

锐能微 MCU
RN8318_RN8615_RN8613_RN8611 V2 版
用户手册
Rev2.2

深圳市锐能微科技有限公司

修改记录

版本号	发布日期	主要更改内容
V1.0	2023-2-14	新建
V1.1	2023-5-5	1、增加 VBAT 域管脚外接电压不能高于 VBAT 电压值说明 2、增加 DSP 模块 3、SPI 管脚说明修订 4、增加 IIC 分频系数配置与频率关系 5、增加系统时钟运行在 32.768KHz 时, uart 口波特率及 38K 红外调制说明
V1.2	2023-8-11 2023-9-18	1、增加液晶驱动管脚作为输出管脚使用时, 需增加外部上拉的说明 2、管脚复用的模拟输入增加具体模拟应用名称说明 (CMP/AIN) 3、对 VCC 域的 RTCOUT 增加注意事项说明 4、RN8613 引脚图增加 COM2 管脚; RN8611 引脚图 P34 复用 CF1 名称修为 D2F_OUT1 5、增加 RTC 用户温补模式 0 和 1 应用注意事项 6、SAR 做 ADC 和 VBAT 电压检测步骤增加判断上次测试是否完成的判断 7、增加 RTC 秒脉冲和秒寄存器关系说明 8、P30 管脚增加复用为 TX4 9、P32 管脚增加复用为 KEY5 描述 10、修正功率输入寄存器描述的错别字 11、增加包装信息
V1.3	2023-9-26	1、修订 RN8613 及 RN8611 资源 2、修改手册名称及产品型号名称
Rev1.4	2023-12-29	1、修改 RN8613/RN8615/RN8611 的 INTC 资源数 2、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 3、增加 P141 的管脚说明, 默认上拉 5K 电阻开启, 不能直接接地, 且该引脚与其他引脚不同, 输入/输出模式上拉配置均有效。其他引脚上拉配置, 只在输入模式有效。改管脚只在 RN8611 型号上存在, 其他型号不存在该管脚。 4、RN8613/RN8615 的 27 脚删除 P140 功能, 改为 NC 5、SIMP_TCx 的描述修改, 同步修改中断分配序号 6、SYS_PD 中 BOR5 的描述, 删上电阈值配置的描述, 因上电阈值会被复位, 配置无效 7、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 8、IOCNT 转发功能单独成章, 并增加波形分析; 修改 PF/QF/SF 字样为 IOCNT
Rev1.5	2024-1-8	1、RTC 章节 88/8C 地址寄存器名称(由 RTC->IOMODE、RTC->IOWEN 改为 RTC->IOEN,RTC->IOMODE)及描述修订 2、统一产品版本说明 3、勘误湿敏等级缩写为 MSL 4、增加 LCDVx 管脚应用说明
Rev1.6	2024-3-30	1、错误修订: RN8613 引脚图, 第 50 51 脚, 删 NC 字样 2、错误修订: RN8613 引脚图, 第 20 56 57 58 59 92 93 94 脚, 修改 TCI 为 TCIN

		3、错误修订：RN8613 引脚图，第 80 脚，LVDIN0 修改为 LVDIN 4、错误修订：RN8613 引脚图，第 89 脚，增加功能 P122 和 COM2 5、错误修订：引脚定义 P55 功能 TC1_P[0] 改为 TC1_P[0] 6、错误修订：RN8613 的 FLASH 大小改为 256K、 7、错误修订：引脚图、引脚定义说明及定时器输入源内的 PF\QF\SF 修改为 IOCNTx_OUTx，与管脚说明章节保持一致 8、增加描述：GPIO 章节概述增加上/下拉电阻说明
Rev1.7	2024-4-11	1、规格修改：重新开放 RN8613/RN8615 型号 P140 管脚功能 2、优化描述：D2F 高频脉冲常数寄存器 HFCONST3 的计算公式与三相和单相计量芯片的 HFCONST1 公式保持统一，并优化描述结构，使 HFCONST 公式和对接应用中的 HFCONST 设置便于理解。
Rev1.8	2024-5-13	1、错误修订：由于实测高速翻转对测温无影响，删除了 Rev1.4 更新记录第 2 条增加的“具有模拟测量的 AIN 功能管脚，不能使用高速翻转功能”描述。 2、增加内容：4.3.5 EEPROM 章节 3、增加内容：MCU 产品型号资源对照表增加 EEPROM 资源 4、优化内容：合并 MCU 产品型号列表和资源对照表 5、增加内容：P140\P141\P142 管脚上增加不同版本芯片的状态及应用说明
Rev1.9	2024-6-24	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。 2、错误修改：DSP 中 DMA_SRBADR 等寄存器描述有效位数修正为 15 位 3、增加描述：增加 RTC 温补和结果输出寄存器 4、优化描述：统一用户手册名称 5、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 6、错误修改：LOSC_FAIL 位修改为可读可写
Rev2.0	2024-9-29	1、修订 SAR_STATUS 的 bit0 清零，要求至少等待 30us 2、修订 RTC 工作范围，在高温 85°C 下为 2.0V~5.5V 3、错误修订：删除 P36 P37 管脚在 V2 版上支持输出功能的说明 4、增加描述：RTC 章节中增加测温启动和温度寄存器 3 说明 5、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 6、增加描述：UARTx_STA 寄存器中接收标志增加读后清零说明
Rev2.1	2025-01-03	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：开放 MOD1 中 DSP APB 时钟位 4、增加描述：增加 UARTx_DMA_CTRL 寄存器位描述 5、增加描述：LCD_CTRL.EN 及 LCD_STATUS.BUSY 位增加低功耗处理流程描述 6、增加描述：增加 CP4 的描述 7、增加描述：增加 WDT 配置寄存器描述 8、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

		9、增加描述：引脚定义说明内，VREF 管脚描述，要求外接 0.22uF 电容 10、增加描述：P00~P04,P44~P45 管脚增加输入电压范围说明 11、增加描述：增加 LCDVxx 管脚应用描述
Rev2.2	2025-03-12	1、错误修订：修订寄存器描述表格中芯片唯一码寄存器地址 2、错误修订：SIMP_TC 中 LOAD 和 VAL 寄存器位宽修改为 15:0 3、增加描述：增加 UART 波特率配置说明 4、错误修订：VCC 域休眠功耗由 5.5uA 增大 1.2uA 到 6.7uA；休眠功耗由 7.3uA 增大 1.2uA 到 8.5uA； 5、错误修订：bitband 支持范围

目录

REV2.2	1
修改记录	2
目录	5
1 概述	16
1.1 简介	16
1.2 产品特点	16
1.3 MCU 产品型号资源对照表	18
1.4 实现架构	19
1.5 管脚排列	20
1.5.1 RN8318 LQFP128L 引脚图	20
1.5.2 RN8613/8615 LQFP100L 引脚图	21
1.5.3 RN8611 LQFP64L 引脚图	22
1.5.4 引脚类型说明	23
1.5.5 引脚定义说明	23
1.5.6 硬件设计规则	35
1.6 IO 口功能框图	36
1.6.1 PABULD3	36
1.6.2 PABUS3	37
1.6.3 PBDS3	38
1.6.4 PBDG3	39
1.6.5 PBULD3/6	40
1.6.6 PBUS6	41
1.6.7 PIUX	42
2 电气特性	43
2.1 性能参数	43
2.2 可靠性参数	45
3 系统控制	46
3.1 电源域划分	46
3.2 时钟源	46
3.3 时钟切换	47
3.4 MCU 的低功耗模式	49
3.5 复位	50
3.5.1 外部 PIN 复位	50
3.5.2 上下电复位	50
3.5.3 软件复位	50
3.5.4 看门狗复位	50
3.6 掉电处理	50
3.7 寄存器描述	50
3.7.1 系统 OSC 控制寄存器 OSC_CTL1(0x00)	51

3.7.2 系统模式设置寄存器 SYS_MODE(0x04).....	52
3.7.3 系统掉电控制寄存器 SYS_PD (0x08).....	52
3.7.4 系统 OSC 控制寄存器 2OSC_CTL2(0x10).....	54
3.7.5 系统复位寄存器 SYS_RST(0x14).....	55
3.7.6 系统映射控制寄存器 SYS_MAPCTL(0x18).....	56
3.7.7 模块使能 0 寄存器 MOD0_EN (0x1c).....	56
3.7.8 模块使能 1 寄存器 MOD1_EN(0x20).....	58
3.7.9 INTC 使能寄存器 INTC_EN(0x24).....	59
3.7.10 KBI 使能寄存器 KBI_EN(0x28).....	59
3.7.11 器件 ID 寄存器 CHIP_ID(0x2C).....	60
3.7.12 系统控制密码寄存器 SYS_PS(0x30).....	60
3.7.13 红外配置寄存器 IRFR_CTL (0x34).....	61
3.7.14 时钟校正配置寄存器 TRIM_CFG1(0x78).....	61
3.7.15 时钟校正启动寄存器 TRIM_START(0x7C).....	61
3.7.16 DMA 优先级配置寄存器 DMA_PRI(0x80).....	62
3.7.17 芯片唯一码寄存器 0 FAB_UID0(0xF0).....	63
3.7.18 芯片唯一码寄存器 1 FAB_UID1(0xF4).....	63
4 处理器架构	64
4.1 概述	64
4.2 CORTEX-M0 处理器	64
4.2.1 中断配置	65
4.3 MCU 存储映射	67
4.3.1 MCU 存储映射	68
4.3.2 存储重映射	69
4.3.3 Bitband 功能	69
4.3.4 SRAM	70
4.3.5 FLASH	70
4.3.6 EEPROM	70
4.4 中断应用	71
5 RTC	72
5.1 概述	72
5.2 特点	72
5.3 寄存器描述	72
5.3.1 RTC 寄存器组	75
5.3.2 二次分段补偿寄存器组	85
5.3.3 开盖检测寄存器组	86
5.3.1 二次曲线补偿寄存器组	88
5.3.2 四次曲线补偿寄存器组	90
5.3.3 结果输出寄存器组	93
5.4 RTC 时钟读写步骤	95
5.5 RTC 校准步骤	95
5.6 RTC 定时器操作步骤	95

6 WDT	97
6.1 概述	97
6.2 看门狗定时器的配置	97
6.3 寄存器描述	98
6.3.1 WDT_EN (0x0)	98
6.3.2 WDT_CTRL (0x04)	98
6.3.3 WDT_PASS (0x08)	99
6.3.4 WDT_HALT (0x14)	99
6.3.5 WDT_STBY (0x18)	100
6.4 WDT 操作步骤	100
7 LCD	100
7.1 概述	100
7.1.1 扫描时钟频率	101
7.1.2 闪烁模式	101
7.1.3 LCD 驱动波形	102
7.1.4 偏置电压	107
7.1.5 LCD BUFFER 映射	108
7.2 寄存器描述	108
7.2.1 LCD 控制寄存器 LCD_CTL(0x00)	108
7.2.2 LCD 状态寄存器 LCD_STATUS(0x04)	109
7.2.3 LCD 时钟控制寄存器 LCD_CLKDIV(0x08)	110
7.2.4 LCD 闪烁控制寄存器 LCD_BLINK(0x0c)	110
7.2.5 LCD 闪烁控制寄存器 LCD_PS(0x10)	111
7.2.6 LCD 内部电阻串控制寄存器 LCD_RESCTL(0x14)	111
7.2.7 LCD 数据寄存器 LCD_BUF[i] (8 位寄存器)	112
7.3 LCD 操作步骤	112
8 定时器	114
8.1 概述	114
8.2 功能框图	115
8.3 寄存器描述	115
8.4 典型应用	120
8.4.1 自动运行模式, 定时功能	120
8.4.2 输入捕获模式, 脉宽测量功能	121
8.4.3 比较输出模式, 方波输出功能	121
8.4.4 比较输出模式, PWM 输出功能	122
8.4.5 从模式, 外部清零和门控功能	124
8.5 操作步骤	124
9 模拟外设	125
9.1 特点	125
9.2 寄存器	125
9.2.1 ADC 控制寄存器 SAR_CTL(0x00)	126
9.2.2 SAR-ADC 启动寄存器 SAR_START(0x04)	127

9.2.3 SAR-ADC 状态寄存器 SAR_STATUS(0x08)	127
9.2.4 ADC 数据寄存器 SAR_DAT(0x0c)	128
9.2.5 LVD 控制寄存器 LVD_CTL(0x10)	128
9.2.6 LVD 状态寄存器 LVD_STAT(0x14)	128
9.2.7 ADC 控制寄存器 1 SAR_CTL1(0x18) (新增)	129
9.2.8 ADC 数据寄存器 2 SAR_DAT2(0x1C) (新增)	129
9.2.9 AIN 功能配置寄存器 ANA_PAD	129
9.2.10 局部复位寄存器 ANA_RST	130
9.3 ADC 电压检测步骤	130
9.4 VBAT 电压检测	130
9.5 低电压检测应用	131
10 GPIO	132
10.1 概述	132
10.2 寄存器描述	132
10.2.1 PA 口模式寄存器 PMA (输入或者输出) (0x00)	134
10.2.2 PA 口数据寄存器 PA (0x04)	134
10.2.3 PA 口复用 0 寄存器 PCA0 (0x08) (修改)	134
10.2.4 PA 口复用 1 寄存器 PCA1 (0x0C) (修改)	136
10.2.5 PA 口上拉选择寄存器 PUA (0x10)	138
10.2.6 PA 口输入模式配置寄存器 PIMA (0x14)	138
10.2.7 PA 口输入使能寄存器 PIEA (0x18)	138
10.2.8 PB 口模式寄存器 PMB (输入或者输出) (0x1C)	139
10.2.9 PB 口数据寄存器 PB (0x20)	139
10.2.10 PB 口复用寄存器 PCB (0x24)	140
10.2.11 PB 口复用寄存器 2 PCB2 (0x4CH) (修改)	141
10.2.12 PB 口上下拉选择寄存器 PUB (0x28)	144
10.2.13 PB 口输入模式寄存器 PIMB (0x2C)	144
10.2.14 PB 口输入使能寄存器 PIEB (0x30)	145
10.2.15 PC 口模式寄存器 PMC (输入或者输出) (0x34)	145
10.2.16 PC 口数据寄存器 PC (0x38)	145
10.2.17 PC 口复用寄存器 PCC (0x3C) (修改)	146
10.2.18 PC 口上下拉选择寄存器 PUC (0x40)	147
10.2.19 PC 口输入使能寄存器 PIEC (0x44)	147
10.2.20 PC 口输入模式寄存器 PIMC (0x48)	147
10.2.21 PD 口模式寄存器 PMD (0x50)	148
10.2.22 PD 口数据寄存器 PD (0x54)	148
10.2.23 PD 口复用寄存器 PCD (0x58)	148
10.2.24 PD 口上下拉选择寄存器 PUD (0x5C)	149
10.2.25 PD 口输入使能寄存器 PIED (0x84)	150
10.2.26 PD 口输入模式寄存器 PIMD (0x88)	150
10.2.27 SEGCOM 口复用寄存器 PCE (0x60)	150
10.2.28 PA 口数据置位寄存器 PASET (0x64)	151
10.2.29 PA 口清零置位寄存器 PACLR (0x68)	151
10.2.30 PB 口数据置位寄存器 PBSET (0x6C)	151

10.2.31 PB 口清零置位寄存器 PBCLR (0x70)	152
10.2.32 PC 口数据置位寄存器 PCSET (0x74)	152
10.2.33 PC 口清零置位寄存器 PCCLR (0x78)	153
10.2.34 PD 口数据置位寄存器 PDSET (0x7C)	153
10.2.35 PD 口清零置位寄存器 PDCLR (0x80)	153
10.3 GPIO 操作步骤	154
11 脉冲转发功能 IOCNT	154
11.1 功能特性	154
11.2 复用关系	154
11.3 功能描述	154
11.4 波形分析	155
11.4.1 输入信号类型	155
11.4.2 脉冲计数	155
11.4.3 脉冲分频	158
11.4.4 脉冲电平反向输出	162
11.5 寄存器描述	162
11.5.1 脉冲转发配置寄存器 0~2 (0x100~0x108)	162
11.5.2 脉冲转发控制寄存器 IOCNT_CTL (0x10C)	163
11.6 操作步骤	164
11.6.1 高电平脉宽有效	164
11.6.2 低电平脉宽有效	164
12 外部中断控制器	165
12.1 概述	165
12.2 寄存器描述	165
12.2.1 INTC_CTL	165
12.2.2 INTC_MODE	165
12.2.3 INTC_MASK	166
12.2.4 INTC_STA	167
13 KBI	168
13.1 特性	168
13.2 寄存器描述	168
13.2.1 控制寄存器 (0x0)	168
13.2.2 选择寄存器 (0x4)	168
13.2.3 数据寄存器 (0x8)	168
13.2.4 屏蔽寄存器 (0xC)	169
13.3 KBI 操作步骤	169
14 UART	170
14.1 概述	170
14.2 寄存器描述	170
14.2.1 UART 控制寄存器 (0x00)	171
14.2.2 UART 波特率配置寄存器 (0x4)	172

14.2.3 UART 状态指示寄存器 (0x8)	172
14.2.4 UART 发送数据寄存器 (0xC)	173
14.2.5 UART 接收数据寄存器 (0x10)	174
14.2.6 UART 波特率小数分频配置寄存器 (0x14)	174
14.2.7 UART DMA 控制寄存器 (0x18)	174
14.2.8 UART DMA 发送起始地址寄存器 (0x1C)	175
14.2.9 UART DMA 接收起始地址寄存器 (0x20)	175
14.2.10 UART DMA 发送长度寄存器 (0x24)	175
14.2.11 UART DMA 接收长度寄存器 (0x28)	175
14.2.12 UART DMA 当前发送地址寄存器 (0x2C)	175
14.2.13 UART DMA 当前接收地址寄存器 (0x30)	175
14.2.14 UART DMA 中断使能寄存器 (0x34)	175
14.2.15 UART DMA 中断标志寄存器 (0x38)	176
14.2.16 UART DMA 接收超时配置寄存器 (0x3C)	176
14.3 UART 数据接收及发送操作步骤	176
15 ISO7816	178
15.1 概述	178
15.2 寄存器描述	178
15.2.1 控制寄存器 (0x0)	178
15.2.2 ISO7816 控制寄存器 1 (0x04)	180
15.2.3 ISO7816_CLK (0x08)	182
15.2.4 ISO7816 波特率系数 0 寄存器 (0x0c)	182
15.2.5 ISO7816 波特率系数 1 寄存器 (0x10)	182
15.2.6 ISO7816 状态 0 寄存器 (0x14)	183
15.2.7 ISO7816 状态 1 寄存器 (0x18)	184
15.2.8 ISO7816 数据 0 寄存器 (0x1C)	186
15.2.9 ISO7816 数据 1 寄存器 (0x20)	186
15.3 7816 与 ESAM 通讯操作步骤	186
15.4 7816 与卡通讯操作步骤	186
16 IIC 接口	188
16.1 概述	188
16.2 寄存器描述	188
16.2.1 控制寄存器 (0x0)	188
16.2.2 时钟配置寄存器 (0x4)	189
16.2.3 状态指示寄存器 (0x8)	190
16.2.4 从设备地址寄存器 (0xC)	191
16.2.5 收发数据寄存器 (0x10)	191
17 SPI 接口	192
17.1 概述	192
17.2 复用关系	192
17.3 功能描述	192
17.4 寄存器描述	193

17.4.1 SPI 控制寄存器 (0x0)	194
17.4.2 SPI 状态寄存器 (0x4)	195
17.4.3 SPI 数据发送寄存器 (0x8)	196
17.4.4 SPI 数据接收寄存器 (0xC)	196
17.4.5 SPI 默认发送数据寄存器 (0x10)	196
17.4.6 SPI DMA 控制寄存器 (0x14)	196
17.4.7 SPI DMA 发送起始地址寄存器 (0x18)	197
17.4.8 SPI DMA 接收起始地址寄存器 (0x1C)	197
17.4.9 SPI DMA 发送长度寄存器 (0x20)	197
17.4.10 SPI DMA 接收长度寄存器 (0x24)	197
17.4.11 SPI DMA 当前发送地址寄存器 (0x28)	197
17.4.12 SPI DMA 当前接收地址寄存器 (0x2C)	198
17.4.13 SPI DMA 中断使能寄存器 (0x30)	198
17.4.14 SPI DMA 中断标志寄存器 (0x34)	198
18 电能积分单元 D2F	200
18.1 概述	200
18.2 特点	200
18.3 寄存器描述	200
18.3.1 D2F 高频脉冲常数寄存器 HFCONST3(0x00)	200
18.3.2 D2F 中断使能寄存器 IE(0x04)	202
18.3.3 D2F 中断标志寄存器 IF(0x08)	202
18.3.4 D2F 配置寄存器 CFG(0x0C)	203
18.3.5 D2F 功率输入寄存器 D2FP(0x10~0x3C)	203
18.3.6 D2F 自定义能量寄存器 D2FE(0x40~0x6C)	203
19 安全密码加速器 SEA	203
19.1 AES 硬件加速单元	204
19.1.1 特点	204
19.1.2 时耗信息	204
19.2 ECC 硬件加速单元	205
19.2.1 特点	205
19.2.2 时耗信息	205
19.3 RSA 硬件加速单元	205
19.3.1 特点	205
19.3.2 时耗信息	205
19.4 HASH 硬件加速单元	206
19.4.1 特点	206
19.4.2 时耗信息	206
19.5 TRNG 真随机数发生器	206
19.5.1 特点	206
19.5.2 时耗信息	206
19.6 软件配置流程	206
20 简单定时器 SIMP_TC	208

20.1 概述	208
20.2 功能描述	208
20.3 寄存器描述	208
20.3.1 SIMP_TCx 控制寄存器 CTRL (0x0)	208
20.3.2 SIMP_TCx 目标计数值寄存器 LOAD (0x4)	209
20.3.3 SIMP_TCx 当前计数值寄存器 VAL (0x8)	209
21 内存搬运单元 M2M	209
21.1 特点	209
21.2 功能描述	209
21.2.1 输入输出数据有效位宽配置	209
21.2.2 输出地址无效时可配 dummy 值	210
21.2.3 输出数据反序放置	210
21.2.4 源和目标地址可配置, 输入数据长度可配置	210
21.2.5 数据按地址反序排列	210
21.3 实例说明	210
21.3.1 输入数据 4Bytes 有效, 输出 3Bytes 有效	210
21.3.2 输入数据 2Bytes 有效, 输出 4Bytes 有效	210
21.3.3 输入数据按 byte 反序存放 (输入 4Bytes 有效, 输出 2Bytes 有效)	211
21.3.4 输出数据按地址反序排列	211
21.4 搬运速度	211
21.5 寄存器描述	211
21.5.1 M2M_MODE (0x0)	211
21.5.2 M2M_CTL (0x4)	212
21.5.3 M2M_DUMMY (0x8)	212
21.5.4 M2M_SADDR (0xC)	212
21.5.5 M2M_DADDR (0x10)	212
21.5.6 M2M_ILEN (0x14)	213
21.5.7 M2M_IE (0x18)	213
21.5.8 M2M_IF (0x1C)	213
21.6 软件使用流程	213
22 DSP 核	213
22.1 特点	215
22.2 计算引擎基本原理	215
22.2.1 单精度浮点数表示	215
22.2.2 特殊数值	215
22.2.3 浮点数舍入处理	215
22.2.4 IEEE754 标准 rounding 模式	216
22.2.5 本芯片的 rounding 模式	216
22.2.6 整数转浮点数原理	216
22.2.7 浮点数转整数原理	217
22.2.8 浮点数乘法原理	217
22.2.9 浮点数加法原理	217
22.2.10 蝶形运算原理	217

22.2.11 IIR 滤波器原理	218
22.2.12 FIR 滤波器原理	218
22.2.13 线性插值	219
22.2.14 Lagrange 插值	219
22.3 运算指令	220
22.3.1 整数转浮点数 (int2fp/int2fp_dma)	220
22.3.2 浮点数转整数 (fp2int/fp2int_dma)	221
22.3.3 浮点数乘法 (fp_mult)	222
22.3.4 浮点数加法 (fp_add)	222
22.3.5 浮点数减法 (fp_sub)	223
22.3.6 浮点数乘加运算 (fp_mlad)	223
22.3.7 浮点数蝶形运算 (单次) (btfy/btfy_dma)	224
22.3.8 正弦余弦计算 (sin_cos)	226
22.3.9 均方根和反正切 (fp_sqrt/ fp_atan)	226
22.3.10 浮点数除法器 (fp_div)	227
22.3.11 IIR 滤波器	227
22.3.12 FIR 滤波器	229
22.3.13 线性插值	230
22.3.14 拉格朗日插值	231
22.4 实现说明	232
22.4.1 整数转浮点数	232
22.4.2 浮点数转整数	232
22.4.3 FFT 说明	232
22.4.4 Cordic 说明	234
22.5 寄存器	235
22.5.1 寄存器列表	235
22.5.2 MAC_CTL0 (0x0)	236
22.5.3 MAC_CTL1 (0x04)	237
22.5.4 MAC_CTL2 (0x08)	238
22.5.5 MAC_IN0 (0x0C)	238
22.5.6 MAC_IN1 (0x10)	238
22.5.7 MAC_IN2 (0x14)	238
22.5.8 MAC_IN3 (0x18)	238
22.5.9 MAC_IN4 (0x1C)	239
22.5.10 MAC_IN5 (0x20)	239
22.5.11 MAC_OUT0 (0x24)	239
22.5.12 MAC_OUT1 (0x28)	239
22.5.13 MAC_OUT2 (0x2C)	239
22.5.14 MAC_OUT3 (0x30)	239
22.5.15 DIV_IN0 (0x34)	239
22.5.16 DIV_IN1 (0x38)	240
22.5.17 DIV_OUT0 (0x3C)	240
22.5.18 DMA_SRBADR (0x40)	240
22.5.19 DMA_SIBADR (0x44)	240

22.5.20 DMA_PRBADR (0x48)	240
22.5.21 DMA_PIBADR (0x4C)	240
22.5.22 DMA_TRBADR (0x50)	241
22.5.23 DMA_TIBADR (0x54)	241
22.5.24 DMA_LEN (0x58)	241
22.5.25 DSP_IE (0x5C)	241
22.5.26 DSP_FLG (0x60)	242
22.5.27 ALU_STA0 (0x64)	243
22.5.28 ALU_STA1 (0x68)	243
22.5.29 CRD_CTL (0x6C)	243
22.5.30 CRD_XIN (0x70)	244
22.5.31 CRD_YIN (0x74)	244
22.5.32 CRD_AMP (0x78)	244
22.5.33 CRD_PHASE (0x7C)	244
22.5.34 CRD_ANGLE (0x80)	244
22.5.35 CRD_COSINE (0x84)	244
22.5.36 CRD_SINE (0x88)	245
22.5.37 CRD_IE (0x8C)	245
22.5.38 CRD_FLG (0x90)	245
22.5.39 INTP_LEN (0x94)	245
22.5.40 INTP_LOC (0x98)	245
22.5.41 INTP_STEP (0x9C)	246
22.6 软件操作流程	246
22.6.1 完整方案说明:	246
22.6.2 操作流程:	246
23 选项字节	247
23.1 芯片保护设置	247
23.2 WDT 设置	248
23.3 RTC 设置	248
24 编程支持	250
24.1 概述	250
24.2 FLASH 保护机制	250
24.3 在系统编程 (ISP)	250
24.3.1 ISP 通讯协议	251
24.3.2 使用的资源	252
24.3.3 ISP 命令	252
24.3.4 ISP 返回代码	256
24.4 在应用编程 (IAP)	257
24.4.1 IAP 命令	257
24.4.2 IAP 使用	258
24.5 量产平台	258
25 封装尺寸及焊接条件	259

25.1 封装尺寸	259
25.1.1 LQFP128L	259
25.1.2 LQFP100L	260
25.1.3 LQFP64L	261
25.2 回流焊炉的温度设定条件	264
26 包装信息	265
26.1 托盘规格	265

1 概述

1.1 简介

MCU 芯片 RN8318 V2 版、RN8613 V2 版、RN8615 V2 版、RN8611 V2 版是锐能微第二代 MCU 芯片，在第一代产品的基础上提高性能，拓展功能，并向下兼容第一代产品。其中型号为 RN8318、RN8613、RN8615 的芯片，分别还有一个合封了一颗 EEPROM 的型号，对应产品型号分别为 RN8318 V2EE512K、RN8613 V2EE512K、RN8615 V2EE512K。

RN8318 V2 版: 封装为 LQFP128L，和 V1 pin to pin 兼容，寄存器兼容，软件库函数需升级到 V2 版本。SRAM 扩容至 96KB+4KB，优化功耗、RTC 性能，增加 ECC/AES/HASH/TRNG 硬件加速单元，增加 3 路 SPI，增加 SPI/UART DMA 通道。典型应用领域：国网 21 版三相表、海外三相表。

RN8318 V2EE512K 版: 在 RN8318 V2 版基础上合封 1 颗 512K 的 EEPROM。

RN8615 V2 版: 封装为 LQFP100L，和 V1 pin to pin 兼容，寄存器兼容，软件库函数需升级到 V2 版本。SRAM 扩容至 96KB+4KB，优化功耗、RTC 性能，增加 ECC/AES/HASH/TRNG 硬件加速单元，增加 3 路 SPI，增加 SPI/UART DMA 通道。典型应用领域：海外单三相表。

RN8615 V2EE512K 版: 在 RN8615 V2 版基础上合封 1 颗 512K 的 EEPROM。

RN8613 V2 版: 封装为 LQFP100L，和 V1 pin to pin 兼容，寄存器兼容，软件库函数需升级到 V2 版本。优化功耗、RTC 性能，增加 3 路 SPI，增加 SPI/UART DMA 通道。典型应用领域：国网 21 版单相表。

RN8613 V2EE512K 版: 在 RN8613 V2 版基础上合封 1 颗 512K 的 EEPROM。

RN8611 V2 版: 封装为 LQFP64L，新增型号。FLASH 容量 512KB，SRAM 容量 96KB。典型应用领域：单三相物联表分离方案、终端交采板。

1.2 产品特点

基本特点：

- 高集成：32bit ARM M0 CPU + 最大 512KB Flash/100KB SRAM+独立供电硬件温补 RTC+ LCD 控制器；
- 宽电压：
 - ✓ 典型条件下 2.3V~5.5V 电压范围保证 CPU 正常工作；
 - ✓ 典型条件下 2.5V~5.5V 电压范围保证 RTC 温补正常；
 - ✓ 高温 85°C 下 2.0V~5.5V 电压范围保证 RTC 温补正常；
 - ✓ 典型条件下 1.8V~5.5V 电压范围万年历计时正常；
- 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：
 - ✓ CPU 子系统工作在 32.768KHz 下功耗优于 19μA；
 - ✓ 休眠模式下芯片整体功耗约为 8.5μA (RTC 自动温补；ram 保持；CPU 及数字外设不掉电；中断唤醒)；
 - ✓ VBAT 域功耗典型值约为 1.8μA；
 - ✓ LCD 功耗约为 5μA，常显芯片整体功耗 14uA (电阻串分压模式，含 6COM 显示屏功耗)；
 - ✓ LCD 功耗约为 10μA，常显芯片整体功耗 19uA (电荷泵模式，含 8COM 显示屏功耗)；
- 高精度：
 - ✓ RTC 在全温度范围 (-40 度~85 度) 内秒脉冲误差 <±5ppm，最小校正刻度为 0.0339ppm；
- 处理器相关：
 - ✓ ARM Cortex-M0 内核；
 - ✓ RN8318/RN8615/RN8611：512KB FLASH 存储器；

- ✓ RN8613: 256KB FLASH 存储器;
- ✓ RN8318/RN8615/RN8611: 96KB+4KB SRAM;
- ✓ RN8613: 32KB SRAM;
- 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
- DSP:
 - ✓ 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
- RTC:
 - ✓ 硬件自动温补, 满足标准要求的精度和功耗, 温补不需要 CPU 参与。
 - ✓ 温度传感器: -25°C~70°C 范围内测温精度为±1°C
 - ✓ RTC 万年历及自动温补电路独立使用 VBAT 引脚供电
 - ✓ VBAT 域功耗典型值优于 1.8uA
 - ✓ 开启温补时间约为 2ms, 温度典型功耗 250μA, 如果 30 秒开启一次, 平均功耗为 0.016μA;
- LCD:
 - ✓ RN8318: 4*40、6*38、8*36;
 - ✓ RN8615/RN8613: 4*34、6*32、8*30;
 - ✓ 支持 ChargePUMP 和内置电阻列分压模式可选, 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。
 - ✓ LCD 模块功耗优于 5μA (电阻列分压方案)。
- 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。
- 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;
 - ✓ RN8613 不支持加密功能;
 - ✓ 加密说明文档见锐能微应用笔记;

其他外设:

- 高速 GPIO, 支持与不同电压外设器件的接口:
 - ✓ RN8318: 108 个 IO
 - ✓ RN8615/RN8613: 83 个 IO
 - ✓ RN8611: 50 个 IO
- 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
- 两个低功耗比较器 CMP: 检测外部电压
- 扩展定时器: 2 个 32bit 定时器, 4 个 16bit 简单定时器
- UART: 6 个, 支持自动波特率, 支持红外调制 (32.768KHz 主频下也可支持红外调制), 支持 UART 唤醒, 支持电平反转, 支持 DMA
- 7816 口: 2 个
- I2C: 1 个
- SPI: 4 个, 支持 DMA
- 硬件看门狗
- 按键中断: 8 个, 管脚复用
- 外部中断口: 8 个, 管脚复用
- EEPROM: 512K

1.3 MCU 产品型号资源对照表

型号	RN8318 V2 版/V2EE512K 版	RN8615 V2 版 /V2EE512K 版	RN8613 V2 版/V2EE512K 版	RN8611 V2 版
封装	LQFP128L	LQFP100L	LQFP100L	LQFP64L
CPU	Cortex-M0			
Typical Freq.	14.7456M			
Max Freq.	29.4912M			
FLASH	512KB		256KB	512KB
RAM	96KB+4KB		32KB	96KB+4KB B
EEPROM	✖/512KB (V2 版无 EEPROM, V2EE512K 版内含 512K 的 EEPROM)			
加密 SEA	√	√	-	√
TRNG	√	√	-	√
Timers	32bit Timer	2	2	2
	16bit Timer	4	4	4
	systick	1	1	1
RTC	1	1	1	1
WDT	1	1	1	1
KBI	8	4	4	3
INTC	8	8	8	8
SPI	4	4	4	4
UART	6	6	6	6
I2C	1	1	1	1
7816	2	2	2	2

GPIO	108	83	83	50
DMA		√		
D2F		√		
M2M		√		
DSP	√	√	-	√
LCD	4*40/6*38/8*36	4*34/6*32/8*30		-
LCD booster	√	√	√	-
CMP	2	2	2	1
LVD	1	1	1	-
SAR-ADC	7	7	7	2
TempSensor		√		

注: √表示具有对应模块, ×表示没有对应模块。

表 1-2 MCU 产品资源对照表

1.4 实现架构

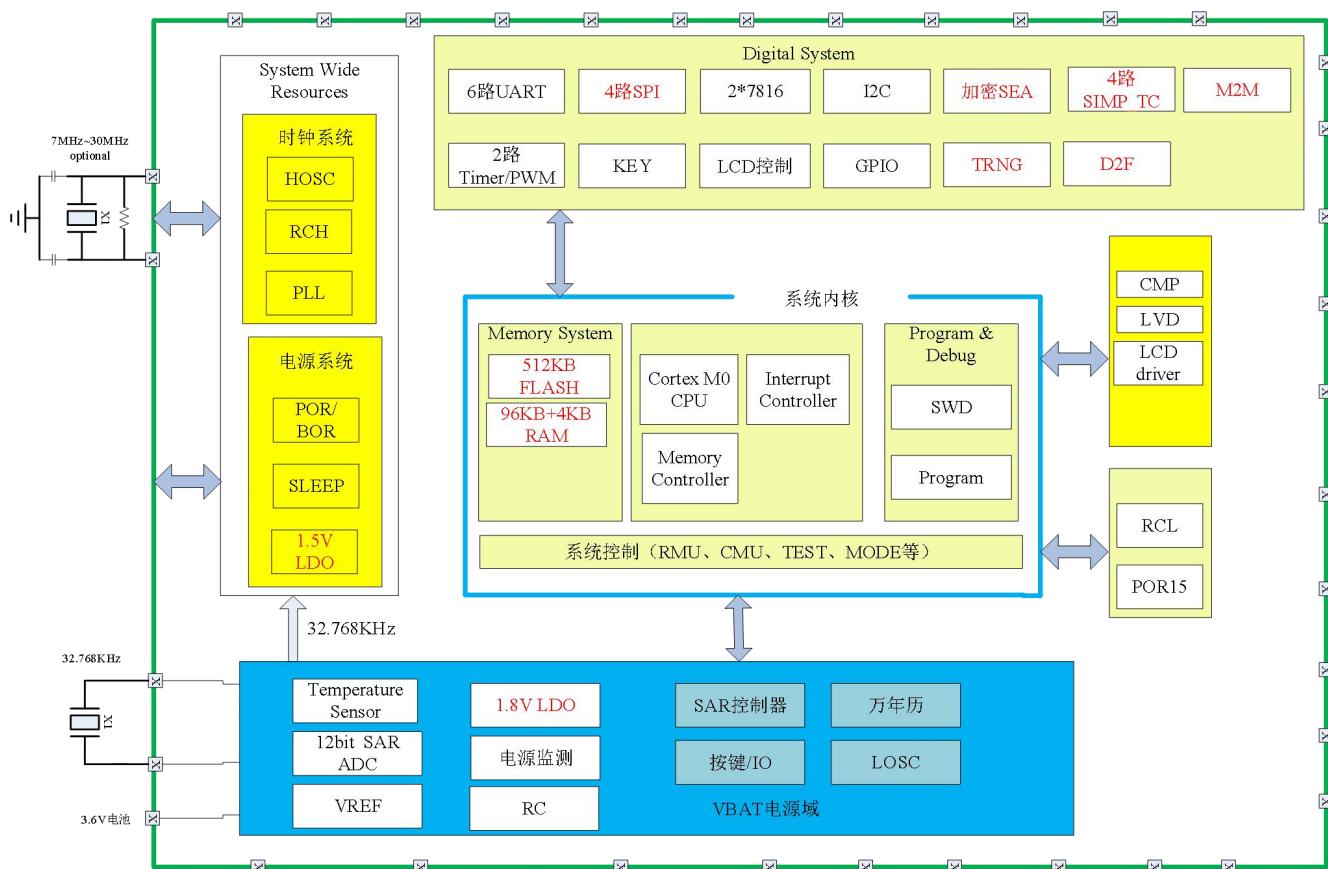


图 1-1 RN 系列 MCU 结构框图

1.5 管脚排列

1.5.1 RN8318 LQFP128L 引脚图

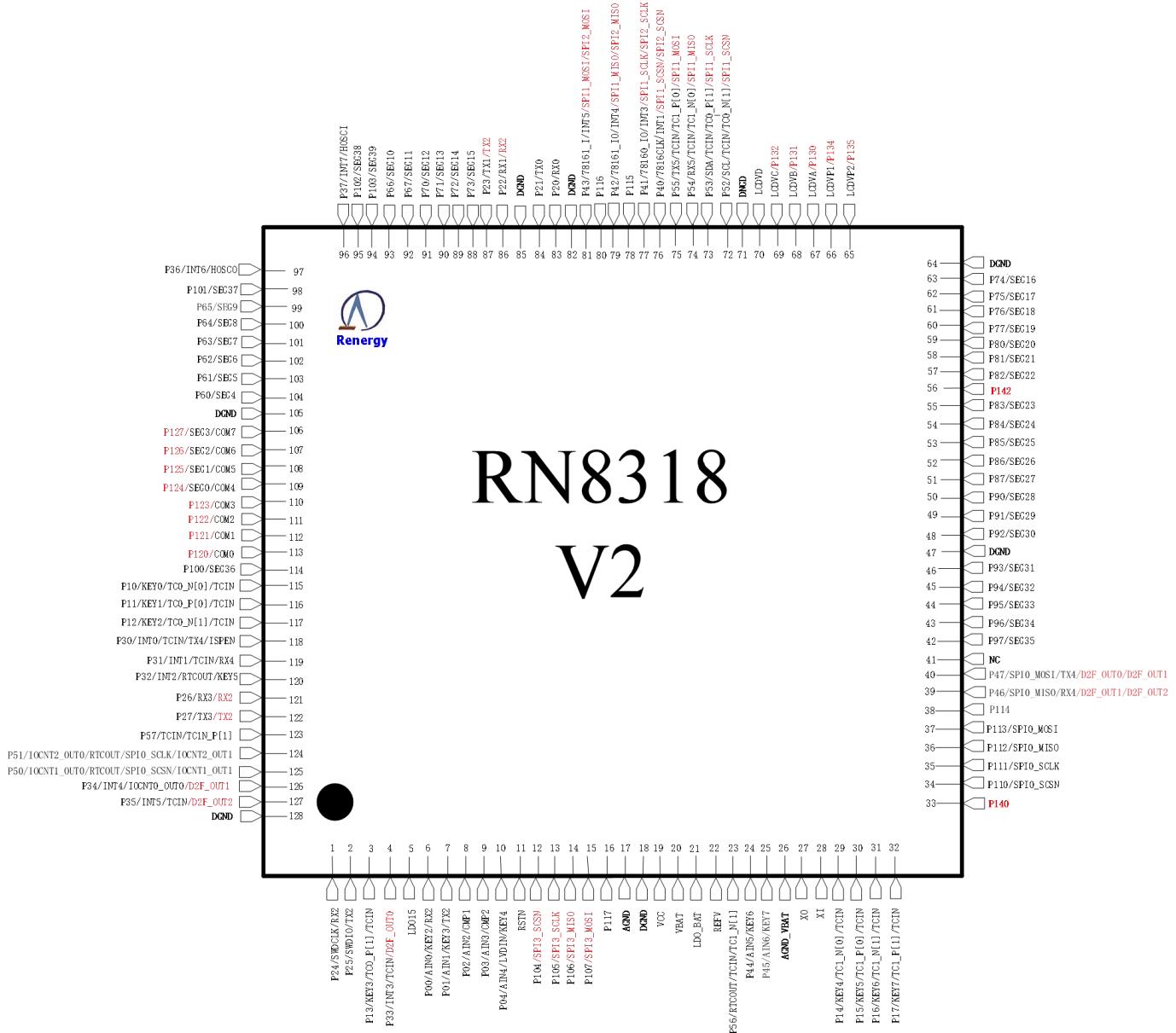


图 1-2 RN8318 管脚排列图

备注：

1. VBAT 域管脚有：VBAT、LDO_VBAT、REFV、P56、P44、P45、GND_VBAT、XO、XI。
2. 注意 P56/P44/P45 引脚输出高电平等于 VBAT 电压。
3. 注意 P56/P44/P45 引脚输入电压范围是：VBAT+0.3V~VSS-0.3V。
4. 红色字体的管脚是 RN8318 V1 与 RN8318 V2 有差异的管脚。
5. LCD 驱动功能管脚类型为 PBDG3，作为 GPIO 进行输出时，不能直接输出高电平，需加外部上拉。

1.5.2 RN8613/8615 LQFP100L 引脚图

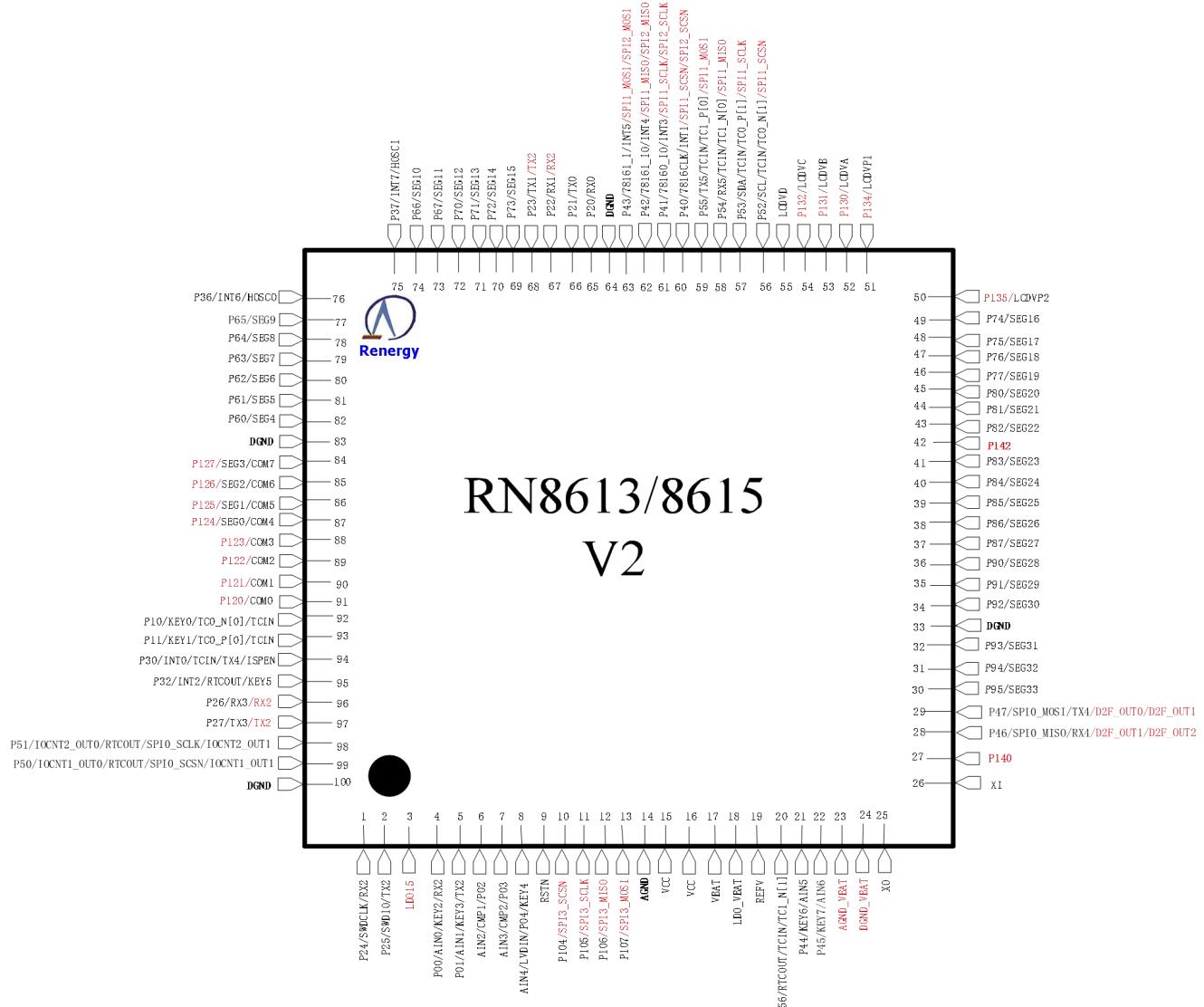


图 1-3 RN8613/RN8615 管脚排列图

备注：

1. VBAT 域管脚有：VBAT、LDO_VBAT、REFV、P56、P44、P45、AGND_VBAT、DGND_VBAT、XO、XI。
2. 注意 P56/P44/P45 引脚输出高电平等于 VBAT 电压。
3. 注意 P56/P44/P45 引脚输入电压范围是：VBAT+0.3V~VSS-0.3V。
4. 红色字体的管脚是 RN8613/8615 V1 与 RN8613/8615 V2 有差异的管脚。
5. LCD 驱动功能管脚类型为 PBDSG3，作为 GPIO 进行输出时，不能直接输出高电平，需加外部上拉。
- 6、注意 P140 管脚在未来 V3 版 RN8613 时，默认配置下，上拉电阻开启，阻值约为 5K。若该管脚硬件接地的情况下，软件需将该管脚的上拉关闭，否则会产生漏电。

1.5.3 RN8611 LQFP64L 引脚图

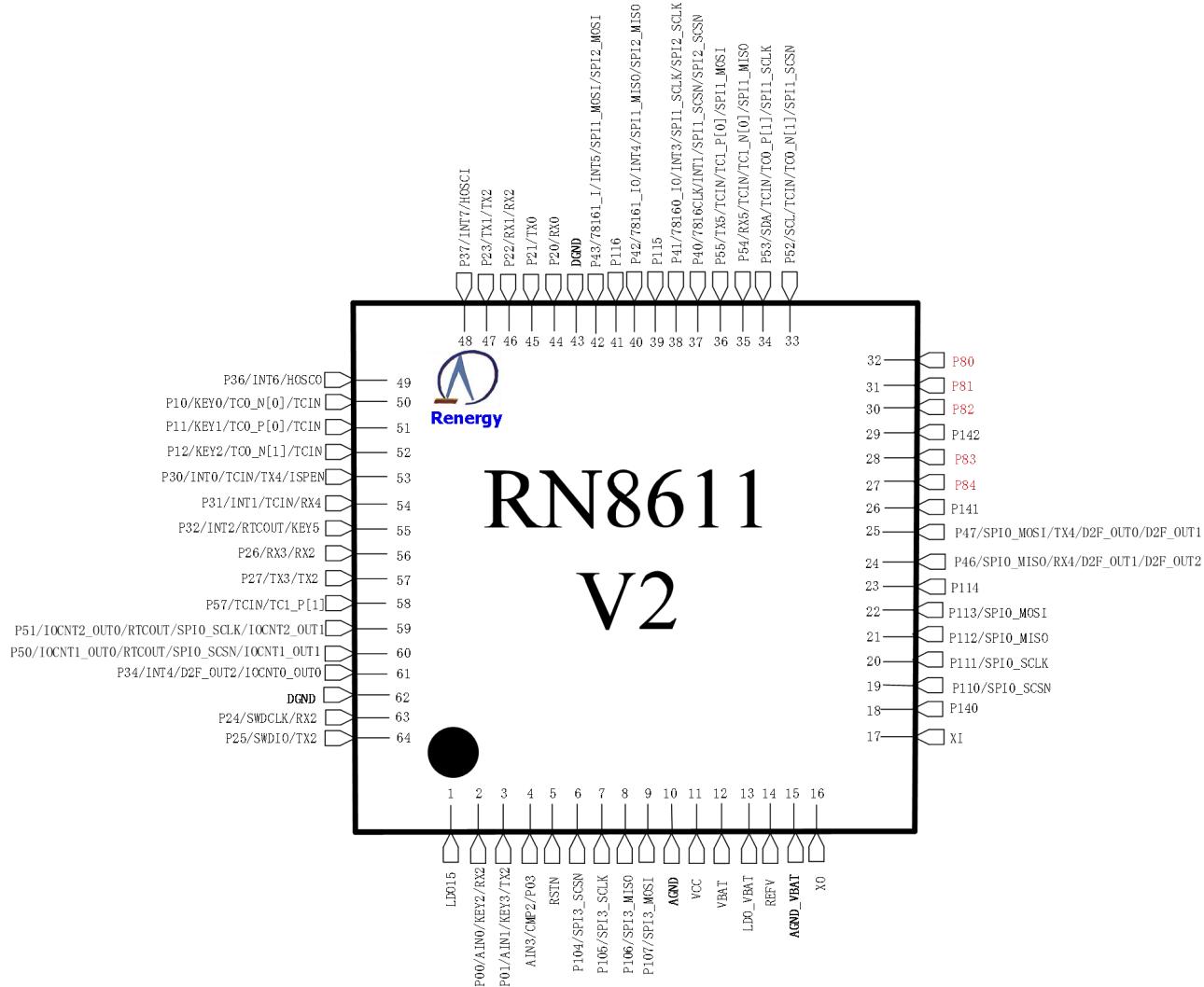


图 1-4 RN8611 管脚排列图

备注：

- 1.VBAT 域管脚有：VBAT、LDO_VBAT、REFV、GND_VBAT、XO、XI。
2. 红色引脚 P80、P81、P82、P83、P84 为开漏结构，输出时必需外接上拉电阻。

1.5.4 引脚类型说明

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

注：PBDS3 和 PBDSG3 类型管脚作为 GPIO 进行输出时，为开漏输出，必须外接上拉，才可输出高电平。

1.5.5 引脚定义说明

Pin Number			Pin Function	Type	Descriptions
RN8318	RN8615 RN8613	RN8611			
LQFP128L	LQFP100L	LQFP64L			
1	1	63	P24	PBULD3	GPIO
			SWDCLK		SWD 时钟口
			RX2		UART2 接收
2	2	64	P25	PBULD3	GPIO
			SWDIO		SWD 数据口
			TX2		UART2 发送
3			P13	PBULD3	GPIO
			KEY3		按键 3 输入
			TC0_P[1]		定时器 0 通道 1 比较正向输出
			TCIN		定时器输入
4			P33	PBUS6	GPIO
			INT3		外部中断 3 输入
			TCIN		定时器输入
			D2F_OUT0		电能积分模块 D2F0 的输出
5	3	1	LDO15	电源	内置 LDO 的输出，外接 1uF 并联 0.1uF 滤波电容。V1 版输出 1.8V；V2 版输出 1.5V
6	4	2	P00	PABUS3	GPIO，当管脚复用为 GPIO 的输入或输出功能时，引脚电压范围是：VCC+0.3V~VSS-0.3V

			AIN0	PABUS 3	SAR-ADC 外部通道 0 输入, 此时引脚输入电压范围是: VBAT+0.3V~VSS-0.3V AIN0 输入阻抗约 3M 建议外接 0.1uF 对地电容
			KEY2		按键 2 输入
			RX2		UART2 接收
7	5	3	P01	PABUS 3	GPIO, 当管脚复用为 GPIO 的输入或输出功能时, 引脚电压范围是: VCC+0.3V~VSS-0.3V
			AIN1		SAR-ADC 外部通道 1 输入, 此时引脚输入电压范围是: VBAT+0.3V~VSS-0.3V AIN1 输入阻抗约 3M 建议外接 0.1uF 对地电容
			KEY3		按键 3 输入
			TX2		UART2 发送
8	6		P02	PABUS 3	GPIO, 当管脚复用为 GPIO 的输入或输出功能时, 引脚电压范围是: VCC+0.3V~VSS-0.3V
			AIN2		SAR-ADC 外部通道 2 输入, 此时引脚输入电压范围是: VBAT+0.3V~VSS-0.3V 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 建议外接 0.1uF 对地电容
			CMP1		比较器 1 输入
9	7	4	P03	PABUS 3	GPIO, 当管脚复用为 GPIO 的输入或输出功能时, 引脚电压范围是: VCC+0.3V~VSS-0.3V
			AIN3		SAR-ADC 外部通道 3 输入, 此时引脚输入电压范围是: VBAT+0.3V~VSS-0.3V 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
10	8		P04	PABUS 3	GPIO, 当管脚复用为 GPIO 的输入或输出功能时, 引脚电压范围是: VCC+0.3V~VSS-0.3V
			AIN4		SAR-ADC 外部通道 4 输入, 此时引脚输入电压范围是: VBAT+0.3V~VSS-0.3V; 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 4 输入
11	9	5	RSTN	PIU	PIN 复位输入
12	10	6	P104	PBUS6	GPIO
			SPI3_SCSN		SPI3 片选
13	11	7	P105	PBUS6	GPIO
			SPI3_SCLK		SPI3 时钟
14	12	8	P106	PBUS6	GPIO
			SPI3_MISO		SPI3 主机输入从机输出数据口
15	13	9	P107	PBUS6	GPIO
			SPI3_MOSI		SPI3 主机输出从机输入数据口
16			P117	PBUILD 3	GPIO
17	14	10	AGND	-	模拟地
18			DGND	地	地
	15		VCC	电源	主电电源输入
19	16	11	VCC	电源	主电电源输入，应外接 4.7uF 电容并联 0.1uF 电容去耦。
20	17	12	VBAT	电源	3.6V 电池或超级电容输入引脚，建议外接 RC 滤波，R/C: 10Ω/1uF
21	18	13	LDO_VBAT	电源	内置 VBAT 域 LDO 输出，外部需接 0.22uF 电容。V1 版输出电压 2.0V；V2 版输出电压 1.8V
22	19	14	REFV	电源	SARADC 的参考输出，外部应接 0.22μF 电容
23	20		P56	PBUILD 3	VBAT 域 GPIO，当管脚复用为 GPIO 的输入或输出功能时，引脚电压范围是：VBAT+0.3V~VSS-0.3V
			RTCOUT		RTCOUT 输出，由寄存器 RTC.VBAT_IOEN[7:4]进行配置，RTCOUT 优先级高于其他复用功能。
			TC1_N[1]		定时器 1 通道 1 比较反向输出
			TCIN		定时器输入
24	21		P44	PABUL D3	VBAT 域 GPIO，当管脚复用为 GPIO 的输入或输出功能时，引脚电压范围是：VBAT+0.3V~VSS-0.3V。
			AIN5		SAR-ADC 外部通道 5 输入，由寄存器 MADC.ANA_PAD[0]进行配置，AIN5 优先级高于其他复用功能。作为 AIN5 使用时，AIN5 输入电压范围是：VBAT+0.3V~VSS-0.3V AIN5 输入阻抗约 3M

					建议外接 0.1uF 对地电容
			KEY6		按键 6 输入
25	22		P45	PABUL D3	VBAT 域 GPIO, 当管脚复用为 GPIO 的输入或输出功能时, 引脚电压范围是: VBAT+0.3V~VSS-0.3V
			AIN6		SAR-ADC 外部通道 6 输入, 由寄存器 MADC.ANA_PAD[0]进行配置, AIN6 优先级高于其他复用功能。 作为 AIN6 使用时需注意: 1、建议管脚接 0.1uF 对地电容 2、AIN6 作为输入时, 输入电压范围是: VBAT+0.3V~VSS-0.3V 3、AIN6 在被选通进行 SAR 测量时, 内部有两个 300K 电阻做分压; 但因 SAR 测量时间短暂, 在外部管脚有 0.1uF 对地电容情况下, 单次 SAR 测量结果不受分压电阻影响
			KEY7		按键 7 输入
26	23	15	AGND_VBAT	地	VBAT 域模拟地
	24		DGND_VBAT	-	地
27	25	16	XO	时钟	32.768KHz 无源晶振输出。
28	26	17	XI	时钟	32.768KHz 无源晶振输入。 XO/XI 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; XO 和 XI 引脚附近 IO 不使用高速翻转的功能, 如作为通讯口
29			P14	PBULD 3	GPIO, 建议不要进行高速翻转, 如作为通讯口, 会影响 LOSC 晶体性能
			KEY4		按键 4 输入
			TC1_N[0]		定时器 1 通道 0 比较反向输出
			TCIN		定时器输入
30			P15	PBULD 3	GPIO
			KEY5		按键 5 输入
			TC1_P[0]		定时器 1 通道 0 比较正向输出
			TCIN		定时器输入
31			P16	PBULD 3	GPIO
			KEY6		按键 6 输入
			TC1_N[1]		定时器 1 通道 1 比较反向输出
			TCIN		定时器输入
32			P17	PBULD 3	GPIO
			KEY7		按键 7 输入
			TC1_P[1]		定时器 1 通道 1 比较正向输出
			TCIN		定时器输入

					V1 版为数字地； V2 版为 GPIO，应用上接地不影响芯片功能。 V3 版 RN8613，为 GPIO，默认配置下，上拉电阻开启，阻值约为 5K。 V3EE512K 版 RN8613，该管脚为 NC，应用上接地不影响芯片功能。 V2/V3 版应用注意： 1、RN8613/RN8611 芯片，该管脚靠近 LOSC 引脚，设计时需要注意，否则会影响 LOSC 的性能：a) 建议不要进行连续的电平翻转，例如作为 UART，模拟 I2C 等接口。b) 建议晶体走线需要包地，管脚尽量不输出高。 2、RN8613 V3 版本，此管脚默认上拉开启，上拉电阻 5KΩ，当管脚做为输出时，如果电路是高电平驱动，默认的上拉电阻可能会引起误动作，不建议设计输出高驱动，对低电平驱动，默认上拉电阻可在程序启动后关闭，无影响。当此口做为输入时，程序启动后可关闭上拉配置，注意低功耗情况下，客户需要判断是否关闭默认上拉电阻。 3、RN8613 V3 版本，此口的上拉关闭与其他位是反逻辑，关闭上拉电阻：PUD 寄存器 bit16 配置为 1，打开上拉电阻：PUD 寄存器 bit16 配置为 0，如程序中对寄存器值有判断，需注意此位的配置
34		19	P110 SPI0_SCSN	PBUILD 3	GPIO SPI0 片选
35		20	P111 SPI0_SCLK	PBUILD 3	GPIO SPI0 时钟
36		21	P112 SPI0_MISO	PBUILD 3	GPIO SPI0 主机输入从机输出数据口
37		22	P113 SPI0_MOSI	PBUILD 3	GPIO SPI0 主机输出从机输入数据口
38		23	P114	PBUILD 3	GPIO
39	28	24	P46 SPI0_MISO RX4 D2F_OUT1	PBUILD 6	GPIO SPI0 主机输入从机输出数据口 UART4 接收 电能积分模块 D2F1 的输出，source 驱动能力：V1 版 3mA，V2 版 6mA

			D2F_OUT2		电能积分模块 D2F2 的输出, source 驱动能力: V1 版 3mA, V2 版 6mA
40	29	25	P47	PBUDL 6	GPIO
			SPI0_MOSI		SPI0 主机输出从机输入数据口
			TX4		UART4 发送
			D2F_OUT0		电能积分模块 D2F0 的输出, source 驱动能力: V1 版 3mA, V2 版 6mA
			D2F_OUT1		电能积分模块 D2F1 的输出, source 驱动能力: V1 版 3mA, V2 版 6mA
41			NC	-	悬空
42			P97	PBDSG 3	GPIO
			SEG35		LCD 驱动 SEG 端口
43			P96	PBDSG 3	GPIO
			SEG34		LCD 驱动 SEG 端口
44	30		P95	PBDSG 3	GPIO
			SEG33		LCD 驱动 SEG 端口
45	31		P94	PBDSG 3	GPIO
			SEG32		LCD 驱动 SEG 端口
46	32		P93	PBDSG 3	GPIO
			SEG31		LCD 驱动 SEG 端口
		26	P141	PBUDL 3	<p>V2 版为 GPIO, 默认配置下, 上拉 5K 电阻开启, 不能直接接地</p> <p>V2EE512K 版, 该管脚为 NC, 应用上接地不影响芯片功能。</p> <p>V3EE512K 版, 该管脚为 NC, 应用上接地不影响芯片功能。</p> <p>注: 该引脚与其他引脚不同, 输入/输出模式上拉配置均有效。其他引脚上拉配置, 只在输入模式有效</p>
47	33		DGND	地	地
48	34		P92	PBDSG 3	GPIO
			SEG30		LCD 驱动 SEG 端口
49	35		P91	PBDSG 3	GPIO
			SEG29		LCD 驱动 SEG 端口
50	36		P90	PBDSG 3	GPIO
			SEG28		LCD 驱动 SEG 端口
51	37		P87	PBDSG 3	GPIO
			SEG27		LCD 驱动 SEG 端口
52	38		P86	PBDSG 3	GPIO
			SEG26		LCD 驱动 SEG 端口
53	39		P85	PBDSG 3	GPIO
			SEG25		LCD 驱动 SEG 端口
54	40	27	P84	PBDSG	GPIO

			SEG24	3	LCD 驱动 SEG 端口
55	41	28	P83	PBDSG 3	GPIO
			SEG23		LCD 驱动 SEG 端口
56	42	29	P142	PBUILD 3	<p>V1 版为数字地； V2 版为 GPIO，应用上接地不影响芯片功能。 V2EE512K 版，该管脚为 NC，应用上接地不影响芯片功能。。 V3 版为 GPIO，应用上接地不影响芯片功能。</p>
57	43	30	P82	PBDSG 3	GPIO
			SEG22		LCD 驱动 SEG 端口
58	44	31	P81	PBDSG 3	GPIO
			SEG21		LCD 驱动 SEG 端口
59	45	32	P80	PBDSG 3	GPIO
			SEG20		LCD 驱动 SEG 端口
60	46		P77	PBDSG 3	GPIO
			SEG19		LCD 驱动 SEG 端口
61	47		P76	PBDSG 3	GPIO
			SEG18		LCD 驱动 SEG 端口
62	48		P75	PBDSG 3	GPIO
			SEG17		LCD 驱动 SEG 端口
63	49		P74	PBDSG 3	GPIO
			SEG16		LCD 驱动 SEG 端口
64			DGND	-	地
65	50		P135	PBDS3	<p>GPIO V2 版引脚类型修改，可做 GPIO，V1 版不支持 GPIO 功能。 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，如果要用作其他功能，需通过软件配置。 LCD 选择为 PUMP 方式时，要求配置为该模式，且 LCDVP2 和 LCDVP1 之间应该连接一个 100nF 的电容。 LCD 如果使用电阻分压方式，管脚可悬空或接地。</p>
66	51		P134	PBDS3	<p>GPIO 应用说明同 P135 描述</p>

			LCDVP1		模拟输出 应用方法见 LCDVP2 说明
67	52		P130	PBDS3	GPIO 应用说明同 P135 描述
			LCDVA		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。 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
68	53		P131	PBDS3	GPIO 应用说明同 P135 描述
			LCDVB		LCD 电压输出 应用方法见 LCDVA 说明
69	54		P132	PBDS3	GPIO 应用说明同 P135 描述
			LCDVC		LCD 电压输出 应用方法见 LCDVA 说明
70	55		LCDVD	PBDS3	LCD 电压输出, 需要外接 470nF 电容。 V2 版引脚类型修改, 不支持 GPIO 功能
71			DGND	地	地
72	56	33	P52	PBULD 3	GPIO
			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
73	57	34	P53	PBULD 3	GPIO
			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
74	58	35	P54	PBULD 3	GPIO
			RX5		UART5 接收
			TC1_N[0]		定时器 0 通道 0 比较反向输出
			TCIN		定时器输入
			SPI1_MISO		SPI1 主机输入从机输出数据口
75	59	36	P55	PBULD 3	GPIO
			TX5		UART5 发送
			TC1_P[0]		定时器 1 通道 0 比较正向输出
			SPI1_MOSI		SPI1 主机输出从机输入数据口
			TCIN		定时器输入
76	60	37	P40	PBULD 3	GPIO
			7816CLK		7816 时钟输出
			INT1		外部中断 1 输入

			SPI1_SCSN		SPI1 片选
			SPI2_SCSN		SPI2 片选
77	61	38	P41	PBULD 3	GPIO
			78160_IO		7816 0 双向数据口
			INT3		外部中断 3 输入
			SPI1_SCLK		SPI1 时钟
			SPI2_SCLK		SPI2 时钟
78		39	P115	PBULD 3	GPIO
79	62	40	P42	PBULD 3	GPIO
			78161_IO		7816 1 双向数据口
			INT4		外部中断 4 输入
			SPI1_MISO		SPI1 主机输入从机输出数据口
			SPI2_MISO		SPI2 主机输入从机输出数据口
80		41	P116	PBULD 3	GPIO
81	63	42	P43	PBULD 3	GPIO
			78161_I		7816 1 数据输入
			INT5		外部中断 5 输入
			SPI1_MOSI		SPI1 主机输出从机输入数据口
			SPI2_MOSI		SPI2 主机输出从机输入数据口
82	64	43	DGND	地	地
83	65	44	P20	PBULD 3	GPIO
			RX0		UART0 接收
84	66	45	P21	PBULD 3	GPIO
			TX0		UART0 发送
85			DGND	-	地
86	67	46	P22	PBULD 3	GPIO
			RX1		UART1 接收
			RX2		UART2 接收
87	68	47	P23	PBULD 3	GPIO
			TX1		UART1 发送
			TX2		UART2 发送
88	69		P73	PBDSG 3	GPIO
			SEG15		LCD 驱动 SEG 端口
89	70		P72	PBDSG 3	GPIO
			SEG14		LCD 驱动 SEG 端口
90	71		P71	PBDSG 3	GPIO
			SEG13		LCD 驱动 SEG 端口
91	72		P70	PBDSG 3	GPIO
			SEG12		LCD 驱动 SEG 端口
92	73		P67	PBDSG 3	GPIO
			SEG11		LCD 驱动 SEG 端口

93	74		P66	PBDSG 3	GPIO
			SEG10		LCD 驱动 SEG 端口
94			P103	PBDSG 3	GPIO
			SEG39		LCD 驱动 SEG 端口
95			P102	PBDSG 3	GPIO
			SEG38		LCD 驱动 SEG 端口
96	75	48	P37	PIUX	GPIO, 只能作为输入引脚, 不能输出
			INT7		外部中断 7 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	P36	PIUX	GPIO, 只能作为输入引脚, 不能输出
			INT6		外部中断 6 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			P101	PBDSG 3	GPIO
			SEG37		LCD 驱动 SEG 端口
99	77		P65	PBDSG 3	GPIO
			SEG9		LCD 驱动 SEG 端口
100	78		P64	PBDSG 3	GPIO
			SEG8		LCD 驱动 SEG 端口
101	79		P63	PBDSG 3	GPIO
			SEG7		LCD 驱动 SEG 端口
102	80		P62	PBDSG 3	GPIO
			SEG6		LCD 驱动 SEG 端口
103	81		P61	PBDSG 3	GPIO
			SEG5		LCD 驱动 SEG 端口
104	82		P60	PBDSG 3	GPIO
			SEG4		LCD 驱动 SEG 端口
105	83		DGND	地	地
106	84		P127	PBDSG 3	V2 版为 GPIO, V1 版不支持 GPIO 功能
			SEG3		LCD 驱动 SEG 端口
			COM7		LCD 驱动 COM 端口
107	85		P126	PBDSG 3	V2 版为 GPIO, V1 版不支持 GPIO 功能
			SEG2		LCD 驱动 SEG 端口
			COM6		LCD 驱动 COM 端口
108	86		P125	PBDSG 3	V2 版为 GPIO, V1 版不支持 GPIO 功能
			SEG1		LCD 驱动 SEG 端口
			COM5		LCD 驱动 COM 端口
109	87		P124	PBDSG 3	V2 版为 GPIO, V1 版不支持 GPIO 功能
			SEG0		LCD 驱动 SEG 端口
			COM4		LCD 驱动 COM 端口

110	88		P123	PBDSG 3	V2 版为 GPIO, V1 版不支持 GPIO 功能	
			COM3		LCD 驱动 COM 端口	
111	89		P122	PBDSG 3	V2 版为 GPIO, V1 版不支持 GPIO 功能	
			COM2		LCD 驱动 COM 端口	
112	90		P121	PBDSG 3	V2 版为 GPIO, V1 版不支持 GPIO 功能	
			COM1		LCD 驱动 COM 端口	
113	91		P120	PBDSG 3	V2 版为 GPIO, V1 版不支持 GPIO 功能	
			COM0		LCD 驱动 COM 端口	
114			P100	PBDSG 3	GPIO	
			SEG36		LCD 驱动 SEG 端口	
115	92	50	P10	PBULD 3	GPIO	
			KEY0		按键 0 输入	
			TC0_N[0]		定时器 0 通道 0 比较反向输出	
			TCIN		定时器输入	
116	93	51	P11	PBULD 3	GPIO	
			KEY1		按键 1 输入	
			TC0_P[0]		定时器 0 通道 0 比较正向输出	
			TCIN		定时器输入	
117		52	P12	PBULD 3	GPIO	
			KEY2		按键 2 输入	
			TC0_N[1]		定时器 0 通道 1 比较反向输出	
			TCIN		定时器输入	
118	94	53	P30	PBUS6	GPIO	
			INT0		外部中断 0 输入	
			TX4		UART4 发送	
			TCIN		定时器输入	
			ISPEN		发生复位后, BOOTROM 会检测该端口的状态, 如果输入低电平, 系统会进入 ISP。实际应用时需要注意该问题。	
119		54	P31	PBUS6	GPIO	
			INT1		外部中断 1 输入	
			RX4		UART4 接收	
			TCIN		定时器输入	
120	95	55	P32	PBUS6	GPIO	
			INT2		外部中断 2 输入	
			RTCONUT		RTCONUT 输出 该管脚位于 VCC 域, 其 RTCONUT 输出频率的补偿值, 在温度补偿时由 VBAT 域的 RTC 模块计算后更新到 VCC 域; 应用时注意 VCC 域掉电后重新上电, 需及时启动温度补偿, 以加载 RTCONUT 的补偿值, 防止 RTCONUT 输出频率出现偏差。	

			KEY5		按键 5 输入
121	96	56	P26	PBULD 3	GPIO
			RX3		UART3 接收
			RX2		UART2 接收
122	97	57	P27	PBULD 3	GPIO
			TX3		UART3 发送
			TX2		UART2 发送
123		58	P57	PBULD 3	GPIO
			TC1_P[1]		定时器 1 通道 1 比较正向输出
			TCIN		定时器输入
124	98	59	P51	PBUS6	GPIO
			IOCNT2_OUT0		IOCNT2_OUT0 是 INT5 输入后直接输出
			RTCOOUT		RTCOOUT 输出 该管脚位于 VCC 域，其 RTCOUT 输出频率的补偿值，在温度补偿时由 VBAT 域的 RTC 模块计算后更新到 VCC 域；应用时注意 VCC 域掉电后重新上电，需及时启动温度补偿，以加载 RTCOUT 的补偿值，防止 RTCOUT 输出频率出现偏差。
			SPI0_SCLK		SPI0 时钟
			IOCNT2_OUT1		IOCNT2_OUT1 是 INT7 输入后直接输出
			RTCOOUT		RTCOOUT 输出 该管脚位于 VCC 域，其 RTCOUT 输出频率的补偿值，在温度补偿时由 VBAT 域的 RTC 模块计算后更新到 VCC 域；应用时注意 VCC 域掉电后重新上电，需及时启动温度补偿，以加载 RTCOUT 的补偿值，防止 RTCOUT 输出频率出现偏差。
125	99	60	P50	PBUS6	GPIO
			IOCNT1_OUT0		IOCNT1_OUT0 是 INT2 输入后直接输出
			RTCOOUT		RTCOOUT 输出 该管脚位于 VCC 域，其 RTCOUT 输出频率的补偿值，在温度补偿时由 VBAT 域的 RTC 模块计算后更新到 VCC 域；应用时注意 VCC 域掉电后重新上电，需及时启动温度补偿，以加载 RTCOUT 的补偿值，防止 RTCOUT 输出频率出现偏差。
			SPI0_SCSN		SPI0 片选
			IOCNT1_OUT1		IOCNT1_OUT1 是 INT6 输入后直接输出
126		61	P34	PBUS6	GPIO
			INT4		外部中断 4 输入
			IOCNT0_OUT0		IOCNT0_OUT0 是 INT4 输入后直接输出
			D2F_OUT1		电能积分模块 D2F1 的输出

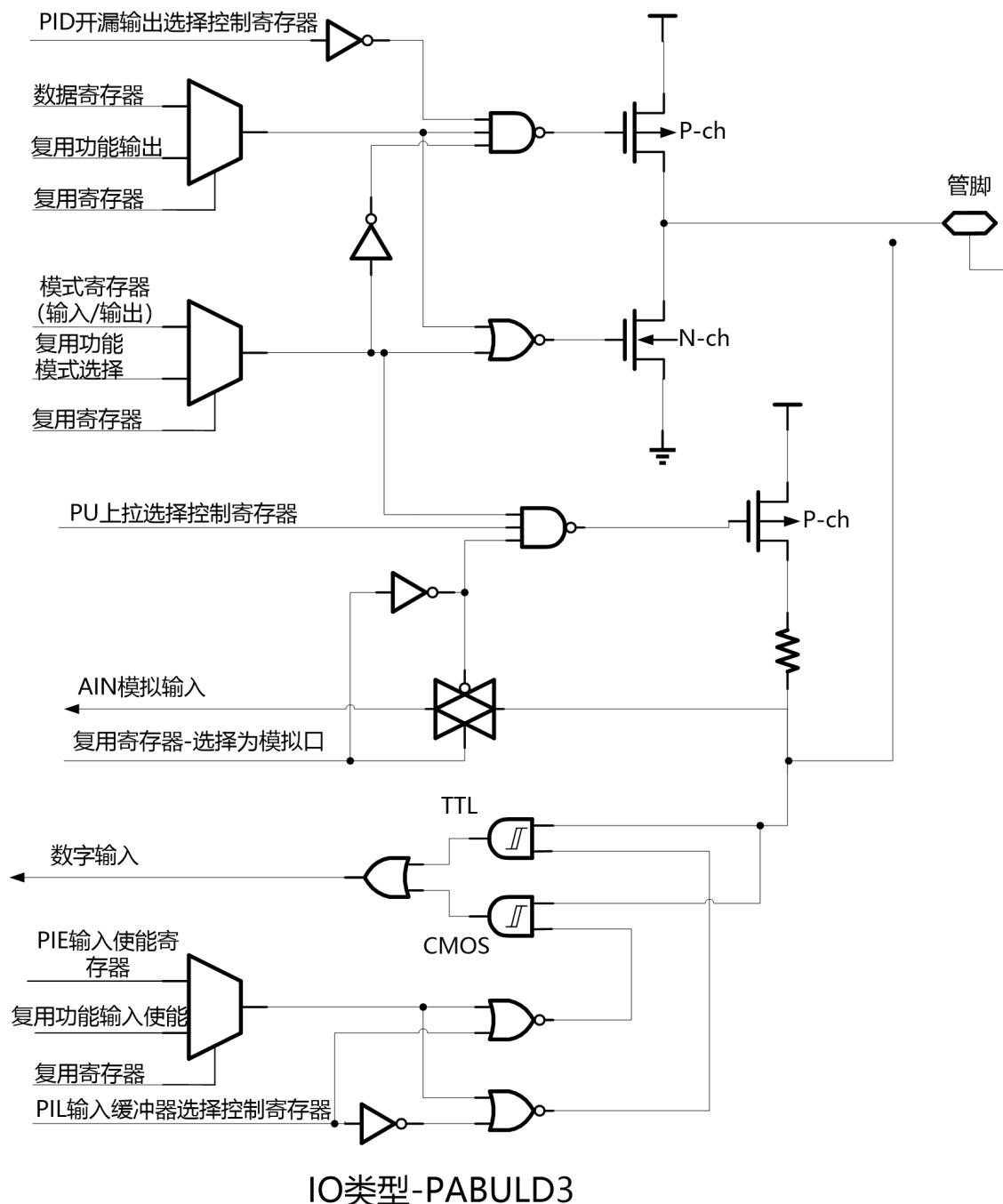
127			P35	PBUS6	GPIO
			INT5		外部中断 5 输入
			TCIN		定时器输入
			D2F_OUT2		电能积分模块 D2F2 的输出
128	100	62	DGND	地	地

1.5.6 硬件设计规则

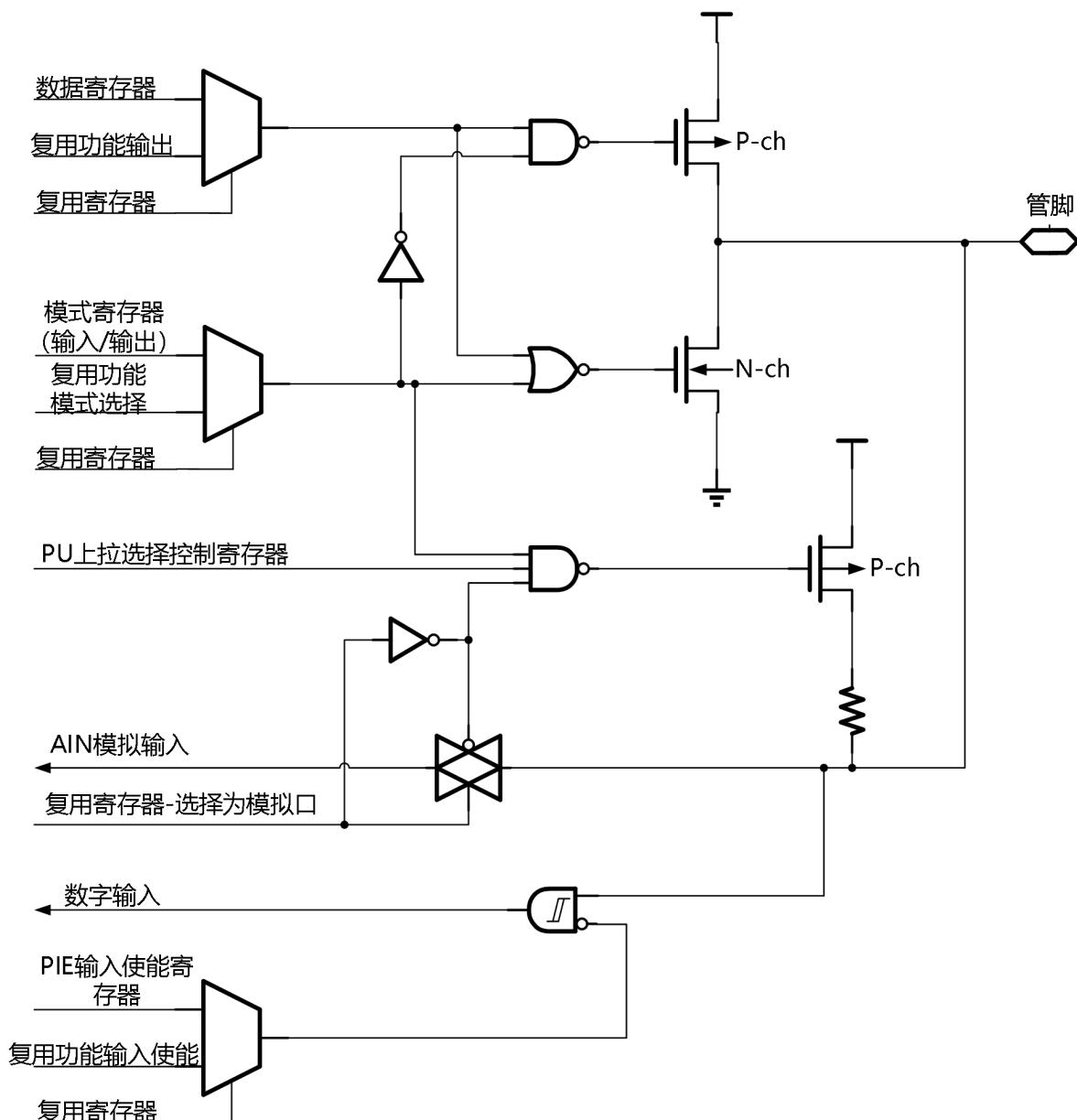
- 1、 AGND是模拟地， DGND是数字地；实际使用时可根据经验合理布局。
- 2、 LDO15：应外接1μf电容并联0.1μf电容去耦。
- 3、 REFV是SAR-ADC的参考输入，外部应接0.22μf电容。
- 4、 VBAT是3.6V电池或超级电容输入引脚：建议外接RC滤波， R:10Ω， C:1μF。
- 5、 VCC是主电输入，正常工作范围2.3V~5.5V，应外接4.7μf电容并联0.1μf电容去耦。
- 6、 LDO_VBAT: VBAT域LDO输出，外部需接0.22μf电容。
- 7、 XO和XI之间跨接32.768KHz晶体，最好用地线将之隔离，不需外接电阻和电容。**XO和XI引脚附近IO不使用高速翻转的功能，如作为通讯口。**
- 8、 LCDVD、LCDVC、LCDVB、LCDVA是LCD电压输出，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。**
- 9、 LCD CHARGE PUMP模式：LCDVP1和LCDVP2之间应连接一个100nf的电容；内部电阻列模式：LCDVP1、LCDVP2可悬空，或LCDVP1悬空、LCDVP2接地。**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。**
- 10、**SEG/IO复用引脚做为IO口使用时，是开漏结构，输出必须接外部上拉。**
- 11、对VBAT引脚和AIN6引脚输入的电压进行测量时，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。
- 12、P36、P37 做为 IO 口使用时，只能作为输入引脚，不能作为输出；
- 13、P56/P44/P45 引脚在 VBAT 电源域，输出高电平与 VBAT 相等。
- 14、定时器输入时可为 TC0 或者 TC1，具体是哪个定时器的输入，由定时器配置寄存器决定。
- 15、UART 设计说明，同一个 UART，不同的 IO 复用不能同时生效，只能有一组复用生效，需要规避。
- 16、SPI 设计说明，同一个 SPI，不同的 IO 复用不能同时生效，只能有一组复用生效，需要规避。

1.6 IO 口功能框图

1.6.1 PABULD3

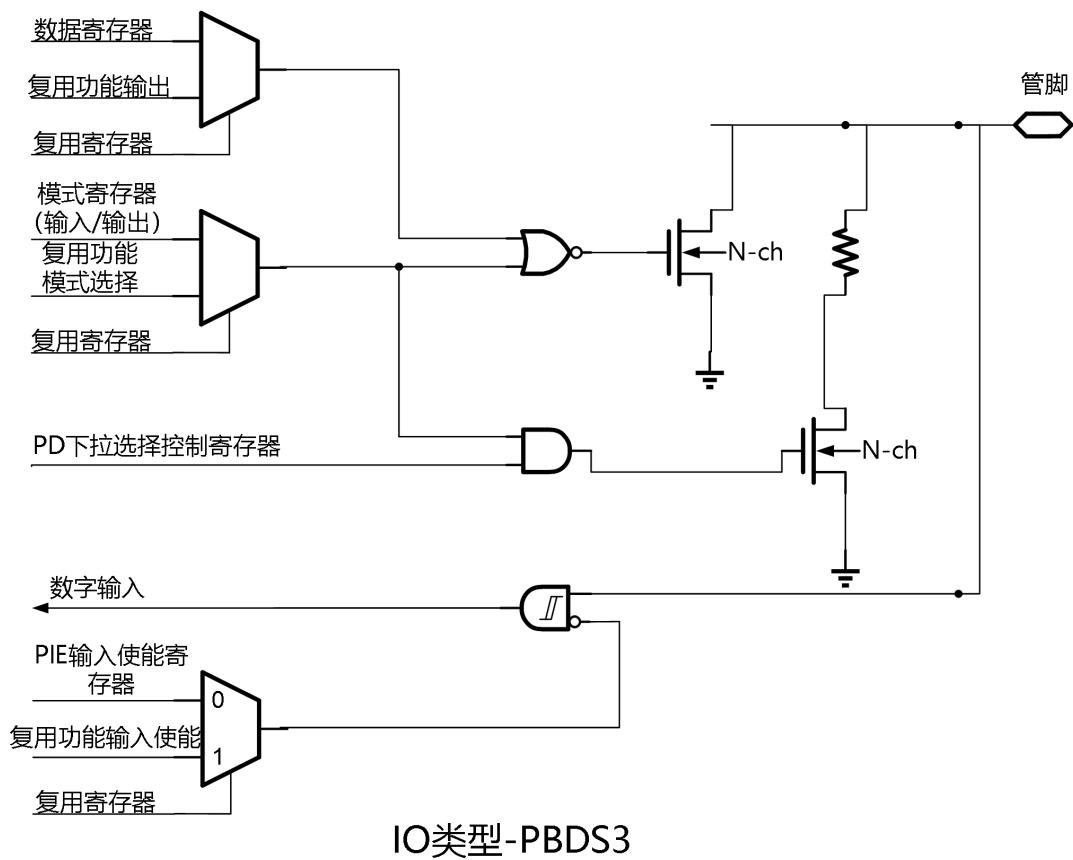


1.6.2 PABUS3

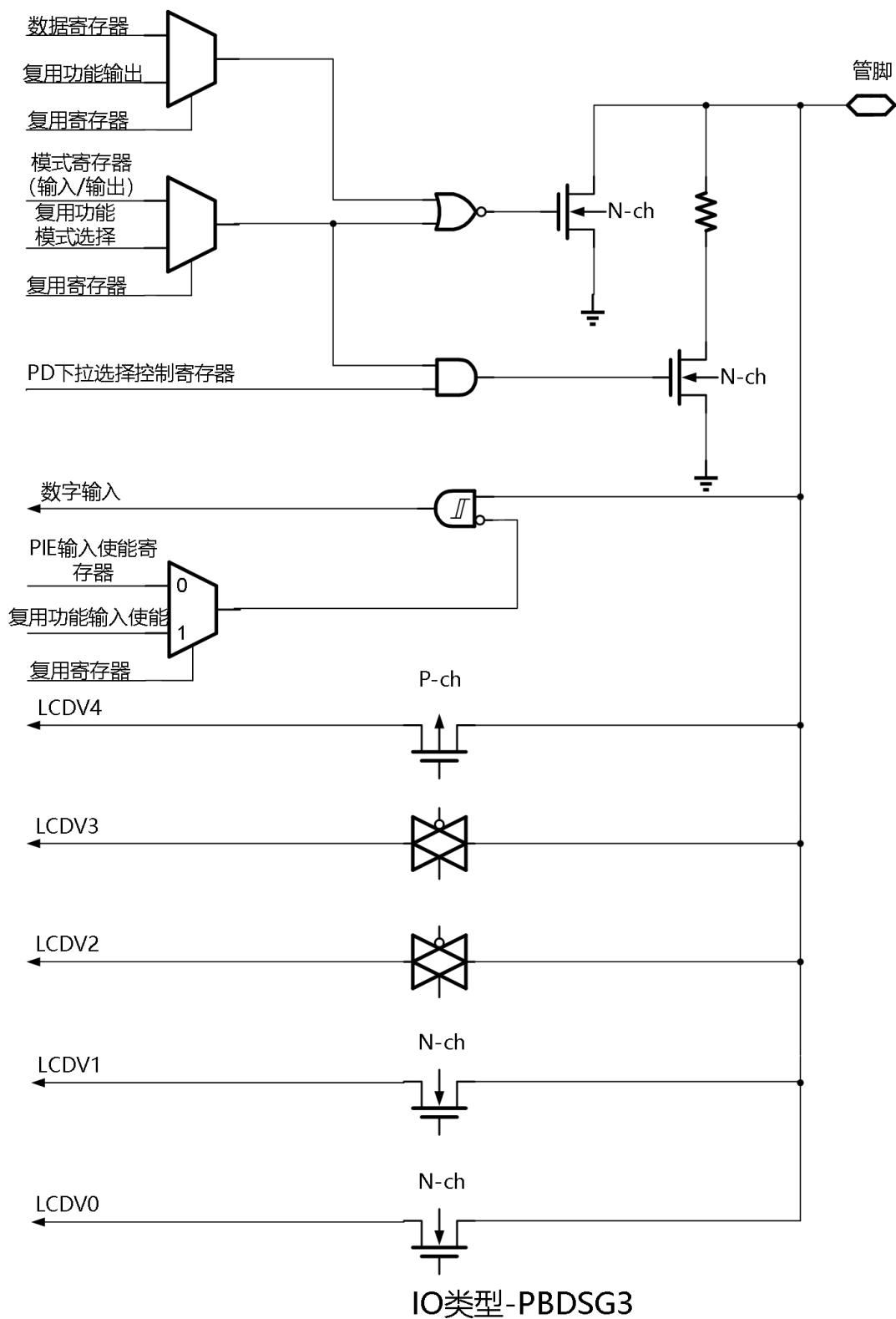


IO类型-PABUS3

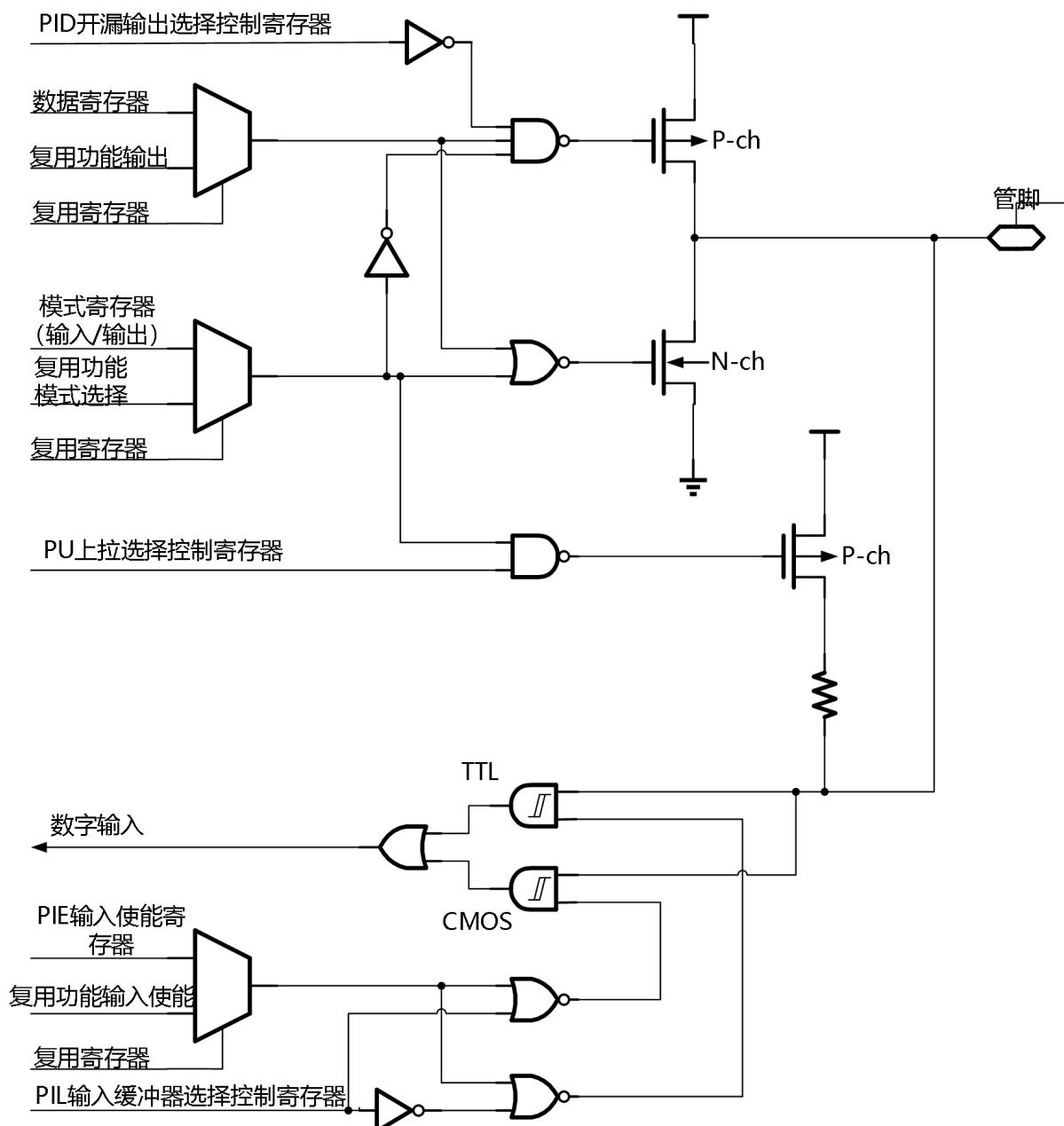
1.6.3 PBDS3



1.6.4 PBDG3

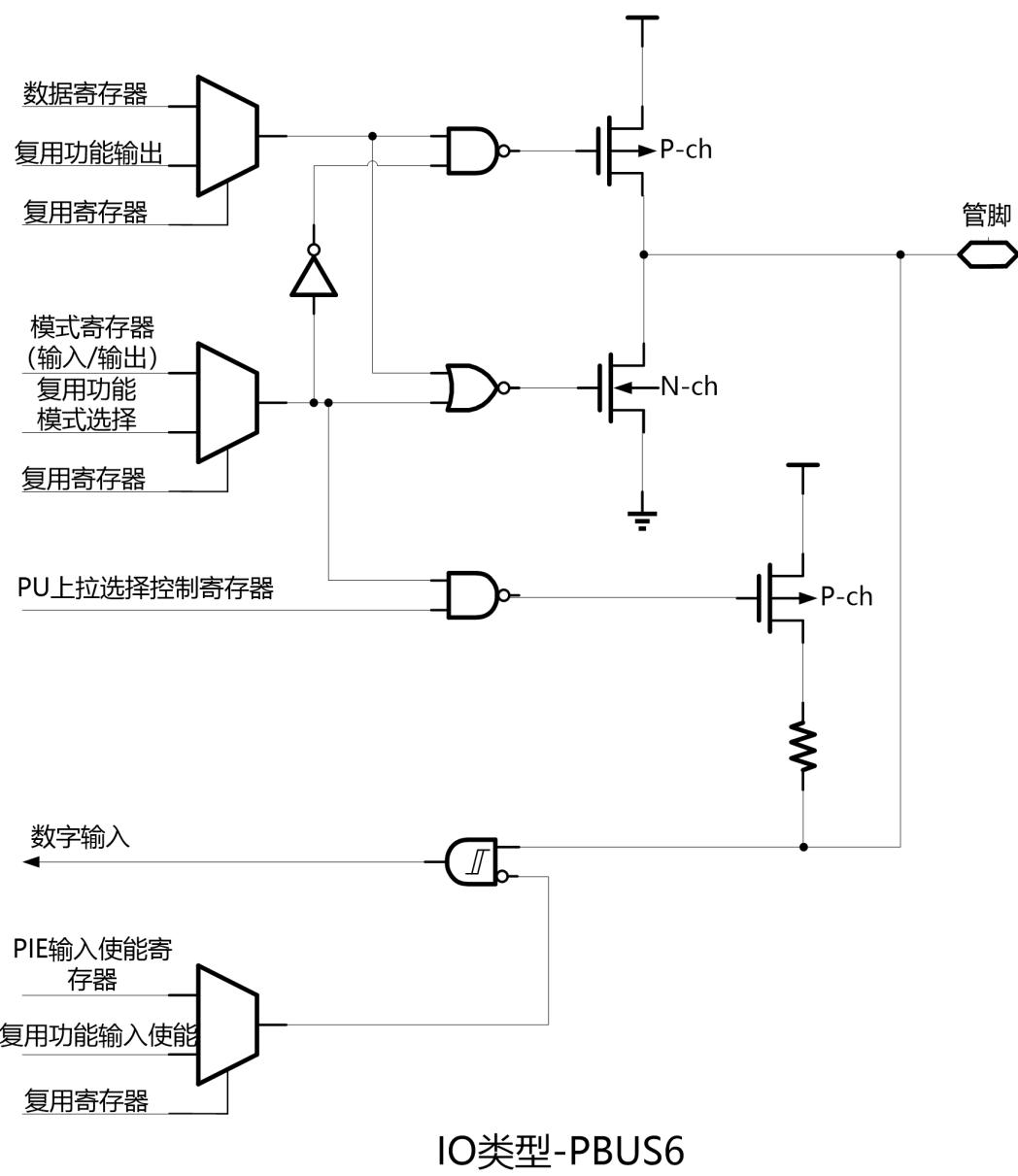


1.6.5 PBULD3/6

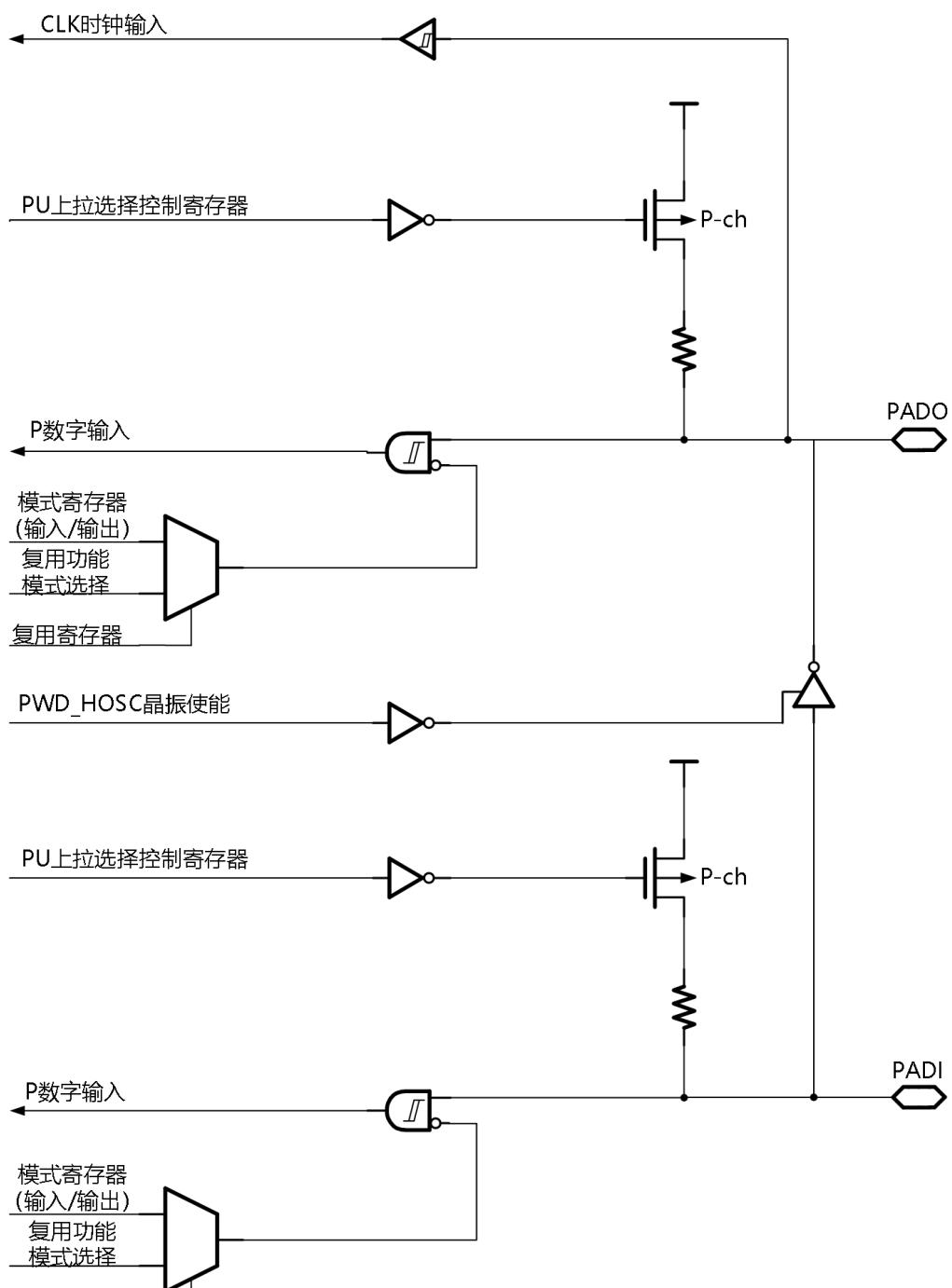


IO类型-PBULD3/PBULD6

1.6.6 PBUS6



1.6.7 PIUX



IO类型-PIUX

2 电气特性

2.1 性能参数

测试项目	符号	最小	典型	最大	单位	测试条件和注释
电源(温度范围: -40°C~+85°C)						
主电源	VCC	2.3	5/3.3	5.5	V	建议主电源选择为5V±5%或者3.3V±5%
CPU 最低工作电压	Vil		1.8		V	1.8V、2.2V 两档可调
VBAT 供电	VBAT1	2.5	3.6	5.5	V	RTC 温补电路正常工作, RTC 精度保证的电压工作范围
	VBAT2	1.8	3.6	5.5	V	RTC 万年历翻转正常, 温补电路不保证
数字电流	DIdd		1.5		mA	CPU运行在3.6864MHz (PLL)
数字电流	DIdd		2.5		mA	CPU运行在7.3728MHz (PLL)
数字电流	DIdd		4		mA	CPU运行在14.7456MHz (PLL)
VCC 休眠功耗	SIdd		6.7		µA	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 典型条件: Vcc=3.6V; Tc= 25 度
VBAT 供电功耗	SIdd		1.8		uA	典型条件: Vbat=3.6V; Tc=25 度; 极限条件: Vbat=3.6V; Tc=85 度;
瞬间温补功耗	TPSIdd		250		uA	典型条件: Vbat=3.6V; Tc=25度;
LDO15	V1P8	1.35	1.5	1.65	V	CPU内核电压, 全温度范围±10%
极限参数(温度范围: -40°C~+85°C)						
主电电压	Vvcc	-0.3	--	+7	V	
电池输入电压	Vvbat	-0.3	--	+7	V	
DV _{DD} to DGND		-0.3	--	+7	V	
DV _{DD} to AV _{DD}		-0.3		+0.3	V	
数字IO输出高电平	VOH		--	VCC+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	V _{IL}		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	--	A V _{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	
基准电压 (VCC=3V~5.5V, 温度范围: -40°C~+85°C)						
		min.	typ.	max.	unit	Notes
输出电压	V _{ref}	1.245	1.25	1.255	V	
温度系数	T _c		5	15	ppm/°C	
模拟外设(温度范围: -40°C~+85°C)						
低功耗比较器 CMP1/CMP2/LVDIN 阈值 ViL1	CMP	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; 输出高电平比较结果阈值比该阈值高 220mv。
低功耗比较器 CMP1/CMP2 阈值 ViL2	CMP	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; 输出高电平比较结果阈值比该阈值高 140mv。
SAR ADC 输入范围	SAR-IN	0		REFV	V	REFV 为内部基准电压, 典型值为 1.26V
LCD PUMP 输出电压	LCDVD	4.85	5	5.15	V	5V 电荷泵, 全温度范围测试
LCD 电阻串输出电压	LCDVD	3.135	3.3	3.465	V	3.3V 电阻串分压, 全温度范围
VBAT 测量	VBATD	0	3.6	5	V	SAR ADC 对 VBAT 引脚电压的测量范围;
每次温补时间	T _{tps}		2		ms	
时钟参数(温度范围: -40°C~+85°C)						
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	32.768	40	KHz	用于WDT时钟

2.2 可靠性参数

测量项目	符号	测试条件	值	单位
静电放电	ESD	人体模型（HBM），按照标准JEDEC EIA/JESD22-A114，在所有引脚上进行	4000	V
		机械模型（MM），按照标准JEDEC EIA/JESD22-A115C，在所有引脚上进行	200	V
		充电器件模型（CDM），按照标准JEDEC EIA/JESD22-C101F,在所有引脚上进行	500	V
闩锁试验	LatchUP	按照标准JEDEC STANDARD NO.78D NOVEMBER 2011,在所有引脚上进行	200	mA
湿度敏感性	MSL	按照标准IPC/JEDEC J-STD-020D.1评定	3级	/

3 系统控制

3.1 电源域划分

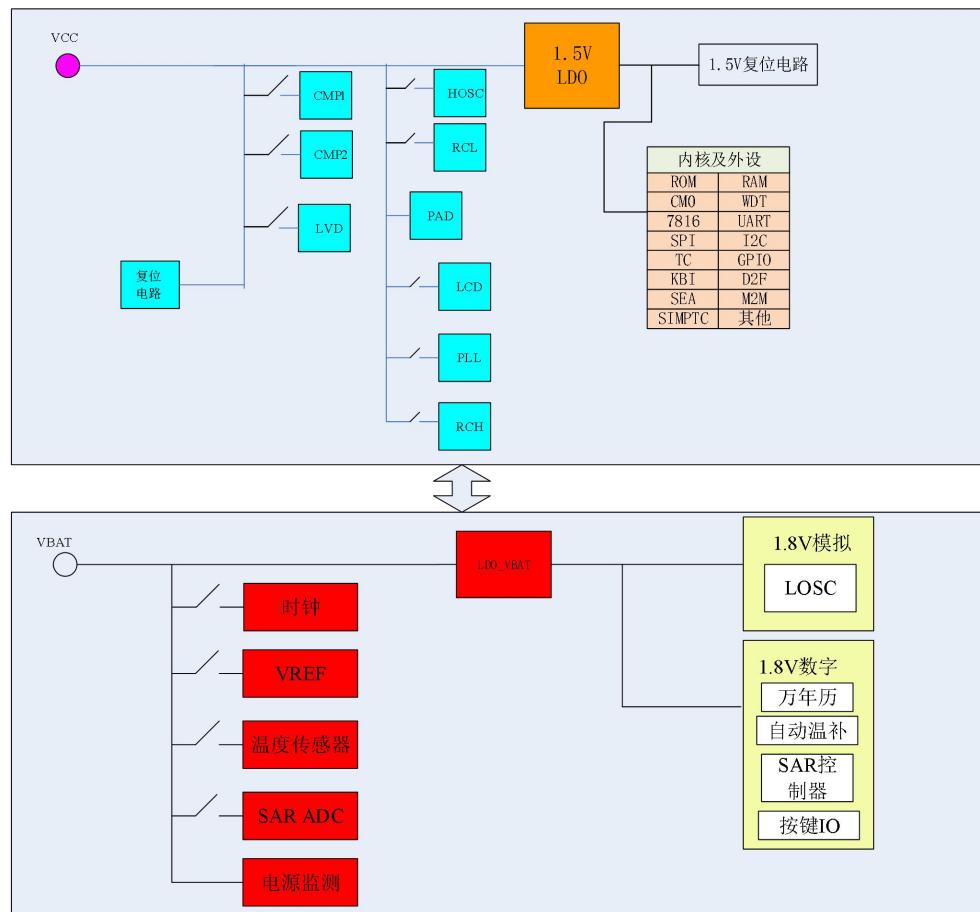


图 3-1 内部电源管理

VCC 与 VBAT 独立供电，RTC 相关电路（32.768KHz 晶振/万年历/自动温补电路等）、SAR-ADC 测量相关电路、P44/P45/P56 由 VBAT 供电，CPU 系统及其他外设由 VCC 供电。

实际应用中，可将 VCC 或 VBAT 连在一起供电，也可独立供电。

注意 P44、P45、P56 这三个 IO 口由 VBAT 供电，其输出高电平等于 VBAT，输入高电平不能高于 VBAT，并满足 V_{IH} 电平要求（ $0.7*V_{BAT}$ ）。如果只有 VBAT 上电，VCC 没有上电对 VBAT 域做初始化配置，P56 口默认输出 1Hz，一旦 VCC 完成上电对 VBAT 域做了初始化配置，P56 口恢复为默认高阻状态。

由于 RTC 模块部分配置寄存器（晶振参数设置等）需要由 CPU 在上电初始化时做配置（由 bootloader 调取 FLASH 选项字节中的参数，这些参数由客户在量产编程环节写入），因此需要保证配置参数能够正常写入。如果 VBAT 上电过慢，当 VCC 上电初始化对 VBAT 域做配置时，VBAT 尚未准备就绪，造成温补参数不能正常写入。建议 CPU 软件对 VBAT 域配置寄存器做有效管理，可调用锐能微库函数对相关寄存器做二次配置，以确保配置参数被正确写入。

3.2 时钟源

- 2 个外部时钟源：

LOSC：外部 32.768KHz 晶振，用于 RTC 时钟和低频工作下 CPU 时钟，永不关断。32.768KHz 晶振不需要外接电容和电阻，芯片已内置，建议选择负载电容为 12.5pF 的外部晶振。

HOSC：外部高频晶振，可支持外接 7.3728MHz、14.7456MHz、29.4912MHz 晶振。

- 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 1.8MHz; 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。

- 低频运行模式可以选择为 LOSC 或者 RCL

- LCD 时钟可选择为 LOSC 或者 RCL

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。

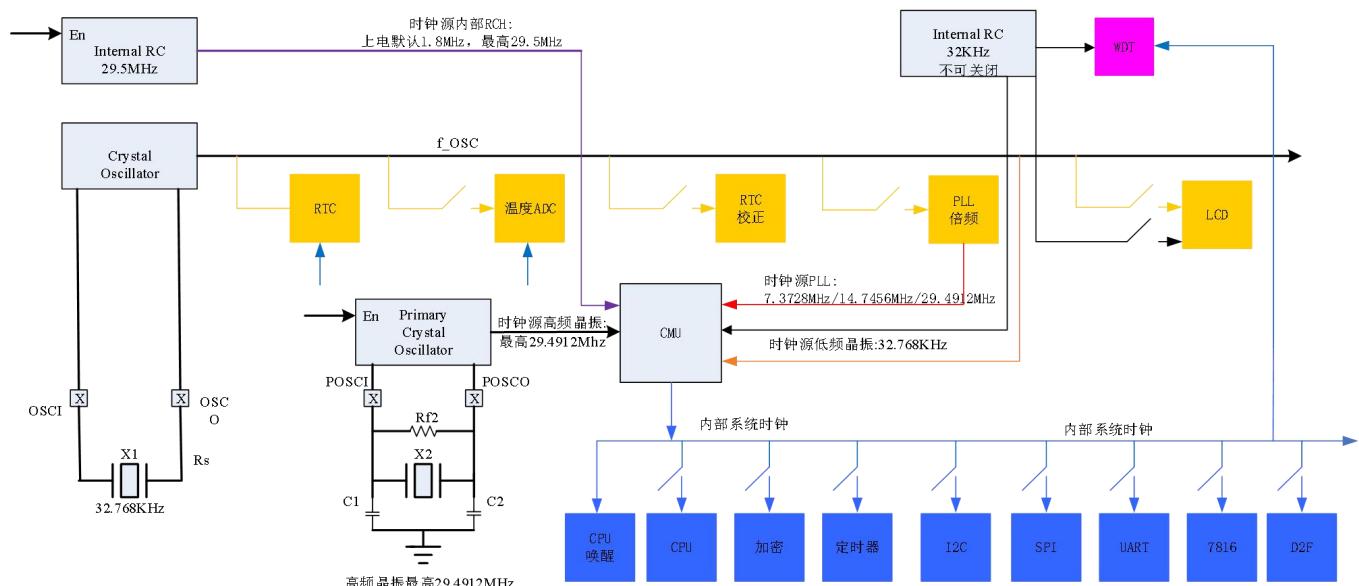


图 3-2 时钟源

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

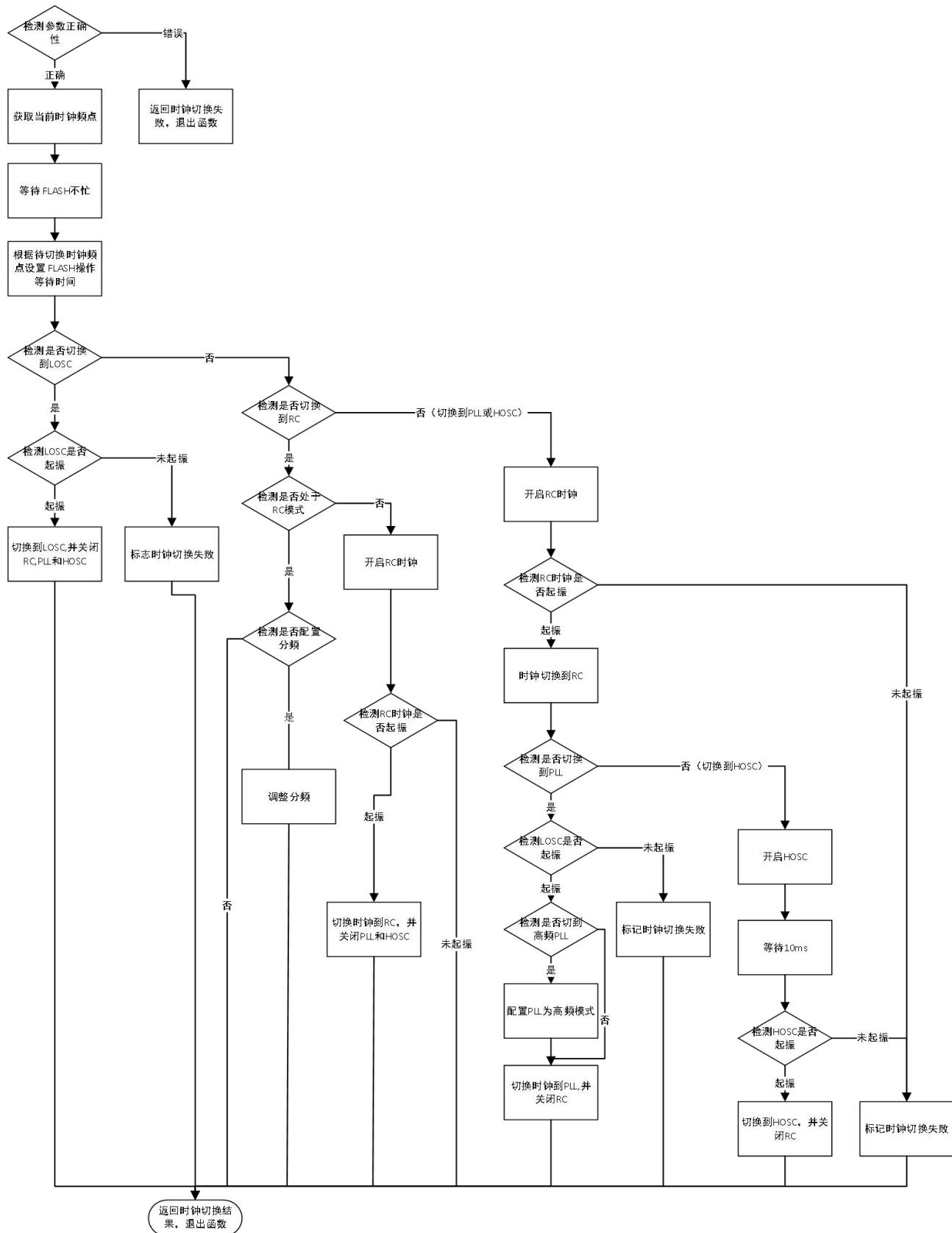


图 3-3 库函数时钟切换流程图

3.4 MCU 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，MCU 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 8.5uA 左右；

用户可根据 MCU 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 数字域（来自 VCC 供电）	
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 休眠后自动休眠
WDT	开启，不可关断
其他外设	关闭，可关时钟，永不掉电
VCC 域	
1.5V LDO	开启，不可关断
RCL	开启，不可关断
RCH	开启，可关电源
LCD	关闭，可关电源
LVD	关闭，可关电源
比较器 CMP1	关闭，可关电源
比较器 CMP2	开启，可关电源
复位电路	一直开启
PLL	关闭，可关电源
HOSC	关闭，可关电源
VBAT 域	
RTC	开启，万年历不可关断
温度测量 ADC	定时开启
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

内置三个上电复位 (POR) 电路和两个掉电复位 (BOR) 电路，分别对 VCC、LDO15 (1.5V LDO) 及 VBAT 进行监测，其中 VBAT 内只有 POR 电路。

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。

3.6 掉电处理

MCU 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_CTL1	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_CTL	0xC	ADC 控制寄存器

OSC_CTL2	0x10	系统 OSC 控制寄存器 2
SYS_RST	0x14	系统复位寄存器
MAP_CTL	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	器件 ID 寄存器
SYS_PS	0x30	系统控制寄存器密码保护位
IRFR_CTL	0x34	RCH 模式下红外时钟分频系数
TRIM_CFG1	0x78	时钟 TRIM 配置寄存器
TRIM_START	0x7C	时钟 TRIM 结果寄存器
FAB_UID0	0xF0	芯片唯一码寄存器 0
FAB_UID1	0xF4	芯片唯一码寄存器 0

只有当 SYS_PS (0x30) = 8'h82 时, 0x00~0x28、0x34/0x78/0x7C 寄存器才可写。

3.7.1 系统 OSC 控制寄存器 OSC_CTL1(0x00)

偏移地址 0x00

比特位	名称	描述	读/写标志	复位值
31:16	---	预留	R	0
15:11	CLOCK_FLAG	时钟开启标志位: 如果时钟是开启了, 则此位为 1: { HOSC, RCL, RCH, PLL, LOSC }	R	01101
10:8	SYCLK_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: 保留	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; 系统运行在其他时钟时, 该位为 0。	R	1
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;	R/W	01

		10: 以 PLL、HOSC 的四分频作为 CPU 主时钟; 11: 以 PLL、HOSC (时钟频率选择为 14MHz 和 29MHz 时) 的八分频作为 CPU 主时钟, 若 PLL、HOSC 为 7.3728MHz 表示为四分频; 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。		
1	IRCH_PD	内部 RCH 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(0x04)

偏移地址 0x04

比特位	名称	描述	读/写标志	复位值
31:6	---	预留	R	0
5	FLASH_BUSY	Flash busy 状态, 不能进入模式切换: 0: idle 1: busy	R	0
4	Reserved	预留	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 (0x08)

偏移地址 0x08

比特位	名称	描述	读/写标志	复位值
31:17	---	预留	R	0
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	Reserved	预留	R/W	0
14	Reserved	预留	R/W	0
13	Reserved	预留	R/W	0

12	Reserved	预留	R/W	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	Reserved	预留	R/W	0
6	Reserved	预留	R/W	0
5	CMP2_PD	比较器 2 电源开关 0: 上电 1: 掉电 缺省开启。 如果要使用该功能, 还需要对 GPIO 复用寄存器进行配置。	R/W	0
4	CMP1_PD	比较器 1 电源开关 0: 上电 1: 掉电 如果要使用该功能, 还需要对 GPIO 复用寄存器进行配置。	R/W	1
3	LVD_PD	LVD 电源开关 0: 上电	R/W	1

		1: 掉电 如果要使用该功能, 还需要对 GPIO 复用寄存器进行配置。		
2:0	Reserved	预留	R	0

3.7.4 系统 OSC 控制寄存器 2OSC_CTL2(0x10)

偏移地址 0x10

比特位	名称	描述	读/写标志	复位值
31:19	保留	保留		
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	Reserved	预留	R	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_STB	=0: 低频运行模式 LCM 选择 LOSC 外部低频晶振作为时钟源; =1: 低频运行模式 LCM 选择 RCL 内部低频晶振作为时钟源; 该寄存器只能在 RC 模式或高频模式修改。	R/W	0
11:10	Reserved	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	Reserved	预留	R	0
8	Reserved	预留	R	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; 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	R/W	1
2:0	HOSC_FREQ	000: 外接高频晶振7.3728MHz 001: 外接高频晶振14.7456MHz 010: 保留 011: 外接高频晶振29.4912MHz	R/W	000

系统时钟配置真值表 (PLL 和 HOSC 模式下) :

PLL_HOSC_SE L	PLL_FRE Q	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

3.7.5 系统复位寄存器 SYS_RST(0x14)

偏移地址 0x14

(此寄存器的 10 到 5 位只有上电复位可以复位这些位)

比特位	名称	描述	读/写标志	复位值
31:11	Reserved	预留	R	0
10	boi_vbat_ie	VBAT 域掉电中断使能位： =0 关闭中断 =1 使能中断	R/W	0
9	boi_vbat_flag	=0, 正常 =1, VBAT 曾发生过掉电 写 1 清零	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/W	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/W	0
6	PIN_RST	外部管脚复位表示： =1 表示发生过该复位, =0 表示没有发生。写 1 清零	R/W	0

5	POWEROK_RST	电源复位标志 =1 表示发生过该复位, =0 表示没有发生。写 1 清零	R/W	1
4:3	Reserved	预留	R	0
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	Reserved	预留	R	0
0	Reserved	预留	R	0

3.7.6 系统映射控制寄存器 SYS_MAPCTL(0x18)

偏移地址 0x18

比特位	名称	描述	读/写标志	复位值
31:3	Reserved	预留	R	0
2:0	REMAP	地址映射： 000: FLASH 映射在 0 地址 (正常模式) 001: FLASH 映射在 1/2 容量地址 (仅 512KB 容量支持该功能) 010: FLASH 与 SRAM 映射地址互换 011: BOOTROM 映射在 0 地址 100: FLASH 映射在 1/2 容量地址 (仅 512KB 容量支持该功能) 其他: 保留.	R/W	000

3.7.7 模块使能 0 寄存器 MOD0_EN (0x1c)

偏移地址 0x1C

比特位	名称	描述	读/写标志	复位值
31:21	---	预留	R	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	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

		步关闭此时钟: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能		
15	SPI_EN	SPI 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
14	I2C_EN	I2C 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
13	ISO7816_EN	ISO7816 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
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
10	UART2_EN	UART2 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
9	UART1_EN	UART1 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
8	UART0_EN	UART0 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
7	UART5_EN	UART5 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
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 模块使能清零, 时钟门控:	R/W	0

		0: 时钟停止, 模块清零 1: 时钟启动, 模块使能		
4	TC0_EN	TC0 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	---	预留	R/W	0
2	Reserved	预留	R	0
1	Reserved	预留	R	0
0	Reserved	预留	R	0

3.7.8 模块使能 1 寄存器 MOD1_EN(0x20)

偏移地址 0x20

比特位	名称	描述	读/写标志	复位值
31:16	Reserved	预留	R	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; 1: 时钟启动;	R/W	0
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	--	保留, soc 为 NVM 寄存器时钟门控, 默认为 0	R	0
11	CMPLVD_EN	CMPLVD 模块使能清零, apb 时钟门控: 0: 时钟停止 1: 时钟启动	R/W	0
10	RTC_SAR_EN	RTC/SAR apb 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	R/W	1
9	WDT_EN	WDT apb 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	R/W	1
8	Reserved	预留	R	0
7	Reserved	预留	R	0
6	LCD_EN	LCD 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	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, 模块清零	R/W	0

		1: 时钟启动, 模块使能		
4	---	保留	R	0
3	---	保留	R	0
2	---	保留	R	0
1	---	保留	R	0
0	---	保留	R	0

3.7.9 INTC 使能寄存器 INTC_EN(0x24)

偏移地址 0x24

比特位	名称	描述	读/写标志	复位值
31:9	Reserved	预留	R	0
8	INTC_EN	INTC apb 模块时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止 1: 时钟启动	R/W	0
7	INTC7_EN	INTC7 模块使能清零, 时钟门控: 0: 时钟停止 1: 时钟启动	R/W	0
6	INTC6_EN	INTC6 模块使能清零, 时钟门控: 0: 时钟停止 1: 时钟启动	R/W	0
5	INTC5_EN	INTC5 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	INTC4 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	INTC 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	INTC2_EN	INTC 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	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

3.7.10 KBI 使能寄存器 KBI_EN(0x28)

偏移地址 0x28

比特位	名称	描述	读/写标志	复位值
31:9	Reserved	预留	R	0

8	KBI_EN	KBI apb 模块时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止 1: 时钟启动	R/W	0
7	KBI7_EN	KBI7 模块使能清零, 时钟门控: 0: 时钟停止 1: 时钟启动	R/W	0
6	KBI6_EN	KBI6 模块使能清零, 时钟门控: 0: 时钟停止 1: 时钟启动	R/W	0
5	KBI5_EN	KBI5 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	KBI4_EN	KBI4 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	KBI0_EN	KBI0 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

3.7.11 器件 ID 寄存器 CHIP_ID(0x2C)

比特位	名称	描述	读/写标志	复位值
31:24	---	预留	R	0
15:0	CHIP_ID	芯片型号: V1 版: 8213 V2 版: 8220	R	xxxx

3.7.12 系统控制密码寄存器 SYS_PS(0x30)

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:0	SYS_PS	当 SYS_PS=0x82 时, 0x00~0x28、0x34/0x78/0x7C 寄存器可写; 当 SYS_PS= 其他值时, 0x00~0x28、0x34/0x78/0x7C 寄存器不可写;	R/W	00

		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。		
--	--	--------------------------------------	--	--

3.7.13 红外配置寄存器 IRFR_CTL (0x34)

比特位	名称	描述	读/写标志	复位值
31:6	---	预留	R	0
5:0	IRFR_CYCLE	该寄存器的密码是 0x82 RCH 模式下, 红外时钟分频系数, 基于 RCH 为 3.6864MHz。 ... 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; ... 计算公式为: IRFR 值=RCH 频率 29.5M/32/38K	R/W	0x18

3.7.14 时钟校正配置寄存器 TRIM_CFG1(0x78)

比特位	名称	描述	读/写标志	复位值
31:29	---	预留	R	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	被校正时钟源选择 0: 被校正时钟源选择 RCH; 1: 被校正时钟源选择 RCL;	R/W	0
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	参考时钟计数值	R/W	0x10000

3.7.15 时钟校正启动寄存器 TRIM_START(0x7C)

比特位	名称	描述	读/写标志	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;	R/W	0

		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。		
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	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准时钟返回的计数值 CAL_CLK_CNT=0x 61A80, 十进制为 400000;
5. 那么测量到的 RCH 频率值为:

$$\begin{aligned}
 & (\text{CAL_CLK_CNT}/\text{REF_CLK_CNT}) * 32768\text{Hz} \\
 & = (400000/4096) * 32768\text{Hz} \\
 & = 3200000\text{Hz} \\
 & = 3.2\text{MHz}
 \end{aligned}$$

3.7.16 DMA 优先级配置寄存器 DMA_PRI(0x80)

比特位	名称	描述	读 / 写 标志	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	保留	保留	R	0x0
25:24	DMA_CH12_PRI	通道 12:CPU 访问 SRAM 优先级配置 优先级配置同 CH0	R/W	0x0
23:22	DMA_CH11_PRI	通道 11:M2M 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:DSP 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:UART5 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:UART4 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:UART3 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:UART2 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:UART1 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:UART0 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:SPI3 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:SPI2 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:SPI1 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:SPI0 DMA 访问 SRAM 优先级配置 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/W	0x0

备注:

1、 系统上一共有 13 个主机访问 SRAM, 分别是 SPI0-DMA、SPI1-DMA、SPI2-DMA、SPI3-DMA、UART0-DMA、UART1-DMA、UART2-DMA、UART3-DMA、UART4-DMA、UART5-DMA、DSP-DMA、M2M-DMA、ARM-M0 CPU;

2、当不同主机访问相同 SRAM 时, 会产生竞争关系, 系统会根据优先级配置进行仲裁, 优先级高主机通道优先获得 SRAM 的访问权限。

- a) DMA_PRI 寄存器定义主机访问 SRAM 的优先级, 优先级策略为: 可配置优先级和通道号绝对优先级。
- b) 每个通道优先级配置有 2 个 bit, 可配置 4 级优先级, 优先级顺序为 3>2>1>0, 默认为 0。
- c) 优先级相同时, 通道号小的优先级高

3.7.17 芯片唯一码寄存器 0 FAB_UID0(0xF0)

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

3.7.18 芯片唯一码寄存器 1 FAB_UID1(0xF4)

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

4 处理器架构

4.1 概述

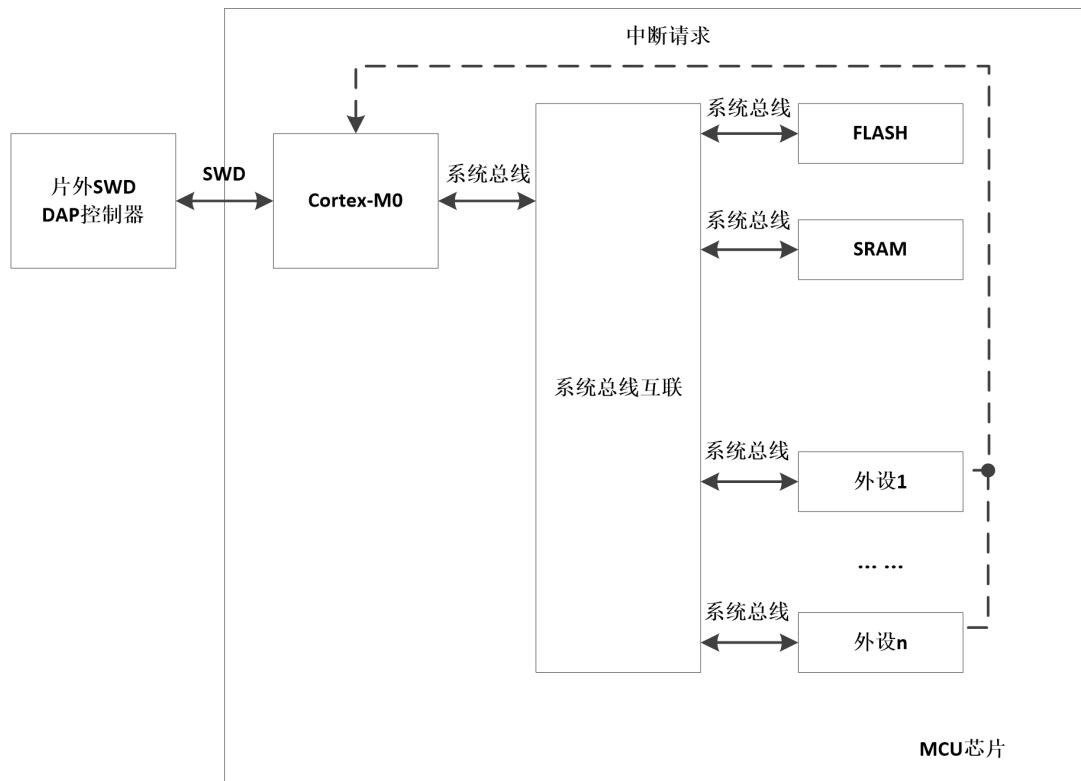
有二种方式（二个主设备）可以发起对 MCU 内置设备的访问：

- 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；

MCU 内置的从设备资源包括存储器（FLASH、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。

图 4-1 MCU 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 ARM Cortex 系列处理器，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
- CM0 内嵌系统定时器，具体请参考 ARM 文档。

有关 Cortex-M0 的详细资料可参阅 ARM 文档。

4.2.1 中断配置

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 4-1 中断/异常向量表及其配置信息

异常编号	中断编号	向量名称	地址	优先级	激活方式
-	-	MSP 初始值	0x00	-	-
1	-	复位	0x04	-3, 最高	与处理器异步
2	-14	不可屏蔽中断	0x08	-2	与处理器异步
3	-13	硬件故障	0x0C	-1	与处理器同步
4-10	-12~6	保留	0x10~0x28	-	
11	-5	系统调用	0x2C	可配置	与处理器同步
12-13	-4~3	保留	0x30~0x34	-	
14	-2	PendSV	0x38	可配置	与处理器异步
15	-1	系统节拍计数器	0x3C	可配置	与处理器异步
16	0	系统控制	0x40	可配置	与处理器异步
17	1	比较器 1 和 2、LVD	0x44	可配置	与处理器异步
18	2	保留	0x48	-	
19	3	RTC	0x4C	可配置	与处理器异步
20	4	D2F	0x50	-	
21	5	多路复用 ADC	0x54	可配置	与处理器异步
22	6	UART0	0x58	可配置	与处理器异步

23	7	UART1	0x5C	可配置	与处理器异步
24	8	UART2	0x60	可配置	与处理器异步
25	9	UART3	0x64	可配置	与处理器异步
26	10	SPI0	0x68	可配置	与处理器异步
27	11	I2C	0x6C	可配置	与处理器异步
28	12	7816_0/SPI3	0x70	可配置	与处理器异步
29	13	7816_1/SPI2	0x74	可配置	与处理器异步
30	14	TC0	0x78	可配置	与处理器异步
31	15	TC1	0x7C	可配置	与处理器异步
32	16	UART4	0x80	可配置	与处理器异步
33	17	UART5	0x84	可配置	与处理器异步
34	18	看门狗	0x88	可配置	与处理器异步
35	19	KBI	0x8C	可配置	与处理器异步
36	20	LCD	0x90	可配置	与处理器异步
37	21	加密 SEA	0x94	可配置	与处理器异步
38	22	保留	0x98	-	
39	23	SPI1	0x9C	-	
40	24	外部中断 0/INTn (INT0~7 合并)	0xA0	可配置	与处理器异步
41	25	外部中断 1/SIMP_TC0	0xA4	可配置	与处理器异步
42	26	外部中断 2/SIMP_TC1	0xA8	可配置	与处理器异步
43	27	外部中断 3/SIMP_TC2	0xAC	可配置	与处理器异步
44	28	外部中断 4/SIMP_TC3	0xB0	可配置	与处理器异步
45	29	外部中断 5/M2M	0xB4	可配置	与处理器异步

46	30	外部中断 6	0xB8	可配置	与处理器异步
47	31	外部中断 7	0xBC	可配置	与处理器异步

4.3 MCU 存储映射

MCU 的存储映射请参考“图 4-2MCU 存储映射”。

MCU 的高速外设包括:

- GPIO
- 加密 SEA

MCU 的低速外设包括:

- 6 个 UART 接口
- 2 个 32 位定时器
- 4 个 16 位简单定时器
- 4 个 SPI 接口
- 1 个 I2C 接口
- 1 个 KEY 控制器
- 1 个通用 ADC 接口
- 1 个看门狗单元
- 1 个系统控制单元
- 2 个 7816 接口
- 1 个 RTC 接口
- 1 个中断控制接口
- 1 个 LCD 控制器
- 1 个 D2F 能量积分单元
- 1 个 M2M 内存搬运模块
-

4.3.1 MCU 存储映射

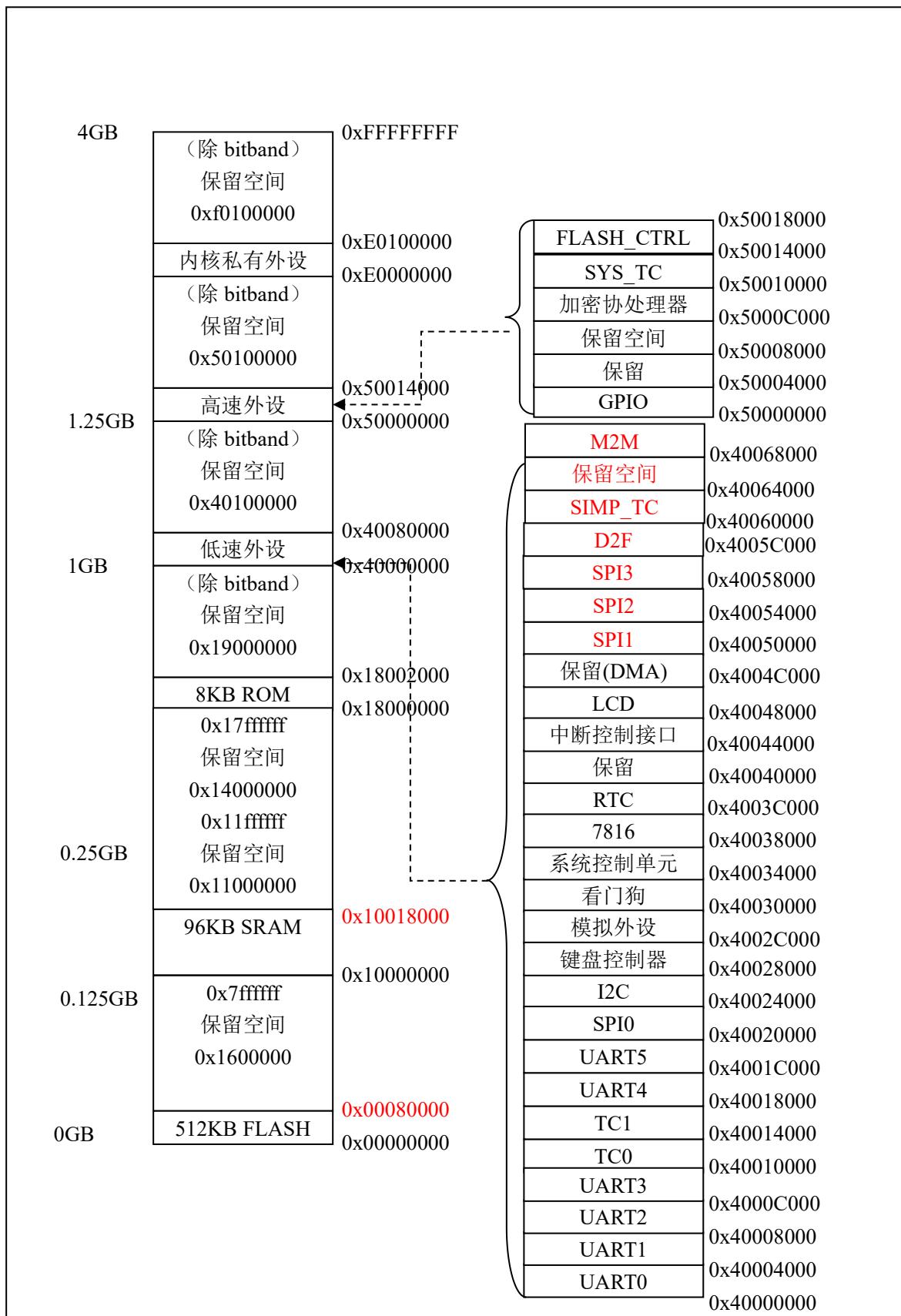


图 4-2 MCU 存储映射

4.3.2 存储重映射

MCU 支持对 2 个存储区域，包括 FLASH、SRAM 的基地址互相交换。交换操作通过配置系统控制器中的 REMAP 寄存器完成。除 FLASH、SRAM 之外的其他设备的地址分配均不受存储重映射的影响。

表 4-2 存储重映射配置

存储器设备	REMAP	映射地址
FLASH	0	0x00000000
	1	0x00000000
	2	0x10000000
	3	0x18000000
SRAM	0	0x10000000
	1	0x10000000
	2	0x00000000
	3	0x10000000

4.3.3 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：

$$Z = (X \& 0xFC\ 000000) + 0x02000000 + (Y \ll 2) + ((X \ll 5) \& 0x03\ FFFF)$$

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 0x52000000~0x5207FFFE0; (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
GPIO	0x50000000	0x52000000	除位操作 (SET/CLR) 以外寄存器 注：vbat 域内 GPIO 不支持 bitband
M2M	0x40068000	0x42f80000	CTRL、CLK、BDDIV 寄存器支持
SIMP_TC	0x40060000	0x42c00000	CTRL 寄存器
D2F	0x4005c000	0x4b800000	
SPI3	0x40058000	0x4b000000	
SPI2	0x40054000	0x4a800000	
SPI1	0x40050000	0x42a00000	SPI_CTRL 和 SPI_DMA_CTRL 寄存器
LCD	0x40048000	0x42900000	除了 LCD_STATUS 以外寄存器
INTC	0x40044000	0x42280000	---
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	CTRL
SPI0	0x40020000	0x42400000	SPI_CTRL 和 SPI_DMA_CTRL 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

4.3.4 SRAM

- 系统 SRAM

片内 SRAM 最大容量为 96KB，地址为 0x1000_0000~0x1001_8000，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。

- 加密算法 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。

4.3.5 FLASH

片内最大 512KB FLASH，支持：

- 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)

4.3.6 EEPROM

该模块为 512k bits 的串行带电可擦可编程只读存储器 (EEPROM)，共有 65536 个字节，每个字节 8bits。内置存储空间共分为 512 页，每页 128 字节，需要使用 16 位的数据字地址进行随机字寻址。该 EEPROM 提供了一个额外的 128 字节的识别页，该页可在后续设置中被永久锁定为只读模式，可用于存储敏感的应用参数。

➤ 容量

- 512 Kbits (64 Kbytes) 的存储空间；
- 128 bytes 的额外识别页；

- 额外识别页写锁定（永久只读）；

➤ 寿命

- 1 百万次写循环；
- 数据保存年限：100 年；

➤ 读操作

- 支持随机和顺序读取模式；

➤ 写操作

- 写操作小于 3ms；
- 页写操作小于 3ms；
- 支持分区页写；

4.4 中断应用

在头文件中加入 SOC 头文件 `#include <RN8xxx.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_DisableIRQ(IRQn_t IRQn)`。

5 RTC

5.1 概述

BaseAddr 为: 0x4003C000;

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。

该模块由 VBAT 独立供电。

5.2 特点

- 提供准确的温度值，-25°C~70°C范围内测温精度为±1°C
- 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 个晶振停振中断功能，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 32768Hz
- 可输出校正后的频率 1Hz/4Hz/8Hz/16Hz/(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

5.3 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_CTL	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	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_DOTA0	Offset+0x48	RTC 初始偏差校正寄存器
LOSC_CFG1	Offset+0x6C	32768 振荡器配置寄存器
RTC_CALPS	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	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, 二进制补码格式; 补偿温度范围: T < -30 度
RTC_CAL_T1	Offset+0xD4	补偿温度范围: -30 <= T < -20 度
RTC_CAL_T2	Offset+0xD8	补偿温度范围: -20 <= T < -10 度
RTC_CAL_T3	Offset+0xDC	补偿温度范围: -10 <= T < 0 度
RTC_CAL_T4	Offset+0xE0	补偿温度范围: 0 <= T <= 10 度
RTC_CAL_T5	Offset+0xE4	补偿温度范围: 35 < T <= 45 度
RTC_CAL_T6	Offset+0xE8	补偿温度范围: 45 < T <= 55 度
RTC_CAL_T7	Offset+0xEC	补偿温度范围: 55 < T <= 65 度
RTC_CAL_T8	Offset+0xF0	补偿温度范围: 65 < T <= 75 度
RTC_CAL_T9	Offset+0xF4	补偿温度范围: 75 < T <= 85 度
LOSC_CFG1	Offset+0x6C	LOSC 配置寄存器
VBAT_IOEN	Offset+0x88	VBAT 域 IO 口配置寄存器, 可配置 P44 P45 P56 的 IO 功能, 具体定义参考寄存器描述。
VBAT_IOMODE	Offset+0x8C	VBAT_IOEN 写使能及模式, 写 VBAT_IOEN 前需先配置该寄存器, 具体定义参考寄存器描述。
P44N0_TIME0	Offset+0x90	P44 口第一次下降沿冻结的时间寄存器
P44N0_TIME1	Offset+0x94	P44 口第一次下降沿冻结的时间寄存器

P44N0_TIME2	Offset+0x98	P44 口第一次下降沿冻结的时间寄存器
P44P0_TIME0	Offset+0x9C	P44 口第一次上升沿冻结的时间寄存器
P44P0_TIME1	Offset+0xA0	P44 口第一次上升沿冻结的时间寄存器
P44P0_TIME2	Offset+0xA4	P44 口第一次上升沿冻结的时间寄存器
P44N1_TIME0	Offset+0xA8	P44 口第二次下降沿冻结的时间寄存器
P44N1_TIME1	Offset+0xAC	P44 口第二次下降沿冻结的时间寄存器
P44N1_TIME2	Offset+0xB0	P44 口第二次下降沿冻结的时间寄存器
P44P1_TIME0	Offset+0xB4	P44 口第二次上升沿冻结的时间寄存器
P44P1_TIME1	Offset+0xB8	P44 口第二次上升沿冻结的时间寄存器
P44P1_TIME2	Offset+0xBC	P44 口第二次上升沿冻结的时间寄存器
P45N0_TIME0	Offset+0x100	P45 口第一次下降沿冻结的时间寄存器
P45N0_TIME1	Offset+0x104	P45 口第一次下降沿冻结的时间寄存器
P45N0_TIME2	Offset+0x108	P45 口第一次下降沿冻结的时间寄存器
P45P0_TIME0	Offset+0x10C	P45 口第一次上升沿冻结的时间寄存器
P45P0_TIME1	Offset+0x110	P45 口第一次上升沿冻结的时间寄存器
P45P0_TIME2	Offset+0x114	P45 口第一次上升沿冻结的时间寄存器
P45N1_TIME0	Offset+0x118	P45 口第二次下降沿冻结的时间寄存器
P45N1_TIME1	Offset+0x11C	P45 口第二次下降沿冻结的时间寄存器
P45N1_TIME2	Offset+0x120	P45 口第二次下降沿冻结的时间寄存器
P45P1_TIME0	Offset+0x124	P45 口第二次上升沿冻结的时间寄存器
P45P1_TIME1	Offset+0x128	P45 口第二次上升沿冻结的时间寄存器
P45P1_TIME2	Offset+0x12C	P45 口第二次上升沿冻结的时间寄存器

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

5.3.1 RTC 寄存器组

5.3.1.1 RTC 控制寄存器 RTC_CTL(0x00)

偏移地址 0x00

比特位	名称	描述	读/写标志	复位值
31:12	---	预留	R	0
11	---	保留位, 不要配置	R/W	0
10	Cal_busy	RTC 校正计算忙标识。 =1 时表示 RTC 正在进行校正计算; =0 时表示 RTC 校正计算完成。 该位只读, 每次校正时会置高, 无实际意义。	R	0
9	Wr_busy	RTC 寄存器写操作忙。写多个寄存器, 不需要等待。 但是写入后读出, 需要等待 busy 位从 1 变为 0 后, 才能读回正确值。 因此建议应用程序在写万年历后统一等待 300us 后再读取。	R	0
8	WRTC	RT 寄存器组写允许:	R/W	0

		<p>0: 禁止 RTC 寄存器写操作； 1: 允许 RTC 寄存器写操作。</p> <p>注意：该位对 RTC 寄存器组 00~1C/3C/C4 有效，对 RTC_CTRL[7:0]也有效。</p> <p>写万年历时间寄存器的两种方法：</p> <ol style="list-style-type: none"> 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。 <p>注 1: 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”。</p> <p>注 2: 万年历时间寄存器的写入必须按照上述顺序来，并且一次时间写入要写完整的 6 个寄存器（年月日时分秒），不能仅仅更新部分（比如只写分和秒）。</p>										
7:6	TSE	<p>温度传感器允许位</p> <p>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 MCU 测量得到。</p> <p>注 1: 用户温补模式 0 下，需要等待 RTC 校正不忙时，才可启动温补，即要等待 RTC_CTRL 寄存器的 bit10 为 0 时，才能写温度寄存器。</p> <p>注 2: 用户温补模式 0 下，需要等待 SAR 不忙时，才可启动温补，即要等待 RTC_CTRL 寄存器的 bit10 和 SAR_STATUS 寄存器的 bit1 都为 0 时，才能写温度寄存器。</p>	R/W	00								
5:3	TCP	<p>温度补偿周期：</p> <table> <tr> <td>000:2S</td> <td>001:10S 默认</td> </tr> <tr> <td>010:20s</td> <td>011:30s</td> </tr> <tr> <td>100:1 分钟</td> <td>101:2 分钟</td> </tr> <tr> <td>110:5 分钟</td> <td>111:10 分钟</td> </tr> </table>	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
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											
02:00	FOUT	<p>000: 禁止输出 001: 1Hz 输出 010: 1/30Hz 输出 011: 32768Hz 输出 100: 16Hz 输出 101: 8Hz 输出 110: 4Hz 输出 111: 保留</p>	R/W	000								

注：只有上电复位，会将该寄存器恢复到默认值，其它复位不会恢复。

5.3.1.2 秒寄存器 RTC_SC(0x04)

偏移地址 0x04

比特位	名称	描述	读/写标志	复位值
31:07	Reserved	预留	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	-

5.3.1.3 分钟寄存器 RTC_MN(0x08)

偏移地址 0x08

比特位	名称	描述	读/写标志	复位值
31:07	Reserved	预留	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	-

5.3.1.4 小时寄存器 RTC_HR(0x0c)

偏移地址 0x0C

比特位	名称	描述	读/写标志	复位值
31:06	Reserved	预留	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	-

5.3.1.5 日期寄存器 RTC_DT(0x10)

偏移地址 0x10

比特位	名称	描述	读/写标志	复位值
31:06	Reserved	预留	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	-

5.3.1.6 月份寄存器 RTC_MO(0x14)

偏移地址 0x14

比特位	名称	描述	读/写标志	复位值
31:05	Reserved	预留	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	-

5.3.1.7 年份寄存器 RTC_YR(0x18)

偏移地址 0x18

比特位	名称	描述	读/写标志	复位值
31:08	Reserved	预留	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	-

5.3.1.8 星期寄存器 RTC_DW(0x1C)

偏移地址 0x1C

比特位	名称	描述	读/写标志	复位值
31:03	Reserved	预留	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	-

5.3.1.9 RTC 定时寄存器 1 RTC_CNT1(0x20)

偏移地址 0x20

比特位	名称	描述	读/写标志	复位值
31:09	Reserved	预留	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: 当RTC-IE-> RTC_1S_SEL为0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RTC-IE-> RTC_1S_SEL为1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1。	R/W	0

5.3.1.10 RTC 定时寄存器 2 RTC_CNT2(0x24)

偏移地址 0x24

比特位	名称	描述	读/写标志	复位值
31:10	Reserved	预留	R	0
09	CNT2_CLK_SEL	CNT2_CLK 的时钟来源选择 0: 32768Hz 1: RCL	R/W	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

5.3.1.11 秒闹钟寄存器 RTC_SCA(0x28)

偏移地址 0x28

比特位	名称	描述	读/写标志	复位值
31:07	Reserved	预留	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

5.3.1.12 分钟闹钟寄存器 RTC_MNA(0x2c)

偏移地址 0x2C

比特位	名称	描述	读/写标志	复位值
31:07	Reserved	预留	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

5.3.1.13 小时闹钟寄存器 RTC_HRA(0x30)

偏移地址 0x30

比特位	名称	描述	读/写标志	复位值
31:06	Reserved	预留	R	0
05:00	HRA	小时闹钟值 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。	R/W	0

注: 04~30H 寄存器没有复位值。

5.3.1.14 RTC 中断使能寄存器 RTC_IE(0x34)

偏移地址 0x34

比特位	名称	描述	读/写标志	复位值
31:10	Reserved	预留	R	0
9	RTC_1S_SEL	秒中断来源选择:	R/W	0

		=0: 根据系统时钟模式: 高频模式下选择 pll_1hz, 非高频模式下选择 RTC 秒脉冲 =1: 一直选择 RTC 的秒脉冲 注: 芯片复位或 pll 时钟重新启动后, pll_1hz 和秒寄存器更新不同步, 差异随机; 写秒寄存器后, 偏差随时间从 0.5S 开始递增; RTC 秒脉冲与秒寄存器同步。建议客户配置该寄存器为 1, 选择 RTC 的秒脉冲。		
08	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 0 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 RTC-IE-> RTC_1S_SEL 为 1 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 1。	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 1: 使能	R/W	0

5.3.1.15 RTC 中断标志寄存器 RTC_IF(0x38)

偏移地址 0x38

比特位	名称	描述	读/写标志	复位值
31:12	Reserved	预留	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	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

5.3.1.16 当前温度寄存器 RTC_TEMP(0x3c)

偏移地址 0x3C

比特位	名称	描述	读/写标志	复位值
31:10	Reserved	预留	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</p> <p>若符号位为 1, 则温度=(2^10- 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 /2^6=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)

5.3.1.17 当前温度寄存器 2 RTC_TEMP2(0xF8)

Offset = 0xF8

比特位	名称	描述	读/写标志	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	-

5.3.1.18 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	WO	0

		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。		
--	--	--	--	--

5.3.1.19 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] // 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。	R	0

5.3.1.20 初始频率偏差校正寄存器 RTC_DOTA0 (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 $\sigma_0 = -6$ ppm，校正值 DOTA0 = 0x1f40。 注：大部分 32768 晶振，频率偏差典型值为 ± 20 ppm，最大值为 ± 30 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 data);”，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。

5.3.1.21 LOSC 配置寄存器 LOSC_CFG1(0x6c)

比特位	名称	描述	读/写标志	复位值
31:11	---	预留	R	0
10	VCC_BOI_FLAG	=0: 正常; =1, VCC 电源域曾发生过掉电; 写 1 清零	R/W	0
9	LOSC_FAIL	=0: 正常; =1: LOSC 曾发生过停振 写 1 清零。	R/W	0
8	LOSC_WEN	=0: LOSC_PD 不可写 =1: LOSC_PD 可写	R/W	0
7:0	LOSC_PD	LOSC 使能位: =其他: 打开; =8'ha8: 关闭。 备注: 同时也是外灌时钟使能信号, =1 使能外灌时钟。 关闭 LOSC 并使能外灌功能步骤: LOSC_CFG1 = (1<<8); LOSC_CFG1 = (0xa8<<0);	R/W	0

注: 如果使用外灌 32.768kHz 时钟, 需要从 XI 灌入, 同时要关闭芯片的起振电路 (即关闭 LOSC 使能位)。

5.3.1.22 VBAT 域 IO 口配置寄存器 VBAT_IOEN(0x88)

比特位	名称	描述	读/写标志	复位值
31:12	---	预留	R	0
11:8	P45_KEY	开盖检测(P45)配置 =4'h5, 使能 P45 口按键自动记录功能; =4'hE, 对 0xC0~0xEC 寄存器清零; =其他值, 表示不使能按键自动记录功能。 该功能与 GPIO 章节 P45 口的相关输入功能可同时存在。	R/W	0
7:4	P56_RTCOUT	=4'ha 时, 使能 P56 口输出 RTC_OUT 功能; =其他值, P56 功能由 GPIO 章节寄存器定义。 该功能的优先级高于 GPIO 章节对于 P56 的复用配置。 当 VBAT 独立上电而 VCC 无电时, P56 默认输出 1Hz; 当 VCC 上电后, CPU 初始化会将此寄	R/W	0

		存器配置为 0。		
3:0	P44_KEY	<p>开盖检测(P44)配置 4'h5, 使能 P44 口按键自动记录功能; 4'hE, 对 0x90~0xBC 寄存器清零; 其他值, 表示不使能按键自动记录功能。 该功能与 GPIO 章节 P44 口的相关输入功能可同时存在。</p>	R/W	0

5.3.1.23 VBAT 域 IO 口配置寄存器写使能 VBAT_IOMODE(0x8C)

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:4	IOWEN	=4'hB 时, 使能 VBAT_IOEN 寄存器写操作; =其他, 不允许写 VBAT_IOEN 寄存器。	R/W	0
3:2	---	预留	R	0
1	P45_KEYMODE	<p>Bit1: 开盖检测(P45)模式配置 0: 记录 IO 开盖检测使能后最早两次按键事件; 1: 记录 IO 开盖检测使能后最晚两次按键事件, 最晚两次记录时没有区分缓存中存的时间先后, 应由软件做判断。 使能 KEY 功能前, 应先对 KEYMODE 做配置。</p>	R/W	0
0	P44_KEYMODE	<p>IO 口模式选择: Bit0: 开盖检测(P44)模式配置 0: 记录 IO 开盖检测使能后最早两次按键事件; 1: 记录 IO 开盖检测使能后最晚两次按键事件, 最晚两次记录时没有区分缓存中存的时间先后, 应由软件做判断。 使能 KEY 功能前, 应先对 KEYMODE 做配置。</p>	R/W	0

5.3.2 二次分段补偿寄存器组

5.3.2.1 二次补偿密码寄存器 RTC_CALPS(0xCC)

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	R/W	0

	寄存器才会起作用。	
--	-----------	--

5.3.2.2 二次分段补偿寄存器组 RTC_CAL_T0~T9(0xD0~0Xf4)

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	<p>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；</p> <p>T0 补偿温度范围：T < -30°C</p> <p>T1 补偿温度范围：-30°C <= T < -20°C</p> <p>T2 补偿温度范围：-20°C <= T < -10°C</p> <p>T3 补偿温度范围：-10°C <= T < 0°C</p> <p>T4 补偿温度范围：0°C <= T <= 10°C</p> <p>T5 补偿温度范围：35°C < T <= 45°C</p> <p>T6 补偿温度范围：45°C < T <= 55°C</p> <p>T7 补偿温度范围：55°C < T <= 65°C</p> <p>T8 补偿温度范围：65°C < T <= 75°C</p> <p>T9 补偿温度范围：T > 75°C</p> <p>备注：实际设计，上述范围应该加上 0.1875°C，但在应用层不需要体现，手册按照上述描述。</p>	R/W	0

5.3.3 开盖检测寄存器组

5.3.3.1 P44 第一次冻结时间

冻结时间数据 16bit 有效，均为 BCD 码格式。

地址	名称	描述	读/写标志	复位值
0x90	P44N0_TIME0	P44 第一次下降沿冻结时间分钟、秒寄存器 [15:8]: 分钟 [7:0]: 秒	R	0
0x94	P44N0_TIME1	P44 第一次下降沿冻结时间日、小时寄存器 [15:8]: 日 [7:0]: 小时	R	0
0x98	P44N0_TIME2	P44 第一次下降沿冻结时间年、月寄存器 [15:8]: 年 [7:0]: 月	R	0
0x9C	P44P0_TIME0	P44 第一次上升沿冻结时间分钟、秒寄存器 [15:8]: 分钟 [7:0]: 秒	R	0
0xA0	P44P0_TIME1	P44 第一次上升沿冻结时间日、小时寄存器 [15:8]: 日 [7:0]: 小时	R	0
0xA4	P44P0_TIME2	P44 第一次上升沿冻结时间年、月寄存器 [15:8]: 年 [7:0]: 月	R	0

5.3.3.2 P44 第二次冻结时间

冻结时间数据 16bit 有效, 均为 BCD 码。

地址	名称	描述	读/写标志	复位值
0xA8	P44N1_TIME0	P44 第二次下降沿冻结时间分钟、秒寄存器 [15:8]: 分钟 [7:0]: 秒	R	0
0xAC	P44N1_TIME1	P44 第二次下降沿冻结时间日、小时寄存器 [15:8]: 日 [7:0]: 小时	R	0
0xB0	P44N1_TIME2	P44 第二次下降沿冻结时间年、月寄存器 [15:8]: 年 [7:0]: 月	R	0
0xB4	P44P1_TIME0	P44 第二次上升沿冻结时间分钟、秒寄存器 [15:8]: 分钟 [7:0]: 秒	R	0
0xB8	P44P1_TIME1	P44 第二次上升沿冻结时间日、小时寄存器 [15:8]: 日 [7:0]: 小时	R	0
0xBC	P44P1_TIME2	P44 第二次上升沿冻结时间年、月寄存器 [15:8]: 年 [7:0]: 月	R	0

5.3.3.3 P45 第一次冻结时间

冻结时间数据 16bit 有效, 均为 BCD 码。

地址	名称	描述	读/写标志	复位值
0x100	P45N0_TIME0	P45 第一次下降沿冻结时间分钟、秒寄存器 [15:8]: 分钟 [7:0]: 秒	R	0
0x104	P45N0_TIME1	P45 第一次下降沿冻结时间日、小时寄存器 [15:8]: 日 [7:0]: 小时	R	0
0x108	P45N0_TIME2	P45 第一次下降沿冻结时间年、月寄存器 [15:8]: 年 [7:0]: 月	R	0
0x10C	P45P0_TIME0	P45 第一次上升沿冻结时间分钟、秒寄存器 [15:8]: 分钟 [7:0]: 秒	R	0
0x110	P45P0_TIME1	P45 第一次上升沿冻结时间日、小时寄存器 [15:8]: 日 [7:0]: 小时	R	0
0x114	P45P0_TIME2	P45 第一次上升沿冻结时间年、月寄存器 [15:8]: 年	R	0

		[7:0]: 月		
--	--	----------	--	--

5.3.3.4 P45 第二次冻结时间

冻结时间数据 16bit 有效, 均为 BCD 码。

地址	名称	描述	读/写标志	复位值
0x118	P45N1_TIME0	P45 第二次下降沿冻结时间分钟、秒寄存器 [15:8]: 分钟 [7:0]: 秒	R	0
0x11C	P45N1_TIME1	P45 第二次下降沿冻结时间日、小时寄存器 [15:8]: 日 [7:0]: 小时	R	0
0x120	P45N1_TIME2	P45 第二次下降沿冻结时间年、月寄存器 [15:8]: 年 [7:0]: 月	R	0
0x124	P45P1_TIME0	P45 第二次上升沿冻结时间分钟、秒寄存器 [15:8]: 分钟 [7:0]: 秒	R	0
0x128	P45P1_TIME1	P45 第二次上升沿冻结时间日、小时寄存器 [15:8]: 日 [7:0]: 小时	R	0
0x12C	P45P1_TIME2	P45 第二次上升沿冻结时间年、月寄存器 [15:8]: 年 [7:0]: 月	R	0

5.3.1

二次曲线补偿寄存器组

5.3.1.1 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

5.3.1.2 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。

5.3.1.3 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 = 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 $\sigma_0 = -6$ ppm, 校正值 DOTA0=0x1f40。 注: 大部分 32768 晶振, 频率偏差典型值为 ± 20 ppm, 最大值为 ± 30 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。

5.3.1.4 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°C 到 T0 这一段的温度系数。（格式为无符号二进制） ALPHAL = round($\alpha * 2048 * 16$) (单位为 ppm/°C ² , 一般温度系数为 -0.034ppm/°C ²) (α 可表示范围: 0 – 0.0625 ppm/°C ²)	R/W	0

只有上下电复位能起作用，bootrom 会修改该值。

$0.034 * 2048 * 16 * (T - T0) ^ 2 = 0.034 * 2048 * 16 * (-40 - 25) ^ 2 = 4707123.2$ 约为 143.65ppm

$round(0.034 * 2048 * 16) * (T - T0) ^ 2 = round(0.034 * 2048 * 16) * (-40 - 25) ^ 2 = 1114 * 4225 = 4706650$ 约为 143.635ppm

误差为 0.02ppm

5.3.1.5 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°C 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/°C ² , 一般温度系数为 -0.034ppm/°C ²) (α 可表示范围: 0 – 0.0625 ppm/°C ²)	R/W	0

只有上下电复位能起作用，bootrom 会修改该值。

5.3.1.6 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°C) (T_0 可表示范围: 0 – 31.5°C) (分辨度为 0.25 度。实际上应该写入 25±5 的值。)	R/W	-

5.3.1.7 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°C/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°C 温度偏差为 260-290k) (1°C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°C)	R/W	-

5.3.1.8 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	-

5.3.2 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。

5.3.2.1 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
------	-----------	---	-----	---

5.3.2.2 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	-

5.3.2.3 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 -0.017 ppm/ $^{\circ}\text{C}$, 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: $\text{ALPHA} = \text{round}(\alpha * 2^{15*8})$ 如果 α 为负, 那么: $\text{ALPHA} = \text{round}(2^{16} + \alpha * 2^{15*8})$ 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: ± 0 ~0.125 ppm/ $^{\circ}\text{C}$ 一个 LSB 代表: $1/2^{18} = 3.8 * 10^{-6}$ ppm/ $^{\circ}\text{C}$, 量化误差最大影响为: $\text{LSB} * (40+25) = 0.00025$ ppm, 可以忽略。	R/W	0

5.3.2.4 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	晶振频率温度系数的二次系数 β , 典型值为 -0.0334 ppm/ $^{\circ}\text{C}^2$, 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/ $^{\circ}\text{C}^2$) β 正常都为负, 寄存器应该写入: $\text{BETA} = \text{round}(-\beta * 2^{15*16})$	R/W	0

		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/ $^{\circ}$ C, 注意 bit15 只能为 0。 一个 LSB 代表: $1/2^{19}=1.9073*10^{-6}$ ppm/ $^{\circ}$ C, 量化误差最大影响为: $LSB*(40+25)^2=0.00806$ ppm, 可以忽略。		
--	--	---	--	--

5.3.2.5 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	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 $-4.9*10^{-5}$ ppm/ $^{\circ}$ C ³ , 最大 $-4.4*10^{-5}$ ppm/ $^{\circ}$ C ³ , 最小 -5.210^{-5} ppm/ $^{\circ}$ C ³ 。实测值为 $-6.3*10^{-5}$ ppm/ $^{\circ}$ C ³ 。(格式为仍然使用有符号二进制表示) γ 正常都为负, 那么寄存器应该写入: $GAMMA = round(2^{16} + \gamma * 2^{15} * 2048 * 4)$ γ 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: $\pm 0 \sim 12.207^{-5}$ ppm/ $^{\circ}$ C 一个 LSB 代表: $1/2^{28}=0.37253*10^{-8}$ ppm/ $^{\circ}$ C, 量化误差最大影响为: $LSB*(40+25)^3=0.001023$ ppm, 可以忽略。	R/W	0

5.3.2.6 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	晶振频率温度系数的四次系数 ζ , 实测值为 $4.692437*10^{-7}$ ppm/ $^{\circ}$ C ⁴ 。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) ζ 如果为正, 那么寄存器应该写入: $ZETA = round(\zeta * 2^{15} * 65536 * 8)$ ζ 如果为负, 那么寄存器应该写入: $ZETA = round(2^{16} + \zeta * 2^{15} * 65536 * 8)$ 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: $\pm 0 \sim 1.9075^{-6}$ ppm/ $^{\circ}$ 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 DataT 寄存器溢出, 并无实际意义。 一个 LSB 代表: $1/2^{34}=5.8208*10^{-11}$ ppm/ $^{\circ}$ C,	R/W	0

		量化误差最大影响为: $LSB \cdot (40+25)^4 = 0.001039\text{ppm}$, 可以忽略。		
--	--	---	--	--

注意: 参考资料所列参数推算为 $1.22 \cdot 10^{-8}$

5.3.3

结果输出寄存器组

5.3.3.1 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 符号 ppm 0.03125ppm 表示范围: 1024ppm ~ -1023.96785ppm 实际表示范围: Dotat 实际范围一般在 128ppm 以内。</p> <p>二次曲线计算原理: $\text{dotat}[15:0]/32 = \text{dota0}[12:0] / 32 + \alpha[10:0] / (2048 * 16) * (\text{temp}[9:0] / 4 - \{3'h0, \text{xt0}[6:0]\} / 4)^2$ $\text{dota0}[12:0] = \text{round}[\sigma_0 * 32] \quad \text{范围: } -128\text{ppm} \sim 127.96875\text{ppm}$ $\alpha[10:0] = \text{round}[\alpha * 2048 * 16] \quad \text{范围: } 0 \sim 0.034 \sim 0.0625$ $\text{temp}[9:0] = T * 4 \quad \text{范围: } -128 \text{ 度} \sim +127.75 \text{ 度}$ $\text{xt0} = T0 * 4 \quad \text{范围: } 0 \sim 31.5^\circ\text{C}$ $\text{Dota0} \text{ 一般为 } 20 \sim 30\text{ppm, } \alpha \text{ 为 } 0.034 \text{ 左右, } \text{temp} \text{ 范围是 } -40 \sim 85 \text{ 度, } \text{xt0} \text{ 一般为 } 25 \text{ 度左右。}$ </p> <p>四次曲线计算原理: $\text{dotat}[15:0]/32 = \text{dota0}[12:0]/32 + \alpha[15:0]/(2^{15} * 8) * (\text{temp}[9:0] / 4 - \text{xt1}[12:0] / 256) + \beta[14:0]/(-2^{15} * 16) * (\text{temp}[9:0] / 4 - \text{xt1}[12:0] / 256)^2 + \gamma[15:0]/(2^{15} * 2048 * 4) * (\text{temp}[9:0] / 4 - \{\text{xt1}[12:0] / 256\})^3 + \zeta[15:0]/(2^{15} * 65536 * 8) * (\text{temp}[9:0] / 4 - \{\text{xt1}[12:0] / 256\})^4$ $\text{dota0}[12:0] = \text{round}[\sigma_0 * 32] \quad \text{范围: } -128\text{ppm} \sim 127.96875\text{ppm}$ $\alpha[15:0] = \text{round}[\alpha * 2^{15} * 8] \quad \text{范围: } \pm 0 \sim 0.125 \text{ ppm/}^\circ\text{C}$ $\beta[14:0] = \text{round}(-\beta * 2^{15} * 16) \quad \text{范围: } 0 \sim -0.0625 \text{ ppm/}^\circ\text{C}$ $\gamma[15:0] = \text{round}(2^{16} + \gamma * 2^{15} * 2048 * 4) \quad \text{范围: } \pm 0 \sim 12.207 \sim -5 \text{ ppm/}^\circ\text{C}$ $\zeta[15:0] = \text{round}(2^{16} + \zeta * 2^{15} * 65536 * 8) \quad \text{范围: } \pm 0 \sim 1.9075 \sim -6 \text{ ppm/}^\circ\text{C}$ $\text{temp}[9:0] = T * 4 \quad \text{范围: } -128 \text{ 度} \sim +127.75 \text{ 度}$ </p>	R	-

	// xt1=T0*256	范围: 0 ~ 31.99609375°C		
--	---------------	-----------------------	--	--

5.3.3.2 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=fix(dotat/0.030518ppm) 有符号数, 代表 PLL 模式时每 1S 需要补偿的时钟周期数。	R	0

5.3.3.3 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=fix(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

5.3.3.4 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= (dotat - FDTR1S * 30.5175) 每隔 30S 需要多补偿的量 B=fix(A/1.017257) 每隔 30S 补偿寄存器的值 FDTR30S= FDRT1S +B 举例: 如果 dotat=50ppm, 那么 FDTR1S=1, 1S 补偿的余量 A= (50 - fix(50/30.5) * 30.5)=19.5 每隔 30S 需要多补偿的量 B=fix(19.5/1.01725)=19 每隔 30S 补偿寄存器的值 FDTR30S= FDRT1S +19=20 如果 dotat=-50ppm, 那么 FDTR1S=-1, 1S 补偿的余量 A= (-50 - fix(-50/30.5) * 30.5)=-19.5 每隔 30S 需要多补偿的量 B=fix(-19.5/1.01725)=-19 每隔 30S 补偿寄存器的值 FDTR30S= FDRT1S -19=-20 有符号数, 代表 RTC 秒脉冲每 30S 需要补偿的时钟周期数。	R	0

5.3.3.5 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 $A = (dotat - FDTR1S * 30.5175 - (FDTR30S - FDTR1S) * 1.01725)$ 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 $FDTR120S = FDTR30S + B$ 有符号数，代表 RTC 每 120S 需要补偿的时钟周期数	R		-

5.4 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。

5.5 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。

5.6 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 RTC->CNT1 = 0x00；即为 1S 产生 1 次中断。

4、设置 RTC->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。

6 WDT

内置硬件看门狗，用于检测程序的异常执行。

6.1 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；

6.2 看门狗定时器的配置

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 备注：采用内部低频 RCL 计时，具体时间并不精确，但可保证溢出时间比上述标准时间要长。	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	0

	(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。	
--	---	--

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:



6.3 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

6.3.1 WDT_EN (0x0)

比特位	名称	描述	读/写标志	复位值
31:9	---	预留	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

6.3.2

WDT_CTRL (0x04)

看门狗配置寄存器

Offset=0x04

比特位	名称	描述	读/写标志	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。

6.3.3

WDT_PASS (0x08)

看门狗密码寄存器

Offset=0x08

比特位	名称	描述	读/写标志	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

6.3.4

WDT_HALT (0x14)

看门狗 HALT 控制寄存器

Offset=0x14

比特位	名称	描述	读/写标志	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；

6.3.5

WDT_STBY (0x18)

看门狗 STBY 控制寄存器

Offset=0x18

比特位	名称	描述	读/写标志	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；

6.4 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 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。

7 LCD

MCU 内置段码式 LCD 控制器。

最大支持 8COM*36SEG，支持电荷泵和电阻分压两种方式。

7.1 概述

LCD 具有如下特性

- ◆ RN8318: 4*40、6*38、8*36; RN8615/RN8613/RN8612: 4*34、6*32、8*30
- ◆ 支持 Type A, Type B 两种驱动波形；
- ◆ 支持 1/3bias 和 1/4bias;
- ◆ 支持 static, 1/2, 1/3, 1/4, 1/6, 1/8duty;
- ◆ 支持 16 级对比度可调的 LCD 驱动模式；
- ◆ LCD 模块关闭后，所配置的各 COM、SEG 管脚自动下拉到地（**注意：停显时请保持 COM、SEG 状态不变，不要将管脚复用关系切换到 IO 口，也不要改变 bias 及 duty 配置**）。
- ◆ RN8318/RN8615/RN8613 支持 charge pump 和内部电阻串分压方式实现 LCD Bias 电压

◆ RN8612 只支持内部电阻串分压方式实现 LCD Bias 电压

7.1.1 扫描时钟频率

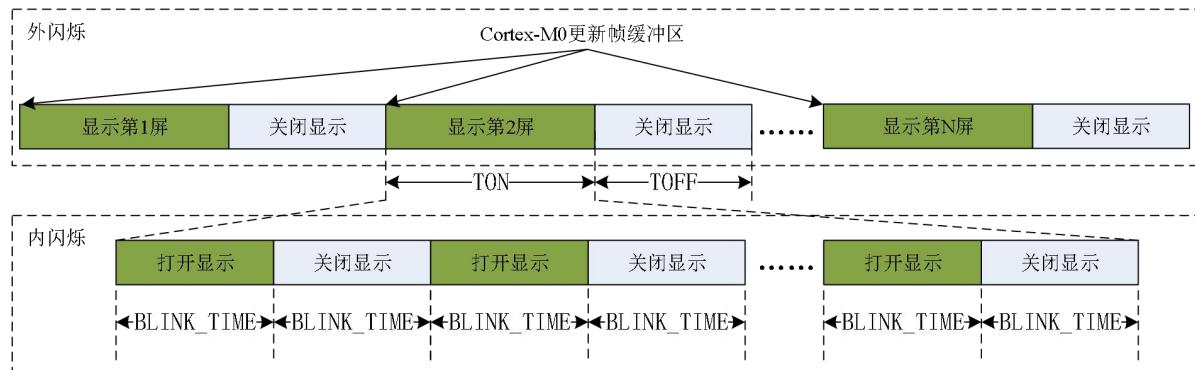
LCD 单元的时钟来自 fosc 或 RCL (由系统控制的 OSC_CTL2 寄存器选择)，频率为 32768Hz。Fosc 或 RCL 经过分频之后作为 LCD 波形扫描频率 flcd，波形扫描频率 flcd 可以通过寄存器 LCD_CLKDIV[7:0]进行配置。
 $F_{fram} = flcd / (\text{com 数目})$ 。一般要求 LCD 屏的刷新频率略微大于 60Hz。

表 7-1 示出不同 duty 对应的时钟设置。（绿色标注的为正常使用的帧频）

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

7.1.2 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 7-1 LCD 闪烁方式


如图 7-1, 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 TOFF 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。用户可以使用这些事件更新帧缓冲区。

7.1.2.1 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。

7.1.2.2 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。

7.1.3 LCD 驱动波形

LCD 驱动波形与软件设置的 LCD Type, Duty 数和 Bias 数 (LCD_CTL 寄存器) 有关。

MCU 支持 Type A 和 Type B 两种驱动波形。Type A 驱动为行反转方式, 即在 1 个 frame 的时间内完成一次正负驱动的交替; Type B 驱动为帧反转方式, 即需要在 2 个 frame 的时间内才能完成一次正负驱动的交替。具体可以参考本章节描述的 LCD 驱动波形。当驱动 Duty 数较高时, Type 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;

7.1.3.1 LCD 类型 A 驱动波形

图 7-2 LCD 驱动波形 (1/4 Duty, 1/3 Bias, Type A)

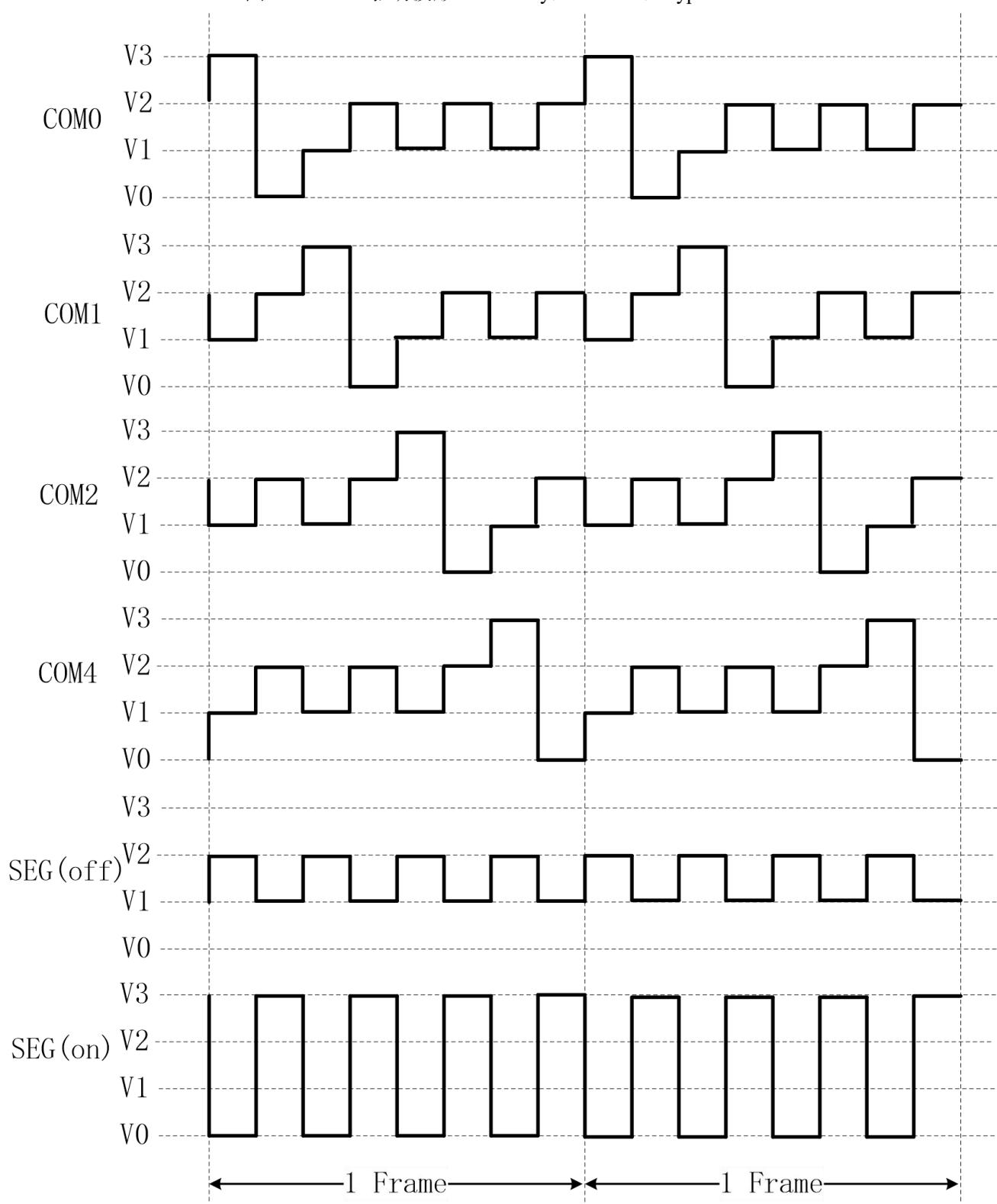
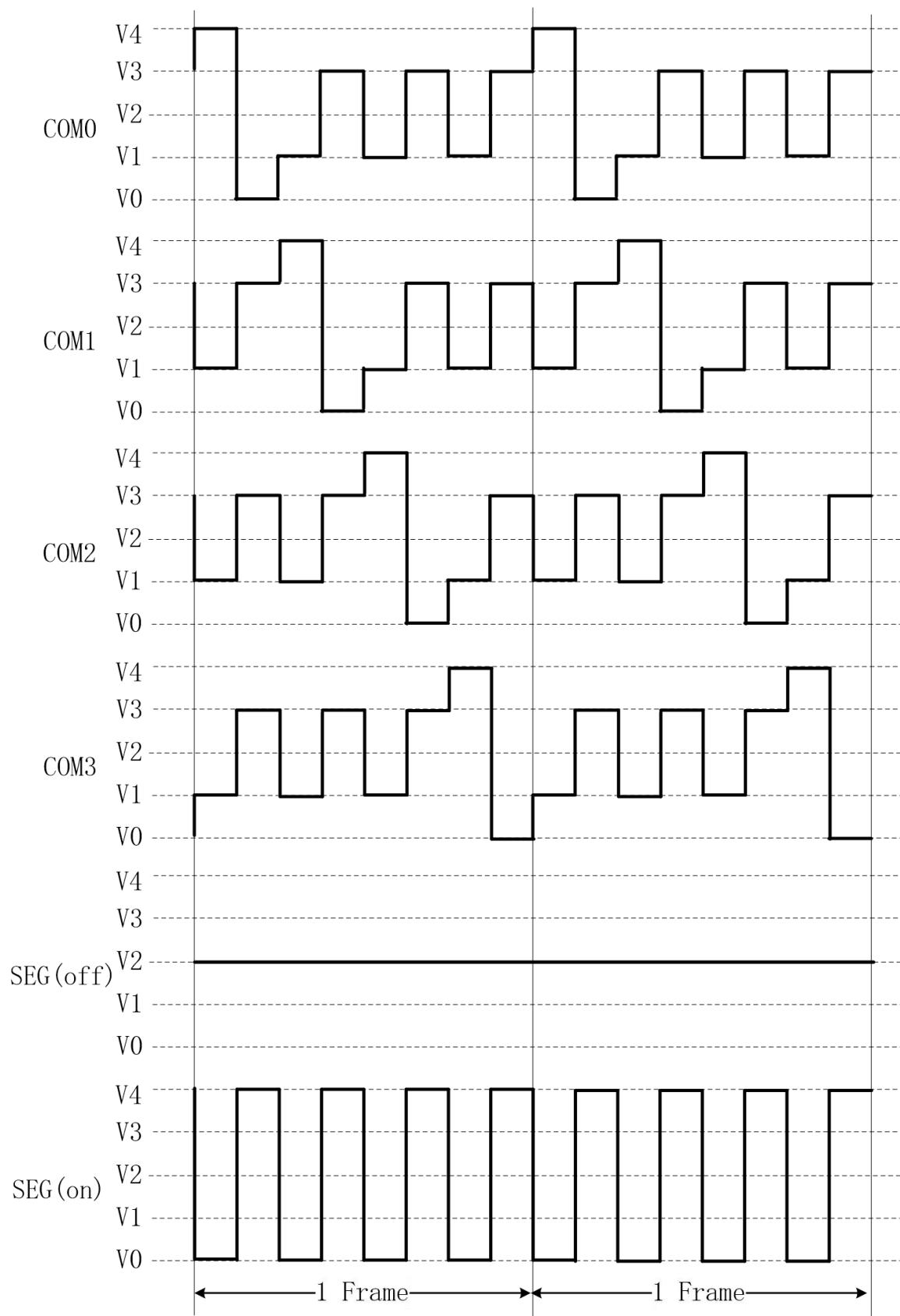


图 7-3 LCD 驱动波形 (1/4 Duty, 1/4 Bias, Type A)



7.1.3.2 类型 B 驱动波形

图 7-4 LCD 驱动波形 (1/4 Duty, 1/3 Bias, Type B)

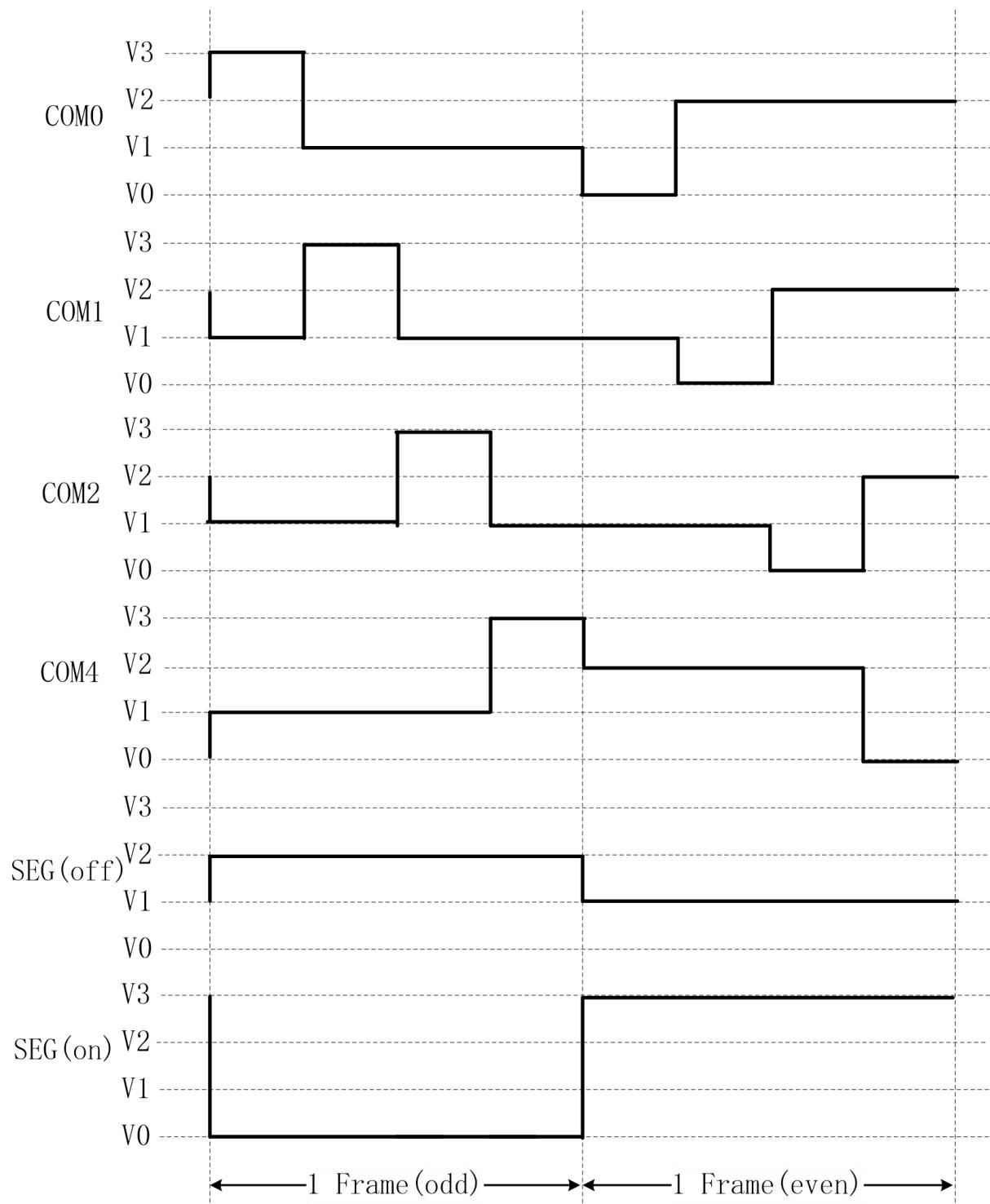
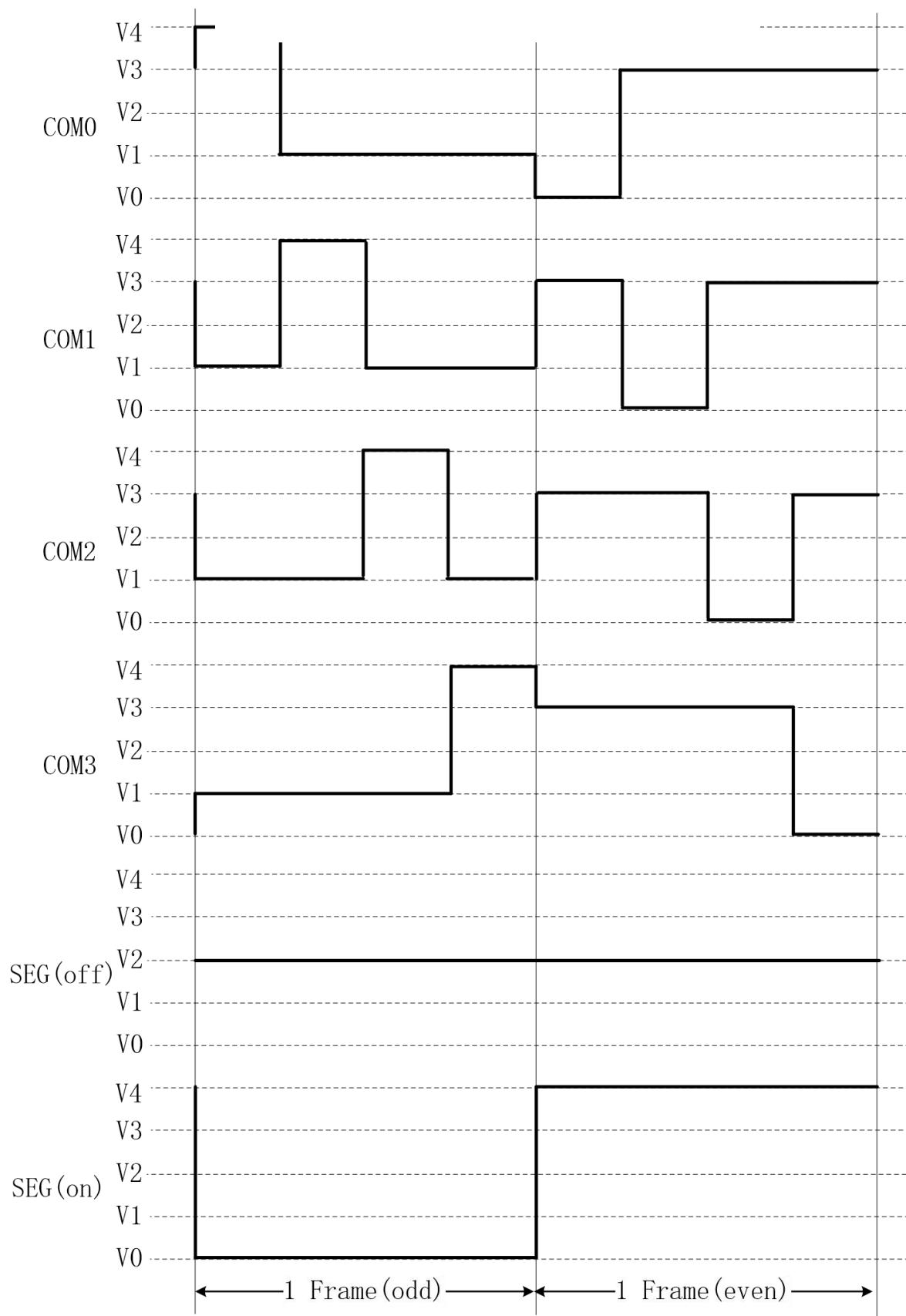


图 7-5 LCD 驱动波形 (1/4 Duty, 1/4 Bias, Type B)



7.1.4 偏置电压

7.1.4.1 Charge Pmup 方式提供 LCD 偏置电压

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 (Va, Vb, Vc, Vd) , 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 7-2 所示。

表 7-2 LCD 驱动电压与偏压比关系

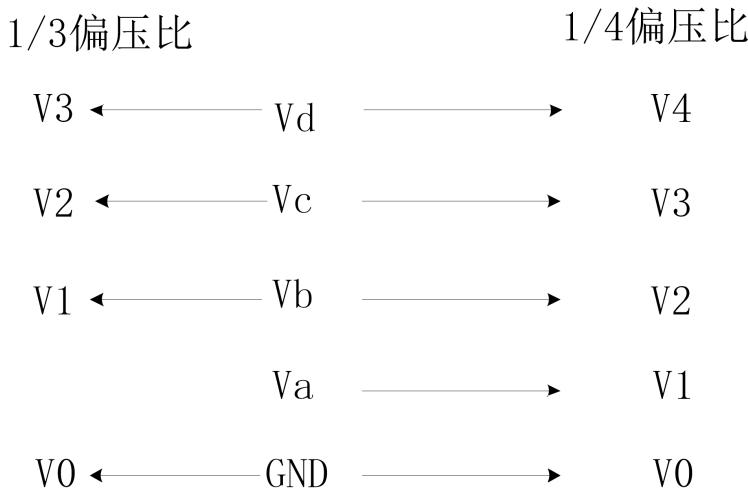
偏压比	灰度选择	Va	Vb	Vc	Vd	Vd (MAX)
1/3 偏压比	BIASLVL[5] = 0	Va = Vref*(32+BIASLVL[4:0])/63	Vb = Va	Vc = 2*Va	Vd = 3*Va	3.75 V
	BIASLVL[5] = 1	Va = Vref*(1 + BIASLVL[4:0]/63)	Vb = Va	Vc = 2*Va	Vd = 3*Va	5.59 V
1/4 偏压比	BIASLVL[5] = 0	Va = Vref*(32+BIASLVL[4:0])/63	Vb = 2*Va	Vc = 3*Va	Vd = 4*Va	5.0 V
	BIASLVL[5] = 1	Va = Vref*(1 + BIASLVL[4:0]/63)	Vb = 2*Va	Vc = 3*Va	Vd = 4*Va	6.032V

举例: LCD 所需的 Vd 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 选择内部 Vref 做为基准, Vref 输出典型值为 1.25V。

1/3 和 1/4 偏压比应用电压选择如图 7-6 所示。

图 7-6 偏置电压选择



7.1.4.2 内部电阻串分压方式提供 LCD 偏置电压

内置 LDO, 输出 2.7~3.6V 可调, 调节步距为 60mV, 支持 3.0V 和 3.3V LCD 屏; 具体配置参见 LCD 内部电阻串控制寄存器 (LCD_RESCTL)。

1/3 和 1/4 偏压比应用电压选择如图 7-6 所示。

LCD 内部电阻串模式支持两种模式, 大小电阻串分时开启和小电阻/开路分时开启模式, 推荐使用小电阻/开路模式, 需外接 470nf 电容滤波;

快速充电只是在第一次启动刷屏时才会起作用, 对正常的显示没有影响, 快速充电的目的是为了让片外的电容尽快的充满电, 默认情况下建议使能该位。

7.1.5 LCD BUFFER 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。

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

LCD_BUFX[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_BUFX, 最大可以支持 6*38 的 LCD 屏

LCD_BUFX[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_BUFX, 最大可以支持 4*40 的 LCD 屏

LCD_BUFX[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
------------------------------------	--------------------	--------------------	--------------------	--------------------	------------------	------------------	------------------	------------------

7.2 寄存器描述

LCD 模块的基址

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

LCD 模块的寄存器偏移地址

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

7.2.1 LCD 控制寄存器 LCD_CTL(0x00)

偏移地址 0x00

比特位	名称	描述	读/写标志	复位值
31:13	Reserved	预留	R	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; 备注: RN8612 为保留寄存器位, 用户不要操作。	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 备注: RN8612 为保留寄存器位, 用户不要操作。	R/W	0
4	BIAS	LCD Bias 控制 0: 1/3Bias 1: 1/4Bias	R/W	0
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

7.2.2 LCD 状态寄存器 LCD_STATUS(0x04)

偏移地址 0x04

比特位	名称	描述	读/写标志	复位值
31:7	Reserved	预留	R	0
6	LCD_BUSY	LCD Busy Bit 0: 不忙 1: 忙 Note:	R	0

		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。		
5	保留	保留寄存器位，用户不要操作		
4	保留	保留寄存器位，用户不要操作		
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

7.2.3 LCD 时钟控制寄存器 LCD_CLKDIV(0x08)

偏移地址 0x08

比特位	名称	描述	读/写标志	复位值
31:8	Reserved	预留	R	0
7:0	CLKDIV	LCD Clock 分频系数 LCD_CLK=fosc/(2*(CLKDIV+1)) (fosc 为 32768Hz)	R/W	0

7.2.4 LCD 闪烁控制寄存器 LCD_BLINK(0x0c)

偏移地址 0x0c

比特位	名称	描述	读/写标志	复位值
31:24	Reserved	预留	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	R/W	0

		BLINK_TIME 的 2n 倍 (n 为大于 0 的整数)。		
17:9	TOFF	步长是 0.25s, 支持 0~127.5s, 请设置成大于>3s 0.25*TOFF	R/W	0
8:0	TON	步长是 0.25s, 支持 0~127.5s, 请设置成大于>3s 0.25*TON	R/W	0

7.2.5 LCD 闪烁控制寄存器 LCD_PS(0x10)

LCD PUMP 建立时间寄存器

Offset=0x10

比特位	名称	描述	读/写标志	复位值
31:16	---	预留	R	0
15:0	PS	LCD PUMP Setup time Time = Tosc * (PS+4) (Tosc 为 30.5uS)	R/W	0xcc

备注：RN8612 不支持该寄存器。

7.2.6 LCD 内部电阻串控制寄存器 LCD_RESCTL(0x14)

偏移地址 0x14

比特位	名称	描述	读/写标志	复位值																																		
31:13	Reserved	预留	R	0																																		
12:9	LDOS	LDO 输出电平选择信号 LDO 输出作为 LCDVD 电压，并通过电阻分压产生 LCDVC、LCDVB、LCDVA； LDO 输出电平从 2.7V~3.6V, 共 16 档, 档位步进 0.06V； <table border="1" data-bbox="409 1170 811 1927"> <tr> <th>LDOS 配置</th> <th>LDO 输出电压 V</th> </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> 复位时 LDO 输出 3.0V	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	R/W	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	快速充电控制：	R/W	1																																		

		0: 关闭快速充电功能 1: 打开快速充电功能; 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。		
7	RES_AO	电阻串分时驱动开关 0: 电阻串分时驱动, 根据 RES_DT 和 RES_FT 产生驱动信号 1: 电阻串始终被驱动, RES_DT 和 RES_FT 配置无效。 选择 TYPE A 时, RES_AO 恒为 1。	R/W	1
6: 4	RES_DT	电阻串分压方式, 刷屏周期内, 20k 电阻驱动时间配置 驱动时间 $T_d = (BIAS_DT[2:0]+1) * T_{osc}$ (T_{osc} 为 30.5uS)。	R/W	000
3: 2	RES_FT	电阻串分压方式, 刷屏周期内, 20k 电阻驱动次数配置 00: 驱动 1 次 01: 驱动 2 次 10: 驱动 3 次 11: 驱动 4 次	R/W	00
1	RSM	内部电阻分压电阻模式选择信号 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	R/W	0
0	IRSN	电阻分压模式选择 0 内部电阻分压模式 1 保留 如果使用电阻分压方式, 该寄存器只能配置为 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。

7.2.7 LCD 数据寄存器 LCD_BUF[i] (8 位寄存器)

偏移地址 0x20–0x45

比特位	名称	描述	读/写标志	复位值
31:8	Reserved	预留	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

7.3 LCD 操作步骤

- 1、将系统控制章节 MOD1_EN 模块使能 1 寄存器第 6 位配置为 1, 打开 LCD 模块时钟门控。
- 2、将 LCD 控制寄存器 LCD_CTL 设置为 0, 关闭 LCD 模块。
- 3、等待 LCD_STATUS 状态寄存器第 6 位是否为 0, 空闲状态, 如空闲则进入下一步, 否则等待, 等待时

间大概为 16MS 左右。

- 4、设置 LCD 控制寄存器 LCD_CTL，设置 LCD 显示模式、液晶显示电压、开启 PUMP。例如使用 8COM, 5v 液晶，LCD_CTL 值为 0x3fb。也可配置 LCD_RESCTL，选择电阻分压方式。
- 5、设置 LCD 时钟控制寄存器 LCD_CLKDIV 设置 LCD 的分频系数。
- 6、设置 LCD 闪烁控制寄存器，如不需闪烁功能则可直接配置为 **0x01**。
- 7、完成初始化设置，可直接将显示数据送入 LCD_BUF[i] 中即可。

8 定时器

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

8.1 概述

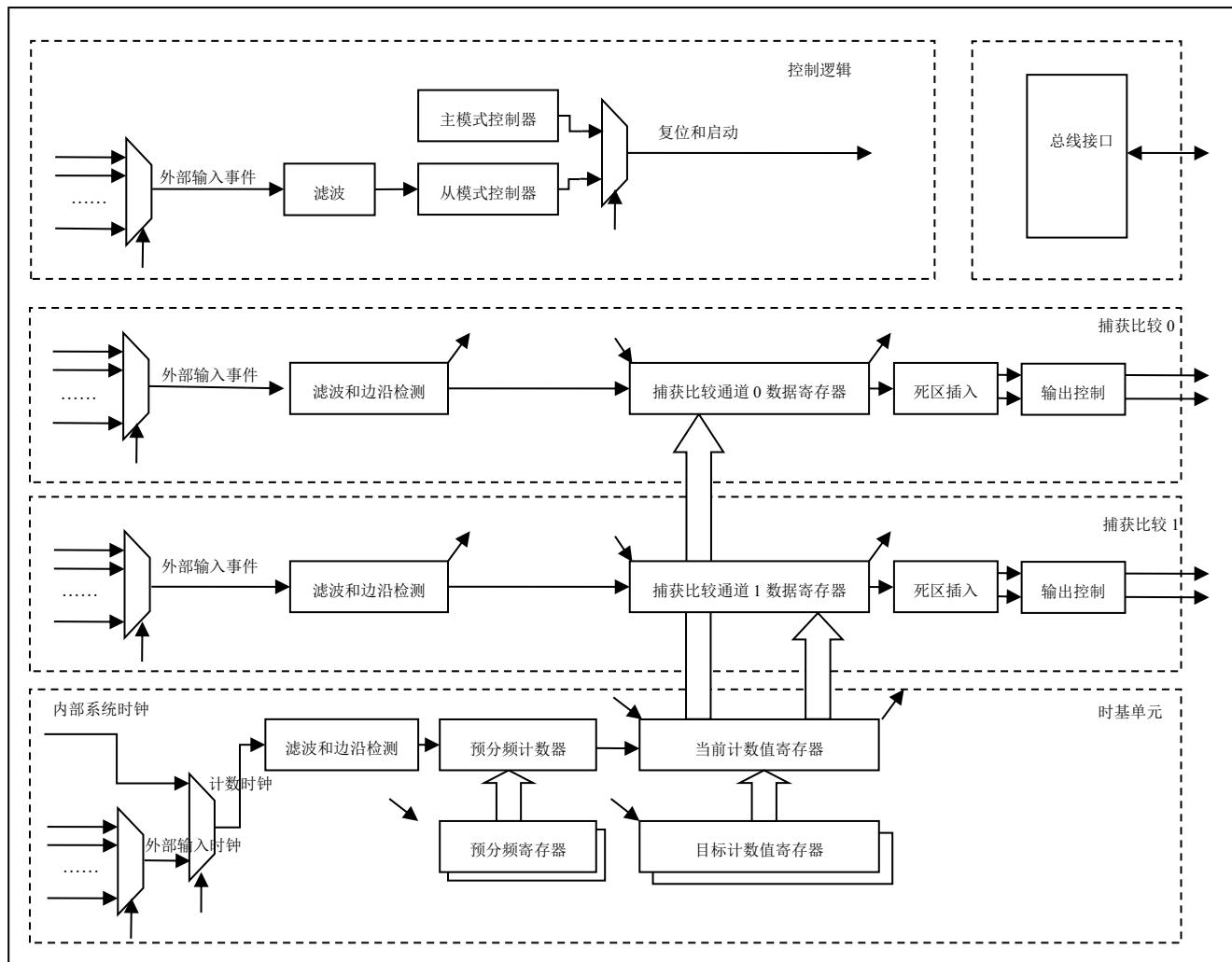
定时器具备如下特性：

- 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；

8.2 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。



8.3 寄存器描述

模块寄存器地址

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

8.3.1.1.1 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

8.3.1.1.2 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	-	保留	R	0
15:0	PS	分频系数, 分频值 (PS+1), 0 为不分频	R/W	0

8.3.1.1.3 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

8.3.1.1.4 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

8.3.1.1.5 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

8.3.1.1.6 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	-	保留	R	0

23:16	FLTOPT	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	-	保留	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: IOCNT0_OUT0 17: IOCNT2_OUTx 18: IOCNT1_OUTx 19: rtc_out 20: p1[0]外部IO口 21: p1[1] 外部IO口 22: p1[2]、P5[2]外部IO口 23: p1[3]、P5[3]外部IO口 24: p1[4]、P5[4]外部IO口 25: p1[5]、P5[5]外部IO口 26: p1[6]、P5[6]外部IO口 27: p1[7]、P5[7]外部IO口 28: p3[0]外部IO口 29: p3[1]外部IO口 30: p3[3]外部IO口 31: p3[5]外部IO口	R/W	0
7:2	-	保留	R	0
1	FLTEN	外部输入时钟滤波使能 0: 不使能 1: 使能	R/W	0
0	CM	计数时钟源选择: 0: 内部系统时钟	R/W	0

		1: 外部输入时钟 (由 CS 选择时钟源)		
--	--	------------------------	--	--

8.3.1.1.7 控制寄存器 TC_CR (0x020)

比特位	名称	描述	读/写标志	复位值
31:29	-	保留	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	--	保留	R/W	0
26	--	保留	R/W	0
25	--	保留	R/W	0
24	--	保留	R/W	0
23:21	--	保留	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	SLVFLTOPT	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	-	保留	R	0
0	START	定时器启动: 0: 停止 1: 启动	R/W	0

8.3.1.1.8 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	-	保留	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	FLTOPT	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

8.3.1.9 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	-	保留	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

8.3.1.10 状态寄存器 TC_STA (0x30)

比特位	名称	描述	读/写标志	复位值
31:4	-	保留	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

8.4 典型应用

8.4.1 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。
- 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。

8.4.2 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：

- 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 配置为捕获模式，CPOL 选择捕获极性，CS 选择外部输入事件。
- 3、中断使能寄存器中使能捕获比较通道 0/1 中断使能。
- 4、控制寄存器，启动定时器。

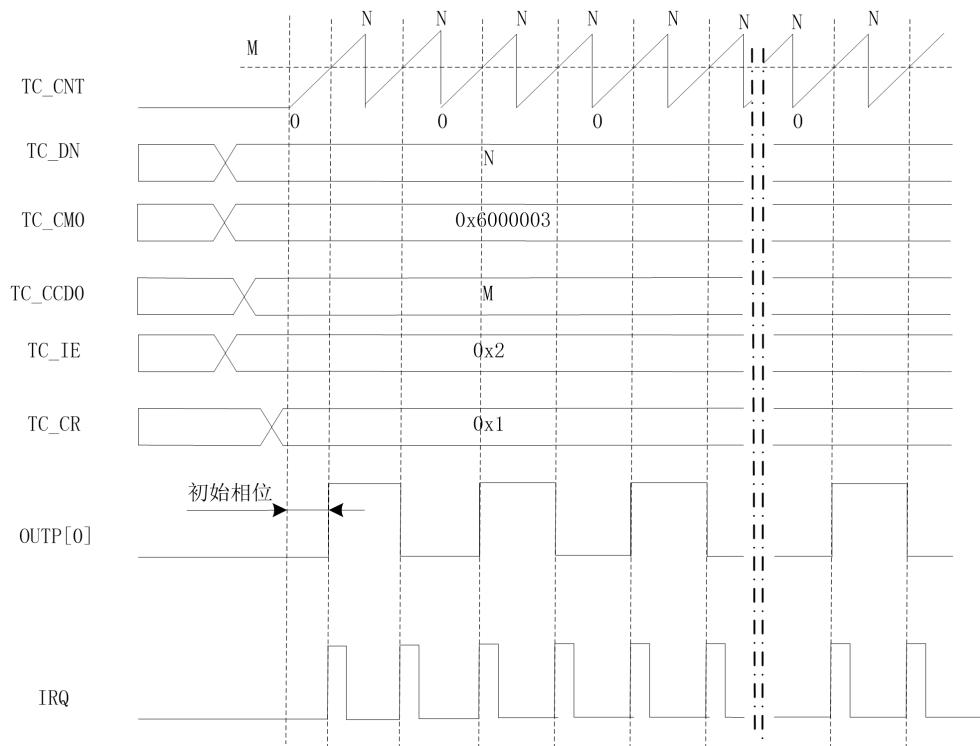
定时器捕获到外部输入事件的捕获极性时，产生中断，同时当前计数值会保存在捕获比较通道数据寄存器中。若采用两个通道，一个通道捕获上升沿，一个通道捕获下降沿，则可通过两个通道的数据寄存器计算出脉冲宽度。

常用可选择功能配置说明：

- 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、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 FLTOPT 设置滤波周期数。
- 4、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。

8.4.3 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 FLTOPT 设置滤波周期数。

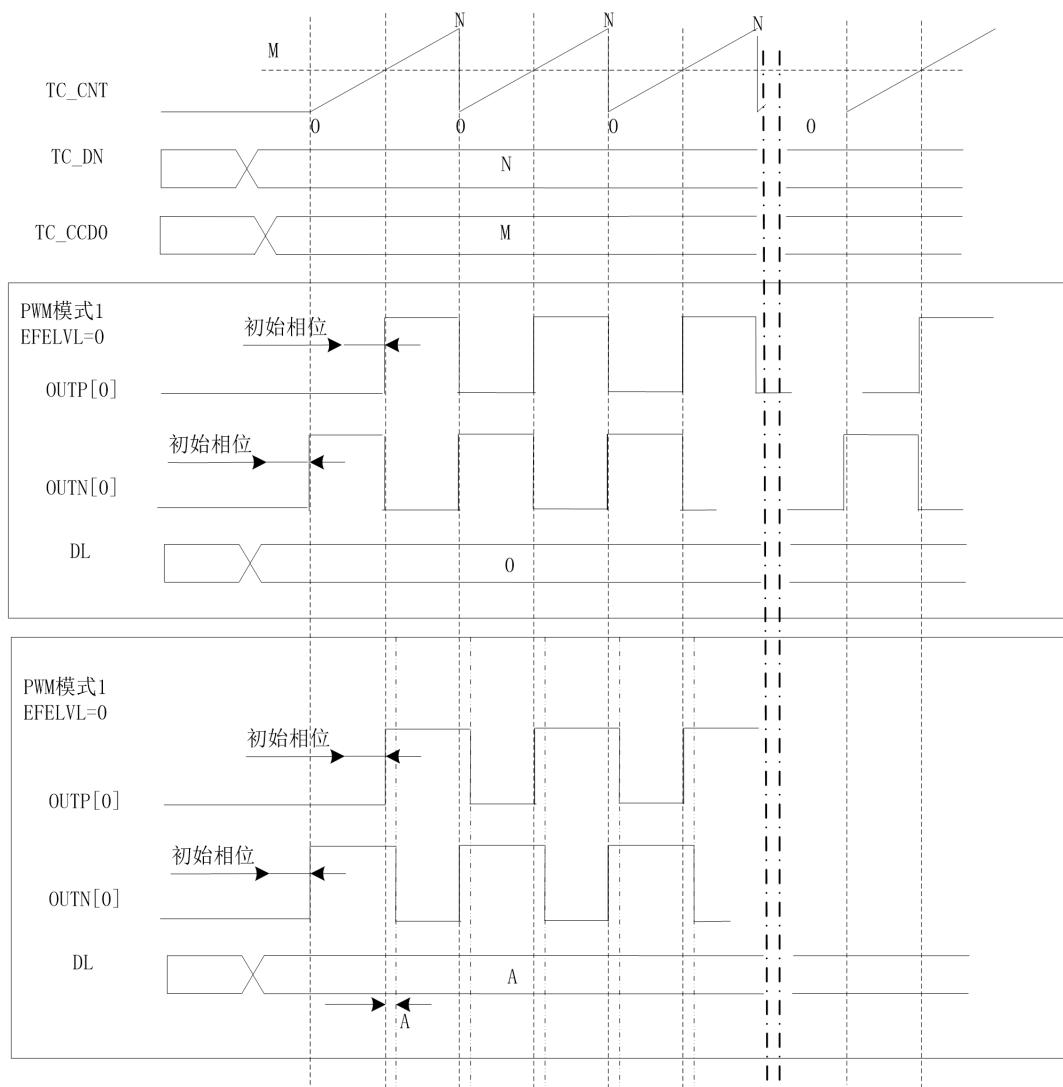
8.4.4 比较输出模式，PWM 输出功能

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。

8.4.5 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。

8.5 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;

配置目标标记数值寄存器 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。(3.6864MHz/(255+1) / (13+1) = 1MS)。

- 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。

9 模拟外设

9.1 特点

由 10bit 升级为 12bit SAR ADC;

- 对 VBAT 引脚和 AIN6 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。
- 9 通道模拟多路选择器，其中 AIN0 – AIN6 通道用于外部模拟信号测量，VBAT 用于电池电压测量，Temp Sensor 用于测量温度传感器的输出；温度传感器的优先级最高。
- AIN0~AIN5 的输入阻抗约为 3M 欧姆；
- AIN2 和 AIN3 输入阻抗受相应管脚上 CMP 内部 600K 影响，当对应 CMP 的内部阻抗有效时，相应管脚选择 AIN2 或 AIN3 功能时也会有 600K 对地阻抗；
- **AIN6 在被选通进行 SAR 测量时，内部有两个 300K 电阻做分压；最大支持 2.5V 输入；**
- 除了 AIN6，其他 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；**
- 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 2ms。

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 2.3V 到 4.9V 共分为 14 档；
- 当选择为外部 PIN 输入时，阈值固定为 1.25V 左右，内阻约为 1M 欧姆
- 功耗约 5uA

两路比较器电路 CMP1 和 CMP2，主要特点如下：

- 外部 PIN 输入，阈值约为 1.28V 左右
- 功耗约 0.3uA，可使用比较器进行主电上下电监测。
- 注意 CMP1 和 CMP2 内部有 600K 的下拉电阻，可通过寄存器 (SYS_PD (0x08)) 选择关闭内部电阻。

9.2 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_CTL	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_CTL	Offset+0x10	比较器控制寄存器
LVD_STAT	Offset+0x14	比较器状态寄存器
SAR_CTL1 (新增)	Offset+0x18	SAR-ADC 增益控制寄存器
SAR_DAT2 (新增)	Offset+0x1C	SAR-ADC 12bit 数据寄存器

ANA_PAD	Offset+0x3c	VBAT 域模拟 PAD 配置寄存器
ANA_RST	Offset+0x40	VBAT 域软件复位寄存器

9.2.1 ADC 控制寄存器 SAR_CTL(0x00)

偏移地址 0x00

比特位	名称	描述	读/写标志	复位值
31:17	Reserved	预留	R	0
16:12	REF_WAIT	从开启 REF 到开启 ADC 需要等待的时间： 等待时间最小刻度的典型值是 122 微秒。 5'd0: 122 微秒 5'd31: (31+1)* 122 微秒 注：建议客户应用程序将该寄存器配置为 0x10，即 2074 微秒。	R/W	0x5
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: 30.5 微秒 5'd31: (31+1)* 30.5 微秒 即等待时间=(SAR_WAIT+1)*30.5 微秒 注：启动 ADC 测量的步骤： 开启 REF，等待 REF_WAIT 时间，建议使用默认值； 开启 ADC 和温度传感器，等待 SAR_WAIT 时间； 16 个 SAR 采样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 1：由于内部时基存在偏差，如果软件设置超时等待，超时等待时间，建议留出 3-5 倍的余量。 注 2：建议客户应用程序将该寄存器配置为默认值 0xE，即 457.5 微秒。	R/W	0xE
6	SAR_CH3	SAR-ADC Channel 选择，与 SAR_CH[2:0]组成一个 4bit 寄存器共同定义。具体定义参考 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	R/W	0
2:0	SAR_CH	SAR-ADC Channel 选择，与 bit[6] SAR_CH3 组合为 4bit，共同定义 SAR-ADC 通道的选择，配置 {SAR_CH3, SAR_CH[2:0]} 定义如下： 0000: Temp Sensor 0001: VBAT(对于 3.6V 电池，1/2 分压后得到 1.8V， PGA 采用 0.5 倍，测量输入 0.9V)	R/W	0

		<p>0010: AIN0 0011: AIN1 0100: AIN2 0101: AIN3 0110: AIN4 0111: AIN5 1000: AIN6 (内部有两个 300K 电压进行分压) =其他: 保留 Note: 不管设置为哪个通道, 自动温度测量的优先级最高。</p>		
备注: 以上寄存器位只有当 SAR_START 中的 ST=0 时才可写。				

9.2.2 SAR-ADC 启动寄存器 SAR_START(0x04)

偏移地址 0x04

比特位	名称	描述	读/写标志	复位值
31:01	Reserved	预留	R	0
0	ST	<p>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) + 3ms。</p>	R/W	0

9.2.3 SAR-ADC 状态寄存器 SAR_STATUS(0x08)

偏移地址 0x08

比特位	名称	描述	读/写标志	复位值
31:02	Reserved	预留	R	0
1	TPS_BUSY	<p>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。</p>	R	0
0	DREADY	<p>ADC Date Ready Pending Bit 0: ADC 转换结果未完成 1: ADC 转换结果已完成 Note: 1、写 1 清零; 该位置 1 后, 要求至少等待 30uS 后发起清零操作; 2、由 RTC 控制的自动温度测量不在该状态中指示;</p>	R/W	0

9.2.4 ADC 数据寄存器 SAR_DAT(0x0c)

偏移地址 0x0C

比特位	名称	描述	读/写标志	复位值
15:10	Reserved	预留	R	0
9:0	SAR-DAT	ADC 转换结果	R	0

9.2.5 LVD 控制寄存器 LVD_CTL(0x10)

偏移地址 0x10

比特位	名称	描述	读/写标志	复位值
31:8	Reserved	预留	R	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	Reserved	预留	R	0
3:0	LVDS	LVD 阈值电压设置: 0000 2.7 0001 2.7 0010 2.7 0011 2.9 0100 3.1 0101 3.3 0110 3.5 0111 3.7 1000 3.9 1001 4.1 1010 4.3 1011 4.5 1100 4.7 1101 4.9 1110 检测外部管脚 LVDIN0 的电压, 与 LBGR (1.25V) 进行比较; 1111 保留;	R/W	0

备注: LVD、比较器 1 和比较器 2 合并一个中断向量; SAR 单独一个中断向量。

9.2.6 LVD 状态寄存器 LVD_STAT(0x14)

偏移地址 0x14

比特位	名称	描述	读/写标志	复位值
31:7	Reserved	预留	R	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;	R/W	0

		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；		
3	Reserved	预留	R	0
2	CMP2IF	比较器2状态标志 =0: 低于阈值； =1: 高于阈值；只读	R	0
1	CMP1IF	比较器1状态标志 =0: 低于阈值； =1: 高于阈值；只读	R	0
0	LVDIF	LVD状态标志 =0: 低于阈值； =1: 高于阈值；只读	R	0

9.2.7 ADC 控制寄存器 1 SAR_CTL1(0x18) (新增)

SAR-ADC 增益控制寄存器

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, 不能操作	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

9.2.8 ADC 数据寄存器 2 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

9.2.9 AIN 功能配置寄存器 ANA_PAD

ANA PAD 寄存器

Offset = 0x3c

比特位	名称	描述	读/写标志	复位值
31:3	---	预留	R	0
1	P45_AEN	SAR ADC 测量输入开关 0: 关; 1: 开;	R/W	1'b0
0	P44_AEN	SAR ADC 测量输入开关 0: 关; 1: 开;	R/W	1'b0

注意：如果要使用 P44/P45 口的 AIN 功能（SAR ADC 测量输入口），必须配置 ANA_PAD 寄存器（0x3C）。

9.2.10 局部复位寄存器 ANA_RST

ANA RST 寄存器

Offset = 0x40

比特位	名称	描述	读/写标志	复位值
31:16	---	预留	R	0
15:8	ps	[7:0]的写密码保护位 =0xa5, [7:0]可以写 =其他值, [7:0]无法写	R/W	0
7:0	ana_rst	强制复位 VBAT 电源域所有带复位端的寄存器, 不含万年历寄存器 =0xa8, 复位 VBAT 电源域所有带复位端的寄存器 =其他, 不起作用	R/W	0

注：若需对 VBAT 电源域的寄存器进行复位应按如下步骤进行：

- 1、开 ANA RST 寄存器的密码保护位：ANA RST = 0xa500;
- 2、使 VBAT 电源域处于复位态：ANA RST = 0xa5a8;
- 3、延时 300us;
- 4、释放 VBAT 电源域复位态：ANA RST = 0xa500;
- 5、关 ANA RST 寄存器的密码保护位：ANA RST = 0x0000;

9.3 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、检测 SAR_START 的 bit0 的状态，若为 1 则等待；检测到 0 则延时 100us 后进入下一步。
- 3、判断 SAR-ADC 状态寄存器 SAR_STATUS 的 bit1 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。
- 4、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。
- 5、SAR-ADC 启动寄存器 SAR_START 配置为 1，启动 ADC 转换。
- 6、判断 SAR-ADC 状态寄存器 SAR_STATUS 的 bit0 位，等待转换完成。
- 7、读取 ADC 转换数据寄存器 SAR_DAT。
- 8、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。
- 9、转换及计算完成。

9.4 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、检测 SAR_START 的 bit0 的状态，若为 1 则等待；检测到 0 则延时 100us 后进入下一步。
- 3、判断 SAR-ADC 状态寄存器 SAR_STATUS 的 bit1 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。
- 4、配置 ADC 控制寄存器 SAR_CTL 通道选择为 VBAT，配置等待时间；
 - 5、SAR-ADC 启动寄存器 SAR_START 配置为 1，启动 ADC 转换。
 - 6、判断 SAR-ADC 状态寄存器 SAR_STATUS 的 bit0 位，等待转换完成。
 - 7、读取 ADC 转换数据寄存器 SAR_DAT。
 - 8、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。
 - 9、转换及计算完成。

9.5 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，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。

10 GPIO

10.1 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、8 个 P10 口、8 个 P11 口
- **PD 口包含 8 个 P12 口、5 个 P13 口、3 个 P14 口**
- GPIO 是 AHB 的外设
- 支持 bitband 操作；
- 注意：RN8613/8615/8318 的 SEG/IO 复用引脚做为 IO 口使用时为开漏结构。
- **P141 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。**

10.2 寄存器描述

GPIO 寄存器地址：

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

GPIO 寄存器的偏移地址

寄存器名	地址偏移量	描述
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
PIEA	0x18H	PA 口输入使能选择
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
PIEB	0x30H	PB 口输入使能选择
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 PCC PC 口复用寄存器
PIEC	0x44H	PC 口输入使能选择
PIMC	0x48H	PC 口输入模式寄存器
PCB2	0X4CH	PB 口复用寄存器 2
PMD	0x50H	PD 口模式寄存器 (输入或者输出)
PD	0x54H	PD 口数据寄存器
PCD	0x58H	PD 口复用寄存器
PUD	0x5CH	PUD 口上拉选择寄存器
PCE	0x60H	SEGCOM 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	0X7CH	PD 口数据置位寄存器, 写 1 到该寄存器, PD 口对应位会被写 1;
PDCLR	0X80H	PD 口数据清零寄存器, 写 1 到该寄存器, PD 口对应位会被清零;
PIED	0x84H	PD 口输入使能选择

PIMD	0x88H	PD 口输入模式寄存器
------	-------	-------------

注: IO 口类型见第 1.4 章管脚排列。

建议使用 bitband 功能 (见 4.3.2 章节) 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 (0x64H~0x78H) 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, 上拉选择、输入模式选择在所有复用配置下均有效。
上拉使能除 P141 外仅支持输入模式; P141 引脚输入/输出模式上拉配置均有效。

10.2.1 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

10.2.2 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

10.2.3 PA 口复用 0 寄存器 PCA0 (0x08) (修改)

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	定义 UART2 的复用配置: =00: 选择 P24/P25 为 UART2 =01: 选择 P00/P01 为 UART2 =10: 选择 P22/P23 为 UART2 =11: 选择 P26/P27 为 UART2	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; P24 复用为 SWDCLK P25 复用为 SWDIO	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。 P26 复用为 RX3 P27 复用为 TX3	R/W	00
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。 P24 复用为 RX2 P25 复用为 TX2	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。 P22 复用为 RX1 P23 复用为 TX1	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。 P20 复用为 RX0 P21 复用为 TX0	R/W	00
24: 9	PC17[1:0] PC16[1:0] PC15[1:0] PC14[1:0] PC13[1:0] PC12[1:0] PC11[1:0] PC10[1:0]	分别定义端口 P10~P17 复用配置: =00: 选择为 IO 口 =01: 选择为 KEY 输入口 =10: 选择为 TC 输出 =11: 选择为 TC 输入, 对 TC0/TC1 可同时有效 P17~P10 对应的 KEY 输入、TC 输出分别为: P17 对应 KEY7, TC1_P[1] P16 对应 KEY6, TC1_N[1] P15 对应 KEY5, TC1_P[0] P14 对应 KEY4, TC1_N[0] P13 对应 KEY3, TC0_P[1] P12 对应 KEY2, TC0_N[1] P11 对应 KEY1, TC0_P[0] P10 对应 KEY0, TC0_N[0]	R/W	00

8	KEY4_SEL	与 PCA0 寄存器的 bit[6]共同定义端口 P04 的复用配置, 具体配置和定义见 PCA0 寄存器的 bit[6]位描述;	R/W	0
7	PC05	定义端口 P02~P05 的复用配置: =0: 选择为 IO 口; =1: 选择为模拟输入口	R/W	0
6	PC04	由{PCA0[8],PCA0[6]}定义端口 P04 的复用配置: =00: 选择为 IO 口; =01: 选择为 AIN3/LVDIN 模拟输入口; =1x: P04 选择为 KEY4 (PC14 选择为 KEY4 优先级高);	R/W	0
5	PC03	定义端口 P02~P05 的复用配置: =0: 选择为 IO 口; =1: 选择为 AIN3/CMP2 模拟输入口	R/W	0
4	PC02	定义端口 P02~P05 的复用配置: =0: 选择为 IO 口; =1: 选择为 AIN2/CMP1 模拟输入口	R/W	0
3:2	PC01	定义端口 P01 复用配置: =00: 选择为 IO 口; =01: 选择为 AIN1 模拟输入口; =10: 选择为 KEY3; (PC13 选择为 KEY3 优先级高) =11: 选择为 TX2;	R/W	0
1:0	PC00	定义端口 P00 复用配置: =00: 选择为 IO 口; =01: 选择为 AIN0 模拟输入口; =10: 选择为 KEY2 (PC12 选择为 KEY2 优先级高); =11: 选择为 RX2;	R/W	0

10.2.4 PA 口复用 1 寄存器 PCA1 (0x0C) (修改)

比特位	名称	描述	读/写标志	复位值
31:18	---	预留	R	0
17	PC267_2	定义端口 P26 和 P27 复用配置 2: =0, P26 和 P27 的功能由 PC267 寄存器位 (PCA0.bit28) 决定 =1, P26 和 P27 选择为 UART2 P26 复用为 RX2 P27 复用为 TX2	R/W	0
16	PC223_2	定义端口 P22 和 P23 复用配置 2: =0, P22 和 P23 的功能由 PC223 寄存器位 (PCA0.bit26) 决定 =1, P22 和 P23 选择为 UART2 P22 复用为 RX2 P23 复用为 TX2	R/W	0
15:14	PC37[1:0]	定义端口 P37 复用配置: =00: 选择为 IO 口; =01: 选择为外部中断输入口 INT7; =1x: 选择为晶体 POSCI 管脚	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 POSC		
13:12	PC36[1:0]	定义端口 P36 复用配置： =00: 选择为 IO 口； =01: 选择为外部中断输入口 INT6； =1x: 选择为晶体 POSCO 管脚 备注：只要 PC36[1]和 PC37[1]中任何一位为高，那么就选择为 POSC	R/W	0
11:10	PC35[1:0]	定义端口 P35 复用配置： =00: 选择为 IO 口； =01: 选择为外部中断输入口 INT5； =10: 选择为 TC 输入 =11: 选择为电能积分模块 D2F 的输出 D2F_OUT2	R/W	0
9:8	PC34[1:0]	定义端口 P34 复用配置： =00: 选择为 IO 口； =01: 选择为外部中断输入口 INT4； =10: 选择 IOCNT0_OUT0 =11: 选择为电能积分模块 D2F 的输出 D2F_OUT1 IOCNT0_OUT0 是 INT4 输入经 IOCNT0 转发后输出，可配置为反向输出，或者分频输出，具体配置参考脉冲转发功能。此功能用于从计量芯片输入脉冲后直接转发输出做电能精度校正用。	R/W	0
7:6	PC33[1:0]	定义端口 P33 复用配置： =00: 选择为 IO 口； =01: 选择为外部中断输入口 INT3； =10: 选择为 TC 输入 =11: 选择为电能积分模块 D2F 的输出 D2F_OUT0	R/W	0
5:4	PC32[1:0]	定义端口 P32 复用配置： =00: 选择为 IO 口； =01: 选择为外部中断输入口 INT2； =10: 选择为 RTC 输出 RTC_OUT (默认选择为 RTC 输出) =11: 选择 KEY5 该管脚位于 VCC 域，其 RTCOUT 输出频率的补偿值，在温度补偿时由 VBAT 域的 RTC 模块计算后更新到 VCC 域；应用时注意 VCC 域掉电后重新上电，需及时启动温度补偿，以加载 RTCOUT 的补偿值，防止 RTCOUT 输出频率出现偏差。	R/W	10
3:2	PC31[1:0]	定义端口 P31 复用配置： =00: 选择为 IO 口； =01: 选择为外部中断输入口 INT1； =10: 选择为 TC 输入 =11: 选择为 RX4	R/W	0
1:0	PC30[1:0]	定义端口 P30 复用配置： =00: 选择为 IO 口； =01: 选择为外部中断输入口 INT0； =10: 选择为 TC 输入 =11: 选择为 TX4	R/W	0

		=01: 选择为外部中断输入口 INT0; =10: 选择为 TC 输入 =11: 选择为 TX4		
--	--	---	--	--

10.2.5 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；	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

10.2.6 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, Vil=0.3VCC Vih=0.7VCC； =1: TTL 缓冲器, Vil=0.16VCC Vih=0.4VCC；	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, Vil=0.3VCC Vih=0.7VCC； =1: TTL 缓冲器, Vil=0.16VCC Vih=0.4VCC；	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

10.2.7 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：	R/W	FF

		=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。		
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

10.2.8 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。

10.2.9 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。 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	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

10.2.10 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 LCD。	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 LCD。	R/W	00
15:14	PC57~PC56	PC57~P56 端口复用配置： =0: 选择为 IO 口 =1: 保留 注意: P56 可作为 RTCOUT 输出, 其复用关系由 RTC 章节寄存器 VBAT_IOMODE 寄存器的配置决定, 且优先级高于 GPIO 复用关系的配置。 当 VCC 无电, VBAT 初次上电时, P56 默认输出 1HZ。一旦 VCC 上电, P56 输出重新初始化, 1HZ 输出会被关闭, 外围电路设计需要注意该问题。 P56 为 VBAT 域管脚作为输入时, 输入电压不能高于芯片 VBAT 管脚供电电压; 输出时, 其输出电平压为芯片 VBAT 管脚电压。	R/W	0
13:12	PC55~PC54	P55~P54 端口复用配置： =0: 选择为 IO 口 =1: 选择为 UART5 P54 复用为 RX5 P55 复用为 TX5	R/W	0
11:10	PC53~PC52	P53~P52 端口复用配置： =0: 选择为 IO 口 =1: 选择为 I2C P52 复用为 SCL P53 复用为 SDA	R/W	0
9	PC51	P51 端口复用配置： =0: 选择为 IO 口 =1: 选择为 IOCNT2_OUT0 输出 IOCNT2_OUT0 是 INT5 输入经 IOCNT2 转发后输出, 可配置为反向输出, 或者分频输出, 具体配置参考脉冲转发功能。 此功能用于从计量芯片输入脉冲后直接转发输出做电能精度校正用。	R/W	0
8	PC50	P50 端口复用配置： =0: 选择为 IO 口 =1: 选择为 IOCNT1_OUT0 输出 IOCNT1_OUT0 是 INT2 输入经 IOCNT1 转发后输出, 可配置为反向输出, 或者分频输出, 具体配置参考脉冲转发功能。 此功能用于从计量芯片输入脉冲后直接转发输出做电能精度校正用。	R/W	0

7:6	PC47~PC46	P47~P46 端口复用配置: =0: 选择为 IO 口 =1: 选择为 SPI0 P46 复用为 SPI0_MISO P47 复用为 SPI0_MOSI	R/W	0
5:4	PC45~PC44	P45~P44 端口复用配置: =0: 选择为 IO 口 =1: 选择为 SPI0 (因电源域区别, 此复用不建议使用, 可选 P51/P50 复用为此功能) P44 复用为 SPI0_SCSN P45 复用为 SPI0_SCLK P44/P45 可复用为 AIN5/AIN6, 由模拟外设章节 ANA_PAD 的配置决定, 且优先级最高。 P44/P45 为 VBAT 域管脚作为输入时, 输入电压不能高于芯片 VBAT 管脚供电电压; 输出时, 其输出电平压为芯片 VBAT 管脚电压。	R/W	0
3:0	PC43~PC40	P43~P40 端口复用配置: =0: 选择为 IO 口 =1: 选择为 7816 P40 复用为 7816_CLK P41 复用为 78160_IO P42 复用为 78161_IO P43 复用为 78161_I	R/W	0

10.2.11 PB 口复用寄存器 2 PCB2 (0x4CH) (修改)

比特位	名称	描述	读/写标志	复位值
31:30	PC57_2	P57 端口复用配置 2: =00: P57 口功能由 PC57 寄存器位 (0x24) 决定; =01: P57 选择为 TCIN。 =10: P57 选择为 TC1_P[1]; =11: 保留	R/W	00
29:28	PC56_2	P56 端口复用配置 2: =00: P56 口功能由 PC56 寄存器位 (0x24) 决定; =01: P56 选择为 TCIN。 =10: P56 选择为 TC1_N[1]; =11: 保留 注意: P56 可作为 RTCOUT 输出, 其复用关系由 RTC 章节寄存器 VBAT_IOMODE 寄存器的配置决定, 且优先级高于 GPIO 复用关系的配置。 当 VCC 无电, VBAT 初次上电时, P56 默认输出 1HZ。一旦 VCC 上电, P56 输出重新初始化, 1HZ 输出会被关闭, 外围电路设计需要注意该问题。 P56 为 VBAT 域管脚作为输入时, 输入电压不能高于芯片 VBAT 管脚供电电压; 输出时, 其输出电平压为芯片 VBAT	R/W	00

		管脚电压。		
27:26	PC55_2	P55 端口复用配置 2: =00: P55 口功能由 PC55 寄存器位 (0x24) 决定; =01: P55 选择为 TCIN。 =10: P55 选择为 TC1_P[0]; =11: P55 选择为 SPI1_MOSI	R/W	00
25:24	PC54_2	P54 端口复用配置 2: =00: P54 口功能由 PC54 寄存器位 (0x24) 决定; =01: P54 选择为 TCIN。 =10: P54 选择为 TC1_N[0]; =11: P54 选择为 SPI1_MISO	R/W	00
23:22	PC53_2	P53 端口复用配置 2: =00: P53 口功能由 PC53 寄存器位 (0x24) 决定; =01: P53 选择为 TCIN。 =10: P53 选择为 TC0_P[1]; =11: P53 选择为 SPI1_SCLK	R/W	00
21:20	PC52_2	P52 端口复用配置 2: =00: P52 口功能由 PC52 寄存器位 (0x24) 决定; =01: P52 选择为 TCIN。 =10: P52 选择为 TC0_N[1]; =11: P52 选择为 SPI1_SCSN 注 1: P40~P43, P52~P55 不能同时复用为 SPI1, 应用需要注意。 注 2: P52 仅在 SPI1 从模式下有效, 在 SPI1 主模式下不要配置为 1。	R/W	00
19:18	PC51_2	P51 端口复用配置 2: =00: P51 口功能由 PC51 寄存器位 (0x24) 决定; =01: P51 选择为 RTC_OUT。 =10: P51 选择为 SPI0_SCLK; =11: IOCNT2_OUT1; (INT7 引脚输出) 注 1: IOCNT2_OUT1 是 INT7 输入经 IOCNT2 转发后输出, 可配置为反向输出, 或者分频输出, 具体配置参考脉冲转发功能。此功能用于从计量芯片输入脉冲后直接转发输出做电能精度校正用。 注 2: 该管脚位于 VCC 域, 其 RTCOUT 输出频率的补偿值, 在温度补偿时由 VBAT 域的 RTC 模块计算后更新到 VCC 域; 应用时注意 VCC 域掉电后重新上电, 需及时启动温度补偿, 以加载 RTCOUT 的补偿值, 防止 RTCOUT 输出频率出现偏差。	R/W	00
17:16	PC50_2	P50 端口复用配置 2: =00: P50 口功能由 PC50 寄存器位 (0x24) 决定; =01: P50 选择为 RTC_OUT。 =10: P50 选择为 SPI0_SCSN; =11: IOCNT1_OUT1 (INT6 引脚输出)	R/W	00

		<p>注 1: P50 仅在 SPI0 从模式下有效, 在 SPI0 主模式下不要配置为 1。P50/P51/P46/P47 与 P110~P113 不能同时复用为 SPI0, 并由 PCC (0x3C) 寄存器 bit28 决定哪组 SPI0 生效, 应用需要注意。</p> <p>注 2: IOCNT1_OUT1 是 INT6 输入经 IONCNT1 转发后输出, 可配置为反向输出, 或者分频输出, 具体配置参考脉冲转发功能。此功能用于从计量芯片输入脉冲后直接转发输出做电能精度校正用。</p> <p>注 3: 该管脚位于 VCC 域, 其 RTCOUT 输出频率的补偿值, 在温度补偿时由 VBAT 域的 RTC 模块计算后更新到 VCC 域; 应用时注意 VCC 域掉电后重新上电, 需及时启动温度补偿, 以加载 RTCOUT 的补偿值, 防止 RTCOUT 输出频率出现偏差。</p>		
15:14	PC47_2	<p>P47 端口复用配置 2:</p> <p>=00: P47 口功能由 PC47 寄存器位 (0x24) 决定;</p> <p>=01: P47 选择为 TX4。</p> <p>=10: 选择为电能积分模块 D2F 的输出 D2F_OUT0</p> <p>=11: 选择为电能积分模块 D2F 的输出 D2F_OUT1</p>	R/W	00
13:12	PC46_2	<p>P46 端口复用配置 2:</p> <p>=00: P46 口功能由 PC46 寄存器位 (0x24) 决定;</p> <p>=01: P46 选择为 RX4。</p> <p>=10: 选择为电能积分模块 D2F 的输出 D2F_OUT1</p> <p>=11: 选择为电能积分模块 D2F 的输出 D2F_OUT2</p>	R/W	00
11:10	PC45_2	<p>P45 端口复用配置 2:</p> <p>=00: P45 口功能由 PC45 寄存器位 (0x24) 决定;</p> <p>=01: P45 选择为 KEY7 (PC17 优先级高)。</p> <p>=1x: 保留</p> <p>P45 可复用为 AIN6, 由模拟外设章节 ANA_PAD 的配置决定, 且优先级最高。</p> <p>P45 为 VBAT 域管脚作为输入时, 输入电压不能高于芯片 VBAT 管脚供电电压; 输出时, 其输出电平压为芯片 VBAT 管脚电压。</p>	R/W	00
9:8	PC44_2	<p>P44 端口复用配置 2:</p> <p>=00: P44 口功能由 PC44 寄存器位 (0x24) 决定;</p> <p>=01: P44 选择为 KEY6 (PC16 优先级高)。</p> <p>=1x: 保留</p> <p>P44 可复用为 AIN5, 由模拟外设章节 ANA_PAD 的配置决定, 且优先级最高。</p> <p>P44 为 VBAT 域管脚作为输入时, 输入电压不能高于芯片 VBAT 管脚供电电压; 输出时, 其输出电平压为芯片 VBAT 管脚电压。</p>	R/W	00
7:6	PC43_2	<p>P43 端口复用配置 2:</p> <p>=00: P43 口功能由 PC43 寄存器位 (0x24) 决定;</p> <p>=01: P43 选择为 INT5。 (PC35 优先级高)</p>	R/W	00

		=10: P43 选择为 SPI1_MOSI =11: P43 选择为 SPI2_MOSI		
5:4	PC42_2	P42 端口复用配置 2: =00: P42 口功能由 PC42 寄存器位 (0x24) 决定; =01: P42 选择为 INT4。 (PC34 优先级高) =10: P42 选择为 SPI1_MISO =11: P42 选择为 SPI2_MISO	R/W	00
3:2	PC41_2	P41 端口复用配置 2: =00: P41 口功能由 PC41 寄存器位 (0x24) 决定; =01: P41 选择为 INT3。 (PC33 优先级高) =10: P41 选择为 SPI1_SCLK =11: P41 选择为 SPI2_SCLK	R/W	00
1:0	PC40_2	P40 端口复用配置 2: =00: P40 口功能由 PC40 寄存器位 (0x24) 决定; =01: P40 选择为 INT1。 (PC31 优先级高) =10: P40 选择为 SPI1_SCSN =11: P40 选择为 SPI2_SCSN 注 1: P40~P43, P52~P55 不能同时复用为 SPI1, 应用需要注意。 注 2: P40 仅在 SPI1 从模式下有效, 在 SPI1 主模式下不要配置为 1。	R/W	00

10.2.12 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

10.2.13 PB 口输入模式寄存器 PIMB (0x2C)

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, Vil=0.3VCC Vih=0.7VCC; =1: TTL 缓冲器, Vil=0.16VCC Vih=0.4VCC;	R/W	00

		其中 PIL51 和 PIL50 只读位 0;		
23:16	PIL47~PIL40	定义端口 P40~P47 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
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; 其中 PID51 和 PID50 只读位 0;	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

10.2.14 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

10.2.15 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
31:24	PM117~PM110	=0 输出模式 =1 输入模式	R/W	FF
23:16	PM107~PM100	=0 输出模式 =1 输入模式	R/W	FF
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

10.2.16 PC 口数据寄存器 PC (0x38)

比特位	名称	描述	读 / 写 标志	复位值
31:24	P117~P110	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	P107~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

10.2.17 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 P50/P51/P46/P47 作为 SPI0 接口； =1: 选择 P110~P113 作为 SPI0 接口。	R/W	00
27:24	PC113~PC110	PC113~PC110 定义端口复用配置： =0: 选择为 IO 口； =1: 选择 SPI0 接口。 P110 复用为 SPI0_SCSN P111 复用为 SPI0_SCLK P112 复用为 SPI0_MISO P113 复用为 SPI0_MOSI PC110 仅在 SPI0 从模式下有效，在 SPI0 主模式下不要配置为 1.	R/W	00
23:20	PC107~PC104	PC107~PC104 定义端口复用配置： =0: 选择为 IO 口； =1: 选择为 SPI3。 P104 复用为 SPI3_SCSN P105 复用为 SPI3_SCLK P106 复用为 SPI3_MISO P107 复用为 SPI3_MOSI PC104 仅在 SPI3 从模式下有效，在 SPI3 主模式下不要配置为 1.	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 LCD。	R/W	0
15:8	PC97~PC90	PC97~PC90 定义端口复用配置： =0: 选择为 IO 口； =1: 选择 LCD。	R/W	00
7:0	PC87~PC80	PC87~PC80 定义端口复用配置：	R/W	00

		=0: 选择为 IO 口; =1: 选择 LCD。		
--	--	------------------------------	--	--

10.2.18 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
31:24	PU117~PU110	PU113~PU110 定义端口是否内接下拉: =0: 不接上拉; =1: 内接上拉。	R/W	00
23:16	PU107~PU100	PU107~PU100 定义端口是否内接下拉: =0: 不接下拉; =1: 内接下拉。	R/W	00
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; =1: 内接下拉。	R/W	00

10.2.19 PC 口输入使能寄存器 PIEC (0x44)

比特位	名称	描述	读/写标志	复位值
31:24	PIE117~PIE110	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF
23:16	PIE107~PIE100	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF
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

10.2.20 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
15:12	PIL117~PIL114	定义端口 P117~P114 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	PID117~PID114	定义端口 P117~P114 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

7:4	PIL113~PIL110	定义端口 P113~P110 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
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

10.2.21 PD 口模式寄存器 PMD (0x50)

比特位	名称	描述	读 / 写 标志	复位值
31:24	---	预留	R	0
23:16	PM147~PM140	=0 输出模式 =1 输入模式 P143~P147 暂不支持, 配置无效	R/W	FF
15:8	PM137~PM130	=0 输出模式 =1 输入模式 P133、P136、P137 暂不支持, 配置无效	R/W	FF
7:0	PM127~PM120	=0 输出模式 =1 输入模式	R/W	FF

10.2.22 PD 口数据寄存器 PD (0x54)

比特位	名称	描述	读/写标志	复位值
31:24	---	预留	R	0
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 P143~P147 暂不支持, 配置无效	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 P133、P136、P137 暂不支持, 配置无效	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

10.2.23 PD 口复用寄存器 PCD (0x58)

比特位	名称	描述	读 / 写 标志	复位值
31:22	---	预留	R	00
21:20	PC142	PC142 定义 P142 口复用配置: =00: 选择为 IO 口; =其他: 预留	R/W	0
19:18	PC141	PC141 定义 P141 口复用配置: =00: 选择为 IO 口; =其他: 预留	R/W	0

17:16	PC140	PC140 定义 P140 口复用配置: =00: 选择为 IO 口; =其他: 预留	R/W	0
15:8	PC137~PC130	PC137~PC130 定义端口复用配置: =0: 选择为 IO 口; =1: 选择 LCD 相关引脚或保留 P137~P136 保留 P135 对应 LCDVP2 P134 对应 LCDVP1 P133 保留 P132 对应 LCDVC P131 对应 LCDVB P130 对应 LCDVA	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 LCD P127~P120 对应 COM7~COM0	R/W	0xFF

10.2.24 PD 口上下拉寄存器 PUD (0x5C)

比特位	名称	描述	读 / 写 标志	复位值
31:24	---	预留	R	0
23:18	PU147~PU142	PU147~PU142 定义端口是否内接上拉: =0: 不接上拉; =1: 内接上拉。 P143~P147 暂不支持	R/W	00
17	PU141	PU141 定义端口是否内接上拉: =0: 内接上拉; =1: 不接上拉。 注: 该引脚与其他引脚不同, 输入/输出模式上拉配置均有效。 其他引脚上拉配置, 只在输入模式有效。且该管脚上拉到 3.3V 时, 上拉电阻约 5K, 其他管脚上拉到 3.3V 时, 上拉电阻约 70K。	R/W	0
16	PU140	PU140 定义端口是否内接上拉: =0: 不接上拉; (V3 版 RN8613 为内接上拉) =1: 内接上拉。 (V3 版 RN8613 为不接上拉) 注: V2 版和 V3 版 RN8613 该管脚的上拉定义有区别, 默认情况下, 该位都为 0; 但 V2 版芯片, 该位为 0 时, 表示不接上拉; V3 版芯片, 该位为 0 时, 表示内接上拉。	R/W	0
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;	R/W	00

		=1: 内接下拉。		
--	--	-----------	--	--

10.2.25 PD 口输入使能寄存器 PIED (0x84)

比特位	名称	描述	读 / 写 标志	复位值
31:24	---	预留	R	0
23:16	PIE147~PIE140	输入使能： =1: 不使能输入； =0: 使能输入； P143~P147 暂不支持	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

10.2.26 PD 口输入模式寄存器 PIMD (0x88)

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; P143~P147 暂不支持	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; P144~P147 暂不支持	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; P143 暂不支持	R/W	00
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; P143 暂不支持	R/W	00

10.2.27 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~	SEG3/COM7~SEG0/COM4 定义端口复用配置：	R/W	00

	SEG0/COM4	=0: 选择为 SEG; =1: 选择为 COM。		
--	-----------	------------------------------	--	--

10.2.28 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

10.2.29 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 1: 该端口清零, 输出低电平	R/W	00
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

10.2.30 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

10.2.31 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

10.2.32 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
31:24	P117~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
23:16	P107~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

10.2.33 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
31:24	P117~P110	清零芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
23:16	P107~P100	清零芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
15:8	P97~P90	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出高电平	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, 输出高电平	R/W	00

Note: 读出值无意义

10.2.34
PD 口数据置位寄存器 PDSET (0x7C)

比特位	名称	描述	读/写标志	复位值
31:24	---	保留	R	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 P143~P147 暂不支持	R/W	00
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

10.2.35
PD 口清零置位寄存器 PDCLR (0x80)

比特位	名称	描述	读/写标志	复位值
31:24	保留	-----	R	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 P143~P147 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	R/W	00

		1: 该端口清零, 输出低电平		
--	--	-----------------	--	--

10.3 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 MCU 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。

11 脉冲转发功能 IOCNT

计量脉冲转发功能，支持从 INT4/2/6/5/7 输入的计量脉冲，经 IOCNT 模块后从 IOCNT0_OUT0/IOCNT1_OUT0/IOCNT1_OUT1/IOCNT2_OUT0/IOCNT2_OUT1 直接输出。

11.1 功能特性

- 支持输入脉冲直接转发输出
- 支持输入脉冲电平反向后再转发输出
- 支持输入脉冲 1~2048 分频后再转发输出
- 输入脉冲计数器上升沿数和下降沿计数可配
- 支持输入脉冲计数值读后清零

11.2 复用关系

- IOCNT0_OUT0 (P34) 是 INT4 输入经 IOCNT0 转发后输出
- IOCNT1_OUT0 (P50) 是 INT2 输入经 IOCNT1 转发后输出
- IOCNT1_OUT1 (P50) 是 INT6 输入经 IOCNT1 转发后输出
- IOCNT2_OUT0 (P51) 是 INT5 输入经 IOCNT2 转发后输出
- IOCNT2_OUT1 (P51) 是 INT7 输入经 IOCNT2 转发后输出

11.3 功能描述

IOCNT0/1/2 三个通路独立，其中 IOCNT0/1/2 分别输出到 P34/P50/P51 管脚，可分别配置不同分频比，以及电平反向输出功能，输出脉冲高电平保持 80ms，如果快于 80ms，则保持等 duty 输出。对输入脉冲进行 32 个系统时钟周期滤波。如 11.1 图所示：



图 11.1

11.4 波形分析

11.4.1 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 11.2 等 duty 信号

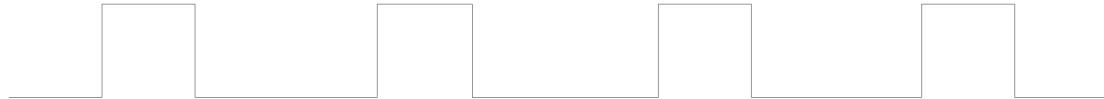


图 11.3 非等 duty 信号

- 脉冲周期非均匀分布



图 11.4 非均匀周期信号

11.4.2 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；

11.4.2.1 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

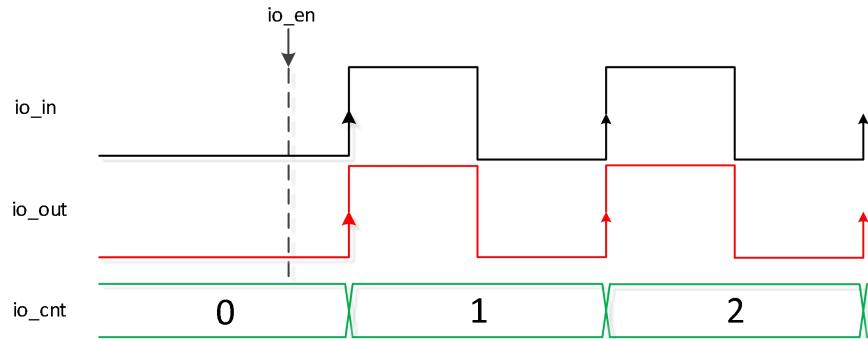


图 11.5 低电平启动上升沿计数（脉冲不分频）

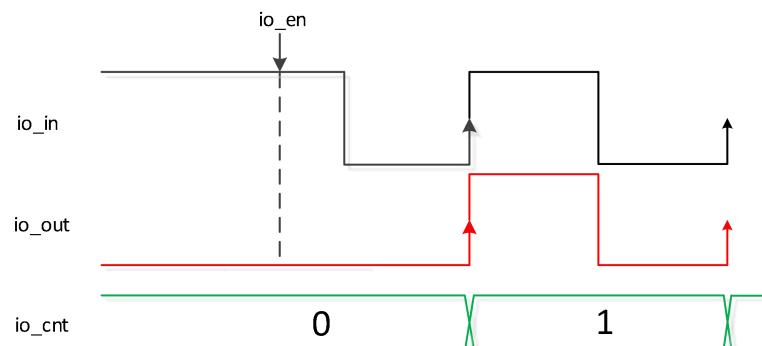


图 11.6 高电平启动上升沿计数（脉冲不分频）

11.4.2.2 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。

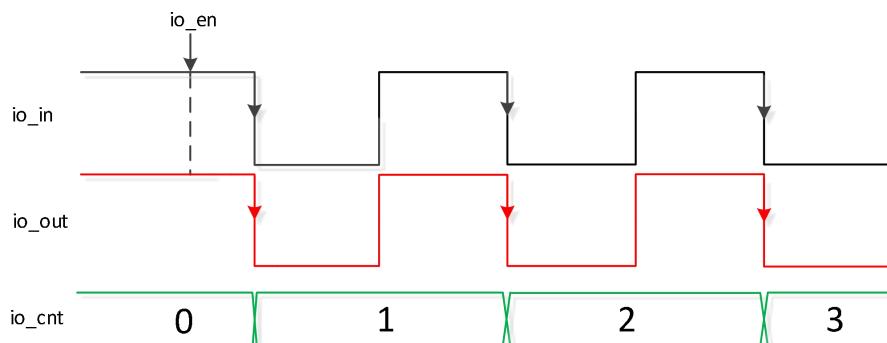


图 11.7 高电平启动下降沿计数（脉冲不分频）

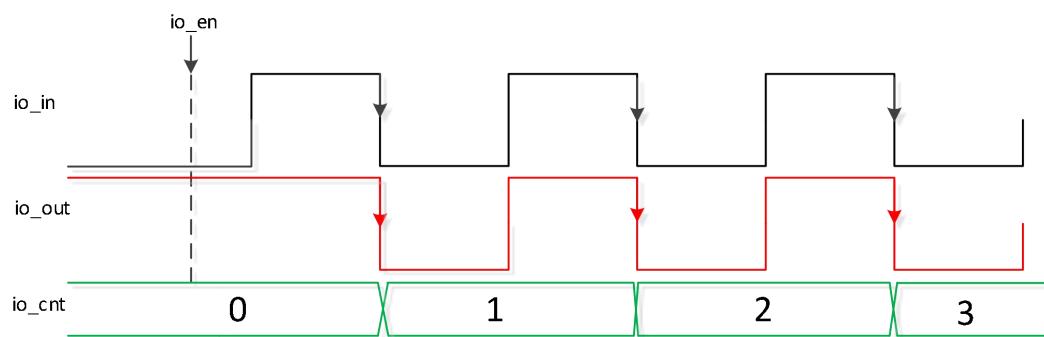


图 11.8 低电平启动下降沿计数 (脉冲不分频)

11.4.2.3 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。

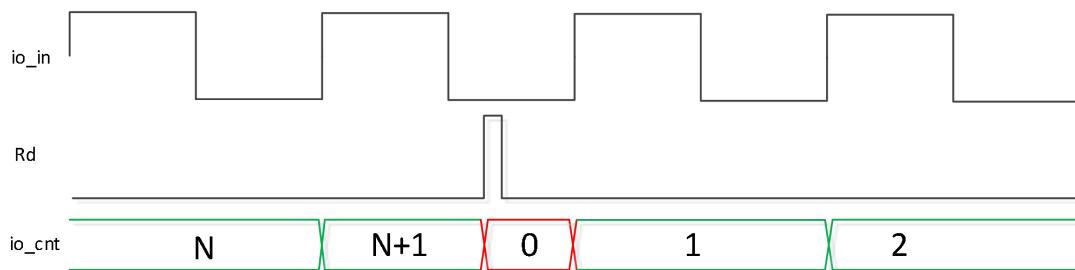


图 11.9 清零操作在 io 边沿前 (上升沿计数)

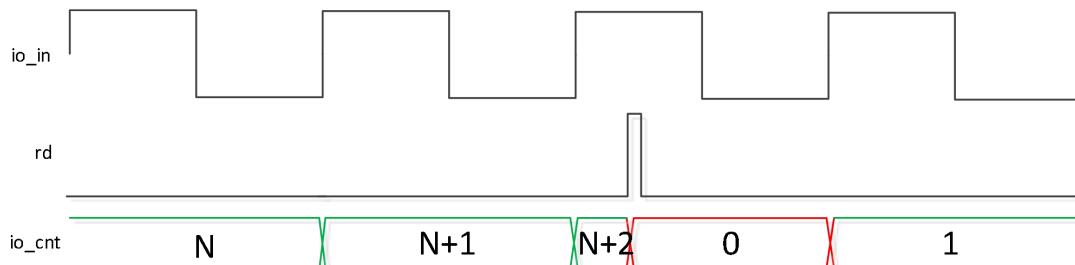


图 11.10 清零操作在 io 边沿后 (上升沿计数)

- 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。

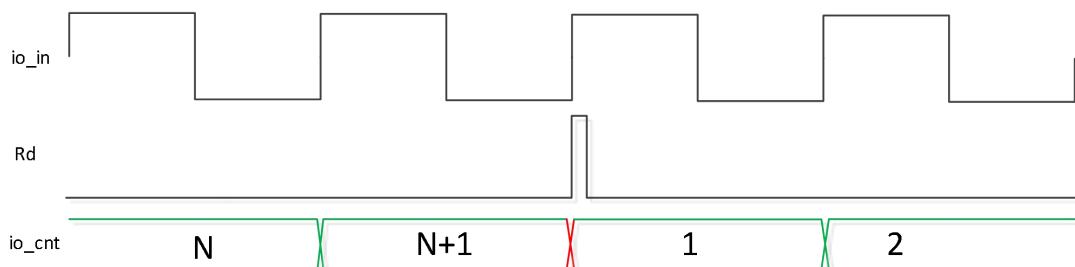


图 11.11 清零操作同 io 边沿 (上升沿计数)

- 计数溢出设计，当 `io_cnt` = 0xffffffff，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。

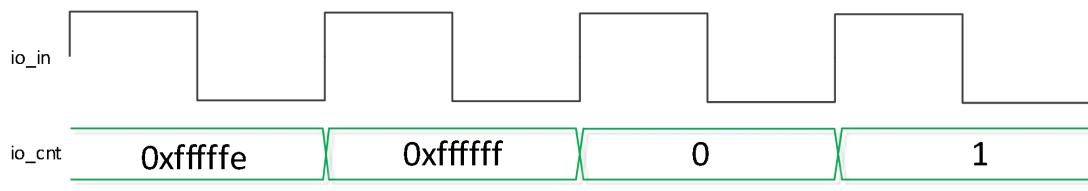


图 11.12 `io_cnt` 溢出 (上升沿计数)

当客户检测到数值溢出时，当前值 a，实际当次计数值 = a + 0xffffffff + 1。

- 溢出与读同时发生，此时读到 0xffffffff，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。

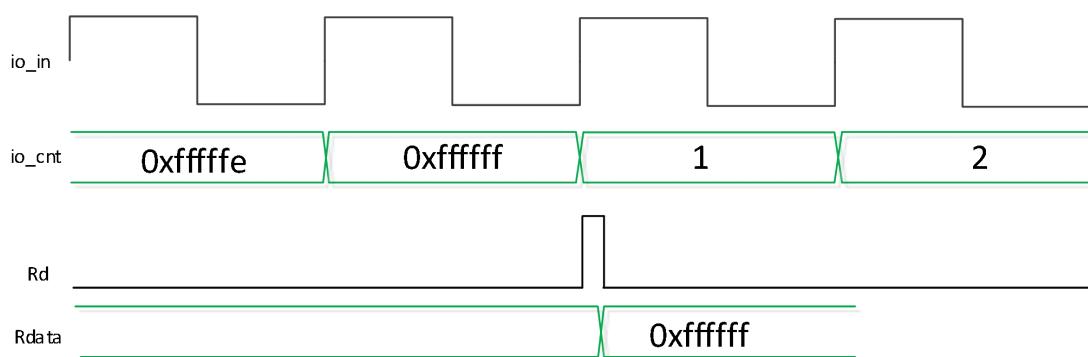


图 11.13 溢出与读同时来 (上升沿计数)

11.4.3 脉冲分频

11.4.3.1 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`。

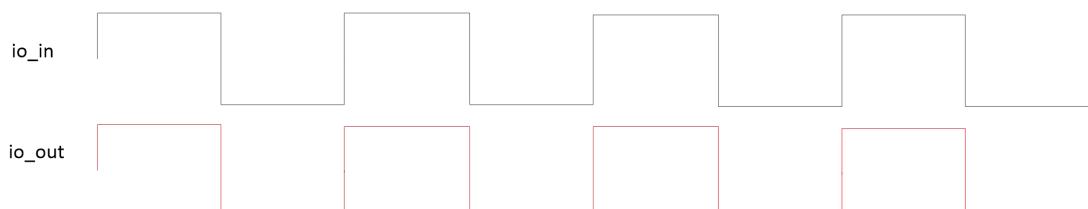


图 11.14 等 duty 信号

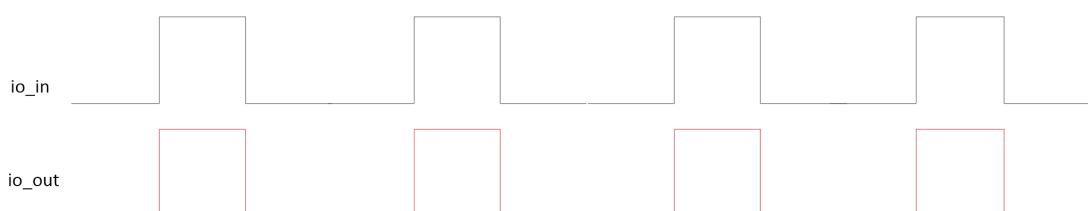


图 11.15 非等 duty 信号

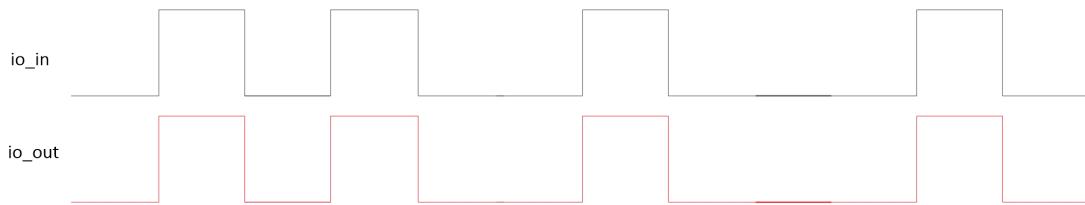


图 11.16 非均匀周期信号

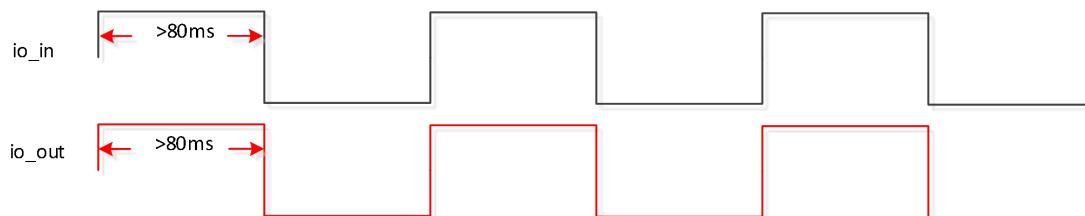


图 11.17 输入信号脉宽大于 80ms

11.4.3.2 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

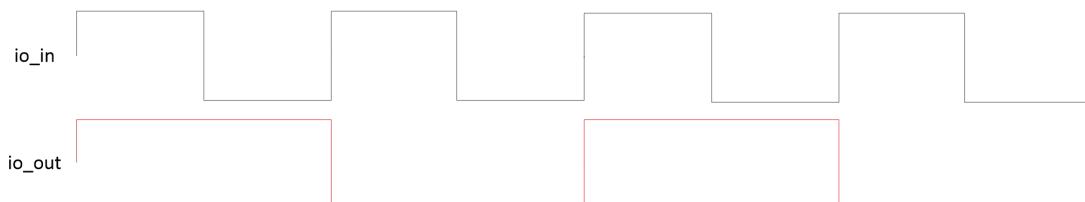
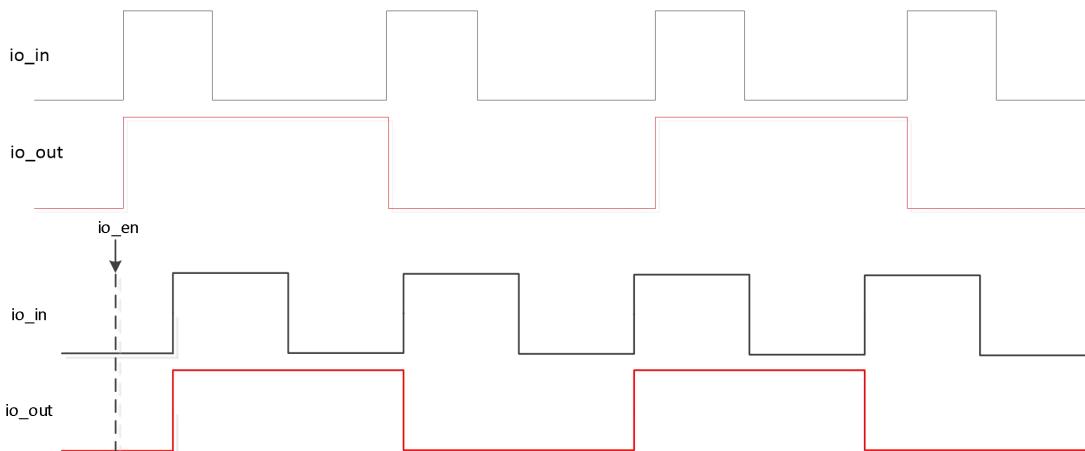
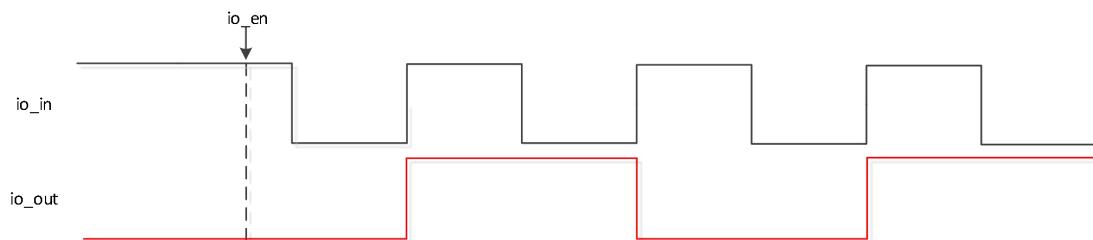
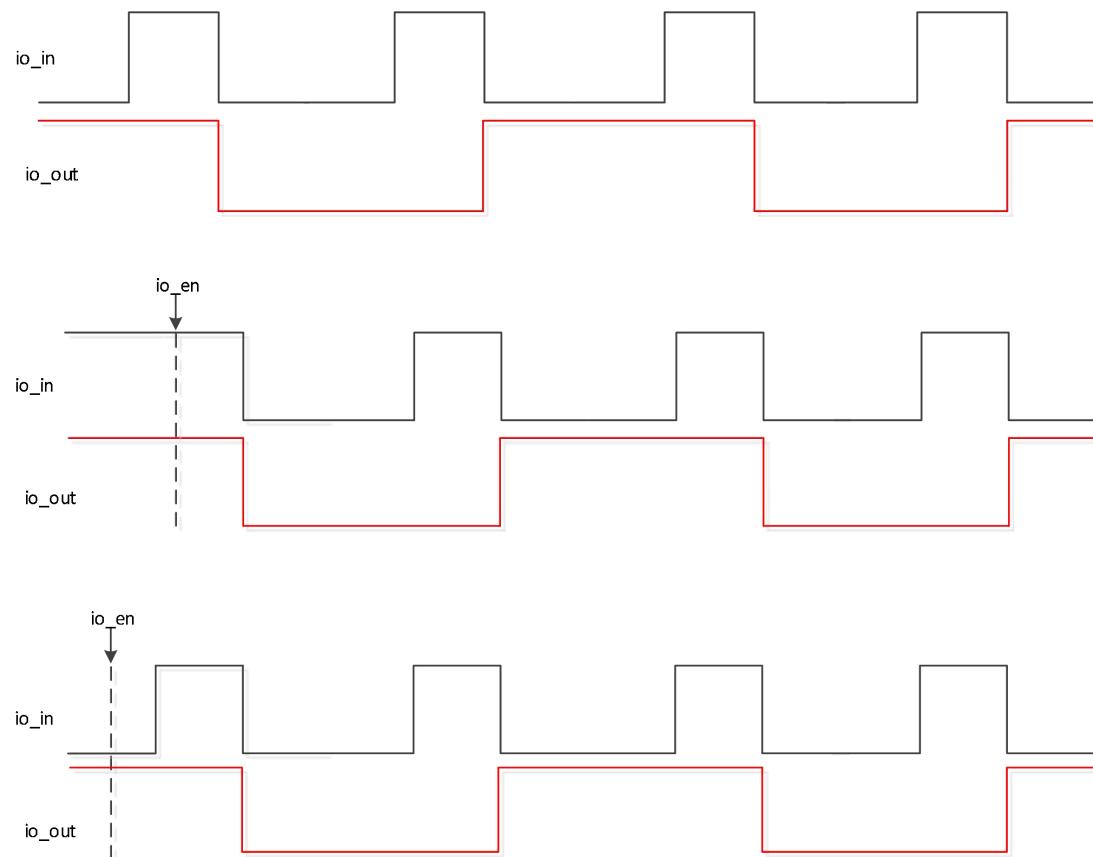


图 11.18 (n+1) 分频

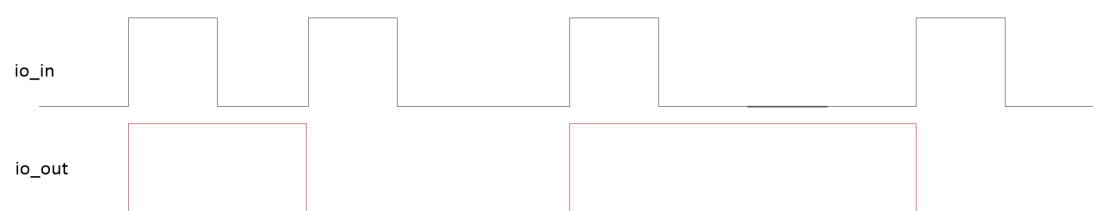




异常点启动，第一个输出脉冲非等 duty。



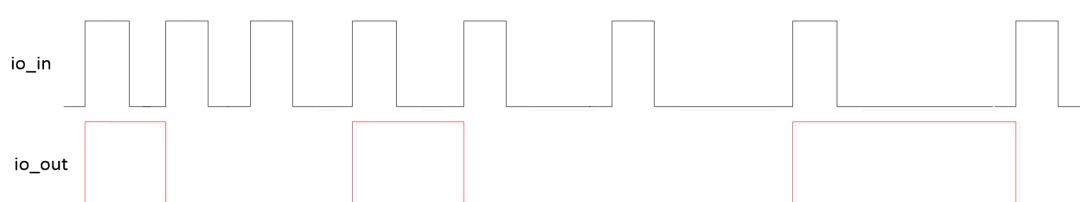
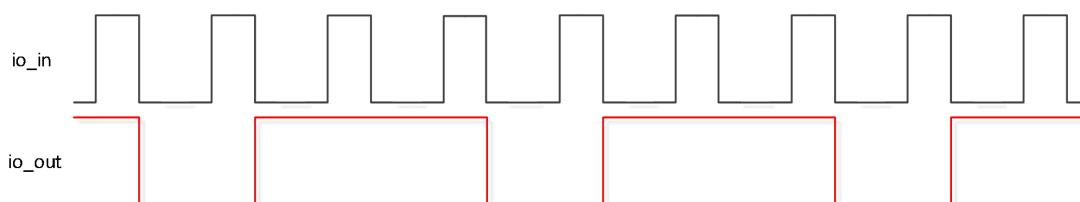
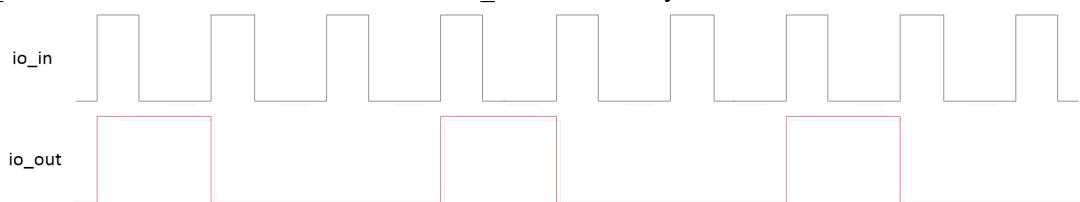
异常点启动，第一个输出脉冲非等 duty。



11.4.3.3 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。

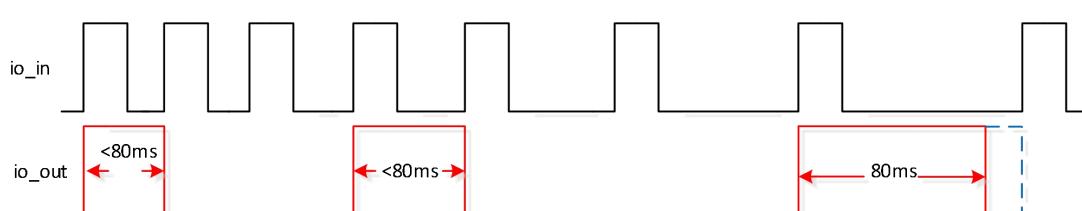
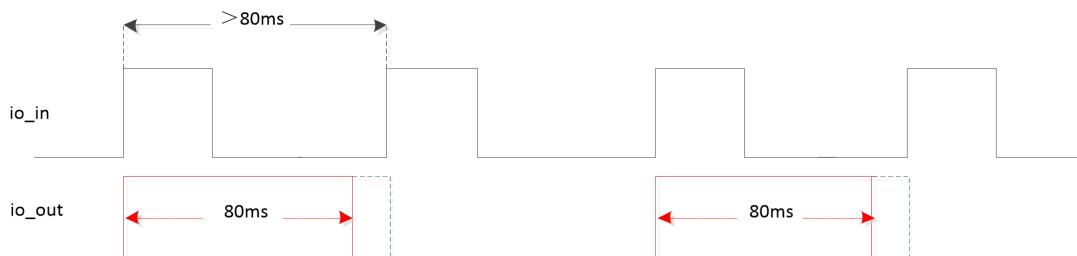


11.4.3.4 分频后有效电平 (高或低) 脉宽 $\geq 80\text{ms}$

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。



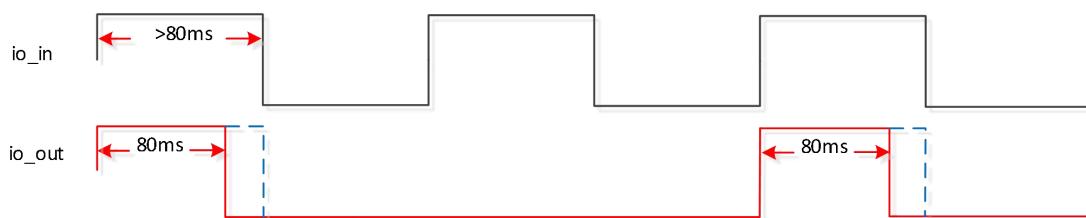


图 11.27 输入有效电平大于 80ms (二分频)

11.4.4 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。

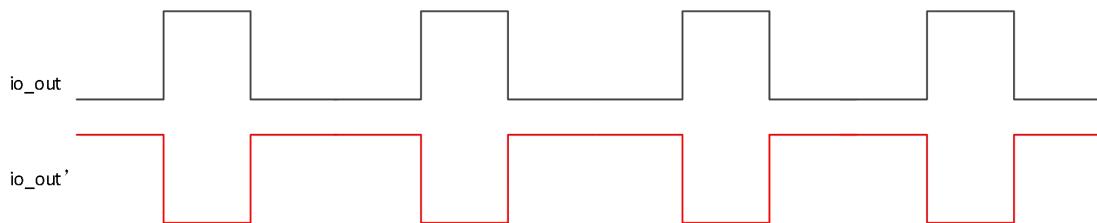


图 11.28 脉冲电平反向输出

11.5 寄存器描述

IOCNT 寄存器地址：

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

IOCNT 寄存器的偏移地址

寄存器名	地址偏移量	描述
CFG0	0X100	脉冲转发配置寄存器 0
CFG1	0X104	脉冲转发配置寄存器 1
CFG2	0X108	脉冲转发配置寄存器 2
CTRL	0X10C	脉冲转发控制寄存器

11.5.1 脉冲转发配置寄存器 0~2 (0x100~0x108)

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) ,	R/W	00

		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。		
19:0	CNT	输入脉冲计数器，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

11.5.2 脉冲转发控制寄存器 IOCNT_CTL (0x10C)

比特位	名称	描述	读 / 写 标志	复位值
31:17	--	预留	R	0
16	CNTREG_CLR	IOCNT 脉冲计数器类型选择 =0, 读后清零型 =1, 累加型 默认读后清零型	R/W	0
15:11	--	预留	R	0
10	IOCNT2_MODE	IOCNT2 脉冲计数边沿选择 =0, 上升沿计数 =1, 下降沿计数 默认上升沿计数	R/W	0
9	IOCNT1_MODE	IOCNT1 脉冲计数边沿选择 =0, 上升沿计数 =1, 下降沿计数 默认上升沿计数	R/W	0
8	IOCNT0_MODE	IOCNT0 脉冲计数边沿选择 =0, 上升沿计数 =1, 下降沿计数 默认上升沿计数	R/W	0
7:3	--	预留	R	0
2	CNT2_EN	IOCNT2 使能配置 =0, 不使能 =1, 使能 IOCNT	R/W	0
1	CNT1_EN	IOCNT1 使能配置 =0, 不使能 =1, 使能 IOCNT	R/W	0
0	CNT0_EN	IOCNT0 使能配置 =0, 不使能 =1, 使能 IOCNT	R/W	0

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修正；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。

11.6 操作步骤

11.6.1 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 (包括输入 INTx 和输出 IOCNTx_OUTx 的选择配置) ;
- 3) 可选配置分频计数和电平反向;
- 4) 配置脉冲转发使能。

11.6.2 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 (包括输入 INTx 和输出 IOCNTx_OUTx 的选择配置) ;
- 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。

12 外部中断控制器

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，可以在 cpu 休眠时通过中断自动唤醒 cpu。

12.1 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，运行模式下滤波时间约为 10 微秒；

12.2 寄存器描述

模块寄存器地址

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

12.2.1 INTC_CTL

INTC 控制寄存器地址 0x40044000+0x0

比特位	名称	描述	读/写标志	复位值
31:09	---	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

12.2.2 INTC_MODE

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 (INT7) 模式选择 00: 上升沿 01: 下降沿	R/W	0

		10: 双边沿 11: 保留		
13:12	MODE6	外部中断请求 6 (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 (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 (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 (INT3) 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留	R/W	0
5:4	MODE2	外部中断请求 2 (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 (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 (INT0) 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留	R/W	0

12.2.3 INTC_MASK

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

12.2.4 INTC_STA

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

13 KBI

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。

13.1 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KEY0~KEY7；
- 支持每个按键状态查询；
- 支持每个按键输入滤波，滤波时间约为24ms；
- 支持每个按键可单独屏蔽中断

13.2 寄存器描述

表 12-1 KBI 寄存器基址

模块名	物理地址	映射地址
KBI	0x40028000	0x40028000

表 12-2 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

13.2.1 控制寄存器 (0x0)

表 12-3 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

13.2.2 选择寄存器 (0x4)

表 12-4 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

13.2.3 数据寄存器 (0x8)

表 12-5 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

13.2.4 屏蔽寄存器 (0xC)

表 12-6 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

13.3 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

14 UART

芯片集成了 6 个 UART 接口，用于与外部进行异步串行通信。

14.1 概述

UART 接口控制器具备如下特性：

- 6 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，**32.768KHz** 主频下也支持
- 支持自动波特率检测
- 支持红外唤醒
- 发送支持 2 个深度的 FIFO，接收不支持 FIFO
- 支持通用 DMA 功能
- 支持 300 波特率@29MHz
- 最高支持 2048 波特率@32.768KHz
- 6 路 UART 均支持 DMA

14.2 寄存器描述

表 13-1 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 13-2 UART 寄存器偏移地址

寄存器名(X=0,1,2,3,4,5)	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_FDIV	0x14	UART 波特率小数分频配置寄存器
UARTx_DMA_CTL	0x18	UART DMA 控制寄存器
UARTx_DMA_TBADR	0x1c	UART DMA 发送起始地址寄存器
UARTx_DMA_RBAD	0x20	UART DMA 接收起始地址寄存器
UARTx_DMA_TLEN	0x24	UART DMA 发送长度寄存器
UARTx_DMA_RLEN	0x28	UART DMA 接收长度寄存器
UARTx_DMA_TADR	0x2c	UART 当前发送 DMA 地址寄存器
UARTx_DMA_RADR	0x30	UART 当前接收 DMA 地址寄存器

UARTx_DMA_IE	0x34	UART DMA 中断使能寄存器
UARTx_DMA_IF	0x38	UART DMA 中断标志寄存器
UARTx_DMA_TO	0x3C	UART 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。

14.2.1 UART 控制寄存器 (0x00)

表 13-3 UART 控制寄存器 UART_CTL

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

注: 当系统时钟为 32.768KHz 时, 也支持红外调制功能。

14.2.2 UART 波特率配置寄存器 (0x4)

表 13-4 UART 波特率配置寄存器 UART_BAUD

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_BADU"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 0xFFFF 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 $14745600/[16*(8191+1)] = 112$ bps, 最高波特率为 $14745600/16 = 921600$ bps。

14.2.3 UART 状态指示寄存器 (0x8)

表 13-5 UART 状态指示寄存器 UART_STA

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

14.2.4 UART 发送数据寄存器 (0xC)

表 13-6 UART 波特率配置寄存器 UART_TXD

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

14.2.5 UART 接收数据寄存器 (0x10)

表 13-7 UART 接收数据寄存器 UART_RXD

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

14.2.6 UART 波特率小数分频配置寄存器 (0x14)

表 13-8 UART 波特率小数分频配置寄存器 UARTx_FDIV

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> $FDIV = \left[\left(\frac{f_i}{16 \times f_o} - \left[\frac{f_i}{16 \times f_o} \right] \right) \times 2^{14} + 0.5 \right]$ <p>其中 f_i 为输入时钟(cpu 当前运行时钟), f_o 为输出时钟, “[]” 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> $FDIV = \left[\left(\frac{1843200}{16 \times 9837} - \left[\frac{1843200}{16 \times 9837} \right] \right) \times 2^{14} + 0.5 \right]$ <p>求得 $F=11647$。</p>	R/W	0

14.2.7 UART DMA 控制寄存器 (0x18)

表 13-9 UART DMA 控制寄存器 UART_DMA_CTL

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 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 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 1: 使能</p>	R/W	0
1	RX_DMA_EN	<p>接收 DMA 使能</p> <p>0: 不使能 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	
--	-------	--

14.2.8 UART DMA 发送起始地址寄存器 (0x1C)

表 13-10 UART DMA 发送起始地址寄存器 **UART_DMA_TBADR**

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

14.2.9 UART DMA 接收起始地址寄存器 (0x20)

表 13-11 UART DMA 接收起始地址寄存器 **UART_DMA_RBADR**

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

14.2.10 UART DMA 发送长度寄存器 (0x24)

表 13-12 UART DMA 发送长度寄存器 **UART_DMA_TLEN**

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

14.2.11 UART DMA 接收长度寄存器 (0x28)

表 13-13 UART DMA 接收长度寄存器 **UART_DMA_RLEN**

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

14.2.12 UART DMA 当前发送地址寄存器 (0x2C)

表 13-14 UART DMA 当前发送地址寄存器 **UART_DMA_TADR**

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

14.2.13 UART DMA 当前接收地址寄存器 (0x30)

表 13-15 UART DMA 当前接收地址寄存器 **UART_DMA_RADR**

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

14.2.14 UART DMA 中断使能寄存器 (0x34)

表 13-16 UART DMA 当前接收地址寄存器 **UART_DMA_IE**

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	R/W	0

		=1, 使能		
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

14.2.15 UART DMA 中断标志寄存器 (0x38)

表 13-17 UART DMA 中断标志寄存器 UART_DMA_IF

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

14.2.16 UART DMA 接收超时配置寄存器 (0x3C)

表 13-18 UART DMA 接收超时配置寄存器 UART_DMA_TO

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

14.3 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->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：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 MCU 的接收中断，接收时关闭发送中断，消除干扰。

15 ISO7816

内置两个 ISO7816 通道，支持外接 2 个 7816 协议接口设备。

15.1 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；

15.2 寄存器描述

表 14-1 ISO7816 寄存器基址

模块名	物理地址	映射地址
ISO7816	0x40038000	0x40038000

表 14-2 ISO7816 寄存器偏移地址

寄存器名	地址偏移量	描述
ISO7816_CTL0	0x0	控制寄存器 0
ISO7816_CTL1	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_STAT0	0x14	状态指示寄存器 0
ISO7816_STAT1	0x18	状态指示寄存器 1
ISO7816_DAT0	0x1c	数据发送寄存器 0
ISO7816_DAT1	0x20	数据发送寄存器 1

15.2.1 控制寄存器 (0x0)

表 14-3 ISO7816 控制寄存器 0 ISO7816_CTL0

比特位	名称	描述	读/写标志	复位值
31:28	---	预留	R	0
27	RX_GTO	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	---	预留	R/W	0
25	---	预留	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	R/W	0

		<p>沿之间的延迟: $12 \text{ etu} + (Q \times (N/f))$。</p> <p>公式中, Q应取下面两个值中的一个:</p> <ul style="list-style-type: none"> ——当复位应答中不存在T=15时, 取F/D; ——当复位应答中存在T=15时, 取 Fi/Di; <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> <ul style="list-style-type: none"> ——T=0时, 12etu ——T=1时, 11etu 		
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 010: 2 次 011: 3 次 100: 4 次 101: 5 次 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 01: 1 etu 10: 1.5 etu 11: 2 etu</p>	R/W	01
10:8	PARSEL0	<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_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 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 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 1: 使能数据接收产生中断</p>	R/W	0

4	TX_IRQ_EN0	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 0: 禁止数据发送产生中断 1: 使能数据发送产生中断	R/W	0
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

15.2.2 ISO7816 控制寄存器 1 (0x04)

表 14-4 ISO7816 控制寄存器 1 ISO7816_CTL1

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	---	预留	R/W	0
26	---	预留	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	R/W	0

		<p>设备发送的) 的起始沿开始的下列延迟:</p> $12 \text{ etu} + (Q \times (N/f))$ <p>公式中, Q应取下面两个值中的一个:</p> <p>F/D, 即用于计算etu的值, 当T=15不存在于复位应答中时,</p> <p>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</p> <p>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</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_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。</p> <p>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	ERRWTH1	<p>错误信号宽度选择位</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	PARSEL1	<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_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	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 0: 禁止数据发送产生中断 1: 使能数据发送产生中断	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

15.2.3 ISO7816_CLK (0x08)

表 14-5 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	---	预留	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: 不支持; 110: 不支持; 111: 不支持;	R/W	0

15.2.4 ISO7816 波特率系数 0 寄存器 (0x0c)

表 14-6 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	---	预留	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

15.2.5 ISO7816 波特率系数 1 寄存器 (0x10)

表 14-7 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	---	预留	R	0
21	FDS0_EN	使能软件配置 F/D 的系数。	R/W	0

		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		
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

15.2.6 ISO7816 状态 0 寄存器 (0x14)

表 14-8 ISO7816 状态 0 寄存器 ISO7816_STAT0

比特位	名称	描述	读/写标志	复位值
31:12	---	预留	R	0
11	FRAME_E RR0	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 RIF0	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_ER RIF0	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。	R/W	0

		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		
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

15.2.7 ISO7816 状态 1 寄存器 (0x18)

表 14-9 ISO7816 状态 1 寄存器 ISO7816_STAT1

比特位	名称	描述	读/写标志	复位值
31:14	---	预留	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_E_RR0	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。 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_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。	R/W	0

		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		
--	--	---	--	--

15.2.8 ISO7816 数据 0 寄存器 (0x1C)

表 14-10 ISO7816 数据 0 寄存器 ISO7816_DAT0

比特位	名称	描述	读/写标志	复位值
保留	--	--	R	0
8	DATA0[8]	当 parsel 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

15.2.9 ISO7816 数据 1 寄存器 (0x20)

表 14-11 ISO7816 数据 1 寄存器 ISO7816_DAT1

比特位	名称	描述	读/写标志	复位值
保留	--	--	R	0
8	DATA1[8]	当 parsel 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

15.3 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。
- 6、可通过中断方式或查询状态方式向 7816 总线读写数据。

15.4 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 大部分使用高速光耦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，在高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 MCU 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，因与卡通讯使用光耦隔离，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 MCU 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。

16 I²C 接口

内置一个 I²C 接口控制器。

16.1 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；

16.2 寄存器描述

表 15-1 I²C 寄存器基址

模块名	物理地址	映射地址
I ² C	0x40024000	0x40024000

表 15-2 I²C 寄存器偏移地址

寄存器名	地址偏移量	描述
I ² C_CTL	0x0	控制寄存器
I ² C_CLK	0x4	时钟配置寄存器
I ² C_STAT	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

16.2.1 控制寄存器 (0x0)

表 15-3 控制寄存器 I²C_CTL

比特位	名称	描述	读/写标志	复位值
31:6	---	预留	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	R/W	0

		11: 保留		
0	EN	模块使能 1: I ² C 打开 0: I ² C 关闭	R/W	0

16.2.2 时钟配置寄存器 (0x4)

表 15-4 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	---	预留	R	0														
2:0	CLKDIV	I ² C 时钟分频参数选择位： I ² C 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 I ² C 高速/正常模式通信时钟。 当配置选项不在下表中时, 默认选择为 10 分频。 <table border="1"> <thead> <tr> <th>系统主频</th> <th>分频参数 /CLKDIV(m)</th> </tr> </thead> <tbody> <tr> <td></td> <td>高速模式 正常模式</td> </tr> <tr> <td>1.8432MHz</td> <td>不支持 010 (20)</td> </tr> <tr> <td>3.6864MHz</td> <td>不支持 011 (38)</td> </tr> <tr> <td>7.3728MHz</td> <td>010 (20) 100 (76)</td> </tr> <tr> <td>14.7456MHz</td> <td>011 (38) 101 (154)</td> </tr> <tr> <td>29.4912MHz</td> <td>100 (76) 110 (306)</td> </tr> </tbody> </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	不支持 011 (38)	7.3728MHz	010 (20) 100 (76)	14.7456MHz	011 (38) 101 (154)	29.4912MHz	100 (76) 110 (306)	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	不支持 011 (38)																	
7.3728MHz	010 (20) 100 (76)																	
14.7456MHz	011 (38) 101 (154)																	
29.4912MHz	100 (76) 110 (306)																	

当主频时钟频率为 1.8432/3.6864/7.3728/14.7456/29.4912MHz 后, 其分频系数及接口时钟误差如下:

系统主频 Mhz	高速模式 (400K)	分 频 系数	接 口 时 钟 (KHz)	误 差	正 常 模 式	分 频 系 数	接 口 时 钟 (KHz)	误 差
1.8432	不支持	/	/	/	010 (20)	20	92.160	-8%
3.6864	不支持	/	/	/	011 (38)	38	97.010	-3%
7.3728	010 (20)	20	368.640	-8%	100 (76)	76	97.010	-3%
14.7456	011 (38)	38	388.042	-3%	101 (154)	154	95.751	-4%
29.4912	100 (76)	76	388.042	-3%	110 (306)	306	96.376	-4%

I²C 时钟频率详细配置表:

CLKD IV	分频系 数 m	系统主频 APB_CLK (MHz) /I ² C 时钟 (KHz)				
		1.8432	3.6864	7.3728	14.7456	29.4912
000	6	307.200	614.400	1228.800	2457.600	4915.200
001	10	184.320	368.640	737.280	1474.560	2949.120
010	20	92.160	184.320	368.640	737.280	1474.560
011	38	48.505	97.010	194.021	388.042	776.084
100	76	24.252	48.505	97.010	194.021	388.042
101	154	11.968	23.937	47.875	95.751	191.501
110	306	6.023	12.047	24.094	48.188	96.376
111	10	184.320	368.640	737.280	1474.560	2949.120

SCL=APBCLK/m

16.2.3 状态指示寄存器 (0x8)

 表 15-5 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	---	预留	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 0: 没有触发溢出中断 1: 触发溢出中断	R/W	0
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

16.2.4 从设备地址寄存器 (0xC)

 表 15-6 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	---	预留	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

16.2.5 收发数据寄存器 (0x10)

 表 15-7 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	---	预留	R	0
7:0	TRDAT	接收/发送数据	R/W	0

17 SPI 接口

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。

17.1 概述

SPI 接口控制器具备如下特性:

- 4 个全双工 SPI;
- 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 3.6864MHz (fcpu/8) ;
- 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

17.2 复用关系

- ✓ SPI0: P50/P51/pP46/P47、P110~P113, 不可同时复用为 SPI0
- ✓ SPI1: P40~P43、P52~P55, 不可同时复用
- ✓ SPI2: P40~P43
- ✓ SPI3: P104~P107

17.3 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 16-1 SPI 时钟工作方式

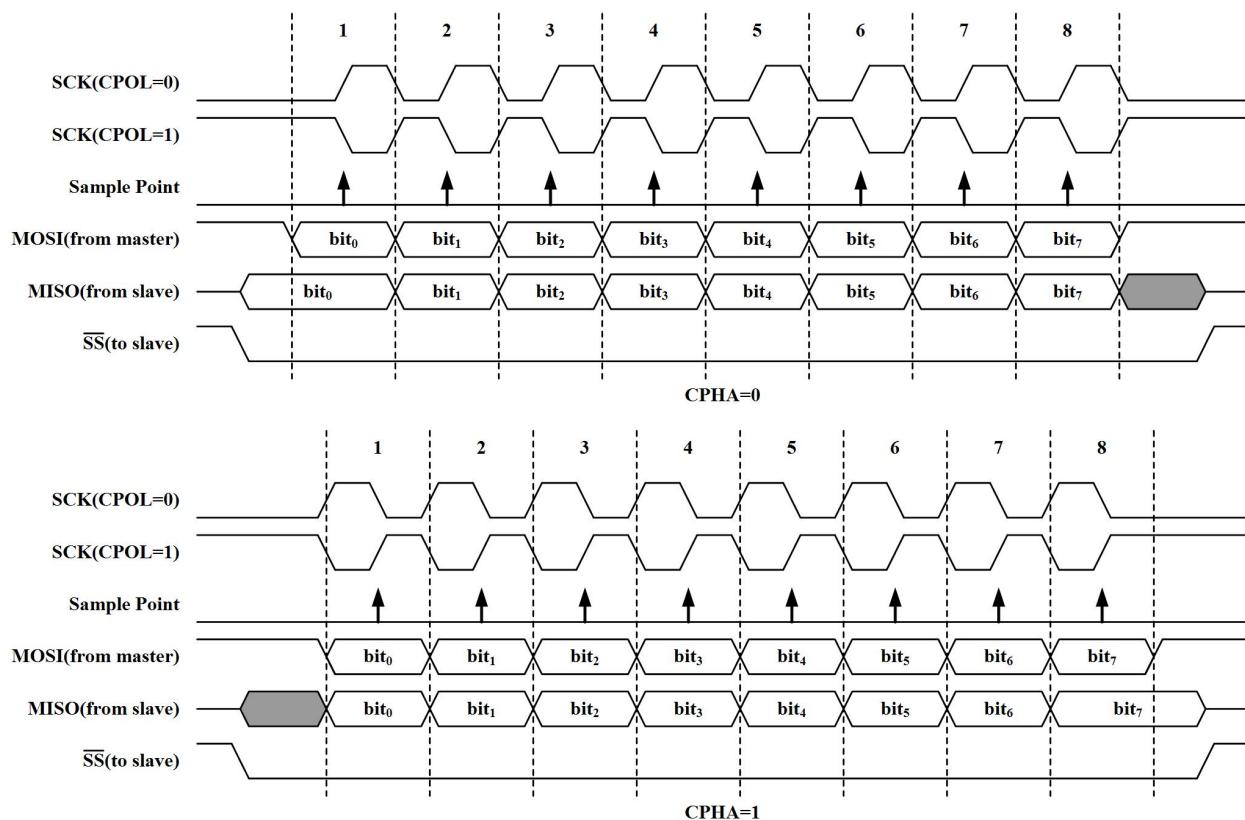
SPI 模式	CPOL/CPHA	前边沿	后边沿
0	0/0	上升沿, 数据采样	下降沿, 数据建立
1	0/1	上升沿, 数据建立	下降沿, 数据采样
2	1/0	下降沿, 数据采样	上升沿, 数据建立
3	1/1	下降沿, 数据建立	上升沿, 数据采样

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 16-1 SPI 时钟工作方式



接收数据溢出: 在下一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 输入为高, 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。

17.4 寄存器描述

表 16-2 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 16-3 SPI 寄存器偏移地址

寄存器名 (X=0,1,2,3)	地址偏移量	描述
SPIx_CTL	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_DMA_CTL	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

17.4.1 SPI 控制寄存器 (0x0)

表 16-4 SPI 控制寄存器 SPI_CTL

比特位	名称	描述	读/写标志	复位值
31:27	---	预留	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, 发送数据内容。 1: 发送 SPI_TXDFLT 寄存器值 0: 发送上次传输 SPI_TXDATA 的值	RW	0
21	---	保留位, 默认为 0, 不要写 1	R/W	0
20	---	保留位, 默认为 0, 不要写 1	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 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	R/W	0

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

17.4.2 SPI 状态寄存器 (0x4)

表 16-5 SPI 状态寄存器 SPI_STAT

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	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 1: 产生发送数据空冲突中断	R/W	0
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	R	0

		1: 数据正在发送过程中, 总线不能发写 SPITX 寄存器命令		
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

17.4.3 SPI 数据发送寄存器 (0x8)

表 16-6 SPI 数据发送寄存器 SPI_TXDATA

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

17.4.4 SPI 数据接收寄存器 (0xC)

表 16-7 SPI 数据接收寄存器 SPI_RXDATA

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

17.4.5 SPI 默认发送数据寄存器 (0x10)

表 16-8 SPI 默认发送数寄存器 SPI_TXDFLT

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/W	0

17.4.6 SPI DMA 控制寄存器 (0x14)

表 16-9 SPI DMA 控制寄存器 SPI_DMA_CTL

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

17.4.7 SPI DMA 发送起始地址寄存器 (0x18)

表 16-10 SPI DMA 发送起始地址寄存器 SPI_DMA_TBADR

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

17.4.8 SPI DMA 接收起始地址寄存器 (0x1C)

表 16-11 SPI DMA 接收起始地址寄存器 SPI_DMA_RBADR

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

17.4.9 SPI DMA 发送长度寄存器 (0x20)

表 16-12 SPI DMAFA 发送长度寄存器 SPI_DMA_TLEN

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

17.4.10 SPI DMA 接收长度寄存器 (0x24)

表 16-13 SPI DMAFA 接收长度寄存器 SPI_DMA_RLEN

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

17.4.11 SPI DMA 当前发送地址寄存器 (0x28)

表 16-14 SPI DMAFA 当前发送地址寄存器 SPI_DMA_TADR

位	位名称	功能描述	读写	复位值

			标志	
31:17	Reserved	保留。	R	0
16:0	DMA_TADR	当前发送 DMA 地址 (Byte 地址)	RO	0

17.4.12 SPI DMA 当前接收地寄存器 (0x2C)

表 16-15 SPI DMAFA 当前接收地址寄存器 SPI_DMA_RADR

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

17.4.13 SPI DMA 中断使能寄存器 (0x30)

表 16-16 SPI DMAFA 中断使能寄存器 SPI_DMA_IE

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

17.4.14 SPI DMA 中断标志寄存器 (0x34)

表 16-17 SPI DMAFA 中断标志寄存器 SPI_DMA_IF

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

18 电能积分单元 D2F

18.1 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 Hfconst3，根据输入功率和脉冲常数 Hfconst3 进行积分，电能存放在 D2FE00~D2FE11 寄存器，其中 D2FP0~D2FP2 支持脉冲输出，分别为 D2F_OUT0~D2F_OUT2，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。

该模块的地址为：0x4005C000

18.2 特点

- 12 路 D2F
- 脉冲常数可配；
- 功率为 28 位有符号数，输入范围为 $\pm 2^{27}$ ；
- 支持脉冲中断；
- 支持输出 3 路校验脉冲 D2F_OUT0~D2F_OUT2；
- 能量寄存器模式可配：累加型或读后清零型；

18.3 寄存器描述

模块名	物理地址	映射地址
D2F	0x4005C000	0x4005C000
寄存器名	地址偏移量	描述
HFconst3	0x0	D2F 高频脉冲常数寄存器
IE	0x4	D2F 中断使能寄存器
IF	0x8	D2F 中断标志寄存器
CFG	0xC	D2F 配置寄存器
D2FP00~D2FP11	0x10~0x3C	D2F 功率输入寄存器
D2FE00~D2FE11	0x40~0x6C	D2F 自定义能量寄存器

以上寄存器不支持 bitband 操作。

18.3.1

D2F 高频脉冲常数寄存器 HFCONST3(0x00)

位	位名称	功能描述	读写标志	复位值
31:16	Reserved	保留	R	0
15:0	HFCONST3	12 个自定义电能寄存器 D2FE00-D2FE11 及 3 个自定义 CF 脉冲 CF0-CF2 的脉冲常数。高频脉冲常数寄存器是 16 位无符号数。	R/W	0

HFCONST3 用于 12 个自定义电能寄存器 D2FE00-D2FE11 及 3 个自定义 CF 脉冲 CF0-CF2 的脉冲常数。

高频脉冲常数寄存器是 16 位无符号数。

HFCONST3 用于 12 个自定义电能寄存器 D2FE00-D2FE11 及 3 个自定义 CF 脉冲 CF0-CF2 的脉冲常数。

高频脉冲常数寄存器是 16 位无符号数。

电能积分单元 D2F, HFCONST3 的计算公式如下：

$$HFCONST3=INT[(P/2^{27})*3.6*10^6*femu / (4*EC**Preal)]$$

上述公式中各符号含义如下：

P: P 为 Preal 对应的功率寄存值，也就是填入 D2FP 寄存器的功率寄存器值，无物理意义，根据用户需

求定义, 定义后可得到功率转换系数: $kp=Preal/P$

Preal: 实际功率值, PF=1.0 时, Preal = $U \cdot I$, U, I 分别为用电电压和电流有效值

femu: MCU 芯片 D2F 积分的时钟是 $3.6864\text{MHz}/8=0.4608\text{MHz}$;

EC: 电表常数

如果该芯片(HFCONST3)与锐能微三相计量芯片(HFCONST1)对接, 假设二者的电表脉冲常数 EC 定义的完全一致, 那么只要满足:

$$HFCONST3 = (0.4608/1.024) * HFCONST1$$

直接读取锐能微三相计量芯片的功率寄存器值 (32bit, PowerP 等), 然后取 32bit 的高 28bit 填入功率寄存器 D2FP, 然后脉冲输出 CF 就能与三相计量芯片对应的 CF 引脚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 HFCONST3 和自定义功率寄存器填入值均可。

如果该芯片(HFCONST3)与锐能微单相计量芯片(HFCONST1)对接, 假设二者的电表脉冲常数 EC 定义的完全一致, 那么只要满足:

$$HFCONST3 = (0.4608/0.89488625/2) * HFCONST1$$

单相计量芯片 D2F 积分的时钟是 $3.579545/4=0.89488625\text{MHz}$, 直接读取锐能微单相计量芯片的功率寄存器值 (32bit, PowerP 等), 然后取 32bit 的高 28bit 填入功率寄存器 D2FP, 然后脉冲输出 CF 就能与单相计量芯片对应的 CF 引脚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 HFCONST3 和自定义功率寄存器填入值均可。

注 1: 锐能微三相计量芯片的 HFCONST1 计算公式如下

$$HFCONST1 = INT[(P/2^{31}) * 3.6 * 10^6 * femu / (16 * EC * Un * Ib)]$$

上述公式中各符号含义如下:

P: 额定输入, PF=1.0, 三相计量芯片标准的有功功率值寄存器值 $P = INT(U_{\text{标准}} * I_{\text{标准}} / 2^{23})$, 式中电压额定输入时标准的电压有效值寄存器值为 $U_{\text{标准}}$, 电流额定输入时标准的电流有效值寄存器值 $I_{\text{标准}}$

femu: 三相计量芯片计量电能积分时钟 4.096Mhz;

Un: 额定输入的电压;

Ib: 额定输入的电流;

EC: 电表常数

注 2: 锐能微单相计量芯片的 HFCONST1 计算公式如下:

$$HFCONST1 = INT[(P/2^{31}) * 3.6 * 10^6 * femu / (2 * EC * Un * Ib)]$$

上述公式中各符号含义如下:

P: 额定输入, PF=1.0, 单相计量芯片标准的有功功率值寄存器值 $P = INT(U_{\text{标准}} * I_{\text{标准}} / 2^{15})$, 式中电压额定输入时标准的电压有效值寄存器值为 $U_{\text{标准}}$, 电流额定输入时标准的电流有效值寄存器值 $I_{\text{标准}}$

femu: 电能积分时钟, 系统时钟/4, 如 $3.579545/4=0.89488625\text{MHz}$;

Un: 额定输入的电压;

Ib: 额定输入的电流;

EC: 电表常数

18.3.2

D2F 中断使能寄存器 IE(0x04)

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)。

位	位名称	功能描述	读写标志	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	D2F9IE	电能积分单元 D2F9 中断使能	R/W	0
8	D2F8IE	电能积分单元 D2F8 中断使能	R/W	0
7	D2F7IE	电能积分单元 D2F7 中断使能	R/W	0
6	D2F6IE	电能积分单元 D2F6 中断使能	R/W	0
5	D2F5IE	电能积分单元 D2F5 中断使能	R/W	0
4	D2F4IE	电能积分单元 D2F4 中断使能	R/W	0
3	D2F3IE	电能积分单元 D2F3 中断使能	R/W	0
2	D2F2IE	电能积分单元 D2F2 中断使能	R/W	0
1	D2F1IE	电能积分单元 D2F1 中断使能	R/W	0
0	D2F0IE	电能积分单元 D2F0 中断使能	R/W	0

18.3.3

D2F 中断标志寄存器 IF(0x08)

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(中断号：4)。

位	位名称	功能描述	读写标志	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	D2F9IF	电能积分单元 D2F9 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	D2F8IF	电能积分单元 D2F8 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	D2F7IF	电能积分单元 D2F7 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	D2F6IF	电能积分单元 D2F6 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	D2F5IF	电能积分单元 D2F5 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	D2F4IF	电能积分单元 D2F4 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	D2F3IF	电能积分单元 D2F3 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	D2F2IF	电能积分单元 D2F2 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	D2F1IF	电能积分单元 D2F1 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	D2F0IF	电能积分单元 D2F0 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

18.3.4

D2F 配置寄存器 CFG(0x0C)

位	位名称	功能描述	读写标志	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

18.3.5

D2F 功率输入寄存器 D2FP(0x10~0x3C)

偏移地址	10H	14H	18H	1CH	20H	24H	28H
寄存器	D2FP00	D2FP01	D2FP02	D2FP03	D2FP04	D2FP05	D2FP06
默认值	0x0						
偏移地址	2CH	30H	34H	38H	3CH		
寄存器	D2FP07	D2FP08	D2FP09	D2FP10	D2FP11		
默认值	0x0	0x0	0x0	0x0	0x0		

功率输入寄存器 D2FP00-11 可采用二进制补码格式, 28 位有符号数, 其中最高位是符号位。

当往 D2FP 寄存器写入功率值时, 该功率会按照脉冲常数 Hfconst3 进行积分, 积分的电能存放在 D2FE00~D2FE11 寄存器中, 其中 D2FE00、D2FE01、D2FE02 可以输出校验脉冲, 分别对应 D2F_OUT0、D2F_OUT1、D2F_OUT2, 相关复用配置见 GPIO 章节。

18.3.6

D2F 自定义能量寄存器 D2FE(0x40~0x6C)

偏移地址	40H	44H	48H	4CH	50H	54H	58H
寄存器	D2FE00	D2FE01	D2FE02	D2FE03	D2FE04	D2FE05	D2FE06
默认值	0x0						
偏移地址	5CH	60H	64H	68H	6CH		
寄存器	D2FE07	D2FE08	D2FE09	D2FE10	D2FE11		
默认值	0x0	0x0	0x0	0x0	0x0		

自定义能量寄存器为 24bit 寄存器 D2FE[23:0], 自定义能量寄存器可以选择为累加型或者读后清零型。

19 安全密码加速器 SEA

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/Programs/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

19.1 AES 硬件加速单元

19.1.1 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(2^{128})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；

19.1.2 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)。

19.2 ECC 硬件加速单元

19.2.1 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 Diffie-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。

19.2.2 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_Brainpool512r1	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)。

19.3 RSA 硬件加速单元

19.3.1 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 $\phi(N)=(P-1)(Q-1)$ 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;

19.3.2 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）。

19.4 HASH 硬件加速单元

19.4.1 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；

19.4.2 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）

19.5 TRNG 真随机数发生器

19.5.1 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；

19.5.2 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）

19.6 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”。

20 简单定时器 SIMP_TC

20.1 概述

模块内部集成了 4 个的 16 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；

20.2 功能描述

SIMP_TC 模块是一个简单的 16 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。

20.3 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

20.3.1

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	R/W	0

		0: 单次计数 1: 循环计数		
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

20.3.2

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
15:0	LOAD	目标计数值: LOAD + 1	R/W	0

20.3.3

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
15:0	VAL	计数器当前计数值	R	0

21 内存搬运单元 M2M

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单位为 1Word, 同时还支持源数据和目标数据有效位配置。

21.1 特点

- 源和目标地址可配置, 地址按 Word 对齐
- 源数据长度可配置, 最小单位为 Word
- 输入数据有效位可配置
- 输出数据有效位可配置
- 可配置每个 Word 输出数据按 Byte 反序存放到目标地址
- 可配置输出数据按照 Word 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

21.2 功能描述

21.2.1 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 Word 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, 每个 Word 数据, 可配置 1~4Bytes 有效。

输入数据的 byte 无效表示源数据中, 每个 Word 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, 每个 Word 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 字节的功能;
 - 也可通过输出输出的组合实现其他功能, 具体实现参见“[实例说明章节](#)”。

21.2.2 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。

21.2.3 输出数据反序放置

配置寄存器 M2M_MODE.ORV_EN=1，可使能输出 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。

21.2.4 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。

21.2.5 数据按地址反序排列

配置寄存器为 M2M_MODE.ADDR_RVEN=1，可使能输出数据实现 Word 地址反序放置。该地址反序功能仅支持按 Word 对齐。每次取到源数据的 1 个 Word 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 1 个 Word。以此类推，直到存完最后一个 Word。注意：使用地址反序放置时，必须保证 M2M_MODE.IVLD = 4'b1111，M2M_MODE.OVLD = 4'b1111，M2M_SADDR[1:0] = 0，M2M_DADDR[1:0] = 0，M2M_HLEN = 4* (为大于 0 的整数)。

21.3 实例说明

21.3.1 輸入數據 4Bytes 有效，輸出 3Bytes 有效

M2M_MODE[4:0] = 11111

M3M_MODE_QVID[3:0] = 0111

Data03	Data02	Data01	Data00
Data13	Data12	Data11	Data10
Data23	Data22	Data21	Data20



Dummy	Data02	Data01	Data00
Dummy	Data11	Data10	Data03
Dummy	Data20	Data13	Data12
Dummy	Data23	Data22	Data21

21.3.2 输入数据 2Bytes 有效，输出 4Bytes 有效

M2M MODE [VLR[3:0] = 0011]

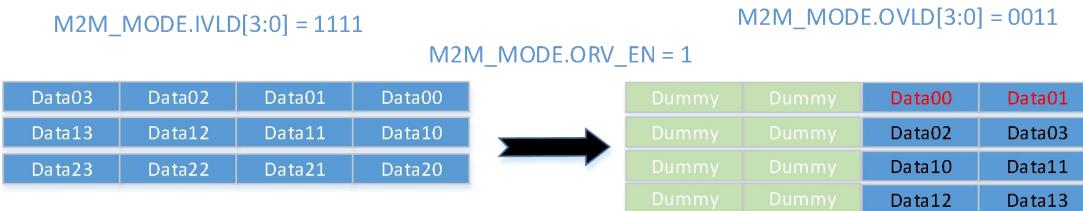
MON, MORE SWIPE[s] 1111

Dummy	Dummy	Data01	Data00
Dummy	Dummy	Data11	Data10
Dummy	Dummy	Data21	Data20
Dummy	Dummy	Data31	Data30
Dummy	Dummy	Data41	Data40



Data11	Data10	Data01	Data00
Data31	Data30	Data21	Data20
		Data11	Data10

21.3.3 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。

21.3.4 输出数据按地址反序排列



从上图可以看到, 源数据的第一个 Word 的数据 (data03, data02, data01, data00), 写到目的地址 daddrn 中, 之后目的地址减 4, 存储下一次取到源数据的 Word。以此类推, 直到存完最后一个 Word。

21.4 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。

21.5 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_IEN	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。

21.5.1 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	RW	0

		=1: 按 Word 地址反序排列 =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;		
9	DUMMY_EN		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

21.5.2 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

21.5.3 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

21.5.4 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, 需按 Word 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

21.5.5 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, 需按 Word 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

21.5.6 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

21.5.7 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

21.5.8 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

21.6 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;

22 DSP 核

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。

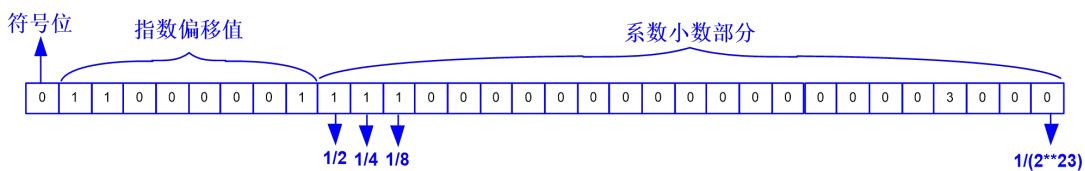
此处所说的浮点数均为单精度浮点数（小数点后有效数字 23 位（2 进制））。

22.1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;

22.2 计算引擎基本原理

22.2.1 单精度浮点数表示



上图为浮点数表示方法。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。

22.2.2 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。

22.2.3 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;

22.2.4 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

22.2.5 本芯片的 rounding 模式

本芯片支持 6 种舍入方法, 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;

22.2.6 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: $fp_sign = \text{bit}(31)$;
 - 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;

22.2.7 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”, 捂成 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;

22.2.8 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 种溢出处理方式, 通过寄存器配置。

22.2.9 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;

22.2.10 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:

$$X_1(k) = X_{1r}(k) + jX_{1i}(k)$$

$$X_2(k) = X_{2r}(k) + jX_{2i}(k)$$

$$W_N^k = W_{Nr}^k + jW_{Ni}^k$$

$$X'_1(k) = X'_{1r}(k) + jX'_{1i}$$

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，

$$\begin{aligned} X'_{1r} &= X_{1r} + W_{Nr}^k X_{2r} + W_{Ni}^k X_{2i} \\ X'_{1i} &= X_{1i} + W_{Nr}^k X_{2i} - W_{Ni}^k X_{2r} \end{aligned}$$

同理：

$$\begin{aligned} X'_{2r} &= X_{1r} - W_{Nr}^k X_{2r} - W_{Ni}^k X_{2i} \\ X'_{2i} &= X_{1i} - W_{Nr}^k X_{2i} + W_{Ni}^k X_{2r} \end{aligned}$$

22.2.11

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)。

22.2.12

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：

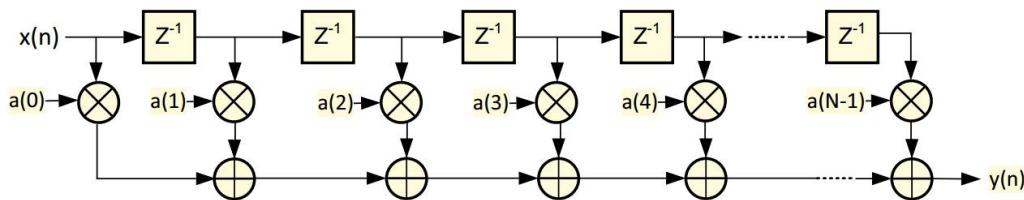


Figure 1-1: Conventional Tapped Delay Line FIR Filter Representation

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：
 $y[n] = a0*x[n] + a1*x[n-1] + a2*x[n-2] + a3*x[n-3] + a4*x[n-4]$

22.2.13 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
```

22.2.14 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 $step$ ：

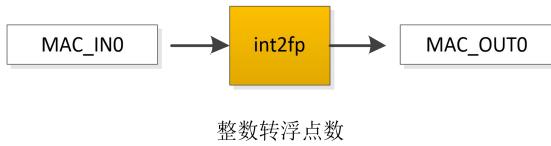
```
for i=1:1:out_len
  n=floor(t);
  y[i]=(t-n)*0.5*((t-n-1)*x[n-1]+(t-n+1)*x[n+1]) - (t-n+1)*(t-n-1)*x[n];
  t=t+step;
```

End

22.3 运算指令

22.3.1 整数转浮点数 (int2fp/int2fp_dma)

22.3.1.1 单数据转换



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。

22.3.1.2 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。

22.3.2 浮点数转整数 (fp2int/fp2int_dma)

22.3.2.1 单数据转换



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。

22.3.2.2 多数据连续转换

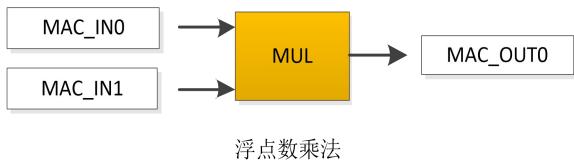
该模式支持把 SRAM 中一整段浮点数转化为整数。并且结果写回 SRAM。可配置目标地址与源数据地址一致，这样会做到转换完数据覆盖原始数据，从而节省 SRAM 空间。否则配置目标地址与源数据地址不同，会把转换后的数据写到 SRAM 中另外区域。

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。

22.3.3 浮点数乘法 (fp_mult)



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。

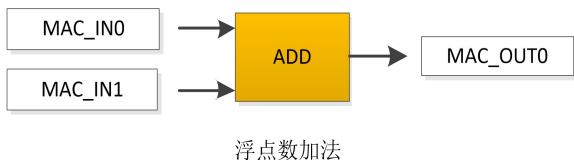
软件操作流程：

- 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。

22.3.4 浮点数加法 (fp_add)



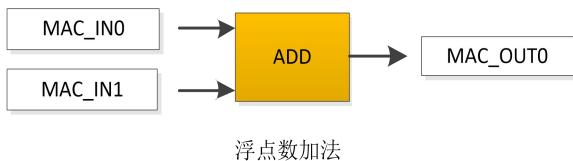
该模式对输入寄存器 MAC_IN0 和 MAC_IN1 的数据，做一个浮点数加法。输入寄存器的数据格式必须为浮点数格式，输出结果也为浮点数格式。

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。

22.3.5 浮点数减法 (fp_sub)



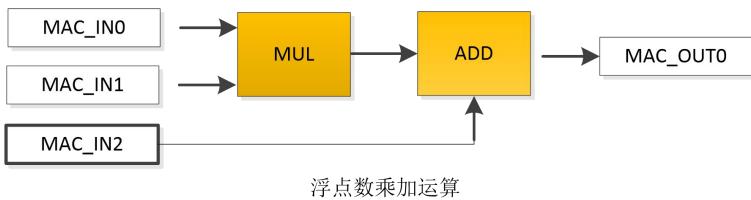
该模式对输入寄存器 MAC_IN0 和 MAC_IN1 的数据，做一个浮点数减法。输入寄存器的数据格式必须为浮点数格式，输出结果也为浮点数格式。

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。

22.3.6 浮点数乘加运算 (fp_mlad)



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。

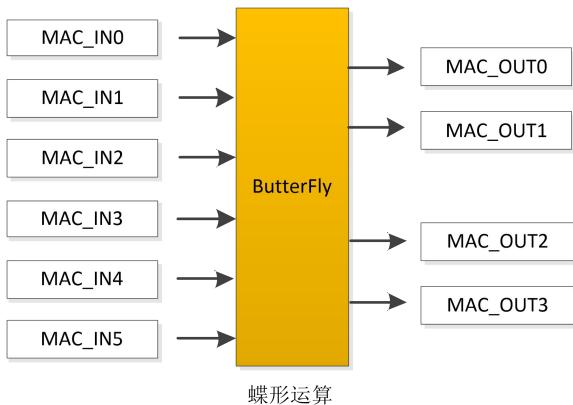
软件操作流程:

- 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。

22.3.7 浮点数蝶形运算（单次）（btfy /btfy_dma）

22.3.7.1 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。

$$\begin{aligned}
 X'_{1r} &= X_{1r} + W_{Nr}^k X_{2r} + W_{Ni}^k X_{2i} \\
 X'_{1i} &= X_{1i} + W_{Nr}^k X_{2i} - W_{Ni}^k X_{2r} \\
 X'_{2r} &= X_{1r} - W_{Nr}^k X_{2r} - W_{Ni}^k X_{2i} \\
 X'_{2i} &= X_{1i} - W_{Nr}^k X_{2i} + W_{Ni}^k X_{2r}
 \end{aligned}$$

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；

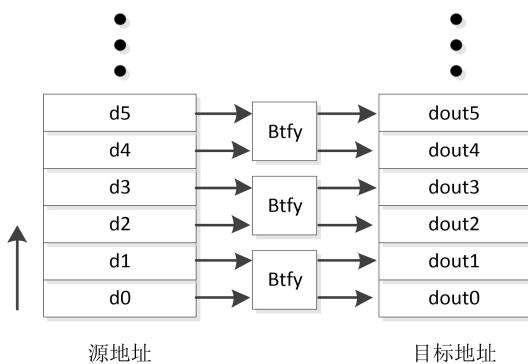
软件操作流程：

- 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；
 ■ MAC_IN0= X_{1r}
 ■ MAC_IN1= X_{1i}
 ■ MAC_IN2= X_{2r}
 ■ MAC_IN3= X_{2i}
 ■ MAC_IN4= W_{Nr}^k
 ■ 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；
 ■ MAC_OUT0= X'_{1r}
 ■ MAC_OUT1= X'_{1i}
 ■ MAC_OUT2= X'_{2r}
 ■ 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。

22.3.7.2 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。

22.3.7.3 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;

22.3.7.4 倒序搬运数据 (bitrev_move)

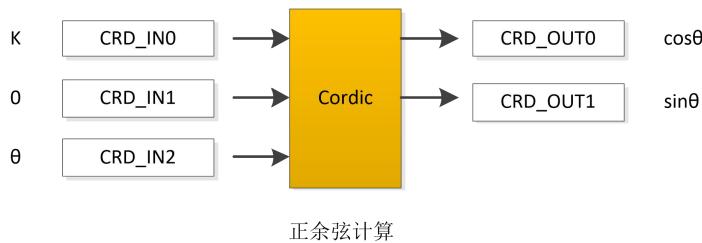
该模式下, 硬件自动把 SRAM 中一段原始数据根据 bit reverse 算法, 重新排序, 并且写到 SRAM 中指定位置。

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;

22.3.8 正弦余弦计算 (sin_cos)



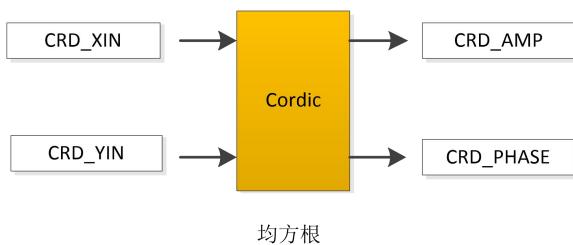
该模式支持输入角度, 硬件计算 sin 和 cos 函数。

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 = $\text{radian} * (2^{32}) / (2 * \pi)$);
- 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;

22.3.9 均方根和反正切 (fp_sqrt/ fp_atan)



该模式计算输入数据 CRD_XIN 和 CRD_YIN 的均方根。

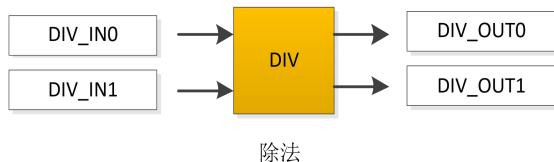
采用 cordic 向量模式, 配置 $X_0=X$, $Y_0=Y$, $Z_0 = 0$, 输出为 $\left[P \sqrt{X_i^2 + Y_i^2}, 0, \arctan\left(\frac{Y_i}{X_i}\right) \right]$

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；

$$CRD_AMP = \sqrt{X_0^2 + Y_0^2}/4*P, CRD_PHASE = \arctan(\frac{Y_0}{X_0}) * 2^{31}/\pi$$

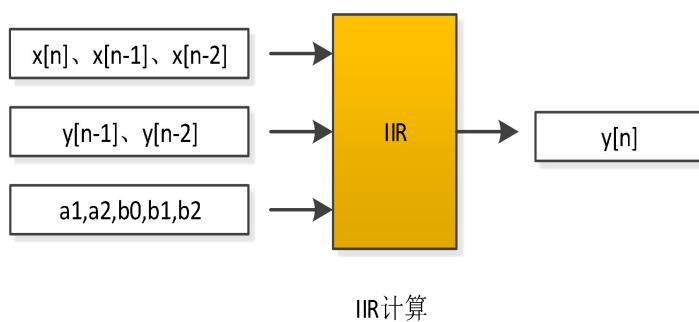
如果配置了中断使能信号，则计算完成之后，产生中断。

22.3.10 浮点数除法器 (fp_div)


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；

22.3.11
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 $y[n]$ 。

22.3.11.1 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]

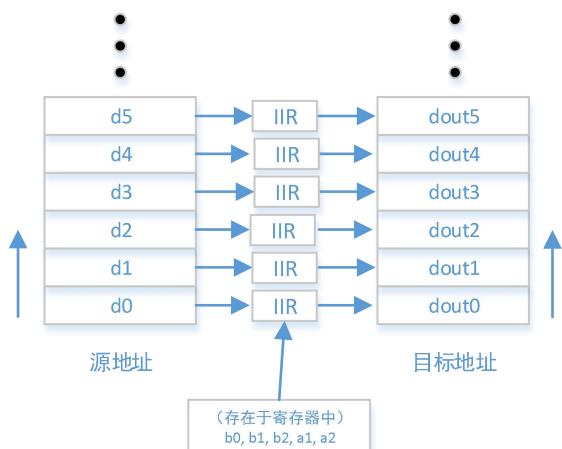
- MAC_IN2=x[n-2]
 - MAC_IN3=b₀
 - MAC_IN4=b₁
 - MAC_IN5=b₂
 - DIV_IN0=a₁
 - DIV_IN1=a₂
 - MAC_OUT0=y[n]
 - MAC_OUT1=y[n-1]
 - MAC_IN0=x[n]
- 4) 等待标志位 MAC_FLG[10] (IIR_ONCE_DONE) 置起, 读取结果 MAC_OUT0;
 ■ MAC_OUT0=y[n]

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 a1,a2, IIR 运算过程中不能改变 DIV_IN0/1。即 IIR 和除法不能同时使用)

22.3.11.2 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= $y[n-1]$

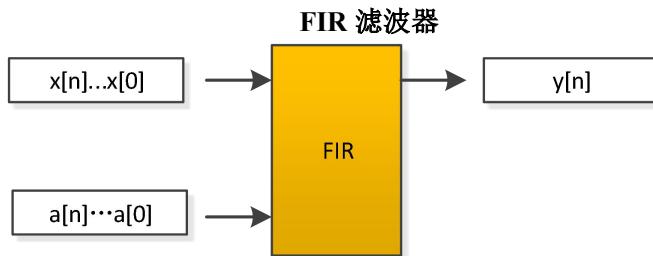
4) 配置好数据 $x[n]$ 起始地址 DMA_SRBADR、数据 $y[n]$ 目标地址 DMA_TRBADR，数据长度 DMA_LEN，DMA 读 SRAM 步进 MAC_CTL1[3:1]；

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 $y[n]$ 。

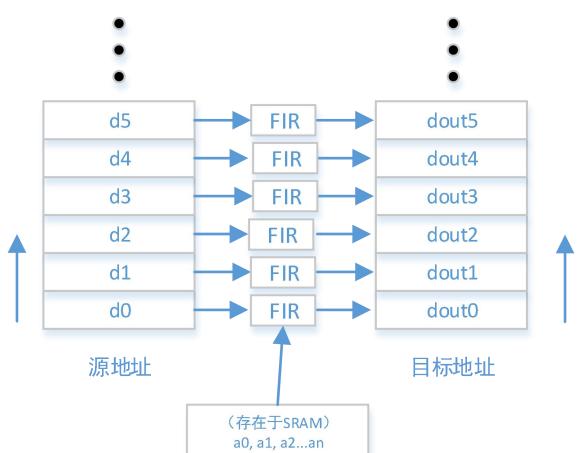
如果配置了中断使能信号(FFT_IE[11].IIR_DMA_IE=1)，则 IIR DMA 运算之后，将产生中断。标志位需要软件清除。

22.3.12



该模式是 FIR 滤波器计算，输入浮点数据 $x[n]...x[0]$ 和参数 $a[n]...a[0]$ ，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。





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。

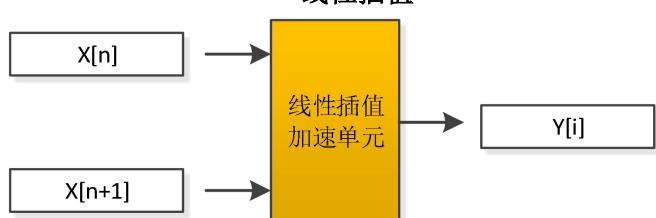
软件配置流程:

- 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。

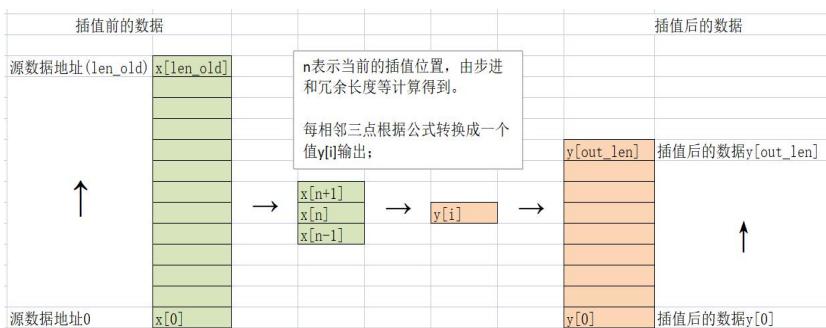
(不允许配置 dma len 小于 fir order, $dma\ len \geq fir\ order$)

22.3.13



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。

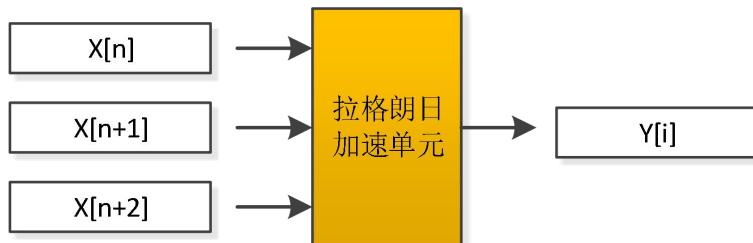
软件配置流程:

- 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。

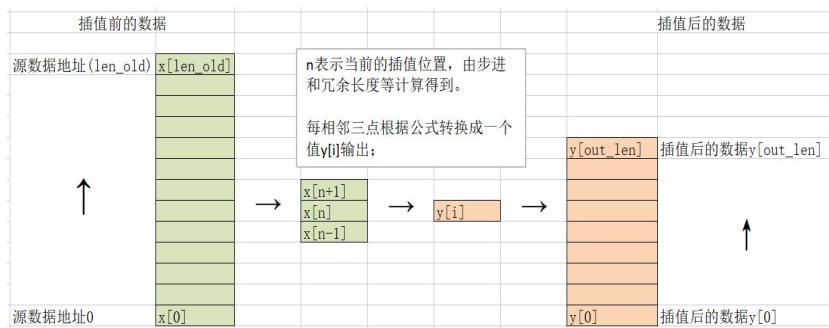
22.3.14

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_INTP), 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] (LAGRINTP_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。

22.4 实现说明

22.4.1 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。

22.4.2 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。

22.4.3 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 bufferfly 运算，完成一次 bufferfly 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) 16M 时钟	
	硬件表	SRAM 表	硬件表	SRAM 表
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，实际情况下，可能有其他模式抢占 SRAM，所以 FFT 时间会加长。

22.4.4 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。

22.4.4.1 向量模式

确认位宽：

- 首先，因为原始数据输入 x 和 y 的位宽是 32bit；
- 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。
- 根据 arctan 的放大倍数，决定迭代次数。
- 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
}
```

22.4.4.2 旋转模式

确认位宽：

- 模块输入为弧度，范围是 $-\pi$ 到 π 。假设输入弧度用 n 位数表示，所以输入弧度放大倍数为 $p0=(2^n)/2\pi$ 。
- 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。 $atan_t(n)=p0*atan(n)$ ，其他参数值按相同比例放大 p0 倍。
- 正常情况下， $atan_t(1)=p0*atan(n)=2^{n-3}$ ，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- π ~ π ;
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。

22.5 寄存器

22.5.1 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

22.5.2 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 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;	R/W	0

22.5.3

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;	R/W	0

		...		
		254: 254 阶; 255: 255 阶;		
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

22.5.4**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。

22.5.5 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

22.5.6 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

22.5.7 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

22.5.8 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

22.5.9 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

22.5.10 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

22.5.11 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

22.5.12 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

22.5.13 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

22.5.14 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

22.5.15 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

22.5.16 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

22.5.17 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

22.5.18 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

22.5.19 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

22.5.20 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

22.5.21 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)	R/W	0

	FFT 或 BTFY 模式下: DMA 旋转因子参数表虚部起始地址	
--	-----------------------------------	--

22.5.22 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

22.5.23 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

22.5.24 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

22.5.25 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

22.5.26

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

22.5.27 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 定义同“乘加单元 0 运算状态标志”	R	1
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

22.5.28 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

22.5.29 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

22.5.30 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

22.5.31 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

22.5.32 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

22.5.33 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

22.5.34 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 pi 寄存器值=radian*2^31/pi	R/W	0

22.5.35 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

22.5.36 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

22.5.37 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

22.5.38 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

22.5.39 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

22.5.40 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
------	----------	-----------------	-----	---

22.5.41

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

22.6 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。

22.6.1 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. 相角计算和諧波功率:

```
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. 计算諧波功率: Phn=FSA*Yun*Yin*PFn, 算 41 次諧波之后累加。其中 FSA 从外部获取。

22.6.2 操作流程:

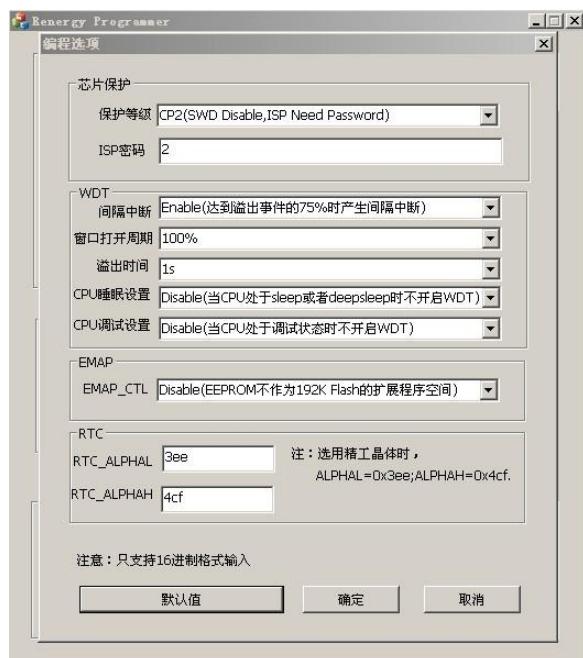
1. ADC 采样数据 sdata0 缓存到 SRAM, 每个点数据 24bit;
2. 使用 i2f_dma 模式把 SRAM 中的 sdata1 转换成浮点数格式并且做归一化得到 sdata2; 注意此时 SRAM 中每个 Word 地址, 只有第 3Byte 为有效数据, 所以需要配置 MAC_CTL0[28]=1, 不关心最高 Byte。
3. 使用 bitrev_mode 模式, 做 bitreverse 把 sdata2 变换成 sdata3;
4. 使用 FFT 计算模块, 把 sdata3 做 FFT 计算, 得到结果 (软件需要提前把 FFT 计算表格写到 SRAM);

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；

23 选项字节

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》）。



23.1 芯片保护设置

选项字节提供的芯片保护功能可以保护内置的Flash，用户可以通关设置保护等级和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。）

23.2 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：

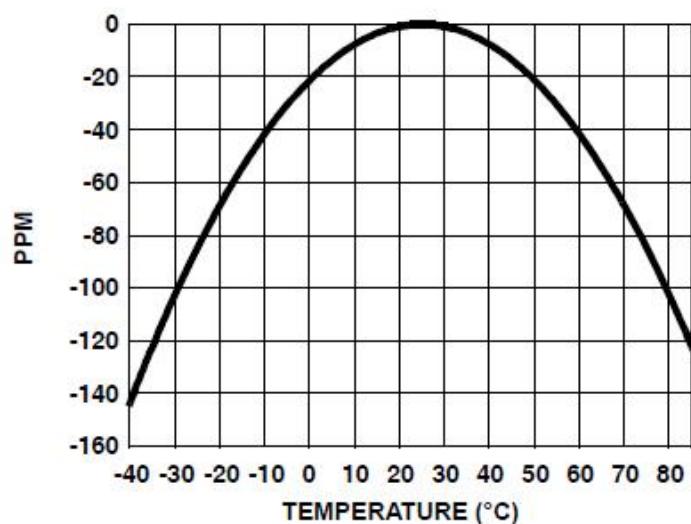
名称	描述	厂家默认值
间隔中断	0: Disable (不使能间隔中断) 1: Enable (达到溢出事件的 75% 时产生间隔中断)	0
窗口打开周期	0: 25% 1: 50% 2: 75% 3: 100% 在窗口打开期间将 0xBB 写入 WDTE 寄存器，看门狗清零并重新计数； 在窗口关闭期间将 0xBB 写入 WDTE 寄存器，会产生内部复位信号。	3
溢出时间	0: 16ms 1: 32ms 2: 128ms 3: 512ms 4: 1s 5: 2s 6: 4s 7: 8s	4
CPU 睡眠设置	0: Disable (当 CPU 处于 sleep 或者 deepsleep 的时候不开启 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)。	0

23.3 RTC 设置

RN8XXX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°C~70°C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°C~85°C) 与低温段 (-25°C~70°C) 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round($\alpha \times 32768$)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。



24 编程支持

芯片支持对内置 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-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。

24.1 概述

芯片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;

24.2 FLASH 保护机制

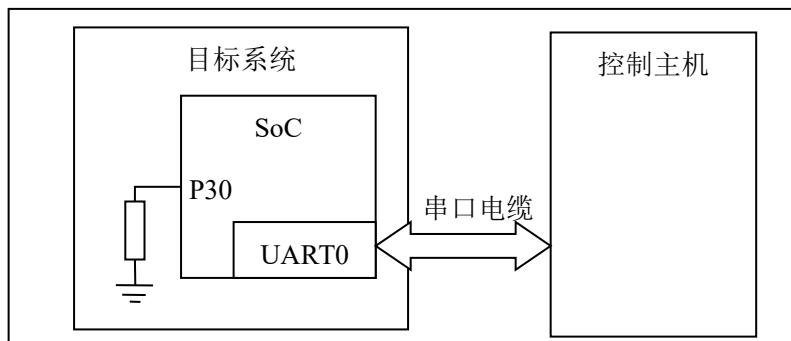
Flash 保护是允许用户在系统中通过使能不同的安全级别来限制对片内 Flash 的访问和 ISP 使用的一种机制。保护机制包含如下保护等级。用户可以通过“选项字节”设置芯片的保护等级。

表22-1 芯片保护等级

保护等级	名称	描述
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降为 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。)

24.3 在系统编程 (ISP)

用户可以将 P30 信号拉低, 并复位芯片, 让芯片进入 ISP 模式。ISP 模式的连接图如图 18-1 所示
图 22-1 ISP 硬件配置图



ISP的主要流程:

1. 按连接图配置并连接目标系统和控制主机;
2. 复位目标系统;
3. 控制主机配置串口为 1 个起始位, 8 个数据位, 1 个停止位, 波特率不大于 115200bps;
4. 控制主机发送“e”;
5. 目标系统响应“Synchnonized/r/n”;
6. 控制主机发送“Synchnonized/r/n”;
7. 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)
8. 控制主机可以根据需要执行对应的 ISP 命令;

24.3.1 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))。

当芯片接收完一个完整的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。

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 22-2 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 22-3 ISP 数据校验格式

行首	1	2	3	4	5	行尾
0x7e	0xff	S0	S1	S2	S3	0x7e

24.3.2 使用的资源

- 使用的 RAM 资源

ISP 使用片内 0x1001_7000 到 0x1001_8000 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。

24.3.3 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:

- 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
- UN 命令在 CP0, CP1, CP2 保护等级 (保护密码不提供) 下, 都能访问
- FC, AL 命令在任何情况下都能访问

表 22-4 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 (包括 flash, sram)	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	
flash 块查空	FQ <块地址>	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>	
获取保护等级	AL	
使能 PFPM	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22-5 ISP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 22-6 ISP 命令

命令	RD <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	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22-7 ISP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	“WM 268436224 4 0” “78” “56” “34” “12” 采用串口向0x10000300地址写入0x12345678

- 读内存

表22-8 ISP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22- 9 ISP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22- 10 ISP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22- 11 ISP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22- 12 ISP命令

命令	FQ <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	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22- 13 ISP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 页其余内容填充为 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 0 268436224 128”将SRAM地址0x10000300开始的128字节复制到FLASH地址0

- 内存比较

表22- 14 ISP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 268436224268436224 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22- 15 ISP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22- 15 ISP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22- 16 ISP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

- 软件复位

表22- 17ISP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22- 18 ISP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	该命令引导Flash代码
范例	“NV 0”将使能Flash写, 编程操作。

24.3.4 ISP 返回代码

表 22- 19 ISP 命令

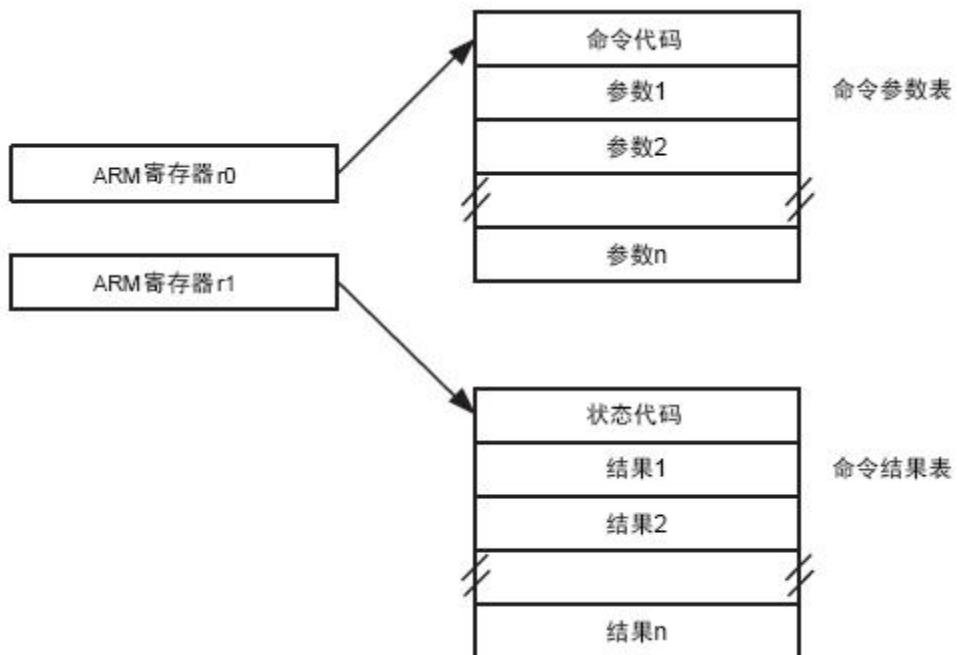
返回代码 (ASCII 码)	符号	描述
0	CMD_SUCCESS	成功执行命令。只有成功执行命令后, ISP处理器才发送该代码
1	INVALID_COMMAND	无效命令
2	INVALID_PARAM	无效参数 (参数的 ASCII 码不为 0-9)
3	INVALID_BAUD_RATE	无效波特率
4	INVALID_STOP_BIT	无效停止位
5	ADDR_NOT_ALIGN	地址不是以字为边界
6	COUNT_ERROR	字节计数值不是 4 的倍数
7	ADDR_NOT_MAPPED	所访问的地址空间越界
8	INVALID_SECTOR/INVALID_PAGE	无效 SECTOR_NUM 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

24.4 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21-7。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 22-2 IAP 参数传递



24.4.1 IAP 命令

IAP 命令	命令代码	描述
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

24.4.2 IAP 使用

IAP的应用主要在以下方面：

- ◆ 在线升级（更新FLASH）

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：

用户需要实现在线升级时，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，并使用 SoC 提供的 IAP 接口，将这些程序写入到 SoC 内部 FLASH 中。

24.5 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《锐能微编程平台使用说明》。

25 封装尺寸及焊接条件

25.1 封装尺寸

25.1.1 LQFP128L

LQFP128L (1414×1.4)		14.00×14.00×1.40	e=0.40
------------------------	---	------------------	--------

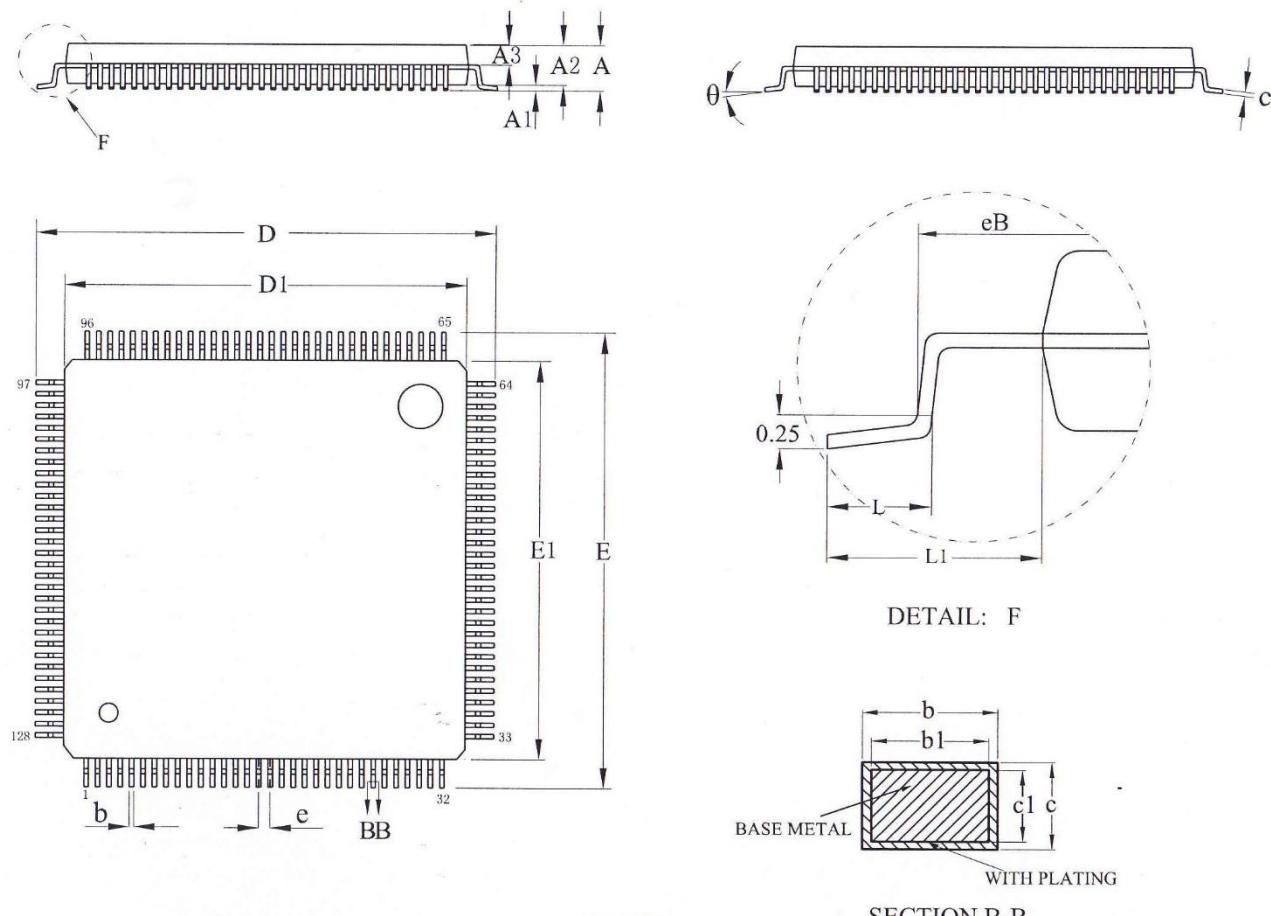


图 22-1 LQFP128L 封装尺寸图

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	---	---	1.60
A1	0.05	---	0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.14	---	0.22
b1	0.13	0.16	0.19
c	0.13	---	0.17

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°

25.1.2 LQFP100L

LQFP100L (1414×1.4)		14.00×14.00×1.40	e=0.50
------------------------	---	------------------	--------

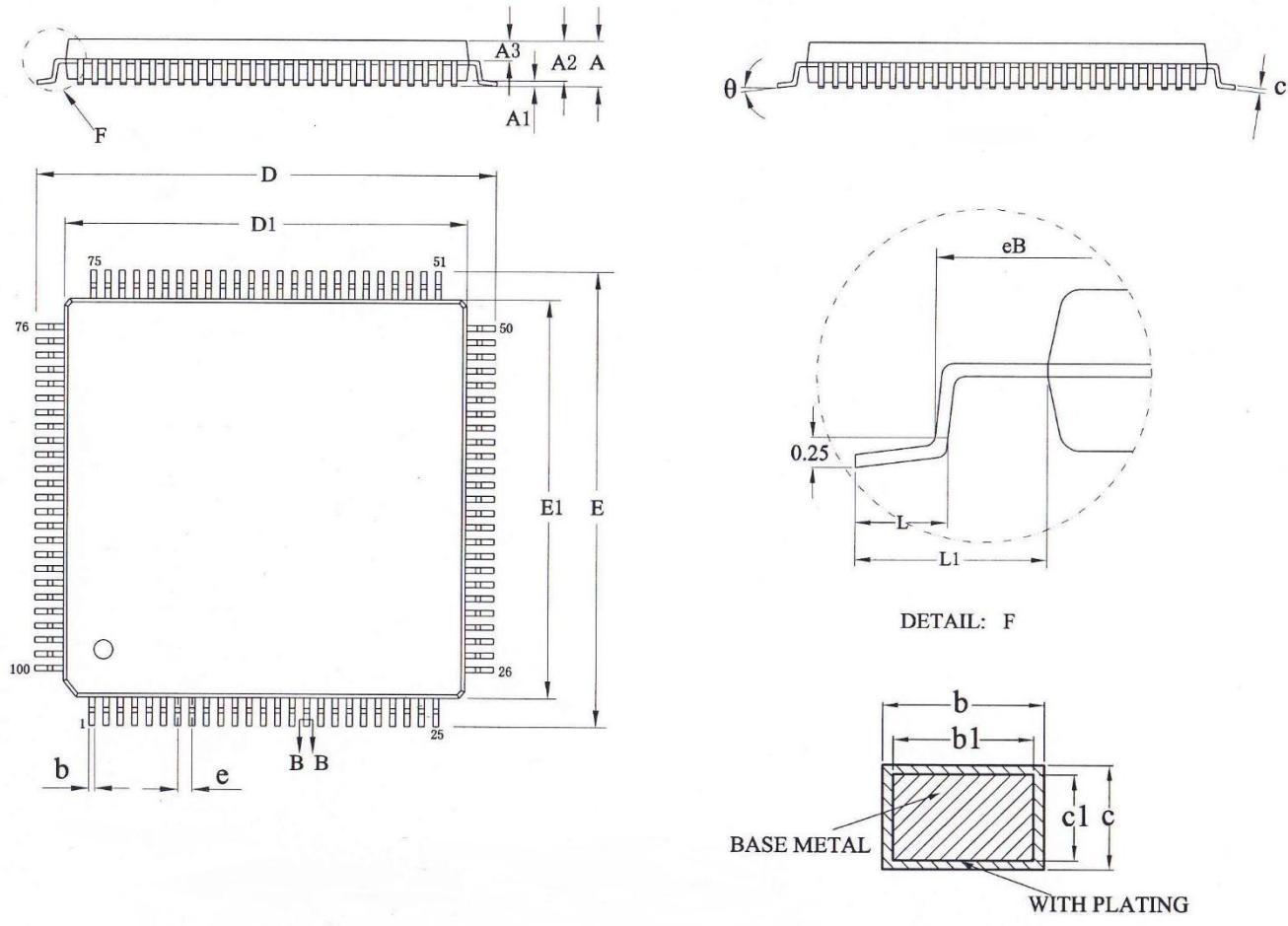


图 22-2 LQFP100L 封装尺寸图

SYMBOL	MILLIMETER		
	MIN	NOM	MAX

A	---	---	1.60
A1	0.05	---	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	---	0.26
b1	0.17	0.20	0.23
c	0.13	---	0.17
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°

25.1.3 LQFP64L

LQFP64L (0707×1.4)		7.00×7.00×1.40	e=0.40
-----------------------	---	----------------	--------

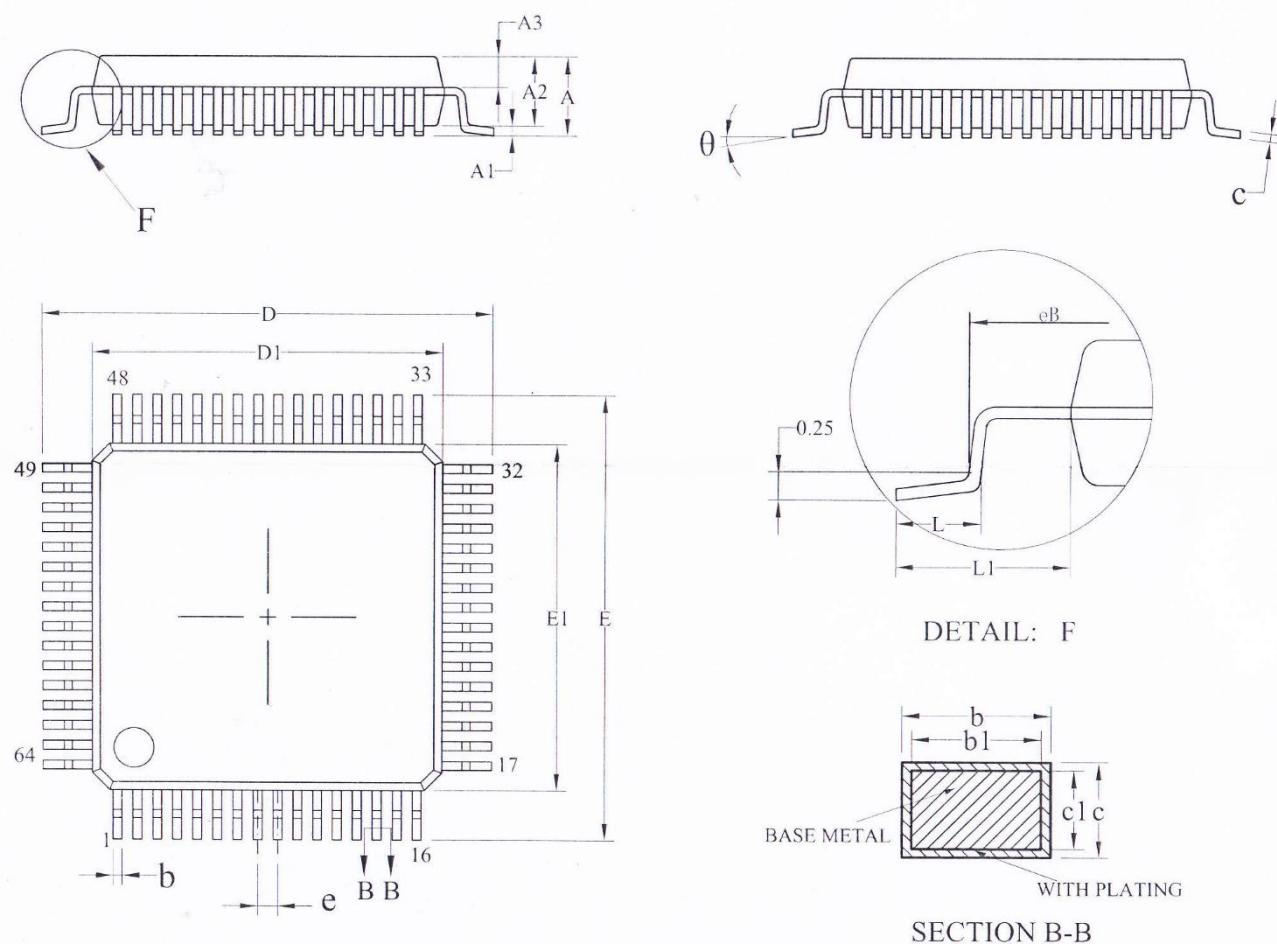
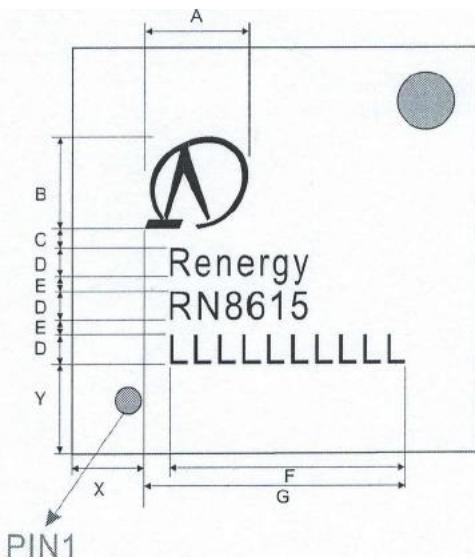


图 22-3 LQFP64L 封装尺寸图

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	---	---	1.60
A1	0.05	---	0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.16	---	0.24
b1	0.15	0.18	0.21
c	0.13	---	0.17
c1	0.12	0.13	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°

产品外观图如下，以 RN8615 V2 版为例，其他产品类似：



第一行为锐能微 Logo；

第二行 Renergy 为锐能微英文简称；

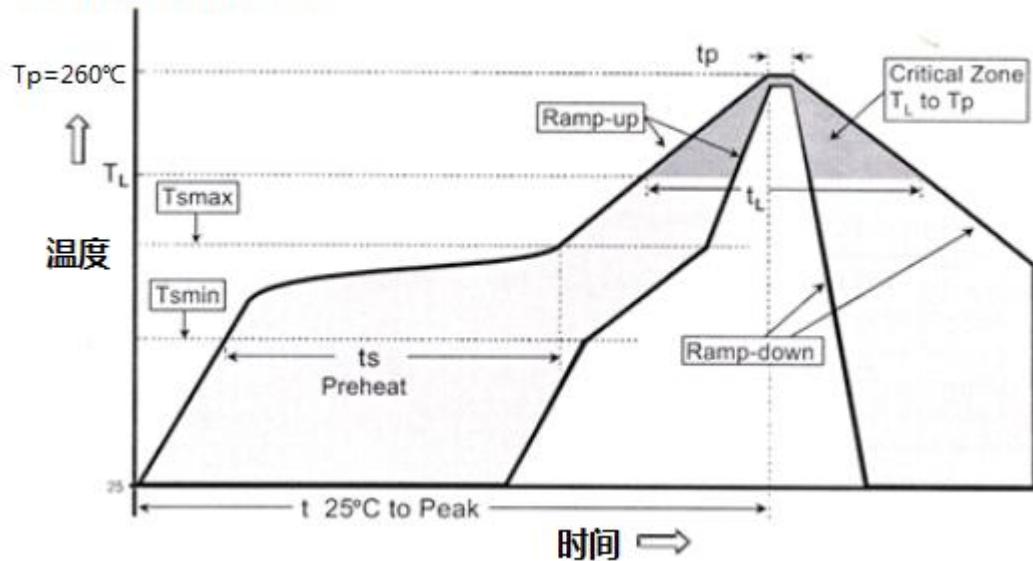
第三行 RN8xxx 为锐能微产品型号；

第四行为产品批号

左下脚的小点为 PIN1 标志。

25.2 回流焊炉的温度设定条件

回流焊炉温度设定条件

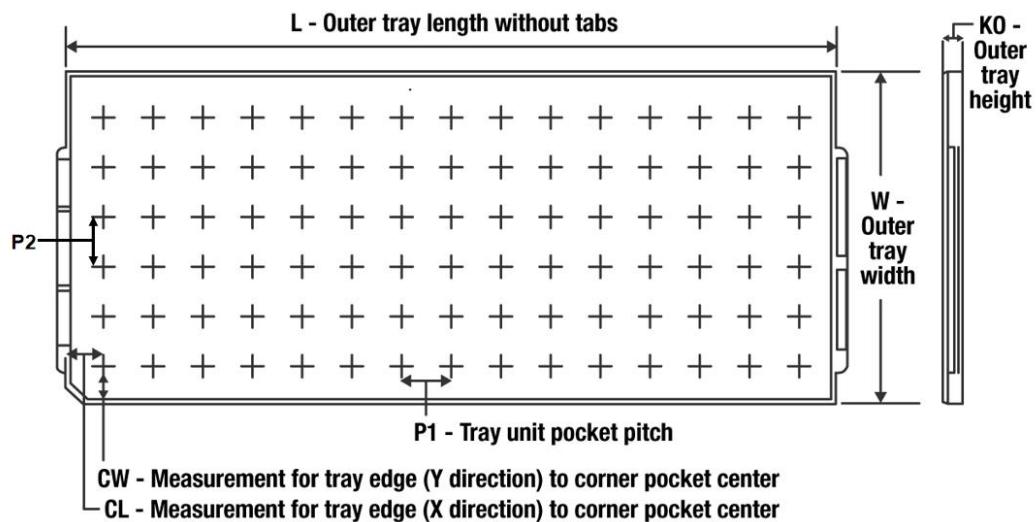


回流焊炉的温度设定曲线

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_s min)	150°C
最大温度 (T_s max)	200°C
时间 (最小-最大) (ts)	60-180 秒
T_s max - T_L 倾斜上升率(T_s max to T_L)	最大 3°C/秒

26 包装信息

26.1 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(0 707*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