乐升半导体
LEVETOP

LT168A/B

Uart TFT 显示控制芯片

规 格 书

V2.2

www.levetop.cn
乐升半导体有限公司

版本记录

版本	日期	描述
V1.0	2023/10/16	● LT168 初步发布
V1.1 A	2024/3/7	● 修正了以前版本中描述的语法错误。 ● 更新第 2.3 节中的信号特性摘要 ● 更新第 33 章的应用电路
V1.2	2024/3/8	● 更新表 2-3 LT168x 比较 ● 更新第 33 章的应用电路
V1.5	2024/6/11	● 添加表 3-1：有无 PSRAM 的 RGB 面板的 LT168B 的差异 ● 添加第 32.4 节 VDD 通电时间 ● 更新第 33 章的应用电路
V2.0	2024/8/16	● 更新表 3-1：TFT 面板 LT168B 的差异 ● 增加了第 34.3 节关于地面 Pa d 的 PCB 设计
V2.1	2025/7/11	● 新增圖 3-16 外加专用复位芯片参考原理图
V2.2	2025/10/15	● 更新表 32-6：复位与唤醒时间特性

版权说明

本文件之版权属于 乐升半导体 所有，若需要复制或复印请事先得到 乐升半导体 的许可。本文件记载之信息虽然都有经过校对，但是 乐升半导体 对文件使用说明的规格不承担任何责任，文件内提到的应用程序仅用于参考，乐升半导体 不保证此类应用程序不需要进一步修改。乐升半导体 保留在不事先通知的情况下更改其产品规格或文件的权利。有关最新产品信息，请访问我们的网站 Http://www.levetop.cn 。

©Levetop Semiconductor Co., Ltd.

内　容

版本记录 2
版权说明 2
内　容 3
图 列 表 20
表 列 表 28
1. LT168 介绍 31
1.1. 基本简介 31
1.2. 内部方块图 32
1.3. 功能说明 33
1.3.1. 32 位 RISC 处理器内核 33
1.3.2. 内建 768K Bytes 静态内存 SRAM 33
1.3.3. 内建 QSPI 闪存 33
1.3.4. 8K 字节的 ROM 33
1.3.5. 32K 字节的高速缓存 33
1.3.6. 外部总线接口 33
1.3.7. RGB 显示面板接口 34
1.3.8. DMA 控制器 34
1.3.9. 复位功能 34
1.3.10. 四周 PIT 定时器 34
1.3.11. 看门狗定时器（Watchdog Timer） 34
1.3.12. Real Time Clock 时钟 35
1.3.13. GPIO 与中断接口（EPORT） 35
1.3.14. 四线串口 SPI 模块（QSPI） 35
1.3.15. 串口通讯模块（SCI / Uart） 35
1.3.16. Canbus 控制器 36
1.3.17. USB2.0 Full Speed 控制器 37
1.3.18. 双脉宽调制器（PWM） 37
1.3.19. 8 通道 ADC 转换器 37
1.3.20. 两个模拟比较器（Comparator） 38
1.3.21. 电源管理单元（PMU） 38
1.3.22. 压检测器 38
1.3.23. 内部时钟 38
1.3.24. 外部晶振时钟 38
1.4. 系统方块 39
1.5. 模块应用 39

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

2. 信号描述 40
2.1. 芯片脚位图 40
2.2. 引脚信号特性 42
2.3. 信号说明 45
2.4. LT168A 与 LT168B 之比较 52
3. 硬件接口 53
3.1. 主控端 MCU 通讯接口 53
3.2. TFT LCD 屏的控制接口 53
3.3. QSPI 接口 56
3.4. LCD 触控屏接口 56
3.5. 时钟信号源接口 57
3.6. LCD 背光控制接口 57
3.7. Can Bus 接口 58
3.8. 声音输出接口 59
3.9. RTC（Real Time Clock）的时钟源与电源 59
3.10. 复位（Reset） 60
3.11. USB 接口 61
3.12. SPI/Uart 扩展接口 61
4. 内存与寄存器配置 62
4.1. 简要说明 62
4.2. 地址配置图 62
5. 嵌入式中断控制器（EIC） 64
5.1. 基本介绍 64
5.2. 功能特点 64
5.3. 内存映射和寄存器 64
5.3.1. 内存映射 64
5.3.2. 寄存器描述 66
5.3.2.1. 中断控制状态寄存器（ICSR） 66
5.3.2.2. 中断启用寄存器（IER） 68
5.3.2.3. 中断等待集寄存器（IPSR） 69
5.3.2.4. 中断待定清除寄存器（IPCR） 70
5.3.2.5. 优先级级别选择寄存器（PLSR） 71
5.3.2.6. 系统优先级别选择寄存器（SYSPLSR） 72
5.4. 功能描述 73

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

5.4.1. 无冲突的中断处理 74
5.4.2. 有冲突的中断 74
5.4.3. 待定陷阱功能 75
5.5. 中断来源 76
6. 32 位 RISC 微处理器介绍 81
6.1. 功能特点 81
6.2. 微体系结构总结 81
6.3. 程序设计模型 82
6.4. 数据格式汇总 83
6.5. 操作和寻址能力 84
6.6. 指令集概述 84
7. 嵌入式可编程定时器（EPT） 87
7.1. 基本介绍 87
7.2. 内存映射和寄存器 87
7.2.1. 内存映射 87
7.2.2. 寄存器描述 88
7.2.2.1. EPT 控制状态寄存器（EPTCSR） 88
7.2.2.2. EPT 重新加载寄存器（EPTRLD） 89
7.2.2.3. EPT 计数寄存器（EPTCNT） 90
7.3. 功能描述 91
7.3.1. 计数时序 91
8. 芯片配置模块（CCM） 92
8.1. 基本介绍 92
8.2. 功能特点 92
8.3. 内存映射和寄存器 92
8.3.1. 内存映射 92
8.3.2. 寄存器描述 93
8.3.2.1. 唤醒配置寄存器（WKUPC） 93
8.3.2.2. 芯片引脚下拉配置寄存器（CPPDC） 95
8.3.2.3. QSPI XIP 模式配置寄存器（QSPI） 96
8.3.2.4. QSPI 32 位密钥寄存器（QSPI QSPI） 97
8.3.2.5. QSPI GPIO 配置寄存器（QSPI） 97
8.3.2.6. MCU 访问 RAM 优先级配置寄存器（内存） 98
8.3.2.7. EPORT2 功能配置寄存器（EPORT2FCR） 100
9. 时钟和电源控制模块（CLKPWRM） 101
9.1. 基本概述 101

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

9.2. 功能特点 101
9.3. 时钟结构 101
9.4. 时钟源选择 102
9.4.1. 低功耗选项 102
9.4.1.1. 等待和多兹模式 102
9.4.1.2. 停止模式 102
9.5. 内存映射和寄存器 102
9.5.1. 内存映射 103
9.5.2. 寄存器说明 104
9.5.2.1. 合成器控制寄存器（SYNCR） 104
9.5.2.2. 低速振荡器控制和状态寄存器（LOSCCSR） 108
9.5.2.3. PLL 配置和状态寄存器（PLLCSR） 110
9.5.2.4. 模块停止控制寄存器（MSCR） 112
9.5.2.5. EPT 外部时钟源启用控制寄存器（ECSECR） 114
9.5.2.6. OSC Bist 测试配置寄存器 1（OBTCR1） 115
9.5.2.7. OSC Bist 测试配置寄存器 2（OBTCR2） 116
9.5.2.8. OSC Bist 测试控制寄存器（OBTCTLR） 117
9.5.2.9. OSC Bist 测试计数器寄存器（OBTCNTR） 118
9.5.2.10. OSC Bist 测试结果寄存器（OBTRR） 119
9.6. 功能描述 120
9.6.1. 打开 PLL 120
9.6.2. PLL 测量的频率 120
9.6.3. 128 KHz 测量的频率 120
10. 复位控制模块（RCM） 121
10.1. 基本概述 121
10.2. 功能特点 121
10.3. 方块图 121
10.4. 内存映射和寄存器 122
10.4.1. 内存映射 122
10.4.2. 寄存器说明 122
10.4.2.1. 复位测试寄存器（RTR） 122
10.4.2.2. 复位状态寄存器（RSR） 123
10.4.2.3. 复位控制寄存器（RCR） 124
10.5. 功能描述 125
10.5.1. 复位源 125
10.5.1.1. 通电复位（POR） 125
10.5.1.2. 看门狗计时器复位 125
10.5.1.3. 软件复位 125

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

10.5.1.4. 可编程电压检测复位 125
10.5.2. 复位控制流程 125
11. 静态随机存取存储器（SRAM） 127
11.1. 功能介绍 127
11.2. 操作模式 127
11.3. 低功耗模式 127
11.4. 复位操作 127
11.5. 中断说明 127
12. 高速缓存模块（CACHEM） 128
12.1. 功能介绍 128
12.2. 方块图 129
12.3. 内存映射和寄存器 130
12.3.1. 内存映射 130
12.3.2. 寄存器说明 131
12.3.2.1. 高速缓存控制寄存器（LMEM_CCR） 131
12.3.2.2. 高速缓存线控制寄存器（LMEM_CLCR） 132
12.3.2.3. 高速缓存搜索地址寄存器（LMEM_CSAR） 134
12.3.2.4. 高速缓存读/写值寄存器（LMEM_CCVR） 135
12.3.2.5. 高速缓存访问寄存器（LMEM_ACRG） 136
12.3.2.6. 高速缓存页面未验证基本地址寄存器（LMEM_PAGE_INV_BADDR） 137
12.3.2.7. 高速缓存页面未验证基本大小寄存器（LMEM_PAGE_INV_SIZE） 138
12.3.2.8. 高速缓存时钟启用寄存器（LMEM_CACHE_CLK_EN） 139
12.4. 缓存功能 140
12.5. Cache 缓存控制 141
12.5.1. 缓存集命令 141
12.5.2. Cache Line Commands 缓存行命令 142
12.5.2.1. 使用缓存地址执行一系列行命令 143
12.5.2.2. 使用物理地址执行一系列行命令 143
12.5.2.3. 行命令结果 144
13. 横杆开关（XBAR） 145
13.1. 功能介绍 145
13.1.1. 基本概述 145
13.1.2. 功能特点 145
13.2. 内存映射和寄存器 146
13.2.1. 内存映射 146
13.2.2. 寄存器描述 147
13.2.2.1. 主优先级寄存器（XBAR_MPRn） 147

©Levetop Semiconductor Co., Ltd.

13.2.2.2. 从属通用控制寄存器（XBAR_SGPCRn） 149
13.3. 功能说明 151
13.3.1. 一般操作 151
13.3.2. 寄存器相干性 151
13.3.3. XBAR 仲裁 151
13.3.3.1. 固定优先级操作 152
13.3.3.2. 循环优先操作 152
13.3.4. 优先级分配 152
13.4. 初始化/应用程序信息 152
14. 直接内存访问控制器（DMAC） 153
14.1. DMA 控制器的具体信息 153
14.1.1. DMAC 功能 153
14.1.2. 通道分配 154
14.2. 功能介绍 155
14.2.1. 模块特点 156
15. Efuse 控制模块（EFM） 157
15.1. 功能介绍 157
15.2. 控制模块特点 157
15.3. 内存映射和寄存器 157
15.3.1. 内存映射 157
15.3.2. 寄存器描述 158
15.3.2.1. Efuse 配置寄存器（EFMCR） 158
15.3.2.2. 清除读取数据寄存器（EFMDR） 159
16. 选项字节（OPB） 160
16.1. 内存映射和寄存器 160
16.1.1. 内存映射 160
16.1.2. 寄存器描述 160
16.1.2.1. 可编程电压检测器配置寄存器（PVDC） 160
16.1.2.2. 客户配置寄存器（CCR） 162
16.1.2.3. 外部高速振荡器稳定时间配置寄存器（EOSCST） 165
16.1.2.4. PLL 锁定时间配置寄存器（请） 166
16.1.2.5. 复位引脚过滤器启用和值寄存器（RFEVR） 166
16.1.2.6. 可编程电压检测器滤波器启用和值寄存器（PVDFEVR） 167
16.1.2.7. 工厂配置寄存器（FCR） 168
16.1.2.8. 信道禁用配置寄存器（附件） 169
16.1.2.9. VREF1P2 修剪配置寄存器（VREFTCR） 170
16.1.2.10. LDO1P2 修剪配置寄存器（LDOTCR） 171
16.1.2.11. RTC 修剪配置寄存器（RTCTCR） 172

©Levetop Semiconductor Co., Ltd.

17. RGB 控制器模块（RGBC） 173
17.1. 功能介绍 173
17.2. 方块图 173
17.3. RGB 接口定义 174
17.4. RGB 接口定时 175
17.5. 内存映射和寄存器 176
17.5.1. 内存映射 176
17.5.2. 寄存器描述 177
17.5.2.1. 水平脉冲宽度和后门廊寄存器（THPB） 177
17.5.2.2. 水平周期和前门廊寄存器（THDF） 178
17.5.2.3. 垂直脉冲宽度和后门廊寄存器（TVPB） 179
17.5.2.4. 垂直周期和前门廊寄存器（TVDF） 180
17.5.2.5. 内存起始地址寄存器（msadr） 181
17.5.2.6. 内存端地址寄存器（MEADDR） 182
17.5.2.7. 已传输的帧数（NFTR） 183
17.5.2.8. 传输阈值寄存器配置的帧数（NFTTCR） 184
17.5.2.9. RGB 控制和状态寄存器（RCSR） 185
17.6. 功能描述 187
18. Blender 控制器（BLDC） 188
18.1. 功能介绍 188
18.2. 混合算法 189
18.3. 内存映射和寄存器 189
18.3.1. 内存映射 189
18.3.2. 寄存器描述 190
18.3.2.1. Blender 控制和状态寄存器（BCSR） 190
18.3.2.2. 源地址寄存器（SADDR） 191
18.3.2.3. 传输总地址长度寄存器（TTALR） 192
18.3.2.4. 目标地址寄存器（DADDR） 193
18.3.2.5. 传输目的地次要长度和偏移寄存器（TDMLOR） 194
19. 外部总线接口（EBI） 195
19.1. 功能介绍 195
19.2. 总线接口特点 195
19.3. 信号描述 195
19.4. 内存映射和寄存器 195
19.4.1. 内存映射 195
19.4.2. 寄存器描述 196
19.4.2.1. EBI 控制寄存器（EBICR） 196

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

19.4.2.2. IO 拉力控制寄存器（IOPCR）.....198
19.4.2.3. IO 控制寄存器（IOCR）.....199
19.4.2.4. GPIO 数据输出寄存器（GPIODO）.....200
19.4.2.5. GPIO 方向寄存器（直径）.....201
19.4.2.6. GPIO 数据输入寄存器（GPIODI）.....202
19.5. 功能描述.....203
19.5.1. EBI_WR#、EBI_RD#信号定时.....203
19.5.2. EBI 功能描述.....204
19.5.2.1. EBI_CS#芯片选择.....204
19.5.2.2. 操作转输.....204
20. 可编程中断定时器（PIT）.....206
20.1. 功能介绍.....206
20.2. 方框方块图.....206
20.3. 操作模式.....206
20.3.1. 等待模式.....206
20.3.2. Doze 模式.....206
20.3.3. 停止模式.....206
20.3.4. 调试模块.....206
20.4. 信号说明.....207
20.5. 内存映射和寄存器.....207
20.5.1. 内存映射.....207
20.5.2. 寄存器描述.....207
20.5.2.1. PIT 模量寄存器（PMR）.....208
20.5.2.2. PIT 控制和状态寄存器（PCSR）.....208
20.5.2.3. PIT 计数寄存器（PCNTR）.....211
20.6. 功能描述.....211
20.6.1. 设置和忘记计时器操作.....211
20.6.2. 自由运行计时器操作.....212
20.6.3. 超时规格.....212
20.7. 中断操作.....212
21. 看门狗计时器（WDT）.....213
21.1. 功能介绍.....213
21.2. 操作模式.....213
21.2.1. 等待模式.....213
21.2.2. Doze 模式.....213
21.2.3. 停止模式.....213
21.2.4. 调试模式.....213

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

21.3. 方块图 214
21.4. 信号说明 214
21.5. 内存映射和寄存器 214
21.5.1. 内存映射 214
21.5.2. 寄存器说明 215
21.5.2.1. 看门狗模块寄存器（WMR） 215
21.5.2.2. 看门狗控制寄存器（WCR） 216
21.5.2.3. 看门狗服务寄存器（WSR） 218
21.5.2.4. 监视器计数寄存器（WCNTR） 218
22. 时钟控制器（RTC） 219
22.1. 功能介绍 219
22.2. RTC 特点 219
22.3. 测试模式 219
22.4. 方块图 219
22.5. 应用电路 220
23. 边缘端口模块（EPORT） 221
23.1. 功能介绍 221
23.2. 低功耗模式运行 221
23.2.1. 等待和多兹模式 221
23.2.2. 停止模式 221
23.3. 中断/通用输入引脚说明 222
23.4. 内存映射和寄存器 222
23.4.1. 内存映射 222
23.4.2. 寄存器说明 223
23.4.2.1. 边缘端口中断启用寄存器（EPIER） 223
23.4.2.2. Eport 数据方向寄存器（EPDDR） 224
23.4.2.3. Eport 引脚分配寄存器（EPPAR） 224
23.4.2.4. Eport 引脚向上上拉启用寄存器（EPPUE） 225
23.4.2.5. 边缘端口标志寄存器（EPFR） 226
23.4.2.6. 边缘端口引脚数据寄存器（EPPDR） 226
23.4.2.7. 边缘端口数据寄存器（EPDR） 227
23.4.2.8. Eport 端口位设置寄存器（EPBSR） 227
23.4.2.9. Eport 数字滤波器控制寄存器（EPFC） 228
23.4.2.10. Eport Open Drain 启用寄存器（EPODE） 228
23.4.2.11. Eport 电平极性寄存器（EPLPR） 229
23.4.2.12. Eport 端口位清除寄存器（EPBCR） 229
24. CANBus 控制程序（CANBC） 230

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

24.1. 功能介绍 230
24.1.1. 基本概述 230
24.1.2. CANBus 模块功能 231
24.1.3. 操作模式 232
24.2. 外部信号描述 232
24.2.1. 基本概述 232
24.2.2. 信号描述 233
24.2.2.1. C ANRX 233
24.2.2.2. C ANTX 233
25. 串行通信接口（SCI） 234
25.1. 功能介绍 234
25.2. 通信接口特点 234
25.3. 操作模式 235
25.4. 方块图 235
25.5. 操作模式 236
25.5.1. 停止模式 236
25.5.2. 等待 Mode 236
25.6. 信号描述 236
25.7. 内存映射和寄存器 237
25.7.1. 内存映射 237
25.7.2. 寄存器说明 238
25.7.2.1. SCI 版本 ID 寄存器（SCI_VERID） 238
25.7.2.2. SCI 参数寄存器（SCI_PARAM） 239
25.7.2.3. SCI 复位寄存器（SCI_RESET） 240
25.7.2.4. SCI 引脚寄存器（SCI_PIN） 241
25.7.2.5. SCI 波特率寄存器（SCI_BAUD） 242
25.7.2.6. SCI 状态寄存器（SCI_STAT） 244
25.7.2.7. SCI 控制寄存器（SCI_CTRL） 248
25.7.2.8. SCI 数据寄存器（SCI_DATA） 253
25.7.2.9. SCI 匹配地址寄存器（SCI_MATCH） 255
25.7.2.10. SCI 调制解调器 IrDA 寄存器（SCI_MODIR） 256
25.7.2.11. SCI FIFO 寄存器（SCI_FIFO） 258
25.7.2.12. SCI 水印寄存器（SCI_WATER） 261
25.7.2.13. SCI 过采样比寄存器（SCI_OSR） 262
25.8. 功能描述 263
25.9. 波特率生成 263
25.10. 发送器功能描述 264
25.10.1. 发送休息和排队的空闲时间 264

©Levetop Semiconductor Co., Ltd.

25.11. 接收器功能说明 265
25.11.1. 数据采样技术 265
25.11.2. 接收器唤醒操作 266
25.11.2.1. 怠速排队 267
25.11.2.2. 地址标记唤醒 267
25.11.2.3. 数据匹配唤醒 267
25.11.2.4. 地址匹配操作 268
25.11.2.5. 空闲匹配操作 268
25.11.2.6. 匹配匹配关闭操作 268
25.11.3. 红外解码器 269
25.11.3.1. 开始位检测 269
25.11.3.2. 噪声滤波 269
25.11.3.3. 低位检测 269
25.11.3.4. 高比特检测 269
25.12. 额外的 SCI 功能 270
25.12.1. 8 位、9 位和 10 位的数据模式 270
25.12.2. 空闲长度 270
25.12.3. 单线操作 270
25.12.4. 循环模式 271
25.13. 红外线接口 271
25.13.1. 红外发射编码器 271
25.13.2. 红外接收解码器 271
25.14. 中断和状态标志 272
26. 同步串行接口（SSI） 273
26.1. 功能介绍 273
26.2. 串行接口特点 273
26.3. 操作模式 273
26.4. 方块图 274
26.5. 应用图 274
26.6. 内存映射和寄存器 275
26.6.1. 内存映射 275
26.6.2. 寄存器描述 276
26.6.2.1. 控制寄存器 0（CTRLR0） 276
26.6.2.2. 控制寄存器 1（CTRLR1） 279
26.6.2.3. SSI 启用寄存器（SSIENR） 280
26.6.2.4. 微线控制寄存器（MWCR） 281
26.6.2.5. 从机启用寄存器（SER） 282
26.6.2.6. 波特率选择（BAUDR） 283

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.7. 传输 FIFO 阈值等级（TXFTLR）.....284
26.6.2.8. 接收 FIFO 阈值级别（RXFTLR）.....285
26.6.2.9. 发射 FIFO 液位寄存器（TXFLR）.....286
26.6.2.10. 接收 FIFO 级别寄存器（RXFLR）.....287
26.6.2.11. 状态寄存器（SR）.....288
26.6.2.12. 中断掩码寄存器（IMR）.....289
26.6.2.13. 中断状态寄存器（ISR）.....291
26.6.2.14. 原始中断状态寄存器（RISR）.....292
26.6.2.15. 传输 FIFO 溢出中断清除寄存器（TXOICR）.....293
26.6.2.16. 接收 FIFO 溢出中断清除寄存器（RXOICR）.....294
26.6.2.17. 接收 FIFO 下流中断清除寄存器（RXUICR）.....295
26.6.2.18. 中断清除寄存器（ICR）.....296
26.6.2.19. DMA 控制寄存器（DMACR）.....297
26.6.2.20. DMA 传输数据级别（DMATDLR）.....298
26.6.2.21. DMA 接收数据级别（DMARDLR）.....299
26.6.2.22. 识别寄存器（IDR）.....300
26.6.2.23. 版本 ID 寄存器（VIDR）.....301
26.6.2.24. SSI 数据寄存器（DRx）.....302
26.6.2.25. RX 取样延迟寄存器（RXSDR）.....303
26.6.2.26. SPI 控制寄存器 0（SPICTRLR0）.....304
26.6.2.27. XIP 模式位（XIPMBR）.....306
26.6.2.28. XIP 中初始寄存器（XIPIIR）.....307
26.6.2.29. XIP 封装初始寄存器（XIPWIR）.....308
26.6.2.30. XIP 控制寄存器（XIPCR）.....309
26.6.2.31. XIP 从属服务器启用寄存器（XIPSER）.....311
26.6.2.32. XIP 接收 FIFO 溢出中断清除寄存器（XRXIOCR）.....312
26.6.2.33. XIP 连续传输超时寄存器（XIP）.....313
26.7. 功能描述.....314
26.7.1. 主模式.....314
26.7.2. 时钟比率.....314
26.7.3. 接收和传输 FIFO 缓冲区.....315
26.7.3.1. 传输 FIFO.....315
26.7.3.2. 接收 FIFO.....315
26.7.4. DMA 操作.....315
26.7.5. SSI 中断.....315
26.7.6. 增强的 SPI 模式.....316
26.7.7. 就地执行（XIP）模式.....317
26.7.8. 在 XIP 中的连续传输模式.....317
26.7.9. 在 XIP 操作中的数据预取.....318
27. Inter-Integrated Circuit（I2C）.....319

©Levetop Semiconductor Co., Ltd.

27.1. 功能介绍 319
27.2. I2C 特点 319
27.3. 系统和方块图 320
27.4. 内存映射和寄存器 320
27.4.1. 内存映射 320
27.4.2. 寄存器描述 321
27.4.2.1. I2C 状态寄存器（I2CS） 321
27.4.2.2. I2C 时钟电压前寄存器（I2CP） 322
27.4.2.3. I2C 控制寄存器（I2CC） 323
27.4.2.4. I2C 从属地址寄存器（I2CSA） 325
27.4.2.5. I2C 端口控制寄存器（I2CPCR） 325
27.4.2.6. I2C 从属高速模式指示器寄存器（I2CSHIR） 326
27.4.2.7. I2C 从属 SDA 保持时间寄存器（I2CSHT） 326
27.4.2.8. I2C 数据寄存器（I2CD） 327
27.4.2.9. I2C 端口方向寄存器（I2CDDR） 327
27.4.2.10. I2C 端口数据寄存器（I2CPDR） 328
27.5. 功能描述 328
27.5.1. 主模式 328
27.5.2. 从属模式 328
27.5.3. 通信协议 329
27.5.4. 仲裁程序 330
27.5.5. 时钟同步 330
27.5.6. 握手机制 330
27.5.7. 时钟拉伸 331
27.5.8. 高速模式运行 331
27.5.9. 软件交易流程图 333
28. 脉冲宽度调制器（PWM） 336
28.1. 功能介绍 336
28.2. PWM 特点 336
28.3. 方块图 337
28.4. 信号描述 337
28.5. 内存映射和寄存器 338
28.5.1. 内存映射 338
28.5.2. 寄存器描述 339
28.5.2.1. PWM 预缩放寄存器（PPR） 339
28.5.2.2. PWM 时钟选择寄存器（PCSR） 340
28.5.2.3. PWM 对照寄存器（PCR） 341
28.5.2.4. PWM 计数器寄存器（PCNRn） 343

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

28.5.2.5. PWM 比较器寄存器（PCMRn）.....346
28.5.2.6. PWM 计时器寄存器（PTRn）.....349
28.5.2.7. PWM 中断启用寄存器（PIER）.....351
28.5.2.8. PWM 中断标志寄存器（PIFR）.....352
28.5.2.9. PWM 捕获控制寄存器（PCCR0/1）.....353
28.5.2.10. PWM 捕获上升锁存寄存器（PCRLRn）.....355
28.5.2.11. PWM 捕获坠落锁存器寄存器（PCFLRn）.....358
28.5.2.12. PWM 端口控制寄存器（PPCR）.....360
28.6. 功能描述.....361
28.6.1. PWM 双缓冲和自动重新加载.....361
28.6.2. 调节占空比.....361
28.6.3. 死区发电机.....362
28.6.4. PWM 计时器启动过程.....362
28.6.5. PWM 计时器停止程序.....362
28.6.6. 捕获启动过程.....363
28.6.7. 捕获基本计时器操作.....363
29. 模拟比较器（COMP）.....364
29.1. 功能介绍.....364
29.2. 方块图.....364
29.3. 操作模式.....365
29.3.1. 等待模式.....365
29.3.2. Doze 模式.....365
29.3.3. 停止模式.....365
29.4. 内存映射和寄存器.....365
29.4.1. 内存映射.....365
29.4.2. 寄存器描述.....366
29.4.2.1. 比较器控制寄存器（CPTCN）.....366
29.4.2.2. 比较器模式选择寄存器（CPTMD）.....367
29.4.2.3. 比较器 MUX 选择寄存器（CPTMX）.....368
29.4.2.4. 比较器输出过滤器选择寄存器（CPTFLS）.....369
29.5. 功能描述.....369
30. USB2.0 控制器（USBC）.....371
30.1. 功能介绍.....371
30.2. USB 模块特点.....371
30.3. 方块图.....372
30.4. 操作模式.....372
30.4.1. 等待模式.....372
30.4.2. Doze 模式.....372

©Levetop Semiconductor Co., Ltd.

Levetop Semiconductor

30.4.3. 停止模式.....372
30.5. 内存映射和寄存器.....373
30.5.1. 内存映射.....373
30.5.2. 寄存器描述.....374
30.5.2.1. USBPHY 控制寄存器 1（USBPHY_CTRL1）.....374
30.5.2.2. 中断状态寄存器（INT_STAT）.....375
30.5.2.3. 中断启用寄存器（INT_ENB）.....376
30.5.2.4. 错误中断状态寄存器（ERR_STAT）.....378
30.5.2.5. 错误中断启用寄存器（ERR_ENB）.....379
30.5.2.6. 状态寄存器（STAT）.....381
30.5.2.7. 控制寄存器（CTL）.....382
30.5.2.8. 地址寄存器（ADDR）.....384
30.5.2.9. EBT 页面寄存器 1（EBT_PAGE_01）.....385
30.5.2.10. 帧数寄存器（FRMNUML）.....386
30.5.2.11. 帧数寄存器（FRMNUMH）.....387
30.5.2.12. 令牌寄存器（TOKEN）.....388
30.5.2.13. SOF 阈值寄存器（SOF_THLD）.....389
30.5.2.14. EBT 页面寄存器 2（EBT_PAGE_02）.....390
30.5.2.15. EBT 页面寄存器 3（EBT_PAGE_03）.....391
30.5.2.16. 端点控制寄存器（ENDPTn）.....391
30.5.2.17. USBPHY 控制寄存器 2（USBPHY_CTRL2）.....393
30.5.2.18. USB PHY 观察寄存器（USB_PHY_OBSERVE）.....394
30.5.2.19. USBPHY GPIO 寄存器（USB_PHY_GPIO）.....395
30.5.2.20. USB 恢复启用寄存器（USB_RESMEN）.....396
30.5.2.21. USB PHY 控制寄存器 3（USBPHY_CTRL3）.....397
30.5.2.22. USB PHY 控制寄存器 4（USBPHY_CTRL4）.....398
30.6. 功能描述.....399
30.6.1. 数据结构.....399
30.6.2. 端点缓冲区表.....399
30.6.3. Rx vs。Tx 作为一个 USB 目标设备.....400
30.6.4. 寻址端点缓冲区表条目.....400
30.6.5. 端点缓冲区表格式.....401
30.6.6. USB Transaction.....404
31. 模数转换器（ADC）.....406
31.1. 功能介绍.....406
31.2. ADC 主要功能.....406
31.3. ADC 功能描述.....407
31.3.1. ADC 开关控制（ADEN、ADDIS、ADRDY）.....408
31.3.2. ADC 时钟.....409

©Levetop Semiconductor Co., Ltd.

31.3.3. 配置 ADC 409
31.3.4. 通道选择（CCWi） 409
31.3.5. 可编程采样时间（SMP） 410
31.3.6. 单次转换模式（CONT = 0） 410
31.3.7. 连续转换模式（CONT = 1） 411
31.3.8. 正在启动转换（ADSTART） 411
31.3.9. 时间安排 412
31.3.10. 停止一个持续的转换（ADSTP） 413
31.4. 在外部触发器和触发器极性上的转换 413
31.4.1. 不连续模式（DISCEN） 415
31.4.2. 可编程分辨率（RES）- 快速转换模式 415
31.4.3. 转换结束，采样阶段结束(EOC、EOSMP Flag） 416
31.4.4. 转换序列的结束(Eoseq Flag） 416
31.4.5. 时序图范例 416
31.5. 数据管理 418
31.5.1. 数据 FIFO 和数据对齐(ADC_FIFO，对齐） 418
31.5.2. ADC 超运行（OVR、OVRMOD） 418
31.5.3. 管理不使用 DMA 而转换的数据序列 419
31.5.4. 管理转换的数据而不使用 DMA 419
31.5.5. 使用 DMA 管理转换后的数据 419
31.6. 低功耗功能 420
31.6.1. 等待模式转换 420
31.6.2. 自动关闭模式（AUTOFF） 420
31.7. 模拟窗口监视器（AWD） 420
31.8. 温度传感器 421
31.9. ADC 中断 421
31.10. 内存映射和寄存器 422
31.10.1. 内存映射 422
31.10.2. 寄存器描述 423
31.10.2.1. ADC 中断和状态寄存器（ADC_ISR） 423
31.10.2.2. ADC 中断启用寄存器（ADC_IER） 425
31.10.2.3. ADC 控制寄存器（ADC_CR） 426
31.10.2.4. ADC 配置寄存器 1（ADC_CFGR1） 428
31.10.2.5. ADC 配置寄存器 2（ADC_CFGR2） 430
31.10.2.6. ADC 采样时间寄存器（ADC_SMPR） 431
31.10.2.7. ADC 看门狗寄存器（ADC_WDG） 432
31.10.2.8. ADC 看门狗阈值寄存器（ADC_TR） 433
31.10.2.9. ADC 通道选择寄存器（ADC_CHSELR） 434
31.10.2.10. ADC FIFO 访问寄存器（ADC_FIFO） 435

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

32.电气特性 436
32.1. 极限参数 436
32.2. 静电放电（ESD）保护 436
32.3. DC 电气参数 437
32.4. VDD 上电时序 438
33.应用电路 439
34.封装讯息 444
34.1. LT168A (QFN-48pin) 444
34.2. LT168B (QFN-68pin) 446
34.3. 芯片接地焊盘的 PCB 设计 447

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

图 列 表

图 1-1：LT168A 和 LT168B 外观图 32
图 1-2：内部方块图 32
图 1-3：LT168 的系统方块图 39
图 1-4：TFT 液晶显示器模块应用示图 39
图 2-1：LT168A（QFN48）脚位图 40
图 2-2：LT168B（QFN68）脚位图 41
图 3-1：LT168 的串口和主控 MCU 的通讯 53
图 3-2：LT168A 与 8 位 8080 并行接口的 TFT 屏连接 53
图 3-3：LT168B 与 16 位 8080 并行接口的 TFT 屏连接 54
图 3-4：LT168B 与 RGB 接口的 TFT 屏连接 54
图 3-5：LT168B 与 QSPI 接口的 TFT 屏连接 55
图 3-6：LT168 连接 QSPI Flash 原理图 56
图 3-7：LT168 连接到电阻触控屏 56
图 3-8：LT168 连接到电容触控屏 56
图 3-9：外部的 12MHz 晶振原理图 57
图 3-10：TFT LCD 背光控制参考原理图 - 1 57
图 3-11：TFT LCD 背光控制参考原理图 - 2 58
图 3-12：Canbus 驱动原理图 58
图 3-13：声音输出参考原理图 59
图 3-14：RTC 时钟源与外部电池电源的参考原理图 59
图 3-15：外部复位参考原理图 60
图 3-16：外加专用复位芯片参考原理图 60
图 3-17：LT168 连接到 USB 端口的应用电路 61
图 3-18：额外的 SPI、Uart 接口参考原理图 61
图 4-1：地址配置图 62
图 5-1：中断控制状态寄存器（ICSR） 66
图 5-2：中断启用寄存器（IER） 68
图 5-3：中断等待集寄存器（IPSR） 69
图 5-4：中断待定清除寄存器（IPCR） 70
图 5-5：优先级选择寄存器（PLSR0-PLSR31） 71
图 5-6：系统优先级级别选择寄存器（SYSPLSR） 72
图 5-7：One Pulse Interrupt without Confliction 74
图 5-8：Level-sensitive Interrupt without Confliction 74
图 5-9：同时发生两次中断 74
图 5-10：具有冲突的较低优先级中断 75
图 5-11：具有冲突的更高优先级中断 75
图 6-1：编程模型 82
图 6-2：内存中的数据结构 83

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

图 6-3：寄存器中的数据结构.....83
图 7-1：EPT 控制状态寄存器（EPTCSR）.....88
图 7-2：EPT 重新加载寄存器（EPTRLD）.....89
图 7-3：EPT 计数器寄存器（EPTCNT）.....90
图 7-4：EPT 计数时序.....91
图 8-1：唤醒配置寄存器（WKUPC）.....93
图 8-2：芯片引脚下拉配置寄存器（CPPDC）.....95
图 8-3：QSPI XIP 模式配置寄存器（QSPI）.....96
图 8-4：QSPI 32 位密钥寄存器（q QSPI keyr）.....97
图 8-5：QSPI GPIO 配置寄存器（QSPIGPIOCR）.....97
图 8-6：MCU 访问 RAM 优先级配置寄存器（MCURAMPRIOCR）.....98
图 8-7：EPORT2 功能配置寄存器（EPORT2FCR）.....100
图 9-1：时钟结构.....101
图 9-2：合成器控制寄存器（SYNCR）.....104
图 9-3：低速振荡器控制和状态寄存器（LOSCCSR）.....108
图 9-4：PLL 配置和状态寄存器（PLLCSR）.....110
图 9-5：模块停止控制寄存器（MSCR）.....112
图 9-6：EPT 外部时钟源启用控制寄存器（ECSECR）.....114
图 9-7：OSC Bist 测试配置寄存器 1（OBTCR1）.....115
图 9-8：OSC Bist 测试配置寄存器 2（OBTCR2）.....116
图 9-9：OSC Bist 测试控制寄存器（OBTCTLR）.....117
图 9-10：OSC Bist 测试计数器寄存器（OBTCNTR）.....118
图 9-11：OSC Bist 测试结果寄存器（OBTRR）.....119
图 10-1：复位控制器的方块图.....121
图 10-2：复位测试寄存器（RTR）.....122
图 10-3：复位状态寄存器（RSR）.....123
图 10-4：复位控制寄存器（RCR）.....124
图 10-5：复位控制流程.....126
图 12-1：高速缓存模块的方块图.....129
图 12-2：高速缓存控制寄存器（LMEM_CCR）.....131
图 12-3：高速缓存线控制寄存器（LMEM_CLCR）.....132
图 12-4：高速缓存搜索地址寄存器（LMEM_CSAR）.....134
图 12-5：高速缓存读/写值寄存器（LMEM_CCVR）.....135
图 12-6：高速缓存访问寄存器（LMEM_ACRG）.....136
图 12-7：高速缓存页不验证基本地址寄存器（LMEM_PAGE_INV_BADDR）.....137
图 12-8：高速缓存页面不验证大小寄存器（LMEM_PAGE_INV_SIZE）.....138
图 12-9：高速缓存时钟启用寄存器（LMEM_CACHE_CLK_EN）.....139
图 12-10：高速缓存标签和数据访问结构.....140
图 13-1：XBAR 设备特定的方块图.....145
图 13-2：主优先级寄存器（XBAR_MPRn）.....147

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

图 13-3：从属通用控制寄存器（XBAR_SGPCRn） 149
图 14-1：DMA 结构图 155
图 15-1：Efuse 模块配置寄存器（EFMCR） 158
图 15-2：清除读取正时 159
图 15-3：Efuse 读取数据寄存器（EFMDR） 159
图 16-1：可编程电压检测器配置寄存器（PVDC） 160
图 16-2：客户配置寄存器（CCR） 162
图 16-3：外部高速振荡器稳定时间配置寄存器（EOSCST） 165
图 16-4：PLL 锁定时间配置寄存器（请） 166
图 16-5：复位引脚过滤器启用和值寄存器（RFEVR） 166
图 16-6：可编程电压检测器滤波器启用和值寄存器（PVDFEVR） 167
图 16-7：工厂配置寄存器（FCR） 168
图 16-8：ADC 信道禁用配置寄存器（ADCCDISR） 169
图 16-9：VREF 修剪配置寄存器（VREFTCR） 170
图 16-10：LDO1P2 修剪配置寄存器（LDOTCR） 171
图 16-11：RTC 修剪配置寄存器（RTCTCR） 172
图 17-1：RGB 控制器的方块图 173
图 17-2：通过 RGB 接口显示 RAM 访问区域 174
图 17-3：RGB 接口中的信号时序图 175
图 17-4：水平脉冲宽度和后门廊寄存器（THPB） 177
图 17-5：水平周期和前门廊寄存器（THDF） 178
图 17-6：垂直脉冲宽度和后门廊寄存器（TVPB） 179
图 17-7：垂直周期和前门廊寄存器（TVDF） 180
图 17-8：内存启动地址寄存器（MSADDR） 181
图 17-9：内存结束地址寄存器（MEADDR） 182
图 17-10：已转移寄存器的帧数（NFTR） 183
图 17-11：传输阈值寄存器配置的帧数（NFTTCR） 184
图 17-12：RGB 控制和状态寄存器（RCSR） 185
图 18-1：Blender 方块图 188
图 18-2：Blender 控制和状态寄存器（BCSR） 190
图 18-3：源地址寄存器（SADDR） 191
图 18-4：传输总地址长度寄存器（TTALR） 192
图 18-5：目标地址寄存器（DADDR） 193
图 18-6：传输目标次要长度和偏移寄存器（TDMLOR） 194
图 19-1：EBI 控制寄存器（EBICR） 196
图 19-2：IO 拉力控制寄存器（IOPCR） 198
图 19-3：IO 控制寄存器（IOCR） 199
图 19-4：GPIO 数据输出寄存器（GPIODO） 200
图 19-5：GPIO 方向寄存器（图形） 201
图 19-6：GPIO 数据输入寄存器（GPIODI） 202

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

图 19-7：EBI_WR#, EBI_RD# Assert/Negate Timing 203
图 20-1：PIT 方块图....206
图 20-2：PIT Modulus 寄存器（PMR） 208
图 20-3：PIT 控制和状态寄存器（PCSR） 208
图 20-4：PIT 计数寄存器（PCNTR） 211
图 20-5：计数器从 Modulus Latch 重新加载 211
图 20-6：在自由运行模式下的计数器....212
图 21-1：看门狗计时器方块图 214
图 21-2：看门狗模量寄存器（WMR） 215
图 21-3：看门狗控制寄存器（WCR） 216
图 21-4：看门狗服务寄存器（WSR） 218
图 21-5：看门机构统计寄存器（WCNTR） 218
图 22-1：RTC 方块图....219
图 22-2：RTC 应用电路....220
图 23-1：EPORT 方块图....221
图 23-2：EPORT 端口中断启用寄存器（EPIER） 223
图 23-3：EPORT 数据方向寄存器（EPDDR） 224
图 23-4：EPORT 引脚分配寄存器（EPPAR） 224
图 23-5：EPORT 引脚上拉式启用寄存器（EPPUE） 225
图 23-6：EPORT 端口标志寄存器（EPFR） 226
图 23-7：EPORT 端口引脚数据寄存器（EPPDR） 226
图 23-8：EPORT 端口数据寄存器（EPDR） 227
图 23-9：EPORT 端口位设置寄存器（EPBSR） 227
图 23-10：EPORT 数字滤波器控制寄存器（EPFC） 228
图 23-11：EPORT Open Drain 启用寄存器（EPODE） 228
图 23-12：EPORT 级极性寄存器（EPLPR） 229
图 23-13：EPORT 端口位清除寄存器（EPBCR） 229
图 24-1：CANBus 方块图 230
图 24-2：Canbus 电路示例....233
图 25-1：SCI 发送器方块图....235
图 25-2：SCI 接收器方块图....236
图 25-3：SCI 版本 ID 寄存器（SCI_VERID） 238
图 25-4：SCI 参数寄存器（SCI_PARAM） 239
图 25-5：SCI 复位寄存器（SCI_RESET） 240
图 25-6：SCI 引脚登记（SCI_PIN） 241
图 25-7：SCI 波特率寄存器（SCI_BAUD） 242
图 25-8：SCI 状态登记器（SCI_STAT） 244
图 25-9：SCI 控制寄存器（SCI_CTRL） 248
图 25-10：SCI 数据寄存器（SCI_DATA） 253
图 25-11：SCI 匹配地址寄存器（SCI_MATCH） 255

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

图 25-12：SCI 调制解调器 IrDA 寄存器（SCI_MODIR） 256
图 25-13：SCI FIFO 寄存器（SCI_FIFO） 258
图 25-14：SCI 水印寄存器（SCI_WATER） 261
图 25-15：SCI 过采样比寄存器（SCI_OSR） 262
图 25-16：SCI 波特率生成 263
图 26-1：SSI 方块图 274
图 26-2：SSI 应用图 274
图 26-3：控制寄存器 0（CTRLR0） 276
图 26-4：控制寄存器 1（CTRLR1） 279
图 26-5：SSI 启用寄存器（SSIENR） 280
图 26-6：微线控制寄存器（MWCR） 281
图 26-7：从属服务器启用寄存器（SER） 282
图 26-8：波特率选择（BAUDR） 283
图 26-9：传输 FIFO 阈值级别（TXFTLR） 284
图 26-10：接收 FIFO 阈值级别（RXFTLR） 285
图 26-11：传输 FIFO 液位寄存器（TXFLR） 286
图 26-12：接收 FIFO 级别寄存器（RXFLR） 287
图 26-13：状态寄存器（SR） 288
图 26-14：中断掩码寄存器（IMR） 289
图 26-15：中断状态寄存器（ISR） 291
图 26-16：原始中断状态寄存器（RISR） 292
图 26-17：传输 FIFO 溢出中断清除寄存器（TXOICR） 293
图 26-18：接收 FIFO 溢出中断清除寄存器（RXOICR） 294
图 26-19：接收 FIFO 下流中断清除寄存器（RXUICR） 295
图 26-20：中断清除寄存器（ICR） 296
图 26-21：DMA 控制寄存器（DMACR） 297
图 26-22：DMA 传输数据级别（DMATDLR） 298
图 26-23：DMA 接收数据级别（DMARDLR） 299
图 26-24：识别寄存器（IDR） 300
图 26-25：版本 ID 寄存器（VIDR） 301
图 26-26：SSI 数据寄存器（DRx） 302
图 26-27：RX 取样延迟寄存器（RXSDR） 303
图 26-28：SPI 控制寄存器 0（SPICTRLR0） 304
图 26-29：XIP 模式位（XIPMBR） 306
图 26-30：XIP 国际寄存器（XIPIIR） 307
图 26-31：XIP 封装初始寄存器（XIPWIR） 308
图 26-32：XIP 控制寄存器（XIPCR） 309
图 26-33：XIP 从属服务器启用寄存器（XIPSER） 311
图 26-34：XIP 接收 FIFO 溢出中断清除寄存器（XRXIOCR） 312
图 26-35：XIP 连续传输超时寄存器（列表列表） 313

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

图 26-36：SSI 配置为主设备.....314
图 26-37：典型的写入操作双/四/八进制 SPI 模式.....316
图 26-38：典型的读取操作双/四/八进制 SPI 模式.....316
图 26-39：XIP 传输与指令阶段.....317
图 27-1：I2C 方块图.....320
图 27-2：I2CS 寄存器显示 I2C 模块的状态。.....321
图 27-3：I2C 时钟计算前寄存器（I2CP）.....322
图 27-4：I2C 控制寄存器（I2CC）.....323
图 27-5：I2C 从属地址寄存器（I2CSA）.....325
图 27-6：I2C 端口控制寄存器（I2CPCR）.....325
图 27-7：I2C 从属高速模式指示器寄存器（I2CSHIR）.....326
图 27-8：I2C 从属 SDA 保持时间寄存器（I2CSHT）.....326
图 27-9：I2C 数据寄存器（I2CD）.....327
图 27-10：I2C 端口方向寄存器（I2CDDR）.....327
图 27-11：I2C 端口数据寄存器（I2CPDR）.....328
图 27-12：I2C 通信协议.....329
图 27-13：重复启动 I2C 协议.....330
图 27-14：SCL 同步.....331
图 27-15：HS 模式下的数据传输格式.....332
图 27-16：一个完整的 HS 模式传输.....332
图 27-17：从机模式初始化.....333
图 27-18：主模式初始化.....333
图 27-19：中断事务.....334
图 28-1：PWM 方块图.....337
图 28-2：PWM 比例前寄存器（PPR）.....339
图 28-3：PWM 时钟选择寄存器（PCSR）.....340
图 28-4：PWM 控制寄存器（PCR）.....341
图 28-5：PWM 计数器寄存器（PCNR）.....345
图 28-6：PWM 比较器寄存器（PCMR）.....347
图 28-7：PWM 输出.....348
图 28-8：PWM 占空比.....348
图 28-9：PWM 计时器寄存器（PTR）.....350
图 28-10：PWM 中断启用寄存器（PIER）.....351
图 28-11：PWM 中断标志寄存器（PIFR）.....352
图 28-12：PWM 捕获控制寄存器（PCCR0）.....353
图 28-13：PWM 捕获控制寄存器（PCCR1）.....354
图 28-14：PWM 捕获上升锁存寄存器（PCRLR0）.....355
图 28-15：PWM 捕获上升锁存寄存器（PCRLR1）.....356
图 28-16：PWM 捕获上升锁存寄存器（PCRLR2）.....356
图 28-17：PWM 捕获上升锁存寄存器（PCRLR3）.....357

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

图 28-18：PWM 捕获坠落锁存寄存器（PCFLR0）.....358
图 28-19：PWM 捕获坠落锁存寄存器（PCFLR13）.....358
图 28-20：PWM 捕获坠落锁存寄存器（PCFLR2）.....359
图 28-21：PWM 捕获坠落锁存寄存器（PCFLR3）.....359
图 28-22：PWM 端口控制寄存器（PPCR）.....360
图 28-23：PWM 双缓冲功能说明.....361
图 28-24：PWM 控制器输出负荷比.....361
图 28-25：死区生成操作.....362
图 28-26：捕获基本计时器操作.....363
图 29-1：比较器方块图.....364
图 29-2：比较器控制寄存器（CPTCN）.....366
图 29-3：比较器模式选择寄存器（CPTMD）.....367
图 29-4：比较器 MUX 选择寄存器（CPTMX）.....368
图 29-5：比较器输出过滤器选择寄存器（CPTFLS）.....369
图 29-6：比较器滞后图.....370
图 30-1：USB 全速设备(USB 模块）方块图.....372
图 30-2：USBPHY 控制寄存器 1（USBPHY_CTRL1）.....374
图 30-3：中断状态寄存器（INT_STAT）.....375
图 30-4：中断启用寄存器（INT_ENB）.....376
图 30-5：错误中断状态寄存器（ERR_STAT）.....378
图 30-6：错误中断启用寄存器（ERR_ENB）.....379
图 30-7：状态寄存器（STAT）.....381
图 30-8：控制寄存器（CTL）.....382
图 30-9：地址寄存器（ADDR）.....384
图 30-10：EBT 页面寄存器 1（EBT_PAGE_01）.....385
图 30-11：帧号寄存器（FRMNUML）.....386
图 30-12：帧号寄存器（FRMNUMH）.....387
图 30-13：令牌寄存器（TOKEN）.....388
图 30-14：SOF 阈值寄存器（SOF_THLD）.....389
图 30-15：EBT 页面寄存器 2（EBT_PAGE_02）.....390
图 30-16：EBT 页面寄存器 3（EBT_PAGE_03）.....391
图 30-17：端点控制寄存器（ENDPTn，n = 0-7）.....392
图 30-18：USBPHY 控制寄存器 2（USBPHY_CTRL2）.....393
图 30-19：USB PHY 观察寄存器（USB_PHY_OBSERVE）.....394
图 30-20：USB PHY GPIO 寄存器（USB_PHY_GPIO）.....395
图 30-21：USB 简历启用寄存器（USB_RESMEN）.....396
图 30-22：USB PHY 控制寄存器 3（USBPHY_CTRL3）.....397
图 30-23：USB PHY 控制寄存器 4（USBPHY_CTRL4）.....398
图 30-24：端点缓冲区表.....400
图 30-25：USB Token Transaction.....404

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

图 31-1：ADC 方块图.....407
图 31-2：启用/禁用 ADC.....408
图 31-3：ADC 时钟方案.....409
图 31-4：Analog to Digital 转换时间.....412
图 31-5：停止一个正在进行的转换.....413
图 31-6：一个序列的单一转换，软件触发器.....416
图 31-7：一个序列的连续转换，软件触发器.....416
图 31-8：一个序列的单一转换，硬件触发器.....417
图 31-9：一个序列的连续转换，硬件触发器.....417
图 31-10：模拟式监视器的监视区域.....421
图 31-11：ADC 中断和状态寄存器（ADC_ISR）.....423
图 31-12：ADC 中断启用寄存器（ADC_IER）.....425
图 31-13：ADC 控制寄存器（ADC_CR）.....426
图 31-14：ADC 配置寄存器 1（ADC_CFGR1）.....428
图 31-15：ADC 配置寄存器 2（ADC_CFGR2）.....430
图 31-16：ADC 采样时间寄存器（ADC_SMPR）.....431
图 31-17：ADC 看门狗寄存器（ADC_WDG）.....432
图 31-18：ADC 看门狗阈值寄存器（ADC_TR）.....433
图 31-19：ADC 通道选择寄存器 1（ADC_CHSELR1）.....434
图 31-20：ADC 通道选择寄存器 2（ADC_CHSELR2）.....434
图 31-21：ADC FIFO 访问寄存器（ADC_FIFO）.....435
图 32-1: VDD 上电要求时序图.....438
图 33-1：应用电路-8 位的 LT168AMCU 面板.....439
图 33-2：应用电路-SPI 面板的 LT168A.....440
图 33-3：应用电路-16 位的 LT168BMCU 面板.....441
图 33-4：应用电路-RGBTFT 面板的 LT168B.....442
图 33-5：应用电路-LT168B.....443
图 34-1：LT168A 外观尺寸图.....444
图 34-2：LT168B 外观尺寸图.....446
图 34-3：LT168B 底部焊盘 PCB 的设计建议-1.....447
图 34-4：LT168B 底部焊盘 PCB 的设计建议-2.....447

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

表 列 表

表 1-1：LT168 型号 31
表 2-1：引脚信号特性说明 42
表 2-2：引脚信号描述 46
表 2-3：LT168x 比较表 52
表 3-1：LT168B 是否外加 PSRAM 的显示功能差异 55
表 4-1：硬件模块与寄存器地址的配置图 63
表 5-1：中断控制器模块内存映射 64
表 5-2：优先值调整 71
表 5-3：优先值调整 72
表 5-4：优先值调整 73
表 5-5：中断来源分配 76
表 6-1：32 位 RISC 指令集 84
表 7-1：可编程定时器模块内存配置 87
表 8-1：CCM 内存映射 92
表 8-2：WKUPSEN 和相应的唤醒源 93
表 8-3：芯片引脚下拉配置 95
表 8-4：EPORT2 功能控制位 100
表 9-1：时钟内存映射 103
表 9-2：PLL 时钟分频器 105
表 9-3：ADC 时钟分频器 106
表 9-4：CLKOUTSEL 模式 107
表 9-5：睡眠模块中的睡眠操作控制位 107
表 9-6：PLL 输入分频数值 111
表 9-7：PLL VCO 输出时钟的除频数值 111
表 9-8：PLL 反馈的除频器数值 112
表 9-9：MS[29:0] 相应的模块 113
表 9-10：EPT 时钟分频器 114
表 10-1：复位控制器地址映像 122
表 10-2：复位源汇总 125
表 12-1：高速缓存模块内存映射 130
表 12-2：缓存集命令 141
表 12-3：缓存行命令 143
表 12-4：行命令结果 144
表 13-1：XBAR 寄存器配置摘要 146
表 14-1：DMA 通道分配 154
表 15-1：Efuse 寄存器内存映射 157
表 16-1：寄存器内存映射 160
表 16-2：可编程电压检测器 162

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

表 16-3：RGB 接口启用控制.....162
表 16-4：编程调试接口控制.....163
表 16-5：RSTOUT 禁用控制.....163
表 16-6：CLKOUT 禁用控制.....164
表 16-7：QSPI2 接口启用控制.....164
表 16-8：ADC 通道禁用控制.....170
表 17-1：RGB 控制器模块内存映射.....176
表 17-2：像素时钟分频器.....186
表 18-1：源数据格式.....189
表 18-2：目标数据格式.....189
表 18-3：Blender 模块内存映射.....189
表 19-1：信号属性.....195
表 19-2：显示 EBI 寄存器内存映射。.....195
表 19-3：EBI_CS#芯片选择等待状态编码.....197
表 19-4：EBI 控制引脚为 EPORT1.....200
表 19-5：EBI_WR#和 EBI_RD# Assert and Negate 时间.....203
表 19-6：芯片选择地址范围编码.....204
表 19-7：EBI_CS#8 位端口数据传输（大端）.....205
表 19-8：EBI_CS#16 位端口数据传输（大端）.....205
表 20-1：可编程中断定时器模块内存映射.....207
表 20-2：Prescaler Select Encoding.....209
表 20-3：PIT 中断请求.....212
表 21-1：看门狗计时器模块记忆地图.....214
表 21-2：看门狗计时器 Prescaler.....217
表 23-1：模块内存映射.....222
表 23-2：EPPAx Field Settings.....225
表 24-1：CANBus 信号.....232
表 25-1：信号属性.....236
表 25-2：SCI 模块记忆地图.....237
表 25-3：断开字符长度.....265
表 25-4：接收器唤醒选项.....266
表 26-1：SSI 内存映射.....275
表 27-1：I2C 内存映射.....320
表 28-1：PWM 信号描述.....337
表 28-2：模块内存映射.....338
表 28-3：计时器 3 时钟源选择.....340
表 29-1：比较器模块内存贴图.....365
表 29-2：比较器模式选择.....367
表 29-3：比较器负向输入 MUX 选择.....368
表 29-4：比较器正输入 MUX 选择.....368

©Levetop Semiconductor Co., Ltd.

表 30-1：USB 模块内存地图 373
表 30-2：端点启用/方向控制 393
表 30-3：USB PHY 振荡器模式选择 399
表 30-4：USB 目标设备的数据方向 400
表 30-5：EBT Address Calculation Fields 401
表 30-6：端点缓冲区表字节格式 402
表 30-7：端点缓冲区表字节字段 402
表 30-8：USB 响应的 DMA 覆盖错误 405
表 31-1：通道解码 410
表 31-2：正在配置触发器的极性。 413
表 31-3：配置触发器的极性 414
表 31-4：数据对齐和分辨率 418
表 31-5：模拟监控器通道选择 421
表 31-6：ADC 中断 421
表 31-7：ADC 模块内存映射 422
表 32-1：电气极限参数表 436
表 32-2：静电排放保护特性 436
表 32-3：IO 电气参数表(3.3V) 437
表 32-4：电源特性 437
表 32-5：功率消耗特性 437
表 32-6：唤醒时间特性 438
表 32-7：VDD 上电（Power Up）特性 438
表 34-1：封装尺寸参数 445
表 34-2：LT168B 封装尺寸参数 446

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

1. LT168 介绍

1.1. 基本简介

LT168 是一款串口屏 TFT 显示控制芯片。它包含一个 32 位 MCU 和图形显示模块，提供串口通信协议，支持 8 位或 16 位并口的 8080 MCU 接口或是 SPI/QSPI 串口接口，同时分辨率 800*480 以内的 TFT 屏。其中 LT168B 还支持 480*480 像素以内的 RGB 接口的 TFT 屏。

LT168 支持 乐升半导体 (Levetop Semiconductor) 的 UI 显示编辑软件 UI_Editor-II。它可以根据产品 TFT 显示面板的分辨率，快速地将显示素材（UI）及运作逻辑进行规划和导入到外部的闪存内。这可以帮助开发人员缩短使用 TFT 显示面板的产品的开发时间。在 UI 编辑软件“UI_Editor-II”进行导入设计和验证后，主控端 MCU 通过串行 Uart 接口与 LT168 进行通信，LT168 内部 Flash 的通信程序会根据 UI_Editor-II 的设计结果，对开发者从主机发出的串口命令进行解析，并将该命令所要显示的 UI 数据传到到 TFT 屏上。有关详细说明，请参阅“UI_Editor-II”应用说明。

LT168 嵌入了一个以最大 200MHz 频率运作的高性能 32 位 RISC 内核、高速内存（包括 768KB SRAM、8K 字节 ROM）、一个 32K 字节双向集关联缓存、支持高达 256M 字节的四信道 SPI 闪存接口，以及一系列增强型外设和 I/O 口。提供了标准通信接口（3 个主 QSPI、3 个 UART、1 个 USB2.0 F/S 控制器、1 个 I2C 和 CANBus 控制器）、8080 MCU 显示面板接口（支持 8080 总线，可与 MCU 面板连接）和 RGB 显示面板接口、定时器（最多 4 个通用 16 位定时器、8 个 PWM 定时器、1 个异步看门狗定时器和 1 个 RTC 定时器），ADC 模拟模块（1 个 1 MSPS ADC，带 8 通道和 2 个比较器）。LT168 还支持 Little VGL 的 GUI 图形库，具有良好的显示流畅性和优异的性价比。

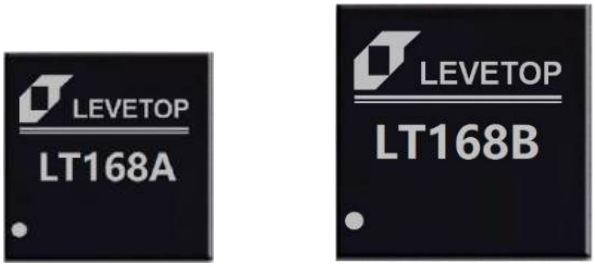
由于含有高容量的 Flash、SRAM 及众多 IO 接口，LT168A/B 也可以将部分资源作为主控的 MCU，将主控及 TFT 显示功能由一颗 LT168A/B 来完成，它的显示功能非常适合用在低分辨率 TFT-LCD 屏的电子产品上，如用来取代原单色屏产品，或是增加产品质感、档次，不会造成产品原主控端 MCU 太多的负担，可应用于各式车载周边、小家电、智能家电、手持控制设备、工业控制板、电子仪器、医疗设备、小型检测设备、小电摩、个人医美、充电设备、水电表、带屏智能音箱、洗脱机、机器人等产品。

LT168 的工作频率高达 200MHz。温度范围为-40°C 至 105°C，工作电压为 3.3V。LT168 有两个不同封装的型号，如下所示：

表 1-1：LT168 型号

型号	封装	内建 Flash	内建 SRAM	适用的 TFT 屏
LT168A	QFN48 (6*6 mm^2)	512K Bytes	256K Bytes + 512K Bytes	● 8-bits 8080 MCU 接口 TFT 屏 ● SPI 接口 TFT 屏
LT168B	QFN68 (8*8 mm^2)	2M Bytes	256K Bytes + 512K Bytes	● 8/16-bits 8080 MCU 接口 TFT 屏 (Max. 800*480) ● RGB 565 接口 TFT 屏 (Max. 480*480) ● SPI/QSPI 接口 TFT 屏

©Levetop Semiconductor Co., Ltd.



QFN48(6*6 mm^2) QFN68(8*8 mm^2)

图 1-1：LT168A 和 LT168B 外观图

1.2. 内部方块图

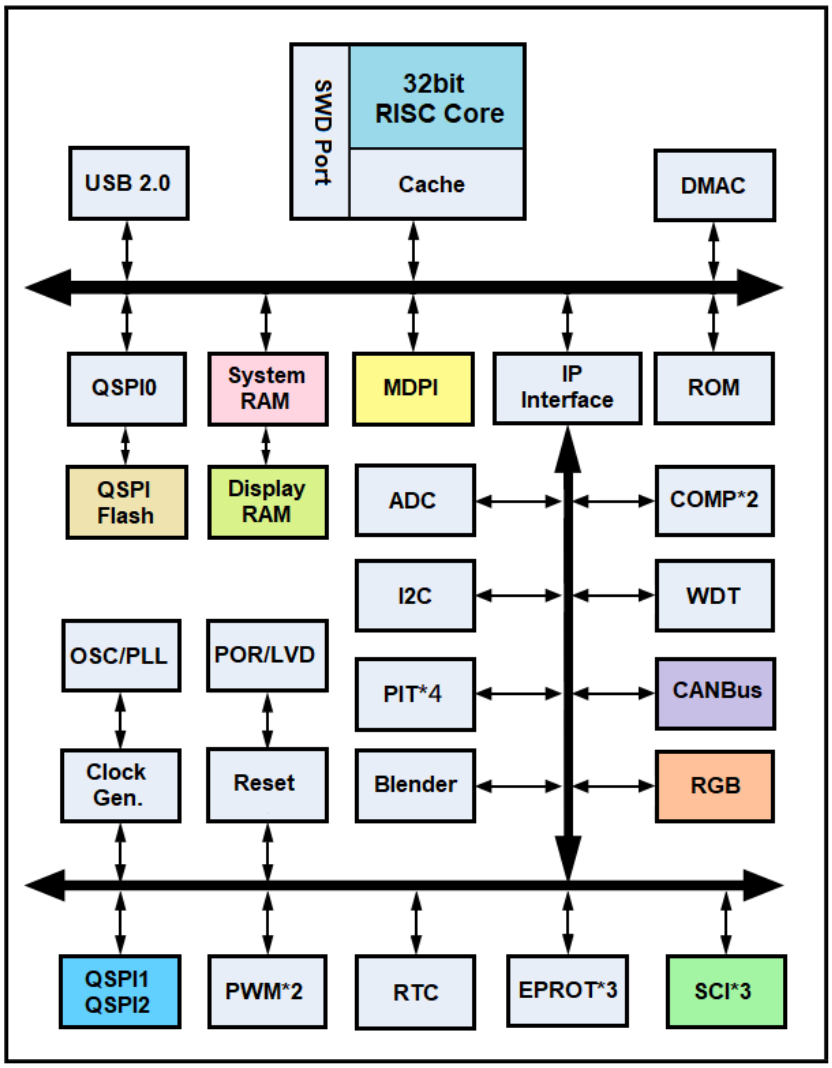


图 1-2：内部方块图

©Levetop Semiconductor Co., Ltd.

1.3. 功能说明

1.3.1. 32 位 RISC 处理器内核

- 32 位负载/存储减少指令集计算机（RISC）架构，具有固定的 16 位指令长度
- 16 条目 32 位通用寄存器文件
- 高效的 3 阶段执行管道，隐藏在应用软件中
- 单指令执行多周期指令，分支三个周期
- 支持字节半字节内存访问
- 嵌入式中断控制器，支持嵌套的向量中断
- 单周期 32 位 x32 位硬件整数乘法器阵列
- 3 个~13 个周期的硬件整数分法器阵列

1.3.2. 内建 768K Bytes 静态内存 SRAM

- 单周期字节、半字（16 位）和字（32 位）读写
- 分两个部分，以提高在特定的应用程序中的性能
  - 系统 RAM：256K 字节，地址范围从 0x800000 到 0x83FFFF
  - 显示 RAM：512K 字节，地址范围从 0x840000 到 0x8BFFFF

1.3.3. 内建 QSPI 闪存

– LT168A 内建 512K Bytes QSPI Flash，支持 乐升半导体 串口通讯软件
– LT168B 内建 2M Bytes QSPI Flash，支持 乐升半导体 串口通讯软件

_ 工程样片内含有 Bootloader 引导程序，批量出货的则不含 Bootloader（空白片）

1.3.4. 8K 字节的 ROM

- 单周期字节、半字（16 位）和字（32 位）读取访问

1.3.5. 32K 字节的高速缓存

- 双向集合关联组织
- 两个 AHB 总线接口，一个主接口和一个从接口

1.3.6. 外部总线接口

- 可编程等待状态-在访问终止之前，可以编程多达 16 个等待状态
- 一个可编程的异步主动低芯片选择。
- 可编程芯片选择等待周期
  - 为了与各种面板接口，最多 16 个芯片选择断言的周期可以被编程
- 可编程的读/写断言周期
- 可编程的读写否定循环
- 支持 8/16 位的端口大小。
- 支持 8080 标准总线
- 支持 8080 MCU 型 TFT 面板：

©Levetop Semiconductor Co., Ltd.

- 800 x 480
- 480 x 272
- 480 x 320
- 320 x 240 或以下

1.3.7. RGB 显示面板接口

- 数据帧格式：红色：5，绿色：6，蓝色：5
- 支持的分辨率：
  - 480 x 480
  - 480 x 272 或以下
- 接口信号的定时是可配置的
  - 极性和像素时钟极性是可配置的

1.3.8. DMA 控制器

- 32 个独立的可编程 DMA 控制器通道
- 8、16、32 位的数据传输
- 支持单次转移，突发 4、8、16 转移，并在特殊情况下爆发。
- 支持单个周期的转移
- 支持自动转换模式
- 支持 LLI 传输模式
- 遵循一个固定的优先级规则

1.3.9. 复位功能

- 复位电路上的内部电源
- 5 个复位源：
  - 通电复位
  - 外部 RESET#复位引脚
  - 软件复位
  - 监视器定时器
  - 程序电压检测复位
- 状态标志表示上次复位的来源

1.3.10. 四周 PIT 定时器

- 16 位计数器与模数器“初始计数”寄存器
- 可选择为自由运行或倒计时
- 16 个可选预压器-2^0 至 2^{15}
- 支持 DMA 接口

1.3.11. 看门狗定时器（Watchdog Timer）

– 16 位计数器与模量“初始计数”寄存器
- 在低功耗模式下的暂停选项

©Levetop Semiconductor Co., Ltd.

Levetop Semiconductor

- 最多可提供 2000 ms 的服务时间

1.3.12. Real Time Clock 时钟

- 支持将时间数据加载到秒计数器和从秒、分钟、小时和天计数器读取时间数据
- 支持报警设置
- 中断源：
  - 秒，分，时，日 中断
  - 可编程报警中断
  - 1 KHz/32 KHz 周期性中断

1.3.13. GPIO 与中断接口（EPORT)

- 每个 EPORT 有 8 个通道
- 上升/下降边缘选择
- 低/高级别敏感
- 可配置为通用 I/O 的中断引脚

1.3.14. 四线串口 SPI 模块（QSPI)

- 串行主操作
- DMA 控制器接口
- 使 SSI 通过传输请求的握手接口通过总线接口到 DMA 控制器。
- 在增强的 SPI 传输中的时钟拉伸支持
- 数据项目大小（4 至 32 位）-每个数据传输的项目大小由程序员控制
- 可配置的传输和接收 FIFO 缓冲区的深度从 2 到 256 字的深度。FIFO 的宽度固定在 32 位
- 增强的 SPI 支持
- 就地执行（XIP）模式支持

1.3.15. 串口通讯模块（SCI / Uart)

- 全双工，标准非归零（NRZ）格式
- 可编程波特率（13 位模分频器），可配置的过采样比从 4 倍到 256 倍
- 中断，轮询操作
  - 传输数据寄存器为空，传输完成
  - 接收数据寄存器已满
  - 接收溢出错误、奇偶校验错误、帧错误和噪声错误
  - 怠速接收器检测
  - 接收引脚上的活动边缘
  - 断路检测支持 LIN
  - 接收数据匹配
- 硬件奇偶校验的生成和检查
- 可编程的 8 位、9 位或 10 位字符长度
- 可编程的 1 位或 2 位停止位
- 三种接收器唤醒方法：

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

  - 怠速线路唤醒
  - 地址标记唤醒
  - 接收数据匹配
- 自动地址匹配，以减少 ISR 开销：
  - 地址标记匹配
  - 空闲线路地址匹配
  - 地址匹配开始，地址匹配结束
- 可选的 13 位中断字符生成/11 位中断字符检测
- 可配置的空闲长度检测，支持 1、2、4、8、16、32、64 或 128 个空闲字符
- 可选择的发送器输出和接收器输入极性
- 可选择的 IrDA 1.4 返回到零反转（RZI）格式，具有可编程脉冲宽度
- 独立于“传输”和“接收”的 FIFO 结构
  - “接收”和“传输”请求的可配置水印
  - 如果“接收”FIFO 不是空，接收方在可配置的空闲字符之后断言请求的选项

1.3.16. Canbus 控制器

- 完全实现了 CAN 协议规范，版本 2.0B
  - 标准数据和远程帧
  - 扩展的数据和远程帧
  - 0-8 字节数据长度
  - 可编程比特率，最高可达 1 Mbit/s
  - 内容相关的寻址
- 64 个数据长度为 0 到 8 个字节的消息缓冲区
- 每个 MB 可配置为 Rx 或 Tx，所有支持标准和扩展消息
- 每个消息缓冲区的单独的 Rx 掩码寄存器
- 包括 1056 字节（64MB）的 SRAM（用于 MB 存储）
- 包括 256 字节（64 MBs）的 SRAM 用于单个 Rx 掩码寄存器
- 全功能的 Rx FIFO，存储容量为 6 帧和内部指针处理
- 强大的 Rx FIFO ID 过滤，能够匹配 8 个扩展的 ID，16 个标准的或 32 个部分（8 位）的 ID，具有单独的掩蔽能力
- 可编程时钟源到 CAN 协议接口，或总线时钟或晶体振荡器
- 未使用的 MB 和 Rx 掩码寄存器空间可用作通用的 SRAM 空间
- 只听模式功能
- 可编程回路返回模式，支持自检操作
- 可编程传输优先级方案：最低 ID、最低缓冲区号或最高优先级
- 基于 16 位自由运行的计时器的时间戳
- 全局网络时间，由一个特定的消息进行同步
- 可移动的中断
- 独立于传输介质（假设有一个外部收发器）
- 由于针对高优先级消息的仲裁方案而导致的延迟时间较短
- 低功耗模式
- Tx 消息缓冲区上的硬件取消

©Levetop Semiconductor Co., Ltd.

1.3.17. USB2.0 Full Speed 控制器

- 支持内部参考时钟或外部 12MHz 晶体参考时钟
- 符合 USB2.0 全速规格，芯片上集成 PHY 模块
- 支持 FS（12Mbps）模式
- 支持最多支持 8 个端点，包括端点 0
- 除端点 0 外，所有端点都可以支持中断和批量传输
- 除端点 0 外，所有端点都可以配置为 8、16、32、64 字节的 FIFO 大小
- 终点 0 支持 s 的控制传输

1.3.18. 双脉宽调制器（PWM）

- 每个 PWM 控制器有四个通道
- 可编程时间
- 可编程占空比
- 两个死区发电机
- 捕获功能
- 引脚可以配置为通用的 I/O

1.3.19. 8 通道 ADC 转换器

- 高性能
  - 12 位、10 位、8 位或 6 位的可配置分辨率
  - ADC 转换时间：12 位分辨率 1.0μs（1MHz），10 位分辨率 0.88 μs，通过降低分辨率可以获得更快的转换时间。
  - 可编程采样时间
  - 具有内置数据一致性的数据对齐
  - DMA 支持
- 低功耗
  - 低功率操作可以降低 PLCK 频率，同时仍然保持最佳的 ADC 性能。例如，无论 PL CK 的频率如何，都保持 1.0 μs 的转换时间。
  - 等待模式：以防止在低频 PLCK 下的应用程序中的 ADC 溢出
  - 自动关闭模式：ADC 会自动关闭，除非在活动转换阶段关闭 ADC。这大大降低了 ADC 的功耗。
- 模拟输入通道
  - 8 个外部模拟输入
- 可以启动转换的开始时间：
  - 通过软件
  - 通过具有可配置极性的硬件触发器
- 转换模式

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

- 可以转换单个通道或可以扫描一系列信道。
- 单模式为每个触发器转换选定的输入一次
- 连续模式可连续地转换选定的输入
- 不连续模式

- 在采样结束、转换结束、序列转换结束以及发生模拟监视器或溢出事件时中断生成。
- 模拟监视器
- 单端和差分输入配置转换器使用内部参考或外部参考

1.3.20. 两个模拟比较器（Comparator）

- 可编程响应时间
- 可编程滞后
- 支持模拟输入多路复用器，有九个选择
- 两个可选输出：已过滤输出或异步输出
- 可选择的上下边缘中断

1.3.21. 电源管理单元（PMU）

- 支持芯片上的 1.2V LDO，最大负载电流为 100 mA
- 1.2V LDO 支持两种模式：低功率、高功率

1.3.22. 压检测器

- 可编程电压检测器

1.3.23. 内部时钟

- 128 KHz 片上振荡器时钟为看门狗和 PMU
- PLL 时钟，可用于系统时钟
- 48MHz USB PLL 时钟，可用于 USB SIE

1.3.24. 外部晶振时钟

- 高达 20Mhz 的外部晶体振荡器时钟，可用于系统时钟
- 32.768Khz 外部晶体振荡器时钟，可用于 RTC

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

1.4. 系统方块

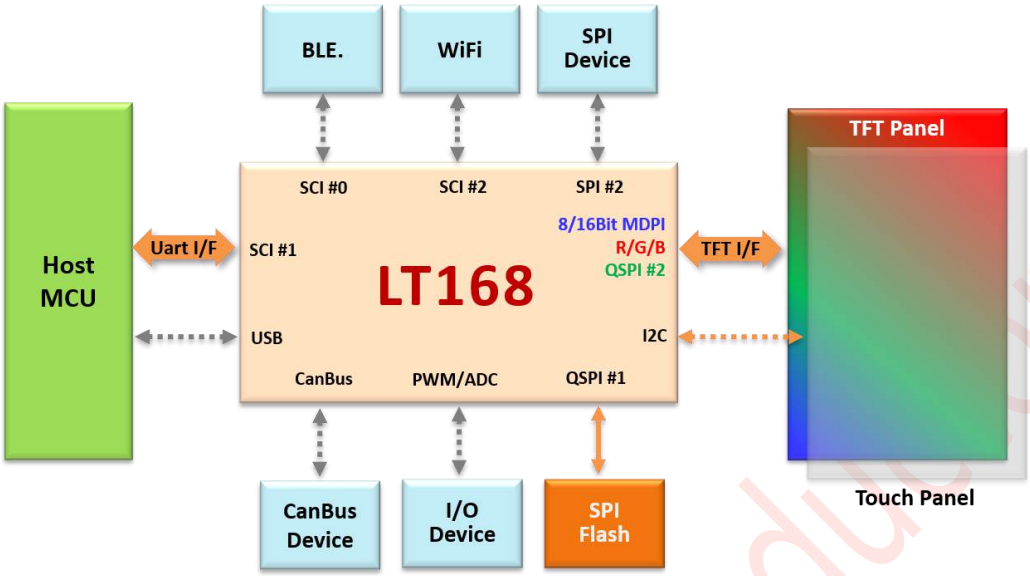


图 1-3：LT168 的系统方块图

1.5. 模块应用

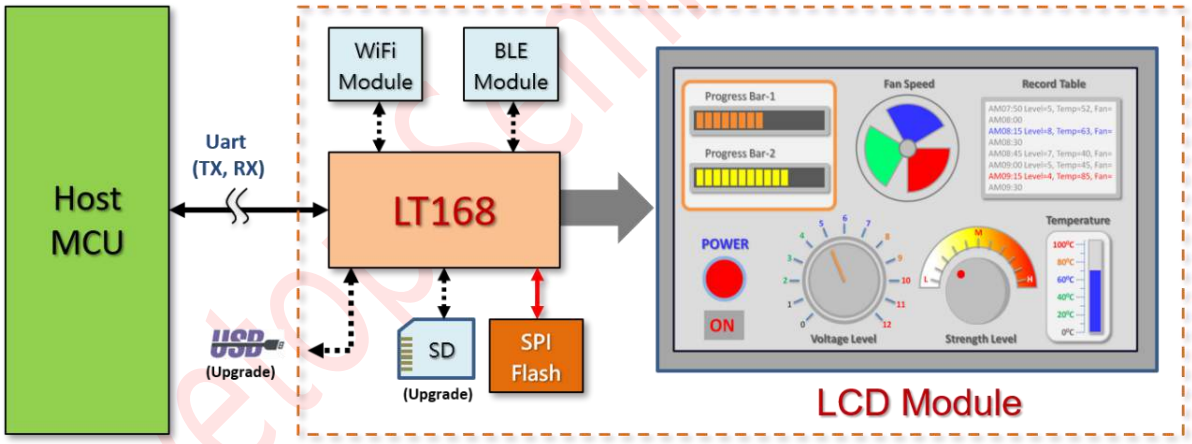


图 1-4：TFT 液晶显示器模块应用示图

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

2. 信号描述

LT168 有两种封装及型号，分别如下：

- QFN48 (6*6 mm^2) – LT168A
- QFN68 (8*8 mm^2) – LT168B

2.1. 芯片脚位图

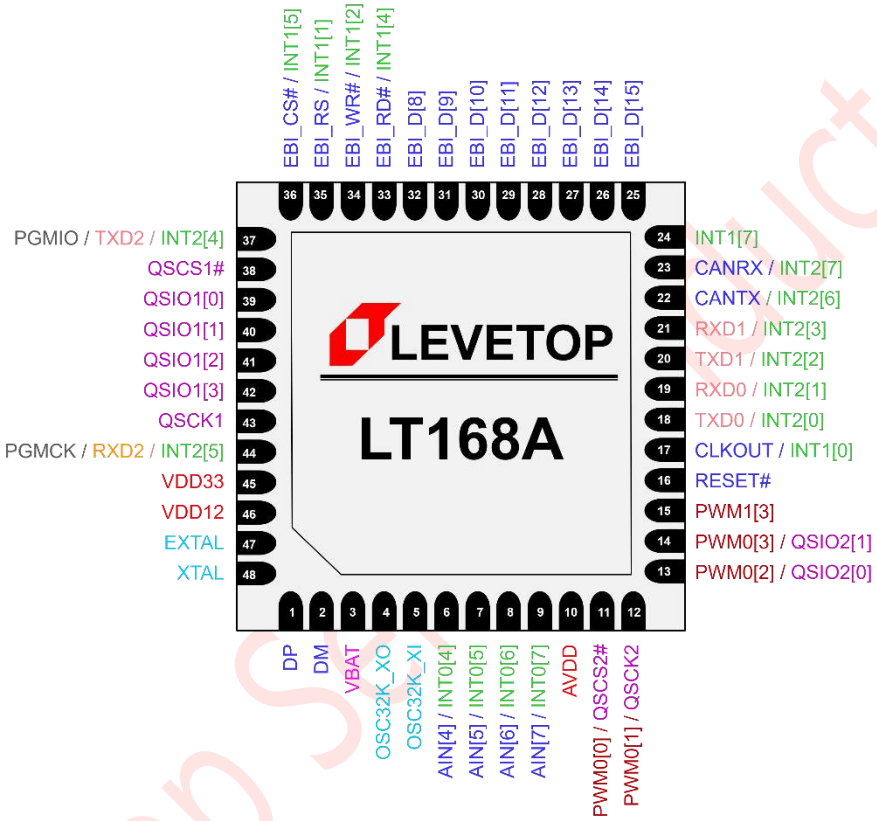


图 2-1：LT168A（QFN48）脚位图

©Levetop Semiconductor Co., Ltd.

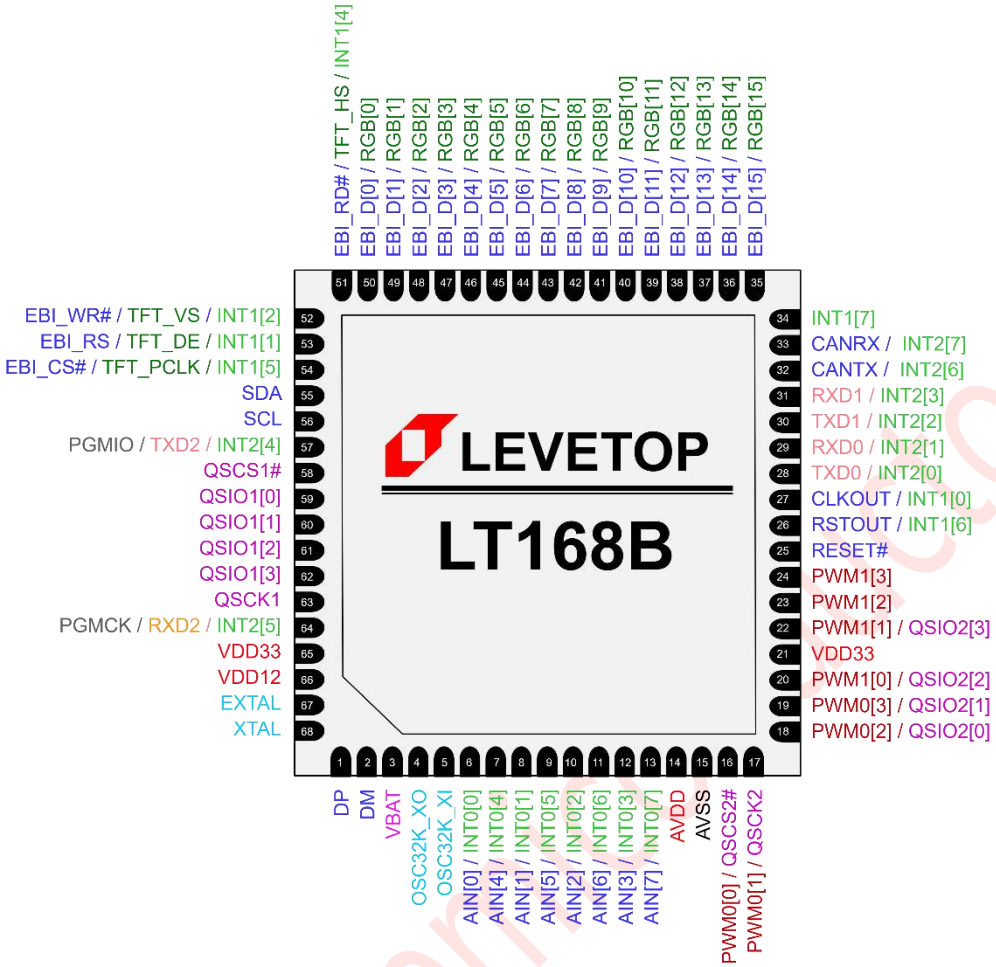


图 2-2：LT168B（QFN68）脚位图

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

2.2. 引脚信号特性

表 2-1：引脚信号特性说明

引脚名称	复用信号 1	复用信号 2	脚号		数量	方向	默认值 Dir *2	Pullup *3	IO 类型*4
			LT168A	LT168B					
SCI (6)									
RXD0 *1	INT2[1]	-	19	29	1	I/O	I	PullUp	PBCUL16R
TXD0	INT2[0]	-	18	28	1	I/O	O(H)	-	PBCUL16R
RXD1	INT2[3]	-	21	31	1	I/O	I	PullUp	PBCUL16R
TXD1	INT2[2]	-	20	30	1	I/O	O(H)	-	PBCUL16R
RXD2	PGMCK	INT2[5]	44	64	1	I/O	I	PullUp	PBCUL16R
TXD2	PGMIO	INT2[4]	37	57	1	I/O	I	PullUp	PBCUL16R
USB (2)									
DP	-	-	1	1	1	Analog	Hiz	-	-
DM	-	-	2	2	1	Analog	Hiz	-	-
I2C (2)									
SCL	GPIO_SCL	-	-	56	1	I/O	I	PullUp	PBCUL16R
SDA	GPIO_SDA	-	-	55	1	I/O	I	PullUp	PBCUL16R
QSPI (18)									
QSIO2[3]	PWM1[1]	-	-	22	1	I/O	I	PullDown	PBCD24R
QSIO2[2]	PWM1[0]	-	-	20	1	I/O	I	PullDown	PBCD24R
QSIO2[1]	PWM0[3]	-	14	19	1	I/O	I	PullDown	PBCD24R
QSIO2[0]	PWM0[2]	-	13	18	1	I/O	I	PullDown	PBCD24R
QSCK2	PWM0[1]	-	12	17	1	I/O	I	PullDown	PBCD24R
QSCS2#	PWM0[0]	-	11	16	1	I/O	I	PullDown	PBCD24R
QSIO1[3]	-	-	42	62	1	I/O	I	PullUp	PBCU24R
QSIO1[2]	-	-	41	61	1	I/O	I	PullUp	PBCU24R
QSIO1[1]	-	-	40	60	1	I/O	I	PullUp	PBCU24R
QSIO1[0]	-	-	39	59	1	I/O	I	PullUp	PBCU24R
QSCK1	-	-	43	63	1	O	O(L)	-	PBCU24R
QSCS1#	-	-	38	58	1	O	O(H)	-	PBCU24R
QSIO0[3] *5	-	-	-	-	1	I/O	I	PullUp	PBCU24R
QSIO0[2]	-	-	-	-	1	I/O	I	PullUp	PBCU24R
QSIO0[1]	-	-	-	-	1	I/O	I	PullUp	PBCU24R
QSIO0[0]	-	-	-	-	1	I/O	I	PullUp	PBCU24R
QSCK0	-	-	-	-	1	O	O(L)	-	PBCU24R
QSCS0#	-	-	-	-	1	O	O(H)	-	PBCU24R
PWM0 (4)									
PWM0[3]	QSIO2[1]	-	14	19	1	I/O	I	PullDown	PBCD24R

©Levetop Semiconductor Co., Ltd.

引脚名称	复用信号 1	复用信号 2	脚号		数量	方向	默认值 Dir *2	Pullup *3	IO 类型*4
			LT168A	LT168B					
PWM0[2]	QSIO2[0]	-	13	18	1	I/O	I	PullDown	PBCD24R
PWM0[1]	QSCK2	-	12	17	1	I/O	I	PullDown	PBCD24R
PWM0[0]	QSCS2#	-	11	16	1	I/O	I	PullDown	PBCD24R
PWM1 (4)									
PWM1[3]	-	-	15	24	1	I/O	I	PullDown	PBCD24R
PWM1[2]	-	-	-	23	1	I/O	I	PullDown	PBCD24R
PWM1[1]	QSIO2[3]	-	-	22	1	I/O	I	PullDown	PBCD24R
PWM1[0]	QSIO2[2]	-	-	20	1	I/O	I	PullDown	PBCD24R
ADC (8)									
AIN[7]	INT0[7]	-	9	13	1	Analog	Hiz	-	PVDD1ANPR
AIN[6]	INT0[6]	-	8	11	1	Analog	Hiz	-	PVDD1ANPR
AIN[5]	INT0[5]	-	7	9	1	Analog	Hiz	-	PVDD1ANPR
AIN[4]	INT0[4]	-	6	7	1	Analog	Hiz	-	PVDD1ANPR
AIN[3]	INT0[3]	-	-	12	1	Analog	Hiz	-	PVDD1ANPR
AIN[2]	INT0[2]	-	-	10	1	Analog	Hiz	-	PVDD1ANPR
AIN[1]	INT0[1]	-	-	8	1	Analog	Hiz	-	PVDD1ANPR
AIN[0]	INT0[0]	-	-	6	1	Analog	Hiz	-	PVDD1ANPR
Edge Port 0 (8)									
INT0[7]	AIN[7]	-	9	13	1	I/O	Hiz	-	PBCUL16R
INT0[6]	AIN[6]	-	8	11	1	I/O	Hiz	-	PBCUL16R
INT0[5]	AIN[5]	-	7	9	1	I/O	Hiz	-	PBCUL16R
INT0[4]	AIN[4]	-	6	7	1	I/O	Hiz	-	PBCUL16R
INT0[3]	AIN[3]	-	-	12	1	I/O	Hiz	-	PBCUL16R
INT0[2]	AIN[2]	-	-	10	1	I/O	Hiz	-	PBCUL16R
INT0[1]	AIN[1]	-	-	8	1	I/O	Hiz	-	PBCUL16R
INT0[0]	AIN[0]	-	-	6	1	I/O	Hiz	-	PBCUL16R
Edge Port 1 (7)									
INT1[7]	-	-	24	34	1	I/O	I	PullUp	PBCUL16R
INT1[6]	RSTOUT	-	-	26	1	I/O	O	-	PBCUL16R
INT1[5]	EBI_CS#	TFT_PCLK	36	54	1	I/O	O(H)	-	PBCU24R
INT1[4]	EBI_RD#	TFT_HS	33	51	1	I/O	O(H)	-	PBCU24R
INT1[2]	EBI_WR#	TFT_VS	34	52	1	I/O	O(H)	-	PBCU24R
INT1[1]	EBI_RS	TFT_DE	35	53	1	I/O	O(L)	-	PBCU24R
INT1[0]	CLKOUT	-	17	27	1	I/O	O	-	PBCUL16R
Edge Port 2 (8)									
INT2[7]	CANRX	-	23	33	1	I/O	I	PullUp	PBCUL16R

©Levetop Semiconductor Co., Ltd.

引脚名称	复用信号 1	复用信号 2	脚号		数量	方向	默认值 Dir *2	Pullup *3	IO 类型*4
			LT168A	LT168B					
INT2[6]	CANTX	-	22	32	1	I/O	O(H)	-	PBCUL16R
INT2[5]	RXD2	PGMCK	44	64	1	I/O	I	PullUp	PBCUL16R
INT2[4]	TXD2	PGMIO	37	57	1	I/O	I	PullUp	PBCUL16R
INT2[3]	RXD1	-	21	31	1	I/O	I	PullUp	PBCUL16R
INT2[2]	TXD1	-	20	30	1	I/O	O(H)	-	PBCUL16R
INT2[1]	RXD0	-	19	29	1	I/O	I	PullUp	PBCUL16R
INT2[0]	TXD0	-	18	28	1	I/O	O(H)	-	PBCUL16R
Programming Port (2)									
PGMCK	RXD2	INT2[5]	44	64	1	I/O	I	PullUp	PBCUL16R
PGMIO	TXD2	INT2[4]	37	57	1	I/O	I	PullUp	PBCUL16R
CLOCK (5)									
EXTAL	-	-	47	67	1	I/O	I	-	PXWE2R
XTAL	-	-	48	68	1	I/O	O	-	PXWE2R
OSC32K_XI	-	-	5	5	1	I/O	I	-	PVDD1ANPR
OSC32K_XO	-	-	4	4	1	I/O	O	-	PVDD1ANPR
CLKOUT	INT1[0]	-	17	27	1	I/O	O	-	PBCUL16R
RESET (2)									
RESET#	-	-	16	25	1	-	I	PullUp	PISUR
RSTOUT	INT1[6]	-	-	26	1	I/O	O	-	PBCUL16R
CAN (2)									
CANRX	INT2[7]	-	23	33	1	I/O	I	PullUp	PBCUL16R
CANTX	INT2[6]	-	22	32	1	I/O	O(H)	-	PBCUL16R
EBI (20)									
EBI_CS# *6	TFT_PCLK	INT1[5]	36	54	1	I/O	O(H)	-	PBCU24R
EBI_RS	TFT_DE	INT1[1]	35	53	1	I/O	O(L)	-	PBCU24R
EBI_RD#	TFT_HS	INT1[4]	33	51	1	I/O	O(H)	-	PBCU24R
EBI_WR#	TFT_VS	INT1[2]	34	52	1	I/O	O(H)	-	PBCU24R
EBI_D[15]	TFT_RGB[15]	GPIO_D[15]	25	35	1	I/O	I	PullUp	PBCU24R
EBI_D[14]	TFT_RGB[14]	GPIO_D[14]	26	36	1	I/O	I	PullUp	PBCU24R
EBI_D[13]	TFT_RGB[13]	GPIO_D[13]	27	37	1	I/O	I	PullUp	PBCU24R
EBI_D[12]	TFT_RGB[12]	GPIO_D[12]	28	38	1	I/O	I	PullUp	PBCU24R
EBI_D[11]	TFT_RGB[11]	GPIO_D[11]	29	39	1	I/O	I	PullUp	PBCU24R
EBI_D[10]	TFT_RGB[10]	GPIO_D[10]	30	40	1	I/O	I	PullUp	PBCU24R
EBI_D[9]	TFT_RGB[9]	GPIO_D[9]	31	41	1	I/O	I	PullUp	PBCU24R
EBI_D[8]	TFT_RGB[8]	GPIO_D[8]	32	42	1	I/O	I	PullUp	PBCU24R
EBI_D[7]	TFT_RGB[7]	GPIO_D[7]	-	43	1	I/O	I	PullUp	PBCU24R

©Levetop Semiconductor Co., Ltd.

引脚名称	复用信号 1	复用信号 2	脚号		数量	方向	默认值 Dir *2	Pullup *3	IO 类型*4
			LT168A	LT168B					
EBI_D[6]	TFT_RGB[6]	GPIO_D[6]	-	44	1	I/O	I	PullUp	PBCU24R
EBI_D[5]	TFT_RGB[5]	GPIO_D[5]	-	45	1	I/O	I	PullUp	PBCU24R
EBI_D[4]	TFT_RGB[4]	GPIO_D[4]	-	46	1	I/O	I	PullUp	PBCU24R
EBI_D[3]	TFT_RGB[3]	GPIO_D[3]	-	47	1	I/O	I	PullUp	PBCU24R
EBI_D[2]	TFT_RGB[2]	GPIO_D[2]	-	48	1	I/O	I	PullUp	PBCU24R
EBI_D[1]	TFT_RGB[1]	GPIO_D[1]	-	49	1	I/O	I	PullUp	PBCU24R
EBI_D[0]	TFT_RGB[0]	GPIO_D[0]	-	50	1	I/O	I	PullUp	PBCU24R
Power Supply									
VDD33	-	-	45	21, 65	2	P	-	-	PVDD2R
VDD12	-	-	46	66	1	P	-	-	PWR
AVDD	-	-	10	14	1	P	-	-	PVDD1ANPR
VBAT	-	-	3	3	1	P	-	-	PWR
VSS	-	-	49*7	69*7	1	G	-	-	PWR
AVSS	-	-	-	15	1	G	-	-	PWR

提示：

1. 红色信号为复用信号中默认的引脚名称。
2. "Default Dir" i 是指复位后的驱动方向。“I”代表输入，“O”代表输出，“O（H）”代表输出高，“O（L）”代表输出低，“Hiz”代表输入和输出全部禁用，上拉/下拉电阻也被禁用。
3. 当信号被程序设计为输出时，所有上拉和下拉电阻都是断开。
4. 输出的 I/O 驱动：
   - PBCU24R = CMOS Tri-state output pad with controllable input and controllable pull-up,
   - PBCUL16R = CMOS Tri-state output pad with controllable input and controllable pullup and limited slew rate,
   - PBCD24R = CMOS Tri-state output pad with enable controlled input and enable controlled pull down,
   - PISUR = schmitt trigger enable controlled input pad with pull-up,
   - PVDD1ANPR = VDD analog PAD with digital power domain,
   - PXWE2R = Crystal oscillator with internal resistor and active high enable,
   - PVDD2R = VDD power pad for I/O post driver,
   - The suffix of cell means the drive strength and x can be 2, 16 and 24. For example, PBCU24R means the drive strength is 24mA.
5. 这些 QSPI0 信号直接连接到芯片内部 SPI 闪存。
6. EBI Bus 的控制信号线为专用的硬件接口，不能用其他 GPIO 引脚替代。
7. 这是 LT168 的背面散热焊盘，它必须直接接地（VSS/GND）。

2.3. 信号说明

本节提供了对这些信号的简要描述。有关更多详细信息，请参阅特定模块部分。

©Levetop Semiconductor Co., Ltd.

表 2-2：引脚信号描述

信号名称	引脚号		引脚说明
	LT168A	LT168B	
串口模块信号 0 (Serial Communications Interface – 0，SCI0)			
RXD0	19	29	**SCI 接收数据** 该信号用于 SCI0 接收器数据输入，当未配置为接收器操作时，也可用于 GPIO（INT2[1]）信号。
TXD0	18	28	**SCI 传输数据** 该信号用于 SCI0 发送器数据输出，当未配置为发送器操作时，也可用于 GPIO（INT2[0]）信号。
串口模块信号 1 (SCI1)			
RXD1	21	31	**SCI 接收数据** 该信号用于 SCI1 接收器数据输入，当未配置为接收器操作时，也可用于 GPIO（INT2[3]）信号。
TXD1	20	30	**SCI 传输数据** 该信号用于 SCI1 发送器数据输出，当未配置为发送器操作时，也可用于 GPIO（INT2[2]）信号。
串口模块信号 2 (SCI2)			
RXD2	44	64	**SCI 接收数据** 该信号用于 SCI2 接收器数据输入，当未配置为接收器操作时，也可用于 GPIO（INT2[5]）或 PGMCK 信号。
TXD2	37	57	**SCI 传输数据** 该信号用于 SCI2 发送器数据输出，当未配置为发送器操作时，也可用于 GPIO（INT2[4]）或 PGMIO 信号。
CAN Bus 模块信号			
CANRX	23	33	**Can Bus 接收数据** 该引脚是来自 CANBus 收发器的接收引脚。显性状态由逻辑电平“0”表示。隐性状态由逻辑电平“1”表示。 当未配置为 Canbus 操作时，此信号也可用于 GPIO（INT2[7]）信号。
CANTX	22	32	**Can Bus 传输数据** 该引脚是 CANBus 收发器的发送引脚。显性状态由逻辑电平“0”表示。隐性状态由逻辑电平“1”表示。 当未配置为 Canbus 操作时，该信号也可用于 GPIO（INT2[6]）信号。

©Levetop Semiconductor Co., Ltd.

信号名称	引脚号		引脚说明
	LT168A	LT168B	
RGB 显示屏接口信号 这些信号仅适用于 LT168B，用来驱动 RGB 接口型的 LCD 屏。			
TFT_PCLK	-	54	**TFT LCD 屏幕扫描时钟信号** 屏幕扫描时钟信号连接至通用的 RGB 屏驱动接口讯号。
TFT_HS	-	51	**TFT LCD 水平同步信号** 水平同步信号 TFT_HS 连接至通用的 RGB 屏接口讯号，用来表示显示画面上一条水平线（Line）的开始。
TFT_VS	-	52	**TFT LCD 垂直同步信号** 垂直同步信号 TFT_VS 连接至通用的 RGB 屏接口讯号，用来表示显示画面上一帧（Frame）图片的开始。
TFT_DE	-	53	**TFT LCD 屏幕数据使能** 此信号为连接至通用 RGB 屏接口的数据有效或数据使能信号。
TFT_RGB[15:0]	-	35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50	**TFT LCD 屏幕 RGB 数据** 这些是连接至通用 RGB 屏接口的红色、绿色和蓝色数据输出信号。
外部总线（External Bus Interface，EBI）信号 外部总线接口负责控制内部总线和外部 MCU 显示屏间的信息传输。LT168A 支持 8 位 EBI，LT168B 支持 8 位或 16 位 EBI。			
EBI_CS#	36	54	**EBI 片选信号（8080 MCU 屏片选信号）** LT168 提供外部总线接口（External Bus Interface (EBI）来驱动 8080 MCU 接口的 LCD 显示面板，该信号是外部总线接口的片选信号。也就是当使用 8/16Bit 并口的 MCU 屏时，此信号为 LT168 对外部 8/16Bit MCU 屏的片选控制信号。
EBI_RS	35	53	**EBI 寄存器选择信号** 当使用 8/16Bit 并口的 MCU 屏时，此信号连接到 MCU 屏 RS 或 A0 信号。
EBI_RD#	33	51	**EBI 数据读取控制信号** 当使用 8/16Bit 并口的 MCU 屏时，此信号为 LT168 对外部 8/16Bit MCU 屏的数据读取控制信号。
EBI_WR#	34	52	**EBI 数据写入控制信号** 当使用 8/16Bit 并口的 MCU 屏时，此信号为 LT168 对外部 8/16Bit MCU 屏的数据写入控制信号。

©Levetop Semiconductor Co., Ltd.

信号名称	引脚号		引脚说明
	LT168A	LT168B	
EBI_D[15:8]	25, 26, 27, 28, 29, 30, 31, 32	35, 36, 37, 38, 39, 40, 41, 42	**EBI 高位数据信号** 当使用 16Bit 并口的 MCU 屏时，这些信号为 LT168 对外部 16Bit MCU 屏的数据 Bit15~Bit8 传输信号。当使用 8Bit 并口的 MCU 屏时，这些信号为 LT168 对外部 8Bit MCU 屏的数据 Bit7~Bit0 传输信号。
EBI_D[7:0]	-	43, 44, 45, 46, 47, 48, 49, 50	**EBI 低位数据信号** 当使用 16Bit 并口的 MCU 屏时，这些信号为 LT168B 对外部 16Bit MCU 屏的数据 Bit7~Bit0 传输信号。
USB 控制信号			
DP	1	1	**USB 数据端（Positive）** 此为 USB 数据端 DP 的信号。LT168 可以透过 USB 接口更新内部 MCU 程序及外接 SPI Flash 内的数据。
DM	2	2	**USB 数据端（Negative）** 此为 USB 数据端 DM 的信号。
I2C 控制信号			
SCL	-	56	**I2C 时钟** 该信号用于 I2C 时钟线信号，当未配置为接收器操作时，也可用于 GPIO 信号。
SDA	-	55	**I2C 数据** 该信号用于 I2C 数据线信号，当未配置为发送器操作时，也可用于 GPIO 信号。
四线 SPI 串口模块信号 0 (QSPI0) QSPI0 专用于芯片内部的闪存，这些信号已经直接连接到内部 QSPI 内存。			
QSIO0[3:0]	-	-	**QSPI0 输入/输出** 这些信号是 QSPI0 在主模式下的数据输出或输入。
QSCS0#	-	-	**QSPI0 片选输出** 该信号是 QSPI0 在主模式和低电平有效模式下的片选信号。
QSCK0	-	-	**QSPI0 时钟输出** 该信号是 QSPI0 在主模式下的串行时钟输出。
四线 SPI 串口模块信号 1 (QSPI1)			
QSIO1[3:0]	42, 41, 40, 39	62, 61, 60, 59	**QSPI1 输入/输出** 这些信号是 QSPI1 在主模式下的数据输出或输入。

©Levetop Semiconductor Co., Ltd.

信号名称	引脚号		引脚说明
	LT168A	LT168B	
QSCS1#	38	58	**QSPI1 片选输出** 该信号是 QSPI1 在主模式和低电平有效模式下的片选信号。
QSCK1	43	63	**QSPI1 时钟输出** 该信号是 QSPI1 在主模式下的串行时钟输出。
四线 SPI 串口模块信号 2 (QSPI2)			
QSIO2[3:0]	-, -, 14, 13	22, 20, 19, 18	**QSPI2 输入/输出** 这些信号是 QSPI2 在主模式下的数据输出或输入。 当未配置为 QSPI2 时，这些信号也可用于 PWM1[1]、PWM1[0]、PWM0[3]和 PWM0[2] 。
QSCS2#	11	16	**QSPI2 片选输出** 该信号是 QSPI2 在主模式和低电平有效模式下的片选信号。 当未针对 QSPI2 进行配置时，该信号也可用于 PWM0[0]。
QSCK2	12	17	**QSPI2 时钟输出** 该信号是 QSPI2 在主模式下的串行时钟输出。 当未针对 QSPI2 进行配置时，该信号也可用于 PWM0[1]。
EPORT 模块 0 信号			
INT0[7:0]	9, 8, 7, 6, -, -, -, -	13, 11, 9, 7, 12, 10, 8, 6	**中断输入 / GPIO 输出入** 这些双向信号可用作外部中断源或 GPIO 使用。 当未为中断源或 GPIO 配置时，这些信号也可用于 AIN[7:0]。
EPORT 模块 1 信号			
INT1[7:0]	24, -, 36, 33, -, 34, 35, 17	34, 26, 54, 51, -, 52, 53, 27	**中断输入 / GPIO 输出入** 这些双向信号可用作外部中断源或 GPIO 使用。
PWM 模块 0 信号			
PWM0[3:0]	14, 13, 12, 11	19, 18, 17, 16	这些输出信号可用作 PWM 0 输出或 GPIO 使用。
PWM 模块 1 信号			
PWM1[3:0]	15, -, -, -	24, 23, 22, 20	这些输出信号可用作 PWM 1 输出或 GPIO 使用。
ADC 模拟信号输入			
AIN[7:0]	9, 8, 7, 6, -, -, -, -	13, 11, 9, 7, 12, 10, 8, 6	**模拟信号输入** 这些模拟信号用作 ADC 模拟输入通道。 当未配置为模拟输入时，这些信号也可用于 INT0[7:0]。

©Levetop Semiconductor Co., Ltd.

信号名称	引脚号		引脚说明
	LT168A	LT168B	
程序烧录信号			
PGMCK	44	64	**程序烧录时钟** 该输入信号是用于对内部闪存进行程序烧录时的时钟信号。
PGMIO	37	57	**程序烧录数据** 该输入信号是用于对内部闪存进行程序烧录时的数据信号。
时钟信号			
EXTAL	47	67	**系统时钟信号源** 此引脚连接至外部 12Mhz 晶振
XTAL	48	68	**系统时钟信号源** 此引脚连接至外部 12Mhz 晶振
OSC32K_XI	5	5	**RTC 晶振输入** 此引脚连接至外部 32.768Khz 晶振。
OSC32K_XO	4	4	**RTC 晶振输出** 此引脚连接至外部 32.768Khz 晶振。
CLKOUT	17	27	**系统时钟信号输出** 该输出信号反映内部系统时钟。 当未配置为时钟输出时，此信号也可用于 INT1[0]。
复位输入信号			
RESET#	16	25	**复位输入信号** 当 RESET# = 0 时，将对内部 MCU 产生复位动作，除了少数由 POR 才能复位的寄存器外，大多数由 MCU 控制的寄存器将回复到默认值。
RSTOUT	-	26	**复位输出信号** 该输出信号指示内部复位控制器已在对芯片进行复位。 0 = 芯片处于复位状态 1 = 芯片未复位状态 当未配置为复位输出时，该信号也可用于 INT1[6]。
电源与接地信号 这些信号为芯片提供系统电源和接地。需确保芯片可以获得足够的电流能力。所有电源信号必须具有足够的旁路电容，以抑制高频噪声。			
VDD33	45	21, 65	**3.3V 电源输入** 供给 I/O 与 LDO 的电源

©Levetop Semiconductor Co., Ltd.

信号名称	引脚号		引脚说明
	LT168A	LT168B	
VDD12	46	66	**1.2V LDO 电源输出（Core）** 供内核数字电路的电源，此引脚必须外接一个 1uF 和一个 0.1uF 滤波电容到地。
AVDD	10	14	**3.3V ADC 等仿真电路的电源输入**
VBAT	3	3	**RTC 电源输入** **独立的 RTC 电源（电池）输入**
VSS	49(1)	69 (1)	**GND 接地**
AVSS	-	15	**模拟组件 GND 接地**

提示(1)：这是散热焊盘（Thermal Pad Zone）必须接到 VSS 或是 GND 。在做 PCB 布局时需要特别注意焊盘的焊接面设计，详细请参考第 34.3 节的说明。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

2.4. LT168A 与 LT168B 之比较

表 2-3：LT168x 比较表

功 能		LT168A	LT168B
项 目	说 明		
TFT LCD 屏	RGB 接口	--	V (480*480 max)
	16bit 8080 接口	--	V (800*480 max)
	8bit 8080 接口	V (480*320 max)	V (480*320 max)
	QSPI 接口	--	V
MCU 内核、内存	内核	32-bits RISC	32-bits RISC
	速度	200MHz	200MHz
	Flash 容量	512KB	2MB
	SRAM 容量	256KB	256KB
	显存(*1) 容量	512KB	512KB
	外接 PSRAM 显存扩展	--	V
其他模块接口	Uart 接口	V (x3)	V (x3)
	SPI 接口	V (x1)	--
	QSPI 接口	V (x1)	V (x2)
	USB 2.0	V	V
	SD 卡	V	V
	PWM 输出	V (x5)	V (x8)
	Can Bus	V (x1)	V (x1)
	ADC 输入	V (x4)	V (x8)
	RTP 接口	V	V
	CTP I2C 接口	V	V
	GPIO 接口	V (x10)	V (x19)
	RTC 时钟	V	V
应用与升级	UI_Editor-II	V	V
	UI_Emulator-II	V	V
	USB 口升级	V	V
	Uart 口升级	V	V
	SD 卡升级	V	V
	二次开发	V	V
电源与封装	电源	3.3V	3.3V
	封装	QFN-48	QFN-68

提示*1： 显存 RAM 可作为普通 SRAM 使用，这也意味着最大支持的 SRAM 高达 768KB。

©Levetop Semiconductor Co., Ltd.

3. 硬件接口

3.1. 主控端 MCU 通讯接口

LT168 和主控 MCU 的通讯模式是透过 UART 接口，两边的 UART 接口 TX 及 RX 必须交叉对接，参考下图所示。如果连接的距离很长还需要增加 RS232 驱动芯片，以避免信号衰减影响通讯。串口通讯的软件设置和通讯协议可参考乐升半导体的的串口屏应用手册（UI_Editor-II_CH_Vxx.pdf）。

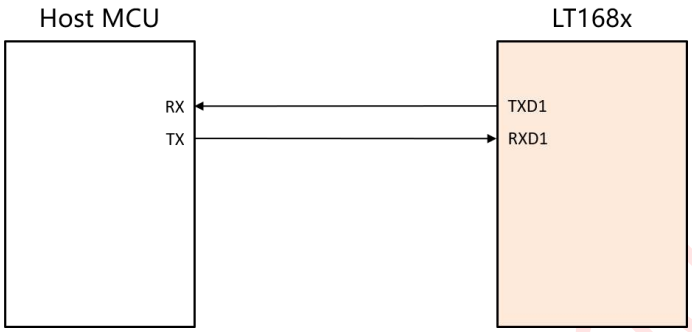


图 3-1：LT168 的串口和主控 MCU 的通讯

3.2. TFT LCD 屏的控制接口

LT168 提供了一个外部总线接口（Exernal Bus Interface, EBI），用于驱动并行 8080 接口的 TFT 面板。该并行接口的数据总线为 8 位或 16 位。其中 LT168A 提供 8 位的 8080 并行接口，可与 8 位 8080 接口的 TFT 屏连接，参考原理图如下图 3-2，LT168B 提供 16 位的 8080 并行接口，可与 16 位 8080 接口的 TFT 屏连接，通常使用 16 位 8080 接口的 TFT 屏可以达到更快的显示刷新速度，这种接口的 TFT 屏 LT168B 最高可以支持到 800x480 的分辨率，参考原理图如下图 3-3 所示：

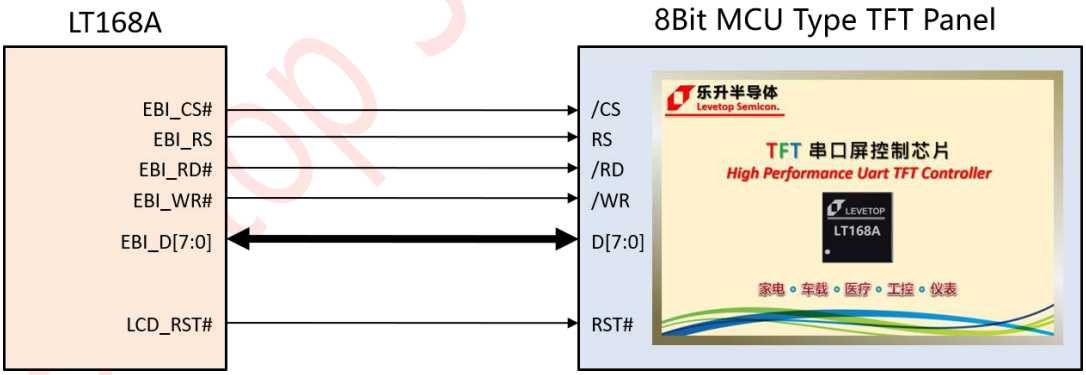


图 3-2：LT168A 与 8 位 8080 并行接口的 TFT 屏连接

©Levetop Semiconductor Co., Ltd.

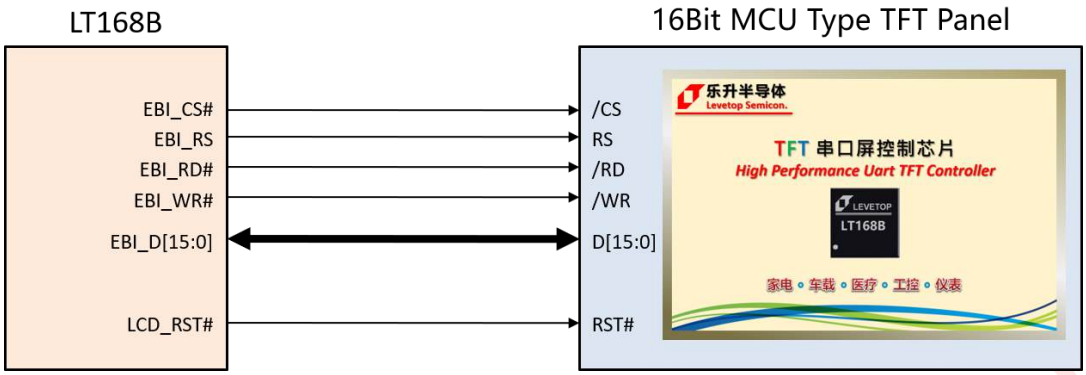


图 3-3：LT168B 与 16 位 8080 并行接口的 TFT 屏连接

LT168B 也提供 RGB 的 LCD 接口模式，可与分辨率在 480x480（含）以下的 RGB 接口 TFT 屏连接，参考原理图如下图所示。图中的外接 PSRAM 是保留用做显存数据暂存区，如果要实现更丰盛的显示效果如图片迭加等功能，可以选择增加此外接 PSRAM 芯片。

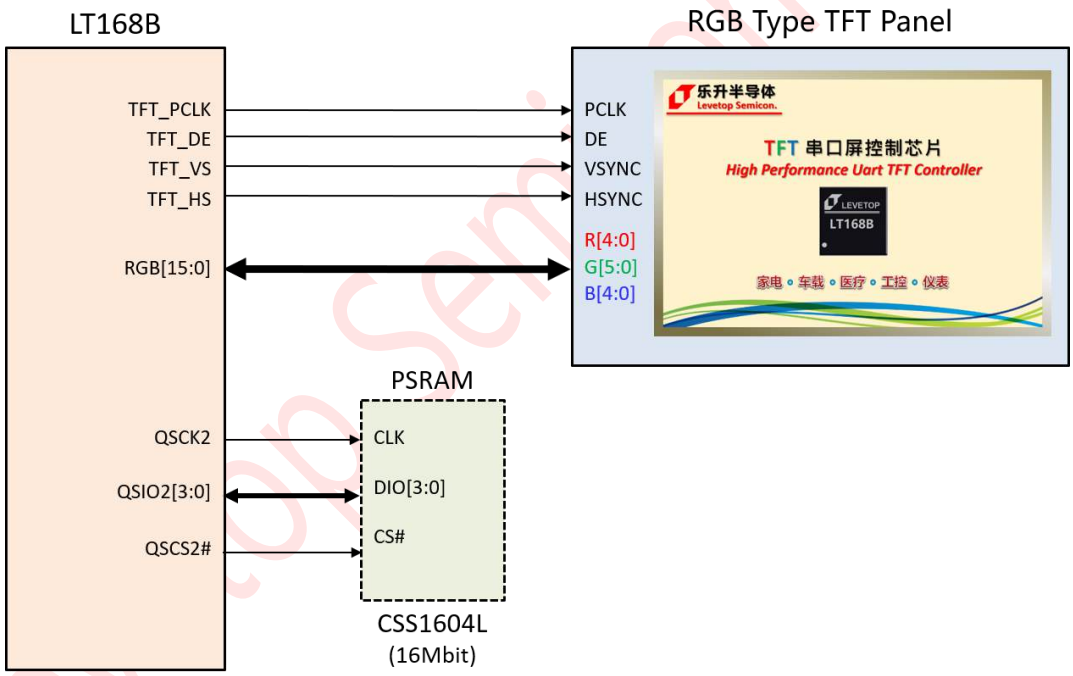


图 3-4：LT168B 与 RGB 接口的 TFT 屏连接

©Levetop Semiconductor Co., Ltd.

表 3-1：LT168B 是否外加 PSRAM 的显示功能差异

分辨率	RGB 480 x 272		RGB 480 x 480		8/16bit MCU 800 x 480	
串口功能	不加 PSRAM	外加 PSRAM	不加 PSRAM	外加 PSRAM	不加 PSRAM	外加 PSRAM
页面缓冲刷新	支持	支持	不支持	支持	不支持	支持
Png/叠加显示限制大小	480 x 272	480 x 272	200 x 200	480 x 480	320 x 320	800 x 480
显示页面 ZIP 压缩	不支持	支持	不支持	支持	不支持	不支持
画面滑动功能	不支持	支持	不支持	支持	不支持	不支持

LT168B 提供两组 QSPI，可与 QSPI 接口的 TFT 屏连接，参考原理图如下图所示：

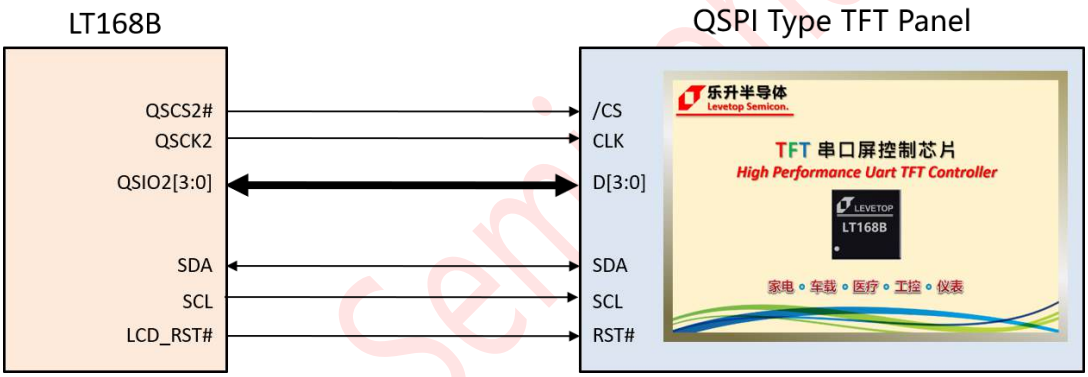


图 3-5：LT168B 与 QSPI 接口的 TFT 屏连接

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

3.3. QSPI 接口

LT168 有两组 QSPI 的接口，一组是用来连接外部的 QSPI Flash，这个外部的 Flash 是被用做储存显示图片、动画、文字和其他信息。当 LT168 收到主控经由 Uart 接口发送过来的串口指令，它会根据指令从 QSPI Flash 提取图片或者其他显示相关信息，和传输 LCD 屏的数据。这些数据可以在 SPI FLASH 中通过 USB 端口连接电脑软件更新（参考串口屏应用手册），参考原理图如下图所示。另外一组 QSPI 可以连接其他的 QSPI 组件，或是上一节提到的 QSPI 屏。

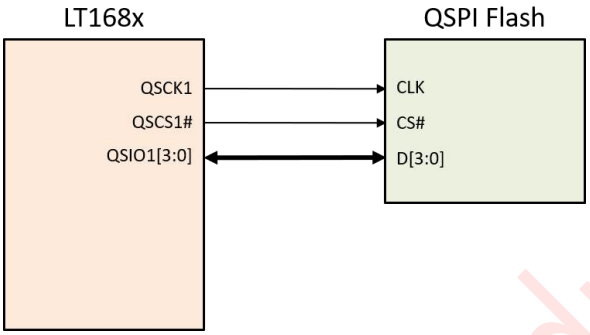


图 3-6：LT168 连接 QSPI Flash 原理图

3.4. LCD 触控屏接口

LT168 有一个 ADC 输入模块和 I2C 的控制电路，可以直接与外部 LCD 上面的触控屏连接，参考原理图如下：

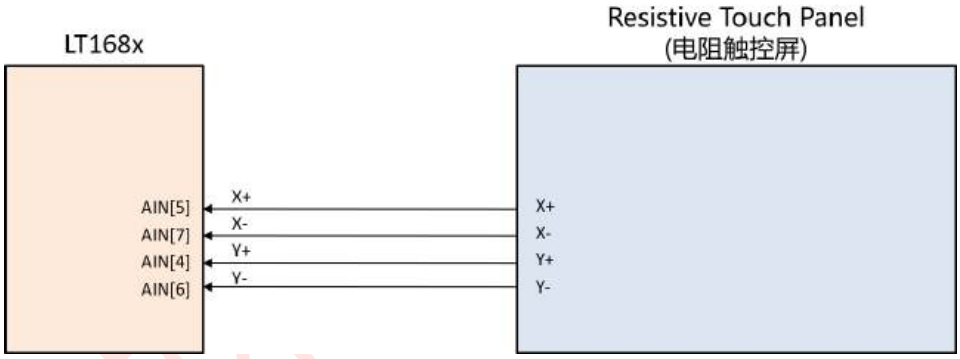


图 3-7：LT168 连接到电阻触控屏

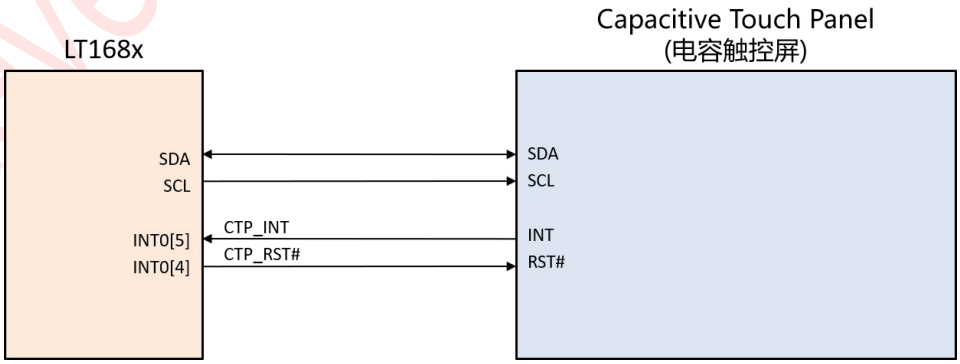


图 3-8：LT168 连接到电容触控屏

©Levetop Semiconductor Co., Ltd.

3.5. 时钟信号源接口

LT168 需要一个外部的 12MHz 晶振作为内部的系统时钟和 USB 的时钟来源，参考原理图如下：

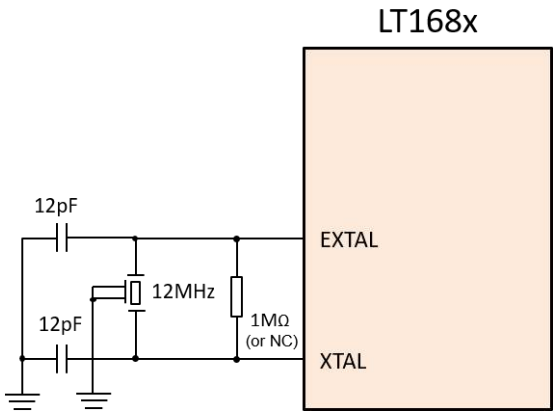


图 3-9：外部的 12MHz 晶振原理图

3.6. LCD 背光控制接口

LT168 使用 PWM1[3] 来提供一个背光控制的信号 - "BL_PWM"，可以用来控制 TFT LCD 屏的背光或是背光的升压电路，参考原理图如下两个范例：

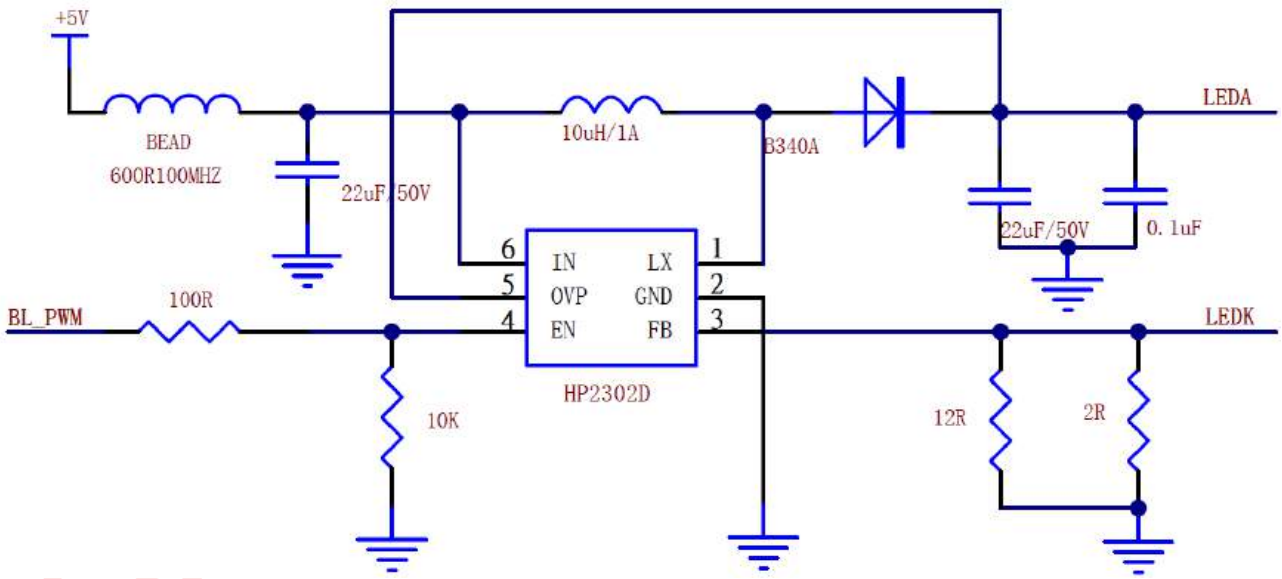


图 3-10：TFT LCD 背光控制参考原理图 - 1

©Levetop Semiconductor Co., Ltd.

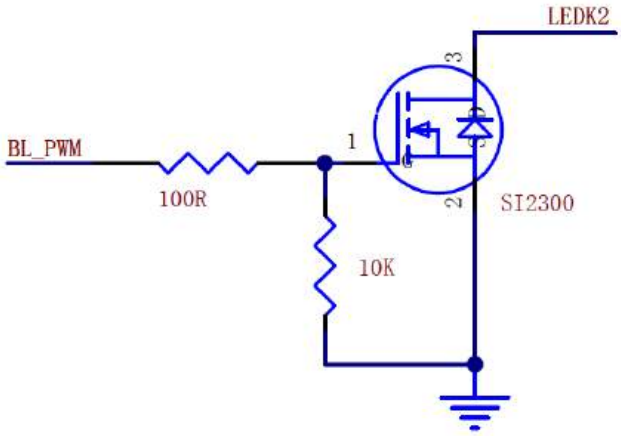


图 3-11：TFT LCD 背光控制参考原理图 - 2

3.7. Can Bus 接口

LT168 支持 Can Bus 协议，提供一组 Can Bus 接口，再经过一个 Can Bus 驱动芯片与外部通讯，参考原理图如下：

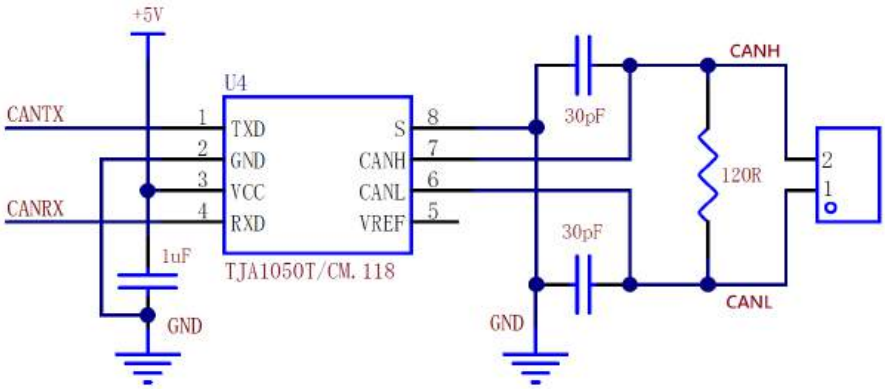


图 3-12：Canbus 驱动原理图

©Levetop Semiconductor Co., Ltd.

3.8. 声音输出接口

LT168 可以用 PWM1[2] 来提供声音输出信号 - "HORN"，再经过一个简易的三级管放大电路推动喇叭，参考原理图如下：

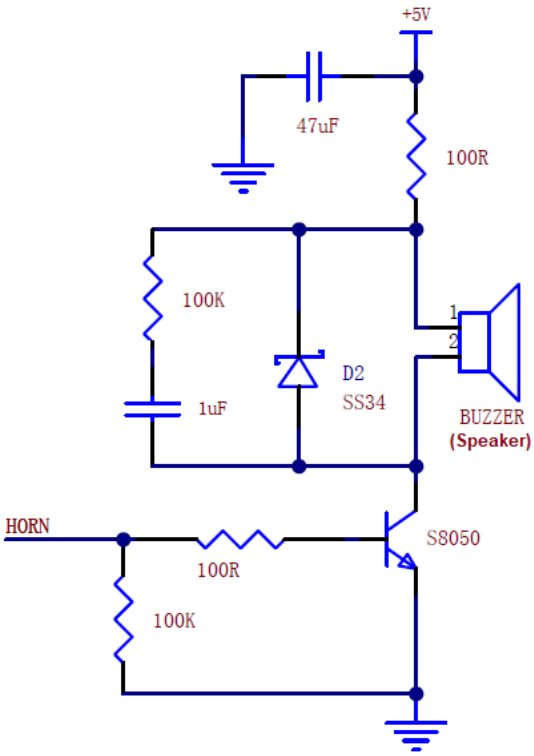


图 3-13：声音输出参考原理图

3.9. RTC（Real Time Clock）的时钟源与电源

LT168 内部有 RTC（Real Time Clock）时钟模块，如果使用这个 RTC 时钟，需要提供一 32.768KHz 的晶振电路，RTC 是独立供电，如果在外部电源关闭时仍保持 RTC 继续运行可以加上一外部电池电源，参考原理图如下：

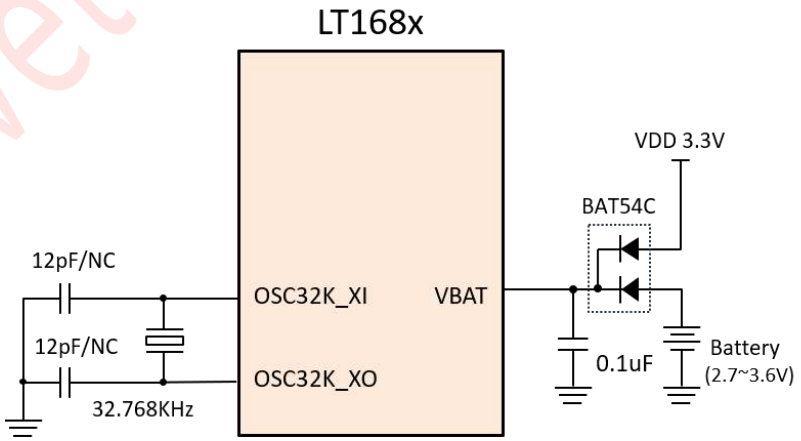


图 3-14：RTC 时钟源与外部电池电源的参考原理图

©Levetop Semiconductor Co., Ltd.

3.10. 复位（Reset）

LT168 的硬件复位来源有 2 种，2 种复位都会经过内部时钟做同步处理：

– 电源开启复位（Power on Reset）

– 外部复位输入信号（External Reset Pin，RESET#）

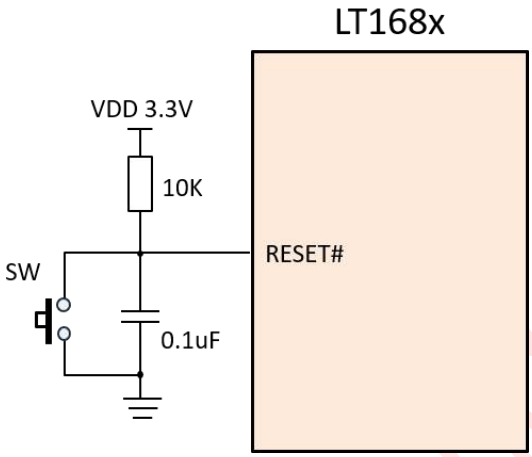


图 3-15：外部复位参考原理图

在工业应用或使用电池供电环境时，为避免电源波动导致内部复位不稳定建议增加专用复位芯片电路（如下图所示）：

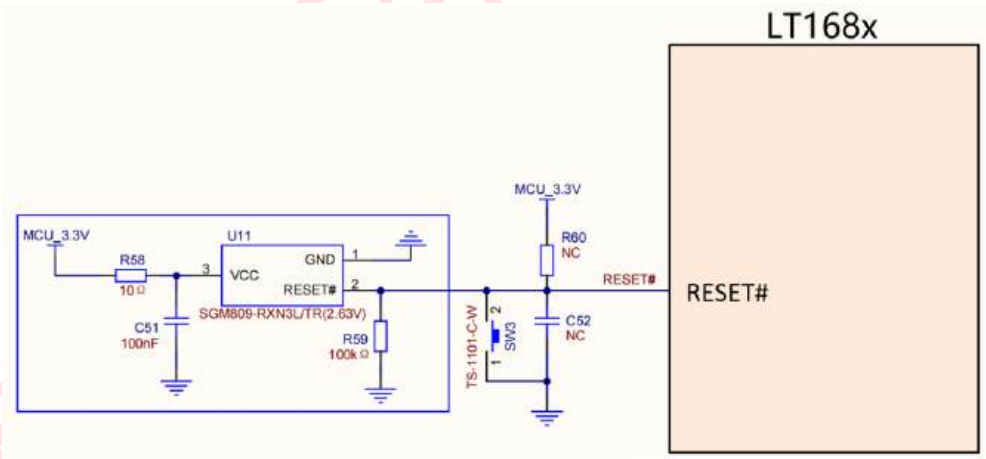


图 3-16：外加专用复位芯片参考原理图

©Levetop Semiconductor Co., Ltd.

3.11. USB 接口

LT168 提供一个从机的 USB 功能。透过 USB 接口可以用专用 PC 电脑软件更新 LT168x 内部 MCU 程序及外接 SPI Flash 内的数据，详细操作模式请参考乐升半导体的串口屏应用手册（UI_Editor-II_CH_Vxx.pdf）。

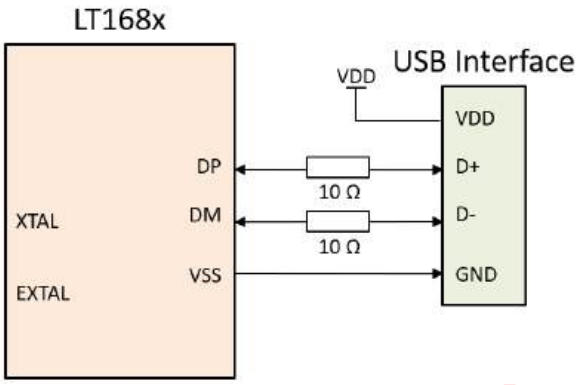


图 3-17：LT168 连接到 USB 端口的应用电路

3.12. SPI/Uart 扩展接口

除了 TFT 串口屏的必要引脚外，LT168 还可以提供一些引脚供外部 SPI/QSPI 组件，其中 QSPI2 若接上 QSPI PSRAM (Pseudo SRAM) 可以用来暂存显示图片素材或数据，方便 MCU 进行图片迭加及大量的数据处理。其余的 Uart 串口可供给 Uart 串口通讯（SCI 接口）的组件，如 WiFi 模块、蓝牙模块等，参考原理图如下：

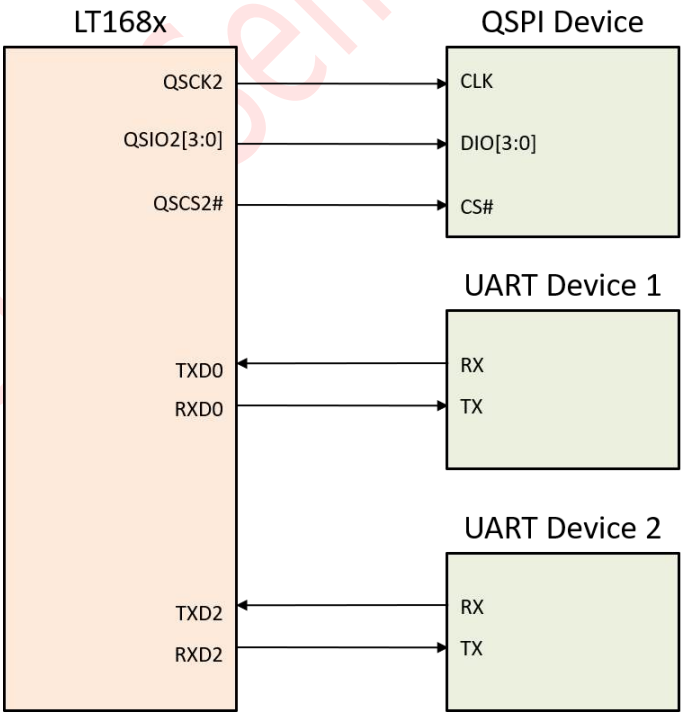


图 3-18：额外的 SPI、Uart 接口参考原理图

©Levetop Semiconductor Co., Ltd.

4. 内存与寄存器配置

4.1. 简要说明

LT168 的内建/外部内存、寄存器包括：

- 最高 128M Bytes 的外部 QSPI Flash 闪存空间
- 内建 512K Bytes (LT168A) 或是 2M Bytes (LT168B) QSPI Flash 闪存
- 8K Bytes 内部 Boot ROM
- 768K Bytes 内部静态 SRAM
  - 系统 RAM：前 256K Bytes，地址由 0x00800000 开始
  - 显存 RAM：后 512K Bytes，地址由 0x00840000 开始
- 各模块内部寄存器

4.2. 地址配置图

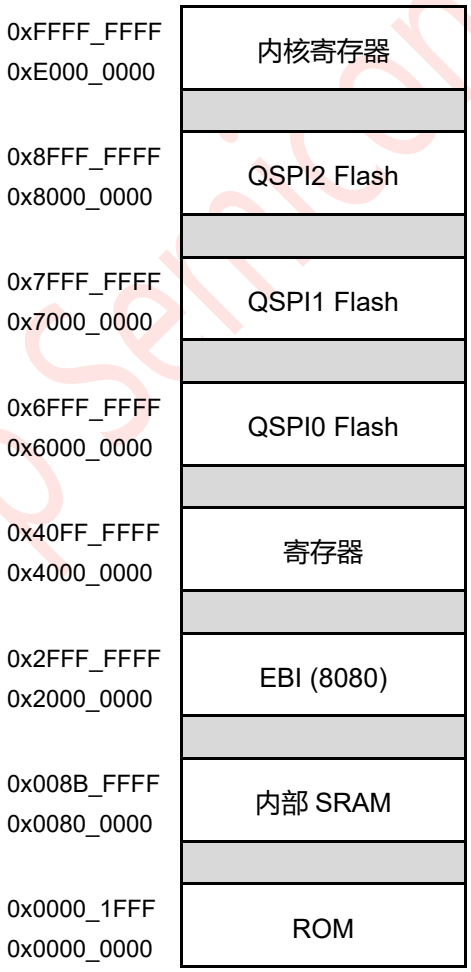


图 4-1：地址配置图

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

表 4-1：硬件模块与寄存器地址的配置图

配置地址	最大区块	硬 件 模 块	章节
0x4000_0000	64Kbyte	Direct Memory Access Controller (DMAC)	14
0x4001_0000	64Kbyte	Chip Configuration Module (CCM)	8
0x4002_0000	64Kbyte	Reset Control Module (RCM)	10
0x4003_0000	64Kbyte	Clock and Power Control Module (CLKPWRM)	9
0x4004_0000	64Kbyte	Programmable Interrupt Timer 0 (PIT0)	20
0x4005_0000	64Kbyte	Programmable Interrupt Timer 1 (PIT1)	20
0x4006_0000	64Kbyte	Programmable Interrupt Timer 2 (PIT2)	20
0x4007_0000	64Kbyte	Programmable Interrupt Timer 3 (PIT3)	20
0x4008_0000	64Kbyte	Serial Communication Interface 1 (SCI1)	25
0x4009_0000	64Kbyte	Serial Communication Interface 0 (SCI0)	25
0x400A_0000	64Kbyte	Analog Comparator 0 (COMP0)	29
0x400B_0000	64Kbyte	Analog Comparator 1 (COMP1)	29
0x400C_0000	64Kbyte	Serial Communication Interface 2 (SCI2)	25
0x400D_0000	64Kbyte	Pulse Width Modulator 0 (PWM0)	28
0x400E_0000	64Kbyte	Pulse Width Modulator 1 (PWM1)	28
0x400F_0000	64Kbyte	Edge Port Module 0 (EPORT0)	23
0x4010_0000	64Kbyte	Edge Port Module 1 (EPORT1)	23
0x4011_0000	64Kbyte	Analog-to-Digital Convertor (ADC)	31
0x4012_0000	64Kbyte	Efuse Control Module (EFM) and Option Byte (OPB)	15 16
0x4013_0000	64Kbyte	WatchDog Timer (WDT)	21
0x4014_0000	64Kbyte	Real Time Controller (RTC)	22
0x4015_0000	64Kbyte	Inter-Integrated Circuit (I2C)	27
0x4016_0000	64Kbyte	USB2.0 Full-Speed Device Controller (USBC)	30
0x4017_0000	64Kbyte	Crossbar Switch (XBAR)	13
0x4018_0000	64Kbyte	External Bus Interface (EBI)	19
0x4019_0000	64Kbyte	CACHE Module (CACHEM)	12
0x401A_0000	64Kbyte	RGB Controller (RGBC)	17
0x401B_0000	64Kbyte	Blender Controller (BLDC)	18
0x401C_0000	64Kbyte	CANBus Controller (CANBC)	24
0x401D_0000	64Kbyte	Edge Port Module 2 (EPORT2)	23
0x6000_0000	64Kbyte	Synchronous Serial Interface 0 (SSI0) - QSPI0	26
0x7000_0000	64Kbyte	Synchronous Serial Interface 1 (SSI1) - QSPI1	26
0x8000_0000	64Kbyte	Synchronous Serial Interface 2 (SSI2) - QSPI2	26
0xE000_0000	4Kbyte	Embedded Interrupt Controller (EIC)	5
0xE000_1000	4Kbyte	Embedded Programmable Timer (EPT)	7

©Levetop Semiconductor Co., Ltd.

5. 嵌入式中断控制器（EIC）

本节介绍了 32 位 RISC 的嵌入式中断控制器。

5.1. 基本介绍

LT168 具有一个中断控制器，它从多个中断源收集请求，并为 CPU 中断逻辑提供一个接口。

5.2. 功能特点

中断控制器的功能包括：

- 可配置的中断源，最多为 32 个
- 每个中断源都有 32 个独特的可编程优先级级别
- 基于优先级独立启用/禁用待起中断
- 每个中断源的一个固定的向量数
- 同时支持电平敏感中断和脉冲中断
- 支持 PendTrap 功能
- 支持软件复位

5.3. 内存映射和寄存器

本节介绍了内存映射图（请参见表 5-1)和寄存器。嵌入式中断控制器的基本地址为 0xE000_0000。

5.3.1. 内存映射

EIC 模块基地址（EIC_BASEADDR）定义为 32 位 RISC 内部参数定义。默认值为 0xE0000000。EIC 寄存器的实际地址是 EIC_BASEADDR 加上每个 EIC 寄存器的偏移量地址。核心内部模块占 64K 地址区。系统应避免将其他寄存器映射到从 EIC_BASEADDR 到 EIC_BASEADDR+0x0000_FFFF 的区域。

表 5-1：中断控制器模块内存映射

偏移地址	Bits[31-24]	Bits[23-16]	Bits[15-8]	Bits[7-0]	访问权限(1)
0x0000_0000	Interrupt control status register (ICSR)				S/U
0x0000_0004	保留				S/U
0x0000_0008	保留				S/U
0x0000_000C	保留				S/U
0x0000_0010	Interrupt Enable Register (IER)				S/U
0x0000_0014	保留				S/U
0x0000_0018	Interrupt Pending Set Register (IPSR)				S/U
0x0000_001C	Interrupt Pending Clear Register (IPCR)				S/U

©Levetop Semiconductor Co., Ltd.

偏移地址	Bits[31-24]	Bits[23-16]	Bits[15-8]	Bits[7-0]	访问权限(1)
0x0000_0020 | 0x0000_003C	未使用(2)				-
Priority level select registers (PLSR0-PLSR31)					
0x0000_0040	PLSR3	PLSR2	PLSR1	PLSR0	S/U
0x0000_0044	PLSR7	PLSR6	PLSR5	PLSR4	S/U
0x0000_0048	PLSR11	PLSR10	PLSR9	PLSR8	S/U
0x0000_004C	PLSR15	PLSR14	PLSR13	PLSR12	S/U
0x0000_0050	PLSR19	PLSR18	PLSR17	PLSR16	S/U
0x0000_0054	PLSR23	PLSR22	PLSR21	PLSR20	S/U
0x0000_0058	PLSR27	PLSR26	PLSR25	PLSR24	S/U
0x0000_005C	PLSR31	PLSR30	PLSR29	PLSR28	S/U
0x0000_0060	System Priority level select register(SYSPLSR)				S/U
0x0000_0064 | 0x0000_007C	未使用(2)				-

提示：

(1) 在 32 位 RISC，该寄存器在任何情况下都可以被访问。

(2) 访问未实现的地址位置没有任何影响，并将导致一个周期终止传输错误。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

5.3.2. 寄存器描述

本节包含了对中断控制器模块寄存器的描述。嵌入式中断控制器的基本地址（EIC_BASEADDR）为 0xE000_0000。

5.3.2.1. 中断控制状态寄存器（ICSR）

32 位中断控制寄存器（ICSR）反映了中断控制器的状态

地址：EIC_BASEADDR+0x0000_0000

	31	30	29	28	27	26	25	24
读:	SRST	0		SetPTrap	ClrPTrap	0		
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0	VEC[6:0]						
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 5-1：中断控制状态寄存器（ICSR）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

SRST — Software Reset Bit

此只写位用于创建一个软件复位请求。设置这个位将在系统重新序列信号上产生一个脉冲。读取总是返回 0。

SetPTrap — Set PendTrap Bit

此读/写位用于创建一个等待处理的软件中断。该操作类似于执行“陷阱”指令。但是，待所有更高优先级的异常/中断退出之前，将不会进入待定的软件中断。当软件中断时，比特将自动清除。复位也会清除这一点。

读内容：

1 = 软件中断正在等待处理

0 = 软件中断没有等待处理

写数据：

1 = 将软件中断设置为等待处理

0 = 无影响

ClrPTrap — Clear PendTrap Bit

读/写 ClrDSI 位用于取消等待处理的软件中断（PendTrap）。

复位清除此位。

读内容：

1 = 软件中断正在等待处理

0 = 软件中断没有等待处理

写数据：

1 = 取消了等待处理的软件中断

0 = 无影响

VEC[6:0] — Interrupt Vector Number Field

只读 VEC[6:0]字段包含 7 位的中断向量数。复位清除 VEC[6:0]。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

5.3.2.2. 中断启用寄存器（IER）

这个可读/写的 32 位中断启用寄存器（IER）分别启用任何当前等待处理的中断，这些中断是作为一个被分配给每个优先级级别的正常中断源。启用具有发送请求的中断源会导致该请求等待处理，如果还未完成，则会对 CPU 发送请求。

地址：EIC_BASEADDR+0x0000_0010

	31	30	29	28	27	26	25	24
读: 写:	IE[31:24]							
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读: 写:	IE[23:16]							
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读: 写:	IE[15:8]							
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读: 写:	IE[7:0]							
复位:	0	0	0	0	0	0	0	0

图 5-2：中断启用寄存器（IER）

IE[31:0] — Interrupt Enable Field

此读/写 IE[31:0]字段允许来自相应优先级的来源中断请求作为中断请求。复位将清除 IE[31:0]。

1 = 允许中断请求

0 = 禁止中断请求

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

5.3.2.3. 中断等待集寄存器（IPSR）

地址：EIC_BASEADDR+0x0000_0018

	31	30	29	28	27	26	25	24
读: 写:	SetPend[31:24]							
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读: 写:	SetPend[23:16]							
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读: 写:	SetPend[15:8]							
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读: 写:	SetPend[7:0]							
复位:	0	0	0	0	0	0	0	0

图 5-3：中断等待集寄存器（IPSR）

SetPend[31:0] — Interrupt Pending Set Field

此读/写 SetPend[31:0]字段设置为等待关联的中断，并指示关联的中断是否等待处理。复位可以清除 SetPend。

读内容：

1 = 相关的中断正在等待处理

0 = 关联的中断未等待处理

写数据：

1 = 将关联中断的状态更改为等待处理

0 = 无影响

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

5.3.2.4. 中断待定清除寄存器（IPCR）

地址：EIC_BASEADDR+0x0000_001C

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

图 5-4：中断待定清除寄存器（IPCR）

ClrPend[31:0] — Interrupt Pending Clear Field

此读/写 ClrPend[31:0]字段清除，正在等待相关的中断，并指示相关的中断是否正在等待。复位清除 ClrPend[31:0]。

读内容：

1 = 相关的中断正在等待处理

0 = 关联的中断未等待处理

写数据：

1 = 将关联中断的状态更改为不等待处理

0 = 无影响

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

5.3.2.5. 优先级级别选择寄存器（PLSR）

此读/写 8 位优先级选择寄存器（PLSRx）是 32 个读/写，8 位优先级选择寄存器 PLSR0-PLSR31，每个中断源一个。PLSRx 寄存器为中断源 x 分配了一个优先级级别。

地址：从 EIC_BASEADDR+0x0000_0040 到 EIC_BASEADDR+0x0000_005C

	7	6	5	4	3	2	1	0
读:	PLSRx[7:6]		0	0	0	0	0	0
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 5-5：优先级选择寄存器（PLSR0-PLSR31）

PLSRx[7:6] — Priority Level Select Field

IRQ0~31 的默认优先级值为 0~31。该值越低，优先级就越高。这意味着 IRQ0 优先级> IRQ1 > ..... > IRQ31 为默认值。但是，用户可以设置 PLSRx[7:6]来调整中断的优先级。优先级级别的实际值是默认值加上 PLSRx[7:6] *64。例如，如果 PLSR1[7:6] = 2 时，IRQ1 的优先级值为 1+2*64 = 129，则 IRQ1 的优先级低于任何优先级值较低的 IRQ。

表 5-2：优先值调整

PLSRx[7:6]	要增加的优先级值
00	0
01	64
10	128
11	192

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

5.3.2.6. 系统优先级别选择寄存器（SYSPLSR）

地址：EIC_BASEADDR+0x0000_0060

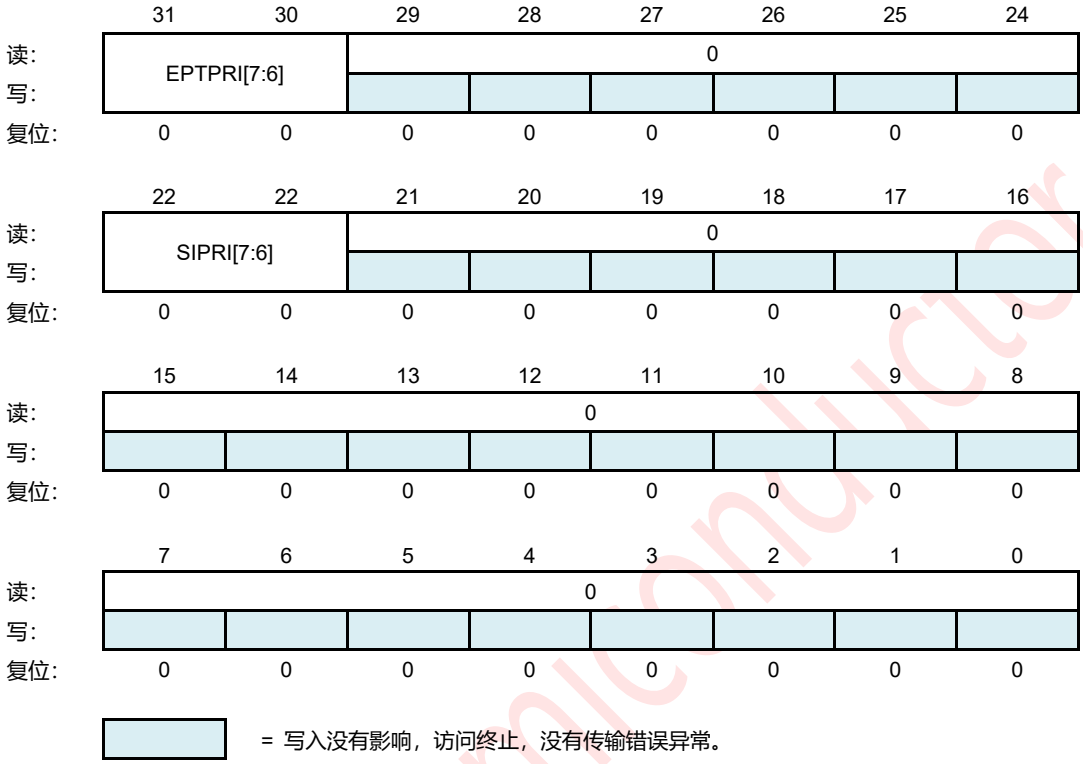


图 5-6：系统优先级级别选择寄存器（SYSPLSR）

EPTPRI[7:6] — EPT Priority Level Select Field

EPT 中断的默认优先级为-2。这意味着 EPT 中断优先级高于其他正常 irq 和系统软件中断（PendTrap）。值越低，优先级越高。但是，用户可以设置 EPTPRI[7:6]来调整 EPT 中断的优先级。优先级级别的实际值是默认值加上 PRI[7:6] *64。例如，如果 PRI[7:6] = 2，则 EPT 中断优先级值为-2+2*64 = 126。

表 5-3：优先值调整

EPTPRI[7:6]	要增加的优先级值
00	0
01	64
10	128
11	192

SIPRI[7:6] — Software Interrupt Priority Level Select Field

软件中断（PendTrap）的默认优先级为-1。这意味着 EPT 中断优先级高于其他正常 irq。值越低，优先级越高。但是，用户可以设置 SIPRI[7:6]来调整软件中断的优先级。优先级级别的实际值是默认值加上 SIPRI[7:6] *64。例如，如果为 SIPRI[7:6] = 2，则软件中断优先级值为-1+2*64 = 127。

©Levetop Semiconductor Co., Ltd.

表 5-4：优先值调整

SIPRI[7:6]	要增加的优先级值
00	0
01	64
10	128
11	192

5.4. 功能描述

EIC 同时支持电平敏感中断和脉冲中断。中断源号是从 1 到 32。

中断将出现以下原因之一：

- EIC 检测到中断信号是活动的，而相应的中断不是活动的
- EIC 检测到中断信号上的上升边缘

等待处理的中断仍保持等待处理，直到出现以下情况之一：

- 处理器进入 ISR 以进行中断。这将将中断的状态从等待处理状态更改为活动状态。然后
  - 对于电级敏感的中断，当处理器从 ISR 返回时，EIC 采样中断信号。如果断言信号，中断状态变为待定，这可能导致处理器立即重新进入 ISR。否则，中断的状态将变为非活动状态。
  - 对于脉冲中断，EIC 将连续监控中断信号。如果中断信号是脉冲，中断的状态变为待起和激活。在这种情况下，当处理器从 ISR 返回时，中断的状态会变为等待处理，这可能会导致处理器立即重新进入 ISR。如果当处理器在 ISR 中时中断信号没有被脉冲，则当处理器从 ISR 返回时，中断的状态将变为非活动状态。
- 软件写入相应的中断，等待清除寄存器位。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

5.4.1. 无冲突的中断处理

如果中断是脉冲，中断的状态将变为待起。没有冲突，中断会导致处理器立即进入 ISR。当处理器从 ISR 返回时，中断的状态将变为非活动状态。

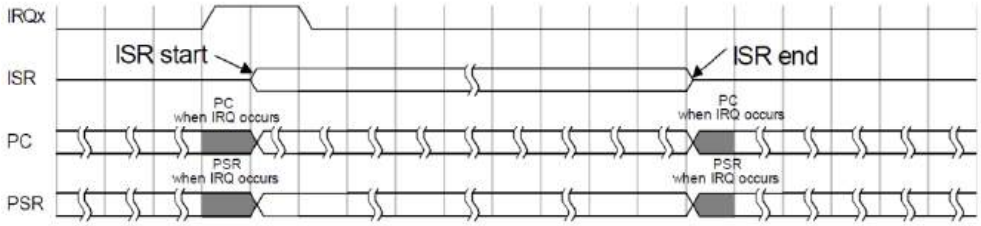


图 5-7：One Pulse Interrupt without Confliction

对于电平敏感的中断，如果断言信号，则中断的状态将变为待决。没有冲突，中断会导致处理器立即进入 ISR。当处理器从 ISR 返回时，EIC 继续采样中断信号。如果该信号未被清除，则处理器将重新进入 ISR。否则，中断的状态将变为非活动状态。

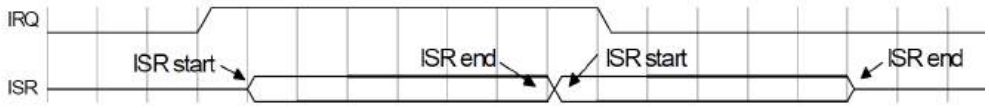


图 5-8：Level-sensitive Interrupt without Confliction

5.4.2. 有冲突的中断

当同时断两个中断信号时，中断仲裁器将判断哪一个具有更大的优先级。例如，如果 IRQx 的优先级大于 IRQy，处理器将输入 ISRx，IRQy 变为待起。当处理器从 ISRx 返回后，处理器将立即进入 ISRy。

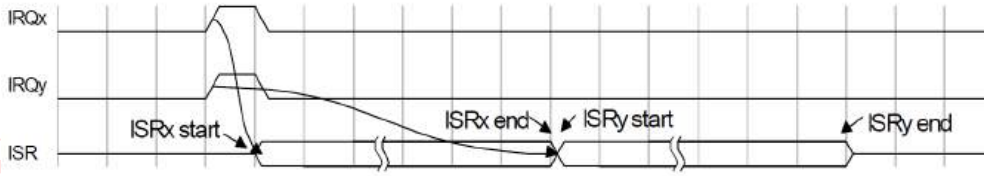


图 5-9：同时发生两次中断

如果在另一个中断处理过程中断言了一个中断信号，则会有两种情况：

1. 断言的中断优先级低于处理中断优先级。在这种情况下，断言的中断状态将等待处理，直到处理中断结束。
2. 断言的中断优先级高于处理中断优先级。在这种情况下，较高优先级的中断处理将嵌套在较低优先级的中断例程中。

©Levetop Semiconductor Co., Ltd.

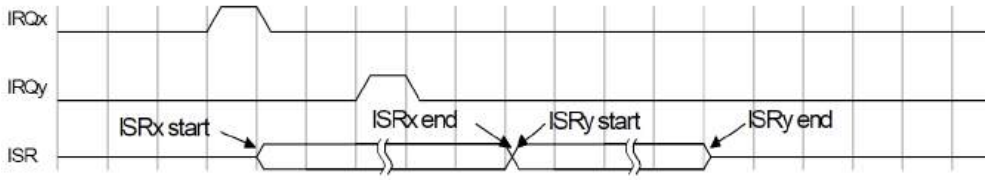


图 5-10：具有冲突的较低优先级中断

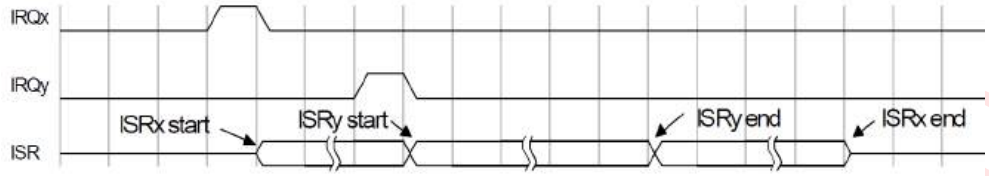


图 5-11：具有冲突的更高优先级中断

5.4.3. 待定陷阱功能

ICSR 中的 SetPTrap/ClrPTrap 位用于创建/取消一个“待定”的软件中断请求。一旦处理器从更高优先级的中断返回，“等待处理”的软件中断将被处理器接受。

通常，等待陷阱功能用于操作系统任务被动切换。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

5.5. 中断来源

中断控制器为每个中断源分配一个数字，如表 5-5 所示：

表 5-5：中断来源分配

根源	模块	标志	来源描述	标志清除机制
0	ADC			
1	QSPI0	XRXOIS	XIP Receive FIFO Overflow Interrupt	
		RXFIS	Receive FIFO Full Interrupt	
		RXOIS	Receive FIFO Overflow Interrupt	
		RXUIS	Receive FIFO Underflow Interrupt	
		TXOIS	Transmit FIFO Overflow Interrupt	
		TXEIS	Transmit FIFO Empty Interrupt	
2	SCI0	TDRE	Transmit Data Register Empty Flag	
		TC	Transmission Complete	
		TXOF	Transmitter Buffer Overflow Flag	
		LBKDIF	LIN Break Detect Interrupt Flag	
		IDLE	Idle Line Flag	
		RXEDGIF	RXD0 Pin Active Edge Interrupt Flag	
		RDRF	Receive Data Register Full Flag	
		MA1F	Match 1 Flag	
		MA2F	Match 2 Flag	
		OR	Receiver Overrun Flag	
		NF	Noise Flag	
		FE	Framing Error Flag	
		PF	Parity Error Flag	
		RXUF	Receiver Buffer Underflow Flag	
3	COMP0	CPRIF		
		CPFIF		
4	COMP1	CPRIF		
		CPFIF		
5	DMAC	DONE[0]		Write DONE[0] = 1

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

根源	模块	标志	来源描述	标志清除机制
		DONE[1]		Write DONE[1] = 1
		DONE[14]		Write DONE[14] = 1
		DONE[15]		Write DONE[15] = 1
		DMA_ESR[GPE]	Group Priority Error	Write channel number to CERR[6:0] to clear error status
		DMA_ESR[CPE]	Channel Priority Error	Write channel number to CERR[6:0] to clear error status
		DMA_ESR[SAE]	Source Address Error	Write channel number to CERR[6:0] to clear error status
		DMA_ESR[SOE]	Source Offset Error	Write channel number to CERR[6:0] to clear error status
		DMA_ESR[DAE]	Destination Address Error	Write channel number to CERR[6:0] to clear error status
		DMA_ESR[DOE]	Destination Offset Error	Write channel number to CERR[6:0] to clear error status
		DMA_ESR[NCE]	Nbytes/Citer Configuration Error	Write channel number to CERR[6:0] to clear error status
		DMA_ESR[SGE]	Scatter/Gather Configuration Error	Write channel number to CERR[6:0] to clear error status
		DMA_ESR[SBE]	Source Bus Error	Write channel number to CERR[6:0] to clear error status
		DMA_ESR[DBE]	Destination Bus Error	Write channel number to CERR[6:0] to clear error status
6	WDT0	IF		
7	PWM0	PIFR[0]		
		PIFR[1]		
		PIFR[2]		
		PIFR[3]		
8	PWM1	PIFR[0]		
		PIFR[1]		
		PIFR[2]		
		PIFR[3]		
9	PIT0	PIF	PIT Flag	Writing a 1 to it or writing to PMR
10	PIT1	PIF	PIT Flag	Writing a 1 to it or writing to PMR
11	PIT2	PIF	PIT Flag	Writing a 1 to it or writing to PMR
12	PIT3	PIF	PIT Flag	Writing a 1 to it or writing to PMR
13	RTC	Day_intf	Day pulse flag	
		Hou_intf	Hour pulse flag	
		Min_intf	Minute pulse flag	
		Sec_intf	Second pulse flag	
		Ala_intf	Alarm flag	

©Levetop Semiconductor Co., Ltd.

根源	模块	标志	来源描述	标志清除机制
		1KHz_intf	1KHz pulse flag	
		32KHz_intf	32KHz pulse flag	
14	USB_DEV	USB_DEV Flag	USB_DEV Flag	
15	I2C	I2C Flag	I2C Flag	
16	EPORT2	EPF0	Edge port 2 flag 0	Write EPF0 = 1
		EPF1	Edge port 2 flag 1	Write EPF1 = 1
		EPF2	Edge port 2 flag 2	Write EPF2 = 1
		EPF3	Edge port 2 flag 3	Write EPF3 = 1
		EPF4	Edge port 2 flag 4	Write EPF4 = 1
		EPF5	Edge port 2 flag 5	Write EPF5 = 1
		EPF6	Edge port 2 flag 6	Write EPF6 = 1
		EPF7	Edge port 2 flag 7	Write EPF7 = 1
17	PVD	PVDO	PVD Flag	
18	CANBUS	CAN_IFRH[BUF63:BUF0]	CAN buffers 63–0 interrupts	This bit is cleared by writing it to ‘1’
19	CANBUS	CAN_ESR[BOFF_INT]	CANBus off interrupt	This bit is cleared by writing it to ‘1’
20	CANBUS	CAN_ESR[ERR_INT]	CAN error interrupt	This bit is cleared by writing it to ‘1’
21	CANBUS	CAN_ESR[TWRN_INT]	CAN transmit warning interrupt	This bit is cleared by writing it to ‘1’
22	CANBUS	CAN_ESR[RWRN_INT]	CAN receive warning interrupt	This bit is cleared by writing it to ‘1’
23	CANBUS	CAN_ESR[WKUP_INT]	Wake Up Interrupt	This bit is cleared by writing it to ‘1’
24	BLENDE	INT_END	Transfer Complete	
		INT_OQ	One quarter of the blending operation of the currently data source is completed	
		INT_HALF	Halt of the blending operation of the currently data source is completed	
25	RGBC	INTV	The completion flag of one frame transmission	
		INTN	Transmission completion flag for the specified number of frames	
26	QSPI1	XRXOIS	XIP Receive FIFO Overflow Interrupt	

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

根源	模块	标志	来源描述	标志清除机制
		RXFIS	Receive FIFO Full Interrupt	
		RXOIS	Receive FIFO Overflow Interrupt	
		RXUIS	Receive FIFO Underflow Interrupt	
		TXOIS	Transmit FIFO Overflow Interrupt	
		TXEIS	Transmit FIFO Empty Interrupt	
27	QSPI2	XRXOIS	XIP Receive FIFO Overflow Interrupt	
		RXFIS	Receive FIFO Full Interrupt	
		RXOIS	Receive FIFO Overflow Interrupt	
		RXUIS	Receive FIFO Underflow Interrupt	
		TXOIS	Transmit FIFO Overflow Interrupt	
		TXEIS	Transmit FIFO Empty Interrupt	
28	SCI1	TDRE	Transmit Data Register Empty Flag	
		TC	Transmission Complete	
		TXOF	Transmitter Buffer Overflow Flag	
		LBKDIF	LIN Break Detect Interrupt Flag	
		IDLE	Idle Line Flag	
		RXEDGIF	RXD1 Pin Active Edge Interrupt Flag	
		RDRF	Receive Data Register Full Flag	
		MA1F	Match 1 Flag	
		MA2F	Match 2 Flag	
		OR	Receiver Overrun Flag	
		NF	Noise Flag	
		FE	Framing Error Flag	
		PF	Parity Error Flag	
		RXUF	Receiver Buffer Underflow Flag	
29	SCI2	TDRE	Transmit Data Register Empty Flag	

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

根源	模块	标志	来源描述	标志清除机制
		TC	Transmission Complete	
		TXOF	Transmitter Buffer Overflow Flag	
		LBKDIF	LIN Break Detect Interrupt Flag	
		IDLE	Idle Line Flag	
		RXEDGIF	RXD2 Pin Active Edge	
			Interrupt Flag	
		RDRF	Receive Data Register Full Flag	
		MA1F	Match 1 Flag	
		MA2F	Match 2 Flag	
		OR	Receiver Overrun Flag	
		NF	Noise Flag	
		FE	Framing Error Flag	
		PF	Parity Error Flag	
		RXUF	Receiver Buffer Underflow Flag	
30	EPORT0	EPF0	Edge port 0 Flag 0	Write EPF0 = 1
		EPF1	Edge port 0 Flag 1	Write EPF1 = 1
		EPF2	Edge port 0 Flag 2	Write EPF2 = 1
		EPF3	Edge port 0 Flag 3	Write EPF3 = 1
		EPF4	Edge port 0 Flag 4	Write EPF4 = 1
		EPF5	Edge port 0 Flag 5	Write EPF5 = 1
		EPF6	Edge port 0 Flag 6	Write EPF6 = 1
		EPF7	Edge port 0 Flag 7	Write EPF7 = 1
31	EPORT1	EPF0	Edge port 1 Flag 0	Write EPF0 = 1
		EPF1	Edge port 1 Flag 1	Write EPF1 = 1
		EPF2	Edge port 1 Flag 2	Write EPF2 = 1
		EPF3	Edge port 1 Flag 3	Write EPF3 = 1
		EPF4	Edge port 1 Flag 4	Write EPF4 = 1
		EPF5	Edge port 1 Flag 5	Write EPF5 = 1
		EPF6	Edge port 1 Flag 6	Write EPF6 = 1
		EPF7	Edge port 1 Flag 7	Write EPF7 = 1

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

6. 32 位 RISC 微处理器介绍

本节描述了 32 位 RISC 微处理器的功能，它基于 M*Core 指令集/架构，专为极低功耗和成本敏感的嵌入式控制应用程序而设计。对于较小的尺寸和功耗，32 位 RISC 是建立在一个新的 3 级管道冯-诺伊曼(von-Neumann)架构上。

32 位的 RISC 还集成了一个 EIC（嵌入式中断控制器），以减少系统面积。外部总线接口协议为 AHB-lite。在 32 位的 RISC 设计中，有更多可配置的选项。通过利用这些可配置的选项，在性能、功能和成本之间的权衡将更加灵活。32 位 RISC 的门计数从 12K 到 20K 不等。

6.1. 功能特点

32 位 RISC 的主要特点如下：

- 32 位负载/存储减少指令集计算机（RISC）体系结构，具有固定的 16 位指令长度
- 16 条目 32 位通用寄存器文件
- 高效的 3 阶段执行管道，隐藏在应用软件中
- 单周期指令执行多个指令，分支执行三个周期
- 支持字节/半字段的内存访问
- 嵌入式中断控制器，支持嵌套的向量中断和低功耗模式唤醒
- 单周期 32 位 x32 位硬件整数乘法器数列
- 3~13 循环硬件整数分法器阵列
- AHB-lite 外部总线

6.2. 微体系结构总结

32 位的 RISC 利用一个 3 阶段的管道来执行指令。指令获取、指令解码/寄存器文件读取和执行/回写阶段以重叠的方式运行，允许对大多数指令执行单个时钟指令。

为源操作数和指令结果提供了 16 个通用寄存器。寄存器 R15 被用作链接寄存器，以保持子程序调用的返回地址，并且寄存器 R0 按约定与当前堆栈指针值相关联。提供了一个双入口 32 位指令缓冲区，以允许指令预取从每个时钟周期的存储器中获得两个指令，从而减少或消除与数据存储器访问的总线资源冲突。统一的总线结构足以维持指令和数据带宽的需求，而不诉诸于昂贵的双总线结构。

为字节、半字和字（32 位）数据提供了内存加载和存储操作，字节和半字加载数据。这些指令可以流水线化，以允许短序列有效的单周期吞吐量。与数据相关的操作可以在两个时钟周期内完成。加载和存储多个寄存器指令允许低开销的上下文保存和恢复操作；这些指令可以在（N 个+1）时钟周期中执行，其中 N 是要传输的寄存器数。提供了一个单一的条件代码/携带(C)位，用于条件测试和用于实现大于 32 位的算术和逻辑操作。通常，C 位仅通过显式的测试/比较操作来设置，而不是作为正常指令操作的副作用。对于需要将条件设置与实际计算结合起来的专门操作，此规则会出现例外。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

6.3. 程序设计模型

32 位 RISC 编程模型分别为两种特权模式定义：管理员和用户。HPROT[1]位用于表示特权模式。程序根据所指示的模式访问寄存器。用户程序只能访问特定于用户模式的寄存器；在监视模式下执行的系统软件可以访问所有寄存器，并使用控制寄存器来执行监视功能。因此，可以限制用户程序访问特权信息，并且操作系统通过协调用户程序的活动来为用户程序执行管理和服务任务。

所有的指令都可以在任何一种模式下执行。用户程序还可以执行停止、睡觉或等待指令。陷阱#n 指令为用户程序提供对操作系统服务的受控访问。为了防止用户程序进入监督模式，除非以受控的方式，可以改变程序状态寄存器（PSR）中的 S-位的指令是有特权的。

当设置了 PSR 中的 S 位时，处理器将在监督模式下执行指令。与指令相关联的总线周期根据模式指示主管或用户访问。处理器在正常用户模式处理时利用用户编程模型。在异常处理期间，处理器将从用户模式更为监控模式。异常处理保存 PSR 的当前值以堆栈内存，然后在 PSR 中设置 S 位，迫使处理器进入监督模式。要返回到以前的操作模式，系统例程可能会执行 RTE（从异常返回）指令，导致指令管道从适当的地址空间被刷新和重新填充。



图 6-1：编程模型

编程模型中描述的寄存器（参见图 6-1)提供了操作数存储和控制。用户编程模型由 16 个通用的 32 位寄存器、32 位程序计数器（PC）和条件/携带(C)位组成。C 位被实现为 PSR 的第 0 位。按照惯例，寄存器 R15 被用作子例程调用的链接寄存器，而寄存器 R0 通常被用作当前的堆栈指针。

©Levetop Semiconductor Co., Ltd.

6.4. 数据格式汇总

整数单位支持的操作数数据格式是标准的 2 的补充数据格式。每个指令的操作数大小可以用指令显式编码（加载/存储指令），也可以由指令操作（索引操作、字节提取）隐式定义。通常，指令对源操作数的所有 32 位都进行操作，并生成 32 位的结果。

根据处理器配置，可以从大或小端字节顺序角度查看内存（参见图 6-2）。在 Big Endian 模式（默认操作模式）中，字符 0 的最重要字节（字节 0）位于地址 0 处。对于小环境模式，字 0 的最重要的再见位于地址 3。在寄存器中，位在以位开始的字中编号（参见图 6-3）。按照惯例，寄存器的字节 0 如何都是最重要的字节。这只是在执行 XTRB[0-3]指令时出现的一个问题。

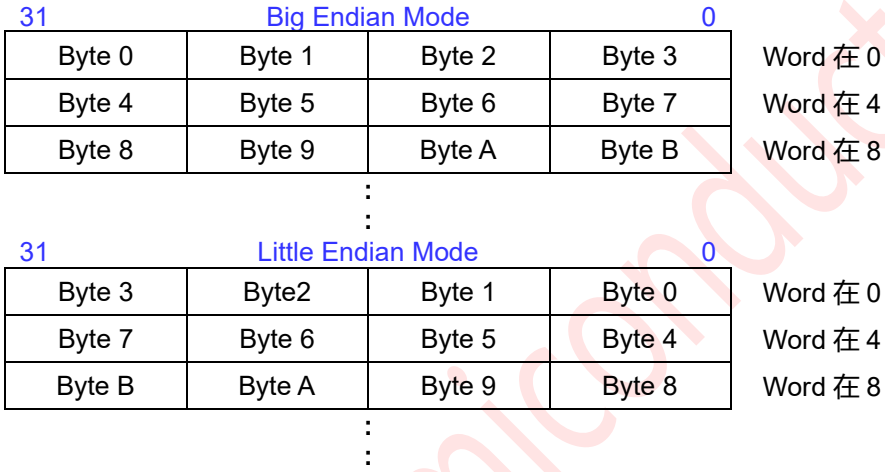


图 6-2：内存中的数据结构

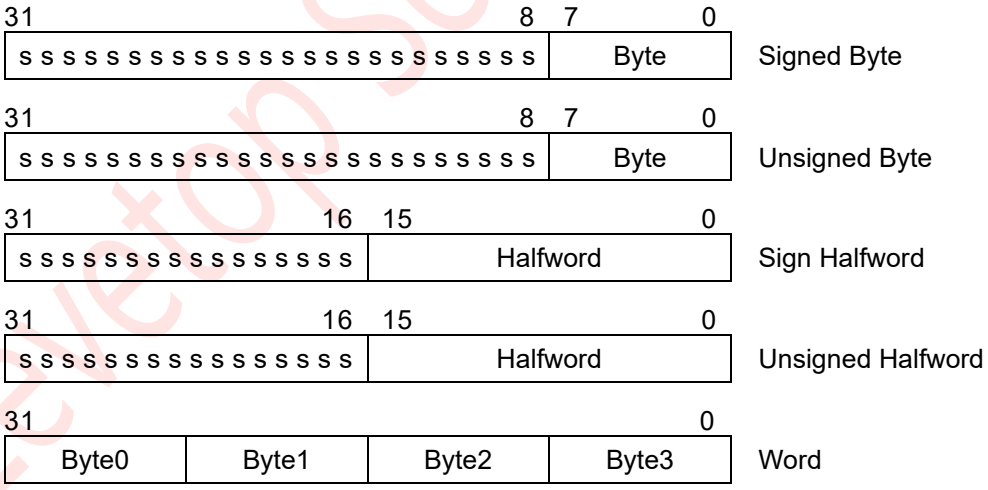


图 6-3：寄存器中的数据结构

©Levetop Semiconductor Co., Ltd.

6.5. 操作和寻址能力

32 位的 RISC 通过加载和存储指令访问所有的内存操作数，在通用寄存器（GPRs）和内存之间传输数据。寄存器+4 位缩放位移寻址模式用于加载和存储指令来地址字节、半字或字（32 位）数据。

加载和存储多个指令允许 16 个 gpr 的一个子集传输到寄存器 R0（约定的默认堆栈指针）指向的基本地址。

加载和存储寄存器象限指令使用寄存器间接寻址传输一个寄存器象限到或从存储器。

6.6. 指令集概述

该指令集是为支持高级语言而定制的，并针对那些最常执行的指令进行了优化。提供了一套标准的算术和逻辑指令，以及对位操作、字节提取、数据移动、控制流修改的指令支持，以及一小组有条件执行的指令，可用于消除短的条件分支。

表 6-1 提供了 32 位 RISC 指令集的字母顺序列表。

表 6-1：32 位 RISC 指令集

助记符号	描 述
ABS	Absolute Value
ADDC	Add with C bit
ADDI	Add Immediate
ADDU	Add Unsigned
AND	Logical AND
ANDI	Logical AND Immediate
ANDN	AND NOT
ASR	Arithmetic Shift Right
ASRC	Arithmetic Shift Right, update C bit
ASRI	Arithmetic Shift Right Immediate
BCLRI	Clear Bit
BF	Branch on Condition False
BGENI	Bit Generate Immediate
BGENR	Bit Generate Register
BKPT	Breakpoint
BMASKI	Bit Mask Immediate
BR	Branch
BGENI	Bit Generate Immediate
BGENR	Bit Generate Register
BKPT	Breakpoint
BMASKI	Bit Mask Immediate
BR	Branch
BREV	Bit Reverse
BSETI	Bit Set Immediate
BSR	Branch to Subroutine
BT	Branch on Condition True
BTSTI	Bit Test Immediate

©Levetop Semiconductor Co., Ltd.

助记符号	描 述
CLRF	Clear Register on Condition False
CLRT	Clear Register on Condition True
CMPHS	Compare Higher or Same
CMPLT	Compare Less-Than
CMPLTI	Compare Less-Than Immediate
CMPNE	Compare Not Equal
CMPNEI	Compare Not Equal Immediate
DECF	Decrement on Condition False
DECGT	Decrement Register and Set Condition if Result Greater-than Zero
DECLT	Decrement Register and Set Condition if Result Less-than Zero
DECNE	Decrement Register and Set Condition if Result Not Equal to Zero
DECT	Decrement On Condition True
DIVS (1)	Divide Signed Integers
DIVU (1)	Divide Unsigned Integers
DOZE	Doze
FF1 (1)	Find First One
INCF	Increment on Condition False
INCT	Increment On Condition True
IXH	Index Halfword
IXW	Index Word
JAVASW	Java interpreter switch
JMP	Jump
JMPI	Jump Indirect
JSR	Jump to Subroutine
JSRI	Jump to Subroutine Indirect
LD.[BHW]	Load
LDM	Load Multiple Registers
LDQ	Load Register Quadrant
LRW	Load Relative Word
LSL, LSR	Logical Shift Left and Right
LSLC, LSRC	Logical Shift Left and Right, update C bit
LSLI, LSRI	Logical Shift Left and Right by Immediate
MFCR	Move from Control Register
MOV	Move
MOVI	Move Immediate
MOVF	Move on Condition False
MOVT	Move on Condition True
MTCR	Move to Control Register
MULSH	Multiply signed Halfwords

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

助记符号	描 述
MULT	Multiply
MVC	Move C bit to Register
MVCV	Move Inverted C bit to Register
NOT	Logical Complement
OR	Logical Inclusive-OR
ROTLI	Rotate Left by Immediate
RSUB	Reverse Subtract
RSUBI	Reverse Subtract Immediate
RTE	Return from Exception
RFI	Return from Interrupt
SEXTB	Sign-extend Byte
SEXTH	Sign-extend Halfword
ST.[BHW]	Store
STM	Store Multiple Registers
STQ	Store Register Quadrant
STOP	Stop
SUBC	Subtract with C bit
SUBU	Subtract
SUBI	Subtract Immediate
SYNC	Synchronize
TRAP	Trap
TST	Test Operands
TSTNBZ	Test for No Byte Equal Zero
WAIT	Wait
XOR	Exclusive OR
XSR	Extended Shift Right
XTRB0	Extract Byte 0
XTRB1	Extract Byte 1
XTRB2	Extract Byte 2
XTRB3	Extract Byte 3
ZEXTB	Zero-extend Byte
ZEXTH	Zero-extend Halfword

提示(1)： 在当前版本中未实现。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

7. 嵌入式可编程定时器（EPT）

7.1. 基本介绍

嵌入式可编程定时器（EPT）是一个 24 位定时器，提供精确的中断在定期间隔和最小的处理器干预。计时器可以从重新加载值开始倒数，也可以是一个自由运行的倒数计数器。

EPT 中断可以触发一个异常（向量号 = 24）。

EPT 模块可以通过清除参数“EPT”到 0，以减少核心门数。

7.2. 内存映射和寄存器

7.2.1. 内存映射

EPT 基地址定义为 EIC_BASEADDR + 0x1000。默认的基于地址（EPT_BASEADDR）是 0xE000_1000。表 7-1 显示了 EPT 寄存器的偏移量地址。EPT 模块占用 4K 地址区。

表 7-1：可编程定时器模块内存配置

偏移地址	Bits[31:0]	访问权限
0x0000_0000	EPT Control and Status Register (EPTCSR)	S/U
0x0000_0004	EPT Reload Register (EPTRLD)	S/U
0x0000_0008	EPT Count Register (EPTCNT)	S/U
0x0000_000 C	保留	S/U

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

7.2.2. 寄存器描述

本节包含了对 EPT 模块寄存器的描述。

7.2.2.1. EPT 控制状态寄存器（EPTCSR）

地址：E PT_BASEADDR+0x0000_0000

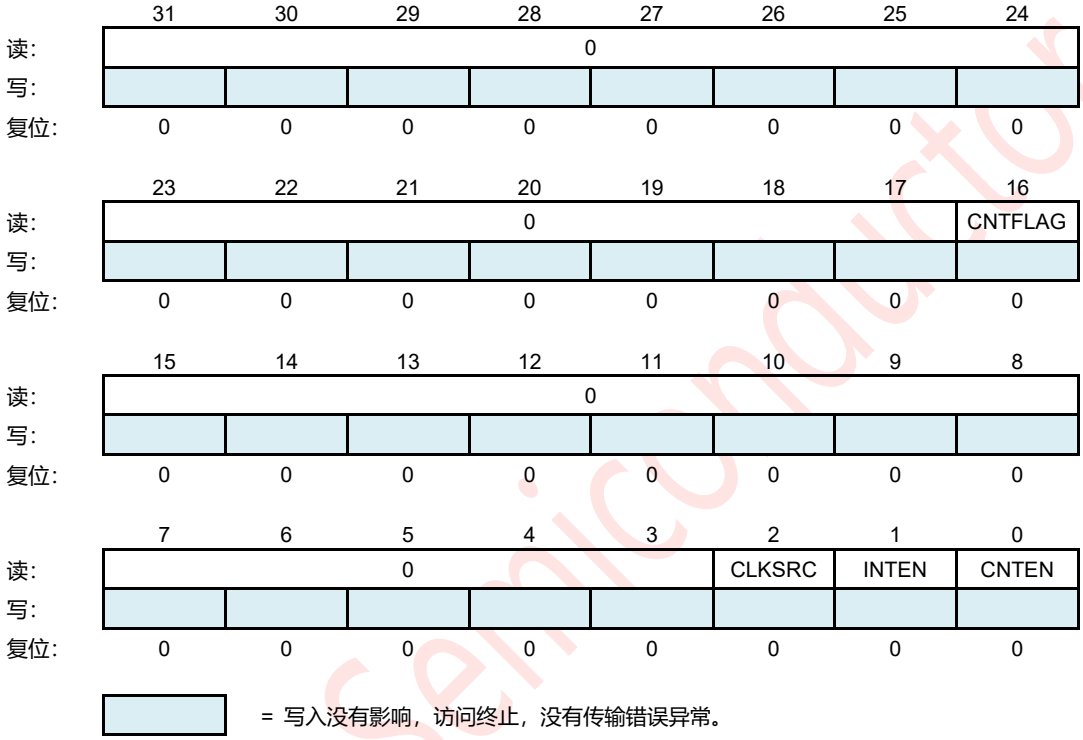


图 7-1：EPT 控制状态寄存器（EPTCSR）

CNTFLAG — Count Down to 0 flag

只读位表示已计数的计时器为 0。它将被 RESET#复位。

1 = 计时器倒数到 0。

0 = 计时器仍在倒计时。

CLKSRC — Count clock source select

读写位用于选择计数时钟源。它将被 RESET#复位。

1 = 核心时钟。

0 = 外部参考时钟。

INTEN — EPT Interrupt Request Enable

读/写位用于启用 EPT 的中断，当计时器倒数到 0，它将通过复位复位。

1 = 当计时器倒数到 0 时，会发生 = EPT 异常请求。

0 = 当计时器计数到 0 时，不会发生 = EPT 异常请求。

©Levetop Semiconductor Co., Ltd.

CNTEN — Counter Enable

读/写位用于启用 EPT 的计数器。它将通过 RESET#来进行复位。

1 = 计数器已启用

0 = 计数器已启用

7.2.2.2. EPT 重新加载寄存器（EPTRLD）

地址：EPT _BASEADDR+0x0000_0004

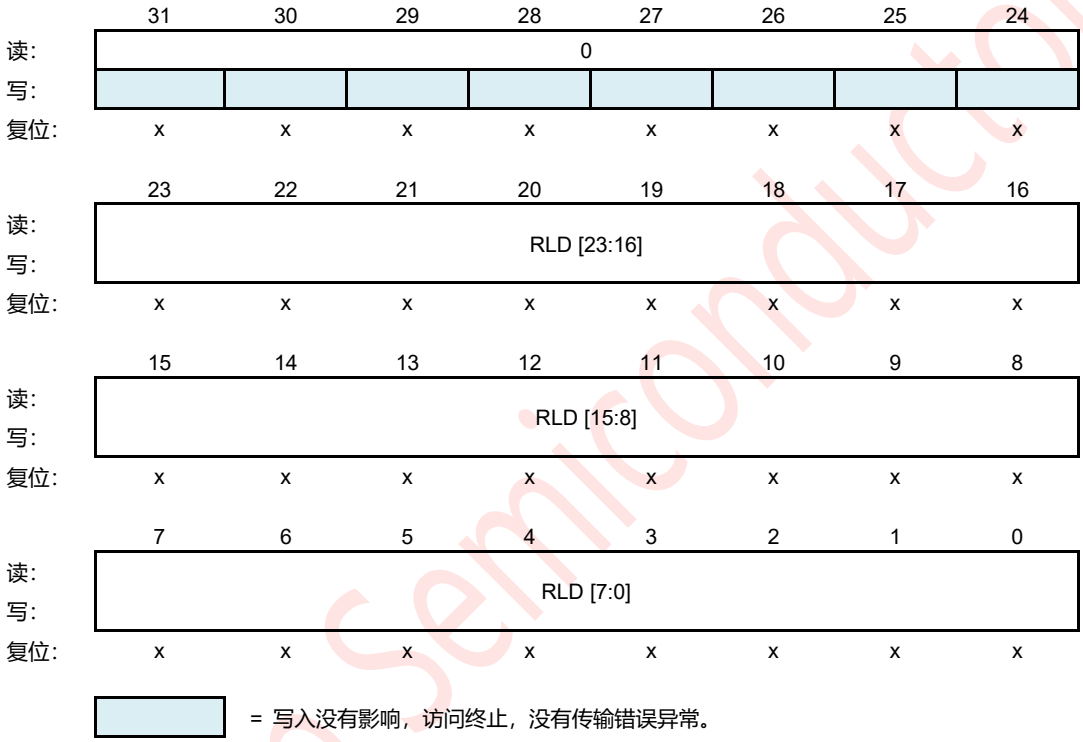


图 7-2：EPT 重新加载寄存器（EPTRLD）

RLD[23:0] — Reload Value

读/写 RLD 字段指定当计时器倒数到 0 时的重新加载值。该寄存器没有复位值。RLD 值可以是 0x000000001~0x00 FFFFFF 范围内的任何值。值 0 没有任何影响。要生成具有 N 个时钟周期的周期定时器，请将 RLD 设置为 N-1。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

7.2.2.3. EPT 计数寄存器（EPTCNT）

地址：EPT _BASEADDR+0x0000_0008

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	x	x	x	x	x	x	x	x

	23	22	21	20	19	18	17	16
读:	CNT[23:16]							
写:								
复位:	x	x	x	x	x	x	x	x

	15	14	13	12	11	10	9	8
读:	CNT[15:8]							
写:								
复位:	x	x	x	x	x	x	x	x

	7	6	5	4	3	2	1	0
读:	CNT[7:0]							
写:								
复位:	x	x	x	x	x	x	x	x

= 写入没有影响，访问终止，没有传输错误异常。

图 7-3：EPT 计数器寄存器（EPTCNT）

CNT[23:0] — EPT Counter Value

只读寄存器表示 EPT 计时器的当前计数值。该寄存器没有复位值。

读取计数器将返回 EPT 计数器的当前值。写入此寄存器的任何值将将计数器值清除为 0，并将 CNTFLAG 清除为 0。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

7.3. 功能描述

启用后，EPT 计数从 RLD 设置的值降为零，并在下一个时钟周期中重新加载 RLD 中的值，然后通过随后的时钟周期降低计数，将零写入 RLD 在下一次封装时禁用计数器。当 EPT 计数 s 为 0 时，CLFAG 位将被设置为 1，然后如果启用了 INTEN，则 EPT 将触发 EPT 中断。

读取 CSR 将 CFLAG 位清除为 0。将任何值写入 CNT 也会将 CFLAG 位清除为 0。

7.3.1. 计数时序

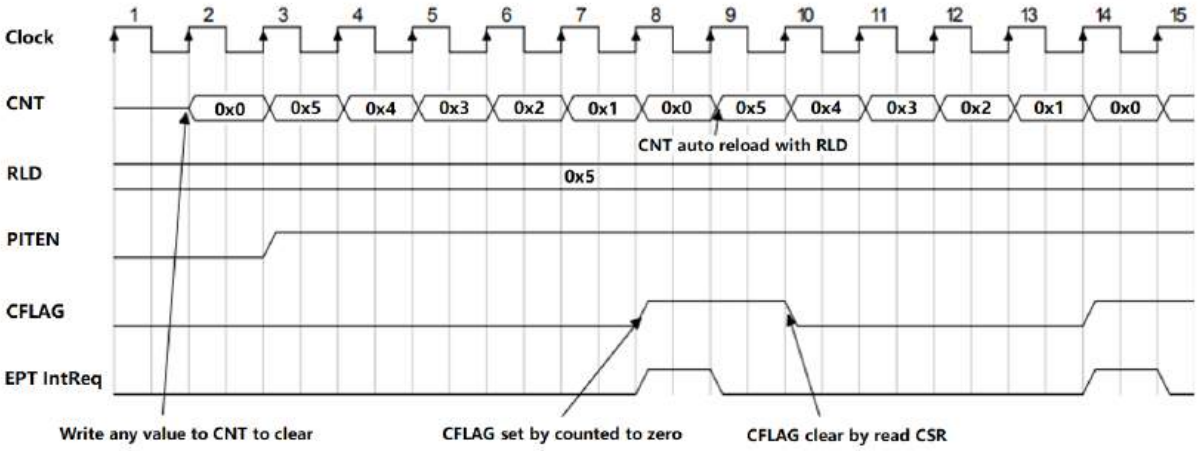


图 7-4：EPT 计数时序

©Levetop Semiconductor Co., Ltd.

8. 芯片配置模块（CCM）

8.1. 基本介绍

LT168 的芯片配置模块（CCM）控制着芯片的配置。

8.2. 功能特点

CCM 可执行以下操作：

- 配置唤醒功能
- 配置 LDO 模式
- 配置 IO 功能

8.3. 内存映射和寄存器

本节提供了对内存映射和寄存器的描述。CCM 基地址为 0x4001_0000。表 8-1 显示了 CCM 寄存器的偏移量地址。

8.3.1. 内存映射

表 8-1：CCM 内存映射

偏移地址	Bits[31:16]	Bits[15:0]	访问权限 [1]
0x0000	WKUPC — Wakeup Configuration Register		S
0x0004	保留		S
0x0008	保留		S
0x000C	CPPDC — Chip Pin Pull Down Configuration Register		S
0x0010	保留		S
0x0014	QSPIXIPCR — QSPI XIP Mode Configuration Register		S
0x0018	保留		S
0x001C	QSPIKEYR — QSPI 32-Bit Key Register		S
0x0020	QSPIGPIOCR — QSPI GPIO Configuration Register		S
0x0024	MCURAMPRIOCR — MCU Access RAM Priority Configuration Register		S
0x0028	EPORT2FCR — EPORT2 Function Configuration Register		S

提示：

S = 仅主管访问。在用户模式下，只访问主管的地址位置没有影响，并导致周期终止传输错误。

©Levetop Semiconductor Co., Ltd.

8.3.2. 寄存器描述

8.3.2.1. 唤醒配置寄存器（WKUPC）

地址：CCM_BASEADDR+0x0000_0000

	31	30	29	28	27	26	25	24
读: 写:	WKUPFILTEREN	WKUPSEN [30:24]						
复位:	0	1	1	1	1	1	1	1

	23	22	21	20	19	18	17	16
读: 写:	WKUPSEN [23:16]							
复位:	1	1	1	1	1	1	1	1

	15	14	13	12	11	10	9	8
读: 写:	WKUPSEN [15:8]							
复位:	1	1	1	1	1	1	1	1

	7	6	5	4	3	2	1	0
读: 写:	WKUPSEN [7:0]							
复位:	1	1	1	1	1	1	1	1

图 8-1：唤醒配置寄存器（WKUPC）

WKUPFILTEREN — Wakeup Source Filter Enable

如果设置了电源，唤醒源将通过过滤器去除玻璃，然后从待机模式唤醒芯片。

1 = 唤醒源过滤器

0 = 唤醒源过滤器已被禁用

WKUPSEN[30:0] — Wakeup Source Enable

该场控制是否使用相应的源作为源，从待机模式唤醒芯片。如果设置，则将相应的源用作唤醒源。

表 8-2：WKUPSEN 和相应的唤醒源

WKUPSEN	Wakeup Source
WKUPSEN[30]	INT2[3]
WKUPSEN[29]	INT2[2]
WKUPSEN[28]	INT2[1]
WKUPSEN[27]	RTC Interrupt
WKUPSEN[26]	PVD Interrupt
WKUPSEN[25]	USB Resume
WKUPSEN[24]	COMP1 Interrupt
WKUPSEN[23]	COMP0 Interrupt
WKUPSEN[22]	INT2[6]

©Levetop Semiconductor Co., Ltd.

WKUPSEN	Wakeup Source
WKUPSEN[21]	INT2[5]
WKUPSEN[20]	I2C
WKUPSEN[19]	WDT0 Interrupt
WKUPSEN[18]	JTAG POWER ON REQUEST
WKUPSEN[17]	RESET# Pin
WKUPSEN[16]	WDT0 Reset
WKUPSEN[15]	INT1[7]
WKUPSEN[14]	INT1[6]
WKUPSEN[13]	INT1[5]
WKUPSEN[12]	INT1[4]
WKUPSEN[11]	INT1[3]
WKUPSEN[10]	INT1[2]
WKUPSEN[9]	INT1[1]
WKUPSEN[8]	INT1[0]
WKUPSEN[7]	INT0[7]
WKUPSEN[6]	INT0[6]
WKUPSEN[5]	INT0[5]
WKUPSEN[4]	INT0[4]
WKUPSEN[3]	INT0[3]
WKUPSEN[2]	INT0[2]
WKUPSEN[1]	INT0[1]
WKUPSEN[0]	INT0[0]

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

8.3.2.2. 芯片引脚下拉配置寄存器（CPPDC）

地址：CCM_BASEADDR+0x0000_000C

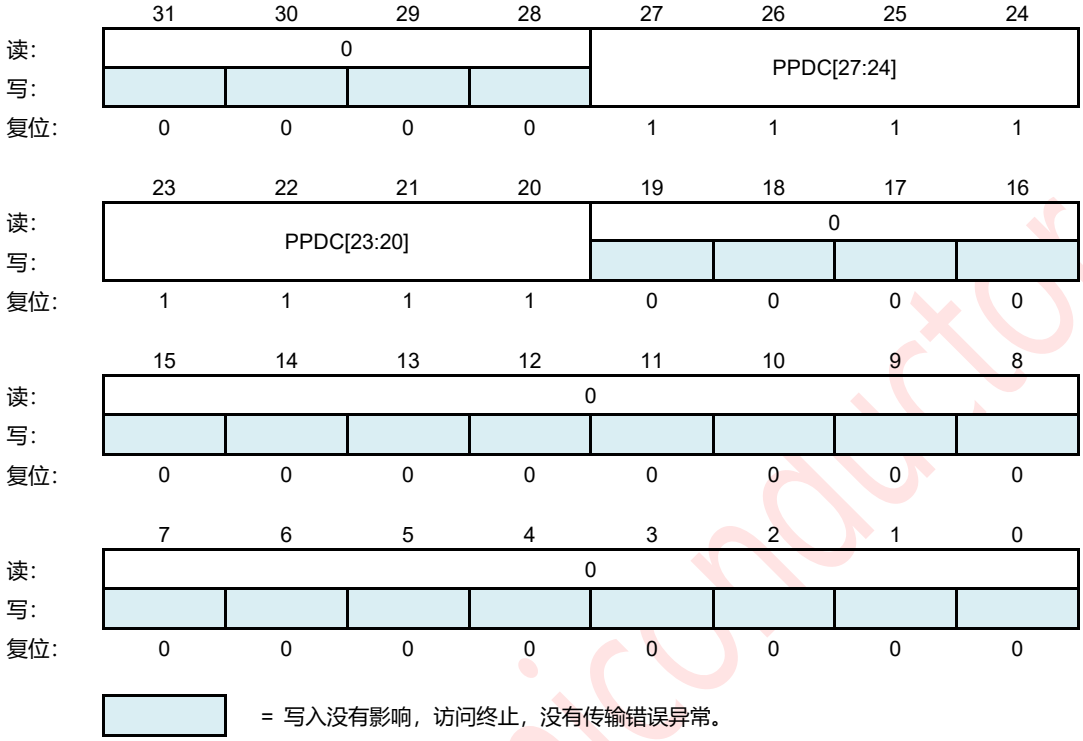


图 8-2：芯片引脚下拉配置寄存器（CPPDC）

PPDC[27:20] — Pin Pull Down Configuration Field

此读/写字段控制相应引脚的下拉功能，如表 8-3 所示。

1 = 启用相应引脚的下拉功能

0 = 相应引脚的下拉功能被禁用

表 8-3：芯片引脚下拉配置

引脚名	下拉配置位	下拉功能
PWM0[3]	PPDC[27]	0：禁用，1：启用
PWM0[2]	PPDC[26]	0：禁用，1：启用
PWM0[1]	PPDC[25]	0：禁用，1：启用
PWM0[0]	PPDC[24]	0：禁用，1：启用
PWM1[3]	PPDC[23]	0：禁用，1：启用
PWM1[2]	PPDC[22]	0：禁用，1：启用
PWM1[1]	PPDC[21]	0：禁用，1：启用
PWM1[0]	PPDC[20]	0：禁用，1：启用

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

8.3.2.3. QSPI XIP 模式配置寄存器（QSPI）

QSPIXIPMCFR 寄存器是一个可读写寄存器。

地址：CCM_BASEADDR+0x0000_0014

	31	30	29	28	27	26	25	24
读:	0						QSPI2_DATA_ENCR_EN	QSPI2_XIPEN
写:								
复位:	0	0	0	0	0	0	0	0

	23	17	16	20	19	18	17	16
读:	0						QSPI1_DATA_ENCR_EN	QSPI1_XIPEN
写:								
复位:	0	0	0	0	0	0	0	0

	15	9	8	12	11	10	9	8
读:	0						QSPI0_DATA_ENCR_EN	QSPI0_XIPEN
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 8-3：QSPI XIP 模式配置寄存器（QSPI）

QSPIx_DATA_ENCR_EN — QSPIx XIP Transfer Data Encrypt function Enable Control bit

1 = QSPIx XIP 传输数据加密功能被启用

0 = QSPIx XIP 传输数据加密功能被禁用

QSPIx_XIPEN — QSPIx XIP Mode Enable Control bit

1 = QSPIx XIP 模式启用

0 = QSPIx XIP 模式禁用

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

8.3.2.4. QSPI 32 位密钥寄存器（QSPI QSPI）

QSPPIKEYR 寄存器是一个可写的寄存器，在读取时总是返回 0's 。

地址：CCM_BASEADDR+0x0000_001C

	31	30	29	28	27	26	25	24
读:	0							
写:	QSPI_KEY[31:24]							
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0	0	0	0	0	0	0	0
写:	QSPI_KEY[23:16]							
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0	0	0	0	0	0	0	0
写:	QSPI_KEY[15:8]							
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0	0	0	0	0	0	0	0
写:	QSPI_KEY[7:0]							
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 8-4：QSPI 32 位密钥寄存器（q QSPI keyr）

8.3.2.5. QSPI GPIO 配置寄存器（QSPI）

QSPIGPIOCR 寄存器是一个可读可写的寄存器。

地址：CCM_BASEADDR+0x0000_0020

	31	30	29	28	27	26	25	24
读:	QSPI2_SS_GPIOEN	QSPI1_SS_GPIOEN	QSPI0_SS_GPIOEN	0		QSPI2_SS_PUE	QSPI1_SS_PUE	QSPI0_SS_PUE
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0					qspi2_ss_obe	QSPI1_SS_OBE	QSPI0_SS_OBE
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0					QSPI2_SS_DO	QSPI1_SS_DO	QSPI0_SS_DO
写:								
复位:	0	0	0	0	0	1	1	1

	7	6	5	4	3	2	1	0
读:	0					QSPI2_SS_DI	QSPI1_SS_DI	QSPI0_SS_DI
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 8-5：QSPI GPIO 配置寄存器（QSPIGPIOCR）

©Levetop Semiconductor Co., Ltd.

QSPIx_SS_GPIOEN — QSPIx CS# Pin GPIO Mode Enable Control bit

1 = QSPIx CS# Pin GPIO 模式已启用

0 = QSPIx CS# Pin GPIO 已禁用

QSPIx_SS_PUE — QSPIx CS# Pin Pullup Enable Control bit in GPIO Input Mode.

提示：QSPI2_SS_PUE 对 LT168 无效。

1 = QSPIx CS# Pin GPIO 拉拔已启用

0 = QSPIx CS# Pin GPIO 拔拔禁用

QSPIx_SS_OBE — QSPIx CS# Pin Output Enable Control bit in GPIO Mode

1 = QSPIx CS# Pin GPIO 输出已启用

0 = QSPIx CS# Pin GPIO 输出已禁用

QSPIx_SS_DO — QSPIx CS# Pin Output Data in GPIO Output Mode

1 = QSPIx CS#将在 GPIO 输出模式下被高驱动

0 = QSPIx CS#将在 GPIO 输出模式下被低驱动

QSPIx_SS_DI — QSPIx CS# Pin value in GPIO Mode

1 = 表明 QSPIx CS# Pin 是在 GPIO 输入模式下的高水平

0 = 指示 QSPIx CS# Pin 在 GPIO 输入模式下为低电平

8.3.2.6. MCU 访问 RAM 优先级配置寄存器（内存）

该寄存器是一个可读可写的寄存器。

地址：CCM_BASEADDR+0x0000_0024

	31	30	29	28	27	26	25	24
读:	RAMPRIOTEST[1:0]		CACHERAMPD	DISPLAYRAMPD	SYSRAM3PD	SYSRAM2PD	SYSRAM1PD	SYSRAM0PD
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0		CACHERAMISOEN	DISPLAYRAMISOEN	SYSRAM3ISOEN	SYSRAM2ISOEN	SYSRAM1ISOEN	SYSRAM0ISOEN
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0						RAM256KP	RAM512KP
写:								
复位:	0	0	0	0	0	0	1	1

= 写入没有影响，访问终止，没有传输错误异常。

图 8-6：MCU 访问 RAM 优先级配置寄存器（MCURAMPRIOCR）

©Levetop Semiconductor Co., Ltd.

RAMPRIOTEST[1:0] — MCURAMPRIOCR Write Access Sequence In

除非写入了正确的序列，否则不能更改 MCURAMPRIOCR 寄存器的可写位。正确的序列是：2'b01 → 2'b10 → 2'b11。在这两个位被这个序列写入后，两位的值== 2'b11，然后可以随意改变内存寄存器的可写位。当值等于 2'b11 时，写入 2'b00 可以清除这两个位。写入其他值没有效果，并返回 2'b11。

CACHERAMPD — The power supply of Cache RAM will be shut-down when writing 1'b to it.

DISPLAYRAMPD — Please refer to CACHERAMPD bit description.

SYSRAM3PD — Please refer to CACHERAMPD bit description.

SYSRAM2PD — Please refer to CACHERAMPD bit description.

SYSRAM1PD — Please refer to CACHERAMPD bit description.

SYSRAM0PD — Please refer to CACHERAMPD bit description.

CACHERAMISOEN — Cache RAM output value will be isolated when writing 1'b1 to it.

提示：在关闭缓存 RAM 的电源之前，缓存 RAM 的输出应被隔离，以避免泄漏电流。

DISPLAYRAMISOEN — Please refer to CACHERAMISOEN bit description.

SYSRAM3ISOEN — Please refer to CACHERAMISOEN bit description.

SYSRAM2ISOEN — Please refer to CACHERAMISOEN bit description.

SYSRAM1ISOEN — Please refer to CACHERAMISOEN bit description.

SYSRAM0ISOEN — Please refer to CACHERAMISOEN bit description.

RAM256KP — MCU Accesses System RAM256K Priority Configuration Register

这位确定当单片机和搅拌机同时访问系统 RAM256K 时，是否具有更高的优先级。

1 = 当单片机和搅拌机同时访问系统 RAM256K 时，单片机具有更高的优先级。

0 = 单片机和搅拌机同时访问系统 RAM256K 时，单片机的优先级较低。

RAM512KP — MCU Accesses Display RAM512K Priority Configuration Register

该位确定当单片机和搅拌机同时访问显示 RAM512K 时，MCU 是否具有更高的优先级。

1 = 单片机和搅拌机同时访问显示 RAM512K 时，具有更高的优先级。

0 = 单片机和搅拌机同时访问显示 RAM512K 时，单片机的优先级较低。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

8.3.2.7. EPORT2 功能配置寄存器（EPORT2FCR）

EPORT2FCR 寄存器是一个可读写寄存器。

地址：CCM_BASEADDR+0x0000_0028

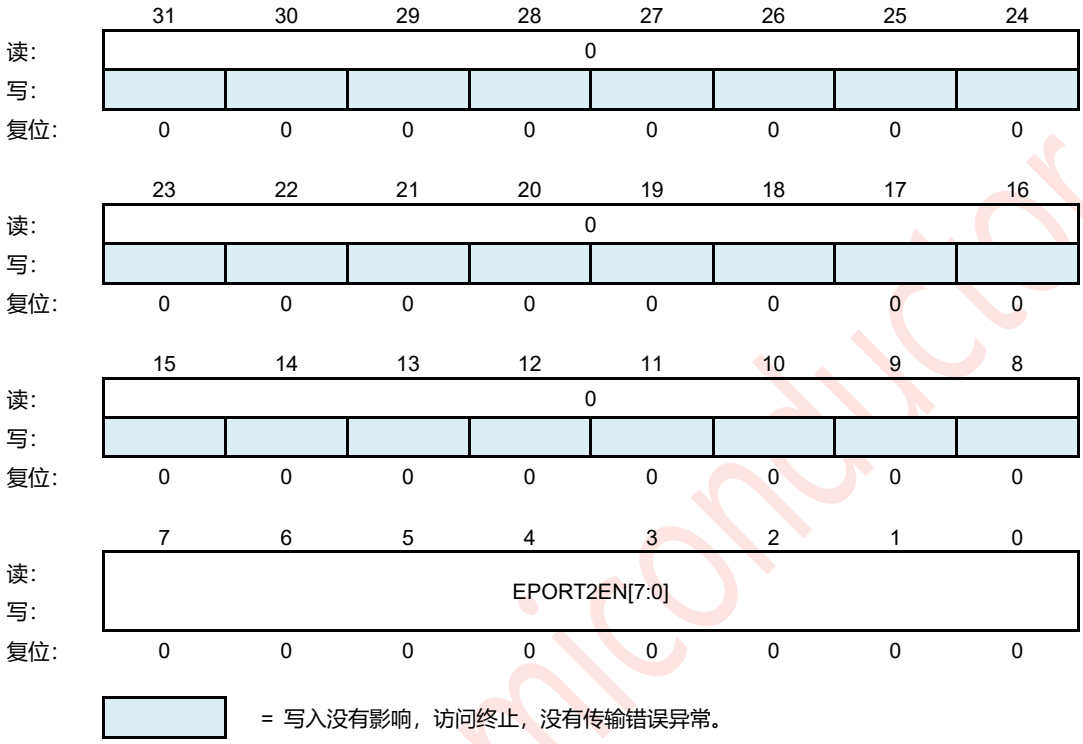


图 8-7：EPORT2 功能配置寄存器（EPORT2FCR）

EPORT2EN[7:0] — EPORT2 Function Enable control bit.

表 8-4：EPORT2 功能控制位

EPORT2 Enable Bit	1’b1	1’b0
EPORT2EN[7]	INT2[7]	CANRX
EPORT2EN[6]	INT2[6]	CANTX
EPORT2EN[5]	INT2[5]	RXD2
EPORT2EN[4]	INT2[4]	TXD2
EPORT2EN[3]	INT2[3]	RXD1
EPORT2EN[2]	INT2[2]	TXD1
EPORT2EN[1]	INT2[1]	RXD0
EPORT2EN[0]	INT2[0]	TXD0

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

9. 时钟和电源控制模块（CLKPWRM）

9.1. 基本概述

时钟模块包含：

- PLL：内部的 VCO PLL
- FXOSC：外部快速速度晶体振荡器（12 MHz）
- SIRC：内部低速 128Khz 振荡器
- SXOSC：外部低速晶体振荡器（32,768 Hz）
- 状态和控制寄存器
- 时钟和电源控制逻辑

9.2. 功能特点

该时钟模块的功能包括：

- 两个系统时钟源
  - 内部 PLL 时钟
  - 外部快速速度晶体振荡器（FXOSC）
- IPS、系统和 ADC 时钟的独立时钟分频器
- 支持低功耗模式
- 模块可以通过设置 MSCR 来单独停止

9.3. 时钟结构

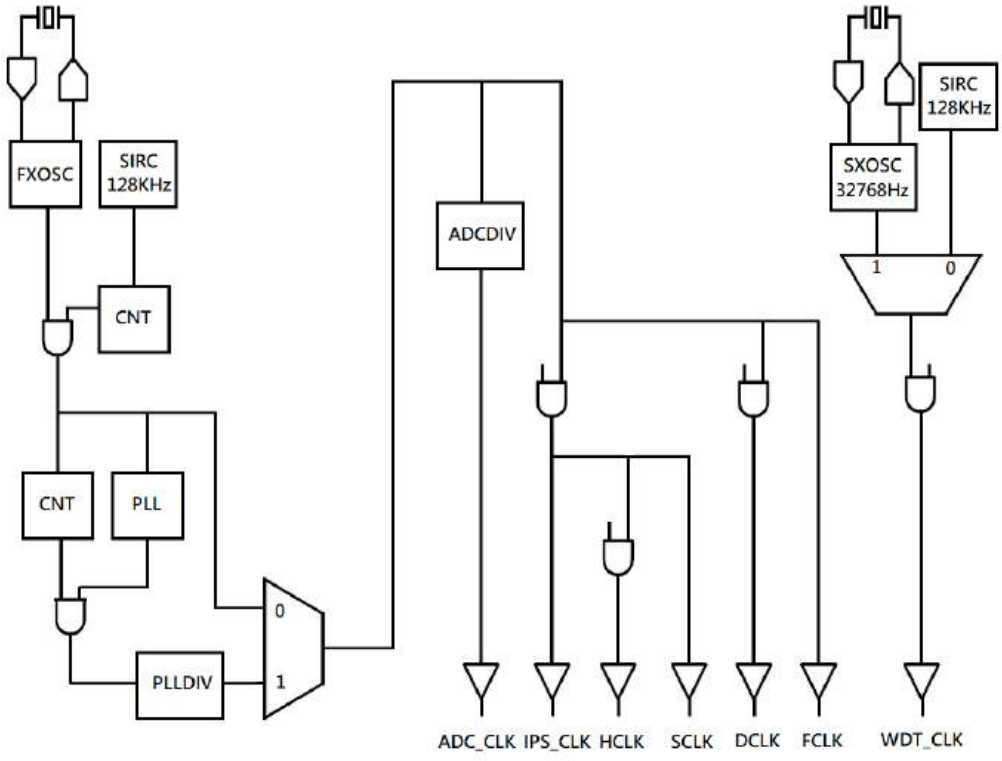


图 9-1：时钟结构

©Levetop Semiconductor Co., Ltd.

9.4. 时钟源选择

系统时钟源可以是内部 PLL 时钟或外部高速晶体振荡器(FXOSC)。时钟源选择是基于 SYNCR 寄存器的 PLLEN 位。如果设置了 PLLEN 位，则内部 PLL 是系统时钟源，否则系统时钟源是外部高速晶体振荡器（FXOSC）。

9.4.1. 低功耗选项

9.4.1.1. 等待和多兹模式

在等待和休眠模式下，系统到外设的时钟被启用，到 CPU、ROM、SRAM 的时钟被停止。每个模块都可以在模块级或通过设置 MSCR 来在本地禁用模块时钟。

9.4.1.2. 停止模式

在停止模式下，将禁用所有系统时钟。
提示：在停止模式下，不要编写程序或删除 EFLASH。

9.5. 内存映射和寄存器

时钟编程模块由以下寄存器组成：

- 合成器控制寄存器（SYNCR）
- 低速振荡器控制寄存器（LOSCCR）
- PLL 配置和状态寄存器（PLLCSR）
- 模块停止控制寄存器（MSCR）
- EPT 外部时钟源启用控制寄存器（ECSECR）
- OSC Bist 测试配置寄存器 1（OBTCR1）
- OSC Bist 测试配置寄存器 2（OBTCR2）
- OSC Bist 测试控制寄存器（OBTCTLR）
- OSC Bist 测试计数器寄存器（OBTCNTR）
- OSC Bist 测试结果寄存器（OBTRR）

本节介绍了时钟模块的内存映射和寄存器。时钟模块的基本地址为 0x4003_0000。表 9-1 显示了时钟寄存器的偏移量地址。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

9.5.1. 内存映射

表 9-1：时钟内存映射

地址	Bits[31:0]	访问权限 [1]
0x0000	Synthesizer Control Register (SYNCR)	S
0x0004	Low Speed Oscillator Control Register (LOSCCR)	S
0x0008	PLL Configuration and Status Register(PLLCSR)	S
0x000C	Module Stop Control Register (MSCR)	S
0x0010	EPT External Clock Source Enable Control Register(ECSECR)	S
0x0014	OSC Bist Test Configuration Register1(OBTCR1)	S
0x0018	OSC Bist Test Configuration Register2(OBTCR2)	S
0x001C	OSC Bist Test Control Register(OBTCTLR)	S
0x0020	OSC BIST Test Counter Register(OBTCNTR)	S
0x0024	OSC BIST Test Result Register(OBTRR)	S

提示：

S = 仅主管访问。在用户模式下，访问主管只有地址位置没有影响，并导致一个周期终止传输错误。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

9.5.2. 寄存器说明

本节提供了对时钟模块寄存器的描述。

9.5.2.1. 合成器控制寄存器（SYNCR）

合成器控制寄存器（SYNCR）始终为读/写。

地址：CLOCK_BASEADDR+0x0000_0000

	31	30	29	28	27	26	25	24
读:	SYNCTEST[1:0]		SYSRAM3 LPEN	SYSRAM2 LPEN	SYSRAM1 LPEN	SYSRAM0 LPEN	PLLOCKM	ENLOWPOWER
写:								
复位:	0	0	0	0	0	0	0	1

	23	22	21	20	19	18	17	16
读:	PLLDIV[5:0]						0	0
写:								
复位:	0	0	0	0	0	1	0	0

	15	14	13	12	11	10	9	8
读:	ADCDIV[3:0]				LOSCEN	PLLSRCEN	PLLEN	SLEEP
写:								
复位:	0	0	1	0	1	1	0	0

	7	6	5	4	3	2	1	0
读:	DISPLAYRAMLPEN	CLKOUTSEL	STBYMD[1:0]		CACHERAMLPEN	ADCEN	0	LOSCLPEN
写:								
复位:	1	0	1	1	1	1	0	1

= 写入没有影响，访问终止，没有传输错误异常。

图 9-2：合成器控制寄存器（SYNCR）

SYNCTEST[1:0] — SYNCR Write Access Sequence In

不能更改 SYNCR 寄存器的可写位，除非写入正确的序列。正确的序列是：2'b01 → 2'b10 → 2'b11。在这两个位被这个序列写入之后，两位的值== 2'b11，那么 SYNCR 寄存器的可写位可以随意更改。当值等于 2'b11 时，写入 2'b00 可以清除这两个位。写入其他值没有效果，并返回 2'b11。→→

SYSRAM3LPEN — 系统 RAM3（地址范围从 0x0083_0000 到 0x83_FFFF）芯片进入低功耗模式时的低功耗启用位。

系统 RAM3 的低功耗模式是通过关闭 RAM 的电源来实现的，RAM 的内容将丢失。

1 = 当芯片进入低功耗模式时，将启用 SYSRAM3 的低功耗模式。

0 = SYSRAM3 在芯片进入低功耗模式时失效。

SYSRAM2LPEN — 系统 RAM2（地址范围从 0x0082_0000 到 0x82_FFFF）芯片进入低功耗模式时的低功耗启用位。

系统 RAM2 的低功耗模式是通过关闭 RAM 的电源来实现的，RAM 的内容将丢失。

©Levetop Semiconductor Co., Ltd.

1 = 当芯片进入低功率模式时，SYSRAM2 将启用低功率模式。

0 = 当芯片进入低功耗模式时，将禁用 SYSRAM2 的低功耗模式。

SYSRAM1LPEN — 系统 RAM11（地址范围从 0x0081_0000 到 0x81_FFFF）芯片进入低功耗模式时的低功耗启用位。

系统 RAM1 的低功耗模式是通过关闭 RAM 的电源来实现的，RAM 的内容将丢失。

1 = 当芯片进入低功率模式时，SYSRAM1 将启用低功率模式。

0 = 当芯片进入低功耗模式时，将禁用 SYSRAM1 的低功耗模式。

SYSRAM0LPEN — 系统 RAM3（地址范围从 0x0080_0000 到 0x80_FFFF）芯片进入低功耗模式时的低功耗启用位。

系统 RAM0 的低功耗模式是通过关闭 RAM 的电源来实现的，RAM 的内容将丢失。

1 = 当芯片进入低功率模式时，将启用 SYSRAM0 的低功率模式。

0 = SYSRAM0 在芯片进入低功耗模式时失效。

PLLOCKM — PLL Lock detection flag which is generated by PLL macro.

1 = PLL Lock 是在设置 PLLEN 时由 PLL 宏生成的。

0 = 当设置了 PLLEN 时，PLL 宏不会生成 PLL 锁。

ENLOWPOWER — Enable Enter Low power mode status Bit

在系统进入待机模式之前，请确保设置了此位，否则将无法成功进入低功耗模式。此位在从低功耗模式恢复后设置，并在进入低功耗模式时由硬件清除

1 = 启用输入低功耗模式状态位

0 = 低功耗模式

PLLDIV[5:0] — PLL Clock Divider

此字段设置 PLL 时钟的除法值。默认值为 6'b000001（除以 2）。参考表 9-2 对于其他的除法值。

表 9-2：PLL 时钟分频器

PLLDIV[5:0]	Divider Value
000000	Divide-by-2
000001	Divide-by-2
000010	Divide-by-4
000011	Divide-by-6
000100	Divide-by-8
000101	Divide-by-10
.....	.....
.....	.....
111111	Divide-by-126

提示：系统时钟的频率不应大于 300MHz。

©Levetop Semiconductor Co., Ltd.

Levetop Semiconductor

ADCDIV[3:0] — ADC Clock Divider

此字段设置 ADC 时钟的除法值。默认值为 4'b0000（除以 1）。参考表 9-3。

表 9-3：ADC 时钟分频器

ADCDIV[3:0]	Divider Value
0000	Divide-by-1
0001	Divide-by-2
0010	Divide-by-3
0011	Divide-by-4
0100	Divide-by-5
0101	Divide-by-6
0110	Divide-by-7
0111	Divide-by-8
1000	Divide-by-9
1001	Divide-by-10
1010	Divide-by-11
1011	Divide-by-12
1100	Divide-by-13
1101	Divide-by-14
1110	Divide-by-15
1111	Divide-by-16

LOSCEN — Internal Low Speed 128KHz Oscillator Enable Bit

1 = 内部低速 128 KHz 振荡器已启用
0 = 内部低速 128 KHz 振荡器被禁用

PLLSRCEN — This bit determines whether system clock will be stopped or not when PLLEN is changed from 0 to 1.

1 = 系统时钟将不会停止，系统时钟源来自 FXOSC，直到 PLL 被锁定
0 = 系统时钟将停止，直到 PLL 被锁定

PLLEN — PLL Enabled control bit.

1 = PLL，系统时钟源为 PLL
0 = PLL 已被禁用，且系统时钟源为 FXOSC

SLEEP — Chip Sleep Mode Control Bit

设置休眠位后，芯片将进入 STBYMD 指示的待机模式[1:0]。

该操作与"停止"指令相同。

提示：睡眠仅在 STBYMD[1] = 1'b1 时有效。这与停止指令的情况不同，因为停止指令总是有效的。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

DISPLAYRAMLPEN — 当芯片进入低功耗模式时，显示 RAM（地址范围从 0x0084_0000 到 0x8B_FFFF）低功耗启用位。

显示 RAM 的低功耗模式是通过关闭 RAM 的电源来实现的，RAM 的内容将丢失。

1 = 当芯片进入低功耗模式时，将启用显示闸板的低功耗模式。

0 = 当芯片进入低功耗模式时，禁用显示器的低功耗模式。

CLKOUTSEL — Clock Out Select Bit

表 9-4：CLKOUTSEL 模式

CLKOUTSEL	CLKOUT
0	系统时钟
1	128 KHz 时钟

STBYMD[1:0] — Sleep Operation Control Bits

STBYMD[1:0]在睡眠模式下控制时钟源、系统时钟操作和 LDO 状态，如表 9-5 所示.

表 9-5：睡眠模块中的睡眠操作控制位

STBYMD	ADC Clock	System Clocks	Clock Source	LDO
00	Enable	Disabled	Enable	Normal
01	Disabled	Disabled	Enable	Normal
10	Disabled	Disabled	Disabled	Normal
11	Disabled	Disabled	Disabled	Standby

CACHERAMLPEN — CACHE RAM Low Power Enable Bit.

当芯片进入低功耗模式时，此位启用。CACHE 快取 RAM 的低功耗模式是通过关闭 RAM 的电源来实现的，RAM 的内容将丢失。

1 = 当芯片进入低功耗模式时，将启用快取缓存 RAM 的低功耗模式。

0 = 当芯片进入低功耗模式时，将禁用快取缓存 RAM 的低功耗模式。

ADCEN — Analog-to-digital converter Clock Enable Bit

1 = ADC 时钟已启用

0 = ADC 时钟被禁用

LOSCLPEN — Internal Low Speed 128KHz Oscillator Low Power Enable

如果设置了故障处理，内部低速 128 KHz 振荡器将在待机模式下停止。

1 = 内部低功率模式 128 KHz 振荡器启用

0 = 内部低速 128 KHz 振荡器的低功率模式被禁用

©Levetop Semiconductor Co., Ltd.

9.5.2.2. 低速振荡器控制和状态寄存器（LOSCCSR）

地址：CLOCK_BASEADDR+0x0000_0004

	31	30	29	28	27	26	25	24
读:	LOSCCSTEST[1:0]		0	SIRCST[4:0]				
写:								
复位:	0	0	0	1	1	1	1	1

	23	22	21	20	19	18	17	16
读:	SXOSCST[15:8]							
写:								
复位:	0	1	0	1	0	0	0	0

	15	14	13	12	11	10	9	8
读:	SXOSCST[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	SIRCRDY	SXOSCRDY	WDTCLKCHGDONE	0			WDTCLKSEL	SXOSCEN
写:								
复位:	1	1	1	0	0	0	0	1

= 写入没有影响，访问终止，没有传输错误异常。

图 9-3：低速振荡器控制和状态寄存器（LOSCCSR）

LOSCCSTEST[1:0] — LOSCCSR Write Access Sequence In

不能更改 IOSCCSR 寄存器的可写位，除非写入了正确的序列。正确的序列是：2'b01 → 2'b10 → 2'b11。在通过序列写入这两位之后，这两位的值 == 2'b11，然后可以随意更改 LOSCCSR 寄存器的可写位。当值等于 2'b11 时，写入 2'b00 可以清除这两个位。写入其他值 s 没有效果，并返回 2'b11。

SIRCST[4:0] — Internal Low Speed Oscillator Stable Time Value

内部低速振荡器（SIRC）将等待 32 KHz（SIRC 除以 4）振荡器的 SIRCST[4:0]周期，然后在打开后准备好进行输出时钟。

SXOSCST[15:0] — External Low Speed Oscillator Stable Time Value

外部低速振荡器（SXOSC）将等待 128 KHz（SIRC 源）振荡器的[15:0]周期，然后在接通后准备好进行输出时钟。

©Levetop Semiconductor Co., Ltd.

SIRCRDY — Internal low speed oscillator (SIRC) ready flag.

1 = 内置低速振荡器（SIRC）已准备就绪

0 = 内部低速振荡器（SIRC）未准备好

SXOSCRDY — External low speed oscillator (SXOSC).

1 = 外部低速振荡器（SXOSC）已准备就绪

0 = 外部低速振荡器（SXOSC）未准备就绪

WDTCLKCHGDONE — WDT clock switch done flag.

当将 WDT 时钟源从 SXOSC 更改为 SIRC 时，应在关闭 SXOSC 之前清除 WDTCLKSEL 位。

1 = WDT 时钟开关完成，WDT 可以正常工作

0 = WDT 时钟正在切换，WDT 不能正常工作

WDTCLKSEL — WDT clock selection control bit.

当将 WDT 时钟源从 SXOSC 更改为 SIRC 时，应该在关闭 SXOSC 之前清除 WDT 位。

1 = WDT 时钟源为 SXOSC（32.768 KHz）

0 = WDT 时钟源是 SIRC 128Khz

SXOSCEN — SXOSC Enable Setting

1 = 为 WDT 应用程序启用了 SXOSC。

0 = SXOSC 晶体禁用 WDT 应用。另外，现在应该清除，否则 WDT 时钟将丢失。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

9.5.2.3. PLL 配置和状态寄存器（PLLCSR）

地址：CLOCK_BASEADDR+0x0000_0008

	31	30	29	28	27	26	25	24
读:	PLLCSRTEST[1:0]		0					PLLOCK
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0				PLLN[3:0]			
写:								
复位:	0	0	0	0	0	0	1	0

	15	14	13	12	11	10	9	8
读:	PLLOD[1:0]		PLLM[13:8]					
写:								
复位:	1	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	PLLM[7:0]							
写:								
复位:	0	0	1	1	0	0	1	0

= 写入没有影响，访问终止，没有传输错误异常。

图 9-4：PLL 配置和状态寄存器（PLLCSR）

$$\text{PLL Output Frequency} = XIN \times \frac{M}{N} \times \frac{1}{NO}$$

PLLCSRTEST[1:0] — PLLCSR Write Access Sequence In

不能更改 PLLCSR 寄存器的可写位，除非写入了正确的序列。正确的序列是：2'b01 → 2'b10 → 2'b11。在用这个序列写入这两位后，这两位的值 == 2'b11，然后可以随意改变 PLLCSR 寄存器的可写位。当值等于 2'b11 时，写入 2'b00 可以清除这两个位。写入其他值 s 没有效果，并返回 2'b11。

提示：使用时请保持以下条件：

1. 1MHz <= XIN/N <= 50 MHz
2. 200MHz <= XIN*M/N <= 400MHz
3. M≥4
4. N≥1

PLLOCK — PLL Lock Flag

一旦启用了 PLL，将在等待在 PLL[PLLST]中设置的循环数之后设置 PLLOCK。

1 = PLL 已被锁定

0 = PLL 未被锁定

©Levetop Semiconductor Co., Ltd.

PLLN[3:0] — Input 4-bits divider control bits

此字段设置 PLL 时钟的输入除法值。默认值为 4'b0000。参考表 9-6 对于可用的除法器值。

表 9-6：PLL 输入分频数值

PLLN[3:0]	N
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

PLLOD[1:0] — Output divider control bits

此字段设置 PLL VCO 时钟的输出除法值。默认值为 2'b00。参考表 9-7 对于可用的除法器值。

表 9-7：PLL VCO 输出时钟的除频数值

PLLOD[1:0]	NO
00	Divide-by-1
01	Divide-by-2
10	Divide-by-4
11	Divide-by-8

PLLM[13:0] — Feedback 14-bits divider control bits

该字段设置 PLL 反馈除法值。默认值为 14'b0。参考表 9-8 对于可用的除法器值。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

表 9-8：PLL 反馈的除频器数值

PLLM[13:0]	M
14’b00_0000_0000_0000	0
14’b00_0000_0000_0001	1
14’b00_0000_0000_0010	2
14’b00_0000_0000_0011	3
14’b00_0000_0000_0100	4
14’b00_0000_0000_0101	5
…..	…..
…..	…..
14’b11_1111_1111_1111	16383

9.5.2.4. 模块停止控制寄存器（MSCR）

模块停止控制寄存器（MSCR）始终为读/写状态。

地址：CLOCK_BASEADDR+0x0000_000C

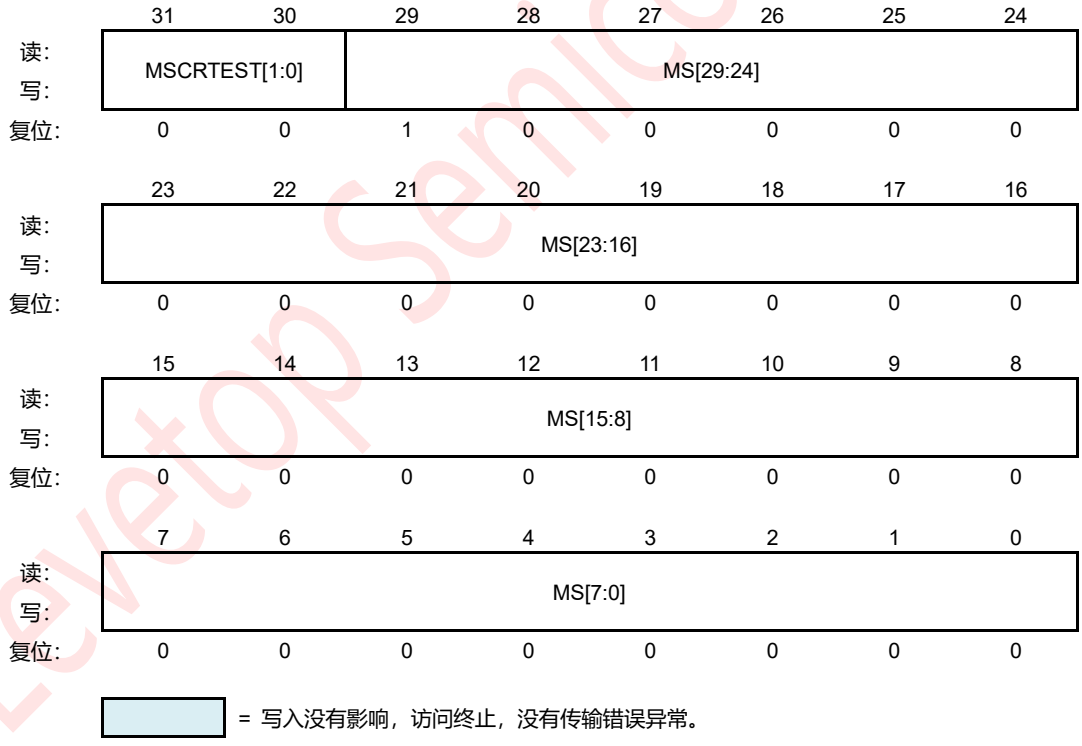


图 9-5：模块停止控制寄存器（MSCR）

MSCRTEST[1:0] — MSCR Write Access Sequence In

不能更改 MSCR 寄存器的可写位，除非写入了正确的序列。正确的序列是：2‘b01 2’b10 2‘b11。在用这个序列写入这两个位后，这两个位的值 == 2‘b11，然后可以随意更改 MSCR 寄存器的可写位。当值等于 2‘b11 时，写入 2’b00 可以清除这两个位。写入其他值 s 没有效果，并返回 2‘b11。

©Levetop Semiconductor Co., Ltd.

MS[29:0] — Module Stop Bits

MS[25:0]位禁用顶层模块的时钟。（参见表 9-9 MS[29:0]位对应模块）。

1 = 模块时钟已被禁用

0 = 模块时钟已被启用

表 9-9：MS[29:0] 相应的模块

MS Bit	Corresponding Module
0	Blender
1	COMP0
2	COMP1
3	ADC
4	PIT0
5	PIT1
6	PIT2
7	PIT3
8	RTC
9	DMA
10	PWM0
11	PWM1
12	EPORT0
13	EPORT1
14	XBAR
15	OPTION
16	RESET
17	WDT
18	SCI0
19	CCM
20	I2C
21	SCI1
22	SCI2
23	CAN
24	EPORT2
25	QSPI0
26	QSPI1
27	QSPI2
28	RGB
29	USBC

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

9.5.2.5. EPT 外部时钟源启用控制寄存器（ECSECR）

地址：CLOCK_BASEADDR+0x0000_0010

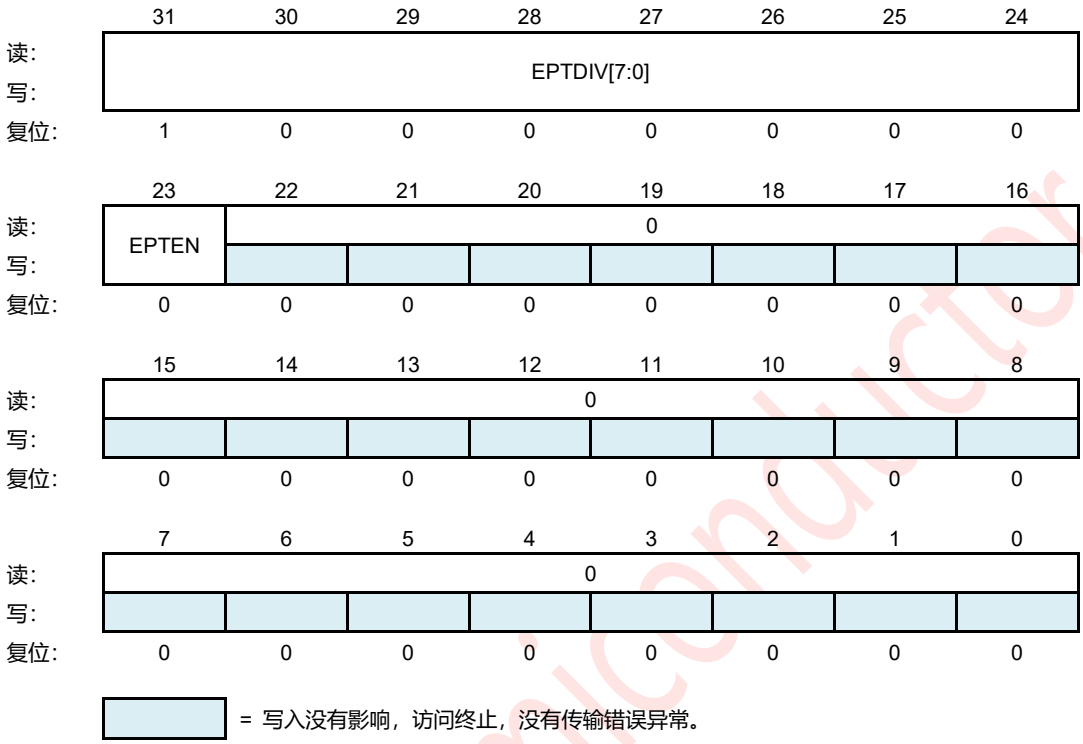


图 9-6：EPT 外部时钟源启用控制寄存器（ECSECR）

EPTDIV[7:0] — EPT Clock Divider

此字段设置 EPT 时钟的除法值。默认值为 8'h80。参考表 9-10 对于其他可用的除法器值。

表 9-10：EPT 时钟分频器

EPTDIV[7:0]	Divider Value
00000000	Divide-by-1
00000001	Divide-by-2
00000010	Divide-by-3
00000011	Divide-by-4
00000100	Divide-by-5
00000101	Divide-by-6
.....	…..
.....	…..
11111110	Divide-by-255
11111111	Divide-by-256

©Levetop Semiconductor Co., Ltd.

EPTEN — EPT Clock Enable Bit

如果设置了 EPTEN 位，EPT 时钟将与系统时钟分开。

1 = EPT 时钟已启用

0 = EPT 时钟被禁用

9.5.2.6. OSC Bist 测试配置寄存器 1（OBTCR1）

地址：CLOCK_BASEADDR+0x0000_0014

	31	30	29	28	27	26	25	24
读: 写:	BIST_HOLD_TARGET[15:8]							
复位:	0	0	0	0	0	0	0	0

	20	19	21	20	19	18	17	16
读: 写:	BIST_HOLD_TARGET[7:0]							
复位:	0	0	0	0	0	0	0	0

	12	11	13	12	11	10	9	8
读: 写:	BIST_TARGET[15:8]							
复位:	0	0	0	0	0	0	0	0

	4	3	5	4	3	2	1	0
读: 写:	BIST_TARGET[7:0]							
复位:	0	0	0	0	0	0	0	0

图 9-7：OSC Bist 测试配置寄存器 1（OBTCR1）

BIST_HOLD_TARGET[15:0] — Bist clk hold count target value

这些位用来设置 clk 等待量，在将微调值更改后的测试为稳定后。

BIST_TARGET[15:0] — Bist clk count target value.

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

9.5.2.7. OSC Bist 测试配置寄存器 2（OBTCR2）

地址：CLOCK_BASEADDR+0x0000_0018

	31	30	29	28	27	26	25	24
读: 写:	BIST_TEST_TARGET[15:8]							
复位:	0	0	0	0	0	0	0	0

	19	22	21	20	19	18	17	16
读: 写:	BIST_TEST_TARGET[7:0]							
复位:	0	0	0	0	0	0	0	0

	11	14	13	12	11	10	9	8
读: 写:	BIST_TEST_CTRIM_MARGIN[15:8]							
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读: 写:	BIST_TEST_CTRIM_MARGIN[7:0]							
复位:	0	0	0	0	0	0	0	0

图 9-8：OSC Bist 测试配置寄存器 2（OBTCR2）

BIST_TEST_TARGET[15:0] — Reserved. Not available to users.

BIST_TEST_CTRIM_MARGIN[15:0] — Reserved. Not available to users.

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

9.5.2.8. OSC Bist 测试控制寄存器（OBTCTLR）

地址：CLOCK_BASEADDR+0x0000_001C

	31	30	29	28	27	26	25	24
读: 写:	BIST_TEST_FTRIM_MARGIN[15:8]							
复位:	0	0	0	0	0	0	0	0

	20	22	21	20	19	18	17	16
读: 写:	BIST_TEST_FTRIM_MARGIN[7:0]							
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	BIST_START	0	BIST_RESETN	BIST_MODE	0	Reserved	BIST_IRC_SEL	BIST_EN
写:								
复位:	0	0	0	1	0	0	1	0

	7	6	5	4	3	2	1	0
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 9-9：OSC Bist 测试控制寄存器（OBTCTLR）

BIST_TEST_FTRIM_MARGIN[15:0] — Reserved. Not available to users.

BIST_START — Bist Start

1 = 启动

0 = 停止

BIST_RESETN — Bist Reset Negate

1 = 否定 Bist 复位

0 = 插入 Bist 复位

BIST_MODE — Bist Mode

1 = 跟踪模式（用于测量 PLL 或 128K 时钟的频率）

0 = 微调模式

提示：该芯片上仅实现了跟踪模式。

BIST_IRC_SEL — PLL or 128KHz clock selection

1 = 选择 PLL 时钟

0 = 选择 128K 时钟

©Levetop Semiconductor Co., Ltd.

BIST_EN — Reference Clock Enable

1 = 启用

0 = 禁用

9.5.2.9. OSC Bist 测试计数器寄存器（OBTCNTR）

地址：CLOCK_BASEADDR+0x0000_0020

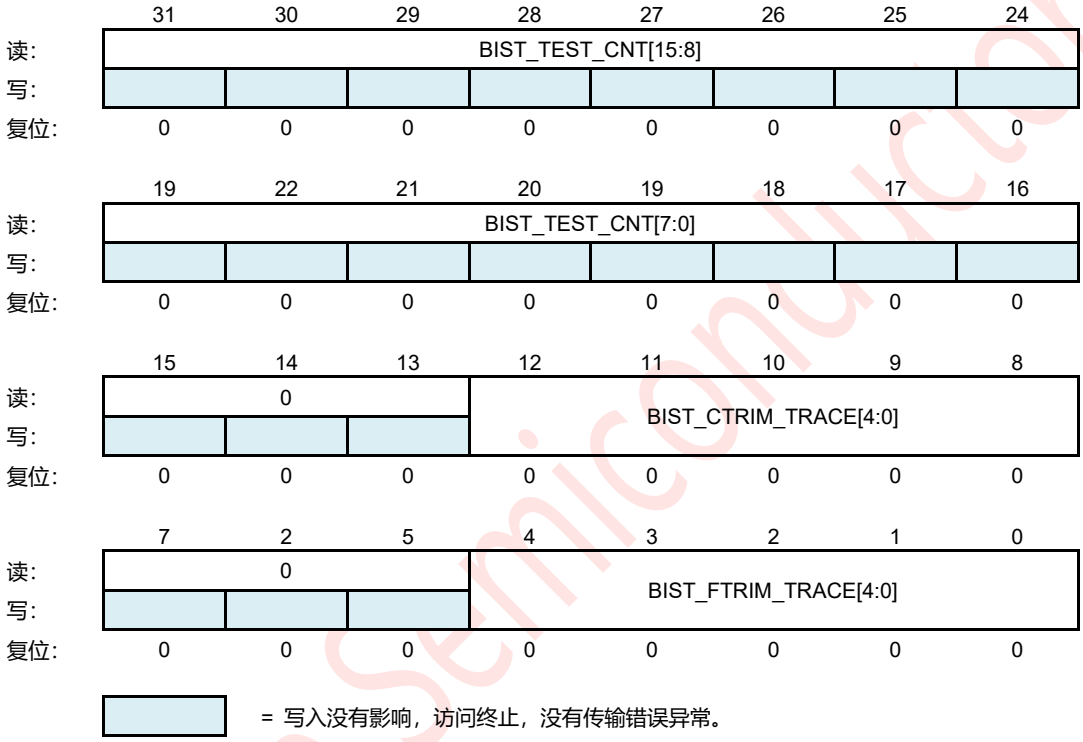


图 9-10：OSC Bist 测试计数器寄存器（OBTCNTR）

BIST_TEST_CNT[15:0] — Bist Test Counter Value

BIST_CTRIM_TRACE[4:0] — Bist Ctrim Trace Value

BIST_FTRIM_TRACE[4:0] — Bist Ftrim Trace Value

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

9.5.2.10. OSC Bist 测试结果寄存器（OBTRR）

地址：CLOCK_BASEADDR+0x0000_0024

	31	30	29	28	27	26	25	24
读:	0						BIST_DONE	BIST_PASS
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							BISTRE-SULT[16]
写:								
复位:	0	0	0	0	0	0	0	1

	15	14	13	12	11	10	9	8
读:	BISTRESULT[15:8]							
写:								
复位:	0	0	0	0	0	1	1	1

	3	6	5	4	3	2	1	0
读:	BISTRESULT[7:0]							
写:								
复位:	0	1	1	1	1	0	1	1

= 写入没有影响，访问终止，没有传输错误异常。

图 9-11：OSC Bist 测试结果寄存器（OBTRR）

BIST_DONE — Bist Done

1 = 完成

0 = 未完成

BIST_PASS — Bist Pass

1 = 通过

0 = 失败

BISTRESULT[16:0] — Bist Trim Result

当 BIST_DONE = 1 和 BIST_PASS = 1 时有效

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

9.6. 功能描述

9.6.1. 打开 PLL

建议采取以下步骤：

1. 首先配置 SYNCR[PLLSRCEN]。
2. 根据目标 VCO 频率配置 PLLCSR[PLLOD]/PLLCSR[PLLN]/PLLCSR[PLLM]。
3. 配置 SYNCR[PLLEN]并打开 PLL。
4. 等待 SYNCR[PLLOCK]状态和系统时钟将被更改为 PLL 时钟。
5. 根据目标系统频率配置 SYNCR[PLLDIV]。

9.6.2. PLL 测量的频率

建议执行以下步骤：

1. 将 BIST_RESETN 位设置为 0 以进行断言复位。
2. 正在设置 BIST_TARGET 计数器的值。
3. 将 BIST_MODE 位设置为 1。
4. 为 PLL 测量，将 IRC_BIST_SEL 位设置为 1。
5. 请将 BIST _START 位设置为 1。
6. 将 BIST_RESETN 位设置为 1。
7. 正在等待 BIST_DONE 位。
8. 读取 BIST_TEST_CNT 来测量 PLL 时钟的频率。
9. 根据 BIST_TEST_CNT 计算 PLL 时钟的频率和 fxosc 时钟的频率。

PLL 时钟的频率计算方法如下：

$$f_{PLL} = f_{fxosc} * BIST_TEST_CNT[15:0] / BIST_TARGET[15:0]$$

9.6.3. 128 KHz 测量的频率

建议执行以下步骤：

1. 将 BIST_RESETN 位设置为 0 以进行断言复位。
2. 正在设置 BIST_TARGET 计数器的值。
3. 将 BIST_MODE 位设置为 1。
4. 为 128 KHz 时钟测量，将 IRC_BIST_SEL 位设置为 0。
5. 请将 BIST _START 位设置为 1。
6. 将 BIST_RESETN 位设置为 1。
7. 正在等待 BIST_DONE 位。
8. 读取 BIST_TEST_CNT，测量 128 KHz 时钟的频率。
9. 根据 BIST_TEST_CNT 计算 128 KHz 时钟的频率和 fxosc 时钟的频率。

128 KHz 的频率计算方法如下：

$$f_{128khz} = f_{fxosc} * BIST_TEST_CNT[15:0] / BIST_TARGET[15:0]$$

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

10.复位控制模块（RCM）

10.1. 基本概述

复位控制模块是用来确定复位的原因，向系统确认适当的复位信号，然后保持复位原因的历史记录。

10.2. 功能特点

模块功能包括：

•复位的五个来源：

-复位电源

-外部复位信号（复位编号）

-软件复位

-看门狗计时器复位

-可编程电压检测复位

•指示上次复位的原因的软件可读状态标志

10.3. 方块图

图 10-1 说明复位控制器

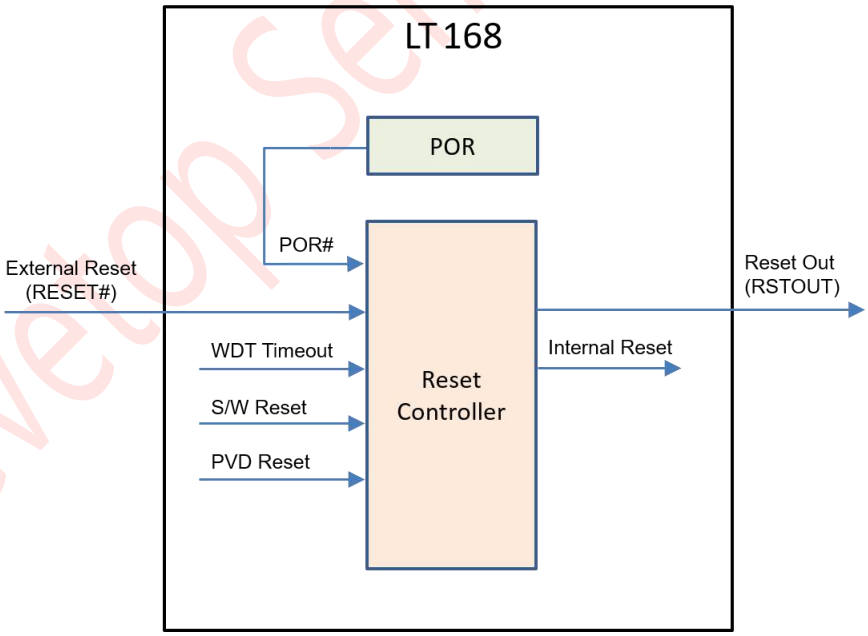


图 10-1：复位控制器的方块图

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

10.4. 内存映射和寄存器

10.4.1. 内存映射

复位控制器编程模型由以下几种寄存器组成：

•复位控制寄存器（RCR）-选择复位控制器功能

•复位状态寄存器（RSR）—反映最后一次复位源的状态

本节描述了复位控制器模块的内存映射和寄存器。复位控制器的基本地址为 0x4002_0000。详见表 10-1 为地址地图和以下段落对寄存器的描述。

表 10-1：复位控制器地址映像

地址	Bits[7:0]	访问权限 [1]
0x0000	保留	S/U
0x0001	RTR—Reset Test Register	S/U
0x0002	RSR—Reset Status Register	S/U
0x0003	RCR—Reset Control Register	S/U

提示： S/U = 主管或用户模式访问。

10.4.2. 寄存器说明

10.4.2.1. 复位测试寄存器（RTR）

复位测试寄存器（RTR）仅用于工厂测试。

地址：R CM_BASEADDR+0x0000_0001

	7	6	5	4	3	2	1	0
读：	TEST[1:0]		0	0	0	0	0	FPOR
写：								
复位：	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 10-2：复位测试寄存器（RTR）

TEST[1:0] — RTR Write Access Sequence In

不能更改 FPOR 寄存器的可写位，除非写入了正确的序列。正确的序列是：2’01 → 2’10 → 2’b11。在用这个序列写入这两个位后，这两个位的值 ==2’b11，然后可以随意更改 FPOR 位的可写位。当值等于 2’b11 时，写入 2’b00 可以清除这两个位。写入其他值 s 没有效果，并返回 2’b11。

©Levetop Semiconductor Co., Ltd.

FPOR — Force Power On Reset

将 0x5B 写入 RTR 寄存器，然后设置这个位将导致系统电源复位。

复位将导致芯片再次修剪。

10.4.2.2. 复位状态寄存器（RSR）

复位状态寄存器（RSR）包含每个复位源的状态位。当输入复位时，复位条件的原因与在复位条件时没有等待处理的其他复位源的值为 0 一起被锁定。这些值随后被反映在 RSR 中。可以同时设置一个或多个状态位。任何后续复位的原因也被记录在寄存器中，覆盖来自前一个复位条件的状态。

RSR 可以随时被读取。写入到 RSR 没有任何效果。

地址：R CM_BASEADDR+0x0000_0002

	7	6	5	4	3	2	1	0
读:	ERST	PVD	SOFT	WDTR	POR	0	0	0
写:								
复位:	0	复位相关						

= 写入没有影响，访问终止，没有传输错误异常。

图 10-3：复位状态寄存器（RSR）

ERST — External Reset

此位表示上次复位状态是由外部复位引起的。

1 = 上次复位状态是由外部复位引起的

0 = 上次复位状态不是由外部复位引起的

PVD — Programmable Voltage Detect

此位表示上次复位状态是由 PVD 复位引起的。

1 = 上次复位状态是由 PVD 复位引起的

0 = 上次复位状态不是由 PVD 复位引起的

SOFT — Software Reset Flag

此位表示上次重位状态是由软件造成的。

1 = 上次复位状态是由软件引起的。

0 = 上次复位状态不是由软件造成的。

WDTR — Watchdog Timer Reset Flag

此位表示上次复位状态是由看门狗计时器超时引起的。

1 = 上次复位状态是由看门狗计时器超时引起的。

0 = 上次复位状态不是由看门狗计时器超时引起的。

©Levetop Semiconductor Co., Ltd.

POR — Power-On Reset Flag

此位表示上次复位状态是由开机复位引起的。

1 = 上次复位状态是由于开机复位所致。

0 = 上次复位状态不是由开机复位引起的。

10.4.2.3. 复位控制寄存器（RCR）

复位控制寄存器（RCR）允许软件控制请求复位。

地址：R CM_BASEADDR+0x0000_0003

	7	6	5	4	3	2	1	0
读:	0	0	0	0	0	0	0	FRCRSTOUT
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 10-4：复位控制寄存器（RCR）

Frcrstout-Force RSTOUT 引脚

输出位允许软件驱动外部 RSTOUT 引脚到低或高状态。

1 = 断 RSTOUT 引脚。写“1”到这个位将驱动 RSTOUT pin 到低。

0 = 否定 RSTOUT 引脚。将“0”写入这个位将驱动 RSTOUT pin 到高。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

10.5. 功能描述

10.5.1. 复位源

表 10-2 定义复位源和由复位控制器驱动的信号。

表 10-2：复位源汇总

复 位 来 源	类 型
POR	异步
ERST	异步
Watchdog timer	异步
Software	同步的
PVD	异步

为了保护数据的完整性，在当前总线周期结束之前，复位控制逻辑不会操作同步复位源。然后，在循环终止后，在系统时钟的下一个上升边缘进行复位。当复位控制逻辑必须同步复位到总线周期结束时，内部总线监视器将自动启用。

异步复位源通常表示发生灾难性故障。因此，复位控制逻辑不等待当前总线周期完成。复位将立即声明到系统。

10.5.1.1. 通电复位（POR）

在通电时，复位控制器断言系统复位。继续进行系统复位，直到 POR 达到可接受的最低水平。

10.5.1.2. 看门狗计时器复位

看门狗计时器超时会导致计时器复位请求被识别并被锁定。

10.5.1.3. 软件复位

如果设置了 32 位 RISC 嵌入式中断控制器（EIC）中的 SYSRESTEQ 位，则将生成软件复位。复位控制器维护系统复位约 2048 个周期。然后，芯片退出复位，恢复操作。

10.5.1.4. 可编程电压检测复位

当设置了嵌入式闪存模块（EFM）中的 CCR 寄存器的 PVDRE 位时，当 VDD 超过 PVD 阈值时，PVD 将产生复位。

10.5.2. 复位控制流程

复位逻辑控制流程如图 10-5 所示.所给出的所有周期计数都是近似的。

©Levetop Semiconductor Co., Ltd.

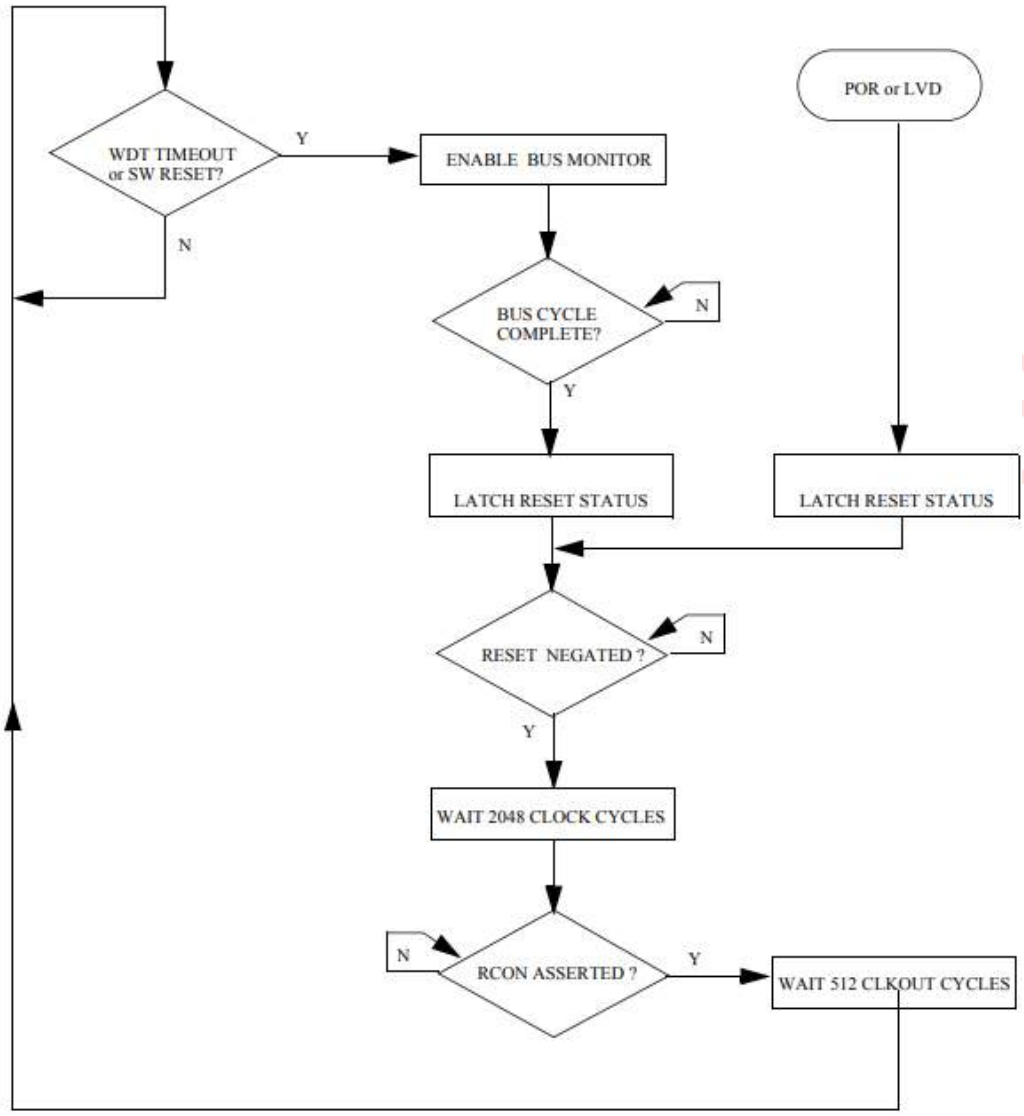


图 10-5：复位控制流程

©Levetop Semiconductor Co., Ltd.

11.静态随机存取存储器（SRAM）

11.1. 功能介绍

LT168 的静态随机存取存储器（SRAM）的功能包括：

- 芯片上的 64K 字节数 SRAM
- 固定地址空间
- 字节、半字（16 位）或 Word（32 位）的读/写访问
- 每次访问一个时钟（包括字节、半字和单词）
- 主管或用户模式访问

11.2. 操作模式

对 SRAM 的访问不受任何限制。该阵列可以在主管模式和用户模式下进行访问。

11.3. 低功耗模式

在等待、睡觉和停止模式下，到 SRAM 的时钟被禁用。当退出这些模式时，不需要任何恢复时间。

11.4. 复位操作

在开机复位后，SRAM 内容立即未定义。SRAM 内容不受系统复位的影响。如果在读取或写访问期间发生同步复位，则访问将正常完成，正在进行的任何管道访问都将停止，而不会损坏 SRAM 内容。

11.5. 中断说明

SRAM 模块不生成中断请求。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

12.高速缓存模块（CACHEM）

12.1. 功能介绍

缓存模块为处理器提供了紧密耦合的处理器-本地内存和到 EBI 和 QSPI0/1/2 内存空间的总线路径。

高速缓存是一个包含地址信息（通常称为标记）和相关数据的高速存储器位置块。其目的是减少内存访问的平均时间。缓存的两个原则：

- 空间局部性—对一个位置的访问可能随后访问相邻位置（例如，顺序指令执行或数据结构的利用）。
- 时间局部性—对内存区域的访问很可能在短时间内重复（例如，执行代码循环）。

为了最小化存储的控制信息的数量，使用空间局部性属性将同一标签下的多个位置分组在一起。此逻辑块通常被称为高速缓存行。

当数据被加载到高速缓存中时，后续加载和存储的访问时间就会减少。这将提高整体性能。对缓存中已经存在的信息的访问称为缓存命中，其他访问称为缓存未命中。

通常，缓存是自我管理的，更新会自动发生。每当处理器想要访问可缓存位置时，都会检查缓存命中。如果访问是缓存命中，则立即发生访问。否则，将分配一个位置，并从内存中加载高速缓存行。可能会有不同的高速缓存拓扑结构和访问策略。

但是，它们必须符合底层体系结构的内存一致性模型。缓存引入了一些潜在的问题，主要原因是：

- 内存访问发生在程序员通常期望的时候；
- 存在可以保存一个数据项的多个物理位置。

缓存控制器的目标是使用任何需要缓存功能的基于 32 位 AHB 总线的应用程序。这个缓存函数可以通过提供对最近使用的代码或数据的快速访问来提高性能。

本地内存控制器支持三种操作模式：

1. 使用此缓存模式对地址空间的通写访问是可缓存的。
   - 输入总线上的通读错误导致包含所需地址的 16 字节对齐的内存地址的输出总线上读取行。此丢失数据被加载到缓存中，并标记为有效且未修改。
   - 对有效缓存位置的穿透读命中从缓存返回数据，而没有输出总线访问。
   - 通写错误绕过缓存并写入输出总线（对通写模式空间无写错误策略分配）。
   - 穿透写入命中更新缓存命中数据并写入输出总线。
2. 使用此缓存模式对地址空间的回写访问是可缓存的。
   - 输入总线上的回写读取错误将导致在包含所需地址的与 16 字节对齐的内存地址的输出总线上读取一条行。此丢失的数据被加载到缓存中，并标记为有效且未修改。
   - 对有效缓存位置的回写读命中将从缓存返回数据，而没有输出总线访问。

©Levetop Semiconductor Co., Ltd.

- 回写错误将执行“读到写”（为回写模式空间分配写错误策略）。执行在包含所需写地址的 16 字节对齐的内存地址的输出总线上读取的行。此丢失数据被加载到缓存中并标记为有效并已修改；然后写入数据将更新适当的缓存数据位置。

3. 不可缓存—对具有此缓存模式的地址空间的访问不可缓存。

- 这些访问会绕过高速缓存并访问输出总线。

缓存控制器有两个 AHB 总线接口、一个主接口和一个从接口。主接口具有解码逻辑，以确定有效地址阶段访问的高速缓存模式。然后，主访问将根据它们的缓存模式访问或绕过缓存。从属接口用于缓存丢失以及绕过缓存的访问。

缓存控制器有一个双向集关联组织。高速缓存有 32 位宽的地址和数据路径，以及 16 字节的行大小。缓存标记和数据存储使用单端口、同步 ram。该控制器具有各种读取和写的数据缓冲区，以提高性能。缓存失败和线推送生成 4 节拍 32 位封装突发访问，首先访问关键字以获得最大的性能。

12.2. 方块图

这些缓存模块提供了对 RAM 和可缓存地址空间的零等待状态访问。

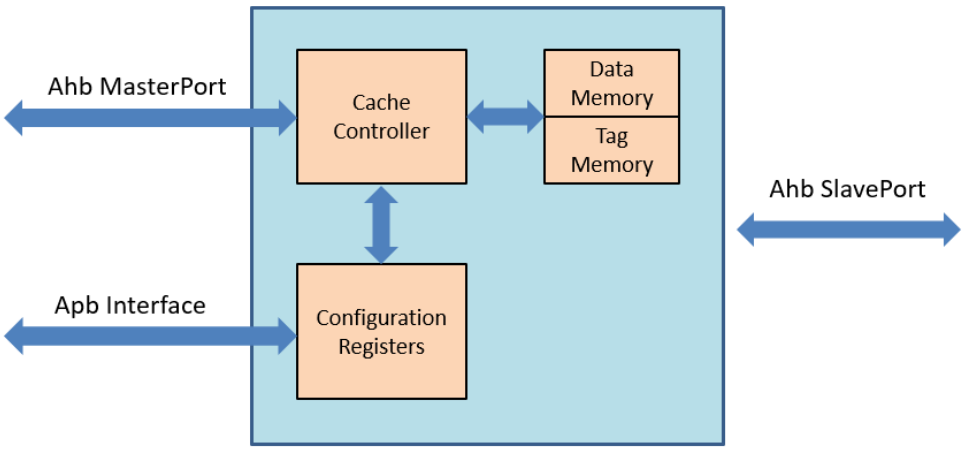


图 12-1：高速缓存模块的方块图

©Levetop Semiconductor Co., Ltd.

12.3. 内存映射和寄存器

12.3.1. 内存映射

本节描述了高速缓存模块的内存映射和寄存器。高速缓存模块的基本地址为 0x4019_0000。表 12-1 显示了这些寄存器的偏移量地址。

表 12-1：高速缓存模块内存映射

地址偏移	Bits[31:0]	访问权限
0x0000	Cache Control Register (LMEM_CCR)	S/U
0x0004	Cache Line Control Register (LMEM_CLCR)	S/U
0x0008	Cache Search Address Register (LMEM_CSAR)	S/U
0x000C	Cache Read/Write Value Register (LMEM_CCVR)	S/U
0x0020	Cache Access Register(LMEM_ACR)	S/U
0x0180	Cache Page Invalidation Start Address	S/U
0x0184	Cache Page Invalidate Size	S/U
0x0188	Cache Clock Gate	S/U

提示：

1. S = CPU 监控模式访问，U = CPU 用户模式访问
2. 在用户模式下访问仅主管地址位置没有影响，并导致周期终止传输错误

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

12.3.2. 寄存器说明

本节提供了对高速缓存模块的描述。

12.3.2.1. 高速缓存控制寄存器（LMEM_CCR）

地址：CACHE_BASEADDR+0x0000_0000

	31	30	29	28	27	26	25	24
读:	GO	0			PUSHW1	INVW1	PUSHW0	INVW0
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0							ENCACHE
写:								
复位:	0	0	1	1	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 12-2：高速缓存控制寄存器（LMEM_CCR）

GO — Initiate Cache Command
设置此位将启动由第 27-24 位指示的高速缓存命令。读取此位表示一个命令是否处于活动状态。这个位一直保持设置，直到命令完成。写零没有任何效果。

1 = 写：由第 27-24 位指示的启动命令。读取：高速缓存命令处于激活状态。

0 = 写：没有效果。读取：没有激活的高速缓存命令。

PUSHW1 — Push way 1

1 = 设置 GO 位时，按所有修改的行 1

0 = 无操作

INVW1 — Invalidate way 1.
如果设置了 PUSHW0 和 INVW0 位，则在设置了 GO 位之后，按方式 1 推送所有修改过的行，并按方式 1 使所有行无效（清除方式 1）。

1 = 在设置 GO 位时，以 1 的方式使所有行无效。

0 = 无操作

©Levetop Semiconductor Co., Ltd.

PUSHW0 — Push way 0

1 = 设置 GO 位时，按 0 推送所有修改的行

0 = 无操作

INVW0 — Invalidate way 0.

如果设置了 PUSHW0 和 INVW0 位，则在设置 GO 位之后，按 0 推送所有修改行，按 0 使所有行无效（清除 0）。

1 = 在设置 GO 位时，将以 0 的方式使所有行无效。

0 = 无操作

ENCACHE — Cache enable.

1 = 缓存已启用

0 = 缓存已禁用

12.3.2.2. 高速缓存线控制寄存器（LMEM_CLCR）

地址：CACHE_BASEADDR+0x0000_0004

	31	30	29	28	27	26	25	24
读:	0				LACC	LADSEL	LCMD[1:0]	
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0	LCR[2:0]			0			TDSEL
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0	WSEL	CACHE_ADDRESS [13:8]					
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	CACHE_ADDRESS [7:2]						0	LGO
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 12-3：高速缓存线控制寄存器（LMEM_CLCR）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

LACC — Line access type.

0 = 读取

1 = 写入

LADSEL — Line Address Select.

当使用缓存地址时，还必须在 CLCR[WSEL]中指定指定的方式。当使用物理地址时，这两种方式都会被搜索，并且只有在命中时才会执行命令。

0 = 缓存地址

1 = 物理地址

LCMD[1:0] — Line command.

00 = 搜索和读取或写。

01 = 无效

10 = 推入

11 = 清除

LCR[2:0] — Line command current line status.

TDSEL — Tag/Data select. Select tag or data for search and read or write commands.

1 = 标签。

0 = 数据。

WSEL — Way select. Select the way of the line commands.

1 = 方式 1。

0 = 方式 0。

CACHE_ADDRESS[13:2] — Cache address.

CLCR[13:4]位用于访问标记数组；CLCR[13:2]位用于访问数据数组。

LGO — Initiate Cache Line Command.

设置此位将启动由 LMEM_CLCR[27-24]指示的高速缓存行命令。读取此位表示行命令是否处于活动状态。这个位一直保持设置，直到命令完成。写零没有任何效果。此位与 CSAR[LGO]共享。

1 = 写入：由第 27-24 位指示的初始化行命令。读取：行命令处于激活状态。

0 = 写：没有效果。读取：没有行命令。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

12.3.2.3. 高速缓存搜索地址寄存器（LMEM_CSAR）

CSAR 寄存器用于为在 CLCR[LADSEL]位中指定的行大小的命令定义显式高速缓存地址或物理地址。

地址：CACHE_BASEADDR+0x0000_0008

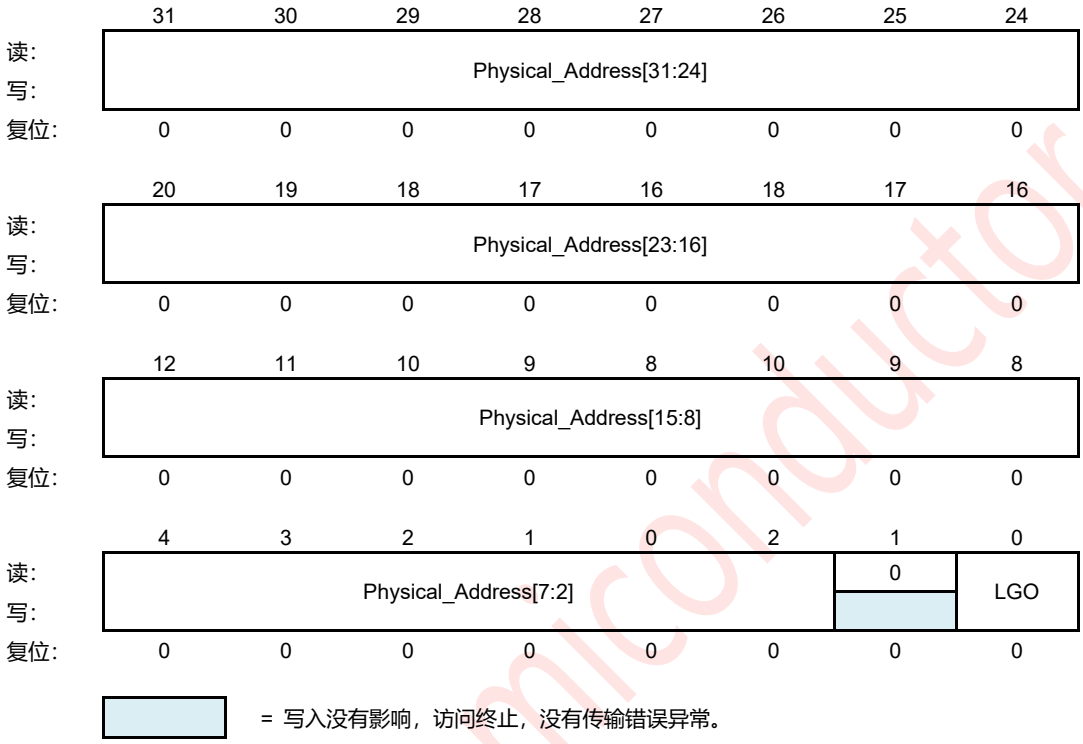


图 12-4：高速缓存搜索地址寄存器（LMEM_CSAR）

Physical_Address[31:2] — Physical Address. This represents bits [31:2] of the system address.
物理地址位用于标记比较

物理地址位用于访问标记数组

物理地址的位用于访问数据数组。

LGO — Initiate Cache Line Command.
设置此位将启动由 LMEM_CLCR[27-24]指示的高速缓存行命令。读取此位表示行命令是否处于活动状态。这个位一直保持设置，直到命令完成。写零没有任何效果。

此位与 CLCR[LGO]共享

- 写：启动由 LMEM_CLCR[27-24]指示的行命令。读取：行命令处于激活状态。
- 写：没有效果。读取：没有行命令。

©Levetop Semiconductor Co., Ltd.

12.3.2.4. 高速缓存读/写值寄存器（LMEM_CCVR）

CCVR 寄存器用于存储在 CLCR 寄存器中指定的命令的写入数据或读取数据。

地址：CACHE_BASEADDR+0x0000_000C

	31	30	29	28	27	26	25	24
读: 写:	DATA[31:24]							
复位:	0	0	0	0	0	0	0	0

	19	22	21	20	19	18	17	16
读: 写:	DATA[23:16]							
复位:	0	0	0	0	0	0	0	0

	11	14	13	12	11	10	9	8
读: 写:	DATA[15:8]							
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读: 写:	DATA[7:0]							
复位:	0	0	0	0	0	0	0	0

图 12-5：高速缓存读/写值寄存器（LMEM_CCVR）

DATA[31:0] — Cache read/write Data

对于标签搜索、读取或写入：

- 数据的位用于标记数组的 R/W 值
- DATA[13:4]位用于读取时的标记集地址；在写入时未使用
- DATA[3:2]位保留

对于数据搜索，请读取。或写下：

- 数据数据位用于数据阵列的收发值

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

12.3.2.5. 高速缓存访问寄存器（LMEM_ACRG）

ACRG 用于控制 10 个不同内存区域的属性，如通写、回写和不可缓存。

地址：CACHE_BASEADDR+0x0000_0020

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	R3_CACHEABLE	R3_WT_WB	R2_CACHEABLE	R2_WT_WB	R1_CACHEABLE	R1_WT_WB	R0_CACHEABLE	R0_WT_WB
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 12-6：高速缓存访问寄存器（LMEM_ACRG）

Region0: 0x2000_0000 ~ 0x2FFF_FFFF
Region1: 0x6000_0000 ~ 0x6FFF_FFFF
Region2: 0x7000_0000 ~ 0x7FFF_FFFF
Region3: 0x8000_0000 ~ 0x8FFF_FFFF

Rx_CACHEABLE — Region x is cacheable.

1 = 可缓存

0 = 不可缓存

Rx_WT_WB — Region x is write-through if cacheable.

1 = 回写

0 = 写入

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

12.3.2.6. 高速缓存页面未验证基本地址寄存器（LMEM_PAGE_INV_BADDR）

地址：CACHE_BASEADDR+0x0000_0180

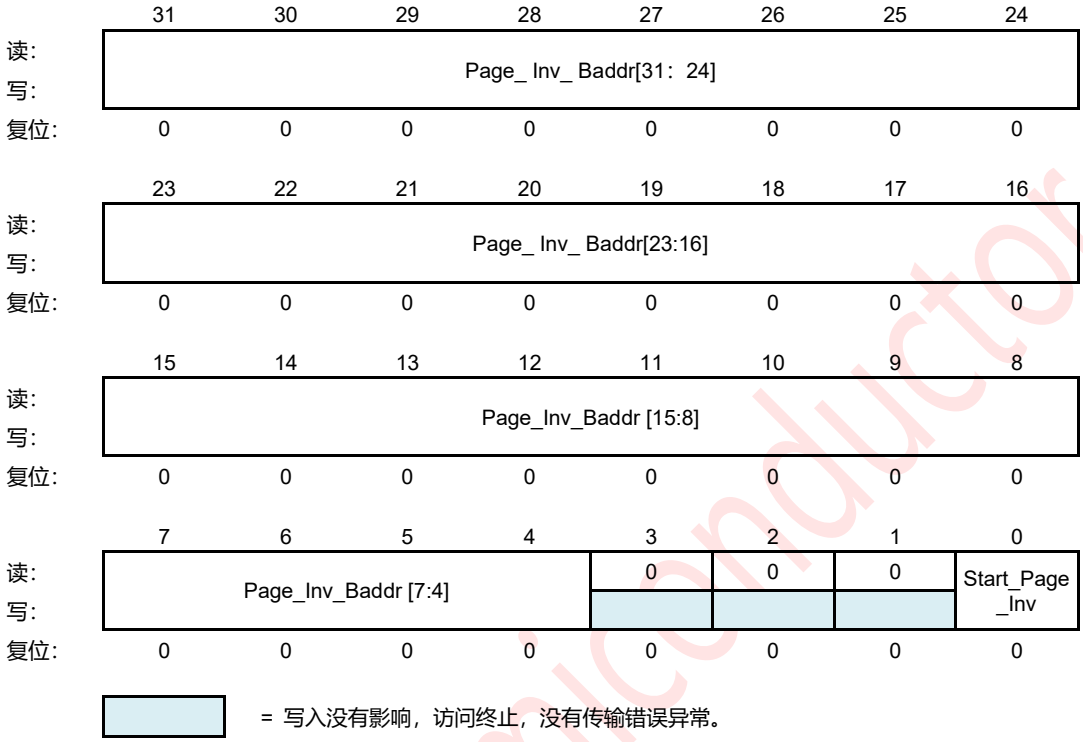


图 12-7：高速缓存页不验证基本地址寄存器（LMEM_PAGE_INV_BADDR）

Page_Inv_Baddr[31:4] — cache invalidate start address for system memory.

由于行对齐，将忽略较低的 4 位。

Start_Page_Inv — start to page invalidate. Cleared if the invalidation is completed.
此位在 LMEM_PAGE_INV_SIZE_REG 的第 0 位中共享。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

12.3.2.7. 高速缓存页面未验证基本大小寄存器（LMEM_PAGE_INV_SIZE）

地址：CACHE_BASEADDR+0x0000_0184

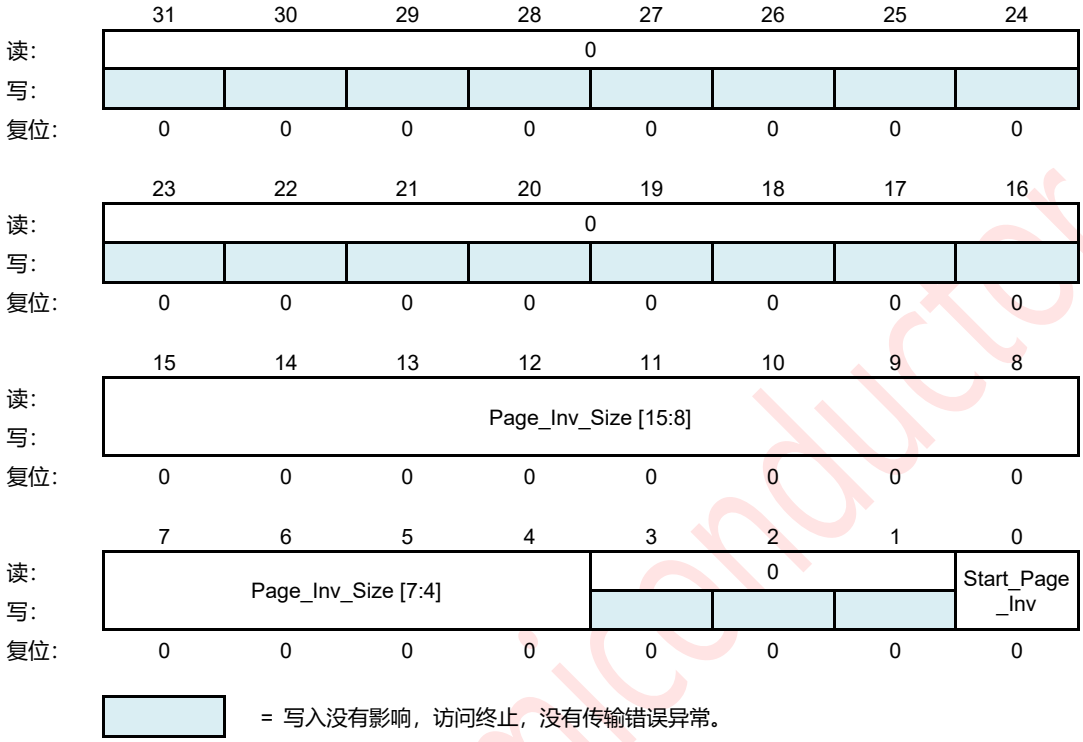


图 12-8：高速缓存页面不验证大小寄存器（LMEM_PAGE_INV_SIZE）

Page_Inv_Size[15:4] — cache invalidate size for system memory.

由于行对齐，将忽略较低的 4 位。

Start_Page_Inv — start to page invalidate. Cleared if the invalidation is completed.

此位在 LMEM_PAGE_INV_BADDR_REG 的第 0 位中共享。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

12.3.2.8. 高速缓存时钟启用寄存器（LMEM_CACHE_CLK_EN）

地址：CACHE_BASEADDR+0x0000_0188

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0							Cache_Clk_Enable
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 12-9：高速缓存时钟启用寄存器（LMEM_CACHE_CLK_EN）

Cache_Clk_Enable — cache clock is enabled. Must be ’1’ if cache is enabled.

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

12.4. 缓存功能

该设备上的缓存的结构如下。这两个缓存都有一个双向集关联缓存结构，总大小为 32K 字节。这些缓存有 32 位的地址和数据路径，以及 16 字节的行大小。缓存标记和数据存储使用外部单端口、同步 ram。

对于这些 32K 字节的缓存，每个缓存 TAG 使用两个 1024x20 位 RAM 数组，而缓存 DATA 使用两个 4096x32 位 RAM 数组。缓存 TAG 条目存储 18 位上地址以及每个缓存行修改的有效位。缓存数据条目存储四个字节的代码或数据。

所有正常的高速缓存访问都将使用物理地址。这将导致以下高速缓存地址的使用：

CACHE - 32K Bytes size = (1024 sets) x (16-byte lines) x (2-way set-associative)

TAG:

- Address[31:14] used in tag for compare (hit) logic
- Address[13:4] used to select 1 of 1024 sets
- Address[3:0] not used

DATA:

- Address[31:14] not used
- Address[13:4] used to select one of 1024 sets
- Address[3:2] used to select one of four 32-bits words within a set
- Aaddress[1:0] used to select the byte within the 32-bits word

缓存使用一个周期的并行 TAG 和数据访问结构。该结构在两阶段 AHB 总线管道中从地址阶段的负边排列到数据阶段的负边：

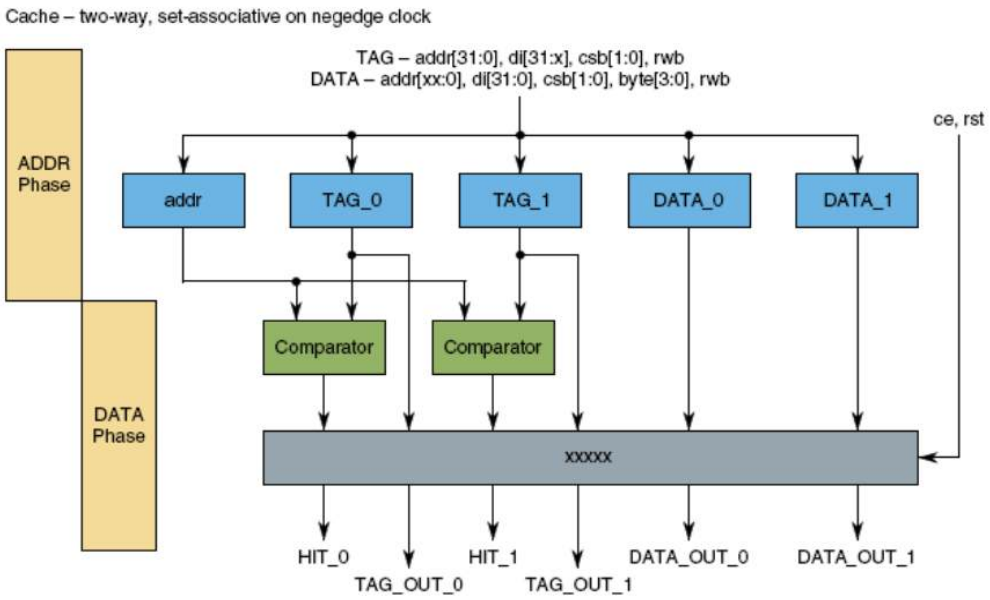


图 12-10：高速缓存标签和数据访问结构

©Levetop Semiconductor Co., Ltd.

在正常的可缓存操作中，完整地址寄存器在缓存控制逻辑中的地址阶段的负边缘上，而地址位[12:4]在 TAG 数组中寄存器，地址位[12:2]在 DATA 数组中寄存器。在从地址阶段的负边缘到数据阶段的负边缘的整个周期中，将访问 TAG 和 data 阵列并分析缓存命中。对于命中值，在数据阶段的后半段返回所需的读取数据，而写入操作则从数据阶段周期的负边缘开始完成。对于失败，数据阶段根据需要保持在缓存阶段，同时使用其从总线（CCM、CSM）通过发送到交叉条开关的行大小的数据传输访问所需的缓存线。

12.5. Cache 缓存控制

代码和系统缓存在复位时被禁用。复位时未清除缓存标记和数据数组。因此，要启用缓存，必须执行缓存命令来清除和初始化所需的标记数组位，并配置和启用缓存。

12.5.1. 缓存集命令

缓存集命令可以用于以下操作：

- 所有方式 0，
- 所有的方式 1，或
- 所有两种方式（完全缓存）。

缓存集命令将使用CCR寄存器中的上位来启动。缓存集命令在缓存上执行CCR的缓存启用位CCR[ENCACHE]。

通过设置 CCR[GO]位来启动高速缓存集命令。此位也充当设置命令的繁忙位。当命令被激活时，它保持设置，并在设置命令完成时由硬件清除。

表 12-2 给出了支持的缓存集命令。设置命令的工作方式如下：

- 无效–无条件清除有效并修改缓存条目的位。
- 按–如果缓存条目有效并已修改 Push a cache entry，然后清除修改位。如果条目无效或未修改，请保持原样。
- 清除–推送缓存条目是否有效，并已修改，然后清除有效和修改的位。如果该项无效或未被修改，请清除该有效位。

表 12-2：缓存集命令

CCR【27:24】				命 令
PUSH W1	INVW1	PUSH W0	INVW0	
0	0	0	0	NOP
0	0	0	1	Invalidate all way 0
0	0	1	0	Push all way 0
0	0	1	1	Clear all way 0
0	1	0	0	Invalidate all way 0
0	1	0	1	Invalidate all way 1; Invalidate all way 0 (invalidate cache)
0	1	1	0	Invalidate all way 1; push all way 0
0	1	1	1	Invalidate all way 1; clear all way 0

©Levetop Semiconductor Co., Ltd.

CCR【27:24】				命 令
PUSH W1	INVW1	PUSH W0	INVW0	
1	0	0	0	Push all way 1
1	0	0	1	Push all way 1; Invalidate all way 0
1	0	1	0	Push all way 1; Push all way 0 (push cache)
1	0	1	1	Push all way 1; clear all way 0
1	1	0	0	Clear all way 0
1	1	0	1	Clear all way 1; Invalidate all way 0
1	1	1	0	Clear all way 1; push all way 0
1	1	1	1	Clear all way 1; clear all way 0 (clear cache)

复位后，在使用缓存之前，先完成一个无效的缓存命令。

12.5.2. Cache Line Commands 缓存行命令

缓存行命令一次对缓存中的一行进行操作。高速缓存行命令可以使用物理地址或高速缓存地址来执行。

- 缓存地址由集合地址和方式选择组成。该行命令作用于指定的高速缓存行。
- 带有物理地址的缓存行命令首先搜索由物理地址的位[13:4]指定的缓存集的两种方式。如果它们命中，命令就会以命中的方式执行操作。

高速缓存行命令将使用 CLCR 寄存器中的上一位来指定。缓存行命令对独立于缓存启用位（CCR[enc 缓存]）执行操作。当使用高速缓存地址时，该命令可以完全由 CLCR 寄存器指定。当使用物理地址时，该命令还必须使用 CSAR 寄存器来指定物理地址。

通过设置行命令 go 位（CLCR[LGO]或 CSAR[LGO]）来启动行缓存命令。这个位也充当行命令的繁忙位。当命令处于活动时保持设置，并在命令完成时被硬件清除。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

CLCR[27:24]位选择以下行命令：

表 12-3：缓存行命令

Clcr[27:24]			命令
LACC	LADSEL	LCMD	
0	0	00	Search by cache address and way
0	0	01	Invalidate by cache address and way
0	0	10	Push by cache address and way
0	0	11	Clear by cache address and way
0	1	00	Search by physical address
0	1	01	Invalidate by physical address
0	1	10	Push by physical address
0	1	11	Clear by physical address
1	0	00	Write by cache address and way
1	0	01	Reserved,NOP
1	0	10	Reserved,NOP
1	0	11	Reserved,NOP
1	1	xx	Reserved,NOP

12.5.2.1. 使用缓存地址执行一系列行命令

只需写入 CLCR，就可以执行一系列具有增量缓存地址的行命令。

- 将该命令放在 CLCR[27:24]中，
- 根据需要设置方式（CLCR[WSEL]）和标签/数据（CLCR[TDSEL]）控件，
- 将缓存地址放在 CLCR[缓存头数]中，和
- 设置行命令 go 位（CLCR[LGO]）。

- 当一个行命令完成后，通过以下步骤启动下一个命令：
  - 增加高速缓存地址（在第 2 位逐步通过数据或在第 4 位逐步通过行），以及
  - 设置行命令 go 位（CLCR[LGO]）。

12.5.2.2. 使用物理地址执行一系列行命令

通过以下步骤执行一系列具有增量物理地址的行命令：

- 将该命令放置在 CLCR 中[27:24]
- 设置标记/数据（CLCR[TDSEL]）控件
- 将物理地址放在 CSAR[PHYADDR]中，并设置行命令去位（CSAR[LGO]）。

©Levetop Semiconductor Co., Ltd.

当一个行命令完成时，通过以下步骤启动下一个命令：

- 增加物理地址（在第 2 位步进数据或在第 4 位步进通过行），以及
- 设置行命令 go 位（CSAR[LGO]）。

行命令 go 位在 CLCR 和 CSAR 寄存器之间共享，因此可以在对 CSAR 寄存器的一次写入中完成上述步骤。

12.5.2.3. 行命令结果

在完成行命令时，CLCR 寄存器包含有关该命令目标的行的初始状态的信息。对于具有缓存地址的行命令，在从目标缓存行执行行命令操作之前读取此信息。对于具有物理地址的行命令，在从命中缓存行执行行命令操作之前读取此信息，或者如果命令失败，则清除初始有效位。通常，如果清除了有效的指示符 CLCR[LCIVB]，则在行命令开始时目标行无效，并且没有执行行操作。

表 12-4：行命令结果

CLCR[22:20]			用于缓存地址命令	用于物理地址命令
LCWY	LCIMB	LCIVB		
0	0	0	Way 0 line Invalid	No hit
0	0	1	Way 0 valid, not modified	Way 0 valid, not modified
0	1	0	Way 0 line was Invalid	No hit
0	1	1	Way 0 valid and modified	Way 0 valid and modified
1	0	0	Way 1 line was Invalid	No hit
1	0	1	Way 1 valid not modified	Way 1 valid not modified
1	1	0	Way 1 line was Invalid	No hit
1	1	1	Way 1 valid and modified	Way 1 valid and modified
4’b1xxx			Bit 23 is available for future use, giving 8 more options if necessary.	

在完成除写入以外的行命令时，CCVR（缓存 R/W 值寄存器）包含有关行标记或命令目标数据的初始状态的信息。对于行命令，CLCR[TDSEL]会在标记和数据之间进行选择。如果行命令使用了物理地址，数据就不在乎。对于写命令，CCVR 保存写数据。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

13.横杆开关（XBAR）

13.1. 功能介绍

13.1.1. 基本概述

本章提供了关于横杆开关（Crossbar Switch）的布局、配置和编程的信息。交叉开关使用硬件互连矩阵连接总线主线和总线从线。这种结构允许所有总线主线同时访问不同的总线从线而不受干扰，同时当总线主线访问同一个从线时在它们之间提供仲裁。各种总线仲裁方法和属性可以在从从的基础上编程。

XBAR 有 3 个主端口和 8 个从端口。图 13-1 显示了 XBAR 的方块图。

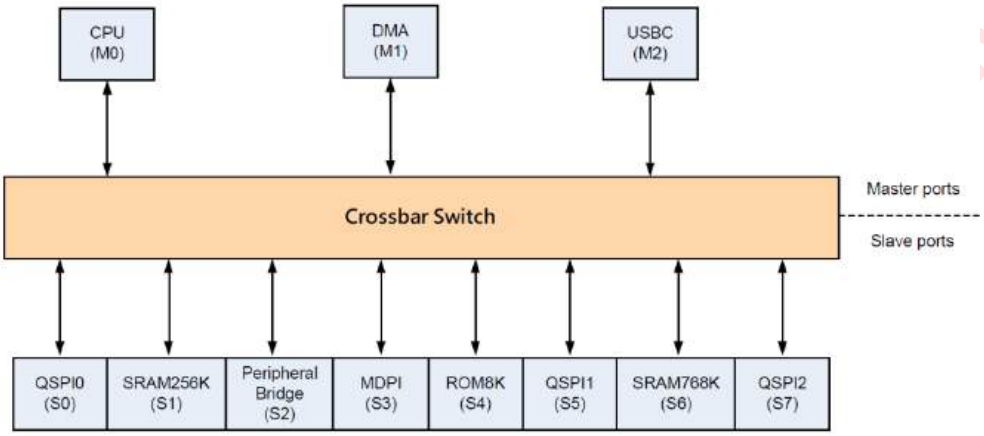


图 13-1：XBAR 设备特定的方块图

13.1.2. 功能特点

横杆开关包括以下显着特征：

- 3 个 Master 主端口：
  - CPU
  - DMA
  - USBC
- 8 个从端口
  - ROM 存储器控制器
  - 两个内部 SRAM memory 控制器
  - 三个 QSPI 接口
  - EBI 接口
  - Peripheral bridges
- 对称横条总线开关的实现
  - 从不同 Master 到不同 Slave 的并发访问
  - 基于逐从操作的可配置的从属仲裁属性

©Levetop Semiconductor Co., Ltd.

- 32 位地址，32 位数据路径
- 固定优先级方案和固定 Parking Strategy
- 支持 8 位、16 位和 32 位的单次传输
- 支持低功耗 Park 式

13.2. 内存映射和寄存器

本节提供了其碱基地址为 0x4017_0000 的 XBAR 寄存器的信息。

13.2.1. 内存映射

横条开关的每个从属端口都包含配置寄存器。配置寄存器的读写传输需要两个总线时钟周期。这些寄存器只能在监控模式下进行读取和写入。此外，这些寄存器只能通过 32 位的访问来读取或写入。

如果在交叉开关中访问未实现的位置，则返回总线错误响应。

从寄存器还具有一点功能，当设置时，可以防止寄存器被写入。寄存器仍然可读，但是未来的写入尝试对寄存器没有影响，并且以对启动写入的主服务器的总线错误响应终止。

提示：本节显示了所有八个 Master 和 Slave 的寄存器。如果在此特定设备上没有使用主或从设备，则在写入其寄存器时会出现意外的结果。请参见设备的芯片配置详细信息。此外，所有对交换交换机寄存器的引用都是基于物理端口连接，而不是逻辑端口号。

XBAR 寄存器的内存映射如表 13-1 所示。

表 13-1：XBAR 寄存器配置摘要

地址偏移	Bits[31:0]	访问权限(1)
0x0000	Master Priority Register for Slave Port 0(MPR0)	S/U
0x0010	General Purpose Control Register for Slave Port 0(SGPCR0)	S/U
0x0100	Master Priority Register for Slave Port 1(MPR1)	S/U
0x0110	General Purpose Control Register for Slave Port 1(SGPCR1)	S/U
0x0200	Master Priority Register for Slave Port 2(MPR2)	S/U
0x0210	General Purpose Control Register for Slave Port 2(SGPCR2)	S/U
0x0300	Master Priority Register for Slave Port 3(MPR3)	S/U
0x0310	General Purpose Control Register for Slave Port 3(SGPCR3)	S/U
0x0400	Master Priority Register for Slave Port 4(MPR4)	S/U
0x0410	General Purpose Control Register for Slave Port 4(SGPCR4)	S/U
0x0500	Master Priority Register for Slave Port 5(MPR5)	S/U
0x0510	General Purpose Control Register for Slave Port 5(SGPCR5)	S/U
0x0600	Master Priority Register for Slave Port 6(MPR6)	S/U
0x0610	General Purpose Control Register for Slave Port 6(SGPCR6)	S/U

©Levetop Semiconductor Co., Ltd.

地址偏移	Bits[31:0]	访问权限(1)
0x0700	Master Priority Register for Slave Port 7(MPR7)	S/U
0x0710	General Purpose Control Register for Slave Port 7(SGPCR7)	S/U

提示(1)：仅限 S = CPU 监控模式访问。S/U = CPU 监控器或用户模式访问。访问用户模式下的地址没有影响，导致周期终止传输错误。

13.2.2. 寄存器描述

13.2.2.1. 主优先级寄存器（XBAR_MPRn）

主优先级寄存器（MPR）位于每个从端口中，并根据每个从端口设置每个主端口的优先级，例如，MPR0 为从端口 0 的每个主端口设置优先级。

主优先级寄存器只能在具有 32 位访问的监督模式下进行访问。一旦在从属通用控制寄存器中设置了 RO（只读）位，主优先级寄存器只能被读取，尝试写入它将对 MPR 没有影响，并导致错误响应。

MPRn：偏移地址:0x0000 + n*0x100

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0					MSTR2[2:0]		
写:								
复位:	0	0	0	0	0	0	1	0

	7	6	5	4	3	2	1	0
读:	0	MSTR1[2:0]			0	MSTR0[2:0]		
写:								
复位:	0	0	0	1	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 13-2：主优先级寄存器（XBAR_MPRn）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

MSTR2[2:0] — Master 2 Priority

这些位为关联的从端口上的主端口 2（USBC）设置仲裁优先级。

这些位是通过硬件复位来初始化的。复位值为 010。

000 = 在访问从属端口时，此主服务器具有 1 级（最高）优先级。

001 = 此主服务器在访问从属端口时具有第 2 级优先级

.....

.....

.....

111 = 在访问从属端口时，此主服务器具有 3 级（最低）优先级。

MSTR1[2:0] — Master 1 Priority

这些位为关联的从端口上的主端口 1（DMA）设置仲裁优先级。

这些位是通过硬件复位来初始化的。复位值为 001。

000 = 在访问从属端口时，此主服务器具有 1 级（最高）优先级。

001 = 此主服务器在访问从属端口时具有第 2 级优先级

.....

.....

.....

111 = 在访问从属端口时，此主服务器具有 3 级（最低）优先级。

MSTR0[2:0] — Master 0 Priority

这些位为相关的从端口上的主端口0（CPU）设置仲裁优先级。

这些位是通过硬件复位来初始化的。复位值为000。

000 = 在访问从属端口时，此主服务器具有1级（最高）优先级。

001 = 此主服务器在访问从属端口时具有第2级优先级

.....

.....

.....

111 = 在访问从属端口时，此主服务器具有 3 级（最低）优先级。

提示： 不能使用相同的优先级来编程两个可用的主端口。尝试对两个或多个具有相同优先级级别的可用主程序进行编程将会导致错误响应，并且 MPR 将不会被更新。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

13.2.2.2. 从属通用控制寄存器（XBAR_SGPCRn）

从属通用控制寄存器（SGPCR）控制每个从属端口的多个功能。只读（RO）位将防止与此从端口关联的任何寄存器被写入一次设置。这个位可以像用户希望的那样用“0”写入多次，但是一旦它被写入“1”，只有一个复位条件才允许再次写入它。

PCTL 位决定了当没有主端口主动发出请求时，从端口将如何停靠。可用的选项是停在主定义的公园，公园在最后一个主使用从端口，或进入一个低功率停车模式将迫使从端口的输出非活动状态时没有主请求访问。如果从端口不饱和，低功率停车功能可以导致整体功率节约；然而，当它不使用时，当任何主机试图访问它时，它将迫使一个额外的时钟延迟，因为它不会停在任何主机上。

公园位决定当没有主发出主动请求时，从 Slave 将停在哪个主站上。请谨慎使用，只选择在设计中实际存在的主端口。如果用户将 PARK 位放到当前设计实现中不存在的主程序，则将导致未定义的行为。

提示：SGPCR 只能在具有 32 位访问的监控模式下访问。一旦设置了 XBAR_SGPCR[RO] ，就只能读取 SGPCR；尝试写入它将不会对 SGPCR 产生影响，并导致错误响应。

SGPCRn：偏移地址: 0x0010 + n*0x100

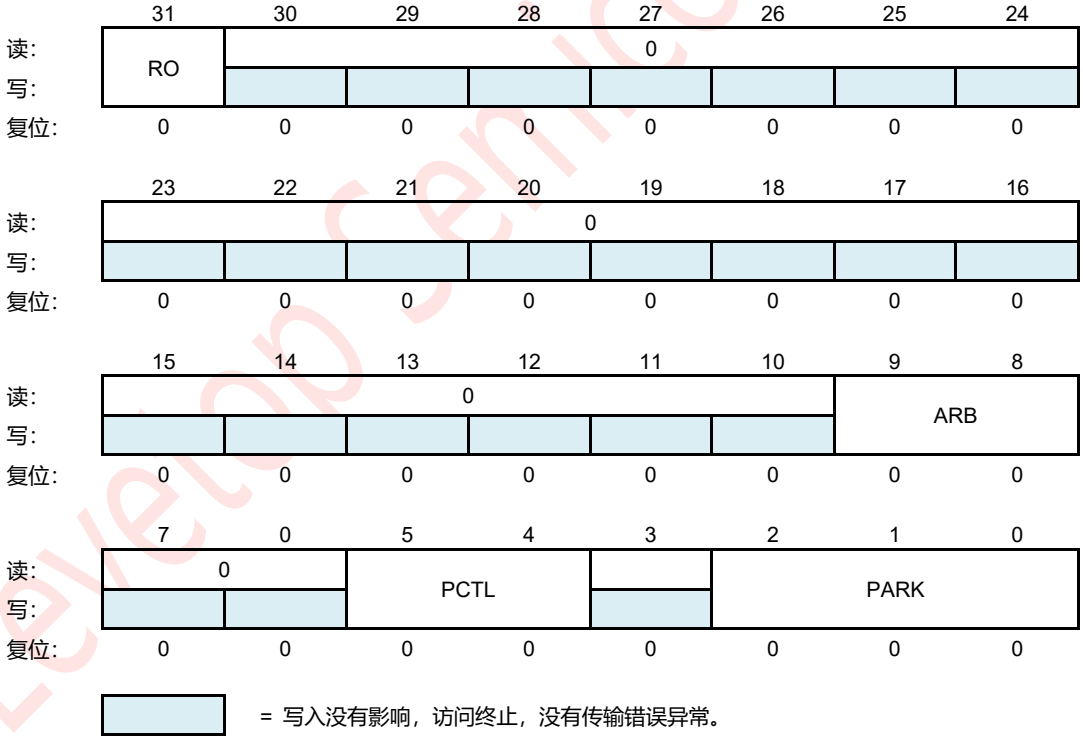


图 13-3：从属通用控制寄存器（XBAR_SGPCRn）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

RO — Read Only

此位用于强制一个从端口的所有寄存器为只读。一旦写入“1”，它只能通过硬件复位来清除。此位通过硬件复位进行初始化。复位值为0。

0 = 可以写入所有这些从属端口的寄存器。

1 = 所有这个从端口的寄存器都是只读的，不能写入（尝试的写入没有效果，并导致错误响应）。

ARB[1:0] — Arbitration Mode

这些位用于为从端口的仲裁策略。

这些位是通过硬件复位来初始化的。复位值为 00。

00 = 固定优先级

01 = 循环（旋转）优先级

10 = 保留

11 = 保留

PCTL[1:0] — Parking Control

确定从属端口的停车控制功能。如果从端口不饱和，低功耗停车功能可以节省整体电力；但是，当任何主端口在不使用时试图访问从端口时，这将强制产生一个额外的延迟时钟，因为它没有停在任何主端口上。

这些位是通过硬件复位来初始化的。复位值为00。

00 = 当没有主端口发出请求时，仲裁者将把从端口存放在由park位字段定义的主端口上。

01 = 当没有主端口发出请求时，仲裁者将把从端口存放在要控制从端口的最后一个主端口上。

10 = 当没有主服务器发出请求时，仲裁者将把从属端口停在没有主服务器上，并将所有输出驱动到一个恒定的安全状态。

11 = 保留

PARK[1:0]

这些位用于确定在没有主端口主动发出请求时，该从端口位于哪个主端口上。

这些位是通过硬件复位来初始化的。复位值为000。

提示：仅选择设备上实际存在的主端口。如果没有，则可能会发生未定义的行为。

000 = Park on Master Port 0 (CPU)
001 = Park on Master Port 1 (DMA)
010 = Park on Master Port 2 (USBC)
011 = Reserved
100 = Reserved
101 = Reserved
110 = Reserved
111 = Reserved

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

13.3. 功能说明

13.3.1. 一般操作

当主服务器发送对横杆开关的访问时，将立即进行访问。如果访问的目标从端口可用，则该访问将立即显示在从端口上。可以通过交叉条进行单时钟（零等待状态）访问。如果访问的目标从端口繁忙或停在其他主端口上，请求主只需看到插入的等待状态，直到目标从端口能够服务主请求。服务请求的延迟取决于每个主服务器的优先级级别和响应从服务器的访问时间。

由于横条开关似乎只是主设备的另一个从端口，所以主设备不知道它是否真的拥有它所瞄准的从端口。虽然主服务器不能控制它所瞄准的从属端口，但它只是进入了一个等待状态。

只有在之前对不同从端口的访问完成之后，主端口才可以控制目标从端口，而不管主端口在新目标从端口上的优先级如何。这可以防止在以下情况下发生死锁：

- 优先级的主机具有：
- 对一个响应时间较长的从属端口的未完成请求
- 对不同从端口的等待处理访问，以及
- 低优先级主服务器还向与高优先级主服务器的等待访问相同的从端口发出请求。

在主端口控制了它所瞄准的从端口之后，主端口仍然控制该从端口，直到它通过运行一个空闲周期或离开该从端口进行下一次访问而放弃从端口。如果另一个优先级较高的主服务器向从端口发出请求，主服务器也可能失去对从端口的控制；但是，如果主服务器正在运行固定长度的突发传输，它将保留对从端口的控制，直到该传输完成。当主服务器控制给定的从服务器端口时，交叉栏将来自从服务器的所有响应信息返回给请求主服务器。

横条终止所有的主 IDLE 传输（而不是允许终止来自其中一个从总线）。此外，当没有主请求访问从端口时，交叉条驱动器 IDLE 传输到从总线上，即使默认主服务器可以被授予对从端口的访问权。

当从总线被横条分开时，它可以将从端口停在由 XBAR x_CRS n[PARK]指示的主端口上。这样做是为了保存仲裁延迟的初始时钟，否则如果主仲裁必须获得对从端口的控制，就会看到。从端口也可以进入低功耗停车模式，尝试使用 XBAR x_CRS[PCTL]节省电。

13.3.2. 寄存器相干性

由于寄存器的内容对横杆的操作有实时影响，因此重要的是要了解，任何寄存器修改在寄存器写入后立即生效。寄存器的值不使用与从端口相关的主访问进行跟踪；相反，它们只使用从访问进行跟踪。

13.3.3. XBAR 仲裁

XBAR 支持两种仲裁方案：简单的固定优先级比较算法和简单的循环公平算法。该仲裁方案对每个从属端口都是独立可编程的。

©Levetop Semiconductor Co., Ltd.

13.3.3.1. 固定优先级操作

在固定优先级模式下操作时，每个主服务器在 XBAR x_PRSn（优先级寄存器）中分配一个唯一的优先级级别。如果两个主服务器请求访问从端口，则在所选优先级寄存器中具有较高优先级的主服务器将获得对从端口的控制。

当主端口向从端口发出请求时，从端口将检查新的请求主端口的优先级是否高于当前控制从端口的主端口的优先级（除非从端口处于停放状态）。从端口在每个时钟边缘执行仲裁检查，以确保适当的主端口（如果有）能够控制从端口。

如果新请求主的优先级高于当前控制从端口的主的优先级，则新请求主将被授予在下一个时钟边缘对从端口的控制权。此规则的例外情况是，当前控制从端口的主服务器正在运行固定长度的突发传输或锁定的传输。在这种情况下，新的请求主服务器必须等待突发传输或锁定传输结束后，才被授予它对从端口的控制权。

如果新请求主的优先级低于当前控制从端口的主，则新请求主将被迫等待当前主运行以下循环之一：

- IDLE 周期
- 到当前从属端口以外的位置的非空闲循环

13.3.3.2. 循环优先操作

在循环模式下操作时，每个主端口根据物理主端口号分配一个相对优先级。将此相对优先级与在从总线上执行传输的最后一个主机的 ID（主端口号）进行比较。最高优先级请求主节点成为下一个传输边界的从总线的所有者（考虑锁定和固定长度的突发传输）。优先级基于请求主的 ID 比最后一个主的 ID 早多远。

在对从端口授予访问权限之后，主服务器可以向该端口执行尽可能多的传输，直到另一个主服务器向同一从端口发出请求为止。下一个主线被允许访问下一个传输边界中的从端口，如果当前主线没有等待处理的访问请求，则可能在下一个时钟周期中访问。

作为循环模式下仲裁的一个示例，假设交叉条使用主端口 0、1、4 和 5 实现。如果从端口的最后一个主端口是主端口 1，而主端口 0、4 和 5 同时发出请求，则它们将按 4、5 和 0 的顺序提供服务。

停车可以继续在循环模式下使用，但它不会影响循环指针，除非停车的主控件实际执行转移。在一个周期的仲裁后，将切换到下一个主线。如果从端口进入低功耗停车模式，循环指针将被复位为指向主端口 0，从而给予其最高优先级。

13.3.4. 优先级分配

需要为每个主端口分配一个唯一的 3 位优先级级别。如果尝试在一个寄存器（MPR）中对具有相同优先级的多个主端口进行编程，XBAR 将会出现错误，寄存器将不会更新。

13.4. 初始化/应用程序信息

无需或对交叉杆开关进行初始化。硬件复位确保交叉开关使用的所有寄存器位正确初始化为有效状态。设置和优先级应被编程，以实现最大的系统性能。

©Levetop Semiconductor Co., Ltd.

14.直接内存访问控制器（DMAC）

14.1. DMA 控制器的具体信息

本章介绍了 LT168 的直接内存访问控制器。

14.1.1. DMAC 功能

- 6 个可编程通道，以支持独立的 8、16 或 32 位单值或块传输
- 支持可变大小的队列和循环队列
- 独立配置为后增量寄存器的源地址和目标地址寄存器保持不变
- 每个传输都由外围设备、CPU、周期性定时器中断或 DMA 通道请求发起
- 外围的 DMA 请求源可能来自 QSPI，QADC
- 每个 DMA 信道能够在完成单个值或块传输时选择性地向 CPU 发送中断请求
- DMA 可能在系统内存和所有可访问的内存映像位置之间进行传输，包括外围设备和寄存器
- DMA 支持以下功能：
  - Scatter Gather
  - Channel Linking
  - Inner Loop Offset
  - Arbitration
    - Fixed Group, Fixed Channel
    - Round Robin Group, Fixed Channel
    - Round Robin Group, Round Robin Channel
    - Fixed Group, Round Robin Channel
  - Channel Preemption
  - Cancel Channel Transfer
- 中断-DMA 对每个实现的通道都有一个中断请求和一个组合的 DMA 错误中断，以标记到系统的传输错误。每个通道 DMA 中断都可以启用或禁用，并提供已完成传输的通知。有关这些中断的分配，请参考参考手册的 EIC 章节中的中断向量。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

14.1.2. 通道分配

从块到 DMA 通道的 DMA 请求之间的分配如表 14-1 所示.

表 14-1：DMA 通道分配

DMA 请求	通道	描述
ADC_ISR[EMPTY]	0	qadc_dma_req，当 FIFO 中的数据号不为空，且设置了位 ADC_CFGR1 [DMAEN]时，
QSPI0 DMA 接收请求	1	当接收 FIFO 中的有效数据条目数等于或高于 DMARDLR[DMARDL] + 1 和 DMACR[RDMAE] = 1 时，将生成 QSPI0 dma_rx_req
QSPI0 DMA 发送请求	2	当传输 FIFO 中的有效数据条目数等于或低于 DMATDLR[DMATDL]和 DMACR[TDMAE] = 1 时，将生成 QSPI0 dma_tx_req 信号
QSPI1 DMA 接收请求	3	当接收 FIFO 中的有效数据条目数量等于或高于 DMARDLR[DMARDL] + 1 和 DMACR[RDMAE] = 1 时，将生成 QSPI1 dma_rx_req
QSPI1 DMA 发送请求	4	当传输 FIFO 中的有效数据项数等于或低于 DMATDLR[DMATDL]和 DMACR[TDMAE] = 1 时，将生成 QSPI1 dma_tx_req 信号
QSPI2 DMA 接收请求	5	当接收 FIFO 中的有效数据条目数等于或高于 DMARDLR[DMARDL] + 1 和 DMACR[RDMAE] = 1 时，将生成 QSPI2 dma_rx_req
QSPI2 DMA 发送请求	6	QSPI2 dma_tx_req 信号产生时，有效的数据条目数在传输中 FIFO 等于或低于 DMATDLR[DMATDL]和 DMACR[TDMAE] = 1
PIT1 DMA 请求	7	当 PIT0 计数器达到 0x0000 和 PCSR[PDMAE] = 1 时，就会产生 pit0_dma_req 信号
PIT2 DMA 请求	8	当 PIT0 计数器达到 0x0000 和 PCSR[PDMAE] = 1 时，就会产生 pit0_dma_req 信号
PIT3 DMA 请求	9	当 PIT0 计数器达到 0x0000 和 PCSR[PDMAE] = 1 时，就会产生 pit0_dma_req 信号
SCI0 TX DMA 请求	10	当传输 FIFOis 中的数据字数等于或小于 SCI_WATER[txwater]所表示的数字时，就会产生 sci0_tx_req 信号)
SCI0 RX DMA 请求	11	当接收缓冲区中的数据字数大于 SCI_WATER[RXWATER]所表示的数字时，将生成 sci0_rx_req 信号
SCI1 TX DMA 请求	12	当传输 FIFOis 中的数据字数等于或小于 SCI_WATER[txwater]所表示的数字时，就会产生 sci1_tx_req 信号)
SCI1 RX DMA 请求	13	当接收缓冲区中的数据字数大于 SCI_WATER[RXWATER]所表示的数字时，将生成 sci1_rx_req 信号

©Levetop Semiconductor Co., Ltd.

DMA 请求	通道	描述
SCI2 TX DMA 请求	14	当传输 FIFOis 中的数据字数等于或小于 SCI_WATER[txwater]所表示的数字时，就会产生 sci2_tx_req 信号)
SCI2 RX DMA 请求	15	当接收缓冲区中的数据字数大于 SCI_WATER[RXWATER]所表示的数字时，将生成 sci2_rx_req 信号

14.2. 功能介绍

直接存储器访问控制器(DMA)能够通过 16 个可编程通道执行复杂的数据移动,而来自主机处理器的最小干预。硬件微架构包括执行源地址计算和目标地址计算的 DMA 引擎，以及实际的数据移动操作，以及包含通道的传输控制描述符（TCD）的基于 SRAM 的内存。这个实现将最小化到整体块的大小。

图 14-1 这是 DMA 模块的结构图。

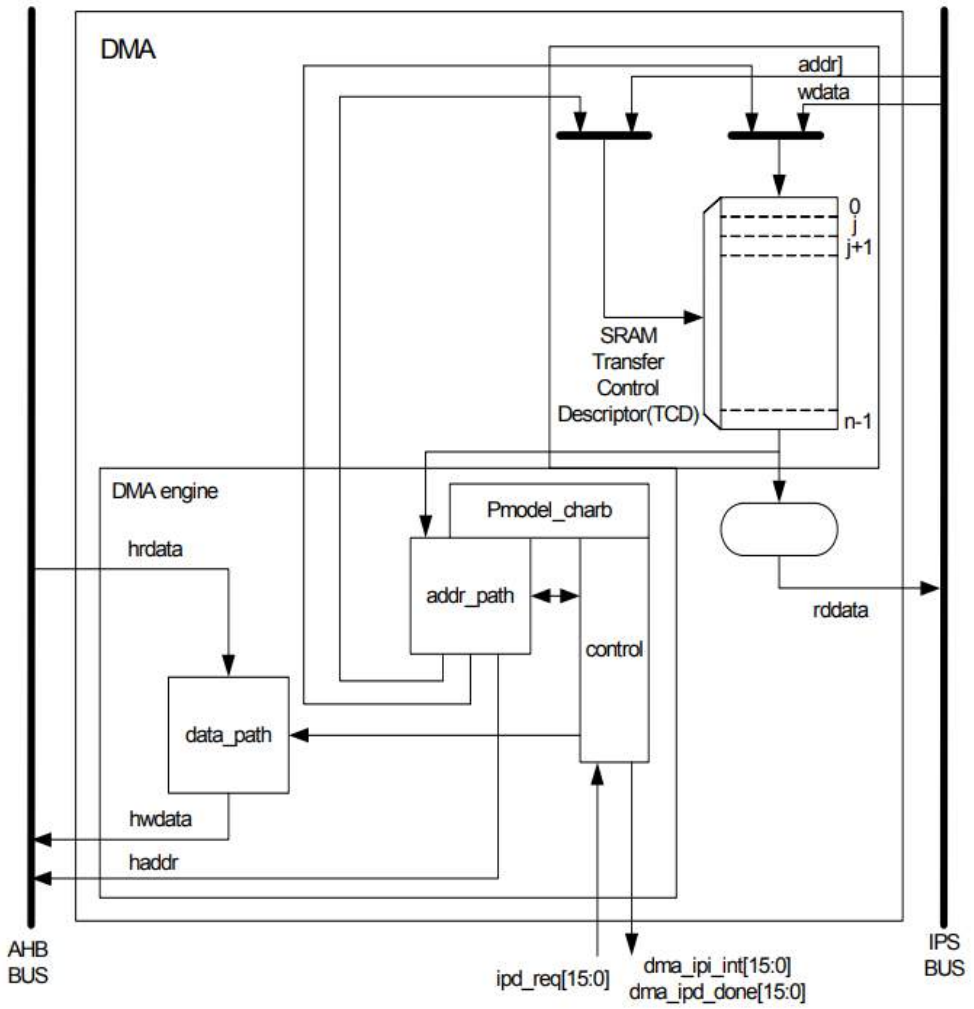


图 14-1：DMA 结构图

©Levetop Semiconductor Co., Ltd.

14.2.1. 模块特点

DMA 模块支持以下功能：

- 所有的数据移动通过双地址传输：从源代码读取，写到目的地
  - 可编程的源，目标地址，传输大小，并支持增强的寻址模式
- 传输控制描述符组织以支持双深度嵌套传输操作
  - 内部数据传输循环由一个“次要”字节传输计数定义
  - 一个外部的数据传输循环是由一个“主要的”迭代计数来定义的
- 通过以下三种方式之一提出的渠道服务请求：
  - 显式软件启动
  - 通过信道到信道的连接机制进行启动，以进行连续传输。在小回路和/或大回路的末端的独立通道连接
  - 外围设备有节奏的硬件请求（每通道一个）
  - 对于所有三种方法，每次执行小循环都需要一个服务请求
- 支持固定优先级和循环信道仲裁
- 通过可选的中断请求报告通道完成情况
  - 每个通道有一个中断，可选择在完成主要迭代计数时断言
  - 错误终止可选地启用每个通道，并逻辑求和形成少量的错误中断输出
- 支持分散/收集 DMA 处理
- 支持复杂的数据结构
- 支持通过软件或硬件进行传输

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

15.Efuse 控制模块（EFM）

15.1. 功能介绍

Efuse 模块是一个并行/并联输出的电保险丝 IP，有 512 位内部非易失性一次性可编程 Efuse 存储。通过一个串行接口，每次在程序模式下都可以编程 1 位，在读取模式下也可以一次读取 8 位。Efuse 阵列用于存储芯片的修剪和配置位。

15.2. 控制模块特点

EFM 的特点包括：

- 一次性可编程的非易失性 EFUSE 存储单元被组织为 64x8 位
- EFUSE 宏有三种操作模式：
- 程序模式
- 读取模式
- 非活动模式

15.3. 内存映射和寄存器

15.3.1. 内存映射

Efuse 模块还包含一组控制寄存器。这些寄存器的内存映射如表 15-1 所示.

表 15-1：Efuse 寄存器内存映射

地址偏移	Bits[31:0]	访问权限 [1, 2]
0x0000	EFMCR	S/U
0x0080~0x00BF	EFMDR[3]	S/U

1. 仅限 S = CPU 监控模式访问。
2. A 在用户模式下只处理主管地址位置没有影响，并导致周期终止传输错误。
3. EFMDR 只能通过 8 位访问来读取。16 位和 32 位的读取数据都将返回错误的读取数据。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

15.3.2. 寄存器描述

Efuse 模块寄存器图如下所示。Efuse 的基本地址为 0x4012_0000。

15.3.2.1. Efuse 配置寄存器（EFMCR）

Efuse 配置寄存器（EFMCR）未打开存储库，用于配置和控制 Efuse 阵列的操作。

地址：EFM_BASEADDR+0x0000_0000

	31	30	29	28	27	26	25	24
读:	0	0	0	0	0	0	0	0
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	AEN_PULSE[7:0]							
写:								
复位:	0	0	0	1	0	0	0	0

	15	14	13	12	11	10	9	8
读:	读取 hold							
写:								
复位:	0	0	0	1	0	0	0	0

	7	6	5	4	3	2	1	0
读:	读取 setup[7:0]							
写:								
复位:	0	0	0	1	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 15-1：Efuse 模块配置寄存器（EFMCR）

AEN_PULSE[7:0] — Efuse AEN pulse width configuration

这些位决定了 Efuse 的 AEN 的宽度。注意，T 安不应小于 100ns。

$$TAEN = (AEN_PULSE[7:0] + 1) * TIPS_CLK$$

READ_HOLD[7:0] — Efuse AEN to Address and RDEN hold time configuration.

这些位决定了 AEN 到地址和 RDEN 的保持时间。请注意，THOLD 不应小于 100ns。

$$T_{HOLD} = (READ_HOLD[7:0] + 1) * T_{IPS_CLK}$$

READ_SETUP[7:0] — Efuse Address and RDEN to AEN setup time configuration.

这些位决定了地址和 RDEN 到 AEN 的设置时间。请注意，T 设置不应小于 100ns。

$$T_{SETUP} = (READ_SETUP[7:0] + 1) * T_{IPS_CLK}$$

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

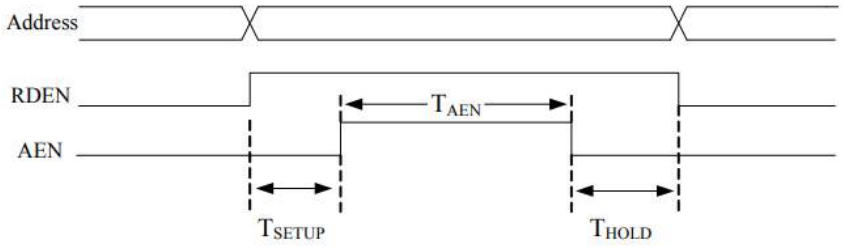


图 15-2：清除读取正时

15.3.2.2. 清除读取数据寄存器（EFMDR）

Efuse 读取数据寄存器（EFMDR）用于存储从 Efuse 阵列读取的数据，而 EFMDR 只能通过 8 位读取指令进行访问。

地址：EFM_BASEADDR+0x0000_0080~00BF

	位 7	6	5	4	3	2	1	位 0
读： 写：	DATA[7:0]							
复位：	0	0	0	0	0	0	0	0

图 15-3：Efuse 读取数据寄存器（EFMDR）

DATA[7:0] — The EFMDR stores the data read from Efuse array.

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

16.选项字节（OPB）

16.1. 内存映射和寄存器

16.1.1. 内存映射

选项字节内存映射如表 16-1 所示，其基本地址为 0x4012_0000，与 Efuse 模块相同。

表 16-1：寄存器内存映射

地址偏移	Bits[31:24]	Bits[23:16]	Bits[15:8]	Bits[7:0]	访问权限 [1,2]
0x001C	CCR		PVDC		S
0x0020	PLLOCKCR		EOSCST		S
0x0024	PVDFEVR		RFEVR		S
0x002C	FCR		保留 3		S
0x0030	保留				S
0x0034	LDOTCR	VREFTCR	ADCCDISR		S
0x0038	RTCTCR		保留		S

提示：

1. 仅限 S = CPU 监控模式访问。
2. A 在用户模式下只处理主管地址位置没有影响，并导致周期终止传输错误。
3. 写入保留地址位置没有影响，读取返回 0 秒。

16.1.2. 寄存器描述

16.1.2.1. 可编程电压检测器配置寄存器（PVDC）

地址：EFM_BASEADDR+0x0000_001C

	15	14	13	12	11	10	9	8
读:	PVDF	PVDPORRE	PVDTEST[1:0]		PVDOE	PVDRE	PVDIE	PVDE
写:								
复位:	0	注 1	0	0	1	注 1	0	1

	7	6	5	4	3	2	1	0
读:	0						PVDC[1:0]	
写:								
复位:	0	0	0	0	0	0	注 1	注 1

= 写入没有影响，访问终止，没有传输错误异常。

提示 1：由保险丝面积的值确定

图 16-1：可编程电压检测器配置寄存器（PVDC）

©Levetop Semiconductor Co., Ltd.

PVDF — Programmable Voltage Detector Flag

PVDF 表示 VDD 是否低于 PVD 阈值。POR 可以清除它。

1 = VDD33 低于 PVD 阈值

0 = VDD33 不低于 PVD 阈值

PVDPORRE — Program Voltage Detector (PVD) Power-On Reset Enable

端口显示程序电压检测器是否启用通电复位。启用后，在 PVD 复位后将设置 RSR[POR]标志。默认值将从 FUSE 区域加载。如果相应的熔断区域处于擦除状态，则清除该位，否则如果相应的熔断区域中的内容匹配，则开机后从熔断区域的相应位加载。此位可在通电复位后由软件设置。

1 = 当 VDD33 低于 PVD 阈值时，PVD 将产生接通电复位

0 = 当 VDD33 低于 PVD 阈值时，PVD 将不会产生通电复位

PVDTEST[1:0] — Write Access Enable Sequence Input

不能对 PVDC 寄存器进行更改，除非写入正确的序列。正确的序列是：2'b01 → 2'b10 → 2'b11。在用这个序列写入这两位后，这两位的值为 == 2'b11，然后就可以随意改变 PVDC 寄存器。当值等于 2'b11 时，写入 2'b00 可以清除这两个位。写入其他值没有效果，并返回 2'b11。

PVDOE — Programmable Voltage Detector Output Enable

1 = PVD 输出已启用

0 = PVD 输出已禁用

PVDRE — Program Voltage Detector (PVD) Reset Enable

PVDRE 显示是否启用了程序电压检测器复位。默认值将从 FUSE 区域加载。当启用和 PVD 孔被禁用时，在 PVD 复位后将设置 RSR[PVDF]标志。如果相应的熔断区域处于擦除状态，则清除该位，否则如果相应的熔断区域中的内容匹配，则开机后从熔断区域的相应位加载。复位后，可通过软件更改。

1 = PVD 复位

0 = PVD 复位

PVDIE — Programmable Voltage Detector Interrupt Enable

PVD IE 用于检查 VCC 是否低于 PVD 阈值

1 = 如果 VCC 低于 PVD 阈值，则将生成一个中断

0 = 如果 VCC 不低于 PVD 阈值，则不会产生任何中断。

PVDE — Programmable Voltage Detector Enable

PVDE 决定是否启用了 PVD。POR 可以清除它。

1 = 启用 PVD

0 = 禁用 PVD

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

PVDC[1:0] — Programmable Voltage Detector Configuration Value.

表 16-2：可编程电压检测器

PVDC	检测电压
2'b00	2.16V
2'b01	2.32V
2'b10	2.48V
2'b11	2.64V

默认值将从 FUSE 区域加载。如果相应的熔断区域处于擦除状态，则复位后将清除 PVDC[1:0]，否则如果相应的熔断区域的内容匹配，则开机后从熔断区域加载这些位。这些位可以在电源复位后通过软件改变。

16.1.2.2. 客户配置寄存器（CCR）

地址：EFM_BASEADDR+0x0000_001E

	15	14	13	12	11	10	9	8
读:	RGBEN	0	RTC_INTERFACE_EN	PGMDIS	RSTOUTDIS	CLKOUTDIS	RTC_INTERFACE_LVD_ISOEN	QSPI2EN
写:								
复位:	0	0	0	注 1	注 1	注 1	0	0

	7	6	5	4	3	2	1	0
读:	0	0	ADCCSPOR	PVDCE	0	EOSCSTE	CCRTEST[1:0]	
写:								
复位:	0	0	注 1	注 1	0	注 1	0	0

= 写入没有影响，访问终止，没有传输错误异常。

提示 1：由保险丝区域的值决定

图 16-2：客户配置寄存器（CCR）

RGBEN — RGB Interface Enable Control Bit

RGBEN 决定了 RGB 接口是否启用。

1 = RGB 接口**启用**

0 = RGB 接口禁用

表 16-3：RGB 接口启用控制

RGB 禁用	RGB 启用
EBI_D[15:0]	TFT_RGB[15:0]
EBI_WR#	TFT_VS
EBI_RD#	TFT_HS
EBI_RS	TFT_DE
EBI_CS#	TFT_PCLK

©Levetop Semiconductor Co., Ltd.

RTC_INTERFACE_EN — This bit determines whether the RTC analog module can be reconfigured or not when PRCSR[Dir] or PRENR[RTC_EN_Dir] is set.

1 = RTC 模拟模块可以重新配置。

0 = RTC 模拟模块无法重新配置。

PGMDIS — Programming Debug Interface Disable Bit

PGMDIS 显示编程调试接口是否被禁用。默认值将从 FUSE 区域加载。如果相应的保险丝区域处于擦除状态，则该位将被清除，否则，如果相应的保险丝区域中的内容被匹配，则将在开机后设置该位。复位后，可通过软件更改。

1 = 编程调试功能已被禁用

0 = 编程调试功能

表 16-4：编程调试接口控制

编程启用	编程禁用
PGMCK	RXD2
PGMIO	TXD2

RSTOUTDIS — RSTOUT Disable Bit

结果显示是否禁用 RSTOUT pin 功能并启用 INT1[6]。默认值将从 FUSE 区域加载。如果相应的保险丝区域处于擦除状态，则该位将被清除，否则，如果相应的保险丝区域中的内容被匹配，则将在开机后设置该位。复位后，可通过软件更改。

1 = RSTOUT Pin 功能被禁用

0 = RSTOUT Pin 功能已启用

表 16-5：RSTOUT 禁用控制

RSTOUT 启用	RSTOUT 禁用
RSTOUT	INT1[6]

CLKOUTDIS — CLKOUT Disable Bit

CLKOUTDIS 结果显示 CLKOUT pin 功能是否已禁用，INT1[0]是否已启用。默认值将从 FUSE 区域加载。如果相应的保险丝区域处于擦除状态，则该位将被清除，否则，如果相应的保险丝区域中的内容被匹配，则将在开机后设置该位。复位后，可通过软件更改。

1 = CLKOUT Pin 功能被禁用

0 = CLKOUT Pin 功能已启用

©Levetop Semiconductor Co., Ltd.

表 16-6：CLKOUT 禁用控制

CLKOUT 启用	CLKOUT 禁用
CLKOUT	INT1[0]

RTC_INTERFACE_LVD_ISOEN — This bit determines whether the RTC analog interface will be isolated or not when Program Voltage Detector event happens.

1 = RTC 模拟接口不隔离。

0 = 当发生低电压时，RTC 模拟接口将被隔离。

QSPI2EN — QSPI2 Interface Enable Control Bit

QSPI2EN 确定 QSPI2 接口是否启用 d。

1 = QSPI2 接口启用

0 = QSPI2 接口被禁用

表 16-7：QSPI2 接口启用控制

QSPI2 禁用	QSPI2 启用
PWM0[0]	QSCS2#
PWM0[1]	QSCK2
PWM0[2]	QSIO2[0]
PWM0[3]	QSIO2[1]
PWM1[0]	QSIO2[2]
PWM1[1]	QSIO2[3]

ADCCSPOR — ADC Channel Setting Bit after power-on

附件显示附件在开机后是否从保险丝区域加载。默认值将从 FUSE 区域加载。如果相应的保险丝区域处于擦除状态，则清除该位，否则，如果相应的保险丝区域中的内容相匹配，则在开机后设置该位。

1 = 附件在通电后从保险丝区域加载

0 = 电源在开机后未从保险丝区域加载

PVDCE — Program Voltage Detector (PVD) Configuration Enable

PVDCE 显示通电后是否从保险丝区域将程序电压检测器配置加载。如果相应的保险丝区域处于擦除状态，则清除该位，否则，如果相应的保险丝区域中的内容相匹配，则在开机后设置该位。

1 = PVD 配置在通电后从保险丝区域加载

0 = PVD 配置开机后未从保险丝区域加载

EOSCSTE — External High Speed Oscillator Stable Time Configuration Enable

该信号显示通电后是否从保险丝区域加载外部高速振荡器稳定时间配置。如果相应的保险丝区域处于擦除状态，则清除该位，否则，如果相应的保险丝区域中的内容相匹配，则在开机后设置该位。

©Levetop Semiconductor Co., Ltd.

1 = 外部高速振荡器（FXOSC）稳定时间配置在通电后从保险丝区域加载

0 = 外部高速振荡器（FXOSC）通电后未从保险丝区域加载

CCRTEST[1:0] — Write Access Enable Sequence Input

不能更改 CCR 寄存器的可写位，除非写入了正确的序列。正确的序列是：2'b01 → 2'b10 → 2'b11。在用这个序列写入这两个位后，这两个位的值 == 2'b11，然后可以随意改变 CCR 寄存器的可写位。当值等于 2'b11 时，写入 2'b00 可以清除这两个位。写入其他值没有效果，并返回 2'b11。

16.1.2.3. 外部高速振荡器稳定时间配置寄存器（EOSCST）

地址：EFM_BASEADDR+0x0000_0020

	15	14	13	12	11	10	9	8
读:	EOSCTEST[1:0]		0	0	EOSCST[11:8]			
写:								
复位:	0	0	0	0	注 1	注 1	注 1	注 1

	7	6	5	4	3	2	1	0
读:	EOSCST[7:0]							
写:								
复位:	注 1	注 1	注 1	注 1	注 1	注 1	注 1	注 1

= 写入没有影响，访问终止，没有传输错误异常。

注 1：由保险丝面积的值确定

图 16-3：外部高速振荡器稳定时间配置寄存器（EOSCST）

EOSCTEST[1:0] — Write Access Enable Sequence Input

不能更改 EOSCST 寄存器的可写位，除非写入了正确的序列。正确的序列是：2'b01 → 2'b10 → 2'b11。在用这个序列写入这两个位后，这两个位的值 == 2'b11，然后可以随意更改 EOSCST 寄存器的可写位。当值等于 2'b11 时，写入 2'b00 可以清除这两个位。写入其他值没有效果，并返回 2'b11。

EOSCST[11:0] — External High Speed Oscillator Stable Time Value.

默认值将从 FUSE 区域加载。如果相应的熔断器区域处于擦除状态，则复位后 EOSCST[11:0]值为 12'h3ff，否则如果相应的保险丝区域中的内容匹配，则开机后从熔断器区域加载。软件可以在复电后更换。在外部振荡器被启用后，它将等待 128 KHz 振荡器的 EOSCST[11:0]周期，然后时钟的门将被打开。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

16.1.2.4. PLL 锁定时间配置寄存器（请）

地址：EFM_BASEADDR+0x0000_0022

	15	14	13	12	11	10	9	8
读:	PLLST[15:14]		PLLST[13:8]					
写:	PLLTEST[1:0] / PLLST[15:14]							
复位:	0	0	1	1	1	1	1	1

	7	6	5	4	3	2	1	0
读:	PLLST[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

图 16-4：PLL 锁定时间配置寄存器（请）

PLLST[15:0] — PLL Lock Time Value.

启用后，PLL 将等待外部高速振荡器（FXOSC）的 PLLST[15:0]周期，然后时钟的门将被打开。

PLLTEST[1:0] — Write Access Enable Sequence Input

不能对 PLLST 寄存器进行更改，除非写入正确的序列。正确的序列是：2'b01 → 2'b10 → 2'b11。在用这个序列写入这两位后，这两位的值为 == 2'b11，然后就可以随意改变 PLLST 寄存器。当值等于 2'b11 时，任何写入操作都可以清除这两个位。

提示：PLLST[15:14]只有在其值等于 2'b11 时才可写。

16.1.2.5. 复位引脚过滤器启用和值寄存器（RFEVR）

地址：EFM_BASEADDR+0x0000_0024

	15	14	13	12	11	10	9	8
读:	0	0	RFEVRTEST[1:0]		0	0	0	0
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	RFV[7:0]							
写:								
复位:	注 1	注 1	注 1	注 1	注 1	注 1	注 1	注 1

= 写入没有影响，访问终止，没有传输错误异常。

注 1：由保险丝面积的值确定

图 16-5：复位引脚过滤器启用和值寄存器（RFEVR）

©Levetop Semiconductor Co., Ltd.

RFEVRTEST [1:0] — Write Access Enable Sequence Input

不能更改 RFEVR 寄存器的可写位，除非写入了正确的序列。正确的序列是：2'b01 → 2'b10 → 2'b11。在用这个序列写入这两个位后，这两个位的值 == 2'b11，然后可以随意更改 RFEVR 寄存器的可写位。当值等于 2'b11 时，写入 2'b00 可以清除这两个位。写入其他值没有效果，并返回 2'b11。

RFV[7:0] — RESET Pin Filter Value

默认值将从 FUSE 区域加载。如果相应的 FUSE 区域处于擦除状态，则复电后 RFV[7:0]值为 8'h0D，否则如果相应的 FUSE 区域的内容匹配，则开机后从 FUSE 区域加载这些位。这些位可以在电源复位后通过软件改变。

16.1.2.6. 可编程电压检测器滤波器启用和值寄存器（PVDFEVR）

地址：EFM_BASEADDR+0x0000_0026

	15	14	13	12	11	10	9	8
读:	PVDFE	0	PVDFTEST[1:0]		0			
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	PVDFV[7:0]							
写:								
复位:	注 1	注 1	注 1	注 1	注 1	注 1	注 1	注 1

= 写入没有影响，访问终止，没有传输错误异常。

注 1： 由保险丝面积的值确定

图 16-6：可编程电压检测器滤波器启用和值寄存器（PVDFEVR）

PVDFE — Programmable Voltage Detector Filter Enable.

PVDFE 显示程序电压检测器滤波器是否已启用。默认值将从 FUSE 区域加载。如果相应的熔断器区域处于擦除状态，则在开机复位后清除 PVDFE，否则，如果相应的保险丝区域中的内容相匹配，则在开机后从熔断器区域加载这些位。复位后，可通过软件更改。

1 = PVD 筛选器启用

0 = PVD 筛选器已被禁用

PVDFTEST[1:0] — Write Access Enable Sequence Input

不能更改的 PVDFEVR 寄存器的可写位，除非写了正确的序列。正确的序列是：2'b01 → 2'b10 → 2'b11。通过这个序列写入这两个位后，这两个位的值 == 2'b11，然后可以随意更改 PVDFEVR 寄存器的可写位。当值等于 2'b11 时，写入 2'b00 可以清除这两个位。写入其他值没有效果，并返回 2'b11。

©Levetop Semiconductor Co., Ltd.

PVDFV[7:0] — Programmable Voltage Detector Filter Value

默认值将从 FUSE 区域加载。如果相应的 FUSE 区域处于擦除状态，则复位后 PVDFV[7:0]值为 8'h0，否则如果相应的 FUSE 区域的内容匹配，则在上电后从 FUSE 区域加载这些位。这些位可以在电源复位后改变。

16.1.2.7. 工厂配置寄存器（FCR）

地址：EFM_BASEADDR+0x0000_002E

	15	14	13	12	11	10	9	8
读:	0	0	RTCTE	TMDIS	RESETFTE	LDO1P2TE	VREF1P2TE	PVDFTE
写:								
复位:	0	0	注 1	注 1	注 1	注 1	注 1	注 1

	7	6	5	4	3	2	1	0
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

注 1：由保险丝面积的值确定

图 16-7：工厂配置寄存器（FCR）

RTCTE — RTC Bias/Cap rimming Enable

RTCT E 显示是否修剪了 RTC 偏差或电容。默认值将从 FUSE 区域加载。如果相应的保险丝区域处于擦除状态，则清除该位，否则，如果相应的保险丝区域中的内容相匹配，则在开机后设置该位。

1 = RTC 偏差或电容被修剪

0 = RTC 偏差或电容没有被修剪

TMDIS — Test Mode Disable Bit

TMDIS 显示是否禁用了测试模式（STB 除外）。默认值将从 FUSE 区域加载。如果相应的保险丝区域处于擦除状态，则清除该位，否则，如果相应的保险丝区域中的内容相匹配，则在开机后设置该位。

1 =测试模式禁用

0 = 测试模式启用

RESETFTE — Reset Pin Filter Trimming Enable

复位显示是否修剪复位引脚过滤器。默认值将从 FUSE 区域加载。如果相应的保险丝区域处于擦除状态，则清除该位，否则，如果相应的保险丝区域中的内容相匹配，则在开机后设置该位。

1 = 复位引脚过滤器被修剪

0 = 复位引脚过滤器未被修剪

©Levetop Semiconductor Co., Ltd.

LDO1P2TE — LDO1P2 Trimming Enable

LDO1P2TE 显示 LDO1P2 是否被修剪。默认值将从 FUSE 区域加载。如果相应的保险丝区域处于擦除状态，则清除该位，否则，如果相应的保险丝区域中的内容相匹配，则在开机后设置该位。

1 = LDO1P2 被修剪

0 = LDO1P2 未被修剪

VREF1P2TE — VREF1P2 Module Trimming Configuration Register

VREF1P2TE 显示 VREF1P2 模块是否被修剪。默认值将从 FUSE 区域加载。如果相应的保险丝区域处于擦除状态，则清除该位，否则，如果相应的保险丝区域中的内容相匹配，则在开机后设置该位。

1 = 内部 VREF1P2 模块被修剪

0 = 内部 VREF1P2 模块未被修剪

PVDFTE — PVD Filter Trimming Enable

PVDFTE 显示 PVD 过滤器是否被修剪。默认值将从 FUSE 区域加载。如果相应的保险丝区域处于擦除状态，则清除该位，否则，如果相应的保险丝区域中的内容相匹配，则在开机后设置该位。

1 = PVD 过滤器已被修剪

0 = PVD 过滤器未被修剪

16.1.2.8. 信道禁用配置寄存器（附件）

地址：EFM_BASEADDR+0x0000_0034

	15	14	13	12	11	10	9	8
读:	ADCCDISTEST[1:0]		0					
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	ADCCDIS[7:0]							
写:								
复位:	注 1	注 1	注 1	注 1	注 1	注 1	注 1	注 1

= 写入没有影响，访问终止，没有传输错误异常。

注 1：由保险丝面积的值确定

图 16-8：ADC 信道禁用配置寄存器（ADCCDISR）

ADCCDISTEST[1:0] — Write Access Enable Sequence Input

除非将正确的序列写入 addctist，否则不能更改可写位[7:0]。正确的序列是：2'b01 → 2'b10 → 2'b11。用这个序列写入这两个位后，这两个位的值 == 2'b11，然后可以随意更改测试寄存器的可写位。当值等于 2'b11 时，写入 2'b00 可以清除这两个位。写入其他值没有效果，并返回 2'b11。

©Levetop Semiconductor Co., Ltd.

ADCCDIS[7:0] — ADC Channel Disable Configuration

确定 AIN[7:0]是作为 ADC 通道还是作为 GPIO 通道。如果相应的 FUSE 区域处于清除状态，则清除这些位，ADC 信道功能有效，否则，如果相应的 FUSE 区域中的内容相匹配，则从 FUSE 区域加载 ADCCDIS[7:0]。这些位可以在电源复位后改变。

表 16-8：ADC 通道禁用控制

ADC 通道禁用	ADC 通道启用
INT0[0]	AIN[0]
INT0[1]	AIN[1]
INT0[2]	AIN[2]
INT0[3]	AIN[3]
INT0[4]	AIN[4]
INT0[5]	AIN[5]
INT0[6]	AIN[6]
INT0[7]	AIN[7]

16.1.2.9. VREF1P2 修剪配置寄存器（VREFTCR）

地址：EFM_BASEADDR+0x0000_0036

	7	6	5	4	3	2	1	0
读: 写:	VREFTCTEST[1:0]		VREFPD	VREFTRIM[4:0]				
复位:	0	0	注 1	注 1	注 1	注 1	注 1	注 1

注 1：由保险丝面积的值确定

图 16-9：VREF 修剪配置寄存器（VREFTCR）

VREFTCTEST[1:0] — WSFTCR Write Access Sequence In

除非写入正确的数据序列，否则不能更改虚拟测试寄存器的可写入位。正确的序列是：2'b01 → 2'b10 → 2'b11。在用这个序列写入这两个位后，这两个位的值 == 2'b11，然后可以随意更改测试寄存器的可写位。当值等于 2'b11 时，写入 2'b00 可以清除这两个位。写入其他值没有效果，并返回 2'b11。

VREFPD — Internal VREF1P2 Module Power Down. The setup time is 10us after Internal VREF1P2 Module is turn on.

默认值将从 FUSE 区域加载。如果相应的保险丝区域处于擦除状态，则在开机复位后设置 VREFPD，否则，如果相应的保险丝区域中的内容相匹配，则在开机后从保险丝区域加载这些位。复位后，可通过软件更改。

1 = 内部 VREF1P2 模块断电

0 = 内部 VREF1P2 模块上电

©Levetop Semiconductor Co., Ltd.

VREFTRIM[4:0] — VREF1P2 Module Trimming Value

默认值将从 FUSE 区域加载。如果相应的保险丝区域处于擦除状态，则开机复位后的 VREFTRIM[4:0]为 5'h18，否则，如果相应的保险丝区域中的内容相匹配，则这些位在开机后从保险丝区域加载。这些位可以在电源复位后通过软件改变。

16.1.2.10. LDO1P2 修剪配置寄存器（LDOTCR）

地址：EFM_BASEADDR+0x0000_0037

	7	6	5	4	3	2	1	0
读:	LDO1P2TC[7:6]		LDO1P2TC[5:0]					
写:	LDO1P2TEST[1:0] / LDO1P2TC[7:6]							
复位:	注 1	注 1	注 1	注 1	注 1	注 1	注 1	注 1

注 1： 由保险丝面积的值确定

图 16-10：LDO1P2 修剪配置寄存器（LDOTCR）

LDO1P2TC[7] — LDO over-current limit function control bit

默认值将从 FUSE 区域加载。如果相应的 FUSE 区域处于擦除状态，则开机复位后 LDO1P2TC[7]为 1'b0，否则如果相应的 FUSE 区域的内容匹配，则在开机后从 FUSE 区域加载。这个位可以在电源复位后改变。

1 = LDO 过电流限制功能将被关闭。

0 = LDO 过流限制功能。

LDO1P2TC[6] — LDO1P2 power down control bit

默认值将从 FUSE 区域加载。如果相应的 FUSE 区域处于擦除状态，则开机复位后 LDO1P2TC[6]为 1'b0，否则如果相应的 FUSE 区域的内容匹配，则在开机后从 FUSE 区域加载。这个位可以在电源复位后改变。

1 = LDO1P2 将断电。

0 = LDO1P2 将通电。

LDO1P2TC[5:0] — LDO1P2 Trimming Value

默认值将从 FUSE 区域加载。如果相应的 FUSE 区域处于擦除状态，则开机复位后 LDO1P2TC[5:0]为 6'h00，否则如果相应的 FUSE 区域中的内容相匹配，则在开机后从 FUSE 区域加载这些位。这些位可以在电源复位后通过软件改变。

LDO1P2TEST[1:0] — LDO1P2TC Write Access Sequence In

不可写的 LDO1P2TC 寄存器不能改变，除非写入正确的数据序列。正确的序列是：2'b01 → 2'b10 → 2'b11。用这个序列写入这两个位后，这两个位的值 == 2'b11，然后可以随意改变 LDO1P2TC 寄存器的可写位。当值等于 2'b11 时，任何写入操作都可以清除这两个位。

提示： 只有当 LDO1P2TEST[1:0]的值等于 2'b11 时，LDO1P2TC[7:6]才能写入。

©Levetop Semiconductor Co., Ltd.

16.1.2.11. RTC 修剪配置寄存器（RTCTCR）

地址：EFM_BASEADDR+0x0000_003A

	15	14	13	12	11	10	9	8
读: 写:	RTCTCTEST[1:0]		0				RTC_TEST_MODE	Trim_Load
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读: 写:	RTC_BIAS_TRIM[3:0]				RTC_CAP_TRIM[3:0]			
复位:	注 1	注 1	注 1	注 1	注 1	注 1	注 1	注 1

= 写入没有影响，访问终止，没有传输错误异常。

注 1：由保险丝面积的值确定

图 16-11：RTC 修剪配置寄存器（RTCTCR）

RTCTCTEST[1:0] — RTCTC Write Access Sequence In

RTCTC 寄存器的可写位不能更改，除非写入了正确的数据序列。正确的序列是：2'b01 → 2'b10 → 2'b11。在用这个序列写入这两个位后，这两个位的值 == 2'b11，然后可以随意更改 RTCTC 寄存器的可写位。当值等于 2'b11 时，写入 2'b00 可以清除这两个位。写入其他值没有效果，并返回 2'b11。

RTC_TEST_MODE — RTC Test Mode enable.

1 = RTC 测试模式已启用

0 = RTC 测试模式已被禁用

Trim_Load — Low to high edge will latch RTC_BIAS_TRIM and RTC_CAP_TRIM value into RTC analog module.

RTC_BIAS_TRIM[3:0] — RTC Oscillator Bias Trimming configuration bits.

默认值将从 FUSE 区域加载。如果相应的 FUSE 区域处于擦除状态，则复位后 RTC_BIAS_TRIM[3:0]为 4'b0111，否则如果相应的 FUSE 区域的内容匹配，则开机后从 FUSE 区域加载。这些位可以在电源复位后通过软件改变。

RTC_CAP_TRIM[3:0] — RTC Oscillator Load Capacitance C1 and C2 Trimming configuration bits.

默认值将从 FUSE 区域加载。如果相应的 FUSE 区域处于擦除状态，则在电源复位后 RTC_BIAS_TRIM[3:0]为 4'b1100，否则如果相应的 FUSE 区域中的内容匹配，则在开位后从 FUSE 区域加载。这些位可以在电源复位后通过软件改变。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

17.RGB 控制器模块（RGBC）

17.1. 功能介绍

LT168 有一个嵌入式 RGB 控制器，可以产生所有驱动 TFT LCD 显示器所需的必要信号。这些信号包括 16 位 RGB 数据总线、像素时钟、数据启用、水平同步和垂直同步。

RGB 控制器具有以下功能：

- 数据帧格式：红色：5，绿色：6，蓝色：5
- 支持的分辨率：
  - 480 x 480
  - 480x272 或以下
- 接口信号的定时是可配置的
- 极性和像素时钟极性是可配置的

17.2. 方块图

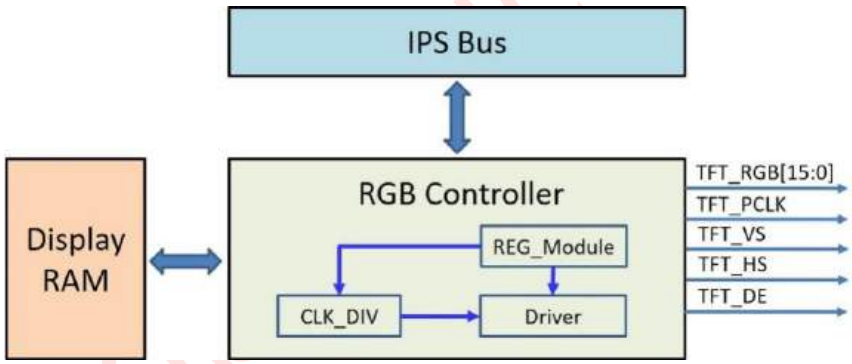


图 17-1：RGB 控制器的方块图

©Levetop Semiconductor Co., Ltd.

17.3. RGB 接口定义

通过 RGB 接口的显示操作与 VSYNC、HSYNC 和 TFT_PCLK 信号同步。这些数据只能在指定的区域内进行写入。后门廊和前门廊用于设置 RGB 接口的时间。

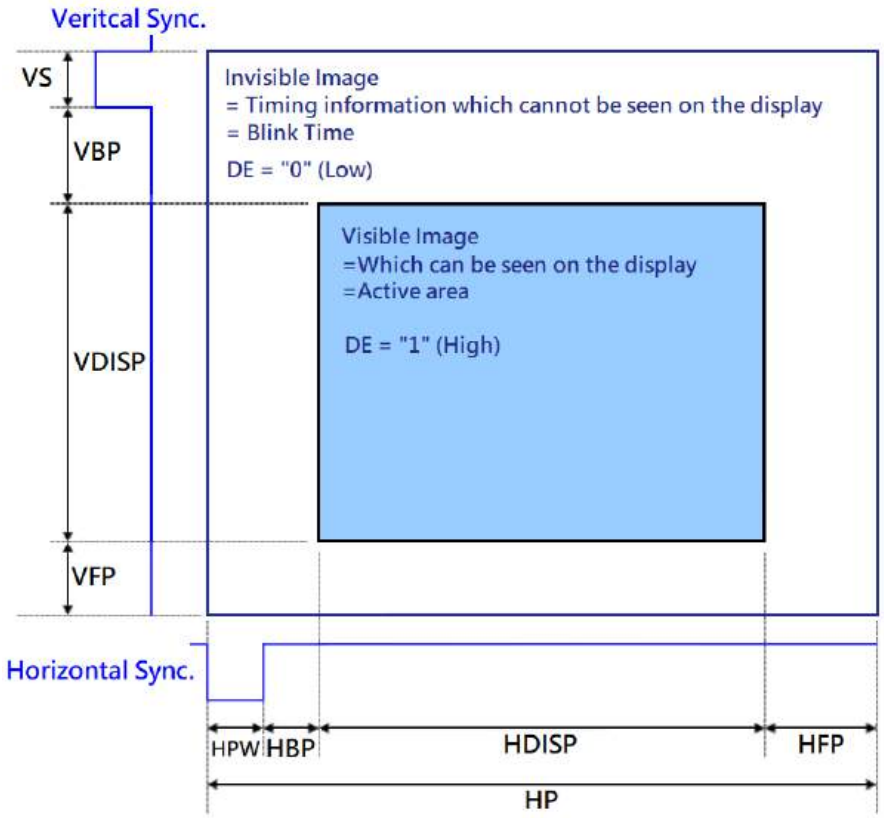


图 17-2：通过 RGB 接口显示 RAM 访问区域

©Levetop Semiconductor Co., Ltd.

17.4. RGB 接口定时

RGB 接口的时序图如下所示。

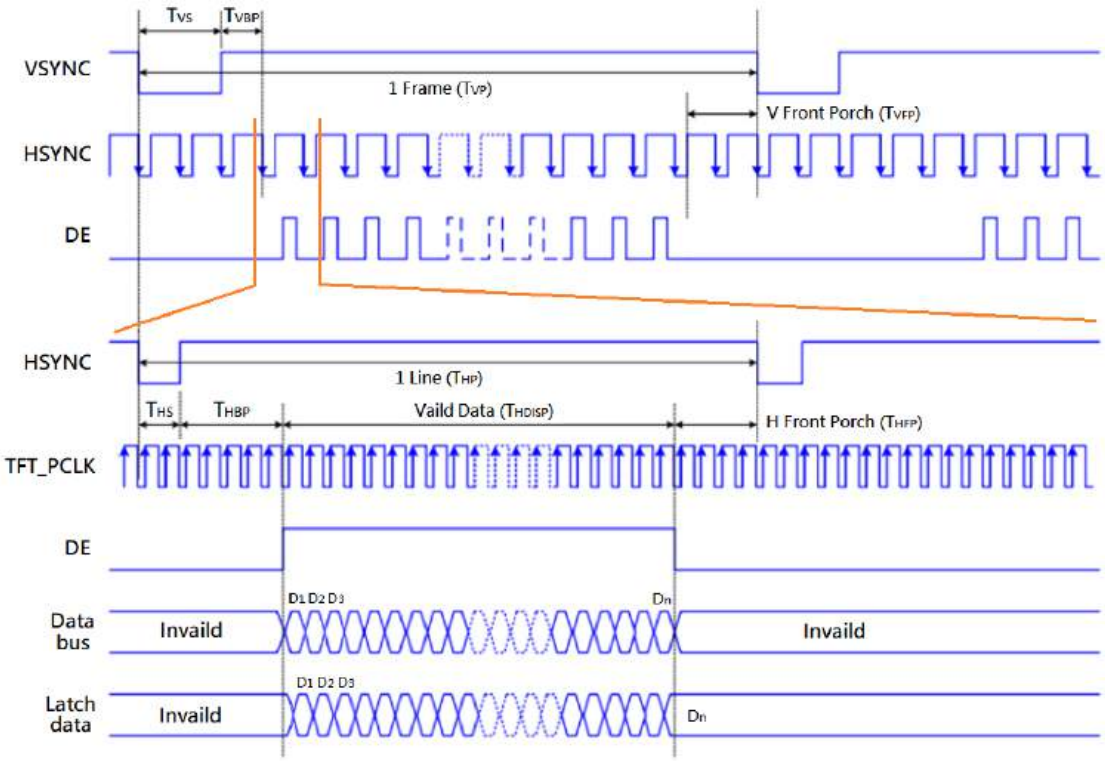


图 17-3：RGB 接口中的信号时序图

Levetop Semi

©Levetop Semiconductor Co., Ltd.

17.5. 内存映射和寄存器

RGB 控制器内存映射如表 17-1 所示，且其基座地址为 0x401A_0000。本节介绍了 RGB 控制器的内存映射和寄存器。

17.5.1. 内存映射

参见表 17-1 到，以获取 RGB 控制器内存映射的概述。

表 17-1：RGB 控制器模块内存映射

偏移地址	Bits[31:0]	访问权限
0x0000	Horizontal pulse width and back porch Register (THPB)[1]	R/W
0x0004	Horizntal period and front porch Register (THDF)[1]	R/W
0x0008	Vertical pulse width and back porch Register (TVPB)[1]	R/W
0x000C	Vertical period and front porch Register (TVDF)[1]	R/W
0x0010	Memory start address Register (MSADDR)	R/W
0x0014	Memory end address Register (MEADDR)	R/W
0x0018	Number of Frames Transferred Register (NFTR)	R
0x001C	Number of Frames Transferred Threshold Configuration Register(NFTTCR)	R/W
0x0020	RGB Control and Status Register(RCSR)	R/W

提示：这些寄存器只能在 RCSR[RGBE] = 1'b0 时进行更改。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

17.5.2. 寄存器描述

17.5.2.1. 水平脉冲宽度和后门廊寄存器（THPB）

地址：RGBC_BASEADDR+0x0000_0000

	31	30	29	28	27	26	25	24
读: 写:	HPW[15:8]							
复位:	0	0	0	0	0	0	0	0

	19	22	21	20	19	18	17	16
读: 写:	HPW[7:0]							
复位:	1	0	1	0	1	0	0	1

	11	14	13	12	11	10	9	8
读: 写:	HBP[15:8]							
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读: 写:	HBP[7:0]							
复位:	0	0	0	0	0	0	1	0

图 17-4：水平脉冲宽度和后门廊寄存器（THPB）

HPW[15:0] — Horizontal synchronizing pulse width (in pixel clock cycles).

$$T_{HPW} = HPW[15:0] * T_{pixel_clock}$$

HBP[15:0] — HSYNC back-porch pulse width (in pixel clock cycles).

$$T_{HBP} = HBP[15:0] * T_{pixel_clock}$$

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

17.5.2.2. 水平周期和前门廊寄存器（THDF）

地址：RGBC_BASEADDR+0x0000_0004

	31	30	29	28	27	26	25	24
读: 写:	HDISP[15;8]							
复位:	0	0	0	0	0	0	0	1

	23	22	21	20	19	18	17	16
读: 写:	HDISP[7:0]							
复位:	1	1	1	0	0	0	0	0

	15	14	13	12	11	10	9	8
读: 写:	HFP[15;8]							
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读: 写:	HFP[7:0]							
复位:	0	0	0	0	0	0	1	0

图 17-5：水平周期和前门廊寄存器（THDF）

HDISP[15:0] — HSYNC active pulse width (in pixel clock cycles).

$$T_{HDISP} = HDISP[15:0] * T_{pixel_clock}$$

HFP[15:0] — HSYNC front-porch pulse width (in pixel clock cycles).

$$T_{HFP} = HFP[15:0] * T_{pixel_clock}$$

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

17.5.2.3. 垂直脉冲宽度和后门廊寄存器（TVPB）

地址：RGBC_BASEADDR+0x0000_0008

	31	30	29	28	27	26	25	24
读: 写:	与[15:8]							
复位:	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
读: 写:	VS[7:0]							
复位:	0	0	0	0	1	0	1	0
	15	14	13	12	11	10	9	8
读: 写:	VBP[15;8]							
复位:	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
读: 写:	VBP[7:0]							
复位:	0	0	0	0	0	0	1	0

图 17-6：垂直脉冲宽度和后门廊寄存器（TVPB）

VS[15:0] — VSYNC active pulse width (in horizontal line cycles).

$$T_{VS} = VS[15:0] * T_{Horizontal_line}$$

VBP[15:0] — VSYNC back-porch pulse width (in horizontal line cycles).

$$T_{VBP} = VBP[15:0] * T_{Horizontal_line}$$

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

17.5.2.4. 垂直周期和前门廊寄存器（TVDF）

地址：RGBC_BASEADDR+0x0000_000C

	31	30	29	28	27	26	25	24
读: 写:	VDISP[15;8]							
复位:	0	0	0	0	0	0	0	1

	23	22	21	20	19	18	17	16
读: 写:	VDISP[7:0]							
复位:	0	0	0	1	0	0	0	0

	15	14	13	12	11	10	9	8
读: 写:	VFP[15;8]							
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读: 写:	VFP[7:0]							
复位:	0	0	0	0	0	0	1	0

图 17-7：垂直周期和前门廊寄存器（TVDF）

VDISP[15:0] — Vertical period pulse width (in horizontal line cycles).

$$T_{VDISP} = VDISP[15:0] * T_{Horizontal_line}$$

VFP[15:0] — VSYNC front-porch pulse width (in horizontal line cycles).

$$T_{VFP} = VFP[15:0] * T_{Horizontal_line}$$

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

17.5.2.5. 内存起始地址寄存器（msadr）

地址：RGBC_BASEADDR+0x0000_0010

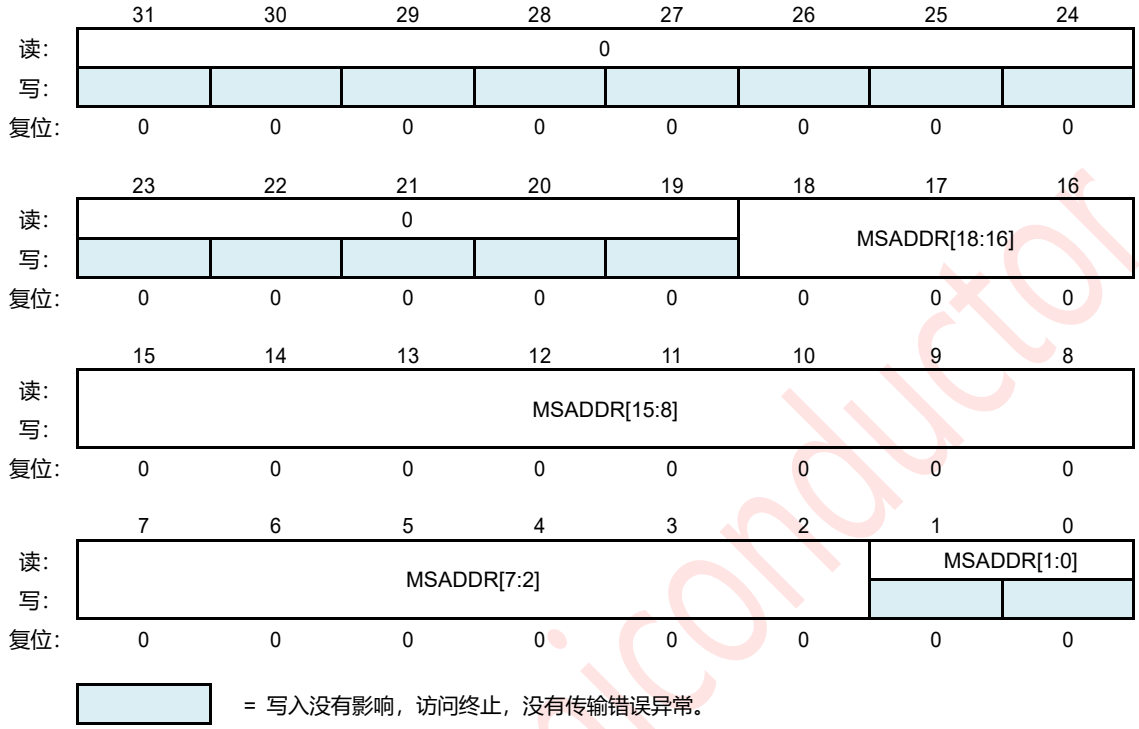


图 17-8：内存启动地址寄存器（MSADDR）

MSADDR[18:0] — The first pixel point offset address at Display RAM, and MSADDR[1:0] are fixed to 2'b00.

一旦配置了新的起始地址和结束地址，就可以完全传输前一帧的数据。

提示： MSADDR 和 MEADDR 必须通过单词访问来写入，否则将不会更改 MSADDR 和 MEADDR 的内容。当需要重新配置 MSADDR 或 MEADDR 时，必须写入两个地址，即使其他地址没有更改，否则重新配置的地址将不会生效。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

17.5.2.6. 内存端地址寄存器（MEADDR）

地址：RGBC_BASEADDR+0x0000_0014

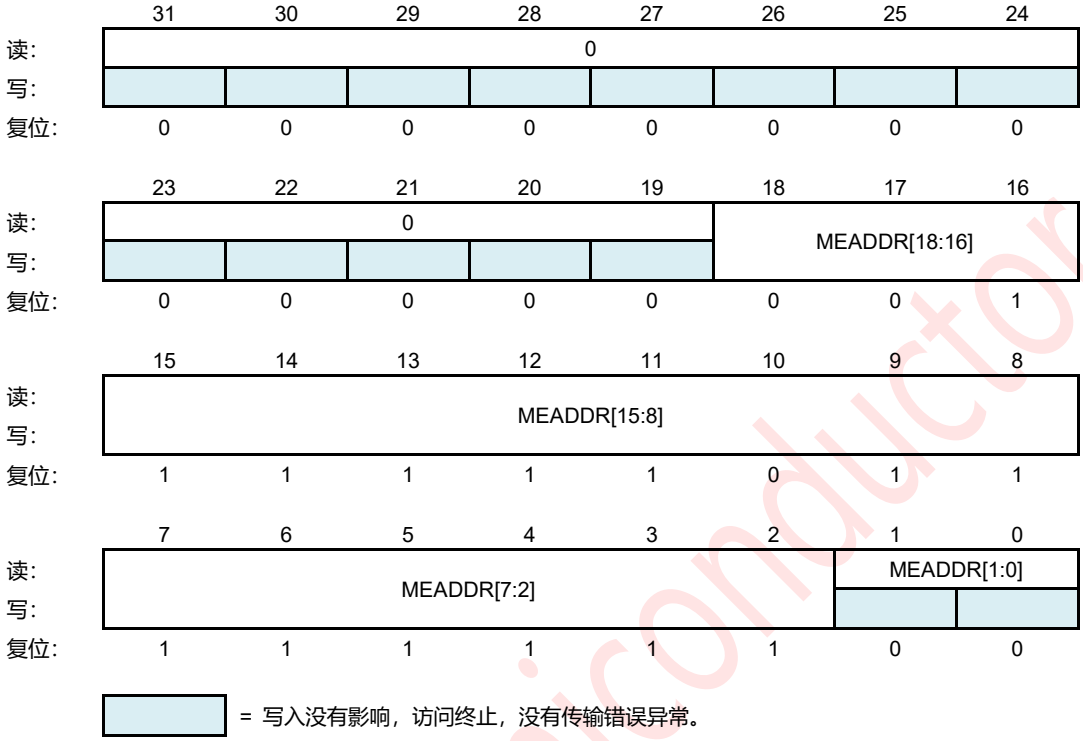


图 17-9：内存结束地址寄存器（MEADDR）

MEADDR[18:0] — The last pixel point offset address at Display RAM, and MEADDR[1:0] are fixed to 2'b00.

提示： MSADDR 和 MEADDR 必须以 Word 写入，否则 MSADDR 和 MEADDR 的内容。当需要重新配置 MSADDR 或 MEADDR 时，必须写入两个地址，即使另一个地址没有更改，否则重新配置的地址将不会生效。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

17.5.2.7. 已传输的帧数（NFTR）

地址：RGBC_BASEADDR+0x0000_0018

	31	30	29	28	27	26	25	24
读:	NFTR[31:24]							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	NFTR[24:16]							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	NFTR[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	NFTR[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 17-10：已转移寄存器的帧数（NFTR）

NFTR[31:0] — The amount of frames have been transmitted after RGB enabled by setting RGBE. NFT can be cleared by set RGBE to 1’b0 or system reset.

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

17.5.2.8. 传输阈值寄存器配置的帧数（NFTTCR）

地址：RGBC_BASEADDR+0x0000_001C

	31	30	29	28	27	26	25	24
读: 写:	ITR[31:24]							
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读: 写:	ITR[23:16]							
复位:	0	0	0	0	0	0	0	0

	11	14	13	12	11	10	9	8
读: 写:	ITR[15:8]							
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读: 写:	ITR[7:0]							
复位:	0	0	0	0	0	0	0	0

图 17-11：传输阈值寄存器配置的帧数（NFTTCR）

ITR[31:0] — Users can configure this register to determine how many frames of data transmission are completed and then RCSR[INTN] bit will be set.

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

17.5.2.9. RGB 控制和状态寄存器（RCSR）

地址：RGBC_BASEADDR+0x0000_00020

	31	30	29	28	27	26	25	24
读:	0				CLK_POL	DE_POL	HS_POL	VS_POL
写:								
复位:	0	0	0	0	1	1	1	1

	23	22	21	20	19	18	17	16
读:	0			DIV_NUM[4:0]				
写:								
复位:	0	0	0	0	0	1	0	1

	15	14	13	12	11	10	9	8
读:	0					PPVF	TCF	TCNF
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0			BEPO	DRC	TCIE	TCNIE	RGBE
写:								
复位:	0	0	0	0	1	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 17-12：RGB 控制和状态寄存器（RCSR）

CLK_POL — This bit determines which edge of pixel clock will be used to drive the TFT LCD display signals.

1 = 驱动 TFT 液晶显示器所需的所有必要信号都将由像素时钟的下降边缘驱动

0 = 驱动 TFT 液晶显示器所需的所有必要信号都将由像素时钟的上升边缘驱动

DE_POL — Polarity change of Data Enable.

1 = 无反转数据启用信号，主动高

0 = 反转数据启用信号，主动低

HS_POL — Polarity change of HSYNC.

1 = 无反转 HSYNC 信号，主动高

0 = 反转 HSYNC 信号，主动低

VS_POL — Polarity change of VSYNC.

1 = 无反转 VSYNC 信号，主动高

0 = 反转 VSYNC 信号，主动低

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

DIV_NUM[4:0] — The DIV_NUM bits control the the divided value of the pixel clock relative to IPS clock.

表 17-2：像素时钟分频器

DIV_NUM[4:0]	除值
5'b00000	Reserved
5'b00001	Divide-by-2
5'b00010	Divide-by-4
5'b00011	Divide-by-6
5'b00100	Divide-by-8
5'b00101	Divide-by-10
5'b00110	Divide-by-12
5'b00111	Divide-by-14
…..	…..
…..	…..
5'b11101	Divide-by-58
5'b11110	Divide-by-60
5'b11111	Divide-by-62

PPVF — Pixel point valid flag. The change of low to high of this flag means the RGB data is going to transfer the first valid pixel data of a frame. The change of high to low means the last valid pixel data has been finished.

1 = RGB 数据传输正在进行中，显示 RAM 正在繁忙。

0 = 无 RGB 数据通信，显示内存免费，可以更新 d。

TCF — One frame transmission completion flag. TCF flag can be cleared by writing one to this bit.

1 = 一帧传输已完成。

0 = 一帧传输未完成。

TCNF — The specified number of frame transmission completion flag. TCNF flag can be cleared by writing one to this bit. This flag will be set period according to NFTTCR[ITR].

1 = 指定数量的帧传输已完成。

0 = 指定数量的帧传输未完成。

BEPO — Big-endian pixel ordering.

1 = 大端像素排序，第一个像素数据存储在上半字 half-word

0 = 小端像素排序，第一个像素数据存储在下半字 half-word

DRC — Data frame format reverse control.

1 = 数据帧格式：R5、G6、B5

0 = 数据帧格式：B5、G6、R5

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

TCIE — TCF Interrupt Enable.

1 = TCF 中断已启用

0 = TCF 中断已被禁用

TCNIE — TCNF Interrupt Enable.

1 = TCNF 中断启用。

0 = TCNF 中断已被禁用。

RGBE — The RGB Controller Enable Bit. The readable/writeable bit enables RGB Controller operation. When the RGB Controller is disabled, RGB data are in unknow state.

1 = RGB 控制器已启用。

0 = RGB 控制器已被禁用。

提示： 要配置输出信号的极性，必须在配置 RGBE 之前先配置极性位。

17.6. 功能描述

RGB 控制器是 IPS 总线上的一个从属控制器；它直接从内存中获取图形源信息，并在将数据传输到 TFT LCD 面板之前动态地执行 RGB 接口定时。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

18.Blender 控制器（BLDC）

18.1. 功能介绍

LT168 的 Blender 是内部 IPS 总线上的从机；它直接从内存中获取图形源信息，并在将数据传输到 TFT LCD 面板之前动态地执行混合和位混合操作。该 Blender 具有以下功能：

- Alpha 混合与 16 位分辨率
- 内环偏移
- SRAM 的 Blender 时间是取两次，发送一次

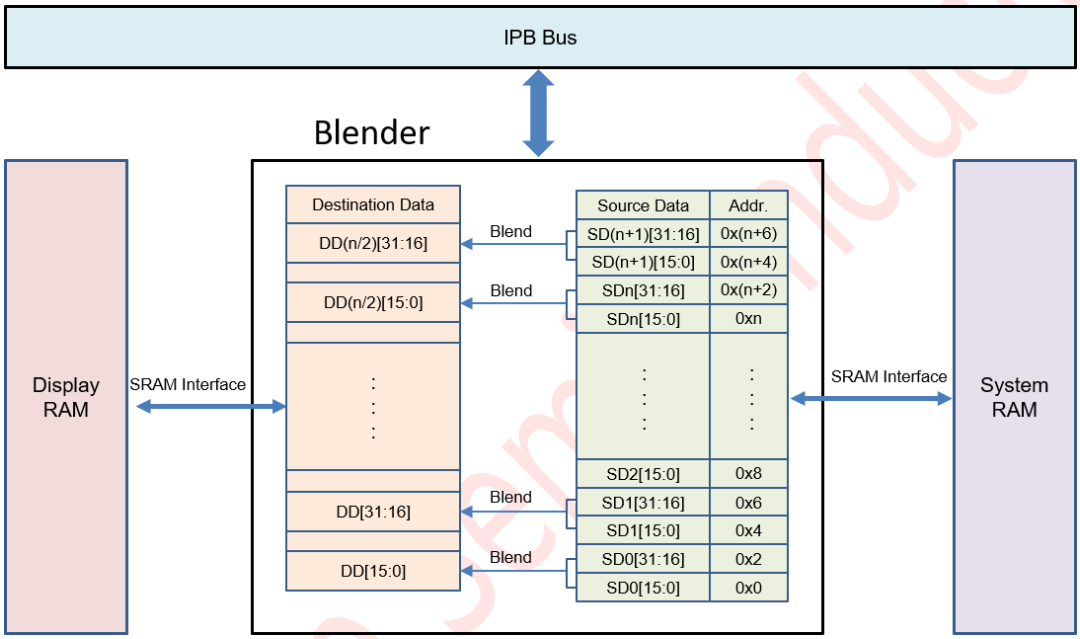


图 18-1：Blender 方块图

©Levetop Semiconductor Co., Ltd.

18.2. 混合算法

表 18-1：源数据格式

bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
R0					G0						B0				
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
α				R1				G1				B1			

表 18-2：目标数据格式

bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
bit15	bit14	bit12	bit12	bit11	bit10	bit9	bit8	bi7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R2					G2						B2				

混合过程的值由以下公式得出：

R2 = (R1<<1)*α + R0*(0xF - α);

G2 = (G1<<2)*α + G0*(0xF - α);

B2 = (B1<<1)*α + B0*(0xF - α);

18.3. 内存映射和寄存器

本节描述了 Blender 的内存映射和寄存器。Blender 模块内存映射如表 18-3 所示，且其基座地址为 0x401B_0000。

18.3.1. 内存映射

参见表 18-3 为 Blender 记忆地图的概述。

表 18-3：Blender 模块内存映射

偏移地址	Bits[31:0]	访问权限
0x0000	Blender Control and Status Register(BCSR)	R/W
0x0004	Source Address Register(SADDR)	R/W
0x0008	Transfer Total Address Length Register (TTALR)	R/W
0x000C	Destination Address Register(DADDR)	R/W
0x0010	Transfer Destination Minor Length and Offset Register(TDMLOR)	R/W

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

18.3.2. 寄存器描述

18.3.2.1. Blender 控制和状态寄存器（BCSR）

地址：Blender_BASEADDR+0x0000_0000

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0					TCF	TOQF	THALFF
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0			BEDM	TCIE	TOQIE	THALFIE	BET
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 18-2：Blender 控制和状态寄存器（BCSR）

TCF — Indicates the transfer of total source data for blending is completed. This flag will be cleared by writing one to it.

1 = 完成混合源数据的传输

0 = 用于混合的源数据的传输没有完成

TOQF — Indicates the transfer of one quarter source data for blending is completed. This flag will be cleared by writing one to it

1 = 完成了四分之一的混合源数据的传输

0 = 完成了一个四分之一的混合源数据的传输

THALFF — Indicates the transfer of one half source data for blending is completed. This flag will be cleared by writing one to it

1 = 完成混合用半源数据的传输

0 = 进行混合的半源数据的传输完成

BEDM — Big-endian data mode for destination address.

1 = 大端数据模式，混合的第一个像素数据存储在上半字

0 = 小端数据模式，混合的第一个像素数据存储在下半字。

©Levetop Semiconductor Co., Ltd.

TCIE — TCF Interrupt Enable

1 = TCF 中断启用

0 = TCF 中断禁用

TOQIE — TOQF Interrupt Enable

1 = TOQF 中断启用

0 = TOQF 中断禁用

THALFIE — THALFF Interrupt Enable

1 = THALFF 中断启用

0 = THALFF 中断禁用

BET — The blender is enabled with a transfer state

1 = 此位被软件设置为进入传输状态，搅拌机在复位此位时可以启动新的数据传输

0 = 当当前传输完成时，硬件清除此位

18.3.2.2. 源地址寄存器（SADDR）

地址：Blender_BASEADDR+0x0000_0004

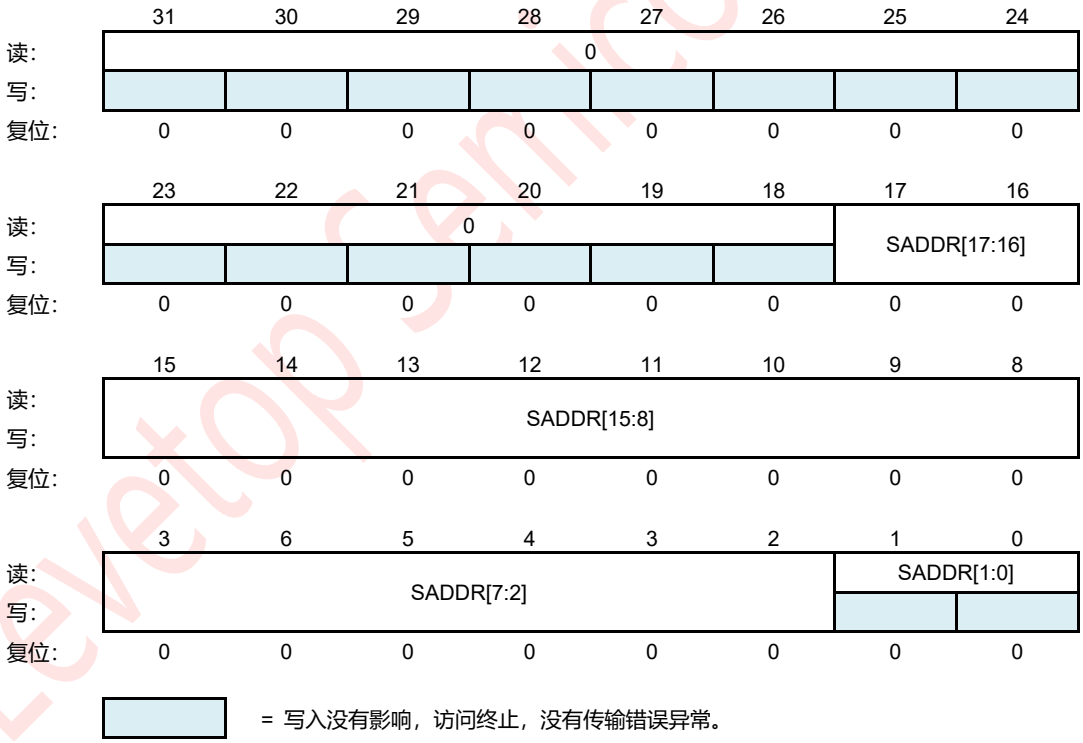


图 18-3：源地址寄存器（SADDR）

SADDR[17:0] — Source offset address in 256K Bytes System RAM. SADDR[1:0] are fixed to 2'b00.

©Levetop Semiconductor Co., Ltd.

18.3.2.3. 传输总地址长度寄存器（TTALR）

地址：Blender_BASEADDR+0x0000_0008

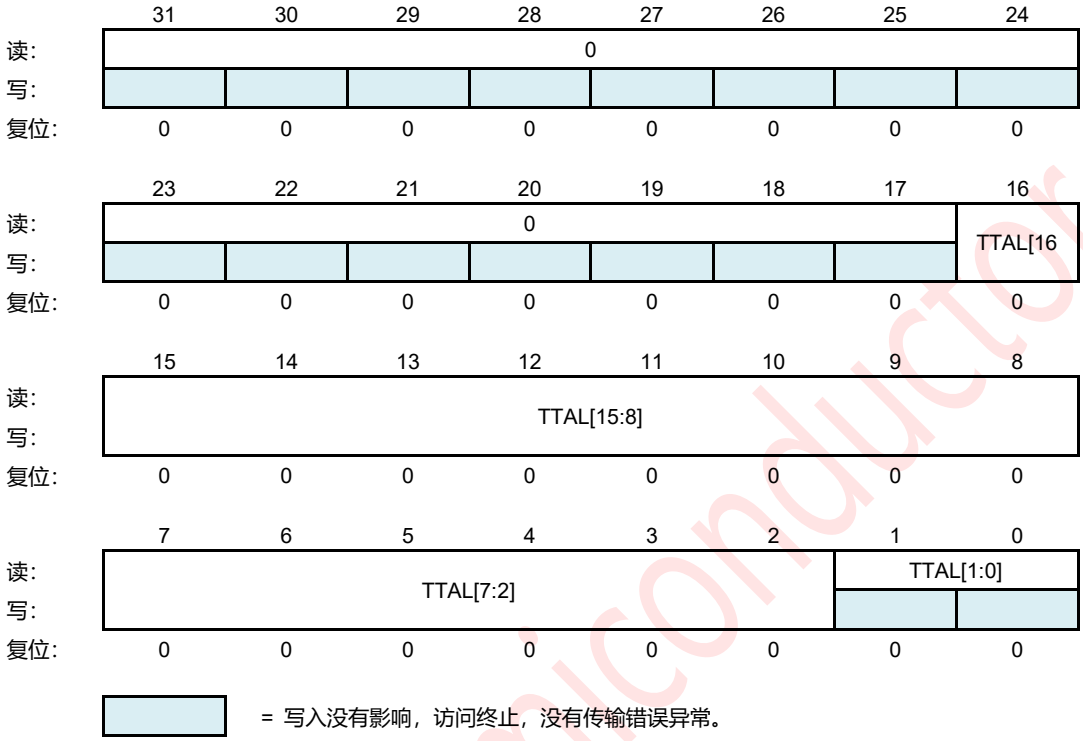


图 18-4：传输总地址长度寄存器（TTALR）

TTAL[16:0] — Source Data Total Transfer Length in word. TTAL[1:0] are fixed to 2’b00.

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

18.3.2.4. 目标地址寄存器（DADDR）

地址：Blender_BASEADDR+0x0000_000C

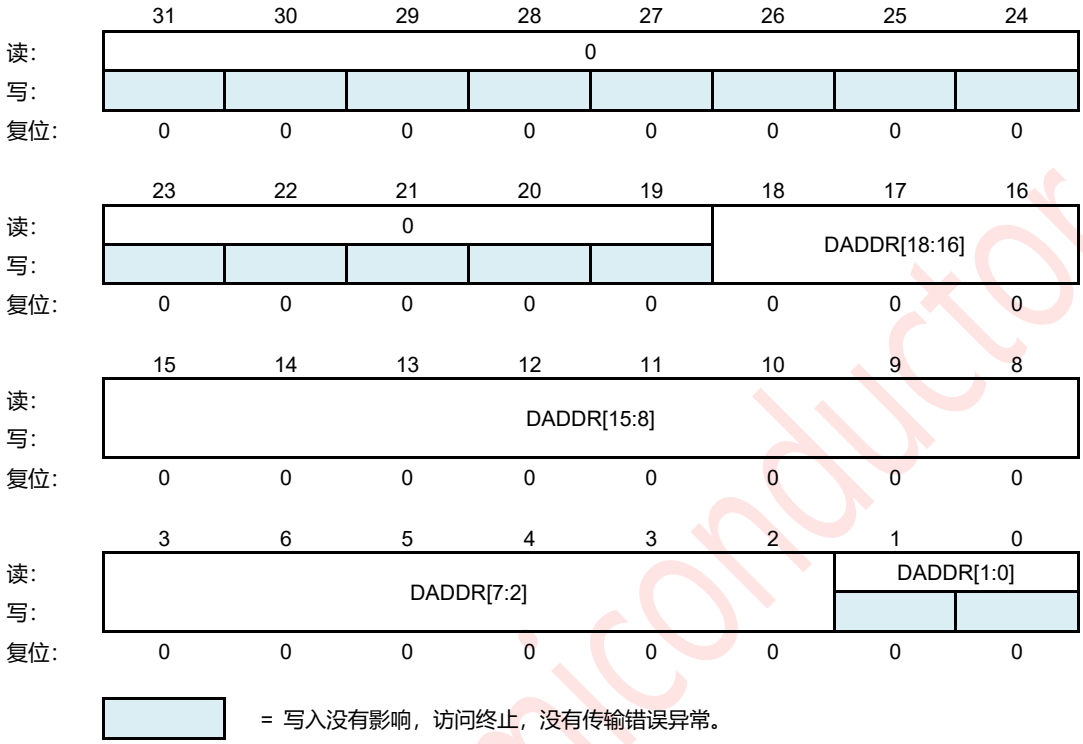


图 18-5：目标地址寄存器（DADDR）

DADDR[18:0] — Destination offset address in 512K Bytes Display RAM. DADDR[1:0] are fixed to 2’b00.

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

18.3.2.5. 传输目的地次要长度和偏移寄存器（TDMLOR）

地址：Blender_BASEADDR+0x0000_0010

	31	30	29	28	27	26	25	24
读: 写:	DMLEN[15:8]							
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读: 写:	DMLEN[7:0]							
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读: 写:	DOFF[15:8]							
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读: 写:	DOFF[7:0]							
复位:	0	0	0	0	0	0	0	0

图 18-6：传输目标次要长度和偏移寄存器（TDMLOR）

DMLEN[15:0] — Destination minor loop length in word

DOFF[15:0] — Destination address signed offset in word. Sign-extended offset is applied to the current destination address to form the next-state value as each destination write is completed.

新的目标偏移地址（当小循环完成时) = DADDR[19:0]+n*（DMLEN[15:0]+DOFF[15:0]），其中 n 代表小循环计数器。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

19.外部总线接口（EBI）

19.1. 功能介绍

LT168 具有一个嵌入式外部总线接口（EBI），用于控制内部总线和外部 8080 MCU 显示面板之间的信息传输。有一个异步芯片选择通道。

19.2. 总线接口特点

外部总线接口的特点包括：

- 降低系统复杂性

  -如果使用芯片选择，典型系统不需要外部粘合逻辑。
- 一个可编程的异步有源低芯片选择。
- 可编程芯片选择等待周期

  -要与各种面板接口，最多 64 个芯片选择断言的周期可以被编程。
- 可编程的读/写断言周期
- 可编程的读/写否定周期

19.3. 信号描述

表 19-1：提供了下面描述的信号的概述。

表 19-1：信号属性

名称	功能	向上拉
EBI_D[15:0]	Data bus	—
EBI_WR#	Write enable and low active	Active
EBI_RD#	Read enable and low active	Active
EBI_RS	Command or Data indication	Active
EBI_CS#	Chip select and low active	Active

19.4. 内存映射和寄存器

19.4.1. 内存映射

EBI 模块内存映射情况见表 19-2，且其基本地址为 0x4018_0000。

表 19-2：显示 EBI 寄存器内存映射。

地址偏移	Bits[31:16]	Bits[15:0]	访问权限 [1,2]
0x0000	CSCR0 — EBI Control Register		S
0x0004	保留		S
0x0008	保留		S
0x000C	IOPCR — IO Pull Control Register		S

©Levetop Semiconductor Co., Ltd.

地址偏移	Bits[31:16]	Bits[15:0]	访问权限 [1,2]
0x0010	IOCR — IO Control Register	保留	S
0x0014	GPIODO — GPIO Data Output Register		S
0x0018	GPIODIR — GPIO Direction Register		S
0x001C	GPIODIN — GPIO Data Input Register		S

提示：

1. 仅限 S = CPU 监控模式访问。
2. 在用户模式下访问仅主管地址位置没有影响，并导致周期终止传输错误。

19.4.2. 寄存器描述

19.4.2.1. EBI 控制寄存器（EBICR）

地址：EBI_BASEADDR+0x0000_0000

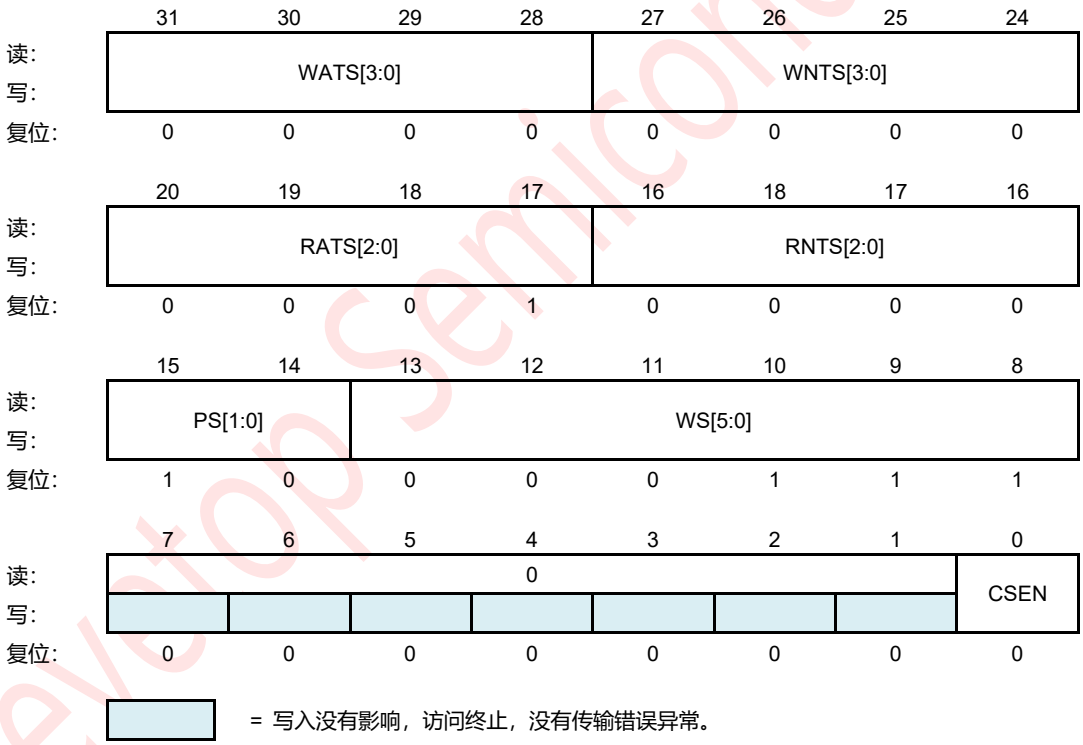


图 19-1：EBI 控制寄存器（EBICR）

WATS[3:0] — Write Signal Assert Timing Select

在写入操作时，这些位选择 EBI_WR#的断言时间。详情请参见功能说明。

WNTS[3:0] — Write Signal Negate Timing Select

当写入操作时，这些位选择了 EBI_WR#的否定时间。详情请参见功能说明。

RATS[3:0] — Read Assert Timing Select

当读取操作时，这些位选择 EBI_RD#的断言时间。详情请参见功能说明。

©Levetop Semiconductor Co., Ltd.

RNTS[3:0] — Read Assert Timing Select

当读取操作时，这些位选择了 EBI_RD#的否定时间。详情请参见功能说明。

PS[1:0] — Port Size Bits

PS 位将芯片选择所支持的外部数据端口的宽度定义为 8 位或 16 位。

00 = 不支持，将导致不可预测的错误

10 = 16 位端口

01 = 8 位端口

11 = 不支持，会导致不可预测的错误

WS[5:0] — Wait States Field

WS 字段确定在断言内部周期终止信号之前，芯片选择逻辑要插入的等待状态数。一个等待状态等于一个系统时钟周期。如果 WS 被配置为零等待状态，则在周期访问开始后的时钟周期中断言内部周期终止信号，从而导致一个时钟传输。配置为一个等待状态的 WS 意味着内部周期终止信号在周期访问开始后断言两个时钟周期。

由于内部循环终止信号是在已编程的等待状态数之后在内部断言的，因此软件可以调整总线定时，以适应外部设备的接入速度。有多达 64 种可能的等待状态，即使是慢的设备也可以与单片机接口。

表 19-3：EBI_CS#芯片选择等待状态编码

WS[5:0]	Number of EBI_CS# Cycles	
	Read Access	Write Access
000000	1	1
000001	2	2
000010	3	3
000011	4	4
000100	5	5
000101	6	6
000110	7	7
000111	8	8
001000	9	9
001001	10	10
001010	11	11
001011	12	12
001100	13	13
001101	14	14
.....	.....	.....
111111	64	64

©Levetop Semiconductor Co., Ltd.

CSEN — Chip Select Enable Bit

CSEN 位启用了芯片选择逻辑。当芯片选择功能禁用时，EBI_CS#信号负高。

1 = 芯片选择功能启用。

0 = 芯片选择功能被禁用。

19.4.2.2. IO 拉力控制寄存器（IOPCR）

地址：EBI_BASEADDR+0x0000_000C

	31	30	29	28	27	26	25	24
读:	PULLUP15:8]							
写:								
复位:	1	1	1	1	1	1	1	1

	19	22	21	20	19	18	17	16
读:	PULLUP[7:0]							
写:								
复位:	1	1	1	1	1	1	1	1

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 19-2：IO 拉力控制寄存器（IOPCR）

PULLUP[15:0] — EBI_D[15:0] Pull up enable control bits when used as GPIO input mode.

1 = EBI_D[15:0]端口拉已启用

0 = EBI_D[15:0]端口拉拔已被禁用

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

19.4.2.3. IO 控制寄存器（IOCR）

地址：EBI_BASEADDR+0x0000_0010

	31	30	29	28	27	26	25	24
读:	0							EBI_LIT_ENDIAN
写:								
复位:	0	0	0	0	0	0	0	1

	23	22	21	20	19	18	17	16
读:	0				D3_GP3	D2_GP2	CTL_GP	0
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 19-3：IO 控制寄存器（IOCR）

EBI_LIT_ENDIAN — EBI ports is little endian

1 = EBI 端口不小

0 = EBI 端口是大的

D3_GP3 — D[15:8] port used as GPIO

写入这一位决定了 D 端口的使用。

1 = D[15:8]端口被用作 GPIO

0 = D[15:8]端口作为显示面板的数据总线

D2_GP2 — EBI_D[7:0] port used as GPIO

写入此位决定了 EBI_D[7:0]端口的使用。

1 = EBI_D[7:0]端口被用作 GPIO

0 = EBI_D[7:0]端口用作显示面板的数据总线

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

CTL_GP — EBI_WR#, EBI_RD#, EBI_CS#, EBI_RS pin as EPORT1 function

表 19-4：EBI 控制引脚为 EPORT1

CTL_GP = 1'b0	CTL_GP = 1'b1
EBI_CS#	INT1[5]
EBI_RD#	INT1[4]
EBI_WR#	INT1[2]
EBI_RS	INT1[1]

19.4.2.4. GPIO 数据输出寄存器（GPIODO）

地址：EBI_BASEADDR+0x0000_0014

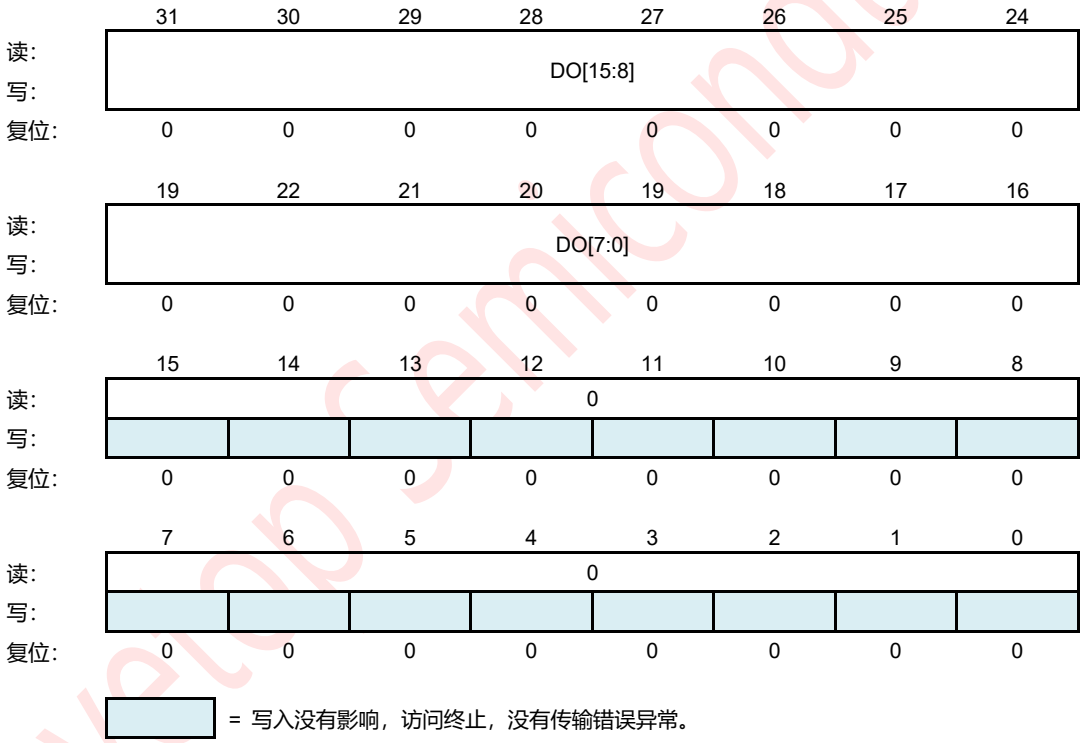


图 19-4：GPIO 数据输出寄存器（GPIODO）

DO[15:0] — Output data of EBI_D[15:0] pin

写入这些位决定了作为 GPIO 使用时 EBI_D[15:0] pin 的输出数据。

©Levetop Semiconductor Co., Ltd.

19.4.2.5. GPIO 方向寄存器（直径）

地址：EBI_BASEADDR+0x0000_0018

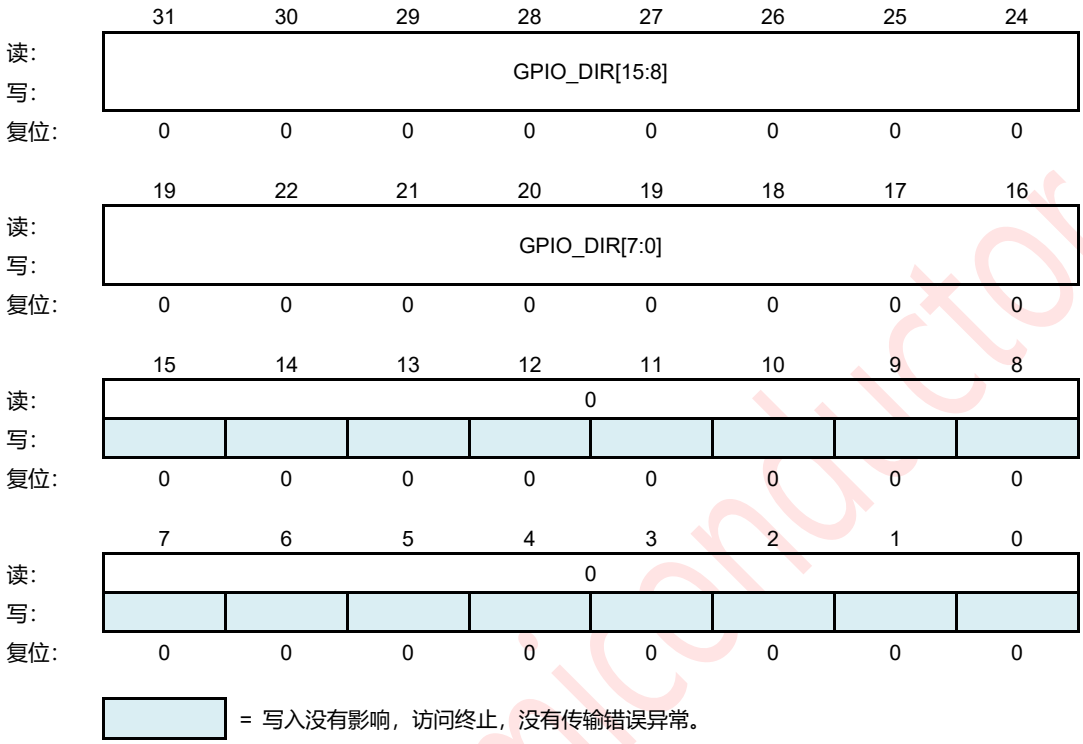


图 19-5：GPIO 方向寄存器（图形）

GPIO_DIR[15:0] — GPIO Pin's Direction

写入这些位 s 决定了 GPIO 引脚的方向。

1 = GPIO 是输出

0 = GPIO 是输入

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

19.4.2.6. GPIO 数据输入寄存器（GPIODI）

地址：EBI_BASEADDR+0x0000_001C

	31	30	29	28	27	26	25	24
读:	GPIO_DI[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	19	22	21	20	19	18	17	16
读:	GPIO_DI[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 19-6：GPIO 数据输入寄存器（GPIODI）

GPIO_DI[15:0] — GPIO's Input Data

读取这些位反映了 EBI_D[15:0]引脚作为 GPIO 使用时的状态。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

19.5. 功能描述

19.5.1. EBI_WR#、EBI_RD#信号定时

对于写和读操作，可以配置 EBI_WR#和 EBI_RD#断言和否定定时。

表 19-5：EBI_WR#和 EBI_RD# Assert and Negate 时间

RATS[3:0]/WATS[3:0]	Assert Time (T_{ass})	RNTS[3:0]/WNTS[3:0]	Negate Time (T_{neg})
0000	1/2	0000	1/2
0001	2/2	0001	2/2
0010	3/2	0010	3/2
0011	4/2	0011	4/2
0100	5/2	0100	5/2
.....	.....	.....	.....
1110	15/2	1110	15/2
1111	16/2	1111	16/2

单位：系统周期

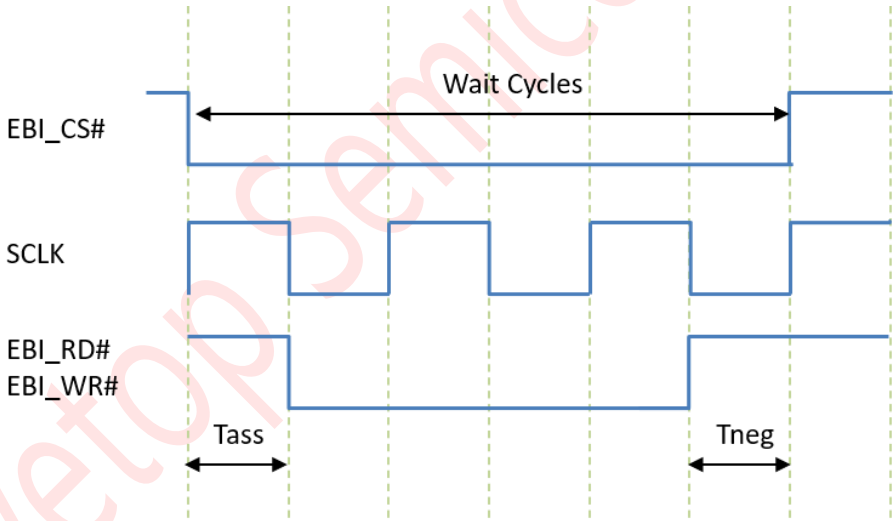


图 19-7：EBI_WR#, EBI_RD# Assert/Negate Timing

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

19.5.2. EBI 功能描述

19.5.2.1. EBI_CS#芯片选择

芯片选择可为单片机面板提供芯片启用信号，并断言内部总线周期终止信号。

在 EBICR 中设置 CSEN 位可使芯片选择提供面板芯片启用信号。

活动芯片选择的配置由等待状态（WS）字段决定，端口大小（PS）字段用于访问。

表 19-6：芯片选择地址范围编码

芯片选择	块大小	正常模式地址范围
EBI_CS#	1M 字节	0x2000_0000-0x2FFF_FFFF

19.5.2.2. 操作转输

内部 AHB 总线可能的操作数访问包括：

- 字节
- 对齐的上半字
- 对齐的下半字
- 对齐的单词

不支持任何不一致的传输。EBI 控制在 AHB 总线和一个 8 位或 16 位 Port 之间的字节、半字或字操作数传输。“Port”是指外部设备在数据传输期间使用的数据路径的宽度。

每个端口都被分配给数据总线的特定位。对于每个芯片选择，一个 8 位端口分配给引脚 D[31:24]，一个 16 位端口分配给引脚 D[31:16]。

表 19-7 到**表 19-8** 展示 EBI_CS#的可能的传输尺寸、对齐方式和端口号。表中显示的数据字节表示外部数据引脚。这些数据被多路复用，并被驱动到外部数据总线，如图所示。不需要带破折号的字节；总线将在读传输时忽略它们，并在写传输时使用未定义的数据驱动它们。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

表 19-7：EBI_CS#8 位端口数据传输（大端）

Transfer Size	Port Width	Internal Addr		External Pins	Data Bus Transfer			
		Bit1	Bit0	EBI_RS				
Byte	8	0	0	0	D[15:8]	--	--	--
		0	1	1	--	D[15:8]	--	--
		1	0	0	--	--	D[15:8]	--
		1	1	1	--	--	--	D[15:8]
Half-word	8	0	0	0	D[15:8]	--	--	--
				1	--	D[15:8]	--	--
		1	0	0	--	--	D[15:8]	--
				1	--	--	--	D[15:8]
Word	8	0	0	0	D[15:8]	--	--	--
				1	--	D[15:8]	--	--
				0	--	--	D[15:8]	--
				1	--	--	--	D[15:8]

表 19-8：EBI_CS#16 位端口数据传输（大端）

Transfer Size	Port Width	Internal Addr		External Pins	Data Bus Transfer			
		Bit1	Bit0	EBI_RS				
Byte	16	0	0	0	D[15:8]	--	--	--
		0	1	0	--	D[7:0]	--	--
		1	0	1	--	--	D[15:8]	--
		1	1	1	--	--	--	D[7:0]
Half-word	16	0	0	0	D[15:8]	D[7:0]	--	--
		1	0	1	--	--	D[15:8]	D[7:0]
Word	16[1]	0	0	0	D[15:8]	D[7:0]	--	--
				1	--	--	D[15:8]	D[7:0]

提示：

1.这个 EBI 对 16 位端口的单词访问运行两个周期。该表显示了两个总线周期的数据位置。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

20.可编程中断定时器（PIT）

20.1. 功能介绍

LT168 的可编程中断计时器（PIT）是一个 16 位计时器，提供精确的中断在定期间隔与最小的处理器干预。计时器可以从模量锁存器中写入的值开始计数，也可以是一个自由运行的下行计数器。

20.2. 方框方块图

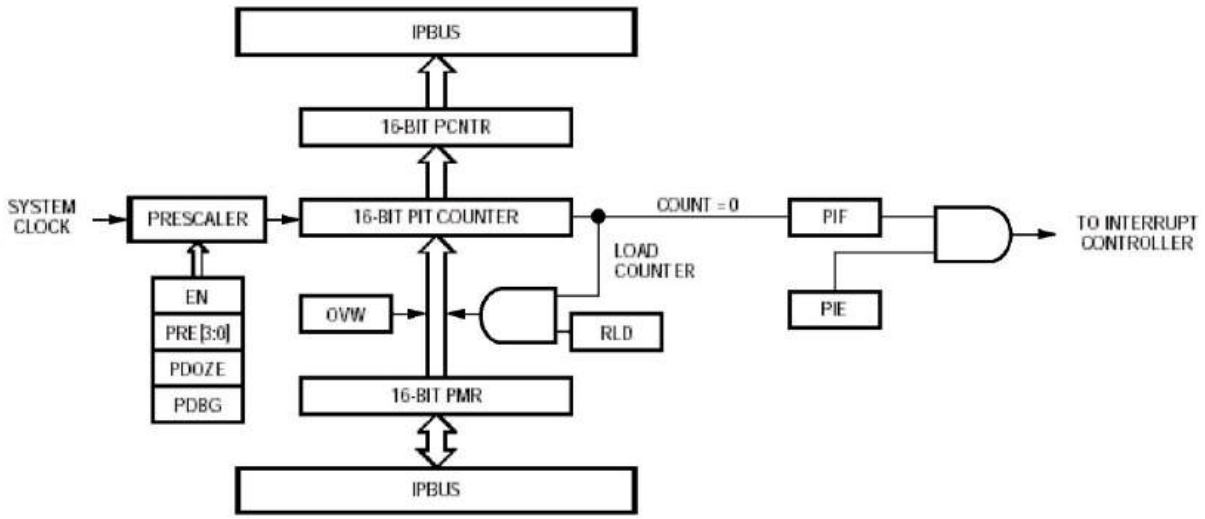


图 20-1：PIT 方块图

20.3. 操作模式

本节介绍了三种低功耗模式和调试模式。

20.3.1. 等待模式

在等待模式下，PIT 模块继续正常运行，并且可以通过产生一个中断请求来配置为退出低功耗模式。

20.3.2. Doze 模式

在 PIT 控制和状态寄存器（PCSR）中设置 PDOZE 位的启动模式下，PIT 模块操作停止。在 PDOZE 位清除的睡觉模式下，睡觉模式不影响 PIT 操作。退出休眠模式后，PIT 操作从进入休眠模式之前的状态继续。

20.3.3. 停止模式

在停止模式下，系统时钟缺失，PIT 模块操作停止。

20.3.4. 调试模块

在 PCSR 模式中设置 PDBG 位的调试模式下，PIT 模块操作停止。在清除 PDBG 位的调试模式下，调试模式不会影响 PIT 操作。退出调试模式后，PIT 操作将从进入调试模式之前的状态继续进行，但在调试模式下进行的任何更新都将保留。

©Levetop Semiconductor Co., Ltd.

20.4. 信号说明

PIT 模块没有芯片外信号。

20.5. 内存映射和寄存器

本节描述了 PIT 的内存映射和寄存器结构。PIT 模块内存映射如表 20-1 所示.PIT0 的基本地址为 0x4004_0000，PIT1 为 0x4005_0000，PIT2 为 0x4006_0000，PIT3 为 0x4007_0000。

20.5.1. 内存映射

见表 20-1 以获取内存映射的描述。

该设备有四个可编程的中断计时器。

表 20-1：可编程中断定时器模块内存映射

PITn 地址	Bits[15:0]	访问权限(1)
0x0	PIT Modulus Register (PMR)	S
0x2	PIT Control and Status Register (PCSR)	S
0x4	Unimplemented(2)	—
0x6	PIT Count Register (PCNTR)	S/U

提示：

(1) 仅限 S = CPU 监控模式访问。S/U = CPU 监控器或用户模式访问。在用户模式下只访问管理员地址没有影响，并导致周期终止传输错误。

(2) 对未实现的地址位置的访问没有影响，并会导致周期终止传输错误。

20.5.2. 寄存器描述

PIT 编程模型由以下寄存器组成：

- PIT 控制和状态寄存器（PCSR）配置计时器的操作。参见章节 20.5.2.2。
- PIT 模量寄存器（PMR）确定定时器模量重新加载值。参见章节 20.5.2.1。
- PIT 计数寄存器（PCNTR）提供了对计数器值的可见性。参见章节 20.5.2.3。

©Levetop Semiconductor Co., Ltd.

20.5.2.1. PIT 模量寄存器（PMR）

16 位读/写 PIT 模量寄存器（PMR）包含当计数达到 0x0000 和 RLD 位被设置时加载到 PIT 计数器的计时器模量值。

当设置了 OVW 位时，PMR 是透明的，并且写入 PMR 的值会立即加载到 PIT 计数器中。每当新值加载到 PIT 计数器以及复位计数器期间，都会复位预调节计数器。读取 PMR 将返回用模量锁存器写入的值。将 PMR 初始化为 0xFFFF。

地址：PITn_BASEADDR+0x0000_0000

	15	14	13	12	11	10	9	8
读: 写:	PM[15:8]							
复位:	1	1	1	1	1	1	1	1

	3	6	5	4	3	2	1	0
读: 写:	PM[7:0]							
复位:	1	1	1	1	1	1	1	1

图 20-2：PIT Modulus 寄存器（PMR）

20.5.2.2. PIT 控制和状态寄存器（PCSR）

地址：PITn_BASEADDR+0x0000_0002

	15	14	13	12	11	10	9	8
读:	0				PRE[3:0]			
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读: 写:	PDMAE	PDOZE	PDBG	OVW	PIE	PIF	RLD	EN
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 20-3：PIT 控制和状态寄存器（PCSR）

PRE[3:0] — Prescaler Bits

读写准备位选择系统时钟除数来生成 PIT 时钟，如表 20-2 所示。

要准确预测下一次计数的时间，仅在清除启用位（EN）时才更改 PRE[3:0]位。更改 PRE，将复位前置调节器计数器。系统复位和将新值加载到计数器也会复位预压器计数器。将 EN 位并写入设置为 PRE[3:0]可以在相同的写入周期中完成。清除 EN 位停止预调器计数器。

©Levetop Semiconductor Co., Ltd.

表 20-2：Prescaler Select Encoding

PRE[3:0]	IPS Clock Divisor
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	512
1010	1,024
1011	2,048
1100	4,096
1101	8,192
1110	16,384
1111	32,768

PDMAE — DMA Enable Control Bit

读/写 PDMAE 位控制是指当 PIT 计数器达到 0x0000 时是否会生成 DMA 请求。

当 PIT 计数器达到 0x0000 时，将生成 1 = DMA 请求。

当 PIT 计数器达到 0x0000 时，将不会生成 0 = DMA 请求。

PDOZE — Doze Mode Bit

读/写 PDOZE 位控制休眠模式下的功能。复位清除 PDOZE。

1 = PIT 功能在休眠模式下停止

0 =PIT 功能休眠模式下不受影响

当退出休眠模式时，计时器操作从进入休眠模式之前的状态继续。

PDBG — Debug Mode Bit

读/写 PDBG 位在调试模式下控制 PIT 的功能。复位清除 PDBG。

1 = 在调试模式下，PIT 功能停止

0 = 在调试模式下，PIT 功能不受影响

在调试模式下，寄存器的读写访问功能正常。退出调试模式后，计时器操作将从进入调试模式之前的状态继续进行，但在调试模式下进行的任何更新都将保留。

提示： 在调试模式中将 PDBG 位从 1 改为 0 将启动 PIT 计时器。同样，在调试模式期间将 PDBG 位从 0 更改为 1 也会停止 PIT 计时器。

©Levetop Semiconductor Co., Ltd.

OVW — Overwrite Bit

读/写 OVW 位允许写入 PMR 以立即覆盖 PIT 计数器中的值。

1 = 写入 PMR 立即替换 PIT 计数器中的值。

0 = 当计数达到 0x0000 时，PMR 中的值替换 PIT 计数器中的值。

PIE — PIT Interrupt Enable Bit

读/写 PIE 位允许 PIF 标志生成中断请求。

1 = PIF 中断请求启用

0 = PIF 中断请求禁用

PIF — PIT Interrupt Flag

当 PIT 计数器达到 0x0000 时，将设置读/写 PIF 标志。通过写入 PIF 或写入 PMR 或 DMA 请求来清除 PIF。写入 0 没有效果。复位清除 PIF。

1 = PIT 计数已达到 0x0000。

0 = PIT 计数未达到 0x0000。

RLD — Reload Bit

读/写 RLD 位允许在计数达到 0x0000 时将 PMR 的值加载到 PIT 计数器中。

1 = 计数器在计算数为 0x0000 时从 PMR 重新加载

0 = 计数器在计数为 0x0000 时卷转到 0x FFFF

EN — PIT Enable Bit

读写 EN 位启用 PIT 操作。禁用该 PIT 时，计数器和预调节器将处于停止状态。

1 = PIT 启用

0 = PIT 禁用

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

20.5.2.3. PIT 计数寄存器（PCNTR）

16 位，只读的 PIT 控制寄存器（PCNTR）包含计数器值。

用两个 8 位读取读取 16 位计数器并不能保证是一致的。

写入 PCNTR 没有效果，写入周期正常终止。

地址：PITn_BASEADDR+0x0000_0006

	15	14	13	12	11	10	9	8
读:	PC[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	PC[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 20-4：PIT 计数寄存器（PCNTR）

20.6. 功能描述

本节介绍了 PIT 的功能操作。

20.6.1. 设置和忘记计时器操作

当设置了 PCSR 寄存器中的 RLD 位时，将选择此操作模式。

当 PIT 计数器达到计数数 0x0000 时，在 PCSR 中设置 PIF 标志。模量锁存器中的值被加载到计数器中，计数器开始向 0x0000 递减。如果 PIE 位设置为 PCSR 位，则 PIF 标志会向 CPU 发出中断请求。

当 OVW 位设置在 PCSR 中时，计数器可以通过写入 PMR 来直接初始化，而不必等待计数达到 0x0000。

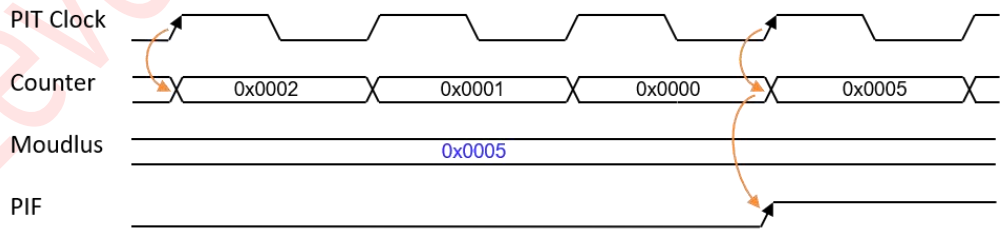


图 20-5：计数器从 Modulus Latch 重新加载

©Levetop Semiconductor Co., Ltd.

20.6.2. 自由运行计时器操作

当清除了 PCSR 中的 RLD 位时，将选择此操作模式。在此模式下，计数器从 0x0000 滚动到 0x FFFF，而不需要从模量锁存器重新加载，并继续衰减。

当计数器达到计数数 0x0000 时，在 PCSR 中设置 PIF 标志。如果 PIE 位设置为 PCSR 位，则 PIF 标志会向 CPU 发出中断请求。

当 OVW 位设置在 PCSR 中时，计数器可以通过写入 PMR 来直接初始化，而不必等待计数达到 0x0000。

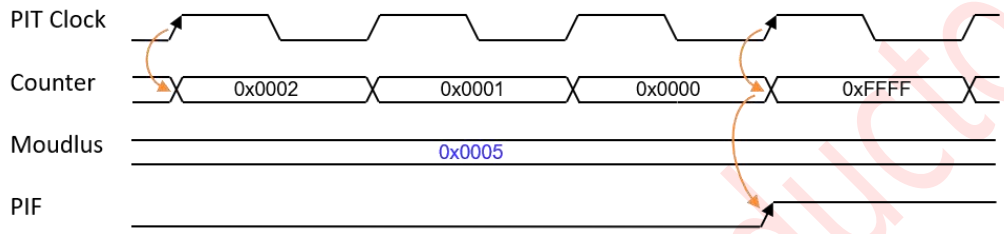


图 20-6：在自由运行模式下的计数器

20.6.3. 超时规格

16 位的 PIT 计数器和预调用器支持不同的超时时间。预计算器按 PCSR 中的 PRE[3:0]位的选择来划分系统时钟。PMR 中的 PM[15:0]位将选择超时时间。

timeout period = PRE[3:0] x (PM[15:0] + 1) clocks

20.7. 中断操作

表 20-3 列出了由 PIT 生成的中断请求。

表 20-3：PIT 中断请求

中断请求	标志	启用位
Timeout	PIF	PIE

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

21.看门狗计时器（WDT）

21.1. 功能介绍

LT168 的监视计时器是一个 16 位计时器，用于帮助软件从失控代码恢复或在操作运行比预期更长时给予中断。看门狗计时器有一个自由运行的下行计数器（看门狗计数器），产生一个复位或中断的下流。为了防止复位，软件必须定期通过维修看门狗来重新启动倒计时。

21.2. 操作模式

本节介绍了看门狗定时器在低功率模式和调试模式下的操作。

21.2.1. 等待模式

在看门狗控制寄存器（WCR）中设置了等待位的等待模式下，看门狗计时器操作停止。在等待位清除的等待模式下，看门狗计时器继续正常工作。

21.2.2. Doze 模式

在 WCR 中设置 doze 位的休眠模式下，看门狗计时器模块的操作停止。在睡觉模式下，看门狗计时器继续正常工作。

21.2.3. 停止模式

在停止模式下，将停止位设置为 WCR，看门狗操作在停止模式下停止。当停止模式退出时，看门狗操作从它在进入停止模式之前的状态继续操作。在停止模式与停止位清除，看门狗计时器继续正常工作。

21.2.4. 调试模式

在 WCR 中设置 DBG 位的调试模式下，看门狗计时器模块的操作停止。在清除的 DBG 位的调试模式下，看门狗计时器继续正常运行。当调试模式退出时，看门狗计时器操作将从进入调试模式之前的状态继续进行，但在调试模式下进行的任何更新都将保留。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

21.3. 方块图

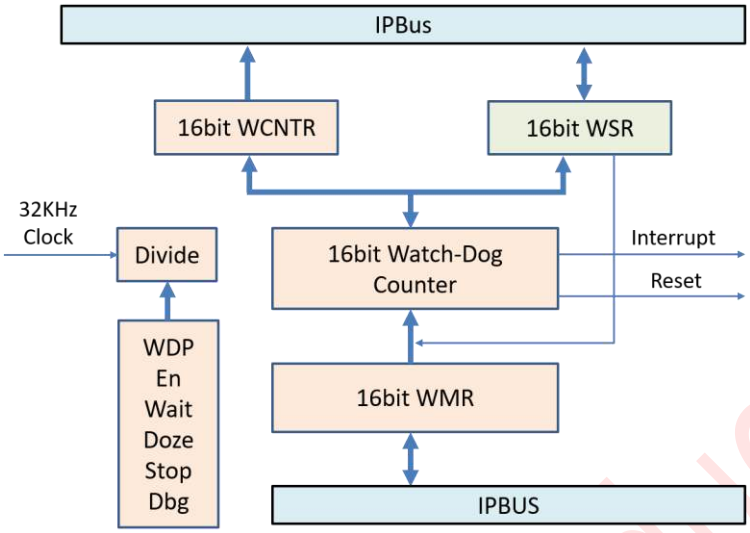


图 21-1：看门狗计时器方块图

21.4. 信号说明

看门器定时器模块没有芯片外信号。

21.5. 内存映射和寄存器

本节描述了看门狗计时器的内存映射和寄存器。WDT 模块的基本地址为 0x4013_0000。

21.5.1. 内存映射

参见表 21-1 为看门狗记忆地图的概述。

表 21-1：看门狗计时器模块记忆地图

偏移地址	Bits[15:0]	访问权限(1)
0x0000	Watchdog Modulus Register (WMR)	S
0x0002	Watchdog Control Register (WCR)	S
0x0004	Watchdog Service Register (WSR)	S/U
0x0006	Watchdog Count Register (WCNTR)	S/U

提示(1) :仅限 S = CPU 监控模式访问。S/U = CPU 监控或用户模式访问。在用户模式下只访问管理员地址没有影响，并导致周期终止传输错误。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

21.5.2. 寄存器说明

看门狗计时器编程模型由以下几种寄存器组成：

- 看门狗控制寄存器（WCR）配置看门狗计时器操作。参见章节 21.5.2.2.
- 看门器模量寄存器（WMR）确定定时器模量重新加载值。参见章节 21.5.2.1 錯誤! 找不到參照來源。.
- 监视器计数寄存器（WCNTR）提供了对监视器计数器值的可见性。参见章节 21.5.2.4.
- 看门狗服务寄存器（WSR）需要一个服务序列，以防止复位。只读的 WC 字段反映了电流看门狗计数器中的值。读取 16 位 WCNTR8 位的读取不能保证返回一个相干的值。写入到 WCNTR 没有作用，写入周期正常终止。这个寄存器是为看门狗工作域，因此读取值可能不是稳定的，请连续不断地阅读。.

21.5.2.1. 看门狗模块寄存器（WMR）

地址：WDT_BASEADDR+0x0000_0000

	15	14	13	12	11	10	9	8
读: 写:	WM[15:8]							
复位:	1	1	1	1	1	1	1	1

	7	6	5	4	3	2	1	0
读: 写:	WM[7:0]							
复位:	1	1	1	1	1	1	1	1

图 21-2：看门狗模量寄存器（WMR）

WM[15:0] — Watchdog Modulus Field

WM 字段包含由服务序列重新加载到看门狗计数器中的模量。写入 WMR 后，会立即将新的模量值加载到看门狗计数器中。该新值也将在下一次和所有后续的重新加载时使用。

读取 WMR 返回模量寄存器中的值。将 WM[15:0]字段初始化为 0x FFFF。

©Levetop Semiconductor Co., Ltd.

21.5.2.2. 看门狗控制寄存器（WCR）

16 位读/写看门狗控制寄存器（WCR）配置看门狗计时器操作。

地址：WDT_BASEADDR+0x0000_0002

	15	14	13	12	11	10	9	8
读:	0				WAIT	DOZE	STOP	DBG
写:								
复位:	0	0	0	0	1	1	1	1

	7	6	5	4	3	2	1	0
读:	IS	WDP[2:0]			IF	IE	0	EN
写:							CU	
复位:	0	1	1	1	0	1	0	1

= 写入没有影响，访问终止，没有传输错误异常。

图 21-3：看门狗控制寄存器（WCR）

WAIT — Wait Mode Bit

等待位控制在等待模式下的看门狗计时器的功能。复位设置等待。

1 = 看门狗计时器在等待模式下停止

0 = 看门狗计时器在等待模式下不受影响

DOZE — Doze Mode Bit

DOZE 位控制起睡模式下看门狗定时器的功能。复位设置大小。

1 = 看门狗计时器在休眠模式下停止

0 = 手表计时器在休眠模式下不受影响

STOP — STOP Mode Bit

停止位控制在停止模式下的看门狗计时器的功能。复位设置停止。

1 = 看门狗计时器在停止模式下已停止

0 = 看门狗计时器在停止模式下不受影响

DBG — Debug Mode Bit

DBG 位在调试模式下控制看门器计时器的功能。在调试模式下，可以正常地写入和读取看门狗定时器寄存器。退出调试模式后，计时器操作将从进入调试模式之前的状态继续进行，但在调试模式下进行的任何更新都将保留。

1 = 监视器计时器在调试模式下停止

0 = 看门狗计时器在调试模式下不受影响

提示：

在调试模式下，将 DBG 位从 1 更改为 0 将启动看门狗计时器。

在调试模式下，将 DBG 位从 0 更改为 1 将停止看门狗计时器。

©Levetop Semiconductor Co., Ltd.

IS — Watchdog Clock Domain Interrupt Status Bit

这个位是只读的，如果这个位是 1'b1，则不清除看门狗时钟域的状态，因此如果 CPU 想要休眠或停止，它将再次被唤醒。因此，在 CPU 想要休眠或停止之前，请先检查这个位。

WDP[2:0] — Watchdog Timer Prescaler

WDP[2:0]位决定了看门狗计时器运行时的看门狗计时器预压。不同的预调整值及其相应的超时时间见表 21-2 看门狗计时器。

表 21-2：看门狗计时器 Prescaler

WDP[2:0]	Prescaler
000	128 KHz/2048
001	128 KHz/1024
010	128 KHz/512
011	128 KHz/256
100	128 KHz/128
101	128 KHz/64
110	128 KHz/32
111	128 KHz/16

IF — Watchdog Interrupt Flag Bit

写一个到这一点将清除的标志。

IE — Watchdog Interrupt Enable Bit

IE 位启用看门狗计时器中断模式。一旦产生了中断，并且 EN 位为 1，该位将被自动清除。

1 = 看门狗定时器中断模式已启用

0 = 看门狗计时器中断模式被禁用

CU — Watchdog Change Update Bit

写一个 CU 位将把 WDP[2:0]和 WMR 更新到工作锁存器。

EN — Watchdog Enable Bit

EN 位启用了看门狗计时器。

1 = 看门狗计时器已启用

0 = 看门狗计时器被禁用

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

21.5.2.3. 看门狗服务寄存器（WSR）

当看门狗计时器启用时，在看门狗计数器超时之前，将 0x5555 和 0x AAAA 写入看门器服务寄存器（WSR），以防止复位。如果 WSR 在超时前没有得到维修，看门狗计时器会向复位控制器或中断控制器模块发送一个信号，并断言一个系统复位或中断。

两个写入操作必须按照超时前列出的顺序发生，但是在两个写入操作之间可以执行任意数量的指令。但是，将除 0x5555 或 0xAAAA 以外的任何值写入 WSR 将复位服务序列，需要写入两个值以防止看门狗计时器导致复位。

地址：WDT_BASEADDR+0x0000_0004

	15	14	13	12	11	10	9	8
读: 写:	WS[15:8]							
复位:	0	0	0	0	0	0	0	0

	4	3	5	4	3	2	1	0
读: 写:	WS[7:0]							
复位:	0	0	0	0	0	0	0	0

图 21-4：看门狗服务寄存器（WSR）

21.5.2.4. 监视器计数寄存器（WCNTR）

地址：WDT_BASEADDR+0x0000_0006

	15	14	13	12	11	10	9	8
读:	wc[15:8]							
写:								
复位:	1	1	1	1	1	1	1	1

	7	6	5	4	3	2	1	0
读:	WC[7:0]							
写:								
复位:	1	1	1	1	1	1	1	1

= 写入没有影响，访问终止，没有传输错误异常。

图 21-5：看门机构统计寄存器（WCNTR）

WC[15:0] — Watchdog Count Field
只读 WC[15:0]字段反映了看门狗计数器中的当前值。用两个 8 位读取读取 16 位 WCNTR 并不能保证返回一个相干值。写入 WCNTR 没有效果，写入周期正常终止。此寄存器为看门狗工作域，读取值可能不稳定，请连续读取。

©Levetop Semiconductor Co., Ltd.

22.时钟控制器（RTC）

22.1. 功能介绍

LT168 的实时控制器控制着一个具有 RTC 功能的综合 PMU。它可以重新配置 RTC 计数器，设置报警，并产生时间/报警中断。

22.2. RTC 特点

本模块的主要特点是：

- 重新配置 RTC 计数器，并读取来自秒、分钟、小时和天的计数器
- 支持报警设置
- 中断源：秒、分钟、小时、日中断、可编程报警中断、1 KHz/32 KHz 周期性中断。

22.3. 测试模式

在测试模式下，RTC 可以通过 CPU 或芯片针脚进行配置，RTC 的状态和时钟信号可以通过芯片针脚进行检查。

22.4. 方块图

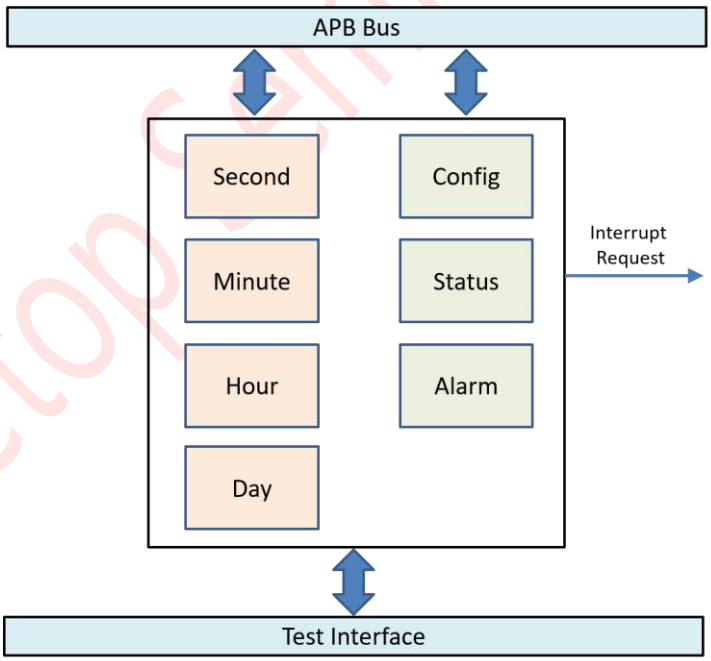


图 22-1：RTC 方块图

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

22.5. 应用电路

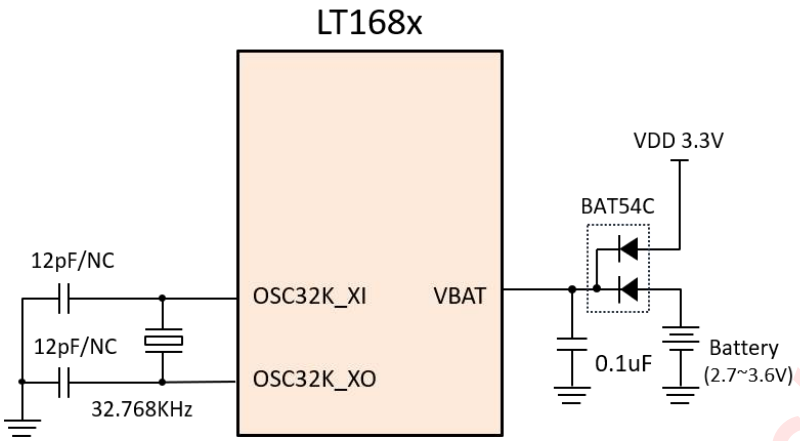


图 22-2：RTC 应用电路

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

23.边缘端口模块（EPORT）

23.1. 功能介绍

LT168 有三个边缘端口模块（EPORT）。每个 EPORT 都有 8 个相应的外部中断引脚。每个引脚可以单独配置为低电平敏感中断引脚、边缘检测中断引脚（上升边、下降边或两者）或通用输入/输出/（I/O）引脚。见图 23-1。

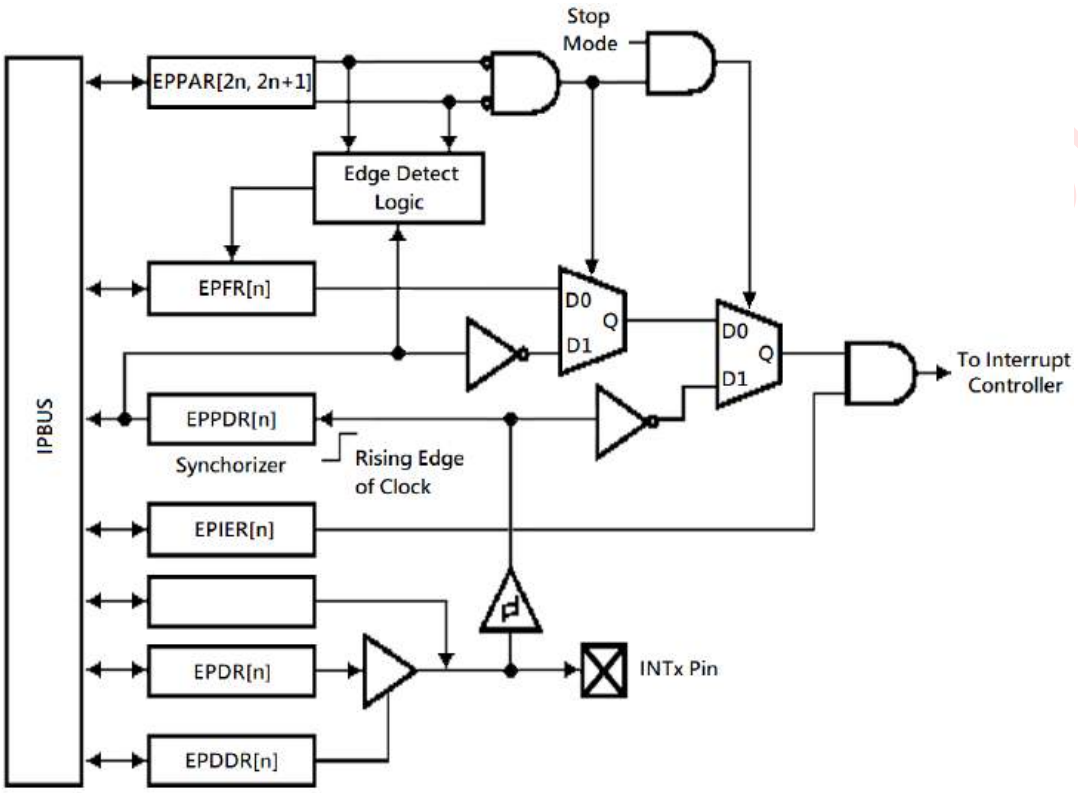


图 23-1：EPORT 方块图

23.2. 低功耗模式运行

本节介绍了 EPORT 模块在低功耗模式下的操作情况。

23.2.1. 等待和多兹模式

在等待和睡眠模式下，EPORT 模块继续正常运行，并且可以被配置为通过在选定的边缘或在外部引脚上的低电平上产生中断请求来退出低功耗模式。

23.2.2. 停止模式

在停止模式下，没有可用来执行边缘检测功能。只有级别检测逻辑是激活的（如果配置），以允许外部中断插脚上的任何低级别生成中断（如果启用）以退出停止模式。

提示：由于没有时钟，电平检测逻辑绕过输入引脚同步器。

©Levetop Semiconductor Co., Ltd.

23.3. 中断/通用输入引脚说明

所有复位引脚时默认为通用输入引脚。当从 EPORT 引脚数据寄存器(EPPDR)读取时，引脚值被同步到 CLKOUT 的上升边缘。在边缘/级别检测逻辑中使用的值也被同步到 CLKOUT 的上升边缘。这些引脚使用施密特触发输入缓冲区，该缓冲区内置迟滞，旨在减少缓慢上升和下降输入信号产生错误边缘触发中断的概率。

23.4. 内存映射和寄存器

EPORT 模块内存映射情况见表 23-1.EPORT0 碱基地址为 0x400F_0000，EPORT1 为 0x4010_0000，EPORT2 为 0x401D_0000。本节介绍了内存映射和寄存器结构。

23.4.1. 内存映射

有关 EPORT 内存映射的描述，请参见表 23-1。

表 23-1：模块内存映射

偏移地址	Bits[15:8]	Bits[7:0]	访问权限(1)
0x0000	EPORT Data Direction Register (EPDDR)	EPORT Interrupt Enable Register (EPIER)	S
0x0002	EPORT Pin Assignment Register (EPPAR)		S
0x0004	EPORT Flag Register (EPFR)	EPORT Pin Pull-up enable Register (EPPUE)	S/U
0x0006	EPORT Data Register (EPDR)	EPORT Pin Data Register (EPPDR)	S/U
0x0008	EPORT Digital Filter Control Register (EPFC)	EPORT Bit Set Register (EPBSR)	S
0x000A	EPORT Level Polarity Register (EPLPR)	EPORT Open Drain Register (EPODE)	S
0x000C	保留		S
0x000E	EPORT Bit Clear Register (EPBCR)	保留	S

注意(1)：仅限 S = CPU 监控模式访问。S/U = CPU 监控或用户模式访问。在用户模式下只访问管理员地址没有影响，并导致周期终止传输错误。

©Levetop Semiconductor Co., Ltd.

23.4.2. 寄存器说明

EPORT 编程模型由以下寄存器组成：

- EPORT 引脚分配寄存器（EPPAR）单独控制每个引脚的功能。
- EPORT 数据方向寄存器（EPDDR）分别控制每一个引脚的方向。
- EPORT 中断启用寄存器（EPIER）分别为每个引脚启用中断请求。
- EPORT 数据寄存器（EPDR）保存要被驱动到引脚的数据。
- EPORT 引脚数据寄存器（EPPDR）反映了引脚的当前状态。
- EPORT 标志寄存器（EPFR）单独锁存 EPORT 边缘事件。
- EPORT 引脚向上拉拉启用寄存器（EPPUE）单独控制每个引脚的拉向上。
- EPORT 电平极性寄存器（EPLPR）控制对电平敏感的每个引脚的电平极性。
- EPORT 打开排水启用寄存器（EPODE）分别控制单独输出的每个引脚的打开排水。
- EPORT 数字滤波器控制寄存器（EPFC）启用滤波器，并控制将被滤波的输入脉冲的宽度。

23.4.2.1. 边缘端口中断启用寄存器（EPIER）

地址：EPORTn_BASEADDR+0x0000_0000

	7	6	5	4	3	2	1	0
读： 写：	EPIE[7:0]							
复位：	0	0	0	0	0	0	0	0

图 23-2：EPORT 端口中断启用寄存器（EPIER）

EPIE[7:0] — Edge Port Interrupt Enable Bits

读/写 EPIE[7:0]位启用 EPORT 中断请求。如果在 EPIER 中设置了一个位，EPORT 会在以下时候生成一个中断请求：

- EPORT 标志寄存器（EPFR）中的相应位被设置或以后被设置，或
- 相应的引脚电平较低，且引脚配置为电平敏感操作

在 EPIER 中清除一点将否定来自相应的 EPORT 引脚的任何中断请求。复位清除 EPIE[7:0]。

1 = 启用了来自相应 EPORT 引脚的中断请求

0 = 来自相应 EPORT 引脚的中断请求已被禁用

©Levetop Semiconductor Co., Ltd.

23.4.2.2. Eport 数据方向寄存器（EPDDR）

地址：EPORTn_BASEADDR+0x0000_0001

	7	6	5	4	3	2	1	0
读: 写:	EPDD[7:0]							
复位:	0	0	0	0	0	0	0	0

图 23-3：EPORT 数据方向寄存器（EPDDR）

EPDD[7:0] — Edge Port Data Direction Bits

在 EPDDR 中设置任何位都会将相应的引脚配置为输出。清除 EPDDR 中的任何位，都会将相应的引脚配置为输入。引脚的方向独立于水平/边缘检测配置。复位清除 EPDD[7:0]。

要使用 EPORT 引脚作为外部中断请求源，其在 EPDDR 中的相应位必须被清除。当 EPDDR 选择输出时，软件可以通过编程 EPORT 数据寄存器来生成中断请求。

1 = 对应的 EPORT 引脚配置为输出

0 = 对应的 EPORT 引脚配置为输入

23.4.2.3. Eport 引脚分配寄存器（EPPAR）

地址：EPORTn_BASEADDR+0x0000_0002

	15	14	13	12	11	10	9	8
读: 写:	eppa7[1:0]		eppa6[1:0]		eppa5[1:0]		eppa4[1:0]	
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读: 写:	eppa3[1:0]		eppa2[1:0]		eppa1[1:0]		eppa0[1:0]	
复位:	0	0	0	0	0	0	0	0

图 23-4：EPORT 引脚分配寄存器（EPPAR）

EPPAx[1:0] — EPORT Pin Assignment Select Fields

读/写 EPPAx 字段配置了水平检测和升降边缘检测的 EPORT 引脚，如表 23-2 所示。

配置为电平敏感的引脚被倒置，以便外部引脚上的逻辑 0 或逻辑 1 表示有效的中断请求。电级敏感的中断输入没有被锁定。为了保证电平敏感的中断请求被确认，中断源必须保持信号直到软件确认。必须选择电平灵敏度，以使设备脱离具有 INTx 中断的停止模式。

配置为边缘触发的引脚被锁住，不需要为中断生成保持断言。配置为边缘检测的引脚无论配置为输入或输出都被监控。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

表 23-2：EPPAx Field Settings

EPPAx	引脚配置
00	Pin INTx level-sensitive
01	Pin INTx rising edge triggered
10	Pin INTx falling edge triggered
11	Pin INTx both falling edge and rising edge triggered

在 EPORT 模块中生成的中断请求可以被中断控制器模块屏蔽。EPPAR 的功能与所选的 Pin 方向无关。

复位将清除 EPPAx 字段。

23.4.2.4. Eport 引脚向上上拉启用寄存器（EPPUE）

地址：EPORTn_BASEADDR+0x0000_0004

	7	6	5	4	3	2	1	0
读: 写:	埃普							
复位:	1	1	1	1	1	1	1	1

图 23-5：EPORT 引脚上拉式启用寄存器（EPPUE）

EPPUE[7:0] — Edge Port Pin Pull-up enable Bits

在 EPPUE 中设置任何位都会配置相应的引脚以启用向上。该设置只有在 EPORT 处于输入模式时才可用。清除 EPPUE 中的任何位将配置相应的引脚以禁用上拉。

复位设置 EPPUE[7:0]。

1 = 相应的 EPORT 引脚，配置为启用引线向上

0 = 相应的 EPORT 引脚，配置为禁用引线向上

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

23.4.2.5. 边缘端口标志寄存器（EPFR）

地址：EPORTn_BASEADDR+0x0000_0005

	7	6	5	4	3	2	1	0
读:	EPF[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 23-6：EPORT 端口标志寄存器（EPFR）

EPF[7:0] — Edge Port Flag Bits

当 EPORT 引脚配置为边缘触发时，其在 EPFR 中对应的读写位表示已检测到所选择的边缘。复位清除 EPF[7:0]。

1 = 已检测到 INTx 引脚的选定边缘。

0 = 未检测到 INTx 引脚的已选边缘。

当在相应的引脚上检测到选定的边时，将设置该寄存器中的位。一点保持设置，直到写入 1 清除。写入 0 没有效果。如果引脚配置为电级敏感（EPPARx = 00），则引脚转换不会影响该寄存器。

23.4.2.6. 边缘端口引脚数据寄存器（EPPDR）

地址：EPORTn_BASEADDR+0x0000_0006

	7	6	5	4	3	2	1	0
读:	EPPD[7:0]							
写:								
复位:	P	P	P	P	P	P	P	P

= 写入没有影响，访问终止，没有传输错误异常。

P = 电流引脚状态

图 23-7：EPORT 端口引脚数据寄存器（EPPDR）

EPD[7:0] — Edge Port Data Bits

只读 EPPDR 反映了 EPORT 引脚的当前状态。写入 EPPDR 没有效果，写入周期正常终止。复位不影响 EPPDR。

©Levetop Semiconductor Co., Ltd.

23.4.2.7. 边缘端口数据寄存器（EPDR）

地址：EPORTn_BASEADDR+0x0000_0007

	7	6	5	4	3	2	1	0
读： 写：	EPD[7:0]							
复位：	1	1	1	1	1	1	1	1

图 23-8：EPORT 端口数据寄存器（EPDR）

EPD[7:0] — Edge Port Data Bits

写入 EPDR 的数据存储在内部寄存器中；如果端口的任何引脚配置为输出，为该引脚存储的位被驱动到引脚上。读取 EDPR 将返回存储在寄存器中的数据。复位设置 EPD[7:0]。

23.4.2.8. Eport 端口位设置寄存器（EPBSR）

地址：EPORTn_BASEADDR+0x0000_0008

	7	6	5	4	3	2	1	0
读：	0							
写：	EPBSR[7:0]							
复位：	0	0	0	0	0	0	0	0

图 23-9：EPORT 端口位设置寄存器（EPBSR）

EPBSR[7:0] — EPORT Port Bit Set Register

1 = 将设置相应的 EPDR 位；

0 = 相应的 EPDR 位将不会被影响；

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

23.4.2.9. Eport 数字滤波器控制寄存器（EPFC）

地址：EPORTn_BASEADDR+0x0000_0009

	7	6	5	4	3	2	1	0
读: 写:	Filter_Width[6:0]							Filter_En
复位:	0	0	0	0	0	0	0	0

图 23-10：EPORT 数字滤波器控制寄存器（EPFC）

Filter_Width[6:0] — EPORT Digital Filter Pulse Width Select Bit

确定要被过滤的输入脉冲的宽度。如果输入的脉冲宽度小于（Filter_Width[6:0]+2），则将对脉冲进行滤波。

Filter_En — EPORT Digital Filter Enable Bit

1 = EPORT 数字滤波器为启用；

0 = EPORT 数字滤波器被禁用；

23.4.2.10. Eport Open Drain 启用寄存器（EPODE）

地址：EPORTn_BASEADDR+0x0000_000A

	7	6	5	4	3	2	1	0
读: 写:	EPODE[7:0]							
复位:	0	0	0	0	0	0	0	0

图 23-11：EPORT Open Drain 启用寄存器（EPODE）

EPODE[7:0] — Edge Port Open Drain enable Bits

如果 EPORT 配置为输出，则设置 EPODE 中的任何位都会将相应的引脚配置为“Open Drain”输出。清除 EPODE 中的任何位，都会将相应的引脚配置为 CMOS 输出。复位清除环。

1 = 对应的 EPORT 引脚配置为 Open Drain 输出

0 = 相应的 EPORT 引脚配置为 CMOS 输出

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

23.4.2.11. Eport 电平极性寄存器（EPLPR）

地址：EPORTn_BASEADDR+0x0000_0000B

	7	6	5	4	3	2	1	0
读: 写:	EPLPR[7:0]							
复位:	0	0	0	0	0	0	0	0

图 23-12：EPORT 级极性寄存器（EPLPR）

EPLPR[7:0]—边缘端口级别的极性位

如果 EPORT 配置为电平敏感，则在 EPLPR 中的任何位配置相应的引脚高电平敏感。清除 EPLPR 中的任何位将配置相应的引脚低电平敏感。复位清除 EPLPR[7:0]。

1 = 对应的 EPORT 引脚配置为高电平敏感型

0 = 相应的 EPORT 引脚配置为低电平敏感的

23.4.2.12. Eport 端口位清除寄存器（EPBCR）

地址：EPORTn_BASEADDR+0x0000_000F

	7	6	5	4	3	2	1	0
R	0							
W	EPBCR[7:0]							
复位:	0	0	0	0	0	0	0	0

图 23-13：EPORT 端口位清除寄存器（EPBCR）

EPBCR[7:0] — EPORT Port Bit Clear Register

1 = 相应的 EPDR 位将被清除；

0 = 相应的 EPDR 位将不会被影响；

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

24.CANBus 控制程序（CANBC）

24.1. 功能介绍

LT168 的 CANBus 模块是一个基于 CAN 2.0B 协议规范实现 CAN 协议的通信控制器。一般的方块图如图 24-1 所示，它描述了在 CANBus 模块中实现的主要子块，包括两个嵌入式存储器，一个用于存储消息缓冲区（MB），另一个用于存储 Rx 单个掩码寄存器。这些子模块的功能将在后续的章节中进行描述。

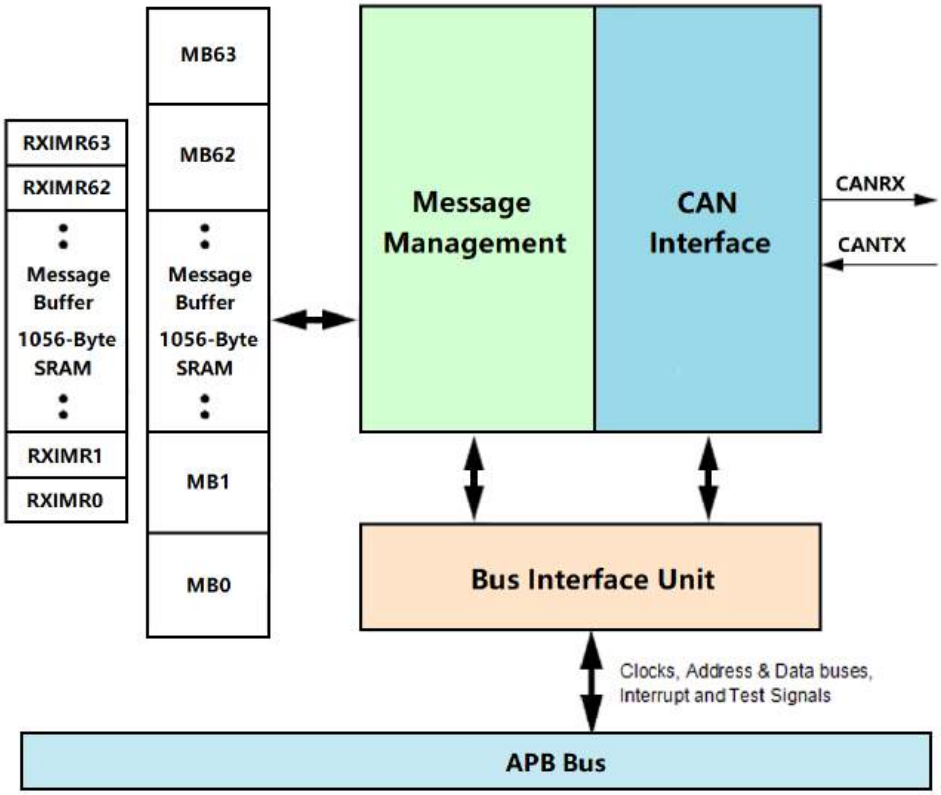


图 24-1：CANBus 方块图

24.1.1. 基本概述

CAN 协议主要但不仅仅是设计为作为车辆串行数据总线，满足该领域的具体要求：实时处理、在车辆 EMI 环境中的可靠操作、成本效益和所需的带宽。CANBus 模块是 CAN 协议规范 2.0 B 版本的完整实现，该规范支持标准的和扩展的消息帧。

最多有 64 个消息缓冲区可用。消息缓冲区存储在一个专用于 CANBus 模块的嵌入式 SRAM 中。

CAN 协议接口（CPI）子模块管理 CANBus 上的串行通信，请求 SRAM 访问以接收和发送消息帧，验证接收到的消息并执行错误处理。消息缓冲区管理（MBM）子模块处理接收和传输的消息缓冲区选择，处理仲裁和 ID 匹配算法。总线接口单元（BIU）子模块控制对内部接口总线的访问，以便建立与 CPU 和其他块的连接。时钟、地址和数据总线、中断输出和测试信号均通过总线接口单元进行访问。

©Levetop Semiconductor Co., Ltd.

24.1.2. CANBus 模块功能

CANBus 模块包括以下显着特点：

- 完全实现了 CAN 协议规范，版本 2.0B
  - 标准数据和远程帧
  - 扩展的数据和远程帧
  - 0~8 字节的数据长度
  - 可编程比特率高达 1 Mbit/s
  - 与内容相关的地址
- 64 个数据长度为 0 到 8 个字节的消息缓冲区
- 每个 MB 可配置为 Rx 或 Tx，所有支持标准和扩展消息
- 每个消息缓冲区的单独的 Rx 掩码寄存器
- 包括 1056 字节（64MB）的 SRAM 用于 MB 存储
- 包括 256 字节（64 MBs）的 SRAM，用于单个 Rx 掩码寄存器
- 全功能的 Rx FIFO 与存储容量为 6 帧和内部指针处理
- 强大的 Rx FIFO ID 过滤，能够匹配输入的 ID 与 8 个扩展，16 个标准或 32 个部分（8 位）ID，具有单独的屏蔽能力
- 可编程时钟源到 CANBUS 协议接口，或总线时钟或晶体振荡器
- 未使用的 MB 和 Rx 掩码寄存器空间可以用作通用的 SRAM 空间
- 只听模式能力
- 可编程回路返回模式，支持自检操作
- 可编程传输优先级方案：最低 ID、最低缓冲区号或最高优先级
- 基于 16 位自由运行计时器的时间戳
- 全局网络时间，由一个特定的消息进行同步
- 可移动的中断
- 独立于传输介质（假定有一个外部收发器）
- 由于针对高优先级消息的仲裁方案而导致的延迟时间较短
- 低功耗模式
- Tx 消息缓冲区上的硬件取消

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

24.1.3. 操作模式

CANBus 模块有四种功能模式：正常模式（用户和主管）、冻结模式、只收听模式和回路模式。还有一个低功率模式：禁用模式。

- **正常模式（用户或主管）**

  在正常模式下，该模块操作接收和/或发送消息帧，错误处理正常，所有的 CANBUS 协议功能都被启用。用户模式和监管模式在对某些受限控制寄存器的访问方面有所不同。

- **冻结模式**

  当断言 MCR 中的 FRZ 位时，将启用它。如果启用，则在设置 MCR 中的 HALT 位或在 MCU 级别请求调试模式时，将输入冻结模式。在这种模式下，没有帧的传输或接收，对 CANBus 的同步性丢失。

- **只听模式**

  当确认控制寄存器中的 LOM 位时，模块进入此模式。在此模式下，传输被禁用，所有错误计数器被冻结，模块以 CANBUS 错误被动模式运行。将只接收由另一个 CAN 站确认的消息。如果 CANBus 检测到一个未被确认的消息，它将标记一个 BIT0 错误（不更改 REC），就像它试图承认该消息一样。

- **回路回路模式**

  当确认控制寄存器中的 LPB 位时，模块进入此模式。在这种模式下，CANBus 执行一个可用于自检操作的内部循环。发送器的比特流输出在内部反馈到接收器输入。Rx CAN 输入引脚被忽略，Tx CAN 输出进入隐性状态（逻辑“1”）。CANBus 的行为与传输时相同，并将自己传输的消息视为从远程节点接收的消息。在此模式下，CANBus 忽略 CAN 帧确认字段中 ACK 槽期间发送的位，以确保正确接收自己的消息。同时产生发送中断和接收中断。

- **模块禁用模式**

  当断言 MCR 中的 MDIS 位时，将输入此低功率模式。当禁用时，该模块将关闭到 CAN 协议接口和消息缓冲区管理子模块的时钟。通过否定 MCR 中的 MDIS 位来退出此模式。

24.2. 外部信号描述

24.2.1. 基本概述

CANBus 模块有两个 I/O 信号连接到外部 MCU 引脚。这些信号汇总在表 24-1 中并在接下来的小节中有更详细的描述。

表 24-1：CANBus 信号

信号名称	方向	描述
CANRX	输入	CANBUS 接收引脚
CANTX	输出	CANBUS 传输引脚

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

24.2.2. 信号描述

24.2.2.1. C ANRX

此引脚是来自 CANBus 收发器的接收引脚。主导状态由逻辑级别“0”表示。隐性状态由逻辑级别“1”表示。

24.2.2.2. C ANTX

此引脚是发送到 CANBus 收发器的传输引脚。主导状态由逻辑级别“0”表示。隐性状态由逻辑级别“1”表示。

以下是 Can 总线应用电路的例子。

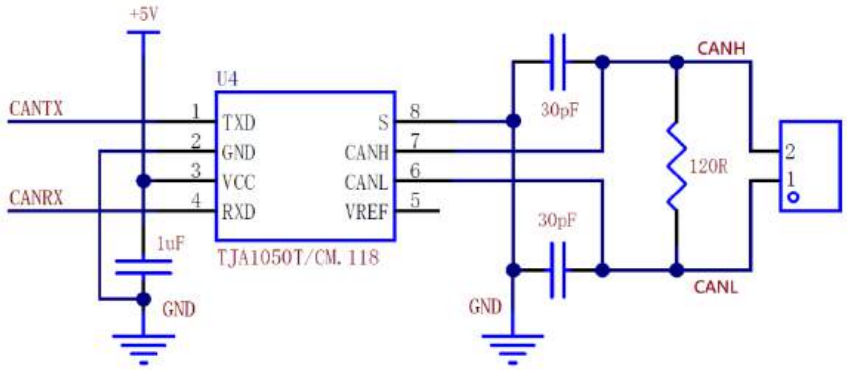


图 24-2：Canbus 电路示例

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

25.串行通信接口（SCI）

25.1. 功能介绍

LT168 的串行通信接口模块（SCI）支持基本的 UART，并允许与外围设备和其他微控制器单元（MCU）进行异步串行通信。该模块还支持 LIN 从属操作。

25.2. 通信接口特点

每个 SCI 模块的特点包括：

- 全双工，标准的非返回到零（NRZ）格式
- 可编程波特率（13 位调制分频器），具有可配置的过采样比，从 4 倍到 256 倍
- 中断，轮询操作：
  - 传输数据寄存器为空，传输完成
  - 完全接收数据寄存器
  - 接收溢出、奇偶校验误差、帧误差和噪声误差
  - 怠速接收器检测
  - 接收引脚上的活动边缘
  - 中断检测所支持的 LIN
  - 接收数据匹配
- 硬件奇偶校验的生成和检查
- 可编程的 8 位、9 位或 10 位字符长度
- 可编程的 1 位或 2 位停止位
- 三种接收器唤醒方法：
  - 怠速线路唤醒
  - 地址标记唤醒
  - 接收数据匹配
- 自动地址匹配，以减少 ISR 开销：
  - 地址标记匹配
  - 空闲线路地址匹配
  - 地址匹配开始，地址匹配结束
- 可选的 13 位中断字符生成/11 位中断字符检测
- 可配置的空闲长度检测，支持 1、2、4、8、16、32、64 或 128 个空闲字符
- 可选择的发送器输出和接收器输入极性

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

- 可选择的 IrDA 1.4 返回到零反转（RZI）格式，具有可编程的脉冲宽度
- 独立的 FIFO 结构
  - 接收和传输请求的可配置水印
  - 如果接收 FIFO 不是空，接收方在可配置的空闲字符之后断言请求的选项

25.3. 操作模式

- 停止模式
- 等待模式

25.4. 方块图

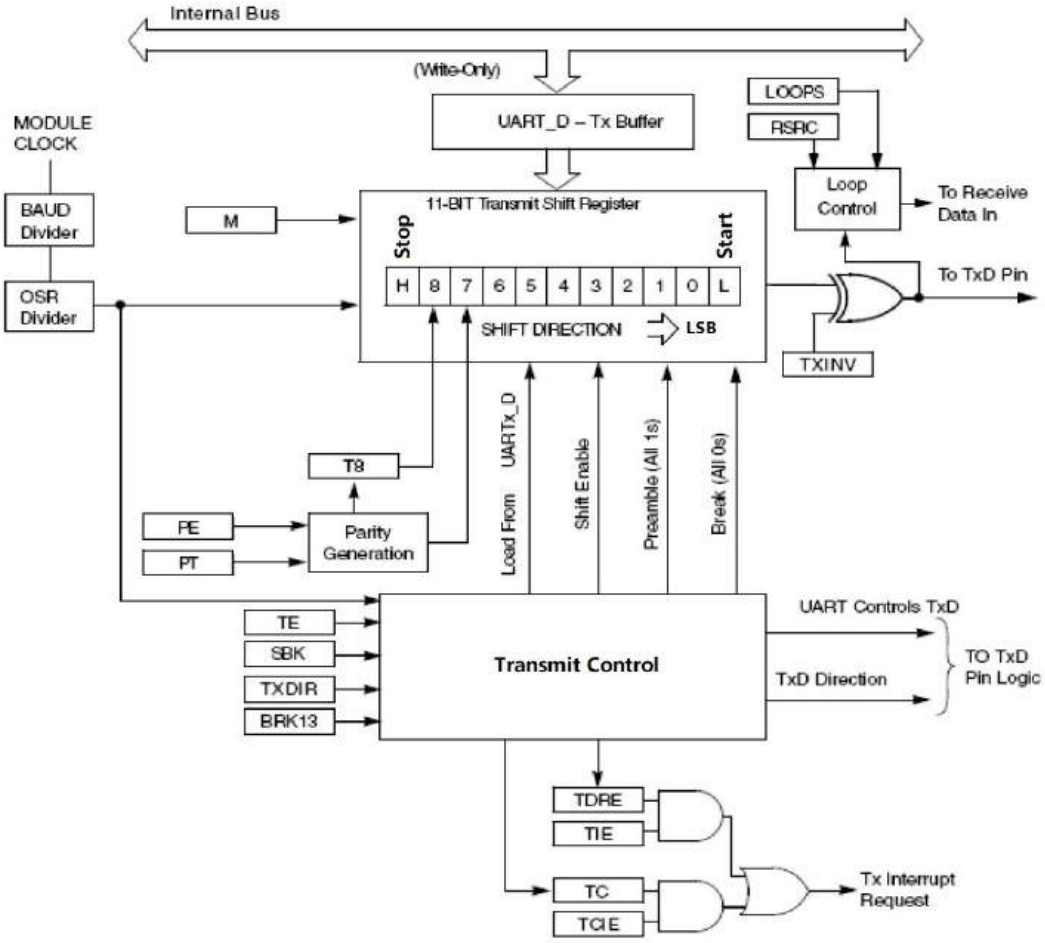


图 25-1：SCI 发送器方块图

©Levetop Semiconductor Co., Ltd.

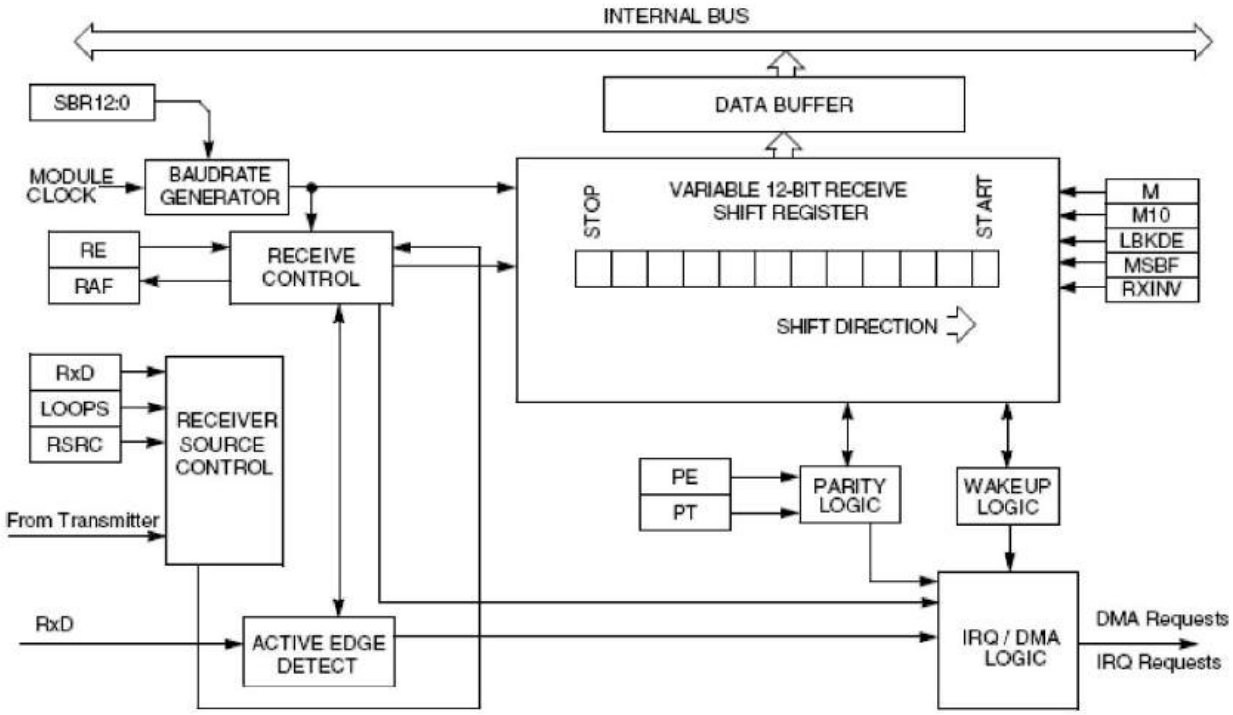


图 25-2：SCI 接收器方块图

25.5. 操作模式

25.5.1. 停止模式

在停止模式下将无法工作。

25.5.2. 等待 Mode

当设置了 DOZEEN 位时，SCI 可以被配置为在等待模式下停止。

发送器和接收器将完成当前单词的发送/接收。

25.6. 信号描述

表 25-1 给出了这里所描述的信号的概述。

表 25-1：信号属性

信号	描述	I/O
TXD	传输数据。此引脚通常是输出，但当发送器被禁用或为接收数据配置传输方向时，它是单线模式下的输入（三态）。	I/O
RXD	接收数据	I
SCI_DE	RS-485 收发信机的控制信号	O

©Levetop Semiconductor Co., Ltd.

25.7. 内存映射和寄存器

25.7.1. 内存映射

SCI 模块记忆图如表 25-2 所示.SCI0 的基本地址是 0x4009_0000，SCI1 是 0x4008_0000，SCI2 是 0x400C_0000。

表 25-2：SCI 模块记忆地图

偏移地址	Bits[31:0]
0x0000	SCI Version ID (SCI_VERID)
0x0004	SCI Parameter (SCI_PARAM)
0x0008	SCI Reset (SCI_RESET)
0x000C	SCI Pin (SCI_PIN)
0x0010	SCI Baud Rate Register (SCI_BAUD)
0x0014	SCI Status Register (SCI_STAT)
0x0018	SCI Control Register (SCI_CTRL)
0x001C	SCI Data Register (SCI_DATA)
0x0020	SCI Match Address Register (SCI_MATCH)
0x0024	SCI Modem IrDA Register (SCI_MODIR)
0x0028	SCI FIFO Register (SCI_FIFO)
0x002C	SCI Watermark Register (SCI_WATER)
0x0030	SCI Oversampling Ratio Register(SCI_OSR)

提示：

每个模块被分配了 16K 字节的地址空间，所有这些都可能无法被解码。在指定的模块内存映像之外的访问将生成总线错误异常。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

25.7.2. 寄存器说明

25.7.2.1. SCI 版本 ID 寄存器（SCI_VERID）

地址：SCIn_BASEADDR+0x0000_0000

	31	30	29	28	27	26	25	24
读:	VERID[31:24]							
写:								
复位:	0	0	0	0	0	1	0	0

	19	22	21	20	19	18	17	16
读:	VERID[23:16]							
写:								
复位:	0	0	0	0	0	0	0	0

	11	14	13	12	11	10	9	8
读:	VERID[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	VERID[7:0]							
写:								
复位:	0	0	0	0	0	0	1	1

= 写入没有影响，访问终止，没有传输错误异常。

图 25-3：SCI 版本 ID 寄存器（SCI_VERID）

VERID_ID[31:0] — SCI Version ID

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

25.7.2.2. SCI 参数寄存器（SCI_PARAM）

地址：SCIn_BASEADDR+0x0000_0004

	31	30	29	28	27	26	25	24
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	保留				RXFIFO_SZ[3:0]			
写:								
复位:	0	0	0	0	0	0	1	1

	7	6	5	4	3	2	1	0
读:	保留				TXFIFO_SZ[3:0]			
写:								
复位:	0	0	0	0	0	0	1	1

= 写入没有影响，访问终止，没有传输错误异常。

图 25-4：SCI 参数寄存器（SCI_PARAM）

RXFIFO_SZ[3:0] — The receive buffer/FIFO size

此只读控制位表示接收缓冲区/FIFO 的大小。

TXFIFO_SZ[3:0] — The transmit buffer/FIFO size

此只读控制位表示传输缓冲区/FIFO 的大小。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

25.7.2.3. SCI 复位寄存器（SCI_RESET）

地址：SCIn_BASEADDR+0x0000_0008

	31	30	29	28	27	26	25	24
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	保留						SWRST	保留
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 25-5：SCI 复位寄存器（SCI_RESET）

SWRST — Software Reset

这个读/写位允许软件复位 SCI IP。

1 = 软件复位已确认

0 = 未确认有任何软件复位

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

25.7.2.4. SCI 引脚寄存器（SCI_PIN）

地址：SCIn_BASEADDR+0x0000_000C

	31	30	29	28	27	26	25	24
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	保留						PINCFG	
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 25-6：SCI 引脚登记（SCI_PIN）

PINCFG — Useless in this module

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

25.7.2.5. SCI 波特率寄存器（SCI_BAUD）

地址：SCIn_BASEADDR+0x0000_0010

	31	30	29	28	27	26	25	24
读:	MAEN1	MAEN2	M10	保留				
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	TDMAE	Reserved	RDMAE	Reserved	MATCFG[1:0]		BOTHEDG	RESYNCDIS
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	LBKDIE	RXEDGIE	SBNS	SBR[12:8]				
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	SBR[7:0]							
写:								
复位:	0	0	0	0	0	1	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 25-7：SCI 波特率寄存器（SCI_BAUD）

MAEN1 — Match Address Mode Enable 1

1 = 启用自动地址匹配或数据匹配模式 MATCH[MA1]

0 = 正常操作

MAEN2 — Match Address Mode Enable 2

1 = 启用自动地址匹配或数据匹配模式 MATCH[MA2]

0 = 正常操作

M10 — 10-bits Mode select

M10 位使十分之一位成为串行传输的一部分。只有当发送器和接收器都被禁用时，才应该更改此位。

1 = 接收器和发送器使用 10 位数据字符

0 = 接收器和发送器使用 8 位或 9 位的数据字符

TDMAE — Transmitter DMA Enable

TDMAE 配置传输数据寄存器空标志 LPUART_STAT[TDRE]，以生成 DMA 请求。

1 = DMA 请求允许

0 = DMA 请求被禁用

©Levetop Semiconductor Co., Ltd.

RDMAE — Receiver Full DMA Enable

RDMAE 配置接收方数据寄存器全标志 LPUART_STAT[RDRF]，以生成 DMA 请求。

1 = DMA 请求允许

0 = DMA 请求已被禁用

MATCFG[1:0] — Match Configuration

配置所使用的匹配寻址模式：

00 -- Address Match Wakeup;
01 -- Idle Match Wakeup;
10 -- Match On and Match Off;
11 -- Enable RWU on Data Match and Match On/Off for transmitter CTS input

BOTHEDGE — Both Edge Sampling

允许在波特率时钟的两侧边缘上对接收到的数据进行采样，在给定的过采样比下，有效地使接收器对输入数据的采样次数翻倍。这个位必须设置为 x4 和 x7 之间的过采样比，并且对于更高的过采样比是可选的。这个位只有在接收器被禁用时才应该改变。

1 = 接收器使用波特率时钟的上升和下降边缘采样输入数据。

0 = 接收器使用波特率时钟的上升边缘采样输入数据。

RESYNCDIS — Resynchronization Disable

当设置时，当检测到数据 1 和数据零转换时，它禁用接收到的数据字的重新同步。只有在禁用接收器的情况下，才应更改此位。

1 = 在接收数据字期间禁用重新同步

0 = 支持接收数据字期间的重新同步

LBKDIE — LIN Break Detect Interrupt Enable

LBKDIE 允许 LIN 中断检测标志 LBKDIF 来生成中断请求。

1 = 当 SCI_STAT[LBKDIF]标志为 1 时，将请求硬件中断

0 = 来自 SCI_STAT[LBKDIF]的硬件中断已被禁用（使用轮询）

RXEDGIE — RX Input Active Edge Interrupt Enable

启用接收输入活动边缘 RXEDGIF 来生成中断请求。在设置 RXEDGIE 时，更改 CTRL[LOOP]或 CTRL[RSRC]可能会导致设置 RXEDGIF。

1 = 当 SCI_STAT 标志为 1 时，请求硬件中断

0 = 个来自 SCI_STAT 的硬件中断被禁用（使用轮询）

SBNS — Stop Bit Number Select SBNS determines whether data characters are one or two stop bits. This bit should only be changed when the transmitter and receiver are both disabled.

1 = 两个停止位

0 = 一个停止位

©Levetop Semiconductor Co., Ltd.

SBR[12:0] — Baud Rate Modulo Divisor

SBR[12:0]中的 13 位数据为波特率发生器设置模除法率。当 SBR 为 1~8191 时，波特率等于“baud clock / ((OSR+1) × SBR)”。只有当发送器和接收器都被禁用（SCI_CTRL[RE] andSCI_CTRL[TE]均为 0）时，才能更新 13 位波特率设置[SBR12:SBR0]。

25.7.2.6. SCI 状态寄存器（SCI_STAT）

地址：SCIn_BASEADDR+0x0000_0014

	31	30	29	28	27	26	25	24
读:	LBKDIF	RXEDGIF	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF
写:	w1c	w1c						
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
写:				w1c	w1c	w1c	w1c	w1c
复位:	1	1	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	MA1F	MA2F	保留					
写:	w1c	w1c						
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	保留							
sbr								
复位:	0	0	0	0	0	1	0	0

= 写入没有影响，访问终止，没有传输错误异常。

w1c = .写 1 到该位将清除它

图 25-8：SCI 状态登记器（SCI_STAT）

LBKDIF — LIN Break Detect Interrupt Flag

当 LIN 断开检测电路启用并检测到 LIN 断开字符时，设置 LBKDIF。LBKDIF 通过写 1 来清除。

1 = 已检测到 LIN 中断字符

0 = 未检测到 LIN 中断字符

©Levetop Semiconductor Co., Ltd.

RXEDGIF — RXD Pin Active Edge Interrupt Flag

RXEDGIF 是在 RXD 引脚上出现活动边时设置的，如果 RXINV = 0 时下降缘，如果 RXINV = 1 时上升缘。RXEDGIF 可以通过写一个 1 来清除它。

1 = 在接收引脚上出现了一个活动的边缘

0 = 在接收引脚上没有发生活动边缘

MSBF — MSB First

设置此位会反转在数据线上传输和接收的位的顺序。这个位不影响这些位的极性、奇偶校验位的位置或开始位或停止位的位置。只有当发送器和接收器都被禁用时，才应该更改此位。

1 = MSB（位 9、位 8、位 7 或位 6 位）是在开始位之后传输的第一个位，这取决于 CTRL[M]、CTRL[PE]和 BAUD[M10]的设置。此外，根据 CTRL[M]和 CTRL[PE]的设置，在开始位之后接收到的第一位被识别为位 9、位 8、位 7 或位 6。

0 = LSB（位 0）是在起始位之后传输的第一个位。此外，在起始位之后接收到的第一位被识别为位 0

RXINV — Receive Data Inversion

设置此位会反转接收到的数据输入的极性。设置 RXINV 会反转所有情况下的 RXD 输入：数据位，开始和停止位，中断，

和空闲。

1 = 接收数据被倒置

0 = 接收数据没有反转

RWUID — Receive Wake Up Idle Detect

对于空闲字符上的 RWU，RWUID 控制唤醒接收器的空闲字符是否设置了 idle 位。对于地址匹配唤醒，RWUID 控制在地址不匹配时是否设置了 IDLE 位。只有在禁用接收器的情况下，才应更改此位。

1 = 在接收待机状态（RWU = 1）期间，在检测到空闲字符时将设置 IDLE 位。在地址匹配唤醒期间，当地址不匹配时，确实会设置 IDLE 位。

0 = 在接收待机状态（RWU = 1）期间，在检测到空闲字符时不会设置 IDLE 位。在地址匹配唤醒期间，当地址不匹配时，不会设置 IDLE 位。

BRK13 — Break Character Generation Length

BRK13 选择一个较长的传输中断字符长度。帧错误的检测不受该位的状态的影响。只有在发送器被禁用时，才应更改这个位。

1 = Break 字符的传输长度为 13 位次（如果 M = 0、SBNS = 0）或 14（如果 M = 1、SBNS = 0 或 M = 0、SBNS = 1）或 15（如果 M = 1、SBNS = 1 或 M10 = 1、SNBS = 0）或 16（如果 M10 = 1、SNBS = 1）。

0 = Break 字符的传输长度为 10 位次（如果 M = 0、SBNS = 0）或 11（如果 M = 1、SBNS = 0 或 M = 0、SBNS = 1）或 12（如果 M = 1、SBNS = 1 或 M10 = 1、SNBS = 0）或 13（如果 M10 = 1、SNBS = 1）

©Levetop Semiconductor Co., Ltd.

LBKDE — LIN Break Detection Enable

LBKDE 选择一个较长的断点字符检测长度。当设置 LBKDE 时，接收数据不存储在接收数据缓冲区中。

1 = 中断字符检测长度为 11 位次 (if M = 0, SBNS = 0) or 12 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 14 (if M = 1, SBNS = 1 or M10 = 1, SNBS = 0) or 15 (if M10 = 1, SNBS = 1)

0 = 中断字符检测长度为 10 位次 (if M = 0, SBNS = 0) or 11 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 12 (if M = 1, SBNS = 1 or M10 = 1, SNBS = 0) or 13 (if M10 = 1, SNBS = 1)

RAF — Receiver Active Flag

当接收方检测到有效启动位的开始时，将设置 RAF，当接收方检测到空闲线时，将自动清除 RAF。

1 = SCI 接收器激活(RXD 输入非空闲)

0 = SCI 接收器空闲，等待一个启动位

TDRE — Transmit Data Register Empty Flag

当发射 FIFO 启用时，TDRE 将设置发射 FIFO （SCI_DATA）中的数据字数等于或小于 SCI_WATER[TXWATER])。要清除 TDRE，请写入 SCI 数据寄存器（SCI_DATA），直到传输 FIFO 中的单词数大于 SCI_WATER[TXWATER]所表示的数字。当传输 FIFO 被禁用时，TDRE 将在传输数据寄存器（SCI_DATA）为空时进行设置。要清除 TDRE，请写入 SCI 数据寄存器（SCI_DATA）。

TDRE 不受传输过程中的字符的影响，它在每个传输字符开始时更新。

1 = 传输数据缓冲区为空

0 = 传输数据缓冲区已满

TC — Transmission Complete Flag

当传输正在进行中或加载了前导字符或断开字符时，TC 将被清除。当传输缓冲区为空且没有传输数据、前导或中断字符时，设置 TC。当 TC 设置时，发射数据输出信号变为空闲（逻辑 1）。TC 通过写入 SCI_DATA 传输新数据，通过清除并设置 SCI_CTRL[TE]，通过写入 1 写入 SCI_CTRL[SBK]排队中断字符。

1 = 发送器空闲（传输活动完成）

0 = 发送器处于活动状态（发送数据、前导码或中断）

RDRF — Receive Data Register Full Flag

当启用接收 FIFO 时，当接收缓冲区中的数据字数大于指示的数字 bySCI_WATER[RXWATER]时，将设置 RDRF。要清除 RDRF，请读取 SCI_DATA，直到接收数据缓冲区中的数据字数等于或小于 SCI_WATER[RXWATER]所指示的数字。当接收 FIFO 被禁用时，当接收缓冲区（SCI_DATA）已满时，将设置 RDRF。若要清除 RDRF，请读取 SCI_DATA 寄存器。

正在接收过程中的字符在接收到整个字符之前不会导致 RDRF 的更改。即使设置了 RDRF，该字符也将继续被接收，直到接收到整个字符后出现溢出情况为止。

1 = 接收数据缓冲区已满

0 = 接收数据缓冲区为空

©Levetop Semiconductor Co., Ltd.

IDLE — Idle Line Flag

当 SCI 接收行在一段活动后完全字符空闲时，设置 IDLE。当 ILT 被清除时，接收器在开始位之后开始计算空闲位时间。如果接收字符全部为 1s，这些位时间和停止位时间计入逻辑高的全部字符时间，10 到 13 位时间，需要接收器检测空闲线。当设置 ILT 时，接收器直到停止位之后才开始计算空闲位时间。前一个字符结尾的停止位和任何逻辑高位时间不计入接收器检测空闲行所需的逻辑高的全部字符时间。

要清除 IDLE，请将逻辑 1 写入 IDLE 标志。清除 IDLE 后，只有在接收缓冲区中存储了一个新字符或一个 LIN 中断字符设置了 LBKDIF 标志之后，才能再次设置它。即使接收行长时间空闲，也只设置一次。

1 = 检测到空闲线（Idle Line）

0 = 未检测到空闲线

OR — Receiver Overrun Flag

当软件无法防止接收数据寄存器溢出数据时，将设置 OR。在完全收到溢出缓冲区的数据字的停止位后，立即设置 OR 位，并阻止所有其他错误标志（FE、NF 和 PF）进行设置。移位寄存器中的数据丢失，但 SCI 数据寄存器中的数据不受影响。如果启用了 LBKDE 并且检测到 LIN Break，则如果在接收下一个数据字符之前没有清除 LBKDIF，LBKDE 字段将断言。

在设置 OR 标志时，即使存在足够的空间，数据缓冲区中也不会存储额外的数据。若要清除 OR，请将逻辑 1 写入 OR 标志。

1 = 接收溢出（新的 SCI 数据丢失）

0 = 无溢出

NF — Noise Flag

接收器中使用的先进采样技术在每个接收的比特中采集三个样本。如果这些样本中的任何一个与帧中的任何比特时间内的其他样本不一致，则检测到该字符的噪声。每当从 SCI_DATA 中读取的下一个字符在字符中检测到噪声时，就设置 NF。要清除 NF，请将逻辑 1 写入 NF。

1 = 在 SCI_DATA 中的接收字符中检测到噪声

0 = 未检测到噪声

FE — Framing Error Flag

当接收到从 SCI_DATA 读取的下一个字符，检测到需要停止位时，就设置 FE。要清除 FE，请将逻辑 1 写入 FE。

1 = 框架错误

0 = 未检测到帧错误。这并不能保证框架是正确的

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

PF — Parity Error Flag

当启用奇偶校验（PE = 1）时，当接收到要从 SCI_DATA 读取的下一个字符，并且接收到的字符中的奇偶校验位与预期的奇偶校验值不一致时，就会设置 PF。要清除 PF，请向 PF 写入一个逻辑 1。

1 = 奇偶校验错误

0 = 无奇偶校验错误

MA1F — Match 1 Flag

当从 SCI_DATA 读取的下一个字符与 MA1 匹配时，将设置 MA1F。要清除 MA1F，请向 MA1F 写入一个逻辑 1。

1 = 接收到的数据等于 MA1

0 = 接收到的数据不等于 MA1

MA2F — Match 2 Flag

当从 SCI_DATA 读取的下一个字符与 MA2 匹配时，将设置 MA2F。要清除 MA2F，请向 MA2F 写入一个逻辑 1。

1 = 接收到的数据等于 MA2

0 = 接收到的数据不等于 MA2

25.7.2.7. SCI 控制寄存器（SCI_CTRL）

地址：SCIn_BASEADDR+0x0000_0018

	31	30	29	28	27	26	25	24
读: 写:	R8T9	R9T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读: 写:	TIE	TCIE	RIE	ILIE	TE	RE	RUW	SBK
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	MA1IE	MA2IE	Reserved			IDLECFG		
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读: sbr	LOOPS	DOZEEN	RSRC	M	WAKE	ILT	PE	PT
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 25-9：SCI 控制寄存器（SCI_CTRL）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

R8T9 — Receive Bit 8 / Transmit Bit 9

R8 是当 SCI 配置为 9 位或 10 位数据格式时接收到的第 9 个数据位。当读取 9 位或 10 位的数据时，在读取 SCI_DATA 之前先读取 R8。

T9 是 SCI 配置为 10 位数据格式时接收的第 1 个数据位。当写入 10 位数据时，在写入 SCI_DATA 之前写入 T9。如果 T9 不需要改变它以前的值，例如当它被用于生成地址标记或奇偶校验时，那么它不需要被写入每个 timeSCI_DATA。

R9T8 — Receive Bit 9 / Transmit Bit 8

R9 是当 SCI 配置为 10 位数据格式时接收到的第 10 个数据位。当读取 10 位数据时，在读取 SCI_DATA 之前读取 R9

T8 是当 SCI 配置为 9 位或 10 位数据格式时接收到的第 9 个数据位。当写入 9 位或 10 位数据时，在 writingSCI_DATA 之前写入 T8。如果 T8 不需要改变它以前的值，例如当它被用于生成地址标记或奇偶校验时，那么它就不需要在每次写 SCI_DATA 时都被写出来。

TXDIR — TXD Pin Direction in Single-Wire Mode

当 SCI 配置为单线半双工操作（LOOPS = RSRC = 1）时，该位确定了 TXD 引脚处的数据方向。当清除 TXDIR 时，发送器将在接收器开始从 TXD 引脚接收数据之前完成接收当前字符（如果有的话）。

1 = TXD 引脚是在单线模式下的输出

0 = TXD 引脚是一个在单线模式下的输入

TXINV — Transmit Data Inversion

设置此位会反转传输数据输出的极性。设置 TXINV 会反转所有情况下的 TXD 输出：数据位、启动和停止位、中断和空闲。

1 = 传输数据被倒置

0 = 传输数据没有反转

ORIE — Overrun Interrupt Enable

此位允许溢出标志（OR）生成硬件中断请求。

1 = 在设置 OR 时，请求硬件中断

0 = 或中断已被禁用；使用轮询

NEIE — Noise Error Interrupt Enable

此位使噪声标志（NF）能够生成硬件中断请求。

1 = 当设置 NF 时，请求硬件中断

0 = NF 中断已被禁用；使用轮询

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

FEIE — Framing Error Interrupt Enable

此位允许框架错误标志（FE）生成硬件中断请求。

1 = 当设置 FE 时，请求硬件中断

0 = FE 中断已被禁用；使用轮询

PEIE — Parity Error Interrupt Enable

此位使奇偶校验错误标志（PF）能够生成硬件中断请求。

1 = 当设置 PF 时，请求硬件中断

0 = PF 中断已被禁用；使用轮询

TIE — Transmit Interrupt Enable

此位使 STAT[TDRE]能够生成中断请求。

1 = 当 TDRE 标志为 1 时，请求硬件中断

0 = 来自 TDRE 的硬件中断已被禁用；使用轮询

TCIE — Transmission Complete Interrupt Enable for

TCIE 允许传输完成标志 TC 生成中断请求。

1 = 当 TC 标志为 1 时，请求硬件中断

0 = 来自 TC 的硬件中断已被禁用；使用轮询

RIE — Receiver Interrupt Enable

此位使 STAT[RDRF]能够生成中断请求。

1 = 当 RDRF 标志为 1 时，请求硬件中断

0 = 来自 RDRF 的硬件中断已被禁用；使用轮询

ILIE — Idle Line Interrupt Enable

ILIE 允许空闲行标志 STAT[IDLE]来生成中断请求。

1 = 当 IDLE 标志为 1 时，请求硬件中断

0 = 来自 IDLE 的硬件中断已被禁用；使用轮询

TE — Transmitter Enable

此位可启用 SCI 发送器。TE 还可以通过清除然后设置 TE 来设置空闲前导队列。当 TE 被清除时，这个寄存器位将读取为 1，直到发送器完成当前字符，TXD 引脚是三态的。

1 = 发送器启用

0 = 发送器被禁用

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

RE — Receiver Enable

此位可启用 SCI 接收器。当 RE 写入 0 时，此寄存器位将读取为 1，直到接收器完成接收到当前字符（如果有的话）。

1 = 接收器启用

0 = 接收器被禁用

RWU — Receiver Wakeup Control

此字段可以设置为将 SCI 接收器置于待机状态。当 RWU 事件发生时，RWU 自动清除，即当 CTRL[WAKE]清除时 IDLE 事件，或当 CTRL[WAKE]与 STAT[RWUID]清除时地址匹配。

如果通道当前没有空闲，则 RWU 必须仅设置为 CTRL[WAKE] = 0（空闲时唤醒）。这可以由国家统计局来确定。如果该标志被设置为唤醒一个 IDLE 事件，而该通道已经空闲，则 SCI 可能会丢弃数据。这是因为在允许重新插入 IDLE 之前，必须在检测到 IDLE 之后接收数据或检测到 LIN 中断。

1 = SCI 接收器处于备用状态，等待唤醒状态

0 = 接收器正常工作

SBK — Send Break

将 1 和 0 写入 SBK 在传输数据流中的中断字符。额外的中断字符为 10 到 13，或 13 到 16，如果设置了 SCI_STATBRK13]，只要设置了 SBK，逻辑 0 的位时间就会排队。根据设置的时间和 SBK 相对于当前正在传输的信息的清除情况，在软件清除 SBK 之前，可能会排队等待第二个中断字符。

1 = 要发送的队列中断字符(s)

0 = 传送器正常工作

MA1IE — Match 1 Interrupt Enable

1 = MA1F 中断启用

0 = MA1F 中断禁用

MA2IE — Match 2 Interrupt Enable

1 = MA2F 中断启用

0 = MA2F 中断禁用

IDLECFG — Idle Configuration

此位确认了在设置 idle 标志之前必须接收到的空闲字符数。

000 --1 idle character;
001 -- 2 idle characters;
010 -- 4 idle characters;
011 -- 8 idle characters;
100 -- 16 idle characters;
101 -- 32 idle characters;
110 -- 64 idle characters;

©Levetop Semiconductor Co., Ltd.

111 -- 128 idle characters

LOOPS — LOOP Mode Select

当设置环路时，RXD 引脚与 SCI 断开，发送器输出在内部连接到接收器输入。发送器和接收器必须启用才能使用循环功能。

1 = 回路模式或单线模式，其中发送器输出内部连接到接收器输入（参见 RSRC 位）

0 = 正常操作- RXD 和 TXD 使用单独的引脚

DOZEEN — Doze Enable

1 = SCI 在打盹模式下被禁用

0 = SCI 在打盹模式下启用

RSRC — Receiver Source Select

除非设置了循环字段，否则此字段没有任何意义或效果。当设置循环时，RSRC 字段确定接收器移位寄存器输入的源。

1 = 单线 SCI 模式，其中 TXD 引脚连接到发送器输出和接收器输入

0 = 提供的循环被设置，RSRC 被清除，选择内环回路模式，SCI 不使用 RXD 引脚

M — 9-Bit or 8-Bit Mode Select

1 = 接收器和发送器使用 9 位数据字符

0 = 接收器和发送器使用 8 位数据字符

WAKE — Receiver Wakeup Method Select

这个位确定当 RWU = 1：地址标记在接收数据字符的最重要位位置或接收引脚输入信号的空闲状态时，哪个状态唤醒 SCI。

1 = 配置 RWU 与地址标记唤醒

0 = 配置 RWU 以进行空闲线路唤醒

ILT — Idle Line Type Select

该位确定接收器何时开始将逻辑 1 计算为空闲字符位。计数可以在有效的开始位之后或在停止位之后开始。如果计数在开始位之后开始，那么在停止位之前的逻辑串 1 可能会导致对空闲字符的错误识别。在停止位后开始计数可以避免错误的空闲字符识别，但需要正确的同步传输。如果 SCI 是用 ILT = 1 编程的，那么一个逻辑 0 会在接收到的停止位后被自动移位，因此会复位空闲计数。

1 = 空闲字符位计数在停止位之后开始

0 = 空闲字符位计数在开始位之后开始

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

PE — Parity Enable

此位支持硬件奇偶校验生成和检查。当启用奇偶校验时，停止位之前的位立即被视为奇偶校验位。

1 =启用奇偶校验

0 = 无硬件奇偶校验生成或检查

PT — Parity Type

如果启用了奇偶校验（PE = 1），该位将选择偶数或奇数奇偶校验。奇数奇偶性是指数据字符中的 1s 的总数，包括奇偶性位，是奇数。偶数意味着数据字符中的 1 总数，包括奇偶校验位，是偶数。

1 = 奇数校验

0 = 偶数校验

25.7.2.8. SCI 数据寄存器（SCI_DATA）

地址：SCIn_BASEADDR+0x0000_001C

	31	30	29	28	27	26	25	24
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	NOISY	PARITYE	FRETSC	RXEMPT	IDLINE	Reserved	R9T9	R8T8
写:								
复位:	0	0	0	1	0	0	0	0

	7	6	5	4	3	2	1	0
读: sbr	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 25-10：SCI 数据寄存器（SCI_DATA）

NOISY — The current received dataword contained in DATA[R9:R0] was received with noise

1 = 数据接收有噪声

0 = 数据字接收到无噪声

©Levetop Semiconductor Co., Ltd.

PARITYE — The current received dataword contained in DATA[R9:R0] was received with a parity error

1 = 该数据字收到了一个奇偶校验错误

0 = 接收到的数据字没有出现奇偶校验错误

FRETSC — Frame Error / Transmit Special Character

对于读取，此位表示 DATA[R9：R0]中包含的当前接收到的数据字存在帧错误。对于写入，此位表示要传输一个中断字符或空闲字符，而不是 DATA[T9：T0]中的内容。T9 用于表示 0 时的中断字符，以及表示 1 时的空闲字符，则 DATA[T8：T0]的内容应该为零。

1 = 数据字收到帧错误，传输空闲或中断字符

0 = 接收到的数据字在读取时没有出现帧错误，在写入时传输一个正常字符

RXEMPT — Receive Buffer Empty

当接收缓冲区中没有数据时，此位会断言。此字段不考虑包含在接收移位寄存器中的数据。

1 = 接收缓冲区为空，读取时返回的数据无效

0 = 接收缓冲区包含有效的数据

IDLINE — Idle Line

此位表示在接收 DATA[9:0]中的字符之前接收线处于空闲状态。与 IDLE 标志不同，这个位可以为接收器第一次接收的第一个字符设置。

1 = 接收器在接收此字符之前已空闲

0 = 接收器在接收此字符之前没有空闲状态

R9T9 — Read receive data buffer 9 or write transmit data buffer 9
R8T8 — Read receive data buffer 8 or write transmit data buffer 8
R7T7 — Read receive data buffer 7 or write transmit data buffer 7
R6T6 — Read receive data buffer 6 or write transmit data buffer 6
R5T5 — Read receive data buffer 5 or write transmit data buffer 5
R4T4 — Read receive data buffer 4 or write transmit data buffer 4
R3T3 — Read receive data buffer 3 or write transmit data buffer 3
R2T2 — Read receive data buffer 2 or write transmit data buffer 2
R1T1 — Read receive data buffer 1 or write transmit data buffer 1
R0T0 — Read receive data buffer 0 or write transmit data buffer 0

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

25.7.2.9. SCI 匹配地址寄存器（SCI_MATCH）

地址：SCIn_BASEADDR+0x0000_0020

	31	30	29	28	27	26	25	24
读:	保留						MA2[9:8]	
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	MA2[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	保留						MA1[9:8]	
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	MA1[7:0]							
sbr								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 25-11：SCI 匹配地址寄存器（SCI_MATCH）

MA2[9:0] — Match Address 2

当设置了最显着的位并设置了相关的 BAUD[MAEN]位时，将 MA1 和 MA2 寄存器与输入数据地址进行比较。如果发生匹配，以下数据将被传输到数据寄存器。如果匹配失败，则将丢弃以下数据。仅在关联的 BAUD[MAEN]位被清除时，软件才应写入 MA 寄存器。

MA1[9:0] — Match Address 1

当设置了最显着的位并设置了相关的 BAUD[MAEN]位时，将 MA1 和 MA2 寄存器与输入数据地址进行比较。如果发生匹配，以下数据将被传输到数据寄存器。如果匹配失败，则将丢弃以下数据。仅在关联的 BAUD[MAEN]位被清除时，软件才应写入 MA 寄存器。

©Levetop Semiconductor Co., Ltd.

25.7.2.10. SCI 调制解调器 IrDA 寄存器（SCI_MODIR）

地址：SCIn_BASEADDR+0x0000_0024

	31	30	29	28	27	26	25	24
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	保留					IREN	TNP	
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	RTSWATER							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	保留		TXCTSSRC	TXCTSC	RXRTSE	TXRTSPOL	TXRTSE	TXCTSE
sbr								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 25-12：SCI 调制解调器 IrDA 寄存器（SCI_MODIR）

IREN — Infrared enable

此位可启用/禁用红外线调制/解调。

1 = IR 启用

0 = IR 被禁用

TNP — Transmitter narrow pulse

该位允许 SCI 是否传输 1/OSR、2/OSR、3/OSR 或 4/OSR 窄脉冲。

0 0 = 1/OSR;
0 1 = 2/OSR;
1 0 = 3/OSR;
1 1 = 4/OSR

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

RTSWATER — Receive RTS Configuration

此位根据可以存储在接收 FIFO 中的附加字符数来配置 RX RTS 输出被否定的点。当配置为 0 时，当检测到将导致 FIFO 变满的字符的起始位时，RTS 为阴性。

1 = 当接收 FIFO 小于或接收 FIFO 等于 RXWATER 配置时，RTS 断言，当接收 FIFO 大于 RXWATER 配置时否定。

0 = 当接收方 FIFO 满或接收到导致 FIFO 满的字符时，RTS 断言。

TXCTSSRC — Transmit CTS Source

此位配置 CTS 输入的源。

1 = CTS 输入是反向接收器匹配结果

0 = CTS 输入为 SCI_CTS 引脚

TXCTSC — Transmit CTS Configuration

此位配置是否在每个字符开始时检查或仅在发送器空闲时检查 CTS 状态。

1 = 在发送器空闲时，对 CTS 输入进行采样

0 = CTS 输入在每个字符的开头进行采样

RXRTSE — Receiver request-to-send enable

这个位允许 RTS 输出控制发送设备的 CTS 输入，以防止接收器溢出。不要同时设置 RXRTSE 和 TXRTSE。

1 = 如果接收数据寄存器满或检测到起始位将导致接收数据寄存器满，则取消 RTS。如果接收数据寄存器没有满，并且没有检测到会导致接收数据寄存器变为满的开始位，则断言 RTS。RTS 断言由 RTSWATER 字段配置

0 = 接收器对 RTS 没有影响

TXRTSPOL — Transmitter request-to-send polarity

这个位控制发送器 RTS 的极性。TXRTSPOL 不影响接收器 RTS 的极性。在活动的低状态下，RTS 将保持否定，除非设置了 TXRTSE。

1 = 发送器 RTS 是高位动作

0 = 发送器 RTS 是低位动作

TXRTSE — Transmitter request-to-send enable

这个位控制传输前后的 RTS。

1 = 当一个字符被放入一个空的发送器数据缓冲区时，RTS 在传输起始位之前断言一个位时间。在发送器数据缓冲器和移位寄存器中的所有字符被完全发送后，RTS 去断言一位时间，包括最后一个停止位

0 = 发送器对 RTS 没有影响

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

TXCTSE — Transmitter clear-to-send enable

TXCTSE 控制变送器的操作。TXCTSE 可以独立于 TXRTSE 和 RXRTSE 的状态进行设置。

1 = 清除发送操作。每次它准备好发送一个字符时，发送器都会检查 CTS 的状态。如果断言了 CTS，则会发送该字符。如果断言 CTS，信号 TXD 保持标记状态，传输延迟，直到断言 CTS。发送的 CTS 的变化不影响其传输

0 = CTS 对发送器没有影响

25.7.2.11. SCI FIFO 寄存器（SCI_FIFO）

地址：SCIn_BASEADDR+0x0000_0028

	31	30	29	28	27	26	25	24
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	TXEMPT	RXEMPT	保留				TXOF	RXUF
写:							w1c	w1c
复位:	1	1	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0	0	保留	Rxiden			TXOFE	RXUFE
写:	TXFLUSH	RXFLUSH						
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	TXFE	TXFIFOSIZE[2:0]			RXFE	RXFIFOSIZE[2:0]		
写:								
复位:	0	0	1	0	0	0	1	0

= 写入没有影响，访问终止，没有传输错误异常。

w1c = 写 1 到该位将清除它

图 25-13：SCI FIFO 寄存器（SCI_FIFO）

TXEMPT — Transmit Buffer/FIFO Empty

当传输 FIFO/缓冲区中没有数据时，此位断言。此字段不考虑包含在传输移位寄存器中的数据。

1 = 传输缓冲区为空

0 = 传输缓冲区不为空

RXEMPT — Receive Buffer/FIFO Empty

当接收 FIFO/缓冲区中没有数据时，此位断言。此字段不考虑包含在接收移位寄存器中的数据。

1 = 接收缓冲区为空

0 = 接收缓冲区不为空

©Levetop Semiconductor Co., Ltd.

TXOF — Transmitter Buffer Overflow Flag

这位表示写入传输缓冲区的数据超过了它所能容纳的数据。此字段将进行断言，而不管 TXOFE 的值如何。但是，只有在设置了 TXOFE 时，才会向主机发出一个中断。通过写入一个 1，可以清除此标志。

1 = 自上次清除该标志以来，至少发生了一次传输缓冲区溢出

0 = 自上次清除该标志以来，没有发生传输缓冲区溢出

RXUF — Receiver Buffer Underflow Flag

此位表示从接收缓冲区读取的数据比现有的数据更多。此字段将进行断言，而不管 RXUFE 的值如何。但是，只有在设置了 RXUFE 时，才会向主机发出一个中断。通过写入一个 1，可以清除此标志。

1 = 自上次清除该标志以来，至少发生了一个接收缓冲区下流

0 = 自上次清除该标志以来，没有发生接收缓冲区下流

TXFLUSH — Transmit FIFO/Buffer Flush

写入此字段会导致存储在传输 FIFO/缓冲区中的所有数据被刷新。这不会影响传输移位寄存器中的数据。

1 = 传输 FIFO/缓冲区中的所有数据都将被清除

0 = 不发生冲洗操作

RXFLUSH — Receive FIFO/Buffer Flush

写入此字段会导致存储在接收 FIFO/缓冲区中的所有数据被刷新。这不会影响接收移位寄存器中的数据。

1 = 接收的 FIFO/缓冲区中的所有数据都将被清除

0 = 不发生冲洗操作

RXIDEN — Receiver Idle Empty Enable

当设置时，当接收器空闲了一些空闲字符且 FIFO 不是空时，它将启用 RDRF 的断言。

000 = Disable RDRF assertion due to partially filled FIFO when receiver is idle;
001 = Enable RDRF assertion due to partially filled FIFO when receiver is idle for 1 character;
010 = Enable RDRF assertion due to partially filled FIFO when receiver is idle for 2 characters;
011 = Enable RDRF assertion due to partially filled FIFO when receiver is idle for 4 characters;
100 = Enable RDRF assertion due to partially filled FIFO when receiver is idle for 8 characters;
101 = Enable RDRF assertion due to partially filled FIFO when receiver is idle for 16 characters;
110 = Enable RDRF assertion due to partially filled FIFO when receiver is idle for 32 characters;
111 = Enable RDRF assertion due to partially filled FIFO when receiver is idle for 64 characters.

TXOFE — Transmit FIFO Overflow Interrupt Enable

当设置此字段时，TXOF 标志将生成一个对主机的中断。

1 = TXOF 标志生成一个对主机的中断

0 = TXOF 标志不会对主机产生中断

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

RXUFE — Receive FIFO Underflow Interrupt Enable

当设置此字段时，RXUF 标志将生成一个对主机的中断。

1 = TRXUF 标志生成一个对主机的中断

0 = RXUF 标志不会对主机产生中断

TXFE — Transmit FIFO Enable

设置此字段后，将启用传输缓冲区的内置 FIFO 结构。FIFO 结构的大小用 txfifoo 的大小表示。如果没有设置此字段，传输缓冲区作为深度为一个数据字的 FIFO 操作，而不管 txfifo 大小的值。在更改此字段之前，必须同时选择 CTRL[TE]和 CTRL[RE]。

1 = 传输 FIFO 已启用。缓冲区用 txfifo 大小表示

0 = 传输 FIFO 被禁用。缓冲区为深度 1。（遗留支持）

TXFIFOSIZE[2:0] — Transmit FIFO/Buffer Depth

此字段表示可以存储在传输缓冲区中的最大传输数据字数。此字段为只读状态。

000 = 传输 FIFO/缓冲区深度 = 1 个数据字;

001 = 传输 FIFO/缓冲区深度 = 4 个数据字;

010 = 传输 FIFO/缓冲区深度 = 8 个数据字;

011 = 传输 FIFO/缓冲区深度 = 16 个数据字;

100 = 传输 FIFO/缓冲区深度 = 32 个数据字;

101 = 传输 FIFO/缓冲区深度 = 64 个数据字;

110 = 传输 FIFO/缓冲区深度 = 128 个数据字;

111 = 传输 FIFO/缓冲区深度 = 256 个数据字。

RXFE — Receive FIFO Enable

设置此字段后，启用接收缓冲区的 FIFO 结构。FIFO 结构的大小由 rxfifo 大小场表示。如果未设置此字段，则接收缓冲区作为深度为一个数据字的 FIFO 操作，而不管 rxfifo 大小中的值如何。在更改此字段之前，必须同时选择 CTRL[TE]和 CTRL[RE]。

1 = 接收 FIFO 已启用。缓冲区是由 RXFIFOSIZE 表示的深度

0 = 接收 FIFO 已被禁用。缓冲区为深度 1。（Legacy support）

RXFIFOSIZE[2:0] — Receive FIFO/Buffer Depth

可以存储在接收缓冲区中的最大传输数据字数。此字段为只读状态。

000 = 接收 FIFO/缓冲区深度 = 1 个数据字;

001 = 接收 FIFO/缓冲区深度 = 4 个数据字;

010 = 接收 FIFO/缓冲区深度 = 8 个数据字;

011 = 接收 FIFO/缓冲区深度 = 16 个数据字;

100 = 接收 FIFO/缓冲区深度 = 32 个数据字;

©Levetop Semiconductor Co., Ltd.

101 = 接收 FIFO/缓冲区深度 = 64 个数据字;

110 = 接收 FIFO/缓冲区深度 = 128 个数据字;

111 = 接收 FIFO/缓冲区深度 = 256 个数据字;

25.7.2.12. SCI 水印寄存器（SCI_WATER）

地址：SCIn_BASEADDR+0x0000_002C

	31	30	29	28	27	26	25	24
读:	RXCOUNT[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	RXWATER[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	TXCOUNT[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	TXWATER[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 25-14：SCI 水印寄存器（SCI_WATER）

RXCOUNT[7:0] — Receive Counter

此寄存器中的值表示接收 FIFO/缓冲区中的数据字数。如果正在接收到一个数据字，即在接收移位寄存器中，则它不包括在计数中。此值可与 FIFO[RXFIFOSIZE]一起使用，以计算在接收 FIFO/缓冲区中还有多少空间。

RXWATER[7:0] — Receive Watermark

当接收 FIFO/缓冲区中的数据字数大于此寄存器字段中的值时，将生成一个中断。为了正确操作，RXWATER 中的值必须设置为小于 FIFO[FIFO 尺寸]和 FIFO[RXFE]所指示的接收 FIFO/缓冲器大小，并且必须大于 0。

TXCOUNT[7:0] — Transmit Counter

该寄存器中的值表示传输 FIFO/缓冲区中的数据字数。如果正在传输一个数据字，即在传输移位寄存器中，则它不包括在计数中。此值可与 FIFO[TXFIFOSIZE]一起使用，以计算在传输的 FIFO/缓冲区中还有多少空间。

TXWATER[7:0] — Transmit Watermark

当发送 FIFO/缓冲区中的数据字数等于或小于该寄存器字段中的值时，将生成一个中断。为了正确操作，TXWATER 中的值必须设置为小于传输缓冲区的大小/由[FIFO 大小]和 FIFO 大小[TXFE]所示的 FIFO 大小。

©Levetop Semiconductor Co., Ltd.

25.7.2.13. SCI 过采样比寄存器（SCI_OSR）

地址：SCIn_BASEADDR+0x0000_0030

	31	30	29	28	27	26	25	24
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	保留							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	OSR[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 25-15：SCI 过采样比寄存器（SCI_OSR）

OSR[7:0] — Oversampling Ratio

该字段为接收器配置 4x（00000011）和 256x（111111111）之间的过采样比。写入无效的过采样比（例如，不介于 4x 和 256x 之间的值）将默认为过采样比 16（00001111）。只有当发送器和接收器都被禁用时，OSR 字段。过采样比 = OSR + 1。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

25.8. 功能描述

SCI 支持全双工、异步和 NRZ 串行通信，并包括一个波特率发生器、发送器和接收器块。发送器和接收器独立工作，尽管它们使用相同的波特率发生器。下面描述了 SCI 的每个方块。

25.9. 波特率生成

波特率发生器中的一个 13 位模量计数器推导出接收器和发送器的波特率。从 1 到写入 SBR[12:0]的 8191 的值确定了异步 SCI 波特时钟的波特时钟除数。SBR 位在 SCI 波特率寄存器、BDH 和 BDL 中。波特率时钟驱动接收器，而发送器则由波特率时钟除以过采样比来驱动。根据过采样比，接收器的采集速率为 4 到 256 个样本。

波特率的产生有两个误差来源：

- 异步 SCI 波特时钟的整数除法可能不能给出准确的目标频率。
- 与异步 SCI 波特时钟的同步可能会导致相移

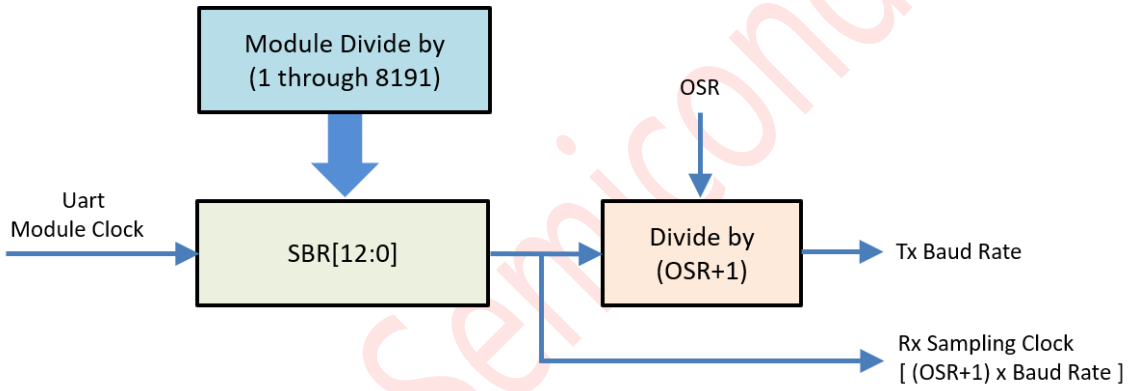


图 25-16：SCI 波特率生成

$$\text{Baud Rate} = \frac{\text{Uart Module Clock}}{\text{SBR[12:0]} \times (\text{OSR+1})}$$

提示：如果 SBR[12:0] = 0，则关闭波特率生成器。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

25.10. 发送器功能描述

本节描述了 SCI 发送器的整体方块图，以及用于发送中断和空闲字符的专门功能。

变送器输出（TXD）空闲状态默认为逻辑高，CTRL[TXINV]在复位后被清除。通过设置 CTRL[TXINV]来反转发送器输出。通过设置 CTRL[TE]位来启用发送器。这将为一个属于空闲状态的完整字符框架的前导字符进行排队。然后发送器保持空闲，直到发射数据缓冲区中的数据可用。程序通过写入 SCI 数据寄存器，将数据存储到传输数据缓冲区中。

SCI 发送器的中心组件是发射移位寄存器，它从 10 位到 13 位长，这取决于 CTRL[M]、BAUD[M10]和 BAUD[SBNS]控制位的设置。对于本节的其余部分，假定 CTRL[M]、BAUD[M10]和 BAUD[SBNS]已被清除，并选择正常的 8 位数据模式。在 8 位数据模式中，移位寄存器包含一个起始位、8 个数据位和一个停止位。当传输移位寄存器可用于新字符时，在传输数据寄存器中等待的值被传输到移位寄存器，与波特率时钟同步，并且发送数据寄存器空（STAT[TDRE]）状态标志被设置为指示另一个字符可以写入传输数据缓冲区。

如果在停止位移出 TXD 引脚后，没有在传输数据缓冲区中等待新字符，则发送机设置发送完整标志并进入空闲模式，TXD 高，等待更多字符传输。

将 0 写入 CTRL[TE]并不会立即禁用发送器。当前正在进行的传输活动必须首先完成（这可能包括一个数据字符、空闲字符或中断字符），尽管发送器将不会开始传输另一个字符。

25.10.1. 发送休息和排队的空闲时间

SCI_CTRL[SBK]位发送中断字符，最初用于获得旧电传类型接收器的注意。中断字符是逻辑 0 的完整字符时间，10 位到 12 位的时间，包括开始位和停止位。通过设置 SCI_STAT[BRK13]，可以启用更长的 13 位时间的中断。通常，程序会等待 SCI_STAT[TDRE]设置为指示消息的最后一个字符已移动到传输移位器，写入 1，然后将 0 写入 SCI_CTRL[SBK]位。一旦转换器可用，此动作将排队排列要发送的中断字符。如果 SCI_CTRL[SBK]在排队中断移动到移位器时保持 1，同步到波特率时钟，则另一个中断字符排队。如果接收设备是另一个飞思卡尔半导体 SCI，则在所有数据位中接收到中断字符为 0s，并发生帧错误（SCI_STAT[FE] = 1）。

还可以通过写入设置为 13 位和数据位被清除的 SCI_DATA 寄存器来传输中断字符。这支持将中断字符作为正常数据流的一部分进行传输。

当使用空闲行唤醒时，消息之间需要整个空闲字符时间（逻辑 1）来唤醒任何休眠的接收器。通常，程序会等待 SCI_STAT[TDRE]设置为指示消息的最后一个字符已移动到传输移位器，然后写入 0，然后将 1 写入 SCI_CTRL[TE]位。一旦转换器可用，此操作将立即发送空闲字符。只要在 SCI_CTRL[TE]被清除时，移位器中的字符没有完成，SCI 发送器就不会真正释放对 TXD 引脚的控制。

空闲字符也可以通过将第 13 位和数据位写入 SCI_DATA 寄存器来传输。这支持将空闲字符作为正常数据流的一部分进行传输。

断裂字符的长度受到 SCI_STAT[BRK13]、SCI_CTRL[M]、SCI_BAUD[M10]和 SCI_BAUD[SNBS]位的影响，如下图所示。

©Levetop Semiconductor Co., Ltd.

表 25-3：断开字符长度

BRK13	M	M10	SBNS	Break Character Length
0	0	0	0	10-bits times
0	0	0	1	11-bits times
0	1	0	0	11-bits times
0	1	0	1	12-bits times
0	X	1	0	12-bits times
0	X	1	1	13-bits times
1	0	0	0	13-bits times
1	0	0	1	13-bits times
1	1	0	0	14-bits times
1	1	0	1	14-bits times
1	X	1	0	15-bits times
1	X	1	1	15-bits times

25.11. 接收器功能说明

在本节中，接收器方块图是整体接收器功能描述的指南。接下来，将更详细地描述了用于重构接收数据的数据采样技术。最后，解释了接收器唤醒功能的不同变化。

通过设置 SCI_STAT[RXINV]来反转接收器输入。通过设置 SCI_CTRL[RE]位来启用接收器。字符帧由逻辑 0 的起始位、8 到 10 个数据位（MSB 或 LSB 优先）和逻辑 1 的一个或两个停止位组成。有关 9 位或 10 位数据模式的信息，请参阅 8 位、9 位和 10 位的数据模式。在本讨论的其余部分中，假设 SCI 被配置为正常的 8 位数据模式。

在接收到停止位进入接收移动器后，如果接收数据寄存器尚未满，则将数据字符传输到接收数据寄存器，并设置接收数据寄存器满（SCI_STAT[RDRF]）状态标志。如果 SCI_STAT 已设置[RDRF]，则表示接收数据寄存器（缓冲区）已满，已设置溢出（OR）状态标志，且新数据丢失。因为 SCI 接收器是双缓冲的，所以在必须读取接收数据缓冲器中的数据之前，在设置了 SCI_STAT[RDRF]之前，程序有一个完整的字符时间，以避免接收器溢出。

当程序检测到接收数据寄存器已满时（SCI_STAT[RDRF] = 1），它通过读取 SCI_DATA 从接收数据寄存器获取数据。有关标志清除的详细信息，请参阅中断标志和状态标志。

25.11.1. 数据采样技术

SCI 接收器支持 4×到 256×的波特率时钟的可配置过采样率。接收器首先以过采样率乘以波特率进行逻辑级采样，以搜索 RXD 串行数据输入引脚上的下降边。一个下降边被定义为经过三个连续的逻辑 1 样本之后的逻辑 0 样本。过采样波特率时钟将比特时间从 1 到 OSR 分成 4 到 256 段（其中 OSR 是配置的过采样比）。当找到一个下降的边缘时，将在（OSR/2）、（OSR/2）+1 和（OSR/2）+2 处再采集三个样本，以确保这是一个真正的起始位，而不仅仅是噪声。如果这三个样本中至少有两个为 0，则接收方假定它与接收到的字符同步。如果在认为接收器同步之前检测到另一个下降边，则接收器从第一段重启采样。

©Levetop Semiconductor Co., Ltd.

然后，接收器在（OSR/2）、（OSR/2）+1 和（OSR/2）+2 处采样每个位时间，包括开始位和停止位，以确定该位的逻辑级别。逻辑级别被解释为在比特时间内采集的大部分样本。如果在字符帧中的任何位时间，包括开始位和停止位，与该位的逻辑级别不一致，当接收字符传输到接收数据缓冲区时设置噪声标志（SCI_STAT[NF]）。

当 SCI 接收器被配置为在波特率时钟的两边进行采样时，每个接收位中的段数有效地增加了一倍（从 1 到 OSR×2）。然后在 OSR、OSR+1 和 OSR+2 处对起始位和数据位进行采样。对于 4×到 7×的过采样率，必须启用时钟两侧的采样，并且对于更高的过采样率是可选的。

下降边检测逻辑不断地寻找下降边。如果检测到边缘，则样本时钟将重新同步到位时间（除非已禁用重新同步）。这提高了接收器在存在噪声或不匹配的波特率的情况下的可靠性。它不会改善最坏情况分析，因为某些字符在字符框架中的任何地方都没有任何额外的下降边。

在帧错误的情况下，如果接收到的字符不是断开字符，则搜索下降边的采样逻辑充满三个逻辑 1 样本，以便几乎可以立即检测到新的开始位。

25.11.2. 接收器唤醒操作

接收器唤醒和接收器地址匹配是一种硬件机制，它允许 SCI 接收器忽略针对不同接收器的消息中的字符。

在接收器唤醒期间，所有接收器评估每个消息的第一个字符(s)，一旦他们确定消息是针对不同的接收器，他们将逻辑 1 写入接收器唤醒控制位（SCI_CTRL[RWU]）。当设置 RWU 位和 SCI_S2[RWUID]位时，与接收器相关的状态标志，除了空闲位的空闲位，被禁止设置，从而消除了处理不重要消息字符的软件开销。在消息结束或下一个消息的开始，所有接收器自动强制 SCI_CTRL[RWU]为 0，以便所有接收器及时醒来以查看下一个消息的第一个字符(s)。

在接收器地址匹配过程中，在硬件中进行地址匹配，SCI 接收器将忽略所有不满足地址匹配要求的字符。

断裂字符的长度受到 SCI_STAT[BRK13]、SCI_CTRL[M]、SCI_BAUD[M10]和 SCI_BAUD[SNBS]位的影响，如下图所示。

表 25-4：接收器唤醒选项

RWU	MA1 | MA2	MATCFG	WAKE: RWUID	Receiver Wakeup
0	0	X	X	Normal operation
1	0	00	00	Receiver wakeup on idle line, IDLE flag not set
1	0	00	01	Receiver wakeup on idle line, IDLE flag set
1	0	00	10	Receiver wakeup on address mark
1	1	11	X0	Receiver wakeup on data match
0	1	00	X0	Address mark address match, IDLE flag not set for discarded

©Levetop Semiconductor Co., Ltd.

RWU	MA1 | MA2	MATCFG	WAKE: RWUID	Receiver Wakeup
				characters
0	1	00	X1	Address mark address match, IDLE flag set for discarded characters
0	1	01	X0	Idle line address match
0	1	10	X0	Address match on and address match off, IDLE flag not set for discarded characters
0	1	10	X1	Address match on and address match off, IDLE flag set for discarded characters

25.11.2.1. 怠速排队

当唤醒被清除时，接收器被配置为空闲线路唤醒。在此模式下，当接收方检测到空闲线级别的完整字符时间时，将自动清除 SCI_CTRL[RWU]。SCI_CTRL[M]和 SCI_BAUD[M10]控制位选择 8 位到 10 位数据模式，SCI_BAUD[SBNS]位选择 1 位或 2 位停止位数，决定需要多少位空闲，因为开始位和停止位需要 10 到 13 位。

当 SCI_CTRL[RWU]为 1 且 SCI_STAT[RWUID]为零时，唤醒接收方的空闲条件不会设置 SCI_STAT[IDLE]标志。接收方唤醒并等待下一个消息的第一个数据字符，该消息设置了 SCI_STAT[RDRF]标志，并在启用时生成一个中断。当 SCI_STAT[RWUID]为 1 时，任何空闲条件都会设置 SCI_STAT[IDLE]标志，并在启用时生成一个中断，而不管 SCI_CTRL[RWU]为 0 还是 1。

空闲线类型（SCI_CTRL[ILT]）控制位选择两种方法之一来检测空闲线。当 SCI_CTRL[ILT]被清除时，空闲位计数器在开始位之后开始，因此停止位和字符结束计数时的任何逻辑 1 接近空闲的全部字符时间。当设置 SCI_CTRL[ILT]时，空闲位计数器直到停止位时间之后才开始，因此空闲检测不受上一条消息最后一个字符中的数据的影响。

25.11.2.2. 地址标记唤醒

设置 SCI_CTRL[WAKE]时，接收器配置为地址标记唤醒。在此模式下，当接收方检测到接收字符的最重要位中的逻辑 1 时，SCI_CTRL 将自动被[RWU]清除。

地址标记唤醒允许消息包含空闲字符，但需要保留 MSB 以便在地址帧中使用。地址帧的 MSB 中的逻辑 1 在接收到停止位之前清除 SCI_CTRL[RWU]位，并设置 SCI_STAT[RDRF]标志。在这种情况下，即使接收器在字符大部分时间睡觉，MSB 集的字符也会被接收。

25.11.2.3. 数据匹配唤醒

当设置了 SCI_CTRL[RWU]并且 SCI_BAUD[MATCFG]等于 11 时，接收器被配置为数据匹配唤醒。在此模式下，当接收方检测到设置 BAUD[MAEN1]时匹配[MA1]字段，或设置 BAUD[MAEN2]时匹配[MA2]的字符时，将自动清除 SCI_CTRL[RWU]。

©Levetop Semiconductor Co., Ltd.

25.11.2.4. 地址匹配操作

当设置了 SCI_BAUD[MAEN1]或 SCI_BAUD[MAEN2]位，并且 SCI_BAUD[MATCFG]等于 00 时，将启用地址匹配操作。在这个功能中，RXD 引脚通过逻辑 1 接收到的一个字符被认为是一个地址，并与相关的匹配[MA1]或匹配[MA2]字段进行比较。如果比较匹配，字符只转移到接收缓冲区，并设置 SCI_STAT[RDRF]。在停止位之前的位位置用逻辑 0 接收的所有后续字符被认为是与地址相关联的数据，并被传输到接收数据缓冲区。如果没有标记的地址匹配，则不会传输到接收数据缓冲区，并且在停止位之前的位位置的逻辑为零的以下字符也将被丢弃。如果 SCI_BAUD[MAEN1]和 SCI_BAUD[MAEN2]位都被否定，则接收器将正常工作，并且接收到的所有数据都将被传输到接收数据缓冲区。

匹配[MA1]和匹配[MA2]字段的相同方式的地址匹配操作功能。

- 如果只断言 SCI_BAUD[MAEN1]和 SCI_BAUD[MAEN2]中的一个，则只将标记的地址与关联的匹配寄存器进行比较，并且仅在匹配时将数据传输到接收数据缓冲区。
- 如果断言 SCI_BAUD[MAEN1]和 SCI_BAUD[MAEN2]，则将标记的地址与两个匹配寄存器进行比较，并且仅在与任何一个寄存器的匹配上传输数据。

25.11.2.5. 空闲匹配操作

当设置了 SCI_BAUD[MAEN1]或 SCI_BAUD[MAEN2]位，并且 SCI _BAUD[MATCFG]等于 01 时，将启用空闲匹配操作。在这个功能中，在空闲行条件之后由 RXD 引脚接收到的第一个字符被认为是一个地址，并与关联的 MA1 或 MA2 寄存器进行比较。如果比较匹配，字符只转移到接收缓冲区，并设置 SCI_STAT[RDRF]。所有后续字符都被认为是与地址关联的数据，并被传输到接收数据缓冲区，直到检测到下一个空闲行状态。如果没有地址匹配，则不会向接收数据缓冲区传输，在下一个空闲状态之前的所有帧也将被丢弃。如果 SCI_BAUD[MAEN1]和 SCI_BAUD[MAEN2]位都被否定，则接收器将正常工作，并且接收到的所有数据都将被传输到接收数据缓冲区。

对于 MA1 和 MA2 寄存器，以相同的方式操作。

- 如果只断言 SCI_BAUD[MAEN1]和 SCI_BAUD[MAEN2]中的一个，则只将空闲行之后的第一个字符与关联的匹配寄存器进行比较，并且仅在匹配时将数据传输到接收数据缓冲区。
- 如果断言 SCI_BAUD[MAEN1]和 SCI_BAUD[MAEN2]，则将空闲行之后的第一个字符与两个匹配寄存器进行比较，并且只在与任何一个寄存器的匹配上传输数据。

25.11.2.6. 匹配匹配关闭操作

当设置 SCI_BAUD[MAEN1]和 SCI_BAUD[MAEN2]且 SCI_BAUD[MATCFG]等于 10 时，将启用“匹配打开，匹配关闭”操作。在此功能中，接收到匹配匹配的 RXD 引脚[MA1]的字符并将其转移到接收缓冲区，并设置 SCI_STAT[RDRF]。所有后续字符都被认为是数据，也被传输到接收数据缓冲区，直到接收到与匹配[MA2]寄存器匹配的字符。丢弃匹配[MA2]的字符，直到收到另一个匹配[MA1]匹配的字符。如果 SCI_BAUD[MAEN1]和 SCI_BAUD[MAEN2]位都被否定，则接收器将正常工作，并且接收到的所有数据都将被传输到接收数据缓冲区。

©Levetop Semiconductor Co., Ltd.

25.11.3. 红外解码器

红外解码器将接收到的字符从 IrDA 格式转换为接收器使用的 NRZ 格式。它也有一个 OSR 过采样波特率时钟计数器，过滤噪声，并指示何时接收到 1。

25.11.3.1. 开始位检测

当清除 STAT[RXINV]时，接收字符的第一个下降边对应于起始位。红外解码器复位其计数器。此时，接收器也开始了它的开始位检测过程。在检测到开始位后，接收器将其位时间同步到此开始位时间。对于其余的字符接收，红外解码器的计数器和接收器的位时间计数器的计数相互独立。

25.11.3.2. 噪声滤波

在红外解码器计数器的前半部分中检测到的任何进一步上升的边缘都被解码器忽略。任何小于一个过采样波特时钟的脉冲都不能被它检测到，无论它是在计数的前半部分或后半部分被看到。

25.11.3.3. 低位检测

在解码器计数的后半部分，一个上升的边被解码为一个 0，它被发送到接收器。解码器计数器也被复位。

25.11.3.4. 高比特检测

在 OSR 的过采样波特率时钟后，如果没有看到一个上升的边缘，则解码器向接收器发送一个 1。

如果下一位是 0，到达较晚，则根据低位检测检测到低位。发送到接收方的值将从 1 更改为 0。然后，如果噪声脉冲发生在接收器的比特时间采样周期之外，则 0 的延迟不被记录为噪声。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

25.12. 额外的 SCI 功能

下面的部分描述了额外的 SCI 功能。

25.12.1. 8 位、9 位和 10 位的数据模式

通过设置 SCI_ CTRL[M]或通过设置 SCI_ BAUD[M10]的 10 位数据模式，SCI 发送器和接收器可以配置为以 9 位数据模式运行。在 9 位模式下，有第九个数据位，而在 10 位模式下，有第 10 个数据位。对于传输数据缓冲区，这些位被存储在 SCI_CTRL[T8]和 SCI_CTRL[T9]中。对于接收器，这些位被保存在 SCI_CTRL[R8]和 SCI_CTRL[R9]中。它们也可以通过对 SCI_DATA 寄存器的 16 位或 32 位的访问来访问。

对于对传输数据缓冲区的相干 8 位写入，请在写入 SCI_DATA[7:0]之前写入 SCI_CTRL[T8]和 SCI_CTRL[T9]。对于对 SCI_DATA 寄存器的 16 位和 32 位写入，所有 10 个传输位同时被写入传输数据缓冲区。

如果作为新字符的第九位和第十位传输的位值与前一个字符的位值相同，则无需再次写入 SCI_CTRL[T8]和 SCI_CTRL[T9]。当数据从传输数据缓冲区传输到传输移位器时，将复制 SCI_CTRL[T8]和 SCI_CTRL[T9]中的值，同时将数据从 SCI_DATA[7:0]传输到移位器。

9 位数据模式通常与奇偶校验一起使用，以允许 8 位数据加上第九位的奇偶校验，或者它与地址标记唤醒一起使用，因此第九位数据位可以作为唤醒位。10 位数据模式通常与奇偶校验和地址标记唤醒一起使用，因此第九数据位可以作为唤醒位，第十数据位可以作为奇偶校验位。在自定义协议中，第九位和/或第十位也可以作为软件控制的标记。

25.12.2. 空闲长度

空闲字符是指起始位、所有数据位和停止位都处于标记位置的字符。CTRL[ILT]寄存器可以配置为从前一开始位（指向空闲字符检测的任何数据位和停止位计数）或从前一停止位开始检测空闲字符。

还可以使用 CTRL[IDLECFG]字段对在检测到空闲行条件之前必须接收到的空闲字符数进行配置。此字段配置在设置 STAT[IDLE]标志、清除 STAT[RAF[RAF]标志和 DATA[IDLINE]标志使用下一个接收字符设置之前必须接收的空闲字符数。

空闲线唤醒和空闲匹配操作也会受到 CTRL 字段的影响。当启用地址匹配或匹配开/关操作时，设置 STAT[RWUID]位将导致将丢弃的字符视为空闲字符。独立于外部系统中的连接，以帮助隔离系统问题。在这种模式下，发送器输出内部连接到接收器输入，RXD 引脚不使用 SCI。

25.12.3. 单线操作

当设置 SCI_CTRL[循环]时，同一寄存器中的 SCI_CTRL[RSRC]位会在循环模式（SCI_CTRL[RSRC] = 0）和单线模式(SCI_CTRL[RSRC] = 1]之间进行选择。循环模式有时用于检查软件，独立于外部系统中的连接，以帮助隔离系统问题。在这种模式下，发送器输出内部连接到接收器输入，RXD 引脚不使用 SCI。

©Levetop Semiconductor Co., Ltd.

25.12.4. 循环模式

当设置了 SCI_CTRL[循环]时，在同一寄存器中的 RSRC 位会在循环模式（SCI_CTRL[RSRC] = 0）或单线模式（SCI_CTRL[RSRC] = 1）之间进行选择。单线模式实现了一个半双工串行连接。接收器内部连接到发送器输出和 TXD 引脚（不使用 RXD 引脚）。

在单线模式下，SCI_CTRL[TXDIR]位控制 TXD 引脚上的串行数据的方向。当 SCI_CTRL[TXDIR]被清除时，TXD 引脚是接收器的输入，发送器暂时与 TXD 引脚断开，以便外部设备可以向接收器发送串行数据。当设置 SCI_CTRL[TXDIR]时，TXD 引脚是由发送器驱动的输出，内环回连接被禁用，因此接收器无法接收发送器发送的字符。

25.13. 红外线接口

SCI 提供了将窄脉冲传输到红外 LED 并接收窄脉冲并将其转换为串行位的能力，并被发送到 SCI 的能力。IrDA 物理层规范定义了一个用于交换数据的半双工红外通信链路。完整的标准包括高达 16 Mbits/s 的数据速率。该设计仅涵盖了 2.4 kbits/s 到 115.2 kbits/s 之间的数据速率。

SCI 有一个红外发射编码器和接收解码器。SCI 传输由红外子模块编码的串行数据位，每一个零位传输一个窄脉冲。每一位不发送脉冲。当接收数据时，使用红外光二极管检测红外脉冲，并由来自 SCI 的外部的红外接收解码器转换到 CMOS 水平。然后，窄脉冲由红外接收解码器拉伸，以返回到要由 UART 接收的串行比特流。传输脉冲和预期接收脉冲的极性可以反向，从而可以直接连接到使用有源高脉冲的外部 IrDA 收发器模块。

红外线子模块从 SCI 接收其时钟源。在红外子模块中选择这两个时钟中的一个，以在传输过程中产生 1/OSR、2/OSR、3/OSR 或 4/OSR 的窄脉冲。

25.13.1. 红外发射编码器

红外发射编码器将数据从发射移位寄存器的串行位转换到 TXD 信号。一个窄脉冲传输零位，没有脉冲。窄脉冲在比特开始时发送，其持续时间为 1/OSR、2/OSR、3/OSR 或 4/OSR 的比特时间。当清除 SCI_CTRL[TXINV]时，发送零位窄低脉冲，而当设置 SCI_CTRL[TXINV]时，为零位发送窄高脉冲。

25.13.2. 红外接收解码器

红外接收块将来自 RXD 信号的数据转换到接收移位寄存器。每个接收到一个零都期望有一个窄脉冲，每个接收到一个都期望没有脉冲。设置零位 whenSCI_STAT[RXINV]时，预计清除窄低脉冲，而设置 SCI_STAT[RXINV]时，预计零位清除窄高脉冲。该接收解码器满足 IrDA 串行红外物理层规范所定义的边缘抖动要求。

©Levetop Semiconductor Co., Ltd.

25.14. 中断和状态标志

SCI 发送器有两个状态标志，可以选择性地生成硬件中断请求。发送数据寄存器为空的 SCI_（STAT[TDRE]）位表示发送数据缓冲区中是否有空间将另一个发送字符写入 SCI_DATA。如果设置了传输中断启用 SCI_CTRL[TIE])位，则在设置 SCI_STAT[TDRE]时请求硬件中断。传输完成（SCI_STAT[TC]）位表示传输机完成了所有数据、前导和断开字符的传输，且 TXD 处于非活动级别。这个标志通常用于具有调制解调器的系统，以确定何时关闭调制解调器是安全的。如果设置了发送完全中断启用（SCI_CTRL[TCIE]）位，则在设置 SCI_STAT[TC]时请求硬件中断。如果清除了相应的 SCI_CTRL[TIE]或 SCI_CTRL[TCIE]本地中断掩码，则可以使用软件轮询来监视 SCI_STAT[TDRE]和 SCI_STAT[TC]状态标志，而不是硬件中断。

当程序检测到接收数据寄存器已满时（SCI_STAT[RDRF] = 1），它通过读取 SCI_DATA 从接收数据寄存器获取数据。通过读取 SCI_DATA，将清除 SCI_STAT[RDRF]标志。

IDLE 状态标志包含了防止在 RXD 行长时间空闲时重复设置它的逻辑。通过将 1 写入 SCI _STAT[IDLE]标志来清除 IDLE。清除 SCI_STAT[IDLE]后，在接收方接收到至少一个新字符并设置了 SCI_STAT[RDRF]之前，才能再次设置它。

如果在接收到的字符中检测到关联的错误，导致设置 SCI_STAT[RDRF]，则错误标志-噪声标志（SCI_STAT[NF]）、帧错误（SCI_STAT[FE]）和奇偶校验错误标志（SCI_STAT[PF]）-与 SCI_STAT[RDRF]同时设置。这些标志不会在溢出的情况下设置。

如果当一个新字符准备从接收移位器传输到接收数据缓冲区时，已经设置了 SCI_STAT[RDRF]，则将设置溢出（SCI_STAT[OR]）标志，而不是伴随任何相关的 NF、FE 或 PF 条件的数据。

如果接收到的字符与匹配[MA1]和/或匹配[MA2]的内容匹配，则在设置 SCI_STAT[MA1F]和/或 SCI_STAT[RDRF]的同时设置 SCI_STAT[MA2F]标志。

在任何时候，RXD 串行数据输入引脚上的活动边都会导致设置 SCI_STAT[SCI_STAT]标志。通过写入 1 来清除 SCI_STAT 标志。此功能取决于已启用的接收器（SCI_CTRL[RE] = 1）。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.同步串行接口（SSI）

26.1. 功能介绍

SSI 是一个可编程的同步串行接口（SSI）外设。这是一个符合 AMBA2.0 标准的 AHB（高级高性能总线）组件。主机处理器通过 AHB 接口访问 SSI 上的数据、控制和状态信息。SSI 还可以使用一组可选的 DMA 信号与 DMA 控制器进行接口，该信号可以在配置期间进行选择。

LT168 的 SSI 模块提供了到 SPI Flash 或其他 SPI 设备的 QSPI0、QSPI1 和 QSPI2 接口。QSPI0 信号直接连接到嵌入式 SPI 闪存上，QSPI1 和 QSPI2 信号可供用户使用。

26.2. 串行接口特点

串行接口包括：

- 串行主操作。
- DMA 控制器接口-这使得 SSI 能够通过使用针对传输请求的握手接口，通过总线与 DMA 控制器进行接口。
- 时钟拉伸支持在增强的 SPI 传输。
- 数据项大小（4 到 32 位）—在程序员控制下的每个数据传输的项目大小。
- FIFO 深度-发射和接收 FIFO 缓冲区是 8 个字的深度。FIFO 的宽度固定在 32 位。
- 增强的 SPI 支持。
- 就地执行（XIP）模式支持。

26.3. 操作模式

SSI 在这三种模式下发挥作用：

1. 运行模式 - 运行模式是正常的操作模式。
2. 打盹模式 - 打盹模式是一种可配置的低功耗模式。
3. 停止模式 - SSI 在停止模式下处于非活动状态。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.4. 方块图

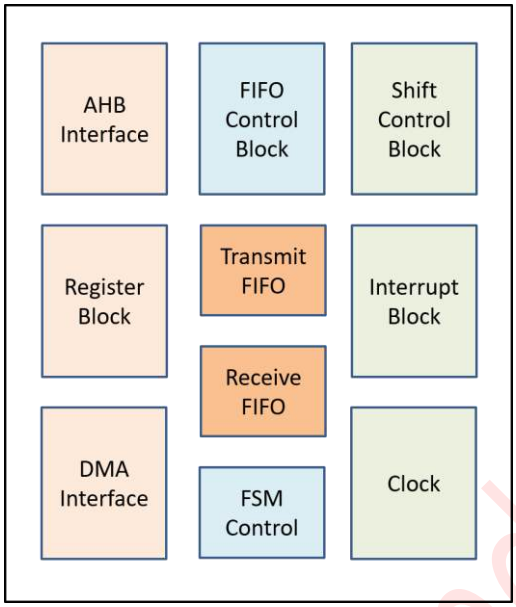


图 26-1：SSI 方块图

26.5. 应用图

以下是 LT168 的 SSI 模块的应用程序。

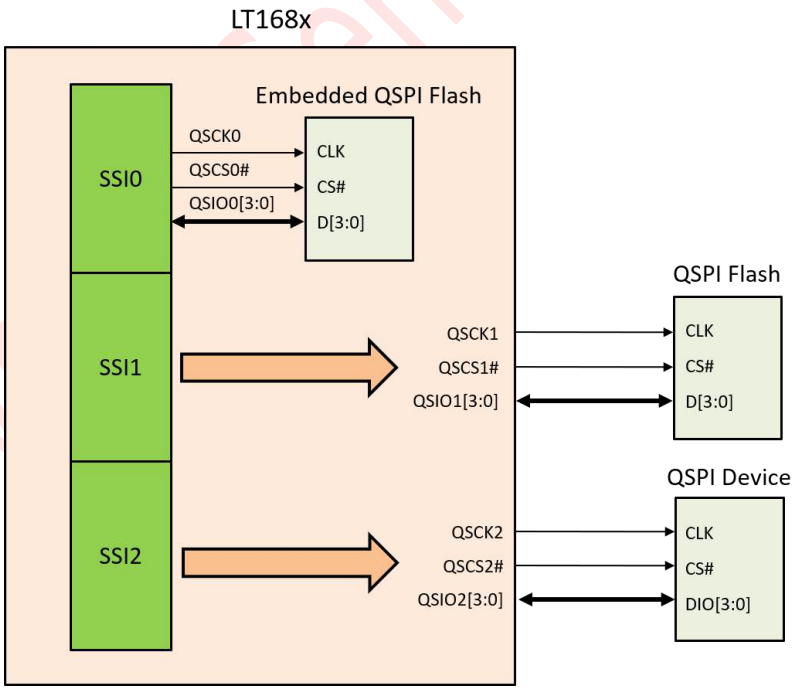


图 26-2：SSI 应用图

©Levetop Semiconductor Co., Ltd.

26.6. 内存映射和寄存器

26.6.1. 内存映射

SSI 模块内存映射如表 26-1 所示.QSPI0 的基本地址为 0x6000_0000，QSPI1 为 0x7000_0000，QSPI2 为 0x8000_0000。

表 26-1：SSI 内存映射

偏移地址	Bits[31:0]	访问权限
0x0000	Control Register 0 (CTRLR0)	S/U
0x0004	Control Register 1 (CTRLR1)	S/U
0x0008	SSI Enable Register (SSIENR)	S/U
0x000C	Microwire Control Register (MWCR)	S/U
0x0010	Slave Select Register (SER)	S/U
0x0014	Baud Rate Select Register (BAUDR)	S/U
0x0018	Transmit FIFO Threshold Level Register (TXFTLR)	S/U
0x001C	Receive FIFO Threshold Level Register (RXFTLR)	S/U
0x0020	Transmit FIFO Level Register (TXFLR)	S/U
0x0024	Receive FIFO Level Register (RXFLR)	S/U
0x0028	Status Register (SR)	S/U
0x002C	Interrupt Mask Register (IMR)	S/U
0x0030	Interrupt Status Register (ISR)	S/U
0x0034	Raw Interrupt Status Register (RISR)	S/U
0x0038	Transmit FIFO Overflow Interrupt Clear Register (TXOICR)	S/U
0x003C	Receive FIFO Overflow Interrupt Clear Register (RXOICR)	S/U
0x0040	Receive FIFO Underflow Interrupt Clear Register (RXUICR)	S/U
0x0048	Interrupt Clear Register (ICR)	S/U
0x004C	DMA Control Register (DMACR)	S/U
0x0050	DMA Transmit Data Level Register (DMATDLR)	S/U
0x0054	DMA Receive Data Level Register (DMARDLR)	S/U
0x0058	Identification Register (IDR)	S/U
0x005C	Version ID Register (VIDR)	S/U
0x0060+i*0x4	SSI Data Register (DRx)	S/U
0x00F0	RX Sample Delay Register (RXSDR)	S/U
0x00F4	SPI Control Register 0 (SPICTRLR0)	S/U
0x00FC	XIP Mode Bits (XIPMBR)	S/U
0x0100	XIP Incr Inst Register (XIPIIR)	S/U
0x0104	XIP Wrap Inst Register (XIPWIR)	S/U
0x0108	XIP Control Register (XIPCR)	S/U
0x010C	XIP Slave Enable Register (XIPSER)	S/U
0x0110	XIP Receive FIFO Overflow Interrupt Clear Register (XRXIOCR)	S/U
0x0114	XIP Continus Transfer Time Out Register (XIPCTTOR)	S/U

©Levetop Semiconductor Co., Ltd.

26.6.2. 寄存器描述

26.6.2.1. 控制寄存器 0（CTRLR0）

地址：QSPIn_BASEADDR+0x0000_0000

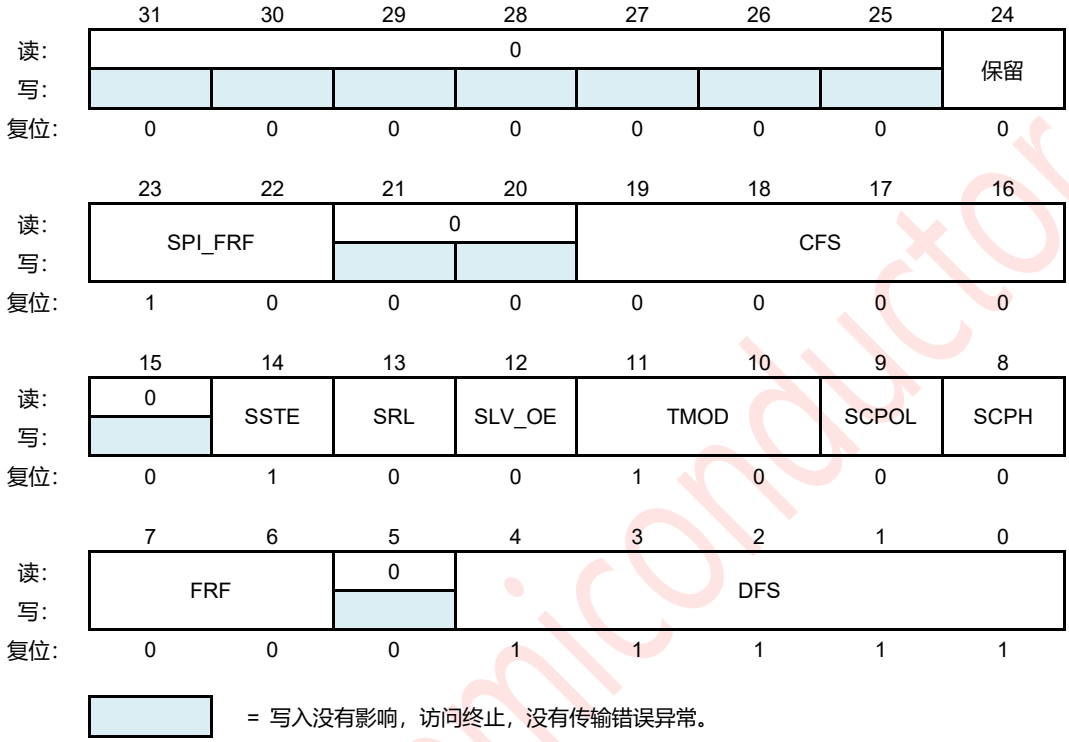


图 26-3：控制寄存器 0（CTRLR0）

此寄存器控制串行数据传输。当启用 SSI 时，不可能写入此寄存器。通过写入 SSIENR 来启用和禁用 SSI。

SPI_FRF — SPI Frame Format

仅当 SSIC_SPI_MODE 设置为“双”、“四”或“八度”模式时，选择“传输/接收数据位”的数据帧格式。

0x0（SPI_STANDARD）：标准 SPI 格式
0x1（SPI_DUAL）：双 SPI 格式
0x2（SPI_QUAD）：四轴 SPI 格式
0x3（SPI_OCTAL）：八进制 SPI 格式

CFS — Control Frame Size

选择微线帧格式的控制字的长度。

0x0（SIZE_01_BIT）：1 位控制 Word
0x1（SIZE_02_BIT）：2 位控制 Word
0x2（SIZE_03_BIT）：3 位控制 Word
0x3（SIZE_04_BIT）：4 位控制 Word
0x4（SIZE_05_BIT）：5 位控制 Word
0x5（SIZE_06_BIT）：6 位控制 Word
0x6（SIZE_07_BIT）：7 位控制 Word

©Levetop Semiconductor Co., Ltd.

0x7（SIZE_08_BIT）：8 位控制 Word
0x8（SIZE_09_BIT）：9 位控制 Word
0x9（SIZE_10_BIT）：10 位控制 Word
0xA（SIZE_11_BIT）：11 位控制 Word
0xB（SIZE_12_BIT）：12 位控制 Word
0xC（SIZE_13_BIT）：13 位控制 Word
0xD（SIZE_14_BIT）：14 位控制 Word
0xE（SIZE_15_BIT）：15 位控制 Word
0xF（SIZE_16_BIT）：16 位控制 Word

SSTE — Slave Select Toggle Enable.

在时钟阶段（SCPH）设置为 0 的 SPI 模式下操作时，此寄存器控制数据帧之间的从属选择线（ss_*_n）的行为。

1 =（TOGGLE_EN）：ss_*_n 行将在连续的数据帧之间切换，串行时钟（sclk）被保持在其默认值，而 ss_*_n 为高

0 =（TOGGLE_DISABLE）：ss_*_n 将保持低，sclk 将在传输期间持续运行

SRL — Shift Register Loop.

仅用于测试目的。当内部活动时，将发射移位寄存器输出连接到接收移位寄存器输入。可在串行从模式和串行主模式中使用。当 SSI 以环回模式配置为从端时，ss_in_n 和 ssi_clk 信号必须由外部源提供。在这种模式下，从服务器不能生成这些信号，因为没有什么可以返回的内容。

1 =（TESTING_MODE）：测试模式操作

0 =（NORMAL_MODE）：正常模式操作

SLV_OE — Slave Output Enable.

仅当 SSI 被配置为串行从属设备时才相关。当配置为串行主节点时，此位字段没有任何功能。

1 = 从输出

0 = 从输出

TMOD — Transfer Mode.

选择串行通信的传输方式。此字段不影响转移的两面性性。仅指示接收数据或传输数据是否有效。

0x0（TX_AND_RX）：传输和接收；不适用于增强型 SPI 工作模式
0x1（TX_ONLY）：仅传输模式；或以增强的 SPI 操作模式写入
0x2（RX_ONLY）：仅接收模式；或在增强的 SPI 操作模式下读取
0x3（EEPROM_READ）：EEPROM 读取模式；不适用于增强型 SPI 操作模式

SCPOL — Serial Clock Polarity.

当帧格式（FRF）设置为摩托罗拉 SPI 时有效。用于选择非活动串行时钟的极性，当 SSI 主服务器不在串行总线上主动传输数据时，该时钟保持为非活动状态。

1 =（INACTIVE_HIGH）：串行时钟的非活动状态较高

0 =（INACTIVE_LOW）：串行时钟的非活动状态较低

©Levetop Semiconductor Co., Ltd.

SCPH — Serial Clock Phase.

当帧格式（FRF）设置为摩托罗拉 SPI 时有效。串行时钟相位选择串行时钟与从机选择信号之间的关系。

当 SCPH = 0 时，数据在串行时钟的第一个边缘捕获数据。当 SCPH = 1 时，在从属选择线被激活后，串行时钟开始切换一个周期，并在串行时钟的第二个边缘捕获数据。

1 =（START_BIT）：在第一位开始时进行串行时钟切换

0 =（MIDDLE_BIT）：串行时钟在第一位的中间进行切换

FRF — Frame Format.

选择由哪个串行协议传输数据。

0x0（SPI）：摩托罗拉 SPI 帧格式

0x1（SSP）：德州仪器公司的 SSP 帧格式

0x2（微线）：国家半导体微线帧格式

0x3（保留）：保留

DFS — Data Frame Size.

选择数据帧长度。当数据帧大小被编程为小于 32 位时，接收数据被接收逻辑自动对正，接收 FIFO 的上限被零填充。

在写入传输 FIFO 之前，您必须正确证明传输数据。传输逻辑在传输数据时忽略上部未使用的位。

0x0（DFS_01_BIT）：保留

0x1（DFS_02_BIT）：保留

0x2（DFS_03_BIT）：保留

0x3（DFS_04_BIT）：4 位串行数据传输

0x4（DFS_05_BIT）：5 位串行数据传输

0x5（DFS_06_BIT）：6 位串行数据传输

0x6（DFS_07_BIT）：7 位串行数据传输

0x7（DFS_08_BIT）：8 位串行数据传输

0x8（DFS_09_BIT）：9 位串行数据传输

0x9（DFS_10_BIT）：10 位串行数据传输

0xA（DFS_11_BIT）：11 位串行数据传输

0xB（DFS_12_BIT）：12 位串行数据传输

0xC（DFS_13_BIT）：13 位串行数据传输

0xD（DFS_14_BIT）：14 位串行数据传输

0xE（DFS_15_BIT）：15 位串行数据传输

0xF（DFS_16_BIT）：16 位串行数据传输

0x10（DFS_17_BIT）：17 位串行数据传输

0x11（DFS_18_BIT）：18 位串行数据传输

0x12（DFS_19_BIT）：19 位串行数据传输

0x13（DFS_20_BIT）：20 位串行数据传输

0x14（DFS_21_BIT）：21 位串行数据传输

0x15（DFS_22_BIT）：22 位串行数据传输

©Levetop Semiconductor Co., Ltd.

0x16（DFS_23_BIT）：23 位串行数据传输
0x17（DFS_24_BIT）：24 位串行数据传输
0x18（DFS_25_BIT）：25 位串行数据传输
0x19（DFS_26_BIT）：26 位串行数据传输
0x1A（DFS_27_BIT）：27 位串行数据传输
0x1B（DFS_28_BIT）：28 位串行数据传输
0x1C（DFS_29_BIT）：29 位串行数据传输
0x1D（DFS_30_BIT）：30 位串行数据传输
0x1E（DFS_31_BIT）：31 位串行数据传输
0x1F（DFS_32_BIT）：32 位串行数据传输

26.6.2.2. 控制寄存器 1（CTRLR1）

只有当 SSI 被配置为主设备时，此寄存器才存在。当 SSI 配置为串行从服务器时，写入此位置无效；从此位置读取返回 0。控制寄存器 1 在 RX-ONLY 模式和 TX-模式 ONLY 模式下控制串行传输的结束。当启用 SSI 时，不可能写入此寄存器。通过写入 SSIENR 寄存器来启用和禁用 SSI。

地址：QSPIn_BASEADDR+0x0000_0004

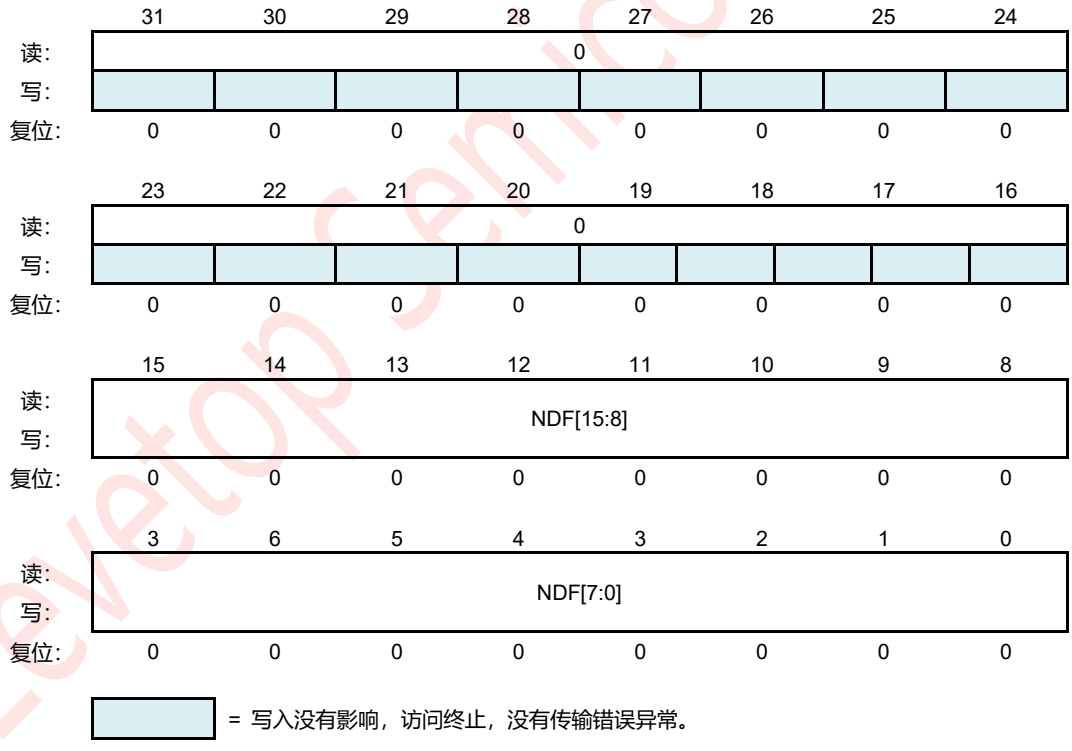


图 26-4：控制寄存器 1（CTRLR1）

NDF[15:0] — Number of Data Frames.

当 TMOD = 01 或 TMOD = 10 或 TMOD = 11 时，该寄存器字段设置了 SSI 要连续接收或传输的数据帧数。SSI 继续接收或传输串行数据，直到数据帧数等于该寄存器值加上 1。当 SSI 被配置为串行从属服务器时，这个寄存器没有任何作用，也不存在。当 TMOD = 01 时，必须设置 SPI_CTRLR0 中的 CLK_STRETCH_EN。

©Levetop Semiconductor Co., Ltd.

26.6.2.3. SSI 启用寄存器（SSIENR）

地址：QSPIn_BASEADDR+0x0000_0008



图 26-5：SSI 启用寄存器（SSIENR）

SSIC_EN — SSI Enable.

启用和禁用所有 SSI 操作。禁用后，所有串行传输都将立即停止。当设备被禁用时，将清除传输和接收 FIFO 缓冲区。当启用时，不可能对某些 SSI 控制寄存器进行编程。当禁用时，设置 SSI 睡眠输出（延迟后），通知系统删除 ssi_clk 是安全的，从而节省系统的功耗。

1 = SSI 已启用。

0 = SSI 禁用。

©Levetop Semiconductor Co., Ltd.

26.6.2.4. 微线控制寄存器（MWCR）

地址：QSPIn_BASEADDR+0x0000_000C

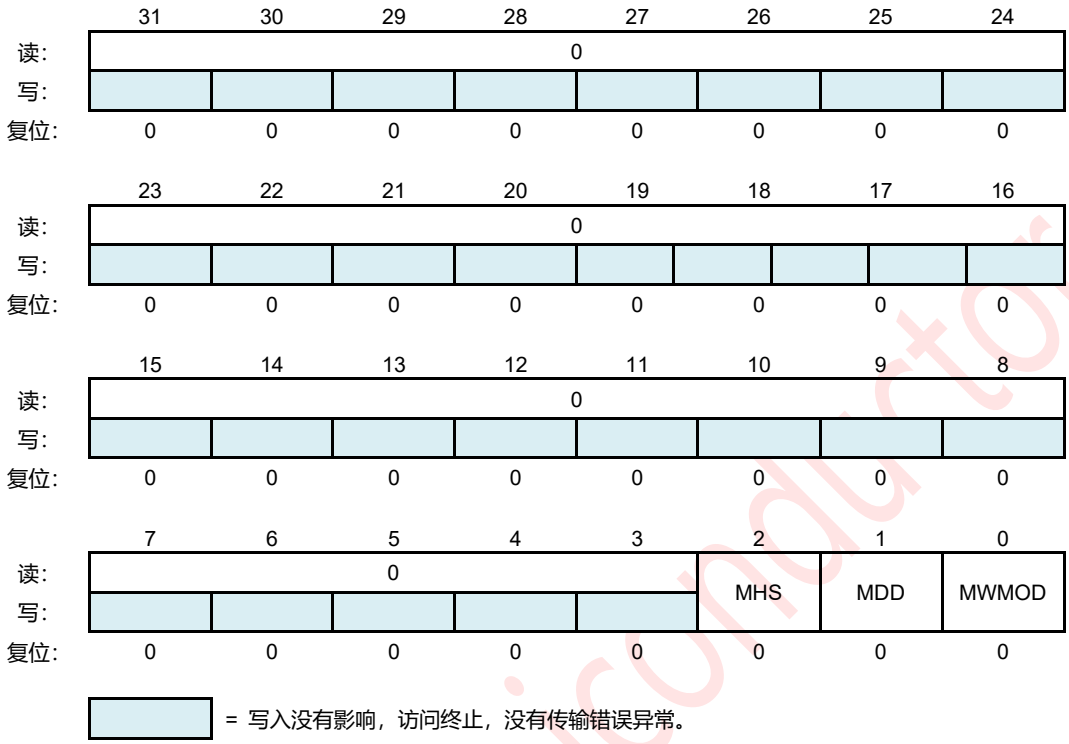


图 26-6：微线控制寄存器（MWCR）

该寄存器控制半双工微线串行协议的数据字的方向。当启用 SSI 时，不可能写入此寄存器。通过写入 SSIENR 寄存器来启用和禁用 SSI。

MHS — Microwire Handshaking.

仅当 SSI 被配置为串行主设备时才相关。当配置为串行从属服务器时，此位字段没有任何功能。用于启用和禁用微线协议的繁忙/准备就绪握手接口。当启用时，SSI 在传输最后一个数据/控制位之后，在清除 SR 寄存器中的 BUSY 状态之前，从目标从服务器检查就绪状态。

1 = 已启用了握手界面

0 = 握手界面禁用

MDD — Microwire Control.

定义使用微线串行协议时数据字的方向。当此位设置为 0 时，SSI MacroCell 从外部串行设备接收数据字。当这个位被设置为 1 时，数据字从 SSI 宏单元传输到外部串行设备。

1 = SSI 传输数据

0 = SSI 接收数据

©Levetop Semiconductor Co., Ltd.

MWMOD — Microwire Transfer Mode.

定义微线传输是顺序的还是非顺序的。当使用顺序模式时，只需要一个控制字来发送或接收一个数据字块。当使用非顺序模式时，对于被传输或接收的每个数据字，必须有一个控制字。

1 = 顺序转移

0 = 非顺序传输

26.6.2.5. 从机启用寄存器（SER）

地址：QSPIn_BASEADDR+0x0000_0010

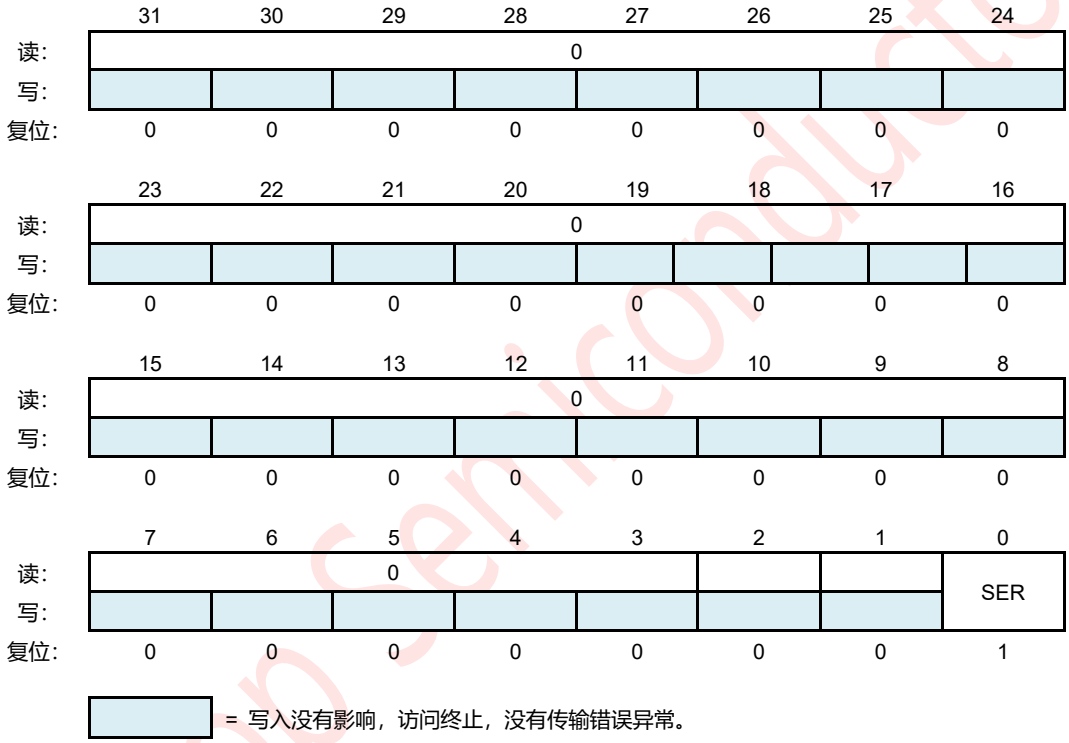


图 26-7：从属服务器启用寄存器（SER）

此寄存器仅在将 SSI 配置为主设备时有效。当 SSI 配置为串行从服务器时，写入此位置无效；从此位置读取返回 0。该寄存器允许单个从服务器从 SSI 主服务器中选择输出行。在 SSI 主服务器上最多有 16 个从属选择输出引脚。当 SSI 忙于和 SSIC_EN = 1 时，您无法写入此寄存器。

SER — Slave Select Enable Flag.

1 = 选择

0 = 未选择

©Levetop Semiconductor Co., Ltd.

26.6.2.6. 波特率选择（BAUDR）

地址：QSPIn_BASEADDR+0x0000_0014

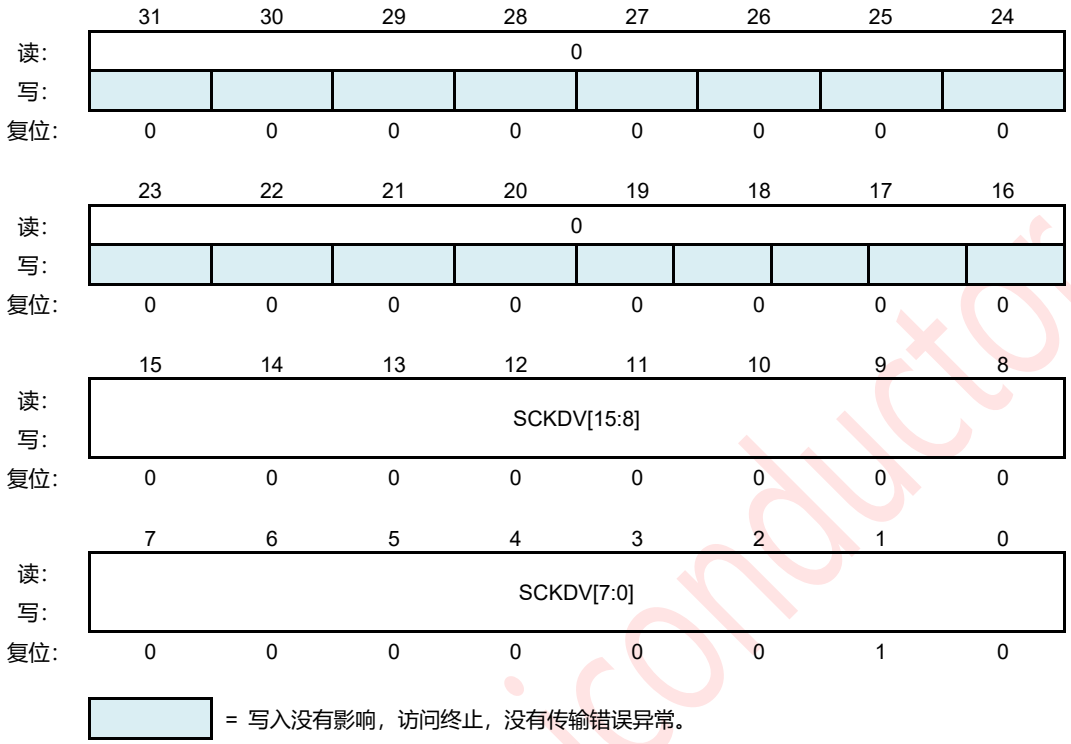


图 26-8：波特率选择（BAUDR）

SCKDV[15:0] — SSI Clock Divider.

此字段的 LSB 总是设置为 0，并且不受写操作的影响，该写操作确保在此寄存器中保持偶数值。如果该值为 0，则会禁用串行输出时钟（sclk_out）。sclk_out 的频率由以下公式得到：

Fsclk_out = Fssi_clk / SCKDV

其中，SCKDV 是介于 2 到 65534 之间的任何偶数值。例如：适用于 Fssi_clk = 3.6864MHz 和 SCKDV = 2 Fsclk_out = 3.6864/2 = 1.8432MHz。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.7. 传输 FIFO 阈值等级（TXFTLR）

地址：QSPIn_BASEADDR+0x0000_0018

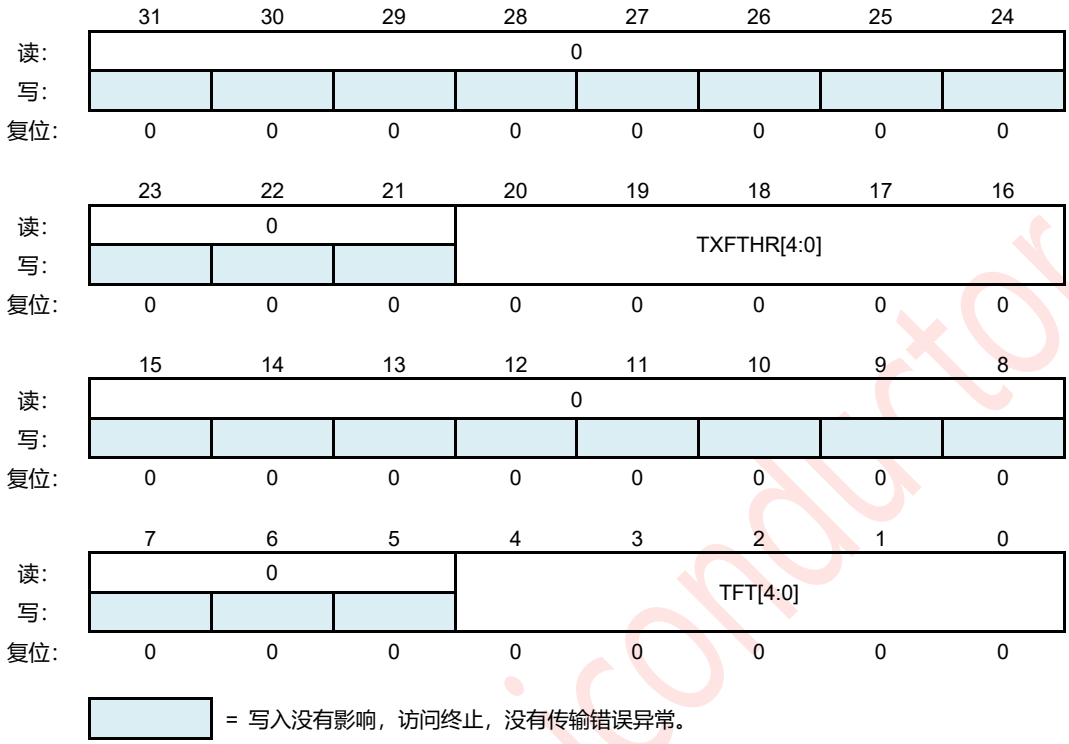


图 26-9：传输 FIFO 阈值级别（TXFTLR）

该寄存器控制传输 FIFO 存储器的阈值。通过写入 SSIENR 寄存器来启用和禁用 SSI。

TXFTHR[4:0] — Transfer start FIFO level.

用于控制传输 FIFO 中的入口级别，超过该级别的传输将从串行线开始。此寄存器可用于确保在开始在串行线上的写入操作之前，在传输 FIFO 中存在足够的数据。这些字段仅对主操作模式有效。

TFT[4:0] — Transmit FIFO Threshold.

控制传输 FIFO 控制器触发中断的条目的级别（或以下）。FIFO 深度可在 8-256 范围内配置；该寄存器的大小符合访问 FIFO 所需的地址位数。如果您试图将此值设置为大于或等于 FIFO 的深度，则不会写入此字段，并保留其当前值。当发送 FIFO 条目的数量小于或等于该值时，将触发发送 FIFO 空中断。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.8. 接收 FIFO 阈值级别（RXFTLR）

地址：QSPIn_BASEADDR+0x0000_001C

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0			RFT[4:0]				
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 26-10：接收 FIFO 阈值级别（RXFTLR）

该寄存器控制接收 FIFO 存储器的阈值。通过写入 SSIENR 寄存器来启用和禁用 SSI。

RFT[4:0] — Receive FIFO Threshold.

控制接收 FIFO 控制器触发中断的条目的级别（或以上）。这个寄存器的大小被调整为访问 FIFO 所需的地址位数。如果您试图将此值设置为大于 FIFO 的深度，则不会写入此字段，并保留其当前值。当接收的 FIFO 条目的数量大于或等于该值+1 时，将触发接收的 FIFO 完全中断。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.9. 发射 FIFO 液位寄存器（TXFLR）

地址：QSPIn_BASEADDR+0x0000_0020

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0		TXTFL[5:0]					
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 26-11：传输 FIFO 液位寄存器（TXFLR）

TXTFL[5:0] — Transmit FIFO Level.

包含传输 FIFO 中的有效数据条目的数量。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.10. 接收 FIFO 级别寄存器（RXFLR）

地址：QSPIn_BASEADDR+0x0000_0024

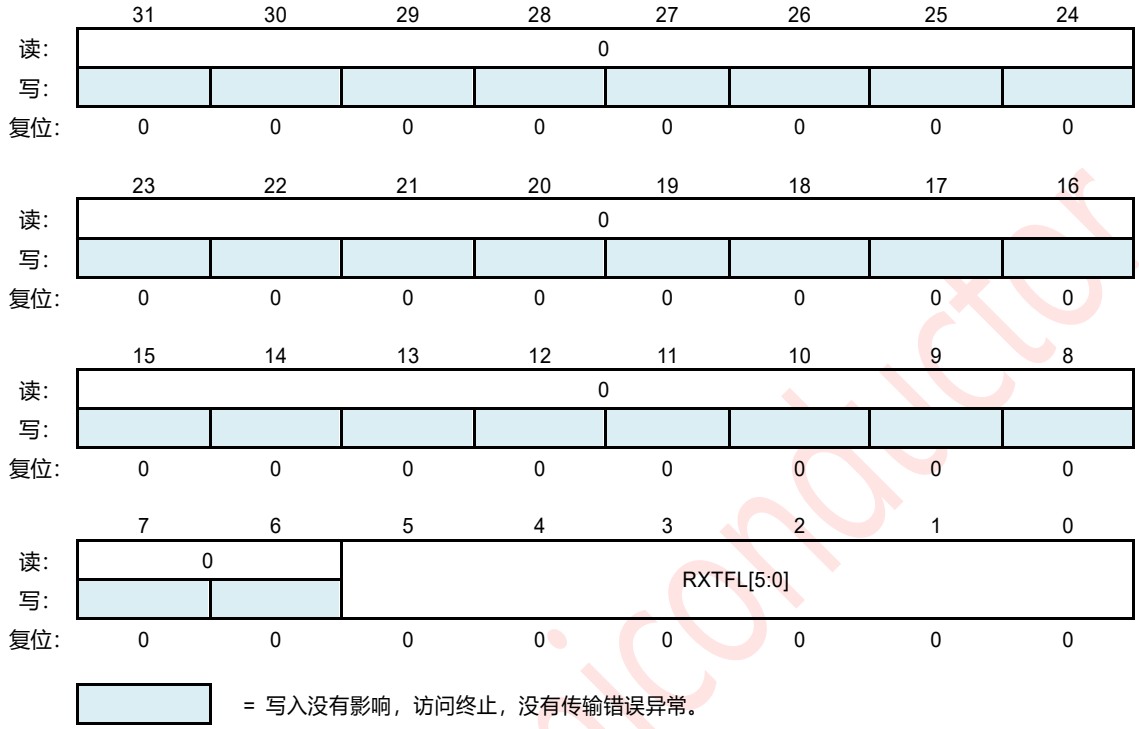


图 26-12：接收 FIFO 级别寄存器（RXFLR）

RXTFL[5:0] — Receive FIFO Level.

包含接收 FIFO 中的有效数据条目的数量。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.11. 状态寄存器（SR）

地址：QSPIn_BASEADDR+0x0000_0028

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0	保留	TXE	RFF	RFNE	TFE	TFNF	BUSY
写:								
复位:	0	0	0	0	0	1	1	0

= 写入没有影响，访问终止，没有传输错误异常。

图 26-13：状态寄存器（SR）

TXE — Transmission Error.

设置在启动传输时，传输 FIFO 是否为空。只有当 SSI 被配置为从设备时，才能设置此位。来自以前传输的数据将在 TXD 线上重新发送。读取时，将清除此位。

1 =（TX_ERROR）：传输错误

0 =（NO_ERROR）：无错误

RFF — Receive FIFO Full.

当接收 FIFO 完全满时，将设置此位。当接收 FIFO 包含一个或多个空位置时，该位将被清除。

1 =（满）：接收 FIFO 已满

0 =（NOT_FULL）：接收 FIFO 未满

RFNE — Receive FIFO Not Empty.

当接收 FIFO 包含一个或多个条目时设置，当接收 FIFO 为空时清除。这个位可以通过软件轮询以完全清空接收 FIFO。

1 =（NOT_EMPTY）：接收 FIFO 不是空的

0 =（空）：接收 FIFO 为空

©Levetop Semiconductor Co., Ltd.

TFE — Transmit FIFO Empty.

当传输 FIFO 完全为空时，将设置此位。当传输 FIFO 包含一个或多个有效条目时，该位将被清除。此位字段不请求中断。

1 = （空）：传输 FIFO 为空

0 =（NOT_EMPTY）：传输 FIFO 不是空的

TFNF — Transmit FIFO Not Full.

当传输 FIFO 包含一个或多个空位置时设置，并在 FIFO 满时清除。

1 =（NOT_FULL）：Tx FIFO 未满

0 = Tx FIFO 已满

BUSY — SSI Busy Flag.

设置时，表示正在进行串行传输；清除时表示 SSI 空闲或禁用。

1 =（活动）：SSI 正在主动传输数据

0 =（非活动）：SSI 已空闲或已禁用

26.6.2.12. 中断掩码寄存器（IMR）

地址：QSPIn_BASEADDR+0x0000_002C

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0	XRXOIM	Reserved	RXFIM	RXOIM	RXUIM	TXOIM	TXEIM
写:								
复位:	0	1	1	1	1	1	1	0

= 写入没有影响，访问终止，没有传输错误异常。

图 26-14：中断掩码寄存器（IMR）

©Levetop Semiconductor Co., Ltd.

XRXOIM — XIP Receive FIFO Overflow Interrupt Mask

1 = （未被屏蔽）：ssi_xrxo_intr 中断未被屏蔽

0 = （被屏蔽）：ssi_xrxo_intr 中断被屏蔽

RXFIM — Receive FIFO Full Interrupt Mask

1 = （未被屏蔽）：ssi_rxf_intr 中断未被屏蔽

0 = （被屏蔽）：ssi_rxf_intr 中断被屏蔽

RXOIM — Receive FIFO Overflow Interrupt Mask

1 = （未被屏蔽）：ssi_rxo_intr 中断未被屏蔽

0 = （被屏蔽）：ssi_rxo_intr 中断被屏蔽

RXUIM — Receive FIFO Underflow Interrupt Mask

1 = （未被屏蔽）：ssi_rxu_intr 中断未被屏蔽

0 = （被屏蔽）：ssi_rxu_intr 中断被屏蔽

TXOIM — Transmit FIFO Overflow Interrupt Mask

1 = （未被屏蔽）：ssi_txo_intr 中断未被屏蔽

0 = （被屏蔽）：ssi_txo_intr 中断被屏蔽

TXEIM — Transmit FIFO Empty Interrupt Mask

1 = （未被屏蔽）：ssi_txe_intr 中断未被屏蔽

0 = （被屏蔽）：ssi_txe_intr 中断被屏蔽

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.13. 中断状态寄存器（ISR）

地址：QSPIn_BASEADDR+0x0000_0030

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0	XRXOIS	保留	RXFIS	RXOIS	RXUIS	TXOIS	TXEIS
写:								
复位:	0	0	0	0	0	0	0	1

= 写入没有影响，访问终止，没有传输错误异常。

图 26-15：中断状态寄存器（ISR）

此寄存器会在屏蔽 SSI 中断后报告它们的状态。

XRXOIS — XIP Receive FIFO Overflow Interrupt Status

1 = （活动）：掩蔽后的 ssi_xrxo_intr 中断被激活

0 = （非活动）：屏蔽后 ssi_xrxo_intr 中断未激活

RXFIS — Receive FIFO Full Interrupt Status

1 = （活动）：掩蔽后的 ssi_rxf_intr 中断被激活

0 = （非活动）：屏蔽后 ssi_rxf_intr 中断未激活

RXOIS — Receive FIFO Overflow Interrupt Status

1 = （活动）：掩蔽后的 ssi_rxo_intr 中断被激活

0 = （非活动）：屏蔽后 ssi_rxo_intr 中断未激活

RXUIS — Receive FIFO Underflow Interrupt Status

1 = （活动）：掩蔽后的 ssi_rxu_intr 中断被激活

0 = （非活动）：屏蔽后 ssi_rxu_intr 中断未激活

©Levetop Semiconductor Co., Ltd.

TXOIS — Transmit FIFO Overflow Interrupt Status

1 = （活动）：掩蔽后的 ssi_txo_intr 中断被激活

0 = （非活动）：屏蔽后 ssi_txo_intr 中断未激活

TXEIS — Transmit FIFO Empty Interrupt Status

1 = （活动）：掩蔽后的 ssi_txe_intr 中断被激活

0 = （非活动）：屏蔽后 ssi_txe_intr 中断未激活

26.6.2.14. 原始中断状态寄存器（RISR）

地址：QSPIn_BASEADDR+0x0000_0034

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0	XRXOIS	保留	RXFIS	RXOIS	RXUIS	TXOIS	TXEIS
写:								
复位:	0	0	0	0	0	0	0	1

= 写入没有影响，访问终止，没有传输错误异常。

图 26-16：原始中断状态寄存器（RISR）

XRXOIS — XIP Receive FIFO Overflow Raw Interrupt Status

1 = （激活）：ssi_xrxo_intr 中断在屏蔽之前是激活的

0 = （非活动）：ssi_xrxo_intr 中断不是活动的之前的屏蔽

RXFIS — Receive FIFO Full Raw Interrupt Status

1 = （激活）：ssi_rxf_intr 中断在屏蔽之前是激活的

0 = （非活动）：ssi_rxf_intr 中断不是活动的之前的屏蔽

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

RXOIS — Receive FIFO Overflow Raw Interrupt Status

1 = （激活）：ssi_rxo_intr 中断在屏蔽之前是激活的

0 = （非活动）：ssi_rxo_intr 中断不是活动的之前的屏蔽

RXUIS — Receive FIFO Underflow Raw Interrupt Status

1 = （激活）：ssi_rxu_intr 中断在屏蔽之前是激活的

0 = （非活动）：ssi_rxu_intr 中断不是活动的之前的屏蔽

TXOIS — Transmit FIFO Overflow Raw Interrupt Status

1 = （激活）：ssi_txo_intr 中断在屏蔽之前是激活的

0 = （非活动）：ssi_txo_intr 中断不是活动的之前的屏蔽

TXEIS — Transmit FIFO Empty Raw Interrupt Status

1 = （激活）：ssi_txe_intr 中断在屏蔽之前是激活的

0 = （非活动）：ssi_txe_intr 中断不是活动的之前的屏蔽

26.6.2.15. 传输 FIFO 溢出中断清除寄存器（TXOICR）

地址：QSPIn_BASEADDR+0x0000_0038

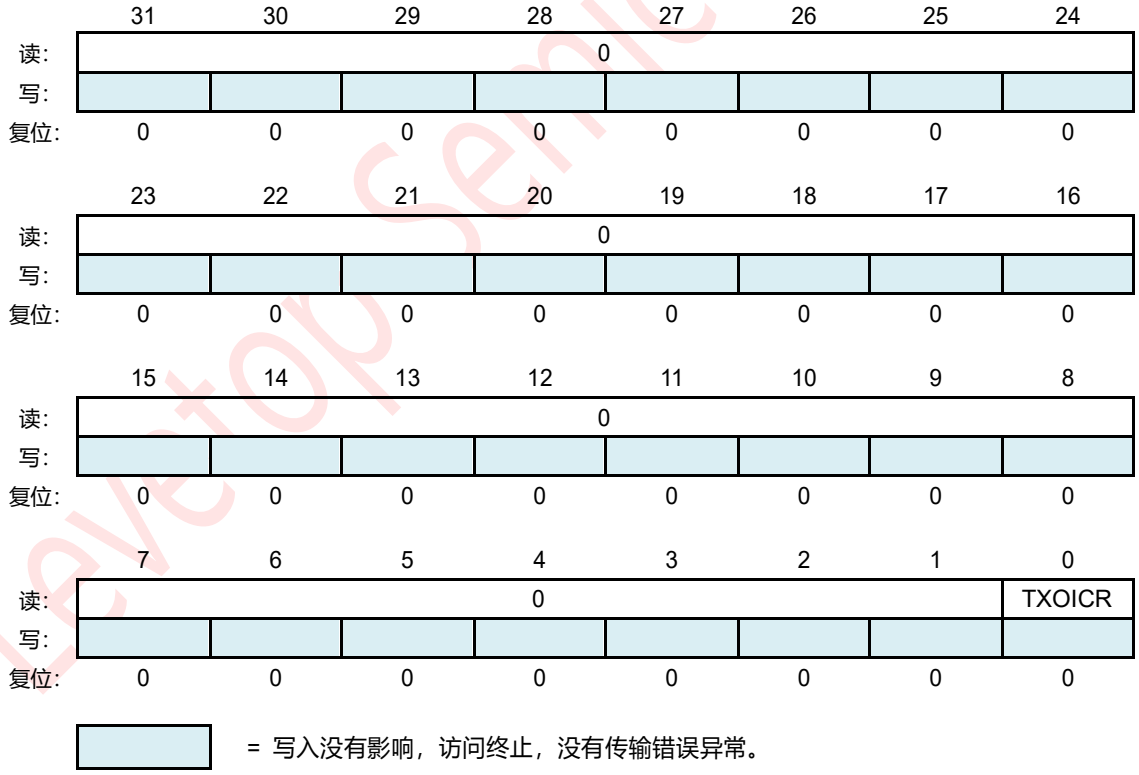


图 26-17：传输 FIFO 溢出中断清除寄存器（TXOICR）

TXOICR — Clear Transmit FIFO Overflow Interrupt.

这个寄存器反映了中断的状态。从这个寄存器读取会清除 ssi_txo_intr 中断；写入没有效果。

©Levetop Semiconductor Co., Ltd.

26.6.2.16. 接收 FIFO 溢出中断清除寄存器（RXOICR）

地址：QSPIn_BASEADDR+0x0000_003C

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0							RXOICR
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 26-18：接收 FIFO 溢出中断清除寄存器（RXOICR）

RXOICR — Clear Receive FIFO Overflow Interrupt.

这个寄存器反映了中断的状态。从这个寄存器读取会清除 ssi_rxo_intr 中断；写入没有效果。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.17. 接收 FIFO 下流中断清除寄存器（RXUICR）

地址：QSPIn_BASEADDR+0x0000_0040

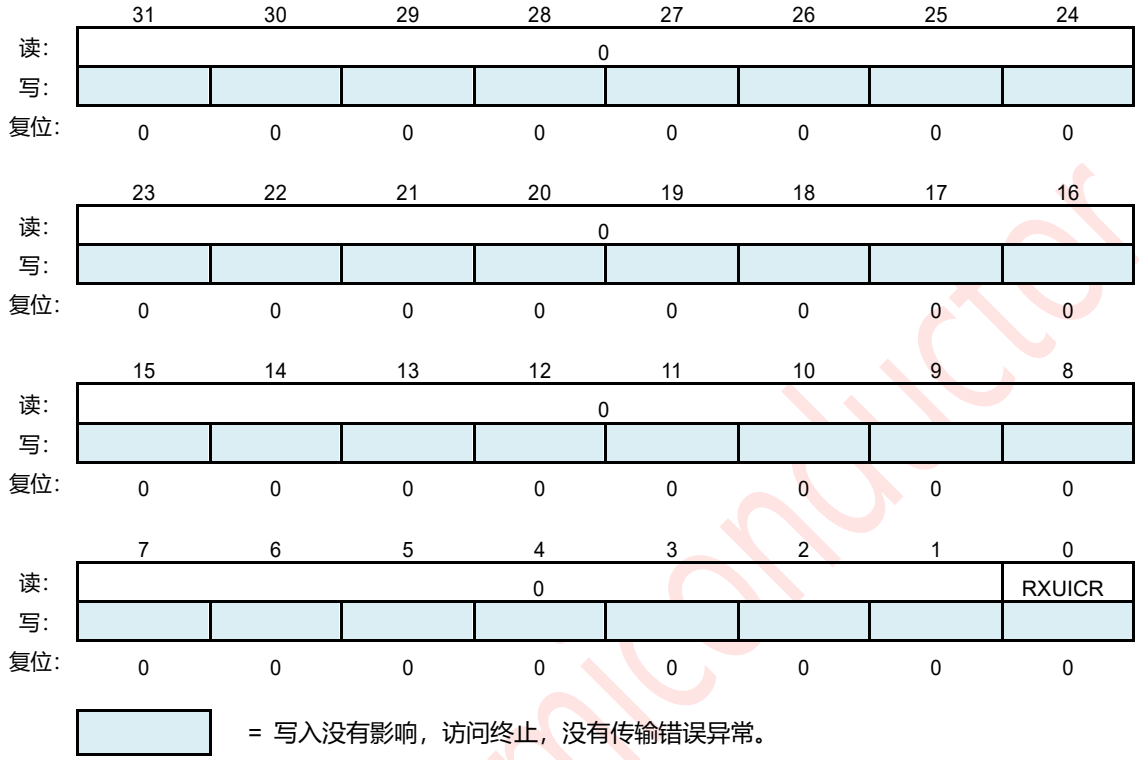


图 26-19：接收 FIFO 下流中断清除寄存器（RXUICR）

RXUICR — Clear Receive FIFO Underflow Interrupt.

这个寄存器反映了中断的状态。从这个寄存器读取会清除 ssi_rxu_intr 中断；写入没有效果。

©Levetop Semiconductor Co., Ltd.

26.6.2.18. 中断清除寄存器（ICR）

地址：QSPIn_BASEADDR+0x0000_0048

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0							ICR
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 26-20：中断清除寄存器（ICR）

ICR — Clear Interrupts.

如果下面的任何一个中断都是活动的，则会设置此寄存器。读取将清除 ssi_txo_intr、ssi_rxu_intr、ssi_rxo_intr 和 ssi_mst_intr 中断。写入此寄存器没有效果。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.19. DMA 控制寄存器（DMACR）

地址：QSPIn_BASEADDR+0x0000_004C

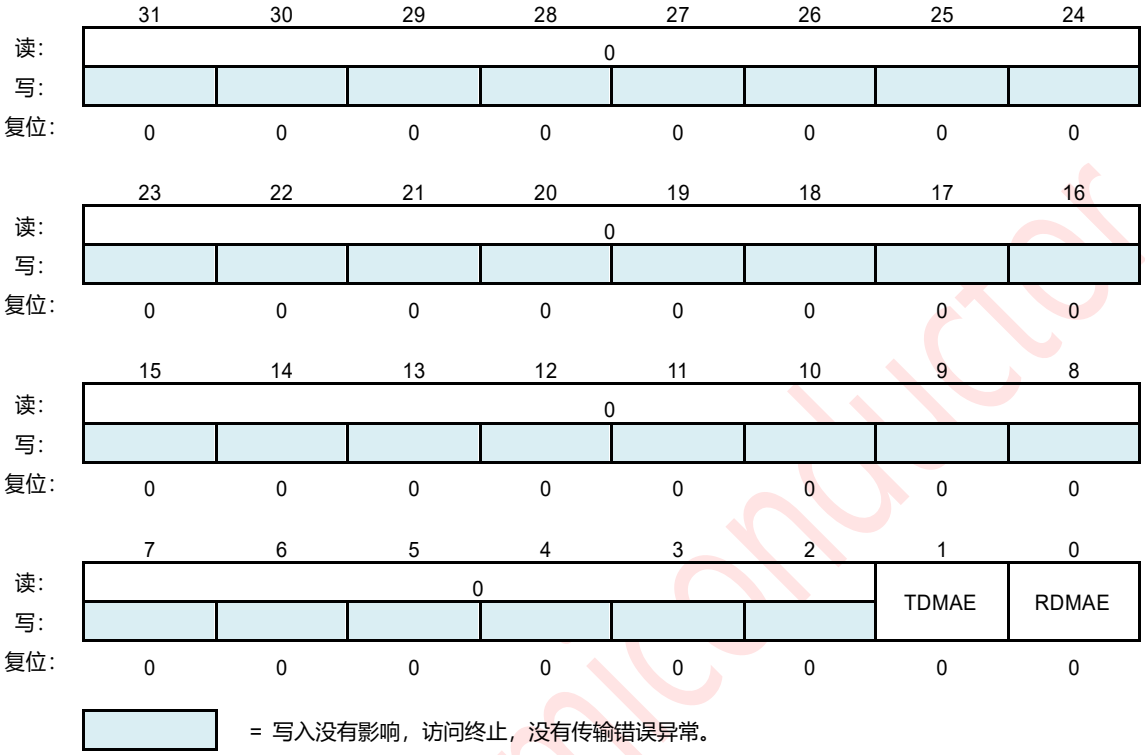


图 26-21：DMA 控制寄存器（DMACR）

此寄存器仅在 SSI 配置了一组 DMA 控制器接口信号（SSIC_HAS_DMA = 1）时有效。当 SSI 未配置为 DMA 操作时，此寄存器将不存在，并且写入该寄存器的地址将无效；从此寄存器地址读取将返回零。该寄存器用于启用 DMA 控制器接口操作。

TDMAE — Transmit DMA Enable

此位可启用/禁用传输 FIFO DMA 信道。

1 = 传输 DMA

0 = 传输 DMA 已被禁用

RDMAE — Receive DMA Enable

此位可启用/禁用接收 FIFO DMA 通道。

1 = 接收 DMA

0 = 接收 DMA 已被禁用

©Levetop Semiconductor Co., Ltd.

26.6.2.20. DMA 传输数据级别（DMATDLR）

地址：QSPIn_BASEADDR+0x0000_0050

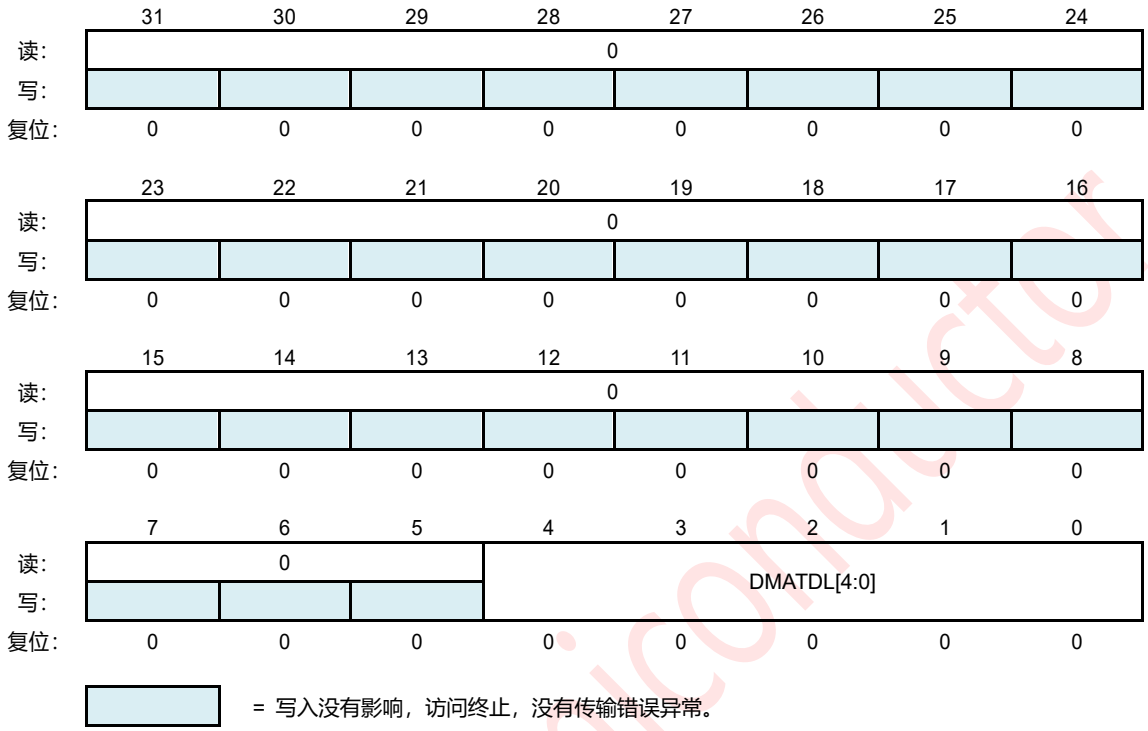


图 26-22：DMA 传输数据级别（DMATDLR）

此寄存器仅在 SSI 配置了一组 DMA 接口信号（SSIC_HAS_DMA = 1）时有效。当未为 DMA 操作配置 SSI 时，此寄存器将不存在，并且写入其地址将无效；从其地址读取将返回零。

DMATDL[4:0] — Transmit Data Level.

该位字段控制由发送逻辑发出 DMA 请求的级别。等于水印电平，即当发送 FIFO 中的有效数据项数等于或低于该字段值时，以及 TDMAE = 1，即产生 dma_tx_req 信号。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.21. DMA 接收数据级别（DMARDLR）

地址：QSPIn_BASEADDR+0x0000_0054

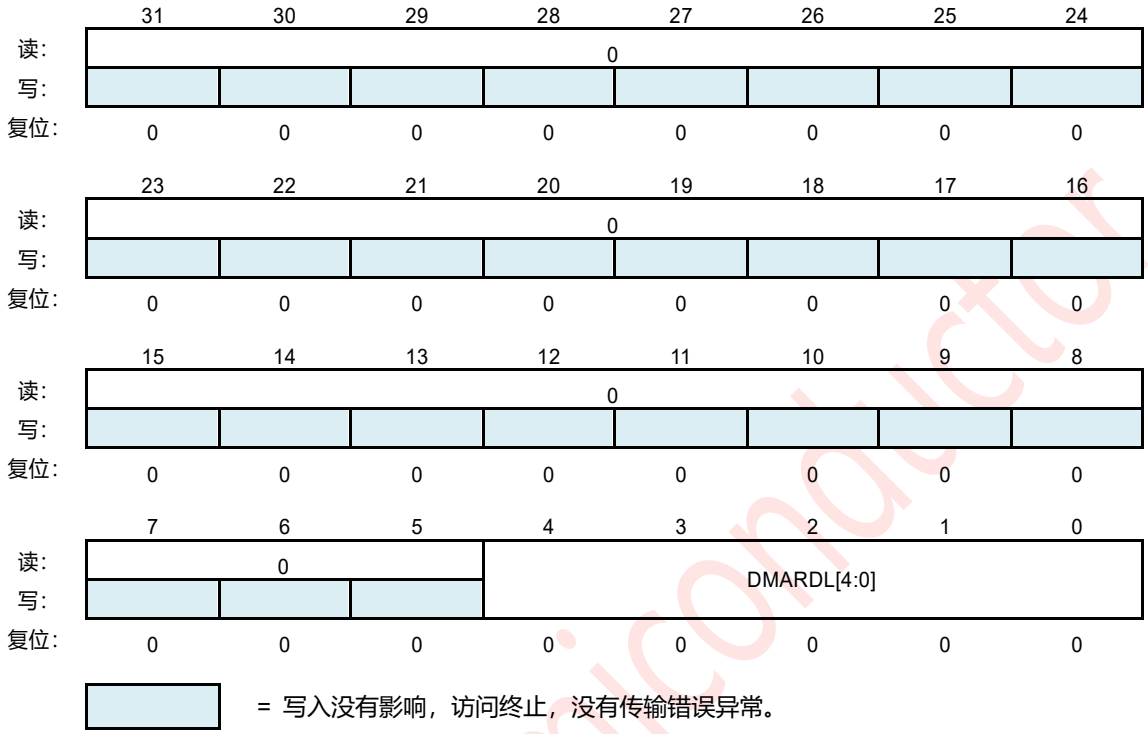


图 26-23：DMA 接收数据级别（DMARDLR）

此寄存器仅在 SSI 配置了一组 DMA 接口信号（SSIC_HAS_DMA = 1）时有效。当未为 DMA 操作配置 SSI 时，此寄存器将不存在，并且写入其地址将无效；从其地址读取将返回零。

DMARDL[4:0] — Receive Data Level.

该位字段控制由接收逻辑发出 DMA 请求的级别。水印级别 = DMARDL+1，即，当接收 FIFO 中的有效数据条目数等于或高于该字段值+1 和 RDMAE = 1 时，生成 dma_rx_req。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.22. 识别寄存器（IDR）

地址：QSPIn_BASEADDR+0x0000_0058

	31	30	29	28	27	26	25	24
读:	IDCODE[31:24]							
写:								
复位:	1	1	1	1	1	1	1	1

	23	22	21	20	19	18	17	16
读:	IDCODE[23:16]							
写:								
复位:	1	1	1	1	1	1	1	1

	15	14	13	12	11	10	9	8
读:	IDCODE[15:8]							
写:								
复位:	1	1	1	1	1	1	1	1

	7	6	5	4	3	2	1	0
读:	IDCODE[7:0]							
写:								
复位:	1	1	1	1	1	1	1	1

= 写入没有影响，访问终止，没有传输错误异常。

图 26-24：识别寄存器（IDR）

IDCODE[31:0] — Identification code.

寄存器包含外围设备的标识代码，在配置时写入寄存器。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.23. 版本 ID 寄存器（VIDR）

地址：QSPIn_BASEADDR+0x0000_005C

	31	30	29	28	27	26	25	24
读:	SSIC_COMP_VERSION[31:24]							
写:								
复位:	0	0	1	1	0	0	0	1

	19	22	21	20	19	18	17	16
读:	SSIC_COMP_VERSION[23:16]							
写:								
复位:	0	0	1	1	0	0	0	0

	11	14	13	12	11	10	9	8
读:	SSIC_COMP_VERSION[15:8]							
写:								
复位:	0	0	1	1	0	0	0	1

	3	6	5	4	3	2	1	0
读:	SSIC_COMP_VERSION[7:0]							
写:								
复位:	0	0	1	0	1	0	1	0

= 写入没有影响，访问终止，没有传输错误异常。

图 26-25：版本 ID 寄存器（VIDR）

SSIC_COMP_VERSION[31:0]

包含同步系统组件版本的十六进制表示形式。由版本中每个数字的 ASCII 值组成，后面跟着*。例如，32_30_31_2A 表示版本 2.01*。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.24. SSI 数据寄存器（DRx）

地址：QSPIn_BASEADDR+0x0000_0060

	31	30	29	28	27	26	25	24
读: 写:	DR[31:24]							
复位:	0	0	0	0	0	0	0	0

	19	22	21	20	19	18	17	16
读: 写:	DR[23:16]							
复位:	0	0	0	0	0	0	0	0

	11	14	13	12	11	10	9	8
读: 写:	DR[15:8]							
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读: 写:	DR[7:0]							
复位:	0	0	0	0	0	0	0	0

图 26-26：SSI 数据寄存器（DRx）

SSI 数据寄存器是发送/接收 FIFOs 的 32 位读/写缓冲区。当读取寄存器时，将访问接收 FIFO 缓冲区中的数据。当它被写入时，数据被移动到传输 FIFO 缓冲区；只有当 SSIC_EN = 1 时才能发生写入。当 SSIC_EN = 为 0 时，fifo 会被复位。

DR[31:0] — Data Register.

在写入此寄存器时，必须正确数据。读取数据将自动被正确证明。

读取 = 接收 FIFO 缓冲区

写入 = 传输 FIFO 缓冲区。

©Levetop Semiconductor Co., Ltd.

26.6.2.25. RX 取样延迟寄存器（RXSDR）

地址：QSPIn_BASEADDR+0x0000_00F0

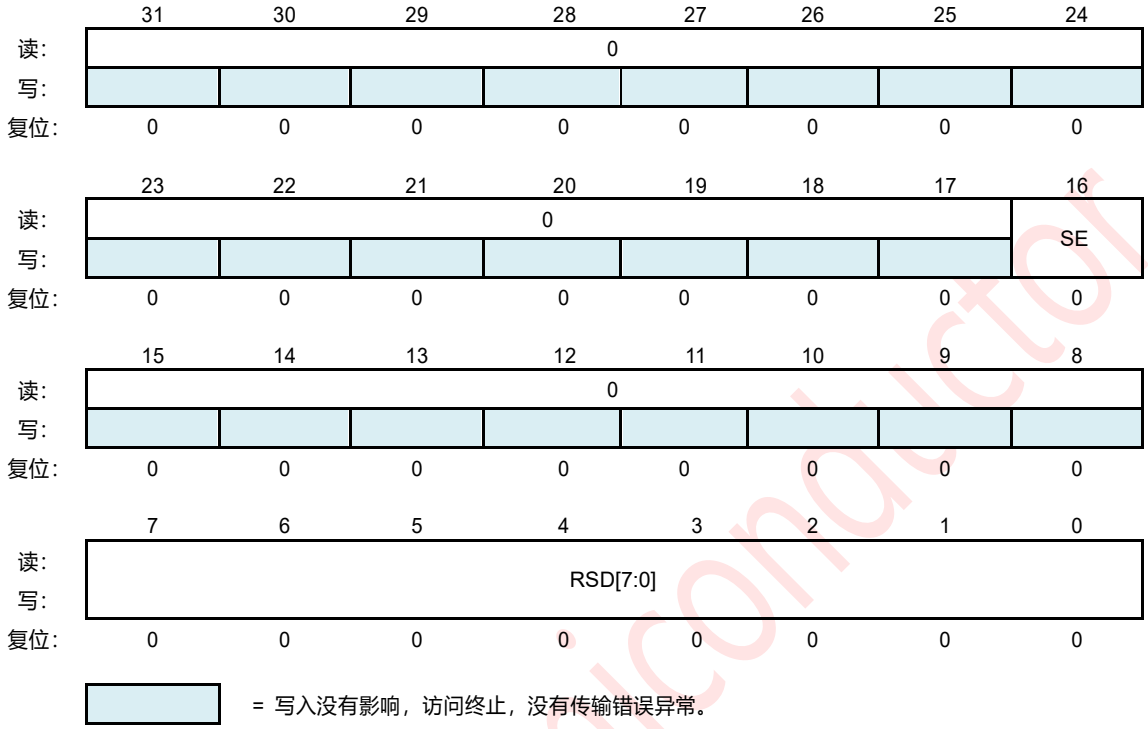


图 26-27：RX 取样延迟寄存器（RXSDR）

SE — Receive Data (RXD) Sampling Edge.

1 = 使用 ssi_clk 的负缘对输入数据进行采样

0 = 使用 ssi_clk 的正缘对输入的数据进行采样

RSD[7:0] — Receive Data (RXD) Sample Delay.

此寄存器用于延迟 RXD 输入端口的样本。每个值代表 RXD 样本上的一个 ssi_clk 延迟。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.26. SPI 控制寄存器 0（SPICTRLR0）

地址：QSPIn_BASEADDR+0x0000_00F4

	31	30	29	28	27	26	25	24
读:	0	CLK_STRETCH_EN	保留	0	保留			
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0	0	保留					
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	WAIT_CYCLES[4:0]					0	INST_L[1:0]	
写:								
复位:	0	0	0	0	0	0	1	0

	7	6	5	4	3	2	1	0
读:	保留	0	ADDR_L[3:0]				TRANS_TYPE[1:0]	
写:								
复位:	0	0	0	1	1	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 26-28：SPI 控制寄存器 0（SPICTRLR0）

CLK_STRETCH_EN

在 SPI 传输中启用时钟拉伸能力。

在写的情况下，如果 FIFO 变成空的，SSI 将拉伸时钟，直到 FIFO 有足够的数据来继续传输。在读取的情况下，如果接收的 FIFO 变成满的，SSI 将停止时钟，直到从 FIFO 读取数据。

提示：建议始终设置此位

1 = clk_stretch_enable

0 = clk_stretch_disable

WAIT_CYCLES[4:0] — Wait cycles in Dual/Quad/Octal mode between control frames transmit and data reception. Specified as number of SPI clock cycles.

INST_L[1:0] — Dual/Quad/Octal mode instruction length in bits.

0x0（INST_L0）：无指令

0x1（INST_L4）：4 位指令长度

0x2（INST_L8）：8 位指令长度

0x3（INST_L16）：16 位指令长度

©Levetop Semiconductor Co., Ltd.

ADDR_L[3:0] — Length of Address to be transmitted

0x0（ADDR_L0）：无地址
0x1（ADDR_L4）：4 位地址长度
0x2（ADDR_L8）：8 位地址长度
0x3（ADDR_L12）：12 位地址长度
0x4（ADDR_L16）：16 位地址长度
0x5（ADDR_L20）：20 位地址长度
0x6（ADDR_L24）：24 位地址长度
0x7（ADDR_L28）：28 位地址长度
0x8（ADDR_L32）：32 位地址长度
0x9（ADDR_L36）：36 位地址长度
0xA（ADDR_L40）：40 位地址长度
0xB（ADDR_L44）：44 位地址长度
0xC（ADDR_L48）：48 位地址长度
0xD（ADDR_L52）：52 位地址长度
0xE（ADDR_L56）：56 位地址长度
0xF（ADDR_L60）：60 位地址长度

TRANS_TYPE[1:0] — Address and instruction transfer format.

选择 SSI 是否将在标准 SPI 模式或在 CTRLR0 中选择的 SPI 模式下传输指令/地址。SPI_FRF 字段。

0x0（TT0）：指令和地址将以标准 SPI 模式发送。
0x1（TT1）：指令将以标准 SPI 模式发送，地址将以 CTRLR0 指定的模式发送。SPI_FRF.
0x2（TT2）：指令和地址都将以 SPI_FRF 指定的方式发送。
0x3（TT3）：保留。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.27. XIP 模式位（XIPMBR）

地址：QSPIn_BASEADDR+0x0000_00FC

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	XIP_MD_BITS[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	XIP_MD_BITS[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 26-29：XIP 模式位（XIPMBR）

该寄存器携带地址阶段后 XIP 模式发送的模式位。这是一个 8 位寄存器，只有在 SSIENR 寄存器设置为 0 时才能写入。

XIP_MD_BITS[15:0]

XIP 传输地址阶段后发送 XIP 模式位。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

26.6.2.28. XIP 中初始寄存器（XIPIIR）

地址：QSPIn_BASEADDR+0x0000_0100

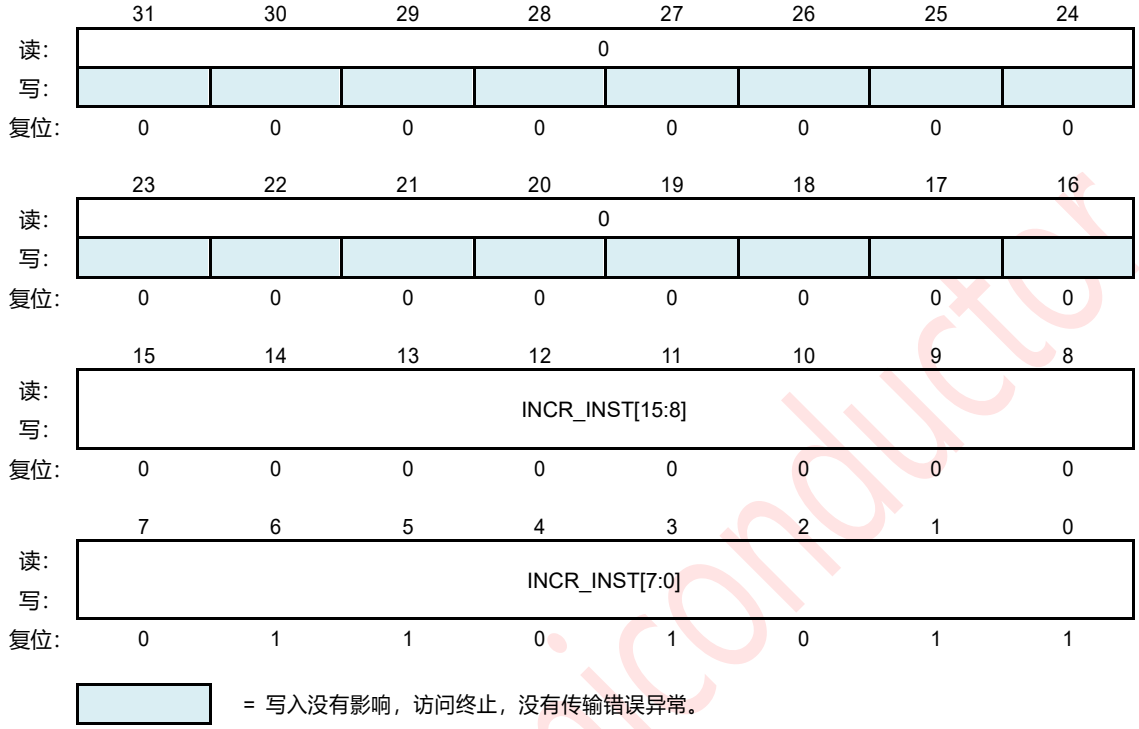


图 26-30：XIP 国际寄存器（XIPIIR）

此寄存器仅在 SSIC_XIP_INST_EN 等于 1 时有效。此寄存器用于存储在 AHB 接口上请求 INCR 事务中使用的指令操作代码。当启用了 SSI（SSIC_EN = 1）时，不可能写入此寄存器。通过写入 SSIENR 寄存器来启用和禁用 SSI。

INCR_INST[15:0] — XIP INCR transfer opcode.

当 SPI_CTRLR0。XIP_INST_EN 位设置为 1，SSI 为所有 XIP 传输发送指令，该寄存器字段存储在 AHB 总线上请求 INCR 类型传输时要发送的指令操作码。在指令阶段要发送的位数由 SPI_CTRL0 决定。INST_L 字段。

©Levetop Semiconductor Co., Ltd.

26.6.2.29. XIP 封装初始寄存器（XIPWIR）

地址：QSPIn_BASEADDR+0x0000_0104

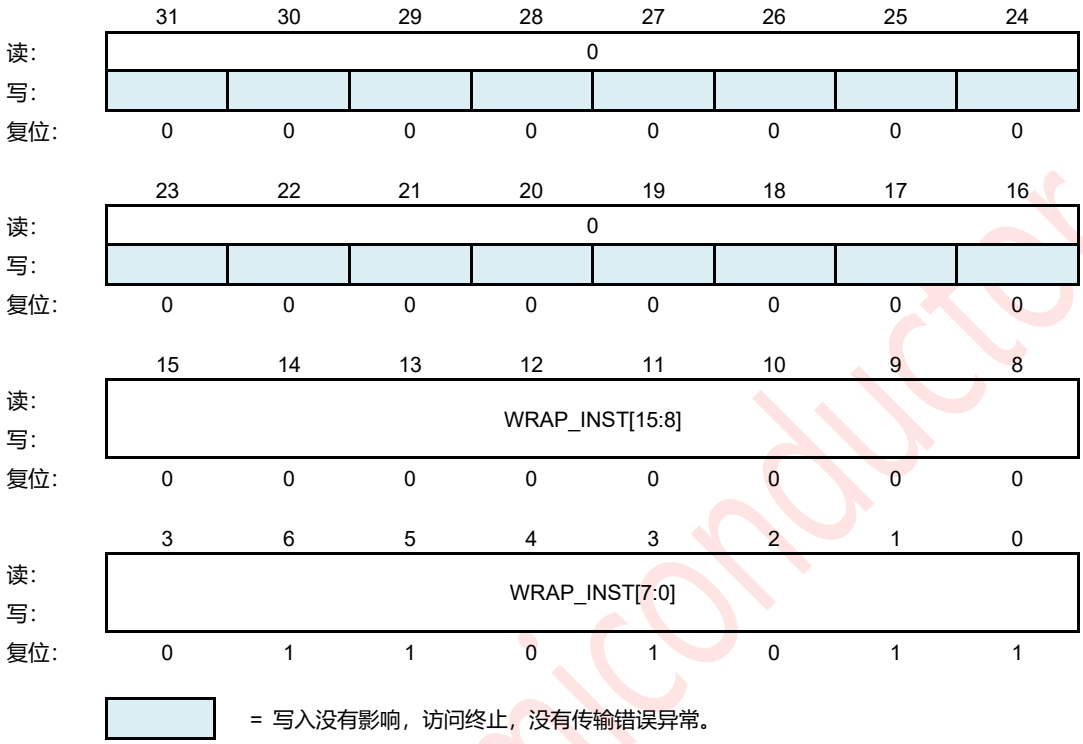


图 26-31：XIP 封装初始寄存器（XIPWIR）

此寄存器仅在 SSIC_XIP_INST_EN 等于 1 时有效。此寄存器用于存储在 AHB 接口上请求时在 WRAP 事务中使用的指令操作代码。当启用了 SSI（SSIC_EN = 1）时，不可能写入此寄存器。通过写入 SSIENR 寄存器来启用和禁用 SSI。

WRAP_INST[15:0] — XIP WRAP transfer opcode.

当 SPI_CTRLR0。XIP_INST_EN 位被设置为 1，SSI 为所有的 XIP 传输发送指令，这个寄存器字段存储在 AHB 总线上请求 WRAP 类型传输时要发送的指令操作码。在指令阶段要发送的位数由 SPI_CTRL0 决定。INST_L 字段。

©Levetop Semiconductor Co., Ltd.

26.6.2.30. XIP 控制寄存器（XIPCR）

地址：QSPIn_BASEADDR+0x0000_0108

	31	30	29	28	27	26	25	24
读:	0	0	XIP_PREFETCH_EN	0	XIP_MBL[1:0]		保留	
写:								
复位:	0	0	1	0	1	0	0	0

	23	22	21	20	19	18	17	16
读:	CONT_XFER_EN	INST_EN	保留				WAIT_CYCLES[4:3]	
写:								
复位:	1	1	0	0	0	0	0	1

	15	14	13	12	11	10	9	8
读:	WAIT_CYCLES[2:0]			MD_BITS_EN	0	INST_L[1:0]		0
写:								
复位:	0	0	0	0	0	1	0	0

	7	6	5	4	3	2	1	0
读:	ADDR_L[3:0]				TRANS_TYPE		FRF	
写:								
复位:	0	1	1	0	0	0	1	0

= 写入没有影响，访问终止，没有传输错误异常。

图 26-32：XIP 控制寄存器（XIPCR）

此寄存器仅在 SSIC_CONCURRENT_XIP_EN 等于 1 时有效。此寄存器用于存储 XIP 传输将在并发模式下使用的控制信息。

XIP_PREFETCH_EN

1 = 在 SSI 中启用 XIP 预取功能。

0 = 在 SSI 中禁用 XIP 预取功能。

XIP_MBL[1:0] — XIP Mode bits length.

设置 XIP 模式操作中的模式位的长度。这些位仅在 XIP_CTRL 时有效。XIP_MD_BIT_EN 被设置为 1。

0x0（MBL_2）：模式位长度等于 2

0x1（MBL_4）：模式位长度等于 4

0x2（MBL_8）：模式位长度等于 8

0x3（MBL_16）：模式位长度等于 16

CONT_XFER_EN

1 = 在 XIP 模式下启用连续传输。

0 = 禁用 XIP 模式下的连续传输。

©Levetop Semiconductor Co., Ltd.

INST_EN

1 = XIP 传输将有指令阶段。

0 = XIP 传输将没有指令阶段。

WAIT_CYCLES[4:0] — Wait cycles in Dual/Quad/Octal mode between control frames transmit and data reception. Specified as number of SPI clock cycles.

MD_BITS_EN — Mode bits enable in XIP mode.

1 = 在地址阶段之后的插入模式位。

0 = 在地址阶段后的无模式位。

INST_L[1:0]

双/八进制指令长度。

0x0（INST_L0）：无指令

0x1（INST_L4）：4 位指令长度

0x2（INST_L8）：8 位指令长度

0x3（INST_L16）：16 位指令长度

ADDR_L[3:0]

此位定义了要传输的地址的长度。只有在这么多的位被编程到 FIFO 后，传输才能开始。

0x0（ADDR_L0）：无地址

0x1（ADDR_L4）：4 位地址长度

0x2（ADDR_L8）：8 位地址长度

0x3（ADDR_L12）：12 位地址长度

0x4（ADDR_L16）：16 位地址长度

0x5（ADDR_L20）：20 位地址长度

0x6（ADDR_L24）：24 位地址长度

0x7（ADDR_L28）：28 位地址长度

0x8（ADDR_L32）：32 位地址长度

0x9（ADDR_L36）：36 位地址长度

0xA（ADDR_L40）：40 位地址长度

0xB（ADDR_L44）：44 位地址长度

0xC（ADDR_L48）：48 位地址长度

0xD（ADDR_L52）：52 位地址长度

0xE（ADDR_L56）：56 位地址长度

0xF（ADDR_L60）：60 位地址长度

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

TRANS_TYPE[1:0]

地址和指令传输格式。

选择 SSI 是否将在标准 SPI 模式或在 CTRLR0 中选择的 SPI 模式下传输指令/地址。SPI_FRF 字段。

0x0（TT0）：指令和地址将以标准 SPI 模式发送。

0x1（TT1）：指令将以标准 SPI 模式发送，地址将以 XIP_CTRL 指定的模式发送。SPI_FRF.

0x2（TT2）：指令和地址都将以 XIP_CTRL 指定的方式发送。FRF.

0x3（TT3）：保留。

FRF[1:0] — SPI Frame Format

选择要传输/接收数据的数据帧格式。

0x0（RSVD）：保留

0x1（SPI_DUAL）：双 SPI 格式

0x2（SPI_QUAD）：四轴 SPI 格式

0x3（SPI_OCTAL）：八进制 SPI 格式

26.6.2.31. XIP 从属服务器启用寄存器（XIPSER）

地址：QSPIn_BASEADDR+0x0000_010C

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0							SER
写:								
复位:	0	0	0	0	0	0	0	1

= 写入没有影响，访问终止，没有传输错误异常。

图 26-33：XIP 从属服务器启用寄存器（XIPSER）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

此寄存器仅在 SSIC_CONCURRENT_XIP_EN 等于 1 时有效。该寄存器使单个从服务器能够从 SSI 主服务器中选择输出行 XIP 操作模式。在 SSI 主服务器上最多有 16 个从属选择输出引脚。当 SSI 繁忙或 SSIC_EN = 1 时，您无法写入此寄存器。

SER — Slave Select Enable Flag.

该寄存器中的每个位对应于 SSI 主服务器的从选择行（ss_x_n）。当设置这个寄存器中的一个位(1)时，当 XIP 传输开始时，主服务器对应的从选择行被激活。应该注意的是，在 XIP 传输开始之前，设置或清除这个寄存器中的位对相应的从属选择输出没有影响。在开始传输之前，您应该启用此寄存器中与主服务器希望与其通信的从设备相对应的位。当不在广播模式下操作时，在此字段中只能设置一个位。

1 = 选择

0 = 未选择

XIP 接收 FIFO 溢出中断清除寄存器（XRXIOCR）

26.6.2.32. XIP 接收 FIFO 溢出中断清除寄存器（XRXIOCR）

地址：QSPIn_BASEADDR+0x0000_0110

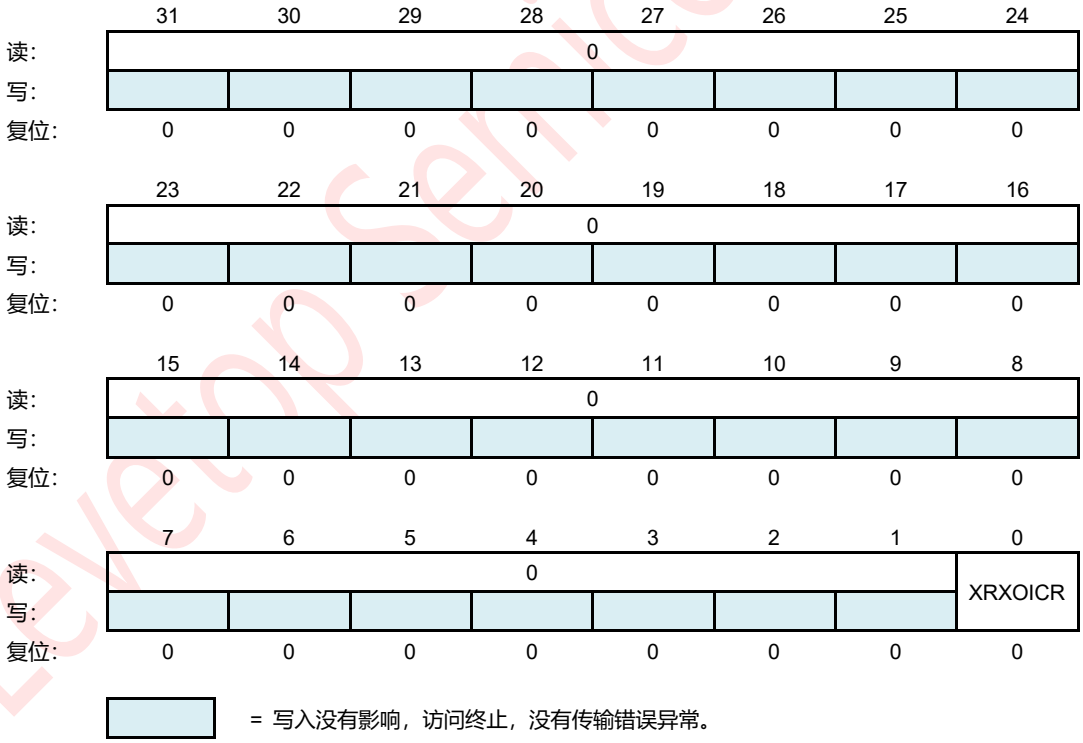


图 26-34：XIP 接收 FIFO 溢出中断清除寄存器（XRXIOCR）

XRXOICR — Clear XIP Receive FIFO Overflow Interrupt.

这个寄存器反映了中断的状态。从这个寄存器读取会清除 ssi_xrxo_intr（_n）中断；写入没有效果。

©Levetop Semiconductor Co., Ltd.

26.6.2.33. XIP 连续传输超时寄存器（XIP）

地址：QSPIn_BASEADDR+0x0000_0114

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	XTOC[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 26-35：XIP 连续传输超时寄存器（列表列表）

XIP 计数下寄存器为连续模式。该计数器用于在连续传输模式下取消选择从属服务器。当启用 SSI 时，不可能写入此寄存器（SSIC_EN = 1）。通过写入 SSIENR 寄存器来启用和禁用 SSI。

XTOC[7:0] — XIP time out value in terms of hclk.

一旦在连续 XIP 模式中选择了从属器，如果在计数器中没有指定的时间请求，该计数器将用于取消选择从属器。

©Levetop Semiconductor Co., Ltd.

26.7. 功能描述

26.7.1. 主模式

此模式允许与串行从属外围设备进行串行通信。当配置为串行主设备时，SSI 将启动并控制所有串行传输。图 26-36 显示了 SSI 配置为串行主服务器的示例，以及串行总线上配置为串行从线服务器的所有其他设备。

由 SSI 产生和控制的串行位率时钟在 sclk_out 线上被输出。当 SSI 被禁用（SSIC_EN = 0）时，可能不会发生串行传输，并且 sclk_out 处于“非活动”状态，这是由它操作的串行协议定义的。

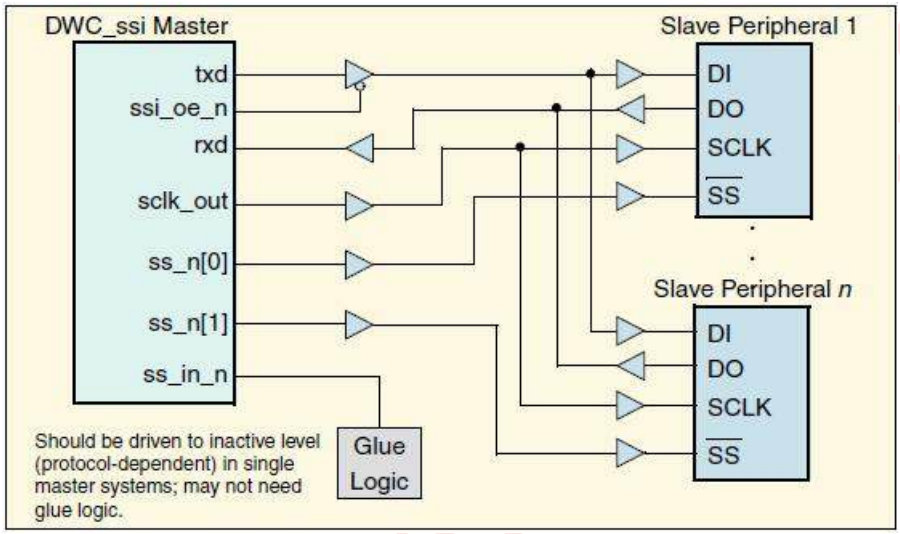


图 26-36：SSI 配置为主设备

26.7.2. 时钟比率

SSI 致力于一个过采样的体系结构。对于主操作模式，外围时钟（sclk_out）周期是内部核心时钟（ssi_clk）的倍数。

当 SSI 宏单元被配置为主设备时，比特率时钟（sclk_out）的最大频率为 ssi_clk 频率的一半。这是为了允许移位控制逻辑捕获 sclk_out 的一个时钟边缘上的数据，并在相反的边缘上传播数据。

sclk_out 的频率可以由以下方程推导出。

$$F_{sclk_out} = \frac{F_{ssi_clk}}{SCKDV}$$

SCKDV 是一个可编程寄存器，保持在 0-65,534 范围内的任何偶数值。如果是 SCKDV = 0，则 sclk_out 将被禁用。

©Levetop Semiconductor Co., Ltd.

26.7.3. 接收和传输 FIFO 缓冲区

SSI 使用的 FIFO 缓冲区是内部的 d 型人字拖。由于串行规范，传输和接收 FIFO 缓冲器的宽度都固定在 32 位，这说明串行传输（数据帧）的长度可以是 4 到 32 位。当写入传输 FIFO 缓冲区时，大小小于 32 位的数据帧必须是正确正确的。移位控制逻辑自动正确证明接收 FIFO 缓冲区中的接收数据。

26.7.3.1. 传输 FIFO

传输的 FIFO 通过 AHB 写入命令加载到 SSI 数据寄存器（DR）。数据通过移位控制逻辑从发射 FIFO 中弹出（移除）到发射移位寄存器中。当 FIFO 中的条目数小于或等于 FIFO 阈值时，发送 FIFO 生成 FIFO 空中断请求（ssi_txe_intr）。通过可编程寄存器 TXFTLR 设置的阈值决定了产生中断的 FIFO 条目的级别。该阈值允许您向处理器提供发送 FIFO 几乎为空的早期指示。如果您试图将数据写入已经完整的传输 FIFO，则会生成传输 FIFO 溢出中断（ssi_txo_intr）。

26.7.3.2. 接收 FIFO

数据通过 AHB 读取命令从接收的 FIFO 弹出到 SSI 数据寄存器（DR）。通过移位控制逻辑从接收移位寄存器加载接收 FIFO。当 FIFO 中的条目数大于或等于 FIFO 阈值加 1 时，接收 FIFO 生成一个 FIFO-满中断请求（ssi_rxf_intr）。通过可编程的寄存器 RXFTLR 设置的阈值决定了生成中断的 FIFO 条目的级别。

该阈值允许您向处理器提供接收 FIFO 已接近满的早期指示。当接收移位逻辑尝试将数据加载到一个完全完整的接收 FIFO 中时，将生成一个接收 FIFO 溢出中断（ssi_rxo_intr）。然而，这些新收到的数据却丢失了。如果您试图从空的接收 FIFO 中读取，则会生成接收 FIFO 下流中断（ssi_rxu_intr）。这将提醒处理器读取的数据无效。

26.7.4. DMA 操作

SSI 具有可选的内置 DMA 功能，可以在配置时进行选择；它有一个到 DMA 控制器的握手接口来请求和控制传输。AHB 总线用于执行与 DMA 之间的数据传输。虽然 SSI DMA 操作以一种通用的方式设计，以尽可能容易地适应任何 DMA 控制器，但它被设计为无缝地工作，并最好地使用。要在 SSI 上启用 DMA 控制器接口，您必须写入 DMA 控制寄存器（DMACR）。将一个 1 写入 DMACR 寄存器的 TDMAE 位字段，可以实现 SSI 传输握手接口。将 1 写入 DMACR 寄存器的 RDMAE 位字段，允许 SSI 接收握手接口。

26.7.5. SSI 中断

SSI 中断的描述如下：

传输 FIFO 空中断（ssi_txe_intr）-当传输 FIFO 等于或低于其阈值，并需要服务以防止运行不足时进行设置。通过软件可编程寄存器设置的阈值决定了产生中断的传输 FIFO 条目的级别。当数据被写入传输 FIFO 缓冲区时，硬件会清除此中断，使其超过阈值级别。

传输 FIFO 溢出中断（ssi_txo_intr）-当 AHB 访问在传输 FIFO 被完全填充后尝试写入时进行设置。当设置时，从 AHB 写入的数据将被丢弃。这个中断保持设置，直到您读取传输 FIFO 溢出中断清除寄存器（TXOICR）。

接收 FIFO 完全中断（ssi_rxf_intr）-当接收 FIFO 等于或高于其阈值加上 1，并且需要提供服务以防止溢出时进

©Levetop Semiconductor Co., Ltd.

行设置。通过软件可编程寄存器设置的阈值决定了产生中断的接收 FIFO 条目的级别。当从接收 FIFO 缓冲区读取数据时，硬件会清除此中断，使其低于阈值水平。

接收 FIFO 溢出中断（ssi_rxo_intr）-当接收逻辑在尝试将数据完全填充后的数据输入接收 FIFO 时进行设置。在设置后，新接收到的数据将被丢弃。此中断保持设置，直到读取接收 FIFO 溢出中断清除寄存器（RXOICR）。

接收 FIFO 下流中断（ssi_rxu_intr）-当 AHB 访问试图从接收 FIFO 为空时读取时设置。设置后，将从接收 FIFO 读回零。这个中断保持设置，直到您读取接收 FIFO 下流中断清除寄存器（RXUICR）。

组合中断请求（ssi_intr）-上述所有中断请求的结果。要屏蔽此中断信号，您必须屏蔽所有其他 SSI 中断请求。

26.7.6. 增强的 SPI 模式

SSI 支持使用 SSIC_SPI_MODE 配置参数的 SPI 的双、四元和八进制模式。该参数的可能值是标准、双 SPI、四 SPI 和八进 SPI 模式。当该参数选择双、四、四或八进制模式时，TXD、RXD 和 ssi_oe_n 信号的宽度分别变为 2、4 或 8。因此，数据被转移到多个线路上，从而增加了总体吞吐量。除了 TXD、RXD 和 ssi_oe_n 信号的宽度外，双 SPI、四位或八位 SPI 模式的功能类似。可以使用 CTRLR0 选择操作模式（写/读取）。TMOD 字段。

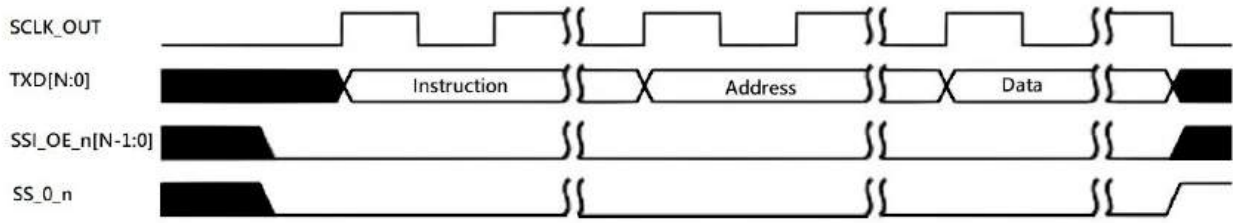


图 26-37：典型的写入操作双/四/八进制 SPI 模式

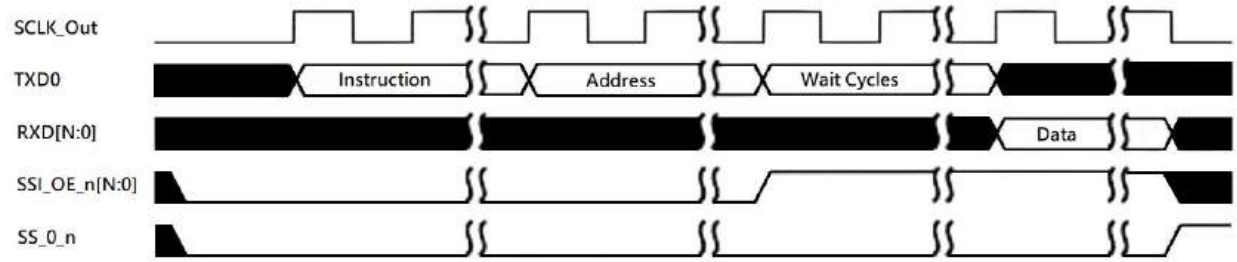


图 26-38：典型的读取操作双/四/八进制 SPI 模式

©Levetop Semiconductor Co., Ltd.

26.7.7. 就地执行（XIP）模式

SSI 提供了一个可以直接从 AHB 事务中执行内存读取操作的功能。这被称为就地执行模式，在其中，SSI 充当到 SPI 内存的内存映射接口。通过选择配置参数 SSIC_XIP_EN，可以在 SSI 中启用 XIP 模式。这包括一个 AHB 接口上额外的边带信号 xip_en。这个信号电平决定了 AHB 传输是寄存器读写还是 XIP 读取。在 XIP 操作期间，只支持 AHB read。如果 xip_en 信号被驱动为 1，那么 SSI 期望在 AHB 接口上发出一个读取请求。此请求被转换为串行接口上的 SPI 读取。一旦收到数据，它就会返回到同一事务中的 AHB 接口。haddr 用于推导出要在 SPI 接口上发送的地址。某些设备期望在 XIP 传输期间存在指令阶段。SSI 支持在 XIP 操作模式期间包含一些固定的指令集。

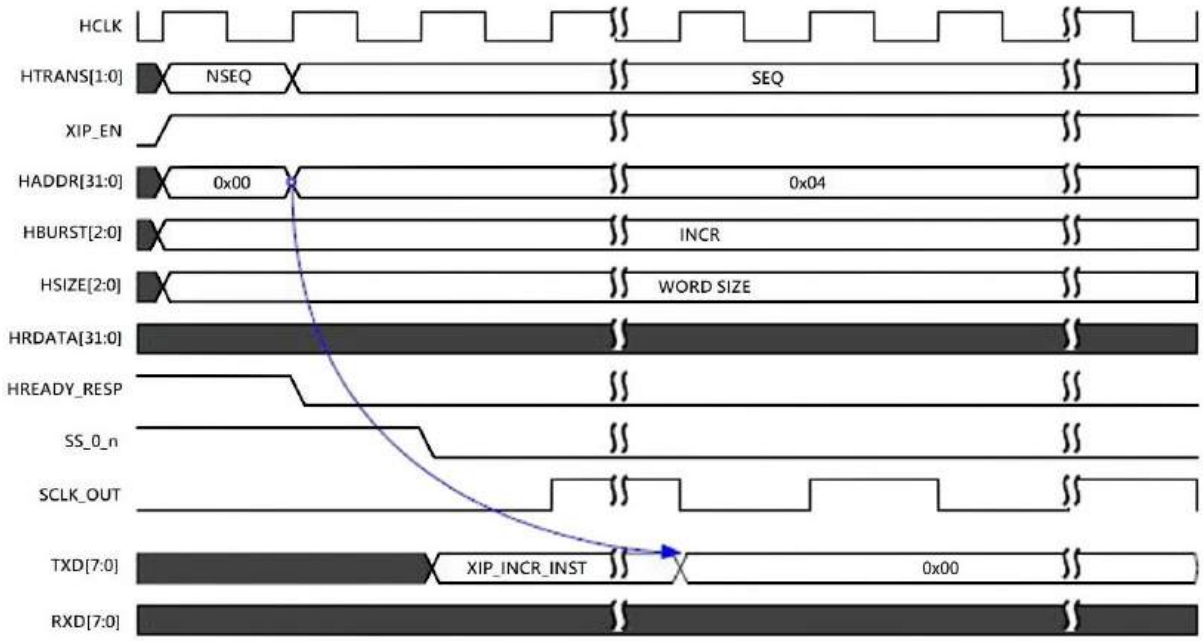


图 26-39：XIP 传输与指令阶段

26.7.8. 在 XIP 中的连续传输模式

当 SSI 接收到一个 XIP 请求时，来自 AHB 接口的地址将被直接传输到 SPI 接口上。AHB 接口上的每个新传输（XIP Read）都以相同的方式处理。因此，对于每个请求，都必须向设备发送一个新的地址，从而导致系统的延迟。

如果一个存储器设备允许在 XIP 读取传输之间拉伸从属选择信号，那么 SSI 可以被编程为连续的 XIP 模式，以实现更高的性能。在这种模式下，主机通过确保该命令和地址不被重新传输，并且主机控制器不需要等待这些突发之间的任何虚拟周期，将两个或多个 AHB 突发请求融合到单个 SPI 命令中。

当此功能被启用时，一旦收到第一个 XIP 命令，SSI 功能就处于连续 XIP 模式。对于第一个 XIP 传输，该地址将在 SPI 接口上发送。在接收到请求的数据后，SSI 继续保持从从被选择，时钟（sclk_out）仍然处于默认状态。对于 AHB 接口上的后续 XIP 传输，SSI 恢复时钟（sclk_out），命令和地址都不会传输到 SPI 接口，以及将立即从设备获取的数据（没有虚拟周期）。

©Levetop Semiconductor Co., Ltd.

•在连续读取模式下，不支持未定义的 INCR（hburst = 001）突发。

在连续传输过程中，由于从设备一直被选择，所以在从设备上消耗了大量的功率。为了避免这种情况，SSI 提供了一个配置选项，使监视狗计时器在计数器用完后取消选择从属器。

SSI 可以在以下条件下取消选择从属服务器：

- 在 XIP 接口上接收非 XIP 命令（实际上任何 xip_en 驱动为 0 的 AHB 事务）。
- 当 AHB 事务是一个非连续的地址时，从属选择将被删除，然后 SSI 启动一个新的 XIP 请求。
- 在 XIP_CNT_TIME_OUT 寄存器中指定的时间段内，SSI 没有检测到 AHB 接口上的任何 XIP 传输。

26.7.9. 在 XIP 操作中的数据预取

使用 SSI 中的数据预取功能，控制器在当前 XIP 事务中对连续突发的数据进行预取。如果对连续地址发出下一个事务请求，则可以直接从 RX FIFO 读取数据，而不是等待将新的地址和数据发送到设备。这提高了系统的整体性能。

要预取的数据量应该等于最后一个 AHB 请求的突发长度或 FIFO 深度（以较低者为准）。例如，如果 AHB 定义了从地址 0x00 开始的 16 个突发长度，那么 SSI 获取 16 个突发以完成当前传输，然后再次获取 16 个数据帧并保存到数据寄存器中。如果 AHB 总线再次请求从结束地址开始的数据以进行最后一次传输，那么其余的数据将从 RX FIFO 本身发送到设备。同时，SSI 再次启动向设备的 XIP 传输，以预取下一个数据块。如果 AHB 主服务器没有定义连续地址，则当前数据将从 FIFO 中刷新，并且控制器将启动一个新的事务。

当 SSI 完成当前的突发，并为下一个突发预取数据时，可能会放置一个新的 XIP 请求。在这种情况下，SSI 可以终止当前的传输，或者根据地址增加要获取的数据拍数。

- 如果启用了 XIP 预抓取，则不允许请求增量传输未定义长度（hburst = 3‘b001）的 AHB 请求。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

27.Inter-Integrated Circuit（I2C）

27.1. 功能介绍

I2C 是一种双线、双向串行总线，它提供了一种简单、有效的数据交换方法，最大限度地减少了设备之间的互连。该总线适用于许多设备之间短距离通信的应用程序。灵活的 I2C 允许将额外的设备连接到总线上，以进行扩展和系统开发。

这两条总线在标准模式下提供高达 100Kbits/s 的数据传输速率，在快速模式下提供的数据传输速率高达 400Kbits/s，在高速模式下提供高达 3.4Mbits/s 的数据传输速率。

I2C 系统是一个真正的多主总线，包括仲裁和碰撞检测，可以在多个设备试图同时控制总线时防止数据损坏。该特性支持具有多处理器控制的复杂应用程序，并可用于通过到装配线计算机的外部连接来快速测试和对齐最终产品。

27.2. I2C 特点

- 支持 7 位寻址。
- 支持标准模式、快速模式和高速模式
- 软件选项，可在高速模式和标准/快速模式之间进行选择
- 兼容 I2C 总线版本 2.1 标准的标准和快速模式。
- 多主操作。
- 软件可编程的 64 个不同的串行时钟频率之一。
- 软件-可选择的承认位。
- 中断驱动的、逐字节的数据传输。
- 仲裁丢失中断与自动模式切换从主到从。
- 传输完成和读取配置中断。
- 启动和停止信号的产生/检测。
- 重复的启动信号的产生。
- 确认位的生成/检测。
- 总线繁忙的检测。
- 在系统时钟停止模式时启用选项从地址接收
- 支持 SCL 或 SDA 行 GPIO 功能

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

27.3. 系统和方块图

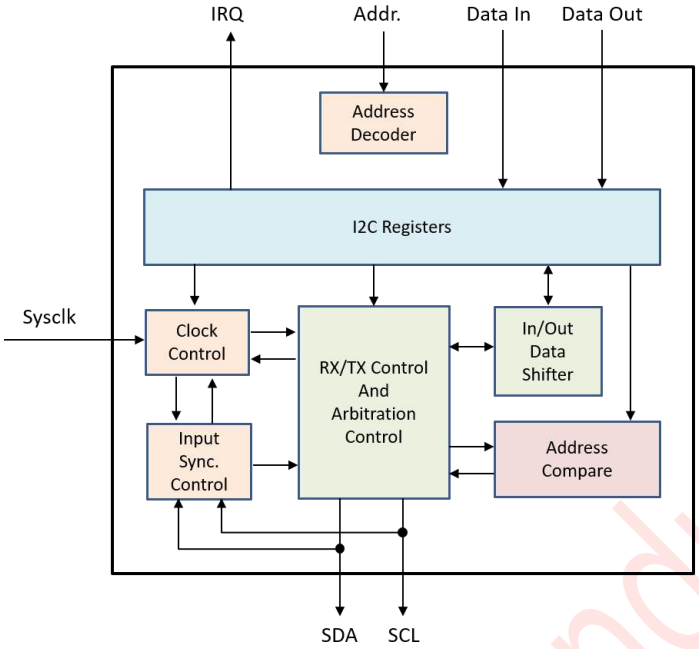


图 27-1：I2C 方块图

27.4. 内存映射和寄存器

27.4.1. 内存映射

I2C 模块的基本地址为 0x4015_0000。在 I2C 内存映射中有 10 个寄存器，如表 27-1 所示.

表 27-1：I2C 内存映射

Address Offset	Bits[7:0]
0x0000	I2C Status Register (I2CS)
0x0001	I2C Clock Prescaler Register (I2CP)
0x0002	I2C Control Register (I2CC)
0x0003	I2C Slave Address Register (I2CSA)
0x0004	I2C Port Control Register (I2CPCR)
0x0005	I2C Slave High-speed Indicator Register
0x0006	I2C Slave SDA Hold Time Register
0x0007	I2C Data Register (I2CD)
0x0008	Reserved
0x0009	Reserved
0x000A	I2C Port Direction Register (I2CDDR)
0x000B	I2C Port Data Register (I2CPDR)

©Levetop Semiconductor Co., Ltd.

27.4.2. 寄存器描述

27.4.2.1. I2C 状态寄存器（I2CS）

地址：I2C_BASEADDR+0x0000_0000

	7	6	5	4	3	2	1	0
读:	AACK	DACK	RXTX	ARBL	BBUSY	AASLV	RC	TF
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 27-2：I2CS 寄存器显示 I2C 模块的状态。

AACK — Address acknowledge error.

该位表示在地址阶段中，主节点是否检测到确认位。它由地址阶段的第九个时钟的上升边缘设置，并通过改变 MSMOD 从 1 到 0 或重复启动条件来清除。

0 = 无地址确认错误。

1 = 地址确认错误。

DACK — Data acknowledge received.

该位表示在地址或数据传输期间是否检测到确认位。它对第九个时钟的上升边缘是有效的。

0 = 未接收到任何确认位。

1 = 接收到确认位。

RXTX — Receive or transmit.

表示 I2C 模块的接收器或发送器的功能。它在第八个时钟的下降边缘是有效的。

0 = 接收，接收数据。

1 = 传输，传输数据。

ARBL — Arbitration lost

这一位显示了总线的仲裁状态。它将在以下情况下被设置，在此期间

SCL 高:

1. 当主节点在启动条件、地址周期、数据传输周期或停止条件下驱动高时，SDA 采样较低。
2. 当在数据接收周期的确认位期间主驱动高时，SDA 采样低。
3. 在总线繁忙时尝试启动周期。

ARBL 必须通过软件读取 I2CS 寄存器来清除。

0 = 仲裁未丢失。

1 = 仲裁失败。

©Levetop Semiconductor Co., Ltd.

BBUSY — I2C bus busy

显示总线状态。

0 = 总线空闲。已从停止位清除。

1 = 巴士很忙。它是从起始位开始设置的。

AASLV — Addressed as a slave

如果 I2C 模块被定位为从地址，并且它自己的从地址与 SDL 上接收的调用地址匹配，那么这个位被设置在 8h 时钟的下降边缘上。如果检测到启动位或停止位，则被清除

0 = 不是作为一个奴隶。

1 = 被称为奴隶。

RC — Receive Complete

此位表示是时候配置接收器了。对于主接收器，无论是否检测到确认位，它都设置在传输的数据或地址字节的第九时钟的下降边缘。对于从接收器，如果检测到确认位，则该位将由接收到的地址或数据字节的第 9 个时钟的下降边来设置。如果 I2CC 中的 IEN 位，将生成一个中断。通过使用 RC 设置读取 I2CS 来清除 RC，然后写入 I2CC。

0 = 无意义，

1 = 表示需要配置接收器。

TF — Transfer Complete Flag

此位表示有传输或接收的数据。对于接收器，它由未接收到的数据（非地址）字节的第九个时钟的下降边设置，无论是否检测到确认位。对于主发送器，该位由所发送的数据或地址字节的第九个时钟的下降边设置，无论是否检测到确认位。对于从发送器，如果检测到确认位，则该位将由所传输的地址或数据字节的第 9 个时钟的下降边来设置。如果 I2CC 中的 IEN 位，将生成一个中断。通过读取带有 TF 设置的 I2CS 来清除 TF，然后访问 I2CD 或主传输模式来写入 I2CC。

0 = 无意义，

1 = 数据或地址传输完成。

27.4.2.2. I2C 时钟电压前寄存器（I2CP）

地址：I2C_BASEADDR+0x0000_0001

	7	6	5	4	3	2	1	0
读:	0	TEST	PRE[5:0]					
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 27-3：I2C 时钟计算前寄存器（I2CP）

©Levetop Semiconductor Co., Ltd.

I2CP 是一个预调用器，它为数据收发器生成一个比特率时钟。由于潜在的缓慢的 SCL 和 SDA 的上升和下降时间，总线信号在预调量器的频率下被采样。串行位时钟频率等于 OPB 时钟除以分频器。

TEST — Clock Test Enable

此位启用了 I2C 时钟的测试模式。在正常模式下，其频率为 fipgclk/（396x（PRE[5:0]+1）+1）。在测试模式下，频率频率为/（8x（PRE[5:0]+1）+1）。

1 = 时钟测试模式已启用

0 = 正常模式

PRE[5:0] — Prescaler Divider Value

27.4.2.3. I2C 控制寄存器（I2CC）

地址：I2C_BASEADDR+0x0000_0002

	7	6	5	4	3	2	1	0
读: 写:	SLV_HSIE	HMS_EN	AMIE	REPSTA	ACKEN	MSMOD	IEN	EN
复位:	0	0	0	0	1	0	0	0

图 27-4：I2C 控制寄存器（I2CC）

I2CC 用于控制 I2C 模块的工作。

SLV_HSIE — Slave High-Speed Mode Interrupt Enable

如果 I2C 中断启用控制（IEN）设置为 1，该位选择从属高速模式状态是否会产生中断请求 s。

1 = 启用从属高速模式状态请求。

0 = 禁用从属高速模式状态请求。

HSM_EN — High Speed Mode Enable

此位元为主模式下的模块选择高速模式或快速/标准模式操作。HSM_EN 位应设置为选择高速模式操作。主代码在 F/S 模式下传输且 HS_I2C 不丢失仲裁后，应通过软件设置 HSM_EN，如果用户准备清除 MSMOD 以传输 STOP，则应在 MSMOD 后使用软件清除 HSM_EN。

0 = 快速/标准模式操作（默认）

1 = 高速模式操作

AMIE — Address Match Interrupt Enable

如果 I2C 中断启用控制（IEN）设置为 1，此位选择从地址匹配是否会产生中断请求。在进入停止模式之前应该设置 AMIE，当从地址匹配时唤醒系统，必须在正常工作模式下清除。

1 = 启用地址匹配中断请求。

0 = 禁用地址匹配中断请求。

©Levetop Semiconductor Co., Ltd.

REPSTA — Repeat Start

在这些情况下：1)在发送从地址或数据字节后，主接收器不被确认，2)主发送器发送了一个地址或数据字节，无论是否被确认，主服务器可以重复开始信号，然后是一个新的从地址。在设置 REPSTA 位后，在配置从地址（通过写入 I2CD）时，将发送重复开始位。

1 = 产生重复启动。

0 = 没有重复启动。

ACKEN — Acknowledge enable control.

这个位指定在主和从接收器的确认周期中驱动到 SDA 的值。请注意，ACKEN 位仅在 I2C 总线接收到数据字节时适用。在接收地址期间，如果接收到的地址与从地址匹配，则确认位将被发送，而不管 ACKEN 位如何。

1 = 在接收到一个字节的数据后，在第九个时钟位向 SDA 发送一个确认信号。

0 = 在接收到一个字节的数据后，在第九个时钟位没有向 SDA 发送确认信号。

MSMOD — I2C master/slave mode selection control.

将 MSMOD 从 1 更改为 0 将在总线上生成一个停止，并选择从属模式。将 MSMOD 从 0 更改为 1 将在总线上生成一个启动，并选择主模式。

1 = 主模式。

0 = 从属模式。

IEN — I2C interrupt enable control.

它在设置 I2C 中断时启用 I2C 中断。

1 = I2C 中断。

0 = I2C 中断已被禁用。

EN — I2C enable/disable control.

1 = 模块。

0 = 模块已被禁用。

它可以启用/禁用该模块，并控制整个 I2C 模块的软件复位。设置位生成对模块的内部复位，在设置位的 2 个时钟后断言，并为 3 个时钟保持断言。因此，在设置 EN 位的 5 个时钟后，复位被否定。

如果该模块在字节传输的中间被启用，则从属模式将忽略当前的 I2C 总线传输，并在检测到下一个启动条件时开始运行。主模式不知道总线很忙，因此启动一个启动周期可能会破坏当前的总线周期，最终导致当前的主模块或 I2C 模块失去仲裁，之后总线操作恢复正常。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

27.4.2.4. I2C 从属地址寄存器（I2CSA）

地址：I2C_BASEADDR+0x0000_0003

	7	6	5	4	3	2	1	0
读:	ADDR[7:1]							0
写:								
复位:	0	1	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 27-5：I2C 从属地址寄存器（I2CSA）

当 I2C 作为从服务器工作时，I2CSA 存储该地址，并响应主服务器发送的地址。

ADDR[7:1] — Module Slave Address.

这些位表示当 I2C 模块处于从属模式时的从属地址。(I2C 默认为从属状态）。

27.4.2.5. I2C 端口控制寄存器（I2CPCR）

地址：I2C_BASEADDR+0x0000_0004

	7	6	5	4	3	2	1	0
读:	SDAPA	SCLPA	WOMI2C[1:0]		Reserved		PUI2C[1:0]	
写:								
复位:	0	0	0	0	0	0	1	1

图 27-6：I2C 端口控制寄存器（I2CPCR）

SDAPA — SDA Port Assignment Bit.

读/写位选择 SDA 的功能模式。

1 = Pin 被配置为 GPIO_SDA。

0 = Pin 被配置为主要功能。

SCLPA — SCL Port Assignment Bit.

读/写位选择 SCL 的功能模式。

1 = Pin 被配置为 GPIO_SCL。

0 = Pin 被配置为主要功能。

WOMI2C[1:0] — Wired-OR Mode Bits

读/写位将相应的 I2C 引脚设置为开漏驱动器模式。WOMI2C[1]用于 SDA 引脚，WOMI2C[0]用于 SCL 引脚。这些位仅在 GPIO 模式下可用

1 = 输出时为 Open Drain。

0 = 输出时为 CMOS 驱动器。

©Levetop Semiconductor Co., Ltd.

Levetop Semiconductor

PUI2C[1:0] — Pull-up Enable Bits

读写位使相应的 I2C 引脚的脉冲成为。PUI2C[1]用于 SDA 引脚，PUI2C[0]用于 SCL 引脚。这些位在 GPIO 输入模式和主功能模式下都可用。

0 拔起。

0 = 拔拔已禁用。

27.4.2.6. I2C 从属高速模式指示器寄存器（I2CSHIR）

地址：I2C_BASEADDR+0x0000_0005

	7	6	5	4	3	2	1	0
读:	0	0	0	0	0	0	0	SLV_HS
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 27-7：I2C 从属高速模式指示器寄存器（I2CSHIR）

SLV_HS — Slave High Speed Mode.

当选择 I2C 作为从机模块时，此位表示该模块是被选择为高速模式还是快速/标准模式的数据传输。此位设置在主代码的第九 SCL 上升边，接收非确认位。此位必须通过写入 1 来清除，否则 SCL 行将被迫为低状态。

0 = I2C 选择用于快速/标准数据传输（默认）。

1 = I2C 选择用于高速模式数据传输。

27.4.2.7. I2C 从属 SDA 保持时间寄存器（I2CSHT）

地址：I2C_BASEADDR+0x0000_0006

	7	6	5	4	3	2	1	0
读:	SCL_FILTER_EN	SDA_FILTER_EN	SLVHT					
写:								
复位:	0	0	0	0	1	0	0	1

图 27-8：I2C 从属 SDA 保持时间寄存器（I2CSHT）

SCL_FILTER_EN — SCL Filter enable.

如果启用了 SCL 滤波器，当 HS 模式传输正在进行时，在 SCL 线上发生的 10ns 脉冲将被过滤掉。如果 F/S 模式传输正在进行中，则在 SCL 线上发生的 50ns 脉冲将被滤波。

SDA_FILTER_EN — SDA Filter enable.

如果启用了 SDA 滤波器，当 HS 模式传输正在进行时，在 SDA 线上发生的 10ns 脉冲将被过滤掉。如果 F/S 模式传输正在进行中，则在 SDA 线上发生的 50ns 脉冲将被滤波。

©Levetop Semiconductor Co., Ltd.

SLVHT — slave SDA line hold time configuration.

当 I2C 作为从输出模式时，SCL 下降边缘后的数据会发生变化，内部 SDA 保持寄存器的值等于 SLVHT。不允许写入值为 0。

27.4.2.8. I2C 数据寄存器（I2CD）

地址：I2C_BASEADDR+0x0000_0007

	7	6	5	4	3	2	1	0
读:	R[7:0]							
写:	T[7:0]							
复位:	0	0	0	0	0	0	0	0

图 27-9：I2C 数据寄存器（I2CD）

I2CD 寄存器保存要传输的数据（下一个字节）或接收到的数据。在主模式下，它还保存了要传输的从属地址和传输方向。来自从属地址的位[7:1]，位[0]是传输方向（R/W）。在主接收器模式下，读取 I2CD 允许读取发生并启动下一个字节数据接收。在主传输模式下，写入 I2CD 将存储下一个传输的字节。在从属模式下，相同的功能在它被处理后可用。

R[7:0] — Receive Bits [7:0]

T[7:0] — Transmit Bits [7:0]

27.4.2.9. I2C 端口方向寄存器（I2CDDR）

地址：I2C_BASEADDR+0x0000_000A

	7	6	5	4	3	2	1	0
读:	0	0	0	0	0	0	DDRI2C[1:0]	
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 27-10：I2C 端口方向寄存器（I2CDDR）

Read: Anytime

Write: Anytime

DDRI2C[1:0] — I2C Data Direction Bits

读/写位控制 I2C 引脚的数据方向。DDRI2C[1]用于 GPIO_SDA 针，DDRI2C[0]用于 GPIO_SCL 针。这些位仅在 GPIO 模式下可用。

1 = 相应的引脚配置为输出

0 = 对应的引脚配置为输入

©Levetop Semiconductor Co., Ltd.

27.4.2.10. I2C 端口数据寄存器（I2CPDR）

地址：I2C_BASEADDR+0x0000_000B

	7	6	5	4	3	2	1	0
读:	0	0	0	0	0	0	PORTI2C[1:0]	
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 27-11：I2C 端口数据寄存器（I2CPDR）

Read: Anytime

Write: Anytime

PORTI2C[1:0] — I2C Port Data Bits

写入这些位设置为配置为 GPIO 的相应 I2C 引脚的输出数据。PORTI2C[1]用于 GPIO_SDA 针，PORTI2C[0]用于 GPIO_SCL 针。读取这个位会返回 I2C 引脚的级别。

27.5. 功能描述

I2C 模块是一种双线、双向串行总线，它提供了一种简单、有效的数据交换方法，最大限度地减少了设备之间的互连。软件可以轮询 I2C 状态标志。此外，I2C 操作也可以被中断驱动。当接收到字节或要传输字节时，将设置 I2CS 中的 TF 位，如果 I2CC 中的 IEN 位也被设置，则将产生一个中断。

如果仲裁在传输过程中丢失，则 ARBL 将被设置，如果 I2CC 中的 IEN 位也被设置，则将产生一个中断。如果没有来自从属服务器（AACK 集）的地址确认，则也可以生成一个中断。

如果该模块不希望接收下一个字节，则可以清除 ACKEN。

27.5.1. 主模式

当 I2C 模块作为主线工作时，如果总线是空闲的（I2CS 的 BBUSY 位被清除），则 I2C 模块可以初始化传输。将 MSMOD 位从 0 更改为 1，将在总线上生成一个开始，并选择主模式。主控件控制传输的方向，即收发位。由主服务器发送的传输的第一个字节是从属地址，下面的字节是数据。如果在每第九周期时钟后没有收到确认，则将 MSMOD 位从 1 更改为 0 以在总线上生成停止。I2CP 中的 PRE[5:0]位控制 I2C 总线的比特率时钟。

通过在设置 REPSTA 位后配置从地址，主地址可以重复开始信号，而不是发出停止信号。

27.5.2. 从属模式

如果清除 MSMOD 位，则模块为从属，可以由其他主寻址。当一个获胜的主人试图解决它时，它将释放 SDA 线，并立即切换到它的从属模式。

提示：I2C 不能同时在从属模式和主模式下工作。

©Levetop Semiconductor Co., Ltd.

27.5.3. 通信协议

I2C 通信协议由六个组件组成：启动、数据源/接收方、数据方向、从属确认、数据、数据确认和停止。这些内容如图 27-12 所示， 并在下文中进行了描述。

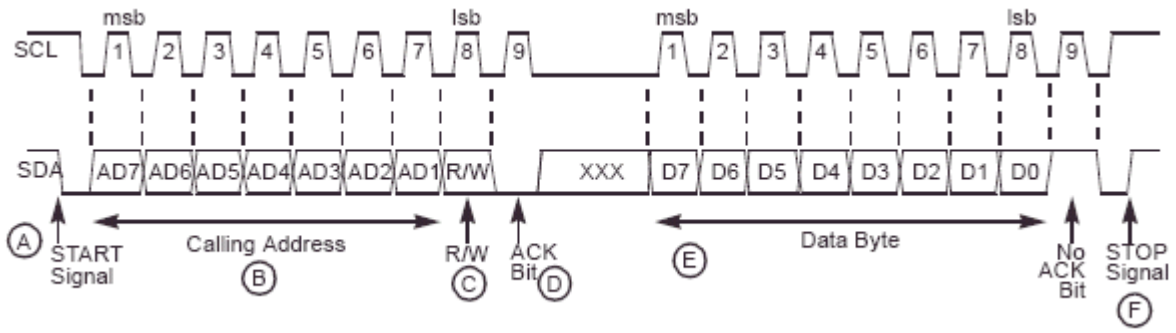


图 27-12：I2C 通信协议

1. 启动信号—当没有其他设备是总线主机（SCL 和 SDA 线都处于逻辑高值）时，设备可以通过发送启动信号来启动通信（参见图 27-12 中的 A).起始信号定义为 SCL 高时 SDA 的高到低转变。这个信号表示数据传输的开始（每个数据传输可以有几个字节长），并唤醒所有从服务器。

2. 从地址传输-主服务器在起始信号(B).之后的第一个字节中发送从地址在 7 位的调用地址之后，它发送 R/W 位(C)，告诉从服务器数据传输方向。

每个从属服务器都必须有一个唯一的地址。I2C 主服务器不能传输与其自己的从地址相同的地址；它不能同时是主和从。与主服务器发送的地址匹配的从服务器在第 9 个时钟(D)处降低 SDA 以返回一个确认位。

3. 数据传输—当成功实现从寻址后，数据传输可以按照调用主服务器发送的 R/W 位指定的方向进行(E)。

数据只能在 SCL 低时进行更改，并且在 SCL 高时必须保持稳定，如图 27-12 所示给…看 SCL 对每个数据位脉冲一次，首先发送 MSB。接收设备必须通过在第 9 个时钟时降低 SDA 来确认每个字节；因此，一个数据字节传输需要 9 个时钟脉冲。

如果它不承认主服务器，从接收器必须保持 SDA 高。然后，主机可以生成 STOP 信号以中止数据传输或生成启动信号（重复启动，如图 27-13 所示)来启动一个新的调用序列。

如果主接收器在字节传输后不承认从发送器，这意味着从端的数据结束。从服务器释放 SDA，为主服务器产生一个停止或启动信号。

4. 停止信号——主信号可以通过产生一个停止信号来释放总线来终止通信。STOP 信号被定义为 SDA 的低到高的过渡，而 SCL 处于逻辑高(F)。请注意，即使从服务器已经进行了确认，主服务器也可以生成一个 STOP，此时从服务器必须释放总线。

©Levetop Semiconductor Co., Ltd.

5. 主信号可以重复开始信号，而不是发出停止信号，然后是一个调用命令，(A 在图 27-13 中).当产生启动信号而不首先产生停止信号以结束通信时，就会发生重复启动。主服务器使用一个重复的启动来与另一个从服务器或以不同的模式（发送/接收模式）与同一从服务器进行通信，而不释放总线。

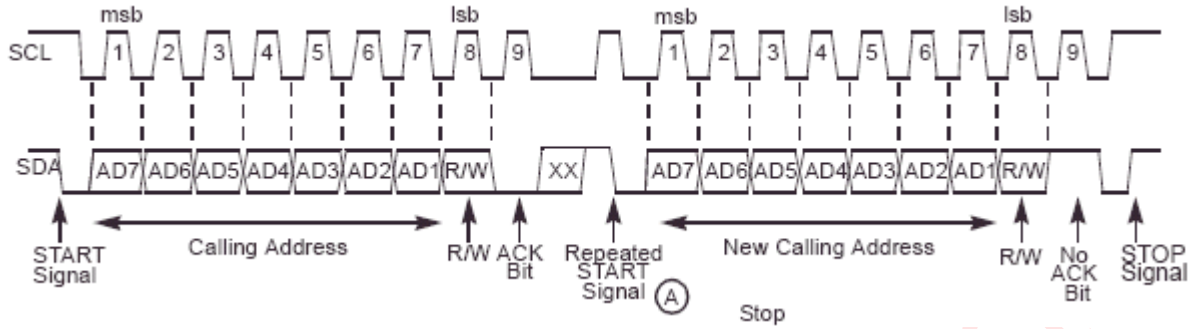


图 27-13：重复启动 I2C 协议

27.5.4. 仲裁程序

如果多个设备同时请求总线，则总线时钟由同步过程确定，其中低周期等于设备之间最长的时钟-低周期，高周期等于最短的周期。数据仲裁程序决定了竞争设备的相对优先级。如果一个设备发送逻辑高，而另一个设备发送逻辑低，则失去仲裁；它立即切换到从接收模式并停止驱动 SDA。

失去仲裁的主服务器可以产生时钟脉冲，直到它失去仲裁的字节结束。获胜的大师有可能正试图解决这个问题。因此，丢失的主模式必须立即切换到其从属模式。

在这种情况下，从主模式到从模式的转换不会生成 STOP 条件。同时，硬件设置 I2CSR 的 ARBL 位来表示仲裁失败。

27.5.5. 时钟同步

由于使用了有线与逻辑，所以 SCL 上的高到低转换会影响连接到总线的所有设备。当主驱动 SCL 低时，设备开始计算其低周期。当设备时钟变低时，它将 SCL 保持为低，直到达到时钟高状态。但是，如果另一个设备时钟仍处于其低周期内，则该设备时钟的从低到高的变化可能不会改变 SCL 的状态。

因此，低周期最长的设备保持同步时钟 SCL 较低。在此期间，低周期较短的设备将进入高等待状态（请参见图 27-14).当所有涉及的设备都计算出其低周期时，同步时钟 SCL 被释放并拉高。

然后设备时钟和 SCL 状态之间没有区别，所以所有设备开始计算它们的高周期。第一个完成其高周期的设备再次将 SCL 拉低。

27.5.6. 握手机制

时钟同步机制可以用作数据传输中的握手。从属设备可以在完成一个字节传输（9 位）后保持 SCL 较低。在这种情况下，时钟机制停止总线时钟，并强制主时钟进入等待状态，直到从机释放 SCL。

©Levetop Semiconductor Co., Ltd.

27.5.7. 时钟拉伸

从程序可以使用时钟同步机制来降低传输比特速率。在主驱动 SCL 低后，从可以将 SCL 驱动低，然后释放它。如果从属 SCL 低周期长于主 SCL 低周期，则所产生的 SCL 总线信号低周期将被拉伸。

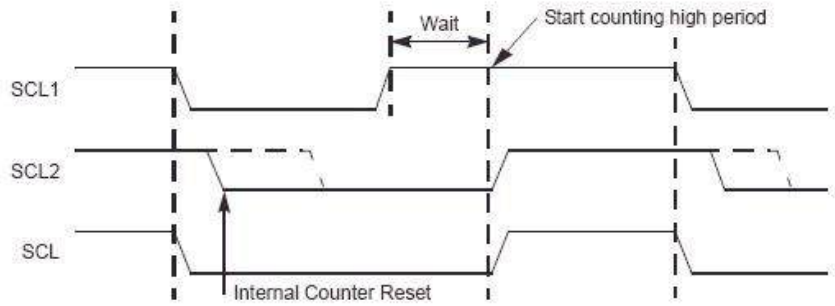


图 27-14：SCL 同步

27.5.8. 高速模式运行

I2C 在 hs 模式下的模块可以以高达 3.4 Mbits/s 的比特率传输信息，但它仍然完全向下兼容快速或标准模式（F/s 模式）设备，用于混合速度总线系统中的双向通信。除了在 HS 模式传输期间不执行仲裁和时钟同步外，还保持了与 F/S 模式系统相同的串行总线协议和数据格式。

hs 模式下的串行数据传输格式符合标准模式 I2c 总线规范。HS 模式只能在满足以下条件后开始（F/S 模式）：

1. 启动条件(S)
2. 8 位主代码（00001XXX）
3. 非确认位(A)

（参见图 27-15HS 模式下的数据传输格式）和（请参见图 27-16 A **完成 HS 模式传输）**请更详细地展示这一点。HS 主代码有两个主要功能：

1. 它允许在 F/s 模式速度下的竞争大师之间的仲裁和同步，从而导致一个获胜的大师。
2. 它表示 hs 模式传输的开始。

hs 模式的主码是保留的 8 位码，不用于从属寻址或其他用途。此外，由于每个主代码都有自己独特的主代码，在一个 I2c 总线系统上最多可以出现 8 个 hs 模式的主代码（尽管主代码 0000 1000 应该保留用于测试和诊断目的）。I2C 模块的主代码是软件可编程的。仲裁和时钟同步只发生在主代码传输和不承认位(A)期间，之后一个获胜的主代码将保持活动状态。主代码向其他设备指示 hs 模式传输将开始，并且所连接的设备必须符合 hs 模式规范。由于不允许任何设备承认主代码，所以主代码后面有一个不承认的(A).在不确认位(A)之后，SCL 线被拉到一个高水平，活动主线切换到 HS 模式并启用(在时刻 tH，参见图 27-16 A Complet e HS 模式**使转移**)针对 SCL 信号的电流源上拉电路。由于其他设备可以通过拉伸 SCL 信号的低周期来延迟 tH 之前的串行传输，当所有设备释放 SCL 线且 SCL 信号达到高水平时，活动主设备将启用其电流源上拉电路，从而加速 SCL 信号的最后一部分上升时间。然后，活动主服务器发送一个重复的开始条件（Sr），然后是一个具有 R/W 位地址的 7 位从服务

©Levetop Semiconductor Co., Ltd.

器地址，并从选定的从服务器接收一个确认位(A)。

在重复启动条件之后，在每个确认位(A)或不确认位(A)之后，主动主服务器禁用其电流源上拉电路。这使得其他设备可以通过延长 SCL 信号的低周期来延迟串行传输。当所有设备都释放，SCL 信号达到较高水平时，主动主控器再次重新启用其电流源上拉电路，从而加速 SCL 信号的最后一部分上升时间。

在下一次重复启动（Sr）之后，数据传输继续处于 HS 模式，并且仅在 STOP 条件(P)之后切换回 F/S 模式。为了减少主代码的开销，主代码可以链接多个 hs 模式传输，并由重复的启动条件（Sr）分隔。

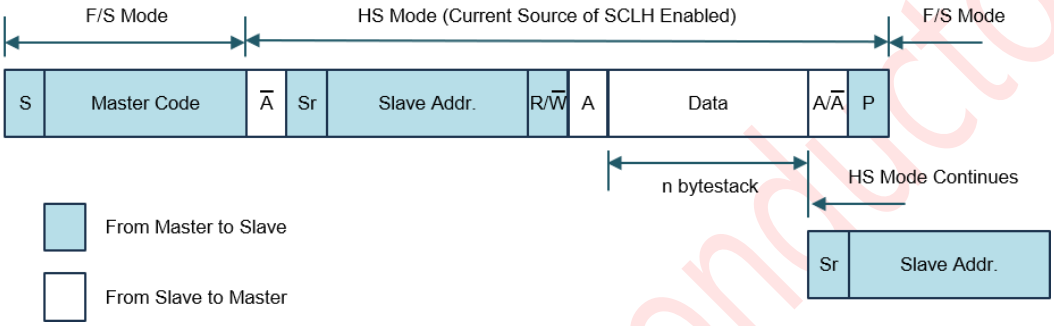


图 27-15：HS 模式下的数据传输格式

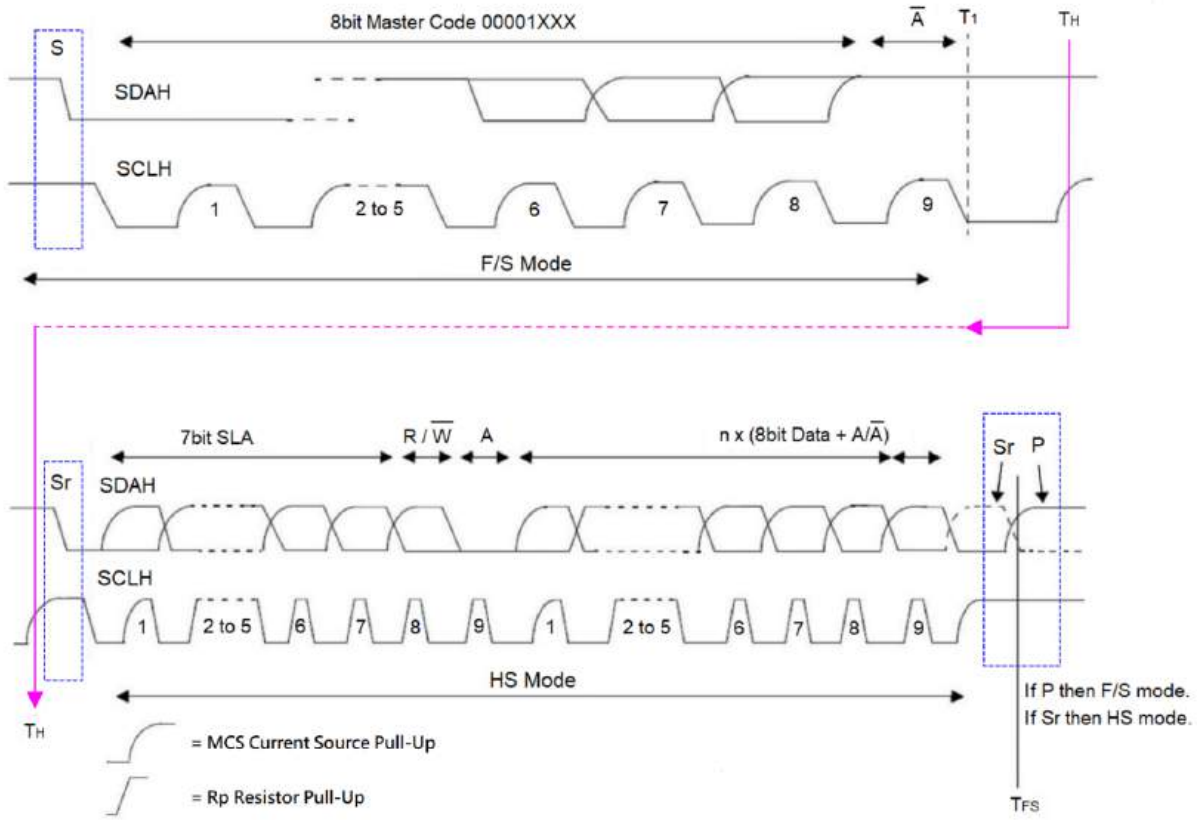


图 27-16：一个完整的 HS 模式传输

©Levetop Semiconductor Co., Ltd.

27.5.9. 软件交易流程图

1. 初始化。

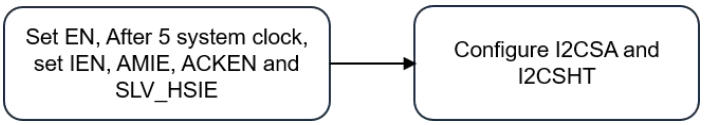


图 27-17：从机模式初始化

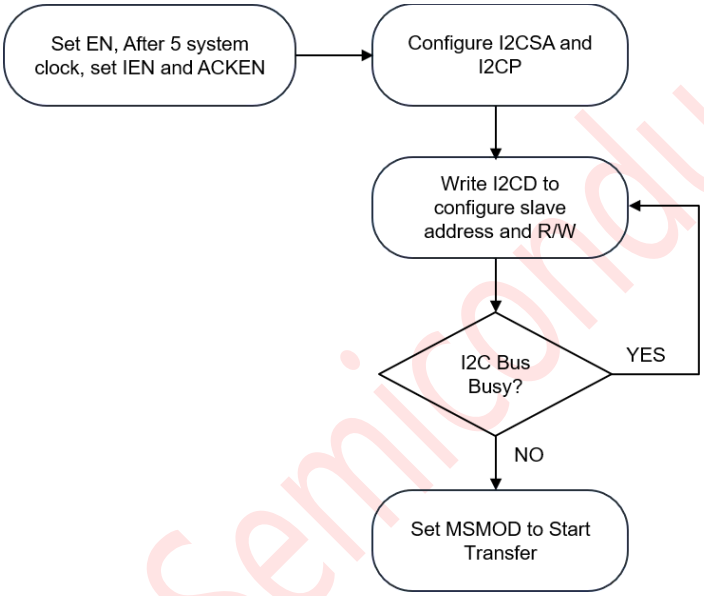


图 27-18：主模式初始化

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

2. 中断事务处理

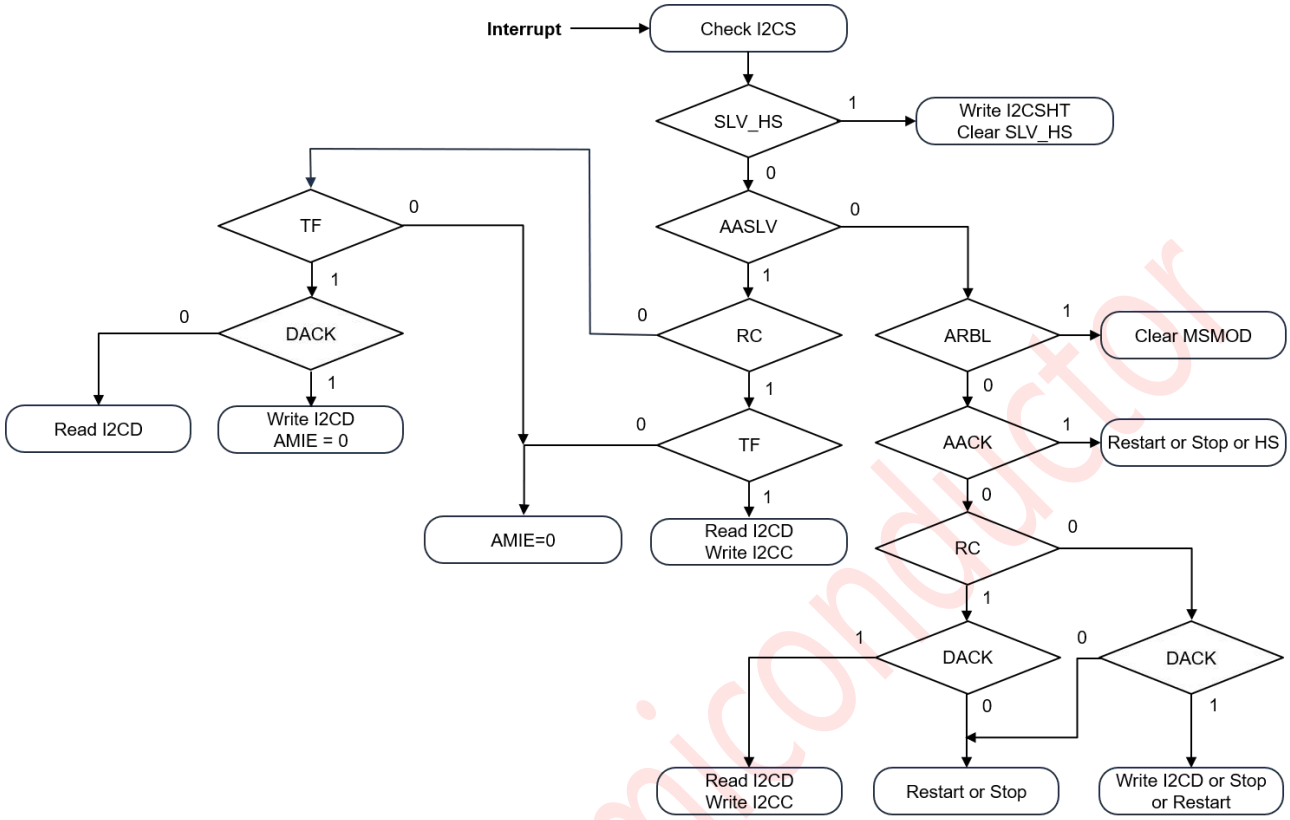


图 27-19：中断事务

如果有来自 I2C 的中断，首先检查 I2CS，以确认 I2C 是处于主模式或从模式。

首先检查 SLV_HS 状态，如果设置了 SLV_HS，然后写入 I2CSHT 以配置高速定时，然后将 1 写入 I2CSHIR 以清除 SLV_HS，否则：

在从模式下，如果设置 RC，表示 I2C 处于从接收模式，然后检查 TF，如果 TF 也设置，表示收到数据（非地址），然后读取 I2CD，写入 I2CC 清除 RC 和 TF 接收下一个字节数据。如果没有设置 TF，则表示只接收了从地址，然后写入 I2CC 清除 RC 和 AMIE，然后接收下一个字节数据。

在从模式下，如果没有设置 RC，则检查 TF，如果 TF 已经设置，然后检查 DACK，如果 DACK 也已经设置，表示 I2C 处于从传输模式，然后写入 I2CD 清除 TF，写入 I2CC 清除 AMIE，传输下一个字节数据。如果未设置 DACK，则读取最后接收字节的 I2CD 以清除 TF。

在从属模式下，如果 TF 和 RC 都没有设置，则写入 I2CC 以清除 AMIE。

在主模式下，如果设置了 ARBL，则表示仲裁丢失，然后写入 I2CC 清除 MSMOD。如果 ARBL 没有设置，然后检查 AACK，如果 AACK 已经设置，这意味着地址确认错误，然后写入 I2CC 重复启动或停止。

©Levetop Semiconductor Co., Ltd.

在主模式下，如果没有设置 ARBL 和 AACK，则检查 RC，如果设置 RC，表示 I2C 处于杆接收模式，然后检查 DACK，如果 DACK 也设置，然后读取 I2CD 清除 RC，写入 I2CC 选择是否确认数据。如果 DACK 尚未设置，请写入 I2CC 以重复启动或停止。

在主模式下，如果没有设置 ARBL 和 AACK 的模式，则检查 RC，如果未设置 RC 的模式，表示 I2C 处于主传输模式，然后检查 DACK，如果设置了 DACK，写入 I2CD 清除 TF，传输下一个字节数据或写入 I2CC 重复启动或停止。如果 DACK 尚未设置，则写入 I2CC 以重复启动或停止。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

28.脉冲宽度调制器（PWM）

28.1. 功能介绍

LT168 嵌入了 4 个 PWM 定时器。4 个 PWM-定时器有 2 个预刻度，2 个时钟分频器，4 个时钟选择器，4 个 16 位计数器，4 个 16 位比较器，和 2 个死区发生器。每个都可以用作计时器和发出中断。

每个两个 PWM-Timers 共享相同的预比例尺。时钟分频器为每个计时器提供 5 个时钟源（1、1/2、1/4、1/8、1/16)。每个计时器中的 16 位计数器接收来自时钟选择器的时钟信号，并可用于处理一个 PWM 周期。16 位比较器将计数器中的数与之前加载的寄存器中的阈值数进行比较，以生成 PWM 占空比。来自时钟分频器的时钟信号被称为 PWM 时钟。死区发生器利用 PWM 时钟作为时钟源。一旦启用了死区发生器，两个 PWM 定时器的输出将被阻塞。两个输出引脚均作为死区发电机的输出信号，用以控制片外电源装置。比较器的值用于脉宽调制。当下计数器的值与比较寄存器的值相匹配时，计数器控制逻辑会改变输出电平。

每个 PWM-Timer 都包含一个捕获通道。捕获 0 和 PWM 0 共享一个包含在 PWM 0 中的计时器；而捕获 1 和 PWM 1 共享另一个计时器，等等。因此，用户必须在打开捕获特性之前设置 PWM-Timer。启用捕获特性后，当输入通道有上升过渡时，捕获总是锁定 PWM 计数器到 CRLR，当输入通道有下降过渡时，捕获总是锁定 PWM 计数器到 CFLR。捕获通道 0 中断可以通过设置 CCR0[1]（上升锁存中断启用）和 CCR0[2]（下降锁存中断启用）来确定中断触发条件。捕获通道 1、2 和 3 也具有相同的功能。每当捕获问题中断 0/1/2/3 时，PWM 计数器 0/1/2/3 将在此时重新加载。最大捕获频率应由中断进程时间来决定。如果中断处理时间为 T0，则捕获信道输入信号在 T0 不变化，最大捕获频率为 1/T0。

从 PWM 到中断控制器（INTC）只有四个中断。PWM 0 和捕获 0 共享相同的中断；PWM1 和捕获 1 共享相同的中断，等等。因此，PWM 功能和捕获功能不能同时使用。

28.2. PWM 特点

脉冲宽度调制器包括以下显著特点：

- 可编程周期
- 可编程占空比
- 两个死区产生器
- 捕获功能
- 引脚可以配置为通用 I/O

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

28.3. 方块图

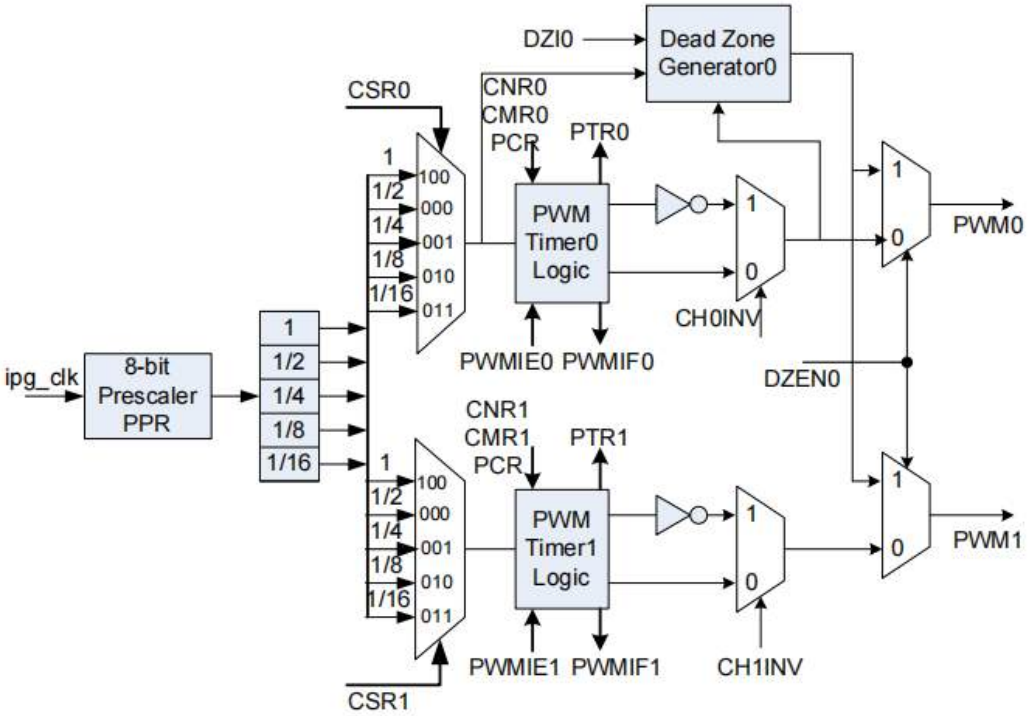


图 28-1：PWM 方块图

28.4. 信号描述

表 28-1：PWM 信号描述

名称	I/O	宽度	复位状态	描述
PWM0	I/O	1	0	PWM0 pin
PWM1	I/O	1	0	PWM1 pin
PWM2	I/O	1	0	PWM2 pin
PWM3	I/O	1	0	PWM3 pin

PWMx 用作通用的输入/输出，也用作 PWM 发送输出或捕获输入。

在默认状态下，它被用作通用的输入端口。

©Levetop Semiconductor Co., Ltd.

28.5. 内存映射和寄存器

PWM 模块内存映射情况见表 28-2.PWM0 的碱基地址为 0x400D_0000，PWM1 为 0x400E_0000。本节描述了 PWM 的内存映射和寄存器结构。

28.5.1. 内存映射

表 28-2：模块内存映射

地址	Bit[31:0]	访问权限(1)
0x0000	PWM Pre-scale Register (PPR)	S/U
0x0004	PWM Clock Select Register (PCSR)	S/U
0x0008	PWM Control Register (PCR)	S/U
0x000C	PWM Counter Register0 (PCNR0)	S/U
0x0010	PWM Comparator Register0 (PCMR0)	S/U
0x0014	PWM Timer Register0 (PTR0)	S/U
0x0018	PWM Counter Register1 (PCNR1)	S/U
0x001C	PWM Comparator Register1 (PCMR1)	S/U
0x0020	PWM Timer Register1 (PTR1)	S/U
0x0024	PWM Counter Register2 (PCNR2)	S/U
0x0028	PWM Comparator Register2 (PCMR2)	S/U
0x002C	PWM Timer Register2 (PTR2)	S/U
0x0030	PWM Counter Register3 (PCNR3)	S/U
0x0034	PWM Comparator Register3 (PCMR3)	S/U
0x0038	PWM Timer Register3 (PTR3)	S/U
0x003C	PWM Interrupt Enable Register (PIER)	S/U
0x0040	PWM Interrupt Flag Register (PIFR)	S/U
0x0044	PWM Capture Control Register0 (PCCR0)	S/U
0x0048	PWM Capture Control Register1 (PCCR1)	S/U
0x004C	PWM Capture Rising Latch Register0 (PCRLR0)	S/U
0x0050	PWM Capture Falling Latch Register0 (PCFLR0)	S/U
0x0054	PWM Capture Rising Latch Register1 (PCRLR1)	S/U
0x0058	PWM Capture Falling Latch Register1 (PCFLR1)	S/U
0x005C	PWM Capture Rising Latch Register2 (PCRLR2)	S/U
0x0060	PWM Capture Falling Latch Register2 (PCFLR2)	S/U
0x0064	PWM Capture Rising Latch Register3 (PCRLR3)	S/U
0x0068	PWM Capture Falling Latch Register3 (PCFLR3)	S/U
0x006C	PWM Port Control Register (PPCR)	S/U

注意(1)：S/U = CPU 主管或用户模式访问。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

28.5.2. 寄存器描述

28.5.2.1. PWM 预缩放寄存器（PPR）

寄存器（PPR）用于设置预调节器和设置死区长度。

地址：PWMn_BASEADDR+0x0000_0000

	31	30	29	28	27	26	25	24
读: 写:	DZI1[7:0]							
复位:	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
读: 写:	DZI0[7:0]							
复位:	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
读: 写:	CP1[7:0]							
复位:	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
读: 写:	CP0[7:0]							
复位:	0	0	0	0	0	0	0	0

图 28-2：PWM 比例前寄存器（PPR）

DZI1[7:0] — Dead zone interval register 1 for PWM2 and PWM3

这 8 位决定了死区的长度。从时钟选择器 1 接收到死区长度的 1 单位时间。

DZI0[7:0] — Dead zone interval register 0 for PWM0 and PWM1

这 8 位决定了死区的长度。从时钟选择器 0 接收到死区长度的 1 个单位时间。

CP1[7:0] — Clock pre-scale 1 for PWM Timer 2 & 3

时钟输入除以（CP1 + 1），然后输入到 PWM 计时器 2 和 3 的计数器。

CP0[7:0] — Clock pre-scale 0 for PWM Timer 0 & 1

时钟输入除以（CP0 + 1），然后输入到 PWM 计时器 0 和 1 的计数器。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

28.5.2.2. PWM 时钟选择寄存器（PCSR）

时钟分频器为每个计时器提供 5 个时钟源（1、1/2、1/4、1/8、1/16）。每个计时器从时钟分频器接收它自己的时钟信号，该分频器接收来自 8 位预缩放的时钟。

地址：PWMn_BASEADDR+0x0000_0004

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0	CSR3[2:0]			0	CSR2[2:0]		
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0	CSR1[2:0]			0	CSR0[2:0]		
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-3：PWM 时钟选择寄存器（PCSR）

CSR3[2:0] — Timer 3 Clock Source Selection

选择计时器 3 选择时钟输入。

表 28-3：计时器 3 时钟源选择

CSR3[2:0]	输入时钟除以
100~111	1
011	16
010	8
001	4
000	2

CSR2[2:0] — Timer 2 Clock Source Selection

选择计时器 2 选择时钟输入。与 CSR3 相同。

©Levetop Semiconductor Co., Ltd.

CSR1[2:0] — Timer 1 Clock Source Selection

为计时器 1 选择时钟输入。与 CSR3 相同。

CSR0[2:0] — Timer 0 Clock Source Selection

选择计时器 0 选择时钟输入。与 CSR3 相同。

28.5.2.3. PWM 对照寄存器（PCR）

这个寄存器是 PWM 控制寄存器。

地址：PWMn_BASEADDR+0x0000_0008

	31	30	29	28	27	26	25	24
读:	0				CH3MOD	CH3INV	0	CH3EN
写:								
复位:	0	0	0	0	0	0	0	0

	23	19	18	20	16	18	17	16
读:	0				CH2MOD	CH2INV	0	CH2EN
写:								
复位:	0	0	0	0	0	0	0	0

	15	11	10	12	8	10	9	8
读:	0				CH1MOD	CH1INV	0	CH1EN
写:								
复位:	0	0	0	0	0	0	0	0

	7	3	2	4	0	2	1	0
读:	0		DZEN1	DZEN0	CH0MOD	CH0EN	0	CH0EN
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-4：PWM 控制寄存器（PCR）

CH3MOD — Timer 3 Auto-load/One-Shot Mode

1 = 自动加载模式

0 = 单镜头模式

提示：如果在该位有一个上升或下降的过渡，它将导致 CNR3 和 CMR3 被清除。

CH3INV — Timer 3 Inverter ON/OFF

1 = 反向器打开

0 = 反向器关闭

©Levetop Semiconductor Co., Ltd.

CH3EN — Timer 3 Enable/Disable

1 = 启用

0 = 禁用

CH2MOD — Timer 2 Auto-load/One-Shot Mode

1 = 自动加载模式

0 = 单镜头模式

提示：如果在该位有上升或下降的过渡，则 CNR2 和 CMR2 将被清除。

CH2INV — Timer 2 Inverter ON/OFF

1 = 反向器打开

0 = 反向器关闭

CH2EN — Timer 2 Enable/Disable

1 = 启用

0 = 禁用

CH1MOD — Timer 1 Auto-load/One-Shot Mode

1 = 自动加载模式

0 = 单镜头模式

提示：如果在该位有上升或下降的过渡，则 CNR1 和 CMR1 将被清除。

CH1INV — Timer 1 Inverter ON/OFF

1 = 反向器打开

0 = 反向器关闭

CH1EN — Timer 1 Enable/Disable

1 = 启用

0 = 禁用

DZEN1 — Dead-Zone 1 Generator Enable/Disable

1 = 启用

0 = 禁用

提示：当启用 DZEN1 时，应禁用 CH3EN。因为通道 3 和通道 2 的输出都由通道 2 决定的。

DZEN0 — Dead-Zone 0 Generator Enable/Disable

1 = 启用

0 = 禁用

提示：当启用 DZEN0 时，应禁用 CH1EN。因为通道 1 和通道 0 的输出都由通道 0 决定的。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

CH0MOD — Timer 0 Auto-load/One-Shot Mode

1 = 自动加载模式

0 = 单镜头模式

提示：如果在此位有一个上升或下降的过渡，它将导致清除 CNR0 和 CMR0。

CH0INV — Timer 0 Inverter ON/OFF

1 = 反向器打开

0 = 反向器关闭

CH0EN — Timer 0 Enable/Disable

1 = 启用

0 = 禁用

28.5.2.4. PWM 计数器寄存器（PCNRn）

这些寄存器通过定义该周期内的计数脉冲的数量来控制 PWM 的周期。

地址：PWMn_BASEADDR+0x0000_000C

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CNR0[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	CNR0[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

地址：PWMn_BASEADDR+0x0000_0018

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CNR1[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	CNR1[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

地址：PWMn_BASEADDR+0x0000_0024

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CNR2[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	CNR2[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

地址：PWMn_BASEADDR+0x0000_0030

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CNR3[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	CNR3[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-5：PWM 计数器寄存器（PCNR）

CNRx[15:0] — Loaded Value for PWM Counter/Timer

数据范围：65535~0（单位：1 PWM 时钟周期）

提示：

1: 一个 PWM 周期宽度 = CNR + 1。如果 CNR 等于零，则 PWM 计数器/计时器将被停止。

2: 每当一个值被写入 CNR 时，它将在下一个 PWM 计数器周期中生效。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

28.5.2.5. PWM 比较器寄存器（PCMRn）

这些寄存器定义了脉冲的宽度。当计数器与此寄存器中的值匹配时，输出将在此期间内复位。

地址：PWMn_BASEADDR+0x0000_0010

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CMR0[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	CMR0[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

地址：PWMn_BASEADDR+0x0000_001C

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CMR1[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	CMR1[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

地址：PWMn_BASEADDR+0x0000_0028

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CMR2[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	CMR2[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

地址：PWMn_BASEADDR+0x0000_0034

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CMR3[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	CMR3[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-6：PWM 比较器寄存器（PCMR）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

CMRx[15:0] — PWM Comparator Register

数据范围：65535~0（单位：1 PWM 时钟周期）

CMR 用于确定 PWM 输出占空比。

假设：PWM 输出初始值：高

CMR >= CNR ： PWM 的输出量总是很高的

CMR < CNR ： PWM 输出高 =（CMR + 1）单元

CMR = 0 ： PWM 输出高 = 1 单元

提示：

1: PWM tuty = CMR+1。如果 CMR 等于零，则 PWM 占空比为 1

2: 每当一个值被写入 CMR 时，它将在下一个 PWM 计数器周期中生效。

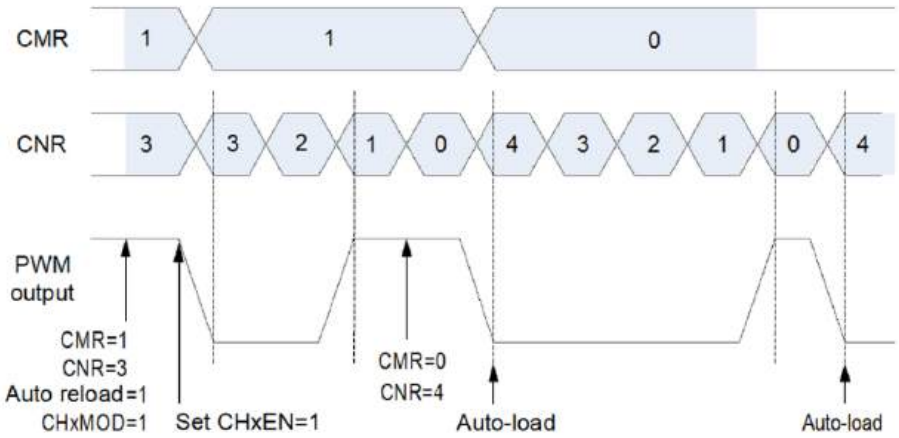


图 28-7：PWM 输出

调制 PWM 控制器输出占空比（CNR = 150）

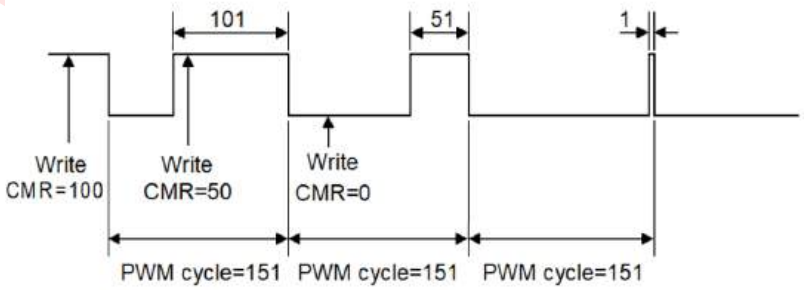


图 28-8：PWM 占空比

©Levetop Semiconductor Co., Ltd.

28.5.2.6. PWM 计时器寄存器（PTRn）

只读的 PWM 计时器寄存器保存当前的计数值。它可以随时阅读而不干扰柜台。

地址：PWMn_BASEADDR+0x0000_0014

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	PTR0[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	PTR0[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

地址：PWMn_BASEADDR+0x0000_0020

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	PTR1[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	4	3	5	4	3	2	1	0
读:	PTR1[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

地址：PWMn_BASEADDR+0x0000_002C

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	PTR2[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	PTR2[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

地址：PWMn_BASEADDR+0x0000_0038

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	PTR3[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	PTR3[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-9：PWM 计时器寄存器（PTR）

PTRx[15:0] — PWM Timer value

只读的 PTR 位保存当前的计数值。用户可以监控 PTR，以获得 16 位下计数器中的当前值。

©Levetop Semiconductor Co., Ltd.

28.5.2.7. PWM 中断启用寄存器（PIER）

这个寄存器用于启用 PWM 计时器中断。

地址：PWMn_BASEADDR+0x0000_003C

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0				PIER3	PIER2	PIER1	PIER0
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-10：PWM 中断启用寄存器（PIER）

PIER3 — PWM Timer 3 Interrupt Enable

1 = 启用

0 = 禁用

PIER2 — PWM Timer 2 Interrupt Enable

1 = 启用

0 = 禁用

PIER1 — PWM Timer 1 Interrupt Enable

1 = 启用

0 = 禁用

PIER0 — PWM Timer 0 Interrupt Enable

1 = 启用

0 = 禁用

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

28.5.2.8. PWM 中断标志寄存器（PIFR）

这个寄存器用于指示 PWM 计时器中断标志。

地址：PWMn_BASEADDR+0x0000_0040

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0				PIFR3	PIFR2	PIFR1	PIFR0
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-11：PWM 中断标志寄存器（PIFR）

PIFR3 — PWM Timer 3 Interrupt Flag.

当 PWM 计时器 3 计数为 0，而 PIER3 = 1 时，PIFR3 将被设置为 1。向此位写入 1 将清除 PIFR3。

1 = 中断标志已打开

0 = 中断标志已关闭

PIFR2 — PWM Timer 2 Interrupt Flag.

当 PWM 计时器 2 计数为 0，而 PIER2 = 1 时，PIFR2 将被设置为 1。向此位写入 1 将清除 PIFR2。

1 = 中断标志已打开

0 = 中断标志已关闭

PIFR1 — PWM Timer 1 Interrupt Flag.

当 PWM 计时器 1 计数为 0，并且 PIER1 = 1 时，PIFR1 将被设置为 1。写入 1 将清除 PIFR1。

1 = 中断标志已打开

0 = 中断标志已关闭

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

PIFR0 — PWM Timer 0 Interrupt Flag.

当 PWM 计时器 0 计数为 0，并且 PIER0 = 1 时，PIFR0 将被设置为 1。写入 1 将清除 PIFR0。

1 = 中断标志已打开

0 = 中断标志已关闭

28.5.2.9. PWM 捕获控制寄存器（PCCR0/1）

这些寄存器用于控制捕获功能。

地址：PWMn_BASEADDR+0x0000_0044

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	CFLRD1	CRLRD1	0	CAPIF1	CAPCH1EN	FL_IE1	RL_IE1	INV1
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	CFLRD0	CRLRD0	0	CAPIF0	CAPCH0EN	FL_IE0	RL_IE0	INV0
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-12：PWM 捕获控制寄存器（PCCR0）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

地址：PWMn_BASEADDR+0x0000_0048

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	CFLRD3	CRLRD3	0	CAPIF3	CAPCH3 EN	FL_IE3	RL_IE3	INV3
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	CFLRD2	CRLRD2	0	CAPIF2	CAPCH2 EN	FL_IE2	RL_IE2	INV2
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-13：PWM 捕获控制寄存器（PCCR1）

CFLRDx — Capture Falling Latch Register load flag

1 = 当输入通道 x 有一个下降的过渡时，CFLRx 被更新，这个位是“1”。

0 = 当输入通道 x 没有一个下降的过渡时。

写下 1 来清除这一位。

CRLRDx — Capture Rising Latch Register load flag

1 = 当输入通道 x 有一个上升的过渡时，CRLRx 被更新，这个位是“1”。

0 = 当输入通道 x 没有一个上升的过渡时。

写下 1 来清除这一位。

CAPIFx — Capture Channel x interrupt flag

1 = 当输入通道 x 有一个下降的过渡，并且启用了 FL_IEx 位时，将设置这个中断标志。当输入通道 x 有一个上升的过渡，并且启用了 RL_IEx 位时，也将设置这个中断标志。

0 = 未设置捕获通道 x 中断标志。

写下 1 来清除这一位。

CAPCHxEN — Capture Channel x Enable/Disable

1 = 启用

0 = 禁用

©Levetop Semiconductor Co., Ltd.

当此位设置为 1 时，捕获锁定 PMW 计数器，该值保存到 CRLR（上升锁存）和 CFLR（下降锁存）。
当此位设置为 0 时，捕获不会更新 CRLR 和 CFLR，并禁用通道 x 中断。

FL_IEx — Channel x Falling Interrupt Enable ON/OFF

1 = 启用

0 = 禁用

当这个位被设置为 1 时，如果捕获检测到通道 x 有一个下降的过渡，捕获将发出一个中断。

RL_IEx — Channel x Rising Interrupt Enable ON/OFF

1 = 启用

0 = 禁用

当此位设置为 1 时，如果捕获检测到通道 x 有一个上升的过渡，捕获将发出一个中断。

INVx — Channel x Inverter ON/OFF

1 = 反向器打开

0 = 反向器关闭

28.5.2.10. PWM 捕获上升锁存寄存器（PCRLRn）

这些寄存器用于在捕获上升过渡时锁定 PWM 计数器。

地址：PWMn_BASEADDR+0x0000_004C

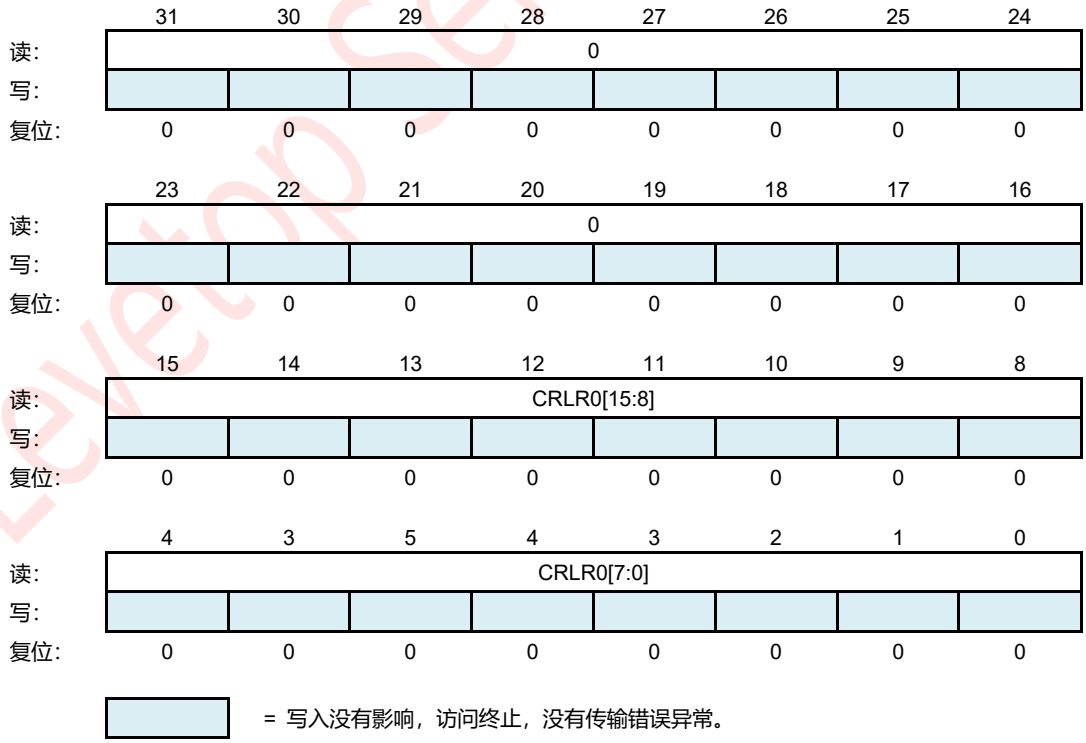


图 28-14：PWM 捕获上升锁存寄存器（PCRLR0）

©Levetop Semiconductor Co., Ltd.

地址：PWMn_BASEADDR+0x0000_0054

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CRLR1[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	CRLR1[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-15：PWM 捕获上升锁存寄存器（PCRLR1）

地址：PWMn_BASEADDR+0x0000_005C

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CRLR2[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	CRLR2[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-16：PWM 捕获上升锁存寄存器（PCRLR2）

©Levetop Semiconductor Co., Ltd.

地址：PWMn_BASEADDR+0x0000_0064

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CRLR3[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	CRLR3[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-17：PWM 捕获上升锁存寄存器（PCRLR3）

CRLRx[15:0] — Capture Rising Latch Registerx

当通道 x 的过渡阶段出现上升时，锁住 PWM 计数器。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

28.5.2.11. PWM 捕获坠落锁存器寄存器（PCFLRn）

这些寄存器用于在捕获一个下降的转换时锁存 PWM 计数器。

地址：PWMn_BASEADDR+0x0000_0050

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CFLR0[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	4	3	5	4	3	2	1	0
读:	CFLR0[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-18：PWM 捕获坠落锁存寄存器（PCFLR0）

地址：PWMn_BASEADDR+0x0000_0058

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CFLR1[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	CFLR1[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-19：PWM 捕获坠落锁存寄存器（PCFLR13）

©Levetop Semiconductor Co., Ltd.

地址：PWMn_BASEADDR+0x0000_0060

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CFLR2[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	CFLR2[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-20：PWM 捕获坠落锁存寄存器（PCFLR2）

地址：PWMn_BASEADDR+0x0000_0068

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	CFLR3[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	CFLR3[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 28-21：PWM 捕获坠落锁存寄存器（PCFLR3）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

CFLRx[15:0] — Capture Falling Latch Registerx

当通道 x 的过渡过程下降时，锁住 PWM 计数器。

28.5.2.12. PWM 端口控制寄存器（PPCR）

寄存器（PPCR）用于控制 PWMx 引脚方向和引脚状态。

地址：PWMn_BASEADDR+0x0000_006C

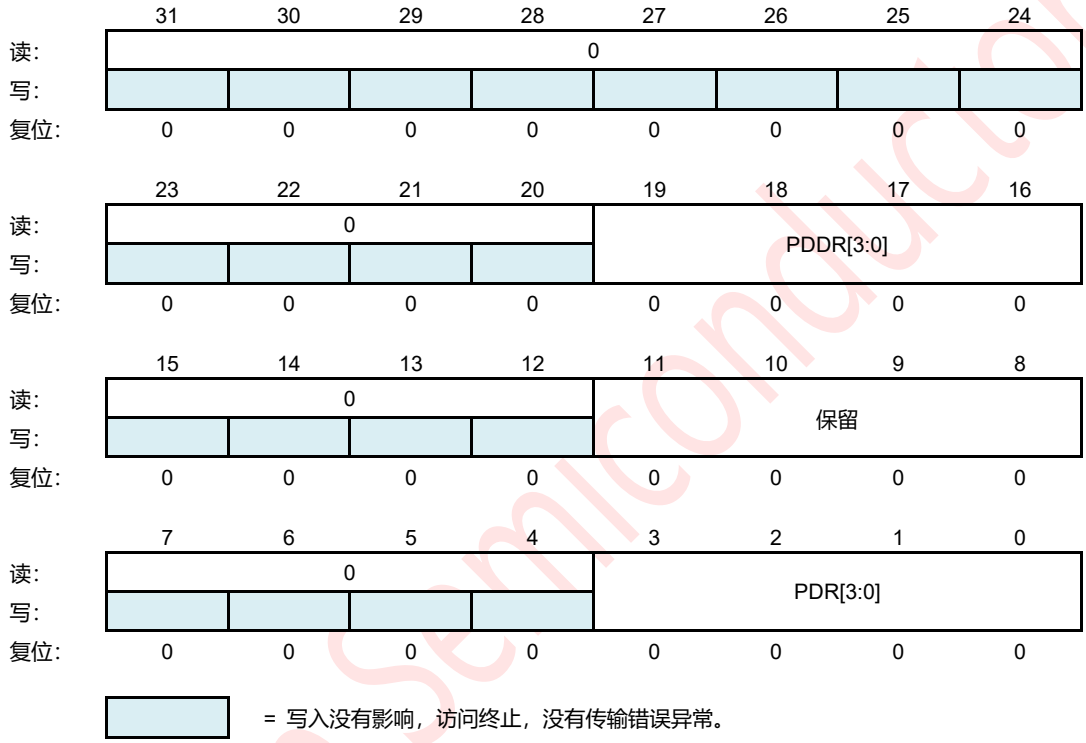


图 28-22：PWM 端口控制寄存器（PPCR）

PDDR[3:0] — Port Data Direction Register

PDDR[3:0]位控制 PWM 引脚的方向。复位清除 PDDR[3:0]。

1 = 对应的引脚配置为输出

0 = 对应的引脚配置为输入

PDR[3:0] — Port Data Register

将数据写入 PDR[3:0]可以驱动配置为通用输出的相应引脚。读取输入(PDDR 位清除）将返回引脚电平。

©Levetop Semiconductor Co., Ltd.

28.6. 功能描述

本节介绍了 PWM 的功能操作。

28.6.1. PWM 双缓冲和自动重新加载

PWM-计时器具有双重缓冲功能，允许为下一个计时器操作更改的重新加载值，而不停止当前计时器操作。虽然设置了新的计时器值，但当前计时器操作仍然成功操作。计数器值可以写入 CNR0~3 中，当前的计数器值可以从 PTR0~3 中读取。当下计数器达到 0 时，自动重新加载操作将将数据从 CNR0~3 复制到下计数器。如果 CNR0~3 设置为零，则当计数器计数为零时，计数器将停止。如果自动重装位设置为零，计数器将立即停止。

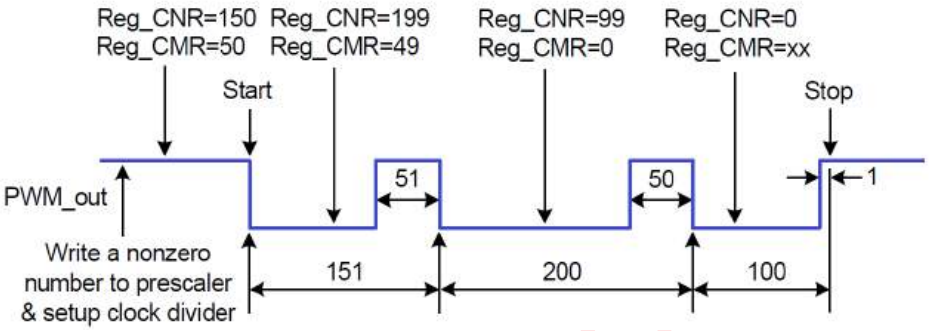


图 28-23：PWM 双缓冲功能说明

28.6.2. 调节占空比

双缓冲功能允许在当前周期中的任何时间点上写入 CMR。所加载的值将从下一个周期开始生效。

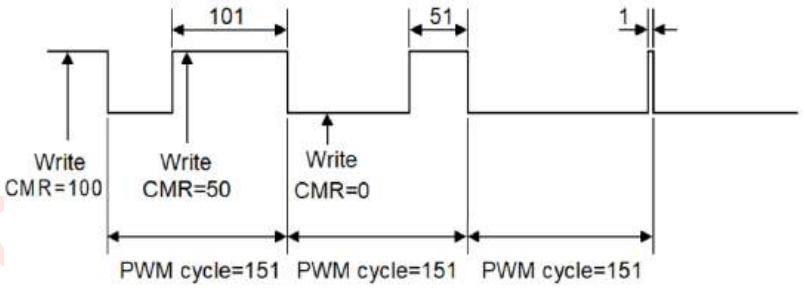


图 28-24：PWM 控制器输出负荷比

©Levetop Semiconductor Co., Ltd.

28.6.3. 死区发电机

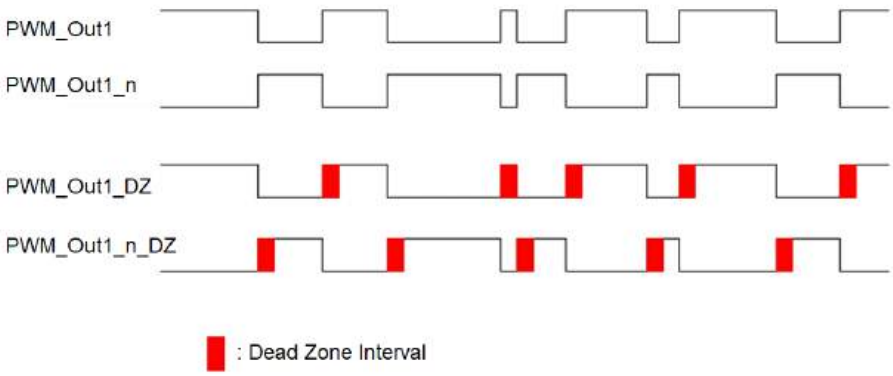


图 28-25：死区生成操作

28.6.4. PWM 计时器启动过程

1. 设置时钟选择器（CSR）
2. 设置比例前 r 和死区间隔（PPR）
3. 设置反向器开/关，死区发生器开/关，切换模式/一次性模式，和 PWM 定时器关闭。（PCR）
4. 设置比较器寄存器（CMR）
5. 设置计数器寄存器（CNR）
6. 设置中断启用寄存器（PIER）
7. 设置 PWMx 作为输出引脚（PPCR）
8. 启用 PWM 计时器（PCR）

28.6.5. PWM 计时器停止程序

方法 1：
将 16 位下计数器（CNR）设置为 0，并监控 PTR。当 PTR 达到 0 时，禁用 PWM 计时器（PCR）。（推荐使用）

方法 2：
将 16 位下计数器（CNR）设置为 0。当中断请求发生时，禁用 PWM 计时器（PCR）。（推荐使用）

方法 3：
直接禁用 PWM 定时器（PCR）。（不推荐使用）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

28.6.6. 捕获启动过程

1. 设置时钟选择器（CSR）
2. 设置预比例尺（PPR）
3. 设置反向器开/关，死区发生器开/关，自动加载模式/一次性模式，和 PWM 定时器关闭。（PCR）
4. 设置计数器寄存器（CNR）
5. 设置捕获寄存器（CCR）
6. 设置 PWMx 作为输入引脚（PPCR）
7. 启用 PWM 计时器（PCR）

28.6.7. 捕获基本计时器操作

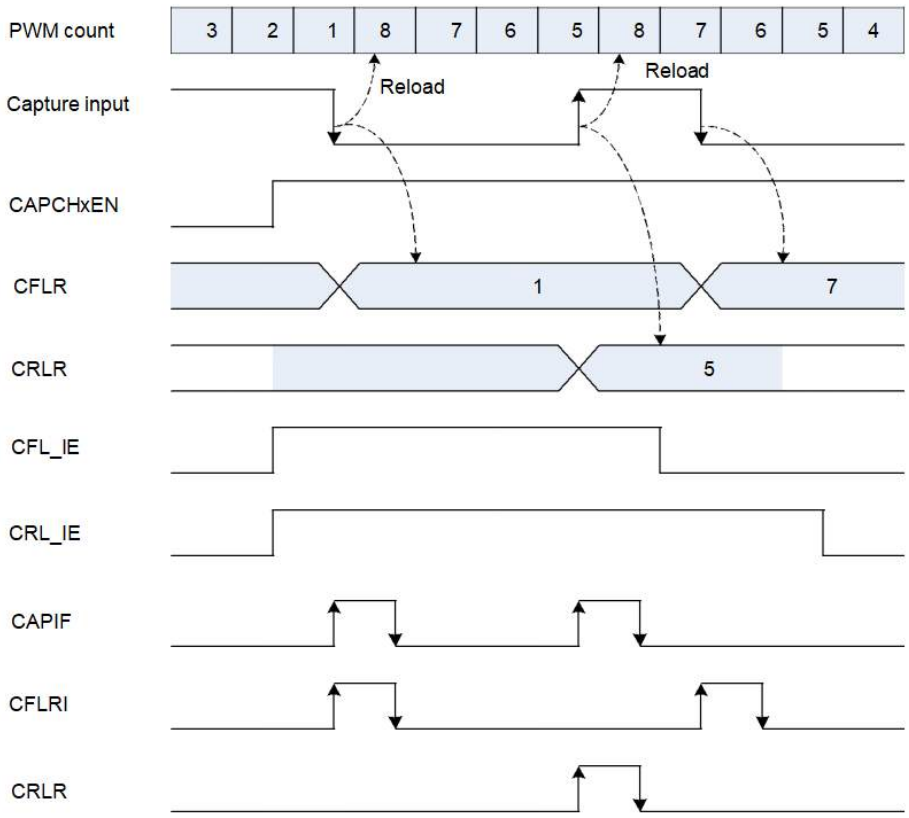


图 28-26：捕获基本计时器操作

在这种情况下，中国北车是 8 号：

1. 当 CAPIFx 设置为 1 时，PWM 计数器 CNRx 将被重新加载。
2. 通道低脉冲宽度为（CNR+1-CRLR）。
3. 通道高脉冲宽度为（CNR +1 - CFLR）。

©Levetop Semiconductor Co., Ltd.

29.模拟比较器（COMP）

29.1. 功能介绍

模拟比较器提供可编程的响应时间，迟滞，一个模拟输入多路复用器，和两个输出，在端口引脚可选：一个同步“过滤”输出（CP），或一个异步“原始”输出（CPA）。即使在系统时钟处于非活动状态，异步 CPA 信号也可用。这允许比较器在停止模式下操作并生成输出。

29.2. 方块图

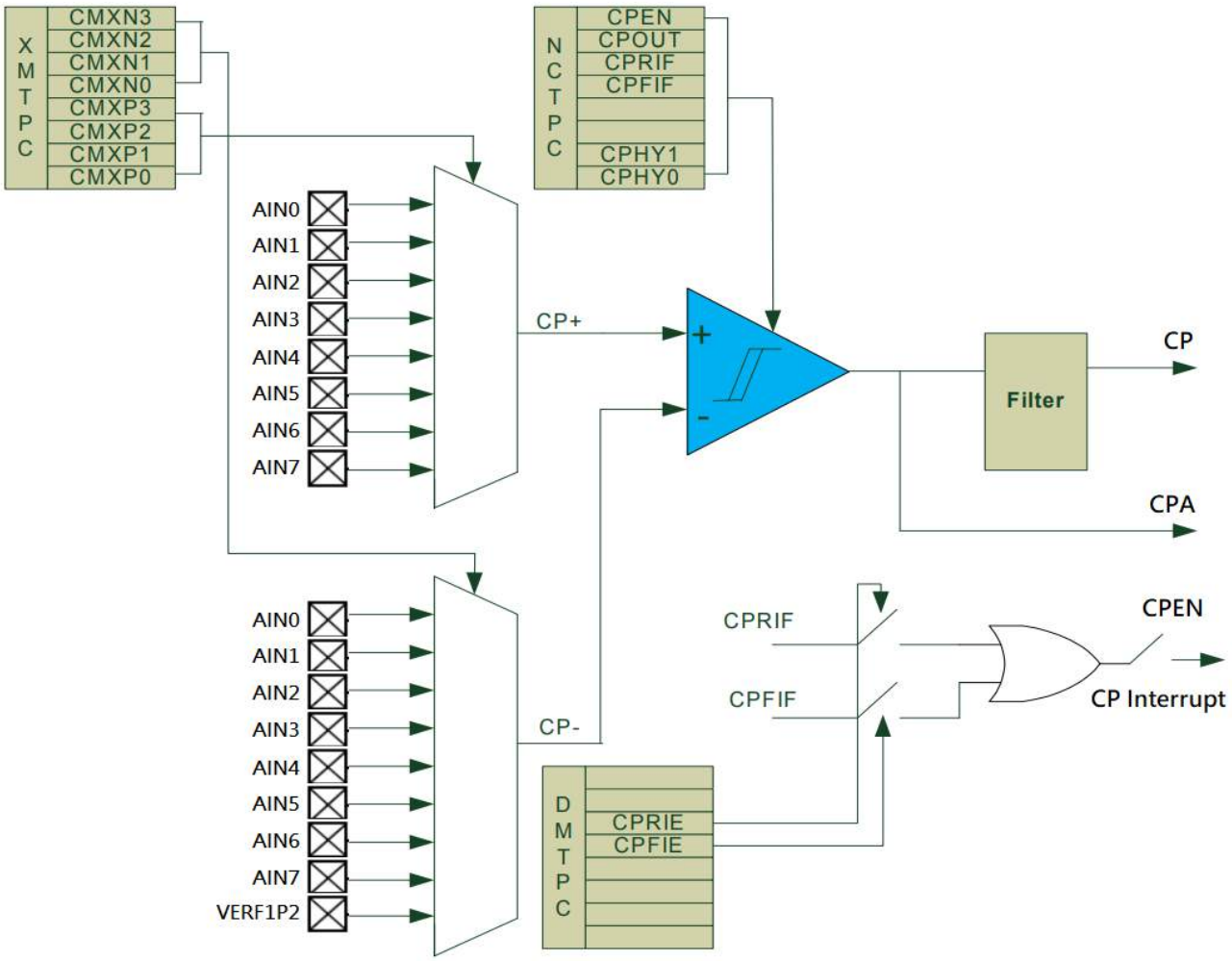


图 29-1：比较器方块图

©Levetop Semiconductor Co., Ltd.

29.3. 操作模式

本节介绍了这三种低功耗模式。

29.3.1. 等待模式

在等待模式下，比较器模块可以通过设置 CPEN 位来继续正常运行，并且可以通过配置为通过产生中断请求来退出低功耗模式。

29.3.2. Doze 模式

在起启动模式下，比较器模块可以通过设置 CPEN 位来继续正常运行，并且可以被配置为通过产生中断请求来退出低功耗模式。

29.3.3. 停止模式

在停止模式下，系统时钟不存在，比较器模块可以通过设置 CPEN 位来继续正常运行，并且可以通过产生一个异步的“原始”输出（CPA）唤醒信号来配置为退出低功耗模式。

29.4. 内存映射和寄存器

比较器模块记忆映射如表 29-1 所示.COMP0 的基本地址为 0x400A_0000，COMP1 为 0x400B_0000。本节介绍了比较器的内存映射和寄存器结构。

29.4.1. 内存映射

见表 29-1 以获取内存映射的描述。
该设备设有两个比较器模块。

表 29-1：比较器模块内存贴图

偏移地址	Bits[7:0]	访问权限(1)
0x3	Comparator Control Register(CPTCN)	S/U
0x2	Comparator0 Mode Selection Register(CPTMD)	S/U
0x1	Comparator MUX Selection Register(CPTMX)	S/U
0x0	Comparator Output Filter Selection Register(CPTFS)	S/U

提示（1）： 仅限 S = CPU 监控模式访问。S/U = CPU 监控器或用户模式访问。在用户模式下只访问管理员地址没有影响，并导致周期终止传输错误。

©Levetop Semiconductor Co., Ltd.

29.4.2. 寄存器描述

比较器编程模型由以下寄存器组成：

•CPTCN：比较器控制寄存器。

•CPTMD：比较器模式选择寄存器。

•CPTMX：比较器 MUX 选择寄存器。

•CPTFLS：比较器输出过滤器选择寄存器

29.4.2.1. 比较器控制寄存器（CPTCN）

地址：COMP0_BASEADDR+0x0000_0003，COMP1_BASEADDR+0x0000_0003

	7	6	5	4	3	2	1	0
读:	CPEN	CPOUT	CPRIF	CPFIF	0	0	CPHY[1:0]	
写:								
复位:	0	0	0	0	0	0	0	1

= 写入没有影响，访问终止，没有传输错误异常。

图 29-2：比较器控制寄存器（CPTCN）

CPEN — Comparator Enable Bit

读/写 CPEN 位启用比较器操作。当禁用比较器时，CPOUT 处于低状态。

1 = 已启用比较器

0 = 已禁用比较器

CPOUT — Comparator Output State Flag.

1 = 电压在 CP+ > CP-

0 = 电压在 CP+ < CP-。

CPRIF — Comparator Rising-Edge Flag. Must be cleared by writing one to this bit.

1 = 比较器上升边缘产生。

0 = 自上次清除此标志以来，未发生比较器上升边缘。

CPFIF — Comparator Falling-Edge Flag. Must be cleared by writing one to this bit.

1 = 比较器下降边缘产生。

0 = 自上次清除此标志以来，未发生比较器掉边。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

CPHY[1:0] — Comparator Hysteresis Control Bits.

00 = 滞后已被禁用。

01 = 迟滞 = 8 mV。

10 = 迟滞性 = 12 mV。

11 = 迟滞 = 15.4 mV。

29.4.2.2. 比较器模式选择寄存器（CPTMD）

地址：COMP0_BASEADDR+0x0000_0002，COMP1_BASEADDR+0x0000_0002

	7	6	5	4	3	2	1	0
读:	CPOEN	0	CPRIE	CPFIE	0	WKUPLEVEL	0	CPMD
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 29-3：比较器模式选择寄存器（CPTMD）

CPOEN — Comparator Output to pad control bit.

1 = 比较器输出可以在 PWM1[0]的 COMP0 中观察到，在 COMP1 的 PWM1[1]中观察到。

0 = 比较器输出禁用。

CPRIE — Comparator Rising-Edge Interrupt Enable.

1 = 比较器上升缘中断允许。

0 = 比较器上升缘中断禁用。

CPFIE — Comparator Falling-Edge Interrupt Enable.

1 = 比较器下降缘中断允许。

0 = 比较器下降缘中断禁用。

WKUPLEVEL — Comparator WakeUp level control bit.

1 = CP+ > CP– 上的电压将在停止模式期间产生唤醒请求。

0 = CP+ < CP– 上的电压将在停止模式期间产生唤醒请求。

CPMD — Comparator Mode Select. These bits select the response time for Comparator.

表 29-2：比较器模式选择

模式	CPMD 响应时间的功耗		
低速	0	500ns	3.3uA
高速	1	80ns	22uA

©Levetop Semiconductor Co., Ltd.

29.4.2.3. 比较器 MUX 选择寄存器（CPTMX）

地址：COMP0_BASEADDR+0x0000_0001，COMP1_BASEADDR+0x0000_0001

	7	6	5	4	3	2	1	0
读: 写:	CMXN[3:0]				CMXP[3:0]			
复位:	1	1	1	1	1	1	1	1

图 29-4：比较器 MUX 选择寄存器（CPTMX）

CMXN[3:0] — Comparator Negative Input MUX Select. These bits select which Port pin is used as the Comparator negative input.

表 29-3：比较器负向输入 MUX 选择

CMXN3	CMXN2	CMXN1	CMXN0	负输入
0	0	0	0	AIN0
0	0	0	1	AIN 1
0	0	1	0	AIN 2
0	0	1	1	AIN 3
0	1	0	0	AIN 4
0	1	0	1	AIN 5
0	1	1	0	AIN 6
0	1	1	1	AIN 7
1	0	0	0	Vref1p2
其他值				None

CMXP[3:0] — Comparator Positive Input MUX Select. These bits select which Port pin is used as the Comparator positive input.

表 29-4：比较器正输入 MUX 选择

CMXP3	CMXP2	CMXP1	CMXP0	正输入
0	0	0	0	AIN0
0	0	0	1	AIN 1
0	0	1	0	AIN 2
0	0	1	1	AIN 3
0	1	0	0	AIN 4
0	1	0	1	AIN 5
0	1	1	0	AIN 6
0	1	1	1	AIN 7
1	x	x	x	None

©Levetop Semiconductor Co., Ltd.

29.4.2.4. 比较器输出过滤器选择寄存器（CPTFLS）

地址：COMP0_BASEADDR+0x0000_0000，COMP1_BASEADDR+0x0000_0000

	7	6	5	4	3	2	1	0
读: 写:	CMFLEN	CMFL_WIDTH[6:0]						
复位:	0	0	0	0	0	0	0	0

图 29-5：比较器输出过滤器选择寄存器（CPTFLS）

CMFLEN — Comparator Output Digital Filter Enable

1 = 比较器输出数字滤波器，滤波器输出产生 CPMRIF 和 CPMFIF；

0 = 比较器输出数字滤波器，CPMRIF 和 CPMFIF 由原始输出生成；

CMFL_WIDTH[6:0] — Comparator Output Digital Filter Pulse Width Selection.

CMFL_WIDTH[6:0]确定将被过滤的输入脉冲的宽度。如果输入脉冲宽度小于（CMFL_WIDTH[6:0]+2）*固定周期，则脉冲将被过滤。

29.5. 功能描述

在 CPTMX 寄存器中选择比较器输入。CMXP3-CMXP0 位选择比较器正输入；CMXN3-CMXN0 位选择比较器负输入。关于比较器输入的重要提示：选择为比较器输入的端口引脚应在其相关的端口配置寄存器中配置为模拟输入，并且应禁用输入、输出、上拉和下拉控制。

比较器的输出可以在软件中进行轮询，用作中断源，和/或路由到端口引脚。当路由到端口引脚时，比较器输出可用，并且与系统时钟具有异步性或同步性；异步输出即使在停止模式（无活动系统时钟）下也可用。禁用时，比较器输出默认为逻辑低状态，其电源电流下降小于 100nA。

比较器滞后是软件可编程通过其比较器控制寄存器 CPTCN。用户可以编程的迟滞电压的值。比较器迟滞现象使用比较器控制寄存器 CPTCN 中的位 1-0 进行编程。

比较器中断可以在上升边缘和下降边缘的输出转换上产生。在比较器下降边出现时，CPFIF 标志被设置为逻辑 1，而在比较器上升边出现时，CPRIF 标志被设置为逻辑 1。一旦设置，这些位将保持设置，直到被软件清除。通过将 CPRIE 设置为逻辑 1 来启用比较器上升边缘中断掩码。通过将 CPFIE 设置为逻辑 1 来启用比较器下降边缘中断掩码。

通过读取 CPOUT 位，可以随时获得比较器的输出状态。比较器通过设置 CPEN 位为逻辑 1 而被启用，并通过清除该位为逻辑 0 而被禁用。

请注意，当比较器第一次通电时，或如果对磁滞控制位进行了更改时，可以检测到错误的上升边和下降边。因此，建议在比较器启用或改变其模式位后的短时间内，将上升边和下降边标志明确清除为逻辑 0。比较器的最小启动时间小于 5 我们。

©Levetop Semiconductor Co., Ltd.

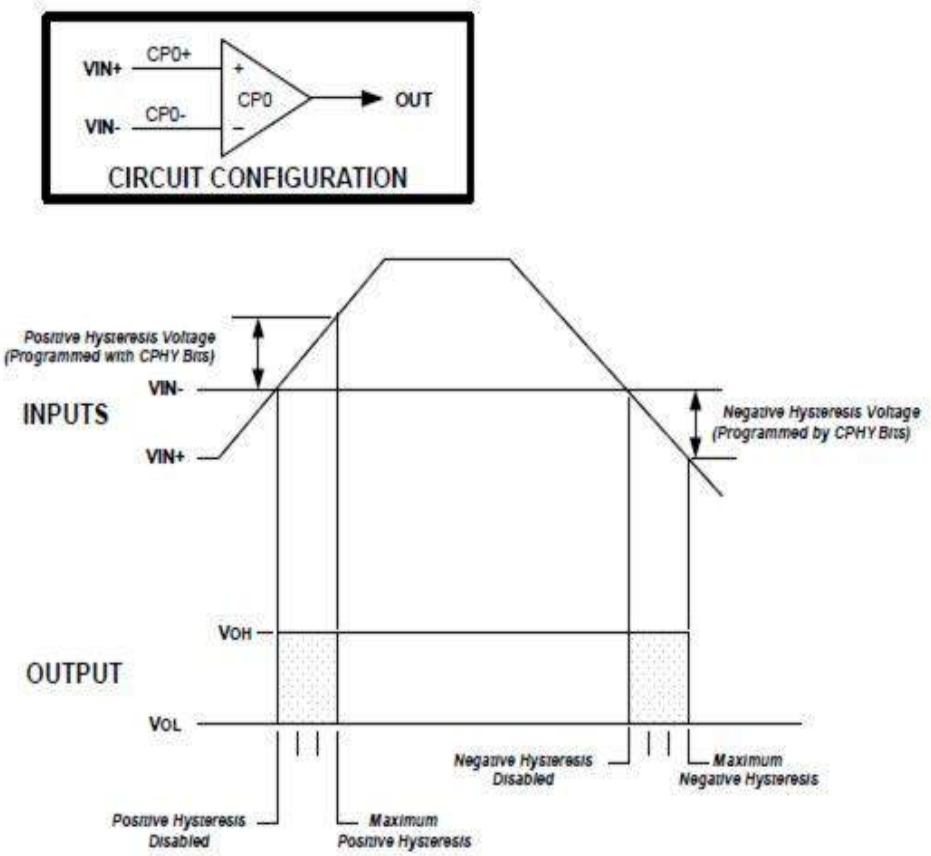


图 29-6：比较器滞后图

©Levetop Semiconductor Co., Ltd.

30.USB2.0 控制器（USBC）

30.1. 功能介绍

本节介绍 USB2.0 全速控制器。本模块中的设备实现提供了实现 USB 2.0 全速/低速兼容外围设备的解决方案。

30.2. USB 模块特点

USB 模块的功能包括：

- USB2.0 仅设备功能控制器
- USB 2.0 兼容
  - 12Mbps 全速（FS）数据速率
  - USB 数据控制逻辑：
    分组识别和解码/生成
    CRC 生成和检查
    NRZI（非返回零倒置）编码/解码
    小填料
    同步检测
    数据包端检测
- 8 个 USB 端点
- USB 内存
  - 2048 字节的缓冲区 RAM，由系统和 USB 模块共享
  - RAM 可以被分配为 USB 控制器或额外的系统 RAM 资源的缓冲区
- USB 复位选项
  - 由单片机生成的 USB 模块复位
  - 由主机生成的总线复位，从而触发 CPU 中断
- 使用远程唤醒支持来暂停和恢复操作
- 收发器功能
  - 将 USB 差分电压转换为数字逻辑信号水平
  - 芯片上的 USB 拔拉电阻器
- 3.3V 调节器

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

30.3. 方块图

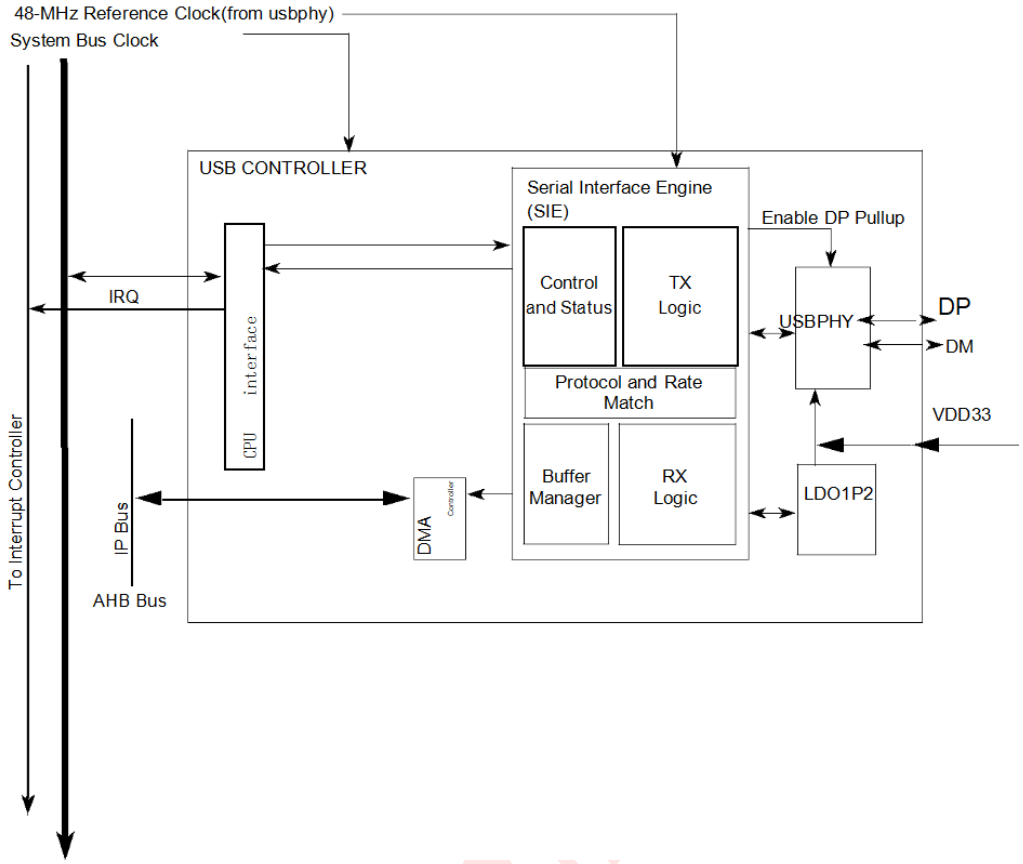


图 30-1：USB 全速设备(USB 模块）方块图

30.4. 操作模式

本节介绍了这三种低功耗模式。

30.4.1. 等待模式

在等待模式下，USB 模块可以继续正常运行，并可以通过产生一个中断请求来配置为退出低功耗模式。

30.4.2. Doze 模式

在 Doze 模式下，USB 模块可以继续正常运行，并且可以通过产生一个中断请求来配置为退出低功耗模式。

30.4.3. 停止模式

USB 模块可选择在停止模式下使用。根据 USB 规范版本，USB 暂停模式可能需要减少当前消耗模式。2.0，停止模式可以降低 MCU 的电流消耗，从而降低整个 USB 设备。在通过固件进入停止之前，用户必须确保设备设置配置为停止，以实现 USB 暂停当前的消耗目标。

当设置休眠标志时，通知 USB 模块进入暂停模式；这发生在 USB 总线空闲 3 ms 之后。USB 设备等待处理模式当前消耗级别要求由 USB 规范版本定义。2.0（低功耗 500uA，大功率 2.5 mA）。

©Levetop Semiconductor Co., Ltd.

如果设置了用户，并且在 USB 总线上检测到 K 状态（恢复信令），则恢复位将被设置。这将触发一个异步中断，将唤醒单片机从停止模式，并启用时钟到 USB 模块。

30.5. 内存映射和寄存器

USB 模块内存映射如表 30-1 所示.USB 控制器的基本地址为 0x4016_0000。本节介绍了 USB 模块的内存映射和寄存器结构。

30.5.1. 内存映射

见表 30-1 以获取内存映射的描述。

表 30-1：USB 模块内存地图

偏移地址	Bits[31:0]	访问权限(1)
0x0000	保留	S/U
0x0004	保留	S/U
0x0008	保留	S/U
0x000C	保留	S/U
0x001C	USBPHY Control Register 1 (USBPHY_CTRL1)	S/U
0x0080	Interrupt Status Register (INT_STAT)	S/U
0x0084	Interrupt Enable Register (INT_ENB)	S/U
0x0088	Error Interrupt Status Register (ERR_STAT)	S/U
0x008C	Error Interrupt Enable Register (ERR_ENB)	S/U
0x0090	Status Register (STAT)	S/U
0x0094	Control Register (CTL)	S/U
0x0098	Address Register (ADDR)	S/U
0x009C	EBT Page Register 1 (EBT_PAGE_01)	S/U
0x00A0	Frame Number Register (FRMNUML)	S/U
0x00A4	Frame Number Register (FRMNUMH)	S/U
0x00A8	Token Register (TOKEN)	S/U
0x00AC	SOF Threshold Register (SOF_THLD)	S/U
0x00B0	EBT Page Register 2 (EBT_PAGE_02)	S/U
0x00B4	EBT Page Register 3 (EBT_PAGE_03)	S/U
0x00C0	Endpoint Control Registers (ENDPT0)	S/U
0x00C4	Endpoint Control Registers (ENDPT1)	S/U
0x00C8	Endpoint Control Registers (ENDPT2)	S/U
0x00CC	Endpoint Control Registers (ENDPT3)	S/U
0x00D0	Endpoint Control Registers (ENDPT4)	S/U
0x00D4	Endpoint Control Registers (ENDPT5)	S/U
0x00D8	Endpoint Control Registers (ENDPT6)	S/U
0x00DC	Endpoint Control Registers (ENDPT7)	S/U

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

偏移地址	Bits[31:0]	访问权限(1)
0x0100	USBPHY Control Register 2 (USBPHY_CTRL2)	S/U
0x0104	USB PHY Observe Register (USB_PHY_OBSERVE)	S/U
0x0108	USB PHY GPIO Register	S/U
0x010C	USB Resume Wakeup Enable Register (USB_RESMEN)	S/U
0x0118	USBPHY Control Register 3 (USBPHY_CTRL3)	S/U
0x011C	USBPHY Control Register 4 (USBPHY_CTRL4)	S/U

注意(1)：仅限 S = CPU 监控模式访问。S/U = CPU 监控器或用户模式访问。在用户模式下只访问管理员地址没有影响，并导致周期终止传输错误。

30.5.2. 寄存器描述

30.5.2.1. USBPHY 控制寄存器 1（USBPHY_CTRL1）

USBPHY 控制寄存器控制数据线终端电阻的操作。

地址：USBC_BASEADDR+0x0000_001C

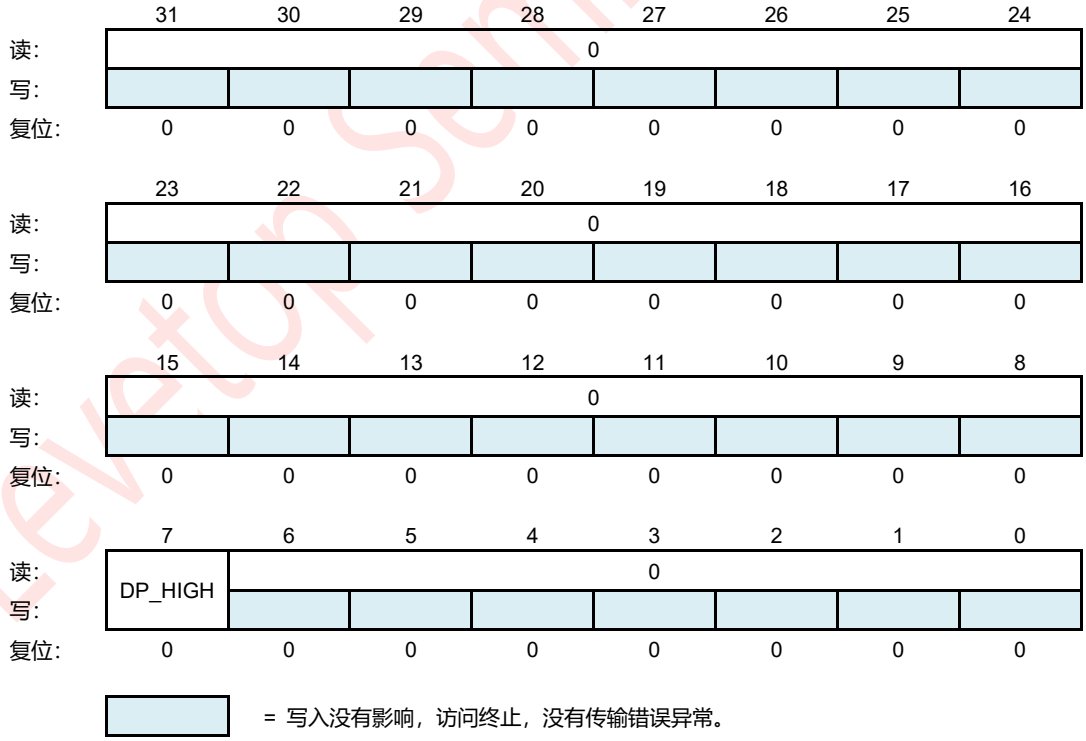


图 30-2：USBPHY 控制寄存器 1（USBPHY_CTRL1）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

DP_HIGH — D+ Data Line pullup resistor enable

0 = D+上拉电阻被禁用

1 = 启用 D+上拉电阻。

30.5.2.2. 中断状态寄存器（INT_STAT）

中断状态寄存器包含了 USB 模块中的每个中断源的位。这些位都用各自的中断启用位限定。这个寄存器的所有位在逻辑上与 OTG 中断状态寄存器（OTG_STAT）一起使用，为处理器的中断控制器形成一个单一的中断源。在设置了中断位之后，只能通过将中断位写入相应的中断位来清除。此寄存器包含复位后的值 0x00。

地址：USBC_BASEADDR+0x0000_0080

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	STALL	0	RESUME	SLEEP	TOKDNE	SOF_TOK	ERROR	USB_RST
写:								
复位:	0	0	0	1	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 30-3：中断状态寄存器（INT_STAT）

STALL — Stall Interrupt, and cleared by writing 1 to this bit.

在设备模式下，当 USB 模块发送停止握手时，将断言此位。在主机模式下，当 USB 模块在 USB 事务的握手阶段检测到停止确认时，将设置此位。此中断可用于确定最后一个 USB 事务是否已成功完成或是否停止。

RESUME — Resume Interrupt, and cleared by writing 1 to this bit.

此位根据 DP/DM 信号设置，可用于信号在 USB 总线上的远程唤醒信号。当未处于等待处理模式时，应禁用此中断。

SLEEP — Sleep Interrupt, and cleared by writing 1 to this bit.

当 USB 模块检测到 USB 总线上持续空闲 3 毫秒时，将设置此位。睡眠计时器通过 USB 总线上的活动进行复位。

©Levetop Semiconductor Co., Ltd.

TOKDNE — Token Done Interrupt, and cleared by writing 1 to this bit.

在正在处理的当前令牌完成时设置此位。处理器应立即读取 STAT 寄存器，以确定用于此令牌的端点和 EB 条目。清除此位（通过写入一个位）会导致清除 STAT 寄存器，或将 STAT 保持寄存器加载到 STAT 寄存器中。

SOF_TOK — SOF Token Interrupt, and cleared by writing 1 to this bit.

当 USB 模块接收到帧的开始（SOF）令牌时，将设置此位。在主机模式下，当达到 SOF 阈值时设置这个位，以便软件可以为下一个 SOF 做准备。

ERROR — Error Interrupt, and cleared by writing 1 to this bit.

当 ERR_STAT 寄存器中的任何错误条件发生时，都将设置此位。然后，处理器必须读取 ERR_STAT 寄存器，以确定错误的来源。

USB_RST — USB RST Interrupt, and cleared by writing 1 to this bit.

当 USB 模块解码有效 USB 复位时设置此位。这将通知微处理器，它应该将 0x00 写入地址寄存器，并启用端点 0。在检测到 USB 复位 2.5 微秒后，将设置 USB_RST。在删除 USB 复位条件并重新插入之前，不会再次断言。

30.5.2.3. 中断启用寄存器（INT_ENB）

中断启用寄存器包含了 USB 模块中的每个中断源的启用位。设置这些位中的任何一个都可以启用 INT_STAT 寄存器中各自的中断源。此寄存器包含复位后的值 0x00。

地址：USBC_BASEADDR+0x0000_0084

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	STALL_EN	ATTACH_EN	RESUME_EN	SLEEP_EN	TOKDNE_EN	SOF_TOK_EN	ERROR_EN	USB_RST_EN
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 30-4：中断启用寄存器（INT_ENB）

©Levetop Semiconductor Co., Ltd.

STALL_EN — Stall Interrupt Enable

0 = 静态中断已被禁用

1 = 停止中断。

ATTACH_EN — Attach Interrupt Enable.

0 = 附加中断已被禁用。

1 = 已启用附加中断。

RESUME_EN — Resume Interrupt Enable.

0 = 恢复中断已被禁用。

1 = 恢复中断。

SLEEP_EN — Sleep Interrupt Enable.

0 = 睡眠中断。

1 = 休眠中断。

TOKDNE_EN — Token Done Interrupt Enable.

0 = 令牌完成中断被禁用。

1 = 启用令牌完成中断。

SOF_TOK_EN — SOF Toke Interrupt Enable.

0 = SOF 令牌中断已被禁用。

1 = 已启用 SOF 令牌中断。

ERROR_EN — Error Interrupt Enable.

0 = 错误中断已被禁用。

1 = 已启用错误中断。

USB_RST_EN — USB Reset Interrupt Enable.

0 = USB 复位中断禁用。

1 = USB 复位中断启用。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

30.5.2.4. 错误中断状态寄存器（ERR_STAT）

错误中断状态寄存器包含对 USB 模块中的每个错误源的启用位。这些位都用它们各自的错误启用位进行限定。这个寄存器的所有位在逻辑上在一起，结果放置在 INT_STAT 寄存器的错误位中。在设置了中断位之后，只能通过将中断位写入相应的中断位来清除。一旦检测到错误条件，就会设置每个位。因此，该中断通常并不对应于正在处理的令牌的结束。此寄存器包含复位后的值 0x00。

地址：USBC_BASEADDR+0x0000_0088

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	BTS_ERR	0	DMA_ERR	BTO_ERR	DFN8	CRC16	CRC5_EOF	PID_ERR
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 30-5：错误中断状态寄存器（ERR_STAT）

BTS_ERR — This bit is set when a bit stuff error is detected. If set, the corresponding packet is rejected due to the error.

DMA_ERR — 如果 USB 模块请求 DMA 访问以读取新的 EBT，但在需要接收或传输数据之前没有得到总线，则设置此位。如果处理 TX 传输，这将导致传输数据下流状态。如果正在处理 RX 传输，这将导致接收数据溢出的情况。当为微处理器和 USB 模块开发设备仲裁硬件以最小化总线请求和总线授予延迟时，此中断非常有用。如果到主机或来自主机的数据包大于 EBT 中分配的缓冲区大小，也会设置此位。在这种情况下，数据包在放入缓冲区内存中时被截断。

BTO_ERR — 在发生总线周转超时错误时设置此位。USB 模块包含一个总线周转计时器，它跟踪令牌与设置或输出令牌的数据阶段或 IN 令牌的数据和握手阶段之间使用的时间量。如果在从 IDLE 转换之前从前一个 EOP 计算了超过 16 位的时间，则会发生总线周转超时错误。

DFN8 — 如果接收到的数据字段的长度不是 8 位，则设置此位。USB 规范 1.0 要求数据字段是一个整数字节数。如果数据字段不是整数字节数，则设置此位。

©Levetop Semiconductor Co., Ltd.

CRC16 — 当数据包由于 CRC16 错误而被拒绝时，设置此位。

CRC5_EOF — 这个错误中断有两个功能。当 USB 模块在设备模式（HOST_MODE_EN = 0）下运行时，此中断会检测到主机生成的令牌数据包中的 CRC5 错误。如果设置，则由于 CRC5 错误，令牌包被拒绝。当 USB 模块在主机模式（HOST_MODE_EN = 1）下运行时，此中断会检测到帧结束（EOF）错误情况。当 USB 模块正在传输或接收数据，而 SOF 计数器为零时，就会发生这种情况。在开发 USB 数据包调度软件时，这个中断很有用，以确保没有 USB 事务跨越下一帧的开始。

PID_ERR — 当 PID 检查字段失败时，将设置此位。

30.5.2.5. 错误中断启用寄存器（ERR_ENB）

错误中断启用寄存器包含对 USB 模块中的每个错误中断源的启用位。设置这些位中的任何一个都可以启用 ERR_STAT 寄存器中各自的中断源。一旦检测到错误条件，就会设置每个位。因此，该中断通常并不对应于正在处理的令牌的结束。此寄存器包含复位后的值 0x00。

地址：USBC_BASEADDR+0x0000_008C

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	BTS_ERR_EN	0	DMA_ERR_EN	BTO_ERR_EN	DFN8_EN	CRC16_EN	CRC5_EOF_EN	PID_ERR_EN
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 30-6：错误中断启用寄存器（ERR_ENB）

BTS_ERR_EN — BTS_ERR Interrupt Enable

0 = BTS_ERR 中断已被禁用。

1 = BTS_ERR 中断已启用。

©Levetop Semiconductor Co., Ltd.

DMA_ERR_EN — DMA_ERR Interrupt Enable.

0 = DMA_ERR 中断已被禁用。

1 = DMA_ERR 中断已启用。

BTO_ERR_EN — BTO_ERR Interrupt Enable.

0 = BTO_ERR 中断已被禁用。

1 = BTO_ERR 中断已启用。

DFN8_EN — DFN8 Interrupt Enable.

0 = DFN8 中断已被禁用。

1 = DFN8 中断。

CRC16_EN — CRC16 Interrupt Enable.

0 = CRC16 中断已被禁用。

1 = CRC16 中断。

CRC5_EOF_EN — CRC5_EOF Interrupt Enable.

0 = CRC5_EOF 中断已被禁用。

1 = CRC5_EOF 中断已启用。

PID_ERR_EN — PID_ERR Interrupt Enable.

0 = PID_ERR 中断已被禁用。

1 = PID_ERR 中断已启用。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

30.5.2.6. 状态寄存器（STAT）

状态寄存器报告 USB 模块内的事务状态。当处理器的中断控制器接收到 TOK_DNE 中断时，应读取状态寄存器以确定先前端点通信的状态。当断言 TOK_DNE 中断位时，状态寄存器中的数据将有效。STAT 寄存器实际上是由 USB 模块维护的状态 FIFO 中的一个读取窗口。当 USB 模块使用 EB 条目时，它会更新状态寄存器。如果在 TOK_DNE 中断之前执行了另一个 USB 事务，USB 模块将在 STATF 中存储下一个事务的 FIFO 的状态。因此，STAT 寄存器实际上是一个四字节的 FIFO，它允许处理器核心在处理 SIE 处理下一个事务时处理一个事务。清除 INT_STAT 寄存器中的 TOK_DNE 位会导致 SIE 用下一个 STAT 值的内容更新 STAT 寄存器。如果 STAT 保持寄存器中的数据有效，则 SIE 立即重新插入 TOK_DNE 中断。

地址：USBC_BASEADDR+0x0000_0090

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	ENDP[3:0]				TX	ODD	0	0
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 30-7：状态寄存器（STAT）

ENDP[3:0] — Endpoint Number.

这四个位对接收或传输前一个令牌的端点地址进行编码。这允许微控制器确定最后一个 EBT 事务更新了哪个 EBT 条目。

0000 : Endpoint 0
0001 : Endpoint 1
0010 : Endpoint 2
0011 : Endpoint 3
0100 : Endpoint 4
0101 : Endpoint 5
0110 : Endpoint 6
0111 : Endpoint 7

©Levetop Semiconductor Co., Ltd.

TX — Transmit Indicator.

0 = 最近的交易是一个接收操作。

1 = 最近的一笔交易是一个传输操作。

ODD — Odd/Even Transaction.

如果更新的最后一个端点缓冲区表在 EBT 的奇数组中，则设置此位。

30.5.2.7. 控制寄存器（CTL）

控制寄存器为 USB 模块提供各种控制和配置信息。

地址：USBC_BASEADDR+0x0000_0094

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	JSTATE	SE0	TXSUSPEND/TOKENBUSY	RESET	HOST_MODE_EN	RESUME	ODD_RST	USB_EN/SOF_EN
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 30-8：控制寄存器（CTL）

JSTATE — Live USB differential receiver JSTATE signal

该信号的极性受到 LS_EN 的当前状态的影响。

SE0 — Live USB Single Ended Zero signal

该信号的极性受到 LS_EN 的当前状态的影响。

©Levetop Semiconductor Co., Ltd.

TXSUSPEND / TOKENBUSY

当 USB 模块处于主机模式时，当 USB 模块忙于执行 USB 令牌时，将设置 TOKEN_BUSY，并且不再将令牌命令写入令牌寄存器。软件应该在将任何令牌写入令牌寄存器之前检查此位，以确保令牌命令不会丢失。在设备模式下，当 SIE 禁用了数据包传输和接收时，将设置 tx 挂起。清除此位将允许 SIE 继续进行令牌处理。此位在接收到安装令牌时由 SIE 设置，允许软件在恢复令牌处理之前删除 EBT 中任何挂起的数据包事务。

RESET — This bit is invalid for current device-role-only function.

设置此位可使 USB 模块能够生成 USB 复位信号。这允许 USB 模块复位 USB 外设。此控制信号仅在主机模式下有效（HOST_MODE_EN = 1）。软件必须在所需的时间内设置为 1，然后将其清除为 0 以结束复位信号。

HOST_MODE_EN — This bit is read-only for this device-role-only function.

当设置为 1 时，此位使 USB 模块能够在主机模式下运行。在主机模式下，USB 模块在主机处理器的编程控制下执行 USB 事务。

RESUME

当设置为 1 时，此位使 USB 模块能够执行恢复信令。

这允许 USB 模块执行远程唤醒。软件必须将所需时间设置为 1，然后将其清除为 0。如果设置了 HOST_MODE_EN 位，当恢复位被清除时，USB 模块附加到恢复信令。

ODD_RST

将此位设置为 1，将将所有 EBT ODD 乒乓球/乒乓球位复位为 0，然后指定 EVEN EBT 库。

USB_EN/SOF_EN — USB Enable.

设置此位会导致 SIE 将其所有的 ODD 位复位为 ebt。因此，设置这个位会复位 SIE 中的大部分逻辑。当启用主机模式时，清除此位将导致 SIE 停止发送 SOF 令牌。

0 = USB 模块已禁用。

1 = USB 模块已启用。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

30.5.2.8. 地址寄存器（ADDR）

地址寄存器保存 USB 模块在外设模式下解码的唯一 USB 地址（HOST_MODE_EN = 0）。当在主机模式（HOST_MODE_EN = 1）下操作时，USB 模块使用令牌数据包传输此地址。这使得 USB 模块能够唯一地处理一个 USB 外设。在任何一种模式下，都必须设置控制寄存器内的 USB_EN 位。在复置输入激活或 USB 模块解码 USB 复置信号后，地址寄存器被复位为 0x00。此操作会初始化地址寄存器，以根据 USB 规范的要求解码地址 0x00。

地址：USBC_BASEADDR+0x0000_0098

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	LS_EN	ADDR[6:0]						
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 30-9：地址寄存器（ADDR）

LS_EN — Low Speed Enable bit.

此位通知 USB 模块，写入令牌寄存器的下一个令牌命令必须以低速执行。这使得 USB 模块能够执行低速数据传输所需的必要的前导码。

ADDR[6:0] — USB address.

这个 7 位值定义了 USB 模块在外设模式下解码或在主机模式时传输的 USB 地址。

©Levetop Semiconductor Co., Ltd.

30.5.2.9. EBT 页面寄存器 1（EBT_PAGE_01）

地址：USBC_BASEADDR+0x0000_009C

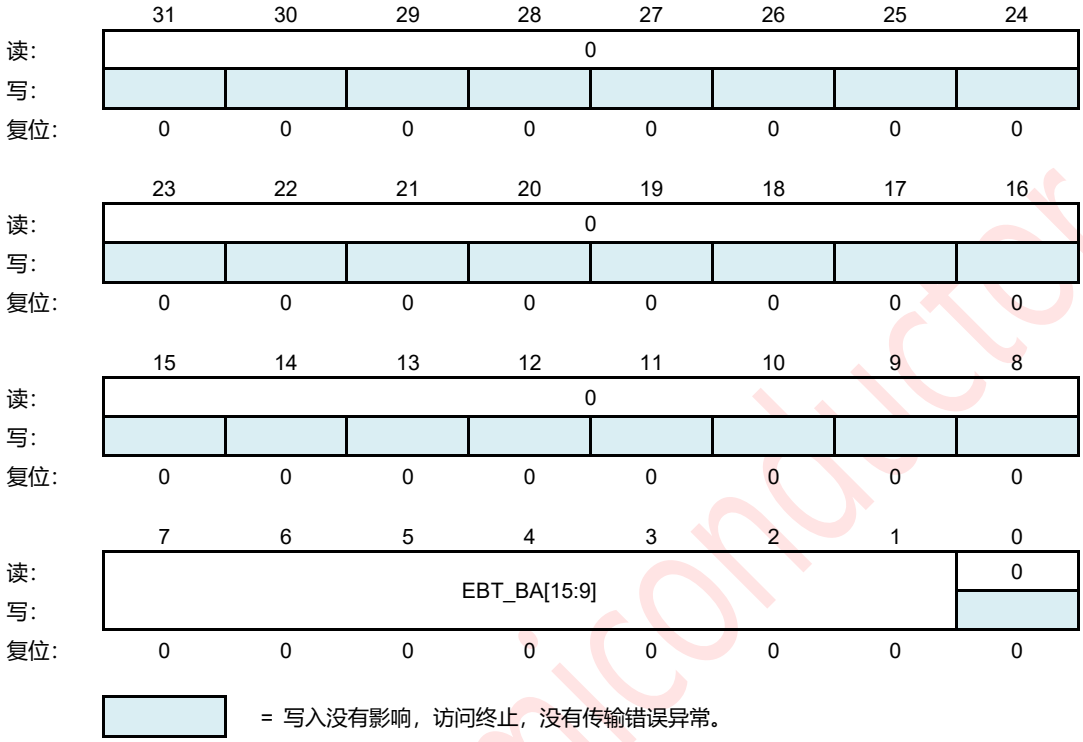


图 30-10：EBT 页面寄存器 1（EBT_PAGE_01）

EBT_BA[15:9] — This 7-bits field provides address bits 15 through 9 of the EBT base address, which defines where the Buffer Descriptor Table resides in system memory.

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

30.5.2.10. 帧数寄存器（FRMNUML）

帧号寄存器中包含 11 位的帧号。帧号寄存器需要两个 8 位寄存器来实现。低阶字节包含在 FRMNUML 中，高阶字节包含在 FRMNUMH 中。每当收到一个 SOF 令牌时，这些寄存器就会用当前的帧号进行更新。

地址：USBC_BASEADDR+0x0000_00A0

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	FRM[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 30-11：帧号寄存器（FRMNUML）

FRM[7:0] — Frame Number.

这些位表示 11 位帧数的低阶位。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

30.5.2.11. 帧数寄存器（FRMNUMH）

帧号寄存器中包含 11 位的帧号。帧号寄存器需要两个 8 位寄存器来实现。低阶字节包含在 FRMNUML 中，高阶字节包含在 FRMNUMH 中。每当收到一个 SOF 令牌时，这些寄存器就会用当前的帧号进行更新。

地址：USBC_BASEADDR+0x0000_00A4

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:						FRM[10:8]		
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 30-12：帧号寄存器（FRMNUMH）

FRM[10:8] — Frame Number.
这些位表示 11 位帧数的高阶位。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

30.5.2.12. 令牌寄存器（TOKEN）

令牌寄存器用于在主机模式下执行 USB 事务 (HOST_MODE_EN = 1)。当处理器核心希望对外围设备执行 USB 事务时，它会将令牌类型和端点写入此寄存器。在写入此寄存器之后，USB 模块将对地址寄存器中包含的地址开始指定的 USB 事务。处理器核心在对令牌寄存器执行写入之前，应始终检查控制寄存器中的 TOKEN_BUSY 位是否没有设置。这可以确保在执行令牌命令之前不会被覆盖。在执行令牌命令时也使用地址寄存器和端点控制寄存器 0，因此也必须写入令牌寄存器之前。地址寄存器用于正确选择由令牌命令传输的 USB 外设地址。端点控制寄存器确定在传输期间使用的握手和重试策略。

地址：USBC_BASEADDR+0x0000_00A8

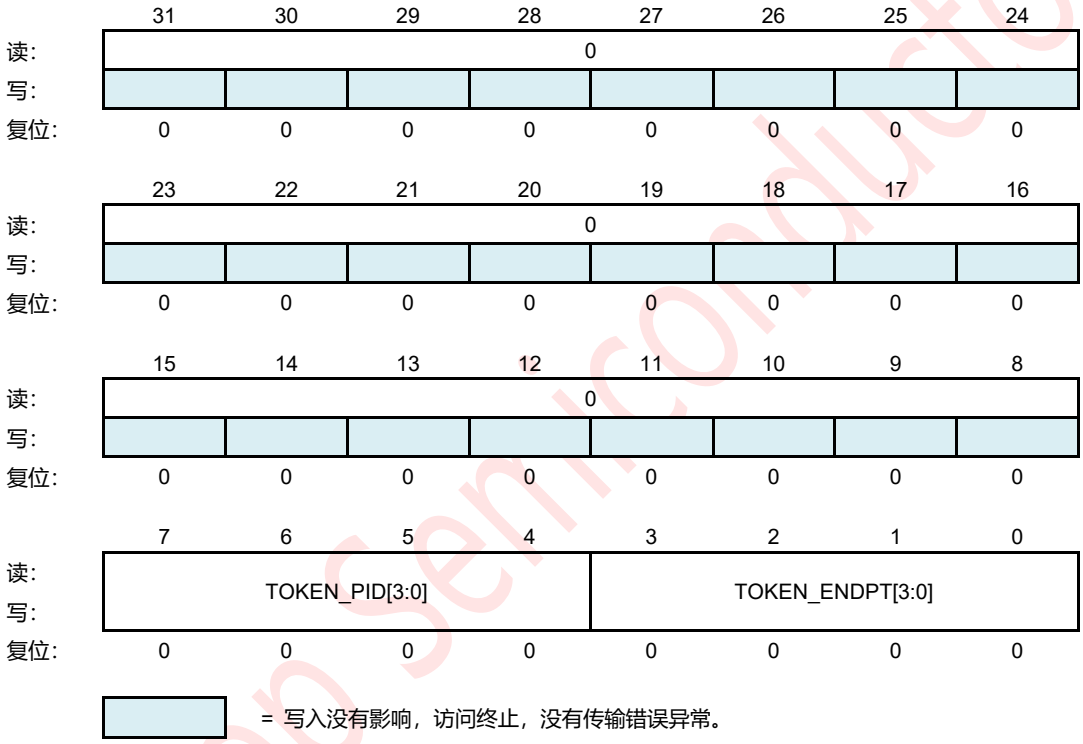


图 30-13：令牌寄存器（TOKEN）

TOKEN_PID[3:0] — This 4-bits field contains the token type executed by the USB Module.

0001 OUT 令牌。USB 模块执行 OUT（TX）事务。

1001 年在令牌。USB 模块执行一个 In（RX）事务。

1101 安装标记。USB 模块执行设置（TX）事务

TOKEN_ENDPT[3:0] — This 4-bits field holds the Endpoint address for the token command. The 4-bits value written must be a valid endpoint.

©Levetop Semiconductor Co., Ltd.

30.5.2.13. SOF 阈值寄存器（SOF_THLD）

SOF 阈值寄存器仅在主机模式下使用（HOST_MODE_EN = 1）。在主机模式下，14 位 SOF 计数器计算 SOF 帧之间的间隔。SOF 必须每 1 毫秒传输一次，因此 SOF 计数器的加载值为 12000。当 SOF 计数器达到零时，将发送帧开始（SOF）令牌。SOF 阈值寄存器用于编程在 SOF 之前的 USB 字节次数，以停止启动令牌数据包事务。此寄存器必须设置为一个值，以确保当 SOF 时间计数为零时不会主动地传输其他数据包。当 SOF 计数器达到阈值时，直到传输 SOF 之后才传输更多的令牌。编程到阈值寄存器的值必须保留足够的时间，以确保最坏情况下的事务完成。一般来说，最坏的情况下的事务是一个 IN 令牌，然后是来自目标的数据包，然后是来自主机的响应。实际所需的时间是总线上最大数据包大小的函数。SOF 阈值的典型值有：

64 字节数据包 = 74；

32 字节数据包 = 42；

16 字节数据包 = 26；

8 字节数据包 = 18。

地址：USBC_BASEADDR+0x0000_00AC

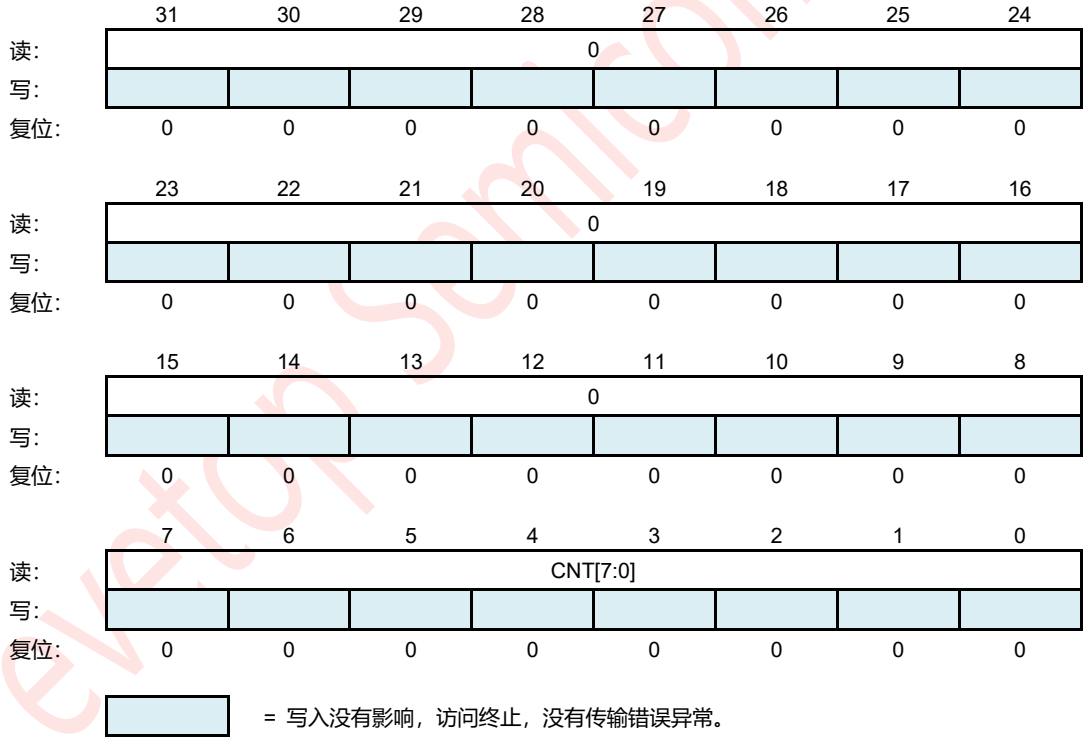


图 30-14：SOF 阈值寄存器（SOF_THLD）

CNT[7:0] — This 8-bits field represents the SOF count threshold in byte times. This register is read only in this device.

©Levetop Semiconductor Co., Ltd.

30.5.2.14. EBT 页面寄存器 2（EBT_PAGE_02）

端点缓冲区表页面寄存器 2 包含一个 8 位值，用于计算当前端点缓冲区表（EBT）在系统内存中的地址。

地址：USBC_BASEADDR+0x0000_00B0

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	EBT_BA[23:16]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 30-15：EBT 页面寄存器 2（EBT_PAGE_02）

EBT_BA[23:16]

这个 8 位字段提供了 EBT 基地址的地址位 23 到 16，它定义了缓冲区描述器表在系统内存中所在的位置。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

30.5.2.15. EBT 页面寄存器 3（EBT_PAGE_03）

端点缓冲区表页面寄存器 3 包含一个 8 位值，用于计算当前端点缓冲区表（EBT）所在在系统内存中所在的地址。

地址：USBC_BASEADDR+0x0000_00B4

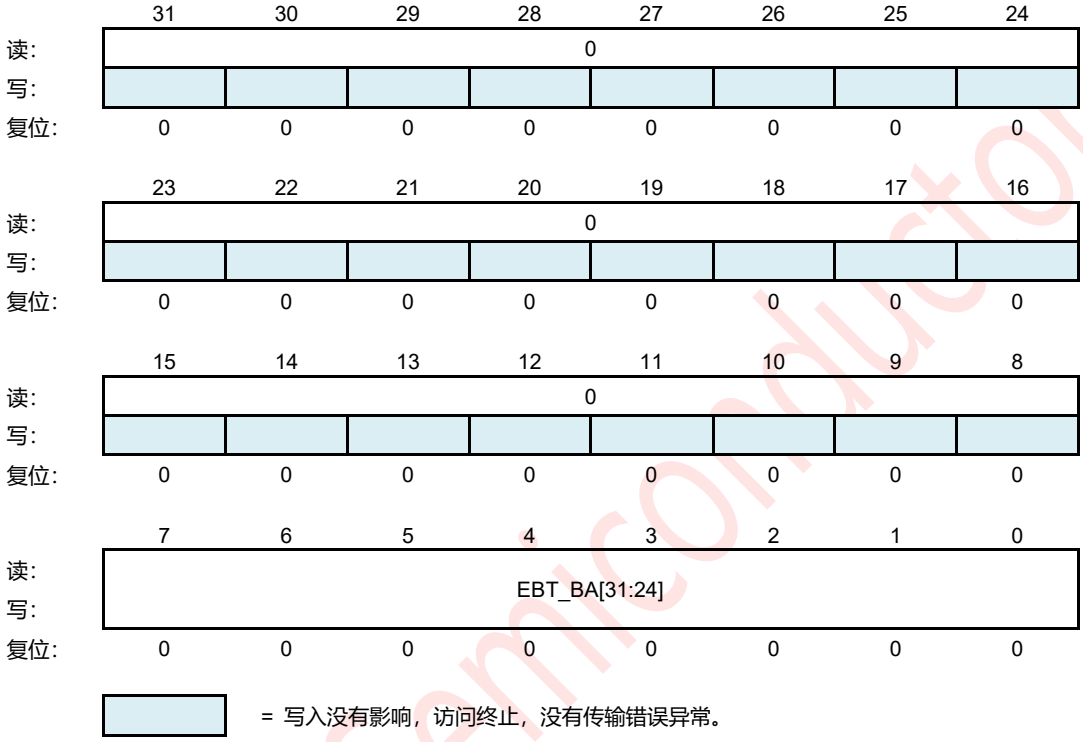


图 30-16：EBT 页面寄存器 3（EBT_PAGE_03）

EBT_BA[31:24]

这个 8 位字段提供了 EBT 基地址的地址位 31 到 24，它定义了缓冲区描述器表在系统内存中所在的位置。

30.5.2.16. 端点控制寄存器（ENDPTn）

端点控制寄存器包含在 USB 模块中可用于解码地址的 8 个端点中的每个端点控制位。这些寄存器的格式如下图所示。端点 0（ENDPT0）与控制管道 0 相关联，这是所有 USB 功能所必需的。因此，在 USB_RST 中断后，处理器核心应该将 ENDPT0 寄存器设置为包含 0x0D。

在主机模式下，ENDPT0 用于确定主机传输的握手、重试和低速特性。对于主机模式控制、批量和中断传输，EP_HSHK 位应该设置为 1。对于等时传输，它应设置为 0。在主机模式下，ENDPT0 的常用值是，控制、批量和中断传输为 0x4D，同步传输为 0x4C。

寄存器偏移地址：

ENDPT0 Address: USBC_BASEADDR + 0x0000_00C0
ENDPT1 Address: USBC_BASEADDR + 0x0000_00C4

©Levetop Semiconductor Co., Ltd.

ENDPT2 Address: USBC_BASEADDR + 0x0000_00C8
ENDPT3 Address: USBC_BASEADDR + 0x0000_00CC
ENDPT4 Address: USBC_BASEADDR + 0x0000_00D0
ENDPT5 Address: USBC_BASEADDR + 0x0000_00D4
ENDPT6 Address: USBC_BASEADDR + 0x0000_00D8
ENDPT7 Address: USBC_BASEADDR + 0x0000_00DC



图 30-17：端点控制寄存器（ENDPTn，n = 0-7）

HOST_WO_HUB — This is a Host mode only bit and is only present in the control register for endpoint 0 (ENDPT0). This bit is read only in this device.

当设置时，此位允许主机与直接连接的低速设备进行通信。当清除后，主机将生成 PRE_PID，然后在根据需要向低速设备发送信号以通过集线器与低速设备进行通信时，切换到低速信令。

RETRY_DIS — This is a Host mode only bit and is only present in the control register for endpoint 0 (ENDPT0). This bit is read only in this device.

设置时，此位将导致主机不重试 NAK 的（否定确认）事务。当事务为 NAK 时，EBT PID 字段用 NAK PID 更新，并设置 TOKEN_DNE 中断。清除此位后，将在硬件中重试 NAKed 事务。当主机试图轮询中断端点时，必须设置此位。

EP_CTL_DIS — This bit, when set, disables control (SETUP) transfers. When this bit is cleared, control transfers are enabled. This applies if and only if the EP_RX_EN and EP_TX_EN bits are also set.

EP_RX_EN — This bit, when set, enables the endpoint for RX transfers.

EP_TX_EN — This bit, when set, enables the endpoint for TX transfers.

©Levetop Semiconductor Co., Ltd.

EP_STALL — 当设置时，此位表示端点已停止。这个位优先于端点启用寄存器中的所有其他控制位，但它仅在 EP_TX_EN = 1 或 EP_RX_EN = 1 时有效。任何对此端点的访问都会导致 USB 模块返回一个停止握手。在端点停止后，它需要从主机控制器中进行干预。

EP_HSHK — When set, this bit enables an endpoint to perform handshaking during a transaction to this endpoint. This bit is generally set unless the endpoint is Isochronous.

表 30-2：端点启用/方向控制

Bit Name			端点启用/方向控制
EP_CTL_DIS	EP_CTL_DIS	EP_CTL_DIS	
X	0	0	Disable endpoint
X	0	1	Enable endpoint for IN(TX) transfers only
X	1	0	Enable endpoint for OUT(RX) transfers only
0	1	1	Enable endpoint for IN, OUT and SETUP transfers.
1	1	1	保留

30.5.2.17. USBPHY 控制寄存器 2（USBPHY_CTRL2）

地址：USBC_BASEADDR+0x0000_0100

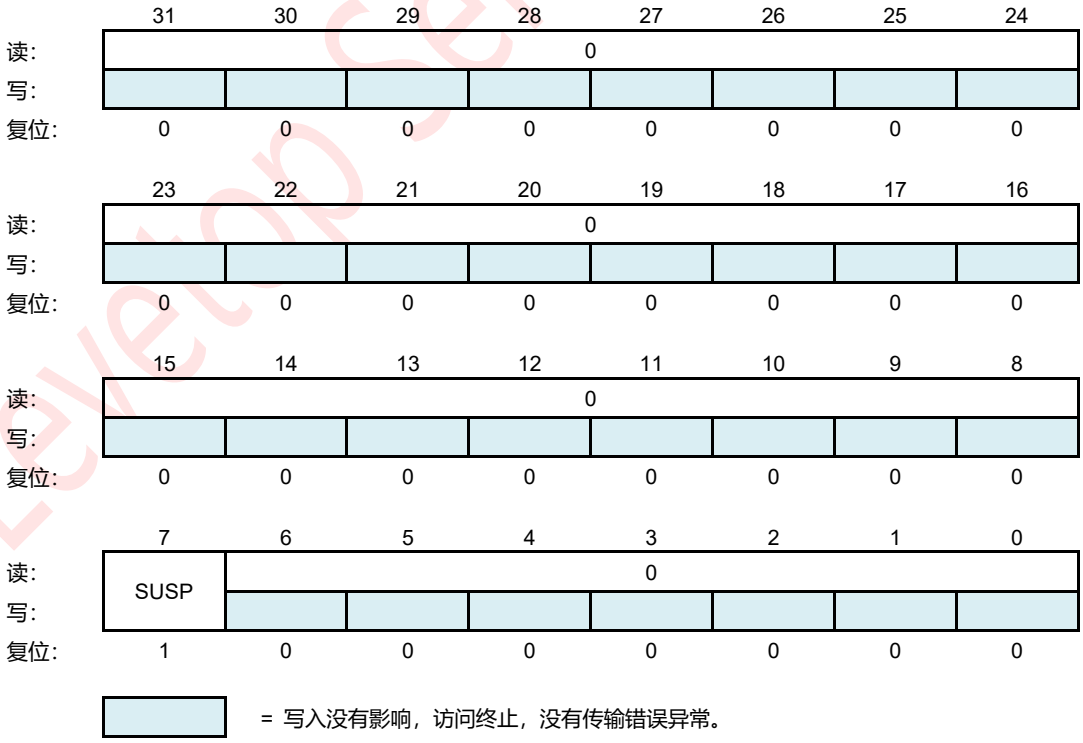


图 30-18：USBPHY 控制寄存器 2（USBPHY_CTRL2）

©Levetop Semiconductor Co., Ltd.

SUSP — Places the USB transceiver into the suspend state.

0 = USB 收发器未处于等待处理状态。

1 = USB 收发器处于等待处理状态。

30.5.2.18. USB PHY 观察寄存器（USB_PHY_OBSERVE）

地址：USBC_BASEADDR+0x0000_0104

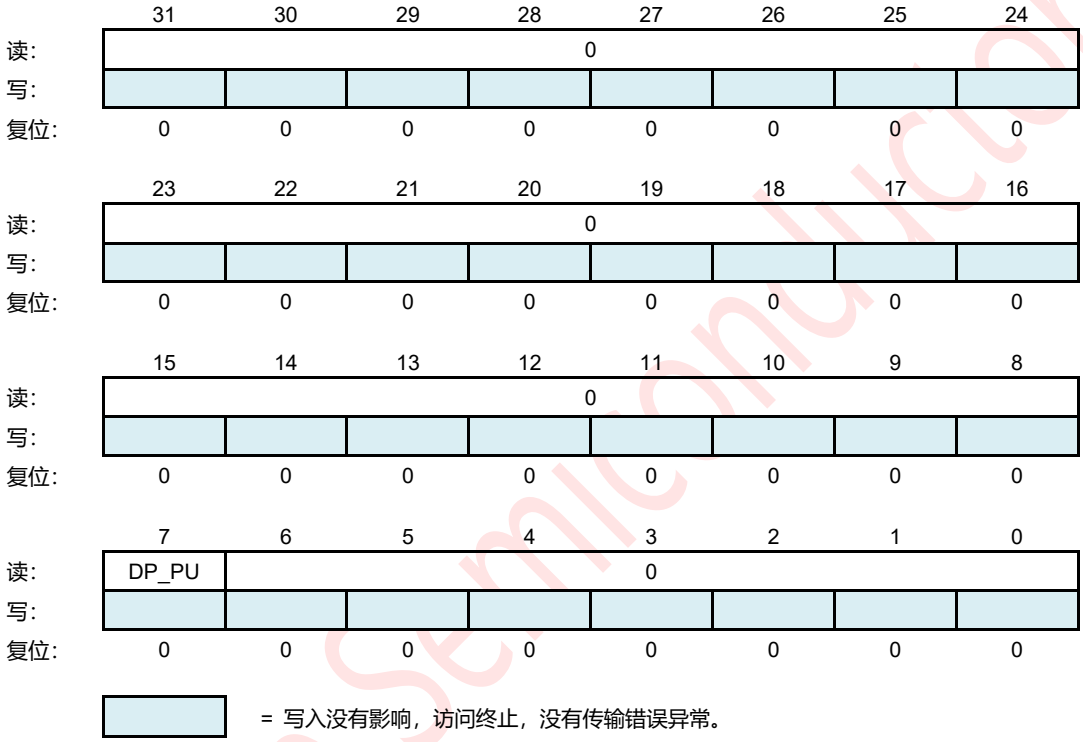


图 30-19：USB PHY 观察寄存器（USB_PHY_OBSERVE）

DP_PU — Provides observability of the D+ Pull Up signal output from the USB OTG module. This bit is useful when interfacing to an external OTG control module via a serial interface.

0 = D+上拉已禁用。

1 = D+上拉已启用。

©Levetop Semiconductor Co., Ltd.

30.5.2.19. USBPHY GPIO 寄存器（USB_PHY_GPIO）

地址：USBC_BASEADDR+0x0000_0108

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	GPIOEN	RX_RCV	RX_DM	RX_DP	TX_DM_EN	TX_DP_EN	TX_DM	TX_DP
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 30-20：USB PHY GPIO 寄存器（USB_PHY_GPIO）

GPIOEN — USB PHY GPIO Mode control.

0 = USB PHY 不处于 GPIO 模式

1 = USB PHY 处于 GPIO 模式

RX_RCV — USB PHY Differential input data during USBPHY GPIO mode.

RX_DM — USB PHY DM input data during USBPHY GPIO mode.

RX_DP — USB PHY DP input data during USBPHY GPIO mode.

TX_DM_EN — USB PHY DM output direction control during USBPHY GPIO mode.

0 = DM 为输入方向

1 = DM 为输出方向

TX_DP_EN — USB PHY DP output direction control during USBPHY GPIO mode.

0 = DP 为输入方向

1 = DP 为输出方向

TX_DM — USB PHY DM output data during USBPHY GPIO mode.

TX_DP — USB PHY DP output data during USBPHY GPIO mode.

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

30.5.2.20. USB 恢复启用寄存器（USB_RESMEN）

地址：USBC_BASEADDR+0x0000_010C

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0	0	USBRESMEN	USBPHYCLKEN	0	0	USBIRQ	RESUME
写:	USB_SFT_RESET							
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 30-21：USB 简历启用寄存器（USB_RESMEN）

USB_SFT_RESET — USB Soft Reset control bit.

USB 模块将通过写入这个位来复位。

USBRESMEN — USB resume wakeup enable control bit

如果设置了用户，并且在 USB 总线上检测到 K 状态（恢复信令），则恢复位将被设置。这将触发一个异步中断，将唤醒单片机从停止模式，并启用时钟到 USB 模块。

USBPHYCLKEN — USB PHY clock enable control bit.

USBIRQ — All enabled interrupt in INT_STAT register.

如果设置了 INT_STAT 中的任何一个，并且设置了相应的 INT_ENB，则将设置此位。

RESUME — USB DP/DM resume status bit.

如果设置了用户位，并且在 USB 总线上检测到 k 状态（恢复信令），则在 CPU 从停止模式唤醒并清除用户位位时设置该位。

©Levetop Semiconductor Co., Ltd.

30.5.2.21. USB PHY 控制寄存器 3（USBPHY_CTRL3）

地址：USBC_BASEADDR+0x0000_0118

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	PHY_1.5V PDnote1	PHY_OEN note1	PHY_RESETb	USB_BYTE_SWAP	PLL_PD	PHY_SUSPEND	PHY_SUSPEND_SEL	48_60M_SEL
写:								
复位:	0	0	0	0	1	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

note1 这些位在这个芯片中没有什么意义

图 30-22：USB PHY 控制寄存器 3（USBPHY_CTRL3）

PHY_1.5VPD — USB PHY core power control.

0 = USB PHY 核心电源将不会丢失。

1 = USB PHY 核心电源将丢失。

PHY_OEN — USB PHY output isolate control.

0 = USB PHY 输出将不会被隔离。

1 = USB PHY 输出将被隔离。

PHY_RESETb — USB PHY RESET control.

0 = USB PHY 将处于复位状态。

1 = USB PHY 将为正常状态。

USB_BYTE_SWAP — When this bit is set, the data received/transmitted by USB will be swapped when reading from or writing to the system memory.

0 = USB 已接收或传输的数据将不会被交换。

1 = USB 已接收或传输的数据将被交换。

PLL_PD — USB PHY PLL power down mode control.

0 = USB PHY PLL未处于断电状态。

1 = USBPHY PLL 处于断电状态。

©Levetop Semiconductor Co., Ltd.

PHY_SUSPEND — USB PHY Suspend control

0 = USB PHY 未处于暂停状态。

1 = USBPHY 处于暂停状态。

PHY_SUSPEND_SEL — USB PHY Suspend control selection.

0 = USB PHY 等待处理由 USBPHY_CTRL2[SUSP]控制。

1 = USB PHY 等待处理系统由 USBPHY_CTRL3[PHY_SUSPEND]控制。

48_60M_SEL — USB PHY 48/60Mhz clock selection.

0 = 48Mhz 被用作 USB SIE 解码时钟。

1 = 60Mhz 被用作 USB SIE 解码时钟。

30.5.2.22. USB PHY 控制寄存器 4（USBPHY_CTRL4）

地址：USBC_BASEADDR+0x0000_011C

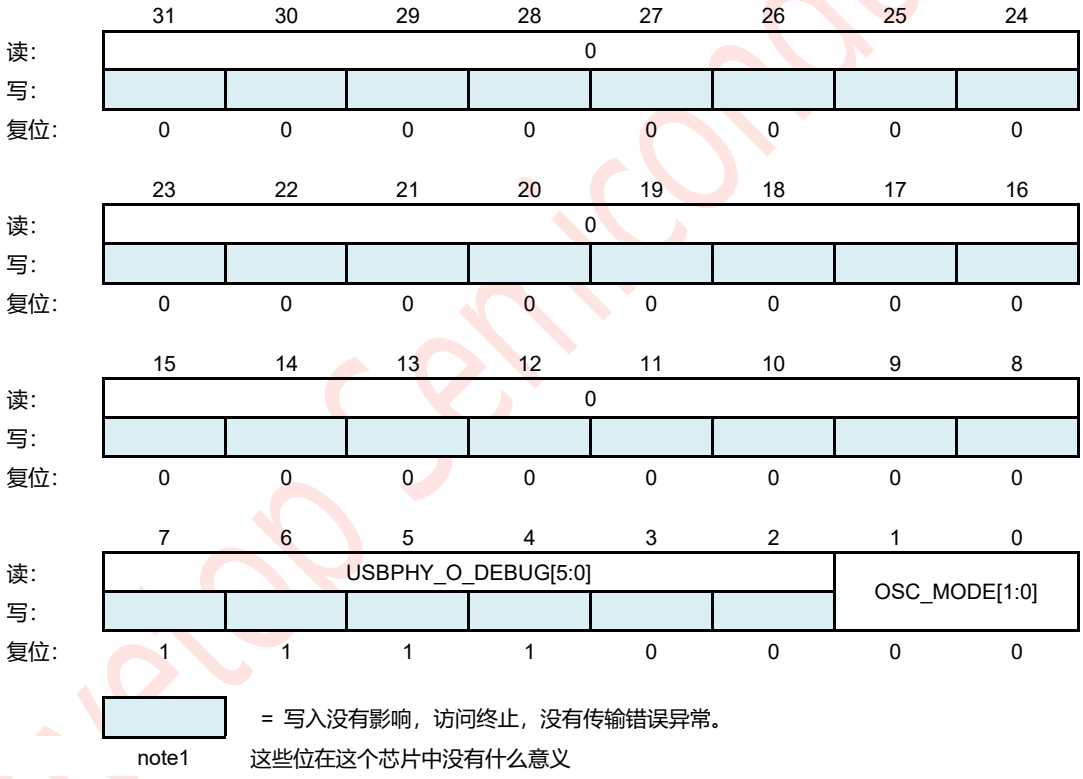


图 30-23：USB PHY 控制寄存器 4（USBPHY_CTRL4）

USBPHY_O_DEBUG[5:0] — USBPHY Output signals are for debug purpose and are meaningless during normal work.

OSC_MODE[1:0] — USB PHY clock mode selection.

©Levetop Semiconductor Co., Ltd.

表 30-3：USB PHY 振荡器模式选择

OSC_MODE[1:0]	振荡器模式选择
00	自动检测振荡器
01	自动检测振荡器（快速模拟模式）
10	选择内部振荡器
11	选择外部振荡器

30.6. 功能描述

USB-FS 2.0 全速/低速模块通过状态寄存器、控制寄存器和内存中的数据结构与处理器核心进行通信。

30.6.1. 数据结构

设备操作的功能是将存储器映像中的请求传输到通用串行总线和从通用串行总线传输。为了有效地管理 USB 端点通信，USB 模块在系统内存中实现了一个端点缓冲区表（EBT）。见图 30-24.

30.6.2. 端点缓冲区表

为了有效地管理 USB 端点通信，USB 模块在系统内存中实现了一个缓冲区描述符表（EBT）。EBT 位于系统内存中的 512 字节边界上，并由 EBT 页面寄存器指向。每个端点方向都需要两个 8 字节的端点缓冲区表条目。

因此，一个有 16 个完全双向端点的系统将需要 512 字节的系统内存来实现 EBT。两个端点缓冲表（EBT）条目允许每个端点方向的 EVEB 条目和 ODD EB 条目。这允许微处理器在处理另一个 EB 条目时，USB 模块正在处理另一个 EB 条目。双缓冲 EB 入口以这种方式允许 USB SIE 轻松地以 USB 提供的最大吞吐量传输数据。

该软件 API 通过在需要时更新 EBT 来智能地管理 USB SIE 的缓冲区。这允许 USB SIE 有效地管理数据传输和接收，而微处理器执行通信开销处理和其他功能依赖的应用程序。因为缓冲区在微处理器和 USB 模块之间共享，所以使用一个简单的信号量机制来区分谁被允许更新系统内存中的 EBT 和缓冲区。当 EB 条目属于微处理器时，一个信号量位，即 OWN 位，被清除为 0。当 OWN 位为 0 时，微处理器允许对系统内存中的缓冲区进行读写访问。当 OWN 位设置为 1 时，系统内存中的 EB 入口和缓冲区归 USB 模块所有。USB 模块现在有完全的读写访问权限，微处理器不应该修改 EB 条目或其相应的数据缓冲区。EB 条目还包含指向实际缓冲区在系统内存中所在位置的间接地址指针。这种间接地址机制如下图所示。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

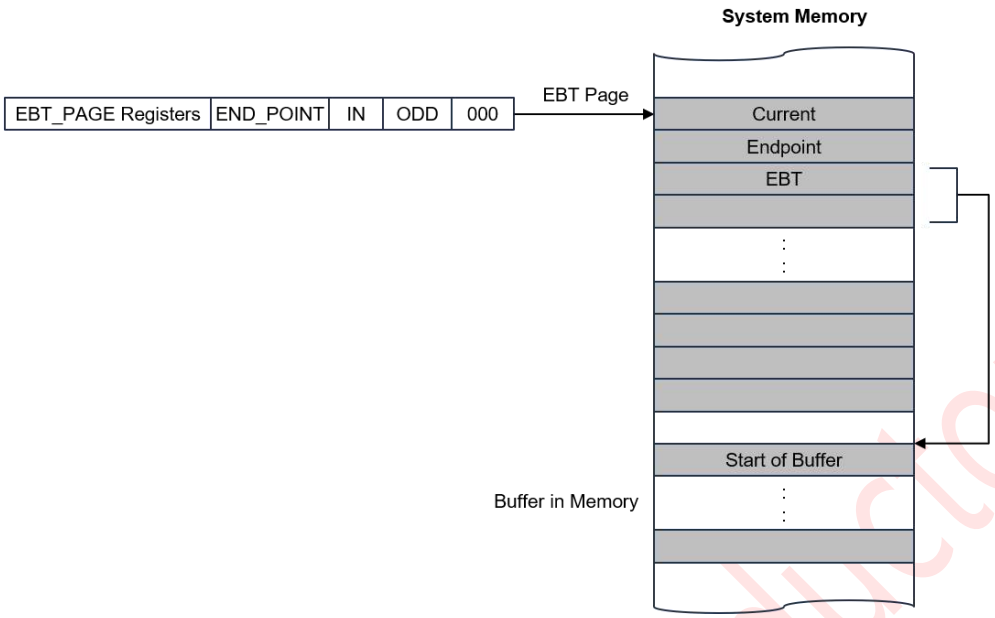


图 30-24：端点缓冲区表

30.6.3. Rx vs。Tx 作为一个 USB 目标设备

中心命名法用于描述 USB SIE 核心和 USB 主机之间的数据传输方向：

Rx（或接收）：描述将数据从 USB 移动到内存的传输。

Tx（或传输）：描述将数据从内存移动到 USB 的传输。

下表显示了数据方向如何对应于主机和目标设备应用程序中的 USB 令牌类型。

表 30-4：USB 目标设备的数据方向

	RX	TX
Device	OUT or Setup	IN

30.6.4. 寻址端点缓冲区表条目

当通过 USB 模块或微处理器访问端点数据时，了解端点缓冲区表的寻址机制是很有用的。一些值得关注的关注点是：

- 端点缓冲区表占用多达 512 字节的系统内存。
- 16 个双向端点的全 EBT 为 512 个字节。
- 每个 USB 端点方向需要 16 个字节。
- 端点少于 16 个端点的应用程序需要更少的 RAM 来实现 EBT。
- EBT 页面寄存器指向 EBT 的起始位置。

©Levetop Semiconductor Co., Ltd.

- EBT 必须位于系统内存中的 512 字节边界上。
- 所有启用的 TX 和 RX 端点 EB 条目都被索引到 EBT 中，以允许通过 USB 模块或 CPU 轻松访问。

当接收到启用端点上的 USB 令牌时，USB 模块使用其集成的 DMA 控制器来询问 EBT。USB SIE 读取相应的端点 EB 条目，以确定它是否拥有系统内存中的 EB 条目和相应的缓冲区。

为了计算 EBT 中的入口点，将 EBT_PAGE 寄存器与当前端点以及 TX 和 ODD 字段连接起来，形成一个 32 位的地址。此地址机制如下图所示：

表 30-5：EBT Address Calculation Fields

Field	Description
EBT_PAGE	EBT_PAGE registers in the Control Register Block
END_POINT	END POINT field from the USB TOKEN
TX	1 for an TX transmit transfers and 0 for an RX receive transfers
ODD	This bit is maintained within the USB SIE. It corresponds to the buffer currently in use.
	The buffers are used in a ping-pong fashion.

30.6.5. 端点缓冲区表格式

端点缓冲区表（EBT）为 USB 模块和微处理器提供端点控制信息。端点缓冲区表根据它是 USB 模块还是微处理器而具有不同的含义。

USB SIE 控制器使用存储在 EB 入口中的数据来确定：

- 谁拥有系统内存中的缓冲区
- 数据 0 或数据 1PID
- 包完成后释放自己 w
- 无地址增量(FIFO 模式)
- 启用数据切换同步
- 要传输或接收多少数据
- 缓冲区存放在系统内存中的位置

而微处理器使用存储在 EB 入口中的数据来确定：

- 谁拥有系统内存中的缓冲区
- 数据 0 或数据 1PID
- 接收到的令牌 PID
- 传输或接收了多少数据
- 缓冲区存放在系统内存中的位置

©Levetop Semiconductor Co., Ltd.

EB 条目的格式如下图所示。

表 30-6：端点缓冲区表字节格式

63	62	61	60	59	58	57	56
保留						BC[9:8]	
55	54	53	52	51	50	49	48
BC[7:0]							
47	46	45	44	43	42	41	40
保留							
39	38	37	36	35	34	33	32
OWN	DATA0/1	KEEP/TOK_PID[3]	NINC/TOK_PID[2]	DTS/TOK_PID[1]	EBT_STALL/TOK_PID[3]	0	0
31	30	29	28	27	26	25	24
ADDR[31:24]							
19	22	21	20	19	18	17	16
ADDR[23:16]							
11	14	13	12	11	10	9	8
ADDR [15:8]							
3	6	5	4	3	2	1	0
ADDR [7:0]							

表 30-7：端点缓冲区表字节字段

Field	描述
OWN	**OWN** — This OWN bit determines who currently owns the buffer. Except when KEEP = 1, the USB SIE writes a 0 to this bit when it has completed a token. The USB module ignores all other fields in the EB entry when OWN = 0. Once the EB entry has been assigned to the USB module (OWN = 1), the MCU should not change it in any way. This byte of the EB entry should always be the last byte the MCU (firmware) updates when it initializes a EB entry. Although the hardware will not block the MCU from accessing the EB entry while owned by the USB SIE, doing so may cause undefined behavior and is generally not recommended. 0: The MCU has exclusive access to the entire EB entry 1: The USB module has exclusive access to the EB entry
DATA0/1	**Data Toggle** — This bit defines if a DATA0 field (DATA0/1 = 0) or a DATA1 (DATA0/1 = 1) field was transmitted or received. It is unchanged by the USB module. 0: Data 0 packet 1: Data 1 packet

©Levetop Semiconductor Co., Ltd.

Field	描述
KEEP/ TOK_PID[3]	**KEEP / EB entry Token PID [3]** — Typically this bit is set (that is, 1) with ISO endpoints feeding a FIFO. The microprocessor is not informed that a token has been processed, the data is simply transferred to or from the FIFO. If KEEP is set, normally the NINC bit is also set to prevent address increment. 0: Bit 3 of the current token PID is written back in to the EB entry by the USB SIE. Allows the USB SIE to release the EB entry when a token has been processed. 1: This bit is unchanged by the USB SIE. If the OWN bit also is set, the EB entry remains owned by the USB SIE forever.
NINC/ TOK_PID[2]	**No Increment / EB entry Token PID [2]** — The No Increment (NINC) bit disables the DMA engine address increment. This forces the DMA engine to read or write from the same address. This is useful for endpoints when data needs to be read from or written to a single location such as a FIFO. Typically this bit is set with the KEEP bit for ISO endpoints that are interfacing to a FIFO. 0: The USB SIE writes bit 2 of the current token PID to the EB entry. 1: This bit is unchanged by the USB SIE.
DTS/ TOK_PID[1]	**Data Toggle Synchronization / EB entry Token PID [1]** — This bit enables data toggle synchronization. Setting this bit enables the USB SIE to perform Data Toggle Synchronization. • If KEEP = 0, bit 1 of the current token PID is written back to the EB entry. • If KEEP = 1, this bit is unchanged by the USB SIE. 0: No data toggle synchronization is performed. 1: Data toggle synchronization is performed.
EBT_STALL/ TOK_PID[0]	**EBT Stall / EB entry Token PID [0]** — Setting this bit will cause the USB module to issue a STALL handshake if a token is received by the SIE that would use the EBT in this location. The EBT is not consumed by the SIE (the OWN bit remains and the rest of the EB entry is unchanged) when the EBT_STALL bit is set. • If KEEP = 0, bit 0 of the current token PID is written back to the EB entry. • If KEEP = 1, this bit is unchanged by the USB SIE. 0: EBT stall is disabled 1: USB will issue a STALL handshake if a token is received by the SIE that would use the EBT in this location
TOK_PID[n]	Bits [37:34] can also represent the current token PID. The current token PID is written back in to the EB entry by the USB SIE when a transfer completes. The values written back are the token PID values from the USB specification: • 0x1 for an OUT token. • 0x9 for an IN token. • 0xD for a SETUP token.
BC[9:0]	**Byte Count** — The Byte Count bits represent the 10-bits byte count. The USB module serial interface engine (SIE) will change this field upon the completion of a RX transfer with the byte count of the data received.
ADDR[31:0]	The Address bits represent the 32-bits buffer address in system memory. These bits are unchanged by the USB SIE.

©Levetop Semiconductor Co., Ltd.

30.6.6. USB Transaction

当 USB SIE 传输或接收数据时，它使用“寻址端点缓冲区表项”表中显示的地址生成来计算 EBT 地址。

如果为 OWN = 1，则会发生以下进程：

1. USB SIE 读取 EBT。
2. SIE 通过 DMA 将数据传输到 EB 条目的 ADDR 字段指向的缓冲区。
3. 当令牌完成时，USB SIE 将更新 EBT，如果是 KEEP = 0，则会将 OWN 位更改为 0。
4. 将更新 STAT 寄存器，并设置 TOK_DNE 中断。
5. 当微处理器处理 TOK_DNE 中断时，它从状态寄存器中读取处理端点所需的所有信息。
6. 此时，微处理器分配一个新的 EB 条目，以便可以为该端点传输或接收额外的 USB 数据，然后处理最后一个 EB 条目。

下图显示了在读取 EBT 和 OWN = 1 后如何处理典型 USB 令牌的时间轴。

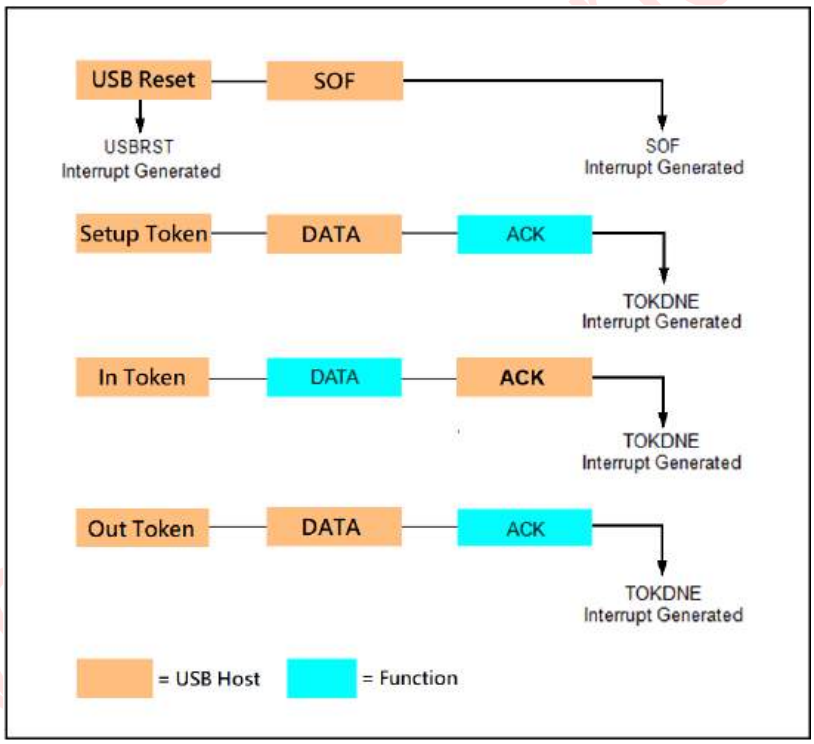


图 30-25：USB Token Transaction

该 USB 有两个来源的 DMA 溢出错误：

Memory Latency 内存延迟

DMA 接口上的内存延迟可能过高，并导致接收 FIFO 溢出。这主要是一个硬件性能问题，通常由瞬时内存访问问题引起。

©Levetop Semiconductor Co., Ltd.

Oversized Packets 超大数据包

接收到的数据包可能大于协商的最大数据包大小。通常，这是由一个软件错误引起的。对于由于数据包超大而导致的 DMA 溢出错误，USB 规范并不明确。它假定双方都有正确的软件驱动程序。打开数据包可能导致已经超大的数据包数据的重新传输。因此，为了响应超大的数据包，USB 核心继续继续进行非等时传输。

表 30-8：USB 响应的 DMA 覆盖错误

由于内存延迟而导致的错误	由于超大号的数据包而导致的错误
不确认（NAK）或总线超时（BTO）—请参见“错误中断状态寄存器（ERR_STAT）”中的第 4 位。	继续确认（确认）非同步用户的数据包。
—	写入内存的数据被剪辑到 MaxPacket 大小，以免损坏系统内存。
DMA_ERR 位是在 ERR_STAT 寄存器中设置的。根据 INT_ENB 和 ERR_ENB 寄存器的值，核心可能会断言一个中断，以通知处理器的 DMA 错误。	断言 ERR_STAT 寄存器的 DMA_ERR 位（可以触发中断），并触发 TOK_DNE 中断。 **提示：EBT 的 TOK_PID 字段不是 1111，因为 DMA_ERR 不是由于延迟所致。**
EBT 没有被写回，也不会触发 TOK_DNE 中断，因为它假设第二次尝试已经排队，并将在未来成功。	写入 EBT 的包长度字段是 MaxPacket 值，它表示实际写入内存的裁剪数据的长度。
从这里开始，软件可以为未来的事务决定一个适当的行动方案，如停止端点、取消转移、禁用端点等。	

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

31.模数转换器（ADC）

31.1. 功能介绍

12 位 ADC 是一个连续的近似模数转换器。它有多达 9 个通道，允许它测量来自 8 个外部源和 2 个内部源的信号。各种通道的 A/D 转换可以在单个、连续、扫描或不连续模式下进行。ADC 的结果存储在一个 12 位 x8 深度的 FIFO 中，数据格式可以是左对齐或向右对齐的。

模拟看门狗功能允许应用程序检测输入电压是否超出用户定义的高或低阈值。

有效的低功率模式允许低频功耗。

31.2. ADC 主要功能

- 高性能
  - 12 位、10 位、8 位或 6 位的可配置分辨率
  - ADC 转换时间：12 位分辨率为 1.0μs（1MHz），10 位分辨率为 0.88 μs，通过降低分辨率可以获得更快的转换时间。
  - 可编程采样时间
  - 具有内置数据一致性的数据对齐
  - DMA 支持
- 低功耗
  - 应用程序可以降低低功率操作时的 PLCK 频率，同时仍然保持最佳的 ADC 性能。例如，无论 PCLK 的频率如何，都保持 1.0 μs 的转换时间。
  - 等待模式：防止具有低频 PLCK 的应用程序中的 ADC 溢出
  - 自动关闭模式：ADC 是自动关闭，除了在活动转换阶段。这大大降低了 ADC 的功耗。
- 模拟输入通道
  - 8 个外部模拟输入
  - 1 个通道的内部参考电压
  - 内部温度传感器的 1 个通道
- 可以启动转换的开始时间：
  - 通过软件
  - 通过具有可配置极性的硬件触发器
- 转换模式
  - 可以转换单个通道或可以扫描一系列通道。
  - 单模式为每个触发器转换选定的输入一次
  - 连续模式连续转换选定的输入

©Levetop Semiconductor Co., Ltd.

-不连续模式

- 在采样结束、转换结束、序列转换结束以及发生模拟看门器或溢出事件时中断生成。
- 模拟监视器
- 单端和差分输入配置
- 转换器使用内部引用或外部引用

31.3. ADC 功能描述

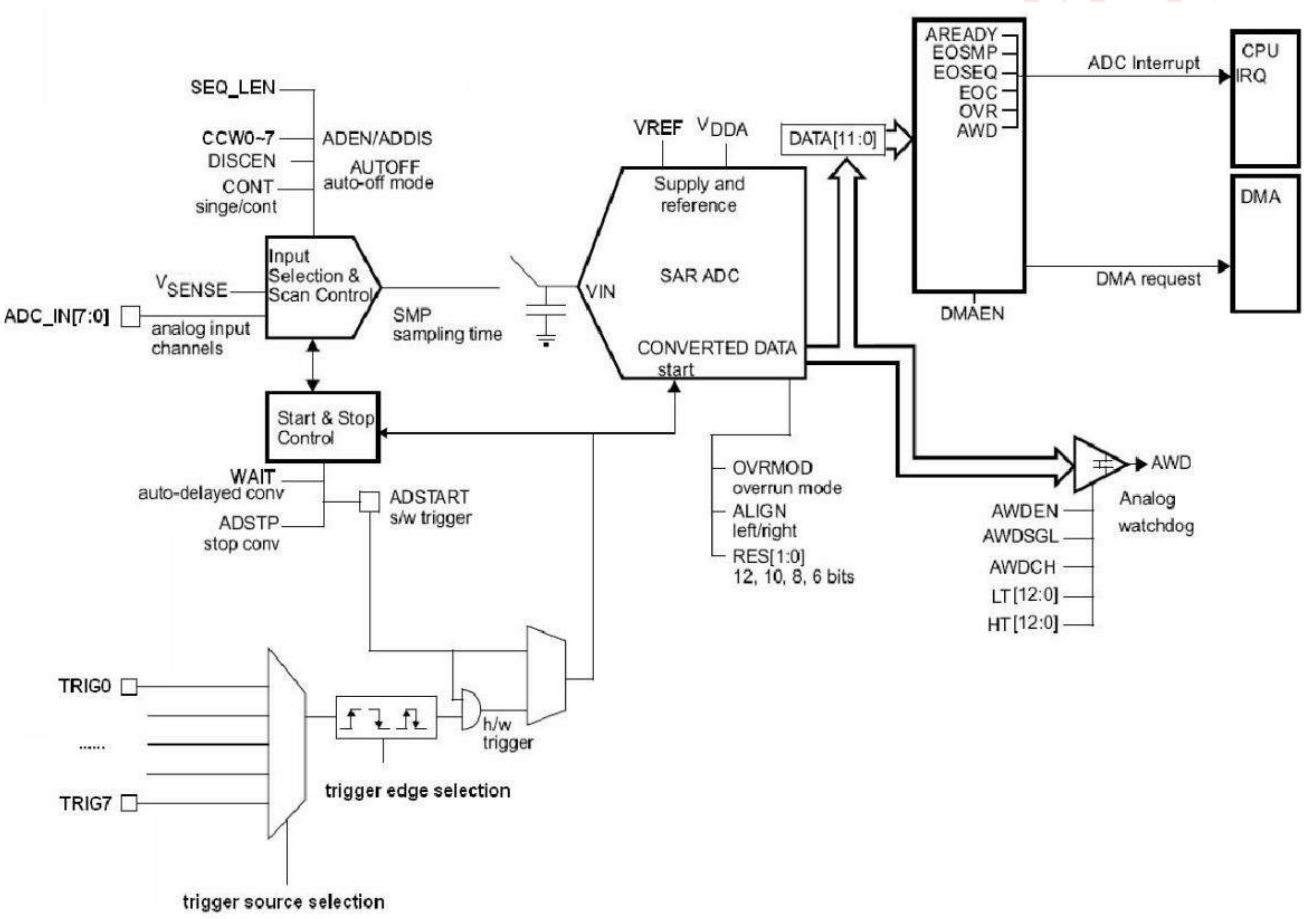


图 31-1：ADC 方块图

©Levetop Semiconductor Co., Ltd.

31.3.1. ADC 开关控制（ADEN、ADDIS、ADRDY）

在单片机通电时，ADC 被禁用并处于断电模式（ADEN = 0）。如图 31-2 所示，ADC 需要一个 tSTAB 的稳定时间（~2.0μs）才能开始准确转换。

有两个控制位用于启用或禁用 ADC：

- 设置 ADEN = 1 以启用 ADC。一旦 ADC 准备好进行操作，就会设置 ADRDY 标志。
- 设置 ADDIS = 1 以禁用 ADC，并将 ADC 置于断电模式。一旦 ADC 被完全禁用，硬件就会自动清除 ADEN 和 ADDIS 位。

然后，可以通过设置已启动 = 1，或者在启用触发器时发生外部触发器事件来开始转换。

按照以下步骤启用 ADC：

- 在 ADC_CR 寄存器中设置 ADEN = 1。
- 等待到 ADC_ISR 寄存器中的 ADRDY = 1（ADRDY 在 ADC 启动时间之后设置）。如果通过在 ADC_IER 寄存器中设置 ADRDYIE 位来启用中断，则可以通过中断来处理。

按照以下步骤禁用 ADC：

- 检查 ADC_CR 寄存器中启动 = 0 以确保没有正在进行的转换。如果需要，请将 1 写入 ADC_CR 寄存器中的 ADSTP 位，并等待，以停止任何正在进行的转换，直到在 0 处读取这个位。
- 在 ADC_CR 寄存器中设置 ADDIS = 1。
- 如果应用程序需要，请等待到 ADC_CR 寄存器中的 ADEN = 0，这表明 ADC 已被完全禁用（一旦 ADEN = 0，ADdis 将自动复位）。

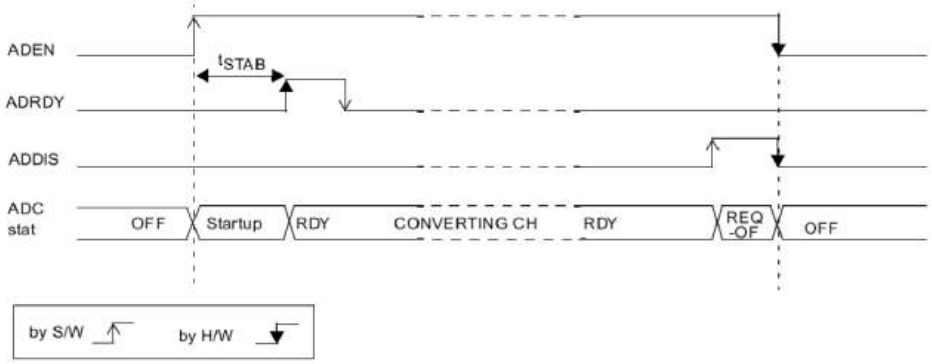


图 31-2：启用/禁用 ADC

提示： 在自动关闭模式（自动关闭 = 1）下，通过硬件自动执行开关阶段，并且不设置 ADRDY 标志。

©Levetop Semiconductor Co., Ltd.

31.3.2. ADC 时钟

ADC 具有双时钟域架构，如图 31-3 所示，其优点是无论选择 IPG 时钟方案，都能达到最大 ADC 时钟频率。

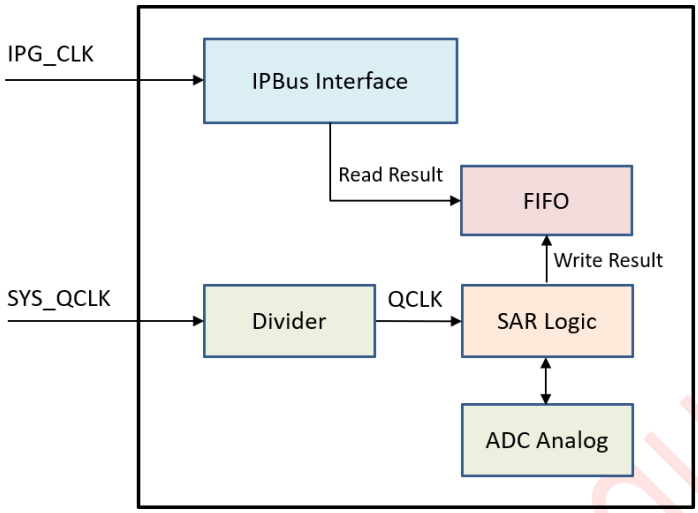


图 31-3：ADC 时钟方案

31.3.3. 配置 ADC

如果 ADC 被禁用（ADEN 必须为 0），则软件必须写入 ADC_CR 寄存器中的 ADEN 位。如果 ADC 已启用，并且没有等待请求禁用 ADC（ADEN = 1 和 ADDIS = 0），则软件只能写入 ADC_CR 寄存器中的 ADSTART 和 ADDIS 位。

对于 ADC_IER、ADC_CFGRi、ADC_SMPR、ADC_TR、ADC_CHSELRi 和 ADC_WDG 寄存器中的所有其他控制位，软件只能在没有正在进行的转换时写入配置控制位（附加启动 = 0）。

如果 ADC 已启用（可能转换）并且没有禁用 ADC 的待定请求（启动 = 1 和 ADDIS = 0），软件必须只写入 ADC_CR 寄存器中的 ADSTP 位。

提示： 没有硬件保护，防止软件进行上述规则所禁止的写操作。如果发生这样种禁止的写访问，ADC 可能进入未定义状态。在这种情况下，要恢复正确的操作，必须禁用 ADC（ADDIS = 1）。

31.3.4. 通道选择（CCWi）

最多有 10 个多路复用通道：

- 8 个模拟输入从引脚(ADC_IN0……ADC_IN7)
- 2 个内部模拟输入（细化和温度传感器）

可以转换单个通道或自动扫描通道序列。

要转换的通道的序列被组织为 CCW[0]，然后是 CCW[1]，……，然后是 CCW[7]。CCWi 是用 ADC_CHSELRi 编写的。序列长度在 ADC_CFGR2 的 SEQ_LEN[2:0]中编程。例如，如果序列长度设置为 3，则序列被组织为 CCW[0]，然后是 CCW[1]，然后是 CCW[2]。

©Levetop Semiconductor Co., Ltd.

通道解码情况如表 31-1 所示.

表 31-1：通道解码

CCWi[3:0]	通道选择
4’ b0000	ADC_IN0
4’ b0001	ADC_IN1
4’ b0010	ADC_IN2
4’ b0011	ADC_IN3
4’ b0100	ADC_IN4
4’ b0101	ADC_IN5
4’ b0110	ADC_IN6
4’ b0111	ADC_IN7
4’b1110	VREFINT
4’b1111	Temperature Sensor

31.3.5. 可编程采样时间（SMP）

在开始转换之前，ADC 需要在要测量的电压源和 ADC 的嵌入式采样电容器之间建立一个直接的连接。该采样时间必须足以使输入电压源为样品充电，并将电容器保持到输入电压电平。

具有可编程的采样时间，允许根据输入电压源的输入电阻来调整转换速度。ADC 对多个 ADC 时钟周期的输入电压进行采样，这些周期可以使用 ADC_SMPR 寄存器中的 SMP[3:0]位进行修改。这个可编程的采样时间对所有通道都是通用的。

ADC 通过设置 EOSMP 标志来表示采样阶段的结束。

31.3.6. 单次转换模式（CONT = 0）

在单转换模式下，ADC 对序列转换一次。当 ADC_CFGR1 寄存器中的 CONT = 0 时，将选择此模式。转换由以下一个开始：

- 在 ADC_CR 寄存器中设置已启动位
- 硬件触发事件

在序列内，每次转换完成后：

- 转换后的数据存储在 FIFO 中
- 已设置了 EOC（转换结束）标志
- 如果设置了 EOCIE 位，则会生成一个中断

在转换顺序完成后：

- 已设置 EOSEQ（序列结束）标志
- 如果设置了 EOSEQIE 位，则会生成一个中断

©Levetop Semiconductor Co., Ltd.

然后 ADC 停止，直到发生新的外部触发事件或重新设置已启动位。

提示：要转换单个通道一次，请编写一个长度为 1 的序列。

31.3.7. 连续转换模式（CONT = 1）

在连续转换模式下，当发生软件或硬件触发事件时，ADC 执行一系列转换，转换一次，然后自动重新启动并连续执行相同的转换序列。当 ADC_CFGR1 寄存器中的 CONT = 1 时，将选择此模式。转换由以下一个开始：

- 在 ADC_CR 寄存器中设置已启动位
- 硬件触发事件

在序列内，每次转换完成后：

- 转换后的数据存储在 FIFO 中
- 已设置了 EOC（转换结束）标志
- 如果设置了 EOCIE 位，则会生成一个中断

在转换顺序完成后：

- 已设置 EOSEQ（序列结束）标志
- 如果设置了 EOSEQIE 位，则会生成一个中断

然后，一个新的序列立即重新启动，ADC 不断重复转换序列。

提示：不可能同时启用不连续模式和连续模式：
禁止同时设置位 DISCEN = 1 和 CONT = 1。

31.3.8. 正在启动转换（ADSTART）

软件通过设置已启动的 = 1 来启动 ADC 转换。
当设置了 ADSTART 时，转换为：

- 如果触发模式 = 0x0（软件触发器），则立即启动
- 在选定的硬件触发器的下一个活动边缘，如果触发模式 = 0x0

该启动位还用于指示当前是否正在进行 ADC 操作。在启动 = 0 时，可以重新配置 ADC，这表明 ADC 处于空闲状态。

重新启动位由硬件清除：

- 在单个模式下与软件触发器
  _在转换序列的任何端（EOSEQ = 1）
- 在不连续的模式下与软件触发器
  _在转换的任何一个结束处
- 在所有情况下
  _在执行了由软件调用的 ADSTP 过程后

©Levetop Semiconductor Co., Ltd.

提示：

- 在连续模式（CONT = 1）中，当设置 EOSEQ 标志时，启动启动位不会被硬件清除，因为序列会自动重新启动。
- 当在单一模式下选择硬件触发器时，当设置 EOSEQ 标志时，硬件不会清除 ADSTART。这避免了软件需要再次设置重新启动位，并确保不会错过下一个触发事件。

31.3.9. 时间安排

转换开始和转换结束之间经过的时间是配置的采样时间加上根据数据分辨率的连续近似时间的总和：

tADC = tSMPL + tSAR = [4|min + 12|12-bits] x tQCLK = 1µs |min (for fQCLK = 16 MHz)

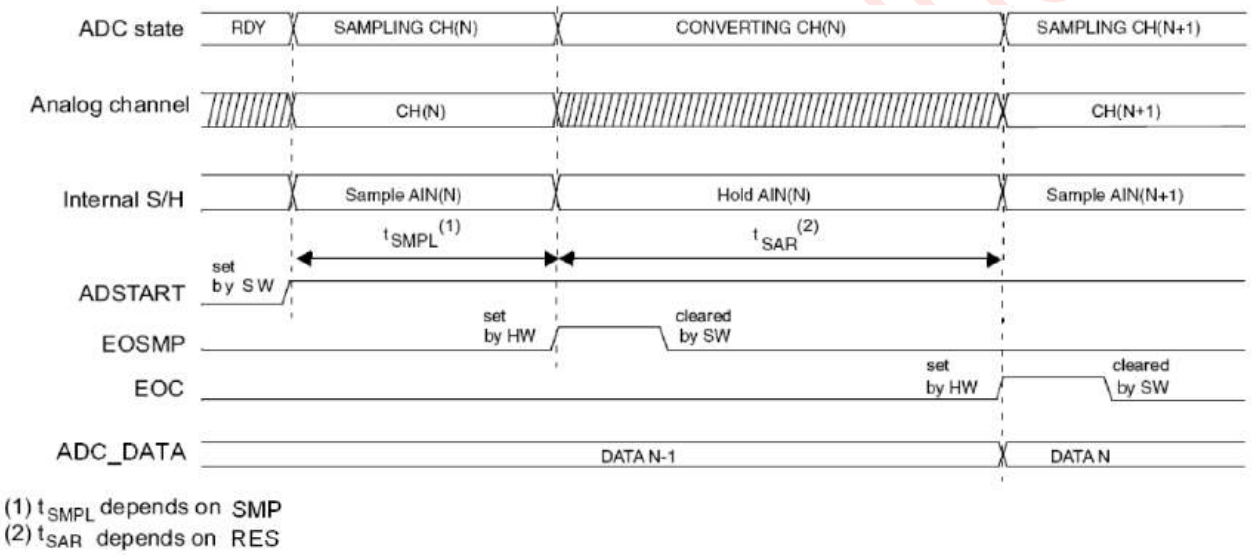


图 31-4：Analog to Digital 转换时间

©Levetop Semiconductor Co., Ltd.

31.3.10. 停止一个持续的转换（ADSTP）

该软件可以通过在 ADC_CR 寄存器中设置 ADSTP = 1 来决定停止任何正在进行的转换。

这将复位 ADC 操作，ADC 将空闲，准备进行新的操作。

当由软件设置 ADSTP 位时，任何正在进行的转换都会被中止，结果也会被丢弃（FIFO 不会用当前的转换进行更新）。扫描序列也会中止和复位（这意味着重新启动 ADC 将重新启动新序列）。

一旦此过程完成，ADSTP 和 ADSTART 都被硬件清除，软件必须等到 ADSTART = 0 才开始新的转换。

提示：QADC_ISR 中的标志不会被 STOP 命令清除，FIFO 中的数据也不会丢失。

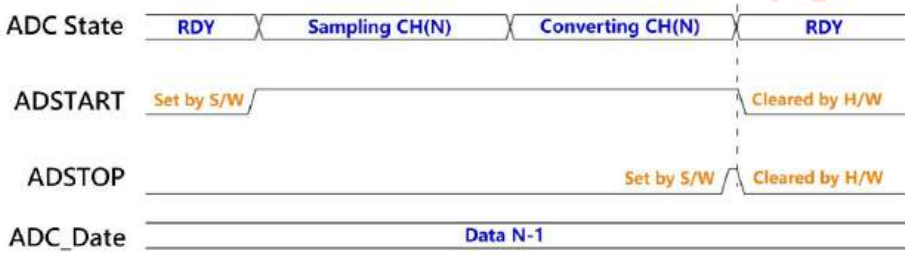


图 31-5：停止一个正在进行的转换

31.4. 在外部触发器和触发器极性上的转换

转换或转换序列可以由软件或外部事件触发。如果三角模式控制位不等于“0”，则外部事件能够触发具有所选极性的转换。一旦软件设置位启动 = 1，触发选择有效。

在正在进行的转换时发生的任何硬件触发器都将被忽略。

如果位启动 = 0，则忽略发生的任何硬件触发器。

表 31-2：提供了三角模式值与触发器极性之间的对应关系。

表 31-2：正在配置触发器的极性。

TRIGMODE[2:0]	Source
3’b000	触发器检测被禁用，软件触发器
3’b001	上升缘检测
3’b010	下降缘检测
3’b011	对上升和下降边缘的检测
3’b100	在高电平电压上的检测
3’b101	在低水平电压上的检测
3’b110	PIT 0 检测
3’b111	在 PWM 0 上检测

©Levetop Semiconductor Co., Ltd.

提示：

只有当 ADC 没有转换时，外部触发器的极性才能被改变（附加启动 = 0）。

触发器控制位用于选择 8 个可能的事件中的哪一个可以触发转换。表 31-3 为常规转换提供了可能的外部触发器。软件源触发器事件可以通过在 ADC_CR 寄存器中设置动态启动位来生成。

表 31-3：配置触发器的极性

TRIGSCR[2:0]	Name	Source
3’b000	TRG0	
3’b001	TRG1	
3’b010	TRG2	
3’b011	TRG3	
3’b100	TRG4	
3’b101	TRG5	
3’b110	TRG6	
3’b111	TRG7	

提示： 只有当 ADC 没有转换时，才能更改触发器选择（已启动 = 0）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

31.4.1. 不连续模式（DISCEN）

通过在 ADC_CFGR1 寄存器中设置 DISCEN 位来启用此模式。

在此模式下 (DISCEN = 1) 中，需要一个硬件或软件触发事件来启动序列中定义的每个转换。相反，如果 DISCEN = 0，则单个硬件或软件触发事件依次启动序列中定义的所有转换。

样例：

- DISCEN = 1，通道要转换为 = 0、3、7、10
  - 第 1 个触发器：通道 0 转换并生成 EOC 事件
  - 第 2 个触发器：转换通道 3 并生成一个 EOC 事件

    外部触发器和触发器极性的转换（三角模式、触发模式）
  - 第 3 个触发器：转换通道 7 并生成一个 EOC 事件
  - 第 4 个触发器：转换通道 10，并生成 EOC 和 EOSEQ 事件。
  - 第 5 个触发器：通道 0 被转换，生成一个 EOC 事件
  - 第 6 个触发器：第 3 通道被转换，并生成一个 EOC 事件

  – .....
- DISCEN = 0，要转换的通道 = 0、3、7、10
  - 第 1 个触发器：完整的序列被转换为：通道 0，然后是通道 3、通道 7 和通道 10。每个转换都会生成一个 EOC 事件，最后一个转换也会生成一个 EOSEQ 事件。
  - 任何后续的触发事件都将重新启动整个序列。

31.4.2. 可编程分辨率（RES）- 快速转换模式

通过降低 ADC 的分辨率，可以获得更快的转换时间 (tSAR)。通过在 ADC_CFGR1 寄存器中编程 RES[1:0]位，可以将分辨率配置为 12、10、8 或 6 位。对于不需要高数据精度的应用程序，较低的分辨率允许更快的转换时间。

提示：

只有在复位 ADEN 位时，才能更改 RES[1:0]位。

转换的结果总是是 13 位宽的，任何未使用的 LSB 位都被读取为零。

较低的分辨率减少了连续近似步骤所需的转换时间。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

31.4.3. 转换结束，采样阶段结束(EOC、EOSMP Flag)

ADC 表示每次转换结束（EOC）事件。

一旦有一个新的转换数据结果可用，ADC 就会在 ADC_ ISR 寄存器中设置 EOC 标志。如果在 ADC_IER 寄存器中设置了 EOCIE 位，则可以产生一个中断。软件可以通过写入 1 或读取 FIFO 来清除 EOC 标志。

ADC 还通过在 ADC_ISR 寄存器中设置 EOSMP 标志来指示采样阶段的结束。软件可以通过写入 1 来清除 EOSMP 标志。如果在 ADC_IER 寄存器中设置了 ADC_IER 位，则可以生成一个中断。

31.4.4. 转换序列的结束(Eoseq Flag)

ADC 通知每个序列（EOSEQ）事件的应用程序。

一旦转换序列的最后一个数据结果在 FIFO 中可用，ADC 就会在 ADC_ISR 寄存器中设置 EOSEQ 标志。如果在 ADC_IER 寄存器中设置了 ADC_IER 位，则可以生成一个中断。软件可以通过写入 1 来清除 EOSEQ 标志。

31.4.5. 时序图范例

本节显示了硬件或软件触发条件下的单个模式和连续模式。

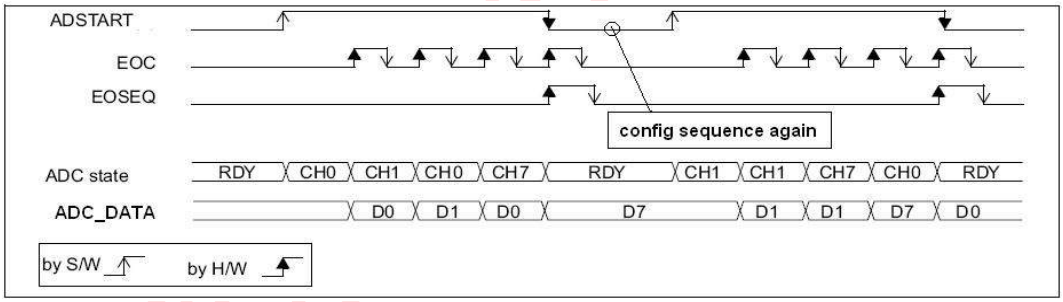


图 31-6：一个序列的单一转换，软件触发器

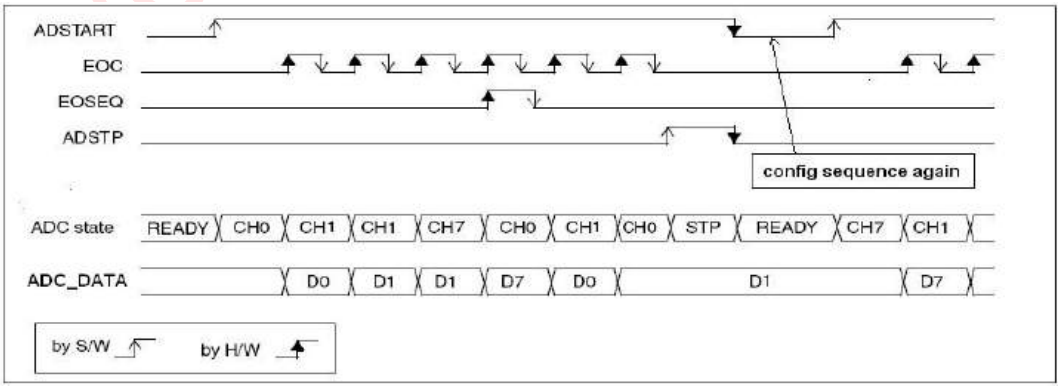


图 31-7：一个序列的连续转换，软件触发器

©Levetop Semiconductor Co., Ltd.

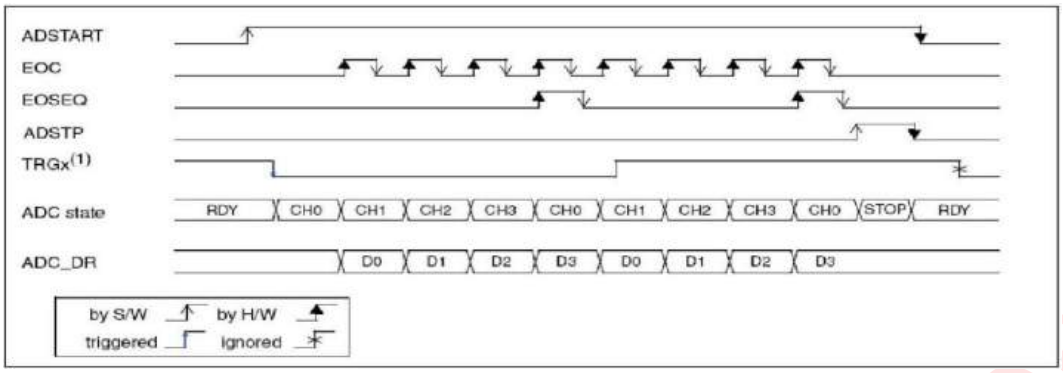


图 31-8：一个序列的单一转换，硬件触发器

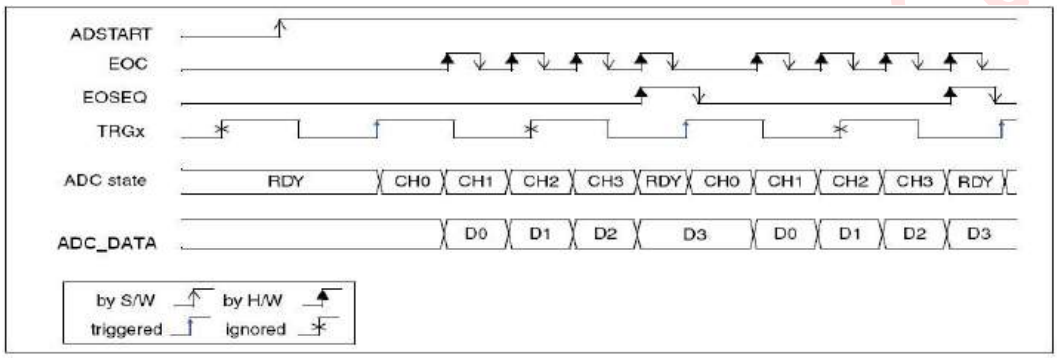


图 31-9：一个序列的连续转换，硬件触发器

©Levetop Semiconductor Co., Ltd.

31.5. 数据管理

31.5.1. 数据 FIFO 和数据对齐(ADC_FIFO，对齐)

在每次转换结束时（当 EOC 事件发生时），转换数据的结果存储在 ADC_FIFO 中，它为 13 位宽 x8 深度。

读出数据的格式取决于所配置的数据对齐和分辨率。

ADC_CFGR1 寄存器中的对齐位选择转换后存储的数据的对齐方式。数据可以是右对齐 (对齐 = 0)或左对齐 (对齐 = 1)，如表 31-4 所示.

表 31-4：数据对齐和分辨率

Align	RES[1:0]	31	30	….	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0x0			….					Data[11:0]											
	0x1			….							Data[9:0]									
	0x2			….									Data[7:0]							
	0x3			….											Data[5:0]					
1	0x0			….	Data[11:0]															
	0x1			….	Data[9:0]															
	0x2			….									Data[7:0]							
	0x3			….									Data[5:0]							

FIFO 支持字节、半字和字的读取，但地址偏移量应始终为 0x4C。
对于不同的数据对齐和分辨率，用户应提示：

- 如果按字读取，则数据格式为数据[31:0]，如表 31-4 所示给…看
- 如果按半字读取，则数据格式为数据[15:0]，如表 31-4 所示给…看
- 如果在数据大于 8 位时按字节读取，则将首先读出高字节，然后应该再次读取低字节。
- 如果在数据不超过 8 位时按字节读取，则数据格式为数据[7:0]，如表 31-4 所示给…看

31.5.2. ADC 超运行（OVR、OVRMOD）

溢出标志（OVR）表示在 FIFO 满之前没有及时读取转换后的数据时的数据溢出事件。

如果在一个新的转换完成时，FULL 标志仍然在“1”，则会在 ADC_ISR 寄存器中设置 OVR 标志。如果在 ADC_IER 寄存器中设置了 OVRIE 位，则可以产生一个中断。

当发生溢出情况时，ADC 将继续运行，并可以继续转换，除非软件决定通过在 ADC_CR 寄存器中设置 ADSTP 位来停止和复位序列。

软件可以通过写入 1 来清除 OVR 标志。

可以通过对 ADC_CFGR1 寄存器中的 OVRMOD 位进行编程，来配置，在溢出事件发生时，数据是否被保留或覆盖：

©Levetop Semiconductor Co., Ltd.

•OVRMOD = 0

-溢出事件保留数据寄存器不被覆盖：维护旧数据，丢弃新的转换。如果 OVR 保持在 1，则可以执行进一步的转换，但结果数据将被丢弃。

•OVRMOD = 1

-数据寄存器被最后的转换结果所覆盖。如果 OVR 保持在 1，则可以执行进一步的转换，并且 FIFO 总是包含来自最新转换的数据。

31.5.3. 管理不使用 DMA 而转换的数据序列

如果转换速度足够慢，则转换顺序可以由软件来处理。在这种情况下，软件可以使用 EOC 标志及其相关的中断来处理每个数据结果。每次转换完成时，在 ADC_ISR 寄存器中设置 EOC 位，并且可以读取 FIFO 寄存器。

软件还可以使用 FIFO 空标志来处理每个数据结果。如果空不是“0”，这意味着 FIFO 有新的数据。

ADC_CFGR1 寄存器中的 OVRMOD 位应该配置为 0，以作为一个错误来管理溢出事件。

31.5.4. 管理转换的数据而不使用 DMA

让 ADC 在一个或多个通道转换后不读取数据可能很有用。在这种情况下，OVRMOD 位必须配置为 1，并且软件应该忽略 OVR 标志。当 OVRMOD = 1 时，溢出事件不会阻止 ADC 继续转换，而 FIFO 始终包含最新的转换数据。

31.5.5. 使用 DMA 管理转换后的数据

一旦 FIFO 中的数据号不是空的，并且设置了位 DMAEN，QADC 将向 DMA 发送请求。这允许将转换后的数据从 FIFO 传输到软件选择的目标位置。

尽管如此，如果由于 DMA 无法及时服务 DMA 传输请求而发生溢出（OVR = 1），ADC 将停止生成 DMA 请求，与新转换对应的数据不会被 DMA 传输。这意味着传输到 RAM 的所有数据都可以被认为是有效的。

根据 OVRMOD 位的配置，数据会被保留或覆盖。

DMA 传输请求被阻止，直到软件清除 OVR 位。

©Levetop Semiconductor Co., Ltd.

31.6. 低功耗功能

31.6.1. 等待模式转换

等待模式转换可用于简化软件，并优化低频应用程序的性能，这可能会有发生 ADC 溢出的风险。当 ADC_CFGR1 寄存器中的 WAIT 位设置为 1 时，只有在 FIFO 未满时才能开始新的转换。

这是一种自动调整 ADC 的速度以适应读取数据的系统的速度的方法。

提示： 在转换进行或读访问之前的等待期间发生的任何硬件触发器将被忽略。

31.6.2. 自动关闭模式（AUTOFF）

ADC 具有一个自动电源管理功能，它被称为自动关闭模式，并通过在 ADC_CFGR1 寄存器中设置自动关闭 = 1 来启用。

当自动关闭 = 1 时，ADC 在不转换时总是关闭，并且当转换启动时自动唤醒（通过软件或硬件触发器）。在启动转换的触发事件和 ADC 的采样时间之间自动插入一个启动时间。一旦转换序列完成，ADC 将被自动禁用。

自动关闭模式可以导致显著降低功耗的应用程序需要相对较少的转换或当转换请求的时间足够远的距离（例如，低频硬件触发）来证明额外的权力和额外的时间用于开关 ADC。

自动关闭模式可以与等待模式转换（WAIT = 1）结合低频。如果 ADC 在等待阶段自动关闭，并在应用程序读取 FIFO 后立即重新启动，那么这种组合可以显著节省电力

31.7. 模拟窗口监视器（AWD）

通过在 ADC_CFGR1 寄存器中设置 AWDEN 位来启用 AWD 模拟看门狗功能。它用于监控一个选定通道或所有启用通道是否保持在配置的电压范围（窗口）内，如图 31-10 所示.

如果由 ADC 转换的模拟电压低于较低的阈值或高于较高的阈值，则设置 AWD 模拟看门狗状态位。这些阈值在 ADC_TR 寄存器中被编程。可以通过在 ADC_IER 寄存器中设置 AWDIE 位来启用中断。

AWD 标志被软件通过写入 1 来清除。

当转换分辨率小于 12 位的数据时（根据位 RES[1:0]），必须清除编程阈值的 LSB，因为总是对完整的 12 位原始转换数据（左对齐）进行内部比较。

表 31-5 显示如何在 theADC_CFGR1 寄存器中配置 AWDSGL 和 AWDEN 位，以在一个或多个通道上启用模拟看门狗。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

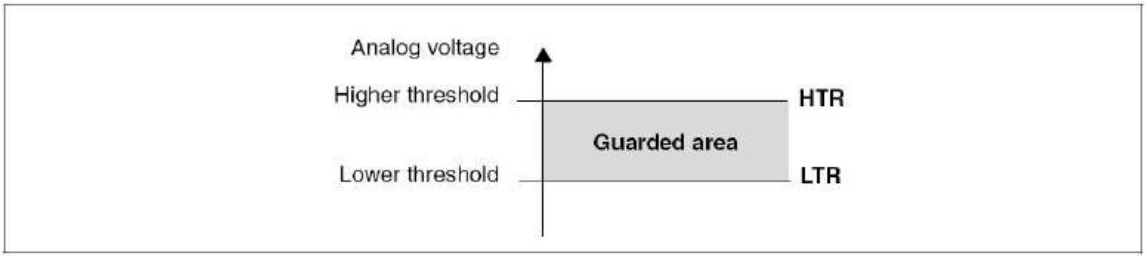


图 31-10：模拟式监视器的监视区域

表 31-5：模拟监控器通道选择

Channels Guarded by the Analog Watchdog	AWDSGL Bit	AWDEN Bit
没有一个	X	0
所有通道	0	1
单个(1)通道	1	1

注(1)：由 AWDCH 选择

31.8. 温度传感器

温度传感器内部连接到 ADC1_IN15 输入通道，用于将传感器的输出电压转换为数字值。

31.9. ADC 中断

以下任一事件都可以产生中断：

- 当 ADC 准备就绪时，ADC 通电（ADRDY 标志）
- 任何转换的结束(EOC 标志)
- 转换序列的结束(EOSEQ 标志)
- 当发生模拟监视器检测时(AWD 标志)
- 当采样阶段结束时（EOSMP Flag）
- 当发生数据溢出(OVR 标志）时，单独的中断启用位可提供灵活性。

表 31-6：ADC 中断

Interrupt Event	Event Flag	Enable Control Bit
ADC Ready	ADRDY	ADRDYIE
End of Conversion	EOC	1EOCIE
End of Sequence of Conversions	EOSEQ	1EOSEQIE
Analog Watchdog Status bit is Set	AWD	AWDIE
End of Sampling Phase	EOSMP	EOSMPIE
Overrun	OVR	OVRIE

©Levetop Semiconductor Co., Ltd.

31.10. 内存映射和寄存器

ADC 模块内存映射如表 31-7 所示.ADC 基地址为 0x4011_0000。本节描述了 ADC 的内存映射和寄存器结构。

31.10.1. 内存映射

见表 31-7 关于 QADC 内存映射的描述。

表 31-7：ADC 模块内存映射

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ADC_ISR																									AWD	Empty	FULL	OVR	EOSEQ	EOC	EOSMP	ADRDY
0x04	ADC_IER																									AWDIE			OVRIE	EOSEQIE	EOCIE	EOSMPIE	ADRDYIE
0x08	ADC_CR																													ADSTP	ADSTART	ADDIS	ADEN
0x0C	ADC_CFGR1	DIFF	OVRMOD				SEQLEN[2:0]			DISCEN	AUTOFF	WAIT	CONT		TRIGSCR[2:0]					TRIGMODE[2:0]			ALIGN	RES[1:0]									DMAEN
0x10	ADC_CFGR2																						QPR[3:0]			STCNT[7:0]							
0x14	ADC_SMPR																									SMP[7:0]							
0x18	ADC_WDG																									AWDEN	AWDSGL			AWDCH[3:0]			
0x1C	ADC_TR					HT[11:0]																LT[11:0]											
0x2C	ADC_CHSELR1					CCW3[3:0]								CCW2[3:0]								CCW1[3:0]								CCW0[3:0]			
0x30	ADC_CHSELR2					CCW7[3:0]								CCW6[3:0]								CCW5[3:0]								CCW4[3:0]			
0x4C	ADC_FIFO																	DATA[15:0]															

提示： 1.所有的寄存器都是 CPU 监视器或用户模式是可访问的。

2.灰色位是保留的，并且必须保持在复位值。

©Levetop Semiconductor Co., Ltd.

31.10.2. 寄存器描述

31.10.2.1. ADC 中断和状态寄存器（ADC_ISR）

地址：ADC_BASEADDR+0x0000_0000

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	AWD	EMPTY	FULL	OVR	EOSEQ	EOC	EOSMP	ADRDY
写:	w1c			w1c	w1c	w1c	w1c	w1c
复位:	0	1	0	0	0	0	0	0

	= 写入没有影响，访问终止，没有传输错误异常。
w1c	= 写 1 到该位将清除它

图 31-11：ADC 中断和状态寄存器（ADC_ISR）

Read: Anytime

Write: Anytime

AWD — Analog watchdog flag

当转换后的电压超过 ADC_TR 寄存器中编程的值时，该位由硬件设置。它可以通过软件写入 1 来清除。

1 = 模拟监视事件

0 = 没有发生模拟看门狗事件（或标志事件已被软件确认和清除）

EMPTY — FIFO empty status

当 FIFO 为空时，此位由硬件设置。当 FIFO 不为空时，它将由硬件清除。

1 = FIFO 为空

0 = FIFO 不是空的

©Levetop Semiconductor Co., Ltd.

FULL — FIFO full status

当 FIFO 已满时，此位将由硬件设置。当 FIFO 未满时，它将由硬件清除。

1 = FIFO 已满

0 = FIFO 未满

OVR — ADC overrun

这个位是在发生溢出时由硬件设置的，这意味着在已经设置了满标志时，一个新的转换已经完成。它可以通过软件写入 1 来清除。

1 = 已发生溢出

0 = 没有发生溢出（或标志事件已被软件确认和清除）

EOSEQ — End of sequence flag

这个位是由硬件在一个序列的转换结束时设置的。它可以通过软件写入 1 来清除。

1 = 转换序列已完成

0 = 转换序列未完成（或标志事件已被软件确认和清除）

EOC — End of conversion flag

当 ADC_FIFO 寄存器中有一个新的数据结果时，由硬件在通道的每次转换结束时设置这个位。通过软件写入 1 或读取 ADC_FIFO 寄存器来清除。

1 = 通道转换完成

0 = 通道转换未完成（或标志事件已被软件确认和清除）

EOSMP — End of sampling flag

这个位是在采样阶段结束时由硬件设置的。

1 = 采样阶段结束

0 = 未在采样阶段结束时（或标志事件已被软件确认和清除）

ADRDY — ADC ready

这个位是在启用 ADC 之后（位 ADEN = 1）和当 ADC 达到它准备好接受转换请求的状态时由硬件设置的。它可以通过软件写入 1 来清除。

1 = ADC 已准备好开始转换

0 = ADC 尚未准备好开始转换（或标志事件已被软件确认和清除）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

31.10.2.2. ADC 中断启用寄存器（ADC_IER）

地址：ADC_BASEADDR+0x0000_0004

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	AWDIE	0	0	OVRIE	EOSEQIE	EOCIE	EOSMPIE	ADRDYIE
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 31-12：ADC 中断启用寄存器（ADC_IER）

Read: Anytime

Write: Before ADC start

AWDIE — Analog watchdog interrupt enable
此位由软件设置和清除，以启用/禁用模拟看门狗中断。
1 = 已启用模拟看门狗中断
0 = 模拟看门狗中断已被禁用

OVRIE — Overrun interrupt enable
软件已设置并清除此位，以启用/禁用溢出中断。
1 = 溢出中断。当设置了 OVR 位时，就会产生一个中断。
0 = 溢出中断

EOSEQIE — End of conversion sequence interrupt enable
该位由软件设置和清除，以启用/禁用转换中断序列的结束。
1 = EOSEQ 中断。当设置了 EOSEQ 位时，就会产生一个中断。
0 = EOSEQ 中断

©Levetop Semiconductor Co., Ltd.

EOCIE — End of conversion interrupt enable

此位由软件设置和清除，以启用/禁用转换中断的结束。

1 = EOC 中断。当设置了 EOC 位时，就会产生一个中断。

0 = EOC 中断

EOSMPIE — End of sampling flag interrupt enable

该位由软件设置和清除，以启用/禁用采样阶段中断的结束。

1 = EOSMP 中断。当设置了 EOSMP 位时，就会产生一个中断。

0 = EOSMP 中断已被禁用。

ADRDYIE — ADC ready interrupt enable

此位被软件设置和清除，以启用/禁用 ADC 准备中断。

1 = ADRDY 中断。当设置了 ADRDY 位时，就会产生一个中断。

0 = ADRDY 中断已被禁用。

31.10.2.3. ADC 控制寄存器（ADC_CR）

地址：ADC_BASEADDR+0x0000_0008

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	1	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0				ADSTP	ADSTART	ADDIS	ADEN
写:					rs	rs	rs	rs
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

rs = 软件可以读取和设置，写 0 没有影响

图 31-13：ADC 控制寄存器（ADC_CR）

Read: Anytime

Write: See each bit description

©Levetop Semiconductor Co., Ltd.

ADSTP — ADC stop conversion command

这个位是由软件设置为停止和丢弃一个正在进行的转换（ADSTP 命令）。当转换被有效地丢弃并且 ADC 准备接受新的开始转换命令时，硬件清除。

1 = 写入 1 以停止 ADC。读取 1 表示一个 ADSTP 命令正在进行中。

0 = 没有 ADC 停止转换命令正在进行中

提示： 软件只允许在 ADSTP 启动 = 1 和 ADDIS = 0 时设置 ADSTP（ADC 已启用，可能正在转换，没有等待的请求来禁用 ADSTP）

ADSTART — ADC start conversion command

这个位是由软件设置来启动 ADC 转换的。根据三叉模式配置位的不同，转换可以立即开始（软件触发配置），或者一旦硬件触发事件发生（硬件触发配置）。

通过硬件进行清除：

- 在选择软件触发器时，单一转换模式：在转换顺序结束（EOSEQ）标志。
- 选择软件触发器时，处于停止转换模式：显示转换结束（EOC）标志。
- 在所有情况下：在执行 ADSTP 命令后，同时，ADSTP 位被硬件清除。

1 = 写入 1 以启动 ADC。读取 1 表示 ADC 正在运行，并且可能正在转换。

0 = 没有正在进行的 ADC 转换。

提示： 只有当 ADEN = 1 和 ADDIS = 0（启用 ADC 且没有禁用 ADC 的待定请求）时，软件才允许设置已启动

ADDIS — ADC disable command

此位由软件设置为禁用 ADC（ADDIS 命令），并将其置于断电状态(OFF 状态）。一旦 ADC 被有效地禁用，硬件就会清除它（此时 ADEN 也会被硬件清除）。

1 = 写入 1 以禁用 ADC。读取 1 表示正在执行一个 ADDIS 命令。

0 = 没有 ADDIS 命令正在进行中

提示： 只有当 ADEN = 1 和 ADSTART = 0 时，软件才允许设置 addis（这将确保不进行转换）

ADEN — ADC enable command

此位由软件设置为启用 ADC。一旦设置了 ADRDY 标志，ADC 将有效地准备好操作。执行 ADDIS 命令后，禁用 ADC 后，硬件清除。

1 = Write 1 以启用 ADC。

0 = ADC 已禁用（关闭状态）。

提示： 只有当 ADC_CR 寄存器的所有位都为 0（ADSTP = 0、ADSTART = 0、ADDIS = 0 和 ADEN = 0）时，软件才允许设置 ADEN

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

31.10.2.4. ADC 配置寄存器 1（ADC_CFGR1）

地址：ADC_BASEADDR+0x0000_000C

	31	30	29	28	27	26	25	24
读:	DIFF	OVRMOD	0			SEQ_LEN[2:0]		
写:								
复位:	0	0	0	0	0	1	1	1

	23	22	21	20	19	18	17	16
读:	DISCEN	AUTOFF	WAIT	CONT	0	TRIGSCR[2:0]		
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0		TRIGMODE[2:0]			ALIGN	RES[1:0]	
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0							DMAEN
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 31-14：ADC 配置寄存器 1（ADC_CFGR1）

Read: Anytime

Write: Before ADC start

DIFF — Select differential-input

此位确定输入是单端输入还是差分输入。

1 = 模拟输入进行差分采样。

0 = 模拟输入为单采样

OVRMOD — Overrun management mode

此位由软件设置和清除，并配置数据溢出的方式。

1 = 当检测到溢出时，ADC_DR 寄存器将被最后一个转换结果覆盖。

0 = 当检测到溢出时，ADC_DR 寄存器将与旧数据一起保存。

SEQ_LEN[2:0] — Sequence length

这些位定义了序列的长度。序列长度为 = SEQ_LEN+1。例如，SEQ_LEN = 7 表示序列长度为 8；SEQ_LEN = 0 表示序列长度为 1。

©Levetop Semiconductor Co., Ltd.

DISCEN — Discontinuous mode

此位通过软件设置和清除，以启用/禁用不连续模式。

1 = 不连续模式

0 = 不连续模式

AUTOFF — Auto-off mode

此位通过软件设置和清除，以启用/禁用自动关闭模式。

1 = 已启用了自动关闭模式

0 = 自动关闭模式

WAIT — Wait conversion mode

此位通过软件设置并清除，以启用/禁用等待转换模式。

1 = 等待转换模式开启

0 = 等待转换模式关闭

CONT — Single / continuous conversion mode

此位元已由软件设置和清除。如果设置，转换持续进行，直到被清除

1 = 连续转换模式

0 = 单次转换模式

TRIGSCR[2:0] — External trigger source

这些位用于选择 8 个可能的事件中哪些可以触发转换。见表 31-3.

TRIGMODE[2:0] — Trigger mode select

这些位用于选择软件触发模式或外部触发极性，见表 31-2.

ALIGN — Data alignment

这个位被软件设置和清除，以选择右或左对齐。见表 31-4.

0 = 右对齐

1 = 左对齐

RES[1:0] — Data resolution

这些位是由软件编写来选择转换的分辨率。见表 31-4.

DMAEN — Direct memory access enable

该位由软件设置和清除，以启用生成 DMA 请求。这允许使用 DMA 控制器来自动管理转换后的数据。

1 = DMA 启用

0 = DMA 已被禁用

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

31.10.2.5. ADC 配置寄存器 2（ADC_CFGR2）

地址：ADC_BASEADDR+0x0000_0010

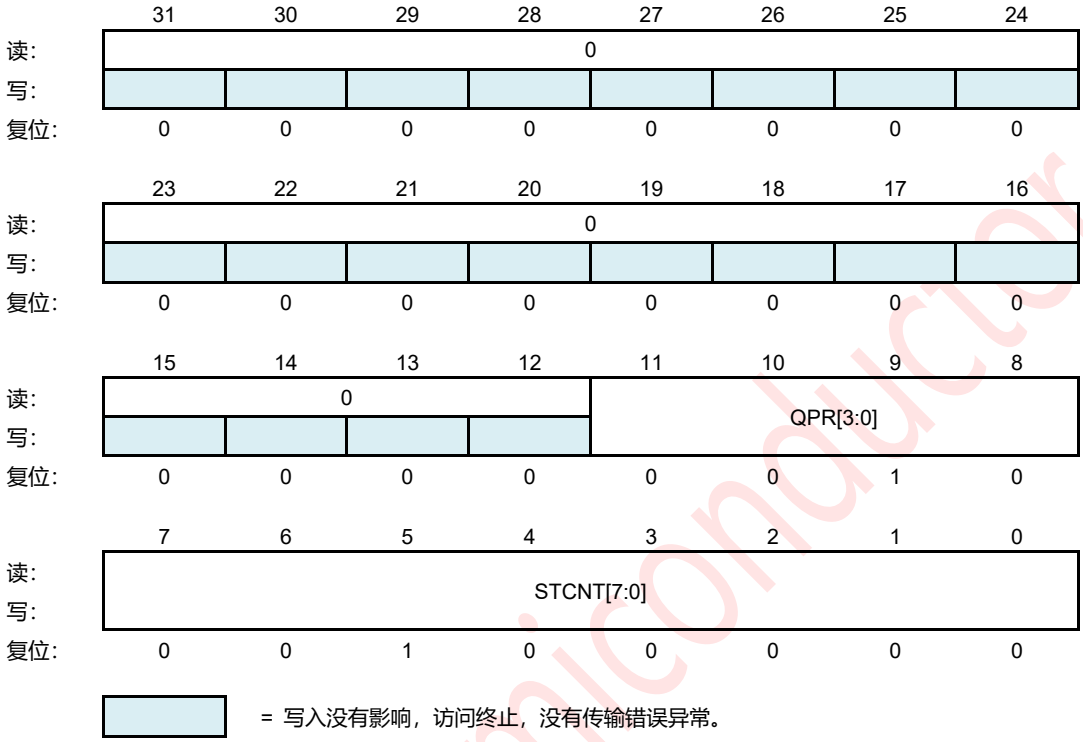


图 31-15：ADC 配置寄存器 2（ADC_CFGR2）

Read: Anytime
Write: Before ADC enable

QPR[3:0] — Prescaler Clock Divider Bits
这些位选择系统时钟除数来生成 QADC 时钟如下：

FQCLK = Fsys_QCLK / (QPR[3:0] + 1)

where:

0 < QPR[3:0] <= 15 and the value 1 is not allowed.

STCNT[7:0] — ADC startup counter bits
ADC 需要 tSTAB（~2us）才能开始精确转换。这个时间是通过计算 QCLK 循环来计算的，直到内部计数器到达 STCNT[7:0]。所以用户应该在 ADC 启用之前设置这些位。例如，如果 QCLK = 为 16MHz，则用户应该设置 STCNT[7:0] = 2000/（1000/16）= 32。

©Levetop Semiconductor Co., Ltd.

31.10.2.6. ADC 采样时间寄存器（ADC_SMPR）

地址：ADC_BASEADDR+0x0000_0014

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	SMP[7:0]							
写:								
复位:	0	0	0	0	0	0	1	0

= 写入没有影响，访问终止，没有传输错误异常。

图 31-16：ADC 采样时间寄存器（ADC_SMPR）

Read: Anytime

Write: Before ADC start

SMP[7:0] — Sampling time selection

这些位是由软件编写，以选择适用于所有通道的采样时间。采样时间计算为（SMP[7:0]+2）QCLKs

例如：SMP[7:0] = 0x2 表示示例时间为 4 个 QCLKs

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

31.10.2.7. ADC 看门狗寄存器（ADC_WDG）

地址：ADC_BASEADDR+0x0000_0018

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	AWDEN	AWDSGL	0	0	AWDCH[3:0]			
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 31-17：ADC 看门狗寄存器（ADC_WDG）

Read: Anytime

Write: Before ADC start

AWDEN — Analog watchdog enable
此位元已由软件设置和清除。

1 = 已启用模拟看门狗

0 = 模拟看门器已禁用

AWDSGL — Enable the watchdog on a single channel or on all channels
此位由软件设置和清除，以启用由 AWDCH[4:0]位识别的通道上或所有通道上的模拟看门狗

1 = 模拟看门狗已在单个通道上启用

0 = 在所有通道上都启用了模拟看门狗

AWDCH[3:0] — Analog watchdog channel selection
这些位是由软件设置和清除。它们选择要由模拟看门狗保护的输入通道。

- 0000：由 AWD 监控的 ADC 模拟输入通道 0
- 0001：ADC 模拟输入通道 1
- .....
- .....

©Levetop Semiconductor Co., Ltd.

- .....
- 0111：ADC 模拟输入通道 7
- 1111：由 AWD 监控的温度传感器
- 其他值：已保留，不能使用

31.10.2.8. ADC 看门狗阈值寄存器（ADC_TR）

地址：ADC_BASEADDR+0x0000_001C

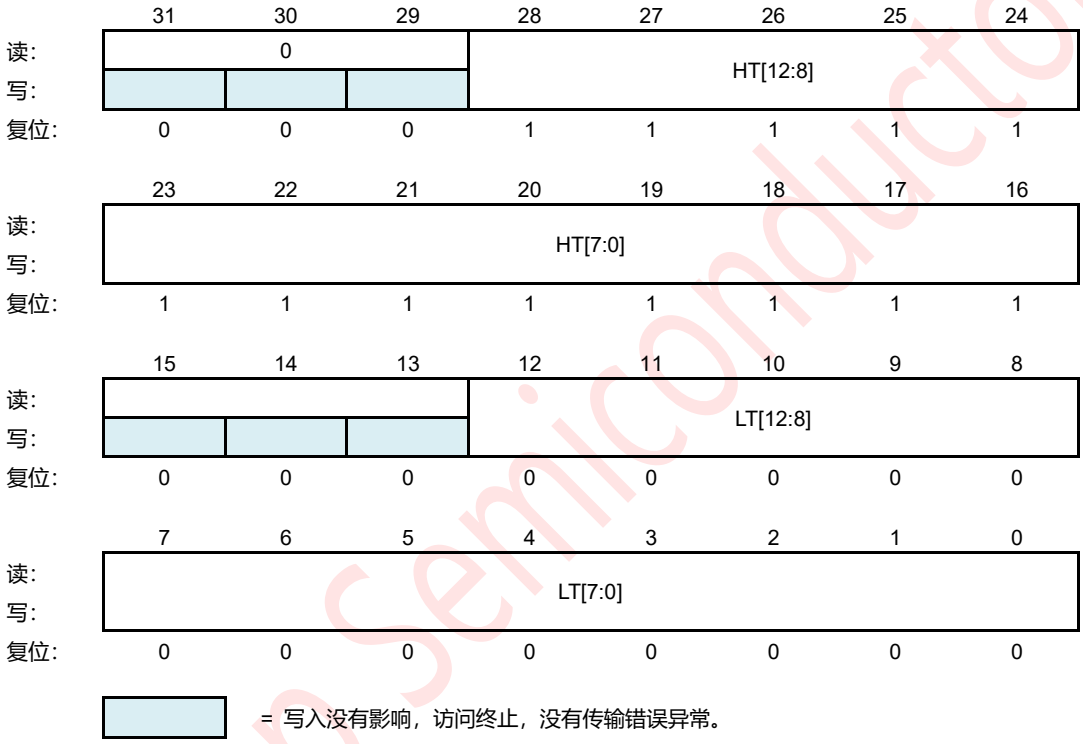


图 31-18：ADC 看门狗阈值寄存器（ADC_TR）

Read: Anytime

Write: Before ADC start

HT[12:0] — Analog watchdog higher threshold

这些位是由软件编写的，以定义模拟看门狗的更高的阈值。

LT[12:0] — Analog watchdog lower threshold

这些位是由软件编写来定义模拟看门狗的下阈值。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

31.10.2.9. ADC 通道选择寄存器（ADC_CHSELR）

地址：ADC_BASEADDR+0x0000_002C

	31	30	29	28	27	26	25	24
读:	0				CCW3[3:0]			
写:								
复位:	0	0	0	0	0	0	0	0

	23	19	18	17	16	18	17	16
读:	0				CCW2[3:0]			
写:								
复位:	0	0	0	0	0	0	0	0

	15	11	10	9	8	10	9	8
读:	0				CCW1[3:0]			
写:								
复位:	0	0	0	0	0	0	0	0

	7	3	2	1	0	2	1	0
读:	0				CCW0[3:0]			
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 31-19：ADC 通道选择寄存器 1（ADC_CHSELR1）

地址：ADC_BASEADDR+0x0000_0030

	31	30	29	28	27	26	25	24
读:	0				CCW7[3:0]			
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0				CCW6[3:0]			
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	0				CCW5[3:0]			
写:								
复位:	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
读:	0				CCW4[3:0]			
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 31-20：ADC 通道选择寄存器 2（ADC_CHSELR2）

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

Read: Anytime

Write: Before ADC start

CCWx[3:0] — The number x conversion select channel, refer to **Table 31-1**..

31.10.2.10. ADC FIFO 访问寄存器（ADC_FIFO）

地址：ADC_BASEADDR+0x0000_004C

	31	30	29	28	27	26	25	24
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	23	22	21	20	19	18	17	16
读:	0							
写:								
复位:	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8
读:	DATA[15:8]							
写:								
复位:	0	0	0	0	0	0	0	0

	3	6	5	4	3	2	1	0
读:	DATA[7:0]							
写:								
复位:	0	0	0	0	0	0	0	0

= 写入没有影响，访问终止，没有传输错误异常。

图 31-21：ADC FIFO 访问寄存器（ADC_FIFO）

Read: Anytime

Write: Never

DATA[15:0] — Converted data

Refer to **Table 31-4.**

©Levetop Semiconductor Co., Ltd.

32.电气特性

本章提供了 LT168 的电气特性参数和极限值。

32.1. 极限参数

表 32-1：电气极限参数表

符号	参数描述	参数范围	单位
V_{DD33}	电源电压	-0.5 ~ 4.6	V
V_{IN}	逻辑输入电压	-0.5 ~ VDD33+0.5	V
V_{OUT}	逻辑输出电压	-0.5 ~ VDD33+0.5	V
P_D	最大功耗	≦300	mW
T_{OPR}	工作温度范围	-40 ~ 105	℃
T_{JT}	工作结温范围	-40 ~ 125	℃
T_{ST}	储存温度范围	-55 ~ 150	℃
T_{SOL}	最高焊接温度	260	℃

提示：最大极限值是指超出该工作范围时，芯片有可能损坏。推荐工作范围是指在该范围内，器件功能正常，但并不完全保证满足个别性能指针。电气参数定义了器件在工作范围内并且在保证特定性能指针的测试条件下的直流和交流电参数规范。对于未给定上下限值的参数，本规范不予保证其精度，但其典型值合理反映了器件性能。

32.2. 静电放电（ESD）保护

表 32-2：静电排放保护特性

ESD 测试	符号	最高	单位	参考标准
Human Body Model	HBM	4000	V	ANSI/ESDA/JEDEC JS-001-2017
Machine Model	MM	200	V	JEDEC JESD22-A115C-2010
Charged Device Model	CDM	800	V	ANSI/ESDA/JEDEC JS-002-2022
Latch Up	LU	200	mA	JEDEC JESD78F.01-2022, @105℃

提示：在进行人工焊接时，建议对人员和设备进行防静电处理。如适当的温湿度环境、焊接设备接地、防静电工作台、焊接人员佩戴防静电腕带等。

©Levetop Semiconductor Co., Ltd.

32.3. DC 电气参数

表 32-3：IO 电气参数表(3.3V)

项目	符号	最小值	典型值	最大值	单位
工作电压	VDD33	2.97	3.3	3.63	V
输入高电位	V_{IH}	2.0	-	VDD33+0.3	V
输入低电位	V_{IL}	-0.3	-	0.8	V
输出高电位	V_{OH}	2.4	-	VDD33	V
输出低电位	V_{OL}	0	-	0.4	V
输入漏电流	I_{IN}	-	-	1	uA
上拉电阻	RPU	33	41	62	KΩ
下拉电阻	RPD	33	42	68	KΩ

表 32-4：电源特性

项目	符号	最小值	典型值	最大值	单位
芯片电源	VDD33	2.97	3.3	3.63	V
ADC 工作电压	AVDD	2.97	3.3	3.63	V
内核工作电压（LDO O/P)	VDD12	1.1	1.2	1.3	V
RTC 工作电压	VBAT	2.7	3.3	3.6	V

表 32-5：功率消耗特性

项目	条件	最小值	典型值	最大值	单位
运行模式 @200MHz	LDO：正常，时钟源：开启，系统时钟：开启，ADC 时钟：开启	-	48.0	-	mA
睡眠模式 00	LDO：正常，时钟源：开启，系统时钟：关闭，ADC 时钟：开启	-	35	-	mA
睡眠模式 01	LDO：正常，时钟源：开启，系统时钟：关闭，ADC 时钟：关闭	-	34	-	mA
睡眠模式 10	LDO：正常，时钟源：关闭，系统时钟：关闭，ADC 时钟：关闭	-	230	-	uA
睡眠模式 11	LDO：STB，时钟来源：关闭，系统时钟：关闭，ADC 时钟：关闭	-	150	-	uA
RTC 操作模式	V_{BAT} = 3.3V, I_{RTC}	-	3.0	-	uA

提示：参见表 9-5 “睡眠模式下的睡眠操作控制位”。

©Levetop Semiconductor Co., Ltd.

表 32-6：复位与唤醒时间特性

项目	符号	最小值	典型值	最大值	单位
复位时间	T_{RST}	70	100	-	μS
复位后到用户程序时间	T_{PRG}	-	15	-	mS
睡眠模式 00/01 唤醒时间	T_{SLP00}	-	105	-	μS
睡眠模式 10/11 唤醒时间	T_{SLP02}	-	105	-	μS

提示：参见表 9-5 “睡眠模式下的睡眠操作控制位”。

32.4. VDD 上电时序

LT168 使用时必须注意 VDD 的上电（Power Up）要求，在上电时 VDD33 必须在低电压（V_L）维持至少 400ms 以上的等待时间（T_{WAIT}），同时 VDD33 由 V_L 到正常工作电压的上升时间（T_R）也不能太长，必须在 500ms 内达正常的工作电压范围，否则容易导致 LT168 内部的 MCU 无法正常启动。

表 32-7：VDD 上电（Power Up）特性

参数	符号	条件与说明	Min.	Nom.	Max.	单位
Rise Time	T_R	输入电压由 V_L 到正常工作电压的上升时间	-	-	500	ms
Wait Time	T_{WAIT}	Power On 之前 VL 的保留时间	400	-	-	ms
VDD 输入电压	V_L	at T = T_1 on Pin VDD33 (Power On 前的输入电压)	-	-	200	mV
VDD 输入电压	V_H	正常工作电压	2.97	3.3	3.63	V

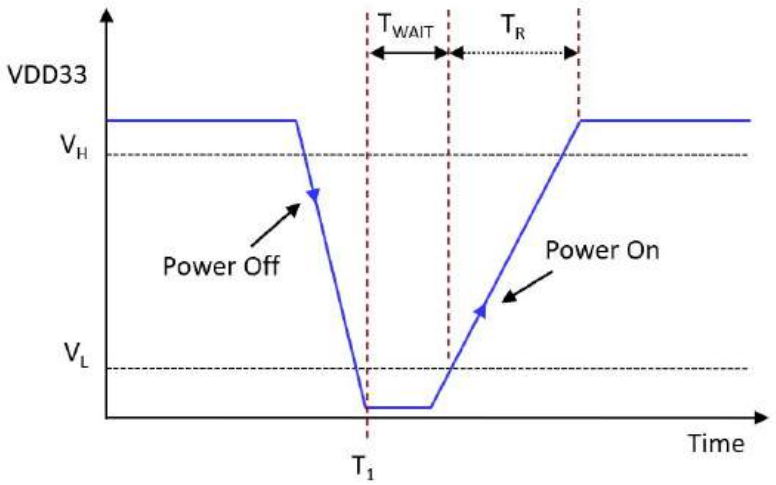


图 32-1: VDD 上电要求时序图

©Levetop Semiconductor Co., Ltd.

33.应用电路

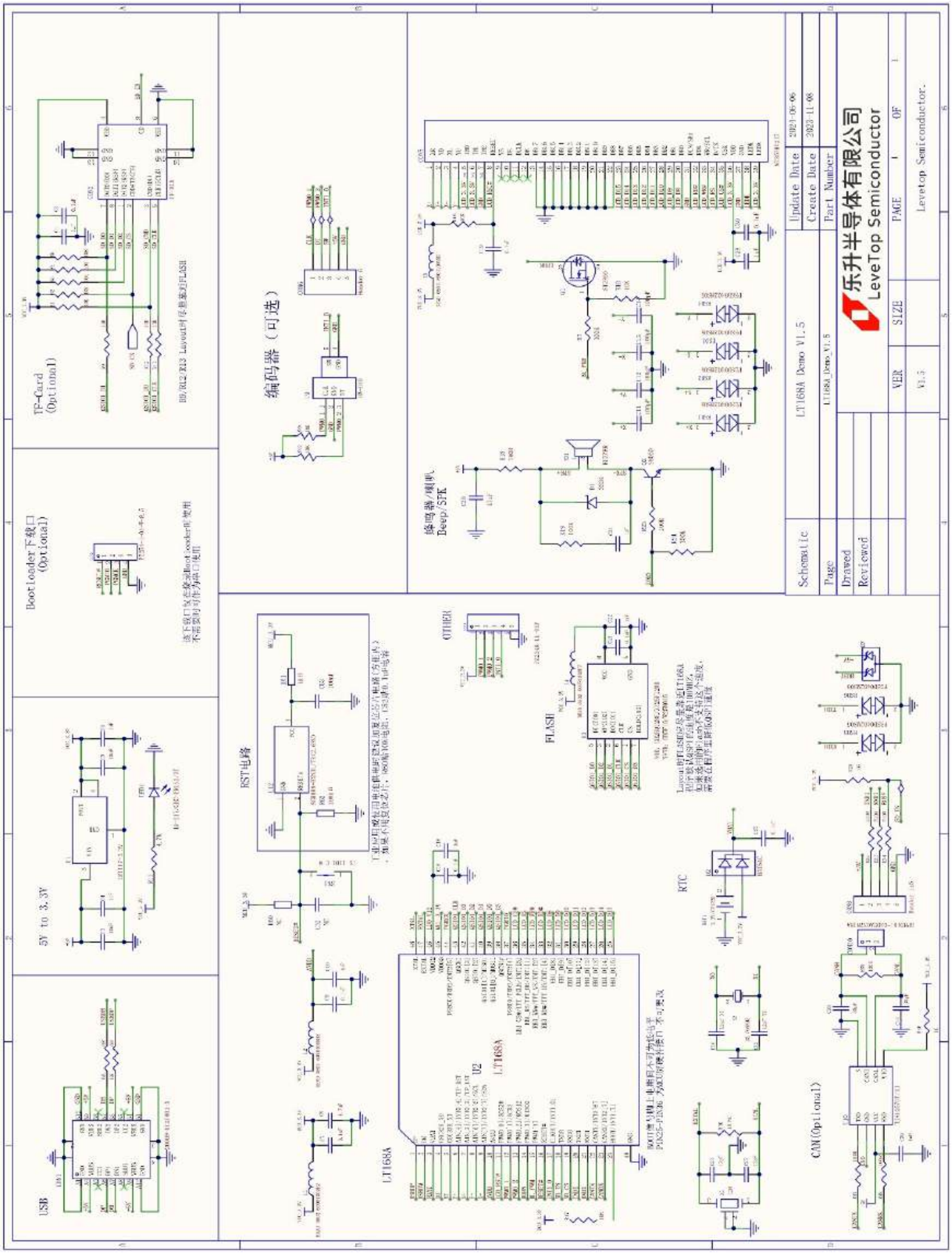


图 33-1：应用电路-8 位的 LT168AMCU 面板

©Levetop Semiconductor Co., Ltd.

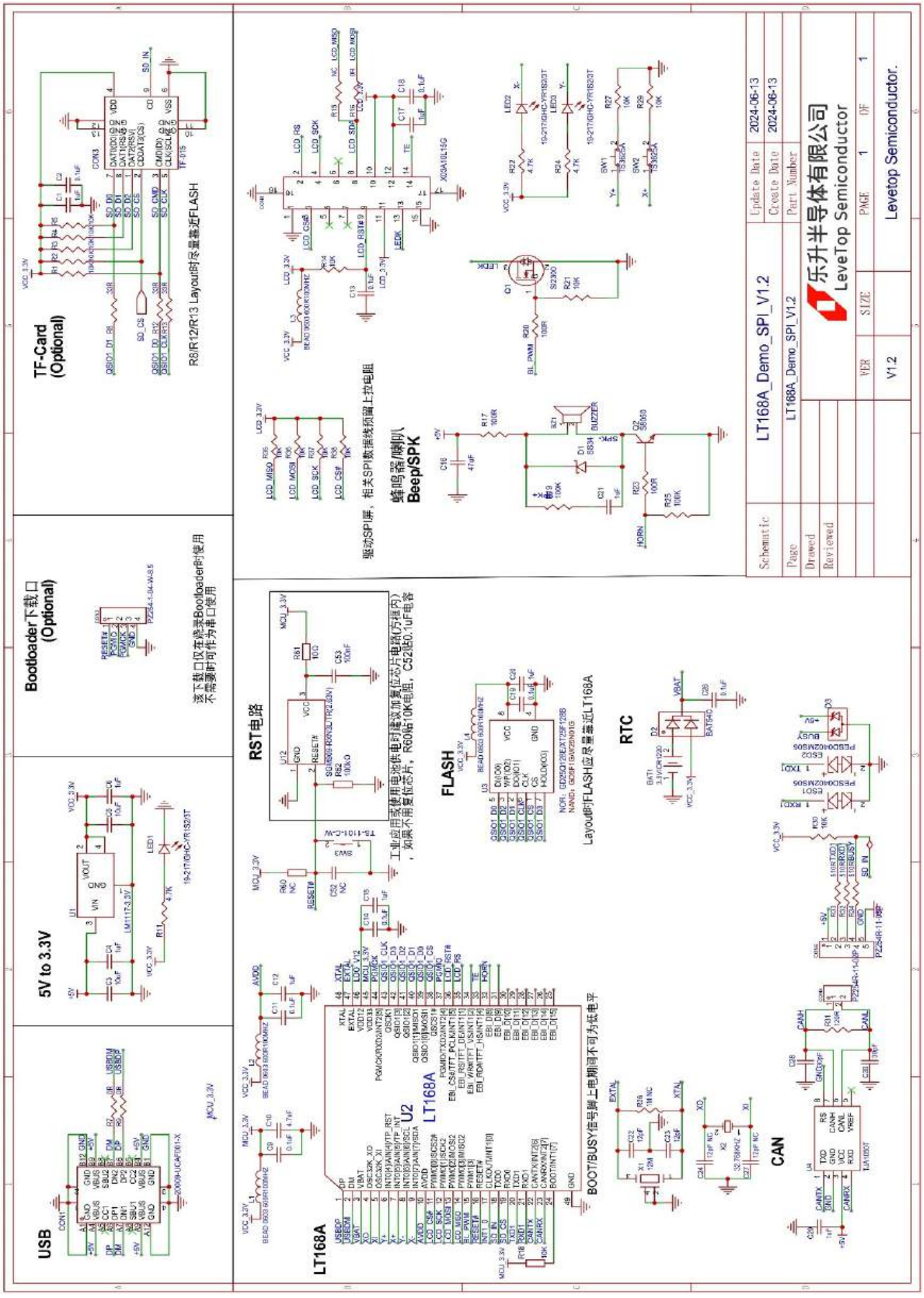


图 33-2：应用电路-SPI 面板的 LT168A

©Levetop Semiconductor Co., Ltd.

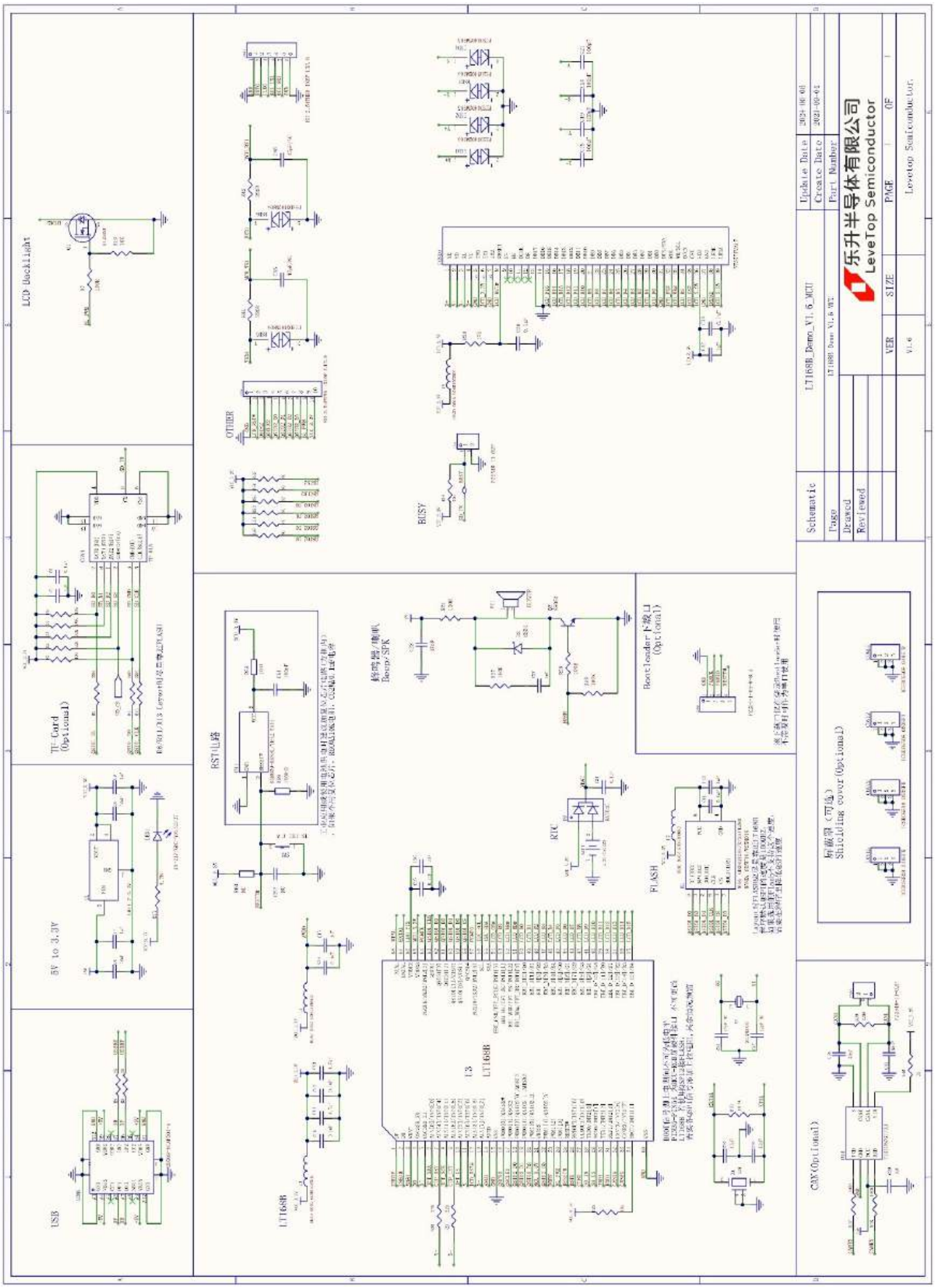


图 33-3：应用电路-16 位的 LT168BMCU 面板

©Levetop Semiconductor Co., Ltd.

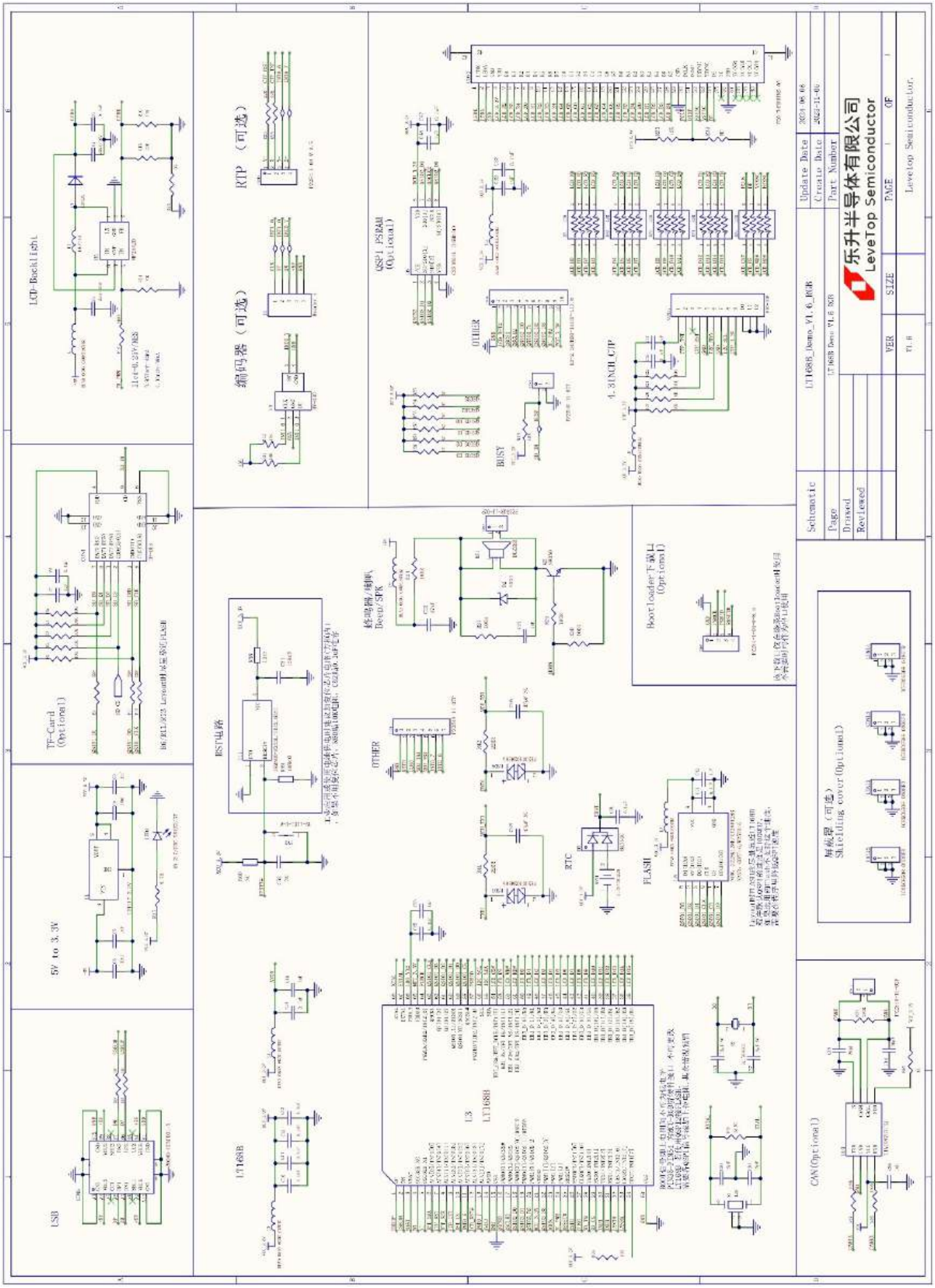


图 33-4：应用电路-RGBTFT 面板的 LT168B

©Levetop Semiconductor Co., Ltd.

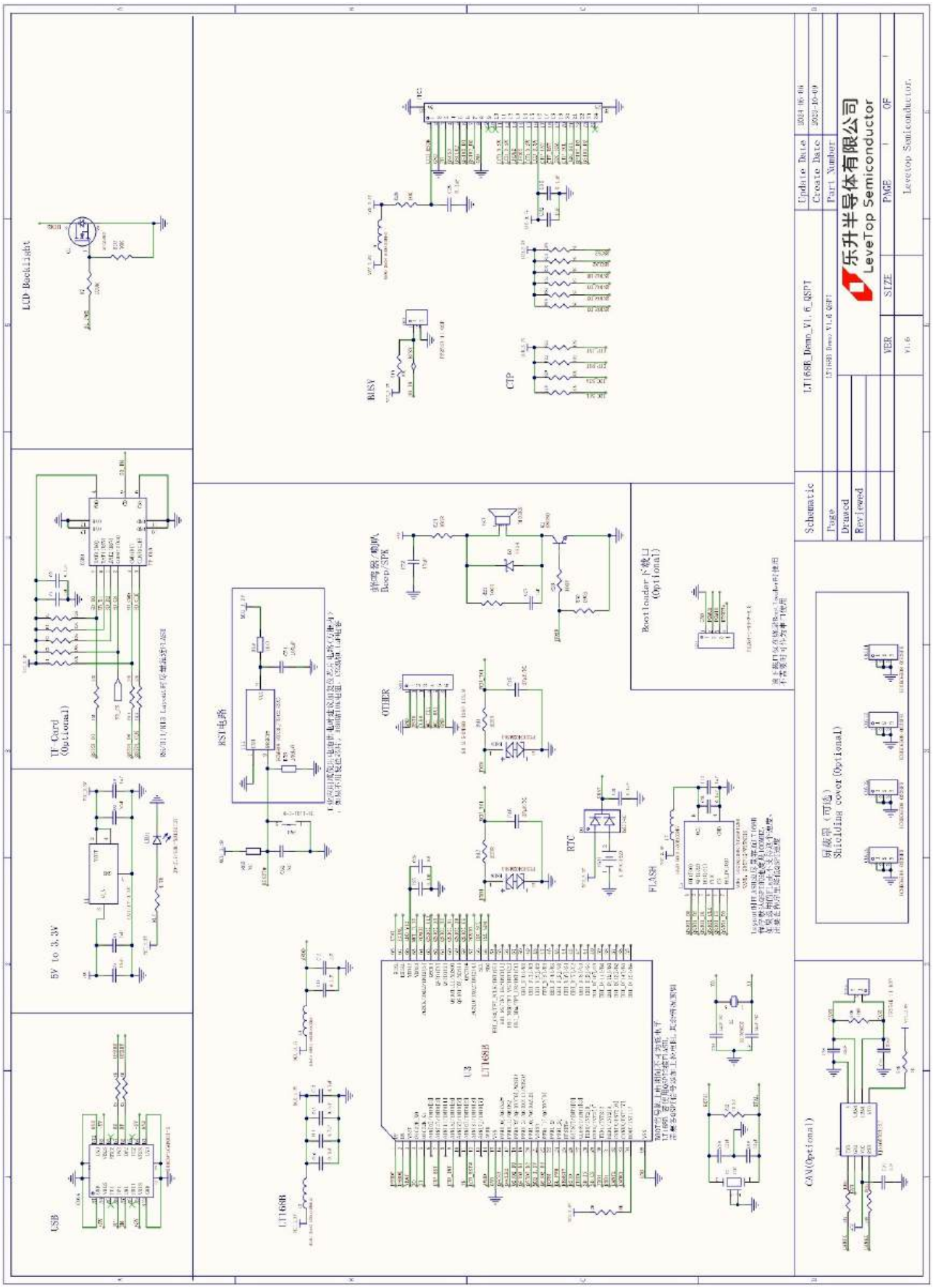


图 33-5：应用电路-LT168B

©Levetop Semiconductor Co., Ltd.

34.封装讯息

34.1. LT168A (QFN-48pin)

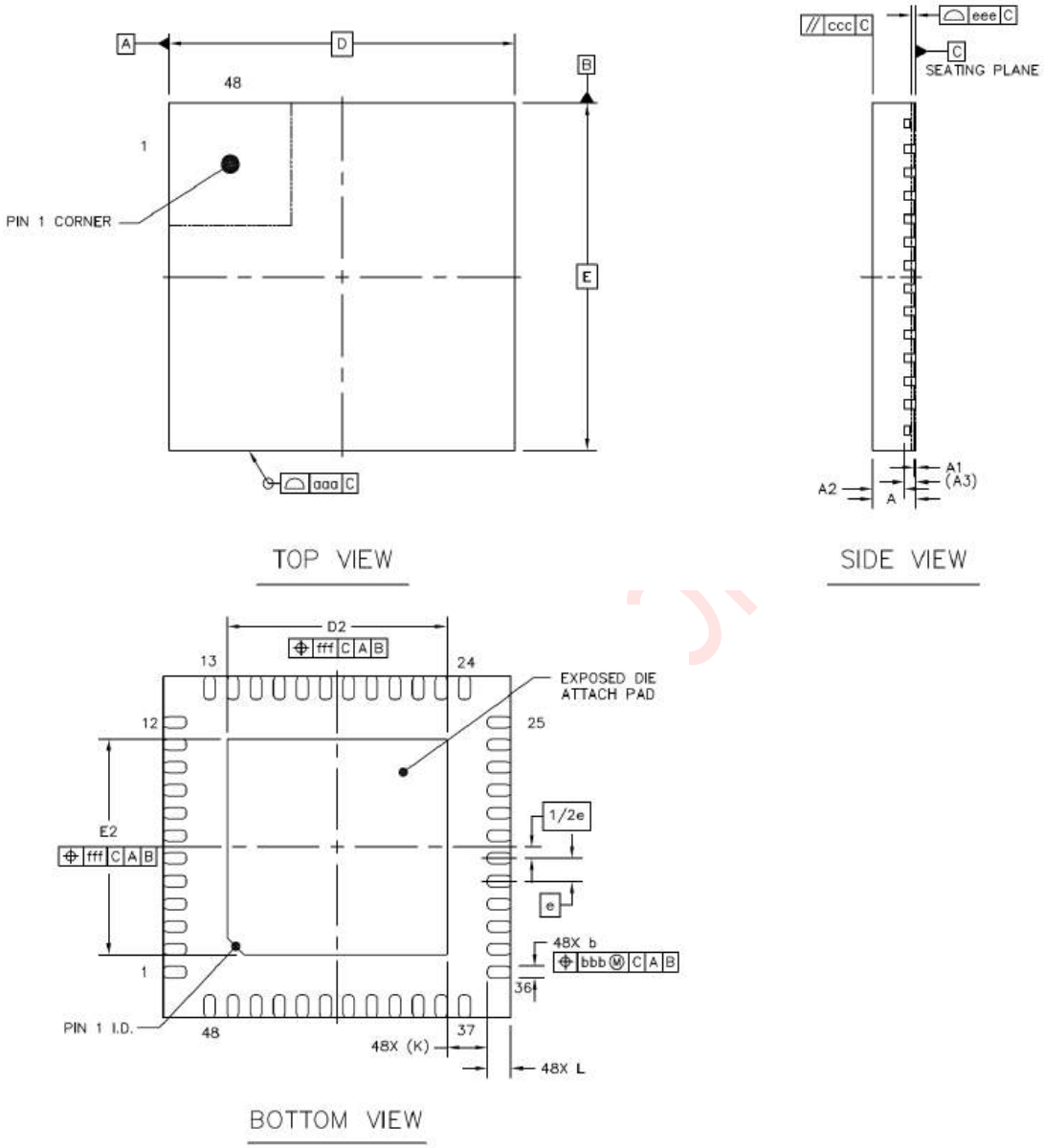


图 34-1：LT168A 外观尺寸图

©Levetop Semiconductor Co., Ltd.

表 34-1：封装尺寸参数

尺寸说明		Symbol	Min.	Nom.	Max.
Total Thickness		A	0.80	0.85	0.90
Stand Off		A1	0	0.02	0.05
Mold Thickness		A2	--	0.55	--
L/F Thickness		A3		0.203 Ref	
Lead Width		b	0.15	0.2	0.25
Body Size	X	D		6 BSC	
	Y	E		6 BSC	
Lead Pitch		e		0.4 BSC	
EP Size	X	D2	3.7	3.8	3.9
	Y	E2	3.7	3.8	3.9
Lead Length		L	0.3	0.4	0.5
Lead Tip to Exposed Pad Edge		K		0.7 Ref	
Package Edge Tolerance		aaa		0.1	
Mold Flatness		ccc		0.1	
Coplanarity		eee		0.08	
Lead Offset		bbb		0.07	
Exposed Pad Offset		fff		0.1	

提示：PCB 布局时，LT168A 背部的散热焊盘（Thermal Pad Zone）必须直接接地。

Levetop Semiconductor

©Levetop Semiconductor Co., Ltd.

34.2. LT168B (QFN-68pin)

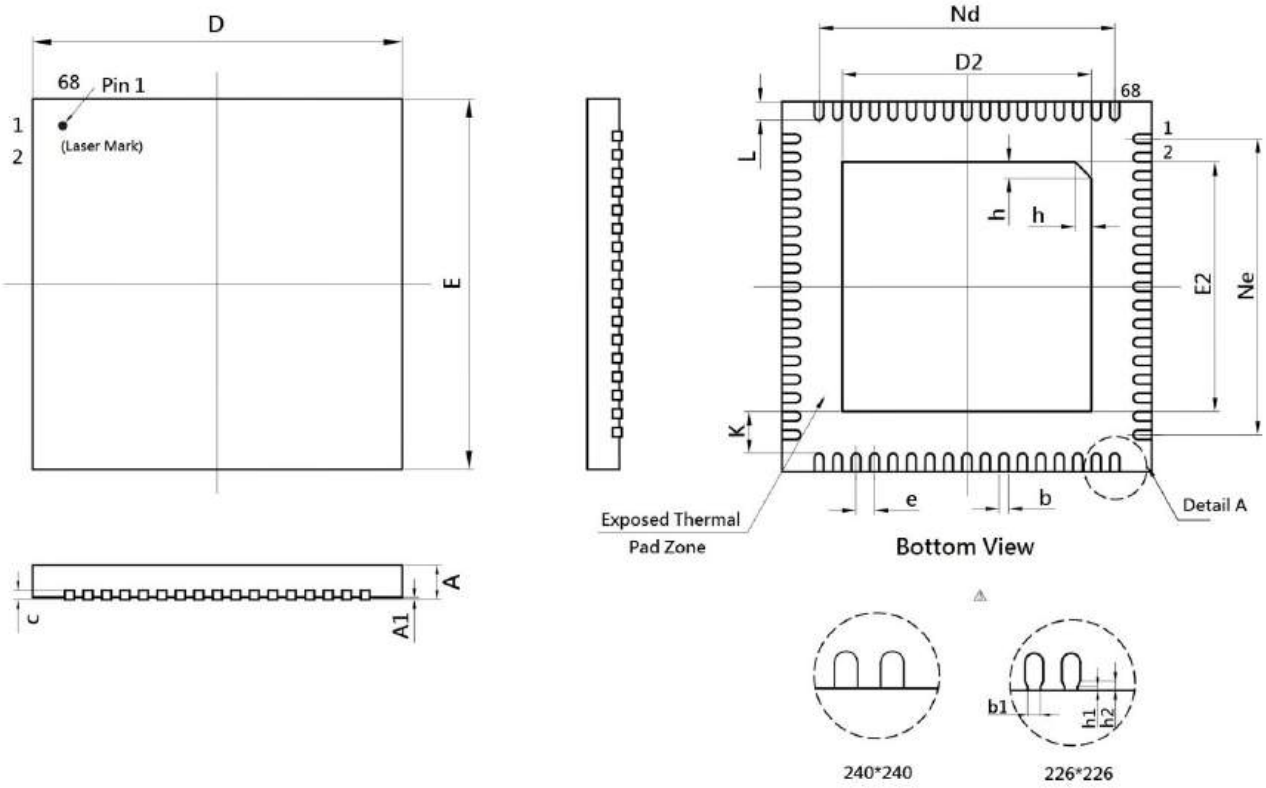


图 34-2：LT168B 外观尺寸图

提示：PCB 布局时，LT168B 背部的散热焊盘（Thermal Pad Zone）必须直接接地。焊盘的 PCB 布线请参考 34.3 节的说明。

表 34-2：LT168B 封装尺寸参数

Symbol	Millimeter			Symbol	Millimeter		
	Min.	Nom.	Max		Min.	Nom.	Max
A	0.80	0.85	0.90	Ne	6.40BSC		
A1	-	0.02	0.05	L	0.35	0.40	0.45
b	0.15	0.20	0.25	K	0.20	-	-
b1	0.14REF			h	0.30	0.35	0.40
c	0.18	0.20	0.25	h1	0.04REF		
D	7.90	8.00	8.10	h2	0.10REF		
e	0.40BSC			D2	5.39	5.49	5.59
Nd	6.40BSC			E2	5.39	5.49	5.59
E	7.9	8.0	8.10				

©Levetop Semiconductor Co., Ltd.

34.3. 芯片接地焊盘的 PCB 设计

LT168 采用 QFN 封装，芯片背部为接地（GND）的散热焊盘，为了达到更好的散热与降低焊接风险，在 PCB Layout 时建议把 LT168 底部焊盘的 PCB 铜箔面分割为四个或是多个小的焊接面（方形或是圆形），并且各焊接面之间的间隔设置在~0.8mm，避免 PCB 使用相同甚至大于 LT168 焊盘大小的完整焊接面而造成焊接不全，或是在焊接冷却后 PCB 与芯片焊盘拉扯导致芯片变形及接触不良。正确的 PCB 焊盘布局如下图 2 个 LT168B 范例，中间浅黄色区是 LT168 底部的接地焊盘，灰色区是 PCB 接地小焊盘（焊接面），每个焊盘过孔接地 1~2 个既可。

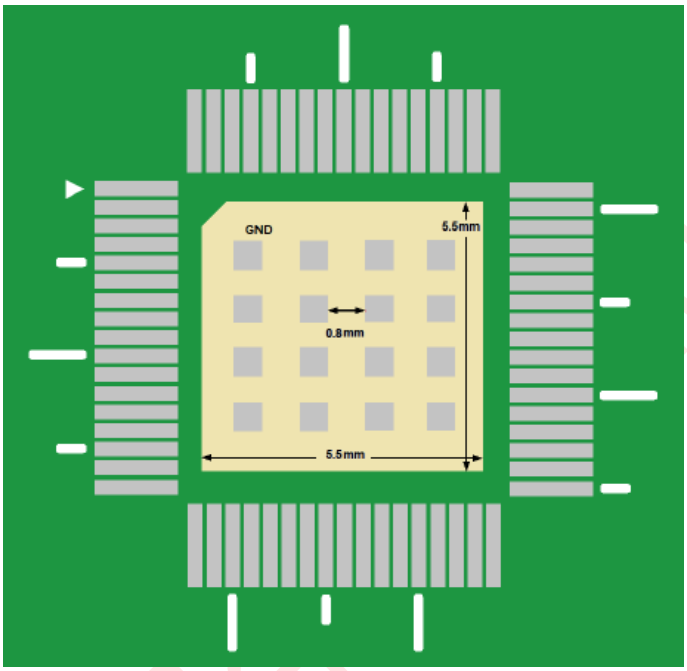


图 34-3：LT168B 底部焊盘 PCB 的设计建议-1

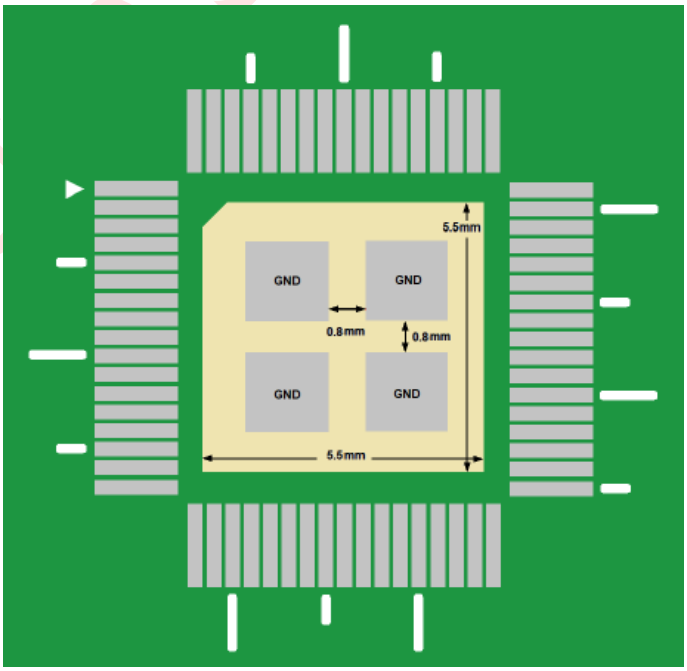


图 34-4：LT168B 底部焊盘 PCB 的设计建议-2

©Levetop Semiconductor Co., Ltd.