

RN8318\_RN8615\_RN8613\_RN8612\_RN8611

用户手册

版本:V1.5

深圳市锐能微科技股份有限公司

## 修改记录

版本号	拟制人/修改人	拟制/修改日期	更改理由	主要更改内容
V1.0	锐能微 客服部	2016-10-09	创建	
V1.1	锐能微 客服部	2017-03-01	修改	<ol style="list-style-type: none"> <li>1. 新增型号 RN8612, 相关章节作出修改;</li> <li>2. 更改 RN8613 和 RN8615 的封装定义, PIN98/99/100/14/15 定义作出调整;</li> <li>3. 完善手册, 修正部分错误。</li> </ol>
V1.2	锐能微 客服部	2017-03-12	修改	完善手册, 补充修改各个章节说明文字。
V1.3	锐能微 客服部	2017-03-15	修改	<ol style="list-style-type: none"> <li>1. 调整第 2 章电气特性部分指标</li> <li>2. 对 3.5 复位章节做更详细描述</li> <li>3. 对第 9 章模拟外设做更详细描述</li> <li>4. 对 LCD/INTC/KBI 部分说明做出调整</li> </ol>
V1.4	锐能微 客服部	2017-3-28	修改	<ol style="list-style-type: none"> <li>1. 修改 4.3 章节 bitband 说明错误</li> <li>2. 修改图 1.4 RN8611 管脚排列图中 P66/SEG10、P67/SEG11 为 P66/P67</li> <li>3. 第 9 章模拟外设增加 AIN 输入阻抗说明</li> <li>4. 第 15 章 IIC 章节修改, 在 1.8432MHz 时钟下不支持高速模式</li> <li>5. 第 14 章 7816 章节修改, 卡时钟输出不支持 32/64/128 分频</li> <li>6. 第 9 章模拟外设修改, LVD 阈值电压设置寄存器描述错误</li> <li>7. 4.3.1 章节 SRAM 章节增加 BOOTROM 占用 RAM 地址空间描述和 CACHE 开启描述。</li> <li>8. 第 2 章电气特性增加极限参数描述、增加更多参数指标。</li> <li>9. 第 1.4 章节, P30 端口描述作出修改</li> <li>10. 第 19 章增加产品外观说明</li> </ol>
V1.5	锐能微 客服部	2017-6-17	修改	<ol style="list-style-type: none"> <li>1、修改 7.2.6 中笔误 BIAS, 正确为 RES</li> <li>2、修改二次温补 T5-T9 的边界温度</li> <li>3、修改 3.5.2 电源监测电路描述</li> <li>4、修改 1.4 管脚描述中 (P18) 51/66 脚的笔误 LVDVP1 为 LCDVP1</li> <li>5、时钟框图中增加 RCL 可作为 LCD 及 CPU 时钟的连线</li> <li>6、修改 9.2.1 REF_WAIT 的描述及并为 SAR_WAIF 的描述增加备注</li> <li>7、3.6.6 增加描述</li> </ol>

				<p>8、修改第 2 章“电气特性”中 HBM CDM MM Latch-up 试验参数独立成表</p> <p>9、修改第 2 章“电气特性”中基准电压、比较器阈值、温补时间、工作电压等指标定义</p> <p>10、修改 1.4 章节管脚排列，RN8613/8615 的 PIN10~PIN13 由 NC 更改为 P104~P107</p>
--	--	--	--	--

## 目录

<b>1</b>	<b>概述.....</b>	<b>9</b>
1.1	简介.....	9
1.2	产品特点.....	9
1.3	实现架构.....	12
1.4	管脚排列.....	13
1.5	IO 口功能框图 .....	24
<b>2</b>	<b>电气特性.....</b>	<b>30</b>
<b>3</b>	<b>系统控制.....</b>	<b>33</b>
3.1	电源域划分.....	33
3.2	时钟源.....	34
3.3	时钟切换.....	34
3.4	MCU 的低功耗模式.....	35
3.5	复位.....	36
3.5.1	外部 PIN 复位 .....	36
3.5.2	上下电复位.....	36
3.5.3	软件复位.....	36
3.5.4	看门狗复位.....	36
3.6	寄存器描述.....	36
3.6.1	系统 OSC 控制寄存器 1OSC_CTL1(0x00) .....	37
3.6.2	系统模式设置寄存器 SYS_MODE(0x04) .....	38
3.6.3	系统掉电控制寄存器 SYS_PD (0x08).....	38
3.6.4	系统 OSC 控制寄存器 2OSC_CTL2(0x10) .....	39
3.6.5	系统复位寄存器 SYS_RST(0x14).....	41
3.6.6	系统映射控制寄存器 SYS_MAPCTL(0x18).....	42
3.6.7	模块使能 0 寄存器 MOD0_EN (0x1c).....	42
3.6.8	模块使能 1 寄存器 MOD1_EN(0x20).....	44
3.6.9	INTC 使能寄存器 INTC_EN(0x24) .....	44
3.6.10	KBI 使能寄存器 KBI_EN(0x28) .....	45
3.6.11	器件 ID 寄存器 CHIP_ID(0x2C).....	46
3.6.12	系统控制密码寄存器 SYS_PS(0x30) .....	46
3.6.13	红外配置寄存器 IRFR_CTL (0x34).....	46
3.6.14	时钟校正配置寄存器 TRIM_CFG1( 0x78).....	47
3.6.15	时钟校正启动寄存器 TRIM_START( 0x7C).....	47
<b>4</b>	<b>处理器架构.....</b>	<b>49</b>
4.1	概述.....	49
4.2	CORTEX-M0 处理器.....	49
4.2.1	中断配置.....	50
4.3	MCU 存储映射.....	52
4.3.1	SRAM .....	54

4.3.2	FLASH .....	54
4.3.3	EEPROM .....	55
4.4	中断应用.....	55
<b>5</b>	<b>RTC .....</b>	<b>57</b>
5.1	概述.....	57
5.2	特点.....	57
5.3	寄存器描述.....	57
5.3.1	RTC 控制寄存器 RTC_CTL(0x00).....	58
5.3.2	秒寄存器 RTC_SC(0x04).....	59
5.3.3	分钟寄存器 RTC_MN(0x08) .....	60
5.3.4	小时寄存器 RTC_HR(0x0c) .....	60
5.3.5	日期寄存器 RTC_DT(0x10) .....	60
5.3.6	月份寄存器 RTC_MO(0x14) .....	60
5.3.7	年份寄存器 RTC_YR(0x18) .....	60
5.3.8	星期寄存器 RTC_DW(0x1c) .....	61
5.3.9	RTC 定时寄存器 1 RTC_CNT1(0x20) .....	61
5.3.10	RTC 定时寄存器 2 RTC_CNT2(0x24) .....	61
5.3.11	秒闹钟寄存器 RTC_SCA(0x28).....	61
5.3.12	分钟闹钟寄存器 RTC_MNA(0x2c).....	62
5.3.13	小时闹钟寄存器 RTC_HRA(0x30) .....	62
5.3.14	RTC 中断使能寄存器 RTC_IE(0x34) .....	62
5.3.15	RTC 中断标志寄存器 RTC_IF(0x38).....	63
5.3.16	当前温度寄存器 RTC_TEMP(0x3c) .....	64
5.3.17	LOSC 配置寄存器 LOSC_CFG1(0x6c) .....	64
5.3.18	VBAT 域 IO 口配置寄存器 VBAT_IOMODE(0x88) .....	65
5.3.19	VBAT 域 IO 口配置寄存器写使能 VBAT_IOWEN(0x8C) .....	65
5.4	RTC 时钟读写步骤 .....	66
5.5	RTC 校准步骤 .....	66
5.6	RTC 定时器操作步骤 .....	66
<b>6</b>	<b>WDT .....</b>	<b>67</b>
6.1	概述.....	67
6.2	看门狗定时器的配置.....	67
6.3	寄存器描述.....	68
6.4	WDT 操作步骤.....	68
<b>7</b>	<b>LCD .....</b>	<b>69</b>
7.1	概述.....	69
7.1.1	扫描时钟频率.....	69
7.1.2	闪烁模式.....	69
7.1.3	LCD 驱动波形.....	70
7.1.4	偏置电压.....	75
7.1.5	LCD BUFFER 映射.....	76
7.2	寄存器描述.....	76

7.2.1	LCD 控制寄存器 LCD_CTL(0x00).....	76
7.2.2	LCD 状态寄存器 LCD_STATUS(0x04).....	77
7.2.3	LCD 时钟控制寄存器 LCD_CLKDIV(0x08) .....	78
7.2.4	LCD 闪烁控制寄存器 LCD_BLINK(0x0c) .....	78
7.2.5	LCD 闪烁控制寄存器 LCD_PS(0x10).....	78
7.2.6	LCD 内部电阻串控制寄存器 LCD_RESCTL(0x14).....	78
7.2.7	LCD 数据寄存器 LCD_BUF[i] (8 位寄存器) .....	80
7.3	LCD 操作步骤.....	80
<b>8</b>	<b>定时器.....</b>	<b>81</b>
8.1	概述.....	81
8.2	功能框图.....	82
8.3	寄存器描述.....	82
8.4	典型应用.....	87
8.4.1	自动运行模式, 定时功能.....	87
8.4.2	输入捕获模式, 脉宽测量功能.....	88
8.4.3	比较输出模式, 方波输出功能.....	88
8.4.4	比较输出模式, PWM 输出功能.....	89
8.4.5	从模式, 外部清零和门控功能.....	91
8.5	操作步骤.....	91
<b>9</b>	<b>模拟外设.....</b>	<b>92</b>
9.1	特点.....	92
9.2	寄存器.....	92
9.2.1	ADC 控制寄存器 SAR_CTL(0x00) .....	92
9.2.2	SAR-ADC 启动寄存器 SAR_START(0x04).....	94
9.2.3	SAR-ADC 状态寄存器 SAR_STATUS(0x08).....	94
9.2.4	ADC DAT 寄存器 SAR_DAT(0x0c).....	94
9.2.5	LVD 控制寄存器 LVD_CTL(0x10).....	94
9.2.6	LVD 状态寄存器 LVD_STAT(0x14) .....	95
9.2.7	ANA_PAD .....	96
9.2.8	ANA_RST.....	96
9.3	ADC 电压检测步骤 .....	96
9.4	VBAT 电压检测 .....	96
9.5	低电压检测应用.....	97
<b>10</b>	<b>GPIO .....</b>	<b>98</b>
10.1	概述.....	98
10.1.1	PA 口模式寄存器 PMA (输入或者输出) (0x00) .....	99
10.1.2	PA 口数据寄存器 PA (0x04) .....	100
10.1.3	PA 口复用 0 寄存器 PCA0 (0x08) .....	100
10.1.4	PA 口复用 1 寄存器 PCA1 (0x0C) .....	101
10.1.5	PA 口上拉选择寄存器 PUA (0x10) .....	102
10.1.6	PA 口输入模式配置寄存器 PIMA (0x14) .....	103
10.1.7	PA 口输入使能寄存器 PIEA (0x18) .....	103

10.1.8	PB 口模式寄存器 PMB (输入或者输出) (0x1C)	103
10.1.9	PB 口数据寄存器 PB (0x20)	104
10.1.10	PB 口复用寄存器 PCB (0x24)	104
10.1.11	PB 口复用寄存器 2PCB2 (0x4CH)	105
10.1.12	PB 口上下拉选择寄存器 PUB (0x28)	107
10.1.13	PB 口输入模式寄存器 PIMB (0x2C)	107
10.1.14	PB 口输入使能寄存器 PIEB (0x30)	108
10.1.15	PC 口模式寄存器 PMC (输入或者输出) (0x34)	108
10.1.16	PC 口数据寄存器 PC (0x38)	108
10.1.17	PC 口复用寄存器 PCC (0x3C)	109
10.1.18	PC 口上下拉选择寄存器 PUC (0x40)	109
10.1.19	PC 口输入使能寄存器 PIEC (0x44)	110
10.1.20	PC 口输入模式寄存器 PIMC (0x48)	110
10.1.21	SEGC0M 口复用寄存器 PCE (0x60)	110
10.1.22	PA 口数据置位寄存器 PASET (0x64)	111
10.1.23	PA 口清零置位寄存器 PACLR (0x68)	111
10.1.24	PB 口数据置位寄存器 PBSET (0x6C)	111
10.1.25	PB 口清零置位寄存器 PBCLR (0x70)	112
10.1.26	PC 口数据置位寄存器 PCSET (0x74)	112
10.1.27	PC 口清零置位寄存器 PCCLR (0x78)	113
10.2	GPIO 操作步骤	113
<b>11</b>	<b>外部中断控制器</b>	<b>114</b>
11.1	概述	114
11.2	寄存器描述	114
<b>12</b>	<b>KBI</b>	<b>117</b>
12.1	特性	117
12.2	寄存器描述	117
12.3	KBI 操作步骤	118
<b>13</b>	<b>UART</b>	<b>119</b>
13.1	概述	119
13.2	寄存器描述	119
13.3	UART 数据接收及发送操作步骤	122
<b>14</b>	<b>ISO7816</b>	<b>124</b>
14.1	概述	124
14.2	寄存器描述	124
14.3	7816 与 ESAM 通讯操作步骤	132
14.4	7816 与卡通讯操作步骤	132
<b>15</b>	<b>IIC 接口</b>	<b>134</b>
15.1	概述	134
15.2	寄存器描述	134

<b>16</b>	<b>SPI 接口 .....</b>	<b>137</b>
16.1	概述.....	137
16.2	功能描述.....	137
16.3	寄存器描述.....	138
<b>17</b>	<b>选项字节.....</b>	<b>141</b>
17.1	芯片保护设置.....	141
17.2	WDT 设置.....	141
17.3	RTC 设置.....	142
<b>18</b>	<b>编程支持.....</b>	<b>143</b>
18.1	概述.....	143
18.2	FLASH 保护机制 .....	143
18.3	在系统编程 (ISP) .....	143
18.3.1	ISP 通讯协议.....	144
18.3.2	使用的资源.....	144
18.3.3	ISP 命令 .....	145
18.3.4	ISP 返回代码.....	150
18.4	在应用编程 (IAP) .....	151
18.4.1	IAP 命令 .....	151
18.4.2	IAP 使用 .....	152
18.5	量产平台.....	152
<b>19</b>	<b>封装尺寸.....</b>	<b>153</b>



# 1 概述

## 1.1 简介

**RN8318:** 集成 ARM Cortex-M0 内核、512KB Flash 和 32KB SRAM、独立供电硬件温补 RTC, LCD 支持 8\*36/6\*38/4\*40。封装 LQFP128, 典型应用领域: 国网 698 协议三相表(使用 LCD)、海外高端三相表。

**RN8615:** 集成 ARM Cortex-M0 内核、512KB Flash 和 32KB SRAM, 独立供电硬件温补 RTC, LCD 支持 8\*30/6\*32/4\*34。封装 LQFP100, 典型应用领域: 国网 698 协议单相表(使用 LCD)、国网 698 协议三相表(不使用 LCD)、海外高端单三相表。

**RN8613:** 集成 ARM Cortex-M0 内核、256KB Flash 和 32KB SRAM, 独立供电硬件温补 RTC, LCD 支持 8\*30/6\*32/4\*34。封装 LQFP100, 典型应用领域: 国网 698 协议单相表(使用 LCD)、海外高端单三相表。

**RN8612:** 集成 ARM Cortex-M0 内核、128KB Flash、8KB SRAM、32KB EEPROM, 独立供电硬件温补 RTC, LCD 支持 8\*30/6\*32/4\*34。封装 LQFP100, 典型应用领域: 南网表、海外单相表。

**RN8611:** 集成 ARM Cortex-M0 内核、256KB Flash 和 32KB SRAM, 独立供电硬件温补 RTC, 不支持 LCD。封装 LQFP64, 典型应用领域: 国网 698 协议单相表(不使用 LCD)。

## 1.2 产品特点

基本特点:

- 高集成: 32bit ARM M0 CPU + 最大 512KB Flash/32KB SRAM+独立供电硬件温补 RTC+ LCD 控制器;
- 宽电压: 典型条件下 2.3V~5.5V 电压范围保证 CPU 正常工作;  
典型条件下 2.7V~5.5V 电压范围保证 RTC 精度;  
典型条件下 1.8V~5.5V 电压范围万年历时间保持;
- 高性能: 32768 单晶振条件下, CPU 最高工作频率到 14.7456MHz (32KHz、1.8432Mhz、7.3728MHz、14.7456MHz 可选)  
32768 晶振+外部高频晶振条件下, CPU 最高工作频率到 29.4912MHz (32KHz、1.8432Mhz、7.3728MHz、14.7456MHz、29.4912MHz 可选)
- 低功耗:  
CPU 子系统工作在 32KHz 下功耗优于 18 $\mu$ A (with cache);  
休眠模式下芯片整体功耗约为 7 $\mu$ A (RTC 自动温补; ram 保持; CPU 及数字外设不掉电; 中断唤醒);  
VBAT 域功耗典型值约为 1.5 $\mu$ A;  
LCD 功耗约为 5 $\mu$ A, 常显芯片整体功耗 13 $\mu$ A (电阻串分压模式, 含 6COM 显示屏功耗);  
LCD 功耗约为 10 $\mu$ A, 常显芯片整体功耗 18 $\mu$ A (电荷泵模式, 含 8COM 显示屏功耗);
- 高精度: RTC 在全温度范围 (-40 度~85 度) 内秒脉冲误差小于 10ppm;  
RTC 在电池供电情况下依然保证秒脉冲的准确度;
- 封装形式: RN8318           LQFP128  
                  RN8613/RN8612   LQFP100  
                  RN8611           LQFP64

处理器相关:

- ARM Cortex-M0 内核
- RN8318: 512KB FLASH 存储器; RN8615: 512KB FLASH 存储器;
- RN8613/RN8611: 256KB FLASH 存储器; RN8612: 128KB FLASH 存储器;
- RN8612 提供 8KBytes SRAM (全部开放给用户使用)
- RN8612 提供 32KBytes EEPROM, 支持 WORD 编程 (注意不是 BYTE 编程), 100 万次。

- 其他系列提供 32KBytes SRAM，其中 2KBytes 作为 Cache 使用；
- 单 cycle 乘法器（32bit\*32bit）；
- CMO 内嵌系统定时器
- 支持外部中断等多种唤醒方式
- 提供完善的集成开发软硬件环境

#### RTC:

- 硬件自动温补，满足标准要求的精度和功耗，温补不需要 CPU 参与。
- 电池供电下能保证精确 1Hz 输出
- 温度传感器：-25℃~70℃ 范围内测温精度为±1℃
- RTC 万年历及自动温补电路独立使用 VBAT 引脚供电
- VBAT 域功耗典型值优于 1.5uA
- 开启温补时间约为 2ms，温度典型功耗 250 μ A，如果 30 秒开启一次，平均功耗为 0.016 μ A；

#### LCD:

- RN8318: 4\*40、6\*38、8\*36；
- RN8615/RN8613/RN8612: 4\*34、6\*32、8\*30；
- RN8612 只支持电阻分压模式，其他系列支持 ChargePUMP 和内置电阻列分压模式可选，电阻列分压模式方案下推荐使用 6com/3V 屏；
- 如果需要支持更多的段数，可选择 8COM 5V 屏（电荷泵方式）；如果需要更低的显示功耗，可选择 6COM 3V 屏（电阻串分压方式）。二者在硬件上是兼容的，只需要对软件配置作出更改即可。
- LCD 模块功耗优于 5μA（电阻列分压方案）。

#### 其他外设:

- 高速 GPIO，支持与不同电压外设器件的接口：RN8318: 93 个；RN8615/RN8613/RN8612: 64 个；RN8611:50 个
- 10bit ADC: 温度传感器/电池电压检测/通用 ADC 分时复用
- 电压检测 LVD: 检测芯片电源电压；检测外部电压
- 两个低功耗比较器 CMP: 检测外部电压
- 扩展定时器: 2 个 32bit 定时器
- UART: 6 个，支持自动波特率，支持红外调制，支持 UART 唤醒，支持电平反转
- 7816 口: 2 个
- I2C: 1 个
- SPI: 1 个
- 硬件看门狗
- 按键中断: 8 个，管脚复用
- 外部中断口: 8 个，管脚复用

#### 加密:

- 硬件真随机数产生器，符合美国 NIST 的 FIPS 140-2 标准；
- AES128/192/256 硬件加密，符合 FIPS197 标准，支持 EBC/CBC/CTR/GCM/GMAC 模式；
- ECC192 硬件加速器；
- RN8612 不支持加密功能；
- 加密说明文档见锐能微应用笔记；

## 锐能微 MCU 型号划分

MCU 系列	FLASH	RAM	EEPROM	加密	封装
RN8318	512KB	32KB	X	Y	LQFP128
RN8615	512KB	32KB	X	Y	LQFP100
RN8613	256KB	32KB	X	Y	LQFP100
RN8611	256KB	32KB	X	Y	LQFP64
RN8612	128KB	8KB	32KB	X	LQFP100

## RN8612 与 RN8613/RN8615 差异表:

	RN8612	RN8613/RN8615
FLASH	128KB	256KB/512KB
RAM	8KB (全部给客户使用)	32KB (30KB 客户使用, 2KB cache 使用)
EEPROM	32KB	无
加密	无	有
LCD	仅支持电阻分压, 不支持 5V 屏, 建议使用 6COM 3/3.3V 屏	支持电阻分压和电荷泵两种方式
管脚定义差异	PIN50/51 是 NC	PIN50/51 是 LCDVP2、LCDVP1, LCD 使用电荷泵方式时需要串接 100nF 电容, 使用电阻串分压方式时可悬空。
	SIG/IO 复用引脚, 做为 IO 口使用时, 为普通 IO 口结构	SIG/IO 复用引脚, 做为 IO 口使用时, 为开漏结构。
	PIN10~PIN13 为 NC	PIN10~PIN13 为 P104~P107

备注: RN8612 与 RN8613/RN8615 管脚兼容, 主要差异见上表。

### 1.3 实现架构

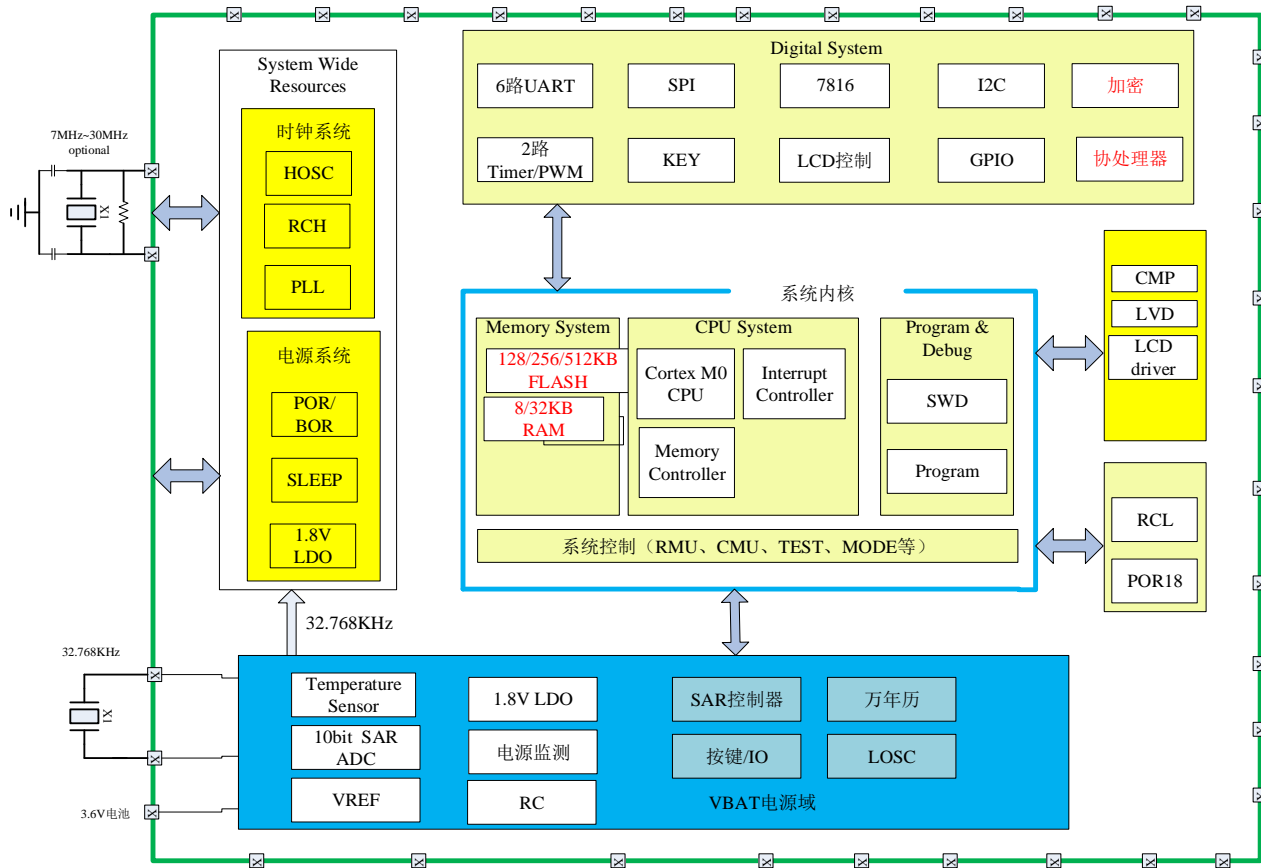


图 1.1RN 系列 MCU 结构框图

### 1.4 管脚排列

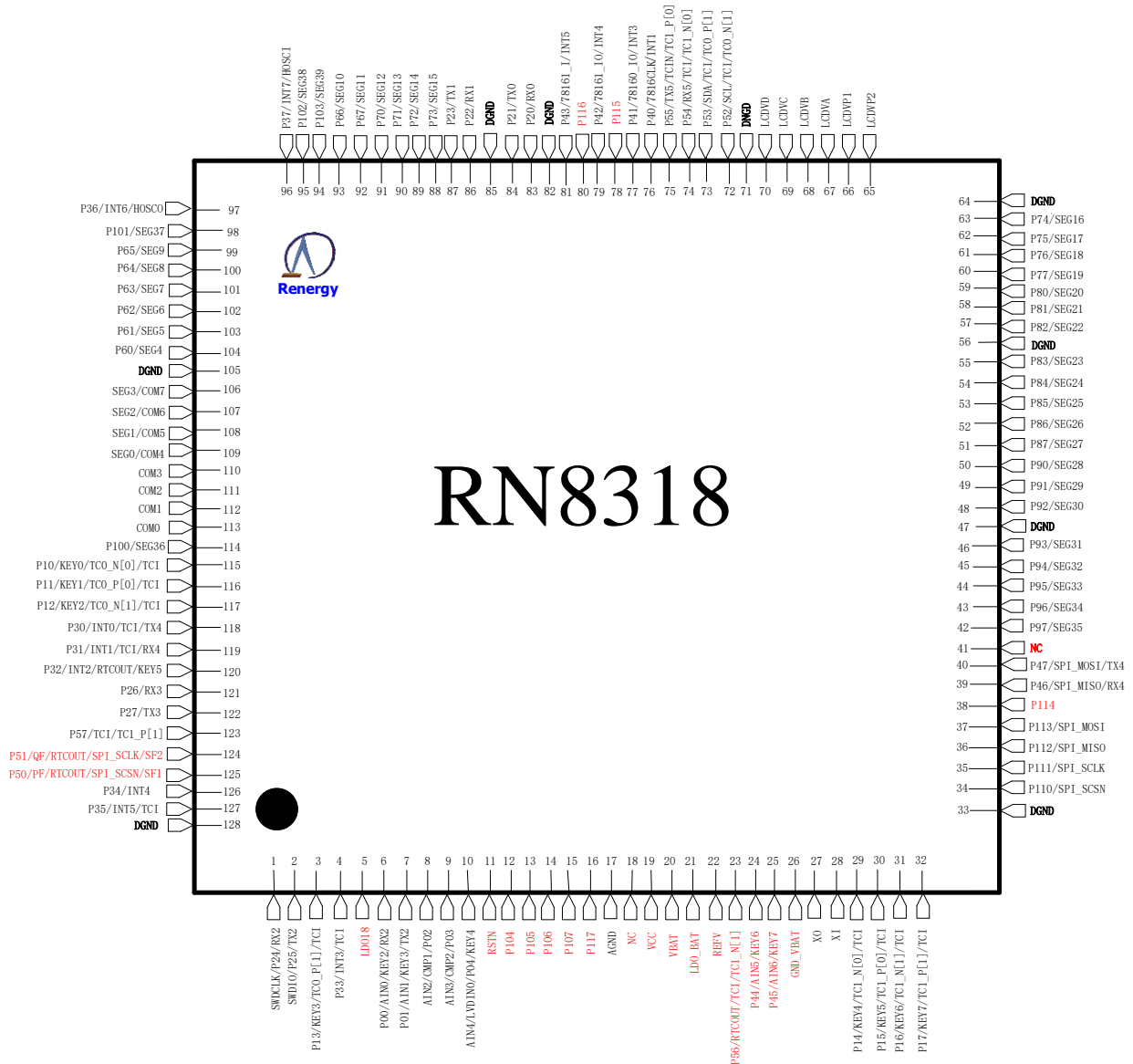


图 1.2 RN8318 管脚排列图

备注:

VBAT 域管脚有: VBAT、LDO\_VBAT、REFV、P56、P44、P45、GND\_VBAT、X0、XI。

注意 P56/P44/P45 引脚输出高电平等于 VBAT 电压。

红色字体的管脚是 RN8318 与 RN8312 有差异的管脚。

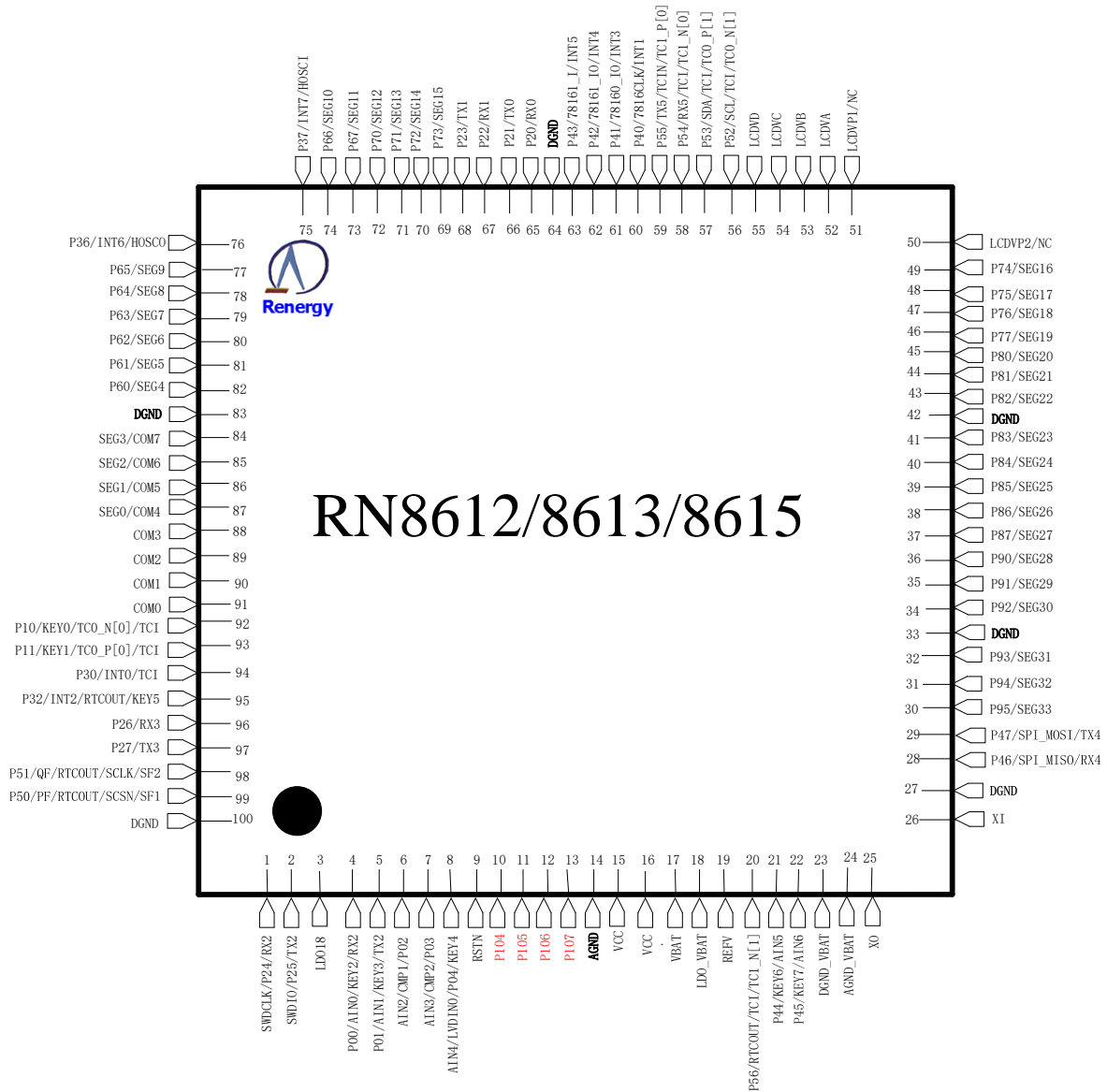


图 1.3RN8612/RN8613/RN8615 管脚排列图

备注:

1. VBAT 域管脚有: VBAT、LDO\_VBAT、REFV、P56、P44、P45、AGND\_VBAT、DGND\_VBAT、XO、XI。
2. 注意 P56/P44/P45 引脚输出高电平等于 VBAT 电压。
3. RN8612 的 PIN50/51 是 NC, RN8613/RN8615 的 PIN50/51 是 LCDVP2/LCDVP1。
4. RN8613/8615 的 SEG/IO 复用引脚做为 IO 口使用时为开漏结构, RN8612 为普通结构。
5. RN8612 的 PIN10~PIN13 为 NC, RN8613/RN8615 的 PIN10~PIN13 是 P104~P107 口。

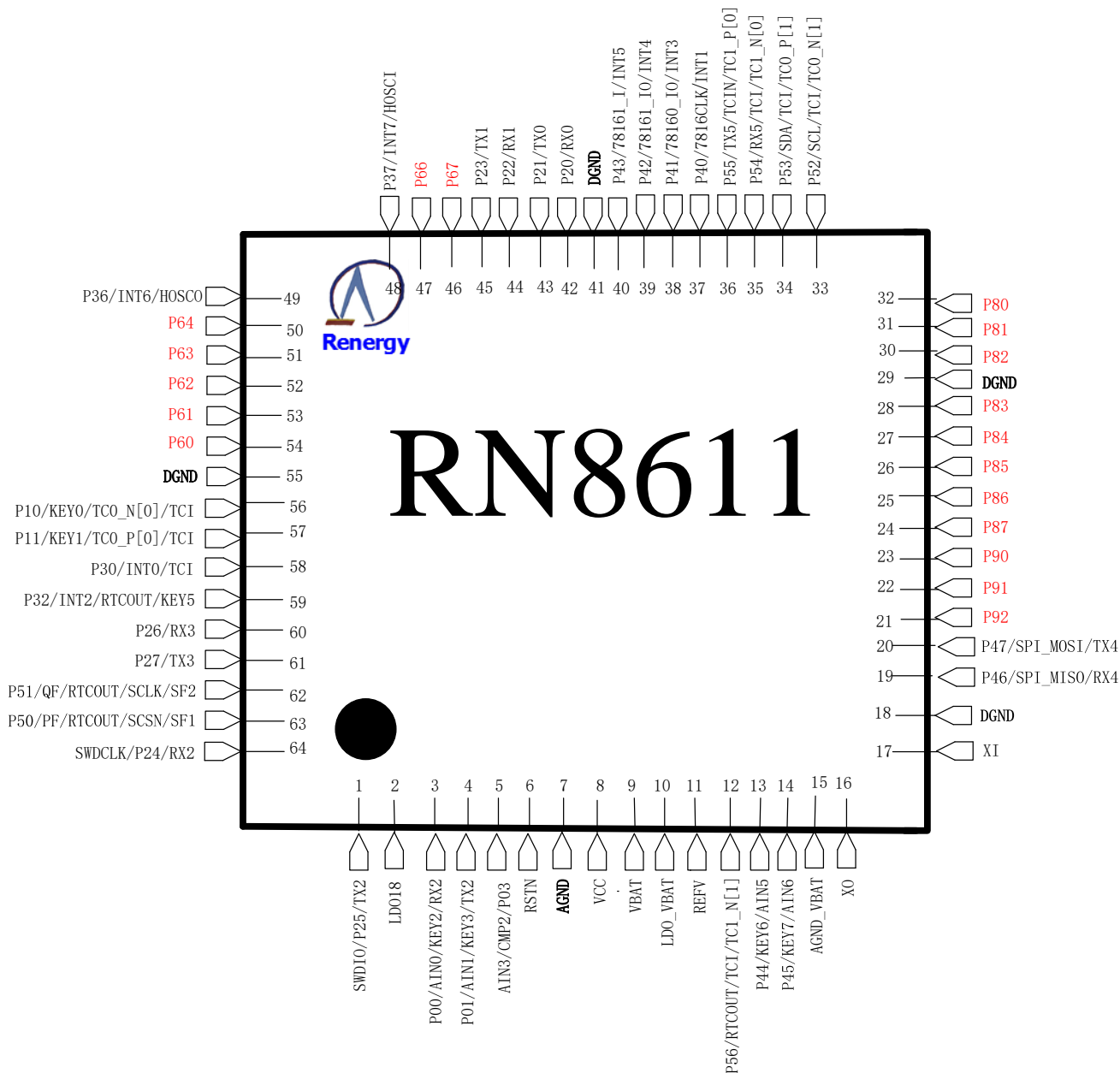


图 1.4 RN8611 管脚排列图

**备注:**

- 1. RN8611 标注红色字体的 IO 口是开漏结构，做为输出时必须外接上拉电阻。
- 2. VBAT 域管脚有：VBAT、LDO\_VBAT、REFV、P56、P44、P45、AGND\_VBAT、XO、XI。
- 2. 注意 P56/P44/P45 引脚输出高电平等于 VBAT 电压。

引脚类型说明:

类型	模 拟 A	双 向 B	输 入 I	输 出 O	上 拉 U	施 密 特 S	TTL/CMOS L	OpenDrain D	晶振 X	SEG G	COM M	驱 动
PBUS6		√	√	√	√	√						6mA

PBULD3		√	√	√	√		√	√				3mA
PABULD3	√	√	√	√	√		√	√				3mA
PABUS3	√	√	√	√		√						3mA
PUXI			√						√			
PBUSG3		√	√	√	√	√		√		√		3mA
PAM	√			√							√	
PAGM	√			√						√	√	

引脚说明:

RN8612 RN8613 RN8615	RN8318	RN8611	标识	特性	功能描述
1	1	64	SWDCLK/P24/RX2	PBULD3	SWD 时钟、P24 口、RX2 输入复用引脚
2	2	1	SWDIO/P25/TX2	PBULD3	SWD 数据口、P25 口、TX2 输入复用引脚
	3		P13/KEY3/TC0_P[1]/TCIN	PBULD3	IO 口、KEY 输入、定时器输出、定时器输入复用；上拉可选、TTL/CMOS 电平可选、漏极开路可选。
	4		P33/INT3/TCIN	PBUS6	P3 口与外部中断、定时器复用；上拉可选、施密特输入、6mA 驱动能力。
3	5	2	LDO18	电源	内置 1.8V LDO 的输出, 应外接 1μf 电容并联 0.1 μf 电容去耦。
4	6	3	P00/AIN0/KEY2/RX2	PABUS3	P00 口、SAR-ADC 输入、KEY2、RX2 复用引脚
5	7	4	P01/AIN1/KEY3/TX2	PABUS3	P01 口、SAR-ADC 输入、KEY3、TX2 复用引脚
6	8	5	AIN2/CMP1/P02	PABUS3	SAR-ADC 输入、比较器 1 输入、P02 复用；
7	9		AIN3/CMP2/P03	PABUS3	SAR-ADC 输入、低功耗比较器 2 输入、P03 复用；
8	10		AIN4/LVDIN0/P04/KEY4	PABUS3	SAR-ADC 输入、LVDIN 输入、P04、KEY4 复用；
9	11	6	RSTN	复位	外部复位引脚, 内置约 50K 上拉电阻；
10	12		P104	PBUS6	P104 口, 上拉可选、施密特输入、6mA 驱动能力。
11	13		P105	PBUS6	P105 口, 上拉可选、施密特输入、6mA 驱动能力。



12	14		P106	PBUS6	P106 口, 上拉可选、施密特输入、6mA 驱动能力。
13	15		P107	PBUS6	P107 口, 上拉可选、施密特输入、6mA 驱动能力。
	16		P117	PBULD3	P117 口, 上拉可选、施密特输入、6mA 驱动能力。
14	17	7	AGND	地	模拟地
15			VCC	电源	2.3V~5.5V 电源输入, 应外接 4.7μf 电容并联 0.1μf 电容去耦。(同 16 脚)
	18		NC		不连接
16	19	8	VCC	电源	2.3V~5.5V 电源输入, 应外接 4.7μf 电容并联 0.1μf 电容去耦。
17	20	9	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 SARADC 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。
18	21	10	LDO_VBAT	电源	VBAT 域 LDO 输出, 外部需接 0.22μf 电容;
19	22	11	REFV	参考电压	SARADC 的内置基准输出, 外部需接 0.22μf 电容;
20	23	12	P56/RTCOUT/TCI/TC1_N [1]	PBULD3	P56、RTCOUT、TCIO 复用 RTCOUT 复用关系由 RTC 章节寄存器决定, 与 GPIO 章节复用配置寄存器无关。当 VCC 无电, VBAT 上电时, 该引脚默认输出 1Hz。一旦 VCC 上电, 1Hz 输出会被关闭。
21	24	13	P44/AIN5/KEY6	PABULD3	P44、AIN5、KEY6 复用。 如果要使用 AIN5 功能, 需要配置 ANA_PAD (第 9 章) 寄存器。
22	25	14	P45/AIN6/KEY7	PABULD3	P45、AIN6、KEY7 复用。 AIN6 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。 如果要使用 AIN6 功能, 需要配置 ANA_PAD (第 9 章) 寄存器。
23	26	15	AGND_VBAT	地	VBAT 域模拟地
24			DGND_VBAT	地	VBAT 域数字地
25	27	16	XO	时钟	32.768KHz 无源晶振输出和输入。 不需要外接电阻和电容, 需要用地线将之隔离。
26	28	17	XI	时钟	
	29		P14/KEY4/TC1_N[0]/TCIN	PBULD3	IO 口、KEY 输入、定时器输出、定时器输入复用; 上拉可选、TTL/CMOS 电平可选、漏极开路可选。
	30		P15/KEY5/TC1_P[0]/TCIN	PBULD3	IO 口、KEY 输入、定时器输出、定时器输入复用; 上拉可选、TTL/CMOS 电平可选、漏极开路可选。
	31		P16/KEY6/TC1_N[1]/TCIN	PBULD3	IO 口、KEY 输入、定时器输出、定时器输入复用; 上拉可选、TTL/CMOS 电平可选、漏极开路可选。
	32		P17/KEY7/TC1_P[1]/TCIN	PBULD3	IO 口、KEY 输入、定时器输出、定时器输入复用; 上拉可选、TTL/CMOS 电平可选、漏极开路可选。
27	33	18	DGND	地	数字地
	34		P110/SPI-SCSN	PBULD3	P11 口与 SPI 复用;
	35		P111/SPI-SCLK	PBULD3	P11 口的 SPI 与 P4 口的 SPI 不能同时使用;
	36		P112/SPI-SDI	PBULD3	上拉可选、TTL/CMOS 电平可选、漏极开路可选。
	37		P113/SPI-SDO	PBULD3	上拉可选、TTL/CMOS 电平可选、漏极开路可选。
	38		P114	PBULD3	P114 口, 上拉可选、TTL/CMOS 电平可选、漏极开路可选。
28	39	19	P46/SPI_MISO/RX4	PBULD3	P4 口、SPI、UART4 复用;
29	40	20	P47/SPI_MOSI/TX4	PBULD3	上拉可选、TTL/CMOS 电平可选、漏极开路可选。
	41		NC		不连接
	42		P97/SEG35	PBUSG3	LCD/GPIO 复用

	43		P96/SEG34	PBUSG3	LCD/GPIO 复用
30	44		P95/SEG33	PBUSG3	LCD/GPIO 复用
31	45		P94/SEG32	PBUSG3	LCD/GPIO 复用
32	46		P93/SEG31	PBUSG3	LCD/GPIO 复用
33	47		DGND	地	地
34	48	21	P92/SEG30	PBUSG3	LCD/GPIO 复用
35	49	22	P91/SEG29	PBUSG3	LCD/GPIO 复用
36	50	23	P90/SEG28	PBUSG3	LCD/GPIO 复用
37	51	24	P87/SEG27	PBUSG3	LCD/GPIO 复用
38	52	25	P86/SEG26	PBUSG3	LCD/GPIO 复用
39	53	26	P85/SEG25	PBUSG3	LCD/GPIO 复用
40	54	27	P84/SEG24	PBUSG3	LCD/GPIO 复用
41	55	28	P83/SEG23	PBUSG3	LCD/GPIO 复用
42	56	29	DGND	地	地
43	57	30	P82/SEG22	PBUSG3	LCD/GPIO 复用
44	58	31	P81/SEG21	PBUSG3	LCD/GPIO 复用
45	59	32	P80/SEG20	PBUSG3	LCD/GPIO 复用
46	60		P77/SEG19	PBUSG3	LCD/GPIO 复用
47	61		P76/SEG18	PBUSG3	LCD/GPIO 复用
48	62		P75/SEG17	PBUSG3	LCD/GPIO 复用
49	63		P74/SEG16	PBUSG3	LCD/GPIO 复用
	64		DGND	地	地
50	65		LCDVP2	模拟	模拟输出，LCDVP2 和 LCDVP1 之间应该连接一个 100nF 的电容。 LCD 如果使用电阻分压方式，可悬空。
51	66		LCDVP1	模拟	模拟输出，LCDVP2 和 LCDVP1 之间应该连接一个 100nF 的电容。 LCD 如果使用电阻分压方式，可悬空。
52	67		LCDVA	模拟	LCD 电压输出，需要外接 470nF 电容。 不管使用电荷泵还是电阻分压方式，都需要外接此电容。
53	68		LCDVB	模拟	LCD 电压输出，需要外接 470nF 电容。 不管使用电荷泵还是电阻分压方式，都需要外接此电容。
54	69		LCDVC	模拟	LCD 电压输出，需要外接 470nF 电容。 不管使用电荷泵还是电阻分

					压方式，都需要外接此电容。
55	70		LCDVD	模拟	LCD 电压输出，需要外接 470nF 电容。 不管使用电荷泵还是电阻分压方式，都需要外接此电容。
	71		DGND	地	地
56	72	33	P52/SCL/TCIN/TC0_N[1]	PBULD3	P5 口、I2C、TCIO 复用。
57	73	34	P53/SDA/TCIN/TC0_P[1]	PBULD3	上拉电阻可选、TTL/CMOS 输入可选、漏极开路可选；
58	74	35	P54/RX5/TCIN/TC1_N[0]	PBULD3	P54、UART5 输入、TCIO 复用
59	75	36	P55/TX5/TCIN/TC1_P[0]	PBULD3	P55、UART5 输出、TCIO 复用
60	76	37	P40/7816CLK/INT1	PBULD3	IO 口、7816、中断口复用的管脚。 上拉电阻可选、TTL/CMOS 输入可选、漏极开路可选； 备注：支持两个 7816 接口。 78160_IO 是 7816 0 的双向数据口； 78161_IO 是 7816 1 的双向数据口； 另外，7816 1 有寄存器可配置为： 78161_IO 作为 7816 1 的数据输出； 78161_I 作为 7816 1 的数据输入。
61	77	38	P41/78160_IO/INT3	PBULD3	
	78		P115	PBULD3	P115 口
62	79	39	P42/78161_IO/INT4	PBULD3	
	80		P116	PBULD3	P116 口
63	81	40	P43/78161_I/INT5	PBULD3	
64	82	41	DGND	地	地
65	83	42	P20/RX0	PBULD3	UART 口与 P2 口复用；
66	84	43	P21/TX0	PBULD3	上拉可选、TTL/CMOS 电平可选、漏极开路可选。
	85		DGND		
67	86	44	P22/RX1	PBULD3	UART 口与 P2 口复用；
68	87	45	P23/TX1	PBULD3	上拉可选、TTL/CMOS 电平可选、漏极开路可选。
69	88		P73/SEG15	PBUSG3	LCD/GPIO 复用
70	89		P72/SEG14	PBUSG3	LCD/GPIO 复用

71	90		P71/SEG13	PBUSG3	LCD/GPIO 复用
72	91		P70/SEG12	PBUSG3	LCD/GPIO 复用
73	92	46	P67/SEG11	PBUSG3	LCD/GPIO 复用
74	93	47	P66/SEG10	PBUSG3	LCD/GPIO 复用
	94		P103/SEG39	PBUSG3	LCD/GPIO 复用
	95		P102/SEG38	PBUSG3	LCD/GPIO 复用
75	96	48	P37/INT7/HOSCI	PUXI	P3 口/中断口/高频晶体复用 高频晶体端口外部应串接一个 10M 欧的电阻，并联两个 15pf 的电容。
76	97	49	P36/INT6/HOSCO	PUXI	
	98		P101/SEG37	PBUSG3	LCD/GPIO 复用
77	99		P65/SEG9	PBUSG3	LCD/GPIO 复用
78	100	50	P64/SEG8	PBUSG3	LCD/GPIO 复用
79	101	51	P63/SEG7	PBUSG3	LCD/GPIO 复用
80	102	52	P62/SEG6	PBUSG3	LCD/GPIO 复用
81	103	53	P61/SEG5	PBUSG3	LCD/GPIO 复用
82	104	54	P60/SEG4	PBUSG3	LCD/GPIO 复用
83	105	55	DGND	地	地
84	106		SEG3/COM7	PAGM	SEG 与 COM 复用的端口
85	107		SEG2/COM6	PAGM	SEG 与 COM 复用的端口
86	108		SEG1/COM5	PAGM	SEG 与 COM 复用的端口
87	109		SEG0/COM4	PAGM	SEG 与 COM 复用的端口
88	110		COM3	PAM	COM 端口
89	111		COM2	PAM	COM 端口
90	112		COM1	PAM	COM 端口
91	113		COM0	PAM	COM 端口
	114		P100/SEG36	PBUSG3	LCD/GPIO 复用
92	115	56	P10/KEY0/TC0_N[0]/TCIN	PBULD3	IO 口、KEY 输入、定时器输入输出复用； 上拉可选、TTL/CMOS 电平可选、漏极开路可选。
93	116	57	P11/KEY1/TC0_P[0]/TCIN	PBULD3	IO 口、KEY 输入、定时器输入输出复用； 上拉可选、TTL/CMOS 电平可选、漏极开路可选。
	117		P12/KEY2/TC0_N[1]/TCIN	PBULD3	IO 口、KEY 输入、定时器输入输出复用； 上拉可选、TTL/CMOS 电平可选、漏极开路可选。
94	118	58	P30/INT0/ TCIN/ TX4/ ISPEN	PBUS6	IO 口、外部中断输入、定时器输入复用； 上拉可选、施密特输入、6mA

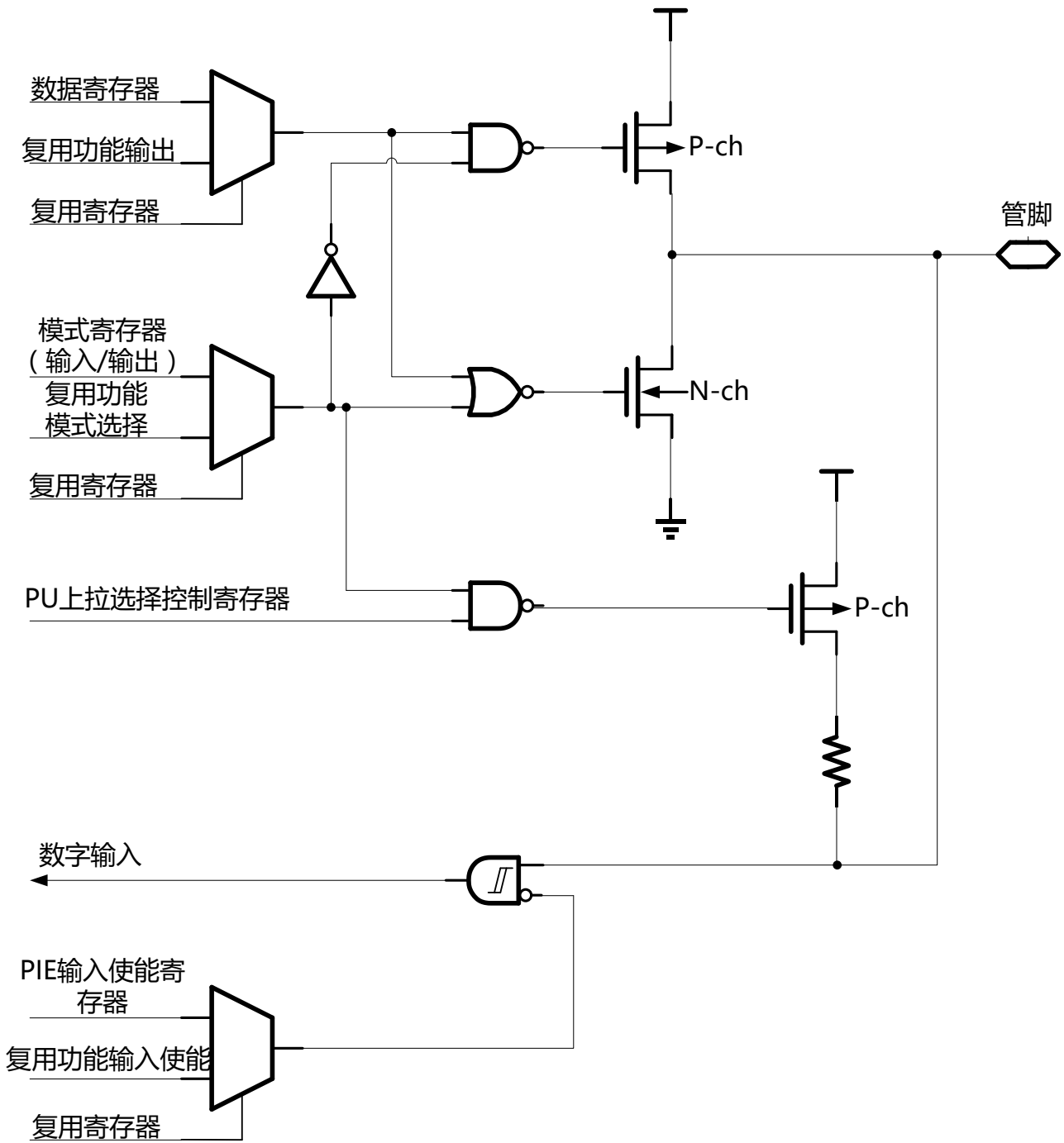
					驱动能力。 发生复位后,BOOTROM 会检测该端口的状态, 如果输入低电平, 系统会进入 ISP。实际应用时需要注意该问题。
	119		P31/INT1/TCIN/ RX4	PBUS6	IO 口、外部中断输入、定时器输入、RX4 复用;
95	120	59	P32/ INT2/ RTCOUT / KEY5	PBUS6	IO 口、外部中断输入、定时器输入复用; 上拉可选、施密特输入、6mA 驱动能力。
96	121	60	P26/RX3	PBULD3	UART 口与 P2 口复用; 上拉可选、TTL/CMOS 电平可选、漏极开路可选。
97	122	61	P27/TX3	PBULD3	UART 口与 P2 口复用; 上拉可选、TTL/CMOS 电平可选、漏极开路可选。
	123		P57/TCIN/TC1_P[1]	PBULD3	P57 口与 TC 输出复用
98	124	62	P51/QF/RTCOUT/SPI_SCLK/SF2	PBUS6	P51 口、QF、RTC 输出、SPI_SCLK、SF2 复用; 上拉可选、施密特输入、6mA 驱动能力 其中 PF 是 INT0 输入后直接输出; SF2 是 INT7 输入后直接输出; 此功能用于从计量芯片输入脉冲后直接再输出做校验用。
99	125	63	P50/PF/RTCOUT/SPI_SC SN/SF1	PBUS6	P50 口、PF、RTC 输出、SPI_SCSN、SF1 复用; 上拉可选、施密特输入、6mA 驱动能力 其中 PF 是 INT2 输入后直接输出; SF1 是 INT6 输入后直接输出; 此功能用于从计量芯片输入脉冲后直接再输出做校验用。
	126		P34/INT4	PBUS6	P34 与外部中断复用引脚;
	127		P35/ INT5/ TCIN	PBUS6	P35 与外部中断、定时器输入复用;

					上拉可选、施密特输入、6mA 驱动能力。
100	128		DGND	地	地

#### 关键管脚说明：

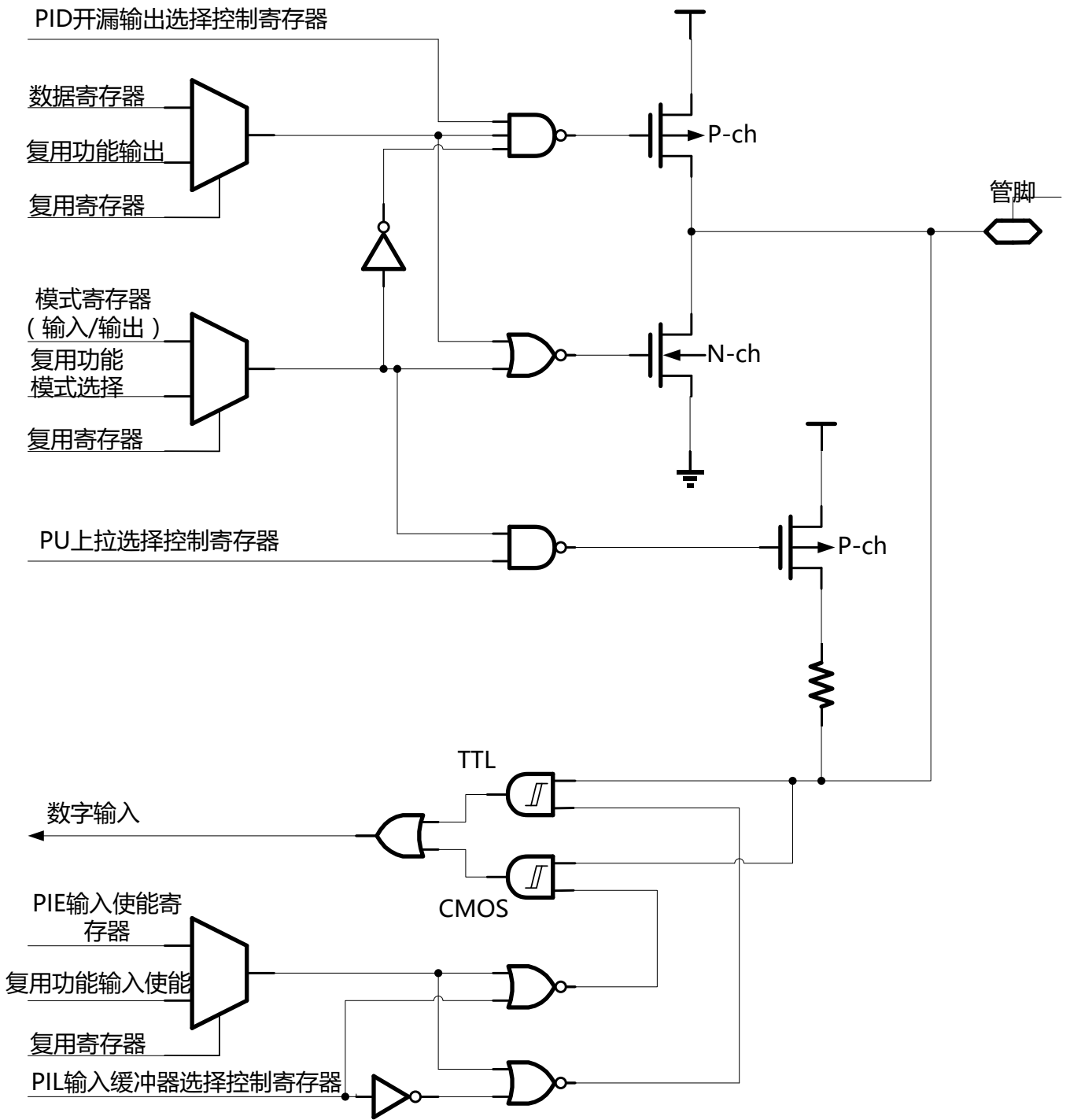
- 1、AGND是模拟地，DGND是数字地；实际使用时可根据经验合理布局。
- 2、LDO18：应外接1 $\mu$ f电容并联0.1 $\mu$ f电容去耦。
- 3、REFV是SAR-ADC的参考输入，外部应接0.22 $\mu$ f电容。
- 4、VBAT是3.6V电池或超级电容输入引脚：建议外接RC滤波，R:10 $\Omega$ ，C:1 $\mu$ F。
- 5、VCC是主电输入，正常工作范围2.3V~5.5V，应外接4.7 $\mu$ f电容并联0.1 $\mu$ f电容去耦。
- 6、LDO\_VBAT: VBAT域LDO输出，外部需接0.22 $\mu$ f电容。
- 7、XO和XI之间跨接32.768KHz晶体，最好用地线将之隔离，不需外接电阻和电容。
- 8、LCDVD、LCDVC、LCDVB、LCDVA是LCD电压输出，每个管脚都应外接470nf电容。
- 9、LCD CHARGE PUMP模式：LCDVP1和LCDVP2之间应连接一个100nf的电容；内部电阻列模式：LCDVP1、LCDVP2可悬空，或LCDVP1悬空、LCDVP2接地。
- 10、除RN8612外，SEG/IO复用引脚，做为IO口使用时，是开漏结构，输出必须接外部上拉。
- 11、对VBAT引脚和AIN6引脚输入的电压进行测量时，内部有两个300K电阻做分压，输入电压会被缩小一半输入到SAR ADC。
- 12、P36、P37 做为 IO 口使用时，只能做为输入引脚，不能做为输出。
- 13、P56/P44/P45 引脚在 VBAT 电源域，输出高电平与 VBAT 相等。

### 1.5 IO 口功能框图

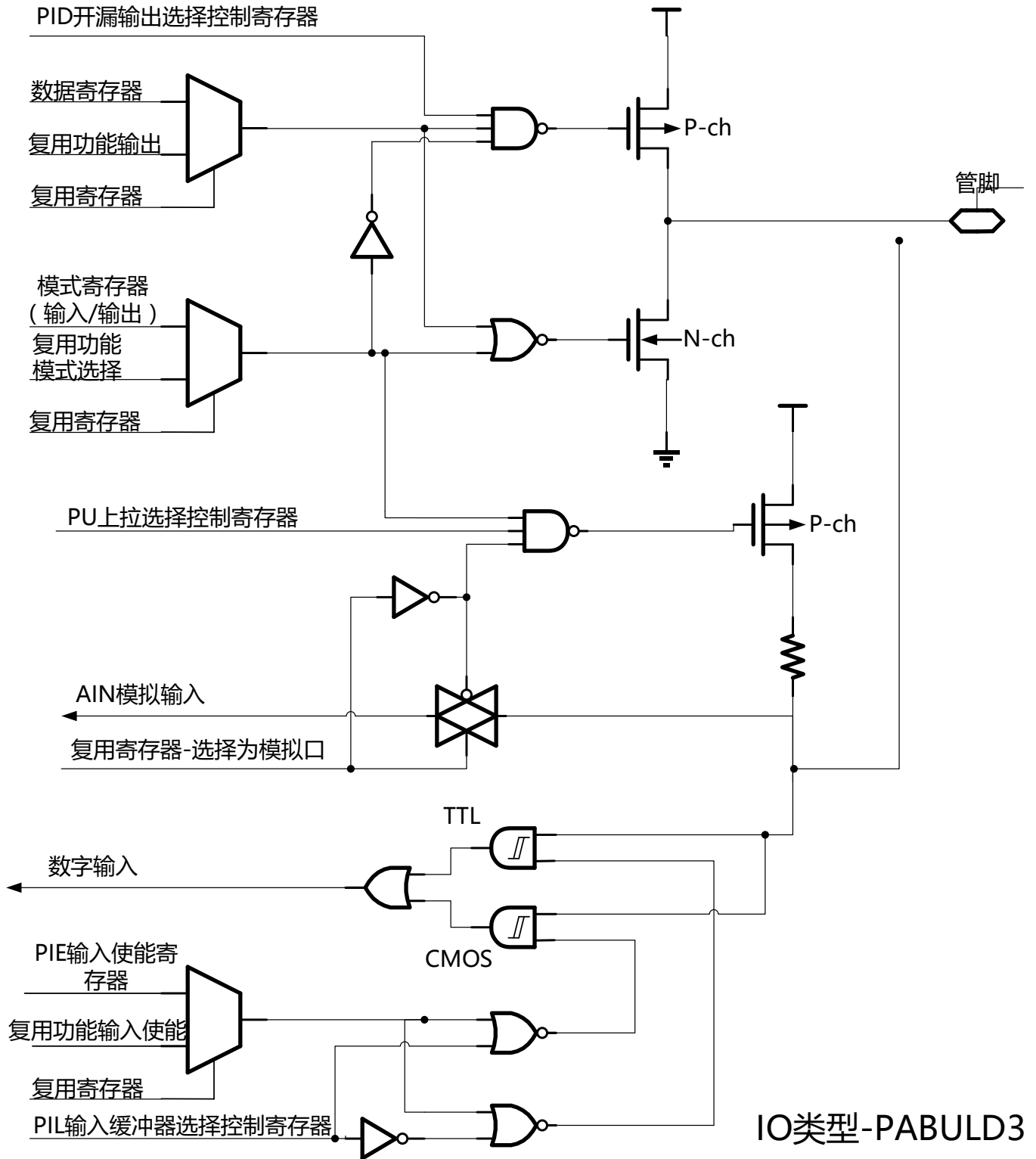


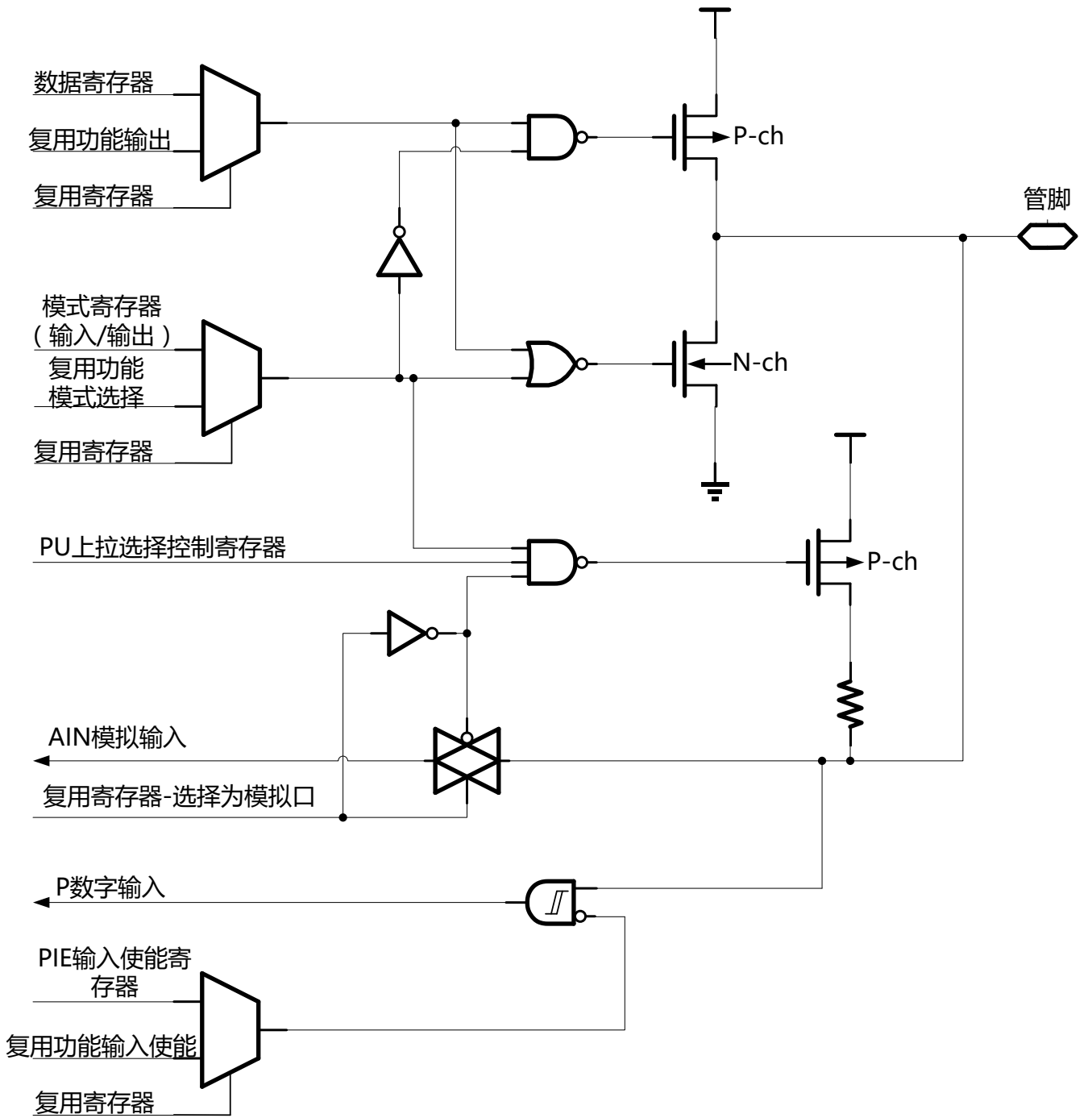
IO类型-PBUS6



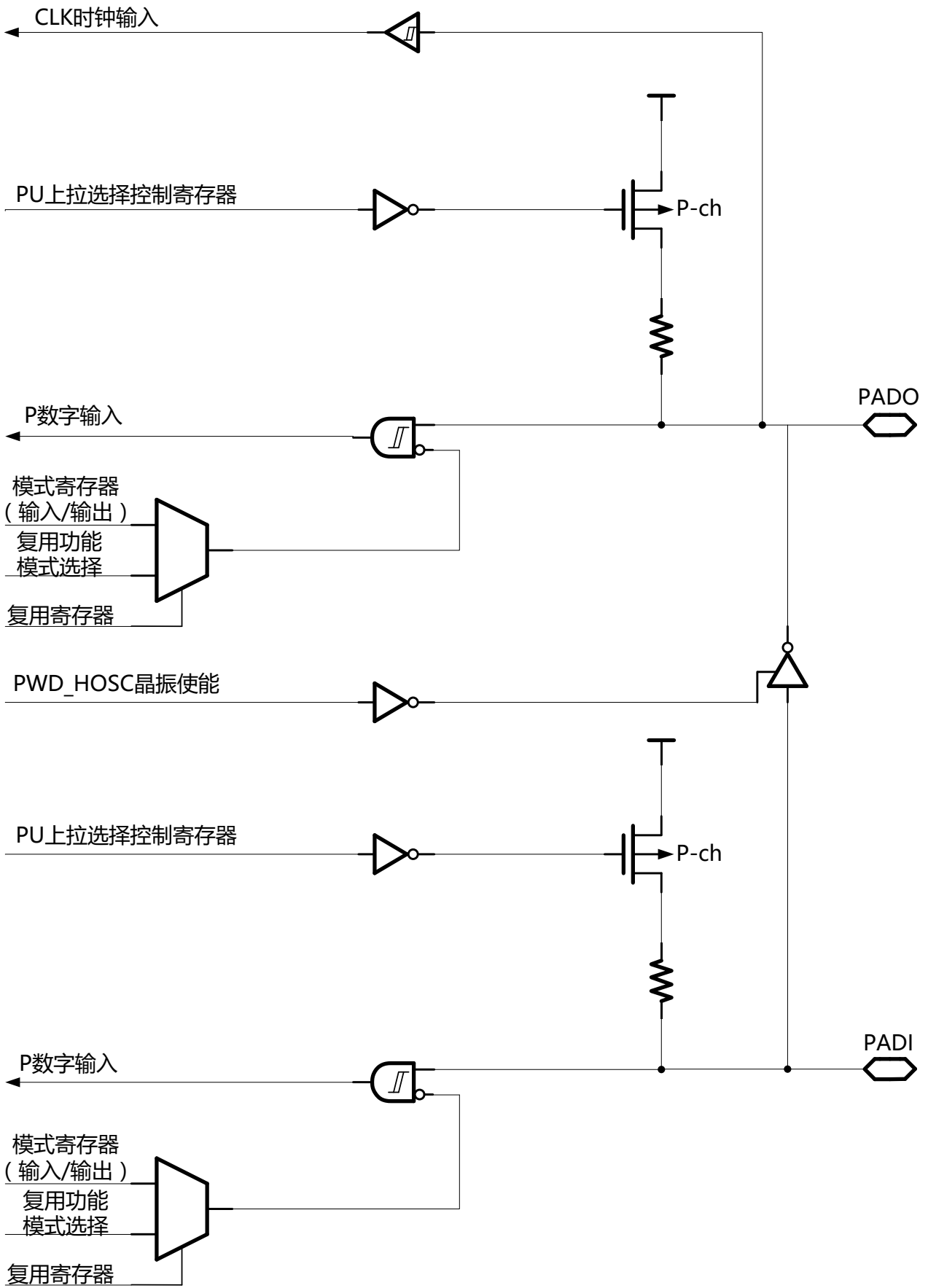


IO类型-PBULD3

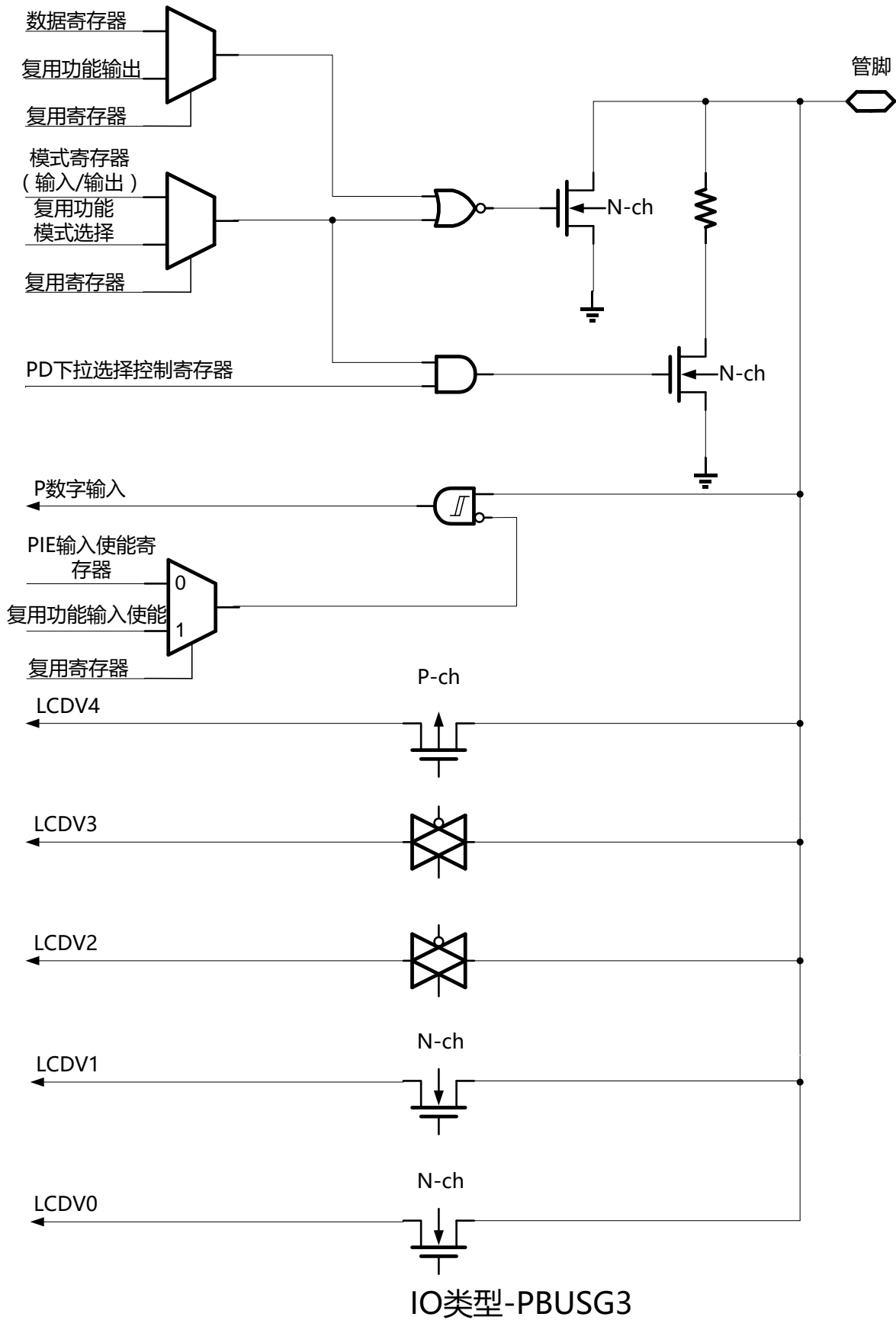




IO类型-PABUS3



### IO类型-PUXI



## 2 电气特性

电源(温度范围: -40°C~+85°C)						
主电源	VCC	2.4	5/3.3	5.5	V	建议主电源选择为 5V ±5% 或者 3.3V ±5%
CPU 最低工作电压	Vil	2.2	2.3	2.4	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
数字电流	DlDd		2.2		mA	CPU运行在3.6864MHz (PLL)
数字电流	DlDd		3.7		mA	CPU运行在7.3728MHz (PLL)
数字电流	DlDd		6.7		mA	CPU运行在 14.7456MHz (PLL)
VCC 休眠功耗	SIdd		5.5		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	uA	典型条件: Vbat=3.6V; Tc=25 度; 极限条件: Vbat=3.6V; Tc=85 度;
瞬间温补功耗	TPSIdd		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, 全温度范围 ±10%
极限参数(温度范围: -40°C~+85°C)						
主电电压	Vvcc	-0.3	--	+7	V	
电池输入电压	Vvbat	-0.3	--	+7	V	
DV <sub>DD</sub> to DGND		-0.3	--	+7	V	
DV <sub>DD</sub> to AV <sub>DD</sub>		-0.3		+0.3	V	
数字IO输出高电平	VOH		--	VCC+0.3	V	
数字IO输出低电平	VOL	-0.3	--		V	
数字IO输入高电平	VIH		0.7VCC			CMOS
数字IO输入低电平	VIL		0.3VCC			CMOS
数字IO输入高电平	VIH		0.4VCC			TTL

数字IO输入低电平	VIL		0.2VCC			TTL
数字IO的Isource	Isource	5		10	mA	6mA类型
数字IO的Isink	Isink	7		15	mA	6mA类型
数字IO的Isource	Isource	3		5	mA	3mA类型
数字IO的Isink	Isink	5		10	mA	3mA类型
模拟输入电压相对于AGND	V <sub>INA</sub>	-0.3	--	A <sub>VDD</sub> +0.3	V	
工作温度范围	T <sub>A</sub>	-40	--	85	°C	
存储温度范围	T <sub>stg</sub>	-65	--	150	°C	
<b>基准电压</b> (VCC=3V~5.5V, 温度范围: -40°C~+85°C)						
		min.	typ.	max.	unit	Notes
输出电压	V <sub>ref</sub>	1.25	1.26	1.27	V	
温度系数	T <sub>c</sub>		5	15	ppm/°C	
<b>模拟外设(温度范围: -40°C~+85°C)</b>						
低功耗比较器CMP1/CMP2/LVDIN 阈值	CMP	1.23	1.28	1.33	V	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200mv。
SAR ADC 输入范围	SAR-IN	0		BGR	V	BGR 为内部基准电压, 典型值为 1.25V
LCD 输出电压	LCDV <sub>D</sub>	4.85	5	5.15	V	5V 电荷泵, 全温度范围测试
LCD 输出电压	LCDV <sub>D</sub>	3.135	3.3	3.465		3.3V 电阻串分压, 全温度范围
VBAT 测量	VBAT <sub>D</sub>	0	3.6	5	V	SAR ADC 对 VBAT 引脚电压的测量范围;
每次温补时间	T <sub>tps</sub>		2		ms	
<b>时钟参数(温度范围: -40°C~+85°C)</b>						
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	3.6864	7.3728	29.4912	Mhz	
内部PLL时钟频率范围	PLL		7.3728	14.7456	MHz	
内部高频RCH	RCH	2.8	3.2	3.6	MHz	用于芯片复位后默认时钟
内部低频RCL	RCL	20	32.768	40	KHz	用于WDT时钟

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	LatchU P	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	/



### 3 系统控制

#### 3.1 电源域划分

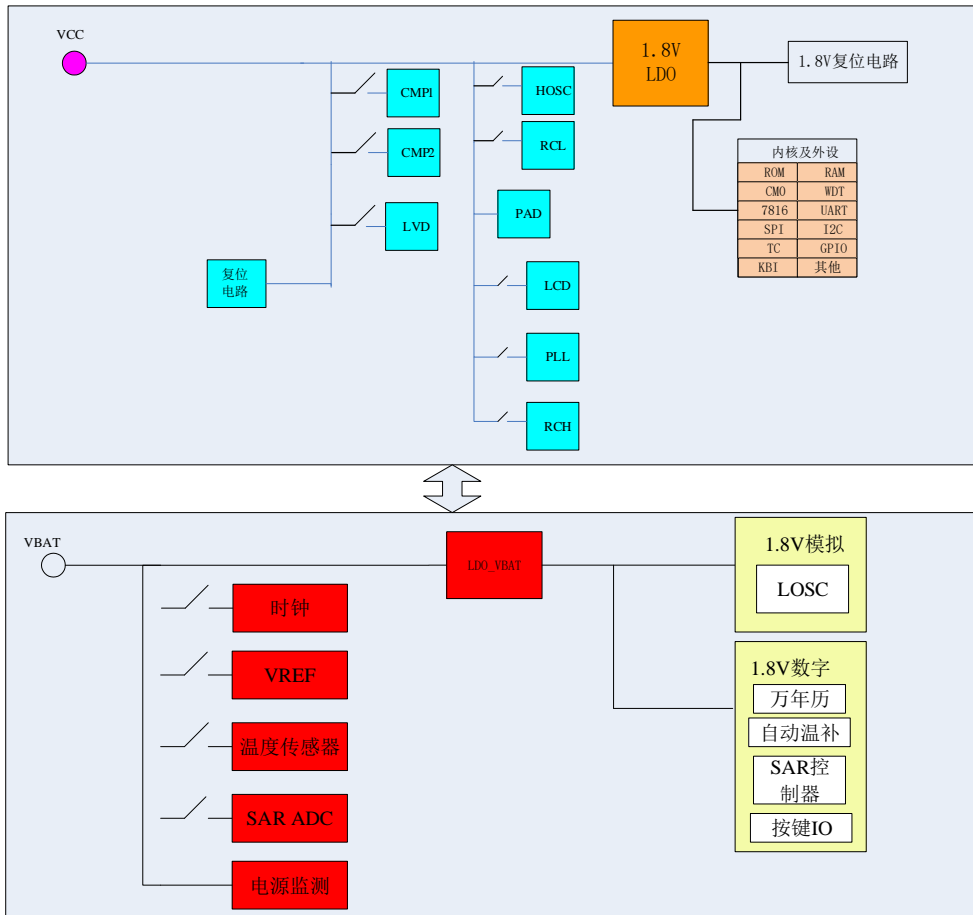


图 3.1 内部电源管理

VCC 与 VBAT 独立供电，RTC 相关电路（32768 晶振/万年历/自动温补电路等）、SAR-ADC 测量相关电路、P44/P45/P56 由 VBAT 供电，CPU 系统及其他外设由 VCC 供电。

实际应用中，可将 VCC 或 VBAT 连在一起供电，也可独立供电，VCC 正常工作的典型范围是 2.3V~5.5V，VBAT 正常工作的典型范围是 2V~5.5V。

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

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

CPU 可读取系统复位寄存器 SYS\_RST(0x14)的 bit9，可了解 VBAT 域电源是否正常以及是否发生过掉电。

### 3.2 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，

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

- 内部时钟源有三个：

RCH:内部高频 RC 时钟（典型值为 3.2MHz），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。

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

PLL: 内部 PLL 时钟，将 32.768KHz 倍频到 7.3728MHz 或者 14.7456Mhz。

- 低频运行模式可以选择为 LOSC 或者 RCL

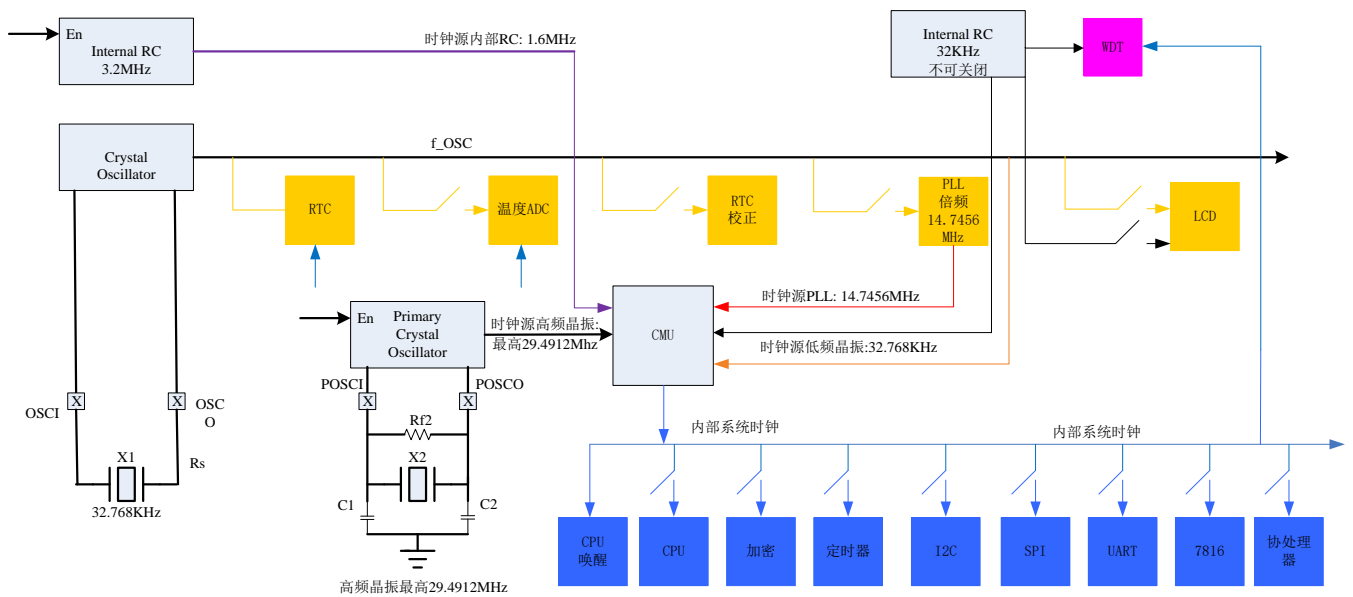
- LCD 时钟可选择为 LOSC 或者 RCL

以上四个时钟源均可用于 CPU 主系统时钟。

其中运行模式时系统时钟源可以选择为 PLL 也可以选择为高频晶振。

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

用户进行时钟切换，必须调用锐能微库函数。用户不应在应用程序中对 OSC\_CTL1(0x0)、SYS\_MODE(0x4) 寄存器进行写操作。如果对 OSC\_CTL2(0x10)进行写操作，应保证只改变需要操作的 bit 位，不要改变其他寄存器位的值。



### 3.3 时钟切换

包含有如下切换：

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

### 3.4 MCU 的低功耗模式

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

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

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

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

1. CPU 可通过调用库函数方式在高频时钟模式 HCM、低频时钟模式 LCM 和系统默认模式 RCH 之间任意切换；
2. CPU、外设的时钟均可关断；
3. 在最低功耗模式（CPU 休眠、SRAM 及数字外设不掉电、RTC 运行）功耗约为 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 数字域（来自 VCC 供电）	
M0 内核	开启，可关时钟，永不掉电
中断系统	开启，可关时钟，永不掉电
SRAM	开启，可关时钟，永不掉电
ROM	开启，可关时钟，永不掉电
FLASH	开启，CPU 休眠后自动休眠
WDT	开启，不可关断
其他外设	关闭，可关时钟，永不掉电
VCC 域	
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
LCD	关闭，可关电源
LVD	关闭，可关电源
比较器 CMP1	关闭，可关电源
比较器 CMP2	开启，可关电源
复位电路	一直开启
PLL	关闭，可关电源
HOSC	关闭，可关电源

VBAT 域	
RTC	开启, 万年历不可关断
温度测量 ADC	定时开启
LOSC	一直开启

默认关闭的模块在三种时钟下都可以选择开启或者关闭。

## 3.5 复位

### 3.5.1 外部 PIN 复位

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

### 3.5.2 上下电复位

内置三个上电复位 (POR) 电路和两个掉电复位 (BOR) 电路, 分别对 VCC、LDO18 (1.8V LDO) 及 VBAT 进行监测, 其中 VBAT 内只有 POR 电路。

该电路始终处于工作状态, 不可被关闭, 保证系统在超过阈值 (约 2.6V) 时可以正常工作; 在低于阈值 (约 2.35V) 时, 器件处于复位状态。不必再使用外部复位电路。

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

### 3.5.3 软件复位

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

### 3.5.4 看门狗复位

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

## 3.6 寄存器描述

系统控制模块的基址:

模块名	物理地址	映射地址
SYSC	0x40034000	0x40034000
寄存器名	地址偏移量	描述
OSC_CTL1	0x0	系统 OSC 控制寄存器 1
SYS_MODE	0x4	系统模式切换寄存器
SYS_PD	0x8	系统掉电控制寄存器
ADC_CTL	0xC	ADC 控制寄存器
OSC_CTL2	0x10	系统 OSC 控制寄存器 2
SYS_RST	0x14	系统复位寄存器
MAP_CTL	0x18	地址映射控制寄存器
MOD0_EN	0x1C	模块使能 0 寄存器
MOD1_EN	0x20	模块使能 1 寄存器
INTC_EN	0x24	INTC 使能寄存器
KBI_EN	0x28	KBI 使能寄存器

CHIP_ID	0x2C	器件 ID 寄存器
SYS_PS	0x30	系统控制寄存器密码保护位
IRFR_CTL	0x34	RCH 模式下红外时钟分频系数
TRIM_CFG1	0x78	时钟 TRIM 配置寄存器
TRIM_START	0x7C	时钟 TRIM 结果寄存器

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

### 3.6.1 系统 OSC 控制寄存器 1OSC\_CTL1(0x00)

偏移地址 0x00

比特位	名称	描述	读 / 写标志	复位值
31:16	---	预留	R	0
15:11	CLOCK_FLAG	时钟开启标志位: 如果时钟是开启了, 则此位为 1: { HOSC,RCL,RCH,PLL, LOSC }	R	01101
10:8	SYSCLK_STAT	系统主时钟频率指示: 000: 当前系统主时钟为 7.3728MHz; 001: 当前系统主时钟为 3.6864MHz; 010: 当前系统主时钟为 1.8432MHz; 011: 当前系统主时钟为 32.768KHz; 100: 当前系统主时钟为 14.7456Mhz; 101: 当前系统主时钟为 29.4912Mhz; (只支持外部晶振) 其他: 保留	R	010
7	PLL_LOCK	PLL 锁定状态 0: 未锁定 1: 锁定	R	0
6	PLL_HOSC_ON	系统运行在外部高频或内部 PLL 时钟时, 该位为 1; 系统运行在其他时钟时, 该位为 0。	R	0
5	IRCH_ON	系统运行在内部高频时钟时, 该位为 1; 系统运行在其他时钟时, 该位为 0。	R	1
4	LOSC_ON	系统运行在外部低频时钟时, 该位为 1; 系统运行在其他时钟时, 该位为 0。	R	0
3:2	PLL_HOSC_DIV	系统主时钟分频选择: (只对高频时钟模式有效) 00: 以 PLL、HOSC 作为 CPU 主时钟; 01: 以 PLL、HOSC 的二分频作为 CPU 主时钟; 10: 以 PLL、HOSC 的四分频作为 CPU 主时钟; 11: 以 HOSC (时钟频率选择为 14MHz 和 29MHz 时) 的八分频作为 CPU 主时钟, 若 HOSC 为 7.3728MHz 或者选择为 PLL 则表	R/W	01

		示为四分频； 备注：只能在 RC 或者 LC 模式才能更改。 备注：上述寄存器只决定分频系数，具体的系统主频需要根据分频系数和当前时钟源选择来确定。		
1	IRCH_PD	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

用户进行时钟切换，建议调用锐能微库函数。不建议用户在应用程序中对 OSC\_CTL1(0x0)寄存器进行写操作。

### 3.6.2 系统模式设置寄存器 SYS\_MODE(0x04)

偏移地址 0x04

比特位	名称	描述	读/写标志	复位值
31:6	---	预留	R	0
5	FLASH_BUSY	Flash busy 状态，不能进入模式切换： 0: idle 1: busy	R	0
4	EEPROM_BUSY	EEPROM busy 状态，不能进入模式切换： 0: idle 1: busy 此寄存器仅对 RN8612 有效。	R	0
3:0	MODE	写入 D，设置进入高频模式 HCM，bit2 读为 1； 写入 E，设置进入 RC 模式 RCM，bit1 读为 1； 写入 F，设置进入 32.768KHz 模式 LCM，bit0 读为 1。 即该寄存器读出值为：{0,HCM,RCM,LCM}	R/W	2

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

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

### 3.6.3 系统掉电控制寄存器 SYS\_PD (0x08)

偏移地址 0x08

比特位	名称	描述	读/写标志	复位值
31:12	---	预留	R	0
15	Reserved	预留	R/W	0

14	Reserved	预留	R/W	0
13	Reserved	预留	R/W	0
12	Reserved	预留	R/W	0
11	hysen_cmp2	cmplp2 的内部迟滞比较器迟滞开关 0x1: 打开迟滞 0x0: 关闭迟滞	R/W	0
10	hysen_cmp1	cmplp1 的内部迟滞比较器迟滞开关 0x1: 打开迟滞 0x0: 关闭迟滞	R/W	0
9	PWD_CMP2R	CMP2 内部 600K 电阻采样开关 0: CMP2 内部电阻采样开启, 外围电路需要注意内部 600K 对地电阻的影响, 比较器 $V_{il}$ 为 1.25V, 迟滞为 0.2V, 此时不要将 bit11 配置为 1; 1: CMP2 内部电阻采样关闭, 比较器阈值为 0.9V, 默认无迟滞, 可将 bit11 置 1 使之有 0.1V 的迟滞;	R/W	0
8	PWD_CMP1R	CMP1 内部 600K 电阻采样开关 0: CMP2 内部电阻采样开启, 外围电路需要注意内部 600K 对地电阻的影响, 比较器 $V_{il}$ 为 1.25V, 迟滞为 0.2V, 此时不要将 bit10 配置为 1; 1: CMP2 内部电阻采样关闭, 比较器阈值为 0.9V, 默认无迟滞, 可将 bit10 置 1 使之有 0.1V 的迟滞;	R/W	0
7	Reserved	预留	R/W	0
6	Reserved	预留	R/W	0
5	CMP2_PD	比较器 2 电源开关 0: 上电 1: 掉电 缺省开启。 如果要使用该功能, 还需要对 GPIO 复用寄存器进行配置。	R/W	0
4	CMP1_PD	比较器 1 电源开关 0: 上电 1: 掉电 如果要使用该功能, 还需要对 GPIO 复用寄存器进行配置。	R/W	1
3	LVD_PD	LVD 电源开关 0: 上电 1: 掉电 如果要使用该功能, 还需要对 GPIO 复用寄存器进行配置。	R/W	1
2:0	Reserved	预留	R	0

### 3.6.4 系统 OSC 控制寄存器 2OSC\_CTL2(0x10)

偏移地址 0x10

比特位	名称	描述	读 / 写标志	复位值
31:15	保留	保留		
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	预留	R	0
13	RCL_LCD	=0: LCD 选择 LOSC 外部低频晶振作为时钟源; =1: LCD 选择 RCL 内部低频晶振作为时钟源;	R/W	0
12	RCL_STB	=0: 低频运行模式 LCM 选择 LOSC 外部低频晶振作为时钟源; =1: 低频运行模式 LCM 选择 RCL 内部低频晶振作为时钟源; 该寄存器只能在 RC 模式或高频模式修改。	R/W	0
11:10	Reserved	可写, 内部测试寄存器, 用户不要改变此寄存器默认值。	R/W	00
9	Reserved	预留	R	0
8	Reserved	预留	R	0
7:5	PLL_FREQ	PLL 频点固定为 14.7456Mhz, 通过数字分频实现频率选择: 000: 运行频率选择为 7.3728MHz; 001: 运行频率选择为 14.7456Mhz; 其他: 保留 客户应调用库函数选择芯片运行频率, 不要在应用程序中改变此位的值。	R/W	000
4	PLL_HOSC_SEL	全速运行时系统主时钟选择: 0: 选择 PLL 输出作为系统主时钟; 1: 选择备用高频晶体作为系统主时钟。 该配置项只能在 RC 模式和低频模式下配置。	R/W	0
3	HOSC_PD	外部高频振荡器使能位: 0: 打开 1: 关闭	R/W	1
2:0	HOSC_FREQ	000: 外接高频晶振7.3728MHz 001: 外接高频晶振14.7456MHz	R/W	000



		010: 保留 011: 外接高频晶振29.4912MHz		
--	--	----------------------------------	--	--

系统时钟配置真值表:

	PLL_HOSC_DIV=00 不分频	PLL_HOSC_DIV=01 二分频	PLL_HOSC_DIV=10 四分频	PLL_HOSC_DIV=11 八分频
PLL_FREQ=000 PLL_HOSC_SEL=0	7.3728Mhz	3.6864MHz	1.8432MHz	不支持八分频, 如配置则为 1.8432Mhz
PLL_FREQ=001 PLL_HOSC_SEL=0	14.7456Mhz	7.3728MHz	3.6864MHz	1.8432Mhz
PLL_FREQ=000 PLL_HOSC_SEL=1 HOSC_FREQ =000	7.3728Mhz	3.6864MHz	1.8432MHz	不支持八分频, 如配置则为 1.8432Mhz
PLL_FREQ=000 PLL_HOSC_SEL=1 HOSC_FREQ =001	14.7456Mhz	7.3728Mhz	3.6864Mhz	1.8432Mhz
PLL_FREQ=000 PLL_HOSC_SEL=1 HOSC_FREQ =011	29.4912Mhz	14.7456Mhz	7.3728Mhz	3.6864Mhz

### 3.6.5 系统复位寄存器 SYS\_RST(0x14)

偏移地址 0x14

(此寄存器的 10 到 5 位只有上电复位可以复位这些位)

比特位	名称	描述	读/写标志	复位值
31:11	Reserved	预留	R	0
10	boi_vbat_ie	VBAT 域掉电中断使能位: =0 关闭中断 =1 使能中断	R/W	0
9	boi_vbat_flag	=0, VBAT 域电源正常 =1, VBAT 域曾发生过掉电 写 1 清零	R/W	0
8	MCU_RST	CPU 复位标志 (发生过软件复位或者 LOCK UP 复位): =1 表示发生过该复位, =0 表示没有发生。 写 1 清零	R/W	0
7	WDT_RST	WDT 复位标志: =1 表示发生过该复位, =0 表示没有发生。 写 1 清零	R/W	0
6	PIN_RST	外部管脚复位表示: =1 表示发生过该复位, =0 表示没有发生。 写 1 清零	R/W	0
5	POWEROK_RST	电源复位标志	R/W	1

		=1 表示发生过该复位，=0 表示没有发生。 写 1 清零		
4:3	Reserved	预留	R	0
2	LOCKUP_ENRST	LOCKUP 使能复位(CPU 发生了两次 Hard Fault 会引起 LOCKUP, 如果使能该位, 可引起系统复位): 0: LOCKUP 不引起系统复位 1: LOCKUP 引起系统复位	R/W	0
1	Reserved	预留	R	0
0	Reserved	预留	R	0

### 3.6.6 系统映射控制寄存器 SYS\_MAPCTL(0x18)

偏移地址 0x18

比特位	名称	描述	读/写标志	复位值
31:3	Reserved	预留	R	0
2:0	REMAP	地址映射: 000: FLASH 映射在 0 地址 (正常模式) 001: FLASH 映射在 1/2 容量地址 (仅 512KB 容量支持该功能)或 FLASH 与 EEPROM 映射地址互换 (仅 RN8612 支持) 010: FLASH 与 SRAM 映射地址互换 011: BOOTROM 映射在 0 地址 100: FLASH 映射在 1/2 容量地址 (仅 512KB 容量支持该功能) 其他: 保留.	R/W	000

### 3.6.7 模块使能 0 寄存器 MOD0\_EN (0x1c)

偏移地址 0x1C

比特位	名称	描述	读/写标志	复位值
31:16	---	预留	R	0
15	SPI_EN	SPI 模块使能清零, 时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
14	I2C_EN	I2C 模块使能清零, 时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
13	ISO7816_EN	ISO7816 模块使能清零, 时钟门控, cm0 进入 deepsleep 同步关闭此时钟:	R/W	0

		0: 时钟停止, 模块清零 1: 时钟启动, 模块使能		
12	UART38K_EN	UART38K 红外调制时钟开启使能, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止 1: 时钟启动	R/W	0
11	UART3_EN	UART3 模块使能清零, 时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
10	UART2_EN	UART2 模块使能清零, 时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
9	UART1_EN	UART1 模块使能清零, 时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
8	UART0_EN	UART0 模块使能清零, 时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
7	UART5_EN	UART5 模块使能清零, 时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
6	UART4_EN	UART4 模块使能清零, 时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
5	TC1_EN	TC1 模块使能清零, 时钟门控: 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	CPC_EN	协处理器模块使能清零, 时钟门控, cm0 进入 deepsleep 同步关闭此时钟: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
2	Reserved	预留, 除 RN8612 外其他型号	R	0
	EEPROM_EN	仅 RN8612 支持: EEPROM 模块使能清零, 时钟门控, cm0 进	R/W	1

		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		
1	Reserved	预留	R	0
0	Reserved	预留	R	0

### 3.6.8 模块使能 1 寄存器 MOD1\_EN(0x20)

偏移地址 0x20

比特位	名称	描述	读/写标志	复位值
31:12	Reserved	预留	R	0
11	CMPLVD_EN	CMPLVD 模块使能清零，apb 时钟门控： 0: 时钟停止 1: 时钟启动	R/W	0
10	RTC_SAR_EN	RTC/SAR apb 时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0: 时钟停止 1: 时钟启动	R/W	1
9	WDT_EN	WDT apb 时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0: 时钟停止 1: 时钟启动	R/W	1
8	Reserved	预留	R	0
7	Reserved	预留	R	0
6	LCD_EN	LCD 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
5	GPIO_EN	GPIO 模块使能清零，时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
4	---	保留	R	0
3	---	保留	R	0
2	---	保留	R	0
1	---	保留	R	0
0	---	保留	R	0

### 3.6.9 INTC 使能寄存器 INTC\_EN(0x24)

偏移地址 0x24

比特位	名称	描述	读/写标志	复位值
31:9	Reserved	预留	R	0
8	INTC_EN	INTC apb 模块时钟门控，cm0 进入 deepsleep	R/W	0

		同步关闭此时钟： 0: 时钟停止 1: 时钟启动		
7	INTC7_EN	INTC7 模块使能清零，时钟门控： 0: 时钟停止 1: 时钟启动	R/W	0
6	INTC6_EN	INTC6 模块使能清零，时钟门控： 0: 时钟停止 1: 时钟启动	R/W	0
5	INTC5_EN	INTC5 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
4	INTC4_EN	INTC4 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
3	INTC3_EN	INTC 3 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
2	INTC2_EN	INTC 2 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
1	INTC1_EN	INTC 1 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0
0	INTC0_EN	INTC0 模块使能清零，时钟门控： 0: 时钟停止，模块清零 1: 时钟启动，模块使能	R/W	0

### 3.6.10 KBI 使能寄存器 KBI\_EN(0x28)

偏移地址 0x28

比特位	名称	描述	读/写标志	复位值
31:9	Reserved	预留	R	0
8	KBI_EN	KBI apb 模块时钟门控，cm0 进入 deepsleep 同步关闭此时钟： 0: 时钟停止 1: 时钟启动	R/W	0
7	KBI7_EN	KBI7 模块使能清零，时钟门控： 0: 时钟停止 1: 时钟启动	R/W	0
6	KBI6_EN	KBI6 模块使能清零，时钟门控： 0: 时钟停止 1: 时钟启动	R/W	0

5	KBI5_EN	KBI5 模块使能清零, 时钟门控: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
4	KBI4_EN	KBI4 模块使能清零, 时钟门控: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
3	KBI3_EN	KBI3 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	KBI2 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	KBI1 模块使能清零, 时钟门控: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0
0	KBI0_EN	KBI0 模块使能清零, 时钟门控: 0: 时钟停止, 模块清零 1: 时钟启动, 模块使能	R/W	0

### 3.6.11 器件 ID 寄存器 CHIP\_ID(0x2C)

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

### 3.6.12 系统控制密码寄存器 SYS\_PS(0x30)

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

### 3.6.13 红外配置寄存器 IRFR\_CTL (0x34)

比特位	名称	描述	读/写标志	复位值
31:6	---	预留	R	0
5:0	IRFR_CYCLE	该寄存器的密码是 0x82 RCH 模式下, 红外时钟分频系数, 基于 RCH 为 3.6864MHZ。	R/W	0x18

		... 0x19:红外输出时钟 36.9K; 0x18:红外输出时钟 38.4K; 0x17:红外输出时钟 40K; ... 计算公式为: IRFR 值=RCH 实测频率/4/38K		
--	--	---	--	--

**3.6.14 时钟校正配置寄存器 TRIM\_CFG1(0x78)**

比特位	名称	描述	读 / 写标志	复位值
31:29	---	预留	R	0
28	CAL_OV_IE	被校准时钟计数器溢出标志中断使能: 0: 不使能中断; 1: 使能中断;	R/W	0
27	CAL_DONE_IE	时钟校准完成标志中断使能: 0: 不使能中断; 1: 使能中断;	R/W	0
26	CAL_CLK_SEL	被校正时钟源选择 0: 被校正时钟源选择 RCH; 1: 被校正时钟源选择 RCL;	R/W	0
25: 24	REF_CLK_SEL	参考时钟源选择 00: 参考时钟源选择 LOSC; 01: 参考时钟源选择 HOSC; 10: 参考时钟源选择 RCH; 11: 低功耗模式;	R/W	11
23:20	---	预留	R	0
19:0	REF_CLK_CNT [19:0]	参考时钟计数值	R/W	0x10000

**3.6.15 时钟校正启动寄存器 TRIM\_START(0x7C)**

比特位	名称	描述	读 / 写标志	复位值
31:28	---	预留	R	0
27	STOP	时钟校准终止位: 0: 无操作; 1: 终止时钟校准; Note: 如果终止时钟校准, 需要将该位写 0 后才能重	R/W	0

		新开始时钟校准。		
26	START	时钟校准启动位： 0：无操作； 1：启动时钟校准； Note：时钟校准完成或被终止后，该位自动清零。	R/W	0
25	CAL_OV	被校正时钟计数器溢出标志： 0：没有溢出； 1：溢出； Note：写 1 清 0。	R/W	0
24	CAL_DONE	时钟校准完成标志： 0：未完成； 1：已完成； Note：写 1 清 0。	R/W	0
23:20	---	预留	R	0
19:0	CAL_CLK_CNT [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：  

$$(CAL\_CLK\_CNT/REF\_CLK\_CNT)*32768Hz$$

$$=(400000/4096)*32768Hz$$

$$=3200000Hz$$

$$=3.2MHz$$



## 4 处理器架构

### 4.1 概述

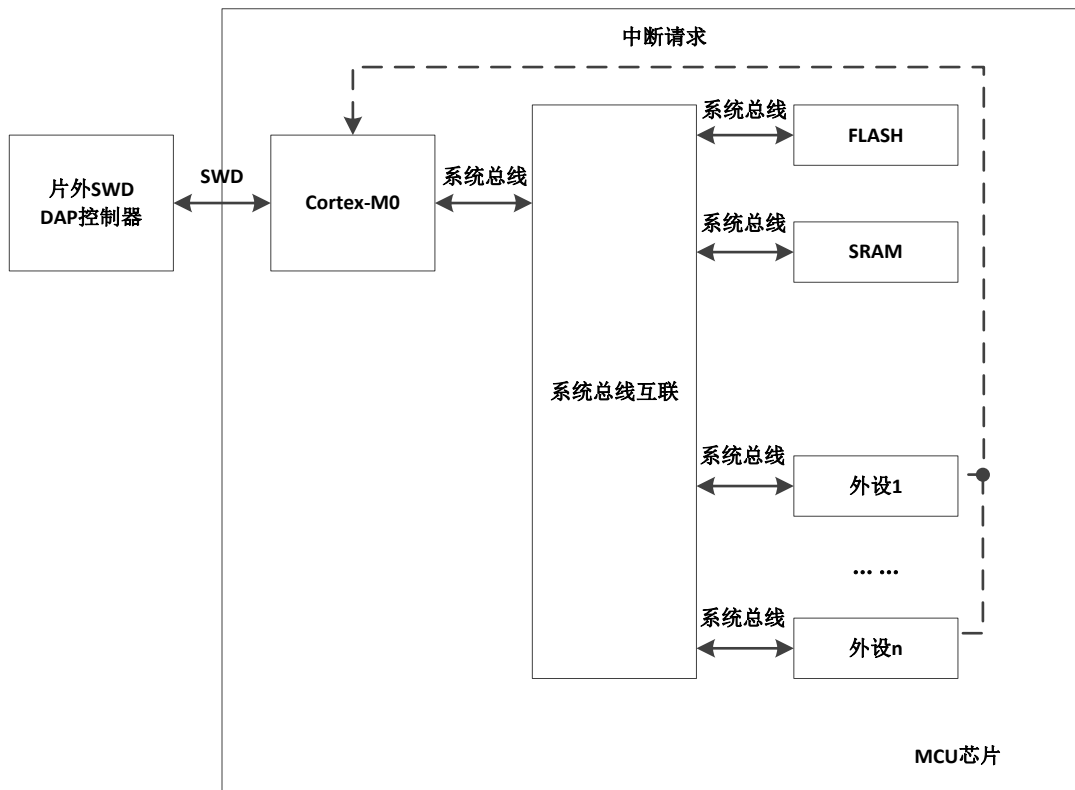
有二种方式（二个主设备）可以发起对 MCU 内置设备的访问：

- Cortex-M0:
  - 指令访问和数据访问；
  - 可访问所有的从设备；
- 外置的 SWD 控制器（如 JLINK 或类似功能的设备）:
  - 调试接口和资源访问；
  - 可访问所有的从设备；

MCU 内置的从设备资源包括存储器（FLASH、SRAM，EEPROM（RN8612））和各种外设（UART、定时器、看门狗等）。

部分外设可发起中断请求，如 UART、定时器等。

图 4-1 MCU 设备物理互联架构



### 4.2 Cortex-M0 处理器

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

- 简便易用的程序模型
- 高代码集成度，具有 32 位的性能
- 工具和二进制代码与 Cortex-M 处理器系列向上兼容，方便升级和扩展
- 集成了极低功耗的睡眠模式
- 高效的代码执行允许处理器时钟更低，或者延长睡眠模式的时间

- 单周期 32 位硬件乘法器
- 零抖动中断处理
- 中断定时确定，中断处理效率高
- 支持中断/异常嵌套和抢占
- 支持 24 位系统节拍计数器
- 提供 4 个中断优先级
- 支持 2 个观察点，4 个硬件断点
- 支持串行调试接口（SWD），实现处理器内部状态高度可视和可控
- CM0 内嵌系统定时器，具体请参考 ARM 文档。

有关 Cortex-M0 的详细资料可参阅 ARM 文档。

#### 4.2.1 中断配置

支持 32 个中断，其中开放有 8 个外部中断，分别为外部中断 0~7。

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

表 4-1 中断/异常向量表及其配置信息

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

23	7	UART1	0x5C	可配置	与处理器异步
24	8	UART2	0x60	可配置	与处理器异步
25	9	UART3	0x64	可配置	与处理器异步
26	10	SPI	0x68	可配置	与处理器异步
27	11	I2C	0x6C	可配置	与处理器异步
28	12	7816_0	0x70	可配置	与处理器异步
29	13	7816_1	0x74	可配置	与处理器异步
30	14	TC0	0x78	可配置	与处理器异步
31	15	TC1	0x7C	可配置	与处理器异步
32	16	UART4	0x80	可配置	与处理器异步
33	17	UART5	0x84	可配置	与处理器异步
34	18	看门狗	0x88	可配置	与处理器异步
35	19	KBI	0x8C	可配置	与处理器异步
36	20	LCD	0x90	可配置	与处理器异步
37	21	协处理器	0x94	可配置	与处理器异步
38	22	保留	0x98	-	
39	23	保留	0x9C	-	
40	24	外部中断 0	0xA0	可配置	与处理器异步
41	25	外部中断 1	0xA4	可配置	与处理器异步
42	26	外部中断 2	0xA8	可配置	与处理器异步
43	27	外部中断 3	0xAC	可配置	与处理器异步
44	28	外部中断 4	0xB0	可配置	与处理器异步
45	29	外部中断 5	0xB4	可配置	与处理器异步
46	30	外部中断 6	0xB8	可配置	与处理器异步

47	31	外部中断 7	0xBC	可配置	与处理器异步
----	----	--------	------	-----	--------

### 4.3 MCU 存储映射

MCU 的存储映射请参考“图 4-2 MCU 存储映射”。

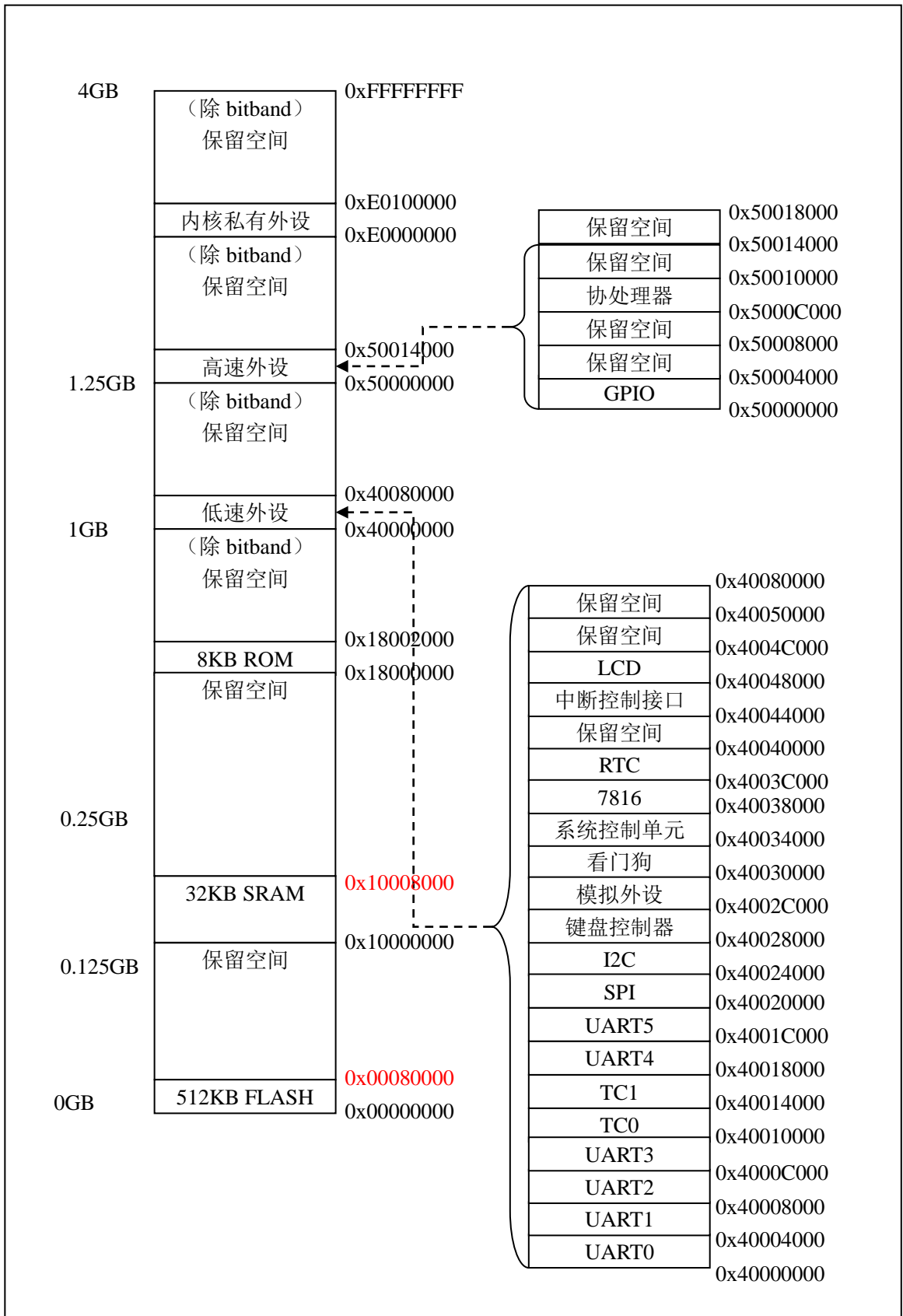
MCU 的高速外设包括：

- GPIO

MCU 的低速外设包括：

- 6 个 UART 接口
- 2 个 32 位定时器
- 1 个 SPI 接口
- 1 个 I2C 接口
- 1 个 KEY 控制器
- 1 个通用 ADC 接口
- 1 个看门狗单元
- 1 个系统控制单元
- 2 个 7816 接口
- 1 个 RTC 接口
- 一个中断控制接口
- 1 个 LCD 控制器

图 4-2 MCU 存储映射



备注：RN8612 支持 EEPROM，地址空间为 0x08000000~0x08008000

● 存储重映射

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

表 4-2 存储重映射配置

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

● 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 btband 地址：

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

4.3.1 SRAM

片内静态 RAM (SRAM)，容量为 32KB（其中高 2KB 做为 CACHE 使用）或 8KB（全部开放给客户使用），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 0x1000\_0300 ~ 0x1000\_03ef 共 240Bytes，一旦系统发生复位，cpu 会从 BOOTROM 中执行启动程序，此地址空间的数据会被占用，使用此地址空间请注意该特性。

注意：RN8318/RN8615/RN8613/RN8611 的 CACHE 默认是关闭的，需要使用下面命令将其打开：

```
*(volatile uint32_t*)(0x50014060) = 0xfeed0001;
```

或者调用锐能微库函数打开 CACHE。

4.3.2 FLASH

片内最大 512KB FLASH，支持：

- 最少 10 万擦写次数；
- 数据最少保存时间 10 年；
- RN8613 存储区包含 16 个块，每个块包含 64 个页，每页 256 字节；
- RN8612 存储区包含 16 个块，每个块包含 64 个页，每页 128 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)

### 4.3.3 EEPROM

片内 32KB EEPROM, 支持:

- 最少 100 万擦写次数;
- 数据最少保存时间 10 年;
- 任意地址进行随机访问, 访问数据宽度支持 8 位, 16 位或者 32 位
- 支持页擦除、块擦除、**字编程**, 具体的操作需要调用锐能微库函数 (nvm.a(IAR)/nvm.lib(KEIL))
- 共分成 4 个区, 每个区 64 个页, 每页 128 字节;

低功耗应用时, 客户需要调用锐能微库函数关闭或开启 **EEPROM**; 库函数 (nvm.a(IAR)/nvm.lib(KEIL))提供的 EEPROM 操作函数接口如下:

uint8_t eepromPageErase(uint32_t pg)
uint8_t eepromSectorErase(uint32_t sec)
uint8_t eepromProgram(uint32_t dst_addr, uint32_t src_addr, uint32_t len)
void eepromStandby(void)
void eepromWakeup(void)

详细的操作请见锐能微应用笔记。

## 4.4 中断应用

在头文件中加入 SOC 头文件 #include <RN821x.h>, 即可使用 SOC 各中断, RN821x.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 RN821x.h 的文件中找到中断号并开启中断, 例如 KBI 的中断号为 KBI\_IRQn, 开启 KBI 中断既为 NVIC\_EnableIRQ(KBI\_IRQn), 如需设置中断的优先级可使用 void NVIC\_SetPriority(IRQn\_t IRQn, uint32\_t priority)。
- 4、编写中断服务函数, 对不同的中断, 函数名已固定, 可在 startup\_RN821x.s 的向量表中查找, 如 KBI 中断服务程序函数名为 KBI\_HANDLER, 中断服务函数可写为:

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

- 5、关闭中断使能: void NVIC\_DisableIRQ0 (IRQn\_t IRQn)。





## 5 RTC

### 5.1 概述

BaseAddr 为：0x4003C000；

RTC 模块提供实时时钟、振荡器温度补偿、日历、闹钟、时钟脉冲输出等功能。

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

集成温度传感器，提供温度测量的数字结果。

该模块由 VBAT 独立供电。

### 5.2 特点

- 提供准确的温度值， $-25^{\circ}\text{C}\sim 70^{\circ}\text{C}$  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 0.00678ppm
- 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

### 5.3 寄存器描述

RTC 模块的基址

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

RTC 模块的寄存器偏移地址

寄存器名	地址偏移量	描述
RTC 寄存器组		
RTC_CTL	Offset+0x00	RTC 控制寄存器
RTC_SC	Offset+0x04	秒寄存器，写保护
RTC_MN	Offset+0x08	分钟寄存器，写保护
RTC_HR	Offset+0x0C	小时寄存器，写保护
RTC_DT	Offset+0x10	日寄存器，写保护
RTC_MO	Offset+0x14	月寄存器，写保护
RTC_YR	Offset+0x18	年寄存器，写保护

RTC_DW	Offset+0x1C	星期寄存器，写保护
RTC_CNT1	Offset+0x20	定时器 1 寄存器
RTC_CNT2	Offset+0x24	定时器 2 寄存器
RTC_SCA	Offset+0x28	秒闹钟寄存器
RTC_MNA	Offset+0x2C	分钟闹钟寄存器
RTC_HRA	Offset+0x30	小时闹钟寄存器
RTC_IE	Offset+0x34	RTC 中断使能寄存器
RTC_IF	Offset+0x38	RTC 状态寄存器
RTC_TEMP	Offset+0x3C	当前温度寄存器，可读可写，写保护
RTC_CALPS	Offset+0xCC	RTC 二次补偿寄存器写保护，写入 8'hA8，T0~T9 寄存器才会起作用。
RTC_CAL_T0	Offset+0xD0	T0~T9 是 8bit 寄存器，在硬件自动温补基础上对 RTC 的误差做二次补偿，刻度是 0.25ppm，二进制补码格式； 补偿温度范围：T < -30 度
RTC_CAL_T1	Offset+0xD4	补偿温度范围：-30 ≤ T < -20 度
RTC_CAL_T2	Offset+0xD8	补偿温度范围：-20 ≤ T < -10 度
RTC_CAL_T3	Offset+0xDC	补偿温度范围：-10 ≤ T < 0 度
RTC_CAL_T4	Offset+0xE0	补偿温度范围：0 ≤ T ≤ 10 度
RTC_CAL_T5	Offset+0xE4	补偿温度范围：35 < T ≤ 45 度
RTC_CAL_T6	Offset+0xE8	补偿温度范围：45 < T ≤ 55 度
RTC_CAL_T7	Offset+0xEC	补偿温度范围：55 < T ≤ 65 度
RTC_CAL_T8	Offset+0xF0	补偿温度范围：65 < T ≤ 75 度
RTC_CAL_T9	Offset+0xF4	补偿温度范围：75 < T ≤ 85 度
LOSC_CFG1	Offset+0x6C	LOSC 配置寄存器
VBAT_IOMODE	Offset+0x88	VBAT 域 IO 口配置寄存器
VBAT_IOWEN	Offset+0x8C	VBAT_IOMODE 写使能寄存器

### 5.3.1 RTC 控制寄存器 RTC\_CTL(0x00)

偏移地址 0x00

比特位	名称	描述	读/写标志	复位值
31:12	---	预留	R	0
11	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。	R	0

		因此建议应用程序在写万年历后统一等待 300us 后再读取。		
8	WRTC	<p>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：                      1) 按照“年月日时分秒”顺序写，当写入秒寄存器后时间开始从写入时刻起累计，注意此种方法存在秒写入之前分钟翻转的可能，所以写入后要读出做校验；                      2) 按照“秒分时分年月日”顺序写，先写秒寄存器，万年历计数器被清零，只要在一秒内写入其他值，可以保证成功写入。                      注意硬件对“年月日”有做合法性判断，不能按照“日月年”顺序写，只能连续写入“年月日”。</p>	R/W	0
7:6	TSE	<p>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：该寄存器仅有上电复位起作用。</p>	R/W	00
5:3	TCP	<p>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</p>	R/W	001
02:00	FOUT	<p>000：禁止输出                      001：1Hz 输出                      010：1/30Hz 输出                      011：32768Hz 输出                      100：16Hz 输出                      101：8Hz 输出                      110：4Hz 输出                      111：保留                      该寄存器只有上电复位起作用。</p>	R/W	000

5.3.2 秒寄存器 RTC\_SC(0x04)

偏移地址 0x04

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

31:07	Reserved	预留	R	0
06:00	SC	存储时钟的秒值 BCD 码格式， <b>SC[6:4]</b> 为秒值的十位， <b>SC[3:0]</b> 为秒值的个位，秒值的范围为 0~59	R/W	-

### 5.3.3 分钟寄存器 RTC\_MN(0x08)

偏移地址 0x08

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

### 5.3.4 小时寄存器 RTC\_HR(0x0c)

偏移地址 0x0C

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

### 5.3.5 日期寄存器 RTC\_DT(0x10)

偏移地址 0x10

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

### 5.3.6 月份寄存器 RTC\_MO(0x14)

偏移地址 0x14

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

### 5.3.7 年份寄存器 RTC\_YR(0x18)

偏移地址 0x18

比特位	名称	描述	读/写标志	复位值
31:08	Reserved	预留	R	0
07:00	YR	存储时钟的年份值 BCD 码格式， <b>YR[7:4]</b> 为年份值的十位， <b>YR[3:0]</b> 为年份值	R/W	-

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

### 5.3.8 星期寄存器 RTC\_DW(0x1c)

偏移地址 0x1C

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

### 5.3.9 RTC 定时寄存器 1 RTC\_CNT1(0x20)

偏移地址 0x20

比特位	名称	描述	读/写标志	复位值
31:09	Reserved	预留	R	0
08	CNT1PD	= 0 : 1 秒定时来自万年历秒中断。 = 1 : 关闭定时器 1，重新开启时计数器会重新开始计(刻度 1 S，与万年历秒中断无关)	R/W	0
07:00	CNT	定时器1计数器预设值 无符号数，计数单位为1s。当计数值=(CNT+1)时，置位 RTCCNT1F标志。（最小可以每1秒产生一次中断，最大可以每256 秒产生一次中断） 注：该定时器在RTC校正后是准确的。	R/W	0

### 5.3.10 RTC 定时寄存器 2 RTC\_CNT2(0x24)

偏移地址 0x24

比特位	名称	描述	读/写标志	复位值
31:10	Reserved	预留	R	0
09	CNT2_CLK_SEL	CNT2_CLK 的时钟来源选择 0: 32768Hz 1: RCL	R/W	0
08	CNT2PD	= 0 : 定时器来自内部固定的 1/256S 中断 = 1 : 关闭定时器 2，重新开启时计数器会重新开始计(刻度 1/256S)	R/W	0
07:00	CNT	定时器2计数器预设值 无符号数，计数单位为1/256s。当计数值=(CNT+1)时，置位 RTCCNT2F标志。（最小可以每1/256秒产生一次中断，最大可以每1秒产生一次中断） 注：该定时器源自32768Hz晶体，未经校正，有一定误差。	R/W	0

### 5.3.11 秒闹钟寄存器 RTC\_SCA(0x28)

偏移地址 0x28

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

31:07	Reserved	预留	R	0
06:00	SCA	秒闹钟值 BCD 码格式, <b>SCA[6:4]</b> 为秒值的十位, <b>SCA[3:0]</b> 为秒值的个位, 秒值的范围为 0~59	R/W	0

### 5.3.12 分钟闹钟寄存器 RTC\_MNA(0x2c)

偏移地址 0x2C

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

### 5.3.13 小时闹钟寄存器 RTC\_HRA(0x30)

偏移地址 0x30

比特位	名称	描述	读/写标志	复位值
31:06	Reserved	预留	R	0
05:00	HRA	小时闹钟值 BCD 码格式, <b>HRA[5:4]</b> 为小时值的十位, <b>HRA[3:0]</b> 为小时值的个位, 小时值的范围为 0~23。	R/W	0

注: 04~30H 寄存器没有复位值。

### 5.3.14 RTC 中断使能寄存器 RTC\_IE(0x34)

偏移地址 0x34

比特位	名称	描述	读/写标志	复位值
31:10	Reserved	预留	R	0
9	RTC_1S_SEL	秒中断来源选择: =0: 根据系统时钟模式: 高频模式下选择 pll_1hz, 非高频模式下选择 RTC 秒脉冲 =1: 一直选择 RTC 的秒脉冲	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; 当 RTC_IE[8:0]全部为低时, 中断模块时钟才关闭;	R/W	0
7	MOIE	月份中断使能 0: 不使能 1: 使能	R/W	0
6	DTIE	日期中断使能 0: 不使能 1: 使能	R/W	0
5	HRIE	小时中断使能 0: 不使能	R/W	0

		1: 使能		
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

### 5.3.15 RTC 中断标志寄存器 RTC\_IF(0x38)

偏移地址 0x38

比特位	名称	描述	读/写标志	复位值
31:12	Reserved	预留	R	0
7	MOF	月份中断标志位 0: 月份计数器未加 1 1: 月份计数器加 1 Note: 写 1 清零	R/W	0
6	DTF	日期中断标志位 0: 日期计数器未加 1 1: 日期计数器加 1 Note: 写 1 清零	R/W	0
5	HRF	小时中断标志位 0: 小时计数器未加 1 1: 小时计数器加 1 Note: 写 1 清零	R/W	0
4	MNF	分钟中断标志位 0: 分钟计数器未加 1 1: 分钟计数器加 1 Note: 写 1 清零	R/W	0
3	SCF	秒中断标志位 0: 秒计数器未加 1 1: 秒计数器加 1 Note: 写 1 清零	R/W	0
2	RTCCNT2F	RTC 定时器 2 中断标志位 0: 定时器 1 中断未发生	R/W	0

		1: 定时器 1 中断发生 Note: 写 1 清零		
1	RTCCNT1F	RTC 定时器 1 中断标志位 0: 定时器 1 中断未发生 1: 定时器 1 中断发生 Note: 写 1 清零	R/W	0
0	ALMF	闹钟事件标志位, 与实时时钟匹配的闹钟事件发生 0: 闹钟事件未发生 1: 闹钟事件发生 Note: 写 1 清零	R/W	0

### 5.3.16 当前温度寄存器 RTC\_TEMP(0x3c)

偏移地址 0x3C

比特位	名称	描述	读/写标志	复位值
31:10	Reserved	预留	R	0
09:00	TEMP	当前温度值。 Bit9 为符号位; Bit8~2 为整数位; Bit1~0 为小数位。 //Temp[9] Temp[8:2] Temp[1] Temp[0] // 符号 -128 度~127 度 0.5 度 0.25 度 表示范围: -128 度~+127.75 度 TSE=00: 禁止自动温补。该寄存器无效, 读出的值无意义。 TSE=01: 按照 TCP 设置周期自动温补。该寄存器显示的是该测量周期的温度值。 TSE=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; TSE=11: 启动用户温补模式 1, 温度寄存器不可更改, 每次写温度寄存器就启动一次温度补偿操作, 温度寄存器的值由 SOC 测量得到。	R/W	-

RTC 自动温补需要定义以下寄存器, 这些寄存器值在客户量产环节获得。

1. 初始频率偏差寄存器 RTC\_DOTA0: 修正晶体的初始频率偏差; (每台表需要获得, 锐能微提供的库函数可以完成该寄存器的操作)
2. 二次曲线顶点温度寄存器 RTC\_XT0 (获得晶体批次参数, 配置选项字节, 通过编程界面写入)
3. 晶振温度系数寄存器 RTC\_ALPHA (获得晶体批次参数, 配置选项字节, 通过编程界面写入)

### 5.3.17 LOSC 配置寄存器 LOSC\_CFG1(0x6c)

比特位	名称	描述	读/写标志	复位值
31:11	---	预留	R	0
10	VCC_BOI_FLAG	=0: 正常; =1, VCC 电源域曾发生过掉电; 写 1 清零	R/W	0



9	LOSC_FAIL	=0:正常; =1: LOSC 曾发生过停振 写 1 清零。	R	0
8	LOSC_WEN	=0: LOSC_PD 不可写 =1: LOSC_PD 可写	R/W	0
7:0	LOSC_PD	LOSC 使能位: =其他: 打开; =8'ha8: 关闭。 备注: 同时也是外灌时钟使能信号, =1 使能外灌时钟。	R/W	0

### 5.3.18 VBAT 域 IO 口配置寄存器 VBAT\_IOMODE(0x88)

比特位	名称	描述	读/写标志	复位值
31:12	---	预留	R	0
11:8	---	预留, 未开放寄存器位	R/W	0
7:4	P56_MODE	=4'ha 时, 使能 P56 口输出 RTC_OUT 功能; =其他值, P56 功能由 GPIO 章节寄存器 定义。 该功能的优先级高于 GPIO 章节对于 P56 的复用配置。 当 VBAT 独立上电而 VCC 无电时, P56 默认输出 1Hz; 当 VCC 上电后, CPU 初始化会将此寄 存器配置为 0。	R/W	0
3:0	---	预留, 未开放寄存器位	R/W	0

### 5.3.19 VBAT 域 IO 口配置寄存器写使能 VBAT\_IOWEN(0x8C)

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:4	IOWEN	=4'hB 时, 使能 VBAT_IOMODE 寄存器写 操作; =其他, 不允许写 VBAT_IOMODE 寄存 器。	R/W	0
3:2	---	预留	R	0
1	---	预留, 未开放寄存器位	R/W	0
0	---	预留, 未开放寄存器位	R/W	0

## 5.4 RTC 时钟读写步骤

- 1、将系统控制章节中模块使能 1 寄存器 MOD1\_EN 第 10 位 RTC\_EN 设置为 1。
- 2、时钟的读取：读取 RTC 的秒、分、时等时间寄存器。
- 3、时钟的写入：

将 RTC\_CTL 的第 8 位 WRTC 置为 1，打开写使能操作。

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

或按照“秒分时分年月日”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。

## 5.5 RTC 校准步骤

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

## 5.6 RTC 定时器操作步骤

以定时器 1 产生 1S 中断为例，操作步骤为：

- 1、将系统控制章节中模块使能 1 寄存器 MOD1\_EN 第 10 位 RTC\_EN 设置为 1 时钟启动。
- 2、将 RTC\_CTL 的第 8 位 WRTC 置为 1，打开写使能操作。
- 3、设置 RTC->CNT1 = 0x00；即为 1S 产生 1 次中断。
- 4、设置 RTC->IE = 0x02；RTC 定时器 1 中断使能。
- 5、开启 RTC 中断使能，NVIC\_EnableIRQ(RTC\_IRQn)；
- 6、编写中断服务程序：

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

- 7、配置完成后既可产生 1S 中断。

## 6 WDT

内置硬件看门狗，用于检测程序的异常执行。

### 6.1 概述

看门狗具备如下特点：

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

出现以下任何一种情况时产生看门狗复位：

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

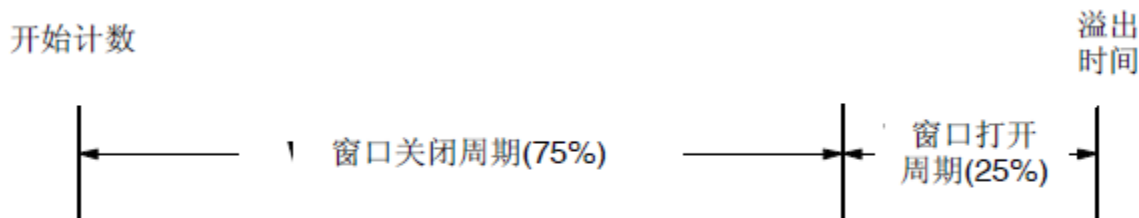
### 6.2 看门狗定时器的配置

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

名称	描述	厂家默认值
间隔中断	0: Disable（不使能间隔中断） 1: Enable（达到溢出事件的 75% 时产生间隔中断）	0
窗口打开周期	0: 25% 1: 50% 2: 75% 3: 100% 在窗口打开期间将 0xBB 写入 WDTE 寄存器，看门狗清零并重新计数； 在窗口关闭期间将 0xBB 写入 WDTE 寄存器，会产生内部复位信号。	3
溢出时间	0: 16ms 1: 32ms 2: 128ms 3: 512ms 4: 1s 5: 2s 6: 4s 7: 8s 备注：采用内部低频 RCL 计时，具体时间并不精确，但可保证溢出时间比上述标准时间要长。	4
CPU 睡眠设置	0: Disable（当 CPU 处于 sleep 或者 deepsleep 的时候不开启 WDT） 1: Enable（当 CPU 处于 sleep 或者 deepsleep 的时候开启 WDT）	0
CPU 调试设置	0: Disable（当 CPU 处于调试状态时不开启 WDT） 1: Enable（当 CPU 处于调试状态时开启 WDT） 注：CPU 处于调试状态指的是用户通过调试接口将 Cortex M0 停住	0

	(PC 指针停止计数)。如果芯片处于开发过程中，不建议使能该设置。因为如果使能该设置，当芯片处于调试状态时 WDT 仍然会计数，溢出时会产生中断，将引起调试无法进行。	
--	---	--

窗口打开周期的定义如下图所示，以 25%的窗口打开周期为示例：



### 6.3 寄存器描述

WDT 寄存器基址

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

WDT 寄存器偏移地址

寄存器名	地址偏移量	描述
WDT_EN	0x0	使能寄存器

WDT\_EN (0x0)

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

### 6.4 WDT 操作步骤

- 1、配置系统控制章节模块使能 1 寄存器 MOD1\_EN 第 9 位为 1，打开 WDT APB 时钟。
- 2、WDT 默认配置为启动，定时器溢出时间为 1S，窗口打开周期为 75%。用户程序可不进行 WDT 初始化配置。
- 3、喂狗操作：WDT->EN = 0xbb;
- 4、WDT 默认为睡眠后关闭 WDT，MOD1\_EN 中 WDT 时钟可不关闭，如关闭，需等待 WDT\_EN 的第 8 位 WR\_BUSY 为 0 后方能关闭 WDT 时钟。
- 5、当硬件仿真将程序停止运行时，WDT 计数也会暂停，不会影响硬件仿真。
- 6、完成。

使用建议：

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

## 7 LCD

MCU 内置段码式 LCD 控制器。

最大支持 8COM\*36SEG，支持电荷泵和电阻分压两种方式。

### 7.1 概述

LCD 具有如下特性

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

#### 7.1.1 扫描时钟频率

LCD 单元的时钟来自 fosc 或 RCL（由系统控制的 OSC\_CTL2 寄存器选择），频率为 32768Hz。Fosc 或 RCL 经过分频之后作为 LCD 波形扫描频率 flcd，波形扫描频率 flcd 可以通过寄存器 LCD\_CLKDIV[7:0]进行配置。

$F_{fram} = flcd / (\text{com 数目})$ 。一般要求 LCD 屏的刷新频率略微大于 60Hz。

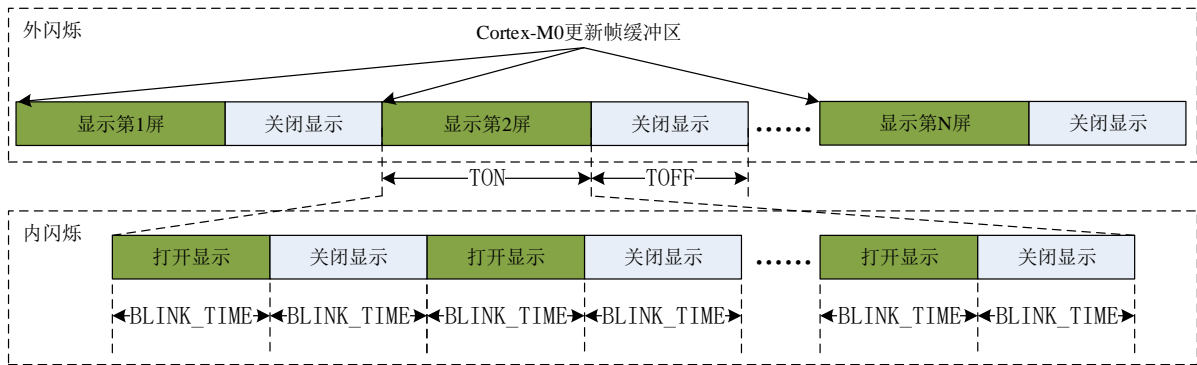
表 7-1 示出不同 duty 对应的时钟设置。（绿色标注的为正常使用的帧频）

LCD_CLKDIV	扫描频率	静态占空比	1/2 占空比	1/3 占空比	1/4 占空比	1/6 占空比	1/8 占空比
0xff	64Hz	64Hz	32Hz	21.3Hz	16Hz	10.7Hz	8Hz
0x7f	128Hz	128Hz	64Hz	42.7Hz	32Hz	21.3Hz	16Hz
0x54	192.8Hz	192.8Hz	96.4Hz	64.3Hz	48.2Hz	32.1Hz	24.0Hz
0x3f	256Hz	256Hz	128Hz	85.3Hz	64Hz	42.7Hz	32Hz
0x2a	381.3Hz	381.3Hz	190.5Hz	127.0Hz	95.3Hz	63.5Hz	47.6Hz
0x1f	512Hz	512Hz	256Hz	170.7Hz	128Hz	85.3Hz	64Hz

#### 7.1.2 闪烁模式

LCD 支持两种闪烁模式：内闪烁和外闪烁。两种模式可以同时使能。

图 7-1 LCD 闪烁方式



如图 7-1，LCD 使能后会根据 LCD\_BLINK 寄存器的 TON 位域定义的时间长度打开显示，随后根据 LCD\_BLINK 寄存器的 TON 位域定义的时间长度关闭显示。

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

### 7.1.2.1 内闪烁模式

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

### 7.1.2.2 外闪烁模式

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

### 7.1.3 LCD 驱动波形

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

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

用户需要根据应用所需的 COM 数选择 LCD 输出波形的占空比：

- ◆ 1 个 COM：选择静态占空比，只使用 COM0；
- ◆ 2 个 COM：选择 1/2 占空比，使用 COM0, COM1；
- ◆ 3 个 COM：选择 1/3 占空比，使用 COM0 ~COM2；
- ◆ 4 个 COM：选择 1/4 占空比，使用 COM0 ~COM3；
- ◆ 6 个 COM：选择 1/6 占空比，使用 COM0 ~COM5；
- ◆ 8 个 COM：选择 1/8 占空比，使用 COM0 ~COM7；

#### 7.1.3.1 LCD 类型 A 驱动波形

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

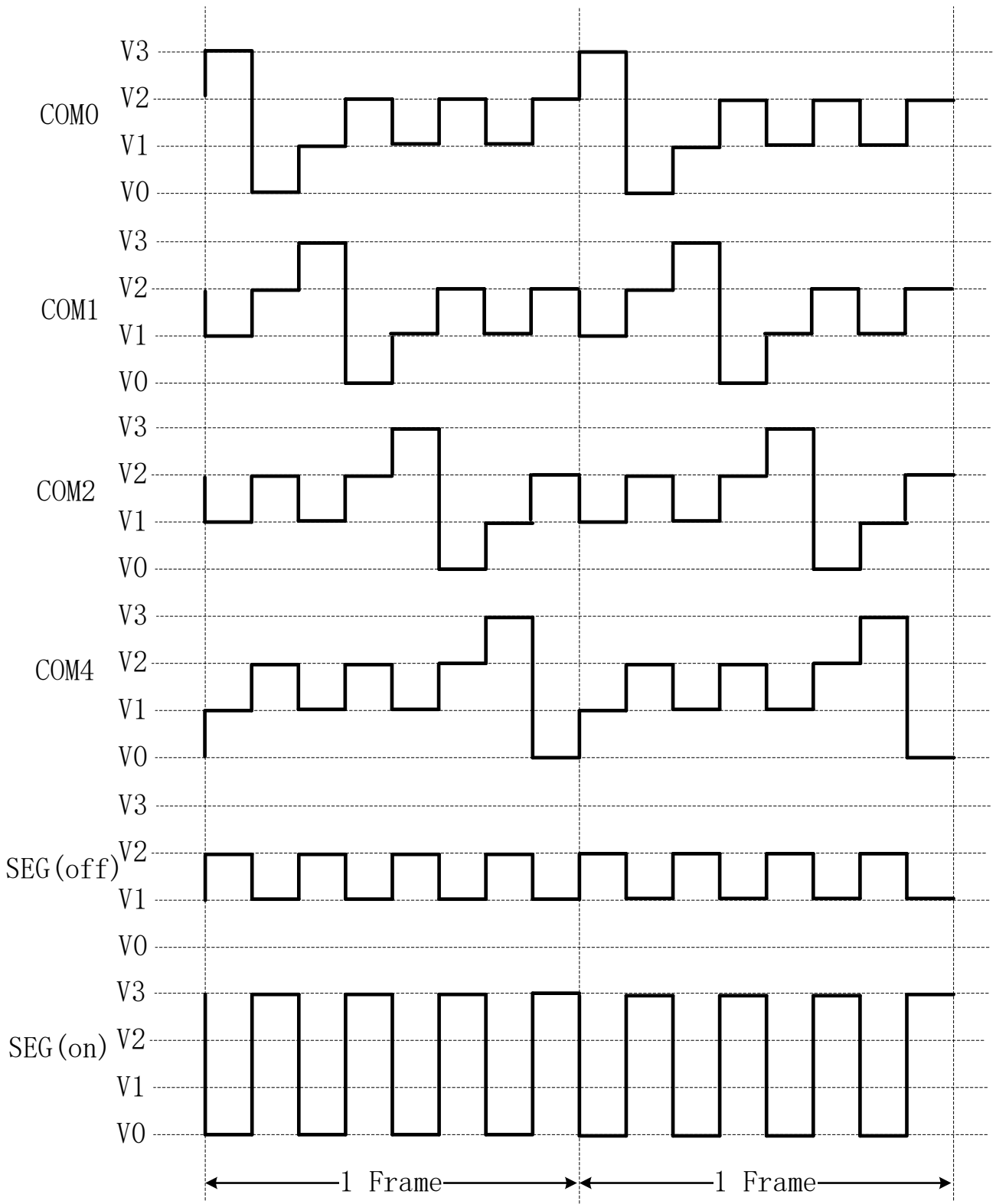
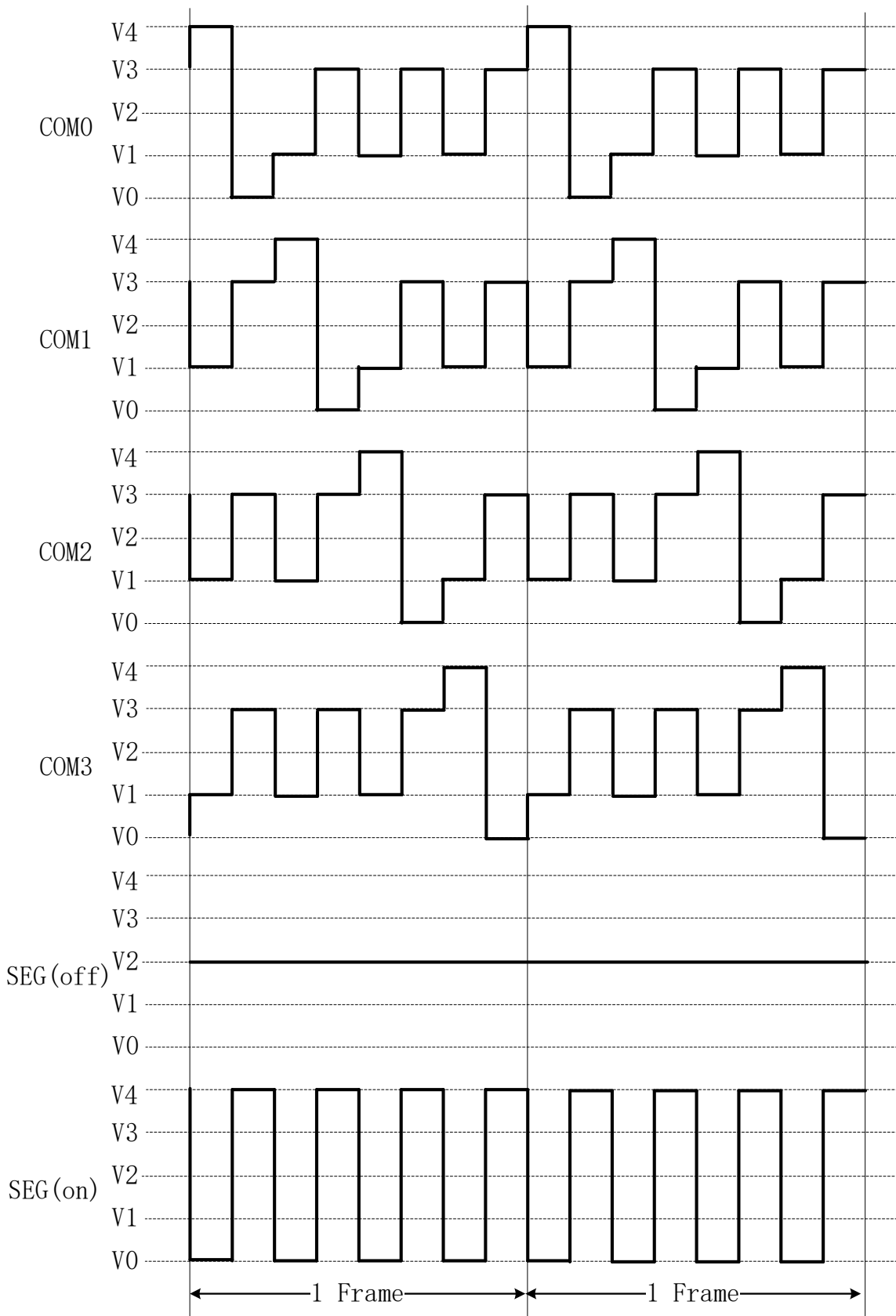


图 7-3 LCD 驱动波形 (1/4 Duty, 1/4 Bias, Type A)





7.1.3.2 类型 B 驱动波形

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

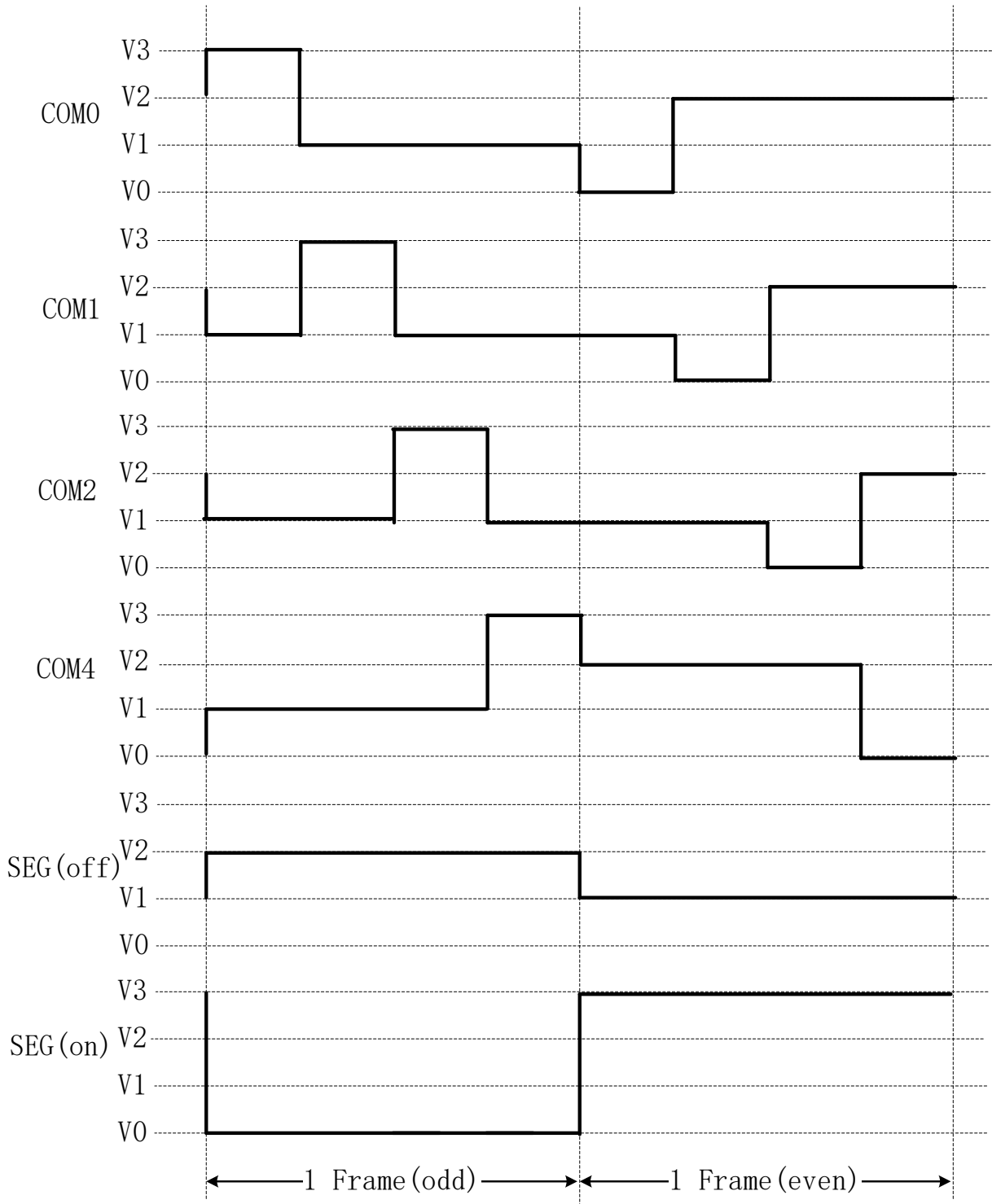
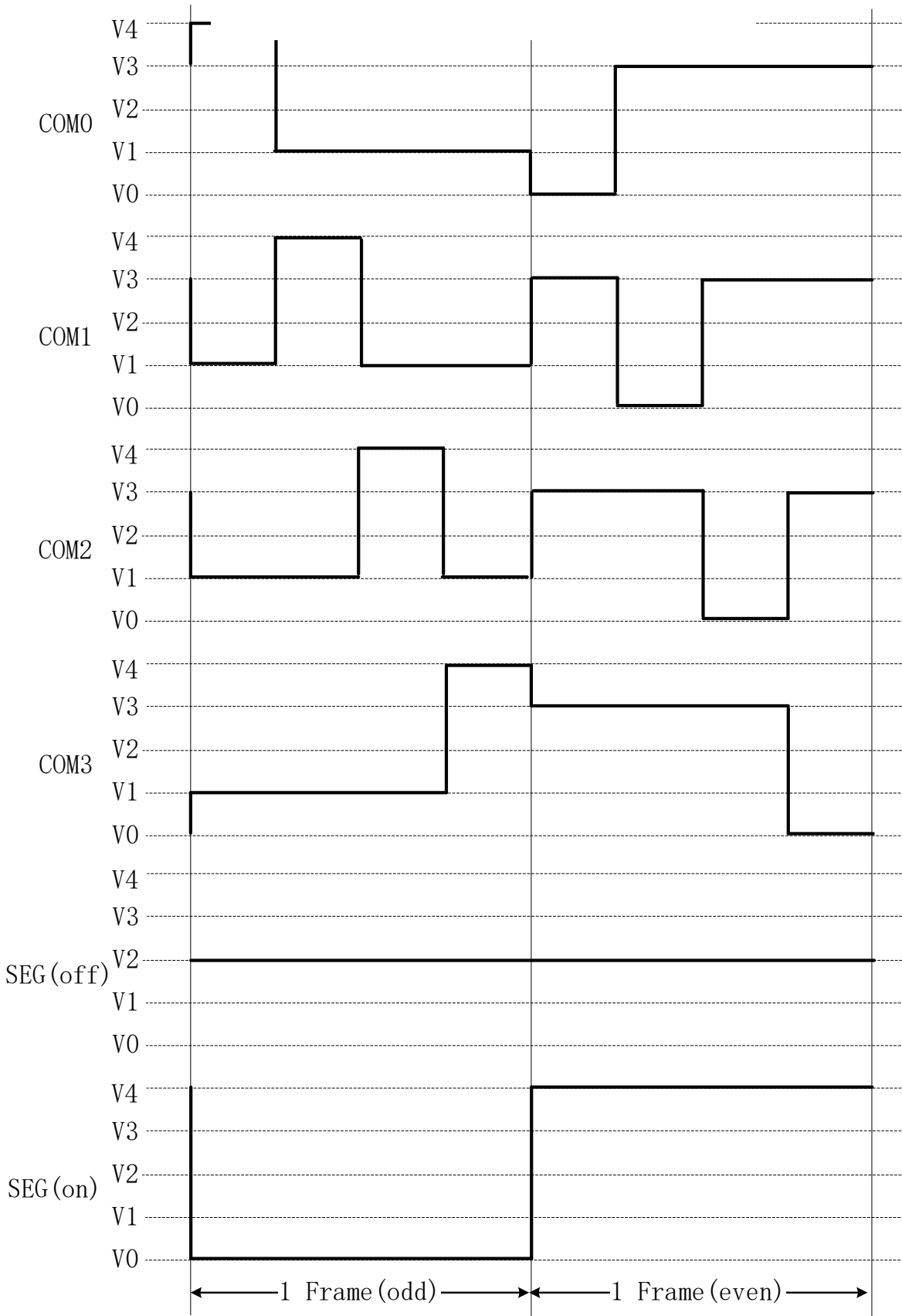


图 7-5 LCD 驱动波形 (1/4 Duty, 1/4 Bias, Type B)



### 7.1.4 偏置电压

#### 7.1.4.1 Charge Pmup 方式提供 LCD 偏置电压

LCD 的偏置电压可采用 Charge Pump 方式提供。Charge Pump 共需要产生 4 个电压 (Va, Vb, Vc, Vd), 以满足 1/4 偏压比的应用。对于不同偏压比设置, Charge Pump 输出的电压模式有所不同, 如表 7-2 所示。

表 7-2 LCD 驱动电压与偏压比关系

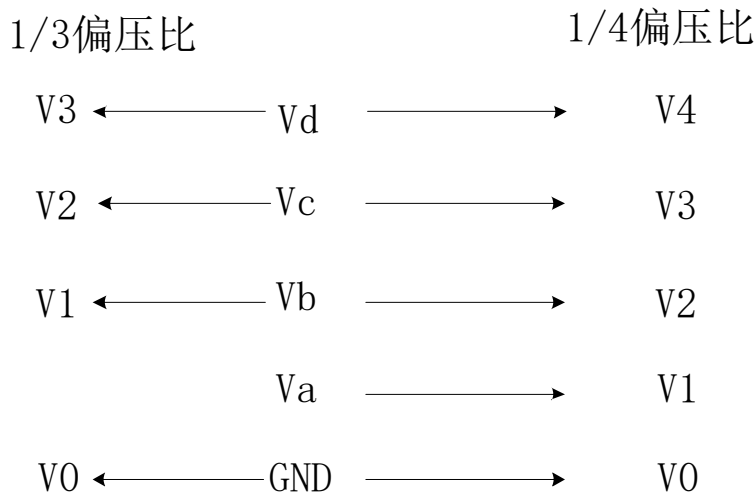
偏压比	灰度选择	Va	Vb	Vc	Vd	Vd (MAX)
1/3 偏压比	BIASLVL[5] = 0	$Va = Vref * (32 + BIASLVL[4:0]) / 63$	$Vb = Va$	$Vc = 2 * Va$	$Vd = 3 * Va$	3.75 V
	BIASLVL[5] = 1	$Va = Vref * (1 + BIASLVL[4:0] / 63)$	$Vb = Va$	$Vc = 2 * Va$	$Vd = 3 * Va$	5.59 V
1/4 偏压比	BIASLVL[5] = 0	$Va = Vref * (32 + BIASLVL[4:0]) / 63$	$Vb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

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

LCD 选择内部 Vref 做为基准, Vref 输出典型值为 1.25V。

1/3 和 1/4 偏压比应用电压选择如图 7-6 所示。

图 7-6 偏置电压选择



#### 7.1.4.2 内部电阻串分压方式提供 LCD 偏置电压

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

1/3 和 1/4 偏压比应用电压选择如图 7-6 所示。

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

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

### 7.1.5 LCD BUFFER 映射

LCD\_BUFx 寄存器与不同段码规格的 LCD 屏映射关系如下所示。

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

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

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

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

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

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

## 7.2 寄存器描述

LCD 模块的基址

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

LCD 模块的寄存器偏移地址

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

### 7.2.1 LCD 控制寄存器 LCD\_CTL(0x00)

偏移地址 0x00

比特位	名称	描述	读/写标志	复位值
31:13	Reserved	预留	R	0
12	PWD_PUMP	LCD PUMP 开关： 0: 开启 PUMP，LCD 电压由内部 PUMP 产生。 1: 关闭 PUMP，使用电阻串分压方案； 备注：RN8612 为保留寄存器位，用户不要操作。	R/W	0

11	TYPE	LCD Drive Type Select 0: Type A 1: Type B	R/W	0
10:5	BIASLVL	LCD Bias 电压调节 控制 Charge Pump 输出不同幅度的电压以控制 LCD 的对比度 <b>备注: RN8612 为保留寄存器位, 用户不要操作。</b>	R/W	0
4	BIAS	LCD Bias 控制 0: 1/3Bias 1: 1/4Bias	R/W	0
3:1	DUTY	LCD 占空比控制 000: 静态输出 (COM0) 001: 1/2 占空比 (COM0~1) 010: 1/3 占空比 (COM0~2) 011: 1/4 占空比 (COM0~3) 100: 1/6 占空比 (COM0~5) 101: 1/8 占空比 (COM0~7) Other:预留	R/W	0
0	EN	LCD 模块使能 0: LCD 模块关闭 1: LCD 模块使能 Note: LCD 模块关闭时, LCD 进入待机模式, 相关模拟电路也要关闭, 以保证低功耗要求。	R/W	0

### 7.2.2 LCD 状态寄存器 LCD\_STATUS(0x04)

偏移地址 0x04

比特位	名称	描述	读/写标志	复位值
31:7	Reserved	预留	R	0
6	LCD_BUSY	LCD Busy Bit 0: 不忙 1: 忙 Note: 当 LCD_BUSY 为 1 时, LCD_CTRL (除了 EN Bit 外)、LCD_CLKDIV、LCD_BLINK、LCD_PS 寄存器不可修改	R	0
5	保留	保留寄存器位, 用户不要操作		
4	保留	保留寄存器位, 用户不要操作		
3	IRQOFFEN	Display Off IRQ Enble Bit 0: 不使能 1: 使能	R/W	0
2	IRQONEN	Display On IRQ Enble Bit 0: 不使能 1: 使能	R/W	0
1	DOFF	Display Off Pending Bit	R/W	0

		0: 无中断事件 1: 显示由亮变灭时置位 Note: 写 1 清零		
0	DON	Display On Pending Bit 0: 无中断事件 1: 显示由灭变亮时置位 Note: 写 1 清零	R/W	0

### 7.2.3 LCD 时钟控制寄存器 LCD\_CLKDIV(0x08)

偏移地址 0x08

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

### 7.2.4 LCD 闪烁控制寄存器 LCD\_BLINK(0x0c)

偏移地址 0x0c

比特位	名称	描述	读/写标志	复位值
31:24	Reserved	预留	R	0
25:18	BLINK_TIME	步长是 0.25s, 支持 0~63.75s 设置 TON 显示周期内, 亮与灭的时间 = 0.25*BLINK_TIME。 Note: 当设置为 0 时, 代表在 TON 显示周期内长亮, 不闪烁。当设置值大于 0 时, TON 必须为 BLINK_TIME 的 2n 倍 (n 为大于 0 的整数)。	R/W	0
17:9	TOFF	步长是 0.25s, 支持 0~127.5s, 请设置成大于>3s 0.25*TOFF	R/W	0
8:0	TON	步长是 0.25s, 支持 0~127.5s, 请设置成大于>3s 0.25*TON	R/W	0

### 7.2.5 LCD 闪烁控制寄存器 LCD\_PS(0x10)

LCD PUMP 建立时间寄存器

Offset=0x10

比特位	名称	描述	读/写标志	复位值
31:12	---	预留	R	0
11:0	PS	LCD PUMP Setup time Time = Tosc * (PS+4) (Tosc 为 30.5uS)	R/W	0xccc

备注: RN8612 不支持该寄存器。

### 7.2.6 LCD 内部电阻串控制寄存器 LCD\_RESCTL(0x14)

偏移地址 0x14

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

12:9	LDOS	<p>LDO 输出电平选择信号 LDO 输出作为 LCDVD 电压，并通过电阻分压产生 LCDVC、LCDVB、LCDVA； LDO 输出电平从 2.7V~3.6V，共 16 档，档位步进 0.06V；</p> <table border="1"> <thead> <tr> <th>LDOS 配置</th> <th>LDO 输出电压 V</th> </tr> </thead> <tbody> <tr><td>0000</td><td>2.7</td></tr> <tr><td>0001</td><td>2.76</td></tr> <tr><td>0010</td><td>2.82</td></tr> <tr><td>0011</td><td>2.88</td></tr> <tr><td>0100</td><td>2.94</td></tr> <tr><td>0101</td><td>3</td></tr> <tr><td>0110</td><td>3.06</td></tr> <tr><td>0111</td><td>3.12</td></tr> <tr><td>1000</td><td>3.18</td></tr> <tr><td>1001</td><td>3.24</td></tr> <tr><td>1010</td><td>3.3</td></tr> <tr><td>1011</td><td>3.36</td></tr> <tr><td>1100</td><td>3.42</td></tr> <tr><td>1101</td><td>3.48</td></tr> <tr><td>1110</td><td>3.54</td></tr> <tr><td>1111</td><td>3.6</td></tr> </tbody> </table> <p>复位时 LDO 输出 3.0V</p>	LDOS 配置	LDO 输出电压 V	0000	2.7	0001	2.76	0010	2.82	0011	2.88	0100	2.94	0101	3	0110	3.06	0111	3.12	1000	3.18	1001	3.24	1010	3.3	1011	3.36	1100	3.42	1101	3.48	1110	3.54	1111	3.6	R/W	0101
LDOS 配置	LDO 输出电压 V																																					
0000	2.7																																					
0001	2.76																																					
0010	2.82																																					
0011	2.88																																					
0100	2.94																																					
0101	3																																					
0110	3.06																																					
0111	3.12																																					
1000	3.18																																					
1001	3.24																																					
1010	3.3																																					
1011	3.36																																					
1100	3.42																																					
1101	3.48																																					
1110	3.54																																					
1111	3.6																																					
8	FCC	<p>快速充电控制： 0：关闭快速充电功能 1：打开快速充电功能； 选择内部电阻分压，节点电压外接 470nf 电容，LCD 模块打开时，内部电阻调整至 5k，并保持 100ms，以完成对电容的快速充电。</p>	R/W	1																																		
7	RES_AO	<p>电阻串分时驱动开关 0：电阻串分时驱动，根据 RES_DT 和 RES_FT 产生驱动信号 1：电阻串始终被驱动，RES_DT 和 RES_FT 配置无效。 选择 TYPE A 时，RES_AO 恒为 1。</p>	R/W	1																																		
6: 4	RES_DT	<p>电阻串分压方式，刷屏周期内，20k 电阻驱动时间配置 驱动时间 <math>T_d = (\text{BIAS\_DT}[2:0] + 1) * T_{osc}</math> (<math>T_{osc}</math> 为 30.5uS)。</p>	R/W	000																																		
3: 2	RES_FT	<p>电阻串分压方式，刷屏周期内，20k 电阻驱动次数配置 00：驱动 1 次 01：驱动 2 次 10：驱动 3 次 11：驱动 4 次</p>	R/W	00																																		
1	RSM	<p>内部电阻分压电阻模式选择信号 0 小电阻+开路，20k 电阻分时开启，200k 电阻始终短</p>	R/W	0																																		

		接 1 大小电阻切换, 20k 和 200k 电阻分时切换		
0	IRSN	电阻分压模式选择 0 内部电阻分压模式 1 保留 如果使用电阻分压方式, 该寄存器只能配置为 0, 配置为 1 没有意义。	R/W	1

如果选择为电阻分压方式, 建议对 LCD\_RESCTL 配置如下:

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

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

### 7.2.7 LCD 数据寄存器 LCD\_BUF[i] (8 位寄存器)

偏移地址 0x20-0x45

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

## 7.3 LCD 操作步骤

- 1、将系统控制章节 MOD1\_EN 模块使能 1 寄存器第 6 位配置为 1, 打开 LCD 模块时钟门控。
- 2、将 LCD 控制寄存器 LCD\_CTL 设置为 0, 关闭 LCD 模块。
- 3、等待 LCD\_STATUS 状态寄存器第 6 位是否为 0, 空闲状态, 如空闲则进入下一步, 否则等待, 等待时间大概为 16MS 左右。
- 4、设置 LCD 控制寄存器 LCD\_CTL, 设置 LCD 显示模式、液晶显示电压、开启 PUMP。例如使用 8COM, 5v 液晶, LCD\_CTL 值为 0x3fb。也可配置 LCD\_RESCTL, 选择电阻分压方式。
- 5、设置 LCD 时钟控制寄存器 LCD\_CLKDIV 设置 LCD 的分频系数。
- 6、设置 LCD 闪烁控制寄存器, 如不需闪烁功能则可直接配置为 0x01。
- 7、完成初始化设置, 可直接将显示数据送入 LCD\_BUF[i]中即可。



## 8 定时器

内置2个32位定时器。每个定时器可完全独立工作；定时器之间不共享任何资源，可同步操作。  
定时器适合多种用途，具有如下功能：

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

### 8.1 概述

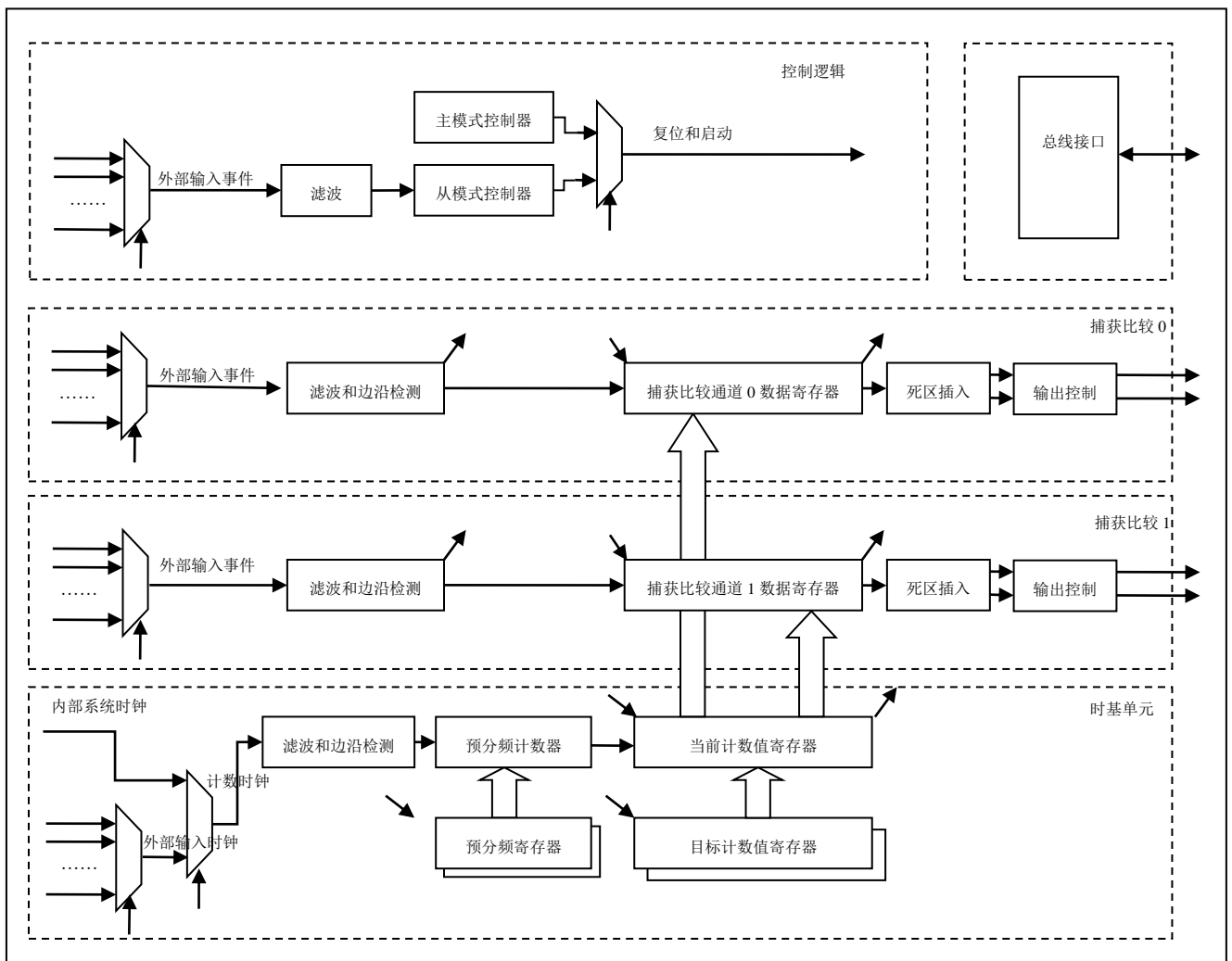
定时器具备如下特性：

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

- 固定电平输出;
- 脉宽可配置脉冲输出;
- 比较寄存器随时更新;
- 支持中断:
  - 计数溢出;
  - 输入捕获;
  - 输出比较;

### 8.2 功能框图

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



### 8.3 寄存器描述

模块寄存器基地址

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

TC 模块寄存器偏移地址

寄存器名	地址偏移量	描述
TC_CNT	0x0	当前计数值指示

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

#### 8.3.1.1.1 当前计数值寄存器 TC\_CNT (0x00)

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

#### 8.3.1.1.2 预分频寄存器 TC\_PS (0x04)

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

#### 8.3.1.1.3 目标计数值寄存器 TC\_DN (0x0C)

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

#### 8.3.1.1.4 捕获比较通道 0 数据寄存器 TC\_CCD0 (0x014)

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

注: 通道 0 配置为捕获功能 (即 TC\_CM0 寄存器的 CCM 位域为 0) 时, TC\_CCD0 寄存器不可写

#### 8.3.1.1.5 捕获比较通道 1 数据寄存器 TC\_CCD1(0x018)

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

注: 通道 1 配置为捕获功能 (即 TC\_CM1 寄存器的 CCM 位域为 0) 时, TC\_CCD1 寄存器不可写

#### 8.3.1.1.6 时钟配置寄存器 TC\_CCFG (0x01C)

比特位	名称	描述	读/写标志	复位值
31:24	-	保留	R	0
23:16	FLTPT	外部输入时钟滤波参数设置, 设置滤波的时钟周期数。	R/W	0

15	-	保留	R	0
14:13	ECLKMODE	外部输入时钟模式： 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留（等效于双边沿）	R/W	0
12:8	CS	外部输入时钟选择： 0: UART0 RXD 1: UART1 RXD 2: UART2 RXD 3: UART3 RXD 4: 另一定时器(TC0或TC1)的输出outn[0] 5: 另一定时器(TC0或TC1)的输出outp[0] 6: 另一定时器(TC0或TC1)的输出outn[1] 7: 另一定时器(TC0或TC1)的输出outp[1] 8: UART4 RXD 9: UART5 RXD 10:7816_0输入P41 11:7816_1输入P42 12:7816_1输入P43 13~15:保留 16: sf_out 17: qf_out 18: pf_out 19: rtc_out 20: p1[0]外部I0口 21: p1[1] 外部I0口 22: p1[2]、P5[2]外部I0口 23: p1[3]、P5[3]外部I0口 24: p1[4]、P5[4]外部I0口 25: p1[5]、P5[5]外部I0口 26: p1[6]、P5[6]外部I0口 27: p1[7]、P5[7]外部I0口 28: p3[0]外部I0口 29: p3[1]外部I0口 30: p3[3]外部I0口 31: p3[5]外部I0口	R/W	0
7:2	-	保留	R	0
1	FLTEN	外部输入时钟滤波使能 0: 不使能 1: 使能	R/W	0
0	CM	计数时钟源选择： 0: 内部系统时钟 1: 外部输入时钟（由 CS 选择时钟源）	R/W	0

## 8.3.1.1.7 控制寄存器 TC\_CR (0x020)

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

## 8.3.1.1.8 捕获比较通道 0/1 模式寄存器 TC\_CM0/1 (0x024 和 0x028)

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

### 8.3.1.1.9 中断使能寄存器 TC\_IE (0x2C)

比特位	名称	描述	读/写标志	复位值
31:4	-	保留	R	0
3	SLVIE	从模式中断使能： 0：不使能 1：使能	R/W	0
2	CC1IE	捕获比较通道1中断使能： 0：不使能 1：使能	R/W	0
1	CC0IE	捕获比较通道0中断使能： 0：不使能 1：使能	R/W	0
0	OVIE	溢出中断使能： 0：不使能 1：使能	R/W	0

### 8.3.1.1.10 状态寄存器 TC\_STA (0x30)

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

## 8.4 典型应用

### 8.4.1 自动运行模式，定时功能

自动运行模式即间隔定时功能。

**基本定时功能，只需要对以下寄存器进行设置：**

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

定时器会以目标计数值为周期产生中断。

**常用可选择功能配置说明：**

- 1、可修改预分频寄存器值，改变定时器计数时钟的频率。
- 2、可配置时钟配置寄存器，CM 修改配置为外部输入时钟，同时修改 CS 位配置的外部输入时钟选择。外部输入时钟频率不可比内部系统时钟频率的二分频高。
- 3、单次计数模式，定时器溢出后就停止；配置控制寄存器的 OPS 位为 1 即为单次计数模式。
- 4、外部输入时钟源为另外一个定时器的输出，则可连接为两个定时器的级联模式，可增加定时器的寄存器的位宽。

**8.4.2 输入捕获模式，脉宽测量功能**

输入捕获模式的主要功能是可以测试脉冲的宽度。

**基本脉宽测试功能，只需要对以下寄存器进行设置：**

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

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

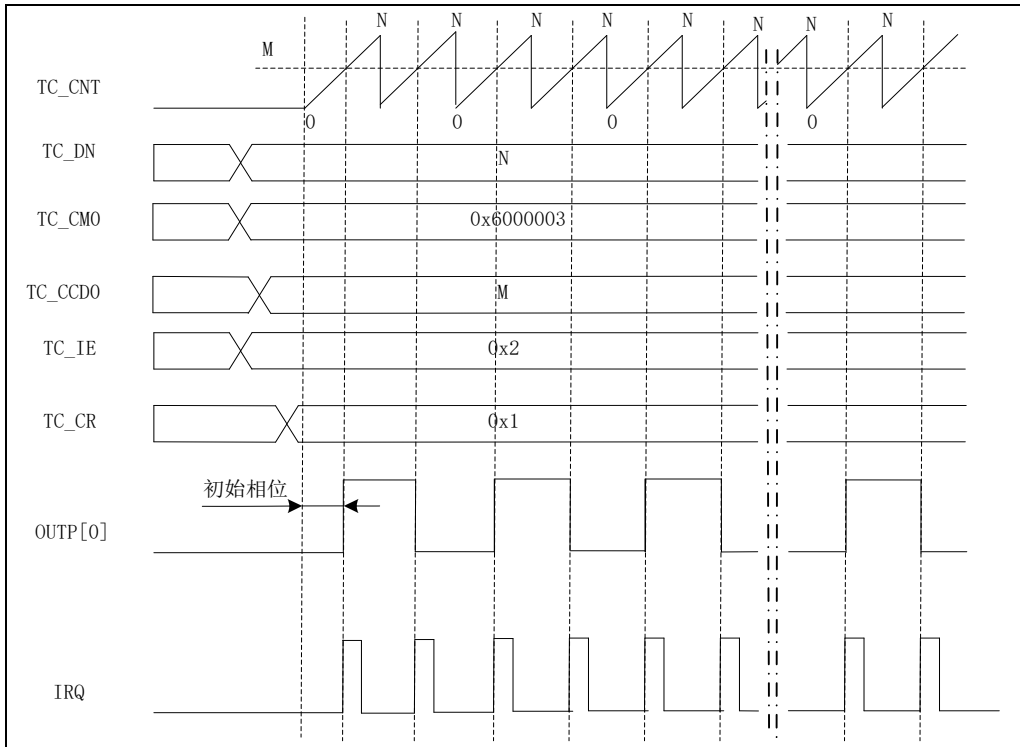
**常用可选择功能配置说明：**

- 1、可修改预分频寄存器值，改变定时器计数时钟频率。
- 2、可配置时钟配置寄存器，CM 修改配置为外部输入时钟，同时修改 CS 位配置的外部输入时钟源。外部输入时钟源频率不可比内部系统时钟频率的二分频高。
- 3、外部输入事件的滤波功能，使能捕获比较通道模式寄存器中 FLTEN 滤波功能，通过配置 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。

**8.4.3 比较输出模式，方波输出功能**

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





**方波输出功能，只需要对以下寄存器进行设置：**

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

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

**常用可选择功能配置说明：**

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

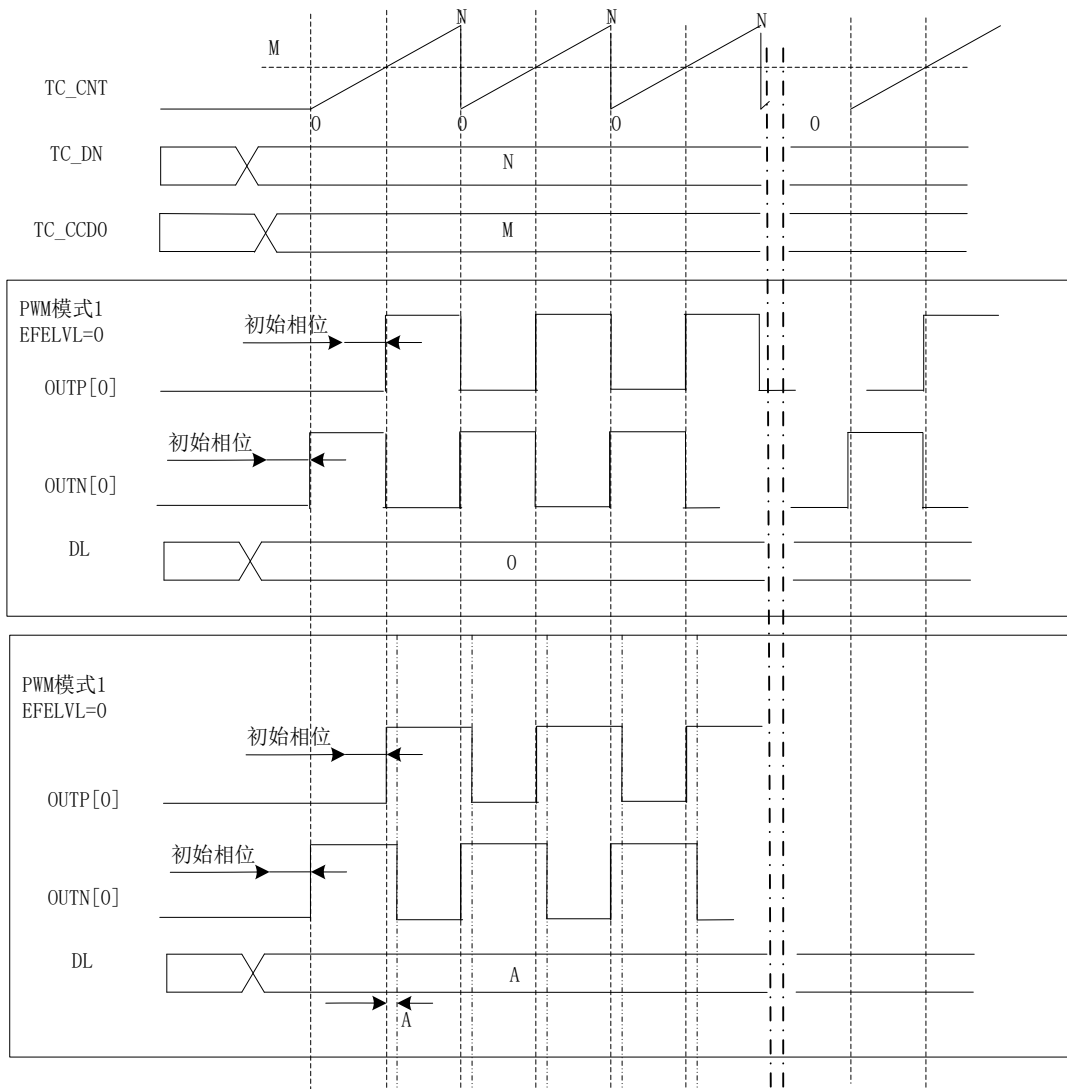
**8.4.4 比较输出模式，PWM 输出功能**

脉冲宽度调制（PWM）模式可以产生一个由 TC\_DN 寄存器确定频率、由 TC\_CCDO 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\_CCDO 时，输出为有效电平，否则为无效电平。

PWM 模式 2：如果 TC\_CNT >= TC\_CCDO 时，输出为有效电平，否则为无效电平。

下图为 PWM 模式 1 的典型应用图。



**PWM 输出功能，只需要对以下寄存器进行设置：**

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

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

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

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

**常用可选择功能配置说明：**

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

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

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

### 8.4.5 从模式，外部清零和门控功能

从模式在原有的功能上增加了外部输入事件控制内部计数器的清零和门控的功能。

**外部清零功能，只需要对以下寄存器进行设置：**

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

这样在自由运行模式下增加了一个外部清零内部 CNT 的功能。

**常用可选择功能配置说明：**

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

## 8.5 操作步骤

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

- 1、配置系统控制章节中模块使能 0 寄存器 MOD0\_EN 第 4 位启动 TC0 的时钟。注意写系统控制章节寄存器需先将 SYS\_PS 写为 0x82，打开写使能，需写入的系统控制寄存器操作完成后需将 SYS\_PS 写入 0，关闭写使能。

- 2、设置 TC0 模块寄存器：

时钟配置寄存器 TC\_CCFG 配置为内部系统时钟，TC0->CCFG = 0；

配置预分频寄存器 TC\_PS；

配置目标计数值寄存器 TC\_DN；

配置控制寄存器 TC\_CR，TC0->CTRL = 0x01；启动 TC0 定时器

设置中断使能寄存器 TC\_IE，TC0->IE = 0x01；配置为溢出中断使能；

当系统时钟为 3.6864MHZ 时，TC0->PS = 255；TC0->DN = 13 配置后既可产生 1MS 中断。 $(3.6864\text{MHZ} / (255+1)) / (13+1) = 1\text{MS}$ 。

- 3、开启 TC0 中断使能，NVIC\_EnableIRQ(TC0\_IRQn)；

- 4、编写中断服务程序：

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

- 5、配置完成后即可产生 1MS 中断。

## 9 模拟外设

### 9.1 特点

10bit SAR ADC;

- 对 VBAT 引脚和 AIN6 引脚输入的电压启动测量时，内部有两个 300K 电阻做分压，输入电压会被缩小一半输入到 SAR ADC，SAR ADC 使用 0.5 倍 PGA 对信号再缩小一倍，即：3.6V 的电池，测量到的信号约为 0.9V。如果不启动测量，内部分压电阻会被关闭。
- 9 通道模拟多路选择器，其中 AIN0 – AIN6 通道用于外部模拟信号测量，VBAT 用于电池电压测量，Temp Sensor 用于测量温度传感器的输出；温度传感器的优先级最高。
- AIN0~AIN5 的输入阻抗约为 3M 欧姆；
- 除了 AIN6，其他 AIN 最大支持 1.25V 输入。增益支持 0.5 倍、1 倍、1.5 倍、2 倍
- ADC 不采样时，自动进入省电模式；每次采样从启动到完成约为 2ms。

一路 LVD 电路，主要特点如下：

- LVD 的输入可选为芯片电源，也可选择为外部 PIN 输入；
- LVD 的阈值可调，从 2.3V 到 4.9V 共分为 14 档；
- 当选择为外部 PIN 输入时，阈值固定为 1.25V 左右，内阻约为 1M 欧姆

两路比较器电路 CMP1 和 CMP2，主要特点如下：

- 外部 PIN 输入，阈值约为 1.25V 左右
- 功耗优于 1uA，可使用比较器进行主电上下电监测。
- 注意 CMP1 和 CMP2 内部有 600K 的下拉电阻，可通过寄存器（SYS\_PD (0x08)）选择关闭内部电阻。

### 9.2 寄存器

模拟外设模块的基址

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

模拟外设模块的寄存器偏移地址

寄存器名	地址偏移量	描述
SAR_CTL	Offset+0x0	SAR-ADC 控制寄存器
SAR_START	Offset+0x4	SAR-ADC 启动寄存器
SAR_STAT	Offset+0x8	SAR-ADC 状态寄存器
SAR_DAT	Offset+0xC	SAR-ADC 数据寄存器
LVD_CTL	Offset+0x10	比较器控制寄存器
LVD_STAT	Offset+0x14	比较器状态寄存器
ANA_PAD	Offset+0x3c	VBAT 域模拟 PAD 配置寄存器
ANA_RST	Offset+0x40	VBAT 域软件复位寄存器

#### 9.2.1 ADC 控制寄存器 SAR\_CTL(0x00)

偏移地址 0x00

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

31:17	Reserved	预留	R	0
16:12	REF_WAIT	<p>从开启 REF 到开启 ADC 需要等待的时间： 等待时间最小刻度的典型值是 <b>30.5 微秒</b>。</p> <p>5'd0: <b>30.5 微秒</b></p> <p>5'd31: (31+1)* <b>30.5 微秒</b></p> <p><b>建议客户应用程序将该寄存器配置为 0x10。</b></p>	R/W	0x5
11:7	SAR_WAIT	<p>SAR ADC 在开启后到开始采样转换需要等待的时间：</p> <p>5'd0: <b>30.5 微秒</b></p> <p>5'd31: (31+1)* <b>30.5 微秒</b></p> <p>即等待时间=(SAR_WAIT+1)*<b>30.5 微秒</b></p> <p>注：启动 ADC 测量的步骤： 开启 REF，等待 REF_WAIT 时间，建议使用默认值； 开启 ADC 和温度传感器，等待 SAR_WAIT 时间； 16 个 SAR 采样时钟周期后得到采样结果。 以上步骤均由硬件自动控制实现。</p> <p><b>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。</b></p>	R/W	0xE
6	SAR_CH3	<p>SAR-ADC Channel 选择，与 SAR_CH[2:0]共同组成一个 4bit 寄存器。</p> <p>{SAR_CH3,SAR_CH[2:0]}=0xxx: 见 SAR_CH 定义</p> <p>{SAR_CH3,SAR_CH[2:0]}=1000: 选择 AIN6</p> <p>{SAR_CH3,SAR_CH[2:0]}=其他: 保留</p>	R/W	0
5	SAR_IE	<p>SAR-ADC 中断控制：</p> <p>1: 使能 ADC 中断输出；</p> <p>0: 不使能 ADC 中断输出。</p>	R/W	0
4:3	SAR_PGA	<p>SAR-ADC 增益控制：</p> <p>00: 0.5 倍</p> <p>01: 1 倍</p> <p>10: 1.5 倍</p> <p>11: 2 倍</p>	R/W	0
2:0	SAR_CH	<p>SAR-ADC Channel 选择</p> <p>000: Temp Sensor</p> <p>001: VBAT (对于 3.6V 电池，1/2 分压后得到 1.8V，PGA 采用 0.5 倍，测量输入 0.9V)</p> <p>010: AIN0</p> <p>011: AIN1</p> <p>100: AIN2</p> <p>101: AIN3</p> <p>110: AIN4</p> <p>111: AIN5</p> <p>Note: 不管设置为哪个通道，自动温度测量的优先级最高。</p>	R/W	0

备注：以上寄存器位只有当 SAR\_START 中的 ST=0 时才可写。

### 9.2.2 SAR-ADC 启动寄存器 SAR\_START(0x04)

偏移地址 0x04

比特位	名称	描述	读/写标志	复位值
31:01	Reserved	预留	R	0
0	ST	SAR-ADC Start Bit 0: SAR-ADC 无操作 1: 启动一次 SAR-ADC 采样, 完成采样后自动清零 Note: 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; 当 TPS_BUSY 为=1 时, ADC_START 寄存器不可写。	R/W	0

### 9.2.3 SAR-ADC 状态寄存器 SAR\_STATUS(0x08)

偏移地址 0x08

比特位	名称	描述	读/写标志	复位值
31:02	Reserved	预留	R	0
1	TPS_BUSY	自动温度测量忙, =1: 自动温度测量正在进行; =0: 自动温度测量没有进行。 当 TPS_BUSY 为=1 时, 软件写 ADC_START 寄存器, 硬件操作会在等待 TPS_BUSY 为 0 后再进行。	R	0
0	DREADY	ADC 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

### 9.2.4 ADC DAT 寄存器 SAR\_DAT(0x0c)

偏移地址 0x0C

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

### 9.2.5 LVD 控制寄存器 LVD\_CTL(0x10)

偏移地址 0x10

比特位	名称	描述	读/写标志	复位值
31:8	Reserved	预留	R	0
7	CMP2IE	比较器 2 中断使能: =0: 不使能中断;	R/W	0

		=1: 使能中断;		
6	CMP1IE	比较器 1 中断使能: =0: 不使能中断; =1: 使能中断;	R/W	0
5	LVDIE	LVD 中断使能: =0: 不使能中断; =1: 使能中断;	R/W	0
4	Reserved	预留	R	0
3:0	LVDS	LVD 阈值电压设置: 0000 2.7 0001 2.7 0010 2.7 0011 2.9 0100 3.1 0101 3.3 0110 3.5 0111 3.7 1000 3.9 1001 4.1 1010 4.3 1011 4.5 1100 4.7 1101 4.9 1110 检测外部管脚 LVDIN0 的电压, 与 LBGR (1.25V) 进行比较; 1111 保留;	R/W	0

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

### 9.2.6 LVD 状态寄存器 LVD\_STAT(0x14)

偏移地址 0x14

比特位	名称	描述	读/写标志	复位值
31:7	Reserved	预留	R	0
6	CMP2IIF	比较器 2 中断标志 =0: 未产生中断; =1: 产生中断; 当输入电压相对于阈值变低或者变高时产生中断, 写 1 清零;	R/W	0
5	CMP1IIF	比较器 1 中断标志 =0: 未产生中断; =1: 产生中断; 当输入电压相对于阈值变低或者变高时产生中断, 写 1 清零;	R/W	0
4	LVDIIF	LVD 中断标志 =0: 未产生中断; =1: 产生中断; 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	Reserved	预留	R	0
2	CMP2IF	比较器 2 状态标志 =0: 低于阈值; =1: 高于阈值; 只读	R	0
1	CMP1IF	比较器 1 状态标志 =0: 低于阈值; =1: 高于阈值; 只读	R	0
0	LVDIF	LVD 状态标志 =0: 低于阈值; =1: 高于阈值; 只读	R	0

### 9.2.7 ANA\_PAD

ANA PAD 寄存器

Offset = 0x3c

比特位	名称	描述	读/写标志	复位值
31:3	---	预留	R	0
1	P45_AEN	P45 口模拟信号测试输出开关 0: 关; 1: 开;	R/W	1'b0
0	P44_AEN	P44 口模拟信号测试输出开关 0: 关; 1: 开;	R/W	1'b0

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

### 9.2.8 ANA\_RST

ANA RST 寄存器

Offset = 0x40

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

## 9.3 ADC 电压检测步骤

- 1、配置系统控制章节模块使能 1 寄存器 MOD1\_EN 第 11 位为 1，开启 SAR\_EN 时钟。
- 2、判断 SAR-ADC 状态寄存器 SAR\_STATUS 第 1 位是否为 0，没有进行自动温度测量，如为 0 进入下一步，如为 1 则等待。
- 3、配置 ADC 控制寄存器 SAR\_CTL，配置等待时间和增益，通道选择对应的 ADC 通道。
- 4、SAR-ADC 启动寄存器 SAR\_START 配置为 1，启动 ADC 转换。
- 5、判断 SAR-ADC 状态寄存器 SAR\_STATUS 第 0 位，等待转换完成。
- 6、读取 ADC 转换数据寄存器 SAR\_DAT。
- 7、计算：ADC 基准电压源为 1.25V，ADC 满度时 ADC DAT 寄存器值为 1024。计算公式即为  $(ADC\ DAT * 1.25) / 1024$ 。当电压值超过 ADC 的满度，转换值为 1024。
- 8、转换及计算完成。

## 9.4 VBAT 电压检测

- 1、配置系统控制章节模块使能 1 寄存器 MOD1\_EN 第 11 位为 1，开启 SAR\_EN 时钟。
- 2、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。

## 9.5 低电压检测应用

- 1、对带电池的电能表，为节省电池功耗，需判断外部电源状态，当电源为低时让系统进入睡眠模式。电源检测模块需一直开启，CMP2/CMP1 比较器为极低功耗比较器，可用此比较器检测电源电压，并在电池模式下工作，CMP 输入管脚电压与基准 1.25V 进行比较，并产生中断及状态标志。
- 2、配置系统控制章节模块使能 1 寄存器 MOD1\_EN 第 11 位为 1，开启 SAR\_EN 时钟。
- 3、配置系统控制章节系统掉电控制寄存器 SYS\_PD 的第 5 位置 0，开启 CMP2 电源。
- 4、配置 LVD 控制寄存器 LVD\_CTL，使能 CMP2 中断。开启 CMP2 中断 NVIC\_EnableIRQ(CMP\_IRQn);
- 5、编写中断服务程序：

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

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

- 6、完成。

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

## 10 GPIO

### 10.1 概述

- 包含 PA、PB、PC 三组 GPIO
- PA 口包含 5 个 P0 口、8 个 P1 口、8 个 P2 口、8 个 P3 口
- PB 口包含 8 个 P4 口、8 个 P5 口、8 个 P6 口、8 个 P7 口
- PC 口包含 8 个 P8 口、8 个 P9 口、8 个 P10 口、8 个 P11 口
- GPIO 是 AHB 的外设
- 支持 bitband 操作；
- 注意：RN8611/8613/8615/8318 的 SEG/IO 复用引脚做为 IO 口使用时为开漏结构，RN8612 为普通结构。

GPIO 寄存器基地址：

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

GPIO 寄存器的偏移地址

寄存器名	地址偏移量	描述
PMA	0x00H	PA 口模式寄存器（输入或者输出）
PA	0x04H	PA 口数据寄存器
PCA0	0x08H	PA 口复用寄存器 0
PCA1	0x0CH	PA 口复用寄存器 1
PUA	0x10H	PA 口上拉选择寄存器
PIMA	0x14H	PA 口输入模式配置
PIEA	0x18H	PA 口输入使能选择
PMB	0x1CH	PB 口模式寄存器（输入或者输出）
PB	0x20H	PB 口数据寄存器
PCB	0x24H	PB 口复用寄存器
PUB	0x28H	PUB 口上拉选择寄存器
PIMB	0x2CH	PB 口输入模式配置
PIEB	0x30H	PB 口输入使能选择

PMC	0x34H	PC 口模式寄存器（输入或者输出）
PC	0x38H	PC 口数据寄存器
PCC	0x3CH	PC 口复用寄存器
PUC	0x40H	PUC 口上拉选择寄存器 PCC PC 口复用寄存器
PIEC	0x44H	PC 口输入使能选择
PIMC	0x48H	PC 口输入模式寄存器
PCB2	0X4CH	PB 口复用寄存器 2
PCE	0x60H	SEGC0M 口复用寄存器
PASET	0X64H	PA 口数据置位寄存器，写 1 到该寄存器，PA 口对应位会被写 1；
PACLR	0X68H	PA 口数据清零寄存器，写 1 到该寄存器，PA 口对应位会被清零；
PBSET	0X6CH	PB 口数据置位寄存器，写 1 到该寄存器，PB 口对应位会被写 1；
PBCLR	0X70H	PB 口数据清零寄存器，写 1 到该寄存器，PB 口对应位会被清零；
PCSET	0X74H	PC 口数据置位寄存器，写 1 到该寄存器，PC 口对应位会被写 1；
PCCLR	0X78H	PC 口数据清零寄存器，写 1 到该寄存器，PC 口对应位会被清零；

注：IO 口类型见第 1.4 章管脚排列。

建议使用 bitband 功能（见 4.3.2 章节）访问 GPIO 的寄存器，便于对 IO 口的相关寄存器进行 bit 操作。

也可采用 SET/CLR 寄存器（0x64H~0x78H）写 GPIO 的数据寄存器；

如果 IO 口复用配置选择为 IO 口之外的功能，模式寄存器、数据寄存器、输入使能寄存器无效，上拉选择、输入模式选择在所有复用配置下均有效。

### 10.1.1 PA 口模式寄存器 PMA（输入或者输出）（0x00）

比特位	名称	描述	读/写标志	复位值
31:24	PM37~PM30	=0 输出模式 =1 输入模式 PM37 和 PM36 只读，读出为 1，只能是输入模式；	R/W	FF
23:16	PM27~PM20	=0 输出模式 =1 输入模式	R/W	FF
15:8	PM17~PM10	=0 输出模式 =1 输入模式	R/W	FF
7:5	---	预留	R	0
4:0	PM04~PM00	=0 输出模式	R/W	1F

		=1 输入模式		
--	--	---------	--	--

### 10.1.2 PA 口数据寄存器 PA (0x04)

比特位	名称	描述	读/写标志	复位值
31:30	P37~P36	<b>P36 和 P37 数据输入寄存器，只读；</b>	<b>R</b>	<b>0</b>
29:24	P35~P30	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00
23:16	P27~P20	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00
15:8	P17~P10	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00
7:5	---	预留	R	0
4:0	P04~P00	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值 如果定义为模拟输入，在输入模式读取到的值为 0。	R/W	00

### 10.1.3 PA 口复用 0 寄存器 PCA0 (0x08)

当选择为模拟输入口时，自动选择为输入模式，PMA 寄存器无效。

比特位	名称	描述	读/写标志	复位值
31	---	预留	R	0
30	UART2_SEL	=0: PC245=1 时 P24 和 P25 可选择为 UART2 接口； =1: PC00 和 PC01 有效时可选择为 UART2 接口；	R/W	0
29	SWD_SEL	=0: P24 和 P25 不选择为 SWD，由 PC245 (bit27) 定义； =1: P24 和 P25 选择为 SWD；	R/W	1
28	PC267	定义端口 P26 和 P27 复用配置： =0: 选择为 IO 口； =1: 选择为 UART3 接口。	R/W	00
27	PC245	定义端口 P24 和 P25 复用配置： =0: 选择为 IO 口； =1: 选择为 UART2 接口。	R/W	00
26	PC223	定义端口 P22 和 P23 复用配置： =0: 选择为 IO 口； =1: 选择为 UART1 接口。	R/W	00
25	PC201	定义端口 P20 和 P21 复用配置： =0: 选择为 IO 口； =1: 选择为 UART0 接口。	R/W	00
24:9	PC17[1:0]	分别定义端口 P10~P17 复用配置：	R/W	00

	PC16[1:0] PC15[1:0] PC14[1:0] PC13[1:0] PC12[1:0] PC11[1:0] PC10[1:0]	=00: 选择为 IO 口; =01: 选择为 KEY 输入口 =10: 选择为 TC 输出 =11: 选择为 TC 输入 P17~P10 对应的 TC 输出分别为 { tc1_p[1],tc1_n[1],tc1_p[0],tc1_n[0], tc0_p[1],tc0_n[1], tc0_p[0],tc0_n[0]}		
8	KEY 4 _SEL	=0: P04 的功能由 PC04 位确定; =1: P04 选择为 KEY 4 ( P C 1 4 选择为 K E Y 4 优先级高);	R/W	0
7	---	预留	R	0
6:4	PC04~PC02	定义端口 P04~P02 的复用配置: =0: 选择为 IO 口; =1: 选择为模拟输入口	R/W	0
3:2	PC01	定义端口 P01 复用配置: =00: 选择为 IO 口; =01: 选择为模拟输入口; =10: 选择为 KEY 3 ; ( P C 1 3 选择为 K E Y 3 优先级高) =11: 选择为 TX2;	R/W	0
1:0	PC00	定义端口 P00 复用配置: =00: 选择为 IO 口; =01: 选择为模拟输入口; =10: 选择为 KEY 2 ( P C 1 2 选择为 K E Y 2 优先级高); =11: 选择为 RX2;	R/W	0

#### 10.1.4 PA 口复用 1 寄存器 PCA1 (0x0C)

比特位	名称	描述	读 / 写 标志	复 位 值
31:16	---	预留	R	0
15:14	PC37[1:0]	定义端口 P37 复用配置: =00: 选择为 IO 口; =01: 选择为外部中断输入口 INT7; =1x: 选择为晶体 POSCI 管脚 备注: 只要 PC36[1]和 PC37[1]中任何一位为高, 那么就选择为 POSC	R/W	0
13:12	PC36[1:0]	定义端口 P36 复用配置: =00: 选择为 IO 口; =01: 选择为外部中断输入口 INT6; =1x: 选择为晶体 POSCO 管脚 备注: 只要 PC36[1]和 PC37[1]中任何一位为高, 那么就选择为 POSC	R/W	0
11:10	PC35[1:0]	定义端口 P35 复用配置: =00: 选择为 IO 口; =01: 选择为外部中断输入口 INT5;	R/W	0

		=10: 选择为 TC 输入 =11: 保留		
9:8	PC34[1:0]	定义端口 P34 复用配置: =00: 选择为 IO 口; =01: 选择为外部中断输入口 INT4; =10: 选择为视在电能脉冲输出 SF_OUT =11: 保留	R/W	0
7:6	PC33[1:0]	定义端口 P33 复用配置: =00: 选择为 IO 口; =01: 选择为外部中断输入口 INT3; =10: 选择为 TC 输入 =11: 保留	R/W	0
5:4	PC32[1:0]	定义端口 P32 复用配置: =00: 选择为 IO 口; =01: 选择为外部中断输入口 INT2; =10: 选择为 RTC 输出 RTC_OUT (默认选择为 RTC 输出) =11: 选择 IO 口	R/W	10
3:2	PC31[1:0]	定义端口 P31 复用配置: =00: 选择为 IO 口; =01: 选择为外部中断输入口 INT1; =10: 选择为 TC 输入 =11: 选择为 RX4	R/W	0
1:0	PC30[1:0]	定义端口 P30 复用配置: =00: 选择为 IO 口; =01: 选择为外部中断输入口 INT0; =10: 选择为 TC 输入 =11: 保留	R/W	0

### 10.1.5 PA 口上拉选择寄存器 PUA (0x10)

备注: 当 IO 口处于输出模式或者模拟 PAD 模式时, 不管 PU 寄存器如何配置, PIN 上拉均不使能。

比特位	名称	描述	读/写标志	复位值
31:24	PU37~PU30	定义端口上拉配置: =0: 不选择上拉; =1: 选择上拉;	R/W	00
23:16	PU27~PU20	定义端口上拉配置: =0: 不选择上拉; =1: 选择上拉; 备注: P24 和 P25 作为 SWD 默认上拉使能。	R/W	30
15:8	PU17~PU10	定义端口上拉配置: =0: 不选择上拉; =1: 选择上拉;	R/W	00
7:5	--	保留	R	0

4:0	PU04~PU00	定义端口上拉配置： =0: 不选择上拉； =1: 选择上拉；	R/W	00H
-----	-----------	--------------------------------------	-----	-----

#### 10.1.6 PA 口输入模式配置寄存器 PIMA (0x14)

比特位	名称	描述	读/写标志	复位值
31:24	PIL27~PIL20	定义端口 P20~P27 输入缓冲器类型： =0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$ ; =1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$ ;	R/W	00
23:16	PIL17~PIL10	定义端口 P10~P17 输入缓冲器类型： =0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$ ; =1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$ ;	R/W	00
15:8	PID27~PID20	定义端口 P20~P27 是否是 N-ch 漏极开路输出： =0: 普通模式； =1: N-ch 漏极开路模式；	R/W	00
7:0	PID17~PID10	定义端口 P10~P17 是否是 N-ch 漏极开路输出： =0: 普通模式； =1: N-ch 漏极开路模式；	R/W	00

#### 10.1.7 PA 口输入使能寄存器 PIEA (0x18)

比特位	名称	描述	读/写标志	复位值
31:24	PIE37~PIE30	输入使能： =1: 不使能输入； =0: 使能输入； 注: P30 上电后需要 BOOTROM 设为输入使能, 方便 ISP 进行检测。	R/W	FF
23:16	PIE27~PIE20	输入使能： =1: 不使能输入； =0: 使能输入；	R/W	FF
15:8	PIE17~PIE10	输入使能： =1: 不使能输入； =0: 使能输入；	R/W	FF
7:5	保留	保留	R	0
4:0	PIE04~PIE00	输入使能： =1: 不使能输入； =0: 使能输入；	R/W	3F

#### 10.1.8 PB 口模式寄存器 PMB (输入或者输出) (0x1C)

比特位	名称	描述	读/写标志	复位值
31:24	PM77~PM70	=0 输出模式	R/W	FF

		=1 输入模式		
23:16	PM67~PM60	=0 输出模式 =1 输入模式	R/W	FF
15:8	PM57~PM50	=0 输出模式 =1 输入模式	R/W	FF
7:0	PM47~PM40	=0 输出模式 =1 输入模式	R/W	FF

当 IO 口设置为 7816 口或者 SPI 口时，方向寄存器不起作用，由通信模块本身控制。

### 10.1.9 PB 口数据寄存器 PB (0x20)

比特位	名称	描述	读/写标志	复位值
31:24	P77~P70	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00
23:16	P67~P60	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00
15:8	P57~P50	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00
7:0	P47~P40	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00

### 10.1.10 PB 口复用寄存器 PCB (0x24)

比特位	名称	描述	读/写标志	复位值
31:24	PC77~PC70	PC77~PC70 定义端口复用配置： =0: 选择为 IO 口； =1: 选择 LCD。	R/W	00
23:16	PC67~PC60	PC67~PC60 定义端口复用配置： =0: 选择为 IO 口； =1: 选择 LCD。	R/W	00
15:8	PC57~PC50	PC57~PC50 定义端口复用配置： =0: 选择为 IO 口； =1: 选择为其他，具体定义如下： P50-PF (INT2 引脚输出)、P51-QF (INT0 引脚输出)、P52-SCL、P53-SDA、P54-UART5-RX、P55- UART5-TX、P56-保留、P57-保留 <b>注意：P56 口的 RTCOUT 复用关系由 RTC 章节寄存器决定，与 GPIO 章节复用配置寄存器无关。</b>	R/W	00



		当 VCC 无电, VBAT 上电时, P56 默认输出 1Hz。一旦 VCC 上电, 1Hz 输出会被关闭。外围电路设计需要注意该问题。		
7:0	PC47~PC40	P47~P40 定义端口复用配置: =0: 选择为 IO 口; =1: P40~P43 选择为 7816 P46~47 选择为 SPI。其中 P44/P45 原来复用为 SPI, 此功能保留 (不建议客户使用), P50/P51 增加此复用关系。 默认选择为 GPIO。	R/W	00

### 10.1.11 PB 口复用寄存器 2PCB2 (0x4CH)

比特位	名称	描述	读/写标志	复位值
31:30	PC57_2	P57 端口复用配置 2: =00: P57 口功能由 PC57 寄存器位 (0x24) 决定; =01: P57 选择为 TCIN。 =10: P57 选择为 TC1_P[1]; =11: 保留	R/W	00
29:28	PC56_2	P56 端口复用配置 2: =00: P56 口功能由 PC56 寄存器位 (0x24) 决定; =01: P56 选择为 TCIN。 =10: P56 选择为 TC1_N[1]; =11: 保留 <b>注意: P56 口的 RTCOUT 复用关系由 RTC 章节寄存器决定, 与 GPIO 章节复用配置寄存器无关。</b> 当 VCC 无电, VBAT 上电时, P56 默认输出 1Hz。一旦 VCC 上电, 1Hz 输出会被关闭。外围电路设计需要注意该问题。	R/W	00
27:26	PC55_2	P55 端口复用配置 2: =00: P55 口功能由 PC55 寄存器位 (0x24) 决定; =01: P55 选择为 TCIN。 =10: P55 选择为 TC1_P[0]; =11: 保留	R/W	00
25:24	PC54_2	P54 端口复用配置 2: =00: P54 口功能由 PC54 寄存器位 (0x24) 决定; =01: P54 选择为 TCIN。 =10: P54 选择为 TC1_N[0]; =11: 保留	R/W	00
23:22	PC53_2	P53 端口复用配置 2: =00: P53 口功能由 PC53 寄存器位 (0x24) 决定; =01: P53 选择为 TCIN。 =10: P53 选择为 TC0_P[1]; =11: 保留	R/W	00
21:20	PC52_2	P52 端口复用配置 2:	R/W	00

		=00: P52 口功能由 PC52 寄存器位 (0x24) 决定; =01: P52 选择为 TCIN。 =10: P52 选择为 TC0_N[1]; =11: 保留		
19:18	PC51_2	P51 端口复用配置 2: =00: P51 口功能由 PC51 寄存器位 (0x24) 决定; =01: P51 选择为 RTC_OUT。 <b>=10: P51 选择为 SPI_SCLK;</b> <b>=11: SF2; (INT7 引脚输出)</b>	R/W	00
17:16	PC50_2	P50 端口复用配置 2: =00: P50 口功能由 PC50 寄存器位 (0x24) 决定; =01: P50 选择为 RTC_OUT。 <b>=10: P50 选择为 SPI_SCSN;</b> <b>=11: SF1 (INT6 引脚输出)</b>	R/W	00
15:14	PC47_2	P47 端口复用配置 2: =00: P47 口功能由 PC47 寄存器位 (0x24) 决定; =01: P47 选择为 T X 4。 =1x: 保留	R/W	00
13:12	PC46_2	P46 端口复用配置 2: =00: P46 口功能由 PC46 寄存器位 (0x24) 决定; =01: P46 选择为 R X 4。 =1x: 保留	R/W	00
11:10	PC45_2	P45 端口复用配置 2: =00: P45 口功能由 PC45 寄存器位 (0x24) 决定; =01: P45 选择为 KEY7 (PC17 优先级高)。 <b>=10: AIN6 (如果要使用 AIN6 功能, 还需要配置 ANA_PAD (第 9 章) 寄存器。)</b> <b>=11: 保留</b>	R/W	00
9:8	PC44_2	P44 端口复用配置 2: =00: P44 口功能由 PC44 寄存器位 (0x24) 决定; =01: P44 选择为 KEY6 (PC16 优先级高)。 <b>=10: AIN5 (如果要使用 AIN5 功能, 还需要配置 ANA_PAD (第 9 章) 寄存器。)</b> <b>=11: 保留</b>	R/W	00
7:6	PC43_2	P43 端口复用配置 2: =00: P43 口功能由 PC43 寄存器位 (0x24) 决定; =01: P43 选择为 INT5。(P C 3 5 优先级高) =1x: 保留	R/W	00
5:4	PC42_2	P42 端口复用配置 2: =00: P42 口功能由 PC42 寄存器位 (0x24) 决定; =01: P42 选择为 INT4。(P C 3 4 优先级高) =1x: 保留	R/W	00
3:2	PC41_2	P41 端口复用配置 2:	R/W	00

		=00: P41 口功能由 PC41 寄存器位 (0x24) 决定; =01: P41 选择为 INT3。(P C 3 3 优先级高) =1x: 保留		
1:0	PC40_2	P40 端口复用配置 2: =00: P40 口功能由 PC40 寄存器位 (0x24) 决定; =01: P40 选择为 INT1。(P C 3 1 优先级高) =1x: 保留	R/W	00

### 10.1.12 PB 口上下拉选择寄存器 PUB (0x28)

比特位	名称	描述	读/写标志	复位值
31:24	PU77~PU70	PU77~PU70 定义 P7 端口是否内接下拉: =0: 不接下拉; =1: 内接下拉。	R/W	00
23:16	PU67~PU60	PU67~PU60 定义 P6 端口是否内接下拉: =0: 不接下拉; =1: 内接下拉。	R/W	00
15:8	PU57~PU50	PU57~PU50 定义 P5 端口是否内接上拉: =0: 不接上拉; =1: 内接上拉。	R/W	00
7:0	PU47~PU40	PU47~PU40 定义 P4 端口是否内接上拉: =0: 不接上拉; =1: 内接上拉。	R/W	00

### 10.1.13 PB 口输入模式寄存器 PIMB (0x2C)

比特位	名称	描述	读/写标志	复位值
31:24	PIL57~PIL50	定义端口 P50~P57 输入缓冲器类型: =0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$ ; =1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$ ; 其中 PIL51 和 PIL50 只读位 0;	R/W	00
23:16	PIL47~PIL40	定义端口 P40~P47 输入缓冲器类型: =0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$ ; =1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$ ;	R/W	00
15:8	PID57~PID50	定义端口 P50~P57 是否是 N-ch 漏极开路输出: =0: 普通模式; =1: N-ch 漏极开路模式; 其中 PID51 和 PID50 只读位 0;	R/W	00
7:0	PID47~PID40	定义端口 P40~P47 是否是 N-ch 漏极开路输出: =0: 普通模式; =1: N-ch 漏极开路模式;	R/W	00

## 10.1.14 PB 口输入使能寄存器 PIEB (0x30)

比特位	名称	描述	读/写标志	复位值
31:24	PIE77~PIE70	输入使能： =1: 不使能输入； =0: 使能输入；	R/W	FF
23:16	PIE67~PIE60	输入使能： =1: 不使能输入； =0: 使能输入；	R/W	FF
15:8	PIE57~PIE50	输入使能： =1: 不使能输入； =0: 使能输入；	R/W	FF
7:0	PIE47~PIE40	输入使能： =1: 不使能输入； =0: 使能输入；	R/W	FF

## 10.1.15 PC 口模式寄存器 PMC (输入或者输出) (0x34)

比特位	名称	描述	读/写标志	复位值
31:24	PM117~PM110	=0 输出模式 =1 输入模式	R/W	FF
23:16	PM107~PM100	=0 输出模式 =1 输入模式	R/W	FF
15:8	PM97~PM90	=0 输出模式 =1 输入模式	R/W	FF
7:0	PM87~PM80	=0 输出模式 =1 输入模式	R/W	FF

## 10.1.16 PC 口数据寄存器 PC (0x38)

比特位	名称	描述	读/写标志	复位值
31:24	P117~P110	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00
23:16	P107~P100	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00
15:8	P97~P90	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输	R/W	00

		出模式下读端口，则读取的是输出锁存器的值		
7:0	P87~P80	定义芯片端口需要输出的数据。 如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值	R/W	00

#### 10.1.17 PC 口复用寄存器 PCC (0x3C)

比特位	名称	描述	读/写标志	复位值
31:29	保留	-----	R	0
28	SPI_MUX	SPI_MUX 定义是 P11 口还是 P4 口作为 SPI 接口： =0: 选择 P4 口作为 SPI 接口； =1: 选择 P11 口作为 SPI 接口。	R/W	00
27:24	PC113~PC110	PC113~PC110 定义端口复用配置： =0: 选择为 IO 口； =1: 选择 SPI 接口。 PC110 仅在 SPI 从模式下有效，在 SPI 主模式下不要配置为 1.	R/W	00
23:20	保留	-----	R	0
19:16	PC103~PC100	PC103~PC100 定义端口复用配置： =0: 选择为 IO 口； =1: 选择 LCD。	R/W	0
15:8	PC97~PC90	PC97~PC90 定义端口复用配置： =0: 选择为 IO 口； =1: 选择 LCD。	R/W	00
7:0	PC87~PC80	PC87~PC80 定义端口复用配置： =0: 选择为 IO 口； =1: 选择 LCD。	R/W	00

#### 10.1.18 PC 口上下拉选择寄存器 PUC (0x40)

比特位	名称	描述	读/写标志	复位值
31:24	PU117~PU110	PU113~PU110 定义端口是否内接下拉： =0: 不接上拉； =1: 内接上拉。	R/W	00
23:16	PU107~PU100	PU107~PU100 定义端口是否内接下拉： =0: 不接下拉； =1: 内接下拉。	R/W	00
15:8	PU97~PU90	PU97~PU90 定义端口是否内接下拉： =0: 不接下拉； =1: 内接下拉。	R/W	00
7:0	PU87~PU80	PU87~PU80 定义端口是否内接下拉：	R/W	00

		=0: 不接下拉; =1: 内接下拉。		
--	--	------------------------	--	--

### 10.1.19 PC 口输入使能寄存器 PIEC (0x44)

比特位	名称	描述	读/写标志	复位值
31:24	PIE117~PIE110	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF
23:16	PIE107~PIE100	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF
15:8	PIE97~PIE90	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF
7:0	PIE87~PIE80	输入使能: =1: 不使能输入; =0: 使能输入;	R/W	FF

### 10.1.20 PC 口输入模式寄存器 PIMC (0x48)

比特位	名称	描述	读/写标志	复位值
31:16	---	预留	R	0
15:12	PIL117~PIL114	定义端口 P117~P114 输入缓冲器类型: =0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$ ; =1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$ ;	R/W	00
11:8	PID117~PID114	定义端口 P117~P114 是否是 N-ch 漏极开路输出: =0: 普通模式; =1: N-ch 漏极开路模式;	R/W	00
7:4	PIL113~PIL110	定义端口 P113~P110 输入缓冲器类型: =0: CMOS 缓冲器, $V_{il}=0.3V_{CC}$ $V_{ih}=0.7V_{CC}$ ; =1: TTL 缓冲器, $V_{il}=0.16V_{CC}$ $V_{ih}=0.4V_{CC}$ ;	R/W	00
3:0	PID113~PID110	定义端口 P113~P110 是否是 N-ch 漏极开路输出: =0: 普通模式; =1: N-ch 漏极开路模式;	R/W	00

### 10.1.21 SEGCOM 口复用寄存器 PCE (0x60)

比特位	名称	描述	读/写标志	复位值
31:4	---	预留	R	0
3:0	SEG3/COM	SEG3/COM7~SEG0/COM4 定义端口复用配置:	R/W	00

	7~SEG0/CO M4	=0: 选择为 SEG; =1: 选择为 COM。		
--	-----------------	------------------------------	--	--

**10.1.22 PA 口数据置位寄存器 PASET (0x64)**

比特位	名称	描述	读/写标志	复位值
31:30	保留	-----	R	0
29:24	P35~P30	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
23:16	P27~P20	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
15:8	P17~P10	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
7:5	---	预留	R	0
4:0	P04~P00	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00

Note: 读出值无意义

**10.1.23 PA 口清零置位寄存器 PACLR (0x68)**

比特位	名称	描述	读/写标志	复位值
31:30	保留	-----	R	00
29:24	P35~P30	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	0
23:16	P27~P20	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00
15:8	P17~P10	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00
7:5	---	预留	R	0
4:0	P04~P00	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00

Note: 读出值无意义

**10.1.24 PB 口数据置位寄存器 PBSET (0x6C)**

比特位	名称	描述	读/写标志	复位值
31:24	P77~P70	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
23:16	P67~P60	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
15:8	P57~P50	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
7:0	P47~P00	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00

Note: 读出值无意义

#### 10.1.25 PB 口清零置位寄存器 PBCLR (0x70)

比特位	名称	描述	读/写标志	复位值
31:24	P77~P70	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	0
23:16	P67~P60	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00
15:8	P57~P50	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00
7:0	P47~P40	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00

Note: 读出值无意义

#### 10.1.26 PC 口数据置位寄存器 PCSET (0x74)

比特位	名称	描述	读/写标志	复位值
31:24	P117~P110	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
23:16	P107~P100	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	0
15:8	P97~P90	置位芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
7:0	P87~P80	置位芯片端口状态	R/W	00



		0: 无影响 1: 该端口置位, 输出高电平		
--	--	---------------------------	--	--

Note: 读出值无意义

### 10.1.27 PC 口清零置位寄存器 PCCLR (0x78)

比特位	名称	描述	读/写标志	复位值
31:24	P117~P110	清零芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
23:16	P107~P100	清零芯片端口状态 0: 无影响 1: 该端口置位, 输出高电平	R/W	00
15:8	P97~P90	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00
7:0	P87~P80	清零芯片端口状态 0: 无影响 1: 该端口清零, 输出低电平	R/W	00

Note: 读出值无意义

## 10.2 GPIO 操作步骤

- 1、配置系统控制章节模块使能 1 寄存器 MOD1\_EN 第 5 位为 1 开启 GPIO 模块时钟。
- 2、配置 GPIO 输入输出模式。
- 3、配置 GPIO 口数据寄存器。
- 4、配置 GPIO 口复用功能, 选择 GPIO 复用功能后, GPIO 的输入输出功能将跟随 GPIO 的复用配置。
- 5、当 MCU 使用 5v 供电, 而外围 I2C、SPI 或其他器件工作电压为 3.3V 时, 可选择能配置为 N-ch 漏极开路输出、输入缓冲器类型可选择 TTL 模式的 GPIO。
- 6、当作为输入 IO 口时, 需将输入使能寄存器对应位配置为 0, 打开输入使能。在低功耗模式, 可将 IO 口配置为输入模式, 并关闭输入使能。

## 11 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。

### 11.1 概述

外部中断控制器具备如下特性：

- 支持 8 个外部中断的模式设置：上下边沿及双边沿可设；
- 支持外部中断状态指示；
- 支持外部中断软件触发；
- 支持外部中断状态；
- 支持外部中断屏蔽；
- 支持外部中断滤波，运行模式下滤波时间约为 10 微秒；

### 11.2 寄存器描述

模块寄存器基址

模块名	物理地址	映射地址
INTC	0x40044000	0x40044000

INTC 模块的寄存器偏移地址

寄存器名	地址偏移量	描述
INTC_CTL	0x0	INTC 控制寄存器
INTC_MODE	0x4	INTC 模式寄存器
INTC_MASK	0x8	INTC Mask 寄存器
INTC_STA	0xc	INTC 状态寄存器

#### ● INTC\_CTL

INTC 控制寄存器地址 0x40044000+0x0

比特位	名称	描述	读/写标志	复位值
31:08	---	预留	R	0
7:0	Enable	使能信号，Enable[7:0]分别对应于外部中断请求 7~0，对应的外部管脚为：P37~P30。 0：关闭对应的外部中断 1：使能对应的外部中断	R/W	0

#### ● INTC\_MODE

INTC 模式寄存器地址 0x40044000+0x4

比特位	名称	描述	读/写标志	复位值
31:16	---	预留	R	0
15:14	MODE7	外部中断请求 7（INT7）模式选择 00：上升沿 01：下降沿 10：双边沿 11：保留	R/W	0
13:12	MODE6	外部中断请求 6（INT6）模式选择 00：上升沿	R/W	0

		01: 下降沿 10: 双边沿 11: 保留		
11:10	MODE5	外部中断请求 5 (INT5) 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留	R/W	0
9:8	MODE4	外部中断请求 4 (INT4) 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留	R/W	0
7:6	MODE3	外部中断请求 3 (INT3) 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留	R/W	0
5:4	MODE2	外部中断请求 2 (INT2) 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留	R/W	0
3:2	MODE1	外部中断请求 1 (INT1) 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留	R/W	0
1:0	MODE0	外部中断请求 0 (INT0) 模式选择 00: 上升沿 01: 下降沿 10: 双边沿 11: 保留	R/W	0

● INTC\_MASK

INTC Mask 寄存器地址 0x40044000+0x8

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:0	MASK	MASK[7:0] 分别对应于外部中断请求 7~0 0: 中断禁止 1: 中断使能	R/W	0

● INTC\_STA

INTC 状态寄存器地址 0x40044000+0xc

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

31:08	---	预留	R	0
7:0	STA	STA[7:0]对应于分别对应于外部中断请求 7~0 0: 中断事件未发生 1: 中断事件发生 注: 写 1 清零	R/W	0

## 12 KBI

内置按键接口控制器，用于处理从芯片管脚输入的中断请求，可以在 cpu 休眠时通过中断自动唤醒 cpu。

### 12.1 特性

按键接口控制器具有如下特性：

- 支持 8 个按键，对应管脚为 KEY0~KEY7；
- 支持每个按键状态查询；
- 支持每个按键输入滤波，滤波时间约为 24ms；
- 支持每个按键可单独屏蔽中断

### 12.2 寄存器描述

表 12-1 KBI 寄存器基址

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

表 12-2 KBI 寄存器偏移地址

寄存器名	地址偏移量	描述
KBI_CTL	0x0	控制寄存器
KBI_SEL	0x4	选择寄存器
KBI_DATA	0x8	数据寄存器
KBI_MASK	0xc	屏蔽寄存器

- 控制寄存器（0x0）

表 12-3 KBI 控制寄存器 KBI\_CTL

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:0	EN	使能信号，EN[7:0]对应于 KEY[7:0]， 对应的外部管脚为：P17/KEY7~P10/KEY0。 0：关闭对应的 KEY 1：使能对应的 KEY	R/W	0

- 选择寄存器（0x4）

表 12-4 KBI 选择寄存器 KBI\_SEL

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:0	SEL	SEL[7:0]对应于 KEY[7:0] 0：上升沿有效 1：下降沿有效	R/W	0

- 数据寄存器（0x8）

表 12-5 KBI 数据寄存器 KBI\_DATA

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:0	DAT	DAT[7:0]对应于 KEY[7:0]。写 1 清零 0：按键没有按下 1：按键被按下	R/W	0

- 屏蔽寄存器（0xc）

表 12-6 KBI 屏蔽寄存器 KBI\_MASK

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:0	MASK	MASK[7:0]对应于 KEY[7:0] 0: 中断禁止 1: 中断使能	R/W	0

### 12.3 KBI 操作步骤

- 1、配置系统控制章节 KBI 使能寄存器 KBI\_EN，将第 8 位及对应的 KBI 时钟位配置为 1，开启时钟。
- 2、设置 KBI 控制寄存器 KBI\_CTL，使能对应的 KBI。
- 3、设置 KBI 选择寄存器 KBI\_SEL，配置对应的 KBI 为上升沿或下降沿。
- 4、配置 KBI\_MASK 开启对应中断使能，并打开 KBI 中断 NVIC\_EnableIRQ(KBI\_IRQn);
- 5、编写 KBI 中断服务程序：

```
void KBI_HANDLER(void)
{
    if(KBI->DATA&0x01)
    {

    }
    KBI->DATA = 0xff;
}
```

所有 KBI 中断均为 1 个入口，需根据 KBI\_DATA 判断何 KBI 产生的中断。

- 6、完成

## 13 UART

内置 6 个 UART 接口，用于与外部进行异步串行通信。

### 13.1 概述

UART 接口控制器具备如下特性：

- 六个全双工 UART 接口；
- 内置波特率发生器，支持不同的波特率配置；
- 数据位宽支持 5/6/7/8bit；
- 停止位可配置成 1 或 2bit；
- 可选 38kHz 红外调制；
- 支持自动波特率检测；
- 支持红外唤醒；

### 13.2 寄存器描述

表 13-1 UART 寄存器基址

模块名	物理地址	映射地址
UART0	0x40000000	0x40000000
UART1	0x40004000	0x40004000
UART2	0x40008000	0x40008000
UART3	0x4000C000	0x4000C000
UART4	0x40018000	0x40018000
UART5	0x4001C000	0x4001C000

表 13-2 UART 寄存器偏移地址

寄存器名	地址偏移量	描述
UART_CTL	0x0	UART 控制寄存器
UART_BAUD	0x4	UART 波特率配置寄存器
UART_STAT	0x8	UART 状态指示寄存器
UART_TXD	0xC	UART 发送数据寄存器
UART_RXD	0x10	UART 接收数据寄存器

注：在运行过程中，不能修改控制寄存器与波特率寄存器。

- 控制寄存器（0x00）

表 13-3 控制寄存器 UART\_CTL

比特位	名称	描述	读/写标志	复位值
31:15	---	预留	R	0
14	NEG	UART 极性选择位： 0：正极性，即默认驱动电平为高，发送/接收的数据极性不变 1：负极性，即默认驱动电平为低，发送/接收的数据极性取反	R/W	0
13	LMSB	LSB/MSB 选择方式 0：LSB 先传输 1：MSB 先传输 注：当 PARS 选择为用户自定义校验时，校验位当作数据扩展	R/W	0

		最高位，此时选择 MSB，最先传输的将是校验位		
12	IRSEL	红外调制极性选择： 0: 正极性，即低电平调制输出，高电平（默认状态）保持 1: 负极性，即数据取反，高电平调制输出，低电平保持 注意：IRSEL 只决定空闲输出（无效电平）时的电平高低，不影响有效数据期间的电平	R/W	0
11	ILBE	内部环回使能 0: 内部环回禁止 1: 内部环回使能，TXD 与 RXD 信号在模块内部短接	R/W	0
10	IRE	红外调制使能位 0: 关闭红外调制输出 1: 打开红外调制输出，用 38k 载波调制输出数据的低电平	R/W	0
9:7	PARS	校验位选择 000: 无校验 001: 奇校验 010: 偶校验 011: 固定为零校验 100: 固定为 1 校验 其它: 用户自定义校验	R/W	0
6:5	DATLEN	传输数据宽度位 00: 5-bit 01: 6-bit 10: 7-bit 11: 8-bit	R/W	0
4	STOPS	停止位位宽选择 0: 1-bit 停止位 1: 2-bit 停止位	R/W	0
3	ERRIE	错误中断使能位，对应的标志位是状态指示寄存器中 bit5~bit2。 0: 关闭中断 1: 打开中断	R/W	0
2	RXIE	接收数据中断使能位，对应的标志位是状态指示寄存器中 bit1。 0: 关闭中断 1: 打开中断	R/W	0
1	TXIE	发送数据中断使能位，对应的标志位是状态指示寄存器中 bit0。 0: 关闭中断 1: 打开中断	R/W	0
0	EN	模块使能 0: 关闭 1: 打开	R/W	0

● 波特率配置寄存器（0x4）

表 13-4 波特率配置寄存器 UART\_BAUD

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



31:12	---	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

● 状态指示寄存器 (0x8)

表 13-5 状态指示寄存器 UART\_STA

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
9	tx_fifo_full	发送 FIFO 满： 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 UART 发送 FIFO 已满后继续写 UART 发送寄存器或在发送过程中写入了新的发送数据，该位会置起 0: 无错误 1: 有错误	R/W	0
4	FE	帧错误，写 1 清零 UART 接收到的数据不符合帧格式标志位，如接收到的停止位是 0 而不是 1，该位会置起 0: 无错误 1: 有错误	R/W	0
3	OE	溢出错误，写 1 清零 UART 接收数据寄存器没有及时读取导致接收溢出，该位会置起 0: 无错误 1: 有错误	R/W	0
2	PE	校验错误，写 1 清零 UART 接收到的数据校验和错误，该位会置起 0: 无错误 1: 有错误	R/W	0
1	TX	发送标识，写 1 清零 0: 数据尚未发送或无待发送数据 1: 数据已发送	R/W	0
0	RX	接收标识，写 1 清零 0: 无接收数据	R/W	0

		1: 数据已接收		
--	--	----------	--	--

● 发送数据寄存器 (0xC)

表 13-6 波特率配置寄存器 UART\_TXD

比特位	名称	描述	读/写标志	复位值
31:9	---	预留	R	0
8	UP	用户自定义校验位	R/W	0
7:0	TXDATA	发送数据寄存器	R/W	0

● 接收数据寄存器 (0x10)

表 13-7 接收数据寄存器 UART\_RXD

比特位	名称	描述	读/写标志	复位值
31:9	---	预留	R	0
8	UP	校验位	R	0
7:0	RXDATA	接收数据寄存器	R	0

● 波特率小数分频配置寄存器 (0x14)

表 13-8 接收数据寄存器 UARTx\_FDIV

比特位	名称	描述	读/写标志	复位值
31:14	---	预留	R	0
13:0	FDIV	小数分频系数。计算公式为： $F = \left[ \left( \frac{f_i}{16 \times f_o} - \left\lfloor \frac{f_i}{16 \times f_o} \right\rfloor \right) \times 2^{14} + 0.5 \right]$ 其中 $f_i$ 为输入时钟 (cpu 当前运行时钟), $f_o$ 为输出时钟, “ $\lfloor \ \rfloor$ ” 为向下求整算符。例如输入时钟为 1.8432MHz, 输出时钟为 9837Hz, 那么： $F = \left[ \left( \frac{1843200}{16 \times 9837} - \left\lfloor \frac{1843200}{16 \times 9837} \right\rfloor \right) \times 2^{14} + 0.5 \right]$ 求得 $F=11647$ 。	R/W	0

### 13.3 UART 数据接收及发送操作步骤

- 1、将系统控制章节模块使能 0 寄存器 MOD0\_EN 中对应的 UART 时钟启动, 模块使能。
- 2、配置波特率配置寄存器 UARTx\_BAUD, 例如当系统时钟为 3.6864MHz 时, 通讯波特率为 9600, 则波特率配置寄存器可设置为:  $UART0->BAUD = 3686400 / (9600 * 16) - 1$ ;
- 3、配置通讯控制寄存器 UARTx\_CTL, 选择数据位、停止位、校验方式及中断使能。
- 4、写 0x3f 清除 UART 状态指示寄存器 (UARTx\_STA) 状态。
- 5、配置 UART 中断使能, 打开 UART 中断 NVIC\_EnableIRQ(UARTx\_IRQn);
- 6、编写中断服务程序, 如 UART0 中断服务程序为:

```
void UART0_HANDLER(void)
{
    u32  status;
    u8   temp;
    status = UART0->STA;
```

```
        /* UART error irq */
        if((UART0->CTRL & 0x8) && (status & 0x3c))
        {
            /* Start adding user code. Do not edit comment generated here */
        }
        /* receive data complete irq */
if((UART0->CTRL & 0x4) && (status & 0x1))
        {
/* Start adding user code. Do not edit comment generated here */
        }
        /* transmit data complete irq */
        if((UART0->CTRL & 0x2) && (status & 0x2))
        {
            /* Start adding user code. Do not edit comment generated here */
        }
    }
}
```

UART 接收、发送、错误中断为同一中断入口，需根据控制寄存器打开的中断使能位及状态标志判断此时为何种中断。

- 7、处理接收或发送的数据，完成。
- 8、注意：UART 口为全双工方式，可同时进行发送及接收，当使用 RS485 半双工通讯方式时，RS485 芯片在发送时，接收端会有干扰信号，此时建议在发送时关闭 MCU 的接收中断，接收时关闭发送中断，消除干扰。

## 14 ISO7816

内置两个 ISO7816 通道，支持外接 2 个 7816 协议接口设备。

### 14.1 概述

ISO7816 接口控制器具备如下特性：

- 支持标准的 ISO7816 协议，工作在主模式；
- 支持卡时钟输出，频率可在 1~5MHz 之间设置；
- 支持 7816 多种分频比设置；
- 支持 MSB 先输出的低逻辑和 LSB 先输出的高逻辑数据编码方式；
- 支持 1, 2ETU 宽度的错误信号宽度设置；
- 支持 0~254ETU 宽度的 EGT 设置；
- 支持发送数据传输错误重发机制，重发次数可在 0~7 之间设置；
- 7816 卡协议栈支持接口两个 (Esam 和卡)： esam 模块接收和发送用一个管脚口；
- 支持卡接口接收和发送分离；

### 14.2 寄存器描述

表 14-1 ISO7816 寄存器基址

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

表 14-2 ISO7816 寄存器偏移地址

寄存器名	地址偏移量	描述
ISO7816_CTL0	0x0	控制寄存器 0
ISO7816_CTL1	0x4	控制寄存器 1
ISO7816_CLK	0x8	时钟配置寄存器
ISO7816_BDDIV0	0xc	波特率配置寄存器 0
ISO7816_BDDIV1	0x10	波特率配置寄存器 1
ISO7816_STAT0	0x14	状态指示寄存器 0
ISO7816_STAT1	0x18	状态指示寄存器 1
ISO7816_DAT0	0x1c	数据发送寄存器 0
ISO7816_DAT1	0x20	数据发送寄存器 1

- 控制寄存器 (0x0)

表 14-3 ISO7816 控制寄存器 0 ISO7816\_CTL0

比特位	名称	描述	读/写标志	复位值
31:28	---	预留	R	0
27	RX_GT0	接收数据 GT 选择位，发送时固定为 2etu 1: 接收数据 GT 为 1etu 0: 接收数据 GT 为 2etu	R/W	0
26	---	预留	R/W	0
25	---	预留	R/W	0
24:17	EGT0	EGT 宽度选择值(0~255)，即额外保护时间 N，默认值 N=0。 在0到254范围内，N用于计算两个相邻数据起始沿之间的延迟： $12 \text{ etu} + (Q \times (N/f))$ 。	R/W	0

		<p>公式中，Q应取下面两个值中的一个：</p> <p>——当复位应答中不存在T=15时，取F/D；</p> <p>——当复位应答中存在T=15时，取 Fi/Di；</p> <p>N=255表示在传输协议期间，两个连续字符的起始沿之间的最小延迟在传输的两个方向上是相同的。这个最小延迟值是：</p> <p>——T=0时，12etu</p> <p>——T=1时，11etu</p>		
16:14	REP_CNT0	<p>数据奇偶校验出错时自动重发次数控制</p> <p>000: 0次 001: 1次</p> <p>010: 2次 011: 3次</p> <p>100: 4次 101: 5次</p> <p>110: 6次 111: 7次</p>	R/W	011
13	RXPAR_ESEL0	<p>接收数据奇偶校验错误处理方式选择</p> <p>1: 奇偶校验错，根据 T=0 协议回发 error signal。置 RX_PAR_ERR 标志，进行中断。</p> <p>0: 奇偶校验错，不发送 error signal，置 RX_PAR_ERR 标志，直接中断。</p>	R/W	1
12:11	ERRWTH0	<p>错误信号宽度选择位，只适用于接收，且 RXPAR_ESEL0=1</p> <p>00: 2 etu</p> <p>01: 1 etu</p> <p>10: 1.5 etu</p> <p>11: 2 etu</p>	R/W	01
10:8	PARSEL0	<p>校验位选择位</p> <p>000: 无校验</p> <p>001: 奇校验</p> <p>010: 偶校验</p> <p>011: 固定为零校验</p> <p>100: 固定为1校验</p> <p>其它: 保留</p>	R/W	010
7	BGT_EN0	<p>数据接收到发送的 BGT 控制位</p> <p>0: 关闭 BGT 功能，数据接收发送之间不插入 BGT</p> <p>1: 打开 BGT 功能，数据接收发送之间插入 BGT (22etu)</p>	R/W	0
6	ERR_IRQ_EN0	<p>传输出错中断使能位，发送数据时数据冲突、接收数据时数据溢出以及接收数据帧格式错误</p> <p>0: 禁止传输出错产生中断</p> <p>1: 使能传输出错产生中断</p>	R/W	0
5	RX_IRQ_EN0	<p>数据接收中断使能位，使能数据从移位寄存器移入到接收缓存寄存器产生中断</p> <p>0: 禁止数据接收产生中断</p> <p>1: 使能数据接收产生中断</p>	R/W	0
4	TX_IRQ_EN0	<p>数据发送中断使能位，使能数据从移位寄存器发送完</p>	R/W	0

		成产生中断 0: 禁止数据发送产生中断 1: 使能数据发送产生中断		
3	RX_EN0	接收数据使能 0: 禁止数据接收 1: 使能数据接收	R/W	0
2	TX_EN0	发送数据使能 0: 禁止数据发送 1: 使能数据发送	R/W	0
1	DIRSEL0	数据编码方式选择位 0: LSB 先传的正逻辑数据编码方式 1: MSB 先传的负逻辑数据编码方式（数据取反）	R/W	0
0	EN0	ISO7816 控制器使能位 0: 控制器关闭 1: 控制器打开	R/W	0

● ISO7816 控制寄存器 1 (0x04)

表 14-4 ISO7816 控制寄存器 1 ISO7816\_CTL1

比特位	名称	描述	读/写标志	复位值
31	CARD1_CHECK_EN	卡拔出检测使能位，只在 OLD 检测功能使能后该位有效 1: 使能卡拔出检测中断功能 0: 不使能卡拔出检测中断功能	R/W	0
30	OLD1_IRQ_EN	OLD 检测中断功能使能，只在 OLD 检测功能使能后该位有效 1: 使能 OLD 检测中断功能 0: 不使能 OLD 检测中断功能	R/W	0
29	OLD1_EN	OLD 检测功能使能 1: 使能 OLD 检测功能 0: 不使能 OLD 检测功能	R/W	0
28	RX1_GT0	接收数据 GT 选择位，发送时固定为 2etu 1: 接收数据 GT 为 1etu 0: 接收数据 GT 为 2etu	R/W	0
27	---	预留	R/W	0
26	---	预留	R/W	0
25	IO1_EN	数据为双向使能信号 1: 78161_IO 端口为双向信号 0: 78161_IO 端口为单向信号，只输出，数据输入从 78161_I 端口输入。	R/W	1
24:17	EGT1	EGT 宽度选择值(0~255)，额外保护时间 N 默认值N=0。 在0到254范围内，N表示在准备好接收下一字符之前，卡要求从前一个字符（也是由卡或接口	R/W	0

		设备发送的) 的起始沿开始的下列延迟: $12 \text{ etu} + (Q \times (N/f))$ 公式中, Q应取下面两个值中的一个: F/D, 即用于计算etu的值, 当T=15不存在于复位应答中时, Fi/Di, 当T=15存在于复位应答中时。 N=255表示在传输协议期间, 两个连续字符的起始沿之间的最小延迟在传输的两个方向上是相同的。这个最小延迟值是 T=0时, 12etu T=1 时, 11etu		
16: 14	REP_CNT1	数据奇偶校验出错时自动重发次数控制 000: 0次    001: 1次 010: 2次    011: 3次 100: 4次    101: 5次 110: 6次    111: 7次	R/W	011
13	RXPAR_ESEL1	接收数据奇偶校验错误处理方式选择 1: 奇偶校验错, 根据 T=0 协议回发 error signal。 置 RX_PAR_ERR 标志, 进行中断。 0: 奇偶校验错, 不发送 error signal, 置 RX_PAR_ERR 标志, 直接中断。	R/W	1
12:11	ERRWTH1	错误信号宽度选择位 00: 2 etu 01: 1 etu 10: 1.5 etu 11: 2 etu	R/W	01
10:8	PARSEL1	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: 保留	R/W	010
7	BGT_EN1	数据接收到发送的 BGT 控制位 0: 关闭 BGT 功能, 数据接收发送之间不插入 BGT 1: 打开 BGT 功能, 数据接收发送之间插入 BGT	R/W	0
6	ERR_IRQ_EN1	传输出错中断使能位, 发送数据时数据冲突、接收数据时数据溢出以及接收数据帧格式错误 0: 禁止传输出错产生中断 1: 使能传输出错产生中断	R/W	0
5	RX_IRQ_EN1	数据接收中断使能位, 使能数据从移位寄存器移入到接收缓存寄存器产生中断 0: 禁止数据接收产生中断 1: 使能数据接收产生中断	R/W	0

4	TX_IRQ_EN1	数据发送中断使能位，使能数据从发送缓存寄存器移入到移位寄存器产生中断 0: 禁止数据发送产生中断 1: 使能数据发送产生中断	R/W	0
3	RX_EN1	接收数据使能 0: 禁止数据接收 1: 使能数据接收	R/W	0
2	TX_EN1	发送数据使能 0: 禁止数据发送 1: 使能数据发送	R/W	0
1	DIRSEL1	数据编码方式选择位 0: LSB 先传的正逻辑数据编码方式 1: MSB 先传的负逻辑数据编码方式	R/W	0
0	EN1	ISO7816 控制器使能位 0: 控制器关闭 1: 控制器打开	R/W	0

● ISO7816\_CLK (0x08)

表 14-5 ISO7816 时钟控制寄存器 1 ISO7816\_CLK

比特位	名称	描述	读/写标志	复位值
31:4	---	预留	R	0
3	CLKO_EN	卡时钟输出使能位 0: 关闭卡时钟输出 1: 使能卡时钟输出	R/W	0
2:0	CLKDIV	ISO7816 时钟输出 CLK_O 的分频系数 ISO7816 模块的源时钟从系统时钟 fsyspll 上直接取得 000: 不分频; 001: 2 分频; 010: 4 分频; 011: 8 分频; 100: 16 分频; 101: 不支持; 110: 不支持; 111: 不支持;	R/W	0

● ISO7816 波特率系数 0 寄存器 (0x0c)

表 14-6 ISO7816 波特率系数 0 寄存器 ISO7816\_BDDIV0

比特位	名称	描述	读/写标志	复位值
31:22	---	预留	R	0
21	FDS0_EN	使能软件配置 F/D 的系数。 1: 波特率系数通过软件写入 FDS0 来确定 0: 波特率系数通过 FD0 来确定	R/W	0
20:8	FDS0	软件配置的波特率系数值，该位只在 FDS0_EN=1 时可写，其它情况下均为 13'd372。	R/W	13'd372
7:0	FD0	复位应答所传送的 8 位 FI 和 DI	R/W	8'h01

● ISO7816 波特率系数 1 寄存器 (0x10)

表 14-7 ISO7816 波特率系数 1 寄存器 ISO7816\_BDDIV1

比特位	名称	描述	读/写标志	复位值
31:22	---	预留	R	0



21	FDS0_EN	使能软件配置 F/D 的系数。 1: 波特率系数通过软件写入 FDS0 来确定 0: 波特率系数通过 FD0 来确定	R/W	0
20:8	FDS0	软件配置波特率系数, 只在 FDS0_EN=1 时可写, 其它情况下均为 372	R/W	13'd372
7:0	FD0	复位应答所传送的 8 位 FI 和 DI	R/W	8'h01

● ISO7816 状态 0 寄存器 (0x14)

表 14-8 ISO7816 状态 0 寄存器 ISO7816\_STAT0

比特位	名称	描述	读/写标志	复位值
31:12	---	预留	R	0
11	FRAME_ERR0	接收数据帧格式错误中断标志位, 该位写 1 清零, 1: 发送接收数据帧格式错误, 当传输错误中断使能位有效时产生中断 0: 未发送接收数据帧格式错误	R/W	0
10	BDDIV_R0	波特率匹配为指示。FI 和 DI 是否匹配指示; FD 默认为 8'h01, 时钟匹配, 当写入的 FD 值不匹配时该位置 1。 1: 匹配 0: 不匹配	R	1
9	TX_FLAG0	发送缓冲区空标志。上电复位后自动置位, 表示缓冲区空, 可以写入数据。MCU 写入数据后标志自动清除, 数据从发送缓冲寄存器移入移位寄存器后置 1。 1: 数据发送缓冲区空 0: 数据发送缓冲区内有数据待发送	R	1
8	RX_FLAG0	数据缓冲区满标志, 7816 接口控制器每收到 1byte 数据, 硬件自动置位, 表示接收到 1byte 数据, 读数据接收缓冲寄存器清零。 1: 接收到 1byte 数据, 数据接收缓冲区满 0: 未接收到数据, 数据接收缓冲区空	R	0
7	RXBUSY0	数据接收忙标志位。硬件置位, 软件清零 硬件自动清置位 0: 数据接收空闲 1: 接收移位寄存器正在接收数据, 收到起始位后自动置 1, 收到停止位后自动清零	R	0
6	TXBUSY0	数据发送忙标志。硬件置位, 软件清零 硬件自动清置位 0: 数据发送空闲 1: 发送移位寄存器正在发送数据, 发送起始位时置 1, 发送停止位时自动清零	R	0
5	TXPAR_ERR0	发送数据奇偶校验错误标志位, 重发次数到后仍然奇偶校验错误则该位置位。 对该位写 1 将清零 1: 发送数据时发生奇偶校验错误 0: 发送数据时无奇偶校验错误	R/W	0

4	RXPAR_ER RIF0	接收数据奇偶校验错误标志位，重发次数到后仍然奇偶校验错误则该位置位。 对该位写 1 将清零 1: 接收数据时发生奇偶校验错误 0: 接收数据时无奇偶校验错误	R/W	0
3	COL_IF0	发送数据冲突错误中断标志位。硬件置位，软件清零 对该位写 1 将清零 0: 没有中断 1: 中断发生	R/W	0
2	OVL_IF0	接收数据溢出错误标志位。硬件置位，软件清零 对该位写 1 将清零 0: 无溢出错误 1: 中断发生，接收缓冲寄存器未被读出，又接收到新的数据，溢出错误标志有效	R/W	0
1	RXIF0	数据接收中断标志位。数据从移位寄存器移入接收缓存寄存器后置 1。硬件置位，软件清零 对该位写 1 将清零 0: 没有中断 1: 中断发生	R/W	0
0	TXIF0	数据发送中断标志位。数据从发送缓冲寄存器移入移位寄存器后置 1。 硬件置位，软件清零 对该位写 1 将清零 0: 没有中断 1: 中断发生	R/W	0

● ISO7816 状态 1 寄存器 (0x18)

表 14-9 ISO7816 状态 1 寄存器 ISO7816\_STAT1

比特位	名称	描述	读/写标志	复位值
31:14	---	预留	R	0
13	CARD_OUT_FLAG	CARD_CHECK_EN 使能后该位有效，该位写 1 清零。 1: 检测到卡拔出（检测到输入端口的高电平脉宽大于 40mS） 0: 未测到卡拔出（检测到输入端口的高电平脉宽不大于 40mS）	R	0
12	OLD_FLAG	OLD_EN 使能后该位有效，为配合 RA9105 的接收到 OLD 信号的中断标志位，该位写 1 清零。 1: 接收到 OLD 信号。 0: 未接收到 OLD 信号。	R/W	0
11	FRAME_ERROR0	接收数据帧格式错误中断标志位，该位写 1 清零 1: 发送接收数据帧格式错误，当传输错误中断使能位有效时产生中断 0: 未发送接收数据帧格式错误	R/W	0
10	BDDIV_R1	波特率匹配为指示。FI 和 DI 是否匹配指示；FD 默认为	R	1

		8'h01, 时钟匹配, 当写入的 FD 值不匹配时该位置 1。 1: 匹配 0: 不匹配		
9	TX_FLAG1	发送缓冲区空标志。上电复位后自动置位, 表示缓冲区空, 可以写入数据。MCU 写入数据后标志自动清除, 数据从发送缓冲寄存器移入移位寄存器后置 1。 1: 数据发送缓冲区空 0: 数据发送缓冲区内有数据待发送	R	1
8	RX_FLAG1	数据接收完成标志, 7816 接口控制器每收到 1byte 数据, 根据接收的通道相应发出一次中断。硬件置位, 读数据接收缓冲寄存器清零。 1: 接收到 1byte 数据, 数据接收缓冲区满 0: 未接收到数据, 数据接收缓冲区空	R	0
7	RXBUSY1	数据接收忙标志位。硬件置位, 软件清零 硬件自动清置位 0: 数据接收空闲 1: 接收移位寄存器正在接收数据, 收到起始位后自动置 1, 收到停止位后自动清零	R	0
6	TXBUSY1	数据发送忙标志。硬件置位, 软件清零 硬件自动清置位 0: 数据发送空闲 1: 发送移位寄存器正在发送数据, 发送起始位时置 1, 发送停止位时自动清零	R	0
5	TXPAR_ER RIF1	发送数据奇偶校验错误标志位。硬件置位, 软件清零 对该位写 1 将清零 1: 发送数据时发生奇偶校验错误 0: 发送数据时无奇偶校验错误	R	0
4	RXPAR_ER RIF1	接收数据奇偶校验错误标志位。硬件置位, 软件清零 对该位写 1 将清零 1: 接收数据时发生奇偶校验错误 0: 接收数据时无奇偶校验错误	R/W	0
3	COL_IF1	发送数据冲突错误中断标志位。硬件置位, 软件清零 对该位写 1 将清零 0: 没有中断 1: 中断发生	R/W	0
2	OVL_IF1	接收数据溢出错误标志位。硬件置位, 软件清零 对该位写 1 将清零 0: 无溢出错误 1: 中断发生, 接收缓冲寄存器未被读出, 又接收到新的数据, 溢出错误标志有效	R/W	0
1	RXIF1	数据接收中断标志位。数据从移位寄存器移入接收缓存寄存器后置 1。硬件置位, 软件清零 对该位写 1 将清零	R/W	0

		0: 没有中断 1: 中断发生		
0	TXIF1	数据发送中断标志位。数据从发送缓冲寄存器移入移位寄存器后置 1。 硬件置位，软件清零 对该位写 1 将清零 0: 没有中断 1: 中断发生	R/W	0

● ISO7816 数据 0 寄存器 (0x1C)

表 14-10 ISO7816 数据 0 寄存器 ISO7816\_DAT0

比特位	名称	描述	读/写标志	复位值
保留	--	--	R	0
8	DATA0[8]	当 <code>parsel</code> 为用户自定义模式时，为数据帧中的 PARITY 位	R/W	0
7:0	DAT0	数据寄存器 0	R/W	0

● ISO7816 数据 1 寄存器 (0x20)

表 14-11 ISO7816 数据 1 寄存器 ISO7816\_DAT1

比特位	名称	描述	读/写标志	复位值
保留	--	--	R	0
8	DATA1[8]	当 <code>parsel</code> 为用户自定义模式时，为数据帧中的 PARITY 位	R/W	0
7:0	DAT1	数据寄存器 1	R/W	0

### 14.3 7816 与 ESAM 通讯操作步骤

- 1、7816 与 ESAM 通讯，不需考虑隔离问题，数据 IO 可共用 1 根线，建议使用 7816 模块 0。
- 2、配置系统控制章节模块使能 0 寄存器 `MOD0_EN`，将第 13 位置为 1，开启 7816 时钟。
- 3、配置 7816 控制寄存器为 `ISO7816_CTL0`，使用国网 ESAM，寄存器可配置为 `0x00000201`。
- 4、清除 `ISO7816` 状态寄存器 `ISO7816_STAT0`。
- 5、打开 7816 总线时钟，例如当系统时钟为 3.6864MHZ 时，`ISO7816 ->CLK=0x09`；此时 7816 模块时钟为 1.8432MHZ。
- 5、可通过中断方式或查询状态方式向 7816 总线读写数据。

### 14.4 7816 与卡通讯操作步骤

- 1、对于卡表，卡需与主电进行隔离，要提供 1-5MHZ 时钟方能正常工作，现有 SOC 大部分使用高速光耦方案进行主电与卡的隔离，此方案成本较高，并且使用高速光耦隔离高频时钟，在高低温下的可靠性较弱。我们提供了一个专用芯片 `RN8501`，用于与卡的连接，此芯片使用 2 个普通光耦与 MCU 进行数据通讯。
- 2、配置系统控制章节模块使能 0 寄存器 `MOD0_EN`，将第 13 位置为 1，开启 7816 时钟。
- 3、配置 7816 控制寄存器为 `ISO7816_CTL1`，因与卡通讯使用光耦隔离，接收与发送分开，`ISO7816->CTRL1` 可配置为 `0x60000201`。
- 4、清除 `ISO7816` 状态寄存器 `ISO7816_STAT0`。
- 5、打开 7816 总线时钟，例如当系统时钟为 3.6864MHZ 时，`ISO7816 ->CLK=0x09`；此时 7816 模块时钟为 1.8432MHZ。

6、可通过中断方式或查询状态方式向 7816 总线读写数据。

使用 RN8501 后卡的插入及拔出检测及卡的复位信息读取会与分离方案有所区别：

- 1、卡的插入检测：将卡座的检测脚连接到 RN8501 的 CHK，当卡插入卡座时，CHK 引脚为低电平，RN8501 通过与 RN821x 连接的 7816 口发送 9MS 左右的低电平信号，当 ISO7816\_CTL1 的 OLD 检测使能打开后，就会产生中断，认为外部有卡插入。
- 2、卡的拔出检测：配置 ISO7816\_CTL1 的卡拔出检测使能位后，可检测卡的拔出。注意：卡操作完成后方能开启卡拔出检测中断。
- 3、卡的复位信息的读取在 MCU 与 RN8501 通讯握手完成后读取。
- 4、更具体操作步骤可参考 RN8501 数据手册。

## 15 IIC 接口

内置一个 I<sup>2</sup>C 接口控制器。

### 15.1 概述

I<sup>2</sup>C 接口控制器具备如下特性：

- 支持主模式和从模式；
- 支持 7-bit 地址；
- 支持多种分频比设置；
- 支持 100kbps 和快速模式 400kbps；

### 15.2 寄存器描述

表 15-1 I<sup>2</sup>C 寄存器基址

模块名	物理地址	映射地址
I <sup>2</sup> C	0x40024000	0x40024000

表 15-2 I<sup>2</sup>C 寄存器偏移地址

寄存器名	地址偏移量	描述
I <sup>2</sup> C_CTL	0x0	控制寄存器
I <sup>2</sup> C_CLK	0x4	时钟配置寄存器
I <sup>2</sup> C_STAT	0x8	状态指示寄存器
I <sup>2</sup> C_ADDR	0xC	从设备地址寄存器
I <sup>2</sup> C_DATA	0x10	收发数据寄存器

- 控制寄存器 (0x0)

表 15-3 控制寄存器 I<sup>2</sup>C\_CTL

比特位	名称	描述	读/写标志	复位值
31:6	---	预留	R	0
5	MODE	模式选择位 1: 主机模式 0: 从机模式	R/W	0
4	ACK	ACK 发送使能 1: 接收到第九个 SCL 的时候, 产生 ACK 0: 接收到第九个 SCL 的时候, 不产生 ACK	R/W	0
3	IRQE	I <sup>2</sup> C 中断使能 0: 禁止中断 1: 使能中断	R/W	0
2:1	BUSCON	总线控制产生位, 开始命令在总线为空闲状态或者主机是发送状态时有效。结束命令在主机是发送状态时有效。 当检测到 start 或者 stop 时序时, 对命令位清零, 00: 没有动作 01: 产生 START 时序 10: 产生 STOP 时序 11: 保留	R/W	0

0	EN	模块使能 1: I <sup>2</sup> C 打开 0: I <sup>2</sup> C 关闭	R/W	0
---	----	--	-----	---

● 时钟配置寄存器 (0x4)

表 15-4 时钟配置寄存器 I<sup>2</sup>C\_CLK

比 特 位	名称	描述	读 / 写 标 志	复位值																					
31:3	---	预留	R	0																					
2:0	CLKDIV	<p>I<sup>2</sup>C 时钟分频参数选择位： I<sup>2</sup>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<sup>2</sup>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。</p> <table border="1"> <thead> <tr> <th>系统主频</th> <th colspan="2">分频参数 /CLKDIV(m)</th> </tr> <tr> <td></td> <th>高速模式</th> <th>正常模式</th> </tr> </thead> <tbody> <tr> <td>1.8432Mhz</td> <td>不支持</td> <td>010 (20)</td> </tr> <tr> <td>3.6864Mhz</td> <td>001 (10)</td> <td>011 (38)</td> </tr> <tr> <td>7.3728Mhz</td> <td>010 (20)</td> <td>100 (76)</td> </tr> <tr> <td>14.7456Mhz</td> <td>011 (38)</td> <td>101 (152)</td> </tr> <tr> <td>29.4912Mhz</td> <td>100 (76)</td> <td>110 (304)</td> </tr> </tbody> </table>	系统主频	分频参数 /CLKDIV(m)			高速模式	正常模式	1.8432Mhz	不支持	010 (20)	3.6864Mhz	001 (10)	011 (38)	7.3728Mhz	010 (20)	100 (76)	14.7456Mhz	011 (38)	101 (152)	29.4912Mhz	100 (76)	110 (304)	R/W	001
系统主频	分频参数 /CLKDIV(m)																								
	高速模式	正常模式																							
1.8432Mhz	不支持	010 (20)																							
3.6864Mhz	001 (10)	011 (38)																							
7.3728Mhz	010 (20)	100 (76)																							
14.7456Mhz	011 (38)	101 (152)																							
29.4912Mhz	100 (76)	110 (304)																							

● 状态指示寄存器 (0x8)

表 15-5 状态指示寄存器 I<sup>2</sup>C\_STAT

比 特 位	名称	描述	读 / 写 标 志	复 位 值
31:9	---	预留	R	0
8	DIR	读写方向标志 1: 读。 0: 写。	R	0
7	MATCH	地址匹配，检测到 start 或者 stop 时序后会清零 0: 地址不匹配 1: 地址匹配	R	0
6	BUSY	通讯状态标志 0: IIC 处于空闲状态 1: IIC 处于正常通讯状态	R	0
5	COL	发送冲突中断标志。写 1 清零 发送数据寄存器不为空或接收数据时，用户向数据寄存器写新的数据，将触发发送冲突中断标志。 0: 没有触发发送冲突中断 1: 触发发送冲突中断	R/W	0
4	OVERF	接收溢出中断标志。写 1 清零 接收数据时，当上一个接收数据未被取走前又收到新的数据，将触发溢出中断标志	R/W	0

		0: 没有触发溢出中断 1: 触发溢出中断		
3	TXEMPT	发送数据寄存器空错误标志。写 1 清零 从模式下, 主机要求从机发送数据, 但发送缓冲区为空时, 触发发送数据寄存器空错误中断标志 0: 没有发生发送数据寄存器空错误 1: 触发发送数据寄存器空错误中断	R/W	0
2	TRANC	传输完成中断标志。写 1 清零 发送数据时发送缓存为空或者接收数据时接收缓存满, 触发传输完成中断标志 0: 传输未完成 1: 传输已完成	R/W	0
1	RX_NACK	收到 NACK 中断标志。写 1 清零 1: 收到 nack 0: 没有接收到 nack	R/W	0
0	STPD	STOP 时序检测中断标志。写 1 清零 关闭模块或者接收到 START 时序后, 该位自动清零 0: 没有检测到 STOP 时序 1: 检测到 STOP 时序	R/W	0

● 从设备地址寄存器 (0xC)

表 15-6 从设备地址寄存器 I<sup>2</sup>C\_ADDR

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:1	SADR	设备地址, 在传输地址期间不可写 主机模式时, 表示从设备的地址; 从机模式时, 该地址用来与主机发来的地址进行比较	R/W	0
0	RW	主机读写方向控制位 0: 写 1: 读	R/W	0

● 收发数据寄存器 (0x10)

表 15-7 收发数据寄存器 I<sup>2</sup>C\_DATA

比特位	名称	描述	读/写标志	复位值
31:8	---	预留	R	0
7:0	TRDATA	接收/发送数据	R/W	0



## 16 SPI 接口

内置一个 SPI 接口。

### 16.1 概述

SPI 接口控制器具备如下特性：

- 支持 SPI 全双工模式；
- 支持主模式和从模式工作；
- 支持时钟的极性和相位设定；
- 支持发送和接收独立双缓冲区；
- 支持 LSB 和 MSB 传输模式 8 位，16 位，32 位可配置；
- 支持 256 种波特率可设，最高 3.6864MHz；
- 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；

### 16.2 功能描述

SPI 接口符合标准的 SPI HOST 协议，SPI 时钟工作方式通过 CPOL(Clock Polarity)和 CPHA(Clock Phase)参数设置：CPOL 决定时钟的前边沿是上升沿还是下降沿，CPHA 决定时钟的前边沿是数据采样还是数据建立。

详细的工作模式如下表：

表 16-1 SPI 时钟工作方式

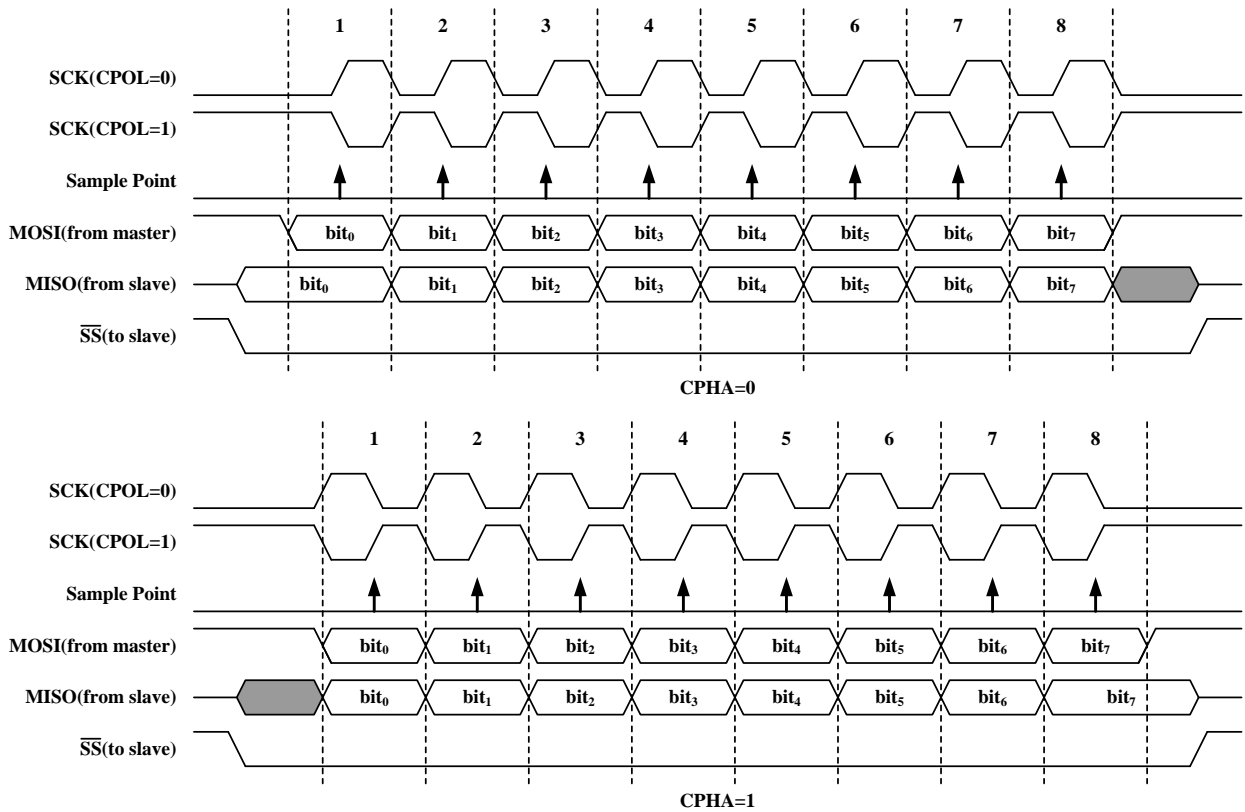
SPI 模式	CPOL/CPHA	前边沿	后边沿
0	0/0	上升沿，数据采样	下降沿，数据建立
1	0/1	上升沿，数据建立	下降沿，数据采样
2	1/0	下降沿，数据采样	上升沿，数据建立
3	1/1	下降沿，数据建立	上升沿，数据采样

数据传输大小支持 8/16/32bit 宽度，SPI 时钟源来自系统时钟，经过一个分频系数后产生通讯时钟。

支持数据发送冲突中断、数据接收溢出中断、传输结束中断和 SS 模式出错中断等四种中断。

发送数据冲突，当一次数据发送正在进行中 (txbusy 为 1)，此时总线又有一次写命令，则 TXCOLIF 置 1，若 COL\_IRQ\_EN=1，则会产生中断，同时该发送命令不会响应，正在发送的数据会正常传输完成。

图 16-1 SPI 时钟工作方式



接收数据溢出：在下次完整的接收数据进入移位寄存器之前，没有读取 RXDATA 寄存器，将产生接收数据溢出，则 RXCOLIF 置 1，若 COL\_IRQ\_EN=1，则会产生中断，同时新的接收数据会保存到接收数据寄存器中，原来没被读走的数据将被覆盖。

传输结束中断：当传输结束时 (sck\_end)，若 TR\_IRQ\_EN=1，则会产生一个中断，同时 TRIF 置 1。

SCSN 模式出错中断：从模式下，SCSN 必须作为输入，在数据传输过程中 SCSN 变高，则 SCSN 模式出错标志置 1；在主模式下，只有使能主模式 SCSN 模式错误检测 (SCSN\_EN=1)，同时 SCSN 输入为低，则 SCSN 模式出错标志置 1。一旦 SCSN 模式出错标志为 1，则会终止正在进行的传输同时复位 SPI 模块，如果 ERR\_IRQ\_EN=1 还会产生一个中断。

**注意：在配置完成控制寄存器后，做为主模式时，只有往数据发送寄存器写数据，才会启动 SPI 读写操作。**

### 16.3 寄存器描述

表 16-2 SPI 寄存器基址

模块名	物理地址	映射地址
UART0	0x40020000	0x40020000

表 16-3 SPI 寄存器偏移地址

寄存器名	地址偏移量	描述
SPI_CTL	0x0	控制寄存器
SPI_STAT	0x4	状态指示寄存器
SPI_TXDATA	0x8	数据发送寄存器
SPI_RXDATA	0xC	数据接收寄存器

- 控制寄存器 (0x0)

表 16-4 SPI 控制寄存器 SPI\_CTL

比特位	名称	描述	读/写标志	复位值
31:22	---	预留	R	0

21	---	预留	R/W	0
20	---	预留	R/W	0
19:12	CLKDIV	SCK 时钟分频系数 SCK 频率=系统时钟频率/(2*(CLKDIV + 1))	R/W	0
11:10	WIDTH	数据宽度选择 0: 8bit 1: 16bit 2: 32bit 3: 预留, 8bit	R/W	0
9	SCSN_EN	SCSN 模式错误检测使能, 只适用于主模式 0: 不使能主模式 SCSN 模式错误检测, SCSN 为通用 IO 1: 使能主模式 SCSN 模式错误检测, SCSN 作为 SPI 的输入	R/W	0
8	CPHA	时钟相位选择 0: 前边沿采样数据 1: 前边沿建立数据	R/W	0
7	CPOL	时钟极性选择 0: “SCK” 在空闲状态时被设置为低电平 1: “SCK” 在空闲状态时被设置为高电平	R/W	0
6	LMSB	LSB/MSB 选择 0: MSB 先传输 1: LSB 先传输	R/W	0
5	TXCOL_IRQ_EN	数据冲突中断使能 0: 关闭写冲突中断 1: 打开写冲突中断	R/W	0
4	RXCOL_IRQ_EN	数据冲突中断使能信号 0: 关闭读冲突中断 1: 打开读冲突中断	R/W	0
3	ERR_IRQ_EN	SCSN 模式错误中断使能 0: 关闭模式错误中断 1: 打开模式错误中断	R/W	0
2	TR_IRQ_EN	数据传输中断使能 0: 关闭发送数据中断 1: 打开发送数据中断	R/W	0
1	MAST/SLAV	主从选择 1: MASTER 0: SLAVE	R/W	1
0	EN	使能信号 0: 关闭 SPI 接口 1: 打开 SPI 接口	R/W	0

● 状态寄存器 (0x4)

表 16-5 SPI 状态寄存器 SPI\_STAT

比特位	名称	描述	读/写标志	复位值
31:5	---	预留	R	0
4	TXBUSY	数据发送忙状态标志。 0: 数据发送空闲, 总线可以发写 SPITX 寄存器命令 1: 数据正在发送过程中, 总线不能发写 SPITX 寄存器命令	R	0
3	TXCOLIF	写冲突标志。写 1 清零 正在发送 (即 TXBUSY 为 1) 时, 用户向 SPI 写入新的发送数据, 则新的发送数据将被丢弃, 并置写冲突标志为 1。 0: 没有写数据冲突中断 1: 产生写数据冲突中断	R/W	0
2	RXCOLIF	接收数据溢出标志。写 1 清零 连续数据接收时, 如用户不读取 RXDATA 寄存器, 将产生接收数据溢出事件 0: 没有接收数据溢出中断 1: 产生接收数据溢出中断	R/W	0
1	ERRIF	SCSN 模式冲突中断标识位: SPI 为主模式, 只有在 SCSN_EN 为 1 时, 同时检测到“SCSN”输入电平为低, 则该位置 1; SPI 为从模式, “SCSN”作为从机的片选输入, 在数据传输过程中, 若“SCSN”输入电平为高, 则该位置 1; 若 ERR_IRQ_EN=1, 则会产生一个中断, 一旦发生模式冲突错误, 则 SPI 模块复位。该位写 1 清零。 0: 没有模式冲突中断 1: 产生模式冲突中断	R/W	0
0	TRIF	数据传输中断标识位, 若数据传输结束, 该位置 1, 若 TR_IRQ_EN=1, 则会产生中断, 该位写 1 清零。 0: 没有数据发送中断 1: 产生数据发送中断, 发送数据寄存器为空	R/W	0

● 数据发送寄存器 (0x8)

表 16-6 SPI 数据发送寄存器 SPI\_TXDATA

比特位	名称	描述	读/写标志	复位值
31:0	TXDATA	数据发送寄存器	R/W	0

● 数据接收寄存器 (0xC)

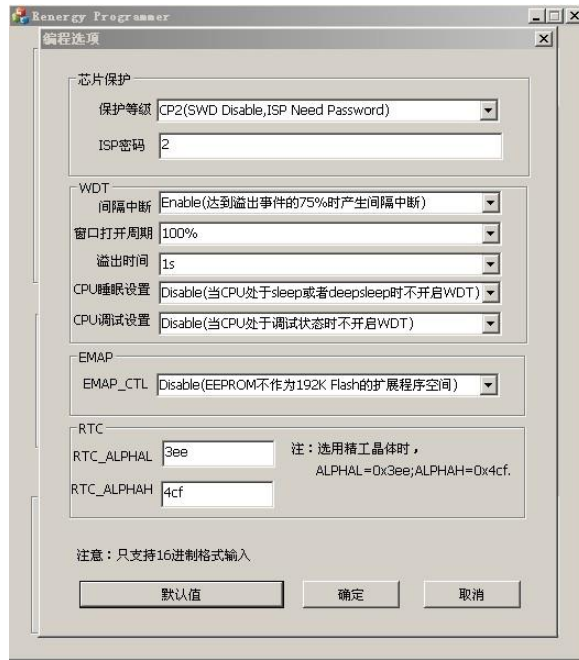
表 16-7 SPI 数据接收寄存器 SPI\_RXDATA

比特位	名称	描述	读/写标志	复位值
31:0	RXDATA	数据接收寄存器	R	0

## 17 选项字节

内置了一个选项字节的区域，当芯片发生复位时，会自动配置选项字节，执行设定的指定功能。选项字节包括对芯片保护，WDT，EMAP 和 RTC 的配置。

选项字节编程可以通过锐能微的编程工具（MINIPRO 编程器或者 ISP 编程工具）进行设置，以 MINIPRO 编程器为例，打开编程选项对话框，即可对选项字节进行设置，如下图所示（详细的操作方法请见《MINIPRO 编程器使用手册》）。



### 17.1 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））

### 17.2 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

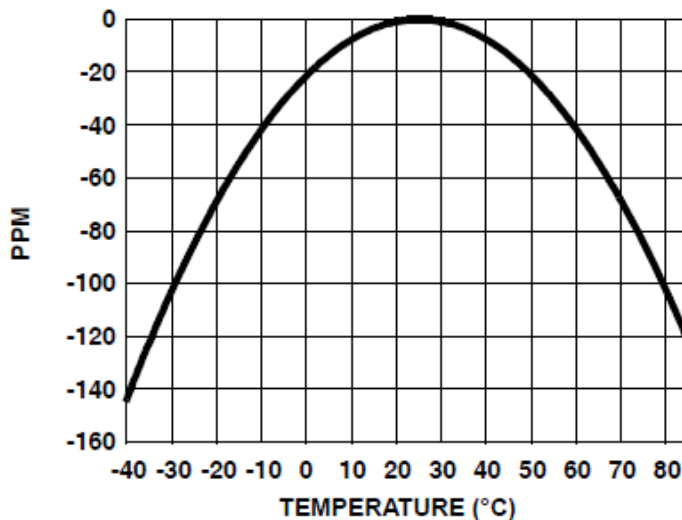
窗口打开周期	0: 25% 1: 50% 2: 75% 3: 100% 在窗口打开期间将 0xBB 写入 WDTE 寄存器，看门狗清零并重新计数； 在窗口关闭期间将 0xBB 写入 WDTE 寄存器，会产生内部复位信号。	3
溢出时间	0: 16ms 1: 32ms 2: 128ms 3: 512ms 4: 1s 5: 2s 6: 4s 7: 8s	4
CPU 睡眠设置	0: Disable (当 CPU 处于 sleep 或者 deepsleep 的时候不开启 WDT) 1: Enable (当 CPU 处于 sleep 或者 deepsleep 的时候开启 WDT)	0
CPU 调试设置	0: 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

### 17.3 RTC 设置

RN821X的RTC内置了自动温补功能，可以对32k晶体进行自动温度补偿，以提供在-25℃~70℃范围输出准确的秒脉冲。

其中，晶体的温度频率曲线如下图所示，是以顶点为25度的二次曲线 ( $f=f_0-\alpha*(T-T_0)$ ,  $T_0$ 为25度)。而事实上高温段 (25℃~85℃) 与低温段 (-25℃~70℃) 的二次曲线参数 $\alpha$ 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。



## 18 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；使用锐能微编程器完成 ISP 功能。

### 18.1 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；

### 18.2 Flash 保护机制

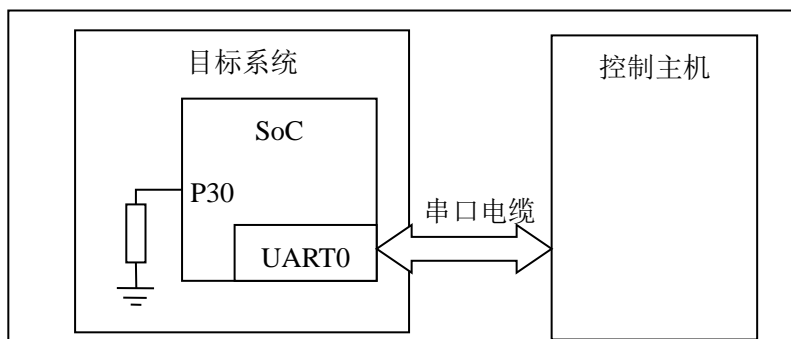
Flash保护是允许用户在系统中通过使能不同的安全级别来限制对片内Flash的访问和ISP使用的一种机制。保护机制包含如下保护等级。用户可以通过“选项字节”设置芯片的保护等级。

表18-1 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））

### 18.3 在系统编程（ISP）

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



ISP 的主要流程：

1. 按连接图配置并连接目标系统和控制主机；

2. 复位目标系统；
3. 控制主机配置串口为 1 个起始位，8 个数据位，1 个停止位，波特率不大于 115200bps；
4. 控制主机发送 “e”；
5. 目标系统响应 “Synchnonized/r/n”；
6. 控制主机发送 “Synchnonized/r/n”；
7. 目标系统响应 “7373(1843)/r/n”；（如果当前系统频率为 7.3728M，则发送 7373；如果为 1.8432M，则发送 1843）
8. 控制主机可以根据需要执行对应的 ISP 命令；

**18.3.1 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 18-2 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 18-3 ISP 数据校验格式

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

**18.3.2 使用的资源**



- 使用的 RAM 资源

ISP使用片内0x10001000到0x10002800范围内的RAM，堆栈位于RAM的顶部。Flash能使用0x10000000-0x10001000（4KB）范围内的RAM进行编程。

### 18.3.3 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18-4 ISP命令

命令	用法	属性
设置波特率	BS <波特率><停止位>	普通命令
回显	RD <开关设置>	普通命令
写内存	WM <地址><字节长度><模式>	普通命令
读内存（包括 flash, sram）	RM <地址><字节长度><模式>	普通命令
Flash 页擦除	FP <页地址>	普通命令
Flash 块擦除	FS <块地址>	普通命令
flash 片擦除	FC	
flash 块查空	FQ <块地址>	普通命令
内存比较	MC <地址1><地址2><字节长度>	普通命令
运行	GO <地址>	普通命令
解锁	UN <密码>	
获取保护等级	AL	
使能 PFP	PM <开关设置>	普通命令
软件复位	RS	普通命令
使能 NVM (FLASH)	NV <NVM选项>	普通命令

- 波特率设置

表18-5 ISP命令

命令	BS <波特率><停止位>
输入	波特率： 9600 或 19200 或 38400 或 57600 或 115200 停止位： 1 或 2
返回代码	CMD_SUCCESS 或 INVALID_BAUD_RATE 或 INVALID_STOP_BIT 或 INVALID_PARAM
说明	改变 ISP 通讯串口帧格式，包括波特率和停止位。串口起始位固定为 1，数据位固定为 8。新帧格式在返回 CMD_SUCCESS 后生效。
范例	“BS 9600 2” 设置串口波特率为9600bps，2个停止位。

- 回显

表 18-6 ISP 命令

命令	RD <开关设置>
输入	开关设置：0（关）或 1（开）
返回代码	CMD_SUCCESS 或 INVALID_PARAM
说明	命令和数据回显。缺省为开。回显打开状态下，SoC 会将收到的命令和数据发回主机。
范例	“RD 0” 关闭回显。

- 写内存

表18-7 ISP命令

命令	WM <地址><字节长度><模式>
输入	地址：起始地址，应该 32 位对齐； 字节数：应该是 4 的倍数； 模式：0 为串口，1 为并口
返回代码	CMD_SUCCESS 或 FM_MODE_ERROR 或 ADDR_NOT_ALIGN 或 COUNT_ERROR 或 COUNT_ERROR 或 ADDR_NOT_MAPPED 或 INVALID_PARAM
说明	向片内 SRAM 写入数据
范例	“WM 268436224 4 0” “78” “56” “34” “12” 采用串口向0x10000300地址写入0x12345678

- 读内存

表18-8 ISP命令

命令	RM <地址><字节长度><模式>
输入	地址：读出地址，应该 32 位对齐； 字节数：应该是 4 的倍数； 模式：0 为串口，1 为并口
返回代码	CMD_SUCCESS 或 FM_MODE_ERROR 或 ADDR_NOT_ALIGN 或 COUNT_ERROR 或 COUNT_ERROR 或 ADDR_NOT_MAPPED 或 INVALID_PARAM
说明	读取 SoC 片内 SRAM 的内容
范例	“RM 268436224 4 0” 采用串口读取片内SRAM地址0x10000300的内容。

- Flash 页擦除

表18-9 ISP命令

命令	FP <页地址>（FPGA 版本为 0 到 3071）
输入	页地址：0 到 1535 之间可选；
返回代码	CMD_SUCCESS 或 INVALID_PAGE 或 INVALID_PARAM
说明	擦除 SoC 片上 FLASH 指定块

范例	“FP 0” 擦除第0页的内容
----	-----------------

● Flash 块擦除

表18-10 ISP命令

命令	FS <块地址>
输入	块地址：0 到 47 之间可选；
返回代码	CMD_SUCCESS 或 INVALID_SECTOR 或 INVALID_PARAM
说明	擦除 SoC 片上 FLASH 指定块
范例	“FS 0” 擦除第0块的内容

● Flash 片擦除

表18-11 ISP命令

命令	FC
输入	无
返回代码	CMD_SUCCESS 或 INVALID_PARAM
说明	擦除 SoC 片上 FLASH 全部内容。
范例	“FC” 擦除片上FLASH全部内容

● Flash 块查空

表18-12 ISP命令

命令	FQ <块地址>
输入	块地址：0 到 47 之间可选；
返回代码	CMD_SUCCESS 或 INVALID_SECTOR 或 INVALID_PARAM
说明	检查片上 FLASH 指定块的内容是否为空（擦除后未编程）
范例	“FQ 1” 检查第1块的内容是否为空

● EEPROM 页擦除

表18-13 ISP命令

命令	EP<页地址>
输入	块地址：0 到 511 之间可选；
返回代码	CMD_SUCCESS 或 INVALID_PAGE 或 INVALID_PARAM
说明	擦除 SoC 片上 EEPROM 指定块
范例	“EP 1” 擦除第1页的内容

● EEPROM 块擦除

表18-14 ISP命令

命令	ES<块地址>
输入	块地址：0 到 7 之间可选；
返回代码	CMD_SUCCESS 或 INVALID_SECTOR 或 INVALID_PARAM
说明	擦除 SoC 片上 EEPROM 指定块
范例	“ES 1” 擦除第1块的内容

● EEPROM 片擦除

表18-15 ISP命令

命令	EC
输入	无
返回代码	CMD_SUCCESS 或 INVALID_PARAM
说明	擦除 SoC 片上 FLASH 全部内容。片擦除命令页将清除密钥和保护等级设置，将芯片置回出厂状态。
范例	“EC” 擦除片上EEPROM全部内容

● EEPROM 块查空

表18-16 ISP命令

命令	EQ<块地址>
输入	页地址：0 到 7 之间可选；
返回代码	CMD_SUCCESS 或 INVALID_SECTOR 或 INVALID_PARAM
说明	检查片上 EEPROM 指定块的内容是否为空（擦除后未编程）
范例	“EQ 1” 检查第1块的内容是否为空

● Flash 编程

表18-17 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

● EEPROM 编程

表18-18 ISP命令

命令	EP<EEPROM 地址><RAM 地址><字节长度>
输入	EEPROM 地址：要写入的 FLASH/EEPROM 目标地址 SRAM 地址：源缓冲区所在的 SRAM 地址 字节长度：写入的字节数量
返回代码	CMD_SUCCESS 或 COUNT_ERROR 或 SRC_ADDR_NOT_MAPPED 或 DST_ADDR_NOT_MAPPED 或 INVALID_PARAM
说明	用于编程EEPROM存储器。
范例	“EP 0 268436224 128” 将SRAM地址0x10000300开始的128字节复制到EEPROM地址 0

● 内存比较

表18-19 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18-20 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18-21 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18-22 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

- 使能 PFPM

表18-23 ISP命令

命令	PM <开关设置>
输入	开关设置：0 (关) 或 1 (开)
返回代码	CMD_SUCCESS 或 INVALID_PARAM
说明	该命令使能/不使能PFPM (并行编程模式)
范例	“PM 1” 将使能PFPM

● 软件复位

表18-24 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

● 使能 NVM

表18-25 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。

18.3.4 ISP 返回代码

表 18-25 ISP 命令

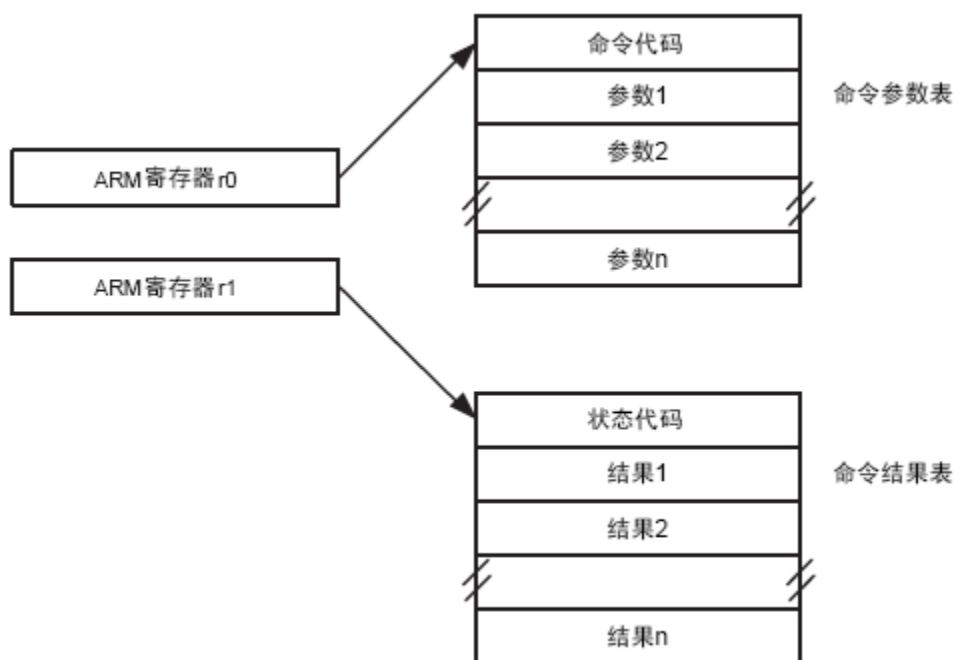
返回代码 (ASCII 码)	符号	描述
0	CMD_SUCCESS	成功执行命令。只有成功执行命令后, ISP处理器才发送该代码
1	INVALID_COMMAND	无效命令
2	INVALID_PARAM	无效参数 (参数的 ASCII 码不为 0-9)
3	INVALID_BAUD_RATE	无效波特率
4	INVALID_STOP_BIT	无效停止位
5	ADDR_NOT_ALIGN	地址不是以字为边界
6	COUNT_ERROR	字节计数值不是 4 的倍数
7	ADDR_NOT_MAPPED	所访问的地址空间越界
8	INVALID_SECTOR/INVALID_PAGE	无效 SECTOR_NUM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

12	DST_ADDR_NOT_ALIGN	目的地址不是以字节为边界
13	DST_ADDR_NOT_MAPPED	所访问的目的地址空间越界
14	COMPARE_ERROR	比对错误
15	FM_MODE_ERROR	操作内存模式错误
16	ADDR_NOT_THUMB	地址不为 Thumb 指令
17	INVALID_PASS	错误密码

### 18.4 在应用编程 (IAP)

对于在应用编程，应当通过寄存器r0中的字指针来调用IAP程序，该字指针指向含有命令代码和参数的存储器 (RAM)。IAP命令的结果返回到寄存器r1所指向的结果表。用户可以把寄存器r0 和r1 中的指针赋予相同的值，如此便能将命令表复用来存放结果。参数表应当大到足够保存所有的结果以防结果的数目大于参数的数目。参数传递见图21-7。参数和结果的数目根据IAP命令而有所不同。“Flash编程”命令参数的最大数目为4。结果的数目为1。命令处理器在接收到一个未定义的命令时发送状态代码INVALID\_COMMAND。IAP程序是Thumb代码，驻留在地址0x1800\_1c01。

图 18-2 IAP 参数传递



#### 18.4.1 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

#### 18.4.2 IAP 使用

IAP 的应用主要在以下方面：

◆ 在线升级（更新 FLASH）

在线升级时需要对 Flash 进行擦除/写操作。Flash 擦写操作持续 4ms 左右，会增加在此期间发生的中断的处理延迟。

一种 IAP 实现方法：

用户需要实现在线升级时，需要在软件设计中增加一个 IAP 升级的程序段。这段程序实现通过通讯口（如 UART）从远程主机接收程序，并使用 SoC 提供的 IAP 接口，将这些程序写入到 SoC 内部 FLASH 中。

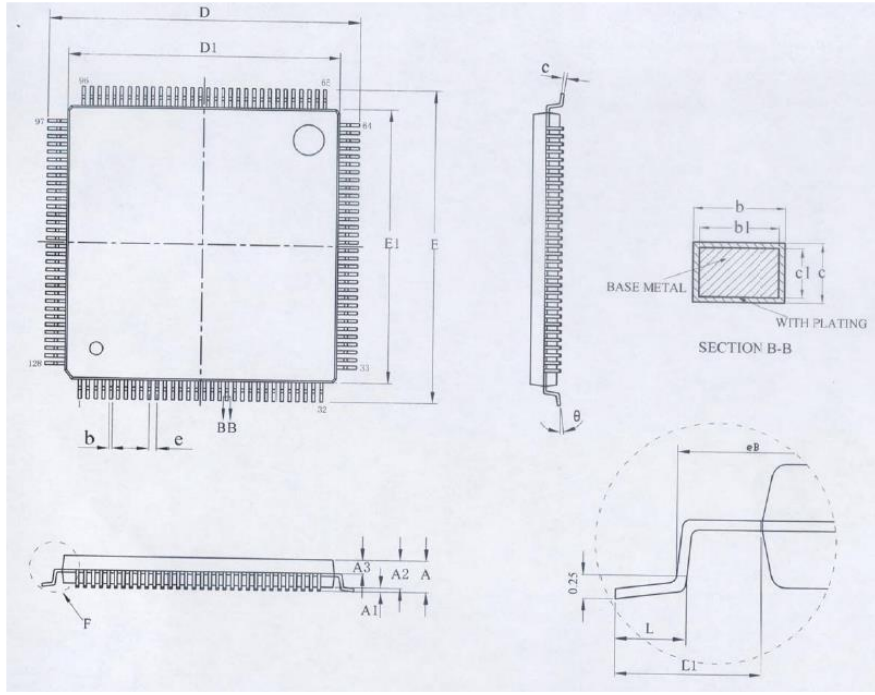
#### 18.5 量产平台

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》。



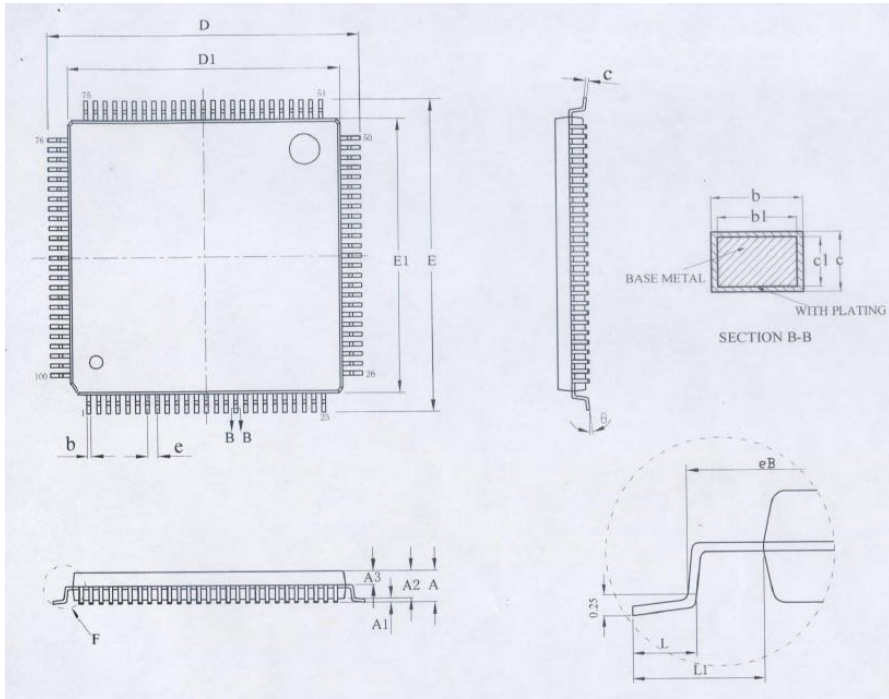
### 19 封装尺寸

<p>LQFP128L (1414×1.4)</p>		<p>14.00×14.00×1.40</p>	<p>e=0.40</p>
--------------------------------	---	-------------------------	---------------



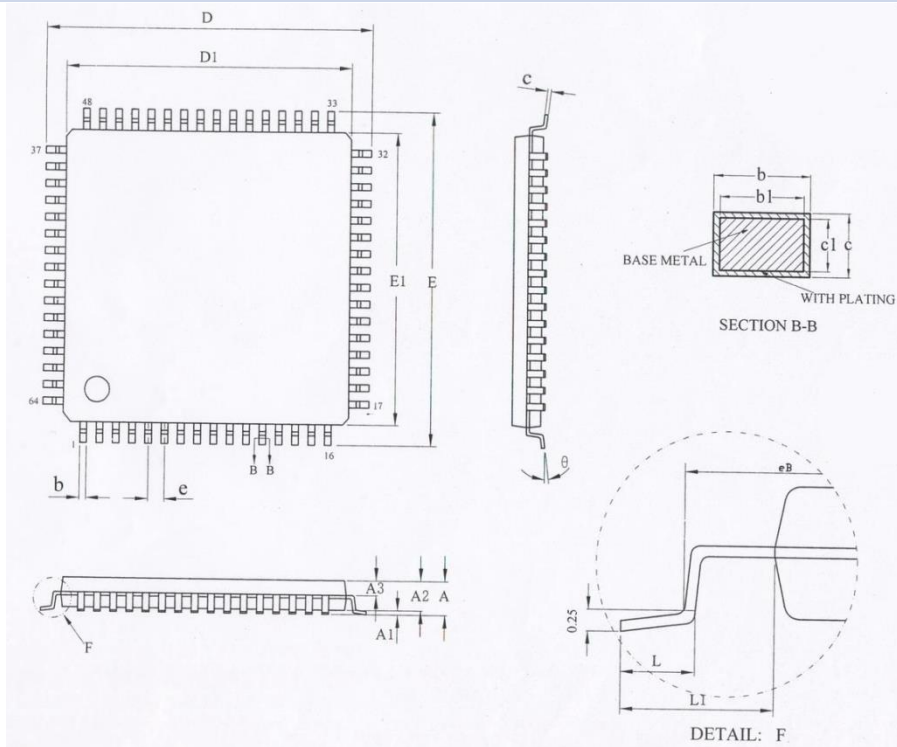
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	---	---	1.6
A1	0.05	---	0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.15	---	0.23
b1	0.14	0.16	0.19
c	0.13	---	0.18
c1	0.12	0.13	0.14
D	15.80	16.00	16.20
D1	13.90	14.00	14.10
E	15.80	16.00	16.20
E1	13.90	14.00	14.10
eB	15.05	---	15.35
e	0.40BSC		
L	0.45	---	0.75
L1	1.00BSC		
θ	0	-----	7°

<p>LQFP100L (1414×1.4)</p>		<p>14.00×14.00×1.40</p>	<p>e=0.50</p>
--------------------------------	---	-------------------------	---------------



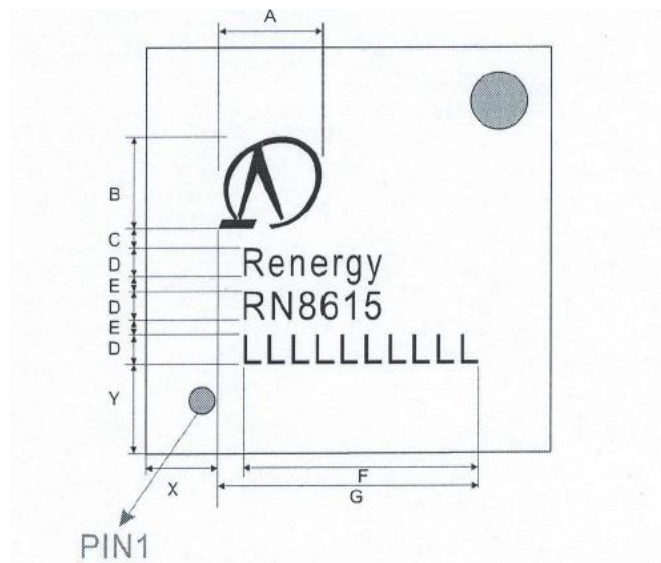
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	---	---	1.6
A1	0.05	---	0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.19	---	0.27
b1	0.18	0.20	0.23
c	0.13	---	0.18
c1	0.12	0.13	0.14
D	15.80	16.00	16.20
D1	13.90	14.00	14.10
E	15.80	16.00	16.20
E1	13.90	14.00	14.10
eB	15.05	---	15.35
e	0.50BSC		
L	0.45	---	0.75
L1	1.00BSC		
θ	0	-----	7°

<p>LQFP64L (0707×1.4)</p>		<p>7.00×7.00×1.40</p>	<p>e=0.40</p>
-------------------------------	---	-----------------------	---------------



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°

产品外观图如下，以 RN8615 为例，其他产品类似：



第一行为锐能微 Logo；

第二行 Renergy 为锐能微英文简称；

第三行 RN8xxx 为锐能微产品型号；

第四行为产品批号

左下脚的小点为 PIN1 标志。