	EXFLT	PWM
位权限	U-0					R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CEN	MOD	CASEN	EDGESEL	CAPMOD	CAPCLR	CAPONCE	CAPEDGE
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

位号	助记符	功能描述
31:11	-	未实现：读为0
10	OPOL	输出极性选择 (Output Polarity) 0：输出不取反 1：输出取反
9	EXFLT	外部引脚输入数字滤波使能 1 = 打开引脚输入信号数字滤波 0 = 关闭引脚输入信号数字滤波
8	PWM	PWM 输出控制 1 = PWM 输出使能 0 = PWM输出禁止
7	CEN	启动控制 1 = 启动定时器，在计数器模式下启动时将计数初值加载至计数器和工作寄存器；在捕捉模式下，启动时计数器由零开始自由计数，计数到0xFFFF后产生溢出信号然后由零开始重新计数 0 = 停止计数器计数
6	MOD	工作模式选择 1 = 捕捉模式 0 = 定时/计数模式
5	CASEN	Cascade Enable，扩展定时器级联使能 1 = ET1（ET3）和 ET2（ET4）级联成 32bit 定时器 0 = 16bit 定时器独立工作
4	EDGESEL	计数模式采沿方式选择 （计数时钟选择mcu_clk时该位无效，总是采用mcu_clk时钟上升沿计数） 1 = 计数模式采下降沿 0 = 计数模式采上升沿
3	CAPMOD	捕捉模式控制 1 = 脉宽捕捉 0 = 脉冲周期捕捉
2	CAPCLR	带清零捕捉模式控制 1 = 事件触发捕捉：使能后计数器保持0，捕捉到第一个有效沿之后timer才开始计数 0 = 捕捉不清零，计数器一直自由计数
1	CAPONCE	单次捕捉控制 1 = 单次捕捉有效，在捕捉到一次脉冲周期后计数器停止，若需要再次捕捉需重新启动 0 = 连续捕捉
0	CAPEDGE	捕捉沿选择 1 = 周期捕捉模式时下沿捕捉 0 = 周期捕捉模式时上沿捕捉

25.3.6.2 Etx 输入源选择寄存器（ETx_INSR）

名称	ETx_INSR(x=1,2,3,4)							
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	SIG2SEL	SIG1SEL	-	GRP2SEL			GRP1SEL	
位权限	R/W-0	R/W-0	U-0	R/W-000			R/W-00	

位号	助记符	功能描述
31:8	-	**未实现，读为0**
7	**SIG2SEL**	**内部信号2源选择(捕捉源)** 1：扩展定时器3的内部信号2选择Group1 0：扩展定时器 3 的内部信号 2 选择 Group2
6	**SIG1SEL**	**内部信号1源选择（在计数模式下计数源仅由此选择，捕捉模式下计数源）** 1：扩展定时器3的内部信号1选择Group2 0：扩展定时器 3 的内部信号 1 选择 Group1
5	-	**未实现，读为0**
4:2	**GRP2SEL**	**GROUP2 信号选择控制** **ET1** 000：UART0_RX 001：UART1_RX 010：XTLF 011：ET1_IN1 100：ET1_IN2 101：CMP1O（比较器1输出） 110：CMP2O（比较器2输出） 111：RCMF **ET2** 000：UART2_RX 001：UART3_RX 010：XTLF 011：ET2_IN1 100：ET2_IN2 101：CMP1O（比较器1输出） 110：CMP2O（比较器2输出） 111：RCMF **ET3** 000：ET3_IN1 001：XTLF 010：UART4_RX 011：UART5_RX 100：RTCSEC 101/110/111：RFU **ET4** 000：ET4_IN1 001：XTLF

位号	助记符	功能描述
		010：UART_RX2 011：UART_RX0 100：CMP1O（比较器1输出） 101：CMP2O（比较器2输出） 110：RTCSEC 111：RCMF
1:0	**GRP1SEL**	**GROUP1 信号选择控制** **ET1** 00：APBCLK 01：XTLF 10：RCLP 11：ET1_IN0 **ET2** 00：APBCLK 01：XTLF 10：RCLP 11：ET2_IN0 **ET3** 00：APBCLK 01：ET3_IN0 10：RTCSEC 11：RCLP **ET4** 00：APBCLK 01：ET4_IN0 10：RTC64HZ 11：RCMF

25.3.6.3 Etx 预分频寄存器 1（ETx_PSCR1）

名称	ETx_PSCR1(x=1,2,3,4)							
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	PRESCALE1							
位权限	R/W-00000000							

位号	助记符	功能描述
31:8	**-**	**未实现，读为0**

位号	助记符	功能描述
7:0	**PRESCALE1**	**输入 Signal1（计数源）的预分频寄存器** 00 表示 1 分频，FF 表示 256 分频

25.3.6.4 Etx 预分频寄存器 2（ETx_PSCR2）

名称	ETx_PSCR2(x=1,2,3,4)							
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	PRESCALE2							
位权限	R/W-00000000							

位号	助记符	功能描述
31:8	**-**	**未实现，读为0**
7:0	**PRESCALE2**	**输入 Signal2（捕捉源）的预分频寄存器** 00 表示 1 分频，FF 表示 256 分频。

25.3.6.5 Etx 初值寄存器（ETx_IVR）

名称	ETx_IVR(x=1,2,3,4)							
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	INITVALUE[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	INITVALUE[7:0]							
位权限	R/W-00000000							

位号	助记符	功能描述
31:16	**-**	**未实现，读为0**
15:0	**INITVALUE**	**扩展定时器初值寄存器**

25.3.6.6 Etx 比较寄存器（ETx_CMPR）

名称	ETx_CMPR(x=1,2,3,4)							
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	CMP[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CMP[7:0]							
位权限	R/W-00000000							

位号	助记符	功能描述
31:16	-	未实现，读为0
15:0	CMP	扩展定时器比较寄存器 该寄存器与计数器比较，若计数值大于等于该寄存器的值，则产生计数匹配信号至输出控制模块，并产生相应中断。

25.3.6.7 Etx 中断使能寄存器（ETx_IER）

名称	ETx_IER(x=1,2,3,4)							
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	-					CMPIE	CAPIE	OVIE
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	CMPIE	扩展定时器比较中断使能 1：使能 0：禁止

位号	助记符	功能描述
1	**CAPIE**	**扩展定时器捕捉中断使能** 1：使能 0：禁止
0	**OVIE**	**扩展定时器3溢出中断使能** 1：使能 0：禁止

25.3.6.8 Etx 中断标志寄存器（ETx_ISR）

名称	ETx_ISR(x=1,2,3,4)							
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	-				CMPIF	EDGESTA	CAPIF	OVIF
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	**-**	**未实现：读为 0**
3	**CMPIF**	**比较状态，写 1 清 0** 1：当前计数器的值大于等于比较寄存器的值 0：当前计数器的值小于比较寄存器的值
2	**EDGESTA**	**捕捉沿状态标志** 1：脉冲宽度捕捉模式时表示捕捉到下沿 0：脉冲宽度捕捉模式时表示捕捉到上沿
1	**CAPIF**	**扩展定时器捕捉产生信号** 1：捕捉到指定的沿 0：未捕捉到指定的沿
0	**OVIF**	**扩展定时器溢出信号，当计数器的值由0xFFFF再增加时将置位** 1：产生计数溢出 0：未产生溢出

25.3.6.9 ETx 计数值寄存器(ETCNT _CNTx)

名称	ETCNT_CNTx(x=1,2,3,4)							
地址	0x4001310C + x*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	ETxCNT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ETxCNT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:16	-	**未实现：读为0**
15:0	CNT	计数器数值

26 基本定时器（BSTIM32）

26.1 概述

FM33A0xxEV包含1个基本定时器。

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。

- 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

26.2 结构框图

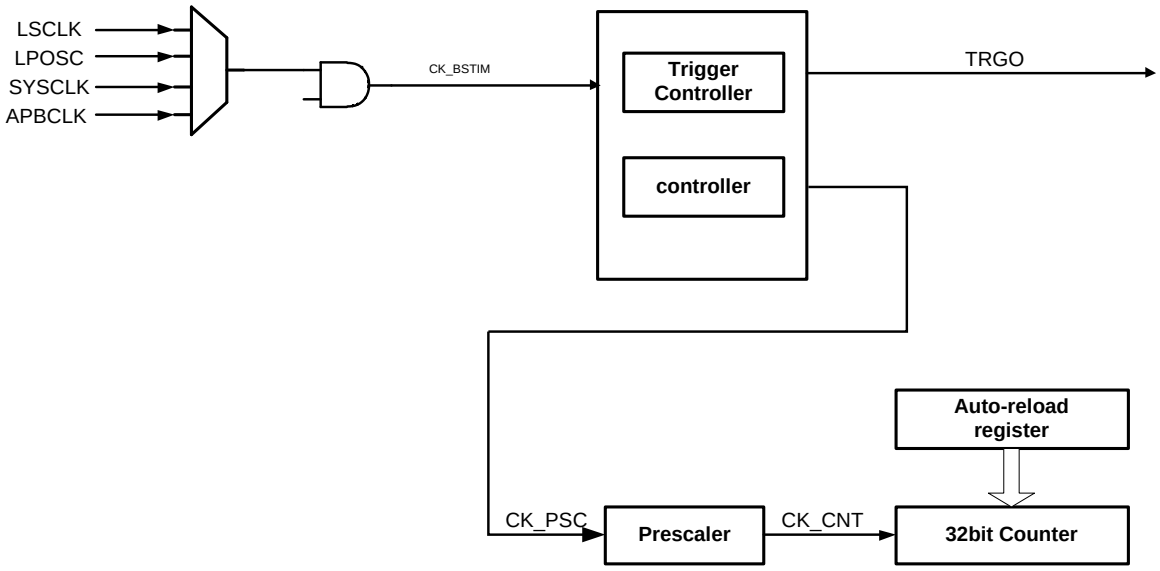


图 26-1 BSTIM32 结构框图

26.3 时钟和复位

BSTIM32的控制和状态寄存器都位于APB1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9时钟管理单元（CMU）。

BSTIM32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9.12.14外设独立工作时钟控制寄存器2可以选择BSTIM32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Prescaler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Prescaler。

BSTIM32模块可以通过操作BST32RST寄存器来复位和撤销复位，详情参见6.6.8APB外设复位寄存器。

26.4 功能描述

26.4.1 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CK_BSTIM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CK_BSTIM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。

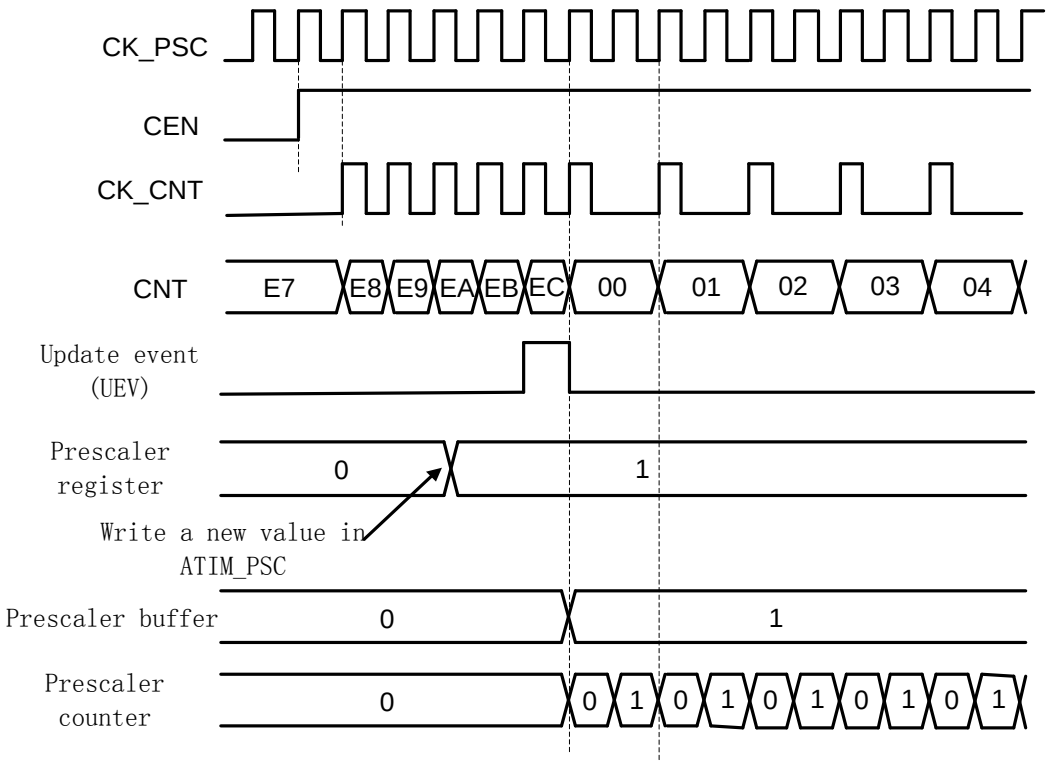


图 26-2 预分频从 1 变为 2 的波形

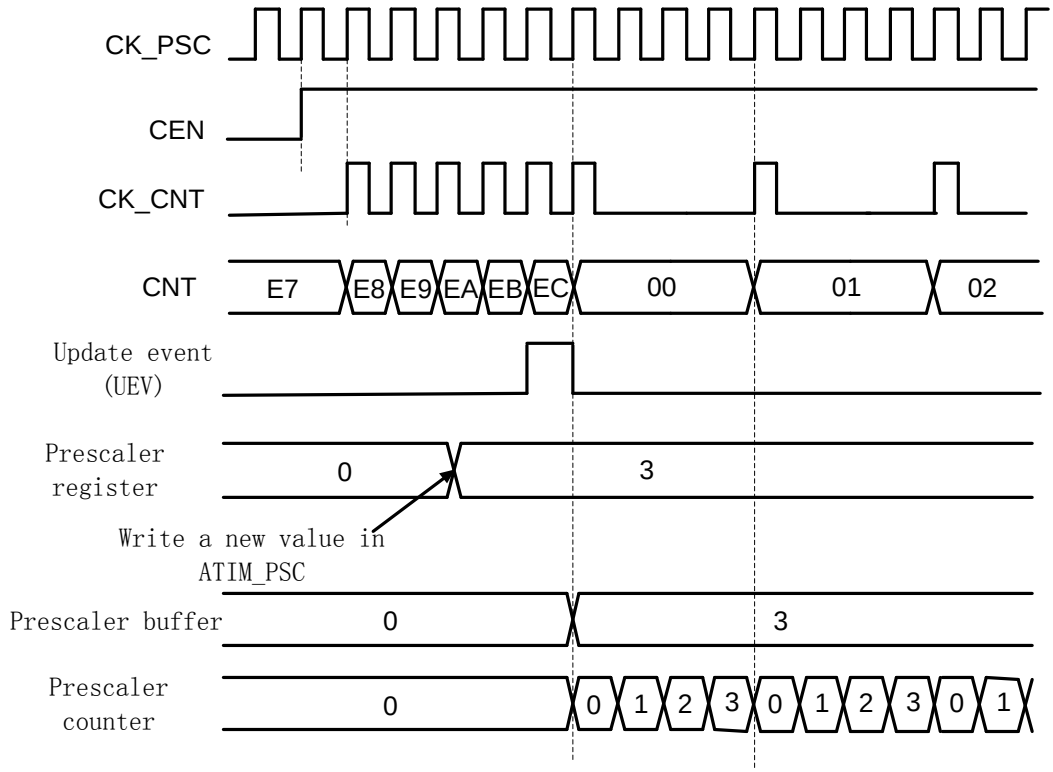


图 26-3 预分频从 1 变为 4 的波形

26.4.2 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_RCR更新为缓存中的值
- BSTIM_ARR更新为缓存中的值
- BSTIM_PSC更新为缓存中的值

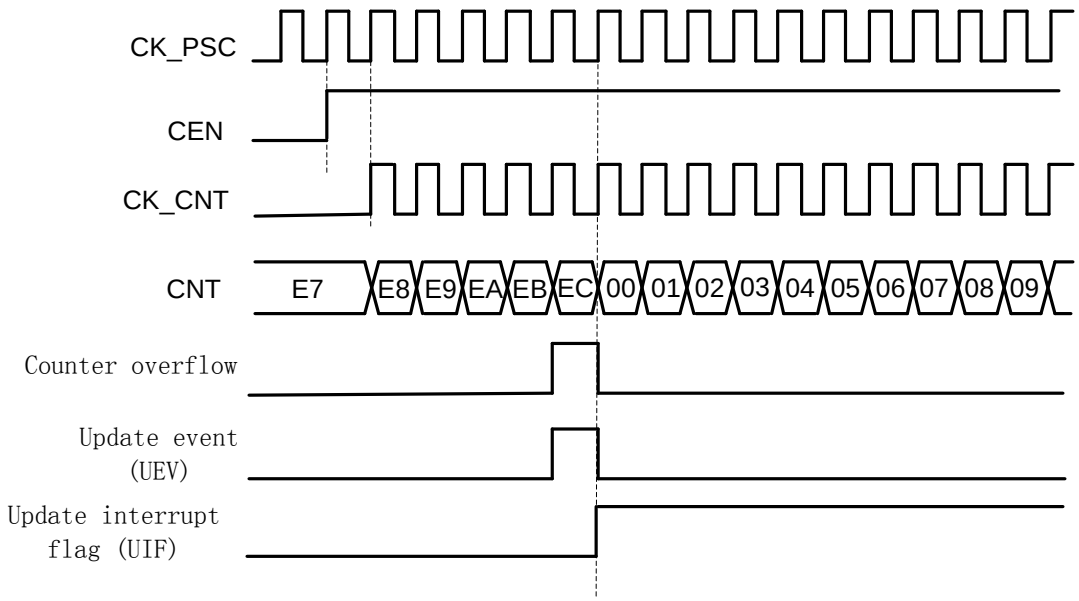


图 26-4 向上计数波形，内部时钟不分频

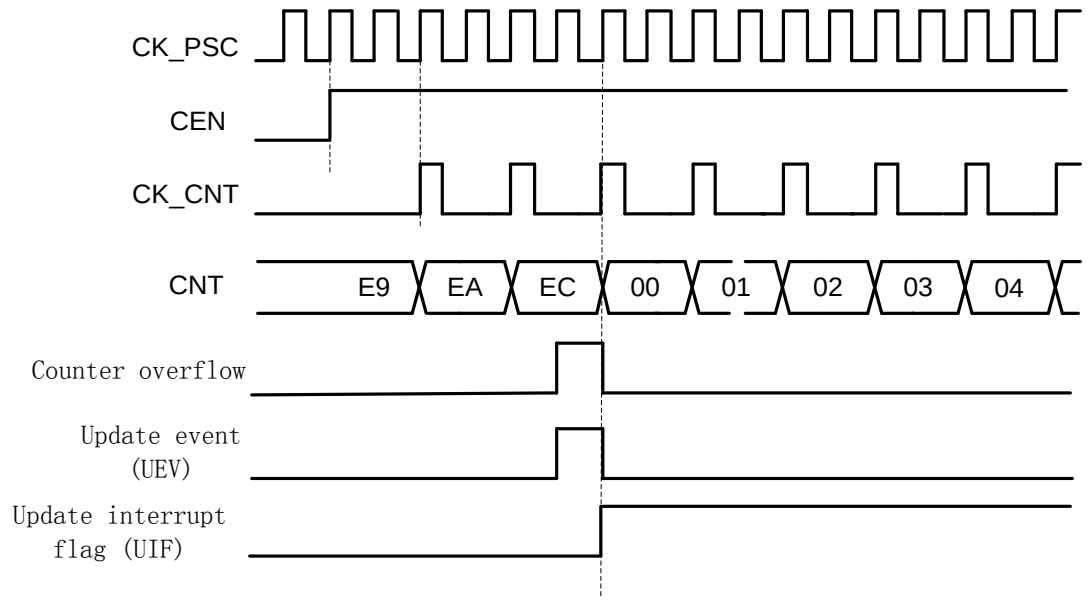


图 26-5 向上计数波形，内部时钟 2 分频

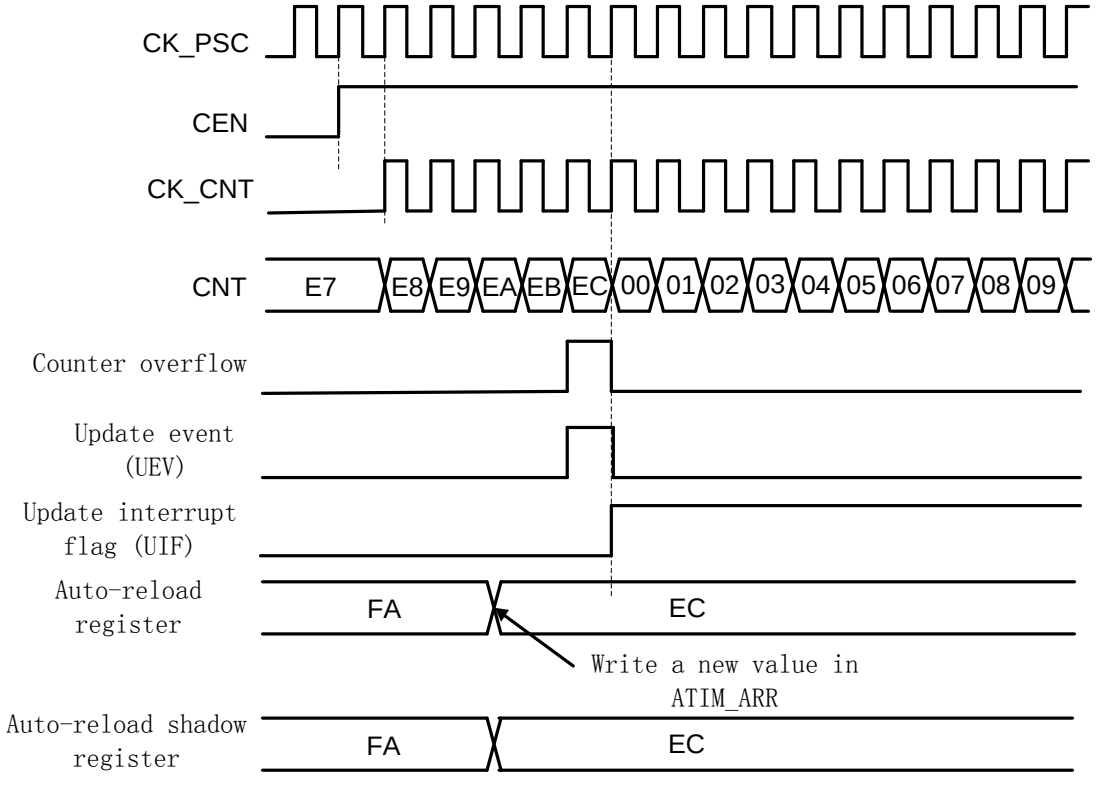


图 26-6 ARPE=0（ARR 没有预装载）时的更新事件

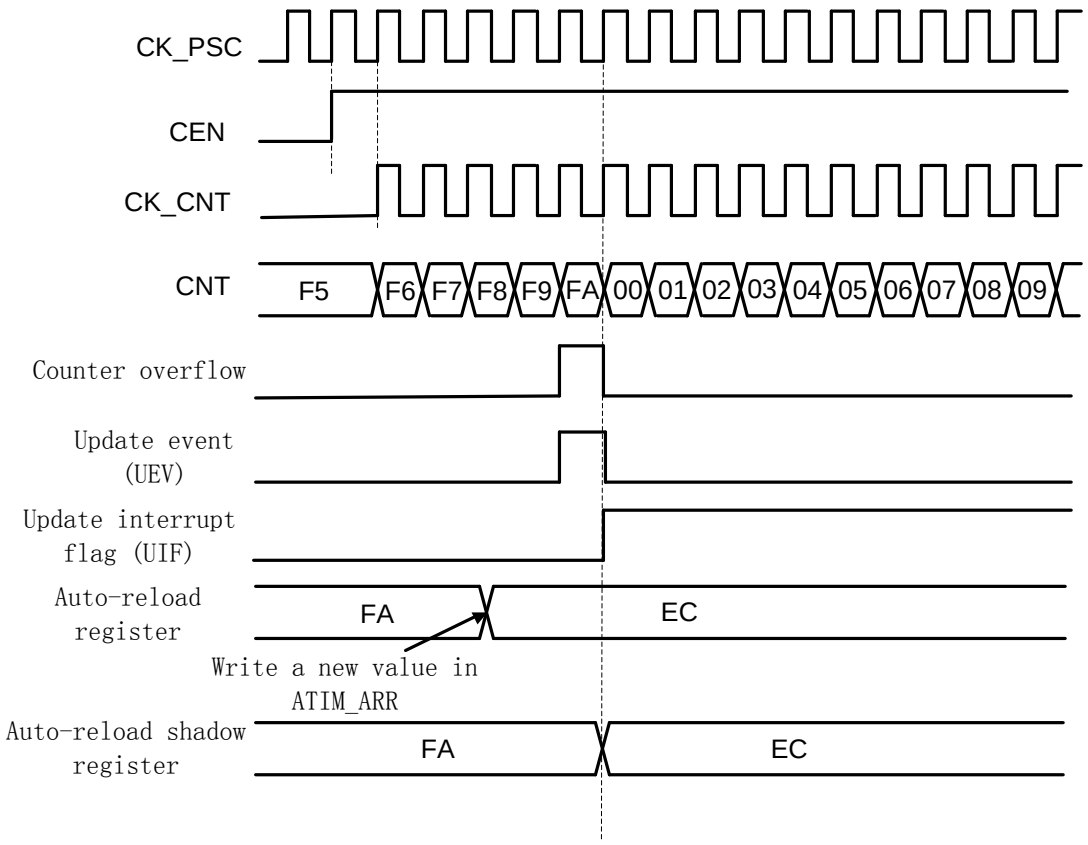


图 26-7 ARPE=1（ARR 预装载）时的更新事件

26.4.3 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。

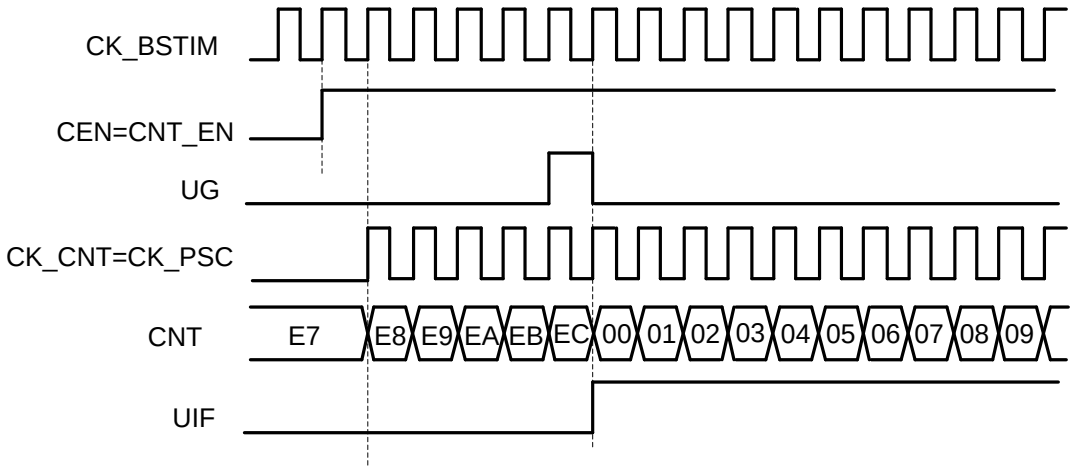


图 26-8 内部时钟源模式，时钟分频因子为 1

26.4.4 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。

26.5 寄存器

模块起始地址：0x4001_6000

offset 地址	名称	符号
BSTIM(模块起始地址:0x40016000)		
0x00	BSTIM 控制寄存器 1 （BSTIM Control Register1）	BSTIM_CR1
0x04	BSTIM 控制寄存器 2 （BSTIM Control Register2）	BSTIM_CR2
0x0C	BSTIM 中断使能寄存器 （BSTIM Interrupt Enable Register）	BSTIM_IER
0x10	BSTIM 中断标志寄存器 （BSTIM Interrupt Status Register）	BSTIM_ISR
0x14	BSTIM 事件产生寄存器 （BSTIM Event Generation Register）	BSTIM_EGR
0x24	BSTIM 计数器寄存器 （BSTIM Counter Register）	BSTIM_CNTR
0x28	BSTIM 预分频寄存器 （BSTIM Prescaler Register）	BSTIM_PSCR
0x2C	BSTIM 自动重载寄存器 （BSTIM Auto-Reload Register）	BSTIM_ARR

26.5.1 BSTIM 控制寄存器 1（BSTIM_CR1）

名称	BSTIM_CR1							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	-			OPM	URS	UDIS	CEN
位权限	R/W-0	U-0			R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:8	-	RFU，未实现，读为 0
7	ARPE	Auto-reload 预装载使能 (Auto-Reload Preload Enable) 0：ARR 寄存器不使能 preload 1：ARR 寄存器使能 preload
6:4	-	RFU，未实现，读为 0
3	OPM	单脉冲输出模式 (One Pulse Mode) 0：Update Event 发生时计数器不停止 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 - 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 - 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

26.5.2 BSTIM 控制寄存器 2（BSTIM_CR2）

名称	BSTIM_CR2							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	MMS			-			
位权限	U-0	R/W-000			U-0			

位号	助记符	功能描述
31:7	-	RFU，未实现，读为 0
6:4	MMS	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源 (Master Mode Select) 000：BSTIM_EGR 的 UG 寄存器被用作 TRGO 001：计数器使能信号 CNT_EN 被用作 TRGO，可用于同时启动多个定时器 010：UE（update event）信号被用作 TRGO 011/100/111：RFU 注意：从机定时器或 ADC 必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
3:0	-	RFU，未实现，读为 0

26.5.3 BSTIM 中断使能寄存器（BSTIM_IER）

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

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	UIE	Update 事件中断使能 (Update event Interrupt Enable) 0：禁止 Update 事件中断 1：允许 Update 事件中断

26.5.4 BSTIM 中断标志寄存器（BSTIM_ISR）

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

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	UIF	Update 事件中断标志，硬件置位，软件写 1 清零。(Update event Interrupt Flag,write 1 to flag) 当以下事件发生时，UIF 置位，并更新 shadow 寄存器 -重复计数器=0，并且 UDIS=0 的情况下，计数器发生溢出 -URS=0 且 UDIS=0 的情况下，软件置位 UG 寄存器初始化计数

位号	助记符	功能描述
		器 -URS=0 且 UDIS=0 的情况下，触发事件初始化计数器

26.5.5 BSTIM 事件产生寄存器（BSTIM_EGR）

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

位号	助记符	功能描述
31:1	**-**	RFU：未实现，读为 0
0	**UG**	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零 (User Generate) 软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。

26.5.6 BSTIM 计数器寄存器（BSTIM_CNTR）

名称	BSTIM_CNTR							
offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CNT[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CNT[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**CNT**	计数器值 (Counter)

26.5.7 BSTIM 预分频寄存器（BSTIM_PSCR）

名称	BSTIM_PSCR							
offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PSC[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PSC[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PSC[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**PSC**	计数器时钟（CK_CNT）预分频值 (Counter Clock Prescaler) $f_{CK_CNT}=f_{CK_PSC}/(PSC[15:0]+1)$ 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

26.5.8 BSTIM 自动重载寄存器（BSTIM_ARR）

名称	BSTIM_ARR							
offset	0x2C							
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

27 低功耗定时器（LPTIM32）

27.1 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、SYSCLK
  - 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

27.2 结构框图

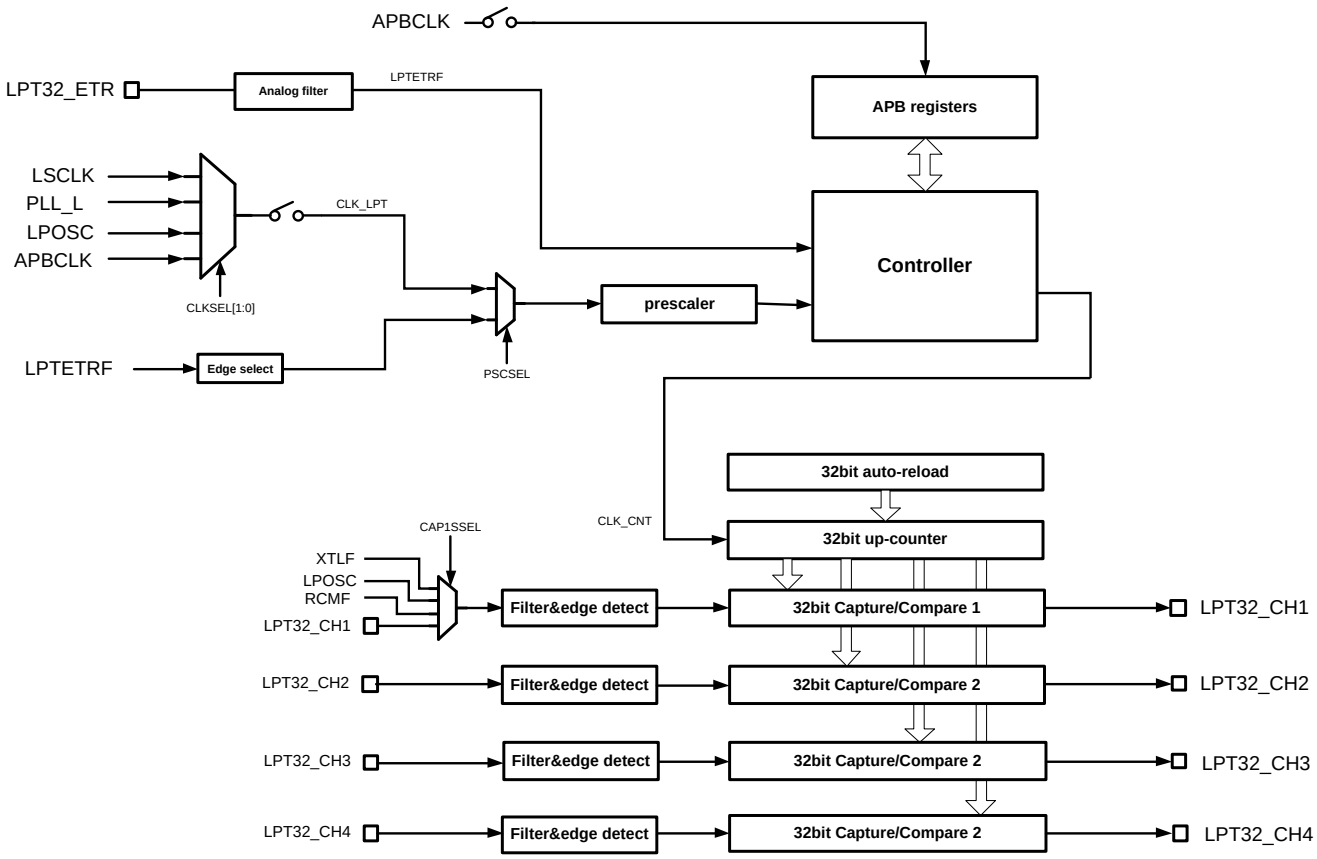


图 27-1 LPTIM32 结构框图

27.3 时钟和复位

LPTIM32的控制和状态寄存器都位于APB1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9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9.12.14外设独立工作时钟控制寄存器2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6.6.8APB外设复位寄存器。

27.4 相关引脚

功能	引脚映射	
	LQFP100	LQFP80
LPT32_ETR	PA13	PA13
LPT32_CH1	PA11	PA11
LPT32_CH2	PA12	PA12
LPT32_CH3	PE8	-
LPT32_CH4	PF0	PF0

表 27-1 LPTIM32 引脚映射

27.5 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。

27.5.1 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。

单次计数模式：计数器被触发后计数到目标值（LPTARR）后回到0，并自动停止，产生溢出中断，同时硬件自动清除LPTEN。

27.5.2 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。

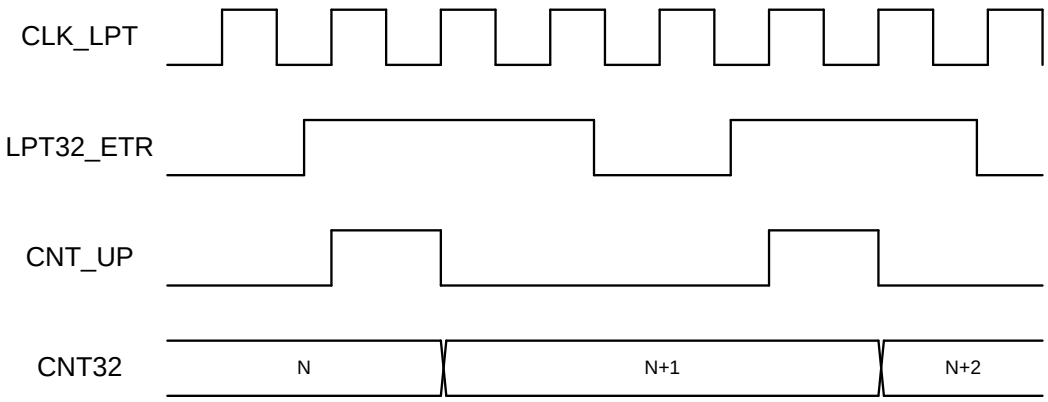


图 27-2 外部 ETR 脉冲上升沿触发计数

27.5.3 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。

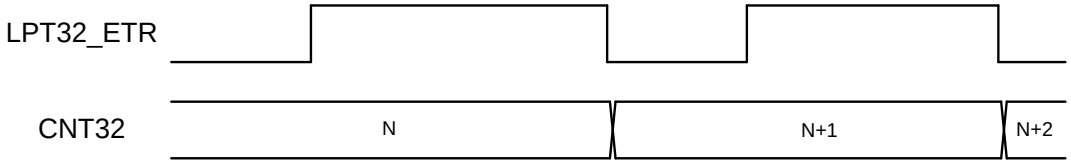


图 27-3 外部 ETR 脉冲异步计数（下降沿）

27.5.4 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。

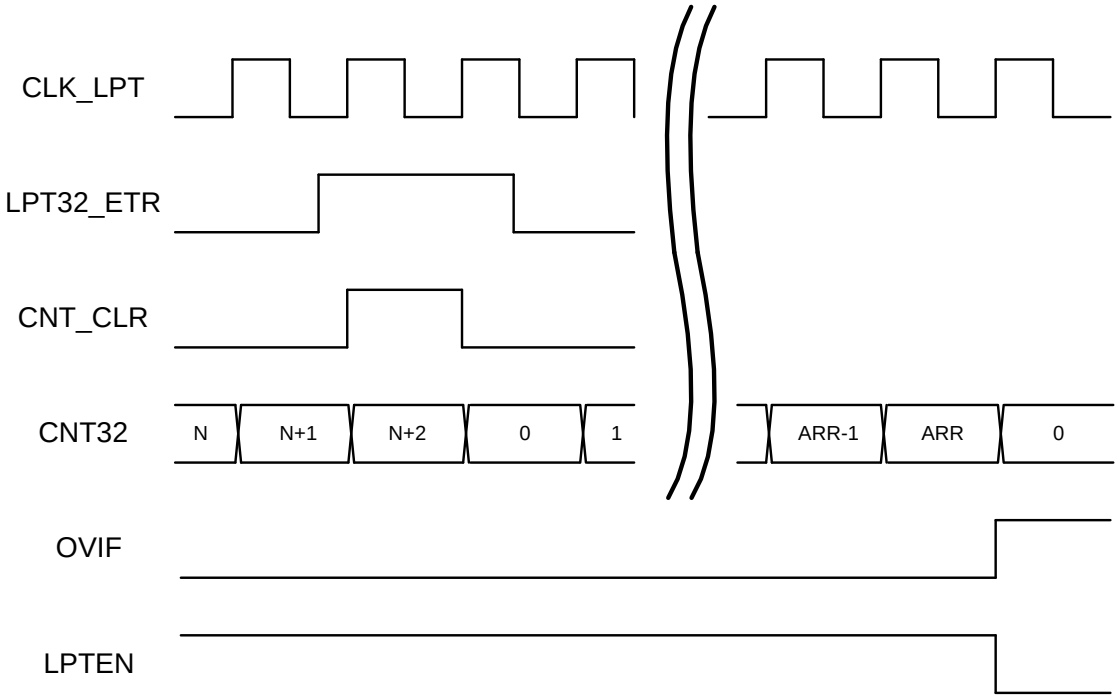


图 27-4TimeOut 模式

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。

27.6 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。

27.6.1 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。

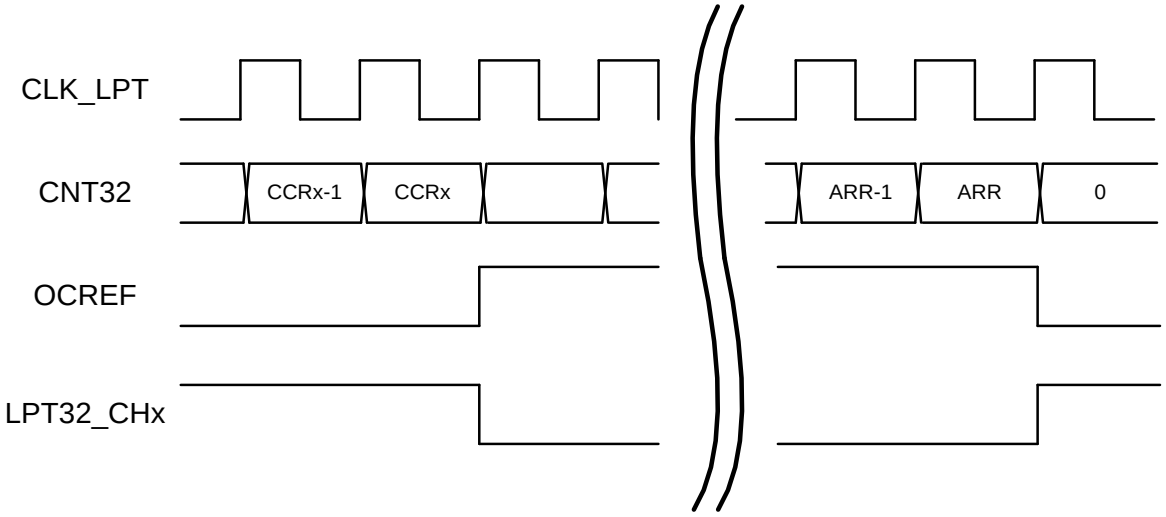


图 27-5PWM 输出

27.6.2 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。

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。

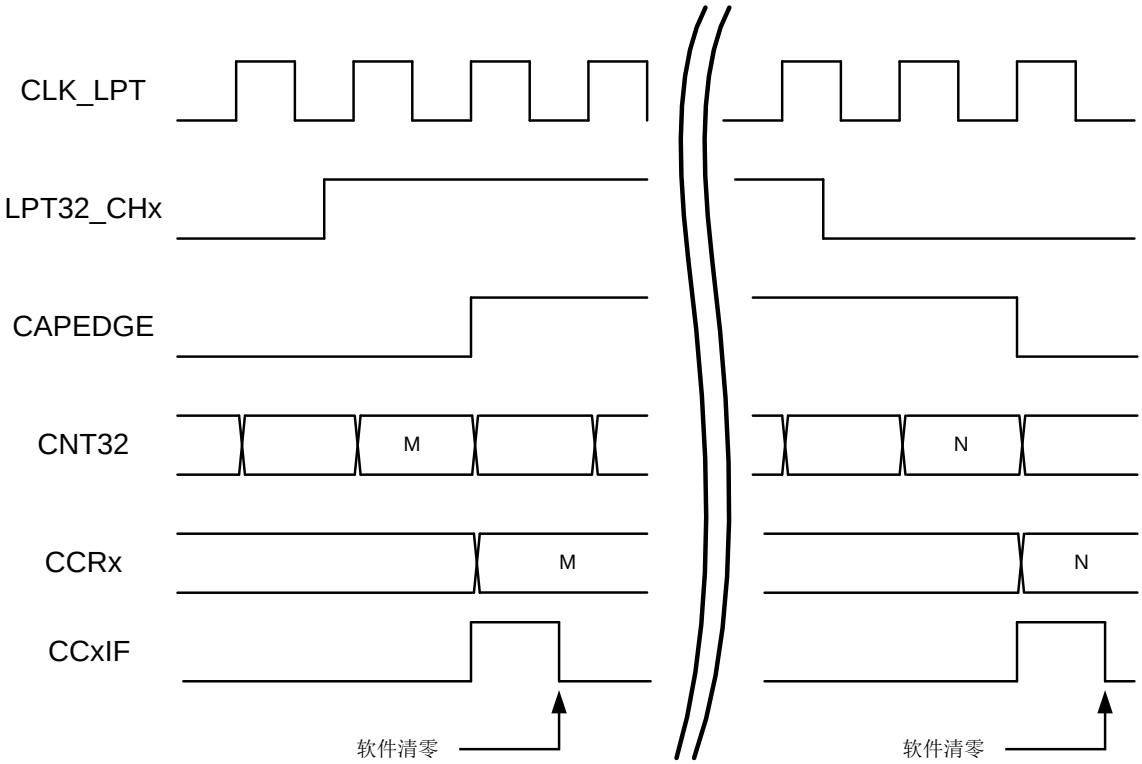


图 27-6 输入信号边沿捕捉

27.7 寄存器

模块起始地址：0x4001_3400

offset 地址	名称	符号
LPTIM(模块起始地址:0x40013400)		
0x00	LPTIM 配置寄存器 (LPTIM Config Register)	LPTIM_CFGR
0x04	LPTIM 计数寄存器 （LPTIM Counter Register）	LPTIM_CNTR
0x08	LPTIM 捕捉比较控制和状态寄存器 （ LPTIM Capture/Compare Control and Status Register）	LPTIM_CCSR
0x0C	LPTIM 目标值寄存器 （LPTIM Auto-Reload Register）	LPTIM_ARR
0x10	LPTIM 中断使能寄存器 （LPTIM Interrupt Enable Register）	LPTIM_IER
0x14	LPTIM 中断标志寄存器 （LPTIM Interrupt Status Register）	LPTIM_ISR
0x18	LPTIM 控制寄存器 （LPTIM Control Register）	LPTIM_CR
0x20	LPTIM 捕捉比较寄存器 1 （LPTIM Capture/Compare Register1）	LPTIM_CCR1
0x24	LPTIM 捕捉比较寄存器 2 （LPTIM Capture/Compare Register2）	LPTIM_CCR2
0x28	LPTIM 捕捉比较寄存器 3 （LPTIM Capture/Compare Register3）	LPTIM_CCR3
0x2C	LPTIM 捕捉比较寄存器 4 （LPTIM Capture/Compare Register4）	LPTIM_CCR4

27.7.1 LPTIM 配置寄存器（LPTIM_CFGR）

名称	LPTIM_CFGR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							ETR_AFEN
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
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	TMODE	
位权限	R/W-0	R/W-00		U-0		R/W-0	R/W-00	

位号	助记符	功能描述

位号	助记符	功能描述
31:25	-	**未实现：读为0**
24	**ETR_AFEN**	LPT32_ETR 输入模拟滤波使能(External Trigger input Analog Filter Enable) 0：关闭模拟滤波 1：使能模拟滤波，滤波宽度约 100ns
23:15	-	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。
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

27.7.2 LPTIM 计数值寄存器（LPTIM_CNTR）

名称	LPTIM_CNTR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CNT32[31:24]							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CNT32[23:16]							

位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT32[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT32[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:0	**CNT32**	32bit 计数器当前计数值(Counter 32bits-wide)

27.7.3 LPTIM 捕捉比较控制和状态寄存器（LPTIM_CCSR）

名称	**LPTIM_CCSR**							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						CAP1SSEL	
位权限	U-0						R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CAP4EDGE	CAP3EDGE	CAP2EDGE	CAP1EDGE	POLAR4	POLAR3	POLAR2	POLAR1
位权限	R/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	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	助记符	功能描述
31:26	-	未实现：读为0
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：LPT_CH1 输入 01：XTLF 10：RCLP 11：RCMF
23	**CAP4EDGE**	通道 4 当前被捕捉的边沿，在 CC2IF 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 CC1IF 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

位号	助记符	功能描述
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
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）

位号	助记符	功能描述
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）

27.7.4 LPTIM 目标值寄存器（LPTIM_ARR）

名称	LPTIM_ARR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ARR[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ARR[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
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

27.7.5 LPTIM 中断使能寄存器（LPTIM_IER）

名称	LPTIM_IER							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				OVR4IE	OVR3IE	OVR2IE	OVR1IE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGIE	OVIE	-		CC4IE	CC3IE	CC2IE	CC1IE
位权限	R/W-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
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 2 中断使能 0：捕捉/比较通道 2 中断禁止
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 1 中断使能 0：捕捉/比较通道 1 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

27.7.6 LPTIM 中断标志寄存器（LPTIM_ISR）

名称	LPTIM_ISR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				CAP4OVR	CAP3OVR	CAP2OVR	CAP1OVR
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGIF	OVIF	-		CC4IF	CC3IF	CC2IF	CC1IF
位权限	R/W-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
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
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

位号	助记符	功能描述
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

27.7.7 LPTIM 控制寄存器（LPTIM_CR）

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

位号	助记符	功能描述
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

27.7.8 LPTIM 捕捉比较寄存器 1（LPTIM_CCR1）

名称	LPTIM_CCR1							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CCR1[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CCR1[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述

位号	助记符	功能描述
31:0	**CCR1**	捕捉/比较值寄存器 1 (Channel1 Capture/Compare Register)

27.7.9 LPTIM 捕捉比较寄存器 2（LPTIM_CCR2）

名称	LPTIM_CCR2							
offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CCR2[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CCR2[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**CCR2**	捕捉/比较值寄存器 2 (Channel2 Capture/Compare Register)

27.7.10 LPTIM 捕捉比较寄存器 3（LPTIM_CCR3）

名称	LPTIM_CCR3							
offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CCR3[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CCR3[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR3[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR3[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**CCR3**	捕捉/比较值寄存器 3 (Channel3 Capture/Compare Register)

27.7.11 LPTIM 捕捉比较寄存器 4（LPTIM_CCR4）

名称	LPTIM_CCR4
offset	0x2C

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CCR4[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CCR4[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR4[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR4[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**CCR4**	捕捉/比较值寄存器 4 (Channel4 Capture/Compare Register)

28 实时时钟（RTC）

28.1 概述

实时时钟(RTC)模块可长时间维持精确计时，功耗极低，在所有功耗模式下都可以工作。

主要特性如下：

- BCD 时间格式，完整万年历
- 支持数字调校，最小步长 0.06ppm，精度可达+/-0.03ppm
- 可输出周期唤醒中断
- 闹钟功能
- RTC 计时部分寄存器不受系统复位影响

28.2 结构框图

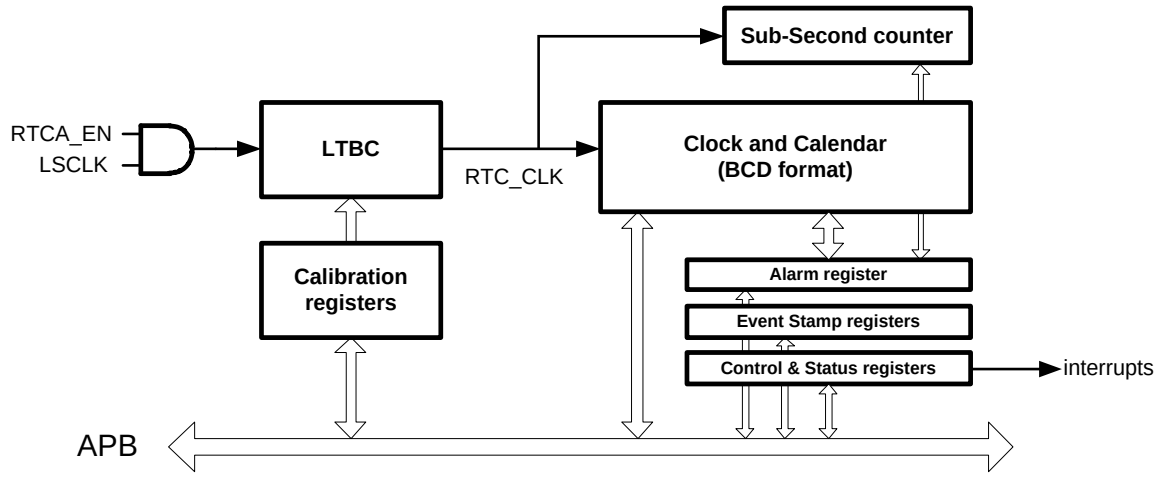


图 28-1 RTC 结构框图

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。

RTC 实现了硬件万年历，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。

28.3 功能说明

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， XTLF 停振时自动将 RTC 时钟切换到 RCLP，RTC 走时有一定误差，但是并不会停止。

28.3.1 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：

- 可通过调整计数周期实现 RTC 时钟的数字调校，调校后最高理论精度+/-0.03ppm
- 通过使用 PLL_L 虚拟调校可得到精确秒时标（无法在低功耗模式下使用）
- 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）
- 预分频电路不受芯片复位影响

28.3.2 LTBC 数字调校

28.3.2.1 +/-0.06ppm

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。

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。FM33A0xxEV 以 256s 为一个调校周期，每个周期内可以调整 0~+/-4095 个 32768Hz 时钟周期，因此最小步长为 30.5us/256s=0.119ppm，最大调校范围为+/-(4095*30.517us/256s)=+/-488.1ppm，调校后平均最小时钟误差为+/-0.06ppm。

调校值由 13bit 寄存器组成，其中最高位为符号位，表示计数值增减，其余 12bit 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 256s 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 12bit 可分为高 4bit 的公共值和 8bit 私有值，其中公共值表示 256s 内每秒都要调整的值，私有值表示 256s 内部分秒需要加减 1。

数字调校值可用如下方式表示：

ADSIGN	ADJUST[11:8]	ADJUST[7:0]
符号位	Common Value I	Differential Value (D)

数字调校公式可表示为：

Correction(ppm) = (C*256 + D)*30.517/256000000

假设使时钟增加0.119ppm，即相当于256s周期内只增加30.5us，调校值写为0_0000_00000001，；
假设使时钟增加488ppm，即相当于256s周期内增加4095个30.5us，调校值写为0_1111_11111111。

调校值举例：

ppm	{ADSIGN,ADJUST}	C	D	Expression
0.238	0_0000_00000010	0	2	2*30.517/256000000
-123.02	1_0100_00001000	4	8	(4*256+8)*30.517/256000000
30.99	0_0001_00000100	1	4	(1*256+4)*30.517/256000000
488.15	0_1111_11111111	15	255	(15*256+255)*30.517/256000000

28.3.2.2 +/-0.03ppm

FM33A0xxEV额外支持以512s为一个调校周期，每个周期内可以调整0~+/-8191个32768Hz时钟周期，因此最小步长为30.5us/512s=0.06ppm，最大调校范围为+/-(8191*30.517us/512s)=+/-488.1ppm，调校后平均最小时钟误差为+/-0.06ppm。

调校值由14bit寄存器组成，其中最高位为符号位，表示计数值增减，其余13bit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512s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13bit可分为高4bit的公共值和9bit私有值，其中公共值表示512s内每秒都要调整的值，私有值表示512s内部分秒需要加减1。

数字调校值可用如下方式表示：

ADSIGN	ADJUST[12:9]	ADJUST[8:0]
符号位	Common Value I	Differential Value (D)

数字调校公式可表示为：

Correction(ppm) = (C*512 + D)*30.517/512000000

假设使时钟增加0.119ppm，即相当于512s周期内只增加2*30.5us，调校值写为0_0000_000000010；
假设使时钟增加488ppm，即相当于512s周期内增加8191个30.5us，调校值写为0_1111_111111111。

调校值举例：

ppm	{ADSIGN,ADJUST}	C	D	Expression

0.238	0_0000_00000010	0	4	4*30.517/512000000
-123.02	1_0100_00001000	4	16	(4*512+16)*30.517/512000000
30.99	0_0001_00000100	1	8	(1*512+8)*30.517/512000000
488.15	0_1111_11111111	15	511	(15*512+511)*30.517/512000000

28.3.3 LTBC 虚拟调校

虚拟调校通过使用16.384Mhz的PLL_L时钟实现。PLL_L时钟频率为32768Hz的500倍，由于频率更高，使用PLL_L计数时，每加减1个时钟周期，对应的微调步长远小于32768Hz时钟，因此可以在短时间内获得精确调整的计数值。

通过虚拟调校方法，可以实现每秒精确的计数结果，并得到高精度秒时标输出。

由于虚拟调校必须使用PLL_L，因此无法在低功耗模式下使用。

28.3.4 亚秒时间计数器

Sub-second计数器使用分频后的64Hz时钟计数，每1s溢出一次，推动BCD时间累加。

Sub-second计数器可以被总线直接读取。

Sub-second计数时间步长是15.625ms。

28.3.5 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

28.3.6 RTC 使能与停止

RTC上电默认不工作，软件应在32.768K晶体振荡器完全起振后再设置当前时间，然后通过设置RTC_CR寄存器启动RTC走时；在晶体振荡器起振之前芯片使用内部环振计时，偏差较大。

28.3.7 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 FM33A0xxEV 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，FM33A0xxEV 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。

28.3.8 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。

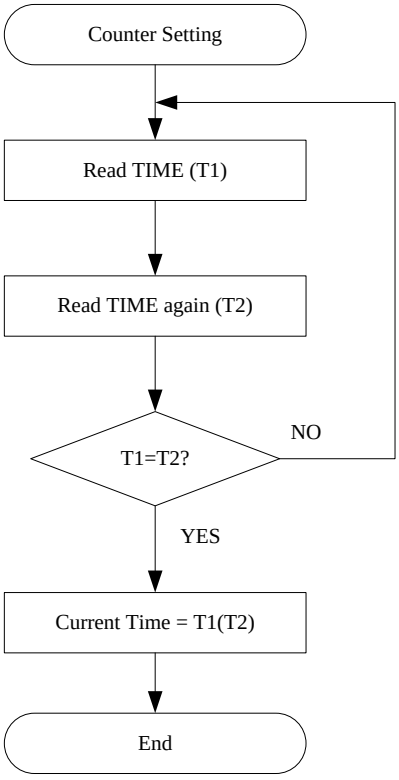


图 28-2 RTC 时间读取流程图

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。

28.3.9 闰年判断

FM33A0xxEV 的 RTC 模块会自动判断闰年。

闰年的条件：(mod 400 == 0) or (mod 4 ==0 and mod 100 <> 0)

这里只考虑 2000-2099 范围，符合闰年条件时 RTC 自动调整计数。

28.4 寄存器

模块起始地址为 0x4001_1000

offset 地址	名称	符号
RTC(模块起始地址:0x40011000)		
0x00	RTC 写使能寄存器 (RTC Write Enable Register)	RTC_WER
0x04	RTC 中断使能寄存器 (RTC Interrupt Enable Register)	RTC_IER
0x08	RTC 中断标志寄存器 (RTC Interrupt Status Register)	RTC_ISR
0x0C	RTC BCD 秒寄存器 (RTC BCD Second register)	RTC_BCDSEC
0x10	RTC BCD 分钟寄存器 (RTC BCD Minute register)	RTC_BCDMIN
0x14	RTC BCD 小时寄存器 (RTC BCD Hour register)	RTC_BCDHOUR
0x18	RTC BCD 日期寄存器 (RTC BCD Date register)	RTC_BCDDATE
0x1C	RTC BCD 星期寄存器 (RTC BCD Week register)	RTC_BCDWEEK
0x20	RTC BCD 月寄存器 (RTC BCD Month register)	RTC_BCDMONTH
0x24	RTC BCD 年寄存器 (RTC BCD Year register)	RTC_BCDYEAR
0x28	RTC 闹钟设置寄存器 (RTC Alarm Register)	RTC_ALARM
0x2C	RTC 时间信号输出寄存器 (RTC Time Mark Select)	RTC_TMSEL
0x30	RTC 数值调整寄存器 (RTC time Adjust Register)	RTC_ADJUST
0x34	RTC 数值调整方向寄存器 (RTC time Adjust Sign Register)	RTC_ADSIGN
0x38	RTC 虚拟调校寄存器 (RTC Virtual Calibration Register)	RTC_VCAL
0x3C	-	-
0x40	RTC 调校步长寄存器 (RTC Calibration Step Register)	RTC_CALSTEP
0x44	RTC 补偿周期寄存器 (RTC Adjust Counter Register)	RTC_ADCNT
0x48	RTC 亚秒计数寄存器 (RTC Sub-Second Register)	RTC_SSR
0x4C	-	-
0x50	RTC 秒时标占空比寄存器 (RTC Time Mark Duty Cycle Register)	RTC_DTR
0x7C	RTC 控制寄存器 (RTC Control Register)	RTC_CR

28.4.1 RTC 写使能寄存器（RTC_WER）

名称	RTC_WER							
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	-							RTCWE
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	RTCWE	RTC 写使能寄存器， 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。

28.4.2 RTC 中断使能寄存器（RTC_IER）

名称	RTC_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	-			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	DATE_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。 1：中断使能打开 0：中断使能禁止

位号	助记符	功能描述
11	ALARM_IE	闹钟中断使能。 1：中断使能打开 0：中断使能禁止
10	1KHZ_IE	1khz 中断使能。 1：中断使能打开 0：中断使能禁止
9	256HZ_IE	256hz 中断使能。 1：中断使能打开 0：中断使能禁止
8	64HZ_IE	64hz 中断使能。 1：中断使能打开 0：中断使能禁止
7	16HZ_IE	16hz 中断使能。 1：中断使能打开 0：中断使能禁止
6	8HZ_IE	8hz 中断使能。 1：中断使能打开 0：中断使能禁止
5	4HZ_IE	4hz 中断使能。 1：中断使能打开 0：中断使能禁止
4	2HZ_IE	2hz 中断使能。 1：中断使能打开 0：中断使能禁止
3	SEC_IE	秒中断使能。 1：中断使能打开 0：中断使能禁止
2	MIN_IE	分中断使能。 1：中断使能打开 0：中断使能禁止
1	HOUR_IE	小时中断使能。 1：中断使能打开 0：中断使能禁止
0	DATE_IE	天中断使能。 1：中断使能打开 0：中断使能禁止

28.4.3 RTC 中断标志寄存器（RTC_ISR）

名称	RTC_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	-			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	DATE_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 1：中断置位 0：无中断产生
11	ALARM_IF	闹钟中断标志。写 1 清零 1：中断置位 0：无中断产生
10	1KHZ_IF	1khz 中断标志。写 1 清零 1：中断置位 0：无中断产生
9	256HZ_IF	256hz 中断标志。写 1 清零 1：中断置位 0：无中断产生
8	64HZ_IF	64hz 中断标志。写 1 清零 1：中断置位 0：无中断产生
7	16HZ_IF	16hz 中断标志。写 1 清零 1：中断置位 0：无中断产生
6	8HZ_IF	8hz 中断标志。写 1 清零 1：中断置位 0：无中断产生
5	4HZ_IF	4hz 中断标志。写 1 清零 1：中断置位 0：无中断产生
4	2HZ_IF	2hz 中断标志。写 1 清零 1：中断置位 0：无中断产生
3	SEC_IF	秒中断标志。写 1 清零 1：中断置位 0：无中断产生
2	MIN_IF	分中断标志。写 1 清零 1：中断置位 0：无中断产生
1	HOUR_IF	小时中断标志。写 1 清零 1：中断置位 0：无中断产生
0	DATE_IF	天中断标志。写 1 清零 1：中断置位 0：无中断产生

28.4.4 RTC BCD 时间秒寄存器（RTC_BCDSEC）

名称	RTC_BCDSEC							
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	-	BCDSEC						
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	BCDSEC	秒时间数值，BCD 格式。

28.4.5 RTC BCD 时间分钟寄存器（RTC_BCDMIN）

名称	RTC_BCDMIN							
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	-	BCDMIN						
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	BCDMIN	分钟时间数值，BCD 格式。

28.4.6 RTC_BCD 时间小时寄存器（RTC_BCDHOUR）

名称	RTC_BCDHOUR							
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	-		BCDHOUR[5:0]					
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	BCDHOUR	小时数值，BCD 格式。

28.4.7 RTC BCD 时间天寄存器（RTC_BCDDATE）

名称	RTC_BCDDATE							
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	-		BCDDATE					
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	BCDDATE	天数值，BCD 格式。

28.4.8 RTC BCD 时间星期寄存器（RTC_BCDWEEK）

名称	RTC_BCDWEEK							
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	-					BCDWEEK		
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	BCDWEEK	周数值，BCD 格式。

28.4.9 RTC BCD 时间月寄存器（RTC_BCDMONTH）

名称	RTC_BCDMONTH							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			BCDMONTH				
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	BCDMONTH	月数值，BCD 格式。

28.4.10 RTC BCD 时间年寄存器（RTC_BCDYEAR）

名称	RTC_BCDYEAR							
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	BCDYEAR							
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	BCDYEAR	年数值，BCD 格式。

28.4.11 RTC 闹钟寄存器（RTC_ALARM）

名称	RTC_ALARM							
offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ALMEN	-						
位权限	R/W-0	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		ALARMHOUR					
位权限	U-0		R/W-00000					
位	Bit15	Bit14	BIT13	BIT12	BIT11	BIT10	Bit9	Bit8
位名	-	ALARMMIN						
位权限	U-0	R/W-00 0000						
位	Bit7	Bit6	BIT5	BIT4	BIT3	BIT2	Bit1	Bit0
位名	-	ALARMSEC						
位权限	U-0	R/W-000 0000						

位号	助记符	功能描述
31	ALMEN	闹钟功能使能 (alarm enable) 1：使能 Alarm 功能 0：禁止 Alarm 功能
30:22	-	RFU：未实现，读为 0
21:16	ALARMHOUR	闹钟的小时数值。
15	-	RFU：未实现，读为 0
14:8	ALARMMIN	闹钟的分数值。
7	-	RFU：未实现，读为 0
6:0	ALARMSEC	闹钟的秒数值。

28.4.12 RTC 时钟信号输出控制寄存器（RTC_TMSEL）

名称	RTC_TMSEL							
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	-	FSEL
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	FSEL	频率输出选择信号： 0000：输出 PLL 分频得到的精确 1 秒方波 0001：输出 PLL 分频的可变占空比的精确秒时标 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 256 秒方波信号 1000：反向输出 PLL 分频的可变占空比的精确秒时标 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：反向输出 PLL 分频的精确 1s 方波信号 1111：输出 RTC 内部秒时标方波

28.4.13 RTC LTBC 数值调整寄存器（RTC_ADJUST）

名称	RTC_ADJUST							
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[12:8]				
位权限	U-0			R/W-XXXX				
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
31:13	-	RFU：未实现，读为 0
12:0	ADJUST	LTBC 补偿调整数值（原码格式）

28.4.14 RTC LTBC 数值调整方向寄存器（RTC_ADSIGN）

名称	RTC_ADSIGN							
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	-							ADSIGN
位权限	U-0							R/W-x

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	ADSIGN	LTBC 补偿方向 0：表示增加计数初值 1：表示减少计数初值

28.4.15 RTC 虚拟调校寄存器（RTC_VCAL）

名称	RTC_VCAL							
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	-							PR1SEN
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	PR1SEN	虚拟调校使能信号 0：表示禁止虚拟调校功能 1：表示使能虚拟调校功能

28.4.16 RTC 调校步长寄存器（RTC_CALSTEP）

名称	RTC_CALSTEP							
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	-						CALSTEP	
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	CALSTEP	LTBC 最小调校步长选择 10：0.06ppm（精度+/-0.03ppm） 01：0.119ppm（精度+/-0.06ppm） 00/11：0.238ppm（精度+/-0.119ppm）

28.4.17 RTC 补偿周期寄存器（RTC_ADCNT）

名称	RTC_ADCNT							
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	-							ADJCNT[8]
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADJCNT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:9	-	RFU：未实现，读为 0
8:0	ADJCNT	RTC 补偿周期计数器值，只读，软件通过查询这个寄存器可以知道当前秒所处的调校周期中的位置。

位号	助记符	功能描述
		计数范围 0~511，对应最大调校周期 512 秒。 当 BCD 秒寄存器有写入行为时，此计数器被清零。

28.4.18 RTC 亚秒计数寄存器（RTC_SSR）

名称	RTC_SSR							
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			SSR					
位权限			R-00 0000					

位号	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5:0	SSR	亚秒计数器 (sub-second counter register) 计数步长 15.625ms

28.4.19 RTC 秒时标占空比寄存器（RTC_DTR）

名称	RTC_DTR							
offset	0x50							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							DUTY[24]
位权限	U-0							R/W-1
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DUTY[23:16]							
位权限	R/W-1010 0100							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DUTY[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DUTY[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:25	-	RFU：未实现，读为 0
24:0	DUTY	RTC 虚拟秒时标输出占空比，定义 1 秒周期内高电平长度，复位默认

位号	助记符	功能描述
		值是 80ms 高电平

28.4.20 RTC 控制寄存器（RTC_CR）

名称	RTC_CR							
offset	0x7C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							RTC_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	RTC_EN	RTC 时钟和日历使能 (RTC Enable) 1：使能 RTC 走时 0：停止 RTC 走时 此寄存器用于门控 LSCLK 输入，如果清零，RTC 将立即停止走时；但是虚拟调校功能不受此寄存器影响。

29 LCD 显示

29.1 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 驱动支持片内电阻模式
- 支持 VLCD 升压电路，最高输出电压不低于 4.5V
- 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

29.2 结构框图

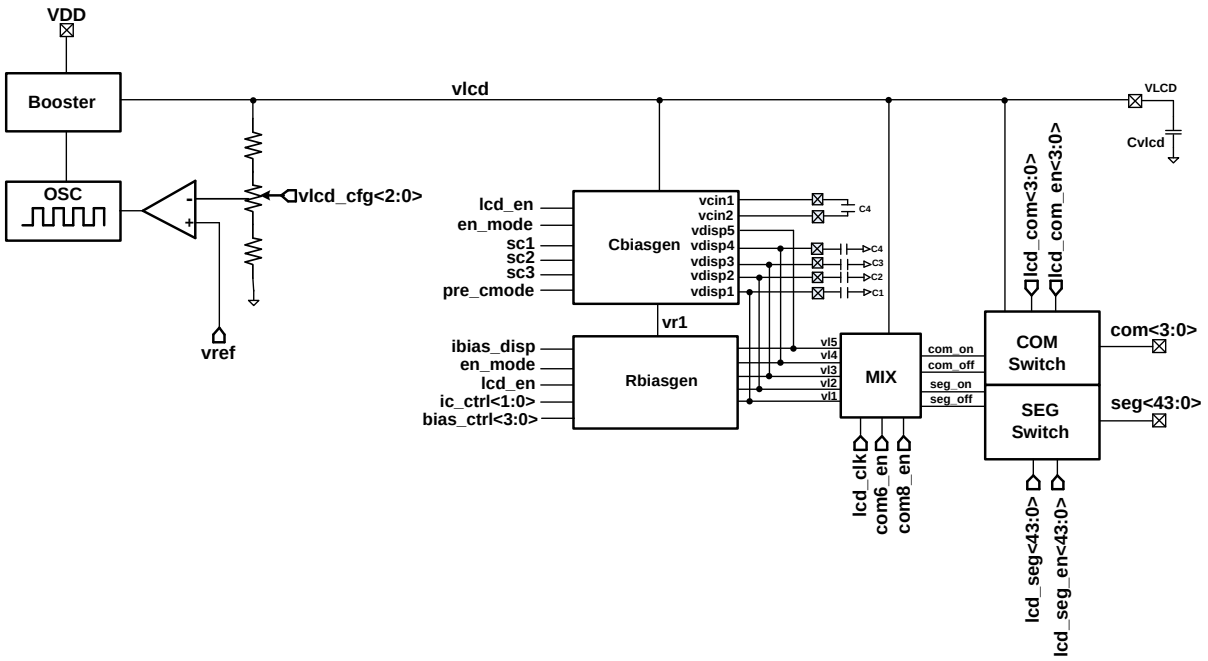


图 29-1 LCD 显示控制模块结构框图

29.3 IO 配置

LCD驱动电路工作时最多会占用48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。

29.4 功能说明

29.4.1 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29-1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29-2典型帧频率和DF的关系

29.4.2 LCD Type A 扫描波形

下图是1/4 duty，1/3bias，TypeA类波形的LCD扫描波形示意图。这里只画出了两个公共端的示例。

4个公共端会依次有效，在某个公共端有效的时间内，SEG输出合适的电平，与COM电平共同施加在LCD面板上，压差大的段码将被显示，压差小的段码不显示。

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。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。

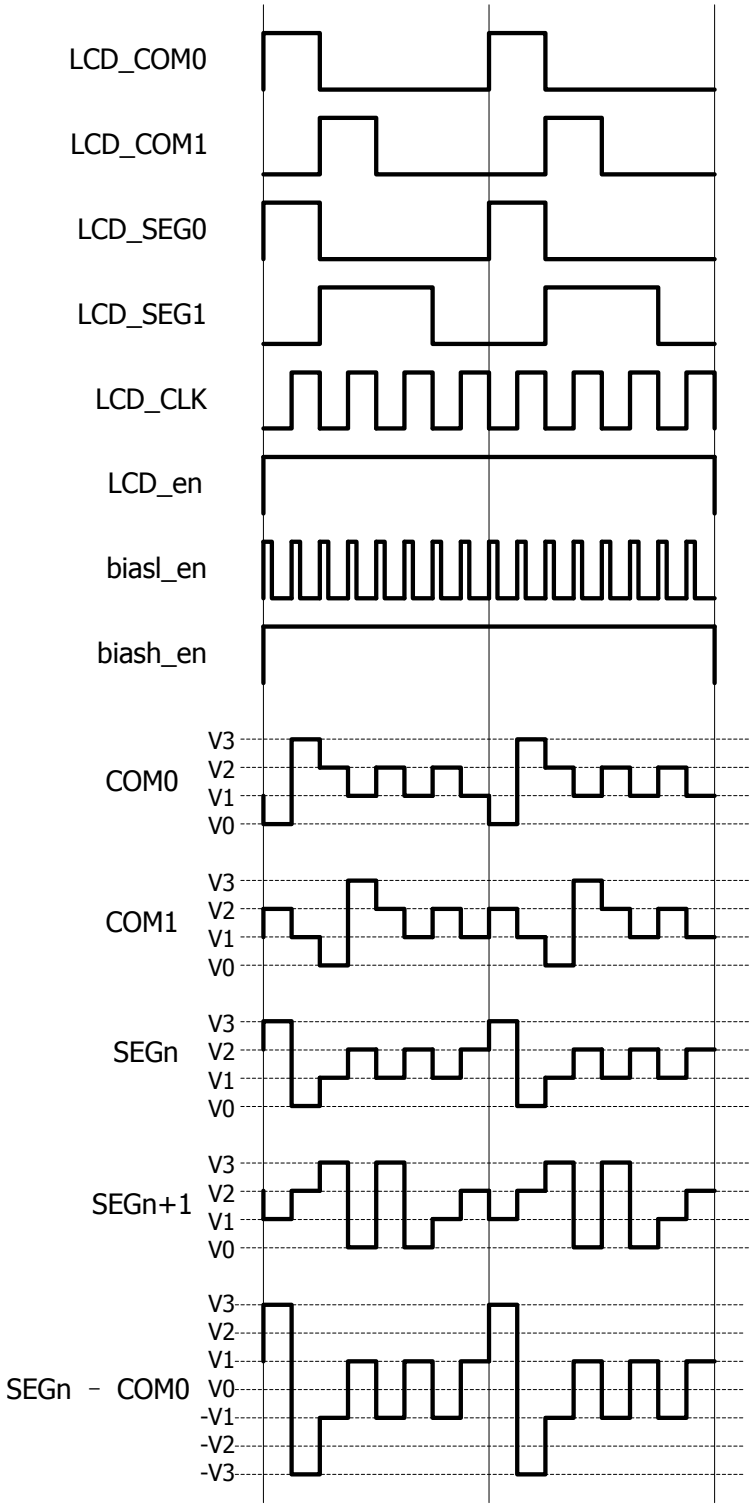


图 29-2 LCD 驱动波形(1/4 duty, 1/3 bias, type A)

29.4.3 LCD Type B 扫描波形

下图是1/4 duty，1/3bias，TypeB类波形的LCD扫描波形示意图。这里只画出了两个公共端的示例。

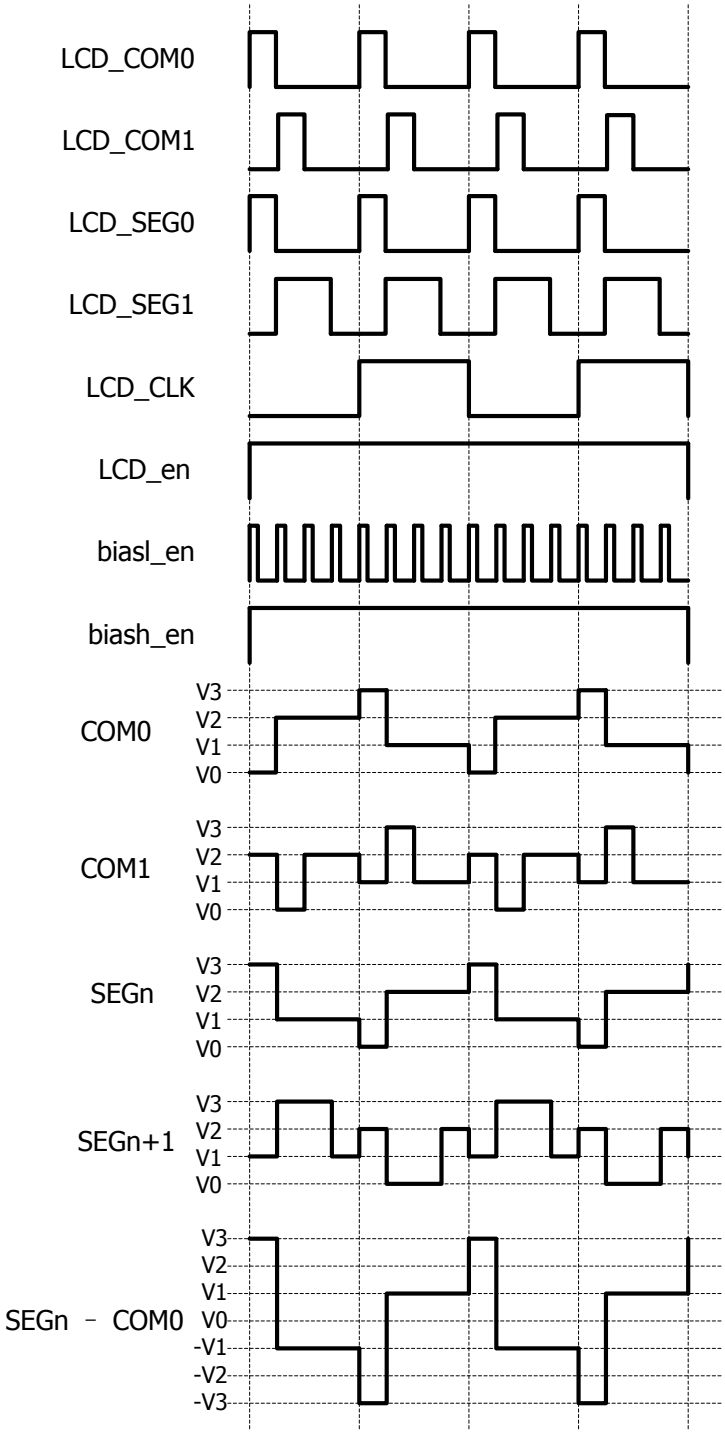


图 29-3 LCD 驱动波形(1/4 duty, 1/3 bias, type B)

29.4.4 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：

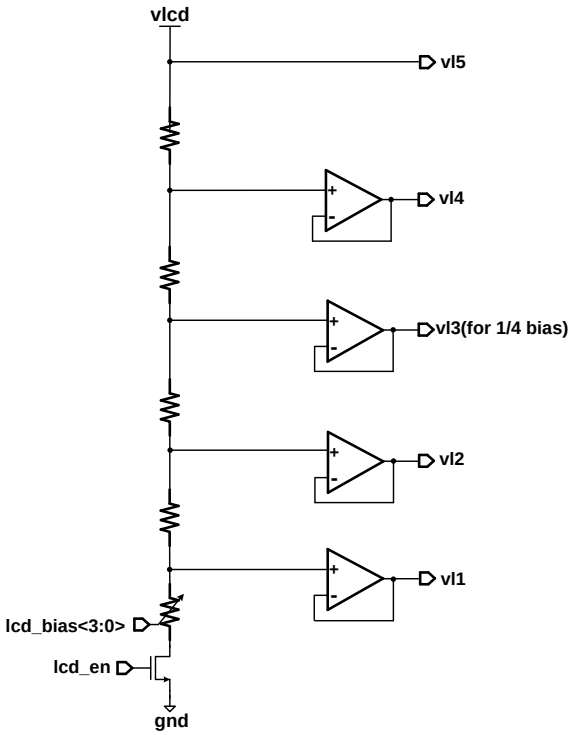


图 29-4 LCD 片内电阻 buffer 型驱动电路

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30.4.7“偏置电压调整”章节。

29.4.5 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Tstep = COM*DF[7:0]*2*16/32768Hz，实际ON/OFF时间为TON/TOFF * Tstep。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。

29.4.6 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：

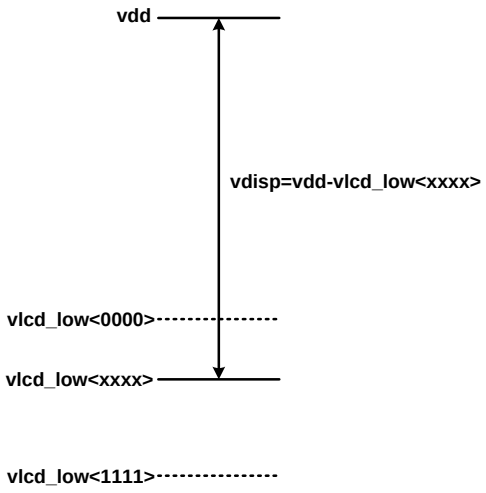


图 29-5 偏置电压调整

应用中应根据实际 LCD 面板特性，选择合适的 VDISP 电压。

29.5 寄存器

模块起始地址：0x4001_0C00

offset 地址	名称	符号
LCD(模块起始地址:0x40010C00)		
0x00	显示控制寄存器 （LCD Control Register）	LCD_CR
0x04	显示测试控制寄存器 （LCD test Register）	LCD_TEST
0x08	显示频率控制寄存器 （LCD Frequency Control Register）	LCD_FCR
0x0C	闪烁时间寄存器 （LCD Flick Time Register）	LCD_FLKT
0x14	显示中断使能寄存器 （LCD Interrupt Enable Register）	LCD_IER
0x18	显示中断标志寄存器 （LCD Interrupt Status Register）	LCD_ISR
0x24	显示数据缓存寄存器 0 （LCD data buffer registers 0）	LCD_DATA0
0x28	显示数据缓存寄存器 1 （LCD data buffer registers 1）	LCD_DATA1
0x2C	显示数据缓存寄存器 2 （LCD data buffer registers 2）	LCD_DATA2
0x30	显示数据缓存寄存器 3 （LCD data buffer registers 3）	LCD_DATA3
0x34	显示数据缓存寄存器 4 （LCD data buffer registers 4）	LCD_DATA4
0x38	显示数据缓存寄存器 5 （LCD data buffer registers 5）	LCD_DATA5
0x3C	显示数据缓存寄存器 6 （LCD data buffer registers 6）	LCD_DATA6
0x40	显示数据缓存寄存器 7 （LCD data buffer registers 7）	LCD_DATA7
0x50	COM 使能控制寄存器 （LCD COM Enable Register）	LCD_COMEN
0x54	SEG 使能控制寄存器 0 （LCD SEG Enable Register0）	LCD_SEGEN0
0x58	SEG 使能控制寄存器 1 （LCD SEG Enable Register1）	LCD_SEGEN1
0x5C	LCD Boost 控制寄存器 （LCD Boost Control Register）	LCD_BSTCR

29.5.1 显示控制寄存器（LCD_CR）

名称	LCD_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	-	SCFSEL			SC_CTRL		IC_CTRL	
位权限	U-0	R/W-00			R/W-00		R/W-01	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ENMODE	FLICK	-		BIAS			
位权限	R/W-0	R/W-0	U-0		R/W-1110			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		BIASMD	ANTIPOLAR	WFT	LMUX		EN
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
31:23	-	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
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：RFU 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	**BIAS**	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 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

位号	助记符	功能描述
		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	**EN**	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

29.5.2 显示测试控制寄存器（LCD_TEST）

名称	**LCD_TEST**							
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	LCCTRL	-						TESTEN

位权限	R/W-0	U-0	R/W-0

位号	助记符	功能描述
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。
6:1	-	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

29.5.3 显示频率控制寄存器（LCD_FCR）

名称	LCD_FCR							
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	DF[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:8	-	未实现，读为0
7:0	**DF**	显示预分频寄存器 (Display Frequency)

29.5.4 闪烁时间寄存器（LCD_FLKT）

名称	LCD_FLKT
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	TOFF							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TON							
位权限	R/W-0000 0000							

位号	助记符	功能描述
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Tstep = COM*DF[7:0]*2*16/32768Hz，实际OFF时间为TOFF * Tstep
7:0	**TON**	闪烁显示点亮时间寄存器 (Display-On Time) TON最小步长为Tstep = COM*DF[7:0]*2*16/32768Hz，实际ON时间为TON * Tstep

29.5.5 显示中断使能寄存器（LCD_IER）

名称	LCD_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	-						DONIE	DOFFIE
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	**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
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

29.5.6 显示中断标志寄存器（LCD_ISR）

名称	LCD_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	-						DONIF	DOFFIF
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	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

29.5.7 显示数据寄存器（LCD_DATAx）

LCD 显示模块内有 10 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)							
offset	0x24+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

29.5.7.1 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_DATA1	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_DATA2	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_DATA3	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	SEG29 COM3	SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
LCD_DATA4	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_DATA5	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

29.5.7.2 6COM 显示数据寄存器

名称	6com 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_DATA1	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_DATA2	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_DATA3	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	SEG29 COM3	SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
LCD_DATA4	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	SEG29 COM4	SEG28 COM4	SEG27 COM4	SEG26 COM4	SEG25 COM4	SEG24 COM4
LCD_DATA5	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
	SEG31 COM5	SEG30 COM5	SEG29 COM5	SEG28 COM5	SEG27 COM5	SEG26 COM5	SEG25 COM5	SEG24 COM5
LCD_DATA6	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	SEG43 COM0	SEG42 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

	SEG35 COM2	SEG34 COM2	SEG33 COM2	SEG32 COM2	SEG43 COM1	SEG42 COM1	SEG39 COM1	SEG38 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG33 COM3	SEG32 COM3	SEG43 COM2	SEG42 COM2	SEG39 COM2	SEG38 COM2	SEG37 COM2	SEG36 COM2
LCD_DATA7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG43 COM3	SEG42 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	SEG43 COM4	SEG42 COM4
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
					SEG43 COM5	SEG42 COM5	SEG39 COM5	SEG38 COM5

29.5.7.3 8COM 显示数据寄存器

名称	8com 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_DATA1	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_DATA2	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 COM02	SEG29 COM2	SEG28 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
LCD_DATA3	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	SEG29 COM3	SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
LCD_DATA4	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	SEG29 COM4	SEG28 COM4	SEG27 COM4	SEG26 COM4	SEG25 COM4	SEG24 COM4
LCD_DATA5	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
	SEG31 COM5	SEG30 COM5	SEG29 COM5	SEG28 COM5	SEG27 COM5	SEG26 COM5	SEG25 COM5	SEG24 COM5
LCD_DATA6	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
	SEG31 COM6	SEG30 COM6	SEG29 COM6	SEG28 COM6	SEG27 COM6	SEG26 COM6	SEG25 COM6	SEG24 COM6

LCD_DATA7	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_DATA8	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_DATA9	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

29.5.8 LCD COM 使能控制寄存器（LCD_COMEN）

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

位权限	R-0000	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 1：COM 输出使能 0：COM 输出禁止

29.5.9 LCD SEG 使能控制寄存器 0（LCD_SEGEN0）

名称	LCD_SEGEN0							
offset	0x54							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SEGENx[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SEGENx[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SEGENx[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SEGENx[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	SEGENx	LCD SEG 输出使能控制，SEGEN0~SEGEN31 1: SEG 输出使能 0: SEG 输出禁止

29.5.10 LCD SEG 使能控制寄存器 1（LCD_SEGEN1）

名称	LCD_SEGEN1							
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	-				SEGENx[11:8]			
位权限	U-0				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SEGENx[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
31:12	-	RFU：未实现，读为 0
11:0	SEGENx	LCD SEG 输出使能控制，SEGEN32~SEGEN43 1：SEG 输出使能 0：SEG 输出禁止

29.5.11 LCD Boost 控制寄存器（LCD_BSTCR）

名称	LCD_BSTCR							
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	OSC_CFG				-		BUF_OFF	BUFBYP
位权限	R/W-0000				U-0		R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	VLCDPD	-	VLCDCFG				BOOST_TEN	BOOST_EN
位权限	R/W-0	U-0	R/W-0100				R/W-0	R/W-0

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:12	OSC_CFG	BOOST 环振频率配置信号
11:10	-	RFU：未实现，读为 0
9	BUF_OFF	基准源 Buffer 使能信号 1：关闭 Buffer 0：打开 Buffer
8	BUFBYP	基准源 Buffer Bypass 1：Bypass Buffer 0：不 Bypass
7	VLCDPD	VLCD 下拉使能 1：开启 VLCD 下拉 0：关闭 VLCD 下拉
6	-	RFU：未实现，读为 0
5:2	VLCDCFG	VLCD 输出电压配置 0000：2.4V 0001：2.6V 0010：2.8V 0011：3.0V 0100：3.2V 0101：3.4V 0110：3.6V 0111：3.8V

位号	助记符	功能描述
		1000：4.0V 1001：4.2V 1010：4.4V 1011：4.6V 1100：4.8V 1101：5.0V 1110：5.2V 1111：5.4V
1	BOOST_TEN	Booster 测试信号 1：Booster 测试模式 0：正常模式
0	BOOST_EN	Booster 使能信号 1：开启 VLCD Booster 0：关闭 VLCD Booster

30 ADC 和温度传感器

30.1 概述

FM33A0xxEV 带有∑-ΔADC，可实现温度、电池电压或其他直流信号的测量功能。

主要特点为：

- 工作电压 2.2～5.5V
- 测量外部信号输入时，输入电压范围 0.2~4.8V（使能输入 buffer）或 0~4.8V（bypass 输入 buffer）
- 最高分辨率可达 16bits
- 温度传感器
- 最多 12 个外部输入通道（LQFP100）或 8 个外部输入通道（LQFP80）
- 累加器单次转换时间典型值 2ms（工作时钟 1MHz，trim=0x7FF）
- CIC 滤波器输出采用可编程 OSR，8~1024 倍抽取

30.2 模块框图

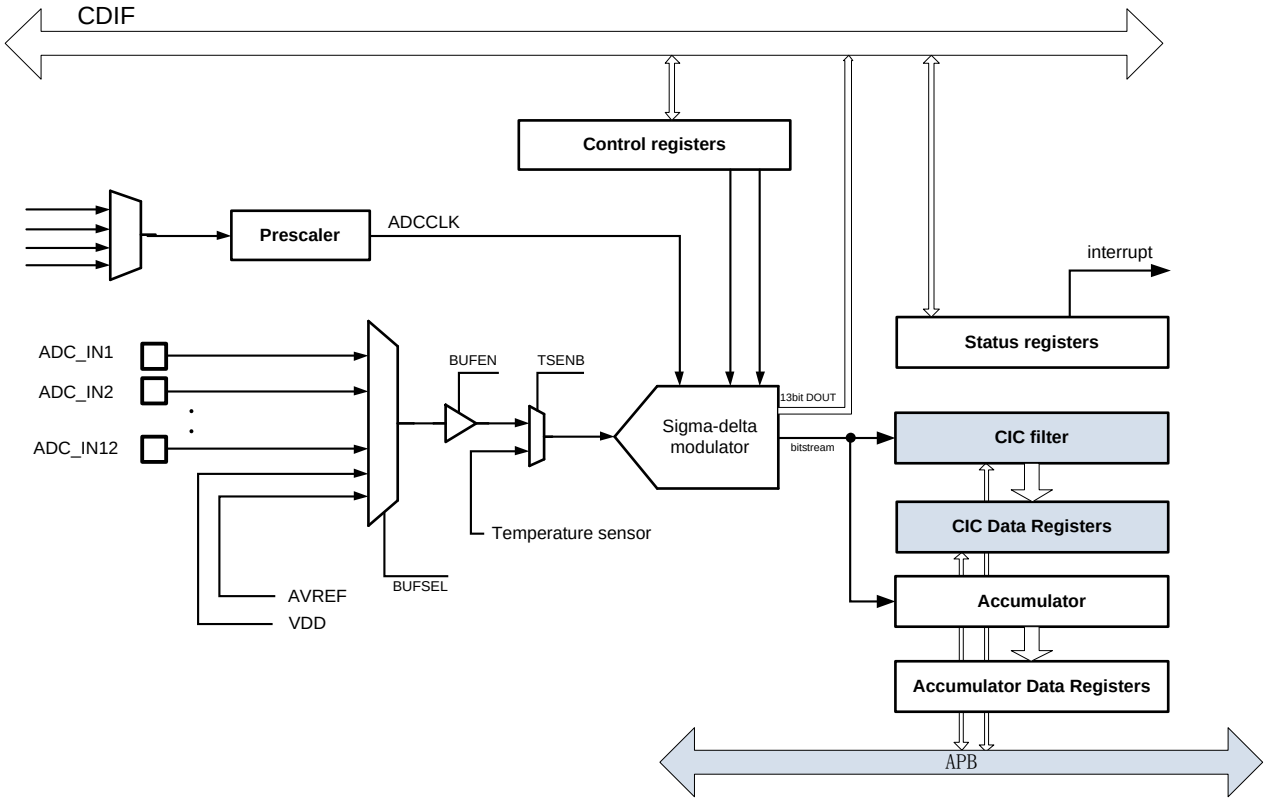


图 30-1 ADC 结构框图

注:buffer 用于增大被采样信号的驱动能力

30.3 工作模式

ADC 采用 sigma-delta modulator 配合过采样输出的方式实现模数转换。

ADC 支持 3 种输出方式：

- 内部累加器输出
- 外部累加器输出
- CIC 滤波器输出

通过 MODE 寄存器可以选择工作模式。

内部累加器模式转换结束后 ADC_DONE 置高并保持，输出数据稳定并保持，直到 EN 信号变低。若 EN 保持为 1，则 ADC 将进行连续转换，每次转换完成后都会产生 ADC 中断，软件清零。关闭 ADC 使能后自动清除 DONE 信号。

外部累加器模式使用数字电路对 ADC 输出的 1bit 码流进行求和累加，软件可以灵活配置累加周期，以平衡输出位数和累加时间。最短累加周期为 1023，最长为 16383，可以得到 10~14bit 无符号输出数据。

CIC 滤波器模式则采用 CIC 抽取滤波滤波器完成对 SDM 码流的积分和转换，根据 OSR 配置不同，转换速率可能是 1Ksps~128Ksps。每次 CIC 输出一个有效数据，将产生一个 CIC 转换完成中断。

不论是累加器输出还是 CIC 输出，结果都会被写入 ADC_DATA 寄存器。累加器模式下，ADC_DATA 寄存器仅最低 13bit 有效，而 CIC 模式下最大字长可能达到 24bit，软件可以根据需要抛弃无效的高位和低位。

30.4 电压测量（TBD）

30.4.1 测量电源电压

30.4.2 测量外部通道输入

30.5 温度传感器

芯片内置高精度温度传感器，可精确测量芯片工作时的环境温度。可利用温度传感器测量的温度值，对RTC的走时进行补偿。

温度传感器特性：

- 温度测量范围：-40°C~85°C
- 温度测量精度/分辨率：±0.02°C（CIC 模式）
- 典型条件下全温区线性度不低于±2°C
- 与芯片内的 sigma-delta ADC 配合，实现温度到数字输出的转换

温度传感器的输出为随温度而变化的电压量。可以使用 ADC 来测量温度传感器的输出，需要先将 ADC 的输入通道选择为温度传感器输入（ADC_VANA_EN=0）。

30.5.1 温度调校

温度传感器输出电压随温度的变化关系是线性的，在未经校准的情况下可以用来测量相对温度的变化。如果需要测量绝对温度，需要进行offset调校和/或slope调校。

FM33A0xxEV芯片出厂前进行单温度点offset调校，调校的环境温度为30℃±1℃，转换结果存储在Flash的特殊信息区。

30.5.2 使用方式

请调用复旦微电子提供的库函数进行温度测量，详情可参考应用笔记。

30.6 CIC 模式

一阶sigma-delta调制器的码流经过CIC降采样滤波后，可以得到最高16位结果，相应的得到更好的温度传感器分辨率。

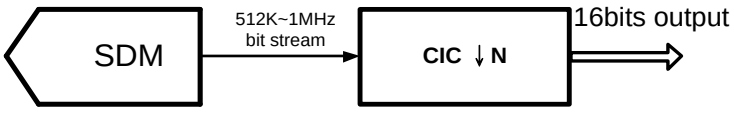


图 30-2 SDM 抽取滤波示意图

CIC的抽取倍数都是软件可配置的，默认值N=512，即OSR=512。当输入码流为1Msps时，OSR=32得到32Ksps的输出结果。

软件可以灵活配置N的值，更大的OSR可以得到更高的精度分辨率，但是输出结果的采样率会同比例降低；而较小的OSR能够得到更高的采样率，代价是降低有效位数。

通过增加抽取倍数，可以不断压缩有效信号带宽并提升信噪比，从而获得更大的输出位数。同时，也可以通过减小抽取倍数来获得更快的转换速度。假设将抽取倍数修改为8，采样率提高到128Ksps，输出最大字长为13bit。

ADC使能后，软件还需要置位CIC滤波器使能，由于CIC滤波器有一个输出建立延迟，最初的几个输出结果是不可用的。软件可以通过NS_DISC寄存器，配置ADC使能后自动丢弃前0~127个转换结果。丢弃指定个数的转换结果后，硬件才置位CIC_IF产生中断和DMA请求。

下图是NS_DISC=4的情况下，CIC中断或DMA请求输出的示意图。

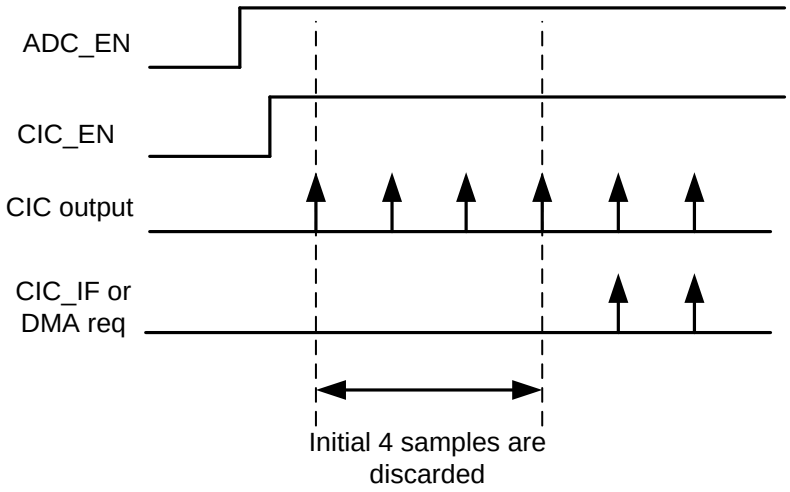


图 30-3 CIC 初始输出采样点丢弃

注意：当CIC抽取后的实际采样率高于2Ksps时，输出数据会围绕实际结果上下周期性波动，需要通过软件平均得到精确结果。

30.7 中断

ADC、累加器和CIC都能产生各自中断，并且都可以独立屏蔽。

模式	寄存器	说明
内部累加	ADC_DONE	仅供查询，只读
	ADC_IF	可中断，可屏蔽，每次转换完成后置位，软件清零
外部累加	INIT_RDY	仅供查询，只读
	ACC_IF	可中断，可屏蔽，每次累加完成后置位，软件清零
CIC滤波	INIT_RDY	仅供查询，只读
	CIC_OVR	CIC输出溢出，可中断，可屏蔽，硬件置位，软件清零
	CIC_IF	CIC数据更新，可中断，可屏蔽，硬件置位，软件清零

表 30-1 ADC 相关中断和标志

注意：当MODE=1时，ADC外部累加器就会开始工作，不论是否启动CIC滤波，然而外部累加器的工作不会影响CIC。如果只想使用CIC，可以屏蔽ACC_IF中断。

30.8 DMA

在使能了DMA通道后，每次产生CIC_IF中断都会触发DMA请求，DMA将CIC输出数据搬运到RAM指定地址。

DMA只支持搬运CIC输出，不支持搬运ADC内部或外部累加器输出。

30.9 ADC 外部输入通道

根据BUFSEL寄存器配置，可以选择以下外部GPIO进行测量：

BUFSEL寄存器	ADC输入通道	GPIO
0001	ADC_IN1	PC12
0010	ADC_IN2	PC13
0011	ADC_IN3	PD0
0100	ADC_IN4	PD1
0101	ADC_IN5	PF6
0110	ADC_IN6	PC15
0111	ADC_IN7	PB2
1000	ADC_IN8	PB3
1001	ADC_IN9	PH0
1010	ADC_IN10	PH1
1011	ADC_IN11	PH2
1100	ADC_IN10	PH3

表 30-2 ADC 输入通道选择

进行测量前，应用需将对应GPIO配置成模拟功能，并且必须关闭上拉电阻。

30.10 寄存器

offset 地址	名称	符号
ADC(模块起始地址:0x4001FA00)		
0x00	ADC 控制寄存器 （ADC Control Register）	ADC_CR
0x04	ADC 调校寄存器 （ADC Trim Register）	ADC_TRIM
0x08	ADC 输出数据寄存器 （ADC Data Register）	ADC_DR
0x0C	ADC 中断标志寄存器 （ADC Interrupt Status Regsiter）	ADC_ISR
0x10	ADC 配置寄存器 （ADC Config Register）	ADC_CFGR
CIC(模块起始地址:0x40015C00)		
0x00	CIC 输出数据寄存器 （CIC Output Data Register）	CIC_DR
0x04	CIC 输出偏移寄存器 （CIC Output Offset Register）	CIC_OS
0x08	CIC 无符号输出结果寄存器 （CIC Unsigned Data Register）	CIC_USDR
0x0C	CIC 滤波器控制寄存器 （CIC Control Register）	CIC_CR
0x10	CIC 中断标志寄存器 （CIC Interrupt Status Register）	CIC_ISR

30.10.1 ADC 控制寄存器(ADC_CR)

名称	ADC_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	ADC_IE	ACC_IE	-		HPEN	MODE	RSTCTRL_EN	EN
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	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7	ADC_IE	ADC 内部累加模式中断使能
6	ACC_IE	ADC 外部累加模式中断使能

位号	助记符	功能描述
5:4	-	RFU：未实现，读为 0
3	HPEN	ADC 功耗模式 0：低功耗模式，工作时钟最高 1MHz 1：高功耗模式，工作时钟最高 2MHz
2	MODE	ADC 工作模式 0：内部累加器模式 1：外部累加器或 CIC 模式
1	RSTCTRL_EN	积分器复位使能，在 MODE=1 并且使用外部累加器时，必须置位，其他条件下必须保持为 0 0：禁止积分器外部复位 1：允许积分器外部复位
0	EN	ADC 使能信号 0：ADC 不使能 1：ADC 使能

30.10.2 ADC 调校寄存器(ADC_TRIM)

名称	ADC_TRIM							
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	-					ADC_TRIM[10:8]		
位权限	U-0					RW-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADC_TRIM[7:0]							
位权限	RW-0000 0000							

位号	助记符	功能描述
31:11	-	RFU：未实现，读为 0
10:0	ADC_TRIM	ADC TRIM 值，仅针对内部累加器模式输出 内部累加器累加周期：Period = TRIM*2*T_{ADC_CLK} 累加周期决定了输出数据位宽，当 TRIM=0x7FF 最大值时，实际累加周期为 4095 cycle，即 ADC 输出数据最大有效位宽是 12bit

30.10.3 ADC 输出数据寄存器(ADC_DR)

名称	ADC_DR							
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	ADC_DATA[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADC_DATA[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	ADC_DATA	ADC 输出数据，为码流累加结果，未经降采样滤波器处理 当 MODE=0 时，输出为 ADC 内部累加结果，最高 12bit，位数由 ADC_TRIM 决定 当 MODE=1 时，输出位 ADC 外部累加结果，最高 14bit，位数由 ACC_PERIOD 决定

30.10.4 ADC 中断标志寄存器(ADC_ISR)

名称	ADC_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	-						INIT_RDY	ACC_IF
位权限	U-0						R-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						ADC_DONE	ADC_IF
位权限	U-0						R-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9	INIT_RDY	ADC 初始化完成标志，仅 MODE=1 时有效（不产生中断） 0：ADC 还未完成初始化 1：ADC 完成初始化 *注：MODE=1，ADC_EN 置位后，等待 MODE_CTRL_DELAY 时间之后，此标志置位，表示 ADC 内部积分器建立完成。MODE=0 时，工作时序由 ADC 内部自行控制，此标志无效。*
8	ACC_IF	累加器完成中断标志，硬件置位，软件写 1 清零，写 0 无效
7:2	-	RFU：未实现，读为 0
1	ADC_DONE	ADC 转换完成输出，软件只读（不产生中断） 转换完成后此信号保持为高电平，只有关闭 ADC 才会清 0

位号	助记符	功能描述
0	ADC_IF	ADC 转换完成中断标志，硬件置位，软件写 1 清零，写 0 无效

30.10.5 ADC 配置寄存器(ADC_CFGR)

名称	ADC_CFGR							
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	-					ACC_PERIOD		
位权限	U-0					R/W-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RST_CTRL_DELAY							
位权限	R/W-0001 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		BUFEN	-	BUFSEL			
位权限	U-0		R/W-0	U-0	R/W-0000			

位号	助记符	功能描述
31:19	-	RFU：未实现，读为 0
18:16	ACC_PERIOD	外部累加器累加周期配置，单位 ADC_CLK 000：1023（对应结果 10bit） 001：2047（对应结果 11bit） 010：4095（对应结果 12bit） 011：8191（对应结果 13bit） 100：16383（对应结果 14bit） Others：4095
15:8	RST_CTRL_DELAY	SDADC 使能后 mode_ctrl 延迟长度配置，单位是 ADC_CLK 周期 00000000：不延迟 11111111：延迟 255 个 ADC_CLK *注意：复位值为 16 个时钟周期，不得修改为小于 16 的数值*
7:6	-	RFU：未实现，读为 0
5	BUFEN	ADC 输入通道 Buffer 使能
4	-	RFU：未实现，读为 0
3:0	BUFSEL	ADC 输入通道选择 0001：ADC_IN1 0010：ADC_IN2 0011：ADC_IN3 0100：ADC_IN4 0101：ADC_IN5 0110：ADC_IN6 0111：ADC_IN7 1000：ADC_IN8 1001：ADC_IN9 1010：ADC_IN10

位号	助记符	功能描述
		1011：ADC_IN11 1100：ADC_IN12 1110：VBAT 1111：TS

30.10.6 CIC 输出数据寄存器(CIC_DR)

名称	CIC_DR							
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	ADC_CIC_DATA[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ADC_CIC_DATA[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADC_CIC_DATA[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:24	-	RFU：未实现，读为 0
23:0	ADC_CIC_DATA	CIC 滤波器输出数据，更新频率由 ADC 工作时钟和 OSR 共同决定；注意这个数据是有符号数，格式为二进制补码。

30.10.7 CIC 输出偏移寄存器(CIC_OS)

名称	CIC_OS							
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	ADC_CIC_OS[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ADC_CIC_OS[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADC_CIC_OS[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
31:24	-	RFU：未实现，读为 0
23:0	ADC_CIC_OS	CIC 滤波器输出数据 offset 调整寄存器，软件写入 offset 值，硬件将 ADC_CIC_DR 加上 OS 后可以得到无符号结果

30.10.8 CIC 无符号输出结果寄存器(CIC_USDR)

名称	CIC_USDR							
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	ADC_CIC_USDR[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ADC_CIC_USDR[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADC_CIC_USDR[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:24	-	RFU：未实现，读为 0
23:0	ADC_CIC_USDR	ADC_CIC_DR + ADC_CIC_OS 得到的无符号数结果

30.10.9 CIC 滤波器控制寄存器(CIC_CR)

名称	CIC_CR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CIC_EN	-						
位权限		U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	NS_DISC							
位权限	R/W-0000 0100							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OVR_IE	CIC_IE	TRUNC			OSR		
位权限	R/W-0	R/W-0	R/W-000			R/W-110		

位号	助记符	功能描述
31	CIC_EN	CIC 滤波器使能 (CIC enable) 0：关闭 CIC 1：使能 CIC

位号	助记符	功能描述
30:16	-	RFU：未实现，读为 0
15:8	NS_DISC	CIC 使能后丢弃的输出点数 (Number of Samples to be Discarded) 可以丢弃 ADC 使能后开头输出的 0~255 个采样点 默认值 0x4，即丢弃前 4 个转换结果，从第 5 个结果开始产生 CIC_IF
7	OVR_IE	CIC Overrun Interrupt enable 0：禁止 CIC 溢出中断 1：允许 CIC 溢出中断
6	CIC_IE	CIC 中断使能 (CIC interrupt enable) 0：禁止 CIC 中断 1：允许 CIC 中断，当 CIC_IF 置位时产生中断事件
5:3	TRUNC	输出数据截取控制 (Truncate)，配置最终结果中丢弃最低位的位数； 比如 TRUNC=0x4，最终结果丢弃最低 4bit 后存入 ADC_CIC_DATA 寄存器
2:0	OSR	过采样率配置 (Over Sampling Rate) 000：x8 001：x16 010：x32 011：x64 100：x128 101：x256 110：x512 111：x1024

30.10.10 CIC 中断标志寄存器(CIC_ISR)

名称	CIC_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	-						CIC_OVR	CIC_IF
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	CIC_OVR	CIC 输出溢出标志，硬件置位，软件写 1 清零。 如果软件或 DMA 没有及时读取 ADC_CIC_DR，下一个转换结果

位号	助记符	功能描述
		将覆盖当前寄存器值，并置位 CIC_OVR 寄存器。
0	CIC_IF	CIC 输出更新中断，硬件置位，软件写 1 清零。 在使能了 DMA 通道后，CIC_IF 置位会触发 DMA 请求，DMA 读取 ADC_CIC_DR 后 CIC_IF 自动清零。

31 I/O 端口

31.1 概述

I/O 端口的主要功能特性：

- 部分 GPIO 引脚支持 3V 供电下的 5V tolerant
- 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

31.2 引脚类型

FM33A0xxEV 主要有多种类型的 GPIO 引脚，其中大部分引脚支持输入输出、数字外设功能、模拟外设通道、可控上拉电阻、可控开漏输出功能；强驱动引脚除了以上功能外，具有增强的推挽输出驱动能力；5V tolerant（FT）类型引脚可以承受高于电源电压的输入，而不会导致漏电现象。

31.2.1 GPIO，输入输出使能，可控上拉电阻，可控开漏输出

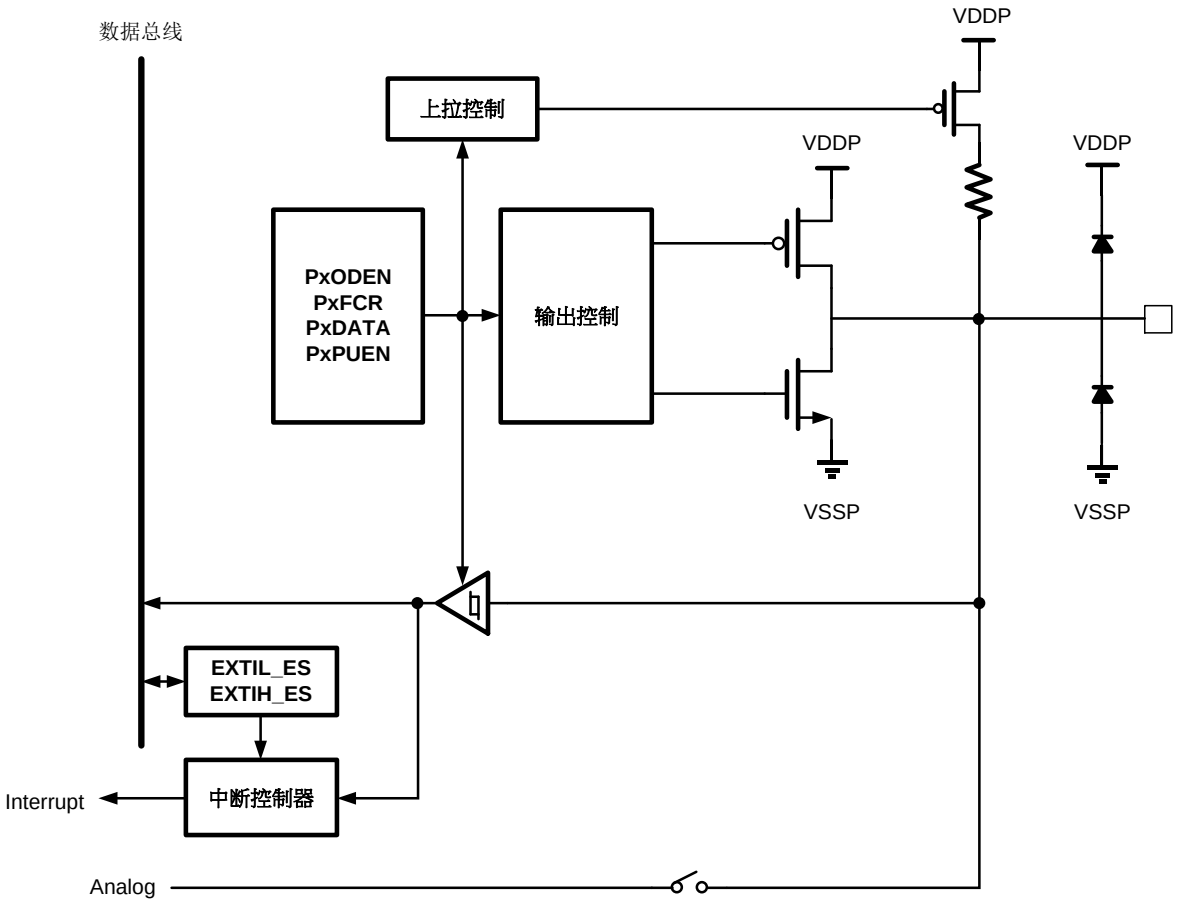


图 31-1 普通 GPIO 结构框图

控制逻辑定义如下：

Registers					PAD Interface		
FCR	INEN	ODEN	PUEN	DATA	INPUT_EN	OUTPUT_EN	PUEN
00	0	x	0/1	x	0	0	0/1
	1				1		
01	x	0	0/1	x	0	1	0/1
	x	1		0	0	1	
				1	0	0	
10	x	x	0/1	外设输入功能	1	0	0/1
	x	0		外设推挽输出功能	0	1	
	x	1		外设开漏输出 0	0	1	
				外设开漏输出 1	0	0	
11	x	x	x	x	0	0	0

表31-1GPIO功能逻辑定义表

31.2.2 GPIO，输入输出使能，2 个可控上拉电阻，可控开漏输出（仅 7816 数据口）

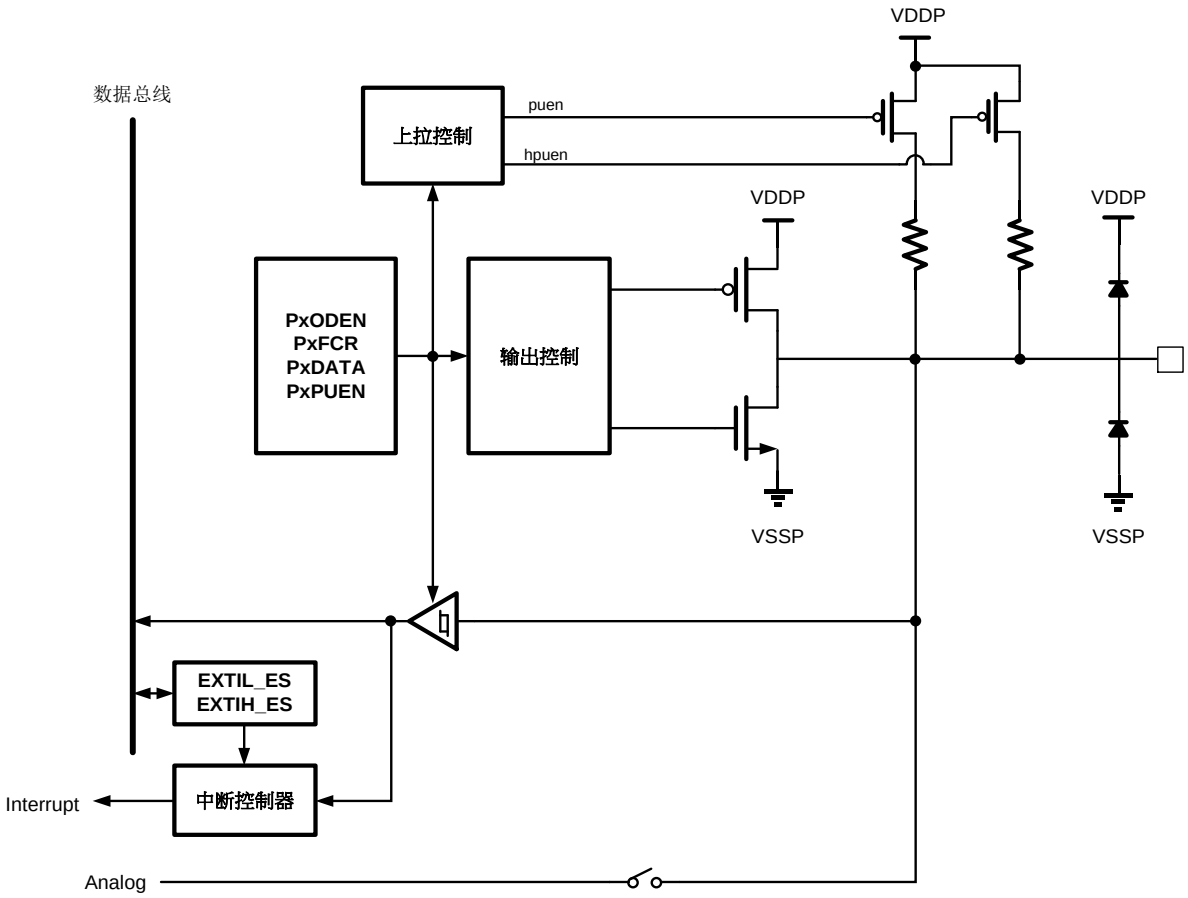


图 31-2 普通 GPIO（两路上拉）结构框图

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。

31.2.3 GPIO，输入输出使能，可控上拉电阻，可控开漏输出，5V tolerant

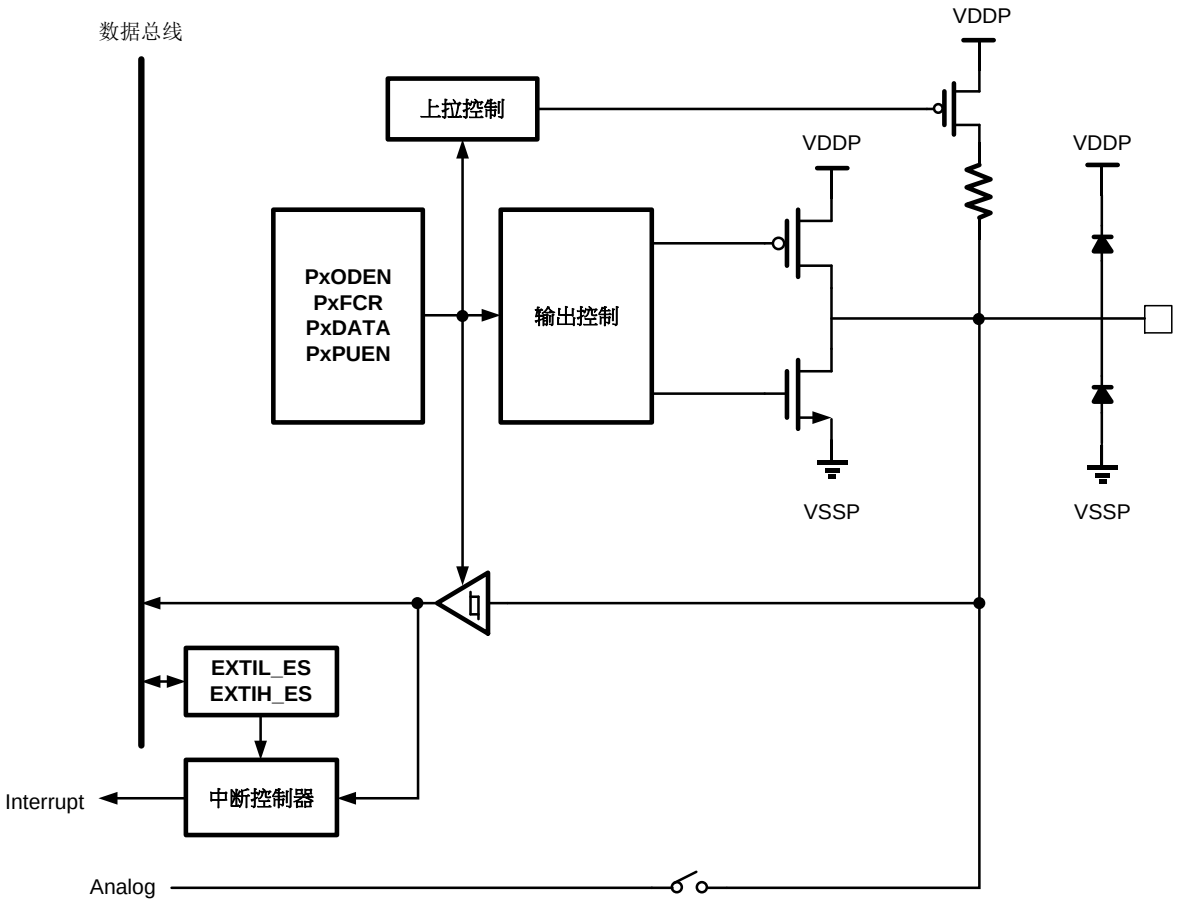


图 31-3 5V-tolerant GPIO 结构框图

此种GPIO可以承受高于电源电压的输入，比如电源为3V时，可以承受最高5.5V输入，而不会引起漏电。

控制逻辑定义如下：

Registers					PAD Interface		
FCR	INEN	ODEN	PUEN	DATA	INPUT_EN	OUTPUT_EN	PUEN
00	0	x	0/1	x	0	0	0/1
	1				1		
01	x	0	0/1	x	0	1	0/1
	x	1		0	0	1	
				1	0	0	
10	x	x	0/1	外设输入功能	1	0	0/1
	x	0		外设推挽输出功能	0	1	
	x	1		外设开漏输出 0	0	1	
				外设开漏	0	0	

				输出 1			
11	x	x	x	x	0	0	0

表31-2GPIO功能逻辑定义表

31.3 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31-3 FCR定义表

31.3.1 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

31.3.2 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

31.3.3 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
- IO 用于外设输出时，软件打开输入使能后，仍可以通过读取 PxDIN 寄存器获得 IO 上的实际电平状态

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（PxDFSx）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
PA8	UART5_RX	BT1_IN2	PADFS[8]
PA9	UART5_TX	BT2_IN2	PADFS[9]
PB1	UART1_TX	SVD_O	PBDFS[1]
PB6	QSPI_DQ2	BT1_OUT	PBDFS[6]
PB7	QSPI_DQ1	ET1_IN0	PBDFS[7]
PB8	QSPI_CS	ET2_IN0	PBDFS[8]
PB9	QSPI_DQ0	ET3_IN0	PBDFS[9]
PB10	QSPI_CLK	ET4_IN0	PBDFS[10]
PB11	QSPI_DQ3	ET1_OUT	PBDFS[11]
PB12	UART2_RX	SPI1_SSN	PBDFS[12]
PB13	UART2_TX	SPI1_SCK	PBDFS[13]
PB14	UART3_RX	SPI1_MISO	PBDFS[14]
PB15	UART3_TX	SPI1_MOSI	PBDFS[15]
PD6	ET3_IN1	SPI4_SCK	PDDFS[6]
PD7	ET4_IN1	SPI4_MISO	PDDFS[7]
PD8	ET3_OUT	COMP1_OUT	PDDFS[8]
PE2	ET1_IN1	FOUT1	PEDFS[2]
PE5	SCL1	SPI4_MOSI	PEDFS[5]
PE6	SDA1	SPI4_MISO	PEDFS[6]
PE7	ET2_IN1	SPI4_SCK	PEDFS[7]
PE8	LPT32_CH3	SPI4_SSN	PEDFS[8]
PF3	UART0_RX	LPUART0_RX	PFDFS[3]
PF4	UART0_TX	LPUART0_TX	PFDFS[4]
PG2	LPUART1_RX	U7816_CLK	PGDFS[2]
PG3	LPUART1_TX	U7816_IO	PGDFS[3]
PG4	LPUART1_RX	ET1_IN2	PGDFS[4]
PG5	LPUART1_TX	ET2_IN2	PGDFS[5]

GPIO	数字功能 1 PxDFS[x]=0	数字功能 2 PxDFS[x]=1	Additional AFSEL
PG6	SPI4_SSN	FOUT0	PGDFS[6]
PG7	SPI4_MOSI	ET4_OUT	PGDFS[7]
PG8	SWCLK	UART0_RX	PGDFS[8]
PG9	SWIO	UART0_TX	PGDFS[9]

表31-4多个数字外设功能选择表

31.3.4 数字外设功能优先级

当多个引脚分配了同一数字外设的同一功能时，其优先级定义如下：

外设功能		高优先级->低优先级	
U7816IO		PG3	PC3
LPUART0_RX	IOSWAP=0	PF3	PC0
	IOSWAP=1	PF4	PC1
LPUART1_RX	IOSWAP=0	PG4	PG2
	IOSWAP=1	PG5	PG3
QSPI_DQ1		PF6	PB7
QSPI_DQ2		PF5	PB6
UART5_RX	IOSWAP=0	PC4	PA8
	IOSWAP=1	PC5	PA9
UART4_RX	IOSWAP=0	PD9	PD0
	IOSWAP=1	PD10	PD1
UART3_RX	IOSWAP=0	PC10	PB14
	IOSWAP=1	PC11	PB15
UART2_RX	IOSWAP=0	PB12	PB2
	IOSWAP=1	PB13	PB3
UART1_RX	IOSWAP=0	PE3	PB0
	IOSWAP=1	PE4	PB1
UART0_RX	IOSWAP=0	PG8	PF3
	IOSWAP=1	PG9	PF4
SPI0_MOSI	IOSWAP=0	PF12	-
	IOSWAP=1	PF13	-
SPI0_MISO	IOSWAP=0	PF13	-
	IOSWAP=1	PF12	-
SPI1_MOSI	IOSWAP=0	PB15	PA7
	IOSWAP=1	PB14	PA6
SPI1_MISO	IOSWAP=0	PB14	PA6
	IOSWAP=1	PB15	PA7
SPI1_SCK		PB13	PA5
SPI1_SSN		PB12	PA4
SPI2_MOSI	IOSWAP=0	PD5	PA3
	IOSWAP=1	PD4	PA2
SPI2_MISO	IOSWAP=0	PD4	PA2
	IOSWAP=1	PD5	PA3
SPI2_SCK		PD3	PA1
SPI2_SSN		PD2	PA0
SPI3_MOSI	IOSWAP=0	PC9	-
	IOSWAP=1	PC8	-
SPI3_MISO	IOSWAP=0	PC8	-
	IOSWAP=1	PC9	-

SPI4_MISO	IOSWAP=0	PE6	PD7
	IOSWAP=1	PG7	PE5
SPI4_MOSI	IOSWAP=0	PG7	PE5
	IOSWAP=1	PE6	PD7
SPI4_SCK		PE7	PD6
SPI4_SSN		PG6	PE8

表31-5相同数字外设功能引脚优先级

注：引脚对应的数字外设功能，还会受到 PxDFS 控制，具体参照寄存器章节 PxDFS 控制寄存器描述

31.3.5 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 LCD 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：

GPIO	模拟功能	IO 模拟开关配置寄存器	说明
PB2	ADC_IN7	PB2ANEN	ADC 需要采样这些通道的输入信号时，必须将对应通道的 IO 模拟开关打开
PB3	ADC_IN8	PB3ANEN	
PC12	ADC_IN1	PC12ANEN	
PC13	ADC_IN2	PC13ANEN	
PC15	ADC_IN6	PC15ANEN	
PD0	ADC_IN3	PD0ANEN	
PD1	ADC_IN4	PD1ANEN	
PF6	ADC_IN5	PF6ANEN	
PC15	COMP1_INP4	PC15ANEN	COMP1/2 工作时必须将对应模拟通道的 IO 模拟
PF6	COMP1_INP1	PF6ANEN	
PC14	COMP1_INN2	PC14ANEN	
PG2	COMP1_INP2	PG2ANEN	
PG3	COMP1_INP3	PG3ANEN	
PE3	COMP2_INN2	PE3ANEN	

PE4	COMP2_INP2	PE4ANEN	开关打开
PB0	COMP2_INN1	PB0ANEN	
PB1	COMP2_INP1	PB1ANEN	
PF5	COMP1_INN1	PF5ANEN	
PD6	REFIN	PD6ANEN	
PE7	VCIN1	PE7ANEN	LCD
PE8	VCIN2	PE8 ANEN	
PF12	VDISP0	PF12ANEN	
PF13	VDISP1	PF13ANEN	
PF14	VDISP2	PF14ANEN	
PF15	VDISP3	PF15ANEN	
PF11	SVS	PF11ANEN	SVD
PD7	AUX	PD7ANEN	BUF4TEST

表31-6模拟信号通道

31.3.6 使用外部晶体引脚

FM33A0xxEV 支持外接 32768Hz 晶体和 4~32MHz 高频晶体。

其中 PF1 和 PF2 默认为 GPIO，配置为模拟功能之后可以作为 XTHIN 和 XTHOUT 外接高频晶体。

32768Hz 低频晶振引脚为专用引脚，不与 GPIO 复用。当需要从外部输入 32K 时钟时，也可以不外接晶体，直接从 XTALIN 引脚灌入时钟；此时可以将芯片配置为无晶体模式，关闭内部 XTLF 偏置电路以降低功耗。

31.3.7 开漏输出

所有 GPIO 都可以配置为开漏输出。ODEN 使能后，PMOS 输出驱动管被关闭，GPIO 无法输出 1，只能输出 0。

与此同时，如果关闭内部上拉，通过外部上拉电阻，由于 GPIO 都是 5V tolerant，因此可以实现当芯片采用 3V 供电时，仍可以与外部 5V 系统通信，而没有漏电风险。

如下图所示：

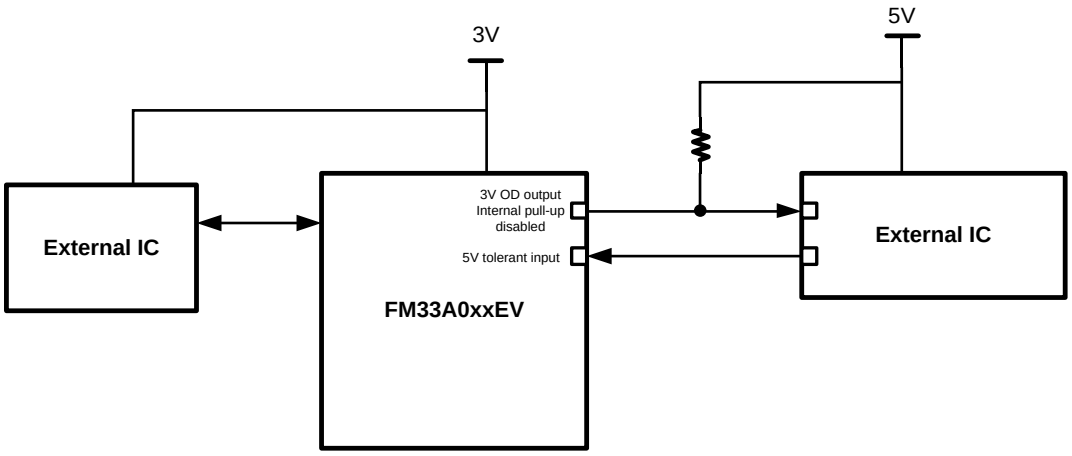


图 31-4 与不同电源供电芯片的连接

31.4 SWD 引脚

SWD 引脚 PG8、PG9 在芯片上电后，默认为输入态，并且为 SWD 功能，其中 PG9（SWIO）默认上拉使能，PG8（SWCLK）默认不使能上拉，建议外接上拉电阻。

31.5 NRST 引脚

NRST 引脚用于产生芯片复位，如果芯片处于低功耗模式，NRST 有效也会使芯片退出低功耗模式。

NRST 引脚输入有数字滤波，滤波采用 LSCLK 分频时钟工作。有晶体模式下，LSCLK 不会停止，因此引脚滤波总是有效的。
无晶体模式下，LSCLK 仅来自于 RCLP，如果软件关闭了 RCLP，则引脚滤波无效，NRST 无法工作。因此除了测试目的之外，一般不建议无晶体模式下软件关闭 RCLP。

2 个引脚数字滤波的长度都是固定 8ms。

31.6 WKUPx 引脚

FM33A0xxEV 有 8 个 WKUP 引脚，能够将芯片从 Sleep/DeepSleep 模式下唤醒，即使片上振荡器都停止工作，WKUP 仍能唤醒芯片。

WKUPx 引脚输入上升沿、下降沿或上升下降沿（软件配置）能够将芯片从休眠模式下唤醒。为了使能此功能，需将对应引脚配置为 GPIO 输入功能，并且相应的 PINWKEN 置位，注意 PAD 内部带有上拉电阻，如果配置为上升沿唤醒，则必须关闭上拉电阻。

每个支持 WKUP 功能的 IO 都带有大约 100ns 的片内模拟滤波，能够滤除输入信号上的毛刺，避免误触发。

Sleep/DeepSleep 模式下，使能了的 WKUPx 引脚上任何大于 100ns 的脉冲都会触发芯片唤醒。

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。

31.7 外部引脚中断（EXTI）

31.7.1 功能说明

FM33A0xxEV 的 7 组 GPIO（PA~PG）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。

使用引脚中断前必须将对应引脚的功能设为 GPIO 输入。各路中断使能与 edge select 整合在一起，edge select 可以选择上升沿、下降沿、双沿触发中断，或禁止中断触发。当 edge select 寄存器配置为禁止中断时，输入采样寄存器时钟将自动关闭。

每个引脚中断输入都可以选择使能或关闭数字滤波功能。

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~PD15(全无)	EXTI_DSEL[7:6]	EXTI[15]
PE2~PE3(无PE0,1)	EXTI_ESEL[1:0]	EXTI[16]
PE4~PE7	EXTI_ESEL[3:2]	EXTI[17]
PE8~PE9(无PE10,11)	EXTI_ESEL[5:4]	EXTI[18]
PE12~PE15(全无)	-	EXTI[19] (RFU)
PF0~PF3	EXTI_FSEL[1:0]	EXTI[20]
PF4~PF6(无PF7)	EXTI_FSEL[3:2]	EXTI[21]
PF11(无PF8,9,10)	EXTI_FSEL[5:4]	EXTI[22]
PF12~PF15(全无)	-	EXTI[23] (RFU)

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。

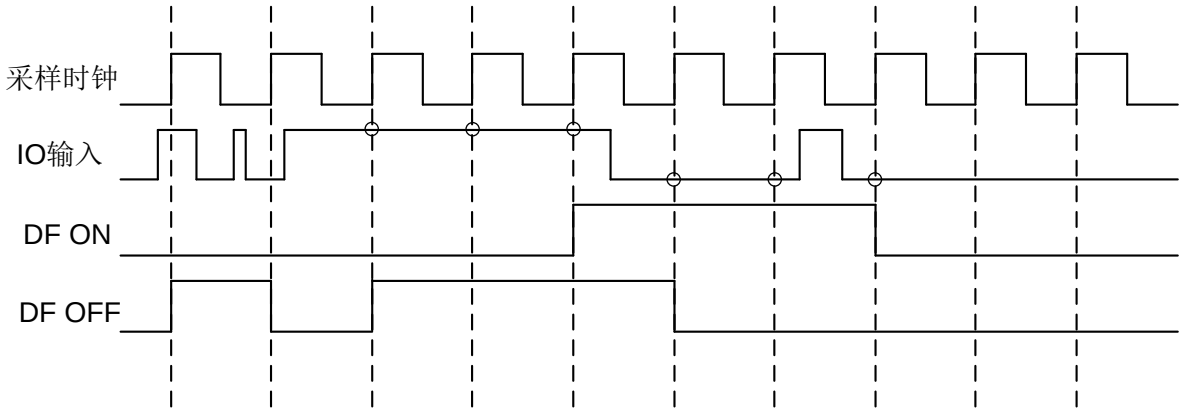


图 31-5 引脚输入数字滤波

软件可以选择数字滤波的采样时钟为 APBCLK 或者 LSCLK。

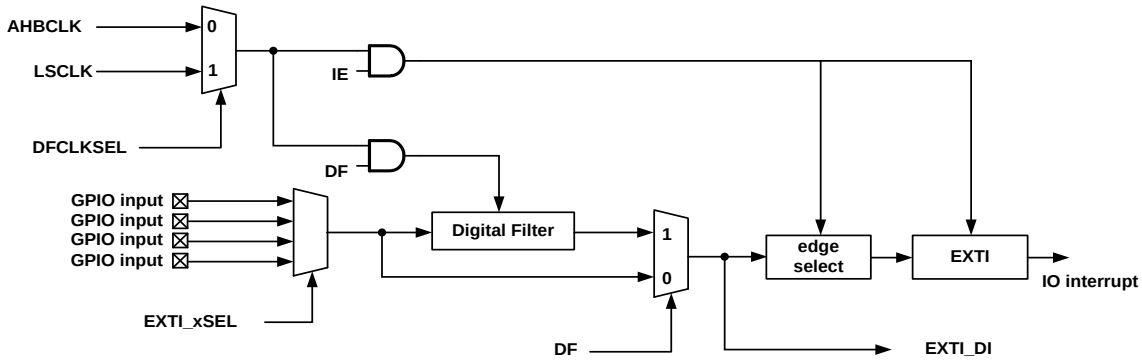


图 31-6 EXTI 信号输入示意图

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。

31.7.2 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

31.8 快速 GPIO 输出

FM33A0xxEV 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。

31.9 IO MUX 输出

支持 4 个引脚（PD6,PD7,PG2,PG3）输入信号经过 MUX 后直接从 1 个引脚（PG7）输出。

使用此功能时，将（PD6,PD7,PG2,PG3）引脚配置为 GPIO 输入，将 PG7 配置为 GPIO 输出，并且配置 IOMUX_SEL 和 IOMUX_EN 寄存器。

31.10 寄存器

模块起始地址：0x4000_0C00

offset 地址	名称	符号
GPIOA(模块起始地址: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_DIN
0x20	GPIOA 额外数字功能寄存器 （GPIOA Digital Function Select）	GPIOA_DFS
-	-	-
0x28	GPIOA 模拟开关使能寄存器 （GPIOAAnalog channel Enable Register）	GPIOA_ANEN
GPIOB(模块起始地址:0X40000C40)		
0x00	GPIOB 输入使能寄存器 （GPIOB Input Enable Register）	GPIOB_INEN
0x04	GPIOB 上拉使能寄存器 （GPIOB Pull-Up Enable Register）	GPIOB_PUEN
0x08	GPIOB 开漏使能寄存器 （GPIOB Opeb-Drain Enable Register）	GPIOB_ODEN
0x0C	GPIOB 功能选择寄存器 （GPIOB Function Control Register）	GPIOB_FCR
0x10	GPIOB 输出数据寄存器 （GPIOB Data Output Register）	GPIOB_DO
0x14	GPIOB 输出数据置位寄存器 （GPIOB Data Set Register）	GPIOB_DSET
0x18	GPIOB 输出数据复位寄存器 （GPIOB Data Reset Register）	GPIOB_DRST
0x1C	GPIOB 输入数据寄存器 （GPIOBData Input Register）	GPIOB_DIN
0x20	GPIOB 额外数字功能寄存器 （GPIOB Digital Function Select）	GPIOB_DFS
-	-	-
0x28	GPIOB 模拟开关使能寄存器 （GPIOB Analog channel Enable Register）	GPIOB_ANEN
GPIOC(模块起始地址:0X40000C80)		
0x00	GPIOC 输入使能寄存器	GPIOC_INEN

offset 地址	名称	符号
	（GPIOC Input Enable Register）	
0x04	GPIOC 上拉使能寄存器 （GPIOC Pull-Up Enable Register）	GPIOC_PUEN
0x08	GPIOC 开漏使能寄存器 （GPIOC Opeb-Drain Enable Register）	GPIOC_ODEN
0x0C	GPIOC 功能选择寄存器 （GPIOC Function Control Register）	GPIOC_FCR
0x10	GPIOC 输出数据寄存器 （GPIOC Data Output Register）	GPIOC_DO
0x14	GPIOC 输出数据置位寄存器 （GPIOC Data Set Register）	GPIOC_DSET
0x18	GPIOC 输出数据复位寄存器 （GPIOC Data Reset Register）	GPIOC_DRST
0x1C	GPIOC 输入数据寄存器 （GPIOC Data Input Register）	GPIOC_DIN
0x20	GPIOC 额外数字功能寄存器 （GPIOC Digital Function Select）	GPIOC_DFS
-	-	-
0x28	GPIOC 模拟开关使能寄存器 （GPIOC Analog channel Enable Register）	GPIOC_ANEN
GPIOD(模块起始地址:0X40000CC0)		
0x00	GPIOD 输入使能寄存器 （GPIOD Input Enable Register）	GPIOD_INEN
0x04	GPIOD 上拉使能寄存器 （GPIOD Pull-Up Enable Register）	GPIOD_PUEN
0x08	GPIOD 开漏使能寄存器 （GPIOD Opeb-Drain Enable Register）	GPIOD_ODEN
0x0C	GPIOD 功能选择寄存器 （GPIOD Function Control Register）	GPIOD_FCR
0x10	GPIOD 输出数据寄存器 （GPIOD Data Output Register）	GPIOD_DO
0x14	GPIOD 输出数据置位寄存器 （GPIOD Data Set Register）	GPIOD_DSET
0x18	GPIOD 输出数据复位寄存器 （GPIOD Data Reset Register）	GPIOD_DRST
0x1C	GPIOD 输入数据寄存器 （GPIOD Data Input Register）	GPIOD_DIN
0x20	GPIOD 额外数字功能寄存器 （GPIOD Digital Function Select）	GPIOD_DFS
-	-	-
0x28	GPIOD 模拟开关使能寄存器 （GPIOD Analog channel Enable Register）	GPIOD_ANEN
GPIOE(模块起始地址:0X40000D00)		
0x00	GPIOE 输入使能寄存器 （GPIOE Input Enable Register）	GPIOE_INEN
0x04	GPIOE 上拉使能寄存器 （GPIOE Pull-Up Enable Register）	GPIOE_PUEN
0x08	GPIOE 开漏使能寄存器 （GPIOE Opeb-Drain Enable Register）	GPIOE_ODEN
0x0C	GPIOE 功能选择寄存器 （GPIOE Function Control Register）	GPIOE_FCR

offset 地址	名称	符号
0x10	GPIOE 输出数据寄存器 （GPIOE Data Output Register）	GPIOE_DO
0x14	GPIOE 输出数据置位寄存器 （GPIOE Data Set Register）	GPIOE_DSET
0x18	GPIOE 输出数据复位寄存器 （GPIOE Data Reset Register）	GPIOE_DRST
0x1C	GPIOE 输入数据寄存器 （GPIOE Data Input Register）	GPIOE_DIN
0x20	GPIOE 额外数字功能寄存器 （GPIOE Digital Function Select）	GPIOE_DFS
-	-	-
0x28	GPIOE 模拟开关使能寄存器 （GPIOE Analog channel Enable Register）	GPIOE_ANEN
GPIOF(模块起始地址:0X40000D40)		
0x00	GPIOF 输入使能寄存器 （GPIOF Input Enable Register）	GPIOF_INEN
0x04	GPIOF 上拉使能寄存器 （GPIOF Pull-Up Enable Register）	GPIOF_PUEN
0x08	GPIOF 开漏使能寄存器 （GPIOF Opeb-Drain Enable Register）	GPIOF_ODEN
0x0C	GPIOF 功能选择寄存器 （GPIOF Function Control Register）	GPIOF_FCR
0x10	GPIOF 输出数据寄存器 （GPIOF Data Output Register）	GPIOF_DO
0x14	GPIOF 输出数据置位寄存器 （GPIOF Data Set Register）	GPIOF_DSET
0x18	GPIOF 输出数据复位寄存器 （GPIOF Data Reset Register）	GPIOF_DRST
0x1C	GPIOF 输入数据寄存器 （GPIOF Data Input Register）	GPIOF_DIN
0x20	GPIOF 额外数字功能寄存器 （GPIOF Digital Function Select）	GPIOF_DFS
-	-	-
0x28	GPIOF 模拟开关使能寄存器 （GPIOF Analog channel Enable Register）	GPIOF_ANEN
GPIOG(模块起始地址:0X40000D80)		
0x00	GPIOG 输入使能寄存器 （GPIOG Input Enable Register）	GPIOG_INEN
0x04	GPIOG 上拉使能寄存器 （GPIOG Pull-Up Enable Register）	GPIOG_PUEN
0x08	GPIOG 开漏使能寄存器 （GPIOG Opeb-Drain Enable Register）	GPIOG_ODEN
0x0C	GPIOG 功能选择寄存器 （GPIOG Function Control Register）	GPIOG_FCR
0x10	GPIOG 输出数据寄存器 （GPIOG Data Output Register）	GPIOG_DO
0x14	GPIOG 输出数据置位寄存器 （GPIOG Data Set Register）	GPIOG_DSET
0x18	GPIOG 输出数据复位寄存器 （GPIOG Data Reset Register）	GPIOG_DRST
0x1C	GPIOG 输入数据寄存器	GPIOG_DIN

offset 地址	名称	符号
	（GPIOG Data Input Register）	
0x20	GPIOG 额外数字功能寄存器 （GPIOG Digital Function Select）	GPIOG_DFS
-	-	-
0x28	GPIOG 模拟开关使能寄存器 （GPIOG Analog channel Enable Register）	GPIOG_ANEN
GPIO (模块起始地址:0X40000DC0)		
0x00	EXTI 输入选择寄存器 0 （External Interrupt input Select Register0）	GPIO_EXTISEL0
0x04	EXTI 输入选择寄存器 1 （External Interrupt input Select Register1）	GPIO_EXTISEL1
0x08	EXTI 边沿选择和使能寄存器 0 （External Interrupt Edge Select and Enable Register0）	GPIO_EXTIEDS0
0x0C	EXTI 边沿选择和使能寄存器 1 （External Interrupt Edge Select and Enable Register1）	GPIO_EXTIEDS1
0x10	EXTI 数字滤波控制寄存器 （External Interrupt Digital Filter Register）	GPIO_EXTIDF
0x14	EXTI 中断标志寄存器 （External Interrupt and Status Register）	GPIO_EXTIISR
0x18	EXTI 输入信号寄存器 （External Interrupt Data Input Register）	GPIO_EXTIDI
0x40	FOUT 配置寄存器 （Frequency Output Select Register）	GPIO_FOUTSEL
0x44	IO MUX 控制寄存器 （IO MUX Control Register）	GPIO_IOMCR
0x140	WKUP 使能寄存器 （Wakeup Enable Register）	GPIO_PINWKEN

31.10.1 GPIO 输入使能寄存器（GPIOx_INEN）

名称	GPIOx_INEN(x=A,B,C,D,E,F,G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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	PxINEN[15:8]							
位权限	R/W-0000 0000							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxINEN[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	PxINEN	GPIO 输入使能控制 0：关闭输入使能 1：打开输入使能

31.10.2 GPIO 上拉使能寄存器（GPIOx_PUEN）

名称	GPIOx_PUEN(x=A,B,C,D,E,F,G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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	PxPUEN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxPUEN[7:0]							
位权限	R/W-0000 0000							

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

31.10.3 GPIO 开漏使能寄存器（GPIOx_ODEN）

名称	GPIOx_ODEN(x=A,B,C,D,E,F,G)

offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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	PxODEN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxODEN[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	PxODEN	GPIO 开漏输出使能 0：关闭开漏输出 1：使能开漏输出

31.10.4 GPIO 功能选择寄存器（GPIOx_FCR）

名称	GPIOx_FCR(x=A,B,C,D,E,F,G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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 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

位号	助记符	功能描述
		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

31.10.5 GPIO 输出数据寄存器（GPIOx_DO）

名称	GPIOx_DO(x=A,B,C,D,E,F,G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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	PxDO[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxDO[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	PxDO	GPIO output data register

31.10.6 GPIO 输出数据置位寄存器（GPIOx_DSET）

名称	GPIOx_DSET(x=A,B,C,D,E,F,G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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	PxSET[15:8]							
位权限	W							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxSET[7:0]							
位权限	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 举例：向 PADSET 写 0x0000_8000，则 PADO[15]置位，其余位保持不变。

31.10.7 GPIO 输出数据复位寄存器（GPIOx_DRST）

名称	GPIOx_DRST(x=A,B,C,D,E,F,G)

offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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	PxRESET[15:8]							
位权限	W							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxRESET[7:0]							
位权限	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

31.10.8 GPIO 输入数据寄存器（GPIOx_DIN）

名称	GPIOx_DIN(x=A,B,C,D,E,F,G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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	PxDIN[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxDIN[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	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

31.10.9 GPIO 额外数字功能选择寄存器（GPIOx_DFS）

名称	GPIOx_DFS(x=A,B,C,D,E,F,G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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 32-4

31.10.10 GPIO 模拟开关使能寄存器（GPIOx_ANEN）

名称	GPIOx_ANEN(x=A,B,C,D,E,F,G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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 详细请参考表 31-7

31.10.1 EXTI 输入选择寄存器 0（GPIO_EXTISEL0）

名称	GPIO_EXTISEL0							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTI_DSEL							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EXTI_CSEL							
位权限	R/W-0000 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	PortD EXTI 中断输入选择 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

位号	助记符	功能描述
		01：PD1 10：PD2 11：PD3
23:16	EXTI_CSEL	PortC EXTI 中断输入选择 EXTI_CSEL[7:6] – 00：PC12 01：PC13 10：PC14 11：PC15 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	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

位号	助记符	功能描述
		11：PA15 EXTI_ASEL[5:4] – 00：PA8 01：PA9 10：PA10 11：PA11 EXTI_ASEL[3:2] – 00：PA4 01：PA5 10：PA6 11：PA7 EXTI_ASEL[1:0] – 00：PA0 01：PA1 10：PA2 11：PA3

31.10.2 EXTI 输入选择寄存器 1（GPIO_EXTISEL1）

名称	GPIO_EXTISEL1							
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	EXTI_FSEL							
位权限	R/W-0000 0000							
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
31:16	-	RFU：未实现，读为 0
15:8	EXTI_FSEL	PortF EXTI 中断输入选择 EXTI_FSEL[7:6] –RFU EXTI_FSEL[5:4] – 00：RFU 01：RFU 10：RFU 11：PF11 EXTI_FSEL[3:2] – 00：PF4 01：PF5 10：PF6 11：RFU

位号	助记符	功能描述
		EXTI_FSEL[1:0] – 00：PF0 01：PF1 10：PF2 11：PF3
7:0	EXTI_ESEL	PortE EXTI 中断输入选择 EXTI_ESEL[7:6] – RFU EXTI_ESEL[5:4] – 00：PE8 01：PE9 10：RFU 11：RFU EXTI_ESEL[3:2] – 00：PE4 01：PE5 10：PE6 11：PE7 EXTI_ESEL[1:0] – 00：RFU 01：RFU 10：PE2 11：PE3

31.10.3 EXTI 边沿选择和使能寄存器 0（GPIO_EXTIEDS0）

名称	GPIO_EXTIEDS0							
offset	0x08							
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
29:28	EXTI14_EDS	EXTI[14]边缘触发选择 00：rising

位号	助记符	功能描述
		01：falling 10：both 11：disable
27:26	EXTI13_EDS	EXTI[13]边缘触发选择 00：rising 01：falling 10：both 11：disable
25:24	EXTI12_EDS	EXTI[12]边缘触发选择 00：rising 01：falling 10：both 11：disable
23:22	EXTI11_EDS	EXTI[11]边缘触发选择 00：rising 01：falling 10：both 11：disable
21:20	EXTI10_EDS	EXTI[10]边缘触发选择 00：rising 01：falling 10：both 11：disable
19:18	EXTI9_EDS	EXTI[9]边缘触发选择 00：rising 01：falling 10：both 11：disable
17:16	EXTI8_EDS	EXTI[8]边缘触发选择 00：rising 01：falling 10：both 11：disable
15:14	EXTI7_EDS	EXTI[7]边缘触发选择 00：rising 01：falling 10：both 11：disable
13:12	EXTI6_EDS	EXTI[6]边缘触发选择 00：rising 01：falling 10：both 11：disable
11:10	EXTI5_EDS	EXTI[5]边缘触发选择 00：rising 01：falling 10：both 11：disable
9:8	EXTI4_EDS	EXTI[4]边缘触发选择

位号	助记符	功能描述
		00：rising 01：falling 10：both 11：disable
7:6	EXTI3_EDS	EXTI[3]边缘触发选择 00：rising 01：falling 10：both 11：disable
5:4	EXTI2_EDS	EXTI[2]边缘触发选择 00：rising 01：falling 10：both 11：disable
3:2	EXTI1_EDS	EXTI[1]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

31.10.4 EXTI 边沿选择和使能寄存器 1（GPIO_EXTIEDS1）

名称	GPIO_EXTIEDS1							
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	EXTI23_EDS		EXTI22_EDS		EXTI21_EDS		EXTI20_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	EXTI19_EDS		EXTI18_EDS		EXTI17_EDS		EXTI16_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
31:16	-	RFU：未实现，读为 0
15:14	EXTI23_EDS	EXTI[23]边缘触发选择 00：rising 01：falling

位号	助记符	功能描述
		10：both 11：disable
13:12	EXTI22_EDS	EXTI[22]边缘触发选择 00：rising 01：falling 10：both 11：disable
11:10	EXTI21_EDS	EXTI[21]边缘触发选择 00：rising 01：falling 10：both 11：disable
9:8	EXTI20_EDS	EXTI[20]边缘触发选择 00：rising 01：falling 10：both 11：disable
7:6	EXTI19_EDS	EXTI[19]边缘触发选择 00：rising 01：falling 10：both 11：disable
5:4	EXTI18_EDS	EXTI[18]边缘触发选择 00：rising 01：falling 10：both 11：disable
3:2	EXTI17_EDS	EXTI[17]边缘触发选择 00：rising 01：falling 10：both 11：disable
1:0	EXTI16_EDS	EXTI1[16] 边缘触发选择 00：rising 01：falling 10：both 11：disable

31.10.5 EXTI 数字滤波控制寄存器（GPIO_EXTIDF）

名称	GPIO_EXTIDF							
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	EXTI_DF[23:16]							
位权限	R/W -0000 0000							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_DF[15:8]							
位权限	R/W -0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI_DF[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:24	-	RFU：未实现，读为 0
23:0	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

31.10.6 EXTI 中断标志寄存器（GPIO_EXTIISR）

名称	GPIO_EXTIISR							
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	EXTI[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:24	-	RFU：未实现，读为 0
23:0	EXTI	外部引脚中断标志寄存器，共可以产生 24 个引脚中断 硬件置位，软件写 1 清零

31.10.7 EXTI 输入信号寄存器（GPIO_EXTIDI）

名称	GPIO_EXTIDI							
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	EXTI_DI[23:16]							
位权限	R-0000 0000							
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
31:24	-	RFU：未实现，读为 0
23:0	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 24 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。*

31.10.8 FOUT 配置寄存器（GPIO_FOUTSEL）

名称	GPIO_FOUTSEL							
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	-			FOUT1SEL				
位权限	U-0			R/W-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			FOUT0SEL				
位权限	U-0			R/W-00000				

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12:8	FOUT1SEL	PE2 输出选择 00000：XTLF 00001：RCLP 00010：RCHF/64 00011：LSCLK 00100：AHBCLK/64 00101：RTC_TM 00110：PLL_H/64 00111：RTCCLK64Hz 01000：APBCLK/64 01001：PLL_L 01010：RCMF 01011：RCHF 01100：XTHF/64 01101：ADCCLK 01110：PLL_H 01111：Reserved 10000：Boost Flag

位号	助记符	功能描述
		10001：Autotrim output 10010：Reserved 10011：Reserved
7:5	-	RFU：未实现，读为 0
4:0	FOUT0SEL	PG6 输出选择 0000：XTLF 0001：RCLP 0010：RCHF/64 0011：LSCLK 0100：AHBCLK/64 0101：RTC_TM 0110：PLL_H/64 0111：RTCCLK64Hz 1000：APBCLK/64 1001：PLL_L 1010：RCMF 1011：RCHF 1100：XTHF/64 1101：ADCCLK 1110：PLL_H 1111：Reserved 10000：Boost Flag 10001：Reserved 10010：Reserved 10011：Reserved

31.10.9 IO MUX 寄存器（GPIO_IOMCR）

名称	GPIO_IOMCR							
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	-					IOMUX_SEL		IOMUX_EN
位权限	U-0					R/W-00		R/W-0

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2:1	IOMUX_SEL	IO MUX 选择 00：PD6

位号	助记符	功能描述
		01：PD7 10：PG2 11：PG3
0	IOMUX_EN	IO MUX 使能 0：关闭 IO MUX 功能 1：使能 IO MUX 功能

31.10.10 WKUP 使能寄存器（GPIO_PINWKEN）

名称	GPIO_PINWKEN							
offset	0x140							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	WKISEL	-						
位权限	R/W-0	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PINWKSEL[23:16]							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PINWKSEL[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PINWKEN							
位权限	R/W-00000000							

位号	助记符	功能描述
31	WKISEL	WKUPx 中断入口选择 0：NMI 中断 1：#38 入口
30:24	-	RFU：未实现，读为 0
23:8	PINWKSEL	WKUPx 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、下降沿唤醒 Bit[9:8]对应 WKUP0 Bit[11:9]对应 WKUP1 Bit[13:12]对应 WKUP2 Bit[15:14]对应 WKUP3 Bit[17:16]对应 WKUP4 Bit[19:18]对应 WKUP5 Bit[21:20]对应 WKUP6 Bit[23:22]对应 WKUP7
7:0	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

31.11 GPIOH 寄存器

模块起始地址：0x4001_FC00

offset 地址	名称	符号
GPIOH(模块起始地址:0x4001FC00)		
0x00	GPIOH 输入使能寄存器 （GPIOH Input Enable Register）	GPIOH_INEN
0x04	GPIOH 上拉使能寄存器 （GPIOH Pull-Up Enable Register）	GPIOH_PUEN
0x08	GPIOH 功能选择寄存器 （GPIOH Function Control Register）	GPIOH_FCR
0x0C	GPIOH 输出数据寄存器 （GPIOH Data Output Register）	GPIOH_DO
0x10	GPIOH 输入数据寄存器 （GPIOH Data Input Register）	GPIOH_DIN

31.11.1 GPIO 输入使能寄存器（GPIOH_INEN）

名称	GPIOH_INEN							
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	-				PHINEN			
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	PHINEN	GPIOH 输入使能控制 0：关闭输入使能 1：打开输入使能

31.11.2 GPIO 上拉使能寄存器（GPIOH_PUEN）

名称	GPIOH_PUEN							
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	-				PHPUEN			
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	PHPUEN	GPIOH 上拉控制 0：关闭上拉 1：使能上拉

31.11.3 GPIO 功能选择寄存器（GPIOH_FCR）

名称	GPIOH_FCR							
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	Px3FCR		Px2FCR		Px1FCR		Px0FCR	
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	Px3FCR	PH[3]引脚功能选择 00：GPIO 输入 01：GPIO 输出 11：Analog function (ADC_IN12)
5:4	Px2FCR	PH[2]引脚功能选择 00：GPIO 输入 01：GPIO 输出 11：Analog function (ADC_IN11)
3:2	Px1FCR	PH[1]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function (RTCTM) 11：Analog function (ADC_IN10)
1:0	Px0FCR	PH[0]引脚功能选择

位号	助记符	功能描述
		00：GPIO 输入 01：GPIO 输出 10：Digital function (RFU) 11：Analog function (ADC_IN9)

31.11.4 GPIO 输出数据寄存器（GPIOH_DO）

名称	GPIOH_DO							
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	-				PHDO			
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	PHDO	GPIO output data register

31.11.5 GPIO 输入数据寄存器（GPIOH_DIN）

名称	GPIOH_DIN							
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	-				PHDIN			
位权限	U-0				R-0000			

位号	助记符	功能描述
31:4	-	RFU：未实现，读为 0
3:0	PHDIN	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

位号	助记符	功能描述
		返回引脚输入信号，芯片并不对引脚输入进行锁存

32 专用编程接口

32.1 概述

FM33A0xxEV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。

32.2 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。

33 调试支持

33.1 概述

FM33A0xxEV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FM33A0xxEV芯片互联，并实现仿真调试。

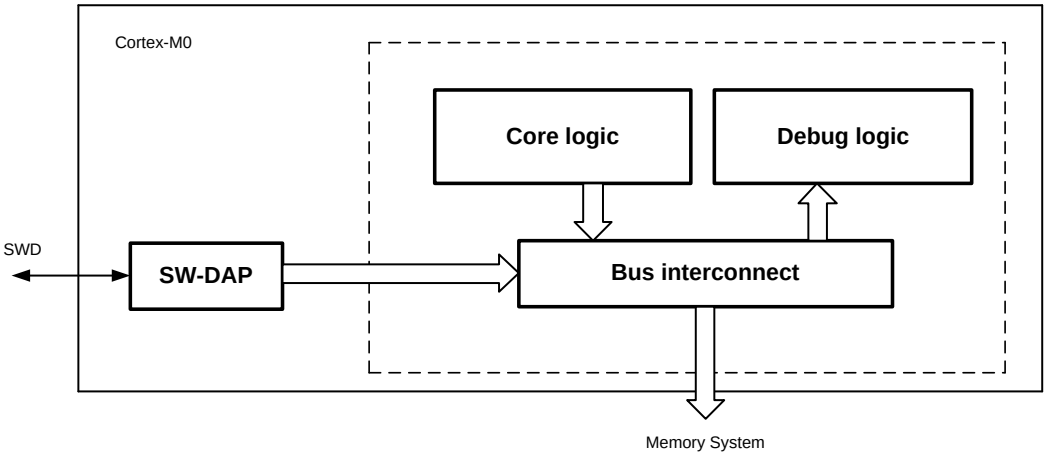


图 33-1 Cortex-M0 调试系统示意图

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。

33.2 Debug 引脚

33.2.1 SWD 引脚

FM33A0xxEV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	PG9
SWCLK	SWD时钟输入	PG8

注意：芯片复位后PG8和PG9都默认为输入状态，与大部分GPIO不同。

33.2.2 上拉电阻

芯片复位后，SWDIO和SWDCLK引脚默认使能内部上拉（约100K欧姆），用户无需在PCB上外接

上拉电阻。

33.3 SWD 接口协议

33.3.1 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。

33.3.2 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

33.3.3 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。

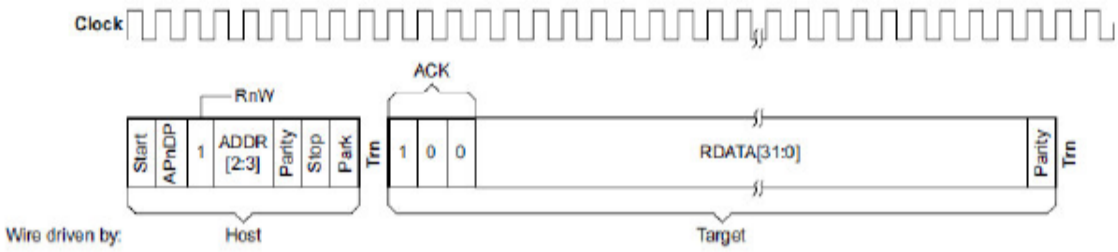
- 芯片复位，或者SWIO拉高50个SWCLK周期后，SW-DP处于RESET状态
- 拉低SWIO至少2个SWCLK周期后，SW-DP进入IDLE状态
- 当SW-DP处于RESET，主机必须先使其进入IDLE，然后对ID code寄存器进行读操作，才能激活SW-DP。否则从机会对主机的通信回应FAULT响应。

33.3.4 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。

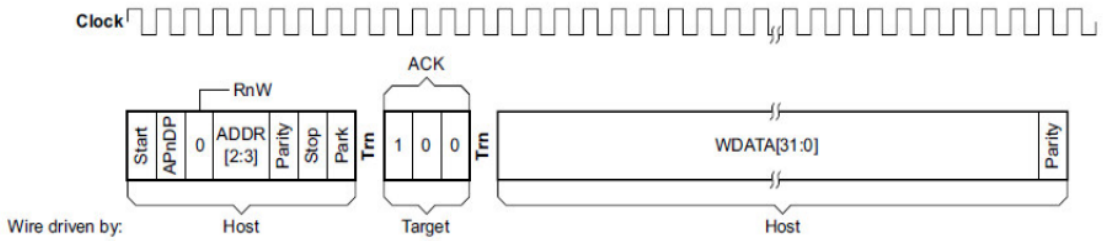


33.3.5 主机写操作

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。



33.4 SWD-DP 寄存器

33.4.1 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.

33.5 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

33.6 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4.6.1MCU DEBUG配置寄存器（DBG_CR）。

34 器件签名信息

每一颗FM33A0xxEV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。

34.1 存储器容量查询

通过查询SYSCON寄存器的bit2，可以获得器件Flash容量信息。

名称	SYSCON							
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	-					FLSCFG		
位权限	U-0					R/Dy-0		

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	FLSCFG	Flash 大小配置 0：512KB 1：256KB
1:0	--	RFU：未实现，读为 0

34.2 器件 UID

FM33A0xxEV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128bits，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_FA10~0x1FFF_FA1F。

34.3 芯片版本信息（ECO）

名称	SCU_VRSR							
地址	0x4000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	Rev							
位权限	R-0000 0001							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CID[15:8]							
位权限	R-0000 0100							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CID[7:0]							
位权限	R-0011 0000							

Bit	助记符	功能描述
31:24	--	RFU：未实现，读为 0
23:16	Rev	Revision，对应芯片版本号
15:0	CID	Chip ID，只读，固定为 0x0430

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/