

技 术 文 件

技术文件名称：单相计量芯 SOC RN2025 (B64/C64) 用户手册

版 本： V 1.7

文件 等级：绝密，仅限战略合作客户

共 页

(包括封面)

拟 制 _____

审 核 _____

会 签 _____

标准化 _____

批 准 _____

深圳市锐能微科技有限公司

版本升级说明

➤ 芯片硬件资源改版点

RN2025A64 和 RN2025B64 的区别

功能点	RN2025A64	RN2025B64
谐波计量	软件处理，不能实现连续计算	新增谐波计算引擎，谐波功率由硬件计算，可以实现连续计算。
基波计量	支持	支持，优化 2 次谐波影响量；
谐波含量	软件计算	新增谐波计算引擎，谐波含量由硬件计算
负荷识别波形输出点数	SPI 口依赖软件搬运，CPU 开销过大	新增专用 DMA，支持硬件 DMA 搬运；
负荷识别波形组帧	波形数据、帧头、帧尾以及校验和完全由软件来处理	硬件上增加灵活的帧格式，自动计算校验和，降低了软件复杂度
负荷识别波形补偿	无补偿	新增支持增益补偿、相位补偿、谐波补偿
闪变	不支持	支持硬件计算
半波有效值	得到的是半波有效值的平方	新增开根号单元，可直接提供半波有效值
SPI 口	3 个，不支持 DMA	5 个，均支持 DMA；
UART 口	都不支持 DMA	UART2 支持 DMA
计量误差自检测	电压通道需要有 K 系数校正，影响精度；	对电压通道作出优化，增大 ADC 内阻，新增 buffer，取消 K 系数
电压通道外围电路	支持两种典型电路	仅支持一种典型电路，详见手册
内部高频 RC	无	新增 32.768Mhz 内部高频 RC，提升系统可靠性
PLL	最高倍频至 16.384Mhz	最高倍频至 32.768Mhz
ADC	多通道 10bit SAR-ADC (AIN 端口)	多通道 10bit SAR-ADC 新增 0.25 倍 PGA (AIN 端口) 新增 16bit GP-ADC (SDAIN 端口)
RTC	支持二次曲线补偿； 校正刻度为 0.061035ppm	新增支持四次曲线补偿； 校正刻度为 0.0305175ppm
基准电压	不支持外接基准	支持外接基准
校表相关	通道相位校正刻度 0.01757 度； 通道相位校正范围 ±2.24896 度； 无校验寄存器；	通道相位校正刻度提升至 0.00879 度； 通道相位校正范围提升至 ±4.5 度； 新增计量校表数据校验寄存器；
IO 口	34 个	37 个，新增 3 个；并新增 IO 口复用关系。

RN2025B64 和 RN2025C64 的区别：

功能点	RN2025B64	RN2025C64
有效值、功率更新时间	250ms	64ms
GPADC	有 R2Rbuf, ADC 内阻较大 (百兆 Ω 级别)； 支持 Vref/2 作为测量基准；	无 R2Rbuf, ADC 内阻较小 (百 k 级或者 M 级)，不同的 PGA 增益和采样时钟频率下内阻不同；

		不支持 Vrf/2 作为测量基准;
--	--	-------------------

硬件设计差异

功能点	RN2025A64	RN2025B64	RN2025C64
电压采样电路	芯片外围电阻串分压, 推荐的阻值分别是 1MΩ 左右 /1KΩ/1KΩ	芯片外围电阻串分压, 推荐的阻值分别是 1.5MΩ 左右 /1KΩ/3kΩ	芯片外围电阻串分压, 推荐的阻值分别是 1.5MΩ 左右 /1KΩ/3KΩ
RP 管脚接地电阻	推荐阻值 3KΩ	推荐阻值 6.2 KΩ	推荐阻值 6.2 KΩ
UASP_400 管脚接线	不用自检测功能时, 可悬空	不用自检测功能时, 可悬空	必须连接到电压采样的 UN 端, 即在不使用自检测功能时, 也必须按照有自检测功能的方式接线。

应用软件兼容性

功能点	RN2025A64	RN2025B64	RN2025C64
电压通道 ADC 计量	支持, 无需调用库函数	支持, 必须调用锐能微库函数文件 <sysctl.c>中的 WriteUEdtVBCtlReg 配置, 具体见《锐能微 RN202X_RN7326_应用笔记 002-库函数使用说明 v1.2》	同 RN2025B64
过零检测	支持	支持	支持, 应用时必须将 ZXOT、ZXOTU 寄存器配置为 0

关于库函数

从 RN2025A64 切换到 RN2025B64, 需要替换库函数:

从 RN2025B64 切换到 RN2025C64, 也必须更新库函数。

功能点	RN2025A64 版本	RN2025B64 版本	RN2025C64 版本
库函数	提供的是封装后的 lib 文件 (keil) 或者 .a 文件 (iar)	提供的是源代码 1、B 版新增对 32.768Mhz 的 Pll 和 RCH 的支持 2、增加了配置电压通道 ADC 的函数; 3、增加了 spl 的容错功能;	增加对 RN2025C64 的支持

关于参考例程

从 RN2025A64 切换到 RN2025(B64/C64), 需要替换参考例程

除了库函数, 锐能微还提供了部分参考例程, 见《Demo 程序》文件夹, 若用户有使用该参考例程, 或者在该例程上修改, 也需对应替换或者修改。以下为部分改动较大的例程:

计量误差自检测例程	<RN_Lib/EDT>	<Edt.c><Edt.h>	修改	1、误差自检电压通道做了优化, 取消了 K 系数;
-----------	--------------	----------------	----	---------------------------

				2、误差自检电压通道修改为 Vbias 模式;
谐波含量 (参考例程)	<RN_Lib/Harmonic.c>	<harmonic.c> <Harmonic.h>	优化	1、硬件新增了谐波计算引擎, 谐波含量改由硬件计算。
		<InterHarmonic.c> <InterH_Const.h> <InterHarmonic.h>	新增	1、新增了间谐波例程
闪变(参考例程)	<RN_Lib/Flick>	<Flick.c><Flick.h>	新增	1、B 版新增硬件支持
GPADC 模块(参考例程)	<Devices/Src>	<Gpadc.c><Gpadc.h>	新增	1、B 版新增功能
GPIO 模块(参考例程)	<Devices/Src>	<GPIODrive.c><GPIODrive.h>	修改	1、B 版新增功能

修改记录

版本号	发布日期	更改人	主要更改内容
V1.0	2020-3-10	客服部	首次发布。 该文档是 RN2025 升级版的用户手册，版本代号为 B64；原有版本代号为 A64，以便于做区分。 两个版本的型号不做更改。
V1.1	2020-6-3	客服部	修改以下问题： 1、Sar ADC 新增的 PGA 实际为 0.25 倍，手册写为 0.2 倍； 2、通用 ADC 的 GAIN_GPADC 寄存器，当配置为“1xx”时，增益实际为 0.2 倍，手册误写为“0.25 倍”。 3、通用 ADC 的 GAIN_GPADC 寄存器，当配置为“1xx”时，增益实际为 0.2 倍，手册误写为“0.25 倍”。 4、将 EMUCFG 的 bit15 名称，由“SAG_INTCFG”和“SAGINT_CFG”，统一成“SAGINT_CFG”。 5、修改 SPI0_DMA_CTRL 的 bit0~bit6 寄存器描述错误。 6、DMA_WAVE_CH_SEL 的 CH0_SEL 描述，“仅在模式 1 下有效”，修改为“仅在模式 1、模式 2 下有效”； 7、修改了“RTC_IOMODE”的描述，修改点为 RTC_IOMODE 的时间基准设置为 2ms 时，配置 bit8:4 为 0，滤波时间由原描述 4ms，修改为实测值 23ms；时间基准设置为 250ms，配置 bit8:4 为 0~3，实际滤波都是 500ms 左右，原公式（时间设置寄存器+1）*250ms 有误。 8、开放了 UADC_SEL 寄存器； 9、将通用 ADC 的管脚名称统一为了“SDAIN”； 10、通用 ADC 模块，增加描述“使用通用 ADC 模块，要求系统主频最低为 8.192Mhz，不支持更低频率下的应用。” 11、修改 DMAWAV_CH_SE 寄存器说明，对不需要做波形缓存的通道，“DMAWAV_CH_SEL 中通道设置要求配置为 7”，原有描述“仅在 DMA_WAVE_CH_SEL 选择的通道总数大于 n 有效。”有误。 12、删除了 CFCFG 的“CFSELAB”位寄存器描述，芯片不支持该功能。
V1.2	2020-8-17	客服部	修改以下问题： 1、更新了推荐原理图，修改地方：RP 管脚对地电阻由 3KΩ 修改为 6.2KΩ； 2、修改了不用误差自检功能的两处错误描述：将 UASP_400 管脚描述内容，由“不用误差自检功能，要接地”，修改为“要悬空或者按带误差自检功能设计”；删除了 RP 管脚“不使用误差自检功能，该引脚接地”的错误描述；增加要调用 WriteUEdtVBCtlReg 函数的说明。 3、增加芯片从 A 版到 B 版的“升级说明”，对芯片硬件资源、硬件设计、软件兼容性、库函数及例程升级都做了说明。

			4、修改了 Checksum 校验和寄存器描述错误，芯片实际为“0x1cc~0x1c8 不参与校验和计算，0x1b8, 0x600~0664 参与校验和计算”
V1.3	2020-9-28	客服部	1、修改了引脚说明里，关于 P21 管脚的复用描述，应该是 TC0_P[0]功能，写成了 TC1_P[1]。 2、修改了“4.5.3 有效值寄存器”中有效值更新时间的描述，由“64ms”，修改为 250ms； 3、修改了“3.2.1 中断向量表”中“FLK”和“谐波 FFT”位置的的反的错误。 4、修改了“hosc 管脚 P16/P17 实际具备输入输出功能，有些地方描述为只读”的问题。 5、修改了 GPADC_CTL1 的寄存器描述，将推荐基准由 Vrep_op6 修改为 Vref。 6、增加了 GPADC 测量范围的说明。 7、“20.3.2.3” Sram 的 Data[30:28]用于控制校验和计算，而不是 Data[31:28]。
V1.4	2021-8-31	系统部	1、修改版本升级说明。 2、修改概述。 3、管脚排列，HOSC 引脚标注 32.768MHZ 晶振使用说明，将 RP 引脚外接电阻有 3K 修改为 6.2K。 4、EMU 章节，增加过零检测 ZXOT、ZXOTU 寄存器使用说明，增加 SAGCFG、UADC_SEL、OI2、OV2 寄存器配置注意事项，修改功率和有效值更新时间 C 版为 64ms，修改超链接描述错误。 5、模拟外设章节，修改 LVD_CTL 寄存器阈值描述，删除 LVD DOWN 功能，以及 SYS_PD 相关寄存器描述。 6、RTC 章节，增加 KEY 功能使用注意事项。 7、NVM 章节，增加校正公式描述，以及稳定时间配置注意事项。 8、EDT 寄存器描述修订。 9、增加软件复位调用函数说明。 10、其他一些描述错误。
V1.5	2021-12-14	客服部	1、增加了 C 版相对 B 版，GPADC 内阻更小的描述。 2、增加了 B 版切换到 C 版，需要更新库函数的描述。 3、修改了文档名称，把文档名称中的“RN2025”修改为“RN2025B64_RN2025C64”
V1.6	2022-01-18	客服部	增加 C 版 UASP_400 管脚，在不用自检测功能时的处理方式差异说明
V1.7	2022-02-10	客服部	1、4.3.13/4.5.38 统一地址为 0x278 电压过压幅值阈值配置寄存器名称为 OV2_LVL 及 0x27C 及电压过压事件阈值配置寄存器名称为 OV2_CNTTH，及有效值位数说明为[15:0] 2、4.5.37.2 OIx_CNT 增加速率为 16Khz 3、4.5.26/4.3.10 增加电流过零依赖电压过零的说明

			<p>4、4.5.15 修订直流 DCOS_IA 与 IA 通路有效值关系式为 $IA' = \sqrt{(IA^2 + ((DCOS_IA - IAd0) * 2^{12}) / 2^{27})^2}$及相关说明。</p> <p>5、4.5.34 增加直流 OFFSET 自动校正结果 DCOSx 寄存器用于 EMU 模式和全失压有效值直流 OFFSET 校正的方法（EMU 模式时直接使用，全失压有效值时需除 0.512）。</p> <p>6、4.5.41 增加 LS_DCOS 值获取说明</p> <p>7、5 全失压章节修订</p> <p>8、2.7.10 增加闪变模块时钟门控说明</p> <p>9、4.4.2 计量配置和状态寄存器中自定义电能寄存器的操作模式改为只读 R</p> <p>10、单相计量单元章节中超链接编号调整</p> <p>11、修订 ISP 章节内容</p>
--	--	--	--

目录

(包括封面).....	1
版本升级说明	2
修改记录	5
1 概述.....	19
1.1 简介.....	19
1.2 芯片特性.....	19
1.3 系统框图.....	22
1.4 管脚排列.....	22
1.5 电气特性.....	30
1.6 应用推荐电路.....	33
2 系统控制.....	36
2.1 电源域划分.....	36
2.2 掉电处理.....	36
2.3 时钟源.....	37
2.4 时钟切换.....	38
2.5 MCU 的低功耗模式.....	40
2.6 复位.....	41
2.6.1 外部 PIN 复位	41
2.6.2 上下电复位.....	41
2.6.3 软件复位.....	41
2.6.4 看门狗复位.....	41
2.7 寄存器.....	41
2.7.1 OSC_CTL1 (0x00).....	42
2.7.2 SYS_MODE (0x04)	43
2.7.3 SYS_PD (0x08).....	43
2.7.4 ADC_CTL (0x0C).....	45
2.7.5 CTT_CTL (0x18)	45
2.7.6 OSC_CTL2(0x1C).....	46
2.7.7 SYS_RST(0x20)	47
2.7.8 SYS_MAPCTL(0x24).....	49
2.7.9 MOD0_EN(0x28)	49
2.7.10 MOD1_EN(0x2C).....	51
2.7.11 INTC_EN(0x30)	52
2.7.12 KBI_EN(0x34).....	52
2.7.13 SYS_PS(0x3C).....	53
2.7.14 IRFR_CTL (0x40).....	54
2.7.15 TRIM_CFG1(0xA0)	54
2.7.16 TRIM_START(0xA4).....	55

2.7.17	DMA_PRI 0xC0.....	55
2.7.18	DMA_RST 0xC4	56
3	处理器架构.....	58
3.1	概述.....	58
3.2	CORTEX-M0 处理器.....	58
3.2.1	中断配置.....	59
3.3	MCU 存储映射.....	60
3.3.1	SRAM	62
3.3.2	FLASH	62
3.4	中断应用.....	63
4	单相计量单元.....	64
4.1	特点.....	64
4.2	系统框图.....	65
4.3	功能说明.....	66
4.3.1	采样通道.....	66
4.3.2	ADC 波形缓存及 DMA	66
4.3.3	有功功率.....	73
4.3.4	无功功率.....	74
4.3.5	视在功率.....	74
4.3.6	电能输出.....	74
4.3.7	启动潜动.....	75
4.3.8	电压电流有效值.....	77
4.3.9	半波有效值.....	77
4.3.10	过零检测.....	78
4.3.11	电压线频率.....	78
4.3.12	电压暂降检测.....	78
4.3.13	过压检测和电压半波峰值.....	78
4.3.14	过流检测.....	79
4.4	寄存器列表.....	79
4.4.1	计量参数寄存器列表.....	79
4.4.2	计量配置和状态寄存器列表.....	81
4.5	寄存器说明.....	87
4.5.1	EMU 写使能寄存器 SPCMD (0x2FC).....	87
4.5.2	波形采样寄存器.....	87
4.5.3	有效值寄存器.....	87
4.5.4	功率寄存器.....	88
4.5.5	快速脉冲计数寄存器.....	89
4.5.6	电能寄存器.....	89
4.5.7	电压线频率寄存器 UFreq (0x450).....	89
4.5.8	半波有效值寄存器(0x574~0x584).....	90
4.5.9	高频脉冲常数寄存器 HFCONST(0x00~0x04).....	90
4.5.10	启动阈值寄存器 Start(0x08~0x0C).....	91

4.5.11	过零阈值寄存器 ZXOT(0x14).....	92
4.5.12	相位补偿区域设置寄存器 PRTHx(0x18~0x24)	93
4.5.13	通道相位校正寄存器 PHSU/I	93
4.5.14	通道增益寄存器 GSU/I	95
4.5.15	通道直流 OFFSET 校正寄存器 DCOSx (修改)	95
4.5.16	有效值 OFFSET 校正寄存器	96
4.5.17	功率增益寄存器.....	96
4.5.18	功率相位校正寄存器.....	97
4.5.19	功率 OFFSET 校正寄存器	98
4.5.20	电压暂降阈值寄存器 SAGCFG(0x160).....	99
4.5.21	过压阈值寄存器 OVLVL(0x164)	99
4.5.22	过流阈值寄存器 OILVL(0x168).....	100
4.5.23	CF 脉冲配置寄存器 CFCFG(0x180).....	100
4.5.24	计量单元配置寄存器 EMUCFG(0x188).....	102
4.5.25	计量控制寄存器 EMUCON(0x18C)	104
4.5.26	EMU 中断寄存器.....	105
4.5.27	功率方向寄存器 PQSign(0x198).....	107
4.5.28	潜动状态标志寄存器 Noload(0x19C).....	107
4.5.29	相电压电流状态寄存器 PHASES(0x1A4).....	108
4.5.30	校验和寄存器 EMU_CKM (0x1A8)	109
4.5.31	读校验寄存器 Rdata(0x1AC)	110
4.5.32	写校验寄存器 WData(0x1B0)	110
4.5.33	电压过零阈值寄存器 ZXOTU(0x1B4).....	110
4.5.34	直流 OFFSET 自动校正使能寄存器 AUTODC_EN(0x1B8)	111
4.5.35	过零计算配置及标志寄存器 ZXOTCFG(0x1BC).....	111
4.5.36	自定义功率/HFCONST3/D2F/能量寄存器	112
4.5.37	短路停电阈值寄存器.....	115
4.5.38	过压和电压峰值寄存器.....	116
4.5.39	计量单元配置寄存器 2 EMU_CFG2(0x270) (B0 修改)	116
4.5.40	EMU 中断 2 寄存器.....	117
4.5.41	DMA 波形缓存配置寄存器.....	119
4.5.42	同步采样通道校正和配置寄存器 (新增)	127
5	全失压测量.....	129
5.1	主要特点.....	129
5.2	工作过程.....	130
5.3	全失压测量时直流 OFFSET 校正过程	131
5.4	寄存器.....	131
5.4.1	寄存器列表.....	131
5.4.2	NVM 中断	132
5.4.3	LS_CFG(0x08).....	133
5.4.4	LS_DCOS_Ix(0x0C~0x10).....	133
5.4.5	LS_THOx(0x18~0x1C)	133
5.4.6	RMS_Lx(0x24~0x34)	134

5.4.7	NVM_PSW(0x3C).....	135
6	谐波计算引擎.....	136
6.1	特点.....	136
6.2	计算引擎基本原理.....	136
6.2.1	单精度浮点数表示.....	136
6.2.2	特殊数值.....	136
6.2.3	浮点数舍入处理.....	137
6.2.4	IEEE754 标准 rounding 模式.....	137
6.2.5	本芯片的 rounding 模式.....	137
6.2.6	整数转浮点数原理.....	137
6.2.7	浮点数转整数原理.....	138
6.2.8	浮点数乘法原理.....	138
6.2.9	浮点数加法原理.....	138
6.2.10	蝶形运算原理.....	138
6.3	运算指令.....	139
6.3.1	整数转浮点数 (int2fp/int2fp_dma)	139
6.3.2	浮点数转整数 (fp2int/fp2int_dma)	140
6.3.3	浮点数乘法 (fp_mult)	141
6.3.4	浮点数加法 (fp_add)	141
6.3.5	浮点数减法 (fp_sub)	142
6.3.6	浮点数乘加运算 (fp_mladd)	142
6.3.7	浮点数蝶形运算 (单次) (btfy/btfy_dma)	143
6.3.8	正弦余弦计算 (sin_cos)	145
6.3.9	均方根和反正切 (fp_sqrt/ fp_atan)	146
6.3.10	浮点数除法器 (fp_div)	146
6.4	实现说明.....	146
6.4.1	整数转浮点数.....	146
6.4.2	浮点数转整数.....	146
6.4.3	FFT 说明.....	147
6.4.4	Cordic 说明.....	148
6.5	寄存器.....	149
6.5.1	寄存器列表.....	149
6.5.2	MAC_CTL0 (0x0)	150
6.5.3	MAC_CTL1 (0x04)	152
6.5.4	MAC_IN0 (0x08)	152
6.5.5	MAC_IN1 (0x0C)	152
6.5.6	MAC_IN2 (0x10)	152
6.5.7	MAC_IN3 (0x14)	152
6.5.8	MAC_IN4 (0x18)	153
6.5.9	MAC_IN5 (0x1C)	153
6.5.10	MAC_OUT0 (0x20)	153
6.5.11	MAC_OUT1 (0x24)	153
6.5.12	MAC_OUT2 (0x28)	153

6.5.13	MAC_OUT3 (0x2C)	153
6.5.14	DIV_IN0 (0x30)	153
6.5.15	DIV_IN1 (0x34)	154
6.5.16	DIV_OUT0 (0x38)	154
6.5.17	DMA_SRBADR (0x3C)	154
6.5.18	DMA_SIBADR (0x40)	154
6.5.19	DMA_PRBADR (0x44)	154
6.5.20	DMA_PIBADR (0x48)	154
6.5.21	DMA_TRBADR (0x4C)	155
6.5.22	DMA_TIBADR (0x50)	155
6.5.23	DMA_LEN (0x54)	155
6.5.24	FFT_IE (0x58)	155
6.5.25	FFT_FLG (0x5C)	156
6.5.26	ALU_STA0 (0x60)	156
6.5.27	ALU_STA1 (0x64)	157
6.5.28	CRD_CTL (0x68)	157
6.5.29	CRD_XIN (0x6C)	157
6.5.30	CRD_YIN (0x70)	157
6.5.31	CRD_AMP (0x74)	157
6.5.32	CRD_PHASE (0x78)	158
6.5.33	CRD_ANGLE (0x7C)	158
6.5.34	CRD_COSINE (0x80)	158
6.5.35	CRD_SINE (0x84)	158
6.5.36	CRD_IE (0x88)	158
6.5.37	CRD_FLG (0x8C)	158
6.6	软件操作流程.....	159
6.6.1	完整方案说明:	159
6.6.2	操作流程:	159
7	计量自检测功能.....	161
7.1	功能描述.....	161
7.1.1	电流通道精度自检测功能.....	161
7.1.2	电压通道精度自检测功能.....	161
7.2	寄存器.....	162
8	RTC	163
8.1	概述.....	163
8.2	特点.....	163
8.3	按键事件记录功能说明.....	164
8.4	I2CM 功能说明	164
8.5	寄存器.....	165
8.5.1	RTC_CTL(0x00).....	170
8.5.2	RTC_SC(0x04).....	171
8.5.3	RTC_MN(0x08).....	172

8.5.4	RTC_HR(0x0C)	172
8.5.5	RTC_DT(0x10)	172
8.5.6	RTC_MO(0x14)	172
8.5.7	RTC_YR(0x18)	172
8.5.8	RTC_DW(0x1C)	173
8.5.9	RTC_CNT1(0x20).....	173
8.5.10	RTC_CNT2(0x24).....	173
8.5.11	RTC_SCA(0x28).....	174
8.5.12	RTC_MNA(0x2C).....	174
8.5.13	RTC_HRA(0x30)	174
8.5.14	RTC_IE(0x34).....	174
8.5.15	RTC_IF(0x38)	175
8.5.16	RTC_TEMP(0x3C)	176
8.5.17	RTC_DOTA0 (0x48)	177
8.5.18	LOSC_CFG1(0x6C)	177
8.5.19	RTC_TEMPOS(0xC4)	177
8.5.20	RTC_TPSIN(0xC8).....	178
8.5.21	I2CMEEP_CTL (0x180)	178
8.5.22	I2CMEEP_CPASS (0x184)	179
8.5.23	I2CMEEP_CBYTE (0x188)	179
8.5.24	I2CMEEP_ABYTE (0x18C)	180
8.5.25	KEYOUT 功能说明	180
8.6	RTC 时钟读写步骤	180
8.7	RTC 校准步骤	180
8.8	RTC 定时器操作步骤	180
9	WDT	182
9.1	主要特点.....	182
9.2	看门狗定时器的配置.....	182
9.3	寄存器.....	183
9.3.1	看门狗使能寄存器 WDT_EN(0x00).....	183
10	定时器.....	184
10.1	特性.....	184
10.2	功能框图.....	185
10.3	寄存器.....	185
10.3.1	当前计数值寄存器 TC_CNT(0x00)	186
10.3.2	预分频寄存器 TC_PS(0x04).....	186
10.3.3	目标计数值寄存器 TC_DN(0x0C).....	186
10.3.4	捕获比较通道 0 数据寄存器 TC_CCD0(0x14).....	187
10.3.5	捕获比较通道 1 数据寄存器 TC_CCD1(0x18).....	187
10.3.6	时钟配置寄存器 TC_CCFG(0x1C).....	187
10.3.7	控制寄存器 TC_CR(0x20).....	188
10.3.8	捕获比较通道 0 模式寄存器 TC_CM0(0x24).....	189

10.3.9	捕获比较通道 1 模式寄存器 TC_CM1(0x28).....	190
10.3.10	中断使能寄存器 TC_IE(0x2C).....	191
10.3.11	状态寄存器 TC_STA(0x30).....	191
10.4	典型应用.....	192
10.4.1	自动运行模式, 定时功能.....	192
10.4.2	输入捕获模式, 脉宽测量功能.....	192
10.4.3	比较输出模式, 方波输出功能.....	192
10.4.4	比较输出模式, PWM 输出功能.....	193
10.4.5	从模式, 外部清零和门控功能.....	195
11	模拟外设.....	196
11.1	特点.....	196
11.2	寄存器.....	196
11.2.1	SAR_CTL(0x00).....	197
11.2.2	SAR_START(0x04).....	198
11.2.3	SAR_STATUS(0x08).....	198
11.2.4	SAR_DAT(0x0C).....	199
11.2.5	LVD_CTL(0x10).....	199
11.2.6	LVD_STAT(0x14).....	199
11.2.7	SAR_CTL1 0x18.....	200
11.3	ADC 电压检测步骤.....	200
11.4	VBAT 电压检测.....	201
11.5	低电压检测应用.....	201
12	GPIO.....	203
12.1	概述.....	203
12.2	功能描述.....	203
12.3	应用注意事项.....	205
12.4	寄存器.....	205
12.4.1	寄存器列表.....	205
12.4.2	GPIO-AHB 寄存器.....	206
12.4.3	GPIO-APB 寄存器.....	221
12.5	GPIO 操作步骤.....	226
13	外部中断控制器.....	227
13.1	特性.....	227
13.2	寄存器.....	227
13.2.1	INTC_CTL(0x00).....	227
13.2.2	INTC_MODE(0x04).....	227
13.2.3	INTC_MASK(0x08).....	228
13.2.4	INTC_STA(0x0C).....	229
14	KBI.....	230
14.1	特性.....	230

14.2	寄存器.....	230
14.2.1	KBI_CTL(0x00).....	230
14.2.2	KBI_SEL(0x04).....	230
14.2.3	KBI_DATA(0x08).....	230
14.2.4	KBI_MASK(0x0C).....	231
14.3	KBI 操作步骤.....	231
15	通用 DMA 控制器 (SPI1/SPIS/SPI3/SPI4/UART2 增加)	232
15.1	特点.....	232
15.2	实现原理.....	232
15.3	模式选择.....	232
15.3.1	只使能 DMA 发送	232
15.3.2	只使能 DMA 接收	232
15.3.3	同时使能接收和发送.....	232
15.4	SRAM 发送 BUFFER.....	232
15.4.1	起始地址和长度配置.....	232
15.4.2	Buffer 中数据发送顺序	233
15.4.3	UART 接口数据位宽小于 1Byte.....	234
15.5	SRAM 接收 BUFFER.....	234
15.6	单次模式和循环模式.....	235
15.6.1	DMA 单次模式	235
15.6.2	DMA 循环模式	235
15.7	DMA 中断	235
15.7.1	DMA 发送半空中断	235
15.7.2	DMA 发送全空中断	236
15.7.3	DMA 接收半满中断	236
15.7.4	DMA 接收全满中断	236
15.7.5	DMA 接收错误中断	236
16	UART	237
16.1	概述.....	237
16.2	寄存器.....	237
16.2.1	寄存器列表.....	237
16.2.2	控制寄存器 UARTx_CTL (0x00).....	238
16.2.3	波特率配置寄存器 UARTx_BAUD (0x04)	239
16.2.4	状态指示寄存器 UARTx_STA (0x08).....	240
16.2.5	发送数据寄存器 UARTx_TXD (0xC).....	240
16.2.6	接收数据寄存器 UARTx_RXD (0x10)	240
16.2.7	波特率小数分频配置寄存器 UARTx_FDIV (0x14).....	241
16.2.8	UART2_DMA_CTL(新增, 0x18).....	241
16.2.9	UART2_DMA_TBADR(新增, 0x1C).....	241
16.2.10	UART2_DMA_RBADR(新增, 0x20).....	241
16.2.11	UART2_DMA_TLEN(新增, 0x24).....	242
16.2.12	UART2_DMA_RLEN(新增, 0x28).....	242

16.2.13	UART2_DMA_TADR(新增, 0x2C).....	242
16.2.14	UART2_DMA_RADR(新增, 0x30).....	242
16.2.15	UART2_DMA_IE(新增, 0x34).....	242
16.2.16	UART2_DMA_FLG(新增, 0x38).....	243
16.2.17	UART2_DMA_TO (新增, UART2 模块专有, 0x3C)	243
16.3	UART 数据接收及发送操作步骤	243
17	SPI 1/3/4 (支持普通 DMA)	246
17.1	概述.....	246
17.2	功能描述.....	246
17.3	寄存器.....	247
17.3.1	寄存器列表.....	247
17.3.2	SPIX 控制寄存器 SPI_CTL (0x00)	248
17.3.3	SPIX 状态标识寄存器 SPI_STAT (0x04).....	250
17.3.4	SPIX 数据发送寄存器 SPI_TXDATA (0x08)	251
17.3.5	SPIX 数据接收寄存器 SPI_RXDATA (0x0C).....	251
17.3.6	SPIX 默认发送数据寄存器 SPI_TXDFLT (0x10).....	251
17.3.7	SPIX_DMA_CTL(新增, 0x14).....	251
17.3.8	SPIX_DMA_TBADR(新增, 0x18).....	251
17.3.9	SPIX_DMA_RBADR(新增, 0x1C).....	251
17.3.10	SPIX_DMA_TLEN(新增, 0x20).....	252
17.3.11	SPIX_DMA_RLEN(新增, 0x24).....	252
17.3.12	SPIX_DMA_TADR(新增, 0x28).....	252
17.3.13	SPIX_DMA_RADR(新增, 0x2C).....	252
17.3.14	SPIX_DMA_IE(新增, 0x30)	252
17.3.15	SPIX_DMA_FLG(新增, 0x34).....	253
18	高速 SPIS (支持普通 DMA)	254
18.1	概述.....	254
18.2	功能描述.....	254
18.3	时序描述.....	255
18.4	寄存器.....	256
18.4.1	寄存器列表.....	256
18.4.2	SPIS_CTL (0x00).....	256
18.4.3	SPIS_STIF (0x04).....	257
18.4.4	SPIS_STIE (0x08).....	259
18.4.5	SPIS_STIFE (0x0C).....	259
18.4.6	SPIS_RXDATA (0x10)	259
18.4.7	SPIS_TXDATA (0x14).....	260
18.4.8	SPIS_TXDFLT (0x18)	260
18.4.9	SPIS_FFCNT	261
18.4.10	SPIS_FFCLR (0x24).....	261
18.4.11	SPIS_DMA_CTL(新增, 0x28)	261
18.4.12	SPIS_DMA_TBADR(新增, 0x2C).....	262

18.4.13	SPIS_DMA_RBADR(新增, 0x30).....	262
18.4.14	SPIS_DMA_TLEN(新增, 0x34).....	262
18.4.15	SPIS_DMA_RLEN(新增, 0x38).....	262
18.4.16	SPIS_DMA_TADR(新增, 0x3C).....	262
18.4.17	SPIS_DMA_RADR(新增, 0x40).....	263
18.4.18	SPIS_DMA_IE(新增, 0x44).....	263
18.4.19	SPIS_DMA_FLG(新增, 0x48).....	263
19	SPI0(支持专用 DMA).....	264
19.1	SPI0 概述.....	264
19.2	普通 DMA 传输.....	264
19.2.1	特点.....	264
19.2.2	详细说明.....	264
19.3	专用 DMA 模式.....	266
19.3.1	特点.....	266
19.3.2	详细描述.....	266
19.4	CRC 校验或校验和计算.....	269
19.4.1	特点.....	269
19.5	寄存器定义.....	269
19.5.1	寄存器列表.....	269
19.5.2	SPI 控制寄存器 SPI_CTL (0x00).....	270
19.5.3	SPI 状态标识寄存器 SPI_STAT (0x04).....	272
19.5.4	SPI 数据发送寄存器 SPI_TXDATA (0x08).....	273
19.5.5	SPI 数据接收寄存器 SPI_RXDATA (原有, 0x0C).....	273
19.5.6	SPI 默认发送数据寄存器 SPI_TXDFLT (原有, 0x10).....	273
19.5.7	SPI0_DMA_CTRL(新增, 0x14).....	273
19.5.8	SPI0_DMA_BADR(新增, 0x18).....	275
19.5.9	SPI0_DMA_ADDR(新增, 0x1C).....	275
19.5.10	SPI0_DMA_LEN(新增, 0x20).....	275
19.5.11	SPI0_DMA_FLAG(新增, 0x24).....	275
19.5.12	SPI0_DMA_IE(新增, 0x28).....	275
19.5.13	SPI0_CRC_INIT(新增, 0x2C).....	276
19.5.14	SPI0_CRC(新增, 0x30).....	276
19.6	软件配置流程.....	276
19.6.1	普通 DMA 模式.....	276
19.6.2	专用 DMA 模式.....	276
19.6.3	CRC DMA 模式.....	276
20	IIC 接口.....	278
20.1	概述.....	278
20.2	功能描述.....	278
20.3	寄存器.....	278
20.3.1	I ² C 控制寄存器 I ² C_CTL (0x00).....	278
20.3.2	I ² C 时钟配置寄存器 I ² C_CLK (0x04).....	279

20.3.3	I ² C 状态指示寄存器 I ² C_STAT (0x08)	280
20.3.4	I ² C 从设备地址寄存器 I ² C_ADDR (0x0C).....	281
20.3.5	I ² C 收发数据寄存器 I ² C_DATA (0x10).....	281
21	通用 ADC.....	282
21.1	主要特点.....	282
21.2	寄存器.....	282
21.2.1	寄存器列表.....	282
21.2.2	GPADC_CTL0 (0x00)	283
21.2.3	GPADC_CTL1 (0x04)	284
21.2.4	GPADC_RATIO (0x08)	285
21.2.5	GPADC_IE (0x0C)	286
21.2.6	GPADC_IF (0x10)	286
21.2.7	GPADC_OUT (0x14).....	287
21.2.8	GPADC_R2RBUF_CTL (0x18)	287
22	选项字节.....	289
22.1	芯片保护设置.....	289
22.2	WDT 设置.....	289
22.3	RTC 设置.....	290
23	编程支持.....	291
23.1	概述.....	291
23.2	FLASH 保护机制	291
23.3	在系统编程 (ISP)	291
23.3.1	ISP 通讯协议.....	292
23.3.2	使用的资源.....	293
23.3.3	ISP 命令	293
23.3.4	ISP 返回代码.....	298
23.4	在应用编程 (IAP)	299
23.4.1	IAP 命令	299
23.4.2	IAP 使用	300
23.5	量产平台.....	300
24	焊接条件.....	301
25	封装尺寸.....	302

1 概述

1.1 简介

RN2025 是一款高集成度、宽电压、高精度、高可靠性、低功耗单相计量 SOC 芯片，针对下一代 IR46 智能电表计量芯设计，并广泛适用于能源管理与能耗分析、电力监控、电气安全等领域多种表型设计。

RN2025 提供 RN2025 (A64e04)、RN2025 (B64e04)、RN2025 (B64e05)、RN2025 (C64e05) 这几种封装型号，主要是版本与封装形式的不同，其中 RN2025 (A64e04) 已逐步停产，必须向其他版本进行升级。版本升级注意点如下：

RN2025 (B64e04) 与 RN2025 (A64e04) 硬件及软件向下兼容。

RN2025 (C64e05) 与 RN2025 (B64e05) 硬件及软件向下兼容。

1.2 芯片特性

✓ 基本特点

- 高集成度：集成 ARM Cortex-M0 核、256KB FLASH、32KB SRAM、3 路 Σ - Δ ADC、单相计量及故障检测模块、计量精度自检测模块、6 路复用的 SAR ADC、6 路复用的 GP-ADC、独立供电硬件温补 RTC、各种外设等
 - 宽电压：
 - VCC 供电域保证测量精度的电压范围为 2.8V~5.5V
 - VCC 供电域 CPU 小系统可运行的典型电压范围为 1.9V~5.5V
 - VBAT 供电域为 RTC 独立电源域，时间保持电压范围为 1.8V~5.5V
 - 高性能：
 - 32768HZ 晶振+PLL 条件下，PLL 下 CPU 最高工作频率可达 32.768MHz
 - HOSC 晶振条件下，CPU 最高工作频率可达 32.768MHz
 - 内部高频 RC 最高可工作在 32.768Mhz 下，全温度范围可保证 $\pm 1\%$ 精度，可用作备份时钟。
 - 高精度：
 - 在 8000:1 动态范围内，有功/无功/视在电能误差小于 0.1%；
 - RTC 在 -25 $^{\circ}$ C ~70 $^{\circ}$ C 内秒脉冲误差小于 ± 5 ppm，最小校正刻度为 0.0305175ppm
 - 低功耗：
 - 典型工作电流：10mA(CPU 运行在 16.384MHz，计量开启)
 - CPU 子系统工作在 32Khz 下功耗：<18 μ A (with cache)；
 - 休眠功耗：8 μ A(RTC 自动温补；RAM 保持；CPU 不掉电；电源监测开启；中断唤醒)
 - VBAT 域功耗：典型值 1.5 μ A；
 - 封装形式 1: LQFP64 (7.00 \times 7.00 \times 1.40 e=0.40)，与 RN2025A64 硬件兼容；简称：RN2025B64e04
 - 封装形式 2: LQFP64 (10.00 \times 10.00 \times 1.40 e=0.50)，简称：RN2025B64e05/RN2025C64e05
- 即：RN2025B64 有两种封装，一种是 (7.00 \times 7.00 \times 1.40 e=0.40)，另外一种是 (10.00 \times 10.00 \times 1.40 e=0.50)。在研发阶段，我司会同时提供两种封装的芯片。RN2025B64e04 可以与 RN2025A64e04 直接替换，方便做软件调试。
- RN2025C64 只有一种封装，即：(10.00 \times 10.00 \times 1.40 e=0.50)，可与 RN2025B64e05 直接替换。

考虑到 0.4 间距的封装对于生产工艺要求较高,而且 RN2025 是计量类产品,我们优先推荐 0.5 间距的封装。建议客户在新设计上采用 0.5 间距的封装,未来 0.4 间距的封装将逐步停产。

✓ 计量精度自检测功能

- 可检测电流采样通道外围器件变化引起的精度漂移,检测精度为 0.2%
- 可检测电压采样通道外围器件变化引起的精度漂移,检测精度为 0.2%; 优化电压通道自检测性能。
- 可检测计量芯片自身故障引起的精度变化(比如基准电压变化、电路失效等)

✓ 事件上报

- 过压、过流事件;
- 电压骤降事件;
- 漏电事件;
- 短路事件;

✓ 高级功能

- 支持 41 次谐波分析,谐波计算由硬件加速引擎完成,可实现每周波连续计算。
- 支持非侵入式用电负荷识别所需要的波形数据,支持灵活的帧格式,支持硬件自动计算校验和,支持灵活的校正手段,支持 DMA 从 RAM 将数据外发到 SPI 口。
- 支持正负总谐波电能计算,可从 CF0~CF4 任意一个 IO 口输出。谐波电能计算由硬件加速引擎完成,可实现连续计算。
- 支持分次谐波电能计算,相关内容不在手册开放,请直接调用锐能微库函数。
- 支持 S 级电能质量分析功能。包括谐波、间谐波、不平衡度、电压波动、电压闪变、骤升、骤降等,以上功能分散在本手册各个章节中,主要运算均有硬件完成,其中电压闪变在手册中未开放。如果需要完整的电能质量分析功能,请直接调用锐能微相关库函数实现。
- 支持故障录波功能。

✓ 计量

- 提供全波、基波有功电能,8000:1 动态范围内,非线性误差<0.1%,满足 0.5S 和 0.2S 级有功电能表精度要求
- 仅提供基波无功电能,8000:1 动态范围内,非线性误差<0.1%
- 提供全波、基波视在电能
- 提供有功、无功功率方向,支持无功四象限判断
- 具有潜动启动功能,启动阈值可调
- 电表常数可调
- 提供有功、无功、视在的快速脉冲计数
- 提供全波、基波的有功、无功和视在脉冲从 CF0~CF4 任意一个 IO 口输出

✓ 测量

- 提供全波和基波有功、视在功率,以及基波无功功率
- 提供全波、基波三相电压电流有效值
- 提供电压线频率,测量误差<0.02%
- 提供过零检测,过零阈值可设置
- 提供灵活的电压、电流波形缓存数据及 DMA 功能

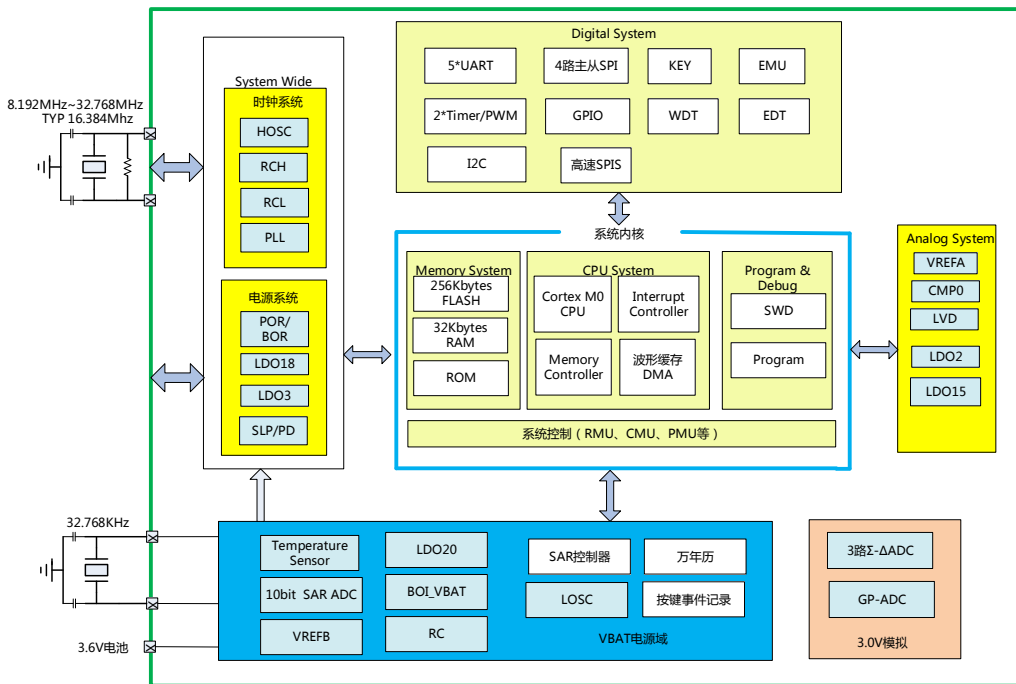
- 提供电压暂降检测
- 提供过压、过流检测
- ✓ 防窃电
 - 提供零线电流测量，零线电流 ADC 的 PGA 倍数最大支持 16 倍；
 - 提供一种低功耗模式 NVM，实现低功耗电流有效值测量，功耗小于 2mA
- ✓ 软件校准
 - 提供三路 ADC 通道增益校正
 - 提供三路 ADC 通道相位校正，其中 A、B 两路电流通道支持分段相位校正
 - 提供功率增益校正
 - 提供有功、无功功率分段相位校正
 - 提供有功、无功、有效值 Offset 校正
- ✓ 内置 1.25V ADC 基准电压，温度系数典型值 10ppm/°C；支持外接基准电压。

- ✓ 处理器相关
 - ARM Cortex-M0 内核，最高运行频率可达 32.768Mhz，典型运行频率 16.384Mhz
 - 256KBytes FLASH 存储器，擦写次数 10 万次，数据保持时间大于 10 年
 - 32KBytes SRAM，其中 2KBytes 作为 Cache 使用，客户可使用 30KBytes。
 - 波形缓存专用 DMA 控制器
 - CM0 内嵌系统定时器
 - 单 cycle 乘法器 (32bit*32bit)
 - 硬件看门狗
 - 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
- ✓ SAR ADC
 - 10bit SAR ADC，内部温度传感器（用于 RTC 温度补偿）、6 路外部引脚输入 AIN 分时复用
 - PGA 支持 0.25 倍/0.5 倍/1 倍/1.5 倍/2 倍，输入信号只要不高于电源电压即可；
- ✓ GP-ADC
 - 16bit Sigma-Delta ADC，支持内部温度传感器测量（用于提升 RTC 打分精度）、支持 6 路外部引脚输入（用于支持端子温度测量）。
 - PGA 支持 0.2 倍/1 倍/2 倍/4 倍
- ✓ VBAT 域电路
 - RTC 硬件自动温补，在-25°C ~70°C 内秒脉冲误差小于 ±5ppm，最小校正刻度为 0.0305175ppm，满足国家标准的精度和功耗要求；
 - 温度传感器：提供准确的温度值，-25°C ~70°C 范围内测温精度为 ±1°C
 - 典型功耗为 1.5μA，全温度范围功耗优于 2uA；
 - 独立电源域，通过 VBAT 引脚独立供电
 - VBAT 引脚与 VCC 引脚可支持不同的供电电压，注意 VBAT 域的四个 IO 口电源为 VBAT
 - VBAT 域的两个按键支持无 CPU 参与情况自动冻结万年历时间，并支持自动通过 IIC 接口将万年历写入外部 EEPROM。

✓ 其他外设

- CF 输出口：5 个
- 高速 GPIO：31 个（**新增 1 个**），支持与不同电压外设器件的接口
- 低速 GPIO：**6 个（新增 2 个）**，在 VBAT 域，建议 VBAT 域的 IO 口不要应用在高速场合。
- 扩展定时器：2 个 32bit 定时器；4 路 PWM 输出；
- 按键中断：8 个，管脚复用
- 外部中断：8 个，管脚复用
- UART：5 路，支持红外调制，支持 UART 唤醒，支持电平反转。**其中 UART2 支持普通 DMA 模式，并且 UART2 还支持在线升级。**
- SPI 口：5 路，其中 1 路专用从模式，4 路主从模式，均支持 DMA 传输。
- SPI2 为高速 SPI 从机模式，支持 10Mbps 传输速率，支持 32*8bit FIFO，**支持 DMA 操作；**
- **SPI0、SPI1、SPI3、SPI4 为主从模式 SPI M/S，其中 SPI0 支持负荷识别专用 DMA 模式；SPI1、SPI3、SPI4 支持普通 DMA 模式。**
- I2C：1 路
- 电压检测 LVD：检测芯片电源电压；一路低功耗比较器 CMP0：检测外部电压，可用于系统掉电及上电检测

1.3 系统框图



1.4 管脚排列

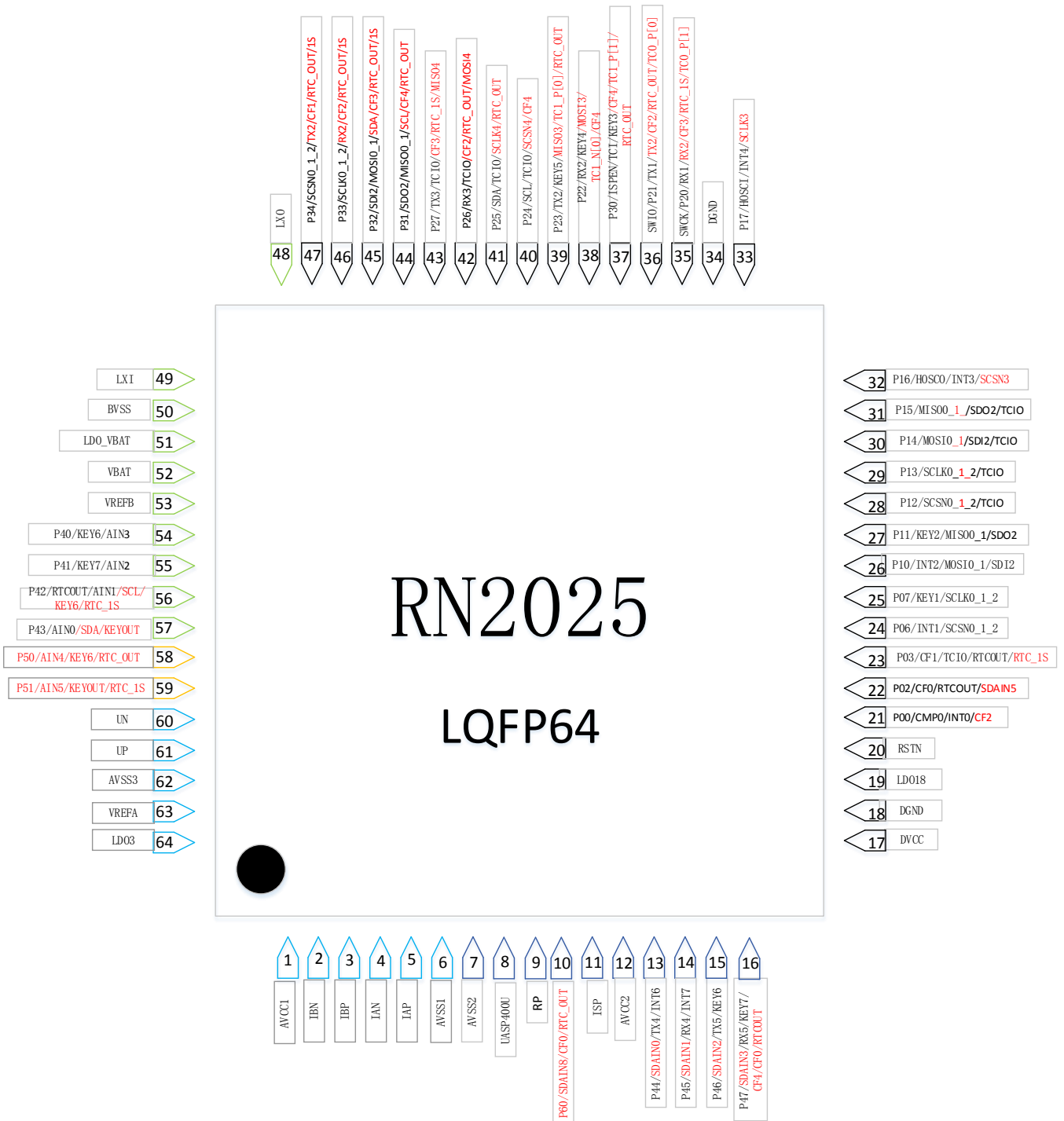


表 1-1 引脚类型说明

类型	模拟	数字								
		IO			输入特性			输出特性		
	模拟	双向	输入	输出	上拉	施密特 CMOS	施密特 TTL+CMOS	OpenDrain	驱动 (mA)	晶振
	A	B	I	O	U	S	L	D	N	X

PABULD6	√	√			√		√	√	6	
PIU			√		√	√				
PUXI		√			√	√				√

表 1-2 引脚说明

引脚	标识	特性	功能描述
1	AVCC1	电源	计量 ADC 电源输入，与 AVCC2 之间应使用 10 欧姆电阻隔离。外接 1uF 电容并联 0.1uF 电容去耦。
2,3	IBN, IBP	模拟输入	电流采样通道 B 的负、正模拟输入引脚。采用完全差分输入方式，正常工作最大差分输入峰值 V_p 为 $\pm 1000\text{mv}$ (PGA=1)。 使用误差自检功能，外围电路请参照手册推荐电路，包括抗混叠 RC 的参数。下同。
4,5	IAN, IAP	模拟输入	电流采样通道 A 的负、正模拟输入引脚。采用完全差分输入方式，正常工作最大差分输入峰值 V_p 为 $\pm 1000\text{mv}$ (PGA=1)。
6	AVSS1	地	内部计量 ADC 的模拟地
7	AVSS2	地	自检测电路模拟地
8	UASP_400	参考电流源	A 相电压故障检测基准 400 μA 电流源方波输出。 对 RN2025A64/RN2025B64 如果不使用误差自检功能，此引脚可浮空，或者按照带误差自检功能的硬件设计（软件不开启误差自检功能）。 对 RN2025C64，即使不使用误差自检功能，也必须按照带误差自检功能的硬件设计（软件不开启误差自检功能）。 但不管采用哪种硬件设计方式，软件都必须调用锐能微库函数文件<sysctl.c>中的 WriteUEdtVBCtlReg 函数，具体见《锐能微 RN202X_RN7326_应用笔记 002-库函数使用说明 v1.2》
9	RP	参考电压	RP 典型输出电压 1.25V，外部应接 6.2K 电阻到地，产生电压通道自检测电路的基准电流。6.2K 电阻要求与电压通道电阻分压电阻采用同类电阻。
10	P60/SDAIN8/CF0/RTC_OUT	IO	P60、SDAIN8、CF0、RTC_OUT 复用引脚。其中 SDAIN 引脚为 GPADC 测量输入（下同），用于直流测试。
11	ISP	参考电压	电流通道自检测电路 P 端的 2V 方波的输出引脚，外接一个 49.9 欧姆左右的低温漂电阻。

			如果不使用误差自检功能，此引脚应浮空。
12	AVCC2	电源	自检电路电源，应外接 220 μ F 电容并联 10 μ F+0.1 μ F 去耦电容。要求供电能力不低于 50mA。
13	P44/ SDAIN0 /TX4/INT6	PABULD6	P44、 SDAIN0 、TX4（不能配置成红外调制输出）、INT6 输入复用；其中 SDAIN 引脚为 GPADC 测量输入（下同），用于直流测试。
14	P45/ SDAIN1 /RX4/INT7	PABULD6	P45、 SDAIN1 、RX4、INT7 输入复用；
15	P46/ SDAIN2 /TX5/KEY6	PABULD6	P46、 SDAIN2 、TX5（不能配置成红外调制输出）、KEY6 输入复用；
16	P47/ SDAIN3 /RX5/KEY7/ CF4/CF0/RTCOUT	PABULD6	P47、 SDAIN3 、RX5、KEY7、 CF4 、 CF0 、 RTCOUT 输入复用；
17	DVCC	电源	数字电源输入，应外接 1 μ F 电容并联 0.1 μ F 电容去耦，与 AVCC2 之间应使用 10 欧姆电阻隔离。
18	DGND	地	内置 LDO18 数字地
19	LDO18	LDO	内置 1.8V LDO 的输出，可给芯片 1.8V 数字域供电，应外接 1 μ F 电容并联 0.1 μ F 电容去耦；
20	RSTN	PIU	外部复位引脚，低电平有效，内置约 50K 上拉电阻；
21	P00/CMP0/INT0/ CF2	PABULD6	P00、比较器 CMP0 输入、外部中断 INT0、 CF2 复用； 外围电路需要注意，默认配置下比较器内部有 600K 对地电阻。
22	P02/CF0/RTCOUT/ SDAIN5	PABULD6	P02、计量脉冲输出 CF0、RTC 秒脉冲 RTCOUT0、 SDAIN5 复用，6mA 驱动能力
23	P03/CF1/TCI/TC0_P[0] /RTC_OUT/RTC_1s	PABULD6	P03、计量脉冲输出 CF1、定时器 0 输出、定时器输入、 RTC_OUT 、 RTC_1s 复用，6mA 驱动能力。其中 RTC_OUT 为每秒准确的秒脉冲信号， RTC_1S 是与万年历同步更新的秒脉冲信号。
24	P06/INT1/SCSN0_1_2	PABULD6	P06-P11：IO 口、INT/KEY、SPI0、SPI1、SPI2 口复用的管脚。上拉电阻可选、TTL/CMOS 输入可选、漏极开路可选； SPI 口可以选择为 SPI0、SPI1 和 SPI2，其中 SPI0 和 SPI1 支持主从模式，SPI2 仅支持从模式。
25	P07/KEY1/SCLK0_1_2	PABULD6	
26	P10/INT2/MOSI0_1/SDI2	PABULD6	

27	P11/KEY2/MISO0_1/SDO2	PABULD6	
28	P12/SCSN0_1_2/TCI/TC0_P[0]	PABULD6	P12-P15: IO 口、SPI0/1 (注: SPI0/1 支持主从模式) / SPI2 (注: SPI2 仅支持从模式)、定时器 0 输出、定时器输入复用; 上拉可选、TTL/CMOS 电平可选、漏极开路可选。
29	P13/SCLK0_1_2/TCI/TC0_N[0]	PABULD6	
30	P14/MOSI0_1/SDI2/TCI/TC0_P[1]	PABULD6	
31	P15/MISO0_1/SDO2/TCI/TC0_N[1]	PABULD6	
32	P16/HOSCO/INT3/SCSN3	PUXI	P16-P17: IO 口/高频晶体/中断口/SPI3 复用。高频晶体典型频率: 8.192Mhz/16.384Mhz/32.768Mhz。 高频晶体端口外部应串接一个 10M 欧的电阻, 并联两个 15pf~22pf 的电容。 需要选择负载电容小于 15pF 的晶振。 注: 选择 32.768Mhz 晶振时, 不推荐使用外接 22pF 电容, 且要求晶振的 ESR<60Ω。
33	P17/HOSCI/INT4/SCLK3	PUXI	
34	DGND	地	数字地
35	SWDCLK/P20/RX1 /RX2/CF3/RTC_1s/TC0_P[1]	PABULD6	SWD 时钟、P20 口、RX1、RX2、CF3、RTC_1s、TC0_P[1] 输入复用引脚。
36	SWDIO/P21/TX1 /TX2/CF2/TC0_P[0]/RTC_OUT	PABULD6	SWD 数据口、P21 口、TX1、TX2、CF2、TC0_P[0]、RTC_OUT 输入复用引脚。
37	P30/ISPEN/TCI/KEY3/ CF4/TC1_P[1]/RTC_OUT	PABULD6	IO 口、定时器输入、KEY 口、CF4、TC1_P[1]、RTC_OUT 复用; 上拉可选、施密特输入、6mA 驱动能力。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。
38	P22/RX2/KEY4 /MOSI3/TC1_N[0]/CF4	PABULD6	IO 口、RX2、KEY4、MOSI3、TC1_N[0]、CF4 输入复用;
39	P23/TX2/KEY5 /MISO3/TC1_P[0]/RTC_OUT	PABULD6	IO 口、TX2、KEY5、MISO3、TC1_P[0]、RTC_OUT 输入复用;
40	P24/SCL/TCI/TC1_P[0] /SCSN4/CF4	PABULD6	P24-P25: P2 口、I2C、定时器 1 输出、定时器输入、CF4、RTC_OUT 复用。 上拉电阻可选、TTL/CMOS 输入可选、漏极开路可选;
41	P25/SDA/TCI/TC1_P[1] /SCLK4/RTC_OUT	PABULD6	
42	P26/RX3/TCI/TC1_N[0] /CF2/RTC_OUT/MOSI4	PABULD6	P26-P27: IO 口、UART3、定时器 1 输出、定时器输入、CF、RTC、SPI4 复用; 上拉可选、TTL/CMOS 电平可选、漏极开路可选。
43	P27/TX3/TCI/TC1_N[1] /CF3/RTC_1s/MISO4	PABULD6	
44	P31/SDO2/MISO0_1/	PABULD6	

	SCL/CF4/RTC_OUT		P31-P34: IO 口与 SPI0/1 (注: SPI0/1 支持主从模式)、SPI2 (注: SPI2 仅支持从模式) 及 CF/RTC/IIC 等复用; 上拉可选、TTL/CMOS 电平可选、漏极开路可选。
45	P32/SDI2/MOSI0_1/ SDA/CF3/RTC_OUT/RTC_1S	PABULD6	
46	P33/SCLK2/SCLK0_1/ RX2/CF2/RTC_OUT/RTC_1S	PABULD6	
47	P34/SCSN2/SCSN0_1/ TX2/CF1/RTC_OUT/RTC_1S	PABULD6	
48	LXO	时钟	32.768KHz 无源晶振输出和输入。
49	LXI	时钟	不需要外接电阻和电容, 需要用地线将之隔离。
50	BVSS	地	VBAT 域地
51	LDO_VBAT	电源	VBAT 域 LDO 输出, 外部需接 0.22μf 电容;
52	VBAT	电源	3.6V 电池或超级电容输入引脚; 仅给 RTC 部分供电。同时也是 SAR ADC 的输入, 对该引脚做测量时内部有两个 300K 的电阻做分压, 并使用 0.5 倍增益。建议外接 RC 滤波, R:10Ω, C:1μF。
53	VREFB	参考电压	SAR ADC 的内置基准输出, 典型值为 1.25V, 外部需接 0.22μf 电容;
54	P40/KEY6/AIN3	PABULD6	VBAT 域 P40、KEY6、SAR-ADC 模拟输入 AIN3 复用。 AIN3 可以支持直接对输入 3.6V 的信号进行测量, 类似 SAR ADC 对 VBAT 引脚电压的测量。启动测量时, 内部使用两个 300K 的电阻将 3.6V 分压到 1.8V, 并使用 0.5 倍 PGA, 只有在测量时才会有功耗。 该引脚做为 KEY6 功能时可在没有 cpu 参与的情况下, 记录按键事件发生的年月日时分秒, 详见 RTC 章节。
55	P41/KEY7/AIN2	PABULD6	VBAT 域 P41、KEY7、SAR-ADC 模拟输入 AIN2 复用。 AIN2 的输入范围不能超过 1.25V (PGA=1) 或者 2.5V (PGA=0.5) 或电源电压 (PGA=0.2 倍)。在外部接 0.1uF 电容到地的情况下, 输入阻抗约为 5 兆欧姆; 如果外部不接电容, 输入阻抗约为 600K 欧姆。 该引脚做为 KEY6 功能时可在没有 cpu 参与的情况下, 记录按键事件发生的年月日时分秒, 详见 RTC 章节。
56	P42/RTCOUT2/AIN1/ SCL/KEY6/RTC_1S	PABULD6	VBAT 域 P42、RTCOUT、SAR-ADC 模拟输入、SCL、KEY6、RTC_1S 复用。

			<p>AIN1 的输入范围不能超过 1.25V (PGA=1) 或者 2.5V (PGA=0.5) 或电源电压 (PGA=0.2 倍)。</p> <p>在外部接 0.1uF 电容到地的情况下，输入阻抗约为 5 兆欧姆；如果外部不接电容，输入阻抗约为 600K 欧姆。</p> <p>注意：该引脚不建议接电池测量。原因：VBAT 域单独上电时，该引脚默认输出 RTCOUT，对外有驱动；当 vcc 上电后，该引脚被修改为 P42（高阻态）。</p>
57	P43/AIN0/SDA/KEYOUT	PABULD6	<p>VBAT 域 P43、SAR-ADC 模拟输入、SDA、KEYOUT 复用。AIN0 的输入范围不能超过 1.25V (PGA=1) 或者 2.5V (PGA=0.5) 或电源电压 (PGA=0.2 倍)。</p> <p>在外部接 0.1uF 电容到地的情况下，输入阻抗约为 5 兆欧姆；如果外部不接电容，输入阻抗约为 600K 欧姆。</p> <p>备注：KEYOUT 为按键触发输出，用于在按键事件发生时控制外部设备的电源（比如 EEPROM 的电源，低有效），事件记录完毕后会自动关闭电源（高阻态）。</p>
58	P50/AIN4/KEY6/RTC_OUT	PABULD6	<p>VBAT 域 P50、SAR-ADC 模拟输入、KEY6、RTC_OUT 复用。AIN4 的输入范围不能超过 1.25V (PGA=1) 或者 2.5V (PGA=0.5) 或电源电压 (PGA=0.2 倍)。</p> <p>在外部接 0.1uF 电容到地的情况下，输入阻抗约为 5 兆欧姆；如果外部不接电容，输入阻抗约为 600K 欧姆。</p>
59	P51/AIN5/KEYOUT/RTC_1S	PABULD6	<p>VBAT 域 P51、SAR-ADC 模拟输入、KEYOUT、RTC_1S 复用。AIN0 的输入范围不能超过 1.25V (PGA=1) 或者 2.5V (PGA=0.5) 或电源电压 (PGA=0.2 倍)。</p> <p>在外部接 0.1uF 电容到地的情况下，输入阻抗约为 5 兆欧姆；如果外部不接电容，输入阻抗约为 600K 欧姆。</p> <p>备注：KEYOUT 为按键触发输出，用于在按键事件发生时控制外部设备的电源（比如 EEPROM 的电源，低有效），事件记录完毕后会自动关闭电源（高阻态）。</p>
60,61	UAN, UAP	模拟输入	<p>电压采样通道 A 的负、正模拟输入引脚。采用完全差分输入方式，正常工作最大差分输入峰值 V_p 为 $\pm 1000\text{mv}$ (PGA=1)。</p>
62	AVSS3	地	<p>内置参考电压模拟地</p>

63	VREFA	参考电压	ADC 的参考基准，典型值为 1.25V，需外接 0.1uF 和 1uF 电容。
64	LDO3	LDO	计量 ADC 的供电 LDO，典型值为 3V，需要外接 0.1uF 和 1uF 电容。

1.5 电气特性

表 1-3 电气特性

测量项目	符号	最小	典型	最大	单位	测试条件和注释
有功电能测量误差	Err			±0.1%		8000:1动态范围
无功电能测量误差	Err			±0.1%		8000:1动态范围
视在电能测量误差	Err			±0.1%		8000:1动态范围
有效值测量误差	Err			±0.2%		2000:1动态范围
功率测量误差	Err			±0.1%		2000:1动态范围
测量带宽	BW		4		kHz	
全失压电流测量误差				±0.5%		400:1动态范围
频率测量误差	Err			0.02%		
频率测量范围		40		75	Hz	
相角测量误差	YErr		0.02		°	电流通道50mV输入， 相角 60° 120° 240° 300°
电能脉冲输出						
最大频率	Hz			20KHz		
占空比	%		50%			当脉宽低于 84ms 时
高电平脉宽	ms		84ms			
Sigma-Delta ADC性能						
最大信号电平	V _{xn}			±1000	mVp	PGA=1, 差分信号
ADC 失调误差	DC _{off}		1		mV	
-3dB 带宽	B _{-3dB}		4		kHz	
电流通道串扰			-110		dB	UA=UB=UC=800mVpp
基准电压						
(VCC=3V~5.5V, 温度范围: -40°C~+85°C)						
输出电压	V _{ref}	1.25	1.26	1.27	V	
温度系数	T _c		10	15	ppm/°C	
计量自检测精度						
(VCC=3.3V±10%, 温度范围: -40°C~+85°C)						
电流通道自检测精度1 (幅度)	IedtA1		0.3%			40mA检测电流 30 秒测量时间 5A(2mA)CT, 2*10Ω 采样电阻
电流通道自检测精度2 (幅度)	IedtA2		0.2%			40mA检测电流 60 秒测量时间 5A(2mA)CT, 2*10Ω 采样电阻
电流通道自检测精度	IedtP1		0.03度			40mA检测电流

(相位)						30 秒测量时间 5A(2mA)CT, 2*10Ω采样电阻
电流通道自检测精度 (相位)	IedtP1		0.03度			40mA检测电流 60 秒测量时间 5A(2mA)CT, 2*10Ω采样电阻
电压通道自检测精度1 (幅度)	UedtA1		0.5%			400uA检测电流 30 秒测量时间
电压通道自检测精度2 (幅度)	UedtA2		0.1%			10uA检测电流 20 秒测量时间
电压通道自检测精度 (相位)	UedtP1		0.03度			10uA检测电流 20秒测量时间
模拟外设(温度范围: -40℃~+85℃)						
SAR ADC 输入范围	SAR-IN	0		BGR	V	BGR 为内部低功耗基准, 典型值为 1.25V
SAR ADC 采样率			1		KHz	系统时钟 Mhz
低功耗比较器 LVD 阈值	LVD	2.3		4.9	V	LVD 的阈值是可配置的, 从 2.3V 到 4.9V
低功耗比较器 CMP0 阈值 Vi11	CMP0	1.23	1.28	1.33	V	Vi1 为默认配置下比较器输出低电平的阈值; 输出高电平比较结果的阈值 Vi1 比 Vi1 高 220mv
低功耗比较器 CMP0 阈值 Vi12	CMP0	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。
VBAT 测量	VBAT D	0	3.6	5	V	SAR ADC 对 VBAT 引脚电压的测量范围;
温度测量			1		℃	
每次温补时间	Ttps		2		ms	
时钟参数(温度范围: -40℃~+85℃)						
输入低频时钟频率范围	XI		32.768		KHz	
输入高频时钟频率范围	HOSI	4.096	16.384	32.768	Mhz	
内部PLL时钟频率范围	PLL		8.192	16.384	MHz	fosc=32.768Khz
高频时钟XI输入电容	Cxi	15		22	pf	
高频时钟XO输入电容	Cxo	15		22	pf	
高频时钟晶振ESR	ESR		80		欧姆	5倍起振裕度
内部高频RCH	RCH	2.6	2.9	3.2	MHz	用于芯片复位后默认

						时钟
内部低频RCL	RCL	20	32	40	KHz	用于WDT时钟
电源(温度范围: -40℃~+85℃)						
主电源	VCC	2.4	3.3/5	5.5	V	建议主电源选择为3.3V±5%或者5V±5%
CPU 最低工作电压	Vil	1.8	1.9	2.0	V	全温度范围
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
工作电流	Idd1		8.5		mA	CPU运行在8.192MHz(PLL), 计量开启
	Idd2		10		mA	CPU运行在16.384MHz(PLL), 计量开启
	Idd3		15		mA	CPU运行在32.768MHz(HOSC), 计量开启
VCC 休眠功耗	SIdd		8		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		1.5	2	μA	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		300		uA	典型条件: Vbat=3.6V; Tc=25度;
LDO18	V1P8	1.62	1.8	1.98	V	CPU内核电压
LDO3	V3	2.85	2.9	2.95	V	Σ-Δ ADC电压
LDO_VBAT	V2	1.9	2	2.2	V	VBAT域数字电压
极限参数(温度范围: -40℃~+85℃)						
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	
IAP, IAN, IBP, IBN, ICP, ICN, INP, INN		-6		+6	V	
VAP, VAN, VBP, VBN, VC		-6		+6	V	

P,V _{VCN}						
数字IO输出高电平	VOH		--	DV _{DD} +0.3	V	
数字IO输出低电平	VOL	-0.3	--		V	
数字IO输入高电平	VIH		0.7VCC			CMOS
数字IO输入低电平	VIL		0.3VCC			CMOS
数字IO输入高电平	VIH		0.4VCC			TTL
数字IO输入低电平	VIL		0.2VCC			TTL
数字IO的I _{source}	I _{source}	5		10	mA	6mA类型
数字IO的I _{sink}	I _{sink}	7		15	mA	6mA类型
模拟输入电压相对于AGND	V _{INA}	-0.3	--	AV _{DD} +0.3	V	
工作温度范围	T _A	-40	--	85	°C	
存储温度范围	T _{stg}	-65	--	150	°C	

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	MSD	按照标准IPC/JEDEC J-STD-020D.1评定	3级	/

1.6 应用推荐电路

图 1: RN2025（A64）推荐图:

说明 1: RN2025（B64e04）跟 RN2025（A64）硬件兼容，可直接替换使用。

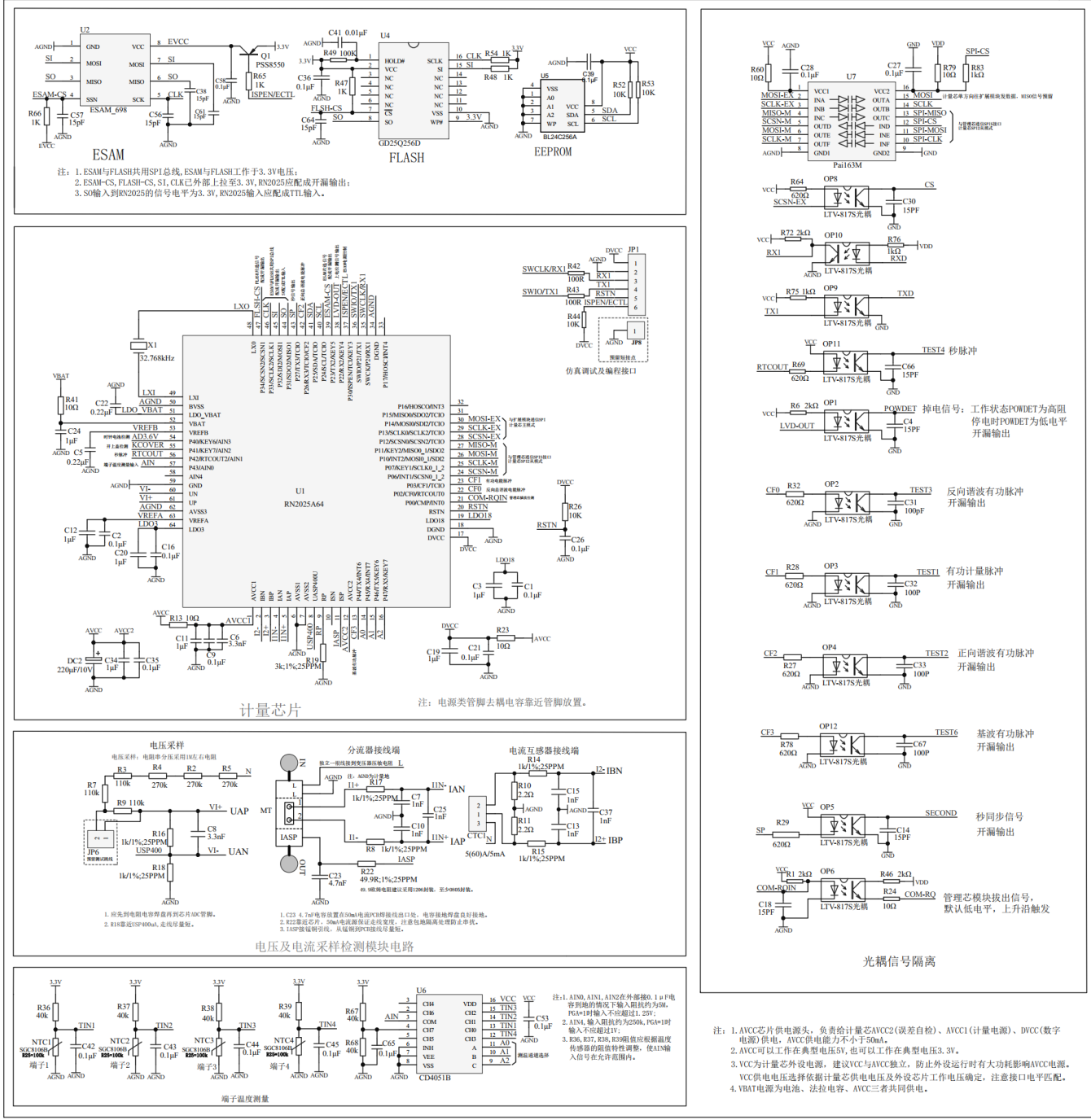
说明 2: RN2025（B64e05）跟 RN2025（C64e05）硬件兼容，可直接替换使用。

图 2: RN2025（B64/C64）推荐图:

RN2025（B64/C64）推荐电路图做了如下改进:

- 1) ESAM 与 Flash 的 spi 口不再需要复用;
- 2) 不再需要外接模拟开关，可直接使用 RN2025（B64/C64）的多路复用 GP-ADC 实现端子测温的测量;
- 3) RP 管脚对地电阻 R29,推荐阻值由“3kΩ”，修改为“6.2kΩ”

图 1：RN2025 (A64) 推荐图



2 系统控制

2.1 电源域划分

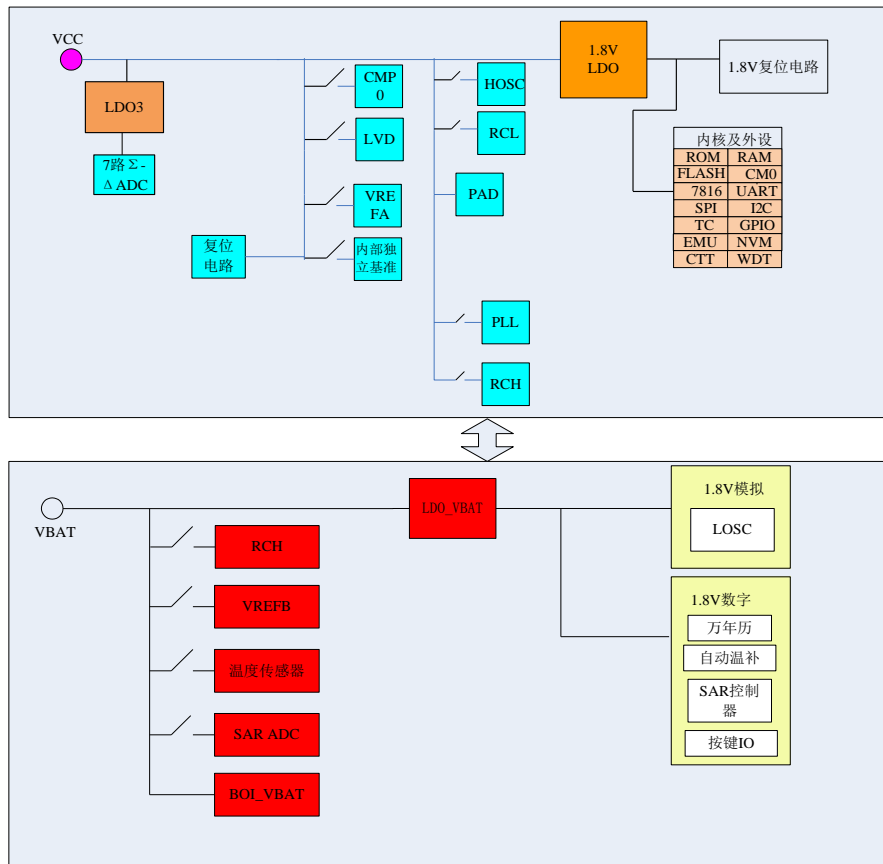


图 2-1 内部电源管理

VCC 与 VBAT 独立供电，RTC 相关电路（32768 晶振/万年历/自动温补电路等）、SAR-ADC 测量相关电路、P40-P43/P50/P51 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，VCC 正常工作的典型范围是 1.9V~5.5V，VBAT 万年历正常工作的典型范围是 1.8V~5.5V，VBAT 温补电路正常工作的典型范围是 2.5V~5.5V。

注意 P40-P43/P50/P51 这 6 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 ViH 电平要求（ $0.7 \times \text{VBAT}$ ）。

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 做配置时，VBAT 尚未准备就绪，造成温补参数不能正常写入。因此建议客户硬件设计时注意 VCC 和 VBAT 的上电顺序，建议同时上电或者 VBAT 先上电。

另外建议 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（RTC 初始频率偏差校正函数）对相关寄存器做二次配置，以确保配置参数被正确写入。这对于 VBAT 上电比 VCC 慢的应用尤其重要。

2.2 掉电处理

RN2025 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. 计量模块的电压暂降检测（SAG）

此功能在检测交流电压的跌落，可设置跌落阈值以及跌落时间，比如设置为 $0.5U_n$ ，持续时间为 100ms，具体的设置阈值及持续时间需要根据硬件来定。

在电表应用中，SAG 功能可第一时间发现掉电事件；

2. 模拟外设模块的 CMP0

CMP0 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 CMP0 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，灵活选择 SAG 或者 CMP0 功能完成掉电检测，或者两个功能一起配合使用做掉电检测。

3. 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 CMP0 大，建议选用 CMP0 完成外部输入电压检测），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，阈值可设）。

建议客户软件启用 LVD 功能，当检测到 VCC 电压低于 2.8V 时，无论软件在做什么，应立即让芯片进入休眠模式或者低频运行模式（32.768KHz 或者内部 RC），不应再跑在全速模式，原因是计量模块、高频晶振模块等在低于 2.8V 时不再保证性能。

2.3 时钟源

● 外部时钟源有两个：

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 8.192MHz，16.384MHz，和 32.768MHz 晶振

● 内部时钟源有四个：

RCH：内部高频 RC 时钟（典型值为 2.9MHz），CPU 上电复位后，时钟默认为 RCH；RCH 可以选择 1/2 分频或不分频。

RC32M：内部 32MHz 高频 RC 时钟（范围为 32.768MHz \pm 1%），可用于系统备份时钟。

RCL：内部低频 RC 时钟，用于 WDT 时钟，也可用于电池供电下的 CPU 低频运行时钟

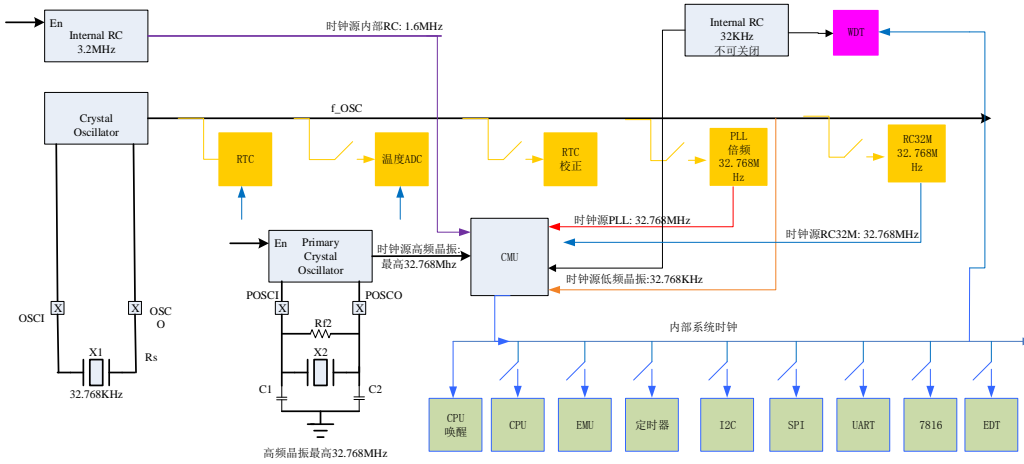
PLL：内部 PLL 时钟，将 32.768KHz 倍频到 8.192M/16.384Mhz/32.768Mhz。

上述时钟源均可作为 cpu 的运行时钟，cpu 工作在如下三种模式：

- **低频运行模式：可以选择为 LOSC 或者 RCL，一般在备用电源下使用。**
- **默认运行模式：CPU 复位后默认运行在 RCH 下，该模式的时钟准确度不高**
- **高频运行模式：可工作在 HOSC 外部高频晶振、内部 PLL，在 HOSC 或者 PLL 都失效时，建议工作在备用高频时钟 RC32M 下。**

CPU 通过指令从低频时钟切换到高频时钟。系统主时钟可在 RCH、PLL（或 HOSC/RC32M）、LOSC（或 RCL）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(0x1C)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。



2.4 时钟切换

包含有如下切换：

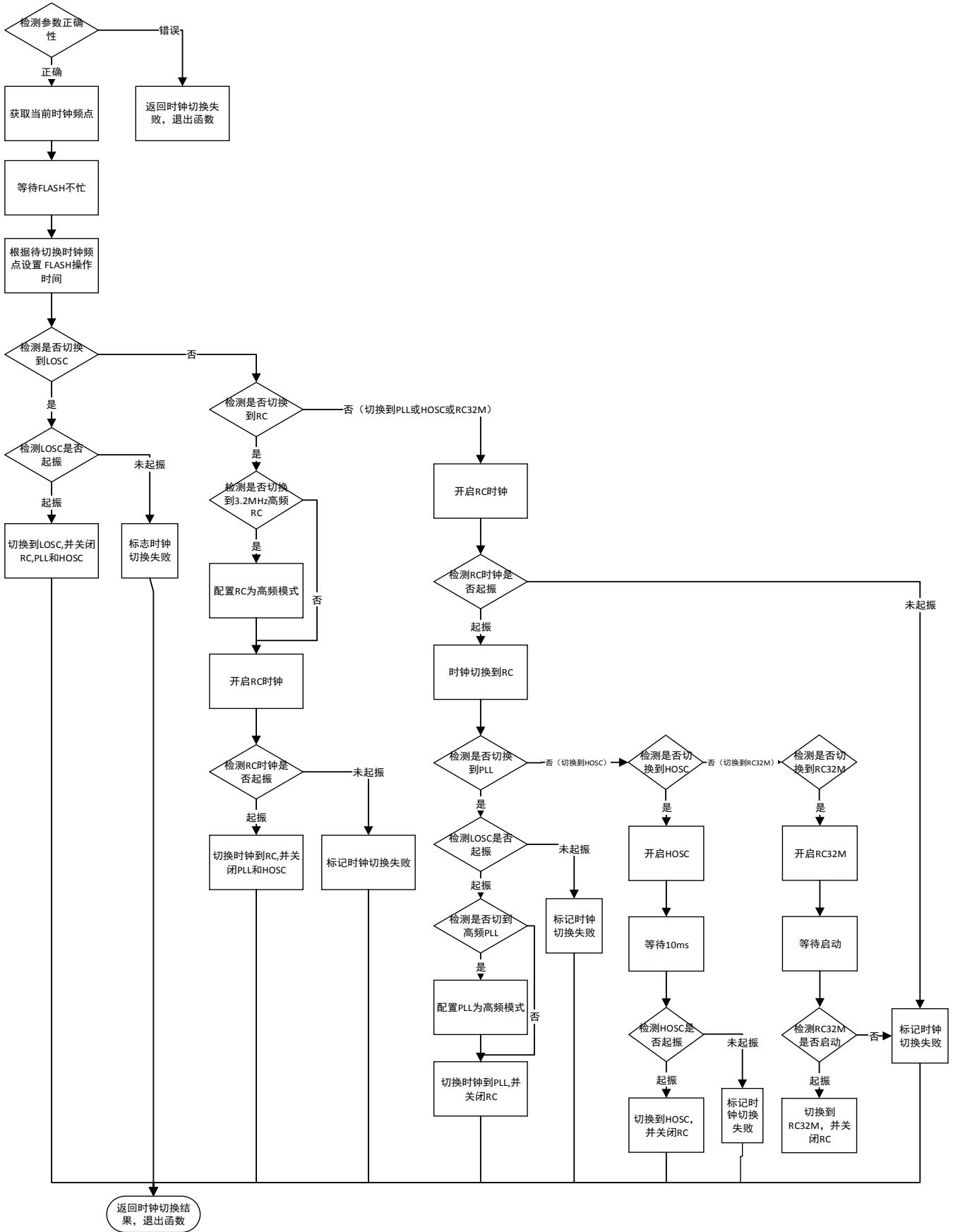
1. 复位后默认为 RCH；
2. PLL/HOSC/RC32M 与 RCH 切换，由 CPU 指令完成；
3. PLL/HOSC/RC32M 与 LOSC/RCL 切换，由 CPU 指令完成；
4. LOSC/RCL 与 RCH 切换，由 CPU 指令完成。

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



2.5 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 7uA 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
1.8V 域	
M0 内核	开启，可关时钟，永不掉电
中断系统	开启，可关时钟，永不掉电
SRAM	开启，可关时钟，永不掉电
ROM	开启，可关时钟，永不掉电
FLASH	开启，CPU 休眠后可自动关电
WDT	开启，不可关断
EMU（正常计量）	关闭，可关时钟，永不掉电
LMM（全失压计量）	关闭，可关时钟，永不掉电
EDT	关闭，可关时钟，永不掉电
其他外设	关闭，可关时钟，永不掉电
VCC 域	
计量 ADC	关闭，可关电源
计量 REF	关闭，可关电源
EDT 基准源	关闭，可关电源
3.3V LDO	关闭，可关电源
1.8V LDO	开启，不可关断
RCL	开启，不可关断
RCH	开启，可关电源
比较器	关闭，可关电源
温度 ADC	定时开启
LVD	关闭，可关电源
比较器	关闭，可关电源
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。

高频模式 HCM:

- 在运行模式下, M0内核、存储器、系统控制(时钟、复位、电源管理、模式控制)等最小系统在运行
- 运行的频率默认为16.384MHz。
- 在运行模式下, 外设的电源和时钟可以根据需要选择打开或关断。

低频模式 LCM:

- 电池供电需要工作在低频模式下。
- 系统控制(时钟、复位、电源管理、模式控制)和 RTC 计时部分一直有电, 并且时钟一直打开工作;
- RTC 校正部分(温度传感器、SAR ADC、RTC 校正计算电路)的电源及时钟根据配置定时打开;
- CPU 最小系统(内核、SRAM 或者 FLASH)一直有电, 时钟可选;
- IO 口一直有电, 以便响应唤醒事件;
- 外设电路(如计量模拟部分、1.8V 数字部分)相关电路可选, 开启后功能不正常, 建议关闭;
- CPU 在复位或者中断来之前都不能执行指令, 唤醒后运行在 32.768Khz 下。

2.6 复位

2.6.1 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。

2.6.2 上下电复位

内置三组上电复位(POR)电路和掉电复位(BOR)电路, 分别对 VCC、LDO18(1.8VLDO)及 VBAT 进行监测。

该电路始终处于工作状态, 不可被关闭, 保证系统在超过阈值(约 1.9V)时可以正常工作; 在低于阈值(约 1.8V)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。

2.6.3 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。可通过调用 M0 提供的函数 NVIC_SystemReset()置位 SCB_AIRCR 寄存器的 SYSRESETREQ 位, 详情详见 M0 说明文档。

2.6.4 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。

2.7 寄存器

下文寄存器以 0x 开头或者以 H 结尾的数值, 均代表十六进制。

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	0x00	系统 OSC 控制寄存器 1
SYS_MODE	0x04	系统模式切换寄存器
SYS_PD	0x08	系统掉电控制寄存器
ADC_CTL	0x0C	ADC 控制寄存器
CTT_CTL	0x18	CTT 控制寄存器
OSC_CTL2	0x1C	系统 OSC 控制寄存器 2
SYS_RST	0x20	系统复位寄存器
SYS_MAPCTL	0x24	地址映射控制寄存器
MOD0_EN	0x28	模块使能 0 寄存器
MOD1_EN	0x2C	模块使能 1 寄存器
INTC_EN	0x30	INTC 使能寄存器
KBI_EN	0x34	KBI 使能寄存器
CHIP_ID	0x38	芯片版本号
SYS_PS	0x3C	系统控制寄存器密码保护位
IRFR_CTL	0x40	RCH 模式下红外时钟分频系数
TRIM_CFG1	0xA0	时钟 TRIM 配置寄存器
TRIM_START	0xA4	时钟 TRIM 结果寄存器

注意：只有当 SYS_PS (0x3C) = 8'h82 时，寄存器才可写

2.7.1 OSC_CTL1 (0x00)

系统 OSC 控制寄存器 1:

比特位	名称	描述	读/写标志	复位值
31:17	---	预留	R	0
16:11	CLOCK_FLAG	时钟开启标志位：如果时钟开启了，则此位为 1： { RC32M,HOSC,RCL,RCH,PLL, LOSC }	R	001101
10:8	SYSCLK_STAT	系统主时钟频率指示： 000：当前系统主时钟为 8.192 MHz； 001：当前系统主时钟为 4.096 MHz / 3MHz； 010：当前系统主时钟为 2.048 MHz / 1.5MHz； 011：当前系统主时钟为 32.768 KHz； 100：当前系统主时钟为 16.384 Mhz； 101：当前系统主时钟为 32.768 Mhz； 其他：保留	R	010
7	PLL_LOCK	PLL 锁定状态 0：未锁定 1：锁定	R	0
6	HCM_ON	系统运行在高频时钟时（PLL、外部晶振或 RC32M），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 IRCH 时钟时，该位为 1； 系统运行在其他时钟时，该位为 0。 复位时，系统默认运行在 1.5Mhz 内部 RCH。	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	HOSC_DIV	系统主时钟分频选择：（只对高频时钟模式有效） 00：以高频时钟源头作为 CPU 主时钟； 01：以高频时钟的二分频作为 CPU 主时钟； 10：以高频时钟的四分频作为 CPU 主时钟； 11：以高频时钟（时钟频率选择为 16.384MHz 和 32.768MHz 时）的八分频作为 CPU 主时钟，若高频时钟为 8.192MHz 则表示为四分频； 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。高频时钟包括外部晶振、PLL、RC32M。	R/W	01
1	IRCH_PD	3.2MHz 内部 RC 使能位： 0：打开； 1：关闭。	R/W	0
0	PLL_PD	PLL 模块使能位 0：打开 1：关闭	RW	1

2.7.2 SYS_MODE (0x04)

系统模式设置寄存器：

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、HCM_ON（OSC_CTL1 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。

2.7.3 SYS_PD (0x08)

系统掉电控制寄存器：

比特位	名称	描述	读/写标志	复位值
31:30	---	预留	R	0
29	---	可写，用户不要改变配置	R/W	0
28	Reserved	可写，内部测试寄存器，默认值为 1，用户不要改变	R/W	1

		此寄存器默认值。		
27	Reserved	可写，内部测试寄存器，默认值为 1，用户不要改变此寄存器默认值。	R/W	1
26	Reserved	可写，内部测试寄存器，默认值为 1，用户不要改变此寄存器默认值。	R/W	1
25	LDO3_PD	LDO3 模块使能，LDO3 用于给 7 路计量 ADC 供电 0: 模块开启 1: 模块关闭 注意：与 ADC PD 不联动，开启计量 ADC 前需软件配置 LDO3_PD 为 0	R/W	1
24	EDT_BGR2_PD	EDT 检测用，内部独立基准电源开关 0: 模块开启 1: 模块关闭	R/W	1
23	VREFA_PD	计量基准电压 VREFA 电源开关， 0: 模块开启 1: 模块关闭 备注：与 ADC PD 不联动，开启计量 ADC 前需软件配置 VREFA_PD 为 0	R/W	1
22	----	可写，内部保留寄存器，默认值为 1，用户不要改变寄存器默认值	R/W	1
21	----	可写，内部保留寄存器，默认值为 1，用户不要改变寄存器默认值	R/W	1
20	AIN4_PD	A 版：AIN4 通道 ADC 的开关； B 版：AIN4_PD 控制位不再使用，默认值为 1，用户不要改变此默认值。 原 AIN4 adc 由 sigma-delta adc 换成了 SAR-ADC。	R/W	1
19	ADCU_PD	UA 通道 ADC 电源开关 0: 模块开启 1: 模块关闭	R/W	1
18	----	可写，内部保留寄存器，默认值为 1，用户不要改变寄存器默认值	R/W	1
17	ADCIB_PD	IB 通道 ADC 电源开关 0: 模块开启 1: 模块关闭	R/W	1
16	ADCIA_PD	IA 通道 ADC 电源开关 0: 模块开启 1: 模块关闭	R/W	1
[15:11]	---	预留	R	0
10	HYSEN_CMP0	CMP0 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	---	预留	R	0
8	PWD_CMP0R	CMP0 内部 600K 电阻采样开关	R/W	0

		0: CMP0 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 V_{il} 典型值为 1.28V, 迟滞为 0.22V, 此时不要将 bit10 配置为 1; 1: CMP0 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 V_{il} 典型值为 0.84V;		
[7:5]	Reserved	保留	R	0
4	CMP0_PD	比较器 0 电源开关 0: 模块开启 1: 模块关闭	R/W	1
3	LVD_PD	LVD 电源开关 0: 模块开启 1: 模块关闭	R/W	1
[2:0]	Reserved	保留	R	3'h0

2.7.4 ADC_CTL (0x0C)

ADC 控制寄存器

比特位	名称	描述	读/写标志	复位值
31:14	---	保留	R	0
13:12	----	可写, 内部保留寄存器, 默认值为 0, 用户不要改变寄存器默认值	R/W	0
11:10	----	可写, 内部保留寄存器, 默认值为 0, 用户不要改变寄存器默认值	R/W	0
9:8	----	可写, 内部保留寄存器, 默认值为 0, 用户不要改变寄存器默认值	R/W	0
7:6	ADCU_PGA	UA 通道 ADC 增益配置 =x00 1 倍 =x01 2 倍 =x10 4 倍 =x11 4 倍	R/W	0
5:4	----	可写, 内部保留寄存器, 默认值为 0, 用户不要改变寄存器默认值	R/W	0
3:2	ADCIB_PGA	IB 通道 ADC 增益配置 =x00 1 倍 =x01 2 倍 =x10 8 倍 =x11 16 倍	R/W	0
1:0	ADCIA_PGA	IA 通道 ADC 增益配置 =x00 1 倍 =x01 2 倍 =x10 8 倍 =x11 16 倍	R/W	0

2.7.5 CTT_CTL (0x18)

CTT 控制寄存器删除, 写入无意义。

A 版本芯片的 P44 口复用配置与此寄存器有关。

B/C 版本芯片的 P44 口复用配置与此寄存器无关。

2.7.6 OSC_CTL2(0x1C)

系统 OSC 控制寄存器 2

比特位	名称	描述	读/写标志	复位值
31:28	保留	保留	R	0x0
27:26	RC32M_OK[1:0]	RC32M_OK[1]表示 RC32M 内部振荡 OK RC32M_OK[0]表示 RC32M 内部 BGR OK 00: 表示 RC32M 尚未 OK 01: 表示内部 BGR 已经 OK, 内部振荡尚未 OK; 该状态下可以快速启动振荡。 10: 表示 BGR 异常, 振荡正常, 这是异常的状态。 11: 表示 RC32M 正常	R	0x0
25:24	RC32M_FREQ[1:0]	00: RC32M 选择为 8.196MHz; 01: RC32M 选择为 16.384Mhz; 10: RC32M 选择为 32.768Mhz 11: 预留	R/W	0x0
23:21	RC32M_EN[2:0]	缺省值 3'h0; 内部 32Mhz RC 振荡器 RC32M 的开启控制位。 000, 关闭; 111, 开启; 开启顺序: 先写 001, 再写 011, 最后写 111,开启 RC32M, 开启后可读取 OSC_CTL1[16] 确认 RC32M 已开启。 关闭顺序: 先写 110; 再写 000; 先开启 RC32M 后, 再选择 RC32M 作为系统时钟。	R/W	0x0
20	RC32M_SEL	若 PLL_HOSC_SEL=0 时, 该位有效, 用于选择 RC32M 作为系统时钟源; 1: 选择 RC32M 作为系统时钟源, 频率由 RC32M_FREQ 定义; 0: RC32M 不作为系统时钟源。 *该配置项只能在 RCM 模式或 LCM 低频模式下配置。	R/W	0x0
19:17	Reserved	保留	R	0x0
16	RCH_FREQ	=0: RC 模式下 RCH 频率为 1.6MHz; =1: RC 模式下 RCH 频率为 3.2MHz; 备注: 此寄存器只能被上下电复位清零。 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	保留	R	0

13	Reserved	保留	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
9:8	Reserved	保留	R	0
7:5	PLL_FREQ	PLL 频点固定为 16.384Mhz, 通过数字分频实现频率选择 000: PLL 倍频选择为 8.196MHz; 001: PLL 倍频选择为 16.384Mhz; 010: PLL 倍频选择为 32.768MHz; 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: 外接高频晶振为 8.192MHz 001: 外接高频晶振为 16.384MHz 010: 外接高频晶振为 32.768MHz 011: 外接高频晶振为 32.768MHz (驱动增强) 其他保留	R/W	000

系统时钟配置真值表:

	HOSC_DIV=00 不分频	HOSC_DIV=01 二分频	HOSC_DIV=10 四分频	HOSC_DIV=11 八分频
HOSC_FREQ =000	8.192Mhz	4.096MHz	2.048MHz	不支持八分频, 如配置则为 2.048Mhz
HOSC_FREQ =001	16.384Mhz	8.192Mhz	4.096Mhz	2.048Mhz
HOSC_FREQ =010	32.768Mhz	16.384Mhz	8.192Mhz	4.096Mhz

2.7.7 SYS_RST(0x20)

系统复位寄存器: (此寄存器的 10 到 3 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
31:16	RST_PROT	EDT、EMU、EMU-REG 以及 FLK 模块软复位	W	0

		保护位，密码是 0xC259。 只有当此寄存器为 0xC259 时，bit4 以及 bit[2:0] 软复位配置才会生效。 该寄存器值为只写寄存器，读出为 0；		
15:11	---	预留	R	0
10	---	预留	R/W	0
9	---	预留	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	FLK_RST_REQ	FLK 模块寄存器和计算模块软件复位 写入 1 复位 FLK 模块; 写入 0 取消 FLK 模块复位。 复位 FLK 模块配置寄存器和计算模块; 注意: EDT_RST_REQ 写 1 会一直复位模块, 直到写 0 才结束复位	R/W	0
3	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
2	EMUREG_RST_REQ	软件复位 EMU(包括 EMU, NVM)配置寄存器 模块: 写入 1 复位 EMU 配置寄存器; 写入 0 取消 EMU 配置寄存器; 不复位 EMU 模块计算模块和结果寄存器; 注意: EMUREG_RST_REQ 写 1 会一直复位目 标, 直到写 0 才结束复位	R/W	0
1	EDT_RST_REQ	软件复位 EDT 计算模块: 写入 1 复位 EDT 模块; 写入 0 取消 EDT 模块复位。 备注 1: 不复位 EDT 模块配置寄存器, 只复位 运算状态机以及计算结果寄存器;	R/W	0

		备注 2: EDT_RST_REQ 写 1 会一直复位模块, 直到写 0 才结束复位		
0	EMU_RST_REQ	软件复位 EMU(包括 EMU, NVM)计算模块: 写入 1 复位 EMU 计算模块; 写入 0 取消 EMU 计算模块复位。 备注 1: 不复位 EMU 模块配置寄存器, 只复位运算状态机以及计算结果寄存器; 备注 2: EMU_RST_REQ 写 1 会一直复位模块, 直到写 0 才结束复位	R/W	0

2.7.8 SYS_MAPCTL(0x24)

系统映射控制寄存器:

比特位	名称	描述	读 / 写标志	复位值
31:3	---	预留	R	0
2:0	REMAP	地址映射: 000: FLASH 映射在 0 地址 (正常模式) 010: FLASH 与 SRAM 映射地址互换 011: 保留 (从 Bootrom 运行) 100: 保留 其他: 等同于 000.	R/W	00

备注: 该寄存器在复位发生后实际复位值为 11, Bootrom 运行后会更改为 00.
 用户看到的复位值是 00.

2.7.9 MOD0_EN(0x28)

模块使能 0 寄存器:

比特位	名称	描述	读 / 写标志	复位值
31:20	---	预留	R	0
19	SPI4_EN	SPI4 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
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	SPIS_EN	SPIS 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:	R/W	0

		0: 时钟停止, 模块清零 1: 时钟启动, 模块使能		
15	SPIO_EN	SPIO 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	保留位	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	保留位	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: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
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
3:1	---	预留	R	0

0	---	该位为内部寄存器位，用户不要修改默认值	R/W	0
---	-----	---------------------	-----	---

2.7.10 MOD1_EN(0x2C)

模块使能 1 寄存器：

比特位	名称	描述	读 / 写标志	复位值
31:18	---	保留	R	0
17	FD2F_EN	预留	R/W	0
16	FLK_EN	闪变模块使能，时钟门控 0：时钟停止 1：时钟启动 注意：配置 FLKEN_CTL 寄存器为 0xec0000e5，该位有效。	R/W	0
15	GPADC_EN	GPADC 模块使能 0：时钟停止 1：时钟启动	R/W	0
14	---	保留位，默认为 0，不要写 1	R/W	0
13	---	保留位，默认为 0，不要写 1	R/W	0
12	EDT_EN	EDT 模块使能清零，时钟门控： 0：时钟停止 1：时钟启动	R/W	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	VBAT_EN	VBAT 域 RTC/SAR/GPIO 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	0
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	0
8	NVM_EN	全失压 NVM 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
7	EMU_EN	计量 EMU 模块使能清零，时钟门控： 0：时钟停止，模块清零 1：时钟启动，模块使能	R/W	0
6	FFT_EN	谐波计算模块使能清零，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	GPIO_EN	VCC 域 GPIO (AHB) 模块使能清零，时钟门控，cm0	R/W	0

		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		
4:0	Reserved	保留	R	0

2.7.11 INTC_EN(0x30)

INTC 使能寄存器:

比特位	名称	描述	读 / 写标志	复位值
31:9	---	保留	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

2.7.12 KBI_EN(0x34)

KBI 使能寄存器:

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

31:9	---	保留	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

2.7.13 SYS_PS(0x3C)

系统控制密码寄存器 SYS_PS:

比特位	名称	描述	读 / 写标志	复位值
31:8	---	预留	R	0
7:0	SYS_PSW	当 SYS_PSW=8'h82 时, 0x00~0x40、0xA0/0xA4、0xC0/0xC4 寄存器可写; 当 SYS_PSW=其他值时, 0x00~0x40、0xA0/0xA4、0xC0/0xC4 寄存器不可写; 该寄存器读出值为写入的值。 建议用户在写操作完成后马上关闭写使能。	R/W	00

2.7.14 IRFR_CTL (0x40)

红外配置寄存器：

比特位	名称	描述	读 / 写标志	复位值
31:6	---	预留	R	0
5:0	INFRARED_CYCLE	该寄存器的密码是 0x82 当系统时钟源来自在外部高频晶体或 32768+PLL 时，会根据不同的系统主频自适应先分频到 2.048M，内部分频系数采用 0x1A，红外 38K 分频是固定的； RCH 模式下，红外时钟分频系数，基于 RCH 为 2.9MHZ。RCH 模式下，计算的公式为： $uart38k = fsys / (2 * (INFRARED_CYCLE + 1))$ ，其中 fsys 为系统时钟，fsys 为 RCH 或者 RCH 的二分频。	R/W	0x18

2.7.15 TRIM_CFG1(0xA0)

时钟校正配置寄存器：

比特位	名称	描述	读 / 写标志	复位值
31:30	---	预留	R	0
30	CAL_RC32M_SEL	该 bit 与 bit29 组成[30:29]，用来选择被测量时钟源 00，选择 RCH_VCC_CLK 作为被测量时钟； 01，选择 RCH_VBAT_CLK 作为被测量时钟 1x，选择 RC32M_CLK 作为被测量时钟	R/W	0
29	RCH_VCC_VBAT_SEL	选择被测量 RCH 时钟来源： 0：为选择 RCH_VCC_CLK 作为被测量时钟； 1：为选择 RCH_VBAT_CLK 作为被测量时钟	R/W	0
28	CAL_OV_IE	被校准时钟计数器溢出标志中断使能： 0：不使能中断； 1：使能中断；	R/W	0
27	CAL_DONE_IE	时钟校准完成标志中断使能： 0：不使能中断； 1：使能中断；	R/W	0
26	CAL_CLK_SEL	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；	R/W	11

		11: 低功耗模式;		
23:20	---	预留	R	0
19:0	REF_CLK_CNT [19:0]	参考时钟计数值	R/W	0x1000 0

2.7.16 TRIM_START(0xA4)

时钟校正启动寄存器:

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

举例:

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

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

2.7.17 DMA_PRI 0xC0

比特	名称	描述	读 / 写	复位
----	----	----	-------	----

位			标志	值
31:24	WKEY	WKEY 是 0~19bit 的写操作密码保护位, 密码是 0xEC。软件写 0~19bit 时必须保证同时写入的数据高 8 位为 0xEC 并且将 SYS_PSW 为 8'h82。	R/W	0
23:20	保留	保留	R/W	0x0
19:18	DMA_CH9_PRI	通道 9:CPU 访问 SRAM 优先级配置	R/W	0x0
17:16	DMA_CH8_PRI	通道 8:FFT DMA 访问 SRAM 优先级配置	R/W	0x0
15:14	DMA_CH7_PRI	通道 7:SPI4 访问 SRAM 优先级配置	R/W	0x0
13:12	DMA_CH6_PRI	通道 6:SPI3 访问 SRAM 优先级配置	R/W	0x0
11:10	DMA_CH5_PRI	通道 5:SPIS 访问 SRAM 优先级配置	R/W	0x0
9:8	DMA_CH4_PRI	通道 4:UART2 DMA 访问 SRAM 优先级配置	R/W	0x0
7:6	DMA_CH3_PRI	通道 3:SPI1 DMA 访问 SRAM 优先级配置	R/W	0x0
5:4	DMA_CH2_PRI	通道 2:SPI0 DMA 访问 SRAM 优先级配置	R/W	0x0
3:2	DMA_CH1_PRI	通道 1:MMU 波形缓存 DMA 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
1:0	DMA_CH0_PRI	通道 0:Cache 访问 SRAM 的优先级配置。固定最高优先级(0x3)	R	0x3

备注:

系统上一共有 10 个主机访问 SRAM, 分别是 Cache 控制器、MMU 波形缓存 DMA、SPI0-DMA、SPI1-DMA、UART-DMA、SPIS-DMA、SPI3-DMA、SPI4-DMA、FFT-DMA 和 ARM-M0 CPU;

当不同主机访问相同 SRAM 时, 会产生竞争关系, 系统根据优先级仲裁, 优先级高主机通道获得该 SRAM 的访问权限,

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。

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。

若优先级配置值相同, 则比较通道号, 通道号小的优先级高。

2.7.18 DMA_RST 0xC4

比特位	名称	描述	读/写标志	复位值
31:16	DMARST_PROT	DMA 软复位密钥, 密码是 0xC259。 只有当 DMARST_PROT ==0xC259 时, bit[7:0]软复位配置才会生效。 该寄存器值不可读, 读到都是 0;	W	0
15:8	---	保留	R	0
7	dma_fft_rst_req	FFT DMA 状态机软复位 (专用 DMA)	R/W	0
6	dma_uart_rst_req	UART2 DMA 状态机软复位 (普通 DMA)	R/W	0
5	dma_spi4_rst_req	SPI4 DMA 状态机软复位 (普通 DMA)	R/W	0
4	dma_spi3_rst_req	SPI3 DMA 状态机软复位 (普通 DMA)	R/W	0
3	dma_spi2_rst_req	SPI2 DMA 状态机软复位 (普通 DMA)	R/W	0
2	dma_spi1_rst_req	SPI1 DMA 状态机软复位 (普通 DMA)	R/W	0

1	dma_spi0_rst_req	SPI0 DMA 状态机软复位（专用 DMA）	R/W	0
0	dma_mmu_rst_req	MMU 波形缓存 DMA 状态机软复位(专用 DMA) 注意： 所有的 DMA_RST_REQ 写 1 会一直复位目标，直到写 0 才结束复位	R/W	0

注：DMA_RST 功能只对状态机进行复位，在 DMA_RST 复位释放后，需要重新使能 DMA 才能正常运行 DMA 功能。

3 处理器架构

3.1 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。

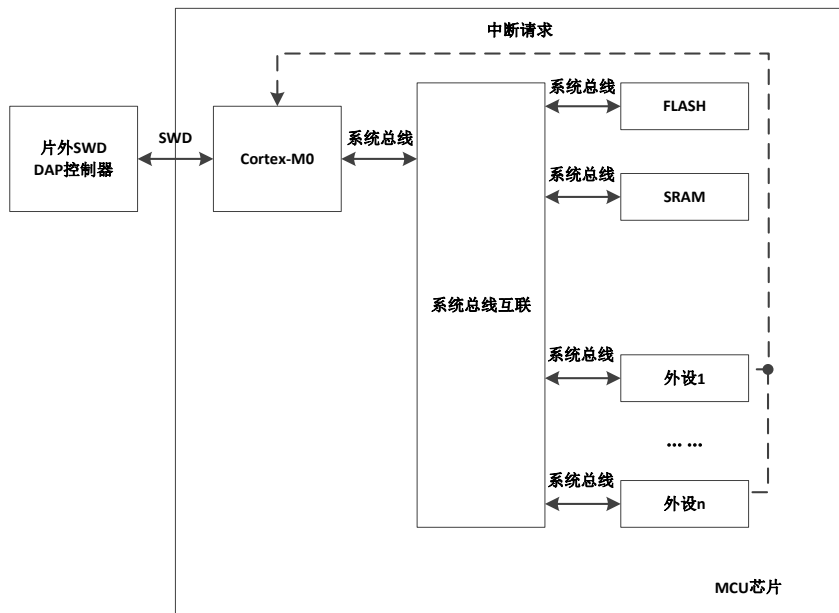


图 3-1 MCU 设备物理互联架构

3.2 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。

3.2.1 中断配置

支持 32 个中断，其中 8 个外部中断引脚合用中断号 31。

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 3-1 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~-7	保留	0x10~0x24	-	
11	-5	系统调用	0x2C	可配置	与处理器同步
12	-4	保留			
13	-3	保留			
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、LVD	0x44	可配置	与处理器异步
18	2	GPADC	0x48	可配置	与处理器同步
19	3	RTC	0x4C	可配置	与处理器异步
20	4	计量 EMU 中断 2	0x50	可配置	与处理器异步
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	保留	0x70	可配置	与处理器异步
29	13	保留	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	SPI1	0x90	可配置	与处理器异步
37	21	SYS_TC	0x94	可配置	与处理器异步
38	22	谐波 FFT	0x98	可配置	与处理器异步

39	23	NVM 全失压	0x9C	可配置	与处理器异步
40	24	计量 EMU 中断 1	0xA0	可配置	与处理器异步
41	25	EDT	0xA4	可配置	与处理器异步
42	26	FLK	0xA8	可配置	与处理器异步
43	27	SPIS	0xAC	可配置	与处理器异步
44	28	SPI3	0xB0	可配置	与处理器异步
45	29	SPI4	0xB4	可配置	与处理器异步
46	30	保留	0xB8	可配置	与处理器异步
47	31	外部中断	0xBC	可配置	与处理器异步

3.3 MCU 存储映射

MCU 的存储映射请参考“图 3-1”。

SoC 的高速外设包括：

- GPIO-AHB;
- 计量 EMU 单元;
- 1 个高速 SPIS 接口;
- SYSTC 系统定时器
- FLASH 控制器
- EDT 单元;

SoC 的低速外设包括：

- 6 个 UART 接口;
- 2 个 32 位定时器;
- 4 个 SPI 接口;
- 1 个 I2C 接口;
- 1 个 KEY 控制器;
- 模拟外设+GPIO-APB;
- 看门狗单元;
- 系统控制单元;
- 1 个 GPADC
- 1 个 RTC 接口;
- 全失压 NVM 单元;
- 1 个外部中断控制器;

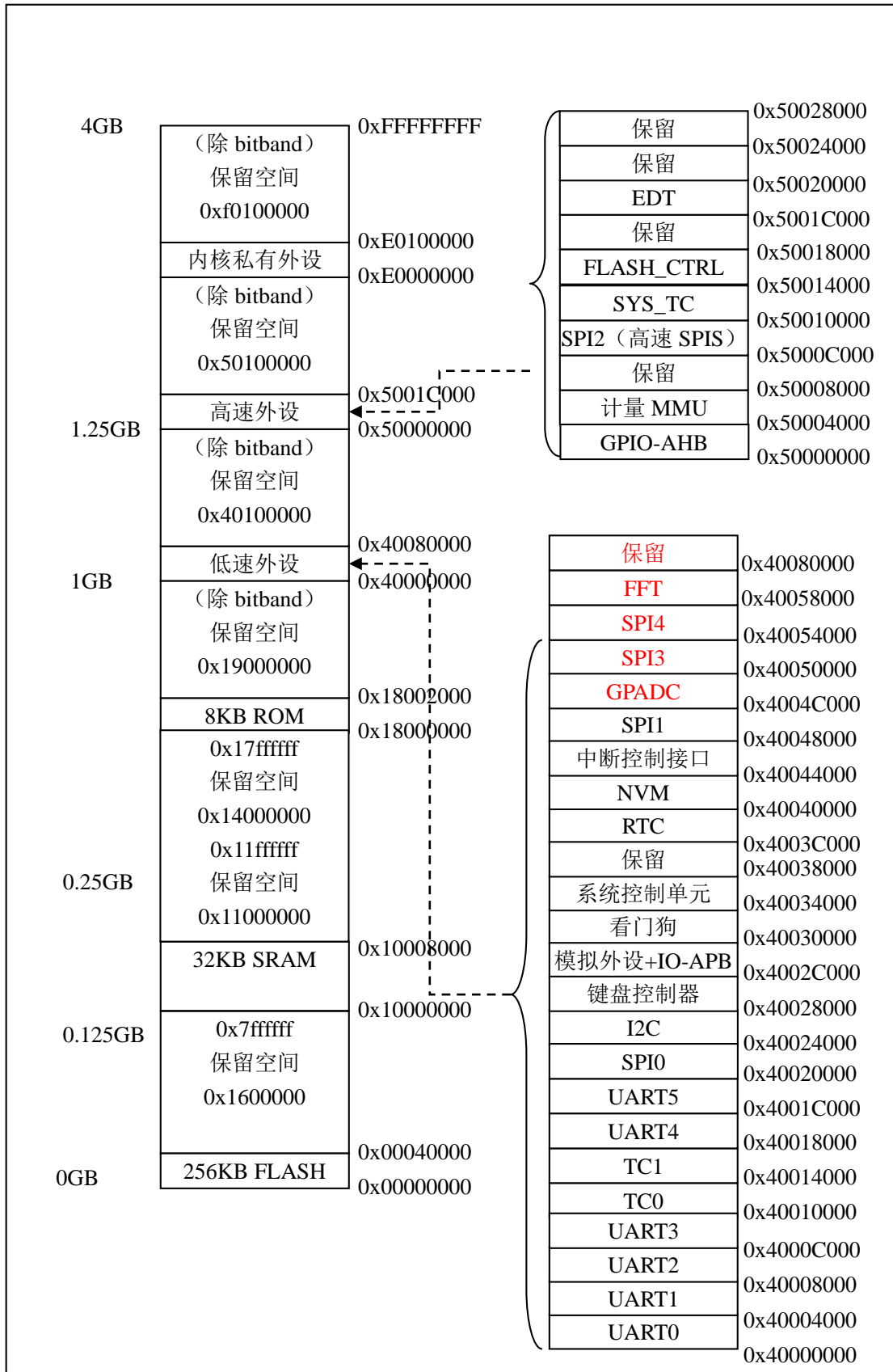


图 3-1 MCU 存储映射

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 3-2 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

● bitband 功能

系统支持以下地址空间的 bitband 功能：

- 0x10000000~0x10007FFF 映射到 0x12000000~0x120FFFFF
- 0x40000000~0x4004FFFF 映射到 0x42000000~0x423FFFFF；
- 0x50000000~0x50007FFF 映射到 0x52000000~0x5203FFFF；

对 bitband 区的访问等效于对外设寄存器中特定位的访问。

地址为 x 的存储单元的第 y 位对应的 bitband 地址：

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

注意：地址 0x40000000~0x4004FFFF 包括了部分 vbat 域相关的模块，有 RTC、模拟外设、GPIO-APB 模块，这些模块的寄存器不支持 bitband 操作方式。

3.3.1 SRAM

片内静态 RAM (SRAM)，容量为 32KB（其中 2KB 可做为 CACHE 使用），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；32KB RAM 位于高位地址的 2KB 可作为 CACHE 用，当做 CACHE 用时，CPU 不要访问。

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 使用了地址空间地址空间 0x1000_0300~0x1000_03ef，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，BOOT 启动完成后，该地址空间又被释放，使用此地址空间请注意该特性。

3.3.2 FLASH

片内 256KB FLASH，支持：

- 最少 10 万擦写次数；
- 数据最少保存时间 10 年；
- 存储区包含 16 个块，每个块包含 64 个页，每页 256 字节；
- 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)

➤ 特殊行说明

Flash 包含一段特殊的存储空间，用来存储量产参数。该特殊空间的地址为：0x00040000~0x000403ff，共分 4 个特殊行，每行 256 字节。分别为：

Spl0: 0x00040000~0x000400ff;

Spl1: 0x00040100~0x000401ff;

Spl2: 0x00040200~0x000402ff;

Spl3: 0x00040300~0x000403ff; 该特殊行不可擦除和编程。

其中，0x00040300~0x00040307 地址空间，对应的总共 2 个 word 数据，是 64bit 的芯片**唯一标识号**，每颗芯片出厂值均不相同。

3.4 中断应用

在头文件中加入 SOC 头文件 #include <RN2025.h>, 即可使用 SOC 各中断, RN2025.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 RN2025.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_RN2025.s 的向量表中查找，如 KBI 中断服务程序函数名为 KBI_HANDLER，中断服务函数可写为：

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

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

4 单相计量单元

4.1 特点

- 基本计量
 - 3路 Σ - Δ ADC，内置 PGA，放大倍数可配置，其中 IA/IB 通道最大支持 16 倍；
 - 内置 $1.25V \pm 1\%$ ADC 基准电压，温度系数典型值 $10\text{ppm}/^\circ\text{C}$ ，最大 $15\text{ppm}/^\circ\text{C}$ ；
 - 支持外灌基准电压，当外灌基准电压时，需要调用锐能微库函数完成相关配置。
 - 提供全波、基波有功电能，8000:1 动态范围内，非线性误差 $<0.1\%$ ，满足 0.5S 和 0.2S 级有功电能表精度要求
 - 提供基波无功电能，8000:1 动态范围内，非线性误差 $<0.1\%$
 - 提供视在电能
 - 提供有功、无功功率方向，支持无功四象限判断
 - 具有潜动启动功能，电流和功率启动方式可选，启动阈值可调
 - 电表常数可调
 - 提供有功、无功、视在的快速脉冲计数
 - 提供 5 路能量脉冲输出，脉冲类型为 A 路/B 路、全波/基波、有功/无功/视在各种组合可选，同时也可以输出自定义电能脉冲。
- 基本测量
 - 提供 3 路 ADC 瞬时采样数据寄存器
 - 提供全波、基波三相电压电流有效值
 - 提供全波/基波的有功/视在功率，提供基波无功功率
 - 提供 3 路过零检测，过零阈值可设置
 - 提供电压线频率，测量误差 $<0.01\%$
- 故障检测
 - 提供过压、过流检测
 - 提供剩余电流检测
 - 提供电压暂降检测
- 同步采样波形输出
 - 提供灵活的 ADC 同步采样数据，可配置采样率；
 - 支持灵活的帧格式，支持硬件自动计算校验和；
 - 支持灵活的校正手段，支持增益校正、相位校正、谐波补偿；
 - 支持通过 DMA 方式将波形数据从计量传输到 RAM，以及从 RAM 传输到通信口。
- 电能质量
 - 支持完整的 S 级电能质量分析功能
 - 提供半波有效值、半波电压峰值，新增零线电流半波有效值，新增开根号后的半波有效值
 - 完整的电能质量分析功能包括：谐波、间谐波、闪变和电压波动、电压骤升骤降、电压中断等。
- 高级计量功能
 - 支持基波有功、无功、视在电能计量，可从 CF0~CF4 任意一个 IO 输出任意一个基波电能；
 - 支持正负总谐波电能计量，正向总谐波电能以及反向总谐波电能可同时从 CF0~CF4 任意 IO 口输出。
 - 谐波电能计算由硬件加速引擎完成，可实现不间断连续计算。

- 可支持分次谐波电能计量；
- 防窃电
 - 提供全失压测量
- 软件校表
 - 提供 3 路 ADC 通道增益校正
 - 提供 3 路 ADC 通道相位校正，其中电流通道支持分段相位校正，校正刻度从 0.01757 度提升至 0.00879 度。
 - 提供功率增益校正
 - 提供有功、无功功率分段相位校正；另外单独做了一套功率相位分段校正寄存器，可与 D2FPxx 寄存器独立使用。
 - 提供有功、无功、有效值 Offset 校正，其中有功 Offset 校正扩大了校正范围。
 - 提供直流 offset 自动校正，校正节点从高通滤波器之后移到了高通滤波器之前。
 - 增加配置寄存器校验和自动计算功能；
- 保证计量精度的工作电压范围：2.8V-5.5V

4.2 系统框图

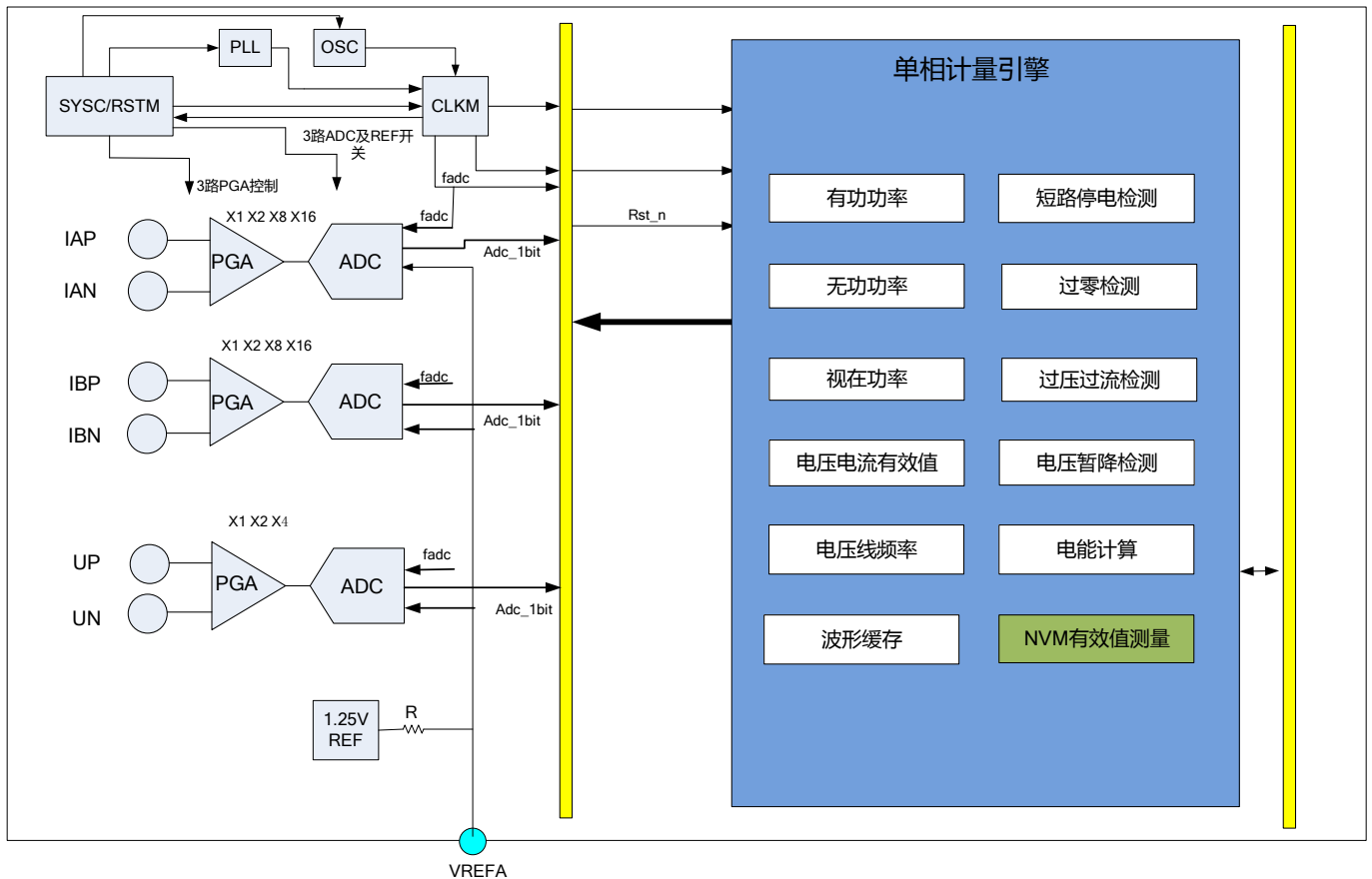


图 4-1 单相计量引擎框图

4.3 功能说明

4.3.1 采样通道

RN2025 采样通道包括三路 Σ - Δ ADC 及其采样数据处理电路。

三路 ADC 其中 IA 用于火线电流采样, IB 用于零线电流采样, U 用于电压采样。ADC 采用全差分方式输入, 电流、电压通道最大差分信号输入峰值 V_{pp} 为 $\pm 1000\text{mv}$ 。

通过配置 ADCCTL 寄存器, 可以分别对三路 ADC 配置 PGA 放大倍数, 电流 1/2/8/16 倍可选, 电压通道 1/2/4 倍可选, ADCCTL 寄存器使用说明参见 [2.7.4 章节](#)。

通过配置 PHSU/I 寄存器可进行三路 ADC 的通道相位校正, PHSU/I 寄存器使用说明参见 [4.5.13 章节](#)。

通过配置 GSU/I 寄存器可进行三路 ADC 的通道增益校正, GSU/I 寄存器使用说明参见 [4.5.14 章节](#)。

通过配置 EMUCFG 寄存器(0x188)中的 HPFOFF [6:0]寄存器位可以配置三路采样通道的高通使能和关闭, 用于测试直流参数。HPFOFF [6:0]具体说明参见寄存器 [4.5.24 章节](#)。

通过配置 AUTODC_EN 寄存器 (0x1B8) 自动计算 DC_OS 寄存器值可进行三路 ADC 的直流 OFFSET 校正, AUTODC_EN 寄存器具体说明参见 [4.5.34 章节](#), DC_OS 寄存器具体说明参见 [4.5.15 章节](#)。

RN2025 的三路 ADC 实时采样数据分为全波数据和基波数据, 可输出到波形采样寄存器中, 数据源为 ADC 经过高通后的数据。波形采样寄存器具体说明参见 [4.5.2 章节](#)。当波形采样寄存器更新数据时, 标志位 EMUIF.WAVUDIF 置位, 若波形采样寄存器更新中断 EMUIE.WAVUDIE 使能, EMUIF.WAVUDIF 置位产生 EMU 中断通知 CPU 读取数据。

4.3.2 ADC 波形缓存及 DMA

三路 24bits ADC 采样数据通过专用 DMA 通道可存放至 SRAM 中, 以减小 CPU 读写开销, 方便 CPU 进一步计算分析。

1. 配置 ADC 波形缓存数据源

通过配置 [DMA_WAVE_CH_SEL](#) 寄存器 (0x2AC) 选择瞬时波形缓存 ADC 通道, IA/IB/U 三路 ADC 可任意配置。通过配置 [DMA_WAVECON](#) (0x2B4) 寄存器进一步配置 ADC 瞬时波形缓存数据源:

可选高通滤波器 HPF 之前固定 8Khz 采样率数据: 此数据经过通道增益和相位校正, 可用于直流参数计算;

可选高通滤波器 HPF 之后固定 8Khz 采样率数据: 此数据经过通道增益和相位校正, 可用于电能质量等参数计算;

可选同步跟踪电网频率整周波采样数据: 此数据未经过通道增益和相位校正环节, 采样率整周波 64/128/256/512 点可选, 可同步跟踪电网频率, 同步跟踪电网频率范围: 40~70Hz, 可用于 FFT 分析及谐波含量计算。

可选同步跟踪电网频率整周波采样数据, 采样率通过 WAVECON.WAVECNT[8:0]寄存器位配置: 此数据未经过通道增益和相位校正。

2. 数据缓存深度、地址和排序配置

ADC 数据缓存数据缓存深度由 DMA_BUF_DEPTH[10:0]寄存器 (0x2A8) 配置, 深度为 (N+1) (Word); 最小 128 Words, 最大 2K Words。ADC 数据缓存目标地址 DMA 功能寄存器的目标地址是 SRAM 的地址, 按 Word 编址, 共 13 位。

目标地址基地址由 DMA_BUF_BASE_ADDR[12:0] (0x2A4) 寄存器配置, 注意基地址要与数据缓存深度要匹配, 如 DMA_depth 为 2K Words, 那么 DMA_BUF_BASE_ADDR[10:7]==4'b0000; [12:11]任意可配; 若深度为 128, 那么[12:7]任意可配; ADC 数据缓存基址默认值 0xC00。

ADC 数据每次采样在数据缓存中的通道排序由 DMA_WAVE_CH_SEL[6:0]寄存器(0x2AC)使能的通道决定, 按 IA/IB/U 顺序排序, 若某通道不使能, 则该通道忽略。

3. DMA 操作控制、中断

DMA 操作模式可通过 DMA_BUF_CTRL[1] 寄存器位 (0x2A0.1) 配置为单次模式或连续模式,

DMA_BUF_CTRL[1]: =0, 单次模式, 写操作数据缓存满之后则不再发起 DMA 操作; =1, 连续模式, 写操作数据缓存满, 折返到起始地址重新开始。

通过配置 DMA_BUF_CTRL[0] (0x2A0[0]) 使能数据缓存; =1, 使能; =0, 关闭。

数据缓存满或半满可产生中断, 通知 CPU 取走数据。数字缓存满时 EMUIE2.DMA_BUFF_IF (0x2BC[15]) 标志置位, 此时若 EMUIE2.DMA_BUFF_IE=1 (0x2B8[15]), 产生数据缓存满中断。数字缓存半满时 EMUIE2.DMA_BUHF_IF (0x2B8[16]) 置位, 此时若 EMUIE2.DMA_BUHF_IE=1 (0x2B8[16]), 产生数据缓存半满中断。

4. DMA 容错处理

在 ADC 采样间隔内, 若有任何一个通道 DMA 操作没有响应, 则放弃本轮所有操作, 记录数据错误发生地址, 发出 DMA 错误中断; 然后发起新一轮操作。

EMUIF2.DMA_BUFIAErr/DMA_BUFUxErr/DMA/DMA_BUFINErr(x=A,B,C)寄存器位 (0x2BC[23:17]) 为七路 ADC 通道 DMA 数据错误标志位, 当相应通道发生 DMA 数据错误时置位, 若此时相应通道 DMA 错误中断使能位 EMUIE2.DMA_BUFIxErr/DMA_BUFUxErr (0x2B8[23:17]) =1, 产生 DMA 错误中断, 并记录数据错误发生地址放置在 DMA_WAVE_ERR_ADDR [12:0] (0x2B0) 寄存器。

DMA 控制和状态寄存器具体说明参见 4.5.41 章节。DMA 中断寄存器具体说明参见 4.5.40 章节。

RN2025 (B64) 对于 ADC 波形缓存及 DMA 做了大幅度的升级优化, 升级点主要是:

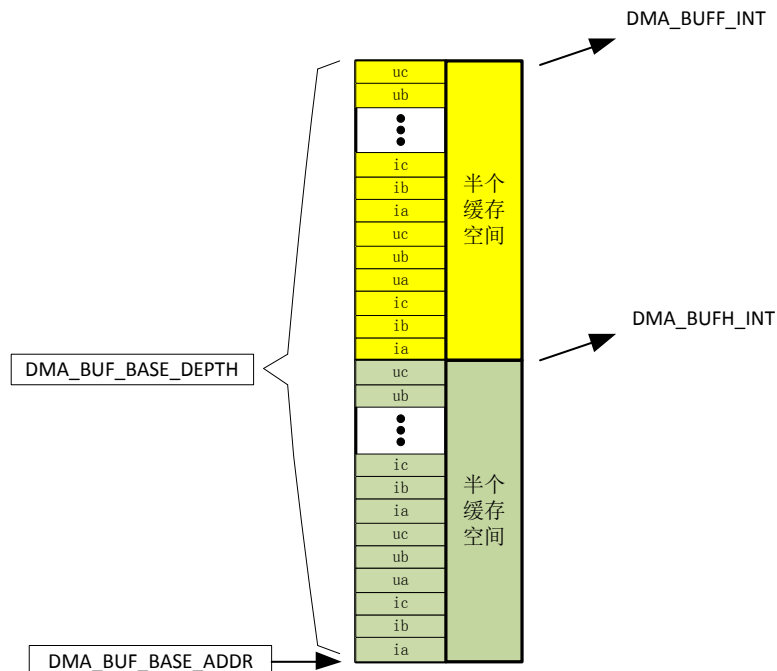
- 1) 新增同步采样波形的全方位校正, 包括增益校正、相位校正、直流偏置校正、增益系数补偿、谐波衰减补偿等等, 详见 4.5.13 章节。
- 2) 对 DMA 做了优化提升, 一共有三种模式, 其中模式 0 与 RN2025 (A64) 兼容, 新增模式 1 和模式 2。

下文对 DMA 的模式做详细说明:

4.3.2.1 模式 0: 不同通道连续存放 (兼容 A 版芯片)

按 IA/IB/U 顺序按 word 在 SRAM 种存放波形数据, 若某几个通道不使能, 则该通道忽略, 其他通道按顺序连续存放放到 SRAM 中。

波形存放示意图:



1) 配置 ADC 波形缓存数据源

通过配置 DMA_MODE_SEL [1:0]=0 选择模式 0。

通过配置 DMA_WAVE_CH_SEL 寄存器选择瞬时波形缓存 ADC 通道，IA/IB/U 三路 ADC 可任意配置。

ADC 数据每次采样在数据缓存中的通道排序由 DMA_WAVE_CH_SEL[6:0]寄存器（0x2AC）使能的通道决定，按 IA/IB/U 顺序排序，若某通道不使能，则该通道忽略。

2) 数据缓存地址和深度

ADC 缓存目标基地址由 DMA_BUF_BASE_ADDR[13:0]（0x2A4）寄存器配置，按 Word 编址，注意基地址要与数据缓存深度匹配。缓存深度由 DMA_BUF_DEPTH[13:0]寄存器（0x2A8）配置，深度为 (N+1) (Word)；最小 128Words，最大 16KWords。详见寄存器配置。

3) DMA 中断配置

DMA 操作模式可通过 DMA_BUF_CTRL[1] 寄存器位（0x2A0.1）配置为单次模式或连续模式，DMA_BUF_CTRL[1]: =0, 单次模式，写操作数据缓存满之后则不再发起 DMA 操作；=1, 连续模式，写操作数据缓存满，折返到起始地址重新开始。

通过配置 DMA_BUF_CTRL[0]（0x2A0[0]）使能数据缓存；=1, 使能；=0, 关闭。

数据缓存满或半满可产生中断，通知 CPU 取走数据。数字缓存满时 EMUIE2.DMA_BUFF_IF（0x2BC[15]）标志置位，此时若 EMUIE2.DMA_BUFF_IE=1（0x2B8[15]），产生数据缓存满中断。数字缓存半满时 EMUIE2.DMA_BUHF_IF（0x2B8[16]）置位，此时若 EMUIE2.DMA_BUHF_IE=1（0x2B8[16]），产生数据缓存半满中断。

4) DMA 容错处理

在 ADC 采样间隔内，若有任何一个通道 DMA 操作没有响应，则放弃本轮所有操作，记录数据错误发生地址，发出 DMA 错误中断；然后发起新一轮操作。

EMUIF2.DMA_BUFIAErr/DMA_BUFUxErr/DMA/DMA_BUFINErr(x=A,B,C)寄存器位（0x2BC[23:17]）为七路 ADC 通道 DMA 数据错误标志位，当相应通道发生 DMA 数据错误时置位，若此时相应通道 DMA 错误中断使能位 EMUIE2.DMA_BUFIxErr/DMA_BUFUxErr（0x2B8[23:17]）=1,产生 DMA 错误中断，并记录数据错误发生地址放置在 DMA_WAVE_ERR_ADDR[12:0]（0x2B0）寄存器。

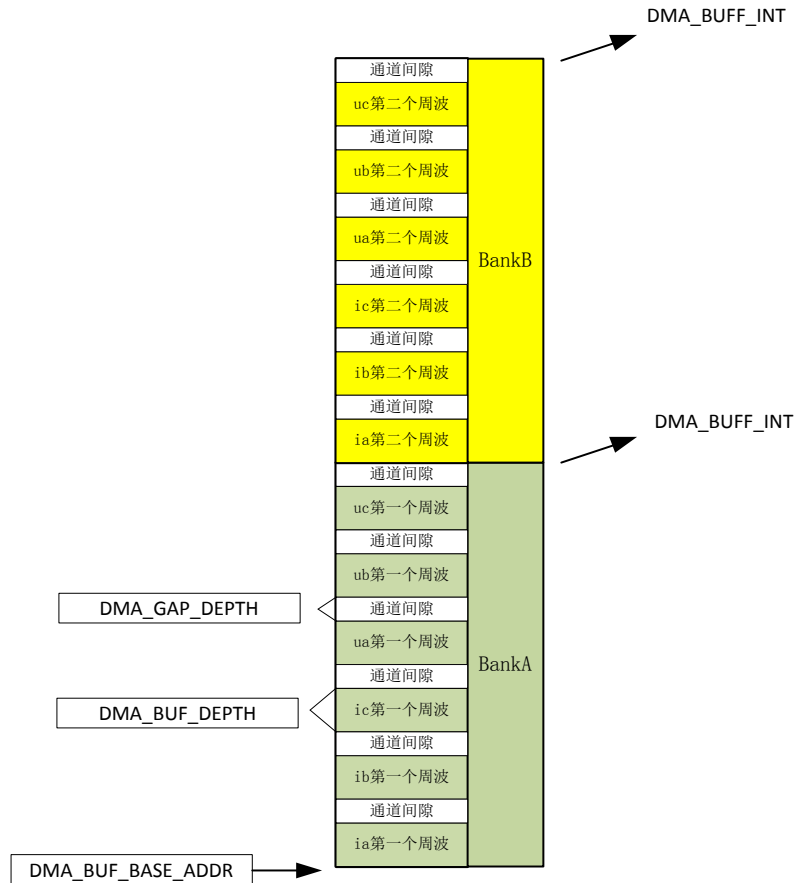
5) DMA 当前指针可读

CPU 可通过读取 DMA_BUFF_ADDR 获取当前波形缓存的 DMA 指针。

4.3.2.2 模式 1：不同通道分块存放

不同通道按一个半波或者一个周波（大小可通过 DMA_BUF_DEPTH 任意配置）存放在一起，并且两个通道之间可以插入空白的 SRAM 区域，用于后面软件组帧。

1) 模式 1 波形存放示意图



2) 模式 1 特点

- 波形缓存 (WaveBuff) 由周波区块 (Bank) 构成, 周波区块的个数表示存取多少个连续周波数。周波区块个数可配置 (1~16 可配置), 如上图中的 BankA、BankB;
- 如果 DMA 配置成连续模式, 假配置了 2 个周波区块, 那么波形缓存的存取顺序为 A->B->A->B->A...;
- 周波区块内包含一个周波的不同通道的波形数据; 同一通道单周波数据连续放在一起, 比如 IA|IB|IC|UA|UB|UC|IN。7 个通道数据是否缓存可以任意使能, 存放排序也可任意配置。
- 波形缓存起始地址可任意配置, 通道深度可任意配置;
- 支持两个通道之间可以插入空白, 称为间隙区, 间隙区大小任意配置; 每个通道后是否插入间隙区, 也可独立配置; 比如 IA 后不插入, IB 后插入;
- 波形缓存深度由周波区块个数、通道深度和通道间隙深度决定;
- 支持在波形缓存过程中, 自动算校验和, 每个通道数据算一个校验和, 校验和结果存到寄存器, CPU 可读。如上图中 “ia 第一个周波” 算一个校验和, “ib 第一个周波” 算一个校验和。

3) 配置 ADC 波形缓存模式

通过配置 DMA_MODE_SEL [1:0]=2'b01 选择 “模式 1”。

DMA 操作模式可通过 DMA_BUF_CTRL[1] 寄存器位 (0x2A0.1) 配置为单次模式或连续模式, DMA_BUF_CTRL[1]=0, 单次模式, 写操作数据缓存满之后则不再发起 DMA 操作; =1, 连续模式, 写操作数据缓存满, 折返到起始地址重新开始。

4) 缓存数据通道选择和顺序

ADC 每次采样数据在缓存中按通道整波存放，相同通道数据连续存放一块。

通过配置 DMA_WAVE_CH_SEL 寄存器选择瞬时波形缓存 ADC 通道，IA/IB/IC/UA/UB/UC/IN 七路 ADC 可任意配置。

数据在数据缓存中的通道排序可任意配置，由 DMA_WAVE_CH_SEL[6:0]寄存器（0x2AC）使能的通道和 DMA_WAVE_CH_SEL[27:7]寄存器配置的通道顺序决定，由软件保证配置的正确性。如 DMA_WAVE_CH_SEL[6:0]=0x5A，IA/IC/UA/UC 一共 4 个通道使能，IB/UB/IN 不使能；然后配置 DMA_WAVE_CH_SEL[27:7]寄存器的 CH0_SEL=0x0（第 0 个通道选择 IA），CH1_SEL=0x3（第 1 个通道选择 UA），CH2_SEL=0x2（第 2 个通道选择 IC），CH3_SEL=0x5（第 3 个通道选择 UC），则每次采样在 SRAM 数据缓存中的排序为 IA/IC/UA/UC。（注意此时 CH0/1/2/3_SEL 只能选择 DMA_WAVE_CH_SEL 中使能的通道）。

5) 波形缓存地址和深度

ADC 波形缓存目标基地址由 DMA_BUF_BASE_ADDR[13:0]寄存器配置，按 Word 编址，支持任意地址。每一个通道缓存深度由 DMA_BUF_DEPTH[13:0]寄存器配置，深度为 (N+1) (Word)，任意配置，每一个通道深度相同。

通道有效数据后面可插入通道间隙，大小由 DMA_GAP_DEPTH 寄存器配置，每个通道间隙大小相同。每个通道后是否插入通道间隙，由 DMA_GAP_CH_SEL 独立配置，比如可选择 IA 后不插入通道间隙，IB 后插入通道间隙。当 DMA_GAP_CH_SEL 对应通道配置为 0，表示该通道后不插入空白区，当 DMA_GAP_DEPTH 配置为 0，表示所有通道后不插入空白区。

把所有通道存满 DMA_BUF_DEPTH 大小的区域称为一个周波区块，通过配置 DMA_BANK_CNT 配置周波区块的个数， $n=(n+1)$ 个周波区块，波形在几个周波区块中循环存储。比如配置为 3 表示开 3 个周波区块 A/B/C，则 DMA 首先把数据存到 A 区，A 区存满之后存到 B 区，B 区存满之后存到 C 区，C 区存满又覆盖存放 A 区。

占用 SRAM 大小 (word) = (每个通道缓存深度 $a * n$ 个通道 + 每个通道间隙大小 $b * m$ 个通道间隙) * k 个周波区块

上式中：

$$a = \text{DMA_BUF_DEPTH}$$

$$b = \text{DMA_GAP_DEPTH}$$

$$n = \text{DMA_WAVE_CH_SEL 配置的通道数}$$

$$m = \text{DMA_GAP_CH_SEL 配置的通道数}$$

$$k = \text{DMA_BANK_CNT} + 1$$

6) DMA 中断配置

通过配置 DMA_BUF_CTRL[0]使能数据缓存；=1，使能；=0，关闭。

数据缓存满一个整波可产生中断，通知 CPU 取走数据。数据缓存满时 DMA_BUFF_IF 标志置位，此时若 DMA_BUFF_IE=1，产生数据缓存满中断。

7) DMA 容错处理

在 ADC 采样间隔内，若有任何一个通道 DMA 操作没有响应，则放弃本轮所有操作，记录数据错误发生地址，发出 DMA 错误中断；然后发起新一轮操作。

EMUIF2.DMA_BUFIAErr/DMA_BUFUxErr/DMA/DMA_BUFINErr(x=A,B,C)寄存器位 (0x2BC[23:17]) 为七路 ADC 通道 DMA 数据错误标志位，当相应通道发生 DMA 数据错误时置位，若此时相应通道 DMA 错误中断使能位 EMUIE2.DMA_BUFIxErr/DMA_BUFUxErr (0x2B8[23:17]) =1,产生 DMA 错误中断，并记录数据错误发生地址放置在 DMA_WAVE_ERR_ADDR[12:0] (0x2B0) 寄存器。

8) DMA 当前指针可读

CPU 可通过读取 DMA_BUFF_ADDR 获取当前波形缓存的 DMA 指针。

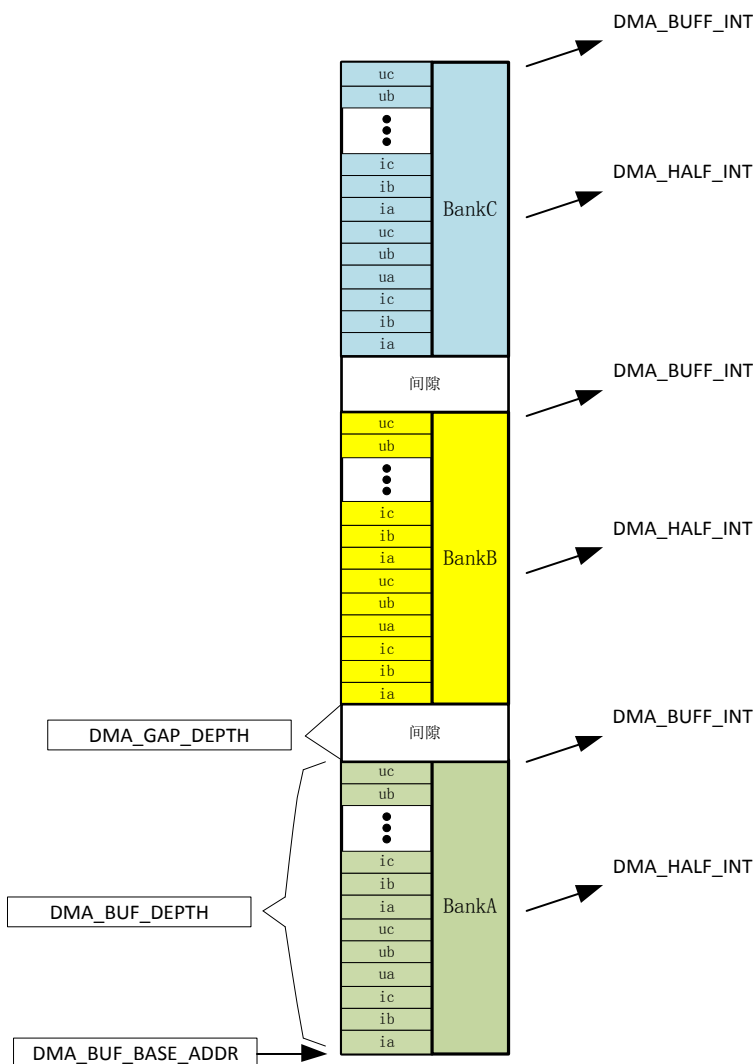
9) 自动算校验和

当配置寄存器 DMA_MODE_SEL[2]=1, 则在波形缓存过程中自动计算校验和 (8bit), 一个周波区块中每个通道数据算出一个校验和, 存放到寄存器 CHECKSUM0 和 CHECKSUM1 中。

4.3.2.3 模式 2: 不同通道连续存放 (向下兼容“模式 0”)

按 IA/IB/IC/UA/UB/UC/IN 顺序按点在 SRAM 中存放波形数据, 若某通道不使能, 则该通道忽略。

1) 模式 2 波形存放示意图



2) 模式 2 特点

该模式寄存器向下兼容“模式 0”。在模式 0 的基础上扩展以下功能:

- 缓存起始地址可任意配置, 当个周波区块的深度可任意配置 (注意必须配置成缓存通道数的整数倍);

- 增加周波区块的定义，支持多个周波区块可配置（1~16）；
- 在两个周波区块间可配置是否插入间隙区，如果配置插入间隙区，则每个周波区块后面都会插入间隙区，否则，都不插入；
- 间隙区的概念只对周波区块大于 1 个时有效；间隙区大小可任意配置；
- 波形缓存通道顺序可任意配置；
- 波形缓存过程中，可配置对整个周波区块数据做校验和计算，结果存放在间隙区和寄存器（当无间隙区时，可通过寄存器获取校验和）。

如上图所示，当配置为只有一个周波区块的时候，跟“模式 0”完全兼容。

3) 配置 ADC 波形缓存模式

通过配置 DMA_MODE_SEL [1:0]=2'b10 选择“模式 2”。

DMA 操作模式可通过 DMA_BUF_CTRL[1]寄存器位配置为单次模式或连续模式，DMA_BUF_CTRL[1]=0，单次模式，写操作数据缓存满之后则不再发起 DMA 操作；=1，连续模式，写操作数据缓存满，折返到起始地址重新开始。

4) 缓存数据通道选择和顺序

ADC 每次采样数据在缓存中按点交叉存放，不同通道数据同一时刻，按点连续存放一块。

通过配置 DMA_WAVE_CH_SEL 寄存器选择瞬时波形缓存 ADC 通道，IA/IB/IC/UA/UB/UC/IN 七路 ADC 可任意配置是否缓存。

数据在数据缓存中的通道排序可任意配置，由 DMA_WAVE_CH_SEL[27:7]寄存器配置的通道顺序决定，由软件保证配置的正确性。如 DMA_WAVE_CH_SEL[6:0]=0x5A，使能 IA/IC/UA/UC 4 个通道，IB/UB/IN 不使能；然后配置 DMA_WAVE_CH_SEL[27:7]寄存器的 CH0_SEL=0x0, CH1_SEL=0x3, CH2_SEL=0x2, CH3_SEL=0x5，选择通道数据存放顺序为 IA/UA/IC/UC/IA/UA/IC/UC。（注意此时 CH0/1/2/3_SEL 只能选择 DMA_WAVE_CH_SEL 中使能的通道）。

5) 数据缓存地址和深度

ADC 波形缓存目标基地址由 DMA_BUF_BASE_ADDR[13:0]寄存器配置，按 Word 编址，支持任意地址。

一个周波区块的深度由 DMA_BUF_DEPTH[13:0]寄存器配置，深度为 (N+1) (Word)，任意配置，每个周波区块深度相同。通过 DMA_BANK_CNT 寄存器配置周波区块的个数，n=(n+1) 个周波区块，最大支持 16 个周波区块。当配置多个周波区块，波形在几个周波区块中循环存储。比如配置为 3 表示开 3 个周波区块 A/B/C，则 DMA 首先把数据存到 A 区，A 区存满之后存到 B 区，B 区存满之后存到 C 区，C 区存满又覆盖存放 A 区。

当周波区块数量大于 1 个时，两个周波区块中间可插入间隙区，大小由 DMA_GAP_DEPTH 寄存器配置，当 DMA_GAP_DEPTH>0 时，在每两个相邻周波区块后面都插入间隙区，当 DMA_GAP_DEPTH=0，所有周波区块间都不插入间隙区。

占用 SRAM 大小 (word) = 每个周波区块深度 $a * n$ 个周波区块 + 每个通道间隙大小 $b * (n-1)$ 个通道间隙
上式中：

$$a = \text{DMA_BUF_DEPTH}$$

$$b = \text{DMA_GAP_DEPTH}$$

$$n = \text{DMA_BANK_CNT} + 1$$

6) DMA 中断配置

通过配置 DMA_BUF_CTRL[0]使能数据缓存；=1，使能；=0，关闭。

数据缓存满或半满可产生中断，通知 CPU 取走数据。数字缓存满时 DMA_BUFF_IF 标志置位，此时若 DMA_BUFF_IE=1，产生数据缓存满中断。数字缓存半满时 DMA_BUHF_IF 置位，此时若 DMA_BUHF_IE=1，产生数据缓存半满中断。

对于开多个周波区块的应用，建议只使能 DMA_BUFF_IE=1，使用全满中断；而如果兼容模式 0，建议使用半满中断。

7) DMA 容错处理

在 ADC 采样间隔内，若有任何一个通道 DMA 操作没有响应，则放弃本轮所有操作，记录数据错误发生地址，发出 DMA 错误中断；然后发起新一轮操作。

DMA_BUFIAErr/DMA_BUFUxErr/DMA/DMA_BUFINErr(x=A,B,C)寄存器位 (0x2BC[23:17]) 为七路 ADC 通道 DMA 数据错误标志位，当相应通道发生 DMA 数据错误时置位，若此时相应通道 DMA 错误中断使能位 EMUIE2.DMA_BUFIxErr/DMA_BUFUxErr (0x2B8[23:17]) =1,产生 DMA 错误中断，并记录数据错误发生地址放置在 DMA_WAVE_ERR_ADDR[12:0] (0x2B0) 寄存器。

8) DMA 当前指针可读

CPU 可通过读取 DMA_BUFF_ADDR 获取当前波形缓存的 DMA 指针。

9) 自动算校验和

当配置寄存器 DMA_MODE_SEL[2]=1，则在波形缓存过程中自动计算校验和 (8bit)，一个周波区块所有数据算出一个校验和，存放在寄存器 CHECKSUM0[7:0]。

10) 兼容“模式 0”的配置

配置 DMA_BANK_CNT=0，只有一个周波区块。其他配置跟模式 0 一样。

4.3.3 有功功率

RN2025 提供有功功率，并具有有功功率符号位，此外还具有有功功率增益校正、相位校正和 Offset 校正寄存器。

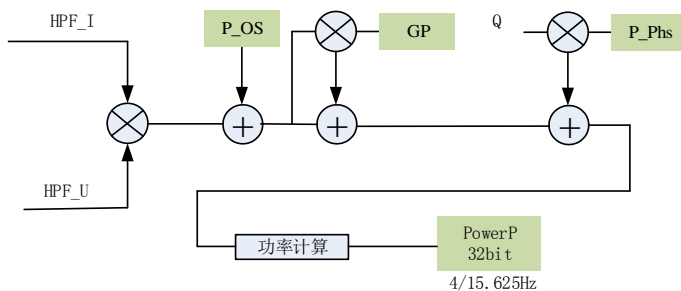


图4-2 有功功率

如图所示，在有功功率通道，可通过寄存器进行有功功率增益校正，GP 寄存器说明详见 [4.5.17 章节](#)。在有功功率通道，可通过 P_PHS 寄存器进行有功功率的相位校正。P_PHS 寄存器说明详见 [4.5.18 章节](#)。在有功功率通道，还可通过 P_OS 寄存器进行有功功率的 Offset 校正，P_OS 寄存器说明详见 [4.5.19 章节](#)。有功功率寄存器数据更新周期 RN2025B64 为 250ms，RN2025C64 为 64ms。

4.3.4 无功功率

RN2025 提供无功功率，并具有无功功率符号位，此外还具有无功功率增益校正、相位校正和 Offset 校正寄存器。

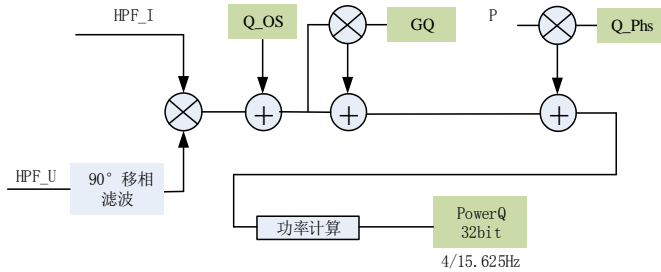


图4-3 无功功率

如图所示，在无功功率通道，可通过寄存器进行无功功率增益校正，GQ 寄存器说明详见 [4.5.17 章节](#)。在无功功率通道，可通过 Q_PHS 寄存器进行无功功率的相位校正。QPHS 寄存器说明详见 [4.5.18 章节](#)。在无功功率通道，还可通过 Q_OS 寄存器进行无功功率的 Offset 校正，Q_OS 寄存器说明详见 [4.5.19 章节](#)。无功功率寄存器数据更新周期 RN2025B64 为 250ms，RN2025C64 为 64ms。

4.3.5 视在功率

RN2025 提供视在功率以及视在功率增益校正寄存器。

视在功率采用 RMS 法，即： $S=U_{rms} \cdot I_{rms}$ 。

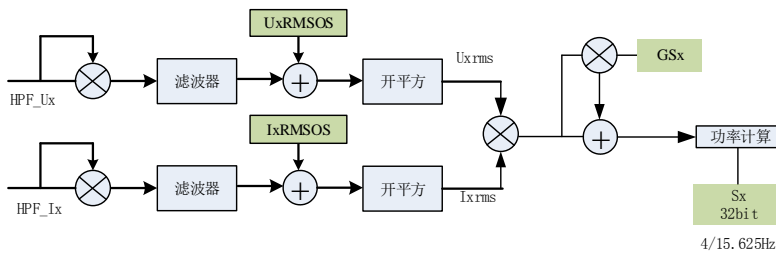


图4-4 分相视在功率

视在功率寄存器说明详见 [4.5.4 章节](#)。

如图所示，可通过通道增益校正寄存器 GSU/I 校正有效值增益进而校正视在功率增益，通道增益校正寄存器 GSU/I 说明 [4.5.14 章节](#)；也可通过视在功率增益校正寄存器来校正视在功率，功率增益校正寄存器说明详见 [4.5.18 章节](#)。

视在功率寄存器数据更新周期 RN2025B64 为 250ms，RN2025C64 为 64ms。

4.3.6 电能输出

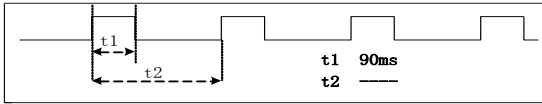
RN2025 具有三种电能输出方式：电能寄存器、CF 电能脉冲输出与快速脉冲计数 Fcnt。

1. 电能寄存器

RN2025 具有多种类型的能量寄存器，包括有功/无功/视在电能寄存器，是相应功率的积分累加输出。详细使用说明参见 [4.5.6 电能寄存器章节](#)。

2. 电能脉冲输出

校表脉冲输出，可以直接接到标准电能表进行误差比对。RN2025 具有 5 个 CF 引脚和 GPIO 复用，CF 引脚定义参见 [1.4 章节](#)。使用 CF 功能时，相应引脚配置需先配置 GPIO 寄存器为 CF 输出。



每个 CF 输出满足下面时序关系:

注意: 当脉冲输出周期小于 160ms 时, 脉冲以等占空比形式输出。

通过配置 CFCFG 寄存器 (0x180) 每个 CF 引脚都可灵活配置为有功、无功、视在中任一种脉冲输出。CFCFG 寄存器说明详见 [4.5.23 章节](#)。

EMUCFG 寄存器 (0x188) 说明详见 [4.5.24 章节](#)。

缺省情况下, 每个 CF 引脚均以 HFCONST1 为高频脉冲常数输出脉冲, 和对应的合相能量寄存器更新同步。通过配置 CFCFG 寄存器 (0x180) 每个 CF 引脚也可配置为以 HFCONST2 为高频脉冲常数输出脉冲, 该功能可用于最大需量脉冲计量。

当 CF 引脚选择使用 HFCONST2 输出脉冲时, 通过配置 EMUCFG.HFconst2_CFG 选择 HFCONST2 和 HFCONST1 是否关联。当该位=0, 两者不关联, 此时若 HFCONST1 不等于 HFCONST2, 则 CF 脉冲输出和对应的合相能量寄存器更新不同步; 当=1 时, 当 HFCONST2_CFG=1 时, HFCONST1 和 HFCONST2 串联使用, 当由 HFCONST1 配置的脉冲输出时, 由 HFCONST2 配置的脉冲计数器加 1, 当计数器值累加到 HFCONST2 的寄存器值时, 由 HFCONST2 配置的脉冲才加 1。

HFCONST 寄存器说明详见 [4.5.9 章节](#)。

3. 快速脉冲计数寄存器

快速脉冲计数寄存器包括有功/无功/视在快速脉冲计数寄存器, 是相应电能的最小累积输出。说明详见 [4.5.5 章节](#)。

4. 自定义功率/HFCONST3/电能脉冲输出/能量寄存器

RN2025 提供 12 个自定义功率寄存器及其对应的能量寄存器, 3 个自定义脉冲输出及其配套的 HFCONST3 供用户使用, 用户可根据需求利用 ADC 数据或已有的电参量计算自定义功率, 并填入自定义功率寄存器中, 配置电表常数 HFCONST3, 积分输出到对应的电能寄存器和 CF 脉冲。

应用自定义功率寄存器 D2FP00-11 和自定义能量寄存器 D2FE00-11 前, 需先配置 D2FCFG[3:0]=4'b1001, 地址段 1CCH-1F8H 寄存器为自定义功率寄存器 D2FP00-11, 地址段 1FCH-228H 寄存器对自定义能量寄存器 D2FE00-11。

自定义功率寄存器 D2FP00-11 可采用二进制补码格式, 24 位有符号数, 其中最高位是符号位。当往该寄存器写入值时, 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 可以输出脉冲, 对应脉冲分别 CF5、CF6、CF7, 可以依据 D2FCFG[13:4]配置从 CF1~CF4 管脚输出。

自定义能量寄存器 D2FE00-11 为 24bit 寄存器; 可以选择为累加型或者读后清零型, 受 EMUCFG.ERegCAR 控制。

HFCONST3 可用于配置电表常数, D2FE00-11 和 CF5-CF7 共用一个 HFCONST; 在同等条件下, 依据 HFCONST1 的计算公式得到的值乘以 4 后可作为 HFCONST3 的值。同等电表常数电能输出, 相应功率寄存器值相应放大 4 倍。

D2FCFG、D2FP00-11、D2FE00-11、HFCONST3 寄存器说明详见 [4.5.36 章节](#)。

4.3.7 启动潜动

RN2025 有功、无功或视在处于潜动状态, 则停止计量, FCnt 停止增加, 电能寄存器停止增加。

RN2025 支持四种潜动启动方式:

- 基波电流有效值启动方式:
基波电流有效值与启动阈值寄存器比较, 做全波/基波、有功/视在/无功电能潜动启动判据。
- 全波电流有效值启动方式:
全波波电流有效值与启动阈值寄存器比较, 做全波有功/视在/无功电能潜动启动判据潜动启动判据,

注意，该方式下基波电能启动仍使用基波电流有效值与阈值比较。

- 功率启动方式 1：
当有功功率超过 Start_PS 设置的阈值则启动全波/基波、有功/视在/无功电能；或当无功功率超过 Start_Q 设置的阈值则启动全波/基波、有功/视在/无功电能。
- 功率启动方式 2：
有功、无功、视在启动判断独立，启动阈值寄存器分开设置判断。有功功率和 Start_PS 设置的阈值比较做全波/基波、有功电能判据。无功功率和 Start_Q 设置的阈值比较做全波/基波无功电能判据。有功和无功只要有一个超过阈值则启动该相视在电能。**建议客户使用该种方式。**

四种启动方式寄存器配置如下：

启动方式	EMUCFG.Pstart_CFG[1:0]	EMUCFG.NoLoadCFG
全波电流有效值启动方式	0x	0
基波电流有效值启动方式	0x	1
功率启动方式 1	10	x
功率启动方式 2	11	x

EMUCFG.Pstart_CFG 和 EMUCFG.NoLoadCFG 寄存器说明见 [4.5.24 章节](#)。

RN2025 提供两个启动阈值寄存器 Start_PS(0x08)和 Start_Q(0x0C)，上述四种启动方式下，Start_PS 和 Start_Q 作用域如下表：

启动方式	Start_PS	Start_Q
电流有效值启动方式 1 (全波电流或基波电流启动)	全波电流超过该阈值，则启动全波有功/视在电能计量； 基波电流超过该阈值，则启动基波有功/视在电能计量。	全波电流超过该阈值，则启动全波无功电能计量； 基波电流超过该阈值，则启动基波无功电能计量。
电流有效值启动方式 2 (基波电流启动)	基波电流超过该阈值，则启动全波/基波、有功/视在电能计量。	基波电流超过该阈值，则启动全波/基波无功电能计量。
功率启动方式 1 (有功或无功功率任意启动)	全波有功功率超过 Start_PS 阈值，或者全波无功功率超过 Start_Q 阈值，只要满足其中一个条件，则启动全波有功/无功/视在电能计量； 基波有功功率超过 Start_PS 阈值，或者基波无功功率超过 Start_Q 阈值，只要满足其中一个条件，则启动基波有功/无功/视在电能计量；	
功率启动方式 2 (有功功率启动、无功功率启动分开)	全波有功功率超过 Start_PS 阈值，则启动全波有功/视在电能计量； 基波有功功率超过 Start_PS 阈值，则启动基波有功/视在电能计量；	全波无功功率超过 Start_Q 阈值，则启动全波无功电能计量； 基波无功功率超过 Start_Q 阈值，则启动基波无功电能计量；

Start_PS 和 Start_Q 计算公式和使用说明详见 [4.5.10 章节](#)。

RN2025 提供启动判断结果寄存器 NoLoad (0x19C) 表示全波有功视在/全波无功/基波有功视在/基波无功/基波视在启动状态。说明详见 [4.5.28 章节](#) 启动状态寄存器 NoLoad。

4.3.8 电压电流有效值

有效值寄存器说明详见 [4.5.3 章节](#)。

1. 全波电压电流有效值

全波电压有效值包括电压有效值 U_{rms} ，全波电流有效值包括 I_{Arms} 、 I_{Brms} 。

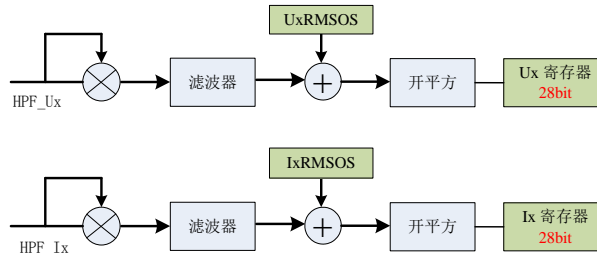


图4-5 电压电流有效值

电压电流有效值的校正：① 通过通道增益寄存器校正增益。详见 [4.5.14 章节](#) 通道增益寄存器 GSU/I 。② 通过有效值 $OFFSET$ 校正寄存器校正有效值 $OFFSET$ ，提高小信号有效值测量精度。详见 [4.5.16 章节](#) 有效值 $OFFSET$ 校正寄存器。

U_{rms} 除做为参数输出，还做为电压过零检测、线电压测频、相角阈值检测的判据。 I_A 、 I_B 电流有效值除做为参数输出，还做为各相全波电流启动判据以及电流过零检测和电流相角阈值的判据。

2. 基波电压电流有效值

基波电压有效值包括基波电压有效值 FU_{rms} ；基波电流有效值包括 FI_{Arms} 、 FIB_{rms} 。

基波电压电流有效值的校正：① 通过通道增益寄存器校正增益。详见 [4.5.14 章节](#) 通道增益寄存器 GSU/I 。② 通过有效值 $OFFSET$ 校正寄存器校正有效值 $OFFSET$ ，提高小信号有效值测量精度。详见 [4.5.16 章节](#) 有效值 $OFFSET$ 校正寄存器。

FI_{Arms} 、 FIB_{rms} 电流有效值除做为参数输出，还做为基波电流启动判据判据。

4.3.9 半波有效值

RN2025 提供半波有效值寄存器 U_{HW} 、 $I_{A_{HW}}$ 、 $I_{B_{HW}}$ ，寄存器说明详见 [4.5.8 章节](#)。

- 半波有效值定义：以过零点为起点，半周波内所有点平方后累加和的均方根值。
- RN2025 半波有效值寄存器提供半周波内采样点的平方和；用户软件做均方根运算，**或者通过寄存器配置选择经过均方根后的结果输出。**

以电压为例，若寄存器配置选择半波有效值为原始数据平方后结果，则：

$$U_{A_{HW}} = K_{UA_{hw}} * \text{sqrt}(\text{reg}(U_{A_{HW}}))$$

其中：

$U_{A_{HW}}$ 为电压半波有效值，此寄存器受通道增益寄存器 GSU_A 影响；

$\text{reg}(U_{A_{HW}})$ 为经过 GSU_A 寄存器增益校正后 U_A 半波有效值寄存器值，如果 A 相电压有效值经过了增益校正，则 $U_{A_{HW}}$ 不需要做额外的增益校正；

sqrt 为平方根运算，由软件完成；

$K_{UA_{hw}}$ 为转换系数，将测试值转换为示值，由软件完成。

- 性能：
 - 电压：参照电能质量标准 S 级：10%~150% U_n ，误差<0.5%；
 - 电流：5% I_n ~ I_{max} 误差<0.5%；1% I_n ~5% I_n 误差<2%

- 寄存器更新周期：100hz(50HZ 应用)/120hz(60HZ 应用)。
- 适用频率：50HZ/60HZ。电网频率通过 EMUCFG.FreqSEL(50hz/60hz 选择)寄存器位来选择。
- 校正方法：增益校正复用有效值 GSU/I 寄存器增益校正过程，转换系数由 MCU 完成。
- 相关寄存器：通道增益寄存器 GSU/I、EMUCFG.FreqSEL(50hz/60hz 选择)、EMUCFG2.RMS_HW_CFG。

4.3.10 过零检测

RN2025 具有三路电压和电流波形正向过零检测功能，当发生正向过零时，相应的过零中断标志位 EMUIF.ZXIxF/ ZXUxF(0x194[7:1])置位，若相应的过零中断 EMUEMUIE.ZXIxE/ ZXUxE(0x190[7:1])使能时，产生过零中断。过零中断使能以及标志寄存器说明详见 [4.5.26 章节](#)。需要注意的是电流过零依赖电压过零，若不存在电压过零时，亦不能产生电流过零，但不要求有对应相电压过零。

RN2025 提供电压电流过零阈值寄存器 ZXOT (0x14)、电压过零及测频阈值寄存器 ZXOTU (0x1B4)，若该路电压有效值小于所配置的过零阈值，则过零中断、测频不输出。电流过零阈值只使用 ZXOT (0x14)。电压过零、测频阈值可选择使用 ZXOT 或 ZXOTU，当 ZXOTCFG.ZXOTUS=1 (0x 1BC) 时可选择 ZXOTU (0x1B4) 作为电压过零、测频阈值；当 ZXOTCFG.ZXOTUS=0 时选择 ZXOT 作为电压过零、测频阈值。默认使用 ZXOT (0x14)。

ZXOT 寄存器说明详见 [4.5.11 章节](#)，ZXOTCFG 寄存器说明详见 [4.5.35 章节](#)，ZXOTU 寄存器说明详见 [4.5.33 章节](#)。

注：RN2025C64 应用时，必须将 ZXOT 和 ZXOTU 寄存器配置为 0。

4.3.11 电压线频率

RN2025 具有电压线频率测量功能。电压线频率寄存器 UFeq (0x450) 说明详见寄存器 [4.5.7 章节](#)。

电压线频率分辨率达 0.001Hz，测量准确度达 0.02%。测频范围：40-70Hz。

电压线频率寄存器更新周期为 32 个周波。

测频基准：若 $U_{rms} > ZXOT$ (归一化值，也可配置 ZXOTCFG.ZXOTUS 寄存器位选择使用 ZXOTU 寄存器作为阈值，下同)，则启动测频；电压有效值 $< ZXOT$ ，频率寄存器输出为 0。

4.3.12 电压暂降检测

RN2025 具有电压暂降检测功能。

电压暂降判据为：电压波形采样值的绝对值的高 16 位小于 SAGCFG.SAGLVL[15:0](0x160[15:0]) 寄存器值，且持续寄存器 SAGCFG.SAGCYC[7:0]个半周波数，判定电压暂降。SAGCFG 寄存器说明详见 [4.5.20 章节](#)。电压暂降将导致 PHASES 寄存器(0x1A4)中 SAGUx 寄存器位置 1。

PHASES.SAGU 置 1 会导致 EMUIF.SAGIF 置 1,同时产生 SAG 中断。

PHASES.SAGU 寄存器说明详见 [4.5.29 章节](#)，SAGCFG 寄存器说明详见 [4.5.20 章节](#)，SAGIE 寄存器位以及 SAGIF 寄存器位说明详见 [4.5.25.1 章节](#)

注：必须配置 SAGCYC>1，且阈值配置 SAGLVL>0xE7。

4.3.13 过压检测和电压半波峰值

RN2025 具有过压判断功能。

电压过压 (OverVoltage) 判据为：电压波形采样值半波峰值绝对值的高 16 位 $U_{xPEAK_HW}[15:0]$ (0x58C-594) 大于过压幅值阈值寄存器 OV2_LVL[15:0] (0x 278)，且过压持续时间大于等于过压时间阈值寄存器 OV2_CNTTH[15:0] (0x27C)个半周波数，判定该相电压过压。

电压过压将导致 EMUIF2.OVx_IF (0x2BC.9-11)标志置位，若 EMUIE2.OVx_IE (0x2B8.9-11)=1，则产生过压中断。

为保证半周波时间计数准确，用户需先配置 EMUCFG.FreqSEL (0x188.11)，进行 50Hz/60Hz 应用配置。

开启电压过压检测功能的条件：OV2_LVL[15:0]≠0 且 OV2_CNTTH[15:0] ≠0

关闭电压过压检测功能的条件：OV2_LVL[15:0]=0 或 OV2_CNTTH[15:0]=0

OV2_LVL、OV2_CNTTH、UxPEAK_HW 寄存器说明详见 [4.5.38 章节](#)。EMUIE2.OVx_IE、EMUIF2.OV2x_IF 寄存器说明详见 [4.5.40 章节](#)。

相关寄存器：通道增益寄存器 GSUx、EMUCFG.FreqSEL(50hz/60hz 选择)。

注：RN2025C64 不支持该功能。

4.3.14 过流检测

RN2025 具有电流的过流判断功能。

电流过流 (OverCurrent) 判据为：当该相电流采样值的绝对值高 16 位大于过流幅值阈值寄存器 OILVL[15:0] (0x168)，判定电流过流。

OILVL[15:0]≠0x0，开启电流过流判断功能。OILVL[15:0]=0x0，则关闭电流过流判断功能。默认关闭。

电流过流将导致 PHASES.OIx 寄存器位 (0x1A4.12-14) 置 1 同时 EMUIF.OIIF 寄存器位 (0x0x194.15) 置 1。若 EMUIE.OIIE (0x190.15) =1，OIIF 置 1 会导致中断。EMUIF.OIIF 写 1 清零并清中断，同时清 PHASES.OIx 标志。

OILVL 寄存器说明详见 [4.5.21 章节](#)。PHASES 寄存器说明详见 [4.5.29 章节](#)。OIIE 寄存器位以及 OIIF 寄存器位说明详见 [4.5.26 章节](#)。

相关寄存器：通道增益寄存器 GSIx。

4.4 寄存器列表

注意：开启 ADC 和基准电压、对 ADC 的增益进行设置，以及开启计量时钟等配置参见系统控制 [2.6.4](#) 寄存器描述章节。

注意：基地址 0x50004000 的计量模块寄存器没有校验和，软件需要定时查询所有开放的寄存器以及保留寄存器符合预期值，如果发现寄存器错误，建议对计量重新初始化（包括对特别列出的保留寄存器位）。

下文寄存器以 0x 开头或者以 H 结尾的数值，均代表十六进制。

下文中红色字体表示 RN2025 (B64) 相对于 RN2025 (A64) 有修改或者新增的内容。

4.4.1 计量参数寄存器列表

表 4-1 计量参数寄存器列表

基地址	0x50004000				
偏移地址	名称	R/W	有效字长	复位值	功能描述
全波计量参数寄存器					
300H	UAWAV	R	3	0x00	电压采样数据
30CH	IAWAV	R	3	0x00	A 路电流采样数据
310H	IBWAV	R	3	0x00	B 路电流采样数据
31CH	Urms	R	4	0x00	电压有效值
320H	AIn4rms (修改)	R	4	0x00	A 版芯片为 AIN4 有效值寄存器； B 新芯片无效；
32CH	IArms	R	4	0x00	A 路电流有效值
330H	IBrms	R	4	0x00	B 路电流有效值
344H	PowerPA	R	4	0x00	A 路有功功率
348H	PowerPB (新增)	R	4	0x00	B 路有功功率

354H	PowerQA	R	4	0x00	A 路无功功率，同 A 路基波无功功率
358H	PowerQB (新增)	R	4	0x00	B 路无功功率，同 B 路基波无功功率
364H	PowerSA	R	4	0x00	A 路视在功率
368H	PowerSB (新增)	R	4	0x00	B 路视在功率
384H	PAFCnt	R/W	3	0x00	A 路有功快速脉冲计数
388H	PBFCnt (新增)	R/W	3	0x00	B 路有功快速脉冲计数
394H	QAFCnt	R/W	3	0x00	A 路无功快速脉冲计数
398H	QBFCnt (新增)	R/W	3	0x00	B 路无功快速脉冲计数
3A4H	SAFCnt	R/W	3	0x00	A 路视在快速脉冲计数
3A8H	SBFCnt (新增)	R/W	3	0x00	B 路视在快速脉冲计数
3B4H	EPA	R	3	0x00	A 路有功能量寄存器
3B8H	EPB (新增)	R	3	0x00	B 路有功能量寄存器
3E4H	EQA	R	3	0x00	A 路无功能量寄存器，脉冲来自 A 路无功基波
3E8H	EQB (新增)	R	3	0x00	B 路无功能量寄存器，脉冲来自 B 路无功基波
414H	ESA	R	3	0x00	A 路视在能量寄存器
418H	ESB (新增)	R	3	0x00	B 路视在能量寄存器
基波谐波计量参数寄存器					
450H	UFreq	R	3	0x140000	电压线频率
454H	FUA	R	4	0x00	基波电压有效值
460H	FIA	R	4	0x00	A 路基波电流有效值
464H	FIB	R	4	0x00	B 路基波电流有效值
46CH	FPA	R	4	0x00	A 路基波有功功率
470H	FPB (新增)	R	4	0x00	B 路基波有功功率
47CH	FQA	R	4	0x00	A 路基波无功功率
480H	FQB (新增)	R	4	0x00	B 路基波无功功率
48CH	FSA	R	4	0x00	A 路基波视在功率
490H	FSB (新增)	R	4	0x00	B 路基波视在功率
4ACH	FPAFCnt	R/W	3	0x00	A 路基波有功快速脉冲计数
4B0H	FPBFCnt (新增)	R/W	3	0x00	B 路基波有功快速脉冲计数
4BCH	FQAFCnt	R/W	3	0x00	A 路基波无功快速脉冲计数
4C0H	FQBFCnt (新增)	R/W	3	0x00	B 路基波无功快速脉冲计数
4CCH	FSAFCnt (新增)	R/W	3	0x00	A 路基波视在快速脉冲计数
4D0H	FSBFCnt (新增)	R/W	3	0x00	B 路基波视在快速脉冲计数
4DCH	FEPA	R	3	0x00	A 路基波有功电能
4E0H	FEPB (新增)	R	3	0x00	B 路基波有功电能
50CH	FEQA	R	3	0x00	A 路基波无功电能
510H	FEQB (新增)	R	3	0x00	B 路基波无功电能
53CH	FESA (新增)	R	3	0x00	A 路基波视在能量寄存器

540H	FESB (新增)	R	3	0x00	B 路基波视在能量寄存器
574H	UA_HW (修改)	R	3	0x00	电压半波有效值
580H	IA_HW (修改)	R	3	0x00	A 路电流半波有效值
584H	IB_HW (修改)	R	3	0x00	B 路电流半波有效值
58CH	UAPEAK_HW	R	2	0x00	电压半波峰值寄存器
59CH	FUA_WAVE (新增)	R	3	0x00	电压基波瞬时采样波形
5A8H	FIA_WAVE (新增)	R	3	0x00	A 路电流基波瞬时采样波形
5ACH	FIB_WAVE (新增)	R	3	0x00	B 路电流基波瞬时采样波形
5B8H	SPL_PA (新增)	R	4	0x00	A 路瞬时有功功率, 更新速度为 8Khz; 如果 P_HW_SEL=1, 那么更新周期为半周波。
5BCH	SPL_PB (新增)	R	4	0x00	B 路瞬时有功功率, 更新速度为 8Khz
5C4H	SPL_QA (新增)	R	4	0x00	A 路瞬时基波无功功率, 更新速度为 8Khz
5C8H	SPL_QB (新增)	R	4	0x00	B 路瞬时基波无功功率, 更新速度为 8Khz
5D0H	SPL_FPA (新增)	R	4	0x00	A 路瞬时基波有功功率, 更新速度为 8Khz
5D4H	SPL_FPB (新增)	R	4	0x00	B 路瞬时基波有功功率, 更新速度为 8Khz
5DCH	SPL_FQA (新增)	R	4	0x00	A 路瞬时基波无功功率, 更新速度为 8Khz
5E0H	SPL_FQB (新增)	R	4	0x00	B 路瞬时基波无功功率, 更新速度为 8Khz

4.4.2 计量配置和状态寄存器列表

表 4-2 计量配置和状态寄存器列表

基地址	0x50004000				
偏移地址	名称	R/W	有效字长	复位值	功能描述
EMU 校表寄存器					
00H	HFCnst1	R/W	2	0x1000	高频脉冲计数寄存器 1
04H	HFCnst2	R/W	2	0x1000	高频脉冲计数寄存器 2
08H	Start_PS	R/W	2	0x0250	有功视在启动阈值寄存器
0CH	Start_Q	R/W	2	0x0250	无功启动阈值寄存器
10H	LostVoltage	R	2	0x0400	失压阈值寄存器
14H	ZXOT	R/W	2	0x0073	过零阈值寄存器 注: RN2025C64 应用时, 必须将该寄存器配置为 0。
18H	PRTH1L	R/W	2	0x0000	相位分段校正电流阈值 1 下限
1CH	PRTH1H	R/W	2	0x0000	相位分段校正电流阈值 1 上限

20H	PRTH2L	R/W	2	0x0000	相位分段校正电流阈值 2 下限
24H	PRTH2H	R/W	2	0x0000	相位分段校正电流阈值 2 上限
30H	PHSUA (修改)	R/W	4	0x80	采样通道 U 相位校正寄存器
34H	保留位	R	4	0x80	只读位, 读出为 0x80, 参与校验和计算。无实际意义。
38H	保留位	R	4	0x80	只读位, 读出为 0x80, 参与校验和计算。无实际意义。
3CH	PHSIA (修改)	R/W	4	0x808080	采样通道 IA 分段相位校正寄存器
40H	PHSIB (修改)	R/W	4	0x808080	采样通道 IB 分段相位校正寄存器
44H	保留位	R	4	0x808080	只读位, 读出为 0x808080, 参与校验和计算。无实际意义。
48H	保留位	R	1	0x80	只读位, 读出为 0x80, 参与校验和计算。无实际意义。
4CH	GSUA	R/W	2	0x0000	采样通道 U 通道增益
58H	GSIA	R/W	2	0x0000	采样通道 IA 通道增益
5CH	GSIB	R/W	2	0x0000	采样通道 IB 通道增益
68H	DCOS_UA	R/W	2	0x0000	采样通道 U 直流 Offset 校正
74H	DCOS_IA	R/W	2	0x0000	采样通道 IA 直流 Offset 校正
78H	DCOS_IB	R/W	2	0x0000	采样通道 IB 直流 Offset 校正
84H	UA_OS	R/W	2	0x0000	电压有效值 Offset
90H	IA_OS	R/W	2	0x0000	A 路电流有效值 Offset
94H	IB_OS	R/W	2	0x0000	B 路电流有效值 Offset
A0H	GPA	R/W	2	0x0000	A 路有功功率增益
A4H	GPB (新增)	R/W	2	0x0000	B 路有功功率增益
ACH	GQA	R/W	2	0x0000	A 路无功功率增益
B0H	GQB (新增)	R/W	2	0x0000	B 路无功功率增益
B8H	GSA	R/W	2	0x0000	A 路视在功率增益
BCH	GSB (新增)	R/W	2	0x0000	B 路视在功率增益
C4H	PA_PHSL	R/W	2	0x0000	A 路有功分段相位校正寄存器-低段
C8H	PB_PHSL (新增)	R/W	2	0x0000	B 路有功分段相位校正寄存器-低段
D0H	QA_PHSL	R/W	2	0x0000	A 路无功分段相位校正寄存器-低段
D4H	QB_PHSL (新增)	R/W	2	0x0000	B 路无功分段相位校正寄存器-低段
DCH	PA_OS	R/W	2	0x0000	A 路有功功率 Offset
E0H	PB_OS (新增)	R/W	2	0x0000	B 路有功功率 Offset
E8H	QA_OS	R/W	2	0x0000	A 路无功功率 Offset
ECH	QB_OS (新增)	R/W	2	0x0000	B 路无功功率 Offset

F4H	FUA_OS (新增)	R/W	2	0x0000	基波电压有效值 Offset
100H	FIA_OS	R/W	2	0x0000	A 路基波电流有效值 Offset
104H	FIB_OS (新增)	R/W	2	0x0000	B 路基波电流有效值 Offset
10CH	GFPA	R/W	2	0x0000	A 路基波有功功率增益
110H	GFPB (新增)	R/W	2	0x0000	B 路基波有功功率增益
118H	GFQA	R/W	2	0x0000	A 路基波无功功率增益
11CH	GFQB (新增)	R/W	2	0x0000	B 路基波无功功率增益
124H	GFSA (新增)	R/W	2	0x0000	A 路基波视在功率增益
128H	GFSB (新增)	R/W	2	0x0000	B 路基波视在功率增益
130H	FPA_PHS	R/W	2	0x0000	A 路基波有功相位校正寄存器
134H	FPB_PHS (新增)	R/W	2	0x0000	B 路基波有功相位校正寄存器
13CH	FQA_PHS	R/W	2	0x0000	A 路基波无功相位校正寄存器
140H	FQB_PHS (新增)	R/W	2	0x0000	B 路基波无功相位校正寄存器
148H	FPA_OS	R/W	2	0x0000	A 路基波有功功率 Offset
14CH	FPB_OS (新增)	R/W	2	0x0000	B 路基波有功功率 Offset
154H	FQA_OS	R/W	2	0x0000	A 路基波无功功率 Offset
158H	FQB_OS (新增)	R/W	2	0x0000	B 路基波无功功率 Offset
160H	SAGCFG	R/W	3	0x000000	电压暂降阈值配置
164H	OVLVL (新增)	R/W	2	0x0000	过压阈值配置
168H	OILVL	R/W	2	0x0000	过流阈值配置
EMU 配置和状态寄存器					
180H	CFCFG	R/W	3	0x043210	CF 引脚配置寄存器
188H	EMUCFG (修改)	R/W	3	0x400000	EMU 配置寄存器
18CH	EMUCON	R/W	3	0x000000	EMU 控制寄存器
190H	EMUIE (修改)	R/W	2	0x0000	EMU 中断允许寄存器, 写保护
194H	EMUIF (修改)	R	2	--	EMU 中断标志和状态寄存器
198H	PQSign	R	2	--	有功无功功率方向寄存器
19CH	Noload	R	2	--	潜动启动状态寄存器
1A0H	IRegionS	R	1	--	电流分区状态寄存器
1A4H	PHASES	R	2	--	相电压电流状态寄存器
1A8H	EMU_CKM(新增)	R	4	--	计量单元配置寄存器校验和
1ACH	RData	R	4		上一次读出的数据
1B0H	WData	R	3		上一次写入的数据
1B4H	ZXOTU	R/W	2	0x0	电压相角及测频阈值寄存器 RN2025C64 应用时, 必须将该寄存器配置为 0。
1B8H	AUOTDC_EN	R/W	2	0x80	直流 OFFSET 自动校正使能寄存器
1BCH	ZXOTCFG	R/W	2	0x0	过零计算配置及标志寄存器

1C0H	D2FTIME (新增)	R/W	3	0x0	自定义功率积分时间设置寄存器
1C4H	D2FCFG	R/W	2	0x0	自定义功率和功率通道相位分段校正配置寄存器
1C8H	HFCONST3 (修改)	R/W	2	0x0	自定义高频脉冲常数
1CCH	PA_PHSM/D2FP00	R/W	3	0x0	A 路有功分段相位校正寄存器-中段, 自定义功率寄存器 0
1D0H	PA_PHSB/D2FP01	R/W	3	0x0	A 路有功分段相位校正寄存器-高段, 自定义功率寄存器 1
1D4H	PB_PHSM/D2FP02	R/W	3	0x0	B 路有功分段相位校正寄存器-中段, 自定义功率寄存器 2
1D8H	PB_PHSB/D2FP03	R/W	3	0x0	B 路有功分段相位校正寄存器-高段, 自定义功率寄存器 3
1DCH	D2FP04	R/W	3	0x0	自定义功率寄存器 4
1E0H	D2FP05	R/W	3	0x0	自定义功率寄存器 5
1E4H	QA_PHSM/D2FP06	R/W	3	0x0	A 路无功分段相位校正寄存器-中段, 自定义功率寄存器 6
1E8H	QA_PHSB/D2FP07	R/W	3	0x0	A 路无功分段相位校正寄存器-高段, 自定义功率寄存器 7
1ECH	QB_PHSM/D2FP08	R/W	3	0x0	B 路无功分段相位校正寄存器-中段, 自定义功率寄存器 8
1F0H	QB_PHSB/D2FP09	R/W	3	0x0	B 路无功分段相位校正寄存器-高段, 自定义功率寄存器 9
1F4H	D2FP10	R	3	0x0	自定义功率寄存器 10
1F8H	D2FP11	R	3	0x0	自定义功率寄存器 11
1FCH	D2FE00	R	3	0x0	自定义电能寄存器 0
200H	D2FE01	R	3	0x0	自定义电能寄存器 1
204H	D2FE02	R	3	0x0	自定义电能寄存器 2
208H	D2FE03	R	3	0x0	自定义电能寄存器 3
20CH	D2FE04	R	3	0x0	自定义电能寄存器 4
210H	D2FE05	R	3	0x0	自定义电能寄存器 5
214H	D2FE06	R	3	0x0	自定义电能寄存器 6
218H	D2FE07	R	3	0x0	自定义电能寄存器 7
21CH	D2FE08	R	3	0x0	自定义电能寄存器 8
220H	D2FE09	R	3	0x0	自定义电能寄存器 9
224H	D2FE10	R	3	0x0	自定义电能寄存器 10
228H	D2FE11	R	3	0x0	自定义电能寄存器 11
260H	OI2_LVL	R/W	2	0x0	电流通道过流检测阈值寄存器
264H	OI2_CNTTH	R/W	1	0x0	电流通道过流判断时间寄存器

270H	EMU_CFG2 (修改)	R/W	1	0x4	半波有效值过零选择寄存器
274H	OIRZ_CNTTH	R/W	2	0x0	保留, 不影响正常计量
278H	OV2_LVL	R/W	2	0x0	电压过压幅值阈值配置寄存器
27CH	OV2_CNTTH	R/W	2	0x0	电压过压时间阈值配置寄存器
2A0H	DMA_BUF_CTRL (修改)	R/W	2	0x0	ADC 波形缓存 DMA 控制寄存器
2A4H	DMA_BUF_BASE_ADDR (修改)	R/W	2	0xC00	ADC 波形缓存基址寄存器
2A8H	DMA_BUF_DEPTH (修改)	R/W	2	0x7F	ADC 波形缓存深度寄存器
2ACH	DMA_WAVE_CHANNEL_SEL (修改)	R/W	1	0x0D63447F	ADC 波形缓存通道选择寄存器
2B0H	DMA_WAVE_ERROR_ADDR (修改)	R	2	0x0	ADC 波形缓存错误地址寄存器
2B4H	DMA_WAVECON (修改)	R/W	2	0x0	ADC 波形缓存控制寄存器
2B8H	EMUIE2	R/W	2	0x0	EMU 中断使能寄存器 2
2BCH	EMUIF2	R/W	2	-	EMU 事件标志寄存器 2
2C0H	DMA_WAVE_MODE_SEL (新增)	R/W	1	0x0	波形缓存 DMA 模式选择
2C4H	DMA_GAP_CHANNEL_SELECT (新增)	R/W	1	0x0	波形缓存通道间隙选择
2C8H	DMA_WAVE_GAP_DEPTH (新增)	R/W	2	0x0	波形缓存通道间隙深度
2CCH	DMA_BANK_COUNT (新增)	R/W	1	0x0	波形缓存周波区块个数
2D0H	DMA_BUF_ADDR (新增)	R	4	0xC00	波形缓存当前 DMA 指针地址
2D4H	DMA_CHKSUM0 (新增)	R	4	0x0	波形缓存校验和 0
2D8H	DMA_CHKSUM1 (新增)	R	4	0x0	波形缓存校验和 1
2FCH	SPCMD	R/W	1	0x0	EMU 写使能密码寄存器
波形缓存校正和配置寄存器 (新增)					
600H	DMA_WAVECON2	R/W	2	0x00	ADC 波形缓存控制寄存器
604H	PHS_UA_WAVE	R/W	2	0x0	ADC 波形缓存 U 通道相位校正寄存器
610H	PHS_IA_WAVE	R/W	2	0x00	ADC 波形缓存 IA 通道相位校正寄存器

614H	PHS_IB_WAVE	R/W	2	0x0	ADC 波形缓存 IB 通道相位校正寄存器
620H	GSUA_WAVE	R/W	2	0x00	ADC 波形缓存 U 通道增益校正寄存器
62CH	GSIA_WAVE	R/W	2	0x00	ADC 波形缓存 IA 通道增益校正寄存器
630H	GSIB_WAVE	R/W	2	0x00	ADC 波形缓存 IB 通道增益校正寄存器
功率相位分段校正寄存器（新增）					
640H	PA_PHSM	R/W	2	0x0	A 路有功分段相位校正寄存器-中段
644H	PA_PHSB	R/W	2	0x0	A 路有功分段相位校正寄存器-高段
648H	PB_PHSM	R/W	2	0x0	B 路有功分段相位校正寄存器-中段
64CH	PB_PHSB	R/W	2	0x0	B 路有功分段相位校正寄存器-高段
658H	QA_PHSM	R/W	2	0x0	A 路无功分段相位校正寄存器-中段
65CH	QA_PHSB	R/W	2	0x0	A 路无功分段相位校正寄存器-高段
660H	QB_PHSM	R/W	2	0x0	B 路无功分段相位校正寄存器-中段
664H	QB_PHSB	R/W	2	0x0	B 路无功分段相位校正寄存器-高段
680H	UADC_SEL	R/W	3	0x000	<p>三字节有效，当高两字节[23:8]写入 0x2026H 后，低字节[5:0]才可写，并且高两字节要一直保持为 0x2026H，低字节才会真正起到切换 ADC 通道的作用。</p> <p>UADC_SEL[1:0] =00/01:火线通道的 U 使用 UA-ADC 计算 =10/11:保留（不建议配置）</p> <p>UADC_SEL[3:2] =00:零线通道的 U 不使用 UA-ADC 计算 =01:零线通道的 U 使用 UA-ADC 计算 =10/11 保留</p>

					UADC_SEL[5:4] =00/01/10/11:保留(默认值 00, 不建议修改) 注意: 输出零线功率、零线电能, 要把 UADC_SEL[3:2]配置为 01; 注: RN2025C64 配置无效。
--	--	--	--	--	--

4.5 寄存器说明

4.5.1 EMU 写使能寄存器 SPCMD (0x2FC)

偏移地址: 2FCH; 默认值: 0x0。

该寄存器用于基址为 0x50004000 三相计量引擎 EMU 可读可写寄存器写保护命令。上电复位和软件复位后该寄存器回到默认写保护状态。

命令	命令字	功能描述
EMU_WREN	0xE5	写使能命令。写使能有效后, 所有可读可写寄存器写使能, 该寄存器读为 0x01。
EMU_WRDIS	0xDC	写保护命令。写保护有效后, 所有可读可写寄存器写保护, 该寄存器读为 0x00。
	其他	无效。芯片不动作

4.5.2 波形采样寄存器

偏移地址	300H	30CH	310H
寄存器	UWAV	IAWAV	IBWAV
偏移地址	59CH	5A8H	5ACH
寄存器	FUAWAV	FIWAV	FIBWAV

RN2025 的三路 ADC 采样数据寄存器, 其中 300H~310H 为全波波形采样值, 59CH~5ACH(新增)为基波波形采样值。

波形采样寄存器为 24 位有符号数, 采用补码格式, 最高位为符号位。采样率固定 8KHz, 数据源为 ADC 经过相位校正/增益校正/高通滤波后的数据。

假设外部输入峰值为 $A=0.1v$ 的交流信号, 那么波形采样寄存器的值为:

$$A * 0.5 * 2^{23} = 419430 \text{ (十进制)} = 24' \text{ H66666 (十六进制)}$$

波形采样寄存器的峰值的归一化值 = 对应全波有效值寄存器的归一化值 * $\sqrt{2} * 0.5$ 。有效值寄存器的归一化值为 $RMS/2^{27}$, 波形采样寄存器的归一化值为 $WAV/2^{23}$ 。即 $WAV/2^{23} = (RMS/2^{27}) * \sqrt{2} * 0.5$ 。

波形采样寄存器更新时会置位标志寄存器 EMUIF.WAVUPIF 标志位 (0x194.0), 该位写 1 清零。若使能波形寄存器更新中断 EMUIE.WAVUPIE(0x190.0), 则每次更新发生, 会产生 EMU 中断。

4.5.3 有效值寄存器

偏移地址	31CH	32CH	330H
寄存器	UA	IA	IB
偏移地址	454H	460H	464H

寄存器	FUA	FIA	FIB
-----	-----	-----	-----

有效值寄存器分为：

全波电压电流有效值（UA/IA/IB）

基波电压电流有效值（FUA/FIA/FIB）

1 有效值是四字节寄存器，为 27 位（bit0-bit26）无符号数，bit27-bit31 始终为零。这四类有效值参数更新的周期 **RN2025B64 为 250ms，RN2025C64 为 64ms。**

假设输入的正弦波电压峰值为 $A=1V$ ，那么有效值寄存器值约为：

$A * 0.70710678 * 2^{27} = 32'd94906255$ （十进制）= $32'h5A82799$ （十六进制）

上述各类有效值计算公式详见 [4.3.8 章节](#)。

ADC PGA 配置、高通使能配置、电流通道增益寄存器 GSI/U 配置对上述电流有效值有效。有效值 OFFSET 寄存器对 1,2 类有效值有效。不论高通是否关闭，通道直流 OFFSET 校正寄存器 DCOSx 对第 1 类有效值有效。

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

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

4.5.4 功率寄存器

平均功率寄存器：（部分新增）

偏移地址	344H	348H	354H	358H
寄存器	PA	PB	QA	QB
偏移地址	364H	368H		
寄存器	SA	SB		
偏移地址	46CH	470H	47CH	480H
寄存器	FPA	FPB	FQA	FQB
偏移地址	48CH	490H		
寄存器	FSA	FSB		

平均功率寄存器包括全波有功功率 PA/PB、全波视在功率 SA/SB；基波有功功率 FPA/FPB、基波无功功率 FQA/FQB、基波视在功率 FSA/FSB。

特别注意：无功功率 QA/QB/读出值同 FQA/FQB，即：RN2025 仅支持基波无功，不支持全波无功。

各类有功功率计算公式和相关校正寄存器详见 [4.3.3 章节](#)。各类无功功率计算公式和相关校正寄存器详见 [4.3.4 章节](#)。各类视在功率计算公式和相关校正寄存器详见 [4.3.5 章节](#)。

功率寄存器采用二进制补码格式，32 位有符号数，其中最高位是符号位。对于视在功率，最高位始终为 0。功率参数更新的周期为 **RN2025B64 为 250ms，RN2025C64 为 64ms。**

对功率参数均有，**实际功率 = $K_{px} * PReg'$**

其中 K_{px} 表示转换系数， $PReg'$ 表示相应功率寄存器 $PReg$ 的补码。该运算由 MCU 完成。

瞬时功率寄存器：（新增）

偏移地址	5B8H	5BCH	5C4H	5C8H
寄存器	SPL_PA	SPL_PB	SPL_QA	SPL_QB
偏移地址	5D0H	5D4H	5DCH	5E0H
寄存器	SPL_FPA	SPL_FPB	SPL_FQA	SPL_FQB

瞬时功率寄存器更新速度为 8KHz，包括有功功率、基波有功功率、基波无功功率。计算公式同平均功率寄存器。其中 **SPL_QA 同 SPL_FQA，SPL_QB 同 SPL_FQB。**

举例：

假设电压通道输入 0.35mv 峰值的正弦波，电流通道输入峰值为 0.05mv 峰值的正弦波，两者之间的相位为 0 度，那么功率寄存器约为：

$$\text{PowerP}=0.35*0.05*0.5*2^{31}=18790481 \text{ (十进制)} =11\text{EB}851 \text{ (十六进制)}$$

4.5.5 快速脉冲计数寄存器

偏移地址	384H	388H	394H	398H	3A4H	3A8H
寄存器	PAFCnt	PBFCnt	QAFCnt	QBFCnt	SAFCnt	SBFCnt
偏移地址	4ACH	4B0H	4BCH	4C0H	4CCH	4D0H
寄存器	FPAFCnt	FPBFCnt	FQAFCnt	FQBFCnt	FSAFCnt	FSBFCnt

快速脉冲计数寄存器包括全波/基波、有功/无功/视在快速脉冲计数寄存器。是相应电能的最小累积输出。

注意 QAFCnt 读出值同 FQAFCnt, QBFCnt 读出值同 FQBFCnt, 而且 QAFCnt 和 QBFCnt 不可写。

快速脉冲计数寄存器为 18 位 (bit0-bit17) 有符号数, 其中 bit17 是符号位, 由功率符号位决定, bit18-bit31 为无效位, 固定为 0。

当输入正向, Fcnt 正向增加, Fcnt/2 和配置的 HFCONST1 及 HFCONST2 进行比较, 相等, Fcnt 清零, 相应能量寄存器增 1, 或相应的 CF 输出脉冲。当输入反向, Fcnt 负向增加, Fcnt/2 的补码和配置的 HFCONST1/2 进行比较, 相等, Fcnt 清零, 相应能量寄存器增 1, 或相应的 CF 输出脉冲。

能量寄存器输出 Fcnt 总是与 HFCONST1 做比较, CF 脉冲输出 Fcnt 可配置为与 HFCONST1 或 HFCONST2 做比较。详见 [4.5.9 章节](#) HFCONST 寄存器说明。

4.5.6 电能寄存器

偏移地址	3B4H	3B8H	3E4H	3E8H	414H	418H
寄存器	EPA	EPB	EQA	EQB	ESA	ESB
偏移地址	4DCH	4E0H	50CH	510H	53CH	540H
寄存器	FEPA	FEPB	FEQA	FEQB	FESA	FESB

RN2025 具有多种类型的能量寄存器, 包括基波(FEx)/全波(Ex)、有功(EPx)/无功(EQx)/视在(ESx), 是相应功率的积分累加输出。

其中 EQA 与 FEQA 的电能脉冲是同源的; EQB 与 FEQB 的电能脉冲是同源的。如果选择为读后清零模式, 他们的清零是独立的; 如果选择为读后不清零模式, 那么他们的值是一致的。

电能寄存器是 24 位无符号数。寄存器最小单位代表的能量为 1/EC KWh, 其中 EC 为电表常数。

与电能寄存器对应的电表常数通过 HFCONST1 寄存器配置, 做比较时, 将 HFCONST1 与合相快速脉冲计数寄存器以及各分相快速脉冲计数寄存器 Fcnt 的绝对值的高 16 位的 1/2 做比较, 如果大于等于 HFConst1 的值, 那么对应的能量寄存器增 1。

能量寄存器既可配置为清零型也可配置为累加型。寄存器位 EMUCFG.ERegCAR (0x188.19) =0, 所有能量寄存器为清零型; =1, 为累加型。缺省为清零型。

正向有功和无功电能寄存器仅计量功率大于 0 的电能; 反向有功和无功电能寄存器仅计量功率小于 0 的电能。

4.5.7 电压线频率寄存器 UFreq (0x450)

偏移地址: 450H; 寄存器: UFreq。

电压线频率寄存器是 24 位无符号数, 表示基波电压的线频率。线频率寄存器的更新周期为 32 个周波。

电压线频率计算公式: 实际频率 = femu*16/REGF

其中: femu=4.096Mhz, REGF 表示频率寄存器的值。

举例: REGF=24'h140000=十进制 1310720, 那么实际频率=4.096*10⁶*16/1310720=50Hz

4.5.8 半波有效值寄存器(0x574~0x584)

偏移地址	574H	580H	584H
寄存器	UA_HW	IA_HW	IB_HW

RN2025 提供 UA/IA/IB 3 路半波有效值寄存器，半波有效值寄存器为各通道半周波内采样点的平方和或者均方根。

注意计量单元配置寄存器 2 中的 **RMS_HW_SQRT** 可决定半波有效值寄存器输出结果的性质：

=1: 32 位半波有效值寄存器表示的是开根号的结果，低 24 位有效，高 8 位固定为 0。

=0: 32 位半波有效值寄存器表示的是原始波形平方后的结果，32 位有效，还需要软件开根号才能得到半波有效值。

如果选择了原始波形平方后的结果

半波有效值寄存器是四字节无符号数，与有效值寄存器的关系见下文。半波有效值寄存器参数更新的周期为 100hz(50HZ 应用)/120hz(60HZ 应用)。

用户软件读取半波有效值寄存器后还需做平方根运算，从而得到完整的半波有效值。以 A 相电压为例：

$$UA_HW = K_UAhw * \sqrt{\text{reg}(UA_HW)}$$

其中：

UA_HW 为 A 相电压半波有效值，此寄存器受通道增益寄存器 GSUA 影响；

reg(UA_HW) 为经过 GSUA 寄存器增益校正后 UA 半波有效值寄存器值，如果 A 相电压有效值经过了增益校正，则 UA_HW 不需要做额外的增益校正；

sqrt 为平方根运算，由软件完成；

K_UAhw 为转换系数，将测试值转换为示值，由软件完成。

如果选择了开根号后的结果，那么半波有效值寄存器 HW 与有效值寄存器 RMS (0x31C~0x338) 的关系是：
 $RMS = HW * 256 / \sqrt{4000 / Ufreq}$ ，其中 Ufreq 为基波频率。

举例：

如果半波有效值寄存器 HW（选择为开根号）等于 32'h11E090=十进制 1171600，此时频率为 50Hz，那么有效值寄存器 RMS 为：

$$RMS = 1171600 * 256 / \sqrt{80} = 299929600 / 8.94427191 = 33533148.7$$
，转换为十六进制为：28'h1FFACDC

如果选择了不做开根号的结果，那么半波有效值寄存器 HW（自乘平方并累加）与有效值寄存器 RMS (0x31C~0x338) 的关系是：

$$RMS = \sqrt{HW * 2^{16}} * 256 / \sqrt{4000 / Ufreq}$$
，其中 Ufreq 为基波频率。

建议选择基波过零做为依据计算。

4.5.9 高频脉冲常数寄存器 HFCONST(0x00~0x04)

偏移地址	00H	04H
寄存器	HFCONST1	HFCONST2
默认值	0x1000	0x1000

高频脉冲常数寄存器是 16 位无符号数，包括两个寄存器：HFCONST1 和 HFCONST2，HFCONST1 可用于计电能和最大需量计算，HFCONST2 可用于 CF 脉冲输出，或校表时小信号 CF 加速功能。

电能寄存器的增加仅和 HFCONST1 相关，和 HFCONST2 无关。做比较时，将 HFCONST1 与合相快速脉冲计数寄存器以及各分相快速脉冲计数寄存器的绝对值的高 16 位的 1/2 做比较，如果大于等于 HFConst1 的值，那么对应的能量寄存器增 1。

每个 CF 引脚都可通过 CFCFG.CFxFConst (0x184) 寄存器选择使用 HFCONST1 还是 HFCONST2 输出脉

冲。做比较时，将其与快速脉冲计数寄存器值的绝对值的高 16 位的 1/2 做比较，如果大于等于 CFCFG 选择的 HFConst 的值，那么就会有对应的 CF 脉冲输出。

在实际应用中，推荐以下两种应用方式：

1、EMUCFG.HFconst2_CFG 配置为 1，即 HFCONST1 和 HFCONST2 串联使用；此时与 HFCONST1 对应的引脚，脉冲输出仅与 HFCONST1 有关；HFCONST2 对应的引脚，脉冲输出与 HFCONST1、HFCONST2 均有关；

当 HFCONST2=1 时，HFONST1 对应引脚出脉冲速度，是 HFCONST2 对应引脚出脉冲速度的 2 倍；举例：当期望 HFCONST1 对应引脚输出脉冲速度，是 HFCONST2 对应引脚的 10 倍时，那么 HFCONST2 要配置为 0x5。

2、EMUCFG.HFconst2_CFG 配置为 0，即 HFCONST1 和 HFCONST2 相互独立；此时 HFCONST1 对应的脉冲输出引脚，脉冲输出仅与 HFCONST1 有关；HFCONST2 类似；

举例：当期望 HFCONST1 对应引脚出脉冲速度，是 HFCONST2 对应引脚的 10 倍时，需要将 HFCONST1 的值配置为 HFCONST2 的 1/10。为了避免不能整除 HFCONST2 引入误差，可在计算 HFCONST2 的值时，调整校表参数，使计算出来的 HFCONST2 的值，刚好是 10 的整数倍；

根据 PF=1.0，标准的有功功率值 P、电表常数 EC 等参数按下式计算理论 HFConst 值：

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

额定输入，PF=1.0，标准的有功功率值 $P = INT(U_{标准} * I_{标准} / 2^3)$ ，

式中电压额定输入时标准的电压有效值寄存器值为 $U_{标准}$ ，电流额定输入时标准的电流有效值寄存器值 $I_{标准}$

femu: 4.096Mhz

Un: 额定输入的电压；

Ib: 额定输入的电流；

EC: 电表常数

按计算结果配置相应的 HFConst 寄存器。

4.5.10 启动阈值寄存器 Start(0x08~0x0C)

偏移地址	08H	0CH
寄存器	Start_PS	Start_Q
默认值	0x250	0x250

RN2025 提供两个启动阈值寄存器 Start_PS 和 Start_Q。四种潜动启动方式下，Start_PS 和 StartQ 作用域如下表：

潜动启动方式	Start_PS	Start_Q
电流有效值启动方式 1 (全波电流或基波电流启动)	全波电流超过该阈值，则启动全波有功/视在电能计量； 基波电流超过该阈值，则启动基波有功/视在电能计量。	全波电流超过该阈值，则启动全波无功电能计量； 基波电流超过该阈值，则启动基波无功电能计量。
电流有效值启动方式 2 (基波电流启动)	基波电流超过该阈值，则启动全波/基波、有功/视在电能计量。	基波电流超过该阈值，则启动全波/基波无功电能计量。
功率启动方式 1 (有功或无功功率任意启动)	全波有功功率超过 Start_PS 阈值，或者全波无功功率超过 Start_Q 阈值，只要满足其中一个条件，则启动全波有功/无功/视在电能计量；	

	基波有功功率超过 Start_PS 阈值，或者基波无功功率超过 Start_Q 阈值，只要满足其中一个条件，则启动基波有功/无功/视在电能计量；	
功率启动方式 2 (有功功率启动、无功功率启动分开)	全波有功功率超过 Start_PS 阈值，则启动全波有功/视在电能计量； 基波有功功率超过 Start_PS 阈值，则启动基波有功/视在电能计量；	全波无功功率超过 Start_Q 阈值，则启动全波无功电能计量； 基波无功功率超过 Start_Q 阈值，则启动基波无功电能计量；

全波/基波电流有效值启动方式下

Start_PS 和 Start_Q 均为 16 位无符号数，做比较时，将其扩展为 32 位 0x000X_XXX0，与各相全波电流有效值 I_{xRMS} 或基波电流有效值 FI_{xRMS} 进行比较，以作启动判断。

Start_PS 和 Start_Q 计算公式：

$$I_{start} = (REGIb * (0.5 \sim 0.8) K / 2^4)$$

其中 I_{Start} 为待设置的启动电流阈值寄存器值，REGIb 为经校正后的 Ib 点的全波或基波电流有效值寄存器值，K=某相启动电流和标称电流值的比值，如启动电流为 0.001I_b，则 K=0.001。

功率启动方式 1 和 2 方式下

Start_PS 和 Start_Q 均为 16 位无符号数，做比较时，将其扩展为 24 位 0xXXXX_00，分别与 PowerP/PowerFP 或 PowerQ/PowerFQ (为 32bit 有符号数)的高 24 位的绝对值进行比较，以作启动判断。

Start_PS 计算公式：

$$P_{start} = REGPx * (0.5 \sim 0.8) * K / 2^8$$

其中 P_{Start} 为待设置的 Start_PS 启动阈值寄存器值，REGPx 为经校正后的 Ib 点的全波分相有功寄存器值，K=某相启动功率和标称功率的比值，如启动电流为 0.002I_b，则 K=0.002。

Start_Q 计算公式：

$$Q_{start} = REGQx * (0.5 \sim 0.8) * K / 2^8$$

其中 Q_{Start} 为待设置的 Start_Q 启动阈值寄存器值，REGQx 为经校正后的 Ib 点的全波分相无功寄存器值，K=某相启动功率和标称功率的比值，如启动电流为 0.002I_b，则 K=0.002。

Start_PS 复位初值和 Start_Q 相同，均为 0x250，用户可根据需要重新设定。

4.5.11 过零阈值寄存器 ZXOT(0x14)

偏移地址：14H；寄存器：ZXOT

过零阈值寄存器 ZXOT 为 16 位无符号数，做比较时，将其与各相电压电流有效值 U_x、I_x (x=A, B, C 下同) bit27-bit12 位进行比较，以作判断。ZXOT 的复位初值为 0x0073，用户可根据需要重新设定。

ZXOT 计算公式：

$$ZXOT = INT((REGI_b * K) / 2^{12})$$

其中 REGI_b 为经校正后的 Ib 点的电流有效值寄存器值，K=某相过零阈值和 Ib 点电流的比值，如过零阈值为 5%额定电流，则 K=0.05。

注：RN2025C64 应用时，必须将该寄存器配置为 0。

4.5.12 相位补偿区域设置寄存器 PRTHx(0x18~0x24)

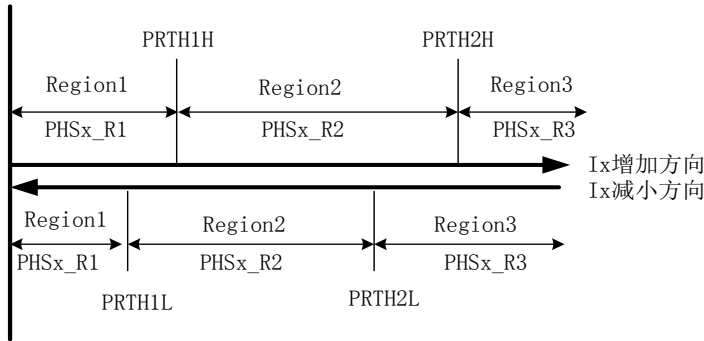
偏移地址	18H	1CH	20H	24H
寄存器	PRTH1L	PRTH1H	PRTH2L	PRTH2H
默认值	0x0	0x0	0x0	0x0

相位补偿区域设置寄存器 PRTHx 用于分段角差校正区域设置，共有两对相位分段校正电流阈值 PRTH 寄存器，分别用于设置小信号区域和大信号区域的相位校正分段点。

PRTHx 为 16 位无符号数，做比较时，将其与全波电流有效值 I_x ($x=A, B$) bit27-bit12 进行比较，以作判断。

如图所示：PRTH1L/ PRTH1H 和 PRTH2L/ PRTH2H 可配置 3 个相位校正区域的分段点，其中 Region1 以 PHSx_R1 为通道相位校正参数，以 P_x PHSL 为有功功率相位校正参数，以 Q_x PHSL 为无功功率相位校正参数；Region2 以 PHSx_R2 为通道相位校正参数，以 P_x PHSM 为有功功率相位校正参数，以 Q_x PHSM 为无功功率相位校正参数；Region3 以 PHSx_R3 为通道相位校正参数，以 P_x PHSH 为有功功率相位校正参数，以 Q_x PHSH 为无功功率相位校正参数。

在电流 I_x 增加方向，以 PRTH1H 为 Region1 和 Region2 的分段点，以 PRTH2H 为 Region2 和 Region3 的分段点；在电流 I_x 减小方向，以 PRTH1L 为 Region1 和 Region2 的分段点，以 PRTH2L 为 Region2 和 Region3 的分段点。



容错处理：对于①当写入 $PRTH2x < PRTH1x$ ，写入失败；②当写入 $PRTH1H < PRTH1L$ 或写入 $PRTH1L > PRTH1H$ ，写入失败；③当写入 $PRTH2H < PRTH2L$ 或写入 $PRTH2L > PRTH2H$ ，写入失败。

当 PRTH1L 寄存器为 0 时，不启动分段相位校正。电流通道以 PHSIx 寄存器的低 8 位做为各通道相位校正值，功率相位以 P_x PHSL、 Q_x PHSL 作为各功率的相位校正值。

4.5.13 通道相位校正寄存器 PHSU/I

偏移地址	30H	3CH	40H
寄存器	PHSUA	PHSIA	PHSIB
默认值	0x80	0x808080	0x808080

默认情况下，通道相位校正寄存器 (0x30~0x48) 的定义如下：

PHSUA 用于 U 通道的相位校正。该寄存器均为 8 位无符号数，默认值为 0x80。在 50HZ，1 LSB 代表 0.017578 %LSB 相位校正。

PHSIA-PHSIB 用于两个电流通道 3 个相位校正区域的分段校正，和相位补偿区域设置寄存器 PRTHx 结合使用。这两个寄存器均为 24 位无符号数。以 PHSIA 为例，该寄存器 3 个字节如下：

{PHSIA_R3[23:16], PHSIA_R2[15:8], PHSIA_R1[7:0]}

其中低 8 位 PHSIA_R1[7:0]代表区域 1 的相位校正值，中间 8 位 PHSIA_R2[15:8]代表区域 2 的相位校正值，高 8 位 PHSIA_R3[23:16]代表区域 3 的相位校正值。默认值为 0x808080。

相位校正范围：50HZ 下， $\pm 2.25^\circ$ 或 4.5°

相位补偿公式:

①通过 0.5L 时有功误差校正, 若某相 0.5L 功率因数角为 60° 时, Ib 点有功误差为 err , 通道角差为 θ , 则

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

若功率因数角为 30° 时, Ib 点有功误差为 err , 通道角差为 θ , 则

$$\theta = \text{Arcsin}(-\sqrt{3} * err)$$

对 50HZ, PHSUA 和 PHSIA 寄存器均有 0.017578 %LSB 的关系。

若调整 PHSUA 寄存器, 则有:

$$PHSUA = 0x80 + \text{INT}(\theta / 0.017578)$$

若调整 PHSIA 寄存器且不考虑分段校正, 则有: $PHSIA_R1[7:0] = 0x80 - \text{INT}(\theta / 0.017578)$

② 以一个通道的输入为基准, 从相角寄存器得知被校通道 1 和基准通道 2 的实际角差与标准源或标准表的角差的差值为 θ , 若被校前该通道的相位寄存器是 PHS1, 对 50HZ, 有:

$$\text{校正通道的相位 } PHS = PHS1 + \text{INT}(\theta / 0.017578)$$

锐能微计量芯升级版新增功能:

当 PHSUA[31:0] (0x30) 的高 8 位 PHSUA[31:24] 被写入 0xA7 时, 所有通道相位校正寄存器的意义发生改变, 具体定义如下:

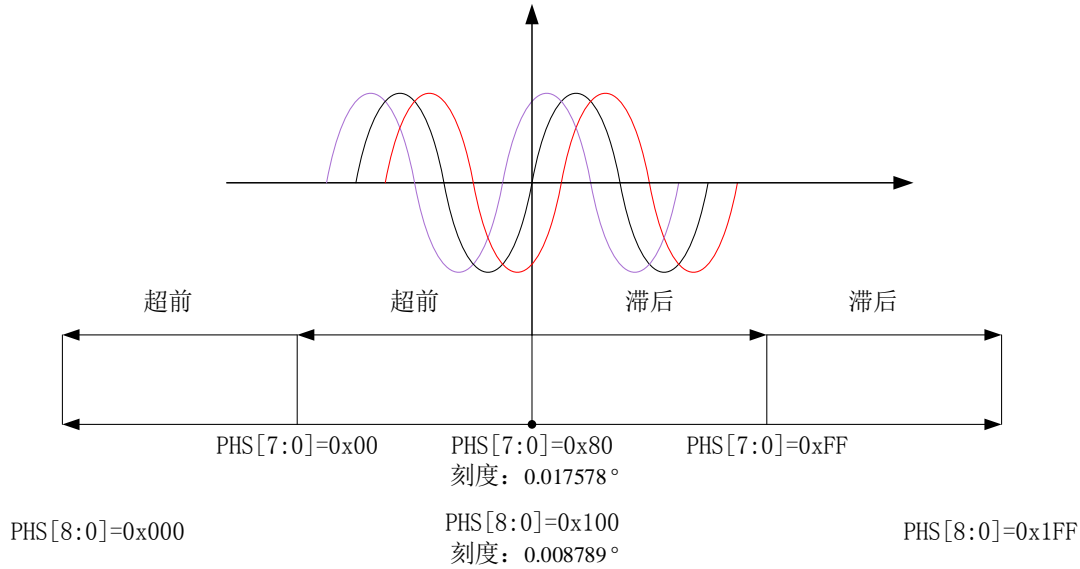
PHSUA[8:0] 为 UA 通道相位校正, 每当 PHSUA[31:24] 写入 0xA7 时, PHSUA[8:0] 的默认值就会自动变为 0x000 (在 cpu 改变此值前, 可读出为 0x000。注意 0x100 为相位偏移中间值。), 操作步骤: 首先将 PHSUA[31:0] 写入 0xA700_0000, 然后再写入相位校正值 (比如 PHSUA[31:0]=0xA700_02A5), 注意要分两次操作。在新的模式下, 应保证 PHSUA[31:24] 一直为 0xA7。

PHSIA[31:0] 的定义更改为 {2'h0, PHSIA_R3[29:20], PHSIA_R2[19:10], PHSIA_R1[9:0]}。默认状态下, 该寄存器定义见上文描述。一旦 PHSUA[31:24] 写入 0xA7, 其默认值自动从 0x808080 变为 0x00000000。在 PHSUA[31:24] 已经写入 0xA7 的情况下, 再将校正值写入 PHSIA[31:0]。其中 bit9/bit19/bit29 可读可写但是无实际意义, 只是为了跟 RN2026 (B80) 保持兼容。

PHSIB[31:0] 的定义更改为 {2'h0, PHSIB_R3[29:20], PHSIB_R2[19:10], PHSIB_R1[9:0]}。默认状态下, 该寄存器定义见上文描述。一旦 PHSUA[31:24] 写入 0xA7, 其默认值自动从 0x808080 变为 0x00000000。在 PHSUA[31:24] 已经写入 0xA7 的情况下, 再将校正值写入 PHSIB[31:0]。其中 bit9/bit19/bit29 可读可写但是无实际意义, 只是为了跟 RN2026 (B80) 保持兼容。

在 PHSUA[31:24] 已经写入 0xA7 的情况下, 所有的通道相位校正寄存器都从 8bit 变更为 9bit。校正刻度从 0.017578 %LSB 提升至 0.008789 %LSB, 校正范围仍然是 $\pm 2.25^\circ$ 或 4.5° 。

校正公式同上文默认情况, 只需要将 0.017578 修改为 0.008789, 即可实现 $\pm 2.25^\circ$ 的校正范围。如果要实现 4.5° 的校正范围, 只需要将 PHSU 的寄存器写为 0, 然后改变 PHSI (从 0x000 到 0x1ff); 或者将 PHSI 的寄存器写为 0, 然后改变 PHSU (从 0x000 到 0x1ff)。



4.5.14 通道增益寄存器 GSU/I

偏移地址	4CH	58H	5CH
寄存器	GSUA	GSIA	GSIB
默认值	0x0	0x0	0x0

通道增益寄存器可用于有效值、功率的增益校正。通道增益寄存器为 2 字节有符号数，采用二进制补码形式，最高位为符号位，表示范围(-1, +1)。

校正公式：

如果 $\text{RegGain} > 2^{15}$ ，则 $\text{Gain} = (\text{RegGain} - 2^{16}) / 2^{15}$ ；否则 $\text{Gain} = \text{RegGain} / 2^{15}$ ；

其中 RegGain 为通道增益寄存器值。

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

$$IB' = IB + IB * \text{Gain}$$

注意该公式有效范围：保证在乘以通道增益后信号仍在 ADC 满量程范围内

通道增益寄存器适用于功率法校表中的增益校正。

4.5.15 通道直流 OFFSET 校正寄存器 DCOSx (修改)

偏移地址	68H	74H	78H
寄存器	DCOS_UA	DCOS_IA	DCOS_IB
默认值	0x0	0x0	0x0

通道直流 OFFSET 校正寄存器用于在测试直流量时替代高通滤波消除失调误差。直流 OFFSET 寄存器所表示的直流分量等效于直接在采样波形上叠加直流分量。直流测量使用时，需将 HPF 关闭。

升级版修改点：该寄存器从高通之后调整到高通之前，当 AUTODC_EN.MMU_DC_DIS (0x1B8[7]) = 0，如果采样通路高通关闭，DCOS 寄存器会影响 EMU 通道有效值、功率、能量等参数；如果采样通路高通使能，DCOS 寄存器不会影响 EMU 通道有效值、功率、能量等参数。

当 AUTODC_EN.MMU_DC_DIS (0x1B8[7]) = 0，软件需要保证上述寄存器符合预期；如果上述寄存器非法写入值，会影响到计量。

当 AUTODC_EN.MMU_DC_DIS (0x1B8[7]) = 1，DCOS 寄存器对 EMU 通道不起作用。默认情况下，AUTODC_EN.MMU_DC_DIS = 1。

直流 OFFSET 校正寄存器均为两字节有符号数，采用二进制补码形式，最高位为符号位。

DCOS 寄存器对有效值的影响：

在直流计量条件下，以 IA 通道为例，直流 DCOS_IA 与 IA 通路有效值的关系如下：

$$IA' = \sqrt{(IA^2 + ((DCOS_IA - IAd0) * 2^{12}) / 2^{27})^2}$$

其中：

DCOS_IA：是指 IA 通道 DCOS 寄存器值，可以是 0~0xffff 之间任意值

IAd0：是指 IA 通道空载且 ADC 输入短接到地情况下，DCOS 自动校正完成后的 DCOS_IA 寄存器值

IA：是指 IA 通道实际输入的任意不超过满量程波形的有效值的归一化值

IA'：是指 IA 通道实际读出的有效值寄存器归一化值

替代高通滤波消除失调误差时，用户可利用直流 OFFSET 自动校正功能得到 DCOS 寄存器值，说明详见[直流 OFFSET 自动校正使能寄存器](#)章节。

4.5.16 有效值 OFFSET 校正寄存器

偏移地址	84H	90H	94H	F4H	100H	104H
寄存器	UA_OS	IA_OS	IB_OS	FUA_OS	FIA_OS	FIB_OS
默认值	0x0	0x0	0x0	0x0	0x0	0x0

有效值 Offset 校正寄存器用于电压/电流/基波电压/基波电流有效值小信号精度的校正。

有效值 Offset 校正寄存器均为两字节有符号数，采用二进制补码形式，最高位为符号位。

有效值 Offset 寄存器仅影响相关通道的有效值，并对视在功率、视在电能有影响。

校正公式：

以 IA 通道为例，假设校正之前 A 相电流有效值寄存器 IA，校正之后为 IA'，则有：

$$IA' = \sqrt{(\text{Abs}(IA^2 + IA_OS * 2^{14}))}$$

4.5.17 功率增益寄存器

偏移地址	A0H	A4H	ACH	B0H	B8H	BCH
寄存器	GPA	GPB	GQA	GQB	GSA	GSB
默认值	0x0	0x0	0x0	0x0	0x0	0x0
偏移地址	10CH	110H	118H	11CH	124H	128H
寄存器	GFPA	GFPB	GFQA	GFQB	GFSA	GFSB
默认值	0x0	0x0	0x0	0x0	0x0	0x0

功率增益寄存器用于有功/无功/视在/基波有功/基波无功/基波视在功率的增益校正。功率增益寄存器均为两字节有符号数，采用二进制补码形式，最高位为符号位。

注意：RN2025 的 GQA 与 FGQA 功能相同，GQB 与 FGQB 功能相同。都是作用于基波无功，如果 GQx 为 0，则填入 FGQx 的值会对最终的无功功率起作用；如果 GQx 不为 0，则以 GQx 填入的值为准计算无功功率。

校正公式为： $P1 = P0(1 + GP)$

$$Q1 = Q0(1 + GQ)$$

$$S1 = S0(1 + GS)$$

其中 P1、Q1、S1 为校正后的有功、无功、视在功率；P0、Q0、S0 为校正前的功率；GP、GQ、GS 分别为有功、无功、视在功率增益校正寄存器的归一化值。

功率增益寄存器适用于传统的脉冲校表法功率增益校正，校正误差范围为 -50%~+∞，校正方法以 A 相 PF=1.0

误差校正为例：假设 A 相 Ib 点输入时，标准表的有功误差读数为 err，则 A 相有功功率增益校正寄存器 GPA 的计算方法如下：

$$Gain = \frac{-Err}{1 + Err}$$

如果 Gain ≥ 0，则 GPA = INT[Gain * 2¹⁵];

如果 Gain < 0，则 GPA = INT[2¹⁶ + Gain * 2¹⁵]

4.5.18 功率相位校正寄存器

偏移地址	C4H	C8H	D0H	D4H
寄存器	PA_PHSL	PB_PHSL	QA_PHSL	QB_PHSL
默认值	0x0	0x0	0x0	0x0
偏移地址	130H	134H	13CH	140H
寄存器	FPA_PHS	FPB_PHS	FQA_PHS	FQB_PHS
默认值	0x0	0x0	0x0	0x0
偏移地址	1CCH /640H	1D0H /644H	1D4H /648H	1D8H /64CH
寄存器	PA_PHSM	PA_PHSH	PB_PHSM	PB_PHSH
默认值	0x0	0x0	0x0	0x0
偏移地址	1E4H /658H	1E8H /65CH	1ECH /660H	1F0H /664H
寄存器	QA_PHSM	QA_PHSH	QB_PHSM	QB_PHSH
默认值	0x0	0x0	0x0	0x0

有功功率相位校正寄存器用于在有功功率通道进行有功相位校正。无功相位校正寄存器用于在无功功率通道进行无功相位校正。功率相位校正寄存器采用两字节二进制补码形式，最高位为符号位。

全波功率相位校正寄存器支持功率分段校正，和相位补偿区域设置寄存器 PRTHx 和结合使用。当 PRTH1L 寄存器为 0 时，不启动分段相位校正；当 PRTH1L 寄存器不等于 0 时，PRTHx 将功率相位校正区域分为三段，低段区域 1、中段区域 2、高段区域 3，参见 4.5.12 章节。

此时，Px_PHSL (x=A,B 下同) 用于区域 1 低段的有功功率相位校正，Px_PHSM 用于区域 2 中段的有功功率相位校正，Px_PHSH 用于区域 3 高段的有功功率相位校正。

Qx_PHSL (x=A,B 下同) 用于区域 1 低段的无功功率相位校正，Qx_PHSM 用于区域 2 中段的无功功率相位校正，Qx_PHSH 用于区域 3 高段的无功功率相位校正。

注意，有功、无功分段相位校正寄存器 Px_PHSM、Px_PHSH、Qx_PHSM、Qx_PHSH (0x1CC~0x1F8H) 和自定义功率寄存器 D2FP00-12 复用 1CCH-1F8H 地址，当 D2FCFG[3:0] 不等于 4'b1001 时，该段地址为中高段有功、无功相位校正寄存器。当 D2FCFG[3:0] 等于 4'b1001 时，使用地址 0x640H~0x66CH 替代。

如果计量单元配置寄存器 EMUCFG 的 bit26 (FP_PHS_SEL) 位配置为 0，基波功率相位校正使用基波独立的一套相位校正寄存器 (0x130~0x144)，不具备分段校正功能；如果该位配置为 1，则基波功率相位校正使用全波相位校正同样的值，如果全波相位校正配置了分段功能，则基波也具备同样功能。

校正公式为：

$$P2 = P1 + P_PHS * Q1$$

$$Q2 = Q1 - Q_PHS * P1$$

其中 P1 为补偿前的有功功率、P2 为补偿后的有功功率、Q1 为补偿前的无功功率、Q2 为补偿后的无功功率。P_PHS、Q_PHS 分别为有功、无功功率相位校正寄存器的归一化值。

校表方法：以 A 相为例，电表在 Ib 点，PF=1.0 下，有功增益校正已完成；切换源输出后，

◇ 当互感器角差较小时，即相位初始误差常规值小于 1%，
 若功率因数角在 60° 附近，即 PF≈0.5L 下，Ib 点标准表有功初始误差记为 err，
 此时可以使用近似校正公式

$$\lambda \approx -err/\sqrt{3}$$

若 $\lambda \geq 0$ ，则 $PA_PHS = \lambda * 2^{15}$ ；若 $\lambda < 0$ ，则 $PA_PHS = \lambda * 2^{15} + 2^{16}$

若 $\lambda \geq 0$ ，则 $QA_PHS = \lambda * 2^{15}$ ；若 $\lambda < 0$ ，则 $QA_PHS = \lambda * 2^{15} + 2^{16}$

若功率因数角在 30° 附近，Ib 点标准表有功误差记为 Err，则

$$\lambda = -\sqrt{3} * Err$$

若 $\lambda \geq 0$ ，则 $PA_PHS = \lambda * 2^{15}$ ；若 $\lambda < 0$ ，则 $PA_PHS = \lambda * 2^{15} + 2^{16}$

若 $\lambda \geq 0$ ，则 $QA_PHS = \lambda * 2^{15}$ ；若 $\lambda < 0$ ，则 $QA_PHS = \lambda * 2^{15} + 2^{16}$

◇ 当互感器角差较大时，建议通过以下流程对计量芯片进行校正，步骤如下：

1、对功率增益寄存器、功率相位寄存器清零；

计算 hfconst 校正值，填写到 hfconst 寄存器；公式参考“4.5.9 高频脉冲常数”章节。

计算功率转换系数 $kp = 3.6 * 10^{12} * 8.192 / [(2^{31}) * 32 * hfconst * EC]$ ，EC 为脉冲常数；

2、获取分相的阻性初始误差 err1，获取分相的电压电流转换系数 ku、ki；

方法：台体输出 Un/Ib/1.0/合相。以 A 相为例，

读出 A 相的功率寄存器值 P_{寄存器值}，标准表显示的 A 相功率值 P_{标准表值}，单位 W；

则 $err1 = (P_{寄存器值} * Kp - P_{标准表值}) / P_{标准表值}$ ；

读取 A 相的电压寄存器值 U_{寄存器值}，标准表显示的 A 相电压值 U_{标准表值}，单位 V；

则 $ku_A = U_{标准表值} / U_{寄存器值}$ ；

3、获取分相的感性初始误差 err2，

方法：台体输出到 Un/Ib/0.5L/合相。以 A 相为例，

读出 A 相的功率寄存器值 P_{寄存器值}，标准表显示的 A 相功率值 P_{标准表值}，单位 W；

则 $err2 = (P_{寄存器值} * Kp - P_{标准表值}) / P_{标准表值}$ ；

4、计算功率相位校正归一化值 λ ，以 A 相为例，

$$\lambda = \frac{err1 - err2}{1.732 * (1 + err1)}$$

5、计算需要填入到功率相位寄存器的值：以 A 相为例，

若 $\lambda \geq 0$ ，则 $PA_PHS = \lambda * 2^{15}$ ；若 $\lambda < 0$ ，则 $PA_PHS = \lambda * 2^{15} + 2^{16}$

6、计算需要填入到功率增益寄存器的值：以 A 相为例，

$$Gain = \frac{1}{(1 + \lambda^2) * (1 + err1)} - 1$$

若 $Gain \geq 0$ ，则 $GP = INT[Gain * 2^{15}]$ ；若 $Gain < 0$ ，则 $GP = INT[2^{16} + Gain * 2^{15}]$

将计算出来的 GP 值分别填入到 GPA/GQA 寄存器；

4.5.19 功率 OFFSET 校正寄存器

偏移地址	DCH	E0H	E8H	ECH
寄存器	PA_OS	PB_OS	QA_OS	QB_OS

默认值	0x0	0x0	0x0	0x0
偏移地址	148H	14CH	154H	158H
寄存器	FPA_OS	FPB_OS	FQA_OS	FQB_OS
默认值	0x0	0x0	0x0	0x0

功率 OFFSET 校正寄存器适合全波有功/全波无功/基波有功/基波无功的小信号的精度校正。

当 EMUCFG.P_OS_SEL=0 时，PA_OS/PB_OS 功率 OFFSET 校正寄存器为两字节有符号数，当 EMUCFG.P_OS_SEL=1 时，PA_OS/PB_OS 功率 OFFSET 校正寄存器为四字节有符号数，默认两字节有符号数。

其他功率 OFFSET 校正寄存器均为两字节有符号数。

注意 Qx_OS 与 FQx_OS 功能等同。当 Qx_OS 写入非 0 值时，则 Qx_OS 对无功功率起作用；如果 Qx_OS 为 0，则 FQx_OS 对无功功率起作用。

功率 OFFSET 校正寄存器采用二进制补码形式，最高位为符号位。

校正公式：

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

$$PA' = PA + PA_OS$$

其中 PA 是 32bit 有符号数，PA_OS 默认是 16bit 有符号数。两者相加时，将 PA_OS 的符号位扩展 16 位再相加。

当计量单元配置寄存器 EMUCFG (0x188) 的 P_OS_SEL=1 时，PA_OS/PB_OS 从 16 位有符号数变更为 32 位有符号数，功率 offset 的校正范围扩大。PA 和 PA_OS 都是 32bit 有符号数，两者可直接相加。

4.5.20 电压暂降阈值寄存器 SAGCFG(0x160)

偏移地址：160H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:24	reserved	保留		
23:16	SAGCYC[7:0]	电压暂降时间阈值。说明见下。 SAGCYC[7:0]=00H，关闭电压暂降判断功能。	R/W	0
15:0	SAGLVL[15:0]	电压暂降峰值阈值配置。 当电压采样值绝对值的峰值的高 16 位小于 SAGLVL[15:0]，且持续时间为 SAGCYC[7:0] 个半周波数，判定该相电压暂降。 SAGLVL[15:0]=0000H，则关闭电压暂降判断功能。	R/W	0

SAGLVL 计算公式：

$$SAGLVL = INT((\sqrt{2} * REGV_n * K) / 2^{12})$$

其中 REGV_n 为经校正后的额定电压输入的电压有效值寄存器值，K=电压暂降电压和额定电压的比值，如电压暂降电压为 10% 额定电压，则 K=0.1。

注：必须配置 SAGCYC>1，且阈值配置 SAGLVL>0xE7。

4.5.21 过压阈值寄存器 OVLVL(0x164)

偏移地址：164H；字长：2 字节；默认值：0x00

当电压采样值的绝对值高 16 位大于 OVLVL[15:0]，判定电压过压。电压过压将导致 PHASES 寄存器中 OVUA 寄存器位置 1 同时 EMMIF 寄存器中的 OVIIF 寄存器位置 1。若 EMMIE 寄存器中的 OVIIE=1，OVIIF 置 1 会导致中断。OVIIF 写 1 清零并清中断，同时清 OVUA 标志。详见 [PHASES 寄存器](#) 和 [EMMIE 寄存器以及 EMMIF 寄存](#)

器说明。

OVLVL[15:0]=0000H, 则关闭电压过压判断功能。

OVLVL 计算公式:

$$OVLVL = INT((\sqrt{2} * REGV_n * K / 2^{12}))$$

其中 REGV_n 为经校正后的额定电压输入的全波电压有效值寄存器值, K=某相过压电压和额定电压的比值, 如过压电压为 1.2 倍额定电压, 则 **K=1.2**。

4.5.22 过流阈值寄存器 OILVL(0x168)

偏移地址: 168H; 有效字长: 2字节; 默认值: 0x0

当某路电流采样值的绝对值高16位大于OILVL[15:0], 判定该相电流过流。

某路电流过流将导致PHASES寄存器中该路OVI_x (x=A, B) 寄存器位置1同时EMUIF寄存器中的OVIIF寄存器位置1。若EMUIE寄存器中的OVIIE=1, OVIIF置1会导致中断。OVIIF写1清零并清中断, 同时清OVI_x标志。详见PHASES寄存器和EMUIE寄存器以及EMUIF寄存器说明。

OILVL[15:0]≠0x0, 开启电流过流判断功能。OILVL[15:0]=0x0, 则关闭电流过流判断功能。默认关闭。

OILVL 计算公式:

$$OILVL = INT((\sqrt{2} * REGI_b * K / 2^{12}))$$

其中 REGI_b 为经校正后的额定电流输入的全波电流有效值寄存器值, K=某路过流电流和额定电流的比值, 如过流电流为 10 倍额定电流, 则 K=10。

4.5.23 CF 脉冲配置寄存器 CFCFG(0x180)

偏移地址: 0x180; 默认值: 0x43210

位	位名称	功能描述	读写标志	复位值	
31:27	Reserved	保留。不可写。	R	0	
26:22	Reserved	保留位, 可读可写, 无实际意义。	R/W	0	
21	Reserved	保留位, 可读可写, 无实际意义。	R/W	0	
20	Reserved	保留位, 可读可写, 无实际意义。	R/W	0	
19	CF4FConst	CF4 高频脉冲选择。=0, 以 HFCONST1 为高频脉冲常数输出脉冲。=1, 以 HFCONST2 为高频脉冲常数输出脉冲。默认为 0。	R/W	0	
18:16	CF4Cfg [2:0]	CF4 输出脉冲控制位。	R/W	3'b100	
		配置			说明
		3'b000			输出全波有功脉冲。
		3'b001			输出基波无功脉冲。
		3'b010			输出全波视在脉冲。
		3'b011			输出基波有功脉冲。
		3'b100			输出基波无功脉冲。
		3'b101			输出基波视在脉冲。
		3'b110			禁止 CF4 引脚输出脉冲
3'b111	禁止 CF4 引脚输出脉冲。				

15	CF3FConst	CF3 高频脉冲选择。=0, 以 HFCONST1 为高频脉冲常数输出脉冲。=1, 以 HFCONST2 为高频脉冲常数输出脉冲。默认为 0。	R/W	0	
14:12	CF3Cfg [2:0]	CF3 输出脉冲控制位。	R/W	3'b011	
		配置			说明
		3'b000			输出全波有功脉冲。
		3'b001			输出基波无功脉冲。
		3'b010			输出全波视在脉冲。
		3'b011			输出基波有功脉冲。
		3'b100			输出基波无功脉冲。
		3'b101			输出基波视在脉冲。
		3'b110			禁止 CF3 引脚输出脉冲
3'b111	禁止 CF3 引脚输出脉冲。				
11	CF2FConst	CF2 高频脉冲选择。=0, 以 HFCONST1 为高频脉冲常数输出脉冲。=1, 以 HFCONST2 为高频脉冲常数输出脉冲。默认为 0。	R/W	0	
10:8	CF2Cfg[2:0]	CF2 输出脉冲控制位。	R/W	3'b010	
		配置			说明
		3'b000			输出全波有功脉冲。
		3'b001			输出基波无功脉冲。
		3'b010			输出全波视在脉冲。
		3'b011			输出基波有功脉冲。
		3'b100			输出基波无功脉冲。
		3'b101			输出基波视在脉冲。
		3'b110			禁止 CF2 引脚输出脉冲
3'b111	禁止 CF2 引脚输出脉冲。				
7	CF1FConst	CF1 高频脉冲选择。=0, 以 HFCONST1 为高频脉冲常数输出脉冲。=1, 以 HFCONST2 为高频脉冲常数输出脉冲。默认为 0。	R/W	0	
6:4	CF1Cfg[2:0]	CF1 输出脉冲控制位。	R/W	3'b001	
		配置			说明
		3'b000			输出全波有功脉冲。
		3'b001			输出基波无功脉冲。
		3'b010			输出全波视在脉冲。
		3'b011			输出基波有功脉冲。
		3'b100			输出基波无功脉冲。
		3'b101			输出基波视在脉冲。
		3'b110			禁止 CF1 引脚输出脉冲
3'b111	禁止 CF1 引脚输出脉冲。				
3	CF0FConst	CF0 高频脉冲选择。=0, 以 HFCONST1 为高频脉冲常数输出脉冲。=1, 以 HFCONST2 为高频脉冲常数输出脉冲。默认为 0。	R/W	0	

2:0	CF0Cfg[2:0]	CF0 输出脉冲控制位。				
		配置	说明			
		3'b000	输出全波有功脉冲。			
		3'b001	输出基波无功脉冲。			
		3'b010	输出全波视在脉冲。	R/W	3'b000	
		3'b011	输出基波有功脉冲。			
		3'b100	输出基波无功脉冲。			
		3'b101	输出基波视在脉冲。			
		3'b110	禁止 CF0 引脚输出脉冲			
		3'b111	禁止 CF0 引脚输出脉冲。			

4.5.24 计量单元配置寄存器 EMUCFG(0x188)

偏移地址：0x188；默认值：0x400000

计量单元配置寄存器 EMUCFG，用于配置 EMU

位	位名称	功能描述	读写标志	复位值
31	Reserved	保留	R	0
30	P_OS_SEL	=0: PA_OS/PB_OS/PC_OS 是 16 位有符号数； =1: PA_OS/PB_OS/PC_OS 是 32 位有符号数；	R/W	0
29:28	WAVUD_CFG	=00: 波形采样中断 8Khz 速率刷新； =01: 波形采样中断 4Khz 速率刷新； =10: 波形采样中断 2Khz 速率刷新； =11: 波形采样中断 1Khz 速率刷新；	R/W	0
27	Reserved	保留位，可读可写，无实际意义。	R/W	0
26	FP_PHS_SEL	=0: 基波功率相位校正使用基波独立的一套相位校正寄存器（0x130~0x140），不具备分段校正功能； =1: 基波功率相位校正使用全波相位校正同样的值，如果全波相位校正配置了分段功能，则基波也具备同样功能。	R/W	0
25	Reserved	保留位，可读可写，无实际意义。	R/W	0
24	Reserved	保留位，可读可写，无实际意义。	R/W	0
23:22	PStart_CFG[1:0]	=0x: 采用电流有效值启动方式，分相电流超过阈值则启动该相有功无功视在的计量。 =10: 采用功率启动方式 1，分相有功功率或无功功率超过阈值则启动该相计量，分相有功功率和无功功率都低于阈值则关闭该相计量。功率阈值寄存器与启动电流寄存器复用。 =11: 采用功率启动方式 2，分相有功功率超过阈值则	R/W	2'b01

		启动该相有功功率计量，分相无功功率超过阈值则启动该相无功，只要有一个超过阈值则启动该相视在，即：有功和无功的阈值是分开的。 注：单相表应用，建议该寄存器配置为 11。选择功率启动方式 2。		
21	EQADDMOD	=0：当功率为正时，无功电能快速脉冲计数器加 1； 当功率为负时，无功电能快速脉冲计数器减 1； =1：不管功率正负，无功电能快速脉冲计数器均加 1。	R/W	0
20	EPADDMOD	=0：当功率为正时，有功电能快速脉冲计数器加 1； 当功率为负时，有功电能快速脉冲计数器减 1； =1：不管功率正负，有功电能快速脉冲计数器均加 1。	R/W	0
19	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	Reserved	保留位，可读可写，无实际意义。	R/W	0
17	Reserved	保留位，可读可写，无实际意义。	R/W	0
16	NoLoadCFG	起动方式选择。=0，全波起动以全波电流有效值与起动阈值做比较判断。=1，全波起动以基波电流有效值与起动阈值做比较判断。默认值为 0。	R/W	0
15	Reserved	保留位，可读可写，无实际意义。	R/W	0
14:12	Reserved	保留位，可读可写，无实际意义。	R/W	0
11	FREQ_SEL	=0: SAG、短路、过压，过流，欠压，归零周期为 50Hz 应用 =1: SAG、短路、过压，过流，欠压，归零周期为 60Hz 应用	R/W	0
10:08	Reserved	保留位，可读可写，无实际意义。	R/W	0
7	HFCNST2_CFG	默认值为 0； =0：HFCNST1 和 HFCNST2 独立并行使用； =1： HFCNST1 和 HFCNST2 串联使用，当由 HFCNST1 配置的脉冲输出时，由 HFCNST2 配置的脉冲计数器加 1，当计数器值累加到 HFCNST2 的寄存器值时，由 HFCNST2 配置的脉冲才加 1。	R/W	0
6:5	Reserved	保留位，可读可写，无实际意义。	R/W	0
4	HPFOFF_IB	采样通道 IB 高通使能选择位。=0，选择高通使能； =1，选择高通关闭。默认状态选择高通使能。	R/W	0
3	HPFOFF_IA	采样通道 IA 高通使能选择位。=0，选择高通使能； =1，选择高通关闭。默认状态选择高通使能。	R/W	0

2	Reserved	保留位，可读可写，无实际意义。	R/W	0
1	HPFOFF_AIN4	A 版为 AIN4 的高通使能开关； B 版取消该功能，用户不要改变此默认值；	R/W	0
0	HPFOFF_UA	采样通道 UA 高通使能选择位。=0，选择高通使能； =1，选择高通关闭。默认状态选择高通使能。	R/W	0

4.5.25 计量控制寄存器 EMUCON(0x18C)

偏移地址：18CH；默认值：0x0

计量控制寄存器 EMUCON 用于控制各路（A/B）有功/无功/RMS 视在/基波有功/基波无功/基波视电能计量的使能关闭。

位	位名称	功能描述	读写标志	复位值
31:23	Reserved	保留。	R	0
22	Reserved	保留位，可读可写，无实际意义。	R/W	0
21:20	FSRUN[1:0]	B/A 路基波视在计量使能位。 FSRUN _x =1，使能该路 RMS 基波电能视在计量 FSRUN _x =0，关闭该路 RMS 基波电能视在计量	R/W	2'b00
19	Reserved	保留。	R	0
18	Reserved	保留位，可读可写，无实际意义。	R/W	0
17:16	FQRUN[1:0]	B/A 路基波无功计量使能位。 FQRUN _x =1，使能该路基波无功电能计量 FQRUN _x =0，关闭该相基波无功电能计量	R/W	2'b00
15	Reserved	保留。	R	0
14	Reserved	保留位，可读可写，无实际意义。	R/W	0
13:12	FPRUN[1:0]	B/A 路基波有功计量使能位。 FPRUN _x =1，使能该路基波有功电能计量 FPRUN _x =0，关闭该路基波有功电能计量	R/W	2'b00
11	Reserved	保留。	R	0
10	Reserved	保留位，可读可写，无实际意义。	R/W	0
9:8	SRUN[1:0]	B/A 路视在计量使能位。 SRUN _x =1，使能该路 RMS 视在电能计量 SRUN _x =0，关闭该路 RMS 视在电能计量	R/W	2'b00
7	Reserved	保留。	R	0
5	Reserved	保留位，可读可写，无实际意义。	R/W	0
5:4	QRUN[1:0]	B/A 路无功计量使能位。 QRUN _x =1，使能该路无功电能计量。注意：由于只有基波无功，该寄存器实际对基波无功计量起作用。	R/W	2'b00

		QRUN _x =0, 关闭该路无功电能计量。注意: 由于只有基波无功, 该寄存器实际对基波无功计量起作用。		
3	Reserved	保留。	R	0
2	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
1:0	PRUN[1:0]	B/A 路有功计量使能位。PRUN0 控制 A 路, PRUN1 控制 B 路。 PRUN _x =1, 使能该相有功电能计量 PRUN _x =0, 关闭该相有功电能计量	R/W	2'b00

4.5.26 EMU 中断寄存器

4.5.26.1 EMUIE 寄存器(0x190)

偏移地址: 190H;

该寄存器为EMU中断使能寄存器。在相应中断允许位配置为1且IF清零, 当该事件发生时, 相应EMUIF置位并产生EMU中断1(中断号: 24 中断地址: 0xA0)。

位	位名称	功能描述	读写标志	复位值
31:19	Reserved	保留	R	0
18	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
17	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
16	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
15	OIIIIE	过压过流中断使能。=1, 使能。=0, 不允许。默认为0, 下同。	R/W	0
14	SAGIE	电压暂降中断使能。	R/W	0
13	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
12	CF4IE	CF4 脉冲输出中断使能。	R/W	0
11	CF3IE	CF3 脉冲输出中断使能。	R/W	0
10	CF2IE	CF2 脉冲输出中断使能。	R/W	0
9	CF1IE	CF1 脉冲输出中断使能。	R/W	0
8	CF0IE	CF0 脉冲输出中断使能。	R/W	0
7	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
6	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
5	ZXIBIE	通道 IB 正向过零中断使能。	R/W	0
4	ZXIAIE	通道 IA 正向过零中断使能。	R/W	0
3	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
2	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
1	ZXUAIE	通道 UA 正向过零中断使能。	R/W	0
0	WAVUDIE	采样波形中断使能。	R/W	0

4.5.26.2 EMUIF 寄存器(0x194)

偏移地址: 194H;

该寄存器为EMU模式下事件状态寄存器。在EMU下, 当某事件发生时, 相应IF位置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: 24 中断地址: 0xA0)。

位	位名称	功能描述	读写标志	复位值
31:19	Reserved	保留	R	0
18	Reserved	保留位，无实际意义。	R	0
17	Reserved	保留位，无实际意义。	R	0
16	Reserved	保留位，无实际意义。	R	0
15	OIIF	过压、过流中断标志。当电流中的任一路发生过流，或者电压发生过压，该位置 1。若 OIIE 中断使能，=1 会导致 EMU 中断。 该位写 1 清零同时清中断和清 PHASES 状态寄存器中的 OIIA、OIIB、OVUA 状态标志位。	R/W	-
14	SAGIF	电压暂降中断标志。 若 SAGIE 中断使能，=1 会导致 EMU 中断。 该位写 1 清零同时清中断和 PHASES 状态寄存器中的 SAGUA 状态标志位。	R/W	-
13	Reserved	保留位，可读可写，无实际意义。	R/W	-
12	CF4IF	CF4 脉冲输出标志。写 1 清零。若 CF4IE=1，清零同时清中断。	R/W	-
11	CF3IF	CF3 脉冲输出标志。写 1 清零。若 CF3IE=1，清零同时清中断。	R/W	-
10	CF2IF	CF2 脉冲输出标志。写 1 清零。若 CF2IE=1，清零同时清中断。	R/W	-
9	CF1IF	CF1 脉冲输出标志。写 1 清零。若 CF1IE=1，清零同时清中断。	R/W	-
8	CF0IF	CF0 脉冲输出标志。写 1 清零。若 CF0IE=1，清零同时清中断。	R/W	-
7	Reserved	保留位，无实际意义。	R	0
6	Reserved	保留位，无实际意义。	R	0
5	ZXIBIF	通道 IB 正向过零标志。写 1 清零。若 ZXIBIE=1，清零同时清中断。	R/W	-
4	ZXIAIF	通道 IA 正向过零标志。写 1 清零。若 ZXIAIE=1，清零同时清中断。	R/W	-
3	Reserved	保留位，无实际意义。	R	0
2	Reserved	保留位，无实际意义。	R	0
1	ZXUAIF	通道 UA 正向过零标志。写 1 清零。若 ZXUAIE=1，清零同时清中断。	R/W	-
0	WAVUDIF	波形采样标志。8Khz 速率刷新。写 1 清零。若 WAVUDIE=1，清零同时清中断。	R/W	-

注：电流过零标志依赖于电压过零，只有存在电压过零标志的情况下，才能产生电流过零标志，但并不要求是对应相电压过零标志。

4.5.27 功率方向寄存器 PQSign(0x198)

偏移地址：198H;

PQSign 寄存器为全波有功/全波无功/基波有功/基波无功方向寄存器。方向以各分相以及合相平均功率符号为判据，并和平均功率寄存器同步更新，更新周期为 250ms。

位	位名称	功能描述	读写标志	复位值
31:16	reserved	保留	R	0
15	reserved	保留	R	0
14	reserved	保留	R	0
13	FQBSIGN	=0, 表示基波 B 路无功符号为+; =1, 表示基波 B 路无功符号为-。	R	-
12	FQASIGN	=0, 表示基波 A 路无功符号为+; =1, 表示基波 A 路无功符号为-。	R	-
11	reserved	保留	R	0
10	reserved	保留	R	0
9	FPBSIGN	=0, 表示基波 B 路有功符号为+; =1, 表示基波 B 路有功符号为-。	R	-
8	FPASIGN	=0, 表示基波 A 路有功符号为+; =1, 表示基波 A 路有功符号为-。	R	-
7	reserved	保留	R	0
6	reserved	保留	R	0
5	QBSIGN	=0, 表示 B 路无功符号为+; =1, 表示 B 路无功符号为-。同 bit13 基波 B 路无功符号。	R	-
4	QASIGN	=0, 表示 A 路无功符号为+; =1, 表示 A 路无功符号为-。同 bit12 基波 A 路无功符号。	R	-
3	reserved	保留	R	0
2	reserved	保留	R	0
1	PBSIGN	=0, 表示 B 路有功符号为+; =1, 表示 B 路有功符号为-。	R	-
0	PASIGN	=0, 表示 A 路有功符号为+; =1, 表示 A 相路功符号为-。	R	-

4.5.28 潜动状态标志寄存器 Noload(0x19C)

偏移地址：19CH;

Noload 寄存器表示各分相全波有功视在/全波无功/基波有功视在/基波无功潜动状态。该寄存器更新周期为 250ms。

位	位名称	功能描述	读写标志	复位值
31:15	Reserved	Reserved	R	0
14	Reserved	Reserved	R	0
13	FNoQldB	=0, 表示基波 B 路无功不处于潜动状态; =1, 表示基波 B 路无功处于潜动状态。	R	-
12	FNoQldA	=0, 表示基波 A 路无功不处于潜动状态; =1, 表示基波 A 路无功处于潜动状态。	R	-
11	Reserved	Reserved	R	-
10	Reserved	Reserved	R	0
9	FNoPSldB	=0, 表示基波 B 路有功和视在不处于潜动状态; =1, 表示基波 B 路有功和视在处于潜动状态。	R	-
8	FNoPSldA	=0, 表示基波 A 路有功和视在不处于潜动状态; =1, 表示基波 A 路有功和视在处于潜动状态。	R	-
7	Reserved	Reserved	R	-
6	Reserved	Reserved	R	0
5	Reserved	Reserved	R	0
4	Reserved	Reserved	R	0
3	Reserved	Reserved	R	0
2	Reserved	Reserved	R	0
1	NoPSldB	=0, 表示 B 路有功和视在不处于潜动状态; =1, 表示 B 路有功和视在处于潜动状态。	R	-
0	NoPSldA	=0, 表示 A 路有功和视在不处于潜动状态; =1, 表示 A 路有功和视在处于潜动状态。	R	-

4.5.29 相电压电流状态寄存器 PHASES(0x1A4)

偏移地址: 1A4H;

PHASES 寄存器表示相电压失压、相序错、电压暂降、过压的状态以及相电流过流状态。该寄存器与寄存器或寄存器位 [LostVoltage](#)、[MODSEL](#) (三相四线和三相三线电压相序错判据不同)、[SAGCFG](#)、[OILVL](#)、[SAGIE](#)、[SAGIF](#)、[OIIE](#)、[OIIF](#) 配合使用。

位	位名称	功能描述	读写标志	复位值
31:15	Reserved	保留	R	0
14	Reserved	保留	R	-
13	OIIB	=0, 表示 IB 通道未发生了过流。=1 表示 IB 通道发生过流。 当 OVIIF 写 1 清零时, 同时清该位。	R	-
12	OIIA	=0, 表示 IA 通道未发生了过流。=1 表示 IA 通道发生过流。	R	-

		当 OVIF 写 1 清零时，同时清该位。		
11	Reserved	保留	R	0
10	Reserved	保留	R	0
9	Reserved	保留	R	0
8	OVUA	=0，表示 UA 通道未发生过压。=1 表示 UA 通道发生过压。 当 OVIF 写 1 清零时，同时清该位。	R	0
7	Reserved	保留	R	0
6	Reserved	保留	R	-
5	Reserved	保留	R	-
4	SAGUA	=0，表示 UA 通道未发生电压暂降。=1，表示 UA 通道发生了电压暂降。 当 SAGIF 写 1 清零时，同时清该位。	R	-
3	Reserved	保留	R	-
2	Reserved	保留	R	-
1	Reserved	保留	R	-
0	Reserved	保留	R	-

4.5.30 校验和寄存器 EMU_CKM (0x1A8)

地址：1A8H；字长：4 字节

该寄存器不受 MMU 写使能寄存器控制。

位	位名称	功能描述	读写标志	复位值
31:27	---	保留		
25	CKSUM_DONE	校验和计算完成，缺省值 0。 0，表示没有计算校验和或者计算校验和尚未结束； 1，表示计算校验和完成。 { CKSUM_DONE,CKSUM_START}: 2'b00, 上电缺省状态，尚未进行过校验和计算； 2'b01, 正在计算校验和； 2'b10, 校验和计算结束，校验和有效； 2'b11, 无效	R	0
24	CKSUM_START	校验和计算开始寄存器，缺省值 0。 0，表示不在计算中； 1、表示正在计算中； 写 1 开始计算，计算期间读该寄存器值为 1，计算结束该寄存器自动清零。若该寄存器为 1 期间写入 1 不影响正在进行的计算。 写入 0，若正在计算期间，则停止正在进行的计算，而且 CKSUM、	R/W	0

		DONE 都会被清零；若不在计算期间，则将 CKSUM、DONE 清零；		
23:0	CKSUM	读：计量模块配置寄存器的校验和结果寄存器，校验和是所有被校验寄存器的低三个字节求和并取反。 写：该 24 位也作为“校验和计算开始寄存器”的写密码； 软件写 CKSUM_START 为 1 时,必须保证写入的数据低 24bits 为 0x33_0F_A5。 软件写 CKSUM_START 为 0 时,必须保证写入的数据低 24bits 为 0xCC_0F_A5。	R/W	0x0

CKSUM 是计量模块配置寄存器的校验和寄存器（不包含 IE 寄存器），校验和计算地址范围分 5 段，其中：

CKSUM1 计算范围：0x000~0x07C

CKSUM2 计算范围：0x080~0x0F0、0x10C、0x110、0x114

CKSUM3 计算范围：0x0F4~0x108、0x118~0x18C，去掉 0x190

CKSUM4 计算范围：0x1B4~0x1B8、0x1C0~0x1C8

CKSUM5 计算范围：0x240~0x27C、0x2A0~0x2AC、0x2B4、（去掉 0x2B8）、0x600~0x664

每段校验和算法为三字节求和，若寄存器长度不足三字节，以零补齐

CKSUM 寄存器为 CKSUM1~5 三字节求和取反，即：

$CKSUM = \sim (CKSUM1 + CKSUM2 + CKSUM3 + CKSUM4 + CKSUM5)$

CKSUM 缺省态是无效的，只有当软件写 CKSUM_START 为 1 后，并等待 CKSUM_DONE 变为 1 后，CKSUM 值有效。从软件写 CKSUM_START 为 1 到计算结束，CKSUM_DONE 变为 1，需要对 173 个寄存器进行校验和计算，用时约 90us 时间，在计算期间，除了 CKSUM 寄存器之外，其他的 MMU 寄存器不可访问（写无效，读不到正确值）。

4.5.31 读校验寄存器 Rdata(0x1AC)

偏移地址：1ACH；有效字长：4 字节。

RData 寄存器保存前次读出的数据，可用于读出数据时的校验。当前次读出的寄存器字长不足四字节时，低位字节对齐。

4.5.32 写校验寄存器 WData(0x1B0)

偏移地址：1B0H；有效字长：3 字节。

WData 寄存器保存前次写入的数据，可用于写入数据时的校验。当前次写入的寄存器字长不足三字节时，低位字节对齐。

4.5.33 电压过零阈值寄存器 ZXOTU(0x1B4)

偏移地址：1B4H；有效字长：2 字节。

ZXOTU 为电压通道过零、相角计算、测频阈值。当 ZXOTCFG[4]=1 (0x92) 时可选择 ZXOTU 作为电压过零、相角计算、测频阈值。

当某通道的有效值小于过零阈值寄存器值时，该通道不输出过零，该通道电压不参与相角计算，相角输出 0x0，同时该相电压不参与测频计算。

过零阈值寄存器 ZXOTU 为 16 位无符号数，做比较时，将其与各相电压有效值 U_x ($x=A, B, C$ 下同) bit27-bit12 位进行比较，以作判断。ZXOTU 的复位初值为 0x0，用户可根据需要重新设定。

ZXOTU 计算公式：

$$ZXOTU = INT((REGUn * Ku)/2^{12})$$

其中 REGUn 为经校正后的额定电压有效值寄存器值，Ku=某相电压过零阈值和额定电压 Un 的比值，如过零阈值为 10%额定电压，则 K=0.1。

注：RN2025C64 应用时，必须将该寄存器配置为 0。

4.5.34 直流 OFFSET 自动校正使能寄存器 AUTODC_EN(0x1B8)

偏移地址：1B8H；默认值：0x00000080。

AUTODC_EN 寄存器用于启动 7 路 ADC 直流 OFFSET 自动计算并反映完成状态。

位	位名称	功能描述	读写标志	复位值
31:8	Reserved	保留。	R	0
7	MMU_DC_DIS	=0, DC offset 自动校正结果即 DCOS 寄存器影响计量通道参数	R/W	1
		=1, DC offset 自动校正结果即 DCOS 寄存器不影响计量通道参数计算		
		默认为 1。		
6	Reserved	保留位，无实际意义。	R	0
5	Reserved	保留位，无实际意义。	R	0
4	Reserved	保留位，无实际意义。	R	0
3	Reserved	保留位，无实际意义。	R	0
2	IB_DC_EN	写 1 启动 IB 自动 offset 校正，此时输入应该接地或空载；128 个周波后校正自动完成，校正完成后该位自动清零，并将校正结果写入 DCOS_IB 寄存器。	R/W	0
1	UA_DC_EN	写 1 启动 UA 自动 offset 校正，此时输入应该空载；128 个周波后校正自动完成，校正完成后该位自动清零，并将校正结果写入 DCOS_UA 寄存器。	R/W	0
0	IA_DC_EN	写 1 启动 IA 自动 offset 校正，此时输入应该接地或空载；128 个周波后校正自动完成，校正完成后该位自动清零，并将校正结果写入 DCOS_IA 寄存器。	R/W	0

注意

1. 使用 AUTODC 做 DC OFFSET 校正时，需先将通道增益校平。
2. 若用于 EMU 模式校正，自动校正完成后，得到的 DCOSx 寄存器值可直接使用。
3. 若用于全失压电流有效值校正，通过 IA_DC_EN、IB_DC_EN、IC_DC_EN 启动自动校正，得到的 DCOSx 寄存器值除 0.512 后填入全失压模块的 LS_DCOS_Ix 寄存器，完成全失压有效值直流 offset 校正。

4.5.35 过零计算配置及标志寄存器 ZXOTCFG(0x1BC)

偏移地址：1BC H；注意：该寄存器不参与校验和计算。

位	位名称	功能描述	读写标志	复位值
31:12	Reserved	保留。	R	0
11:05	ZXFLAG	只读位，存放每个通道的阈值比较标志，为 1 表示该通道大于过零计算阈值；为 0 表示小于阈值，顺序为{x/x/IB/IA/x/x/UA}。其中 x 表示保留位。	R	-
4	ZXOTUS	=1 时，电压过零及测频计算阈值选择为 ZXOTU，=0 时，电压过零及测频计算阈值选择为 ZXOT；	R/W	0
3	Reserved	保留位，可读可写，无实际意义。	R/W	0
2	Reserved	保留位，可读可写，无实际意义。	R/W	0
1	Reserved	保留位，可读可写，无实际意义。	R/W	0
0	ZXOTUA	=1 时，UA 通道一直输出过零信号；=0 时，以阈值比较结果为准选择 UA 是否输出过零信号；	R/W	0

4.5.36 自定义功率/HFCONST3/D2F/能量寄存器

4.5.36.1 自定义功率积分时间设置寄存器 D2FTIME (0x1C0)

偏移地址：0x1C0。默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:14	Reserved	保留。	R	0
13	D2F_START	硬件计算积分时间的启动信号，当该寄存器写入 1 时，硬件开始周期性的计时，计时时间到 D2F_Period 设定的时间时，功率值会被自动装载到积分器去计算电能。如果要停止计数，或者需要重新启动计数，需要先写 0 再写 1。 建议软件与 DMA 波形缓存基本同时启动。这样硬件积分的周期就与 DMA 波形缓存整周波更新周期一致。	R/W	0
12	D2F_MODE	=0：积分开始时间为软件写入功率的时间，在做谐波计量的应用下，软件应每周波写入谐波功率值，但是由于软件延时的不确定性，不能保证积分时间刚好是一个周波； =1：积分开始时间由硬件自动决定，软件只需要保证每周波写入功率值即可，此功率值会被预存到 buffer 中，当硬件的周波间隔信号来时才真正装载入积分器。此方案可以保证每周波的积分时间是准确的。	R/W	0
11:0	D2F_Period	12 位无符号数，用于定义功率积分的时间，最小刻度代表采样时间间隔，采样时间间隔由 DMA_WAVECON(0x2B4)寄存器定义。 比如，当 WAVE_SR 选择为 2'b10 时，代表 256 点同步采样，一个周波就是 256 点采样数据。如果功率计算以一个周波为单位计算，那么此寄存器就应该写入 256-1=255，十六进制为 12'h0FF。	R/W	0

		此配置代表了功率积分的单位周期，正常情况下典型值为20ms。		
--	--	--------------------------------	--	--

4.5.36.2 自定义电能脉冲寄存器 D2FCFG(0x1C4)

偏移地址：0x1C4。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	HFCONST3_CFG	=0: HFCONST3 必须设置为 4 的整数倍，同等条件下，与 HFCONST1 的关系是 HFCONST3=4*HFCONST1； =1: HFCONST3 不必设置为 4 的整数倍，同等条件下，与 HFCONST1 的关系是 HFCONST3=HFCONST1；	R/W	0
13:12	CF4_UDef	=00, CF4 输出参照 CFCFG (180H) 寄存器定义； =01, CF4 输出自定义脉冲 CF5； =10, CF4 输出自定义脉冲 CF6； =11, CF4 输出自定义脉冲 CF7；	R/W	0
11:10	CF3_UDef	=00, CF3 输出参照 CFCFG (180H) 寄存器定义； =01, CF3 输出自定义脉冲 CF5； =10, CF3 输出自定义脉冲 CF6； =11, CF3 输出自定义脉冲 CF7；	R/W	0
9:8	CF2_UDef	=00, CF2 输出参照 CFCFG (180H) 寄存器定义； =01, CF2 输出自定义脉冲 CF5； =10, CF2 输出自定义脉冲 CF6； =11, CF2 输出自定义脉冲 CF7；	R/W	0
7:6	CF1_UDef	=00, CF1 输出参照 CFCFG (180H) 寄存器定义； =01, CF1 输出自定义脉冲 CF5； =10, CF1 输出自定义脉冲 CF6； =11, CF1 输出自定义脉冲 CF7；	R/W	0
5:4	CF0_UDef	=00, CF0 输出参照 CFCFG (180H) 寄存器定义； =01, CF0 输出自定义脉冲 CF5； =10, CF0 输出自定义脉冲 CF6； =11, CF0 输出自定义脉冲 CF7；	R/W	0
3:0	D2FPREG_CFG	≠4'b1001 时，D2FP 寄存器 (0x1CC~1F8) 作为功率通道相位分段校正寄存器； =4'b1001 时，D2FP 寄存器 (0x1CC~1F8) 作为自定义功率寄存器；	R/W	0

4.5.36.3 自定义 HFCONST3 寄存器(0x1C8)

偏移地址：0x1C8。默认值：0x0

当 D2fcfg[3:0]等于 4'b1001 时，HFCONST3 用于 12 个自定义电能寄存器 D2FE00-D2FE11 及 3 个自定义 CF 脉冲 CF5-CF7 的脉冲常数；否则不起作用。

高频脉冲常数寄存器是 16 位无符号数。同等条件下，依据 HFCONST1 的计算公式得到的值乘以 4 后再作为 HFCONST3 的值，即： $HFCONST3 = 4 * HFCONST1$ 。HFCONST1 计算公式参见 4.5.9 章节。升级版芯片可通过设置 D2FCFG(0x1C4)，使 HFconst3 与 HFconst1 不必有 4 倍的关系。

如果 HFCONST3 和 HFCONST1 定义的脉冲常数相同，则在 $HFCONST3=4*HFCONST1$ 的情况下（或者通过设置，满足 $HFCONST3=HFCONST1$ 即可），取 32 位功率寄存器 PT 的高 24 位填入自定义功率寄存器 D2FP，自定义电能脉冲的输出与 HFCONST1 决定的脉冲输出速度一致。

如果定义的脉冲常数不同，有两种调整方式，分别是调整 HFCONST3 和自定义功率寄存器填入值。举例说明，HFCONST1 定义的电表脉冲常数为 1200，HFCONST3 定义的电表脉冲常数为 24000，调整方式分别如下：

方式 1，自定义功率寄存器 $D2FP=PT/2^8$ ，相应调整 $HFCONST3=HFCONST1*4/20=HFCONST1*0.2$ ；

注意：如果 HFCONST3 根据上述公式不能得到整数，可在校表步骤调整 HFCONST1 的值，使其为 5 的整数倍。

方式 2， $HFCONST3=HFCONST1*4$ ，相应的调整填入自定义功率寄存器填入的值，调整后为：

$D2FP=PT/2^8 * (24000/1200)$

4.5.36.4 自定义功率寄存器(0x1CC~0x1F8)

偏移地址	1CCH	1D0H	1D4H	1D8H	1DCH	1E0H	1E4H
寄存器	D2FP00	D2FP01	D2FP02	D2FP03	D2FP04	D2FP05	D2FP06
默认值	0x0	0x0	0x0	0x0	0x0	0x0	0x0
偏移地址	1E8H	1ECH	1F0H	1F4H	1F8H		
寄存器	D2FP07	D2FP08	D2FP09	D2FP10	D2FP11		
默认值	0x0	0x0	0x0	0x0	0x0		

自定义功率寄存器 D2FP00-11 可采用二进制补码格式，24 位有符号数，其中最高位是符号位。

当 D2FCFG[3:0]等于 4'b1001 时，该寄存器作为 12 个自定义功率寄存器，当往该寄存器写入值时，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 可以输出脉冲，对应脉冲分别 CF5、CF6、CF7，可以依据 D2fcfg[13:4]配置从 CF0~CF4 管脚输出。

4.5.36.5 自定义能量寄存器(0x1FC~0x228)

偏移地址	1FCH	200H	204H	208H	20CH	210H	214H
寄存器	D2FE00	D2FE01	D2FE02	D2FE03	D2FE04	D2FE05	D2FE06
默认值	0x0	0x0	0x0	0x0	0x0	0x0	0x0
偏移地址	218H	21CH	220H	224H	228H		
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；当 D2fcfg[3:0]等于 4'b1001 时，该寄存器作为 12 个自定义电能寄存器中的，与自定义功率寄存器一一对应。其中 D2FE00、D2FE01、D2FE02 可以输出脉冲 CF5、CF6、CF7；可以选择为累加型或者读后清零型，受 EMUCFG.ERegCAR 控制。

当 D2fcfg[3:0]不等于 4'b1001 时，这些寄存器没有意义。

4.5.36.6 积分电路标志寄存器 D2F_FLAG(0x230)

位	位名称	功能描述	读写标志	复位值
31:17	Reserved	保留。	R	0
16	D2F_LOAD	当 D2FTIME (0x1C0)寄存器中的 D2F_MODE=1 时, 使用硬件固定时间积分, 每当功率值从 Buffer 中装入积分器, 该位置 1, 软件写 1 清零。典型应用下, 该位置 1 的周期为 20ms。	R/W	0
15:10	Reserved	保留。	R	0
9:0	SR_TIME	10 位无符号数, 只读寄存器, 代表了当前同步波形采样通道的采样时间间隔。 举例: 0x9F 代表当前为 $2.048\text{Mhz}/(\text{SR_TIME}+1)=2.048\text{Mhz}/160=12.8\text{Khz}$ 。	R	0x9F

4.5.37 短路停电阈值寄存器

4.5.37.1 过流幅值阈值寄存器 OI2_LVL(0x260)

偏移地址: 0x260; 有效字长: 2 字节; 默认值: 0x0

该寄存器用于判断短路停电事件的过流幅值阈值配置寄存器, 16 位无符号数, 写保护, 有效位[15:0]。

计算公式: $OI2LVL = INT((\sqrt{2} * REGI_b * K/2^{12}))$

其中: REGI_b 为校正后标称电流 I_b 有效值寄存器值, K 为短路过流系数, 如 K=20。

注意, 实际应用中, 短路过流幅值阈值若超过 ADC 满量程, 可将阈值设置在 ADC 满量程附近。

该寄存器为 0 时, 关闭短路过流检测功能。该寄存器不为 0 且 OI2_CNTTH 不为 0 时, 启动短路过流检测功能。默认关闭。

某相电流峰值绝对值大于短路过流幅值阈值寄存器 OI2_LVL[15:0], 且持续时间大于等于短路过流时间阈值寄存器 OI2_CNTTH [5:0]*(1/32) ms, 判定该相满足短路停电事件的过流条件。

某相过流将导致 EMUIF2. OIx2_IF (x=A,B,C)标志置位, 若 EMUIE2. OIx2_IE=1, 则产生该相的短路过流中断。

注: RN2025C64 不支持该功能。

4.5.37.2 过流时间阈值寄存器 OI2_CNTTH(0x264)

偏移地址: 0x264; 有效字长: 1 字节; 默认值: 0x0

该寄存器用于判断短路停电事件的过流时间阈值配置寄存器, 写保护, 有效位[5:0], 以 1/32ms 为单位。

做过流时间判断时, 该寄存器与内部的 OIx_CNT 计数器进行比较。当某相电流瞬时值绝对值高 16 位大于电流阈值寄存器 OI2_LVL[15:0], 内部 OIx_CNT 启动, 以 16Khz 速率增 1 计数。累计计数 OIx2_CNT 计数值大于预设的 OI2_CNTTH[5:0]寄存器值时, 即判断为该相电流短路过流, 相应的 OI2x_IF 中断标志位置位。

该寄存器为 0 时, 关闭短路过流检测功能。该寄存器不为 0 且 OI2_LVL 不为 0 时, 启动短路过流检测功能。默认关闭。

注: RN2025C64 不支持该功能。

4.5.37.3 过流归零联合事件时间阈值寄存器 OIRZ_CNTTH(0x274)

偏移地址: 0x274; 有效字长: 2 字节; 默认值: 0x0

该寄存器配置电流过流事件和电流联合归零事件均发生的窗口等待时间 T

$$T = \text{OIRZ_CNTTH}[15:0] * 256 * 0.125 \text{ ms}$$

当值为 0 时不使能该功能；写入不为 0 的值使能该功能，电流过流后启动电流归零检测，当窗口内检测结果有中断报出。

注：RN2025C64 不支持该功能。

4.5.38 过压和电压峰值寄存器

4.5.38.1 过压幅值阈值寄存器 OV2_LVL(0x278)

偏移地址：0x278；有效字长：2 字节；默认值：0x0

某相电压半波峰值 $U_x\text{PEAK_HW}[15:0]$ 大于过压幅值阈值寄存器 $\text{OV2_LVL}[15:0]$ ，且过压持续时间大于等于过压时间阈值寄存器 $\text{OV2_CNTTH}[15:0]$ 个半周波数，判定该相电压过压。

电压过压幅值阈值配置寄存器，16 位无符号数，有效位[15:0]，计算公式：

$$\text{OVLVL} = \text{INT}((\sqrt{2} * \text{REGV}_n * K / 2^{12}))$$

其中 REGV_n 为校正后相电压标称电压 U_n 有效值寄存器值，K 为过压系数，如 $K=1.2$ 。

该寄存器为 0 时，关闭过压检测功能。该寄存器不为 0 且 OV2_CNTTH 不为 0 时，启动过压检测功能。默认关闭。

某相电压过压将导致 $\text{EMUIF2.OV}_x_IF(x=A,B,C)$ 标志置位，若 $\text{EMUIE2.OV}_x_IE=1$ ，则产生过压中断。

注：RN2025C64 不支持该功能。

4.5.38.2 过压时间阈值寄存器 OV2_CNTTH(0x27C)

偏移地址：0x27C；有效字长：2 字节；默认值：0x0

电压过压时间阈值配置寄存器，有效位[15:0]，以半周波为单位。使用该寄存器前需配置 50hz/60hz 选择 EMUCFG.FreqSEL 寄存器位。

该寄存器为 0 时，关闭过压检测功能。该寄存器不为 0 且 OV2_LVL 不为 0 时，启动过压检测功能。默认关闭。

做过压判断时，该寄存器与内部的半波计数器进行比较。

注：RN2025C64 不支持该功能。

4.5.38.3 电压半波峰值寄存器(0x58C)

电压半波峰值寄存器为某相电压波形 8K 采样值半波峰值绝对值的高 16 位。有效位为[15:0]，为 16 位无符号数。

内部做过压判定时，该寄存器与 $\text{OV2_LVL}[15:0]$ 阈值进行比较。

相关寄存器：电压通道增益寄存器 GSU_x 、50hz/60hz 选择 EMUCON.FreqSEL 。

4.5.39 计量单元配置寄存器 2 EMU_CFG2(0x270) (B0 修改)

计量单元配置寄存器 EMU_CFG2 ，用于配置半波有效值计算过零点

偏移地址：0x270；默认值：0x04

位	位名称	功能描述	读写标志	复位值
31:4	Reserved	保留	R	0
3	P_HW_SEL	=0，有功电能计算采用默认功率计算方法	R/W	0

		=1, 有功电能计算采用半波更新的功率值		
2	RMS_HW_CFG	=0, 半波有效值选择基波电压过零点计算 =1, 半波有效值选择全波电压过零点计算 注: 建议配置为 0, 选择基波过零。	R/W	1
1	P_HW_CFG	=0, 半波有功功率选择基波电压过零点计算 =1, 半波有功功率选择全波电压过零点计算 注: 建议配置为 0, 选择基波过零。	R/W	0
0	RMS_HW_SQRT	=1: 32 位半波有效值寄存器表示的是开根号的结果, 低 24 位有效, 高 8 位固定为 0. =0: 32 位半波有效值寄存器表示的是原始波形平方后的结果, 32 位有效, 还需要软件开根号才能得到半波有效值。	R/W	0

4.5.40 EMU 中断 2 寄存器

4.5.40.1 EMUIE2 寄存器(0x2B8)

偏移地址: 0x2B8; 默认值: 0x0

该寄存器为EMU中断2使能寄存器。在相应中断允许位配置为1且IF清零, 当该事件发生时, 相应EMUIF2位置位并产生EMU中断2 (中断号4, 中断地址0x50)。

位	位名称	功能描述	读写标志	复位值
31:24	Reserved	保留	R	0
23	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
22	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
21	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
20	DMA_BUFUA_ErrIE	波形缓存 DMA UA 通道数据错误中断使能位。	R/W	0
19	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
18	DMA_BUFIB_ErrIE	波形缓存 DMA IB 通道数据错误中断使能位。	R/W	0
17	DMA_BUFIAErr_IE	波形缓存 DMA IA 通道数据错误中断使能位。	R/W	0
16	DMA_BUFHF_IE	波形缓存 DMA 通道 BUFFER 半满中断使能位。	R/W	0
15	DMA_BUFF_IE	波形缓存 DMA 通道 BUFFER 满中断使能位。	R/W	0
14	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
13	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
12	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
11	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
10	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
9	OVA_IE	电压过压中断使能。	R/W	0
8	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
7	OIBRZ_IE	B 路电流过流归零联合中断使能。	R/W	0
6	OIARZ_IE	A 路电流过流归零联合中断使能。	R/W	0
5	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
4	Reserved	保留位, 可读可写, 无实际意义。	R/W	0
3	Reserved	保留位, 可读可写, 无实际意义。	R/W	0

2	Reserved	保留位，可读可写，无实际意义。	R/W	0
1	OIB2_IE	B 路电流短路过流中断使能。	R/W	0
0	OIA2_IE	A 路电流短路过流中断使能。	R/W	0

备注：EMU中断2中的电压过压及电流过流中断与EMU中断1中的过压/过流中断可以独立设置和产生。

4.5.40.2 EMUIF2 寄存器(0x2BC)

偏移地址：2BCH；**该寄存器不受写保护**

该寄存器为EMU中断2状态寄存器。当EMU中断2某事件发生时，相应IF位置1。当对应中断允许位配置为1，状态位置1会产生EMU 中断2(中断号4，中断地址0x50)。

位	位名称	功能描述	读写标志	复位值
31:28	Reserved	保留	R	0
27	Reserved	保留	R	0
26	Reserved	保留	R	0
25	Reserved	保留	R	0
24	HW_ZXUA	UA 通道半波有效值更新标志，写 1 清零	R/W	1
23	Reserved	保留	R	0
22	Reserved	保留	R	0
21	Reserved	保留	R	0
20	DMA_BUFUA_ErrIF	波形缓存 DMA UA 通道数据错误输出标志。	R/W	0
19	Reserved	保留	R	0
18	DMA_BUFIB_ErrIF	波形缓存 DMA IB 通道数据错误输出标志。	R/W	0
17	DMA_BUFIAErr_IF	波形缓存 DMA IA 通道数据错误输出标志。	R/W	0
16	DMA_BUFHF_IF	波形缓存 DMA 通道 BUFFER 半满输出标志。	R/W	0
15	DMA_BUFF_IF	波形缓存 DMA 通道 BUFFER 满中断输出标志。	R/W	0
14-12	Reserved	保留	R	0
11	Reserved	保留	R	0
10	Reserved	保留	R	0
9	OVA_IF	A 路电压过压输出标志。 若 OVA_IE=1，OVA_IF 置位产生 A 路电压过压中断。	R/W	-
		OVA_IF 写 1 清零同时清中断。		
8	Reserved	保留	R	0
7	OIBRZ_IF	B 路电流过流归零联合事件标志。 若 OIBRZ_IE=1，OIBRZ_IF 置位产生电流过流归零联合中断。OIBRZ_IF 写 1 清零同时清中断。	R/W	-
6		OIARZ_IF		
5-3	Reserved	保留	R	0
2	Reserved	保留	R	0
1	OIB2_IF	B 路电流过流标志。	R/W	-

		若 OIB2_IE=1, OIB2_IF 置位产生 B 路电流过流中断。		
		OIB2_IF 写 1 清零同时清中断。		
0	OIA2_IF	A 路电流过流标志。	R/W	-
		若 OIA2_IE=1, OIA2_IF 置位产生 A 路电流过流中断。		
		OIA2_IF 写 1 清零同时清中断。		

备注：EMU中断2中的电压过压及电流过流中断与EMU中断1中的过压/过流中断可以独立设置和产生。

4.5.41 DMA 波形缓存配置寄存器

4.5.41.1 DMA_BUF_CTRL(0x2A0)

偏移地址：2A0H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31: 2	Reserved	保留。	R	0
1	WMCFG	DMA 单次模式、连续模式选择 =0, 单次模式, 写操作 Buf 满之后则不再发起 DMA 操作; =1, 连续模式, 写操作 Buf 满, 折返到起始地址重新开始。	R/W	0
0	BUF_EN	数据缓存使能位; =0 为不使能, =1 为使能;	R/W	0

4.5.41.2 DMA_BUF_BASE_ADDR(0x2A4)

偏移地址：2A4 H；默认值：0x00000C00

位	位名称	功能描述	读写标志	复位值
31:14	Reserved	保留。	R	0
13:0	DMA_BUF_BASE_ADDR	数据缓存的目标地址的偏移地址。数据缓存目标地址=该寄存器值*4 + ram 基址, 若该寄存器值设置为 0xc00, ram 基址为 0x10000000 则数据缓存目标地址 =0xc00*4 + 0x10000000 = 0x10003000。 模式 0: 注意 1: 该寄存器值要与数据缓存深度匹配, 即其值与深度的关系式为 $DMA_BUF_BASE_ADDR = (DMA_BUF_DEPTH + 1) * N$ 。 如若数据缓存深度为 128W ($DMA_BUF_DEPTH = 127$), 那么 $DMA_BUF_BASE_ADDR = (DMA_BUF_DEPTH + 1) * N = 128 * N$; 如若数据缓存深度为 2KW ($DMA_BUF_DEPTH = 2047$), 那么 $DMA_BUF_BASE_ADDR = (DMA_BUF_DEPTH + 1) * N = 2048 * N$;	R/W	0xC00

		具体匹配规则可参见下表： 注意 2: 该寄存器的 bit6 至 bit0 写 1 无效，芯片固定将这 7 位按 0 处理。 模式 1 和模式 2: 可支持任意地址配置。		
--	--	---	--	--

寄存器 DMA_BUF_DEPTH[10:7]	缓存深度配置范围 (Words)	DMA_BUF_BASE_ADDR 寄存器	ADC 缓存基址的物理地址	
4'b0000	128	[13:7]配置有效, [6:0]配置无效	{DMA_BUF_BASE_ADDR[13:7], 7'b00000000}	SRAM 划分为 64 段 128 Words, 每一段的起始地址
4'b0001	129-256	[13:8]配置有效, [7:0]配置无效	{DMA_BUF_BASE_ADDR[13:8], 8'b00000000}	SRAM 划分为 32 段 256 Words, 每一段的起始地址
4'b0010	257-384	[13:9]配置有效, [8:0]配置无效	{DMA_BUF_BASE_ADDR[13:9], 9'b00000000}	SRAM 划分为 16 段 512 Words, 每一段的起始地址
4'b0011	385-512			
4'b0100	513-640	[13:10]配置有效, [9:0]配置无效	{DMA_BUF_BASE_ADDR[13:10], 10'b0000000000}	SRAM 划分为 8 段 1K Words, 每一段的起始地址
4'b0101	641-768			
4'b0110	769-896			
4'b0111	897-1024			
4'b1000	1025-1152	[13:11]配置有效, [10:0]配置无效	{DMA_BUF_BASE_ADDR[13:11], 11'b000000000000}	SRAM 划分为 4 段 2KWords, 每一段的起始地址
4'b1001	1153-1280			
4'b1010	1281-1408			
4'b1011	1409-1536			
4'b1100	1537-1664			
4'b1101	1665-1792			
4'b1110	1793-1920			
4'b1111	1921-2048			

4.5.41.3 DMA_BUF_DEPTH(0x2A8)

偏移地址: 0x 2A8; 默认值: 0x7F

位	位名称	功能描述	读写标志	复位值
31:11	Reserved	保留	R	0
13:11	DMA_BUF_DEPTH	Buffer 深度配置高位 (新增), 寄存器从 11 位扩展为 14 位。	R/W	0x0

10:0	DMA_BUF_DEPTH	<p>模式 0：波形缓存的总深度 波形缓存总深度设置，深度为 (N+1) (Word)； 最小配置为 128 Words (默认 0x7F)，最大为 16K Words 深度设置：通道数*每个通道点数-1；如 5 个通道，每个通道存 128 个数， DMA_Buf_depth=5*128-1</p> <p>模式 1：意义更改为单通道的深度 该寄存器配置单通道深度，深度为 (N+1) (Word)； 支持任意深度配置。 波形缓存的总深度由通道深度和通道间隙深度计算而来，见 4.3.2 章节说明。</p> <p>模式 2：该寄存器定义一个周波区块的深度 该寄存器配置一个周波区块深度，深度为(N+1)(Word)； 支持任意深度配置。</p>	R/W	0x7F
------	---------------	--	-----	------

4.5.41.4 DMA_WAVE_CH_SEL(0x2AC 修改)

偏移地址：0x2AC；默认值：0x0D63447F

位	位名称	功能描述	读写标志	复位值
31:28	Reserved	Reserved		
27:25	CH6_SEL	第 7 个通道存放数据选择(对应通道选择见 CH0_SEL)	R/W	6
24:22	CH5_SEL	第 6 个通道存放数据选择(对应通道选择见 CH0_SEL)。	R/W	5
21:19	CH4_SEL	第 5 个通道存放数据选择(对应通道选择见 CH0_SEL)	R/W	4
18:16	CH3_SEL	第 4 个通道存放数据选择(对应通道选择见 CH0_SEL)	R/W	3
15:13	CH2_SEL	第 3 个通道存放数据选择(对应通道选择见 CH0_SEL)	R/W	2
12:10	CH1_SEL	第 2 个通道存放数据选择(对应通道选择见 CH0_SEL)	R/W	1
9:7	CH0_SEL	第 1 个通道存放数据选择： 0 = IA 1 = IB 2 = 保留 3 = UA 4 = 保留 5 = 保留 6 = 保留 7 = 关闭 CHn_SEL 用来选择波形存放的顺序，在模式 1、模式 2 下有效。默认是 IA/IB/-/UA/-/-，通过配置可以组成任意数据，比如 IA/UA/IB/-/-	R/W	0
6	DMA_WAVE_IA_SEL	IA 通道是否参与 DMA ADC 波形缓存	R/W	1

		=0, 不选; =1, 选择;		
5	DMA_WAVE_IB_SEL	IB 通道是否参与 DMA ADC 波形缓存 =0, 不选; =1, 选择;	R/W	1
4	DMA_WAVE_IC_SEL	注意: 软件应该配置为 0	R/W	1
3	DMA_WAVE_UA_SEL	UA 通道是否参与 DMA ADC 波形缓存 =0, 不选; =1, 选择;	R/W	1
2	DMA_WAVE_UB_SEL	注意: 软件应该配置为 0	R/W	1
1	DMA_WAVE_UC_SEL	注意: 软件应该配置为 0	R/W	1
0	DMA_WAVE_IN_SEL	注意: 软件应该配置为 0	R/W	1

该寄存器在 RN2025 的定义与 RN2026 一致, RN2026 有 UA/UB/UC/IA/IB/IC/IN 一共 7 个通道, RN2025 只有 UA/IA/IB 三个通道。

ADC 数据每次采样在数据缓存中的通道排序由 DMA_WAVE_CH_SEL[6:0] 寄存器使能的通道决定, 按 IA/IB/IC/UA/UB/UC/IN 顺序排序, 若某通道不使能, 则该通道忽略, 如 DMA_WAVE_CH_SEL[6:0]=0x5A, IB/UB/IN 不使能, 则每次采样在 SRAM 数据缓存中的排序为 IA/IC/UA/UC。

建议参照锐能微提供的例程对该寄存器进行配置。

4.5.41.5 DMA_WAVE_ERR_ADDR(0x2B0)

偏移地址: 0x2B0; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:14	Reserved	保留	R	0
13:0	DMA_WAVE_ERR_ADDR	在 ADC 采样间隔内, 若有任何一个通道 DMA 操作没有响应, 则, 记录数据错误发生地址并发出 DMA 错误中断; 记录数据错误发生地址放置在 DMA_WAVE_ERR_ADDR [13:0] (0x2B0) 寄存器中	R	0

4.5.41.6 DMA_WAVECON(0x2B4)

偏移地址: 0x2B4; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:18	Reserved	保留	R	0
17	DS_CFG1	=0: DMA 波形数据源头由 DS_CFG 寄存器决定; =1: 并且 DS_CFG=2'b01 时, DMA 波形数据源头为基波波形;	R/W	0
16	Wavecnt_mode	用于决定 wavecnt 的模式 =0: wavecnt 低 9 位有效, 公式为 WAVECNT=((1.024*10^6)/freq/N)-1	R/W	0

		=1: wavecnt 共 10 位有效, 公式为 $WAVECNT = ((2.048 * 10^6) / \text{freq} / N) - 1$		
15:14	Reserved	保留	R	0
13:4	WAVECNT	用于改变缓存波形的采样率: 当 wavecnt_mode=0 时: 若 WAVECNT[8:0]=0, 则 WAVE_SR 配置的采样率有效 若 WAVECNT[8:0]≠0, 则 WAVE_SR 配置的采样率无效, 若期望的周波采样点数为 N, 按如下公式配置 WAVECNT 公式为: $WAVECNT = ((1.024 * 10^6) / \text{freq} / N) - 1$; (freq 为电网频率, 一般为 50Hz 或 60Hz)。此时支持的最小写入值为 32. 当 wavecnt_mode=1 时: 若 WAVECNT[9:0]=0, 则 WAVE_SR 配置的采样率有效 若 WAVECNT[9:0]≠0, 则 WAVE_SR 配置的采样率无效, 若期望的周波采样点数为 N, 按如下公式配置 WAVECNT 公式为: $WAVECNT = (2.048 * 10^6) / \text{freq} / N - 1$; (freq 为电网频率, 一般为 50Hz 或 60Hz) 此时支持的最小写入值为 64。当写入的值小于 64 时, 预期值可写入, 但是实际起作用的一直是 63 (即: 采样率一直固定为 32Khz, 最大也只支持 32KHz)。	R/W	0
03:02	WAVE_SR	=2'b11 512 点同步采样; (16.384Mhz 和 32.768Mhz 主频下可用, 其他主频不可用) =2'b10 256 点同步采样; (适用 8.192M/16.384M/32.768Mhz 主频) =2'b01 128 点同步采样; (适用 8.192M/16.384M/32.768Mhz 主频) =2'b00 64 点同步采样; (适用 8.192M/16.384M/32.768Mhz 主频)	R/W	0
01:00	DS_CFG	=2'b1X: 选择同步频率跟踪的波形, 采样点数由 BIT[12: 2]决定; =2'b 01: 选择经过 EMU 通道增益和通道相位校正 [7:0], 高通滤波器前的数据; 固定 8k 采样, 该波形受直流 offset 校正的影响。 =2'b00: 选择经过 EMU 通道增益和通道相位校正、直流 offset 校正、高通滤波器后的数据; (默认不进行直流校正, 高通开启), 固定 8K 采样率	R/W	0

注意: 波形源选择波形缓存同步频率跟踪整周波采样功能, 即 WAVECON[1: 0]=1X 时, 全失压寄存器 (BaseAddr

: 0x40040000) 要做如下配置:

1. NVM_IE (0x00) =0, 关闭全失压有效值更新中断
2. 根据需求配置高通开关控制位LS_HPFOFF (0x08[1]) 或LS_DCOS(0x0C-14)是否校正 (LS_DCOS值的获取可参考4.5.34直流OFFSET自动校正章节说明)。
对于获取交流采样波形应用, LS_HPFOFF可配置为高通开关使能; LS_DCOS可配置为0, 即不校正。
3. LSMODE=1 (0x08[0]), 选择使能同步跟踪频率整周波采样模式
4. NVMEN (0x08[2]) =1, 使能NVM模块
5. 8K采样率下的采样瞬时值, 为wav1; 经过N点同步采样后, 波形采样瞬时值为wav2, 其中N为同步采样点数, N的取值为64/128/256/512/自定义;

二者关系为: $wav2 = wav1 * (8000/Rt)^3$, 其中Rt为采样率, $Rt = N * Freq$, N为采样点数, Freq为电网频率。

8K采样率下, 波形采样寄存器值与电压电流实际值的转换系数k1;

经过N点同步采样后, 瞬时采样峰值wav2与电压电流实际值的转换系数k2;

二者的关系: $k2 = k1 * (1 + Gain) * (Rt/8000)^3$; 其中Gain为相应通道增益的归一化值, 具体见“[4.5.14通道增益寄存器](#)”章节。

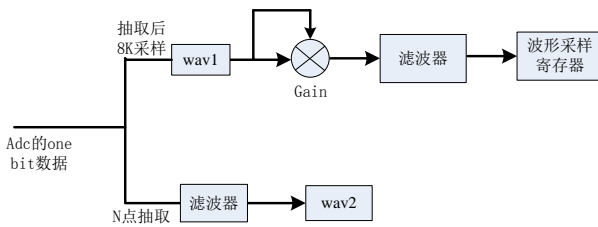


图4-6采样关系

注意：配置DMA_WAVECON2(0x600)中的CIC_AGC_EN=1后, 硬件会做自动的增益补偿, 同步采样通道得到的波形数据幅度与8KHz计量通道会保持基本一致。

4.5.41.7 DMAWAV_MODE_SEL(新增 0x2C0)

偏移地址：0x2C0；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	CKM_EN	该位只在模式 1 有效 校验和使能位 0 = 不计算校验和 1 = 波形缓存过程，按通道计算校验和，校验和结果存放寄存器	R/W	0
1:0	DMAMODE	DMA 模式选择： 00=模式 0，不同通道数据按点交叉存放（单个周波区块） 01=模式 1，不同通道数据按通道存放 10=模式 2，不同通道数据按点交叉存放（多个周波区块） 11=保留 关于模式更详细的定义参见 4.3.2 章节	R/W	0

4.5.41.8 DMAWAV_GAP_CH_SEL(新增 0x2C4)

偏移地址：0x 2C4；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:7	Reserved	保留	R	0
6	CH6_GAP_EN	第 7 个通道后面是否插入间隙，0=不插入，1=插入间隙	R/W	0
5	CH5_GAP_EN	第 6 个通道后面是否插入间隙，0=不插入，1=插入间隙	R/W	0
4	CH4_GAP_EN	第 5 个通道后面是否插入间隙，0=不插入，1=插入间隙	R/W	0
3	CH3_GAP_EN	第 4 个通道后面是否插入间隙，0=不插入，1=插入间隙	R/W	0
2	CH2_GAP_EN	第 3 个通道后面是否插入间隙，0=不插入，1=插入间隙	R/W	0
1	CH1_GAP_EN	第 2 个通道后面是否插入间隙，0=不插入，1=插入间隙	R/W	0
0	CH0_GAP_EN	第 1 个通道后面是否插入间隙，0=不插入，1=插入间隙	R/W	0

注意：该寄存器只在**模式 1**有效。

建议参照锐能微提供的例程对该寄存器进行配置。

4.5.41.9 DMAWAV_GAP_DEPTH(新增 0x2C8)

偏移地址：0x 2C8；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:11	Reserved	保留	R	0

10:0	DMA_GAP_DEPTH	<p>仅模式 1/2 有效:</p> <p>模式 1: 数据缓存两个通道间隙深度配置, 深度为 (N) (Word); 支持任意深度配置。 当配置为 0, 则所有通道后面不插入间隙空间。 注意, 所有通道有效数据区域后面插入的间隙大小相同。</p> <p>模式 2: 数据缓存两个周波区块间隙深度配置, 深度为 (N) (Word); 支持任意深度配置。 当配置为 0, 则所有周波区块后面不插入间隙空间。</p>	R/W	0
------	---------------	--	-----	---

建议参照锐能微提供的例程对该寄存器进行配置。

4.5.41.10 DMAWAV_BANK_CNT(新增 0x2CC)

偏移地址: 0x 2CC; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:11	Reserved	保留	R	0
3:0	DMA_BANK_CNT	<p>仅模式 1/2 有效:</p> <p><u>周波区块</u>个数, 数据存储会按照循环存储, 比如配置为 3, 数据会先存放 A 区, A 区存满存放 B 区, 然后存放 C 区, 然后又存放 A 区</p>	R/W	0

建议参照锐能微提供的例程对该寄存器进行配置。

4.5.41.11 DMAWAV_BUFF_ADDR(新增 0x 2D0)

偏移地址: 0x 2D0; 默认值: 0xC00

位	位名称	功能描述	读写标志	复位值
31:14	Reserved	保留	R	0
13:0	DMA_BUFF_ADDR	软件读该寄存器, 获取当前 DMA 指针地址	R	0xC00

4.5.41.12 DMAWAV_CHECKSUM0(新增 0x 2D4)

偏移地址: 0x 2D4; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:24	CKM_IN	模式 1: IN 通道校验和 模式 2: 上一个周波区块的校验和	R	0
23:16	CKM_IC	模式 1: IC 通道校验和	R	0

		模式 2: 上一个周波区块的校验和		
15:8	CKM_IB	模式 1: IB 通道校验和 模式 2: 上一个周波区块的校验和	R	0
7:0	CKM_IA	模式 1: IA 通道校验和。 模式 2: 上一个周波区块的校验和	R	0

该寄存器在 RN2025 的定义与 RN2026 一致, RN2026 有 UA/UB/UC/IA/IB/IC/IN 一共 7 个通道, RN2025 只有 UA/IA/IB 三个通道

4.5.41.13 DMAWAV_CHECKSUM1(新增 0x 2D8)

偏移地址: 0x 2D8; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:24	Reserved	保留位	R	0
23:16	CKM_UC	模式 1: UC 通道校验和 模式 2: 上一个周波区块的校验和	R	0
15:8	CKM_UB	模式 1: UB 通道校验和 模式 2: 上一个周波区块的校验和	R	0
7:0	CKM_UA	模式 1: UA 通道校验和 模式 2: 上一个周波区块的校验和	R	0

该寄存器在 RN2025 的定义与 RN2026 一致, RN2026 有 UA/UB/UC/IA/IB/IC/IN 一共 7 个通道, RN2025 只有 UA/IA/IB 三个通道

4.5.42 同步采样通道校正和配置寄存器 (新增)

4.5.42.1 DMA_WAVECON2(0x600)

偏移地址: 0x600; 默认值: 0x0000

位	位名称	功能描述	读写标志	复位值
31:11	Reserved	保留位	R	0
10	CIC_AAC_EN	=0: 不使能谐波衰减补偿功能; =1: 使能谐波衰减补偿功能; 使能该功能后, 读取 DMA 波形缓存做 FFT 时就不再需要做谐波系数的补偿。	R/W	0
09	CIC_AGC_EN	=0: 不使能 DMA 波形缓存通道的增益自动调整功能; =1: 使能 DMA 波形缓存通道的增益自动调整功能; 增益自动调整功能, 是为了解决同步采样时工频频率变化引起的增益变化。使能该功能后, 同步采样通道与计量通道之间的增益调整就不再需要, DMA 波形通道得到的波形幅值与计量通道的瞬时波形幅值 (计量通道 8KHz 的瞬时波形采样寄存器(0x300~0x318)) 会保持一致。	R/W	0
08	CIC_DC_EN	=0: 通道直流 OFFSET 校正寄存器 DCOSx(0x68~0x80)不参与 DMA 波形缓存通道的直流 OFFSET 校正;	R/W	0

		=1: 通道直流 OFFSET 校正寄存器 DCOSx(0x68~0x80)参与 DMA 波形缓存通道的直流 OFFSET 校正;		
07	保留	保留	R	0
06:00	WAVE_HPFOFF	DMA 波形通道高通滤波器开关, =0, 使能高通, 默认使能高通; 建议使能高通。 =1, 关闭高通。 具体定义如下: WAVE_HPFOFF[6]: HPFOFF_IN_WAVE WAVE_HPFOFF[5]: HPFOFF_IC_WAVE WAVE_HPFOFF[4]: HPFOFF_IB_WAVE WAVE_HPFOFF[3]: HPFOFF_IA_WAVE WAVE_HPFOFF[2]: HPFOFF_UC_WAVE WAVE_HPFOFF[1]: HPFOFF_UB_WAVE WAVE_HPFOFF[0]: HPFOFF_UA_WAVE	R/W	0

4.5.42.2 DMA_PHS_WAVE (0x604~0x614)

ADC 波形缓存 UA 通道相位校正寄存器

偏移地址	604H	610H	614H
寄存器	PHS_UA_WAVE	PHS_IA_WAVE	PHS_IB_WAVE
默认值	0x00	0x00	0x00

上述相位校正寄存器可以完成波形缓存通道的相位校正, 校正原理同 PHSU/I(通道相位校正寄存器)。

PHS_IA/PHS_IB/PHS_UA 实际有效位数是 10 位。相位校正分为两级:

第一级为 adc 的一位数据流校正, 采样率固定为 2.048Mhz, 分辨率随着工频变化而变化, 以 50Hz 为例, 计算公式: $360/(2.048*10^6/50)=0.008789$ 度。

各寄存器的 PHS[6:0]支持第一级校正, 第一级的校正范围设为 $360 \text{度}/256 \text{点采样}/2=0.7$ 度左右, 实际支持的最大校正值为 87 (十进制, 最大相移为 $87*0.008789=0.7646$ 度), 当 PHS[6:0]写入的值大于 87 时, 按照 87 来做处理。

第一级校正与计量通道的校正原理完全一致, 如果第一级校正范围能够满足应用要求, 这些寄存器可填写与计量通道相同的偏移校正值 (注意计量通道是在 0x80H 基础上做偏移)。

第二级校正为 CIC 抽取滤波器后的正弦波延迟相位校正, 这与计量通道固定采样率 (8Khz) 不同, 如果配置了同步采样, 移动一个采样点代表的相位刻度是固定的 (比如 256 点同步采样, 移动一个采样点相移固定为 $360/256=1.40625$ 度), 实际应用时需要注意这个问题。

第二级校正只考虑 256 点同步采样的情况, 共设置 3 个 bit 位做相位校正 PHS[9:7], 具体定义如下

PHS[9:7]=3'b000: 代表在 CIC 滤波后没有 delay

PHS[9:7]=3'b001: 代表 delay $360 \text{度}/256 \text{点}=1.40625$ 度

PHS[9:7]=3'b010: 代表 delay $2*360 \text{度}/256 \text{点}=2.8125$ 度

PHS[9:7]=3'b011: 代表 delay $3*360 \text{度}/256 \text{点}=4.21875$ 度

PHS[9:7]=3'b100: 代表 delay $4*360 \text{度}/256 \text{点}=5.625$ 度

PHS[9:7]=其他, 保留位, 不允许填写 (如果填写, 效果等同 3'b000)

PHS[9:7]与 PHS[6:0]一起配合, 可以实现 0~6.325 度范围内的相位调整。

如果设置的同步采样点数不是 256 点, 上述公式需要相应的作出调整。

4.5.42.3 DMA_GS_WAVE (0x620~0x630)

偏移地址	620H	62CH	630H
寄存器	GSUA_WAVE	GSIA_WAVE	GSIB_WAVE
默认值	0x0	0x0	0x0

上述寄存器用于 ADC 波形缓存通道的增益校正。增益寄存器为 2 字节有符号数，采用二进制补码形式，最高位为符号位，表示范围(-1, +1)。

校正公式：

如果 $\text{RegGain} \geq 2^{15}$ ，则 $\text{Gain} = (\text{RegGain} - 2^{16}) / 2^{15}$ ；否则 $\text{Gain} = \text{RegGain} / 2^{15}$ ；

其中 RegGain 为通道增益寄存器值。

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

$$\text{IB}' = \text{IB} + \text{IB} * \text{Gain}$$

注意该公式有效范围：保证在乘以通道增益后信号仍在 ADC 满量程范围内

5 全失压测量

全失压是一种窃电状态：当电压小于电能表临界电压，电流回路的负荷电流大于 5% 额定（基本）电流，定义为全失压状态。电能表对全失压事件进行记录，并计算安时值。

RN2025 的 NVM 模块支持全失压状态下的低功耗电流测量，结合 SOC 低功耗模式提供全失压测量完整方案。

5.1 主要特点

- 提供低功耗全失压电流测量（88ms 测量时间，5% Ib 点误差小于 3%，功耗小于 2mA）
- 全失压 NVM 有效值校正
 - 增益校正：NVM 有效值无增益寄存器，增益校正可通过 CPU 软件做，将寄存器值乘以通道增益系数
 - NVM 阈值寄存器配置按理论公式计算后，还需考虑通道增益因素，除以通道增益系数
 - DCOC 校正：NVM 有效值 HPF 可选打开关闭，默认关闭；HPF OFF 情况下需进行 AUTODC 校正，在校正阶段 AUTODC 功能由 EMU 完成，完成后，CPU 从 EMU/DCOS 寄存器读出除 0.512 后填入 LS_DCOS 寄存器
- 全失压测量模式下
 - CPU 工作在 32K 下，电流 ADC 和全失压测量模块工作在 RCH 下，其中 $\text{FADC} = \text{FRCH} / 4$ ， $\text{FNVM} = \text{FRCH}$ ，RCH 典型频率为 3.6863MHz；
 - ADC 只能选择 IA/IB2 路打开，并且可以选择降低功耗；
 - FEMU 关闭，计量停止工作，NVM 有效值测量模块工作；
 - NVM 有效值同时能测量 IA/IB2 路有效值，测量典型时间为 88ms
 - 提供全失压模式 NVM 中断，测量结束或某相电流超过阈值时可通过中断通知 CPU

- 功耗约为 2 mA;
- NVM 模块 HPF 可选打开关闭，默认关闭；AUTODC 功能在校正阶段由 EMU 完成，完成后，从 EMU/DCOS 寄存器读出除 0.512 后填入 LS_DCOS 寄存器
- 计量模式下 25hz 全失压有效值 RMS_Lx1 仍有效

5.2 工作过程

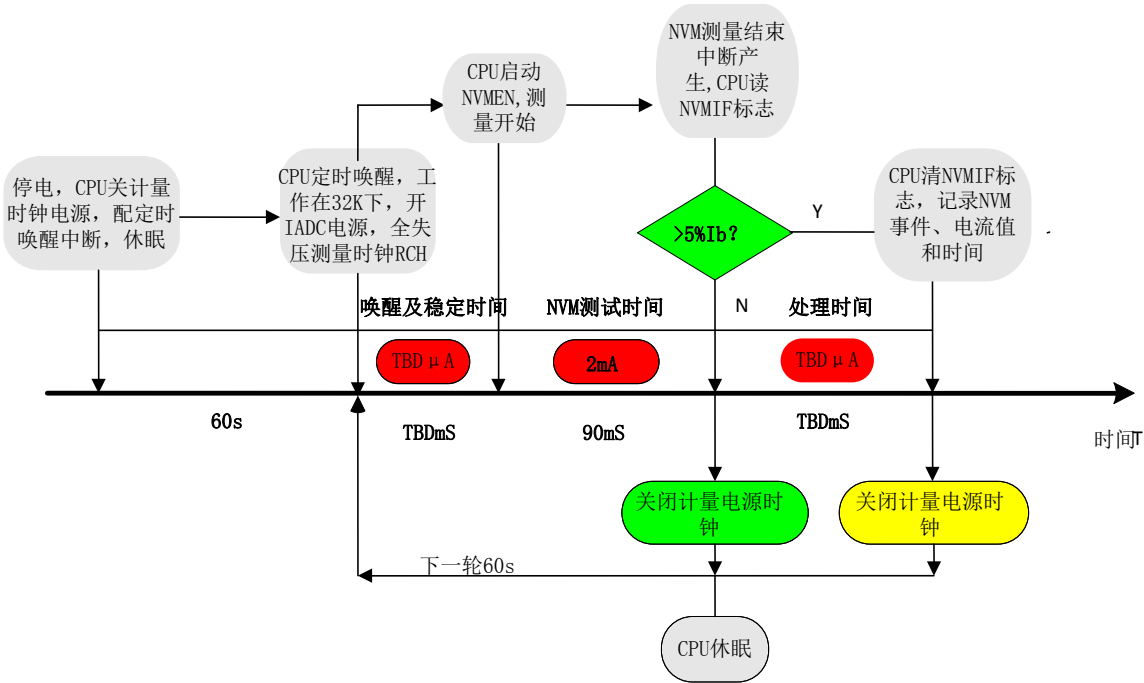


图5-1 电池供电下全失压工作过程

全失压测量过程寄存器配置说明：

- 1.原状态：系统处于 HCM（高频时钟工作模式），NVM 配置好 LS_DCOS_Ix(在全失压测量时直流 offset 校正阶段获得)、LS_THOx 阈值寄存器(默认配置全失压测量时间 88ms)、NVM 中断
- 2.停电发生后：
 - CPU 配置 RTC 中断使能；
 - CPU 配置系统控制寄存器写使能，SYS_PSW=8'h82；
 - CPU 关模块时钟：配置 MOD0_EN, MOD1_EN 寄存器，关闭不需要的模块时钟；
 - CPU 关模块电源：配置 SYS_PD，关闭不用模块电源；
 - CPU 进入 LCM 模式：调用库函数进入 LCM 模式；
 - CPU 关闭系统控制寄存器写使能，SYS_PSW=8'h00
 - CPU 运行 WFI 指令进入休眠模式
- 3. 定时中断产生后
 - CPU 被定时中断唤醒，进入 LCM 模式
 - CPU 配置系统控制寄存器写使能，SYS_PSW=8'h82
 - CPU 打开 RCH 模块：设置 SYSCTL-> OSC_CTRL1 的 bit2 为 0， OSC_CTRL1 &= 0xffffffd；
 - CPU 打开 NVM 所需模块电源，含 IA/IB ADC、VREFA、LDO3：设置 SYSCTL-> SYS_PD 的 bit16,bit17,bit23,bit25 为 0，SYS_PD &= 0xfd7cfff，同时 NVM 计算模块复位释放
 - CPU 打开 NVM 模块时钟：设置 SYSCTL->MOD1_EN 的 bit8 为 1
 - CPU 开启 NVM 寄存器写使能：NVM->NVM_PSW = 0x8B

- CPU 启动 NVM 工作: NVM->LS_CFG 的 bit2 设置为 1, LS_CFG=0x04, NVM 开始工作
 - 软件等待 NVM 测量完成, 在此过程中可查询 NVM->NVM_IF 的 bit0, 如为 1 即代表测量完成, 即可读取测量值; 等待时间依据 LS_THOA 阈值寄存器确定;
 - CPU 查询 NVM->NVM_IF 标志位, 判断是否产生全失压事件, 哪一路电流超过全失压电流阈值
 - 若产生全失压事件, CPU 读超过阈值的相的电流有效值寄存器, 进行增益计算得到准确电流值并保存 (注 5.1), 之后进入过程 4
 - 若未产生全失压事件, CPU 进入过程 4
- 4. 上述工作完成后
- CPU 关闭 NVM 工作:LS_CFG=0x00
 - CPU 关闭 NVM 模块时钟: 设置 SYSCTL->MOD1_EN 的 bit8 为 0
 - CPU 关闭 NVM 启动打开的模块电源, 含 IA/IB ADC、VREFA、LDO3, SYSPD |= 0x02830000
 - CPU 关闭 RCH 模块: 设置 SYSCTL-> OSC_CTRL1 的 bit2 为 1, OSC_CTL1 |= 0x00000002
 - CPU 关闭 NVM 寄存器写使能: NVM->NVM_PSW = 0xDC
 - CPU 关闭系统控制寄存器写使能, SYS_PSW = 8'h00
 - CPU 运行 WFI 指令进入休眠模式, 等待 1s 定时中断唤醒, 之后进入过程 3

注 5.1: 以 IA 通道为例, 当前测量通道的寄存器值, 与正常计量有效值的关系是: $RMS_LA * (1 + IG) = IA * 0.87909 / 16$ 。
 IG 为对应通道校表寄存器的归一化值; 例如当前为 IA 通道, 则计算 GSIA 的归一化值。归一化值计算方法:
 如果 $GSIA > 32768$ 时: $IG = -(0xffff - GSIA + 1) / 32768$;
 如果 $GSIA \leq 32768$ 时: $IG = GSIA / 32768$;

5.3 全失压测量时直流 offset 校正过程

该步骤可在校表时执行

- 1、电表在 Un, 空载情况下, 进行通道的直流 offset 校正;
- 2、电表在正常计量模式时, 需关闭测量通道的数字高通滤波器;
- 3、SYSCTL->ADC_CTL 通道 ADC 增益配置成正常使用时所需倍数;
- 4、设置 SYSCTL->SYS_PD, 开启 VREFA(bit23)、LDO3(bit25)和全失压测量通道电源, 只打开全失压测量通道的电源, 其他 ADC 通道电源关闭;
- 5、SYSCTL->MOD1_EN 的 bit7 和 bit8 设置为 1, 打开 EMU 和 NVM 模块的 APB 时钟;
- 6、SYSCTL->OSC_CTRL1 的 bit1 设置为 0, 打开 RC 时钟;
- 7、打开 NVM 模块密码保护 NVM->PSW 设置为 0x8b;
- 8、设置 NVM->LS_DCOS_lx 为 0, NVM->LS_CFG 的 bit2, bit1 设置为 1, 使能 NVM 模块测量, 关闭 NVM 模块高通;
- 9、开启 EMU 模块密码保护 EMU->SPCMD 设置为 0xE5;
- 10、使能电流通道直流 offset 自动校正功能, EMU->AUOTDC_EN 设置为 0x1;
- 11、等待 2.28S 后, 读取 DCOS_IA 寄存器值除 0.512 后保存在 EEPROM 中, 作为 NVM 模块的直流 offset 校正正值;
- 12、恢复各寄存器的设置;

自动 offset 校正完成。

5.4 寄存器

5.4.1 寄存器列表

基地址	0x40040000
-----	------------

偏移地址	名称	R/W	有效字长	复位值	功能描述
0x0	NVM_IE	R/W	1	0x7	全失压测量中断使能寄存器
0x4	NVM_IF	R/W	1	0x0	全失压测量中断标志寄存器
0x8	LS_CFG	R/W	1	0x2	全失压测量配置寄存器
0xC	LS_DCOS_IA	R/W	2	0x0	全失压 A 路电流直流 Offset 寄存器
0x10	LS_DCOS_IB	R/W	2	0x0	全失压 B 路电流直流 Offset 寄存器
0x14	---	R/W	2	0x0	保留，不要改变默认值
0x18	LS_THOA	R/W	3	0x0	全失压比较 A 路电流阈值寄存器
0x1C	LS_THOB	R/W	3	0x0	全失压比较 B 路电流阈值寄存器
0x20	---	R/W	2	0x0	保留，不要改变默认值
0x24	RMS_LA	R	3	--	全失压 A 路电流有效值
0x28	RMS_LB	R	3	--	全失压 B 路电流有效值
0x2C	---	R	3	--	保留，不要改变默认值
0x30	RMS_LA1	R	3	--	全失压 A 路电流有效值 1
0x34	RMS_LB1	R	3	--	全失压 B 路电流有效值 1
0x38	---	R	3	--	保留
0x3C	NVM_PSW	R/W	1	0x0	全失压写保护寄存器

5.4.2 NVM 中断

NVM_IE

偏移地址：0x0；默认值：0x7

位	位名称	功能描述	读写标志	复位值
3-31	Reserved	保留。	R	
2	NVM_DoneIE	全失压计算完成并且至少有一个 I 通道超过设定阈值；全失压中断使能位： =0：不使能；=1：使能	R/W	1
1	NVCAL_DoneIE	全失压计算完成；中断使能位： =0：不使能；=1：使能	R/W	1
0	Lx1_UDIE	全失压计算有效值固定频率更新（25hz）中断使能位； =0：不使能；=1：使能	R/W	1

NVM_IF

偏移地址：0x4；默认值：0x0

位	位名称	功能描述	读写标志	复位值
6-31	Reserved	保留。	R	
5	---	保留	R/W	0
4	LBGTH_IF	表示 NVM 模式 IB 通道测量结束并大于阈值（IB 溢出）	R/W	0
3	LAGTH_IF	表示 NVM 模式 IA 通道测量结束并大于阈值（IA 溢出）	R/W	0
2	NVM_DoneIF	全失压计算完成并且至少有一个 I 通道超过设定阈值标志；	R/W	0

1	NVCAL_DoneIF	全失压计算完成标志;	R/W	0
0	Lx1_UDIF	全失压计算有效值固定频率更新(25hz)中断标志;	R/W	0

5.4.3 LS_CFG(0x08)

偏移地址: 0x8; 默认值: 0x2

位	位名称	功能描述	读写标志	复位值
3-31	Reserved	保留。	R	
2	NVMEN	=1, 使能该模块; =0 不使能。(NVM 模块需要先配寄存器, 在使能 NVMEN)	R/W	0
1	LS_HPFOFF	=0, 全失压有效值计算通道 HPF 使能 =1, 全失压有效值计算通道 HPF 关闭 默认=1, 高通关闭。 推荐全失压测量时使用直流 offset 自动校正功能, 关闭高通, 以加快 NVM 有效值稳定时间;	R/W	1
0	LSMODE	=0, 选择为全失压计算模式, 全失压有效值计算结果有效; 此配置下, 波形缓存同步跟踪整周波采样功能相关配置位 WAVECON[12:2]配置无效, 不推荐客户使用波形缓存数据。 =1, 选择为波形缓存同步跟踪采样模式, 波形缓存同步跟踪采样功能有效, 即波形缓存控制寄存器 WAVECON[12:2]配置有效。 此配置下, 不推荐客户使用全失压有效值功能且建议 NVMIE 配置为 0。	R/W	0

5.4.4 LS_DCOS_Ix(0x0C~0x10)

偏移地址	0CH	10H
寄存器	LS_DCOS_IA	LS_DCOS_IB
默认值	0x0	0x0

LS_DCOS_Ix用于全失压电流测量时直流Offset校正, 16位有效, Offset寄存器左移6bit与24bit采样值直接相加, 写保护。

校正公式: $RMS_LA'/B'=ABS(RMS_LA/B+LS_DCOS_IA/B*2^7)$

注意, 校正前LS_DCOS_Ix对RMS_Lx的影响:

以 IA 通道为例, 直流 LS_DCOS_IA 与 IA 通路有效值 RMS_LA 的关系如下:

$$RMS_LA' = \sqrt{(RMS_LA^2 + ((LS_DCOS_IA - IAd0) * 2^7) / 2^{23})^2}$$

LS_DCOS_IA: 是指 NVM 下 IA 通道 DCOS 寄存器值, 可以是 0~0xffff 之间任意值

IAd0: 是指 IA 通道空载且 ADC 输入短接到地情况下, DCOS 自动校正完成后的 DCOS_IA 寄存器值

RMS_LA: 是指 NVM 下 IA 通道实际输入的任意不超过满量程波形的有效值的归一化值

RMS_LA': 是指 NVM 下 IA 通道实际读出的有效值寄存器归一化值

5.4.5 LS_THOx(0x18~0x1C)

偏移地址: 0x18; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
24-31	Reserved	保留。	R	
22-23	T_RMSL	=2'b00 NVM 计算时间为 88ms; =2'b01 NVM 计算时间为 120ms; =2'b10 NVM 计算时间为 152ms; =2'b11 NVM 计算时间为 184ms; 该配置适用于 A/B 两路 注 1: 稳定时间 88ms 不够, 应用配置稳定时间推荐大于 120ms 注 2: HPF ON 时, 有效值稳定时间不够, 不推荐 HPF ON 情况的应用。	R/W	0
21-0	LSTHOA	A 路电流全失压测量阈值设置寄存器, 使用时将 RMS_LA[21:0] 与 LS_THOA[21:0] 进行比较, 当 RMS_LA[21:0]>=LS_THOA[21:0]或者 RMS_LB [22]=1 时超过阈值。 LS_THO 阈值配置计算公式: $LS_THOA=K*IA*0.87909/(16*(1+GSIA))$ IA:A 相输入额定电流 I_n 时, A 路电流有效值寄存器值 GSIA:A 路通道增益系数 K:全失压电流门限系数, 如 5% 使用时需考虑通道增益因素	R/W	0

LS_THOB

偏移地址: 0x1C; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
22-31	Reserved	保留。	R	
0-21	LSTHOB	B 路电流全失压测量阈值设置寄存器, 定义同 LSTHOA	R/W	0

5.4.6 RMS_Lx(0x24~0x34)

偏移地址	0x24	0x28	0x30	0x34
寄存器	RMS_LA	RMS_LB	RMS_LA1	RMS_LB1
默认值	0x0	0x0	0x0	0x0

RMS_Lx为单次全失压测量电流有效值, 为23位无符号数, bit23~bit31始终为0;

RMS_Lx未经过通道增益校正, 与正常计量有效值寄存器的关系是:

$$RMS_LA * (1+GSIA) = IA * 0.87909 / 16$$

直接使用时需考虑通道增益因素, 软件乘以通道增益系数。

该寄存器仅适用于全失压低功耗模式, 使用整个流程见5.2章节描述, 注意:

- 1) . NVMEN 0->1, NVM模块仅按配置进行一次全失压有效值RMS_LX计算
- 2) . 按LS_THOA[23:22]配置的计算时间完成后, NVM模块自动置位NVMIF标志位同时更新RMS_LX寄存器3) . CPU清NVMIF并不会清RMS_LX该次计算结果, RMS_LX寄存器会保留至下一次启动全失压检测流程, 或者软件通过配置LS_CFG.NVMEN=0, 再配置LS_CFG.NVMEN=1重新启动一次NVM计算并更新RMS_Lx寄存器。
RMS_Lx1为以固定周期(25Hz)更新全失压测量Ix有效值, 计量模式和全失压模式均能使用。直接使用时

需考虑通道增益因素，软件乘以通道增益系数。

5.4.7 NVM_PSW(0x3C)

偏移地址：3CH；默认值：0x0。

偏移地址	3CH
寄存器	NVM_PSW
默认值	0x0

NVM 密码寄存器，写成 0x8b 之后(0x00~0x20)寄存器可读写，该寄存器读数为 1，写 0xDC 关闭(0x00~0x20)寄存器可写，该寄存器读数为 0。写入其他值，芯片不动作。

上电复位和软件复位后该寄存器回到默认写保护状态。

6 谐波计算引擎

本模块为谐波计算引擎，包含浮点数格式的 FFT 硬件运算单元、整数和浮点数相互转换的运算单元、浮点数运算基本单元、正余弦以及均方根运算单元。

使用谐波计算引擎，可以完成实时的谐波含量运算、谐波功率运算。另外，还可以作为通用的浮点运算单元来使用。

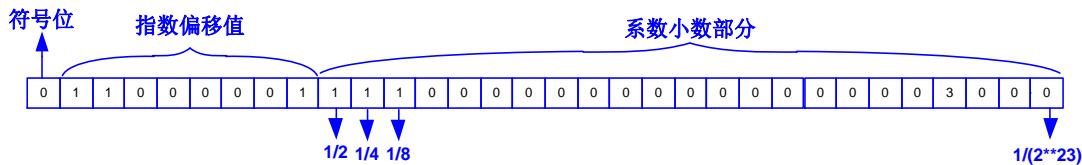
建议用户直接调用锐能微例程，实现谐波含量、谐波功率、谐波电能等运算。软件只需要定时进行简单的配置，即可完成每周波连续的谐波计算，可保证计算的实时性和准确性。

6.1 特点

- 支持整数和单精度浮点数的相互转换；
- 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 运算，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；

6.2 计算引擎基本原理

6.2.1 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 表示正数、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。

6.2.2 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	$1\sim 254$	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。

6.2.3 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；

6.2.4 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

6.2.5 本芯片的 rounding 模式

本芯片支持 6 中舍入方法，根据配置选择：

1. 当配置 rmd=000，为 nearest 舍入；
2. 当配置 rmd=001，为 zero 舍入；
3. 当配置 rmd=010，为 $+\infty$ 舍入；
4. 当配置 rmd=011，为 $-\infty$ 舍入；
5. 当配置 rmd=100，为 up 向上进位；
6. 当配置 rmd=101，为 away 直接舍弃；

6.2.6 整数转浮点数原理

下面简述整数转浮点数的思路与原理。

位置	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
数值	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	1
位置	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
数值	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

1. 符号位：
 - a) 最高位即为浮点数符号位：fp_sign = bit(31)；
 - b) 如果原始数据为负数，则需要转成补码，再做下一步处理；
2. 指数：
 - a) 从最高位开始，查找第一个“1”，所在位置记录为 n。其中指数值就为 n（上表为 27）；

- b) 根据编码规则，指数表示 $fp_exp=n+127$;
3. 小数:
- a) 如果 $n>23$ ，则从 $n-1$ 位开始，截取 24bit，记为 m ；对原始数据中剩余位做 rounding 处理，决定 m 最低位的值。 m 即为小数部分；
- b) 如果 $n=23$ ，则从 $n-1$ 位开始，截取 24bit，直接做为小数部分；
- c) 如果 $n<23$ ，则从 $n-1$ 位开始，截取剩余所有位，并且低位补 0，作为小数部分；

6.2.7 浮点数转整数原理

1. 判断原始数据是否为特殊数 (0, NaN, $\pm\infty$)
2. 根据原始数据的指数部分 e ，判断转化结果是否溢出。如果指数部分 $e>32+127$ ，则转换结果溢出；否则进行下面步骤；
3. 取出 23 位小数部分 m ，最高位补“1”，凑成 24 位小数 m ；
4. 把小数 m 左移 e 位，高位 ((32- e) bit) 和低位 (e bit) 补“0”，得到 56 位数据；
5. 结果右移 24 位，得到最终结果；

6.2.8 浮点数乘法原理

1. 特殊数据相乘：
 - a) NaN 和任何数相乘都是 NaN，
 - b) 0 和无穷相乘为 NaN，
 - c) 0 和其他数相乘为 0。
2. 普通数据相乘：
 - a) 符号位相异或得到符号位结果；
 - b) 指数部分相加再减去 127 ($e = e_0 + e_1 - 127$)；
 - c) 系数部分先分别扩展一位 (共 24bit) 之后做乘法，得到一个 48bit 数据 m ，判断结果是否溢出，如果溢出，则 $e=e+1$ ，判断指数 e 是否溢出，同时把 m 右移三位，否则把 m 右移 2 位；取高 23 位为系数结果。
3. 溢出处理，支持 IEEE754 规定的 4 中溢出处理方式，通过寄存器配置。

6.2.9 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；

6.2.10 蝶形运算原理

$$X'_1(k) = X_1(k) + W_N^k X_2(k)$$

$$X'_2(k) = X_1(k) - W_N^k X_2(k)$$

其中 $X_1(k)$ 、 $X_2(k)$ 为输入， $X'_1(k)$ 、 $X'_2(k)$ 为蝶形运算输出。

如果原始数据为复数，则有：

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

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

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

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

所以：

$$X_1'(k) = X_1(k) + W_N^k X_2(k)$$

$$= (X_{1r}(k) + jX_{1i}(k)) + (W_{Nr}^k + jW_{Ni}^k) * (X_{2r}(k) + jX_{2i}(k))$$

把实部虚部提取，

$$X_{1r}' = X_{1r} + W_{Nr}^k X_{2r} + W_{Ni}^k X_{2i}$$

$$X_{1i}' = X_{1i} + W_{Nr}^k X_{2i} - W_{Ni}^k X_{2r}$$

同理：

$$X_{2r}' = X_{1r} - W_{Nr}^k X_{2r} - W_{Ni}^k X_{2i}$$

$$X_{2i}' = X_{1i} - W_{Nr}^k X_{2i} + W_{Ni}^k X_{2r}$$

6.3 运算指令

6.3.1 整数转浮点数 (int2fp/int2fp_dma)

6.3.1.1 单数据转换



整数转浮点数

该模式把寄存器输入的 **32bit (或者 24bit) 有符号整数** 转化为 32 位单精度浮点数格式，输出到寄存器，CPU 可读。

支持对转换后的浮点数做归一化处理，也就是除以 2^n 。其中 n 可配置为 0~32，当配置为 0 时，表示不做归一化处理。

默认输入寄存器整数为 32bit 有符号数，对于特殊应用，如果输入整数为 24bit 有符号数，则需要配置寄存器 $MAC_CTL0[28]=1$ ，选择该模式。

软件操作流程：

- 1) 配置寄存器 $MAC_CTL0[0]=1$ 选择整数转浮点数单点模式；
- 2) 配置寄存器 $MAC_CTL0[16:14]$ 选择浮点数的 [rounding mode](#)；
- 3) 配置 $MAC_CTL0[21:17]$ 选择归一化级数；
- 4) 对于 24bit 输入数据，配置 $MAC_CTL[0]=1$ ；
- 5) 往寄存器 MAC_IN0 写入待转换的原始整数；
- 6) 读取寄存器 MAC_OUT0 获得转换后的浮点数值；

软件在配置完模式之后，只要配置寄存器 MAC_IN0 就会启动转换，计算结果保存到寄存器 MAC_OUT0 ，所以配置原始数值之后直接读取结果寄存器可以获得转换后的结果。

注意，如果软件需要连续转化多次，上面的模式配置只需要执行一次，在切换模式之前，每次只要写寄存器 MAC_IN0 ，就会执行转换。

6.3.1.2 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_CTL0[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

6.3.2 浮点数转整数 (fp2int/fp2int_dma)

6.3.2.1 单数据转换



浮点数转整数

该模式把寄存器输入的 32bit 浮点数转化为 **32 位有符号整数**格式，输出到寄存器，CPU 可读。支持转换之前先把浮点数放大 2^n 。其中 n 可配置为 0~32，当配置为 0，表示不放大。

软件操作流程：

- 1) 配置寄存器 MAC_CTL0[2]=1 选择浮点数转整数单点模式；
- 2) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#)；
- 3) 如果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。

6.3.2.2 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。

6.3.3 浮点数乘法 (fp_mult)



浮点数乘法

该模式对输入寄存器 `MAC_IN0` 和 `MAC_IN1` 的数据，做一个浮点数乘法。对输入和输出数据格式支持下列四种模式：

- 1) 输入整数、输出整数
- 2) 输入整数，输出浮点数
- 3) 输入浮点数、输出整数
- 4) 输入浮点数，输出浮点数

如果输入为整数，硬件会先把整数转换成浮点数格式，然后再进行浮点乘法；运算得到浮点数格式的乘积，如果选择输出整数格式，则硬件会把浮点格式的乘积转换为整数格式。

软件操作流程：

- 1) 配置寄存器 `MAC_CTL0[4]=1` 选择浮点数乘法模式；
- 2) 配置寄存器 `MAC_CTL0[16:14]`选择浮点数的 [rounding mode](#)；
- 3) 配置寄存器 `MAC_CTL0[13:12]`选择输入输出数据模式；
- 4) 配置寄存器 `MAC_IN0` 和 `MAC_IN1`（乘数和被乘数）；
- 5) 等待标志位 `MAC_FLG[9]=1`，表示乘法完成；
- 6) 读取寄存器 `MAC_OUT0` 获得乘积；

该模式下，每次配置 `MAC_IN1` 寄存器，会自动启动乘法运算，所以软件应该先配置好 `MAC_IN0` 寄存器，再配置 `MAC_IN1` 寄存器。而且每次配置 `MAC_IN1` 寄存器，会自动清除上一次的标志位，直到当前计算完成，再置起标志位。

由于输入输出数据格式不同，需要对数据做格式转换，所以做一次浮点乘法运算的时间与模式相关。

6.3.4 浮点数加法 (fp_add)



浮点数加法

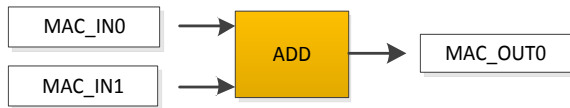
该模式对输入寄存器 MAC_IN0 和 MAC_IN1 的数据，做一个浮点数加法。输入寄存器的数据格式必须为浮点数格式，输出结果也为浮点数格式。

软件操作流程：

- 1) 配置寄存器 MAC_CTL0[5]=1 选择浮点数加法模式；
- 2) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#)；
- 3) 配置寄存器 MAC_IN0 和 MAC_IN1（加数和被加数）；
- 4) 读取寄存器 MAC_OUT0 获得和；

该模式下，每次配置 MAC_IN1 寄存器，会自动启动加法运算，所以软件应该先配置好 MAC_IN0 寄存器，再配置 MAC_IN1 寄存器。

6.3.5 浮点数减法 (fp_sub)



浮点数加法

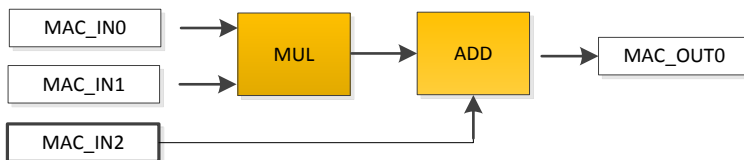
该模式对输入寄存器 MAC_IN0 和 MAC_IN1 的数据，做一个浮点数减法。输入寄存器的数据格式必须为浮点数格式，输出结果也为浮点数格式。

软件操作流程：

- 1) 配置寄存器 MAC_CTL0[6]=1 选择浮点数减法模式；
- 2) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#)；
- 3) 配置寄存器 MAC_IN0 和 MAC_IN1（被减数和减数）；
- 4) 读取寄存器 MAC_OUT0 获得差；

该模式下，每次配置 MAC_IN1 寄存器，会自动启动减法运算，所以软件应该先配置好 MAC_IN0 寄存器，再配置 MAC_IN1 寄存器。

6.3.6 浮点数乘加运算 (fp_mladd)



浮点数乘加运算

该模式对输入寄存器 MAC_IN0 和 MAC_IN1 的数据，先做一个浮点数乘法，然后再与 MAC_IN2 做加法，最后输出为 MAC_OUT0。输入寄存器的数据格式必须为浮点数格式，输出结果也为浮点数格式。

软件操作流程：

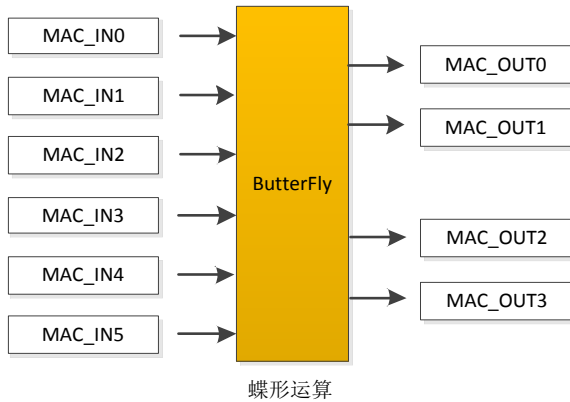
- 5) 配置寄存器 MAC_CTL0[7]=1 选择浮点数乘加模式；
- 6) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#)；

- 7) 配置寄存器 MAC_IN0、MAC_IN1 和 MAC_IN2（乘数、被乘数和被加数）；
- 8) 读取寄存器 MAC_OUT0 获得乘加结果；

该模式下，每次配置 MAC_IN2 寄存器，会自动启动乘加运算，所以软件应该先配置好 MAC_IN0、MAC_IN1 寄存器，再配置 MAC_IN2 寄存器。

6.3.7 浮点数蝶形运算（单次）(btfy /btfy_dma)

6.3.7.1 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：

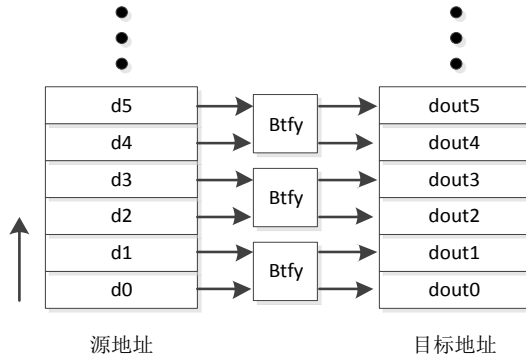
- 1) 配置寄存器 MAC_CTL0[8]=1 选择 butterfly 单次模式；
 - 2) 配置寄存器 MAC_CTL0[16:14]选择浮点数的 [rounding mode](#)；
- 3) 配置寄存器 MAC_IN0/1/2/3/4/5 配置输入数据；
 - MAC_IN0= X_{1r}
 - MAC_IN1= X_{1i}
 - MAC_IN2= X_{2r}
 - MAC_IN3= X_{2i}
 - MAC_IN4= W_{Nr}^k
 - MAC_IN5= W_{Ni}^k
- 4) 配置寄存器 MAC_CTL1[0]=1 启动 butterfly 运算；
- 5) 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。

6.3.7.2 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。

6.3.7.3 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；

6.3.7.4 倒序搬运数据 (bitrev_move)

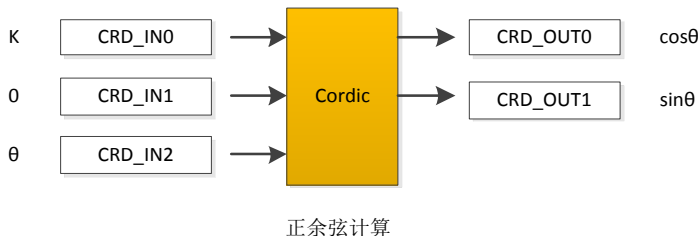
该模式下，硬件自动把 SRAM 中一段原始数据根据 bit reverse 算法，重新排序，并且写到 SRAM 中指定位置。

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；

6.3.8 正弦余弦计算 (sin_cos)



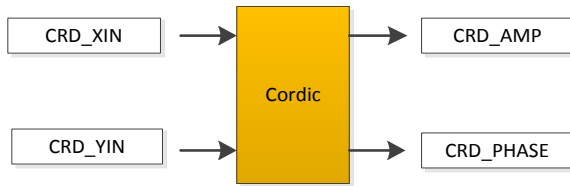
该模式支持输入角度，硬件计算 sin 和 cos 函数。

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

6.3.9 均方根和反正切 (fp_sqrt/ fp_atan)



均方根

该模式计算输入数据 CRD_XIN 和 CRD_YIN 的均方根。

采用 cordic 向量模式，配置 $X_0=X$ ， $Y_0=Y$ ， $Z_0 = 0$ ，输出为 $\left[P\sqrt{X_1^2 + Y_1^2}, 0, \arctan\left(\frac{Y_1}{X_1}\right) \right]$

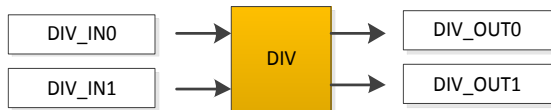
软件配置流程：

- 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=

$$\sqrt{X_0^2 + Y_0^2}/4*P, \text{ CRD}_{\text{PHASE}} = \arctan\left(\frac{Y_0}{X_0}\right) * 2^{31}/\pi$$

如果配置了中断使能信号，则计算完成之后，产生中断。

6.3.10 浮点数除法器 (fp_div)



除法

该模式是一个浮点数的除法运算，输入两个浮点数，做除法之后输出商和余数。

软件配置流程：

- 1) 配置被除数到寄存器 DIV_IN0;
- 2) 配置除数到寄存器 DIV_IN1;
- 3) 配置寄存器 MAC_CTL1[2]=1 启动除法运算;
- 4) 等待完成标志位 MAC_FLG[4]，读取寄存器 DIV_OUT0 获得商;

6.4 实现说明

6.4.1 整数转浮点数

由于输入的整数位宽为 32 位，转换之后的浮点数指数范围很小，所以不会有溢出，并且做归一化的时候最高支持 32 次方，不需要做溢出处理。

归一化，就是把指数部分减去 n，小数部分不变。

6.4.2 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。

6.4.3 FFT 说明

软件只需提前把原始数据存放到 SRAM 中，并且配置好 DMA 起始地址、目标地址，数据长度。硬件自动取数运算，得到 butterfly 的结果。如果数据长度为 64，则每一阶有 $64/2=32$ 次蝶形运算，一共 $\log_2 64 = 6$ 阶，所以硬件会自动算完 $32*6$ 次蝶形运算，并且把结果保存到 SRAM 指定位置，产生完成标志或中断。

硬件每算一次蝶形运算，需要读 4 次 SRAM 获得原始数据（第一轮原始输入数据虚部为 0），写 4 次 SRAM 保存该次蝶形运算结果，（如果参数表格存放 SRAM，还需要读 2 次 SRAM 获取参数）。所以完成一段指定长度数据需要访问 SRAM 周期数为：

$$k_0 = \frac{n}{2} * \log_2 n * (4 + 4) - \frac{n}{2} * 2, \text{ 参数表硬件固化};$$

$$k_0 = \frac{n}{2} * \log_2 n * (4 + 4 + 2) - \frac{n}{2} * 2, \text{ 参数表存放 SRAM};$$

点数	读写 SRAM 周期	
	硬件表	SRAM 表
64	1472	1856
128	3456	4352
256	7936	9984
512	17920	22528
1024	39936	50176

完成 FFT 需要读写 SRAM 周期数

硬件读写 SRAM 加上 butterfly 运算，完成一次 butterfly 最快时间为 10 个时钟周期（参数表硬件固化）或 12 个时钟周期（参数表存放 SRAM）。

时钟周期	1	2	3	4	5	6	7	8	9	10	11	12
硬件行为	读	读	读	读	读	读						
		寄存	寄存	寄存	寄存	寄存	寄存					
								乘	乘	乘	乘	
								加	加	加	加	
								加	加	加	加	
									写	写	写	写

所以总时间为：

$$k_0 = \frac{n}{2} * \log_2 n * 10, \text{ 参数表硬件固化};$$

$$k_0 = \frac{n}{2} * \log_2 n * 12, \text{ 参数表存放 SRAM};$$

点数	FFT 时钟周期		FFT 时间 (us)	
			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。

6.4.4 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。

6.4.4.1 向量模式

确认位宽：

1. 首先，因为原始数据输入 x 和 y 的位宽是 32bit;
2. 该模式下，输入 $z=0$ ，所以 \arctan 的放大倍数跟输入没有关系，可以任意取。此时只需要根据所需的输出角度精度来决定 \arctan 的放大倍数 n。
3. 根据 \arctan 的放大倍数，决定迭代次数。
4. 最终输出的角度精度，跟 \arctan 的放大倍数 n 相关。输出的角度结果必须除以 n，才是实际角度（因为这个角度是累加出来的， $z(i+1)=z(i)-\arctan(1/2^i)$;左右两边同时放大 n）。如果输出结果是取高位输出也是可以的（运算过程比较准确，输出结果比较简短）。

向量模式下，输入 x 和 y 可以等比例放大，计算出来的精度比较高，所以，对于比较小的 x 和 y，只需先放大，再代入 cordic 运算就能得到更高精度的结果。

首先需要做预处理，把所有角度移到第 1 象限内，如果是第 2 象限，移到第一象限后，初始角度 $z=90$ 度。第 3 象限，初始角度 $z=180$ 度；第 4 象限，初始角度 $z=270$ 度。

```

if(x<0){
    x=-x;
    y=-y;
    z=z-(1~3)pi/2*n; //n 是放大倍数
}
    
```

6.4.4.2 旋转模式

确认位宽：

- 1 模块输入为弧度，范围是 $-\pi$ ： π 。假设输入弧度用 n 位数表示，所以输入弧度放大倍数为 $p0=(2^n)/2\pi$ 。
- 2 根据公式， $z(i+1)=z(i)-\arctan(1/2^i)$ ，决定了做加减法运算时， \arctan 放大倍数与输入角度 z 放大倍数相同。 $\text{atan_t}(n)=p0*\text{atan}(n)$ ，其他参数值按相同比例放大 $p0$ 倍。
- 3 正常情况下， $\text{atan_t}(1)=p0*\text{atan}(n)=2^{(n-3)}$ ，所以参数表位宽为 $n-3$ 。如果为了减小定点化的误差，可以把参数表的位数定的更长，计算的时候，只需要输入数据右移放大相同倍数，参与加减。

比如 atan 定点化时位宽为 $n-3+5$ ，

则运算前放大

$$x_in = \{x_in, 5'h0\};$$

$$y_in = \{y_in, 5'h0\};$$

$$z_in = \{z_in, 5'h0\};$$

迭代后输出取高位。

- 4 迭代之后输出结果也是放大了 $p0$ 倍，可以根据输出数据的位宽，调整输出的放大倍数。

旋转模式实现过程：

1. 输入 $X_0 = k$, $Y_0 = 0$, $Z_0 = \theta$; 输入角度范围是 $-\pi \sim \pi$;
2. 首先对角度做预处理，翻转到 $0 : \pi/2$ 范围内;
3. 按照公式迭代得到结果;
4. 结果乘以增益系数 K ;
5. 输入 x 、 y 的位宽与迭代次数相关。因为每次移 1bit。

6.5 寄存器

6.5.1 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_IN0	R/W	4	0x0	MAC 单元数据输入寄存器 0
0xC	MAC_IN1	R/W	4	0x0	MAC 单元数据输入寄存器 1
0x10	MAC_IN2	R/W	4	0x0	MAC 单元数据输入寄存器 2
0x14	MAC_IN3	R/W	4	0x0	MAC 单元数据输入寄存器 3
0x18	MAC_IN4	R/W	4	0x0	MAC 单元数据输入寄存器 4
0x1C	MAC_IN5	R/W	4	0x0	MAC 单元数据输入寄存器 5
0x20	MAC_OUT0	R/W	4	0x0	MAC 单元数据输出寄存器 0
0x24	MAC_OUT1	R/W	4	0x0	MAC 单元数据输出寄存器 1
0x28	MAC_OUT2	R/W	4	0x0	MAC 单元数据输出寄存器 2
0x2C	MAC_OUT3	R/W	4	0x0	MAC 单元数据输出寄存器 3
0x30	DIV_IN0	R/W	4	0x0	除法单元数据输入 0
0x34	DIV_IN1	R/W	4	0x0	除法单元数据输入 1
0x38	DIV_OUT0	R/W	4	0x0	除法单元数据输出

0x3C	DMA_SRBADR	R/W	4	0x0	DMA 源数据 FFT 实部起始地址
0x40	DMA_SIBADR	R/W	4	0x0	DMA 源数据 FFT 虚部起始地址
0x44	DMA_PRBADR	R/W	4	0x0	DMA 旋转因子表格实部起始地址
0x48	DMA_PIBADR	R/W	4	0x0	DMA 旋转因子表格虚部起始地址
0x4C	DMA_TRBADR	R/W	4	0x0	DMA 目标数据 FFT 实部起始地址
0x50	DMA_TIBADR	R/W	4	0x0	DMA 目标数据 FFT 虚部起始地址
0x54	DMA_LEN	R/W	2	0x0	DMA 长度配置
0x58	FFT_IE	R/W	2	0x0	FFT 中断使能寄存器
0x5C	FFT_FLG	R/W	2	0x0	FFT 标志寄存器
0x60	ALU_STA0	R/W	4	0x0	ALU 状态寄存器 0
0x64	ALU_STA1	R/W	2	0x0	ALU 状态寄存器 1
0x68	CRD_CTL	WO	1	0x0	Cordic 控制寄存器
0x6C	CRD_XIN	R/W	4	0x0	Cordic 向量模式数据 X 输入
0x70	CRD_YIN	R/W	4	0x0	Cordic 向量模式数据 Y 输入
0x74	CRD_AMP	R	4	0x0	Cordic 向量模式幅值输出
0x78	CRD_PHASE	R	4	0x0	Cordic 向量模式角度输出
0x7C	CRD_ANGLE	R	4	0x0	Cordic 旋转模式角度输入
0x80	CRD_COS	R	4	0x0	Cordic 旋转模式余弦输出
0x84	CRD_SIN	R	4	0x0	Cordic 旋转模式正弦输出
0x88	CRD_IE	R/W	1	0x0	Cordic 中断使能
0x8C	CRD_FLG	R/W	1	0x0	Cordic 完成标志寄存器

6.5.2 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
31:29	Reserved	保留。	R	0
28	I2F_PRE_EN	整数转浮点数预处理。配置为 1，表示每 4 字节中，最高字节数据无效，有效数据仅为 3 字节；配置为 0，表示 4 字节有效。	R/W	0
27	FFT_TB_EN	FFT 使用芯片固化参数表，仅 64 点和 128 点有效。 =0：不使用芯片固化参数表 =1：使用芯片固化参数表	R	0
26:22	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
21:17	I2F_DIV	整数转浮点数之后，归一化阶数配置	R/W	0

		0 表示不做归一化处理。 n 表示除以 2^n 。		
16:14	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
13	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
12	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
11	BIT_RV	bit reverse 模式使能 =0：不使能 =1：使能	R/W	0
10	FFT	FFT 模式使能 =0：不使能 =1：使能	R/W	0
9	BTFY_DMA	蝶形运算，DMA 使能 =0：不使能 =1：使能	R/W	0
8	BTFY_ONCE	蝶形运算，单次使能 =0：不使能 =1：使能	R/W	0
7	FP_MLAD	浮点数乘加运算，模式使能 ($y=ax+b$) =0：不使能 =1：使能	R/W	0
6	FP_SUB	浮点数减法，模式使能 =0：不使能 =1：使能	R/W	0
5	FP_ADD	浮点数加法，模式使能 =0：不使能 =1：使能	R/W	0
4	FP_MUL	浮点数乘法，模式使能 =0：不使能 =1：使能	R/W	0
3	FP2INT_DMA	浮点数转整数，DMA 使能 =0：不使能 =1：使能	R/W	0

2	FP2INT_ONCE	浮点数转整数，单次使能 =0：不使能 =1：使能	R/W	0
1	INT2FP_DMA	整数转浮点数，DMA 使能 =0：不使能 =1：使能	R/W	0
0	INT2FP_ONCE	整数转浮点数，单次使能 =0：不使能 =1：使能	R/W	0

6.5.3 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
31:3	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为 1 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加 6.3 章节。

6.5.4 MAC_IN0 (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
31:0	MAC_IN0	乘加单元数据输入端	R/W	0

6.5.5 MAC_IN1 (0x0C)

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

6.5.6 MAC_IN2 (0x10)

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

6.5.7 MAC_IN3 (0x14)

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
------	---------	-----------	-----	---

6.5.8 MAC_IN4 (0x18)

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

6.5.9 MAC_IN5 (0x1C)

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

6.5.10 MAC_OUT0 (0x20)

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

6.5.11 MAC_OUT1 (0x24)

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

6.5.12 MAC_OUT2 (0x28)

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

6.5.13 MAC_OUT3 (0x2C)

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

6.5.14 DIV_IN0 (0x30)

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

6.5.15 DIV_IN1 (0x34)

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

6.5.16 DIV_OUT0 (0x38)

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

6.5.17 DMA_SRBADR (0x3C)

偏移地址: 3CH; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:16	Reserved	保留。	R	0
15:0	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

6.5.18 DMA_SIBADR (0x40)

偏移地址: 40H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:16	Reserved	保留。	R	0
15:0	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

6.5.19 DMA_PRBADR (0x44)

偏移地址: 44H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:16	Reserved	保留。	R	0
15:0	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

6.5.20 DMA_PIBADR (0x48)

偏移地址: 48H; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值

31:16	Reserved	保留。	R	0
15:0	DMA_PIBADR	DMA 源数据起始地址：（Word 地址） FFT 或 BTFY 模式下：DMA 旋转因子参数表虚部起始地址	R/W	0

6.5.21 DMA_TRBADR (0x4C)

偏移地址：4CH；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:16	Reserved	保留。	R	0
15:0	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

6.5.22 DMA_TIBADR (0x50)

偏移地址：50H；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:16	Reserved	保留。	R	0
15:0	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

6.5.23 DMA_LEN (0x54)

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

6.5.24 FFT_IE (0x58)

偏移地址：58H；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	MULT_IE	浮点数乘法中断使能	R/W	0
8	BITREV_IE	BIT REVERSE 模式中断使能	R/W	0
7	FFT_IE	FFT 模式中断使能	R/W	0
6	BTFY_DMA_IE	蝶形运算 DMA 中断使能	R/W	0
5	BTFY_ONCE_IE	单次蝶形运算中断使能	R/W	0
4	F2I_DMA_IE	浮点数转整数 DMA 中断使能	R/W	0

3	I2F_DMA_IE	整数转浮点数 DMA 中断使能	R/W	0
2	DIV_IE	除法单元中断使能	R/W	0
1	DMA_IE	DMA 中断使能	R/W	0
0	MAC_IE	乘加单元中断使能	R/W	0

备注：以上 IE 控制位，=1 使能中断，=0 不使能中断。

6.5.25 FFT_FLG (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
31:10	Reserved	保留。	R	0
9	MULT_DONE	浮点数乘法完成中断标志 写 1 清 0	R/W	0
8	BITREV_DONE	BIT REVERSE 模式中断标志 写 1 清 0	R/W	0
7	FFT_DONE	FFT 模式中断标志 写 1 清 0	R/W	0
6	BTFY_DMA_DONE	蝶形运算 DMA 中断标志 写 1 清 0	R/W	0
5	BTFY_ONCE_DONE	单次蝶形运算中断标志 写 1 清 0	R/W	0
4	F2I_DMA_DONE	浮点数转整数 DMA 中断标志 写 1 清 0	R/W	0
3	I2F_DMA_DONE	整数转浮点数 DMA 中断标志 写 1 清 0	R/W	0
2	DIV_DONE	除法单元中断标志 写 1 清 0	R/W	0
1	DMA_DONE	DMA 完成信号中断标志 写 1 清 0	R/W	0
0	MAC_DONE	乘加单元计算 (MUL 或者 BTFY_ONCE) 完成，写“1” 清除该位	R/W	0

6.5.26 ALU_STA0 (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
31:24	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	R	1

		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。		
--	--	--	--	--

6.5.27 ALU_STA1 (0x64)

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

6.5.28 CRD_CTL (0x68)

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

6.5.29 CRD_XIN (0x6C)

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

6.5.30 CRD_YIN (0x70)

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

6.5.31 CRD_AMP (0x74)

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

6.5.32 CRD_PHASE (0x78)

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

6.5.33 CRD_ANGLE (0x7C)

偏移地址：7CH；默认值：0x0

位	位名称	功能描述	读写标志	复位值
31:0	ANGLE_IN	Cordic 旋转模式弧度输入，范围为 $[-\pi:\pi]$ 输入 $-1 * 2^{31}$ 表示 $-\pi$ ，输入 $1 * 2^{31}$ 表示 π 寄存器值= $\text{radian} * 2^{31}/\pi$	R/W	0

6.5.34 CRD_COSINE (0x80)

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

6.5.35 CRD_SINE (0x84)

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

6.5.36 CRD_IE (0x88)

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

6.5.37 CRD_FLG (0x8C)

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

6.6 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。

6.6.1 完整方案说明：

- ADC 采样数据整数 d_0 ，存到 SRAM 中，位宽为 24 位二进制补码格式。
- 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 d_1 。表示的范围为 $-1\sim 1$ 。
- FFT 运算：对所有转换完的数 d_1 做 FFT 运算的输入，复数格式，其中实部为 d_1 ，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。
- 计算谐波含量：

```
repeat(6){
     $Y_0 = \sqrt{Re_0 * Re_0 + Im_0 * Im_0}$ ,
    for(n=1;n<42;n++){
         $Y_n = \sqrt{Re_n * Re_n + Im_n * Im_n}$ ,
         $Y_{in} = k * Y_n / Y_0$ ,
    }
}
```

计算 41 次谐波含量。

电流和电压都需要单独做谐波含量 Y_u 和 Y_i 。

- 相角计算和谐波功率：

```
repeat(3){
    for(n=1;n<42;n++){
         $A_{un} = \text{atan}(Re_n / Im_n)$ , //对电压数据
         $A_{in} = \text{atan}(Re_n / Im_n)$ , //对电流数据
         $A = ((A_u - A_i) + A_0 * n) * 180 / \pi$ ,
         $PF_n = \cos(A * \pi / 180)$ 
    }
}
```

计算 40 次相角，不需要算基波的。要求精度 0.01 度。

- 计算谐波功率： $Ph_n = FSA * Y_{un} * Y_{in} * PF_n$ ，算 41 次谐波之后累加。其中 FSA 从外部获取。

6.6.2 操作流程：

- ADC 采样数据 $sdata_0$ 缓存到 SRAM，每个点数据 24bit；
- 使用 `i2f_dma` 模式把 SRAM 中的 $sdata_1$ 转换成浮点数格式并且做归一化得到 $sdata_2$ ；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。
- 使用 `bitrev_mode` 模式，做 `bitreverse` 把 $sdata_2$ 变换成 $sdata_3$ ；
- 使用 FFT 计算模块，把 $sdata_3$ 做 FFT 计算，得到结果（软件需要提前把 FFT 计算表格写到 SRAM）；
- 使用硬件 `cordic` 模块，除法器模块，计算谐波含量以及谐波功率；

7 计量自检测功能

注意：B 版芯片对于计量自检测做了升级优化，请直接调用锐能微库函数即可。

7.1 功能描述

7.1.1 电流通道精度自检测功能

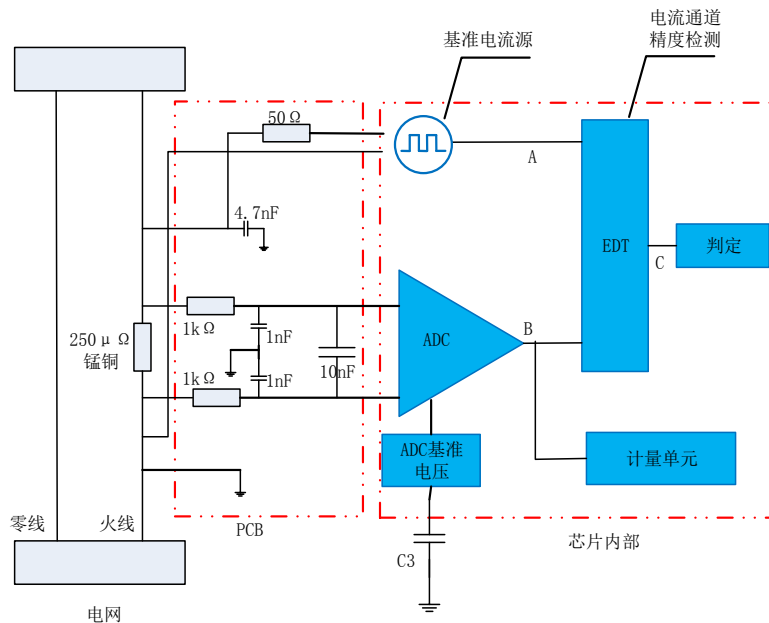
内置高稳定性基准电流源

温度特性： $<20\text{ppm}/^{\circ}\text{C}$

根据软件控制，对计量电流通道发射 40mA 的高频信号，通过测量该信号的幅值和相位，对电流通道的硬件做自检测，可判断出锰铜、锰铜断线（或 CT、CT 断线、）抗混叠电阻电容、计量芯片基准电压等部件的故障造成的计量精度变化。

幅值测量精度： 0.4% （详见第 4 章电气特性）

相位测量精度： $<0.03^{\circ}$



具体电路请参考锐能微参考原理图。

7.1.2 电压通道精度自检测功能

各相提供 2 路高稳定性基准电流源：

- $10\ \mu\text{A}$
- $400\ \mu\text{A}$

分别覆盖电压链路不同故障。

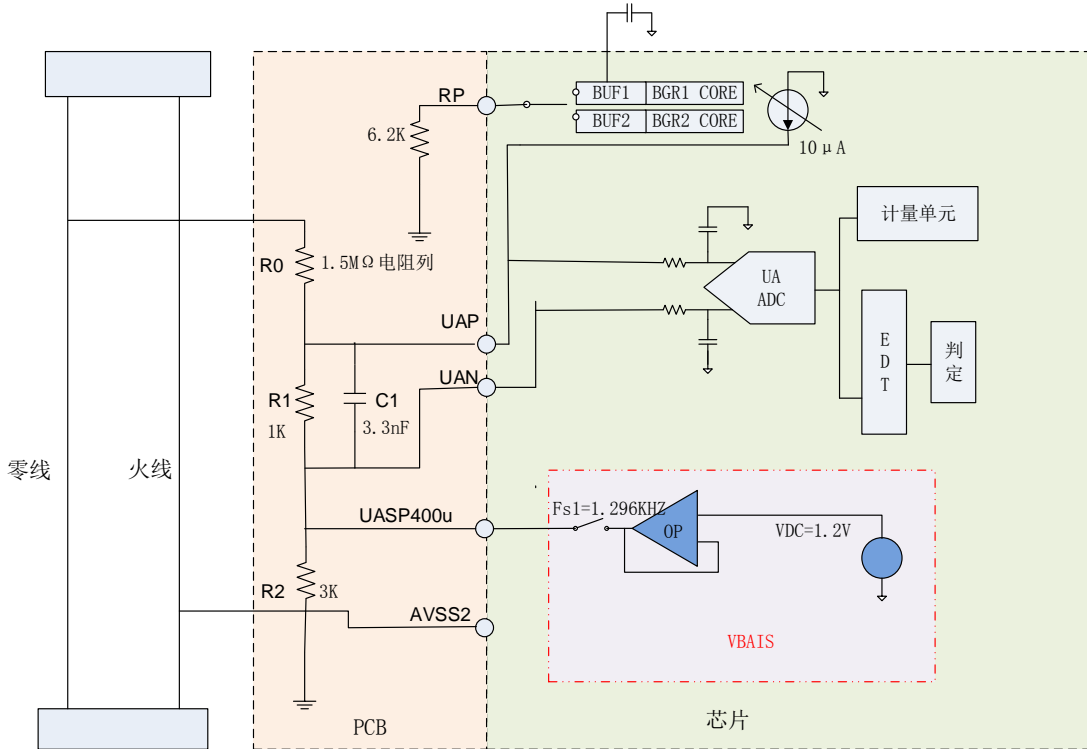
其中 $10\ \mu\text{A}$ 支路覆盖如下故障：

- R1/C1
- ADC
- ADC 基准电压及外围电容

$10\ \mu\text{A}$ 支路的测量精度如下：

- 幅值测量精度： $<0.1\%$

- 相位测量精度: $<0.03^\circ$
- 400uA 支路覆盖如下故障:
- R0-分压电阻列
- 400uA 支路的测量精度如下:
- 幅值测量精度: $<0.5\%$
 - 相位测量精度: $<0.03^\circ$



7.2 寄存器

注意: 用户使用 EDT 相关功能, 必须调用锐能微 EDT 库函数。相关说明参考《锐能微 RN202X_RN7326 应用笔记 005-EDT (误差自检模块) 库函数说明》

8 RTC

8.1 概述

RTC 模块提供实时时钟、振荡器温度补偿、日历、闹钟、时钟脉冲输出等功能。

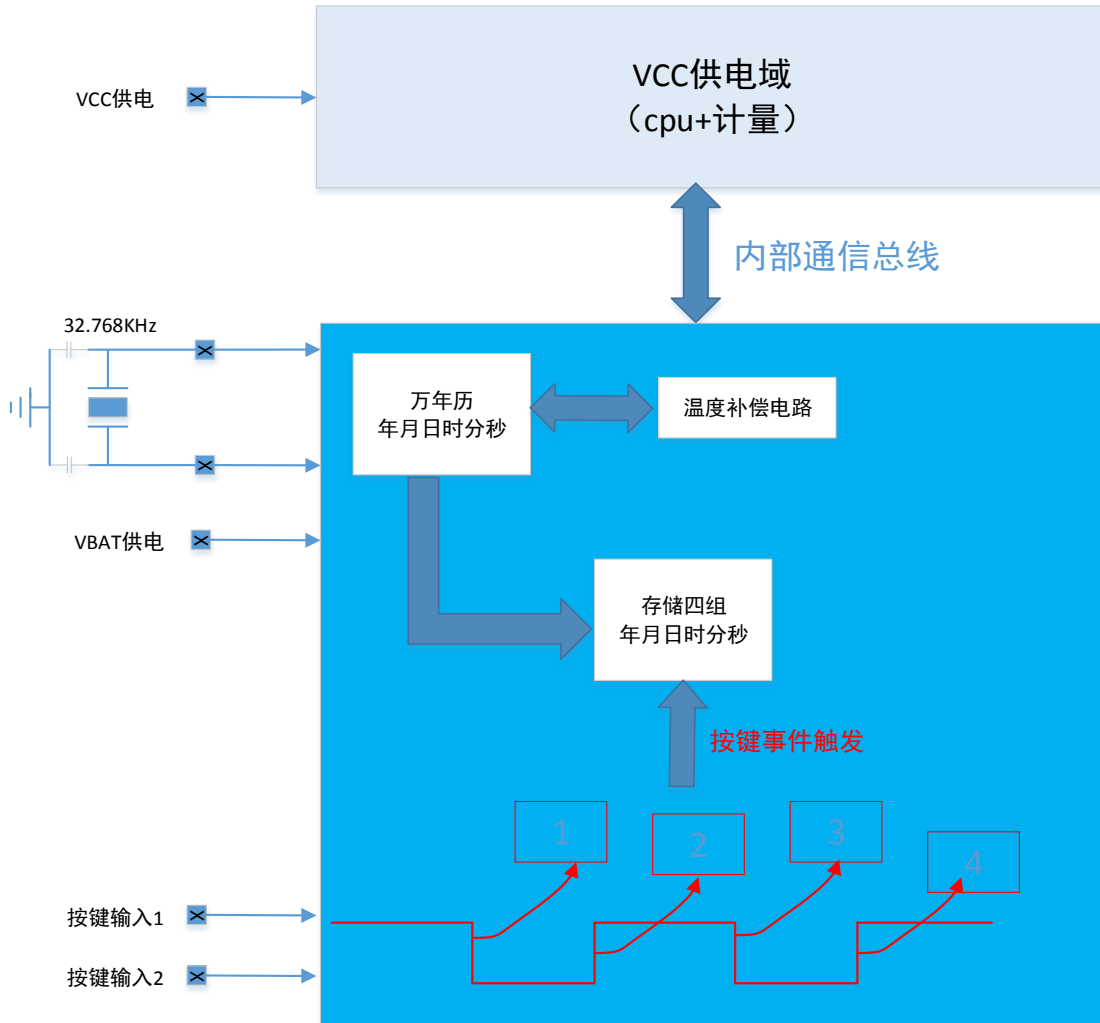
实时时钟用独立的时、分、秒寄存器跟踪时间。日历包括年、月、日以及星期寄存器，具有闰年闰月自动修正功能。时钟脉冲输出具有多种可选择频率用于时钟校准。提供闹钟/报警功能。

集成温度传感器，提供温度测量的数字结果。该模块由 VBAT 独立供电。

8.2 特点

- 提供准确的温度值， $-25^{\circ}\text{C}\sim 70^{\circ}\text{C}$ 范围内测温精度为 $\pm 1^{\circ}\text{C}$
- 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 0.061035ppm
- 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 1024/32768Hz
- 可输出校正后的频率 1/2/4/8/32/128Hz
- 新增四次曲线补偿，实际使用时请调用锐能微库函数改写补偿参数。
- 新增 I2CM 功能，不需要 CPU 参与，可自动通过 IIC 接口将按键事件发生时的时间信息记录到外部 EEPROM 中；
- 新增 KEYOUT 功能，在按键事件发生时，可触发 KEYOUT 端口，用于控制外部 EEPROM 的电源。
- VBAT 域新增两个 IO 口：P50/P51。同时 P40~P43 的 IO 口复用关系更改，具体参见 GPIO 章节和管脚定义章节。

8.3 按键事件记录功能说明



1) 供电说明

VCC 与 VBAT 独立供电，两个电源域可支持不同的电压（比如 VCC 3.3V/VBAT 5V 或者 VCC 5V/VBAT 3.3V）。VCC 掉电，不影响 VBAT 域电路独立运行，VBAT 域电路包括万年历、温补电路、按键记录电路等。

2) 开盖事件说明

步骤 1：CPU 在检测到交流电掉电前配置 VBAT 域的按键检测功能，可记录最早两次或者最晚两次。

步骤 2：在电池供电（或超级电容）情况下，当按键事件发生时（KEY 输入由高变低，或者由低变高），硬件控制器自动将当前万年历数据（年月日时分秒）备份到 VBAT 域的寄存器（类似 RAM，掉电会丢失）。

步骤 3：当 CPU 检测到上电后，CPU 从 VBAT 域读取存储好的万年历数据，记录按键事件。

3) IO 口复用

VBAT 域支持按键时间记录功能的是 KEY6/KEY7，有 P40/P41 等 IO 口可以复用为 KEY 功能。具体复用关系参见管脚定义和 GPIO 章节。

8.4 I2CM 功能说明

当发生按键事件时，RN2026 将发生事件时的时间信息（年月日时分秒）存入寄存器中（0x90~0x12C），这

类寄存器属于掉电即丢失的寄存器，考虑到某些极端情况下，VBAT 域可能遭遇人为断电，从而造成记录在寄存器中的按键时间信息丢失。为了弥补该缺陷，RN2026（B 版）新增了 I2CM 功能，即：在没有 cpu 参与的情况下，仅 VBAT 域有电时，RN2026 可通过 I2C DMA 模式将按键发生时的时间信息存入外部 EEPROM。

具体说明如下：

- 1) I2CM 模块通过 I2C 接口自动将按键事件记录的时间戳信息存入外部 EEPROM
- 2) 支持 I2C Master 模式
- 3) 支持 eeprom I2C 接口的字节写操作（连续写）和复位操作；
- 4) I2CM 运行在内部 RC 下，运行频率 $\leq 3.58\text{Mhz}$ ， I2C 时钟频率最高支持 $\leq 3.58\text{Mhz}/16$ ，分频系数为 16；
- 5) 如果没有收到应答位；放弃本次操作；发停止位；
- 6) 帧结构：先发一个停止位，然后正常的写操作控制位，之后发数据和校验和。
- 7) I2CM 写入 EEPROM 的基地址可配，建议选择为 0x00，将 EEPROM 第一页空间留出来给 I2CM 使用。**

时间戳信息在 EEPROM 中的存储结构定义如下：

目标基地址指 8 组时间戳在 EEPROM 中存在的起始地址，固定为 0x0；8 组时间戳信息在 EEPROM 中的存储结构为下图所示：

组别	名称		0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
1	P40P0	0x0	秒	分	时	日	月	年	校验和	空
2	P40N0	0x8	秒	分	时	日	月	年	校验和	空
3	P40P1	0x10	秒	分	时	日	月	年	校验和	空
4	P40N1	0x18	秒	分	时	日	月	年	校验和	空
5	P41P0	0x20	秒	分	时	日	月	年	校验和	空
6	P41N0	0x28	秒	分	时	日	月	年	校验和	空
7	P41P1	0x30	秒	分	时	日	月	年	校验和	空
8	P41N1	0x38	秒	分	时	日	月	年	校验和	空

备注：P40P0，指的是 KEY6 第一次发生上升沿触发事件；

P40N0，指的是 KEY6 第一次发生下降沿触发事件；

P40P1，指的是 KEY6 第二次发生上升沿触发事件；

P40N1，指的是 KEY6 第二次发生下降沿触发事件；

P41 口对应的是 KEY7，跟上述定义类似。

注意：KEY6/KEY7 也可以复用到其他 IO，不一定以对应 P40 或者 P41。

每组数据除了存储时间戳包含的 6 个字节信息外，还要存储时间戳的校验和结果，校验和的计算方法为：

校验和= 年字节+月字节+日字节+时字节+分字节+秒字节；

即将时间戳各个字节相加得到 8bit 数据，累加结果不需要考虑超过 8bit 的进位。

8.5 寄存器

模块名	物理地址	映射地址
RTC	0x4003C000	0x4003C000
寄存器名	地址偏移量	描述
RTC_CTL	0x00	RTC 控制寄存器

RTC_SC	0x04	秒寄存器，写保护
RTC_MN	0x08	分钟寄存器，写保护
RTC_HR	0x0C	小时寄存器，写保护
RTC_DT	0x10	日寄存器，写保护
RTC_MO	0x14	月寄存器，写保护
RTC_YR	0x18	年寄存器，写保护
RTC_DW	0x1C	星期寄存器，写保护
备注：cpu 写入万年历寄存器组，需要等待 300uS 才能读取到写入的值。		
RTC_CNT1	0x20	定时器 1 寄存器
RTC_CNT2	0x24	定时器 2 寄存器
RTC_SCA	0x28	秒闹钟寄存器
RTC_MNA	0x2C	分钟闹钟寄存器
RTC_HRA	0x30	小时闹钟寄存器
RTC_IE	0x34	RTC 中断使能寄存器
RTC_IF	0x38	RTC 状态寄存器
RTC_TEMP	0x3C	当前温度寄存器，可读可写，写保护
RTC_DOTA0	0x48	RTC 初始偏差校正寄存器
LOSC_CFG1	0x6C	32768 振荡器配置寄存器
RTC_TEMPOS	0xC4	RTC 温度 offset 校正寄存器，写保护
RTC_TPSIN	0xC8	RTC 自动温补时输入的 ADC 测量值
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。 注：RN2025C64 不支持该功能
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，格式为二进制补码； 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：T >= 75 度
RTC_IOMODE	Offset+0x8C	IO 口模式选择：

		<p>Bit0: 开盖检测(P40)模式配置 0: 记录开盖使能后最早两次按键事件; 1: 记录开盖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。 使能开盖检测功能前, 应先对该位做配置。</p> <p>Bit1: 开盖检测(P41)模式配置 0: 记录开盖使能后最早两次按键事件; 1: 记录开盖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。 使能开盖检测功能前, 应先对该位做配置。</p> <p>Bit2: 按键滤波时间基准设置 0: 默认为 2ms; 1: 设置为 250ms;</p> <p>Bit3: 滤波时间写保护 0: bit13~bit4 不可写, 并且读出为 0; 1: bit13~bit4 可写, 并且读出为实际值。</p> <p>Bit8~Bit4: 按键滤波时间设置: 代表的滤波时间为滤波时间寄存器乘以滤波时间基准。 默认值为 5'h0, 如果滤波时间基准设置为 2ms, 不论寄存器如何设置最高位实际一直固定为 1 (读出仍为设置值), 滤波时间大约为时间设置寄存器*时间基准+23ms;</p> <p>如果滤波时间基准设置为 250ms, 当按键滤波时间设置: =0-3: 滤波时间固定为 500ms</p>
--	--	---

		<p>=4~31: 滤波时间 $\approx (\text{滤波时间设置寄存器}+1) * 250\text{ms}$。</p> <p>Bit13~bit9: EEPROM 写入等待时间设置 代表的是按键事件发生后, 相隔多长时间再自动写入 EEPROM; 时间基准为 1ms, 等待时间为: $(\text{寄存器值}+1) * 1\text{ms}$。</p> <p>注意 1: 应用 P40 开盖检测功能时, 时间冻结数据可能不为 0, 建议在休眠前, 先关闭再打开 key6 的复用功能, 即把 GPIO 章节 PCB0 寄存器的 PC40 先配置为 00, 再配置为 01, 时间冻结数据就清零了。 应用 P41 开盖类似。</p> <p>注意 2: 只使用一个按键时, 必须使用 P40, 不能使用 P41, 或两个按键 P40/P41 同时使用。</p>
P40N0_TIME0	Offset+0x90	<p>KEY6 口第一次下降沿冻结的时间数据 两字节有效, 分别是(分钟, 秒)。 低 16 位有效, 两个字段各 8 位, 高 8 位是分钟, 低 8 位是秒 (下同)</p>
P40N0_TIME1	Offset+0x94	<p>KEY6 口第一次下降沿冻结的时间数据 两字节有效, 分别是(日, 小时)</p>
P40N0_TIME2	Offset+0x98	<p>KEY6 口第一次下降沿冻结的时间数据 两字节有效, 分别是(年, 月)</p>
P40P0_TIME0	Offset+0x9C	<p>KEY6 口第一次上升沿冻结的时间数据 两字节有效, 分别是(分钟, 秒)</p>
P40P0_TIME1	Offset+0xA0	<p>KEY6 口第一次上升沿冻结的时间数据 两字节有效, 分别是(日, 小时)</p>
P40P0_TIME2	Offset+0xA4	<p>KEY6 口第一次上升沿冻结的时间数据</p>

		两字节有效，分别是(年，月)
P40N1_TIME0	Offset+0xA8	KEY6 口第二次下降沿冻结的时间数据 两字节有效，分别是(分钟，秒)
P40N1_TIME1	Offset+0xAC	KEY6 口第二次下降沿冻结的时间数据 两字节有效，分别是(日，小时)
P40N1_TIME2	Offset+0xB0	KEY6 口第二次下降沿冻结的时间数据 两字节有效，分别是(年，月)
P40P1_TIME0	Offset+0xB4	KEY6 口第二次上升沿冻结的时间数据 两字节有效，分别是(分钟，秒)
P40P1_TIME1	Offset+0xB8	KEY6 口第二次上升沿冻结的时间数据 两字节有效，分别是(日，小时)
P40P1_TIME2	Offset+0xBC	KEY6 口第二次上升沿冻结的时间数据 两字节有效，分别是(年，月)
P41N0_TIME0	Offset+0x100	KEY7 口第一次下降沿冻结的时间数据 两字节有效，分别是(分钟，秒)。低 16 位有效，两个字段各 8 位，高 8 位是分钟，低 8 位是秒（下同）
P41N0_TIME1	Offset+0x104	KEY7 口第一次下降沿冻结的时间数据 两字节有效，分别是(日，小时)
P41N0_TIME2	Offset+0x108	KEY7 口第一次下降沿冻结的时间数据 两字节有效，分别是(年，月)
P41P0_TIME0	Offset+0x10C	KEY7 口第一次上升沿冻结的时间数据 两字节有效，分别是(分钟，秒)
P41P0_TIME1	Offset+0x110	KEY7 口第一次上升沿冻结的时间数据 两字节有效，分别是(日，小时)
P41P0_TIME2	Offset+0x114	KEY7 口第一次上升沿冻结的时间数据 两字节有效，分别是(年，月)
P41N1_TIME0	Offset+0x118	KEY7 口第二次下降沿冻结的时间数据 两字节有效，分别是(分钟，秒)

P41N1_TIME1	Offset+0x11C	KEY7 口第二次下降沿冻结的时间数据 两字节有效，分别是(日，小时)
P41N1_TIME2	Offset+0x120	KEY7 口第二次下降沿冻结的时间数据 两字节有效，分别是(年，月)
P41P1_TIME0	Offset+0x124	KEY7 口第二次上升沿冻结的时间数据 两字节有效，分别是(分钟，秒)
P41P1_TIME1	Offset+0x128	KEY7 口第二次上升沿冻结的时间数据 两字节有效，分别是(日，小时)
P41P1_TIME2	Offset+0x12C	KEY7 口第二次上升沿冻结的时间数据 两字节有效，分别是(年，月)
I2C 自动写 EEPROM 控制寄存器		
I2CMEEP_CTL	Offset+0x180	I2CMEEP 控制寄存器
I2CMEEP_CPASS	Offset+0x184	配置寄存器密码保护
I2CMEEP_CBYTE	Offset+0x188	控制字节配置寄存器
I2CMEEP_ABYTE	Offset+0x18C	地址字节 1 和地址字节 2 配置寄存器；

8.5.1 RTC_CTL(0x00)

RTC 控制寄存器 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	Losc_cps	保留位，只读	R	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： 0：禁止 RTC 寄存器写操作； 1：允许 RTC 寄存器写操作。 注意：该位对 RTC 寄存器组 00~1C/3C/C4 有效，对 RTC_CTL[7:0]也有效。 写万年历时间寄存器的两种方法：	R/W	0

		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。 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”。		
7:6	TSE	温度传感器允许位 00: 禁止自动温补。 01: 启动自动温补。按照 TCP 的设置进行周期性温补。 10: 启动用户温补模式 0，温度寄存器可更改，由用户填入温度值，用户每写一次温度寄存器启动一次温度补偿； 11: 启动用户温补模式 1，温度寄存器不可更改，每次写温度寄存器就启动一次温度补偿操作，温度寄存器的值由 MCU 测量得到。 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。	R/W	00
5:3	TCP	温度补偿周期： 000:2S 001:10S 默认 010:20s 011:30s 100:1 分钟 101:2 分钟 110:5 分钟 111:10 分钟	R/W	001
02:00	FOUT	000: 禁止输出 001: 1Hz 输出 010: 1/30Hz 输出 011: 32768Hz 输出 100: 16Hz 输出 101: 8Hz 输出 110: 4Hz 输出 111: 保留 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。	R/W	000

8.5.2 RTC_SC(0x04)

秒寄存器 偏移地址 = 0x04

比特位	名称	描述	读/写标志	复位值
31:07	---	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	-

8.5.3 RTC_MN(0x08)

分钟寄存器 偏移地址 = 0x08

比特位	名称	描述	读/写标志	复位值
31:07	---	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	-

8.5.4 RTC_HR(0x0C)

小时寄存器 偏移地址 = 0x0C

比特位	名称	描述	读/写标志	复位值
31:06	---	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	-

8.5.5 RTC_DT(0x10)

日期寄存器 偏移地址 = 0x10

比特位	名称	描述	读/写标志	复位值
31:06	---	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	-

8.5.6 RTC_MO(0x14)

月份寄存器 偏移地址 = 0x14

比特位	名称	描述	读/写标志	复位值
31:05	---	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	-

8.5.7 RTC_YR(0x18)

年份寄存器 偏移地址 = 0x18

比特位	名称	描述	读/写标志	复位值
31:08	---	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	-

		个位, 年份值的范围为 0~99。		
--	--	-------------------	--	--

8.5.8 RTC_DW(0x1C)

星期寄存器 偏移地址=0x1C

比特位	名称	描述	读/写标志	复位值
31:03	---	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	-

8.5.9 RTC_CNT1(0x20)

RTC 定时寄存器 1 偏移地址= 0x20

比特位	名称	描述	读/写标志	复位值
31:09	---	预留	R	0
08	CNT1PD	= 0 : 开启定时器 1 = 1 : 关闭定时器 1 注: 1 秒定时来自万年历秒中断, 关闭后重新开启时计数器会重新开始计 (刻度 1 S)	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: 该定时器在RTC校正后是准确的。	R/W	0

8.5.10 RTC_CNT2(0x24)

RTC 定时寄存器 2 偏移地址= 0x24

比特位	名称	描述	读/写标志	复位值
31:10	---	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 : 开启定时器 2 = 1 : 关闭定时器 2 注: 定时器来自内部固定的 1/256s 中断, 关闭后重新开启时计数器会重新开始计 (刻度 1/256s)	R/W	0
07:00	CNT	定时器2计数器预设值 无符号数, 计数单位为1/256s。当计数值=(CNT+1)时, 置位 RTCCNT2F标志。(最小可以每1/256秒产生一次中断, 最大可以每1秒产生一次中断) 注: 该定时器源自32768Hz晶体, 未经校正, 有一定误差。	R/W	0

8.5.11 RTC_SCA(0x28)

秒闹钟寄存器 偏移地址=0x28

比特位	名称	描述	读/写标志	复位值
31:07	---	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

8.5.12 RTC_MNA(0x2C)

分钟闹钟寄存器 偏移地址= 0x2C

比特位	名称	描述	读/写标志	复位值
31:07	---	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

8.5.13 RTC_HRA(0x30)

小时闹钟寄存器 偏移地址= 0x30

比特位	名称	描述	读/写标志	复位值
31:06	---	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。

注: A 版芯片要求在闹钟应用时, 按实际设置闹钟时间-1s 设置, 如 12:12:30 配置为 12:12:29。

B 版芯片在闹钟应用时, 不需要-1s 设置, 可配置为目标时间。

8.5.14 RTC_IE(0x34)

RTC 中断使能寄存器 偏移地址= 0x34

比特位	名称	描述	读/写标志	复位值
31:10	---	预留	R	0
9	RTC_1S_SEL	秒中断来源选择: =0: 根据系统时钟模式: hcmm 下选择 pll_1hz, 非 hcmm 下选择 RTC 秒脉冲。 =1: 选择 RTC 的秒脉冲。 注: pll_1hz 和秒寄存器更新不同步, 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。	R/W	0
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;	R/W	0

		当 RTC_IE[8:0]全部为低时，中断模块时钟才关闭；		
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	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

8.5.15 RTC_IF(0x38)

RTC 中断标志寄存器 偏移地址= 0x38

比特位	名称	描述	读/写标志	复位值
31:12	---	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

8.5.16 RTC_TEMP(0x3C)

当前温度寄存器 偏移地址=0x3C

比特位	名称	描述	读/写标志	复位值
31:10	---	预留	R	0
09:00	TEMP	当前温度值。二进制补码表示, 10 位有效位, Bit9 为符号位; 最小刻度为 0.25 度。与实际温度的换算公式为: $T = \text{TEMP}/4$ 表示范围: -128 度~+127.75 度 当 RTC_CTL.TSE=00 时: 禁止自动温补。此时 RTC_TEMP 寄存器无效, 读出的值无意义。 当 RTC_CTL.TSE=01: 按照 RTC_CTL.TCP 设置周期自动温补。此时 RTC_TEMP 寄存器显示的是该测量周期的温度值, RTC_TEMP 寄存器更新周期= TCP 设置周期。 当 RTC_CTL.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启动一次温度补偿; 当 RTC_CTL.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, 温度寄存器的值由 SOC 测量得到。	R/W	-

8.5.17 RTC_DOTA0 (0x48)

初始频率偏差校正寄存器

Offset = 0x48

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

注：只有上下电复位，会将该寄存器恢复到默认值，其它复位不会恢复。

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

8.5.18 LOSC_CFG1(0x6C)

偏移地址=0x6c

比特位	名称	描述	读/写标志	复位值
31:17	---	预留	R	0
16	EXT_LOSC_FLAG	=0: 无外灌 losc 指示 =1: 有外灌 losc 指示	R/W	0
15	LOSC_WEN	=0: LOSC_PD 不可写 =1: LOSC_PD 可写	R/W	0
14:0	LOSC_PD	LOSC 使能位： =其他：打开； =15'h75a8：关闭。 备注：同时也是外灌时钟使能信号，EXT_LOSC_FLAG =1 使能外灌时钟；注意可靠性设计。	R/W	0

备注：

1、如果使用外灌 32.768khz 时钟，需要从 XI 灌入，且需灌入方波，同时要关闭芯片的起振电路（即关闭 LOSC 使能位）。

2、关闭 LOSC 使能位，需要操作 3 次 LOSC_CFG1 寄存器，可对 LOSC_CFG1 连续写 3 次 0xf5a8；

8.5.19 RTC_TEMPOS(0xC4)

offset 校正寄存器 偏移地址= 0xC4

比特位	名称	描述	读/写	复位值
-----	----	----	-----	-----

比特位	名称	描述	标志	复位值
31:06	---	预留	R	0
05:00	TEMPOS	当 TSE 等于 01 或者 11 时，温度寄存器 TEMP 读出的值=实际计算值+TEMPOS，TEMPOS 是有符号数，用于对温度进行 offset 校正； 该寄存器为二进制补码格式，Bit5 为符号位，校正刻度为 0.25 度，要使温度值改变 t (°C)，则应填入 TEMPOS=t*4；校正范围是：-8°C~ + 7.75°C。	R/W	0

备注：只有上下电复位。

8.5.20 RTC_TPSIN(0xC8)

温度 ADC 输入测量值 偏移地址= 0xC8

比特位	名称	描述	读/写标志	复位值
31:12	---	预留	R	0
11:00	TPSIN	该寄存器存放的是 SAR ADC 输入给 RTC 的测量值。 RTC 自动温补时温度测量方式（改变 bias 电流）选择： 00:先选择 bias 电流为 2uA，测试 5 个值并求和得到 D1；再选择 bias 电流为 4uA，测试 5 个值并求和得到 D2；得到测量值 ADCO=2*D1-D2 传递给 RTC； RTC 将 ADCO 除以 5 后得到 TPSIN，然后计算温度值。 01:选择 bias 电流为 2uA，测试 5 个值并求和得到 ADCO；ADCO 传递给 RTC； RTC 将 ADCO 除以 5 后得到 TPSIN，然后计算温度值。 10:选择 bias 电流为 4uA，测试 5 个值并求和得到 ADCO；ADCO 传递给 RTC； RTC 将 ADCO 除以 5 后得到 TPSIN，然后计算温度值。 11: 同 00 备注：TPSIN 的具体值可在 RTC 模块读到。	R	-

8.5.21 I2CMEEP_CTL (0x180)

I2CMEEP 控制寄存器

Offset = 0x180

比特位	名称	描述	读/写标志	复位值
31:16	---	保留	R	0
15:8	WKEY[7:0]	该寄存器 0~7 位的写操作密码，密码是 0xB5。 软件写该寄存器 0~7 位时必须保证写入的数据 8-15 位为 0xB5,才能成功写入 0-7 位； 读 WKEY [7:0]的值永远为 0x0。	R/W	0
7:4	---	保留	R	0
3:2	---	保留	R/W	
1	STOP1_DIS	首个停止命令关闭	R/W	0

		0: 在开始写传输之前先在 I2C 总线上发送一个停止命令; 1: 在开始写传输之前在 I2C 总线上不发送一个停止命令;		
0	I2CMEEP_EN	I2CMEEP 模块使能 0: 不使能 1: 使能	R/W	0

8.5.22 I2CMEEP_CPASS (0x184)

I2CMEEP 配置寄存器密码保护寄存器

Offset = 0x184

比特位	名称	描述	读/写标志	复位值
31:16	---	保留	R	0
15:0	I2CMEEP_CPASS[15:0]	I2CMEEP 配置寄存器的密码保护寄存器; 配置寄存器指的是 I2CMEEP_CBYTE[7:0] (0x188) 和 I2CMEEP_ABYTE (0x18C); 若 I2CMEEP_CPASS[15:0]==0xAF53, 则配置寄存器才可写, 只有当 I2CMEEP_CPASS[15:0]==0x50AC, 配置寄存器才有效; 该寄存器的读值永远为 0x00 软件操作流程为: 1、写 I2CMEEP_CPASS[15:0]==0xAF53; (使能配置寄存器写操作) 2、写配置寄存器 (0x188) 3、写 I2CMEEP_CPASS[15:0]==0x50AC; (配置寄存器 0x188 以及 0x18C 有效)	R/W	0

8.5.23 I2CMEEP_CBYTE (0x188)

I2CMEEP 控制字节寄存器

Offset = 0x188

比特位	名称	描述	读/写标志	复位值
31:8	---	保留	R	0
7:0	I2CMEEP_CBYTE[7:0]	I2CMEEP 写操作的控制字节配置寄存器 对应于写操作的控制字节, 其定义为{1010,A2,A1,A0,RW_} 高四位固定为 1010, 当然也可以配置成其他值; A2/A1/A0 对应目标 EEPROM 的器件地址; 如果 EEPROM 容量为 1Mb, EEPROM 的地址位宽为 17 位, 那么 A0 对应时间戳信息存储在 EEPROM 中的目标基地址的第 17 位; RW_表示本次 I2C 操作是写还是读, 该位不可配置, 只能为写操作, RW_=0。 具体存储的定义参见 8.4 章节 I2CM 功能说明。	R/W	0xA0

8.5.24 I2CMEEP_ABYTE (0x18C)

I2CMEEP 地址字节配置寄存器

Offset = 0x18C

比特位	名称	描述	读/写标志	复位值
31:16	---	保留	R	0
15:8	I2CMEEP_A DDR1[7:0]	第二地址字节配置寄存器 表示的是时间戳信息存储的目标基地址的高 8 位；	R/W	0x00
7:0	I2CMEEP_A DDR2[7:0]	第一地址字节配置寄存器 表示的是时间戳信息存储的目标基地址的低 8 位；	R/W	0x00

8.5.25 KEYOUT 功能说明

P43 和 P51 可设置为 KEYOUT 输出，功能如下：

- 1) 一旦 P43 或者 P51 设置为 KEYOUT 输出（见 GPIO 章节），那么默认就是高阻态；
- 2) 如果 VBAT 域发生按键事件，则 KEYOUT 输出为 0，可作为 EEPROM 的 WP 信号，也可作为 EEPROM 电源控制信号。
- 3) 大约 2 秒钟后 KEYOUT 自动关闭，IO 口为高阻态；

8.6 RTC 时钟读写步骤

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

将 RTC_CTL 的第 8 位 WRTC 置为 1，打开写使能操作。

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

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

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

为增强软件可靠性，写入后都建议读出做确认。

8.7 RTC 校准步骤

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

8.8 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。

9 WDT

9.1 主要特点

- 看门狗定时器使用内部低速振荡电路；
- 看门狗定时器溢出时间设置为：16ms、32ms、128ms、512ms、1s、2s、4s、8s
- 看门狗定时器用于检测不期望出现的程序循环
- 当看门狗定时器产生复位时，复位标志寄存器的 WDT_RST 位置 1
- 当达到溢出事件的 75%时，可产生一个间隔中断

出现以下情况时产生看门狗复位：

- 如果看门狗定时器计数器溢出；
- 如果将“BB”以外的数据写入 WDT_EN；
- 如果在窗口关闭期间将数据写入 WDT_EN；
- 如果在 WDT Bitband 空间将数据写入 WDT_EN；

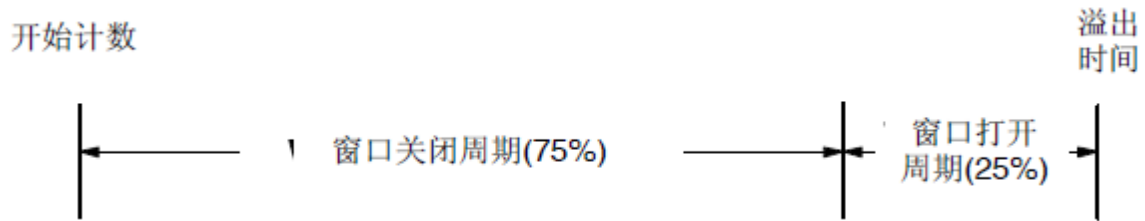
9.2 看门狗定时器的配置

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	4
CPU 睡眠设置	0: Disable（当 CPU 处于 sleep 或者 deepsleep 的时候不开启 WDT） 1: Enable（当 CPU 处于 sleep 或者 deepsleep 的时候开启 WDT）	0
CPU 调试设置	0: Disable（当 CPU 处于调试状态时不开启 WDT） 1: Enable（当 CPU 处于调试状态时开启 WDT） 注：CPU 处于调试状态指的是用户通过调试接口将 Cortex M0 停住（PC 指针停止计数）。如果芯片处于开发过程中，不建议使能该设置。因为如果使能该设置，当芯片处于调试状态时 WDT 仍然会计数，溢出时会产	0

	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：



9.3 寄存器

WDT 模块的寄存器基址

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

WDT 模块的寄存器偏移地址

寄存器名	地址偏移量	描述
WDT_EN	0x0	WDT 使能寄存器

9.3.1 看门狗使能寄存器 WDT_EN(0x00)

偏移地址 0x00

比特位	名称	描述	读/写标志	复位值
31:9	---	预留	R	0
8	WR_BUSY	WDT 寄存器写操作忙。 当 WR_BUSY=1 时，WDT_EN 不可写，只有当 WR_BUSY=0 时，才能对 WDT_EN 寄存器进行写操作。（WDT_PASS 不会 BUSY）	R	0
7:0	WDTE	将 BBH 写入 WDT_EN 可以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位 55H 或者 AAH； WDTON 位=0 时（禁止看门狗操作），WDT_EN 复位值时 55H； WDTON 位=1 时（允许看门狗操作），WDTE 复位值为 AAH。	R/W	55H 或 AAH

10 定时器

MCU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：

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

10.1 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 32 位计时功能；
 - 支持计数值动态访问；
 - 支持 32 位计数功能；
 - 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；
 - PWM 生成（单斜率或双斜率）；
 - 单脉冲输出；
 - 长度可配死区插入的互补 PWM；
 - ◆ 死区长度可编程：
 - 两个边沿的死区长度可独立设置：
 - 6 位定时长度；
 - 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;

10.2 功能框图

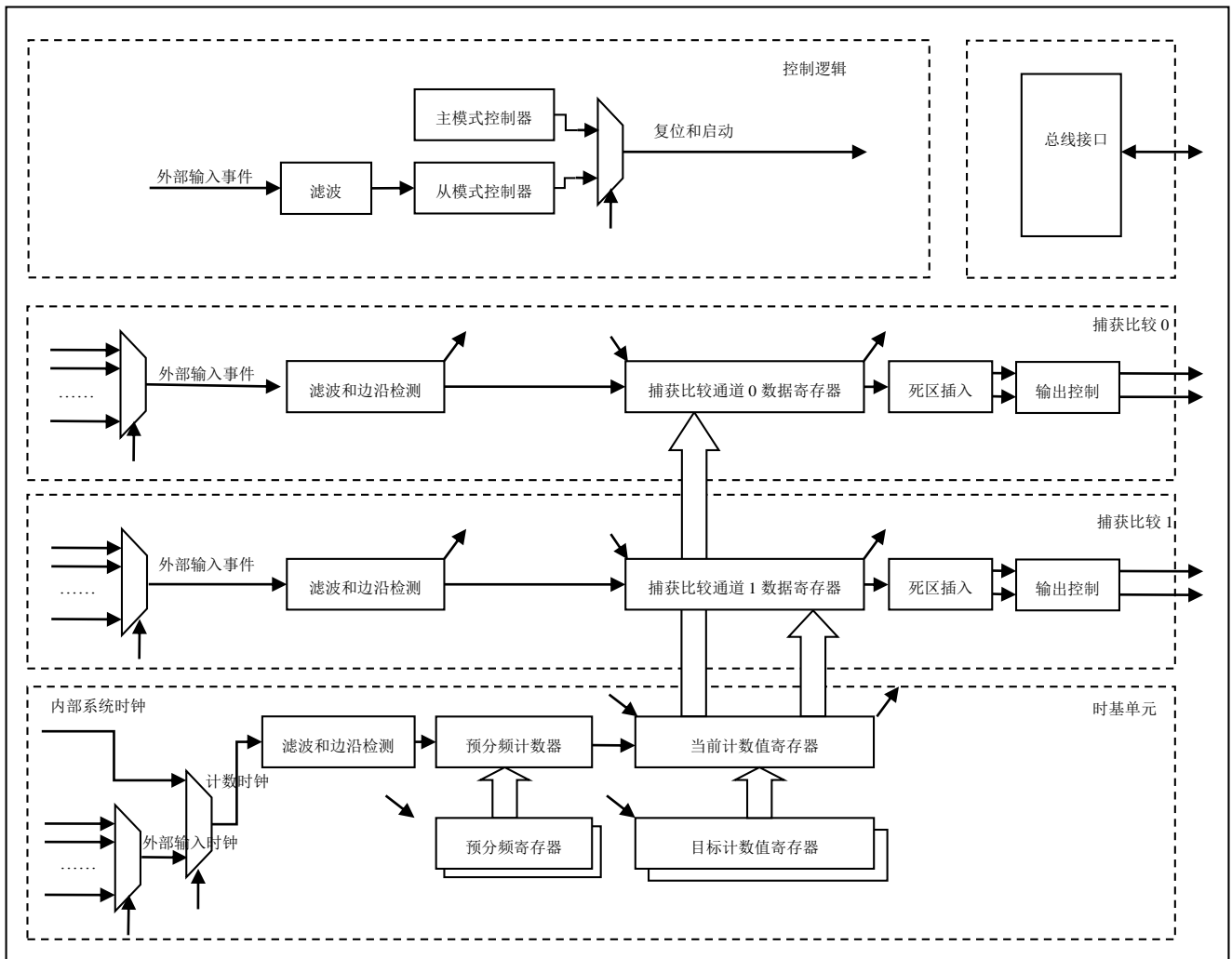


图 11-1 定时器功能框图

10.3 寄存器

TC 模块寄存器基地址

模块名	物理地址	映射地址
-----	------	------

TC0	0x40010000	0x40010000
TC1	0x40014000	0x40014000

TC 模块寄存器偏移地址

地址偏移	寄存器名称	属性	复位值	功能
0x00000000	TC_CNT	只读	0x00000000	当前计数值指示
0x00000004	TC_PS	读/写	0x00000000	预分频寄存器
0x00000008	-	-	-	保留
0x0000000C	TC_DN	读/写	0x00000000	目标计数值寄存器
0x00000010	-	-	-	保留
0x00000014	TC_CCD0	读/写	0x00000000	捕获比较通道 0 数据寄存器
0x00000018	TC_CCD1	读/写	0x00000000	捕获比较通道 1 数据寄存器
0x0000001C	TC_CCFG	读/写	0x00000000	时钟配置寄存器
0x00000020	TC_CR	读/写	0x00000000	控制寄存器
0x00000024	TC_CM0	读/写	0x00000000	捕获比较通道 0 模式寄存器
0x00000028	TC_CM1	读/写	0x00000000	捕获比较通道 1 模式寄存器
0x0000002C	TC_IE	读/写	0x00000000	中断使能寄存器
0x00000030	TC_STA	读/写	0x00000000	状态寄存器

10.3.1 当前计数值寄存器 TC_CNT(0x00)

偏移地址 0x00

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

10.3.2 预分频寄存器 TC_PS(0x04)

偏移地址 0x04

比特位	名称	功能	读/写标识	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

10.3.3 目标计数值寄存器 TC_DN(0x0C)

偏移地址 0x0c

比特位	名称	功能	读/写标识	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

10.3.4 捕获比较通道 0 数据寄存器 TC_CCD0(0x14)

偏移地址 0x14

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

注意，在通道 0 配置为捕获功能（即 TC_CM0 的 CCM 为 0）时，TC_CCD0 寄存器不可写。

10.3.5 捕获比较通道 1 数据寄存器 TC_CCD1(0x18)

偏移地址 0x18

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

注意，在通道 1 配置为捕获功能（即 TC_CM1 的 CCM 为 0）时，TC_CCD1 寄存器不可写。

10.3.6 时钟配置寄存器 TC_CCFG(0x1C)

偏移地址 0x1c

比特位	名称	功能	读/写标识	复位值
31:24	-	保留	R	0
23:16	FLTOPT	滤波参数	R/W	0
15	-	保留	R	0
14:13	ECLKMODE	外部时钟模式： 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	时钟源选择： 00000: UART0 RXD 00001: UART1 RXD 00010: UART2 RXD 00011: UART3 RXD 00100: TC1/TC0的outn[0] 00101: TC1/TC0的outp[0] 00110: TC1/TC0的outn[1] 00111: TC1/TC0的outp[1] 01000: UART4 RXD 01001: UART5 RXD 01010: 保留 01011: 保留 01100: 保留 01101: 保留 01110: cf4_out 01111: cf3_out 10000: cf2_out 10001: cf1_out 10010: cf0_out	R/W	0

		10011: rtc_out 10100: p0[1] (TCI,下同) 10101: p0[3] 10110: p1[2] 10111: p1[3] 11000: p1[4] 11001: p1[5] 11010: p2[4] 11011: p2[5] 11100: p2[6] 11101: p2[7] 11110: p3[0] 11111: p3[6]		
7:2	-	保留	R	0
1	FLTEN	滤波器使能 0: 禁止 1: 使能	R/W	0
0	CM	时钟模式: 0: 内部系统时钟 1: 外部输入时钟/事件	R/W	0

10.3.7 控制寄存器 TC_CR(0x20)

偏移地址 0x20

比特位	名称	功能	读/写标识	复位值
31:29	-	保留	R	0
28	DBGSTBDIS	调试暂停使能: 0: 使能 1: 禁止	R/W	0
27:21	-	保留	R	0
20	SLVGATELVL	从模式门控电平: 0: 低电平计数 1: 高电平计数	R/W	0
19:12	SLVFLTOPT	从模式输入滤波产生	R/W	0
11:10	SLVTRGMODE	从模式控制极性选择: 00: 上升沿 01: 下降沿 10: 双边沿 11: 门控模式	R/W	0
9:5	SLVCHANSEL	从模式输入选择: 同时钟寄存器TC_CCFG 9~5位定义	R/W	0
4	OPS	单脉冲模式选择: 0: 计数溢出后不停止;	R/W	0

		1: 计数溢出后停止		
3	SLVFLTEN	从模式输入滤波使能: 0: 禁止 1: 使能	R/W	0
2	SLVEN	从模式使能: 0: 禁止 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

10.3.8 捕获比较通道 0 模式寄存器 TC_CM0(0x24)

偏移地址 0x24

比特位	名称	功能	读/写标识	复位值
31:30	-	保留	R	0
29	DFTLVL	缺省电平: 0: 低电平 1: 高电平	R/W	0
28	EFELVL	有效电平: 0: 低电平 1: 高电平	R/W	0
27:25	OM	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	捕获通道选择: 同时钟寄存器TC_CCFG 9~5位定义	R/W	0
19	FLTEN	滤波器使能: 0: 禁止 1: 使能	R/W	0
18:11	FLT OPT	滤波器参数	R/W	0
10:9	CPOL	捕获极性选择: 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留	R/W	0

8:3	DL	死区长度	R/W	0
2	DIEN	死区插入使能: 0: 禁止 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: 禁止 1: 使能	R/W	0

10.3.9 捕获比较通道 1 模式寄存器 TC_CM1(0x28)

偏移地址 0x28

比特位	名称	功能	读/写标识	复位值
31:30	-	保留	R	0
29	DFTLVL	缺省电平: 0: 低电平 1: 高电平	R/W	0
28	EFELVL	有效电平: 0: 低电平 1: 高电平	R/W	0
27:25	OM	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	捕获通道选择: 同时钟寄存器TC_CCFG 9~5位定义	R/W	0
19	FLTEN	滤波器使能: 0: 禁止 1: 使能	R/W	0
18:11	FLTOPT	滤波器参数	R/W	0
10:9	CPOL	捕获极性选择: 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留	R/W	0
8:3	DL	死区长度	R/W	0

2	DIEN	死区插入使能： 0: 禁止 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: 禁止 1: 使能	R/W	0

10.3.10 中断使能寄存器 TC_IE(0x2C)

偏移地址 0x2c

比特位	名称	功能	读/写标识	复位值
31:4	-	保留	R	0
3	SLVIE	从模式中断使能： 0: 禁止 1: 使能	R/W	0
2	CC1IE	捕获比较通道1中断使能： 0: 禁止 1: 使能	R/W	0
1	CC0IE	捕获比较通道0中断使能： 0: 禁止 1: 使能	R/W	0
0	OVIE	溢出中断使能： 0: 禁止 1: 使能	R/W	0

10.3.11 状态寄存器 TC_STA(0x30)

偏移地址 0x30

比特位	名称	功能	读/写标识	复位值
31:4	-	保留	R	0
3	SLVIF	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	CC1IF	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	CC0IF	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	OVIF	溢出中断标志：（写1清0） 0: 无溢出事件	R/W	0

		1: 有溢出事件		
--	--	----------	--	--

10.4 典型应用

10.4.1 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。

10.4.2 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。

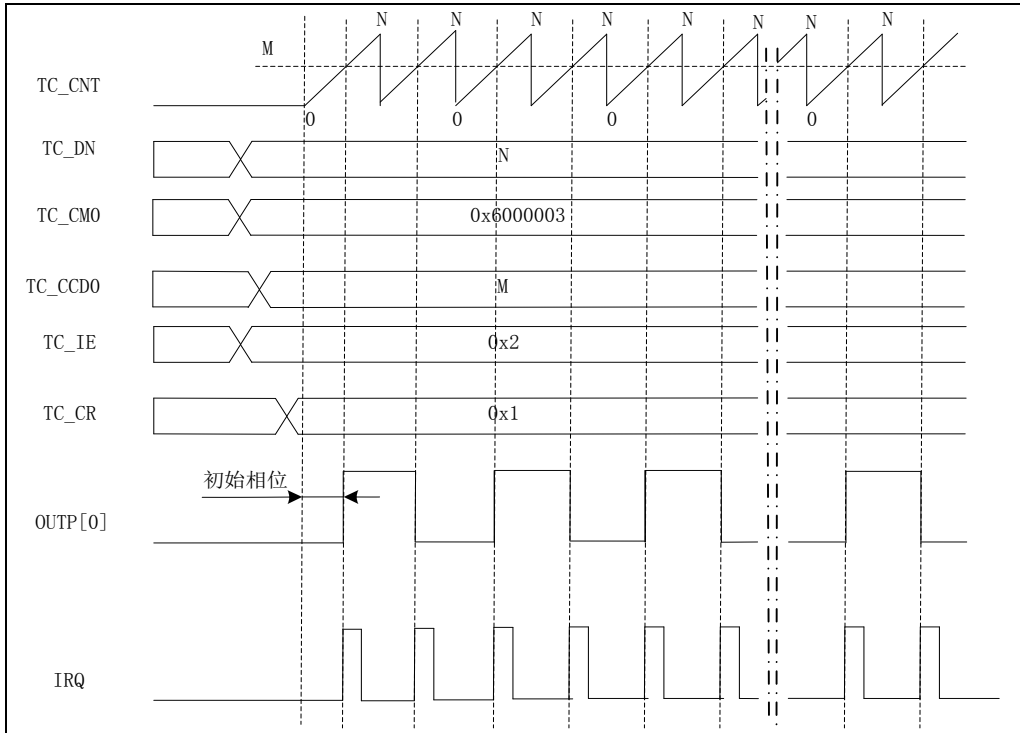
定时器捕获到外部输入事件的捕获极性时，产生中断，同时当前计数值会保存在捕获比较通道数据寄存器中。若采用两个通道，一个通道捕获上升沿，一个通道捕获下降沿，则可通过两个通道的数据寄存器计算出脉冲宽度。

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 FLT OPT 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。

10.4.3 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 FLT0PT 设置滤波周期数。

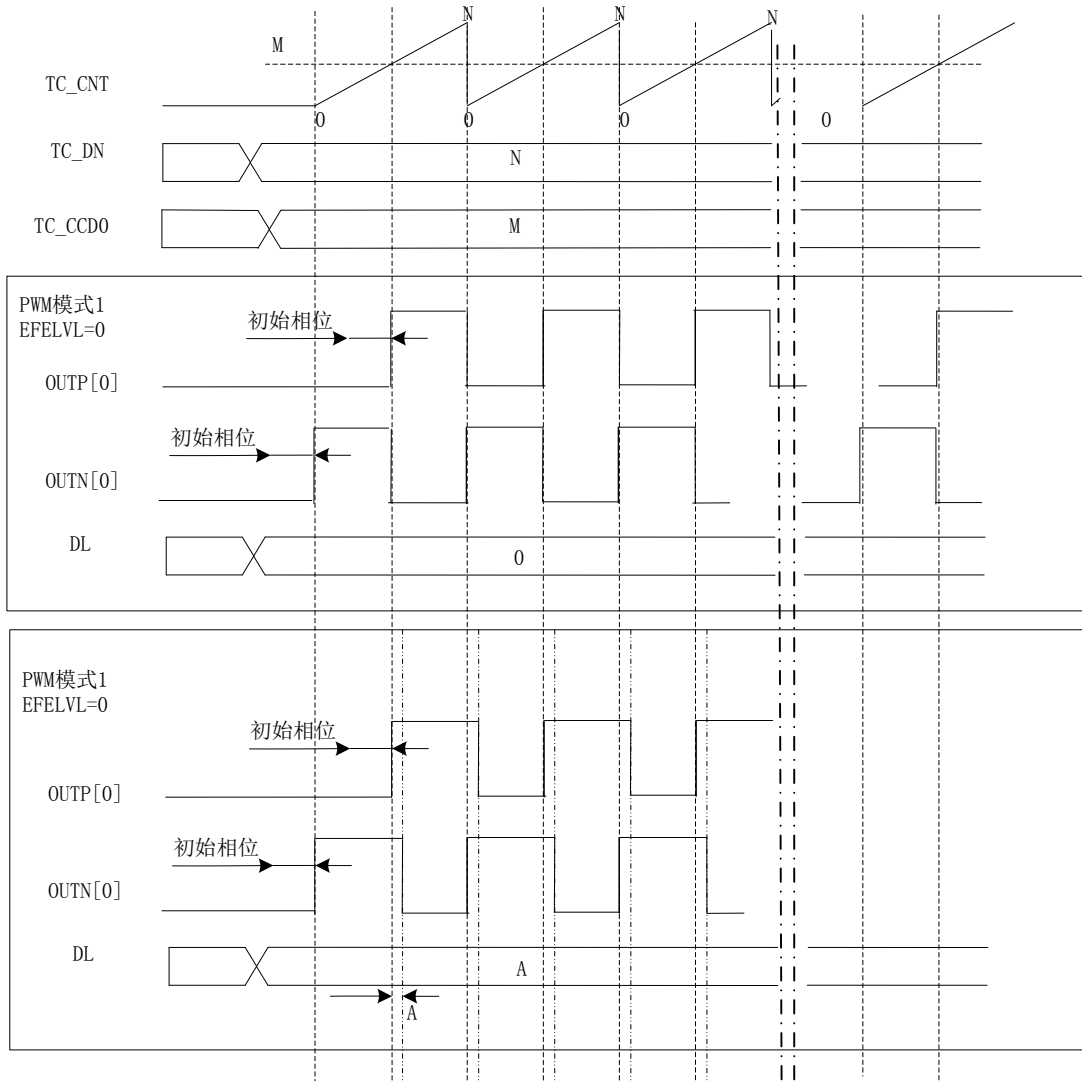
10.4.4 比较输出模式，PWM 输出功能

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。

10.4.5 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：

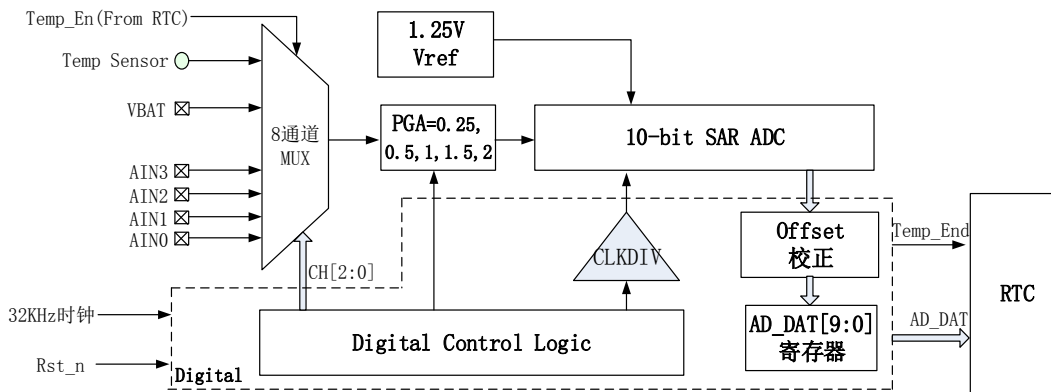
- 1、可修改预分频寄存器值，改变定时器计数时钟频率。
- 2、可配置时钟配置寄存器，CM 修改配置为外部输入时钟，同时修改 CS 位配置的外部输入时钟源。外部输入时钟源频率不可比内部系统时钟频率的二分频高。
- 3、单次计数模式，定时器溢出后就停止；配置控制寄存器的 OPS 位为 1 即为单次计数模式。
- 4、外部输入时钟源为另外一个定时器的输出，则可连接为两个定时器的级联模式，可增加定时器的寄存器的位宽。
- 5、从模式控制模式选择中，若选择为门控模式，SLVGATELVL 从模式门控有效电平配置，从模式的输入在门控有效电平时内部计数器才会计数。
- 6、从模式下配置为输入捕获方式，从模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

11 模拟外设

11.1 特点

多路复用 SAR-ADC

- 10-bit 逐次逼近型 ADC，完成一次转换需要 16 个周期，采样时钟为 16KHz；
- 8 通道模拟多路选择器，其中 AIN0 – AIN5 通道用于外部模拟信号测量，VBAT 用于电池电压产量，Temp Sensor 用于测量温度传感器的输出（理论误差小于 0.125 度）。
- **新增 AIN4、AIN5 两个 SAR-ADC 测量通道。其中 A 版 AIN4 通道是 sigma-delta ADC，B 版更改为 SAR-ADC。**
- 1 倍增益时，最大支持 1.25V 输入；0.5 倍增益时，最大支持 2.5V 输入，以此类推。增益倍数支持 0.5 倍、1 倍、1.5 倍、2 倍，**新增 0.25 倍 PGA。**
- ADC 不采样时，自动进入省电模式



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；**建议实际应用时测量芯片电源，见 2.2 掉电处理章节**
- 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，默认配置下 V_{iL} 阈值约为 1.28V 左右， V_{iH} 阈值约为 1.5V 左右。
- LVD 硬件滤波时间为 100 个系统时钟，建议软件在中断唤醒后，对 LVD 的状态再做确认和滤波

一路比较器电路 CMP0，主要特点如下：

- 对外部 PIN 输入电压做比较，默认配置下 V_{iL} 阈值约为 1.28V 左右， V_{iH} 阈值约为 1.5V 左右。通过更改配置，也可将阈值设置为 0.9V 左右。
- 功耗小于 1uA
- **建议实际应用时测量 LDO 前端电压，见 2.2 掉电处理章节**
- 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

11.2 寄存器

模块名	物理地址	映射地址
ANA	0x4002C000	0x4002C000
寄存器名	地址偏移量	描述
SAR_CTL	0x0	SAR-ADC 控制寄存器

SAR_START	0x4	SAR-ADC 启动寄存器
SAR_STAT	0x8	SAR-ADC 状态寄存器
SAR_DAT	0xC	SAR-ADC 数据寄存器
LVD_CTL	0x10	比较器控制寄存器
LVD_STAT	0x14	比较器状态寄存器
SAR_CTL1	0x18	SAR-ADC 控制寄存器 1

11.2.1 SAR_CTL(0x00)

ADC 控制寄存器

偏移地址= 0x0

比特位	名称	描述	读/写标志	复位值
31:17	---	预留	R	0
16:12	REF_WAIT	从开启 REF 到开启 ADC 需要等待的时间： 等待时间的最小刻度是 122 μs 。 等待时间具体值为： $(REF_WAIT+1)* 122 \mu s$ 5'd0: 122 μs 5'd31: $(31+1)* 122 \mu s$ 建议客户应用程序将此寄存器配置为 0x10, 即 2074 μs	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 μs 5'd31: $(31+1)* 30.5 \mu s$ 即等待时间= $(SAR_WAIT+1)* 30.5 \mu s$ 建议客户软件使用默认值 0xE 注：启动 ADC 测量的步骤： 开启 REF, 等待 REF_WAIT 时间； 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。 软件通过标志位判断 ADC 是否已经测量完成, 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。详见例程。	R/W	0xE
6	SAR_CH3	SAR-ADC Channel 选择, 与 SAR_CH[2:0]共同组成一个 4bit 寄存器。 {SAR_CH3,SAR_CH[2:0]}=0xxx: 见 SAR_CH 定义 {SAR_CH3,SAR_CH[2:0]}=1000: 选择 AIN3, P40/AIN3 复用, 注意 AIN3 的特殊性, 见管脚定义章节。 {SAR_CH3,SAR_CH[2:0]}=其他: 保留	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	R/W	0

		01: 1 倍 10: 1.5 倍 11: 2 倍 SAR-ADC 也支持 0.25 倍 PGA，如果要使用 0.25 倍 PGA，请配置 SAR_CTL1 寄存器（0x18）		
2:0	SAR_CH	SAR-ADC Channel 选择 000: Temp Sensor 001: VBAT（2/3 分压后得到 2.4V，PGA 采用 0.5 倍，测量输入 1.2V） 010: AIN0, P43/AIN0 复用口 011: AIN1, P42/AIN1 复用口 100: AIN5, P51/AIN5 复用口 101: AIN4, P50/AIN4 复用口 110: 无效 111: AIN2, P41/AIN2 复用口 Note: 不管设置为哪个通道，自动温度测量的优先级最高。	R/W	0
备注：以上寄存器位只有当 SAR_START 中的 ST=0 时才可写。				

11.2.2 SAR_START(0x04)

SAR-ADC 启动寄存器

偏移地址= 0x4

比特位	名称	描述	读/写标志	复位值
31:01	---	预留	R	0
0	ST	SAR-ADC Start Bit 0: SAR-ADC 无操作 1: 启动一次 SAR-ADC 采样，完成采样后自动清零 注 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； 注 2: 当该位为 1 时，禁止软件再次写 1 启动 ADC 测量，软件需要查询该标志位，当该标志位为 0 时才能启动一次新的 ADC 测量操作。建议的超时等待时间=3*(REF_WAIT+SAR_WAIT)。	R/W	0

11.2.3 SAR_STATUS(0x08)

SAR-ADC 状态寄存器

偏移地址= 0x8

比特位	名称	描述	读/写标志	复位值
31:02	---	预留	R	0
1	TPS_BUSY	自动温度测量忙，=1: 自动温度测量正在进行； =0: 自动温度测量没有进行。	R	0

		当 TPS_BUSY 为=1 时，ADC_START 寄存器不可写。		
0	DREADY	ADC Data Ready Pending Bit 0: ADC 转换结果未完成 1: ADC 转换结果已完成 Note: 写 1 清零; 由 RTC 控制的自动温度测量不在该状态中指示;	R/W	0

11.2.4 SAR_DAT(0x0C)

ADC DAT 寄存器

偏移地址= 0xC

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

11.2.5 LVD_CTL(0x10)

LVD 控制寄存器

偏移地址= 0x10

比特位	名称	描述	读/写标志	复位值
31:7	---	预留	R	0
6	CMPOIE	比较器 0 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	---	保留, 可写, 用户不要配置	R/W	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 4.0 1010 4.2 1011 4.4 1100 4.6 1101 4.7 1110 检测外部管脚 LVDIN 的电压, 与阈值 1.28V 进行比较; 1111 保留;	R/W	0

备注: LVD、比较器 0 合并一个中断向量; SAR 单独一个中断向量。

11.2.6 LVD_STAT(0x14)

LVD 状态寄存器

偏移地址= 0x14

比特位	名称	描述	读/写标志	复位值
31:6	---	预留	R	0
5	CMP0IIF	比较器 0 中断标志 =0: 未产生中断; =1: 产生中断; 当输入电压相对于阈值变低或者变高时产生中断, 写 1 清零;	R/W	0
4	LVDIIF	LVD 中断标志 =0: 未产生中断; =1: 产生中断; 当输入电压相对于阈值变低或者变高时产生中断, 写 1 清零;	R/W	0
3:2	---	预留	R	0
1	CMP0IF	比较器 0 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

11.2.7 SAR_CTL1 0x18

ADC 控制寄存器 1

Offset = 0x18

比特位	名称	描述	读/写标志	复位值
31:17	---	预留	R	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 100: 0.25 倍 其他: 保留 该寄存器也用于控制 SAR-ADC 的增益 (PGA), 它与 SAR_CTL[4:3]. SAR_PGA [1:0]的关系见 SAR_PGA_SEL。 如果要使用 0.25 倍的 PGA, 才需要用到该寄存器	R/W	0x0

11.3 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 10 位为 1, 开启 SAR_EN 时钟。
- 2、配置相关 IO 口的复用配置, 选择为 AIN 功能;

- 3、判断 SAR-ADC 状态寄存器 SAR_STATUS 第 1 位是否为 0，没有进行自动温度测量，如为 0 进入下一步，如为 1 则等待。
- 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 第 0 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 $(ADC\ DAT * 1.25) / 1024 / PGA$ 。当电压值超过 ADC 的满度，转换值为 1024。
- 9、转换及计算完成。

11.4 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 10 位为 1，开启 SAR_EN 时钟。
- 2、判断 SAR-ADC 状态寄存器 SAR_STATUS 第 1 位是否为 0，没有进行自动温度测量，如为 0 进入下一步，如为 1 则等待。
- 3、配置 ADC 控制寄存器 SAR_CTL 通道选择为 VBAT，配置等待时间；
- 4、SAR-ADC 启动寄存器 SAR_START 配置为 1，启动 ADC 转换。
- 5、判断 SAR-ADC 状态寄存器 SAR_STATUS 第 0 位，等待转换完成。
- 6、读取 ADC 转换数据寄存器 SAR_DAT。
- 7、计算：ADC 基准电压源为 1.25V，ADC 满度时 ADC DAT 寄存器值为 1024。3.6V 电池直接接入 VBAT 引脚，MADC->AD_CTRL 配置为 0x01，此时增益为 0.5 倍，则电压值为 $(ADC\ DAT * 1.25 * 4) / 1024$ ，其中因增益为 0.5 倍，内部对 VBAT 接入有 1/2 分压，所以实际电压需乘 4。
- 8、转换及计算完成。

11.5 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，CMP0 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 1.28V 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 10 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 4 位置 0，开启 CMP0 电源。
- 4、配置 LVD 控制寄存器 LVD_CTL，使能 CMP0 中断。开启 CMP0 中断 NVIC_EnableIRQ(LVDCMP_IRQn)；
- 5、编写中断服务程序：

```
void LVDCMP_HANDLER(void)
{
    if(!(MADC->LVD_STAT & 0x20))
    {
    }
    else
    {
    }
    MADC->LVD_STAT = 0x01ff;
}
```

因 LVD，CMP0 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。

CMP0 硬件滤波时间为 100 个系统时钟，建议软件在中断唤醒后，对 CMP0 的状态再做确认和滤波。

12 GPIO

12.1 概述

- RN2025 包含 PA、PB，共 37 个 GPIO 口，新增 3 个 GPIO 口：
 - PA 口包含 33 个 DVCC 域 IO 口：5 个 P0 口、8 个 P1 口、8 个 P2 口、5 个 P3 口、1 个 P6 口；
 - PB0 口包含 6 个 VBAT 域 IO 口：P40~P43、P50、P51；
 - PB1 口包含 4 个 AVCC2 域 IO 口：P44~P47；
 - 新增的 3 个 IO 口是：P60、P50、P51
- RN2025 GPIO 章节定义与 RN2026 一致，部分未定义的 IO 口也有相关寄存器，实际使用是保持其为默认值即可。
- PA 口、PB1 口 GPIO 在 AHB 下，由 GPIO-AHB 寄存器组控制，支持高速 IO 翻转，支持 bitband。
- PB0 口 GPIO 在 APB 下，由 GPIO-APB 寄存器组控制，不支持高速 IO 翻转，不支持 bitband。
- P0、P2、P3 口、P10~P15、P60~P62 均支持上拉可选、漏极开漏可选、TTL/CMOS 电平可选。
- P16~P17 是与高频晶振复用的引脚，A 版只有输入功能，B 版/C 版可支持输入输出。
- SPI0 支持负荷识别专用 DMA；SPI1/SPI2 支持普通 DMA；UART2 支持普通 DMA；详见各章节描述
- 新增两路 SPI 口，分别是 SPI3 和 SPI4，支持普通 DMA 模式。
- 新增 7 个 Sigma-delta GPADC 输入端口。
- 新增 RTCOUT 和 CF 输出复用关系；
- RTCOUT 与 RTC_1S 的区别：RTCOUT 由 FOUT (rtc rtc_ctl) 定义，可输出每秒经校正准确的 1Hz 信号，但是与万年历时间更新不一定做到秒同步；RTC_1S 可输出与万年历时间每秒同步更新的脉冲。
- P16、P17、P22、P23 为一组，固定选择为 SPI3。
- P24、P25、P26、P27 为一组，固定选择为 SPI4。
- P06、P07、P10、P11 为一组，可选择为 SPI0/SPI1/SPI2，其中 SPI0/SPI1 为主从模式，SPI2 为专用从模式。
- P12、P13、P14、P15 为一组，可选择为 SPI0/SPI1/SPI2，其中 SPI0/SPI1 为主从模式，SPI2 为专用从模式。
- P31、P32、P33、P34 为一组，可选择为 SPI0/SPI1/SPI2，其中 SPI0/SPI1 为主从模式，SPI2 为专用从模式
- P44 口新增支持脉冲计数器，用于支持脉冲式数字温度传感器。

12.2 功能描述

包含 43 个 IO 口，全部有复用功能。

功能名称	I/O 类型	功能	复位后	复用功能
P00	PABULD6	上拉电阻可选、施密特、TTL/CMOS 输入可选、漏极开路可选、6mA 驱动能力；	数字输入	CMP0/INT0/CF2
P01	PABULD6		数字输入	保留
P02	PABULD6		数字输入	CF0/RTCOUT/SDAIN5
P03	PABULD6		数字输入	CF1/TCIO/RTCOUT/RTC_1S
P04	PABULD6		数字输入	保留
P05	PABULD6		数字输入	保留
P06	PABULD6		数字输入	INT1/SCSN0_1_2
P07	PABULD6		数字输入	KEY1/SCLK0_1_2
P10	PABULD6		数字输入	INT2/MOSI0_1/SDI2
P11	PABULD6		数字输入	KEY2/MISO0_1/SDO2
P12	PABULD6		数字输入	SCSN0_1_2/TCIO
P13	PABULD6		数字输入	SCLK0_1_2/TCIO

P14	PABULD6		数字输入	MOSI0_1_2/SDI2/TCIO
P15	PABULD6		数字输入	MISO0_1_2/SDO2/TCIO
P16	PUXI	高频晶振与 IO 复用引脚	数字输入	HOSCO/INT3/SCSN3
P17	PUXI		数字输入	HOSCI/INT4/SCLK3
P20	PABULD6	上拉电阻可 选、施密 特、 TTL/CMOS 输入可选、 漏极开路可 选、6mA 驱 动能力；	数字输入	SWDCLK/P20/RX1/CF3 /RTC_1S/TC0_P[1]/RX2
P21	PABULD6		数字输入	SWDIO/P21/TX1/TX2/ CF2/RTC_OUT/TC0_P[0]
P22	PABULD6		数字输入	RX2/KEY4/MOSI3/TC1_N[0]/CF4
P23	PABULD6		数字输入	TX2/KEY5/MISO3/TC1_P[0]/RTC_O UT
P24	PABULD6		数字输入	SCL/TCIO/SCSN4/CF4
P25	PABULD6		数字输入	SDA/TCIO/SCLK4/ RTC_OUT
P26	PABULD6		数字输入	RX3/TCIO/CF2/RTC_OUT/MOSI4
P27	PABULD6		数字输入	TX3/TCIO/CF3/RTC_1S/MISO4
P30	PABULD6		数字输入	ISPEN/TCI/KEY3/ CF4/TC1_P[1]/RTC_OUT
P31	PABULD6		数字输入	SDO2/MISO1_0/SCL/CF4/RTC_OUT
P32	PABULD6		数字输入	SDI2/MOSI1_0/SDA/CF3/RTC_OUT/ 1S
P33	PABULD6		数字输入	SCLK2/SCLK1/SCLK0/ /RX2/CF2/RTC_OUT/1S
P34	PABULD6		数字输入	SCSN2/SCSN1/SCSN0/ /TX2/CF1/RTC_OUT/1S
P35	PABULD6		数字输入	保留
P36	PABULD6		数字输入	保留
P37	PABULD6		数字输入	保留
P40	PABULD6		数字输入	KEY6/AIN3
P41	PABULD6		数字输入	KEY7/AIN2
P42	PABULD6		数字输入	RTCOUT/AIN1/SCL/KEY6/ RTC_1S
P43	PABULD6		数字输入	AIN0/SDA/KEYOUT
P44	PABULD6		数字输入	SDAIN0/TX4/INT6
P45	PABULD6		数字输入	SDAIN1/RX4/INT7
P46	PABULD6		数字输入	SDAIN2/TX5/KEY6
P47	PABULD6		数字输入	SDAIN3/RX5/KEY7/CF4/CF0 /RTCOUT
P50	PABULD6	数字输入	AIN4/KEY6/RTC_OUT	
P51	PABULD6	数字输入	AIN5/KEYOUT/RTC_1S	
P60	PABULD6	数字输入	SDAIN8/CF0/RTC_OUT	

P62	PABULD6		数字输入	保留
P63	PABULD6		数字输入	保留

12.3 应用注意事项

- VCC 掉电 VBAT 有电场景：VBAT 域 6 个 IO (P40~P43、P50、P51) 配置保持，CPU 在掉电前应根据漏电需求配置好 IO，再上电后 CPU 根据应用需求软件恢复 IO 配置。
- VBAT 域 6 个 IO 的电源是 VBAT，其输出高电平等于 VBAT，输入高电平不能高于 VBAT，并满足 ViH 电平要求 (0.7*VBAT)。
- VBAT 域 6 个 IO 在 APB 下，不支持高速 IO，GPIO-APB 寄存器不支持 bitband 功能。
- IO 复用除 SWD 外，其他引脚上电默认均为 GPIO，SWD 上电默认上拉有效，GPIO 默认高阻态。
- 发生复位后，BOOTROM 会检测 P30 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。P30 口作为 ISP 使能时，BOOTROM 会选择上拉，防止上电时输入状态不确定。
- 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。
- P16/P17 复用为 HOSC 时，只要任何一个引脚选择为 HOSC，就选择为 HOSC。
- 当 IO 口设置为数字功能时，方向寄存器不起作用，由数字模块本身控制。当选择为模拟口时，方向寄存器无效，为模拟口。
- 当 IO 口处于输出模式或者模拟 PIN 模式时，不管上拉寄存器如何配置，上拉均不使能。
- 为提高 IO 翻转速率，数据配置相关寄存器应在 AHB 下。支持 bitband 功能对 IO 口的相关寄存器进行 bit 操作。也可采用 SET/CLR 寄存器写 GPIO 的数据寄存器。

12.4 寄存器

12.4.1 寄存器列表

GPIO-AHB 寄存器组

PA 口、PB1 口 GPIO 在 AHB 下，由 GPIO-AHB 寄存器组控制，支持高速 IO 翻转，支持 bitband

基地址	0x50000000				
偏移地址	名称	R/W	有效字长	复位值	功能描述
0x00	PMA	R/W	4	0xFFFFFFFF	PA 口模式寄存器
0x04	PCA0	R/W	4	0x0	PA 口复用寄存器 0
0x08	PCA1	R/W	4	0x2	PA 口复用寄存器 1
0x0C	PUA	R/W	4	0x00030000	PA 口上拉选择寄存器
0x10	PIMA0	R/W	4	0x0	PA 口内部模式配置 0
0x14	PIMA1	R/W	4	0x0	PA 口内部模式配置 1
0x18	PIEA	R/W	4	0xFFFFFFFF	PA 口输入使能选择
0x1C	PA	R/W	4	0x0	PA 口数据寄存器
0x20	PASET	R/W	4	0x0	PA 口数据置位寄存器
0x24	PACLK	R/W	4	0x0	PA 口数据清零寄存器
0x28	PCA2	R/W	4	0x0	PA 口复用寄存器 2
0x30	PMB1	R/W	4	0x000000F0	PB1 口模式寄存器

0x34	PCB1	R/W	4	0x0	PB1 口复用寄存器
0x38	PUB1	R/W	4	0x0	PUB1 口上拉选择寄存器
0x3C	PIMB1	R/W	4	0x0	PB1 口内部模式配置
0x40	PIEB1	R/W	4	0x000000F0	PB1 口输入使能选择
0x44	PB1	R/W	4	0x0	PB1 口数据寄存器
0x48	PB1SET	R/W	4	0x0	PB1 口数据置位寄存器
0X4C	PB1CLR	R/W	4	0x0	PB1 口数据清零寄存器
0x50	GPIO_PSW0	R/W	4	0x0	GPIO_AHB 写保护寄存器
0x54	IOCNT	R/W	4	0x00	脉冲计数寄存器

GPIO-APB

PB0 口 GPIO 在 APB 下，由 GPIO-APB 寄存器组控制，不支持高速 IO 翻转

基地址	0x4002C000（与 VBAT 域模拟外设共用基地址）				
偏移地址	名称	R/W	有效字长	复位值	功能描述
0x100	PMB0	R/W	1	0x0F	PB0 口模式寄存器（输入或者输出）
0x104	PCB0	R/W	1	0x0	PB0 口复用寄存器
0x108	PUB0	R/W	1	0x0	PUB0 口上拉选择寄存器
0x10C	PIMB0	R/W	1	0x0	PB0 口内部模式配置
0x110	PIEB0	R/W	1	0x0F	PB0 口输入使能选择
0x114	PB0	R/W	1	0x0	PB0 口数据寄存器
0x118	PB0SET	R/W	1	0x0	PB0 口数据置位寄存器
0x11C	PB0CLR	R/W	1	0x0	PB0 口数据清零寄存器
0x120	GPIO_PSW1	R/W	1	0x0	GPIO_APB 写保护寄存器
0x124	PB0MASK	R/W	2	0x0	PB0 控制信号屏蔽寄存器

12.4.2 GPIO-AHB 寄存器

12.4.2.1 PA 口模式寄存器（输入或者输出）：PMA（0x00）

偏移地址：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	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
7:0	PM07~PM00	=0 输出模式 =1 输入模式	R/W	FF

当 IO 口复用为数字功能时，方向寄存器不起作用，由数字模块本身控制。

当选择为模拟口时，PMA 寄存器无效，为模拟口。

12.4.2.2 PA 口复用寄存器 0: PCA0 (0x04)

偏移地址 0x04

比 特 位	名称	描述	读/写标志	复位值
31:30	PC0405_21	保留位，应保持默认值不变	R/W	0
29	PC03_1	该寄存器与 PC03_0[1:0]共同组成 3 位配置字，决定 P03 的复用关系。详见下文。	R/W	0
28:27	PC17	定义端口 P17 复用配置： =00: 选择为 IO 口； =01: 选择为 INT4； =10: 选择为晶体 HOSCI 管脚；（备注：只要 PC16 或 PC17 任何一个选择为 10，就选择为 HOSC） =11: 选择为 SCLK3	R/W	2'b00
26:25	PC16	定义端口 P16 复用配置： =00: 选择为 IO 口； =01: 选择为 INT3； =10: 选择为晶体 HOSCO 管脚；（备注：只要 P16 和 P17 任何一个选择为 10，就选择为 HOSC） =11: SCSN3 备注：P16、P17、P22、P23 为一组，组成 SPI3。	R/W	2'b00
24:23	PC15	定义端口 P12~P15 复用配置： =00: 选择为 IO 口； =01: 选择为 MISO0/1 或 SPI2 的 SDO2；具体由 PCA1.SPI_SEL 配置决定。 =10: 选择为 TC 输出 TC0_N[1] =11: 选择为 TC 输入；	R/W	2'b00
22:21	PC14	定义端口 P14 复用配置： =00: 选择为 IO 口； =01: 选择为 SPI0 的 MOSI0 或 SPI2 的 SDI2； 选择为 MOSI0/1 或 SDI2，具体由 PCA1.SPI_SEL 配置决定。 =10: 选择为 TC 输出 TC0_P[1] =11: 选择为 TC 输入；	R/W	2'b00
20:19	PC13	定义端口 P13 复用配置： =00: 选择为 IO 口； =01: 选择为 SPI0 的 SCLK0 或 SPI2 的 SCLK2。 选择为 SCLK0/1/2，由 PCA1.SPI_SEL 配置决定。 =10: 选择为 TC 输出 TC0_N[0] =11: 选择为 TC 输入；	R/W	2'b00

18:17	PC12	定义端口 P12 复用配置： =00: 选择为 IO 口； =01: 选择为 SPI0/1/2 的 SCSN；仅在 SPI 从模式有效。 具体是哪一个，由 PCA1.SPI_SEL 配置决定。 =10: 选择为 TC 输出 TC0_P[0]， =11: 选择为 TC 输入；	R/W	2'b00
16:15	PC11	定义端口 P11 复用配置： =00: 选择为 IO 口； =01: 保留 =10: 选择为 KEY2 =11: 选择为 SPI0/1/2 的 MISO0/1 或 SDO2；具体是哪一个 SPI 由 PCA1.SPI_SEL 配置位决定	R/W	2'b00
14:13	PC10	定义端口 P10 复用配置： =00: 选择为 IO 口； =01: 保留 =10: 选择为 INT2 =11: 选择为 SPI0/1/2 的 MOSI0/1 或 SDI2；具体是哪一个 SPI 由 PCA1.SPI_SEL 配置位决定	R/W	2'b00
12:11	PC07	分别定义端口 P07 复用配置： =00: 选择为 IO 口； =01: 保留 =10: 选择为 KEY1 =11: 选择为选择为 SPI0/1/2 的 SCLK；具体是哪一个 SPI 由 PCA1.SPI_SEL 配置位决定	R/W	2'b00
10:09	PC06	定义端口 P06 复用配置： =00: 选择为 IO 口； =01: 保留 =10: 选择为 INT1 =11: 选择为 SPI0/1/2 的 SCSN；具体是哪一个 SPI 由 PCA1.SPI_SEL 配置位决定，仅在从模式有效	R/W	2'b00
08	PC0405_0	保留位，应保持默认值不变	R/W	1'b0
07:06	PC03_0	Bit29,bit07,bit06 共同构成三位寄存器，定义端口 P03 复用配置： { PC03_1, PC03_0[1:0]} =000: 选择为 IO 口； =001: 选择为 CF1 输出口； =010: 选择为 TC 输出 TC0_P[0] =011: 选择为 TC 输入 TCI =100: 选择为 RTCOUT =101: 选择为 RTC_1S =其他: 保留，默认为 IO	R/W	2'b00

05:04	PC02	定义端口 P02 复用配置。 =00: 选择为 IO 口; =01: 选择为 CF0 =10: 选择为 RTCOUT0; (其他口若配置为 RTCOUT 时, 可同时输出;) =11: 选择为模拟输入 SDAIN5	R/W	2'b00
03:02	PC01	保留位, 应保持默认值不变	R/W	2'b00
01:00	PC00	定义端口 P00 复用配置: =00: 选择为 IO 口; =01: 选择为模拟输入口 CMP0 ; =10: 选择为 INT0 ; =11: CF2	R/W	2'b00

12.4.2.3 PA 口复用寄存器 1: PCA1 (0x08)

偏移地址 0x08

比特位	名称	描述	读/写标志	复位值
31	--	保留位	R	0
30	SPI_SEL_2	与 bit23:22 的 SPI_SEL01 共同组成 3 位的配置字, 决定 SPI 的复用关系。	R/W	0
29:28	SPIS_SDI_FLTSEL[1:0]	高速 SPIS SDI 引脚的滤波配置 =00:(默认) BYPASS =01: 5ns =10: 10 ns =11: 20 ns	R/W	2'b00
27:26	SPIS_SCK_FLTSEL[1:0]	高速 SPIS SCLK 引脚的滤波配置 =00:(默认) BYPASS =01: 5 ns =10:10 ns =11:20 ns	R/W	2'b00
25:24	SPIS_SCSN_FLTSEL[1:0]	高速 SPIS SCSN 引脚的滤波配置。 =00(默认) BYPASS =01:20 ns =10:40 ns =11:80 ns	R/W	2'b00
23:22	SPI_SEL_01	Bit30,bit23,bit22 共同组成 3 位配置字, 决定 SPI 口复用关系。 {SPI_SEL_2, SPI_SEL_01} =000(默认): P31~34 配置为 SPI 时, 选择为 SPI2(高速 SPIS); P06~P11 配置为 SPI 时, 选择为 SPI0(SPIM/S);	R/W	2'b00

		<p>P12~P15 配置为 SPI 时，选择为 SPI1(SPIM/S); =001:</p> <p>P31~34 配置为 SPI 时，选择为 SPI2(高速 SPIS); P06~P11 配置为 SPI 时，选择为 SPI1(SPIM/S); P12~P15 配置为 SPI 时，选择为 SPI0(SPIM/S); =010:</p> <p>P31~34 配置为 SPI 时，选择为 SPI1(SPIM/S); P06~P11 配置为 SPI 时，选择为 SPI0(SPIM/S); P12~P15 配置为 SPI 时，选择为 SPI2(高速 SPIS); =011:</p> <p>P31~34 配置为 SPI 时，选择为 SPI1(SPIM/S); P06~P11 配置为 SPI 时，选择为 SPI2(高速 SPIS); P12~P15 配置为 SPI 时，选择为 SPI0(SPIM/S); =100:</p> <p>P31~34 配置为 SPI 时，选择为 SPI0(SPIM/S); P06~P11 配置为 SPI 时，选择为 SPI2(高速 SPIS); P12~P15 配置为 SPI 时，选择为 SPI1(SPIM/S); =101:</p> <p>P31~34 配置为 SPI 时，选择为 SPI0(SPIM/S); P06~P11 配置为 SPI 时，选择为 SPI1(SPIM/S); P12~P15 配置为 SPI 时，选择为 SPI2(高速 SPIS); =其他:</p> <p>保留位，使用默认配置</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th></th> <th>000</th> <th>001</th> <th>010</th> <th>011</th> <th>100</th> <th>101</th> <th>其他</th> </tr> </thead> <tbody> <tr> <td>P3</td> <td>SPI2</td> <td>SPI2</td> <td>SPI1</td> <td>SPI1</td> <td>SPI0</td> <td>SPI0</td> <td>SPI2</td> </tr> <tr> <td>P0</td> <td>SPI0</td> <td>SPI1</td> <td>SPI0</td> <td>SPI2</td> <td>SPI2</td> <td>SPI1</td> <td>SPI0</td> </tr> <tr> <td>P1</td> <td>SPI1</td> <td>SPI0</td> <td>SPI2</td> <td>SPI0</td> <td>SPI1</td> <td>SPI2</td> <td>SPI1</td> </tr> </tbody> </table>		000	001	010	011	100	101	其他	P3	SPI2	SPI2	SPI1	SPI1	SPI0	SPI0	SPI2	P0	SPI0	SPI1	SPI0	SPI2	SPI2	SPI1	SPI0	P1	SPI1	SPI0	SPI2	SPI0	SPI1	SPI2	SPI1		
	000	001	010	011	100	101	其他																													
P3	SPI2	SPI2	SPI1	SPI1	SPI0	SPI0	SPI2																													
P0	SPI0	SPI1	SPI0	SPI2	SPI2	SPI1	SPI0																													
P1	SPI1	SPI0	SPI2	SPI0	SPI1	SPI2	SPI1																													
21:20	PC37	<p>端口 P37 复用配置:</p> <p>=00: 选择为 IO 口;</p> <p>=01: 选择为 CF4;</p> <p>=1x: 选择为 RTCOUT; (P02, P42 若配置为 RTCOUT 时, 可同时输出;)</p>	R/W	2'b00																																
19:18	PC36	<p>定义端口 P36 复用配置:</p> <p>=00: 选择为 IO 口;</p> <p>=01: 选择为 CF3 输出口;</p> <p>=10: 选择为 TC 输出 TC0_P[1]</p> <p>=11: 选择为 TC 输入 TCI</p>	R/W	2'b00																																
17:16	PC35	<p>端口 P35 复用配置:</p> <p>=00: 选择为 IO 口;</p> <p>=01: 选择为 CF2 输出口;</p> <p>=1x: 选择为 INT5 输入口;</p>	R/W	2'b00																																
15	PC3134	端口 P31~34 复用配置:	R/W	1'b0																																

		=0: 选择为 IO 口; =1: 选择为 SPI 接口, 进一步选择为 SPI0/1(SPIM/S)接口还是 SPI2(高速 SPIS)接口由 SPI_SEL 位决定。 当选择为 SPI2 接口 P31 为 SDO2, P32 为 SDI2, P33 为 SCLK2, 从模式时 P34 为 SCSN2, 此时需将 P34 输入使能打开, 即 PIEA[28]=0 当选择为 SPI0/1(SPIM/S)接口 P31 为 MISO0/1, P32 为 MOSI0/1, P33 为 SCLK0/1, 从模式时 P34 为 SCSN0/1, 此时需将 P34 输入使能打开, 即 PIEA[28]=0; 注意: 主模式时 P34 为普通 IO 口。		
14:13	PC30	定义端口 P30 复用配置: =00: 选择为 IO 口; =01: 选择为 TCI; =1x: 选择为 KEY3;	R/W	2'b00
12:11	PC27	端口 P26 和 P27 复用配置: =00: 选择为 IO 口; =01: 选择为 UART3 接口, P27 为 TX3; =10: 选择为 TC 输出; P27 为 TC1_N[1]。 =11: 选择为 TC 输入;	R/W	2'b00
10:09	PC26	端口 P26 和 P27 复用配置: =00: 选择为 IO 口; =01: 选择为 UART3 接口, P26 为 RX3; =10: 选择为 TC 输出; P26 为 TC1_N[0]。 =11: 选择为 TC 输入;	R/W	2'b00
08:07	PC25	端口 P25 复用配置: =00: 选择为 IO 口; =01: 选择为 I2C 接口, P25 为 SDA; =10: 选择为 TC 输出; P25 为 TC1_P[1] =11: 选择为 TC 输入;	R/W	2'b00
06:05	PC24	端口 P24 复用配置: =00: 选择为 IO 口; =01: 选择为 I2C 接口, P24 为 SCL; =10: 选择为 TC 输出; P24 为 TC1_P[0] =11: 选择为 TC 输入;	R/W	2'b00
04	UART2_SEL	=0: P22 和 P23 配置由 PC22 和 PC23 决定; =1: P22/P23 配置为 UART2,P22 为 RX2, P23 为 TX2;	R/W	1'b0
03	PC23	端口 P23 复用配置: =0: 选择为 IO 口; =1: 选择为 KEY5	R/W	1'b0
02	PC22	端口 P22 复用配置: =0: 选择为 IO 口; =1: 选择为 KEY4	R/W	1'b0

01:00	PC2021	端口 P20 和 P21 复用配置： =2'b00: 选择为 IO 口； =2'b01: 选择为 UART1 接口； =2'b10: 选择为 SWD，默认配置； =2'b11: 选择为 UART2 接口	R/W	2'b10
-------	--------	---	-----	-------

12.4.2.4 PA 口复用寄存器 2: PCA2 (0x28)

偏移地址 0x08

比特位	名称	描述	读/写标志	复位值
31:30	PC37_2	=2'b00: P37 口复用功能由 PCA1(0x08) 决定 =2'b01: 选择为 CF0 =2'b10: 保留 =2'b11: 保留	R/W	2'b00
29:28	PC36_2	=2'b00: P36 口复用功能由 PCA1(0x08) 决定 =2'b01: 选择为 SDA =2'b10: 保留 =2'b11: 保留	R/W	2'b00
27:26	PC35_2	=2'b00: P35 口复用功能由 PCA1(0x08) 决定 =2'b01: 选择为 SCL =2'b10: 保留 =2'b11: 保留	R/W	2'b00
25:24	PC34_2	=2'b00: 由 PCA1 寄存器 =2'b01: 选择为 TX2 =2'b10: 选择为 CF1 =2'b11: 选择为 RTC_1S	R	2'b00
23:22	PC33_2	=2'b00: 由 PCA1 寄存器决定 =2'b01: 选择为 RX2 =2'b10: 选择为 CF2 =2'b11: 选择为 RTC_OUT	R	2'b00
21:20	PC32_2	=2'b00: 由 PCA1 寄存器决定 =2'b01: 选择为 SDA =2'b10: 选择为 CF3 =2'b11: 选择为 RTC_1S	R	2'b00
19:18	PC31_2	=2'b00: 由 PCA1 寄存器决定 =2'b01: 选择为 SCL =2'b10: 选择为 CF4 =2'b11: 选择为 RTC_OUT	R	2'b00

17:16	PC30_2	=2'b00: P30 口复用功能由 PCA1(0x08) 决定 =2'b01: 选择为 CF4 =2'b10: 选择为 TC1_P[1] =2'b11: 选择为 RTC_OUT	R/W	2'b00
15:14	PC27_2	=2'b00: P27 口复用功能由 PCA1(0x08) 决定 =2'b01: 选择为 CF3 =2'b10: 选择为 RTC_1S =2'b11: 选择为 MISO4	R/W	2'b00
13:12	PC26_2	=2'b00: P26 口复用功能由 PCA1(0x08) 决定 =2'b01: 选择为 CF2 =2'b10: 选择为 RTC_OUT =2'b11: 选择为 MOSI4	R/W	2'b00
11:10	PC25_2	=2'b00: P25 口复用功能由 PCA1(0x08) 决定 =2'b01: 选择为 SCLK4 =2'b10: 保留 =2'b11: RTC_OUT	R/W	2'b00
09:08	PC24_2	=2'b00: P24 口复用功能由 PCA1(0x08) 决定 =2'b01: 选择为 SCSN4 =2'b10: 保留 =2'b11: CF4 备注: P24、P25、P26、P27 为一组, 组成 SPI4。	R/W	2'b00
07:06	PC23_2	=2'b00: P23 口复用功能由 PCA1(0x08) 决定 =2'b01: 选择为 MISO3 =2'b10: 选择为 TC1_P[0] =2'b11: RTC_OUT	R/W	2'b00
05:04	PC22_2	=2'b00: P22 口复用功能由 PCA1(0x08) 决定 =2'b01: 选择为 MOSI3 =2'b10: 选择为 TC1_N[0] =2'b11: CF4 备注: P16、P17、P22、P23 为一组, 组成 SPI3。	R/W	2'b00
03:02	PC21_2	=2'b00: P21 口复用功能由 PCA1(0x08) 决定 =2'b01: 选择为 CF2 =2'b10: 选择为 RTC_OUT =2'b11: 选择为 TC0_P[0]	R/W	2'b00
01:00	PC20_2	=2'b00: P20 口复用功能由 PCA1(0x08) 决定 =2'b01: 选择为 CF3 =2'b10: 选择为 RTC_1S =2'b11: 选择为 TC0_P[1]	R/W	2'b00

12.4.2.5 PA 口上拉选择寄存器: PUA (0x0C)

偏移地址 0x0C

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:	R/W	0x00
		=0: 不选择上拉;		
		=1: 选择上拉;		
		P30 口作为 ISP 使能时, BOOTROM 会选择上拉, 防止上电时输入状态不确定。		
23:16	PU27~PU20	定义端口上拉配置:	R/W	0x03
		=0: 不选择上拉;		
		=1: 选择上拉;		
		备注: P20 和 P21 作为 SWD 默认上拉使能。		
15:08	PU17~PU10	定义端口上拉配置:	R/W	0x00
		=0: 不选择上拉;		
		=1: 选择上拉;		
07:00	PU07~PU00	定义端口上拉配置:	R/W	0x00
		=0: 不选择上拉;		
		=1: 选择上拉;		

备注: 当 IO 口处于输出模式或者模拟功能时, 不管 PU 寄存器如何配置, PIN 上拉均不使能。

注意: PU01/PU04/PU05/PU35/PU36/PU37 相关 bit 是保留位, 应保持默认值不变。

12.4.2.6 PA 口内部模式配置 0: PIMA0 (0x10)

偏移地址 0x10

比特位	名称	描述	读/写标志	复位值
31:24	PID37~PID30	定义端口 P30~P37 是否是 N-ch 漏极开路输出:	R/W	0x00
		=0: 普通模式;		
		=1: N-ch 漏极开路模式;		
23:16	PID27~PID20	定义端口 P27~P20 是否是 N-ch 漏极开路输出:	R/W	0x00
		=0: 普通模式;		
		=1: N-ch 漏极开路模式;		
15:08	PID17~PID10	定义端口 P10~P17 是否是 N-ch 漏极开路输出:	R/W	0x0
		=0: 普通模式;		
		=1: N-ch 漏极开路模式;		
07:00	PID07~PID00	定义端口 P00~P07 是否是 N-ch 漏极开路输出:	R/W	0x00
		=0: 普通模式;		
		=1: N-ch 漏极开路模式;		

注意: PID01/PID04/PID05/PID35/PID36/PID37 相关 bit 是保留位, 应保持默认值不变。

12.4.2.7 PA 口内部模式配置 1: PIMA1 (0x14)

偏移地址 0x14

比特位	名称	描述	读/写标志	复位值
31:24	PIL37~PIL30	定义端口 P30~P37 输入缓冲器类型:	R/W	0x00
		=0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$;		
		=1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$;		
23:16	PIL27~PIL20	定义端口 P27~P20 输入缓冲器类型:	R/W	0x00
		=0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$;		
		=1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$;		
15:08	PIL17~PIL10	定义端口 P10~P17 输入缓冲器类型:	R/W	0x0
		=0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$;		
		=1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$;		
07:00	PIL07~PIL00	定义端口 P00~P07 输入缓冲器类型:	R/W	0x00
		=0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$;		
		=1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$;		

注意: PIL01/PIL04/PIL05/PIL35/PIL36/PIL37 相关 bit 是保留位, 应保持默认值不变。

12.4.2.8 PA 口输入使能: PIEA (0x18)

偏移地址 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	0xFF
		=1: 不使能输入;		
		=0: 使能输入;		
		注: 上电后 P30 由 BOOTROM 设为输入使能, 并选择上拉使能, 方便 ISP 进行检测。		
23:16	PIE27~PIE20	输入使能:	R/W	0xFF
		=1: 不使能输入;		
		=0: 使能输入;		
15:08	PIE17~PIE10	输入使能:	R/W	0xFF
		=1: 不使能输入;		
		=0: 使能输入;		
07:00	PIE07~PIE00	输入使能:	R/W	0xFF
		=1: 不使能输入;		
		=0: 使能输入;		

注意: PIE01/PIE04/PIE05/PIE35/PIE36/PIE37 相关 bit 是保留位, 应保持默认值不变。

12.4.2.9 PA 口数据寄存器: PA (0x1C)

偏移地址: 0x1C

比特位	名称	描述	读/写标志	复位值
31:24	P37~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	R/W	0x00

		在输出模式下读端口，则读取的是输出锁存器的值		
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	0x00
15:14	P17~P16	P17 和 P16 数据输入寄存器，只读；(HOSC 时钟复用口)	R	0x00
13:8	P15~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	0x00
7:0	P07~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	0x00

注意：P01/P04/P05/P35/P36/P37 相关 bit 是保留位，应保持默认值不变。

12.4.2.10 PA 口数据置位寄存器：PASET (0x20)

偏移地址 0x20

比特位	名称	描述	读/写标志	复位值
31:24	P37~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	0x00
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	0x00
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	0x00
7:0	P07~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	0x00

注意：1. 读出值无意义。2. 本寄存器不能用 bitband 方式来设置。

注意：PIL01/PIL04/PIL05/PIL35/PIL36/PIL37 相关 bit 是保留位，应保持默认值不变。

注意：P01/P04/P05/P35/P36/P37 相关 bit 是保留位，应保持默认值不变。

12.4.2.11 PA 口清零寄存器：PACLR (0x24)

偏移地址 0x24

比特位	名称	描述	读/写标志	复位值
31:24	P37~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	0x00
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	0x00

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	0x00
7:0	P07~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	0x00

注意: 1. 读出值无意义。2. 本寄存器不能用 bitband 方式来设置。

注意: P01/P04/P05/P35/P36/P37 相关 bit 是保留位, 应保持默认值不变。

12.4.2.12 PB1 口模式寄存器 (输入或者输出): PMB1 (0x30)

偏移地址 0x30

比特位	名称	描述	读/写标志	复位值
31:16	Reserved	保留	R	0x0
15:08	PM67~PM60	=0 输出模式 =1 输入模式 PM61/PM62/PM63/PM64/PM65/PM66/PM67 应保持默认值	R/W	0xFF
07:04	PM47~PM44	=0 输出模式 =1 输入模式	R/W	0xF
3:0	Reserved	保留	R	0x0

当 IO 口设置为数字功能时, 方向寄存器不起作用, 由数字模块本身控制。

当选择为模拟口时, PMB 寄存器无效, 为模拟口。

12.4.2.13 PB1 口复用寄存器: PCB1 (0x34)

偏移地址 0x34

比特位	名称	描述	读/写标志	复位值
31:28	Reserved	保留。注: 内部使用, 用户不可写	R/W	0
27:26	PC67	保留, 不要改变默认值	R/W	2'b00
25:24	PC66	保留, 不要改变默认值	R/W	2'b00
23:22	PC65	保留, 不要改变默认值	R/W	2'b00
21:20	PC64	保留, 不要改变默认值	R/W	2'b00
19:18	PC63	保留, 不要改变默认值	R/W	2'b00
17:16	PC62	保留, 不要改变默认值	R/W	2'b00
15:14	PC61	保留, 不要改变默认值	R/W	2'b00
13:12	PC60	端口 P60 复用配置: =00: 选择为 IO 口; =01: 选择为 sigma-delta ADC 输入 SDAIN8; =10: CF0; =11: RTC_OUT;	R/W	2'b00
11:09	Reserved	保留	R	0
08:06	PC47	端口 P47 复用配置:	R/W	3'b000

		=000: 选择为 IO 口; =001: 选择为 sigma-delta ADC 输入 SDAIN3; =010: 选择为 RX5; =011: 选择为 KEY7; (PC41 选择 KEY7 较 PC47 优先级高) =100: 选择为 CF4 =101: 选择为 CF0 =110: 选择为 RTC_OUT =其他: 保留, 不要配置		
05:04	PC46	端口 P46 复用配置: =00: 选择为 IO 口; =01: 选择为 sigma-delta ADC 输入 SDAIN2; =10: 选择为 TX5; =11: 选择为 KEY6; (PC40 选择 KEY6 较 PC46 优先级高)	R/W	2'b00
03:02	PC45	端口 P45 复用配置: =00: 选择为 IO 口; =01: 选择为 sigma-delta ADC 输入 SDAIN1; =10: 选择为 RX4; =11: 选择为 INT7;	R/W	2'b00
01:00	PC44	定义端口 P44 复用配置: =00: 选择为 IO 口; =01: 选择为 sigma-delta ADC 输入 SDAIN0 =10: 选择为 TX4; =11: 选择为 INT6;	R/W	2'b00

12.4.2.14 PUB1 口上下拉选择寄存器: PUB1 (0x38)

偏移地址 0x38

比特位	名称	描述	读/写标志	复位值
31:12	Reserved	保留	R	0
15:8	PU67~PU60	PU67~PU60 定义 P6 端口是否内接上拉: =0: 不接上拉; =1: 内接上拉。 PU61/PU62/PU63/PU64/PU65/PU66/PU67 应保持默认值	R/W	00
7:4	PU47~PU44	PU47~PU44 定义 P4 端口是否内接上拉: =0: 不接上拉; =1: 内接上拉。	R/W	00
3:0	Reserved	保留	R	0

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。

12.4.2.15 PB1 口内部模式配置: PIMB1 (0x3C)

偏移地址 0x3C

比特位	名称	描述	读/写标志	复位值
31:24	PIL67~PIL60	定义端口 P67~P60 输入缓冲器类型: =0: CMOS 缓冲器, $V_{il}=0.3VCC$ $V_{ih}=0.7VCC$; =1: TTL 缓冲器, $V_{il}=0.16VCC$ $V_{ih}=0.4VCC$; PIL61/PIL62/PIL63//PIL64/PIL65/PIL66/PIL67 应保持默认值	R/W	0
23:20	PIL47~PIL44	定义端口 P40~P47 输入缓冲器类型: =0: CMOS 缓冲器, $V_{il}=0.3VCC$ $V_{ih}=0.7VCC$; =1: TTL 缓冲器, $V_{il}=0.16VCC$ $V_{ih}=0.4VCC$;	R/W	0
19:16	Reserved	保留	R	0
15:08	PID67~PID60	定义端口 P67~P60 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; PID61/PID62/PID63/PID64/PID65/PID66/PID67 应保持默认值	R/W	0
07:04	PID47~PID44	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	0
3:0	Reserved	保留	R	0

12.4.2.16 PB1 口输入使能: PIEB1 (0x40)

偏移地址 0x40

比特位	名称	描述	读/写标志	复位值
31:16	Reserved	保留	R	0x 0
15:8	PIE67~PIE60	输入使能: =1: 不使能输入; =0: 使能输入; PIE61/PIE62/PIE63/PIE64/PIE65/PIE66/PIE67 应保持默认值	R/W	0xFF
7:4	PIE47~PIE44	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	0xF
3:0	Reserved	保留	R	0x0

12.4.2.17 PB1 口数据寄存器: PB1 (0x44)

偏移地址 0x44

比特位	名称	描述	读/写标志	复位值
31:16	Reserved	保留	R	0
15:8	P67~P60	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 P61/P62/P63/P64/P65/P66/P67 应保持默认值		
7:4	P47~P44	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
3:0	Reserved	保留	R	0

12.4.2.18 PB1 口数据置位寄存器：PBSET1 (0x48)

偏移地址 0x48

比特位	名称	描述	读/写标志	复位值
31:16	Reserved	保留	R	0
15:8	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 P61/P62/P63/P64/P65/P66/P67 应保持默认值	R/W	00
7:4	P47~P44	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
3:0	Reserved	保留	R	0

注意：1. 读出值无意义。2. 本寄存器不能用 bitband 方式来设置。

12.4.2.19 PB1 口清零置位寄存器：PBCLR1 (0x4C)

偏移地址 0x4C

比特位	名称	描述	读/写标志	复位值
31:16	Reserved	保留	R	0
15:8	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 P61/P62/P63/P64/P65/P66/P67 应保持默认值	R/W	00
7:4	P47~P44	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
3:0	Reserved	保留	R	0

注意：1. 读出值无意义。2. 本寄存器不能用 bitband 方式来设置。

12.4.2.20 GPIO_AHB 写保护寄存器：GPIO_PSW0 (0x50)

偏移地址：0x50

比特位	名称	描述	读/写标志	复位值
31:8	--	--	--	--

7:0	GPIO_PSW0	当 PSW=0x82, 0x0~0x18、0x30~0x40 寄存器可配置	R/W	0
-----	-----------	---------------------------------------	-----	---

12.4.2.21 脉冲计数寄存器: IOCNT (0x054)

偏移地址: 0x054

比特位	名称	描述	读/写标志	复位值
31:18	---	保留位	--	0x0
17	IOCNT_EN	=0: 禁止脉冲计数功能, 写 0 时同时清 CNT[15:0] =1: 使能脉冲计数功能;	R/W	0x0
16	IOCNT_OK	计算完成标志 =0: 表示计数正在进行中; =1: 表示计数已完成。 当 CNT=0 时, IOCNT_OK 一直为 0, 即 IOCNT_EN 写 0 也可以清 IOCNT_OK; 当 CNT 不等于 0 时, 并且 CNT 已经持续 1ms 没有更新, 那么表示计数已完成, 该位被置 1。 关于 1ms 定时: cpu 16MHz 运行时是 1ms。	R/W	--
15:0	CNT	当 IOCNT_EN=0 时, CNT 自动清零, 并且不做计数; 当 IOCNT_EN=1 时, 一旦有脉冲下沿出现, 计数器加 1, 最大计数值限定为 65535 (16'hFFFF)	R	0

该功能选择在 P44 口, 用于支持脉冲计数器类型的温度传感器。

需要将 P44 口复用为 GPIO 功能, 并设置为输入模式-输入使能。

编程模型:

- 1) CPU 通过 IO 口使能某路温度测量;
- 2) CPU 配置 IOCNT_EN=1;
- 3) CPU 在 ms 定时中查询 IOCNT_OK 标志位;
- 4) CPU 查询到 IOCNT_OK 标志位=1, 读取 CNT 寄存器值, 计算得到该路温度值;
- 5) CPU 配置 IOCNT_EN=0;
- 6) CPU 通过 IO 口关闭该路温度测量。

12.4.3 GPIO-APB 寄存器

12.4.3.1 PB0 口模式寄存器 (输入或者输出): PMB0 (0x100)

偏移地址 0x100

比特位	名称	描述	读/写标志	复位值
31:10	Reserved	保留	R	0
9:8	PM51~PM50	=0 输出模式 =0 输出模式	R/W	0x3
7:4	Reserved	保留	R	0
3:0	PM43~PM40	=0 输出模式 =0 输出模式	R/W	0xF

当 IO 口设置为数字功能时，方向寄存器不起作用，由数字模块本身控制。

当选择为模拟口时，PMB 寄存器无效，为模拟口。

12.4.3.2 PB0 口复用寄存器：PCB0 (0x104)

偏移地址 0x104

比特位	名称	描述	读/写标志	复位值
31:16	Reserved	保留	R	0
15:14	PC51	端口 P51 复用配置： =00：选择为 IO 口； =01：选择为模拟输入口 AIN5； =10：选择为 KEYOUT =11：选择为 RTC_1S	R/W	0
13:12	PC50	端口 P50 复用配置： =00：选择为 IO 口； =01：选择为模拟输入口 AIN4； =10：选择为 KEY6，P40 和 P42 优先级高 =11：选择为 RTC_OUT	R/W	0
11:9	Reserved	保留	R	0
08	PC42_1	PC42_1 与 PC42 组成 3bit 数 {PC42_1, PC42[1:0]}，定义端口 P42 复用配置	R/W	0
07:06	PC43	端口 P43 复用配置： =00：选择为 IO 口； =01：选择为模拟输入口 AIN0； =10：选择为 SDA =11：选择为按键触发指示信号输出 KEYOUT	R/W	0
05:04	PC42	PC42_1 (bit8) 与 PC42 (bit5~bit4) 共同组成 3bit 数 {PC42_1, PC42[1:0]}，定义端口 P42 复用配置： =000：选择为 IO 口； =001：选择为 RTCOUT；（P02，P37 若配置为 RTCOUT 时，可同时输出；） =01x：选择为模拟输入口 AIN1； =100：选择为 SCL =101：选择为 KEY6，同时选择时 P40 口优先级高 =110：选择为 RTC_1S(直接是万年历秒脉冲，不是 PLL_1Hz) =111：保留	R/W	0

03:02	PC41	端口 P41 的复用配置： =00：选择为 IO 口； =01：选择为 KEY7；（1. 具备开盖事件记录功能；2、PC41 选择 KEY7 较 P47 优先级高）； =1x：选择为模拟输入口 AIN2；	R/W	0
01:00	PC40	端口 P40 的复用配置： =00：选择为 IO 口； =01：选择为 KEY6；（1. 具备开盖事件记录功能；2. PC40 选择 KEY6 较 P46 优先级高）； =1x：选择为模拟输入口 AIN3；	R/W	0

关于按键触发指示信号输出 KEYOUT 的说明：

1. 当 cpu 使能 P43 为该功能，默认情况下 P43 为高阻态；
2. 当开盖检测按键 KEY6/KEY7 功能一个或两个使能时，任意一个触发动作都会让 P43 输出低电平。该触发动作包括 KEY6 或 KEY7 的上沿或者下沿中断。
3. 经过大约 2 秒左右的延迟，P43 就自动从输出 0 变为高阻态；
4. CPU 也可通过软件配置 P43 为 IO 口/高阻态。由 CPU 控制 P43 口退出低电平状态。

关于 VBAT 域 IIC 的说明：

1. 芯片运行模式下，cpu 设置 P42 和 P43 的复用功能为 IIC，P42 和 P43 由 IIC 硬件控制器控制，用于跟外部 EEPROM 之间的通信；
2. 芯片在休眠模式下，cpu 可以选择使能 IIC 的 DMA 传输模式，DMA 传输的配置放在 RTC 模块。具体的过程是：cpu 使能 DMA，IIC 控制权交给 DMA；当 KEY6/KEY7 功能一个或者两个使能时，任意一个触发动作都会让 DMA 启动，DMA 启动后将万年历时间信息自动写入 EEPROM 的指定地址。
详见 RTC 章节描述。

12.4.3.3 PUB0 口上下拉选择寄存器：PUB0 (0x108)

偏移地址 0x108

比特位	名称	描述	读/写标志	复位值
31:10	Reserved	保留	R	0
9:8	PU51~PU50	PU51~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
7:4	Reserved	保留		
3:0	PU43~PU40	PU43~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

备注：当 IO 口处于输出模式或者模拟功能时，不管 PU 寄存器如何配置，PIN 上拉均不使能。

12.4.3.4 PB0 口内部模式配置 0: PIMB0 (0x10C)

偏移地址:0x10C

比特位	名称	描述	读/写标志	复位值
31:14	Reserved	保留	R	0
13:12	PIL51~PIL50	定义端口 P51~P50 输入缓冲器类型: =0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$; =1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$;	R/W	0
11:10	Reserved	保留	R	0
09:08	PID51~PID50	定义端口 P51~P50 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	0
07:04	PIL43~PIL40	定义端口 P43~P40 输入缓冲器类型: =0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$; =1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$;	R/W	0
03:00	PID43~PID40	定义端口 P43~P40 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	0

12.4.3.5 PB0 口输入使能: PIEB0 (0x110)

偏移地址 0x110

比特位	名称	描述	读/写标志	复位值
31:10	Reserved	保留	R	0
9:8	PIE51~PIE50	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	0x3
7:4	Reserved	保留	R	0
3:0	PIE43~PIE40	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	0xF

12.4.3.6 PB0 口数据寄存器: PB0 (0x114)

偏移地址 0x114

比特位	名称	描述	读/写标志	复位值
31:10	Reserved	保留	R	0
9:8	P51~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	0
7:4	Reserved	保留	R	0
3:0	P43~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	R/W	0

		在输出模式下读端口，则读取的是输出锁存器的值		
--	--	------------------------	--	--

12.4.3.7 PB0 口数据置位寄存器：PBSET0 (0x118)

偏移地址 0x118

比特位	名称	描述	读/写标志	复位值
31:10	Reserved	保留	R	0
9:8	P51~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	0
7:4	Reserved	保留	R	0
3:0	P43~P40	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

注意：1. 读出值无意义。

12.4.3.8 PB0 口清零置位寄存器：PBCLR0 (0x11C)

偏移地址 0x11C

比特位	名称	描述	读/写标志	复位值
31:10	Reserved	保留	R	0
9:8	P51~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	0
7:4	Reserved	保留	R	0
3:0	P43~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	0

注意：1. 读出值无意义。

12.4.3.9 GPIO_APB 写保护寄存器：GPIO_PSW1 (0x120)

偏移地址：0x120

比特位	名称	描述	读/写标志	复位值
31:8	--	--	--	--
7:0	GPIO_PSW1	当 PSW=0x82，0x100~0x110、0x124 寄存器可配置	R/W	0

12.4.3.10 PB0 控制信号屏蔽寄存器：PB0MASK (0x124)

偏移地址：0x124

比特位	名称	描述	读/写标志	复位值
31:16	--	--	--	--
15:0	PB0MASK	PB0MASK[15:0]=16'h8a5e，PB0 的控制信号 id/il/up/ien/oen/aen 有效；若为其他值，控制信号被屏	R/W	0

		蔽；PB0 处于缺省态。 使用时，先配置 0x100~0x110 寄存器，之后配置 PBMASK[15:0]=16'h8a5e，0x100~0x110 配置值才有效，否则无效，保持原状态。		
--	--	---	--	--

12.5 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、配置写保护寄存器 GPIO_PSW0 和 GPIO_PSW1 为 0x82,其中 GPIO_PSW0 用于 PA/PB1 写使能，GPIO_PSW1 用于 PB0 的写使能
- 3、配置 GPIO 输入输出模式。
- 4、配置 GPIO 口数据寄存器。
- 5、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。
- 6、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。
- 7、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。
- 8、若使用 PB0 管脚，还需配置 PBMASK[15:0]=16'h8a5e，其他引脚忽略此步骤。

13 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。

SoC 具有 8 个外部中断引脚，共用 1 个中断号。中断号的分配请参考表 3-1 “中断/异常向量表及其配置信息”。

SoC 支持中断的高电平触发模式。中断高电平脉宽至少保持当前处理器时钟的两拍以上。

INTC 的硬件滤波时间为 10 个系统时钟。

13.1 特性

- 支持 8 个外部中断的模式设置：上下边沿，双边沿；
- 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；

13.2 寄存器

基地址	0x40044000				
偏移地址	名称	R/W	有效字长	复位值	功能描述
0x00	INTC_CTL	R/W	1	0x0	INTC 控制寄存器
0x04	INTC_MODE	R/W	2	0x0	INTC 模式寄存器
0x08	INTC_MASK	R/W	1	0x0	INTC Mask 寄存器
0x0C	INTC_STA	R/W	1	0x0	INTC 状态寄存器

13.2.1 INTC_CTL(0x00)

INTC 控制寄存器 偏移地址=0x0

比特位	名称	描述	读/写标志	复位值
31:08	---	预留	R	0
7:0	Enable	使能信号，Enable[7:0]对应于 Extern_in[7:0] 0: 关闭对应的外部中断 1: 使能对应的外部中断	R/W	0

13.2.2 INTC_MODE(0x04)

INTC 模式寄存器 偏移地址=0x4

比特位	名称	描述	读/写标志	复位值
31:16	---	预留	R	0
15:14	MODE7	Extern Int 7 模式选择 00: 上升沿 01: 下降沿 10: 双边沿	R/W	0

		Others: Reserved		
13:12	MODE6	Extern Int 6 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 Others: Reserved	R/W	0
11:10	MODE5	Extern Int 5 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 Others: Reserved	R/W	0
9:8	MODE4	Extern Int 4 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 Others: Reserved	R/W	0
7:6	MODE3	Extern Int 3 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 Others: Reserved	R/W	0
5:4	MODE2	Extern Int 2 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 Others: Reserved	R/W	0
3:2	MODE1	Extern Int 1 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 Others: Reserved	R/W	0
1:0	MODE0	Extern Int 0 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 Others: Reserved	R/W	0

13.2.3 INTC_MASK(0x08)

INTC Mask 寄存器

偏移地址=0x8

比特位	名称	描述	读/写标志	复位值
31:08	---	预留	R	0
07:00	MASK	MASK[7:0]对应于 Extern_int[7:0] 0: 中断禁止 1: 中断使能	R/W	0

13.2.4 INTC_STA(0x0C)

INTC 状态寄存器

偏移地址=0xc

比特位	名称	描述	读/写标志	复位值
31:08	---	预留	R	0
07:00	STA	STA[7:0]对应于 Extern_int[7:0] 0: 中断事件未发生 1: 中断事件发生 Note: 写 1 清零	R/W	0

14 KBI

14.1 特性

SoC 支持 8 个按键中断。8 个按键共用一个中断号，中断号分配请参考表 3-1 “中断/异常向量表及其配置信息”。

- 每个按键可独立设置；
- 按键状态可读取；
- 支持输入滤波，滤波时间为 24ms；
- 每个按键中断可单独屏蔽

14.2 寄存器

基地址	0x40028000				
偏移地址	名称	R/W	有效字长	复位值	功能描述
0x00	KBI_CTL	R/W	1	0x0	KBI 控制寄存器
0x04	KBI_SEL	R/W	1	0x0	KBI 选择寄存器
0x08	KBI_DATA	R/W	1	0x0	KBI 数据寄存器
0x0C	KBI_MASK	R/W	1	0x0	KBI Mask 寄存器

14.2.1 KBI_CTL(0x00)

KBI 控制寄存器 偏移地址=0x00

比特位	名称	描述	读/写标志	复位值
31:08	---	预留	R	0
7:0	EN	使能信号，EN[7:0]对应于 KEY[7:0] 0: 关闭对应的 KEY 1: 使能对应的 KEY	R/W	0

14.2.2 KBI_SEL(0x04)

KBI 选择寄存器 偏移地址=0x4

比特位	名称	描述	读/写标志	复位值
31:08	---	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

14.2.3 KBI_DATA(0x08)

KBI 数据寄存器 偏移地址=0x8

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
07:00	DAT	DAT[7:0]对应于 KEY[7:0] 0: 按键没有按下 1: 按键被按下	R/W	0

		Note: 写1清零		
--	--	------------	--	--

14.2.4 KBI_MASK(0x0C)

KBI Mask 寄存器

偏移地址=0x0c

比特位	名称	描述	读/写标志	复位值
31:08	---	预留	R	0
07:00	MASK	MASK[7:0]对应于 KEY[7:0] 0: 中断禁止 1: 中断使能	R/W	0

14.3 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;
}
```

6. 所有 KBI 中断均为 1 个入口，需根据 KBI_DATA 判断 KBI 产生的中断。
7. 完成

15 通用 DMA 控制器（SPI1/SPIS/SPI3/SPI4/UART2 增加）

SPI1、SPIS、SPI3、SPI4 以及 UART2 支持通用 DMA 功能。该章节对通用 DMA 的主要功能做简单的介绍，相关的寄存器定义在各自的章节里。

通用 DMA 可支持 DMA 同时收发或只收、只发。接收指的是从通信接口收到数据通过 DMA 存到 SRAM，发送指的是把 SRAM 数据通过 DMA 传送到通信口，然后再发送出去。

15.1 特点

- DMA 起始地址支持按 Byte 对齐；该地址指的是 SRAM 基地址的低 16bit。
- 支持 $n \times \text{Byte}$ 位数据接收/发送，其中 n 为任意正整数；
- 支持 DMA 发送、接收独立使能，地址和长度可独立配置，并且支持同时收发；
- 支持 DMA buffer 发送/接收半满、全满中断；
- 支持循环发送/接收模式（对发送/接收 buffer 进行循环操作）或者单次模式；
- 支持 DMA 发送/接收错误检测；

15.2 实现原理

通用 DMA 模块内部有一个 32bit 的发送缓存和一个 32bit 的接收缓存，并且有一个发送 shift 寄存器和一个接收 shift 寄存器。

如果使能了 DMA 发送功能，当 DMA 指定长度数据没有发送完成，并且发送缓存为空，则会自动发起 DMA 读请求，从 SRAM 指定地址读取数据到发送缓存。

如果使能了 DMA 接收功能，每次接收 shift 寄存器接收完指定长度数据（Byte/Halfword/Word），就会把数据放进接收缓存，只要缓存里面有数据，则会发起一次 DMA 写请求，把数据写进 SRAM 指定位置。

15.3 模式选择

模块支持 DMA 接收和发送，并且两个功能是相互独立的，可任意配置。

15.3.1 只使能 DMA 发送

配置 DMA_CTL[1:0]=0x1，使能 DMA 发送功能，DMA 接口从 SRAM 指定位置读取数据，发送到串行接口。注意此时 DMA 接收到的数据会在寄存器中被覆盖，接收数据丢失。

15.3.2 只使能 DMA 接收

配置 DMA_CTL[1:0]=0x2，使能 DMA 接收功能，把从串行接收上接收到的数据通过 DMA 接口写到 SRAM 指定位置。注意此时 DMA 发送的数据无效（将会重复发送指定寄存器的值或者发送无效数据，根据具体模块设计）。

15.3.3 同时使能接收和发送

配置 DMA_CTL[1:0]=0x3，同时使能 DMA 发送和接收功能，DMA 从 SRAM 指定位置读取数据，发送到串行接口，同时从串行接口收数据，写到 SRAM 中指定位置。

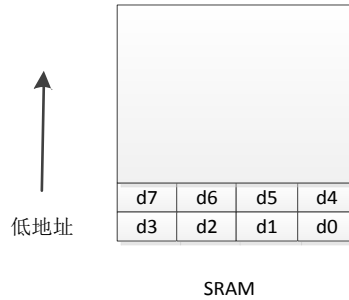
15.4 SRAM 发送 Buffer

15.4.1 起始地址和长度配置

使用该模式前，软件需要在 SRAM 中规划一块 Buffer，用于缓存需要发送的数据。提前在 Buffer 中准备好

待发送的数据；然后配置 DMA_TBADR 选择 DMA 起始地址（按 Byte 配置），配置 DMA_TLEN 选择发送 DMA 数据长度（按 Byte 配置）。

DMA 起始地址和长度根据模块接口数据位宽配置：如果 SPI 接口数据宽度配置为 8bit 模式，则 DMA 起始地址和长度按 Byte 任意配置；如果 SPI 接口数据宽度配置为 16bit 模式，则 DMA 起始地址和长度必须按照 16bit 对齐配置；如果 SPI 接口数据宽度配置为 32bit 模式，则 DMA 起始地址和长度必须按照 32bit 对齐配置。



如上图所示，一个小格表示 1Byte 数据，假设 d0 所在地址为 0，d1 所在地址 1，d2 所在地址为 2...d4 所在地址为 4，d7 所在地址为 7。

如果要发送 d0~d7，则配置 DMA_TBADR=0，DMA_TLEN=8；

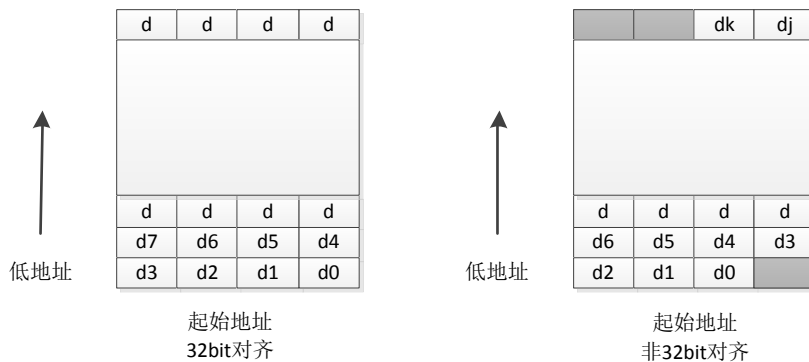
如果要发送 d0~d6，则配置 DMA_TBADR=0，DMA_TLEN=7；

如果要发送 d1~d7，则配置 DMA_TBADR=1，DMA_TLEN=7；

如果要发送 d2~d4，则配置 DMA_TBADR=2，DMA_TLEN=3；

15.4.2 Buffer 中数据发送顺序

如果 SPI 接口数据位宽为 8bit，从 SRAM 中每次读到 4Byte 数据，低地址数据先发送，高地址后发送。

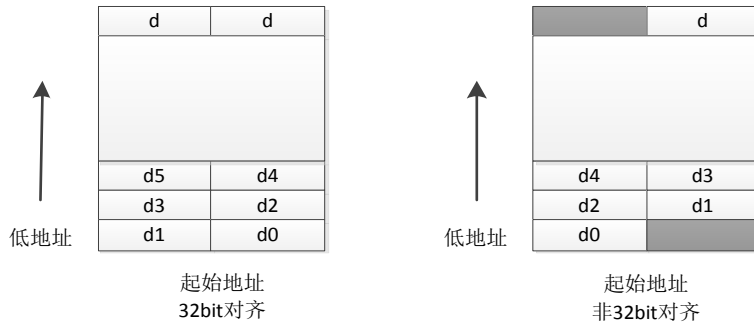


如上图所示，其中一个小格表示 1Byte 数据，d0 所在位置为 Buffer 最低地址。左图为起始地址和长度按 32Bit 对齐，右图起始地址和长度按 8Bit 对齐。

左图：DMA 每次从 SRAM 中读到 32Bit 数据，按最低地址有效数据先发送的顺序，依次发送 d0->d1->d2->d3。接着继续读下一个 32Bit 数据，依次发送 d4->d5->d6->d7。

对于右图，第一次读到的 32Bit 数据中，只有 3Byte 有效数据，所以第一次发送 d0->d1->d2，接着读下一个 32Bit 数据，依次发送 d3->d4->d5->d6。最后一次读到的 4Byte 数据，根据配置，只有低地址的 2Byte 有效，所以，最后一次读到的 32Bit 数据，只发送 dj->dk 两 Byte 数据，然后 DMA 结束。

如果 SPI 接口数据位宽为 16bit，从 SRAM 中每次读到 4Byte 数据，低 16bit 数据线发送，高 16bit 数据后发送。



如上图所示，其中一个小格表示 2Byte 数据，d0 所在位置为 Buffer 最低地址。作图起始地址和长度按 32Bit 对齐，右图起始地址按 16Bit 对齐，长度按 32Bit 对齐。

左图：DMA 每次从 SRAM 中读到 32Bit 数据，按最低地址有效数据先发送顺序，一次发送 d0->d1。接着继续读下一个 32Bit 数据，依次发送 d2->d3。

右图：第一次读到的 32Bit 数据中只有 2Byte 有效数据，所以第一次发送 d0，接着读下一个 32Bit 数据，依次发送 d1->d2。最后一次读到的 4Byte 数据，根据配置，只有低 2Byte 有效，所以，只发送 d，然后 DMA 结束。

对于普通 DMA 模式，只有第一次和最后一次读到的 4Byte 数据中，存在不需要发送的无效数据，其他中间位置的所有 4Byte 数据有效。

15.4.3 UART 接口数据位宽小于 1Byte

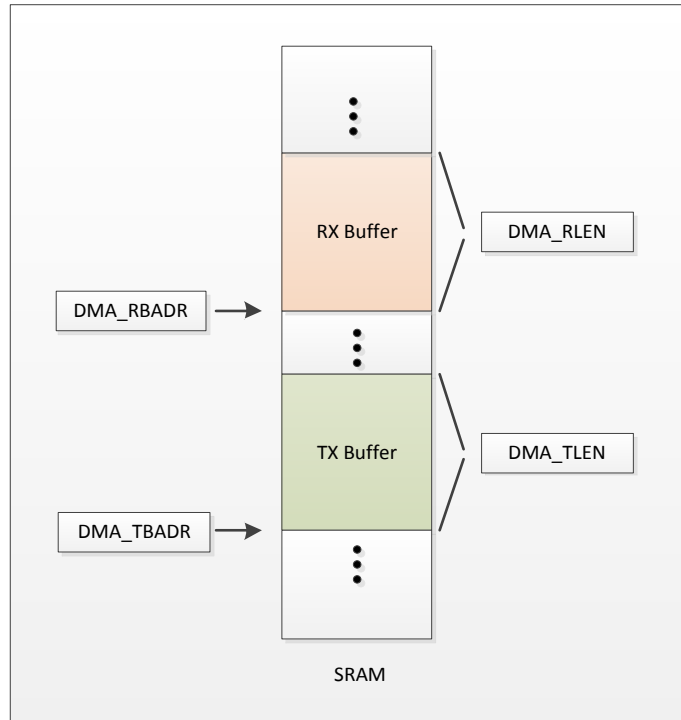
当 UART 接口数据位宽小于 1Byte，把待发送数据存到每 8bit 中的低位。

15.5 SRAM 接收 Buffer

使用该模式前，软件需要在 SRAM 中规划一块 Buffer，用于缓存从串行接口收到的数据。配置 DMA_RBADR 选择 DMA 起始地址（按 Byte 配置），配置 DMA_RLEN 选择发送 DMA 数据长度（按 Byte 配置）。

DMA 起始地址和长度根据模块接口数据位宽配置：如果 SPI 接口数据宽度配置为 8bit 模式，则 DMA 起始地址和长度按 Byte 任意配置；如果 SPI 接口数据宽度配置为 16bit 模式，则 DMA 起始地址和长度必须按照 16bit 对齐配置；如果 SPI 接口数据宽度配置为 32bit 模式，则 DMA 起始地址和长度必须按照 32bit 对齐配置。

DMA 发送和接收地址和长度可任意配置，所以需要软件保证发送和接收 Buffer 在 SRAM 中不会发生重叠。



15.6 单次模式和循环模式

15.6.1 DMA 单次模式

当配置为单次模式的时候，对于 DMA 发送，从 DMA_TBADR 指定地址开始，发送完 DMA_TLEN 指定长度数据之后，发送结束，停止 DMA 发送。

对于 DMA 接收，把接收到的数据存放到 DMA_RBADR 指定地址，存满 DMA_RLEN 指定长度数据之后，停止 DMA 接收。

15.6.2 DMA 循环模式

当配置为循环模式的时候，对于 DMA 发送，从 DMA_TBADR 指定地址开始，发送完 DMA_TLEN 指定长度数据之后，返回 DMA_TBADR，循环发送数据，直到 CPU 配置 DMA_CTL[0]=0 关闭 DMA 发送使能。所以软件需要使用 TX 半满中断，往发送 Buffer 里面更新数据，保证连续传输数据的准确性。

对于 DMA 接收，把接收到的数据存放到 DMA_RBADR 指定地址，存满 DMA_RLEN 指定长度数据之后，返回 DMA_RBADR，循环存储数据，直到 CPU 配置 DMA_CTL[1]=0，停止接收数据。所以软件需要使用 RX 半满中断，及时读走接收 Buffer 里面的数据，保证数据不会被覆盖。

15.7 DMA 中断

一共支持 4 个中断，可独立配置使能：

15.7.1 DMA 发送半空中断

DMA 发送半空中断，通过配置 DMA_IE[0]=1 使能该中断，当发送了 DMA_TLEN 一半长度的数据，产生半满标志 DMA_FLG[0]=1，如果此时配置中断使能，则产生 DMA 发送半空中断。

注意：最小 5Byte 才能产生发送半空中断。DMA 从 SRAM 读取一半数据，则产生发送半空中断，注意此时只是表明 buffer 的一半数据已经被搬运到寄存器，但是并非表明这一半数据都已经通过串行接口发送出去。

15.7.2 DMA 发送全空中断

DMA 发送全空中断，通过配置 $DMA_IE[1]=1$ 使能该中断，当发送了 DMA_TLEN 长度的数据，产生发送全空标志 $DMA_FLG[1]=1$ ，如果此时配置中断使能，则产生 DMA 发送全空中断。

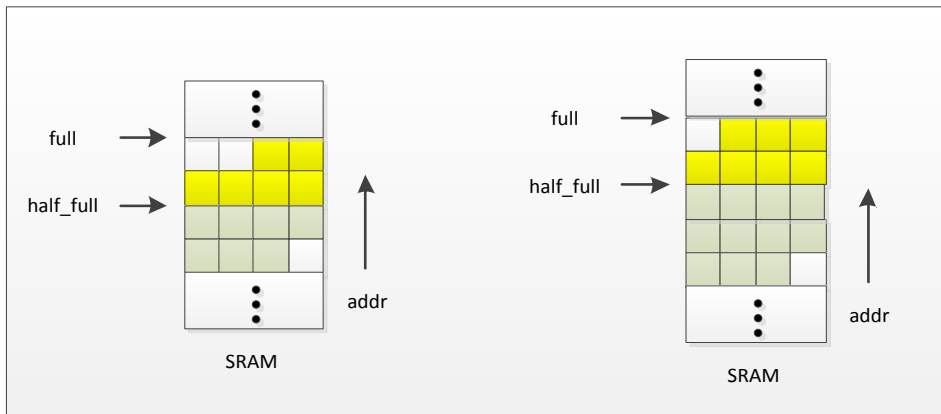
该中断表明整个 buffer 中的数据已经全部通过串口发送出去。

15.7.3 DMA 接收半满中断

DMA 接收半满中断，通过配置 $DMA_IE[2]=1$ 使能该中断，当接收了 DMA_RLEN 一半长度的数据，产生半满标志 $DMA_FLG[2]=1$ ，如果此时配置中断使能，则产生 DMA 接收半满中断。

下文说明的“占用空间”为物理上占用 SRAM 的空间，物理 SRAM 按 32bit 对齐，由于起始地址和长度可配置为非 32bit 对齐，所以占用空间不是直接看配置长度，而是要结合起始地址。

注意：必须占用 3 个 word 以上空间才能产生半满中断。如果占用空间为 2 的倍数，SRAM 中一半 buffer 已经填上数据之后，产生半满中断；如果占用空间非 2 的倍数，则 SRAM 中一半（加一）buffer 已经填上数据之后，产生半满中断。



15.7.4 DMA 接收全满中断

DMA 接收全满中断，通过配置 $DMA_IE[3]=1$ 使能该中断，当接收了 DMA_RLEN 长度的数据，产生全满标志 $DMA_FLG[3]=1$ ，如果此时配置中断使能，则产生 DMA 接收全满中断。

15.7.5 DMA 接收错误中断

如果当前 DMA 数据还没有写入 SRAM，这时接口上又收到新的数据，则会产生接收错误中断，此时数据将丢失，并且 DMA 功能异常，软件必须在中断复位 DMA 功能。

16 UART

16.1 概述

芯片集成了 5 个 UART 接口 UART1-5。UART 用于与外部进行异步串行通信。具有如下特点：

- 全双工；
- 内置波特率发生器，最小波特率支持 300，最大波特率支持 $f_{cpu}/16$ ，支持小数波特率分频；
- 数据位宽支持 5/6/7/8bit
- 停止位可配置成 1 或 2bit；
- 支持多种工作模式
- 一个 38kHz 红外调制时钟发生器；UART1/2/3 均可配置成红外调制输出；
- 串口的输入也可配置为中断输入口，方便实现自动波特率功能和红外唤醒；
- 红外解调由外部器件完成；
- **UART2 支持通用 DMA(新增)；**
- **UART2 支持在线升级（新增），具体实现请参考锐能微例程。**

16.2 寄存器

16.2.1 寄存器列表

模块名	物理地址	映射地址
UART1	0x40004000	0x40004000
UART2（支持通用 DMA）	0x40008000	0x40008000
UART3	0x4000C000	0x4000C000
UART4	0x40018000	0x40018000
UART5	0x4001C000	0x4001C000
寄存器名	地址偏移量	描述
UARTx_CTL	0x0	UARTx 控制寄存器
UARTx_BAUD	0x4	UARTx 波特率配置寄存器
UARTx_STAT	0x8	UARTx 状态指示寄存器
UARTx_TXD	0xC	UARTx 发送数据寄存器
UARTx_RXD	0x10	UARTx 接收数据寄存器
UARTx_FDIV	0x14	UARTx 波特率小数分频配置寄存器
UART2_DMA_CTL(新增)	0x18	UART2 DMA 控制
UART2_DMA_TBADR(新增)	0x1c	UART2 DMA 发送起始地址
UART2_DMA_RBADR(新增)	0x20	UART2 DMA 接收起始地址
UART2_DMA_TLEN(新增)	0x24	UART2 DMA 发送长度
UART2_DMA_RLEN(新增)	0x28	UART2 DMA 接收长度

UART2_DMA_TADR (新增)	0x2c	UART2 当前发送 DMA 地址
UART2_DMA_RADR (新增)	0x30	UART2 当前接收 DMA 地址
UART2_DMA_IE (新增)	0x34	UART2 DMA 中断
UART2_DMA_FLG (新增)	0x38	UART2 DMA 标志
UART2_DMA_TO (新增)	0x3C	UART2 DMA 接收超时配置寄存器

16.2.2 控制寄存器 UARTx_CTL (0x00)

偏移地址 0x0

比特位	名称	描述	读/写标志	复位值
31:13	---	预留	R	0
12	IRSEL	红外调制极性选择位： 0: 正极性，即默认驱动电平为高 1: 负极性，即默认驱动电平为低	R/W	0
11	ILBE	内部环回使能位 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	校验位选择位 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 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	R/W	0

		bit1. 0: 关闭中断 1: 打开中断		
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	UARTx 模块使能位 0: 关闭 1: 打开	R/W	0

16.2.3 波特率配置寄存器 UARTx_BAUD (0x04)

偏移地址 0x4

比特位	名称	描述	读/写标志	复位值
31:12	---	预留	R	0
11:0	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 14.2.6），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 16.384MHz 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_BA UD	UARTx_FD IV			UARTx_BA UD	UARTx_FD IV	
1200	354	1555	- 0.02%	38400	19	2AAB	0.07%
2400	1A9	2AAB	0.22%	56000	11	1249	0.24%
4800	D4	1555	- 0.26%	115200	7	38E4	0.13%
9600	69	2AAB	- 0.26%	256000	3	0	- 0.06%
14400	46	71c	0.07%	750000	0	1762	0.20%
19200	34	1555	- 0.04%	921600	0	71C	0.18%
28800	22	238e	0.02%	100000	0	189	- 0.08%
33600	1d	1e7a	0.03%	---	---	---	---

举例：程序运行在 16.384MHZ 频率下，要配置 uart0 的波特率为 1200bps，只需将 UART0_Baud 寄存器配置为 0x354, UART0_FDIV 寄存器配置为 0x1555。其它对应波特率见表格。

如果程序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。

16.2.4 状态指示寄存器 UARTx_STA (0x08)

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： 0: 不满 1: 满	R	0
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 0: 无错误 1: 有错误	R/W	0
4	FE	帧错误，写 1 清零 0: 无错误 1: 有错误	R/W	0
3	OE	溢出错误，写 1 清零 0: 无错误 1: 有错误	R/W	0
2	PE	校验错误，写 1 清零 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 0: 无接收数据 1: 数据已接收	R/W	0

16.2.5 发送数据寄存器 UARTx_TXD (0xC)

比特位	名称	描述	读/写标志	复位值
31:9	---	预留	R	0
8	UP	用户自定义校验位	R/W	0
7:0	TXDATA	发送数据寄存器	R/W	0

16.2.6 接收数据寄存器 UARTx_RXD (0x10)

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

31:9	---	预留	R	0
8	UP	校验位	R	0
7:0	RXDATA	接收数据寄存器	R	0

16.2.7 波特率小数分频配置寄存器 UARTx_FDIV (0x14)

比特位	名称	描述	读/写标志	复位值
31:14	---	预留	R	0
13:0	FDIV	小数分频系数。计算公式为： $F = \left[\left(\frac{fi}{16 \times fo} - \left[\frac{fi}{16 \times fo} \right] \right) \times 2^{14} + 0.5 \right]$ 其中 fi 为系统时钟，fo 为输出时钟， $\left[\right]$ 为向下求整算符。例如系统时钟为 16.384MHz，输出时钟为 115200Hz，那么： $FDIV = \left[\left(\frac{16384000}{16 \times 115200} - \left[\frac{16384000}{16 \times 115200} \right] \right) \times 2^{14} + 0.5 \right]$ 求得 FDIV=14564。	R/W	0

16.2.8 UART2_DMA_CTL(新增, 0x18)

偏移地址: 0x18; 默认值: 0x0

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

16.2.9 UART2_DMA_TBADR(新增, 0x1C)

偏移地址: 0x1C; 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
15:0	DMA_TBADR	DMA 发送起始地址 (Byte 地址) 最低两位必须配置根据具体接口数据位宽配置。	R/W	0

16.2.10 UART2_DMA_RBADR(新增, 0x20)

偏移地址: 0x20; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:16	Reserved	保留。	R	0
15:0	DMA_RBADR	DMA 接收起始地址 (Byte 地址)	R/W	0

		最低两位必须配置根据具体接口数据位宽配置。		
--	--	-----------------------	--	--

16.2.11 UART2_DMA_TLEN(新增, 0x24)

偏移地址: 0x24; 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
15:0	DMA_TLEN	DMA 发送长度 (Byte 地址) = (n) Byte 最低两位必须配置根据具体接口数据位宽配置。	R/W	0

16.2.12 UART2_DMA_RLEN(新增, 0x28)

偏移地址: 0x28; 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
15:0	DMA_RLEN	DMA 接收长度 (Byte 地址) = (n) Byte 最低两位必须配置根据具体接口数据位宽配置。	R/W	0

16.2.13 UART2_DMA_TADR(新增, 0x2C)

偏移地址: 0x2C; 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
15:0	DMA_TADR	当前发送 DMA 地址 (对应 RAM 地址的低 16 位, 以 Byte 为单 位)	RO	0

16.2.14 UART2_DMA_RADR(新增, 0x30)

偏移地址: 0x30; 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
15:0	DMA_RADR	当前接收 DMA 地址 (对应 RAM 地址的低 16 位, 以 Byte 为单 位)	RO	0

16.2.15 UART2_DMA_IE(新增, 0x34)

偏移地址: 0x34; 默认值: 0x0

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	R/W	0
4	TX_ERR_IE	发送数据错误中断使能	R/W	0
3	RX_FIE	DMA 接收全满中断使能	R/W	0
2	RX_HIE	DMA 接收半满中断使能	R/W	0

1	TX_FIE	DMA 发送全空中断使能	R/W	0
0	TX_HIE	DMA 发送半空中断使能	R/W	0

16.2.16 UART2_DMA_FLG(新增, 0x38)

偏移地址: 0x38; 默认值: 0x0

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, 写“1”清该位	R/WC	0
4	TX_ERR	发送数据错误标志, 写“1”清该位	R/WC	0
3	RX_FDONE	DMA 接收全满中断标志, 写“1”清该位	R/WC	0
2	RX_HDONE	DMA 接收半满中断标志, 写“1”清该位	R/WC	0
1	TX_FDONE	DMA 发送全空中断标志, 写“1”清该位	R/WC	0
0	TX_HDONE	DMA 发送半空中断标志, 写“1”清该位	R/WC	0

16.2.17 UART2_DMA_TO (新增, UART2 模块专有, 0x3C)

偏移地址: 3CH; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:4	Reserved	保留。	R	0
3:0	TIMEOUT_CNT	UART 接收超时配置, 当 UART 超过一定时间没有接收到起始位, DMA 自动结束。n=n 个 UART 数据位宽时间。 配置为 0 无效	RW	0

16.3 UART 数据接收及发送操作步骤

- 1、将系统控制章节模块使能 0 寄存器 MOD0_EN 中对应的 UART 时钟启动, 模块使能。
- 2、配置波特率配置寄存器 UARTx_BAUD, 例如当系统时钟为 3.6864MHz 时, 通讯波特率为 9600, 则波特率配置寄存器可设置为: $UART0 \rightarrow BAUD = 3686400 / (9600 * 16) - 1$;
- 3、配置通讯控制寄存器 UARTx_CTL, 选择数据位、停止位、校验方式及中断使能。
- 4、写 0x3f 清除 UART 状态指示寄存器 (UARTx_STA) 状态。
- 5、配置 UART 中断使能, 打开 UART 中断 NVIC_EnableIRQ(UARTx_IRQn);
- 6、编写中断服务程序, 如 UART0 中断服务程序为:

```
void UART0_HANDLER(void)
{
    u32  status;
    u8   temp;
    status = UART0->STA;

    /* UART error irq */
    if((UART0->CTRL & 0x8) && (status & 0x3c))
    {
        /* Start adding user code. Do not edit comment generated here */
    }
}
```

```
/* receive data complete irq */
if((UART0->CTRL & 0x4) && (status & 0x1))
{
/* Start adding user code. Do not edit comment generated here */
}
/* transmit data complete irq */
if((UART0->CTRL & 0x2) && (status & 0x2))
{
/* Start adding user code. Do not edit comment generated here */
}
}
```

UART 接收、发送、错误中断为同一中断入口，需根据控制寄存器打开的中断使能位及状态标志判断此时为何种中断。

7、处理接收或发送的数据，完成。

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。

17 SPI 1/3/4（支持普通 DMA）

17.1 概述

芯片集成了 3 个具备普通 DMA 的 SPI M/S 接口：SPI1 和 SPI3，SPI4，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。

IO复用关系：详见1.4管脚排列章节和GPIO章节寄存器。中断号分配详见3.2.1章节表 3-1中断/异常向量表及其配置信息。

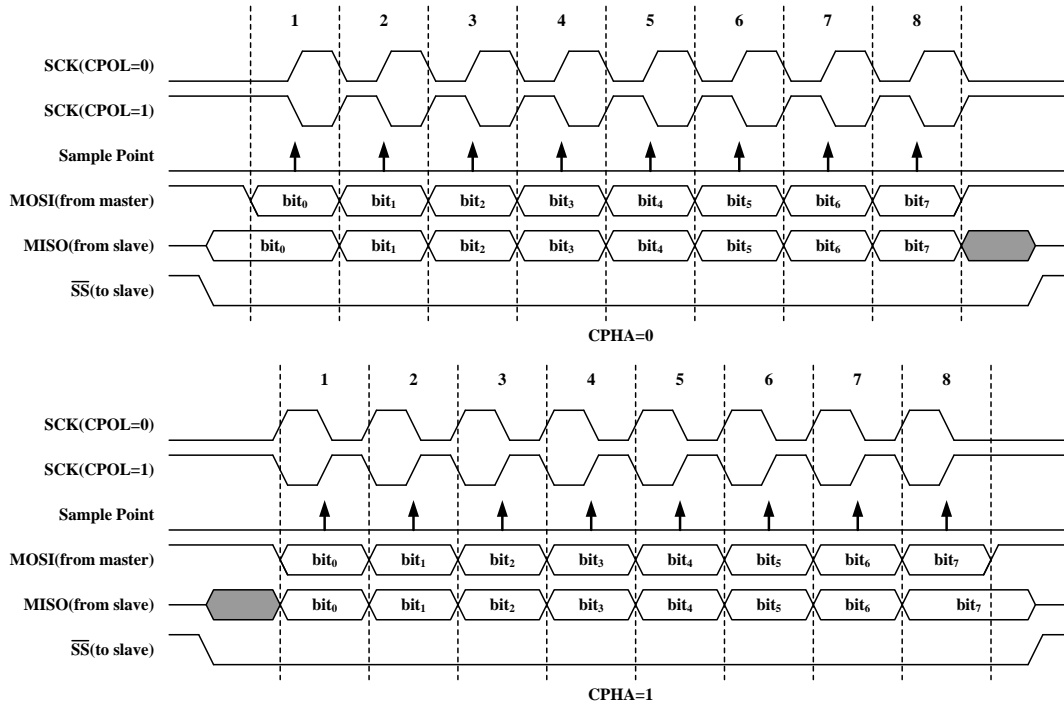
特点如下：

- 支持 SPI 全双工模式，可以编程实现主模式和从模式工作；
- 支持 SPI Clock 的极性和相位设定；
- 双缓冲区，RXDATA 和 TXDATA 独立；
- 支持 LSB 和 MSB 传输模式 8 位，16 位，32 位可配置；
- 主机 256 种波特率可设($f_{cpu}/(2*(1\sim 256))$)，最高 16.384Mhz($f_{cpu}=32.768\text{Mhz}$)；
- 从机最高支持 $f_{cpu}/8$ ，最高 4.096 Mhz($f_{cpu}=32.768\text{Mhz}$)；
- 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；
- 与主机 CPU 的专用功能寄存器接口；
- 支持第 15 章节所述 通用 DMA 功能（新增）

17.2 功能描述

符合标准的 SPI HOST 协议，SPI Clock 通过 CPOL(Clock Polarity)和 CPHA(Clock Phase)参数设置不同的工作方式，其中 CPOL 决定 SPI Clock 的前边沿是上升沿还是下降沿，CPHA 决定 SPI Clock 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：

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



数据传输支持 LSB 和 MSB 两种方向，数据传输大小支持 8/16/32bit 宽度，SPI 时钟源来自 APB 总线时钟，经过一个分频系数后产生 SPI 模块时钟。

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。

写数据冲突：当有 BUF 满时，总线向 SPI 发送写数据的命令，则会将 TXCOLIF 位置 1，同时 SPI 不会响应本次写操作的命令，若 TXCOL_IRQ_EN 中断使能，则会产生中断。

接收数据溢出：在下次完整的接收数据进入移位寄存器之前，没有读取 RXDATA 寄存器，将产生接收数据溢出，则 RXCOLIF 置 1，若 COL_IRQ_EN=1，则会产生中断，同时新的接收数据会保存到接收数据寄存器中，原来没被读走的数据将被覆盖。

传输结束中断：当传输结束时 (sck_end)，若 TR_IRQ_EN=1,则会产生一个中断，同时 TRIF 置 1。

SCSN 模式出错中断：从模式下，SCSN 必须作为输入，在数据传输过程中 SCSN 变高，则 SCSN 模式出错标志置 1；在主模式下，只有使能主模式 SCSN 模式错误检测 (SCSN_EN=1)，同时 SCSN 输入为低，则 SCSN 模式出错标志置 1。一旦 SCSN 模式出错标志为 1，则会终止正在进行的传输同时复位 SPI 模块，如果 ERR_IRQ_EN=1 还会产生一个中断。

采用单次传输完成中断+CSN 拉低拉高中断。软件在 CSN 拉低中断和传输完成中断服务程序中完成 txbuf 的填充，CSN 拉高中断则代表帧结束。实现一次 CSN 拉低过程传输一帧数据。

支持 TX BUFFER 为空时，配置发送上次发送数据，或者寄存器可配的默认值。

通用 DMA 功能见第 15 章节所述

17.3 寄存器

17.3.1 寄存器列表

模块名	物理地址	映射地址
SPI1	0x40048000	0x40048000

SPI3	0x40050000	0x40050000
SPI4	0x40054000	0x40054000
寄存器名 (X=1,3,4)	地址偏移量	描述
SPIX_CTL	0x0	SPI 控制寄存器
SPIX_STAT	0x4	SPI 状态指示寄存器
SPIX_TXDATA	0x8	SPI 数据发送寄存器
SPIX_RXDATA	0xC	SPI 数据接收寄存器
SPIX_TXDFLT	0x10	SPI 发送数据为空时，默认发送数据配置
SPIX_DMA_CTL(新增)	0x14	SPIX DMA 控制
SPIX_DMA_TBADR(新增)	0x18	SPIX DMA 发送起始地址
SPIX_DMA_RBADR(新增)	0x1C	SPIX DMA 接收起始地址
SPIX_DMA_TLEN (新增)	0x20	SPIX DMA 发送长度
SPIX_DMA_RLEN (新增)	0x24	SPIX DMA 接收长度
SPIX_DMA_TADR (新增)	0x28	SPIX 当前发送 DMA 地址
SPIX_DMA_RADR (新增)	0x2C	SPIX 当前接收 DMA 地址
SPIX_DMA_IE (新增)	0x30	SPIX DMA 中断
SPIX_DMA_FLG (新增)	0x34	SPIX DMA 标志

17.3.2 SPIX 控制寄存器 SPI_CTL (0x00)

偏移地址=0x0

比特位	名称	描述	读/写标志	复位值
31:26	---	预留	R	0
25	SCSN_POS_IRQ EN	SCSN_POS 中断使能 0, 关闭中断 1, 打开中断	RW	0
24	SCSN_NEG_IRQ EN	SCSN_NEG 中断使能 0, 关闭中断 1, 打开中断	RW	0
23	TXEMPT_IRQE N	TXEMPT 中断使能 0, 关闭中断	RW	0

		1, 打开中断		
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	R/W	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 $SCK = APBClock / (2 * (CLKDIV + 1))$	R/W	000
11:10	WIDTH	数据宽度选择 00: 8bit 01: 16bit 10: 32bit 11: 预留, 8bit	R/W	00
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 选择方式 0: MSB 先传输 1: LSB 先传输	R/W	0
5	TXCOL_IRQ_EN	数据冲突中断使能信号 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 模式错误中断使能信号 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	数据传输中断使能信号 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	R/W	0

		0: 关闭 SPI 接口 1: 打开 SPI 接口		
--	--	------------------------------	--	--

17.3.3 SPIX 状态标识寄存器 SPI_STAT (0x04)

偏移地址=0x4

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7	SCSN_PO S	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_N EG	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	TXBUF_ VLD	TXBUF 非空标志位。 0: TXBUF 为空, 软件可以向 TXBUF 中写入新数据 1: TXBUF 非空, 软件不可向 TXBUF 中写入新数据	R/W	0
3	TXCOLIF	写数据冲突标志位。当有 BUF 满时, 总线向 SPI 发送写数据的命令, 则会将该位置 1, 同时不会响应本次写操作的命令, 若 TXCOL_IRQ_EN 中断使能, 则会产生中断, 该位写 1 清零。 0: 没有发生写数据冲突中断 1: 产生写数据冲突中断	R/W	0
2	RXCOLIF	接收数据溢出: 在下次完整的接收数据进入移位寄存器之前, 没有读取 RXDATA 寄存器, 将产生接收数据溢出, 则 RXCOLIF 置 1, 若 COL_IRQ_EN=1, 则会产生中断, 同时新接收的数据会覆盖之前的数据, 该位写 1 清零。 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	R/W	0

		1: 产生数据发送中断, 发送数据寄存器为空		
--	--	------------------------	--	--

17.3.4 SPIX 数据发送寄存器 SPI_TXDATA (0x08)

偏移地址=0x8

比特位	名称	描述	读/写标志	复位值
31:0	TXDATA	数据发送寄存器	R/W	0

17.3.5 SPIX 数据接收寄存器 SPI_RXDATA (0x0C)

偏移地址=0xC

比特位	名称	描述	读/写标志	复位值
31:0	RXDATA	数据接收寄存器	R	0

17.3.6 SPIX 默认发送数据寄存器 SPI_TXDFLT (0x10)

偏移地址=0x10

比特位	名称	描述	读/写	复位值
31:0	TXDFLT	默认发送数据寄存器	R/W	0

17.3.7 SPIX_DMA_CTL(新增, 0x14)

偏移地址: 0x14; 默认值: 0x0

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

17.3.8 SPIX_DMA_TBADR(新增, 0x18)

偏移地址: 0x18; 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
15:0	DMA_TBADR	DMA 发送起始地址 (对应 RAM 地址的低 16 位, 以 Byte 为单 位) 最低两位必须根据具体接口数据位宽配置。	R/W	0

17.3.9 SPIX_DMA_RBADR(新增, 0x1C)

偏移地址: 0x1C; 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
15:0	DMA_RBADR	DMA 接收起始地址（对应 RAM 地址的低 16 位，以 Byte 为单 位 ） 最低两位必须根据具体接口数据位宽配置。	R/W	0

17.3.10 SPIX_DMA_TLEN(新增, 0x20)

偏移地址: 0x20; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:16	Reserved	保留。	R	0
15:0	DMA_TLEN	DMA 发送长度 (Byte 地址) = (n) Byte 最低两位必须根据具体接口数据位宽配置。	R/W	0

17.3.11 SPIX_DMA_RLEN(新增, 0x24)

偏移地址: 0x24; 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
15:0	DMA_RLEN	DMA 接收长度 (Byte 地址) = (n) Byte 最低两位必须根据具体接口数据位宽配置。	R/W	0

17.3.12 SPIX_DMA_TADR(新增, 0x28)

偏移地址: 0x28; 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
15:0	DMA_TADR	当前发送 DMA 地址（对应 RAM 地址的低 16 位，以 Byte 为单 位 ）	RO	0

17.3.13 SPIX_DMA_RADR(新增, 0x2C)

偏移地址: 0x2C; 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
15:0	DMA_RADR	当前接收 DMA 地址（对应 RAM 地址的低 16 位，以 Byte 为单 位 ）	RO	0

17.3.14 SPIX_DMA_IE(新增, 0x30)

偏移地址: 0x30; 默认值: 0x0

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	R/W	0
4	TX_ERR_IE	发送数据错误中断使能	R/W	0
3	RX_FIE	DMA 接收全满中断使能	R/W	0
2	RX_HIE	DMA 接收半满中断使能	R/W	0
1	TX_FIE	DMA 发送全空中断使能	R/W	0
0	TX_HIE	DMA 发送半空中断使能	R/W	0

17.3.15 SPIX_DMA_FLG(新增, 0x34)

偏移地址: 0x34; 默认值: 0x0

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, 写“1”清该位	R/WC	0
4	TX_ERR	发送数据错误标志, 写“1”清该位	R/WC	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

18 高速 SPIS（支持普通 DMA）

18.1 概述

芯片集成了一个高速 SPIS（SPI2）接口，支持 SPI 全双工模式，只支持从模式，可用于管理芯对计量芯的访问；SPI 时钟频率要小于系统主频，SPIS 时钟频率最大支持 10Mhz。

中断号：IRQ27；IO 复用关系：详见 1.4 管脚排列章节和 GPIO 章节 PCA0/PCA1 寄存器特点如下：

- 只支持从模式工作；
- 支持 SPI 全双工模式；
- 从机频率须小于系统主时钟；且小于等于 10MHz。
- SPI 时钟极性只支持 CPOL=0，即在空闲状态时被设置为低电平
- SPI 时钟相位只支持 CPHA=1，即前边沿是上升沿，数据建立；后边沿是下降沿，数据采样
- SPI 帧结构只支持 8bits；
- 双缓冲区，RxFIFO 和 TxFIFO 独立，宽度均为 8bit，深度 32 级；
- 只支持 MSB 传输模式；
- SCSN、SCLK、SDI 支持输入滤波，分别 20ns/10ns，5ns，5ns；可配置有无，寄存器放在 GPIO 中。
- 支持接收缓存：非空（空）、半满、满、满溢出状态及中断
- 支持发送缓存：不满（满）、半空、空、空溢出状态及中断
- **支持第 15 章节所述 通用 DMA 功能（新增）**

18.2 功能描述

该从机 SPI 接口符合标准的 SPI HOST 协议；

只支持从模式，输入管脚 SCSN、SCK、SDI，输出管脚 SDO；

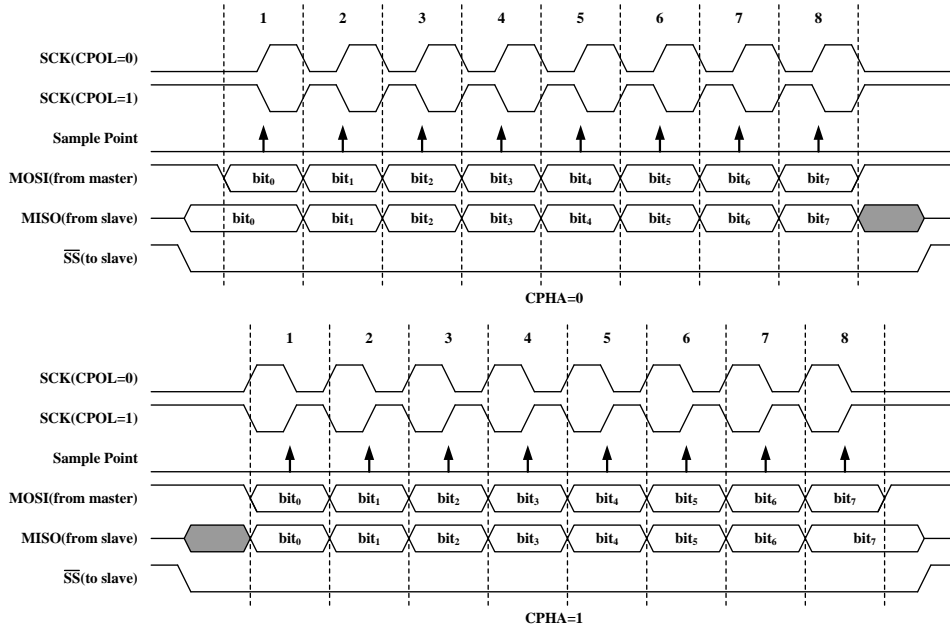
支持全双工模式，SCSN 有效，且接收到主机发过来的 SCK 后，该从机在 SCK 下沿接收 SDI 数据，同时在 SCK 上沿通过 SDO 发送数据。

SPI Clock 时钟极性 CPOL(Clock Polarity)和时钟相位 CPHA(Clock Phase)只支持 CPOL/CPHA=0/1 模式。

其中 CPOL 决定 SPI Clock 的前边沿是上升沿还是下降沿，CPHA 决定 SPI Clock 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 20-1：

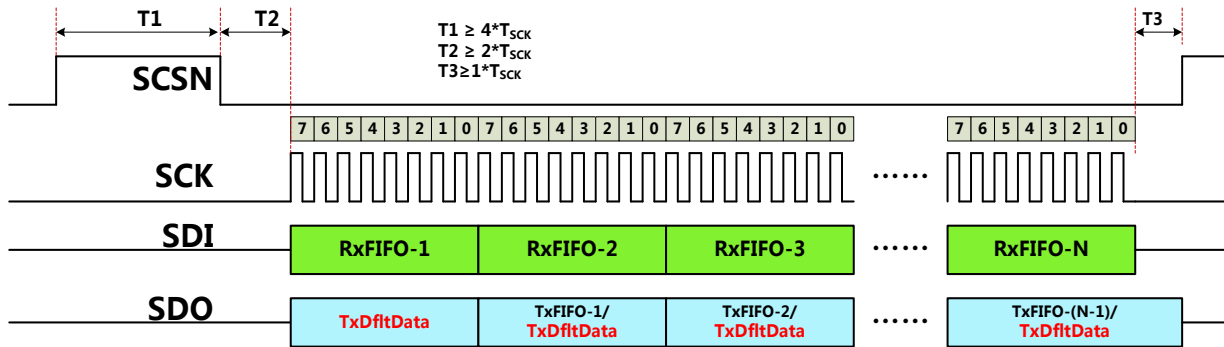
表 20-1 SPI 工作模式列表

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



数据传输只支持 MSB，不支持 LSB。数据传输大小支持 8bit 宽度。SPI 时钟源自外部主机；

18.3 时序描述



关于 SCSN，SCSN 下降沿表示一次 SPI 传输的开始。上升沿表示一次 SPI 传输的结束；

无效时间 T1 至少大于 4 个 T_{SCK} ；T2 至少大于 2 个 T_{SCK} ；T3 至少大于 1 个 T_{SCK} ；

关于接收：

关于发送：

- 1、SCSN 有效后，**从机从 SDO 发送的第一个字节来自“发送缺省数据寄存器”（SPIS_TXDFLT，CPU 可配）（重要信息，请注意）**。即 SCSN 有效期间，若总共发送了 N 个字节数据，那么其中第一个字节永远是无效的数据；

- 2、随后的发送的字节从 Tx FIFO 中读取。

若 Tx FIFO 空，此时会发出空溢出中断，但是 SPI 从机在总线上发送“发送缺省数据寄存器”的值（若 TXDFLT_EN=1）或者重复发送上一帧发送的值（若 TXDFLT_EN=1）；

若 Tx FIFO 不空，则 SPI 从机在总线侧发送 Tx FIFO 中的值；

通用 DMA 功能见第 15 章节所述

18.4 寄存器

18.4.1 寄存器列表

模块名	物理地址	映射地址
SPIS	0x5000C000	0x5000C000
寄存器名	地址偏移量	描述
SPIS_CTL	0x0	SPIS 控制寄存器
SPIS_STIF	0x4	SPIS 状态和中断原始信息寄存器
SPIS_STIE	0x8	SPIS 状态中断使能寄存器
SPIS_STIFE	0xC	SPIS 状态使能状态和中断信息寄存器
SPIS_RXDATA	0x10	SPI 接收数据缓存窗口寄存器
SPIS_TXDATA	0x14	SPIS 发送数据缓存窗口寄存器
SPIS_TXDFLT	0x18	SPIS 发送缺省数据寄存器
SPIS_FFCNT	0x20	SPIS FIFO 指针寄存器
SPIS_FFCLR	0x24	SPIS 接收和发送 FIFO 清零寄存器
SPIX_DMA_CTL(新增)	0X28	SPIS DMA 控制
SPIX_DMA_TBADR(新增)	0X2C	SPIS DMA 发送起始地址
SPIX_DMA_RBADR(新增)	0X30	SPISDMA 接收起始地址
SPIX_DMA_TLEN (新增)	0X34	SPIS DMA 发送长度
SPIX_DMA_RLEN (新增)	0X38	SPIS DMA 接收长度
SPIX_DMA_TADR (新增)	0X3C	SPIS 当前发送 DMA 地址
SPIX_DMA_RADR (新增)	0X40	SPIS 当前接收 DMA 地址
SPIX_DMA_IE (新增)	0X44	SPIS DMA 中断
SPIX_DMA_FLG (新增)	0X48	SPIS DMA 标志

18.4.2 SPIS_CTL (0x00)

SPIS 控制寄存器

偏移地址=0x0

比特位	名称	描述	读/写标志	复位值
31:10	---	预留	R	0
9:8	TRD_WIDTH	数据宽度选择, 只支持 8bit 00: 8bit 01: 16bit 10: 32bit 11: 预留, 8bit	R	00
7	CPHA	时钟相位选择, 固定为 1 0: 前边沿采样数据 1: 前边沿建立数据	R	1
6	CPOL	时钟极性选择, 固定为 0 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	0
5	LMSB	LSB/MSB 选择方式, 固定为 0 0: MSB 先传输 1: LSB 先传输	R	0
4	MAST/SLAV	主从选择, 固定为 0 1: MASTER 0: SLAVE	R	0
3:2	---	保留	R	0
1	TXDFLT_EN	发送 TXFIFO 空时, 使能发送 SPIS_TXDFLT 寄存器数据。 1: 发送 SPIS_TXDFLT 寄存器值 0: 发送上次发送的值 备注: 在主机发起传输后, 不论 TXDFLT_EN 是否设置, 从机发送的第一个字节就是 SPIS_TXDFLT 的值。	R/W	0
0	EN	使能信号 0: 关闭 SPI 接口 1: 打开 SPI 接口	R/W	0

18.4.3 SPIS_STIF (0x04)

SPIS 状态和中断标志寄存器

偏移地址=0x4

该寄存器包含中断标志信息和状态信息。中断标志由事件触发, 中断标志可写 1 清零;

状态信息只读, 若状态信息在 SPIS_STIE 对应的使能位有效时, 也会产生中断, 状态信息不能写清零。但是 CPU 可以进行相应的操作, 改变 TXFIFO 和 RXFIFO 的状态, 相应的状态信息位会跟随变化。比如若 TE_ST=1, 表示 TxFIFO 空, 若此时 CPU 向 SPIS_TXDATA 至少写入一个字节数据, TxFIFO 状态变为非空, 那么 TE_ST 会变为 0。

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

31:5	---	预留	R	0
11	TEE_IF	发送缓存空溢出标志；发送缓存空而且 SPI 从机又有发送动作；会触发该中断 中断信息，写 1 清零； 此时会发送根据 TXDDFLT_EN 的配置选择发送寄存器 SPIS_TXDDFLT 值或上次发送的值；	R/W	0
10	TE_ST	发送缓存空；1，表示发送缓存空；0，表示非空； TXFIFO 状态信息	R	1
9	THE_ST	发送缓存半空；1，表示发送缓存内可发送数据小于半数；0，表示发送缓存可发送数据超过半数； TXFIFO 状态信息；	R	1
8	TNF_ST	发送缓存不满；1，表示不满，；0，表示满； TXFIFO 状态信息； 若不满，CPU 可以写入；若满，CPU 不要写入如果强制写入，写无效；	R	1
7	RFE_IF	接收缓存满溢出，接收缓存已满且此时 SPI 又收到数据； 中断信息，写 1 清零	R/W	0
6	RF_ST	接收缓存满，1，表示满；0，表示不满； RXFIFO 状态信息；	R	0
5	RHF_ST	接收缓存至少半满，1，表示至少半满；0，表示少于半满； RXFIFO 状态信息；	R	0
4	RNE_ST	接收缓存非空，1，表示非空；0，表示空； RXFIFO 状态信息；若非空，CPU 就可以读 FIFO；	R	0
3	SCSN_NEG_IF	SCSN 下降沿中断，SCSN 从高变低会触发该中断，表示一次 SPI 传输开始； 中断信息，写 1 清零。	R/W	0
2	SCSN_POS_IF	SCSN 上升沿中断，SCSN 从低变高会触发该中断表示一次 SPI 传输结束； 中断信息，写 1 清零。	R/W	0
1	SCSN_ERR_IF	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；该位写 1 清零。 0：没有模式冲突中断 1：产生模式冲突中断 中断信息，写 1 清零	R/W	0
0	TRANS_IF	数据传输中断标识位，传输达到 8Bit 数据，该位置 1； 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 中断信息，写 1 清零	R/W	0

18.4.4 SPIS_STIE (0x08)

SPIS 状态和中断标识使能寄存器

偏移地址=0x8

比特位	名称	描述	读/写标志	复位值
31:5	---	预留	R	0
11	TEE_IE	发送缓存空溢出中断标志使能位；	R/W	0
10	TE_IE	发送缓存空状态使能位；使能后，当 TXFIFO 空时，会产生中断；	R/W	0
9	THE_IE	发送缓存半空状态使能位	R/W	0
8	TNF_IE	发送缓存不满状态使能位。	R/W	0
7	RFE_IE	接收缓存满溢出中断使能位。	R/W	0
6	RF_IE	接收缓存满状态使能位	R/W	0
5	RHF_IE	接收缓存至少半满状态使能位	R/W	0
4	RNE_IE	接收缓存非空状态使能位	R/W	0
3	SCSN_NEG_IE	SCSN 下降沿中断使能位。	R/W	0
2	SCSN_POS_IE	SCSN 上升沿中断使能位。	R/W	0
1	SCSN_ERR_IE	SCSN 模式错误中断使能位。	R/W	0
0	TRANS_IE	数据帧（8bits）传输完成中断标使能位。	R/W	0

18.4.5 SPIS_STIFE (0x0C)

SPIS 使能后状态和中断标志寄存器

偏移地址=0xC

该寄存器存放使能后的 SPIS_STIF 值，全部只读。

比特位	名称	描述	读/写标志	复位值
31:5	---	预留	R	0
11	TEE_IFE	发送缓存空溢出中断标志使能后的值；	R	0
10	TE_STE	发送缓存空状态使能后的值	R	0
9	THE_STE	发送缓存半空状态使能后的值	R	0
8	TNF_STE	发送缓存不满状态使能后的值	R	0
7	RFE_IFE	接收缓存满溢出中断使能后的值	R	0
6	RF_STE	接收缓存满状态使能后的值	R	0
5	RHF_STE	接收缓存至少半满状态使能后的值	R	0
4	RNE_STE	接收缓存非空状态使能后的值	R	0
3	SCSN_NEG_IFE	SCSN 下降沿中断使能后的值	R	0
2	SCSN_POS_IFE	SCSN 上升沿中断使能后的值	R	0
1	SCSN_ERR_IFE	SCSN 模式错误中断使能后的值	R	0
0	TRANS_IFE	数据帧（8bits）传输完成中断标使能后的值	R	0

18.4.6 SPIS_RXDATA (0x10)

SPIS 数据接收寄存器

偏移地址=0x10

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

31:8	---	保留	R	0
7:0	RXDATA	数据接收寄存器，只读寄存器，低 8 位有效。 若 RXFIFO 非空，每次读，可以从 RXFIFO 中读取一个 BYTE 数据，同时 RXFIFO 指针加 1； 若 RXFIF 空，读到的数无效，也不会引起 RXFIFO 指针增加。 写无效；	R	8'H00

18.4.7 SPIS_TXDATA (0x14)

SPIS 数据发送寄存器

偏移地址=0x14

比特位	名称	描述	读/写标志	复位值
31:8	---	保留	R	0
7:0	TXDATA	数据发送寄存器，低 8 位有效。 若 TXFIFO 不满，写该寄存器，数据将压入 TXFIFO，同时 TXFIFO 指针加一。 若 TXFIFO 满，写该寄存器，数据不会写进 TXFIFO，TXFIFO 指针不增加； 读该寄存器读到的是寄存器值，寄存器缺省值或者上次写入的值。	R/W	8'H00

18.4.8 SPIS_TXDFLT (0x18)

SPIS 发送缺省数据寄存器

偏移地址=0x18

比特位	名称	描述	读/写标志	复位值
31:8	---	保留	R	0
7:0	SPIS_TXDFLT	SPI 发送缺省数据寄存器； 1、SPI 传输开始后从机发送的第一字节来自该寄存器； 2、SPI 传输过程中，若 TXFIFO 空，若 SPIS_CTL.TXDFLT_EN=1，则 SPI 从机发送该寄存器的值；若 SPIS_CTL.TXDFLT_EN=0，则 SPI 从机发送上次发送的值；	R/W	8'H00

18.4.9 SPIS_FFCNT

SPIS FIFO 指针寄存器

Offset=0x20

比特位	名称	描述	读/写标志	复位值
31:27	TXFIFO_RP	TXFIFO 读指针	R	0
26:22	TXFIFO_WP	TXFIFO 写指针	R	0
21:16	TXFIFO_VDCNT	TXFIFO 有效数据个数	R	0
15:11	RXFIFO_RP	RXFIFO 读指针	R	0
10:6	RXFIFO_WP	RXFIFO 写指针	R	0
5:0	RXFIFO_VDCNT	RXFIFO 有效数据个数	R/W	0

18.4.10 SPIS_FFCLR (0x24)

SPIS FIFO 清零命令寄存器

Offset=0x24

比特位	名称	描述	读/写标志	复位值
15:8	TXFIFO_CLEAR	TXFIFO 清零命令寄存器 在该寄存器写入 0XC4, 会清零 TXFIFO(读写指针归零, 内部有效数据归零, 满标志归零, 空标志置起)	W	0
7:0	RXFIFO_CLEAR	RXFIFO 清零命令寄存器 在该寄存器写入 0XC3, 会清零 RXFIFO(读写指针归零, 内部有效数据归零, 满标志归零, 空标志置起)	W	0

18.4.11 SPIS_DMA_CTL(新增, 0x28)

偏移地址: 0x28; 默认值: 0x0

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

18.4.12 SPIS_DMA_TBADR(新增, 0x2C)

偏移地址: 0x2C; 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
15:0	DMA_TBADR	DMA 发送起始地址 (对应 RAM 地址的低 16 位, 以 Byte 为单 位) 最低两位必须配置根据具体接口数据位宽配置。	R/W	0

18.4.13 SPIS_DMA_RBADR(新增, 0x30)

偏移地址: 0x30; 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
15:0	DMA_RBADR	DMA 接收起始地址 (对应 RAM 地址的低 16 位, 以 Byte 为单 位) 最低两位必须配置根据具体接口数据位宽配置。	R/W	0

18.4.14 SPIS_DMA_TLEN(新增, 0x34)

偏移地址: 0x34; 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
15:0	DMA_TLEN	DMA 发送长度 (Byte 地址) = (n) Byte 最低两位必须配置根据具体接口数据位宽配置。	R/W	0

18.4.15 SPIS_DMA_RLEN(新增, 0x38)

偏移地址: 0x38; 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
15:0	DMA_RLEN	DMA 接收长度 (Byte 地址) = (n) Byte 最低两位必须配置根据具体接口数据位宽配置。	R/W	0

18.4.16 SPIS_DMA_TADR(新增, 0x3C)

偏移地址: 0x3C; 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
15:0	DMA_TADR	当前发送 DMA 地址 (对应 RAM 地址的低 16 位, 以 Byte 为单 位)	RO	0

18.4.17 SPIS_DMA_RADR(新增, 0x40)

偏移地址: 0x40; 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
15:0	DMA_RADR	当前接收 DMA 地址 (对应 RAM 地址的低 16 位, 以 Byte 为单 位)	RO	0

18.4.18 SPIS_DMA_IE(新增, 0x44)

偏移地址: 0x44; 默认值: 0x0

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	R/W	0
4	TX_ERR_IE	发送数据错误中断使能	R/W	0
3	RX_FIE	DMA 接收全满中断使能	R/W	0
2	RX_HIE	DMA 接收半满中断使能	R/W	0
1	TX_FIE	DMA 发送全空中断使能	R/W	0
0	TX_HIE	DMA 发送半空中断使能	R/W	0

18.4.19 SPIS_DMA_FLG(新增, 0x48)

偏移地址: 0x48; 默认值: 0x0

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, 写“1”清该位	R/WC	0
4	TX_ERR	发送数据错误标志, 写“1”清该位	R/WC	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

19 SPI0(支持专用 DMA)

19.1 SPI0 概述

除了 3 个具备普通 DMA 的 SPI M/S 接口：SPI1 和 SPI3，SPI4，还支持具备专用 DMA 功能的 SPI M/S 接口：SPI0。SPI0 可支持 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。

IO复用关系：详见1.4管脚排列章节和GPIO章节寄存器。中断号分配详见3.2.1章节表 3-1中断/异常向量表及其配置信息。

特点如下：

- 支持 SPI 全双工模式，可以编程实现主模式和从模式工作；
- 支持 SPI Clock 的极性和相位设定；
- 双缓冲区，RXDATA 和 TXDATA 独立；
- 支持 LSB 和 MSB 传输模式 8 位，16 位，32 位可配置；
- 主机 256 种波特率可设($f_{cpu}/(2*(1\sim 256))$)，最高 16.384Mhz($f_{cpu}=32.768\text{Mhz}$)；
- 从机最高支持 $f_{cpu}/8$ ，最高 4.096 Mhz($f_{cpu}=32.768\text{Mhz}$)；
- 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；
- 与主机 CPU 的专用功能寄存器接口；
- 支持专用 DMA 功能（新增）

其中专用 DMA 包括三种模式：普通 SPI DMA 模式、专用 SPI DMA 模式和 CRC DMA 模式。普通 SPI DMA 模式把连续存放在 SRAM 中的一片数据，通过 SPI 接口发送出去；专用 DMA 模式只支持特殊数据格式的 DMA 传输，主要用来加速负荷识别的波形传输；CRC DMA 模式把连续存放在 SRAM 中的一片数据，做校验计算(CRC 或者校验和)，校验结果 CPU 可读。

19.2 普通 DMA 传输

软件在 SRAM 中规划一片区域，作为待发送数据的 Buffer，把待发送数据存放到 Buffer 中，然后配置 SPI DMA 模块寄存器，配置 Buffer 的起始地址和长度，然后启动 SPI TX DMA。硬件通过 DMA 总线从 SRAM 读数据，发送到 SPI 总线。

19.2.1 特点

- 支持 $n*\text{Byte}$ 位数据传输，其中 n 为任意正整数；
- SPI 接口数据宽度可选择 8bit、16bit、32bit；
- 当 SPI 接口数据宽度选择 8bit 模式，DMA 起始地址和长度按 Byte 任意配置；
- 当 SPI 接口数据宽度选择 16bit 模式，DMA 起始地址和长度最低一位配置无效；
- 当 SPI 接口数据宽度选择 32bit 模式，DMA 起始地址和长度最低两位配置无效；
- 支持 DMA buffer 发送完成中断；

19.2.2 详细说明

19.2.2.1 模式选择

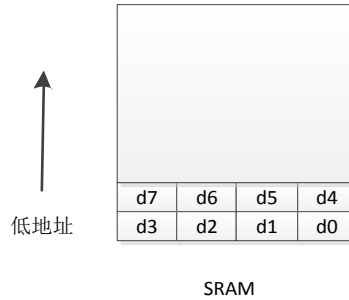
CPU 先配置 SPI 接口基本模式，然后配置 SPI_DMA_CTRL[3:2]=0，选择普通 DMA 模式，配置 SPI_DMA_CTRL[0]=1，使能 SPI TX DMA。

19.2.2.2 SPI DMA Buffer

使用该模式前，软件需要在 SRAM 中规划一块 Buffer，用于缓存需要发送的数据。提前在 Buffer 中准备好待发送的数据；然后配置 SPI_DMA_BADR 选择 DMA 起始地址(按 Byte 配置)，配置 SPI_DMA_LEN 选择 DMA 数据长度（按 Byte 配置）。

如果 SPI 接口数据宽度配置为 8bit 模式，则 DMA 起始地址和长度按 Byte 任意配置；如果 SPI 接口数据宽度配置为 16bit 模式，则 DMA 起始地址和长度必须按照 16bit 对齐配置；如果 SPI 接口数据宽度配置为 32bit 模式，则 DMA 起始地址和长度必须按照 32bit 对齐配置。

起始地址和长度配置

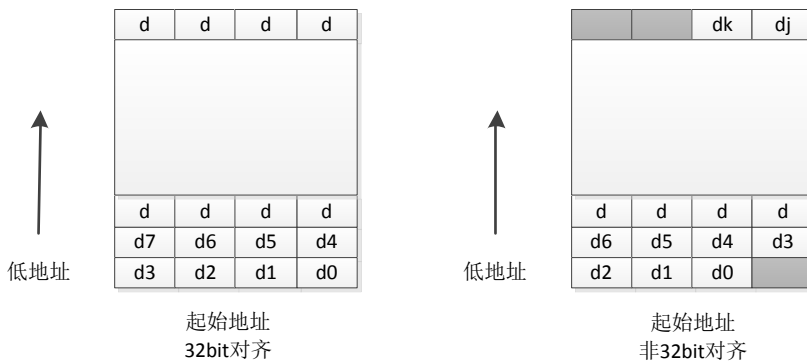


如上图所示，一个小格表示 1Byte 数据，假设 d0 所在地址为 0，d1 所在地址为 1，d2 所在地址为 2...d4 所在地址为 4，d7 所在地址为 7。

- 如果要发送 d0~d7，则配置 SPI_DMA_BADR=0，SPI_DMA_LEN=8；
- 如果要发送 d0~d6，则配置 SPI_DMA_BADR=0，SPI_DMA_LEN=7；
- 如果要发送 d1~d7，则配置 SPI_DMA_BADR=1，SPI_DMA_LEN=7；
- 如果要发送 d2~d4，则配置 SPI_DMA_BADR=2，SPI_DMA_LEN=3；

19.2.2.3 Buffer 中数据发送顺序

如果 SPI 接口数据位宽为 8bit，从 SRAM 中每次读到 4Byte 数据，低地址数据先发送，高地址后发送。

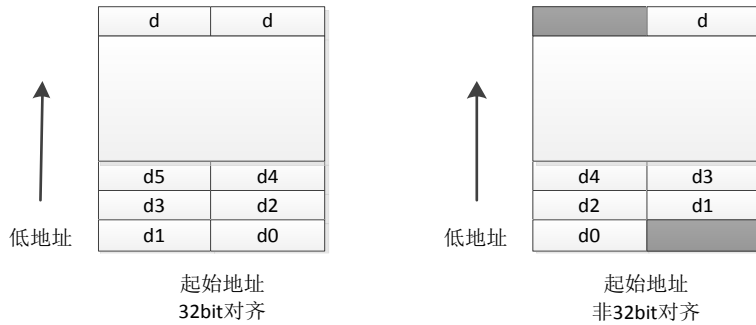


如上图所示，其中一个小格表示 1Byte 数据，d0 所在位置为 Buffer 最低地址。左图为起始地址和长度按 32Bit 对齐，右图起始地址和长度按 8Bit 对齐。

左图：DMA 每次从 SRAM 中读到 32Bit 数据，按最低地址有效数据先发送的顺序，依次发送 d0->d1->d2->d3。接着继续读下一个 32Bit 数据，依次发送 d4->d5->d6->d7。

对于右图，第一次读到的 32Bit 数据中，只有 3Byte 有效数据，所以第一次发送 d0->d1->d2，接着读下一个 32Bit 数据，依次发送 d3->d4->d5->d6。最后一次读到的 4Byte 数据，根据配置，只有低地址的 2Byte 有效，所以，最后一次读到的 32Bit 数据，只发送 dj->dk 两 Byte 数据，然后 DMA 结束。

如果 SPI 接口数据位宽为 16bit，从 SRAM 中每次读到 4Byte 数据，低 16bit 数据线发送，高 16bit 数据后发送。



如上图所示，其中一个小格表示 2Byte 数据，d0 所在位置为 Buffer 最低地址。作图起始地址和长度按 32Bit 对齐，右图起始地址按 16Bit 对齐，长度按 32Bit 对齐。

左图：DMA 每次从 SRAM 中读到 32Bit 数据，按最低地址有效数据先发送顺序，一次发送 d0->d1。接着继续读下一个 32Bit 数据，依次发送 d2->d3。

右图：第一次读到的 32Bit 数据中只有 2Byte 有效数据，所以第一次发送 d0，接着读下一个 32Bit 数据，依次发送 d1->d2。最后一次读到的 4Byte 数据，根据配置，只有低 2Byte 有效，所以，只发送 d，然后 DMA 结束。

对于普通 DMA 模式，只有第一次和最后一次读到的 4Byte 数据中，存在不需要发送的无效数据，其他中间位置的所有 4Byte 数据有效。

19.2.2.4 DMA 中断

启动 DMA 之后，硬件自动从 SRAM 中读取数据，发送到 SPI 总线，当发送完指定长度数据后，SPI 停止，置位 SPI_DMA_FLAG[0]，表示 DMA 发送完成。如果配置 SPI_DMA_IE[0]=1，则会产生中断。

19.3 专用 DMA 模式

该模式下，DMA 从 SRAM 指定位置取数据，每次读取 4Byte 数据，对最高地址的 1Byte 数据做解析，判断低 3Byte 数据是否需要发送以及是否做校验和或 CRC 运算，把需要发送的数据存入模块内部发送缓存，按顺序发送，把需要校验数据存入模块内部数据校验缓存，等待做校验计算。

19.3.1 特点

- 该模式下 SPI 接口数据宽度必须配置为 8bit；
- DMA 起始地址和长度按 32Bit 对齐，低两位配置无效，固定锁到 0；
- 支持 DMA buffer 发送完成中断；
- 支持 SPI DMA 过程，自动算 CRC ($y=x^{16}+x^{12}+x^5+1$) 或校验和 (1Byte)；
- 支持 SPI 发送的过程中，自动发送校验和或 CRC；

19.3.2 详细描述

19.3.2.1 模式选择

CPU 先配置 SPI 接口基本模式，然后配置 SPI_DMA_CTRL[3:2]=1，选择专用 DMA 模式，配置 SPI_DMA_CTRL[0]=1，使能 SPI TX DMA。

19.3.2.2 DMA Buffer

使用该模式前，软件需要在 SRAM 中规划一块 Buffer，用于缓存需要发送的数据。提前在 Buffer 中准备好待发送的数据；待发送的数据按 4Byte 组成一个固定格式，最高 Byte 用于表示后 3Byte 的数据信息，具体看下文《数据格式解析》。然后配置 SPI_DMA_BADR 选择 DMA 起始地址（按 4Byte 对齐），配置 SPI_DMA_LEN 选择 DMA 数据长度（按 4Byte 对齐）。

19.3.2.3 数据格式解析

每次 DMA 从 SRAM 中取到数据，先做格式解析，然后再发送有效数据。
 对存放在 SRAM 中数据有特殊格式要求，具体如下：

SRAM 中 32 位数据存放格式

CHKN[3:0]	DSN	BSN[2:0]	Data[23:16]	Data[15:8]	Data[7:0]
-----------	-----	----------	-------------	------------	-----------

➤ 是否通过 SPI 传输

其中 Data[26:24]用于控制 Data[23:0]三个 Bytes 是否通过 SPI 传输，命名为 BSN[2:0]，其定义如下：
 BSN[0]=0，Data[7:0]传输；反之不传
 BSN[1]=0，Data[15:8]传输；反之不传
 BSN[2]=0，Data[23:16]传输；反之不传

➤ 数据区还是帧头帧尾

Data[27]用于控制 Data[23:0]中三个 Byte 是帧头帧尾，还是有效数据，命名为 DSN。该位主要是配置间隔发送使用，具体见《[等间隔发送有效数据](#)》章节

DSN=0，表示 Data[23:0]是有效数据
 DSN=1，表示 Data[23:0]是帧头或帧尾

➤ 是否参与校验计算

Data[30:28]用于控制 Data[23:0]中三个 byte 是否参与校验计算，命名为 CHKN[2:0]，其定义如下：
 CHKN[0]=0，Data[7:0]参与校验计算；反之不参与
 CHKN[1]=0，Data[15:8]参与校验计算；反之不参与
 CHKN[2]=0，Data[23:16]参与校验计算；反之不参与

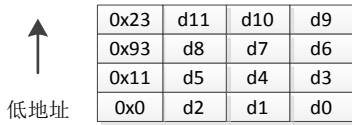
➤ 校验结束并开始发送校验和

Data[31]用于控制是否校验结束，命名为 CHKN[3]，其定义如下：

CHKN[3]=0，不是参与校验的最后字节；1，参与校验的最后字节；传完该 word 所有有效数据后，下一个字节开始传输校验位（如果是校验和模式，校验和是 1 字节，如果是 CRC 模式，校验和是 2 字节）。

当一次 DMA 发送的数据中包含多个校验结束标志，则发送多次 DMA，每次发送校验结构之后，自动把 CRC_INIT 设置为初始值，后续收到的数据重新开始做 CRC 校验。

应用示例：



如上图，DMA 首先从 SRAM 中读到第一个 4Byte 数据，解析最高 Byte 数据“0x0”，得到需要发送的数据为 d0, d1, d2, 需要校验数据为 d0, d1, d2;

接着读下一个 4Byte 数据，解析最高 Byte 数据“0x11”，得到需要发送数据为 d4、d5，需要校验数据位 d4、d5;

读下一个 4Byte 数据，解析最高 Byte 数据“0x93”，得到需要发送数据为 d8，需要校验数据为 d7、d8，并且在发送完 d8 之后，发送校验结果。

读下一个 4Byte 数据，解析最高 Byte 数据“0x23”，得到需要发送数据为 d11，下一次数据继续做 CRC 校验。

19.3.2.4 发送数据位宽可配置

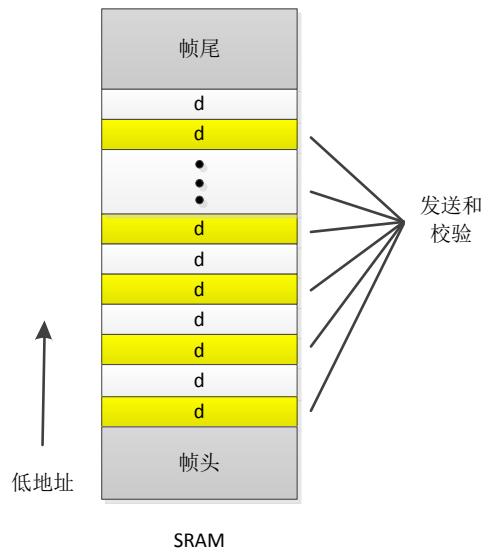
通过配置 SPI_DMA_CTRL 寄存器的 DMA_MASK 位，可以选择发送的每个 ADC 采样点数据的位宽，3Byte 原始数据可任意配置是否有效。比如 DMA_MASK=110，则表示 24Bit 原始数据只有高 16Bit 有效，当 DMA 4Byte 数据中最高 Byte 解析得到 BSN=0, CHKN=0 时，SPI 发送的时候，只会发送高 16Bit 原始数据数据，参与校验的数据也只有高 16Bit。

注意：

- 1) 该配置和《数据格式解析》章节提到的 BSN 共同决定当前数据是否发送，与 CHKN 共同决定当前数据是否做 CRC 校验。
- 2) 该配置只对数据区有效，对于帧头和帧尾区，数据是否发送，只根据最高 Byte 中的 BSN 位决定。

19.3.2.5 等间隔发送有效数据

该模式下，可以支持对一帧数据，其中采样点数据，等间隔抽取，只发送其中一部分有效数据；而帧头和帧尾全部发送。如下图所示：



上图中，是一个两倍抽取发送的例子，其中帧头帧尾全部发送，数据区，每 2 个 ADC 采样数据，只发送第一个。

支持 2/4/8/16 倍抽取发送，通过 SPI_DMA_CTRL 寄存器 EXT_CFG 配置。每次 n 倍抽取发送的数据，总是 n 个数据最先采样得到的数据。

19.3.2.6 校验计算

支持两种校验方式：校验和（8bit）、CRC 校验（ $x^{16} + x^{12} + x^5 + 1$ ）。

通过配置寄存器 SPI_DMA_CTL[2]=1, 使能在 DMA 传输过程中, 自动做校验计算。配合数据格式中的 CHKN, 会得到一个校验结果, 如果是校验和方式, 则结果存放到 SPI_CRC 低 8bit; 如果是 CRC 方式, 则校验结果存放到 SPI_CRC 低 16bit。

参与校验的数据, 由《[数据格式解析](#)》章节中的 CHKN、寄存器 SPI_DMA_CTRL 寄存器的 DAT_SEL 以及抽取配置 EXT_CFG 共同决定。

如果配置了 SPI_DMA_CTRL[3]=1, 则根据数据格式中的 CHKN[3], 在校验结束点, SPI 插入发送校验结果。

19.3.2.7 DMA 中断

启动 DMA 之后, 硬件自动从 SRAM 中读取数据, 发送到 SPI 总线, 当发送完指定长度数据后, SPI 停止, 置位 SPI_DMA_FLAG[0], 表示 DMA 发送完成。如果配置 SPI_DMA_IE[0]=1, 则会产生中断。

19.4 CRC 校验或校验和计算

当 SPI 接口的 DMA 功能空闲时, 把 CRC 或者校验和作为一个独立的功能, 用来计算 SRAM 中一整片区域数据的 CRC 或校验和。结果存放到寄存器, CPU 可读。

19.4.1 特点

- 当配置 CRC 校验时, DMA 起始地址和长度按 16 位对齐, 任意配置
- 当配置校验和计算时, DMA 起始地址和长度可按任意 Byte 对齐
- CRC 运算初始值可任意配置
- 校验和运算初始值可任意配置
- 使用该模式时, 如果 SPI 接口不使用 DMA, 则 SPI 口和校验计算能够同时工作。

19.4.1.1 模式选择

通过配置 SPI_DMA_CTRL[3:2]=2'b10 使能该模式。配置 SPI_DMA_CTRL[6]=1 选择 CRC 模式, 配置 SPI_DMA_CTRL[6]=0 选择校验和模式。通过 SPI_DMA_BADR 和 SPI_DMA_LEN 配置 buffer 起始地址和长度, 配置 SPI_CRC_INIT 配置初始值, 等待 SPI_DMA 完成中断, 读取寄存器获得 CRC 或者校验和的值。

19.5 寄存器定义

19.5.1 寄存器列表

模块名	物理地址	映射地址
SPI0	0x40020000	0x40020000

寄存器名	地址偏移量	描述
SPI0_CTL	0x0	SPI0 控制寄存器
SPI0_STAT	0x4	SPI0 状态指示寄存器
SPI0_TXDATA	0x8	SPI0 数据发送寄存器
SPI0_RXDATA	0xC	SPI0 数据接收寄存器
SPI0_TXDFLT	0x10	SPI0 发送数据为空时，默认发送数据配置
SPI0_DMA_CTRL(新增)	0x14	SPI0 DMA 控制寄存器
SPI0_DMA_BADR(新增)	0x18	SPI0 DMA 起始地址
SPI0_DMA_ADDR(新增)	0x1C	SPI0 DMA 当前指针
SPI0_DMA_LEN(新增)	0x20	SPI0 DMA 长度
SPI0_DMA_FLAG(新增)	0x24	SPI0 DMA 标识
SPI0_DMA_IE(新增)	0x28	SPI0 DMA 中断
SPI0_CRC_INIT(新增)	0x2C	SPI0 DMA CRC 校验初始值
SPI0_CRC(新增)	0x30	SPI0 DMA CRC 校验结果

19.5.2 SPI 控制寄存器 SPI_CTL (0x00)

偏移地址=0x0

比特位	名称	描述	读/写标志	复位值
31:26	---	预留	R	0
25	SCSN_POS_IRQ EN	SCSN_POS 中断使能 0, 关闭中断 1, 打开中断	RW	0
24	SCSN_NEG_IRQ EN	SCSN_NEG 中断使能 0, 关闭中断 1, 打开中断	RW	0
23	TXEMPT_IRQE N	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	RW	0

		0: 发送上次传输 SPI_TXDATA 的值		
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=APBClock/(2*(CLKDIV + 1))	R/W	000
11:10	WIDTH	数据宽度选择 00: 8bit 01: 16bit 10: 32bit 11: 预留, 8bit	R/W	00
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 选择方式 0: MSB 先传输 1: LSB 先传输	R/W	0
5	TXCOL_IRQ_EN	数据冲突中断使能信号 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 模式错误中断使能信号 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	数据传输中断使能信号 0: 关闭发送数据中断 1: 打开发送数据中断	R/W	0
1	MAST/SLAV	主从选择 1: MASTER 0: SLAVE	R/W	1
0	EN	使能信号 0: 关闭 SPI 接口 1: 打开 SPI 接口	R/W	0

19.5.3 SPI 状态标识寄存器 SPI_STAT (0x04)

偏移地址=0x4

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7	SCSN_PO S	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_N EG	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	TXBUF_ VLD	TXBUF 非空标志位。 0: TXBUF 为空，软件可以向 TXBUF 中写入新数据 1: TXBUF 非空，软件不可向 TXBUF 中写入新数据	R/W	0
3	TXCOLIF	写数据冲突标志位。当有 BUF 满时，总线向 SPI 发送写数据的命令，则会将该位置 1，同时不会响应本次写操作的命令，若 TXCOL_IRQ_EN 中断使能，则会产生中断，该位写 1 清零。 0: 没有发生写数据冲突中断 1: 产生写数据冲突中断	R/W	0
2	RXCOLIF	接收数据溢出：在下次完整的接收数据进入移位寄存器之前，没有读取 RXDATA 寄存器，将产生接收数据溢出，则 RXCOLIF 置 1，若 COL_IRQ_EN=1，则会产生中断，同时新接收的数据会覆盖之前的数据，该位写 1 清零。 0: 没有接收数据溢出中断 1: 产生接收数据溢出中断	R/W	0
1	ERRIF	SCSN 模式冲突中断标识位：SPI 为主模式，只有在 SCSN_EN 为 1 时，同时检测到“SCSN”输入电平为低，则该位置 1；SPI 为从模式，“SCSN”作为从机的片选输入，在数据传输过程中，若“SCSN”输入电平为高，则该位置 1；若 ERR_IRQ_EN=1，则会产生一个中断，一旦发生模式冲突错误，则 SPI 模块复位。该位写 1 清零。 0: 没有模式冲突中断 1: 产生模式冲突中断	R/W	0
0	TRIF	数据传输中断标识位，若数据传输结束，该位置 1，若 TR_IRQ_EN=1，则会产生中断，该位写 1 清零。 0: 没有数据发送中断 1: 产生数据发送中断，发送数据寄存器为空	R/W	0

19.5.4 SPI 数据发送寄存器 SPI_TXDATA (0x08)

偏移地址=0x8

比特位	名称	描述	读/写标志	复位值
31:0	TXDATA	数据发送寄存器	R/W	0

19.5.5 SPI 数据接收寄存器 SPI_RXDATA (原有, 0x0C)

偏移地址=0xC

比特位	名称	描述	读/写标志	复位值
31:0	RXDATA	数据接收寄存器	R	0

19.5.6 SPI 默认发送数据寄存器 SPI_TXDFLT (原有, 0x10)

偏移地址=0x10

比特位	名称	描述	读/写	复位值
31:0	TXDFLT	默认发送数据寄存器	R/W	0

19.5.7 SPI0_DMA_CTRL(新增, 0x14)

偏移地址: 0x14; 默认值: 0x18180

位	位名称	功能描述	读写标志	复位值
31:18	Reserved	保留。	R	0
17	SPC_TB_INV	专用 DMA 模块, 该位配置为“1”, 表示每次从 SRAM 一次数据, {d3[7:0], d2[7:0], d1[7:0], d0[7:0]}, 按照 d0→d1→d2 顺序发送到 SPI 上。如果该位配置为 0, 则按照 d2→d1→d0 顺序发送	R/W	0
16	CRC_INV	=1, CRC 最终计算结果按位取反 计算结果 {d15, d14, d13...d2, d1, d0} 变成 {~d15, ~d14, ~d13...~d2, ~d1, ~d0}	R/W	1
15	CRC_TX_LSB	CRC 结构配置, =0, SPI 先发送 CRC 高 Byte =1, SPI 先发送 CRC 低 Byte	R/W	1
14:12	EXT_CFG	该位域只用于专用 DMA 模式 抽取倍数配置, 用于配置只发送有效数据中一部分 000 = 全部发送 001 = 1/2 抽取发送 010 = 1/4/抽取发送 011 = 1/8 抽取发送 100 = 1/16 抽取发送	R/W	0

		其他 = 保留		
11:9	DAT_MASK	该位域只用于专用 DMA 模式 表示每次从 SRAM 数据区中读到 4Byte 数据，低 3Byte 是否有效（数据区与帧头尾区划分依据《 数据格式解析 》中的 DSN） DAT_MASK[0]选择最低 Byte， DAT_MASK[1]选择次低 Byte， DAT_MASK[2]选择倒数第 3Byte， 其中 DAT_MASK[n]=0, 表示对应 Byte 有效，否则无效。 最终决定数据有效与否，取决于每 4Byte 数据最高 Byte 和该寄存器配置，只有同时有效，数据才有效。	R/W	0
8	CRC_DOUT_LSB	校验数据输出反序：1=反序， 0=不反序 如果是 CRC 模式，反序时，每 16 位数据取反 bit0 → bit15 bit1 → bit14 ... 如果是校验和模式，反序时，每 8 位数据取反 bit0 → bit7 bit1 → bit6 ...	R/W	1
7	CRC_DIN_LSB	校验数据输入反序：1=反序， 0=不反序	R/W	1
6:5	CHK_MODE	校验模式选择 00 = CRC 01 = checksum 8bit 10 = checksum 16bit 11 = checksum 32bit	R/W	0
4	CHKN_TX	在 SPI 发送中间，发送 CRC 或校验和 1=使能 0=不使能 该位只有在 CHKN_EN=1 时有效。	R/W	0
3	CHKN_EN	在 DMA 过程中计算 CRC 或者校验和使能： 1=使能 0=不使能 如果使能了 CHKN_EN，则每次从 SRAM 中读到一个 32bit 数据，会判断第 29~27 bit，自动做校验和，并且判断第 30bit，如果为 1，则发送校验和。	R/W	0
2:1	MODE_SEL	DMA 模式选择： 00 = SPI 普通模式 01 = SPI 专用模式 10 = 校验模式 11 = 保留	R/W	0
0	TX_DMA_EN	TX DMA 使能位；=0 为不使能，=1 为使能； DMA 完成后，该位自动清零。	R/W	0

19.5.8 SPI0_DMA_BADR(新增, 0x18)

偏移地址: 0x18; 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
15:0	SPI0_DMA_BASE_ADDR	DMA 起始地址 (Byte 地址, 对应 SRAM 地址的低 16 位) 当 SPI 接口数据宽度配置为 32bit 时, 最低两位必须配置为 0 当 SPI 接口数据宽度配置为 16bit 时, 最低位必须配置为 0	R/W	0

19.5.9 SPI0_DMA_ADDR(新增, 0x1C)

偏移地址: 0x1C; 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
15:0	SPI0_DMA_ADDR	当前 DMA 地址 (Byte 地址, 对应 SRAM 地址的低 16 位) 软件通过读取该寄存器, 可以获得当前的 DMA 指针。	R	0

19.5.10 SPI0_DMA_LEN(新增, 0x20)

偏移地址: 0x20; 默认值: 0x0

位	位名称	功能描述	读写标志	复位值
31:16	Reserved	保留。	R	0
15:0	SPI0_DMA_LEN	DMA 长度, n=n Byte 当 SPI 接口数据宽度配置为 32bit 时, 最低两位必须配置为 0, 即使配置了也不起作用 当 SPI 接口数据宽度配置为 16bit 时, 最低位必须配置为 0, 即使配置了也不起作用	R/W	0

19.5.11 SPI0_DMA_FLAG(新增, 0x24)

偏移地址: 2A0H; 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	SPI0_DMA_DONE	DMA 完成标志 读“1”表示 DMA 传输完成 写“1”清除该标志位, 写“0”无效	R/WC	0

19.5.12 SPI0_DMA_IE(新增, 0x28)

偏移地址: 2A0H; 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	SPI0_DMA_DONE_IE	DMA 完成中断使能 1=使能完成中断 0=不使能完成中断	R/W	0

19.5.13 SPI0_CRC_INIT(新增, 0x2C)

偏移地址: 2A0H; 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
15:0	SPI0_CRC_INIT	校验初始值 CRC 模式: CRC 初始值 校验和模式: 低 8bit 为校验和初始值, 高 8bit 无效	RW	0

19.5.14 SPI0_CRC(新增, 0x30)

偏移地址: 2A0H; 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
15:0	SPI0_CRC	校验结果 CRC 模式: CRC 结果 校验和模式: 低 8bit 为校验和结果	R	0

19.6 软件配置流程

19.6.1 普通 DMA 模式

```

... //配置 SPI 接口
SPI->SPI_DMA_BADR = baddr; //配置 DMA 起始地址
SPI->SPI_DMA_LEN = len; //配置 DMA 长度
SPI->SPI_DMA_IE = 1; //配置 DMA 中断使能
SPI->SPI_DMA_CTL = (0x0<<2) | (0x1<<0); //配置 DMA 模式、使能 DMA
    
```

19.6.2 专用 DMA 模式

```

... //配置 SPI 接口
SPI->SPI_DMA_BADR = baddr; //配置 DMA 起始地址
SPI->SPI_DMA_LEN = len; //配置 DMA 长度
SPI->SPI_DMA_IE = 1; //配置 DMA 中断使能
SPI->SPI_DMA_CTL = (chk_mode<<6)|(chk_tx<<5)|(chk_en<<4)|(0x1<<2)|(0x1<<0); //配置校验模式、是否发送校验和、校验使能、DMA 模式、使能 DMA
    
```

19.6.3 CRC DMA 模式

```

... //配置 SPI 接口
    
```

```
SPI->SPI_DMA_BADR = baddr; //配置 DMA 起始地址
SPI->SPI_DMA_LEN  = len;   //配置 DMA 长度
SPI->SPI_DMA_IE   = 1;     //配置 DMA 中断使能
SPI->SPI_DMA_CTL  =(chk_mode<<6)|(0x1<<2)|(0x1<<0); //配置校验模式、DMA 模式、使能 DMA
```

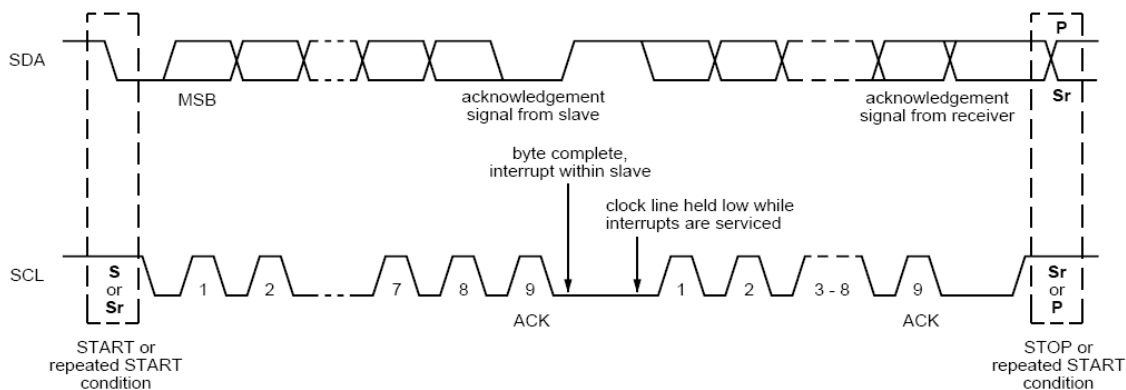
20 IIC 接口

20.1 概述

- 一路 I²C 接口模块;
- 支持 Master 和 Slave 模式;
- 7-bit 地址位
- I²C 外部总线需要外接上拉电阻;
- 传输速度支持标准模式 100kbps 和快速模式 400kbps;
- **新增 IIC 专用 DMA 模式，支持按键触发时通过 VBAT 域 IIC 自动写入万年历时间。详见 RTC 章节。**

20.2 功能描述

I²C 总线由 SDA, SCL 两个信号组成, SDA 是串行数据线, SCL 是串行时钟线。一个完整的 I²C 传输由几部分组成: 起始位 S, 结束位 P, 传输响应位 ACK 以及传输地址/数据字节。传输总是以起始位开始, 以结束位结束或者继续以起始位开始新的传输, 传输过程中每个字节需要对应的响应位。



20.3 寄存器

模块名	物理地址	映射地址
I ² C	0x40024000	0x40024000
寄存器名	地址偏移量	描述
I ² C_CTL	0x0	I ² C 控制寄存器
I ² C_CLK	0x4	I ² C 时钟配置寄存器
I ² C_STAT	0x8	I ² C 状态指示寄存器
I ² C_ADDR	0xC	I ² C 从设备地址寄存器
I ² C_DATA	0x10	I ² C 收发数据寄存器

20.3.1 I²C 控制寄存器 I²C_CTL (0x00)

偏移地址= 0x0

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 信号的接受和发送位 接收数据时： 1: 在接收到第九个 SCL 的时候，产生 ACK 信号 0: 在接收到第九个 SCL 的时候，不产生 ACK 信号	R/W	0
3	IRQE	I ² C 中断使能位 0: 禁止中断 1: 使能中断	R/W	0
2:1	BUSCON	总线控制产生位，开始命令在总线为空闲状态或者主机是发送状态时有效。结束命令在主机是发送状态时有效。 当检测到 start 或者 stop 时序时，对命令位清零， 00: 没有动作 01: 产生 START 时序 10: 产生 STOP 时序 11: 保留	R/W	0
0	EN	I ² C 模块使能位 1: I ² C 打开 0: I ² C 关闭	R/W	0

20.3.2 I²C 时钟配置寄存器 I²C_CLK (0x04)

偏移地址= 0x4

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。	R/W	011														
		<table border="1"> <thead> <tr> <th rowspan="2">系统主频 (MHz)</th> <th colspan="2">分 频 参 数 /CLKDIV(m)</th> </tr> <tr> <th>高速模式</th> <th>正常模式</th> </tr> </thead> <tbody> <tr> <td>8.192</td> <td>010 (22)</td> <td>100 (84)</td> </tr> <tr> <td>16.384</td> <td>011 (42)</td> <td>101 (168)</td> </tr> <tr> <td>32.768</td> <td>100 (84)</td> <td>110 (336)</td> </tr> </tbody> </table>	系统主频 (MHz)	分 频 参 数 /CLKDIV(m)		高速模式	正常模式	8.192	010 (22)	100 (84)	16.384	011 (42)	101 (168)	32.768	100 (84)	110 (336)		
系统主频 (MHz)	分 频 参 数 /CLKDIV(m)																	
	高速模式	正常模式																
8.192	010 (22)	100 (84)																
16.384	011 (42)	101 (168)																
32.768	100 (84)	110 (336)																

的主频时钟频率为 8.192/16.364/32.768MHz 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
8.192	010 (22)	22	372.363636	-7%	100 (84)	84	97.52381	-2%
16.384	011 (42)	42	390.095238	-2%	101 (168)	168	97.52381	-2%
32.768	100 (84)	84	390.095238	-2%	110 (336)	336	97.52381	-2%

20.3.3 I²C 状态指示寄存器 I²C_STAT (0x08)

偏移地址= 0x8

比特位	名称	描述	读/写标志	复位值
31:9	---	预留	R	0
8	DIR	读写方向标志位 1: 读。 0: 写。	R	0
7	MATCH	地址匹配位, 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	通讯状态标志位 0: IIC 处于空闲状态 1: IIC 处于正常通讯状态	R	0
5	COL	发送冲突中断标识位 在从模式下, 发送数据时, 发送数据寄存器不为空, CPU 给数据寄存器写新的数据, 触发发送冲突中断。接收数据时, CPU 给数据寄存器写数据, 触发发送冲突中断。 对该位写 1 将清零 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	接收溢出中断标识位 在从模式下, 接收数据时, 当上一个字节还没有被取走前又收到新的字节, 将触发溢出中断 对该位写 1 将清零 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识位 在从模式下, 写数据时, 数据写完后, 新的数据还没有到来的之前主机发送 SCL 要求从机发送新的数据, 这种情况下会触发发送数据寄存器空错误中断 对该位写 1 将清零 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	传输完成中断标志位 对该位写 1 将清零; 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 0: 传输未完成 1: 传输已完成	R/W	0
1	RX_NACK	收到 NACK 中断标志位 对该位写 1 清零 1: 收到 nack 0: 没有接收到 nack	R/W	0

0	STPD	STOP 时序检测中断标志位 对该位写 1 将清零。关闭 I ² C 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
---	------	---	-----	---

20.3.4 I²C 从设备地址寄存器 I²C_ADDR (0x0C)

偏移地址= 0xC

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:1	SADR	设备地址位，在传输地址期间不能向该寄存器写入地址。 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送过来的地址进行比较	R/W	0
0	RW	主机读写方向控制位 0: 写 1: 读	R/W	0

20.3.5 I²C 收发数据寄存器 I²C_DATA (0x10)

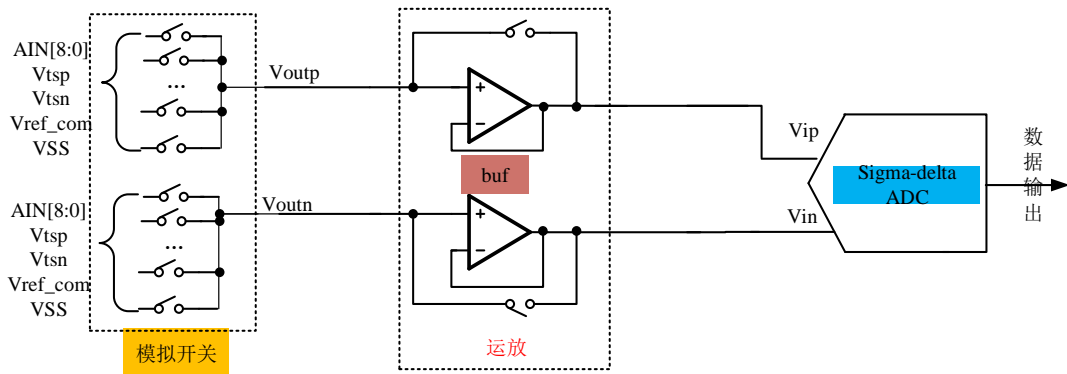
偏移地址= 0x10

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:0	TRDAT	主机发送数据时发送缓存为空，数据才能写入。 接收/发送数据。	R/W	0

21 通用 ADC

21.1 主要特点

- 提供 24bit 多路复用的高精度直流测量 Sigma-Delta ADC。
- 内置运放，使能运放时，内部输入内阻无穷大。
- 支持可变采样率，可根据有效位数需求调整采样率；
- 支持可变增益，PGA 支持 0.2 倍、1 倍、2 倍、4 倍。



GPADC 的相关电路框图如上图所示，电路按功能划分主要分为以下 2 种模式：

- 通用测量模式，信号由 SDAIN[8:0]单端输入，另一端接固定电平 Vcm。经过单位增益 buf，到达 sigma-delta ADC 进行信号处理。
- 内部测温模式，信号由 TPS 温度 sensor 差分输入，经过单位增益 buf，到达 sigma-delta ADC 进行信号处理。

建议客户直接调用锐能微例程实现上述功能。

注意：

- 使用通用 ADC 模块，要求系统主频最低运行在 8.192Mhz，不支持更低频率下的应用。
- RN2025 只支持 SDAIN0/1/2/3/5/8。

21.2 寄存器

21.2.1 寄存器列表

基地址	0x4004C000				
偏移地址	名称	R/W	有效字长	复位值	功能描述
0x0	GPADC_CTL0	R/W	4	0x109	通用 ADC 配置寄存器 0
0x4	GPADC_CTL1	R/W	4	0x000	通用 ADC 配置寄存器 1
0x8	RATIO	R/W	1	0x0	通用 ADC 速率设置
0xC	GPADC_IE	R/W	1	0x0	通用 ADC 中断使能寄存器
0x10	GPADC_IF	R/W	1	0x0	通用 ADC 中断标志寄存器

0x14	GPADC_OUT	R/W	3	0x0	通用 ADC 结果输出寄存器
0x18	GPADC_R2RBUF_CTL	R/W	1	0x0	通用 ADC R2RBUF 配置寄存器

21.2.2 GPADC_CTL0 (0x00)

通用 ADC 配置寄存器 0 偏移地址=0x00

比特位	名称	描述	读 / 写标志	复位值
31:22	---	预留	R	0
21:20	freqsel_gpadc	GPADC 采样时钟选择: 2'b00: 1 倍分频, 2.048Mhz 2'b01: 2 倍分频, 1.024Mhz 2'b10: 4 倍分频, 0.512Mhz 2'b11: 8 倍分配, 0.256Mhz	R/W	0x0
19:16	IBSEL	GPADC 模块偏置电流控制位, default=0, 功耗最大。 Ibsel[3:2]控制 OP1 的偏置电流, 2'b00 =10uA; 2'b01 =7.5uA; 2'b10 =5uA; 2'b11 =2.5uA; ibsel[1]控制 OP2 的偏置电流: =0 表示 5uA; =1 表示 2.5uA; ibsel[0]控制 cmp 的偏置电流: =0 表示 5uA; =1 表示 2.5uA;	R/W	0
15	---	预留	R	0
14	SHORT_EN	预留	R/W	0
13	IBX2_TPS	将 VCC 域温度传感器的工作电流提高一倍: ts_ibx2=0 为默认电流 ts_ibx2=1 为默认电流 x2	R/W	0
12:08	DEM_TPS	VCC 域温度传感器的 DEM 输入控制信号	R/W	5'b00001
07	BUFFP_BYPASS_EN	GPADC 正端输入 Vip 直连模式(无高阻 buffer 电路)使能。此寄存器需配合寄存器 BUFFP_EN 使用, 确保只开启一条通路 0: 断开 1: 直连模式	R/W	0
06	BUFFP_EN	GPADC 正端输入 Vip 高阻 buffer 输入模式使能。此寄存器需配合寄存器 BUFFP_BYPASS_EN 使用, 确保只开启一条通路 0: 断开 1: 高阻 buffer 连接模式	R/W	0
05	BUFFN_BYPASS_EN	GPADC 负端输入 Vin 直连模式(无高阻 buffer 电路)使能。此寄存器需配合寄存器 BUFFN_EN 使用, 确保只	R/W	0

		开启一条通路 0: 断开 1: 直连模式		
04	BUFFN_EN	GPADC 负端输入 Vin 高阻 buffer 输入模式使能。此寄存器需配合寄存器 BUFFN_BYPASS_EN 使用, 确保只开启一条通路 0: 断开 1: 高阻 buffer 连接模式	R/W	0
03:01	GAIN_GPADC	通用通道 gpadc 模块增益控制档位 3'b000 增益=1; 3'b001 增益=2; 3'b01x 增益=4; 3'b1xx 增益=0.2;	R/W	3'b100
00	PWD_GPADC	=0:开启 GPADC =1:关闭 GPADC	R/W	1'b1

21.2.3 GPADC_CTL1 (0x04)

通用 ADC 配置寄存器 1 偏移地址=0x04

比特位	名称	描述	读 / 写标志	复位值
31:29	---	预留	R	0
28:16	VIP_CTL	接入 GPADC 正端 Vip 的信号, '1'表示信号接入; 使用时需同时配置 GPADC 的正负两端接入信号, 外部单端测量时建议另一端接固定电平 vref (约 1.25V) VIP_CTL <12:0> Vip 接入信号(GPADC 正端输入信号) 13'b00000 0000 0001 SDAIN[8] 13'b00000 0000 0010 SDAIN[7] 13'b00000 0000 0100 SDAIN[6] 13'b00000 0000 1000 SDAIN[5] 13'b00000 0001 0000 SDAIN[4] 13'b00000 0010 0000 SDAIN[3] 13'b00000 0100 0000 SDAIN[2] 13'b00000 1000 0000 SDAIN[1] 13'b00001 0000 0000 SDAIN[0] 13'b00010 0000 0000 vtsp-vtsn(此模式为差分模式, 无需配置低 13bits 寄存器) 13'b00100 0000 0000 vss 13'b01000 0000 0000 vref_op6 13'b10000 0000 0000 vref	R/W	0

15:13	---	预留	R	0
12:00	VIN_CTL	接入 GPADC 负端 Vin 的信号，‘1’ 表示信号接入； 使用时需同时配置 GPADC 的正负两端接入信号， 外部单端测量时建议另一端接固定电平 vref（约 1.25V） VIN_CTL <12:0> Vin 接入信号(GPADC 负端输入信号) 13'b00000 0000 0001 SDAIN[8] 13'b00000 0000 0010 SDAIN[7] 13'b00000 0000 0100 SDAIN[6] 13'b00000 0000 1000 SDAIN[5] 13'b00000 0001 0000 SDAIN[4] 13'b00000 0010 0000 SDAIN[3] 13'b00000 0100 0000 SDAIN[2] 13'b00000 1000 0000 SDAIN[1] 13'b00001 0000 0000 SDAIN[0] 13'b00010 0000 0000 -(vtsp-vtsn)(此模式为差分模式，无需配置低 13bits 寄存器) 13'b00100 0000 0000 vss 13'b01000 0000 0000 vref_0p6 13'b10000 0000 0000 vref (RN2025C64 不支持 Vref_0p6)	R/W	0

21.2.4 GPADC_RATIO (0x08)

通用 ADC 数据更新速率设置寄存器 偏移地址=0x08

比特位	名称	描述	读 / 写标志	复位值
31:03	---	预留	R	0
02:00	RATIO	3'b000 : OSR=128 , ADC 数据更新速率 =freq_gpadc/128; 3'b001 : OSR=256 , ADC 数据更新速率 =freq_gpadc/256; 3'b010 : OSR=512 , ADC 数据更新速率 =freq_gpadc/512; 3'b011 : OSR=1024 , ADC 数据更新速率 =freq_gpadc/1024; 3'b100 : OSR=2048 , ADC 数据更新速率 =freq_gpadc/2048; 3'b101 : OSR=4096 , ADC 数据更新速率 =freq_gpadc/4096; 3'b110 : OSR=8192 , ADC 数据更新速率 =freq_gpadc/8192;	R/W	0

		3'b111 : OSR=16384 , ADC 数据更新速率 =freq_gpadc/16384; 其中 freq_gpadc 由寄存器 GPADC_CTL0.freqsel_gpadc 配置, 2.048M/1.024M/0.512M/0.256M 四挡可选。		
--	--	---	--	--

21.2.5 GPADC_IE (0x0C)

通用 ADC 中断使能寄存器 偏移地址=0x0C

比特位	名称	描述	读 / 写标志	复位值
31:1	---	预留	R	0
1	START	写 1 启动 ADC 测量, 测量完成的时间为 4 个数据更新周期 (比如设定为 8Khz 更新, 那么 0.5ms 后测量完成)。 软件需要先配置系统控制章节的 GPADC 模块时钟 (MOD1_EN 寄存器), 然后配置 GPADC_CTL0/GPADC_CTL1 /GPADC_RATIO, 最后再启动该位。 写 0 关闭测量, 同时清 GPADC.IF 标志和内部状态机。启动一次新的测量时, 软件需要先将 START 写 0, 然后再写 1, 写 0 再写 1 的时间间隔需要大于 500ns。	R/W	0
0	IE	=0: 不使能 GPADC 测量完成中断 =1: 使能 GPADC 测量完成中断	R/W	0

21.2.6 GPADC_IF (0x10)

通用 ADC 中断标志寄存器 偏移地址=0x10

比特位	名称	描述	读 / 写标志	复位值
31:1	---	预留	R	0
0	IF	=0: GPADC 测量未完成 =1: GPADC 测量完成, 若 GPADC_IE.IE=1, 则产生中断。 该位写 1 清零, 或者 GPADC_IE.START 写 0 关闭测量时自动清零。 GPADC_IE.START 写 0 再写 1 启动 ADC 测量, 需经过 4 个 ADC 数据更新周期数据稳定, 同时 GPADC_IF 置 1, 若 GPADC_IE.IE=1, 则产生中断。 GPADC_IF 写 1 清零, 之后每个 ADC 数据更新周期 GPADC_IF 置 1。 即 GPADC_IE.START 写 0 再写 1 启动 ADC 测量后, 第一次 IF 置位时间是 4 个 ADC 数据更新周期, 后续 IF 置位时间是 1 个 ADC 数据更新周期。	R/W	0

--	--	--	--	--	--

21.2.7 GPADC_OUT (0x14)

通用 ADC 数据输出寄存器 偏移地址=0x0C

比特位	名称	描述	读 / 写标志	复位值
31:0	GPADC_OUT	GPADC 测量结果输出，有效位数是 32 位； 二进制补码格式，最高位是符号位。 测量结果更新周期= freq_gpadc/OSR 由寄存器 GPADC_CTL0.freqsel_gpadc 和 GPADC_RATIO 配置确定。 不论数据更新速率如何设置，在相同的 ADC 输入情况下，得到的 GPADC_OUT 值都是基本相同的。 如果输入到 GPADC 的信号幅值为 A，那么寄存器值为： $A*0.5*2^{31}$ 。其中 A 的单位是伏，最大不超过 1V。 举例： 输入为 0.1V 的直流信号，那么 GPADC_OUT 寄存器值为： $0.1*0.5*2^{31}=32'H666_6666$ 。	R	0

备注：

不同 PGA 下，一端固定为 Vref(1.25V)，另一端输入信号的测量范围如下，

PGA=1，则需满足 $|V_{in} - 1.25| \leq 1V$ ，可得输入信号范围为 $0.25V \leq V_{in} \leq 2.25V$

PGA=2，则需满足 $|V_{in} - 1.25| \leq 0.5V$ ，可得输入信号范围为 $0.75V \leq V_{in} \leq 1.75V$

PGA=4，则需满足 $|V_{in} - 1.25| \leq 0.25V$ ，可得输入信号范围为 $1V \leq V_{in} \leq 1.5V$

PGA=0.2，则需满足 $|V_{in} - 1.25| \leq 5V$ ，可得输入信号范围为 $-3.75V \leq V_{in} \leq 6.25V$ 。

虽然使用 PGA=0.2 增益档位可增大输入信号范围，但是考虑到 GPADC 内部开关的耐压问题，需要限制通用测量时外部输入信号过高，实际应用下要求 $0.02V < V_{in} < \text{芯片供电电压 (3.3v 或者 5v)}$

21.2.8 GPADC_R2RBUF_CTL (0x18)

(RN2025C64 无 R2RBUF，该寄存器不起作用)

比特位	名称	描述	读 / 写标志	复位值
31:3	---	预留	R	0x0
2	En_gmn_r2rbuf	R2R BUF 输入级 NMOS Gm 差分队使能信号 0，关闭；1，开启；	R/W	0x1

1	En_gmp_r2rbuf	R2R BUF 输入级 PMOS Gm 差分队使能信号 0, 关闭; 1, 开启;	R/W	0x1
0	Chop_r2rbuf	R2R BUF 内部 chop 信号, 用于减小 R2R BUF 本身的失配。	R/W	0x1

22 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，如图 22-1 所示（详细的操作方法请见《MINIPRO 编程器使用手册》）。

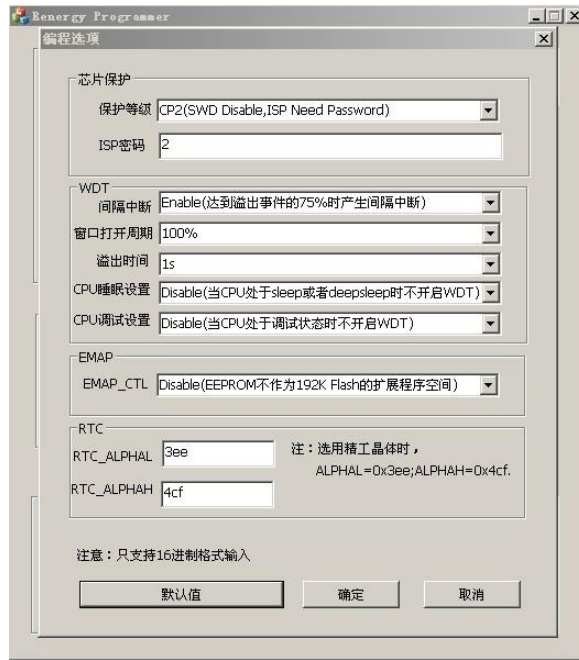


图 22-1 选项字节设置界面

22.1 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，用户可以通过设置保护等级和ISP密码对芯片进行保护。保护机制提供了如下保护等级：

保护等级	名称	说明
0	CP0	无任何保护（ISP 访问也不需要密码）
1	CP1	SWD 接口可访问芯片，ISP 访问需要密码
2	CP2	禁止通过 SWD 接口访问芯片，ISP 访问需要密码
3	CP3	禁止通过 SWD 和 ISP 接口访问芯片（ISP 只提供整片擦除 FLASH 的功能（在该保护等级下擦除操作会使芯片的保护等级将为 CP0））

22.2 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：

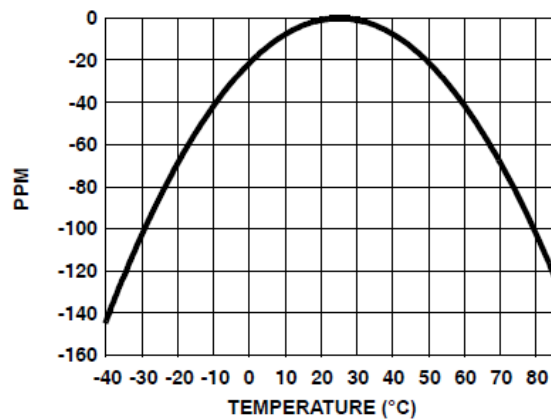
名称	描述	厂家默认值
间隔中断	0: Disable (不使能间隔中断) 1: Enable (达到溢出事件的 75%时产生间隔中断)	0
窗口打开周期	0: 25% 1: 50% 2: 75% 3: 100% 在窗口打开期间将 0xBB 写入 WDTE 寄存器, 看门狗清零并重新计数; 在窗口关闭期间将 0xBB 写入 WDTE 寄存器, 会产生内部复位信号。	3
溢出时间	0: 16ms 1: 32ms 2: 128ms 3: 512ms 4: 1s 5: 2s 6: 4s 7: 8s	4
CPU 睡眠设置	0: Disable (当 CPU 处于 sleep 或者 deepsleep 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

22.3 RTC 设置

RN2026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 ($f=f_0-\alpha*(T-T_0)^2$, 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 $\text{round}(\alpha*32768)$ 其中round为四舍五入操作。

如果选用的是精工 (Seiko) 提供的高一致性晶体 (VT-200-F), ALPHAL=0x3ee, ALPHAH=0x4cf。



升级版芯片在兼容 A 版二次曲线补偿的情况下，还提供了四次曲线补偿，可以更准确的对晶振曲线进行补偿。可调用锐能微库函数实现对四次曲线参数的管理。

23 编程支持

芯片支持对内置 FLASH 的编程。

建议客户调用锐能微库函数实现 IAP 功能。

23.1 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；

23.2 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 23-1 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））

23.3 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 22-1 所示

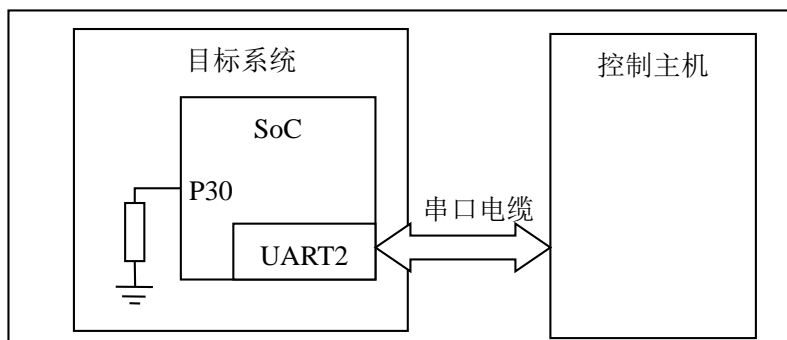


图 23-1 ISP 硬件配置图

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 57600bps，保证波特率偏差；
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“8192(1843)/r/n”；（如果当前系统频率为 8.192M，则发送 8192；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；

23.3.1 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 23-2 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 23-3 ISP 数据校验格式

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

23.3.2 使用的资源

使用的 RAM 资源

ISP使用片内0x10007800到0x10008000范围内的RAM，堆栈位于RAM的顶部。Flash能使用0x10000000-0x10007800（30KB）范围内的RAM进行编程。

23.3.3 ISP 命令

每个ISP命令都支持具体的状态代码。当接收到未定义命令时，命令处理程序发送返回代码INVALID_COMMAND。

命令和返回代码为ASCII 格式。只有当接收到的ISP命令执行完毕时，ISP 命令处理器才会发送CMD_SUCCESS，这时主机才能发送新的ISP 命令。

ISP命令分成三种：

1. 普通命令：只在CP0，或者在CP1，CP2并且保护密码正确的情况下，可以访问
2. UN命令在CP0，CP1，CP2保护等级（保护密码不提供）下，都能访问
3. FC, AL命令在任何情况下都能访问

表 23-4 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
Flash 编程	FW <FLASH地址><RAM地址><字节长度>	普通命令
内存比较	MC <地址1><地址2><字节长度>	普通命令
运行	GO <地址>	普通命令
解锁	UN <密码>	
获取保护等级	AL	
软件复位	RS	普通命令
引导 Flash 代码	BF	普通命令
使能 NVM (FLASH)	NV <NVM选项>	普通命令
软 ISPEN 使能	IE <SOFT ISPEN使能命令><UART2 IO 复用选择>	普通命令
软 ISPEN 清除	IC <SOFT ISPEN清除命令>	普通命令
内存校验和比较	CS <地址><字节长度><校验和>	普通命令

● 波特率设置

表 23-5 波特率设置 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 或 256000 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 23-6 回显 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 23-7 写内存 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 23-8 读内存 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 23-9 Flash ISP 命令

命令	FP <页地址> (FPGA 版本为 0 到 3071)
输入	页地址: 0 到 1024 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 23-10 Flash 块擦除 ISP 命令

命令	FS <块地址>
输入	块地址: 0 到 16 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 23-11 Flash 片擦除 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 23-12 Flash 块查空 ISP 命令

命令	FQ <块地址>
输入	块地址: 0 到 16 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 23-13 Flash 编程 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 也其余内容填充为 0)

返回代码	CMD_SUCCESS 或 COUNT_ERROR 或 SRC_ADDR_NOT_ALIGN 或 SRC_ADDR_NOT_MAPPED 或 DST_ADDR_NOT_ALIGN 或 DST_ADDR_NOT_MAPPED 或 INVALID_PARAM
说明	用于编程Flash 存储器。
范例	“FW 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 23-14 内存比较 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 23-15 运行 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 23-16 解锁 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 23-17 获取保密等级 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 23-18 软件复位 ISP 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

- 引导 Flash 代码

表23-18 软件复位ISP命令

命令	BF
输入	无
返回代码	CMD_SUCCESS 或 INVALID_PARAM
说明	该命令引导Flash代码
范例	“BF” 将引导Flash代码

- 使能 NVM

表 23-19 使能 NVM 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。

- 软 ISPEN 使能

表23-21 软件复位ISP命令

命令	IE <SOFT ISPEN 使能命令><UART2 IO 复用选择>
输入	SOFT ISPEN使能命令 (SOFT_ISPEN_CMD) : =0xe7a5185a, 使能软ISPEN; =其他值, 不使能软ISPEN; UART2 IO 复用选择 (UART2_IO): =0x7e5a1234, UART2 复用 IO 选择 P04、P05 =0x7e5a5678, UART2 复用 IO 选择 P20、P21 =0x7e5a9abc, UART2 复用 IO 选择 P22、P23
返回代码	CMD_SUCCESS 或 INVALID_COMMAND 或 SECTOR _NOT_BLANK 或 INVALID_PARAM
说明	该命令用来使能SOFT_ISPEN
范例	“IE 3886356570 2119832116” SOFT_ISPEN_CMD =0xe7a5185a, 使能SOFT_ISPEN, UART2_IO =0x7e5a1234, 选择P04/P05作为 RX2/TX2

- 软 ISPEN 清除

表23-22 软件复位ISP命令

命令	IC<SOFT ISPEN 清除命令>
输入	SOFT ISPEN清除命令 (SOFT_ISPEN_CLR) : =0x69789687, 永久清除软ISPEN, 不允许再次使能软ISPEN; =其他值, 清除软ISPEN, 可再次使能软ISPEN;
返回代码	CMD_SUCCESS 或 INVALID_COMMAND 或 SECTOR_NOT_BLANK 或 INVALID_PARAM
说明	该命令用来使能SOFT_ISPEN
范例	“IC 1769510535” SOFT_ISPEN_CLR =0x69789687, 永久清除SOFT_ISPEN

- 内存校验和比较

表23-23 软件复位ISP命令

命令	CS <地址><字节长度><校验和>
输入	地址 (SRC) : 要比较的内存区域起始地址。应当与字对齐; 字节长度: 待计算校验和的字数; 校验和: 待比较区域预先计算好的校验和, 32bit 累加和的负数补码;
返回代码	CMD_SUCCESS 或 COUNT_ERROR 或 COMPARE_ERROR 或 INVALID_PARAM
说明	该命令用来比较存储器某一段内容的校验和
范例	“CS 402653184 200 12345678” 将FLASH地址0x18000000的200个WORD校验和与12345678进行比较

23.3.4 ISP 返回代码

表 23-204 ISP 返回代码

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

23.4 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22-2。参数和结果的数目根据IAP命令而有所不同。“Flash编程”命令参数的最大数目为4。结果的数目为1。命令处理器在接收到一个未定义的命令时发送状态代码INVALID_COMMAND。IAP程序是Thumb代码，驻留在地址0x1800_1c01。

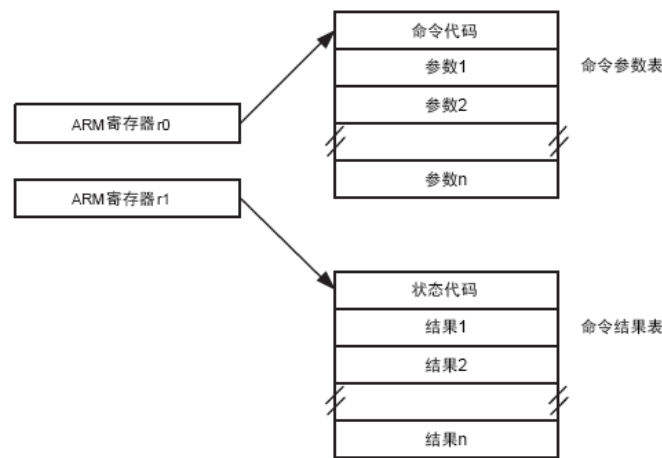


图 23-2 IAP 参数传递

23.4.1 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

软 ISPEN 使能	0x5c	参看 ISP 章节
软 ISPEN 清除	0x5d	参看 ISP 章节
内存校验和比较	0x5e	参看 ISP 章节

23.4.2 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。

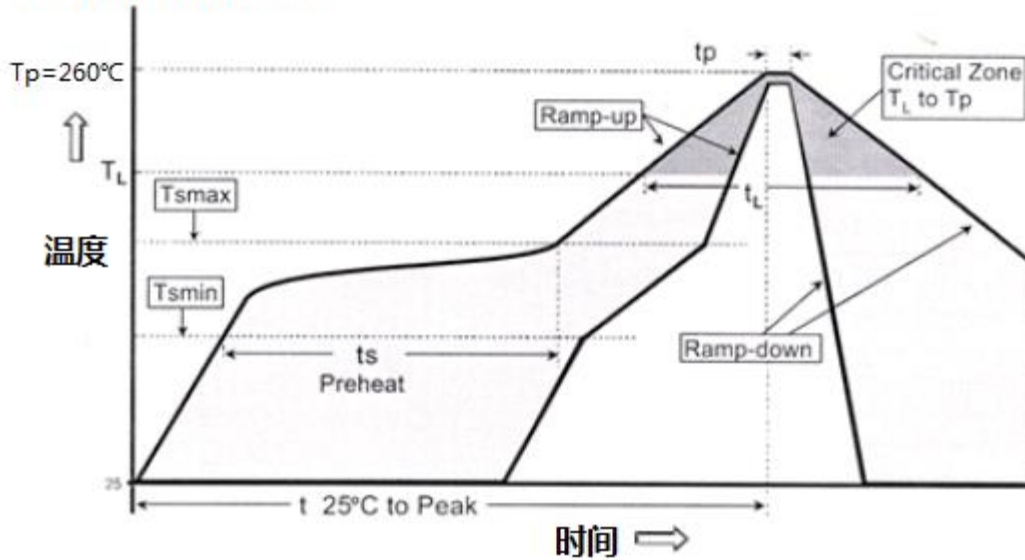
23.5 量产平台

Reenergy 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》。

24 焊接条件

回流焊炉的温度设定条件

回流焊炉温度设定条件

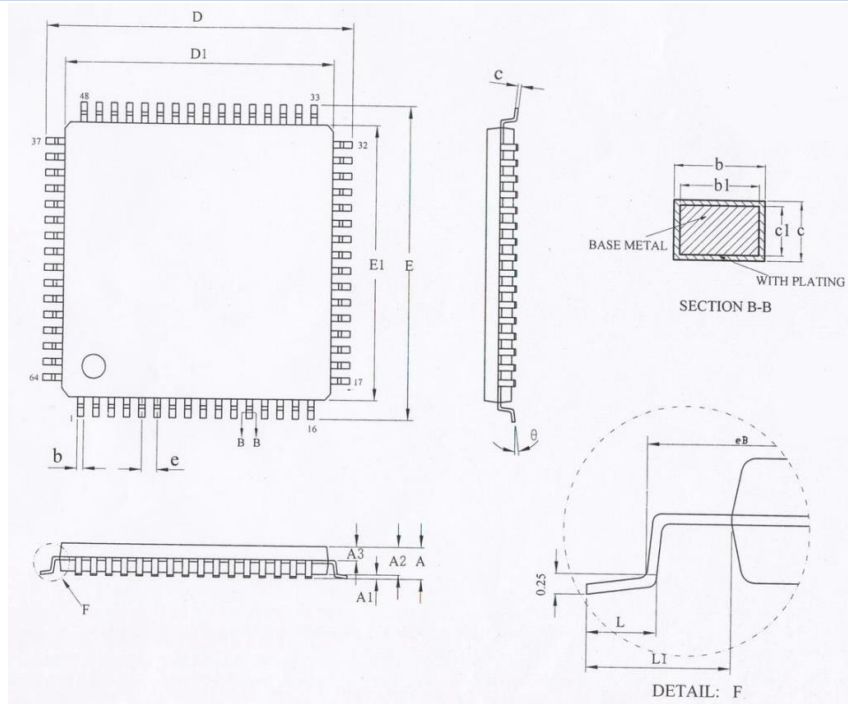


回流焊炉的温度设定曲线


分布图特征	值
平均倾斜上升率 (TL to Tp)	最大 3°C/秒
预热 最小温度 (Ts min)	150°C
最大温度 (Ts max)	200°C
时间 (最小-最大) (ts)	60-180 秒
Ts max - TL 倾斜上升率(Ts max to TL)	最大 3°C/秒
保持以上时间 -温度 (TL)	217°C
-时间 (tL)	60-150 秒
峰值温度 (Tp)	260+5/-0°C
实际峰值温度 5°C 内的时间 (tp)	20-40 秒
倾斜下降率	最大 6°C/秒
25°C 到峰值温度的时间	最大 8 分钟
保温温度 TL	217°C
峰值温度 Tp	260°C
平均倾斜上升率(TL to Tp)	最大 3°C/秒

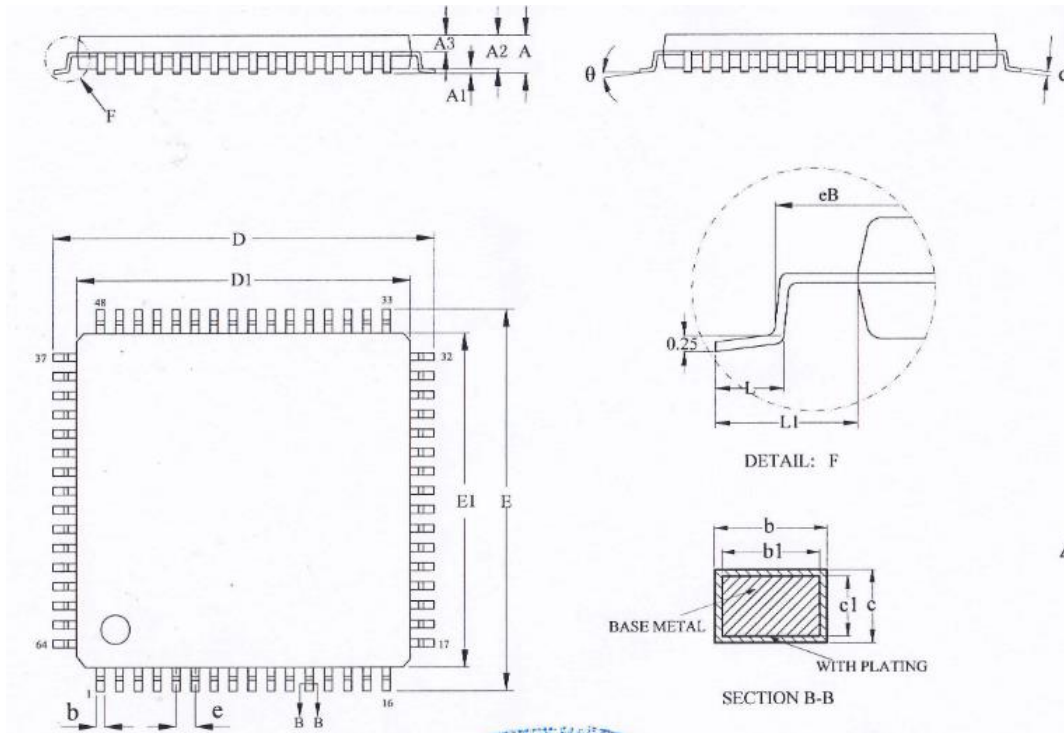
25 封装尺寸

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



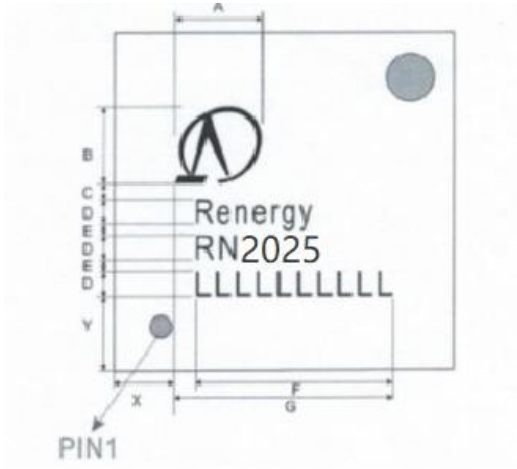
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	---	---	1.6
A1	0.05	---	0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.17	---	0.25
b1	0.16	0.18	0.20
c	0.13	---	0.18
c1	0.12	0.127	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
eB	8.10	---	8.25
e	0.40BSC		
L	0.40	---	0.65
L1	1.00BSC		
θ	0	-----	7°

LQFP64L (1010×1.4)		10.00×10.00×1.40	e=0.50
-----------------------	---	------------------	--------



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	---	---	1.6
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	11.80	12.00	12.20
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
eB	11.05	---	11.25
e	0.50BSC		
L	0.45	---	0.75
L1	1.00BSC		
θ	0	-----	7°

产品外观图如下，以 RN2025 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 RN2025 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。