

单相计量 SOC
RN821x_RN7213 V2 版
用户手册
Rev1.8

深圳市锐能微科技有限公司

修改记录

版本号	发布日期	更改人	主要更改内容
V1.0	2023-10-18	系统部	首次发布
Rev1.1	2023-12-26 2023-12-29	客服部	<p>1、修改一处笔误，P46 写成了 P48</p> <p>2、管脚图修订:P22 P23 删 TC 功能;P31 P32 对应 D2F 及 CF_OUT 序号修改; RN8213 管脚图与 RN8217 分离; RN8211B 管脚图删除 P10 P11 P57 P50 P51 的脉冲转发和 D2F 输出功能,删 P32 P56 管脚上的 D2F 输出功能;</p> <p>3、RN8213B 引脚图下的 PIN 脚差异说明中修改 41 和 56 脚的说明, 41 脚由 P140 改成 NC; 56 脚由 NC 改为 P141;</p> <p>4、引脚说明章节修订管脚功能并增加 RN8213 和 RN8211B 不支持功能的注释</p> <p>5、IOCNT 复用关系表格修订,删除 P36 P37 作为 INT6 和 INT7 的功能输入功能;</p> <p>6、中断分配中 SIMP_TCx 序号修订为 0-3;</p> <p>7、修改文件名为可直接体现产品型号的名称</p> <p>8、增加管脚使用注意事项,包括 XO/XI 晶振附近管脚及具有模拟测量功能管脚,不能使用高速翻转功能</p> <p>9、删除 P14~P17 管脚上的 SPI 功能,解决与第 8 处修订的冲突;</p> <p>10、增加 LCDVA、LCDVB、LCDVC、LCDVD、LCDVP1、LCDVP2 液晶驱动引脚复用为 GPIO 使用时,液晶驱动电压无法输出,在需要使用液晶的应用上,不能将这些引脚做 GPIO 使用的说明。</p> <p>11、增加脉冲转发在检测到第一个待转发脉冲就输出第一个转发脉冲的注意事项、影响及规避方案。</p> <p>12、修改 D2F 模块脉冲输出配置和说明的笔误</p>
Rev1.2	2023-12-29	客服部	<p>1、修改 IOCNT_CTL 的 bit 位分配笔误</p> <p>2、相位校正修订扩展后范围是 bit0-bit10 有效, bit10 为符号位</p> <p>3、统一产品版本说明</p> <p>4、勘误湿敏等级缩写笔误 MSD 为 MSL</p> <p>5、增加 LCDVx 管脚应用说明</p>
Rev1.3	2024-03-19 2024-03-29	客服部	<p>1、优化描述:删除同步采样相位校正寄存器说明见其他章节的描述,直接增加相位校正寄存器的功能描述</p> <p>2、增加描述:GPIO 章节概述增加上/下拉电阻说明</p> <p>3、错误修订: PUD140 管脚上拉配置默认值对应上拉状态修改为上拉使能</p> <p>4、错误修订: 1.6.6 管脚定义说明</p> <p>P146 删除功能 ADC_CLKO 和 TRIG_OUT</p> <p>P31 修改功能为 CF_OUT0 和 D2F_OUT0</p> <p>P32 修改功能为 CF_OUT1 和 D2F_OUT1</p> <p>P20 增加功能 TC1_N[1]</p> <p>P21 增加功能 TC1_P[1]</p> <p>P22 增加 SPI1_MOSI</p> <p>P117 管脚序号修改为 128</p>

			<p>5、错误修订：RN8213B 引脚图管脚功能</p> <p>4 脚删除 D2F_OUT5 功能</p> <p>123 脚删除 TRIG_OUT</p> <p>31 脚 INT7 功能修改成 TRIG_OUT</p> <p>6、错误修订：RN8217 引脚图管脚功能</p> <p>20 脚删除 TRIG_OUT 功能</p> <p>83 脚删 ADC_CLK 和 TRIG_OUT 功能</p> <p>98 脚删除 TRIG_OUT</p> <p>21、22 脚增加 ADC_CLK 功能</p> <p>75 脚加 INT7 功能</p> <p>76 脚加 INT6 功能</p> <p>7、错误修订：RN8213 引脚图管脚功能</p> <p>20 脚删除 TRIG_OUT 功能</p> <p>83 脚删 ADC_CLK 和 TRIG_OUT 功能</p> <p>98 脚删除 TRIG_OUT</p> <p>21、22 脚增加 ADC_CLK 功能</p> <p>75 脚加 INT7 功能</p> <p>76 脚加 INT6 功能</p> <p>92、93、98、99、100 脚保留 IOCNT_OUT 功能，删除 CF_OUT 和 D2F_OUT 功能</p> <p>8、错误修订：RN7213 引脚图管脚功能</p> <p>2、3 脚删除 U_IN 功能</p> <p>18 脚删 SPI3_MOSI 功能</p> <p>48 脚加 INT7 功能</p> <p>49 脚加 INT6 功能</p> <p>92、93、98、99、100 脚保留 IOCNT_OUT 功能，删除 CF_OUT 和 D2F_OUT 功能</p> <p>9、错误修订：RN8211B 引脚图管脚功能</p> <p>6 脚删除 U_IN 功能</p> <p>48 脚加 INT7 功能</p> <p>49 脚加 INT6 功能</p> <p>58、59、62 脚保留 IOCNT_OUT 功能，删除 CF_OUT 和 D2F_OUT 功能</p>
Rev1.4	2024-06-20	客服部	1、错误修订：RN8217、RN8213、RN8215 的第 26 脚，修改为 GND，不再作为 GPIO 管脚使用。
Rev1.5	2024-08-19	客服部	<p>1、优化描述：SPI 章节中 DMA 地址和长度说明进行优化；删除 UART 章节中 DMA 地址和长度的多余描述防止误导。</p> <p>2、优化描述：统一手册名称</p> <p>3、增加描述：增加 RTC 温补和结果输出寄存器</p> <p>4、错误修改：DSP 中 DMA_SRBADR 等寄存器描述有效位数修正为 15 位</p>

			5、错误修订：删除 TC_CTRL[27:24]位描述，实际已无相应功能 6、错误修订：UARTx_BAUD 寄存器位宽修订为 13 位 7、增加描述：编程支持内增加下载到 FLASH 内代码文件第 8 个 word 的处理描述
Rev1.6	2024-09-29	客服部	1、优化描述：优化计量章节中用于 A 路电能计算功率状态标志位说明 2、优化描述：增加 P52、P53 寄存器复用位 TC_N 等功能时，管脚配置流程说明 3、增加描述：RTC 章节中增加测温启动和温度寄存器 3 说明 4、修改描述：修改 SAR_CTRL1 寄存器中 SAR_CONVERT 的说明为预留位，不能修改；防止客户改动后影响 RTC 测温时间 5、增加描述：UARTx_STA 寄存器中接收标志增加读后清零说明 6、增加描述：2.1 性能参数中增加模拟参数数据 7、优化描述：修改系统复位寄存器中位 EMU_RST_REQ 描述，“不复位 EMU 模块配置寄存器”改成“不复位 EMU 模块寄存器”
Rev1.7	2025-01-03	客服部	1、增加描述：RTC_TEMP 寄存器内增加用户温补模式 0，且为四次曲线温补时，寄存器扩展为 16 位 2、修改描述：TEMP_CAL 修改为 RTC_TEMP3，并为其增加与实际温度转换说明 3、增加描述：增加 UARTx_DMA_CTRL 寄存器位描述 4、增加描述：LCD_CTRL.EN 及 LCD_STATUS.BUSY 位增加低功耗处理流程描述 5、增加描述：有效值寄存器增加与外部采样信号关系描述 6、增加描述：波形采样寄存器增加与外部采样信号关系、及与有效值寄存器关系描述 7、增加描述：同步采样配置下增加同步采样波形数据与计量通道波形采样数据的关系描述 8、增加描述：增加 WDT 配置寄存器描述 9、修改描述：LCD 使用电阻分压方式时，同 Pump 方式，LCDVx 管脚也不支持 GPIO 功能 10、增加描述：VBAT 管脚增加“外接 0.1uF 电容”的描述 11、增加描述：LCDVxx 管脚应用增加描述
Rev1.8	2025-03-12	客服部	1、增加描述：增加 UART 波特率配置说明 2、错误修订：休眠功耗由 7uA 增加 1uA 到 8uA 3、错误修订：bitband 支持范围修改

目录

REV1.8.....	1
修改记录.....	2
目录.....	5
1 概述.....	21
1.1 简介.....	21
1.2 产品特点.....	21
1.2.1 基本特点.....	21
1.2.2 处理器相关.....	21
1.2.3 计量.....	21
1.2.4 RTC.....	22
1.2.5 LCD.....	22
1.2.6 其他外设.....	22
1.2.7 DSP 协处理器.....	23
1.2.8 加密.....	23
1.2.9 卡表隔离.....	23
1.3 SOC 产品型号列表.....	24
1.4 SOC 产品资源对照表.....	24
1.5 系统框图.....	25
1.6 管脚说明.....	26
1.6.1 RN8213B LQFP128L 引脚图.....	26
1.6.2 RN8217/RN8215 LQFP100L 引脚图.....	28
1.6.3 RN8213 LQFP100L 引脚图.....	29
1.6.4 RN7213 LQFP64L 引脚图.....	31
1.6.5 RN8211B LQFP64L 引脚图.....	32
1.6.6 引脚类型说明.....	33
1.6.7 管脚定义说明.....	33
1.6.8 硬件设计规则.....	48
1.7 IO 口功能框图.....	49
1.7.1 PABULD3.....	49
1.7.2 PABUS3.....	50
1.7.3 PBDS3.....	51
1.7.4 PBDSG3.....	52
1.7.5 PBULD3/6.....	53
1.7.6 PBUS6.....	54
1.7.7 PIUX.....	55
2 电气特性.....	56
2.1 性能参数.....	56
2.2 可靠性参数.....	58

3 系统控制	59
3.1 电源管理方案	59
3.2 时钟源	59
3.3 时钟切换	60
3.4 Soc 的低功耗模式	61
3.5 复位	63
3.5.1 外部 PIN 复位	63
3.5.2 上下电复位	63
3.5.3 软件复位	63
3.5.4 看门狗复位	63
3.5.5 CM0 复位	63
3.6 掉电处理	63
3.7 寄存器描述	63
3.7.1 系统 OSC 控制寄存器 1 OSC_CTRL1(0x0)	64
3.7.2 系统模式设置寄存器 SYS_MODE(0x4)	65
3.7.3 系统掉电控制寄存器 SYS_PD(0x8)	65
3.7.4 ADC 控制寄存器 ADC_CTRL(0xC)	67
3.7.5 系统 OSC 控制寄存器 2 OSC_CTRL2(0x10)	68
3.7.6 系统复位寄存器 SYS_RST(0x14)	70
3.7.7 地址映射控制寄存器 MAP_CTRL(0x18)	71
3.7.8 模块使能 0 寄存器 MOD0_EN(0x1C)	71
3.7.9 模块使能 1 寄存器 MOD1_EN(0x20)	73
3.7.10 INTC 使能寄存器 INTC_EN(0x24)	74
3.7.11 KBI 使能寄存器 KBI_EN(0x28)	75
3.7.12 器件 ID 寄存器 CHIP_ID(0x2C)	76
3.7.13 系统控制密码寄存器 SYS_PS(0x30)	76
3.7.14 红外配置寄存器 IRFR_CTRL (0x34)	76
3.7.15 系统配置寄存器 SYS_CFG(0x38) (新增)	77
3.7.16 时钟校正配置寄存器 TRIM_CFG1(0x78)	77
3.7.17 时钟校正启动寄存器 TRIM_START(0x7C)	78
3.7.18 DMA 优先级配置寄存器 1DMA_PRI1(0x80) (新增)	79
3.7.19 DMA 优先级配置寄存器 2DMA_PRI2(0xFC) (新增)	79
3.7.20 芯片唯一码寄存器 0FAB_UID0(0xF0) (新增)	80
3.7.21 芯片唯一码寄存器 1FAB_UID1(0xF4) (新增)	80
3.7.22 ADC 外灌控制寄存器 ADCIN_CTRL(0x114) (新增)	80
3.7.23 系统斩波配置寄存器 SYSCP_CON(0x118) (新增)	80
4 CPU 系统	81
4.1 概述	81
4.2 CORTEX-M0 处理器	82
4.3 存储映射	83
4.3.1 存储重映射	83
4.3.2 Bitband 功能	84
4.3.3 SRAM	85

4.3.4 FLASH	85
4.4 中断分配	85
4.5 中断应用	87
5 计量	88
5.1 主要特点	88
5.2 寄存器列表	88
5.2.1 计量配置和状态寄存器列表	88
5.2.2 计量参数寄存器列表	92
5.2.3 同步采样寄存器列表	94
5.2.4 DMA 波形缓存寄存器列表	95
5.2.5 智能微断 RCD 寄存器列表	96
5.3 寄存器描述	96
5.3.1 计量配置寄存器	96
5.3.2 计量参数寄存器	117
5.3.3 状态和中断寄存器	121
5.3.4 同步采样配置和参数寄存器（新增）	131
5.3.5 DMA 缓存配置和参数寄存器(新增)	135
5.3.6 智能微断 RCD 配置和参数寄存器	140
5.4 特殊命令	145
5.5 校表方法	146
5.5.1 脉冲法校表	146
5.5.2 功率法校表	147
6 全失压测量 NVM	148
6.1 主要特点	148
6.2 功能框图	149
6.3 NVM 测量模式	149
6.3.1 单通道模式（向下兼容）	149
6.3.2 双通道同时测量模式	149
6.4 寄存器列表	149
6.5 寄存器说明	150
6.5.1 NVM_IE (0x0)	150
6.5.2 NVM_IF (0x4)	151
6.5.3 LSCFG (0x8)	152
6.5.4 LSDCOS (0xC)	153
6.5.5 LSTHO (0x10)	153
6.5.6 LSRMSx (0x14~0x18/0x40/0x44, 新增)	154
6.5.7 HFConst (0x20)	154
6.5.8 D2FPx (0x24~0x28)	154
6.5.9 LSMODE (0x2C, 新增)	154
6.5.10 LSDCOSIB (0x30, 新增)	155
6.5.11 LSTHOIB (0x34, 新增)	155
6.5.12 LSGSIX (0x38~0x3C, 新增)	155
6.5.13 LSADCINCFG (0x48 新增)	155

6.5.14 LSRMSU (0x4C, 新增)	156
6.5.15 LSPx (0x50~0x54, 新增)	156
6.6 实现方式	157
6.6.1 实现流程图	157
6.6.2 程序实现步骤	157
6.7 全失压测量时直流 OFFSET 校正过程	158
6.8 电能积分	158
7 闪变 FLK (新增)	158
7.1 主要特点	158
7.2 使用方法	158
7.3 寄存器列表	159
7.4 寄存器定义	159
7.4.1 FLK_EN (0x00)	159
7.4.2 FLK_IE (0x04)	160
7.4.3 FLK_IF (0x08)	160
7.4.4 UA_600HZ (0x0C)	160
7.4.5 FLK_PASS (0x10)	160
8 RTC	160
8.1 概述	160
8.2 特点	161
8.3 寄存器描述	161
8.3.1 RTC 寄存器组	162
8.3.2 二次曲线补偿寄存器组	170
8.3.3 四次曲线补偿寄存器组	172
8.3.4 结果输出寄存器组	175
8.4 RTC 时钟读写步骤	177
8.5 RTC 校准步骤	177
8.6 RTC 定时器操作步骤	177
9 WDT	179
9.1 概述	179
9.2 看门狗定时器的配置	179
9.3 寄存器描述	180
9.3.1 WDT_EN (0x0)	180
9.3.2 WDT_CTRL (0x04)	180
9.3.3 WDT_PASS (0x08)	181
9.3.4 WDT_HALT (0x14)	181
9.3.5 WDT_STBY (0x18)	181
9.4 WDT 操作步骤	182
10 LCD	182
10.1 概述	182
10.1.1 扫描时钟频率	182

10.1.2 闪烁模式	183
10.1.3 LCD 驱动波形	183
10.1.4 LCD 偏置电压-电荷泵方式	189
10.1.5 LCD 偏置电压-电阻串方式	189
10.1.6 LCD 帧缓冲映射	189
10.2 寄存器描述	190
10.2.1 LCD 控制寄存器 LCD_CTL (0x0)	190
10.2.2 LCD 状态寄存器 LCD_STATUS (0x4)	191
10.2.3 LCD 时钟控制寄存器 LCD_CLKDIV (0x8)	192
10.2.4 LCD 闪烁控制寄存器 LCD_BLINK (0xC)	192
10.2.5 LCD 电荷泵建立时间寄存器 LCD_PS (0x10) (修改)	192
10.2.6 LCD 内部电阻串控制寄存器 LCD_RESCTL (0x14)	193
10.2.7 LCD 数据寄存器 LCD_BUFx (x=0~34) (地址 0x20-0x41)	194
11 定时器 (修改)	195
11.1 概述	195
11.2 功能框图	196
11.3 PWM 调制控制模式 (新增)	197
11.4 寄存器描述	198
11.4.1 当前计数值寄存器 TC_CNT (0x00)	198
11.4.2 预分频寄存器 TC_PS (0x04)	198
11.4.3 目标计数值寄存器 TC_DN (0x0C)	198
11.4.4 捕获比较通道 0 数据寄存器 TC_CCD0 (0x014)	198
11.4.5 捕获比较通道 1 数据寄存器 TC_CCD1(0x018)	198
11.4.6 时钟配置寄存器 TC_CCFG (0x01C)	199
11.4.7 控制寄存器 TC_CTRL (0x020)	200
11.4.8 捕获比较通道 0/1 模式寄存器 TC_CM0/1 (0x024 和 0x028)	201
11.4.9 中断使能寄存器 TC_IE (0x2C)	201
11.4.10 状态寄存器 TC_STA (0x30)	202
11.5 典型应用	202
11.5.1 自动运行模式, 定时功能	202
11.5.2 输入捕获模式, 脉宽测量功能	203
11.5.3 比较输出模式, 方波输出功能	203
11.5.4 比较输出模式, PWM 输出功能	204
11.5.5 从模式, 外部清零和门控功能	206
11.6 操作步骤	206
11.7 PWM 模式寄存器描述 (新增)	207
11.7.1 PWM_CFG (0x34)	207
11.7.2 PWM_CTL (0x38)	208
11.7.3 PWM_STA (0x3C)	208
11.7.4 PWM_CNT (0x40)	209
11.7.5 PWM_DMA_BADR (0x44)	209
11.7.6 PWM_DMA_LEN (0x48)	209
11.7.7 PWM_DMA_ADR (0x4C)	210
11.8 PWM 模式软件操作流程	210

11.8.1 参数限制	210
11.8.2 DMA 控制模式	210
11.8.3 软件控制模式	210
12 模拟外设	212
12.1 特点	212
12.2 寄存器	212
12.2.1 SAR_CTRL(0x0)	213
12.2.2 SAR_START(0x4)	214
12.2.3 SAR_STAT(0x8)	214
12.2.4 SAR_DAT(0xC)	215
12.2.5 LVD_CTRL(0x10)	215
12.2.6 LVD_STAT(0x14)	216
12.2.7 SAR_CTRL1(0x18) (新增)	216
12.2.8 SAR_DAT2(0x1C) (新增)	217
12.2.9 SAR_CTRL3 (0x20) (新增)	217
12.2.10 SAR_DAT3 (0x24) (新增)	218
12.3 ADC 电压检测步骤	218
12.4 VBAT 电压检测	218
12.5 低电压检测应用	219
13 GPIO (修改)	220
13.1 概述	220
13.2 寄存器描述	220
13.3 PA 口	221
13.3.1 PA 口模式寄存器 PMA (输入或者输出) (0x00)	221
13.3.2 PA 口数据寄存器 PA (0x04)	222
13.3.3 PA 口复用 0 寄存器 PCA0 (0x08)	222
13.3.4 PA 口复用 1 寄存器 PCA1 (0x0C)	226
13.3.5 PA 口复用寄存器 2 PCA2 (0x8C) (新增)	228
13.3.6 PA 口复用寄存器 3 PCA3 (0x90) (新增)	229
13.3.7 PA 口上拉选择寄存器 PUA (0x10)	231
13.3.8 PA 口输入模式配置寄存器 PIMA (0x14)	231
13.3.9 PA 口输入模式配置 PIMA2 (0xB0) (新增)	231
13.3.10 PA 口输入使能寄存器 PIEA (0x18)	232
13.4 PB 口	232
13.4.1 PB 口模式寄存器 PMB (输入或者输出) (0x1C)	232
13.4.2 PB 口数据寄存器 PB (0x20)	232
13.4.3 PB 口复用寄存器 PCB (0x24)	233
13.4.4 PB 口复用寄存器 2 PCB2 (0x4C) (修改)	237
13.4.5 PB 口复用寄存器 3 PCB3 (0x94) (新增)	237
13.4.6 PB 口上下拉选择寄存器 PUB (0x28)	238
13.4.7 PB 口输入模式寄存器 PIMB (0x2C) (修改)	238
13.4.8 PB 口输入使能寄存器 PIEB (0x30)	239
13.5 PC 口	239

13.5.1 PC 口模式寄存器 PMC (输入或者输出) (0x34)	239
13.5.2 PC 口数据寄存器 PC (0x38)	239
13.5.3 PC 口复用寄存器 PCC (0x3C) (修改)	240
13.5.1 PC 口复用寄存器 2 PCC2 (0x98) (新增)	242
13.5.2 PC 口复用寄存器 3 PCC3 (0x9C) (新增)	242
13.5.3 PC 口复用寄存器 4 PCC4 (0xA0) (新增)	243
13.5.4 PC 口上下拉选择寄存器 PUC (0x40)	243
13.5.5 PC 口输入使能寄存器 PIEC (0x44) (修改)	244
13.5.6 PC 口输入模式寄存器 PIMC (0x48)	244
13.6 PD 口 (新增)	244
13.6.1 PD 口模式寄存器 PMD (0x50) (输入或者输出) (新增)	244
13.6.2 PD 口数据寄存器 PD (0x54) (新增)	245
13.6.3 PD 口复用寄存器 PCD (0x58) (新增)	245
13.6.4 PD 口复用寄存器 PCD2 (0xA8) (新增)	245
13.6.5 PD 口上/下拉选择寄存器 PUD (0x5C) (新增)	246
13.6.6 PD 口输入使能 PIED (0x84) (新增)	247
13.6.7 PD 口输入模式寄存器 PIMD (0x88) (新增)	247
13.7 COM 端口	248
13.7.1 SEGCOM 口复用寄存器 PCE (0x60)	248
13.8 置位和清零寄存器	248
13.8.1 PA 口数据置位寄存器 PASET (0x64)	248
13.8.2 PA 口清零置位寄存器 PACLR (0x68)	248
13.8.3 PB 口数据置位寄存器 PBSET (0x6C)	249
13.8.4 PB 口清零置位寄存器 PBCLR (0x70)	249
13.8.5 PC 口数据置位寄存器 PCSET (0x74)	249
13.8.6 PC 口清零置位寄存器 PCCLR (0x78)	250
13.8.1 PD 口数据置位寄存器 PDSET (0x7C) (新增)	250
13.8.2 PD 口清零置位寄存器 PDCLR (0x80) (新增)	251
13.9 IO 配置寄存器	251
14 外部中断控制器 INTC (修改)	251
14.1 概述	251
14.2 寄存器描述	251
14.2.1 INTC_CTL (0x0) (修改)	252
14.2.2 INTC_MODE (0x4)	252
14.2.3 INTC_MASK (0x8)	253
14.2.4 INTC_STA (0xC)	253
14.2.5 LPUART 复用配置寄存器 LURT_CFG (0x100) (新增)	254
14.2.6 IO 驱动配置寄存器 IOCFG (0x104) (新增)	254
14.3 GPIO 操作步骤	255
15 KBI	256
15.1 特性	256
15.2 寄存器描述	256
15.2.1 控制寄存器 KBI_CTL (0x0)	256

15.2.2 选择寄存器 KBI_SEL (0x4)	256
15.2.3 数据寄存器 KBI_DATA (0x8)	256
15.2.4 屏蔽寄存器 KBI_MASK (0xC)	257
15.3 KBI 操作步骤	257
16 UART (修改)	258
16.1 概述	258
16.2 寄存器描述	258
16.2.1 控制寄存器 UARTx_CTL (0x0)	259
16.2.2 波特率整数分频寄存器 UARTx_BAUD (0x4)	260
16.2.3 状态指示寄存器 UARTx_STA (0x8)	260
16.2.4 发送数据寄存器 UARTx_TXD (0xC)	261
16.2.5 接收数据寄存器 UARTx_RXD (0x10)	262
16.2.6 波特率小数分频配置寄存器 UARTx_FDIV (0x14)	262
16.2.7 DMA 控制寄存器 UARTx_DMA_CTL (0x18)	262
16.2.8 DMA 发送起始地址寄存器 UARTx_DMA_TBADR (0x1C)	263
16.2.9 DMA 接收起始地址寄存器 UARTx_DMA_RBADR (0x20)	263
16.2.10 DMA 发送长度寄存器 UARTx_DMA_TLEN (0x24)	263
16.2.11 DMA 接收长度寄存器 UARTx_DMA_RLEN (0x28)	263
16.2.12 DMA 当前发送地址寄存器 UARTx_DMA_TADR (0x2C)	263
16.2.13 DMA 当前接收地址寄存器 UARTx_DMA_RADR (0x30)	263
16.2.14 DMA 中断使能寄存器 UARTx_DMA_IE (0x34)	263
16.2.15 DMA 中断标志寄存器 UARTx_DMA_IF (0x38)	264
16.2.16 DMA 接收超时配置寄存器 UARTx_DMA_TO (0x3C)	264
16.3 UART 数据接收及发送操作步骤	264
17 ISO7816	266
17.1 概述	266
17.2 寄存器描述	266
17.3 7816 与 ESAM 通讯操作步骤	274
17.4 7816 与卡通讯操作步骤	274
18 IIC 接口	276
18.1 概述	276
18.2 寄存器描述	276
19 SPI 接口 (修改)	279
19.1 概述	279
19.2 复用关系	279
19.3 功能描述	279
19.4 寄存器描述	280
19.4.1 SPI 控制寄存器 SPIx_CTRL (0x0)	281
19.4.2 SPI 状态寄存器 SPIx_STAT (0x4)	282
19.4.3 SPI 数据发送寄存器 SPIx_TXDATA (0x8)	283
19.4.4 SPI 数据接收寄存器 SPIx_RXDATA (0xC)	283

19.4.5 SPI 默认发送数据寄存器 SPIx_TXDFLT (0x10)	283
19.4.6 SPI DMA 控制寄存器 SPIx_DMA_CTRL (0x14)	284
19.4.7 SPI DMA 发送起始地址寄存器 SPIx_DMA_TBADR (0x18)	284
19.4.8 SPI DMA 接收起始地址寄存器 SPIx_DMA_RBADR (0x1C)	284
19.4.9 SPI DMA 发送长度寄存器 SPIx_DMA_TLEN (0x20)	284
19.4.10 SPI DMA 接收长度寄存器 SPIx_DMA_RLEN (0x24)	284
19.4.11 SPI DMA 当前发送地址寄存器 SPI_DMA_TADR (0x28)	284
19.4.12 SPI DMA 当前接收地址寄存器 SPIx_DMA_RADR (0x2C)	285
19.4.13 SPI DMA 中断使能寄存器 SPIx_DMA_IE (0x30)	285
19.4.14 SPI DMA 中断标志寄存器 SPIx_DMA_IF (0x34)	285
19.5 应用方法	286
20 LPUART (新增)	286
20.1 概述	286
20.2 特点	286
20.3 功能框图	287
20.4 帧结构	287
20.5 FIFO 功能	288
20.6 发送引擎	288
20.6.1 发送特性	288
20.6.2 单字节通信流程	288
20.7 接收引擎	289
20.7.1 传输特性	289
20.7.2 起始位检测	289
20.7.3 停止位检测	289
20.7.4 空闲帧检测	289
20.7.5 接收错误	290
20.7.6 波特率发生器	290
20.7.7 奇偶检验位	290
20.7.8 低功耗唤醒模式	290
20.8 寄存器描述	291
20.8.1 寄存器列表	291
20.8.2 LPUART_MODE (0x00)	292
20.8.3 LPUART_IE (0x04)	293
20.8.4 LPUART_STA (0x08)	294
20.8.5 LPUART_BAUD (0xC)	296
20.8.6 LPUART_TXD (0x10)	296
20.8.7 LPUART_RXD (0x14)	296
20.8.8 LPUART_DMR (0x18)	297
20.9 软件操作流程	297
20.9.1 发送流程	297
20.9.2 接收流程	297
21 CRC (新增)	297
21.1 概述	297

21.2 特点	297
21.3 基本原理	298
21.4 多项式	298
21.5 余数初始值	298
21.6 结果异或值	299
21.7 输入数据反转	299
21.8 DMA 接口	299
21.8.1 CRC 计算	299
21.8.2 Checksum 计算	299
21.9 计算速度评估	300
21.9.1 CRC 计算	300
21.9.2 Checksum 计算	300
21.10 寄存器描述	300
21.10.1 CRC_DR (0x00)	300
21.10.2 CRC_STA (0x04)	301
21.10.3 CRC_CTRL (0x08)	301
21.10.4 CRC_INIT (0x0C)	302
21.10.5 CRC_POL (0x10)	303
21.10.6 CRC_XOR (0x14)	303
21.10.7 CRC_DMA_CTL (0x18)	304
21.10.8 CRC_DMA_BADR (0x1C)	304
21.10.9 CRC_DMA_LEN (0x20)	305
21.10.10 CRC_DMA_ADR (0x24)	305
21.10.11 CRC_DMA_IE (0x28)	305
21.10.12 CRC_DMA_FLG (0x2C)	305
21.11 CRC 软件操作流程	306
21.11.1 配置 DR 寄存器进行计算	306
21.11.2 使用 DMA 进行计算	307
21.12 CHECKSUM 软件操作流程	308
21.12.1 使用 DMA 进行计算	308
21.12.2 计算过程中关闭 DMA	309
22 脉冲转发 IOCNT (新增)	309
22.1 概述	309
22.2 功能特性	309
22.3 复用关系	309
22.4 功能框图	311
22.5 波形分析	311
22.5.1 输入信号类型	311
22.5.2 脉冲计数	312
22.5.3 脉冲分频	315
22.5.4 脉冲电平反向输出	319
22.6 寄存器描述	319
22.6.1 寄存器列表	319
22.6.2 IOCNT_CFG0~4 (0x00~0x10) (新增)	319

22.6.3 IOCNT_OUT0~4 (0x20~0x30) (新增)	320
22.6.4 IOCNT_CHNL (0x40) (新增)	320
22.6.5 IOCNT_CTL (0x48) (新增)	320
22.7 使用流程	322
22.7.1 内部脉冲转发	322
22.7.2 高电平脉宽有效	322
22.7.3 低电平脉宽有效	322
23 安全密码加速器 SEA (新增)	322
23.1 AES 硬件加速单元	323
23.1.1 特点	323
23.1.2 时耗信息	323
23.2 ECC 硬件加速单元	323
23.2.1 特点	323
23.2.2 时耗信息	323
23.3 RSA 硬件加速单元	324
23.3.1 特点	324
23.3.2 时耗信息	324
23.4 HASH 硬件加速单元	324
23.4.1 特点	324
23.4.2 时耗信息	324
23.5 TRNG 真随机数发生器	325
23.5.1 特点	325
23.5.2 时耗信息	325
23.6 软件配置流程	325
24 简单定时器 SIMP_TC (新增)	326
24.1 概述	326
24.2 功能描述	326
24.3 寄存器描述	326
24.3.1 SIMP_TCx 控制寄存器 CTRL (0x0)	326
24.3.2 SIMP_TCx 目标计数值寄存器 LOAD (0x4)	327
24.3.3 SIMP_TCx 当前计数值寄存器 VAL (0x8)	327
25 内存搬运单元 M2M (新增)	327
25.1 特点	327
25.2 功能描述	327
25.2.1 输入输出数据有效位宽配置	327
25.2.2 输出地址无效时可配 dummy 值	328
25.2.3 输出数据反序放置	328
25.2.4 源和目标地址可配置, 输入数据长度可配置	328
25.2.5 数据按地址反序排列	328
25.3 实例说明	328
25.3.1 输入数据 4Bytes 有效, 输出 3Bytes 有效	328
25.3.2 输入数据 2Bytes 有效, 输出 4Bytes 有效	329

25.3.3 输入数据按 byte 反序存放（输入 4Bytes 有效，输出 2Bytes 有效）	329
25.3.4 输出数据按地址反序排列	329
25.4 搬运速度	329
25.5 寄存器描述	329
25.5.1 M2M_MODE (0x0)	330
25.5.2 M2M_CTL (0x4)	330
25.5.3 M2M_DUMMY (0x8)	330
25.5.4 M2M_SADDR (0xC)	331
25.5.5 M2M_DADDR (0x10)	331
25.5.6 M2M_ILEN (0x14)	331
25.5.7 M2M_IE (0x18)	331
25.5.8 M2M_IF (0x1C)	331
25.6 软件使用流程	332
26 DSP 核（新增）	332
26.1 特点	333
26.2 计算引擎基本原理	333
26.2.1 单精度浮点数表示	333
26.2.2 特殊数值	333
26.2.3 浮点数舍入处理	333
26.2.4 IEEE754 标准 rounding 模式	334
26.2.5 本芯片的 rounding 模式	334
26.2.6 整数转浮点数原理	334
26.2.7 浮点数转整数原理	335
26.2.8 浮点数乘法原理	335
26.2.9 浮点数加法原理	335
26.2.10 蝶形运算原理	335
26.2.11 IIR 滤波器原理	336
26.2.12 FIR 滤波器原理	337
26.2.13 线性插值	337
26.2.14 Lagrange 插值	338
26.3 运算指令	338
26.3.1 整数转浮点数 (int2fp/int2fp_dma)	338
26.3.2 浮点数转整数 (fp2int/fp2int_dma)	339
26.3.3 浮点数乘法 (fp_mult)	340
26.3.4 浮点数加法 (fp_add)	341
26.3.5 浮点数减法 (fp_sub)	341
26.3.6 浮点数乘加运算 (fp_mlad)	341
26.3.7 浮点数蝶形运算（单次）(btfy/btfy_dma)	342
26.3.8 正弦余弦计算 (sin_cos)	345
26.3.9 均方根和反正切 (fp_sqrt/ fp_atan)	345
26.3.10 浮点数除法器 (fp_div)	345
26.3.11 IIR 滤波器	346
26.3.12 FIR 滤波器	348
26.3.13 线性插值	349

26.3.14 拉格朗日插值	350
26.4 实现说明	351
26.4.1 整数转浮点数	351
26.4.2 浮点数转整数	351
26.4.3 FFT 说明	351
26.4.4 CORDIC 说明	352
26.5 寄存器	353
26.5.1 寄存器列表	353
26.5.2 MAC_CTL0 (0x0)	355
26.5.3 MAC_CTL1 (0x04)	356
26.5.4 MAC_CTL2 (0x08)	356
26.5.5 MAC_IN0 (0x0C)	357
26.5.6 MAC_IN1 (0x10)	357
26.5.7 MAC_IN2 (0x14)	357
26.5.8 MAC_IN3 (0x18)	357
26.5.9 MAC_IN4 (0x1C)	357
26.5.10 MAC_IN5 (0x20)	357
26.5.11 MAC_OUT0 (0x24)	357
26.5.12 MAC_OUT1 (0x28)	358
26.5.13 MAC_OUT2 (0x2C)	358
26.5.14 MAC_OUT3 (0x30)	358
26.5.15 DIV_IN0 (0x34)	358
26.5.16 DIV_IN1 (0x38)	358
26.5.17 DIV_OUT0 (0x3C)	358
26.5.18 DMA_SRBADR (0x40)	358
26.5.19 DMA_SIBADR (0x44)	359
26.5.20 DMA_PRBADR (0x48)	359
26.5.21 DMA_PIBADR (0x4C)	359
26.5.22 DMA_TRBADR (0x50)	359
26.5.23 DMA_TIBADR (0x54)	359
26.5.24 DMA_LEN (0x58)	359
26.5.25 DSP_IE (0x5C)	360
26.5.26 DSP_FLG (0x60)	361
26.5.27 ALU_STA0 (0x64)	361
26.5.28 ALU_STA1 (0x68)	362
26.5.29 CRD_CTL (0x6C)	362
26.5.30 CRD_XIN (0x70)	362
26.5.31 CRD_YIN (0x74)	362
26.5.32 CRD_AMP (0x78)	362
26.5.33 CRD_PHASE (0x7C)	363
26.5.34 CRD_ANGLE (0x80)	363
26.5.35 CRD_COSINE (0x84)	363
26.5.36 CRD_SINE (0x88)	363
26.5.37 CRD_IE (0x8C)	363

26.5.38 CRD_FLG (0x90)	363
26.5.39 INTP_LEN (0x94)	364
26.5.40 INTP_LOC (0x98)	364
26.5.41 INTP_STEP (0x9C)	364
26.6 软件操作流程	364
26.6.1 完整方案说明:	364
26.6.2 操作流程:	365
27 电能积分单元 D2F (新增)	365
27.1 概述	365
27.2 主要特点	365
27.3 使用流程	366
27.4 寄存器列表	366
27.5 寄存器定义	367
27.5.1 自定义 D2F 高频脉冲常数寄存器 HFConst4/5/6(0x00~0C)	367
27.5.2 自定义 D2F 配置寄存器 D2FCFG(0x10)	368
27.5.3 自定义 D2F 脉冲输出配置寄存器 D2FOUT_CFG(0x14)	368
27.5.4 自定义 D2F 中断使能寄存器 IE(0x18)	369
27.5.5 自定义 D2F 中断标志寄存器 IF(0x1C)	369
27.5.6 自定义 D2F 快速脉冲计数器 (0x20~0x4C)	370
27.5.7 自定义 D2F 功率寄存器 (0x50~0x6C)	370
27.5.8 自定义 D2F 能量寄存器 (0x80~0xAC)	371
28 计量误差温补模块 ECT (新增)	371
28.1 概述	371
28.2 温补增益系数的计算公式	371
28.3 特点	372
28.4 功能描述	372
28.4.1 功能框图	372
28.4.2 工作模式	373
28.4.3 温度保护	373
28.4.4 高温低温分段判断	373
28.4.5 温度越界事件处理	374
28.4.6 温度防抖	374
28.4.7 增益系数计算	374
28.4.8 误差越界事件处理	374
28.5 寄存器描述	374
28.5.1 寄存器列表	374
28.5.2 WREN (0x0)	375
28.5.3 CTRL (0x4)	375
28.5.4 EN (0x8)	376
28.5.5 STATUS (0xC)	376
28.5.6 IE (0x10)	377
28.5.7 LT_SET (0x14)	377
28.5.8 HT_SET (0x18)	377

28.5.9	TIMER_SET (0x1C)	378
28.5.10	PROT_TEMP (0x20)	378
28.5.11	PROT_IAGAIN (0x24)	378
28.5.12	PROT_IBGAIN (0x28)	378
28.5.13	PROT_UGAIN (0x2C)	379
28.5.14	LT_KIA (0x30)	379
28.5.15	LT_KIB (0x34)	379
28.5.16	LT_KU (0x38)	379
28.5.17	HT_KIA (0x3C)	380
28.5.18	HT_KIB (0x40)	380
28.5.19	HT_KU (0x44)	380
28.5.20	TEMP (0x48)	380
28.5.21	IAGAIN (0x4C)	381
28.5.22	IBGAIN (0x50)	381
28.5.23	UGAIN (0x54)	381
28.5.24	TEMP_UD (0x58)	381
28.6	应用流程	382
28.6.1	温补 K 系数标定	382
28.6.2	手动温补模式	382
28.6.3	单次自动温补模式	383
28.6.4	循环自动温补模式	383
29	选项字节	383
29.1	芯片保护设置	384
29.2	WDT 设置	384
29.3	RTC 设置	385
30	编程支持	386
30.1	概述	386
30.2	FLASH 保护机制	386
30.3	在系统编程 (ISP)	386
30.3.1	ISP 通讯协议	387
30.3.2	使用的 SoC 资源	388
30.3.3	ISP 命令	388
30.3.4	ISP 返回代码	392
30.4	在应用编程 (IAP)	392
30.4.1	IAP 命令	393
30.4.2	IAP 使用	393
30.5	量产平台	393
31	封装尺寸及焊接条件	394
31.1	封装尺寸	394
31.2	回流焊炉的温度设定条件	397
32	包装信息	398

32.1 托盘规格	398
-----------------	-----

1 概述

1.1 简介

单相 SOC 芯片 RN821x V2 版, RN721x V2 版是锐能微第二代单相 SOC 芯片, 在第一代的基础上优化性能, 拓展功能, 并向下兼容第一代产品, 即 V1 版。

V2 版与 V1 版芯片硬件 pin to pin 兼容, 寄存器兼容, 软件库函数需要升级到 V2 版。FLASH 最大支持 512KB, SRAM 最大支持 96KB+4KB。优化功耗, 提升 RTC 性能和计量性能, 支持智能微型断路器应用, 新增 ECC/AES/HASH/TRNG 加密硬件加速单元, 新增 DSP 核协处理器, SPI/UART 支持 DMA 通道。典型应用领域: 海外单相表、导轨表、智能用电安全产品等。

1.2 产品特点

1.2.1 基本特点

- 高集成: 集成 32bit ARM Cortex-M0、计量模块、硬件温补 RTC、LCD 控制器、DSP 协处理器、加密硬件加速;
- 宽电压: 保证计量精度的电压范围为 2.8V~5.5V;
CPU 小系统可运行的典型电压范围为 1.8V~5.5V;
GPIO 支持与不同工作电压器件的对接。
- 高性能: 32.768KHz 单晶振和 32.768KHz 晶振+外部高频晶振条件下, CPU 最高工作频率都可达到 29.4912MHz (32.768KHz、1.8432MHz、7.3728MHz、14.7456MHz、29.4912MHz 可选);
内部高频 RCH 最高可工作在 29.5Mhz 下, 全温度范围可保证±1%精度, 可用作备份时钟;
- 低功耗: 单相智能表计应用时 7.3728Mhz 功耗约为 3.5mA
系统工作在 32Khz 下功耗约为 22μA;
睡眠模式下芯片整体功耗约为 8μA。
- 高精度: 在 8000:1 动态范围内有功误差小于 0.1%; 计量参考基准温度系数典型值为 5ppm;
RTC 在-25℃~70℃内秒脉冲误差小于±5ppm, 最小校正刻度为 0.0339ppm;
- 封装形式:
RN8213B V2 版: LQFP128L
RN8217 V2 版/RN8215 V2 版/RN8213 V2 版: LQFP100L
RN7213 V2 版/RN8211B V2 版: LQFP64L

1.2.2 处理器相关

- ARM Cortex-M0 内核;
- 最大支持 512KBytes FLASH 存储器, 擦写次数 10 万次, 数据保持时间大于 20 年;
- 最大支持 96KBytes SRAM, 其中最高 4KBytes 为内部 cache 使用, 开放给客户使用的为 92KBytes;
另外 4KBytes SRAM 为加密模块 RAM, 不使用加密功能时, 这 4KBytes SRAM 可被 CPU 使用。
- 单 cycle 乘法器 (32bit*32bit)
- CM0 内嵌系统定时器;
- 支持外部中断等多种唤醒方式;
- 提供完善的集成开发软硬件环境;
- 支持 CM0 独立复位, 在线升级后不需要复位整个芯片, 可保证在线升级不影响计量

1.2.3 计量

- 在 8000:1 动态范围内有功计量误差小于 0.1%;

- 参考电压温度系数典型值为 5ppm/°C；
- 支持零线和火线双通道有功功率、无功功率、视在功率、电流有效值同时测量；
- 支持零线和火线双通道有功电能、无功电能、视在电能同时计量；
- 提供电压有效值及电压线频率测量；
- 提供采样通道增益及 offset 校正功能；
- 提供功率因数；
- 提供全失压计量解决方案
- 提供直流计量解决方案
- 提供电压骤升、骤降事件监测；提供电流过载事件监测；提供谐波分析解决方案；
- 提供三路计量 sigma-delta ADC 原始采样数据，便于做二次算法开发。
- 提供智能安全用电解决方案
- 支持罗氏线圈
- 支持半波计量模式
- 支持双向计量模式
- 提供基波计量模式
- 电表常数可设置，且 3 套独立配置
- 支持 ADC 输入反向功能
- 支持 IA/IB 通道 ADC 外灌功能
- 有功、无功支持 4 种电能累加方式：代数和、正向、绝对值、反向
- 提供灵活的 ADC 同步采样波形数据
- 支持灵活的校正手段，支持增益校正、相位校正、谐波补偿；
- 支持通过 DMA 方式将波形数据从计量传输到 RAM。
- 提供同步采样通道半波有功功率和半波有效值
- 提供半周期更新的全波电压电流有效值、全波有功功率
- 提供半周期更新的基波电压电流有效值、基波有功功率
- 提供闪变硬件运算模块 FLK
- 提供计量误差温度系数补偿模块 ECT

1.2.4 RTC

- 硬件自动温补，满足国家标准的精度和功耗要求；
- 温度传感器：提供准确的温度值，-25°C~70°C 范围内测温精度为±1°C；

1.2.5 LCD

- 支持 4*34、6*32、8*30 (LQFP100L)；或 4*40、6*38、8*36 (LQFP128L)；
- 支持 Charge pump 模式，支持宽电压、全温度范围清晰显示；
- 支持电阻列分压模式模式
- 两种模式硬件兼容，如果需要更低的显示功耗，可选择电阻串分压方式；
- 整机功耗优于 25μA。

1.2.6 其他外设

- 高速 GPIO，支持与不同电压外设器件的接口；
- 12bit ADC：温度传感器/电池电压检测/通用 ADC 分时复用；
- 电压检测 LVD：检测芯片电源电压；检测外部电压。
- 两个比较器 CMP1 和 CMP2：检测外部电压，功耗约为 0.5uA，支持停电下的低功耗电源监测。
- 定时器：2 个 32bit 扩展定时器，2 个 RTC 定时器，1 个 CM0 内嵌系统定时器，4 个 32bit 简单定时器；

- UART: 最多 6 个, 支持自动波特率, 支持红外调制, 支持 UART 唤醒, 支持电平反转, 支持 DMA, 注意 DMA 与 V1 版本不兼容
- 7816 口: 2 个
- I2C: 1 个
- SPI: 4 个, 支持 DMA, 注意 DMA 与 V1 版本不兼容
- LPUART: 1 个低功耗 UART, 最高支持 9600 波特率, 支持红外调制, 支持 4 种唤醒模式;
- 看门狗: 硬件看门狗;
- 按键中断: 最多 8 个, 管脚复用;
- 外部中断: 最多 8 个, 管脚复用;
- 电能积分单元 D2F: 提供 12 个 D2F 积分器, 其中 3 个积分器支持脉冲输出。
- 内存搬运单元 M2M: 1 个 M2M 模块, 可实现内存数据搬移。
- 提供脉冲转发功能 IOCNT
- 提供 7/8/16/32 位 CRC 检验

1.2.7 DSP 协处理器

- 支持整数和浮点数的相互转换
- 支持浮点数加法、减法、乘法、除法
- 支持单次蝶形运算（复数）以及 DMA 连续蝶形运算
- 支持基-2FFT 整个过程, 支持点数 64、128、256、512、1024
- 支持 bit reverse 自动搬运数据操作, 支持点数 4、8、16、32、64、128、256、512、1024
- 支持正弦、余弦计算
- 支持方均根计算
- 支持反正切计算
- 支持 IIR 单次计算, 支持 IIR DMA 计算
- 支持 FIR 滤波运算
- 支持线性插值运算
- 支持拉格朗日插值运算

1.2.8 加密

（与 V1 版本不兼容, 需要重新修改应用程序）

- 硬件真随机数产生器, 符合美国 NIST 的 FIPS 140-2 标准;
- AES 硬件加速单元;
- ECC 硬件加速单元;
- RSA 硬件加速单元;
- HASH 散列算法硬件加速单元;
- 加密说明文档见锐能微应用笔记;

1.2.9 卡表隔离

- 提供卡表隔离低成本高可靠解决方案, 详见《RN8501 用户手册》;

1.3 SOC 产品型号列表

型号	FLASH 容量	RAM 容量	计量 ADC	LCD	VCC 与 VBAT 切换	加密	DSP 核	封装
RN8213B	512KB	96KB+4KB	3 路	√	√	√	√	LQFP128L
RN8217	512KB	96KB+4KB	3 路	√	√	√	√	LQFP100L
RN8215								
RN7213	512KB	96KB+4KB	3 路	×	√	√	√	LQFP64L
RN8213	256KB	32KB	3 路	√	√	×	×	LQFP100L
RN8211B	256KB	32KB	3 路	√	×	×	×	LQFP64L

表 1-1 SOC 产品型号列表

1.4 SOC 产品资源对照表

型号	RN8213B	RN8217 RN8215	RN7213	RN8213	RN8211B
CPU	Cortex-M0				
Max Freq.	29.4912M				
FLASH	512KB	512KB	512KB	256KB	256KB
CACHE	√	√	√	×	×
RAM	96KB+4KB	96KB+4KB	96KB+4KB	32KB	32KB
加密 SEA	√	√	√	×	×
TRNG	√	√	√	×	×
Timers	32bit Timer	2	2	2	2
	32bit Simptc	4	4	4	4
	systick	1	1	1	1
RTC	1	1	1	1	1
WDT	1	1	1	1	1
KBI	8	8	6	6	5
INTC	8	8	8	8	4
SPI	4	4	4	4	2
UART	6	5	5	5	4
I2C	1	1	1	1	1
7816	2	2	2	0	0
GPIO	108	82	46	46	48
DMA	√				
计量 EMU	√				
DSP	√	√	√	×	×
D2F	√	√	√	×	×
M2M	√	√	√	×	×
LPUART	√	√	√	×	×
ECT	√	√	√	×	×

FLK	√	√	√	×	×
CRC	√	√	√	×	×
IOCNT	√	√	√	×	×
LCD	4*40/6*38/8*36	4*34/6*32/8*30	×	4*34/6*32/8*30	4*16/6*14/8*12
CMP	2	2	1	2	2
LVD	1	1	0	1	0
计量 ADC	3	3	3	3	3
SAR-ADC	7	6	5	6	5
TempSensor	√				

表 1-2 SOC 产品资源对照表

1.5 系统框图

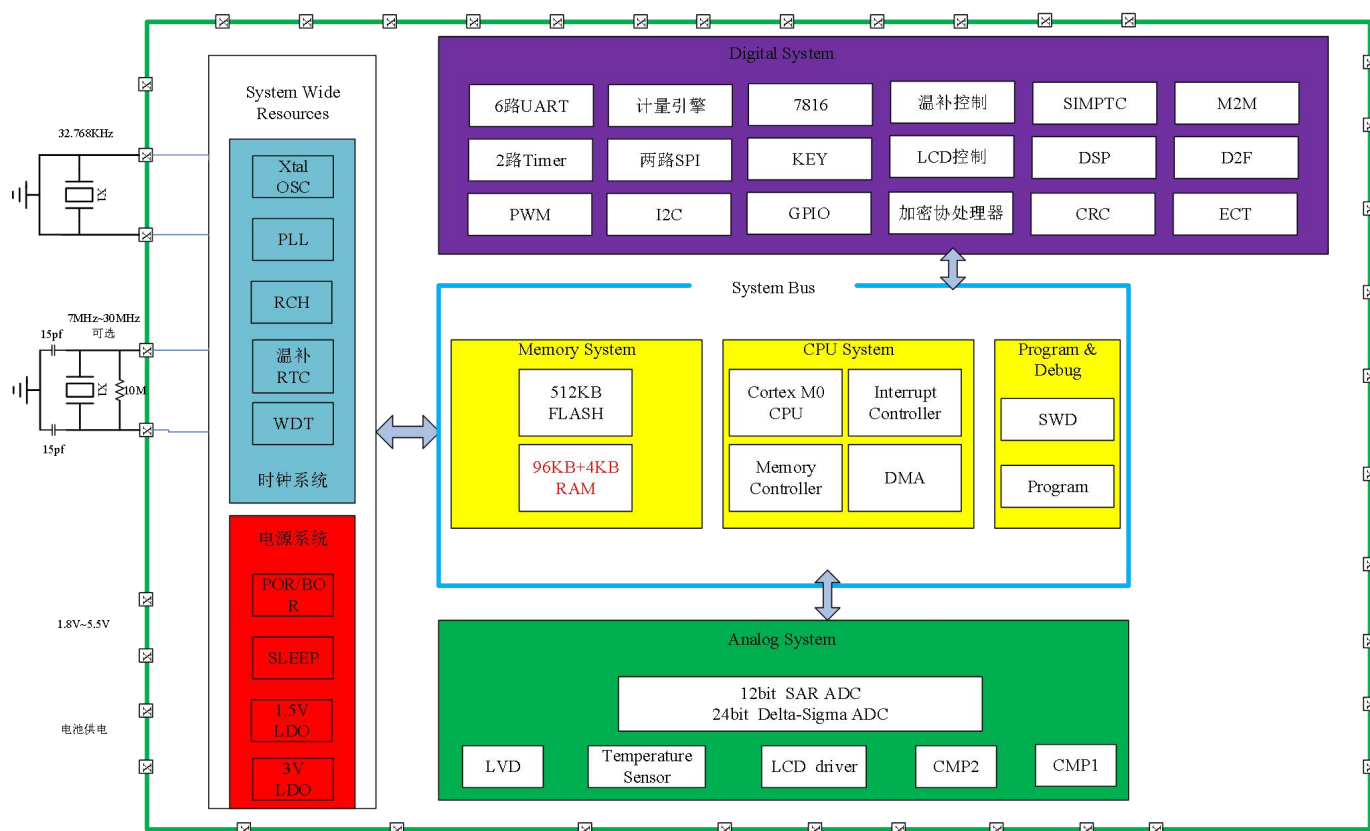


图 1.1 单相 SOC 系统框图

1.6 管脚说明

1.6.1 RN8213B LQFP128L 引脚图

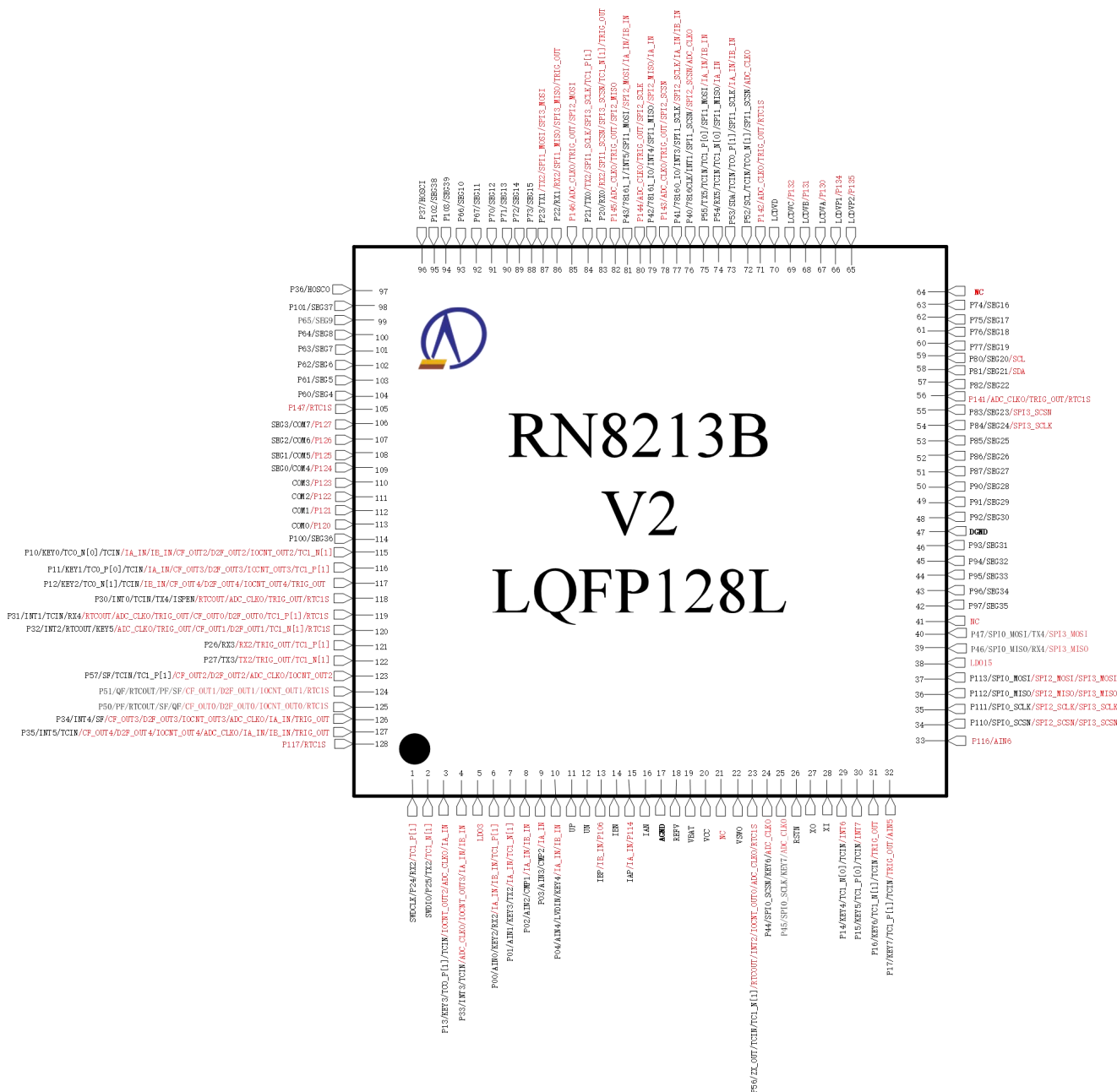


图 1-2 RN8213B 管脚排列图

PIN 脚差异说明:

RN8213B V2 新增 22 个 IO，其中 13 个与原 LCD 引脚复用，另外 9 个新增 IO 原来为地、电源或 NC 引脚，修改后默认向下兼容。

型号	PIN 脚	V1 版	V2 版	数量	说明
RN8213B	5	LDO33	LDO3	1	修改
	33	DGND	P116/AIN6	1	新增
	38	LDO18	LDO15	1	修改

	41	DGND	NC，要求悬空，不可接地	1	修改
	56	DGND	P141/ADC_CLKO/TRIG_OUT/RTC1S	1	新增
	64	DGND	NC 引脚，悬空	1	修改
	65~69	LCD	P135/P134,P130~P132	5	新增
	71	DGND	P142/ADC_CLKO/TRIG_OUT/RTC1S	1	新增
	78	NC	P143/ADC_CLKO/TRIG_OUT/SPI2_SCSN	1	新增
	80	DVDD	P144/ADC_CLKO/TRIG_OUT/SPI2_SCLK	1	新增
	82	DGND	P145/ADC_CLKO/TRIG_OUT/SPI2_MISO， 输出时仅支持开漏输出，使用时需要配置为 开漏模式	1	新增
	85	DGND	P146/ADC_CLKO/TRIG_OUT/SPI2_MOSI	1	新增
	105	DGND	P147/ADC_CLKO/TRIG_OUT/RTC1S	1	新增
	106~113	SEG/COM	P127~P120	8	新增
	128	DGND	P117/RTC1S	1	新增

1.6.2 RN8217/RN8215 LQFP100L 引脚图

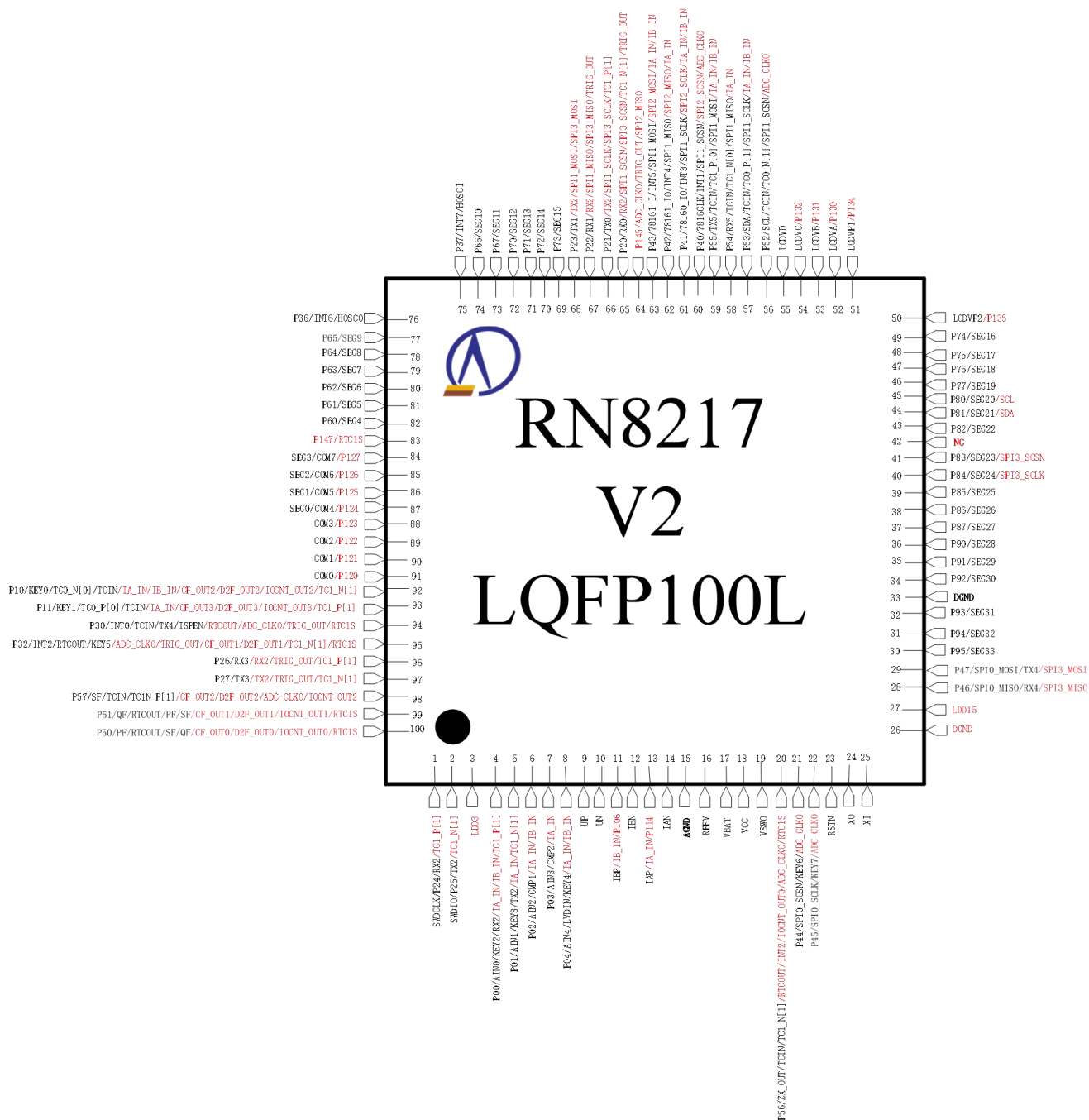


图 1-3 RN8217/RN8215 管脚排列图

PIN 脚差异说明:

RN8217/RN8215 V2 新增 15 个 IO，其中 13 个与原 LCD 引脚复用，另外 2 个新增 IO 原来为地引脚，修改后默认向下兼容。

型号	PIN 脚	V1 版	V2 版	数量	说明
RN8217	3	LDO33	LDO3	1	修改
RN8215	27	LDO18	LDO15	1	修改

42	DGND	NC 引脚，悬空	1	新增
50~54	LCD	P135/P134,P130~P132	5	新增
64	DGND	P145/ADC_CLKO/TRIG_OUT/SPI2_MISO， 输出时仅支持开漏输出，使用时需要配置为 开漏模式	1	新增
83	DGND	P147/ADC_CLKO/TRIG_OUT/RTC1S	1	新增
84~91	SEG/COM	P127~P120	8	新增

1.6.3 RN8213 LQFP100L 引脚图

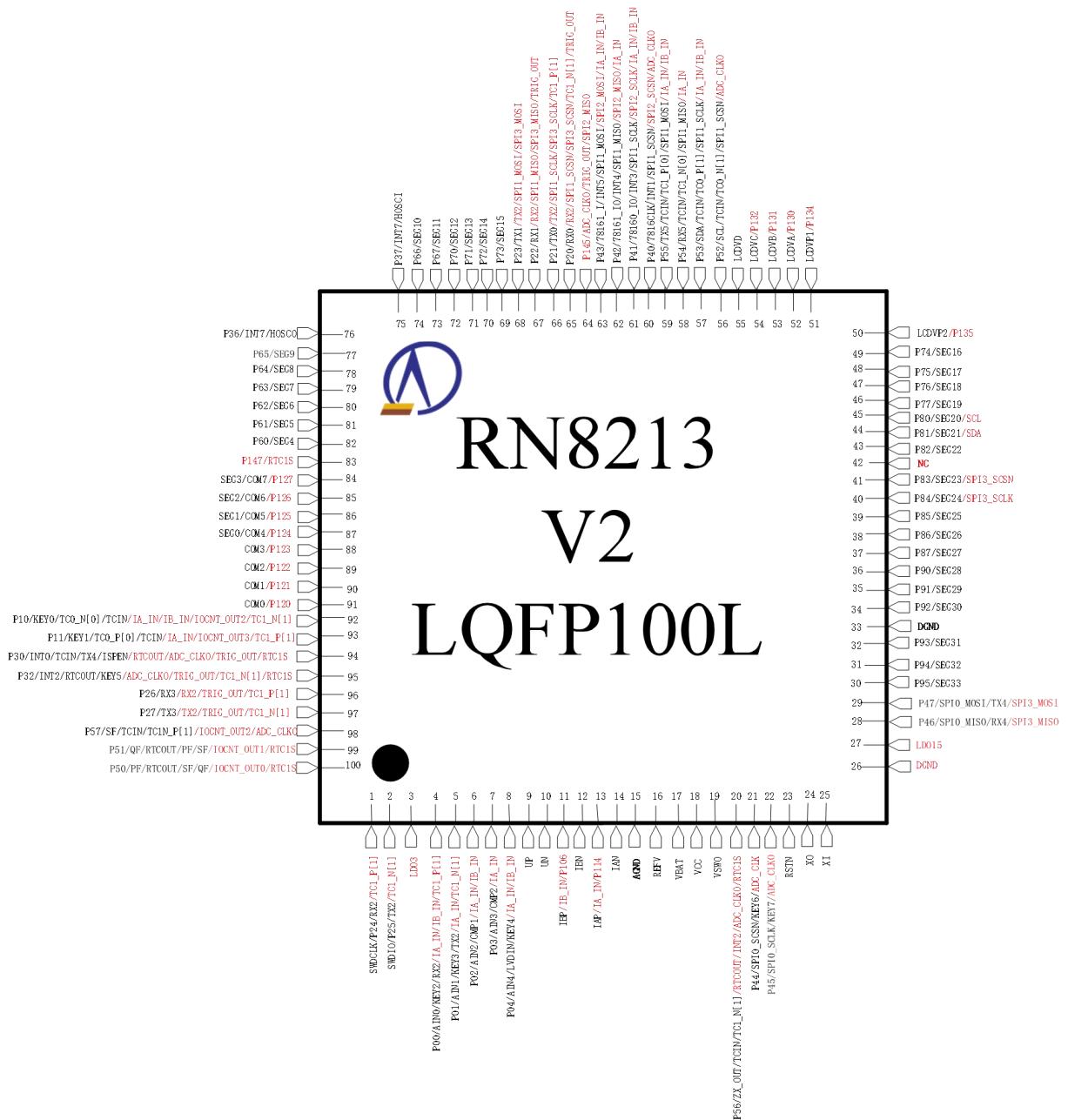


图 1-4 RN8213 管脚排列图

PIN 脚差异说明:

RN8213 V2 新增 15 个 IO，其中 13 个与原 LCD 引脚复用，另外 2 个新增 IO 原来为地引脚，修改后默认向下兼容。

型号	PIN 脚	V1 版	V2 版	数量	说明
RN8213	3	LDO33	LDO3	1	修改
	27	LDO18	LDO15	1	修改
	42	DGND	NC 引脚，悬空	1	新增
	50~54	LCD	P135/P134,P130~P132	5	新增
	64	DGND	P145/ADC_CLKO/TRIG_OUT/SPI2_MISO， 输出时仅支持开漏输出，使用时需要配置为 开漏模式	1	新增
	83	DGND	P147/ADC_CLKO/TRIG_OUT/RTC1S	1	新增
	84~91	SEG/COM	P127~P120	8	新增

1.6.4 RN7213 LQFP64L 引脚图

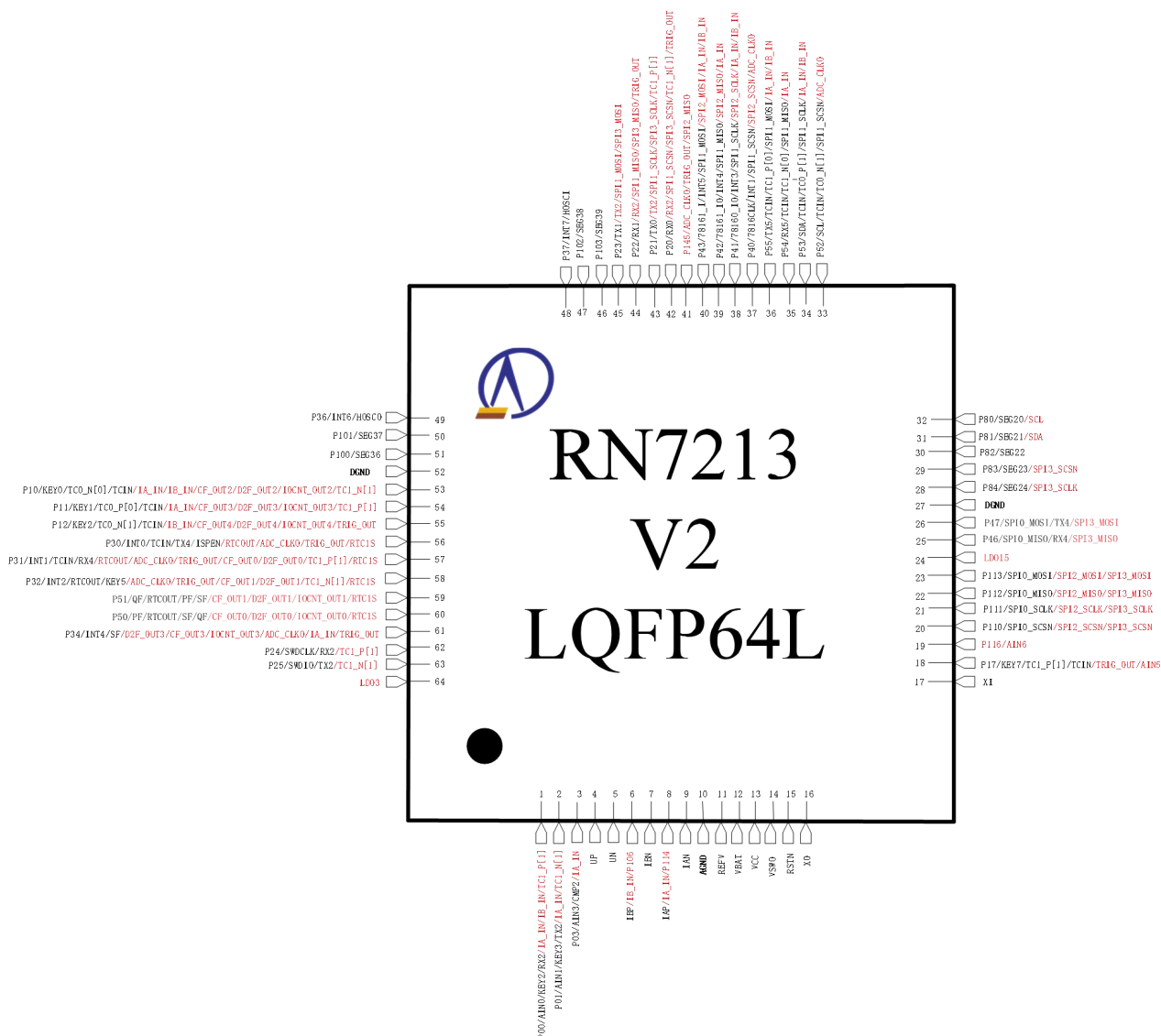


图 1-4 RN7213 管脚排列图

PIN 脚差异说明:

RN7213 V2 新增 2 个 IO， 2 个新增 IO 原来为地引脚，修改后默认向下兼容。

型号	PIN 脚	V1 版	V2 版	数量	说明
RN7213	19	DGND	P116/AIN6	1	新增
	24	LDO18	LDO15	1	修改
	41	DGND	P145/ADC_CLKO/TRIG_OUT/SPI2_MISO， 输出时仅支持开漏输出，使用时需要配置为 开漏模式	1	新增
	64	LDO33	LDO3	1	修改

1.6.5 RN8211B LQFP64L 引脚图

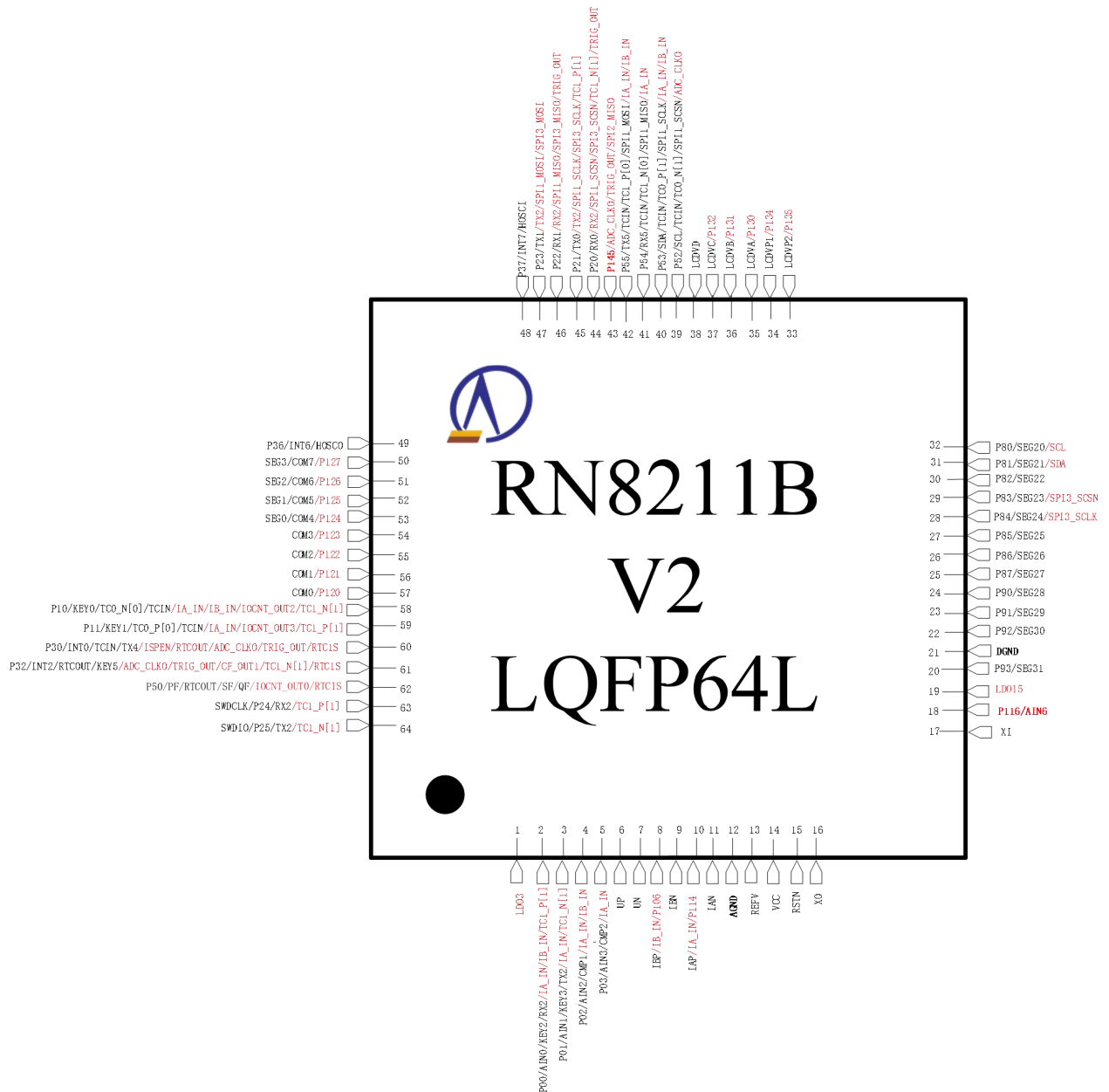


图 1-6 RN8211B 管脚排列图

PIN 脚差异说明:

RN8211B V2 新增 15 个 IO，其中 13 个与原 LCD 引脚复用，另外 2 个新增 IO 原来为地引脚，修改后默认向下兼容。

型号	PIN 脚	V1 版	V2 版	数量	说明
RN8211B	1	LDO33	LDO3	1	修改
	18	DGND	P116/AIN6	1	新增
	19	LDO18	LDO15	1	修改
	33~37	LCD	P135/P134,P130~P132	5	新增
	43	DGND	P145/ADC_CLK0/TRIG_OUT/SPI2_MISO, 输出时仅支持开漏输出，使用时需要配置为	1	新增

			开漏模式		
	50~57	SEG/COM	P127~P120	8	新增

1.6.6 引脚类型说明

类型	Cell Type				Input Option				Osc Option	Lcd Option	Adc Option	Output Option	
	模拟	双向	输入	输出	上拉	下拉	施密特	TTL/CMOS	晶振	SEG/COM	ADC	Open Drain	驱动
	A	B	I	O	U	D	S	L	X	G	C	D	
PABULD3	√	√			√			√				√	3mA
PABUS3	√	√			√		√						3mA
PBDS3		√				√	√					开漏	3mA
PBDSG3		√				√	√			√		开漏	3mA
PBULD3		√			√			√				√	3mA
PBULD6		√			√			√				√	6mA
PBUS6		√			√		√						6mA
PILC			√								√		
PID			√			√							
PIU			√		√								
PIUX			√		√				√				

1.6.7 管脚定义说明

RN8213B	RN8213 RN8215 RN8217	RN7213	RN8211B	Type	Pin Function	Descriptions
1	1	62	63	PBULD3	P24	GPIO
					SWDCLK	SWD 时钟口
					RX2	UART2 接收
					TC1_P[1]	定时器 1 通道 1 比较正向输出
2	2	63	64	PBULD3	P25	GPIO
					SWDIO	SWD 数据口
					TX2	UART2 发送
					TC1_N[1]	定时器 1 通道 1 比较反向输出
3				PBULD3	P13	GPIO
					KEY3	外部中断输入
					TC0_P[1]	定时器 0 通道 1 比较正向输出
					TCIN	定时器输入
					IOCNT_OUT2	脉冲转发输出 2
					ADC_CLKO	ADC 时钟输出

					IA_IN	IA 通道外灌 1bit 输入
4				PBUS6	P33	GPIO
					INT3	外部中断输入
					TCIN	定时器输入
					IOCNT_OUT3	脉冲转发输出 3
					ADC_CLKO	ADC 时钟输出
					IA_IN	IA 通道外灌 1bit 输入
					IB_IN	IB 通道外灌 1bit 输入
5	3	64	1	电源	LDO3	内置 3V LDO 的输出，给 ADC 供电，应外接 1uf 电容并联 0.1uf 电容去耦
6	4	1	2	PABUS3	P00	GPIO，建议不要进行高速翻转，如作为通讯口，否则影响测温性能
					AIN0	SAR-ADC 输入，输入电平不要超过芯片供电电压 AIN0 输入阻抗约 3M 建议外接 0.1uF 对地电容
					KEY2	按键输入
					RX2	UART2 接收
					IA_IN	IA 通道外灌 1bit 输入
					IB_IN	IB 通道外灌 1bit 输入
					TC1_P[1]	定时器 1 通道 1 比较正向输出
7	5	2	3	PABUS3	P01	GPIO，建议不要进行高速翻转，如作为通讯口，否则影响测温性能
					AIN1	SAR-ADC 输入，输入电平不要超过芯片供电电压 AIN1 输入阻抗约 3M 建议外接 0.1uF 对地电容
					KEY3	按键输入
					TX2	UART2 发送
					IA_IN	IA 通道外灌 1bit 输入
					TC1_N[1]	定时器 1 通道 1 比较反向输出
8	6		4	PABUS3	P02	GPIO，建议不要进行高速翻转，如作为通讯口，否则影响测温性能
					AIN2	SAR-ADC 输入，输入电平不要超过芯片供电电压 AIN2 输入阻抗默认配置下约 3M； 当 CMP1 的内部阻抗配置有效时，AIN2 也会有 600K 对地阻抗，此时需考虑该内阻影响或确保关闭

						内部阻抗
					CMP1	比较器 1 输入
					IA_IN	IA 通道外灌 1bit 输入
					IB_IN	IB 通道外灌 1bit 输入
9	7	3	5	PABUS3	P03	GPIO, 建议不要进行高速翻转, 如作为通讯口, 否则影响测温性能
					AIN3	SAR-ADC 输入, 输入电平不要超过芯片供电电压 AIN2 输入阻抗默认配置下内有 600K 对地电阻; 做 AIN 使用时, 需考虑该内阻影响或在系统掉电控制寄存器中关闭该对地电阻, 关闭对地电阻后输入阻抗约 3M 建议外接 0.1uF 对地电容
					CMP2	比较器 2 输入
					IA_IN	IA 通道外灌 1bit 输入
10	8			PABUS3	P04	GPIO, 建议不要进行高速翻转, 如作为通讯口, 否则影响测温性能
					AIN4	SAR-ADC 输入, 输入电平不要超过芯片供电电压 AIN4 输入阻抗约 3M 建议外接 0.1uF 对地电容
					LVDIN	掉电检测输入
					KEY4	按键输入
					IA_IN	IA 通道外灌 1bit 输入
					IB_IN	IB 通道外灌 1bit 输入
11	9	4	6	PILC	UP	电压通道正端输入
12	10	5	7	PILC	UN	电压通道负端输入
13	11	6	8	PILC	P106	不支持 GPIO 功能
					IB_IN	IB 通道外灌 1bit 输入
					IBP	电流通道 IB 正端输入
14	12	7	9	PILC	P107	不支持 GPIO 功能
					IBN	电流通道 IB 负端输入
15	13	8	10	PILC	IAP	电流通道 IA 正端输入
					IA_IN	IA 通道外灌 1bit 输入
					P114	不支持 GPIO 功能
16	14	9	11	PILC	IAN	电流通道 IA 负端输入
17	15	10	12	地	AGND	模拟地
18	16	11	13	电源	REFV	计量 ADC 的 1.25V 基准电压, 也可以接外灌基准; 应外接 1uf 电容并联 0.1uf 电容去耦。

19	17	12		电源	VBAT	电池或超级电容输入引脚，建议外接 0.1uF 滤波电容。
20	18	13	14	电源	VCC	主电电源输入，应外接 4.7uf 电容并联 0.1uf 电容去耦。
21				NC	NC	NC 引脚，可以悬空，也可以接地或接电源
22	19	14		电源	VSWO	VCC 与 VBAT 切换后的电源输出，给芯片供电，应外接 1uf 电容并联 0.1uf 电容去耦。
23	20			PBULD6	P56	GPIO，V2 版驱动能力 6mA，V1 版为 3mA
					RTCOUNT	RTCOUNT 输出
					TC1_N[1]	定时器 1 通道 1 比较反向输出
					TCIN	定时器输入
					ZX_OUT	电压过零输出
					INT2	外部中断输入
					IOCNT_OUT0	脉冲转发输出 0（RN8213 及 RN8211B 不具备该功能）
					ADC_CLKO	ADC 时钟输出
24	21			PBULD3	RTCIS	万年历 1HZ 秒脉冲输出
					P44	GPIO，V2 版驱动能力 3mA(default)/1.5mA 可配置，详细请参考 GPIO_IOCFG 寄存器
					KEY6	按键输入
					SPI0_SCSN	SPI 片选
25	22			PBULD3	ADC_CLKO	ADC 时钟输出
					P45	GPIO，V2 版驱动能力 3mA(default)/1.5mA 可配置，详细请参考 GPIO_IOCFG 寄存器
					KEY7	按键输入
					SPI0_CLK	SPI 时钟
26	23	15	15	PIU	RSTN	PIN 复位输入
27	24	16	16	时钟	XO	32.768KHz 无源晶振输出。
28	25	17	17	时钟	XI	32.768KHz 无源晶振输入。 XI/XO 不需要外接电阻和电容，需要用地线将之隔离。 建议选择负载电容为 12.5pF 的晶体。 X0 和 XI 引脚附近 IO 不使用高速翻转的功能，如作为通讯口。
29				PBULD3	P14	GPIO，V2 版驱动能力 3mA(default)/1.5mA 可配置，详

						细请参考 GPIO_IOCFG 寄存器；不能进行高速翻转，如作为通讯口，否则影响 LOSC 晶体性能
					KEY4	按键输入
					TC1_N[0]	定时器 1 通道 0 比较反向输出
					TCIN	定时器输入
					INT6	外部中断输入
30				PBULD3	P15	GPIO，V2 版驱动能力 3mA(default)/1.5mA 可配置，详细请参考 GPIO_IOCFG 寄存器
					KEY5	按键输入
					TC1_P[0]	定时器 1 通道 0 比较正向输出
					TCIN	定时器输入
					INT7	外部中断输入
31				PBULD3	P16	GPIO，V2 版驱动能力 3mA(default)/1.5mA 可配置，详细请参考 GPIO_IOCFG 寄存器
					KEY6	按键输入
					TC1_N[1]	定时器 1 通道 1 比较反向输出
					TCIN	定时器输入
					TRIG_OUT	智能微断脱扣输出信号
32		18		PABULD3	P17	GPIO，V2 版驱动能力 3mA(default)/1.5mA 可配置，详细请参考 GPIO_IOCFG 寄存器；不能进行高速翻转，如作为通讯口，否则影响测温性能
					KEY7	按键输入
					TC1_P[1]	定时器 1 通道 1 比较正向输出
					TCIN	定时器输入
					TRIG_OUT	智能微断脱扣输出信号
					AIN5	SAR-ADC 输入，输入电平不能超过芯片供电电压
33	26	19	18	PABULD3	P116	GPIO，V2 版新增 IO；不能进行高速翻转，如作为通讯口，否则影响测温性能 注：RN8217/RN8215/RN8213 该管脚为 DGND
					AIN6	SAR-ADC 输入，输入电平不能超过芯片供电电压 注：RN8217/RN8215/RN8213 该管脚为 DGND
34		20		PBULD3	P110	GPIO，V2 版驱动能力 3mA(default)/1.5mA 可配置，详

						细请参考 GPIO_IOCFG 寄存器
					SPI0_SCSN	SPI0 片选
					SPI2_SCSN	SPI2 片选
					SPI3_SCSN	SPI3 片选
35		21		PBULD3	P111	GPIO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					SPI0_SCLK	SPI0 时钟
					SPI2_SCLK	SPI2 时钟
					SPI3_SCLK	SPI3 时钟
36		22		PBULD3	P112	GPIO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					SPI0_MISO	SPI0 数据
					SPI2_MISO	SPI2 数据
					SPI3_MISO	SPI3 数据
37		23		PBULD3	P113	GPIO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					SPI0_MOSI	SPI0 数据
					SPI2_MOSI	SPI2 数据
					SPI3_MOSI	SPI3 数据
38	27	24	19	电源	LDO15	内置 1.5V LDO 输出, 给数字 1.5V 域供电, 外接 1uF+0.1uF 去耦电容
39	28	25		PBULD3	P46	GPIO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					SPI0_MISO	SPI0 数据
					RX4	UART4 接收
					SPI3_MISO	SPI3 数据
40	29	26		PBULD3	P47	GPIO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					SPI0_MOSI	SPI0 数据
					TX4	UART4 发送
					SPI3_MOSI	SPI3 数据
41				NC	NC	悬空, 不可接地
42				PBDSG3	P97	GPIO
					SEG35	LCD 驱动 SEG 端口
43				PBDSG3	P96	GPIO
					SEG34	LCD 驱动 SEG 端口
44	30			PBDSG3	P95	GPIO

					SEG33	LCD 驱动 SEG 端口
45	31			PBDSG3	P94	GPIO
					SEG32	LCD 驱动 SEG 端口
46	32		20	PBDSG3	P93	GPIO
					SEG31	LCD 驱动 SEG 端口
47	33	27	21	地	DGND	地
48	34		22	PBDSG3	P92	GPIO
					SEG30	LCD 驱动 SEG 端口
49	35		23	PBDSG3	P91	GPIO
					SEG29	LCD 驱动 SEG 端口
50	36		24	PBDSG3	P90	GPIO
					SEG28	LCD 驱动 SEG 端口
51	37		25	PBDSG3	P87	GPIO
					SEG27	LCD 驱动 SEG 端口
52	38		26	PBDSG3	P86	GPIO
					SEG26	LCD 驱动 SEG 端口
53	39		27	PBDSG3	P85	GPIO
					SEG25	LCD 驱动 SEG 端口
54	40	28	28	PBDSG3	P84	GPIO
					SEG24	LCD 驱动 SEG 端口
					SPI3_SCLK	SPI3 时钟
55	41	29	29	PBDSG3	P83	GPIO
					SEG23	LCD 驱动 SEG 端口
					SPI3_SCSN	SPI3 片选
56	42			NC	NC	悬空，不连接
57	43	30	30	PBDSG3	P82	GPIO
					SEG22	LCD 驱动 SEG 端口
58	44	31	31	PBDSG3	P81	GPIO
					SEG21	LCD 驱动 SEG 端口
					SDA	I2C 数据
59	45	32	32	PBDSG3	P80	GPIO
					SEG20	LCD 驱动 SEG 端口
					SCL	I2C 时钟
60	46			PBDSG3	P77	GPIO
					SEG19	LCD 驱动 SEG 端口
61	47			PBDSG3	P76	GPIO
					SEG18	LCD 驱动 SEG 端口
62	48			PBDSG3	P75	GPIO
					SEG17	LCD 驱动 SEG 端口
63	49			PBDSG3	P74	GPIO
					SEG16	LCD 驱动 SEG 端口
64				NC	NC	NC 引脚，悬空
65	50		33	PBDS3	P135	GPIO

						<p>V2 版引脚类型修改，可做 GPIO，V1 版不支持 GPIO 功能</p> <p>默认配置下为 LCD 引脚，复位后引脚默认接地，向下兼容 V1 版；无 LCD 应用时可用作 GPIO，考虑复位后默认接地，该引脚不能用于低驱动电路应用，例如，低电平驱动 LED 灯应用，当芯片发生复位时，会异常点亮 LED 灯。</p>
					LCDVP2	<p>模拟输出，管脚默认功能</p> <p>LCD 使用 PUMP 方式时，为模拟输出，LCDVP2 和 LCDVP1 之间应该连接一个 100nF 的电容。</p> <p>LCD 如果使用电阻分压方式，可悬空或接地。</p>
66	51		34	PBDS3	P134	<p>GPIO</p> <p>应用说明同 P135 描述</p>
					LCDVP1	<p>模拟输出</p> <p>应用方法见 LCDVP2 说明</p>
67	52		35	PBDS3	P130	<p>GPIO</p> <p>应用说明同 P135 描述</p>
					LCDVA	<p>LCD 电压输出，管脚默认功能，如果要用作其他功能，需通过软件配置。</p> <p>选择为该功能时，不管 LCD 选择是 PUMP 方式还是电阻串方式，都需要外接 470nF 电容</p>
68	53		36	PBDS3	P131	<p>GPIO</p> <p>应用说明同 P135 描述</p>
					LCDVB	<p>LCD 电压输出</p> <p>应用方法见 LCDVA 说明</p>
69	54		37	PBDS3	P132	<p>GPIO</p> <p>应用说明同 P135 描述</p>
					LCDVC	<p>LCD 电压输出</p> <p>应用方法见 LCDVA 说明</p>
70	55		38	PBDS3	LCDVD	<p>LCD 电压输出，需要外接 470nF 电容。不支持 GPIO 功能</p>
71				PBULD3	P142	<p>GPIO，V2 版新增 IO</p>
					ADC_CLKO	<p>ADC 时钟输出</p>
					TRIG_OUT	<p>智能微断脱扣输出信号</p>
					RTC1S	<p>万年历 1HZ 秒脉冲输出</p>
72	56	33	39	PBULD3	P52	<p>GPIO，V2 版驱动能力</p> <p>3mA(default)/1.5mA 可配置，详</p>

						细请参考 GPIO_IIOCFG 寄存器
					SCL	I2C 时钟
					TC0_N[1]	定时器 0 通道 1 比较反向输出
					TCIN	定时器输入
					SPI1_SCSN	SPI1 片选
					ADC_CLKO	ADC 时钟输出
73	57	34	40	PBULD3	P53	GPIO，V2 版驱动能力 3mA(default)/1.5mA 可配置，详细请参考 GPIO_IIOCFG 寄存器
					SDA	I2C 数据
					TC0_P[1]	定时器 0 通道 1 比较正向输出
					TCIN	定时器输入
					SPI1_SCLK	SPI1 时钟
					IA_IN	IA 通道外灌 1bit 输入
					IB_IN	IB 通道外灌 1bit 输入
74	58	35	41	PBULD3	P54	GPIO，V2 版驱动能力 3mA(default)/1.5mA 可配置，详细请参考 GPIO_IIOCFG 寄存器
					RX5	UART5 接收
					TC1_N[0]	定时器 1 通道 0 比较反向输出
					TCIN	定时器输入
					SPI1_MISO	SPI1 数据
					IA_IN	IA 通道外灌 1bit 输入
75	59	36	42	PBULD3	P55	GPIO，V2 版驱动能力 3mA(default)/1.5mA 可配置，详细请参考 GPIO_IIOCFG 寄存器
					TX5	UART5 发送
					TC1_P[0]	定时器 1 通道 0 比较正向输出
					SPI1_MOSI	SPI1 数据
					TCIN	定时器输入
					IA_IN	IA 通道外灌 1bit 输入
					IB_IN	IB 通道外灌 1bit 输入
76	60	37		PBULD3	P40	GPIO，V2 版驱动能力 3mA(default)/1.5mA 可配置，详细请参考 GPIO_IIOCFG 寄存器
					7816CLK	7816 时钟输出
					INT1	外部中断输入
					SPI1_SCSN	SPI 片选
					SPI2_SCSN	SPI 片选
					ADC_CLKO	ADC 时钟输出
77	61	38		PBULD3	P41	GPIO，V2 版驱动能力 3mA(default)/1.5mA 可配置，详

						细请参考 GPIO_IOCFG 寄存器
					78160_IO	7816 0 双向数据口
					INT3	外部中断输入
					SPI1_SCLK	SPI 时钟
					SPI2_SCLK	SPI 时钟
					IA_IN	IA 通道外灌 1bit 输入
					IB_IN	IB 通道外灌 1bit 输入
78				PBULD3	P143	GPIO, V2 版新增 IO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					ADC_CLKO	ADC 时钟输出
					TRIG_OUT	智能微断脱扣输出信号
					SPI2_SCSN	SPI 片选
79	62	39		PBULD3	P42	GPIO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					78161_IO	7816 1 双向数据口
					INT4	外部中断输入
					SPI1_MISO	SPI 数据
					SPI2_MISO	SPI 数据
					IA_IN	IA 通道外灌 1bit 输入
80				PBULD3	P144	GPIO, V2 版新增 IO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					ADC_CLKO	ADC 时钟输出
					TRIG_OUT	智能微断脱扣输出信号
					SPI2_SCLK	SPI 时钟
81	63	40		PBULD3	P43	GPIO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					78161_I	7816 1 数据输入
					INT5	外部中断输入
					SPI1_MOSI	SPI 数据
					SPI2_MOSI	SPI 数据
					IA_IN	IA 通道外灌 1bit 输入
					IB_IN	IB 通道外灌 1bit 输入
82	64	41	43	PBDS3	P145	GPIO, V2 版新增 IO, 输出时仅支持开漏输出, 使用时需要配置为开漏模式, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器

					ADC_CLKO	ADC 时钟输出
					TRIG_OUT	智能微断脱扣输出信号
					SPI2_MISO	SPI 数据
83	65	42	44	PBULD3	P20	GPIO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					RX0	UART0 接收
					RX2	UART2 接收
					SPI1_SCSN	SPI 片选
					SPI3_SCSN	SPI 片选
					TC1_N[1]	定时器 1 通道 1 比较反向输出
					TRIG_OUT	智能微断脱扣输出信号
84	66	43	45	PBULD3	P21	GPIO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					TX0	UART0 发送
					TX2	UART2 发送
					SPI1_SCLK	SPI 时钟
					SPI3_SCLK	SPI 时钟
85				PBULD3	P146	GPIO, V2 版新增 IO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					ADC_CLKO	ADC 时钟输出
					TRIG_OUT	智能微断脱扣输出信号
					SPI2_MOSI	SPI 数据
86	67	44	46	PBULD3	P22	GPIO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					RX1	UART1 接收
					RX2	UART2 接收
					SPI1_MISO	SPI 数据
					SPI3_MISO	SPI 数据
87	68	45	47	PBULD3	TRIG_OUT	智能微断脱扣输出信号
					P23	GPIO, V2 版驱动能力 3mA(default)/1.5mA 可配置, 详细请参考 GPIO_IOCFG 寄存器
					TX1	UART1 发送
					TX2	UART2 发送
					SPI1_MOSI	SPI 数据
88	69			PBDSG3	SPI3_MOSI	SPI 数据
					P73	GPIO

					SEG15	LCD 驱动 SEG 端口
89	70			PBDSG3	P72	GPIO
					SEG14	LCD 驱动 SEG 端口
90	71			PBDSG3	P71	GPIO
					SEG13	LCD 驱动 SEG 端口
91	72			PBDSG3	P70	GPIO
					SEG12	LCD 驱动 SEG 端口
92	73			PBDSG3	P67	GPIO
					SEG11	LCD 驱动 SEG 端口
93	74			PBDSG3	P66	GPIO
					SEG10	LCD 驱动 SEG 端口
94		46		PBDSG3	P103	GPIO
					SEG39	LCD 驱动 SEG 端口
95		47		PBDSG3	P102	GPIO
					SEG38	LCD 驱动 SEG 端口
96	75	48	48	PIUX	P37	GPIO
					INT7	外部中断输入
					HOSCI	高频晶体输入口，与 HOSCO 之间应串接一个 10M 欧的电阻，并联 15pf 的电容到地。
97	76	49	49	PIUX	P36	GPIO
					INT6	外部中断输入
					HOSCO	高频晶体输出口，与 HOSCI 之间应串接一个 10M 欧的电阻，并联 15pf 的电容到地。
98		50		PBDSG3	P101	GPIO
					SEG37	LCD 驱动 SEG 端口
99	77			PBDSG3	P65	GPIO
					SEG9	LCD 驱动 SEG 端口
100	78			PBDSG3	P64	GPIO
					SEG8	LCD 驱动 SEG 端口
101	79			PBDSG3	P63	GPIO
					SEG7	LCD 驱动 SEG 端口
102	80			PBDSG3	P62	GPIO
					SEG6	LCD 驱动 SEG 端口
103	81			PBDSG3	P61	GPIO
					SEG5	LCD 驱动 SEG 端口
104	82			PBDSG3	P60	GPIO
					SEG4	LCD 驱动 SEG 端口
105	83			PBULD3	P147	GPIO, V2 版新增 IO
					RTC1S	万年历 1HZ 秒脉冲输出
106	84		50	PBDSG3	P127	GPIO, V2 版新增 IO
					SEG3	LCD 驱动 SEG 端口

					COM7	LCD 驱动 COM 端口
107	85		51	PBDSG3	P126	GPIO, V2 版新增 IO
					SEG2	LCD 驱动 SEG 端口
					COM6	LCD 驱动 COM 端口
108	86		52	PBDSG3	P125	GPIO, V2 版新增 IO
					SEG1	LCD 驱动 SEG 端口
					COM5	LCD 驱动 COM 端口
109	87		53	PBDSG3	P124	GPIO, V2 版新增 IO
					SEG0	LCD 驱动 SEG 端口
					COM4	LCD 驱动 COM 端口
110	88		54	PBDSG3	P123	GPIO, V2 版新增 IO
					COM3	LCD 驱动 COM 端口
111	89		55	PBDSG3	P122	GPIO, V2 版新增 IO
					COM2	LCD 驱动 COM 端口
112	90		56	PBDSG3	P121	GPIO, V2 版新增 IO
					COM1	LCD 驱动 COM 端口
113	91		57	PBDSG3	P120	GPIO, V2 版新增 IO
					COM0	LCD 驱动 COM 端口
114		51		PBDSG3	P100	GPIO
					SEG36	LCD 驱动 SEG 端口
		52		地	DGND	地
115	92	53	58	PBULD3	P10	GPIO
					KEY0	按键输入
					TC0_N[0]	定时器 0 通道 0 比较反向输出
					TCIN	定时器输入
					IA_IN	IA 通道外灌 1bit 输入
					IB_IN	IB 通道外灌 1bit 输入
					CF_OUT2	计量脉冲输出 (RN8213 和 RN8211B 没有该功能)
					D2F_OUT2	D2F 的脉冲输出 (RN8213 和 RN8211B 没有该功能)
					IOCNT_OUT2	脉冲转发输出
					TC1_N[1]	定时器 1 通道 1 比较反向输出
116	93	54	59	PBULD3	P11	GPIO
					KEY1	按键输入
					TC0_P[0]	定时器 0 通道 0 比较正向输出
					TCIN	定时器输入
					IA_IN	IA 通道外灌 1bit 输入
					CF_OUT3	计量脉冲输出 (RN8213 和 RN8211B 没有该功能)
					D2F_OUT3	D2F 的脉冲输出 (RN8213 和 RN8211B 没有该功能)
					IOCNT_OUT3	脉冲转发输出

					TC1_P[1]	定时器 1 通道 1 比较正向输出
					P12	GPIO
					KEY2	按键输入
					TC0_N[1]	定时器 0 通道 1 比较反向输出
					TCIN	定时器输入
117		55		PBULD3	IB_IN	IB 通道外灌 1bit 输入
					CF_OUT4	计量脉冲输出
					D2F_OUT4	D2F 的脉冲输出
					IOCNT_OUT4	脉冲转发输出
					TRIG_OUT	智能微断脱扣输出信号
					P30	GPIO，同时也是 ISPEN 检测引脚，芯片发生复位后，会检测该端口的状态，如果输入低电平，系统会进入 ISP 模式，不会引导客户程序。硬件和软件设计注意规避该引脚在复位过程处于低电平状态。
118	94	56	60	PBUS6	INT0	外部中断输入
					TX4	UART4 发送
					TCIN	定时器输入
					RTCOUT	秒脉冲输出
					ADC_CLKO	ADC 时钟输出
					TRIG_OUT	智能微断脱扣输出信号
					RTC1S	万年历 1HZ 秒脉冲输出
					P31	GPIO
					INT1	外部中断输入
					RX4	UART4 接收
					TCIN	定时器输入
					RTCOUT	秒脉冲输出
					ADC_CLKO	ADC 时钟输出
					TRIG_OUT	智能微断脱扣输出信号
					CF_OUT0	计量脉冲输出
					D2F_OUT0	D2F 的脉冲输出
					TC1_P[1]	定时器 1 通道 1 比较正向输出
					RTC1S	万年历 1HZ 秒脉冲输出
					P32	GPIO
					INT2	外部中断输入
					RTCOUT	RTCOUT 输出
					KEY5	按键输入
					ADC_CLKO	ADC 时钟输出
					TRIG_OUT	智能微断脱扣输出信号
					CF_OUT1	计量脉冲输出（RN8213 和 RN8211B 没有该功能）
120	95	58	61	PBUS6		

					D2F_OUT1	D2F 的脉冲输出（RN8213 和 RN8211B 没有该功能）
					TC1_N[1]	定时器 1 通道 1 比较反向输出
					RTC1S	万年历 1HZ 秒脉冲输出
121	96			PBULD3	P26	GPIO
					RX3	UART3 接收
					RX2	UART2 接收
					TRIG_OUT	智能微断脱扣输出信号
					TC1_P[1]	定时器 1 通道 1 比较正向输出
122	97			PBULD3	P27	GPIO
					TX3	UART3 发送
					TX2	UART2 发送
					TRIG_OUT	智能微断脱扣输出信号
					TC1_N[1]	定时器 1 通道 1 比较反向输出
123	98			PBULD6	P57	GPIO, V2 版驱动能力 6mA, V1 版为 3mA
					TC1_P[1]	定时器 1 通道 1 比较正向输出
					TCIN	定时器输入
					SF	视在脉冲输出
					CF_OUT2	计量脉冲输出（RN8213 和 RN8211B 没有该功能）
					D2F_OUT2	D2F 的脉冲输出（RN8213 和 RN8211B 没有该功能）
					IOCNT_OUT2	脉冲转发输出
					ADC_CLKO	ADC 时钟输出
124	99	59		PBUS6	P51	GPIO
					QF	无功脉冲输出
					RTCOUT	RTCOUT 输出
					PF	有功脉冲输出
					SF	视在脉冲输出
					CF_OUT1	计量脉冲输出（RN8213 和 RN8211B 没有该功能）
					D2F_OUT1	D2F 的脉冲输出（RN8213 和 RN8211B 没有该功能）
					IOCNT_OUT1	脉冲转发输出
					RTC1S	万年历 1HZ 秒脉冲输出
125	100	60	62	PBUS6	P50	GPIO
					PF	有功脉冲输出
					RTCOUT	RTCOUT 输出
					SF	视在脉冲输出
					QF	无功脉冲输出
					CF_OUT0	计量脉冲输出（RN8213 和 RN8211B 没有该功能）

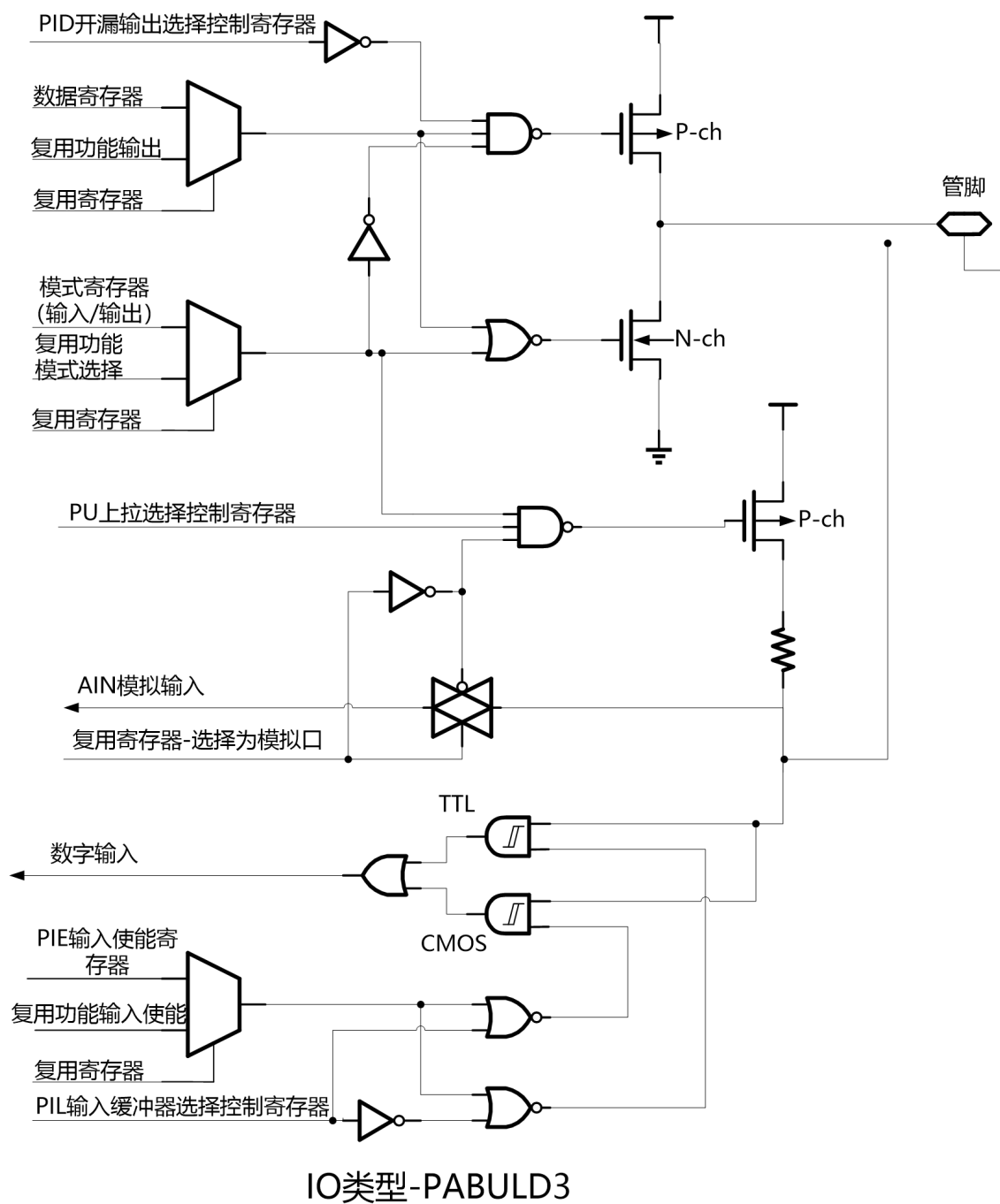
					D2F_OUT0	D2F 的脉冲输出（RN8213 和 RN8211B 没有该功能）
					IOCNT_OUT0	脉冲转发输出
					RTC1S	万年历 1HZ 秒脉冲输出
126		61		PBUS6	P34	GPIO
					INT4	外部中断输入
					SF	视在脉冲输出
					CF_OUT3	计量脉冲输出
					D2F_OUT3	D2F 的脉冲输出
					IOCNT_OUT3	脉冲转发输出
					ADC_CLKO	ADC 时钟输出
					IA_IN	IA 通道外灌 1bit 输入
					TRIG_OUT	智能微断脱扣输出信号
127	NC	NC	NC	PBUS6	P35	GPIO
					INT5	外部中断输入
					TCIN	定时器输入
					CF_OUT4	计量脉冲输出
					D2F_OUT4	D2F 的脉冲输出
					IOCNT_OUT4	脉冲转发输出
					ADC_CLKO	ADC 时钟输出
					IA_IN	IA 通道外灌 1bit 输入
					IB_IN	IB 通道外灌 1bit 输入
					TRIG_OUT	智能微断脱扣输出信号
128	NC	NC	NC	PBULD3	P117	GPIO, V2 版新增 IO
					RTC1S	万年历 1HZ 秒脉冲输出

1.6.8 硬件设计规则

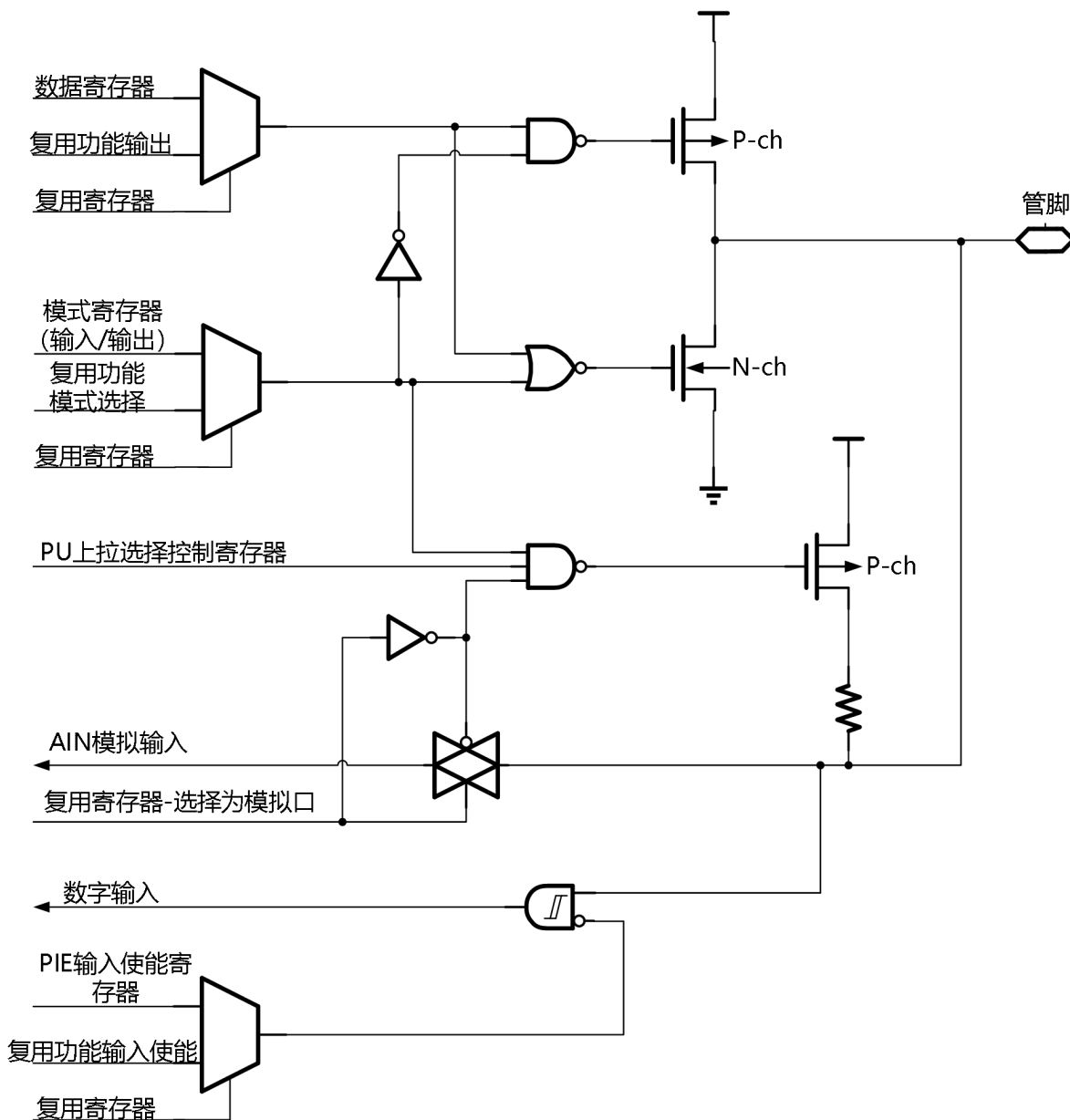
- 1.LDO3是3V LDO的输出，给计量ADC提供电源；外部应并接0.1uf和1uf电容；
- 2.UP、UN、IBP、IBN、IAP、IAN采用计量芯片常规接法；
- 3.AGND是模拟地，DGND是数字地；实际使用时可根据经验合理布局；
- 4.REFV是计量ADC的参考输入，外部应并接0.1uf和1uf电容；
- 5.VBAT是3.6V电池输入；
- 6.VCC是主电输入，正常工作范围2.8V~5.5V，应外接4.7uf电容并联0.1uf电容去耦；
- 7.VSWO是VCC与VBAT切换后的电源输出，应外接1uf电容并联0.1uf电容去耦；
- 8.XO和XI之间跨接32.768KHz晶体，最好用地线将之隔离，不需外接电阻和电容；且其附近IO不要使用高速翻转的功能，如作为通讯口，防止晶振受到影响；
- 9.LDO18是1.8V LDO的输出，给芯片1.8V数字域供电，应外接1uf电容并联0.1uf电容去耦；
- 10.LCDVD、LCDVC、LCDVB、LCDVA做LCD电压输出时，每个管脚都应外接470nf电容；做普通IO使用后，液晶驱动电压无法输出，需要液晶驱动的应用，该管脚不能做普通IO使用。
- 11.当使用电荷泵时，LCDVP1 和 LCDVP2 之间应连接一个 100nf 的电容；当使用电阻串分压时，这两个管脚可以悬空。做 IO 使用后，液晶驱动电压无法输出，需要液晶驱动的应用，该管脚不能做 GPIO 使用。

1.7 IO 口功能框图

1.7.1 PABULD3

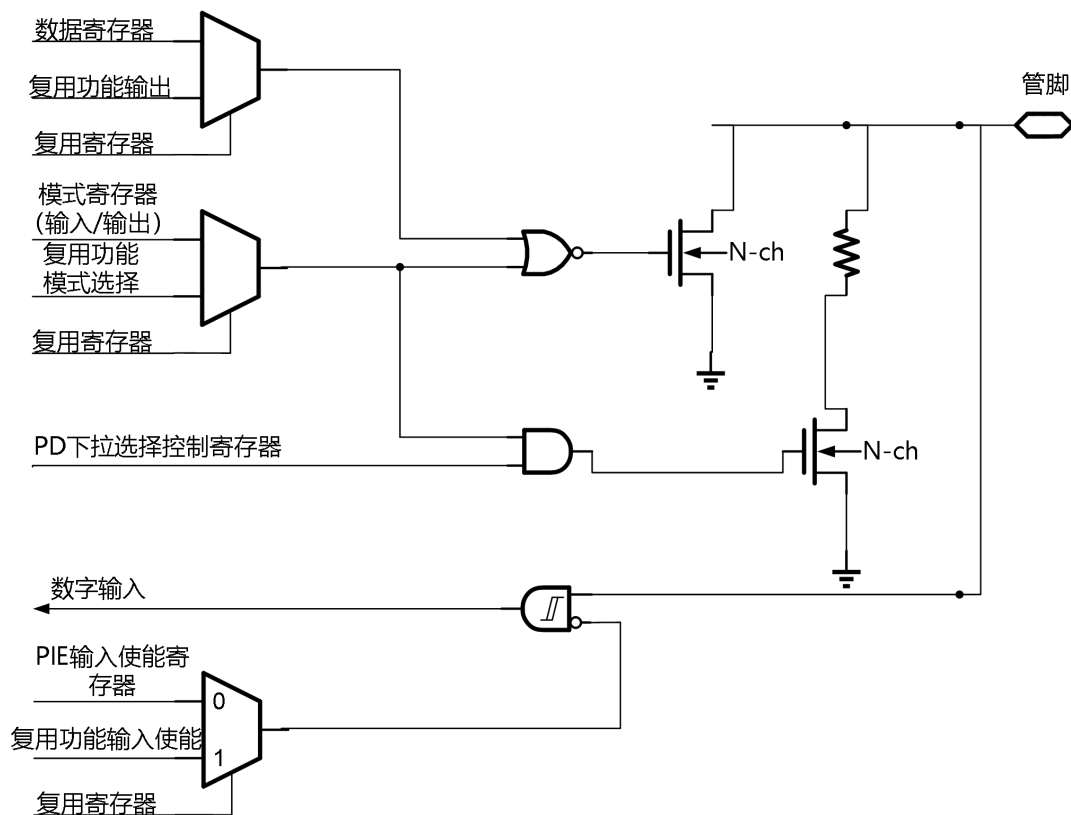


1.7.2 PABUS3



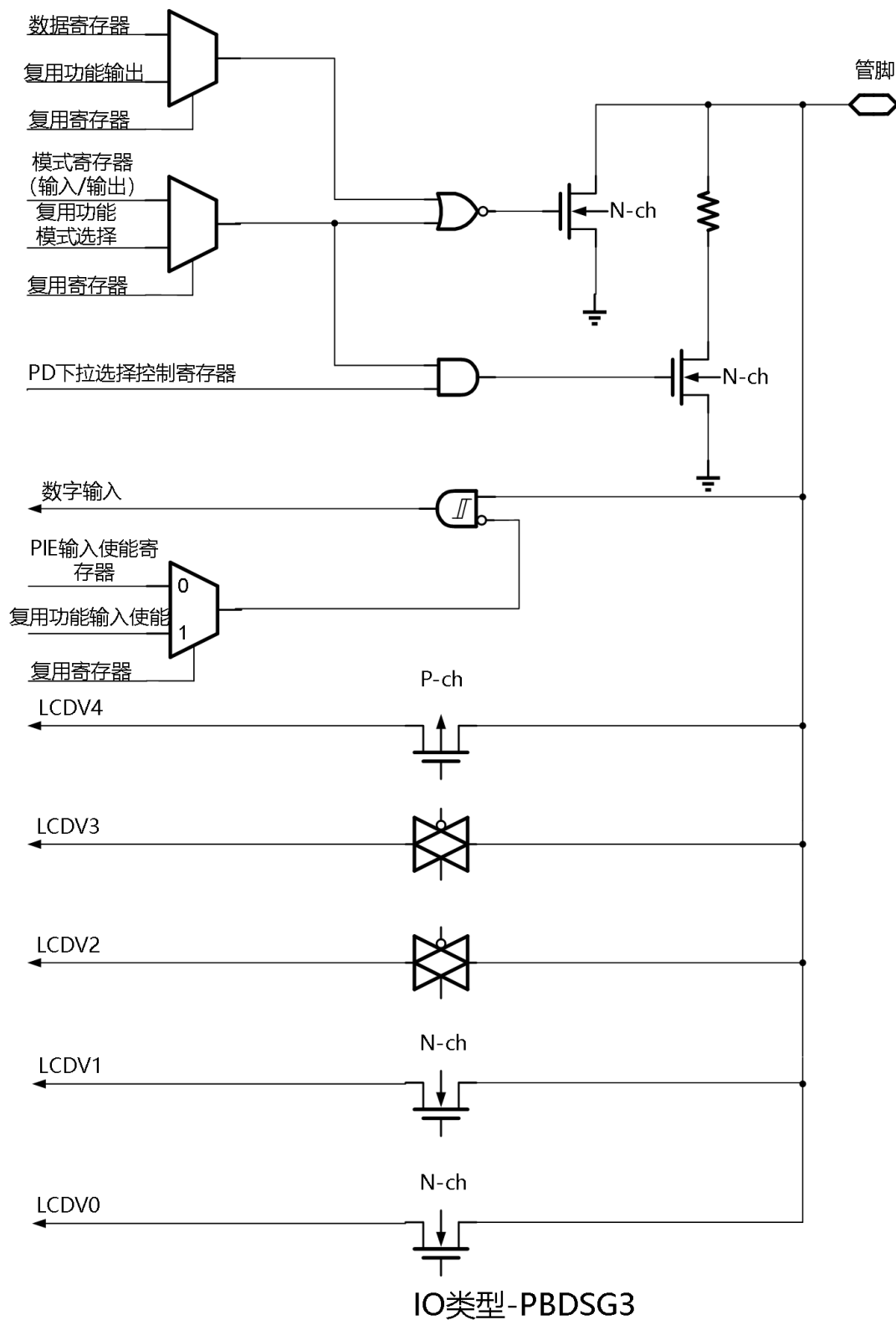
IO类型-PABUS3

1.7.3 PBDS3

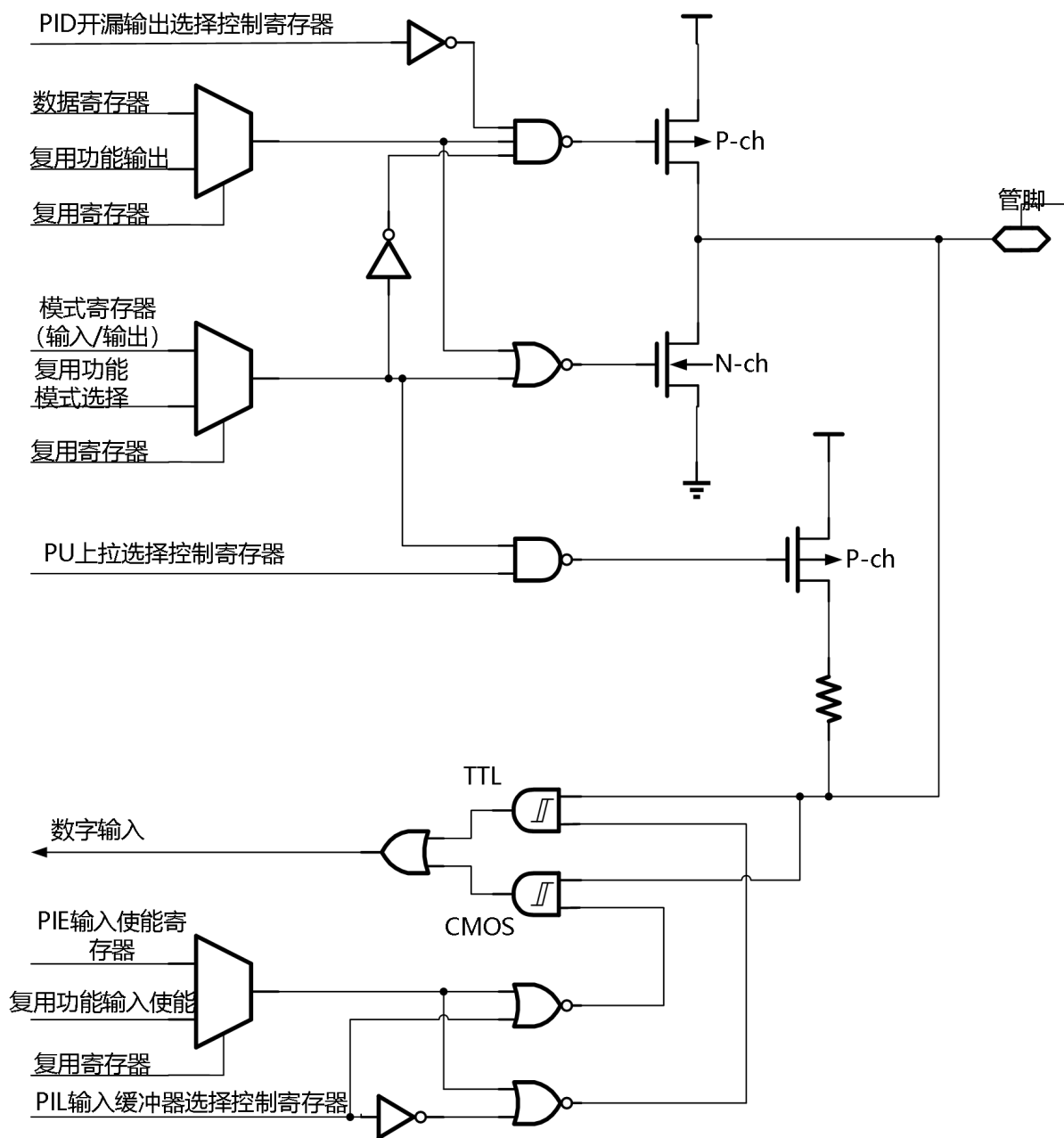


IO类型-PBDS3

1.7.4 PBDSG3

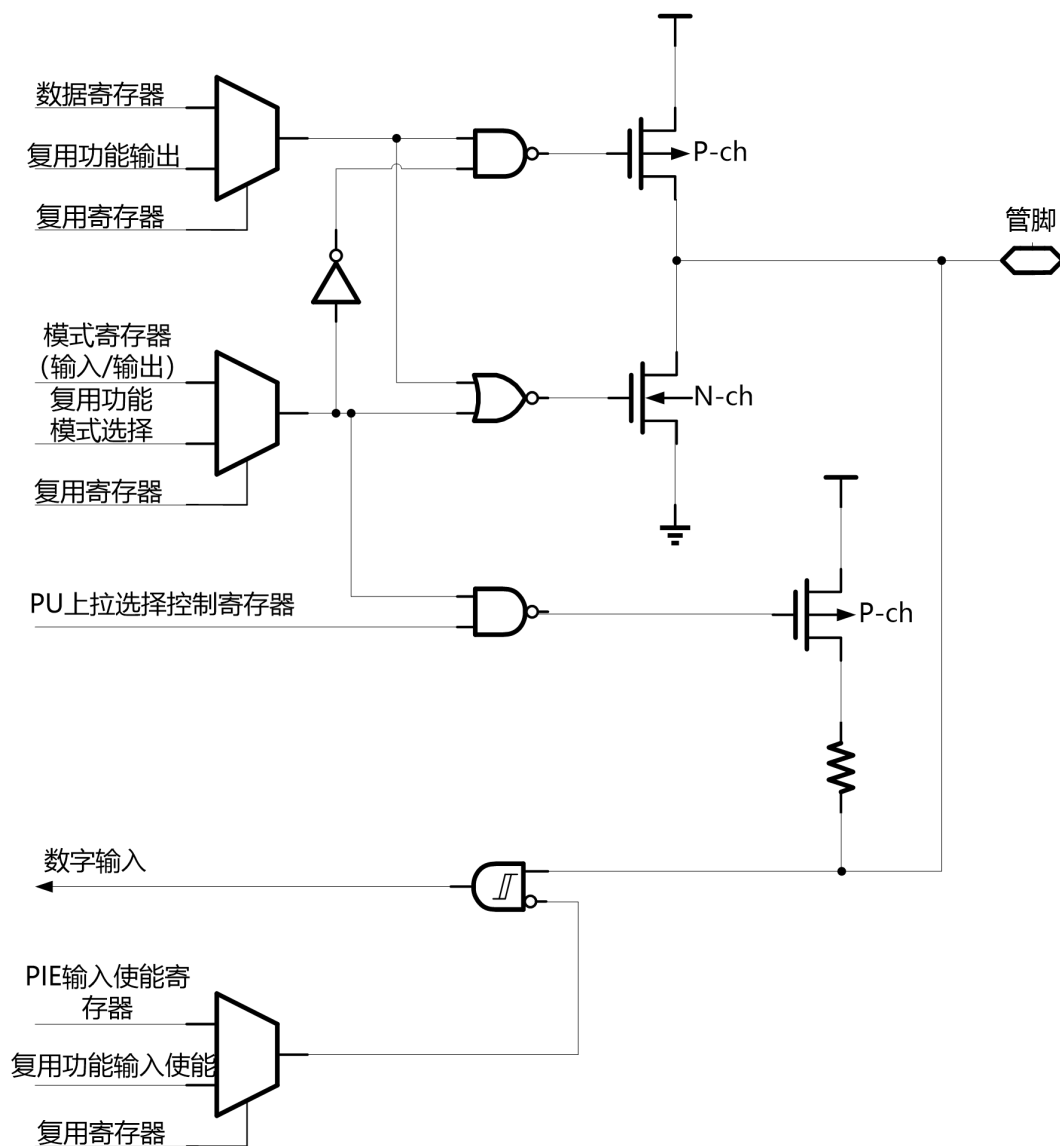


1.7.5 PBULD3/6



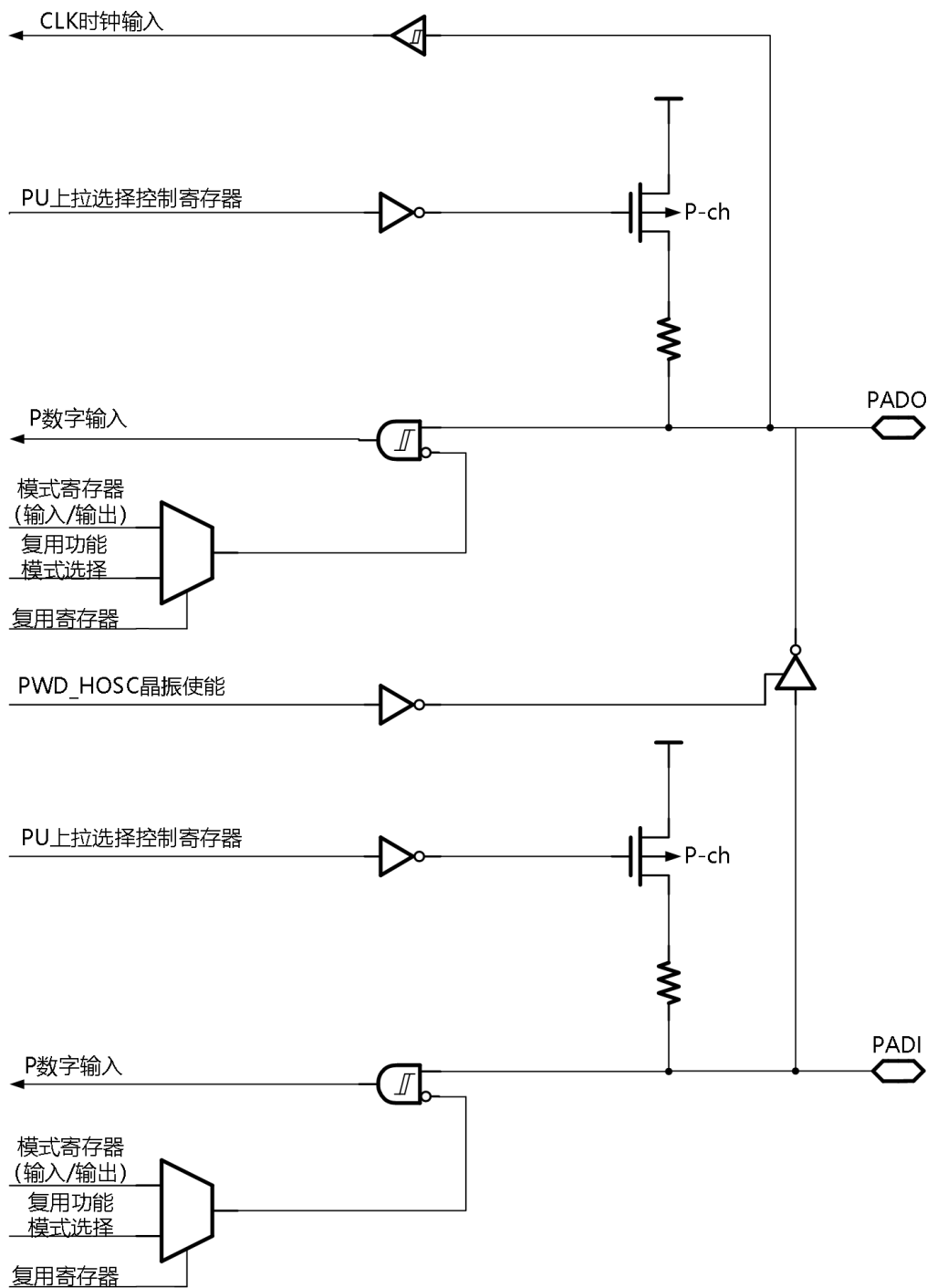
IO类型-PBULD3/PBULD6

1.7.6 PBUS6



IO类型-PBUS6

1.7.7 PIUX



IO类型-PIUX

2 电气特性

2.1 性能参数

计量参数 (VCC=3V~5.5V, 室温)						
测量项目	符号	最小	典型	最大	单位	测试条件和注释
有功电能测量误差	Err			±0.1%		常温 8000:1 的动态范围; -40℃ ~ +85℃ 内温度系数典型值 5ppm, 最大 15ppm;
有功电能测量带宽	BW		7		kHz	
无功电能测量误差	Err			±0.1%		
视在电能测量误差	Err			±0.1%		
有效值测量误差	Err			±0.2%		常温 1000:1 的动态范围; -40℃ ~ +85℃ 内温度系数典型值 5ppm, 最大 15ppm;
功率测量误差	Err			±0.1%		
电能脉冲输出	最大频率			20KHz	Hz	
	占空比		50%		%	当脉宽低于 84ms 时, 等 duty 输出
	高电平脉宽		84ms		ms	
Sigma-Delta ADC性能						
最大信号电平	V _{xn}			±1000	mV	差分后信号
ADC 失调误差	DC _{off}		1		mV	
-3dB 带宽	B _{-3dB}		7		kHz	
基准电压(VCC=3V~5.5V, 温度范围: -40℃~+85℃)						
输出电压REFV	V _{ref}	1.24	1.25	1.26	V	
温度系数	T _c		5	15	ppm/℃	
模拟外设(温度范围: -40℃~+85℃)						
低功耗比较器 CMP1/CMP2/LVDIN 阈值 1	V _{il1}	1.23	1.28	1.33	V	默认配置下, 该阈值为比较器输出低电平比较结果阈值;
	V _{ih1}	1.43	1.48	1.53	V	默认配置下, 该阈值为比较器输出高电平比较结果阈值;
	迟滞		200		mV	V _{ih} -V _{il}
低功耗比较器 CMP1/CMP2 阈值 2	V _{il2}	0.8	0.84	0.88	V	当选择 0.9V 档并且选择有迟滞时, 该阈值为比较器输出低电平比较结果阈值;
	V _{ih2}	0.94	0.98	0.102	V	当选择 0.9V 档并且选择有迟滞时, 该阈值为比较器输出高电平比较结果阈值;
	迟滞		140		mV	V _{ih} -V _{il}
SAR ADC 输入范	SAR-IN	0		REFV	V	REFV 为内部低功耗基

围						准，典型值为 1.25V
切换到主电阈值	上电切换阈值	2.6	2.8	3.0	V	当主电(VCC)高于该阈值或者高于 VBAT 时供电切换到 VCC。 建议主电选择为3.3V时，需要保证电源电压范围为 3.3V±5%。
切换到电池阈值	掉电切换阈值	2.5	2.7	2.9	V	当主电(VCC)低于该阈值并且低于 VBAT 时供电切换到 VBAT。
LCD 输出电压	LCDVD	4.85	5.05	5.25	V	全温度范围测试
VBAT 测量	VBATD	0	3.6	3.8	V	SAR ADC 对 VBAT 的测量范围；
时钟参数(温度范围：-40℃~+85℃)						
输入低频时钟频率范围	XI		32.768		KHz	
输入高频时钟频率范围	HOSI	7.3728	14.7456	29.4912	Mhz	
内部PLL时钟频率范围	PLL		14.7456	29.4912	MHz	
内部高频RCH	RCH		29.5		MHz	用于芯片复位后默认时钟，分频后频率为1.8M。 RCH全温度范围误差<±1%
内部低频RCL	RCL	20	32	40	KHz	用于WDT时钟
电源						
主电源	VCC	2.8	5/3.3	5.5	V	
cpu 最低工作电压	Vil	1.8	1.85	2	V	常温
	Vil	1.8	1.87	2	V	低温-40 度
	Vil	1.8	1.87	2	V	高温 85 度
电池	VBAT		3.6		V	
模拟电流	AId		1.8		mA	三路ADC均开启
数字电流	DId		3.5		mA	CPU运行在7.3728MHz，三路计量开启
数字电流	DId		4.5		mA	CPU运行在14.7456MHz，三路计量开启
数字电流	DId		6		mA	CPU运行在29.4912MHz，三路计量开启
休眠功耗	SId		8		μ A	RTC自动温补； RAM保持； CPU及数字外设不掉电； WDT开启；

						电源监测开启; 中断唤醒
LDO3	V33	2.9	3.0	3.1	V	
LDO15	V1P5	1.35	1.5	1.65	V	
极限参数						
主电电压	V _{vcc}	-0.3	--	+7	V	
电池输入电压	V _{bat}	-0.3	--	+7	V	
DV _{DD} to DGND		-0.3	--	+7	V	
DV _{DD} to AV _{DD}		-0.3		+0.3	V	
IAP, IAN, IBP, IBN, UP, UN		-6		+6	V	
数字IO输出高电平	VOH		--	DV _{DD} +0.3	V	
数字IO输出低电平	VOL	-0.3	--		V	
数字IO输入高电平	VIH		0.7VCC			CMOS
数字IO输入低电平	VIL		0.3VCC			CMOS
数字IO输入高电平	VIH		0.4VCC			TTL
数字IO输入低电平	VIL		0.2VCC			TTL
数字IO的I _{source}	I _{source}	5		10	mA	6mA类型
数字IO的I _{sink}	I _{sink}	7		15	mA	6mA类型
数字IO的I _{source}	I _{source}	3		5	mA	3mA类型
数字IO的I _{sink}	I _{sink}	5		10	mA	3mA类型
模拟输入电压相对于AGND	V _{INA}	-0.3	--	AV _{DD} +0.3	V	
工作温度范围	T _A	-40	--	85	°C	
存储温度范围	T _{stg}	-65	--	150	°C	
最大结温	T _J		125		°C	
无铅焊接温度	T _{SDR}	--	260		°C	

2.2 可靠性参数

测量项目	符号	条件	值	单位
静电放电	HBM	符合JEDEC EIA/JESD22-A114标准	4000	V
	MM	符合JEDEC EIA/JESD22-A115标准	200	V
	CDM	符合JEDEC EIA/JESD22-C101F: 2013标准	500	V
闩锁试验	Latchup	符合JESD78F标准	200	mA
湿度敏感性	MSL	符合IPC/JEDEC J-STD-020D.1标准	3级	/

3 系统控制

3.1 电源管理方案

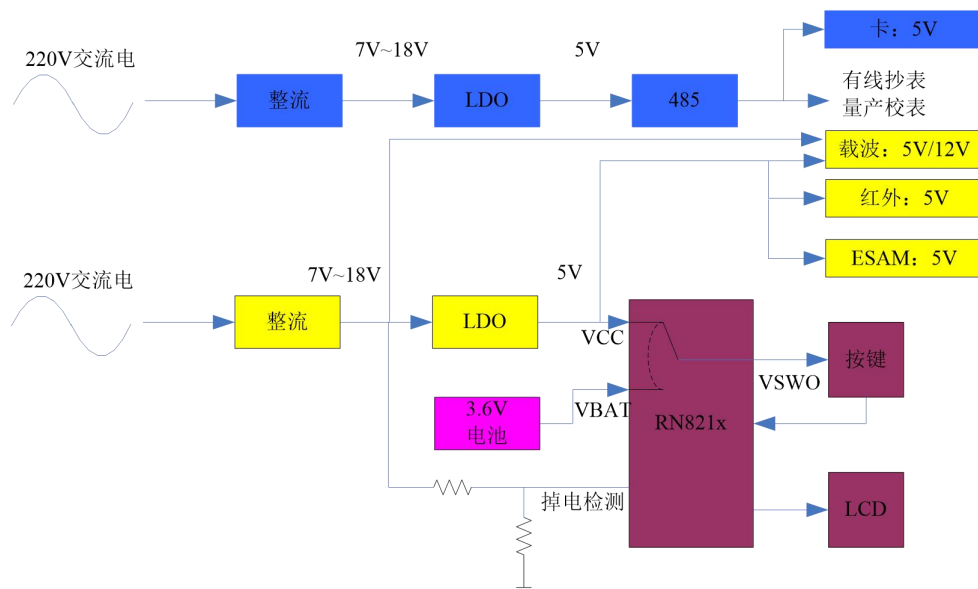


图 3.1 单相表电源管理方案：内部电池切换

3.2 时钟源

● 2 个外部时钟源

LOSC: 32.768KHz 低频晶振

- 1) 用于 RTC 时钟和低频工作下 CPU 时钟;
- 2) 支持 32.768KHz 时钟外灌, 需配置外灌时钟使能, 外灌时不增加额外功耗;
- 3) 32.768KHz 晶振不需要外接电容和电阻, 建议外接负载电容典型值为 12.5pF 的晶振。

HOSC: 外部高频晶振

- 1) 支持 7.3728Mhz、14.7456Mhz、29.4912Mhz, 可用于 CPU 主系统时钟;
- 2) 支持外部高频时钟外灌, 无需配置, 直接从 HOSCI 引脚灌入, 不额外增加功耗;
- 3) 需外接负载电容和电阻, 推荐负载电容 15pF, 电阻 10M, 建议选择负载电容 15pF 的晶振, 7.3728Mhz 晶振 ESR 小于 80Ω, 14.7456Mhz、29.4912Mhz 晶振 ESR 小于 40Ω。

● 3 个内部时钟源

RCH: 内部高频 RC 时钟 (典型值为 29.5MHz), CPU 复位后, 芯片默认运行时钟为 RCH 的 16 分频, 即复位默认主频为 1.8M; 支持 1/2/4/8/16 分频任意切换, 全温度范围内变化 $< \pm 1\%$ 。

RCL: 内部低频 RC 时钟, 用于 WDT 时钟, 也可用于电池供电下的 CPU 时钟和 LCD 时钟。

PLL: 内部 PLL 高频时钟, 由 32.768KHz 倍频到 7.3728MHz、14.7456MHz、29.4912MHz。

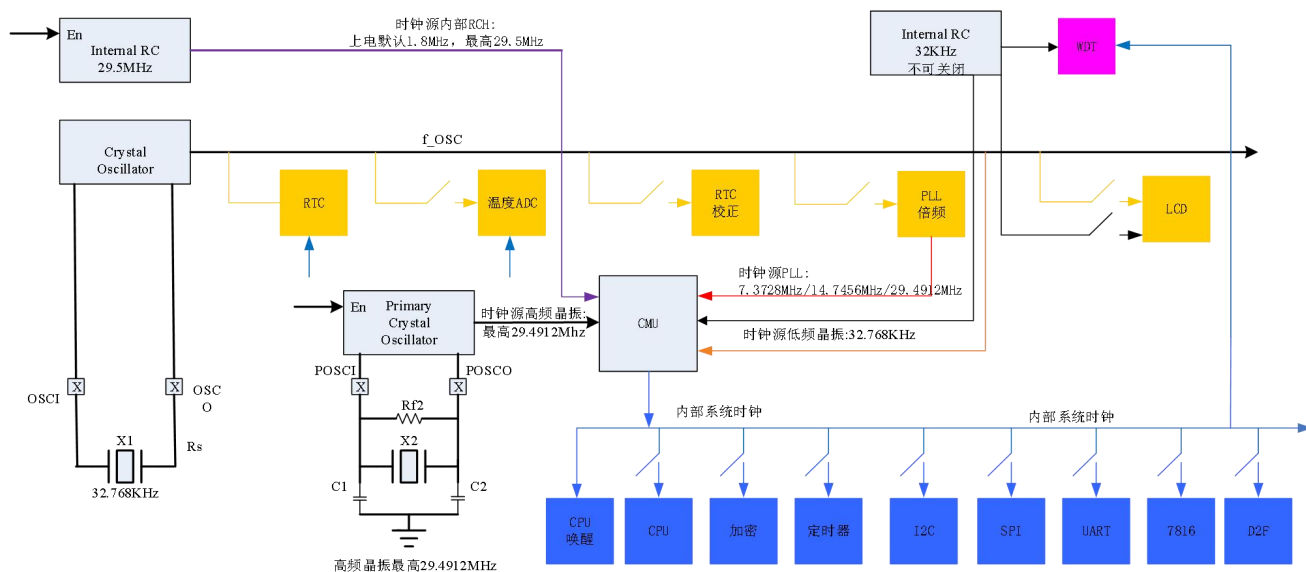
以上 5 个时钟源均可用于 CPU 主系统时钟。

其中运行模式时系统时钟源可以选择为 PLL 也可以选择为高频晶振。

CPU 通过指令从低频时钟切换到高频时钟。系统主时钟可在 RC、PLL (或 HOSC)、LOSC 时钟之间切换。为保证时钟精度, 正常运行模式下应选择 PLL 或外部高频时钟作为系统主时钟。

用户进行时钟切换, 必须调用锐能微库函数。用户不应在应用程序中对 OSC_CTL1(0x0)、SYS_MODE(0x4)

寄存器进行写操作。如果对 OSC_CTL2(0x10)进行写操作，应保证只改变需要操作的 bit 位，不要改变其他寄存器位的值。



3.3 时钟切换

包含有如下切换：

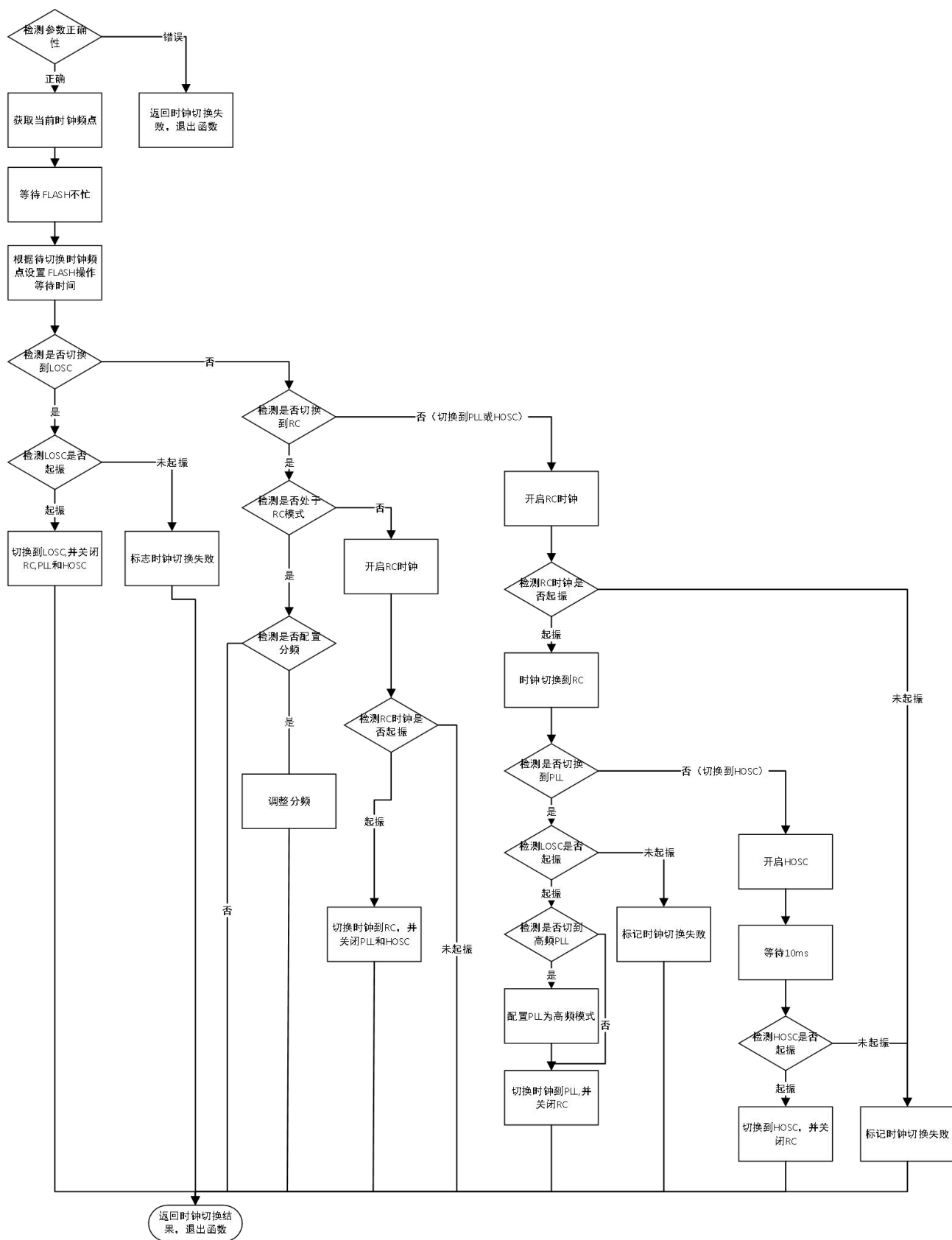
1. 复位后默认为 RCH；
2. PLL/HOSC 与 RCH 切换，由 CPU 指令完成；
3. PLL/HOSC 与 LOSC/RCL 切换，由 CPU 指令完成；
4. LOSC/RCL 与 RCH 切换，由 CPU 指令完成。

时钟切换请调用锐能微提供的库函数完成。

如果选择外部高频晶振 HOSC 作为系统主时钟，在调用库函数前需要配置 OSC_CTL2 寄存器。

如果选择 PLL 作为系统主时钟，并且为第一次上电，需要等待 32KHz 晶振起振后（起振时间约为 0.5s）再调用库函数进行时钟切换。

库函数时钟切换流程图



3.4 Soc 的低功耗模式

M0 的低功耗模式有 Sleep 和 DeepSleep。这两条指令的区别是：

1. CPU 运行 Sleep 指令后，只会关闭 CPU 内核时钟，外设时钟需要寄存器来关闭（见 MODE0 和 MODE1）；
2. CPU 运行 DeepSleep 指令后，不仅会关闭 CPU 内核时钟，也会自动关闭大部分外设时钟（见 MODE0 和 MODE1）；

建议软件不区分 Sleep 和 DeepSleep，直接使用 Sleep，其他外设时钟均采用程序关闭。

除了 M0 本身的低功耗模式，SOC 提供了灵活的机制实现用户不同功耗模式的需求：

1. CPU 可通过调用库函数方式在高频时钟模式 HCM、低频时钟模式 LCM 和系统默认模式 RCH 之间任意切换；
2. CPU、外设的时钟均可关断；
3. 在最低功耗模式（CPU 休眠、SRAM 及数字外设不掉电、RTC 运行）功耗约为 8uA 左右；

用户可根据 SOC 提供的如上机制灵活的实现自己需要的低功耗模式。主要模块上电后的默认状态：

主要模块	默认工作状态
1.5V 电压阈	
M0 内核	开启，可关时钟，永不掉电
中断系统	开启，可关时钟，永不掉电
SRAM	开启，可关时钟，永不掉电
ROM	开启，可关时钟，永不掉电
FLASH	开启，CPU 休眠后可自动关电
RTC	开启，万年历不可关断，没有复位
EMM（正常计量）	关闭，可关时钟，永不掉电
其他外设	关闭，可关时钟，永不掉电
5V 电压阈	
计量 ADC	关闭，可关电源
计量 ref	关闭，可关电源
3V LDO	关闭，可关电源
1.5V LDO	开启，不可关断
RC	开启，可关电源
比较器 CMP2	开启，可关电源
比较器 CMP1	关闭，可关电源
LCD	关闭，可关电源
温度 ADC	定时开启
LVD	关闭，可关电源
比较器	关闭，可关电源
电源复位系统	一直开启
PLL	关闭，可关电源
HOSC	关闭，可关电源
LOSC	一直开启

默认关闭的模块在三种时钟下都可以选择开启或者关闭。

3.5 复位

3.5.1 外部 PIN 复位

外部管脚 RSTN 内置约 50K 欧姆上拉电阻，输入电平为 CMOS 电平。内部滤波时间为 1ms，外部输入低电平超过 1ms 会发生复位。

3.5.2 上下电复位

内置两个上电复位（POR）电路和两个掉电复位（BOR）电路，分别对 VSWO、LDO15（1.5V LDO）进行监测。

该电路始终处于工作状态，不可被关闭，保证系统在超过阈值（约 2.6V）时可以正常工作；在低于阈值（2.2V/1.8V 档可配）时，器件处于复位状态。不必再使用外部复位电路。

本产品还内置了一个可编程电压监视器 LVD，可以监视 VCC 并与设定的阈值做比较，当 VCC 低于或者高于阈值时产生中断。

3.5.3 软件复位

Cortex M0 内置 SCB_AIRCR 寄存器，只需置位该寄存器的 SYSRESETREQ 位即可把引起整个芯片系统的复位，复位效果等同外部 PIN 复位。详情详见 M0 说明文档。

3.5.4 看门狗复位

如果不能在规定时间内喂狗，或者使用非法指令喂狗，芯片内置的硬件看门狗会复位整个芯片，复位效果等同外部 PIN 复位。

3.5.5 CM0 复位

V2 版新增 CM0 复位，支持 MCU 单独复位，在线升级后不需要复位整个芯片，可保证在线升级不影响计量。

3.6 掉电处理

SOC 芯片提供了如下手段做掉电检测，应用软件在检测到掉电后应及时处理完必要的工作，关闭外设，让 cpu 进入低频运行模式或者休眠模式。

1. 模拟外设模块的 CMP1/CMP2

CMP1/CMP2 是低功耗比较器，对输入到该 IO 口的电压作出检测，可检测给芯片供电的 LDO 前端的供电电压，同时也是上电判断的唯一依据；此模块功耗低于 1uA，在实际应用中，可一直开启，客户软件可使用 CMP1/CMP2 中断或者标志位查询的方法来做上下电的判断。

2. 模拟外设模块的 LVD

LVD 模块既可以检测外部引脚 LVDIN 的输入电压（需要外部引脚，功耗比 CMP1/CMP2 大），也可以对芯片的电源电压检测（不需要外部引脚输入，在芯片内部完成对 VCC 电源的检测，阈值多档可设），客户软件可使用中断或者标志位查询的方法来做上下电的判断。

客户可根据实际硬件电路设计，选择 CMP1/CMP2 或 LVD 完成掉电检测功能，优化系统电源管理。

3.7 寄存器描述

系统控制模块的基址：

模块名	物理地址	映射地址
SYSC	0x40034000	0x40034000
寄存器名	地址偏移量	描述

OSC_CTRL1	0x0	系统 OSC 控制寄存器 1
SYS_MODE	0x4	系统模式切换寄存器
SYS_PD	0x8	系统掉电控制寄存器
ADC_CTRL	0xC	ADC 控制寄存器
OSC_CTRL2	0x10	系统 OSC 控制寄存器 2
SYS_RST	0x14	系统复位寄存器
MAP_CTRL	0x18	地址映射控制寄存器
MOD0_EN	0x1C	模块使能 0 寄存器
MOD1_EN	0x20	模块使能 1 寄存器
INTC_EN	0x24	INTC 使能寄存器
KBI_EN	0x28	KBI 使能寄存器
CHIP_ID	0x2C	芯片版本号
SYS_PS	0x30	系统控制寄存器密码保护位
IRFR_CTRL	0x34	RCH 模式下红外时钟分频系数
SYS_CFG (新增)	0x38	系统配置寄存器
TRIM_CFG1	0x78	时钟校正配置寄存器
TRIM_START	0x7C	时钟校正启动寄存器
DMA_PRI1 (新增)	0x80	DMA 优先级配置寄存器 1
FAB_UID0 (新增)	0xF0	芯片唯一码寄存器 0
FAB_UID1 (新增)	0xF4	芯片唯一码寄存器 1
DMA_PRI2 (新增)	0xFC	DMA 优先级配置寄存器 2
ADCIN_CTRL (新增)	0x114	ADC 外灌控制寄存器
SYSCP_CON (新增)	0x118	系统斩波配置寄存器

3.7.1 系统 OSC 控制寄存器 1 OSC_CTRL1(0x0)

位	名称	描述	读/写	复位值
31:17	---	只读，不可写。	R	0
16:11	CLOCK_FLAG	系统时钟开启标志位：如果时钟是开启了，则此位为 1： { RCM,HOSC,RCL,RCH,PLL, LO SC}	R	001101
10:8	SYSCLK_STAT	系统主时钟频率指示： 000：当前系统主时钟为 7.3728MHz； 001：当前系统主时钟为 3.6864MHz； 010：当前系统主时钟为 1.8432MHz； 011：当前系统主时钟为 32.768KHz； 100：当前系统主时钟为 14.7456Mhz； 101：当前系统主时钟为 29.4912Mhz； 其他：无意义	R	010
7	PLL_LOCK	PLL 锁定状态 0：未锁定 1：锁定	R	0
6	PLL_HOSC_ON	系统运行在外部高频或内部 PLL 时钟时，该位为 1； 系统运行在其他时钟时，该位为 0。	R	0
5	IRCH_ON	系统运行在内部高频时钟时，该位为 1；	R	1

		系统运行在其他时钟时，该位为 0。		
4	LOSC_ON	系统运行在外部低频时钟时，该位为 1； 系统运行在其他时钟时，该位为 0。	R	0
3:2	PLL_HOSC_DIV	系统主时钟分频选择：（只对高频时钟模式有效） 00：以 PLL、HOSC 作为 CPU 主时钟； 01：以 PLL、HOSC 的二分频作为 CPU 主时钟； 10：以 PLL、HOSC 的四分频作为 CPU 主时钟； 11：以 HOSC（时钟频率选择为 14MHz 和 29MHz 时） 的八分频作为 CPU 主时钟； 备注：只能在 RC 或者 LC 模式才能更改。 备注：上述寄存器只决定分频系数，具体的系统主频 需要根据分频系数和当前时钟源选择来确定。	R/W	01
1	IRCH_PD	内部 RC 使能位： 0：打开； 1：关闭。	R/W	0
0	PLL_PD	PLL 模块使能位 0：打开 1：关闭	RW	1

用户进行时钟切换，建议调用锐能微库函数。不建议用户在应用程序中对 OSC_CTL1(0x0)寄存器进行写操作。

3.7.2 系统模式设置寄存器 SYS_MODE(0x4)

位	名称	描述	读/写	复位值
31:6	---	只读，不可写	R	0
5	FLASH_BUSY	Flash busy 状态，不能进入模式切换： 0：idle 1：busy	R	0
4	---	保留位	R	0
3:0	MODE	写入 D，设置进入高频模式 HCM，bit2 读为 1； 写入 E，设置进入 RC 模式 RCM，bit1 读为 1； 写入 F，设置进入 32.768KHz 模式 LCM，bit0 读为 1。 即该寄存器读出值为：{0,HCM,RCM,LCM}	R/W	2

注：当前模式状态的指示应该读取 LOSC_ON、IRCH_ON、PLL_HOSC_ON（OSC_CTL 寄存器 bit4~6）这三个状态。而不是读取该寄存器，该寄存器只代表模式切换命令写入，不代表已经切换到预期模式。

用户进行时钟切换，建议调用锐能微库函数。不建议用户在应用程序中对 SYS_MODE(0x4)寄存器进行写操作。

3.7.3 系统掉电控制寄存器 SYS_PD(0x8)

位	名称	描述	读/写	复位值
31:18	---	只读，不可写	R	0
17	ldo3_pd	LDO3 电源开关 =0：掉电 =1：上电 LDO3 受三路 ADC 开关控制，只要有一路 ADC 开启，	R/W	0

		LDO3 就开启。		
16	vsel_bor5	BOR5 掉电阈值选择信号 =0: vil=2.2v, 默认配置 =1: vil=1.8v 芯片上电时复位阈值一直为 vih=2.5V	R/W	0
[15:12]	Reserved	保留	R	0
11	hysen_cmp2	cmplp2 的内部迟滞比较器迟滞开关 0x1: 打开迟滞 0x0: 关闭迟滞	R/W	0
10	hysen_cmp1	cmplp1 的内部迟滞比较器迟滞开关 0x1: 打开迟滞 0x0: 关闭迟滞	R/W	0
9	PWD_CMP2R	CMP2 内部 600K 电阻采样开关 0: CMP2 内部电阻采样开启。外围电路需要注意内部 600K 对地电阻的影响, 比较器 Vil 典型值为 1.28V, 迟滞为 0.22V, 此时不要将 bit11 配置为 1; 1: CMP2 内部电阻采样关闭。比较器阈值典型值为 0.9V, 默认无迟滞; 可将 bit11 置 1 使之有 0.14V 的迟滞, 此时 Vil 典型值为 0.84V; 注 1: 当 CMP2_PD=1 时, 内部采样电阻关闭, 当 CMP2_PD=0 时, 内部采样电阻由 PWD_CMP2Rt 进行配置; 默认配置下内部 600K 电阻开启。 注 2: 此内部 600K 电阻对相应 AIN 模拟测试通道同样有效; 如做 AIN 使用, 需考虑该内阻影响或确保关闭该内阻。	R/W	0
8	PWD_CMP1R	CMP1 内部 600K 电阻采样开关 0: CMP1 内部电阻采样开启, 外围电路需要注意内部 600K 对地电阻的影响, 比较器 Vil 典型值为 1.28V, 迟滞为 0.22V, 此时不要将 bit10 配置为 1; 1: CMP1 内部电阻采样关闭, 比较器阈值典型值为 0.9V, 默认无迟滞; 可将 bit10 置 1 使之有 0.14V 的迟滞, 此时 Vil 典型值为 0.84V; 注 1: 当 CMP2_PD=1 时, 内部采样电阻关闭, 当 CMP2_PD=0 时, 内部采样电阻由 PWD_CMP2Rt 进行配置; 默认配置下内部 600K 电阻关闭。 注 2: 此内部 600K 电阻对相应 AIN 模拟测试通道同样有效; 如做 AIN 使用, 需考虑该内阻影响或确保关闭该内阻。	R/W	0
7	BGRLP_PD	BGRLP 电源开关 0: 上电 1: 掉电 备注: 当 OSC_CTL2.PM_SEL=1 时, 选择 BGRLP 作为计量和测	R/W	0

		温基准, IA 通道 ADC、IB 通道 ADC、U 通道 ADC、温度测量启动、BGR_LP_PD 寄存器开启满足其中一个条件 BGR_LP 就开启。		
6	BGR_PD	BGR 电源开关 0: 上电 1: 掉电 备注: 当 OSC_CTL2.PM_SEL=0 时, 选择 BGR_LP 作为计量和测温基准, I1 通道 ADC、I2 通道 ADC、U1 通道 ADC、温度测量启动、BGR_PD 寄存器开启满足其中一个条件 BGR 就开启。	R/W	1
5	CMP2_PD	比较器 2 电源开关 0: 上电 1: 掉电	R/W	0
4	CMP1_PD	比较器 1 电源开关 0: 上电 1: 掉电	R/W	1
3	LVD_PD	LVD 电源开关 0: 上电 1: 掉电	R/W	1
2	ADCU_PD	U 通道 ADC 电源开关 0: 上电 1: 掉电	R/W	1
1	ADCI2_PD	I2 通道 ADC 电源开关 0: 上电 1: 掉电	R/W	1
0	ADCI1_PD	I1 通道 ADC 电源开关 0: 上电 1: 掉电	R/W	1

3.7.4 ADC 控制寄存器 ADC_CTRL(0xC)

比特位	名称	描述	读/写标志	复位值
31:12	---	只读, 不可写	R	0
11:9	保留	保留位, 可写为 1, 无实际意义。	R/W	0
8:6	ADCU_PGA	U 通道 ADC 增益配置 =x00 1 倍 =x01 2 倍 =x10 4 倍 =x11 4 倍	R/W	0
5:3	ADCI2_PGA	I2 通道 ADC 增益配置 =x00 1 倍 =x01 2 倍 =x10 4 倍 =x11 4 倍	R/W	0
2:0	ADCI1_PGA	I1 通道 ADC 增益配置 =x00 1 倍 =x01 2 倍 =x10 8 倍 =x11 16 倍	R/W	0

3.7.5 系统 OSC 控制寄存器 2 OSC_CTRL2(0x10)

比特位	名称	描述	读/写标志	复位值
31:23	---	只读，不可写	R	0
22	PM_SEL	计量功耗模式选择： 0：正常功耗模式。 EMU 模式时，计量时钟跟随系统时钟，不支持 32K 主频； NVM 模式时，NVM 时钟跟随系统时钟，32K 主频时 NVM 时钟使用 RCH 时钟，向下兼容； 基准电压来自 BGR。 1：低功耗模式。 EMU 模式不支持配置为低功耗模式； NVM 模式时，NVM 时钟固定为 RCM； 基准电压来自 BGR LP。	R/W	0
21	---	保留	R	0
20	RCM_PD	内部 RCM 使能位： 0：关闭； 1：开启。	R/W	0
18:16	RCH_FREQ	000：RCM 模式下 RCH 的频率为 1.8MHz； 001：RCM 模式下 RCH 的频率为 3.6MHz； 010：RCM 模式下 RCH 的频率为 7.3MHz； 011：RCM 模式下 RCH 的频率为 14.7Mhz； 100：RCM 模式下 RCH 的频率为 29.5Mhz； 其他：保留 备注：客户应调用库函数选择芯片运行频率，不要在应用程序中改变此位的值。	R/W	0
15	RCL_LOSC_FLT_SEL	滤波时钟源选择 0：滤波时钟选择 LOSC； 1：滤波时钟选择 RCL	R/W	0
14	RCL_LOSC_RTC_SEL	RTC 模块时钟源选择 0：RTC 模块时钟选择 LOSC； 1：RTC 模块时钟选择 RCL；	R/W	0
13	RCL_LCD	0：LCD 选择 LOSC 外部低频晶振作为时钟源； 1：LCD 选择 RCL 内部低频晶振作为时钟源；	R/W	0
12	RCL_LOSC_SYS_SEL	CPU 系统低频时钟源选择 0：CPU 系统低频时钟选择 LOSC； 1：CPU 系统低频时钟选择 RCL；	R/W	0
11:10	保留	可写，内部测试寄存器，用户不要改变此寄存器默认值。	R/W	00
9	LOSC_WEN	=0：LOSC_PD 位不可写 1； =1：LOSC_PD 位可写 1 必须先将 LOSC_WEN 位写 1 然后再将 PD 位写 1。	R/W	0

8	LOSC_PD	LOSC 使能位： 0：打开； 1：关闭。 该位同时也是外灌时钟使能信号，=1 使能外灌时钟； 只有系统运行在 RC 模式时才写可。	R/W	0
7:5	PLL_FREQ	PLL 频点固定为 14.7456MHz，通过数字分频实现频率选择： 000：运行频率选择为 7.3728MHz； 001：运行频率选择为 14.7456MHz； 010：运行频率选择为 29.4912Mhz； 011：PLL 输出频点选择为 29.4912Mhz；（为 58.9824Mhz 二分频） 其他：保留 客户应调用库函数选择芯片运行频率，不要在应用程序中改变此位的值。	R/W	000
4	PLL_HOSC_SEL	全速运行时系统主时钟选择： 0：选择 PLL 输出作为系统主时钟； 1：选择备用高频晶体作为系统主时钟。 该配置项只能在 RC 模式和低频模式下配置。	R/W	0
3	HOSC_PD	外部高频振荡器使能位： 0：打开 1：关闭	RW	1
2:0	HOSC_FREQ	000：外接高频晶振为 7.3728MHz 001：外接高频晶振为 14.7456MHz 010：保留，用户不要使用该选项 011：外接高频晶振为 29.4912MHz（驱动增强）	RW	000

如果系统时钟选择为外部高频晶振，在调用锐能微库函数进行时钟切换前，用户程序需要对 OSC_CTL2 寄存器进行配置。

系统时钟配置真值表（PLL 和 HOSC 模式下）：

PLL_HOSC_SEL	PLL_FREQ	时 钟 源	PLL_HOSC_DIV =000	PLL_HOSC_DIV =001	PLL_HOSC_DIV =010	PLL_HOSC_DIV =011
			不分频	二分频	四分频	八分频
0	000	PLL	7.3728Mhz	3.6864MHz	1.8432MHz	不支持八分频， 如 配 置 则 为 1.8432MHz
1	000	HOSC	7.3728Mhz	3.6864MHz	1.8432MHz	不支持八分频， 如 配 置 则 为 1.8432MHz
0	001	PLL	14.7456Mhz	7.3728Mhz	3.6864Mhz	1.8432Mhz
1	001	HOSC	14.7456Mhz	7.3728Mhz	3.6864Mhz	1.8432Mhz
0	010	PLL	29.4912Mhz	14.7456Mhz	7.3728Mhz	3.6864Mhz
1	011	HOSC	29.4912Mhz	14.7456Mhz	7.3728Mhz	3.6864Mhz

3.7.6 系统复位寄存器 SYS_RST(0x14)

此寄存器的 8 到 5 位只能被上电/掉电复位，上电复位后，会误触发其他复位，会导致复位标志置起，应用时，在上电后，需要先将标志清 0，再使用这些标志，保证这些复位标志的可靠性。

比特位	名称	描述	读/写	复位值
31:21	---	预留	R	0
20	I2C_RST_REQ	软件复位 I2C 模块： 写入 1 复位 I2C 计算模块，寄存器和数字逻辑都复位； 写入 0 取消 I2C 模块复位。	R/W	0
19:14	UART_CLKG	UART5~UART0 CM0 复位的锁定使能 =0：不使能 =1：使能锁定功能。当使能 CM0 复位时，UART 可以正常工作，但锁定 CPU 访问 UART 功能，CM0 复位完成后，自动释放。	R/W	0
13	CM0_RST	CM0 复位标志 =0 表示没有发生 =1 表示发生过该复位 写 1 清零，除了 CM0 复位不能清除该标志位，其他复位源都可以复位该位。	R/W	0
12	CM0_ENRST	CM0 复位使能 写入 1 复位 CM0，但不会复位计量以及与计量相关的模块，如包含脉冲转发功能的 UART 等 写入 0 取消复位 CM0 注意：CM0 复位为上升沿触发，建议每次使能 CM0 复位前现将该位清 0 或，每次 CM0 复位后将该位清 0。	R/W	0
11	BOR_V2P8_IF	BOR28 中断标志，写 1 清 0 =0：电源电压正常 =1：电源电压低于 BOR 28 阈值 当 BOR_V2P8_IE=1 时，产生 CPU 中断。	R/W	0
10	BOR_V2P8_IE	BOR28 中断使能 =0：不使能 =1：使能	R/W	0
9	BOR_V2P8_ENRST	BOR28 复位 CPU 使能 =0：当电源电压低于 2.8V 时不复位 CPU =1：当电源电压低于 2.8V 时复位 CPU	R/W	0
8	MCU_RST	CPU 复位标志（发生过软件复位或者 LOCK UP 复位）： =1 表示发生过该复位，=0 表示没有发生。写 1 清零	R	0
7	WDT_RST	WDT 复位标志： =1 表示发生过该复位，=0 表示没有发生。写 1 清零	R	0
6	PIN_RST	外部管脚复位标志： =1 表示发生过该复位，=0 表示没有发生。写 1 清零	R	0
5	POWEROK_RST	电源上下电复位标志 =1 表示发生过该复位，=0 表示没有发生。写 1 清零	R	1
4	---	只读，不可写	R	0
3	EMUREG_RST_REQ	软件复位 EMU/NVM/FLK 模块寄存器：	R/W	0

		写入 1 复位 EMU/NVM/FLK 配置寄存器； 写入 0 取消 EMU/NVM/FLK 配置寄存器； 不复位 EMU 模块计算模块和结果寄存器； 注意：EMUREG_RST_REQ 写 1 会一直复位目标，直到写 0 才结束复位		
2	LOCKUP_ENRST	LOCKUP 使能复位（CPU 发生了两次 Hard Fault 会引起 LOCKUP，如果使能该位，可引起系统复位）： 0: LOCKUP 不引起系统复位 1: LOCKUP 引起系统复位	R/W	0
1	NVM_RST_REQ	软件复位全失压计算模块： 写入 1 复位全失压计算模块； 写入 0 取消全失压计算模块复位。 不复位全失压模块配置寄存器；	R/W	0
0	EMU_RST_REQ	软件复位 EMU 计算模块： 写入 1 复位 EMU 计算模块； 写入 0 取消 EMU 计算模块复位。 不复位 EMU 模块寄存器；	R/W	0

3.7.7 地址映射控制寄存器 MAP_CTRL(0x18)

比特位	名称	描述	读/写	复位值
31:5	---	只读，不可写	R	0
4	保留	可写，有密码保护	R/W	0
3	---	只读，不可写	R	0
2:0	REMAP	地址映射： 000: FLASH 映射在 0 地址（正常模式） 001: 保留 010: FLASH 与 SRAM 映射地址互换 011: 保留，用户不要使用该选项 100: FLASH 映射在 1/2 容量地址 其他: 保留，用户不要使用该选项	R/W	00

3.7.8 模块使能 0 寄存器 MOD0_EN(0x1C)

比特位	名称	描述	读/写	复位值
31:28	---	只读，不可写	R	0
27	---	预留	R/W	0
26	CRC_EN	CRC 模块使能清零，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
25	LPUART_EN	LPUART 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
24	---	预留	R/W	0
23	---	预留	R/W	0

22	---	预留	R/W	0
21	---	预留	R/W	0
20	SIMP_TC_EN	SIMP_TC 模块使能清零，时钟门控： 0：时钟停止，模块清， 1：时钟启动，模块使能	R/W	0
19	---	预留	R	0
18	SPI3_EN	SPI3 模块使能清零，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0：时钟停止，模块清零 1：时钟启动，模块使能	R/W	0
17	SPI2_EN	SPI2 模块使能清零，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0：时钟停止，模块清零 1：时钟启动，模块使能	R/W	0
16	--	预留	R	0
15	SPIO_EN	SPIO 模块使能，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0：时钟停止，模块关闭 1：时钟启动，模块使能	R/W	0
14	I2C_EN	I2C 模块使能，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0：时钟停止，模块关闭 1：时钟启动，模块使能	R/W	0
13	ISO7816_EN	ISO7816 模块使能，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0：时钟停止，模块关闭 1：时钟启动，模块使能	R/W	0
12	UART38K_EN	UART38K 红外调制时钟开启使能，cm0 进入 deepsleep 同步关闭此时钟： 0：时钟停止 1：时钟启动	R/W	0
11	UART3_EN	UART3 模块使能，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0：时钟停止，模块关闭 1：时钟启动，模块使能	R/W	0
10	UART2_EN	UART2 模块使能，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0：时钟停止，模块关闭 1：时钟启动，模块使能	R/W	0
9	UART1_EN	UART1 模块使能，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0：时钟停止，模块关闭 1：时钟启动，模块使能	R/W	0
8	UART0_EN	UART0 模块使能，时钟门控，cm0 进入 deepsleep 同	R/W	0

		步关闭此时钟： 0: 时钟停止，模块关闭 1: 时钟启动，模块使能		
7	UART5_EN	UART5 模块使能，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0: 时钟停止，模块关闭 1: 时钟启动，模块使能	R/W	0
6	UART4_EN	UART4 模块使能清零，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
5	TC1_EN	TC1 模块使能，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0: 时钟停止，模块关闭 1: 时钟启动，模块使能	R/W	0
4	TC0_EN	TC0 模块使能，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0: 时钟停止，模块关闭 1: 时钟启动，模块使能	R/W	0
3	---	预留	R/W	0
2	---	保留位，可读可写，默认值为 1；无实际意义。	R/W	1
1	---	只读，不可写	R	0
0	---	预留	R/W	0

3.7.9 模块使能 1 寄存器 MOD1_EN(0x20)

比特位	名称	描述	读/写	复位值
31:19	---	只读，不可写	R	0
18	ECT_EN	ECT 模块时钟使能： 0: 时钟停止 1: 时钟启动	R/W	0
17	IOCNT_EN	IOCNT 模块时钟使能： 0: 时钟停止 1: 时钟启动	R/W	0
16	FLK_EN	FLK 模块运算时钟使能： 0: 时钟停止； 1: 时钟启动； FLK 模块寄存器时钟同 EMU	R/W	0
15	M2M_EN	M2M 模块时钟使能，cm0 进入 deepsleep 同步关闭此时钟： 0: 时钟停止； 1: 时钟启动；	R/W	0
14	DSP_EN	硬件 DSP 核模块(FFT 等)时钟使能，cm0 进入 deepsleep 同步关闭此时钟： 0: 时钟停止；	R/W	0

		1: 时钟启动;		
13	D2F_EN	硬件 D2F 模块（电能积分器）时钟使能: 0: 时钟停止; 1: 时钟启动;	R/W	0
12	NVM_REG_EN	NVM 模块寄存器读写时钟门控: 0: 时钟启动 1: 时钟停止 默认启动	R/W	0
11	SAR_EN	SAR 模块使能, apb 总线时钟门控: 0: 时钟停止 1: 时钟启动	R/W	0
10	RTC_EN	RTC apb 总线时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止 1: 时钟启动 建议客户不要关闭该时钟。	R/W	1
9	WDT_EN	WDT apb 总线时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止 1: 时钟启动 建议客户不要关闭该时钟。	R/W	1
8	NVM_EN	全失压计算模块使能, 时钟门控: 0: 时钟停止 1: 时钟启动	R/W	0
7	EMU_EN	EMU 模块时钟门控和 FLK 模块寄存器时钟门控: 0: 时钟停止 1: 时钟启动	R/W	0
6	LCD_EN	LCD 模块使能, 时钟门控: 0: 时钟停止, 模块关闭 1: 时钟启动, 模块使能	R/W	0
5	GPIO_EN	GPIO 模块使能清零, 时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止, 模块关闭 1: 时钟启动, 模块使能	R/W	0
4	---	预留	R/W	0
3	SPI1_EN	SPI1 模块使能清零, 时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
2	---	只读, 不可写	R	0
1:0	--	预留	R/W	0

3.7.10 INTIC 使能寄存器 INTIC_EN(0x24)

比特位	名称	描述	读 / 写 标	复 位
-----	----	----	---------	-----

			志	值
31:9	---	只读，不可写	R	0
8	INTC_EN	INTC apb 模块时钟门控： 0: 时钟停止 1: 时钟启动	R/W	0
7	INTC7_EN	INTC7 模块使能，时钟门控： 0: 时钟停止 1: 时钟启动	R/W	0
6	INTC6_EN	INTC6 模块使能，时钟门控： 0: 时钟停止 1: 时钟启动	R/W	0
5	INTC5_EN	INTC 5 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
4	INTC4_EN	INTC 4 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
3	INTC3_EN	INTC3 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
2	INTC2_EN	INTC 2 模块使能，时钟门控： 0: 时钟停止，模块关闭 1: 时钟启动，模块使能	R/W	0
1	INTC1_EN	INTC 1 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
0	INTC0_EN	INTC0 模块使能，时钟门控： 0: 时钟停止，模块关闭 1: 时钟启动，模块使能	R/W	0

3.7.11 KBI 使能寄存器 KBI_EN(0x28)

比特位	名称	描述	读/写	复位值
31:9	---	只读，不可写	R	0
8	KBI_EN	KBI apb 模块时钟门控： 0: 时钟停止 1: 时钟启动	R/W	0
7	KBI7_EN	KBI 7 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
6	KBI6_EN	KBI 6 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
5	KBI5_EN	KBI 5 模块使能清零，时钟门控： 0: 时钟停止，模块清零	R/W	0

		1: 时钟启动, 模块使能		
4	KBI4_EN	KBI 4 模块使能清零, 时钟门控: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
3	KBI3_EN	KBI 3 模块使能清零, 时钟门控: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
2	KBI2_EN	KBI 2 模块使能清零, 时钟门控: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
1	KBI1_EN	KBI 1 模块使能, 时钟门控: 0: 时钟停止, 模块关闭 1: 时钟启动, 模块使能	R/W	0
0	KBI0_EN	KBI0 模块使能, 时钟门控: 0: 时钟停止, 模块关闭 1: 时钟启动, 模块使能	R/W	0

3.7.12 器件 ID 寄存器 CHIP_ID(0x2C)

比特位	名称	描述	读/写	复位值
31:24	---	只读, 不可写	R	0
15:0	CHIP_ID	芯片版本号: 8217 V2 版所有型号读出均为 8217 D 版所有型号读出均为 RN8217 B 版和 C 版读出为 8215。	R	8217

3.7.13 系统控制密码寄存器 SYS_PS(0x30)

位	名称	描述	读/写	复位值
31:8	---	预留	R	0
7:0	SYS_PSW	当 SYS_PSW=0x82 时, 0x00~0x28、0x38、0x80、0x114 寄存器可写; 当 SYS_PSW=其他值时, 0x00~0x28、0x38、0x80、0xFC、0x114、0x118 寄存器不可写; 该寄存器读出值为写入的值。 建议用户在写操作完成后马上关闭写使能。	R/W	00

3.7.14 红外配置寄存器 IRFR_CTRL(0x34)

位	名称	描述	读/写	复位值
31:6	---	预留	R	0
5:0	IRFR_CYCLE	RCH 模式下, 红外时钟分频系数 ... 0x19:红外输出时钟 36.9K; 0x18:红外输出时钟 38.4K; 0x17:红外输出时钟 40K; ...	R/W	0x18

		计算公式: $RCH29M/(32*INFRARED_CYCLE)$ 不可配置为 0。		
--	--	---	--	--

3.7.15 系统配置寄存器 SYS_CFG(0x38) (新增)

位	名称	描述	读/写	复位值
31:16	---	预留	R	0
15:8	RTC_SW_EN	RTC 时钟切换使能配置 =0x75: 切断 RTC 万年历时钟 =其他: RTC 万年历正常运行 当 RTC 时钟需要进行 LOSC 与 RCL 切换时, 需要配置 RTC_SW_EN 先关闭 RTC 万年历时钟,再配置 OSC_CTL2[14], RCL_LOSC_RTC_SEL 位选择时钟, 然后再配置 RTC_SW_EN, 使能万年历运行。 当 SYS_PS=0x82 时该寄存器可写。 RTC 时钟切换使能配置步骤: SYSCTL->SYS_PS=0x82; SYSCTL->SYS_CFG =0x75<<8; SYSCTL->OSC_CTL2 =(1<<14);	R/W	0
7:0	EXT_REF_EN	VREF 外灌使能配置 =0xE8: 使能 VREF 外灌 =其他: 不使能 VREF 外灌 当 SYS_PS=0x82 时该寄存器可写。 当 EXT_REF_EN 高 4bit[7:4]=0xE 时, 低 4bit[3:0]才可写。 VREF 外灌使能配置步骤: SYSCTL->SYS_PS=0x82; SYSCTL->SYS_CFG=0xE0; SYSCTL->SYS_CFG=0xE8;	R/W	0

3.7.16 时钟校正配置寄存器 TRIM_CFG1(0x78)

位	名称	描述	读/写	复位值
31:30	---	预留	R	0
29	CAL_CLK_SEL1	具体定义参考 bit26 CAL_CLK_SEL 的描述。	R/W	0
28	CAL_OV_IE	被校准时钟计数器溢出标志中断使能: 0: 不使能中断; 1: 使能中断;	R/W	0
27	CAL_DONE_IE	时钟校准完成标志中断使能: 0: 不使能中断; 1: 使能中断;	R/W	0
26	CAL_CLK_SEL	{CAL_CLK_SEL1, CAL_CLK_SEL}共同定义被校正时钟源选择: CAL_CLK_SEL1 为 bit29 00: 被校正时钟源选择 RCH; 01: 被校正时钟源选择 RCL; 10: 被校正时钟源选择 RCM;	R/W	0

		11: 保留		
25: 24	REF_CLK_SEL	参考时钟源选择 00: 参考时钟源选择 LOSC; 01: 参考时钟源选择 HOSC; 10: 参考时钟源选择 RCH; 11: 参考时钟源选择 PLL;	R/W	11
23:20	---	预留	R	0
19:0	REF_CLK_CNT [19:0]	参考时钟计数值	R/W	0x10000

3.7.17 时钟校正启动寄存器 TRIM_START(0x7C)

位	名称	描述	读/写	复位值
31:28	---	预留	R	0
27	STOP	时钟校准终止位: 0: 无操作; 1: 终止时钟校准; Note: 如果终止时钟校准, 需要将该位写 0 后才能重新开始时钟校准。	R/W	0
26	START	时钟校准启动位: 0: 无操作; 1: 启动时钟校准; Note: 时钟校准完成或被终止后, 该位自动清零。	R/W	0
25	CAL_OV	被校正时钟计数器溢出标志: 0: 没有溢出; 1: 溢出; Note: 写 1 清 0。	R/W	0
24	CAL_DONE	时钟校准完成标志: 0: 未完成; 1: 已完成; Note: 写 1 清 0。	R/W	0
23:20	---	预留	R	0
19:0	CAL_CLK_CNT [19:0]	被校准时钟返回的计数值	R	0

举例:

1. 选择参考时钟为 LOSC、选择被校准时钟为 RCH;
2. 选择参考时钟计数器 REF_CLK_CNT 为 0x1000, 计数时间为 0.125S;
3. 启动时钟校正操作, 查询标志位或者等待系统控制中断产生;
4. 假设读取得到的被校正时钟返回的计数值 CAL_CLK_CNT=0x 61A80, 十进制为 400000;
5. 那么测量到的 RCH 频率值为:

$$\begin{aligned}
 & (\text{CAL_CLK_CNT}/\text{REF_CLK_CNT}) * 32768\text{Hz} \\
 & = (400000/4096) * 32768\text{Hz} \\
 & = 3200000\text{Hz}
 \end{aligned}$$

=3.2MHz

3.7.18 DMA 优先级配置寄存器 1DMA_PRI1(0x80) (新增)

位	名称	描述	读/写	复位值
31:28	WKEY	WKEY 是 0~25bit 的写操作密码保护位，密码是 0xE。软件写 0~25bit 时必须保证同时写入的数据高 4 位为 0xE 并且将 SYS_PSW 为 8'h82。	R	0
27:26	DMA_CH13_PRI	通道 13: TC1_PWM 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
25:24	DMA_CH12_PRI	通道 12: 保留	R/W	0x0
23:22	DMA_CH11_PRI	通道 11: UART5 DMA 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
21:20	DMA_CH10_PRI	通道 10: UART4 DMA 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
19:18	DMA_CH9_PRI	通道 9: UART3 DMA 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
17:16	DMA_CH8_PRI	通道 8: UART2 DMA 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
15:14	DMA_CH7_PRI	通道 7: UART1 DMA 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
13:12	DMA_CH6_PRI	通道 6: UART0 DMA 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
11:10	DMA_CH5_PRI	通道 5: SPI3 DMA 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
9:8	DMA_CH4_PRI	通道 4: SPI2 DMA 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
7:6	DMA_CH3_PRI	通道 3: SPI1 DMA 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
5:4	DMA_CH2_PRI	通道 2: SPI0 DMA 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
3:2	DMA_CH1_PRI	通道 1: EMU DMA 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
1:0	DMA_CH0_PRI	通道 0: CACHE 访问 SRAM 优先级配置，优先级固定为 3 0x3: 最高优先级; 0x2: 次高优先级 0x1: 低优先级; 0x0: 最低优先级 优先级相同时，通道号小的优先级高。	R	0x3

3.7.19 DMA 优先级配置寄存器 2DMA_PRI2(0xFC) (新增)

位	名称	描述	读/写	复位值
31:28	WKEY	WKEY 是 0~25bit 的写操作密码保护位，密码是 0xE。软件写 0~25bit 时必须保证同时写入的数据高 4 位为 0xE 并且将 SYS_PSW 为 8'h82。	R	0

27:8	---	保留	R	0x0
7:6	DMA_CH17_PRI	通道 17: CPU 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
5:4	DMA_CH16_PRI	通道 16: M2M 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
3:2	DMA_CH15_PRI	通道 15: CRC 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
1:0	DMA_CH14_PRI	通道 14: DSP DMA 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0

3.7.20 芯片唯一码寄存器 0FAB_UID0(0xF0) (新增)

比特位	名称	描述	读 / 写 标志	复位值
31:0	FAB_UID0	与 FAB_UID1 构成芯片唯一标识码	R	0

3.7.21 芯片唯一码寄存器 1FAB_UID1(0xF4) (新增)

比特位	名称	描述	读 / 写 标志	复位值
31:0	FAB_UID1	与 FAB_UID0 构成芯片唯一标识码	R	0

3.7.22 ADC 外灌控制寄存器 ADCIN_CTRL(0x114) (新增)

位	名称	描述	读/写	复位值
31:10	---	保留	R	0
9	ADC_CLKO_SEL	ADC 1bit 外灌模式, ADC_CLKO 时钟频率选择, =0: 1.8432Mhz =1: 3.6864Mhz	R/W	0
8	ADC_CLK_OEN	ADC 1bit 外灌模式, ADC 时钟输出使能 =0: 不使能 =1: 使能时钟输出 需配置 GPIO 复用为 ADC_CLKO 才有时钟输出	R/W	0
7:4	---	保留	R	0
3	---	保留	R/W	0
2	---	保留	R	0
1	ADCIN_IB	IB 通道外灌 1bit 使能配置 =0: 不使能 =1: 使能 IB 通道外灌 1bit 模式, 从 IB_IN 灌入 1bit	R/W	0
0	ADCIN_IA	IA 通道外灌 1bit 使能配置 =0: 不使能 =1: 使能 IA 通道外灌 1bit 模式, 从 IA_IN 灌入 1bit	R/W	0

3.7.23 系统斩波配置寄存器 SYSCP_CON(0x118) (新增)

针对直流应用配置

位	位名称	功能描述	读/写	复位值
31:9	--	只读，不可写。	R	0
8	adc_syscp_ph	仅在 adc_syscp_mode=1 时有效： =0，相位为低，正向采样； =1，相位为高，反向采样。	R/W	0
7:6	--	只读，不可写。	R	0
5:4	adc_syscp_sel	ADC 输入信号正反向采样切换频率配置寄存器，仅在 adc_syscp_mode=0 时有效： =0，N=2048； =1，N=4096； =2，N=8192； =3，N=16384； 备注：正反向时间相同，即 N/2 个 1bit 正向，N/2 个 1bit 反向。 切换周期=（1.8432e6/N）Hz	R/W	0
3	adc_syscp_mode	syschop 模式选择 0：自动模式，ADC 输入采样信号方向自动切换 1：手动模式，由 adc_syscp_ph 配置决定 ADC 输入采样信号方向正接还是反接。 备注：手动模式时，计量模块根据系统给到的 adc system chop 相位判断是否进行 0/1 取反操作。相位为低，模拟 ADC 为信号正向采样，计量模块不取反；相位为高，模拟 ADC 为信号反向采样，计量模块做 0/1 取反。	R/W	0
2	adc_syscpu_en	U 通道 syschop 使能： 0：不使能 1：使能	R/W	0
1	adc_syscpib_en	IB 通道 syschop 使能： 0：不使能 1：使能	R/W	0
0	adc_syscpia_en	IA 通道 syschop 使能： 0：不使能 1：使能	R/W	0

4 CPU 系统

4.1 概述

有 2 种方式（2 个主设备）可以发起对 SoC 内置设备的访问：

◎ Cortex-M0：

- 指令访问和数据访问；
- 可访问所有的从设备；

◎ 外置的 SWD 控制器（如 JLINK 或类似功能的设备）：

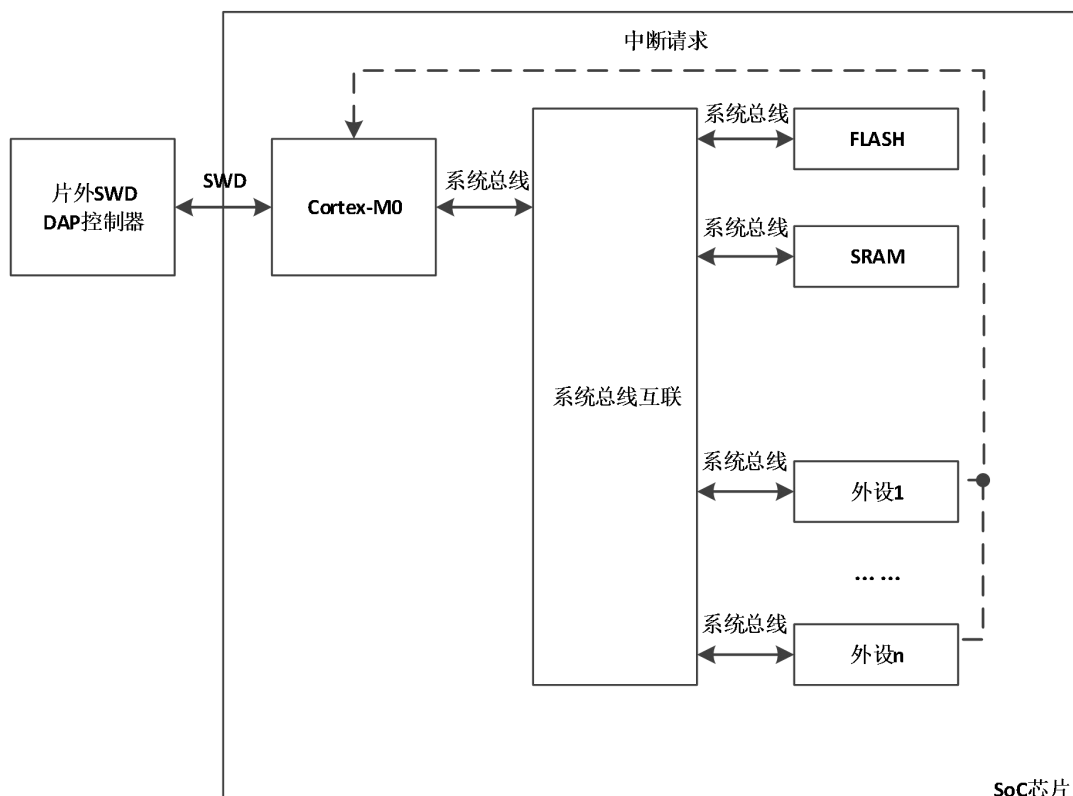
- 调试接口和资源访问；
- 可访问所有的从设备；

SoC 内置的从设备资源包括存储器（FLASH 和 SRAM）和各种外设（UART、定时器、看门狗等）。

部分外设可发起中断请求，如 UART、定时器等。

部分外设可发起 DMA 请求，如 UART、LCD 等。

图 4-1 SoC 设备物理互联架构



4.2 Cortex-M0 处理器

Cortex-M0 处理器是一个为嵌入式系统应用设计的 32 位处理器，具有如下特性：

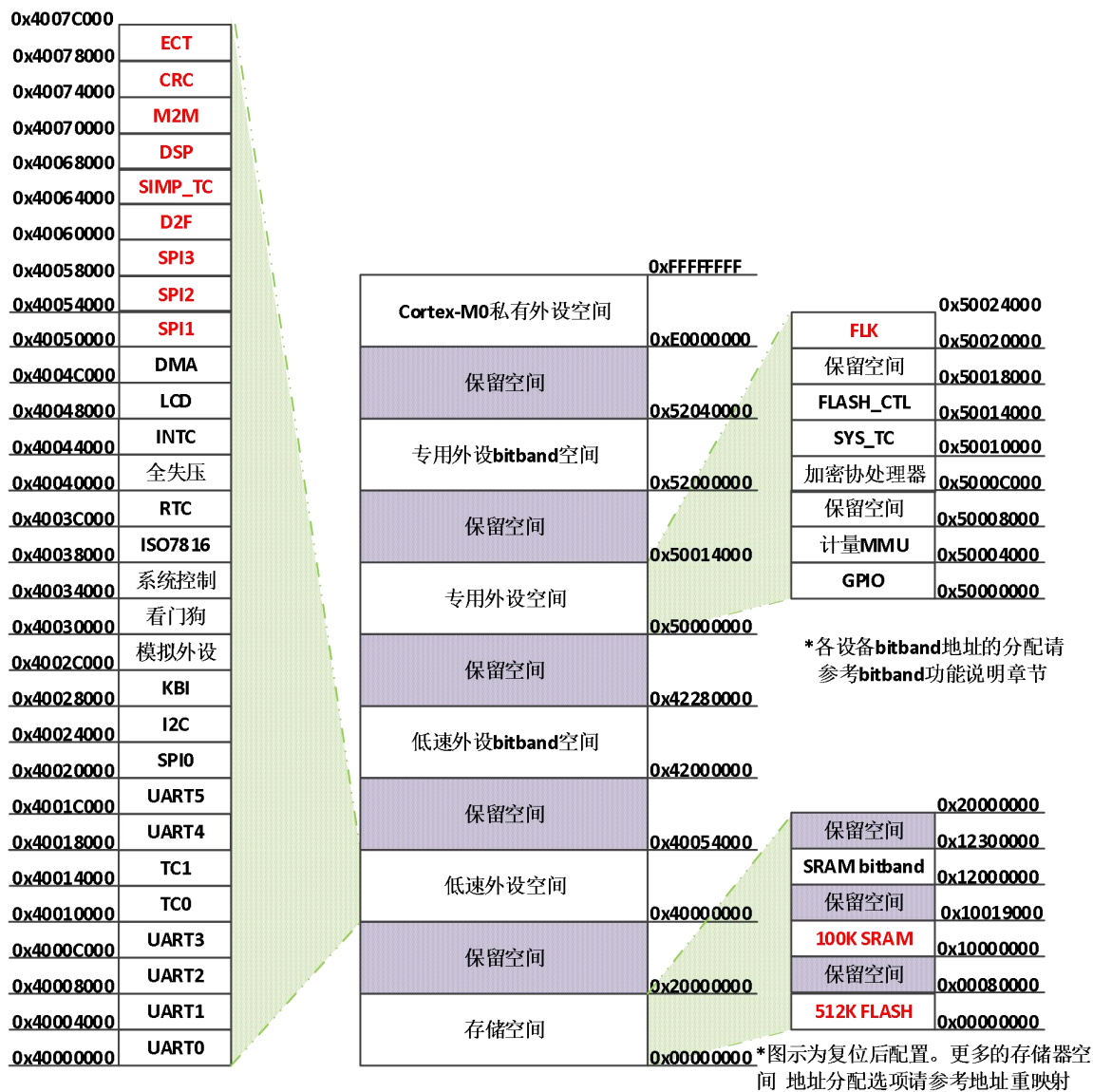
- ◎ 简便易用的程序模型
- ◎ 高代码集成度，具有 32 位的性能
- ◎ 工具和二进制代码与 Cortex-M 处理器系列向上兼容，方便升级和扩展
- ◎ 集成了极低功耗的睡眠模式
- ◎ 高效的代码执行允许处理器时钟更低，或者延长睡眠模式的时间
- ◎ 单周期 32 位硬件乘法器
- ◎ 零抖动中断处理
- ◎ 中断定时确定，中断处理效率高
- ◎ 支持中断/异常嵌套和抢占
- ◎ 支持 24 位系统节拍计数器
- ◎ 提供 4 个中断优先级
- ◎ 支持 2 个观察点，4 个硬件断点
- ◎ 支持串行调试接口（SWD），实现处理器内部状态高度可视和可控

有关 Cortex-M0 的详细资料可参阅 ARM 文档。

4.3 存储映射

SoC 的存储映射请参考“图 4-2SoC 地址空间映射”。

图 4-2 SoC 地址空间映射



4.3.1 存储重映射

SoC 支持对 2 个存储器，包括 FLASH 和 SRAM 的地址空间进行地址重新映射。

存储重映射操作通过配置系统控制器中的寄存器 SYS_CTL 的 REMAP 位域完成。

外设的地址分配均不受存储重映射的影响。

表 4-1 存储重映射配置

存储器设备	REMAP	映射地址
FLASH	0	0x00000000~0x0007FFFF
	1	0x00000000~0x0007FFFF
	2	0x10000000~0x1007FFFF
	3	保留，不可用

SRAM	0	0x10000000~0x10018FFF
	1	0x10000000~0x10018FFF
	2	0x00000000~0x00018FFF
	3	保留，不可用

4.3.2 Bitband 功能

对 bitband 区的访问等效于对外设寄存器中特定位的访问。

地址为 x 的存储单元的第 y 位对应的 bitband 地址：

$$Z = (X \& 0xFC000000) + 0x02000000 + (Y \ll 2) + ((X \ll 5) \& 0x03FFFFFF)$$

系统支持三个地址空间的 bitband 功能：

SRAM 空间：

。0x10000000~0x10018FFF 映射到 0x12000000~0x1231FFE0

外设空间：

。0x40000000~0x4004FFFF 映射到 0x42000000~0x429FFFE0；

。0x50000000~0x50003FFF 映射到 0x52000000~0x5207FFE0；（GPIO）

注意：外设空间对 bitband 支持并非全部，下表按模块列出支持 bitband 操作的寄存器：

外设名称	起始地址区间	映射起始地址	支持 bitband 操作的寄存器
SYS_TC	0x50010000	0x52200000	CTRL 寄存器
EMU	0x50004000	0x52080000	---
GPIO	0x50000000	0x52000000	除位操作（SET/CLR）以外寄存器
ETC	0x40080000	0x43000000	---
CRC	0x40074000	0x42e80000	CRC_CR
M2M	0x40070000	0x42e00000	CTRL、CLK、BDDIV 寄存器
DSP	0x40068000	0x42d00000	CTRL、CLK、BDDIV 寄存器支持
SIMP_TC	0x40064000	0x42c80000	CTRL 寄存器
D2F	0x40060000	0x42c00000	---
SPI3	0x40058000	0x42b00000	SPI_CTL 和 SPI_DMA_CTL 寄存器
SPI2	0x40054000	0x42a80000	SPI_CTL 和 SPI_DMA_CTL 寄存器
SPI1	0x40050000	0x42a00000	SPI_CTL 和 SPI_DMA_CTL 寄存器
DMA	0x4004c000	0x42980000	---
LCD	0x40048000	0x42900000	除了 LCD_STATUS 以外寄存器
INTC	0x40044000	0x42880000	---
全失压	0x40040000	0x42800000	
RTC	0x4003c000	0x42780000	RTC_CTRL、RTC_IE、RTC_IF 寄存器
7816	0x40038000	0x42700000	CTRL、CLK、BDDIV 寄存器
系统控制单元	0x40034000	0x42680000	模块内寄存器均支持
看门狗	0x40030000	---	---
模拟外设	0x4002c000	---	---
KBI	0x40028000	0x42500000	模块内寄存器均支持
I2C	0x40024000	0x42480000	CTL
SPI0	0x40020000	0x42400000	SPI_CTL 和 SPI_DMA_CTL 寄存器
UART5	0x4001c000	0x42380000	CTRL、CLK、BDDIV 寄存器支持
UART4	0x40018000	0x42300000	CTRL、CLK、BDDIV 寄存器支持

TC1	0x40014000	0x42280000	模块内寄存器均支持
TC0	0x40010000	0x42200000	模块内寄存器均支持
UART3	0x4000c000	0x42180000	CTRL、CLK、BDDIV 寄存器支持
UART2	0x40008000	0x42100000	CTRL、CLK、BDDIV 寄存器支持
UART1	0x40004000	0x42080000	CTRL、CLK、BDDIV 寄存器支持
UART0	0x40000000	0x42000000	CTRL、CLK、BDDIV 寄存器支持

4.3.3 SRAM

● 系统 SRAM

片内 SRAM 最大容量为 96KB，地址为 0x1000_0000~0x1001_8000，其中高 4KB 为 CACHE 使用，用户不要使用；

RAM 运行频率与处理器同频；

支持 8 位、16 位或 32 位数据随机访问，可用作代码或数据的存储。

● 算法 SRAM

片内含 4KB SRAM 作为加密模块算法 RAM，当加密模块不工作时，也可作为普通 RAM，被 CPU 访问，地址为 0x1001_8000~0x1001_9000。

WDT、外部管脚、软件复位等不会清除 SRAM 的数据，但是需要注意：BOOTROM 使用了地址空间 92KB~96KB，一旦系统发生复位，cpu 会从 BOOTROM 中执行启动程序，此地址空间的数据会被占用，使用此地址空间请注意该特性。

4.3.4 FLASH

SoC 内置最大 512KB FLASH：

- ◎ 最少 10 万擦写次数；
- ◎ 数据最少保存时间 20 年；
- ◎ 存储区包含 32 个块，每个块包含 32 个页，每个页包含 512Bytes
- ◎ 支持 8 位、16 位和 32 位随机读；
- ◎ 支持页擦除、块擦除、页编程，具体的操作需要调用锐能微库函数（nvm.a(IAR)/nvm.lib(KEIL)）
- ◎ 低功耗应用时，FLASH 会自动关闭或者开启；

库函数（nvm.a(IAR)/nvm.lib(KEIL)提供的 FLASH 操作函数接口如下：

uint8_t	flashPageErase(uint32_t pg)
uint8_t	flashSectorErase(uint32_t sec)
uint8_t	flashProgram(uint32_t dst_addr, uint32_t src_addr, uint32_t len)

详细的操作请见《SOC_MCU 应用笔记 002-库函数使用说明》。

4.4 中断分配

SoC 支持 32 个中断，其中开放有 8 个外部中断，分别为外部中断 0~7。

中断的详细信息，如优先级屏蔽寄存器、嵌套向量中断控制器（NVIC）等请参考 ARM-M0 手册。

表 4-2 中断/异常向量表及其配置信息

异常编	中断编	向量名称	中断向量地址	优先级
-----	-----	------	--------	-----

号	号			
-	-	MSP 初始值	0x00	-
1	-	复位	0x04	-3, 最高
2	-14	不可屏蔽中断	0x08	-2
3	-13	HARDFault 中断	0x0C	-1
4~10	-12~-6	保留	0x10~0x28	-
11	-5	系统调用	0x2C	可配置
12~13	-4~-3	保留	0x30~0x34	-
14	-2	PendSV	0x38	可配置
15	-1	系统节拍计数器	0x3C	可配置
16	0	系统控制/EMU_RCD	0x40	可配置
17	1	CMP/LVD	0x44	可配置
18	2	PWRSWH/EMU2	0x48	可配置
19	3	RTC	0x4C	可配置
20	4	EMU/D2F	0x50	可配置
21	5	MADC/FLK	0x54	可配置
22	6	UART0	0x58	可配置
23	7	UART1	0x5C	可配置
24	8	UART2	0x60	可配置
25	9	UART3	0x64	可配置
26	10	SPI0	0x68	可配置
27	11	I2C	0x6C	可配置
28	12	7816_0/SPI3	0x70	可配置
29	13	7816_1/SPI2	0x74	可配置
30	14	TC0	0x78	可配置
31	15	TC1	0x7C	可配置
32	16	UART4	0x80	可配置
33	17	UART5/LPUART	0x84	可配置
34	18	看门狗 WDT	0x88	可配置
35	19	KBI	0x8C	可配置
36	20	LCD/DSP 核	0x90	可配置
37	21	SEA/SYS_TC	0x94	可配置
38	22	EMU_DMA	0x98	可配置
39	23	NVM 全失压 / SPI1	0x9C	保留
40	24	外部中断 0/INTx(0~7 合并)	0xA0	可配置
41	25	外部中断 1/SIMP_TC0	0xA4	可配置
42	26	外部中断 2/SIMP_TC1	0xA8	可配置
43	27	外部中断 3/SIMP_TC2	0xAC	可配置
44	28	外部中断 4/SIMP_TC3	0xB0	可配置
45	29	外部中断 5/M2M	0xB4	可配置
46	30	外部中断 6/CRC	0xB8	可配置
47	31	外部中断 7/ECT	0xBC	可配置

注：实际栈顶比编译器分配的高两个 word，比如编译器分配的栈顶为 0x10001918，实际为 0x10001920；应用

时注意不要将这两个 word 分配给其他变量。

4.5 中断应用

在头文件中加入 SOC 头文件 `#include <RN8xxx_V2.h>`，即可使用 SOC 各中断，RN8xxx.h 文件中包含了 Cortex-M0 所定义的部分头文件，`core_cmFunc.h`、`core_cmFunc.h`、`core_cmInstr.h`。上述文件均可在锐能微公司提供的头文件中找到。

关闭中断使能： `__disable_irq()`;

使能总中断： `__enable_irq()`;

中断操作

各模块中断程序可完全使用 C 语言进行编写，用户无须考虑入栈及出栈问题，中断操作步骤如下，以 KBI 中断为例：

- 1、使能总中断： `__enable_irq()`;
- 2、配置需产生中断的模块，例如 KBI 模块，将 `KBI_MASK` 设置为中断使能。
- 3、使能 KBI 中断：在 RN8xxx.h 的文件中找到中断号并开启中断，例如 KBI 的中断号为 `KBI_IRQn`，开启 KBI 中断既为 `NVIC_EnableIRQ(KBI_IRQn)`，如需设置中断的优先级可使用 `void NVIC_SetPriority(IRQn_t IRQn, uint32_t priority)`。
- 4、编写中断服务函数，对不同的中断，函数名已固定，可在 `startup_RN821x.s` 的向量表中查找，如 KBI 中断服务程序函数名为 `KBI_HANDLER`，中断服务函数可写为：

```
void KBI_HANDLER(void)
{
    /* Start adding user code. Do not edit comment generated here */
}
```

- 5、关闭中断使能： `void NVIC_DisableIRQ0 (IRQn_t IRQn)`。

5 计量

5.1 主要特点

➤ 基本计量

- 3路 $\Sigma-\Delta$ ADC，内置 PGA，放大倍数可配置，其中 IA 通道最大支持 16 倍；
- 在 8000:1 动态范围内有功计量及无功计量误差小于 0.1%
- 参考电压温度系数典型值为 5ppm/°C
- 支持外灌基准电压，当外灌基准电压时，需要调用锐能微库函数完成相关配置。
- 支持零线和火线双通道有功功率、无功功率、视在功率、电流有效值同时测量
- 支持零线和火线双通道有功电能、无功电能、视在电能同时计量
- 提供电压有效值及电压线频率测量，频率更新速度 1 周波、32 周波可配
- 提供采样通道增益及 offset 校正功能
- 提供功率因数
- 提供全失压计量解决方案
- 提供直流计量解决方案
- 提供电压骤升、骤降事件监测；提供电流过载事件监测；提供谐波分析解决方案；
- 提供三路计量 sigma-delta ADC 原始采样数据，便于做二次算法开发。
- 提供智能安全用电解决方案
- 支持罗氏线圈
- 支持半波计量模式
- 支持双向计量模式
- 提供基波计量模式
- 电表常数可设置，且 3 套独立配置
- 支持 ADC 输入反向功能
- 支持 ADC 外灌功能
- 有功、无功支持 4 种电能累加方式：代数和、正向、绝对值、反向

➤ 同步采样波形输出

- 提供灵活的 ADC 同步采样波形数据
- 支持灵活的校正手段，支持增益校正、相位校正、谐波补偿；
- 支持通过 DMA 方式将波形数据从计量传输到 RAM。

➤ 电能质量

- 提供同步采样通道半波有功功率和半波有效值
- 提供半周期更新的全波电压电流有效值、全波有功功率
- 提供半周期更新的基波电压电流有效值、基波有功功率

注意 1：开启 ADC、对 ADC 的增益进行设置，以及开启计量时钟等配置参见系统控制章节。

5.2 寄存器列表

5.2.1 计量配置和状态寄存器列表

偏移	名称	R/W	字	复位值	功能描述	校验和
----	----	-----	---	-----	------	-----

地址			长			
校表参数和计量控制寄存器，基地址：0x50004000						
00H	EMUCON	R/W	3	1C0007	计量控制寄存器，写保护。	校验和 1
04H	EMUCON2	R/W	3	0	计量控制寄存器，写保护。	校验和 1
08H	HFConst	R/W	2	1000	脉冲频率寄存器，写保护。	校验和 1
0CH	PStart	R/W	2	60	有功起动功率设置，写保护。	校验和 1
10H	QStart	R/W	2	120	无功起动功率设置，写保护。	校验和 1
14H	GPQA	R/W	2	0	A 路功率增益校正寄存器，写保护。	校验和 1
18H	GPQB	R/W	2	0	B 路功率增益校正寄存器，写保护。	校验和 1
1CH	PhsA	R/W	2	0	通道 A 相位校正寄存器，校正刻度约为 0.01 度，写保护。校正范围有变化，详见功能介绍或寄存器说明。	校验和 1
20H	PhsB	R/W	2	0	通道 B 相位校正寄存器，校正刻度约为 0.01 度，写保护。校正范围有变化，详见功能介绍或寄存器说明。	校验和 1
24H	QPhsCal	R/W	2	0	无功相位补偿，写保护。	校验和 1
28H	APOSA	R/W	2	0	A 路有功功率 Offset 校正寄存器，写保护。	校验和 1
2CH	APOSB	R/W	2	0	B 路有功功率 Offset 校正寄存器，写保护。	校验和 1
30H	RPOSA	R/W	2	0	A 路无功功率 Offset 校正寄存器，写保护。	校验和 1
34H	RPOSB	R/W	2	0	B 路无功功率 Offset 校正寄存器，写保护。	校验和 1
38H	IARMSOS	R/W	2	0	电流通道 A 有效值 Offset 补偿，写保护。	校验和 1
3CH	IBRMSOS	R/W	2	0	电流通道 B 有效值 Offset 补偿，写保护。	校验和 1
40H	URMSOS	R/W	2	0	电压通道有效值 Offset 补偿，写保护。同 IARMSOS 和 IBRMSOS。对有效值、视在功率、视在电能起作用。	校验和 1
44H	IAGAIN	R/W	2	0	电流通道 A 增益设置，写保护；使用方法同 IBGAIN，对有效值、功率、电能均起作用。	校验和 1
48H	IBGAIN	R/W	2	0	电流通道 B 增益设置，写保护。	校验和 1
4CH	UGAIN	R/W	2	0	电压通道增益设置，写保护；使用方法同 IBGAIN，对有效值、功率、电能均起作用。	校验和 1
50H	IADCOS	R/W	3	0	电流通道 A 直流 Offset 校正，24bit，写保护。	校验和 1
54H	IBDCOS	R/W	3	0	电流通道 B 直流 Offset 校正，24bit，写保护。	校验和 1
58H	UDCOS	R/W	3	0	电压通道直流 Offset 校正，24bit，写	校验和 1

					保护。	
5CH	UADD	R/W	3	0	电压通道偏置寄存器，24bit，用于视在电能计量时电压写入固定值，只对视在电能有影响。对有功、无功、有效值无影响，写保护。	校验和 1
60H	USAG	R/W	2	0	电压跌落阈值设置，写保护；当值为 0 时不使能该功能；当写入不为 0 的值后启动跌落检测，检测结果有中断报出。	校验和 1
64H	IAPEAK	R/W	2	0	电流通道 A 峰值检测阈值设置，写保护。	校验和 1
68H	IBPEAK	R/W	2	0	电流通道 B 峰值检测阈值设置，写保护。	校验和 1
6CH	UPEAK	R/W	2	0	电压峰值检测阈值设置，写保护。	校验和 1
70H	D2FP	R/W	4	0	自定义功率寄存器，当 SADD=011 时，将功率值写入该寄存器，可以通过视在通道对写入的功率值进行积分计算电能。	不校验
74H	EMUCON3	R/W	3	0	计量控制寄存器 3，写保护。	校验和 2
78H	EMUCON4	R/W	3	F5F5	计量控制寄存器 4，用于配置第 2、3 套有功、无功能量累加模式，写保护。	校验和 2
7CH	EMUCON5	R/W	3	D34	计量控制寄存器 5，用于配置基波有功累加方式，写保护。	校验和 2
80H	CF_CFG	R/W	3	10543	脉冲输出配置寄存器，用于配置脉冲输出模式 1，写保护。	校验和 2
84H	HWRMS_CFG	R/W	3	0	半波有效值配置寄存器，对半周期更新的有效值进行配置，写保护。	校验和 2
88H	HWP_CFG	R/W	3	0	半波有功配置寄存器，对半周期更新的全波有功功率进行配置，写保护。	校验和 2
8CH	HWFP_CFG	R/W	3	0	半波基波有功配置寄存器，对半周期更新的基波有功功率进行配置，写保护。	校验和 2
90H	HWQ_CFG	R/W	3	0	半波无功配置寄存器，对半周期更新的无功功率进行配置，写保护。	校验和 2
94H	HFCnst2	R/W	2	1000	对快速脉冲计数 2 的常数进行设置，写保护。	校验和 2
98H	HFCnst3	R/W	2	1000	对快速脉冲计数 3 的常数进行设置，写保护。	校验和 2
9CH	ADCNEG_EN	R/W	3	0	ADC 输入反向使能配置寄存器，写保护。	校验和 2
A0H	EMUMODE	R/W	3	0	计量模式配置寄存器，写保护。	校验和 2
A4H	ATCHOP_CFG	R/W	3	0	自动模式配置寄存器寄存器，写保	校验和 2

					护。	
A8H	FGAIN	R/W	2	-	基波通道增益校正寄存器，写保护。	不校验
ACH	APOSFA	R/W	2	0	A 路基波有功功率 Offset 校正寄存器，写保护。	校验和 2
B0H	APOSFB	R/W	2	0	B 路基波有功功率 Offset 校正寄存器，写保护。	校验和 2
B4H	IAHWRMSOS	R/W	2	0	电流通道 A 半周期更新的有效值 Offset 校正寄存器，写保护。	校验和 2
B8H	IBHWRMSOS	R/W	2	0	电流通道 B 半周期更新的有效值 Offset 校正寄存器，写保护。	校验和 2
BCH	UHWMSOS	R/W	2	0	电压通道半周期更新的有效值 Offset 校正寄存器，写保护。	校验和 2
300H	ZXOTCFG	R/W	2	1C	过零计算配置和标志寄存器，写保护。	不校验
304H	ZXOTI	R/W	2	34	电流通道过零阈值寄存器，写保护。	校验和 2
308H	ZXOTU	R/W	2	2D0	电压过零阈值寄存器，写保护。	校验和 2
30CH	ROS_CTRL	R/W	2	0	罗氏线圈积分使能控制寄存器，写保护	校验和 2
310H	ROS_DCATT	R/W	2	7FDF	罗氏线圈积分直流衰减系数寄存器，写保护。	校验和 2
314H	ROS_TRANK	R/W	2	2CB	罗氏线圈积分转换系数寄存器，写保护。	不校验
318H	PQSRUN	R/W	3	0	电能累加使能寄存器，写保护。	校验和 2
330H	ECT_EN	R/W	4	0	ECT 对计量生效使能寄存器。 任意一个字节值满足 0xd9715a33， ECT_Gain 即对计量起作用。	不校验
334H	ECT_IAGAIN	R/W	2	0	IA 通道 ECT 增益寄存器。	不校验
338H	ECT_IBGAIN	R/W	2	0	IB 通道 ECT 增益寄存器。	不校验
33CH	ECT_UGAIN	R/W	2	0	U 通道 ECT 增益寄存器。	不校验
中断及 DMA 寄存器						
18CH	EMUIE	R/W	4	0	中断使能寄存器，写保护。	
190H	EMUIF	R/W	4	0	中断标志寄存器，写 1 清零。	
1A0H	EMUIE2	R/W	3	0	中断使能寄存器 2，写保护。	
1A4H	EMUIF2	R/W	3	0	中断标志寄存器 2，写 1 清零。	
1B0H	EMUIE3	R/W	3	0	中断使能寄存器 3，DMA 相关中断，写保护。	
1B4H	EMUIF3	R/W	3	0	中断标志寄存器 3，写 1 清零。	
194H	--	R/W	3	0	删除，保留地址	
状态寄存器						
188H	EMUStatus2	R	4	0	计量状态寄存器 2。	
198H	Rdata	R	4	--	上一次读出的数据。	
19CH	Wdata	R	4	--	上一次写入的数据。	
1ACH	EMUStatus3	R	3	FD5460	校验和寄存器 2。	

1B8H	---	R	3	0	保留	
特殊寄存器						
1A8H	SPCMD				特殊命令寄存器	

5.2.2 计量参数寄存器列表

偏移地址	名称	R/W	字长	复位值	功能描述
计量参数和状态寄存器，基地址：0x50004000					
C0H	PFBCnt	R/W	2	0	B 路快速有功脉冲计数，写保护。
C4H	QFBCnt	R/W	2	0	B 路快速无功脉冲计数，写保护。
C8H	SFBCnt	R/W	2	0	B 路快速视在脉冲计数，写保护。
E0H	PFBCnt2	R/W	2	0	B 路快速有功脉冲计数 2，写保护。
E4H	QFBCnt2	R/W	2	0	B 路快速无功脉冲计数 2，写保护。
E8H	PFBCnt3	R/W	2	0	B 路快速有功脉冲计数 3，写保护。
ECH	QFBCnt3	R/W	2	0	B 路快速无功脉冲计数 3，写保护。
F0H	PFACnt2	R/W	2	0	A 路快速有功脉冲计数 2，写保护。
F4H	QFACnt2	R/W	2	0	A 路快速无功脉冲计数 2，写保护。
F8H	PFACnt3	R/W	2	0	A 路快速有功脉冲计数 3，写保护。
FCH	QFACnt3	R/W	2	0	A 路快速无功脉冲计数 3，写保护。
100H	PFACnt	R/W	2	0	A 路快速有功脉冲计数，写保护。
104H	QFACnt	R/W	2	0	A 路快速无功脉冲计数，写保护。
108H	SFACnt	R/W	2	0	A 路快速视在脉冲计数，写保护。
10CH	IARMS	R	3	0	ADC 采样电流通道 A 有效值，更新速度 14.0625Hz。
110H	IBRMS	R	3	0	ADC 采样电流通道 B 有效值，更新速度 14.0625Hz。
114H	URMS	R	3	0	ADC 采样通道电压有效值，更新速度 14.0625Hz。
118H	UFREQ	R	2	0x2400	电压频率。更新速度有变化，详见功能介绍或寄存器说明。
11CH	PowerPA	R	4	0	A 路有功功率，更新速度 1.7578125Hz。
120H	PowerPB	R	4	0	B 路有功功率，更新速度 1.7578125Hz。
124H	PowerQA	R	4	0	A 路无功功率，更新速度 14.0625Hz。
128H	PowerQB	R	4	0	B 路无功功率，更新速度 14.0625Hz。
12CH	PowerSA	R	4	0	A 路视在功率，更新速度 14.0625Hz。
130H	PowerSB	R	4	0	B 路视在功率，更新速度 14.0625Hz。
134H	EnergyPA	R	3	0	A 路有功能量，读后清零或者不清零可配置，默认读后清零。
138H	EnergyPB	R	3	0	B 路有功能量，读后清零或者不清零可配置，默认读后清零。
13CH	EnergyQA	R	3	0	A 路无功能量，读后清零或者不清零可配置，默认读后清零。

140H	EnergyQB	R	3	0	B 路无功能量, 读后清零或者不清零可配置, 默认读后清零。
144H	EnergySA	R	3	0	A 路视在能量, 读后清零或者不清零可配置, 默认读后清零。
148H	PFA	R	3	0	A 路功率因数。
14CH	PFB	R	3	0	B 路功率因数。
150H	ANGLEA	R	2	0	基波电流通道 A 与基波电压夹角。
154H	ANGLEB	R	2	0	基波电流通道 B 与基波电压夹角。
15CH	SPL_IA	R	3	0	计量采样电流通道 A 高通滤波器后瞬时波形采样值, 更新速度 7.2KHz。
160H	SPL_IB	R	3	0	计量采样电流通道 B 高通滤波器后瞬时波形采样值, 更新速度 7.2KHz。
164H	SPL_U	R	3	0	计量采样电压通道高通滤波器后瞬时波形采样值, 更新速度 7.2KHz。
168H	PowerPA2	R	4	0	A 路有功功率 2, 更新速度 14.0625Hz。
16CH	PowerPB2	R	4	0	B 路有功功率 2, 更新速度 14.0625Hz。
170H	EnergySB	R	3	0	B 路视在能量, 读后清零或者不清零可配置, 默认读后清零。
174H	SPL_PA	R	3	0	A 路瞬时有功功率, 更新速度是 7.2KHz 或者半周期 (典型值 100Hz)。 半周期更新时切换为 32bit 输出, 24bit 精度不够。
178H	SPL_PB	R	3	0	B 路瞬时有功功率, 更新速度是 7.2KHz 或者半周期 (典型值 100Hz)。 半周期更新时切换为 32bit 输出, 24bit 精度不够。
17CH	SPL_QA	R	3	0	A 路瞬时无功功率, 更新速度是 7.2KHz 或者半周期 (典型值 100Hz)。 半周期更新时切换为 32bit 输出, 24bit 精度不够。
180H	SPL_QB	R	3	0	B 路瞬时无功功率, 更新速度是 7.2KHz 或者半周期 (典型值 100Hz)。 半周期更新时切换为 32bit 输出, 24bit 精度不够。
184H	---	R	4	0	保留, 只读。
200H	HW_RMSIA	R	3	0	半周期更新的基波电流通道 A 基波电流有效值, 更新速度典型值为 100Hz。
204H	HW_RMSIB	R	3	0	半周期更新的基波电流通道 B 基波电流有效值, 更新速度典型值为 100Hz。
208H	HW_RMSU	R	3	0	半周期更新的基波电压有效值, 更新速度典型值为 100Hz。
20CH	HW_FPA	R	4	0	半周期更新的 A 路基波有功功率, 更新速度典型值为 100Hz。

210H	HW_FPB	R	4	0	半周期更新的 B 路基波有功功率，更新速度典型值为 100Hz。
214H	SPL_IA2	R	3	0	计量采样电流通道 A 高通滤波前的瞬时波形采样值，更新速度 14.4KHz。
218H	SPL_IB2	R	3	0	计量采样电流通道 B 高通滤波前的瞬时波形采样值，更新速度 14.4KHz。
21CH	SPL_U2	R	3	0	ADC 采样电压通道高通滤波前的瞬时波形采样值，更新速度 7.2KHz。
22CH	SPL_FIA	R	3	0	基波电流通道 A 瞬时基波波形采样值，更新速度 7.2KHz。
230H	SPL_FIB	R	3	0	基波电流通道 B 瞬时基波波形采样值，更新速度 7.2KHz。
234H	SPL_FU	R	3	0	基波电压通道瞬时基波波形采样值，更新速度 7.2KHz。
238H	EnergyPA2	R	3	0	A 路有功能量 2，读后清零或者不清零可配置，默认读后清零。
23CH	EnergyPB2	R	3	0	B 路有功能量 2，读后清零或者不清零可配置，默认读后清零。
240H	EnergyQA2	R	3	0	A 路无功能量 2，读后清零或者不清零可配置，默认读后清零。
244H	EnergyQB2	R	3	0	B 路无功能量 2，读后清零或者不清零可配置，默认读后清零。
248H	EnergyPA3	R	3	0	A 路有功能量 3，读后清零或者不清零可配置，默认读后清零。
24CH	EnergyPB3	R	3	0	B 路有功能量 3，读后清零或者不清零可配置，默认读后清零。
250H	EnergyQA3	R	3	0	A 路无功能量 3，读后清零或者不清零可配置，默认读后清零。
254H	EnergyQB3	R	3	0	B 路无功能量 3，读后清零或者不清零可配置，默认读后清零。
258H	IADCOS_Calc	R	3	0	AUTO DC 电流通道 A 直流 Offset 测量值寄存器。
25CH	IBDCOS_Calc	R	3	0	AUTO DC 电流通道 B 直流 Offset 测量值寄存器。
260H	UDCOS_Calc	R	3	0	AUTO DC 电压通道直流 Offset 测量值寄存器。

5.2.3 同步采样寄存器列表

基地址：0x40040080，无校验和

偏移地址	名称	R/W	字长	复位值	功能描述	校验和
校表参数和计量控制寄存器，基地址：0x40040080						
00H	WAVE_WKEY	R/W	1	0	同步采样写密码。	

04H	WAVECFG	R/W	3	0	同步采样通道配置寄存器, 写保护。	
08H	WAVECFG2	R/W	3	0	同步采样通道配置寄存器, 写保护。	
0CH	WAVE_EN	R/W	1	0	同步采样使能寄存器, 写保护。	
10H	WAVECNT	R/W	2	0	同步采样通道采样率控制寄存器, 写保护。	
14H	WAVE_DC_EN	R/W	2	0	同步采样通道 AUTO DC 使能寄存器, 写保护。	
18H	WAVE_PhsIA	R/W	2	0	同步采样电流通道 A 相位校正寄存器, 写保护。	
1CH	WAVE_PhsIB	R/W	2	0	同步采样电流通道 B 相位校正寄存器, 写保护。	
20H	WAVE_PhsU	R/W	2	0	同步采样电压通道相位校正寄存器, 写保护。	
24H	WAVE_IAGain	R/W	2	0	同步采样电流通道 A 增益寄存器, 写保护。	
28H	WAVE_IBGain	R/W	2	0	同步采样电流通道 B 增益寄存器, 写保护。	
2CH	WAVE_UGain	R/W	2	0	同步采样电压通道增益寄存器, 写保护。	
50H	WAVE_HW_RMSIA	R	3	0	同步采样电流通道 A 半波有效值	
54H	WAVE_HW_RMSIB	R	3	0	同步采样电流通道 B 半波有效值	
58H	WAVE_HW_RMSU	R	3	0	同步采样电压通道半波有效值	
5CH	WAVE_HW_PA	R	4	0	同步采样通道 A 路半波有功功率	
60H	WAVE_HW_PB	R	4	0	同步采样通道 B 路半波有功功率	

5.2.4 DMA 波形缓存寄存器列表

基地址: 0x50004000

校表参数和计量控制寄存器, 基地址: 0x40040080						
偏移地址	名称	R/W	字长	复位值	功能描述	校验和
400H	DMA_WAVE_CFG	R/W	2	1C	DMA 波形缓存配置寄存器, 写保护。	校验和 2
404H	DMA_BUF_CTRL	R/W	1	0	DMA 波形缓存使能寄存器, 写保护。	校验和 2
408H	DMA_BUF_BADDR	R/W	2	0	DMA 波形缓存基址寄存器, 写保护。	校验和 2
40CH	DMA_BUF_DEPTH	R/W	2	008F	DMA 波形缓存深度寄存器, 写保护。	校验和 2
410H	DMA_GAP_CFG	R/W	3	0	通道间隙配置寄存器, 写保护。	校验和 2
414H	DMA_BUF_ADDR	R	2	0	当前 DMA 指针地址。	不校验
418H	DMA_ERR_ADDR	R	2	0	DMA 波形缓存错误地址寄存器。	不校验
41CH	DMA_CHECKSUM	R	3	0	DMA 波形校验和寄存器。	不校验
420H	DMA_RCD_CFG	R/W	3	0	剩余电流 DMA 波形缓存配置寄存器, 写保护。	校验和 2
424H	DMA_BUF_RCD_CTL	R/W	3	0	剩余电流 DMA 波形缓存使能寄存器, 写保护。	校验和 2

428H	DMA_BUF_RCD_BA DDR	R/W	3	0	剩余电流 DMA 数据缓存目标地址的 偏移地址，写保护。	校验和 2
42CH	DMA_BUF_RCD_D EPTH	R/W	3	0	剩余电流 DMA 数据缓存 Bank 区块 的大小，写保护。	校验和 2
430H	DMA_BUF_RCD_AD DR	R	2	0	剩余电流 当前 DMA 指针地址。	不校验
434H	DMA_BUF_RCD_ER R_ADDR	R	2	0	剩余电流 DMA 波形缓存错误地址 寄存器。。	不校验

5.2.5 智能微断 RCD 寄存器列表

基地址：0x50004000，无校验和

偏移 地址	名称	R/W	字长	复位值	功能描述	
智能微断 RCD						
480H	RCD_CTRL	R/W	4	0xF0235	RCD 控制寄存器	
484H	RCD_EN	R/W	4	0	RCD 使能控制寄存器	
488H	RCD_THRE	R/W	4	0x6CA024 F	RCD 输入信号阈值寄存器	
48CH	RCD_ATTHRE	R/W	4	0xC800C8	RCD 方案 A 阈值寄存器	
490H	RCD_BTTHRE	R/W	4	0x4C	RCD 方案 B 脱扣阈值寄存器	
494H	RCD_ACNT	R	4	0	RCD 方案 A 积分器输出结果寄存 器	
498H	RCD_BCNT	R	4	0	RCD 方案 B 积分器输出结果寄存 器	
49CH	RCD_IE	R/W	4	0	RCD 中断使能寄存器	
4A0H	RCD_IF	R/W	4	0	RCD 中断标志寄存器	
4A4H	RCD_STA	R	4	0	RCD 状态寄存器	
智能微断脱扣信号发生器						
4B0H	TRIG_CTRL	R/W	4	0x7	TRIG 控制寄存器	
4B4H	TIRG_EN	R/W	4	0	通用脱扣软件使能控制寄存器	
4B8H	TRIG_STOP	R/W	4	0	脱扣结束控制寄存器	
4BCH	TRIG_LEN	R/W	4	0x12001	通用脱扣信号长度寄存器	
4C0H	TRIG_DLY	R/W	4	0	通用脱扣启动延时寄存器	
4C4H	TRIG_STA	R	4	0	通用脱扣状态寄存器	
4C8H	TRIG_LEN2	R/W	4	0x12001	专用硬件脱扣长度寄存器	
4CCH	TRIG_STA2	R	4	0	专用硬件脱扣状态寄存器	

5.3 寄存器描述

5.3.1 计量配置寄存器

5.3.1.1 计量控制寄存器 EMUCON(0x0)

计量控制寄存器 EMUCON

偏移地址：00H；字长：3 字节；默认值：0x1C0007

位	名称	功能描述	R/W	复位值
31:24	--	只读，不可写	R	0
23	Cf2_cfg2	与 EMUCON2.CF2_CFG 配合使用，决定 QF 引脚输出哪种电能脉冲。	R/W	0
22	Cf1_cfg2	与 EMUCON2.CF1_CFG 配合使用，决定 PF 引脚输出哪种电能脉冲。	R/W	0
21	Sag_Freq_sel	=0: SAG (电压骤降) 周期为 50Hz 应用， =1: SAG (电压骤降) 周期为 60Hz 应用	R/W	0
20	SBRUN	=1: 使能 B 路视在电能寄存器累加； =0: 关闭 B 路视在电能寄存器累加。	R/W	1
19	QBRUN	=1, 使能 B 路无功电能寄存器累加； =0, 关闭 B 路无功电能寄存器累加。	R/W	1
18	PBRUN	=1: 使能 B 路有功电能寄存器累加； =0: 关闭 B 路有功电能寄存器累加。	R/W	1
17	CF3_CFG	=0: 原 SF 引脚=SFA； =1: 原 SF 引脚=SFB。	R/W	0
16	U_start	=0: 视在电能计算时电压通道参与计算； =1: 视在电能计算时电压通道不参与计算，只有 UADD 参与。	R/W	0
15:14	QMOD	无功能量累加方式选择： =00, 代数和方式，正反向功率都参与累加，负功率有 REVQ 符号指示； =01, 正向方式，只累加正向功率； =10, 绝对值方式，正反向功率取绝对值后都参与累加，无负功率符号指示； =11, 负向方式，只累加负向功率。	R/W	0
13:12	PMOD	有功能量累加方式选择： =00, 代数和方式，正反向功率都参与累加，负功率有 REVP 符号指示； =01, 正向方式，只累加正向功率； =10, 绝对值方式，正反向功率取绝对值后都参与累加，无负功率符号指示； =11, 负向方式，只累加负向功率。	R/W	0
11	ZXD1	ZX 输出初始值为 0，根据 ZXD1 和 ZXD0 的配置输出不同的过零波形 ZX_OUT： ZXD1=0, 表示仅在选择的过零点处 ZX 输出发生变化； ZXD1=1, 表示在正向和负向过零点处 ZX 输出均发生变化。	R/W	0
10	ZXD0	ZXD0=0, 表示选择正向过零点作为过零检测信号； ZXD0=1, 表示选择负向过零点作为过零检测信号。	R/W	0
9	Energy_clr	=0: 所有电能寄存器读后清零； =1: 所有电能寄存器读后不清零。	R/W	0

8	HPFIBOFF	=0: 使能 IB 通道数字高通滤波器; =1: 关闭 IB 通道数字高通滤波器。	R/W	0
7	HPFIAOFF	=0: 使能 IA 通道数字高通滤波器; =1: 关闭 IA 通道数字高通滤波器。	R/W	0
6	HPFUOFF	=0: 使能 U 通道数字高通滤波器; =1: 关闭 U 通道数字高通滤波器。	R/W	0
5	CFSUEN	CFSUEN 是 PF/QF 脉冲输出加速模块的控制位, =0, 关闭脉冲加速模块, 脉冲正常输出; =1, 使能脉冲加速模块, 脉冲的输出速率提高 $2^{(CFSU[1:0]+1)}$ 倍。(验证时注意负功率)	R/W	0
4:3	CFSU	该位和 CFSUEN 配合使用, 见 CFSUEN 说明。	R/W	0
2	SRUN	=1, 使能 SF 脉冲输出和视在电能寄存器累加; =0, 关闭 SF 脉冲输出和视在电能寄存器累加。 默认状态为 1。	R/W	1
1	QRUN	=1, 使能 QF 脉冲输出和无功电能寄存器累加; =0, 关闭 QF 脉冲输出和无功电能寄存器累加。 默认状态为 1。	R/W	1
0	PRUN	=1, 使能 PF 脉冲输出和有功电能寄存器累加; =0, 关闭 PF 脉冲输出和有功电能寄存器累加。 默认状态为 1。 配置 PRUN/QRUN/SRUN/PBRUN/QBRUN/SBRUN 为 1, 并且对任何计量寄存器发起一次读操作或对寄存器 PQSRUN.bit4~bit23 发生写操作, 计量才会真正开启; 如果只是对 PRUN/QRUN/SRUN/PBRUN/QBRUN/SBRUN 写 1, 那么不会开始进行计量。 这样可以让多个积分器能够同时进行积分, 保证多套积分器的积分一致性, 同时也可实现软件向下兼容, 因为 MCU 一定会在配置完成后发起读操作。	R/W	1

5.3.1.2 计量控制寄存器 EMUCON2(0x4)

计量控制寄存器 EMUCON2

偏移地址: 04H; 字长: 3 字节; 默认值: 0x0

位	名称	功能描述	R/W	复位值
31:24	--	只读, 不可写	R	0
23:21	SADD[2:0]	通道 A 视在电能通道功率输入选择 SADD[2:0]: =000, S=SA 或 SB, 由 CHNSEL 决定; =001, S=SB; =010, S=QB; =011, S=自定义功率; =100, S=+SA+SB; =101, S=+SA-SB;	R/W	0

		=110, S=-SA+SB; =111, S=-SA-SB。		
20:18	QADD[2:0]	通道 A 无功电能通道功率输入选择 QADD[2:0]: =000, Q=QA 或 QB, 由 CHNSEL 决定; =001, Q=PA; =010, Q=PB; =011, Q= QA + QB ; =100, Q=+QA+QB; =101, Q=+QA-QB; =110, Q=-QA+QB; =111, Q=-QA-QB。	R/W	0
17:15	PADD[2:0]	通道 A 有功电能通道功率输入选择 PADD[2:0]: =000, P=PA 或 PB, 由 CHNSEL 决定; =001, P=PA; =010, P=PB; =011, P= PA + PB ; =100, P=+PA+PB; =101, P=+PA-PB; =110, P=-PA+PB; =111, P=-PA-PB。	R/W	0
14:13	CF2_CFG[1:0]	与 cf2_cfg2 配合使用, 决定 P51/QF 引脚的电能脉冲输出选择 [cf2_cfg2, cf2_cfg]: =000, 原 QF 引脚=QFA; =001, 原 QF 引脚=QFB; =010, 原 QF 引脚=SFA; =011, 原 QF 引脚=SFB; =100, 原 QF 引脚=PFA; =101, 原 QF 引脚=PFB。	R/W	0
12:11	CF1_CFG[1:0]	与 cf1_cfg2 配合使用, 决定 P50/PF 引脚的电能脉冲输出选择, [cf1_cfg2, cf1_cfg]: =000, 原 PF 引脚=PFA; =001, 原 PF 引脚=PFB; =010, 原 PF 引脚=QFA; =011, 原 PF 引脚=QFB; =100, 原 PF 引脚=SFA; =101, 原 PF 引脚=SFB。 备注: PFA/QFASFA 分别对应电能寄存器 EnergyPA/ EnergyQA/ EnergySA; PFB/QFB/SFB 分别对应电能寄存器 EnergyPB/ EnergyQB/ EnergySB。	R/W	0
10:3	usag_cfg[7:0]	usag_cfg[7:0]用于配置电压跌落检测的半周期	R/W	0

		数。		
2	u_dc_en	u_dc_en、ib_dc_en、ia_dc_en 写 1 使能通道直流 Offset1 测量，测量完成后自动清零。 直流 Offset 计算模式不同，实现的功能也不同，详细功能描述见 DC Offset 校正相关描述。	R/W	0
1	ib_dc_en		R/W	0
0	ia_dc_en		R/W	0

5.3.1.3 计量控制寄存器 EMUCON3(0x74 新增)

计量控制寄存器 EMUCON3

偏移地址：74H；字长：3 字节；默认值：0x0

位	名称	功能描述	R/W	复位值
31:18	--	只读，不可写。	R	0
17	AUTO_DC_MODE	AUTO DC 模式控制寄存器： =0，自动模式，测量完成后，测量值自动写入计量通道直流 Offset 校正寄存器 0x38~0x40，默认对计量通道和同步采样通道均起作用，可单独配置令其对计量通道和同步采样通道不起作用。 =1，手动模式，测量完成后，测量值仅写入直流 Offset 测量值参数寄存器 0x258~0x260，软件判断直流 Offset 测量值的可靠性后软件填写直流 Offset 校正寄存器。 归一化直流 Offset 计算公式为： $RegValue/2^{23}$ 。	R/W	0
16	Bypass_EMU_DCOS_EN	Bypass DC Offset 校正使能位： =0：不使能，DC Offset 校正有效，即 UDCOS/IADCOS/IBDCOS 寄存器的值影响计量通道； =1：使能，DC Offset 校正无效，即 UDCOS/IADCOS/IBDCOS 寄存器的不影响计量通道。	R/W	0
15:14	--	只读，不可写。	R	0
13	ZX_CFG	根据 ZX_CFG 和 EMUCON.ZXD0 的配置 UZX/IAZX/IBZX 输出不同更新周期的过零中断： =1，表示仅在 ZXD0 选择的过零点处输出过零中断； =0，表示在正向和负向过零点处均输出过零中断。	R/W	0
12	ZX_WAVE_SEL	UZX/IAZX/IBZX 的过零检测的波形数据源： =0，选择为计量采样通道波形数据 SPL_U/SPL_IA/SPL_IB；		

		=1, 选择为基波波形数据 SPL_FU/SPL_FIA/SPL_FIB。		
11	rmshw_zx_sel	半波有效值过零源头选择: =0, 基波过零, 默认; =1, 全波过零。	R/W	0
10	Lpf_10Hz_en	10Hz 低通滤波器使能配置位: =0, 不使能; =1, 使能, 直流计量使用, 此时基波滤波器替换成带宽 10Hz 的低通滤波器, 同时需配置 FGain_CalcAutoDis=1。	R/W	0
9	FGain_CalcAutoDis	=0, 使能基波通道自动增益校正功能, 此时基波通道增益校正寄存器可写, 但写入的值不起作用; =1, 除能基波通道自动增益校正功能, 软件填写基波通道增益校正值。	R/W	0
8	FreqCnt	FreqCnt 决定频率寄存器更新周期: =0, 频率更新周期 32 周波, 向下兼容; =1, 频率更新周期 1 周波。	R/W	0
4:3	QB_MOD	B 路无功 1 累加方式选择, 仅在 PQMOD_ABINDEP=1 时有效: =00, 代数和累加; =01, 正向累加; =10, 绝对值累加; =11, 反向累加。	R/W	00
7:5	--	只读, 不可写。	R	0
2:1	PB_MOD	B 路有功 1 累加方式选择, 仅在 PQMOD_ABINDEP=1 时有效: =00, 代数和累加; =01, 正向累加; =10, 绝对值累加; =11, 反向累加。	R/W	00
0	PQMOD_ABINDEP	A、B 路有功 1、无功 1 累加方式独立配置使能位: =0, 不使能, EMUCON.PMOD/QMOD 控制 2 路有功 1/无功 1 累加方式; =1, 使能, EMUCON.PMOD/QMOD 控制 A 路有功 1/无功 1 累加方式, EMUCON3.PB_MOD/QB_MOD 控制 B 路有功 1/无功 1 累加方式。	R/W	0

5.3.1.4 计量控制寄存器 EMUCON4(0x78 新增)

计量控制寄存器 EMUCON4

偏移地址: 78H; 字长: 3 字节; 默认值: 0xF5F5

位	名称	功能描述	R/W	复位值
31:16	--	只读，不可写	R	0
15:14	QB3_MOD	B 路无功 3 累加方式选择：功能描述见 EPA1_MOD。	R/W	11
13:12	PB3_MOD	B 路有功 3 累加方式选择：功能描述见 EPA1_MOD。	R/W	11
11:10	QB2_MOD	B 路无功 2 累加方式选择：功能描述见 PA2_MOD。	R/W	01
9:8	PB2_MOD	B 路有功 2 累加方式选择：功能描述见 PA2_MOD。	R/W	01
7:6	QA3_MOD	A 路无功 3 累加方式选择：功能描述见 PA2_MOD。	R/W	11
5:4	PA3_MOD	A 路有功 3 累加方式选择：功能描述见 PA2_MOD。	R/W	11
3:2	QA2_MOD	A 路无功电能 2 累加方式选择：功能描述见 PA2_MOD。	R/W	01
1:0	PA2_MOD	A 路有功电能 2 累加方式选择： =00，代数和累加； =01，正向累加； =10，绝对值累加； =11，反向累加。	R/W	01

5.3.1.5 计量控制寄存器 EMUCON5(0x7C 新增)

计量控制寄存器 EMUCON5，主要用于基波累加方式配置。

偏移地址：7CH；字长：3 字节；默认值：0xD34

位	名称	功能描述	R/W	复位值
31:12	--	只读，不可写	R	0
11:10	FPB3_MOD	基波 B 路有功 3 累加方式选择，功能描述见 FPA1_MOD。	R/W	11
9:8	FPB2_MOD	基波 B 路有功 1 累加方式选择，功能描述见 FPA1_MOD。	R/W	01
7:6	FPB1_MOD	基波 B 路有功 1 累加方式选择，功能描述见 FPA0_MOD。	R/W	00
5:4	FPA3_MOD	基波 A 路有功 2 累加方式选择，功能描述见 FPA1_MOD。	R/W	11
3:2	FPA2_MOD	基波 A 路有功 1 累加方式选择，功能描述见 FPA1_MOD。	R/W	01
1:0	FPA1_MOD	基波 A 路有功 1 累加方式选择： =00，代数和累加； =01，正向累加； =10，绝对值累加； =11，反向累加。	R/W	00

5.3.1.6 电能累加使能寄存器 PQSRUN (0x318 新增)

偏移地址：318H；字长：3 字节；默认值：0x0

位	名称	功能描述	R/W	复位值
31:24	--	只读，不可写	R	0
23	FPB3_AutoCalc	B 路基波有功电能 3 自动输出控制位，功能描述见 FPA1_AutoCalc，使用 D2FP5 单元。	R/W	0
22	FPB2_AutoCalc	B 路基波有功电能 2 自动输出控制位，功能描述见 FPA1_AutoCalc，使用 D2FP4 单元。	R/W	0
21	FPB1_AutoCalc	B 路基波有功电能 1 自动输出控制位，功能描述见 FPA0_AutoCalc，使用 D2FP3 单元。	R/W	0
20	FPA3_AutoCalc	A 路基波有功电能 3 自动输出控制位，功能描述见 FPA1_AutoCalc，使用 D2FP2 单元。	R/W	0
19	FPA2_AutoCalc	A 路基波有功电能 2 自动输出控制位，功能描述见 FPA1_AutoCalc，使用 D2FP1 单元。	R/W	0
18	FPA1_AutoCalc	A 路基波有功电能 1 自动输出控制位： =0，关闭； =1，使能，硬件可自动读取 32bit SPL_FPB 寄存器值填入 32bit 的 D2FP0 寄存器进行积分，输出电能和脉冲。 注意：该模式下 D2FP0 不支持软件手动写入。	R/W	0
17	SBRUN	B 路视在电能累加使能位： =0，关闭； =1，使能。	R/W	0
16	SARUN	A 路视在电能累加使能位： =0，关闭； =1，使能。	R/W	0
15	QBRUN3	B 路无功电能 3 累加使能位： =0，关闭； =1，使能。	R/W	0
14	QBRUN2	B 路无功电能 2 累加使能位： =0，关闭； =1，使能。	R/W	0
13	QBRUN	B 路无功电能 1 累加使能位： =0，关闭； =1，使能。	R/W	0
12	QARUN3	A 路无功电能 3 累加使能位： =0，关闭； =1，使能。	R/W	0
11	QARUN2	A 路无功电能 2 累加使能位： =0，关闭； =1，使能。	R/W	0

10	QARUN	A 路无功电能 1 累加使能位： =0，关闭； =1，使能。	R/W	0
9	PBRUN3	B 路有功电能 3 累加使能位： =0，关闭； =1，使能。	R/W	0
8	PBRUN2	B 路无功电能 2 累加使能位： =0，关闭； =1，使能。	R/W	0
7	PBRUN	B 路有功电能 1 累加使能位： =0，关闭； =1，使能。	R/W	0
6	PARUN3	A 路有功电能 3 累加使能位： =0，关闭； =1，使能。	R/W	0
5	PARUN2	A 路有功电能 2 累加使能位： =0，关闭； =1，使能。	R/W	0
4	PARUN	A 路有功电能 1 累加使能位： =0，关闭； =1，使能。	R/W	0
3:1	--	只读，不可写	R	0
0	RUN_SEL	第 1 套 A 路和 B 路有功/无功/视在电能累加使能位选择： =0，默认向下兼容，使用 EMUCON 寄存器控制； =1，使用 PQSRUN 寄存器控制。		

5.3.1.7 脉冲输出配置寄存器 CF_CFG(0x80 新增)

脉冲输出控制寄存器 CF_CFG

偏移地址：80H；字长：3 字节；默认值：0x10543

位	名称	功能描述	R/W	复位值
31:20	--	只读，不可写。	R	0
19:16	CF_OUT4_CFG	CF_OUT4 输出的脉冲类型选择寄存器： 配置选项同 CF_OUT0_CFG，默认输出 PFA 脉冲。	R/W	0001
15:12	CF_OUT3_CFG	CF_OUT3 输出的脉冲类型选择寄存器： 配置选项同 CF_OUT0_CFG，默认输出 PFA 脉冲。	R/W	0000
11:8	CF_OUT2_CFG	CF_OUT2 输出的脉冲类型选择寄存器： 配置选项同 CF_OUT0_CFG，默认输出 SFB 脉	R/W	0101

		冲。		
7:4	CF_OUT1_CFG	CF_OUT1 输出的脉冲类型选择寄存器： 配置选项同 CF_OUT0_CFG，默认输出 QFB 脉冲。	R/W	0100
3:0	CF_OUT0_CFG	CF_OUT0 输出的脉冲类型选择寄存器： =0000，PFA 脉冲； =0001，QFA 脉冲； =0010，SFA 脉冲； =0011，PFB 脉冲； =0100，QFB 脉冲； =0101，SFB 脉冲； =0110，PFA2 脉冲； =0111，QFA2 脉冲； =1000，PFA3 脉冲； =1001，QFA3 脉冲； =1010，PFB2 脉冲； =1011，QFB2 脉冲； =1100，PFB3 脉冲； =1101，QFB3 脉冲； =其他，保留。	R/W	0011

5.3.1.8 半波有效值配置寄存器 HWRMS_CFG(0x84 新增)

半波有效值配置寄存器 HWRMS_CFG

偏移地址：84H；字长：3 字节；默认值：0x0

位	名称	功能描述	R/W	复位值
31:15	保留	只读，不可写	R	0
14:13	HW_RMS_SEL	=00：选择半周期更新的 3 路有效值数据计算源头为 ADC 采样通道波形数据 SPL_U/SPL_IA/SPL_IB； =01：选择半周期更新的 3 路有效值数据计算源头为 ADC 采样通道波形数据 SPL_U2/SPL_IA2/SPL_IB2； =10：选择半周期更新的 3 路有效值数据计算源头为基波波形数据； =11：保留。	R/W	0
12	HW_RMS_MODE	=0：3 路有效值按照 HW_RMS_NUM 配置的固定采样点数累加求平均并开方计算； =1：3 路有效值采用过零驱动进行累加并求平均并开方计算。	R/W	0
11:9	保留	只读，不可写	R	0
8:0	HW_RMS_NUM	用于半周期更新的有效值累加求平均计算： = 9'b0，按半周波 72 点累加求平均计算； = 9'b1，按半周期 1 点累加求平均计算；	RW	9'b0

		= 9'b1x, 按半周期 2 点累加求平均计算; = 9'b1xx, 按半周期 4 点累加求平均计算; = 9'b1xxx, 按半周期 8 点累加求平均计算; = 9'b1xxxx, 按半周期 16 点累加求平均计算; = 9'b1xxxxx, 按半周期 32 点累加求平均计算; = 0x40~0x1FF, 若期望的半周期累加求平均点数为 N, 则 NUM = N, 由软件决定半周波采样点数 N。		
--	--	--	--	--

5.3.1.9 半波有功配置寄存器 HWP_CFG (0x88 新增)

半波有功配置寄存器 HWP_CFG

偏移地址: 88H; 字长: 3 字节; 默认值: 0x0

位	名称	功能描述	R/W	复位值
31:14	保留	只读, 不可写	R	0
13	P_D2F_SEL	=0: ADC 采样通道有功功率及电能的计算源头为瞬时功率, 更新速度为 7.2KHz; =1: ADC 采样通道有功功率及电能的计算源头为半周期更新的功率, 更新速度典型值为 100Hz。	R/W	0
12	HW_P_MODE	=0: 半周期更新的 2 路 ADC 采样通道有功功率采用固定采样点数进行累加求平均; =1: 半周期更新的 2 路 ADC 采样通道有功功率采用过零驱动进行累加并求平均。	R/W	0
11:9	保留	只读, 不可写	R	0
8:0	HW_P_NUM	用于半周期更新的 ADC 采样通道有功功率累加求平均计算: = 9'b0, 按半周波 72 点累加求平均计算; = 9'b1, 按半周期 1 点累加求平均计算; = 9'b1x, 按半周期 2 点累加求平均计算; = 9'b1xx, 按半周期 4 点累加求平均计算; = 9'b1xxx, 按半周期 8 点累加求平均计算; = 9'b1xxxx, 按半周期 16 点累加求平均计算; = 9'b1xxxxx, 按半周期 32 点累加求平均计算; = 0x40~0x1FF, 若期望的半周期累加求平均点数为 N, 则 NUM = N, 由软件决定半周波采样点数 N。	R/W	9'b0

5.3.1.10 半波基波有功配置寄存器 HWFP_CFG (0x8C 新增)

半波基波有功配置寄存器 HWFP_CFG

偏移地址: 8CH; 字长: 3 字节; 默认值: 0x0

位	名称	功能描述	R/W	复位值
31:13	保留	只读, 不可写	R	0

12	HW_FP_MODE	=0: 半周期更新的 2 路基波有功功率采用固定采样点数进行累加求平均; =1: 半周期更新的 2 路基波有功功率采用过零驱动进行累加并求平均。	R/W	0
11:9	保留	只读, 不可写	R	0
8:0	HW_FP_NUM	用于半周期更新的基波有功功率累加求平均计算: = 9'b0, 按半周波 72 点累加求平均计算; = 9'b1, 按半周期 1 点累加求平均计算; = 9'b1x, 按半周期 2 点累加求平均计算; = 9'b1xx, 按半周期 4 点累加求平均计算; = 9'b1xxx, 按半周期 8 点累加求平均计算; = 9'b1xxxx, 按半周期 16 点累加求平均计算; = 9'b1xxxxx, 按半周期 32 点累加求平均计算; = 0x40~0x1FF, 若期望的半周期累加求平均点数为 N, 则 NUM = N, 由软件决定半周波采样点数 N。	R/W	9'b0

5.3.1.11 半波无功配置寄存器 HWQ_CFG (0x90 新增)

半波无功配置寄存器 HWQ_CFG

偏移地址: 90H; 字长: 3 字节; 默认值: 0x0

位	名称	功能描述	R/W	复位值
31:15	保留	只读, 不可写	R	0
14	Q_SEL	=0: A、B 路无功为 ADC 采样通道无功; =1: A、B 路无功为基波无功。		
13	Q_D2F_SEL	=0: 无功功率及电能的计算源头为瞬时功率, 更新速度为 7.2KHz; =1: 无功功率及电能的计算源头为半周期更新的功率, 更新速度典型值为 100Hz。	R/W	0
12	HW_Q_MODE	=0: 半周期更新的 2 路无功功率采用固定采样点数进行累加求平均; =1: 半周期更新的 2 路无功功率采用过零驱动进行累加并求平均。	R/W	0
11:9	保留	只读, 不可写	R	0
8:0	HW_Q_NUM	用于半周期更新的无功功率累加求平均计算: = 9'b0, 按半周波 72 点累加求平均计算; = 9'b1, 按半周期 1 点累加求平均计算; = 9'b1x, 按半周期 2 点累加求平均计算; = 9'b1xx, 按半周期 4 点累加求平均计算; = 9'b1xxx, 按半周期 8 点累加求平均计算; = 9'b1xxxx, 按半周期 16 点累加求平均计算; = 9'b1xxxxx, 按半周期 32 点累加求平均计算; = 0x40~0x1FF, 若期望的半周期累加求平均点	R/W	9'b0

		数为 N，则 NUM = N，由软件决定半周波采样点数 N。		
--	--	--------------------------------	--	--

5.3.1.12 脉冲频率寄存器 HFConst(0x8)

脉冲频率寄存器 HFConst

偏移地址：08H；字长：2 字节；默认值：0x1000

位	名称	功能描述	R/W	复位值
31:16	---	预留	R	0
15:0	HFConst	HFConst 是 16 位无符号数，当快速脉冲计数寄存器 FCntx 计数值的绝对值的 2 倍大于等于 HFConst，即 $2* FCntx \geq HFConst$ 时，相应会有脉冲溢出，能量寄存器的值会相应的加 1。 HFConst 对应快速脉冲寄存器（有功、无功、视在）。	R/W	1000

5.3.1.13 脉冲频率寄存器 HFConst2(0x94 新增)

脉冲频率寄存器 HFConst2

偏移地址：94H；字长：2 字节；默认值：0x1000

位	名称	功能描述	R/W	复位值
31:16	---	预留	R	0
15:0	HFConst2	HFConst2 是 16 位无符号数，当快速脉冲计数寄存器 FCntx 计数值的绝对值的 2 倍大于等于 HFConst，即 $2* FCntx \geq HFConst$ 时，相应会有脉冲溢出，能量寄存器的值会相应的加 1。 HFConst2 对应快速脉冲寄存器 2（有功 2、无功 2）；	R/W	1000

5.3.1.14 脉冲频率寄存器 HFConst3(0x98 新增)

脉冲频率寄存器 HFConst3

偏移地址：98H；字长：2 字节；默认值：0x1000

位	名称	功能描述	R/W	复位值
31:16	---	预留	R	0
15:0	HFConst3	HFConst3 是 16 位无符号数，当快速脉冲计数寄存器 FCntx 计数值的绝对值的 2 倍大于等于 HFConst，即 $2* FCntx \geq HFConst$ 时，相应会有脉冲溢出，能量寄存器的值会相应的加 1。 HFConst3 对应快速脉冲寄存器 3（有功 3、无功 3）；	R/W	1000

5.3.1.15 ADC 输入反向使能寄存器 ADCNEG_EN(0x9C 新增)

ADC 输入反向使能寄存器。

偏移地址：9CH；默认值：0x0

位	名称	功能描述	R/W	复位值
---	----	------	-----	-----

31:24	WKEY	写密码寄存器： =0xEA，低 24 位可写； =其他值，低 24 位不可写。	WO	0
23:6	--	只读，不可写。	R	
5	U_ADCCLK	U 通道 ADC_CLK 通道采样沿选择寄存器： =0，上升沿； =1，下降沿。	R/W	0
4	IB_ADCCLK	IB 通道 ADC_CLK 通道采样沿选择寄存器： =0，上升沿； =1，下降沿。	R/W	0
3	IA_ADCCLK	IA 通道 ADC_CLK 通道采样沿选择寄存器： =0，上升沿； =1，下降沿。	R/W	0
2	U_Neg_EN	U 通道 ADC 输入反向使能寄存器： =0，无操作； =1，反向。	R/W	0
1	IB_Neg_EN	IB 通道 ADC 输入反向使能寄存器： =0，无操作； =1，反向。	R/W	0
0	IA_Neg_EN	IA 通道 ADC 输入反向使能寄存器： =0，无操作； =1，反向。 备注：用于 ADC 输入反接的情况。若 ADC 输入反接，配置 ADC 输入反向可令采样信号符号正确，抵消 ADC 输入反接的错误。	R/W	0

5.3.1.16 计量模式 EMUMODE(0xA0 新增)

计量模式配置寄存器，默认值：0x0

位	位名称	功能描述	R/W	复位值
31:24	WKEY	写密码寄存器： =0xEA，低 24 位可写； =其他值，低 24 位不可写。	WO	0
23:2	--	只读，不可写。	R	0
1	DC_SEL	直流计量数据源头选择寄存器： =0，选择 14.4KHz 更新的波形； =1，选择 7.2KHz 更新的波形。	R/W	0
0	EMUMODE	计量模式配置寄存器： =0，交流计量模式；	R/W	0

=1，直流计量模式。

5.3.1.17 自动直流计量模式配置 ATCP_CFG(0xA4 新增)

自动模式配置寄存器，默认值：0x50

位	位名称	功能描述	R/W	复位值
31:10	--	只读，不可写。	R	0
9:4	Del_Num	ADC 输入信号正反向采样切换时删除点数 Del_Num: =0，不删除； =1~63，删除 1~63 点。 备注：推荐 Del_Num≥5，在正相切反向和反向切正向时均删除 Del_Num 个采样点。 例如：配置配置 DC_SEL=0, chop_div=2048, Ave_Num=16，此时为保证 Ave_Num 是 chop_div 的整周波，宜配置 Del_Num=6。（此时每个 chop_div 周期剩余 4 个点去做平均，每 4 个周期计量采样通道更新一次 SPL_Px，更新速度 225Hz）	R/W	5
3	--	只读，不可写。	R	0
2:0	Ave_Num	直流计量采样通道平均点数配置寄存器： =0，不平均； =1，16 点平均； =2，32 点平均； =3，64 点平均； =4，128 点平均； =5，256 点平均； =6，512 点平均； =7，1024 点平均。	R/W	0

5.3.1.18 潜动与启动阈值寄存器(0x0C/0x10)

有功潜动与启动阈值寄存器 PStart

偏移地址：0CH；字长：2 字节；默认值：0x0060

位	名称	功能描述	R/W	复位值
31:16	---	预留	R	0
15:0	PStart	有功潜动与启动阈值寄存器	R/W	0060

无功潜动与启动阈值寄存器 QStart

偏移地址：10H；字长：2 字节；默认值：0x0120

位	名称	功能描述	R/W	复位值
31:16	---	预留	R	0
15:0	QStart	无功潜动与启动阈值寄存器	R/W	0120

启动阈值可由 PStart 和 QStart 寄存器配置。它们是 16 位无符号数，做比较时，将其分别与 PowerP 和 PowerQ（为 32bit 有符号数）的高 24 位的绝对值进行比较，以作起动判断。

|PowerP|<PStart 时，PF 不输出脉冲。

|PowerQ|<QStart 时，QF 不输出脉冲。

当|PowerP|<PStart 且|PowerQ|<QStart 时，SF 也不输出脉冲。

5.3.1.19 功率增益校正寄存器(0x14/0x18)

A 路功率增益校正寄存器 GPQA

偏移地址：14H；字长：2 字节；默认值：0x0000

位	名称	功能描述	R/W	复位值
31:16	---	预留	R	0
15:0	GPQA	A 路功率增益校正寄存器	R/W	0

B 路功率增益校正寄存器 GPQB

偏移地址：18H；字长：2 字节；默认值：0x0000

位	名称	功能描述	R/W	复位值
31:16	---	预留	R	0
15:0	GPQB	B 路功率增益校正寄存器	R/W	0

A 路功率增益校正寄存器 GPQA 和 B 路功率增益校正寄存器 GPQB，为 16 位二进制补码格式，最高位为符号位。

GPQA 用于电流通道 IA 和电压通道有功/无功/视在功率的校正。GPQB 用于电流通道 IB 和电压通道有功/无功的增益校正。

校正公式为：P1=P0(1+GPQA)

Q1=Q0(1+GPQA)

S1=S0(1+GPQA)

其中 GPQA 为增益校正寄存器的归一化值，范围是±1。所以增益校正的校正范围是：对功率最多放大 2 倍（*2），缩小可以缩到 0（*0）。

5.3.1.20 通道相位校正寄存器(0x1C/0x20 修改)

通道 A 相位校正寄存器 PhsA

偏移地址：1CH；字长：2 字节；默认值：0x0

位	名称	功能描述	R/W	复位值
31:9	---	预留	R	0
8:0	PhsA	通道 A 相位校正寄存器	R/W	0

通道 B 相位校正寄存器 PhsB

偏移地址：20H；字长：2 字节；默认值：0x0

位	名称	功能描述	R/W	复位值
31:9	---	预留	R	0
8:0	PhsB	通道 B 相位校正寄存器	R/W	0

相位校正寄存器包括 IA 和 U 通道的相位校正 PhsA 以及 IB 和 U 通道的相位校正 PhsB。这两个寄存器均为带符号二进制补码，bit0~bit8 有效，其中 bit8 为符号位。

相位校正刻度：50Hz 下，1 LSB 代表 0.009766°相位校正

相位校正范围：50Hz 下，±2.5°

V2 版新增功能：

当 PhsA[31:0] (0x30) 的高 8 位 PhsA[31:24] 被写入 0xE5 时, 所有通道相位校正寄存器的意义发生改变, 相位校正刻度不变, 相位校正范围由 $\pm 2.5^\circ$ 提升到 $\pm 10^\circ$ 。

具体定义如下:

PhsA 为 IA 和 U 通道相位校正, 每当 PhsA[31:24] 写入 0xE5 时, PhsA[8:0] 有效 (bit8 为符号位) 变为 PhsA[10:0] 有效 (bit10 为符号位), 低 3 字节 bit[23:0] 默认值不变, 不影响校验和。

操作步骤: 首先将 PhsA[31:24] 写入 0xE5, 此时读 PhsA[31:24] 寄存器值为 0xA5, 表示扩位成功; 然后再写入相位校正值 (比如 PhsA[31:0]=0xE5000215), 注意要分两次操作。在新的模式下, 应保证 PhsA[31:24] 写入值一直为 0xE5。

PhsB 为 IB 和 U 通道相位校正, PhsA[31:24] 已经写入 0xE5 的情况下, PhsB[8:0] 有效 (bit8 为符号位) 变为 PhsB[10:0] 有效 (bit10 为符号位), 低 3 字节 bit[23:0] 默认值不变, 不影响校验和。

操作步骤: 首先将 PhsA[31:24] 写入 0xE5; 然后再写入相位校正值 (比如 PhsB[10:0]=0x215), 注意要分两次操作。

相位校正分两级, 组合实现 $\pm 10^\circ$ 校正:

相位校正 1 校正源头为 1bit 数据, 50Hz 时 1 LSB 代表 0.009766° 相位校正, 可校正范围 $\pm 2.5^\circ$;

相位校正 2 校正源头为 14.4KHz 更新的波形, 可校正范围 $\pm 7.5^\circ$ 。

5.3.1.21 无功相位补偿寄存器 QPhsCal (0x24)

无功相位补偿寄存器 QPhsCal

偏移地址: 24H; 字长: 2 字节; 默认值: 0x0000

位	名称	功能描述	R/W	复位值
31:16	---	预留	R	0
15:0	QPhsCal	无功相位补偿寄存器	R/W	0

无功相位补偿寄存器用于 U 通道 90° 移相滤波器在无功计算中的相位补偿。无功相位补偿寄存器采用十六位二进制补码形式, 最高位为符号位。

校正公式: $Q2 = Q1 - QPhs * P1$

其中 P1 为有功功率, Q1 为补偿前的无功功率, Q2 为补偿后的无功功率。

两路无功功率使用相同的相位补偿寄存器, 校正时分别乘以对应的有功功率。

5.3.1.22 功率 Offset 校正寄存器 (0x28 0x2C/0x30 0x34/0xAC 0xB0 新增)

偏移地址	28H	2CH	30H	34H	ACH	B0H
寄存器	APOSA	APOSB	RPOSA	RPOSB	APOSFA	APOSFB
默认值	0	0	0	0	0	0

功率 Offset 校正适合小信号的精度校正。

功率 Offset 校正寄存器分三类: 有功功率 OFFSET 校正寄存器、无功功率 Offset 校正寄存器、基波有功功率 OFFSET 校正寄存器。

APOSA 寄存器为 A 路有功功率 Offset 校正值; APOSB 寄存器为 B 路有功功率 Offset 校正值。

RPOSA 寄存器为 A 路无功功率 Offset 校正值; RPOSB 寄存器为 B 路无功功率 Offset 校正值。

APOSFA 寄存器为 A 路基波有功功率 Offset 校正值; APOSFB 寄存器为 B 路基波有功功率 Offset 校正值。

寄存器均为 16 位二进制补码格式, 最高位为符号位。

校正公式:

以 A 路有功功率为例, 假设校正之前 A 路有功功率寄存器 PA, 校正之后为 PA', 则有:

$$PA'=PA+PA_OS。$$

5.3.1.23 有效值 Offset 校正寄存器(0x38~40/0xB4~BC 新增)

偏移地址	38H	3CH	40H	B4H	B8H	BCH
寄存器	IARMSOS	IBRMSOS	URMSOS	IAHWRMSOS	IBHWRMSOS	UHW RMSOS
默认值	0	0	0	0	0	0

有效值 Offset 校正寄存器用于有效值小信号精度的校正。

IARMSOS 寄存器为电流通道 A 有效值 Offset 值,IBRMSOS 寄存器为电流通道 B 有效值 Offset 值,URMSOS 寄存器为电压通道有效值 Offset 值。

IAHWRMSOS、IBHWRMSOS、UHW RMSOS 为半周期更新的有效值 Offset 校正寄存器。

寄存器均为 16 位二进制补码格式，最高位为符号位。

校正公式：

以电流通道 A 为例，假设校正之前有效值寄存器 IARMS，校正之后为 IARMS'，则有：

$$IARMS'=Sqrt(Abs(IARMS^2+IARMS_OS*2^8))。$$

5.3.1.24 通道增益校正寄存器(0x44~4C/0xA8 新增)

偏移地址	44H	48H	4CH	A8H
寄存器	IAGain	IBGain	UGain	FGain
默认值	0	0	0	0

IAGain、IBGain、UGain 分别为 IA、IB、U 三路 ADC 通道增益校正寄存器，，对有效值、功率、电能均起作用。

FGain 为 IA、IB、U 基波通道增益寄存器，主要对 60Hz 应用进行补偿，对基波有效值、基波有功功率起作用。

通道增益寄存器均为 16 位二进制补码格式，最高位为符号位，表示范围(-1,+1)。

如果 $RegGain \geq 2^{15}$ ，则 $Gain=(RegGain-2^{16})/2^{15}$ ；否则 $Gain=RegGain/2^{15}$ ；其中 RegGain 为通道增益寄存器值。

以 IB 通道为例，假设校正之前 B 路电流有效值 IB，校正之后 IB'，两者关系为：

$$IB'=IB+IB*Gain$$

注意该公式有效范围：保证在乘以通道增益后信号仍在 ADC 满量程范围内。

基波通道增益校正功能可配置寄存器 EMUCON3.FGain_CalcAutoDis 选择自动校正或软件手动校正：

EMUCON3.FGain_CalcAutoDis=0，使能基波通道自动增益校正功能，此时基波通道增益校正寄存器不可写；

EMUCON3.FGain_CalcAutoDis=1，除能基波通道自动增益校正功能，用户软件自行计算基波通道增益校正值，填入基波通道增益校正寄存器。

注意：0x9C 寄存器不参与校验和计算，由用户软件自行管理。

5.3.1.25 通道直流 Offset 校正寄存器(0x50~58/0x258~260 新增)

偏移地址	50H	54H	58H
寄存器	IADCOS	IBDCOS	UDCOS
默认值	0	0	0

通道直流 Offset 校正和参数寄存器均为 24bit 有符号数，最高位为符号位。

支持三个通道的直流 Offset 校正(0x50~0x58),用于不需要高通滤波器或需提高小信号精度或罗氏线圈等场合。

偏移地址	258H	25CH	260H
寄存器	IADCOS_Calc	IBDCOS_Calc	ICDCOS_Calc
默认值	0	0	0

IADCOS_Calc、IBDCOS_Calc、UDCOS_Calc 为 AUTO DC 直流 Offset 测量值寄存器，只读。

5.3.1.26 电压通道偏置寄存器 UADD(0x5C)

电压通道偏置寄存器 UADD

偏移地址：5CH；字长：3 字节；默认值：0x0

位	名称	功能描述	R/W	复位值
31:24	---	预留	R	0
23:0	UADD	电压通道偏置寄存器	R/W	0

电压通道偏置寄存器，24位二进制补码格式，最高位是符号位。

使用场景：视在电能计量时电压写入固定值，只对视在电能有影响。对有功、无功、有效值无影响。

UADD 的最小刻度是有效值寄存器的 1.414 倍。

5.3.1.27 事件阈值寄存器(0x60~6C)

偏移地址	60H	64H	68H	6CH
寄存器	USAG	IAPEAK	IBPEAK	UPEAK
默认值	0	0	0	0

事件阈值寄存器包括：电压跌落阈值寄存器 USAG、电流通道 A 峰值检测阈值寄存器、电流通道 B 峰值检测阈值寄存器、电压通道峰值检测阈值寄存器。

电压跌落阈值寄存器，16 位无符号数。当值为 0 时不使能该功能；当写入不为 0 的值后启动跌落检测，将阈值与 U 通道 ADC 的 24 位波形采样值的高 16 位进行比较，检测的半周期数由 EMUCON2.usag_cfg[7:0]决定，EMUCON.Sag_Freq_sel=0，sag 为周期 50Hz 应用，1 个半周期时间固定为 10ms；EMUCON.Sag_Freq_sel=1，sag 为周期 60Hz 应用，1 个半周期时间固定为 8.333ms。当跌落持续时间超过 usag_cfg，触发电压跌落事件，检测结果有中断报出。

峰值检测阈值寄存器，16 位无符号数。当值为 0 时不使能该功能；当写入不为 0 的值后启动跌落检测，将阈值与 ADC 的 24 位波形采样值的高 16 位进行比较，当检测到波形采样值大于阈值，触发过载事件，检测结果有中断报出。

5.3.1.28 自定义功率寄存器 D2FP(0x70)

自定义功率寄存器 D2FP

偏移地址：70H；字长：4 字节；默认值：0x0

位	名称	功能描述	R/W	复位值
31:0	D2FP	自定义功率寄存器	R/W	0

D2FP 为 32bit 有符号数，最高位为符号位。

当 EMUCON2.SADD=011 时，将功率值写入该寄存器，可以通过视在通道对写入的功率值进行积分计算电能。

5.3.1.29 过零配置和标志寄存器 ZXOTCFG(0x300 新增)

过零配置和标志寄存器 ZXOTCFG

偏移地址：300H；字长：2 字节；默认值：0x1C

位	名称	功能描述	R/W	复位值
31:5	保留	保留	R	0
4:2	ZXFLAG	只读位，存放每个通道的阈值比较标志。 ZXOTCFG=0,标志为 1, 不进行有效值与过零阈值比较； ZXOTCFG=1, ZXFLAG 指示有效值与过零阈值的比较结果。ZXFLAG=1 表示该通道大于过零计算阈值；ZXFLAG=0 表示小于阈值。 ZXFLAG 通道顺序为{IB/IA/UA}。	R	7
1	ZXOT_CFG	ZXIA/ZXIB/ZXU 通道过零输出控制寄存器： =0 时，一直进行过零判断，计量采样通道波形符号发生改变，即产生过零信号； =1 时，以阈值比较结果为准选择是否进行过零判断，输出过零信号。	R/W	0
0	ZXOTU_CFG	ZX 过零（包括测频、夹角）输出控制寄存器： =0，向下兼容，电压有效值寄存器 URMS<2 ¹⁶ （归一化值 7.813mV），不输出过零 ZX_OUT，不产生 ZX 过零中断，不测频，不计算相角。 =1，以阈值 ZXOTU 和电压有效值 URMS 比较结果为准选择电流通道是否输出过零信号。	R/W	0

注意：该寄存器不参与校验和计算，由用户软件自行管理。

5.3.1.30 过零阈值寄存器 ZXOT(0x304/0x308 新增)

过零阈值寄存器 ZXOTI

偏移地址：304H；字长：2 字节；默认值：34

位	名称	功能描述	R/W	复位值
15:0	ZXOTI	电流通道(包括 2 路)过零阈值寄存器	R/W	34

过零阈值寄存器 ZXOTU

偏移地址：308H；字长：2 字节；默认值：0x2D0

位	名称	功能描述	R/W	复位值
15:0	ZXOTU	电压通道过零及测频阈值寄存器	R/W	0x2D0

功能介绍详见功能说明章节过零检测。

5.3.1.31 罗氏线圈控制寄存器 ROS_CTRL(0x30C 新增)

罗氏线圈控制寄存器 ROS_CTRL

偏移地址：30CH；字长：2 字节；默认值：0

位	名称	功能描述	R/W	复位值
31:2	--	只读，不可写	R	0

1	RosIB_EN	=0, 关闭 IB 通道罗氏线圈积分; =1, 使能 IB 通道罗氏线圈积分。	R/W	0
0	RosIA_EN	=0, 关闭 IA 通道罗氏线圈积分; =1, 使能 IA 通道罗氏线圈积分。	R/W	0

5.3.1.32 罗氏线圈积分直流衰减系数寄存器 ROS_DCATTTC(0x310 新增)

罗氏线圈积分直流衰减系数寄存器 ROS_DCATTTC

偏移地址: D0H; 字长: 2 字节; 默认值: 0x7FDF

位	名称	功能描述	R/W	复位值
15:0	Ros_DCATTTC	电流通道罗氏线圈积分直流衰减系数寄存器	R/W	7FDF

该寄存器仅在罗氏线圈积分使能情况下有效。

5.3.1.33 罗氏线圈积分转换系数寄存器 ROS_TRANK(0x314 新增)

罗氏线圈积分转换系数寄存器 ROS_TRANK

偏移地址: 314H; 字长: 2 字节; 默认值: 0x2CB

位	名称	功能描述	R/W	复位值
15:0	tran_k	该寄存器仅在罗氏线圈积分使能情况下有效。 应用时客户根据公式软件填写该寄存器, 该寄存器不参与校验, 由用户软件管理。 计算公式: $trans_k = 2 * \pi * freq / 14400$ (freq 为基波频率, 单位 Hz) 默认值 freq=50, trans_k=0x2CB。	R/W	2CB

5.3.1.34 ECT 对计量生效使能寄存器 ECT_EN(0x330)

位	名称	功能描述	R/W	复位值
31:0	ECT_EN_PS	ECT_EN_PS[7:0]=0x33 或 ECT_EN_PS[15:8]=0x5A 或 ECT_EN_PS[23:16]=0x71 或 ECT_EN_PS[31:24]=0xD9 任意一个字节满足, ECT_GAIN 值对计量起作用。 建议全配置, ECT_EN_PS[31:0]=0xD9715a33	R/W	0

5.3.1.35 ECT 增益寄存器 ECT_GAIN(0x334~0x33C)

334H	338H	33CH
ECT_IAGAIN	ECT_IBGAIN	ECT_UGAIN

为 16 位二进制补码格式, 最高位为符号位。

校正公式为: $IA' = IA(1 + ECT_IAGAIN)$

$IB' = IB(1 + ECT_IBGAIN)$

$U' = U(1 + ECT_UGAIN)$

5.3.2 计量参数寄存器

5.3.2.1 快速脉冲计数器寄存器(0xC0~C8/0x100~108/0xE0~FC 新增)

偏移地址	C0H	C4H	C8H	E0H	E4H	E8H	ECH
寄存器	PFBCnt	QFBCnt	SFBCnt	PFBCnt2	QFBCnt2	PFBCnt3	QFBCnt3
默认值	0	0	0	0	0	0	0

偏移地址	100H	104H	108H	F0H	F4H	F8H	FCH
寄存器	PFACnt	QFACnt	SFACnt	PFACnt2	QFACnt2	PFACnt3	QFACnt3
默认值	0	0	0	0	0	0	0

有功、无功快速脉冲寄存器新增 2 套快速脉冲计数器分别对应新增的 2 套能量单元。

快速脉冲计数器寄存器均为 2 字节有符号数，可读可写。

为了防止上下电时丢失电能，掉电时 MCU 将寄存器 PFCntx/QFCntx/SFCntx 值读回并进行保存，然后在下次上电时 MCU 将这些值重新写入到 PFCntx/QFCntx/SFCntx 中去。

当快速脉冲计数寄存器 PFCntx/QFCntx/SFCntx 计数值的绝对值的 2 倍大于等于 HFConst 倍时，相应会有脉冲溢出，能量寄存器的值会相应的加 1。

5.3.2.2 有效值寄存器(0x10C~114/0x200~208 新增)

偏移地址	10CH	110H	114H	200H	204H	208H
寄存器	IARMS	IBRMS	URMS	HW_RMSIA	HW_RMSIB	HW_RMSU
默认值	0	0	0	0	0	0

ADC 采样通道有效值 IARMS、IBRMS、URMS，更新速度为 14.0625Hz，稳定时间约 300ms。

新增半波有效值 HW_RMSIA、HW_RMS_IB、HW_RMSU。

配置 HWRMS_CFG.HW_RMS_SEL 选择半波有效值的计算源头：

=0，3 路有效值计算源头为全波数据；

=1，3 路有效值计算源头为基波数据。

配置 HWRMS_CFG.HW_RMS_MODE 选择半波有效值计算模式：

=0，3 路有效值按照 HWRMS_CFG.HW_RMS_NUM 配置的固定采样点数累加求平均并开方计算；

=1，3 路有效值采用过零驱动进行累加求平均并开方计算。

有效值寄存器均为 24 位有符号数，最高位为 0 表示有效数据，最高位为 1 时读数做零处理，归一化值 =RMSReg/2²³。有效值寄存器值 RMSReg 与输入到芯片采样管脚正端 V_p 和负端 V_n 之间待测电压信号有效值 RMS_{Spn} 的关系为：

$$\text{RMSReg} = (\text{Vpn} * \text{PGA} / 1) * 2^{23}$$

RMSReg：有效值寄存器值；

RMS_{Spn}：测量通道正端 V_p 和负端 V_n 之间待测电压信号有效值，单位为 V；要求输入信号瞬时值不能超过 1V，如输入信号为正弦交流信号时，信号峰值即有效值乘 $\sqrt{2}$ 不能大于 1V；

PGA：测量通道内部 ADC 增益配置值；

假设输入的正弦波电压峰值为 A=1V（有效值为 A*0.70710678），内部 ADC 增益为 1，那么有效值寄存器值约为：

$A \cdot 0.70710678 \cdot 2^{23} = 32 \cdot d5931641$ (十进制) $= 32 \cdot h5A8279$ (十六进制)

ADC PGA 配置、高通使能配置、电流通道增益寄存器 GSI/U 配置对上述电流有效值有效。有效值 OFFSET 寄存器对上述有效值有效。不论高通是否关闭，通道直流 OFFSET 校正寄存器 DCOSx 对有效值均有效。

对有效值参数均有，实际电压电流有效值 $= K_{rmsx} \cdot RMSReg'$

其中 K_{rmsx} 表示转换系数， $RMSReg'$ 表示上述有效值寄存器 $RMSReg$ 的补码。该运算由 MCU 完成。

5.3.2.3 电压频率寄存器 UFREQ (0x118 修改)

电压频率寄存器 UFREQ

偏移地址：118H；字长：2 字节；默认值：0x2400

位	名称	功能描述	R/W	复位值
15:0	UFreq	电压频率寄存器	R	0

电压频率寄存器主要测量基波频率，测量带宽约 250Hz。频率值是一个 16 位的无符号数，参数格式化公式为：

$Freq = CLKIN / 4 / Reg(UFREQ)$ ，其中 $CLKIN = 1.8432MHz$ ， $Freq$ 单位/Hz。

电压频率更新周期由 $EMUCON3.FreqCnt$ 决定：

$FreqCnt = 0$ ，频率更新周期 32 周波；

$FreqCnt = 1$ ，频率更新周期 1 周波。

5.3.2.4 有功功率寄存器(0x11C~120/0x168~16C/0x174~178/0x20C~210 新增)

偏移地址	11CH	120H	168H	16CH	174H	178H
寄存器	PowerPA	PowerPB	PowerPA2	PowerPB2	SPL_PA	SPL_PB
默认值	0	0	0	0	0	0

偏移地址	20CH	210H
寄存器	HW_FPA	HW_FPB
默认值	0	0

平均有功功率：

平均有功功率 PowerPA、PowerPB，4 字节，更新速度为 1.7578125Hz；

平均有功功率 PowerPA2、PowerPB2，4 字节，更新速度为 14.0625Hz，稳定时间约 300ms；

修改全波有功功率：

配置 $HWP_CFG.P_D2F_SEL$ 选择全波有功功率计算源头：

=0，向下兼容，全波有功功率及电能计算源头为瞬时功率，功率寄存器 SPL_PA、SPL_PB，3 字节，更新速度 7.2KHz；

=1，全波有功功率及电能计算源头为半周期更新的功率，功率寄存器 SPL_PA、SPL_PB，4 字节，典型更新速度为 100Hz。

配置 $HWP_CFG.HW_P_MODE$ 选择半周波更新的功率的计算模式：

=0，2 路全波有功功率按照 $HWP_CFG.HW_P_NUM$ 配置的固定采样点数累加求平均计算；

=1，2 路全波有功功率采用过零驱动进行累加求平均计算。

新增基波有功功率 HW_FPA、HW_FPB，典型更新速度为 100Hz。

配置 $HWFP_CFG.FP_D2F_SEL$ 选择全波有功功率计算源头：

=0, 基波有功功率及电能计算源头为瞬时功率, HW_FPA、HW_FPB, 4 字节, 更新速度 7.2KHz;

=1, 基波有功功率及电能计算源头为半周期更新的功率, HW_FPA、HW_FPB, 4 字节, 典型更新速度为 100Hz。

配置 HWFP_CFG.HW_FP_MODE 选择半周波更新的功率的计算模式:

=0, 2 路基波有功功率按照 HWP_CFG.HW_FP_NUM 配置的固定采样点数累加求平均计算;

=1, 2 路基波有功功率采用过零驱动进行累加求平均计算。

有功功率寄存器均为二进制补码格式, 其中最高位是符号位, 只读。

5.3.2.5 无功功率寄存器(0x124/0x128/0x17C/0x180 修改)

偏移地址	124H	128H	17CH	180H
寄存器	PowerQA	PowerQB	SPL_QA	SPL_QB
默认值	0	0	0	0

在原有无功功率的基础上, 新增无功类型选择、无功计算模式选择等功能。

无功类型选择功能, 原 ADC 采样通道全波无功和新增基波无功共用一套电路与寄存器, 使用时只能二选一, 配置 HWQ_CFG.Q_SEL 选择无功计算源头:

=0, A、B 路无功为 ADC 采样通道全波无功。默认模式, 向下兼容, 无功计算通道输入为 7.2KHz 更新的 ADC 采样通道输出的波形 SPL_U、SPL_IA 和 SPL_IB。这种模式对于谐波有一定衰减, 但是衰减的不够彻底, 按照 IEC 标准测试 3 次谐波时, 误差大约为 0.8%。

=1, A、B 路无功为基波无功。无功计算通道输入为 7.2KHz 更新的基波采样通道输出的波形 SPL_FU、SPL_FIA 和 SPL_FIB。这种模式对于谐波衰减比较彻底, 按照 IEC 标准测试 3 次谐波时, 误差接近 0。

配置 HWQ_CFG.Q_D2F_SEL 选择全波有功功率计算源头:

=0, 向下兼容, 无功功率及电能计算源头为瞬时功率, 功率寄存器 SPL_QA、SPL_QB, 3 字节, 更新速度 7.2KHz;

=1, 无功功率及电能计算源头为半周期更新的功率, 功率寄存器 SPL_QA、SPL_QB, 4 字节, 典型更新速度为 100Hz。

配置 HWQ_CFG.HW_Q_MODE 选择半周波更新的无功功率的计算模式:

=0, 2 路无功功率按照 EMUCON9.HW_Q_NUM 配置的固定采样点数累加求平均计算;

=1, 2 路无功功率采用过零驱动进行累加求平均计算。

平均无功功率 PowerQA、PowerQB, 4 字节, 更新速度为 14.0625Hz。

无功功率寄存器均为二进制补码格式, 其中最高位是符号位, 只读。

5.3.2.6 视在功率寄存器(0x12C/0x130)

偏移地址	12CH	130H
寄存器	PowerSA	PowerSB
默认值	0	0

PowerSA 为 A 路视在功率, PowerSB 为 B 路视在功率, 更新速度为 14.0625Hz。稳定时间约为 300ms。

视在功率采用 RMS 法, 即: $S=U_{rms}*I_{rms}$

视在功率寄存器均为二进制补码格式, 32 位有符号数, 其中最高位是符号位, 视在功率总为正, 符号位保持为 0, 只读。

5.3.2.7 电能寄存器(0x134~144/0x170/0x238~254 新增)

偏移地址	134H	138H	13CH	140H	144H	170H
寄存器	EnergyPA	EnergyPB	EnergyQA	EnergyQB	EnergySA	EnergySB
默认值	0	0	0	0	0	0

偏移地址	238H	23CH	240H	244H	248H	24CH	250H	254H
寄存器	EnergyPA2	EnergyPB2	EnergyQA2	EnergyQB2	EnergyPA3	EnergyPB3	EnergyQA3	EnergyQB3
默认值	0	0	0	0	0	0	0	0

电能寄存器分为三类：有功能量、无功能量和视在能量。其中，有功和无功寄存器分别有 3 套，新增两套。每套能量单元均可独立配置代数和/正向/绝对值/反向（新增）4 种电能累加方式，有独立的 HFConst 控制，有独立的电能累加使能控制位，有独立的脉冲和中断及溢出中断输出。

原有的第 1 套有功、无功使用 HFConst，新增第 2 套有功、无功使用 HFConst2，新增第 3 套有功、无功使用 HFConst3。

配置 EMUCON3.PQMOD_ABINDEP，选择是否独立配置 A、B 路有功 1、无功 1 累加方式：

=0，不使能，EMUCON.PMOD/QMOD 控制 2 路有功 1/无功 1 累加方式；

=1，使能，EMUCON.PMOD/QMOD 控制 A 路有功 1/无功 1 累加方式，EMUCON3.PB_MOD/QB_MOD 控制 B 路有功 1/无功 1 累加方式。

有功 2、无功 2、有功 3、无功 3 电能单元累加模式及使能控制寄存器 EMUCON3，第 2、3 套均默认关闭，第 2 套默认正向累加方式，第 3 套默认反向累加方式。

电能寄存器类型可配置 EMUCON.Energy_clr 寄存器选择为读后清零或者不清零型，默认为读后清零。

电能参数是 24bit 无符号数，只读，代表对应脉冲的累加个数。寄存器最小单位代表的能量为 1/EC kWh，其中 EC 为电表常数。

能量寄存器在 0xFFFFF 溢出到 0x000000 时，会产生溢出标志 IF；若使能对应中断，则会产生相应中断。

5.3.2.8 功率因数寄存器(0x148/0x14C)

偏移地址	148H	14CH
寄存器	PfA	PfB
默认值	0	0

PfA 为 A 路功率因数，PfB 为 B 路功率因数。

功率因数寄存器为 24bit 二进制补码格式，高位是符号位，只读，归一化/2²³。

功率因数为 RMS 功率因数，即 $Pf=P/S$ 。

5.3.2.9 相角寄存器(0x150/0x154)

偏移地址	150H	154H
寄存器	ANGLEA	ANGLEB
默认值	0	0

ANGLEA 为基波电流通道 A 与基波电压夹角，ANGLEB 为基波电流通道 B 与基波电压夹角。

相角寄存器为 16bit 有符号数，只读。

计算公式：角度值=(RegValue/2¹⁵)*360 度

5.3.2.10 波形采样寄存器(0x15C~164/0x214~234 新增)

偏移地址	15CH	160H	164H	214H	218H	21CH
寄存器	SPL_IA	SPL_IB	SPL_U	SPL_IA2	SPL_IB2	SPL_U2
默认值	0	0	0	0	0	0

偏移地址	22CH	230H	234H
寄存器	SPL_FIA	SPL_FIB	SPL_FU
默认值	0	0	0

波形采样寄存器分以下几类：

SPL_IA、SPL_IB、SPL_U，高通滤波后数据，更新速度为 7.2KHz，受通道相位校正、DC_Offset 校正、ADC 通道增益校正影响；

SPL_IA2、SPL_IB2、SPL_U2，高通滤波前数据，更新速度为 14.4KHz，受通道相位校正、DC_Offset 校正 1、ADC 通道增益校正影响；

SPL_FIA、SPL_FIB、SPL_FU 为基波电流通道 A、电流通道 B 和电压通道波形采样值，更新速度为 7.2KHz，受通道相位校正、DC_Offset 校正、通道增益校正、基波通道增益校正影响。

波形采样寄存器均为二进制补码格式，24bit 有符号数，2 位符号位，归一化值为 SPL_xx/2²²。波形采样寄存器值 SPL_xx 与输入到芯片采样管脚正端 V_p 和负端 V_n 之间待测电压信号瞬时值 V_{pn} 的关系为：

$$SPL_xx = (V_{pn} * PGA / 1) * 2^{22}$$

SPL_xx：波形采样寄存器值；

V_{pn}：测量通道正端 V_p 和负端 V_n 之间待测电压信号瞬时值，单位为 V；要求输入信号瞬时值不能超过 1V；

PGA：测量通道内部 ADC 增益配置值；

假设外部输入峰值为 A=0.1v 的交流信号，那么波形采样寄存器的值为：

$$A * 2^{22} = 419430 \text{ (十进制)} = 24'H66666 \text{ (十六进制)}$$

波形采样寄存器的峰值的归一化值 = 对应全波有效值寄存器的归一化值 * sqrt(2) * 0.5。有效值寄存器的归一化值为 RMS/2²³，波形采样寄存器的归一化值为 WAV/2²²。则波形采样寄存器峰值 WAV_{peak}/2²² = (RMS/2²³) * sqrt(2) * 0.5。

波形采样寄存器更新时会置位标志寄存器 EMUIF.WAVUPIF 标志位 (0x190.0)，该位写 1 清零。若使能波形寄存器更新中断 EMUIE.WAVUPIE(0x18c.0)，则每次更新发生，会产生 EMU 中断。

5.3.3 状态和中断寄存器

5.3.3.1 计量状态寄存器 EMUStatus(0x158)

计量状态寄存器 EMUStatus

偏移地址：158H；字长：4 字节；默认值：0xE3EE78

位	位名称	功能描述	R/W	复位值
31	NoSAld	当用于计算 A 路电能的视在功率小于起动功率时，NoSAld 被置为 1； 当用于计算 A 路电能的视在功率大于/等于起动功率时 NoSAld 清为 0。	R	0
30	WREN	写使能标志：	R	0

		=1: 允许写入带写保护的寄存器; =0: 不允许写入带写保护的寄存器。		
29	CHNSEL	电流通道选择状态标识位: =0: 当前用于计算 A 路有功/无功电能的电流通道为通道 A; =1: 当前用于计算 A 路有功/无功电能的电流通道为通道 B。	R	0
28	NoQAld	当用于计算 A 路电能的无功功率小于起动功率时, NoQAld 被置为 1; 当用于计算 A 路电能的有功功率大于/等于起动功率时 NoQAld 清为 0。	R	0
27	NoPAld	当用于计算 A 路电能的有功功率小于起动功率时, NoPAld 被置为 1; 当用于计算 A 路电能的有功功率大于/等于起动功率时 NoPALd 清为 0。	R	0
26	REVQA	当前用于计算 A 路电能的反向无功功率指示标识信号: 当检测到负无功功率时, 该信号为 1; 当再次检测到正无功功率时, 该信号为 0。	R	0
25	REVPA	当前用于计算 A 路电能的反向有功功率指示标识信号: 当检测到负有功功率时, 该信号为 1; 当再次检测到正有功功率时, 该信号为 0。	R	0
24	Chksum1Busy	校表数据校验计算状态寄存器: =0: 表示校表数据校验和计算已完成, 校验值可用; =1: 表示校表数据校验和计算未完成, 校验值不可用。	R	0
23:0	Chksum1	校验和输出	R	E3EE78

此寄存器包括计量状态寄存器和校验和寄存器两部分。

EMUStatus [23:0]存放校表参数配置寄存器的 24 位校验和 1, CPU 可以检测这个寄存器来监控校表数据是否错乱。

一般应用场景:在校表全部完成后读取校验和, 存在 EEPROM 作为基准, 后面 CPU 定时读校验和寄存器做对比监控校表数据是否错乱。应用可自然规避进行自动 DC_Offset 校正完成后使能位自动清零而造成校验和寄存器改变的问题。

校验和的算法为三字节累加后取反。对于双字节寄存器, 将其扩展为三字节后累加, 扩展的字节为 00H。

根据默认值计算得到的校验和为 0xE3EE78。

以下三种情况下, 重新开始一次校验和计算: 系统复位、00H~6CH 某个寄存器发生写操作、EMUStatus 寄存器发生读操作。一次校验和计算需要 32 个 CPU 时钟。

此寄存器中, 当前用于 A 路电能计算的功率标志, 反应的是 A 路电能计算所用功率通道的状态, 受通道切换命令影响; 当用于 A 路电能计算通道为 A 通道时, 反应 A 路功率状态; 当用于 B 路电能计算通道为 B 通道时, 反应 B 路功率状态。

5.3.3.2 计量状态寄存器 2 EMUStatus2(0x188)

计量状态寄存器 2 EMUStatus2

偏移地址：188H；字长：4 字节；默认值：0

位	位名称	功能描述	R/W	复位值
31:15	--	只读，不可写。	R	0
14	SNegS_Flag	ADC 输入信号反向采样指示标志： =0，正向采样； =1，反向采样。	R	0
13	NoSbLd	当 B 路视在功率小于起动功率时，置 1； 当 B 路视在功率大于/等于起动功率时，清 0。	R	0
12	NoQbLd	当 B 路无功功率小于起动功率时，置 1； 当 B 路无功功率大于/等于起动功率时，清 0。	R	0
11	Nopbld	当 B 路有功功率小于起动功率时，置 1； 当 B 路有功功率大于/等于起动功率时，清 0。	R	0
10	REVQB	B 路无功功率反向指示信号： 当检测到负无功功率时，该信号为 1； 当再次检测到正有功功率时，该信号为 0。	R	0
9	REVPB	B 路有功功率反向指示信号： 当检测到负有功功率时，该信号为 1。 当再次检测到正有功功率时，该信号为 0。	R	0
8	Vref_flag	VREF 标志位： =1：VREF 正常； =0：VREF 低于跌落阈值。	R	0
7	Ldo_flag	计量 LDO 标志位： =1：计量 LDO 正常； =0：计量 LDO 低于跌落阈值。	R	0
6	NoSald	当用于计算 A 路电能的视在功率小于起动功率时，置 1； 当用于计算 A 路电能的视在功率大于/等于起动功率时，清 0。	R	0
5	WREN	写使能标志： =1：允许写入带写保护的寄存器； =0：不允许写入带写保护的寄存器。	R	0
4	CHNSEL	电流通道选择状态标识位： =1：当前用于计算 A 路有功/无功电能的电流通道为通道 B； =0：当前用于计算 A 路有功/无功电能的电流通道为通道 A。 默认状态下该位为 0，标识选择通道 A 用于电能计量。	R	0
3	NoQaLd	当用于 A 路电能计算的无功功率小于启动功率时，置 1； 当用于 A 路电能计算的有功功率大于/等于启动功率时，清 0。	R	0

2	NoPaLd	当用于 A 路电能计算的有功功率小于起动功率时, 置 1; 当用于 A 路电能计算的有功功率大于/等于起动功率时, 清 0。	R	0
1	REVQA	当前用于 A 路电能计算的反向无功功率指示标识信号: 当检测到负无功功率时, 该信号为 1; 当再次检测到正无功功率时, 该信号为 0。	R	0
0	REVPA	当前用于 A 路电能计算的反向有功功率指示标识信号: 当检测到负有功功率时, 该信号为 1; 当再次检测到正有功功率时, 该信号为 0。	R	0

此寄存器中, 当前用于 A 路电能计算的功率标志, 反应的是 A 路电能计算所用功率通道的状态, 受通道切换命令影响; 当用于 A 路电能计算通道为 A 通道时, 反应 A 路功率状态; 当用于 B 路电能计算通道为 B 通道时, 反应 B 路功率状态。

5.3.3.3 计量状态寄存器 3 EMUStatus3(0x1AC 新增)

计量状态寄存器 3 EMUStatus3

偏移地址: 1ACH; 字长: 3 字节; 默认值: FD5361

位	位名称	功能描述	R/W	复位值
31:25	--	保留	R	0
24	Chksum2Busy	校表数据校验计算状态寄存器: =0: 表示校表数据校验和计算已完成, 校验值可用; =1: 表示校表数据校验和计算未完成, 校验值不可用。	R	0
23:0	Chksum2	新增配置寄存器的校验和输出	R	FD5460

Chksum2 存放校表参数配置寄存器的 24 位校验和, CPU 可以检测这个寄存器来监控校表数据是否错乱。

校验范围基地址 0x50004000:

0x74~0xA4、0xAC~0xBC、0x304~0x310、0x318 (PQSRUN)、0x400~0x410 (DMA 通用波形缓存)、0x420~0x42C (RCD 专用波形缓存)。

FGain(0xA8)因涉及到基波自动增益校正功能, 寄存器值会自动改写, 因此不算入校验和范围。

ZXOTCFG(0x300)因包含过零标志位, 因此不算入校验和范围。

ROS_Tran_K(0x314)因需根据电压频率实时调整, 因此不算入校验和范围。

ECT 相关寄存器不计入校验和范围。

智能微断和 RCD 相关寄存器不计入校验和范围。

校验算法: 三字节累加后取反, 位数不足寄存器高位补 0。

5.3.3.4 中断使能寄存器 IE(0x18C 修改)

中断使能寄存器 IE

偏移地址: 18CH; 字长: 4 字节; 默认值: 0x0

当中断允许位配置为 1 且中断产生时, IRQ_N 引脚输出低电平。写保护寄存器, 配置该寄存器前需将写使能打开。

位	位名称	功能描述	R/W	复位值
31:24	--	只读，不可写。	R	0
23	IBZXIE	=0: 关闭 IB 电流通道过零中断; =1: 开启 IB 电流通道过零中断。	R/W	0
22	IAZXIE	=0: 关闭 IA 电流通道过零中断; =1: 开启 IA 电流通道过零中断。	R/W	0
21	UZXIF	=0: 关闭电压通道过零中断; =1: 开启电压通道过零中断。 当外部输入相同时, IBZX、IAZX、UZX 三个过零中断是同相位的。	R/W	0
20	SBEOIE	=0: 关闭视在电能寄存器 ESB 溢出中断; =1: 使能视在电能寄存器 ESB 溢出中断。	R/W	0
19	QBEOIE	=0: 关闭无功电能寄存器 EQB 溢出中断; =1: 使能无功电能寄存器 EQB 溢出中断。	R/W	0
18	PBEOIE	=0: 关闭有功电能寄存器 EPB 溢出中断; =1: 使能有功电能寄存器 EPB 溢出中断。	R/W	0
17	SFBIE	=0: 关闭 SFB 中断; =1: 打开 SFB 中断。	R/W	0
16	QFBIE	=0: 关闭 QFB 中断; =1: 打开 QFB 中断。	R/W	0
15	PFBIE	=0: 关闭 PFB 中断; =1: 打开 PFB 中断。	R/W	0
14	SFAIE	=0: 关闭 SFA 中断; =1: 打开 SFA 中断。	R/W	0
13	SEOIE	=0: 关闭视在电能寄存器溢出中断; =1: 使能视在电能寄存器溢出中断。	R/W	0
12	VREFIE	=0: 关闭 VREF 跌落中断; =1: 使能 VREF 跌落中断。	R/W	0
11	LDOIE	=0: 关闭 LDO 跌落中断; =1: 使能 LDO 跌落中断。	R/W	0
10	SPLIE	=0: 关闭 7.2KHz 波形更新中断; =1: 使能 7.2KHz 波形更新中断。	R/W	0
9	USAGIE	=0: 关闭 U 通道跌落中断; =1: 使能 U 通道跌落中断。	R/W	0
8	UpeakIE	=0: 关闭 U 通道过载中断; =1: 使能 U 通道过载中断。	R/W	0
7	IApeakIE	=0: 关闭 IA 通道过载中断; =1: 使能 IA 通道过载中断。	R/W	0
6	IBpeakIE	=0: 关闭 IB 通道过载中断; =1: 使能 IB 通道过载中断。	R/W	0
5	ZXIE	=0: 关闭过零中断; =1: 使能过零中断。 该过零信号与 UZX(bit21)过零都来自电压通道 ADC, 但是相位有延迟。	R/W	0

4	QEOIE	=0: 关闭无功电能寄存器溢出中断; =1: 使能无功电能寄存器溢出中断。	R/W	0
3	PEOIE	=0: 关闭有功电能寄存器溢出中断; =1: 使能有功电能寄存器溢出中断。	R/W	0
2	QFAIE	=0: 关闭QFA中断; =1: 打开QFA中断。	R/W	0
1	PFAIE	=0: 关闭PFA中断; =1: 打开PFA中断。	R/W	0
0	DUPDIE	=0: 关闭数据更新中断; =1: 使能数据更新中断。 数据 PowerPA、PowerPB、PowerQ、IARMS、IBRMS、URMS 寄存器发生数据更新时, 标志置位。	R/W	0

5.3.3.5 中断状态寄存器 IF(0x190 修改)

中断状态寄存器 IF

偏移地址: 190H; 字长: 4 字节; 默认值: 0x0

当某中断事件产生时, 硬件会将相应的中断标志置 1。

IF 中断标志的产生不受中断允许寄存器 IE 的控制, 只由中断事件是否发生决定。

位	位名称	功能描述	R/W	复位值
31:25	--	只读, 不可写。	R	0
24	DUPDIF2	=0: 新增计量数据更新事件未发生; =1: 新增计量数据更新事件发生。	R/W	0
23	IBZXIF	=0: 电流通道 B 过零事件未发生; =1: 电流通道 B 过零事件发生。	R/W	0
22	IAZXIF	=0: 电流通道 A 过零事件未发生; =1: 电流通道 A 过零事件发生。	R/W	0
21	UZXIF	=0: 电压通道过零事件未发生; =1: 电压通道过零事件发生。	R/W	0
20	SBEOIF	=0: 视在电能寄存器 ESB 溢出事件未发生 =1: 视在电能寄存器 ESB 溢出事件发生。	R/W	0
19	QBEOIF	=0: 无功电能寄存器 EQB 溢出事件未发生; =1: 无功电能寄存器 EQB 溢出事件发生。	R/W	0
18	PBEOIF	=0: 有功电能寄存器 EPB 溢出事件未发生; =1: 有功电能寄存器 EPB 溢出事件发生。	R/W	0
17	SFBIF	=0: 未发生 SFB 脉冲输出事件; =1: 发生 SFB 脉冲输出事件。	R/W	0
16	QFBIF	=0: 未发生 QFB 脉冲输出事件; =1: 发生 QFB 脉冲输出事件。	R/W	0
15	PFBIF	=0: 未发生 PFB 脉冲输出事件; =1: 发生 PFB 脉冲输出事件。	R/W	0
14	SFAIF	=0: 未发生 SFA 脉冲输出事件; =1: 发生 SFA 脉冲输出事件。	R/W	0

13	SEAOIF	=0: 未发生视在电能寄存器 ESA 溢出事件; =1: 发生视在电能寄存器 ESA 溢出事件。	R/W	0
12	VREFIF	=0: VREF 跌落事件未发生; =1: VREF 跌落事件发生。	R/W	0
11	LDOIF	=0: LDO 跌落事件未发生; =1: LDO 跌落事件发生。	R/W	0
10	SPLIF	=0: 7.2KHz 波形更新事件未发生; =1: 7.2KHz 波形更新事件发生。	R/W	0
9	USAGIF	=0: U 通道跌落事件未发生; =1: U 通道跌落事件发生。	R/W	0
8	UpeakIF	=0: U 通道过载事件未发生; =1: U 通道过载事件发生。	R/W	0
7	IApeakIF	=0: IA 通道过载事件未发生; =1: IA 通道过载事件发生。	R/W	0
6	IBpeakIF	=0: IB 通道过载事件未发生; =1: IB 通道过载事件发生。	R/W	0
5	ZXIF	=0: 过零事件未发生; =1: 过零事件发生。	R/W	0
4	QEOIF	=0: 无功电能寄存器溢出事件未发生; =1: 无功电能寄存器溢出事件发生。	R/W	0
3	PEOIF	=0: 有功电能寄存器溢出事件未发生; =1: 有功电能寄存器溢出事件发生。	R/W	0
2	QFAIF	=0: QFA 脉冲输出事件未发生; =1: QFA 脉冲输出事件发生。	R/W	0
1	PFAIF	=0: PFA 脉冲输出事件未发生; =1: PFA 脉冲输出事件发生。	R/W	0
0	DUPDIF	=0: 数据更新事件未发生; =1: 数据更新事件发生。	R/W	0

5.3.3.6 中断使能寄存器 EMUIE2(0x1A0 新增)

中断使能寄存器 EMUIE2

偏移地址: 1A0H; 字长: 3 字节; 默认值: 0x0

中断号 2。

位	位名称	功能描述	R/W	复位值
31:16	--	只读, 不可写。	R	0
15	QB3EOIE	=0: 关闭无功电能寄存器 EQB3 溢出中断; =1: 使能无功电能寄存器 EQB3 溢出中断。	R/W	0
14	PB3EOIE	=0: 关闭有功电能寄存器 EPB3 溢出中断; =1: 使能有功电能寄存器 EPB3 溢出中断。	R/W	0
13	QB2EOIE	=0: 关闭无功电能寄存器 EQB2 溢出中断; =1: 使能无功电能寄存器 EQB2 溢出中断。	R/W	0
12	PB2EOIE	=0: 关闭有功电能寄存器 EPB2 溢出中断; =1: 使能有功电能寄存器 EPB2 溢出中断。	R/W	0

11	QA3EOIE	=0: 关闭无功电能寄存器 EQA3 溢出中断; =1: 使能无功电能寄存器 EQA3 溢出中断。	R/W	0
10	PA3EOIE	=0: 关闭有功电能寄存器 EPA3 溢出中断; =1: 使能有功电能寄存器 EPA3 溢出中断。	R/W	0
9	QA2EOIE	=0: 关闭无功电能寄存器 EQA2 溢出中断; =1: 使能无功电能寄存器 EQA2 溢出中断。	R/W	0
8	PA2EOIE	=0: 关闭有功电能寄存器 EPA2 溢出中断; =1: 使能有功电能寄存器 EPA2 溢出中断。	R/W	0
7	QFB3IE	=0: 关闭QFB3中断; =1: 打开QFB3中断。	R/W	0
6	PFB3IE	=0: 关闭PFB3中断; =1: 打开PFB3中断。	R/W	0
5	QFB2IE	=0: 关闭QFB2中断; =1: 打开QFB2中断。	R/W	0
4	PFB2IE	=0: 关闭PFB2中断; =1: 打开PFB2中断。	R/W	0
3	QFA3IE	=0: 关闭QFA3中断; =1: 打开QFA3中断。	R/W	0
2	PFA3IE	=0: 关闭PFA3中断; =1: 打开PFA3中断。	R/W	0
1	QFA2IE	=0: 关闭QFA2中断; =1: 打开QFA2中断。	R/W	0
0	PFA2IE	=0: 关闭PFA2中断; =1: 打开PFA2中断。	R/W	0

写保护寄存器，配置该寄存器前需将写使能打开。中断号 2。

5.3.3.7 中断状态寄存器 EMUIF2(0x1A4 新增)

中断状态寄存器 EMUIF2

偏移地址: 1A4H; 字长: 3 字节; 默认值: 0x0

位	位名称	功能描述	R/W	复位值
31:16	--	只读，不可写。	R	0
15	QB3EOIF	=0: 无功电能寄存器 EQB3 溢出事件未发生; =1: 无功电能寄存器 EQB3 溢出事件发生。	R/W	0
14	PB3EOIF	=0: 有功电能寄存器 EPB3 溢出事件未发生; =1: 有功电能寄存器 EPB3 溢出事件发生。	R/W	0
13	QB2EOIF	=0: 无功电能寄存器 EQB2 溢出事件未发生; =1: 无功电能寄存器 EQB2 溢出事件发生。	R/W	0
12	PB2EOIF	=0: 有功电能寄存器 EPB2 溢出事件未发生; =1: 有功电能寄存器 EPB2 溢出事件发生。	R/W	0
11	QB3EOIF	=0: 无功电能寄存器 EQB3 溢出事件未发生; =1: 无功电能寄存器 EQB3 溢出事件发生。	R/W	0
10	PA3EOIF	=0: 有功电能寄存器 EPA3 溢出事件未发生; =1: 有功电能寄存器 EPA3 溢出事件发生。	R/W	0

9	QA2EOIF	=0: 无功电能寄存器 EQA2 溢出事件未发生; =1: 无功电能寄存器 EQA2 溢出事件发生。	R/W	0
8	PA2EOIF	=0: 有功电能寄存器 EPA2 溢出事件未发生; =1: 有功电能寄存器 EPA2 溢出事件发生。	R/W	0
7	QFB3IF	=0: 未发生 QFB3 脉冲输出事件; =1: 发生 QFB3 脉冲输出事件。	R/W	0
6	PFB3IF	=0: 未发生 PFB3 脉冲输出事件; =1: 发生 PFB3 脉冲输出事件。	R/W	0
5	QFB2IF	=0: 未发生 QFB2 脉冲输出事件; =1: 发生 QFB2 脉冲输出事件。	R/W	0
4	PFB2IF	=0: 未发生 PFB2 脉冲输出事件; =1: 发生 PFB2 脉冲输出事件。	R/W	0
3	QFA3IF	=0: 未发生 QFA3 脉冲输出事件; =1: 发生 QFA3 脉冲输出事件。	R/W	0
2	PFA3IF	=0: 未发生 PFA3 脉冲输出事件; =1: 发生 PFA3 脉冲输出事件。	R/W	0
1	QFA2IF	=0: 未发生 QFA2 脉冲输出事件; =1: 发生 QFA2 脉冲输出事件。	R/W	0
0	PFA2IF	=0: 未发生 PFA2 脉冲输出事件; =1: 发生 PFA2 脉冲输出事件。	R/W	0

5.3.3.8 中断使能寄存器 EMUIE3(0x1B0 新增)

中断使能寄存器 EMUIE3

偏移地址: 1B0H; 字长: 3 字节; 默认值: 0x0

中断号 0。

位	位名称	功能描述	R/W	复位值
31:9	--	只读, 不可写。	R	0
8	WAVEHW_IE	=0: 关闭同步采样通道半波参数更新中断; =1: 使能同步采样通道半波参数更新中断。	R/W	0
7	DMA_RCD_BUF_ErrIE	=0: 关闭波形缓存 DMA RCD 通道数据错误中断; =1: 开启波形缓存 DMA RCD 通道数据错误中断。	R/W	0
6	DMA_RCD_BUFHF_IE	=0: 关闭波形缓存 DMA RCD 通道 BUFFER 半满中断; =1: 开启波形缓存 DMA RCD 通道 BUFFER 半满中断。	R/W	0
5	DMA_RCD_BUFF_IE	=0: 关闭波形缓存 DMA RCD 通道 BUFFER 满中断; =1: 开启波形缓存 DMA RCD 通道 BUFFER 满中断。	R/W	0
4	DMA_BUFIB_ErrIE	=0: 关闭波形缓存 DMA IB 通道数据错误中断; =1: 开启波形缓存 DMA IB 通道数据错误中断。	R/W	0

		=1: 开启波形缓存 DMA IB 通道数据错误中断。		
3	DMA_BUFIA_ErrIE	=0: 关闭波形缓存 DMA IA 通道数据错误中断; =1: 开启波形缓存 DMA IA 通道数据错误中断。	R/W	0
2	DMA_BUFUErr_IE	=0: 关闭波形缓存 DMA U 通道数据错误中断; =1: 开启波形缓存 DMA U 通道数据错误中断。	R/W	0
1	DMA_BUFHF_IE	=0: 关闭波形缓存 DMA 通道 BUFFER 半满中断; =1: 开启波形缓存 DMA 通道 BUFFER 半满中断。	R/W	0
0	DMA_BUFF_IE	=0: 关闭波形缓存 DMA 通道 BUFFER 满中断; =1: 开启波形缓存 DMA 通道 BUFFER 满中断。	R/W	0

5.3.3.9 中断状态寄存器 EMUIF3(0x1B4 新增)

位	位名称	功能描述	R/W	复位值
31:9	--	只读, 不可写。	R	0
8	WAVEHW_IF	=0: 同步采样通道半波参数更新事件未发生; =1: 同步采样通道半波参数更新事件发生。	R/W	0
7	DMA_RCD_BUF_ErrIF	=0: 波形缓存 DMA RCD 通道数据错误事件未发生; =1: 波形缓存 DMA RCD 通道数据错误事件发生。	R/W	0
6	DMA_RCD_BUFHF_IF	=0: 波形缓存 DMA RCD 通道 BUFFER 半满事件未发生; =1: 波形缓存 DMA RCD 通道 BUFFER 半满事件发生。	R/W	0
5	DMA_RCD_BUFF_IF	=0: 波形缓存 DMA RCD 通道 BUFFER 满事件未发生; =1: 波形缓存 DMA RCD 通道 BUFFER 满事件发生。	R/W	0
4	DMA_BUFIB_ErrIF	=0: 波形缓存 DMA IB 通道数据错误事件未发生; =1: 波形缓存 DMA IB 通道数据错误事件发生。	R/W	0
3	DMA_BUFIA_ErrIF	=0: 波形缓存 DMA IA 通道数据错误事件未发生; =1: 波形缓存 DMA IA 通道数据错误事件发生。	R/W	0

		生。		
2	DMA_BUFUErr_IF	=0: 波形缓存 DMA U 通道数据错误事件未发生; =1: 波形缓存 DMA U 通道数据错误事件发生。	R/W	0
1	DMA_BUFHF_IF	=0: 波形缓存 DMA 通道 BUFFER 半满事件未发生; =1: 波形缓存 DMA 通道 BUFFER 半满事件发生。	R/W	0
0	DMA_BUFF_IF	=0: 波形缓存 DMA 通道 BUFFER 满事件未发生; =1: 波形缓存 DMA 通道 BUFFER 满事件发生。	R/W	0

5.3.4 同步采样配置和参数寄存器（新增）

基地址：0x40040080

5.3.4.1 WAVE_WKEY(0x00 新增)

同步采样写密码寄存器

偏移地址：00H；默认值：0x0

位	位名称	功能描述	R/W	复位值
7:0	WKEY	写 0xE5，使能同步采样写操作； 写 0xDC，关闭同步采样写操作。	WO	0

5.3.4.2 WAVECFG (0x04 新增)

同步采样通道配置寄存器

偏移地址：04H；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:4	--	只读，不可写。	R	0
3:2	HW_WAVE	半波有效值和半波功率点数配置寄存器： =00，128 点累加求平均； =01，64 点累加求平均； =10，32 点累加求平均； =11，256 点累加求平均； 注：该累加点数与同步采样点数需要独立设置。 注：参数更新中断允许位 ENUIE3.bit8，中断标志位 EMUIF3.bit8。	R/W	0
1	WAVE_AGC_OFF 你	通道增益自动调整关闭控制位： =0，开启 AGC； =1，关闭 AGC。	R/W	0

		备注：增益自动调整功能，是为了解决 DEC 滤波器抽取倍数变化引起的增益变化。开启功能，NVM 通道有效值、同步采样通道得到的波形幅值与计量通道一致。		
0	MODE_SEL	模式选择控制寄存器： =0，NVM 模式； =1，同步采样模式。	R/W	0

5.3.4.3 WAVECFG2 (0x08 新增)

同步采样通道配置寄存器 2

偏移地址：08H；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:10	--	只读，不可写。	R	0
9:8	WAVE_DIV	输出给到波形缓存的分频： =00，不分频； =01,2 分频； =10,4 分频； =11,8 分频。	R	0
7:5	WAVE_SR	同步采样一周波点数配置寄存器： =000，256 点同步采样； =001，128 点同步采样； =010，64 点同步采样； =011，512 点同步采样； =100，10 周波 1024 点同步采样； =101，10 周波 2048 点同步采样； =其他，保留。	R/W	0
4	Bypass_WAVE_DCOS_EN	Bypass 同步采样通道 DC Offset 校正使能位： =0，不使能，计量通道 DC Offset 校正对同步采样通道有效； =1，使能，计量通道 DC Offset 校正对同步采样通道无效。	R/W	0
3	WAVE_AAC_OFF	谐波衰减补偿关闭控制位： =0，开启 AAC； =1，关闭 AAC。 备注：开启该功能后，读取同步采样通道缓存波形做 FFT 时就不再需要做谐波系数的补偿。	R/W	0
2:0	WAVE_HPFON	同步采样通道高通滤波器使能位： =0，关闭高通； =1，使能高通。 具体定义如下：	R/W	0

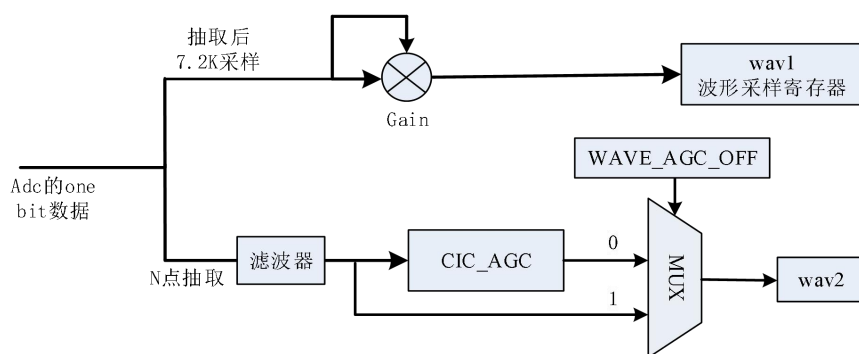
		bit2: WAVE_HP FON_U bit1: WAVE_HP FON_IB bit0: WAVE_HP FON_IA		
--	--	---	--	--

注:

7.2K采样率下的采样瞬时值, 为wav1; 经过N点同步采样后, 波形采样瞬时值为wav2, 其中N为同步采样点数, N的取值为64/128/256/512/自定义;

默认情况下, 即WAVECFG(0x04)中的WAVE_AGC_OFF=0时, 硬件会做自动的增益补偿, 同步采样通道得到的波形数据幅度与7.2KHz计量通道保持一致, 即wav1 = wav2。

当配置WAVECFG(0x04)中的WAVE_AGC_OFF=1后, AGC自动增益不开启, 二者关系为: $2 * wav2 = wav1 * (7200/Rt)^3$, 其中Rt为采样率, $Rt = N * Freq$, N为采样点数, Freq为电网频率。



5.3.4.4 WAVE_EN(0x0C 新增)

同步采样使能寄存器

偏移地址: 0CH; 默认值: 0x0

位	位名称	功能描述	R/W	复位值
31:1	--	只读, 不可写。	R	0
0	WAVE_EN	同步采样模块使能寄存器: =0, 不使能; =1, 使能。	R/W	0

5.3.4.5 WAVECNT (0x10 新增)

采样率控制寄存器

偏移地址: 10H; 默认值: 0x0

位	位名称	功能描述	R/W	复位值
31:10	--	只读, 不可写。	R	0
9:0	WAVECNT	用于改变同步采样缓存波形的采样率: 若 WAVECNT[9:0]=0, 则 WAVE_SR 配置的采样率有效; 若 WAVECNT[9:0]≠0, 则 WAVE_SR 配置的采样率无效, 若期望的周波采样点数为	R/W	0

		<p>N，按如下公式配置 WAVECNT。</p> <p>公式为：</p> $\text{WAVECNT} = (1.8432 \times 10^6) / \text{freq} / N - 1;$ <p>(freq 为电网频率，一般为 50Hz 或 60Hz)</p> <p>此时支持的最小写入值为 32。当写入的值小于 32 时，预期值可写入，但是实际起作用的一直是 0x8F。</p> <p>WAVECNT 读为当前实际使用的抽取倍数。</p>		
--	--	--	--	--

5.3.4.6 WAVE_DC_EN (0x14 新增)

AUTO DC 使能寄存器

偏移地址：14H；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:3	--	只读，不可写。	R	0
2	waveu_dc_en	waveu_dc_en、waveib_dc_en、waveia_dc_en 写 1 使能同步采样通道直流 Offset 测量，测量完成后自动清零。	R/W	0
1	waveib_dc_en		R/W	0
0	waveia_dc_en		R/W	0

5.3.4.7 WAVE_Phs(0x18~20 新增)

偏移地址	18H	1CH	20H
寄存器	WAVE_PhsIA	WAVE_PhsIB	WAVE_PhsU
默认值	0	0	0

同步采样通道相位校正寄存器，R/W，7bit 有符号数，寄存器的 WAVE_Phs[6:0]有效，最高位为符号位，校正源头为 1bit 数据，50Hz 时 1 LSB 代表 0.009766° 相位校正，校正范围为 1.25°。

5.3.4.8 WAVE_Gain(0x24~2C 新增)

偏移地址	1CH	20H	24H
寄存器	WAVE_IAGain	WAVE_IBGain	WAVE_UGain
默认值	0	0	0

同步采样通道增益校正寄存器，16bit 有符号数，最高位为符号位，R/W。

5.3.4.9 WAVE_HW_RMS(0x50~58 新增)

偏移地址	50H	54H	58H
寄存器	WAVE_HW_RMSIA	WAVE_HW_RMSIB	WAVE_HW_RMSU
默认值	0	0	0

同步采样通道 3 路有效值按照 WAVECFG.HW_NUM 配置的固定采样点数取绝对值累加求平均计算，典型更新速度半周波。

有效值寄存器为 24 位有符号数，最高位为 0 表示有效数据，R，归一化值=RegValue/2²³。

5.3.4.10 WAVE_HW_P(0x5C~60 新增)

偏移地址	5CH	60H
寄存器	WAVE_HW_PA	WAVE_HW_PB
默认值	0	0

同步采样通道 2 路功率按照 WAVECFG.HW_NUM 配置的固定采样点数累加求平均并开方计算，典型更新速度半周波。

功率寄存器为 32 位有符号数，R，归一化值=RegValue/2³¹。

5.3.5 DMA 缓存配置和参数寄存器(新增)

5.3.5.1 DMA_WAVE_CFG(0x400 新增)

DMA 通用波形缓存配置寄存器

偏移地址：400H；字长：2 字节；默认值：0x70

位	位名称	功能描述	R/W	复位值
31:15	--	只读，不可写。	R	0
14	EXTRACT	间隔缓存使能位：=0，不使能；=1，使能。 两个点缓存一个点，即把波形数据源头采样率降半后再缓存数据。	R/W	0
13:10	DMA_BANK_CNT	周波区块个数，数据存储会按照循环存储。 比如配置为 2，则有 3 个区块，数据会先存放 A 区，A 区存满存放 B 区，然后存放 C 区，然后又存放 A 区。寄存器写入 n，表示有 n+1 个周波区块，可配置实际区块范围 1~16。	R/W	0
9	CKM_EN	校验和使能位： =0，不计算校验和； =1，波形缓存过程，模式 0 按通道计算校验和，模式 1 计算每个周波区块的校验和，校验和结果存放寄存器。	R/W	0
8	WMCFG	DMA 单次模式、连续模式选择： =0，单次模式，写操作 Buf 满之后则不再发起 DMA 操作； =1，连续模式，写操作 Buf 满，折返到起始地址重新开始。	R/W	0
7	DMAMODE	DMA 模式选择： =0，模式 0，不同通道数据按通道存放； =1，模式 1，不同通道数据按点交叉存放（多个周波区块）。	R/W	0
6	IB_SEL	IB 通道 DMA 波形缓存使能位： =0，不使能；=1，使能。	R/W	1
5	IA_SEL	IA 通道 DMA 波形缓存使能位：	R/W	1

		=0, 不使能; =1, 使能。		
4	U_SEL	U 通道 DMA 波形缓存使能位: =0, 不使能; =1, 使能。	R/W	1
3:2	--	只读, 不可写。	R	0
1:0	DS_CFG	DMA 波形数据源头选择寄存器: =00, 计量采样通道高通滤波器后 7.2KHz 更新的波形数据; =01, 计量采样通道高通滤波器前 14.4KHz 更新的波形数据; =10, 选择经过基波增益校正后 7.2KHz 更新的基波数据; =11, 选择同步采样波形。	R/W	00

5.3.5.2 DMA_BUF_CTRL(0x404 新增)

DMA 通用波形缓存使能寄存器

偏移地址: 404H; 字长: 1 字节; 默认值: 0x0

位	位名称	功能描述	R/W	复位值
31:1	Reserved	保留。	R	0
0	BUF_EN	数据缓存启动位: =0, 关闭; =1, 启动。 注意: 使能必须放在配置的最后一步, 且仅在 BUF_EN=0, 数据缓存关闭时, 才可修改其他配置寄存器。	R/W	0

5.3.5.3 DMA_BUF_BADDR(0x408 新增)

DMA 通用数据缓存的目标地址的偏移地址寄存器

偏移地址: 408H; 字长: 2 字节; 默认值: 0x0

位	位名称	功能描述	R/W	复位值
31:15	--	只读, 不可写。	R	0
14:0	DMA_BUF_BASE_ADDR	数据缓存的目标地址的偏移地址。 数据缓存目标地址=该寄存器值*4 + ram 基址, 若该寄存器值设置为 0xC00, ram 基址为 0x10000000 则数据缓存目标地址 = 0xC00*4 + 0x10000000 = 0x10003000。	R/W	0x0

5.3.5.4 DMA_BUF_DEPTH(0x40C 新增)

DMA 通用数据缓存深度寄存器

偏移地址: 0xBC; 字长: 2 字节; 默认值: 0x8F

位	位名称	功能描述	R/W	复位值
31:15	--	只读, 不可写。	R	0

14:0	DMA_BUF_DEPTH	支持任意深度配置。 模式 0：该寄存器定义为单通道的深度，深度为(N+1)(Word)； 波形缓存的总深度由通道深度和通道间隙深度计算而来。 模式 1：该寄存器定义为一个周波区块的深度，深度为(N+1)(Word)。	R/W	8F
------	---------------	---	-----	----

5.3.5.5 DMA_GAP_CFG(0x410 新增)

DMA 通用波形缓存通道间隙配置寄存器

偏移地址：0x410；字长：3 字节；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:18	Reserved	保留	R	0
17	IB_GAP_EN	第 2 个通道后面是否插入间隙， 0=不插入，1=插入间隙，只在模式 0 有效	R/W	0
16	IA_GAP_EN	第 1 个通道后面是否插入间隙， 0=不插入，1=插入间隙，只在模式 0 有效	R/W	0
15	U_GAP_EN	第 0 个通道后面是否插入间隙， 0=不插入，1=插入间隙，只在模式 0 有效	R/W	0
14:0	DMA_GAP_DEPTH	模式 0： 数据缓存两个通道间隙深度配置，深度为(N)(Word)；支持任意深度配置。 当配置为 0，则所有通道后面不插入间隙空间。 注意：所有通道有效数据区域后面插入的间隙大小相同。 模式 1： 数据缓存两个周波区块间隙深度配置，深度为(N)(Word)；支持任意深度配置。 当配置为 0，则所有周波区块后面不插入间隙空间。	R/W	0

5.3.5.6 DMA_WAVE_ADDR(0x414 新增)

DMA 通用波形缓存当前 DMA 指针地址寄存器

偏移地址：0x414；字长：2 字节；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:14	--	只读，不可写。	R	0
13:0	DMA_BUFF_ADDR	软件读该寄存器，获取当前 DMA 指针地址	R	0

5.3.5.7 DMA_ERR_ADDR(0x418 新增)

DMA 通用波形缓存数据错误发生地址寄存器 DMAWAVE_ERR_ADDR

偏移地址：0x418；字长：2 字节；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:14	--	只读，不可写。	R	0
13:0	DMAWAVE_ERR_ADDR	在 ADC 采样间隔内，若有任何一个通道 DMA 操作没有响应，则记录数据错误发生地址并发出 DMA 错误中断；记录数据错误发生地址放置在 DMA_WAVE_ERR_ADDR 寄存器中。	R	0

5.3.5.8 DMA_CHECKSUM(0x41C 新增)

DMA 波形校验和寄存器

偏移地址：0x41C；字长：3 字节；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:24	CKM_RCD	专用波形缓存数据校验和。	R	0
23:16	CKM_IB	模式 0：IB 通道校验和 模式 1：上一个周波区块 IB 通道的校验和	R	0
15:8	CKM_IA	模式 0：IA 通道校验和 模式 1：上一个周波区块 IA 通道的校验和	R	0
7:0	CKM_U	模式 0：U 通道校验和 模式 1：上一个周波区块 U 通道的校验和	R	0

5.3.5.9 DMA_RCD_CFG(0x420)(新增)

DMA 专用波形缓存配置寄存器

偏移地址：0x420；字长：3 字节；默认值：0x0

位	名称	功能描述	R/W	复位值
31:14	--	只读，不可写。	R	0
13	RCD_EXTRACT	专用波形缓存二抽取使能	R /W	0
11:2	DMA_BANK_RCD_CNT	SPL_IB DMA 数据缓存 Bank 区块数量。可配置范围 1~1024。	R/W	0
1	RCD_CKM_EN	专用波形缓存校验和使能位： = 0，不计算校验和； = 1，波形缓存过程，模式 0 按通道计算校验和，模式 1 计算每个波形区块各个通道的校验和，校验和结果存放寄存器。	R/W	0
0	RCD_WMCFG	DMA 单次模式、连续模式选择 =0，单次模式，写操作 Buf 满之后则不再发起 DMA 操作； =1，连续模式，写操作 Buf 满，折返到起始地址重新开始。	R/W	0

5.3.5.10 DMA_RCD_CTRL(0x424) (新增)

DMA 专用波形缓存使能寄存器

偏移地址：424H；字长：3 字节；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:1	--	只读，不可写。	R	0
0	BUF_RCD_EN	数据缓存启动位： =0，关闭；=1，启动。 注意：使能必须放在配置的最后一步，且仅在 BUF_RCD_EN=0，数据缓存关闭时，才可修改其他配置寄存器。	R/W	0

5.3.5.11 DMA_RCD_BADDR(0x428)(新增)

DMA 专用波形缓存的目标地址的偏移地址寄存器

偏移地址：428H；字长：3 字节；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:15	--	只读，不可写。	R	0
14:0	DMA_BUF_RCD_BADDR	数据缓存的目标地址的偏移地址。 数据缓存目标地址=该寄存器值*4 + ram 基址，若该寄存器值设置为 0xC00，ram 基址为 0x10000000 则数据缓存目标地址 = 0xC00*4 + 0x10000000 = 0x10003000。	R/W	0x0

5.3.5.12 DMA_RCD_DEPTH(0x42C)(新增)

DMA 专用波形缓存深度寄存器

偏移地址：42CH；字长：3 字节；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:9	--	只读，不可写。	R	0
8:0	DMA_BUF_RCD_DEPTH	SPL_IB DMA 数据缓存 Bank 区块的大小 以 word 为单位，每个 word 包含两点 16bit 波形数据。	R/W	0

5.3.5.13 DMA_RCD_DEPTH(0x430)(新增)

DMA 专用波形缓存配置寄存器

偏移地址：0x430；字长：2 字节；默认值：0x0

位	名称	功能描述	R/W	复位值
31:15	--	只读，不可写。	R	0
14:0	DMA_RCD_GAP_DEPTH	数据缓存两个周波区块间隙深度配置，深度为(N)(Word)；支持任意深度配置。 当配置为 0，则所有周波区块后面不	R/W	0

		插入间隙空间。		
--	--	---------	--	--

5.3.5.14 DMA_RCD_ERR_ADDR (0x434) (新增)

DMA 专用波形缓存 DMA 错误地址寄存器

偏移地址：434H；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:15	--	只读，不可写。	R	0
14:0	DMA_BUF_RCD_ERR_ADDR	在 ADC 采样间隔内，若有任何一个通道 DMA 操作没有响应，则记录数据错误发生地址并发出 DMA 错误中断；记录数据错误发生地址放置在该寄存器中。	R	0

5.3.6 智能微断 RCD 配置和参数寄存器

5.3.6.1 RCD_CTRL(0x480 新增)

RCD 控制寄 RCD_CTRL

位	位名称	功能描述	R/W	复位值
31:20	--	保留	R	0
19:16	RCDDec_A	剩余电流检测方案 A 积分器递减量，有符号数	R/W	0xF
15:12	--	保留	R	0
11:8	RCDInc_A	剩余电流检测方案 A 积分器递增量，有符号数	R/W	0x2
7	--	保留	R	0
6:4	RCDTapSel_B	剩余电流方案 B 积分器抽头选择 '000': 216 '001': 198 '010': 180 '011': 144 '100': 126 '101': 108 '110': 72 '111': 36	R/W	0x3
3	--	保留	R	0
2:1	RCDModeCntl	剩余电流检测功能模式选择配置： '00': 方案 A '01': 方案 B '10': 方案 A&B '11': 保留	R/W	0x2

0	RCDTrigLatchEn	RCD Trig 信号输出锁存使能 ‘1’: RCD Trig_ 输出一旦从低电平切换到高电平生效后, 则始终保持在高电平直到模块复位或重新使能启动工作为止。 ‘0’: 不锁存	R/W	1
---	----------------	--	-----	---

5.3.6.2 RCD_EN(0x484 新增)

RCD 方案 A 输入信号阈值寄存器

位	位名称	功能描述	R/W	复位值
31:16	--	保留	R	0
0	RCD_En	剩余电流模块工作使能配置: ‘1’: 模块使能, 除配置寄存器外, 其他 RCD 数字电路及状态寄存器均先复位, 再启动工作。 ‘0’: 模块关闭 模块使能, 应用时, 先写 0 再写 1, 保证使能正确。	R/W	0

5.3.6.3 RCD_THRE(0x488 新增)

RCD 方案 B 输入信号阈值寄存器

位	位名称	功能描述	R/W	复位值
31:16	RCDIsThre_B	输入信号比较门限 无符号数	R/W	0x6CA
15:0	RCDIsThre_A	输入信号比较门限 无符号数	R/W	0x24F

5.3.6.4 RCD_ATTHRE(0x48C 新增)

RCD 方案 A 脱扣阈值寄存器

位	位名称	功能描述	R/W	复位值
31:16	RCDCntThre_A	积分结果上限, 有符号数	R/W	0xC8
15:0	RCDTrigThre_A	积分结果脱扣判决门限, 有符号数	R/W	0xC8

5.3.6.5 RCD_BTTHRE(0x490 新增)

RCD 方案 B 脱扣阈值寄存器

位	位名称	功能描述	R/W	复位值
31:8	--	保留	R	0
7:0	RCDTrigThre_B	积分结果脱扣判决门限, 无符号数	R/W	0x4C

5.3.6.6 RCD_ACNT(0x494 新增)

RCD 方案 A 积分器输出结果寄存器

位	位名称	功能描述	R/W	复位值
31:16	--	保留	R	0
15:0	RCDCnt_A	积分结果，有符号数	R	0

5.3.6.7 RCD_BCNT(0x498 新增)

RCD 方案 B 积分器输出结果寄存器

位	位名称	功能描述	R/W	复位值
31:8	--	保留	R	0
7:0	RCDCnt_B	积分结果，无符号数。	R	0

5.3.6.8 RCD_IE (0x49C 新增)

RCD 中断使能寄存器

位	位名称	功能描述	R/W	复位值
31:3	--	保留	R	0
2	TrigSigHw_IE	专用硬件触发脱扣信号发生结束中断使能，高电平有效。	R/W	0
1	TrigSig_IE	通用脱扣信号发生结束中断使能，高电平有效。	R/W	0
0	RCDTrig_IE	RCD Trig 中断使能，高电平有效。	R/W	0

5.3.6.9 RCD_IF(0x4A0 新增)

RCD 中断标志寄存器

位	位名称	功能描述	R/W	复位值
31:3	--	保留	R	0
2	TrigSigHw_IF	专用硬件触发脱扣信号发生结束中断信号，高电平有效，写 1 清 0	R/W	0
1	TrigSig_IF	通用脱扣信号发生结束中断信号，高电平有效，写 1 清 0	R/W	0
0	RCDTrig_IF	RCD Trig 端口输出触发中断信号，高电平有效，写 1 清 0	R/W	0

5.3.6.10 RCD_STA(0x4A4 新增)

RCD 状态寄存器

位	位名称	功能描述	R/W	复位值
31:3	--	保留	R	0
2	RCDTrig_B	Trig_B 内部信号锁存寄存器，高电平有效。寄存器输出一旦从低电平切换到高电平生效，则始终保持在高电平直到模块复位或重新使能启动工作为止。	R	0

1	RCDTrig_A	Trig_A 内部信号锁存寄存器，高电平有效。寄存器输出一旦从低电平切换到高电平生效，则始终保持在高电平直到模块复位或重新使能启动工作为止。	R	0
0	RCDTrig	RCD Trig 端口输出信号，高电平有效	R	0

5.3.6.11 TRIG_CTRL(0x4B0 新增)

TRIG 控制寄存器

位	位名称	功能描述	R/W	复位值
31:6	--	保留	R	0
5	RCDPostSel	剩余电流检测输入信号选择 0: RCD 锁存前输入 1: RCD 锁存后输入	R/W	0
4	RcdTrigSel	硬件脱扣信号选择 0: 专用硬件脱扣信号 1: 剩余电流检测输入信号	R/W	0
3	TrigOutSel	脱扣信号输出选择 0: 通用 or 专用脱扣信号发生信号 1: 通用脱扣信号发生信号	R/W	0
2	TrigMaskHw	通用信号发生器硬件触发屏蔽控制 ‘1’: 屏蔽硬件源触发（不响应剩余电流硬件检测触发信号） ‘0’: 不屏蔽硬件源触发（响应剩余电流硬件检测触发信号）	R/W	1
1	TrigSigVal	脱扣信号发生器 Trig 信号输出有效电平控制 ‘1’高电平有效 ‘0’低电平有效	R/W	1
0	TrigSigMode	脱扣 Trig 信号功能模式选择 ‘0’: 始终有效模式 ‘1’: 长度可配模式	R/W	1

5.3.6.12 TRIG_EN(0x4B4 新增)

通用脱扣软件使能控制寄存器

位	位名称	功能描述	R/W	复位值
31:1	--	保留	R	0
0	TrigEnSw	通用脱扣发生器软件使能控制 0: 无动作 1: 启动脱扣信号发生 上升沿触发，每次配置前先写 0 再写 1。当脱扣信号发生处于空闲状态时配置该位有效。	R/W	0

5.3.6.13 TRIG_STOP(0x4B8 新增)

脱扣结束控制寄存器

位	位名称	功能描述	R/W	复位值
31:2	--	保留	R	0
1	TrigStopHw	硬件专用脱扣信号发生器结束控制 ‘0’：无动作 ‘1’：结束当前脱扣信号发生进入空闲状态 上升沿触发，该寄存器控制仅在工作状态下配置有效,每次配置前先写 0 再写 1， 配置该信号关闭信号不触发中断	R/W	0
0	TrigStopSw	通用脱扣信号发生器结束控制 ‘0’：无动作 ‘1’：结束当前脱扣信号发生进入空闲状态 上升沿触发，该寄存器控制仅在工作状态下配置有效,每次配置前先写 0 再写 1， 配置该信号关闭信号不触发中断	R/W	0

5.3.6.14 TRIG_LEN(0x4BC 新增)

通用脱扣 TRIG 信号长度寄存器

位	位名称	功能描述	R/W	复位值
31:18	--	保留	R	0
17:0	TrigSigLen	脱扣信号发生器 Trig 信号长度配置 实际长度: $(N+1)*542.528\text{ns}$ 18 位@1.84M 时钟 默认值为 40ms	R/W	0x12001

5.3.6.15 TRIG_DLY(0x4C0 新增)

通用脱扣 TRIG 启动延迟寄存器

位	位名称	功能描述	R/W	复位值
31:16	--	保留	R	0
15:0	TrigSigStatDelay	脱扣信号启动延迟 当配置成 0 时，无 delay。 当配置成非 0 的其他值时， 时间延迟: $(N+1)*542.528\text{ns}$ 16 位@1.84M 时钟	R/W	0

5.3.6.16 TRIG_STA(0x4C4 新增)

通用脱扣 TRIG 状态寄存器

位	位名称	功能描述	R/W	复位值
31:22	--	保留	R	0
21:4	TrigCnt	脱扣内部计数器	R	0

3	--	保留	R	0
2	TrigGen	脱扣信号有效状态寄存器 '0': 无效状态 '1': 有效状态	R	0
1	TrigBusy	脱扣信号忙状态寄存器 '0': 空闲状态 '1': 工作状态	R	0
0	TrigOut	脱扣输出信号 (TrigOutSel 选择控制)	R	0

5.3.6.17 TRIG_LEN2(0x4C8 新增)

专用硬件脱扣长度寄存器

位	位名称	功能描述	R/W	复位值
31:18	--	保留	R	0
17:0	TrigSigLenHw	专用硬件脱扣信号发生器 Trig 信号长度配置 实际长度: $(N+1)*542.528\text{ns}$ 18 位@1.84M 时钟 该配置参数需大于 0, 默认值为 40ms	R/W	0x12001

5.3.6.18 TRIG_STA2(0x4CC 新增)

专用硬件脱扣状态寄存器

位	位名称	功能描述	R/W	复位值
31:22	--	保留	R	0
21:4	TrigCntHw	脱扣内部计数器	R	0
3:2	--	保留	R	0
1	TrigBusyHw	脱扣信号忙状态寄存器 '0': 空闲状态 '1': 工作状态	R	0
0	--	保留	R	0

5.4 特殊命令

命令名称	命令寄存器	数据	描述
写使能命令	1A8	0xE5	使能计量模块写操作
写保护命令	1A8	0xDC	关闭计量模块写操作
电流通道 A 选择命令	1A8	0x5A	电流通道 A 设置命令, 指定当前用于计算有功电能/无功电能的电流通道为通道 A; 对有效值和功率寄存器无影响; 当写使能之后, 系统才接受该命令; 计量状态寄存器中的 CHNSEL 寄存器位反映了该命令的执行结果。

电流通道 B 选择命令	1A8	0xA5	<p>电流通道 B 设置命令，指定当前用于计算有功电能/无功电能的电流通道为通道 B；对有效值和功率寄存器无影响；</p> <p>当写使能之后，系统才接受该命令；计量状态寄存器中的 CHNSEL 寄存器位反映了该命令的执行结果。</p>
----------------	-----	------	--

写保护的范围：

原有配置寄存器(0x00~0x6C)、新增配置寄存器(0x74~0xD0)、新增 DMA 波形缓存配置寄存器(0x400~0x410、0x420~0x42C)、新增 RCD 配置寄存器(0x480~0x490)、原有快速脉冲寄存器(0xC0~C8、0x100~108)、新增快速脉冲寄存器(0xE0~FC)、原有中断使能寄存器 EMMIE (0x18C)、**新增中断使能寄存器 EMMIE2(0x1A0)、EMMIE3(0x1B0)**用特殊命令写使能后才能写入修改，具体命令格式如上表。

5.5 校表方法

支持的校表方法：

- 脉冲法校表
- 功率法校表

5.5.1 脉冲法校表

步骤及算法：

A 基本参数确定（该项内容可以在 soc 程序中固定）

1) 电压、电流转换系数

电压转换系数： $K_v = R_a / (R_t * U_{pga} * 2^{23})$

R_t: 电阻分压串的取样电阻

R_a: 电阻串的总电阻

U_{pga}: 电压通道 ADC 增益放大倍数

电流转换系数： $K_i = 1 / (R_i * I_{pga} * 2^{23})$

R_i: 锰铜取样电阻（在采用互感器的情况下 R_i=R₀/PT,其中 R₀ 为互感器二次侧的取样电阻，PT 为互感器的变比）

I_{pga}: 电流通道 ADC 增益放大倍数

2) 确定电表脉冲常数 EC 及 HFConst 寄存器值、功率转换系数

电表脉冲常数 EC: 供测试和电能计算（表型确定，如 1200imp/kwh）

功率转换系数： $K_p = R_a / [(R_i * I_{pga}) * (R_t * U_{pga}) * 2^{31}]$

R_t: 电压采样回路分压电阻串的取样电阻

R_a: 电压采样回路分压电阻串的总电阻

R_i: 锰铜取样电阻（在采用互感器的情况下 R_i=R₀/PT,其中 R₀ 为互感器二次侧的取样电阻，PT 为互感器的变比）

U_{pga}: 电压通道 ADC 增益放大倍数

I_{pga}: 电流通道 ADC 增益放大倍数

HFConst=INT[(R_i*I_{pga})*(R_t*U_{pga})/R_a]*3.6*10⁶*fd2f/(2*EC)]
=INT[1.8*10⁶*fd2f*(R_i*I_{pga}*R_t*U_{pga})/(R_a*EC)]
 fd2f 为 0.9216Mhz (0.9216*10⁶)

3) 有功启动功率、无功启动功率确定

P_{start} (0CH) = 0.7 * P_{启动} * (1/K_p) / 2⁸ ;

P_{启动}: 启动时的功率 单位: w

K_p: 功率转换系数

系数 0.7 为建议值，实际应用时可做调整。
Qstart(10H)值与 Pstart 相等或者根据需要自行调整。

B 电压、电流有效值校正：台体加 Un、Ib、功率因数 1.0

主要得到电流、电压通道增益 IAGain (44h)、UGain (4ch) 值

$$IAGain = \begin{cases} (I0/I) - 1) * 2^{15} & I0 > I \\ (I0/I) - 1) * 2^{15} + 2^{16} & I0 < I \end{cases}$$

$$I0 = Ib * 1/Ki$$

I: 电流有效值寄存器 (IARMS (10CH)) 测量值

Ib: 标准表显示电流值

Ki: 电流转换系数

$$UGain = \begin{cases} (U0/U) - 1) * 2^{15} & U0 > U \\ (U0/U) - 1) * 2^{15} + 2^{16} & U0 < U \end{cases}$$

$$U0 = Un * 1/Kv$$

U: 电压有效值寄存器 (URMS (114H)) 测量值

Un: 标准表显示电压值

Kv: 电压转换系数

C 1.0 功率增益的误差校正：台体仍加 Un、Ib、功率因数 1.0

经过前面步骤 B 后，1.0 误差基本准确，要求不高可以忽略。

主要确定功率增益寄存器 GPQA (14H) (无功增益写 GPQA 相同值)

$$Pgain = \frac{-err}{1 + err} \quad (err: \text{台体显示误差值})$$

如果 Pgain ≥ 0, 则 GPQA = INT[Pgain * 2¹⁵]

否则 Pgain < 0, 则 GPQA = INT[2¹⁶ + Pgain * 2¹⁵]

D 0.5L 相位校正：台体仍加 Un、Ib、功率因数 0.5L

$$\theta = \arcsin \frac{-err}{\sqrt{3}} \quad (err: \text{台体显示误差值})$$

对 50HZ, PHSA/B 有 0.0097656⁰/LSB 的关系, 则有

如果 θ ≥ 0, PHSA/B = INT((θ * 180 / 3.1415928) / 0.0097656⁰)

如果 θ < 0, PHSA/B = INT(2⁹ + θ * 180 / 3.1415928) / 0.0097656⁰)

对于 60Hz, 校正刻度为 0.01171875⁰/LSB

E 有功偏置 OFFSET 校正：台体加 Un、10%Ib 或 5%Ib、功率因数 1.0

$$APOSa(\text{地址 } 28H) = \begin{cases} (P0 * 1/Kp) * (-err) & (err < 0 \text{ 时}) \\ = 2^{16} + (P0 * 1/Kp) * (-err) & (err > 0 \text{ 时}) \end{cases}$$

P0: 标准表显示功率值 err: 台体显示误差值

F 电流有效值 OFFSET 校正：台体加 Un 空载

读取 10 次电流有效值寄存器 (间隔时间 100ms 或更长时间) 求平均, 平方后取反, 将 bit23~bit8 写入电流 offset 寄存器 IARMSOS (38h)

$$IARMSOS(38H) = (2^{24} - Iavreg^2) / 2^8$$

Iavreg: 10 次电流有效值寄存器 (IARMS (10cH)) 平均值

5.5.2 功率法校表

功率校表法相对脉冲校表法具有快捷、简单、高效的优点，可以支持单点，但对台体的稳定性有要求。

台体加 U_n 、 I_b 、功率因数 0.5L

步骤及算法：

A 基本参数确定（该项内容可以在 soc 程序中固定，同脉冲法）

- 1) 电压、电流转换系数
- 2) 确定电表脉冲常数 EC 及 HFConst 寄存器值、功率转换系数
- 3) 有功启动功率、无功启动功率确定

B 电压、电流有效值校正

同脉冲校表法

C 1.0 功率增益的误差校正：可忽略

D 0.5L 相位校正：

计算公式参照脉冲精度校表法，不同点功率法的 err 是通过功率计算的。

公式 $err = [P - P_0 * (1/K_p)] / (P_0 * (1/K_p))$

$$\theta = \text{Arcsin} \frac{-err}{\sqrt{3}}$$

P：测量的功率寄存器值

对 50HZ，PHSA/B 有 $0.0097656^0/\text{LSB}$ 的关系，则有

如果 $\theta \geq 0$ ，PHSA/B = $\text{INT}((\theta * 180 / 3.1415928) / 0.0097656^0)$

如果 $\theta < 0$ ，PHSA/B = $\text{INT}(2^9 + \theta * 180 / 3.1415928) / 0.0097656^0$

E 有功偏置 OFFSET 校：同脉冲精度校表法

$err = [P - P_0 * (1/K_p)] / (P_0 * (1/K_p))$

P：测量的功率寄存器值（读 10 次求平均）

P0：标准表显示的功率值

$APOSA(28) = (P_0 * 1/K_p) * (-err)$ ($err < 0$ 时)

$= 2^{16} + (P_0 * 1/K_p) * (-err)$ ($err > 0$ 时)

F 电流有效值 OFFSET 校正：同脉冲精度校表法

读取 10 次电流有效值寄存器（间隔时间 100ms）求平均，平方后取反，

将 bit23~bit8 写入电流 offset 寄存器 IARMSOS (38h)

$IARMSOS(38h) = (2^{24} - I_{avreg}^2) / 2^8$

I_{avreg} ：10 次电流有效值寄存器（IARMS (10cH)）平均值

6 全失压测量 NVM

全失压测量是一种低功耗的计量模式，CPU 工作在 32K 时，NVM 计量功耗优于 600uA。

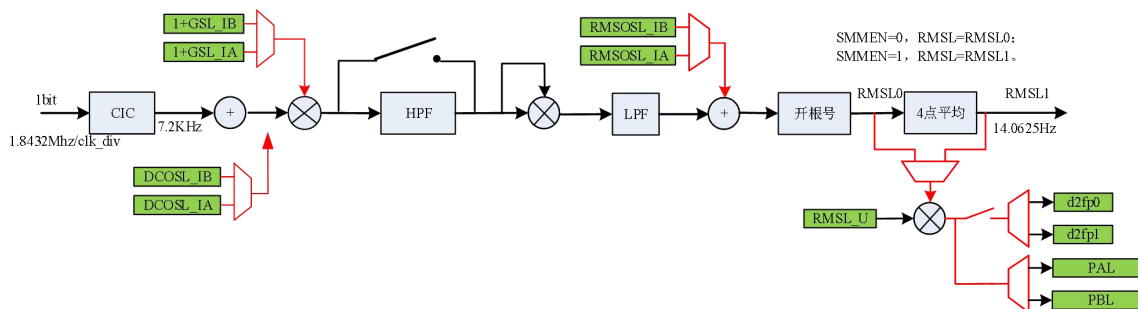
6.1 主要特点

BaseAddr 为：0x40040000

- 5% I_b 点误差小于 0.5%；
- 功耗小于 0.6mA；
- 支持 IA/IB 测量，单路测量时间 80ms；
- 极低直流 offset；

- 支持 IA/IB 双路直流 Offset 校正;
- 支持 IA/IB 双路增益校正;
- 支持 IA/IB 双路交流 Offset 校正;
- 支持 1bit 速率 1.8432MHz 及其 2/4/8 分频;
- 支持自动将功率值填入 D2F 寄存器进行积分

6.2 功能框图



NVM 有两种功耗模式:

PM_SEL=0 时, ADC 时钟源为 RCH 分频后的时钟 1.8M, 默认向下兼容, 该模式下 NVM 模式为大功耗模式, NVM 测量时功耗 1.5mA。

PM_SEL=1 时, ADC 时钟源为 RCM 分频后的时钟 1.8M, 该模式下 NVM 模式为小功耗模式, NVM 测量时功耗 600uA。

PM_SEL 的描述见 OSC_CTL2 寄存器 bit22。

6.3 NVM 测量模式

6.3.1 单通道模式 (向下兼容)

1. 通过系统控制使能 IA 或者 IB 通道的 ADC (每次只可使能一个)。
2. 启动 NVM 测量。
3. 测量完成, 通过寄存器读取当前通道的有效值。

两个通道都是用同一套寄存器 LS_DCOS、LS_THO、LS_RMS。

6.3.2 双通道同时测量模式

1. 通过配置寄存器, 选择 NVM 新模式, 该模式下两个通道的 DCOS、THO、GS、RMS 都是独立寄存器配置。并且 IA 和 IB 通道的有效值比较结果分别输出到两个独立的标志位, 可出中断。
2. 通过系统控制同时使能 IA 和 IB 通道的 ADC。
3. 启动 NVM 测量。
4. 测量完成, 通过寄存器同时获得 IA 和 IB 通道的电流有效值。同时可获得两个通道有效值的比较结果。

6.4 寄存器列表

BaseAddr 为: 0x40040000

计量部分寄存器列表

地	名称	R/W	有效字长	复位值	功能描述
---	----	-----	------	-----	------

址					
校表参数和计量控制寄存器					
00H	NVM_IE	R/W	1	07H	全失压中断使能寄存器
04H	NVM_IF	R/W	1	0	全失压中断标志寄存器
08H	LSCFG	R/W	2	0	全失压配置寄存器
0CH	LSDCOS	R/W	3	0	全失压直流offset校正寄存器
10H	LSTHO	R/W	3	0	全失压测量阈值设置寄存器
14H	LSRMS	R	3	0	全失压测量有效值寄存器，周期更新
18H	LSRMS1	R	3	0	全失压测量有效值寄存器，计算完成更新
1CH	---	R/W	1	0	保留
20H	HFCnst	R/W	2	0	自定义脉冲频率寄存器
24H	D2FP0	R/W	3	0	自定义功率寄存器 0
28H	D2FP1	R/W	3	0	自定义功率寄存器 1
2CH	LSMODE	R/W	2	0	NVM 模式配置寄存器
30H	LSDCOSIB	R/W	3	0	NVM 模式 IB 通道直流 Offset 校正寄存器，仅在双通道模式有效。
34H	LSTHOIB	R/W	3	0	NVM 模式 IB 通道全失压阈值寄存器，仅在双通道模式有效。
38H	LSGSIA	R/W	2	0	NVM 模式 IA 通道增益寄存器
3CH	LSGSIB	R/W	2	0	NVM 模式 IB 通道增益寄存器，仅在双通道模式有效。
40H	LSRMSIB	R	3	0	NVM 模式 IB 通道有效值寄存器，按照固定周期在更新；仅在双通道模式有效。
44H	LSRMSIB1	R	3	0	NVM 模式 IB 通道有效值寄存器，在计算完成后只更新 1 次。仅在双通道模式有效。
48H	LSADCINCFG	R/W	1	0	NVM 模式 ADC 输入配置寄存器
4CH	LSRMSUA	R/W	3	0	NVM 模式自定义电压有效值寄存器
50H	LSPA	R/W	3	0	NVM 模式 A 路有功功率寄存器
54H	LSPB	R/W	3	0	NVM 模式 B 路有功功率寄存器

6.5 寄存器说明

6.5.1 NVM_IE (0x0)

全失压中断使能寄存器

偏移地址：0x0；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:10	---	保留	R	
9	ib_ov_ie	全失压计算完成并且 IA 通道超过设定阈值中断使能： =0：不使能 =1：使能	R/W	0
8	ia_ov_ie	全失压计算完成并且 IA 通道超过设定阈值中断使能： =0：不使能 =1：使能	R/W	0
7:6	--	保留	R	0
5	D2F1_CF_IE	D2F1 脉冲输出中断使能 =0：不使能	R/W	0

		=1: 使能		
4	D2F0_CF_IE	D2F0 脉冲输出中断使能 =0: 不使能 =1: 使能	R/W	0
3	VREF_LOW_IE	发生 VREF 复位中断使能 =0: 不使能 =1: 使能	R/W	0
2	LDO3_LOW_IE	发生 LDO3 复位中断使能 =0: 不使能; =1: 使能	R/W	0
1	NVM_DoneIE	全失压计算完成并且超过设定阈值中断使能: =0: 不使能 =1: 使能	R/W	0
0	NVCAL_DoneIE	全失压计算完成中断使能 =0: 不使能 =1: 使能	R/W	0

6.5.2 NVM_IF (0x4)

全失压中断标志寄存器

偏移地址: 0x4; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:9	---	保留	R	
9	ib_ov_if	全失压计算完成并且 IB 通道超过设定阈值中断标志: =0: 未发生 =1: 产生事件	R/W	0
8	ia_ov_if	全失压计算完成并且 IA 通道超过设定阈值中断标志: =0: 未发生 =1: 产生事件	R/W	0
7	nvm_bgrok_sp	BGR 工作状态标志 =0: 未正常工作 =1: 正常工作	R/W	0
6	nvm_avddok_sp	AVDD 工作状态标志 =0: 未正常工作 =1: 正常工作	R/W	0
5	D2F1_CF_IF	D2F1 脉冲输出中断标志 =0: 未发生 =1: 产生事件	R/W	0
4	D2F0_CF_IF	D2F0 脉冲输出中断标志 =0: 未发生 =1: 产生事件	R/W	0
3	VREF_LOW_IF	发生 VREF 复位中断标志 =0: 未发生 =1: 产生事件	R/W	0
2	LDO3_LOW_IF	发生 LDO3 复位中断标志	R/W	0

		=0: 未发生 =1: 产生事件		
1	NVM_DoneIF	全失压计算完成并且超过设定阈值中断标志 =0: 未发生 =1: 产生事件	R/W	0
0	NVCAL_DoneIF	全失压计算完成中断标志 =0: 未发生 =1: 产生事件	R/W	0

6.5.3 LSCFG (0x8)

全失压配置寄存器

偏移地址: 0x8; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:13	---	保留	R	
12	D2F_ATMODE	D2F 自动模式使能位 =0: 不自动把功率填进 D2F; =1: 自动把功率填进 D2F; 单通道模式: 如果使能 A 通道, PA 填进 D2FP0; 如果使能 B 通道, PB 填进 D2FP1。 双通道模式: 同时把 PA 填进 D2FP0, PB 填进 D2FP1。 注: 自动 D2F 时, 有效值计算完成后, 需要等待 1clk 功率才更新	R/W	0
11	CIC_MODE	=0, CIC 固定点数, 每周波 128 点; =1, 内部有两个 CIC, 实时测频进行补偿;	R/W	0
10:8	RMS_DIV	有效值 RMSL_IA 和 RMSL_IB 输出值配置: =000, 不平均; =001, 2 点平均; =010, 4 点平均; =011, 8 点平均; =100, 16 点平均; =101, 32 点平均; =110, 64 点平均; =111, 128 点平均。	R/W	0
7:6	RMS_SP	有效值 RMSL_IA 和 RMSL_IB 输出值配置, 开始平均点数配置寄存器: =00, 有效值从第 1 个点开始平均计算; =01, 有效值从第 2 个点开始平均计算, 前 1 个点丢弃; =10, 有效值从第 3 个点开始平均计算, 前 2 个点丢弃; =11, 有效值从第 4 个点开始平均计算, 前 3 个点丢	R/W	0

		弃。		
5	D2F1_CF_EN	D2F1 脉冲输出使能： =0 时 D2F1 模块的脉冲不输出到 IO 口，但是中断和标志都有； =1 时 D2F1 模块的脉冲输出输出到 QF 脉冲的 IO 口（P50 或 P51，依据 IO 口复用配置）	R/W	0
4	D2F0_CF_EN	D2F0 脉冲输出使能： =0 时 D2F0 模块的脉冲不输出到 IO 口，但是中断和标志都有； =1 时 D2F0 模块的脉冲输出输出到 PF 脉冲的 IO 口（P50 或 P51，依据 IO 口复用配置）	R/W	0
3	SMMEN	NVM 有效值更新时间配置： =0：有效值 20ms 更新，80ms 时稳定并发中断 =1：有效值 80ms 更新，160ms 时稳定并发中断	R/W	0
2	NVMEN	NVM 计算使能： =0：不使能 =1：使能 NVM 计算	R/W	0
1	HPFON_LS	NVM 高通使能： =0：关闭高通 =1：开启高通 默认高通关闭，建议使用直流 offset 自动校正功能，不使用高通，以加快稳定时间。	R/W	0
0	LSDC_EN	直流 offset 自动校正使能： =0：不使能直流 offset 自动校正功能 =1：使能全失压直流 offset 自动校正工	R/W	0

6.5.4 LSDCOS (0xC)

全失压直流 offset 寄存器

偏移地址：0xC；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:24	---	保留	R	
23:0	LS_DCOS	全失压测量时进行直流 offset 校正，offset 值与 24bit 采样值直接相加 LS_MODE.NVM_MODE=0：IA 通道和 IB 通道都用该寄存器配置。 LS_MODE.NVM_MODE=1：IA 通道使用该寄存器配置，IB 通道使用 LS_DCOSIB 进行配置。	R/W	0

6.5.5 LSTHO (0x10)

全失压有效值阈值设置寄存器

偏移地址：0xC；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:24	---	保留	R	
23:0	LS_THO	全失压测量阈值设置寄存器，将 RMS_L 与该阈值进行	R/W	0

		比较。 LS_MODE.NVM_MODE=0: IA 通道和 IB 通道都用该阈值比较寄存器。 LS_MODE.NVM_MODE=1: IA 通道使用该阈值比较寄存器, IB 通道使用 LS_THOIB 进行配置。		
--	--	---	--	--

6.5.6 LSRMSx (0x14~0x18/0x40/0x44, 新增)

全失压有效值寄存器

偏移地址	14H	18H	40H	44H
寄存器	LSRMS	LSRMS1	LSRMSIB	LSRMSIB1
默认值	0x0	0x0	0x0	0x0

全失压有效值寄存器均为 24 位有符号数, 最高位为 0 表示有效数据, 最高位为 1 时读数做零处理。

LS_RMSx 按照固定周期在更新, 更新时间根据 LS_CFG 寄存器进行配置;

LS_RMSx1 在计算完成后不再更新, 更新时间根据 LS_CFG 寄存器进行配置。

LS_MODE.NVM_MODE=0: 为单通道模式, LS_RMS 和 LS_RMS1 根据系统控制 SYS_PD 寄存器使能哪路 ADC, 则输出该通道的有效值 (IA 或 IB), LS_RMSIB 和 LS_RMSIB1 无效。

LS_MODE.NVM_MODE=1: 为双通道模式, LS_RMS 和 LS_RMS1 固定输出 IA 通道有效值, LS_RMSIB 和 LS_RMSIB1 固定输出 IB 通道有效值。

有效值计算起始点及平均点数可配, 见 LS_CFG 寄存器 RMS_SP 和 RMS_DIV 描述。

以 IA 通道为例, 全失压测量有效值与正常计量有效值的关系为: $LS_RMS=(2*\sqrt{2}/\pi)*IARMS$

6.5.7 HFConst (0x20)

自定义脉冲频率寄存器

偏移地址: 0x20; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:16	---	保留	R	
15:0	HFConst	自定义脉冲频率寄存器, 同时作为 D2F 模块的时钟使能信号 =0: 时钟关闭 =其他, 时钟开启	R/W	0

6.5.8 D2FPx (0x24~0x28)

自定义功率寄存器

偏移地址	24H	28H
寄存器	D2FP0	D2FP1
默认值	0x0	0x0

自定义功率寄存器, 24bit 有效, 最高位为符号位。

用户可填入功率值, 依据 HFConst 的配置输出不同频率的脉冲 D2Fx_CF, 高电平脉宽固定为 82ms, 当周期小于 164ms 时输出等 duty 波形。

6.5.9 LSMODE (0x2C, 新增)

全失压模式配置寄存器

位	位名称	功能描述	读写标志	复位值
---	-----	------	------	-----

31:16	---	保留	R	0
15:0	NVM_MODE	NVM 模式配置寄存器： =写 0x5901，NVM 模式配置为双通道模式，读为 1； =写其他值，向下兼容，NVM 模式配置为单通道模式，读为 0。	R/W	0

6.5.10 LSDCOSIB (0x30, 新增)

全失压 IB 通道直流 offset 寄存器，WAVECFG.MODE_SEL=0 且 LS_MODE.NVM_MODE=1 时有效

位	位名称	功能描述	读写标志	复位值
31:24	---	保留	R	0
23:0	LS_DCOSIB	全失压测量时进行 IB 通道直流 Offset 校正，Offset 值与 24bit 采样值直接相加。 EMU_WAVECFG.MODE_SEL=0 且 LS_MODE.NVM_MODE=1	R/W	0

6.5.11 LSTHOIB (0x34, 新增)

全失压 IB 通道有效值阈值设置寄存器，WAVECFG.MODE_SEL=0 且 LS_MODE.NVM_MODE=1 时有效

位	位名称	功能描述	读写标志	复位值
31:24	---	保留	R	0
23:0	LS_THOIB	全失压测量 IB 通道阈值设置寄存器，将 LS_RMSIB 与该阈值进行比较。 EMU_WAVECFG.MODE_SEL=0 且 LS_MODE.NVM_MODE=1	R/W	0

6.5.12 LSGSIX (0x38~0x3C, 新增)

全失压有效值增益寄存器

偏移地址	2CH	30H
寄存器	LSGSIA	LSGSIB
默认值	0x0	0x0

全失压有效值增益寄存器，16bit 有符号数，WAVECFG.MODE_SEL=0 时有效。

功率自动积分模式使用，硬件自动计算有效值 $LS_RMS1 * LS_GSIA$ ，填入 D2FP0；硬件自动计算有效值 $LS_RMSIB * LS_GSIB$ ，填入 D2FP1。

6.5.13 LSADCINCFG (0x48 新增)

NVM 模式 ADC 输入配置寄存器

偏移地址：48H；默认值：0x0

位	名称	功能描述	读写标志	复位值
31:24	WKEY	写密码寄存器： =0xEA，低 24 位可写； =其他值，低 24 位不可写。	WO	0
23:6	--	只读，不可写。	R	0
5	ua_smp_sel	U 通道 1bit 采样边沿选择寄存器： =0，上升沿；	R/W	0

		=1, 下降沿。		
4	ib_smp_sel	IB 通道 1bit 采样边沿选择寄存器: =0, 上升沿; =1, 下降沿。	R/W	0
3	ia_smp_sel	IA 通道 1bit 采样边沿选择寄存器: =0, 上升沿; =1, 下降沿。	R/W	0
2	ua_inv_en	U 通道 1bit 输入反向使能寄存器: =0, 无操作; =1, 反向。	R/W	0
1	ib_inv_en	IB 通道 1bit 输入反向使能寄存器: =0, 无操作; =1, 反向。	R/W	0
0	ia_inv_en	IA 通道 1bit 输入反向使能寄存器: =0, 无操作; =1, 反向。 备注: 用于 ADC 输入反接的情况。若 ADC 输入反接, 配置 1bit 输入反向可令采样信号符号正确, 抵消 ADC 输入反接的错误。	R/W	0

6.5.14 LSRMSU (0x4C, 新增)

NVM 模式自定义电压有效值寄存器

偏移地址	4CH
寄存器	RMSL_U
默认值	0x0

NVM 模式自定义电压有效值寄存器为 24 位有符号数。

该寄存器与 RMS_L1_IA 和 RMS_L1_IB 做乘法, 得到功率保存到功率寄存器 LS_PA 和 LS_PB, 同时可配置自动填入 D2F。

6.5.15 LSPx (0x50~0x54, 新增)

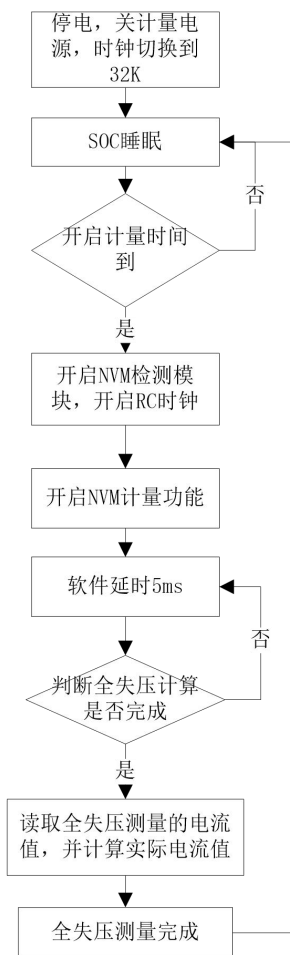
NVM 模式有功功率寄存器

偏移地址	50H	54H
寄存器	LSPA	LSPB
默认值	0x0	0x0

自定义功率寄存器, 24bit 有效, 最高位为符号位。

6.6 实现方式

6.6.1 实现流程图



6.6.2 程序实现步骤

按照上面的流程图，程序可周期性的打开全失压模块来进行电流的测量，程序实现及寄存器设置步骤如下：

- 1、SOC 掉电，进入低功耗模式，所有外设时钟均关闭，SOC 主时钟运行在 32768Hz。
- 2、设定的计量间隔时间满足后，初始化 NVM 模块，进行电流测量，初始化步骤：
 - 1) 使能系统控制寄存器密码保护，既 SYSCTL->SYS_PS 写为 82H。
 - 2) 设置 SYSCTL->SYS_PD 寄存器第 bit0 为 0，I1 通道电源上电。
 - 3) 如电流测量通道使用锰铜分流器，设置 SYSCTL->ADC_CTRL 寄存器设置为 03H，I1 通道的增益设置为 16 倍，如使用互感器，则设置为 0，1 倍增益。
 - 4) 将 SYSCTL->MOD1_EN 的 bit8 设置为 1，既开启 NVM 模块的 APB 时钟。
 - 5) 将 SYSCTL->OSC_CTRL1 的 bit1 设置为 0，打开 RC 时钟，给 NVM 模块提供时钟。
 - 6) 设置 NVM->LS_DCOS，将校正过的直流偏置值写入。
 - 7) 清除 NVM->NVM_IF 的标志。
 - 8) 设置 NVM->LS_CFG 的 bit2 为 1，启动 NVM 的测量。
 经过上述步骤后，NVM 模块的初始化设置完成。
- 3、初始化完成后，软件进行延时，等待 NVM 测量完成，在此过程中可查询 NVM->NVM_IF 的 bit0，如为 1 即代表测量完成，可进行测量值的读取。一般等待时间为 80MS 左右。

- 4、读取 NVM->RMS_L 值，既为当前测量通道的寄存器值，与正常计量有效值的关系应是： $IARMS = RMS_L / (2 * \sqrt{2} / \pi)$ 。如在校表时对 IAGain 进行了校正，那么计算 IARMS 时需考虑 IAGain 的影响。
- 5、电流测量及计算完成，设置 SYSCTL->SYS_PD 的 bit0 为 1，关闭测量通道电源，设置 SYSCTL->MOD1_EN 的 bit8 为 0，关闭 NVM 模块的 APB 时钟，设置 SYSCTL->OSC_CTRL1 的 bit1 为 1，关闭 RC 时钟。
- 6、测量完成，等待测量间隔时间到在次启动测量。

6.7 全失压测量时直流 offset 校正过程

该步骤可在校表时执行

- 1、电表只加电压不加电流情况下，进行电流的自动 offset 校正
 - 2、电表在正常计量模式时，需关闭测量通道的数字高通滤波器。
 - 3、EMU->IAGAIN 通道增益寄存器设置为 0。
 - 4、设置 SYSCTL->SYS_PD，只打开全失压测量通道的电源，其他 ADC 通道电源关闭。
 - 5、SYSCTL->MOD1_EN 的 bit8 设置为 1，打开 NVM 模块的 APB 时钟。
 - 6、SYSCTL->OSC_CTRL1 的 bit1 设置为 0，打开 RC 时钟。
 - 7、设置 NVM->LS_DCOS 为 0，NVM->LS_CFG 的 bit2 设置为 1，启动 NVM 模块测量。
 - 8、等待 80MS 后，读取 NVM->RMS_L 寄存器值并保存在 EEPROM 中，做为 NVM 模块的直流 offset 校正值。
 - 9、恢复前面设置的各寄存器。
- 自动 offset 校正完成。

6.8 电能积分

部分应用需要在电池供电情况下，测量电流回路电流，电压按照额定电压进行电能积分，按照上面步骤，我们可以准确得到电流值，乘上额定电压得到当前的功率值。计算得到功率后建议使用以下方法进行电能的积分操作：

- 1、设置该章节 HFConst 寄存器（20H），当脉冲电能累加到 HFConst 值时，即累加得到 1/EC Kwh 的电能。可根据需要调整 HFConst 寄存器大小，以保证一定功率下输出准确的脉冲。
- 2、定时启动电流测量，将每次启动测量得到的电流值乘以固定系数，得到功率值，写入 D2FP0 或 D2FP1 寄存器；
- 3、soc 硬件自动完成电能积分，可以通过中断方式通知 CPU 积分得到一个脉冲。也可以通过 IO 口输出脉冲信号，用于精度校验。

7 闪变 FLK（新增）

7.1 主要特点

- 仅提供 600Hz 更新的瞬时电压值。
- 软件根据此电压值自行计算瞬时视感度(600Hz)、短时闪变 Pst(10 分钟)和长时闪变 Plt(2 小时)。

7.2 使用方法

闪变计算过程：

步骤	数据操作
1.采样	adc_in, 7.2KHz 更新
2.低通滤波器	可将 7.2Khz 直接降频到 600Hz。
3.抽取	7.2KHz 600Hz

4.平方并计算有效值	此部分由软件实现
5.平方去直流	此部分由软件实现
6.归一化	此部分由软件实现
7.高通	此部分由软件实现
8.巴特沃斯低通	此部分由软件实现
9.平滑处理	此部分由软件实现
10.加权	此部分由软件实现
12.平方	此部分由软件实现
13.低通	此部分由软件实现
14.乘系数	此部分由软件实现
15.计数统计	此部分由软件实现
16.Pst 计算	此部分由软件实现

硬件输入：瞬时电压值，更新速度为 7.2KHz

硬件输出：FLK 模块硬件计算到步骤 3，将 7.2KHz 瞬时采样值经过一个低通滤波器，降频到 600Hz 输出 24bit 瞬时电压值，对于 300Hz 以上分量有 40dB 的衰减，用于低成本闪变方案，CPU 读取该瞬时值，可用软件实现计算瞬时视感度(600Hz)、Pst(10 分钟)和 Plt(2 小时)。

具体操作请参考锐能微闪变应用笔记。

7.3 寄存器列表

基地址	0x50020000				
偏移地址	名称	R/W	有效字长	复位值	功能描述
0x0	FLK_EN	R/W	1	0x00	闪变模块使能，使能信号由 0 变为 1 时会将计算模块全部复位。
0x4	FLK_IE	R/W	1	0x00	闪变模块中断使能
0x8	FLK_IF	R/W	1	0x2	闪变模块中断标志
0xC	UA_600Hz	R/W	3	0x0	UA 通道瞬时采样值，更新速度为 600Hz
0x10	FLK_PASS	R/W	2	0x0	模块密码

注意：要使能该模块，首先要开启该模块的时钟，开启时钟的方法：

7.4 寄存器定义

7.4.1 FLK_EN (0x00)

闪变模块使能寄存器

偏移地址：0x00；字长：1 字节；默认值：0

比特位	名称	描述	R/W	复位值
31:01	---	只读，不可写。	R	0
0	FLK_EN	=0：闪变模块不计算； =1：闪变模块开始计算。 由 0 变 1 时，会将闪变模块计算单元复位。 只有 FLK_PASS 密码对时，该位才能写为 1。	R/W	0x0

7.4.2 FLK_IE (0x04)

闪变模块中断使能寄存器

偏移地址：0x04；字长：1 字节；默认值：0

比特位	名称	描述	R/W	复位值
31:01	---	预留	R	0
0	FLK_IE	=0: 不使能闪变模块计算完成中断 =1: 使能闪变模块计算完成中断	R/W	0

7.4.3 FLK_IF (0x08)

闪变模块标志寄存器

偏移地址：0x08；字长：1 字节；默认值：0

比特位	名称	描述	R/W	复位值
31:01	---	预留	R	0
0	FLK_IF	=0: 闪变模块计算没有完成 =1: 闪变模块计算完成 该标志位写 1 清零 该标志位是否产生与 IE 没有关系。	R/W	0

7.4.4 UA_600HZ (0x0C)

瞬时采样值寄存器

偏移地址：0CH；字长：3 字节；默认值：0

比特位	名称	描述	R/W	复位值
31:24	---	预留	R	0
23:00	UA_600hz	电压通道瞬时采样值, 7.2KHz 经过一个低通滤波器后抽取到 600Hz。用于软件自行计算闪变值。	R	0

7.4.5 FLK_PASS (0x10)

闪变模块密码寄存器

偏移地址：10H；字长：2 字节；默认值：0

比特位	名称	描述	R/W	复位值
31:16	---	预留	R	0
15:0	FLK_PASS	当该寄存器写入 0x2025 时, FLK_EN 寄存器才能写为 1; 该寄存器读数为 0。	W	0

8 RTC

8.1 概述

RTC 模块提供实时时钟、振荡器温度补偿、日历、闹钟、时钟脉冲输出等功能。

实时时钟用独立的时、分、秒寄存器跟踪时间。日历包括年、月、日以及星期寄存器，具有闰年闰月自动修正功能。时钟脉冲输出具有多种可选择频率用于时钟校准。提供闹钟/报警功能。

集成温度传感器，提供温度测量的数字结果。

8.2 特点

- 提供准确的温度值，-25℃~70℃范围内测温精度为±1℃
- 在常温下实现 RTC 的初始校正
- 自动完成 RTC 的温度补偿操作，不需要 CPU 参与
- 低功耗设计
- 频率调节精度达到 0.0339ppm
- 高稳定性的振荡器
- RTC 在不同模式下都不关闭，在低功耗下仍然正常工作
- 提供时钟和日历功能：输出寄存器中包括秒，分钟，小时，日期，月份，年份和星期等
- 具有自动闰年闰月调整功能，计时范围 100 年（00-99）
- 1 个闹钟中断功能，2 个定时器周期性中断功能，5 个时间中断功能（秒，分，时，月，日）
- 可输出未校正的频率 4Hz，8Hz，16Hz，32768Hz
- 可输出校正后的频率 1Hz，1/30Hz
- 新增支持 RTC 四次曲线温补，详细使用说明请参考锐能微应用笔记
- 提供 RTC 二次补偿

8.3 寄存器描述

RTC 模块的基址

模块名	物理地址	映射地址
RTC	BaseAddr 为：0x4003C000	Base1

RTC 模块的寄存器偏移地址

寄存器名	地址偏移量	描述
RTC 寄存器组		
RTC_CTRL	Offset+0x00	RTC 控制寄存器
RTC_SC	Offset+0x04	秒寄存器，写保护
RTC_MN	Offset+0x08	分钟寄存器，写保护
RTC_HR	Offset+0x0C	小时寄存器，写保护
RTC_DT	Offset+0x10	日寄存器，写保护
RTC_MO	Offset+0x14	月寄存器，写保护
RTC_YR	Offset+0x18	年寄存器，写保护
RTC_DW	Offset+0x1C	星期寄存器，写保护
RTC_CNT1	Offset+0x20	定时器 1 寄存器
RTC_CNT2	Offset+0x24	定时器 2 寄存器
RTC_SCA	Offset+0x28	秒闹钟寄存器
RTC_MNA	Offset+0x2C	分钟闹钟寄存器
RTC_HRA	Offset+0x30	小时闹钟寄存器
RTC_IE	Offset+0x34	RTC 中断使能寄存器
RTC_IF	Offset+0x38	RTC 状态寄存器
RTC_TEMP	Offset+0x3C	当前温度寄存器，可读可写，写保护
RTC_TEMP2（新增）	Offset+0xF8	当前温度寄存器 2，12bit，只读。
TPS_START（新增）	Offset+0x150	测温启动寄存器
TEMP_CAL（新增）	Offset+0x154	温度寄存器 3

RTC_CALPS (新增)	Offset+0xCC	RTC 二次补偿寄存器写保护, 写入 8'hA8, T0~T9 寄存器才会起作用。
RTC_CAL_T0 (新增)	Offset+0xD0	T0~T9 是 8bit 寄存器, 在硬件自动温补基础上对 RTC 的误差做二次补偿, 刻度是 0.25ppm; 补偿温度范围: $T < -30$ 度
RTC_CAL_T1 (新增)	Offset+0xD4	补偿温度范围: $-30 \leq T < -20$ 度
RTC_CAL_T2 (新增)	Offset+0xD8	补偿温度范围: $-20 \leq T < -10$ 度
RTC_CAL_T3 (新增)	Offset+0xDC	补偿温度范围: $-10 \leq T < 0$ 度
RTC_CAL_T4 (新增)	Offset+0xE0	补偿温度范围: $0 \leq T \leq 10$ 度
RTC_CAL_T5 (新增)	Offset+0xE4	补偿温度范围: $35 < T \leq 45$ 度
RTC_CAL_T6 (新增)	Offset+0xE8	补偿温度范围: $45 < T \leq 55$ 度
RTC_CAL_T7 (新增)	Offset+0xEC	补偿温度范围: $55 < T \leq 65$ 度
RTC_CAL_T8 (新增)	Offset+0xF0	补偿温度范围: $65 < T \leq 75$ 度
RTC_CAL_T9 (新增)	Offset+0xF4	补偿温度范围: $T > 75$ 度
温度补偿二次曲线寄存器组		
RTC_PS	0x40	补偿寄存器写保护寄存器
RTC_MODE	0x44	温度补偿模式寄存器
RTC_DOTA0	0x48	RTC 初始偏差校正寄存器
RTC_ALPHAL	0x4C	晶振低温部分温度系数寄存器
RTC_ALPHAH	0x50	晶振高温部分温度系数寄存器
RTC_XT0	0x54	晶振温度曲线顶点温度寄存器
RTC_TADJ	0x58	量产温度单点校正寄存器
RTC_ZT	0x60	摄氏零度校正值、ADCOffset 校正值
温度补偿四次曲线寄存器组		
RTC_MODE1	0x130	温度补偿曲线选择寄存器
RTC_XT1	0x134	晶振温度曲线顶点温度寄存器 1
RTC_ALPHA	0x138	晶振一次温度系数寄存器
RTC_BETA	0x13C	晶振二次温度系数寄存器
RTC_GAMMA	0x140	晶振三次温度系数寄存器
RTC_ZETA	0x144	晶振四次温度系数寄存器
结果输出寄存器组		
RTC_DOTAT	0x64	补偿总量寄存器
RTC_FPTR	0x74	PLL_1HZ 补偿寄存器
RTC_FDTR1S	0x78	RTC_1S 每 1S 补偿寄存器
RTC_FDTR30S	0x7C	RTC_1S 每 30S 补偿寄存器
RTC_FDTR120S	0x84	RTC_1S 每 120S 补偿寄存器

8.3.1

RTC 寄存器组

8.3.1.1 RTC 控制寄存器 RTC_CTRL(0x00)

比特位	名称	描述	读/写标志	复位值
31:12	---	只读, 不可写	R	0
11	---	保留位, 不要配置	R/W	0

10	Cal_busy	RTC 校正忙。=1 时表示 RTC 正在进行校正；=0 时表示 RTC 校正完成。	R	0
9	Wr_busy	RTC 寄存器写操作忙；A 版写操作需要等待 wr_busy=0，B/C 版不需要。	R	0
8	WRTC	RT 寄存器组写允许： 0：禁止 RTC 寄存器写操作； 1：允许 RTC 寄存器写操作。 注意： 该位对 RTC 寄存器组 00~1C/3C 有效，对 RTC_CTL[7:0]也有效。 写万年历时间寄存器的两种方法： 1) 按照“年月日时分秒”顺序写，当写入秒寄存器后时间开始从写入时刻起累计，注意此种方法存在秒写入之前分钟翻转的可能，所以写入后要读出做校验； 2) 按照“秒分时分年月日”顺序写，先写秒寄存器，万年历计数器被清零，只要在一秒内写入其他值，可以保证成功写入。 注意硬件对“年月日”有做合法性判断，不能按照“日月年”顺序写，只能连续写入“年月日”。	R/W	0
7:6	TSE	温度传感器允许位 00：禁止自动温补。 01：启动自动温补。按照 TCP 的设置进行周期性温补。 10：启动用户温补模式 0，温度寄存器可更改，由用户填入温度值，用户每写一次温度寄存器启动一次温度补偿； 11：启动用户温补模式 1，温度寄存器不可更改，每次写温度寄存器就启动一次温度补偿操作，温度寄存器的值由 SOC 测量得到。注：该寄存器仅有上电复位起作用。	R/W	00
5:3	TCP	温度补偿周期： 000:2S 001:10S 默认 010:20s 011:30s 100:1 分钟 101:2 分钟 110:5 分钟 111:10 分钟	R/W	001
02:00	FOUT	000：禁止输出 001：1Hz 输出（建议使用） 010：1/30Hz 输出 011：32768Hz 输出 100：16Hz 输出 101：8Hz 输出 110：4Hz 输出 111：1Hz 输出 注：该寄存器仅有上电复位起作用。	R/W	000

8.3.1.2 秒寄存器 RTC_SC(0x04)

地址：0x4003C000+ 0x04

比特位	名称	描述	读/写标志	复位值
31:07	---	只读，不可写	R	0
06:00	SC	存储时钟的秒值 BCD 码格式， SC[6:4] 为秒值的十位， SC[3:0] 为秒值的个位，秒值的范围为 0~59	R/W	-

8.3.1.3 分钟寄存器 RTC_MN(0x8)

比特位	名称	描述	读/写标志	复位值
31:07	---	只读，不可写	R	0
06:00	MN	存储时钟的分钟值 BCD 码格式， MN[6:4] 为分钟值的十位， MN[3:0] 为分钟值的个位，分钟值的范围为 0~59	R/W	-

8.3.1.4 小时寄存器 RTC_HR(0xC)

比特位	名称	描述	读/写标志	复位值
31:06	---	只读，不可写	R	0
05:00	HR	存储时钟的小时值 BCD 码格式， HR[5:4] 为小时值的十位， HR[3:0] 为小时值的个位，小时值的范围为 0~23。	R/W	-

8.3.1.5 日期寄存器 RTC_DT(0x10)

比特位	名称	描述	读/写标志	复位值
31:06	---	只读，不可写	R	0
05:00	DT	存储时钟的日期值 BCD 码格式， DT[5:4] 为日期值的十位， DT[3:0] 为日期值的个位，日期值的范围为 1~31。	R/W	-

8.3.1.6 月份寄存器 RTC_MO(0x14)

比特位	名称	描述	读/写标志	复位值
31:05	---	只读，不可写	R	0
04:00	MO	存储时钟的月份值 BCD 码格式， MO[4] 为月份值的十位， MO[3:0] 为月份值的个位，月份值的范围为 1~12	R/W	-

8.3.1.7 年份寄存器 RTC_YR(0x18)

比特位	名称	描述	读/写标志	复位值
31:08	---	只读，不可写	R	0
07:00	YR	存储时钟的年份值 BCD 码格式， YR[7:4] 为年份值的十位， YR[3:0] 为年份值的个位，年份值的范围为 0~99。	R/W	-

8.3.1.8 星期寄存器 RTC_DW(0x1C)

比特位	名称	描述	读/写标志	复位值
31:03	---	只读，不可写	R	0
02:00	DW	存储当前日期所对应的星期。 DW[2:0] 的计数循环为 0-1-2-3-4-5-6-0-1-2-...。	R/W	-

注：04~1CH 寄存器没有复位值，由于软件引起的复位不会造成时间信息改变。

8.3.1.9 RTC 定时寄存器 1RTC_CNT1(0x20)

比特位	名称	描述	读/写标志	复位值
31:09	---	只读，不可写	R	0
08	CNT1PD	= 0：1 秒定时来自万年历秒中断。 = 1：关闭定时器 1，重新开启时计数器会重新开始计（刻度 1 S，与万年历秒中断无关）	R/W	0
07:00	CNT	定时器1计数器预设值 无符号数，计数单位为1s。当计数值=(CNT+1)时，置位 RTCCNT1F标志。（最小可以每1秒产生一次中断，最大可以每256 秒产生一次中断） 注1：该定时器在RTC校正后是准确的。 注2：该中断与秒寄存器更新不同步，即该中断不一定在秒计数起始点产生，可在秒计数的任意时刻产生。	R/W	0

8.3.1.10 RTC 定时寄存器 2RTC_CNT2(0x24)

比特位	名称	描述	读/写标志	复位值
31:09	---	只读，不可写	R	0
08	CNT2PD	= 0：定时器来自内部固定的 1/256S 中断 = 1：关闭定时器 2，重新开启时计数器会重新开始计（刻度 1/256S）	R/W	0
07:00	CNT	定时器2计数器预设值 无符号数，计数单位为1/256s。当计数值=(CNT+1)时，置位 RTCCNT2F标志。（最小可以每1/256秒产生一次中断，最大可以每1秒产生一次中断） 注：该定时器源自32768Hz晶体，未经校正，有一定误差。	R/W	0

8.3.1.11 秒闹钟寄存器 RTC_SCA(0x28)

比特位	名称	描述	读/写标志	复位值
31:07	---	只读，不可写	R	0
06:00	SCA	秒闹钟值 BCD 码格式，SCA[6:4]为秒值的十位，SCA[3:0]为秒值的个位，秒值的范围为 0~59	R/W	0

8.3.1.12 分钟闹钟寄存器 RTC_MNA(0x2C)

比特位	名称	描述	读/写标志	复位值
31:07	---	只读，不可写	R	0
06:00	MNA	分钟闹钟值 BCD 码格式，MNA[6:4]为分钟值的十位，MNA[3:0]为分钟值的个位，分钟值的范围为 0~59	R/W	0

8.3.1.13 小时闹钟寄存器 RTC_HRA(0x30)

比特位	名称	描述	读/写标志	复位值
31:06	---	只读，不可写	R	0
05:00	HRA	小时闹钟值	R/W	0

		BCD 码格式， HRA[5:4] 为小时值的十位， HRA[3:0] 为小时值的个位，小时值的范围为 0~23。		
--	--	---	--	--

8.3.1.14 RTC 中断使能寄存器 RTC_IE(0x34)

比特位	名称	描述	读/写标志	复位值
31:10	---	只读，不可写	R	0
9	RTC_1S_SEL	秒中断（bit3 定义）来源选择： =0：选择 RTC 的秒脉冲，此选项秒中断跟万年历时间更新同步，建议客户选择此选项。 =1：根据系统时钟模式：hcmm 下选择 pll_1hz，非 hcmm 下选择 RTC 秒脉冲，注意秒中断跟万年历时间更新不同步。	R/W	0
8	IECLKEN	RTC 中断产生时钟使能； 当 RTC_IE[8:0]任何一位为高时，中断模块时钟打开； 当 RTC_IE[8:0]全部为低时，中断模块时钟才关闭；	R/W	0
7	MOIE	月份中断使能 0：不使能 1：使能	R/W	0
6	DTIE	日期中断使能 0：不使能 1：使能	R/W	0
5	HRIE	小时中断使能 0：不使能 1：使能	R/W	0
4	MNIE	分钟中断使能 0：不使能 1：使能	R/W	0
3	SCIE	秒中断使能 0：不使能 1：使能 注： 当 RTC-IE->RTC_1S_SEL 为 1 且处于 HCMM 模式，即选择 pll_1hz 为秒中断源时，秒中断与秒寄存器更新不同步，秒中断不一定在秒计数起始点产生，可在秒计数的任意时刻产生。 当 RTCIE->RTC_1S_SEL 为 0 即选择 RTC 秒脉冲为秒中断源时，秒中断与秒寄存器更新一致。 建议选择 RTC-IE->RTC_1S_SEL 为 0。	R/W	0
2	RTCCNT2IE	RTC 定时器 2 中断使能 0：不使能 1：使能	R/W	0
1	RTCCNT1IE	RTC 定时器 1 中断使能 0：不使能 1：使能	R/W	0
0	ALMIE	闹钟事件中断使能 0：不使能	R/W	0

		1: 使能		
--	--	-------	--	--

8.3.1.15 RTC 中断标志寄存器 RTC_IF(0x38)

比特位	名称	描述	读/写标志	复位值
31:8	---	只读, 不可写	R	0
7	MOF	月份中断标志位 0: 月份计数器未加 1 1: 月份计数器加 1 Note: 写 1 清零	R/W	0
6	DTF	日期中断标志位 0: 日期计数器未加 1 1: 日期计数器加 1 Note: 写 1 清零	R/W	0
5	HRF	小时中断标志位 0: 小时计数器未加 1 1: 小时计数器加 1 Note: 写 1 清零	R/W	0
4	MNF	分钟中断标志位 0: 分钟计数器未加 1 1: 分钟计数器加 1 Note: 写 1 清零	R/W	0
3	SCF	秒中断标志位 0: 秒计数器未加 1 1: 秒计数器加 1 Note: 写 1 清零, 秒中断位置为秒开始	R/W	0
2	RTCCNT2F	RTC 定时器 2 中断标志位 0: 定时器 1 中断未发生 1: 定时器 1 中断发生 Note: 写 1 清零	R/W	0
1	RTCCNT1F	RTC 定时器 1 中断标志位 0: 定时器 1 中断未发生 1: 定时器 1 中断发生 Note: 写 1 清零	R/W	0
0	ALMF	闹钟事件标志位, 与实时时钟匹配的闹钟事件发生 0: 闹钟事件未发生 1: 闹钟事件发生 Note: 写 1 清零	R/W	0

8.3.1.16 当前温度寄存器 RTC_TEMP(0x3C)

比特位	名称	描述	读/写标志	复位值
31:10	---	只读, 不可写	R	0
09:00	TEMP	当前温度值。 Bit9 为符号位; Bit8~2 为整数位; Bit1~0 为小数位。 // Temp[9] Temp[8:2] Temp[1] Temp[0] // 符号 -128 度~127 度 0.5 度 0.25 度	R/W	-

		<p>表示范围：-128 度(0x200)~+127.75 度(0x1ff)</p> <p>温度换算公式：若符号位为 0，则温度= TEMP/4 若符号位为 1，则温度=(2¹⁰- TEMP)/4</p> <p>TSE=00：禁止自动温补。此时 RTC_TEMP 寄存器无效，读出的值无意义；</p> <p>TSE=01：按照 RTC_CTL->TCP 设置的周期进行自动温补。此时 RTC_TEMP 寄存器显示的是该温度补偿周期的温度值，RTC_TEMP 寄存器的更新周期为 RTC_CTL->TCP 设置的温度补偿周期；</p> <p>TSE=10：启动用户温补模式 0。此时 RTC_TEMP 温度寄存器可更改，由用户填入温度值，用户每写一次温度寄存器就启动一次温度补偿；</p> <p>TSE=11：启动用户温补模式 1。此时 RTC_TEMP 温度寄存器不可更改，每次写温度寄存器就启动一次温度补偿操作，RTC_TEMP 温度寄存器的值由 SOC 测量得到。</p> <p>注意：在用户温补模式 0 下，并且当 RTC_MODE1=16'hEA65 启用四次曲线补偿时，TEMP 扩展为 16bit 数，bit15 是符号位，最小刻度为 0.25 度/2⁶=0.00390625 度；该模式下读取无意义。其他情况为 10bit 寄存器。</p>		
--	--	---	--	--

RTC 自动温补需要定义以下寄存器，这些寄存器值在客户量产环节获得。

1. 初始频率偏差寄存器 RTC_DOTA0：修正晶体的初始频率偏差；（每台表需要获得，锐能微提供的库函数可以完成该寄存器的操作）
2. 二次曲线顶点温度寄存器 RTC_XT0 （获得晶体批次参数，配置选项字节，通过编程界面写入）
3. 晶振温度系数寄存器 RTC_ALPHA （获得晶体批次参数，配置选项字节，通过编程界面写入）

8.3.1.17 当前温度寄存器 2RTC_TEMP2(0xF8) （新增）

比特位	名称	描述	读/写	复位值
31:12	---	预留	R	0
11:00	TEMP2	<p>当前温度值。</p> <p>Bit11 为符号位；Bit10~4 为整数位；Bit3~0 为小数位。</p> <p>// Temp[11] Temp[10:4] Temp[3] Temp[2] Temp[1] Temp[0]</p> <p>// 符号 -128 度~127 度 0.5 度 0.25 度 0.125 度 0.0625 度</p> <p>表示范围：-128 度~+127.75 度</p>	R	-

8.3.1.18 温度 OS 寄存器 RTC_TEMPOS(0xC4) （新增）

正常模式下温度 offset 寄存器，用户软件自行管理，不使用时请配置为 0

比特位	名称	描述	读 / 写 标志	复位值
31:06	---	预留	R	0
05:00	TEMPOS	当 TSE 等于 01 或者 11 时，温度寄存器 TEMP 读出的值=实际	R/W	0

		计算值+TEMPOS/4。 TEMPOS 是有符号数，用于对温度进行 offset 校正。 校正的范围是：±8 度 备注：只有上下电复位。仍然是跟 10bit 的温度值对应，即最小刻度仍然是 0.25 度		
--	--	---	--	--

8.3.1.19 温度 OS 寄存器 2 RTC_TEMPOS2(0x188) (新增)

低功耗模式下温度 offset 寄存器，用户软件自行管理，不使用时请配置为 0

比特位	名称	描述	读 / 写标志	复位值
31:06	---	预留	R	0
05:00	TEMPOS2	当 TSE 等于 01 或者 11 时，温度寄存器 TEMP 读出的值=实际计算值+TEMPOS2/4。 TEMPOS2 是有符号数，用于对温度进行 offset 校正。 校正的范围是：±8 度 备注：只有上下电复位。仍然是跟 10bit 的温度值对应，即最小刻度仍然是 0.25 度	R/W	0

8.3.1.20 测温启动寄存器 TPS_START (0x150) (新增)

Offset = 0x150，复位值 0

比特位	名称	描述	读/写标志	复位值
31:1	---	预留	R	0
0	START	1: 启动 SAR 进行 TPS 测温，并且数字电路根据温度公式计算出温度值，输出到温度寄存器 3 TEMP_CAL； 0: 无效 注 1：该寄存器仅在用户温补模式 0 下有效，仅启动测温不进行温补，方便应用上获取温度进行软件平均。 注 2：测温启动后，需等 4.5ms 才能完成测温；需在启动测量 4.5ms 后查询 RTC->CTRL 的 bit10 (Cal_busy 位) (不能立即查询)，为 0 后获取温度寄存器 3 TEMP_CAL 值；查询超时时间，建议在 10ms 以上。	WO	0

8.3.1.21 温度寄存器 3 RTC_TEMP3 (0x154) (新增)

Offset = 0x154，复位值 0

比特位	名称	描述	读/写标志	复位值
31:12	---	预留	R	0
11:0	TEMP	温度寄存器 在用户温补模式 0 下，软件配置 TPS_START 信号启动测温后，等待 RTC_CTRL 寄存器中的 Cal_busy 为 0 后，可从该寄存器获取当次测温计算得到的温度值。 // Temp[11] Temp[10:4] Temp[3] Temp[2] Temp[1] Temp[0]	R	0

		// 符号 -128 度~127 度 0.5 度 0.25 度 0.125 度 0.0625 度 表示范围: -128 度~+127.75 度 注 1: 测温启动后, 需等 4.5ms 才能完成测温; 需在启动测量 4.5ms 后, 查询 RTC->CTRL 的 bit10 (Cal_busy 位) (不能立即查询), 为 0 后获取温度寄存器 RTC_TEMP3 值; 查询超时时间, 建议在 10ms 以上。 注 2: 其他温补模式下的硬件测温结果也会输出到该寄存器。		
--	--	---	--	--

8.3.1.22 二次补偿密码寄存器 RTC_CALPS(0xCC) (新增)

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:0	CALPS	RTC 二次补偿寄存器写保护, 写入 8'hA8, T0~T9 寄存器才会起作用。	R/W	0

8.3.1.23 二次分段补偿寄存器组 RTC_CAL_T0~T9(0xD0~0xF4) (新增)

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:0	CAL_T0~T9	T0~T9 是 8bit 寄存器, 在硬件自动温补基础上对 RTC 的误差做二次补偿, 刻度是 0.25ppm; T0 补偿温度范围: $T < -30^{\circ}\text{C}$ T1 补偿温度范围: $-30^{\circ}\text{C} \leq T < -20^{\circ}\text{C}$ T2 补偿温度范围: $-20^{\circ}\text{C} \leq T < -10^{\circ}\text{C}$ T3 补偿温度范围: $-10^{\circ}\text{C} \leq T < 0^{\circ}\text{C}$ T4 补偿温度范围: $0^{\circ}\text{C} \leq T \leq 10^{\circ}\text{C}$ T5 补偿温度范围: $35^{\circ}\text{C} < T \leq 45^{\circ}\text{C}$ T6 补偿温度范围: $45^{\circ}\text{C} < T \leq 55^{\circ}\text{C}$ T7 补偿温度范围: $55^{\circ}\text{C} < T \leq 65^{\circ}\text{C}$ T8 补偿温度范围: $65^{\circ}\text{C} < T \leq 75^{\circ}\text{C}$ T9 补偿温度范围: $T > 75^{\circ}\text{C}$	R/W	0

8.3.2 二次曲线补偿寄存器组

8.3.2.1 RTC_PS (0x40)

补偿寄存器密码保护 偏移地址= 0x40

比特位	名称	描述	读/写标志	复位值
31:08	---	预留	R	0
07:00	RTC_PS	当 RTC_PS=8'h8E 时, 补偿寄存器 44H~60H、70H、80H、130H~144H、150H 才可写; 当 RTC_PS=其他值时, 补偿寄存器 44H~60H、70H、80H、130H~144H、150H 不可写。	R/W	0

8.3.2.2 RTC_MODE (0x44) (boot 配置)

温度补偿模式寄存器 偏移地址= 0x44

比特位	名称	描述	读/写标志	复位值
-----	----	----	-------	-----

31:01	---	预留	R	0
00	PLL_MODE	RTC 校正方式（秒脉冲产生方案） 0： 保留 1： 纯数字校正方式（PLL 的 1Hz 仅用于校正，RTC 万年历使用 1S/30S/120S 补偿方式） 无实际意义，仅控制 FOUT 输出。	R/W	1

注：只有上下电复位起作用。Bootrom 会修改该值。

8.3.2.3 RTC_DOTA0 (0x48) (boot 配置)

初始频率偏差校正寄存器 偏移地址= 0x48

比特位	名称	描述	读/写标志	复位值
31:13	---	预留	R	0
12:0	DOTA0	校正值 DOTA0 为有符号二进制补码形式，bit12 为符号位： $DOTA0 = \text{round}(\sigma_0 * 32)$ ，其中 σ_0 为初始频率偏差，单位为 ppm，可表示范围：-128ppm ~ 127.96875ppm； DOTA0 分辨度为 0.03125ppm。 例：初始频率偏差 $\sigma_0 = -6\text{ppm}$ ，校正值 $DOTA0 = 0x1f40$ 。 注：大部分 32768 晶振，频率偏差典型值为 $\pm 20\text{ppm}$ ，最大值为 $\pm 30\text{ppm}$	R/W	0

注：只有上下电复位，会将该寄存器恢复到默认值，其它复位不会恢复。

用户对 RTC 的初始频率偏差进行校准时，建议用户调用锐能微库函数 “void RtcWriteDota(uint16_t dota);”，该库函数除了具备对 DOTA0 寄存器校准功能，还具备对 vbat 域配置寄存器做有效管理的功能，可以确保 Vbat 域相关参数被正确写入，这对于 Vbat 上电比 Vcc 慢的应用尤其重要。

8.3.2.4 RTC_ALPHAL (0x4C) (boot 配置)

晶振低温部分温度系数寄存器 偏移地址= 0x4C

比特位	名称	描述	读/写标志	复位值
31:11	---	预留	R	0
10:00	ALPHAL	晶振频率温度系数的低温部分温度系数，即 -40℃ 到 T0 这一段的温度系数。（格式为无符号二进制） $ALPHAL = \text{round}(\alpha * 2048 * 16)$ （单位为 ppm/℃ ² ，一般温度系数为 -0.034ppm/℃ ² ）（ α 可表示范围：0 - 0.0625 ppm/℃ ² ）	R/W	0

只有上下电复位能起作用，bootrom 会修改该值。

$$0.034 * 2048 * 16 * (T - T_0)^2 = 0.034 * 2048 * 16 * (-40 - 25)^2 = 4707123.2 \quad \text{约为 } 143.65\text{ppm}$$

$$\text{round}(0.034 * 2048 * 16 * (T - T_0)^2) = \text{round}(0.034 * 2048 * 16 * (-40 - 25)^2) = 1114 * 4225 = 4706650 \quad \text{约为 } 143.635\text{ppm}$$

误差为 0.02ppm

8.3.2.5 RTC_ALPHAH (0x50)

晶振高温部分温度系数寄存器 偏移地址= 0x50

比特位	名称	描述	读/写标志	复位值
-----	----	----	-------	-----

31:11	---	预留	R	0
10:00	ALPHAH	晶振频率温度系数的高温部分温度系数，即 T0 到 85℃ 这一段的温度系数。（格式为无符号二进制） ALPHAH = round($\alpha * 2048 * 16$)（单位为 ppm/℃ ² ，一般温度系数为 -0.034ppm/℃ ² ）（ α 可表示范围：0 – 0.0625 ppm/℃ ² ）	R/W	0

只有上下电复位能起作用，bootrom 会修改该值。

8.3.2.6 RTC_XT0 (0x54)

晶振温度曲线顶点温度寄存器 偏移地址= 0x54

比特位	名称	描述	读/写标志	复位值
31:07	---	预留	R	0
06:00	XT0	晶振温度曲线（二次曲线）顶点温度值（格式为无符号数） XT0 = round($T_0 * 4$)（ T_0 单位为℃）（ T_0 可表示范围：0 – 31.5℃） （分辨度为 0.25 度。实际上应该写入 25±5 的值。）	R/W	-

8.3.2.7 RTC_TADJ (0x58)

量产温度单点校正寄存器 偏移地址= 0x58

比特位	名称	描述	读/写标志	复位值
31:10	---	预留	R	0
10:0	TADJ	量产温度单点校正参数（格式为无符号二进制） 量产平台根据 $TADJ = \frac{T_0 + 273}{D_0 + 1021}$ 算出校正参数。 DATA = round($2^{11} * TADJ * 4$)（TADJ 单位为℃/LSB）（TADJ 可表示范围：0 – 0.249。 温度曲线的斜率 k 在 0.1859 – 0.2055 时对应于 0℃ 温度偏差为 260-290k）（1℃ 的变化大约为 1ppm。TADJ 的精度为 0.0001，能保证温度计算结果的精度为 0.125℃）	R/W	-

8.3.2.8 RTC_ZT (0x60)

零度设置寄存器、ADC offset 校正寄存器 偏移地址= 0x60

比特位	名称	描述	读/写标志	复位值
31:15	---	预留	R	-
14:09	ADCOS[5:0]	该寄存器对计算温度的 ADC 采样值做 offset 校正，ADC 采样值为 10bit，ADCOS 与其低位对齐相加，ADCOS[5]是符号位。	R/W	-
08:0	ZT	为设计预留 option，一般情况下，该寄存器应该写入 273（十进制）。	R/W	-

8.3.3 四次曲线补偿寄存器组

四次曲线补偿寄存器组寄存器配置，需要先配置 RTC_PS=8'h8E，写使能生效后才可以配置。

四次曲线补偿初始偏差校正同二次曲线补偿寄存器 RTC_DOTA0，应用需要注意。

8.3.3.1 RTC_MODE1 (0x130)

温补补偿曲线选择寄存器 偏移地址= 0x130

比特位	名称	描述	读 / 写标志	复位值
31:16	---	预留	R	0
15:0	RTC_MODE1	=16'hEA65: 启用四次曲线补偿，使用 0x134~0x144 寄存器 = 其他值：使用二次曲线补偿，起作用的寄存器是 0x4C/0x50/0x54。	R/W	0

8.3.3.2 RTC_XT1 (0x134)

晶振温度曲线顶点温度寄存器 1 偏移地址= 0x134

比特位	名称	描述	读/写标志	复位值
31:13	---	预留	R	0
12:0	XT1	晶振温度曲线（四次曲线）顶点温度值（格式为无符号数） $XT1 = \text{round}(T_0 * 256)$ (T_0 单位为 $^{\circ}\text{C}$) (T_0 可表示范围：0 – 31.99609375 $^{\circ}\text{C}$)（分辨度为 0.00390625 度）。 典型值为 25 $^{\circ}\text{C}$ 。	R/W	-

8.3.3.3 RTC_ALPHA (0x138)

晶振一次温度系数寄存器 偏移地址= 0x138

比特位	名称	描述	读/写标志	复位值
31:16	---	预留	R	0
15:0	ALPHA	晶振频率温度系数的一次系数 α ，典型值为 -0.017ppm/ $^{\circ}\text{C}$ ，最大 0.012，最小 -0.052。（格式为有符号二进制，注意一次系数可能为正，也可能为负。单位为 ppm/ $^{\circ}\text{C}$ ） 如果 α 为正，那么： $ALPHA = \text{round}(\alpha * 2^{15*8})$ 如果 α 为负，那么： $ALPHA = \text{round}(2^{16} + \alpha * 2^{15*8})$ α 用 16 位有符号数表示，归一化值放大了 8 倍，可以表示的范围是： $\pm 0 \sim 0.125\text{ppm}/^{\circ}\text{C}$ 一个 LSB 代表： $1/2^{18} = 3.8 * 10^{-6} \text{ppm}/^{\circ}\text{C}$ ， 量化误差最大影响为： $LSB * (40+25) = 0.00025\text{ppm}$ ，可以忽略。	R/W	0

8.3.3.4 RTC_BETA (0x13C)

晶振二次温度系数寄存器 偏移地址= 0x13C

比特位	名称	描述	读/写标志	复位值
31:15	---	预留	R	0

14:0	BETA	<p>晶振频率温度系数的二次系数 β，典型值为-0.0334ppm/°C²，最大-0.0332，最小-0.0336。（二次系数都为负值，格式定义为无符号二进制。单位为 ppm/°C²）</p> <p>β 正常都为负，寄存器应该写入：</p> <p>BETA = round($-\beta * 2^{15*16}$)</p> <p>β 用 16 位无符号数表示，归一化值放大了 16 倍，可以表示的范围是：0~ -0.0625 ppm/°C，注意 bit15 只能为 0。</p> <p>一个 LSB 代表：1/2¹⁹=1.9073*10⁻⁶ ppm/°C， 量化误差最大影响为：LSB*(40+25)²=0.00806ppm，可以忽略。</p>	R/W	0
------	------	--	-----	---

8.3.3.5 RTC_GAMMA (0x140)

晶振三次温度系数寄存器 偏移地址= 0x140

比特位	名称	描述	读/写标志	复位值
31:16	---	预留	R	0
15:0	GAMMA	<p>晶振频率温度系数的三次系数 γ，精工晶振参数表典型值为 -4.9*10⁻⁵ppm/°C³，最大 -4.4*10⁻⁵ppm/°C³，最小 -5.210⁻⁵ppm/°C³。实测值为-6.3*10⁻⁵ppm/°C³。（格式为仍然使用有符号二进制表示）</p> <p>γ 正常都为负，那么寄存器应该写入：</p> <p>GAMMA = round($2^{16} + \gamma * 2^{15*2048*4}$)</p> <p>$\gamma$ 用 16 位有符号数表示，归一化值放大了 8192 倍，可以表示的范围是：±0~12.207⁻⁵ ppm/°C</p> <p>一个 LSB 代表：1/2²⁸=0.37253*10⁻⁸ ppm/°C， 量化误差最大影响为：LSB*(40+25)³=0.001023ppm，可以忽略。</p>	R/W	0

8.3.3.6 RTC_ZETA (0x144)

晶振四次温度系数寄存器 偏移地址= 0x144

比特位	名称	描述	读/写标志	复位值
31:16	---	预留	R	0
15:0	ZETA	<p>晶振频率温度系数的四次系数 ζ，实测值为 4.692437*10⁻⁷ppm/°C⁴。精工晶振参数表无此参数。（格式为仍然使用有符号二进制表示）</p> <p>ζ 如果为正，那么寄存器应该写入：</p> <p>ZETA = round($\zeta * 2^{15*65536*8}$)</p> <p>ζ 如果为负，那么寄存器应该写入：</p>	R/W	0

		$ZETA = \text{round}(2^{16} + \zeta * 2^{15} * 65536 * 8)$ <p> ζ 用 16 位有符号数表示, 归一化值放大了 65536*8 倍, 可以表示的范围是: $\pm 0 \sim 1.9075 \times 10^{-6} \text{ ppm}/^{\circ}\text{C}$。注意: 实际上四次曲线不必有这么大的表示范围, 填入过大的值会造成 DotaT 寄存器溢出, 并无实际意义。 一个 LSB 代表: $1/2^{34} = 5.8208 \times 10^{-11} \text{ ppm}/^{\circ}\text{C}$, 量化误差最大影响为: $LSB * (40+25)^4 = 0.001039 \text{ ppm}$, 可以忽略。 </p>		
--	--	---	--	--

注意: 参考资料所列参数推算为 1.22×10^{-8}

8.3.4

结果输出寄存器组

8.3.4.1 RTC_DOTAT (0x64)

补偿总量寄存器 偏移地址= 0x64

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
15:0	DOTAT	<p>数据格式:</p> <p>15 14~5 4:0</p> <p>符号 ppm 0.03125ppm</p> <p>表示范围: 1024ppm ~ -1023.96785ppm</p> <p>实际表示范围:</p> <p>Dotat 实际范围一般在 128ppm 以内。</p> <p>二次曲线计算原理:</p> $\text{dotat}[15:0]/32 = \text{dota0}[12:0]/32 + \alpha[10:0]/(2048*16) * (\text{temp}[9:0]/4 - \{3^{\text{h0}}, \text{xt0}[6:0]\}/4)^2$ <p>// dota0[12:0]=round[σ_0*32] 范围: -128ppm ~127.96875ppm // alpha[10:0]=round[α *2048*16] 范围: 0~0.034~0.0625 // temp[9:0]=T*4 范围: -128 度~+127.75 度 // xt0=T0*4 范围: 0~31.5℃ // Dota0 一般为 20~30ppm, alpha 为 0.034 左右, temp 范围是-40~85 度, xt0 一般为 25 度左右。</p> <p>四次曲线计算原理:</p> $\begin{aligned} \text{dotat}[15:0]/32 = & \text{dota0}[12:0]/32 + \\ & \alpha[15:0]/(2^{15}*8) * (\text{temp}[9:0]/4 - \text{xt1}[12:0]/256) + \\ & \beta[14:0]/(-2^{15}*16) * (\text{temp}[9:0]/4 - \text{xt1}[12:0]/256)^2 + \\ & \gamma[15:0]/(2^{15}*2048*4) * (\text{temp}[9:0]/4 - \{\text{xt1}[12:0]/256\})^3 + \\ & \zeta[15:0]/(2^{15}*65536*8) * (\text{temp}[9:0]/4 - \{\text{xt1}[12:0]/256\})^4 \end{aligned}$ <p>// dota0[12:0]=round[σ_0*32] 范围: -128ppm ~127.96875ppm // alpha[15:0]=round[α *2^{15}*8] 范围: $\pm 0 \sim 0.125 \text{ ppm}/^{\circ}\text{C}$</p>	R	-

		// beta[14:0]=round($-\beta * 2^{15} * 16$) 范围: 0~-0.0625 ppm/°C // gamma[15:0]=round($2^{16} + \gamma * 2^{15} * 2048 * 4$) 范围: $\pm 0 \sim 12.207 \times 10^{-5}$ ppm/°C // zeta[15:0]=round($2^{16} + \zeta * 2^{15} * 65536 * 8$) 范围: $\pm 0 \sim 1.9075 \times 10^{-6}$ ppm/°C // temp[9:0]=T*4 范围: -128 度~+127.75 度 // xt1=T0*256 范围: 0 ~ 31.99609375°C		
--	--	---	--	--

8.3.4.2 RTC_FPTR (0x74)

PLL 补偿寄存器 偏移地址= 0x74

比特位	名称	描述	读/写标志	复位值
31:16	---	预留	R	0
15:0	FPTR	PLL 补偿寄存器 $FPTR = \text{fix}(\text{dotat} / 0.030518 \text{ ppm})$ 有符号数, 代表 PLL 模式时每 1S 需要补偿的时钟周期数。	R	0

8.3.4.3 RTC_FDTR1S (0x78)

每 1S 补偿寄存器 偏移地址= 0x78

比特位	名称	描述	读/写标志	复位值
31:6	---	预留	R	0
5:0	FDTR1S	RTC 每 1S 补偿寄存器 $FDTR1S = \text{fix}(\text{dotat} / 30.5175)$ 有符号数, 代表 RTC 秒脉冲每 1S 需要补偿的时钟周期数。	R	0

8.3.4.4 RTC_FDTR30S (0x7C)

每 30S 补偿寄存器 偏移地址= 0x7c

比特位	名称	描述	读 / 写 标志	复位值
31:7	---	预留	R	0
6:0	FDTR30S	RTC 每 30S 补偿寄存器 RTC 每隔 30s 时的数字补偿量: 1S 补偿的余量 $A = (\text{dotat} - FDTR1S * 30.5175)$ 每隔 30S 需要多补偿的量 $B = \text{fix}(A / 1.017257)$ 每隔 30S 补偿寄存器的值 $FDTR30S = FDTR1S + B$ 举例: 如果 $\text{dotat} = 50 \text{ ppm}$, 那么 $FDTR1S = 1$, 1S 补偿的余量 $A = (50 - \text{fix}(50 / 30.5)) * 30.5 = 19.5$ 每隔 30S 需要多补偿的量 $B = \text{fix}(19.5 / 1.01725) = 19$ 每隔 30S 补偿寄存器的值 $FDTR30S = FDTR1S + 19 = 20$	R	0

		如果 $\text{dotat} = -50\text{ppm}$ ，那么 $\text{FDTR1S} = -1$ ， 1S 补偿的余量 $A = (-50 - \text{fix}(-50/30.5) * 30.5) = -19.5$ 每隔 30S 需要多补偿的量 $B = \text{fix}(-19.5/1.01725) = -19$ 每隔 30S 补偿寄存器的值 $\text{FDTR30S} = \text{FDTR1S} - 19 = -20$ 有符号数，代表 RTC 秒脉冲每 30S 需要补偿的时钟周期数。		
--	--	--	--	--

8.3.4.5 RTC_FDTR120S (0x84)

每 120S 补偿寄存器 偏移地址 = $0x84$

比特位	名称	描述	读/写标志	复位值
31:9	---	预留	R	0
8:0	RTC_FDTR120S	RTC 每 120S 补偿寄存器 每隔 120s 时的数字补偿量： 1S 和 30S 补偿的余量 $A = (\text{dotat} - \text{FDTR1S} * 30.5175 - (\text{FDTR30S} - \text{FDTR1S}) * 1.01725)$ 每隔 120S 需要多补偿的量 $B = \text{fix}(A/0.254313)$ 每隔 120S 补偿寄存器的值 $\text{FDTR120S} = \text{FDTR30S} + B$ 有符号数，代表 RTC 每 120S 需要补偿的时钟周期数	R	-

8.4 RTC 时钟读写步骤

- 1、将系统控制章节中模块使能 1 寄存器 MOD1_EN 第 10 位 RTC_EN 设置为 1。
- 2、时钟的读取：读取 RTC 的秒、分、时等时间寄存器。
- 3、时钟的写入：

将 RTC_CTL 的第 8 位 WRTC 置为 1，打开写使能操作。

按照“年月日时分秒”顺序写，当写入秒寄存器后时间开始从写入时刻起累计，注意此种方法存在秒写入之前分钟翻转的可能，所以写入后要读出做校验；

或按照“秒分时分年月日”顺序写，先写秒寄存器，万年历计数器被清零，只要在一秒内写入其他值，可以保证成功写入。

注意硬件对“年月日”有做合法性判断，不能按照“日月年”顺序写，只能连续写入“年月日”。

为增强软件可靠性，写入后都建议读出做确认。

8.5 RTC 校准步骤

用户只需要对 32.768KHz 的初始偏差进行校准即可。误差通过锐能微编程器界面或者使用库函数写入。

8.6 RTC 定时器操作步骤

以定时器 1 产生 1S 中断为例，操作步骤为：

- 1、将系统控制章节中模块使能 1 寄存器 MOD1_EN 第 10 位 RTC_EN 设置为 1 时钟启动。
- 2、将 RTC_CTL 的第 8 位 WRTC 置为 1，打开写使能操作。
- 3、设置 $\text{RTC} \rightarrow \text{CNT1} = 0x00$ ；即为 1S 产生 1 次中断。
- 4、设置 $\text{RTC} \rightarrow \text{IE} = 0x02$ ；RTC 定时器 1 中断使能。
- 5、开启 RTC 中断使能，NVIC_EnableIRQ(RTC_IRQn);

6、编写中断服务程序：

```
void RTC_HANDLER(void)
{
    if(RTC->IF&0x02)        // 定时 1
    {
        /* Start adding user code. Do not edit comment generated here */
    }
}
```

7、配置完成后既可产生 1S 中断。

9 WDT

SoC 内置硬件看门狗，用于检测程序的异常执行。

9.1 概述

看门狗具备如下特点：

- ◎ 溢出时间可设置为：16ms、32ms、128ms、512ms、1s、2s、4s、8s；
- ◎ 喂狗窗口期可设置：

出现以下任何一种情况时产生看门狗复位：

- ◎ 看门狗定时器计数器溢出；
- ◎ 将 0xBB 以外的数据写入 WDT_EN；
- ◎ 在喂狗窗口关闭期间将数据写入 WDT_EN；
- ◎ 通过 bitband 空间将数据写入 WDT_EN；

9.2 看门狗定时器的配置

RN821X 的 WDT 为硬件看门狗，不能通过寄存器直接进行配置，需要通过设置“选项字节”的方式对其进行配置。看门狗的配置有间隔中断，窗口打开周期，溢出时间，CPU 睡眠设置，CPU 调试设置等选项。

名称	描述	厂家默认值
间隔中断	0: Disable（不使能间隔中断） 1: Enable（达到溢出事件的 75%时产生间隔中断）	0
窗口打开周期	0: 25% 1: 50% 2: 75% 3: 100% 在窗口打开期间将 0xBB 写入 WDTE 寄存器，看门狗清零并重新计数； 在窗口关闭期间将 0xBB 写入 WDTE 寄存器，会产生内部复位信号。	3
溢出时间	0: 16ms 1: 32ms 2: 128ms 3: 512ms 4: 1s 5: 2s 6: 4s 7: 8s	4
CPU 睡眠设置	0: Disable（当 CPU 处于 sleep 或者 deepsleep 的时候不开启 WDT） 1: Enable（当 CPU 处于 sleep 或者 deepsleep 的时候开启 WDT）	0
CPU 调试设置	0: Disable（当 CPU 处于调试状态时不开启 WDT） 1: Enable（当 CPU 处于调试状态时开启 WDT） 注：CPU 处于调试状态指的是用户通过调试接口将 Cortex M0 停住（PC 指针停止计数）。如果芯片处于开发过程中，不建议使能该设置。因为如果使能该设置，当芯片处于调试状态时 WDT 仍然会计数，溢出时会产生中断，将引起调试无法进行。	0

窗口打开周期的定义如下图所示，以 25%的窗口打开周期为示例：



9.3 寄存器描述

WDT 寄存器基址

模块名	物理地址	映射地址
WDT	0x40030000	0x40030000

WDT 寄存器偏移地址

寄存器名	地址偏移量	描述
WDT_EN	0x0	使能寄存器
WDT_CTRL	Offset+0x4	WDT 配置寄存器
WDT_PASS	Offset+0x8	WDT 密码寄存器
WDT_HALT	Offset+0x14	控制 CPU HALT 时 WDT 是否停止计数
WDT_STBY	Offset+0x18	控制 CPU SLEEP 时 WDT 是否停止计数

9.3.1 WDT_EN (0x0)

比特位	名称	描述	读/写	复位值
31:9	---	只读，不可写	R	0
8	WR_BUSY	WDT 忙 喂狗与 BUSY 位无关。	R	0
7:0	WDTE	写入 0xBB 对看门狗定时器清零并再次开始计数操作。 复位信号的产生将该寄存器设置为 0x55	R/W	55

9.3.2 WDT_CTRL (0x04)

看门狗配置寄存器

Offset=0x04

比特位	名称	描述	读/写	复位值
31:8	---	预留	R	0
7	WDTINT	看门狗定时器间隔中断 =0 不使用间隔中断 =1 达到溢出事件的 75%时产生间隔中断。	R/W	0
6:5	WINDOW	窗口打开周期： 00 25% 01 50% 10 75% 11 100% 如果在窗口打开期间将 BBH 写入 WDTE，则看门狗清零并重新计数； 如果在窗口关闭期间将 BBH 写入 WDTE，会产生内部复位信号。	R/W	11

4	--	保留	R	0
3:1	WDCS	看门狗定时器的溢出时间： 000 16ms 001 32ms 010 128ms 011 512ms 100 1s 101 2s 110 4s 111 8s	R/W	100
0	--	保留	R	0

备注：该寄存器存放在选项字节中,复位后芯片 boot 自动加载选项字节中配置值,选项字节内容可通过锐能微编程器修改。

9.3.3

WDT_PASS (0x08)

看门狗密码寄存器

Offset=0x08

位	名称	描述	读/写	复位值
31:8	---	预留	R	0
7:4	PASS	密码为 0x7 Note: 只有在该密码正确的情况下, WDT_PASS[3:0]才可以写入	R/W	0
3:0	PASS	密码为 0xE Note: 只有在 PASS[7:0]=0x7E 的情况下, WDT_CTRL 才可以写入 配置 WDT_PASS 的步骤: WDT->PASS = 0x70; WDT->PASS = 0x7e;	R/W	0

9.3.4

WDT_HALT (0x14)

看门狗 HALT 控制寄存器

Offset=0x14

位	名称	描述	读/写	复位值
31:8	---	预留	R	0
15:0	HALT_EN	当 CPU 处于 HALT 时 (调试时使用), 如果 HALT_EN=16'h7e5a, 那么暂停 WDT 计数; 如果 HALT_EN=其他值, 那么不暂停 WDT 计数;	R/W	0x7e5a

备注:

1. 默认复位后使能 HALT 功能;
2. 设立该寄存器的目的是降低上电复位不成功情况下将 WDT 关闭的概率, 使 WDT 能够将系统复位;
3. 该寄存器受密码保护位保护;
4. 该寄存器存放在选项字节中, 复位后芯片 boot 自动加载选项字节中配置值, 选项字节内容可通过锐能微编程器修改;

9.3.5

WDT_STBY (0x18)

看门狗 STBY 控制寄存器

Offset=0x18

位	名称	描述	读/写	复位值
31:8	---	预留	R	0
15:0	STBY_EN	当 CPU 处于 sleep 或者 deepSleep 时 (睡眠时使用), 如果 STBY_EN=16'h a538, 那么当 CPU 睡眠时关闭 WDT; 如果 STBY_EN=其他值, 那么当 CPU 睡眠时不关闭 WDT;	R/W	0xA538

备注：

1. 默认复位后使能 STBY 功能；
2. 设立该寄存器的目的是降低上电复位不成功情况下将 WDT 关闭的概率，使 WDT 能够将系统复位；
3. 该寄存器受密码保护位保护；
4. 该寄存器存放在选项字节中，复位后芯片 boot 自动加载选项字节中配置值，选项字节内容可通过锐能微编程器修改；

9.4 WDT 操作步骤

- 1、配置系统控制章节模块使能 1 寄存器 MOD1_EN 第 9 位为 1，打开 WDT APB 时钟。
- 2、WDT 默认配置为启动，定时器溢出时间为 1S，窗口打开周期为 75%。用户程序可不进行 WDT 初始化配置。
- 3、喂狗操作：WDT->EN = 0xbb;
- 4、WDT 默认为睡眠后关闭 WDT，MOD1_EN 中 WDT 时钟可不关闭，如关闭，需等待 WDT_EN 的第 8 位 WR_BUSY 为 0 后方能关闭 WDT 时钟。
- 5、当硬件仿真将程序停止运行时，WDT 计数也会暂停，不会影响硬件仿真。
- 6、完成。

使用建议：

由于 RN821x 的 WDT 功耗极低，在 CPU 休眠时开启 WDT 增加的额外功耗小于 1uA，从系统更高的可靠性角度考虑，建议客户在 CPU 休眠时开启 WDT，可使用 RTC 中的秒定时器唤醒 CPU 进行喂狗操作。

10 LCD

SoC内置段码式LCD控制器。

10.1 概述

LCD 控制器具备如下特性：

- ◎ 最高支持 4x40、6x38、8x36LCD 驱动模式；
- ◎ 支持 A 类和 B 类两种驱动波形；
- ◎ 支持 1/3 和 1/4 偏压比；
- ◎ 支持静态，1/2，1/3，1/4，1/6，1/8 占空比；
- ◎ 支持 16 级对比度驱动模式；
- ◎ 支持电荷泵和内部电阻串分压方式实现 LCD Bias 电压；

10.1.1 扫描时钟频率

LCD 波形扫描频率来自对 LOSC（频率为 32768Hz）的分频。分频系数通过寄存器 LCD_CLKDIV 配置。一般要求 LCD 屏的帧刷新频率略微大于 60Hz。表 8-1 中绿色标注的为正常使用的帧频。

表 8-1 LCD 扫描频率与帧频

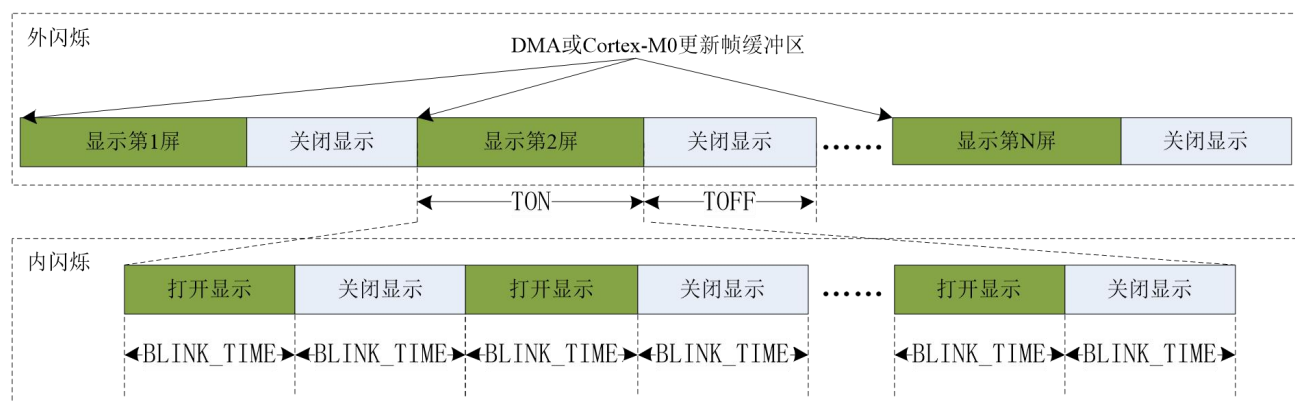
LCD_CLKDIV	扫描频率	静态占空比	1/2 占空比	1/3 占空比	1/4 占空比	1/6 占空比	1/8 占空比
0xff	64Hz	64Hz	32Hz	21.3Hz	16Hz	10.7Hz	8Hz
0x7f	128Hz	128Hz	64Hz	42.7Hz	32Hz	21.3Hz	16Hz
0x54	192.8Hz	192.8Hz	96.4Hz	64.3Hz	48.2Hz	32.1Hz	24.0Hz

0x3f	256Hz	256Hz	128Hz	85.3Hz	64Hz	42.7Hz	32Hz
0x2a	381.3Hz	381.3Hz	190.5Hz	127.0Hz	95.3Hz	63.5Hz	47.6Hz
0x1f	512Hz	512Hz	256Hz	170.7Hz	128Hz	85.3Hz	64Hz

10.1.2 闪烁模式

LCD 支持两种闪烁模式：内闪烁和外闪烁。两种模式可以同时使能。

图 8-1 LCD 闪烁方式



如图 8-1，LCD 使能后会根据 LCD_BLINK 寄存器的 TON 位域定义的时间长度打开显示，随后根据 LCD_BLINK 寄存器的 TON 位域定义的时间长度关闭显示。

在显示打开和关闭时，LCD 可以发出中断请求或 DMA 请求。用户可以使用这些事件更新帧缓冲区。

配合 DMA 的回绕特性，可以实现对多个帧缓冲区的自动轮显。

10.1.2.1 内闪烁模式

LCD 支持在由 LCD_BLINK 寄存器的 TON 位域指定长度的显示期间，插入闪烁模式。闪烁的间隔由 LCD_BLINK 寄存器的 BLINK_TIME 位域给出。当 BLINK_TIME 为 0 时，内闪烁模式被禁止；当 BLINK_TIME 不为 0 时，TON 必须为 BLINK_TIME 的偶数倍。

10.1.2.2 外闪烁模式

当 LCD_BLINK 寄存器中 TOFF 不为 0 时，闪烁功能被使能。Blink Mode 使能后，根据 LCD_BLINK 寄存器的 TON 和 TOFF 的值确定闪烁频率。

10.1.3 LCD 驱动波形

LCD 驱动波形与显示波形类型，占空比和偏压比有关。

显示波形类型 A 为行反转驱动方式，即在每个帧内完成一次正负驱动交替；显示波形类型 B 驱动为帧反转方式，即在每两个帧内完成一次正负驱动交替。当占空比较大时，采用显示波形类型 B 驱动方式显示效果会更优。

用户需要根据应用所需的 COM 数选择 LCD 输出波形的占空比：

- ◆ 1 个 COM：选择静态占空比，只使用 COM0；
- ◆ 2 个 COM：选择 1/2 占空比，使用 COM0，COM1；
- ◆ 3 个 COM：选择 1/3 占空比，使用 COM0 ~COM2；
- ◆ 4 个 COM：选择 1/4 占空比，使用 COM0 ~COM3；
- ◆ 6 个 COM：选择 1/6 占空比，使用 COM0 ~COM5；

- ◆ 8 个 COM: 选择 1/8 占空比, 使用 COM0 ~COM7;

10.1.3.1 类型 A 驱动波形

图 8-2 LCD 驱动波形 (1/4 Duty, 1/3 Bias, Type A)

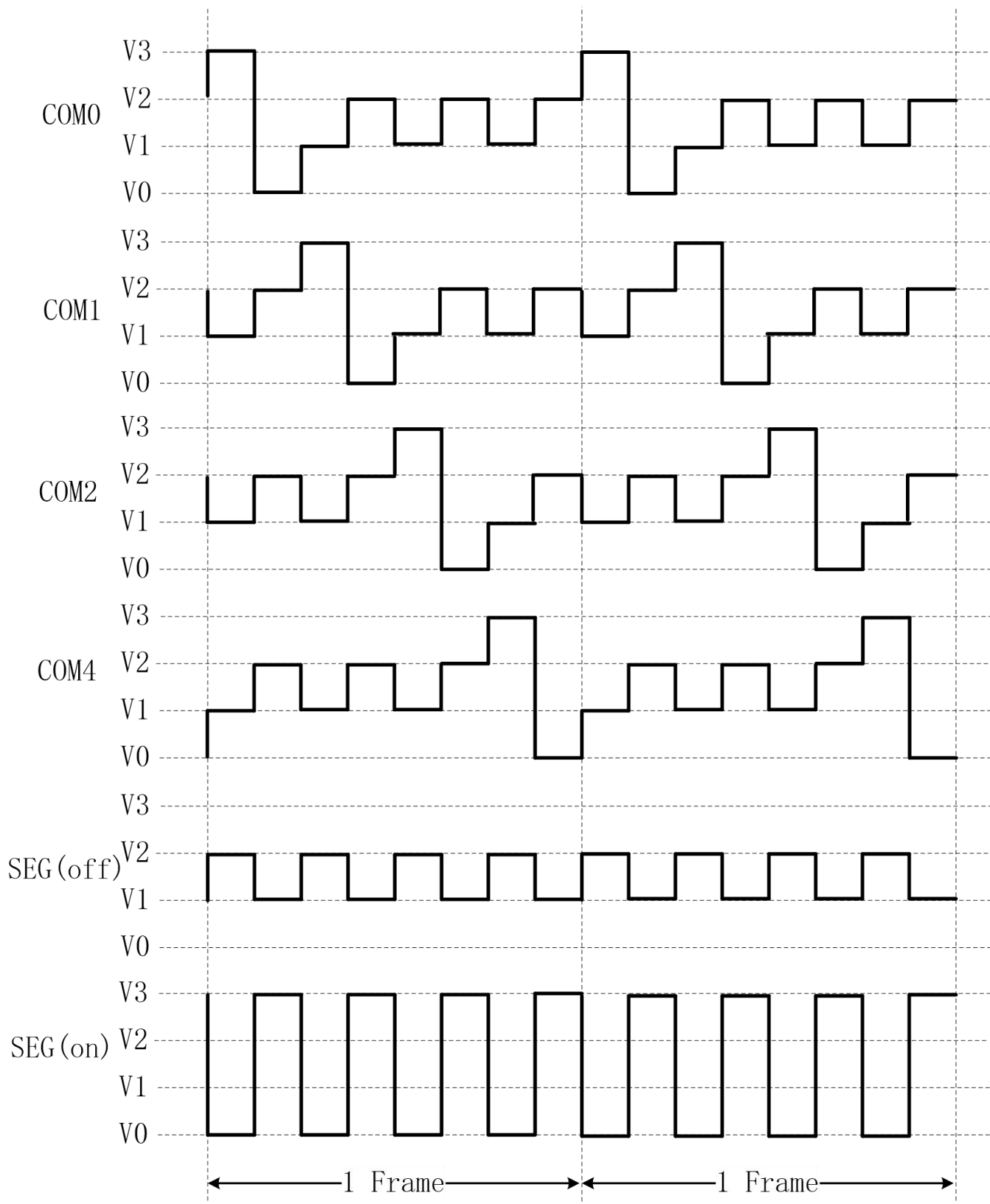
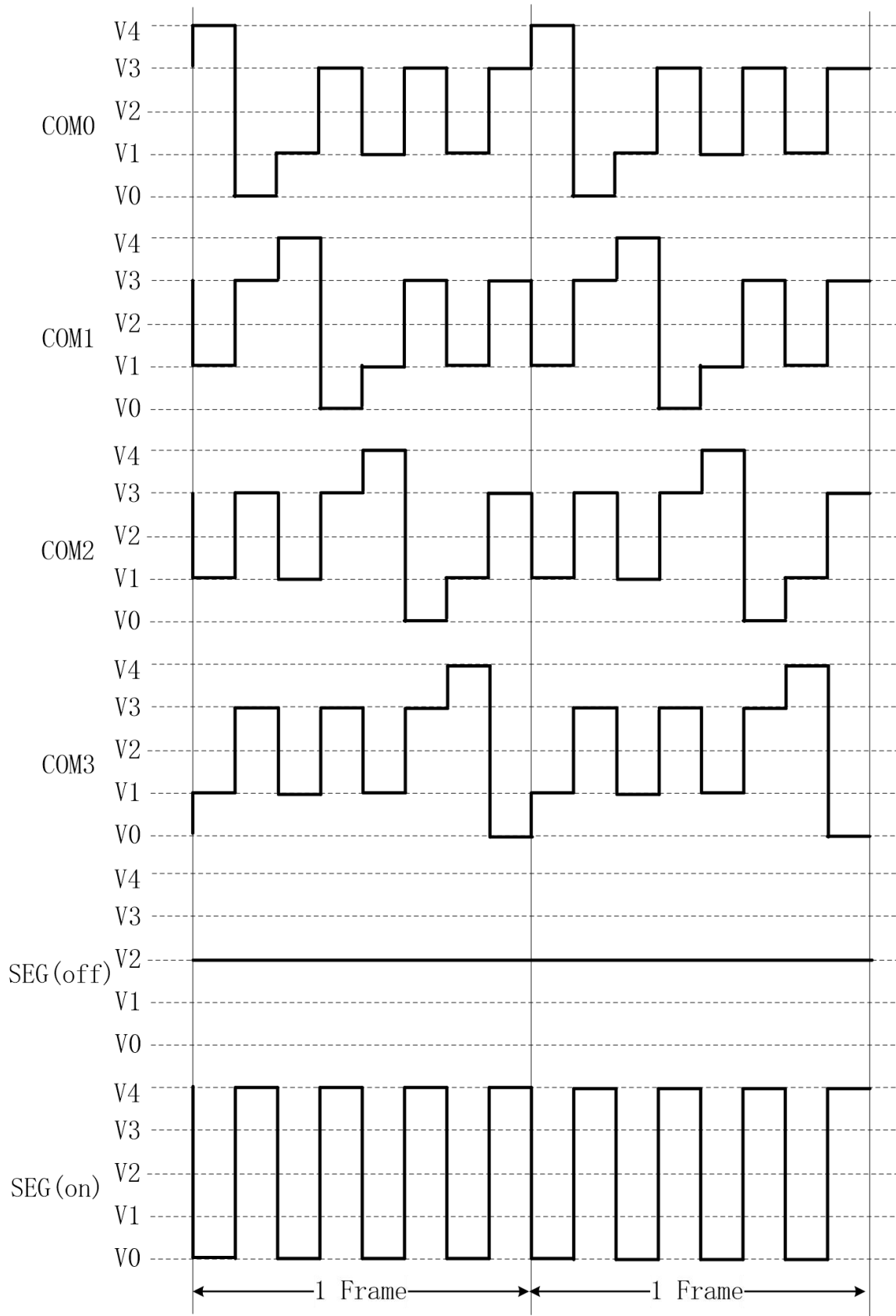


图 8-3 LCD 驱动波形 (1/4 Duty, 1/4 Bias, Type A)



10.1.3.2 类型 B 驱动波形

图 8-4 LCD 驱动波形 (1/4 Duty, 1/3 Bias, Type B)

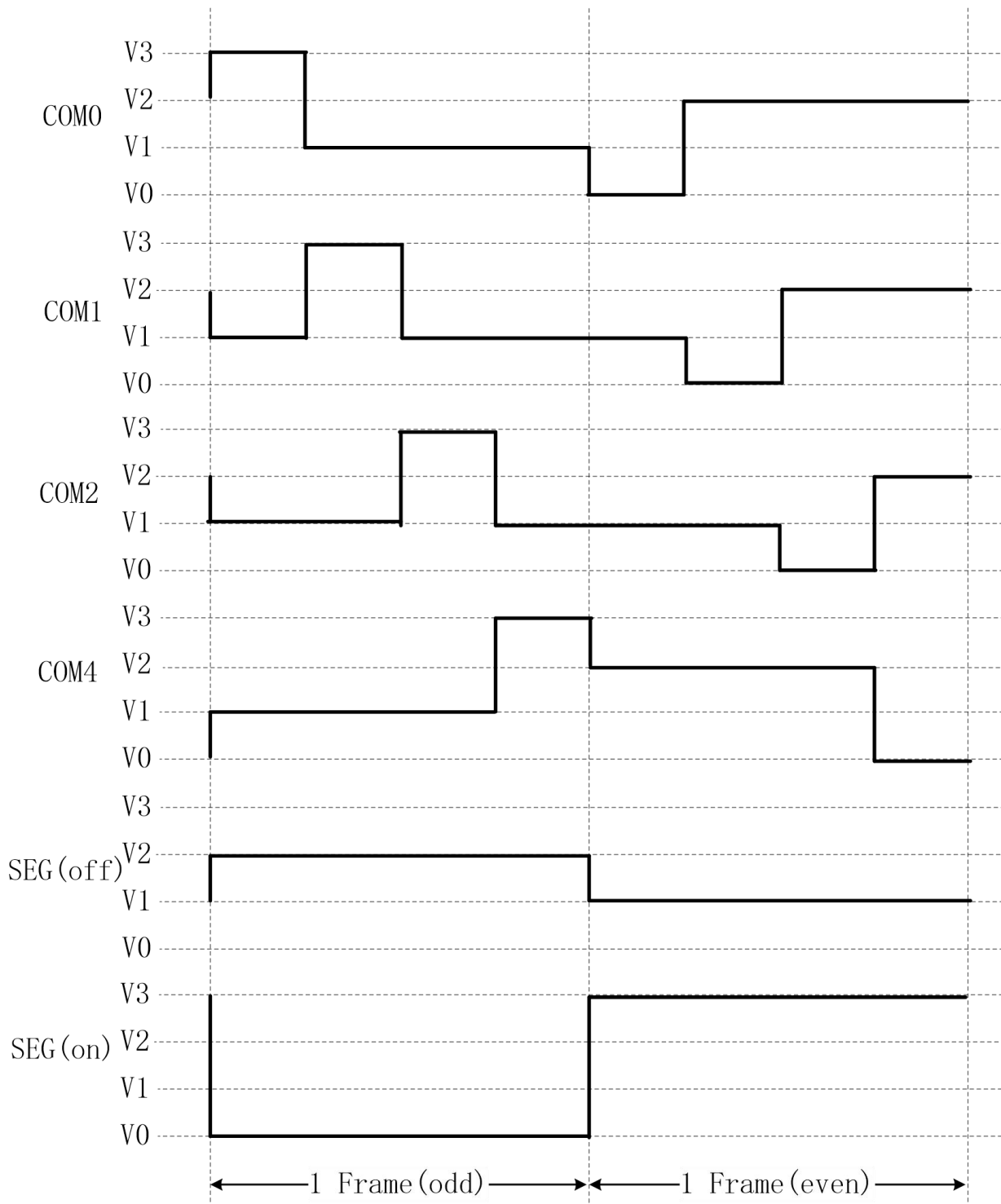
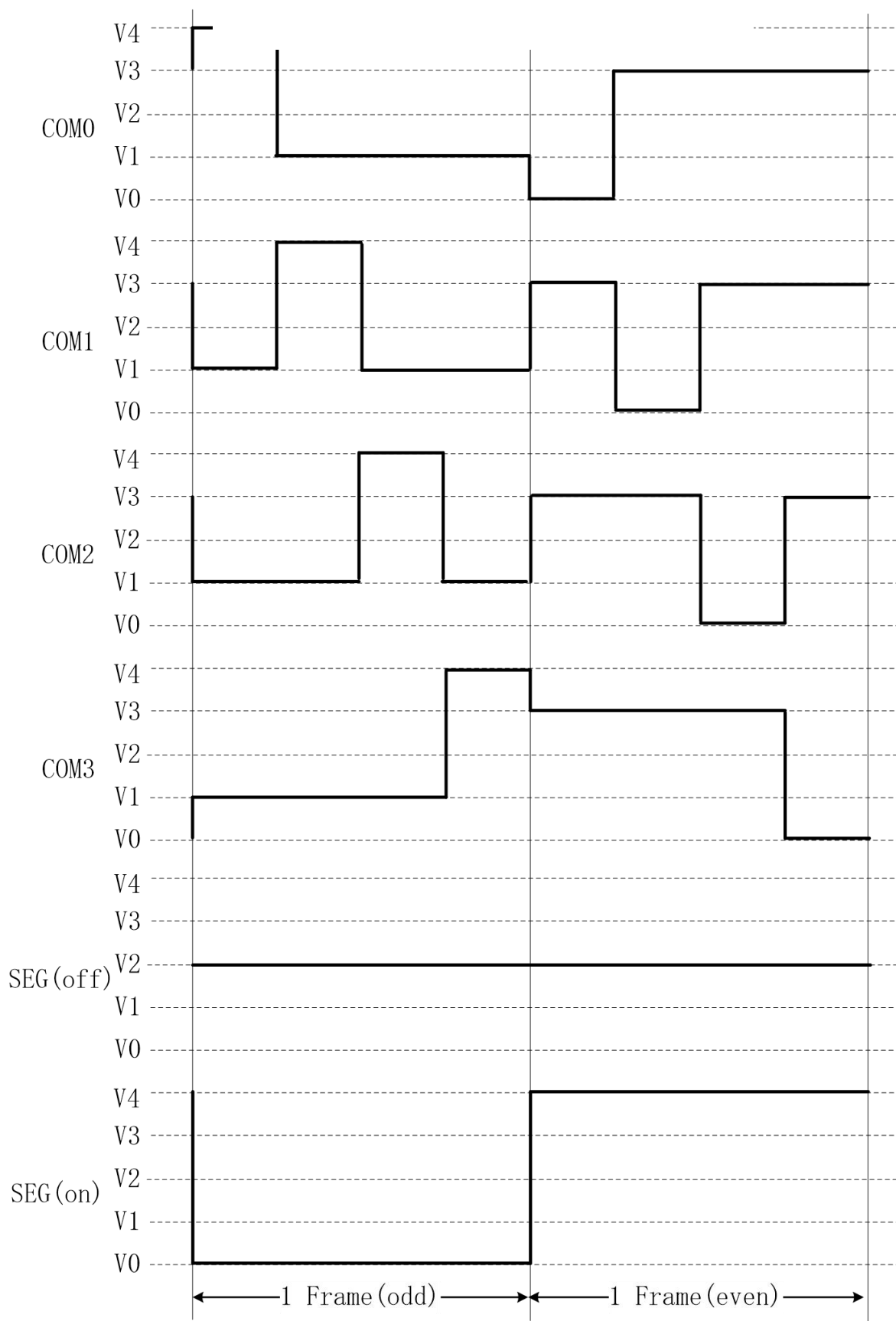


图 8-5 LCD 驱动波形 (1/4 Duty, 1/4 Bias, Type B)



10.1.4 LCD 偏置电压-电荷泵方式

LCD 的偏置电压可采用 Charge Pump 方式提供。Charge Pump 共需要产生 4 个电压 (V_a , V_b , V_c , V_d), 以满足 1/4 偏压比的应用。对于不同偏压比设置, Charge Pump 输出的电压模式有所不同, 如表 8-2 所示。

表 8-2 LCD 驱动电压与偏压比关系

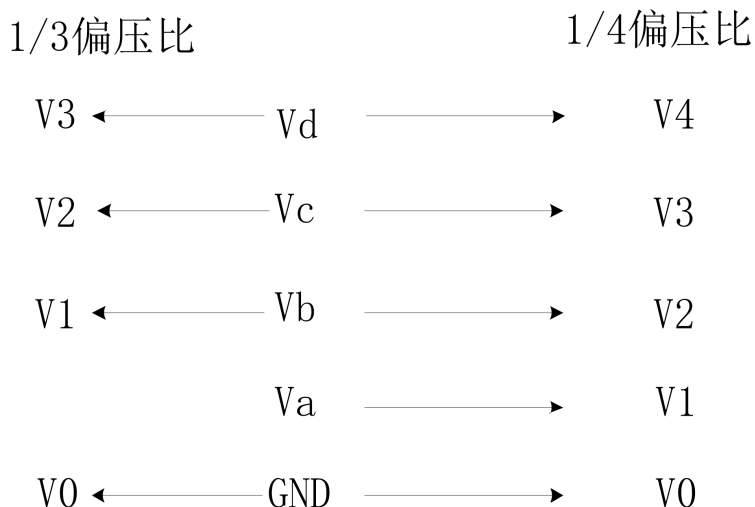
偏压比	灰度选择	V_a	V_b	V_c	V_d	V_d (MAX)
1/3 偏压比	BIASLVL[5] = 0	$V_a = V_{ref} * (32 + \text{BIASLVL}[4:0]) / 63$	$V_b = V_a$	$V_c = 2 * V_a$	$V_d = 3 * V_a$	3.75 V
	BIASLVL[5] = 1	$V_a = V_{ref} * (1 + \text{BIASLVL}[4:0] / 63)$	$V_b = V_a$	$V_c = 2 * V_a$	$V_d = 3 * V_a$	5.59 V
1/4 偏压比	BIASLVL[5] = 0	$V_a = V_{ref} * (32 + \text{BIASLVL}[4:0]) / 63$	$V_b = 2 * V_a$	$V_c = 3 * V_a$	$V_d = 4 * V_a$	5.0 V
	BIASLVL[5] = 1	$V_a = V_{ref} * (1 + \text{BIASLVL}[4:0] / 63)$	$V_b = 2 * V_a$	$V_c = 3 * V_a$	$V_d = 4 * V_a$	6.032V

LCD 所需的 V_d 的最大值为 5.2V。当选择为 1/4 偏压比时, 当 BIASLVL[5:0] 设置大于 6'h2d 时, LCD 控制器自动把 BIASLVL[5:0] 钳位到 6'h2d。

LCD 默认选择 LBGR 作为基准, LBGR 输出典型值为 1.27V。

1/3 和 1/4 偏压比应用电压选择如图 8-6 所示:

图 8-6 偏置电压选择



10.1.5 LCD 偏置电压-电阻串方式

内置 LDO, 输出 2.7~3.6V 可调, step 60mV, 支持 3.0V 和 3.3V LCD 屏; 内部电阻串分为大小电阻串两档, 小电阻 20k, 大电阻 220k; 小电阻分时开启方案, 需外接 470nf 电容滤波。

小电阻分时开启和大小电阻切换方案仅针对驱动波形选择为 TYPE B 时。当驱动波形选择为 TYPE A, 选择为小电阻始终驱动。

10.1.6 LCD 帧缓冲映射

LCD_BUFx 寄存器与不同段码规格的 LCD 屏映射关系如下所示。

- 当使用的是 8COM 时, 需要 36 个 LCD_BUF, 最大可以支持 8*36 的 LCD 屏

LCD_BUF[i] i=0~35 SEG 最大 36	SEG[i+4] COM7	SEG[i+4] COM6	SEG[i+4] COM5	SEG[i+4] COM4	SEG[i+4] COM3	SEG[i+4] COM2	SEG[i+4] COM1	SEG[i+4] COM0
-----------------------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------

ii. 当使用的是 6COM 时, 需要 38 个 LCD_BUF, 最大可以支持 6*38 的 LCD 屏

LCD_BUF[i] i=0~37 SEG 最大 38	-	-	SEG[i+2] COM5	SEG[i+2] COM4	SEG[i+2] COM3	SEG[i+2] COM2	SEG[i+2] COM1	SEG[i+2] COM0
-----------------------------------	---	---	------------------	------------------	------------------	------------------	------------------	------------------

iii. 当使用的是 4COM/3COM/2COM/1COM 时, 需要 20 个 LCD_BUF, 最大可以支持 4*40 的 LCD 屏

LCD_BUF[i] i=0~19 SEG 最大 40	SEG[2*i+1] COM3	SEG[2*i+1] COM2	SEG[2*i+1] COM1	SEG[2*i+1] COM0	SEG[2*i] COM3	SEG[2*i] COM2	SEG[2*i] COM1	SEG[2*i] COM0
-----------------------------------	--------------------	--------------------	--------------------	--------------------	------------------	------------------	------------------	------------------

10.2 寄存器描述

LCD 寄存器基地址

模块名	物理地址	映射地址
LCD	0x40048000	0x40048000

LCD 寄存器偏移地址

寄存器名	地址偏移量	描述
LCD_CTL	0x0	LCD 控制寄存器
LCD_STATUS	0x4	LCD 状态寄存器
LCD_CLKDIV	0x8	LCD 时钟控制寄存器
LCD_BLINK	0xc	LCD 闪烁控制寄存器
LCD_PS (修改)	0x10	LCD PUMP 建立时间寄存器
LCD_RESCTL	Offset+0x14	LCD 内部电阻串控制寄存器
LCD_BUF[i]	0x20+i*1 (i=0-37)	LCD 数据寄存器 (共 38 个 8 位寄存器)

10.2.1 LCD 控制寄存器 LCD_CTL (0x0)

比特位	名称	描述	读 / 写标志	复位值
31:14	---	只读, 不可写	R	0
13	保留	保留寄存器位, 用户不要写 1.	R/W	0
12	PWD_PUMP	LCD PUMP 开关: 0: 开启 PUMP, LCD 电压由内部 PUMP 产生。 1: 关闭 PUMP, 使用电阻串分压方案	R/W	0
11	TYPE	LCD Drive Type Select 0: Type A 1: Type B	R/W	0
10:5	BIASLVL	LCD Bias 电压调节 控制 Charge Pump 输出不同幅度的电压以控制 LCD 的对比度	R/W	0
4	BIAS	LCD Bias 控制	R/W	0

		0: 1/3Bias 1: 1/4Bias		
3:1	DUTY	LCD 占空比控制 000: 静态输出 (COM0) 001: 1/2 占空比 (COM0~1) 010: 1/3 占空比 (COM0~2) 011: 1/4 占空比 (COM0~3) 100: 1/6 占空比 (COM0~5) 101: 1/8 占空比 (COM0~7) Other:预留	R/W	0
0	EN	LCD 模块使能 0: LCD 模块关闭 1: LCD 模块使能 Note: 1、LCD.EN=1 约 250ms 后, LCD_BUSY=1;且 LCD_BUSY 会一直保持 1; 2、当 LCD_BUSY 为 1 时, LCD_CTRL (除了 EN Bit 外), LCD_CLKDIV,LCD_BLINK,LCD_PS 寄存器不可修改; 3、LCD.EN=0 后 LCD_BUSY 的持续时间最长是 32 个 LCD 刷新周期, 如 LCD_CLKDIV 配置为 64HZ, LCD_BUSY 的持续时间最长为 $32 * (1/64) = 0.5s$; 故低功耗处理流程中 LCD.EN=0 后, 至少等 0.5s 后, 才能关闭 LCD 的 APB 时钟; 实际应用可不关闭 LCD 的 APB 时钟。 4、低功耗处理流程中, LCD 模块关闭前, 需先关闭相关模拟电路, 以保证低功耗要求	R/W	0

10.2.2 LCD 状态寄存器 LCD_STATUS (0x4)

比特位	名称	描述	读/写标志	复位值
31:7	---	只读, 不可写	R	0
6	LCD_BUSY	LCD Busy Bit 0: 不忙 1: 忙 Note: 1、LCD.EN=1 约 250ms 后, LCD_BUSY=1;且 LCD_BUSY 会一直保持 1; 2、当 LCD_BUSY 为 1 时, LCD_CTRL (除了 EN Bit 外), LCD_CLKDIV,LCD_BLINK,LCD_PS 寄存器不可修改; 3、LCD.EN=0 后 LCD_BUSY 的持续时间最长是 32 个 LCD 刷新周期, 如 LCD_CLKDIV 配置为 64HZ, LCD_BUSY 的持续时间最长为 $32 * (1/64) = 0.5s$; 故低功耗处理流程中 LCD.EN=0 后, 至少等 0.5s 相关模拟电路关闭后, 才能关闭 LCD 的 APB 时钟; 实际应用可不关闭 LCD 的 APB 时钟。	R	0
5	---	保留	R/W	0
4	---	保留	R/W	0

3	IRQOFFEN	Display Off IRQ Enble Bit 0: 不使能 1: 使能	R/W	0
2	IRQONEN	Display On IRQ Enble Bit 0: 不使能 1: 使能	R/W	0
1	DOFF	Display Off Pending Bit 0: 无中断事件 1: 显示由亮变灭时置位 Note: 写 1 清零	R/W	0
0	DON	Display On Pending Bit 0: 无中断事件 1: 显示由灭变亮时置位 Note: 写 1 清零	R/W	0

10.2.3 LCD 时钟控制寄存器 LCD_CLKDIV (0x8)

比特位	名称	描述	读/写标志	复位值
31:8	---	只读, 不可写	R	0
7:0	CLKDIV	LCD Clock 分频系数 $LCD_CLK = f_{osc} / (2 * (CLKDIV + 1))$ (f_{osc} 为 32768Hz)	R/W	0

10.2.4 LCD 闪烁控制寄存器 LCD_BLINK (0xC)

比特位	名称	描述	读/写标志	复位值
31:26	---	只读, 不可写	R	0
25:18	BLINK_TIME	步长是 0.25s, 支持 0~63.75s 设置 TON 显示周期内, 亮与灭的时间 = $0.25 * BLINK_TIME$ 。 Note: 当设置为 0 时, 代表在 TON 显示周期内长亮, 不闪烁。 当设置值大于 0 时, TON 必须为 BLINK_TIME 的 2^n 倍 (n 为大于 0 的整数)。	R/W	0
17:9	TOFF	步长是 0.25s, 支持 0~127.5s, 当使用该功能时请设置成大于 >3s; 实际时间为: $0.25s * TOFF$	R/W	0
8:0	TON	步长是 0.25s, 支持 0~127.5s, 当使用该功能时请设置成大于 >3s; 实际时间为: $0.25s * TON$	R/W	0

10.2.5 LCD 电荷泵建立时间寄存器 LCD_PS (0x10) (修改)

比特位	名称	描述	读/写标志	复位值
31:16	---	只读, 不可写	R	0
15:0	PS	LCD PUMP Setup time $Time = T_{osc} * (PS + 4)$ (T_{osc} 为 30.5uS) 备注: 用户不需要对该寄存器进行配置。 V2 版扩至 16bit	R/W	0xccc

10.2.6 LCD 内部电阻串控制寄存器 LCD_RESCTL (0x14)

Offset=0x14

比特位	名称	描述	读/写标志	复位值																																		
31:13	---	预留	R	0																																		
12:9	LDOS	<div>LDO 输出电平选择信号</div> <div>LDO 输出作为 LCDVD 电压，并通过电阻分压产生 LCDVC、LCDVB、LCDVA；</div> <div>LDO 输出电平从 2.7V~3.6V，共 16 档，档位步进 0.06V；</div> <table><tr><td>LDOS 配置</td><td>LDO 输出电压 V</td></tr><tr><td>0000</td><td>2.7</td></tr><tr><td>0001</td><td>2.76</td></tr><tr><td>0010</td><td>2.82</td></tr><tr><td>0011</td><td>2.88</td></tr><tr><td>0100</td><td>2.94</td></tr><tr><td>0101</td><td>3</td></tr><tr><td>0110</td><td>3.06</td></tr><tr><td>0111</td><td>3.12</td></tr><tr><td>1000</td><td>3.18</td></tr><tr><td>1001</td><td>3.24</td></tr><tr><td>1010</td><td>3.3</td></tr><tr><td>1011</td><td>3.36</td></tr><tr><td>1100</td><td>3.42</td></tr><tr><td>1101</td><td>3.48</td></tr><tr><td>1110</td><td>3.54</td></tr><tr><td>1111</td><td>3.6</td></tr></table> <div>复位时 LDO 输出 3.0V</div>	LDOS 配置	LDO 输出电压 V	0000	2.7	0001	2.76	0010	2.82	0011	2.88	0100	2.94	0101	3	0110	3.06	0111	3.12	1000	3.18	1001	3.24	1010	3.3	1011	3.36	1100	3.42	1101	3.48	1110	3.54	1111	3.6		0101
LDOS 配置	LDO 输出电压 V																																					
0000	2.7																																					
0001	2.76																																					
0010	2.82																																					
0011	2.88																																					
0100	2.94																																					
0101	3																																					
0110	3.06																																					
0111	3.12																																					
1000	3.18																																					
1001	3.24																																					
1010	3.3																																					
1011	3.36																																					
1100	3.42																																					
1101	3.48																																					
1110	3.54																																					
1111	3.6																																					
8	FCC	<div>快速充电控制：</div> <div>0：关闭快速充电功能</div> <div>1：打开快速充电功能；</div> <div>选择内部电阻分压，节点电压外接 470nf 电容，LCD 模块打开时，内部电阻调整至 5k，并保持 100ms，以完成对电容的快速充电。</div>		0																																		
7	RES_AO	<div>电阻串分时驱动开关</div> <div>0：电阻串分时驱动，根据 RES_DT 和 RES_FT 产生驱动信号</div> <div>1：电阻串始终被驱动，RES_DT 和 RES_FT 配置无效。</div> <div>选择 TYPE A 时，RES_AO 恒为 1。</div>		1																																		
6：4	RES_DT	<div>电阻串分压方式，刷屏周期内，20k 电阻驱动时间配置</div> <div>驱动时间 $Td=(RES_DT[2:0]+1)*Tosc$ (Tosc 为 30.5uS) 。</div>		000																																		
3：2	RES_FT	<div>电阻串分压方式，刷屏周期内，20k 电阻驱动次数配置</div> <div>00：驱动 1 次</div> <div>01：驱动 2 次</div> <div>10：驱动 3 次</div> <div>11：驱动 4 次</div>		00																																		
1	RSM	<div>内部电阻分压电阻模式选择信号</div>		0																																		

		0 小电阻+开路, 20k 电阻分时开启, 200k 电阻始终短接 1 大小电阻切换, 20k 和 200k 电阻分时切换		
0	IRSN	电阻分压模式选择 0 内部电阻分压模式 1 保留 注意: 如果使用电阻分压模式, 该位必须配置为 0, 配置为 1 没有意义	R/W	1

如果选择为电阻分压方式, 建议对 LCD_RESCTL 配置如下:

典型的 3V 屏, LCD->RES_CTRL 寄存器配置值为: 0xb14;

典型的 3.3V 屏, LCD->RES_CTRL 寄存器配置值为: 0x1514。

10.2.7 LCD 数据寄存器 LCD_BUFx (x=0~34) (地址 0x20-0x41)

比特位	名称	描述	读/写标志	复位值
31:8	---	只读, 不可写	R	0
7:0	LCD_BUFx	LCD 屏 SEG 显示数据, 每位的物理意义如下: 0: 对应显示单元不显示 1: 对应显示单元显示	R/W	0

11 定时器（修改）

SoC内置2个32位定时器。每个定时器可完全独立工作；定时器之间不共享任何资源，可同步操作。

定时器适合多种用途，具有如下功能：

- ◎ 间隔定时
- ◎ 方波输出
- ◎ 外部/内部事件计数
- ◎ 单脉冲输出
- ◎ PWM 输出
- ◎ 脉宽测量

11.1 概述

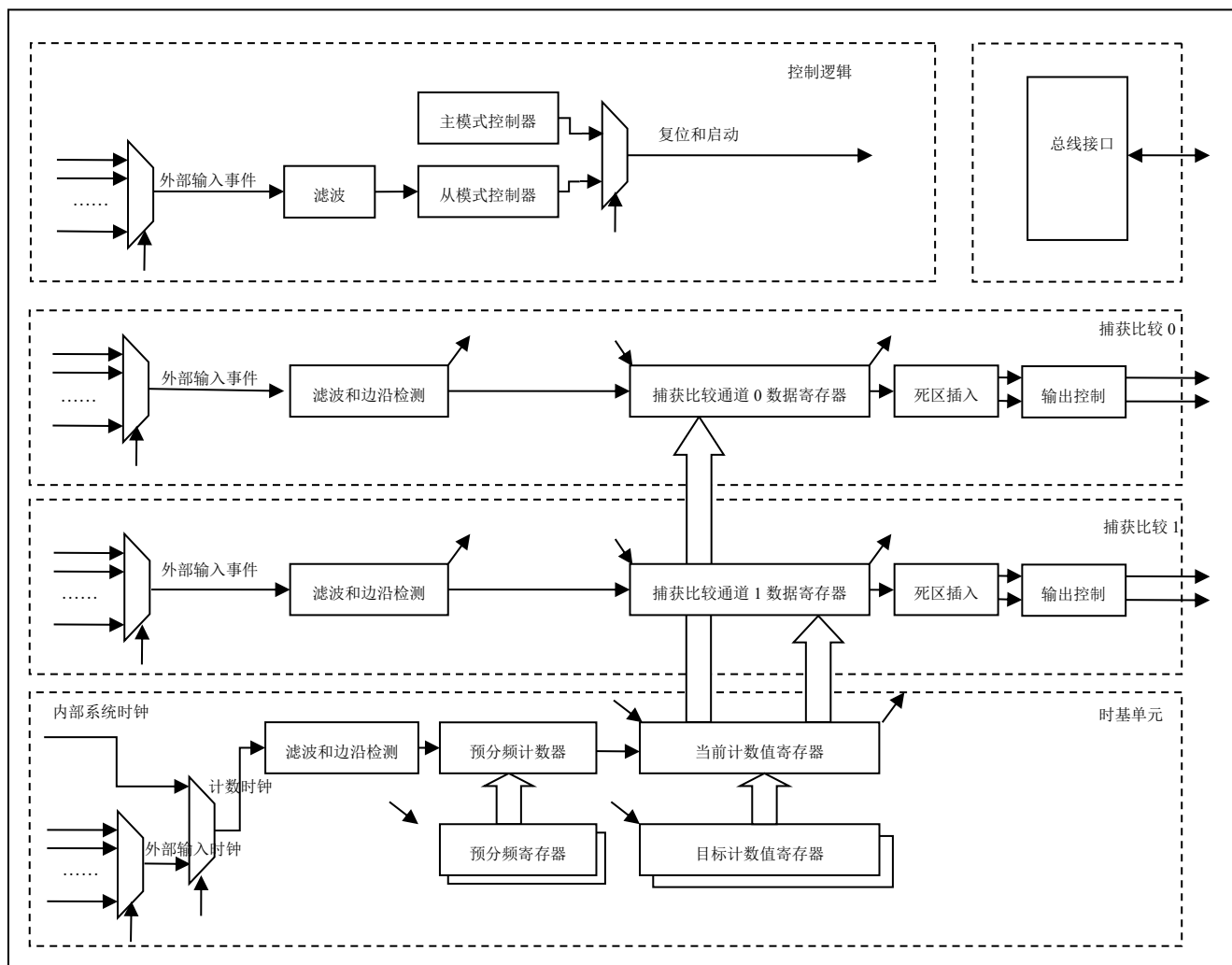
定时器具备如下特性：

- ◎ 2 个 32 位定时器，每个定时器：
 - 具有 1 个 32 位递增自动重装计数器；
 - 具有 16 位可编程预分频器，分频系数从 1~65535 之间可选；
 - 支持计数值动态访问；
 - 支持自由运行模式；
 - 支持单次运行；
- ◎ 每个定时器具有 2 个捕获/比较通道，每个通道可独立配置成：
 - 输入捕获；
 - 输出比较；
 - 单脉冲输出；
 - 互补 PWM：
 - ◆ 死区长度可编程：
 - 两个边沿的死区长度可独立设置；
 - 输出极性可配置；
 - ◆ 可配置的失效处理：
 - 输出失效；
 - 输出清除；
 - 输出三态；
- ◎ 从模式支持：
 - 外部复位和重新启动；
 - 外部门控；
- ◎ 输入捕获支持：
 - 上升沿捕获；
 - 下降沿捕获；
 - 双沿捕获；
 - 周期测量；
 - 脉宽测量；
 - 可选滤波；
- ◎ 输出比较支持：
 - 三态输出；
 - 反转输出；

- 固定电平输出；
- 脉宽可配置脉冲输出；
- 比较寄存器随时更新；
- ◎ 支持中断：
 - 计数溢出；
 - 输入捕获；
 - 输出比较；
- ◎ 灵活互补 PWM 波形调制（仅 TC1 CM1 通道）
 - 调制周期参数可配置；
 - 支持 DMA 读取参数
 - 互补 PWM 输出，死区可配置

11.2 功能框图

计数定时器的功能框图请参考下图所示。每个计数定时器内含一个 32 位计数器和 4 个 32 位捕获/比较通道。



11.3 PWM 调制控制模式（新增）

11.3.1.1 DMA 控制模式

通用控制模式 PWM 参数表长度为 K（以 Word 为单位），其中[31:16]表示电平有效周期数（N_Duty），[15:0]表示 1 个 PWM 周期所对应的系统时钟周期数（N_Pwm）。

波形发生以波形周期为循环处理单位，每个波形周期从 0 到 K-1 顺序读取 PWM 参数表的参数信息，动态更新每个 PWM 周期配置参数产生相应的 PWM 输出信号，一个波形周期对应 K 个 PWM 周期；当前波形周期结束后，继续循环产生下一个波形周期信号。

实际计数时钟周期为 N_Pwm+1，有效电平数周期为 N_Duty+1。

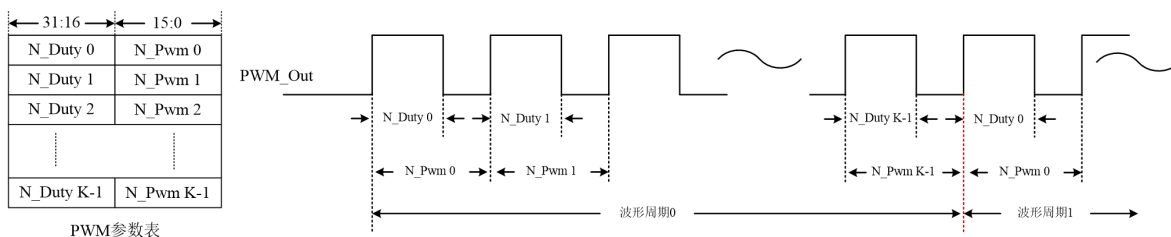


图 通用控制模式 PWM 参数表与 PWM 输出信号的关系图

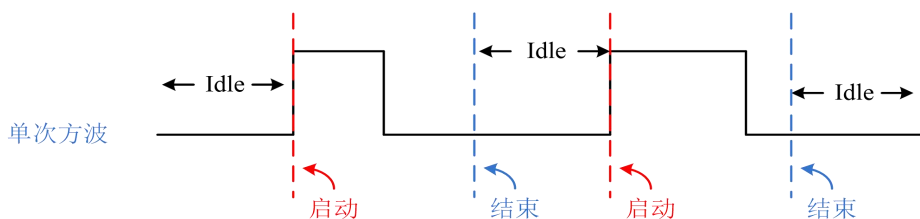
11.3.1.2 软件控制模式

通过寄存器作为 PWM 参数表，PWM_CNT 寄存器[15:0]表示 N_Pwm，TC1_CCD1 寄存器表示 N_Duty。

场景一：单次方波

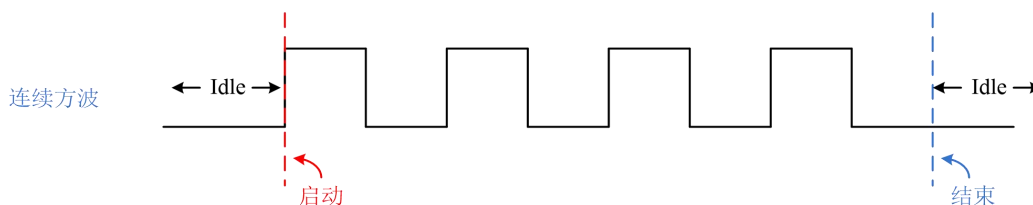
每次输入 1 个参数，发送一个方波。

首先输入 1 个参数，启动模块，发送一个方波后关闭模块，再输入参数，启动。



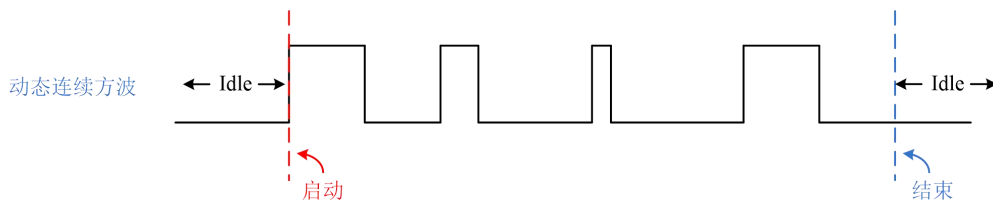
场景二：连续方波

输入 1 个参数，启动，输出指定长度方波或者用户结束输出



场景三：动态连续方波

输入 1 个参数，启动，根据 PWM_STA.BUFVLD 中断，每次中断写入一个新的数据。



11.4 寄存器描述

模块寄存器基地址

模块名	物理地址	映射地址
TC0	0x40010000	0x40010000
TC1	0x40014000	0x40014000

TC 模块寄存器偏移地址

寄存器名	地址偏移量	描述
TC_CNT	0x0	当前计数值指示
TC_PS	0x4	预分频寄存器
TC_DN	0xC	目标计数值寄存器
TC_CCD0	0x14	捕获比较通道 0 数据寄存器
TC_CCD1	0x18	捕获比较通道 1 数据寄存器
TC_CCFG	0x1C	时钟配置寄存器
TC_CTRL	0x20	控制寄存器
TC_CM0	0x24	捕获比较通道 0 模式寄存器
TC_CM1	0x28	捕获比较通道 1 模式寄存器
TC_IE	0x2C	中断使能寄存器
TC_STA	0x30	状态寄存器

11.4.1 当前计数值寄存器 TC_CNT (0x00)

比特位	名称	描述	读/写标志	复位值
31:0	CNT	当前计数值	R	0

11.4.2 预分频寄存器 TC_PS (0x04)

比特位	名称	描述	读/写标志	复位值
31:16	----	只读，不可写	R	0
15:0	PS	分频系数，分频值 (PS+1)，0 为不分频	R/W	0

11.4.3 目标计数值寄存器 TC_DN (0x0C)

比特位	名称	描述	读/写标志	复位值
31:0	DN	目标计数值，实际计数时钟周期为 DN+1	R/W	0

11.4.4 捕获比较通道 0 数据寄存器 TC_CCD0 (0x014)

比特位	名称	描述	读/写标志	复位值
31:0	CCD	捕获比较数据	R/W	0

注：通道 0 配置为捕获功能（即 TC_CM0 寄存器的 CCM 位域为 0）时，TC_CCD0 寄存器不可写

11.4.5 捕获比较通道 1 数据寄存器 TC_CCD1(0x018)

比特位	名称	描述	读/写标志	复位值
31:0	CCD	捕获比较数据	R/W	0

注：通道 1 配置为捕获功能（即 TC_CM1 寄存器的 CCM 位域为 0）时，TC_CCD1 寄存器不可写

11.4.6 时钟配置寄存器 TC_CCFG (0x01C)

比特位	名称	描述	读/写标志	复位值
31:24	---	只读，不可写	R	0
23:16	FLT0PT	外部输入时钟滤波参数设置，设置滤波的时钟周期数。	R/W	0
15	---	只读，不可写	R	0
14:13	ECLKMODE	外部输入时钟模式： 00：上升沿 01：下降沿 10：双边沿 11：保留（等效于双边沿）	R/W	0
12:8	CS	外部输入时钟选择： 0：UART0 RXD 1：UART1 RXD 2：UART2 RXD 3：UART3 RXD 4：另一定时器(TC0或TC1)的输出outn[0] 5：另一定时器(TC0或TC1)的输出outp[0] 6：另一定时器(TC0或TC1)的输出outn[1] 7：另一定时器(TC0或TC1)的输出outp[1] 8：UART4 RXD 9：UART5 RXD 10:7816_0输入P41 11:7816_1输入P42 12:7816_1输入P43 13~15：保留 16：sf_out 17：qf_out 18：pf_out 19：rtc_out 20：p1[0]外部I0口 21：p1[1]外部I0口 22：p1[2]、P5[2]外部I0口 23：p1[3]、P5[3]外部I0口 24：p1[4]、P5[4]外部I0口 25：p1[5]、P5[5]外部I0口 26：p1[6]、P5[6]外部I0口 27：p1[7]、P5[7]外部I0口 28：p3[0]外部I0口 29：p3[1]外部I0口 30：p3[3]外部I0口 31：p3[5]外部I0口	R/W	0
7:2	---	只读，不可写	R	0
1	FLTEN	外部输入时钟滤波使能 0：不使能	R/W	0

		1: 使能		
0	CM	计数时钟源选择: 0: 内部系统时钟 1: 外部输入时钟 (由 CS 选择时钟源)	R/W	0

11.4.7 控制寄存器 TC_CTRL (0x020)

比特位	名称	描述	读/写标志	复位值
31:29	---	只读, 不可写	R	0
28	DBGSTBDIS	调试时计数器计数使能: 0: 不使能 (当CPU处于调试状态时计数器停止计数) 1: 使能 (当CPU处于调试状态时计数器继续计数) 注: CPU处于调试状态指的是用户通过调试接口将Cortex M0停住 (PC指针停止计数)。	R/W	0
27	Reserved	保留, 请勿操作	R/W	0
26	Reserved	保留, 请勿操作	R/W	0
25	Reserved	保留, 请勿操作	R/W	0
24	Reserved	保留, 请勿操作	R/W	0
23:21	---	只读, 不可写	R	0
20	SLVGATELVL	从模式门控模式有效电平: 0: 有效电平为低电平 1: 有效电平为高电平	R/W	0
19:12	SLVFLTPT	从模式输入滤波参数	R/W	0
11:10	SLVTRGMODE	从模式控制模式选择: 00: 上升沿清零内部计数器 01: 下降沿清零内部计数器 10: 双边沿清零内部计数器 11: 门控模式 (当外部输入信号为有效电平时内部计数器计数)	R/W	0
9:5	SLVCHANSEL	从模式外部输入事件选择: 与时钟配置寄存器 (0x01C) 中 CS 位域定义的外部输入时钟定义一致。	R/W	0
4	OPS	单次计数模式选择: 0: 不使能单次计数模式 (计数溢出后不停止, 循环计数); 1: 使能单次计数模式 (计数溢出后停止)	R/W	0
3	SLVFLTEN	从模式外部输入事件滤波使能: 0: 不使能 1: 使能	R/W	0
2	SLVEN	从模式使能: 0: 不使能 1: 使能	R/W	0
1	---	只读, 不可写	R	0
0	START	定时器启动: 0: 停止 1: 启动	R/W	0

11.4.8 捕获比较通道 0/1 模式寄存器 TC_CM0/1 (0x024 和 0x028)

比特位	名称	描述	读/写标志	复位值
31:30	---	只读，不可写	R	0
29	DFTLVL	比较输出缺省电平： 0：低电平 1：高电平	R/W	0
28	EFELVL	比较输出有效电平： 0：低电平 1：高电平	R/W	0
27:25	OM	比较输出模式： 000：无输出（三态） 001：设置为有效电平 010：设置为无效电平 011：翻转 100：强制为有效电平 101：强制为无效电平 110：PWM模式1 111：PWM 模式 2	R/W	0
24:20	CS	捕获外部输入事件选择： 与时钟配置寄存器（0x01C）中 CS 位域定义的外部输入时钟定义一致。	R/W	0
19	FLTEN	捕获外部输入事件滤波器使能： 0：不使能 1：使能	R/W	0
18:11	FLTOUT	捕获外部输入事件滤波器参数	R/W	0
10:9	CPOL	捕获外部输入事件极性选择： 00：上升沿 01：下降沿 10：双边沿 11：保留	R/W	0
8:3	DL	比较输出死区长度（只支持PWM模式1和PWM模式2，其他模式下该位无效）	R/W	0
2	DIEN	比较输出死区插入使能：（只支持PWM模式1和PWM模式2，其他模式下该位无效） 0：不使能 1：使能	R/W	0
1	CCM	捕获比较模式选择： 0：捕获 1：比较	R/W	0
0	ENABLE	通道使能： 0：不使能 1：使能	R/W	0

11.4.9 中断使能寄存器 TC_IE (0x2C)

比特位	名称	描述	读/写标志	复位值
-----	----	----	-------	-----

31:4	---	只读，不可写	R	0
3	SLVIE	从模式中断使能： 0：不使能 1：使能	R/W	0
2	CC1IE	捕获比较通道1中断使能： 0：不使能 1：使能	R/W	0
1	CC0IE	捕获比较通道0中断使能： 0：不使能 1：使能	R/W	0
0	OVIE	溢出中断使能： 0：不使能 1：使能	R/W	0

11.4.10 状态寄存器 TC_STA (0x30)

比特位	名称	描述	读/写标志	复位值
31:4	---	只读，不可写	R	0
3	SLVF	从模式事件标志：（写1清0） 0：无从模式事件 1：有从模式事件	R/W	0
2	CC1F	捕获比较通道1事件标志：（写1清0） 0：无捕获或比较事件 1：有捕获或比较事件	R/W	0
1	CC0F	捕获比较通道0事件标志：（写1清0） 0：无捕获或比较事件 1：有捕获或比较事件	R/W	0
0	OVF	溢出中断标志：（写1清0） 0：无溢出事件 1：有溢出事件	R/W	0

11.5 典型应用

11.5.1 自动运行模式，定时功能

自动运行模式即间隔定时功能。

基本定时功能，只需要对以下寄存器进行设置：

- 1、目标计数值寄存器，即为定时时长，由计数时钟计数。
- 2、中断使能寄存器中使能溢出中断使能。
- 3、控制寄存器，启动定时器。

定时器会以目标计数值为周期产生中断。

常用可选择功能配置说明：

- 1、可修改预分频寄存器值，改变定时器计数时钟的频率。
- 2、可配置时钟配置寄存器，CM 修改配置为外部输入时钟，同时修改 CS 位配置的外部输入时钟选择。外部输入时钟频率不可比内部系统时钟频率的二分频高。
- 3、单次计数模式，定时器溢出后就停止；配置控制寄存器的 OPS 位为 1 即为单次计数模式。

- 外部输入时钟源为另外一个定时器的输出，则可连接为两个定时器的级联模式，可增加定时器的寄存器的位宽。

11.5.2 输入捕获模式，脉宽测量功能

输入捕获模式的主要功能是可以测试脉冲的宽度。

基本脉宽测试功能，只需要对以下寄存器进行设置：

- 目标计数值寄存器，由计数时钟计数，可设置为最大值。
- 捕获比较通道 0/1 模式寄存器设置，ENABLE 通道 0/1 使能，CCM 配置为捕获模式，CPOL 选择捕获极性，CS 选择外部输入事件。
- 中断使能寄存器中使能捕获比较通道 0/1 中断使能。
- 控制寄存器，启动定时器。

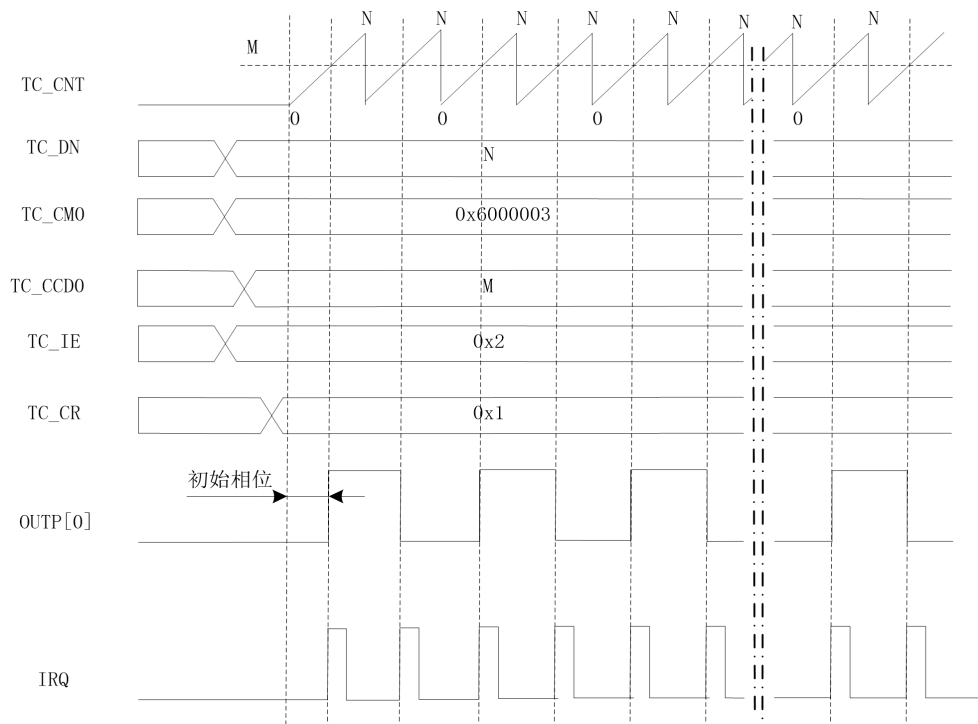
定时器捕获到外部输入事件的捕获极性时，产生中断，同时当前计数值会保存在捕获比较通道数据寄存器中。若采用两个通道，一个通道捕获上升沿，一个通道捕获下降沿，则可通过两个通道的数据寄存器计算出脉冲宽度。

常用可选择功能配置说明：

- 可修改预分频寄存器值，改变定时器计数时钟频率。
- 可配置时钟配置寄存器，CM 修改配置为外部输入时钟，同时修改 CS 位配置的外部输入时钟源。外部输入时钟源频率不可比内部系统时钟频率的二分频高。
- 外部输入事件的滤波功能，使能捕获比较通道模式寄存器中 FLTEN 滤波功能，通过配置 FLTOUT 设置滤波周期数。
- 若外部输入事件为另外一个定时器的输出，则可连接为两个定时器的级联模式。

11.5.3 比较输出模式，方波输出功能

方波输出功能即对 TC 的计数时钟进行分频输出的功能。每个定时器有两个输出通道，每个通道有 P 和 N 两个输出端，其中 P 为正向输出端口，N 为 P 的反向输出端口。



方波输出功能，只需要对以下寄存器进行设置：

- 1、目标计数值寄存器，由计数时钟计数，可设置为最大值。
- 2、捕获比较通道 0/1 模式寄存器设置，ENABLE 通道 0/1 使能，CCM 配置为比较模式，DFTLVL 配置缺省电平，EFELVL 配置有效电平，OM 输出配置为翻转功能。
- 3、设置捕获比较通道 0/1 数据寄存器。（设置值不大于目标计数值寄存器）
- 4、中断使能寄存器中使能捕获比较通道 0/1 中断使能。
- 5、控制寄存器，启动定时器。

目标计数值寄存器的值决定输出方波的周期，捕获比较通道 0/1 数据寄存器的值为输出翻转点。

常用可选择功能配置说明：

- 1、可修改预分频寄存器值，改变定时器计数时钟的频率。
- 2、可配置时钟配置寄存器，CM 修改配置为外部输入时钟，同时修改 CS 位配置的外部输入时钟源。外部输入时钟源频率不可比内部系统时钟频率的二分频高。
- 3、外部输入时钟滤波功能，使能时钟配置寄存器中 FLTEN 滤波功能，通过配置 FLT OPT 设置滤波周期数。

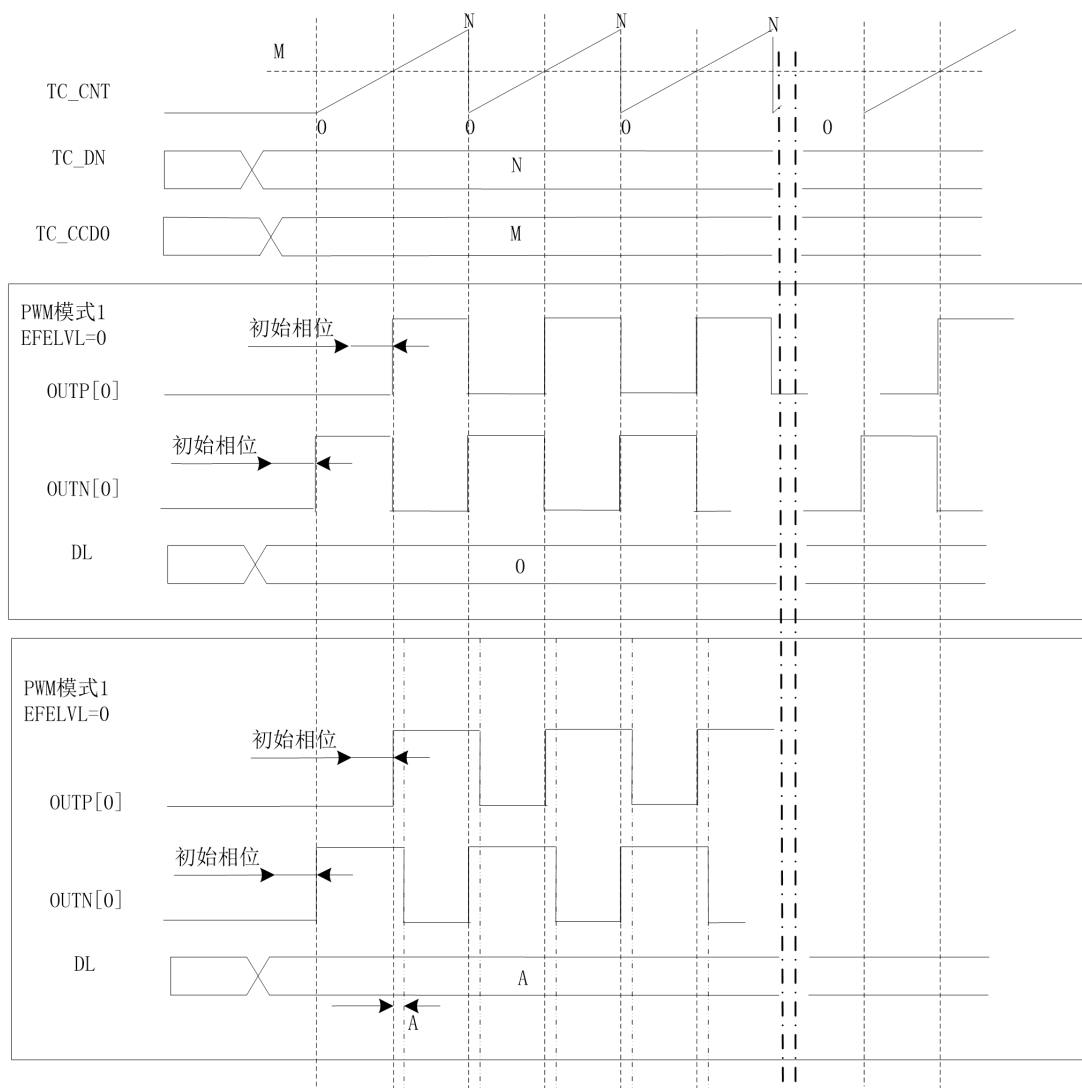
11.5.4 比较输出模式，PWM 输出功能

脉冲宽度调制（PWM）模式可以产生一个由 TC_DN 寄存器确定频率、由 TC_CCDx 寄存器确定占空比的信号。支持两种 PWM 模式：PWM 模式 1 和 PWM 模式 2：

PWM 模式 1：如果 $TC_CNT < TC_CCDx$ 时，输出为有效电平，否则为无效电平。

PWM 模式 2：如果 $TC_CNT \geq TC_CCDx$ 时，输出为有效电平，否则为无效电平。

下图为 PWM 模式 1 的典型应用图。



PWM 输出功能，只需要对以下寄存器进行设置：

- 1、目标计数值寄存器，由计数时钟计数。
- 2、捕获比较通道 0/1 模式寄存器设置，ENABLE 通道 0/1 使能，CCM 配置为比较模式，DFTLVL 配置缺省电平，EFELVL 配置有效电平，OM 输出配置为 PWM 模式 1 或者 PWM 模式 2。
- 3、设置捕获比较通道 0/1 数据寄存器，必须比目标计数值寄存器小。
- 4、控制寄存器，启动定时器。

在通道的 P 端则输出 PWM 模式 1/PWM 模式 2 的正向波形，在通道的 N 端输出与 P 端反向的波形。

PWM 模式 1：周期为目标计数值寄存器值加 1，有效电平周期为通道数据寄存器值的周期数加 1。

PWM 模式 2：周期为目标计数值寄存器值加 1，无效电平周期为通道数据寄存器值的周期数加 1。

常用可选择功能配置说明：

- 1、可修改预分频寄存器值，改变定时器计数时钟频率。
- 2、可配置时钟配置寄存器，CM 修改配置为外部输入时钟，同时修改 CS 位配置的外部输入时钟源。外部输入时钟源频率不可比内部系统时钟频率的二分频高。
- 3、死区插入的互补输出，DIEN 死区插入使能，DL 配置死区插入长度。在 P 和 N 的两个沿切换中间加入延迟，即不让两个沿同时翻转。

当有效电平 EFELVL 为低电平时：P 和 N 的输出下降沿延后 DL 个周期。

当有效电平 EFELVL 为高电平时：P 和 N 的输出上升沿延后 DL 个周期。

11.5.5 从模式，外部清零和门控功能

从模式在原有的功能上增加了外部输入事件控制内部计数器的清零和门控的功能。

外部清零功能，只需要对以下寄存器进行设置：

- 1、目标计数值寄存器，由计数时钟计数。
- 2、中断使能寄存器中使能溢出中断使能。
- 3、控制寄存器，SLVEN 从模式使能，SLVTRGMODE 从模式控制模式选择，SLVCHANSEL 从模式外部输入事件选择，
- 4、控制寄存器，启动定时器。

这样在自由运行模式下增加了一个外部清零内部 CNT 的功能。

常用可选择功能配置说明：

- 1、可修改预分频寄存器值，改变定时器计数时钟频率。
- 2、可配置时钟配置寄存器，CM 修改配置为外部输入时钟，同时修改 CS 位配置的外部输入时钟源。外部输入时钟源频率不可比内部系统时钟频率的二分频高。
- 3、单次计数模式，定时器溢出后就停止；配置控制寄存器的 OPS 位为 1 即为单次计数模式。
- 4、外部输入时钟源为另外一个定时器的输出，则可连接为两个定时器的级联模式，可增加定时器的寄存器的位宽。
- 5、从模式控制模式选择中，若选择为门控模式，SLVGATELVL 从模式门控有效电平配置，从模式的输入在门控有效电平时内部计数器才会计数。
- 6、从模式下配置为输入捕获方式，从模式的外部输入事件和捕获的外部输入事件配置为相同时，捕获选择一个沿，从模式极性选择另外一个沿，则可由捕获比较通道数据寄存器直接获取脉冲宽度。

11.6 操作步骤

简单介绍作为间隔定时器的操作步骤，将 TC0 设置为 1MS 间隔定时中断：

- 1、配置系统控制章节中模块使能 0 寄存器 MOD0_EN 第 4 位启动 TC0 的时钟。注意写系统控制章节寄存器需先将 SYS_PS 写为 0x82，打开写使能，需写入的系统控制寄存器操作完成后需将 SYS_PS 写入 0，关闭写使能。
- 2、设置 TC0 模块寄存器：
时钟配置寄存器 TC_CCFG 配置为内部系统时钟，TC0->CCFG = 0；
配置预分频寄存器 TC_PS；
配置目标计数值寄存器 TC_DN；
配置控制寄存器 TC_CR，TC0->CTRL = 0x01；启动 TC0 定时器
设置中断使能寄存器 TC_IE，TC0->IE = 0x01；配置为溢出中断使能；
当系统时钟为 3.6864MHZ 时，TC0->PS = 255；TC0->DN = 13 配置后既可产生 1MS 中断。 $(3.6864\text{MHZ} / (255+1)) / (13+1) = 1\text{MS}$ 。
- 3、开启 TC0 中断使能，NVIC_EnableIRQ(TC0_IRQn)；
- 4、编写中断服务程序：

```
void TC0_HANDLER(void)
{
    /* Start adding user code. Do not edit comment generated here */
}
```

- 5、配置完成后即可产生 1MS 中断。

11.7 PWM 模式寄存器描述（新增）

寄存器列表

寄存器名	地址偏移量	描述
PWM_CFG	Offset+0x34	PWM 配置寄存器
PWM_CTL	Offset+0x38	PWM 控制寄存器
PWM_STA	Offset+0x3C	PWM 状态寄存器
PWM_CNT	Offset+0x40	PWM 参数寄存器
PWM_DMA_BADR	Offset+0x44	PWM DMA 起始地址寄存器
PWM_DMA_LEN	Offset+0x48	PWM DMA 数据长度寄存器
PWM_DMA_ADR	Offset+0x4C	PWM DMA 当前地址寄存器

11.7.1 PWM_CFG (0x34)

PWM 控制寄存器，支持 bitband 操作

若在波形产生过程中配置下面寄存器的控制位会对应改变输出结果，所以在波形产生过程中请勿配置下面寄存器

比特位	名称	描述	读/写标志	复位值
31:17	---	预留	R	0
16:7	WL	PWM wave length, PWM 输出信号长度配置参数 仅在长度可配模式有效。 表示的是输出 WL+1 个完整信号周期 例：dam 模式下，使用 PWM 模拟正弦波，表示输出 WL+1 个正弦波的周期，即遍历完成 dam 数据表算一个完整信号周期；软件模式下，输出方波时，表示输出 WL+1 个方波，即每使用一次 PWM 参数算一个完整信号周期。	R/W	0
6	ORV	Output Remain Valid, 输出信号始终有效 0: 长度可配模式 1: 始终有效模式 长度可配表示使用 WL 参数；始终有效则是一直输出波形直到关闭使能。	R/W	0
5	CSS	Complete Signal selection 信号完整性输出选择 0: 软件关闭模块后会在完成一个完整信号的输出后停止波形输出。在 DMA 模式下，dma 长度结束表示一个完整信号；软件模式下，一个 pwm 波表示一个完整信号。 1: 软件关闭模块后立即停止 PWM 输出	R/W	0
4	CMS	Control mode selection, PWM 调制控制模式。 0: dma 模式 1: 软件控制模式	R/W	0
3	CYCDON EIE	输出信号周期完成中断使能 0: 不使能	R/W	0

		1: 使能		
2	BUFVLDIE	PWM 参数缓存空中断使能（仅软件模式使用） 0: 不使能 1: 使能	R/W	0
1	DATERRIE	数据错误中断使能 0: 不使能 1: 使能	R/W	0
0	DONEIE	PWM 输出完成中断 0: 不使能 1: 使能	R/W	0

11.7.2 PWM_CTL (0x38)

PWM 控制寄存器

比特位	名称	描述	读/写标志	复位值
31:1	---	预留	R	0
0	EN	PWM 模块使能位 0: PWM 调制关闭 1: PWM 调制打开 只通过软件关闭和打开。 在 DMA 模式下，dma_len 为 0 时，使能后，必须记得关闭使能，否则会影响 TC 功能。	R/W	0

11.7.3 PWM_STA (0x3C)

PWM 状态指示寄存器

比特位	名称	描述	读/写标志	复位值
31:4	---	预留	R	0
4	CYCDONE	输出信号周期完成中断 0: 未完成一个周期信号的输出 1: 完成一个周期信号的输出 写 1 清 0 DMA 模式下，当遍历完成 DMA 表后置起，例，模拟正弦波输出，当正弦波一个周期结束后置起。 软件控制模式下，每个 PWM 方波输出完成后置起，即每个参数计数结束后置起。	R/W	0
3	BUFVLD	PWM 参数缓存空中断（仅软件模式使用） 0: 软件控制模式时，缓存中存在 PWM 参数 1: 软件控制模式时，可配置新的 PWM 参数 写 1 清 0 在波形输出开始时置起；在每个 PWM 参数计数结束后置起。	R/W	0
2	DATERR	数据空错误中断 0: 正常读取数据 1: DMA 模式下由于竞争，无法及时读取数据；软件	R/W	0

		控制模式下，由于用户没有及时写入新的数据。 写 1 清 0 DMA 模式下，当竞争时间大于设定的 PWM 参数，置起。 软件控制模式下，在上一个 PWM 参数计数结束前没有写入新的数据，计数结束后置起；启动前没有写入 PWM 参数，启动后置起。		
1	DONE	PWM 输出完成中断 0: 波形输出未完成 1: 波形输出完成 写 1 清 0 当 WL 计数完成后置起；当 CSS=0，关断后，波形输出完成后置起；当 CSS=1，关断后，不置起。	R/W	0
0	BUSY	PWM 信号发送状态 0: 空闲状态 1: 波形发生状态	R	0

11.7.4 PWM_CNT (0x40)

PWM 参数寄存器

N_Duty 的配置在 TC1_CCD1

比特位	名称	描述	读/写标志	复位值
31:16	--	保留	R/W	0
15:0	N_Pwm	表示当前 PWM 周期的系统时钟周期长度。 实际计数时钟周期为 N_Pwm+1 在软件控制模式中，缓存空可配置，非空时配置覆盖原来的值 在 DMA 控制模式中，请勿配置。	R/W	0

11.7.5 PWM_DMA_BADR (0x44)

PWM DMA 起始地址

比特位	名称	描述	读/写标志	复位值
31:16	Reserved	保留。	R	0
14:0	BADR	DMA 起始地址（Word 地址）	R/W	0

11.7.6 PWM_DMA_LEN (0x48)

PWM DMA 数据长度

比特位	名称	描述	读/写标志	复位值
31:15	Reserved	保留。	R	0
14:0	LEN	DMA 数据长度，即 PWM 参数表的长度，配置为 n，长度= n Word，表的大小为 n 建议输入长度 LEN>1；若需要 LEN=1 时建议使用软件控制模式。在 DMA 模式下，当 LEN=1 时，若使用长度可配置功能，最终长度输出为 WL+2。	R/W	0

11.7.7 PWM_DMA_ADR (0x4C)

PWM DMA 当前地址

比特位	名称	描述	读/写标志	复位值
31:15	Reserved	保留。	R	0
14:0	ADR	DMA 当前地址	R	0

11.8 PWM 模式软件操作流程

11.8.1 参数限制

占空比周期数限制：0<N_Duty<N_Pwm-1

在软件模式下产生动态连续方波时，N_PWM 建议大于 11，当小于等于 11 时，无法及时写入新的数据（进中断->写 N_DUTY->写 N_PWM->清中断；这些操作需要一定的时间）。

死区长度 DL 建议：DL<=N_Duty，DL<N_Pwm-N_Duty。

11.8.2 DMA 控制模式

1. MOD0_EN = 1<<5; //开启 TC1 的时钟
2. TC1_CM1 = 1 | 1<<1 | DIEN<<2 | DL<<3 | OM <<25 | EFELVL << 28 | DFTLVL <<29;
//配置 TC1 的 1 通道，其中 OM 可选 PWM 模式 1 或 2。
3. PWM_STA = 0xE; //清除所有中断标志
4. PWM_DMA_BADR = 0xxxx; //配置 DMA 数据起始地址
5. PWM_DMA_LEN = 0xxxx; //配置数据长度
6. PWM_CFG = 0xxxx; //配置相关控制
7. PWM_CTL = 0x1; //使能
- 8.1 若设置长度可配置，等待结束，通过中断 DONE 或标志 BUSY 来判断
- 8.2 若设置持续输出，则软件关闭
9. PWM_CTL = 0x0; //关闭使能
10. PWM_STA = 0xE; //清除中断

11.8.3 软件控制模式

11.8.3.1 单次方波

1. MOD0_EN = 1<<5; //开启 TC1 的时钟
2. TC1_CM1 = 1 | 1<<1 | DIEN<<2 | DL<<3 | OM <<25 | EFELVL << 28 | DFTLVL <<29;
//配置 TC1 的 1 通道，其中 OM 可选 PWM 模式 1 或 2。
3. PWM_STA = 0xE; //清除所有中断标志
4. TC1_CCD1 = 0xxxx; //配置 N_Duty 参数
5. PWM_CNT = 0xxxx; //配置 N_Pwm 参数
6. PWM_CFG = 0xxxx; //配置相关控制，关闭 DATERR，BUFVLD 中断使能
//使用长度可配模式，设置 WL=0;
7. PWM_CTL = 0x1; //使能
8. 等待 DONE 中断或者标志 BUSY=0;
9. PWM_STA = 0xE; //清除中断
10. PWM_CTL = 0x0; //关闭使能
11. TC1_CCD1 = 0xxxx; //配置 N_Duty 参数
12. PWM_CNT = 0xxxx; //配置 N_Pwm 参数

- 13. PWM_CTL = 0x1; //使能
- 14. 等待 DONE 中断或者标志 BUSY=0;
- 15. PWM_STA = 0xE; //清除中断
- 16. PWM_CTL = 0x0; //关闭使能

11.8.3.2 连续方波

- 1. MOD0_EN = 1<<5; //开启 TC1 的时钟
- 2. TC1_CM1 = 1 | 1<<1 | DIEN<<2 | DL<<3 | OM <<25 | EFELVL << 28 | DFTLVL <<29;
//配置 TC1 的 1 通道, 其中 OM 可选 PWM 模式 1 或 2。
- 3. PWM_STA = 0xE; //清除所有中断标志
- 4. TC1_CCD1 = 0xxxx; //配置 N_Duty 参数
- 5. PWM_CNT = 0xxxx; //配置 N_Pwm 参数
- 6. PWM_CFG = 0xxxx; //配置相关控制, 关闭 DATERR, BUFVLD 中断使能
- 7. PWM_CTL = 0x1; //使能
- 8.1 若设置长度可配置, 等待结束, 通过中断 DONE 或标志 BUSY 来判断
- 8.2 若设置持续输出, 则软件关闭
- 9. PWM_CTL = 0x0; //关闭使能
- 10. PWM_STA = 0xE; //清除中断

11.8.3.3 动态连续方波

- 1. MOD0_EN = 1<<5; //开启 TC1 的时钟
- 2. TC1_CM1 = 1 | 1<<1 | DIEN<<2 | DL<<3 | OM <<25 | EFELVL << 28 | DFTLVL <<29;
//配置 TC1 的 1 通道, 其中 OM 可选 PWM 模式 1 或 2。
- 3. PWM_STA = 0xE; //清除所有中断标志
- 4. TC1_CCD1 = 0xxxx; //配置 N_Duty 参数
- 5. PWM_CNT = 0xxxx; //配置 N_Pwm 参数
- 6. PWM_CFG = 0xxxx; //配置相关控制, 使能 BUFVLD 中断
- // 若配置长度始终有效, 建议配置 CSS=0;
- 7. PWM_CTL = 0x1; //使能
- 8. 等待 BUFVLF 中断
- 9. TC1_CCD1 = 0xxxx; //配置 N_Duty 参数
- 10. PWM_CNT = 0xxxx; //配置 N_Pwm 参数
- //N_Duty、N_Pwm 按需配置, 可只配置一个。
- //若两个参数都没有配置, 则会触发 daterr 中断标志。
- 11. PWM_STA = 0xE; //清除中断标志
- 12. 循环步骤 8~11;
- 13.1 若配置长度可配模式
- 14.1 所有数据写完后, 等待 BUFVLF 中断。
- 15.1 PWM_STA = 0xE; //清除中断标志。
- 16.1 等待 DONE 中断或者标志 BUSY=0;
- 17.1 PWM_CTL = 0x0; //关闭使能
- 18.1 PWM_STA = 0xE; //清除中断标志。
- 13.2 若配置长度始终有效模式且 CSS=0;

14.2 所有数据写完后，等待 BUFVLF 中断。

15.2 PWM_CTL = 0x0; //关闭使能

16.2 等待 DONE 中断或者标志 BUSY=0;

17.2 PWM_STA = 0xE; //清除中断标志。

12 模拟外设

12.1 特点

12bit SAR ADC;

- 对 VBAT 引脚输入的电压启动测量时，内部有两个 300K 电阻做分压，输入电压会被缩小一半输入到 SAR ADC，SAR ADC 使用 0.5 倍 PGA 对信号再缩小一倍，即：3.6V 的电池，测量到的信号约为 0.9V。如果不启动测量，内部分压电阻会被关闭。
- 如果选择其他引脚（AIN）对输入电压进行测量，请将被测量电压缩小到 SAR ADC 测量范围内。
- AIN 最大支持 1.25V 输入，增益支持 0.5 倍、1 倍、1.5 倍、2 倍，**新增支持 0.25 倍增益**
- 多路复用，支持 AIN0~AIN6、VBAT、温度传感器做为输入，其中温度传感器的优先级最高；
- AIN0~AIN6 的输入阻抗约为 3M 欧姆，外部需要接 0.1uF 电容到地；
- ADC 不采样时，自动进入省电模式；每次采样从启动到完成约为 12ms。

一路 LVD 电路，主要特点如下：

- LVD 的输入可选为芯片电源，也可选择为外部 PIN 输入；
- LVD 的阈值可调，从 2.7V 到 4.9V 多个档位可设；
- 当选择为外部 PIN 输入时（LVDIN0），阈值固定为 1.25V 左右，内阻约为 1M 欧姆；

两路比较器电路 CMP1 和 CMP2，主要特点如下：

- 外部 PIN 输入，阈值固定为 1.25V 左右
- 功耗优于 1uA，可使用比较器进行主电上下电监测。
- 注意 CMP1 和 CMP2 内部默认有 600K 的内阻，可通过寄存器（SYS_PD (0x08)）选择关闭内部电阻。

12.2 寄存器

模拟外设模块的基址

模块名	物理地址	映射地址
ANA	0x4002C000	0x4002C000

模拟外设模块的寄存器偏移地址

寄存器名	地址偏移量	描述
SAR_CTRL	Offset+0x0	SAR-ADC 控制寄存器
SAR_START	Offset+0x4	SAR-ADC 启动寄存器
SAR_STAT	Offset+0x8	SAR-ADC 状态寄存器
SAR_DAT	Offset+0xC	SAR-ADC 数据寄存器
LVD_CTRL	Offset+0x10	比较器控制寄存器

LVD_STAT	Offset+0x14	比较器状态寄存器
SAR_CTRL1 (新增)	Offset+0x18	SAR-ADC 控制寄存器 1
SAR_DATA2 (新增)	Offset+0x1C	SAR-ADC 数据寄存器 2
SAR_CTRL3 (新增)	Offset+0x20	SAR-ADC 控制寄存器 3
SAR_DATA3 (新增)	Offset+0x24	SAR-ADC 数据寄存器 3

12.2.1 SAR_CTRL(0x0)

ADC 控制寄存器 地址 0x4002C000+ 0x0

比特位	名称	描述	读/写标志	复位值
31:17	---	只读，不可写	R	0
16:12	REF_WAIT	REF 在开启后到开启 ADC 需要等待的时间： 5'd0: 976uS 5'd31: (31+1)*976=31.232ms 即等待时间=(REF_WAIT+1)* 976uS 如果在运行模式下，计量开启则 REF 也开启，此等待时间可设为 0。 如果在计量未开启时，建议设置的等待时间大于 10ms。 注意：计量基准 REF 外接电容为 1uF+0.1uF。	R/W	0
11:7	SAR_WAIT	SAR ADC 在开启后到开始采样转换需要等待的时间： 5'd0: 30.5uS 5'd31: (31+1)*30.5=976us 即等待时间=(SAR_WAIT+1)*30.5us 注 1：启动 ADC 测量的步骤： 1、开启 REF，等待 REF_WAIT 时间； 2、开启 ADC 和温度传感器，等待 SAR_WAIT 时间； 3、输入时钟和复位信号，16 个时钟周期后得到采样结果。 以上步骤均由硬件自动控制实现。 注 2：建议客户应用程序将该位配置成 0xE，即 457.5us。	R/W	0xE
6	SAR_CH3	SAR-ADC Channel 选择，与 SAR_CH[2:0]共同组成一个 4bit 寄存器。参考 SAR_CH[2:0]说明。	R/W	0
5	SAR_IE	SAR-ADC 中断控制： 1：使能 ADC 中断输出； 0：不使能 ADC 中断输出。	R/W	0
4:3	SAR_PGA	SAR-ADC 增益控制； 00: 0.5 倍 01: 1 倍 10: 1.5 倍 11: 2 倍 新增支持 0.25 倍增益，详细见当 SAR_CTL1 寄存器 SAR_PGA_SEL[2:0]配置	R/W	0
2:0	SAR_CH	由{SAR_CH3,SARCH[2:0]}决定 SAR-ADC Channel 选择 0000: 温度测量 0001: VBAT (对于 3.6V 电池，1/2 分压后得到 1.8V，PGA	R/W	0

		采用 0.5 倍，测量输入 0.9V) 0010: 外部管脚输入 AIN0 0011: 外部管脚输入 AIN1 0100: 外部管脚输入 AIN2 0101: 外部管脚输入 AIN3 0110: 外部管脚输入 AIN4 0111: 外部管脚输入 AIN5 (新增) 1000: 外部管脚输入 AIN6 (新增) 0111: 保留 不管设置为哪个通道，自动温度测量的优先级最高。		
备注：以上寄存器位只有当 SAR_START 中的 ST=0 时才可写。				

12.2.2 SAR_START(0x4)

SAR-ADC 启动寄存器 地址 0x4002C000+ 0x4

比特位	名称	描述	读 / 写标志	复位值
31:01	---	只读，不可写	R	0
0	ST	SAR-ADC Start Bit 0: SAR-ADC 无操作 1: 启动一次 SAR-ADC 采样，完成采样后自动清零 Note: 1、由 RTC 控制的自动温度测量不受该位控制，并且优先级高于该配置位； 2、当该位 ADC_START 为 1 时，禁止软件再次写 1 启动 SAR-ADC 测量；上次 SAR-ADC 转换完成后即该位为 0 后 100us (即查到 0 后，再延时 100us)，才能启动新的 SAR-ADC 测量；建议超时等待时间=2*(REF_WAIT + SAR_WAIT) + 51ms。	R/W	0

12.2.3 SAR_STAT(0x8)

SAR-ADC 状态寄存器 地址 0x4002C000+ 0x8

比特位	名称	描述	读 / 写标志	复位值
31:02	---	只读，不可写	R	0
1	TPS_BUSY	自动温度测量忙 =1: 自动温度测量正在进行； =0: 自动温度测量没有进行。 当 TPS_BUSY 为=1 时，软件写 ADC_START 寄存器，硬件操作会在等待 TPS_BUSY 为 0 后再进行。	R	0
0	DREADY	ADC 转换完成标志位 0: ADC 转换结果未完成 1: ADC 转换结果已完成 Note: 1、写 1 清零；	R/W	0

		2、由 RTC 控制的自动温度测量不在该状态中指示;		
--	--	----------------------------	--	--

12.2.4 SAR_DAT(0xC)

ADC DAT 寄存器 地址 0x4002C000+ 0xC

比特位	名称	描述	读/写标志	复位值
15:10	---	只读, 不可写	R	0
9:0	SAR-DAT	ADC 转换结果	R	0

12.2.5 LVD_CTRL(0x10)

LVD 控制寄存器 地址 0x4002C000+ 0x10

比特位	名称	描述	读/写标志	复位值		
31:10	---	只读，不可写	R	0		
9	SWHBIE	切换到电池中断使能： =0：不使能中断； =1：使能中断；	R/W	0		
8	SWHMIE	切换到主电中断使能： =0：不使能中断； =1：使能中断；	R/W	0		
7	CMP2IE	比较器 2 中断使能： =0：不使能中断； =1：使能中断；	R/W	0		
6	CMP1IE	比较器 1 中断使能： =0：不使能中断； =1：使能中断；	R/W	0		
5	LVDIE	LVD 中断使能： =0：不使能中断； =1：使能中断；	R/W	0		
4	---	只读，不可写	R	0		
3:0	LVDS	LVDS 阈值电压设置：		R/W	0	
		LVDS[3:0]	Vi1(V)			Vih(V)
		0000	2.3			2.4
		0001	2.5			2.6
		0010	2.7			2.8
		0011	2.9			3.0
		0100	3.1			3.2
		0101	3.3			3.4
		0110	3.5			3.6
		0111	3.7			3.8
		1000	3.9			4.0
		1001	4.0			4.1
		1010	4.2			4.3

	1011	4.4	4.5
	1100	4.6	4.7
	1101	4.7	4.8
	111x(LVDIN)	1.25	1.45

注：当 LVDS[3:0]=4' b111x 时，检测外部 LVDIN 引脚输入电压。等于其他值时，检测 VCC 输入电压。

备注：LVD、比较器 1 和比较器 2 合并一个中断向量；电源切换单独一个中断向量；SAR-ADC 单独一个中断向量。

12.2.6 LVD_STAT(0x14)

LVD 状态寄存器 地址 0x4002C000+ 0x14

比特位	名称	描述	读/写标志	复位值
31:10	---	只读，不可写	R	0
9	SWHF	电源切换状态标志： =0：主电模式； =1：电池模式。只读	R	0
8	SWHBIF	切换到电池中断标志： =0：未产生中断；=1：产生中断； 当电源从主电切换到电池时产生中断，写 1 清零；	R/W	0
7	SWHMIF	切换到主电中断标志： =0：未产生中断；=1：产生中断； 当电源从电池切换到主电时产生中断，写 1 清零；	R/W	0
6	CMP2IIF	比较器 2 中断标志 =0：未产生中断；=1：产生中断； 当输入电压相对于阈值变低或者变高时产生中断，写 1 清零；	R/W	0
5	CMP1IIF	比较器 1 中断标志 =0：未产生中断；=1：产生中断； 当输入电压相对于阈值变低或者变高时产生中断，写 1 清零；	R/W	0
4	LVDIIF	LVD 中断标志 =0：未产生中断；=1：产生中断； 当输入电压相对于阈值变低或者变高时产生中断，写 1 清零；	R/W	0
3	---	只读，不可写	R	0
2	CMP2IIF	比较器 2 状态标志 =0：低于阈值； =1：高于阈值；	R	0
1	CMP1IIF	比较器 1 状态标志 =0：低于阈值； =1：高于阈值；	R	0
0	LVDIF	LVD 状态标志 =0：低于阈值； =1：高于阈值；	R	0

12.2.7 SAR_CTRL1(0x18) (新增)

SAR-ADC 增益控制寄存器 1

比特位	名称	描述	读 / 写标志	复位值
31:15	---	预留	R	0x0
14:8	SAR_CONVERT	预留，不能修改	R/W	0x29
7	--	预留	R	0
6:4	SAR_SAMPLE	采样周期时钟数量： 000: 1 clk 001: 2 clk 010: 3 clk 111: 8 clk	R/W	0x0
3	SAR_PGA_SEL	SAR-ADC 增益控制寄存器选择指示 =0，选择了 SAR_CTL[4:3]作为 SAR-ADC 的增益寄存器； =1，选择了 SAR_CTL1[2:0]作为 SAR-ADC 的增益寄存器；	R/W	0x0
2:0	SAR_PGA	SAR-ADC 增益控制位 000: 0.5 倍 001: 1 倍 010: 1.5 倍 011: 2 倍 1xx: 0.25 倍	R/W	0x0

12.2.8 SAR_DAT2(0x1C) (新增)

ADC DAT 寄存器 2

比特位	名称	描述	读/写标志	复位值
15:12	---	预留	R	0
11:0	SAR_DAT2	ADC 转换结果	R	0

12.2.9 SAR_CTRL3 (0x20) (新增)

SAR 控制寄存器 3

比特位	名称	描述	读/写标志	复位值
31:7	---	预留	R	0x0
14:7	SAR_NUM_CFG	SAR ADC 输出数据累加点数配置 =0, 1 点 =1, 2 点累加 ... =N, N+1 点累加 最大支持 256 点累加	R/W	0x0
6:0	SAR_CLK_DIV	SAR_CLK 接口时钟分频系数：（以 SAR 控制器工作时钟频率作为参考） =0: 2 分频 =1: 2 分频 ... =N: N+1 分频 SAR_CLK 时钟源头来自 RCM3.6M，经 14 分频后输出	R/W	0xf

		频率为 256K, 经过 SAR_CLK_DIV 分频后得到 SAR 采样时钟, 默认采样时钟 16KHZ。		
--	--	--	--	--

12.2.10 SAR_DAT3 (0x24) (新增)

ADC DAT 寄存器 3

比特位	名称	描述	读/写标志	复位值
31:16	---	预留	R	0
15:0	SAR_DAT3	ADC 累加结果, 结果处理如下表所示。	R	0

累加点数	SAR_DAT_SUM 右移位数	SAR_DAT3
$128 < (SAR_NUM_CFG + 1) \leq 256$	4	SAR_DAT_SUM[19:4]
$64 < (SAR_NUM_CFG + 1) \leq 128$	3	SAR_DAT_SUM[18:3]
$32 < (SAR_NUM_CFG + 1) \leq 64$	2	SAR_DAT_SUM[17:2]
$16 < (SAR_NUM_CFG + 1) \leq 32$	1	SAR_DAT_SUM[16:1]
$0 < (SAR_NUM_CFG + 1) \leq 16$	0	SAR_DAT_SUM[15:0]

12.3 ADC 电压检测步骤

- 1、配置系统控制章节模块使能 1 寄存器 MOD1_EN 第 11 位为 1, 开启 SAR_EN 时钟。
- 2、判断 SAR-ADC 状态寄存器 SAR_STATUS 第 1 位是否为 0, 没有进行自动温度测量, 如为 0 进入下一步, 如为 1 则等待。
- 3、配置 ADC 控制寄存器 SAR_CTL, 配置等待时间和增益, 通道选择对应的 ADC 通道。
- 4、SAR-ADC 启动寄存器 SAR_START 配置为 1, 启动 ADC 转换。
- 5、判断 SAR-ADC 状态寄存器 SAR_STATUS 第 0 位, 等待转换完成。
- 6、读取 ADC 转换数据寄存器 SAR_DAT。
- 7、计算: ADC 基准电压源为 1.25V, ADC 满度时 ADC DAT 寄存器值为 1024。计算公式即为 $(ADC_DAT * 1.25) / 1024$ 。当电压值超过 ADC 的满度, 转换值为 1024。
- 8、转换及计算完成。

12.4 VBAT 电压检测

- 1、配置系统控制章节模块使能 1 寄存器 MOD1_EN 第 11 位为 1, 开启 SAR_EN 时钟。
- 2、判断 SAR-ADC 状态寄存器 SAR_STATUS 第 1 位是否为 0, 没有进行自动温度测量, 如为 0 进入下一步, 如为 1 则等待。
- 3、配置 ADC 控制寄存器 SAR_CTL 通道选择为 VBAT, 配置等待时间;
- 4、SAR-ADC 启动寄存器 SAR_START 配置为 1, 启动 ADC 转换。
- 5、判断 SAR-ADC 状态寄存器 SAR_STATUS 第 0 位, 等待转换完成。
- 6、读取 ADC 转换数据寄存器 SAR_DAT。
- 7、计算: ADC 基准电压源为 1.25V, ADC 满度时 ADC DAT 寄存器值为 1024。3.6V 电池直接接入 VBAT 引脚, MADC->AD_CTRL 配置为 0x01, 此时增益为 0.5 倍, 则电压值为 $(ADC_DAT * 1.25 * 4) / 1024$, 其中因增益为 0.5 倍, 内部对 VBAT 接入有 1/2 分压, 所以实际电压需乘 4。
- 8、转换及计算完成。

12.5 低电压检测应用

- 1、对带电池的电能表，为节省电池功耗，需判断外部电源状态，当电源为低时让系统进入睡眠模式。
电源检测模块需一直开启，RN821x 的 CMP2/CMP1 比较器为极低功耗比较器，可用此比较器检测电源电压，并在电池模式下工作，CMP 输入管脚电压与基准 1.25V 进行比较，并产生中断及状态标志。
- 2、配置系统控制章节模块使能 1 寄存器 MOD1_EN 第 11 位为 1，开启 SAR_EN 时钟。
- 3、配置系统控制章节系统掉电控制寄存器 SYS_PD 的第 5 位置 0，开启 CMP2 电源。
- 4、配置 LVD 控制寄存器 LVD_CTL，使能 CMP2 中断。开启 CMP2 中断 NVIC_EnableIRQ(CMP_IRQn);
- 5、编写中断服务程序：

```
void CMP_HANDLER(void)
{
    if(!(MADC->LVD_STAT & 0x04))
    {
    }
    else
    {
    }
    MADC->LVD_STAT = 0x01ff;
}
```

因 LVD，CMP1，CMP2 使用同一中断服务程序，所以需根据 LVD_STAT 判断中断状态。

- 6、完成。

CMP 硬件滤波时间为 100 个系统时钟，建议软件在中断唤醒后，对 CMP 的状态再做确认和滤波。

13 GPIO（修改）

13.1 概述

- 包含 PA、PB、PC 三组 GPIO
- PA 口包含 5 个 P0 口、8 个 P1 口、8 个 P2 口、8 个 P3 口
- PB 口包含 8 个 P4 口、8 个 P5 口、8 个 P6 口、8 个 P7 口
- PC 口包含 8 个 P8 口、8 个 P9 口、4 个 P10 口、6 个 P11 口
- PD 口包含 8 个 P12 口、5 个 P13 口、8 个 P14 口
- GPIO 是 AHB 的外设
- 支持 bitband 操作；
- LCD 复用引脚做为 IO 口使用时为开漏结构
- P140 引脚输入/输出模式上拉配置均有效，芯片工作在 3.3V~5V 时，上拉电阻约 5K。其他引脚上拉配置，只在输入模式有效，芯片工作在 3.3V 时，上拉电阻约 70K；芯片工作在 5V 时，上拉电阻约 50K。下拉配置在输入/输出模式下均有效，下拉电阻阻值与上拉电阻一致。

13.2 寄存器描述

GPIO 寄存器基地址：

模块名	物理地址	映射地址
GPIO	0x50000000	0x50000000

GPIO 寄存器的偏移地址：

PMA	0x00H	PA 口模式寄存器（输入或者输出）
PA	0x04H	PA 口数据寄存器
PCA0	0x08H	PA 口复用寄存器 0
PCA1	0x0CH	PA 口复用寄存器 1
PUA	0x10H	PA 口上拉选择寄存器
PIMA	0x14H	PA 口输入模式配置寄存器
PIEA	0x18H	PA 口输入使能选择寄存器
PMB	0x1CH	PB 口模式寄存器（输入或者输出）
PB	0x20H	PB 口数据寄存器
PCB	0x24H	PB 口复用寄存器
PUB	0x28H	PUB 口上拉选择寄存器
PIMB	0x2CH	PB 口输入模式配置寄存器
PIEB	0x30H	PB 口输入使能选择寄存器
PMC	0x34H	PC 口模式寄存器（输入或者输出）
PC	0x38H	PC 口数据寄存器
PCC	0x3CH	PC 口复用寄存器
PUC	0x40H	PUC 口上拉选择寄存器
PIEC	0x44H	PC 口输入使能选择寄存器

PIMC	0x48H	PC 口输入模式配置寄存器
PCB2	0x4CH	PB 口复用寄存器 2
PMD	0x50	PD 口模式寄存器（输入或者输出）
PD	0x54	PD 口数据寄存器
PCD	0x58	PD 口复用寄存器
PUD	0x5C	PUD 口上拉选择寄存器
PCE	0x60H	SEGC0M 口复用寄存器
PASET	0x64H	PA 口数据置位寄存器，写 1 到该寄存器，PA 口对应位会被写 1；
PACLR	0x68H	PA 口数据清零寄存器，写 1 到该寄存器，PA 口对应位会被清零；
PBSET	0x6CH	PB 口数据置位寄存器，写 1 到该寄存器，PB 口对应位会被写 1；
PBCLR	0x70H	PB 口数据清零寄存器，写 1 到该寄存器，PB 口对应位会被清零；
PCSET	0x74H	PC 口数据置位寄存器，写 1 到该寄存器，PC 口对应位会被写 1；
PCCLR	0x78H	PC 口数据清零寄存器，写 1 到该寄存器，PC 口对应位会被清零；
PDSET	0x7C	PD 口数据置位寄存器，写 1 到该寄存器，PC 口对应位会被写 1；
PDCLR	0x80	PD 口数据清零寄存器，写 1 到该寄存器，PC 口对应位会被清零；
PIED	0x84	PD 口输入使能选择寄存器
PIMD	0x88	PD 口输入模式配置寄存器
PCA2	0x8C	PA 口复用寄存器 2
PCA3	0x90	PA 口复用寄存器 3
PCB3	0x94	PB 口复用寄存器 3
PCC2	0x98	PC 口复用寄存器 2
PCC3	0x98	PC 口复用寄存器 2
PCC4	0xA0	PC 口复用寄存器 2
PCC5	0xA4	PC 口复用寄存器 2
PCD2	0xA8	PD 口复用寄存器 2
PCD3	0xAC	PD 口复用寄存器 3
PIMA2	0xB0	PA 口输入模式配置寄存器 2
LURT_CFG	0x100	LPUART 配置寄存器
IOCFG	0x104	IO 驱动配置寄存器

注：IO 口类型见第管脚排列。

建议使用 bitband 功能访问 GPIO 的寄存器，便于对 IO 口的相关寄存器进行 bit 操作。

也可采用 SET/CLR 寄存器写 GPIO 的数据寄存器；

如果 IO 口复用配置选择为 IO 口之外的功能，模式寄存器、数据寄存器、输入使能寄存器无效。

上拉使能除 P140 外仅支持输入模式；P140 引脚输入/输出模式上拉配置均有效。

13.3 PA 口

13.3.1 PA 口模式寄存器 PMA（输入或者输出）（0x00）

比特位	名称	描述	读/写标志	复位值
31:24	PM37~PM30	=0 输出模式 =1 输入模式 PM37 和 PM36 只读，读出为 1，只能是输入模式；	R/W	FF

23:16	PM27~PM20	=0 输出模式 =1 输入模式	R/W	FF
15:8	PM17~PM10	=0 输出模式 =1 输入模式	R/W	FF
7:5	---	预留	R	0
4:0	PM04~PM00	=0 输出模式 =1 输入模式	R/W	1F

13.3.2 PA 口数据寄存器 PA (0x04)

比特位	名称	描述	读 / 写标志	复位值
31:30	P37~P36	P36 和 P37 数据输入寄存器，只读；	R	0
29:24	P35~P30	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00
23:16	P27~P20	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00
15:8	P17~P10	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00
7:5	---	预留	R	0
4:0	P04~P00	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值 如果定义为模拟输入，在输入模式读取到的值为 0.	R/W	00

13.3.3 PA 口复用 0 寄存器 PCA0 (0x08)

当选择为模拟输入口时，自动选择为输入模式，PMA 寄存器无效。

比特位	名称	描述	读 / 写标志	复位值
31:30	UART2_SEL	由 {PCA3[28],PCA0[31:30]} 共同定义那一路 IO 选择为 UART2 的配置： =000: P24/P25 选择为 UART2，具体定义见 PCA0[27] =001: P00/P01 选择为 UART2，具体定义见 PCA0[3:0] =010: P20/P21 选择为 UART2，具体定义见 PCA0[25] =011: P22/P23 选择为 UART2，具体定义见 PCA0[26] =100: P26/P27 选择为 UART2，具体定义见 PCA0[28] =其他: 保留	R/W	0
29	SWD_SEL	=0: P24 和 P25 不选择为 SWD，由 PC245 (bit27) 定义； =1: P24 和 P25 选择为 SWD；	R/W	1
28	PC267	定义端口 P26 和 P27 复用配置： =0: 选择为 IO 口； =1: 选择为 UART3 接口。	R/W	00

		注：当 PCA3 寄存器的 bit[27]=0 时，该配置有效，向下兼容； 当 PCA3 寄存器的 bit[27]=1 时，由 PCA3 寄存器的 bit[26:21] 决定 P26/P27 的复用。		
27	PC245	定义端口 P24 和 P25 复用配置： =0：选择为 IO 口； =1：选择为 UART2 接口。 注：当 PCA3 寄存器的 bit[20]=0 时，该配置有效，向下兼容； 当 PCA3 寄存器的 bit[20]=1 时，由 PCA3 寄存器的 bit[19:14] 决定 P24/P25 的复用。	R/W	00
26	PC223	定义端口 P22 和 P23 复用配置： =0：选择为 IO 口； =1：选择为 UART1 接口。 注：当 PCA3 寄存器的 bit[13]=0 时，该位配置有效，向下兼容； 当 PCA3 寄存器的 bit[13]=1 时，由 PCA3 寄存器的 bit[12:7] 决定 P22/P23 的复用。	R/W	00
25	PC201	定义端口 P20 和 P21 复用配置： =0：选择为 IO 口； =1：选择为 UART0 接口。 注：当 PCA3 寄存器的 bit[6]=0 时，该配置有效，向下兼容； 当 PCA3 寄存器的 bit[6]=1 时，由 PCA3 寄存器的 bit[5:0] 决定 P20/P21 的复用。	R/W	00
24:23	PC17	由 {PCA2[30:29],PCA0[24:23]} 定义端口 P17 复用配置： =0000：选择为 IO 口 =0001：选择为 KEY7 =0010：选择为 TC1_P[1] =0011：选择为 TCIN，对 TC0/TC1 可同时有效 =0100：保留 =0101：选择为 TRIG_OUT(智能微断需求) =0110：选择为模拟输入 AIN5 =其它：保留 默认向下兼容	R/W	00
22:21	PC16	由 {PCA2[28:27],PCA0[22:21]} 定义端口 P16 复用配置： =0000：选择为 IO 口 =0001：选择为 KEY6 =0010：选择为 TC1_N[1] =0011：选择为 TCIN，对 TC0/TC1 可同时有效 =0100：保留 =0101：选择为 TRIG_OUT(智能微断需求) =其它：保留 默认向下兼容	R/W	00
20:19	PC15	由 {PCA2[26:25],PCA0[20:19]} 定义端口 P15 复用配置： =0000：选择为 IO 口 =0001：选择为 KEY5 =0010：选择为 TC1_P[0]	R/W	00

		=0011: 选择为 TCIN, 对 TC0/TC1 可同时有效 =0100: 保留 =0101: 选择为 INT7 =其它: 保留 注: 当 P37 和 P15 都选择为 INT7 时, P37 优先级高于 P15, P15 输入无效 默认向下兼容		
18:17	PC14	由 {PCA2[24:23],PCA0[18:17]} 定义端口 P14 复用配置: =0000: 选择为 IO 口 =0001: 选择为 KEY4 =0010: 选择为 TC1_N[0] =0011: 选择为 TCIN, 对 TC0/TC1 可同时有效 =0100: 保留 =0101: 选择为 INT6 =其它: 保留 注: 当 P36 和 P14 都选择为 INT7 时, P36 优先级高于 P14, P14 输入无效 默认向下兼容	R/W	00
16:15	PC13	由 {PCA2[22:21],PCA0[16:15]} 定义端口 P13 复用配置: =0000: 选择为 IO 口 =0001: 选择为 KEY3 =0010: 选择为 TC0_P[1] =0011: 选择为 TCIN, 对 TC0/TC1 可同时有效 =0100: 选择为 IOCNT_OUT2 =0101: 选择为 ADC_CLKO =0110: 选择为 IA_IN =0111: 保留 其他: 保留 默认向下兼容	R/W	00
14:13	PC12	由 {PCA2[20:19],PCA0[14:13]} 定义端口 P12 复用配置: =0000: 选择为 IO 口 =0001: 选择为 KEY2 =0010: 选择为 TC0_N[1] =0011: 选择为 TCIN, 对 TC0/TC1 可同时有效 =0100: 选择为 IB_IN =0101: 保留 =0110: 选择为 CF_OUT4 =0111: 选择为 D2F_OUT4 =1000: 选择为 IOCNT_OUT4 =1001: 选择为 TRIG_OUT(智能微断需求) =其他: 保留 默认向下兼容	R/W	00
12:11	PC11	由 {PCA2[18:17],PCA0[12:10]} 定义端口 P11 复用配置: =0000: 选择为 IO 口	R/W	00

		=0001: 选择为 KEY1 =0010: 选择为 TC0_P[0] =0011: 选择为 TCIN, 对 TC0/TC1 可同时有效 =0100: 选择为 IA_IN =0101: 保留 =0110: 选择为 CF_OUT3 =0111: 选择为 D2F_OUT3 =1000: 选择为 IOCNT_OUT3 =1001: 选择为 TC1_P[1] =其他: 保留 默认向下兼容		
10:9	PC10	由 {PCA2[16:15],PCA0[10:9]} 定义端口 P10 复用配置: =0000: 选择为 IO 口 =0001: 选择为 KEY0 =0010: 选择为 TC0_N[0] =0011: 选择为 TCIN, 对 TC0/TC1 可同时有效 =0100: 选择为 IA_IN =0101: 选择为 IB_IN =0110: 选择为 CF_OUT2 =0111: 选择为 D2F_OUT2 =1000: 选择为 IOCNT_OUT2 =1001: 选择为 TC1_N[1] =其他: 保留 默认向下兼容	R/W	00
8	KEY 4 _SEL	具体定义见 PCA0 寄存器的 bit[6]	R/W	0
7	---	预留	R/W	0
6	PC04	由 {PCA2[11:10],PCA0[8],PCA0[6]} 定义端口 P04 复用配置: =0000: 选择为 IO 口 =0001: 选择为模拟输入 AIN4/LVDIN =001x: P04 选择为 KEY4 =0100: 选择为 IA_IN =0101: 选择为 IB_IN =其他: 保留 注: 当 P14 和 P04 都选择为 KEY4 时, P14 优先级高于 P04, P04 输入无效 默认向下兼容	R/W	0
5	PC03	由 {PCA2[9:7],PCA0[5]} 定义端口 P03 的复用配置: =0000: 选择为 IO 口 =0001: 选择为模拟输入 AIN3/CMP2 =0010: 选择为 IA_IN =0011: 保留 =其他: 保留		
4	PC02	由 {PCA2[6:4],PCA0[4]} 定义端口 P02 的复用配置: =0000: 选择为 IO 口		

		=0001: 选择为模拟输入口 AIN2/CMP1 =0010: 选择为 IA_IN =0011: 选择为 IB_IN =其他: 保留		
3:2	PC01	由 {PCA2[3:2],PCA0[3:2]} 定义端口 P01 复用配置: =0000: 选择为 IO 口 =0001: 选择为模拟输入口 AIN1 =0010: 选择为 KEY3 =0011: 选择为 TX2 =0100: 选择为 IA_IN =0101: 保留 =0110: 选择为 TC1_N[1] =其他: 保留 注: 当 P13 选择为 KEY3 时, P13 优先级高于 P01, P01 输入无效	R/W	0
1:0	PC00	由 {PCA2[1:0],PCA0[1:0]} 定义端口 P00 复用配置: =0000: 选择为 IO 口 =0001: 选择为模拟输入口 AIN0 =0010: 选择为 KEY2 =0011: 选择为 RX2 =0100: 选择为 IA_IN =0101: 选择为 IB_IN =0110: 选择为 TC1_P[1] =其他: 保留 注: 当 P12 选择为 KEY2 时, P12 优先级高于 P00, P00 输入无效	R/W	0

13.3.4 PA 口复用 1 寄存器 PCA1 (0x0C)

比特位	名称	描述	读 / 写标志	复位值
31:30	PC37_2	具体定义见 PCA1 寄存器的 bit [15:14]	R/W	0
29:28	PC36_2	具体定义见 PCA1 寄存器的 bit [13:12]	R/W	0
27:26	PC35_2	具体定义见 PCA1 寄存器的 bit [11:10]	R/W	0
25:24	PC34_2	具体定义见 PCA1 寄存器的 bit [9:8]	R/W	0
23:22	PC33_2	具体定义见 PCA1 寄存器的 bit [7:6]	R/W	0
21:20	PC32_2	具体定义见 PCA1 寄存器的 bit [5:4]	R/W	0
19:18	PC31_2	具体定义见 PCA1 寄存器的 bit [3:2]	R/W	0
17:16	PC30_2	具体定义见 PCA1 寄存器的 bit [1:0]	R/W	0
15:14	PC37[1:0]	由 {PCA1[31:30],PCA1[15:14]} 定义端口 P37 复用配置: =0000: 选择为 IO 口 =0001: 选择为外部中断 INT7 =001x: 选择为高频晶体管脚 HOSCI =0100: 保留 =其他: 保留	R/W	0

		备注：只要 PC36[1]和 PC37[1]中任何一位为高，那么就选择为 HOSC		
13:12	PC36[1:0]	{PCA1[29:28],PCA1[13:12]} 定义端口 P36 复用配置： =0000：选择为 IO 口 =0001：选择为外部中断 INT6 =001x：选择为高频晶体管脚 HOSCO =0100：保留 =其他：保留 备注：只要 PC36[1]和 PC37[1]中任何一位为高，那么就选择为 HOSC	R/W	0
11:10	PC35[1:0]	{PCA1[27:26],PCA1[11:10]} 定义端口 P35 复用配置： =0000：选择为 IO 口 =0001：选择为外部中断输入口 INT5 =0010：选择为 TCIN =0011：选择为 D2F_OUT4 =0100：选择为 IOCNT_OUT4 =0101：选择为 ADC_CLKO =0110：选择为 IA_IN =0111：选择为 IB_IN =1000：选择为 TRIG_OUT =1001：选择为 CF_OUT4 =其他：保留	R/W	0
9:8	PC34[1:0]	{PCA1[25:24],PCA1[9:8]} 定义端口 P34 复用配置： =0000：选择为 IO 口 =0001：选择为外部中断输入口 INT4 =0010：选择为视在电能脉冲输出 SF_OUT =0011：选择为 D2F_OUT3 =0100：选择为 CF_OUT3 =0101：选择为 IOCNT_OUT3 =0110：选择为 ADC_CLKO =0111：选择为 IA_IN =1000：选择为 TRIG_OUT =其他：保留	R/W	0
7:6	PC33[1:0]	{PCA1[23:22],PCA1[7:6]} 定义端口 P33 复用配置： =0000：选择为 IO 口 =0001：选择为外部中断输入口 INT3 =0010：选择为 TCIN =0011：选择为 ADC_CLKO =0100：选择为 IOCNT_OUT3 =0101：选择为 IA_IN =0110：选择为 IB_IN =其他：保留	R/W	0
5:4	PC32[1:0]	{PCA1[21:20],PCA1[5:4]} 定义端口 P32 复用配置： =0000：选择为 IO 口	R/W	10

		=0001: 选择为外部中断输入口 INT2 =0010: 选择为 RTC 输出 RTC_OUT (默认选择为 RTC 输出) =0011: 选择为 KEY5 =0100: 选择为 ADC_CLKO =0101: 选择为 TRIG_OUT =0110: 选择为 CF_OUT1 =0111: 选择为 D2F_OUT1 =1000: 选择为 TC1_N[1] =1001: 选择为万年历秒输出 RTC1S =其他: 保留		
3:2	PC31[1:0]	{PCA1[19:18],PCA1[3:2]} 定义端口 P31 复用配置: =0000: 选择为 IO 口 =0001: 选择为外部中断输入口 INT1 =0010: 选择为 TCIN =0011: 选择为 RX4 =0100: 选择为 RTC_OUT =0101: 选择为 ADC_CLKO =0110: 选择为 TRIG_OUT =0111: 保留 =1000: 选择为 CF_OUT0 =1001: 选择为 D2F_OUT0 =1010: 选择为 TC1_P[1] =1011: 选择为万年历秒输出 RTC1S =其他: 保留	R/W	0
1:0	PC30[1:0]	由 {PCA1[17:16],PCA1[1:0]} 定义端口 P30 复用配置: =0000: 选择为 IO 口 =0001: 选择为外部中断输入口 INT0 =0010: 选择为 TCIN =0011: 选择为 TX4 =0100: 选择为 RTC_OUT =0101: 选择为 ADC_CLKO =0110: 选择为 TRIG_OUT =0111: 保留 =1000: 选择为万年历秒输出 RTC1S =其他: 保留	R/W	0

13.3.5 PA 口复用寄存器 2 PCA2 (0x8C) (新增)

比特位	名称	描述	读/写标志	复位值
31	---	保留	R	0
30:29	PC17_2	具体定义见 PCA0 寄存器的 bit[24:23]	R/W	0
28:27	PC16_2	具体定义见 PCA0 寄存器的 bit[22:21]	R/W	0
26:25	PC15_2	具体定义见 PCA0 寄存器的 bit[20:19]	R/W	0
24:23	PC14_2	具体定义见 PCA0 寄存器的 bit[18:17]	R/W	0

22:21	PC13_2	具体定义见 PCA0 寄存器的 bit[16:15]	R/W	0
20:19	PC12_2	具体定义见 PCA0 寄存器的 bit[14:13]	R/W	0
18:17	PC11_2	具体定义见 PCA0 寄存器的 bit[12:11]	R/W	0
16:15	PC10_2	具体定义见 PCA0 寄存器的 bit[10:9]	R/W	0
14:12	PC05_2	具体定义见 PCA0 寄存器的 bit[7]	R/W	0
11:10	PC04_2	具体定义见 PCA0 寄存器的 bit[6]	R/W	0
9:7	PC03_2	具体定义见 PCA0 寄存器的 bit[5]	R/W	0
6:4	PC02_2	具体定义见 PCA0 寄存器的 bit[4]	R/W	0
3:2	PC01_2	具体定义见 PCA0 寄存器的 bit[3:2]	R/W	0
1:0	PC00_2	具体定义见 PCA0 寄存器的 bit[1:0]	R/W	0

13.3.6 PA 口复用寄存器 3 PCA3 (0x90) (新增)

比特位	名称	描述	读/写标志	复位值
31:29	---	预留	R	0
28	UART2_SEL_2	具体定义见 PCA0 寄存器的 bit[31:30]	R/W	0
27	PC267_2	=0: P26/P27 具体定义由 PCA0 寄存器的 bit[28]决定 =1: P26/P27 具体定义由 PCA3 寄存器的 bit[26:21]决定	R/W	0
26:24	PC27_2	定义端口 P27 复用配置 2: =000: 选择为 GPIO =001: 选择为 TX3 =010: 选择为 TX2 =011: 选择为 TRIG_OUT =100: 保留 =101: 选择为 TC1_N[1] =其他: 保留	R/W	0
23:21	PC26_2	定义端口 P26 复用配置 2: =000: 选择为 GPIO =001: 选择为 RX3 =010: 选择为 RX2 =011: 选择为 TRIG_OUT =100: 保留 =101: 选择为 TC1_P[1] =其他: 保留	R/W	0
20	PC245_2	=0: P24/P25 具体定义由 PCA0 寄存器的 bit[27]决定 =1: P24/P25 具体定义由 PCA3 寄存器的 bit[19:14]决定	R/W	0
19:17	PC25_2	定义端口 P25 复用配置 2: =000: 选择为 GPIO =001: 选择为 TX2 =010: 保留 =011: 选择为 TC1_N[1] =其他: 保留	R/W	0
16:14	PC24_2	定义端口 P24 复用配置 2: =000: 选择为 GPIO =001: 选择为 RX2	R/W	0

		=010: 保留 =011: 选择为 TC1_P[1] =其他: 保留		
13	PC223_2	=0: P22/P23 具体定义由 PCA0 寄存器的 bit[26]决定 =1: P22/P23 具体定义由 PCA3 寄存器的 bit[12:7]决定	R/W	0
12:10	PC23_2	定义端口 P23 复用配置 2: =000: 选择为 GPIO =001: 选择为 TX1 =010: 选择为 TX2 =011: 选择为 SPI1_MOSI =100: 选择为 SPI3_MOSI =101: 保留 =其他: 保留	R/W	0
9:7	PC22_2	定义端口 P22 复用配置 2: =000: 选择为 GPIO =001: 选择为 RX1 =010: 选择为 RX2 =011: 选择为 SPI1_MISO =100: 选择为 SPI3_MISO =101: 保留 =110: 选择为 TRIG_OUT =其他: 保留	R/W	0
6	PC201_2	=0: P20/P21 具体定义由 PCA0 寄存器的 bit[25]决定 =1: P20/P21 具体定义由 PCA3 寄存器的 bit[5:0]决定	R/W	0
5:3	PC21_2	定义端口 P21 复用配置 2: =000: 选择为 GPIO =001: 选择为 TX0 =010: 选择为 TX2 =011: 选择为 SPI1_SCLK =100: 选择为 SPI3_SCLK =101: 保留 =110: 选择为 TC1_P[1] =其他: 保留	R/W	0
2:0	PC20_2	定义端口 P20 复用配置 2: =000: 选择为 GPIO =001: 选择为 RX0 =010: 选择为 RX2 =011: 选择为 SPI1_SCSN =100: 选择为 SPI3_SCSN =101: 保留 =110: 选择为 TC1_N[1] =111: 选择为 TRIG_OUT	R/W	0

13.3.7 PA 口上拉选择寄存器 PUA (0x10)

备注：当 IO 口处于输出模式或者模拟 PAD 模式时，不管 PU 寄存器如何配置，PIN 上拉均不使能。

比特位	名称	描述	读 / 写标志	复位值
31:24	PU37~PU30	定义端口上拉配置： =0：不选择上拉； =1：选择上拉； P30 默认上拉开启	R/W	00
23:16	PU27~PU20	定义端口上拉配置： =0：不选择上拉； =1：选择上拉； 备注：P24 和 P25 作为 SWD 默认上拉使能。	R/W	30
15:8	PU17~PU10	定义端口上拉配置： =0：不选择上拉； =1：选择上拉；	R/W	00
7:5	--	保留	R	0
4:0	PU04~PU00	定义端口上拉配置： =0：不选择上拉； =1：选择上拉；	R/W	00H

13.3.8 PA 口输入模式配置寄存器 PIMA (0x14)

比特位	名称	描述	读 / 写标志	复位值
31:24	PIL27~PIL20	定义端口 P20~P27 输入缓冲器类型： =0：CMOS 缓冲器， $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$ ； =1：TTL 缓冲器， $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$ ；	R/W	00
23:16	PIL17~PIL10	定义端口 P10~P17 输入缓冲器类型： =0：CMOS 缓冲器， $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$ ； =1：TTL 缓冲器， $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$ ；	R/W	00
15:8	PID27~PID20	定义端口 P20~P27 是否是 N-ch 漏极开路输出： =0：普通模式； =1：N-ch 漏极开路模式；	R/W	00
7:0	PID17~PID10	定义端口 P10~P17 是否是 N-ch 漏极开路输出： =0：普通模式； =1：N-ch 漏极开路模式；	R/W	00

13.3.9 PA 口输入模式配置 PIMA2 (0xB0) (新增)

比特位	名称	描述	读/写标志	复位值
31:16	--	保留	R	0
15:8	PIL37~PIL30	定义端口 P30~P37 输入缓冲器类型： =0：CMOS 缓冲器， $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$ ； =1：TTL 缓冲器， $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$ ；	R/W	00
7:0	PID37~PID30	定义端口 P30~P37 是否是 N-ch 漏极开路输出：	R/W	00

		=0: 普通模式; =1: N-ch 漏极开路模式;		
--	--	-------------------------------	--	--

13.3.10 PA 口输入使能寄存器 PIEA (0x18)

比特位	名称	描述	读 / 写 标志	复 位 值
31:24	PIE37~PIE30	输入使能: =1: 不使能输入; =0: 使能输入; 注: P30 上电后需要 BOOTROM 设为输入使能, 方便 ISP 进行检测。	R/W	FF
23:16	PIE27~PIE20	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF
15:8	PIE17~PIE10	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF
7:5	保留	保留	R	0
4:0	PIE04~PIE00	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	3F

13.4 PB 口

13.4.1 PB 口模式寄存器 PMB (输入或者输出) (0x1C)

比特位	名称	描述	读 / 写 标志	复 位 值
31:24	PM77~PM70	=0 输出模式 =1 输入模式	R/W	FF
23:16	PM67~PM60	=0 输出模式 =1 输入模式	R/W	FF
15:8	PM57~PM50	=0 输出模式 =1 输入模式	R/W	FF
7:0	PM47~PM40	=0 输出模式 =1 输入模式	R/W	FF

当 IO 口设置为 7816 口或者 SPI 口时, 方向寄存器不起作用, 由通信模块本身控制。

13.4.2 PB 口数据寄存器 PB (0x20)

比特位	名称	描述	读 / 写 标志	复 位 值
31:24	P77~P70	定义芯片端口需要输出的数据。	R/W	00

		如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值		
23:16	P67~P60	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00
15:8	P57~P50	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00
7:0	P47~P40	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00

13.4.3 PB 口复用寄存器 PCB (0x24)

比特位	名称	描述	读/写标志	复位值
31:24	PC77~PC70	PC77~PC70 定义端口复用配置： =0：选择为 IO 口 =1：选择为 SEG	R/W	00
23:16	PC67~PC60	PC67~PC60 定义端口复用配置： =0：选择为 IO 口 =1：选择为 SEG	R/W	00
15	PC57	由 {PCB3[15],PCB2[31:30],PCB[15]} 定义端口 P57 复用配置： =0000：选择为 IO 口 =0001：选择为视在脉冲输出 SF =0010：选择为 TCIN =0011：选择为 CF_OUT2 =0100：选择为 TC1_P[1] =0101：选择为 IOCNT_OUT2 =0110：选择为 D2F_OUT2 =0111：选择为 ADC_CLKO =其他：保留	R/W	0
14	PC56	由 {PCB3[14],PCB2[29:28],PCB[14]} 定义端口 P56 复用配置： =0000：选择为 IO 口 =0001：选择为过零输出 ZX_OUT =0010：选择为 TCIN =0011：选择为 INT2 =0100：选择为 TC1_N[1] =0101：选择为 RTC_OUT =0110：选择为 IOCNT_OUT0 =0111：选择为 ADC_CLKO =1000：选择为万年历秒输出 RTC1S =其他：保留 注：当 P32 和 P56 都选择为 INT2 时，P32 优先级高于 P56，P56 输入无效	R/W	0
13	PC55	由 {PCB3[13],PCB2[27:26],PCB[13]} 定义端口 P55 复用配置：	R/W	0

		=0000: 选择为 IO 口 =0001: 选择为 TX5 =0010: 选择为 TCIN =0011: 选择为 IA_IN =0100: 选择为 TC1_P[0] =0101: 选择为 IB_IN =0110: 选择为 SPI1_MOSI =0111: 保留 =其他: 保留		
12	PC54	由 {PCB3[12],PCB2[25:24],PCB[12]} 定义端口 P54 复用配置: =0000: 选择为 IO 口 =0001: 选择为 RX5 =0010: 选择为 TCIN =0011: 选择为 IA_IN =0100: 选择为 TC1_N[0] =0101: 保留 =0110: 选择为 SPI1_MISO =0111: 保留 =其他: 保留	R/W	0
11	PC53	由 {PCB3[11],PCB2[23:22],PCB[11]} 定义端口 P53 复用配置: =0000: 选择为 IO 口 =0001: 选择为 I2C_SDA =0010: 选择为 TCIN =0011: 选择为 IA_IN =0100: 选择为 TC0_P[1] =0101: 选择为 IB_IN =0110: 选择为 SPI1_SCLK =0111: 保留 注: 当 P53 复用为 I2C_SDA、IA_IN 或 IB_IN 时, 默认为开漏输出模式, 可通过 PIMB 寄存器 bit11 配置为普通模式。配置为其他复用, 默认为普通模式。 管脚配置顺序, 复用配置, 模式配置。	R/W	0
10	PC52	由 {PCB3[10],PCB2[21:20],PCB[10]} 定义端口 P52 复用配置: =0000: 选择为 IO 口 =0001: 选择为 I2C_SCL =0010: 选择为 TCIN =0011: 选择为 ADC_CLKO =0100: 选择为 TC0_N[1] =0101: 保留 =0110: 选择为 SPI1_SCSN =0111: 保留 注 1: PC52 仅在 SPI 从模式下有效, 在 SPI 主模式下不要配置为 1。 注 2: 当 P52 复用为 I2C_SCL 或 ADC_CLKO 时, 默认为开漏输	R/W	0

		出模式，可通过 PIMB 寄存器 bit10 配置为普通模式。配置为其他复用时，默认为普通模式。 该管脚配置顺序，复用配置，模式配置。		
9	PC51	由 {PCB3[9],PCB2[19:18],PCB[9]} 定义端口 P51 复用配置： =0000：选择为 IO 口 =0001：选择为无功脉冲输出 QF =0010：选择为 RTC_OUT =0011：选择为 CF_OUT1 =0100：选择为有功脉冲输出 PF =0101：选择为 IOCNT_OUT1 =0110：选择为视在脉冲输出 SF =0111：选择为 D2F_OUT1 =1000：选择为万年历秒输出 RTC1S	R/W	1
8	PC50	由 {PCB3[8],PCB2[17:16],PCB[8]} 定义端口 P50 复用配置： =0000：选择为 IO 口 =0001：选择为有功脉冲输出 PF =0010：选择为 RTC_OUT =0011：选择为 CF_OUT0 =0100：选择为视在脉冲输出 SF =0101：选择为 IOCNT_OUT0 =0110：选择为无功脉冲输出 QF =0111：选择为 D2F_OUT0 =1000：选择为万年历秒输出 RTC1S	R/W	1
7	PC47	由 {PCB3[7],PCB2[15:14],PCB[7]} 定义端口 P47 复用配置： =0000：选择为 IO 口 =0001：选择为 SPI0_MOSI =0010：选择为 TX4 =0011：保留 =0100：选择为 SPI3_MOSI =其他：保留	R/W	00
6	PC46	由 {PCB3[6],PCB2[13:12],PCB[6]} 定义端口 P46 复用配置： =0000：选择为 IO 口 =0001：选择为 SPI0_MISO =0010：选择为 RX4 =0011：保留 =0100：选择为 SPI3_MISO =其他：保留	R/W	00
5	PC45	由 {PCB3[5],PCB2[11:10],PCB[5]} 定义端口 P45 复用配置： =0000：选择为 IO 口 =0001：选择为 SPI0_SCLK =0010：选择为 KEY7 =0011：选择为 ADC_CLKO =其他：保留 注：当 P17 和 P45 都选择为 KEY7 时，P17 优先级高于 P45，P45	R/W	00

		输入无效		
4	PC44	<p>由 {PCB3[4],PCB2[9:8],PCB[4]} 定义端口 P44 复用配置:</p> <p>=0000: 选择为 IO 口</p> <p>=0001: 选择为 SPI0_SCSN</p> <p>=0010: 选择为 KEY6</p> <p>=0011: 选择为 ADC_CLKO</p> <p>=其他: 保留</p> <p>注 1: PC44 仅在 SPI 从模式下有效, 在 SPI 主模式下不要配置为 1。</p> <p>注 2: 当 P16 和 P44 都选择为 KEY6 时, P16 优先级高于 P44, P44 输入无效</p> <p>p44-47 靠近模拟, 原则上不建议用为 spi 功能, 减少对计量影响, 建议使用模拟对边的 gpio 做为 spi。</p>	R/W	00
3	PC43	<p>由 {PCB3[3],PCB2[7:6],PCB[3]} 定义端口 P43 复用配置:</p> <p>=0000: 选择为 IO 口</p> <p>=0001: 选择为 78161_I</p> <p>=0010: 选择为 INT5</p> <p>=0011: 选择为 SPI2_MOSI</p> <p>=0100: 选择为 SPI1_MOSI</p> <p>=0101: 选择为 IA_IN</p> <p>=0110: 选择为 IB_IN</p> <p>=0111: 保留</p> <p>注: 当 P35 和 P43 都选择为 INT5 时, P35 优先级高于 P43, P43 输入无效</p>	R/W	0
2	PC42	<p>由 {PCB3[2],PCB2[5:4],PCB[2]} 定义端口 P42 复用配置:</p> <p>=0000: 选择为 IO 口</p> <p>=0001: 78161_IO</p> <p>=0010: 选择为 INT4</p> <p>=0011: 选择为 SPI2_MISO</p> <p>=0100: 选择为 SPI1_MISO</p> <p>=0101: 选择为 IA_IN</p> <p>=0110: 保留</p> <p>=0111: 保留</p> <p>注: 当 P34 和 P42 都选择为 INT4 时, P34 优先级高于 P42, P42 输入无效</p>	R/W	0
1	PC41	<p>由 {PCB3[1],PCB2[3:2],PCB[1]} 定义端口 P41 复用配置:</p> <p>=0000: 选择为 IO 口</p> <p>=0001: 选择为 78160_IO</p> <p>=0010: 选择为 INT3</p> <p>=0011: 选择为 SPI2_SCLK</p> <p>=0100: 选择为 SPI1_SCLK</p> <p>=0101: 选择为 IA_IN</p> <p>=0110: 选择为 IB_IN</p> <p>=0111: 保留</p>	R/W	0

		注：当 P33 和 P41 都选择为 INT3 时，P33 优先级高于 P41，P41 输入无效		
0	PC40	<p>由 {PCB3[0],PCB2[1:0],PCB[0]} 定义端口 P40 复用配置：</p> <p>=0000：选择为 IO 口</p> <p>=0001：选择为 7816_CLK</p> <p>=0010：选择为 INT1</p> <p>=0011：选择为 SPI2_SCSN</p> <p>=0100：选择为 SPI1_SCSN</p> <p>=0101：选择为 ADC_CLKO</p> <p>=其他：保留</p> <p>注 1：PC40 仅在 SPI 从模式下有效，在 SPI 主模式下不要配置为 1。</p> <p>注 2：当 P31 和 P40 都选择为 INT1 时，P31 优先级高于 P40，P40 输入无效</p>		0

13.4.4 PB 口复用寄存器 2 PCB2 (0x4C) (修改)

比特位	名称	描述	读/写标志	复位值
31:30	PC57_2	具体定义见 PCB 寄存器的 bit [15]	R/W	00
29:28	PC56_2	具体定义见 PCB 寄存器的 bit [14]	R/W	00
27:26	PC55_2	具体定义见 PCB 寄存器的 bit [13]	R/W	00
25:24	PC54_2	具体定义见 PCB 寄存器的 bit [12]	R/W	00
23:22	PC53_2	具体定义见 PCB 寄存器的 bit [11]	R/W	00
21:20	PC52_2	具体定义见 PCB 寄存器的 bit [10]	R/W	00
19:18	PC51_2	具体定义见 PCB 寄存器的 bit [9]	R/W	00
17:16	PC50_2	具体定义见 PCB 寄存器的 bit [8]	R/W	00
15:14	PC47_2	具体定义见 PCB 寄存器的 bit [7]	R/W	00
13:12	PC46_2	具体定义见 PCB 寄存器的 bit [6]	R/W	00
11:10	PC45_2	具体定义见 PCB 寄存器的 bit [5]	R/W	00
9:8	PC44_2	具体定义见 PCB 寄存器的 bit [4]	R/W	00
7:6	PC43_2	具体定义见 PCB 寄存器的 bit [3]	R/W	00
5:4	PC42_2	具体定义见 PCB 寄存器的 bit [2]	R/W	00
3:2	PC41_2	具体定义见 PCB 寄存器的 bit [1]	R/W	00
1:0	PC40_2	具体定义见 PCB 寄存器的 bit [0]	R/W	00

13.4.5 PB 口复用寄存器 3 PCB3 (0x94) (新增)

比特位	名称	描述	读/写标志	复位值
31:16	--	保留	R	0
15	PC57_3	具体定义见 PCB 寄存器的 bit [15]	R/W	
14	PC56_3	具体定义见 PCB 寄存器的 bit [14]	R/W	0
13	PC55_3	具体定义见 PCB 寄存器的 bit [13]	R/W	0
12	PC54_3	具体定义见 PCB 寄存器的 bit [12]	R/W	0
11	PC53_3	具体定义见 PCB 寄存器的 bit [11]	R/W	0
10	PC52_3	具体定义见 PCB 寄存器的 bit [10]	R/W	0

9	PC51_3	具体定义见 PCB 寄存器的 bit [9]	R/W	0
8	PC50_3	具体定义见 PCB 寄存器的 bit [8]	R/W	0
7	PC47_3	具体定义见 PCB 寄存器的 bit [7]	R/W	0
6	PC46_3	具体定义见 PCB 寄存器的 bit [6]	R/W	0
5	PC45_3	具体定义见 PCB 寄存器的 bit [5]	R/W	0
4	PC44_3	具体定义见 PCB 寄存器的 bit [4]	R/W	0
3	PC43_3	具体定义见 PCB 寄存器的 bit [3]	R/W	0
2	PC42_3	具体定义见 PCB 寄存器的 bit [2]	R/W	0
1	PC41_3	具体定义见 PCB 寄存器的 bit [1]	R/W	0
0	PC40_3	具体定义见 PCB 寄存器的 bit[0]	R/W	0

13.4.6 PB 口上下拉选择寄存器 PUB (0x28)

比特位	名称	描述	读 / 写 标志	复 位 值
31:24	PU77~PU70	PU77~PU70 定义 P7 端口是否内接下拉： =0: 不接下拉； =1: 内接下拉。	R/W	00
23:16	PU67~PU60	PU67~PU60 定义 P6 端口是否内接下拉： =0: 不接下拉； =1: 内接下拉。	R/W	00
15:8	PU57~PU50	PU57~PU50 定义 P5 端口是否内接上拉： =0: 不接上拉； =1: 内接上拉。	R/W	00
7:0	PU47~PU40	PU47~PU40 定义 P4 端口是否内接上拉： =0: 不接上拉； =1: 内接上拉。	R/W	00

13.4.7 PB 口输入模式寄存器 PIMB (0x2C) (修改)

比特位	名称	描述	读 / 写 标志	复 位 值
31:24	PIL57~PIL50	定义端口 P50~P57 输入缓冲器类型： =0: CMOS 缓冲器， $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$ ； =1: TTL 缓冲器， $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$ ；	R/W	00
23:16	PIL47~PIL40	定义端口 P40~P47 输入缓冲器类型： =0: CMOS 缓冲器， $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$ ； =1: TTL 缓冲器， $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$ ；	R/W	00
15:8	PID57~PID50	定义端口 P50~P57 是否是 N-ch 漏极开路输出： =0: 普通模式； =1: N-ch 漏极开路模式；	R/W	00
7:0	PID47~PID40	定义端口 P40~P47 是否是 N-ch 漏极开路输出： =0: 普通模式； =1: N-ch 漏极开路模式；	R/W	00

13.4.8 PB 口输入使能寄存器 PIEB (0x30)

比特位	名称	描述	读 / 写 标志	复 位 值
31:24	PIE77~PIE70	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF
23:16	PIE67~PIE60	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF
15:8	PIE57~PIE50	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF
7:0	PIE47~PIE40	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF

13.5 PC 口

13.5.1 PC 口模式寄存器 PMC (输入或者输出) (0x34)

比特位	名称	描述	读 / 写 标志	复 位 值
31:30	PM117~PM116	=0 输出模式 =1 输入模式	R/W	3
29:28	保留	保留, 请不要配置	R/W	3
27:24	PM113~PM110	=0 输出模式 =1 输入模式	R/W	F
23:20	保留	保留, 请不要配置	R/W	F
19:16	PM103~PM100	=0 输出模式 =1 输入模式	R/W	F
15:8	PM97~PM90	=0 输出模式 =1 输入模式	R/W	FF
7:0	PM87~PM80	=0 输出模式 =1 输入模式	R/W	FF

13.5.2 PC 口数据寄存器 PC (0x38)

比 特 位	名称	描述	读 / 写 标志	复 位 值
31:30	PM117~PM116	定义芯片端口需要输出的数据。 如果在输入模式下读端口, 则读取的是引脚电平。如果在输出模式下读端口, 则读取的是输出锁存器的值	R/W	0
29:28	保留	保留, 请不要配置	R/W	0
27:24	P113~P110	定义芯片端口需要输出的数据。 如果在输入模式下读端口, 则读取的是引脚电平。如果在输出模式下读端口, 则读取的是输出锁存器的值	R/W	00

23:20	保留	保留, 请不要配置	R/W	0
19:16	P103~P100	定义芯片端口需要输出的数据。 如果在输入模式下读端口, 则读取的是引脚电平。如果在输出模式下读端口, 则读取的是输出锁存器的值	R/W	00
15:8	P97~P90	定义芯片端口需要输出的数据。 如果在输入模式下读端口, 则读取的是引脚电平。如果在输出模式下读端口, 则读取的是输出锁存器的值	R/W	00
7:0	P87~P80	定义芯片端口需要输出的数据。 如果在输入模式下读端口, 则读取的是引脚电平。如果在输出模式下读端口, 则读取的是输出锁存器的值	R/W	00

13.5.3 PC 口复用寄存器 PCC (0x3C) (修改)

比特位	名称	描述	读 / 写 标志	复位值
31:29	保留	-----	R	0
28	SPI_MUX	SPI_MUX 定义是 P11 口还是 P4 口作为 SPI0 接口: =0: 选择 P44~P47 作为 SPI0 接口; =1: 选择 P110~P113 作为 SPI0 接口。	R/W	0
27	PC113	{PCC3[11:9],PCC[27]}定义 P113 的配置 =0000: 选择为 IO 口 =0001: 选择为 SPI0_MOSI =0010: 选择为 SPI2_MOSI =0011: 选择为 SPI3_MOSI =其他: 保留	R/W	0
26	PC112	{PCC3[8:6],PCC[26]}定义 P112 的配置 =0000: 选择为 IO 口 =0001: 选择为 SPI0_MISO =0010: 选择为 SPI2_MISO =0011: 选择为 SPI3_MISO =其他: 保留	R/W	0
25	PC111	{PCC3[5:3],PCC[25]}定义 P111 的配置 =0000: 选择为 IO 口 =0001: 选择为 SPI0_SCLK =0010: 选择为 SPI2_SCLK =0011: 选择为 SPI3_SCLK =其他: 保留	R/W	0
24	PC110	由 {PCC3[2:0],PCC[24]}定义 P110 的配置 =0000: 选择为 IO 口 =0001: 选择为 SPI0_SCSN =0010: 选择为 SPI2_SCSN =0011: 选择为 SPI3_SCSN =其他: 保留 注 1: PC110 仅在 SPI 从模式下有效, 在 SPI 主模式下不要配置为 1.	R/W	0

		注 2: P40~P43 口和 P110~P113 口不能同时复用为 SPI2 注 3: P14~P17, P20~P23 口和 P110~P113 口不能同时复用为 SPI3		
23	PC107	由 {PCC2[23:21],PCC[23]} 定义端口 P107 复用配置 =0000: 保留 IO 功能, 用户不要配置 =0001: 保留 =其他: 保留 注: 作为 ADC 模拟输入一直有效, 不需要配置复用	R/W	0
22	PC106	由 {PCC2[20:18],PCC[22]} 定义端口 P106 复用配置 =0000: 保留 IO 功能, 用户不要配置 =0001: IB_IN =其他: 保留 注 1: 作为 ADC 模拟输入一直有效, 不需要配置复用; 作为 IB_IN 口时, 仅支持输入功能, 可外灌 ADC 1bit 信号。	R/W	0
21	PC105	由 {PCC2[17:15],PCC[21]} 定义端口 P105 复用配置 =0000: 保留 IO 功能, 用户不要配置 =0001: 保留 =其他: 保留 注: 作为 ADC 模拟输入一直有效, 不需要配置复用	R/W	0
20	PC104	由 {PCC2[14:12],PCC[20]} 定义端口 P104 复用配置 =0000: 保留 IO 功能, 用户不要配置 =0001: 保留 =其他: 保留 注 1: 作为 ADC 模拟输入一直有效, 不需要配置复用;	R/W	0
19:16	PC103~PC100	PC103~PC100 定义端口复用配置: =0: 选择为 IO 口; =1: 选择为 SEG。	R/W	00
15:8	PC97~PC90	PC97~PC90 定义端口复用配置: =0: 选择为 IO 口; =1: 选择为 SEG。	R/W	00
7:0	PC87~PC85	PC87~PC85 定义端口复用配置: =0: 选择为 IO 口; =1: 选择为 SEG。	R/W	0
4	PC84	由 {PCC4[14:12],PCC[4]} 定义端口 P84 复用配置 =0000: 选择为 IO 口 =0001: 选择为 SEG =0010: 选择为 SPI3_SCLK (开漏引脚) =其他: 保留	R/W	0
3	PC83	由 {PCC4[11:9],PCC[3]} 定义端口 P83 复用配置 =0000: 选择为 IO 口 =0001: 选择为 SEG =0010: 选择为 SPI3_SCSN (开漏引脚) =其他: 保留	R/W	0
2	PC82	PC82 定义端口 P82 复用配置:	R/W	0

		=0: 选择为 IO 口; =1: 选择为 SEG。		
1	PC81	由 {PCC4[5:3],PCC[1]} 定义端口 P81 复用配置 =0000: 选择为 IO 口 =0001: 选择为 SEG =0010: 选择为 SDA =其他: 保留	R/W	0
0	PC80	由 {PCC4[2:0],PCC[0]} 定义端口 P80 复用配置 =0000: 选择为 IO 口 =0001: 选择为 SEG =0010: 选择为 SCL =其他: 保留	R/W	0

13.5.1 PC 口复用寄存器 2 PCC2 (0x98) (新增)

比特位	名称	描述	读 / 写 标志	复位值
31:24	---	预留	R/W	0
23:21	PC107_2	具体定义见 PCC 寄存器的 bit[23]	R/W	0
20:18	PC106_2	具体定义见 PCC 寄存器的 bit[22]	R/W	0
17:15	PC105_2	具体定义见 PCC 寄存器的 bit[21]	R/W	0
14:12	PC104_2	具体定义见 PCC 寄存器的 bit[20]	R/W	0
11:9	PC103_2	预留	R	0
8:6	PC102_2	预留	R	0
5:3	PC101_2	预留	R	0
2:0	PC100_2	预留	R	0

13.5.2 PC 口复用寄存器 3 PCC3 (0x9C) (新增)

比特位	名称	描述	读 / 写 标志	复位值
31:28	---	预留	R	0
27:24	PC117	PC117 定义端口 P117 复用配置 =0000: 选择为 IO 口 =0001: 选择为万年历秒输出 RTC1S =其他: 保留	R/W	0
23:20	PC116	PC116 定义端口 P116 复用配置 =0000: 选择为 IO 口 (开漏引脚, 仅支持开漏应用) =0001: 选择为模拟输入 AIN6 =其他: 保留	R/W	0
19:16	PC115	PC115 定义端口 P115 复用配置 =0000: 保留 IO 功能, 用户不要配置 =0001: 保留 =其他: 保留 注: 作为 ADC 模拟输入一直有效, 不需要配置复用	R/W	0
15:12	PC114	PC114 定义端口 P114 复用配置	R/W	0

		=0000: 保留 IO 功能, 用户不要配置 =0001: IA_IN =其他: 保留 注 1: 作为 ADC 模拟输入一直有效, 不需要配置复用; 作为 IA_IN 时, 仅支持输入功能。		
11:9	PC113_2	具体定义见 PCC 寄存器的 bit[27]	R/W	0
8:6	PC112_2	具体定义见 PCC 寄存器的 bit[26]	R/W	0
5:3	PC111_2	具体定义见 PCC 寄存器的 bit[25]	R/W	0
2:0	PC110_2	具体定义见 PCC 寄存器的 bit[24]	R/W	0

13.5.3 PC 口复用寄存器 4 PCC4 (0xA0) (新增)

比特位	名称	描述	读 / 写标志	复位值
31:24	---	预留	R	0
23:21	PC87_2	预留	R	0
20:18	PC86_2	预留	R	0
17:15	PC85_2	预留	R	0
14:12	PC84_2	具体定义见 PCC 寄存器的 bit[4]	R/W	0
11:9	PC83_2	具体定义见 PCC 寄存器的 bit[3]	R/W	0
8:6	PC82_2	预留	R	0
5:3	PC81_2	具体定义见 PCC 寄存器的 bit[1]	R/W	0
2:0	PC80_2	具体定义见 PCC 寄存器的 bit[0]	R/W	0

13.5.4 PC 口上下拉选择寄存器 PUC (0x40)

比特位	名称	描述	读 / 写标志	复位值
31:30	PU117~PU116	PU117~PU116 定义端口是否内接上拉: =0: 不接上拉; =1: 内接上拉。	R/W	00
29:28	---	保留	R	00
27:24	PU113~PU110	PU113~PU110 定义端口是否内接上拉: =0: 不接上拉; =1: 内接上拉。	R/W	00
23:20	保留	-----	R	0
19:16	PU103~PU100	PU103~PU100 定义端口是否内接下拉: =0: 不接下拉; =1: 内接下拉。	R/W	0
15:8	PU97~PU90	PU97~PU90 定义端口是否内接下拉: =0: 不接下拉; =1: 内接下拉。	R/W	00
7:0	PU87~PU80	PU87~PU80 定义端口是否内接下拉: =0: 不接下拉;	R/W	00

		=1: 内接下拉。		
--	--	-----------	--	--

13.5.5 PC 口输入使能寄存器 PIEC (0x44) (修改)

比特位	名称	描述	读 / 写标志	复位值
31:30	PIE117~PIE116	输入使能: =0: 不使能输入; =1: 使能输入;	R/W	3
29:28	--	保留	R/W	3
27:24	PIE113~PIE110	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	F
23:20	保留	-----	R/W	F
19:16	PIE103~PIE100	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	F
15:8	PIE97~PIE90	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF
7:0	PIE87~PIE80	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF

13.5.6 PC 口输入模式寄存器 PIMC (0x48)

比特位	名称	描述	读 / 写标志	复位值
31:16	---	预留	R	0
15:14	PIL117~PIL116	定义端口 P117~P116 输入缓冲器类型: =0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$; =1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$;	R/W	0
13:12	---	保留	R/W	0
11:10	---	保留	R/W	0
9:8	---	保留	R/W	0
7:4	PIL113~PIL110	定义端口 P113~P110 输入缓冲器类型: =0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$; =1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$;	R/W	00
3:0	PID113~PID110	定义端口 P113~P110 是否是 N-ch 漏极开路输出: =0: 普通模式; =1: N-ch 漏极开路模式;	R/W	00

13.6 PD 口 (新增)

13.6.1 PD 口模式寄存器 PMD (0x50) (输入或者输出) (新增)

比特位	名称	描述	读 / 写标志	复位值
-----	----	----	---------	-----

			志	
31:24	---	预留	R/W	FF
23:16	PM147~PM140	=0 输出模式 =1 输入模式	R/W	FF
15:8	PM137~PM130	=0 输出模式 =1 输入模式	R/W	FF
7:0	PM127~PM120	=0 输出模式 =1 输入模式	R/W	FF

13.6.2 PD 口数据寄存器 PD (0x54) (新增)

比特位	名称	描述	读/写标志	复位值
31:24	---	预留	R/W	00
23:16	P147~P140	定义芯片端口需要输出的数据。 如果在输入模式下读端口,则读取的是引脚电平。如果在输出模式下读端口,则读取的是输出锁存器的值	R/W	00
15:8	P137~P130	定义芯片端口需要输出的数据。 如果在输入模式下读端口,则读取的是引脚电平。如果在输出模式下读端口,则读取的是输出锁存器的值	R/W	00
7:0	P127~P120	定义芯片端口需要输出的数据。 如果在输入模式下读端口,则读取的是引脚电平。如果在输出模式下读端口,则读取的是输出锁存器的值	R/W	00

13.6.3 PD 口复用寄存器 PCD (0x58) (新增)

比特位	名称	描述	读 / 写 标志	复位值
31:14	---	预留	R	0
13:8	PC135~PC130	PC137~PC130 定义端口复用配置: =0: 选择为 IO 口; =1: 选择 LCD 电压引脚 P133 保留	R/W	0x3F
7:0	PC127~PC120	PC127~PC120 定义端口复用配置: =0: 选择为 IO 口; =1: 选择为 SEG/COM P127~P120 对应 COM7~COM0	R/W	0xFF

13.6.4 PD 口复用寄存器 PCD2 (0xA8) (新增)

比特位	名称	描述	读 / 写 标志	复位值
31:28	PC147	PC147 定义 P147 口复用配置: =0000: 选择为 IO 口; =0001: 选择为万年历秒输出 RTC1S =其他: 预留	R/W	00
27:24	PC146	PC146 定义 P146 口复用配置: =0000: 选择为 IO 口;	R/W	00

		=0001: 选择为 ADC_CLKO =0010: 选择为智能微断 TRIG_OUT =0011: 选择为 SPI2_MOSI =其他: 预留		
23:20	PC145	PC145 定义 P145 口复用配置: =0000: 选择为 IO 口; (开漏引脚, 仅支持开漏应用) =0001: 选择为 ADC_CLKO =0010: 选择为智能微断 TRIG_OUT =0011: 选择为 SPI2_MISO =其他: 预留	R/W	00
19:16	PC144	PC144 定义 P144 口复用配置: =0000: 选择为 IO 口; =0001: 选择为 ADC_CLKO =0010: 选择为智能微断 TRIG_OUT =0011: 选择为 SPI2_SCLK =其他: 预留	R/W	00
15:12	PC143	PC143 定义 P143 口复用配置: =0000: 选择为 IO 口; =0001: 选择为 ADC_CLKO =0010: 选择为智能微断 TRIG_OUT =0011: 选择为 SPI2_SCSN =其他: 预留 注 1: PC143 仅在 SPI 从模式下有效, 在 SPI 主模式下不要配置为 1.	R/W	00
11:8	PC142	PC142 定义 P142 口复用配置: =0000: 选择为 IO 口; =0001: 选择为 ADC_CLKO =0010: 选择为智能微断 TRIG_OUT =0011: 选择为万年历秒输出 RTC1S =其他: 预留	R/W	0
7:4	PC141	预留	R/W	00
3:0	PC140	PC140 定义 P140 口复用配置: =0000: 选择为 IO 口; =0001: 选择为 ADC_CLKO =0010: 选择为智能微断 TRIG_OUT =0011: 选择为万年历秒输出 RTC1S =其他: 预留	R/W	00

13.6.5 PD 口上/下拉选择寄存器 PUD (0x5C) (新增)

比特位	名称	描述	读 / 写 标志	复位值
31:24	---	保留	R/W	00
23:17	PU147~PU141	PU147~PU140 定义端口是否内接上拉: =0: 不接上拉;	R/W	00

		=1: 内接上拉。 P141 暂不支持		
16	PU140	PU140 定义端口是否内接上拉: =0: 内接上拉; =1: 不接上拉。		
15:8	PU137~PU130	PU137~PU130 定义端口是否内接下拉: =0: 不接下拉; =1: 内接下拉。 P133/P136/P137 暂不支持	R/W	00
7:0	PU127~PU120	PU127~PU120 定义端口是否内接下拉: =0: 不接下拉; =1: 内接下拉。	R/W	00

13.6.6 PD 口输入使能 PIED (0x84) (新增)

比特位	名称	描述	读 / 写 标志	复位值
31:24	---	保留	R/W	FF
23:16	PIE147~PIE140	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF
15:8	PIE137~PIE130	输入使能: =1: 不使能输入; =0: 使能输入; P133/P136/P137 暂不支持	R/W	FF
7:0	PIE127~PIE120	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF

13.6.7 PD 口输入模式寄存器 PIMD (0x88) (新增)

比特位	名称	描述	读 / 写 标志	复位值
31:16	---	预留	R	0
15:12	PIL147~PIL144	定义端口 P147~P144 输入缓冲器类型: =0: CMOS 缓冲器, Vil=0.3VCC Vih=0.7VCC; =1: TTL 缓冲器, Vil=0.16VCC Vih=0.4VCC;	R/W	00
11:8	PID147~PID144	定义端口 P147~P144 是否是 N-ch 漏极开路输出: =0: 普通模式; =1: N-ch 漏极开路模式; P145 作为输出时, 请配置为开漏模式使用	R/W	00
7:4	PIL143~PIL140	定义端口 P143~P140 输入缓冲器类型: =0: CMOS 缓冲器, Vil=0.3VCC Vih=0.7VCC; =1: TTL 缓冲器, Vil=0.16VCC Vih=0.4VCC;	R/W	00

		P141 暂不支持		
3:0	PID143~PID140	定义端口 P143~P140 是否是 N-ch 漏极开路输出： =0：普通模式； =1：N-ch 漏极开路模式； P141 暂不支持	R/W	00

13.7 COM 端口

13.7.1 SEGCOM 口复用寄存器 PCE (0x60)

比特位	名称	描述	读/写标志	复位值
31:4	---	预留	R	0
3:0	SEG3/COM7~SEG0/COM4	SEG3/COM7~SEG0/COM4 定义端口复用配置： =0：选择为 SEG； =1：选择为 COM。	R/W	00

13.8 置位和清零寄存器

13.8.1 PA 口数据置位寄存器 PASET (0x64)

比特位	名称	描述	读/写标志	复位值
31:30	保留	-----	R	0
29:24	P35~P30	置位芯片端口状态 0：无影响 1：该端口置位，输出高电平	R/W	00
23:16	P27~P20	置位芯片端口状态 0：无影响 1：该端口置位，输出高电平	R/W	00
15:8	P17~P10	置位芯片端口状态 0：无影响 1：该端口置位，输出高电平	R/W	00
7:5	---	预留	R	0
4:0	P04~P00	置位芯片端口状态 0：无影响 1：该端口置位，输出高电平	R/W	00

Note：读出值无意义

13.8.2 PA 口清零置位寄存器 PACLR (0x68)

比特位	名称	描述	读/写标志	复位值
31:30	保留	-----	R	00
29:24	P35~P30	清零芯片端口状态 0：无影响 1：该端口清零，输出低电平	R/W	0
23:16	P27~P20	清零芯片端口状态 0：无影响	R/W	00

		1: 该端口清零, 输出低电平		
15:8	P17~P10	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00
7:5	---	预留	R	0
4:0	P04~P00	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00

Note: 读出值无意义

13.8.3 PB 口数据置位寄存器 PBSET (0x6C)

比特位	名称	描述	读/写标志	复位值
31:24	P77~P70	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
23:16	P67~P60	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
15:8	P57~P50	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
7:0	P47~P00	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00

Note: 读出值无意义

13.8.4 PB 口清零置位寄存器 PBCLR (0x70)

比特位	名称	描述	读/写标志	复位值
31:24	P77~P70	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	0
23:16	P67~P60	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00
15:8	P57~P50	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00
7:0	P47~P40	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00

Note: 读出值无意义

13.8.5 PC 口数据置位寄存器 PCSET (0x74)

比特位	名称	描述	读/写标志	复位值
31:30	P117~P116	清零芯片端口状态 0: 无影响	R/W	0

		1: 该端口置位, 输出高电平		
29: 28	---	保留	R/W	0
27:24	P113~P110	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
23:20	保留	-----	R	0
19:16	P103~P100	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	0
15:8	P97~P90	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
7:0	P87~P80	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00

Note: 读出值无意义

13.8.6 PC 口清零置位寄存器 PCCLR (0x78)

比特位	名称	描述	读/写标志	复位值
31:30	P117~P116	清零芯片端口状态 0: 无影响 1: 该端口置位, 输出低电平	R/W	0
29: 28	---	保留	R/W	0
27:24	P113~P110	清零芯片端口状态 0: 无影响 1: 该端口置位, 输出低电平	R/W	00
23:20	保留	-----	R	0
19:16	P103~P100	清零芯片端口状态 0: 无影响 1: 该端口置位, 输出低电平	R/W	00
15:8	P97~P90	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00
7:0	P87~P80	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00

Note: 读出值无意义

13.8.1 PD 口数据置位寄存器 PDSET (0x7C) (新增)

比特位	名称	描述	读/写标志	复位值
31:24	---	保留	R/W	00
23:16	P147~P140	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平 P141 暂不支持	R/W	0

15:8	P137~P130	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平 P133/P136/P137 暂不支持	R/W	00
7:0	P127~P120	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00

Note: 读出值无意义

13.8.2 PD 口清零置位寄存器 PDCLR (0x80) (新增)

比特位	名称	描述	读/写标志	复位值
31:24	---	保留	R/W	00
23:16	P147~P140	清零芯片端口状态 0: 无影响 1: 该端口置位, 输出低电平 P141 暂不支持	R/W	00
15:8	P137~P130	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平 P133/P136/P137 暂不支持	R/W	00
7:0	P127~P120	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00

Note: 读出值无意义

13.9 IO 配置寄存器

14 外部中断控制器 INTC (修改)

SoC 内置外部中断控制器 (INTC), 用于处理从芯片管脚输入的中断请求。

14.1 概述

外部中断控制器具备如下特性:

- ◎ 支持 8 个外部中断的模式设置: 上下边沿及双边沿可设;
- ◎ 支持外部中断状态指示;
- ◎ 支持外部中断软件触发;
- ◎ 支持外部中断状态;
- ◎ 支持外部中断屏蔽;
- ◎ 支持外部中断滤波, 滤波时间约为 10 个系统时钟周期;

14.2 寄存器描述

模块寄存器基址

模块名	物理地址	映射地址
INTC	0x40044000	0x40044000

INTC 模块的寄存器偏移地址

寄存器名	地址偏移量	描述
INTC_CTL	0x0	INTC 控制寄存器
INTC_MODE	0x4	INTC 模式寄存器
INTC_MASK	0x8	INTC Mask 寄存器
INTC_STA	0xc	INTC 状态寄存器

14.2.1 INTC_CTL (0x0) (修改)

INTC 控制寄存器

地址 0x40044000+0x0

比特位	名称	描述	读/写标志	复位值
31:08	---	预留	R	0
8	IRQ_CTL	外部中断合并功能配置： 0：8 路外部中断 INT0~INT7 独立，向下兼容 1：8 路外部中断 INT0~INT7 共用一个中断号 EXT0，通过查询 STA 寄存器中断标志来确定哪个外部中断产生。	R/W	0
7:0	Enable	使能信号，Enable[7:0]分别对应于外部中断请求 7~0，对应的外部管脚为：P37~P30。 0：关闭对应的外部中断 1：使能对应的外部中断	R/W	0

14.2.2 INTC_MODE (0x4)

INTC 模式寄存器

地址 0x40044000+0x4

比特位	名称	描述	读/写标志	复位值
31:16	---	预留	R	0
15:14	MODE7	外部中断请求 7 (P37/INT7) 模式选择 00：上升沿 01：下降沿 10：双边沿 11：保留	R/W	0
13:12	MODE6	外部中断请求 6 (P36/INT6) 模式选择 00：上升沿 01：下降沿 10：双边沿 11：保留	R/W	0
11:10	MODE5	外部中断请求 5 (P35/INT5) 模式选择 00：上升沿 01：下降沿 10：双边沿 11：保留	R/W	0
9:8	MODE4	外部中断请求 4 (P34/INT4) 模式选择 00：上升沿 01：下降沿 10：双边沿 11：保留	R/W	0
7:6	MODE3	外部中断请求 3 (P33/INT3) 模式选择	R/W	0

		00: 上升沿 01: 下降沿 10: 双边沿 11: 保留		
5:4	MODE2	外部中断请求 2 (P32/INT2) 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留	R/W	0
3:2	MODE1	外部中断请求 1 (P31/INT1) 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留	R/W	0
1:0	MODE0	外部中断请求 0 (P30/INT0) 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留	R/W	0

14.2.3 INTC_MASK (0x8)

INTC Mask 寄存器

地址 0x40044000+0x8

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:0	MASK	MASK[7:0] 分别对应于外部中断请求 7~0 0: 中断禁止 1: 中断使能	R/W	0

14.2.4 INTC_STA (0xC)

INTC 状态寄存器

地址 0x40044000+0xc

比特位	名称	描述	读/写标志	复位值
31:08	---	预留	R	0
7:0	STA	STA[7:0]对应于分别对应于外部中断请求 7~0 0: 中断事件未发生 1: 中断事件发生 注: 写 1 清零	R/W	0

14.2.5 LPUART 复用配置寄存器 LURT_CFG (0x100) (新增)

LPUART 与 UARTx 复用 IO 配置寄存器

比特位	名称	描述	读/写标志	复位值
31:3	---	预留	R	0
2:0	LURT_CFG	000: LPUART 无效 001: LPUART 复用 UART0 的 IO 010: LPUART 复用 UART1 的 IO 011: LPUART 复用 UART2 的 IO 100: LPUART 复用 UART3 的 IO 101: LPUART 复用 UART4 的 IO 110: LPUART 复用 UART5 的 IO 111: 保留 LPUART 有效后, 原 UARTx 的复用失效, 具体的复用关系根据 GPIO 章节进行配置。	R/W	0

14.2.6 IO 驱动配置寄存器 IOCFG (0x104) (新增)

比特位	名称	描述	读/写标志	复位值
31:28	---	预留	R	0
27:24	IOCFG_6	P143~P146 IO 驱动配置: 1: 驱动能力为 3mA, slew rate 为 fast 模式 0: 驱动能力为 1.5mA, slew rate 为 slow 模式	R/W	1
23:20	IOCFG_5	P110~P113 IO 驱动配置: 1: 驱动能力为 3mA, slew rate 为 fast 模式 0: 驱动能力为 1.5mA, slew rate 为 slow 模式	R/W	1
19:16	IOCFG_4	P52~P55 IO 驱动配置: 1: 驱动能力为 3mA, slew rate 为 fast 模式 0: 驱动能力为 1.5mA, slew rate 为 slow 模式	R/W	1
15:12	IOCFG_3	P44~P47 IO 驱动配置: 1: 驱动能力为 3mA, slew rate 为 fast 模式 0: 驱动能力为 1.5mA, slew rate 为 slow 模式	R/W	1
11:8	IOCFG_2	P40~P43 IO 驱动配置: 1: 驱动能力为 3mA, slew rate 为 fast 模式 0: 驱动能力为 1.5mA, slew rate 为 slow 模式	R/W	1
7:4	IOCFG_1	P20~P23 IO 驱动配置: 1: 驱动能力为 3mA, slew rate 为 fast 模式 0: 驱动能力为 1.5mA, slew rate 为 slow 模式	R/W	1
3:0	IOCFG_0	P14~P17 IO 驱动配置: 1: 驱动能力为 3mA, slew rate 为 fast 模式 0: 驱动能力为 1.5mA, slew rate 为 slow 模式	R/W	1

14.3 GPIO 操作步骤

- 1、配置系统控制章节模块使能 1 寄存器 MOD1_EN 第 5 位为 1 开启 GPIO 模块时钟。
- 2、配置 GPIO 输入输出模式。
- 3、配置 GPIO 口数据寄存器。
- 4、配置 GPIO 口复用功能，选择 GPIO 复用功能后，GPIO 的输入输出功能将跟随 GPIO 的复用配置。
- 5、当 RN821x 使用 5v 供电，而外围 I2C,SPI 或其他器件工作电压为 3.3V 时，可选择能配置为 N-ch 漏极开路输出、输入缓冲器类型可选择 TTL 模式的 GPIO。
- 6、当作为输入 IO 口时，需将输入使能寄存器对应位配置为 0，打开输入使能。在低功耗模式，可将 IO 口配置为输入模式，并关闭输入使能。

15 KBI

SoC 内置按键接口控制器，用于处理从芯片管脚输入的中断请求，可以在 **cpu** 休眠时通过中断自动唤醒 **cpu**。

15.1 特性

按键接口控制器具有如下特性：

- ◎ 支持 8 个按键，对应管脚为 P10/KEY0~P17/KEY7；
- ◎ 支持每个按键状态查询；
- ◎ 支持每个按键输入滤波，滤波时间为 24ms；
- ◎ 支持每个按键可单独屏蔽中断

15.2 寄存器描述

表 13-1 KBI 寄存器基址

模块名	物理地址	映射地址
KBI	0x40028000	0x40028000

表 13-2 KBI 寄存器偏移地址

寄存器名	地址偏移量	描述
KBI_CTL	0x0	控制寄存器
KBI_SEL	0x4	选择寄存器
KBI_DATA	0x8	数据寄存器
KBI_MASK	0xc	屏蔽寄存器

15.2.1 控制寄存器 KBI_CTL (0x0)

表 13-3 KBI 控制寄存器 KBI_CTL

比特位	名称	描述	读 / 写 标志	复位值
31:8	---	预留	R	0
7:0	EN	使能信号，EN[7:0]对应于 KEY[7:0]， 对应的外部管脚为：P17/KEY7~P10/KEY0。 0：关闭对应的 KEY 1：使能对应的 KEY	R/W	0

15.2.2 选择寄存器 KBI_SEL (0x4)

表 13-4 KBI 选择寄存器 KBI_SEL

比特位	名称	描述	读 / 写 标志	复位值
31:8	---	预留	R	0
7:0	SEL	SEL[7:0]对应于 KEY[7:0] 0：上升沿有效 1：下降沿有效	R/W	0

15.2.3 数据寄存器 KBI_DATA (0x8)

表 13-5 KBI 数据寄存器 KBI_DATA

比特位	名称	描述	读 / 写 标志	复位值
31:8	---	预留	R	0
7:0	DAT	DAT[7:0]对应于 KEY[7:0]。写 1 清零 0: 按键没有按下 1: 按键被按下	R/W	0

15.2.4 屏蔽寄存器 KBI_MASK (0xC)

表 13-6 KBI 屏蔽寄存器 KBI_MASK

比特位	名称	描述	读 / 写 标志	复位值
31:8	---	预留	R	0
7:0	MASK	MASK[7:0]对应于 KEY[7:0] 0: 中断禁止 1: 中断使能	R/W	0

15.3 KBI 操作步骤

- 1、配置系统控制章节 KBI 使能寄存器 KBI_EN，将第 8 位及对应的 KBI 时钟位配置为 1，开启时钟。
- 2、设置 KBI 控制寄存器 KBI_CTL，使能对应的 KBI。
- 3、设置 KBI 选择寄存器 KBI_SEL，配置对应的 KBI 为上升沿或下降沿。
- 4、配置 KBI_MASK 开启对应中断使能，并打开 KBI 中断 NVIC_EnableIRQ(KBI_IRQn);
- 5、编写 KBI 中断服务程序：

```
void KBI_HANDLER(void)
{
    if(KBI->DATA&0x01)
    {

    }
    KBI->DATA = 0xff;
}
```

所有 KBI 中断均为 1 个入口，需根据 KBI_DATA 判断何 KBI 产生的中断。

- 6、完成

16 UART（修改）

SoC 内置 6 个 UART 接口，用于与外部进行异步串行通信。

16.1 概述

UART 接口控制器具备如下特性：

- ◎ 六个全双工 UART 接口；
- ◎ 内置波特率发生器，支持不同的波特率配置；
- ◎ 数据位宽支持 5/6/7/8bit；
- ◎ 停止位可配置成 1 或 2bit；
- ◎ 可选 38kHz 红外调制；
- ◎ 支持自动波特率检测；
- ◎ 支持红外唤醒；
- ◎ 支持通用 DMA 功能
- ◎ 支持 300 波特率@29M
- ◎ 6 路 UART 均支持 DMA

16.2 寄存器描述

表 14-1 UART 寄存器基址

模块名	物理地址	映射地址
UART0	0x40000000	0x40000000
UART1	0x40004000	0x40004000
UART2	0x40008000	0x40008000
UART3	0x4000C000	0x4000C000
UART4	0x40018000	0x40018000
UART5	0x4001C000	0x4001C000

表 14-2 UART 寄存器偏移地址

寄存器名 x 指 0~5	地址偏移量	描述
UARTx_CTL	0x0	UART 控制寄存器
UARTx_BAUD	0x4	UART 波特率配置寄存器
UARTx_STAT	0x8	UART 状态指示寄存器
UARTx_TXD	0xC	UART 发送数据寄存器
UARTx_RXD	0x10	UART 接收数据寄存器
UARTx_DMA_CTL（新增）	0x18	UARTx DMA 控制寄存器
UARTx_DMA_TBADR（新增）	0x1C	UARTx DMA 发送起始地址寄存器
UARTx_DMA_RBAD（新增）	0x20	UARTx DMA 接收起始地址寄存器
UARTx_DMA_TLEN（新增）	0x24	UARTx DMA 发送长度寄存器
UARTx_DMA_RLEN（新增）	0x28	UARTx DMA 接收长度寄存器
UARTx_DMA_TADR（新增）	0x2C	UARTx 当前发送 DMA 地址寄存器
UARTx_DMA_RADR（新增）	0x30	UARTx 当前接收 DMA 地址寄存器
UARTx_DMA_IE（新增）	0x34	UARTx DMA 中断使能寄存器
UARTx_DMA_FLG（新增）	0x38	UARTx DMA 中断标志寄存器

UARTx_DMA_TO (新增)	0x3C	UARTx DMA 接收超时配置寄存器
-------------------	------	---------------------

注：在运行过程中，不能修改控制寄存器与波特率寄存器。

16.2.1 控制寄存器 UARTx_CTL (0x0)

比特位	名称	描述	读 / 写 标志	复位值
31:15	---	预留	R	0
14	NEG	UART 极性选择位： 0：正极性，即默认驱动电平为高，发送/接收的数据极性不变 1：负极性，即默认驱动电平为低，发送/接收的数据极性取反	R/W	0
13	LMSB	LSB/MSB 选择方式 0：LSB 先传输 1：MSB 先传输 注：当 PARS 选择为用户自定义校验时，校验位当作数据扩展最高位，此时选择 MSB，最先传输的将是校验位	R/W	0
12	IRSEL	红外调制极性选择： 0：正极性，即低电平调制输出，高电平（默认状态）保持 1：负极性，即数据取反，高电平调制输出，低电平保持 注意：IRSEL 只决定空闲输出（无效电平）时的电平高低，不影响有效数据期间的电平	R/W	0
11	ILBE	内部环回使能 0：内部环回禁止 1：内部环回使能，TXD 与 RXD 信号在模块内部短接	R/W	0
10	IRE	红外调制使能位 0：关闭红外调制输出 1：打开红外调制输出，用 38k 载波调制输出数据的低电平	R/W	0
9:7	PARS	校验位选择 000：无校验 001：奇校验 010：偶校验 011：固定为零校验 100：固定为 1 校验 其它：用户自定义校验	R/W	0
6:5	DATLEN	传输数据宽度位 00：5-bit 01：6-bit 10：7-bit 11：8-bit	R/W	0
4	STOPS	停止位位宽选择 0：1-bit 停止位 1：2-bit 停止位	R/W	0
3	ERRIE	错误中断使能位，对应的标志位是状态指示寄存器中 bit5~bit2。 0：关闭中断	R/W	0

		1: 打开中断		
2	RXIE	接收数据中断使能位, 对应的标志位是状态指示寄存器中 bit1. 0: 关闭中断 1: 打开中断	R/W	0
1	TXIE	发送数据中断使能位, 对应的标志位是状态指示寄存器中 bit0. 0: 关闭中断 1: 打开中断	R/W	0
0	EN	模块使能 0: 关闭 1: 打开	R/W	0

16.2.2 波特率整数分频寄存器 UARTx_BAUD (0x4)

比特位	名称	描述	读 / 写 标志	复 位 值
31:13	---	预留	R	0
12:0	CLKDIV	UARTx clock 分频数 波特率的计算公式是: 系统时钟/[16*(CLKDIV+1)]	R/W	0

仅使用 UARTx_BAUD 寄存器, 波特率会存在偏差。为解决此问题, 该芯片还提供了波特率小数分频配置寄存器 UARTx_FDIV (见后文 15.2.6), 这两个寄存器配合使用, 可得到满足要求的波特率。

当系统时钟为 14.7456MHz 时, 典型波特率配置如下表。

第 1 列				第 2 列			
目标波特率 (bps)	波特率配置(hex)		实测误差	目标波特率 (bps)	波特率配置(hex)		实测误差
	UARTx_ BAUD	UARTx_ FDIV			UARTx_ BAUD	UARTx_ FDIV	
120	1DFF	0	0.40%	9837	5C	2BF9	0.10%
300	BFF	0	-0.20%	14400	3F	0	0.06%
1200	2FF	0	0.16%	19200	2F	0	-0.22%
2400	17F	0	0.16%	38400	17	0	0.16%
4800	BF	0	0.16%	115200	7	0	0.01%
9600	5F	0	0.16%	921600	0	0	0.10%

举例: 芯片运行在 14.7456MHZ 频率下, 要配置 uart0 的波特率为 1200bps, 只需将 UART0_Baud 寄存器配置为 0x2FF; 要配置 uart0 的波特率为 750000bps, 需将 UART0_Baud 寄存器配置为 0x0, UART0_FDIV 寄存器配置为 0xEA5, 其它对应波特率见表格。

如果芯片运行在其他其它频率下, 设置值可参考“14.2.2 波特率配置寄存器 UARTx_BAUD”和“14.2.6 波特率小数分频配置寄存器 UARTx_FDIV”公式。

注: UARTx_BAUD 寄存器有效位数是 13 位, 当其值取最大 0x1FFF 时, 对应 UART 支持的最小波特率; 相应的, 其值取最小 0 时, 对应 UART 支持的最高波特率。如芯片运行在 14.7456MHZ 频率下, 最小波特率为 $14745600/[16*(8191+1)]=112\text{bps}$, 最高波特率为 $14745600/16=921600\text{bps}$ 。

16.2.3 状态指示寄存器 UARTx_STA (0x8)

比特位	名称	描述	读 / 写 标志	复 位 值
31:8	---	预留	R	0
9	tx_fifo_full	发送 FIFO 满:	R	0

		0: 不满 1: 满		
8	tx_fifo_empty	发送 FIFO 空: 0: 非空 1: 空	R	1
7	TB	发送状态标志位 0: 没有发送 1: 正在发送数据	R	0
6	RB	接收状态标志位 0: 没有接收 1: 正在接收数据	R	0
5	DE	数据错误, 写 1 清零 UART 发送 FIFO 已满后继续写 UART 发送寄存器或在发送过程中写入了新的发送数据, 该位会置起 0: 无错误 1: 有错误	R/W	0
4	FE	帧错误, 写 1 清零 UART 接收到的字节数据不符合帧格式标志位 (如奇偶校验位错), 如接收到的停止位是 0 而不是 1, 该位会置起 0: 无错误 1: 有错误	R/W	0
3	OE	溢出错误, 写 1 清零 UART 接收数据寄存器没有及时读取导致接收溢出, 该位会置起 0: 无错误 1: 有错误	R/W	0
2	PE	校验错误, 写 1 清零 UART 接收到的数据校验和错误, 该位会置起 0: 无错误 1: 有错误	R/W	0
1	TX	发送标识, 写 1 清零 0: 数据尚未发送或无待发送数据 1: 数据已发送	R/W	0
0	RX	接收标识, 写 1 清零/读后清零 0: 无接收数据 1: 数据已接收 注: 该位在上述两种情况下都会被清零	R/W	0

16.2.4 发送数据寄存器 UARTx_TXD (0xC)

比特位	名称	描述	读 / 写 标志	复位值
31:9	---	预留	R	0
8	UP	用户自定义校验位	R/W	0
7:0	TXDATA	发送数据寄存器	R/W	0

16.2.5 接收数据寄存器 UARTx_RXD (0x10)

比特位	名称	描述	读 / 写 标志	复位值
31:9	---	预留	R	0
8	UP	校验位	R	0
7:0	RXDATA	接收数据寄存器	R	0

16.2.6 波特率小数分频配置寄存器 UARTx_FDIV (0x14)

比特位	名称	描述	读/写标志	复位值
31:14	---	预留	R	0
13:0	FDIV	<p>小数分频系数。计算公式为：</p> $FDIV = \left[\left(\frac{f_i}{16 \times f_o} - \left\lfloor \frac{f_i}{16 \times f_o} \right\rfloor \right) \times 2^{14} + 0.5 \right]$ <p>其中 fi 为输入时钟(cpu 当前运行时钟), fo 为输出时钟, “$\left\lfloor \right\rfloor$” 为向下求整算符。例如输入时钟为 1.8432MHz, 输出时钟为 9837Hz, 那么：</p> $FDIV = \left[\left(\frac{1843200}{16 \times 9837} - \left\lfloor \frac{1843200}{16 \times 9837} \right\rfloor \right) \times 2^{14} + 0.5 \right]$ <p>求得 F=11647。</p>	R/W	0

16.2.7 DMA 控制寄存器 UARTx_DMA_CTL (0x18)

位	位名称	功能描述	读写标志	复位值
31:5	Reserved	保留。	R	0
4	DMA_PARS	<p>DMA 校验位</p> <p>0: DMA 发送固定 0 校验</p> <p>1: DMA 发送固定 1 校验</p> <p>当 UART_CTRL 寄存器位 PARS<2:0>配置为用户自定义校验时该配置有效, 当 TX_DMA_EN=1 发送 DMA 使能时, 校验位跟随数据一起发送。</p>	R/W	0
3	RX_CYC_MODE	<p>接收循环模式使能</p> <p>0: 不使能</p> <p>1: 使能</p>	R/W	0
2	TX_CYC_MODE	<p>发送循环模式使能</p> <p>0: 不使能</p> <p>1: 使能</p>	R/W	0
1	RX_DMA_EN	<p>接收 DMA 使能</p> <p>0: 不使能</p> <p>1: 使能</p>	R/W	0
0	TX_DMA_EN	<p>发送 DMA 使能</p> <p>0: 不使能</p>	R/W	0

		1: 使能		
--	--	-------	--	--

16.2.8 DMA 发送起始地址寄存器 UARTx_DMA_TBADR (0x1C)

位	位名称	功能描述	读写标志	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_TBADR	DMA 发送起始地址 (Byte 地址)	R/W	0

16.2.9 DMA 接收起始地址寄存器 UARTx_DMA_RBADR (0x20)

位	位名称	功能描述	读写标志	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_RBADR	DMA 接收起始地址 (Byte 地址)	R/W	0

16.2.10 DMA 发送长度寄存器 UARTx_DMA_TLEN (0x24)

位	位名称	功能描述	读写标志	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_TLEN	DMA 发送长度 (Byte 地址) = (n) Byte	R/W	0

16.2.11 DMA 接收长度寄存器 UARTx_DMA_RLEN (0x28)

位	位名称	功能描述	读写标志	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_RLEN	DMA 接收长度 (Byte 地址) = (n) Byte	R/W	0

16.2.12 DMA 当前发送地址寄存器 UARTx_DMA_TADR (0x2C)

位	位名称	功能描述	读写标志	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_TADR	当前发送 DMA 地址 (Byte 地址)	RO	0

16.2.13 DMA 当前接收地址寄存器 UARTx_DMA_RADR (0x30)

位	位名称	功能描述	读写标志	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_RADR	当前接收 DMA 地址 (Byte 地址)	RO	0

16.2.14 DMA 中断使能寄存器 UARTx_DMA_IE (0x34)

位	位名称	功能描述	读写标志	复位值
31:6	Reserved	保留。	R	0
5	RX_ERR_IE	接收数据覆盖中断使能 =0, 不使能 =1, 使能	R/W	0
4	TX_ERR_IE	发送数据错误中断使能 =0, 不使能 =1, 使能	R/W	0
3	RX_FIE	DMA 接收全满中断使能 =0, 不使能	R/W	0

		=1, 使能		
2	RX_HIE	DMA 接收半满中断使能 =0, 不使能 =1, 使能	R/W	0
1	TX_FIE	DMA 发送全空中断使能 =0, 不使能 =1, 使能	R/W	0
0	TX_HIE	DMA 发送半空中断使能 =0, 不使能 =1, 使能	R/W	0

16.2.15 DMA 中断标志寄存器 UARTx_DMA_IF (0x38)

位	位名称	功能描述	读写标志	复位值
31:7	Reserved	保留。	R	0
6	RX_DONE	UART 模块专有, 如果发生接收超时, 则该位置起。写“1”清该位	R/WC	0
5	RX_ERR	接收数据覆盖标志 (该通道 DMA 被其他高优先级 DMA 阻塞未能及时获取 UARTx_RXD 数据, 导致数据被覆盖), 写“1”清该位	R/WC	0
4	--	-保留	R	0
3	RX_FDONE	DMA 接收全满中断标志, 写“1”清该位	R/WC	0
2	RX_HDONE	DMA 接收半满中断标志, 写“1”清该位	R/WC	0
1	TX_FDONE	DMA 发送全空中断标志, 写“1”清该位	R/WC	0
0	TX_HDONE	DMA 发送半空中断标志, 写“1”清该位	R/WC	0

16.2.16 DMA 接收超时配置寄存器 UARTx_DMA_TO (0x3C)

位	位名称	功能描述	读写标志	复位值
31:4	Reserved	保留。	R	0
3:0	TIMEOUT_CNT	UART 接收超时配置, 当 UART 超过一定时间没有接收到起始位, DMA 自动结束。n=n 个 UART 数据位宽时间。配置为 0 无效	RW	0

16.3 UART 数据接收及发送操作步骤

- 1、将系统控制章节模块使能 0 寄存器 MOD0_EN 中对应的 UART 时钟启动, 模块使能。
- 2、配置波特率配置寄存器 UARTx_BAUD, 例如当系统时钟为 3.6864MHz 时, 通讯波特率为 9600, 则波特率配置寄存器可设置为: $UART0 \rightarrow BAUD = 3686400 / (9600 * 16) - 1$;
- 3、配置通讯控制寄存器 UARTx_CTL, 选择数据位、停止位、校验方式及中断使能。
- 4、写 0x3f 清除 UART 状态指示寄存器 (UARTx_STA) 状态。
- 5、配置 UART 中断使能, 打开 UART 中断 NVIC_EnableIRQ(UARTx_IRQn);
- 6、编写中断服务程序, 如 UART0 中断服务程序为:

```
void UART0_HANDLER(void)
```

```
{  
    u32  status;  
    u8   temp;  
    status = UART0->STA;  
  
    /* UART error irq */  
    if((UART0->CTRL & 0x8) && (status & 0x3c))  
    {  
        /* Start adding user code. Do not edit comment generated here */  
    }  
    /* receive data complete irq */  
    if((UART0->CTRL & 0x4) && (status & 0x1))  
    {  
        /* Start adding user code. Do not edit comment generated here */  
    }  
    /* transmit data complete irq */  
    if((UART0->CTRL & 0x2) && (status & 0x2))  
    {  
        /* Start adding user code. Do not edit comment generated here */  
    }  
}
```

UART 接收、发送、错误中断为同一中断入口，需根据控制寄存器打开的中断使能位及状态标志判断此时为何种中断。

7、处理接收或发送的数据，完成。

8、注意：RN821x 的 UART 口为全双工方式，可同时进行发送及接收，当使用 RS485 半双工通讯方式时，RS485 芯片在发送时，接收端会有干扰信号，此时建议在发送时关闭 RN821x 的接收中断，接收时关闭发送中断，消除干扰。

17 ISO7816

SoC 内置两个 ISO7816 通道，支持外接 2 个 7816 协议接口设备。

17.1 概述

ISO7816 接口控制器具备如下特性：

- ◎ 支持标准的 ISO7816 协议，工作在主模式；
- ◎ 支持卡时钟输出，频率可在 1~5MHz 之间设置；
- ◎ 支持 7816 多种分频比设置；
- ◎ 支持 MSB 先输出的低逻辑和 LSB 先输出的高逻辑数据编码方式；
- ◎ 支持 1, 2ETU 宽度的错误信号宽度设置；
- ◎ 支持 0~254ETU 宽度的 EGT 设置；
- ◎ 支持发送数据传输错误重发机制，重发次数可在 0~7 之间设置；
- ◎ 7816 卡协议栈支持接口两个（Esam 和卡）： esam 模块接收和发送用一个管脚口；
- ◎ 支持卡接口接收和发送分离；

17.2 寄存器描述

表 15-1 ISO7816 寄存器基址

模块名	物理地址	映射地址
ISO7816	0x40038000	0x40038000

表 15-2 ISO7816 寄存器偏移地址

寄存器名	地址偏移量	描述
ISO7816_CTRL0	0x0	控制寄存器 0
ISO7816_CTRL1	0x4	控制寄存器 1
ISO7816_CLK	0x8	时钟配置寄存器
ISO7816_BDDIV0	0xc	波特率配置寄存器 0
ISO7816_BDDIV1	0x10	波特率配置寄存器 1
ISO7816_STA0	0x14	状态指示寄存器 0
ISO7816_STA1	0x18	状态指示寄存器 1
ISO7816_DATA0	0x1c	数据发送寄存器 0
ISO7816_DATA1	0x20	数据发送寄存器 1

● 控制寄存器（0x0）

表 15-3 ISO7816 控制寄存器 0 ISO7816_CTRL0

比特位	名称	描述	读/写标志	复位值
31:28	---	只读，不可写	R	0
27	RX_GT0	接收数据 GT 选择位，发送时固定为 2etu 1：接收数据 GT 为 1etu 0：接收数据 GT 为 2etu	R/W	0
26	Reserved	保留，请勿操作	R/W	0
25	Reserved	保留，请勿操作	R/W	0
24:17	EGT0	EGT 宽度选择值(0~255)，即额外保护时间 N，默认值 N=0。 在 0 到 254 范围内，N 用于计算两个相邻数据起始沿之间的延迟： $12 \text{ etu} + (Q \times (N/f))$ 。	R/W	0

		<p>公式中，Q应取下面两个值中的一个：</p> <p>——当复位应答中不存在T=15时，取F/D；</p> <p>——当复位应答中存在T=15时，取 Fi/Di；</p> <p>N=255表示在传输协议期间，两个连续字符的起始沿之间的最小延迟在传输的两个方向上是相同的。这个最小延迟值是：</p> <p>——T=0时，12etu</p> <p>——T=1时，11etu</p>		
16:14	REP_CNT0	<p>数据奇偶校验出错时自动重发次数控制</p> <p>000: 0 次 001: 1 次</p> <p>010: 2 次 011: 3 次</p> <p>100: 4 次 101: 5 次</p> <p>110: 6 次 111: 7 次</p>	R/W	011
13	RXPAR_ESEL0	<p>接收数据奇偶校验错误处理方式选择</p> <p>1: 奇偶校验错，根据 T=0 协议回发 error signal。置 RX_PAR_ERR 标志，进行中断。</p> <p>0: 奇偶校验错，不发送 error signal，置 RX_PAR_ERR 标志，直接中断。</p>	R/W	1
12:11	ERRWTH0	<p>错误信号宽度选择位，只适用于接收，且 RXPAR_ESEL0=1</p> <p>00: 2 etu</p> <p>01: 1 etu</p> <p>10: 1.5 etu</p> <p>11: 2 etu</p>	R/W	01
10:8	PARSEL0	<p>校验位选择位</p> <p>000: 无校验</p> <p>001: 奇校验</p> <p>010: 偶校验</p> <p>011: 固定为零校验</p> <p>100: 固定为 1 校验</p> <p>其它: 保留</p>	R/W	010
7	BGT_EN0	<p>数据接收到发送的 BGT 控制位</p> <p>0: 关闭 BGT 功能，数据接收发送之间不插入 BGT</p> <p>1: 打开 BGT 功能，数据接收发送之间插入 BGT (22etu)</p>	R/W	0
6	ERR_IRQ_EN0	<p>传输出错中断使能位，发送数据时数据冲突、接收数据时数据溢出以及接收数据帧格式错误</p> <p>0: 禁止传输出错产生中断</p> <p>1: 使能传输出错产生中断</p>	R/W	0
5	RX_IRQ_EN0	<p>数据接收中断使能位，使能数据从移位寄存器移入到接收缓存寄存器产生中断</p> <p>0: 禁止数据接收产生中断</p> <p>1: 使能数据接收产生中断</p>	R/W	0
4	TX_IRQ_EN0	<p>数据发送中断使能位，使能数据从移位寄存器发送完成产生中断</p> <p>0: 禁止数据发送产生中断</p>	R/W	0

		1: 使能数据发送产生中断		
3	RX_EN0	接收数据使能 0: 禁止数据接收 1: 使能数据接收	R/W	0
2	TX_EN0	发送数据使能 0: 禁止数据发送 1: 使能数据发送	R/W	0
1	DIRSEL0	数据编码方式选择位 0: LSB 先传的正逻辑数据编码方式 1: MSB 先传的负逻辑数据编码方式（数据取反）	R/W	0
0	EN0	ISO7816 控制器使能位 0: 控制器关闭 1: 控制器打开	R/W	0

● ISO7816 控制寄存器 1 (0x04)

表 15-4 ISO7816 控制寄存器 1 ISO7816_CTRL1

比 特 位	名称	描述	读/写标志	复位值
31	CARD1_CHECK_EN	卡拔出检测使能位, 只在 OLD 检测功能使能后该位有效 1: 使能卡拔出检测中断功能 0: 不使能卡拔出检测中断功能	R/W	0
30	OLD1_IRQ_EN	OLD 检测中断功能使能, 只在 OLD 检测功能使能后该位有效 1: 使能 OLD 检测中断功能 0: 不使能 OLD 检测中断功能	R/W	0
29	OLD1_EN	OLD 检测功能使能 1: 使能 OLD 检测功能 0: 不使能 OLD 检测功能	R/W	0
28	RX1_GT0	接收数据 GT 选择位, 发送时固定为 2etu 1: 接收数据 GT 为 1etu 0: 接收数据 GT 为 2etu	R/W	0
27	Reserved	保留, 请勿操作	R/W	0
26	Reserved	保留, 请勿操作	R/W	0
25	IO1_EN	数据为双向使能信号 1: 78161_IO 端口为双向信号 0: 78161_IO 端口为单向信号, 只输出, 数据输入从 78161_I 端口输入。	R/W	1
24:17	EGT1	EGT 宽度选择值(0~255), 额外保护时间 N 默认值N=0。 在0到254范围内, N表示在准备好接收下一字符之前, 卡要求从前一个字符(也是由卡或接口设备发送的)的起始沿开始的下列延迟: $12 \text{ etu} + (Q \times (N/f))$ 公式中, Q应取下面两个值中的一个: F/D, 即用于计算etu的值, 当T=15不存在于复位应	R/W	0

		<p>答中时， Fi/Di，当T=15存在于复位应答中时。</p> <p>N=255表示在传输协议期间，两个连续字符的起始沿之间的最小延迟在传输的两个方向上是相同的。这个最小延迟值是</p> <p>T=0时，12etu T=1 时，11etu</p>		
16:14	REP_CNT1	<p>数据奇偶校验出错时自动重发次数控制</p> <p>000: 0 次 001: 1 次 010: 2 次 011: 3 次 100: 4 次 101: 5 次 110: 6 次 111: 7 次</p>	R/W	011
13	RXPAR_ESEL1	<p>接收数据奇偶校验错误处理方式选择</p> <p>1: 奇偶校验错，根据 T=0 协议回发 error signal。置 RX_PAR_ERR 标志，进行中断。</p> <p>0: 奇偶校验错，不发送 error signal，置 RX_PAR_ERR 标志，直接中断。</p>	R/W	1
12:11	ERRWITH1	<p>错误信号宽度选择位</p> <p>00: 2 etu 01: 1 etu 10: 1.5 etu 11: 2 etu</p>	R/W	01
10:8	PARSEL1	<p>校验位选择位</p> <p>000: 无校验 001: 奇校验 010: 偶校验 011: 固定为零校验 100: 固定为 1 校验 其它: 保留</p>	R/W	010
7	BGT_EN1	<p>数据接收到发送的 BGT 控制位</p> <p>0: 关闭 BGT 功能，数据接收发送之间不插入 BGT</p> <p>1: 打开 BGT 功能，数据接收发送之间插入 BGT</p>	R/W	0
6	ERR_IRQ_EN1	<p>传输出错中断使能位，发送数据时数据冲突、接收数据时数据溢出以及接收数据帧格式错误</p> <p>0: 禁止传输出错产生中断</p> <p>1: 使能传输出错产生中断</p>	R/W	0
5	RX_IRQ_EN1	<p>数据接收中断使能位，使能数据从移位寄存器移入到接收缓存寄存器产生中断</p> <p>0: 禁止数据接收产生中断</p> <p>1: 使能数据接收产生中断</p>	R/W	0
4	TX_IRQ_EN1	<p>数据发送中断使能位，使能数据从发送缓存寄存器移入到移位寄存器产生中断</p> <p>0: 禁止数据发送产生中断</p> <p>1: 使能数据发送产生中断</p>	R/W	0

3	RX_EN1	接收数据使能 0: 禁止数据接收 1: 使能数据接收	R/W	0
2	TX_EN1	发送数据使能 0: 禁止数据发送 1: 使能数据发送	R/W	0
1	DIRSEL1	数据编码方式选择位 0: LSB 先传的正逻辑数据编码方式 1: MSB 先传的负逻辑数据编码方式	R/W	0
0	EN1	ISO7816 控制器使能位 0: 控制器关闭 1: 控制器打开	R/W	0

● ISO7816_CLK (0x08)

表 15-5 ISO7816 时钟控制寄存器 1 ISO7816_CLK

比特位	名称	描述	读/写标志	复位值
31:4	---	只读, 不可写	R	0
3	CLKO_EN	卡时钟输出使能位 0: 关闭卡时钟输出 1: 使能卡时钟输出	R/W	0
2:0	CLKDIV	ISO7816 时钟输出 CLK_O 的分频系数 ISO7816 模块的源时钟从系统时钟 fsyspll 上直接取得 000: 不分频; 001: 2 分频; 010: 4 分频; 011: 8 分频 100: 16 分频; 101: 32 分频; 110: 64 分频; 111: 128 分频	R/W	0

● ISO7816 波特率系数 0 寄存器 (0x0c)

表 15-6 ISO7816 波特率系数 0 寄存器 ISO7816_BDDIV0

比特位	名称	描述	读 / 写 标志	复位值
31:22	---	只读, 不可写	R	0
21	FDS0_EN	使能软件配置 F/D 的系数。 1: 波特率系数通过软件写入 FDS0 来确定 0: 波特率系数通过 FD0 来确定	R/W	0
20:8	FDS0	软件配置的波特率系数值, 该位只在 FDS0_EN=1 时可写, 其它情况下均为 13'd372。	R/W	13'd372
7:0	FD0	复位应答所传送的 8 位 FI 和 DI	R/W	8'h01

● ISO7816 波特率系数 1 寄存器 (0x10)

表 15-7 ISO7816 波特率系数 1 寄存器 ISO7816_BDDIV1

比特位	名称	描述	读 / 写 标志	复位值
31:22	---	只读, 不可写	R	0
21	FDS0_EN	使能软件配置 F/D 的系数。 1: 波特率系数通过软件写入 FDS0 来确定 0: 波特率系数通过 FD0 来确定	R/W	0

20:8	FDS0	软件配置波特率系数，只在 FDS0_EN=1 时可写，其它情况下均为 372	R/W	13'd372
7:0	FD0	复位应答所传送的 8 位 FI 和 DI	R/W	8'h01

● ISO7816 状态 0 寄存器 (0x14)

表 15-8 ISO7816 状态 0 寄存器 ISO7816_STA0

比特位	名称	描述	读/写标志	复位值
31:12	---	只读，不可写	R	0
11	FRAME_ERR0	接收数据帧格式错误中断标志位，该位写 1 清零， 1：发送接收数据帧格式错误，当传输错误中断使能位有效时产生中断 0：未发送接收数据帧格式错误	R/W	0
10	BDDIV_R0	波特率匹配为指示。FI 和 DI 是否匹配指示；FD 默认为 8'h01，时钟匹配，当写入的 FD 值不匹配时该位置 1。 1：匹配 0：不匹配	R	1
9	TX_FLAG0	发送缓冲区空标志。上电复位后自动置位，表示缓冲区空，可以写入数据。MCU 写入数据后标志自动清除，数据从发送缓冲寄存器移入移位寄存器后置 1。 1：数据发送缓冲区空 0：数据发送缓冲区内有数据待发送	R	1
8	RX_FLAG0	数据缓冲区满标志，7816 接口控制器每收到 1byte 数据，硬件自动置位，表示接收到 1byte 数据，读数据接收缓冲寄存器清零。 1：接收到 1byte 数据，数据接收缓冲区满 0：未接收到数据，数据接收缓冲区空	R	0
7	RXBUSY0	数据接收忙标志位。硬件置位，软件清零 硬件自动清位置位 0：数据接收空闲 1：接收移位寄存器正在接收数据，收到起始位后自动置 1，收到停止位后自动清零	R	0
6	TXBUSY0	数据发送忙标志。硬件置位，软件清零 硬件自动清位置位 0：数据发送空闲 1：发送移位寄存器正在发送数据，发送起始位时置 1，发送停止位时自动清零	R	0
5	TXPAR_ERRIF0	发送数据奇偶校验错误标志位，重发次数到后仍然奇偶校验错误则该位置位。 对该位写 1 将清零 1：发送数据时发生奇偶校验错误 0：发送数据时无奇偶校验错误	R/W	0
4	RXPAR_ERRIF0	接收数据奇偶校验错误标志位，重发次数到后仍然奇偶校验错误则该位置位。 对该位写 1 将清零 1：接收数据时发生奇偶校验错误	R/W	0

		0: 接收数据时无奇偶校验错误		
3	COL_IF0	发送数据冲突错误中断标志位。硬件置位，软件清零 对该位写 1 将清零 0: 没有中断 1: 中断发生	R/W	0
2	OVL_IF0	接收数据溢出错误标志位。硬件置位，软件清零 对该位写 1 将清零 0: 无溢出错误 1: 中断发生，接收缓冲寄存器未被读出，又接收到新的数据，溢出错误标志有效	R/W	0
1	RXIF0	数据接收中断标志位。数据从移位寄存器移入接收缓存寄存器后置 1。硬件置位，软件清零 对该位写 1 将清零 0: 没有中断 1: 中断发生	R/W	0
0	TXIF0	数据发送中断标志位。数据从发送缓冲寄存器移入移位寄存器后置 1。硬件置位，软件清零 对该位写 1 将清零 0: 没有中断 1: 中断发生	R/W	0

● ISO7816 状态 1 寄存器 (0x18)

表 15-9 ISO7816 状态 1 寄存器 ISO7816_STA1

比特位	名称	描述	读/写标志	复位值
31:14	---	只读，不可写	R	0
13	CARD_OUT_FLAG	CARD_CHECK_EN 使能后该位有效，该位写 1 清零。 1: 检测到卡拔出（检测到输入端口的高电平脉宽大于 40mS） 0: 未检测到卡拔出（检测到输入端口的高电平脉宽不大于 40mS）	R	0
12	OLD_FLAG	OLD_EN 使能后该位有效，为配合 RA9105 的接收到 OLD 信号的中断标志位，该位写 1 清零。 1: 接收到 OLD 信号。 0: 未接收到 OLD 信号。	R/W	0
11	FRAME_ERROR0	接收数据帧格式错误中断标志位，该位写 1 清零 1: 发送接收数据帧格式错误，当传输错误中断使能位有效时产生中断 0: 未发送接收数据帧格式错误	R/W	0
10	BDDIV_R1	波特率匹配为指示。FI 和 DI 是否匹配指示；FD 默认为 8'h01，时钟匹配，当写入的 FD 值不匹配时该位置 1。 1: 匹配 0: 不匹配	R	1
9	TX_FLAG1	发送缓冲区空标志。上电复位后自动置位，表示缓冲区空，可以写入数据。MCU 写入数据后标志自动清除，数据从发送缓冲寄存器移入移位寄存器后置 1。	R	1

		1: 数据发送缓冲区空 0: 数据发送缓冲区内有数据待发送		
8	RX_FLAG1	数据接收完成标志, 7816 接口控制器每收到 1byte 数据, 根据接收的通道相应发出一次中断。硬件置位, 读数据接收缓冲寄存器清零。 1: 接收到 1byte 数据, 数据接收缓冲区满 0: 未接收到数据, 数据接收缓冲区空	R	0
7	RXBUSY1	数据接收忙标志位。硬件置位, 软件清零 硬件自动清置位 0: 数据接收空闲 1: 接收移位寄存器正在接收数据, 收到起始位后自动置 1, 收到停止位后自动清零	R	0
6	TXBUSY1	数据发送忙标志。硬件置位, 软件清零 硬件自动清置位 0: 数据发送空闲 1: 发送移位寄存器正在发送数据, 发送起始位时置 1, 发送停止位时自动清零	R	0
5	TXPAR_ER RIF1	发送数据奇偶校验错误标志位。硬件置位, 软件清零 对该位写 1 将清零 1: 发送数据时发生奇偶校验错误 0: 发送数据时无奇偶校验错误	R	0
4	RXPAR_ER RIF1	接收数据奇偶校验错误标志位。硬件置位, 软件清零 对该位写 1 将清零 1: 接收数据时发生奇偶校验错误 0: 接收数据时无奇偶校验错误	R/W	0
3	COL_IF1	发送数据冲突错误中断标志位。硬件置位, 软件清零 对该位写 1 将清零 0: 没有中断 1: 中断发生	R/W	0
2	OVL_IF1	接收数据溢出错误标志位。硬件置位, 软件清零 对该位写 1 将清零 0: 无溢出错误 1: 中断发生, 接收缓冲寄存器未被读出, 又接收到新的数据, 溢出错误标志有效	R/W	0
1	RXIF1	数据接收中断标志位。数据从移位寄存器移入接收缓存寄存器后置 1。硬件置位, 软件清零 对该位写 1 将清零 0: 没有中断 1: 中断发生	R/W	0
0	TXIF1	数据发送中断标志位。数据从发送缓冲寄存器移入移位寄存器后置 1。硬件置位, 软件清零 对该位写 1 将清零 0: 没有中断 1: 中断发生	R/W	0

● ISO7816 数据 0 寄存器 (0x1C)

表 15-10 ISO7816 数据 0 寄存器 ISO7816_DATA0

比特位	名称	描述	读 / 写 标志	复 位 值
保留	--	只读, 不可写	R	0
8	DATA0[8]	当 <code>parsel</code> 为用户自定义模式时, 为数据帧中的 PARITY 位	R/W	0
7:0	DAT0	数据寄存器 0	R/W	0

● ISO7816 数据 1 寄存器 (0x20)

表 15-11 ISO7816 数据 1 寄存器 ISO7816_DATA1

比特位	名称	描述	读 / 写 标志	复 位 值
保留	--	只读, 不可写	R	0
8	DATA1[8]	当 <code>parsel</code> 为用户自定义模式时, 为数据帧中的 PARITY 位	R/W	0
7:0	DAT1	数据寄存器 1	R/W	0

17.3 7816 与 ESAM 通讯操作步骤

- 1、7816 与 ESAM 通讯, 不需考虑隔离问题, 数据 IO 可共用 1 根线, 建议使用 7816 模块 0。
- 2、配置系统控制章节模块使能 0 寄存器 `MOD0_EN`, 将第 13 位置为 1, 开启 7816 时钟。
- 3、配置 7816 控制寄存器为 `ISO7816_CTL0`, 使用国网 ESAM, 寄存器可配置为 `0x00000201`。
- 4、清除 ISO7816 状态寄存器 `ISO7816_STAT0`。
- 5、打开 7816 总线时钟, 例如当系统时钟为 3.6864MHZ 时, `ISO7816->CLK=0x09`; 此时 7816 模块时钟为 1.8432MHZ。
- 5、可通过中断方式或查询状态方式向 7816 总线读写数据。

17.4 7816 与卡通讯操作步骤

- 1、对于卡表, 卡需与主电进行隔离, 要提供 1-5MHZ 时钟方能正常工作, 现有 SOC 大部分使用高速光藕方案进行主电与卡的隔离, 此方案成本较高, 并且使用高速光耦隔离高频时钟, 在低温下的可靠性较弱。我们提供了一个专用芯片 `RN8501`, 用于与卡的连接, 此芯片使用 2 个普通光耦与 SOC 进行数据通讯。
- 2、配置系统控制章节模块使能 0 寄存器 `MOD0_EN`, 将第 13 位置为 1, 开启 7816 时钟。
- 3、配置 7816 控制寄存器为 `ISO7816_CTL1`, 因与卡通讯使用光藕隔离, 接收与发送分开, `ISO7816->CTRL1` 可配置为 `0x60000201`。
- 4、清除 ISO7816 状态寄存器 `ISO7816_STAT0`。
- 5、打开 7816 总线时钟, 例如当系统时钟为 3.6864MHZ 时, `ISO7816->CLK=0x09`; 此时 7816 模块时钟为 1.8432MHZ。
- 6、可通过中断方式或查询状态方式向 7816 总线读写数据。

使用 `RN8501` 后卡的插入及拔出检测及卡的复位信息读取会与分离方案有所区别:

- 1、卡的插入检测: 将卡座的检测脚连接到 `RN8501` 的 `CHK`, 当卡插入卡座时, `CHK` 引脚为低电平, `RN8501` 通过与 `RN821x` 连接的 7816 口发送 9MS 左右的低电平信号, 当 `ISO7816_CTL1` 的 `OLD` 检测使能打开后, 就会产生中断, 认为外部有卡插入。
- 2、卡的拔出检测: 配置 `ISO7816_CTL1` 的卡拔出检测使能位后, 可检测卡的拔出。注意: 卡操作完成后方能开启卡拔出检测中断。
- 3、卡的复位信息的读取在 `RN821x` 与 `RN8501` 通讯握手完成后读取。
- 4、更具体操作步骤可参考 `RN8501` 数据手册。

18 IIC 接口

SoC 内置一个 I²C 接口控制器。

18.1 概述

I²C 接口控制器具备如下特性：

- ◎ 支持主模式和从模式；
- ◎ 支持 7-bit 地址；
- ◎ 支持多种分频比设置；
- ◎ 支持 100kbps 和快速模式 400kbps；

18.2 寄存器描述

表 16-1 I²C 寄存器基址

模块名	物理地址	映射地址
I ² C	0x40024000	0x40024000

表 16-2 I²C 寄存器偏移地址

寄存器名	地址偏移量	描述
I ² C_CTRL	0x0	控制寄存器
I ² C_CLK	0x4	时钟配置寄存器
I ² C_STA	0x8	状态指示寄存器
I ² C_ADDR	0xC	从设备地址寄存器
I ² C_DATA	0x10	收发数据寄存器

- 控制寄存器（0x0）

表 16-3 控制寄存器 I²C_CTRL

比特位	名称	描述	读/写标志	复位值
31:6	---	只读，不可写	R	0
5	MODE	模式选择位 1：主机模式 0：从机模式	R/W	0
4	ACK	ACK 发送使能 1：接收到第九个 SCL 的时候，产生 ACK 0：接收到第九个 SCL 的时候，不产生 ACK	R/W	0
3	IRQE	I ² C 中断使能 0：禁止中断 1：使能中断	R/W	0
2:1	BUSCON	总线控制产生位，开始命令在总线为空闲状态或者主机是发送状态时有效。结束命令在主机是发送状态时有效。 当检测到 start 或者 stop 时序时，对命令位清零， 00：没有动作 01：产生 START 时序 10：产生 STOP 时序 11：保留	R/W	0

0	EN	模块使能 1: I ² C 打开 0: I ² C 关闭	R/W	0
---	----	--	-----	---

● 时钟配置寄存器 (0x4)

表 16-4 时钟配置寄存器 I²C_CLK

比 特 位	名称	描述	读 / 写 标志	复位值																					
31:3	---	只读，不可写	R	0																					
2:0	CLKDIV	<p>I²C 时钟分频参数选择位： I²C 通信时钟速率计算公式为： SCL=APBCLK/m，其中 m 由 CLKDIV 产生，如下表。 根据不同的系统主频和分频参数选择位产生 I²C 高速/正常模式通信时钟。 当配置选项不在下表时，默认选择为 10 分频。</p> <table><tr><th>系统主频</th><th colspan="2">分频参数 /CLKDIV(m)</th></tr><tr><td></td><th>高速模式</th><th>正常模式</th></tr><tr><td>1.8432Mhz</td><td>不支持</td><td>010（20）</td></tr><tr><td>3.6864Mhz</td><td>001（10）</td><td>011（38）</td></tr><tr><td>7.3728Mhz</td><td>010（20）</td><td>100（76）</td></tr><tr><td>14.7456Mhz</td><td>011（38）</td><td>101（152）</td></tr><tr><td>29.4912Mhz</td><td>100（76）</td><td>110（304）</td></tr></table>	系统主频	分频参数 /CLKDIV(m)			高速模式	正常模式	1.8432Mhz	不支持	010（20）	3.6864Mhz	001（10）	011（38）	7.3728Mhz	010（20）	100（76）	14.7456Mhz	011（38）	101（152）	29.4912Mhz	100（76）	110（304）	R/W	001
系统主频	分频参数 /CLKDIV(m)																								
	高速模式	正常模式																							
1.8432Mhz	不支持	010（20）																							
3.6864Mhz	001（10）	011（38）																							
7.3728Mhz	010（20）	100（76）																							
14.7456Mhz	011（38）	101（152）																							
29.4912Mhz	100（76）	110（304）																							

● 状态指示寄存器 (0x8)

表 16-5 状态指示寄存器 I²C_STAT

比 特 位	名称	描述	读 / 写 标志	复 位 值
31:9	---	只读, 不可写	R	0
8	DIR	读写方向标志 1: 读。 0: 写。	R	0
7	MATCH	地址匹配, 检测到 start 或者 stop 时序后会清零 0: 地址不匹配 1: 地址匹配	R	0
6	BUSY	通讯状态标志 0: IIC 处于空闲状态 1: IIC 处于正常通讯状态	R	0
5	COL	发送冲突中断标志。写 1 清零 发送数据寄存器不为空或接收数据时, 用户向数据寄存器写新的数据, 将触发发送冲突中断标志。 0: 没有触发发送冲突中断 1: 触发发送冲突中断	R/W	0
4	OVERF	接收溢出中断标志。写 1 清零 接收数据时, 当上一个接收数据未被取走前又收到新的数据, 将触发溢出中断标志	R/W	0

		0: 没有触发溢出中断 1: 触发溢出中断		
3	TXEMPT	发送数据寄存器空错误标志。写 1 清零 从模式下, 主机要求从机发送数据, 但发送缓冲区为空时, 触发发送数据寄存器空错误中断标志 0: 没有发生发送数据寄存器空错误 1: 触发发送数据寄存器空错误中断	R/W	0
2	TRANC	传输完成中断标志。写 1 清零 发送数据时发送缓存为空或者接收数据时接收缓存满, 触发传输完成中断标志 0: 传输未完成 1: 传输已完成	R/W	0
1	RX_NACK	收到 NACK 中断标志。写 1 清零 1: 收到 nack 0: 没有接收到 nack	R/W	0
0	STPD	STOP 时序检测中断标志。写 1 清零 关闭模块或者接收到 START 时序后, 该位自动清零 0: 没有检测到 STOP 时序 1: 检测到 STOP 时序	R/W	0

● 从设备地址寄存器 (0xC)

表 16-6 从设备地址寄存器 I²C_ADDR

比 特 位	名称	描述	读/写标志	复 位 值
31:8	---	只读, 不可写	R	0
7:1	SADR	设备地址, 在传输地址期间不可写 主机模式时, 表示从设备的地址; 从机模式时, 该地址用来与主机发来的地址进行比较	R/W	0
0	RW	主机读写方向控制位 0: 写 1: 读	R/W	0

● 收发数据寄存器 (0x10)

表 16-7 收发数据寄存器 I²C_DATA

比 特 位	名称	描述	读/写标志	复 位 值
31:8	---	只读, 不可写	R	0
7:0	TRDATA	接收/发送数据	R/W	0

19 SPI 接口（修改）

芯片集成了 4 个具备普通 DMA 的 SPI M/S 接口，支持 SPI 全双工模式，用于与外部 SPI 接口的设备通信，可以编程实现主模式和从模式工作。

19.1 概述

SPI 接口控制器具备如下特性：

- ◎ 支持 SPI 全双工模式；
- ◎ 支持主模式和从模式工作；
- ◎ 支持时钟的极性和相位设定；
- ◎ 支持发送和接收独立双缓冲区；
- ◎ 支持 LSB 和 MSB 传输模式 8 位，16 位，32 位可配置；
- ◎ 支持 256 种波特率可设，最高 14.7456MHz（fcpu=29.4912MHz）；
- ◎ 从机低速和高速模式可配，低速模式从机最高支持 fcpu/8；高速模式支持 fcpu/4；
- ◎ 支持数据传输完成中断；
- ◎ 支持数据传输冲突中断；
- ◎ 支持 SCSN 模式错误中断；
- ◎ 支持 DMA 功能

19.2 复用关系

SPI0：P44~P47、P110~P113；

SPI1：P52~P55、P20~P23、P40~P43；

SPI2：P110~P113、P40~P43；

SPI3：P110~P113、P20~P23、P47&P46&P84&P83

同一个 SPI，不可同时复用到不同的 IO 上使用，否则数据紊乱，设计注意规避。

19.3 功能描述

SPI 接口符合标准的 SPI HOST 协议，SPI 时钟工作方式通过 CPOL(Clock Polarity)和 CPHA(Clock Phase)参数设置：CPOL 决定时钟的前边沿是上升沿还是下降沿，CPHA 决定时钟的前边沿是数据采样还是数据建立。

详细的工作模式如下表：

表 17-1 SPI 时钟工作方式

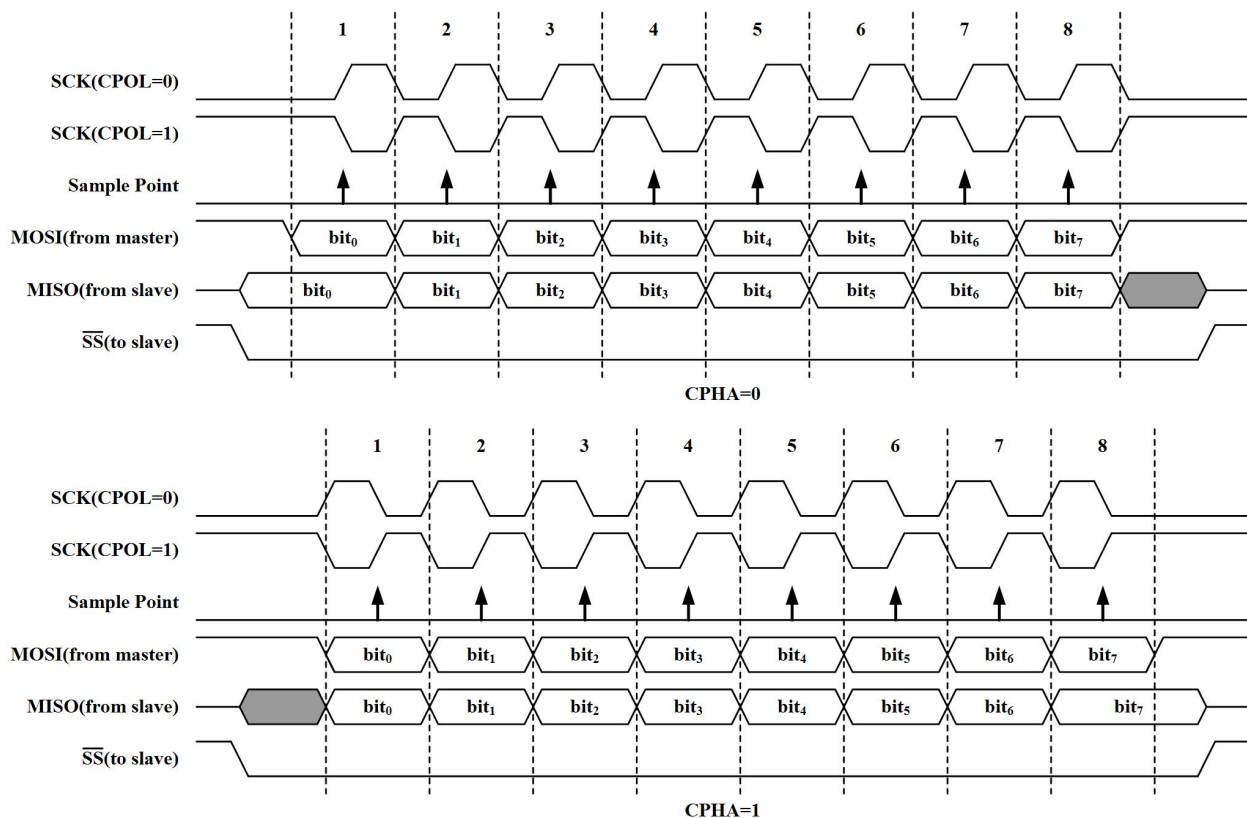
SPI 模式	CPOL/CPHA	前边沿	后边沿
0	0/0	上升沿，数据采样	下降沿，数据建立
1	0/1	上升沿，数据建立	下降沿，数据采样
2	1/0	下降沿，数据采样	上升沿，数据建立
3	1/1	下降沿，数据建立	上升沿，数据采样

数据传输大小支持 8/16/32bit 宽度，SPI 时钟源来自系统时钟，经过一个分频系数后产生通讯时钟。

支持数据发送冲突中断、数据接收溢出中断、传输结束中断和 SS 模式出错中断等四种中断。

发送数据冲突，当一次数据发送正在进行中（txbusy 为 1），此时总线又有一次写命令，则 TXCOLIF 置 1，若 COL_IRQ_EN=1，则会产生中断，同时该发送命令不会响应，正在发送的数据会正常传输完成。

图 14-2 SPI 时钟工作方式



接收数据溢出：在下次完整的接收数据进入移位寄存器之前，没有读取 RXDATA 寄存器，将产生接收数据溢出，则 RXCOLIF 置 1，若 COL_IRQ_EN=1，则会产生中断，同时新的接收数据会保存到接收数据寄存器中，原来没被读走的数据将被覆盖。

传输结束中断：当传输结束时（sck_end），若 TR_IRQ_EN=1,则会产生一个中断，同时 TRIF 置 1。

SCSN 模式出错中断：从模式下，SCSN 必须作为输入，在数据传输过程中 SCSN 变高，则 SCSN 模式出错标志置 1；在主模式下，只有使能主模式 SCSN 模式错误检测（SCSN_EN=1），同时 SCSN 输入为低，则 SCSN 模式出错标志置 1。一旦 SCSN 模式出错标志为 1，则会终止正在进行的传输同时复位 SPI 模块，如果 ERR_IRQ_EN=1 还会产生一个中断。

注意：在配置完控制寄存器后，做为主模式时，只有往数据发送寄存器写数据，才会启动 SPI 读写操作。

19.4 寄存器描述

表 17-2 SPI 寄存器基址

模块名	物理地址	映射地址
SPI0	0x40020000	0x40020000
SPI1	0x40050000	0x40050000
SPI2	0x40054000	0x40054000
SPI3	0x40058000	0x40058000

表 17-3 SPIx 寄存器偏移地址

寄存器名(x 指“0~3”)	地址偏移量	描述
SPIx_CTRL	0x0	控制寄存器
SPIx_STAT	0x4	状态指示寄存器
SPIx_TXDATA	0x8	数据发送寄存器
SPIx_RXDATA	0xC	数据接收寄存器
SPIx_TXDFLT	0x10	SPI 发送数据为空时，默认发送

		数据配置
SPIx_DMA_CTRL	0x14	SPIx DMA 控制寄存器
SPIx_DMA_TBADR	0x18	SPIx DMA 发送起始地址寄存器
SPIx_DMA_RBADR	0x1c	SPIx DMA 接收起始地址寄存器
SPIx_DMA_TLEN	0x20	SPIx DMA 发送长度寄存器
SPIx_DMA_RLEN	0x24	SPIx DMA 接收长度寄存器
SPIx_DMA_TADR	0x28	SPIx 当前发送 DMA 地址寄存器
SPIx_DMA_RADR	0x2c	SPIx 当前接收 DMA 地址寄存器
SPIx_DMA_IE	0x30	SPIx DMA 中断使能寄存器
SPIx_DMA_FLG	0x34	SPIx DMA 中断标志寄存器

19.4.1 SPI 控制寄存器 SPIx_CTRL (0x0)

比特位	名称	描述	读/写标志	复位值
31:27	---	只读，不可写	R	0
26	SLV_TX_ADV	SPI 从机高速模式使能位 =0, SPI 从机低速模式, fcpu/8, 向下兼容 =1, SPI 从机高速模式, 从机速率可达到 fcpu/4。 备注: 从机高速模式, 不考虑路径延时, 从机速率 可达到 fcpu/2。	R/W	0
25	SCSN_POS_IRQEN	SCSN_POS 中断使能 0, 关闭中断 1, 打开中断	RW	0
24	SCSN_NEG_IRQEN	SCSN_NEG 中断使能 0, 关闭中断 1, 打开中断	RW	0
23	TXEMPT_IRQEN	TXEMPT 中断使能 0, 关闭中断 1, 打开中断	RW	0
22	TX_DFLT_EN	发送 BUF 空时, 是否发送 SPI_TXDFLT 寄存器数 据。 0: 发送上次传输 SPI_TXDATA 的值 1: 发送 SPI_TXDFLT 寄存器值	RW	0
21	Reserved	保留, 请勿操作	R/W	0
20	Reserved	保留, 请勿操作	R/W	0
19:12	CLKDIV	SCK 时钟分频系数 SCK 频率=系统时钟频率/(2*(CLKDIV + 1))	R/W	0
11:10	WIDTH	数据宽度选择 0: 8bit 1: 16bit 2: 32bit 3: 预留, 8bit	R/W	0
9	SCSN_EN	SCSN 模式错误检测使能, 只适用于主模式	R/W	0

		0: 不使能主模式 SCSN 模式错误检测, SCSN 为通用 IO 1: 使能主模式 SCSN 模式错误检测, SCSN 作为 SPI 的输入		
8	CPHA	时钟相位选择 0: 前边沿采样数据 1: 前边沿建立数据	R/W	0
7	CPOL	时钟极性选择 0: “SCK” 在空闲状态时被设置为低电平 1: “SCK” 在空闲状态时被设置为高电平	R/W	0
6	LMSB	LSB/MSB 选择 0: MSB 先传输 1: LSB 先传输	R/W	0
5	TXCOL_IRQ_EN	数据冲突中断使能 0: 关闭写冲突中断 1: 打开写冲突中断	R/W	0
4	RXCOL_IRQ_EN	数据冲突中断使能信号 0: 关闭读冲突中断 1: 打开读冲突中断	R/W	0
3	ERR_IRQ_EN	SCSN 模式错误中断使能 0: 关闭模式错误中断 1: 打开模式错误中断	R/W	0
2	TR_IRQ_EN	数据传输中断使能 0: 关闭发送数据中断 1: 打开发送数据中断	R/W	0
1	MAST/SLAV	主从选择 1: MASTER 0: SLAVE	R/W	1
0	EN	使能信号 0: 关闭 SPI 接口 1: 打开 SPI 接口	R/W	0

19.4.2 SPI 状态寄存器 SPIx_STAT (0x4)

比特位	名称	描述	读/写标志	复位值
31:8	---	只读, 不可写	R	0
7	SCSN_POS	作为从机时, CSN 拉高事件标志。 0: 未发生 CSN 拉高事件 1: 发生了 CSN 拉高事件	R/W	0
6	SCSN_NEG	作为从机时, CSN 拉低事件标志。 0: 未发生 CSN 拉低事件 1: 发生了 CSN 拉低事件	R/W	0
5	TXEMPT	发送数据空冲突标志位。当 BUF 为空时, SPI 总线需要发送数据时, 发生写数据空冲突 0: 没有发送数据空冲突中断	R/W	0

		1: 产生发送数据空冲突中断		
4	TXBUSY	数据发送忙状态标志。 0: 数据发送空闲, 总线可以发写 SPITX 寄存器命令 1: 数据正在发送过程中, 总线不能发写 SPITX 寄存器命令	R	0
3	TXCOLIF	写冲突标志。写 1 清零 正在发送 (即 TXBUSY 为 1) 时, 用户向 SPI 写入新的发送数据, 则新的发送数据将被丢弃, 并置写冲突标志为 1。 0: 没有写数据冲突中断 1: 产生写数据冲突中断	R/W	0
2	RXCOLIF	接收数据溢出标志。写 1 清零 连续数据接收时, 如用户不读取 RXDATA 寄存器, 将产生接收数据溢出事件 0: 没有接收数据溢出中断 1: 产生接收数据溢出中断	R/W	0
1	ERRIF	SCSN 模式冲突中断标识位: SPI 为主模式, 只有在 SCSN_EN 为 1 时, 同时检测到 “SCSN” 输入电平为低, 则该位置 1; SPI 为从模式, “SCSN” 作为从机的片选输入, 在数据传输过程中, 若 “SCSN” 输入电平为高, 则该位置 1; 若 ERR_IRQ_EN=1, 则会产生一个中断, 一旦发生模式冲突错误, 则 SPI 模块复位。该位写 1 清零。 0: 没有模式冲突中断 1: 产生模式冲突中断	R/W	0
0	TRIF	数据传输中断标识位, 若数据传输结束, 该位置 1, 若 TR_IRQ_EN=1, 则会产生中断, 该位写 1 清零。 0: 没有数据发送中断 1: 产生数据发送中断, 发送数据寄存器为空	R/W	0

19.4.3 SPI 数据发送寄存器 SPIx_TXDATA (0x8)

比特位	名称	描述	读/写标志	复位值
31:0	TXDATA	数据发送寄存器	R/W	0

19.4.4 SPI 数据接收寄存器 SPIx_RXDATA (0xC)

比特位	名称	描述	读/写标志	复位值
31:0	RXDATA	数据接收寄存器	R	0

19.4.5 SPI 默认发送数据寄存器 SPIx_TXDFLT (0x10)

比特位	名称	描述	读/写	复位值
31:0	TXDFLT	默认发送数据寄存器	R	0

19.4.6 SPI DMA 控制寄存器 SPIx_DMA_CTRL (0x14)

位	位名称	功能描述	读写标志	复位值
31:4	Reserved	保留。	R	0
3	RX_CYC_MODE	接收循环模式使能	R/W	0
2	TX_CYC_MODE	发送循环模式使能	R/W	0
1	RX_DMA_EN	接收 DMA 使能	R/W	0
0	TX_DMA_EN	发送 DMA 使能	R/W	0

19.4.7 SPI DMA 发送起始地址寄存器 SPIx_DMA_TBADR (0x18)

位	位名称	功能描述	读写标志	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_TBADR	DMA 发送起始地址 (Byte 地址) 最低两位必须根据具体传输数据宽度配置。	R/W	0

19.4.8 SPI DMA 接收起始地址寄存器 SPIx_DMA_RBADR (0x1C)

位	位名称	功能描述	读写标志	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_RBADR	DMA 接收起始地址 (Byte 地址) 最低两位必须根据具体传输数据宽度配置。	R/W	0

19.4.9 SPI DMA 发送长度寄存器 SPIx_DMA_TLEN (0x20)

位	位名称	功能描述	读写标志	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_TLEN	DMA 发送长度 (Byte 地址) = (n) Byte 最低两位必须根据具体传输数据宽度配置。	R/W	0

19.4.10 SPI DMA 接收长度寄存器 SPIx_DMA_RLEN (0x24)

位	位名称	功能描述	读写标志	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_RLEN	DMA 接收长度 (Byte 地址) = (n) Byte 最低两位必须根据具体传输数据宽度配置。	R/W	0

19.4.11 SPI DMA 当前发送地址寄存器 SPI_DMA_TADR (0x28)

位	位名称	功能描述	读写标志	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_TADR	当前发送 DMA 地址 (Byte 地址)	RO	0

19.4.12 SPI DMA 当前接收地寄存器 SPIx_DMA_RADR (0x2C)

位	位名称	功能描述	读写标志	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_RADR	当前接收 DMA 地址 (Byte 地址)	RO	0

19.4.13 SPI DMA 中断使能寄存器 SPIx_DMA_IE (0x30)

位	位名称	功能描述	读写标志	复位值
31:6	Reserved	保留。	R	0
5	RX_ERR_IE	接收数据覆盖中断使能 =0, 不使能 =1, 使能	R/W	0
4	TX_ERR_IE	发送数据错误中断使能 =0, 不使能 =1, 使能	R/W	0
3	RX_FIE	DMA 接收全满中断使能 =0, 不使能 =1, 使能	R/W	0
2	RX_HIE	DMA 接收半满中断使能 =0, 不使能 =1, 使能	R/W	0
1	TX_FIE	DMA 发送全满中断使能 =0, 不使能 =1, 使能	R/W	0
0	TX_HIE	DMA 发送半满中断使能 =0, 不使能 =1, 使能	R/W	0

19.4.14 SPI DMA 中断标志寄存器 SPIx_DMA_IF (0x34)

位	位名称	功能描述	读写标志	复位值
31:6	Reserved	保留。	R	0
5	RX_ERR	接收数据覆盖标志(该通道 DMA 被其他高优先级 DMA 阻塞未能及时获取 SPIx_RXDATA 数据, 导致数据被覆盖), 写“1”清该位	R/WC	0
4	--	保留	R	0
3	RX_FDONE	DMA 接收全满中断标志, 写“1”清该位	R/WC	0
2	RX_HDONE	DMA 接收半满中断标志, 写“1”清该位	R/WC	0
1	TX_FDONE	DMA 发送全满中断标志, 写“1”清该位	R/WC	0
0	TX_HDONE	DMA 发送半满中断标志, 写“1”清该位	R/WC	0

19.5 应用方法

1、在系统控制章节，使能 SPI 时钟。

2、作为主机：

1) 配置相应 GPIO 复用为 SPI0/1 引脚，注意：主模式应用时，SCSN 引脚配置为 IO 口，用软件来产生 SCSN 电平；

2) 使能 SPI 中断；

3) 初始化 SPI0/1，配置 TXDFLT、CTRL 等寄存器，将 SPI_CTL 的 MAST/SLAV 位置 1，选择 SPI MAST 模式。数据传输的频率由 SPI_CTL 的 CLKDIV 配置，将 SPI_CTL 数据传输中断使能位 TR_IRQ_EN 置 1；

4) 准备好待发送数据，将 SCSN 拉低，将第一个待发送数据写入 SPI_TXDATA，开始发送数据；

5) 若发生 SPI 中断，则进入 IRQ 处理函数，若查询到 SPI_STA.TRIF 为 1，表明传输一次位宽（由 SPI_CTL.WIDTH 配置）数据完成，读取 SPI_RXDATA 寄存器接收数据，并填入下一个待发送的数据到 SPI_TXDATA；

6) 重复 5) 直至待发送数据全部发完，拉高 SCSN；

3、作为从机

1) 配置相应 GPIO 复用为 SPI0/1 引脚，注意：从模式应用时，SCSN 引脚应配置为 SCSN 功能；

2) 使能 SPI 中断；

3) 初始化 SPI0/1，配置 TXDFLT、CTRL 寄存器，将 SPI_CTL 的 MAST/SLAV 位置 0，即选择 SPI SLAVE 模式，数据传输的频率由主机决定，将 SPI_CTL 寄存器的 TR_IRQ_EN、SCSN_NEG 及 SCSN_POS 置 1，准备好待发送数据；

4) 若发生 SPI 中断，则进入 IRQ 处理函数，若查询到 SPIS_STIF.SCSN_NEG_IF 为 1，表示主机发起了 SPI 通信，从机接收 SDI 上的 SPI_RXDATA 并解析，将待发送数据写入 SPI_TXDATA 中；若查询到 SPI_STA.TRIF 为 1 表明传输一次位宽（由 SPI_CTL.WIDTH 配置）数据完成，取出 SPI_RXDATA 寄存器接收数据并填入下一个待发送的数据到 SPI_TXDATA；若查询到 SPI_STA.SCSN_POS_IF 为 1 表明此帧数据传输完毕，取出 SPI_RXDATA 寄存器接收数据。

20 LPUART（新增）

20.1 概述

低功耗通用串口 LPUART 模块能够实现有限功耗的全双工 UART 通信。工作时钟只需要配置为 32.768kHz 就可以达到 9600 波特率通信。LPUART 可以在低频模式下工作实现低功耗异步数据收发。在 CPU 处于 SLEEP 模式时，检测到唤醒事件后，LPUART 可产生唤醒中断快速唤醒 CPU。

LPUART 协议包括两个数据线 LPUART_TX 和 LPUART_RX，其中 LPUART_TX 为发送数据线，LPUART_RX 为接收数据线。该模块的功能主要包括 3 个部分：配置数据帧格式，进行数据串行发送；检测起始位，进行数据接收；从低功耗模式唤醒。

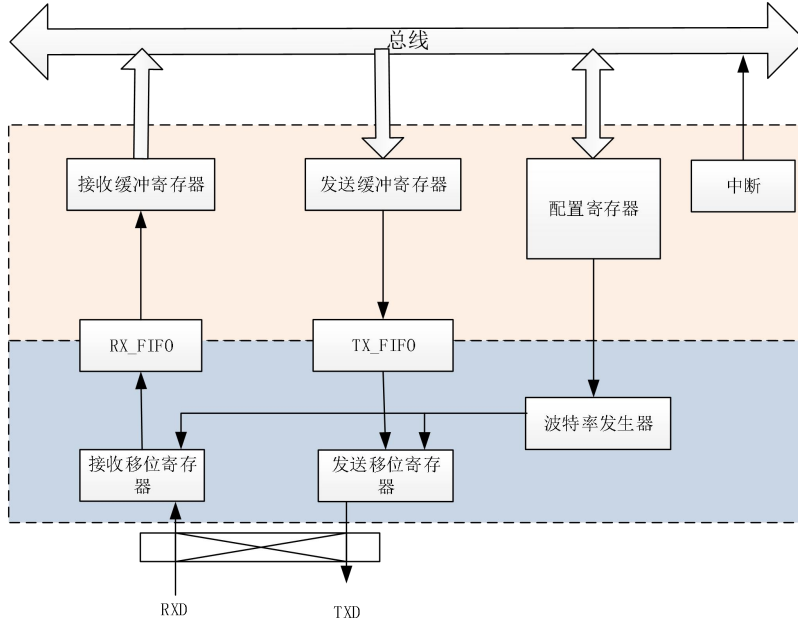
20.2 特点

- 32.768k 工作时钟下最大 9600 波特率，波特率范围 300~9600bps
- 发送接收缓存支持各自独立 4 字节 FIFO
- 支持数据位宽为 5/6/7/8bit，停止位 1bit 或 2bit
- 奇偶校验位 支持奇校验、偶校验、全 0/1、无校验位或用户自定义校验位
- 支持 4 种低功耗唤醒方式：RX 下降沿唤醒、起始位检测唤醒、（单帧或多帧）数据接收完成唤醒、一帧数据匹配唤醒

- 发送和接收中断使能独立
- 支持 38kHz 红外调制输出
- 接收支持空闲帧检测

20.3 功能框图

LPUART 的工作时钟是 32.768KHz，时钟源可以选择内部低频时钟 RCL32k 和外部低频晶振 LOSC。LPUART 寄存器读写时钟为系统时钟。

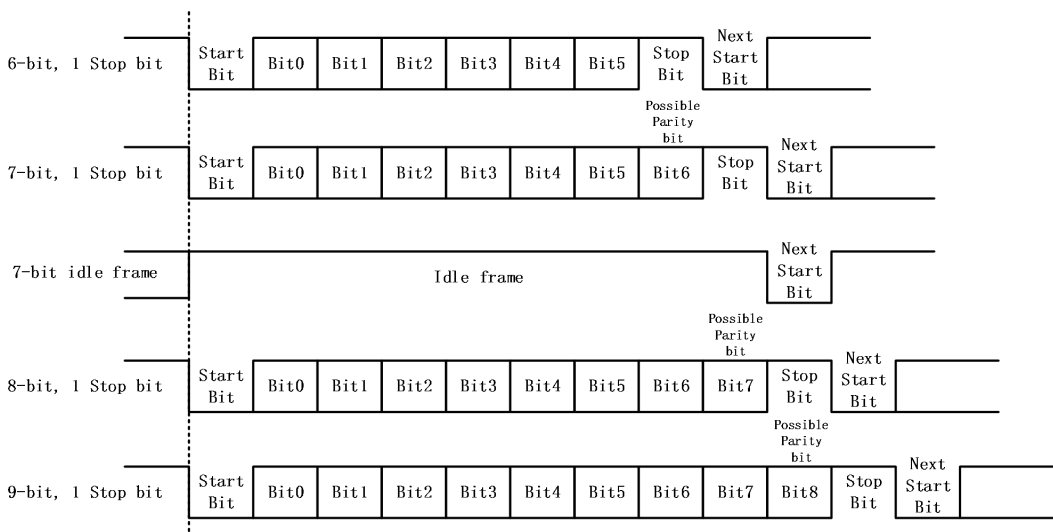


20.4 帧结构

通过对 LPUART_MODE 寄存器的 DATLEN 位进行编程，数据帧的长度可以设置为 5~8 位。一帧数据的格式为：起始位+数据位+校验位+停止位。

默认情况下，TX 和 RX 的起始位是低电平，结束位为高电平，也可以通过极性配置将每位信号极性反转。奇偶校验位设置在数据的最后一位，停止位可以设置为 1bit 或 2bit。若校验位选择用户自定义校验位并设置高位数据先发送，则先发送校验位。

空闲帧是一段包括起始位和停止位的连续一帧数据长度的高电平，从上一帧的停止位结束开始计算，数据的长度不同空闲帧的长度也不同。



20.5 FIFO 功能

LPUART 具有一个宽度 9 位的发送 FIFO 和接收 FIFO，FIFO 深度都是 4。当发送 FIFO 状态为空/半空，接收 FIFO 状态为非空/满/半满都会置起中断标志位，可以配置中断使能。发送和接收 FIFO 分别在发送使能和接收使能开启时重置。

发送端使能时，向 LPUART_TXD 寄存器写入数据的指令会向 TXFIFO 增加一个数据，写入到 LPUART_TXD 寄存器的数据会在 TXFIFO 中排队。当 TXFIFO 置起空和半空标志时都可以继续进行写操作，不需要在每次写入时判断 TXFIFO 是否为满：

- ① 产生 TXFIFO 空标志时软件控制向 TXFIFO 写入 4 字节；
- ② 产生 TXFIFO 半空标志时软件控制向 TXFIFO 写 2 字节。

接收端 RXFIFO 的非空/半满/全满标志均表示数据可以读取，读取 LPUART_RXD 会返回输入到 RXFIFO 中的最早数据。

- ① 当 RXFIFO 全满时可以产生中断软件一次读取 4 字节数据；
- ② 当 RXFIFO 半满时可以产生中断软件一次读取 2 字节数据；
- ③ 当接收端检测到空闲帧时，认定为当前数据包传输结束，自动关闭波特率发生器使能，当中断信号使能时可以产生空闲中断软件读取 RXFIFO 的数据。

20.6 发送引擎

20.6.1 发送特性

发送引擎包括发送控制寄存器、发送缓冲寄存器和发送移位寄存器。发送引擎可以根据控制寄存器的配置发送数据长度、奇偶校验位、停止位宽度。

在第一次发送时向发送控制寄存器设置发送使能位 TE，在数据传输过程中不能重置 TE。在 LPUART 传输期间，TX 引脚默认先发送最低位 LSB。

向 LPUART_TXD 写入要发送的数据时会向 TXFIFO 增加一个数据，写入的数据会在 TXFIFO 中排队。读 LPUART_TXD 寄存器返回 TXFIFO 中最早的数据。

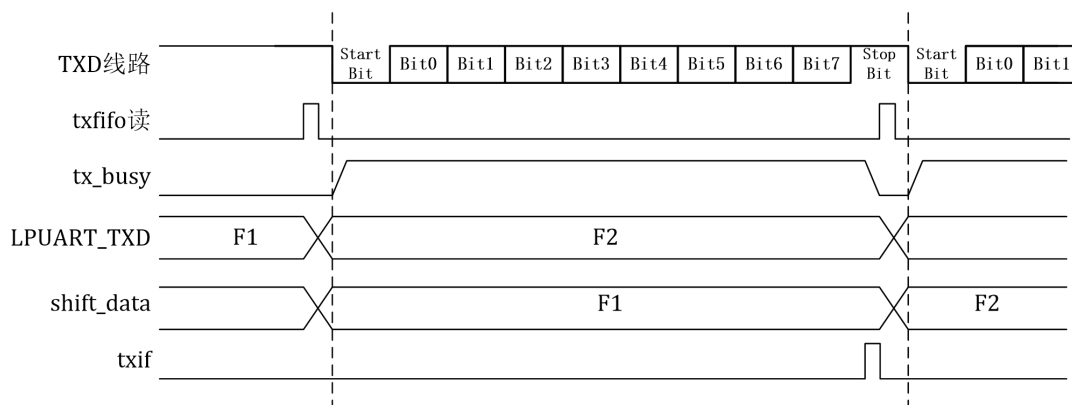
数据全部写完后，必须等待 LPUART_STA 的 TXIF 位和 TXFFIF 位都等于 1，表示全部帧发送完成且 TXFIFO 和移位寄存器均为空，才能进入 SLEEP 模式。

20.6.2 单字节通信流程

TXFNFIF 标志由硬件置 1，表示 TXFIFO 未滿，下一个数据可以写到 LPUART_TXD 寄存器中而且不会覆盖旧数据。

正在发送时，对 LPUART_TXD 寄存器的写入指令将数据存储在 TXFIFO 中，等到当前正在进行的传输结束时，将数据复制到移位寄存器中。TXFIFO 空时，TXFEIF 为 1，表示可向 TXFIFO 写入 4 字节；半空时，TXFHEIF 为 1，表示可继续写入 2 字节。若使能了 TXFIFO 的中断则可以产生空/半空中断。

如果当前帧已经发送完，表示缓冲寄存器空，TXIF 将在一帧的最后一个结束位变为高电平。如果设置了 TXIE 可以产生中断。向寄存器写入最后一个数据后，必须等待 TXIF=1 和 TXFEIF=1 后才能禁用 LPUART 或者 MCU 进入 SLEEP 模式。



单字节发送时序示意图

20.7 接收引擎

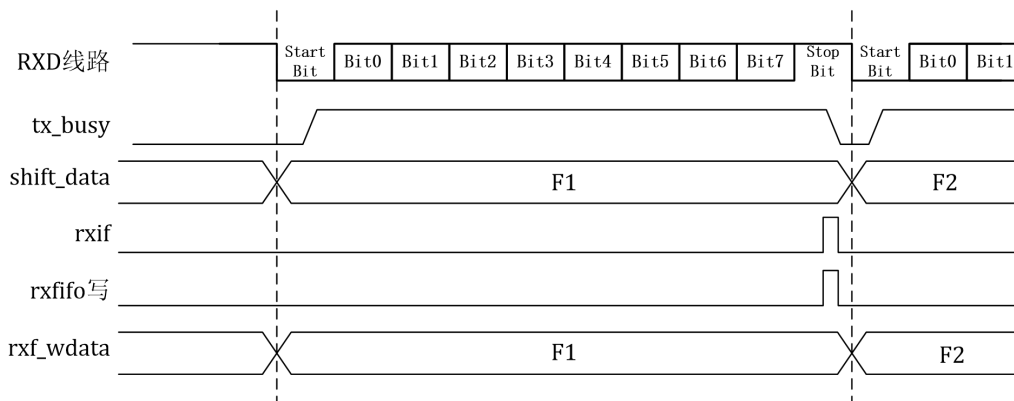
20.7.1 传输特性

数据位宽 5/6/7/8bit，在 LPUART_MODE 寄存器中设置 DATLEN 位和停止位长度 STOPS。

在 LPUART 接收期间，数据首先通过 RX 引脚移位最低位。如果使能内部回环，则将发送端 TX 和接收端 RX 在模块内部短接。

RXFNEIF 位置 1，表示 RXFIFO 非空，读取 LPUART_RXD 会返回输入到 RXFIFO 中的最早数据。在 RXFIFO 为空时，RXFNE 位清零。RXFIFO 已满时，必须在结束接收下一个字符前读取 RXFIFO 中的第一条数据，以免发生溢出错误。

如果接收期间已经检测到帧错误、噪声错误、溢出错误，对应错误标志位置 1。



单字节接收时序示意图

20.7.2 起始位检测

检测 RX 线的下降沿，然后在起始位的中间位置采样，确认维持为 0。如果起始位采样判断为 1，则设置噪声错误标志 NEIF，该起始位被丢弃，接收器等待一个新的起始位。

20.7.3 停止位检测

1 stop bit：在停止位采样位置进行取样

2 stop bit：在第二个停止位的位置取样，第一个停止位不用来检查帧格式错误

20.7.4 空闲帧检测

当使能 RXFIFO 时，如果 RXFIFO 设置了半满或全满标志但在接收完最后一帧后没有达到 FIFO 阈值，则 FIFO 有可能一直在等待接收数据，导致缓存的数据没有被及时读出，需要利用空闲帧检测来判断。

软件使用示例：在接收使能打开，接收到数据时，空闲帧检测开启，当检测到 RX 线高电平超过一帧数据长度时，产生空闲帧中断标志，若使能了中断进入相应的中断服务程序。当标志置起时，停止空闲帧检测，直至下次有数据接收再重新开始检测。若使能了空闲帧中断可以产生中断标志。

20.7.5 接收错误

1) 溢出错误

接收移位寄存器准备好将数据传到 FIFO 且接收 FIFO 已满时会发生溢出错误。在 RXFIFO 中存在一个空闲位置之前，数据无法从移位寄存器传送到 LPUART_RXD 寄存器。RXFIFO 设置了满和半满标志，如果 RXFFIF=1 时还接收新数据就产生溢出错误。发生上溢错误时：OEIF 位置 1，RXFIFO 中的数据不会丢失，移位寄存器将被覆盖，溢出期间接收到的数据都会丢失。

3) 帧格式错误

当结束位没有在预期接收时间被识别（即停止位采样值=0）。检测到帧错误时：FEIF 由硬件设置、无效数据从移位寄存器传输到 LPUART_RXD。单字节通信时，模块会由 RXFNE 位产生中断，如果发生帧格式错误，FEIF 置位。

20.7.6 波特率发生器

在 LPUART 模块中，TX 端波特率选择使用小数分频方式产生，不同波特率对应的寄存器位宽调制配置 MCTL[11:0]需要分别配置，MCTL[0]必须配置为 1。

表：LPUART 位宽调制配置 MCTL[11:0]

波特率/bps	LPUART_BAUD[2:0]	LPUART_BAUD[27:16].MCTL
300	101/110/111	0x111
600	100	0xB6D
1200	011	0x249
2400	010	0x6DB
4800	001	0xF7F
9600	000	0x4A5

20.7.7 奇偶检验位

在 LPUART_MODE 寄存器中设置 PARS 奇偶校验位。在 5/6/7/8bit 长度的数据传输时可以配置校验位。

接收时进行奇偶校验检查：如果奇偶校验检查失败，则 LPUART_STA 寄存器的 PEIF 置 1；如果 PEIE 位为 1 则会产生中断。接收端不检查用户自定义校验位。

发送时的奇偶校验生成：如果 PARS 位不为 000 或用户自定义校验位，则在数据寄存器中所写入的数据会先进行传送，然后再传输一位奇偶校验位（偶校验，1 的数量为偶数；奇校验，1 的数量为奇数）。若发送端选择用户自定义校验位，则在 LMSB=1 时先发送校验位。

20.7.8 低功耗唤醒模式

LPUART 在 Sleep 模式下进行数据接收并唤醒 CPU。在系统内核时钟关闭之前需要使用寄存器配置唤醒方式和唤醒中断使能，直到特定事件到来后唤醒芯片退出睡眠模式。LPUART 支持 4 种低功耗唤醒事件，可以在寄存器中配置：RX 下降沿唤醒、起始位唤醒、数据接收完成唤醒和一帧数据匹配唤醒。

进入低功耗模式之前，LPUART 确保没有进行数据收发（检查 BUSY 标志），进入休眠模式后会关闭内核时钟。LPUART 模块时钟在休眠时不关闭，在检测到 RX 下降沿时自动开启数据接收。还需要提前配置 LPUART_MODE 的 WUSEL 选择唤醒中断事件，将 WUFIE 位置 1 使能 LPUART 低功耗唤醒中断，将 RE 置 1

打开接收使能，再进入 SLEEP 模式。

在进行低功耗唤醒时，为了不丢失数据，需要保证 CPU 在接收数据缓冲寄存器/接收 FIFO 满之前完成唤醒流程，开始将数据写入存储器，否则容易发生溢出错误。

现有的 MCU 运行 Sleep 指令只会关闭 CPU 内核时钟，外设时钟不关闭，CPU 内核从产生唤醒中断至进入 LCMM 模式需要 3 pclk。

20.7.8.1 RX 下降沿唤醒

芯片在休眠模式下仍旧保持 RX 引脚的监听，配置寄存器 LPUART_MODE 的 RX 下降沿检测中断使能 WUSEL。当在休眠模式下接收机检测到 RX 下降沿事件打开模块时钟，自动启动数据接收，并请求唤醒 CPU。下降沿唤醒没有进行起始位的噪声检测。

20.7.8.2 起始位唤醒

配置寄存器进行起始位检测唤醒。在监听到 RX 下降沿后，在起始位中间位置采样值为 0 时判断为起始位，在起始位结束时向 MCU 发送唤醒事件。

20.7.8.3 数据接收完成唤醒

在休眠模式下，监听到 RX 端下降沿后，开始接收数据并缓存，在数据接收完毕后，不进行数据匹配，直接产生中断唤醒 CPU。

LPUART 的唤醒中断源可以设置为：

RXFNEIF 标志（RXFIFO 非空，已接收到 1 字节），RXFNEIE 必须在休眠前置 1。

RXFHFIF 标志（RXFIFO 半满，已接收到 2 字节），RXFHFIE 必须在休眠前置 1。

RXFFIF 标志（RXFIFO 已满，已接收到 4 字节），RXFFIE 必须在休眠前置 1。

20.7.8.4 帧数据匹配唤醒

提前在数据匹配寄存器中配置好 MDATA 数据，在休眠模式下第一帧接收时按位比较第一帧数据与 MDATA 数据，如果数据匹配成功则 MDATIF 标志位为 1，触发中断，可以用于休眠模式下的数据接收唤醒；如果数据匹配失败则忽略收到的数据。在休眠模式时，如果 RXFNEIF=0，则将下一次接收到的数据帧确定为第一帧进行数据匹配。

20.8 寄存器描述

20.8.1 寄存器列表

模块	物理地址	映射地址
LPUART	0x40070000	0x40070000
寄存器名	地址偏移量	描述
LPUART_MODE	0x0	LPUART 模式配置寄存器
LPUART_IE	0x4	LPUART 中断使能寄存器
LPUART_STA	0x8	LPUART 状态指示寄存器
LPUART_BAUD	0xC	LPUART 波特率调制寄存器
LPUART_TXD	0x10	LPUART 发送数据缓冲寄存器
LPUART_RXD	0x14	LPUART 接收数据缓冲寄存器
LPUART_DMR	0x18	LPUART 数据匹配寄存器

注：LPUART_MODE & LPUART_BAUD 在传输期间禁止修改。

20.8.2 LPUART_MODE (0x00)

LPUART 模式配置寄存器

复位值: 0x0

位	位名称	功能描述	读 写 标志	复位值
31:15	---	预留	R	0
14:13	WUSEL	唤醒中断事件配置 00: START 起始位检测 01: RX 下降沿检测 10: 数据接收完成 11: 一帧数据匹配成功	R/W	0
12	ILBE	内部回环使能位 0: 内部回环禁止 1: 内部回环使能, TX 与 RX 在模块内部短接	R/W	0
11	LMSB	LSB/MSB 传输机制选择方式 0: LSB 先传 1: MSB 先传 注: 当 PARS 选择为用户自定义校验时, 校验位当做数据扩展最高位, 若此时选择方式为 MSB, 最先传输的将是校验位	R/W	0
10	STOPS	停止位宽选择 0: 1-bit 停止位 1: 2-bit 停止位	R/W	0
9:7	PARS[2:0]	校验位选择位 000: 无校验 001: 奇校验 010: 偶校验 011: 固定为 0 校验 100: 固定为 1 校验 101/110/111: 用户自定义校验	R/W	0
6:5	DATLEN[1:0]	传输数据宽度位 (不包括校验位) 00: 5-bit 01: 6-bit 10: 7-bit 11: 8-bit	R/W	0
4	IRPOL	红外调制输出极性选择 0: 正极性, 即默认驱动电平保持为高, 低电平调制输出 1: 负极性, 即默认驱动电平保持为低, 高电平调制输出 注: IRSEL 只决定空闲输出 (无效电平) 时的电平高低, 不影响有效数据期间的电平	R/W	0
3	NEG	发送和接收数据极性配置 0: 正极性, 默认驱动电平为高 1: 负极性, 默认驱动电平为低	R/W	0
2	IRE	红外调制输出功能配置 0: 关闭	R/W	0

		1: 打开		
1	RE	LPUART 接收器使能位 0: 关闭 1: 打开	R/W	0
0	TE	LPUART 发送器使能位 0: 关闭 1: 打开	R/W	0

20.8.3 LPUART_IE (0x04)

LPUART 中断使能寄存器

默认复位值: 0x0

位	位名称	功能描述	读 写 标志	复位值
31:14	---	预留	R	0
13	RXFFIE	接收 FIFO 满中断标志使能 0: 不使能 1: 使能	R/W	0
12	RXFHFIE	接收 FIFO 半满中断标志使能 0: 不使能 1: 使能	R/W	0
11	RXFNEIE	接收 FIFO 非空中断标志使能 0: 不使能 1: 使能	R/W	0
10	TXFHEIE	发送 FIFO 半空中断标志使能 0: 不使能 1: 使能	R/W	0
9	TXFEIE	发送 FIFO 空中断标志使能 0: 不使能 1: 使能	R/W	0
8	IDLEIE	空闲帧检测中断使能 0: 关闭中断 1: 打开中断	R/W	0
7	NEIE	起始位噪声检测使能 0: 关闭中断 1: 打开中断	R/W	0
6	DEIE	数据错误使能 0: 关闭中断 1: 打开中断	R/W	0
5	FEIE	帧格式错误使能 0: 关闭中断 1: 打开中断	R/W	0
4	OEIE	溢出错误使能 0: 关闭中断 1: 打开中断	R/W	0

3	PEIE	奇偶校验错误使能 0: 关闭中断 1: 打开中断	R/W	0
2	WUFIE	从休眠模式下唤醒中断使能，必须在进入休眠模式之前打开中断使能，仅在休眠模式下 WUF 有效。唤醒后软件关闭该中断使能。 0: 关闭中断 1: 打开中断	R/W	0
1	RXIE	接收数据完成中断使能 0: 关闭中断 1: 打开中断	R/W	0
0	TXIE	发送数据完成中断使能 0: 关闭中断 1: 打开中断	R/W	0

20.8.4 LPUART_STA (0x08)

LPUART 状态指示寄存器

复位值: 0x0

写入 0x3ff 复位

位	位名称	功能描述	读 写 标志	复位值
31:24	---	预留	R	0
22:20	RXFCNT[2:0]	接收 FIFO 内数据个数	R	0
19:17	TXFCNT[2:0]	发送 FIFO 内数据个数	R	0
16	RXFFIF	接收 FIFO 满中断标志位，写 1 清零 0: 非满 1: 已满	R/W	0
15	RXFHFIF	接收 FIFO 半满中断标志位，表示接收 FIFO 有 2 个数据，写 1 清零 0: 非半满 1: 半满	R/W	0
14	RXFNEIF	接收 FIFO 非空中断标志，写 1 清零 0: 已空 1: 非空	R/W	0
13	TXFF	发送 FIFO 满标志，写 1 清零或硬件自动清零 0: 未滿 1: 已滿	R/W	0
12	TXFHEIF	发送 FIFO 半空中断标志位，表示发送 FIFO 中有 2 个空位，写 1 清零 0: 非半空 1: 半空	R/W	0
11	TXFEIF	发送 FIFO 空中断标志位，写 1 清零 0: 非空	R/W	1

		1: 已空		
10	RBUSY	接收状态标志位 0: 没有接收 1: 正在接收数据	R	0
9	TBUSY	发送状态标志位 0: 没有发送 1: 正在发送标志位	R	0
8	IDLEIF	空闲帧检测中断标志, 写 1 清零 0: 未检测到空闲帧 1: 检测到空闲帧	R/W	0
7	NEIF	起始位噪声检测标志, 写 1 清零 0: 无噪声 1: 有噪声	R/W	0
6	DEIF	数据错误标志, 写 1 清零 LPUART 发送 FIFO 已满后继续写 LPUART 发送寄存器, 该位会置起 0: 无错误 1: 有错误	R/W	0
5	FEIF	帧格式错误标志位, 写 1 清零 LPUART 接收到的数据不符合帧格式标志位, 即接收到的停止位是 0 而不是 1, 该位会置起 0: 无错误 1: 有错误或检测到中断字符	R/W	0
4	OEIF	溢出错误标志位, 写 1 清零 LPUART 接收数据缓冲寄存器没有及时读取导致接收溢出 0: 无错误 1: 有错误	R/W	0
3	PEIF	奇偶校验错误标志位, 写 1 清零 0: 无错误 1: 有错误	R/W	0
2	WUF	接收唤醒事件标志, 写 1 清零 0: 未发生唤醒事件 1: 已发生唤醒事件 唤醒事件通过 WUSEL 选择: RX 下降沿唤醒、起始位检测唤醒、数据接收完成唤醒、一帧数据匹配唤醒	R/W	0
1	RXIF	接收完成中断标志, 写 1 清零 0: 接收数据未完成 1: 接收数据完成, 可从数据缓冲寄存器中读出在每帧接收完成时触发。	R/W	0
0	TXIF	发送完成中断标志, 写 1 清零 0: 发送未完成 1: 发送完成 当前发送字节的 STOP 位已经发送完成后触发。	R/W	0

注: 软件在进入 FIFO 中断后先读写数据, 后清标志位。

20.8.5 LPUART_BAUD (0xC)

LPUART 波特率调制寄存器

复位值: 0x0

位	位名称	功能描述	读 写 标志	复位值
31:28	---	保留	R	0
27:16	MCTL	对应 LPUART 每个 bit 的位宽调制信号。不同波特率 (bps) 对应的位宽调制推荐值: 9600: 0x4A5 4800: 0xF7F 2400: 0x6DB 1200: 0x249 600: 0xB6D 300: 0x111	R/W	0
15:3	---	保留	R	0
2:0	BAUD	波特率控制 (bps) 000: 9600 001: 4800 010: 2400 011: 1200 100: 600 101/110/111: 300 注: 关闭使能时才能修改波特率调制寄存器	R/W	0

20.8.6 LPUART_TXD (0x10)

LPUART 发送数据缓冲寄存器

复位值: 0x0

位	位名称	功能描述	读 写 标志	复位值
31:9	---	预留	R	0
8	UP	用户自定义校验位	R/W	0
7:0	TXDATA	发送数据寄存器 注: 开启发送使能后才能写发送数据	R/W	0

20.8.7 LPUART_RXD (0x14)

LPUART 接收数据缓冲寄存器

复位值: 0x0

位	位名称	功能描述	读 写 标志	复位值
31:9	---	预留	R	0
8	UP	校验位	R	0
7:0	RXDATA	接收数据寄存器	R	0

20.8.8 LPUART_DMR (0x18)

LPUART 数据匹配寄存器

复位值: 0x0

位	位名称	功能描述	读 写 标志	复位值
31:8	---	预留	R	0
7:0	MDATA	与接收到的第一帧数据比较, 如果检测到数据相同, 触发唤醒中断, 用于休眠模式下接收唤醒。	R/W	0

注: 仅在接收不使能时可以写入该寄存器。

20.9 软件操作流程

20.9.1 发送流程

- 1) 配置 LPUART 配置寄存器 LURT_CFG[2:0]选择 LPUART 与 UARTx 复用 IO 配置, 并配置复用管脚;
- 2) 对模块使能 0 寄存器 MOD0_EN[25].LPUART_EN 位写 1 使能 LPUART 模块;
- 3) 设定字符长度、停止位宽度、校验方式、波特率、中断使能等配置;
- 4) 写 0x1fff 清除 LPUART_STA 寄存器状态;
- 5) 配置 LPUART 中断使能和优先级, 编写中断服务程序。LPUART 接收、发送、错误中断为同一中断入口, 需根据中断使能位及状态标志判断此时为何种中断。
- 6) 在第一次发送时向 LPUART_MODE 的 TE 位置 1 使能发送端;
- 7) 向发送数据缓冲寄存器写入数据;
- 8) 等待中断事件, 继续向发送数据缓冲寄存器写入数据。

20.9.2 接收流程

- 1) 配置 LPUART 配置寄存器 LURT_CFG[2:0]选择 LPUART 与 UARTx 复用 IO 配置, 并配置复用管脚;
- 2) 对模块使能 0 寄存器 MOD0_EN[25].LPUART_EN 位写 1 使能 LPUART 模块;
- 3) 设定字符长度、停止位宽度、校验方式、波特率、中断使能等配置;
- 4) 写 0x1fff 清除 LPUART_STA 寄存器状态;
- 5) 配置 LPUART 中断使能和优先级, 编写中断服务程序。LPUART 接收、发送、错误中断为同一中断入口, 需根据中断使能位及状态标志判断此时为何种中断;
- 6) 在第一次接收前向 LPUART_MODE 的 RE 位置 1 使能接收端;
- 7) 等待中断事件, 处理接收的数据和错误。

21 CRC (新增)

21.1 概述

循环冗余校验 (Cyclic redundancy check, CRC) 主要用来检测或校验数据传输或者存储的完整性。CRC 计算单元使用多项式发生器从一个 8bit/16bit/32bit 的数据中产生 CRC 码。

通过软件控制可进行 7816、I2C、UART 和 SPI 模块数据的 CRC 计算。

21.2 特点

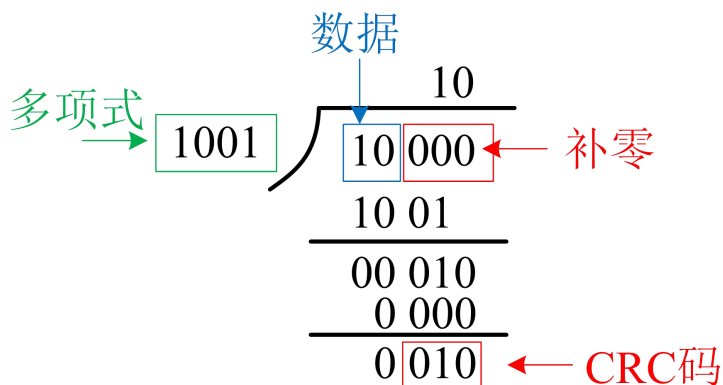
- 支持 7/8/16/32 位 CRC, 支持任意多项式
- 处理 8 位、16 位、32 位数据大小

- 可编程 CRC 初值
- 支持输入数据按字节/半字/全字颠倒
- 支持输出结果反转，异或
- 1 周期完成 8bit CRC 运算，4 周期完成 32bit 运算
- 支持通用 DMA 功能

21.3 基本原理

CRC 的基本原理，利用传输的数据除以另一个数（多项式）得到的余数即为 CRC 码

最基本的 CRC 运算采用模 2 除法，模 2 除法每一位除的结果不影响其他位，即不向上位借位，所以实际上就是按位异或，每次计算需要左移 1 位，当被除数的位数小于除数，则结果为余数。



最基本的 CRC 除法有个很明显的缺陷，就是数据一开始添加一些 0 并不影响最后校验字的结果，并且还需要完整的数据才可以计算出 CRC 码。实际上，真正的 CRC 计算时在原始的 CRC 算法的基础上做了些小改动。增加了两个概念。第一个是余数初始值，第二个是结果异或值。

21.4 多项式

多项式的最高位和最低位肯定为 1，常用的多项式如下：

$$\text{CRC8} = X^8 + X^5 + X^4 + 1$$

$$\text{CRC-CCITT} = X^{16} + X^{12} + X^5 + 1$$

$$\text{CRC16} = X^{16} + X^{15} + X^2 + 1$$

$$\text{CRC32} = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X^1 + 1$$

多项式中的位宽指的是其二进制位数减 1，如 CRC8 的位宽为 8，但二进制数有九位。为了简记多项式的值，采用 16 进制的写法来表示，因为多项式的最高为必定为 1，在简记中会省略最高位的表示，如 CRC-CCITT 的简记为 0x1021，实际上表示的是 0x11021。

21.5 余数初始值

余数初始值就是在计算 CRC 值的开始，给 CRC 寄存器一个初始值。在 CRC 的分段计算中，每一次计算的结果都会作为下一计算的余数初始值。如使用 CRC-CCITT 多项式的 16bit 数据的计算模型，计算 32bit 的数据 0xaabbccdd，首先计算高位 0xaabb 的 CRC 值，然后将该值作为计算 0xccdd 的余数初始值，计算 0xccdd 的 CRC 值，最后的结果等同于 0xaabbccdd 的 CRC 值。

21.6 结果异或值

结果异或值时在其余计算完成后将 CRC 寄存器的值在与这个值进行异或操作后作为最后的校验值。

21.7 输入数据反转

支持输入数据，按字节执行位反转、按半字执行位反转、按字执行位反转

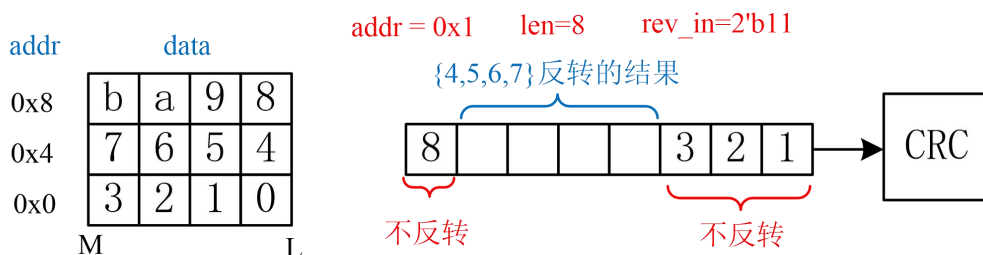
例如：输入数据为 0x1A2B3C4D

- 按字节执行位反转，数据变为 0x58D43CB2；//[31:0]->[0,1,2,3,.....,31]
- 按半字执行位反转，数据变为 0xD458B23C；//[31:0]->[16,15,.....,31,0,1....15]
- 按字执行位反转，数据变为 0xB23CD458。//[31:0]->[24,....,31,16,....,23,8,....,15,0,....,7]

当有效数据位宽不等于反转的位宽的倍数，就不会执行反转。例有效数据 data[7:0]，设定执行按半字/字反转，将不会执行反转操作。

DMA 取数时会根据当前地址取到的有效数据来判断是否可以执行反转。

异常情况，例：起始地址为 0x1，按字执行位反转，第一次取到的数据为{1,2,3}，仅有 24bit 有效数据，不进行反转，第二次取到的数据为{4,5,6,7}，32bit 有效数据，按字执行位反转。第三次取到的数据为{8}，仅有 8bit 有效数据，不进行反转。

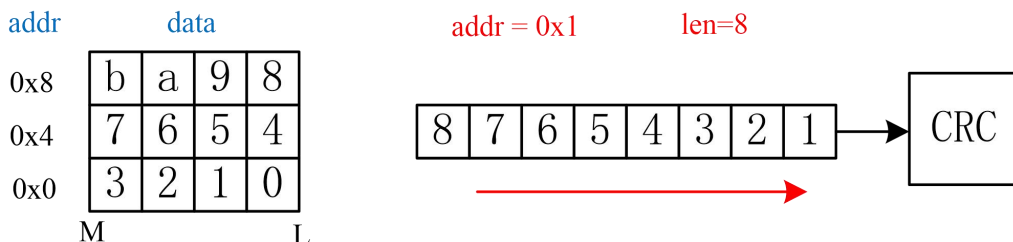


21.8 DMA 接口

CRC 模块通过 DMA 从 SRAM 里读取数据并进行校验，校验的结果仍然存放在寄存器中。

21.8.1 CRC 计算

数据计算顺序：从地址的最低 byte 开始计算。例：下图中每个方块表示 1byte，起始地址为 0x1，长度为 8。通过 DMA 取得数据 {1,2,3,4,5,6,7,8}。以该顺序依次进行 CRC 计算。即首先计算“1”的 CRC，然后计算“2”的 CRC，直到最后一个 byte。



21.8.2 Checksum 计算

校验和计算仅在 CHKSUM_DMA 使能时使用，可通过设置 CRC_CR.OPDW，来确定累加位宽为 8bit/16bit/32bit，计算结果保存为 32bit 于 CRC_DR 寄存器。支持设置计算初值、输出反转和异或、输入数据反

转，但当有效数据位宽不等于反转的位宽的倍数，就不会执行反转。

输出反转是将 32bit 结果进行反转。

设定累加位宽为 8bit，DMA 的长度和起始地址可以为任意值；

设定累加位宽为 16bit，DMA 长度可以为任意值，起始地址的低两位必须为 0，配置其它值亦会自动为 0。当剩余长度不满足 16bit 时，会自动在高位补 0；

设定累加位宽为 32bit，DMA 长度可以为任意值，起始地址的低两位必须为 0，配置其它值亦会自动为 0。当剩余长度不满足 32bit 时，会自动在高位补 0。

21.9 计算速度评估

21.9.1 CRC 计算

配置寄存器计算时，8bit 数据需要 1 周期，16bit 数据需要 2 周期，32bit 数据需要 3 周期。

使用 DMA 时，长度 $n \times \text{byte}$ ，最快需要 $4+n$ 个周期完成，实际运算周期还与配置的地址以及 SRAM 的竞争相关。

21.9.2 Checksum 计算

累加位宽为 8bit，DMA 长度为 $n \times \text{byte}$ ，最快需要 $4+n$ 个周期完成。

累加位宽为 16bit，DMA 长度为 $n \times \text{byte}$ ，最快需要 $2 + n/4 \times 3$ 个周期完成。

累加位宽为 32bit，DMA 长度为 $n \times \text{byte}$ ，最快需要 $2 + n/4 \times 3$ 个周期完成。

21.10 寄存器描述

寄存器列表

CRC 模块的基址

模块名	物理地址	映射地址
CRC	0x40074000	0x40074000

CRC 模块的寄存器偏移地址

寄存器名	地址偏移量	描述
CRC_DR	Offset+0x0	CRC 数据寄存器
CRC_STA	Offset+0x4	CRC 状态寄存器
CRC_CTRL	Offset+0x8	CRC 控制寄存器
CRC_INIT	Offset+0xC	CRC 初始值寄存器
CRC_POL	Offset+0x10	CRC 多项式寄存器
CRC_XOR	Offset+0x14	CRC 输出异或寄存器
CRC_DMA_CTL	Offset+0x18	CRC DMA 控制寄存器
CRC_DMA_BADR	Offset+0x1C	CRC DMA 起始地址寄存器
CRC_DMA_LEN	Offset+0x20	CRC DMA 数据长度寄存器
CRC_DMA_ADR	Offset+0x24	CRC DMA 当前地址寄存器
CRC_DMA_IE	Offset+0x28	CRC DMA 中断使能寄存器
CRC_DMA_FLG	Offset+0x2C	CRC DMA 中断标志寄存器

21.10.1 CRC_DR (0x00)

CRC 数据寄存器

偏移地址=0x0

比特位	名称	描述	读/写	复位值
31:0	DR	<p>CRC 计算:</p> <p>该寄存器用于作为数据输入寄存器,并且在运算结束后保存 CRC 计算结果。</p> <p>写入该寄存器时,作为数据输入寄存器,每次写入数据 CRC 计算一次,DMA 使能时写无效;</p> <p>读取该寄存器时,返回 CRC 计算的结果。</p> <p>作为输入时:</p> <p>若 OPDW==00,写入数据 DR[31:0]有效;</p> <p>若 OPDW==01,写入数据 DR[15:0]有效;</p> <p>若 OPDW==1x,写入数据 DR[7:0]有效。</p> <p>保存结果时:</p> <p>若为 7 位多项式结果保存在 DR[6:0];</p> <p>若为 8 位多项式结果保存在 DR[7:0];</p> <p>若为 16 位多项式结果保存在 DR[15:0];</p> <p>若为 32 位多项式结果保存在 DR[31:0]。</p> <p>注意: CRC 按 byte 的形式计算,首先计算 DR[7:0],然后计算 DR[15:8]直到最后一个有效 byte。</p> <p>校验和计算:</p> <p>该寄存器仅用作于保存校验和的结果。DMA 使能时写无效。</p>	R/W	0

21.10.2 CRC_STA (0x04)

CRC 状态寄存器

偏移地址=0x04

比特位	名称	描述	读/写	复位值
31:1	---	预留	R	0
0	DONE	<p>CRC 运算完成标志</p> <p>0: CRC 运算未完成</p> <p>1: CRC 运算完成</p> <p>在使用 DMA 时,当 DMA 全部数据完成运算后,才表示 CRC 运算完成</p>	RO	1

21.10.3 CRC_CTRL (0x08)

CRC 控制寄存器

偏移地址=0x08

比特位	名称	描述	读/写	复位值
31:9	---	预留	R	0
7:6	OPDW	<p>计算数据位宽选择 (Operation by data width)</p> <p>00: 32bit 数据计算</p>	R/W	00

		01: 16bit 数据计算 1x: 8bit 数据计算 不使能 DMA 时，作为写寄存器计算 CRC 的计算数据位宽选择。 CRC_DMA 使能时，硬件会根据 DMA 长度进行自适应计算，该配置无效。 CHKSUM_DMA 使能时，作为累加位宽的选择。 在 CRC 运算时配置，将会作为下次计算的配置。 在校验和运算时不可配，配置无效		
5:4	REV_IN	反转输入数据（Reverse input data） 00: 不反转 01: 按字节执行位反转 10: 按半字执行位反转 11: 按字执行位反转 例如：计算数据为 0x1A2B3C4D REV_IN==00，直接使用数据 0x1A2B3C4D 计算； REV_IN==01，将数据变为 0x58D43CB2，再计算； REV_IN==10，将数据变为 0xD458B23C，再计算； REV_IN==11，将数据变为 0xB23CD458，再计算。 当有效数据位宽不等于反转的位宽的倍数，就不会执行反转（详细见 29.7 描述）。 在运算时不可配置，配置无效。	R/W	00
3	REV_OUT	反转输出数据（Reverse output data） 0: 输出不反转 1: 输出按 CRC 结果位长反转 例如：CRC 结果为 0xAABB REV_OUT==1，则输出的结果为 0xDD55； REV_OUT==0，则输出的结果为 0xAABB。 校验和计算会按照 32bit 反转。 在运算时不可配置，配置无效。	R/W	0
2	XOR	输出数据按位异或使能 0: 输出数据不操作 1: 输出数据与 CRC_XOR 寄存器的值异或 在运算时不可配置，配置无效。	R/W	0
1:0	POLYSIZE	多项式大小 00: 32 位多项式 01: 16 位多项式 10: 8 位多项式 11: 7 位多项式 在 CRC 运算时不可配置，配置无效 在校验和计算时不使用该配置位。	R/W	00

21.10.4 CRC_INIT (0x0C)

CRC 初始值寄存器

偏移地址=0x0C

比特位	名称	描述	读/写	复位值
31:0	INIT	<p>CRC 初始值。</p> <p>若选择 32 位多项式大小，则 INIT[31:0]有效；</p> <p>若选择 16 位多项式大小，则 INIT[15:0]有效；</p> <p>若选择 8 位多项式大小，则 INIT[7:0]有效；</p> <p>若选择 7 位多项式大小，则 INIT[6:0]有效。</p> <p>在运算时不可配置，配置无效。</p> <p>CRC 计算：</p> <p>使用配置寄存器计算 CRC 时，进行 CRC 计算前，没有配置该寄存器，那么余数初始值会采用上一个数据计算的结果。</p> <p>使用 DMA 时，会自动采用该寄存器的值作为余数初始值，无须重新配置，若需要更改余数初始值，必须在计算前配置。</p> <p>校验和计算：</p> <p>作为累加的初始数值。</p>	R/W	0

21.10.5 CRC_POL (0x10)

CRC 多项式寄存器

偏移地址=0x10

比特位	名称	描述	读/写	复位值
31:0	POLY	<p>用于 CRC 计算的多项式，若多项式小于 32 位则使用最低有效位</p> <p>写入多项式的值忽略最高位的值。</p> <p>例如，CRC-CCITT 的多项式是 $x^{16}+x^{12}+x^5+1$，16 进制即是 0x11021，忽略最高位，写入的值为 0x1021</p> <p>若选择 32 位多项式大小，则 POLY [31:0]有效；</p> <p>若选择 16 位多项式大小，则 POLY [15:0]有效；</p> <p>若选择 8 位多项式大小，则 POLY [7:0]有效；</p> <p>若选择 7 位多项式大小，则 POLY [6:0]有效。</p> <p>在 CRC 运算时不可配置，配置无效</p>	R/W	0

21.10.6 CRC_XOR (0x14)

CRC 输出异或寄存器

偏移地址=0x14

比特位	名称	描述	读/写	复位值
31:0	XOR	<p>CRC 运算结果异或寄存器 (exclusive OR) 当 CRC_CR.XOR 为 1 时，CRC/校验和结果输出前将异或此寄存器的数据。</p> <p>若选择 32 位多项式大小，则 XOR [31:0]有效；</p> <p>若选择 16 位多项式大小，则 XOR [15:0]有效；</p> <p>若选择 8 位多项式大小，则 XOR [7:0]有效；</p> <p>若选择 7 位多项式大小，则 XOR [6:0]有效。</p> <p>在运算时不可配置，配置无效</p>	R/W	0

21.10.7 CRC_DMA_CTL (0x18)

CRC DMA 控制寄存器

偏移地址: 0x18

比特位	名称	描述	读/写	复位值
31:2	Reserved	保留。	R	0
2	CHANNEL	数据通道选择 0: SRAM 通道 1: ROM 通道	R/W	0
1:0	DMA_EN	DMA 使能。 00: 不使能 DMA; 01: 使能 CRC_DMA; 10: 使能 CHKSUM_DMA; 11: 保留, 不使能 DMA; 每次使能进行一次 CRC/校验和计算, 计算结束自动关闭使能, 下次计算需要重新配置使能 若在校验和计算的过程中, 配置校验和计算, 最终的结果会错乱。 若在校验和计算的过程中, 配置 CRC 计算, 最终的结果会错乱。 若在计算过程中需要更改计算方式, 必须先关闭 DMA, 再重新启动。 注意: 不允许配置 DMA_LEN=0, DMA_EN=1, 否则 DMA_EN 会一直为 1, 不会自动清除, 并且 DONE 标志一直为 0。 如果配置 DMA_LEN=0, DMA_EN=1, 在进行下次计算之前, 需要将 DMA_EN 写 0。	R/W	0

21.10.8 CRC_DMA_BADR (0x1C)

CRC DMA 起始地址寄存器

偏移地址: 0x1C

比特位	名称	描述	读/写	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_BADR	DMA 起始地址 (Byte 地址) 在校验和计算使用 16/32bit 累加时, 低两位必须为 0, 低两位配置了也无效。 其他情况可任意配置。 在使用 ROM 通道时, 地址最大是 14bit	R/W	0

21.10.9 CRC_DMA_LEN (0x20)

CRC DMA 长度寄存器

偏移地址: 0x20;

比特位	名称	描述	读/写	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_LEN	DMA 长度 (Byte 地址) = (n) Byte 起始地址+数据长度不能超出 SRAM 的地址范围 在使用 ROM 通道时, 长度最大是 14bit	R/W	0

21.10.10 CRC_DMA_ADR (0x24)

CRC 当前 DMA 地址寄存器

偏移地址: 0x24;

比特位	名称	描述	读/写	复位值
31:17	Reserved	保留。	R	0
16:0	DMA_ADR	当前 DMA 地址 (Byte 地址)	RO	0

21.10.11 CRC_DMA_IE (0x28)

CRC DMA 中断使能寄存器

偏移地址: 0x28

比特位	名称	描述	读/写	复位值
31:1	Reserved	保留。	R	0
0	CRC_DONE_IE	CRC 运算完成中断使能 =0, 不使能 =1, 使能	R/W	0

21.10.12 CRC_DMA_FLG (0x2C)

CRC DMA 中断标志寄存器

偏移地址: 0x2C; 默认值: 0x0

比特位	名称	描述	读/写	复位值
31:1	Reserved	保留。	R	0
0	CRC_DONE	CRC 运算完成中断标志, 写“1”清该位 当 DMA 所有数据计算完成后中断标志置起。 软件关闭 DMA, 该标志不会置起	R/WC	0

21.11 CRC 软件操作流程

21.11.1 配置 DR 寄存器进行计算

21.11.1.1 计算单次数据

```
CRC_INIT    = 0xxxxx; //新的数据计算必须重新配置 CRC_INIT 寄存器
CRC_CR      = 0xxxxx; //配置相关控制 OPDW, REV_IN, REV_OUT, XOR, POLYSIZE
CRC_POL     = 0xxxxx; //配置多项式
CRC_DR      = 0xxxxx; //配置计算数据——必须最后才配置数据
//等待计算结束
//8bit 数据计算不需要等待
// NOP; //16bit 数据计算等待 1 个周期
// NOP; NOP; NOP; //32bit 数据计算等待 3 个周期;
crc_result  = CRC_DR[x:0];          //读取 CRC 计算结果
//根据配置的 POLYSIZE 读取结果
//如配置 CRC7,   crc_result  = CRC_DR[6:0];
//CRC8,          crc_result  = CRC_DR[7:0];
//CRC16,         crc_result  = CRC_DR[15:0];
//CRC32,         crc_result  = CRC_DR[31:0];

//使用标志位
CRC_INIT    = 0xxxxx; //新的数据计算必须重新配置 CRC_INIT 寄存器
CRC_CR      = 0xxxxx; //配置相关控制 OPDW, REV_IN, REV_OUT, XOR, POLYSIZE
CRC_POL     = 0xxxxx; //配置多项式
CRC_DR      = 0xxxxx; //配置计算数据——必须最后才配置数据
while(!(CRC_STA & 0x1));           //等待标志位
crc_result  = CRC_DR[x:0];          //读取 CRC 计算结果
//根据配置的 POLYSIZE 读取结果
//如配置 CRC7,   crc_result  = CRC_DR[6:0];
//CRC8,          crc_result  = CRC_DR[7:0];
//CRC16,         crc_result  = CRC_DR[15:0];
//CRC32,         crc_result  = CRC_DR[31:0];
```

21.11.1.2 计算多次数据

以计算 6byte 数据为例:

```
CRC_INIT    = 0xxxxx; //新的数据计算必须重新配置 CRC_INIT 寄存器
CRC_CR      = 0xxxxx; //配置相关控制 OPDW, REV_IN, REV_OUT, XOR, POLYSIZE
//OPWD = 2'b00 ;先计算 4byte 数据
CRC_POL     = 0xxxxx; //配置多项式
CRC_DR      = 0xxxxx; //配置计算数据, 写入高位 4byte 数据
CRC_CR      = 0xxxxx; //配置相关控制 OPDW, REV_IN, REV_OUT, XOR, POLYSIZE
//OPWD = 2'b01 ;再计算 2byte 数据
CRC_DR      = 0xxxxx; //配置计算数据, 写入最后 2byte 数据
NOP();       //等待计算结束
crc_result  = CRC_DR[x:0];          //读取 CRC 计算结果
```

//根据配置的 POLYSIZE 读取结果

```
//如配置 CRC7,   crc_result   = CRC_DR[6:0];  
//CRC8,          crc_result   = CRC_DR[7:0];  
//CRC16,         crc_result   = CRC_DR[15:0];  
//CRC32,         crc_result   = CRC_DR[31:0];
```

以计算 8byte 数据为例:

```
CRC_INIT   = 0xxxxx; //新的数据计算必须重新配置 CRC_INIT 寄存器  
CRC_CR     = 0xxxxx; //配置相关控制 OPDW, REV_IN, REV_OUT, XOR, POLYSIZE  
                //OPWD = 2'b00 ;计算 4byte 数据  
CRC_POL    = 0xxxxx; //配置多项式  
CRC_DR     = 0xxxxx; //配置计算数据, 写入高位 4byte 数据  
NOP();  
CRC_DR     = 0xxxxx; //配置计算数据, 写入低位 4byte 数据  
NOP();  
NOP();  
NOP();  
crc_result = CRC_DR[x:0]; //读取 CRC 计算结果  
//根据配置的 POLYSIZE 读取结果  
//如配置 CRC7,   crc_result   = CRC_DR[6:0];  
//CRC8,          crc_result   = CRC_DR[7:0];  
//CRC16,         crc_result   = CRC_DR[15:0];  
//CRC32,         crc_result   = CRC_DR[31:0];
```

21.11.2 使用 DMA 进行计算

21.11.2.1 计算 DMA 数据

```
CRC_INIT   = 0xxxxx; //配置初值, 若初始值不变可不配  
CRC_CR     = 0xxxxx; //配置相关控制 REV_IN, REV_OUT, XOR, POLYSIZE  
CRC_POL    = 0xxxxx; //配置多项式  
DMA_BADR   = 0xxxxx; //配置 DAM 起始地址  
DMA_LEN    = 0xxxxx; //配置 DMA 数据长度  
DMA_CTL    = 0x1;     //使能 DMA  
while(!(CRC_STA & 0x1)); //等待标志位  
crc_result = CRC_DR [x:0]; //读取 CRC 计算结果  
//根据配置的 POLYSIZE 读取结果  
//如配置 CRC7,   crc_result   = CRC_DR[6:0];  
//CRC8,          crc_result   = CRC_DR[7:0];  
//CRC16,         crc_result   = CRC_DR[15:0];  
//CRC32,         crc_result   = CRC_DR[31:0];  
  
//使用中断标志  
CRC_INIT   = 0xxxxx; //配置初值, 若初始值不变可不配
```

```

CRC_CR    = 0xxxxx;    //配置相关控制 REV_IN, REV_OUT, XOR, POLYSIZE
CRC_POL   = 0xxxxx;    //配置多项式
DMA_IE    = 0x1;       //中断使能
DMA_BADR  = 0xxxxx;    //配置 DAM 起始地址
DMA_LEN   = 0xxxxx;    //配置 DMA 数据长度
DMA_CTL   = 0x1;       //使能 DMA
while(1);    //等待中断
----进入中断程序
crc_result = CRC_DR[x:0]; //读取 CRC 计算结果
//根据配置的 POLYSIZE 读取结果
//如配置 CRC7,   crc_result = CRC_DR[6:0];
//CRC8,         crc_result = CRC_DR[7:0];
//CRC16,        crc_result = CRC_DR[15:0];
//CRC32,        crc_result = CRC_DR[31:0];
DMA_FLG    = 0x1;       //写 1 清除中断, 退出中断
----退出中断程序

```

21.11.2.2 多段 DMA 数据计算

```

CRC_INIT   = 0xxxxx;    //配置初值, 若初始值不变可不配
CRC_CR     = 0xxxxx;    //配置相关控制 REV_IN, REV_OUT, XOR, POLYSIZE
CRC_POL    = 0xxxxx;    //配置多项式
DMA_BADR   = 0xxxxx;    //配置 DAM 起始地址
DMA_LEN    = 0xxxxx;    //配置 DMA 数据长度
DMA_CTL    = 0x1;       //使能 DMA
DMA_BADR   = 0xxxxx;    //配置下一段的 DAM 起始地址
DMA_LEN    = 0xxxxx;    //配置下一段的 DMA 数据长度
while(!(CRC_STA & 0x1)); //等待标志位
crc_result = CRC_DR [x:0]; //读取 CRC 计算结果
CRC_INIT   = crc_result; //配置初值
DMA_CTL    = 0x1;       //使能 DMA
while(!(CRC_STA & 0x1)); //等待标志位
crc_result = CRC_DR [x:0]; //读取 CRC 计算结果
//根据配置的 POLYSIZE 读取结果
//如配置 CRC7,   crc_result = CRC_DR[6:0];
//CRC8,         crc_result = CRC_DR[7:0];
//CRC16,        crc_result = CRC_DR[15:0];
//CRC32,        crc_result = CRC_DR[31:0];

```

21.11.2.3 计算过程中关闭 DMA

关闭 DMA 后, CRC 运算会立即停止计算。

21.12 Checksum 软件操作流程

21.12.1 使用 DMA 进行计算

```

CRC_INIT   = 0xxxxx;    //配置初值, 若初始值不变可不配

```

```

CRC_CR      = 0xxxxx;           //配置相关控制 REV_IN, REV_OUT, XOR
DMA_BADR    = 0xxxxx;           //配置 DMA 起始地址
DMA_LEN     = 0xxxxx;           //配置 DMA 数据长度
DMA_CTL     = 0x2;              //使能 DMA
while(!(CRC_STA & 0x1));         //等待标志位
chksum_result = CRC_DR[31:0];    //读取校验和计算结果

//使用中断标志
DMA_FLG     = 0x1;              //写 1 清除中断
CRC_INIT    = 0xxxxx;           //配置初值, 若初始值不变可不配
CRC_CR      = 0xxxxx;           //配置相关控制 REV_IN, REV_OUT, XOR
DMA_BADR    = 0xxxxx;           //配置 DMA 起始地址
DMA_LEN     = 0xxxxx;           //配置 DMA 数据长度
DMA_CTL     = 0x2;              //使能 DMA
while(1);    //等待中断
----进入中断程序
chksum_result = CRC_DR[31:0];    //读取校验和计算结果
DMA_FLG     = 0x1;              //写 1 清除中断, 退出中断
----退出中断程序

```

21.12.2 计算过程中关闭 DMA

关闭 DMA 后, 校验和运算会立即停止计算。

22 脉冲转发 IOCNT (新增)

22.1 概述

脉冲转发功能, 又称 IOCNT 功能, 即具有将 GPIO 上输入的脉冲或者内部所有计量脉冲, 经过分频, 电平取反等操作后再从 GPIO 口转发出来的功能。同时具备记录输入脉冲个数的功能。

22.2 功能特性

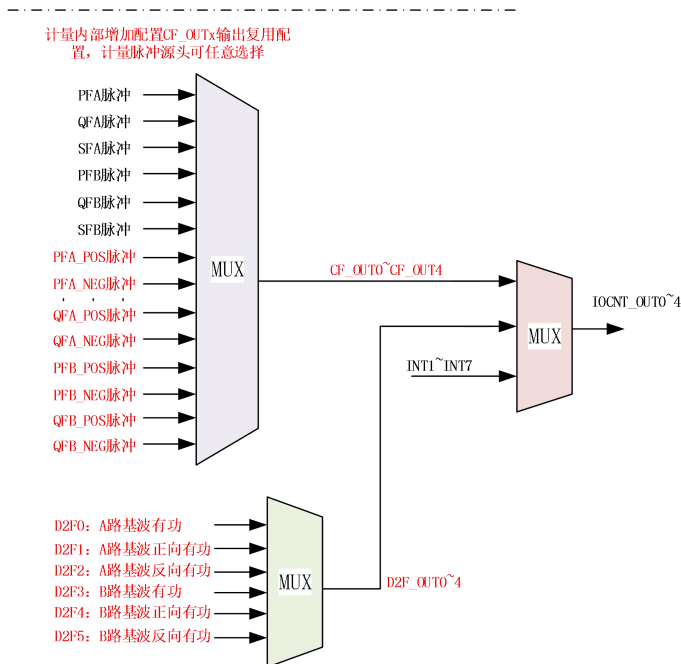
- 支持 5 个 IOCNT
- 支持 1~2048 分频
- 支持脉冲个数记录, 计数边沿可配, 计数器读后清零和不清零可配
- 支持输入脉冲电平反向转发
- 支持输入通道自由可配

22.3 复用关系

所有的 IOCNT_OUT 支持内部 CF_OUT0~4、D2F_OUT0~4 以及 INT1~INT7 可选。

通道选择 IOCNTx_SEL[5:0]	输入	复用 IO		输出	复用 IO		
0x0	CF_OUT0	内部	-	IOCNT_OUT0	P50	P56	-
0x1	CF_OUT1	内部	-	IOCNT_OUT1	P51	-	-
0x2	CF_OUT2	内部	-	IOCNT_OUT2	P57	P13	P10

0x3	CF_OUT3	内部	-	IOCNT_OUT3	P34	P33	P11
0x4	CF_OUT4	内部	-	IOCNT_OUT4	P35	-	P12
0x5	D2F_OUT0	内部	-	-	-	-	-
0x6	D2F_OUT1	内部	-	-	-	-	-
0x7	D2F_OUT2	内部	-	-	-	-	-
0x8	D2F_OUT3	内部	-	-	-	-	-
0x9	D2F_OUT4	内部	-	-	-	-	-
0xA	INT1	P31	P40	-	-	-	-
0xB	INT2	P32	P56	-	-	-	-
0xC	INT3	P33	P41	-	-	-	-
0xD	INT4	P34	P42	-	-	-	-
0xE	INT5	P35	P43	-	-	-	-
0xF	INT6	-	P14	-	-	-	-
0x10	INT7	-	P15	-	-	-	-
0x11	UART0 PF	内部	-	-	-	-	-
0x12	UART1 PF	内部	-	-	-	-	-
0x13	UART2 PF	内部	-	-	-	-	-
0x14	UART3 PF	内部	-	-	-	-	-
0x15	UART4 PF	内部	-	-	-	-	-
0x16	UART5 PF	内部	-	-	-	-	-
0x17	UART0 QF	内部	-	-	-	-	-
0x18	UART1 QF	内部	-	-	-	-	-
0x19	UART2 QF	内部	-	-	-	-	-
0x1A	UART3 QF	内部	-	-	-	-	-
0x1B	UART4 QF	内部	-	-	-	-	-
0x1C	UART5 QF	内部	-	-	-	-	-
0x1D	UART0 FPF	内部	-	-	-	-	-
0x1E	UART1 FPF	内部	-	-	-	-	-
0x1F	UART2 FPF	内部	-	-	-	-	-
0x20	UART3 FPF	内部	-	-	-	-	-
0x21	UART4 FPF	内部	-	-	-	-	-
0x22	UART5 FPF	内部	-	-	-	-	-



22.4 功能框图

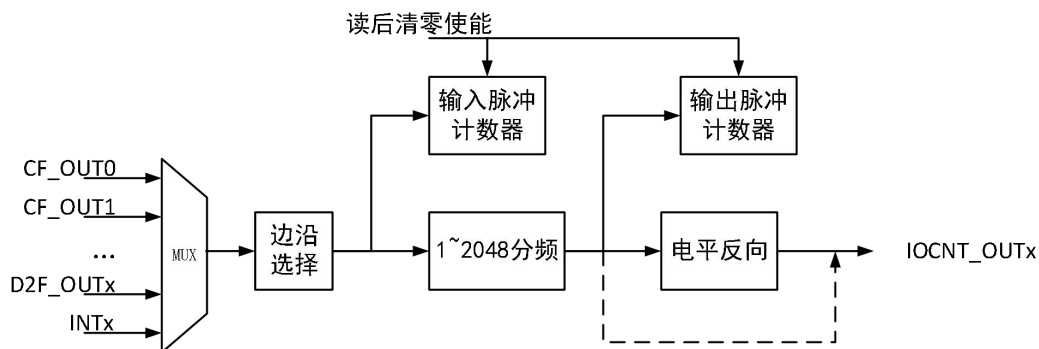


图 1 脉冲转发功能框图

22.5 波形分析

22.5.1 输入信号类型

输入信号有周期均匀分布和周期非均匀分布情况，周期均匀情况又分为等占空比和不等占空比情况，如下所示，

- 脉冲周期均匀分布



图 2 等 duty 信号



图 3 非等 duty 信号

- 脉冲周期非均匀分布



图 4 非均匀周期信号

22.5.2 脉冲计数

脉冲计数为输入信号 `io_in` 的边沿计数，上升沿计数或下降沿计数可通过控制寄存器配置信号定义：

- `io_in`: PAD 输入；
- `io_out`: 上图中分频器与反相器之间的信号；

22.5.2.1 上升沿计数

当输入信号 `io_in` 输入默认为低电平，建议使用上升沿计数。该模式下 `io_out` 默认输出低电平。

启动时间点：

- 当 `io_in` 处于低电平时启动，从第一个上升沿开始计数，如图 5 所示
- 当 `io_in` 处于高电平时启动，计数值加 1，如图 6 所示

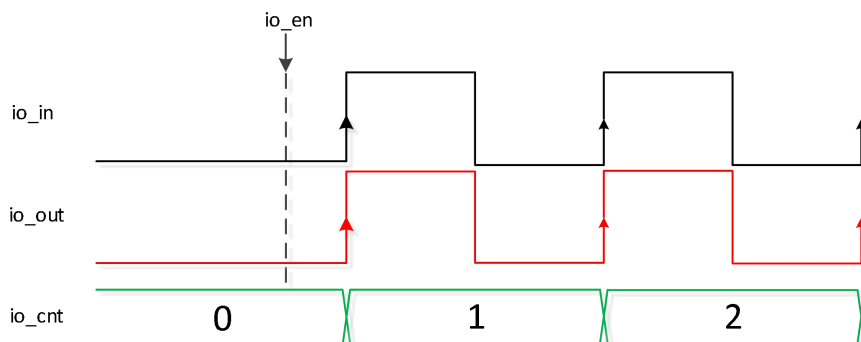


图 5 低电平启动上升沿计数（脉冲不分频）

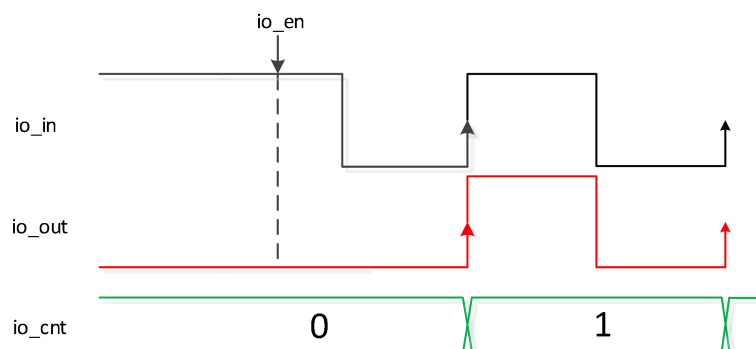


图 6 高电平启动上升沿计数（脉冲不分频）

22.5.2.2 下降沿计数

当输入信号 `io_in` 输入默认为高电平，建议使用下降沿计数。该模式下 `io_out` 默认输出高电平。

当 `io_in` 状态处于高电平启动时，第一个下降沿有效开始计数，如下图 7 所示。

当 `io_in` 状态处于低电平启动时，计数值加 1，如下图 8 所示。

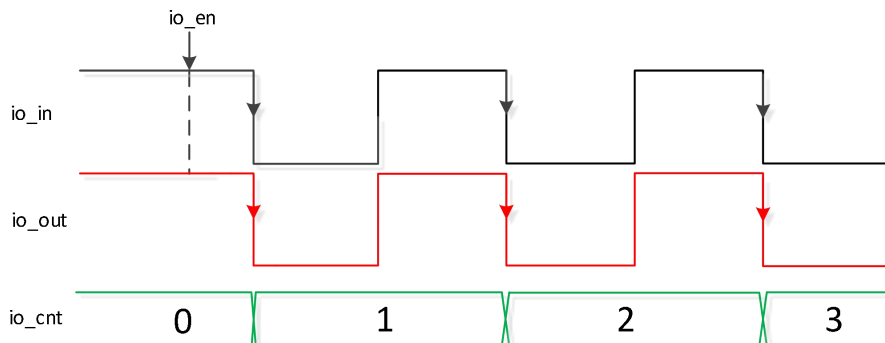


图 7 高电平启动下降沿计数（脉冲不分频）

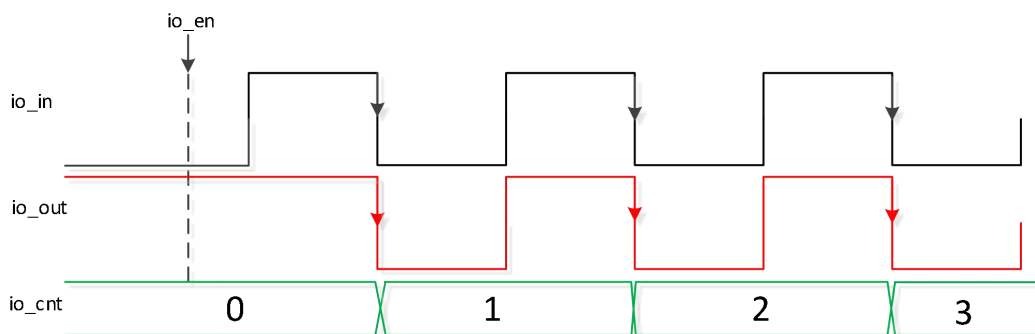


图 8 低电平启动下降沿计数（脉冲不分频）

22.5.2.3 计数清零和计数溢出

每次读 `io_cnt` 寄存器时，会清除历史计数值。

- 当读操作在输入信号边沿变化前或变化后到来，当次边沿累计到下一次 `io_cnt`，如图 9，图 10 所示。

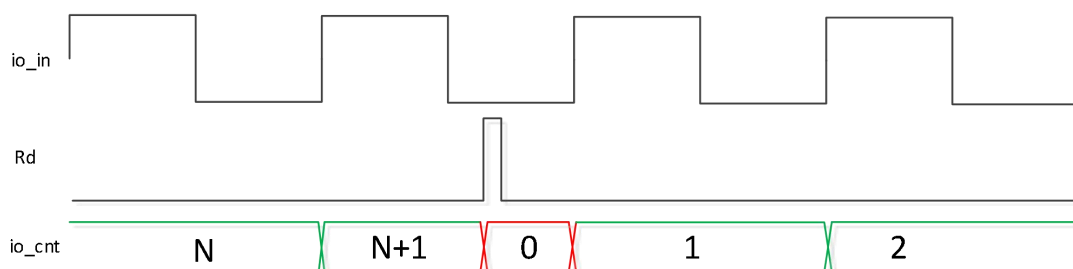


图 9 清零操作在 `io` 边沿前（上升沿计数）

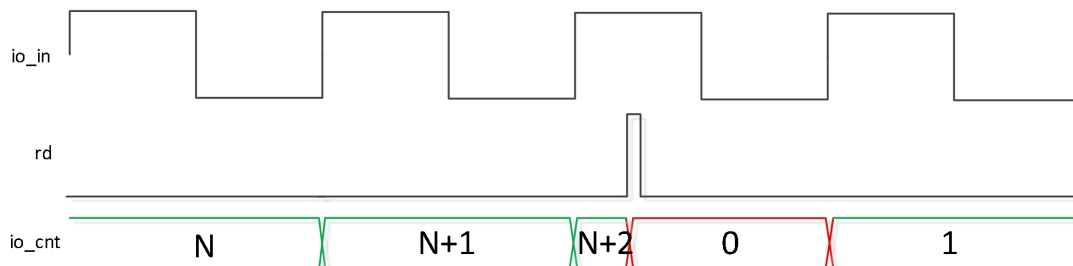


图 10 清零操作在 io 边沿后（上升沿计数）

- 读操作在输入信号边沿同时发生，当次读到的为上一个计数值，当次边沿累计到下一次 io_cnt，如图 11 所示。

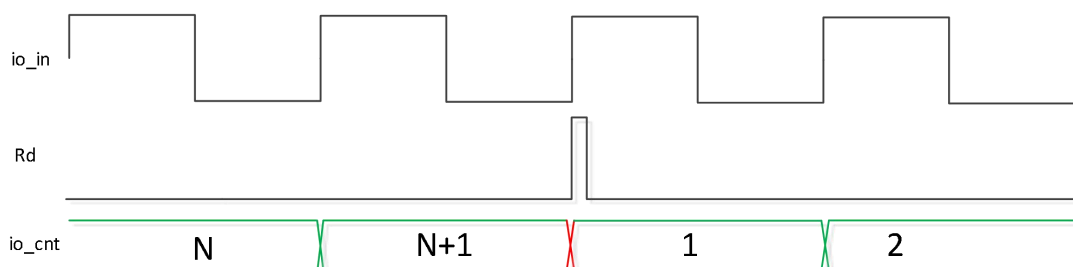


图 11 清零操作同 io 边沿（上升沿计数）

- 计数溢出设计，当 io_cnt = 0xfffff，再来一个脉冲时，计数溢出，计数 io_cnt = 0，从头开始计数，后续脉冲来 io_cnt = io_cnt + 1，如图 12 所示。

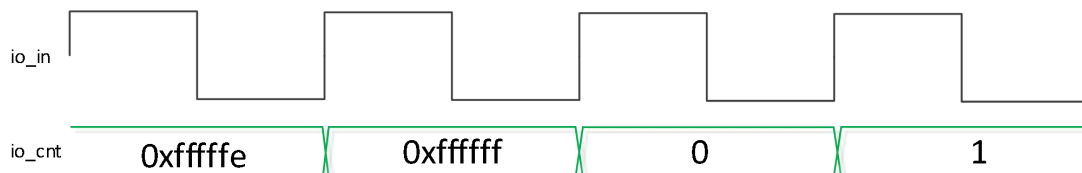


图 12 io_cnt 溢出（上升沿计数）

当客户检测到数值溢出时，当前值 a，实际当次计数值 = $a + 0xfffff + 1$ 。

- 溢出与读同时发生，此时读到 0xfffff，同时计数值变为 1。当溢出和读清零同时发生时，io_cnt = 1。如图 13 所示。

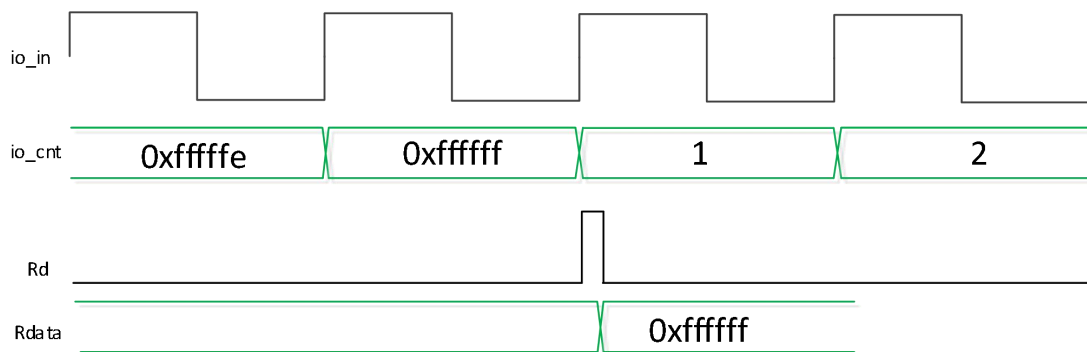


图 13 溢出与读同时来（上升沿计数）

22.5.3 脉冲分频

22.5.3.1 不分频（脉冲跟随）

- 不分频时，io_out 跟随 io_in 变化，不做处理
- 当 io_in 高电平超过 80ms，io_out 也不做任何特殊处理，io_out 跟随 io_in。

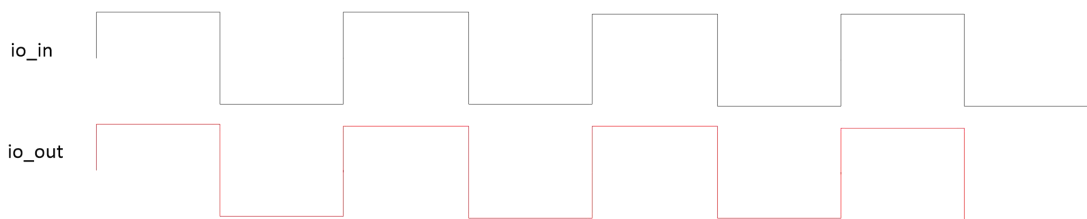


图 14 等 duty 信号

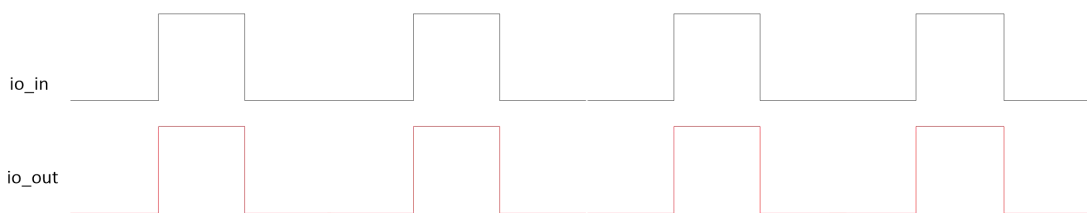


图 15 非等 duty 信号

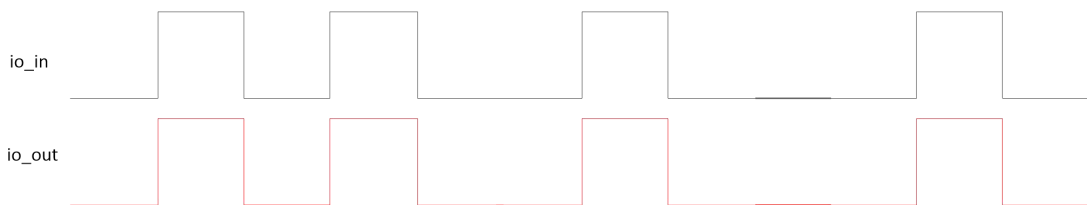


图 16 非均匀周期信号

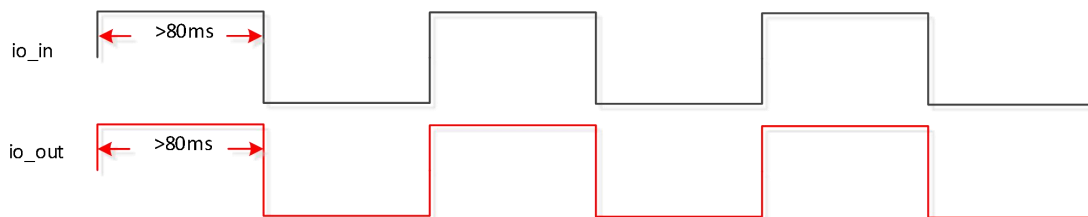


图 17 输入信号脉宽大于 80ms

22.5.3.2 偶数分频 (io_out 高电平<80ms)

- io_out 是 io_in 的分频，分频比 $N=n+1$ ，n 为寄存器分频系数配置值
- io_out 与 io_in 的边沿对齐，由边沿配置模式决定
- io_in 为均匀周期分布时，偶数分频后 io_out 保持等 duty 波形，如图 18，图 19，图 20 所示
- io_in 为非均匀周期分布时，偶数分频后 io_out 为非等 duty 波形，如图 21 所示

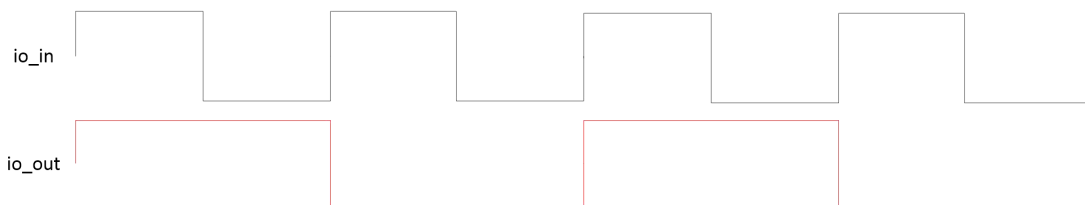


图 18 (n+1) 分频

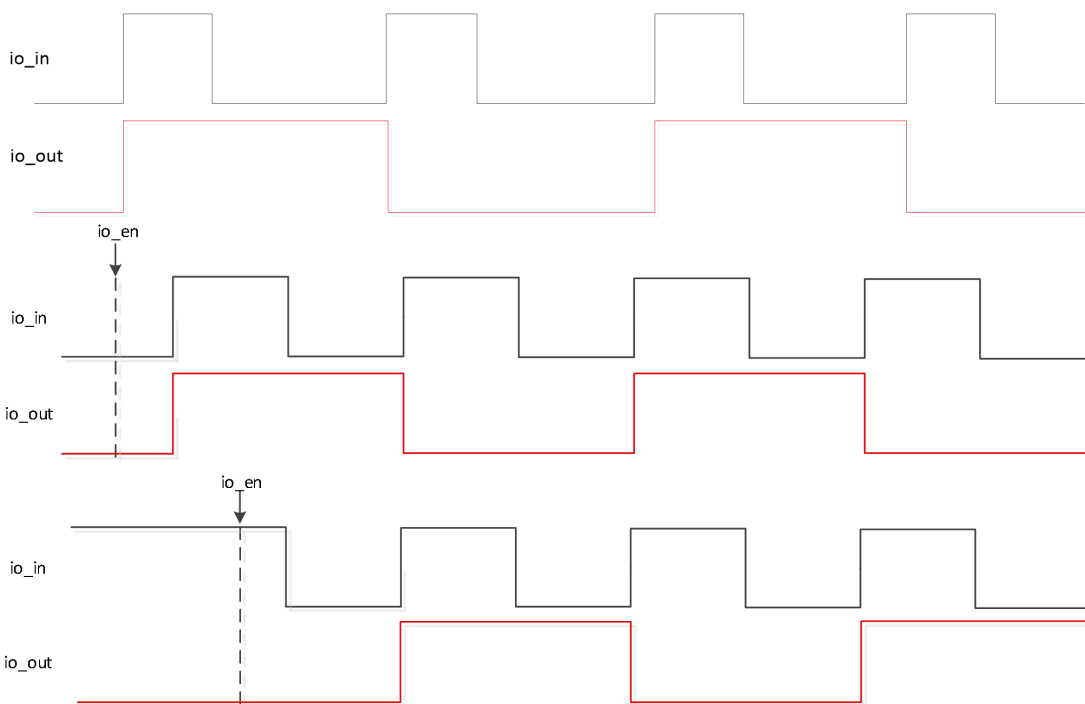


图 19 上升沿对齐

异常点启动，第一个输出脉冲非等 duty。

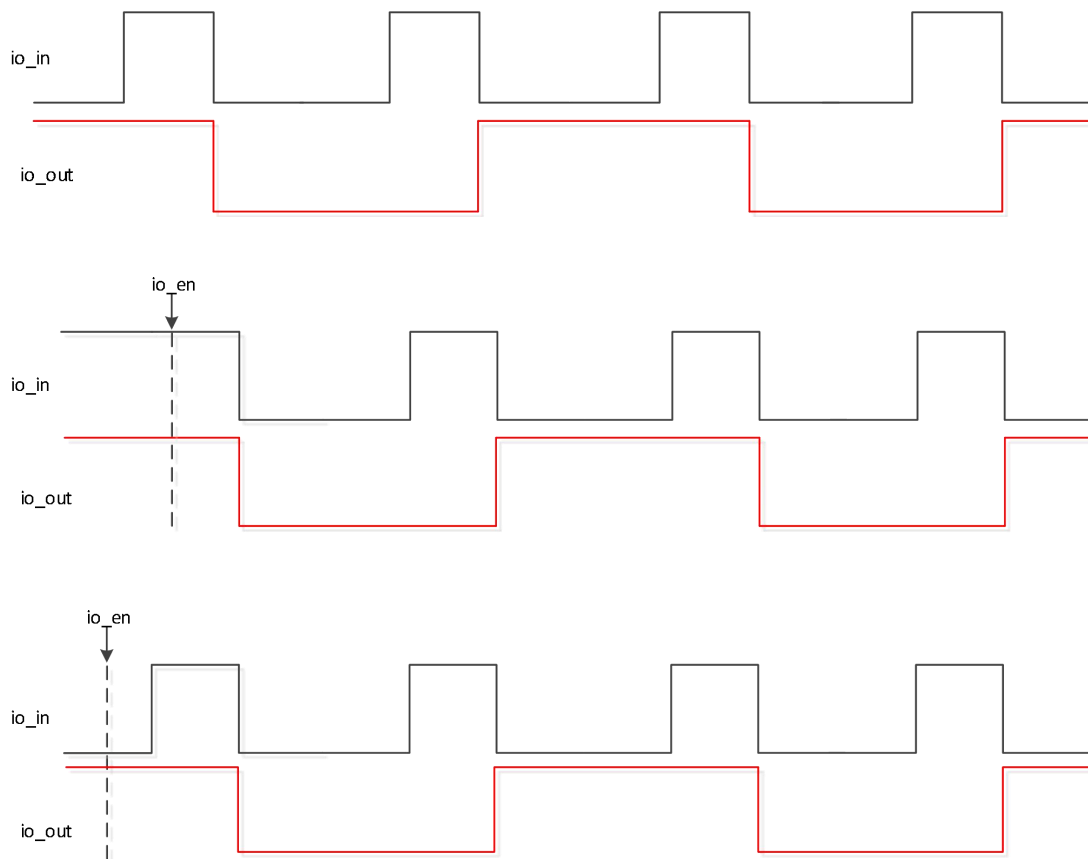


图 20 下降沿对齐

异常点启动，第一个输出脉冲非等 duty。



图 21 非均匀周期分频

22.5.3.3 奇数分频 (io_out 高电平<80ms)

- io_out 是 io_in 的分频，分频比由寄存器 n+1 配置。
- io_out 与 io_in 的边沿对齐，根据边沿计数模式决定。
- io_in 为均匀周期分布时，奇数分频后 io_out 为非等 duty 波形，如图 22，图 23 所示。
- io_in 为非均匀周期分布时，奇数分频后 io_out 为非等 duty 波形，如图 24 所示。

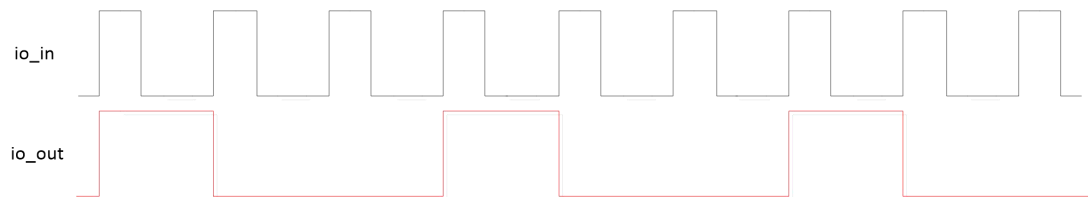


图 22 上升沿对齐

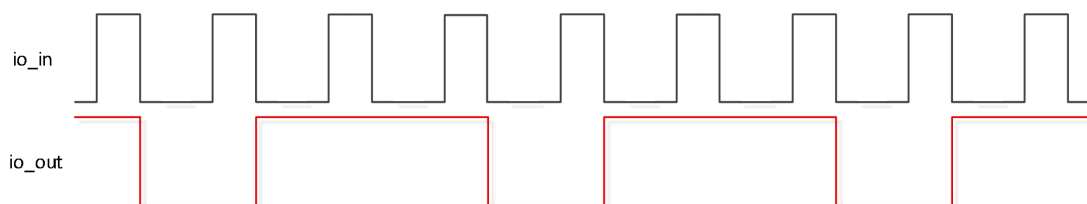


图 23 下降沿对齐

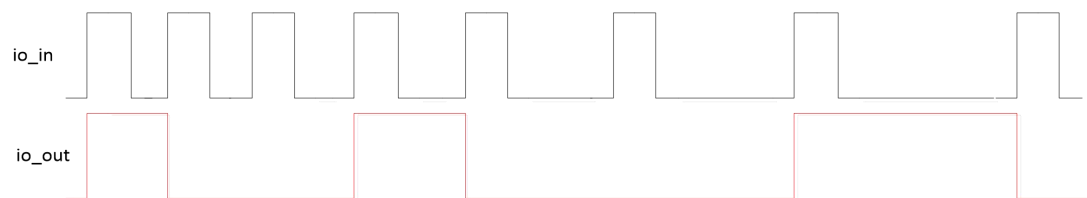


图 24 非均匀周期

22.5.3.4 分频后有效电平（高或低）脉宽 $\geq 80\text{ms}$

当配置为上升沿计数时，有效电平为高电平；

当配置为下降沿计数时，有效电平为低电平；

- 上升沿计数，分频后当高电平脉宽 $\geq 80\text{ms}$ ，高电平则被拉低，保持高电平为 80ms ，低电平保持至分频周期结束。
- 反之，下降沿计数情况同样处理。

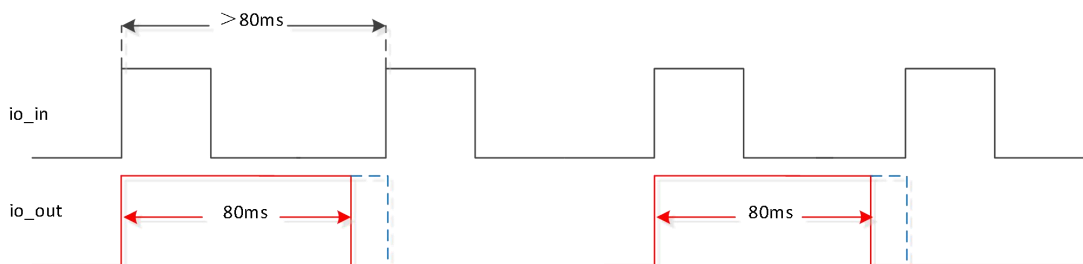


图 25 非等 duty（二分频）

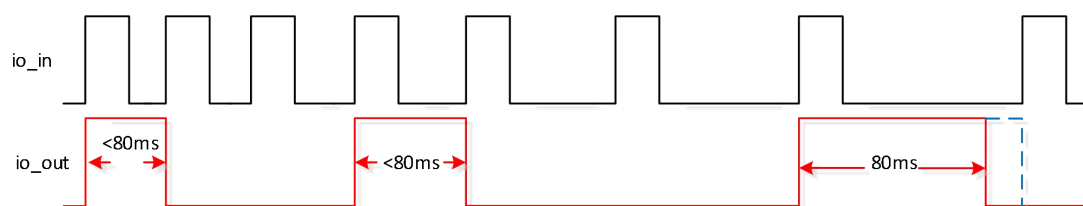
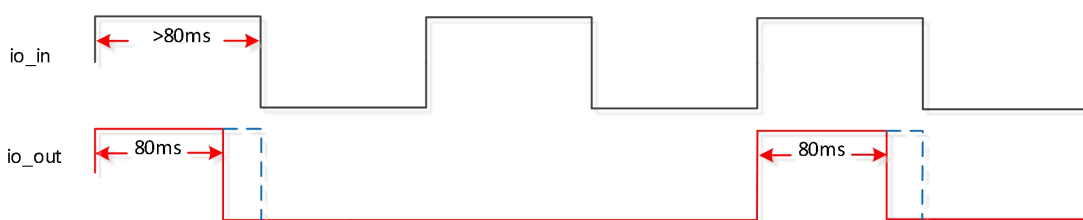


图 26 非均匀周期（三分频）


图 27 输入有效电平大于 80ms （二分频）

22.5.4 脉冲电平反向输出

- 支持将 io_out 信号电平翻转再输出，其中 io_out 为分频后的输出。

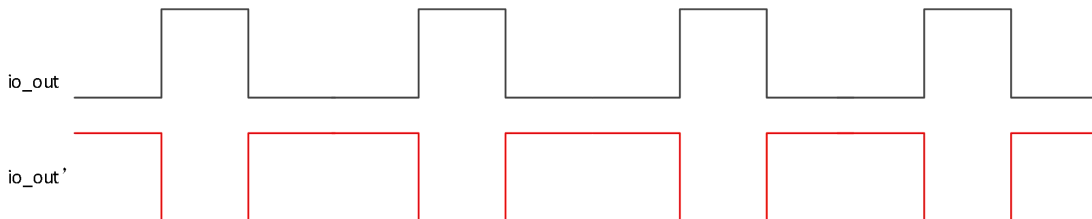


图 28 脉冲电平反向输出

22.6 寄存器描述

22.6.1 寄存器列表

IOCNT 模块的寄存器基址

模块名	物理地址	映射地址
IOCNT	0x4006C000	0x4006C000

IOCNT 接口的寄存器偏移地址

寄存器名	地址偏移量	描述
IOCNT_CFG0	offset+0x0	脉冲转发配置寄存器 0
IOCNT_CFG1	Offset+0x4	脉冲转发配置寄存器 1
IOCNT_CFG2	Offset+0x8	脉冲转发配置寄存器 2
IOCNT_CFG3	Offset+0xC	脉冲转发配置寄存器 3
IOCNT_CFG4	Offset+0x10	脉冲转发配置寄存器 4
IOCNT_OUT0	Offset+0x20	脉冲转发输出脉冲计数器 0
IOCNT_OUT1	Offset+0x24	脉冲转发输出脉冲计数器 1
IOCNT_OUT2	Offset+0x28	脉冲转发输出脉冲计数器 2
IOCNT_OUT3	Offset+0x2C	脉冲转发输出脉冲计数器 3
IOCNT_OUT4	Offset+0x30	脉冲转发输出脉冲计数器 4
IOCNT_CHNL	Offset+0x40	脉冲转发输入通道选择寄存器
IOCNT_CTL	Offset+0x48	IOCN 脉冲转发控制寄存器

应用注意：

- 不支持再模块使能情况下，修改寄存器配置。
- 支持在线系统时钟频率调整，模式切换后会清掉 80ms 计数器并重新开始计数。
- 关闭模块使能，会复位模块，所有计数值清零。

22.6.2 IOCNT_CFG0~4 (0x00~0x10) (新增)

脉冲转发配置寄存器 0~4

比特位	名称	描述	读/写标志	复位值
31	INT	脉冲反向输出使能： =0：输出脉冲保持与输入脉冲电平方向一致； =1：输出脉冲反向后输出。	R/W	00

30:20	CFG	脉冲分频系数配置： 分频系数 = (CFG+1)， 其中 CFG=0 时不分频，最高支持 2048 分频。 备注：分频后的脉冲要求高电平保持 80ms，如果快于 80ms， 则保持等 duty 输出；建议 CFG 配置为奇数，这样易于实现等 duty。 注：使能 UART 进行脉冲转发时，分频系数必须配置大于等于 2 分频，必须关闭滤波功能	R/W	00
19:0	CNT	脉冲计数器，根据配置的触发沿记录输入脉冲分频前的脉冲个 数。 CNT 寄存器读后清零，设计参考电能寄存器。	R	00

22.6.3 IOCNT_OUT0~4 (0x20~0x30) (新增)

脉冲转发输出脉冲计数器 0

比特位	名称	描述	读/写标志	复位值
31:20	---	保留	R	00
19:0	CNT	脉冲计数器，根据配置的触发沿记录脉冲分频后的输出脉冲个 数。 CNT 寄存器读后清零，设计参考电能寄存器。	R	00

22.6.4 IOCNT_CHNL (0x40) (新增)

脉冲转发输入通道选择寄存器 0

比特位	名称	描述	读/写标志	复位值
31:30	---	保留	R	0
29:24	IOCNT4_SEL	同 IOCNT0_SEL	R/W	0
23:18	IOCNT3_SEL	同 IOCNT0_SEL	R/W	0
17:12	IOCNT2_SEL	同 IOCNT0_SEL	R/W	0
11:6	IOCNT1_SEL	同 IOCNT0_SEL	R/W	0
5:0	IOCNT0_SEL	=0x0~0x4：内部输入 CF_OUT0~CF_OUT4 =0x5~0x9：内部输入 D2F_OUT0~D2F_OUT4 =0xA~0x10：外部中断输入 INT1~INT7 =0x11~0x16：内部输入 UART0_PF~UART5_PF =0x17~0x1C：内部输入 UART0_QF~UART5_QF =0x1D~0x22：内部输入 UART0_FPF~UART5_FPF =其他：保留 注：使能 UART 进行脉冲转发时，分频系数必须配置大于等 于 2 分频，必须关闭滤波功能	R/W	0

22.6.5 IOCNT_CTL (0x48) (新增)

IOCNT 脉冲转发控制寄存器

比特位	名称	描述	读/写标志	复位值
31:25	FLT_CFG	IOCNT4~0 滤波周期数配置	R/W	0x1f

		滤波周期=FLT_CFG +1 时钟周期，最大可配置 128 周期 只有当宽度小于滤波周期的时候，才滤掉。大于或等于滤波周期，不会滤掉。如 FLT_CFG=3，则滤波周期是为 4，只有波形宽度小于 3，才会被过滤掉。		
24:22	--	保留	R	0
21	FLT_BYPASS4	IOCNT4 滤波功能配置 =0，滤波功能使能，按照 flt_cfg 配置进行滤波 =1，不使能滤波功能	R/W	0
20	FLT_BYPASS3	IOCNT3 滤波功能配置 =0，滤波功能使能，按照 flt_cfg 配置进行滤波 =1，不使能滤波功能	R/W	0
19	FLT_BYPASS2	IOCNT2 滤波功能配置 =0，滤波功能使能，按照 flt_cfg 配置进行滤波 =1，不使能滤波功能	R/W	0
18	FLT_BYPASS1	IOCNT1 滤波功能配置 =0，滤波功能使能，按照 flt_cfg 配置进行滤波 =1，不使能滤波功能	R/W	0
17	FLT_BYPASS0	IOCNT0 滤波功能配置 =0，滤波功能使能，按照 flt_cfg 配置进行滤波 =1，不使能滤波功能	R/W	0
16	CNT_CLR	脉冲计数器类型配置 =0，所有脉冲计数器读后清零 =1，所有脉冲计数器读后不清零	R/W	0
15:13	--	保留	R	0
12:8	IOCNT_MODE	IOCNT4~IOCNT0 脉冲计数模式选择： =0，上升沿触发 =1，下降沿触发 默认上升沿触发	R/W	0
7:5	---	保留	R	0
4:0	IOCNTx_EN	IOCNT4~IOCNT0 脉冲转发功能使能配置： =0，不使能 =1，使能 IOCNT 注：脉冲转发功能使能后，脉冲转发模块启动工作，待其检测第一个待转发脉冲时，输出第一个转发脉冲，并开始计数，待其计数到分频数后，转发输出第二个脉冲，依此类推输出后面的脉冲。这种模式，会导致电表的第一个误差偏正；规避方法是，应用程序在数到“分频数-1”个脉冲后，再使能脉冲转发功能。	R/W	0

22.7 使用流程

22.7.1 内部脉冲转发

待补充

22.7.2 高电平脉宽有效

当输入信号高电平有效时，建议 INTx 硬件电路外接下拉电阻，使 INTx 默认输入为低电平，并使用上升沿计数，步骤如下：

- 1) 配置 IOCNT_CTL 寄存器 PXX_MODE，选择上升沿计数
- 2) 配置 GPIO 复用关系；
- 3) 可选配置分频计数和电平反向；
- 4) 配置脉冲转发使能。

22.7.3 低电平脉宽有效

当输入信号低电平有效时，建议 INTx 硬件电路外接上拉电阻，使 INTx 默认输入为高电平，并使用下降沿计数，步骤如下：

- 1) 配置 IOCNT_CTL 寄存器 PXX_MODE，选择上升沿计数
- 2) 配置 GPIO 复用关系；
- 3) 可选配置分频计数和电平反向；
- 4) 配置脉冲转发使能。

注：脉冲转发功能使能后，脉冲转发模块启动工作，待其检测第一个待转发脉冲时，输出第一个转发脉冲，并开始计数，待其计数到分频数后，转发输出第二个脉冲，依此类推输出后面的脉冲。这种模式，会导致电表的第一个误差偏正；规避方法是，应用程序在数到“分频数-1”个脉冲后，再使能脉冲转发功能。

23 安全密码加速器 SEA（新增）

安全密码加速器（Security Encryption Accelerator）是为对称加密算法（AES）、公钥密码算法（ECDSA、ECDH 等）提供硬件加速的模块。包含 AES 硬件加速单元、PKA 公钥算法加速器（ECC 硬件加速单元、RSA 硬件加速单元）、散列算法加速器、128 位有限域乘法单元、真随机数发生器。

以本模块为基础的固件 RS-SEA 通过 FIPS140-3 算法认证，认证的算法列表如下，详情请参见 NIST 官网：

[Cryptographic Algorithm Validation Program | CSRC \(nist.gov\)](https://csrc.nist.gov/Cryptographic-Algorithm-Validation-Program)。

模块	分项	具体内容	符合标准
AES	ECB	Key Length-128、192、256	SP800-38A
	CBC	Key Length-128、192、256	SP800-38A
	CTR	Key Length-128、192、256	SP800-38A
	CFB128	Key Length-128、192、256	SP800-38A
	OFB	Key Length-128、192、256	SP800-38A
	GMAC	Key Length-128、192、256	SP800-38D
	GCM	Key Length-128、192、256	SP800-38D
DRBG	CTR	Mode: AES-128、AES-192、AES-256	SP800-90A
HASH	SHA-1	Message Length: 0~65536 increment 8	FIPS 180-4
	SHA-224	Message Length: 0~65536 increment 8	FIPS 180-4
	SHA-256	Message Length: 0~65536 increment 8	FIPS 180-4
	SHA-384	Message Length: 0~65536 increment 8	FIPS 180-4
	SHA-512	Message Length: 0~65536 increment 8	FIPS 180-4
ECDSA	KeyGen	P-192、P-224、P-256、P-384、P-521	FIPS 186-4

KeyVer	P-192、P-224、P-256、P-384、P-521	FIPS 186-4
SigGen	P-192、P-224、P-256、P-384、P-521	FIPS 186-4
SigVer	P-192、P-224、P-256、P-384、P-521	FIPS 186-4

23.1 AES 硬件加速单元

23.1.1 特点

AES 硬件加速单元可以支持客户对数据包进行对称 AES 算法的加密/解密，其主要功能如下：

支持 128bit/192bit/256bit 的密钥长度；

支持 KeyExp 密钥扩展；

支持 ECB、CBC、CTR、CFB128、OFB、GCM 六种数据流处理模式；

支持 GF(2¹²⁸)域下的乘法，配合软件可以加速 GCM 算法中的身份认证（GMAC）过程；

23.1.2 时耗信息

模块内硬件加速单元	时耗 (Cycles)
128 bits 密钥拓展	126
192 bits 密钥拓展	141
256 bits 密钥拓展	162
128 bits 加/解密单元	55
192 bits 加/解密单元	63
256 bits 加/解密单元	71
128 位有限域乘法单元	128

注：时耗，指启动运算到运算结束耗费 AHB 时钟周期数 (Cycles)。

23.2 ECC 硬件加速单元

23.2.1 特点

ECC 硬件加速单元为椭圆曲线 (EC, Elliptic Curve) 运算加速，可显著提高基于椭圆曲线的加解密协议的实现效率。本模块配合软件可完成的椭圆曲线协议包括：NIST(P)、SEC(p)、SEC(k)、Brainpool 等，可完成的签名认证算法包括：ECDSA(EC Digital Signature Algorithm)、ECDH(EC Deffie-Hellman)及其变种算法。

其主要功能如下：

支持最高 521bit 字长的模加、模减运算；

支持最高 521bit 字长蒙哥马利参数预计算；

支持最高 521bit 字长的蒙哥马利模乘、模逆运算；

支持从点加、倍点运算前坐标从整数域到蒙哥马利域的域转换；

支持字长从 160bit~521bit 的倍点、点加两种基本点运算；

支持点运算结束坐标从蒙哥马利域到整数域的域转换；

注：锐能微为客户提供基于 ECC 硬件加速的算法库函数，便于客户快速实现 ECDSA 等算法应用。

23.2.2 时耗信息

硬件加速单元	时耗 (Cycles)
J 参数计算	64
H 参数计算	2021~24960
蒙哥马利模乘	2151~25670
蒙哥马利模逆	6330~108764

模加	21~50
模减	21~50
点加	7610~123982
倍点	7317~116492
算法	实际耗时（参考）
ECDSA_NIST-P256	72ms~78ms（29MHz 时钟）
ECDSA_Brainpoolp512r1	350ms~359ms（29MHz 时钟）

注：时耗，指启动运算到运算结束耗费 AHB 时钟周期范围（Cycles）。

23.3 RSA 硬件加速单元

23.3.1 特点

RSA 硬件加速单元为 RSA 加解密运算提供硬件加速，可显著提高 RSA 加解密协议的实现效率，本模块支持字长从 32bit 至 576bit 的 RSA 加解密。

其主要功能如下：

支持基于欧拉函数 $\varphi(N)=(P-1)(Q-1)$ 的 RSA 加解密协议；

支持字长从 32bit 至 576bit 的模幂运算；

支持字长从 32bit 至 2048bit 的模乘运算；

23.3.2 时耗信息

硬件加速单元	时耗（Cycles）
J 参数计算	64
H 参数计算	2021~24960
模乘	2366~28043
模幂	2675~1010246

注：时耗，指启动运算到运算结束耗费 AHB 时钟周期范围（Cycles）。

23.4 HASH 硬件加速单元

23.4.1 特点

按哈希值位数将安全散列算法分为：160 位的 SHA-1，224 位的 SHA-224，256 位的 SHA-256，384 位的 SHA-384，512 位的 SHA-512 等。

散列算法加速器主要功能如下：

支持 SHA-1、SHA-224、SHA-256、SHA-384、SHA-512 安全散列算法加速，实际执行分组和数据传入由软件完成；

支持 CPU 状态查询和中断模式；

23.4.2 时耗信息

模块内硬件加速单元	时耗（Cycles）
SHA-1	101
SHA-224	85
SHA-256	85
SHA-384	101
SHA-512	101

注：时耗，指启动运算到运算结束耗费 AHB 时钟周期数（Cycles）

23.5 TRNG 真随机数发生器

23.5.1 特点

真随机数发生器使用斐波拉契环振作熵源，可在产生 ns 级尺度随机 bit 流。特点如下：

- 最高支持系统主频 29MHz 输出随机数比特流；
- 可以以当前系统主频/N（1~65536）的频率输出随机数比特流；
- 支持随机数随机性检测并报错功能，支持纠错算法；
- 支持 LFSR 伪随机算法；
- 支持地址交织随机性增强算法；

支持以中断方式每次输出 128bit 随机数；

23.5.2 时耗信息

生成数据类型	时耗 (Cycles)
使能模块后第一组 128 bits 数据	209
其他组 128 bits 数据	128

注：时耗，启动运算到 128bits 生成耗费 AHB 时钟周期数 (Cycles)

23.6 软件配置流程

请参见“锐能微库函数”关于本模块固件 RS-SEA v1.0.0 相关章节，使用流程及应用方案请参见“锐能微应用笔记”。

24 简单定时器 SIMP_TC（新增）

24.1 概述

模块内部集成了 4 个的 32 位系统节拍计数器，有两种工作模式：

- 单次工作模式；
- 循环工作模式；

24.2 功能描述

SIMP_TC 模块是一个简单的 32 位系统节拍计数器，使用 CPU 运行的系统时钟进行计数，当计数值达到预设的目标值时，产生溢出标志位；

计数器支持单次和循环两种不同的模式，单次模式在启动计数后，计数值达到预设的目标值后计数器产生溢出标志位，并停止计数；循环模式在启动计数后，计数值达到预设的目标值后计数器产生溢出标志位，并从 0 开始继续计数。

计数器在软件使能计数使能位后开始计数，软件禁止计数使能位后停止计数。

24.3 寄存器描述

SIMP_TC 寄存器基址

模块名	物理地址	映射地址
SIMP_TC0	0x40060000	0x40060000
SIMP_TC1	0x4006000C	0x4006000C
SIMP_TC2	0x40060018	0x40060018
SIMP_TC3	0x40060024	0x40060024

SIMP_TC 接口的寄存器偏移地址

寄存器名(x 指 0/1/2/3)	地址偏移量	描述
SIMP_TCx_CTRL	0x0	SIMP_TCx 控制寄存器
SIMP_TCx_LOAD	0x4	SIMP_TCx 目标计数值寄存器
SIMP_TCx_VAL	0x8	SIMP_TCx 当前计数值寄存器

24.3.1 SIMP_TCx 控制寄存器 CTRL (0x0)

比特位	名称	描述	读/写标志	复位值
31:4	---	预留	R	0
3	IRQEN	中断使能位 0: 关闭中断 1: 使能中断	R/W	0

2	MODE	计数模式位 0: 单次计数 1: 循环计数	R/W	0
1	OV	计数溢出标志位 0: 计数器没有溢出 1: 计数器溢出 注: 写 1 清零	R/W	0
0	EN	计数使能位 0: 停止计数 1: 开始计数	R/W	0

24.3.2 SIMP_TCx 目标计数值寄存器 LOAD (0x4)

比特位	名称	描述	读/写标志	复位值
31:0	LOAD	目标计数值: LOAD + 1	R/W	0

24.3.3 SIMP_TCx 当前计数值寄存器 VAL (0x8)

比特位	名称	描述	读/写标志	复位值
31:0	VAL	计数器当前计数值	R	0

25 内存搬运单元 M2M (新增)

Mem2Mem 模块采用 DMA 方式, 在 CPU 不参与的情况下, 自动在 SRAM 中进行数据搬运, 从指定的源地址把数据搬运到目标地址。搬运可配置的最小数据粒度为 1Byte, 同时还支持源数据和目标数据有效位配置。该模块仅适用于 SRAM 位宽为 32bit 的系统。

25.1 特点

- 源和目标地址可配置, 地址可配置 Byte 对齐
- 源数据长度可配置, 最小单位为 Byte
- 输入数据有效位可配置
- 输出数据有效位可配置
- 可配置输出数据按 Byte 反序存放到目标地址
- 可配置输出数据按照地址反序存放到目标地址
- 数据搬运完成会产生标志, 可配置产生中断

25.2 功能描述

25.2.1 输入输出数据有效位宽配置

这里的输入输出数据配置, 均指每 4Bytes 数据中有几个 Bytes 有效。

支持输入和输出数据有效位宽可通过寄存器为 ivld[3:0]和 ovld[3:0]进行独立配置, 每 4 个 Bytes 数据, 可配置 1~4Bytes 有效。

输入数据的 byte 无效表示源数据中, 每 32bit 数据中对应的 byte 将在数据搬运过程被舍弃。

输出数据的 byte 无效表示目标地址中, 每 32bit 的 SRAM, 对应的 Byte 不进行有效数据的存放。

- 当输入输出数据有效位宽均配置为 4 个 Bytes 时，可实现数据在 SRAM 中的直接搬移；
- 当输入数据有效位宽小于 4 个 Bytes 时，可实现输入源数据中无效数据的丢弃；
- 当输出数据有效位宽小于 4 个 Bytes 时，可实现目标数据插入 dummy 的功能；
- 也可通过输出输出的组合实现其他功能；
具体功能看后面实例描述。

25.2.2 输出地址无效时可配 dummy 值

可通过寄存器配置实现，当输出地址的位无效时，填入配置的寄存器值。默认情况下，对无效地址不进行写操作。

源数据长度 $ilen$ 、源数据有效位宽 $ivld$ 、目标数据有效位宽 $ovld$ 。有可能出现 $ilen*ivld/ovld$ 非整数的情况，该应用定义为非常规应用。按如下处理：

- (1) 如果 $DUMMY_EN=0$ ，则有效 byte 写入对应的有效数据，多余的 byte 不进行写操作；
- (2) 如果 $DUMMY_EN=1$ ，则有效 byte 写入对应的有效数据，多余的 byte 写入 dummy 值。

25.2.3 输出数据反序放置

配置寄存器 $M2M_MODE.ORV_EN=1$ ，可使得输出 Word 数据反序。该反序功能仅根据输出 Word 数据的位宽决定，比如输出 Word 数据字节位宽为 3Bytes，则每次取到源数据的 3 个 Bytes 的数据（A0，A1，A2），进行一次反序排列（A2，A1，A0），然后存放到 SRAM 中。

25.2.4 源和目标地址可配置，输入数据长度可配置

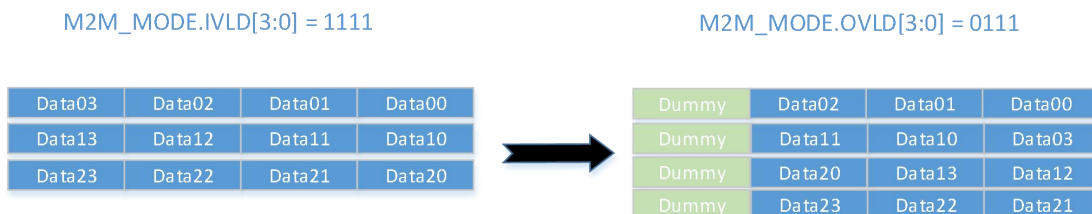
通过寄存器配置源数据地址和目标数据地址，支持按 Word 对齐的配置。同时输入数据长度可通过寄存器配置，最小刻度为 Word。输出数据长度根据输入输出长度和输入输出数据有效位进行自适应。

25.2.5 数据按地址反序排列

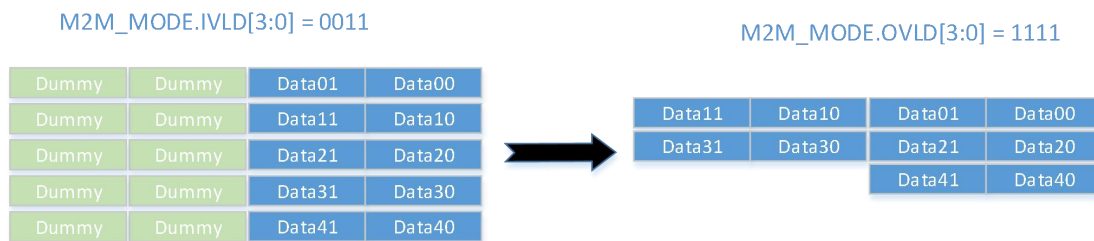
配置寄存器为 $addr_rven=1$ ，可使得输出数据实现地址反序放置。比如取到源数据的 4 个 Bytes 的数据（A3，A2，A1，A0），将（A3，A2，A1，A0）写到目的地址中，之后目的地址减 4，存储下一次取到源数据的 4 个 Bytes。以此类推，直到存完最后一个 byte。

25.3 实例说明

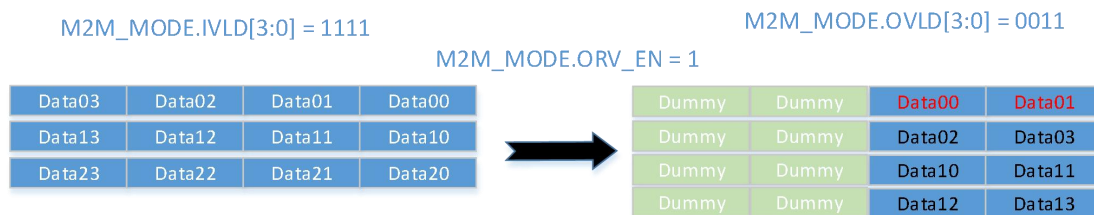
25.3.1 输入数据 4Bytes 有效，输出 3Bytes 有效



25.3.2 输入数据 2Bytes 有效，输出 4Bytes 有效

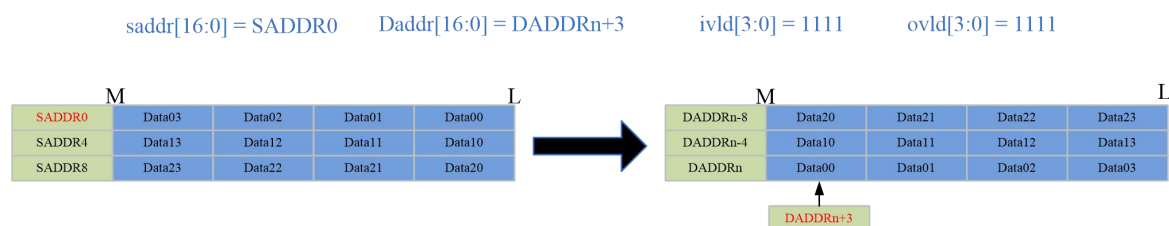


25.3.3 输入数据按 byte 反序存放（输入 4Bytes 有效，输出 2Bytes 有效）



从上图可以看到，输出地址与输入地址反序排放；源地址中 data00 放到一个 Word 的最低地址，目标地址中，data00 放在一个 Word 的最高地址。

25.3.4 输出数据按地址反序排列



从上图可以看到，首先从 SADDR0 取数据（data03，data02，data01，data00），写到地址 DADDRn+3 中，按 byte 从高地址写到低地址，根据输出有效 ovld，存储下次的数据，直到最后一个 byte。SADDR 地址和 DADDR 地址按照 Byte 对齐，输入输出有效和长度可以任意配置。

25.4 搬运速度

最快可以做到 6 个周期完成一个 Word 的数据搬运。实际速度还跟配置的 M2M_MODE.IVLD 和 OVLD，以及 SRAM 的竞争相关。

25.5 寄存器描述

模块名	物理地址	映射地址
M2M	0x4006_8000	0x4006_8000

M2M 模块的寄存器偏移地址

寄存器名	地址偏移量	R/W	复位值	描述
M2M_MODE	0x00	R/W	0xFF	M2M 模式配置寄存器
M2M_CTL	0x04	R/W	0x0	M2M 控制寄存器
M2M_DUMMY	0x08	R/W	0x0	M2M 无效地址 DUMMY 值配置寄存器
M2M_SADDR	0x0C	R/W	0x0	M2M 源数据地址配置寄存器
M2M_DADDR	0x10	R/W	0x0	M2M 目标数据地址配置寄存器

M2M_ILEN	0x14	R/W	0x0	M2M 源数据长度配置寄存器
M2M_IE	0x18	R/W	0x0	M2M 中断使能寄存器
M2M_IF	0x1C	R/W	0x0	M2M 中断标志寄存器

以上寄存器不支持 bitband 操作。

25.5.1 M2M_MODE (0x0)

M2M 模式配置寄存器，地址：0x4006_8000；默认值：0xFF；

位	位名称	功能描述	读/写	复位值
31:11	--	保留	RO	0
10	ADDR_RVEN	数据按 Word 地址反序排列 =0：不反序 =1：按 Word 地址反序排列	RW	0
9	DUMMY_EN	=0：目标地址中的无效 byte 位置，不进行写操作； =1：目标地址中的无效 byte 位置，写入 dummy 值；	RW	0
8	ORV_EN	输出 Word 数据按 Byte 反序放置 =0：不反序 =1：反序	RW	0
7:4	OVLD	输出 Word 数据有效 Byte 配置 =0：Byte 无效 =1：Byte 有效 对应 Byte 为 0，表示目标地址中的该 Byte 地址无效或 M2M_MODE.DUMMY_EN 有效时写入 dummy 值。	RW	0xf
3:0	IVLD	输入 Word 数据有效 Byte 配置 =0：Byte 无效 =1：Byte 有效 4 个 bit 配置按 Byte 与每个 Word 数据中的每一个 Byte 一一对应，高电平表示对应 byte 有效。	RW	0xf

25.5.2 M2M_CTL (0x4)

M2M 控制寄存器，地址：0x4006_8004；默认值：0x0；

位	位名称	功能描述	读/写	复位值
31:1	--	保留	RO	0
0	M2M_EN	=0：不动作 =1：启动数据搬运。指定长度数据搬运完成之后自动清零。 当该位为 1 时，写 0 可终止当前搬运过程。	RW	0

25.5.3 M2M_DUMMY (0x8)

M2M 无效地址 DUMMY 值配置寄存器，地址：0x4006_8008；默认值：0x0；

位	位名称	功能描述	读/写	复位值
31:0	DUMMY	当 M2M_MODE.DUMMY_EN=1 时，该寄存器有效。 该 32bit 值按 Byte 与 M2M_MODE.OVLD 配置无效地址的 Byte 一一对应。	RW	0

25.5.4 M2M_SADDR (0xC)

M2M 源数据地址配置寄存器，地址：0x4006_800C；默认值：0x0；

位	位名称	功能描述	读/写	复位值
31:17	--	保留	RO	0
16:0	SADDR	源数据地址，按 Byte 对齐。 这里的地址为 SRAM 偏移地址，源数据地址+源数据长度不能超出 SRAM 地址范围	RW	0

25.5.5 M2M_DADDR (0x10)

M2M 目标数据地址配置寄存器，地址：0x4006_8010；默认值：0x0；

位	位名称	功能描述	读/写	复位值
31:17	--	保留	RO	0
16:0	DADDR	目标数据地址，按 Byte 对齐。 这里的地址为 SRAM 偏移地址，目标数据地址+目标数据长度不能超出 SRAM 地址范围	RW	0

25.5.6 M2M_ILEN (0x14)

M2M 源数据长度配置寄存器，地址：0x4006_8014；默认值：0x0；

位	位名称	功能描述	读/写	复位值
31:17	--	保留	RO	0
16:0	ILEN	输入数据长度配置，按 Word 对齐 源数据长度不能超出 SRAM 的大小。 应用注意：不允许在 ILEN=0 使能 M2M。	RW	0

25.5.7 M2M_IE (0x18)

M2M 中断使能寄存器，地址：0x4006_8018；默认值：0x0；

位	位名称	功能描述	读/写	复位值
31:1	--	保留	RO	0
0	DONE_IE	搬运完成中断使能 =0：不使能 =1：使能中断	RW	0

25.5.8 M2M_IF (0x1C)

M2M 中断标志寄存器，地址：0x4006_801C；默认值：0x0；

位	位名称	功能描述	读/写	复位值
31:1	--	保留	RO	0
0	DONE	搬运完成标志，写 1 清零 =0：搬运未完成 =1：搬运完成 每次配置 M2M_CTL.M2M_EN=1，该位自动清零。	RW1C	0

25.6 软件使用流程

- 1) 配置源数据起始地址 M2M_SADDR;
- 2) 配置目标数据起始地址 M2M_DADDR;
- 3) 配置源数据长度 M2M_ILEN;
- 4) 配置模式寄存器 M2M_MODE;
- 5) 配置中断使能 M2M_IE;
- 6) 配置启动 M2M_CTL;
- 7) 等待中断或标志, 从目标地址读取数据, 并清除标志;

26 DSP 核（新增）

本模块为数据信号处理 DSP 模块, 包含浮点数格式的 FFT 运算、整数和浮点数相互转换的运算单元、浮点数运算基本单元、cordic 算法计算正余弦以及均方根, IIR 加速运算、FIR 加速运算、线性插值运算、拉格朗日插值运算。

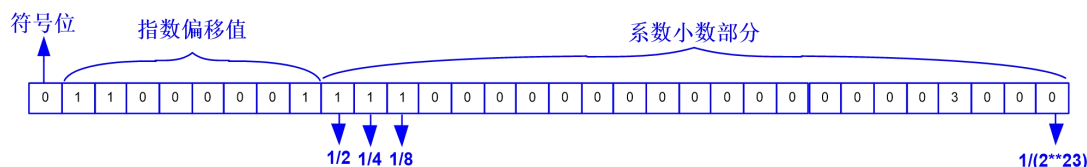
此处所说的浮点数均为单精度浮点数（小数点后有效数字 23 位（2 进制））。

26.1 特点

- 支持整数和浮点数的相互转换；
- 支持浮点数加法、减法、乘法、除法；
- 支持单次蝶形运算（复数）以及 DMA 连续蝶形运算；
- 支持基-2 FFT 整个过程，支持点数 64、128、256、512、1024；
- 支持 bit reverse 自动搬运数据操作，支持点数 4、8、16、32、64、128、256、512、1024；
- 支持正弦、余弦计算；
- 支持均方根计算；
- 支持反正切计算；
- 支持 IIR 单次计算，支持 IIR DMA 运算；
- 支持 FIR 滤波运算；
- 支持线性插值运算；
- 支持拉格朗日插值运算；

26.2 计算引擎基本原理

26.2.1 单精度浮点数表示



上图浮点数表示方法。浮点数数值= $\pm 2^n(1 + f)$ ，其中 n 为指数值， f 为小数部分。

符号位代表浮点数的正负值，0 表示整数、1 表示负数。

指数部分，采用移码表示，也即是 8 位“指数偏移值”中最高位为指数符号位，低 7 位为实际指数值。比如指数值 1，用移码表示为 $1+127=128$ ；指数值 -10，用移码表示为 $-10+127=117$ 。

系数小数部分，代表小数点之后的数。

26.2.2 特殊数值

分类	符号位(1bit)	指数偏移值(8bit)	系数小数部分(23bit)
0	0/1	0	0
非正常	0/1	0	非 0
无穷	0/1	255	0
NaN	0/1	255	非 0
正常	0/1	1~254	任意

上表列出了浮点数表示中的一些特殊数值，下面再做浮点数运算时需要判断输入输出数据是否为特殊值，做特殊处理，并给出标志位。

26.2.3 浮点数舍入处理

当浮点数运算中出现系数小数部分的位宽大于实际能表示的位宽时，需要对数据做舍入处理，也就是 rounding。

首先定义以下概念：



如上图，m 位数据要截断为 n 位时：

- rounding bit: 从左边最高位开始，第 n+1 位；
- stick bit: 从左边最高位开始，从低 n+2 位之后的所有位；

26.2.4 IEEE754 标准 rounding 模式

IEEE754 标准，舍入方法有 4 种

1. nearest: 就近舍入，rounding bit=1 和 stick bit 非 0 则进 1
2. $+\infty$: 对于正数，rounding bit=1 或 stick bit 非 0 则进 1，其余不进位
3. $-\infty$: 对于负数，rounding bit=1 或 stick bit 非 0 则进 1，其余不进位
4. zero : rounding bit 和 sticky bit 直接舍弃

26.2.5 本芯片的 rounding 模式

本芯片支持 6 中舍入方法，根据配置选择：

1. 当配置 rnd=000，为 nearest 舍入；
2. 当配置 rnd=001，为 zero 舍入；
3. 当配置 rnd=010，为 $+\infty$ 舍入；
4. 当配置 rnd=011，为 $-\infty$ 舍入；
5. 当配置 rnd=100，为 up 向上进位；
6. 当配置 rnd=101，为 away 直接舍弃；

26.2.6 整数转浮点数原理

下面简述整数转浮点数的思路与原理。

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

1. 符号位：
 - a) 最高位即为浮点数符号位：fp_sign = bit(31);
 - b) 如果原始数据为负数，则需要转成补码，再做下一步处理；
2. 指数：
 - a) 从最高位开始，查找第一个“1”，所在位置记录为 n。其中指数值就为 n（上表为 27）；
 - b) 根据编码规则，指数表示 fp_exp=n+127；
3. 小数：
 - a) 如果 n>23，则从 n-1 位开始，截取 24bit，记为 m；对原始数据中剩余位做 rounding 处理，决定 m 最低位的值。m 即为小数部分；
 - b) 如果 n=23，则从 n-1 位开始，截取 24bit，直接做为小数部分；
 - c) 如果 n<23，则从 n-1 位开始，截取剩余所有位，并且低位补 0，作为小数部分；

26.2.7 浮点数转整数原理

1. 判断原始数据是否为特殊数（0，NaN， $\pm\infty$ ）
2. 根据原始数据的指数部分 e ，判断转化结果是否溢出。如果指数部分 $e > 32 + 127$ ，则转换结果溢出；否则进行下面步骤：
3. 取出 23 位小数部分 m ，最高位补“1”，凑成 24 位小数 m ；
4. 把小数 m 左移 e 位，高位（ $(32 - e)$ bit）和低位（ e bit）补“0”，得到 56 位数据；
5. 结果右移 24 位，得到最终结果；

26.2.8 浮点数乘法原理

1. 特殊数据相乘：
 - a) NaN 和任何数相乘都是 NaN，
 - b) 0 和无穷相乘为 NaN，
 - c) 0 和其他数相乘为 0。
2. 普通数据相乘：
 - a) 符号位相异或得到符号位结果；
 - b) 指数部分相加再减去 127（ $e = e_0 + e_1 - 127$ ）；
 - c) 系数部分先分别扩展一位（共 24bit）之后做乘法，得到一个 48bit 数据 m ，判断结果是否溢出，如果溢出，则 $e = e + 1$ ，判断指数 e 是否溢出，同时把 m 右移三位，否则把 m 右移 2 位；取高 23 位为系数结果。
3. 溢出处理，支持 IEEE754 规定的 4 中溢出处理方式，通过寄存器配置。

26.2.9 浮点数加法原理

1. 判断特殊数据，NaN 与任何数相加都是 NaN；
2. 正常数相加：
 - a) 用绝对值大的数的符号位作为输出数据符号位；
 - b) 判断两个指数部分大小，用大的数 e_0 减去小的数 e_1 得到相差倍数 F ；
 - c) 对指数相对较小的数的系数部分右移 F 位，然后两个系数部分都扩 1 位相加；
 - d) 指数：如果溢出，则指数 $e = e_0 + 1$ ，否则 $e = e_0$ ；
 - e) 系数部分去掉相加之后扩充的高 2bit，如果溢出，系数部分右移 1 位（相当于除 2），即为结果系数部分；

26.2.10 蝶形运算原理

$$X'_1(k) = X_1(k) + W_N^k X_2(k)$$

$$X'_2(k) = X_1(k) - W_N^k X_2(k)$$

其中 $X_1(k)$ 、 $X_2(k)$ 为输入， $X'_1(k)$ 、 $X'_2(k)$ 为蝶形运算输出。

如果原始数据为复数，则有：

$$X_1(k) = X_{1r}(k) + jX_{1i}(k)$$

$$X_2(k) = X_{2r}(k) + jX_{2i}(k)$$

$$W_N^k = W_{Nr}^k + jW_{Ni}^k$$

$$X'_1(k) = X'_{1r}(k) + jX'_{1i}(k)$$

所以：

$$X'_1(k) = X_1(k) + W_N^k X_2(k)$$

$$= (X_{1r}(k) + jX_{1i}(k)) + (W_{Nr}^k + jW_{Ni}^k) * (X_{2r}(k) + jX_{2i}(k))$$

把实部虚部提取，

$$X'_{1r} = X_{1r} + W_{Nr}^k X_{2r} + W_{Ni}^k X_{2i}$$

$$X'_{1i} = X_{1i} + W_{Nr}^k X_{2i} - W_{Ni}^k X_{2r}$$

同理：

$$X'_{2r} = X_{1r} - W_{Nr}^k X_{2r} - W_{Ni}^k X_{2i}$$

$$X'_{2i} = X_{1i} - W_{Nr}^k X_{2i} + W_{Ni}^k X_{2r}$$

26.2.11

IIR 滤波器原理

IIR 滤波器是线性数字滤波器中最为常见的一种类型。在一个给定的时间上 IIR 的输出依赖于它们的输入和先前计算的输入值。

IIR 滤波器是递归的，原因是差分方程有反馈，所以它的稳定时间比 FIR 滤波器要长，冲激响应可能有无限宽度，需要注意滤波器的稳定性问题。

IIR 滤波器的传递函数（z 域）如下：

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{a_0 + \sum_{k=1}^N a_k z^{-k}} = \frac{Y(z)}{X(z)}$$

IIR 滤波器的时域计算公式（直接 I 型）如下：

$$y[n] = \sum_{k=0}^M b_k x[n-k] - \sum_{k=1}^N a_k y[n-k]$$

对于二阶 IIR 滤波器，即上述公式中的 M 和 N 均为 2，一般而言， a_0 也会设计为 1，那么上述公式就会变为：

$$y[n] = b_0 \times x(n) + b_1 \times x(n-1) + b_2 \times x(n-2) - a_1 \times y(n-1) - a_2 \times y(n-2)$$

其中 $x(n)$ 是当前输入数据， $x(n-1)$ 是上一个输入数据（时间上滞后一个单位）， $x(n-2)$ 是再上一个输入数据（时间上滞后两个单位）； $y(n)$ 是当前输出数据， $y(n-1)$ 是上一个输出数据（时间上滞后一个单位）， $y(n-2)$ 是再上一个输出数据（时间上滞后两个单位）。

26.2.12

FIR 滤波器原理

FIR 滤波器是非递归型滤波器的简称，又叫有限长单位冲激响应滤波器。带有常系数的 FIR 滤波器是一种 LTI(线性时不变)数字滤波器。冲激响应是有限的意味着在滤波器中**没有反馈**。长度为 N 的 FIR 输出对应于输入时间序列 $x[n]$ 的关系由一种有限卷积和的形式给出，具体形式如下：

$$y[n] = \sum_{k=0}^N a[k] * x[n-k]$$

上式中， $a[k]$ 为滤波器的系数， $x[n-k]$ 为 $x[n]$ 延时 k 个周期。

直接形式 FIR 滤波器图解：

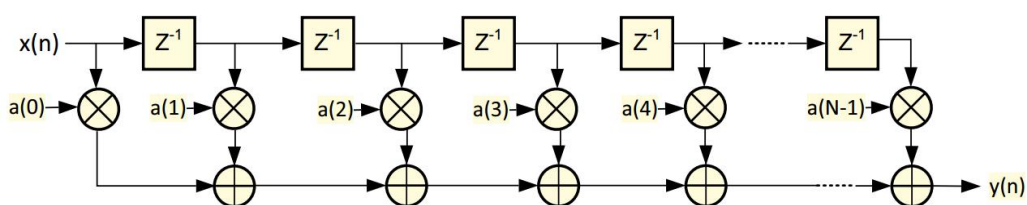


Figure 1-1: Conventional Tapped Delay Line FIR Filter Representation

上式表达的是一个 $N-1$ 阶的 FIR 滤波器，它有 N 个抽头（系数）。因此有 N 个乘法器， $N-1$ 个累加器组成。输入信号是有时间性的，随着时间的改变而改变，FIR 滤波器最终的输出是各个时刻的输入乘以相应的权重（系数），然后进行叠加输出。

以 4 阶 FIR 为例，那么有 5 个系数，采用串行的方法实现，需要调用 5 次乘法器，4 次加法器完成一次计算：
 $y[n] = a_0 * x[n] + a_1 * x[n-1] + a_2 * x[n-2] + a_3 * x[n-3] + a_4 * x[n-4]$

26.2.13

线性插值

已知 n 点： $(1, y_1), \dots, (n, y_n)$ ，求： $y(i)$ ， x 为第 i 个输出数据横坐标

公式：

$$y(i) = y_{[x]} + (x - [x]) * (y_{[x]+1} - y_{[x]}), \quad [x] = x \text{ 向下取整}$$

逐点计算输出数据，每次循环插值位置增加 step:

```
for i=1:1:out_len
    n=floor(t);
    y[i]= x[n]+ (t-n) * (x[n+1]-x[n]);
    t=t+step;
End
```

26.2.14

Lagrange 插值

已知 n 点: $(1, y_1), \dots, (n, y_n)$, 求: $y(i)$, x 为第 i 个输出数据横坐标

3 点 Lagrange 插值即每个小区间一个二次曲线, 公式:

$cur = round(x)$;

$$y(i) = \frac{x - cur}{2} * [(x - (cur + 1)) * y_{cur-1} + (x - (cur - 1)) * y_{cur+1}] - (x - (cur - 1)) * (x - (cur + 1)) * y_{cur}$$

逐点计算输出数据, 每次循环插值位置增加step:

```
for i=1:1:out_len
    n=floor(t);
    y[i]=(t-n)*0.5*((t-n-1)*x[n-1]+(t-n+1)*x[n+1]) - (t-n+1)*(t-n-1)*x[n]
    t=t+step;
End
```

26.3 运算指令

26.3.1 整数转浮点数 (int2fp/int2fp_dma)

26.3.1.1 单数据转换



整数转浮点数

该模式把寄存器输入的 **32bit (或者 24bit) 有符号整数** 转化为 32 位单精度浮点数格式, 输出到寄存器, CPU 可读。

支持对转换后的浮点数做归一化处理, 也就是除以 2^n 。其中 n 可配置为 0~32, 当配置为 0 时, 表示不做归一化处理。

默认输入寄存器整数为 32bit 有符号数, 对于特殊应用, 如果输入整数为 24bit 有符号数, 则需要配置寄存器 $MAC_CTL0[28]=1$, 选择该模式。

软件操作流程:

- 1) 配置寄存器 $MAC_CTL0[0]=1$ 选择整数转浮点数单点模式;
- 2) 配置寄存器 $MAC_CTL0[16:14]$ 选择浮点数的 [rounding mode](#);
- 3) 配置 $MAC_CTL0[21:17]$ 选择归一化级数;
- 4) 对于 24bit 输入数据, 配置 $MAC_CTL[0]=1$;
- 5) 往寄存器 MAC_IN0 写入待转换的原始整数;
- 6) 读取寄存器 MAC_OUT0 获得转换后的浮点数值;

软件在配置完模式之后, 只要配置寄存器 MAC_IN0 就会启动转换, 计算结果保存到寄存器 MAC_OUT0 , 所以配置原始数值之后直接读取结果寄存器可以获得转换后的结果。

注意, 如果软件需要连续转化多次, 上面的模式配置只需要执行一次, 在切换模式之前, 每次只要写寄存器 MAC_IN0 , 就会执行转换。

26.3.1.2 多数据连续转换

该模式支持把 SRAM 中一整段整数转化为浮点数。并且结果写回 SRAM。注意存放在 SRAM 中的原始数据为带符号整数（32bit 有符号整数或者 24bit 有符号整数）。对于特殊应用，如果 24bit 整数占用 4Byte，有效数据存放在低 3Byte 位置，并且最高 Byte 无效，则需要配置 MAC_CTL[0]=1。

可配置目标地址与源数据地址一致，这样会做到转换完数据覆盖原始数据，从而节省 SRAM 空间。否则配置目标地址与源数据地址不同，会把转换后的数据写到 SRAM 中另外区域。

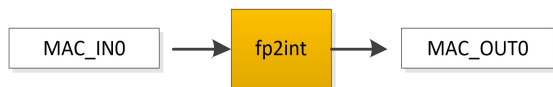
软件操作流程：

- 1) 配置寄存器 MAC_CTL0[1]=1 选择整数转浮点数 DMA 模式；
- 2) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#)；
- 3) 配置 MAC_CTL0[21:17]选择归一化级数；
- 4) 配置寄存器 DMA_SRBADR 选择源数据 DMA 起始地址；
- 5) 配置寄存器 DMA_TRBADR 选择目标数据 DMA 起始地址；
- 6) 配置寄存器 DMA_LEN 选择 DMA 长度；
- 7) 配置寄存器 MAC_CTL1[0]=1 启动转换；
- 8) 等待标志位 MAC_FLG[3]置 1，CPU 可从 SRAM 中获得转换后的结果；

在转换完指定长度数据之后，会产生完成标志，如果配置了中断使能信号，则会产生完成中断。软件需要在中断中清除标志位

26.3.2 浮点数转整数（fp2int/fp2int_dma）

26.3.2.1 单数据转换



浮点数转整数

该模式把寄存器输入的 32bit 浮点数转化为 **32 位有符号整数**格式，输出到寄存器，CPU 可读。支持转换之前先把浮点数放大 2^n 。其中 n 可配置为 0~32，当配置为 0，表示不放大。

软件操作流程：

- 1) 配置寄存器 MAC_CTL0[2]=1 选择浮点数转整数单点模式；
- 2) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#)；
- 3) 如果需要把转换前的浮点数放大，配置 MAC_CTL0[26:24]=n；
- 4) 往寄存器 MAC_IN0 写入待转换的原始浮点数；
- 5) 读取寄存器 MAC_OUT0 获得转换后的整数值。

软件在配置完模式之后，只要配置寄存器 MAC_IN0 就会启动转换，计算结果保存到寄存器 MAC_OUT0，所以配置原始数值之后直接读取结果寄存器可以获得转换后的结果。

注意，如果软件需要连续转化多次，上面的模式配置只需要执行一次，在切换模式之前，只要写寄存器 MAC_IN0，就会执行转换。

26.3.2.2 多数据连续转换

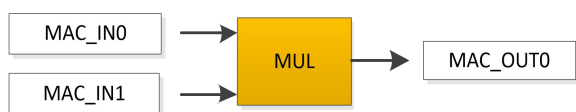
该模式支持把 SRAM 中一整段浮点数转化为整数。并且结果写回 SRAM。可配置目标地址与源数据地址一致，这样会做到转换完数据覆盖原始数据，从而节省 SRAM 空间。否则配置目标地址与源数据地址不同，会把转换后的数据写到 SRAM 中另外区域。

软件操作流程：

- 1) 配置寄存器 MAC_CTL0[3]=1 选择浮点数转整数 DMA 模式;
- 2) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#);
- 3) 如果需要把转换前的浮点数放大, 配置 MAC_CTL0[26:24]=n;
- 4) 配置寄存器 DMA_SRBADR 选择源数据 DMA 起始地址;
- 5) 配置寄存器 DMA_TRBADR 选择目标数据 DMA 起始地址;
- 6) 配置寄存器 DMA_LEN 选择 DMA 长度;
- 7) 配置寄存器 MAC_CTL1[0]=1 启动转换;
- 8) 等待标志位 MAC_FLG[4], CPU 可从 SRAM 中获得转换后的结果;

在转换完指定长度数据之后, 会产生完成标志, 如果配置了中断使能信号, 则会产生完成中断。软件需要在中断中清除标志位。

26.3.3 浮点数乘法 (fp_mult)



浮点数乘法

该模式对输入寄存器 MAC_IN0 和 MAC_IN1 的数据, 做一个浮点数乘法。对输入和输出数据格式支持下列四种模式:

- 1) 输入整数、输出整数
- 2) 输入整数, 输出浮点数
- 3) 输入浮点数、输出整数
- 4) 输入浮点数, 输出浮点数

如果输入为整数, 硬件会先把整数转换成浮点数格式, 然后再进行浮点乘法; 运算得到浮点数格式的乘积, 如果选择输出整数格式, 则硬件会把浮点格式的乘积转换为整数格式。

软件操作流程:

- 1) 配置寄存器 MAC_CTL0[4]=1 选择浮点数乘法模式;
- 2) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#);
- 3) 配置寄存器 MAC_CTL0[13:12]选择输入输出数据模式;
- 4) 配置寄存器 MAC_IN0 和 MAC_IN1 (乘数和被乘数);
- 5) 等待标志位 MAC_FLG[9]=1, 表示乘法完成;
- 6) 读取寄存器 MAC_OUT0 获得乘积;

该模式下, 每次配置 MAC_IN1 寄存器, 会自动启动乘法运算, 所以软件应该先配置好 MAC_IN0 寄存器, 再配置 MAC_IN1 寄存器。而且每次配置 MAC_IN1 寄存器, 会自动清除上一次的标志位, 直到当前计算完成, 再置起标志位。

由于输入输出数据格式不同, 需要对数据做格式转换, 所以做一次浮点乘法运算的时间与模式相关。

26.3.4 浮点数加法 (fp_add)



浮点数加法

该模式对输入寄存器 MAC_IN0 和 MAC_IN1 的数据，做一个浮点数加法。输入寄存器的数据格式必须为浮点数格式，输出结果也为浮点数格式。

软件操作流程：

- 1) 配置寄存器 MAC_CTL0[5]=1 选择浮点数加法模式；
- 2) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#)；
- 3) 配置寄存器 MAC_IN0 和 MAC_IN1（加数和被加数）；
- 4) 读取寄存器 MAC_OUT0 获得和；

该模式下，每次配置 MAC_IN1 寄存器，会自动启动加法运算，所以软件应该先配置好 MAC_IN0 寄存器，再配置 MAC_IN1 寄存器。

26.3.5 浮点数减法 (fp_sub)



浮点数加法

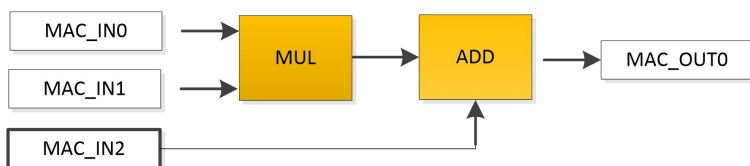
该模式对输入寄存器 MAC_IN0 和 MAC_IN1 的数据，做一个浮点数减法。输入寄存器的数据格式必须为浮点数格式，输出结果也为浮点数格式。

软件操作流程：

- 1) 配置寄存器 MAC_CTL0[6]=1 选择浮点数减法模式；
- 2) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#)；
- 3) 配置寄存器 MAC_IN0 和 MAC_IN1（被减数和减数）；
- 4) 读取寄存器 MAC_OUT0 获得差；

该模式下，每次配置 MAC_IN1 寄存器，会自动启动减法运算，所以软件应该先配置好 MAC_IN0 寄存器，再配置 MAC_IN1 寄存器。

26.3.6 浮点数乘加运算 (fp_mlad)



浮点数乘加运算

该模式对输入寄存器 MAC_IN0 和 MAC_IN1 的数据，先做一个浮点数乘法，然后再与 MAC_IN2 做加法，

最后输出为 MAC_OUT0。输入寄存器的数据格式必须为浮点数格式，输出结果也为浮点数格式。

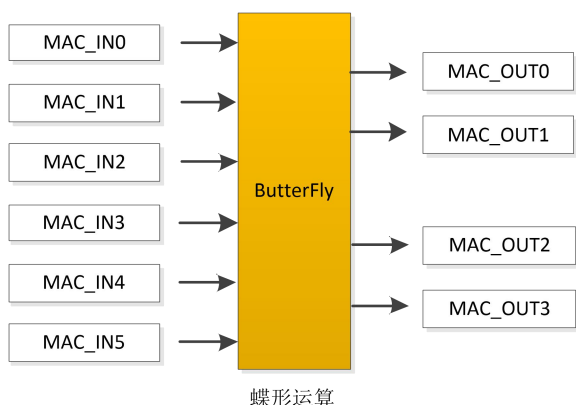
软件操作流程:

- 5) 配置寄存器 MAC_CTL0[7]=1 选择浮点数乘加模式;
- 6) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#);
- 7) 配置寄存器 MAC_IN0、MAC_IN1 和 MAC_IN2 (乘数、被乘数和被加数);
- 8) 读取寄存器 MAC_OUT0 获得乘加结果;

该模式下,每次配置 MAC_IN2 寄存器,会自动启动乘加运算,所以软件应该先配置好 MAC_IN0、MAC_IN1 寄存器,再配置 MAC_IN2 寄存器。

26.3.7 浮点数蝶形运算 (单次) (btfy /btfy_dma)

26.3.7.1 单数据转换



该模式对输入寄存器的两个原始数据、一个参数做一次 butterfly 操作,结果保存到寄存器中,供 CPU 读取。该模式下,输入的原始数据和参数必须为浮点数格式。

$$X'_{1r} = X_{1r} + W_{Nr}^k X_{2r} + W_{Ni}^k X_{2i}$$

$$X'_{1i} = X_{1i} + W_{Nr}^k X_{2i} - W_{Ni}^k X_{2r}$$

$$X'_{2r} = X_{1r} - W_{Nr}^k X_{2r} - W_{Ni}^k X_{2i}$$

$$X'_{2i} = X_{1i} - W_{Nr}^k X_{2i} + W_{Ni}^k X_{2r}$$

一次 butterfly 执行上面运算公式,式中, X_{1r} 、 X_{1i} 为输入原始数据 1 的实部和虚部, X_{2r} 、 X_{2i} 为输入原始数据 2 的实部和虚部, W_{Nr}^k 、 W_{Ni}^k 为输入参数的实部和虚部;

软件操作流程:

- 1) 配置寄存器 MAC_CTL0[8]=1 选择 butterfly 单次模式;
- 2) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#);
- 3) 配置寄存器 MAC_IN0/1/2/3/4/5 配置输入数据;

$$\text{MAC_IN0}=X_{1r}$$

$$\text{MAC_IN1}=X_{1i}$$

$$\text{MAC_IN2}=X_{2r}$$

$$\text{MAC_IN3}=X_{2i}$$

$$\text{MAC_IN4}=W_{Nr}^k$$

$$\text{MAC_IN5}=W_{Ni}^k$$

- 4) 配置寄存器 MAC_CTL1[0]=1 启动 butterfly 运算;
- 5) 等待标志位 MAC_FLG[5]置起, 读取结果 MAC_OUT0/1/2/3;

$$\text{MAC_OUT0}=X'_{1r}$$

$$\text{MAC_OUT1}=X'_{1i}$$

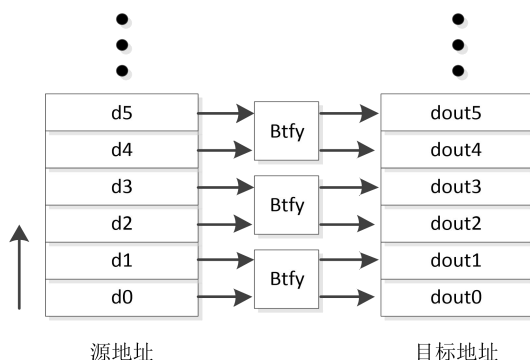
$$\text{MAC_OUT2}=X'_{2r}$$

$$\text{MAC_OUT3}=X'_{2i}$$

如果配置了中断使能信号, 则单次蝶形运算之后, 将产生中断。每次配置寄存器启动 butterfly 操作, 会自动清除标志位。

26.3.7.2 多数据连续转换

该模式是计算 SRAM 中多组数据的蝶形运算, 相邻地址的两个数做一次蝶形运算, 结果写回 SRAM 中。



如上图, 从低地址到高地址, 每次从源数据区域取出两个数据计算依次 butterfly 之后, 写到目标 SRAM 中。

支持源 DMA 地址和目标 DMA 地址相同, 覆盖源数据, 节省 SRAM。

软件配置流程:

- 1) 配置寄存器 MAC_CTL0[9]=1 选择 butterfly DMA 模式;
- 2) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#);
- 3) 配置寄存器 DMA_SRBADR 选择源数据实部 DMA 起始地址;

- 4) 配置寄存器 DMA_SIBADR 选择源数据虚部 DMA 起始地址;
- 5) 配置寄存器 DMA_PRBADR 选择参数表实部 DMA 起始地址;
- 6) 配置寄存器 DMA_PIBADR 选择参数表虚部 DMA 起始地址;
- 7) 配置寄存器 DMA_TRBADR 选择目标数据实部 DMA 起始地址;
- 8) 配置寄存器 DMA_TIBADR 选择目标数据虚部 DMA 起始地址;
- 9) 配置寄存器 DMA_LEN 选择数据点数 (2 的 n 次方);
- 10) 配置寄存器 MAC_CTL1[0]=1 启动计算;
- 11) 等待标志位 MAC_FLG[6], 从 SRAM 中获得计算后的数据;

所有数据计算完成之后, 会产生标志位, 如果配置了中断使能, 则会产生中断。标志位需要软件清除。

26.3.7.3 FFT (fp_fft)

该模式下, 硬件自动把 SRAM 中的指定数据 (浮点数) 做基-2 的 FFT 计算, 结果保存到 SRAM 中, 并且产生完成标志。其中数据长度可配置为 64、128、256、512、1024;

由于 FFT 运算需要用到参数表, 为了加快 FFT 过程, 对于 128/64 点的 FFT, 硬件固化了一张参数表, 从而每次计算直接取表值, 减少了访问 SRAM 的时间。

支持源数据和目标数据地址相同, 做到数据覆盖, 减少 SRAM 开销。

软件配置流程:

- 1) 配置寄存器 MAC_CTL0[10]=1 选择 FFT 模式;
- 2) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#);
- 3) 如果使用内部参数表, 则配置 MAC_CTL0[27]=1; 否则, 表格存放在 SRAM 中, 配置寄存器 DMA_PRBADR 选择表格实部的 DMA 起始地址, 配置寄存器 DMA_PIBADR 选择表格虚部的 DMA 起始地址;
- 4) 配置寄存器 DMA_SRBADR 选择源数据实部 DMA 起始地址;配置寄存器 DMA_SIBADR 选择源数据虚部 DMA 起始地址;
- 5) 配置寄存器 DMA_TRBADR 选择目标数据实部起始 DMA 地址; 配置寄存器 DMA_TIBADR 选择目标数据虚部起始 DMA 地址;
- 6) 配置寄存器 DMA_LEN 选择数据长度;
- 7) 等待完成标志位 MAC_FLG[7]=1, 表示 FFT 完成;

26.3.7.4 倒序搬运数据 (bitrev_move)

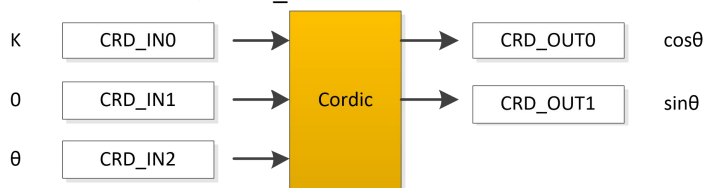
该模式下, 硬件自动把 SRAM 中一段原始数据根据 bit reverse 算法, 重新排序, 并且写到 SRAM 中指定位置。

数据长度支持 4、8、16、32、64、128、256、512、1024。并且每个数据占用 4Byte 的 SRAM 空间, 在 SRAM 中按 Word 对齐存放。

软件配置流程:

- 1) 配置寄存器 MAC_CTL0[11]=1 选择 bit reverse 模式;
- 2) 配置寄存器 DMA_SRBADR 选择源数据 DMA 起始地址;
- 3) 配置寄存器 DMA_TRBADR 选择目标 DMA 起始地址;
- 4) 配置寄存器 DMA_LEN 选择数据长度;
- 5) 等待完成标志 MAC_FLG[8]=1, 表示完成;

26.3.8 正弦余弦计算 (sin_cos)



正余弦计算

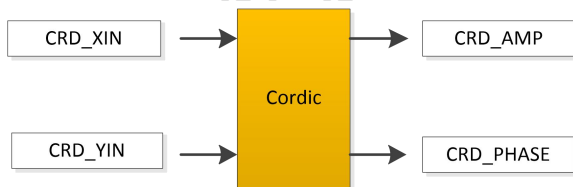
该模式支持输入角度，硬件计算 sin 和 cos 函数。

采用 cordic 旋转模式，配置 $X_0 = \frac{1}{P} = K = 0.60725$, $Y_0 = 0$, $Z_0 = \theta$ ，输出结果为 $[\cos\theta, \sin\theta, 0]$

软件配置流程：

- 1) 配置寄存器 CRD_ANGLE 输入角度（寄存器值 = $\text{radian} \times (2^{32}) / (2 \times \pi)$ ）；
- 2) 配置寄存器 CRD_CTL[1]=1 启动运算；
- 3) 等待完成标志 CRD_FLG[1]，读取寄存器 CRD_COSINE/CRD_SINE 得到计算结果；

26.3.9 均方根和反正切 (fp_sqrt/ fp_atan)



均方根

该模式计算输入数据 CRD_XIN 和 CRD_YIN 的均方根。

采用 cordic 向量模式，配置 $X_0 = X$, $Y_0 = Y$, $Z_0 = 0$ ，输出为 $\left[P \sqrt{X_i^2 + Y_i^2}, 0, \arctan\left(\frac{Y_i}{X_i}\right) \right]$

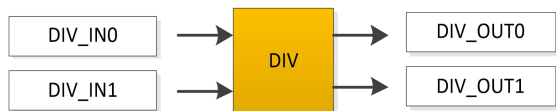
软件配置流程：

- 1) 配置寄存器 CRD_XIN=X；
- 2) 配置寄存器 CRD_YIN=Y；
- 3) 配置寄存器 CRD_CTL[0]=1 启动运算；
- 4) 等待完成标志 CRD_FLG[0]，读取寄存器 CRD_AMP/CRD_PHASE 得到计算结果；

$$\text{CRD_AMP} = \sqrt{X_0^2 + Y_0^2} / 4 \times P, \text{CRD_PHASE} = \arctan\left(\frac{Y_0}{X_0}\right) \times 2^{31} / \pi$$

如果配置了中断使能信号，则计算完成之后，产生中断。

26.3.10 浮点数除法器 (fp_div)



除法

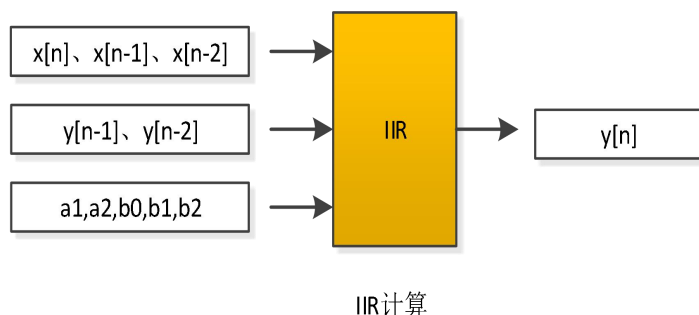
该模式是一个浮点数的除法运算，输入两个浮点数，做除法之后输出商和余数。

软件配置流程:

- 1) 配置被除数到寄存器 DIV_IN0;
- 2) 配置除数到寄存器 DIV_IN1;
- 3) 配置寄存器 MAC_CTL1[2]=1 启动除法运算;
- 4) 等待完成标志位 MAC_FLG[4], 读取寄存器 DIV_OUT0 获得商;

26.3.11

IIR 滤波器



该模式是 IIR 滤波器计算，输入浮点数据 a_1 、 a_2 、 b_0 、 b_1 、 b_2 、 $y[n-2]$ 、 $y[n-1]$ 、 $x[n]$ 、 $x[n-1]$ 、 $x[n-2]$ 后，做 IIR 运算后输出 $y[n]$ 。

26.3.11.1 单数据转换

- 1) 配置寄存器 MAC_CTL0[4:0]=13, (IIR_ONCE_MODE) 选择 IIR 单次模式;
- 2) 配置寄存器 MAC_CTL0[9:7]选择浮点数的 [rounding mode](#);
- 3) 配置寄存器 MAC_IN01/2/3/4/5, DIV_IN0/1, MAC_OUT0/1 配置输入数据;

(MAC_IN0 必须放在最后配置, 写完 MAC_IN0, 硬件会自动启动 IIR 运算)

MAC_IN1= $x[n-1]$

MAC_IN2= $x[n-2]$

MAC_IN3= b_0

MAC_IN4= b_1

MAC_IN5= b_2

DIV_IN0= a_1

DIV_IN1= a_2

MAC_OUT0= $y[n]$

MAC_OUT1= $y[n-1]$

MAC_IN0= $x[n]$

- 4) 等待标志位 MAC_FLG[10] (IIR_ONCE_DONE) 置起, 读取结果 MAC_OUT0;

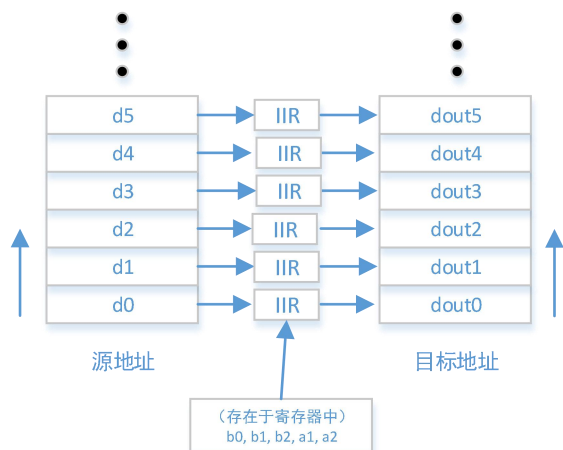
MAC_OUT0= $y[n]$

如果配置了中断使能信号(FFT_IE[10].IIR_ONCE_IE=1), 则 IIR 运算之后, 将产生中断。需要软件清除标志位。

(因为 IIR 用来 DIV_IN0/1 配置参数 a_1, a_2 , IIR 运算过程中不能改变 DIV_IN0/1。即 IIR 和除法不能同时使用)

26.3.11.2 多数据连续转换

该模式是计算 SRAM 中多组数据的 IIR 运算，相邻地址的 3 个数做一次 IIR 运算，结果写回 SRAM 中。



如上图，从低地址到高地址，每次从源数据区域取出 3 个数据进行 IIR 计算之后，写到目标 SRAM 中。

软件配置流程:

- 1) 配置寄存器 MAC_CTL0[4:0]=14, 选择 IIR DMA 模式;
- 2) 配置寄存器 MAC_CTL0[9:7]选择浮点数的 [rounding mode](#);
- 3) 配置寄存器 MAC_IN1/2/3/4/5, DIV_IN0/1, MAC_OUT0/1 配置输入数据:

$$MAC_IN1 = x[n-1]$$

$$MAC_IN2 = x[n-2]$$

$$MAC_IN3 = b_0$$

$$MAC_IN4 = b_1$$

$$MAC_IN5 = b_2$$

$$DIV_IN0 = a_1$$

$$DIV_IN1 = a_2$$

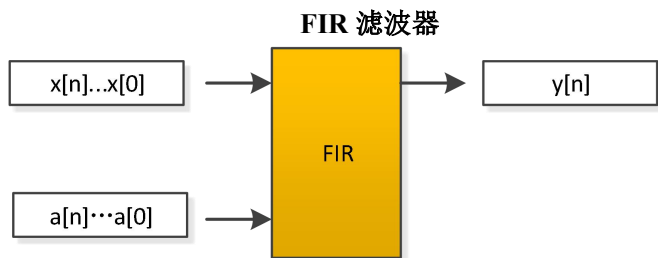
$$MAC_OUT0 = y[n]$$

$$MAC_OUT1 = y[n-1]$$

- 4) 配置好数据 $x[n]$ 起始地址 DMA_SRBADR、数据 $y[n]$ 目标地址 DMA_TRBADR, 数据长度 DMA_LEN, DMA 读 SRAM 步进 MAC_CTL1[3:1];
- 5) 配置寄存器 MAC_CTL2[0]=1 (DMA_EN) 启动运算; (建议先清零再写 1)
- 6) 等待标志位 MAC_FLG[11] (IIR_DMA_DONE) 置起, 读取 SRAM 的目标地址得到 $y[n]$ 。

如果配置了中断使能信号(FFT_IE[11].IIR_DMA_IE=1), 则 IIR DMA 运算之后, 将产生中断。标志位需要软件清除。

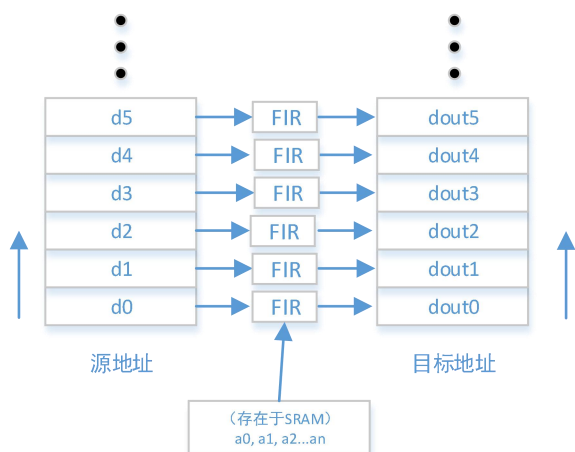
26.3.12



FIR 运算

该模式是 FIR 滤波器计算，输入浮点数据 $x[n] \dots x[0]$ 和参数 $a[n] \dots a[0]$ ，做 FIR 运算后输出 $y[n]$ 。

该模式是计算 SRAM 中多个数据的 FIR 运算，相邻地址的 N 个数做一次 FIR 运算，结果写回 SRAM 中。



		数据源地址L	$x[L]$	数据目的地址L-n	$y[L-n]$
			$x[L-1]$		$y[L-n-1]$
			$x[L-2]$		$y[L-n-2]$
			\vdots		\vdots
			$x[L-n]$		\vdots
			\vdots		$y[2]$
			$x[n+1]$		$y[1]$
			$x[n]$	数据目的地址0	$y[0]$
			$x[n-1]$		
			\vdots		
			$x[2]$		
			$x[1]$		
			$x[0]$		
参数源地址n	$a[n]$				
	$a[n-1]$				
	$a[n-2]$				
	\vdots				
	$a[2]$				
	$a[1]$				
参数源地址0	$a[0]$	数据源地址0			

$x[0] \sim x[n-1]$ 放初始值， $x[n]$ 放第一个要转换的数。

首次的时候， $x[0] \sim x[n]$ 需要写0；

其他次数时， $x[0] \sim x[n-1]$ 配成上一次周期的最后n点的值。

n阶：由寄存器FIR_ORDER[7:0]决定

数据长度L：由DMA_LEN[14:0]寄存器决定

实际转换出来的 $y[n]$ 个数 = $L+1-(n+1) = L-n$

如上图，从低地址到高地址，每次从源数据区域取出 3 个数据进行 FIR 计算之后，写到目标 SRAM 中。

软件配置流程：

- 1) 配置寄存器 MAC_CTL0[4:0]=15，选择 FIR DMA 模式；
- 2) 配置寄存器 MAC_CTL0[9:7]选择浮点数的 [rounding mode](#)；

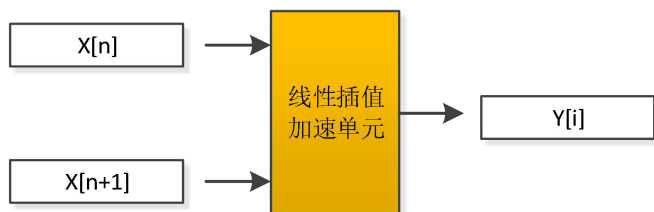
- 3) 配置好 **FIR 的阶数**，配置好 DMA 数据 $x[n]$ 起始地址 DMA_SRBADR、参数 a 起始地址 DMA_PRBADR、数据 $y[n]$ 目标地址 DMA_TRBADR，数据长度 DMA_LEN，DMA 读 SRAM 步进 MAC_CTL1[3:1]；
- 4) 配置 MAC_CTL2[0]=1 (DMA_EN) 启动运算；(建议先清零再写 1)
- 5) 等待标志位 MAC_FLG[11] (FIR_DMA_DONE) 置起，读取 SRAM 的目标地址得到 $y[n]$ 。

如果配置了中断使能信号(FFT_IE[11].FIR_DMA_IE=1)，则 FIR DMA 运算之后，将产生中断。需要软件清除标志位。

(不允许配置 dma_len 小于 fir_order, dma_len >= fir_order)

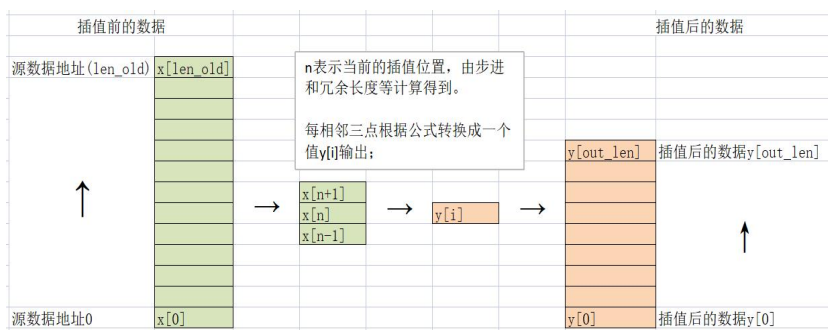
26.3.13

线性插值



该模式是实现线性插值加速算法。每相邻两个点根据公式转换出一个 $y[i]$ 值输出。

该模式是计算 SRAM 中多个数据的线性插值运算，相邻地址的 2 个数做一次线性插值运算，结果写回 SRAM 中。



如上图，从低地址到高地址，每次从源数据区域取出 3 个数据进行线性插值计算之后，写到目标 SRAM 中。

软件配置流程：

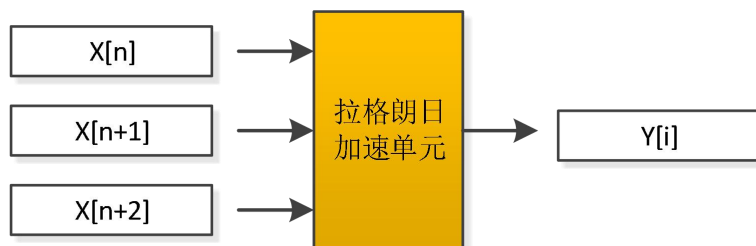
- 1) 软件计算各种参数，包括实际可用输入数据长度 len、插值后的数据长度 out_len、当前插值位置 t 等；
- 2) 配置寄存器 MAC_CTL0[4:0]=16 (LINE_INTP)，选择线性插值模式；
- 3) 配置寄存器 MAC_CTL0[9:7]选择浮点数的 [rounding mode](#)；
- 4) 配置好数据 $x[n]$ 起始地址 DMA_SRBADR、数据 $y[i]$ 目标地址 DMA_TRBADR；
- 5) 配置寄存器 MAC_IN5[31:0]；(上周期的最后一个数据)
- 6) 配置寄存器 MAC_CTL1[0]，选择是否使用上周期最后一个数据；
- 7) 配置寄存器 INTP_LEN[12:0]，选择实际可用输入数据长度；
- 8) 配置寄存器 DMA_LEN[10:0]，选择插值后的数据长度；
- 9) 配置寄存器 INTP_LOC[31:0]，选择当前插值位置；
- 10) 配置寄存器 INTP_STEP[31:0]，选择步进；
- 11) 配置 MAC_CTL2[0]=1 (DMA_EN) 启动运算；(建议先清零再写 1)
- 12) 等待标志位 FFT_FLG[13] (LINE_INTP_DONE) 置起，读取 SRAM 的目标地址得到 $y[n]$ 。

- 13) 软件读取 $INTP_LOC[31:0]$, 令 $t=INTP_LOC[31:0]$, 软件计算本次插值计算完成后冗余长度 $red_len=len-t+step$
- 14) 软件根据具体情况, 选择是否要保存最后一个数据, 如需保存, 将值写到配置寄存器 $MAC_IN5[31:0]$; (上周期的最后一个数据)

如果配置了中断使能信号($FFT_IE[13].LINE_INTP_IE=1$), 则线性插值运算之后, 将产生中断。需要软件清除标志位。

26.3.14

拉格朗日插值



该模式是实现拉格朗日插值加速算法。每相邻 3 个点根据公式转换出一个 $y[i]$ 值输出。

该模式是计算 SRAM 中多个数据的拉格朗日插值运算, 相邻地址的 3 个数做一次拉格朗日插值运算, 结果写回 SRAM 中。



如上图, 从低地址到高地址, 每次从源数据区域取出 3 个数据进行拉格朗日计算之后, 写到目标 SRAM 中。

软件配置流程:

- 1) 软件计算各种参数, 包括实际可用输入数据长度 len 、插值后的数据长度 out_len 、当前插值位置 t 等;
- 2) 配置寄存器 $MAC_CTL0[4:0]=17$ (LAGR_INTPT), 选择拉格朗日插值模式;
- 3) 配置寄存器 $MAC_CTL0[9:7]$ 选择浮点数的 [rounding mode](#);
- 4) 配置好 DMA 数据 $x[n]$ 起始地址 DMA_SRBADR 、数据 $y[i]$ 目标地址 DMA_TRBADR ;
- 5) 配置寄存器 $MAC_IN5[31:0]$; (上周期的最后一个数据)
- 6) 配置寄存器 $MAC_CTL1[0]$, 选择是否使用上周期最后一个数据;
- 7) 配置寄存器 $INTP_LEN[12:0]$, 选择实际可用输入数据长度;
- 8) 配置寄存器 $DMA_LEN[9:0]$, 选择插值后的数据长度;
- 9) 配置寄存器 $INTP_LOC[31:0]$, 选择当前插值位置;
- 10) 配置寄存器 $INTP_STEP[31:0]$, 选择步进;
- 11) 配置 $MAC_CTL2[0]=1$ (DMA_EN) 启动运算; (建议先清零再写 1)
- 12) 等待标志位 $MAC_FLG[14]$ (LAGRINTPT_DONE) 置起, 读取 SRAM 的目标地址得到 $y[n]$;

- 13) 软件读取 INTP_LOC[31:0], 令 $t=INTP_LOC[31:0]$, 软件计算本次插值计算完成后冗余长度 $red_len=len-t+step$
- 14) 软件根据具体情况, 选择是否要保存最后一个数据, 如需保存, 将值写到配置寄存器 MAC_IN5[31:0]; (上周期的最后一个数据)

如果配置了中断使能信号(FFT_IE[14].LAGR_INTP_IE=1), 则拉格朗日插值运算之后, 将产生中断。需要软件清除标志位。

26.4 实现说明

26.4.1 整数转浮点数

由于输入的整数位宽为 32 位, 转换之后的浮点数指数范围很小, 所以不会有溢出, 并且做归一化的时候最高支持 32 次方, 不需要做溢出处理。

归一化, 就是把指数部分减去 n , 小数部分不变。

26.4.2 浮点数转整数

需要先把浮点数放大, 让整数位足够大, 再进行转换, 才能保证更多有效位。

放大 2^n 就是把指数部分加上 n , 小数部分不变。

26.4.3 FFT 说明

软件只需提前把原始数据存放到 SRAM 中, 并且配置好 DMA 起始地址、目标地址, 数据长度。硬件自动取数运算, 得到 butterfly 的结果。如果数据长度为 64, 则每一阶有 $64/2=32$ 次蝶形运算, 一共 $\log_2 64 = 6$ 阶, 所以硬件会自动算完 $32*6$ 次蝶形运算, 并且把结果保存到 SRAM 指定位置, 产生完成标志或中断。

硬件每算一次蝶形运算, 需要读 4 次 SRAM 获得原始数据 (第一轮原始输入数据虚部为 0), 写 4 次 SRAM 保存该次蝶形运算结果, (如果参数表格存放 SRAM, 还需要读 2 次 SRAM 获取参数)。所以完成一段指定长度数据需要访问 SRAM 周期数为:

$$k_0 = \frac{n}{2} * \log_2 n * (4 + 4) - \frac{n}{2} * 2, \text{ 参数表硬件固化};$$

$$k_0 = \frac{n}{2} * \log_2 n * (4 + 4 + 2) - \frac{n}{2} * 2, \text{ 参数表存放 SRAM};$$

点数	读写 SRAM 周期	
	硬件表	SRAM 表
64	1472	1856
128	3456	4352
256	7936	9984
512	17920	22528
1024	39936	50176

完成 FFT 需要读写 SRAM 周期数

硬件读写 SRAM 加上 butterfly 运算, 完成一次 butterfly 最快时间为 10 个时钟周期 (参数表硬件固化) 或 12 个时钟周期 (参数表存放 SRAM)。

时钟周期	1	2	3	4	5	6	7	8	9	10	11	12
硬件行为	读	读	读	读	读	读						
		寄存	寄存	寄存	寄存	寄存	寄存					
								乘	乘	乘	乘	
								加	加	加	加	
								加	加	加	加	
									写	写	写	写

所以总时间为：

$$k_0 = \frac{n}{2} * \log_2 n * 10, \text{ 参数表硬件固化;}$$

$$k_0 = \frac{n}{2} * \log_2 n * 12, \text{ 参数表存放 SRAM;}$$

点数	FFT 时钟周期		FFT 时间 (us)	
	硬件表	SRAM 表	16M 时钟	
64	1920	2304	120	144
128	4480	5376	280	336
256	10240	12288	640	768
512	23040	27648	1440	1728
1024	51200	61440	3200	3840

所以完成 FFT 最快时间为上表计算时间加上配置 DMA 等模式寄存器的时间。上表中的时间是基于 SRAM 空闲，只有 FFT 模块访问时算出来的，实际情况，可能有其他模式抢占 SRAM，所以 FFT 时间会加长。

26.4.4 Cordic 说明

arctan 参数表，从 arctan(1),45 度，arctan(1/2)。。。到 arctan(1/2^n)。

所有参数需要按照比例放大，把 arctan(1)=2^n 做为标准，其他放大相同的倍数 a。其中放大倍数 a 是一个定点化过程，会引入误差。

每次迭代过程 x 和 y 做移位处理（向下逼近），会引入误差。如果改成四舍五入误差会减小。

迭代次数根据放大后的参数表决定，如果参数表值为 0，表示迭代结束。

26.4.4.1 向量模式

确认位宽：

1. 首先，因为原始数据输入 x 和 y 的位宽是 32bit；
2. 该模式下，输入 z=0，所以 arctan 的放大倍数跟输入没有关系，可以任意取。此时只需要根据所需的输出角度精度来决定 arctan 的放大倍数 n。
3. 根据 arctan 的放大倍数，决定迭代次数。
4. 最终输出的角度精度，跟 arctan 的放大倍数 n 相关。输出的角度结果必须除以 n，才是实际角度（因为这个角度是累加出来的， $z(i+1)=z(i)-\arctan(1/2^i)$ ；左右两边同时放大 n）。如果输出结果是取高位输出也是可以的（运算过程比较准确，输出结果比较简短）。

向量模式下，输入 x 和 y 可以等比例放大，计算出来的精度比较高，所以，对于比较小的 x 和 y ，只需先放大，再代入 cordic 运算就能得到更高精度的结果。

首先需要做预处理，把所有角度移到第 1 象限内，如果是第 2 象限，移到第一象限后，初始角度 $z=90$ 度。第 3 象限，初始角度 $z=180$ 度；第 4 象限，初始角度 $z=270$ 度。

```
if(x<0){
    x=-x;
    y=-y;
    z=z-(1~3)pi/2*n; //n 是放大倍数
}
```

26.4.4.2 旋转模式

确认位宽：

- 1 模块输入为弧度，范围是 $-\pi$ ： π 。假设输入弧度用 n 位数表示，所以输入弧度放大倍数为 $p0=(2^n)/2\pi$ 。
- 2 根据公式， $z(i+1)=z(i)-\arctan(1/2^i)$ ，决定了做加减法运算时， \arctan 放大倍数与输入角度 z 放大倍数相同。 $\text{atan_t}(n)=p0*\text{atan}(n)$ ，其他参数值按相同比例放大 $p0$ 倍。
- 3 正常情况下， $\text{atan_t}(1)=p0*\text{atan}(n)=2^{(n-3)}$ ，所以参数表位宽为 $n-3$ 。如果为了减小定点化的误差，可以把参数表的位数定的更长，计算的时候，只需要输入数据右移放大相同倍数，参与加减。

比如 atan 定点化时位宽为 $n-3+5$ ，
则运算前放大

```
x_in = {x_in,5'h0};
y_in = {y_in,5'h0};
z_in = {z_in,5'h0};
```

迭代后输出取高位。

- 4 迭代之后输出结果也是放大了 $p0$ 倍，可以根据输出数据的位宽，调整输出的放大倍数。

旋转模式实现过程：

1. 输入 $X_0 = k$, $Y_0=0$, $Z_0 = \theta$ ；输入角度范围是 $-\pi \sim \pi$ ；
2. 首先对角度做预处理，翻转到 0 ： $\pi/2$ 范围内；
3. 按照公式迭代得到结果；
4. 结果乘以增益系数 K ；
5. 输入 x 、 y 的位宽与迭代次数相关。因为每次移 1bit。

26.5 寄存器

26.5.1 寄存器列表

基地址	0x40058000				
偏移地址	名称	R/W	有效字长	复位值	功能描述

0x0	MAC_CTL0	R/W	4	0x0	MAC 单元控制寄存器 0
0x4	MAC_CTL1	R/W	4	0x0	MAC 单元控制寄存器 1
0x8	MAC_CTL2	R/W	4	0x0	MAC 单元控制寄存器 2
0xC	MAC_IN0	R/W	4	0x0	MAC 单元数据输入寄存器 0
0x10	MAC_IN1	R/W	4	0x0	MAC 单元数据输入寄存器 1
0x14	MAC_IN2	R/W	4	0x0	MAC 单元数据输入寄存器 2
0x18	MAC_IN3	R/W	4	0x0	MAC 单元数据输入寄存器 3
0x1C	MAC_IN4	R/W	4	0x0	MAC 单元数据输入寄存器 4
0x20	MAC_IN5	R/W	4	0x0	MAC 单元数据输入寄存器 5
0x24	MAC_OUT0	R/W	4	0x0	MAC 单元数据输出寄存器 0
0x28	MAC_OUT1	R/W	4	0x0	MAC 单元数据输出寄存器 1
0x2C	MAC_OUT2	R/W	4	0x0	MAC 单元数据输出寄存器 2
0x30	MAC_OUT3	R/W	4	0x0	MAC 单元数据输出寄存器 3
0x34	DIV_IN0	R/W	4	0x0	除法单元数据输入 0
0x38	DIV_IN1	R/W	4	0x0	除法单元数据输入 1
0x3C	DIV_OUT0	R/W	4	0x0	除法单元数据输出
0x40	DMA_SRBADR	R/W	4	0x0	DMA 源数据 FFT 实部起始地址
0x44	DMA_SIBADR	R/W	4	0x0	DMA 源数据 FFT 虚部起始地址
0x48	DMA_PRBADR	R/W	4	0x0	DMA 旋转因子表格实部起始地址
0x4C	DMA_PIBADR	R/W	4	0x0	DMA 旋转因子表格虚部起始地址
0x50	DMA_TRBADR	R/W	4	0x0	DMA 目标数据 FFT 实部起始地址
0x54	DMA_TIBADR	R/W	4	0x0	DMA 目标数据 FFT 虚部起始地址
0x58	DMA_LEN	R/W	2	0x0	DMA 长度配置
0x5C	DSP_IE	R/W	2	0x0	FFT 中断使能寄存器
0x60	DSP_FLG	R/W	2	0x0	FFT 标志寄存器
0x64	ALU_STA0	R/W	4	0x0	ALU 状态寄存器 0
0x68	ALU_STA1	R/W	2	0x0	ALU 状态寄存器 1
0x6C	CRD_CTL	WO	1	0x0	Cordic 控制寄存器
0x70	CRD_XIN	R/W	4	0x0	Cordic 向量模式数据 X 输入
0x74	CRD_YIN	R/W	4	0x0	Cordic 向量模式数据 Y 输入
0x78	CRD_AMP	R	4	0x0	Cordic 向量模式幅值输出
0x7C	CRD_PHASE	R	4	0x0	Cordic 向量模式角度输出
0x80	CRD_ANGLE	R	4	0x0	Cordic 旋转模式角度输入
0x84	CRD_COS	R	4	0x0	Cordic 旋转模式余弦输出
0x88	CRD_SIN	R	4	0x0	Cordic 旋转模式正弦输出
0x8C	CRD_IE	R/W	1	0x0	Cordic 中断使能
0x90	CRD_FLG	R/W	1	0x0	Cordic 完成标志寄存器

0x94	INTP_LEN	R/W	2	0x0	插值输入数据长度寄存器
0x98	INTP_LOC	R/W	4	0x0	当前插值位置寄存器
0x9C	INTP_STEP	R/W	4	0x0	插值算法步进寄存器

26.5.2 MAC_CTL0 (0x0)

偏移地址：00H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31: 23	保留	保留	R	0
22:21	I2F_PRE_EN	整数转浮点数预处理， 2' b00: 4Byte 都有效 2' b01: 表示每 4Byte 中，最高 Byte 数据无效。有效数据仅为 3Byte； 2' b10: 表示每 4Byte 中，最高 2 Byte 数据无效。有效数据仅为 2 Byte； 2' b11: 表示每 4Byte 中，最高 3 Byte 数据无效。有效数据仅为 1 Byte；	R/W	0
20	FFT_TB_EN	FFT 使用内部参数表，仅 64 点和 128 点有效。1=使能	R/W	0
19:15	F2I_MUL	浮点数转整数之前，对浮点数进行放大 0 表示不放大 n 表示放大 2^n	R/W	0
14:10	I2F_DIV	整数转浮点数之后，归一化阶数配置 0 表示不做归一化处理。 n 表示除以 2^n 。	R/W	0
9:7	ROUND_MODE	浮点数运算过程中，舍入模式配置： 当配置 rnd=000，为 nearest 舍入； 当配置 rnd=001，为 zero 舍入； 当配置 rnd=010，为 $+\infty$ 舍入； 当配置 rnd=011，为 $-\infty$ 舍入； 当配置 rnd=100，为 up 向上进位； 当配置 rnd=101，为 away 直接舍弃； 其他：保留。	R/W	0
6	MUL_OUT_FM	浮点数乘法运算，输出数据格式 0=浮点数 1=整数	R/W	0
5	MUL_IN_FM	浮点数乘法运算，输入数据格式 0=浮点数 1=整数	R/W	0
4:0	MODE_SEL	5' d0: 所有模式都不使能； 5' d1: 整数转浮点数，单次使能 5' d2: 整数转浮点数，DMA 使能 5' d3: 浮点数转整数，单次使能 5' d4: 浮点数转整数，DMA 使能 5' d5: 浮点数乘法，模式使能	R/W	0

		5' d6: 浮点数加法, 模式使能 5' d7: 浮点数减法, 模式使能 5' d8: 浮点数乘加运算, 模式使能 ($y=ax+b$) 5' d9: 蝶形运算, 单次使能 5' d10: 蝶形运算, DMA 使能 5' d11: FFT 模式使能 5' d12: bit reverse 模式使能 5' d13: 二阶 IIR 单次运算, 模式使能 5' d14: 二阶 IIR DMA 运算, 模式使能 5' d15: N 阶 FIR DMA 运算, 模式使能 5' d16: 线性插值运算, 模式使能 5' d17: 拉格朗日插值运算, 模式使能 5' d18~31: 所有模式都不使能;		
--	--	--	--	--

26.5.3

MAC_CTL1 (0x04)

偏移地址: 04H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:12	Reserved	保留。	R	0
11:4	FIR_ORDER	FIR 阶数配置; 0~3: 3 阶; 4: 4 阶; 5: 5 阶; ... 254: 254 阶; 255: 255 阶;	R/W	0
3:1	DMA_STEP	DMA_读 SRAM 的步进, 只对 IIR, FIR DMA 模式起效。 0: 步进 1; 1: 步进 2; ... 7: 步进 8;	R/W	0
0	INTP_LAST_EN	插值模式使用上周期的最后一个数据, 使能信号	R/W	0

26.5.4

MAC_CTL2 (0x08)

偏移地址: 08H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:3	Reserved	保留。	R	0
2	DIV_KICK	除法器启动使能信号	WO	0
1	BTFY_ONCE_KICK	单次蝶形运算模式启动信号, 高电平有效	WO	0
0	DMA_EN	DMA 使能信号, 所有与 DMA 有关的操作通过配置该位启动, DMA 完成后该位自动清 0	R/W	0

注意: Bit1 和 Bit2 是只写位

使用 MAC 单元可实现的运算指令及具体配置参见 21.3 章节。

26.5.5 MAC_IN0 (0x0C)

偏移地址：08H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	MAC_IN0	乘加单元数据输入端	R/W	0

26.5.6 MAC_IN1 (0x10)

偏移地址：0CH；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	MAC_IN1	乘加单元数据输入端	R/W	0

26.5.7 MAC_IN2 (0x14)

偏移地址：10H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	MAC_IN2	乘加单元数据输入端	R/W	0

26.5.8 MAC_IN3 (0x18)

偏移地址：14H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	MAC_IN3	乘加单元数据输入端	R/W	0

26.5.9 MAC_IN4 (0x1C)

偏移地址：18H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	MAC_IN4	乘加单元数据输入端	R/W	0

26.5.10 MAC_IN5 (0x20)

偏移地址：1CH；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	MAC_IN5	乘加单元数据输入端	R/W	0

26.5.11 MAC_OUT0 (0x24)

偏移地址：20H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	MAC_OUT0	乘加单元数据输出端	R/W	0

26.5.12 MAC_OUT1 (0x28)

偏移地址：24H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	MAC_OUT1	乘加单元数据输出端	R/W	0

26.5.13 MAC_OUT2 (0x2C)

偏移地址：28H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	MAC_OUT2	乘加单元数据输出端	R/W	0

26.5.14 MAC_OUT3 (0x30)

偏移地址：2CH；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	MAC_OUT3	乘加单元数据输出端	R/W	0

26.5.15 DIV_IN0 (0x34)

偏移地址：30H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	DIV_IN0	除法单元数据输入端（被除数）	R/W	0

26.5.16 DIV_IN1 (0x38)

偏移地址：34H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	DIV_IN1	除法单元数据输入端（除数）	R/W	0

26.5.17 DIV_OUT0 (0x3C)

偏移地址：38H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	DIV_OUT0	除法单元输出端（商）	R	0

26.5.18 DMA_SRBADR (0x40)

偏移地址：3CH；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:15	Reserved	保留。	R	0
14:0	DMA_SRBADR	DMA 源数据起始地址：（Word 地址） FFT/BTFY_DMA 模式：DMA 源数据 FFT 实部起始地址 I2F_DMA/F2I_DMA/BIT_REV 模式：DMA 源数据起始地址	R/W	0

26.5.19 DMA_SIBADR (0x44)

偏移地址：40H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:15	Reserved	保留。	R	0
14:0	DMA_SIBADR	DMA 源数据起始地址：（Word 地址） FFT/BTFY_DMA 模式：DMA 源数据 FFT 虚部起始地址	R/W	0

26.5.20 DMA_PRBADR (0x48)

偏移地址：44H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:15	Reserved	保留。	R	0
14:0	DMA_PRBADR	DMA 源数据起始地址：（Word 地址） FFT 或 BTFY 模式下：DMA 旋转因子参数表实部起始地址	R/W	0

26.5.21 DMA_PIBADR (0x4C)

偏移地址：48H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:15	Reserved	保留。	R	0
14:0	DMA_PIBADR	DMA 源数据起始地址：（Word 地址） FFT 或 BTFY 模式下：DMA 旋转因子参数表虚部起始地址	R/W	0

26.5.22 DMA_TRBADR (0x50)

偏移地址：4CH；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:15	Reserved	保留。	R	0
14:0	DMA_TRBADR	DMA 目标数据起始地址：（Word 地址） FFT/BTFY_DMA 模式：DMA 目标数据 FFT 实部起始地址 I2F_DMA/F2I_DMA/BIT_REV 模式：DMA 源数据目标地址	R/W	0

26.5.23 DMA_TIBADR (0x54)

偏移地址：50H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:15	Reserved	保留。	R	0
14:0	DMA_TIBADR	DMA 目标数据起始地址：（Word 地址） FFT/BTFY_DMA 模式：DMA 目标数据 FFT 虚部起始地址	R/W	0

26.5.24 DMA_LEN (0x58)

偏移地址：54H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:10	Reserved	保留。	R	0
9:0	DMA_LEN	点数配置 n, 长度= (n+1) Word I2F_DMA/F2I_DMA/BTFY_DMA 模式: 支持任意配置 bit reverse 模式: 只支持点数为 4、8、16、32、64、128、256、512、1024 FFT 模式: 只支持 64、128、256、512、1024 点	R/W	0

26.5.25

DSP_IE (0x5C)

偏移地址: 5CH; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:15	Reserved	保留。	R	0
14	LAGR_DMA_IE	=0, 不使能 =1, 拉格朗日插值完成中断使能	R/W	0
13	LINE_DMA_IE	=0, 不使能 =1, 线性插值完成中断使能	R/W	0
12	FIR_DMA_IE	=0, 不使能 =1, FIR DMA 运算完成中断使能	R/W	0
11	IIR_DMA_IE	=0, 不使能 =1, IIR DMA 运算完成中断使能	R/W	0
10	IIR_ONCE_IE	=0, 不使能 =1, 单次 IIR 运算完成中断使能	R/W	0
9	MULT_IE	=0, 不使能 =1, 浮点数乘法中断使能	R/W	0
8	BITREV_IE	=0, 不使能 =1, BIT REVERSE 模式中断使能	R/W	0
7	FFT_IE	=0, 不使能 =1, FFT 模式中断使能	R/W	0
6	BTFY_DMA_IE	=0, 不使能 =1, 蝶形运算 DMA 中断使能	R/W	0
5	BTFY_ONCE_IE	=0, 不使能 =1, 单次蝶形运算中断使能	R/W	0
4	F2I_DMA_IE	=0, 不使能 =1, 浮点数转整数 DMA 中断使能	R/W	0
3	I2F_DMA_IE	=0, 不使能 =1, 整数转浮点数 DMA 中断使能	R/W	0
2	DIV_IE	=0, 不使能 =1, 除法单元中断使能	R/W	0
1	DMA_IE	=0, 不使能 =1, DMA 中断使能	R/W	0
0	MAC_IE	=0, 不使能 =1, 乘加单元中断使能	R/W	0

26.5.26

DSP_FLG (0x60)

偏移地址: 60H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:15	Reserved	保留。	R	0
14	LAGR_DMA_DONE	拉格朗日插值完成中断标志 写 1 清 0	R/WC	0
13	LINE_DMA_DONE	线性插值完成中断标志 写 1 清 0	R/WC	0
12	FIR_DMA_DONE	FIR DMA 运算完成中断标志 写 1 清 0	R/WC	0
11	IIR_DMA_DONE	IIR DMA 运算完成中断标志 写 1 清 0	R/WC	0
10	IIR_ONCE_DONE	单次 IIR 运算完成中断标志 写 1 清 0	R/WC	0
9	MULT_DONE	浮点数乘法完成中断标志 写 1 清 0	R/WC	0
8	BITREV_DONE	BIT REVERSE 模式中断标志 写 1 清 0	R/WC	0
7	FFT_DONE	FFT 模式中断标志 写 1 清 0	R/WC	0
6	BTFY_DMA_DONE	蝶形运算 DMA 中断标志 写 1 清 0	R/WC	0
5	BTFY_ONCE_DONE	单次蝶形运算中断标志 写 1 清 0	R/WC	0
4	F2I_DMA_DONE	浮点数转整数 DMA 中断标志 写 1 清 0	R/WC	0
3	I2F_DMA_DONE	整数转浮点数 DMA 中断标志 写 1 清 0	R/WC	0
2	DIV_DONE	除法单元中断标志 写 1 清 0	R/WC	0
1	DMA_DONE	DMA 完成信号中断标志 写 1 清 0	R/WC	0
0	MAC_DONE	乘加单元计算 (MUL 或者 BTFY_ONCE) 完成, 写 “1” 清除该位	R/WC	0

26.5.27 ALU_STA0 (0x64)

偏移地址: 60H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:24	F2I_STATUS	浮点数转整数单元运算状态标志 定义同“乘加单元 0 运算状态标志”	R	1
23:16	I2F_STATUS	整数转浮点数单元运算状态标志 定义同“乘加单元 0 运算状态标志”	R	1
15:8	ADDSUB1_STATUS	乘加单元 1 运算状态标志	R	1

		定义同“乘加单元 0 运算状态标志”		
7:0	ADDSUB0_STATUS	乘加单元 0 运算状态标志 bit0: 整数或浮点输出为零。 bit1: 浮点输出为无穷大。 bit2: 浮点运算无效。当其中一个输入为 NaN 时, 它也被设置为 1。 bit3: 浮点数小于最小标准化数 bit4: 浮点数大于最大标准化数 bit5: 整数或浮点输出不等于无限精确的结果。 bit6: 舍入后的整数结果的大小大于具有相同符号的最大可表示的两个整数。 bit7: 保留。	R	1

26.5.28 ALU_STA1 (0x68)

偏移地址: 64H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:16	Reserved	保留。	R	0
15:8	DIV_STATUS	除法单元运算状态, 定义同“乘加单元 0 运算状态标志”	R	x
7:0	MUL_STATUS	乘法单元运算状态, 定义同“乘加单元 0 运算状态标志”	R	1

26.5.29 CRD_CTL (0x6C)

偏移地址: 68H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:1	Reserved	保留。	R	0
1	CRD_ROT_KICK	cordic 旋转模式启动信号	WO	0
0	CRD_VEC_KICK	cordic 向量模式启动信号	WO	0

这个寄存器没有做读功能, 只写

26.5.30 CRD_XIN (0x70)

偏移地址: 6CH; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:0	X_IN	Cordic 向量模式数据 X 输入端	R/W	0

26.5.31 CRD_YIN (0x74)

偏移地址: 70H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:0	Y_IN	Cordic 向量模式数据 Y 输入端	R/W	0

26.5.32 CRD_AMP (0x78)

偏移地址: 74H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值

31:0	AMP_OUT	Cordic 向量模式数据输出端 $\sqrt{X_0^2 + Y_0^2}/4*P$	R/	0
------	---------	---	----	---

26.5.33 CRD_PHASE (0x7C)

偏移地址: 78H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:0	PHASE_OUT	Cordic 向量模式数据输出端 寄存器值= $\arctan(\frac{Y_0}{X_0}) * 2^{31}/\pi$	R	0

26.5.34 CRD_ANGLE (0x80)

偏移地址: 7CH; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:0	ANGLE_IN	Cordic 旋转模式弧度输入, 范围为 $[-\pi:\pi]$ 输入 $-1*2^{31}$ 表示 $-\pi$, 输入 $1*2^{31}$ 表示 π 寄存器值= $\text{radian} * 2^{31}/\pi$	R/W	0

26.5.35 CRD_COSINE (0x84)

偏移地址: 80H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:0	COSINE_OUT	Cordic 旋转模式余弦输出 寄存器值= $\cos(\text{angle}) * (2^{(31-2)}) / (2*\pi)$ $\text{angle} = \text{radian} * 180/\pi$	R	0

26.5.36 CRD_SINE (0x88)

偏移地址: 84H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:0	SINE_OUT	Cordic 旋转模式正弦输出 寄存器值= $\sin(\text{angle}) * (2^{(31-2)}) / (2*\pi)$	R	0

26.5.37 CRD_IE (0x8C)

偏移地址: 88H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:1	Reserved	保留。	R	0
0	CRD_ROT_IE	Cordic 旋转模式完成中断使能位	R/W	0
0	CRD_VEC_IE	Cordic 向量模式完成中断使能位	R/W	0

26.5.38 CRD_FLG (0x90)

偏移地址: 8CH; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:1	Reserved	保留。	R	0
1	CRD_ROT_DONE	Cordic 旋转模式完成标志 写 1 清 0	R/W	0
0	CRD_VEC_DONE	Cordic 向量模式完成标志 写 1 清 0	R/W	0

26.5.39

INTP_LEN (0x94)

偏移地址：94H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:13	Reserved	保留。	R	0
12:0	INTP_LEN	插值算法实际可用输入数据长度，即 x[n]的个数 点数配置 n, 长度= (n+1) Word。（整数型）	R/W	0

26.5.40

INTP_LOC (0x98)

偏移地址：98H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	INTP_LOC	插值算法当前插值位置（浮点型）	R/W	0

26.5.41

INTP_STEP (0x9C)

偏移地址：9CH；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	INTP_STEP	插值算法的步进（浮点型）	R/W	0

26.6 软件操作流程

具体实现请参照锐能微提供的例程，下列说明仅供学习参考。

26.6.1 完整方案说明：

1. ADC 采样数据整数 d0，存到 SRAM 中，位宽为 24 位二进制补码格式。
2. 数据预处理：对 ADC 数据归一化（二进制补码格式），转化为带符号的浮点格式 d1。表示的范围为 -1~1。
3. FFT 运算：对所有转换完的数 d1 做 FFT 运算的输入，复数格式，其中实部为 d1，虚部为 0。FFT 运算得到 FFT 输出结果。输出为复数，实部 Re 和虚部 Im 都是浮点数表示。
4. 计算谐波含量：

repeat(6){

$$Y_0 = \sqrt{Re0 * Re0 + Im0 * Im0},$$

```

for(n=1;n<42;n++){

$$Y_n = \sqrt{Ren * Ren + Imn * Imn},$$

Yin=k*Yn/Y0,
}
}

```

计算 41 次谐波含量。

电流和电压都需要单独做谐波含量 Yu 和 Yi。

5. 相角计算和谐波功率:

```

repeat(3){
for(n=1;n<42;n++){
Aun=atan(Ren/Imn), //对电压数据
Ain=atan*(Ren/Imn), //对电流数据
A=((Au-Ai)+A0*n)*180/pi,
PFn=cos(A*pi/180)
}
}

```

计算 40 次相角，不需要算基波的。要求精度 0.01 度。

6. 计算谐波功率: $Phn = FSA * Yun * Yin * PFn$ ，算 41 次谐波之后累加。其中 FSA 从外部获取。

26.6.2 操作流程:

1. ADC 采样数据 sdata0 缓存到 SRAM，每个点数据 24bit;
2. 使用 i2f_dma 模式把 SRAM 中的 sdata1 转换成浮点数格式并且做归一化得到 sdata2; 注意此时 SRAM 中每个 Word 地址，只有第 3Byte 为有效数据，所以需要配置 MAC_CTL0[28]=1，不关心最高 Byte。
3. 使用 bitrev_mode 模式，做 bitreverse 把 sdata2 变换成 sdata3;
4. 使用 FFT 计算模块，把 sdata3 做 FFT 计算，得到结果（软件需要提前把 FFT 计算表格写到 SRAM）;
5. 使用硬件 cordic 模块，除法器模块，计算谐波含量以及谐波功率;

27 电能积分单元 D2F（新增）

27.1 概述

芯片集成了 12 路自定义电能积分单元，输入为功率信号 D2FP0~D2FP11，可设置脉冲常数 HFConst4~7，根据输入功率和脉冲常数 HFConst4~7 进行积分，电能存放在 D2FE00~D2FE11 寄存器，其中 D2FP0~D2FP5 支持脉冲输出，可从 D2F_OUT0~D2F_OUT4 这 5 个管脚输出，作为校验脉冲输出，支持电能脉冲中断。输出脉冲周期大于等于 160ms 时，校验脉冲高电平固定保持 80ms，低电平根据周期值进行变化；输出脉冲周期小于 160ms 时，输出等 duty 脉冲。

该模块的基址为：0x4005C000

27.2 主要特点

- 12 路 D2F;
- D2F00~D2F05 共 6 个支持输出脉冲
- 有 5 个脉冲输出口 D2F_OUT0~D2F_OUT4

- 脉冲输出，若输出脉冲周期 $\geq 160\text{ms}$ ，脉冲高电平固定保持 80ms ，低电平根据周期值进行变化；周期 $< 160\text{ms}$ ，就输出等 duty 脉冲；
- 支持脉冲常数可配：一共 4 套脉冲常数寄存器，其中 HFONST4 负责 D2F00/03，HFONST5 负责 D2F01/04，HFONST6 负责 D2F02/05，HFONST7 负责 D2F06~11；
- D2F00~D2F02 支持自动进行 A 通道基波有功功率积分，积分方式代数和/正向/绝对值/反向可选（积分方式配置根据 EMU 章节 EMUCON5 寄存器配置），输出能量和脉冲；
- D2F03~D2F05 支持自动进行 B 通道基波有功功率积分，积分方式代数和/正向/绝对值/反向可选（积分方式配置根据 EMU 章节 EMUCON5 寄存器配置），输出能量和脉冲；
- 功率为 32 位有符号数，即输入范围为 $\pm 2^{31}$ ；
- 每个积分器有各自的中断状态位和中断使能位；
- 能量寄存器模式可配：累加型或读后清零型；
- 系统时钟 32.768KHz 时，积分时钟为 32.768KHz ；其他系统时钟频率下，积分时钟为 921.6KHz 。

27.3 使用流程

软件流程：

1. 配置 MOD1_EN 寄存器 D2F_EN 位，打开模块时钟；
2. 配置 D2FCFG. D2F_DISABLE=0，使能 D2F 模块；
3. 按需求配置能量寄存器类型 D2FCFG. ERegCAR；
4. 按需求配置 D2F 中断使能和脉冲输出口；
5. 配置脉冲常数寄存器 HFConstx；
6. 按需求配置 D2FPxx，启动 D2Fxx 积分器。

应用 1：自定义功率和能量

软件将 32bit 功率值填入到 D2FP00~D2FP11 寄存器，D2FPx 为二进制补码格式，实际计算电能时硬件全部转给正功率计算，电能的正反向需要软件自己判断；D2F 模块按照积分时钟进行积分，输出能量、脉冲和中断。

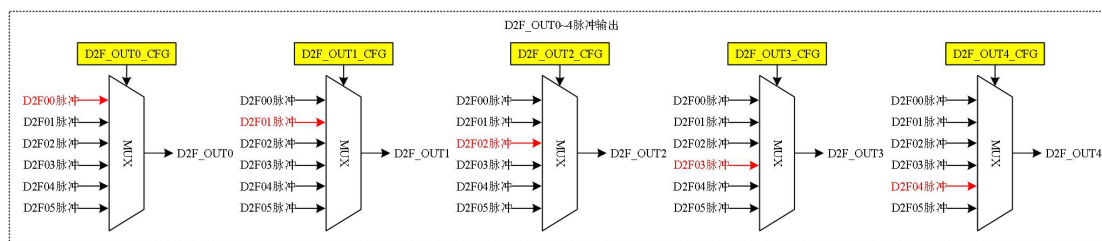
应用 2：基波有功电能自动输出

配置 HFConst4/5/6，配置 EMUCON5.FPx_MOD 选择能量累加方式，

配置 EMUCON5.FEPx_AutoCalc=1 使能基波有功电能自动输出功能，硬件可自动读取 32bit SPL_FPx 寄存器值填入 D2FPx 寄存器按照配置的累加方式进行积分，输出电能和脉冲。

注意：该模式下相应的 D2FPx 不支持软件手动写入。

自定义 D2F 脉冲输出示意图：



27.4 寄存器列表

基地址	0x4005C000
-----	------------

偏移地址	名称	R/W	有效 字长	复位值	功能描述
0x0	HFCnst4	R/W	2	0x0	D2F00/03 的高频脉冲常数寄存器
0x4	HFCnst5	R/W	2	0x0	D2F01/04 的高频脉冲常数寄存器
0x8	HFCnst6	R/W	2	0x0	D2F02/05 的高频脉冲常数寄存器
0xC	HFCnst7	R/W	2	0x0	D2F06~11 的高频脉冲常数寄存器
0x10	D2FCFG	R/W	1	0x0	D2F 配置寄存器
0x14	D2FOUT_CFG	R/W	3	0x43210	自定义D2F脉冲输出配置寄存器
0x18	IE	R/W	2	0x0	D2F 中断使能寄存器
0x1C	IF	R/W	2	0x0	D2F 中断标志寄存器
0x20~0x4C	D2FFCnt00~ D2FFCnt11	R/W	2	0x0	D2F 快速脉冲计数器
0x50~0x7C	D2FP00~D2FP11	R/W	4	0x0	D2F 功率输入寄存器
0x80~0xAC	D2FE00~D2FE11	R	3	0x0	D2F 自定义能量寄存器

注意：该模块寄存器不支持 bitband 操作。

27.5 寄存器定义

27.5.1 自定义 D2F 高频脉冲常数寄存器 HFCnst4/5/6(0x00~0C)

D2F 高频脉冲常数寄存器

偏移地址：00H；字长：4 字节；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:16	Reserved	保留	R	0
15:0	HFCnst4	自定义电能寄存器 D2FE00/03 及自定义脉冲 D2F_OUT0/D2F_OUT3 的脉冲常数。高频脉冲常数寄存器是 16 位无符号数。	R/W	0

偏移地址：04H；字长：4 字节；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:16	Reserved	保留	R	0
15:0	HFCnst5	自定义电能寄存器 D2FE01/04 及自定义脉冲 D2F_OUT1/D2F_OUT4 的脉冲常数。高频脉冲常数寄存器是 16 位无符号数。	R/W	0

偏移地址：08H；字长：4 字节；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:16	Reserved	保留	R	0
15:0	HFCnst6	自定义电能寄存器 D2FE02/05 及自定义脉冲 D2F_OUT2/D2F_OUT5 的脉冲常数。高频脉冲常数寄存器是 16 位无符号数。	R/W	0

偏移地址：0CH；字长：4 字节；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:16	Reserved	保留	R	0

15:0	HFConst7	自定义电能寄存器 D2FE06~11 的脉冲常数。高频脉冲常数寄存器是 16 位无符号数。	R/W	0
------	----------	---	-----	---

HFConst4/5/6/7 是 12 个自定义电能寄存器 D2FE00-D2FE11 及 6 个自定义脉冲 D2F_OUT0~D2F_OUT5 的脉冲常数，对应关系见上表。

高频脉冲常数寄存器是 16 位无符号数。

计量通道的 D2F 积分的时钟是 921.6KHz，自定义 D2F 积分的时钟也为 921.6KHz；

若想保证自定义 D2F 与计量通道出脉冲速度一致，假设二者的电表脉冲常数 EC 定义的完全一致，那么就要满足两者的 HFConst 寄存器一致。

直接读取芯片的功率寄存器值（32bit，例如 PA 等），填入功率寄存器 D2FP，然后脉冲输出 D2F_OUT 就能与芯片计量通道对应的脉冲输出速度保持一致。

如果定义的脉冲常数不同，通过调整 HFConst 和自定义功率寄存器填入值均可。

27.5.2 自定义 D2F 配置寄存器 D2FCFG(0x10)

D2F配置寄存器

偏移地址：10H；字长：1字节；默认值：0x2

位	位名称	功能描述	R/W	复位值
31:2	Reserved	保留	R	0
1	D2F_DISABLE	D2F 模块使能信号： =0，使能 D2F 模块 =1，复位所有计数器和数字运算单元，不复位寄存器值，默认值为 1。 当启动积分时，需要先将该位配置为 0，此后保持该位为 0，直接填写 D2FP 寄存器会立即启动积分。	R/W	1
0	ERegCAR	能量寄存器类型选择： =0，读后清零型。 =1，累加型。默认为 0。	R/W	0

27.5.3 自定义 D2F 脉冲输出配置寄存器 D2FOUT_CFG(0x14)

自定义D2F脉冲输出配置寄存器

偏移地址：14H；字长：3字节；默认值：0x43210

位	位名称	功能描述	R/W	复位值
31:19	Reserved	保留	R	0
18:16	D2F_OUT4_CFG	D2F_OUT4 输出的脉冲类型选择寄存器，配置选项同 D2F_OUT0_CFG，默认输出 D2F01 脉冲。	R/W	100
15	Reserved	保留	R	0
14:12	D2F_OUT3_CFG	D2F_OUT3 输出的脉冲类型选择寄存器，配置选项同 D2F_OUT0_CFG，默认输出 D2F01 脉冲。	R/W	011
11	Reserved	保留	R	0
10:8	D2F_OUT2_CFG	D2F_OUT2 输出的脉冲类型选择寄存器，配置选项同 D2F_OUT0_CFG，默认输出 D2F01 脉冲。	R/W	010

7	Reserved	保留	R	0
6:4	D2F_OUT1_CFG	D2F_OUT1 输出的脉冲类型选择寄存器，配置选项同 D2F_OUT0_CFG，默认输出 D2F01 脉冲。	R/W	001
3	Reserved	保留	R	0
2:0	D2F_OUT0_CFG	D2F_OUT0 输出的脉冲类型选择寄存器： =000，D2F00 脉冲； =001，D2F01 脉冲； =010，D2F02 脉冲； =011，D2F03 脉冲； =100，D2F04 脉冲； =101，D2F05 脉冲； =其他，保留。	R/W	000

27.5.4 自定义 D2F 中断使能寄存器 IE(0x18)

D2F中断使能寄存器

偏移地址：18H；字长：2 字节；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:12	Reserved	保留	R	0
11	D2F11IE	电能积分单元 D2F11 中断使能	R/W	0
10	D2F10IE	电能积分单元 D2F10 中断使能	R/W	0
9	D2F09IE	电能积分单元 D2F09 中断使能	R/W	0
8	D2F08IE	电能积分单元 D2F08 中断使能	R/W	0
7	D2F07IE	电能积分单元 D2F07 中断使能	R/W	0
6	D2F06IE	电能积分单元 D2F06 中断使能	R/W	0
5	D2F05IE	电能积分单元 D2F05 中断使能	R/W	0
4	D2F04IE	电能积分单元 D2F04 中断使能	R/W	0
3	D2F03IE	电能积分单元 D2F03 中断使能	R/W	0
2	D2F02IE	电能积分单元 D2F02 中断使能	R/W	0
1	D2F01IE	电能积分单元 D2F01 中断使能	R/W	0
0	D2F00IE	电能积分单元 D2F00 中断使能	R/W	0

该寄存器为D2F中断使能寄存器。当该事件发生时，相应D2FIF置位，如果在相应中断允许位配置为1则产生D2F中断。中断号4。

27.5.5 自定义 D2F 中断标志寄存器 IF(0x1C)

D2F中断标志寄存器

偏移地址：1CH；字长：2 字节；默认值：0x0

位	位名称	功能描述	R/W	复位值
31:12	Reserved	保留	R	0
11	D2F11IF	电能积分单元 D2F11 电能脉冲标志。写 1 清零，若对应位的 D2FIE=1，清零同时清中断。	R/W	0
10	D2F10IF	电能积分单元 D2F10 电能脉冲标志。写 1 清零，若对应位的 D2FIE=1，清零同时清中断。	R/W	0

9	D2F09IF	电能积分单元 D2F09 电能脉冲标志。写 1 清零，若对应位的 D2FIE=1，清零同时清中断。	R/W	0
8	D2F08IF	电能积分单元 D2F08 电能脉冲标志。写 1 清零，若对应位的 D2FIE=1，清零同时清中断。	R/W	0
7	D2F07IF	电能积分单元 D2F07 电能脉冲标志。写 1 清零，若对应位的 D2FIE=1，清零同时清中断。	R/W	0
6	D2F06IF	电能积分单元 D2F06 电能脉冲标志。写 1 清零，若对应位的 D2FIE=1，清零同时清中断。	R/W	0
5	D2F05IF	电能积分单元 D2F05 电能脉冲标志。写 1 清零，若对应位的 D2FIE=1，清零同时清中断。	R/W	0
4	D2F04IF	电能积分单元 D2F04 电能脉冲标志。写 1 清零，若对应位的 D2FIE=1，清零同时清中断。	R/W	0
3	D2F03IF	电能积分单元 D2F03 电能脉冲标志。写 1 清零，若对应位的 D2FIE=1，清零同时清中断。	R/W	0
2	D2F02IF	电能积分单元 D2F02 电能脉冲标志。写 1 清零，若对应位的 D2FIE=1，清零同时清中断。	R/W	0
1	D2F01IF	电能积分单元 D2F01 电能脉冲标志。写 1 清零，若对应位的 D2FIE=1，清零同时清中断。	R/W	0
0	D2F00IF	电能积分单元 D2F00 电能脉冲标志。写 1 清零，若对应位的 D2FIE=1，清零同时清中断。	R/W	0

该寄存器为D2F事件状态寄存器，当D2FE能量寄存器加1时，相应的标志位会置1。当对应中断允许位配置为1，状态位置1会导致产生CPU EMU中断1。

27.5.6 自定义 D2F 快速脉冲计数器 (0x20~0x4C)

偏移地址	20H	24H	28H	2CH	30H	34H
寄存器	D2FFCnt00	D2FFCnt01	D2FFCnt02	D2FFCnt03	D2FFCnt04	D2FFCnt05
默认值	0x0	0x0	0x0	0x0	0x0	0x0

偏移地址	38H	3CH	40H	44H	48H	4CH
寄存器	D2FFCnt06	D2FFCnt07	D2FFCnt08	D2FFCnt09	D2FFCnt10	D2FFCnt11
默认值	0x0	0x0	0x0	0x0	0x0	0x0

快速脉冲计数器寄存器为 2 字节无符号数，可读可写。

当快速脉冲计数寄存器 D2FFCntx 计数值的绝对值的 2 倍大于等于对应的 HFConst 时，相应会有脉冲溢出，能量寄存器的值会相应的加 1。

27.5.7 自定义 D2F 功率寄存器 (0x50~0x6C)

移地址	50H	54H	58H	5CH	60H	64H
寄存器	D2FP00	D2FP01	D2FP02	D2FP03	D2FP04	D2FP05
默认值	0x0	0x0	0x0	0x0	0x0	0x0
偏移地址	68H	6CH	70H	74H	78H	7CH
寄存器	D2FP06	D2FP07	D2FP08	D2FP09	D2FP10	D2FP11
默认值	0x0	0x0	0x0	0x0	0x0	0x0

功率输入寄存器 D2FP00- D2FP11 为二进制补码格式，32 位有符号数，其中最高位是符号位，可读可写。

当往 D2FP 寄存器写入功率值时，该功率会按照对应的脉冲常数 HFConst 进行积分，积分的电能存放在 D2FE00~D2FE11 寄存器中，其中 D2FE00~D2FE05 可以输出脉冲，对应脉冲分别是 D2F_OUT0~D2F_OUT5，可以根据 GPIO 复用配置从 IO 口输出。脉冲按照电表要求，高电平固定保持 80ms，低电平根据周期值进行变化；如果周期小于 160ms，那么就输出等 duty 脉冲。

27.5.8 自定义 D2F 能量寄存器 (0x80~0xAC)

偏移地址	80H	84H	88H	8CH	90H	94H
寄存器	D2FE00	D2FE01	D2FE02	D2FE03	D2FE04	D2FE05
默认值	0x0	0x0	0x0	0x0	0x0	0x0

偏移地址	98H	9CH	A0H	A4H	A8H	ACH
寄存器	D2FE06	D2FE07	D2FE08	D2FE09	D2FE10	D2FE11
默认值	0x0	0x0	0x0	0x0	0x0	0x0

自定义电能参数是 24bit 无符号数，只读，代表对应脉冲的累加个数。寄存器最小单位代表的能量为 1/EC kWh，其中 EC 为电表常数。

能量寄存器在 0xFFFFF 到 0x000000 时溢出，由用户软件自行判断管理，硬件不做处理。

自定义能量寄存器可以选择为累加型或者读后清零型，由 D2FCFG 配置寄存器控制。

28 计量误差温补模块 ECT（新增）

28.1 概述

影响整表计量精度温度特性的主要因素包括：

- 电流采样电路（锰铜、电流互感器及取样电阻等）
- 电压采样电路（电阻列分压、电压互感器等）
- ADC 温度系数
- 基准电压温度系数等

整表计量精度的温度特性由上述多种因素综合而成，且经验表明，在外部采样电路及 ADC 固定的条件下，单一优化基准电压温度系数 Trim 值难以得到精细化的、理想的温度补偿曲线。在实际应用中，多采用实测整表温度曲线，标定分段线性拟合 K 系数，现场定时测温，软件实时计算并调节误差补偿增益系数的方法。

ETC（Error Temperature Compensate）模块主要目的就是解决上述问题，通过定时测温，实时计算更新计量通道的增益寄存器，达到自动温度补偿的效果。上述描述中的 K 系数由软件标定并且配置到 ETC 模块，得到更好的计量精度。

28.2 温补增益系数的计算公式

已知高温段误差温补线性拟合系数为 HT_KI，实测温度为 T，求误差温度补偿增益系数 HT_IGAIN 计算公式。

以 A 相电流为例，高温段误差温补线性拟合系数为 HT_KIA，电流通过误差温度补偿增益前为 I，增益补偿后为 I'，误差温度补偿增益系数为：HT_IGAIN，补偿公式为：

$$I' = I(1 + ETCHT_IGAIN) = \frac{I}{1 + INF_{HT_T'}} = \frac{I}{1 + HT_KIA(T' - 25^{\circ}\text{C})}$$

跟据上述等式可以推导得：

$$ETCHT_IAGAIN = \frac{-HT_KIA(T'-25^{\circ}\text{C})}{1+HT_KIA(T'-25^{\circ}\text{C})}$$

同理可得低温段误差温度补偿增益:

$$ETCLT_IAGAIN = \frac{-LT_KIA(T'-25^{\circ}\text{C})}{1+LT_KIA(T'-25^{\circ}\text{C})}$$

电压通道依次类推。

28.3 特点

- 支持 U/IA/IB 3 个通道误差自动温度补偿
- 每个通道高温段、低温段均有独立的补偿系数
- 支持三种工作模式：
 - 手动模式，CPU 直接写增益
 - 单次自动模式，cpu 配置温度值进行温度补偿
 - 循环自动模式，可实现定时温度补偿全程软件无干预
- 循环自动模式温补周期 1s-256s 可配置
- 支持温度防抖功能，|两次测温温差| > 设定的温度阈值，才启动温度补偿增益系数的计算
- 支持 25℃附近的温度保护功能，且温度保护区域可配置
- 支持温度越界保护功能，温补低温下限高温上限均可配置，可选温度越界时是否进行温度补偿计算及更新 ECT 温补增益寄存器
- 支持误差越界保护功能，误差越界阈值可配置，支持误差越界报警中断，越界的温补增益系数不更新至 EMU 通道温补增益系数寄存器
- 支持 ECT 温补增益系数更新中断

28.4 功能描述

28.4.1 功能框图

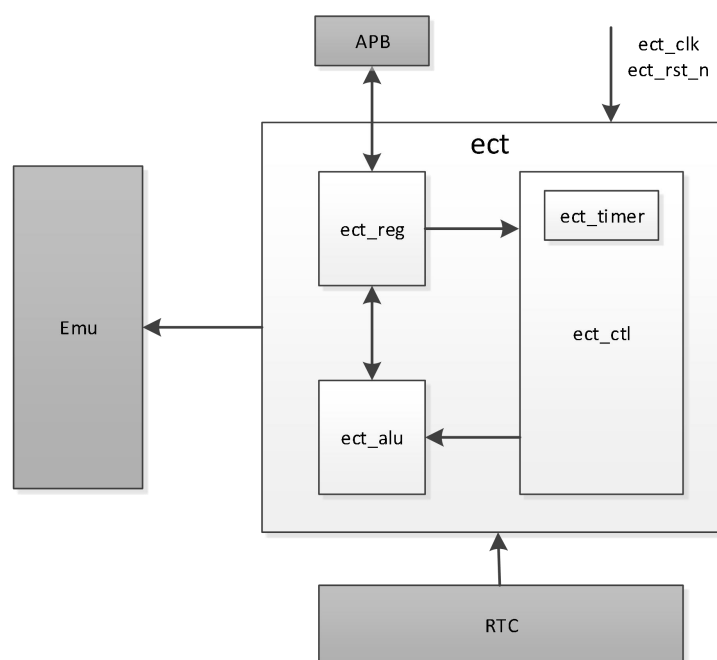


图 ECT 功能框图

如图所示，ECT（Error Temperature Compensation）模块包括：

Ect_reg: 寄存器模块

Ect_ctl: 进行高低温判断、阈值判断、误差判断等

Ect_alu: 增益计算单元

28.4.2 工作模式

ECT 支持三种工作模式：手动模式、单次自动模式、循环自动模式。三种工作模式对应不同的工作流程和软件控制方法。

- 手动模式

- 1) CPU 从 RTC 读温度值->
- 2) 软件计算增益->
- 3) 软件填写 TEMP_UD 和 XXGAIN 寄存器

注：硬件无标识表明是手动模式，只是 TEMP_UD 和 XXGAIN 寄存器留有 CPU 可配置的选项

- 单次自动模式

- 1) CPU 填温度值->
- 2) 温度保护事件判断及处理->
- 3) 高温段低温段判断->
- 4) 温度越界事件判断及处理->
- 5) 温度防抖，温度阈值及启动计算单元判断->
- 6) 启动计算单元完成温补增益系数的计算->
- 7) 误差越界事件判断及中断输出->
- 8) ECT 温补增益系数寄存器更新->
- 9) TEMP_UD 输出

- 循环自动模式

从 RTC 自动读取温度值->事件处理->计算增益

ECT 启动 ECT_Timer 计时器，ECT 按 ECT_Timer_SET 设置的周期，周而复始地循环上述流程。

与单次自动不同的是温度来源不同，单次自动温度值需要 CPU 写进寄存器，循环自动则是从 RTC 自动的获取温度值

28.4.3 温度保护

当 CTRL 寄存器的 PROT_EN=1 温度保护功能开启时，若检测到的温度减去 25℃ 得到的绝对温度值小于 PROT_TEMP 寄存器设定的 25℃ 温度保护阈值，则触发自动温补温度保护事件，即

$|\text{TEMP} - 25^\circ\text{C}| < \text{PROT_TEMP}$ 时，触发自动温补温度保护事件。

该事件发生后 ECT 停止自动温补系数的计算，并将所有增益系数 xxGAIN 寄存器清零，同时 TEMP_PROT_IF 标志置位。

若清零动作导致 xxGAIN 发生了更新，则 TEMP_UD 也更新为当前检测温度值，GAINUD[2:0] 相应通道标志也置位。

28.4.4 高温低温分段判断

若未发生温度保护事件，控制单元下一步会进行高温段和低温段判断，同样和 25℃ 比较。

温度 $< 25^\circ\text{C}$ ，判断为低温段，后续流程使用低温段一套寄存器及判断准则，包括低温段 K 系数寄存器，低温越界阈值寄存器 LT_LL，低温段温度阈值寄存器 LT_GAP。

温度 $\geq 25^{\circ}\text{C}$ ，判断为高温段，后续流程使用高温段一套寄存器及判断准则，包括高温段 K 系数寄存器，高温越界阈值寄存器 HT_LL，低温段温度阈值寄存器 HT_GAP。

28.4.5 温度越界事件处理

高温段低温段确定后，控制单元下一步会进行温度越界事件判断及处理。

温度越界事件定义为：自动温补模式下，当 CTRL 寄存器 TEMPEL_PORT_EN=1，温度越界保护功能开启，若 $\text{TEMP} > \text{HT_UL}$ 设置的自动温补温度上限，或 $\text{TEMP} < \text{LT_LL}$ 设置的自动温补温度下限时，发生温度越界事件，将标志位 TEMP_EL_IF 置位。

发生温度越界事件时，可通过 CTRL 寄存器 TEMPEL_GAINCAL_EN 位选配是否停止自动下一步温补系数的计算。

若 TEMPEL_GAINCAL_EN=1，将启动下一步温度阈值的判断；

若 TEMPEL_GAINCAL_EN=0，本轮温补结束。

28.4.6 温度防抖

若未发生温度保护和温度越界事件，或者发生了温度越界事件但 TEMPEL_GAINCAL_EN=1 时，启动温度阈值判断功能。

当前温度减去上次更新温度值的绝对值大于或等于低温段或高温段温度设置阈值时，启动 ECT 计算单元计算温补系数，即

$|\text{TEMP} - \text{TEMP_UD}| \geq \text{LT_GAP}/\text{HT_GAP}$ 时，符合计算条件，启动 ECT 计算单元计算温补该温度段的温补增益系数，计算完成后，进行误差越界事件判断和处理。否则，本轮温补结束。

28.4.7 增益系数计算

$$\text{ECT_TxGAIN} = \frac{-\text{LT_KIX} * (T - 25)}{1 + \text{LT_KIX} * (T - 25)}$$

其中 LT_KIX 为低温段 Ix 通道误差自动温补线性拟合系数，详见 LT_KIX 寄存器说明；T 为本次补偿的测温值 TEMP。高温段使用高温段补偿 K 系数 HT_KIX。

28.4.8 误差越界事件处理

当温补误差保护功能开启时，若计算模块算出的某通道高温段或低温段的增益系数超过 PROT_xxGAIN 设置的阈值时，该通道发生误差越界事件。

当某通道发生误差越界事件时，该通道误差越界标志 ERR_EL_xx 置位。任何一位标志置位均会使 ERREL_IF 置位，若 ERREL_IE 中断使能，任何一位标志置位均会产生 ERREL_IF 中断，清除中断，同时会清除 ERR_EL_xx 标志位。

28.5 寄存器描述

28.5.1 寄存器列表

模块名	物理地址	映射地址
ECT	0x40078000	0x40078000
寄存器名	地址偏移量	描述
WREN	Offset+0x0	写使能寄存器
CTRL	Offset+0x4	控制寄存器
EN	Offset+0x8	模块使能寄存器
STATUS	Offset+0xC	状态寄存器
IE	Offset+0x10	中断使能寄存器
LT_SET	Offset+0x14	低温段阈值配置寄存器
HT_SET	Offset+0x18	高温段阈值配置寄存器
TIMER_SET	Offset+0x1C	循环周期配置寄存器

PROT_TEMP	Offset+0x20	25℃温度保护阈值配置寄存器
PROT_IAGAIN	Offset+0x24	IA 通道误差保护阈值寄存器
PROT_IBGAIN	Offset+0x28	IB 通道误差保护阈值寄存器
PROT_UGAIN	Offset+0x2C	U 通道误差保护阈值寄存器
LT_KIA	Offset+0x30	低温段 IA 通道 K 系数寄存器
LT_KIB	Offset+0x34	低温段 IB 通道 K 系数寄存器
LT_KU	Offset+0x38	低温段 U 通道 K 系数寄存器
HT_KIA	Offset+0x3C	高温段 IA 通道 K 系数寄存器
HT_KIB	Offset+0x40	高温段 IB 通道 K 系数寄存器
HT_KU	Offset+0x44	高温段 U 通道 K 系数寄存器
TEMP	Offset+0x48	当前温度寄存器
IAGAIN	Offset+0x4C	IA 通道增益补偿寄存器
IBGAIN	Offset+0x50	IB 通道增益补偿寄存器
UGAIN	Offset+0x54	U 通道增益补偿寄存器
TEMP_UD	Offset+0x58	当前补偿对应温度寄存器

28.5.2 WREN (0x0)

配置写使能寄存器

位	位名称	功能描述	读/写标志	复位值
31:8	reserved	保留位	R/W	0
7:0	WREN	其他寄存器的写使能： 写 0xEA，其余寄存器可写；写入其他值不可写	R/W	0

28.5.3 CTRL (0x4)

控制寄存器

位	位名称	功能描述	读/写标志	复位值
31:8	Reserved	保留	RO	0
7	TEMPEL_PORT_EN	0: 关闭温度越界保护功能 1: 启动温度越界保护功能	R/W	0
6	TEMPEL_GAINCAL_EN	0: 温度越界事件发生，计算引擎不计算温补增益系数，仅报温度越界事件 1: 温度越界事件发生，不仅报温度越界事件，还启动计算引擎计算温补增益系数，并写入相应的温补增益系数寄存器，并触发 GAIN_UD 标志置位 注：该配置位仅在 TEMPEL_PORT_EN=1 时有效	R/W	0
5	ERREL_PROT_EN	0: 关闭温补误差越界保护功能 1: 启动温补误差越界保护功能	R/W	0
4	PORT_EN	0: 关闭温补温度区域保护功能。 1: 启动温补温度区域保护功能。 该 bit 主要保护 25℃附近保持稳定。	R/W	0

3:1	CH_SEL	自动温补通道选择配置： CH_SEL[2:0]对应 U，IB，IA 通道 0: 不选择{IA,IB,U}通道自动温补 1: 选择{IA,IB,U}通道自动温补	R/W	0
0	MODE	自动温补模式配置： 0: 单次自动温补模式； ECT_TEMP 寄存可读可写，CPU 写入温度，进行单次自动温补，若满足温补条件，温补增益寄存器单次自动调节。 1: 循环自动温补模式； 按 ECT_Timer_SET 寄存器设置的周期，自动从 RTC 读取温度值循环温补。若满足温补条件，温补增益寄存器周期自动调节。	R/W	0

28.5.4 EN (0x8)

模块使能寄存器

位	位名称	功能描述	读/写标志	复位值
31:8	reserved	保留位	R/W	0
15:0	EN	其余寄存器的写使能； 写 0x685E，启动 ECT 温补 写其他，关闭 ECT 温补	R/W	0

28.5.5 STATUS (0xC)

状态寄存器

位	位名称	功能描述	读/写标志	复位值
31:11	Reserved	保留	R	0
10	ERR_EL_U	U 通道自动温补误差越界状态位	R	0
9	ERR_EL_IB	IB 通道自动温补误差越界状态位	R	0
8	ERR_EL_IA	IA 通道自动温补误差越界状态位	R	0
7	GAINUD_U	U 通道增益系数更新标志	R	0
6	GAINUD_IB	IB 通道增益系数更新标志	R	0
5	GAINUD_IA	IA 通道增益系数更新标志	R	0
5	TEMP_JIT_IF	温度抖动中断标志，写 1 清零	RW1C	0
4	TEMP_EL_IF	温度越界中断标志，写 1 清零	RW1C	0
3	TEMP_PROT_IF	温度保护中断标志，写 1 清零	RW1C	0
2	DONE_IF	自动温补完成一轮增益计算中断标志，写 1 清零	RW1C	0
1	ERREL_IF	增益误差越界中断标志，写 1 清零，清中断时会请 ERR_EL_XX 标志位。	RW1C	0
0	GAINUD_IF	温补增益系数更新中断标志，写 1 清零	RW1C	0

28.5.6 IE (0x10)

中断使能寄存器

位	位名称	功能描述	读/写标志	复位值
31:6	Reserved	保留	R	0
5	TEMP_JIT_IE	温度抖动中断使能	RW	0
4	TEMP_EL_IE	温度越界中断使能	RW	0
3	TEMP_PROT_IE	温度保护中断使能	RW	0
2	DONE_IE	自动温补完成一轮增益计算中断使能	RW	0
1	ERREL_IE	增益误差越界中断使能	RW	0
0	GAINUD_IE	温补增益系数更新中断使能	RW	0

28.5.7 LT_SET (0x14)

低温段阈值配置寄存器

位	位名称	功能描述	读/写标志	复位值
31:26	Reserved	保留	RO	0
25:16	LT_LL	低温段 (<25℃为低温段, 下同) 误差温度补偿的温度下限值配置。TEMP_EL 使能, 检测温度低于该温度点, 发生自动温补温度越界事件。 该寄存器为二进制有符号数, 最高位为符号位, 配置最小为刻度为 0.25℃。默认为-40℃。	R/W	0x360
15:10	Reserved	保留	R/W	0
9:0	LT_GAP	当检测温度处于低温段 (<25℃) 时, 满足 TEMP - TEMP_UD ≥ LT_GAP 设置的阈值温度, 启动自动温补。配置最小为刻度为 0.25℃。默认为 5℃。	R/W	0x14

28.5.8 HT_SET (0x18)

高温段阈值配置寄存器

位	位名称	功能描述	读/写标志	复位值
31:26	Reserved	保留	RO	0
25:16	HT_UL	高温段 (≥25℃为高温段, 下同) 误差温度补偿的温度上限值配置, 若 TEMP_EL 使能, 检测温度高于该温度点, 发生自动温补温度越界事件。 该寄存器为二进制有符号数, 最高位为符号位, 配置最小为刻度为 0.25℃, 默认值为 85℃。	R/W	0x154
15:10	Reserved	保留	R/W	0
9:0	HT_GAP	当检测温度处于高温段 (≥25℃) 时, 满足 TEMP - TEMP_UD ≥ HT_GAP 设置的阈值温度时, 启动自动温补。配置最小为刻度为 0.25℃。默认为 5℃。	R/W	0x14

28.5.9 TIMER_SET (0x1C)

循环周期配置寄存器

位	位名称	功能描述	读/写标志	复位值
31:8	Reserved	保留	RO	0
7:0	Timer_Cyc	单位: s。 启动循环自动温补后, 自动温补单元按 (Timer_Cyc + 1) s 为周期定时检测温度。 默认值: 0x9。最高可设为 256s	R/W	0x9

28.5.10 PROT_TEMP (0x20)

25℃温度保护阈值配置寄存器

位	位名称	功能描述	读/写标志	复位值
31:10	Reserved	保留	RO	0
9:0	PROT_TEMP	自动温补温度保护区域设置。 当 PROT_EN 使能, 若 $ TEMP - 25^{\circ}C < PROT_TEMP$ 设定的保护温度阈值时, 触发自动温补温度保护事件。 配置最小为刻度为 0.25℃。默认设置在 2℃	R/W	0x8

28.5.11 PROT_IAGAIN (0x24)

IA 通道误差保护阈值寄存器

位	位名称	功能描述	读/写标志	复位值
31:16	Reserved	保留	RO	0
15:0	PROT_IAGAIN	自动温补误差保护阈值设置。 当 IA 通道补偿的电流通道增益系数的绝对值大于设置的阈值时, 即 $ IAGAIN > PROT_IAGAIN$ 时, 发生该电流通道温补误差越界事件。 一般设置在 $1\% \times 2^{15}$ 。	R/W	0x147

28.5.12 PROT_IBGAIN (0x28)

IB 通道误差保护阈值寄存器

位	位名称	功能描述	读/写标志	复位值
31:16	Reserved	保留	RO	0
15:0	PROT_IBGAIN	自动温补误差保护阈值设置。 当 IB 通道补偿的电流通道增益系数的绝对值大于设置的阈值时, 即 $ IBGAIN > PROT_IBGAIN$ 时, 发生该电流通道温补误差越界事件。 一般设置在 $1\% \times 2^{15}$ 。	R/W	0x147

28.5.13 PROT_UGAIN (0x2C)

U 通道误差保护阈值寄存器

位	位名称	功能描述	读/写标志	复位值
31:16	Reserved	保留	RO	0
15:0	PRO_UGAIN	电压通道自动温补误差保护阈值设置。 当 U 通道补偿的电流通道增益系数的绝对值大于设置的阈值时, 即 $ UGAIN > PROT_UGAIN$ 时, 发生该电压通道温补误差越界事件。 一般设置在 $1\% \times 2^{15}$ 。	R/W	0x147

28.5.14 LT_KIA (0x30)

低温段 IA 通道 K 系数寄存器

位	位名称	功能描述	读/写标志	复位值
31:17	Reserved	保留	RO	0
16:0	LT_KIA	低温段 IA 通道误差温补线性拟合系数 LT_KIA 寄存器。 根据用户在校表阶段测得 I _x 通道的低温段特性, 求该值, 并写入 EEPROM 保存, 上电时, 由 CPU 写入该寄存器。 电流通道增益校正后, 扫描-40℃~25℃标称电流温度曲线, 取最合适的一次线性拟合温度点 T', 测得 25℃电流值为 I _x (25℃), 拟合温度 T' 的电流值 I _x (T'); 使整个低温段温度曲线离 I _x (25)和 I _x (T')连接而成的直线间距最小, 则得 LT_KIx 的计算公式: $LT_KIx = ERR / (T' - 25)$, 其中 $ERR = (I_x(T') - I_x(25^\circ C)) / I_x(25^\circ C)$ 最高位为符号位, 满量程为 0.022, 默认值为 0	R/W	0

28.5.15 LT_KIB (0x34)

低温段 IB 通道 K 系数寄存器

位	位名称	功能描述	读/写标志	复位值
31:17	Reserved	保留	RO	0
16:0	LT_KIB	低温段 IB 通道误差温补线性拟合系数 LT_KIB 寄存器。 计算公式同 LT_KIA。	R/W	0

28.5.16 LT_KU (0x38)

低温段 U 通道 K 系数寄存器

位	位名称	功能描述	读/写标志	复位值
31:17	Reserved	保留	RO	0

16:0	LT_KU	低温段 U 通道误差温补线性拟合系数 LT_KU 寄存器。 计算公式同 LT_KIA。	R/W	0
------	-------	--	-----	---

28.5.17 HT_KIA (0x3C)

高温段 IA 通道 K 系数寄存器

位	位名称	功能描述	读/写标志	复位值
31:17	Reserved	保留	RO	0
16:0	HT_KIA	高温段 IA 通道误差温补线性拟合系数 HT_KIA 寄存器。 根据用户在校表阶段测得 I _x 通道的高温段特性，求该值，并写入 EEPROM 保存，上电时，由 CPU 写入该寄存器。 电流通道增益校正后，扫描 25℃~85℃ 标称电流温度曲线，取最合适的一次线性拟合温度点 T'，测得 25℃ 电流值为 I _x (25)，拟合温度 T' 的电流值 I _x (T')；使整个高温段温度曲线离 I _x (25) 和 I _x (T') 连接而成的直线间距最小，则得 HT_KI _x 的计算公式： $HT_KI_x = ERR / (T' - 25)$ ，其中 $ERR = (I_x(T') - I_x(25^\circ C)) / I_x(25^\circ C)$ 最高位为符号位，满量程为 0.022，默认值为 0	R/W	0

28.5.18 HT_KIB (0x40)

高温段 IB 通道 K 系数寄存器

位	位名称	功能描述	读/写标志	复位值
31:17	Reserved	保留	RO	0
16:0	HT_KIB	高温段 IB 通道误差温补线性拟合系数 HT_KIB 寄存器。 计算公式同 HT_KIA。	R/W	0

28.5.19 HT_KU (0x44)

高温段 U 通道 K 系数寄存器

位	位名称	功能描述	读/写标志	复位值
31:17	Reserved	保留	RO	0
16:0	HT_KU	高温段 U 通道误差温补线性拟合系数 HT_KU 寄存器。 计算公式同 HT_KIA。	R/W	0

28.5.20 TEMP (0x48)

当前温度值寄存器

位	位名称	功能描述	读/写标志	复位值
31:10	Reserved	保留	RO	0

9:00	TEMP	<p>1. 该寄存器保存本次从 RTC 读取的温度值 二进制补码表示，10 位有效位，Bit9 为符号位；最小刻度为 0.25 度。与实际温度的换算公式为：T = TEMP/4。</p> <p>2. 用户可在单次自动模式下通过软件填该温度值进行增益计算</p>	R/W	0x9
------	------	--	-----	-----

28.5.21 IAGAIN (0x4C)

IA 通道增益补偿寄存器

位	位名称	功能描述	读 / 写标志	复位值
31:16	Reserved	保留	RO	0
15:0	IAGAIN	<p>IA 通道的温补增益系数，在 EMU 计量通道增益校正之后。 计算公式： 以低温段为例， $IxGAIN = -LT_Kl_x * (T-25) / (1 + LT_Kl_x * (T-25))$， 其中，LT_Kl_x 为低温段 Ix 通道误差自动温补线性拟合系数，T 为本次补偿的测温值。 高温段使用高温段补偿 K 系数 HT_Kl_x，公式相同。 使用高温段还是低温段计算由硬件自动判断 最高位为符号位，满量程为 1，默认值为 0</p>	R/W	0

28.5.22 IBGAIN (0x50)

IB 通道增益补偿寄存器

位	位名称	功能描述	读/写标志	复位值
31:16	Reserved	保留	RO	0
15:0	IBGAIN	<p>IB 通道的温补增益系数，在 EMU 计量通道增益校正之后。 计算公式同 IAGAIN。 最高位为符号位，满量程为 1，默认值为 0</p>	R/W	0

28.5.23 UGAIN (0x54)

U 通道增益补偿寄存器

位	位名称	功能描述	读/写标志	复位值
31:16	Reserved	保留	RO	0
15:0	UGAIN	<p>U 通道的温补增益系数，在 EMU 计量通道增益校正之后。 计算公式同 IAGAIN。 最高位为符号位，满量程为 1，默认值为 0</p>	R/W	0

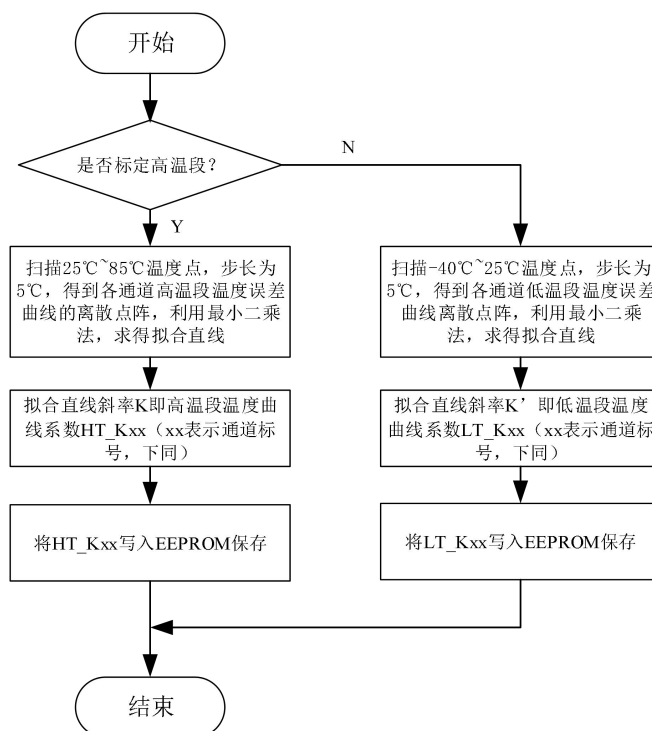
28.5.24 TEMP_UD (0x58)

当前补偿对应温度寄存器

位	位名称	功能描述	读/写标志	复位值
31:10	Reserved	保留	RO	0
9:00	TEMP_UD	该寄存器保存最近一次更新了温度补偿系数寄存器组温度值。 单次和循环自动温补模式由 ECT 在更新温补系数寄存器组后自动写入。手动模式，该寄存器由软件填写。	R/W	0x64

28.6 应用流程

28.6.1 温补 K 系数标定



28.6.2 手动温补模式

1. 用户通过多次测温得到“有效值相对 25℃ 的温度影响 (%) --温度”离散曲线；
2. 按照 25℃ 分界划分高低温段，根据步骤 1 测量的离散点，做高低温段线性拟合，拟合系数 K 的公式，计算如下：

$$K = \frac{\sum_{i=1}^n y_i (x_i - 25)}{\sum_{i=1}^n (x_i - 25)^2}$$

3. 用户通过读取 RTC_TEMP 得到温度值，并根据高低温段划分和相应温补线性拟合系数 K，确定误差温度补偿增益系数；
4. 填写 TEMP_UD 寄存器和 xxGAIN 寄存器；

继续手动温补，重复步骤 3~4。

28.6.3 单次自动温补模式

- 1、配置 CH_SEL 使能 U/IB/IA 哪个通道自动温补；
- 2、配置 MODE=0，选择温补模式为单次自动温补模式；
- 3、配置 LT_SET/HT_SET 寄存器，配置高低温段阈值；
- 4、配置相应的保护区设置寄存器 PROT_TEMP/PROT_xxGAIN；
- 5、配置相应的事件中断；
- 6、将标定好的温补线性拟合 K 系数进 LT_Kxx/HT_Kxx 寄存器；
- 7、配置 EN=0x685e 启动 ECT 温补；
- 8、查询 DONE_IF 标志，当 DONE_IF 置 1，完成一轮温补增益系数计算，ECT 模块又回到待命状态；

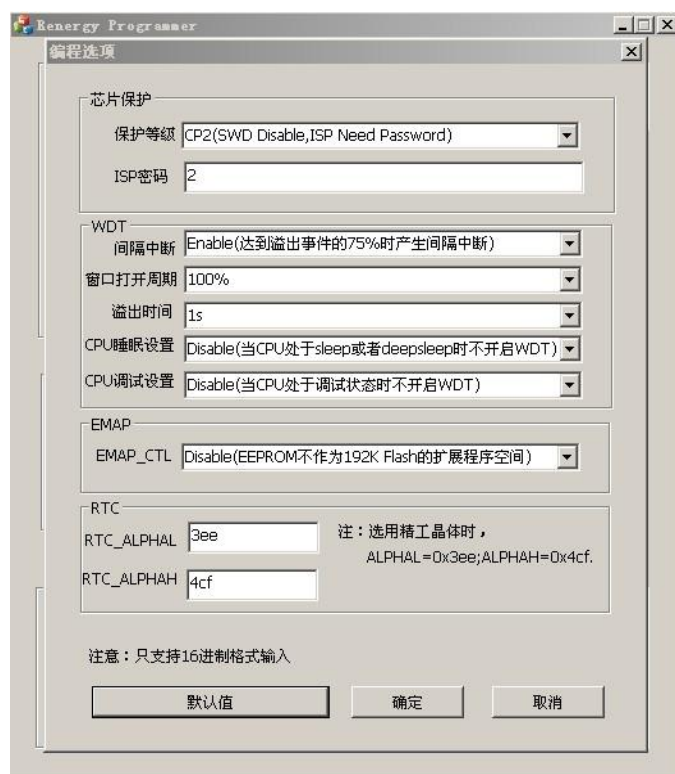
28.6.4 循环自动温补模式

- 1、配置 CH_SEL 使能 U/IB/IA 哪个通道自动温补；
- 2、配置 MODE=1，选择温补模式为循环自动温补模式；
- 3、配置 LT_SET/HT_SET 寄存器，配置高低温段阈值；
- 4、配置 Timer_SET 设置自动循环温补周期；
- 5、配置相应的保护区设置寄存器 PROT_TEMP/PROT_xxGAIN；
- 6、配置相应的事件中断；
- 7、将标定好的温补线性拟合 K 系数进 LT_Kxx/HT_Kxx 寄存器。
- 8、配置 EN=0x685e 启动 ECT 温补；
- 9、查询 DONE_IF 标志，当 DONE_IF 置 1，完成一轮温补增益系数计算；
- 10、按 Timer_SET 设定周期启动下一轮温补增益系数计算。

29 选项字节

Soc 内置了一个选项字节的区域，当芯片发生复位时，会自动配置选项字节，执行设定的指定功能。选项字节包括对芯片保护，WDT，EMAP 和 RTC 的配置。

选项字节编程可以通过锐能微的编程工具（MINIPRO 编程器或者 ISP 编程工具）进行设置，以 MINIPRO 编程器为例，打开编程选项对话框，即可对选项字节进行设置，如下图所示（详细的操作方法请见《MINIPRO 编程器使用手册》）。



29.1 芯片保护设置

选项字节提供的芯片保护功能可以保护Soc内置的Flash，用户可以通过设置保护等级和ISP密码对芯片进行保护。保护机制提供了如下保护等级：

保护等级	名称	说明
0	CP0	无任何保护（ISP 访问也不需要密码）
1	CP1	SWD 接口可访问芯片，ISP 访问需要密码
2	CP2	禁止通过 SWD 接口访问芯片，ISP 访问需要密码
3	CP3	禁止通过 SWD 和 ISP 接口访问芯片（ISP 只提供整片擦除 FLASH 的功能（在该保护等级下擦除操作会使芯片的保护等级将为 CP0））
4	CP4	禁止通过 SWD 和 ISP 接口访问芯片（ISP 只响应 CMD_UN 和 CMD_AL 命令，该保护等级的设置需要与客户确认，谨慎设置；当设置并通过编程器烧录后，没有任何途径擦、编、读 FLASH。）

29.2 WDT 设置

选项字节提供WDT的间隔中断，窗口打开周期，溢出时间，CPU睡眠设置，CPU调试设置，详细的意义请见WDT章节。如下表所示：

名称	描述	厂家默认值
间隔中断	0: Disable（不使能间隔中断） 1: Enable（达到溢出事件的 75%时产生间隔中断）	0
窗口打开周期	0: 25%	3

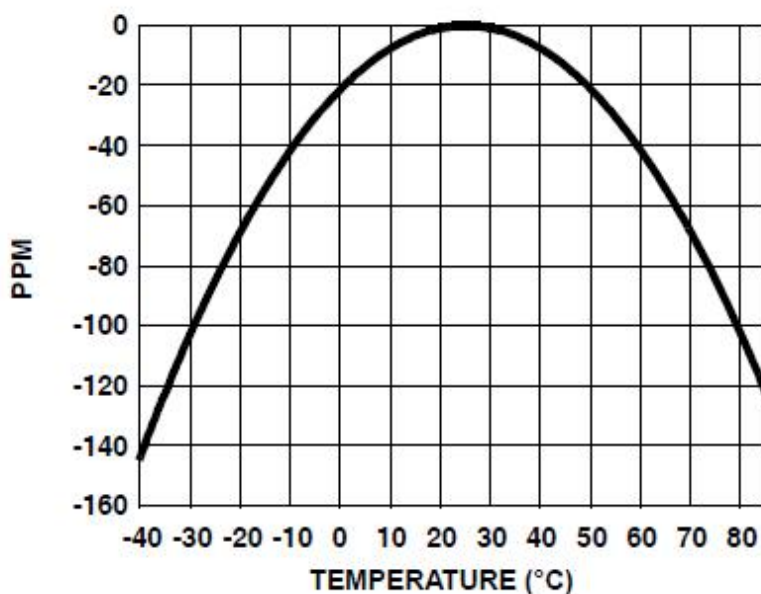
	1: 50% 2: 75% 3: 100% 在窗口打开期间将 0xBB 写入 WDTE 寄存器，看门狗清零并重新计数； 在窗口关闭期间将 0xBB 写入 WDTE 寄存器，会产生内部复位信号。	
溢出时间	0: 16ms 1: 32ms 2: 128ms 3: 512ms 4: 1s 5: 2s 6: 4s 7: 8s	4
CPU 睡眠设置	0: Disable (当 CPU 处于 sleep 或者 deepsleep 的时候不开启 WDT) 1: Enable (当 CPU 处于 sleep 或者 deepsleep 的时候开启 WDT)	0
CPU 调试设置	0: Disable (当 CPU 处于调试状态时不开启 WDT) 1: Enable (当 CPU 处于调试状态时开启 WDT) 注: CPU 处于调试状态指的是用户通过调试接口将 Cortex M0 停住 (PC 指针停止计数)。	0

29.3 RTC 设置

Soc的RTC内置了自动温补功能，可以对32k晶体进行自动温度补偿，以提供在-25℃~70℃范围输出准确的秒脉冲。

其中，晶体的温度频率曲线如下图所示，是以顶点为25度的二次曲线 ($f=f_0-\alpha*(T-T_0)$, T_0 为25度)。而事实上高温段 (25℃~85℃) 与低温段 (-25℃~70℃) 的二次曲线参数 α 并不相同，所以选项字节提供了RTC_ALPHAL和RTC_ALPHAH参数，填入的值为 $\text{round}(\alpha*32768)$ 其中round为四舍五入操作。

如果选用的是精工 (Seiko) 提供的高一致性晶体 (VT-200-F)，ALPHAL=0x3ee, ALPHAH=0x4cf。



30 编程支持

SoC 支持对内置 FLASH 的编程。

下载到 FLASH 中的代码文件需要加入校验字，即将（前 7 个 word 累加和的负数 $0-\text{sum}(w0...w6)$ ）存到 0x1c 地址。芯片 ROM 中的 bootloader 代码会检查该校验字是否正确，只有校验正确了才会跳出芯片 ROM 中的 bootloader 代码，跳转到 FLASH 运行用户代码。可通过“系统控制”章节中的“地址映射寄存器 MAP_CTRL”，查看芯片是否运行在运行 FLASH 中的用户代码，如果为 0 或 4 表示在运行 FLASH 中的用户代码，如果为 3 表示在运行 ROM 中的 bootloader。锐能微提供的编程算法和 minipro 编程器，已按此需求修改校验字。

建议客户调用锐能微库函数实现 IAP 功能；使用锐能微编程器完成 ISP 功能。

30.1 概述

SoC 编程系统具备如下特性：

- ◎ 内置对 FLASH 内容的保护机制
- ◎ 支持 ISP 编程模式；
- ◎ 支持 IAP 编程模式；
- ◎ 支持通过 SWD 下载烧录模式；
- ◎ 支持量产烧录模式；

30.2 Flash 保护机制

Flash 保护是允许用户在系统中通过使能不同的安全级别来限制对片内 Flash 访问的保护机制。保护机制保护了如下保护等级。用户可以通过“选项字节”设置芯片的保护等级。

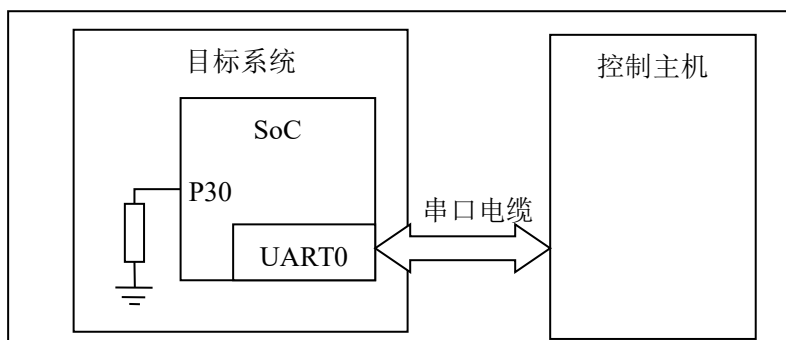
表 19-1 SoC 保护等级

保护等级	名称	说明
0	CP0	无任何保护（ISP 访问也不需要密码）
1	CP1	SWD 接口可访问芯片，ISP 访问需要密码
2	CP2	禁止通过 SWD 接口访问芯片，ISP 访问需要密码
3	CP3	禁止通过 SWD 和 ISP 接口访问芯片（ISP 只提供整片擦除 FLASH 的功能（在该保护等级下擦除操作会使芯片的保护等级将为 CP0））
4	CP4	禁止通过 SWD 和 ISP 接口访问芯片（ISP 只响应 CMD_UN 和 CMD_AL 命令，该保护等级的设置需要与客户确认，谨慎设置；当设置并通过编程器烧录后，没有任何途径擦、编、读 FLASH。）

30.3 在系统编程（ISP）

用户可以将 P30 信号拉低，并复位 SoC，让 SoC 进入 ISP 模式。ISP 模式的连接图如图 18-1 所示。

图 18-1 ISP 硬件配置图



ISP 的主要流程：

按连接图配置并连接目标系统和控制主机；

复位目标系统；

控制主机配置串口为 1 个起始位，8 个数据位，1 个停止位；

控制主机发送 “e” ；

目标系统响应 “Synchronized/r/n” ；

控制主机发送 “Synchronized /r/n” ；

目标系统响应 “7373(1843)/r/n” ；（如果当前系统频率为 7.3728M，则发送 7373；如果为 1.8432M，则发送 1843）

控制主机可以根据需要执行对应的 ISP 命令；

30.3.1 ISP 通讯协议

所有ISP 命令都以单个ASCII 字符串形式发送。字符串应当以回车（/r）和/或换行（/n）控制字符作为结束符。

所有ISP 响应都是以<CR><LF>结束的ASCII 字符串形式发送。

数据以原始数据（不转化为ASCII码）发送和接收。

- 命令格式

命令 参数 0 参数 1 ... 参数 n/r/n

{数据}

- 响应格式

返回代码/r/n

响应0/r/n

响应1/r/n

...

响应n/r/n

{数据}

- 数据格式

在启动WM，RM两个命令后，会启动ISP的数据传输。数据以行为单位进行传输，1行最多包含16个32 bit（不满16个数据，则发相应个数）的数据；每传输完1个Block（1个Block最多包含32行（不满32行，则发相应行数））数据，发送一个校验行（该Block数据的累加校验和的负数补码（以word进行计算））。

当Soc接收完一个完整的Block数据后，会对数据进行校验，如果校验通过，则发送 “OK/r/n”命令；如果数据校验出错，或者接收到不合法的数据包，发送 “RS/r/n”，当编程器接收到该命令，需要重新发送该Block数据。

数据传输形式：

当行内数据为 0x7e 时，转义成 0x7d, 0x5e 发送；当行内数据为 0x7d 时，转义为 0x7d, 0x5d 发送
数据行格式：（B 代表传输数据 Byte，为 16 进制）

表 19-2 ISP 数据传输格式

行首	1	2	3	4	5	6	64	65	行尾
0x7e	Num	B0	B1	B2	B3	B4	B62	B63	0x7e

校验行格式：（ASCII 码。S 代表累加校验和 SUM）

表 19-3 ISP 数据校验格式

行首	1	2	3	4	5	行尾
0x7e	0xff	S0	S1	S2	S3	0x7e

30.3.2 使用的 SoC 资源

ISP使用片内0x10017000到0x10018000范围内的RAM，堆栈位于RAM的顶部。Flash能使用0x10000000-0x10001000（4KB）范围内的RAM进行编程。

30.3.3 ISP 命令

每个ISP命令都支持具体的状态代码。当接收到未定义命令时，命令处理程序发送返回代码INVALID_COMMAND。

命令和返回代码为ASCII 格式。只有当接收到的ISP命令执行完毕时，ISP 命令处理器才会发送CMD_SUCCESS，这时主机才能发送新的ISP 命令。

ISP命令分成三种：

1. 普通命令：只在CP0，或者在CP1，CP2并且保护密码正确的情况下，可以访问
2. UN命令在CP0，CP1，CP2保护等级（保护密码不提供）下，都能访问
3. FC,AL命令在任何情况下都能访问

表 19-4 ISP命令

命令	用法	属性
设置波特率	BS <波特率> <停止位>	普通命令
回显	RD <开关设置>	普通命令
写内存	WM <地址> <字节长度> <模式>	普通命令
读内存	RM <地址> <字节长度> <模式>	普通命令
Flash 页擦除	FP <页地址>	普通命令
Flash 块擦除	FS <块地址>	普通命令
Flash 片擦除	FC	特殊命令
Flash 块查空	FQ <块地址>	普通命令
FLASH 编程	FW <FLASH 地址> <RAM 地址> <字节长度>	普通命令
内存比较	MC <地址 1> <地址 2> <字节长度>	普通命令
运行	GO <地址>	普通命令
解锁	UN <密码>	特殊命令
获取保护等级	AL	特殊命令
使能 PFP	PM <开关设置>	普通命令
软件复位	RS	普通命令
使能 NVM (FLASH)	NV <NVM选项>	普通命令

● 波特率设置

表 19-5 ISP 波特率设置命令

命令	BS <波特率> <停止位>
输入	波特率：9600 或 19200 或 38400 或 57600 或 115200 停止位：1 或 2
返回代码	CMD_SUCCESS 或 INVALID_BAUD_RATE 或 INVALID_STOP_BIT 或 INVALID_PARAM

说明	改变 ISP 通讯串口帧格式，包括波特率和停止位。串口起始位固定为 1，数据位固定为 8。新帧格式在返回 CMD_SUCCESS 后生效。
范例	“BS 9600 2” 设置串口波特率为9600bps，2个停止位。

● 回显

表 19-6 ISP 回显命令

命令	RD <开关设置>
返回代码	CMD_SUCCESS 或 INVALID_PARAM
说明	命令和数据回显。缺省为开。回显打开状态下，SoC 会将收到的命令和数据发回主机。
范例	“RD 0” 关闭回显。

● 写内存

表 19-7 ISP 写内存命令

命令	WM <地址> <字节长度> <模式>
输入	地址：起始地址，应该 32 位对齐； 字节数：应该是 4 的倍数； 模式：0 为串口，1 为并口
返回代码	CMD_SUCCESS 或 FM_MODE_ERROR 或 ADDR_NOT_ALIGN 或 COUNT_ERROR 或 COUNT_ERROR 或 ADDR_NOT_MAPPED 或 INVALID_PARAM
说明	向片内 SRAM 写入数据
范例	采用串口向0x10000300地址写入0x12345678： 1. ASCII码发送：“WM 268436224 4 0” 2. 二进制发送： 数据行：7e 04 78 56 34 12 7e 校验和行 7e ff 88 a9 cb ed 7e

● 读内存

表 19-8 ISP 读内存命令

命令	RM <地址> <字节长度> <模式>
输入	地址：读出地址，应该 32 位对齐； 字节数：应该是 4 的倍数； 模式：0 为串口，1 为并口
返回代码	CMD_SUCCESS 或 FM_MODE_ERROR 或 ADDR_NOT_ALIGN 或 COUNT_ERROR 或 COUNT_ERROR 或 ADDR_NOT_MAPPED 或 INVALID_PARAM
说明	读取 SoC 片内 SRAM 的内容
范例	“RM 268436224 4 0” 采用串口读取片内SRAM地址0x10000300的内容。

● Flash 页擦除

表 19-9 ISP Flash 页擦除命令

命令	FP <页地址> (FPGA 版本为 0 到 3071)
输入	页地址：0 到 1535 之间可选；
返回代码	CMD_SUCCESS 或 INVALID_PAGE 或 INVALID_PARAM
说明	擦除 SoC 片上 FLASH 指定块
范例	“FP 0” 擦除第0页的内容

● Flash 块擦除

表 19-10 ISP Flash 块擦除命令

命令	FS <块地址>
输入	块地址：0 到 47 之间可选；
返回代码	CMD_SUCCESS 或 INVALID_SECTOR 或 INVALID_PARAM
说明	擦除 SoC 片上 FLASH 指定块
范例	“FS 0” 擦除第0块的内容

- Flash 片擦除

表 19- 11 ISP Flash 片擦除命令

命令	FC
输入	无
返回代码	CMD_SUCCESS 或 INVALID_PARAM
说明	擦除 SoC 片上 FLASH 全部内容。
范例	“FC” 擦除片上FLASH全部内容

- Flash 块查空

表 19- 12 ISP Flash 块查空命令

命令	FQ <块地址>
命令	
输入	块地址：0 到 47 之间可选；
返回代码	CMD_SUCCESS 或 INVALID_SECTOR 或 INVALID_PARAM
说明	检查片上 FLASH 指定块的内容是否为空（擦除后未编程）
范例	“FQ 1” 检查第1块的内容是否为空

- Flash 编程

表 19- 13 ISP Flash 编程命令

命令	FW <FLASH 地址> <RAM 地址> <字节长度>
输入	FLASH 地址：要写入的 FLASH 目标地址 RAM 地址：源缓冲区所在的 SRAM 地址 字节长度：写入的字节数量。（如果字节长度不为 Flash 页的字节数，则该 Flash 也其余内容填充为 0）
返回代码	CMD_SUCCESS 或 COUNT_ERROR 或 SRC_ADDR_NOT_ALIGN 或 SRC_ADDR_NOT_MAPPED 或 DST_ADDR_NOT_ALIGN 或 DST_ADDR_NOT_MAPPED 或 INVALID_PARAM
说明	用于编程Flash 存储器。
范例	“FW 402653184 268436224 128” 将SRAM地址0x10000300开始的128字节复制到FLASH地址0x18000000 (ISP模式下, REMAP为3, 此时的FLASH映射地址)

- 内存比较

表 19- 14 ISP 内存比较命令

命令	MC <地址 1> <地址 2> <字节长度>
输入	地址1（DST）：要比较的内存区域1起始地址。应当与字对齐； 地址2（SRC）：要比较的内存区域2起始地址。应当与字对齐； 字节长度：待比较的字节数，应当为 4 的倍数；
返回代码	CMD_SUCCESS 或 COUNT_ERROR 或 SRC_ADDR_NOT_ALIGN 或 SRC_ADDR_NOT_MAPPED 或 DST_ADDR_NOT_ALIGN 或 DST_ADDR_NOT_MAPPED 或 COMPARE_ERROR 或 INVALID_PARAM

说明	该命令用来比较存储器两个区域的内容
范例	“MC 268436224 268436224 4” 将SRAM地址0x10000300的4个字节与SRAM地址0x10000300的4个字节进行比较

- 运行

表 19-15 ISP 内存比较命令

命令	GO <地址>
输入	地址：代码执行起始的 Flash 或 RAM 地址。该地址必须为 Thumb 地址
返回代码	CMD_SUCCESS 或 ADDR_NOT_THUMB 或 ADDR_NOT_MAPPED 或 INVALID_PARAM
说明	该命令用于执行位于RAM 或Flash 存储器当中的程序。一旦成功执行该命令，就有可能不再返回ISP 命令处理程序。
范例	“GO 5” 跳转到地址0x00000004处执行

- 解锁

表 19-16 ISP 解锁命令

命令	UN
输入	密码：32 位 16 进制数
返回代码	CMD_SUCCESS 或 INVALID_PASS 或 INVALID_PARAM
说明	该命令用于解锁ISP。
范例	“UN 567” 输入密码567解锁ISP

- 获取保密等级

表 19-17 ISP 获取保密等级命令

命令	AL
输入	无
返回代码	CMD_SUCCESS 或 INVALID_PARAM
说明	该命令用于获取SoC当前的保护等级
范例	“AL” 将返回SoC当前的保护等级

- 使能 PFPM

表 19-25 ISP 使能 PFPM 命令

命令	PM <开关设置>
输入	开关设置：0（关）或 1（开）
返回代码	CMD_SUCCESS 或 INVALID_PARAM
说明	该命令使能/不使能PFPM（并行编程模式）
范例	“PM 1” 将使能PFPM

- 软件复位

表 19-26 ISP 系统复位命令

命令	RS
输入	无
返回代码	CMD_SUCCESS 或 INVALID_PARAM
说明	该命令触发软件复位
范例	“RS”将触发软件复位

- 使能 NVM

表 19-27 ISP 使能 NVM 命令

命令	NV <NVM 选项>
输入	NVM 选项: 0 (Flash) 或 1 (保留)
返回代码	CMD_SUCCESS 或 INVALID_PARAM
说明	该命令使能Flash
范例	“NV 0” 将使能Flash写, 编程操作。

30.3.4 ISP 返回代码

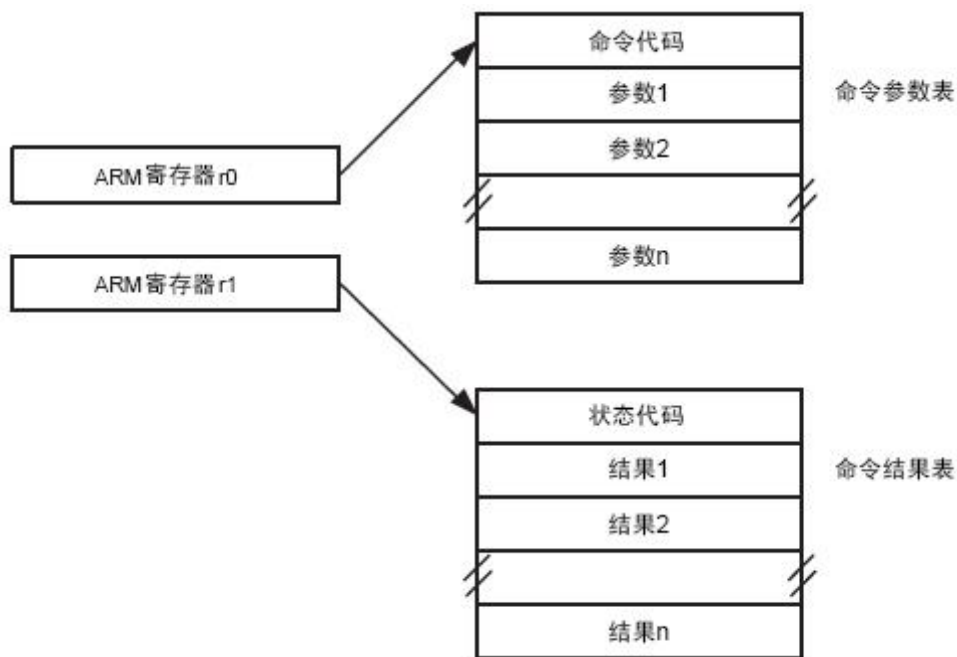
表 19- 18 ISP 返回码

返回码 (ASCII 码)	符号	说明
0	CMD_SUCCESS	成功执行命令。只有成功执行命令后, ISP 处理器才发送该代码
1	INVALID_COMMAND	无效命令
2	INVALID_PARAM	无效参数 (参数的 ASCII 码不为 0-9)
3	INVALID_BAUD_RATE	无效波特率
4	INVALID_STOP_BIT	无效停止位
5	ADDR_NOT_ALIGN	地址不是以字为边界
6	COUNT_ERROR	字节计数值不是 4 的倍数
7	ADDR_NOT_MAPPED	所访问的地址空间越界
8	INVALID_SECTOR/INVALID_PAGE	无效 SECTOR_NUM 或者 PAGE_NUM
9	SECTOR_NOT_BLANK	SECTOR 非空
10	SRC_ADDR_NOT_ALIGN	源地址不是以字为边界
11	SRC_ADDR_NOT_MAPPED	所访问的源地址空间越界
12	DST_ADDR_NOT_ALIGN	目的地址不是以字节为边界
13	DST_ADDR_NOT_MAPPED	所访问的目的地址空间越界
14	COMPARE_ERROR	比对错误
15	FM_MODE_ERROR	操作内存模式错误
16	ADDR_NOT_THUMB	地址不为 Thumb 指令
17	INVALID_PASS	错误密码

30.4 在应用编程 (IAP)

对于在应用编程, 应当通过寄存器r0中的字指针来调用IAP程序, 该字指针指向含有命令代码和参数的存储器 (RAM)。IAP命令的结果返回到寄存器r1所指向的结果表。用户可以把寄存器r0 和r1 中的指针赋予相同的值, 如此便能将命令表复用来存放结果。参数表应当大到足够保存所有的结果以防结果的数目大于参数的数目。参数传递见图18-2。参数和结果的数目根据IAP命令而有所不同。“Flash编程”命令参数的最大数目为4。结果的数目为1。命令处理器在接收到一个未定义的命令时发送状态代码INVALID_COMMAND。IAP程序是Thumb代码, 驻留在地址0x1800_1c01。

图18-2 IAP参数传递



30.4.1 IAP 命令

表 19-19 IAP 命令

IAP 命令	命令代码	说明
Flash 页擦除	0x50	参看 ISP 章节
Flash 块擦除	0x51	参看 ISP 章节
flash 片擦除	0x52	参看 ISP 章节
Flash 块查空	0x53	参看 ISP 章节
Flash 编程	0x58	参看 ISP 章节
使能 NVM	0x5a	参看 ISP 章节
模拟软件复位	0x5b	参看 ISP 章节

30.4.2 IAP 使用

IAP用于以下两个方面：

- ◎ 在线升级（更新 FLASH）；
- ◎ 客户数据信息更新；

在线升级时需要对Flash进行擦除/写操作。Flash擦写操作持续4ms左右，会增加在此期间发生的中断的处理延迟。

一种 IAP 实现方法：

用户要实现在线升级时，需要在软件设计中增加一个 IAP 升级的程序段。这段程序实现通过通讯口（如 UART）从远程主机接收程序或数据，并使用 SoC 提供的 IAP 接口，将这些程序或数据写入到 SoC 内部的 FLASH 中。

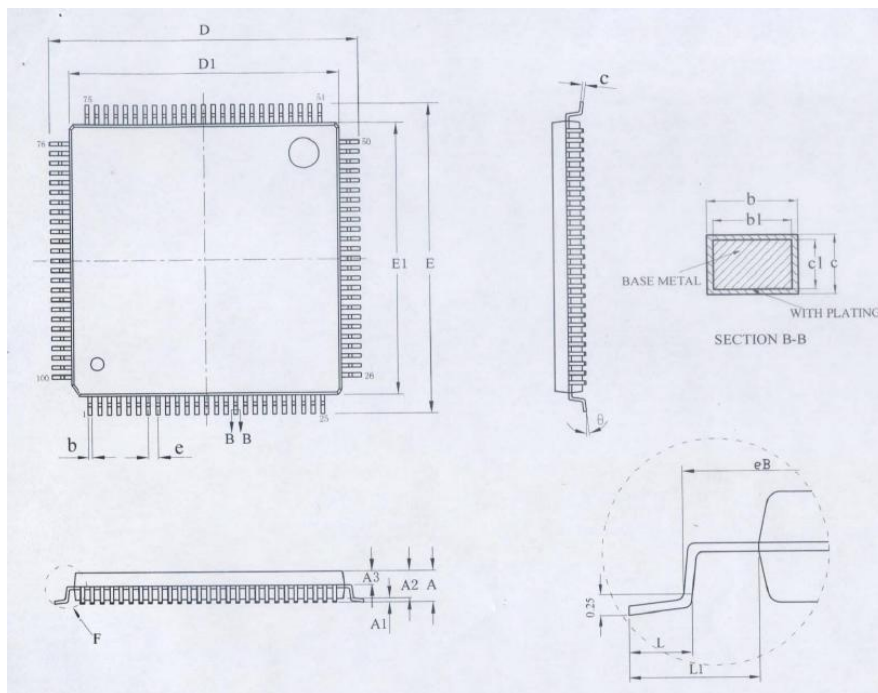
30.5 量产平台

Renergy 提供了多种编程手段对芯片进行程序编程和选项编程，具体可参见《RN821x_RN831x 应用笔记 008-编程平台使用说明》。

31 封装尺寸及焊接条件

31.1 封装尺寸

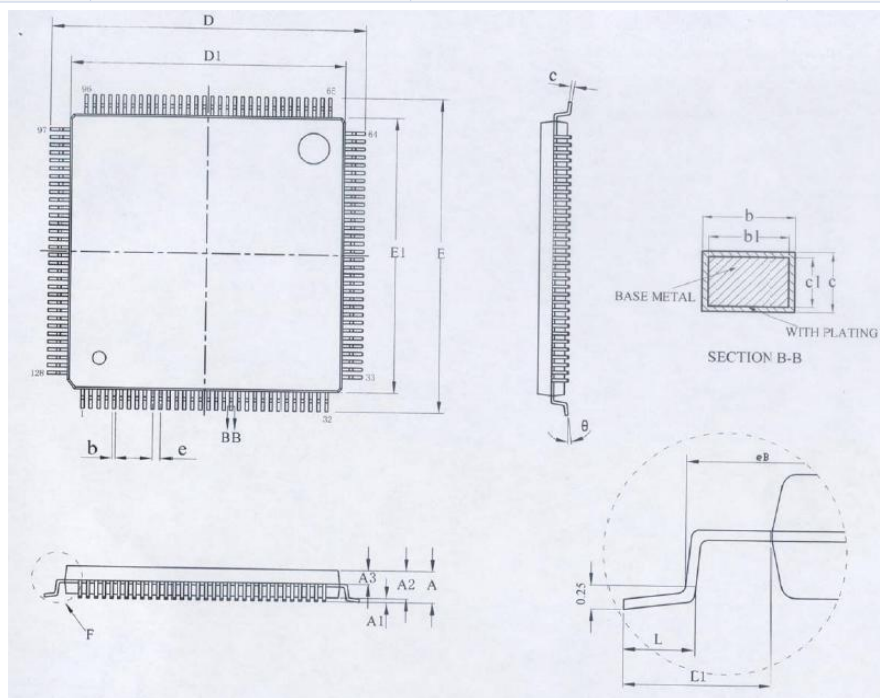
LQFP100L (1414×1.4)		14.00×14.00×1.40	e=0.50
------------------------	---	------------------	--------



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	---	---	1.6
A1	0.05	---	0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.19	---	0.27
b1	0.18	0.20	0.23
c	0.13	---	0.18
c1	0.12	0.13	0.14
D	15.80	16.00	16.20
D1	13.90	14.00	14.10
E	15.80	16.00	16.20
E1	13.90	14.00	14.10
eB	15.05	---	15.35
e	0.50BSC		
L	0.45	---	0.75

L1	1.00BSC		
θ	0	----	7°

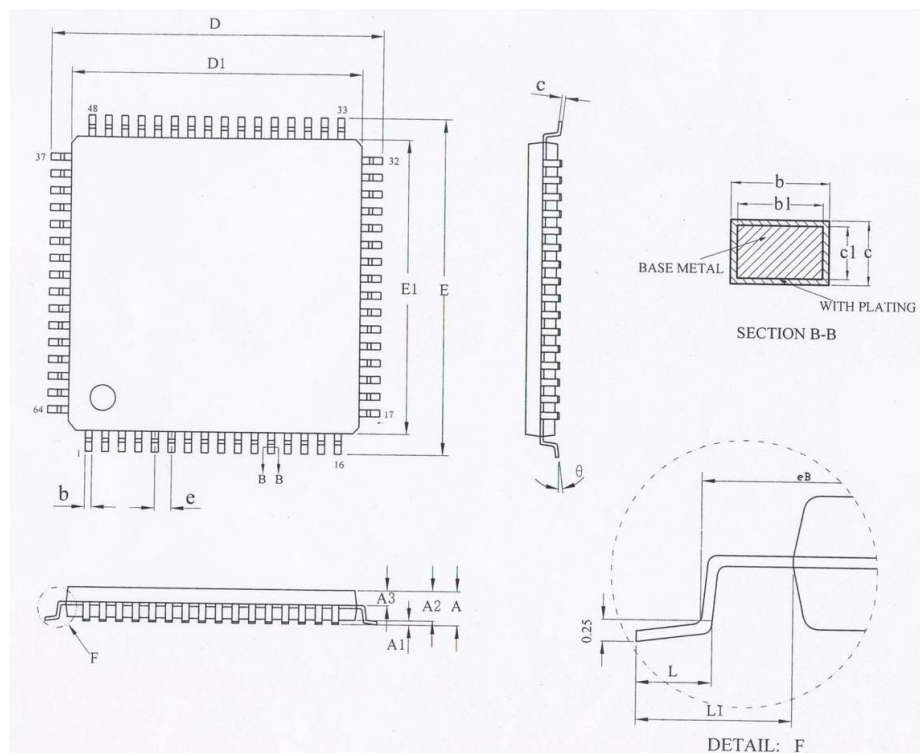
LQFP128L (1414×1.4)		14.00×14.00×1.40	e=0.40
------------------------	---	------------------	--------



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	---	---	1.6
A1	0.05	---	0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.15	---	0.23
b1	0.14	0.16	0.19
c	0.13	---	0.18
c1	0.12	0.13	0.14
D	15.80	16.00	16.20
D1	13.90	14.00	14.10
E	15.80	16.00	16.20
E1	13.90	14.00	14.10
eB	15.05	---	15.35
e	0.40BSC		
L	0.45	---	0.75
L1	1.00BSC		

θ	0	-----	7°
----------	---	-------	-------------

LQFP64L (0707×1.4)		7.00×7.00×1.40	e=0.40
-----------------------	---	----------------	--------

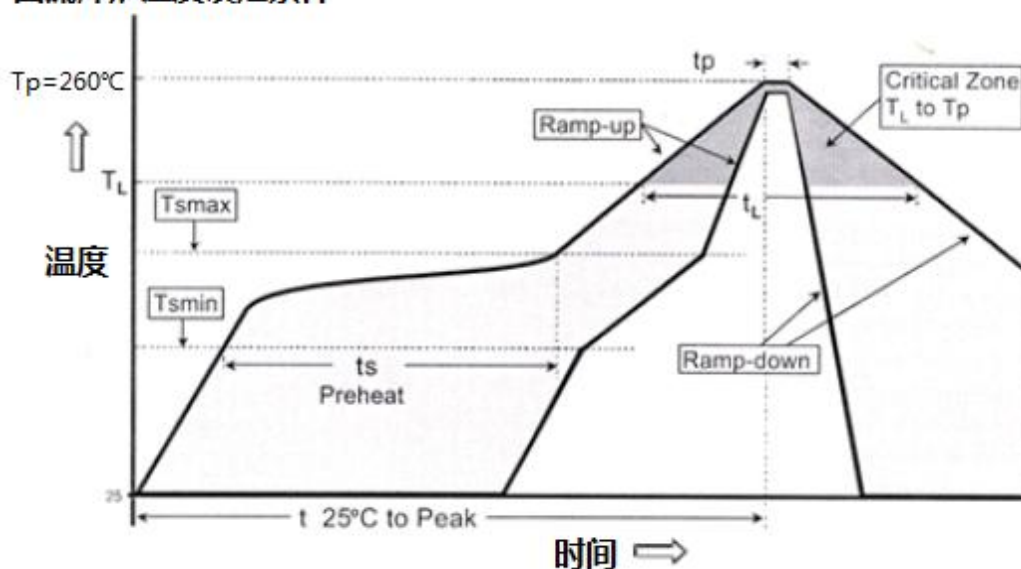


SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	---	---	1.6
A1	0.05	---	0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.17	---	0.25
b1	0.16	0.18	0.20
c	0.13	---	0.18
c1	0.12	0.127	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
eB	8.10	---	8.25
e	0.40BSC		
L	0.40	---	0.65

L1	1.00BSC		
θ	0	-----	7°

31.2 回流焊炉的温度设定条件

回流焊炉温度设定条件

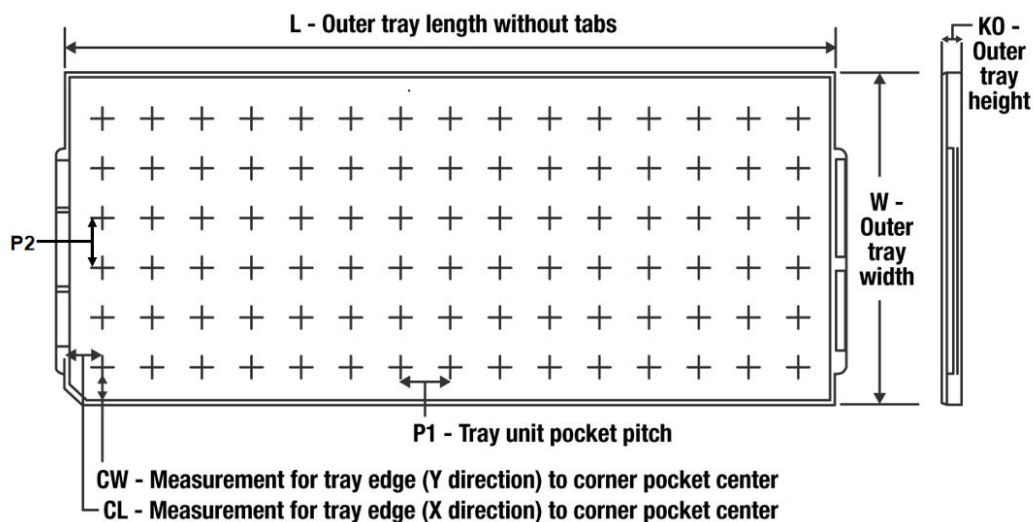


回流焊炉的温度设定曲线

分布图特征	值
保温温度 T_L	217°C
峰值温度 T_p	260°C
平均倾斜上升率(T_L to T_p)	最大 3°C/秒
预热	
最小温度 (T_{smin})	150°C
最大温度 (T_{smax})	200°C
时间 (最小-最大) (t_s)	60-180 秒
T_{smax} - T_L 倾斜上升率(T_{smax} to T_L)	最大 3°C/秒

32 包装信息

32.1 托盘规格



封装形式	每盘数量	行列分布	L (mm)	W (mm)	K0 (mm)	P1 (mm)	P2 (mm)	CL (mm)	CW (mm)	每盒数量	每箱数量
LQFP64(0707*1.4)	250	10 x 25	315	135.9	7.62	12.00	12.60	11.10	11.25	2500	15000
LQFP100(1414*1.4)	90	6 x 15	315	135.9	7.62	20.30	21.00	15.40	15.45	900	5400
LQFP128(1414*1.4)	90	6 x 15	315	135.9	7.62	20.30	21.00	15.40	15.45	900	5400