

# HT5007\_HT5009

# 用户手册

Tel: 021-51035886

Fax: 021-50277833

Email: [sales@hitrendtech.com](mailto:sales@hitrendtech.com)

Web: <http://www.hitrendtech.com>

**版本更新说明**

版本号	修改时间	修订人	修改内容
V1.0	2014-11-12		初版
V1.1	2015-1-14		1. flash 的加密功能的值 2. BOR 阈值档位配置 3. P1offset P2offset Q1offset Q2offset 计算公式
V1.2	2015-1-29		增加 HT5009 产品信息
V1.3	2015-2-11		1. 用户在使用 INT 外部中断引脚功能的时候，必须要将 PINFLT 寄存器中的相应的引脚数字滤波功能也打开。 2.修改 ADC 波形采样寄存器，Ipeak Upeak 说明，ADC 位数和符号位说明。 3.增加快速慢速寄存器在低功耗模式下的说明。 4.修改 EMU 部分寄存器说明。 5.增加 RTC 模块最低工作电压描述 6.唤醒复位还不能复位所有中断标志寄存器 7.增加“按照年、月、日、时、分、秒、周的顺序操作”的说明 8.ADC 通道 1 测量电压计算公式修改 9.芯片引脚结构说明图更改
V1.4	2015-5-27		1.明显的描述错误勘误； 2.更改和软件库函数无法对应的查询内容
	2015-06-08		1.LVDIN 分时开启说明更改,LVDIN 检测阈值修改为 1.21V（之前 1.18V 数据有误） 2.措辞和文档结构优化 3.MCON01、MCON23、MCON45 默认值修改 4.WDT 章节修改 5.QPHSCal 默认值改为 0x0000，笔误 6.按键扫描这一章勘误
	2015-09-29		1.功耗参数修改，增加数字模块关闭数值 2.ADC1 测量值大于 ADC1CMP 寄存器值改为小于，属笔误 3.WAKEIF 寄存器唤醒源的补充 4.RTC 读写流程
V1.41	2016-3-12	Hma	1. RTC 模块概述修改 2. 删除 RTC 章节的 524288Hz 3. RTC2CAL 寄存器默认值有误，修改为 0x7fff 4. LRC 频率分布值

V1.5	2017-08-21	Hma	<ol style="list-style-type: none"> <li>1.删除 SPI,I2C 章节;</li> <li>2.删除 TMR 中 PWM, 捕获等功能;</li> <li>3.勘误。</li> </ol>
V1.6	2019-11-6	Hshan	<ol style="list-style-type: none"> <li>1.增加 1.2.2 章节处理器和外设 INT 描述</li> <li>2.删除寄存器, 框图 SPI, UART1/2/5, INT5/6 描述</li> <li>3.删除引脚定义表格多余引脚描述</li> <li>4.修改 Vbat 电池电压计算公式</li> <li>5. LCD 章节删掉小电流模式, 更改为 1/64 快速充电模式, 以及内阻阻值修改, LCD 功耗参数修改</li> <li>6.外部中断数字滤波使能的备注删除</li> <li>7.键盘扫描模块 20.1 概述描述修改</li> <li>8.修改“引脚定义”, F 版之后 ADCIN0~2 输入“滤波 2us”描述去掉。</li> <li>9.修改 LCDCON 对比度基准电压为 VCC 的百分比, 修改充电模式的电阻网路值。</li> <li>10.修改 I10ff/I20ff/Uoff 和 SPL 的对齐方式, 和 HT5X2X 兼容</li> <li>11.增加 15.3.4 传输模式 包含 IIC 的时序逻辑图</li> <li>12.修改 RTC 读写指示图</li> <li>13.在 I2C 章节修改 I2C 口的映射管脚, PF0/PF1 修改为 PC13、PC14, 与管脚定义一致;</li> <li>14.修改 DMA CTL 寄存器列表, 删除</li> <li>15.定时器的周期定时和 PWM 功能描述以及图有修改</li> <li>16.更改极限参数 Via 的最大值</li> <li>17.修改 QPhsCal 寄存器说明以及校表过程 HFCONST 计算公式的 femu 改为采样率 fs</li> <li>18.修改 DFih 寄存器描述, DFih 为 21 位有效位</li> <li>19.修改极限参数 Vesd 的值</li> <li>20.看门狗复位脉冲由 1088Fosc 改为 64Fosc</li> <li>21.修改复位章节图表, 并且复位时钟都增加了 Flrc</li> <li>22.增加 Sleep、hold 功耗的最大值</li> <li>23.修改 SAG、OVI 功能描述</li> <li>24.删除描述关于关闭低功耗下看门狗的 CLKCTRL0 的 WDT_EN 位, 不存在该位</li> <li>25.修改 BOR 复位等待计数时间 1024flrc 为 1088flrc/fosc</li> <li>26.增加 2.4 Flash 控制功能章节</li> <li>27.PMU 章节 VDETPCFG. VDET_TIME[1:0]=11 增加 (Recommend), 同时增加注释: VCC_DET、LVDIN_DET 和 BOR_DET 模块关闭再开启后, 内部检测模块会等待 VDET_TIME 时间后, 开始更新检测状态。VDET_PRD[2:0]增加注释: 检测周期设定需大于检测时间。</li> <li>28.修改 EMU 模块中 Upeak Ipeak 寄存器说明中的位数;</li> <li>29.修改 Timer 章节举例定时最大周期计算公式</li> </ol>

			<ul style="list-style-type: none"><li>30. 删除 IO 写保护描述，添加 RTC 写保护</li><li>31. 1.5 章节补充 PA6 引脚各功能说明</li><li>32. 修正捕获中断 CMPIE 为 CAPIE</li><li>33. 修正 DFIL 描述</li><li>34. 补充 tbs 分时开启功耗说明、adc 通道功耗说明及时序图</li><li>35. 删除定时器 13.3.2 比较中断描述，以及寄存器关于捕获，PWM，比较中断的部分描述</li><li>36. 增加 EMU 模块计量/校表寄存器列表到寄存器说明跳转链接设置</li></ul>
--	--	--	--

## 目录

目录.....	5
<b>1 HT500X 概述.....</b>	<b>11</b>
1.1 简介.....	11
1.2 特点.....	11
1.2.1 基本特点.....	11
1.2.2 处理器及外设.....	11
1.2.3 电能计量.....	12
1.3 缩略语.....	12
1.4 引脚排列.....	14
1.5 引脚定义.....	15
<b>2 存储器模块.....</b>	<b>19</b>
2.1 概述.....	19
2.2 存储器映射图.....	20
2.3 FLASH 操作.....	21
2.3.1 Flash 的读保护.....	21
2.3.2 Code Flash 的操作说明.....	21
2.3.3 Information Block 的操作说明.....	22
2.4 FLASH 控制功能.....	22
2.5 写保护寄存器列表.....	23
2.6 特殊功能寄存器列表.....	24
2.7 特殊功能寄存器说明.....	24
<b>3 时钟单元.....</b>	<b>27</b>
3.1 时钟分类.....	27
3.2 时钟框图.....	27
3.3 时钟停振检测框图.....	29
3.4 时钟说明.....	29
3.4.1 内部低频 RC 时钟 (Flrc).....	29
3.4.2 内部高频 RC 时钟 (Fhrc).....	29
3.4.3 内部 PLL 时钟 (Fpll).....	30
3.4.4 时钟安全机制.....	30
3.4.5 时钟异常状态处理.....	31
3.5 特殊功能寄存器列表.....	32
3.6 特殊功能寄存器说明.....	32
<b>4 电源单元.....</b>	<b>42</b>
4.1 概述.....	42

4.2	框图.....	43
4.3	电源单元详细功能说明.....	44
4.3.1	电源切换.....	44
4.3.2	电源实时监测.....	44
4.3.3	内建 1.5V 电源.....	44
4.3.4	BOR 检测功能(BOR_DET).....	45
4.3.5	系统电源检测功能(VSYS_DET).....	47
4.3.6	低电压检测功能(LVD_DET).....	48
4.3.7	VSYS_DET, BOR_DET, LVD_DET 分时检测时序.....	48
4.3.8	电池防钝化功能.....	49
4.4	特殊功能寄存器列表.....	49
4.5	特殊功能寄存器说明.....	49
<b>5</b>	<b>调试支持.....</b>	<b>55</b>
5.1	概况.....	55
5.2	SW 引脚分布.....	55
5.3	SW 引脚使用说明.....	55
<b>6</b>	<b>工作模式.....</b>	<b>56</b>
6.1	芯片电源域分配.....	56
6.2	工作模式.....	56
6.3	睡眠模式 (SLEEP).....	57
6.3.1	SLEEP 模式下各模块开关.....	57
6.3.2	SLEEP 模式下的唤醒.....	57
6.3.3	从 SLEEP 模式唤醒后的唤醒方式确认.....	58
6.3.4	进入 Sleep 模式.....	58
6.4	待机模式 (HOLD).....	59
6.4.1	进入 Hold 模式.....	59
6.4.2	模式转换图.....	59
6.5	特殊功能寄存器列表.....	60
6.6	特殊功能寄存器说明.....	60
<b>7</b>	<b>GPIO 模块.....</b>	<b>61</b>
7.1	概述.....	61
7.2	芯片引脚结构说明.....	62
7.3	芯片引脚描述.....	62
7.4	I/O 端口基地址列表.....	63
7.5	特殊功能寄存器说明.....	63
<b>8</b>	<b>中断模块.....</b>	<b>68</b>
8.1	中断向量说明.....	68
8.2	中断使能和禁止.....	69
8.2.1	中断使能和禁止相关寄存器列表.....	69
8.2.2	中断使能和禁止相关寄存器说明.....	70

8.3	特殊功能寄存器列表.....	71
8.4	特殊功能寄存器说明.....	71
<b>9</b>	<b>RESET 模块.....</b>	<b>74</b>
9.1	复位优先级.....	74
9.2	复位说明.....	75
9.2.1	上电复位.....	75
9.2.2	低电压检测复位.....	75
9.2.3	外部引脚复位.....	76
9.2.4	掉电复位.....	76
9.2.5	看门狗复位.....	77
9.2.6	软复位.....	77
9.2.7	调试复位.....	78
9.2.8	唤醒复位.....	78
9.3	特殊功能寄存器列表.....	79
9.4	特殊功能寄存器说明.....	79
<b>10</b>	<b>UART/7816 通讯模块.....</b>	<b>83</b>
10.1	功能说明.....	83
10.2	波特率计算.....	84
10.3	串口通讯模式说明.....	84
10.3.1	方式1.....	84
10.3.2	方式2.....	85
10.3.3	方式3.....	85
10.3.4	方式4.....	86
10.4	7816 接收和发送.....	87
10.4.1	7816 数据发送.....	87
10.4.2	7816 数据接收.....	87
10.4.3	7816 通讯示意图.....	88
10.5	特殊功能寄存器列表.....	91
10.6	特殊功能寄存器说明.....	91
<b>11</b>	<b>LCD 模块.....</b>	<b>98</b>
11.1	概述.....	98
11.2	LCD 与 GPIO 口复用表.....	98
11.3	LCD 框图.....	98
11.4	输出波形.....	99
11.5	LCD 显示操作.....	103
11.6	特殊功能寄存器列表.....	104
11.7	特殊功能寄存器说明.....	104
<b>12</b>	<b>WDT 模块.....</b>	<b>108</b>
12.1	概述.....	108
12.2	工作模式.....	108

12.3 特殊功能寄存器列表.....	108
12.4 特殊功能寄存器说明.....	109
<b>13 定时器模块.....</b>	<b>110</b>
13.1 定时器单元概述.....	110
13.2 周期定时功能.....	110
13.3 中断功能.....	111
13.3.1 定时器周期中断.....	111
13.4 特殊功能寄存器列表.....	111
13.5 特殊功能寄存器说明.....	111
<b>14 I2C 模块.....</b>	<b>115</b>
14.1 概述.....	115
14.2 框图.....	115
14.3 功能描述.....	115
14.3.1 操作模式.....	116
14.3.2 串行时钟生成.....	116
14.3.3 中断生成.....	116
14.3.4 传输模式.....	116
14.4 特殊功能寄存器列表.....	124
14.5 特殊功能寄存器说明.....	125
<b>15 RTC 模块.....</b>	<b>133</b>
15.1 概述.....	133
15.2 功能描述.....	133
15.3 时钟校正.....	133
15.4 RTC 补偿系数寄存器和 INFO FLASH 对应关系.....	133
15.5 时间和万年历.....	134
15.6 中断功能.....	134
15.7 RTC 指示寄存器读写流程.....	135
15.7.1 读取 RTC 指示寄存器流程.....	135
15.7.2 写入 RTC 指示寄存器流程.....	135
15.8 校时记录.....	136
15.9 辅助 RTC.....	136
15.10 特殊功能寄存器列表.....	136
15.11 特殊功能寄存器说明.....	138
15.12 特殊功能寄存器说明.....	138
<b>16 TBS 模块.....</b>	<b>157</b>
16.1 概述.....	157
16.2 功能描述.....	157
16.3 特殊功能寄存器列表.....	158
16.4 特殊功能寄存器说明.....	158
<b>17 AES&amp;GHASH 模块.....</b>	<b>166</b>



17.1 AES 概述 .....	166
17.2 框图.....	166
17.3 特殊功能寄存器列表.....	166
17.4 特殊功能寄存器说明.....	168
<b>18 DMA 功能 .....</b>	<b>179</b>
18.1 概述.....	179
18.2 功能描述.....	179
18.3 DMA 通道请求列表 .....	179
18.4 DMA 数据传输说明 .....	181
18.5 特殊功能寄存器列表.....	182
18.6 特殊功能寄存器说明.....	182
<b>19 按键扫描模块 (KEYSCAN) .....</b>	<b>187</b>
19.1 概述.....	187
19.2 特殊功能寄存器列表.....	187
19.3 特殊功能寄存器说明.....	187
<b>20 CORTEX-M0 内核简要说明 .....</b>	<b>189</b>
20.1 概述.....	189
20.2 系统定时器 SYSTICK .....	189
20.3 中断优先级说明.....	189
20.4 CMSIS 函数说明.....	189
<b>21 EMU 模块.....</b>	<b>191</b>
21.1 概述.....	191
21.2 功能描述.....	191
21.2.1 模数转换器 .....	191
21.2.2 ADC 采样输出和功率波形输出.....	192
21.2.3 有功功率、无功功率和视在功率.....	192
21.2.4 有效值 .....	193
21.2.5 电压线频率 .....	194
21.2.6 起/潜动.....	194
21.2.7 功率反向指示 .....	194
21.2.8 直流偏置校正 .....	194
21.2.9 电能脉冲输出 .....	194
21.2.10 窃电检测 .....	196
21.2.11 中断源.....	197
21.2.12 SAG 功能描述.....	197
21.2.13 OVI 功能描述.....	198
21.2.14 掉零线计量模式(Loss of Voltage).....	198
21.3 特殊功能寄存器列表.....	199
21.3.1 计量参数寄存器.....	199
21.3.2 校表参数寄存器.....	211

---

21.4 校表过程.....	244
21.4.1 方案一.....	244
21.4.2 方案二.....	245
<b>22 电气规格.....</b>	<b>248</b>
22.1 DC 参数.....	248
22.2 极限参数.....	249
22.3 功耗参数.....	249
22.4 计量 ADC 参数.....	251
22.5 VBAT 测试参数.....	251
22.6 VCC 测试参数.....	251
22.7 IO 口性能指标.....	251
<b>23 封装.....</b>	<b>253</b>
23.1 HT5007:LQFP64.....	253
23.2 HT5009:LQFP48.....	254

# 1 HT500X 概述

## 1.1 简介

HT500X 是一颗低功耗、高性能的单相电能计量 SoC 芯片，片内集成 32-bit ARM 内核、64K flash、4K SRAM，支持断相防窃电功能的硬件 EMU 模块，带有温度自补偿功能的高精度 RTC 模块，以及 LCD 驱动等功能，为单相多功能、防窃电电能表提供高集成的单芯片解决方案。

## 1.2 特点

### 1.2.1 基本特点

- 工作电压范围：2.0V~5.5V
- 工作温度范围：-40°C~85°C
- 无铅绿色封装：LQFP64：HT5007  
LQFP48：HT5009

### 1.2.2 处理器及外设

- 基于 ARM 32-bit Cortex-M0 CPU Core 设计，支持 Thumb-2 指令集，包含嵌套向量中断控制器 NVIC，以及可扩展的调试技术
- 存储器资源：64K Flash+512bytes Information Block、4K SRAM
- 高速系统时钟：采用 PLL 倍频模式  
插入等待 CPU 最高工作频率 39.32MHz  
无等待 CPU 最高工作频率 19.66MHz
- 系统低功耗：待机模式（Hold 模式）下最低功耗 3.7uA  
睡眠模式（Sleep 模式）下最低功耗 2.9uA
- 具有电源监测功能，外部系统电源 VSYS 和电池输入 VBAT 两个电源内部切换
- 内置 9.5MHz HRC、32KHz LRC，具有多路时钟检测功能
- 支持 SW 调试协议
- RTC 模块：  
外置 32K 低频晶体，集成晶体所需电阻、电容  
自动温补：内置 RTC 温度曲线数字补偿系数，在全温度范围内，RTC 补偿无需用户软件参与  
辅助 RTC：外部 OSC 晶振停振时，系统可切换至内部低频 RC 获得计时时间
- 高精度温度传感器 TPS：-40 度 ~ +85 度温度范围内，温度传感器一致性优于正负 0.5 度
- LCD：支持 4COM, 6COM, 8COM 的 LCD 显示，SEG 接口最多支持 24 段
- 最多支持 3 路 UART，其中复用 2 路硬件 7816 协议功能
- 正常模式下，WDT 模块不可关闭，保证系统可靠运行
- 内置硬件 AES/GHASH 加密/解密算法

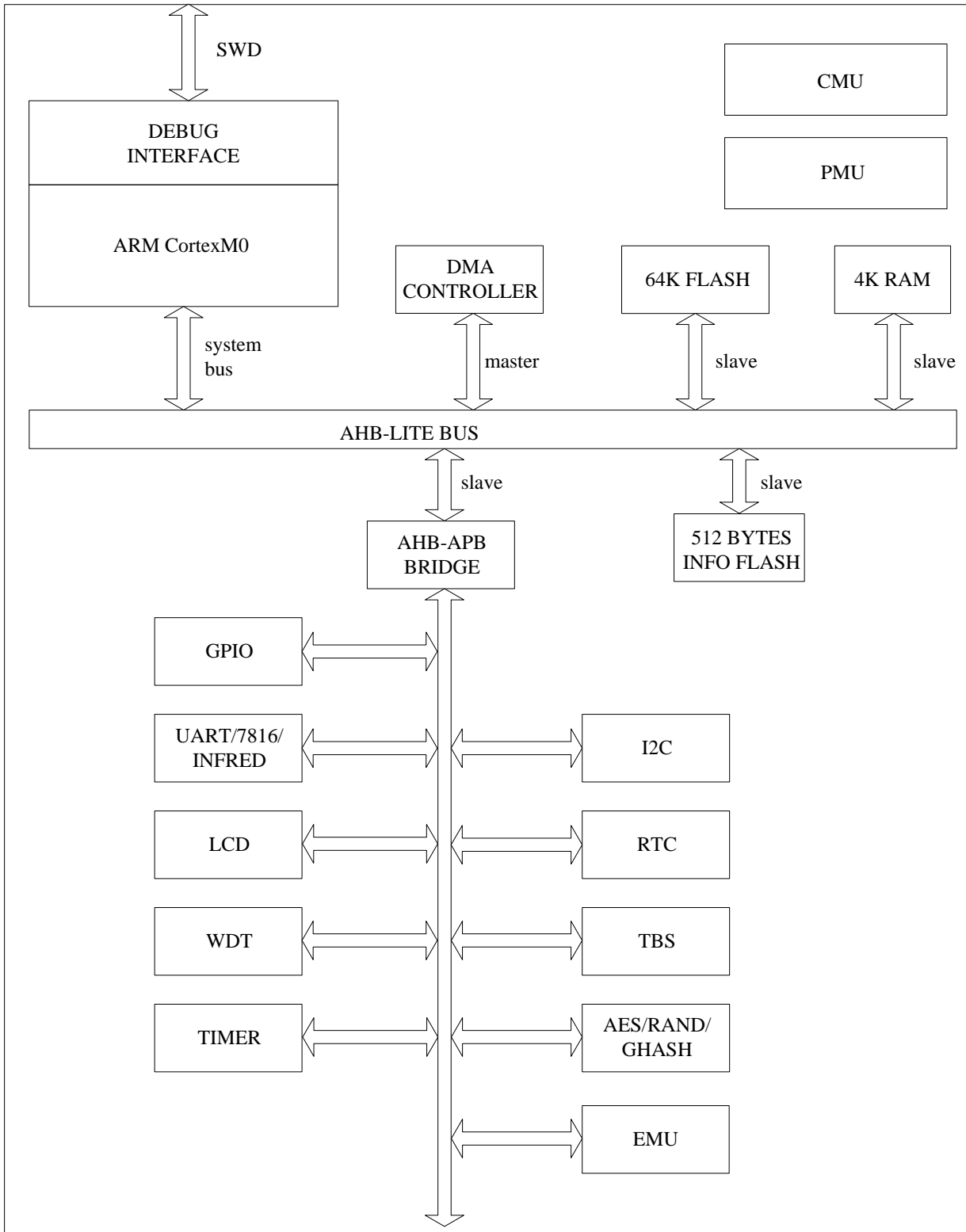
- 内置 DMA 功能
- 内置按键扫描功能，支持 1\*4，2\*4 两种键盘扫描
- 支持 I2C
- 支持 4 路外部中断 INT

### 1.2.3 电能计量

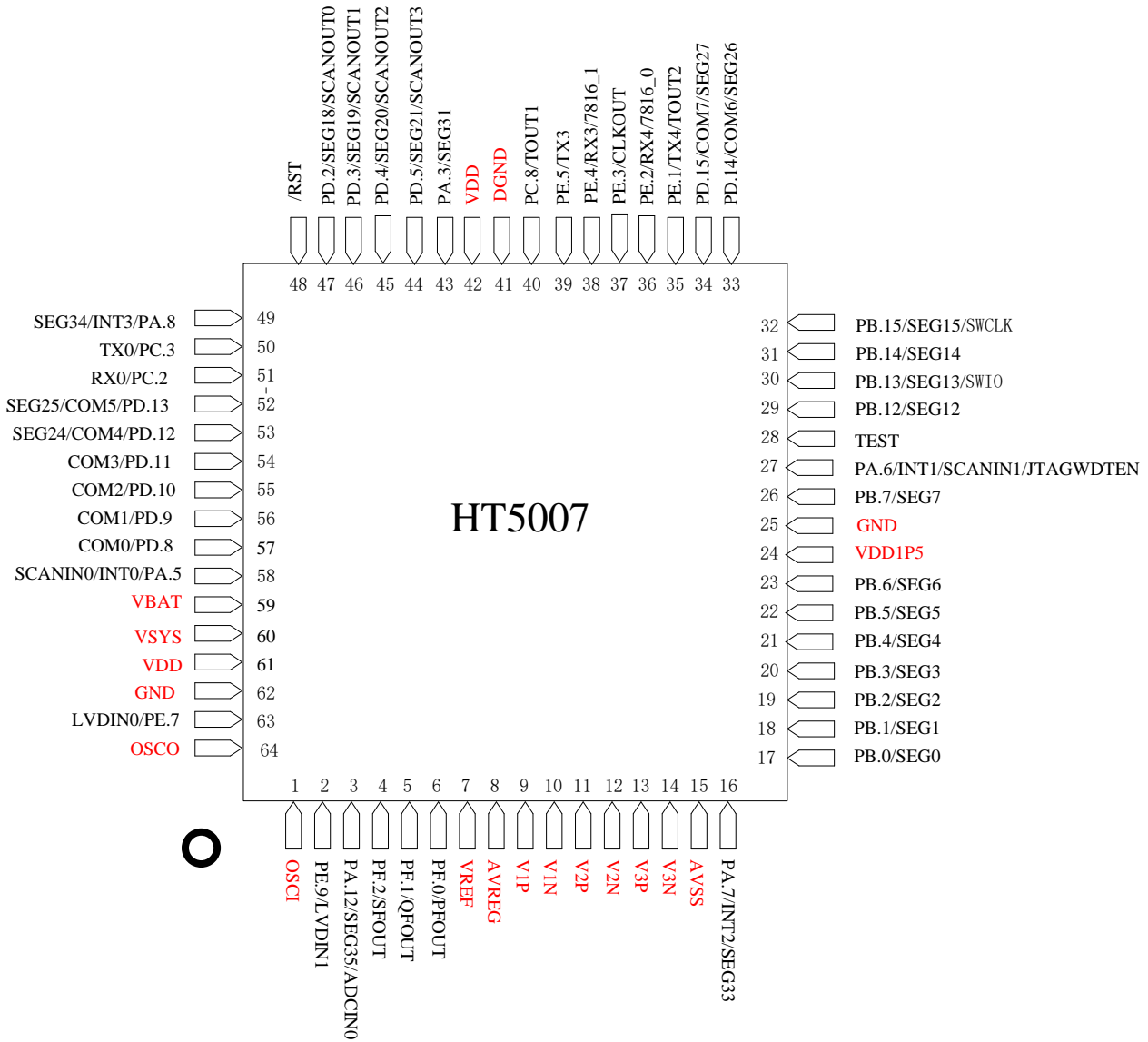
- 有功电能测量误差小于 0.1%，动态范围大于 5000:1，支持 IEC62053-21、IEC62053-22、IEC62053-23
- 三路  $\Sigma$ - $\Delta$  ADC，可同时输出两个计量通道的参数，支持灵活的防窃电功能
- 提供有功功率、无功功率、视在功率，有功能量、无功能量、视在能量
- 支持有功、无功、视在脉冲输出，并开放脉冲计数寄存器
- 提供高精度有效值、频率、ADC 波形数据等计量参数
- 支持 SAG、PEAK 功能
- 提供多种能量累加方式选择
- 支持增益误差、相位误差的软件校表
- 分别提供两个计量通道的可配置防潜动功能
- 支持 EMU 低功耗模式：EMU 低功耗工作频率可配置为 204.8KHz 或 32KHz，支持掉零线、掉火线的防窃电计量，低功耗连续计量，系统功耗优于 650uA
- 支持直流计量
- 支持单相三线计量制
- 片内基准电压：1.2V，温度系数  $\pm 10\text{ppm}/^\circ\text{C}$

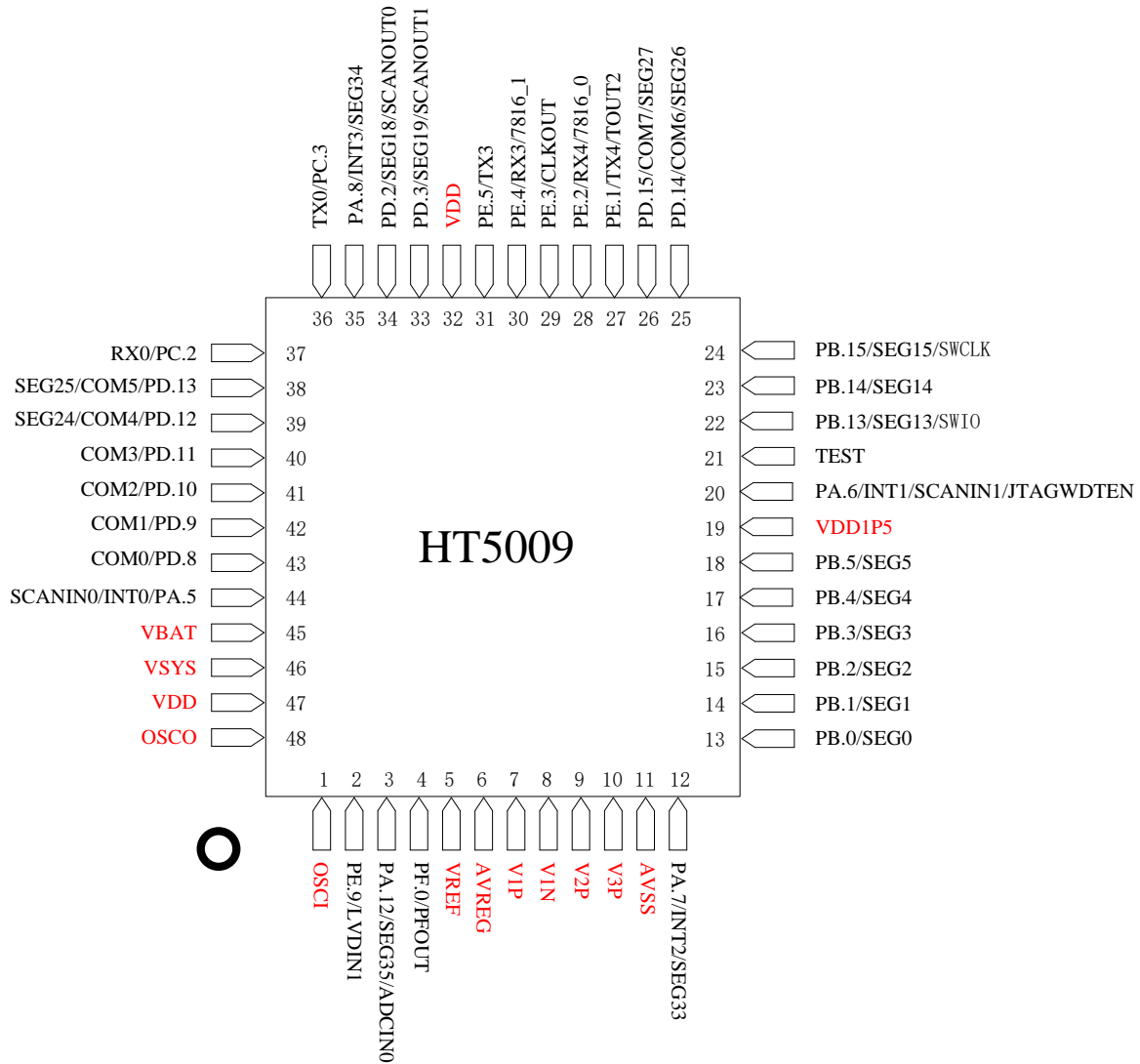
## 1.3 缩略语

缩略语	英文原文	中文含义
WDT	Watch Dog Timer	看门狗
GPIO	General Purpose IO	通用 I/O
TBS	Temperature Battery Sensor	温度&电池传感器
LVD	Low Voltage Detect	低电压检测
POR	Power On Reset	上电复位
BOR	Brown Out Reset	掉电复位
WKR	Wakeup Reset	唤醒复位
EMU	Energy Mearsurment Unit	电能计量单元
PMU	Power Management Unit	系统电源管理单元
CMU	Clock Management Unit	系统时钟管理单元
RTC	Real Time Clock	实时时钟
RSRV	Reserved	保留位



## 1.4 引脚排列





## 1.5 引脚定义

64 PIN	48 PIN	标识 (IOCFG=0)	引脚类型	第一复用功能 (IOCFG=1 AFCFG=0)	第二复用功能 (IOCFG=1 AFCFG=1)	引脚说明
1	1	OSCI	I			外部低频 32KHz 时钟输入
2	2	PE. 9	I/O	LVDIN1	LVDIN1	外部电源检测
3	3	PA. 12	I/O	SEG35	ADCIN0	外部电压信号采样输入 0
4		PF. 2	I/O	SFOUT	SFOUT	默认视在脉冲输出, 5mA 驱动能力, 可通过 EMCON 配置为其它类型脉冲输出
5		PF. 1	I/O	QFOUT	QFOUT	默认无功脉冲输出, 5mA 驱动

						能力,可通过 EMCON 配置为其它类型脉冲输出
6	4	PF. 0	I/O	PFOUT	PFOUT	默认有功脉冲输出, 5mA 驱动能力,可通过 EMCON 配置为其它类型脉冲输出
7	5	VREF	OUT			1.185V 计量基准电压输出,需外接滤波电容 0.1uF&1uF
8	6	AVREG	P			2.8V 内部模拟电源输出, 需外接滤波电容 0.1uF&10uF
9	7	V1P	I			电流通道 1 模拟信号输入端(正), 内部 ESD 保护, 最大信号±800mVp
10	8	V1N	I			电流通道 1 模拟信号输入端(负), 内部 ESD 保护, 最大信号±800mVp
11	9	V2P	I			电流通道 2 模拟信号输入端(正), 内部 ESD 保护, 最大信号±800mVp
12		V2N	I			电流通道 2 模拟信号输入端(负), 内部 ESD 保护, 最大信号±800mVp
13	10	V3P	I			电压通道模拟信号输入端(正), 内部 ESD 保护, 最大信号±800mVp
14		V3N	I			电压通道模拟信号输入端(负), 内部 ESD 保护, 最大信号±800mVp
15	11	AVSS	G			模拟地
16	12	PA. 7	I/O	INT2	SEG33	滤波 2us, 输出驱动 30mA
17	13	PB. 0	I/O	SEG0	SEG0	LCD Segment
18	14	PB. 1	I/O	SEG1	SEG1	LCD Segment
19	15	PB. 2	I/O	SEG2	SEG2	LCD Segment
20	16	PB. 3	I/O	SEG3	SEG3	LCD Segment
21	17	PB. 4	I/O	SEG4	SEG4	LCD Segment
22	18	PB. 5	I/O	SEG5	SEG5	LCD Segment
23		PB. 6	I/O	SEG6	SEG6	LCD Segment
24	19	VDD1P5	P			内部 1.5V 输出, 需外接 0.1uF 滤波电容
25		GND	G			模拟地
26		PB. 7	I/O	SEG7	SEG7	LCD Segment



27	20	PA. 6	I/O	INT1	SCANIN1	大功耗 GPIO(输出驱动 30mA)\外部中断(滤波 2us)\键盘扫描列线扫描输入口线 仿真控制接口 JTAGWDTEN: 当 TEST PIN 和此 PIN 同时为低电平时, 系统进入仿真模式。
28	21	TEST	I			测试引脚, 滤波 2us
29		PB. 12	I/O	SEG12	SEG12	LCD Segment
30	22	PB. 13	I/O	SEG13	SEG13	SW-I/O
31	23	PB. 14	I/O	SEG14	SEG14	LCD Segment
32	24	PB. 15	I/O	SEG15	SEG15	SW-CLK
33	25	PD. 14	I/O	COM6	SEG26	LCD Segment
34	26	PD. 15	I/O	COM7	SEG27	LCD Segment
35	27	PE. 1	I/O	TX4	TOUT2	UART4 发送端、秒脉冲输出 2, 输出驱动 5mA
36	28	PE. 2	I/O	RX4	7816_0	UART4 接收端, 滤波 2us
37	29	PE. 3	I/O	CLKOUT	CLKOUT	ESAM/CARD_CLK, 时钟输出
38	30	PE. 4	I/O	RX3	7816_0	UART3接收端, 滤波2us
39	31	PE. 5	I/O	TX3	TX3	UART3发送端
40		PC. 8	I/O	TOUT1	TOUT1	秒脉冲输出 1
41		DGND	G			芯片数字地
42	32	VDD	P			内部电源输入, 必需由另一个 VDD 外部连接, 提供内部数字电源
43		PA. 3	I/O	SEG31	SEG31	LCD Segment
44		PD. 5	I/O	SEG21	SCANOUT3	键盘扫描口线, 输出驱动 5mA
45		PD. 4	I/O	SEG20	SCANOUT2	键盘扫描口线, 输出驱动 5mA
46	33	PD. 3	I/O	SEG19	SCANOUT1	键盘扫描口线, 输出驱动 5mA
47	34	PD. 2	I/O	SEG18	SCANOUT0	键盘扫描口线, 输出驱动 5mA
48		/RST	I			复位信号(低电平有效), 滤波 2us
49	35	PA. 8	I/O	INT3	SEG34	外部中断 3, 滤波 2us, 输出驱动 30mA
50	36	PC. 3	I/O	TX0	TX0	UART0 发送端
51	37	PC. 2	I/O	RX0	RX0	UART0 接收端, 滤波 2us
52	38	PD. 13	I/O	COM5	SEG25	LCD Segment
53	39	PD. 12	I/O	COM4	SEG24	LCD Segment

54	40	PD. 11	I/O	COM3	COM3	LCD command
55	41	PD. 10	I/O	COM2	COM2	LCD command
56	42	PD. 9	I/O	COM1	COM1	LCD command
57	43	PD. 8	I/O	COM0	COM0	LCD command
58	44	PA. 5	I/O	INT0	SCANINO	外部中断 0, 滤波 2us
59	45	VBAT	P			电池供电输入
60	46	VSYS	P			系统电源输入
61	47	VDD	P			系统电源输出
62		GND	G			模拟地
63		PE. 7	I/O	LVDINO	LVDINO	外部电源检测
64	48	OSCO	O			外部低频时钟驱动输出

- 注：1. I=输入；O=输出；P=电源；G=地；
2. 数字输出引脚都可配开漏功能(Open Drain)。
3. 数字输入引脚（除 RST/TEST/JTAGWDTEN，这三个引脚内部恒定上拉）都可配上拉功能。
4. PA. 6 引脚为一个特殊引脚，当 TEST=0 时，该引脚为输入 JTAG\_WDTEN 功能。
5. 两个 VDD 引脚外部需要连接在一起。
6. 48PIN 封装（HT5009）V2N 和 V3N 引脚内部连接至 GND。

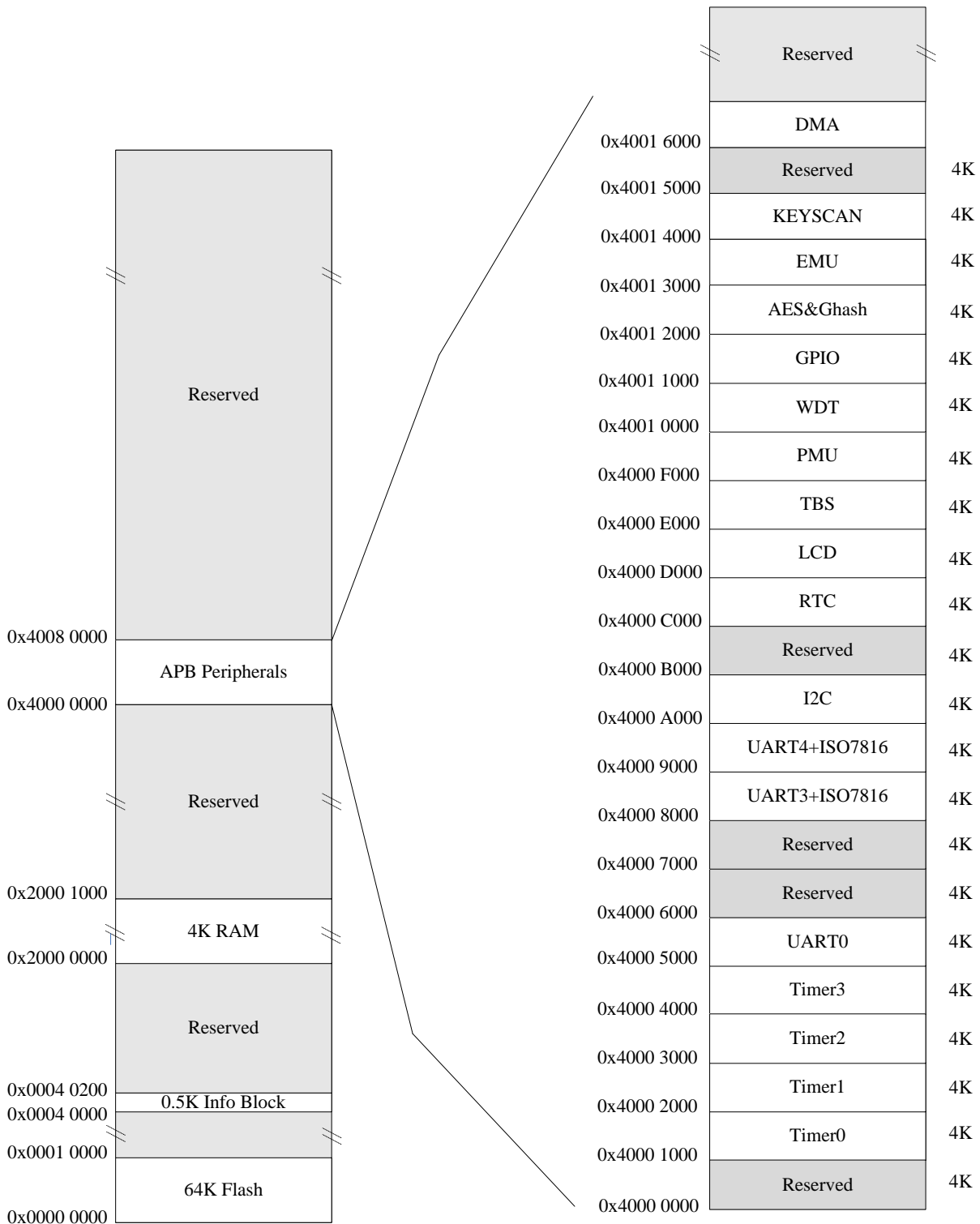
## 2 存储器模块

### 2.1 概述

HT500X 内置可编程高可靠 64K Flash+ 512 bytes InformationBlock 和 4K RAM。其中 Flash 具有读保护功能，可进行读、写、页擦除和全擦除操作，Flash 的特性如下：

- Flash 字节读取时间：40ns
- Flash 字节写时间：20us (max)
- Flash 页擦除时间：2ms (max)
- Flash 全擦除时间：10ms (max)
- Code Flash 页面大小：1K bytes/page
- Information Block: 256 bytes/page
- 擦写次数：100,000 次
- 数据保持时间：20 年 (min)
- 操作温度：-45 度到+105 度

## 2.2 存储器映射图



## 2.3 .Flash 操作

### 2.3.1. Flash 的读保护

Flash 空间具有读保护功能，可以防止用户代码被读取。

将 Flash 的 00000FC1H 地址写入非 0FFH 的值后，开启读保护功能，64K Flash 空间的数据无法读出，Information Block 中的数据依旧可以读出。（在线仿真时需要实时读取 Flash 内容，读保护下不能实现在线仿真）。

### 2.3.2. Code Flash 的操作说明

64K Code Flash 可以执行写/页擦除/全擦除操作，说明如下（伪代码举例，后同）。

推荐首先使用宏定义的方式来实现对 Memory 中的地址写入操作，HT500X 支持字节操作，半字操作，字操作，但用户需注意对齐访问，否则会触发一个 HardFault。

宏定义方式：

```
#define M8(adr)  *((uint8_t *) (adr))    //无需考虑对齐问题
#define M16(adr) *((uint16_t *) (adr))   //需半字对齐，即 adr 的 bit0 必须为 0
#define M32(adr) *((uint32_t *) (adr))   //需字对齐，即 adr 的 bit0 与 bit1 都必须为 0
以上宏定义实现对 Flash 地址 addr 的取址
```

对 64K Code Flash 的字节写操作流程：

```
WPREG = 0xA55A;
FLASHLOCK = 0x7A68;           //unlock flash memory

FLASHCON = 0x01;              //program
M32(prog_address) = prog_data; //prog_data 为需要编写的数据（32bit），
                                //prog_address 为需要写入的 flash 地址
//M16(prog_address) = prog_data; //prog_data 为需要编写的数据（16bit），
                                //prog_address 为需要写入的 flash 地址
//M8(prog_address) = prog_data;  //prog_data 为需要编写的数据（8bit），
                                //prog_address 为需要写入的 flash 地址
//当进行字(32bit)写入时， prog_address 如果要增加，需以 4 为单位递增
//当进行半字(16bit)写入时， prog_address 如果要增加，需以 2 为单位递增
//当进行字节 (8bit)写入时， prog_address 如果要增加，需以 1 为单位递增

while (FLASHCON.BUSY)         //等待 flash 写操作完成，最长 20us
;
```

对 64K Code Flash 的页擦除操作流程:

```
WPREG = 0xA55A;
FLASHLOCK = 0x7A68;           //unlock flash memory

FLASHCON = 0x02;             //page erase
M32(prog_address) = prog_data; //prog_data 可以为任意的数据 (32bit),
                                //prog_address 为需要擦除的 Flash 页内的任意一个地址

while (FLASHCON.BUSY);       //等待 flash 页擦除操作完成, 最长 2ms
```

对 64K Code Flash 的全擦除操作流程:

```
WPREG = 0xA55A;
FLASHLOCK = 0x7A68;           //unlock flash memory

FLASHCON = 0x03;             // mass erase
M32(prog_address) = prog_data; //prog_data 可以为任意的数据 (32bit),
                                // prog_address 为 64K Flash 的任意地址

while (FLASHCON.BUSY);       //等待 flash 全擦除操作完成, 最长 10ms
                                //全擦除会导致用户执行的代码全部被擦除掉
```

### 2.3.3. Information Block 的操作说明

Information Block 的写/页擦除/全擦除操作与 Code Flash 的写/擦除操作方式是一样的, 区别在于 Flash 地址不一样, 以及对 Information Block 操作还需要再配置一个解锁的寄存器, 如下所示:

```
FLASHLOCK = 0x7A68;           //unlock flash memory
INFOLOCK = 0xF998;           //unlock information Block memory
```

上面的两个解锁寄存器都需要配置, 用户才可以操作 Information Block, 其他操作同上 Code Flash 操作。

Information Block 共 512bytes (0x00040000~0x000401FF), 分 2 页, 256bytes/页, 其中第 2 页 (0x00040100~0x000401FF) 存储有芯片出厂信息, 请勿进行写/擦除操作。

## 2.4 Flash 控制功能

HT500X 微控制器中 Flash 存储器的 0FC0H~0FC3H 区域为 Flash 控制选项字节区域。当芯片打开电源或从复位状态重启时, 系统自动参考选项字节, 并按照其配置设置指定的功能。使用 HT500X 时, 必须使用选项字节设置以下几项功能。

- Flash 加密控制功能

- POR/LBOR复位RTC使能控制
- Info Block参数自动装载使能控制

HT500X Flash选项字控制作用如下：

Flash 控制功能说明			基地址： 偏移地址：					
Flash 地址	Bit7	6	5	4	3	2	1	Bit0
FC1H	FLASH[7:0]							
Reset:	1	1	1	1	1	1	1	1
Flash 地址	Bit7	6	5	4	3	2	1	Bit0
FC0H	X	X	X	X	RTCRST	AUTORELOAD	X	X
Reset:	1	1	1	1	0	0	1	0

位	功能描述
FLASH[7:0]	如果 Flash[7:0]=0xFF，则 Flash 不加密 其他：Flash 加密，该寄存器位只能读，不能写
RTCRST	=1, LBOR, POR 可以复位 RTC 计时寄存器 =0, LBOR, POR 不能复位 RTC 计时寄存器 <b>可以写</b>
AUTORELOAD	=1, 自动装载功能使能 =0, 自动装载功能屏蔽

注：其他位不可以修改，保持默认值。

## 2.5 写保护寄存器列表

被写保护的寄存器分布在 CMU 模块，PMU 模块，RTC 模块，详细列表如下：

CMU 模块寄存器基地址：0x4000F000			
偏移地址	名称	复位值	功能描述
0x00	WPREG	0x0000	写保护控制寄存器
0x04	SYSCLKCFG	0x0002	系统时钟配置寄存器（写保护）
0x08	OSCADJ	0x0001	OSC 时钟电流偏置调整寄存器（写保护）
0x0C	LRCADJ	0x0009	低频 RC 调整寄存器（写保护）
0x10	HRCADJ	0x0040	高频 RC 调整寄存器（写保护）
0x14	HRCDIV	0x0002	高频 RC 分频寄存器（写保护）
0x1C	SYSCLKDIV	0x0001	系统时钟分频寄存器（写保护）
0x24	CLKOUTSEL	0x0000	CLKOUT 时钟选择寄存器（写保护）
0x28	CLKOUTDIV	0x0000	CLKOUT 时钟分频寄存器（写保护）
0x2C	CLKCTRL0	0x04E0	内部模块使能寄存器 0（写保护）
0x30	CLKCTRL1	0x0000	内部模块使能寄存器 1（写保护）

0x34	FLASHCON	0x0000	Flash 访问控制寄存器（写保护）
0x3C	FLASHDLY	0x0000	Flash 延迟取指寄存器

### PMU 模块寄存器基地址：0x4000F400

偏移地址	名称	复位值	功能描述
0x00	PMUCON	0x0017	PMU配置寄存器（写保护）

### RTC 模块寄存器基地址：0x4000C000

偏移地址	名称	复位值	功能描述
0x18	SECR	0x0000	秒寄存器（写保护）
0x1C	MINR	0x0000	分寄存器（写保护）
0x20	HOURR	0x0000	时寄存器（写保护）
0x24	DAYR	0x0001	日寄存器（写保护）
0x28	MONTHR	0x0001	月寄存器（写保护）
0x2C	YEARR	0x0000	年寄存器（写保护）
0x30	WEEKR	0x0001	周寄存器（写保护）
0x200	SECR2	0x0000	秒寄存器（写保护）
0x204	MINR2	0x0000	分钟寄存器（写保护）
0x208	HOURR2	0x0000	小时寄存器（写保护）
0x20C	DAYR2	0x0001	天寄存器（写保护）
0x210	MONTHR2	0x0001	月寄存器（写保护）
0x214	YRR2	0x0000	年寄存器（写保护）
0x218	WEEKR2	0x0001	星期寄存器（写保护）

## 2.6 特殊功能寄存器列表

### CMU模块寄存器基地址：0x4000F000

偏移地址	名称	读写方式	复位值	功能描述
0x00	WPREG	R/W	0x0000	写保护控制寄存器
0x34	FLASHCON	R/W	0x00	Flash 控制寄存器（写保护）
0x38	FLASHLOCK	R/W	0x0000	Flash 锁定寄存器
0x50	INFOLOCK	R/W	0x0000	Information Block 锁定寄存器

## 2.7 特殊功能寄存器说明

WPREG (写保护寄存器)	基地址： 0x4000F000 偏移地址： 00H
-------------------	------------------------------



	Bit15	14	13	12	11	10	9	Bit8
Read:	WPREG[15:8]							
Write:	WPREG[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	WPREG[7:0]							
Write:	WPREG[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
WPREG[15:0]	<ol style="list-style-type: none"> <li>WPREG 写入 0xA55A, 则关闭写保护功能, 用户可以写操作被保护的寄存器。</li> <li>WPREG 写非 0xA55A, 则开启写保护功能, 用户禁止写操作被保护的寄存器。</li> <li>读该寄存器: 0x0001: 表示写保护关闭, 用户可以写操作被保护的寄存器 0x0000: 表示写保护开启, 用户禁止写操作被保护的寄存器</li> </ol>

FLASHCON (写保护) (Flash 控制寄存器)		基地址: 0x4000F000 偏移地址: 34H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	BUSY	FOP[1:0]	
Write:	X	X	X	X	X	X	FOP[1:0]	
Reset:	0	0	0	0	0	0	0	0

位	功能描述															
BUSY	<b>FLASH 忙标志位</b> 0: 表示 Flash 空闲, 可以进行操作。 1: 表示 Flash 正在进行写/擦除操作。 只读状态寄存器位, 写入无效。															
FOP[1:0]	<b>FLASH 操作模式选择</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>FOP1</th> <th>FOP0</th> <th>FLASH 操作</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>处于 Flash 只读模式</td> </tr> <tr> <td>0</td> <td>1</td> <td>对 STR/STRH 所指 FLASH 区执行 Flash 写操作</td> </tr> <tr> <td>1</td> <td>0</td> <td>对 STR/STRH 所指 FLASH 区执行 Flash 页擦除操作</td> </tr> <tr> <td>1</td> <td>1</td> <td>对 STR/STRH 所指 FLASH 区执行 Flash 全擦除操作</td> </tr> </tbody> </table>	FOP1	FOP0	FLASH 操作	0	0	处于 Flash 只读模式	0	1	对 STR/STRH 所指 FLASH 区执行 Flash 写操作	1	0	对 STR/STRH 所指 FLASH 区执行 Flash 页擦除操作	1	1	对 STR/STRH 所指 FLASH 区执行 Flash 全擦除操作
FOP1	FOP0	FLASH 操作														
0	0	处于 Flash 只读模式														
0	1	对 STR/STRH 所指 FLASH 区执行 Flash 写操作														
1	0	对 STR/STRH 所指 FLASH 区执行 Flash 页擦除操作														
1	1	对 STR/STRH 所指 FLASH 区执行 Flash 全擦除操作														

FLASHLOCK (Flash 锁定寄存器)		基地址: 0x4000F000 偏移地址: 38H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	KEY[15:8]							
Write:	KEY[15:8]							

Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	KEY[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
KEY[15:0]	<p><b>Flash 锁定控制位</b></p> <p>对该寄存器写入 0x7A68 后， FLASH 被解锁，用户可以写操作 FLASH。 写入非 0x7A68 数据后， FLASH 被锁定，用户禁止写操作 FLASH。 默认为锁定状态，Flash 不可执行写/页擦除/全擦除 操作 用户写入的是 0x7A68, 读出值为 1；写入的是非 0x7A68, 读出值为 0</p>

INFOLOCK (InfoFlash 锁定寄存器)		基地址: 0x4000F000 偏移地址: 50H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	KEY[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	KEY[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
KEY[15:0]	<p><b>Information Block 锁定控制</b></p> <p>对该寄存器写入 0xF998 后， Information Block 被解锁，用户可以写操作 Information Block。 写入非 0xF998 数据后， Information Block 被锁定，用户禁止写操作 Information Block。 默认为锁定状态，InformationBlock 不可执行写/页擦除/全擦除 操作 用户写入的是 0xF998, 读出值为 1；写入的是非 0xF998, 读出值为 0</p>

## 3 时钟单元

### 3.1 时钟分类

测试温度范围：-40℃~85℃

名称	频率	精度	功耗		
			MIN	TYP	MAX
内部低频 RC 时钟 (Flrc)	Min: 13KHz Max: 50KHz			1uA	
内部高频 RC 时钟 (Fhrc)	9.5MHz	+/- 3%		114uA	
外部低频 OSC 晶振 (Fosc)	32768Hz				1uA
内部 PLL (Fpll)	19.660800MHz			200uA	

### 3.2 时钟框图

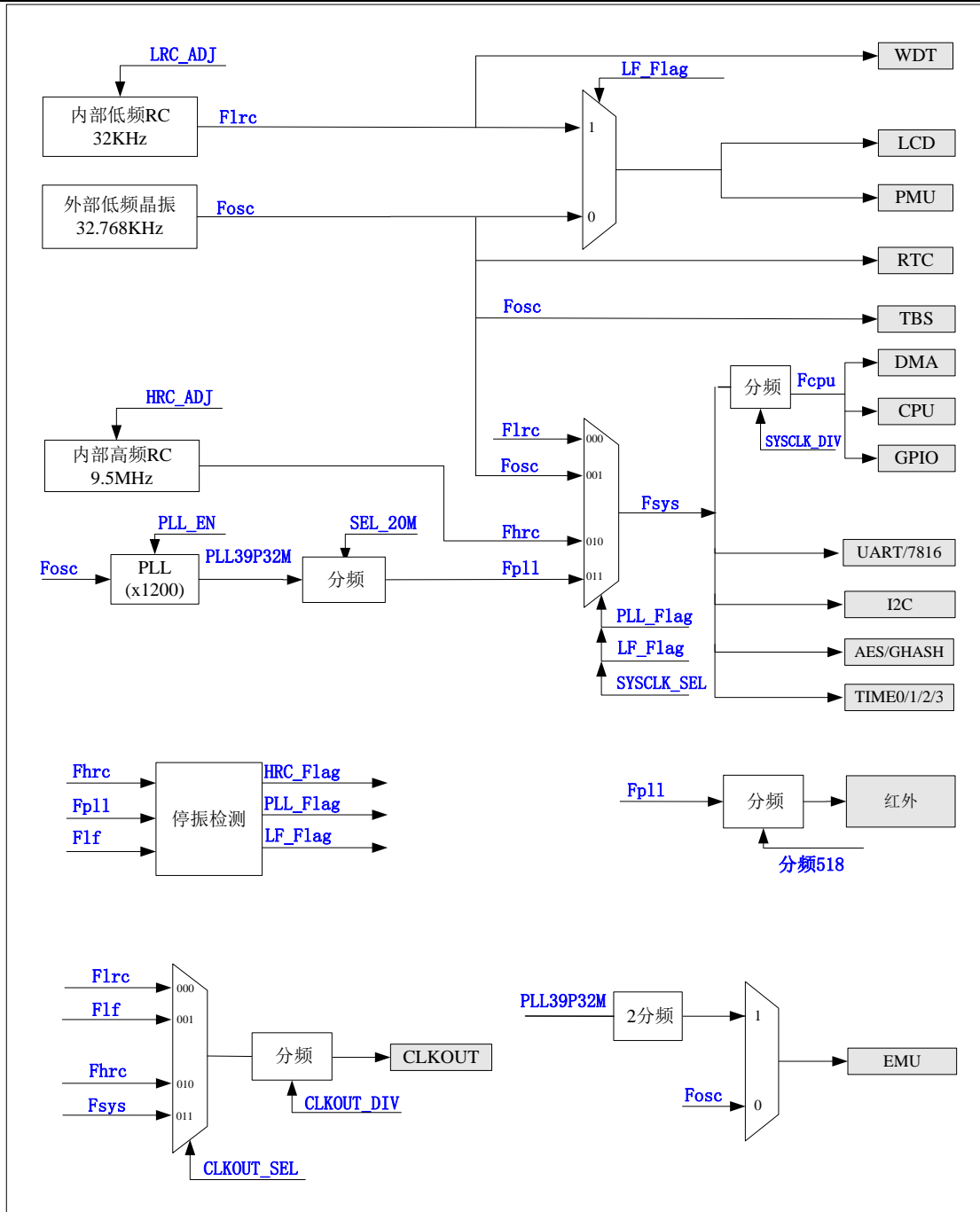
时钟符号说明：

Flrc: 内部低频 RC 时钟 (32768Hz)，也作为看门狗时钟源。

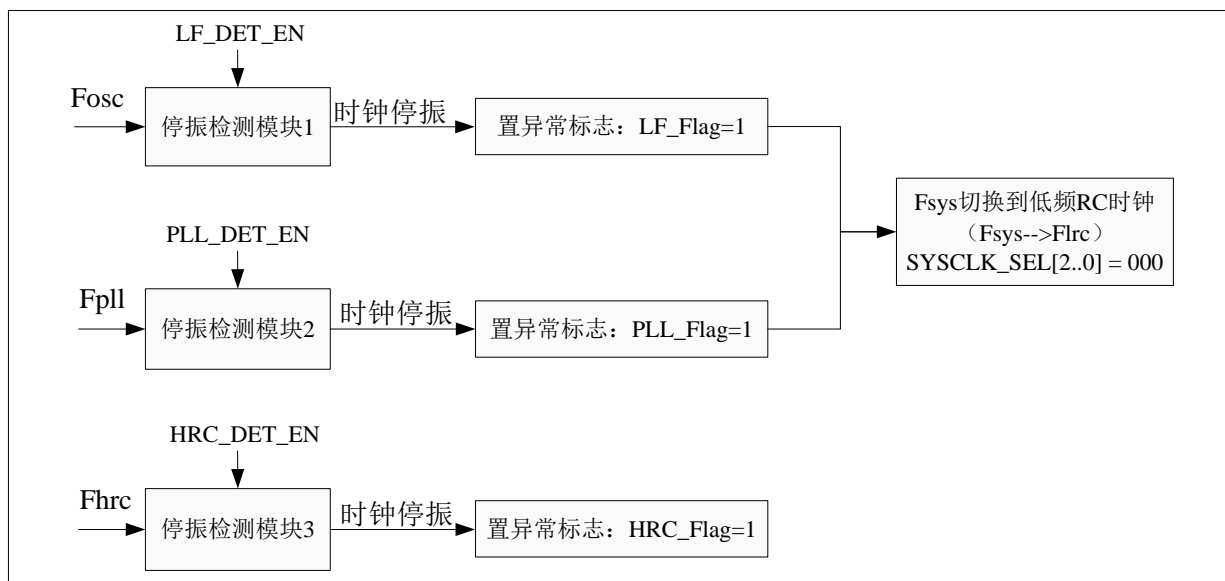
Fhrc: 内部高频RC时钟 (9.5MHz)。

Fosc: 外部低频OSC晶振时钟 (32768Hz)。

Fpll: 内部PLL产生的高频时钟 (39.321600MHz)，来源为Fosc。



### 3.3 时钟停振检测框图



### 3.4 时钟说明

#### 3.4.1 内部低频 RC 时钟 (Flrc)

内部低频 RC 时钟振荡频率为 32768Hz，提供给看门狗使用，可以选择该低频 RC 时钟作为系统时钟 (SYSCLK\_SEL[2:0]=000)。

内部低频 RC 时钟可以通过寄存器 LRCADJ 校准，以提高钟精度。

在 Sleep 或 Hold 模式下，内部低频 RC 时钟可由用户软件关闭，控制位为 ControlByFlash 寄存器的 LRC\_CTRL 位。

#### 3.4.2 内部高频 RC 时钟 (Fhrc)

内部高频 RC 时钟频率为 9.5MHz，可以选择该高频 RC 时钟的分频输出时钟作为系统时钟 (SYSCLK\_SEL[2:0]=010)。

内部高频 RC 时钟通过寄存器 HRCADJ 校准后，精度可以达到 1%。

内部高频 RC 时钟全温度范围温度采样误差小于 1%。

系统复位后，系统时钟默认选择内部高频 RC 时钟 (SYSCLK\_SEL[2:0]=010)。

选择内部高频 RC 作为系统时钟时，不能关闭内部高频 RC 时钟，对 HRC\_EN 写“0”操作无效。

#### 外部低频晶振时钟 (Fosc)

芯片外接 32K 低功耗晶体振荡器，时钟频率 Fosc=32768Hz，芯片内部集成了其震荡所需的电阻和电容，匹配电容大约为 12.5pF。

### 3.4.3 内部 PLL 时钟 (F<sub>pll</sub>)

内部 PLL 用于对外部低频晶振时钟 F<sub>osc</sub> (32768Hz) 倍频 (倍频值=1200), 以对系统提供最高达 39.321600MHz 的高频时钟 F<sub>pll</sub>, PLL 稳定输出后会给出稳定标志位 PLL\_LOCK。

### 3.4.4 时钟安全机制

芯片内部集成有 3 个独立的时钟停振检测模块, 分别对外部低频晶振时钟 F<sub>osc</sub>, PLL 输出时钟 F<sub>pll</sub>, 和内部高速时钟 F<sub>hrc</sub> 作检测。时钟停振检测功能默认开启, 可以由用户软件关闭, 控制位分别为 LF\_DET\_EN, PLL\_DET\_EN, HRC\_DET\_EN。

时钟停振检测模块的时钟源为内部低频 RC 时钟 F<sub>lrc</sub>。

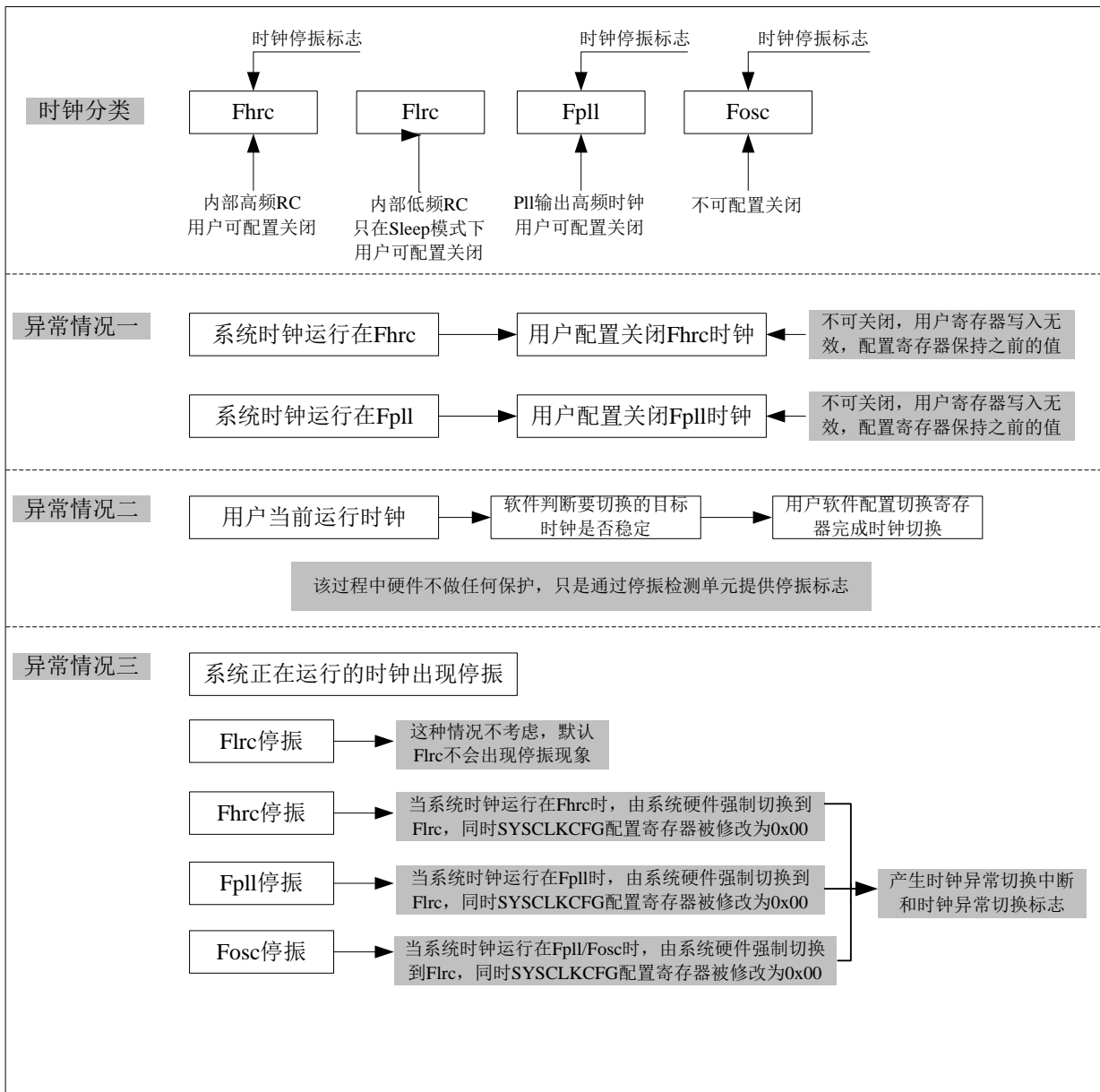
当对应的的时钟停振检测模块功能开启时, 外部低频晶振时钟 F<sub>osc</sub> 发生停振, PLL 时钟 F<sub>pll</sub> 发生停振, 或内部高频 RC 时钟 F<sub>hrc</sub> 发生停振, 都会产生相应的时钟故障标志 (LF\_FLAG, PLL\_FLAG, HRC\_FLAG)。

当停振检测模块检测到 F<sub>osc</sub> 停振, 系统给出时钟停振标志 LF\_FLAG, 如系统时钟 F<sub>sys</sub> 选择 F<sub>osc</sub> 或 F<sub>pll</sub> (F<sub>osc</sub> 为 F<sub>pll</sub> 的时钟源) 时, 系统会由硬件强制系统时钟 F<sub>sys</sub> 切换到内部低频 RC 时钟 F<sub>lrc</sub>, 且产生中断 (NMI 中断), 同时将寄存器 SYSCLK\_SEL[2:0] 的值置为 000。

当停振检测模块检测到 F<sub>pll</sub> 停振, 系统给出时钟停振标志 PLL\_FLAG, 如系统时钟 F<sub>sys</sub> 选择 F<sub>pll</sub> 时, 系统会由硬件强制系统时钟 F<sub>sys</sub> 切换到内部低频 RC 时钟 F<sub>lrc</sub>, 且产生中断 (NMI 中断), 同时将寄存器 SYSCLK\_SEL[2:0] 的值置为 000。

当停振检测模块检测到 F<sub>hrc</sub> 停振, 系统给出标志位 HRC\_FLAG, 如系统时钟 F<sub>sys</sub> 选择 F<sub>hrc</sub> 时, 系统不会由硬件强制切换系统时钟, 此时系统将停止运行, 而等待看门狗复位。

## 3.4.5 时钟异常状态处理



1. 系统运行于低频时钟 Fosc 时，如 PLL 停振，芯片硬件不做任何处理，仅给出停振标志位 PLL\_FLAG。
2. 系统运行于低频时钟 Fosc 时，如 Fosc 的时钟源停振，系统时钟由硬件强制切换到 Flrc，同时产生 NMI 中断。
3. 系统运行于 PLL 时钟 Fpll 时，如 Fosc 停振或 Fpll 停振，系统时钟由硬件强制切换到 Flrc，同时产生 NMI 中断。

### 3.5 特殊功能寄存器列表

CMU模块寄存器基地址: 0x4000F000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	WPREG	R/W	0x0000	写保护控制寄存器
0x04	SYCLKCFG	R/W	0x0002	系统时钟配置寄存器 (写保护)
0x08	JTAGSTA	R/W	0x0001	JTAG 状态寄存器 (写保护)
0x0C	LRCADJ	R/W	0x0009	LRC 时钟调整寄存器 (写保护)
0x10	HRCADJ	R/W	0x0040	HRC 时钟调整寄存器 (写保护)
0x18	CLKSTA	R	0x0000	时钟状态寄存器 (只读)
0x1C	SYCLKDIV	R/W	0x0001	系统时钟分频寄存器 (写保护)
0x24	CLKOUTSEL	R/W	0x0002	CLKOUT 时钟选择寄存器 (写保护)
0x28	CLKOUTDIV	R/W	0x0000	CLKOUT 时钟分频寄存器 (写保护)
0x2C	CLKCTRL0	R/W	0x24E0	内部模块使能寄存器 0 (写保护)
0x30	CLKCTRL1	R/W	0x0000	内部模块使能寄存器 1 (写保护)
0x34	FLASHCON	R	<b>0x00</b>	Flash 控制寄存器 (写保护)
0x38	FLASHLOCK	W	0x0000	Flash 锁定寄存器
0x3C	FLASHDLY	R/W	0x0000	<b>Flash 延迟取指寄存器</b>
0xF00	CHIPID	R		芯片版本寄存器, 只读

### 3.6 特殊功能寄存器说明

WPREG (写保护寄存器)			基地址: 0x4000F000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	WPREG[15:8]							
Write:	WPREG[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	WPREG[7:0]							
Write:	WPREG[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
WPREG[15:0]	4. WPREG 写入 0xA55A, 则关闭写保护功能, 用户可以写操作被保护的寄存器。 5. WPREG 写非 0xA55A, 则开启写保护功能, 用户禁止写操作被保护的寄存器。 6. 读该寄存器: 0x0001: 表示写保护关闭, 用户可以写操作被保护的寄存器 0x0000: 表示写保护开启, 用户禁止写操作被保护的寄存器



SYSCLKCFG (写保护) (系统时钟配置寄存器)			基地址: 0x4000F000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	WCLKEN	X	X	X	X	SYSCLK_SEL[2:0]		
Write:								
Reset:	0	0	0	0	0	0	1	0

位	功能描述																				
WCLKEN	<p><b>时钟配置寄存器写保护位</b></p> <p>如果用户要更改系统时钟选择, 必须同时将 WCLKEN 位置 1, 例如: ‘b1xxxxxxx’, 才可以对系统时钟选择位 SYSCLK_SEL[2:0] 进行写操作。</p>																				
SYSCLK_SEL[2:0]	<p><b>系统时钟选择控制位:</b></p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="3">SYSCLK_SEL[2:0]</th> <th>系统时钟选择Fsys</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Flrc</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Fosc</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Fhrc (Default)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Fpll</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>系统时钟可作为芯片内部硬件外设模块的时钟源, 经分频后也可作为 CPU 和 GPIO 的时钟源。</li> <li>复位后, 系统时钟默认为内部高频 RC 时钟 Fhrc (SYSCLK_SEL[2:0]=010)。</li> <li>当选择外部低频晶振时钟 Fosc 作为系统时钟时, 检测到低频时钟 Fosc 发生停振, 此时系统会由硬件强制将 Fsys 切换到内部低速 RC 时钟 Flrc, 同时系统时钟控制位 SYSCLK_CLK[2:0] 被置为 000</li> <li>当选择 PLL 输出时钟 Fpll 作为系统时钟时, 检测到低频时钟 Fosc 或者 PLL 发生停振, 此时系统会由硬件强制将 Fsys 切换到内部低速 RC 时钟 Flrc, 同时系统时钟控制位 SYSCLK_CLK[2:0] 被置为 000。</li> </ol>	SYSCLK_SEL[2:0]			系统时钟选择Fsys	0	0	0	Flrc	0	0	1	Fosc	0	1	0	Fhrc (Default)	0	1	1	Fpll
SYSCLK_SEL[2:0]			系统时钟选择Fsys																		
0	0	0	Flrc																		
0	0	1	Fosc																		
0	1	0	Fhrc (Default)																		
0	1	1	Fpll																		

JTAGSTA (芯片状态指示寄存器)			基地址: 0x4000F000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	FLAG	X
Write:								
Reset:	0	0	0	0	0	0	0	1

位	功能描述
FLAG	该位用于指示芯片是否处于调试状态 0: 表示芯片处于正常运行状态。 1: 表示芯片处于调试状态。

LRCADJ (写保护) (LRC 时钟调整寄存器)		基地址: 0x4000F000 偏移地址: 0CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	LRC_ADJ[3:0]			
Write:								
Reset:	0	0	0	0	1	0	0	1

位	功能描述
LRC_ADJ[3:0]	LRC 输出频率调节控制位

HRCADJ (写保护) (HRC 时钟调整寄存器)		基地址: 0x4000F000 偏移地址: 10H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	HRC_ADJ[5:0]					
Write:								
Reset:	0	1	0	0	0	0	0	0

位	功能描述
HRC_ADJ[5:0]	HRC 输出频率调节控制位

CLKSTA (时钟状态寄存器)		基地址: 0x4000F000 偏移地址: 18H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								

	0	0	0	0	0	0	0	0
<b>Reset</b>	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	X	X	PLL_LOCK	PLL_FLAG	HRC_FLAG	X	X	LF_FLAG
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset</b>	0	0	0	0	0	0	0	0

注：此寄存器是只读状态寄存器

位	功能描述
PLL_LOCK	<b>PLL 时钟锁定状态标志寄存器</b> 0: PLL 时钟锁定异常。 1: PLL 时钟锁定正常。 <b>注：用来指示芯片内部 PLL 的工作稳定状态，调试模式下，PLL_LOCK 固定为 1</b>
PLL_FLAG	<b>PLL 时钟 Fp11 停振标志</b> 0: 正常。 1: 停振。
HRC_FLAG	<b>内部高频 RC 时钟 Fhrc 停振标志</b> 0: 正常。 1: 停振。
LF_FLAG	<b>外部低频晶振时钟 Fosc 停振标志</b> 0: 正常 1: 停振 （在 SLEEP 或 Hold 模式下，如果内部低频 RC 时钟 Flrc 关闭，外部低频晶振时钟 Fosc 停振检测模块 LF_DET 默认输出为检测 LF 时钟正常，即 LF_FLAG =0；只有当系统时钟 Fsys 选择为外部低频晶振时钟 Fosc 或者 PLL 输出时钟 Fp11 下时，如停振检测模块检测到相应时钟停振，系统时钟会强制切换到内部低频 RC 时钟 Flrc，同时修改寄存器 SYSCLK_SEL[2:0]的值为 000。）

<b>SYSCLKDIV (写保护)</b>			<b>基地址: 0x4000F000</b>					
<b>(系统时钟分频寄存器)</b>			<b>偏移地址: 1CH</b>					
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	X	X	X	X	X	X	X	X
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	X	X	X	X	SEL_20M	SYSCLK_DIV[2:0]		
<b>Write:</b>	X	X	X	X	SEL_20M	SYSCLK_DIV[2:0]		
<b>Reset:</b>	0	0	0	0	1	0	0	1

位	功能描述
SEL_20M	<b>PLL输出时钟预分频寄存器控制位：</b> =0: PLL 输出 19.66MHz 时钟 =1: PLL 输出 39.32MHz 时钟

SYSCLK_DIV[2:0]	系统时钟分频得到 Fcpu, 设置如下:						
	SYSCLK_DIV [2:0]			内核时钟 (Fcpu)			
	0	0	0	Fsys			
	0	0	1	Fsys/2(Default)			
	0	1	0	Fsys/4			
	0	1	1	Fsys/8			
	1	0	0	Fsys/16			
	1	0	1	Fsys/32			
	1	1	0	Fsys/64			
1	1	1	Fsys/128				

CLKOUTSEL (写保护) (CLKOUT 时钟选择寄存器)			基地址: 0x4000F000 偏移地址: 24H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	CLKOUT_SEL[2:0]		
Write:								
Reset:	0	0	0	0	0	0	1	0

位	功能描述							
CLKOUT_SEL[2:0]	CLKOUT 时钟输出引脚配置							
	CLKOUT_SEL[2:0]			CLKOUT时钟选择				
	0	0	0	Flrc				
	0	0	1	Fosc				
	0	1	0	Fhrc(Default)				
	0	1	1	Fsys				
	1	X	X	Fosc				
1, 用户可将芯片内部时钟源从 CLKOUT 引脚引出, 以观测内部时钟。 2, 用户可用 CLKOUTDIV 寄存器将内部时钟分频后引出, 可作为外部设备的时钟源。								

CLKOUTDIV (写保护) (CLKOUT 时钟分频寄存器)			基地址: 0x4000F000 偏移地址: 28H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X

Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	CLKOUT_DIV[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CLKOUT_DIV[3:0]	$\text{CLKOUT 输出频率} = \frac{\text{CLKOUT 选择的时钟源}}{2 \times (\text{CLKOUT\_DIV}[3..0] + 1)}$

CLKCTRL0 (写保护) (内部模块使能控制寄存器 0)			基地址: 0x4000F000 偏移地址: 2CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	EMUEN	AES_EN	1P5LBOR_EN	CLKOUT_EN	KEYEN	OSC_SLP	HRC_DET_EN	PLL_DET_EN
Write:								
Reset:	0	0	1	0	0	1	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	LF_DET_EN	PLL_LOK_EN	HRC_EN	PLL_EN	I2C_EN	X	LCD_EN	X
Write:								
Reset:	1	1	1	0	0	0	0	0

位	功能描述
EMUEN	<b>EMU 模块使能信号</b> 0: 关闭 (default) 1: 打开
AES_EN	<b>AES 模块使能信号</b> 0: 关闭 (default) 1: 打开
1P5LBOR_EN	<b>低功耗 LDO_1P5 内部 LBOR 使能信号</b> 0: 关闭 1: 打开 (default) 建议保留默认配置。
CLKOUT_EN	<b>CLKOUT 使能信号</b> 0: 关闭 (default) 1: 打开
KEY_EN	<b>按键扫描功能使能</b> 0: 禁止 1: 使能
OSC_SLP	<b>OSC 低功耗使能控制信号</b> 0: 大功耗 1: 小功耗 (default)

HRC_DET_EN	<b>HRC 时钟检测模块控制信号</b> 0: 关闭 HRC 时钟停振检测单元 (default) 1: 使能 HRC 时钟停振检测单元
PLL_DET_EN	<b>PLL 时钟检测模块控制信号</b> 0: 关闭 PLL 时钟停振检测单元 (default) 1: 使能 PLL 时钟停振检测单元
LF_DET_EN	<b>LF 时钟检测模块控制信号</b> 0: 关闭 LF 时钟停振检测单元 1: 使能 LF 时钟停振检测单元 (default)
PLL_LOCK_EN	<b>PLL 强制锁定使能控制位</b> 0: PLL 不强制锁定 1: PLL 强制锁定 (default) 注意: 当使能该位时, 相应的 PLL 锁定标志位 PLL_LOCK 固定为 1, 反之锁定标志位 PLL_LOCK 将反应 PLL 的实际工作状态。
HRC_EN	<b>HF RC 时钟振荡器使能位</b> 0: 关闭高频 RC 时钟模块; 1: 使能高频 RC 时钟模块; (default) 注意: 当用户选择 Fsys 为 Fhrc 时, 此时不能关闭 HRC_EN, 该寄存器位写入无效
PLL_EN	<b>PLL 模块时钟使能位</b> 0: 关闭 PLL 模块 (default) 1: 使能 PLL 模块 注意: 当用户选择 Fsys 为 Fp11 时, 此时不能关闭 PLL_EN, 该寄存器位写入无效
I2C_EN	<b>I2C 模块时钟使能位</b> 0: 关闭 I2C 模块 (default) 1: 使能 I2C 模块
LCD_EN	<b>LCD 模块时钟使能位</b> 0: 关闭 LCD 模块 (default) 1: 使能 LCD 模块

<b>CLKCTRL1 (写保护)</b>		<b>基地址: 0x4000F000</b>						
<b>(内部模块使能控制寄存器 1)</b>		<b>偏移地址: 30H</b>						
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	LRC_RTC	LRC_TFR	X	X	X	X	X	UART4_7
<b>Write:</b>	2_EN	EQ_RTC2						816_EN
<b>Reset:</b>	0	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	UART3_7	X	X	UART0_EN	TMR3_EN	TMR2_EN	TMR1_EN	TMRO_EN
<b>Write:</b>	816_EN			N				
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>位</b>	<b>功能描述</b>
----------	-------------

LRC_RTC2_EN	<b>辅助 RTC 的工作使能位</b> 0: 关闭 1: 开启
LRC_TFREQ_RTC2	<b>辅助 RTC 的测频使能位</b> 0: 关闭 1: 开启
UART4_7816_EN	<b>UART4 时钟使能位</b> 0: 关闭 1: 使能
UART3_7816_EN	<b>UART3 时钟使能位</b> 0: 关闭; 1: 使能
UART0_EN	<b>UART0 时钟使能位</b> 0: 关闭 1: 使能
TMR3_EN	<b>Timer3 时钟使能位</b> 0: 关闭 1: 使能
TMR2_EN	<b>Timer2 时钟使能位</b> 0: 关闭 1: 使能
TMR1_EN	<b>Timer1 时钟使能位</b> 0: 关闭 1: 使能
TMRO_EN	<b>Timer0 时钟使能位</b> 0: 关闭 1: 使能

FLASHCON (写保护) (Flash 控制寄存器)			基地址: 0x4000F000 偏移地址: 34H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	BUSY	FOP[1:0]	
Write:						X		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
BUSY	<b>FLASH 忙标志位</b> 0: 表示 Flash 空闲, 可以进行操作。 1: 表示 Flash 正在进行写/擦除操作。 只读状态寄存器位, 写入无效。

FOP[1:0]	FLASH 操作模式选择		
	FOP1	FOP0	FLASH 操作
	0	0	处于 Flash 只读模式
	0	1	对 STR/STRH 所指 FLASH 区执行 Flash 写操作
	1	0	对 STR/STRH 所指 FLASH 区执行 Flash 页擦除操作
	1	1	对 STR/STRH 所指 FLASH 区执行 Flash 全擦除操作

FLASHLOCK (Flash 锁定寄存器)		基地址: 0x4000F000 偏移地址: 38H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	KEY[15:8]							
Write:	KEY[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	KEY[7:0]							
Write:	KEY[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
KEY[15:0]	<p><b>Flash 锁定控制位</b></p> <p>对该寄存器写入 0x7A68 后, FLASH 被解锁, 用户可以写操作 FLASH。 写入非 0x7A68 数据后, FLASH 被锁定, 用户禁止写操作 FLASH。 默认为锁定状态, Flash 不可执行写/页擦除/全擦除 操作 该寄存器不可读, 只能写入</p>

FLASH_DLY (写保护) (Flash 控制寄存器)		基地址: 0x4000F000 偏移地址: 3CH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	FLASH_DLY
Write:	X	X	X	X	X	X	X	FLASH_DLY
Reset:	0	0	0	0	0	0	0	0

位	功能描述
FLASH_DLY	<p>系统时钟在 39.32M 时, 需要配置该寄存器位为 1</p> <p>0: 在 Flash 中取指令时不等待 1: 在 Flash 中取指令时等待 (PLL 输出 39.32M 时使用)</p>

CHIPID 芯片版本寄存器		基地址: 0x4000F000 偏移地址: F00H						
	Bit15	14	13	12	11	10	9	Bit8



Read:	CHIPID[15...8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CHIPID[7...0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CHIPID[15...0]	芯片版本号: 0x5001

## 4 电源单元

### 4.1 概述

PMU 为芯片的电源管理单元，功能如下：

- 工作电压范围：2.2V~5.5V
- 内部比较器 VSYS\_DET 监测系统电源 VSYS, LVDIN\_DET 检测外部低电压检测输入端 LVDINO 和 LVDIN1，当供电电压低于或高于设定阈值时产生中断信号。
- 自动完成 VSYS 电源和 VBAT 电源的切换，电源状态在 PMSR 寄存器中的状态位实时更新。
- 监测芯片内部的工作电源 VDD，可以根据设定阈值产生中断信号
- 监测芯片内部的工作电源 VDD，可以根据设定阈值产生 BOR、LBOR、POR 复位信号
- 为计量模块提供 2.8V 电源
- 测量芯片引脚 ADCINO, ADCIN1, ADCIN2 的电压
- 为芯片内部数字模块提供 1.5V 电源



## 4.3 电源单元详细功能说明

### 4.3.1 电源切换

外部供电电源的切换由芯片内部硬件自动完成。系统上电后，VSYS\_DET模块将一直开启。VSYS\_DET模块监测外部供电的VSYS电压，根据VSYS的电压状态，自动完成系统电源VSYS和电池电源VBAT之间的切换；同时将当前VSYS的电压状态更新到寄存器PMUSTA的状态位VSYS\_FLG。

当电池电源VBAT电压大于1.65V，VSYS\_DET模块监测到VSYS的电压低于设定阈值时，切换到VBAT供电，即VDD连接到VBAT；当电池电源VBAT电压大于1.65V，VSYS\_DET模块监测到VSYS的电压高于设定阈值时，切换到VSYS供电，即VDD连接到VSYS。

当电池电源VBAT电压小于1.65V，则系统一直保持VSYS供电，不受VSYS\_DET模块影响。

### 4.3.2 电源实时监测

PMU单元共内置5个电源检测模块，分别实时监测工作电源状态，并将监测结果以三个中断信号和三个复位信号的形式反馈给用户。

- **LVD\_DET模块：**

监测外部引脚LVDIN0或者LVDIN1的电压，当电压低于或高于1.21V时，置位LVDIN0IF/ LVDIN1IF标志位；如果使能了LVDIN0IE/ LVDIN1IE中断，就会产生LVDIN中断。

- **VSYS\_DET模块：**

监测系统电源VSYS的电压，当电压低于或高于设定阈值时，置位VSYSIF标志位；如果使能VSYS检测中断（VSYSIE），就会产生VSYS中断；检测阈值可通过寄存器VDETCFG的位VSYS\_LVL[3:0]来设置。

- **BOR\_DET模块：**

监测系统电源VDD的电压，当电压低于或高于设定阈值时，置位BORIF标志位；如果设置了BOR复位（BORRST=1），将会立即产生BOR复位；如果没有设置BOR复位（BORRST=0），但使能了BOR检测中断（BORIE），就会产生BOR中断；检测阈值可通过寄存器VDETCFG的位BOR\_LVL[1:0]来设置。

- **LBOR\_DET模块：**

监测系统电源VDD的电压，当电压低于阈值1.9V时，产生LBOR掉电复位。

- **POR\_DET模块：**

监测系统电源VDD的电压，当电压从低于阈值电压0.3V到上升到阈值电压以上时，产生POR上电复位。

### 4.3.3 内建 1.5V 电源

芯片内部通过子模块 VREG 将 VDD 电压调制成 1.5V 电压，供芯片内部的 1.5V 工作域使用，并通过引脚 VDD 输出。在输出引脚 VDD 上需要外接 0.1uF 电容，以协助芯片提供稳定的 1.5V 内部数字电源。

## 4.3.4 BOR 检测功能(BOR\_DET)

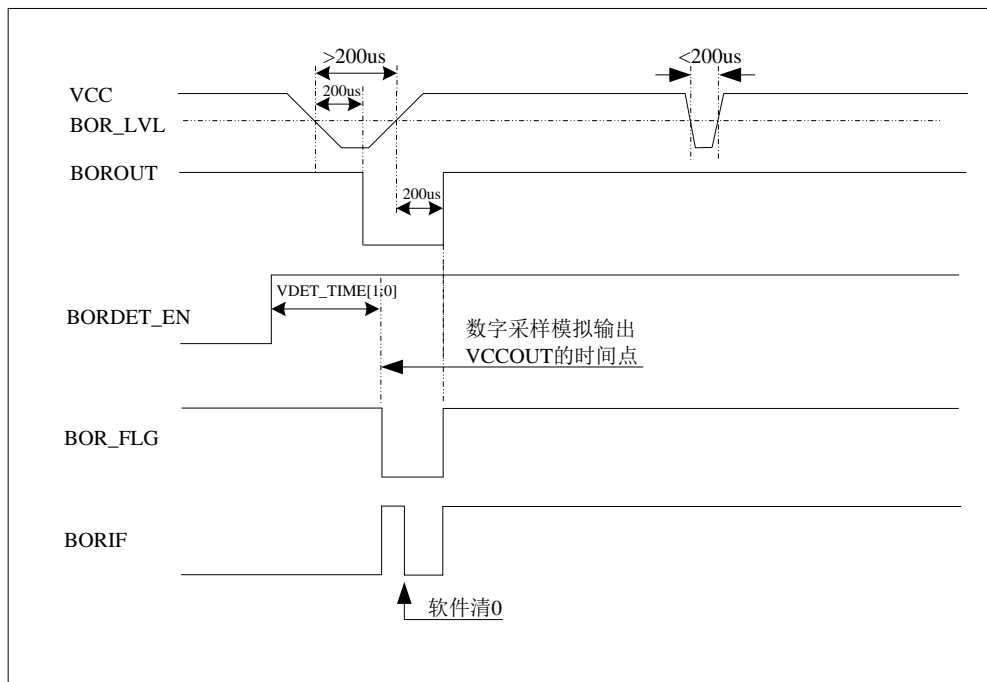
当 BOR 检测模块检测到系统电源 VDD 低于设定电压  $V_{borl}$  时, BOR 检测模块内部信号 BOROUT 输出低电平, 内部复位信号 IRST 也将变为低电平, 复位状态寄存器 RSTSR 的 BOR 标志位被置为 1。当掉电检测电路检测到系统电源 VDD 电压高于设定电压  $V_{borh}$  时, BOR 检测模块内部信号 BOROUT 输出高电平, 在该高电平持续的 1088 个  $F_{lrc}/F_{osc}$  周期后, 内部复位信号 IRST 也变为高电平。

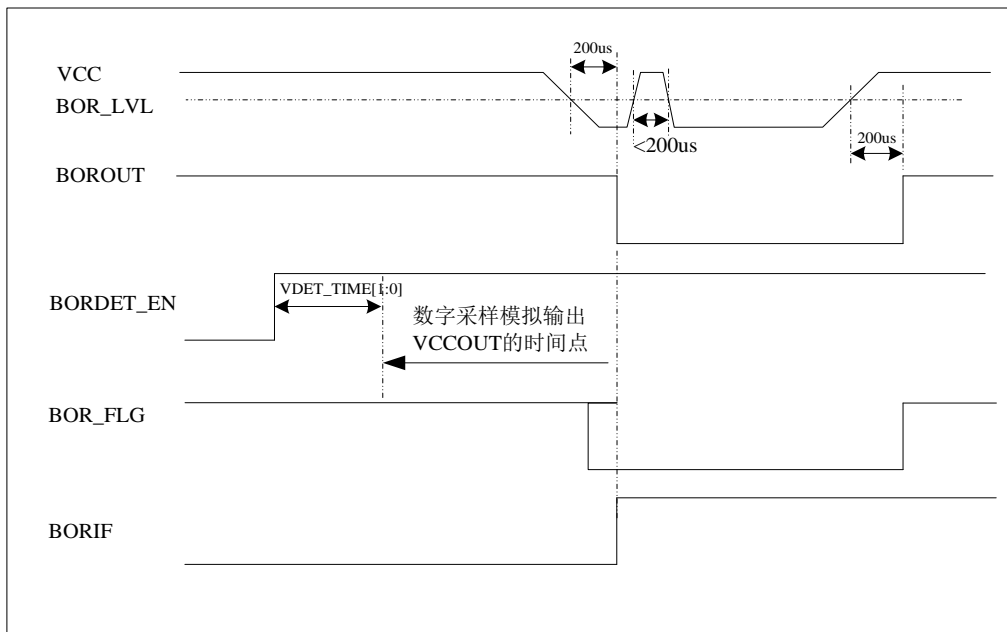
$V_{borh}$  和  $V_{borl}$  之间的迟滞为 200mV, BOR 模块的检测阈值可通过 VDETCFG 中的 BOR\_LVL[1:0] 设置。

掉电复位 BOR 产生时, 下面的事件将会发生:

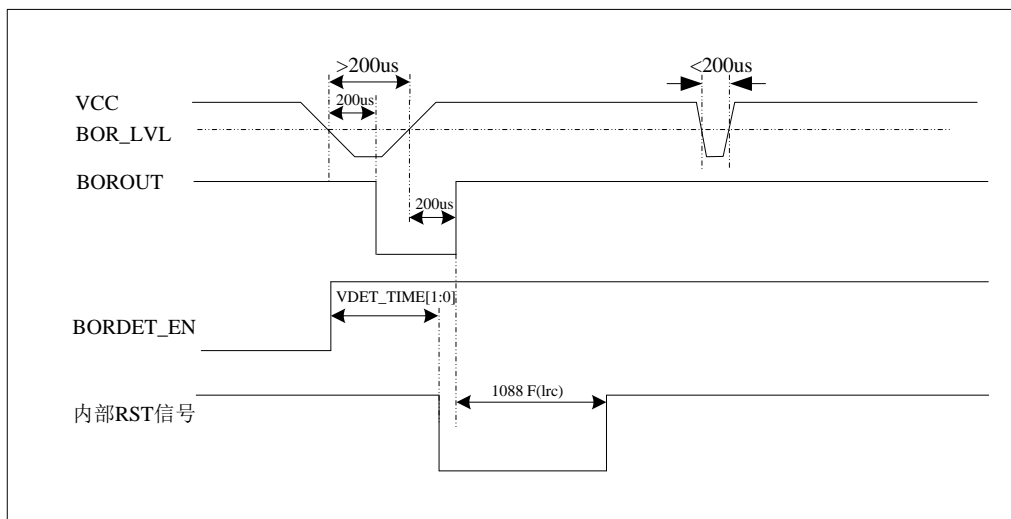
- 产生一个 BOR 脉冲
- 内部复位信号 IRST 有效
- 计数 1088 个  $F_{lrc}/F_{osc}$
- 复位状态寄存器 RSTSR 的掉电复位标志位 BOR 被设置为 1。
- CPU 从 0000H 开始执行程序

LBOR\_DET 与 BOR\_DET 的检测过程基本相同。



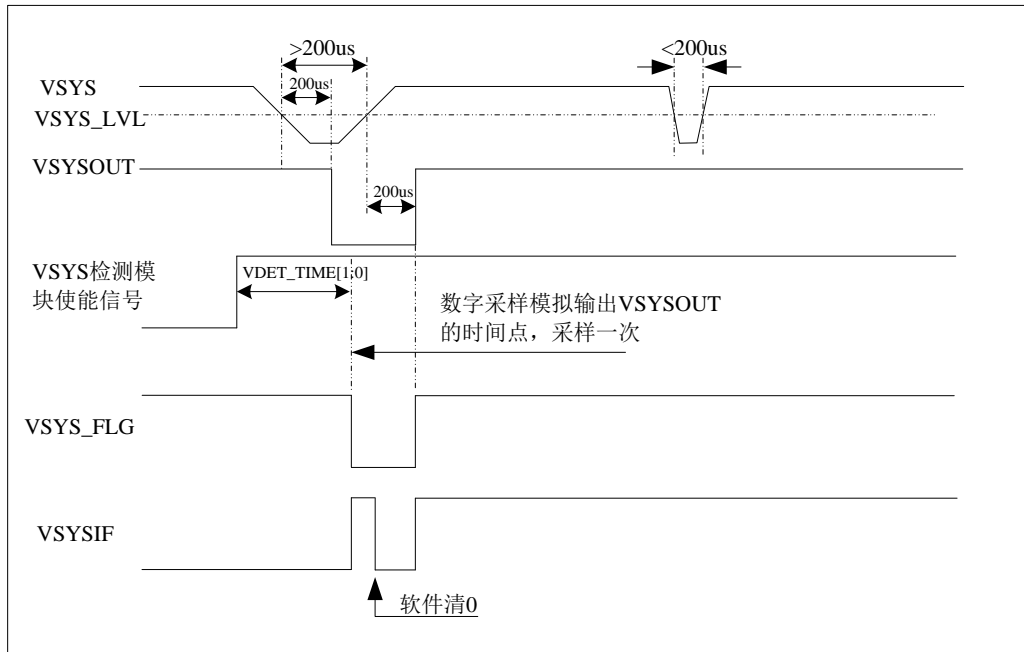


BOR 中断信号产生示意图

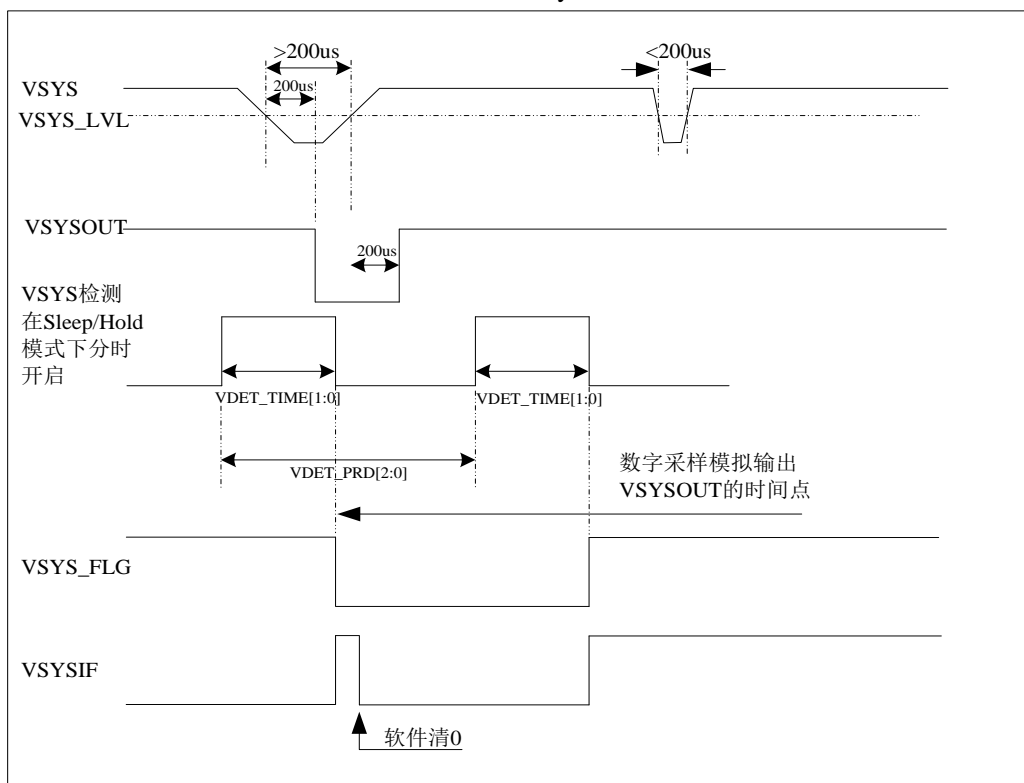


BOR 复位示意图

### 4.3.5 系统电源检测功能(VSYS\_DET)

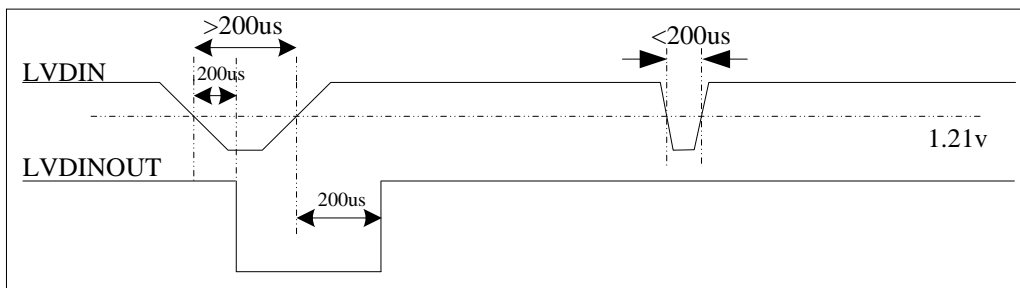


Normal 模式 Vsys 检测



Sleep 模式下 Vsys 分时检测

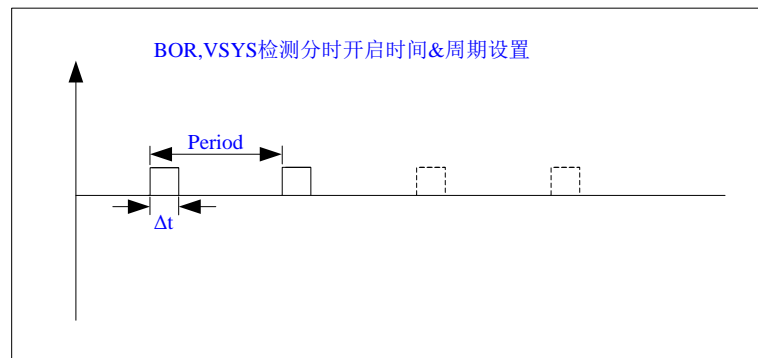
### 4.3.6 低电压检测功能(LVD\_DET)



LVDIN 检测信号示意图

### 4.3.7 VSYS\_DET, BOR\_DET, LVD\_DET 分时检测时序

系统在Hold或Sleep低功耗模式下时，为进一步降低系统功耗，LVD\_DET、VSYS\_DET、BOR\_DET采用分时开启的方式工作：



其中Period为LVD\_DET、VSYS\_DET和BOR\_DET检测模块分时开启的周期，可通过VDETPCFG寄存器的VDET\_PRD[2:0]位设置。 $\Delta t$ 是分时检测使能到LVD\_DET、VSYS\_DET和BOR\_DET开始采样的等待时间，可通过VDETPCFG寄存器的VDET\_TIME[1:0]位设置。

建议用户在实际应用中：

上电时使用 VSYS\_DET 或 BOR\_DET 模块检测系统电源 VSYS 的电压，即检测电源的后级状态，以使系统能进入一个可靠的状态。

掉电时使用 LVD\_DET 模块检测外部引脚 LVDIN 的电压，即检测电源的前级状态，以使系统能快速的检测到电源异常，并作相应的处理。



### 4.3.8 电池防钝化功能

当 VBAT Pin 连接外部电池时,如果置位 DISCHARGE\_EN 控制位,则使能了电池防钝化功能,此时 VBAT Pin 上会流过相应电流,用户可以配置 DISCHARGE\_CTRL 来控制防钝化电流大小。

## 4.4 特殊功能寄存器列表

PMU 模块寄存器基地址: 0x4000F400				
偏移地址	名称	读写方式	复位值	功能描述
0x00	PMUCON	R/W	0x001F	PMU配置寄存器(写保护)
0x04	VDETCFG	R/W	0x0005	电源检测阈值配置寄存器
0x08	VDETPCFG	R/W	0x0089	电源检测时间周期配置寄存器
0x0C	PMUIE	R/W	0x0000	PMU中断使能寄存器
0x10	PMUIF	R/W	0x0000	PMU中断标志寄存器
0x14	PMUSTA	R	0x0000	PMU状态指示寄存器
0x18	WAKEIF	R/W	0x00000000	唤醒源标志寄存器

## 4.5 特殊功能寄存器说明

PMUCON (写保护) (PMU 配置寄存器)		基地址: 0x4000F400 偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	DISCHAR	X	X	X	X	X	X	X
Write:	GE_EN							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	Hold_LD	LVDIN1_	LVDINO_	BORRST	BORDET_
Write:				0	EN	EN		EN
Reset:	0	0	0	1	1	1	1	1

位	功能描述
DISCHARGE_EN	<b>电池防钝化功能使能:</b> 0: 电池防钝化功能关闭 1: 电池防钝化功能开启
Hold_LDO	<b>在 Hold 模式下选择打开/关闭 大功耗 LDO (默认打开)</b> 0: 关闭大功耗 LDO 1: 打开大功耗 LDO (default)

LVDIN1_EN	<b>LVD_DET 模块使能信号，监测 LVDIN1 输入引脚</b> 0: 关闭 LVDIN1_DET 模块 1: 开启 LVDIN1_DET 模块 (default)
LVDINO_EN	<b>LVD_DET 模块使能信号，监测 LVDINO 输入引脚</b> 0: 关闭 LVDINO_DET 模块 1: 开启 LVDINO_DET 模块 (default)
BORRST	<b>BOR复位/中断选择位</b> 0: VCC电压低/高于设定阈值时产生BOR中断 1: VCC电压低于VDETCFG[1:0]设定阈值时产生BOR复位 (default)
BORDET_EN	<b>BORDET_EN 模块使能信号</b> 0: 关闭 BOR_DET 模块 1: 开启 BOR_DET 模块 (default)

<b>VDETCFG</b> (电源检测阈值配置寄存器)		<b>基地址: 0x4000F400</b> <b>偏移地址: 04H</b>						
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	DISCHAR	X	X	X	X	X	X	X
<b>Write:</b>	GE_CTRL							
<b>Reset:</b>	0	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	X	X	X	X	VSYS_LV L[1]	VSYS_LV L[0]	BOR_LVL[1:0]	
<b>Write:</b>								
<b>Reset:</b>	1	0	0	0	1	0	0	1

位	功能描述															
DISCHARGE_CTRL	<b>电池防钝化功能，电流大小选择:</b> 0: 1mA 1: 2mA															
VSYS_LVL [3:0]	<b>VSYS_DET检测阈值控制位</b> <table border="1"> <thead> <tr> <th colspan="2">VSYS_LVL[3:0]</th> <th>检测电压</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>2.4V</td> </tr> <tr> <td>0</td> <td>1</td> <td>2.6V</td> </tr> <tr> <td>1</td> <td>0</td> <td>2.8V(default)</td> </tr> <tr> <td>1</td> <td>1</td> <td>3.0V</td> </tr> </tbody> </table>	VSYS_LVL[3:0]		检测电压	0	0	2.4V	0	1	2.6V	1	0	2.8V(default)	1	1	3.0V
VSYS_LVL[3:0]		检测电压														
0	0	2.4V														
0	1	2.6V														
1	0	2.8V(default)														
1	1	3.0V														
BOR_LVL[1:0]	<b>BOR_DET检测阈值控制位</b> <table border="1"> <thead> <tr> <th colspan="2">BOR_LVL[1:0]</th> <th>检测电压</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>2.4V</td> </tr> <tr> <td>0</td> <td>1</td> <td>2.2V (default)</td> </tr> <tr> <td>1</td> <td>0</td> <td>2.8V</td> </tr> <tr> <td>1</td> <td>1</td> <td>2.6V</td> </tr> </tbody> </table>	BOR_LVL[1:0]		检测电压	0	0	2.4V	0	1	2.2V (default)	1	0	2.8V	1	1	2.6V
BOR_LVL[1:0]		检测电压														
0	0	2.4V														
0	1	2.2V (default)														
1	0	2.8V														
1	1	2.6V														

VDETPCFG (电源检测周期配置寄存器)			基地址: 0x4000F400 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	RSRV	VDET_TIME[1:0]		VDET_PRD[2:0]		
Write								
Reset:	0	0	1	0	0	0	1	0

位	功能描述																																				
RSRV	内部保留位，POR 复位后默认为 1，其它复位不可改变其值。 应用时可保留默认值 1，也可改写为 0。																																				
VDET_TIME[1:0]	<p>Hold&amp;Sleep 模式下 VSYS_DET, BOR_DET, LVDIN_DET 分时检测的时间设定</p> <table border="1"> <thead> <tr> <th colspan="2">VDET_Time[1:0]</th> <th>检测时间</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>336us(default)</td> </tr> <tr> <td>0</td> <td>1</td> <td>397us</td> </tr> <tr> <td>1</td> <td>0</td> <td>519us</td> </tr> <tr> <td>1</td> <td>1</td> <td>1068us(Recommend)</td> </tr> </tbody> </table> <p>注：VCC_DET、LVDIN_DET 和 BOR_DET 模块关闭再开启后，内部检测模块会等待 VDET_TIME 时间后，开始更新检测状态</p>	VDET_Time[1:0]		检测时间	0	0	336us(default)	0	1	397us	1	0	519us	1	1	1068us(Recommend)																					
VDET_Time[1:0]		检测时间																																			
0	0	336us(default)																																			
0	1	397us																																			
1	0	519us																																			
1	1	1068us(Recommend)																																			
VDET_PRD[2:0]	<p>Hold&amp;Sleep 模式下 VSYS_DET, BOR_DET, LVDIN_DET 分时检测的周期设定</p> <table border="1"> <thead> <tr> <th colspan="3">VDET_PRD[2:0]</th> <th>检测周期</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>16.5ms</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>33ms</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>67ms(default)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>134ms</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>268ms</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>536ms</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1072ms</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>2144ms</td> </tr> </tbody> </table> <p>注：设定的检测周期需大于检测时间。</p>	VDET_PRD[2:0]			检测周期	0	0	0	16.5ms	0	0	1	33ms	0	1	0	67ms(default)	0	1	1	134ms	1	0	0	268ms	1	0	1	536ms	1	1	0	1072ms	1	1	1	2144ms
VDET_PRD[2:0]			检测周期																																		
0	0	0	16.5ms																																		
0	0	1	33ms																																		
0	1	0	67ms(default)																																		
0	1	1	134ms																																		
1	0	0	268ms																																		
1	0	1	536ms																																		
1	1	0	1072ms																																		
1	1	1	2144ms																																		

PMUIE (PMU 中断使能寄存器)			基地址: 0x4000F400 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8

Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	X	LVDIN1IE	LVDINOIE	BORIE	VSYSIE
Write					E	E		
Reset:	0	0	0	0	0	0	0	0

注：需要同时使能 PMUIE 使能的中断才有效。

位	功能描述
LVDIN1IE	LVDIN1检测中断使能位 0: 关闭 1: 允许
LVDINOIE	LVDINO检测中断使能位 0: 关闭 1: 允许
BORIE	BOR检测中断使能位 0: 关闭 1: 允许
VSYSIE	VSYS检测中断使能位 0: 关闭 1: 允许

PMUIF (PMU 中断标志寄存器)			基地址: 0x4000F400 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	X	LVDIN1IF	LVDINOIF	BORIF	VSYSIF
Write					F	F		
Reset:	0	0	0	0	0	0	0	0

注：该寄存器不能被 Wake\_UP 唤醒复位。

位	功能描述
LVDIN1IF	LVDIN1检测中断标志位 当外部引脚LVDIN电压下降到低于1.21V或上升到高于1.21V时，该位置1，软件清0。
LVDINOIF	LVDINO检测中断标志位 当外部引脚LVDIN电压下降到低于1.21V或上升到高于1.21V时，该位置1，软件清0。
BORIF	BOR检测中断标志位 当内部工作电压VDD下降到低于设定阈值或上升到高于设定阈值时，并且在BORRST=0的情况下，该位置1，软件清0。

VSYSIF	<b>VSYS检测中断标志位</b> 当系统电源VSYS电压下降到低于设定阈值或上升到高于设定阈值时，该位置1，软件清0。							
<b>PMUSTA</b> (PMU 状态寄存器寄存器)		<b>基地址: 0x4000F400</b> <b>偏移地址: 14H</b>						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	X	LVDIN1_FLG	LVDINO_FLG	BOR_FLG	VSYS_FLG
Write					X	X	X	X
Reset:	0	0	0	0	0X	0	0	0

注：该寄存器不能被 Wake\_UP 唤醒 RESET。

位	功能描述
LVDIN1_FLG	<b>LVDIN1 引脚电压状态</b> 0: 表示 LVDIN1 引脚电压小于 1.21V 阈值 1: 表示 LVDIN1 引脚电压大于 1.21V 阈值
LVDINO_FLG	<b>LVDINO 引脚电压状态</b> 0: 表示 LVDINO 引脚电压小于 1.21V 阈值 1: 表示 LVDINO 引脚电压大于 1.21V 阈值
BOR_FLG	<b>内部工作电压 VDD 电压状态</b> 0: 表示 VDD 小于设定阈值 (BOR_LVL[1:0]) 1: 表示 VDD 大于设定阈值 (BOR_LVL[1:0])
VSYS_FLG	<b>系统电源 VSYS 电压状态</b> 0: 表示 VSYS 小于设定阈值 (VSYS_LVL[1:0]) 1: 表示 VSYS 大于设定阈值 (VSYS_LVL[1:0])

<b>WAKEIF</b> (唤醒标志寄存器)		<b>基地址: 0x4000F400</b> <b>偏移地址: 18H</b>						
	Bit31	30	29	28	27	26	25	Bit24
Read:	X	X	X	X	KEYWKIF	DMAWKIF	EMUWKIF	SelfTest
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	I2CWKIF	RTCWKIF	TBSWKIF	TMR3WKIF	TMR2WKIF	TMR1WKIF
Write:						F	F	F
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMR0WKIF	X	RX4WKIF	RX3WKIF	X	X	RX0WKIF	X

Write:	F							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:			INT3WKI	INT2WKI	INT1WKI	INT0WKI	AES/Random/Ghash	PMUWKIF
Write:	X	X	F	F	F	F		
Reset:	0	0	0	0	0	0	0	0

注：1. Sleep 唤醒和 Hold 唤醒共用此标志位

2. DMAWKIF / EMUWKIF / TMR3WKIF / TMR2WKIF / TMR1WKIF / TMROWKIF / AES/Random/Ghash 只能 Hold 下才行

位	功能描述
KEYWKIF	按键扫描唤醒标志
SelfTestFreq	自测频中断，不能唤醒，该位恒为0
RTCWKIF	RTC唤醒标志 SLEEP模式下RTC中断发生时将会产生RTC唤醒，此位置为1。 (具体是RTC哪个唤醒源，需要查看RTCIF寄存器)
TBSWKIF	TBS唤醒标志 SLEEP模式下TBS中断发生时将会产生TBS唤醒，此位置为1。 (具体是TBS哪个唤醒源，需要查看TBSIF寄存器)
RX4WKIF	RX4唤醒标志 RX4唤醒发生时，设置标志位为1，硬件进入Sleep后再唤醒则清除之前的标志
RX3WKIF	RX3唤醒标志 RX3唤醒发生时，设置标志位为1，硬件进入Sleep后再唤醒则清除之前的标志
RX0WKIF	RX0唤醒标志 RX0唤醒发生时，设置标志位为1，硬件进入Sleep后再唤醒则清除之前的标志
EXTI4WKIF	INT4唤醒标志 INT4唤醒发生时，设置标志位为1，硬件进入Sleep后再唤醒则清除之前的标志
EXTI3WKIF	INT3唤醒标志 INT3唤醒发生时，设置标志位为1，硬件进入Sleep后再唤醒则清除之前的标志
EXTI2WKIF	INT2唤醒标志 INT2唤醒发生时，设置标志位为1，硬件进入Sleep后再唤醒则清除之前的标志
EXTI1WKIF	INT1唤醒标志 INT1唤醒发生时，设置标志位为1，硬件进入Sleep后再唤醒则清除之前的标志
EXTI0WKIF	INT0唤醒标志 INT0唤醒发生时，设置标志位为1，硬件进入Sleep后再唤醒则清除之前的标志
PMUWKIF	PMU唤醒标志 SLEEP模式下PMU事件发生时将会产生PMU唤醒，此位置为1，硬件进入Sleep后再唤醒则清除之前的标志 (具体那个PMU唤醒源需要查看PMUIF寄存器。)

## 5 调试支持

### 5.1 概况

HT500X 采用的是 Cortex-M0 内核，该内核含有硬件调试模块。Cortex-M0 处理器支持以下调试特性：

- 程序的暂停、恢复以及单步执行；
- 访问处理器内核寄存器和特殊寄存器；
- 硬件断点（最多 4 个）；
- 软件断点（BKPT 指令）；
- 数据监视点（最多两个）；
- 动态存储器访问；
- 支持 SW 调试协议；

SW 为 2 针接口，该接口包含的信号如下表

SW 信号	描述
SWIO	数据信号
SWCLK	时钟信号

### 5.2 SW 引脚分布

SW 口调试接口		引脚描述
类型	描述	
输入/输出	SW 数据	PB. 13/ SEG13/SWIO
输出	SW 时钟	PB. 15/ SEG15/SWCLK

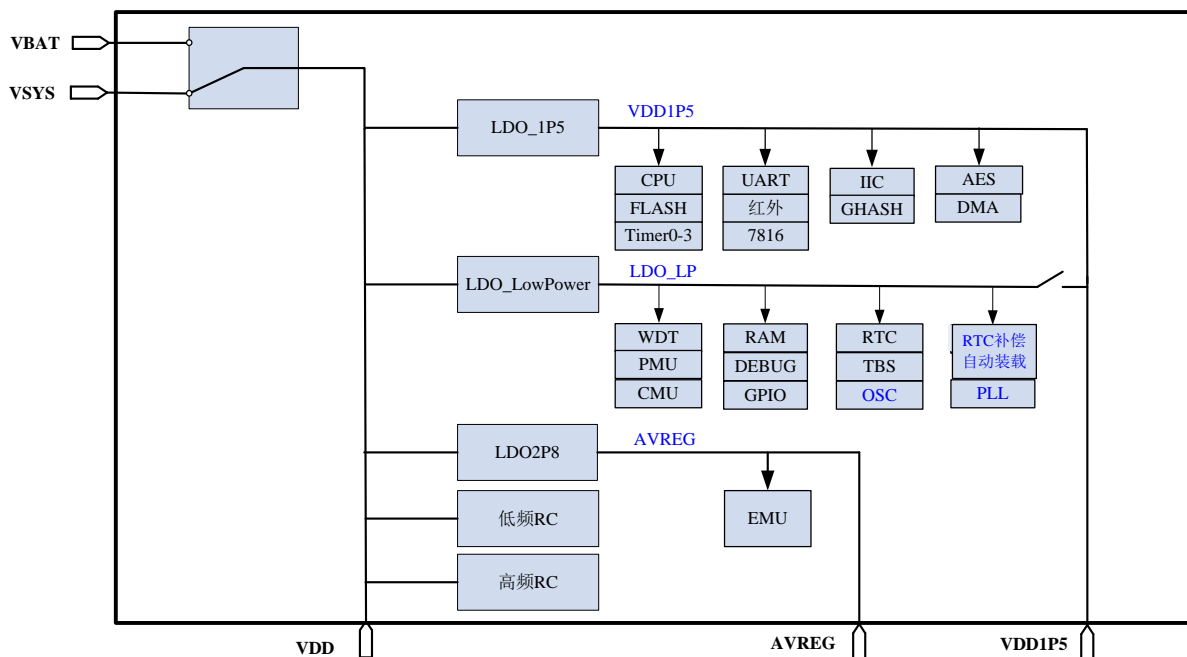
### 5.3 SW 引脚使用说明

工作模式	管脚功能		
	PA. 6/INT1/JTAGWDTEN	PB. 13/ SEG13/SWIO	PB. 15/ SEG15/SWCLK
测试模式	X	X	X
调试模式	JTAGWDTEN	SWIO	SWCLK
正常模式	PA. 6/INT1	PB. 13/SEG13	PB. 15/SEG15

注：X 表示不可用

## 6 工作模式

### 6.1 芯片电源域分配



### 6.2 工作模式

芯片共有 5 种模式：正常模式，调试模式，测试模式，Sleep 模式，Hold 模式

TEST	JTAGWDTEN	工作模式
0	0	<b>调试模式</b> 看门狗硬件禁止，可软件开启 PB13, PB15 被固定为 SW 接口
0	1	<b>测试模式</b>
1	X	<b>正常模式</b> 芯片内部所有模块电源正常供电，系统时钟和模块开关配置根据用户软件决定。 看门狗不能被禁止，只能在 Sleep 或 Hold 模式下通过关闭内部低频 RC 时钟来禁止看门狗功能
		<b>Sleep 模式</b> 在正常模式下，CPU通过执行如下指令进入Sleep模式： SCB->SCR = 0x0004;



	<pre>__WFI();</pre>
	<p><b>Hold 模式</b></p> <p>在正常模式下，CPU通过执行如下指令进入Hold模式：</p> <pre>SCB-&gt;SCR = 0x0000;</pre> <pre>__WFI();</pre>

芯片在正常模式下可以通过软件配置进入两种不同的低功耗模式，分别是 Sleep 模式和 Hold 模式。Sleep 模式和 Hold 模式最主要有以下两点区别：

- Sleep 模式可以获得更低的功耗
- Sleep 模式的唤醒等同复位，而 Hold 模式唤醒则是接着原来运行的代码继续运行

## 6.3 睡眠模式 (Sleep)

### 6.3.1 SLEEP 模式下各模块开关

- 数字电源LDO\_1P5关闭，其供电的模块相应关闭；
- RTC供电模块不关闭，RTC相关的晶振电路，TBS模块，分频补偿电路一直开启；
- 进入Sleep后，如果用户配置开启BOR和VCC检测功能，则BOR\_DET、LBOR\_DET和VCC\_DET模块会由硬件分时开启以降低功耗；
- WDT默认开启，在SLEEP模式下，WDT计数溢出时，系统会发生WDT复位；
- 进入SLEEP之前，如果配置LCD、TBS模块开启，在进入SLEEP模式后，即可实现LCD静态显示，温度和电池电压测量功能；
- 为降低SLEEP模式下的功耗，可以在进入SLEEP模式之前，配置GPIO的状态（详见GPIO章节），控制好芯片和外设的状态，防止通过GPIO往外部漏电；
- 如果用户期望在Sleep达到最低功耗：
  - (1) CLKCTRL0和CLKCTRL1寄存器全部清0
  - (2) CTRLBYFLASH中的LRC\_EN清0，关闭内部低频RC
- 进入SLEEP 模式后，芯片内部会自动关闭LDO\_1P5（大功耗）输出，LDO\_LowPower（低功耗）保持输出1.5V。

### 6.3.2 SLEEP 模式下的唤醒

CPU从SLEEP状态下唤醒等同复位，因此不进入中断向量，不会执行中断服务程序，程序从复位地址0000H开始执行。

在SLEEP模式下，Reset复位信号是不可被屏蔽的，包括POR，BOR，LBOR，外部RESET PIN上产生的外部复位信号，以及内部的WDT复位信号。当系统进入SLEEP模式后，如果以上复位信号产生，能够使芯片出现复位动作，程序从复位地址0000H开始执行。

要实现在SLEEP模式下的唤醒功能，进入SLEEP模式前需进行以下功能配置，且必需先使能模块的总中断：

- (1) 外部INT中断和串口RX引脚唤醒：相应的PIN要配置为INT或RX功能（详见GPIO单元），并使能外部中断和UART中断（NVIC\_EnableIRQ(EXTI0\_IRQn)、NVIC\_EnableIRQ(UART0\_IRQn)）。

若RX引脚唤醒，只要相应PIN上出现**电平翻转**并保持高/低电平不少于2个Fcpu的时间，CPU可从SLEEP模式下唤醒；若外部INT唤醒，当相应PIN上出现**触发沿**并保持高/低电平不少于2个Fcpu的时间，CPU可从SLEEP模式下唤醒，触发沿类型由EXTIE寄存器的RIE和FIE控制位配置。

注意：

- ✓ 这里的Fcpu时钟为CPU时钟分频寄存器的输出时钟，如果用户在进入SLEEP之前CPU时钟分频寄存器SYSCLKDIV的分频值过大（如1/128），则会造成外部唤醒PIN上要给出很长时间（超过2个Fcpu时钟）的低电平信号，才能将CPU从SLEEP模式下唤醒。
- ✓ 若使用外部INT唤醒SLEEP，必须要将PINFLT/PINFLT2寄存器中的相应的引脚数字滤波功能打开。

(2) RTC中断唤醒：配置RTC模块中断使能（NVIC\_EnableIRQ(RTC\_IRQn)），配置对应RTC的子中断源（仅配置RTCIE相应位中断使能即可），当RTC使能的中断时间到时，或者RTC使能的闹钟定时或定时器定时时间到时，CPU从SLEEP模式下唤醒。

(3) PMU中断和TBS中断唤醒：配置PMU/ TBS模块中断使能（NVIC\_EnableIRQ(PMU\_IRQn)或NVIC\_EnableIRQ(TBS\_IRQn)），使能PMU/ TBS对应的子中断源（配置PMUIE和TBSIE相应位中断使能），当电源检测超过阈值或温度检测超过阈值时，CPU从SLEEP模式下唤醒。

(4) **KEY中断唤醒**：相应的PIN要配置为KEY扫描功能，**配置按键扫描模块中断使能（NVIC\_EnableIRQ(KEY\_IRQn)）**，当检测到按键按下时，CPU从SLEEP模式下唤醒。

### 6.3.3 从 SLEEP 模式唤醒后的唤醒方式确认

从SLEEP模式唤醒后，可以查询复位标志（RSTSR寄存器的WKR位），如果该位为1，则说明确实发生了唤醒复位，然后再通过唤醒标志寄存器WAKEIF确定具体的唤醒源，其中：

- 1) WAKEIF的RTCWKIF位为1，表示CPU是由RTC中断信号引起了唤醒。具体的中断源由RTC的8种中断源确定，可以查询RTCIF对应的位来确认是哪种RTC中断引起的唤醒，详细见RTC单元。
- 2) WAKEIF的PMUWKIF位为1，表示CPU是由PMU中断信号引起了唤醒。具体的中断源由PMU的4种中断源确定，可以查询PMUIF对应的位来确认是BORIF、VCCIF、LVDIF、POWIF，详细见PMU单元。
- 3) WAKEIF的TBSWKIF位为1，表示CPU是由TBS中断信号引起了唤醒。具体的中断源由TBS的9种中断源确定，可以查询TBSIF对应的位来确认是TPSIF、VbatIF、VccIF、ADC0IF、ADC1IF、ADC2IF、VbatCMPIF、ADC0CMPIF、ADC1CMPIF，详细见TBS单元。
- 4) WAKEIF的INTxWKIF（x=0~3）位为1，表示CPU是由相应的外部INT引脚信号引起了唤醒。
- 5) WAKEIF的RXxWKIF（x=0, 3, 4）位为1，表示CPU是由相应的RX引脚信号引起了唤醒。

### 6.3.4 进入 Sleep 模式

Sleep模式通过Cortex M0的系统自带指令WFI进入，即使在调试状态下，也是可以进入Sleep模式的。进入Sleep指令如下：

```
SCB->SCR = 0x0004;  
__WFI();
```

## 6.4 待机模式 (Hold)

Hold 模式与 Sleep 模式的区别就是在 Hold 模式下，LD0\_1P5 和 LD0\_LowPower 两个是由用户控制开关的，数字的 LD0\_LowPower 供电一直打开，但是由于其低输出驱动能力 (20-30uA)，导致在这个状态下，很多数字功能模块不能使能，用户可配置开启大功耗大输出驱动能力的 LD0\_1P5 来适应其应用的需求。

为了降低 Hold 模式下的功耗，BOR\_DET, VCC\_DET 模块由芯片硬件分时开启。

如果系统在进入 Hold 模式之前配置了中断使能，在进入 Hold 模式后发生相应的中断事件，则会导致芯片从 Hold 模式下唤醒，并进入相应的中断处理程序。

WDT 默认开启，在 Hold 模式下，WDT 计数溢出时，系统会发生 WDT 复位，但可以配置 LRC\_EN =0 在 HOLD 模式下关闭 WDT (详见 CTRLBYFLASH 寄存器)；

### 6.4.1 进入 Hold 模式

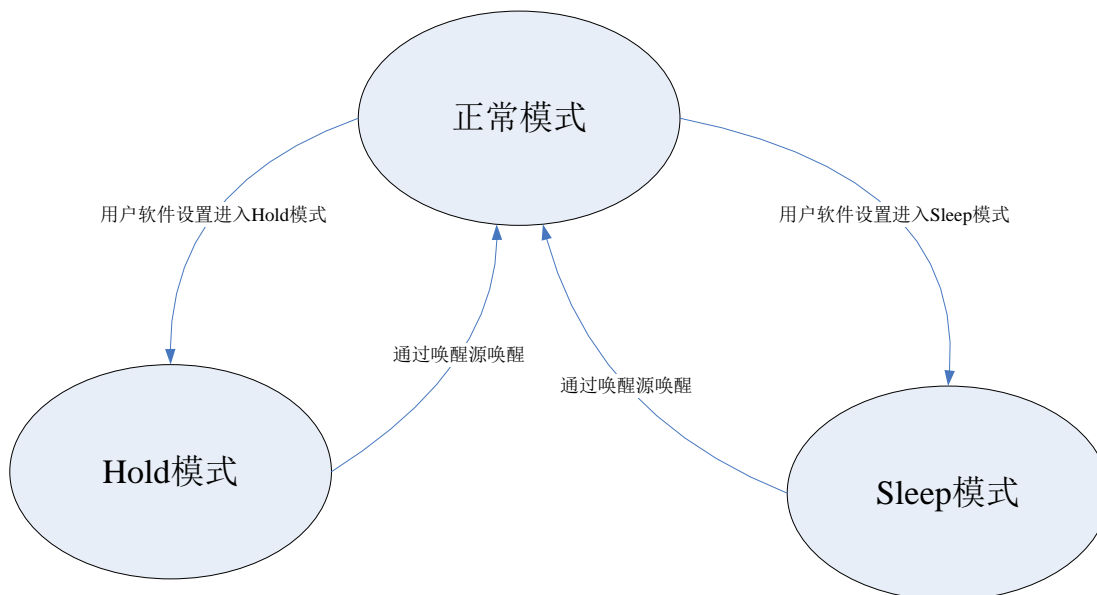
Hold模式通过CortexM0的系统自带指令WFI进入。

进入Hold指令如下：

```
SCB->SCR = 0x00;
```

```
__WFI();
```

### 6.4.2 模式转换图



## 6.5 特殊功能寄存器列表

基地址: 0xE000ED00				
偏移地址	名称	读写方式	复位值	功能描述
0x10	SCR	R/W	0x0000	系统控制寄存器
模块寄存器基地址: 0x4000F400				
偏移地址	名称	读写方式	复位值	功能描述
0x18	WAKEIF	R/W	0x0000	唤醒源标志寄存器

## 6.6 特殊功能寄存器说明

SCR (系统控制寄存器)		基地址: 0xE000ED10 偏移地址: 10H						
	Bit31	30	29	28...11	10	9	Bit8	
Read:	SCR[31:8]							
Write:	SCR[31:8]							
Reset:	0	0	0	0	0	0	0	
	Bit7	6	5	4	3	2	1	Bit0
Read:				RESERVE		SLEEPDE	RESERVE	
Write:				D		EP	D	
Reset:	0	0	0	0	0	0	0	0

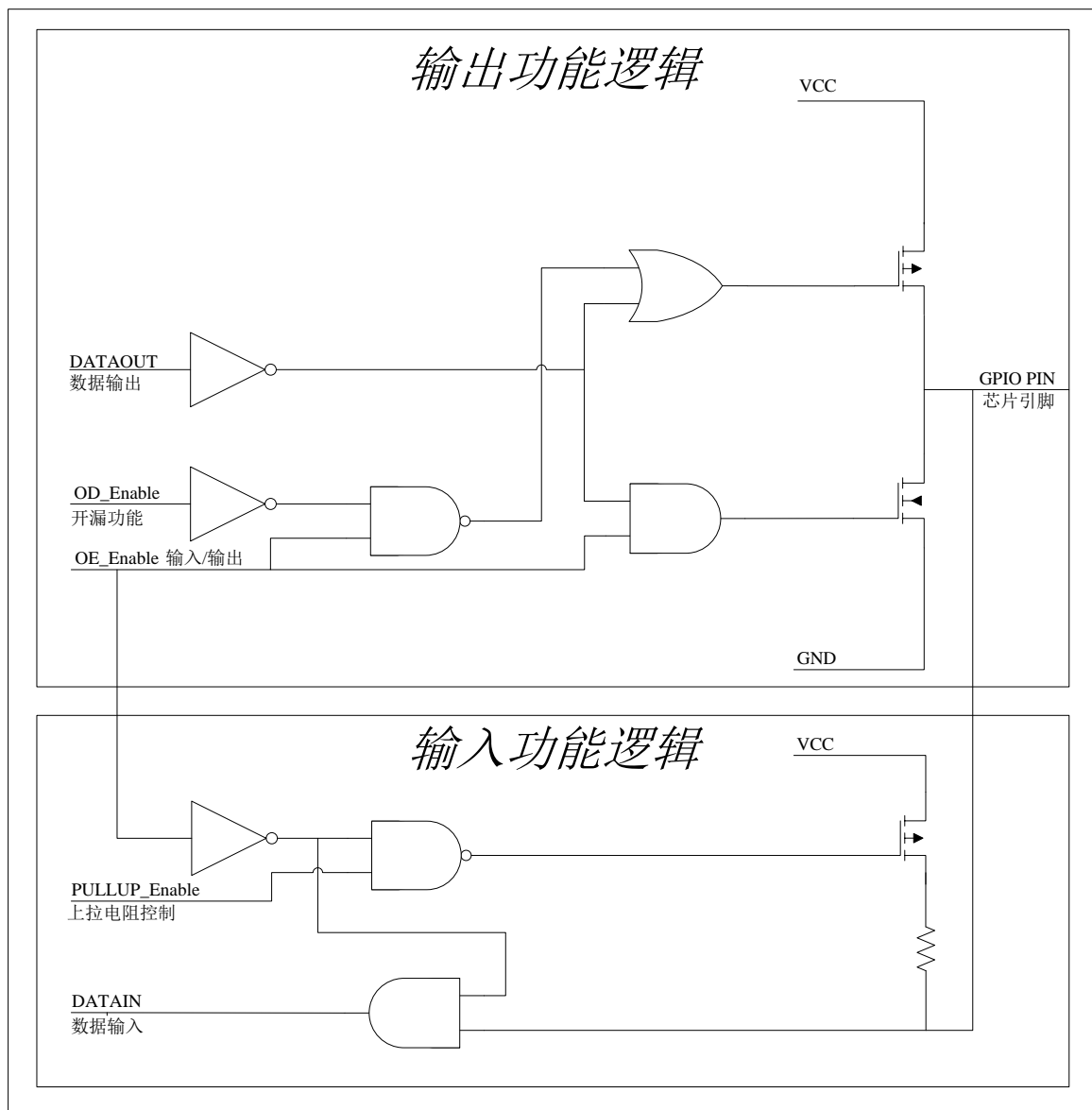
位	功能描述
SLEEPDEEP	=1: 在执行 WFI 指令后, 芯片进入 Sleep 模式 =0: 在执行 WFI 指令后, 芯片进入 Hold 模式

## 7 GPIO 模块

### 7.1 概述

- HT5007 提供 PA3, PA[5..8], PA12, PB[0..7], PB[12..15], PC[2..3], PC8, PD[2..5], PD[8..15], PE[1..5], PE7, PE9, PF[0..2] 并行端口, 支持 43 个双向 I/O 引脚, 可以分别配置成输入或者输出模式。作为输入方式时, 内部上拉 88K 电阻可配置。
- HT5009 提供 PA[5, 6, 7, 8, 12], PB[0..5], PB[13..15], PC[0, 1], PD[2, 3], PD[8..15], PE[1, 2, 3, 4, 5], PE9, PF0 并行端口, 支持 33 个双向 I/O 引脚, 可以分别配置成输入或者输出模式。作为输入方式时, 内部上拉 88K 电阻可配置。
- 各个I/O都具有最小3mA的驱动能力, 部分具备大驱动能力的IO如引脚说明里所述。

## 7.2 芯片引脚结构说明



## 7.3 芯片引脚描述

详见 1.5 章节。

## 7.4 I/O 端口基地址列表

GPIO 模块寄存器基地址:

0x40011000 (PA 端口);  
 0x40011100 (PB 端口);  
 0x40011200 (PC 端口);  
 0x40011300 (PD 端口);  
 0x40011400 (PE 端口);  
 0x40011600 (PF 端口);

偏移地址	名称	读写方式	复位值	功能描述
0x00	IOCFG	R/W	0x0000	端口功能配置寄存器 1 (写保护)
0x04	AFCFG	R/W	0x0000	端口功能配置寄存器 2 (写保护)
0x08	PTDIR	R/W	0x0000	端口方向配置寄存器
0x0C	PTUP	R/W	0x0000	端口上拉配置寄存器
0x10	PTDAT	R/W	0x0000	端口数据寄存器
0x14	PTSET	W	0x0000	端口设置寄存器 (只写)
0x18	PTCLR	W	0x0000	端口复位寄存器 (只写)
0x1C	PTTOG	W	0x0000	端口翻转寄存器 (只写)
0x20	PTOD	R/W	0xFFFF	端口 OD 功能配置寄存器

GPIO 模块寄存器基地址: 0x40011500 (大电流端口);

偏移地址	名称	读写方式	复位值	功能描述
0x00	HDPORT	R/W	0x0000	大电流端口配置寄存器

## 7.5 特殊功能寄存器说明

IOCFG (写保护) (端口功能配置寄存器 1)			基地址: 0x40011000--0x40011400 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PT[15:8]							
Write:	PT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PT[7:0]							
Write:	PT[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PT[15:0]	<b>端口 IO 功能配置位</b> 0: 对应的端口配置为 GPIO 1: 对应的端口配置为功能 PIN

AFCFG (写保护) (端口功能配置寄存器 2)			基地址: 0x40011000--0x40011400 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PT[15:8]							
Write:	PT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PT[7:0]							
Write:	PT[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PT[15:0]	端口复用功能配置位 (此寄存器只在对应端口配置为功能 PIN 时才有效) 0: 复用功能 1 1: 复用功能 2

PTDIR (端口方向配置寄存器)			基地址: 0x40011000--0x40011400 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PT[15:8]							
Write:	PT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PT[7:0]							
Write:	PT[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PT[15:0]	端口方向配置位 (此寄存器只在对应端口配置为 GPIO 时才有效) 0: 输入 1: 输出

PTUP (端口上拉配置寄存器)			基地址: 0x40011000--0x40011400 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PT[15:8]							
Write:	PT[15:8]							
Reset:	0	0	0	0	0	0	0	0



	Bit7	6	5	4	3	2	1	Bit0
Read:	PT[7:0]							
Write:	PT[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PT[15:0]	端口上拉配置位（此寄存器只在对应端口配置为数字输入时才有效） 0: 使能上拉 1: 禁止上拉（浮空）

PTDAT (端口数据寄存器)		基地址: 0x40011000--0x40011400 偏移地址: 10H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	PT[15:8]							
Write:	PT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PT[7:0]							
Write:	PT[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PT[15:0]	端口数据位（此寄存器只在对应端口配置为 GPIO 时才有效） 当端口配置为输入时为读到的 IO 口状态 0: 读到的为低电平 1: 读到的为高电平 当端口配置为输出时 0: 输出低电平 1: 输出高电平

PTSET (端口设置寄存器)		基地址: 0x40011000--0x40011400 偏移地址: 14H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	PT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	X
Write:	PT[7:0]							
Reset:	0	0	0	0	0	0	0	0

注：本寄存器只可写入。

位	功能描述
PT[15:0]	端口设置位（此寄存器只在对应端口配置为 GPIO 且输出时才有效） 0: 写 0 无效 1: 写 1 将对应的端口输出高电平（同时更新 PxDA 中对应的值）

PTCLR (端口复位寄存器)			基地址: 0x40011000--0x40011400 偏移地址: 18H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	PT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	X
Write:	PT[7:0]							
Reset:	0	0	0	0	0	0	0	0

注：本寄存器只可写入。

位	功能描述
PT[15:0]	端口复位位（此寄存器只在对应端口配置为 GPIO 且输出时才有效） 0: 写 0 无效 1: 写 1 将对应的端口输出低电平（同时更新 PxDA 中对应的值）

PTTOG (端口翻转寄存器)			基地址: 0x40011000--0x40011400 偏移地址: 1CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	PT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	X
Write:	PT[7:0]							
Reset:	0	0	0	0	0	0	0	0

注：本寄存器只可写入。

位	功能描述
PT[15:0]	端口翻转位（此寄存器只在对应端口配置为 GPIO 且输出时才有效） 0: 写 0 无效 1: 写 1 将使对应的端口输出电平发生翻转（同时更新 PxDA 中对应的值）

PTOD (端口开漏配置寄存器)			基地址: 0x40011000--0x40011400 偏移地址: 20H					
---------------------	--	--	--	--	--	--	--	--

	Bit15	14	13	12	11	10	9	Bit8
Read:	PT[15:8]							
Write:								
Reset:	1	1	1	1	1	1	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	PT[7:0]							
Write:								
Reset:	1	1	1	1	1	1	1	1

位	功能描述
PT[15:0]	<p><b>端口开漏配置位（此寄存器只在对应端口配置为数字输出时才有效）</b></p> <p>0: 开漏功能使能（PMOS 禁止）</p> <p>1: 开漏功能无效（PMOS 使能）</p> <p>注：对于复用功能，比如 IIC 需要开漏功能，当用户将对应的 IO 配置为 IIC 功能时，且需配置开漏功能使能。</p>

<b>HDPORT</b> （大电流端口配置寄存器）		<b>基地址： 0x40011500</b> <b>偏移地址： 00H</b>						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	PA8HD	PA7HD	PA6HD	PC0HD
Write:								
Reset:	0	0	0	0	0	0	0	0

注：该寄存器为单独的起始地址

位	功能描述
PA8HD, PA7HD, PA6HD, PC0HD	<p><b>端口大电流驱动配置（分别对应 PA8, PA7, PA6, 和 PC0 端口）</b></p> <p>0: 普通驱动能力（2mA-3mA）</p> <p>1: 超大电流驱动能力（20mA-30mA）</p>

## 8 中断模块

### 8.1 中断向量说明

系统中断	中断号	中断使能	中断标志	功能描述
NMI	-14			硬件强制切换低频 RC 中断
HardFault	-13			故障/异常触发中断
SVCall	-5			软件触发中断
PendSV	-2			软件触发中断
SysTick	-1			系统定时器周期中断
<b>以上为内核自带</b>				
PMU	0	PMUIE. POWIE	PMUIF. POWIF	POW检测中断
		PMUIE. LVDIE	PMUIF. LVDIF	LVD检测中断
		PMUIE. BORIE	PMUIF. BORIF	BOR检测中断
		PMUIE. VCCIE	PMUIF. VCCIF	VCC检测中断
AES	1	AESGHASEIE. AESIE	AESGHASEIF. AESIF	AES 中断
		AESGHASEIE. GHASHIE	AESGHASEIF. GHASHIF	GHASH 中断
EXTIO-3	2-5	EXTIE. RIE[0:3]	EXTIF. RIF[0:3]	外部输入引脚上升沿中断
		EXTIE. FIE[0:3]	EXTIF. FIF[0:3]	外部输入引脚下降沿中断
UART0-5 (UART 3 和 UART 4 有 7816 功能)	9, 12, 13	UARTCFG. RXIE	UARTSTA. RXIF	UART 接收中断
		UARTCFG. TXIE	UARTSTA. TXIF	UART 发送中断
		7816CFG. OVERIE	7816STA. OVERIF	7816 溢出中断 (SCI3, SCI4)
		7816CFG. RXIE	7816STA. RXIF	7816 接收中断 (SCI3, SCI4)
		7816CFG. TXIE	7816STA. TXIF	7816 发送中断 (SCI3, SCI4)
TMRO-3	15-18	TMRIE. CMPIE	TMRIF. CMPIF	比较中断
		TMRIE. CAPIE	TMRIF. CAPIF	捕获中断
		TMRIE. OVERIE	TMRIF. OVERIF	周期性溢出中断
TBS	19	TBSIE. ADC1CMPIE	TBSIF. ADCOCMPIF	ADC1 比较中断
		TBSIE. ADCOCMPIE	TBSIF. ADCOCMPIF	ADCO 比较中断
		TBSIE. VBATCMPIE	TBSIF. VBATCMPIF	电池电压比较中断
		TBSIE. ADC2IE	TBSIF. ADC2IF	ADC 通道 1 测量中断
		TBSIE. ADC1IE	TBSIF. ADC1IF	ADC 通道 1 测量中断
		TBSIE. ADC0IE	TBSIF. ADC0IF	ADC 通道 0 测量中断
		TBSIE. VCCIE	TBSIF. VCCIF	电源电压测量中断
		TBSIE. VBATIE	TBSIF. VBATIF	电池电压测量中断
		TBSIE. TMPIE	TBSIF. TMPIF	温度测量中断
RTC	20	RTCIE. ALMIE	RTCIF. ALMIF	闹钟中断
		RTCIE. RTC2IE	RTCIF. RTC2IF	RTC 定时器 2 中断

		RTCIE. RTC1IE	RTCIF. RTC1IF	RTC 定时器 1 中断
		RTCIE. MTHIE	RTCIF. MTHIF	RTC 月中断
		RTCIE. DAYIE	RTCIF. DAYIF	RTC 日中断
		RTCIE. HRIE	RTCIF. HRIF	RTC 小时中断
		RTCIE. MINIE	RTCIF. MINIF	RTC 分钟中断
		RTCIE. SECIE	RTCIF. SECIF	RTC 秒中断
I2C	21		I2CCON. SI	
SelfTestFreq	24			
EMU	25		子中断标志见 EMU 章节中断说明	EMU 中断
DMA	26	DMAIE. TCIE	DMAIF. TCIF	DMA 传输完成中断
		DMAIE. BCIE	DMAIF. BCIF	DMA 块传输完成中断
		DMAIE. TEIE	DMAIF. TEIF	DMA 传输错误中断
KEY	27		KEYIF. KEYIF	KEY 中断

注:

1. 只有 UART3 和 UART4 有 7816 功能，因此只有该两个通讯端口有 7816 溢出中断，7816 接收中断，7816 发送中断；
2. 对于中断号大于等于 0 的中断，每个中断都有一个对应的中断使能信号，具体配置参见 21.4 CMSIS 函数说明。
3. NMI/HardFault/SVCALL/PendSV/SysTick 为内核自带，这些中断没有专门的中断使能控制位。

## 8.2 中断使能和禁止

中断的使能和禁止使用的是 CortexM0 内核的 NVIC 模块，使用 CortexM0 软件系统自带的函数：

```
void NVIC_EnableIRQ(IRQn_Type IRQn)
void NVIC_DisableIRQ(IRQn_Type IRQn)
void NVIC_SetPendingIRQ(IRQn_Type IRQn)
void NVIC_ClearPendingIRQ(IRQn_Type IRQn)
uint32_t NVIC_GetPendingIRQ(IRQn_Type IRQn)
void NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority)
uint32_t NVIC_GetPriority(IRQn_Type IRQn)
```

### 8.2.1 中断使能和禁止相关寄存器列表

寄存器地址	名称	读写方式	复位值	功能描述
0xE000E100	ISER	R/W	0x00000000	中断使能配置寄存器
0xE000E180	ICER	R/W	0x00000000	中断禁止配置寄存器
0xE000E200	ISPR	R/W	0x00000000	中断标志置位寄存器
0xE000E280	ICPR	R/W	0x00000000	中断标志清零寄存器

0xE000E400-0xE000E41C	IPRO-IPR7	R/W	0x00000000	中断优先级寄存器
-----------------------	-----------	-----	------------	----------

### 8.2.2 中断使能和禁止相关寄存器说明

ISER (中断使能配制寄存器)		寄存器地址: 0xE000E100							
		Bit31...Bit0							
Read:	SETENA[31:0]								
Write:									
Reset:	0	0	0	0	0	0	0	0	

中断使能寄存器，共 32 个控制位对应 32 个中断，每一位的对应关系见中断向量说明，如：

SETNA[0] 对应 PMU

SETNA[1] 对应 3DES

SETNA[2] 对应 EXTIO

SETNA[3] 对应 EXTI1

.....

推荐使用 CortexM0 库函数中的 NVIC\_EnableIRQ 来使能中断。

ICER (中断禁止配置寄存器)		寄存器地址: 0xE000E180							
		Bit31...Bit0							
Read:	CLRENA[31:0]								
Write:									
Reset:	0	0	0	0	0	0	0	0	

中断禁止寄存器，共 32 个控制位对应 32 个中断，每一位的对应关系见中断向量说明。

推荐使用 CortexM0 库函数中的 NVIC\_DisableIRQ 来禁止中断。

ISPR (中断标志置位寄存器)		寄存器地址: 0xE000E200							
		Bit31...Bit0							
Read:	SETPEND[31:0]								
Write:									
Reset:	0	0	0	0	0	0	0	0	

中断标志置位寄存器，共 32 个控制位对应 32 个中断，每一位的对应关系见中断向量说明。

推荐使用 CortexM0 库函数中的 NVIC\_SetPendingIRQ 来置位中断标志

ICPR (中断标志清零寄存器)		寄存器地址: 0xE000E280							
---------------------	--	-------------------	--	--	--	--	--	--	--

	Bit31…Bit0							
Read:	CLRPEND[31:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

中断标志清零寄存器，共 32 个控制位对应 32 个中断，每一位的对应关系见中断向量说明。

推荐使用 CortexM0 库函数中的 NVIC\_ClearPendingIRQ 来清零中断标志

IPRO --- IPR7 (中断优先级寄存器)			寄存器地址: 0xE000E400--0xE000E417					
	Bit31…Bit0							
Read:	Priority [31:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

推荐使用 CortexM0 库函数中的 NVIC\_SetPriority 来设置中断优先级

### 8.3 特殊功能寄存器列表

基地址: 0x40011800				
偏移地址	名称	读写方式	复位值	功能描述
0x00	EXTIE	R/W	0x0000	外部中断输入边沿控制寄存器
0x04	EXTIF	R/W	0x0000	外部中断输入滤波器设置
0x08	PINFLT	R/W	0x0000	引脚数字滤波器使能

### 8.4 特殊功能寄存器说明

EXTIE (外部中断边沿配置寄存器)		基地址: 0x40011800 偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	RIE[6:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	FIE[6:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RIE[6:0]	INT 外部输入引脚上升沿中断使能

	0: 禁止 1: 使能
FIE[6:0]	INT 外部输入引脚下降沿中断使能 0: 禁止 1: 使能

EXTIF (外部中断标志寄存器)		基地址: 0x40011800 偏移地址: 04H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	RIF[6:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	FIF[6:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RIF[6:0]	INT 外部输入引脚上升沿中断标志 0: 未产生中断 1: 产生中断
FIF[6:0]	INT 外部输入引脚下降沿中断标志 0: 未产生中断 1: 产生中断

PINFLT (外部引脚滤波器配置寄存器)		基地址: 0x40011800 偏移地址: 08H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	RXFLT[5:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	INTFLT[6:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RXFLT[5:0]	RX 外部输入引脚数字滤波功能使能 0: 禁止 1: 使能



INTFLT[6:0]	INT 外部输入引脚数字滤波功能使能 0: 禁止 1: 使能
-------------	--------------------------------------

## 9 Reset 模块

### 9.1 复位优先级

芯片共有 8 种复位方式，可分四种复位优先级。

RTC 模块为独立的，有其单独的 RTC 的 POR，MCU 和其它外设的 POR 模块与 RTC 的 POR 模块分开。

序号	复位源	复位等级	不能复位的寄存器
1	上电复位 (POR)	一级	1, 复位状态寄存器 RSTSR
2	低电压检测复位 (LBOR)		2, RTC 模块的寄存器
3	外部引脚 /RST 复位	二级	1, 复位状态寄存器 RSTSR
4	软件 POR/LBOR 复位		2, RTC 模块的寄存器
5	掉电复位 (BOR)		3, PMU 模块的寄存器 4, 唤醒标志寄存器 WAKEIF
6	看门狗复位 (WatchDog)	三级	1, 复位状态寄存器 RSTSR 2, RTC 模块的寄存器 3, PMU 模块的寄存器 4, 唤醒标志寄存器 WAKEIF 5, GPIO 模块的寄存器: IOCFG, AFCFG, PTDIR, PTUP, PTDAT, PTOD 6, LCD 相关寄存器: LCDCLK, LCDCON, LCD_BUF[i] 7, CMU 相关寄存器: CLKCTRL0, CLKCTRL1, LRCADJ, HRCADJ
7	调试复位 (Debug Reset)	四级	1, 复位状态寄存器 RSTSR 2, RTC 模块的寄存器 3, PMU 模块的寄存器 4, 唤醒标志寄存器 WAKEIF 5, GPIO 模块的寄存器: IOCFG, AFCFG, PTDIR, PTUP, PTDAT, PTOD
8	唤醒复位 (WakeUp Reset)		6, LCD 相关寄存器: LCDCLK, LCDCON, LCD_BUF[i] 7, CMU 相关寄存器: CLKCTRL0, CLKCTRL1, LRCADJ, HRCADJ 8, EXT 模块寄存器: EXTIE, EXTIF 9, TBS 模块寄存器 (除 DAT 寄存器) 10, KEY 模块寄存器

注 1: LRCADJ 会被 Watchdog 和 Debug Reset 复位

注 2: RSTSR 的 POR 复位标志和 LBOR 复位标志之间可以互相清除

注 3: 唤醒复位还不能复位所有中断标志寄存器

注 4: 所有的复位除内核软复位之外均会复位掉 CLKCTRL0 的 HRC\_EN 信号，复位之后系统跑 HRC。

## 9.2 复位说明

任何复位源产生复位后，CPU 的程序指针恢复到 0000H，绝大部分寄存器恢复到缺省值：

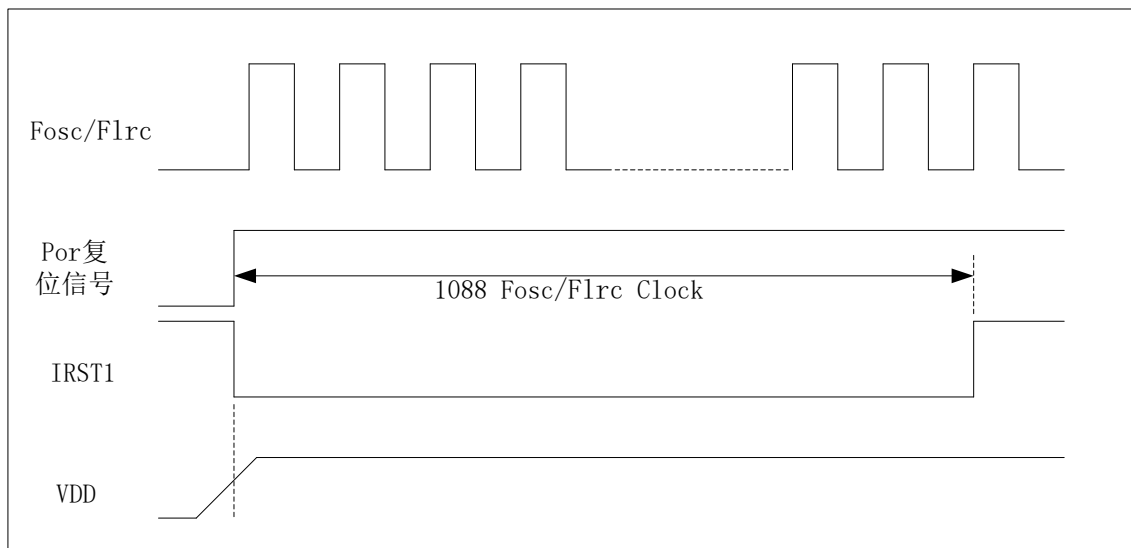
- 1) POR 和 LBOR 复位时，内部复位信号 IRST1 将保持有效，并保持 1088 个 Fosc/Flrc。
- 2) BOR 和外部 RST 复位时，内部复位信号 IRST2 将保持有效，并保持 1088 个 Fosc/Flrc。
- 3) WDT Reset 复位时，内部复位信号 IRST3 将保持有效，并保持 64 个 Fosc/Flrc。
- 4) SoftReset 和 Debug Reset 复位时，内部 IRST4 信号有效，并保持 64 个 Fosc/Flrc。
- 5) 只有第一次 POR 复位会对 RTC 复位

### 9.2.1 上电复位

当电源第一次加到芯片上时，上电复位电路检测电源电压 Vcc 上升到阈值 0.3V 时，POR 输出高电平，指示发生上电。内部复位信号 IRST1 保持为低电平，1088 个 Fosc/Flrc 后，IRST1 才会变为高电平。

上电复位 POR 产生时，下面的事件将会发生：

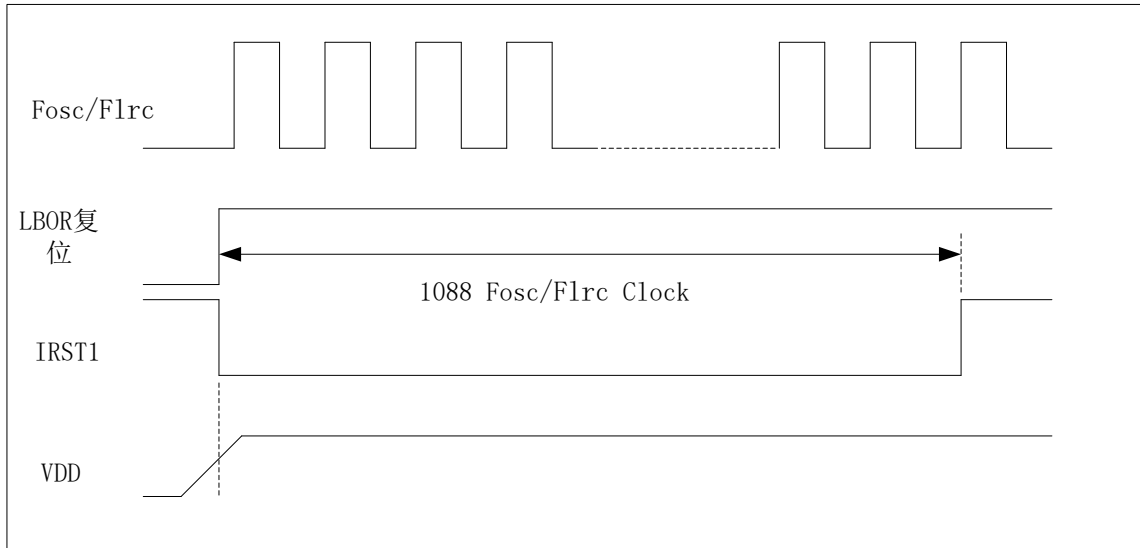
- POR 输出高电平
- 第一次 POR 复位会将 RTC 模块复位（RTC 的复位只能由 VRTC 的电源的上电 POR 复位）
- 内部复位信号 IRST1 有效
- 计数 1088 个 Fosc/Flrc
- 复位状态寄存器 RSTSTA 的上电复位标志位 POR 被设置为 1，其他 RSTSTA 为被清为 0。
- CPU 从地址 0000H 执行程序



上电复位 POR 说明

### 9.2.2 低电压检测复位

低电压检测复位（LBOR）在掉电后重新上电的复位过程与上电复位（POR）相同。

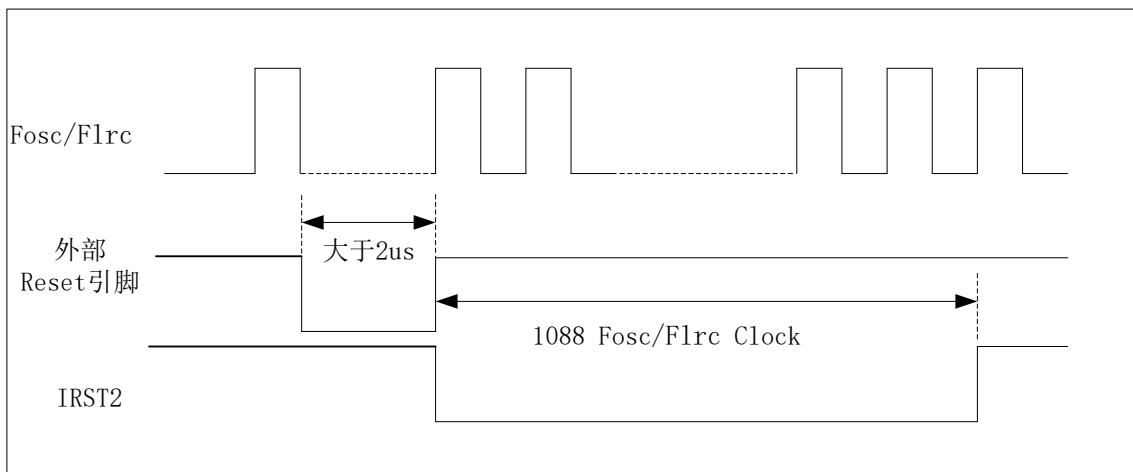


LBOR 复位说明

### 9.2.3. 外部引脚复位

外部复位引脚/RST 出现比 2us 宽的低电平时，内部复位信号 IRST2 有效，复位状态寄存器的复位标志位 RST 被设置为 1；内部复位信号 IRST2 有效脉宽为 1088 个 Fosc/Flrc。

如果/RST 低电平脉宽比 2us 窄，系统不发生复位。



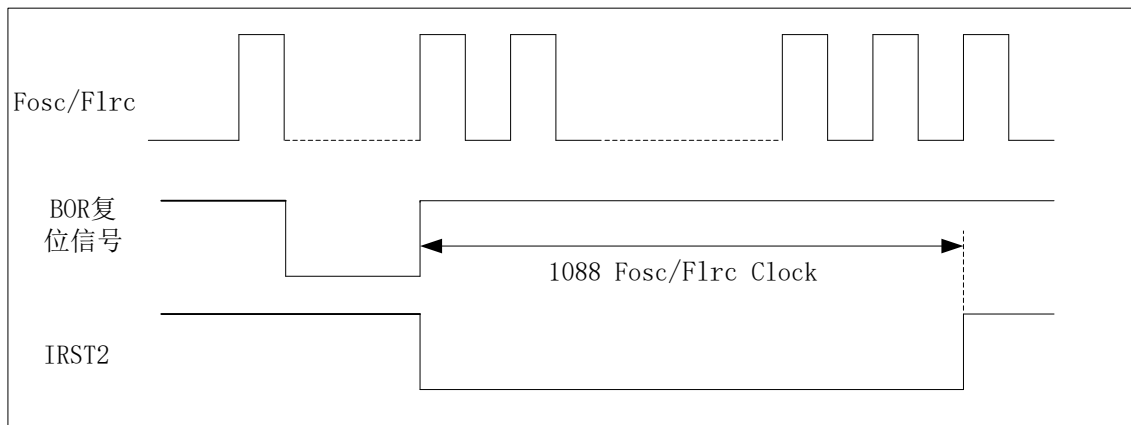
外部引脚复位说明

### 9.2.4. 掉电复位

当掉电检测电路检测到电源电压低于电压  $V_{bor}$  时，BOR 输出低电平，内部复位信号 IRST2 将变为低电平，复位状态寄存器 RSTSTA 的 BOR 标志位被置为 1。当掉电检测电路检测到电源电压高于电压  $V_{bor}$  时，BOR 输出高电平，IRST2 在 1088 个 Fosc/Flrc 时间之后变为高电平。

掉电复位 BOR 产生时，下面的事件将会发生：

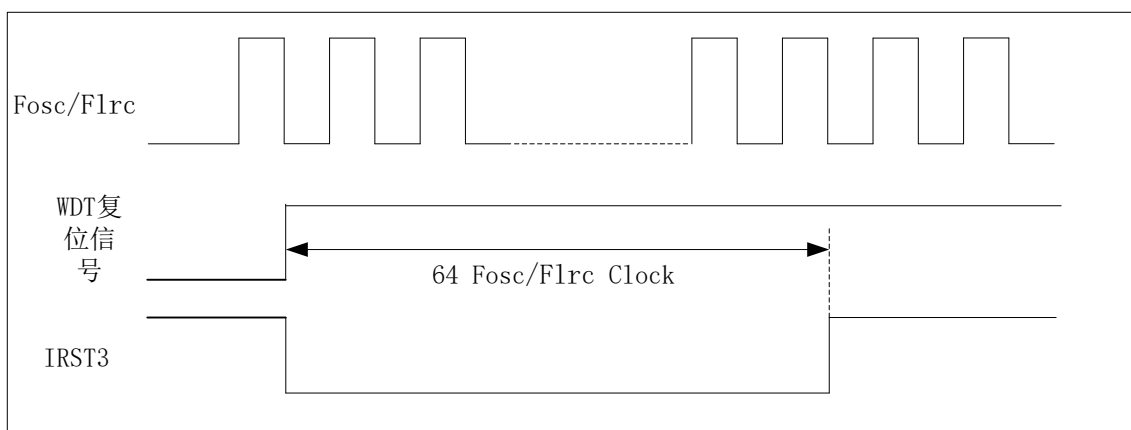
- 产生一个 BOR 脉冲
- 内部复位信号 IRST2 有效
- 计数 1088 个 Fosc/Flrc
- 复位状态寄存器 RSTSTA 的掉电复位标志位 BOR 被设置为 1，其他 RSTSTA 为被清为 0。
- CPU 从 0000H 开始执行程序



BOR 复位说明

## 9.2.5. 看门狗复位

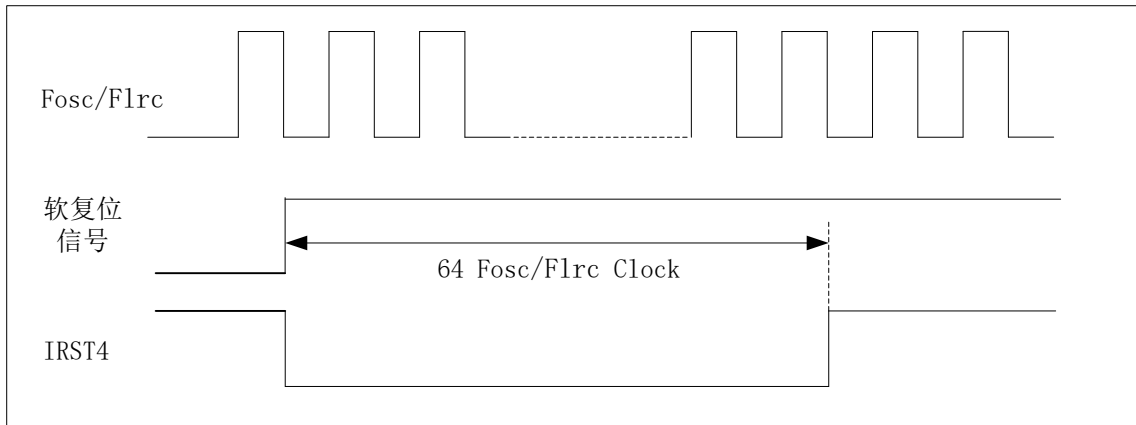
WatchDog Timer 溢出时将会产生导致内部复位 IRST3 有效，复位状态寄存器的 WDT 复位标志位 WDT 被设置为 1。WDT 的复位脉宽为 64 个 Fosc/Flrc。



WDT 复位

## 9.2.6. 软复位

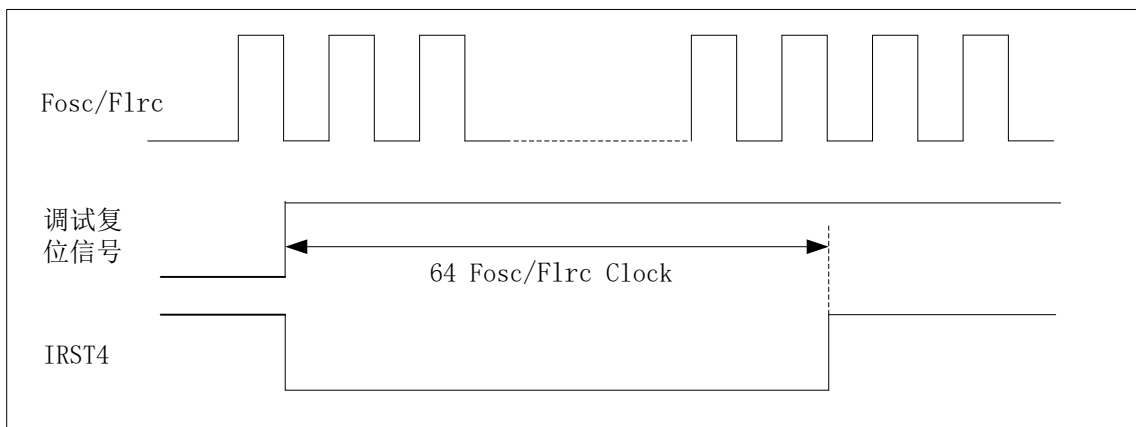
当向系统控制块的应用中断与控制状态寄存器寄存器 AIRCR bit2 写入 1 时，则产生一个软复位 (Cortex-M0 自带功能)。



软复位

### 9.2.7. 调试复位

这种复位只在 SW 调试状态下才有可能产生。

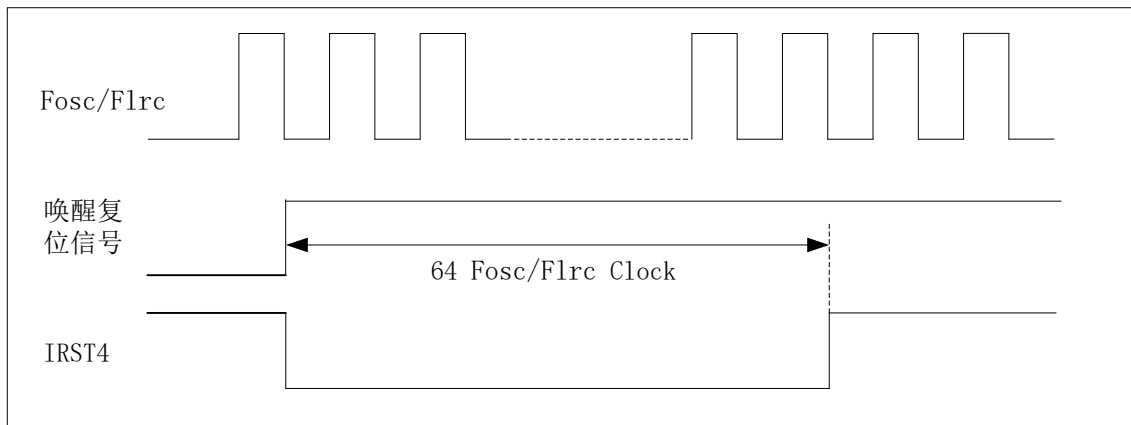


调试复位

### 9.2.8. 唤醒复位

出现 sleep 唤醒事件时，按照下面顺序执行：

- 内部复位信号 IRST4 有效
- 复位状态寄存器 RSTSTA 的掉电复位标志位 WKR 被设置为 1
- 计数 64 个 Fosc/Flrc 后，释放内部复位信号 IRST4



唤醒复位

## 9.3 特殊功能寄存器列表

基地值：0x4000F400（与 PMU 模块相同）

偏移地址	名称	读写方式	复位值	功能描述
0x30	RSTSTA	R/*W	-----	复位标志寄存器

注：该章节寄存器只有一个，故将这个寄存器位置放到 PMU 模块内。

基地值：0x4000F400（与 PMU 模块相同）

偏移地址	名称	读写方式	复位值	功能描述
0x18	WAKEIF	R/W	0x0000	唤醒源标志寄存器

基地值：0xE000ED00

偏移地址	名称	读写方式	复位值	功能描述
0x0C	AIRCR	R/W	-----	应用中断与控制状态寄存器

注：此寄存器为内核寄存器

## 9.4 特殊功能寄存器说明

RSTSTA (复位标志寄存器)			基地址： 0x4000F400 偏移地址： 30H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Hold_Fl	Sleep_	X	X	X	X	X	BORRST
Write:	ag	Flag						
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DebugRS	SoftRS	ExtRST	FlashWR	Wakeup	WDTRST	LBORRST	PORRST
Write:	T	T		_LBOR	RST			

Reset:	0	0	0	0	0	0	0	0
--------	---	---	---	---	---	---	---	---

位	功能描述
Hold_flag	<b>Hold 模式下中断唤醒标志位</b> 0: 未发生 Hold 模式下中断唤醒标志位 1: 发生了 Hold 模式下中断唤醒标志位 写 0 清 0, 不可写 1
Sleep_flag	<b>Sleep 模式下中断唤醒标志位</b> 0: 未发生 Sleep 模式下唤醒 1: 发生了 Sleep 模式下唤醒 写 0 清 0
BORRST	<b>BOR 复位标志位</b> 0: 未发生 BOR 复位 1: 发生了 BOR 复位 写 0 清 0, 不可写 1
DebugRST	<b>调试复位标志位</b> 0: 未发生 Debug Reset 复位 1: 发生了 Debug Reset 复位 写 0 清 0, 不可写 1
SoftRST	<b>软复位标志位</b> 0: 未发生 Soft Reset 复位 1: 发生了 Soft Reset 复位 写 0 清 0
ExtRST	<b>外部 RST 复位标志位</b> 0: 未发生 RST 复位 1: 发生了 RST 复位 写 0 清 0, 不可写 1
WakeupRST	<b>唤醒复位标志位</b> 0: 未发生 Wakeup Reset 复位 1: 发生了 Wakeup Reset 复位 写 0 清 0, 不可写 1
WDTRST	<b>Watch Dog 复位标志位</b> 0: 未发生 WDT 复位 1: 发生了 WDT 复位 写 0 清 0, 不可写 1
LBORRST	<b>LBOR 复位标志位</b> 0: 未发生 LBOR 复位 1: 发生了 LBOR 复位 写 0 清 0, 不可写 1
PORRST	<b>POR 复位标志位</b> 0: 未发生 POR 复位 1: 发生了 POR 复位 写 0 清 0, 不可写 1



WAKEIF (唤醒标志寄存器)			基地址: 0x4000F400 偏移地址: 18H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	X	X	X	X	KEYIF	DMAWKIF	EMUWKIF	SelfTest
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	RTCWKIF	TBSWKIF	TMR3WKIF	TMR2WKIF	TMR1WKIF
Write:						F	F	F
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMROWKIF	X	RX4WKIF	RX3WKIF	X	X	RXOWKIF	X
Write:	F							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	INT3WKIF	INT2WKIF	INT1WKIF	INTOWKIF	AES/Random/Ghash	PMUWKIF
Write:			F	F	F	F		
Reset:	0	0	0	0	0	0	0	0

注: 1. Sleep 唤醒和 Hold 唤醒共用此标志位

2. DMAWKIF / EMUWKIF / TMR3WKIF / TMR2WKIF / TMR1WKIF / TMROWKIF / AES/Random/Ghash 只能 Hold 下才行  
该寄存器为只读寄存器, 它永远会保持上一次导致芯片唤醒的唤醒源头, 当一个新的唤醒事件产生时候,  
由硬件产生新的唤醒源标志, 同时将之前的唤醒标志清 0.

位	功能描述
KEYWKIF	按键扫描唤醒标志
SelfTestFreq	自测频中断, 不能唤醒, 该位恒为0
RTCWKIF	RTC唤醒标志 SLEEP模式下RTC中断发生时将会产生RTC唤醒, 此位置为1。 (具体是RTC哪个唤醒源, 需要查看RTCIF寄存器)
TBSWKIF	TBS唤醒标志 SLEEP模式下TBS中断发生时将会产生TBS唤醒, 此位置为1。 (具体是TBS哪个唤醒源, 需要查看TBSIF寄存器)
RX4WKIF	RX4唤醒标志 RX4唤醒发生时, 设置标志位为1, 硬件进入Sleep后再唤醒则清除之前的标志
RX3WKIF	RX3唤醒标志 RX3唤醒发生时, 设置标志位为1, 硬件进入Sleep后再唤醒则清除之前的标志
RXOWKIF	RX0唤醒标志 RX0唤醒发生时, 设置标志位为1, 硬件进入Sleep后再唤醒则清除之前的标志
EXTI3WKIF	INT3唤醒标志 INT3唤醒发生时, 设置标志位为1, 硬件进入Sleep后再唤醒则清除之前的标志
EXTI2WKIF	INT2唤醒标志 INT2唤醒发生时, 设置标志位为1, 硬件进入Sleep后再唤醒则清除之前的标志

EXTI1WKIF	<b>INT1唤醒标志</b> INT1唤醒发生时，设置标志位为1，硬件进入Sleep后再唤醒则清除之前的标志
EXTI0WKIF	<b>INT0唤醒标志</b> INT0唤醒发生时，设置标志位为1，硬件进入Sleep后再唤醒则清除之前的标志
PMUWKIF	<b>PMU唤醒标志</b> SLEEP模式下PMU事件发生时将会产生PMU唤醒，此位置为1，硬件进入Sleep后再唤醒则清除之前的标志（具体那个PMU唤醒源需要查看PMUIF寄存器。）

<b>AIRCR</b> (应用中断与控制状态寄存器)			<b>基地址: 0xE000ED00</b>					
			<b>偏移地址: 0CH</b>					
	Bit31	30	29	28...11		10	9	Bit8
Read:	AIRCR [31:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	SYSRESE TREQ	RESERVE D	X
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SYSRESETREQ	=1: 芯片发生软复位 =0: 无效

## 10 UART/7816 通讯模块

### 10.1 功能说明

UART 串行通信模块实现与外部设备的异步串行通信。

特点:

- 共三路 UART
- UART3, UART4 分别与两路 7816 接口复用, 通过 MODESEL 寄存器选择
- 波特率可软件设置, 最高波特率 115200
- 全双工通信口, 可配置为红外调制输出, 红外调制极性可选
- 发送支持 1 个停止位或 2 个停止位
- 数据位宽支持 7 或 8 位
- 硬件自动完成奇偶校验, 数据接收完成的同时判断并提示奇偶校验错误, 给出标志。
- 接收/发送中断使能分别独立

串口提供灵活的全双工异步通信的接收器/发送器, 通过寄存器 UARTCFG 配置串口工作在不同的工作模式, 列举如下:

- 方式 1: 通过 TXD 发送或通过 RXD 接收 7 个数据位, 无奇偶校验, 波特率可变。
- 方式 2: 通过 TXD 发送或通过 RXD 接收 7 个数据位, 和 1 个奇偶校验位, 波特率可变。
- 方式 3: 通过 TXD 发送或通过 RXD 接收 8 个数据位, 无奇偶校验, 波特率可变。
- 方式 4: 通过 TXD 发送或通过 RXD 接收 8 个数据位, 和 1 个奇偶校验位, 波特率可变。

三路 UART 的输出 TX 都可以调制成 38K 红外信号。调制信号的占空比可调, 最大波特率不超过 2400bps。

三路 UART 的输入 RX 也都可以配置作为外部中断的输入, 可以很方便的实现在 SLEEP 和 HOLD 模式下的外部通信唤醒功能。

芯片最多提供两路 ISO7816 接口, 支持 2 个外部 7816 设备。

7816 接口主要特点如下:

- 两路 7816 接口与 UART3, UART4 分别复用, 通过 MODESEL 寄存器选择
- 波特率设置与 UART 波特率设置相同, 常用波特率覆盖, 最高波特率 115200
- 响应位长度支持 1、1.5 或 2bit, 发送奇偶校验支持奇、偶和固定校验
- 支持收发数据状态查询, 硬件给出接收/发送数据正确性, 给出标志
- 接收/发送中断使能分别独立, 支持错误重收发功能和重收发次数设置

## 10.2 波特率计算

在串口波特率由波特率生成器的值确定：

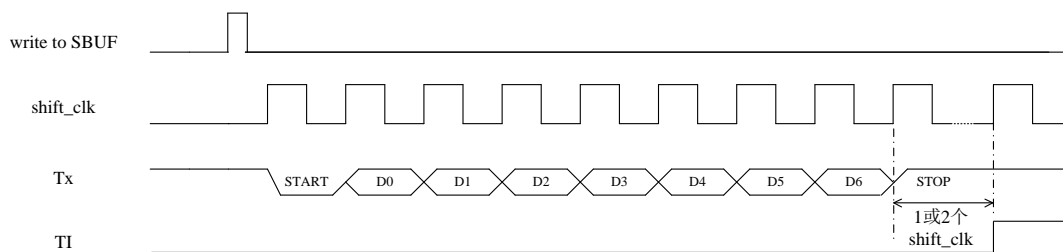
$$\text{波特率} = \frac{F_{\text{sys}}}{2 \times (SREL + 1)}$$

其中 SREL 是 16 位无符号数；F<sub>sys</sub> 是系统时钟，波特率最高值为 115200bps。

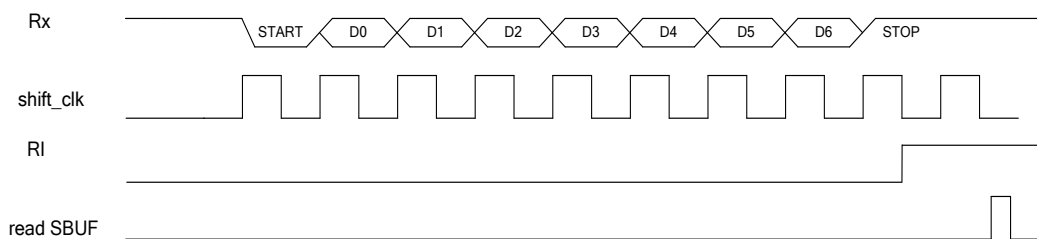
## 10.3 串口通讯模式说明

### 10.3.1 方式 1

方式 1 是一种标准的异步通信方式，每帧包含 9 或 10 位数据信息：1 位起始位（0），7 位数据位（低位在前），1 或 2 位停止位（1）。在这种方式中，TXD 引脚为数据发送端，RXD 引脚为数据接收端，其波形如下图所示：



图示：方式 1 时串行发送数据时序

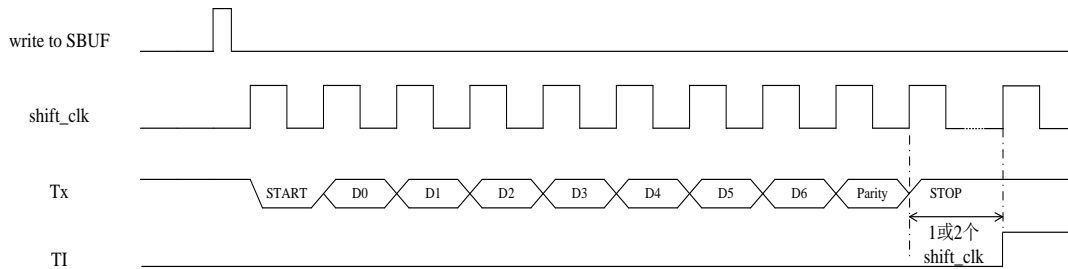


图示：方式 1 时串行接收数据时序

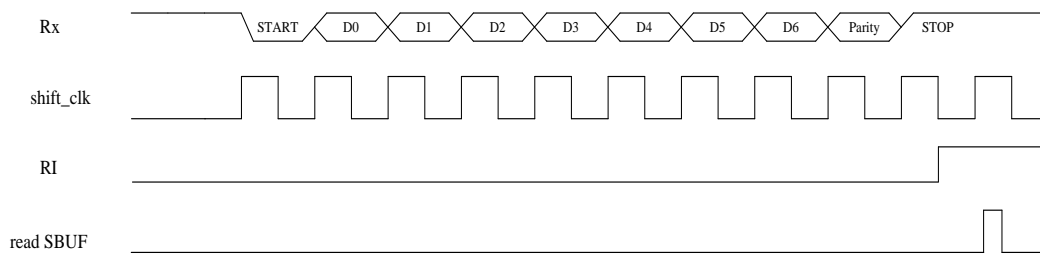
在方式 1 中，发送状态时，当一帧中最后一个数据发送完时，发送中断标志 TI 置“1”；接收状态时，接收完最后一个数据位时，接收中断标志 RI 置 1。

## 10.3.2 方式 2

方式 2 是每帧包含 10 或 11 位数据信息：1 位起始位 (0)，7 位数据位 (低位在前)，1 位奇偶校验数据位，1 或 2 位停止位 (1)。TXD 引脚为数据发送端，RXD 引脚为数据接收端，其波形如下图所示：



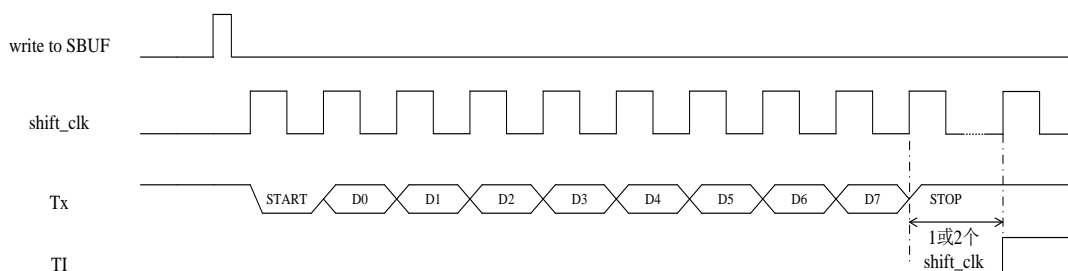
图示：方式 2 时串行发送数据时序



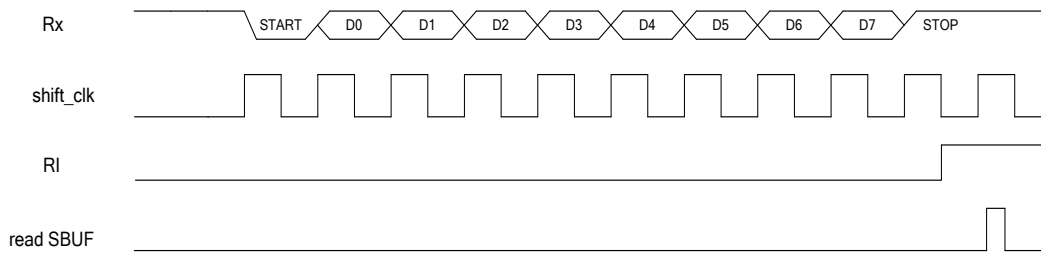
图示：方式 2 时串行接收数据时序

## 10.3.3 方式 3

方式 3 是一种标准的异步通信方式，每帧包含 10 或 11 位数据信息：1 位起始位 (0)，8 位数据位 (低位在前)，1 或 2 位停止位 (1)。在这种方式中，TXD 引脚为数据发送端，RXD 引脚为数据接收端，其波形如下图所示：



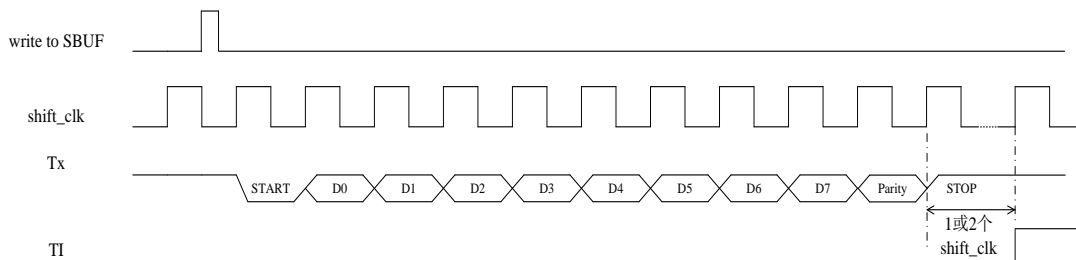
图示：方式 3 时串行发送数据时序



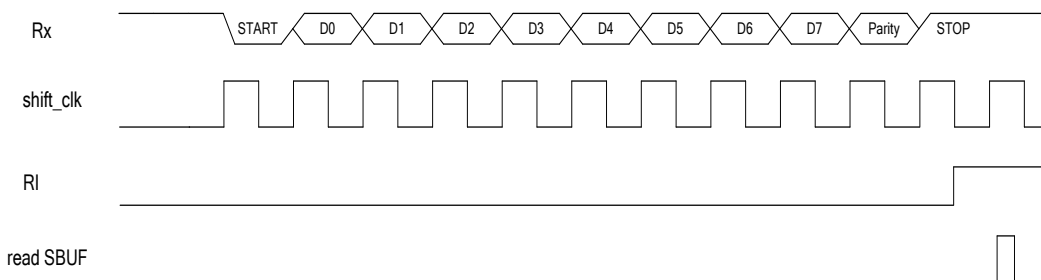
图示：方式 3 时串行接收数据时序

## 10.3.4 方式 4

方式 4 是使用第 9 位数据的通信方式，每帧包含 11 或 12 位数据信息：1 位起始位 (0)，8 位数据位 (低位在前)，1 个奇偶校验或自定义数据位，1 或 2 位停止位 (1)。TXD 引脚为数据发送端，RXD 引脚为数据接收端，其波形如下图所示：



图示：方式 4 时串行发送数据时序



图示：方式 4 时串行接收数据时序

## 10.4 7816 接收和发送

### 10.4.1 7816 数据发送

对数据缓冲寄存器SBUF3/4进行写操作即可以启动一次发送数据流程，该流程包括几个步骤。

1. 发送起始位(0); (第1ETU)
2. 发送 8bit 数据位; (第2-9ETU)
3. 发送 1bit 校验位; (第10ETU)
4. 读取接收到的CKACK信号; 如果CKACK=0, TX\_PAR置为“1”, 如果CKACK=1, TX\_PAR置为“0”; (第11ETU)
5. 处于发送等待状态2个ETU;
6. 此时一个数据帧发送完成, SDIF=“1”, 如果SDIE=1, 则此状态结束时, 产生发送中断。  
如果CKACK=1, 或禁止自动重发(AUTOSD=0), 则UART接口回到IDLE 状态。  
如果CKACK=0, 且使能自动重发(AUTOSD=1), 则UART接口进入重发等待状态。
7. 重新开始发送上一帧数据。

### 10.4.2 7816 数据接收

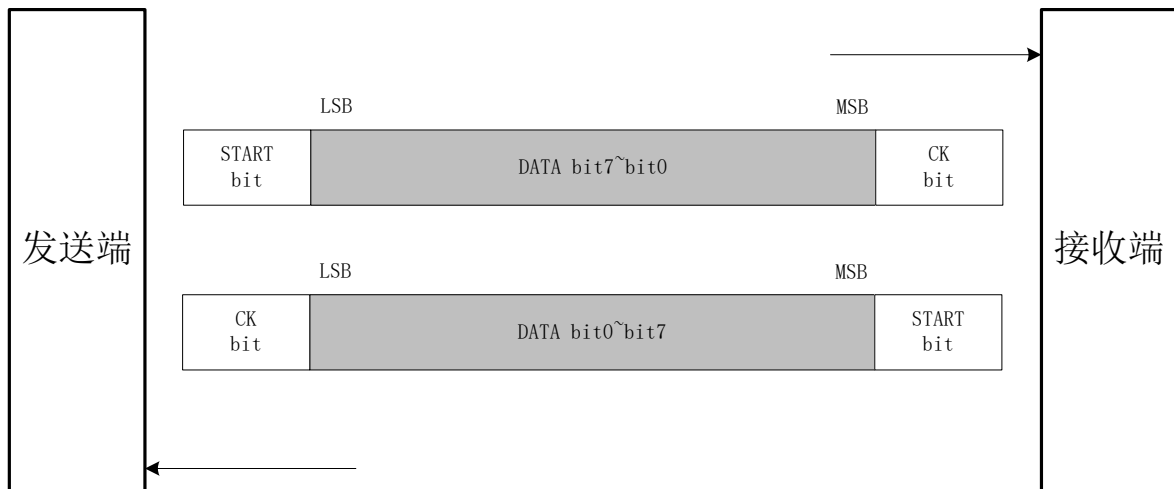
在IDLE状态下, 如果在接收端口(IO)上检测到下降沿, 即启动一次接收数据流程。该流程包括几个步骤, 每一步均需要一个或者几个ETU。

1. 接收起始位(0); (第1ETU)
2. 接收 8bit 数据位; (第2-9ETU)
3. 接收 1bit 校验位; (第10ETU)
4. 向发送端发送CKACK信号。如果校验正确, 或者禁止自动重收(AUTORC=0), 则发送1, 否则发送0。(CKACK的宽度, 可以通过ACKLEN来配置)

校验位	AUTORC	IO1
正确	“0” 禁止自动重收	“1”
正确	“1” 使能自动重收	“1”
错误	“0”	“0”
错误	“1”	“0”

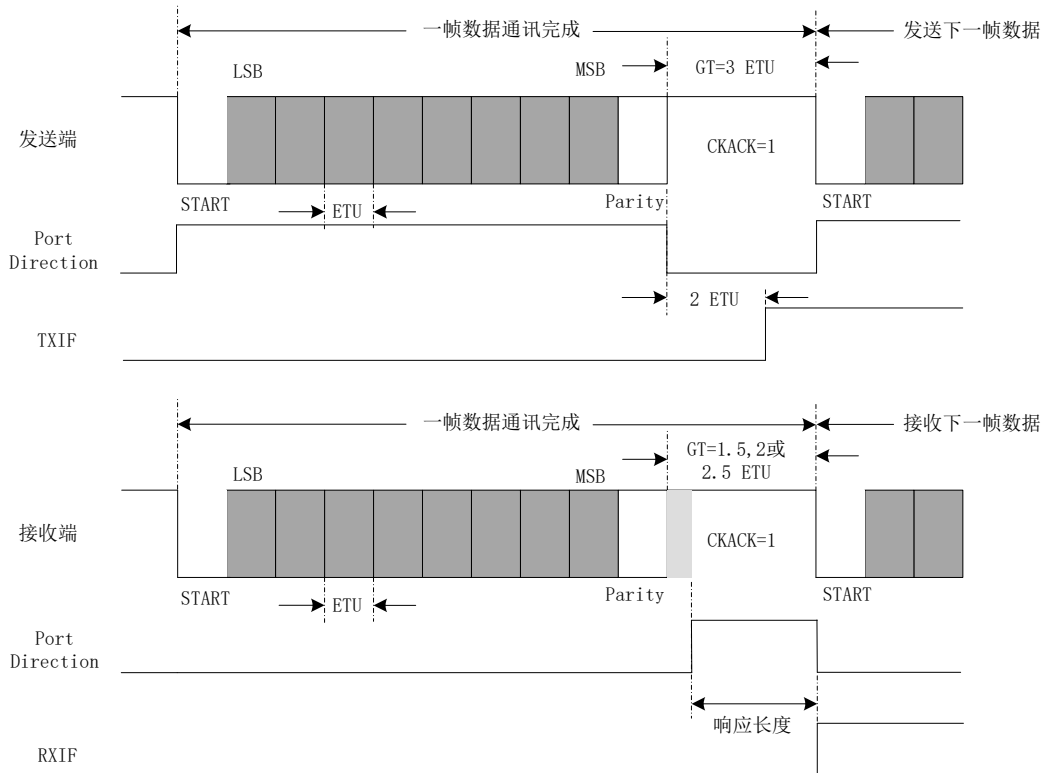
5. 此状态结束时, 回到 IDLE 状态, 产生接收中断。在中断中判断, 如果校验正确, 读取接收SBUF中的数据。(在接收的过程中, 如果程序有对SBUF3/4写入的动作, 该写入是无效的, 需要等待)

## 10.4.3 7816 通讯示意图



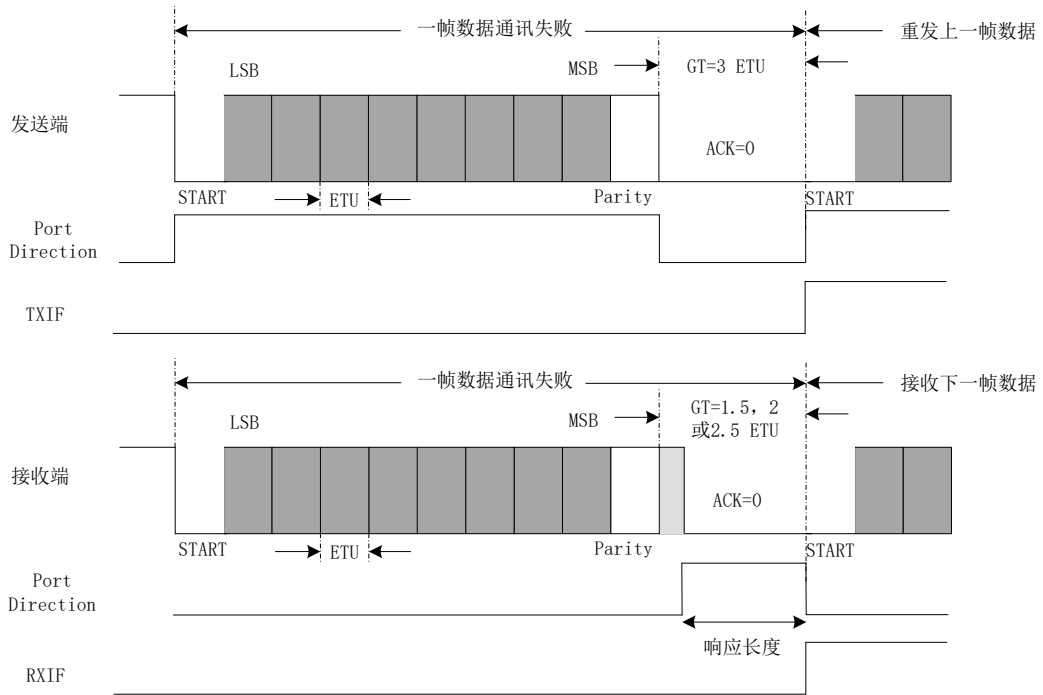
7816 通讯数据示意图

进行通信时，收发一个bit 的时间被定义为基础时间单位ETU (Elementary Time Unit)。发送端发送完一个数据帧后，接收端需要一定的时间对接收到的数据进行校验，然后再根据校验结果发送下一个数据帧或重发上一帧数据，即，连续发送两个数据帧之间的等待时间，该等待时间被定义为检测时间GT(Guarding Time)，一般， $1 GT = 3 ETU$ 。

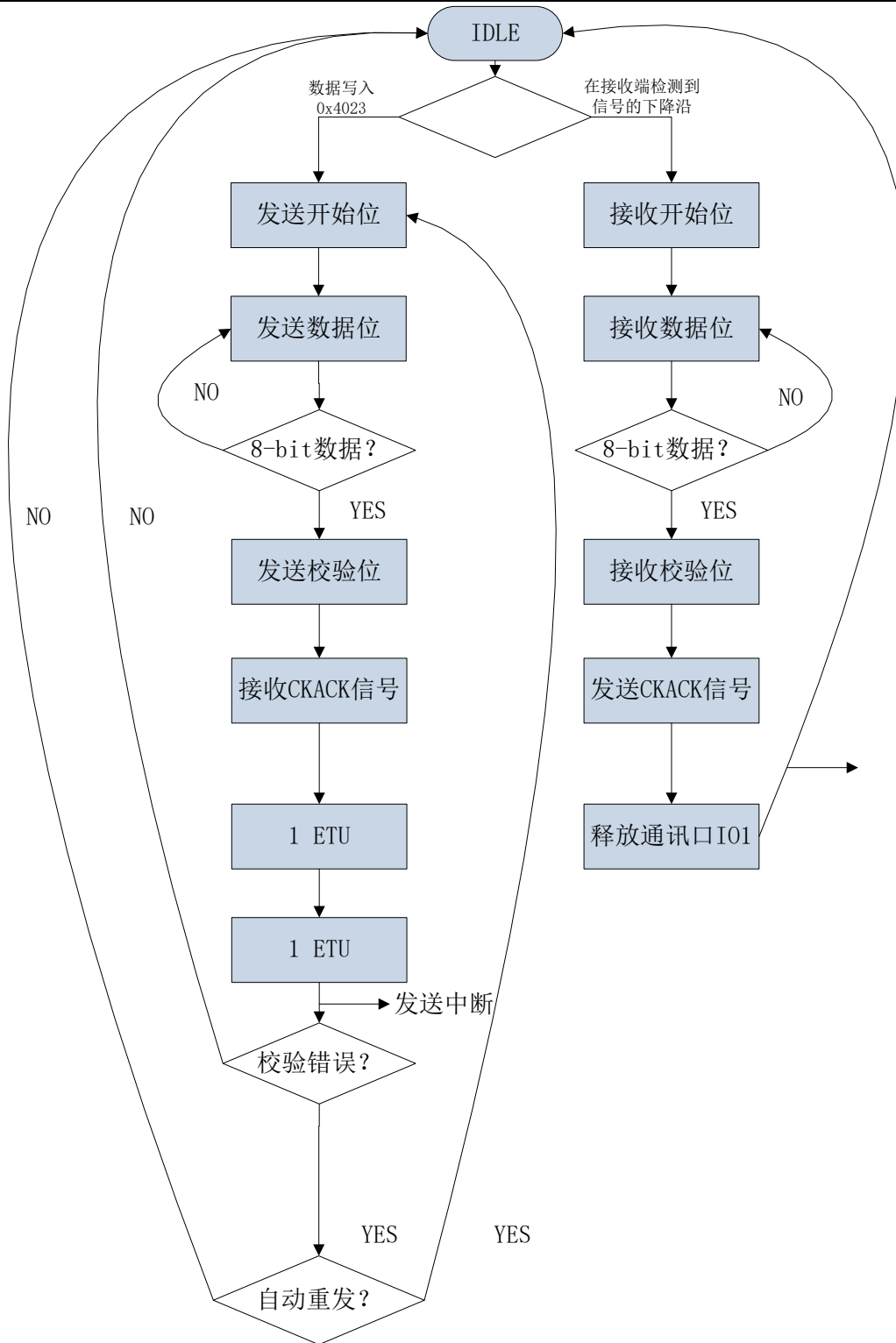


收发数据成功时序





收发数据失败时序



## 10.5 特殊功能寄存器列表

UART模块寄存器基地址： 0x40005000 (UART0端口)； 0x40008000 (UART3端口)； 0x40009000 (UART4端口)；				
偏移地址	名称	读写方式	复位值	功能描述
0x00	MODESEL	R/W	0x0000	串口功能选择寄存器
0x04	UARTCON	R/W	0x0000	UART 功能配置寄存器
0x08	ISO7816CON	R/W	0x0000	7816 功能配置寄存器
0x0C	SREL	R/W	0x0000	串口波特率发生寄存器
0x10	SBUF	R/W	0x0000	串口数据缓冲寄存器
0x14	UARTSTA	R/W	0x0000	UART 状态寄存器
0x18	ISO7816STA	R/W	0x0000	7816 状态寄存器
0x30	IRCON	R/W	0x0000	红外调制控制寄存器
0x34	IRDUTY	R/W	0x0000	红外调制脉宽调整寄存器

注：7816 的相关寄存器只对 UART3 与 UART4 的基地址有效

## 10.6 特殊功能寄存器说明

MODESEL (串口功能选择寄存器)		基地址： 0x40005000/0x40008000/0x40009000 偏移地址： 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	Mode
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Mode	串口功能选择控制位： 0: UART 功能 1: 7816 功能

UARTCON (UART 功能配置寄存器)			基地址: 0x40005000/0x40008000/0x40009000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	NEGFIX	UNEG	STOPSEL
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	LENSEL	PARITYSEL[1:0]		PARITYEN	RXIE	TXIE	RXEN	TXEN
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
NEGFIX	<b>UART 通讯中的逻辑锁定:</b> 0: 由用户软件控制 UART 通讯是正逻辑还是负逻辑, 通过 UNEG 控制位 1: 由硬件自动配置 UART 通讯是正逻辑还是负逻辑, UNEG 控制位无效
UNEG	<b>UART 通讯中的正逻辑或者是负逻辑</b> 0: 正逻辑 (默认) 1: 负逻辑
STOPSEL	<b>UART 通讯停止位长度选择位</b> 0: 1bit 1: 2bit
LENSEL	<b>UART 通讯数据长度选择位</b> 0: 8bit 1: 7bit
PARITYSEL	<b>UART 奇偶校验选择位</b> 00: 固定为 0 01: 奇校验 10: 偶校验 11: 固定为 1
PARITYEN	<b>UART 奇偶校验使能位</b> 0: 禁止 1: 使能
RXIE	<b>UART 接收中断使能位</b> 0: 禁止 1: 使能
TXIE	<b>UART 发送中断使能位</b> 0: 禁止 1: 使能
RXEN	<b>UART 接收使能位</b> 0: 禁止 1: 使能
TXEN	<b>UART 发送使能位</b> 0: 禁止

	1: 使能
--	-------

IS07816CON (7816 功能配置寄存器)			基地址: 0x40008000—0x40009000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	7816PARITY	AUTORXEN
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	AUTOTXEN	REPTR1	REPTRO	ACKLEN1	ACKLENO	PRDIE	RXIE	TXIE
Write:	N							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
7816PARITY	<b>奇偶校验选择位</b> 1: 奇校验 0: 偶校验
AUTORXEN	<b>自动重接收使能位</b> 1: 使能 0: 禁止
AUTOTXEN	<b>自动重发送使能位</b> 1: 使能 0: 禁止
REPTR[1:0]	<b>自动重收重发的次数上限</b> 00: 0 次 01: 1 次 10: 2 次 11: 3 次
ACKLEN[1:0]	<b>响应位的长度</b> 00: 1bit 01: 1.5bit 10/11: 2bit
PRDIE	<b>上溢中断使能位</b> 1: 使能 0: 禁止
RXIE	<b>接收中断使能位</b> 1: 使能 0: 禁止
TXIE	<b>发送中断使能位</b> 1: 使能 0: 禁止

SREL (串口波特率发生寄存器)			<b>基地址:</b> 0x40005000/0x40008000/0x40009000 <b>偏移地址:</b> 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	SREL[15:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SREL[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

串口/7816波特率发生寄存器，是一个16位的波特率分频系数，其值可为0~65535之间的任一整数，最高波特率为115200。波特率计算公式：

$$\text{波特率} = \frac{F_{\text{sys}}}{2 \times (SREL + 1)}$$

SBUF (串口数据缓冲寄存器)			<b>基地址:</b> 0x40005000/0x40008000/0x40009000 <b>偏移地址:</b> 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SBUF[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

低8位有效，对寄存器SBUF写操作，则串口将开始向外传输发送缓存数据；对寄存器SBUF读操作，则串口将从串行接收缓存中读取数据。

UARTSTA (UART状态寄存器)			<b>基地址:</b> 0x40005000/0x40008000/0x40009000 <b>偏移地址:</b> 14H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	PARITY	RXIF	TXIF
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PARITY	接收时奇偶校验的状态 1: 错误 0: 正确 写 0 清零, 写 1 无效。
RXIF	接收中断标志 1: 接收数据完成, 可从寄存器 SBUF 中读出 0: 接收数据还未完成 写 0 清零, 写 1 无效。
TXIF	发送中断标志 1: 发送完成 0: 发送未完成 写 0 清零, 写 1 无效。

IS07816STA (7816 状态寄存器)		基地址: 0x40008000—0x40009000 偏移地址: 18H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	RXERRST	TXERRST	PRDIF	RXIF	TXIF
Write:				AT	AT			
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RXERRSTAT	接收数据的状态 1: 错误 0: 正确 写 0 清零, 写 1 无效。
TXERRSTAT	发送数据的状态 1: 错误 0: 正确 写 0 清零, 写 1 无效。
PRDIF	接收上溢中断标志 1: 产生上溢; 0: 未产生上溢; 写 0 清零, 写 1 无效。
RXIF	接收中断标志 1: 接收数据完成, 可从寄存器 SBUF 中读出 0: 接收数据还未完成 写 0 清零, 写 1 无效。

TXIF	<b>发送中断标志</b> 1: 发送完成 0: 发送未完成 写 0 清零, 写 1 无效。
------	---

<b>IRCON</b> (红外调制控制寄存器)			<b>基地址:</b> 0x40005000/0x40008000/0x40009000 <b>偏移地址:</b> 30H					
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	X	X	X	X	X	X	X	X
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	X	X	X	X	X	X	IRLVL	IRTX
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
IRLVL	<b>红外调制输出极性选择</b> 1: 负极性。 0: 正极性。
IRTX	<b>红外调制功能使能控制</b> 1: 使能 TX 输出的红外调制功能。 0: 关闭 TX 输出的红外调制功能。

注意: 只有 PLL\_EN=1, PLL 打开的情况下, 写 IRTX=1 的红外调制使能功能才是有效的。(否则写这两位使能无效的, 因为红外调制没有时钟源)

<b>IRDUTY</b> (红外调制脉宽寄存器)			<b>基地址:</b> 0x40005000/0x40008000/0x40009000 <b>偏移地址:</b> 34H					
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	X	X	X	X	X	X	X	X
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	X	X	X	X	X	X	IRDUTY[1:0]	
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0



位	功能描述
IRDUTY[1:0]	<b>调制波形占空比配置</b> 00: 50% 01: 25% 10: 12.5% 11: 6.25%

# 11 LCD 模块

## 11.1 概述

HT5007 (64pin) 最多可以支持 20 (Segment) \* 8 (Common) 输出。

HT5009 (48pin) 最多可以支持 14 (Segment) \* 8 (Common) 输出

主要特征如下

- 软件可编程驱动方式。(具体要根据最终的 GPIO 引脚来定)
- 20 \* 8 (64pin) | 14 \* 8 (48pin)
- 22 \* 6 (64pin) | 16 \* 6 (48pin)
- 24 \* 4 (64pin) | 18 \* 4 (48pin)
- 软件控制对比度可调
- LCD 驱动电压可选
- 1/3 Bias, 1/4 Bias 可选
- 1/4 , 1/6, 1/8 Duty 可选

## 11.2 LCD 与 GPIO 口复用表

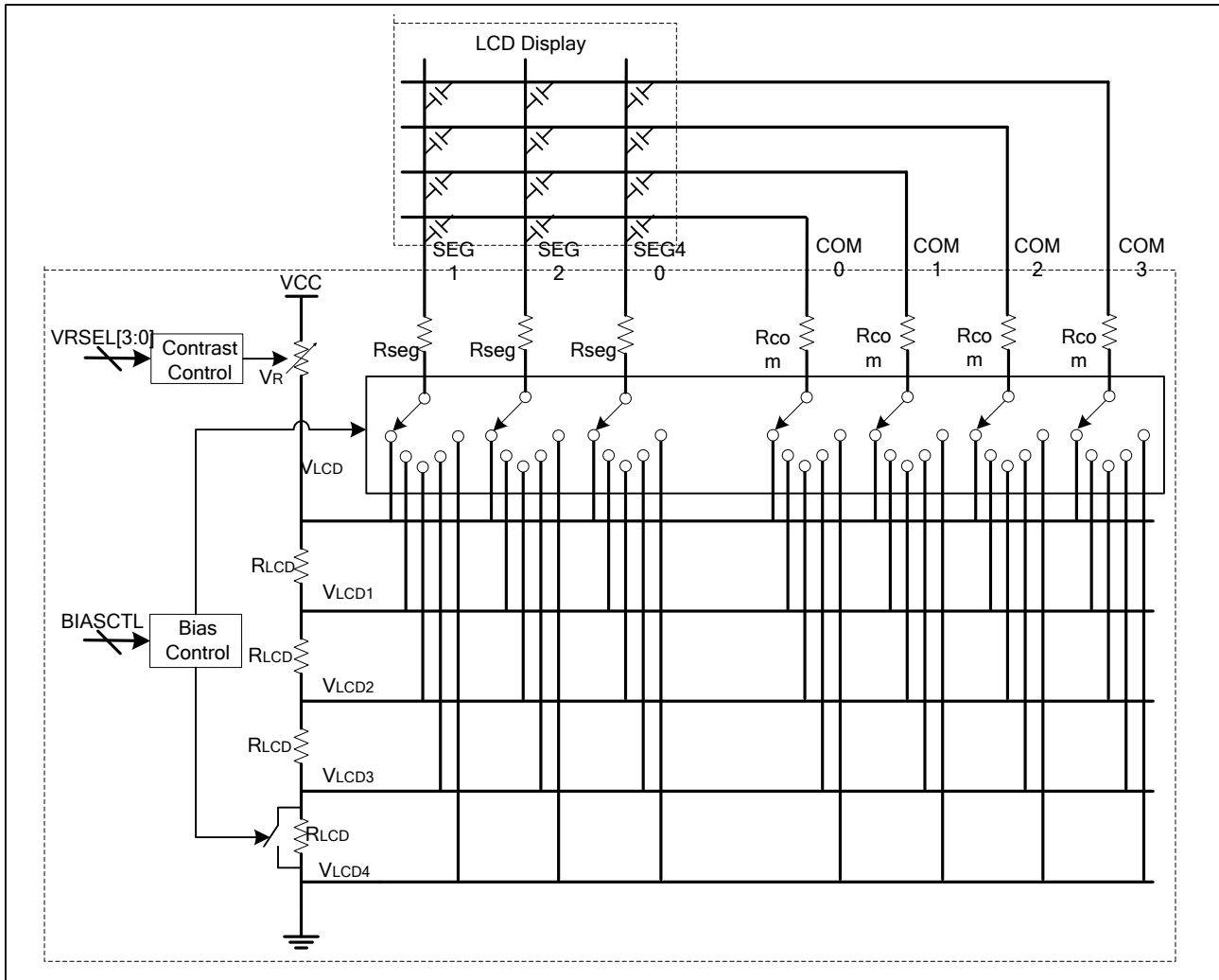
LCD 引脚定义(100PIN)	功能引脚定义	芯片引脚定义
COM0-COM3	PD. 8—PD. 11	COM0/PD. 8 - COM3/PD. 11
COM4/SEG24—COM5/SEG25	PD. 12 - PD. 13	SEG24/COM4/PD. 12 - SEG25/COM5/PD. 13
COM6/SEG26 - COM7/SEG27	PD. 14—PD. 15	SEG26/COM6/PD. 14 - SEG27/COM7/PD. 15
SEG0-SEG7	PB. 0—PB. 7	SEG0/PB. 0 - SEG15/PB. 15
SEG12-SEG15	PB. 12—PB. 15	
SEG18-SEG21	PD. 2—PD. 5	SCANOUT0/SEG18/PD. 2—SCANOUT3/SEG21/PD. 5
SEG31	PA. 3	SEG31/PA. 3
SEG33-SEG34	PA. 7—PA. 8	SEG33/INT2/PA. 7—SEG34/INT3/PA. 8
SEG35	PA. 12	SEG35/ADCIN0/PA12

## 11.3 LCD 框图

LCD 驱动单元采用 2 种工作方式: 1/3 Bias 和 1/4 Bias。LCD 的电源由 VLCD 提供, LCD 驱动电压 VLCD1、VLCD2、VLCD3、VLCD4 由内部的电阻网络生成。

LCD 电压来自 VLCD 引脚, 但是不能比 VCC 大, 即  $VLCD \leq VCC$ 。VLCD1、VLCD2、VLCD3 和 VLCD4 是 LCD 输出波形的内部偏置电压。

VRSEL[3:0]用于控制 LCD 对比度, 参考 LCD 对比度部分。



LCD 模块功能框图

LCD 的显示数据放在 LCD BUF 中，用于控制 LCD segment 的开关。当将某一段对应的 seg 和 com 都为 1 时，该段就会被点亮；否则不被点亮。

在不需要 LCD 显示时，LCD EN (CLKCTRL0.1) 写为 0，可以关闭 LCD 单元。LCD 关闭后，所有的 SEG 和 COM 都输出高电平，内部电阻分压网络以及模拟电路被关闭，LCD 单元的时钟也被关闭。

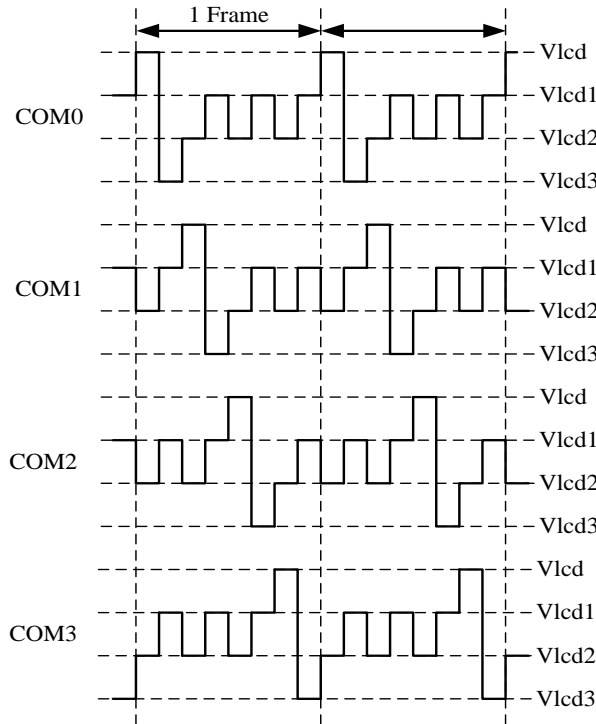
## 11.4 输出波形

LCD 输出波形的 Duty，取决于需要的 COMMON 数，提供三种 Duty：

- DUTY[1:0]=00: 1/4 duty——COM0、COM1、COM2、COM3 被使用
- DUTY[1:0]=01: 1/6 duty——COM0 到 COM5 都被使用
- DUTY[1:0]=1x: 1/8 duty——COM0 到 COM7 都被使用

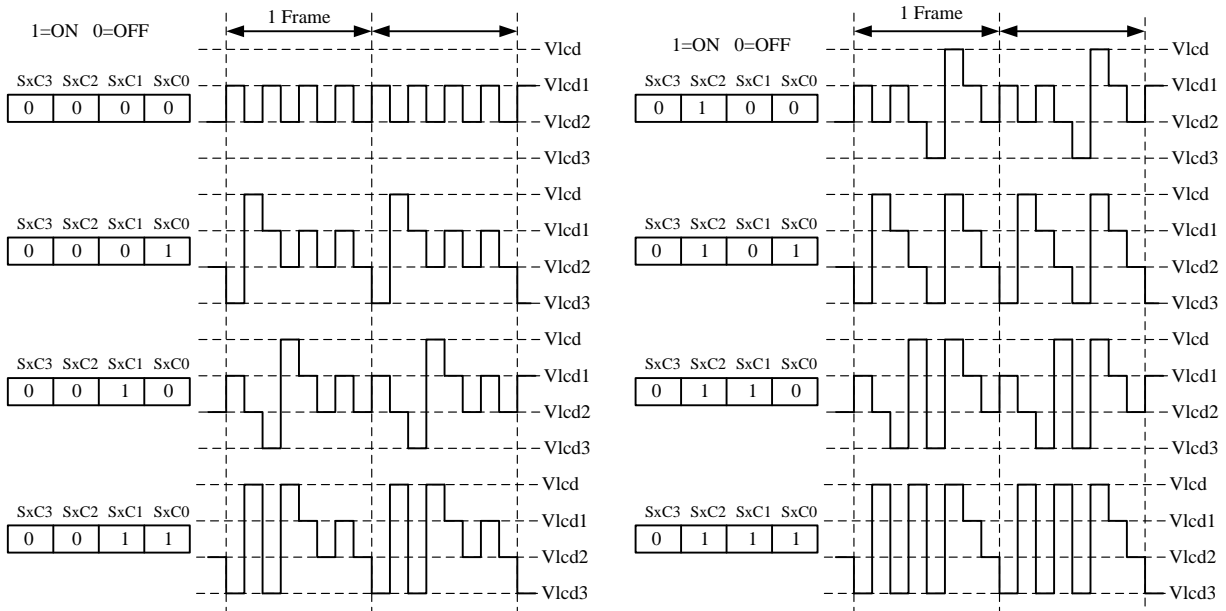
LCD 驱动电压：LCD 电压来自 VLCD， $VLCD \leq VCC$ 。VLCD1、VLCD2、VLCD3 和 VLCD4 是 LCD 输出波形的内部偏置电压。

COM 输出波形 (1/4 Duty)：使用 COM0/1/2/3 都被使用。1 帧等于 4 个 LCD 波形时钟周期。

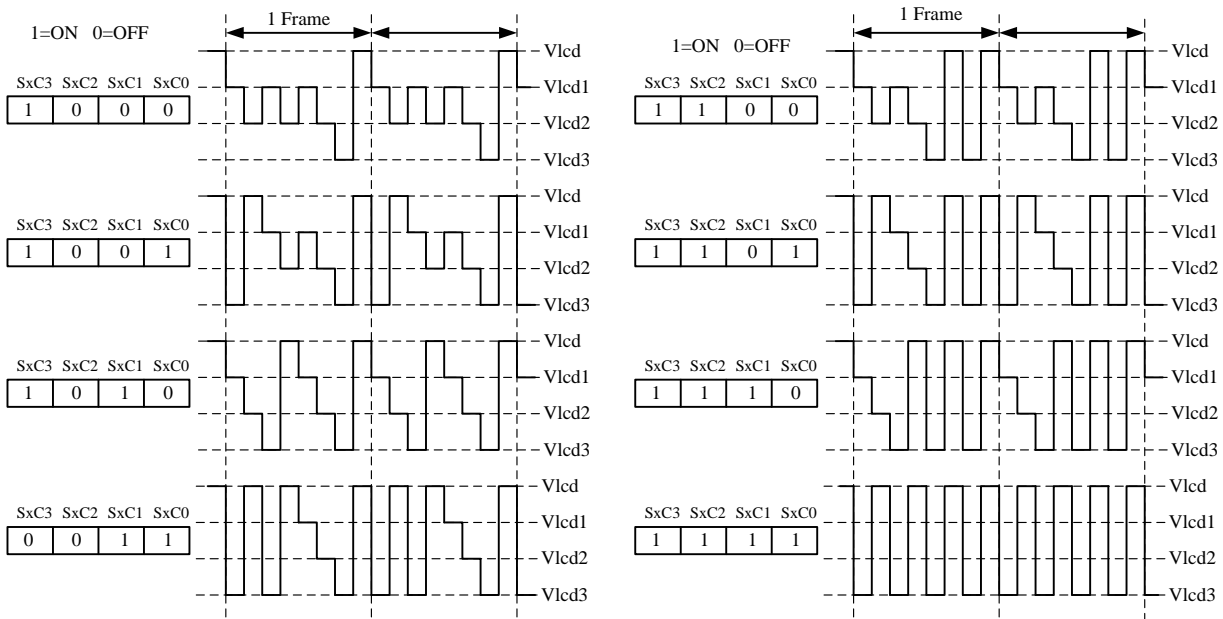


Duty=1/4 COM Output

SEGMENT 输出波形 (1/4 Duty)：

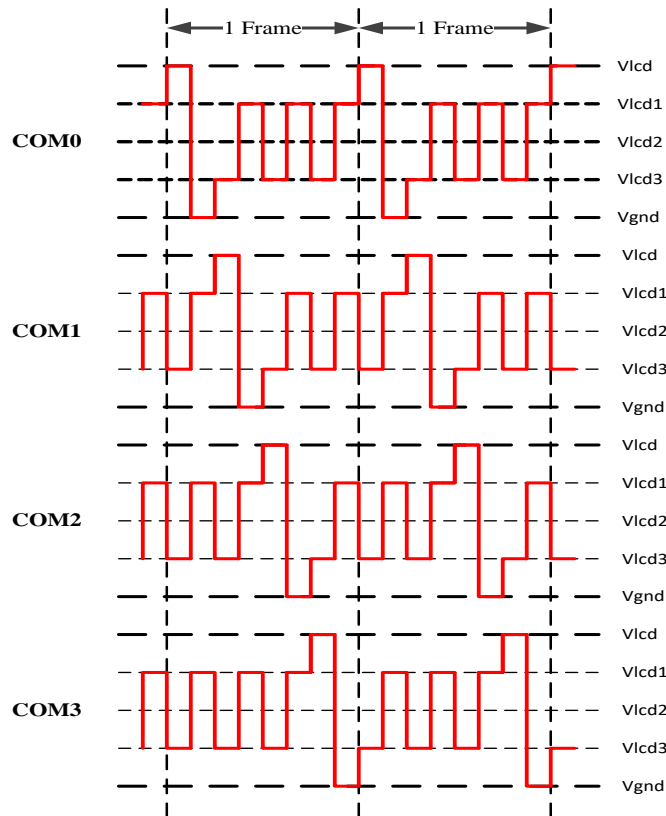


Duty=1/4 SEGx Output

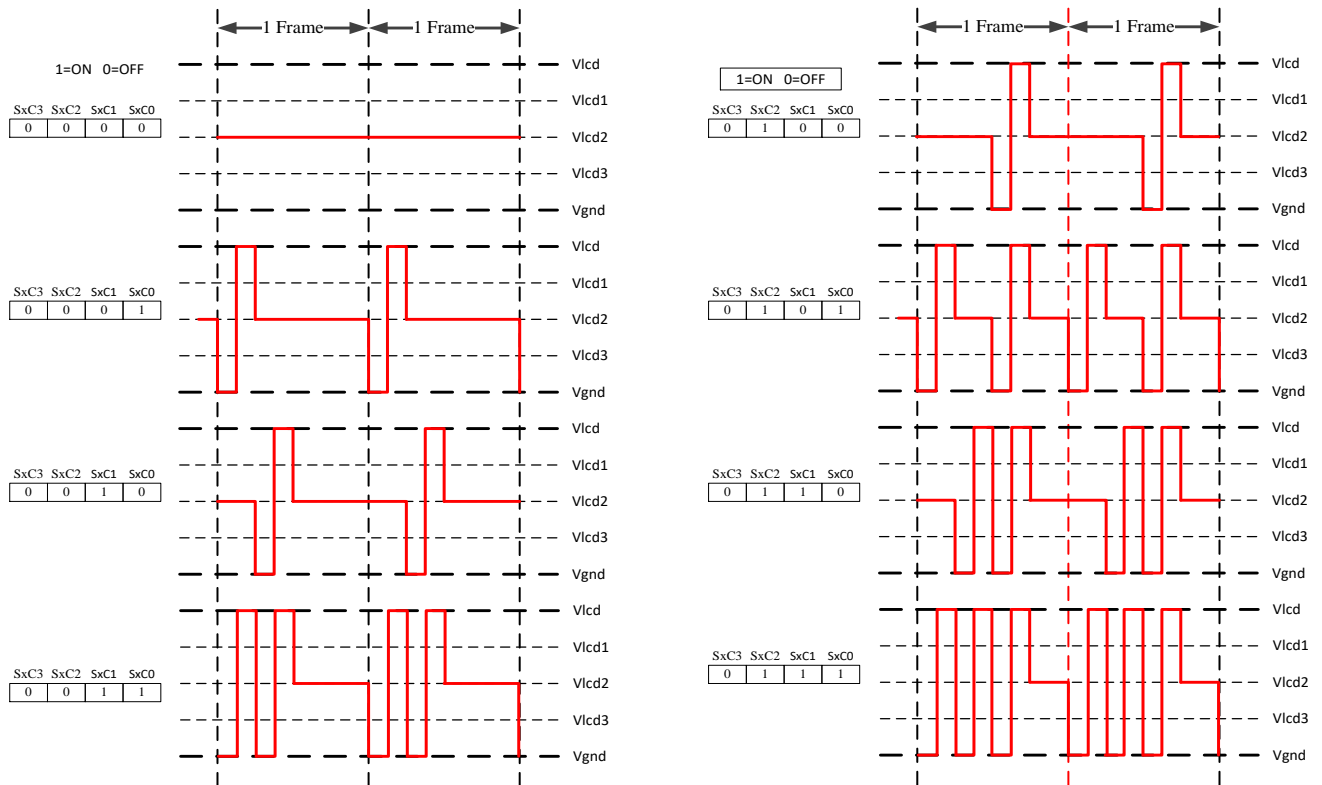


Duty=1/4 SEGx Output

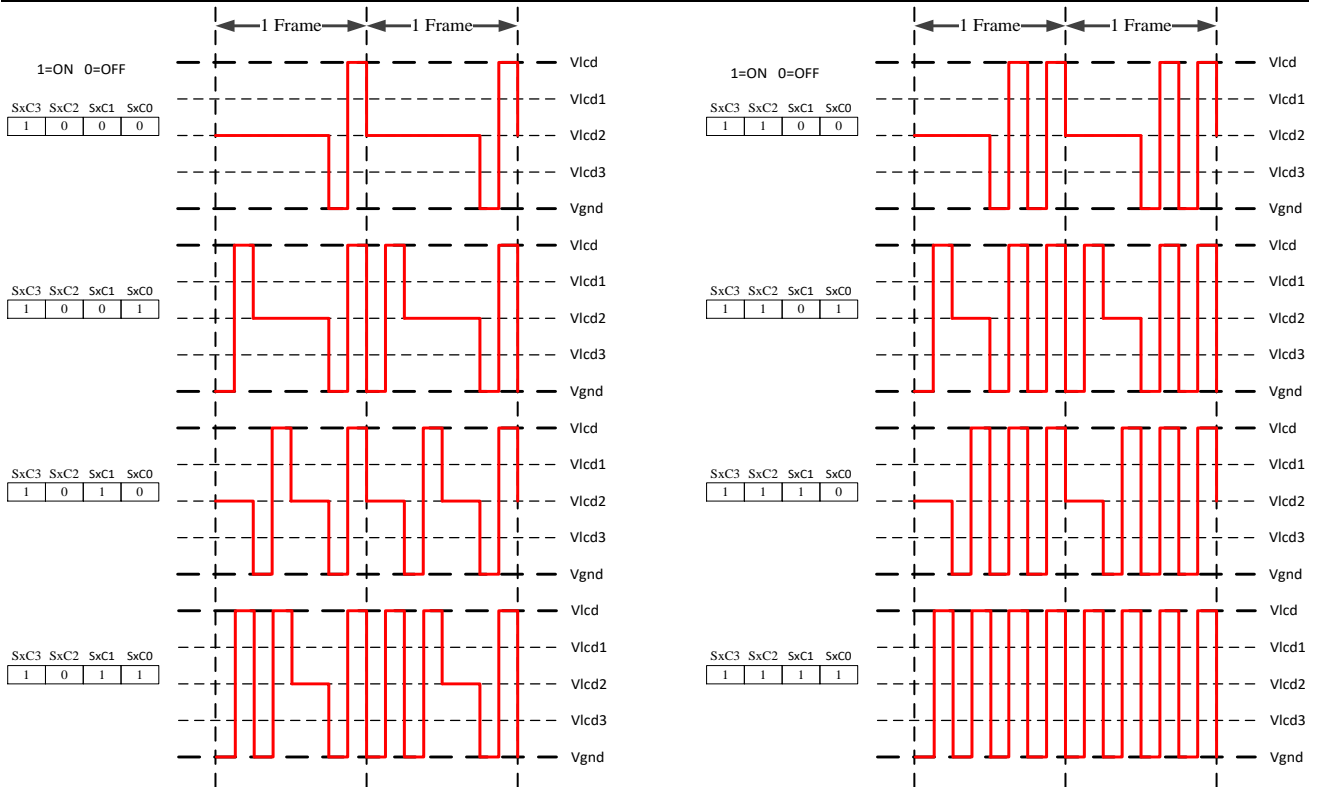
## COM 波形 (1/4Bias, 1/4Duty)



## SEGMENT 输出波形 (1/4Bias, 1/4Duty)



1/4Duty 1/4Bias SEGx Output



1/4Duty 1/4Bias SEGx Output

## 11.5 LCD 显示操作

LCD 的显示数据寄存器写功能:

LCD 单元的显示输出是采用往 LCD BUFF 内写入和对应得 SEG 输出的数据来输出显示的 LCD 段码。

40 Bytes 寄存器做为 LCD BUFF, 操作方式与一般寄存器相同, 同时与 LCD 的对应关系如下:

地址 LCD\_BUF[0]对应 SEG0 (COM7—COM0)

地址 LCD\_BUF[1]对应 SEG1 (COM7—COM0)

地址 LCD\_BUF[2]对应 SEG2 (COM7—COM0)

.....

地址 LCD\_BUF[35]对应 SEG35 (COM7—COM0)

## 11.6 特殊功能寄存器列表

LCD SFR 寄存器列表

偏移地址	名称	复位值	功能描述
<b>Base ADDR: 0x4000D000</b>			
0x00	LCDCLK	0x0080	LCD 时钟频率选择寄存器
0x04	LCDCON	0x0089	LCD 驱动控制寄存器
0x10+i×1. (i=0~42)	LCD_BUF[i]	0x0000	LCD 显示数据寄存器

## 11.7 特殊功能寄存器说明

LCD Clock Register (LCDCLKLCD 时钟频率选择寄存器)		基地址: <b>0x4000D000</b> 偏移地址: <b>00H</b>						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	BIASCTL	X	X	DUTY1	DUTY0	LCLK2	LCLK1	LCLK0
Write:								
Reset:	1	0	0	0	0	0	0	0

位	功能描述																			
BIASCTL	偏压驱动选择位 1: 表示选择1/3 bias 偏压驱动 0: 表示选择1/4 bias 偏压驱动 为了更好地提高显示效果, 在设置DUTY1:0]=01 or 1x时, 1/4duty建议选择1/3bias, 1/8 duty建议选择1/4 bias 偏压驱动, 1/6 duty根据液晶屏的参数选择1/3bias或者1/4bias。																			
DUTY[2:0]	LCD的duty控制选择位 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>DUTY1</th> <th>DUTY0</th> <th>COMMON选择</th> <th>LCD波形Duty</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Com0~Com3</td> <td>1/4 duty</td> </tr> <tr> <td>0</td> <td>1</td> <td>Com0~Com5</td> <td>1/6 duty</td> </tr> <tr> <td>1</td> <td>0</td> <td>Com0~Com7</td> <td>1/8 duty</td> </tr> </tbody> </table>				DUTY1	DUTY0	COMMON选择	LCD波形Duty	0	0	Com0~Com3	1/4 duty	0	1	Com0~Com5	1/6 duty	1	0	Com0~Com7	1/8 duty
DUTY1	DUTY0	COMMON选择	LCD波形Duty																	
0	0	Com0~Com3	1/4 duty																	
0	1	Com0~Com5	1/6 duty																	
1	0	Com0~Com7	1/8 duty																	



LCLK[2:0]	<p><b>LCD扫描频率选择位：</b></p> <p>LCD 单元的时钟来自低频时钟频率 F(1f)，F(1f)经过分频之后作为 LCD 波形扫描频率 flcd，flcd 可以通过寄存器 LCLK[2:0]进行配置（详见下表）。LCD 帧扫描频率 ffrm=flcd*duty。如：1/4 duty 时，帧扫描频率 F(frm)=flcd/4。</p>
-----------	--

LCLK[2:0]			Divide Ratio	Flcd (Hz)	frame rate (Hz)		
LCLK2	LCLK1	LCLK0			1/4	1/6	1/8
1	0	0	64	512	128	85.3	64
0	0	0	128	256	64	42.7	32
0	0	1	256	128	32	21.3	16
0	1	0	512	64	16	10.7	8
0	1	1	1024	32	8	5.3	4

注：BIASCTL，DUTY[2:0]的设置参考液晶规格！LCLK[2:0]应设为 64Hz~100Hz。

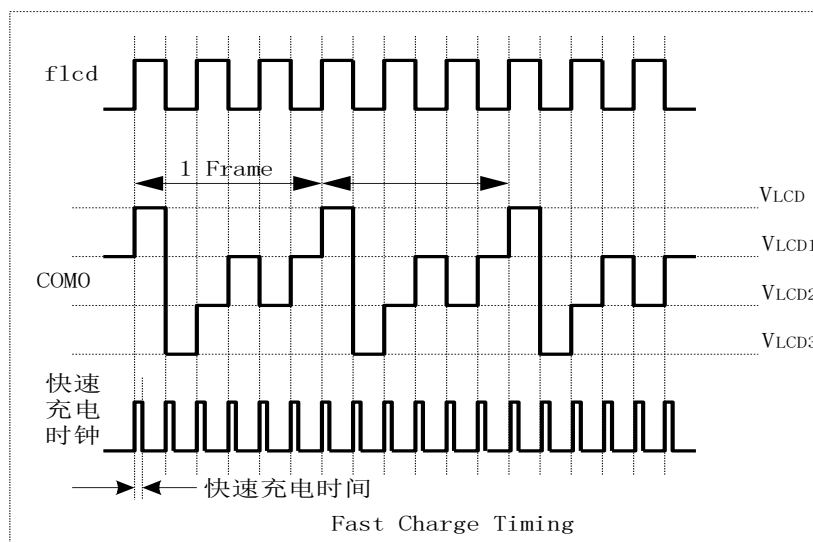
LCD Control Register (LCDCONLCD 驱动控制寄存器)	基地址： 0x4000D000 偏移地址： 04H							
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:								
Write:	FCSET1	FCSET0	FC	LC	VRSEL 3	VRSEL 2	VRSEL 1	VRSEL 0
Reset:	1	0	0	0	1	0	0	1

位	功能描述		
FCSET[1:0]	<b>快速充电时间选择位</b>		
	FCSET1	FCSET0	快速充电时间
	0	X	1/16个flcd周期
	1	0	1/32个flcd周期
	1	1	1/64个flcd周期
详见下图 5-4-5 快速充电时序图			
FC	<b>快速充电模式选择位</b>		
	<p>和LC配合使用，用于确定LCD充电模式。</p> <p>缺省状态下，LCD 分压网络的电阻 RLCD 为 <b>12KΩ</b>。根据 LCD 差异，可以选择 RLCD=<b>735KΩ</b>，此时流过电阻网络的电流较小。</p> <p>在选择RLCD=<b>735KΩ</b>时，设置控制位FC位1，可以选择快速充电模式，也就是说，每次LCD输出波形改变时，先选择RLCD=<b>12KΩ</b>进行快速充电，然后再切换到 RLCD=<b>735KΩ</b>模式。快速充电时间可以通过FCSET[1:0]进行选择。</p>		

LC	慢速充电模式选择位 和FC配合，用于确定LCD充电模式				
	FC	LC	LCD充电旧模式		
	X	0	RLCD=12KΩ，大电流充电模式		
	0	1	RLCD=735KΩ，小电流充电模式		
	1	1	短时间大电流，快速充电模式		

VRSEL[3:0]	LCD显示对比度设置位					
	VRSEL3	VRSEL2	VRSEL1	VRSEL0	1/3 bias 对比度 (% of VCC)	1/4 bias 对比度 (% of VCC)
	0	0	0	0	99.83	99.86
	0	0	0	1	93.63	95.17
	0	0	1	0	88.13	90.83
	0	0	1	1	83.27	86.9
	0	1	0	0	78.87	83.27
	0	1	0	1	74.93	79.97
	0	1	1	0	71.37	76.67
	0	1	1	1	68.17	74.07
	1	0	0	0	65.13	71.37
	1	0	0	1	62.47	68.93
	1	0	1	0	59.97	66.63
	1	0	1	1	57.67	64.5
	1	1	0	0	55.53	62.47
	1	1	0	1	53.57	60.6
	1	1	1	0	51.7	58.8
1	1	1	1	50	57.13	



快速充电时序图



## 12 WDT 模块

### 12.1 概述

看门狗只有复位功能，没有中断功能，看门狗的开关会受到两个信号控制：

(1) 当芯片处于调试模式（TEST=0, JTAGWDTEN=0）时，看门狗一定被关闭；当芯片处于正常工作模式（TEST=1）时，看门狗是否关闭受到低频 RC 的开关控制

(2) 低频 RC 的开关由寄存器 ControlByFlash 的 LRC\_CTRL 位控制，而 LRC\_CTRL 的开关控制和 Hold 模式以及 Sleep 模式都不关联，向 LRC\_CTRL 写 0 关闭低频 RC，需要更多的复杂逻辑才能有效，具体说明见 LRC\_CTRL 控制

Watchdog Timer 是一个特殊的定时器，其时钟为内部低频 RC 时钟，计时器计满预定时间则发出溢出脉冲，产生 WDT 复位信号；在溢出脉冲发生前将 Watchdog Timer 清零，则不会发出 WDT 复位。特点如下：

- 采用硬件狗设计
- 可以通过外部引脚 TEST和JTAG\_WDTEN进行控制

### 12.2 工作模式

当正常模式时WDT均被使能；当调试模式时关闭WDT计数，并且将WDT\_CNT进行清零。

TEST	JTAGWDTEN	WDT 模块
0	0	关闭
1	X	是否开启根据低频 RC（LRC）的开关决定

注意：当关闭低频 RC 导致 WDT 被关闭时，需要同时对 WDT Counter 清零，确保 WDT 重新打开后，计时是从 0 开始进行。

### 12.3 特殊功能寄存器列表

WDT 模块寄存器基地值：0x40010000				
偏移地址	名称	读写方式	复位值	功能描述
0x04	WDTCLR	R/W	0x0040	看门狗喂狗与时间配置寄存器
0x08	WDCNT	R	0x0000	看门狗计数寄存器（只读）

## 12.4 特殊功能寄存器说明

WDTCLR (WDT 喂狗与时间配置寄存器)			基地址: 0x40010000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	CLR[7:0]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SET[7:0]							
Write:	SET[7:0]							
Reset:	0	1	0	0	0	0	0	0

位	功能描述
CLR[7:0]	<b>WDT 喂狗控制位:</b> 当该 8bit 写入 0xAA, 则清狗, 清除 WDT 内部计数器 WDCNT, 写入其他值无效 该高 8bit 只能写入, 不能读取, 读出值永远为 0
SET[7:0]	<b>WDT 溢出时间设置:</b> WDT 溢出时间=64ms * (1+SET[7:0]) SET[7:0]为 8 位无符号数, 由上面公式可以得出, 最短的定时时间为 64ms, 最长为 16384ms, 默认为 4160ms。该默认值是按照 LRC 为 32KHz 来计算的, 实际值应该是根据具体 LRC 的频率来计算

WDCNT (WDT 计数寄存器)			基地址: 0x40010000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	CNT[15:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CNT[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CNT[15:0]	<b>WDT 计数寄存器:</b> 指示当前 WDT 内部的计数值

## 13 定时器模块

### 13.1 定时器单元概述

HT500X 共有 4 路定时器，定时器 0、1、2、3 的时钟源即系统时钟  $F_{sys}$ ，可根据  $SYSCLK\_SEL[2:0]$  选择为：内部低频 RC 时钟 ( $F_{lrc}$ )，内部高频 RC 时钟 ( $F_{hrc}$ )，外部低频时钟 ( $F_{osc}$ ) 和内部高频时钟 ( $F_{pll}$ )。

所有定时器单元相关寄存器的配置，都需要在  $CLKCTRL1$  寄存器中使能相应的定时器模块后才能配置。

定时器主要包括以下功能：

1. 定时功能

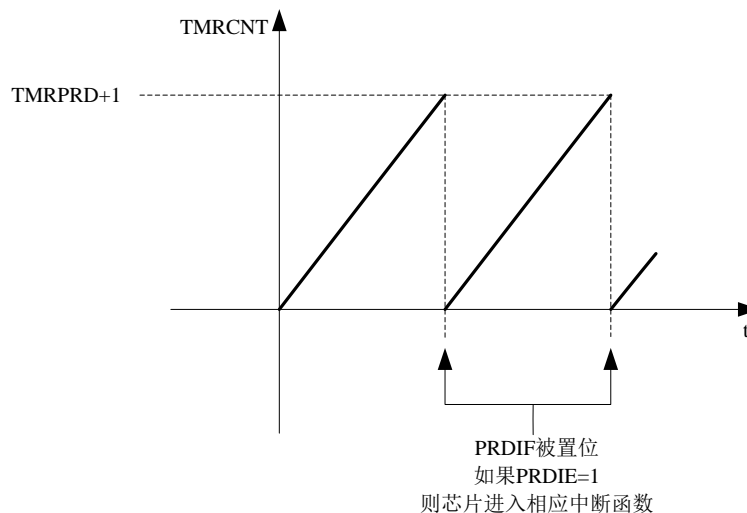
### 13.2 周期定时功能

通用定时器包括一个 16 位计数器和周期寄存器。计数器的时钟由系统时钟 ( $F_{sys}$ ) 通过定时器单元内的预分频器 ( $TMRDIV$ ) 分频得到，当使能计数器 ( $CNTEN=1$ ) 后，定时器的计数器从 0 开始计数，当计数寄存器 ( $TMRCNT$ ) 的值等于设定的周期寄存器 ( $TMRPRD+1$ ) 时会置位周期定时中断标志 ( $PRDIF=1$ )，如果使能周期定时中断 ( $PRDIE=1$ )，则会触发定时器周期中断，进入相应的周期中断服务程序。

当周期定时中断标志置位后 ( $PRDIF=1$ )， $TMRCNT$  的值自动清零，然后重新开始计数。

$TMRPRD$  如被修改，在完成本次定时之后下一次生效。

功能主要相关寄存器： $TMRCNT$ ， $TMRPRD$ 。



## 13.3 中断功能

### 13.3.1 定时器周期中断

当计数器的值 TMCNT 与周期寄存器的值 TMRPRD 相等时，如使能了周期中断 TMRIE. PRDIE=1，则发生周期中断，此中断在任何功能模式下都会产生。

## 13.4 特殊功能寄存器列表

TMR 模块寄存器基地址： 0x40001000 (TMR0)； 0x40002000 (TMR1)； 0x40003000 (TMR2)； 0x40004000 (TMR3)；				
偏移地址	名称	读写方式	复位值	功能描述
0x00	TMRCON	R/W	0x0000	控制寄存器
0x04	TMRDIV	R/W	0x0000	预分频寄存器
0x08	TMRPRD	R/W	0x0000	周期寄存器
0x10	TMCNT	R/*W	0x0000	计数器寄存器
0x18	TMRIE	R/W	0x0000	中断使能寄存器
0x1C	TMRIF	R/W	0x0000	中断标志寄存器

## 13.5 特殊功能寄存器说明

TMRCON (定时器控制寄存器)			基地址： 0x40001000--0x40004000 偏移地址： 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	RESERVED	RESERVED	RESERVED		RESERVED	MODE[1:0]		CNTEN
Write:		D			D			
Reset:	0	0	0	0	0	0	0	0

位	功能描述
---	------

RESERVED	固定为 0
MODE[1:0]	<b>Timer 功能选择:</b> 00: 关闭 01: Reserved 10: Reserved 11: 周期定时功能
CNTEN	<b>计数器使能位:</b> 0: 关闭 1: 使能

<b>TMRDIV</b> (预分频寄存器)		<b>基地址:</b> 0x40001000--0x40004000						
		<b>偏移地址:</b> 04H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRDIV[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRDIV[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMRDIV[15:0]	预分频的范围在 0-65535 之间 经预分频器后的频率等于输入频率的 1/ (TMRDIV[15:0] +1)

<b>TMRPRD</b> (周期寄存器)		<b>基地址:</b> 0x40001000--0x40004000						
		<b>偏移地址:</b> 08H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRPRD[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRPRD[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMRPRD[15:0]	该寄存器是一个 16 的周期寄存器 计数的周期寄存器和 PWM 的周期寄存器都是该寄存器 在使用任何模式功能之前, 需要设置周期寄存器。



TMRCNT (计数寄存器)			基地址: 0x40001000—0x40004000 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRCNT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRCNT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMRCNT[15:0]	计数器当前的计数值

TMRIE (定时器中断使能寄存器)			基地址: 0x40001000—0x40004000 偏移地址: 18H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	PRDIE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PRDIE	周期性溢出中断使能 0: 关闭 1: 使能

TMRIF (定时器中断标志寄存器)			基地址: 0x40001000—0x40004000 偏移地址: 1CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	PRDIF
Write:								

Reset:	0	0	0	0	0	0	0	0
--------	---	---	---	---	---	---	---	---

位	功能描述
PRDIF	周期性溢出中断标志 0: 未产生中断 1: 产生中断 (写 0 清 0)

## 14 I2C 模块

### 14.1 概述

I2C 模块的芯片引脚为 SCL/PC13 和 SDA/PC14。

I2C 模块提供一个符合 Philips I2C 总线规范的串行接口，用两根线实现设备与总线之间的数据传输，通过状态寄存器 I2CSTA 反映了 I2C 总线控制器的实时状态。

### 14.2 框图

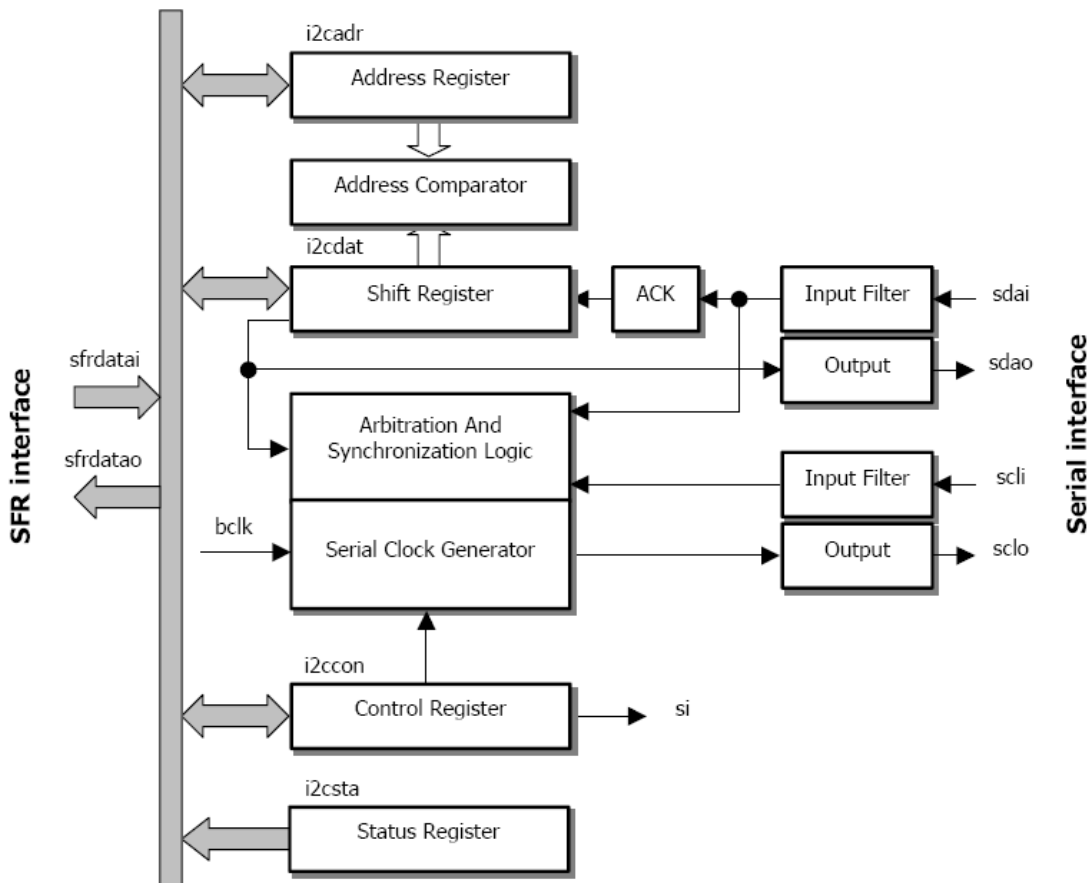


图 1 I2C 模块功能框图

### 14.3 功能描述

I2C 用两根线实现设备与总线之间的数据传输：串行时钟 SCL 和串行数据 SDA。每一个与总线相连的设备都有一个唯一的地址。I2C 是一个真正的多主机总线，它包含冲突检测和仲裁机制，以防止多个主机同时开始数据传输时的数据丢失。

### 14.3.1 操作模式

I2C 数据传输是以 8-bit 进行双向数据传输，标准模式下可达 100kbit/s 的传输速率，快速模式可达 400kbit/s 的速率。它可以下边四种模式工作：

- 主机发送模式：串行数据通过 SDA 输出，串行时钟通过 SCL 输出
- 主机接收模式：串行数据通过 SDA 输入，串行时钟通过 SCL 输出
- 从机接收模式：串行数据通过 SDA 输入，串行时钟通过 SCL 输入
- 从机发送模式：串行数据通过 SDA 输出，串行时钟通过 SCL 输入

### 14.3.2 串行时钟生成

当 I2C 处于主机模式时，可编程的时钟发生器提供 SCL 时钟；当 I2C 处于从机模式时，时钟发生器被关闭，接收来自主机的时钟。时钟发生器的输出频率可以由寄存器 I2CCON 中的位 CR[2:0]控制。其中包含 I2CCON[0...1]，I2CCON[7...14]。

### 14.3.3 中断生成

使能 ENS1，启动 I2C 模块，I2C 模块实时监测 I2C 总线状态，并根据用户设置对总线进行相应的操作及回应。当检测到总线有应用需求情况时，寄存器 I2CCON 中的标志位 SI 会被置位，并将当前应用状态写入状态寄存器 I2CSTA 中。若 I2C 中断使能打开，则产生 I2C 中断。

中断标志 SI 被置起后，时钟线 SCL 被拉低，通讯被暂停，用户需清除标志才能继续通讯。

### 14.3.4 传输模式

I2C 数据传输是以 8-bit 进行双向数据传输，标准模式下可达 100kbit/s 的传输速率。它可以下边四种模式工作：

- 主机发送模式：串行数据通过 SDA 输出，串行时钟通过 SCL 输出
- 主机接收模式：串行数据通过 SDA 输入，串行时钟通过 SCL 输出
- 从机接收模式：串行数据通过 SDA 输入，串行时钟通过 SCL 输入
- 从机发送模式：串行数据通过 SDA 输出，串行时钟通过 SCL 输入

下面将分别介绍 I2C 通讯的四种主要模式，并对所有可能的状态码进行了描述。下图中有如下缩写：

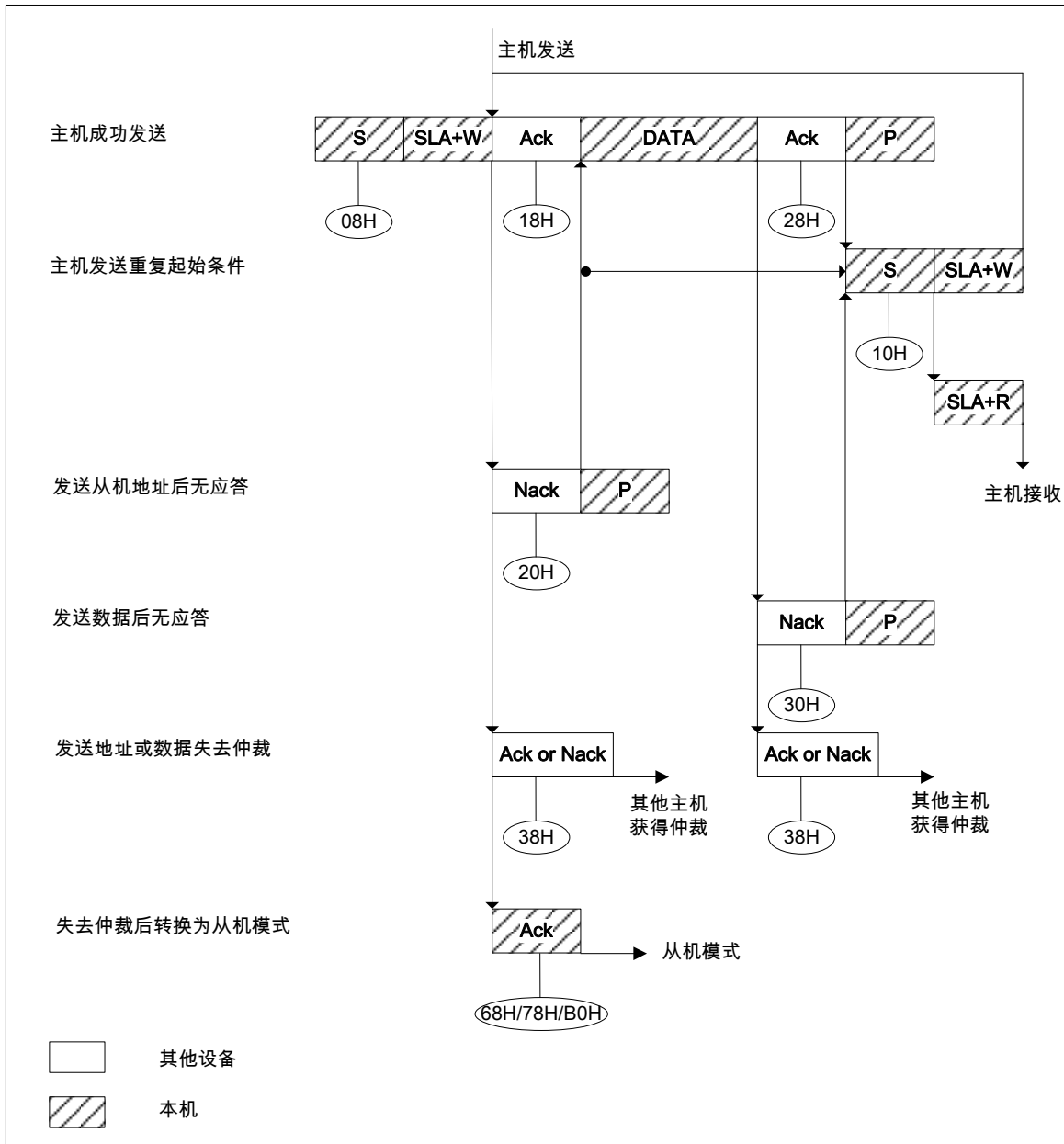
S : 开始条件  
Rs : 重复开始条件  
R : 读控制位  
W : 写控制位  
A : 应答位  
 $\bar{A}$  : 无应答位  
DATA : 8 位数据  
P : 终止条件  
SLA: 从机地址

圆形用于表示中断标志已被置起。其中的数字表示当前状态寄存器 I2CSTA 中被掩去低三位的状态码。在 SI 被清除之前，I2C 通讯会暂停，应用软件必须决定是继续通讯还是终止当前传输。对每一个状态码，所需要的软件动作和随后的传输细节均有描述。

## I2C 主机发送模式:

主机发送模式中，主机发送一系列数据到从机。一个开始条件(S)，随后一个从机地址(SLA)+写控制字(W)，表示进入主机发送模式。

状态代码	I2C 状态	应用程序配置					I2C 硬件响应
		I2CDAT	I2CCON				
			sta	sto	si	aa	
08H	起始条件已被发送	加载 SLA+W	X	0	0	X	SLA+W 将被发送 ACK 将被接收
10H	重复起始条件已被发送	加载 SLA+W	X	0	0	X	同上 SLA+R 将被发送 I2C 将转换为“主接收器”模式
		或者加载 SLA+R	X	0	0	X	
18H	SLA+W 已被发送; ACK 已被接收	加载数据字节	0	0	0	X	数据字节将被发送; ACK 将被接收 重复起始条件将被发送 终止条件将被发送; sto 标志将被复位 起始条件被发送后将再发送一个终止条件; sto 标志将被复位
		或无动作	1	0	0	X	
		或无动作	0	1	0	X	
		或无动作	1	1	0	X	
20H	SLA+W 已被发送; “not ACK”已被接收	加载数据字节	0	0	0	X	数据字节将被发送; ACK 将被接收 重复起始条件将被发送 终止条件将被发送; sto 标志将被复位 起始条件被发送后将再发送一个终止条件; sto 标志位将被复位
		或无动作	1	0	0	X	
		或无动作	0	1	0	X	
		或无动作	1	1	0	X	
28H	i2cdat 的数据字节已被发送; ACK 已被接收	加载数据字节	0	0	0	X	数据字节将被发送; 将发送 ACK 字节 重复起始条件将被发送。 终止条件将被发送; sto 标志将被复位 起始条件被发送后将再发送一个终止条件; sto 标志将被复位
		或无动作	1	0	0	X	
		或无动作	0	1	0	X	
		或无动作	1	1	0	X	
30H	i2cdat 的数据字节已被发送	数据字节	0	0	0	X	数据字节将被发送; ACK 将被接收 重复起始条件将被发送; 终止条件将被发送; sto 标志将被复位 起始条件被发送后将再发送一个终止条件; sto 标志将被复位
		或无动作	1	0	0	X	
		或无动作	0	1	0	X	
		或无动作	1	1	0	X	
38H	SLA+R/W 或数据字节仲裁失败	无动作	0	0	0	X	I2C 总线将被释放; 将进入“未寻址从机”状态; 当总线空闲时将发送一个起始条件
		或无动作	1	0	0	X	

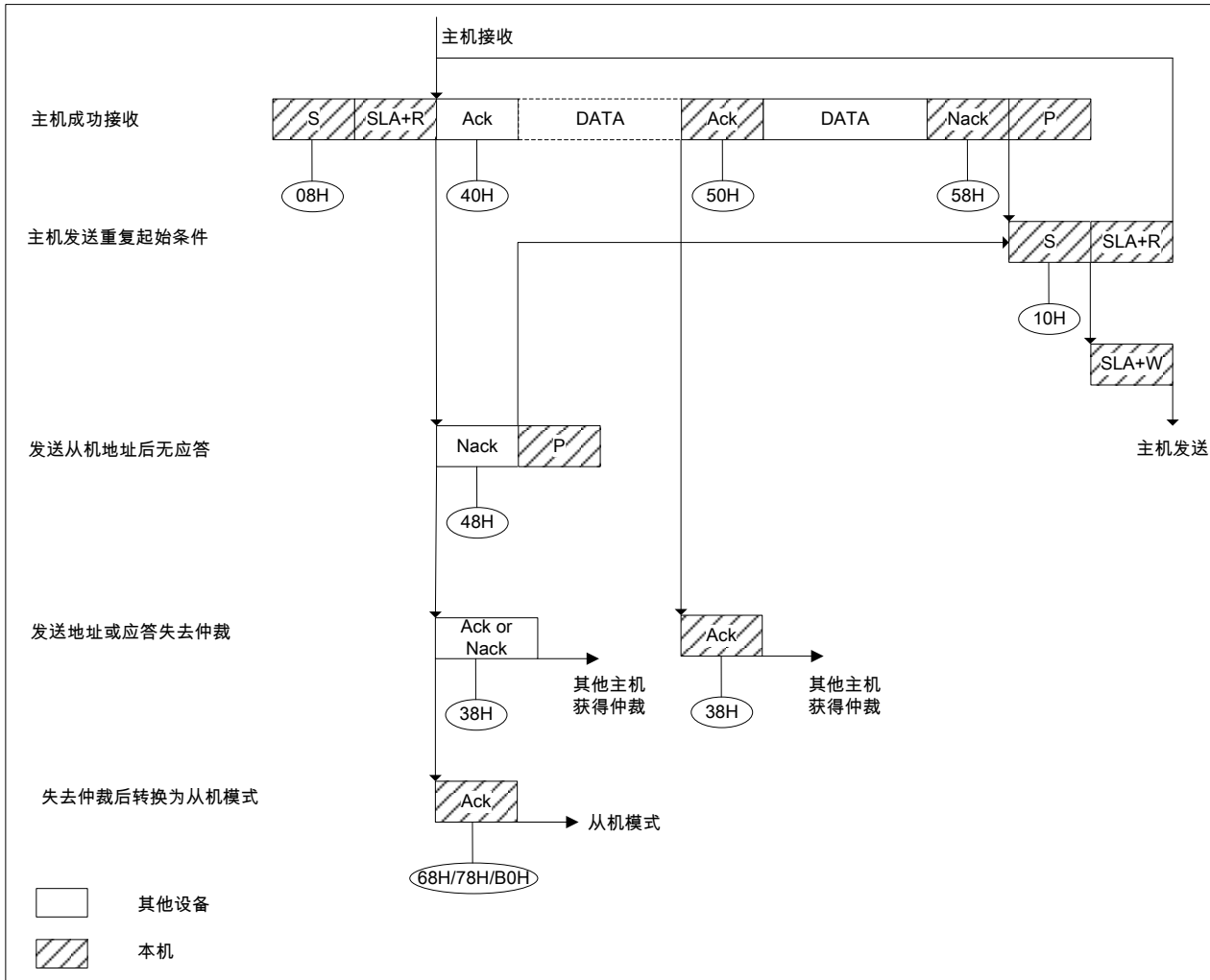


### I2C 主机接受模式:

主机接收模式中，主机从从机接收一系列数据。一个开始条件 (S)，随后一个从机地址 (SLA)+读控制

字(R)表示进入主机接收模式。

状态代码	I2C 状态	应用程序配置					I2C 硬件响应
		I2CDAT	I2CCON				
			sta	sto	si	aa	
08H	起始条件已被发送	加载 SLA+R	X	0	0	X	SLA+R 将被发送; ACK 将被接收
10H	重复起始条件已被发送	加载 SLA+R	X	0	0	X	同上 SLA+W 将被发送; I2C 将转换为“主接收器”模式
		或者加载 SLA+W	X	0	0	X	
38H	“not ACK”位仲裁失败	无动作	0	0	0	X	I2C 总线将被释放; I2C 将会进入“从机”模式 当总线空闲时将发送一个起始条件
		或者无动作	1	0	0	X	
40H	SLA+R 已被发送; ACK 已被接收	无动作	0	0	0	0	数据字节将被接收; 将返回“not ACK” 数据字节将被接收; 将返回“not ACK”
		或者无动作	0	0	0	1	
48H	SLA+R 已被发送; “not ACK”已被接收	无动作	1	0	0	X	重复起始条件将被发送 终止条件将被发送; sto 标志将被复位 起始条件被发送后将再发送一个终止条件 ; sto 标志将被复位
		或无动作	0	1	0	X	
		或无动作	1	1	0	X	
50H	数据字节已被接收; 已返回 ACK	读取数据字节	0	0	0	0	数据字节将被接收; 将返回“not ACK” 数据字节将被接收; 将返回 ACK
		或读取数据字节	0	0	0	1	
58H	数据字节将被接收; 已返回“not ACK”	读取数据字节	1	0	0	X	重复起始条件将被发送 终止条件将被发送; sto 标志将被复位 起始条件被发送后将再发送一个终止条件 ; sto 标志将被复位
		或读取数据字节	0	1	0	X	
		或读取数据字节	1	1	0	X	



## I2C 从机接受模式:

从机接收模式中，从机从主机接收一系列数据。

进入从机模式前，需设置从机地址， I2CADR 中 I2CADR[7..1]位为从机地址。如果 I2CADR[0]置位，从机也将响应广播呼叫地址 (00H)；否则将不响应广播呼叫地址。

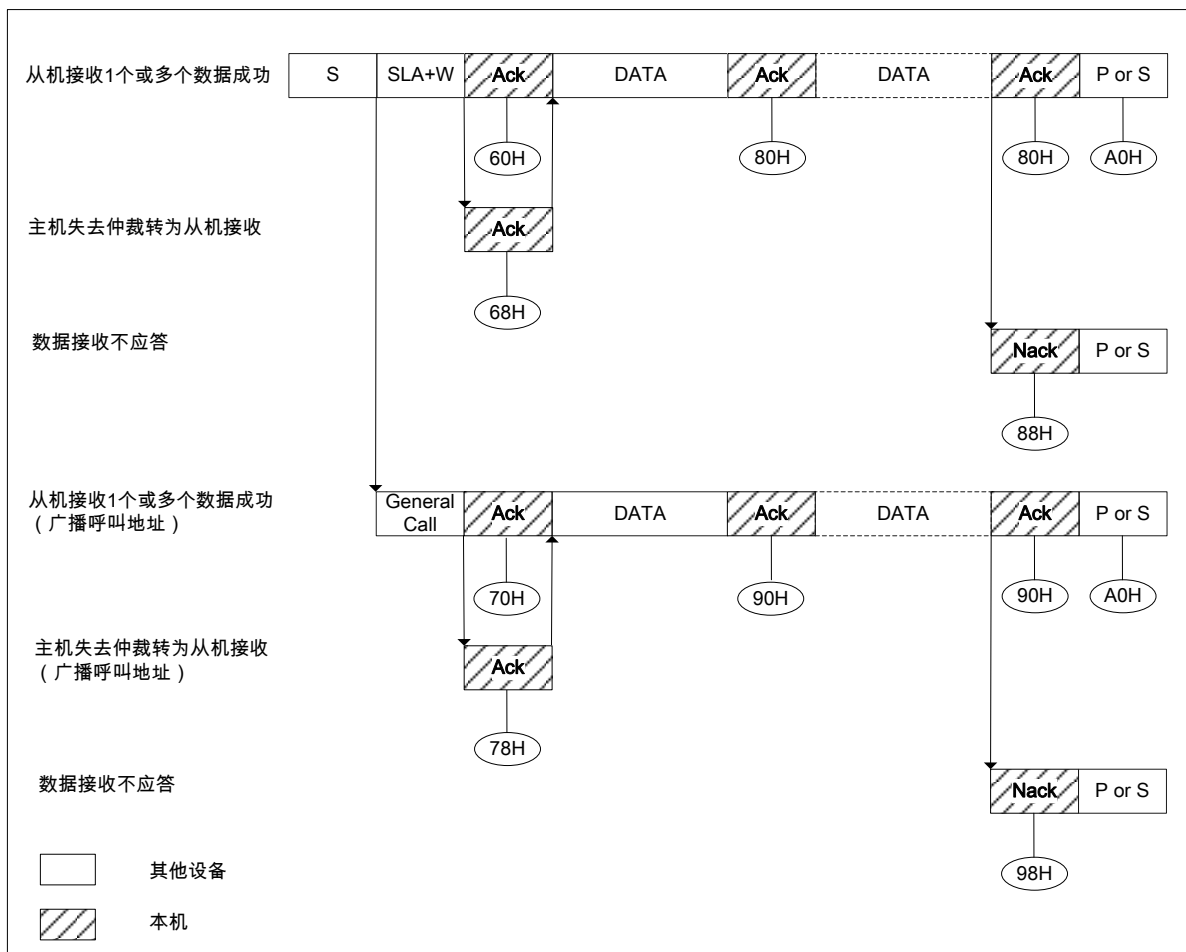
从机模式中， I2C 模块等待总线对本机地址或广播呼叫地址 (如果 I2CADR[0]被置位)的寻址。如果读写数据位是‘写’，则 I2C 进入从机接收模式，否则将进入从机发送模式。

地址和读写数据位接受完成后，中断标志 (SI) 置位，状态寄存器 I2CSTA 写入当前状态。



状态代码	I2C 状态	应用程序配置				I2C 硬件响应	
		I2CDAT	I2CCON				
			sta	sto	si		aa
60H	自身的 SLA+W 已被接收； 已返回 ACK	无动作	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或无动作	X	0	0	1	数据字节将被接收并返回 ACK
68H	主机 SLA+R/W 仲裁失败；自身的 SLA+W 已被接收，返回 ACK	无动作	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或无动作	X	0	0	1	数据字节将被接收并返回 ACK
70H	呼叫地址（00H）已被接收；已返回 ACK	无动作	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或无动作	X	0	0	1	数据字节将被接收并返回 ACK
78H	主机 SLA+R/W 仲裁失败；呼叫地址已被接收，返回 ACK	无动作	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或无动作	X	0	0	1	数据字节将被接收并返回 ACK
80H	预先写入自身 SLV 地址；DATA 字节已被接收；返回 ACK	读取数据字节	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或者 读取数据字节	X	0	0	1	数据字节将被接收并返回 ACK
88H	预先写入自身 SLA；DATA 字节已被接收；返回 “not ACK”	读取数据字节	0	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址
		或者 读取数据字节	0	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址
		或者 读取数据字节	1	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
		或者 读取数据字节	1	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
90H	预先写入呼叫地址；DATA 字节已被接收；返回 ACK	读取数据字节	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或者 读取数据字节	X	0	0	1	数据字节将被接收并返回 ACK
98H	预先写入呼叫地址；DATA 字节已被接收；返回 ACK	读取数据字节	0	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址
		或者 读取数据字节	0	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址
		或者 读取数据字节	1	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
		或者 读取数据字节	1	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件

A0H	终止条件或重复起始条件在被配置为 SLV/REC 或 SLV/TRX 时被接收	无动作	0	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址
		或者	0	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址
		或者	1	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
		无动作	1	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件



## I2C 从机发送模式:

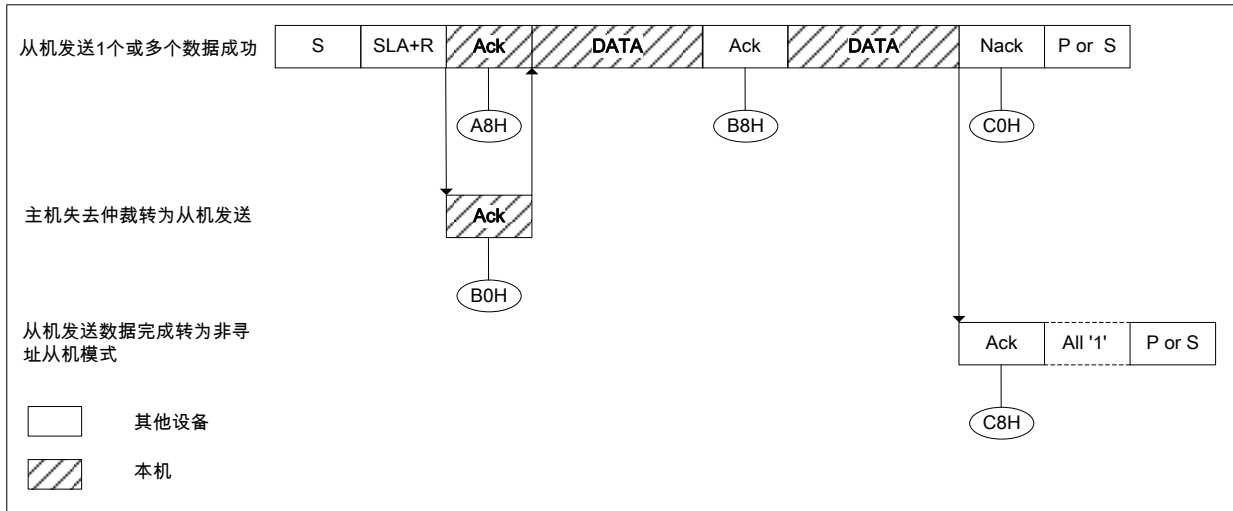
从机发送模式中，从机发送一系列数据到主机。

进入从机模式前，需设置从机地址， I2CADR 中 I2CADR[7..1]位为从机地址。如果 I2CADR[0]置位，从机也将响应广播呼叫地址 (00H)；否则将不响应广播呼叫地址。

从机模式中，I2C 模块等待总线对本机地址或广播呼叫地址(如果 I2CADR[0]被置位)的寻址。如果读写数据位是‘写’，则 I2C 进入从机接收模式，否则将进入从机发送模式。

地址和读写数据位接受完成后，中断标志(SI)置位，状态寄存器 I2CSTA 写入当前状态。

状态代码	I2C 状态	应用程序配置					I2C 硬件响应
		I2CDAT	I2CCON				
			sta	sto	si	aa	
A8H	自身 SLA+R 已被接收；返回 ACK	加载数据字节	X	0	0	0	最后一个数据字节将被发送并接收 ACK
		或者 加载数据字节	X	0	0	1	数据字节将被发送；ACK 将被接收
BOH	主机 SLA+R 仲裁失败；自身 SLA+R 已被接收；返回 ACK	加载数据字节	X	0	0	0	最后一个数据字节将被发送并接收 ACK
		或者 加载数据字节	X	0	0	1	数据字节将被发送；ACK 将被接收
B8H	数据字节已被发送；ACK 已被接收	加载数据字节	X	0	0	0	最后一个数据字节将被发送并接收 ACK
		或者 加载数据字节	X	0	0	1	数据字节将被发送；ACK 将被接收
COH	数据字节已被发送；“not ACK”已被接收	无动作	0	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址
		或者无动作	0	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址
		或者无动作	1	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
		或者无动作	1	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
C8H	最后一个数据字节已被发送；ACK 已被接收	无动作	0	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址
		或者无动作	0	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址
		或者无动作	1	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
		或者无动作	1	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件



### I2C 复合状态:

状态代码	I2C 状态	应用程序配置				I2C 硬件响应	
		I2CDAT	I2CCON				
			sta	sto	si		aa
F8H	没有可利用信息的相关状态； si=0	无动作	无动作				等待或继续进行传递
00H	MST 或选择从机模式中的总线错误	无动作	0	1	0	X	只有当被配置为“主机”或“从机”模式时 I2C 硬件才会被触发 在所有情况下，总线将被释放并且 I2C 将切换到“未寻址从机”模式。sto 标志将被复位

## 14.4 特殊功能寄存器列表

微控制器与 I2C 组件的接口通过以下四个特殊功能寄存器来实现:

I2C 模块寄存器基地址: 0x4000A000				
偏移地址	名称	读写方式	复位值	功能描述
00H	I2CDAT	R/W	0000H	I2C数据寄存器
04H	I2CADR	R/W	0000H	I2C地址寄存器
08H	I2CCON	R/W	4000H	I2C控制寄存器
0CH	I2CSTA	R/W	00F8H	I2C状态寄存器

## 14.5 特殊功能寄存器说明

I2CDAT (I2C 数据寄存器)			基地址: 0x4000A000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2CDAT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

寄存器I2CDAT是将要被传送到总线上的数据，或者是刚从总线上接收到的数据。寄存器I2CDAT没有设置影子寄存器，也没有双缓存，所以当I2C中断发生时，MCU需要及时从它读取数据，以免数据丢失。

I2CADR (地址寄存器)			基地址: 0x4000A000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2CADR[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
I2CADR[7:1]	I2C从机地址 (7位)
I2CADR[0]	呼叫地址确认位 当此位置1时，呼叫地址可以被识别，否则不能被识别。

I2CCON (控制寄存器)			基地址: 0x4000A000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	CR[9:3]						
Write:								
Reset:	0	1	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CR2	ENS1	STA	STO	SI	AA	CR[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CR[9:0]	<b>I2C时钟频率控制位</b> I2C CLOCK=f <sub>sys</sub> /( CR[9:0]+1)/4
ENS1	<b>I2C 使能位</b> 1: 打开IIC模块; 0: 关闭IIC模块;
STA	<b>开始标志位</b> 1: 检查IIC总线的状态, 如果空闲则生成开始信号; 0: 不会生成开始信号;
STO	<b>停止标志位</b> 1: 当处于主机模式, 则向总线传输停止信号 0: 不向总线传输停止信号;
SI	<b>中断标志位</b> 当进入25种IIC状态之一时, SI由硬件置位, 唯一不置位的状态是“F8h”; 写0清0, 写1无影响。
AA	<b>生成应答标志位</b> 1: 应答在以下情况下被返回: 接收到自身作为从机的地址; gc被置位的情况下接收到地址呼叫; 主机接收模式下一个字节接收完成; 从机接收模式下一个字节接收完成; 0: 非应答在以下情况下被返回: 主机接收模式下一个字节接收完成; 从机接收模式下一个字节接收完成;

I2CSTA (状态寄存器)			基地址: 0x4000A000 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2CSTA[4:0]					X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
I2CSTA[4:0]	<b>I2C状态码</b>

寄存器“i2csta”反映 I2C 模块的实时状态。这个寄存器的低三位始终为 0。总共有 26 种可能的状态。当进入 25 种状态的其中一种时, 都会产生中断; 唯一一种不产生中断的情况是状态 F8h。

在下表中, “SLA” 指从机地址, “R” 指与从机地址一起传送的读/写位是读, “W” 指与从机地址一起传送的读/写位是写。

I2C 主机发送模式状态描述如下:

状态代码	I2C 状态	应用程序配置					I2C 硬件响应
		I2CDAT	I2CCON				
			sta	sto	si	aa	
08H	起始条件已被发送	加载 SLA+W	X	0	0	X	SLA+W 将被发送 ACK 将被接收
10H	重复起始条件已被发送	加载 SLA+W	X	0	0	X	同上
		或者 加载 SLA+R	X	0	0	X	SLA+R 将被发送 I2C 将转换为“主接收器”模式
18H	SLA+W 已被发送； ACK 已被接收	加载数据字节	0	0	0	X	数据字节将被发送； ACK 将被接收
		或无动作	1	0	0	X	重复起始条件将被发送
		或无动作	0	1	0	X	终止条件将被发送；sto 标志将被复位
		或无动作	1	1	0	X	起始条件被发送后将再发送一个终止条件；sto 标志将被复位
20H	SLA+W 已被发送； “not ACK” 已被接收	加载数据字节	0	0	0	X	数据字节将被发送； ACK 将被接收
		或无动作	1	0	0	X	重复起始条件将被发送
		或无动作	0	1	0	X	终止条件将被发送；sto 标志将被复位
		或无动作	1	1	0	X	起始条件被发送后将再发送一个终止条件；sto 标志位将被复位
28H	i2cdat 的数据字节已被发送； ACK 已被接收	加载数据字节	0	0	0	X	数据字节将被发送； 将发送 ACK 字节
		或无动作	1	0	0	X	重复起始条件将被发送。
		或无动作	0	1	0	X	终止条件将被发送；sto 标志将被复位
		或无动作	1	1	0	X	起始条件被发送后将再发送一个终止条件；sto 标志将被复位
30H	i2cdat 的数据字节已被发送	数据字节	0	0	0	X	数据字节将被发送； ACK 将被接收
		或无动作	1	0	0	X	重复起始条件将被发送；
		或无动作	0	1	0	X	终止条件将被发送；sto 标志将被复位
		或无动作	1	1	0	X	起始条件被发送后将再发送一个终止条件；sto 标志将被复位
38H	SLA+R/W 或数据	无动作	0	0	0	X	I2C 总线将被释放；将进入

	字节仲裁失败 或无动作		1	0	0	X	“未寻址从机”状态； 当总线空闲时将发送一个起始条件
--	----------------	--	---	---	---	---	-------------------------------

I2C 主机接受模式状态描述如下：

状态代码	I2C 状态	应用程序配置					I2C 硬件响应
		I2CDAT	I2CCON				
			sta	sto	si	aa	
08H	起始条件已被发送	加载 SLA+R	X	0	0	X	SLA+R 将被发送； ACK 将被接收
10H	重复起始条件已被发送	加载 SLA+R	X	0	0	X	同上  SLA+W 将被发送； I2C 将转换为“主接收器”模式
		或者 加载 SLA+W	X	0	0	X	
38H	“not ACK”位仲裁失败	无动作	0	0	0	X	I2C 总线将被释放；I2C 将会进入“从机”模式 当总线空闲时将发送一个起始条件
		或者 无动作	1	0	0	X	
40H	SLA+R 已被发送； ACK 已被接收	无动作	0	0	0	0	数据字节将被接收； 将返回“not ACK” 数据字节将被接收； 将返回“not ACK”
		或者 无动作	0	0	0	1	
48H	SLA+R 已被发送； “not ACK”已被接收	无动作	1	0	0	X	重复起始条件将被发送 终止条件将被发送；sto 标志将被复位 起始条件被发送后将再发送一个终止条件；sto 标志将被复位
		或无动作	0	1	0	X	
		或无动作	1	1	0	X	
50H	数据字节已被接收； 已返回 ACK	读取数据字节	0	0	0	0	数据字节将被接收； 将返回“not ACK” 数据字节将被接收； 将返回 ACK
		或者 读取数据字节	0	0	0	1	
58H	数据字节将被接收； 已返回“not ACK”	读取数据字节	1	0	0	X	重复起始条件将被发送 终止条件将被发送；sto 标志将被复位 起始条件被发送后将再发送一个终止条件；sto 标志将被复位
		或者 读取数据字节	0	1	0	X	
		或者 读取数据字节	1	1	0	X	



I2C 从机接受模式状态描述如下：

状态代码	I2C 状态	应用程序配置					I2C 硬件响应
		I2CDAT	I2CCON				
			sta	sto	si	aa	
60H	自身的 SLA+W 已被接收； 已返回 ACK	无动作	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或无动作	X	0	0	1	数据字节将被接收并返回 ACK
68H	主机 SLA+R/W 仲裁失败；自身的 SLA+W 已被接收， 返回 ACK	无动作	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或无动作	X	0	0	1	数据字节将被接收并返回 ACK
70H	呼叫地址（00H） 已被接收；已返 回 ACK	无动作	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或无动作	X	0	0	1	数据字节将被接收并返回 ACK
78H	主机 SLA+R/W 仲裁失败；呼叫地 址已被接收，返 回 ACK	无动作	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或无动作	X	0	0	1	数据字节将被接收并返回 ACK
80H	预先写入自身 SLV 地址；DATA 字节已被接收； 返回 ACK	读取数据字节 或者	X	0	0	0	数据字节将被接收并返回 “not ACK”
		读取数据字节	X	0	0	1	数据字节将被接收并返回 ACK
88H	预先写入自身 SLA；DATA 字节已 被接收；返回 “not ACK”	读取数据字节 或者	0	0	0	0	切换为“未寻址从机”模式； 不识别自身从机地址或呼叫 地址
		读取数据字节 或者	0	0	0	1	切换为“未寻址从机”模式； 识别自身从机地址或呼叫地 址
		读取数据字节 或者	1	0	0	0	切换为“未寻址从机”模式； 不识别自身从机地址或呼叫 地址；当总线空闲时将发送 一个起始条件
		读取数据字节	1	0	0	1	切换为“未寻址从机”模式； 识别自身从机地址或呼叫地 址；当总线空闲时将发送一 个起始条件
90H	预先写入呼叫地 址；DATA 字节已 被接收；返回 ACK	读取数据字节 或者	X	0	0	0	数据字节将被接收并返回 “not ACK”
		读取数据字节	X	0	0	1	数据字节将被接收并返回 ACK

98H	预先写入呼叫地址；DATA 字节已被接收；返回 ACK	读取数据字节或者	0	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址
		读取数据字节或者	0	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址
		读取数据字节或者	1	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
		读取数据字节	1	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
AOH	终止条件或重复起始条件在被配置为 SLV/REC 或 SLV/TRX 时被接收	无动作或者	0	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址
		无动作或者	0	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址
		无动作或者	1	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
		无动作	1	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件

I2C 从机发送模式状态描述如下：

状态代码	I2C 状态	应用程序配置					I2C 硬件响应
		I2CDAT	I2CCON				
			sta	sto	si	aa	
A8H	自身 SLA+R 已被接收；返回 ACK	加载数据字节或者	X	0	0	0	最后一个数据字节将被发送并接收 ACK
		加载数据字节	X	0	0	1	数据字节将被发送；ACK 将被接收
BOH	主机 SLA+R 仲裁失败；自身 SLA+R 已被接收；返回 ACK	加载数据字节或者	X	0	0	0	最后一个数据字节将被发送并接收 ACK
		加载数据字节	X	0	0	1	数据字节将被发送；ACK 将被接收
B8H	数据字节已被发	加载数据字节	X	0	0	0	最后一个数据字节将被发送

	送; ACK 已被接收	或者 加载数据字节	X	0	0	1	并接收 ACK 数据字节将被发送; ACK 将被接收
COH	数据字节已被发送; “not ACK” 已被接收	无动作 或者	0	0	0	0	切换为“未寻址从机”模式; 不识别自身从机地址或呼叫地址
		无动作 或者	0	0	0	1	切换为“未寻址从机”模式; 识别自身从机地址或呼叫地址
		无动作 或者	1	0	0	0	切换为“未寻址从机”模式; 不识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件
		无动作	1	0	0	1	切换为“未寻址从机”模式; 识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件
C8H	最后一个数据字节已被发送; ACK 已被接收	无动作 或者	0	0	0	0	切换为“未寻址从机”模式; 不识别自身从机地址或呼叫地址
		无动作 或者	0	0	0	1	切换为“未寻址从机”模式; 识别自身从机地址或呼叫地址
		无动作 或者	1	0	0	0	切换为“未寻址从机”模式; 不识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件
		无动作	1	0	0	1	切换为“未寻址从机”模式; 识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件

I2C 复合状态:

状态代码	I2C 状态	应用程序配置				I2C 硬件响应	
		I2CDAT	I2CCON				
			sta	sto	si		aa
F8H	没有可利用信息的相关状态; si=0	无动作	无动作				等待或继续进行电流传递
00H	MST 或选择从机模式中的总线错误	无动作	0	1	0	X	只有当被配置为“主机”或“从机”模式时 I2C 硬件才会被触发 在所有情况下, 总线将被释

							放并且 I2C 将切换到“未寻址从机”模式。sto 标志将被复位
--	--	--	--	--	--	--	----------------------------------

# 15 RTC 模块

## 15.1 概述

RTC 单元提供实时时钟、日历功能，自动闰年调整，支持闹钟和周期性中断。

RTC 模块在各种工作模式下都不会被关闭，在低功耗下仍然正常运行。

RTC 输出寄存器、RTC 时钟校正寄存器不会被复位，以保持 RTC 的准确性。

RTC 模块工作电压最低可至 1v（注：因为 TBS 最低工作电压为 2v，所以如果要保证 RTC 精度，工作电压至少要达 2v）

## 15.2 功能描述

- 提供时钟和日历功能：输出寄存器包含秒、分、时、日、月、年和星期
- 具有自动闰年闰月调整功能
- 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、日、月）
- 可输出频率为 1/2/8/16/32/64/128/的方波
- 可输出每秒补偿的校验脉冲
- 增加只读寄存器，用于保存校时的次数以及最后一次校时前的时间

## 15.3 时钟校正

由 RTC 模块读取 TPS 输出的温度值，根据 OSC 的温度特性，计算实时频率偏差 dFi，送至分频模块进行时钟校正。芯片内置了一个可修改系数的多项式补偿曲线。DFi 计算公式如下：

$$DFi = (DFA + DFB * (TMPDAT - Toff) + DFC * (TMPDAT - Toff)^2 + DFD * (TMPDAT - Toff)^3 + DFE * (TMPDAT - Toff)^4) \gg 2$$

DFA / DFB / DFC / DFD / DFE 为 0 次到 4 次项补偿系数，TMPDAT 为温度传感器输出值，Toff 为温度传感器偏置校正。

DFi 每个 LSB 表示 0.06ppm(对于 OSC，每个 LSB 表示 1/512Hz)。

## 15.4 RTC 补偿系数寄存器和 Info Flash 对应关系

RTC 的补偿系数寄存器在芯片上电后会自动装载，考虑到系统的可靠性，用户可以软件读取 Information Block 的对应地址的值，然后再写入到对应的 RTC 模块的寄存器中。

寄存器偏移地址	寄存器名称	Information Block 对应偏移地址 (Information Block 基地址：
---------	-------	---

		0x00040000)
0x50	DFAH	0x104
0x54	DFAL	0x108
0x58	DFBH	0x10C
0x5C	DFBL	0x110
0x60	DFCH	0x114
0x64	DFCL	0x118
0x68	DFDH	0x11C
0x6C	DFDL	0x120
0x70	DFEH	0x124
0x74	DFEL	0x128
0x78	Toff	0x12C
0x7C	MCON01	0x130
0x80	MCON23	0x134
0x84	MCON45	0x138
----	----	0x13c

注:

1. 当 Flash 的 FC0 地址的 bit2 为 1 时，上电后才会将 InfoFlash 中的数据装载到对应的寄存器中。
2. 0x13c 存放从 0x104 到 0x138 的 32 位无符号数累加和，如果累加和正确，则说明 info 中数据有效，否则则认为 info 中数据无效。这个是方便用户软件判断。
3. MCONxx 寄存器为内部控制寄存器，上电后也要一同装载

## 15.5 时间和万年历

RTC 提供秒、分、时、日、月、年和星期输出寄存器。

通过 RTC 的输出寄存器，可以得到自动闰年校正的万年历功能，其范围从 2000 年 1 月 1 日到 2099 年 12 月 31 日。

## 15.6 中断功能

RTC 一共提供 8 种中段源，公用 MCU 的 IRQ-RTC 中断向量 12。RTC 的 8 种中断源由 RTCIE(AAH)控制其使能。

具体的中断产生条件和中断清除步骤如下:

**ALMF:** 闹钟中断标志

当小时和分钟与设定的闹钟匹配时，产生闹钟中断，ALMF 被置为 1。

对该位写 0 清标志。

**RTC1F:** RTC 定时器 1 中断标志

如设置  $RTC1CNT=X$ ，使能计数 RTC1EN 后，经过  $(X+1)*1S$  后，该标志位置位 1。

对该位写 0 清标志。

**RTC2F:** RTC 定时器 2 中断标志

如设置  $RTC2CNT=X$ ，使能计数 RTC2EN 后，经过  $(X+1)*0.0625S$  后，该标志位置位 1。

对该位写 0 清标志。

**MTHF:** 月中断

月计数器 MTHR 加 1 时，产生一个月中断，MTHF 被置为 1。

对该位写 0 清标志。

**DAYF:** 日中断

日期计数器 DAYR 加 1 时，产生一个日中断，DAYF 被置为 1。

对该位写 0 清标志。

**HRF:** 小时中断

小时计数器 HRR 加 1 时，产生一个小时中断，HRF 被置为 1。

对该位写 0 清标志。

**MINF:** 分钟中断

分钟计数器 MINR 加 1 时，产生一个分钟中断，MINF 被置为 1。

对该位写 0 清标志。

**SECF:** 秒中断

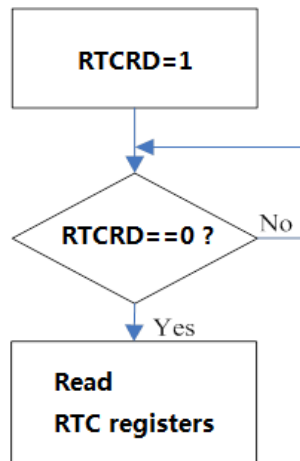
秒计数器 SECR 加 1 时，产生一个秒中断，SECF 被置为 1。

对该位写 0 清标志。

## 15.7 RTC 指示寄存器读写流程

### 15.7.1 读取 RTC 指示寄存器流程

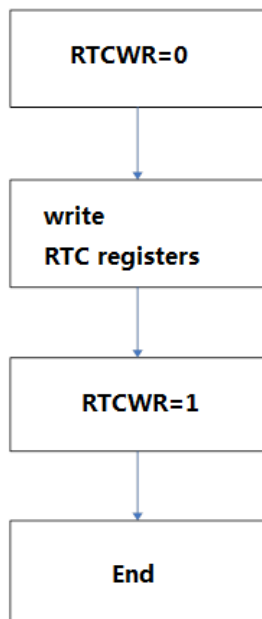
在用户需要读取 RTC 指示寄存器 (SECR, MINR, HRR, DAYR, MTHR, YRR, DOWR) 的时候，用户应该按照以下流程操作：



注：辅助 RTC 读流程与上图一致，只是读控制寄存器为 RTCRD2

### 15.7.2 写入 RTC 指示寄存器流程

在用户需要更新 RTC 指示寄存器 (SECR, MINR, HRR, DAYR, MTHR, YRR, DOWR) 的时候，用户应该按照以下流程操作，并且按照年、月、日、时、分、秒、周的顺序操作：



RTC 指示寄存器写流程

注：辅助 RTC 写流程与上图一致，只是写控制寄存器为 RTCWR2

## 15.8 校时记录

当对年(YRR)、月(MTHR)、日(DAYR)、时(HRR)、分(MINR)、秒(SECR)、周(DOWR)这七个寄存器中任何一个寄存器进行写操作时，则校时次数寄存器(RTCCNT)加1。

校时次数增加以一次 UpdatFlag 置 1 为准，即使一次更新多个寄存器，但是只有一次 UpdatFlag 置 1，则 RTCCNT 加 1。

如果用户写入的值超过正常范围导致写失败(比如月寄存器写 13)，那么只要 UpdatFlag 置 1，则 RTCCNT 加 1。

## 15.9 辅助 RTC

系统内建 1 个辅助 RTC 模块，提供第二套年/月/日/时/分/秒/周寄存器，由 LRC 驱动。外部 OSC 晶振停振时，系统可切换至内部低频 RC 获得计时时间。辅助 RTC 读写流程与第一套类似，只是其有单独的 RTCWR2 和 RTCRD2。

## 15.10 特殊功能寄存器列表

RTC 模块寄存器基地址：0x4000C000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	RTCCON	R/W	0x0000	RTC 控制寄存器



0x04	RTCIE	R/W	0x0000	RTC 中断使能寄存器
0x08	RTCIF	R/W	0x0000	RTC 中断标志寄存器
0x0C	ALMR	R/W	0x0000	闹钟寄存器
0x10	RTCTMR1	R/W	0x0000	RTC 定时器 1 计数设置
0x14	RTCTMR2	R/W	0x0000	RTC 定时器 2 计数设置
0x18	SECR	R/W	0x0000	秒寄存器（写保护）
0x1C	MINR	R/W	0x0000	分寄存器（写保护）
0x20	HOURR	R/W	0x0000	时寄存器（写保护）
0x24	DAYR	R/W	<b>0x0001</b>	日寄存器（写保护）
0x28	MONTHR	R/W	<b>0x0001</b>	月寄存器（写保护）
0x2C	YEARR	R/W	<b>0x0000</b>	年寄存器（写保护）
0x30	WEEKR	R/W	<b>0x0001</b>	周寄存器（写保护）
0x34	RTCCNTH	R/W	0x0000	校时次数寄存器高 16 位
0x38	RTCCNTL	R/W	0x0000	校时次数寄存器低 16 位
0x3C	RTCRd	R/W	0x0000	RTC 读控制寄存器
0x40	RTCWr	R/W	0x0000	RTC 写控制寄存器
0x50	DFAH	R/W	0x0000	RTC 硬件补偿常数项系数高位
0x54	DFAL	R/W	0x0000	RTC 硬件补偿常数项系数低位
0x58	DFBH	R/W	0x0000	RTC 硬件补偿一次项系数高位
0x5C	DFBL	R/W	0x0000	RTC 硬件补偿一次项系数低位
0x60	DFCH	R/W	0x0000	RTC 硬件补偿二次项系数高位
0x64	DFCL	R/W	0x0000	RTC 硬件补偿二次项系数低位
0x68	DFDH	R/W	0x0000	RTC 硬件补偿三次项系数高位
0x6C	DFDL	R/W	0x0000	RTC 硬件补偿三次项系数低位
0x70	DFEH	R/W	0x0000	RTC 硬件补偿四次项系数高位
0x74	DFEL	R/W	0x0000	RTC 硬件补偿四次项系数低位
0x78	Toff	R/W	0x0000	温度偏置寄存器
0x7C	MCON01	R/W	<b>0x0000</b>	Mems 控制寄存器（保持默认值）
0x80	MCON23	R/W	<b>0x0000</b>	Mems 控制寄存器（保持默认值）
0x84	MCON45	R/W	<b>0x8600</b>	Mems 控制寄存器（保持默认值）
0x88	DFiH	R/W	0x0000	频率误差寄存器高 5 位
0x8C	DFiL	R/W	0x0000	频率误差寄存器低 16 位
0x100	CTRLByFlash	R/*W	-----	
0x104	LRCCOMAND	W	0	LRC 写关闭命令寄存器

### RTC 模块寄存器基地址：0x4000C000

偏移地址	名称	读写方式	复位值	功能描述
0x200	SECR2	R/W	0x0000	辅助秒寄存器，写保护
0x204	MINR2	R/W	0x0000	辅助分钟寄存器，写保护
0x208	HOURR2	R/W	0x0000	辅助小时寄存器，写保护
0x20C	DAYR2	R/W	0x0001	辅助天寄存器，写保护

0x210	MONTHR2	R/W	0x0001	辅助月寄存器，写保护
0x214	YEARR2	R/W	0x0000	辅助年寄存器，写保护
0x218	WEEKR2	R/W	0x0001	辅助星期寄存器，写保护
0x21C	RTC2CAL	R/W	0x7FFF	辅助 RTC2 校正寄存器
0x220	RTCRD2	R/W	0x0000	辅助 RTC 读控制寄存器
0x224	RTCWR2	R/W	0x0000	辅助 RTC 写控制寄存器
0x228	FRE_LRC	R/W	0x0000	辅助 LRC 的频率测量寄存器

## 15.11 特殊功能寄存器说明

自动装载的 RTC 寄存器与 InfoFlash 的地址对应关系		
偏移地址	名称	功能描述
0x50	DFAH	0x104
0x54	DFAL	0x108
0x58	DFBH	0x10C
0x5C	DFBL	0x110
0x60	DFCH	0x114
0x64	DFCL	0x118
0x68	DFDH	0x11C
0x6C	DFDL	0x120
0x70	DFEH	0x124
0x74	DFEL	0x128
0x78	Toff	0x12C
0x7C	MCON01	0x130
0x80	MCON23	0x134
0x84	MCON45	0x138

注：自动装载功能为数字电路在复位完成之前自动完成的

注：自动装载以 32 位形式加载。

## 15.12 特殊功能寄存器说明

RTCCON (RTC 控制寄存器)		基地址: 0x4000C000 偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	RTC2EN	RTC1EN	X	TOUT[2:0]			AutoC

Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RTC2EN	<b>RTC 定时器 2 使能位</b> RTC2EN=0: RTC 定时器 2 被关闭 RTC2EN=1: RTC 定时器 2 被使能, 溢出产生 RTC2IF 标志。
RTC1EN	<b>RTC 定时器 1 使能位</b> RTC1EN=0: RTC 定时器 1 被关闭 RTC1EN=1: RTC 定时器 1 被使能, 溢出产生 RTC1IF 标志。
TOUT[3:0]	<b>TOUT输出频率说明见下表</b>
AutoC	<b>自动补偿控制位</b> 0: 使能自动补偿, 此时用户对 DFiH/ DFiL 写无效。 1: 手动补偿, 补偿值 (频率差值) 由用户写入 DFiH/ DFiL 寄存器。

TOUT[2:0]			TOUT	TOUT (PLL 使能)
0	0	0	0	0
0	0	1	1	1
0	1	0	32768Hz	32768Hz
0	1	1	内部 RTC 1Hz	由高频补偿 128Hz 后得到的 1Hz
1	0	0	内部 RTC 2Hz	由高频补偿 128Hz 后得到的 2Hz
1	0	1	内部 RTC 4Hz	由高频补偿 128Hz 后得到的 4Hz
1	1	0	内部 RTC 8Hz	由高频补偿 128Hz 后得到的 8Hz
1	1	1	内部 RTC 1Hz	内部 RTC 1Hz

RTCIE (RTC 中断使能寄存器)			基地址: 0x4000C000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ALMIE	RTC2IE	RTC1IE	MTHIE	DAYIE	HRIE	MINIE	SECIE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ALMIE	<b>闹钟中断使能位</b> 0: 关闭 1: 打开

RTC2IE	RTC 定时器 2 中断使能位 0: 关闭 1: 打开
RTC1IE	RTC 定时器 1 中断使能位 0: 关闭 1: 打开
MTHIE	RTC 月中断使能位 0: 关闭 1: 打开
DAYIE	RTC 日中断使能位 0: 关闭 1: 打开
HRIE	RTC 小时中断使能位 0: 关闭 1: 打开
MINIE	RTC 分钟中断使能位 0: 关闭 1: 打开
SECIE	RTC 秒中断使能位 0: 关闭 1: 打开

RTCIF (RTC 中断标志寄存器)			基地址: 0x4000C000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ALMIF	RTC2IF	RTC1IF	MTHIF	DAYIF	HRIF	MINIF	SECIF
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ALMIF	闹钟中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
RTC2IF	RTC 定时器 2 中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
RTC1IF	RTC 定时器 1 中断标志位 0: 未产生中断

	1: 产生中断, 写 0 清 0
MTHIF	RTC 月中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
DAYIF	RTC 日中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
HRIF	RTC 小时中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
MINIF	RTC 分钟中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
SECIF	RTC 秒中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0

ALMR (闹钟寄存器)		基地址: 0x4000C000 偏移地址: 0CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	ALMH[5:0]				
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	ALMM[5:0]					
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ALMH[4:0]	闹钟中断小时设置, 允许写入 0-23 以外的数, 但是闹钟中断将永远不会产生。
ALMM[5:0]	闹钟中断分钟设置, 允许写入 0-59 以外的数, 但是闹钟中断将永远不会产生。 说明: 当小时和分钟寄存器与闹钟寄存器中的值相匹配时 (且秒寄存器为 0), 才会产生中断。

RTCTMR1 (RTC 定时器 1 寄存器)		基地址: 0x4000C000 偏移地址: 10H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	CNT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0

Read:	CNT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CNT[15:0]	<p>最小分格为 1s，即最小可以每 1 秒产生一次中断，最大可以每 65536 秒产生一次中断，当计数溢出时，置位 RTC1IF 标志。</p> <p>CNT[15: 0]用来表示一个 16BIT 的二进制的无符号整数，如果设置 CNT[15: 0]=00H，表示 RTC 内部的秒表功能中断每经过 (00H+1)*1S =1*1S=1S 的计时周期后，置位 RTC1IF 标志。</p> <p>说明：当定时器溢出时，如果用户没有关闭定时器，则定时器将从 0 开始重新计数。</p>

<b>RTCTMR2</b> (RTC 定时器 2 寄存器)			<b>基地址: 0x4000C000</b> <b>偏移地址: 14H</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	CNT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CNT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CNT[15:0]	<p>最小分格为 0.0625s，即最小可以每 0.0625s 秒产生一次中断，最大可以每 4096 秒产生一次中断，当计数溢出时，置位 RTC2IF 标志。</p> <p>CNT[15:0]用来表示一个 16BIT 的二进制的无符号整数，如果设置 CNT[15:0] = 00H，表示 RTC 内部的秒表功能中断每经过 (00H+1)*0.0625S =1*0.0625S=0.0625S 的计时周期后，置位 RTC2IF 标志。</p> <p>说明：当定时器溢出时，如果用户没有关闭定时器，则定时器将从 0 开始重新计数。</p>

<b>SECR</b> (RTC 秒寄存器)			<b>基地址: 0x4000C000</b> <b>偏移地址: 18H</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	SEC[5:0]					
Write:								

Reset:	0	0	0	0	0	0	0	0
--------	---	---	---	---	---	---	---	---

位	功能描述
SEC[5:0]	<b>秒计数器:</b> 可设范围: 0-59。写入 0-59 以外的任何数值, 对该寄存器没有影响。

MINR (RTC 分寄存器)			基地址: 0x4000C000 偏移地址: 1CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	MIN[5:0]					
Write:	X	X	MIN[5:0]					
Reset:	0	0	0	0	0	0	0	0

位	功能描述
MIN[5:0]	<b>分计数器:</b> 可设范围: 0-59。写入 0-59 以外的任何数值, 对该寄存器没有影响。

HOURL (RTC 时寄存器)			基地址: 0x4000C000 偏移地址: 20H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	HOUR[4:0]				
Write:	X	X	X	HOUR[4:0]				
Reset:	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	<b>时计数器:</b> 可设范围: 0-23。写入 0-23 以外的任何数值, 对该寄存器没有影响。

DAYR (RTC 日寄存器)			基地址: 0x4000C000 偏移地址: 24H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	DAY[4:0]				
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAY[4:0]	<b>日计数器:</b> 可设范围: 1-28/29/30/31。写入与年、月不匹配的任何数值, 对该寄存器没有影响。

MONTHR (RTC 月寄存器)			基地址: 0x4000C000 偏移地址: 28H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	MONTH[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
MONTH[3:0]	<b>月计数器:</b> 可设范围: 1-12。写入 1-12 以外的任何数值, 对该寄存器没有影响

YEARR (RTC 年寄存器)			基地址: 0x4000C000 偏移地址: 2CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	YEAR[6:0]						
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
---	------



YEAR[6:0]	<b>年计数器:</b> 可设范围: 0-99。写入 0-99 以外的任何数值, 对该寄存器没有影响。
-----------	--

WEEKR (RTC 周寄存器)			<b>基地址: 0x4000C000</b> <b>偏移地址: 30H</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	WEEK[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
WEEK[2:0]	<b>周计数器:</b> 可设范围: 1-7。写入 1-7 以外的任何数值, 对该寄存器没有影响。

RTCCNTH (RTC 校时次数寄存器高 16 位)			<b>基地址: 0x4000C000</b> <b>偏移地址: 34H</b>					
	Bit15...Bit0							
Read:	RTCCNTH[15:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

RTCCNTL (RTC 校时次数寄存器低 16 位)			<b>基地址: 0x4000C000</b> <b>偏移地址: 38H</b>					
	Bit15...Bit0							
Read:	RTCCNTL[15:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RTCCNTH[15:0] RTCCNTL[15:0]	RTCCNTH 和 RTCCNTL 构成 32 位寄存器, 用于记录校时次数。该寄存器只读。

RTC RD (RTC 读控制寄存器)			<b>基地址: 0x4000C000</b> <b>偏移地址: 3CH</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								

Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	READFLAG
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
READFLAG	RTC 指示寄存器读控制位，具体使用参见 16.7.1 RTC 指示寄存器读写流程

RTCWR (RTC 写控制寄存器)		基地址: 0x4000C000 偏移地址: 40H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	UPDATE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
UPDATE	RTC 指示寄存器写控制位，具体使用参见 16.7.2 RTC 指示寄存器读写流程

DFAH (常数项高 7 位)		基地址: 0x4000C000 偏移地址: 50H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	DFx[22:16]						
Write:								
Reset:	0	0	0	0	0	0	0	0

DFAL (常数项低 16 位)		基地址: 0x4000C000 偏移地址: 54H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	DFx[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0

	Bit7	6	5	4	3	2	1	Bit0
Read:	DFx[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DFx[22:0]	DFAH/DFAL 构成 23 位有符号数, 为频率误差计算的常数项

<b>DFBH</b> (常数项高 7 位)			<b>基地址: 0x4000C000</b> <b>偏移地址: 58H</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	DFx[22:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DFx[22:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0

<b>DFBL</b> (常数项低 16 位)			<b>基地址: 0x4000C000</b> <b>偏移地址: 5CH</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	DFx[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DFx[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DFx[22:0]	DFAH/DFAL 构成 23 位有符号数, 为频率误差计算的一次项

<b>DFCH</b> (常数项高 7 位)			<b>基地址: 0x4000C000</b> <b>偏移地址: 60H</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	DFx[22:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DFx[22:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0

Write:								
Reset:	0	0	0	0	0	0	0	0

DFCL (常数项低 16 位)			基地址: 0x4000C000 偏移地址: 64H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	DFx[15:8]							
Write:	DFx[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DFx[7:0]							
Write:	DFx[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DFx[22:0]	DFAH/DFAL 构成 23 位有符号数, 为频率误差计算的二次项

DFDH (常数项高 7 位)			基地址: 0x4000C000 偏移地址: 68H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	DFx[22:16]						
Write:	X	DFx[22:16]						
Reset:	0	0	0	0	0	0	0	0

DFDL (常数项低 16 位)			基地址: 0x4000C000 偏移地址: 6CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	DFx[15:8]							
Write:	DFx[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DFx[7:0]							
Write:	DFx[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DFx[22:0]	DFDH/DFDL 构成 23 位有符号数, 为频率误差计算的三次项

DFEH (常数项高 7 位)			基地址: 0x4000C000 偏移地址: 70H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	DFx[22:16]						
Write:								
Reset:	0	0	0	0	0	0	0	0

DFEL (常数项低 16 位)			基地址: 0x4000C000 偏移地址: 74H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	DFx[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DFx[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DFx[22:0]	DFEH/DFEL 构成 23 位有符号数, 为频率误差计算的四次项

Toff (温度偏置寄存器)			基地址: 0x4000C000 偏移地址: 78H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	DFx[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DFx[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DFx[15:0]	16 位有符号数

DFiH (常数项高 5 位)			基地址: 0x4000C000 偏移地址: 88H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	DFx[20:16]				
Write:								
Reset:	0	0	0	0	0	0	0	0

DFiL (常数项低 16 位)			基地址: 0x4000C000 偏移地址: 8CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	DFx[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DFx[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DFx[22:0]	21 位寄存器，最高位为符号位。 当 AutoC 为 0 时，此时用户对 DFiH/ DFiL 写无效。 当 AutoC 为 1 时，手动补偿方式，补偿值由用户写入 DFiH/ DFiL 寄存器。用户必须先写高位，再写低位(即按照 DFiH—DFiL 顺序写寄存器)。

CTRLByFlash			基地址: 0x4000C000 偏移地址: 100H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	FLASH[7:0]							
Write:								
Reset:	1	1	1	1	1	1	1	1
	Bit7	6	5	4	3	2	1	Bit0

Read:	x	x	x	x	Reseved	AUTOREL OAD	LRC_CTRL L	x
Write:								
Reset:	1	0	1	0	1	1	1	0

位	功能描述
FLASH[7:0]	<b>代码空间加密位</b> 如果 Flash[7:0]=0xFF, 则 Flash 不加密 其他: Flash 加密, 该寄存器位只能读, 不能写
AUTORELOAD	<b>自动装载使能位</b> =1, 自动装载功能使能 =0, 自动装载功能屏蔽
LRC_CTRL	<b>低频 RC 控制位</b> =1: 打开低频 RC =0: 关闭低频 RC <b>注: 在 LRC_CTRL 等于 0 情况下, 用户还应向 LRCCOMAND 寄存器先写入 0x5555 再写入 0xAAAA, 才能真正关闭 LRC</b>

LRCCOMMAND (LRC 写关闭命令寄存器)		基地址: 0x4000C000 偏移地址: 104H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Bit[15:0]	在 LRC_CTRL 等于 0 情况下, 向 LRCCOMAND 寄存器先写入 0x5555 再写入 0xAAAA, 才能真正关闭 LRC。

以下寄存器为辅助 RTC 时钟寄存器, 这套时钟是以 LRC 为时钟源头

SECR2 (辅助 RTC 秒寄存器)		基地址: 0x4000C000 偏移地址: 200H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0

	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	SEC[5:0]					
Write:	X	X	SEC[5:0]					
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SEC[5:0]	<b>辅助秒计数器（以 LRC 为时钟基准）：</b> 可设范围：0-59。写入 0-59 以外的任何数值，对该寄存器没有影响。

<b>MINR2</b> (辅助 RTC 分寄存器)			<b>基地址： 0x4000C000</b> <b>偏移地址： 204H</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	MIN[5:0]					
Write:	X	X	MIN[5:0]					
Reset:	0	0	0	0	0	0	0	0

位	功能描述
MIN[5:0]	<b>辅助分计数器：</b> 可设范围：0-59。写入 0-59 以外的任何数值，对该寄存器没有影响。

<b>HRR2</b> (辅助 RTC 时寄存器)			<b>基地址： 0x4000C000</b> <b>偏移地址： 208H</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	HOUR[4:0]				
Write:	X	X	X	HOUR[4:0]				
Reset:	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	<b>辅助时计数器（以 LRC 为时钟基准）：</b> 可设范围：0-23。写入 0-23 以外的任何数值，对该寄存器没有影响。



DAYR2 (辅助 RTC 日寄存器)			<b>基地址:</b> 0x4000C000 <b>偏移地址:</b> 20CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	DAY[4:0]				
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAY[4:0]	<b>辅助日计数器 (以 LRC 为时钟基准):</b> 可设范围: 1-28/29/30/31。写入与年、月不匹配的任何数值, 对该寄存器没有影响。

MOUTH2R2 (辅助 RTC 月寄存器)			<b>基地址:</b> 0x4000C000 <b>偏移地址:</b> 210H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	MONTH[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
MONTH[3:0]	<b>辅助月计数器 (以 LRC 为时钟基准):</b> 可设范围: 1-12。写入 1-12 以外的任何数值, 对该寄存器没有影响

YEARR2 (辅助 RTC 年寄存器)			<b>基地址:</b> 0x4000C000 <b>偏移地址:</b> 214H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	YEAR[6:0]						
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	<b>辅助年计数器（以 LRC 为时钟基准）：</b> 可设范围：0-99。写入 0-99 以外的任何数值，对该寄存器没有影响。

WEEKR2 (辅助 RTC 周寄存器)			基地址: 0x4000C000 偏移地址: 218H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	WEEK[2:0]		
Write:	X	X	X	X	X	WEEK[2:0]		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
WEEK[2:0]	<b>辅助周计数器（以 LRC 为时钟基准）：</b> 可设范围：1-7。写入 1-7 以外的任何数值，对该寄存器没有影响。

RTC2CAL (辅助 RTC 校正寄存器)			基地址: 0x4000C000 偏移地址: 21CH					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	X	X	X	X	CAL16
Write:	X	X	X	X	X	X	X	CAL16
Reset:	0	1	1	1	1	1	1	1
	Bit15	14	13	12	11	10	9	Bit8
Read:	CAL15	CAL14	CAL13	CAL12	CAL11	CAL10	CAL9	CAL8
Write:	CAL15	CAL14	CAL13	CAL12	CAL11	CAL10	CAL9	CAL8
Reset:	1	1	1	1	1	1	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
Write:	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
Reset:	1	1	1	1	1	1	1	1

位	功能描述
CAL[16...0]	辅助 RTC 的校正寄存器，用于配置秒脉冲的时间基准，由于 LRC 时钟的初值不准确，通过此寄存器来配置决定计数多少个 LRC 的时钟为 1s 例：

如果该寄存器用户配置为 2000,则代表计数 2000 个 LRC 的时钟就产生一个秒脉冲

RTC RD2 (辅助 RTC 读控制寄存器)			基地址: 0x4000C000 偏移地址: 220H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	READFLAG
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
READFLAG	RTC 指示寄存器读控制位, 具体使用参见 RTC 指示寄存器读写流程

RTC WR2 (辅助 RTC 写控制寄存器)			基地址: 0x4000C000 偏移地址: 224H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	UPDATE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
UPDATE	RTC 指示寄存器写控制位, 具体使用参见 RTC 指示寄存器读写流程

FRE_LRC (RTC2 校正寄存器自动测试)			基地址: 0x4000C000 偏移地址: 228H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	X	X	X	X	LRC16
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8

<b>Read:</b>	LRC15	LRC14	LRC13	LRC12	LRC11	LRC10	LRC9	LRC8
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	LRC7	LRC6	LRC5	LRC4	LRC3	LRC2	LRC1	LRC0
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
LRC[16...0]	<p><b>LRC 频率寄存器:</b></p> <p>该寄存器主要为配合 RTC2CAL 使用, 使用 CLKCTRL1.14 使能测 LRC 频率功能以后, 芯片会以 LRC 时钟作为基础时钟来测量芯片的第一套 RTC 秒脉冲, 测得值会直接写入 FRE_LRC 寄存器, 也就是 LRC 的时钟频率, 推荐用户直接将该值写入 RTC2CAL, 通过这种方式, 可以使辅助 RTC 的寄存器和第一套的 RTC 寄存器尽可能的同步。</p>

## 16 TBS 模块

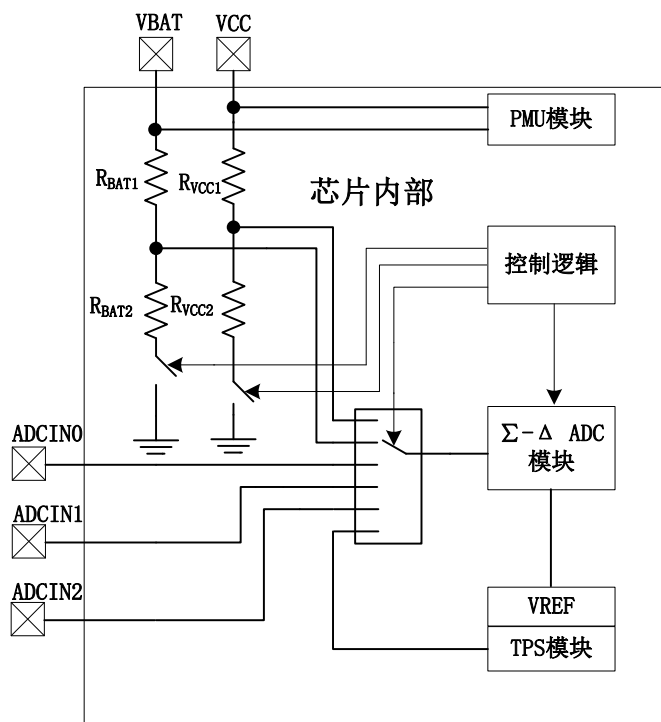
### 16.1 概述

芯片可以对温度和 5 路 ADC (V<sub>SYS</sub>, V<sub>BAT</sub>, ADCIN0, ADCIN1, ADCIN2) 进行定量的测量, 并将测量的结果保存在相应的寄存器中。

### 16.2 功能描述

TBS 主要包括两部分功能:

- 测量 IC 基底的温度
- 测试 5 路 ADC 电压 (V<sub>SYS</sub>, V<sub>BAT</sub>, ADCIN0, ADCIN1, ADCIN2)



V<sub>BAT</sub> 检测内阻为 30K (R<sub>BAT1</sub> 为 24K, R<sub>BAT2</sub> 为 6K), V<sub>CC</sub> 检测内阻为 42K (R<sub>VCC1</sub> 为 36K, R<sub>VCC2</sub> 为 6K)。ADCIN0, ADCIN1, ADCIN2 内部均无内阻。

## 16.3 特殊功能寄存器列表

TBS 模块寄存器基地址: 0x4000E000				
偏移地址	名称	读写方式	复位值	功能描述
00H	TBSCON	R/W	0x0101	TBS设置寄存器
04H	TBSPRD	R/W	0x0000	TBS打开周期设置寄存器
08H	TBSIE	R/W	0x0000	TBS中断使能寄存器
0CH	TBSIF	R/W	0x0000	TBS中断标志寄存器
10H	TMPDAT	R/W	0x0000	温度测量输出值
14H	VBATDAT	R/W	0x0000	电池电压测量输出值
18H	VCCDAT	R/W	0x0000	电源电压测量输出值
1CH	ADCODAT	R/W	0x0000	ADC通道0测量输出值
20H	ADC1DAT	R/W	0x0000	ADC通道1测量输出值
24H	ADC2DAT	R/W	0x0000	ADC通道2测量输出值
28H	VBATCMP	R/W	0x0000	电池电压比较寄存器
34H	ADCOCMP	R/W	0x0000	ADC0比较值, 16位有符号数,
38H	ADC1CMP	R/W	0x0000	ADC1比较值, 16位有符号数

## 16.4 特殊功能寄存器说明

TBSCON (TBS 设置寄存器)			基地址: 0x4000E000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	x	x	x	ADC1CMP	ADCOCMP	x	Filter1	Filter0
Write:				En	En			
Reset:	0	0	0	0	0	0	0	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	x	VbatCMP	ADC2En	ADC1En	ADC0En	VccEn	VbatEn	TPSEn
Write:		En						
Reset:	0	0	0	0	0	0	0	1

位	功能描述
ADC1CMPEn	此位只在 ADC1En 使能的情况下才有效 ADC1CMPEn =0: 关闭 ADC1DAT 与 ADC1CMP 两个寄存器的比较 ADC1CMPEn =1: 使能 ADC1DAT 与 ADC1CMP 两个寄存器的比较
ADCOCMPEn	此位只在 ADC0En 使能的情况下才有效 ADCOCMPEn =0: 关闭 ADCODAT 与 ADCOCMP 两个寄存器的比较 ADCOCMPEn =1: 使能 ADCODAT 与 ADCOCMP 两个寄存器的比较
Filter[1..0]	只对温度测量:

	Filter [1..0] =00b: 对 ADC 输出的值经数字直接输出 Filter [1..0] =01b: 对 ADC 输出的 2 次数值做平均 Filter [1..0] =10b: 对 ADC 输出的 4 次数值做平均 Filter [1..0] =11b: 对 ADC 输出的 8 次数值做平均
VbatCMPEn	此位只在 VbatEn 使能的情况下才有效 VbatCMPEn=0: 关闭 VBATDAT 与 VBATCMP 两个寄存器的比较 VbatCMPEn=1: 使能 VBATDAT 与 VBATMP 两个寄存器的比较
ADC2En	ADC2En =0: ADC 通道 2 关闭 ADC2En =1: ADC 通道 2 使能 注: 如使用该功能, 用户需在 GPIO 模块将对应 Pin 配置为 ADC 输入 Pin 才行
ADC1En	ADC1En =0: ADC 通道 1 关闭 ADC1En =1: ADC 通道 1 使能 注: 如使用该功能, 用户需在 GPIO 模块将对应 Pin 配置为 ADC 输入 Pin 才行
ADC0En	ADC0En =0: ADC 通道 0 关闭 ADC0En =1: ADC 通道 0 使能 注: 如使用该功能, 用户需在 GPIO 模块将对应 Pin 配置为 ADC 输入 Pin 才行
VccEn	VccEn =0: Vcc 测量关闭 VccEn =1: Vcc 测量使能
VbatEn	VbatEn =0: 电池电压测量关闭 VbatEn =1: 电池电压测量使能
TPSEn	TDCEn =0: 温度测量关闭 TDCEn =1: 温度测量使能

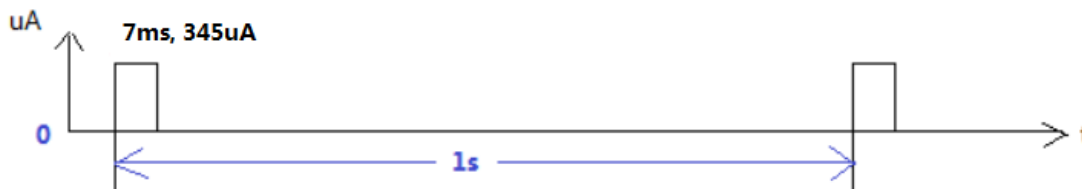
TBSPRD (TBS 打开周期设置寄存器)		<b>基地址: 0x4000E000</b> <b>偏移地址: 04H</b>						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	ADC2PRD	ADC2PRD	ADC1PRD	ADC1PRD
Write:					1	0	1	0
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADCOPRD	ADCOPRD	VccPRD1	VccPRD0	VbatPRD	VbatPRD	TPSPRD1	TPSPRD0
Write:	1	0			1	0		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADC2PRD[1..0]	ADC2PRD[1..0]=00b ADC 通道 2 1s 打开一次 ADC2PRD[1..0]=01b ADC 通道 2 4s 打开一次 ADC2PRD[1..0]=10b ADC 通道 2 8s 打开一次 ADC2PRD[1..0]=11b ADC 通道 2 32s 秒打开一次
ADC1PRD[1..0]	ADC1PRD[1..0]=00b ADC 通道 1 1s 打开一次 ADC1PRD[1..0]=01b ADC 通道 1 4s 打开一次 ADC1PRD[1..0]=10b ADC 通道 1 8s 打开一次 ADC1PRD[1..0]=11b ADC 通道 1 32s 打开一次

ADCOPRD[1..0]	ADCOPRD[1..0]=00b ADC 通道 0 1s 打开一次 ADCOPRD[1..0]=01b ADC 通道 0 4s 打开一次 ADCOPRD[1..0]=10b ADC 通道 0 8s 打开一次 ADCOPRD[1..0]=11b ADC 通道 0 32s 打开一次
VccPRD[1..0]	VccPRD [1..0]=00b Vcc 检测 1s 打开一次 VccPRD [1..0]=01b Vcc 检测 4s 打开一次 VccPRD [1..0]=10b Vcc 检测 8s 打开一次 VccPRD [1..0]=11b Vcc 检测 32s 打开一次
VbatPRD[1..0]	VbatPRD [1..0]=00b Vbat 检测 1s 打开一次 VbatPRD [1..0]=01b Vbat 检测 4s 打开一次 VbatPRD [1..0]=10b Vbat 检测 8s 打开一次 VbatPRD [1..0]=11b Vbat 检测 32s 打开一次
TPSPRD[1..0]	TPSPRD [1..0]=00b TPS 检测 1s 打开一次 TPSPRD [1..0]=01b TPS 检测 8s 打开一次 TPSPRD [1..0]=10b TPS 检测 32s 打开一次 TPSPRD [1..0]=11b TPS 检测 1/8s 打开一次 (打开频率 8Hz, 即 1s 打开 8 次)

注：推荐正常模式下，**TMPPRD** 打开频率都为最快。低功耗模式下，建议 8s 及以上打开一次。  
**TPS** 分时开启：每次开启的峰值功耗为 345uA，持续时间为 7ms，各 ADC 通道采集和温度通道一样。  
 例如 1s 分时开启示意图如下

TPS 分时1s开启一次的 功耗&开启时间 示意图



TBSIE (TBS 中断使能寄存器)		基地址: 0x4000E000 偏移地址: 08H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	ADC1CMP IE	ADCOCMP IE
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	VbatCM PIE	ADC2IE	ADC1IE	ADCOIE	VccIE	VbatIE	TPSIE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
---	------



ADC1CMPIE	ADC1CMPIE =0: ADC1 测量比较功能中断关闭 ADC1CMPIE =1: ADC1 测量比较功能中断使能
ADC0CMPIE	ADC0CMPIE =0: ADC0 测量比较功能中断关闭 ADC0CMPIE =1: ADC0 测量比较功能中断使能
VbatCMPIE	VbatCMPIE=0: 电池电压比较功能中断关闭 VbatCMPIE=1: 电池电压比较功能中断使能
ADC2IE	ADC2IE=0: ADC 通道 2 检测中断关闭 ADC2IE=1: ADC 通道 2 检测中断使能
ADC1IE	ADC1IE=0: ADC 通道 1 检测中断关闭 ADC1IE=1: ADC 通道 1 检测中断使能
ADC0IE	ADC0IE=0: ADC 通道 0 检测中断关闭 ADC0IE=1: ADC 通道 0 检测中断使能
VccIE	VccIE=0: Vcc 测量中断关闭 VccIE=1: Vcc 测量中断使能
VbatIE	VbatIE=0: 电池电压测量中断关闭 VbatIE=1: 电池电压测量中断使能
TPSIE	TPSIE =0: 温度测量中断关闭 TPSIE =1: 温度测量中断使能

TBSIF (TBS 中断使能寄存器)		基地址: 0x4000E000 偏移地址: 0CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	ADC1CMP	ADC0CMP
Write:							IF	IF
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	VbatCM	ADC2IF	ADC1IF	ADC0IF	VccIF	VbatIF	TPSIF
Write:		PIF						
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADC1CMPIF	ADC1CMPIF =0: 未发生 ADC1 测量比较功能中断 ADC1CMPIF =1: 发生 ADC1 测量比较功能中断 写 0 清 0
ADC0CMPIF	ADC0CMPIF =0: 未发生 ADC0 测量比较功能中断 ADC0CMPIF =1: 发生 ADC0 测量比较功能中断 写 0 清 0
VbatCMPIF	VbatCMPIF=0: 未发生电池电压比较功能中断 VbatCMPIF=1: 发生电池电压比较功能中断 写 0 清 0
ADC2IF	ADC2IF=0: 未发生 ADC 通道 2 检测中断 ADC2IF=1: 发生 ADC 通道 2 检测中断

	写 0 清 0
ADC1IF	ADC1IF=0: 未发生 ADC 通道 1 检测中断 ADC1IF=1: 发生 ADC 通道 1 检测中断 写 0 清 0
ADC0IF	ADC0IF=0: 未发生 ADC 通道 0 检测中断 ADC0IF=1: 发生 ADC 通道 0 检测中断 写 0 清 0
VccIF	VccIF=0: 未发生 Vcc 测量中断 VccIF=1: 发生 Vcc 测量中断 写 0 清 0
VbatIF	VbatIF=0: 未发生电池电压测量中断 VbatIF=1: 发生电池电压测量中断 写 0 清 0
TPSIF	TPSIF =0: 未发生温度测量中断 TPSIF =1: 发生温度测量中断 写 0 清 0

TMPDAT (温度测量输出寄存器)		<b>基地址: 0x4000E000</b> <b>偏移地址: 10H</b>						
	Bit15...Bit0							
Read:	DAT[15:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[15:0]	温度测量输出寄存器, 16 位有符号数

温度计算公式:

$$\text{温度 } Tr = 12.9852 - \text{TMPDAT} * 0.0028$$

其中: Tr为实际的温度 (°C)

VBATDAT (电池电压测量输出寄存器)		<b>基地址: 0x4000E000</b> <b>偏移地址: 14H</b>						
	Bit15...Bit0							
Read:	DAT[15:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[15:0]	电池电压测量输出寄存器, 16 位有符号数

电池电压计算公式:

$$Vbat = 0.06466 * \text{VBATDAT} + 2124;$$

其中：Vbat为实际电池电压（mV）

VCCDAT (电源电压测量输出寄存器)		基地址： 0x4000E000 偏移地址： 18H						
Bit15…Bit0								
Read:	DAT[15:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[15:0]	电源电压测量输出寄存器，16位有符号数

电源电压计算公式：

$$V_{cc} = 0.089 * VCCDAT + 2942$$

其中：Vcc为实际电源电压（mV）

ADC0DAT (ADC0 测量输出寄存器)		基地址： 0x4000E000 偏移地址： 1CH						
Bit15…Bit0								
Read:	DAT[15:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[15:0]	ADC 通道 0 测量输出寄存器，16位有符号数

ADC 通道 0 测量电压计算公式：

$$V_{adc0} = 0.01285 * ADC0DAT + 421;$$

其中：Vadc0为实际ADC测量电压（mV）

ADC1DAT (ADC1 测量输出寄存器)		基地址： 0x4000E000 偏移地址： 20H						
Bit15…Bit0								
Read:	DAT[15:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[15:0]	ADC 通道 1 测量输出寄存器，16位有符号数

ADC 通道 1 测量电压计算公式:

$$V_{adc1} = 0.01285 * ADC1DAT + 421;$$

其中:  $V_{adc1}$  为实际ADC测量电压 (mV)

ADC2DAT (ADC2 测量输出寄存器)		<b>基地址:</b> 0x4000E000 <b>偏移地址:</b> 24H						
		Bit15...Bit0						
<b>Read:</b>	DAT[15:0]							
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DAT[15:0]	ADC 通道 2 测量输出寄存器, 16 位有符号数

ADC 通道 2 测量电压计算公式:

$$V_{adc2} = 0.01285 * ADC1DAT + 421;$$

其中:  $V_{adc2}$  为实际ADC测量电压 (mV)

VBATCMP (电池电压比较寄存器)		<b>基地址:</b> 0x4000E000 <b>偏移地址:</b> 28H						
		Bit15...Bit0						
<b>Read:</b>	DAT[15:0]							
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DAT[15:0]	电池电压测量比较寄存器, 16 位有符号数

当使能了电池电压测量 (VBATEn) 功能并且使能电池电压比较中断 (VbatCMPIE) 后, VBATDAT 低于 VDRCMP 时将产生中断。

ADCOCMP (ADC0 测量比较寄存器)		<b>基地址:</b> 0x4000E000 <b>偏移地址:</b> 34H						
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	DFx15	DFx14	DFx13	DFx12	DFx11	DFx10	DFx9	DFx8
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	DFx7	DFx6	DFx5	DFx4	DFx3	DFx2	DFx1	DFx0
<b>Write:</b>								

Reset:	0	0	0	0	0	0	0	0
--------	---	---	---	---	---	---	---	---

位	功能描述
DFx[15..0]	ADC0 测量比较寄存器，16 位有符号数

当 ADC0 测量值大于 ADC0CMP 寄存器值时，产生中断。

ADC1CMP (ADC1 测量比较寄存器)		<b>基地址:</b> 0x4000E000 <b>偏移地址:</b> 38H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	DFx15	DFx14	DFx13	DFx12	DFx11	DFx10	DFx9	DFx8
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DFx7	DFx6	DFx5	DFx4	DFx3	DFx2	DFx1	DFx0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DFx[15..0]	ADC1 测量比较寄存器，16 位有符号数

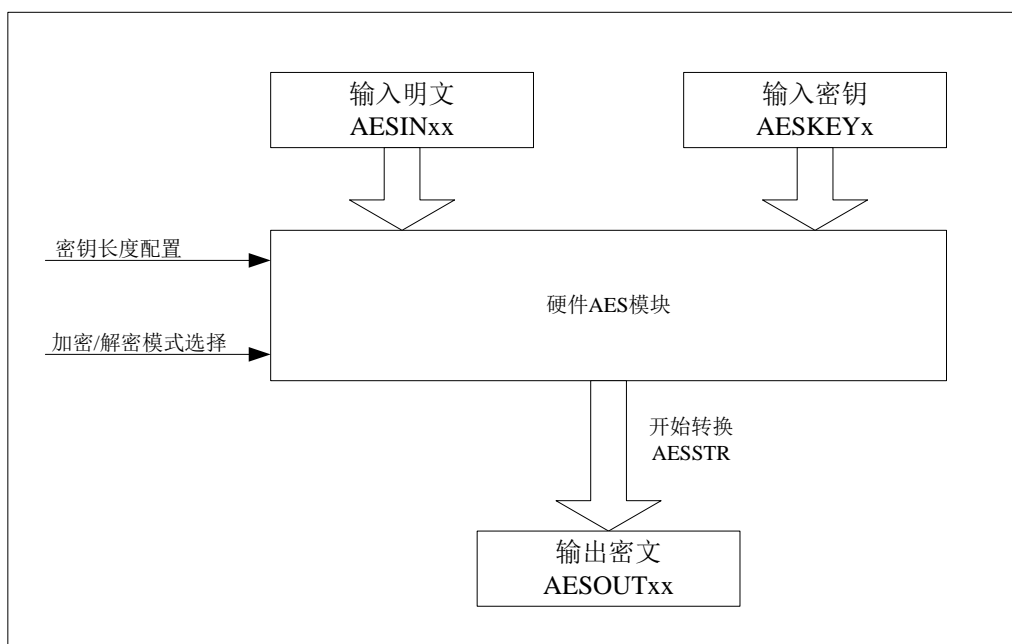
当 ADC1 测量值小于 ADC1CMP 寄存器值时，产生中断。

## 17 AES&Ghash 模块

### 17.1 AES 概述

HT500X 内部集成了 AES 硬件加密/解密模块，符合 FIPS197 标准，通过相应的软件配置，支持 EBC/CBC/CTR/GCM/GMAC 多种模式，AES 的密钥长度支持 128bit, 192bit, 256bit。通过该硬件模块和 Ghash 乘法器模块，大大的缩短了了用户软件实现 AES 算法以及不同的加密/解密模式的时间。

### 17.2 框图



### 17.3 特殊功能寄存器列表

AES 模块寄存器基地址: 0x40012000				
偏移地址	名称	读写方式	复位值	功能描述
00H	AESCON	R/W	0000H	3DES配置寄存器
04H	AESSTR	W	0000H	3DES启动命令寄存器
08H	AESFLG	R/W	0000H	3DES标志寄存器
0CH	AESINLL	R/W	00000000H	AES输入待加/解密数据
10H	AESINML	R/W	00000000H	AES输入待加/解密数据
14H	AESINHM	R/W	00000000H	AES输入待加/解密数据
18H	AESINHH	R/W	00000000H	AES输入待加/解密数据
1CH	AESOUTLL	R/W	00000000H	AES输出已加/解密数据
20H	AESOUTML	R/W	00000000H	AES输出已加/解密数据
24H	AESOUTHM	R/W	00000000H	AES输出已加/解密数据

28H	AESOUTH	R/W	00000000H	AES输出已加/解密数据
2CH	AESKEY0	R/W	00000000H	AES密钥寄存器
30H	AESKEY1	R/W	00000000H	AES密钥寄存器
34H	AESKEY2	R/W	00000000H	AES密钥寄存器
38H	AESKEY3	R/W	00000000H	AES密钥寄存器
3CH	AESKEY4	R/W	00000000H	AES密钥寄存器
40H	AESKEY5	R/W	00000000H	AES密钥寄存器
44H	AESKEY6	R/W	00000000H	AES密钥寄存器
48H	AESKEY7	R/W	00000000H	AES密钥寄存器

**GHASH 模块寄存器基地址: 0x40012100**

偏移地址	名称	读写方式	复位值	功能描述
00H	GHASHSTR	W	0000H	GHASH启动命令寄存器
04H	GHASHFLG	R/W	0000H	GHASH标志寄存器
08H	INPUT1LL	R/W	00000000H	GHASH输入数据1
0CH	INPUT1ML	R/W	00000000H	GHASH输入数据1
10H	INPUT1HM	R/W	00000000H	GHASH输入数据1
14H	INPUT1HH	R/W	00000000H	GHASH输入数据1
18H	INPUT2LL	R/W	00000000H	GHASH输入数据2
1CH	INPUT2ML	R/W	00000000H	GHASH输入数据2
20H	INPUT2HM	R/W	00000000H	GHASH输入数据2
24H	INPUT2HH	R/W	00000000H	GHASH输入数据2
28H	OUTPUTLL	R/W	00000000H	GHASH输出数据
2CH	OUTPUTML	R/W	00000000H	GHASH输出数据
30H	OUTPUTHM	R/W	00000000H	GHASH输出数据
34H	OUTPUTHH	R/W	00000000H	GHASH输出数据
38H	AESGHASHIE	R/W	00H	AES, Ghash中断使能位
3CH	AESGHASHIF	R/W	00H	AES, Ghash中断标志位

## 17.4 特殊功能寄存器说明

AESCON (AES 配置寄存器)			基地址: 0x40012000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	KEY_MOD E[1]	KEY_MOD E[0]	MODE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
KEY_MODE	<b>KEY_MODE[1... 0]:</b> =B00: AES-128, 密钥为 128bits(KEY0—KEY3)KEY0 为高位, KEY3 为低位 =B01: AES-192, 密钥为 192bits(KEY0—KEY5)KEY0 为高位, KEY5 为低位 =B10: AES-256, 密钥为 256bits(KEY0—KEY7)KEY0 为高位, KEY7 为低位 =B11: AES-256, 密钥为 256bits(KEY0—KEY7)KEY0 为高位, KEY7 为低位
MODE	<b>编码/解码模式选择控制:</b> 0: 加密 1: 解密

AESSTR (AES 启动寄存器)			基地址: 0x40012000 偏移地址: 04H					
	Bit15...Bit0							
Read:	X							
Write:	DAT[15:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[15:0]	寄存器写入 0x8329 则启动 AES 加密/解密 该控制寄存器只能写入, 不能读取。该寄存器读出永远为 0 寄存器写入 0x8581, 当 Key 不变的时候, 启动明文编码, 这种方式可以提升 AES 计算时间, 当用户配置的 KEY 不修改的时候, 不需要反复的计算 AES 运算中需要的 KEY 转换部分。

AESFLG (AES 标志寄存器)			基地址: 0x40012000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8



Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	BUSY
Write:								X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
BUSY	1: AES 加密/解密未完成 0: AES 加密/解密完成 该寄存器位只读 当加密/解密完成的时候, 会产生相应的中断标志, 如果中断使能打开, 会进入相应的中断。AES 对应中断向量号为 1。

AESINLL (AES 数据输入中低位)	<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 0CH
	Bit31...Bit0
Read:	DAT[31:0]
Write:	
Reset:	0 0 0 0 0 0 0 0 0

位	功能描述
DAT[31:0]	AES 输入待加密/解密数据 128bit 的 bit0-bit31

AESINML (AES 数据输入中低位)	<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 18H
	Bit31...Bit0
Read:	DAT[31:0]
Write:	
Reset:	0 0 0 0 0 0 0 0 0

位	功能描述
DAT[31:0]	AES 输入待加密/解密数据 128bit 的 bit32-bit63

AESINHM (AES 数据输入中高位)	<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 10H
	Bit31...Bit0
Read:	DAT[31:0]

Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 输入待加密/解密数据 128bit 的 bit64-bit95

AESINH (AES 数据输入高位)		<b>基地址: 0x40012000</b> <b>偏移地址: 14H</b>						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 输入待加密/解密数据 128bit 的 bit96-bit127

AESOUTLL (AES 数据输出低位)		<b>基地址: 0x40012000</b> <b>偏移地址: 1CH</b>						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 输出已加密/解密数据 128bit 的 bit0-bit31

AESOUTML (AES 数据输出中低位)		<b>基地址: 0x40012000</b> <b>偏移地址: 20H</b>						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 输出已加密/解密数据 128bit 的 bit32-bit63

AESOUTHM (AES 数据输出中高位)		<b>基地址: 0x40012000</b> <b>偏移地址: 24H</b>						
---------------------------	--	--	--	--	--	--	--	--

	Bit31…Bit0							
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 输出已加密/解密数据 128bit 的 bit64-bit95

AESOUTHH (AES 数据输出高位)	<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 28H							
	Bit31…Bit0							
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 输出已加密/解密数据 128bit 的 bit96-bit127

AESKEY0 (AES 密钥 0)	<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 2CH							
	Bit31…Bit0							
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 密钥 128bit 的 bit96-bit127, 该寄存器可以写入也可以读出 AES 密钥 192bit 的 bit160-bit191, 该寄存器可以写入也可以读出 AES 密钥 256bit 的 bit224-bit256, 该寄存器可以写入也可以读出

AESKEY1 (AES 密钥 1)	<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 30H							
	Bit31…Bit0							
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 密钥 128bit 的 bit64-bit95, 该寄存器可以写入也可以读出

	AES 密钥 192bit 的 bit128-bit159, 该寄存器可以写入也可以读出 AES 密钥 256bit 的 bit192-bit223, 该寄存器可以写入也可以读出
--	--

<b>AESKEY2</b> (AES 密钥 2)	<b>基地址:</b> 0x40012000							
	<b>偏移地址:</b> 34H							
	Bit31...Bit0							
<b>Read:</b>	DAT[31:0]							
<b>Write:</b>	DAT[31:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 密钥 128bit 的 bit32-bit63, 该寄存器可以写入也可以读出 AES 密钥 192bit 的 bit96-bit127, 该寄存器可以写入也可以读出 AES 密钥 256bit 的 bit160-bit191, 该寄存器可以写入也可以读出

<b>AESKEY3</b> (AES 密钥 3)	<b>基地址:</b> 0x40012000							
	<b>偏移地址:</b> 38H							
	Bit31...Bit0							
<b>Read:</b>	DAT[31:0]							
<b>Write:</b>	DAT[31:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 密钥 128bit 的 bit0-bit31, 该寄存器可以写入也可以读出 AES 密钥 192bit 的 bit64-bit95, 该寄存器可以写入也可以读出 AES 密钥 256bit 的 bit128-bit159, 该寄存器可以写入也可以读出

<b>AESKEY4</b> (AES 密钥 4)	<b>基地址:</b> 0x40012000							
	<b>偏移地址:</b> 3CH							
	Bit31...Bit0							
<b>Read:</b>	DAT[31:0]							
<b>Write:</b>	DAT[31:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 密钥 192bit 的 bit32-bit63, 该寄存器可以写入也可以读出 AES 密钥 256bit 的 bit96-bit127, 该寄存器可以写入也可以读出

AESKEY5 (AES 密钥 5)		<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 40H						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 密钥 192bit 的 bit0-bit31, 该寄存器可以写入也可以读出 AES 密钥 256bit 的 bit64-bit95, 该寄存器可以写入也可以读出

AESKEY6 (AES 密钥 6)		<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 44H						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 密钥 256bit 的 bit32-bit63, 该寄存器可以写入也可以读出

AESKEY7 (AES 密钥 7)		<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 48H						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	AES 密钥 256bit 的 bit0-bit31, 该寄存器可以写入也可以读出

GHASHSTR (GHASH 启动寄存器)		<b>基地址:</b> 0x40012100 <b>偏移地址:</b> 00H						
		Bit15...Bit0						
Read:	X							
Write:	DAT[15:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
---	------

DAT[15:0]	寄存器写入 0x0501 则启动 GHASH 乘法计算 该控制寄存器只能写入，不能读取。该寄存器读出永远为 0
-----------	--

GHASHFLG (GHASH 标志寄存器)			基地址: 0x40012100 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	BUSY
Write:								X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
BUSY	1: GHASH 乘法运算未完成 0: GHASH 乘法运算完成 该寄存器位为只读 当乘法运算完成的时候，会产生相应的中断标志，如果中断使能打开，会进入相应的中断。AESGHASH 对应中断向量号为 1。

INPUT1LL (数据 1 输入低位)			基地址: 0x40012100 偏移地址: 08H					
	Bit31...Bit0							
Read:	DAT[31:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	Ghash 输入计算数据 1 的 128bit 的 bit0-bit31

INPUT1ML (数据 1 输入中低位)			基地址: 0x40012100 偏移地址: 0CH					
	Bit31...Bit0							
Read:	DAT[31:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
---	------

DAT[31:0]	Ghash 输入计算数据 1 的 128bit 的 bit32-bit63
-----------	---------------------------------------

INPUT1MH (数据 1 输入中高位)		<b>基地址: 0x40012100</b> <b>偏移地址: 10H</b>						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	Ghash 输入计算数据 1 的 128bit 的 bit64-bit95

INPUT1HH (数据 1 输入高位)		<b>基地址: 0x40012100</b> <b>偏移地址: 14H</b>						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	Ghash 输入计算数据 1 的 128bit 的 bit96-bit127

INPUT2LL (数据 2 输入低位)		<b>基地址: 0x40012100</b> <b>偏移地址: 18H</b>						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	Ghash 输入计算数据 1 的 128bit 的 bit0-bit31

INPUT2ML (数据 2 输入中低位)		<b>基地址: 0x40012100</b> <b>偏移地址: 1CH</b>						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	Ghash 输入计算数据 1 的 128bit 的 bit32-bit63

INPUT2MH (数据 2 输入中高位)		<b>基地址:</b> 0x40012100 <b>偏移地址:</b> 20H						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	Ghash 输入计算数据 1 的 128bit 的 bit64-bit95

INPUT2HH (数据 2 输入高位)		<b>基地址:</b> 0x40012100 <b>偏移地址:</b> 24H						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	Ghash 输入计算数据 1 的 128bit 的 bit96-bit127

OUTPUTLL (数据输出低位)		<b>基地址:</b> 0x40012100 <b>偏移地址:</b> 28H						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	Ghash 输入计算数据 1 的 128bit 的 bit0-bit31

OUTPUTML (数据输出中低位)		<b>基地址:</b> 0x40012100 <b>偏移地址:</b> 2CH						
		Bit31...Bit0						
Read:	DAT[31:0]							



Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	Ghash 输入计算数据 1 的 128bit 的 bit32-bit63

OUTPUTMH (数据输出中高位)		基地址: 0x40012100 偏移地址: 30H						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	Ghash 输入计算数据 1 的 128bit 的 bit64-bit95

OUTPUTHH (数据输出高位)		基地址: 0x40012100 偏移地址: 34H						
		Bit31...Bit0						
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	Ghash 输入计算数据 1 的 128bit 的 bit96-bit127

AESGHASHIE (AESGHASH 中断使能)			基地址: 0x40012100 偏移地址: 38H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	GHASHIE	AESIE
Write:	X	X	X	X	X	X	GHASHIE	AESIE
Reset:	0	0	0	0	0	0	0	0

位	功能描述
GHASHIE	GHASH 计算中断使能

AESIE	AES 计算中断使能
-------	------------

AESGHASHIF (AESGHASH 中断标志)			<b>基地址:</b> 0x40012100 <b>偏移地址:</b> 3CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	GHASHIF	AESIF
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
GHASHIF	中断标志
AESIF	AES 计算中断标志

## 18 DMA 功能

### 18.1 概述

DMA 模块提供高速的数据传输在外设和 RAM 之间或者 RAM 和 RAM 之间, 在 DMA 数据搬运的过程中不需要 CPU 的参与, 但是占用系统总线。

### 18.2 功能描述

- (1) DMA 模块共有 3 个独立的可配置通道
- (2) DMA 的传输可以外设到外设, 外设到存储器, 存储器到外设, 存储器到存储器
- (3) DMA 与 CPU 之间交替占用总线
- (4) DMA 支持最多 32 个外部请求源, 每个通道还有一个软件请求源
- (5) 三个通道有优先级的设置, 一旦一个通道的一次传输开始后不能被打断。当三个通道同时请求 DMA 传输时, 优先级高的被响应。优先级低的通道只有在高优先级的通道传输完成后才能被响应
- (6) 当两个通道选择了同一个外部请求源时, 只有优先级高的通道响应此请求
- (7) DMA 支持两种传输方式: 一次请求传输一个数据、一次请求传输所有的数据
- (8) DMA 支持三种中断: 传输完成中断, 传输错误中断, 半传输中断
- (9) 在非循环模式下, DMA 完成数据传输后自动关闭相应通道, 当选择为循环模式的时候, 用户可配置循环次数 CHNxULKNUM[8:15], 达到循环次数后自动关闭相应通道

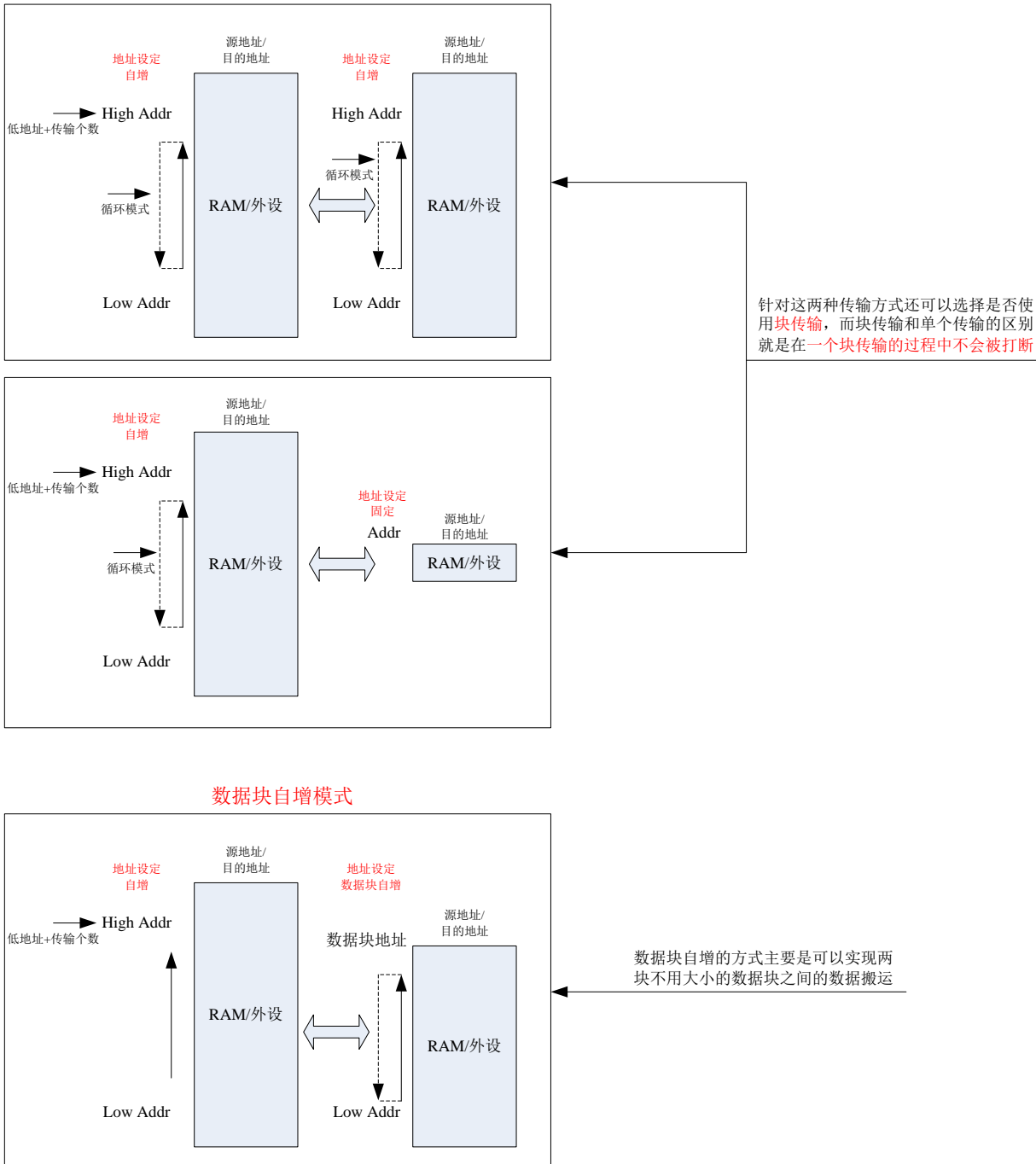
### 18.3 DMA 通道请求列表

DMA_CTL[8...12]	说明
0	Soft request
1	UART0 发送
2	UART0 接收
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	UART3 发送
8	UART3 接收
9	UART4 发送
10	UART4 接收
11	Reserved
12	Reserved

13	ISO7816_0 发送
14	ISO7816_0 接收
15	ISO7816_1 发送
16	ISO7816_1 接收
17	Reserved
18	Reserved
19	I2C 发送
20	I2C 接收
21	EMU 电压波形数据
22	EMU 电流通道 1 波形数据
23	EMU 电流通道 2 波形数据
24	Timer0
25	Timer1
26	Timer2
27	Timer3
28	Reserved
29	Reserved
30	Reserved
31	Reserved
32	Reserved

注：当选择 DMA 发送或接收串口数据（UART）时，UART 对应的 TXIF/RXIF 会置位。

## 18.4 DMA 数据传输说明



## 18.5 特殊功能寄存器列表

DMA 模块寄存器基地址: 0x40016000				
偏移地址	名称	读写方式	复位值	功能描述
00H	DMAIE	R/W	0000H	DMA中断使能寄存器
04H	DMAIF	R/W	0000H	DMA中断标志寄存器
08H	CHNSTA	R/W	0000H	DMA状态寄存器
DMA 通道配置寄存器基地址: 0x4001600C (Channel0) 0x40016024 (Channel1) 0x4001603C (Channel2)				
偏移地址	名称	读写方式	复位值	功能描述
00H	CHNCTL	R/W	0000H	通道控制寄存器
04H	CHNSRC	R/W	0000H	通道源地址寄存器
08H	CHNTAR	R/W	0000H	通道目的地址寄存器
0CH	CHNCNT	R/W	0000H	通道传输数量设置寄存器
10H	CHNTCCNT	R	0000H	通道传输完成数据个数
14H	CHNBULKNUM	R/W	0000H	通道块传输个数设置

## 18.6 特殊功能寄存器说明

DMAIE (DMA 中断使能寄存器)		基地址: 0x40016000 偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:						TEIE2	TEIE1	TEIE0
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:		BCIE2	BCIE1	BCIE0		TCIE2	TCIE1	TCIE0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TCIE[2...0]	通道 0/1/2 传输结束中断使能 0: 禁止 1: 使能
BCIE[2...0]	通道 0/1/2 块传输中断使能 0: 禁止

	1: 使能
TEIE[2...0]	通道 0/1/2 传输错误中断使能 0: 禁止 1: 使能

DMAIF (DMA 中断标志寄存器)			基地址: 0x40016000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:						TEIF2	TEIF1	TEIF0
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:		BCIF2	BCIF1	BCIF0		TCIF2	TCIF1	TCIF0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TCIF[2...0]	通道 0/1/2 传输结束中断标志 0: 未产生中断 1: 产生中断
BCIF[2...0]	通道 0/1/2 块传输完成中断标志 0: 未产生中断 1: 产生中断
TEIF[2...0]	通道 0/1/2 传输错误中断标志 0: 未产生中断 1: 产生中断

注意:

1. BCIE[2...0] 和 BCIF[2...0] 位只有传输模式为块传输时 (TMOD=1) 有效, 单次传输模式无效;
2. 传输模式设置为块传输, BCIF[2...0] 在每次块传输完成置位, 所有数据块 (CHNCNT 设置传输数量) 传输完成 TCIF[2...0] 和 BCIF[2...0] 同时置位。

CHNSTA (DMA 状态寄存器)			基地址: 0x40016000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:						BUSY2	BUSY1	BUSY0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
BUSY[2...0]	通道 0/1/2 传输 BUSY 标志 0: 空闲 1: 忙碌

CHNCTL (DMA 通道控制寄存器)			基地址: 0x4001600C, 0x40016024, 0x4001603C 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:			Channel4	Channel3	Channel2	Channel1	Channel0	DESTIN_INC1
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DESTIN_INC0	SOURC_INC1	SOURC_INC0	CYCLE	MODE	SIZE1	SIZE0	DMA_CHN0EN
Write:								
Reset:	0	0	0	0	0	0	0	0

注: x 为 0,1,2

位	功能描述
Channel[4...0]	触发通道选择, 根据上述的 DMA 通道请求列表来定
DESTIN_INC[1...0]	目的地址增量模式 00: 不增加 01: 增加 10: 数据块内循环增加 11: 数据块内循环增加
SOURC_INC[1...0]	源地址地址增量模式 00: 不增加 01: 增加 10: 数据块内循环增加 11: 数据块内循环增加
TMOD	传输模式 0: 单次传输模式 1: 块传输模式 (1 块数据传输过程中不会被打断)
CYCLE	0: 不循环模式 1: 循环模式
PSIZE[1...0]	外设的传送位数: 00: 8 位 01: 16 位 10: 32 位 11: 32bit
DMA_CHN0EN	DMA 通道 0 使能 0: 禁止 1: 使能



<b>CHNSRC</b> (DMA 通道源地址寄存器)			<b>基地址: 0x4001600C, 0x40016024, 0x4001603C</b> <b>偏移地址: 04H</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
Write:								
Reset:	0	0	0	0	0	0	0	0

注: x 为 0,1,2

<b>位</b>	<b>功能描述</b>
ADDR[15...0]	数据传输源地址寄存器

<b>CHNTAR</b> (DMA 通道目的地址寄存器)			<b>基地址: 0x4001600C, 0x40016024, 0x4001603C</b> <b>偏移地址: 08H</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
Write:								
Reset:	0	0	0	0	0	0	0	0

注: x 为 0,1,2

<b>位</b>	<b>功能描述</b>
ADDR[15...0]	数据传输目的地址寄存器

<b>CHNCNT</b> (DMA 通道传输数量寄存器)			<b>基地址: 0x4001600C, 0x40016024, 0x4001603C</b> <b>偏移地址: 0CH</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num15	Num14	Num13	Num12	Num11	Num10	Num9	Num8
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num7	Num6	Num5	Num4	Num3	Num2	Num1	Num0
Write:								
Reset:	0	0	0	0	0	0	0	0

注: x 为 0,1,2

位	功能描述
Num[15...0]	DMA 数据传输个数设置寄存器 最大设置到 65535 个传输数据 如果用户设置的是块传输, 那么该寄存器则表示用户需要传输的数据块个数

CHNTCCNT (DMA 通道已传输数据个数)			基地址: 0x4001600C, 0x40016024, 0x4001603C 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num15	Num14	Num13	Num12	Num11	Num10	Num9	Num8
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num7	Num6	Num5	Num4	Num3	Num2	Num1	Num0
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

注: x 为 0,1,2

位	功能描述
Num[15...0]	指示 DMA 已经传输完成的数据个数 如果用户设置的是块传输, 那么该寄存器则表示 DMA 已经传输完成的数据块个数

CHNBULKNUM (DMA 通道块传输设置寄存器)			基地址: 0x4001600C, 0x40016024, 0x4001603C 偏移地址: 14H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num7	Num6	Num5	Num4	Num3	Num2	Num1	Num0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
NUM[7...0]	块传输数据个数选择, 表示每一个数据块内有多少个数据: 0 — 255 个 如果用户设置的是块传输, 那么该寄存器则表示每一个数据块内有多少个数据

## 19 按键扫描模块（KEYSCAN）

### 19.1 概述

KEYSCAN 的 SCANIN 的功能在 PA5, PA6, PC10, PC9 的复用功能上, SCANOUT 的功能在 SCANOUT0 对应 PD2 的第二功能, SCANOUT1 对应 PD3 的第二功能, SCANOUT2 对应 PD4 的第二功能, SCANOUT3 对应 PD5 的第二功能

按键扫描模块涉及的芯片引脚为固定的 4 个 SCANIN[0...3], 分别为 PC10, PC9, PA5, PA6 的复用功能, 用户如果需要使用按键扫描功能, 则必须把这 4 个引脚配置为第二功能。对于没有引出的 PIN 脚(例如: PC10, PC9), 也必需同时配置。

用户可选择的 SCANOUT[0...3], 如果选择 4\*1 键盘, 则为 4 SCANIN + 1 SCANOUT, 如果选择 4\*2 键盘, 则为 4 SCANIN + 2 SCANOUT。

SCANIN 功能只检测其引脚上的下降沿信号。

当KEYEN=1时, 同时用户配置I/O口为SCANIN功能时, 按键扫描单元的4个SCANIN引脚进入键盘扫描模式(用户可以根据需要选择SCANOUT的数量(最多4个)), 被选择为SCANIN模式的引脚出现下降沿并保持最多20个fosc的电平时间, 则按键扫描中断标志置位, 按键扫描中断使能为CortexM0的第26号中断使能。

### 19.2 特殊功能寄存器列表

KEYSCAN 模块寄存器基地址: 0x40014000				
偏移地址	名称	读写方式	复位值	功能描述
00H	KEYSTA	R	0000H	KEYSCAN状态寄存器
04H	KEYIF	R/W	0000H	按键中断标志寄存器

### 19.3 特殊功能寄存器说明

KEYSTA (KEYSCAN 状态寄存器)			基地址: 0x40014000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	STA15	STA14	STA13	STA12	STA11	STA10	STA9	STA8
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	STA7	STA6	STA5	STA4	STA3	STA2	STA1	STA0
Write:	X	X	X	X	X	X	X	X

Reset:	0	0	0	0	0	0	0	0
--------	---	---	---	---	---	---	---	---

位	功能描述				
STA[0...15]	指示被按下的 KEY 的位置，最多支持 16 个 KEY，也就是 16 个状态 该寄存器共 16bit，也就是代表了 16 个 KEY 的状态，如果有相应的 KEY 被按下， 则相应的状态位被置 1，如果有两个按键同时被按下，则相应的两个位被置 1				
		SCANIN0	SCANIN1	SCANIN2	SCANIN3
	SCANOUT0	STA0	STA4	STA8	STA12
	SCANOUT1	STA1	STA5	STA9	STA13
	SCANOUT2	STA2	STA6	STA10	STA14
	SCANOUT3	STA3	STA7	STA11	STA15

KEYIF (KEY 标志寄存器)			基地址: 0x40014000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:								KEYIF
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
KEYIF	按键中断标志寄存器 该寄存器写 0 清 0

## 20 Cortex-M0 内核简要说明

### 20.1 概述

Cortex-M0 处理器基于一个高集成度、低功耗的 32 位处理器内核，采用一个 3 级流水线冯·诺伊曼结构（Von Neumann architecture）。通过简单、功能强大的指令集以及全面优化的设计（提供包括一个单周期乘法器在内的高端处理硬件），Cortex-M0 处理器可实现极高的能效。

Cortex-M0 处理器采用 ARMv6-M 结构，基于 16 位的 Thumb 指令集，并包含 Thumb-2 技术。提供了一个现代 32 位结构所希望的出色性能，代码密度比其他 8 位和 16 位微控制器都要高。

### 20.2 系统定时器 SysTick

推荐用户用 CMSIS 接口函数 SysTick\_Config(uint32\_t ticks)去配置系统定时器。

比如 Fcpu 为 19.660800MHz，如果将 SysTick 配置为 100mS 周期中断，则只需按照如下方式调用函数即可：

```
SysTick_Config(19660800/10);
```

注意：由于 SysTick 计数器只有 24 位，用户需保证 ticks 不能超过 0xFFFFF。

### 20.3 中断优先级说明

Cortex-M0 处理器紧密集成了一个可配置的嵌套向量中断处理器（NVIC），提供业界领先的中断性能。NVIC 具有以下功能：

- 包含一个不可屏蔽的中断（NMI），优先级固定为-2，为最高
- 包含一个异常处理中断（HardFault），优先级固定为-1，仅低于 NMI
- 其余中断可以配置 4 个优先级，但优先级都低于以上两个中断

4 个可配置中断优先级为 0x0, 0x1, 0x2, 0x3，其中 0x0 优先级最高，0x3 优先级最低。

推荐用户用 CMSIS 接口函数 NVIC\_SetPriority(IRQn\_Type IRQn, uint32\_t priority)去配置中断优先级。

比如配置 DMA 中断优先级为 0（最高），则可以按照如下方式调用函数即可：

```
NVIC_SetPriority(DMA_IRQn, 0x0);
```

注：DMA\_IRQn 为 DMA 中断的中断号。更多中断号参见 8.1 中断向量说明

### 20.4 CMSIS 函数说明

以下表格罗列了部分 CMSIS 函数，方便用户配置中断等相关功能。建议用户统一使用 CMSIS 函数去配置内核寄存器，不要直接去操作内核寄存器。

CMSIS 函数	函数说明
<code>__enable_irq()</code>	使能全局中断
<code>__disable_irq()</code>	关闭全局中断 (注: NMI 和 HardFault 不会被屏蔽)
<code>void NVIC_EnableIRQ(IRQn_Type IRQn)</code>	使能中断号为 <code>IRQn</code> 的中断 (注: 需满足 <code>IRQn &gt;= 0</code> , 对 <code>IRQn &lt; 0</code> 的中断, 没有中断使能位)
<code>void NVIC_DisableIRQ(IRQn_Type IRQn)</code>	屏蔽中断号为 <code>IRQn</code> 的中断 (注: 需满足 <code>IRQn &gt;= 0</code> , 对 <code>IRQn &lt; 0</code> 的中断, 没有中断使能位)
<code>uint32_t NVIC_GetPendingIRQ(IRQn_Type IRQn)</code>	获取中断号为 <code>IRQn</code> 中断的挂起状态
<code>void NVIC_SetPendingIRQ(IRQn_Type IRQn)</code>	设置中断号为 <code>IRQn</code> 中断的挂起状态 (注: 如果对应的中断已经使能, 则调用此函数会触发芯片进入相应的中断处理程序)
<code>void NVIC_ClearPendingIRQ(IRQn_Type IRQn)</code>	清除中断号为 <code>IRQn</code> 中断的挂起状态 (注: 进入相应中断处理程序后, 中断挂起状态会被自动清除)
<code>void NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority)</code>	设置中断号为 <code>IRQn</code> 中断的优先级 (注: NMI 和 HardFault 不可设置优先级, 优先级设置范围为 <code>0x0—0x3</code> )
<code>uint32_t NVIC_GetPriority(IRQn_Type IRQn)</code>	获取中断号为 <code>IRQn</code> 的中断优先级
<code>void NVIC_SystemReset(void)</code>	系统软复位
<code>uint32_t SysTick_Config(uint32_t ticks)</code>	系统定时器配置 (注: <code>ticks</code> 不能超过 <code>0xFFFFFFFF</code> )

注: 中断号 `IRQn` 详细见 8.1 中断向量说明

## 21 EMU 模块

### 21.1 概述

提供单相电能计量所需要的功能，包括有功功率与有功电能、无功功率与无功电能、视在功率与视在电能、电压有效值、电流有效值及频率计算等，支持灵活的防窃电方案和校表方案。

- 在动态范围 5000: 1 内有功误差小于 0.1%
- 无功精度远优于国标 2 级要求
- 有功功率与有功电能；无功功率与无功电能；视在功率与视在电能
- 同时提供两个通道的快速和慢速有功功率、无功功率、视在功率
- 同时支持快速和慢速电压、电流有效值
- 电压频率测量
- 三路 ADC 同步采样波形数据；有功功率、无功功率、视在功率波形数据
- 灵活的自动防窃电方案，窃电阈值通过寄存器可调
- 灵活的潜动与启动方案
- 多种能量计算模式
- 开放快速脉冲计数寄存器，防止上下电时丢失电能
- 脉冲输出 PF/QF/SF 脉宽可选
- 支持单相三线制
- 过零中断检测信号
- 无功移相补偿
- 有效值偏置校正
- 有功功率 P，无功功率 Q 偏置校正
- 超低功耗支持灵活的断零线窃电计量
- 提供 3 路 ADC 自动/手动温度补偿功能

### 21.2 功能描述

电能计量单元 EMU 包括三路完全独立的  $\Sigma$ - $\Delta$  ADC 以及数字信号处理部分。三路 ADC 完成两路电流信号和一路电压信号的采样，数字信号处理部分完成有功功率与有功电能、无功功率与无功电能、视在功率与视在电能、电压有效值、电流有效值及频率计算等计量功能。

通过 SFR 寄存器和中断的方式，可以对数字信号处理部分进行校表参数配置和计量参数读取；计量的结果还通过 PF/QF/SF 引脚输出，也即校表脉冲输出，可以直接接到标准表进行误差对比。

#### 21.2.1 模数转换器

HT500X 内置三路完全独立的二阶  $\Sigma$ - $\Delta$  ADC，每路 ADC 都有一个模拟增益放大器 (PGA)，内部有一个 1.2V 的高稳定度片内基准电压，每路 ADC 可以独立开关，通过寄存器 EMUCTRL 进行设置。

模拟增益放大器 (PGA) 完成输入差分信号的幅度放大，放大后的信号再送给 ADC 进行采样，在极小

信号输入时能够保证测量的线性度。通过寄存器 ADCCON(68H) 可以对三路 ADC 独立配置模拟放大倍数。

通过寄存器 ADCCON(68H) 可以对三路 ADC 采到的信号进行数字增益设置, 三路 ADC 通道有 1、2、4、8 倍四种设置。在大信号不溢出的情况下, 数字增益放大可以增加小信号计算的有效位数, 进一步提高计量精度。

3 路 ADC 默认关闭, 可以通过校表参数寄存器 EMUCTRL 单独选择关闭或者开启。

第 2 路电流通道 ADC 同时提供增益校正寄存器 I2GAIN(30H), 可以对第 2 路电流通道 ADC 的幅值进行比例缩放, 以用在自动防窃电等功能上。

### 21.2.2 ADC 采样输出和功率波形输出

HT500X 将三路 ADC 输出的 22bit 波形数据 SPLI1 (00H)、SPLI2(04H)、SPLU(08H) 开放给用户。同时也将作为电能累加的功率波形数据 SPLP(0CH)、SPLQ(10H) 开放给用户, 波形采样数据可通过寄存器 ADCCFG (50H.WaveSel) 配置为高通前或高通后。

EMU 的系统时钟频率可通过 EMUCTRL 配置为  $F_{emu}=819.2\text{ KHz}$  和  $409.6\text{ KHz}$ , 波形数据更新频率可通过 ADCCFG (50H.SPL[2: 0]) 选择为  $12.8\text{ kHz}$ 、 $6.4\text{ kHz}$ 、 $3.2\text{ kHz}$ 、 $1.6\text{ kHz}$ 、 $0.8\text{ kHz}$ 。

注意: 由于波形数据更新频率最快可以达到  $12.8\text{ KHz}$ , 因此 ADC\_IRQ 的中断请求最快也可以达到  $12.8\text{ KHz}$ 。用户在编程时需要注意 CPU 程序是否有足够的时间来响应中断。

### 21.2.3 有功功率、无功功率和视在功率

HT500X 同时输出两路计量通道的有功功率、无功功率, 并提供两路独立的校表参数, 提供视在功率输出寄存器。



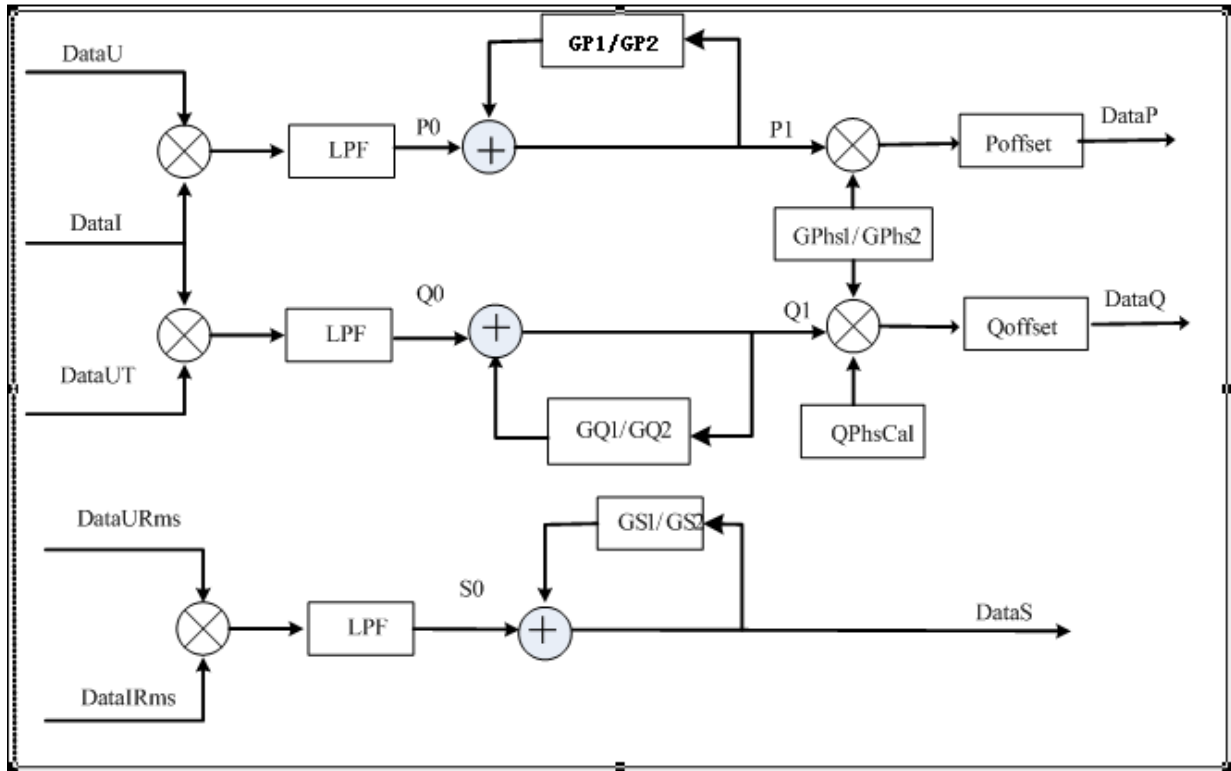


图 22-2-1 功率计算和补偿

有功功率通过电压、电流相乘经过低通滤波器后得到。

无功功率计算时，先将电压移相 90 度，然后通过移相后电压和电流相乘，经过低通滤波器得到。

视在功率通过电压有效值与电流有效值相乘得到。

对于通道 1 和通道 2 的功率分别提供增益校正和相位校正，同时针对小信号的精度问题，提供了偏置校正来消除外界干扰。

对电压移相 90 度的滤波器性能与输入信号的频率及 ADC 采样率相关，可以通过无功相位补偿寄存器 QphsCal(2CH) 对 90° 移相进行相位补偿。

## 21.2.4 有效值

HT500X 同时输出两路电流和一路电压的有效值。

有效值可以保证在动态范围为 3000: 1 时，精度达到 0.5%。

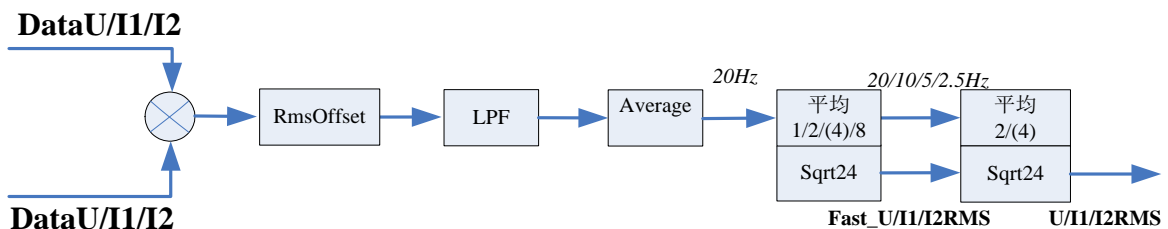


图 22-2-2

在正常模式下（即 EMU 工作时钟为 819.2KHz/408.6kHz），有效值寄存器输出如上图所示，用户可根据需要选择快速或慢速有效值。

在 EMU 处于低功耗计量模式下（即 EMU 工作时钟为 204.8KHz/32kHz），芯片只提供快速有效值，更新时间为 0.5s/1s/2s/4s，可通过校表寄存器 94HROSICTRL 的 LpCtrl[1:0]来设置，此时慢速有效值寄存器读出固定为 0。

## 21.2.5 电压线频率

通过对电压采样信号过零点计数的方式，提供电压频率输出。  
频率的精度可以保证准确到 0.01Hz。

## 21.2.6 起动/潜动

通过设置校表寄存器 PQStart(40H)，用户可以灵活的完成防潜动与起动的功能。

|P|小于 PStart 时，有功脉冲 PF 不输出脉冲。

|Q|小于 QStart 时，无功脉冲 QF 不输出脉冲。

S小于SStart时，视在脉冲SF不输出脉冲。

HT500X内部计量参数寄存器提供快速(24H~38H)、慢速(48H~5CH)两组功率寄存器,用户可通过校表寄存器ADCCFG(50H)中的控制位PQSSStarScouceSel[1.0]选择快速或者慢速功率作为判断源，其中选择快速功率作为起动功率源头可以有效满足动态负荷试验，而选择慢速功率作为起动功率源头有利于可靠防潜。

另外状态寄存器 EMUSR(00H)的 NOQLD1 NOPLD1 NOQLD2 NOPLD2 能够实时显示两路电能是否起动，方便用户对阈值的选取。

## 21.2.7 功率反向指示

通过状态寄存器 EMUSR(00H)的 REVQ、REVP 可以指示无功功率、有功功率是否反向。

REVQ、REVP 在 PF、QF 发脉冲的开始沿同步更新。

## 21.2.8 直流偏置校正

用户通过 CHNLCCR(54H)的 HPFONU、HPFONI2、HPFONI1 可以分别控制电压和两路电流的高通滤波器环节是否打开。当这些位为 1 时，表示高通开启，反之高通关闭。

当高通滤波器关闭后，直流信号可以进入后续计算通道，可以通过偏置校正寄存器 I1Off(34H)、I2Off(38H)、UOff(3CH)对采样到的数据进行直流偏置校正。

进行直流偏置校正时，需要将输入通道短接（即输入为 0），读取 ADC 采样输出波形寄存器 SPL\_I1(00H)、SPL\_I2(01H)、SPL\_U(02H)进行平均后，写入偏置校正寄存器 I1Off(34H)、I2Off(38H)、UOff(3CH)，即完成直流偏置校正。

## 21.2.9 电能脉冲输出

HT500X 提供有功能量寄存器 ENERGY\_P(60H)、无功能量寄存器 ENERGY\_Q(64H)和视在能量寄存器

ENERGY\_S(68H)，同时提供相应的脉冲输出引脚 PF、QF 和 SF 用于校准。

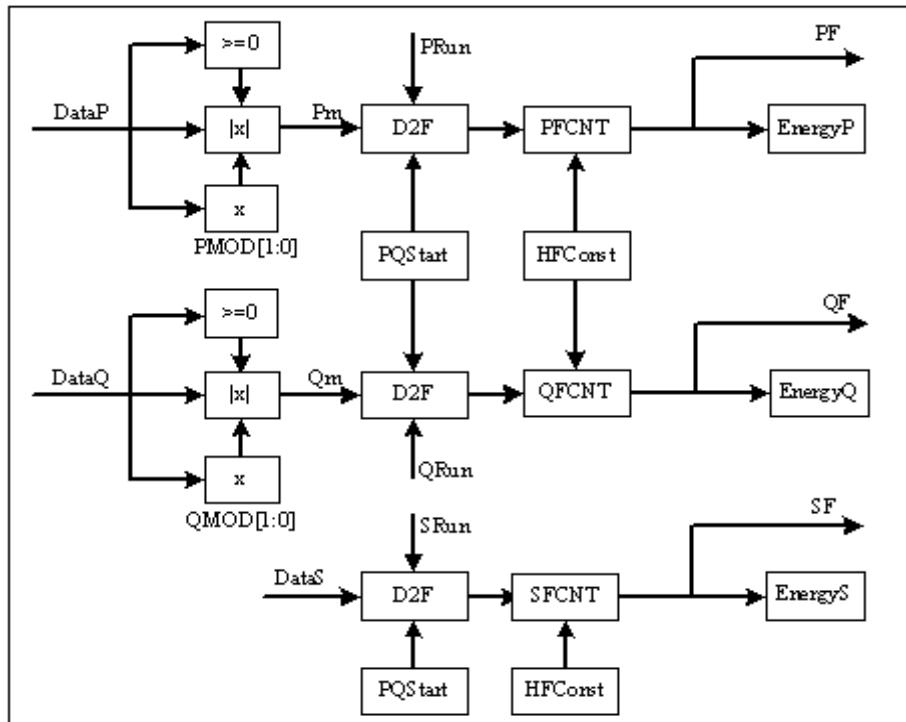


图 22-2-3 能量计量和脉冲输出

脉冲输出前的内部电能累加方式可以通过 EMCON(58H)的 QMOD、PMOD 选择正向计量、绝对值计量、代数和计量三种累加方式。

内部功率值寄存器对功率进行累加，溢出后会产生一个溢出脉冲到快速脉冲寄存器 PFCNT(5CH)、QFCNT(60H)和 SFCNT(64H)。快速脉冲计数寄存器对溢出的次数进行累加计数。当快速脉冲寄存器中的计数绝对值大于等于输出脉冲频率设置寄存器 HFCnst(4CH)的设置时，即发出一个 CF 脉冲，同时相应能量寄存器的值增加 1。

HT500X 开放了快速脉冲计数器 PFCNT/QFCNT/SFCNT，可以通过掉电时读取这些寄存器内不足一个脉冲的能量值并保存，重新上电时再将保存值写回快速脉冲计数器寄存器，可以防止掉电时少计电能。

脉冲输出管脚 PF/QF/SF 和能量寄存器受到 EMCON(58H)的 PRun/QRun/SRun 以及 PQStart(40H)的控制。用户可以通过 POS (52H.6) 选择 PF/QF/SF 的有效电平。POS 为 0 时，脉冲高电平有效；POS 为 1 时，脉冲低电平有效。

PF/QF/SF 引脚默认分别输出有功、无功、视在电能，可通过 EMCON(58H)寄存器中的 CFxCFG 进行配置，同时可通过控制位 POS(54H.6)选择 PF/QF/SF 引脚输出有效电平。

PF/QF/SF 输出满足下面时序关系：

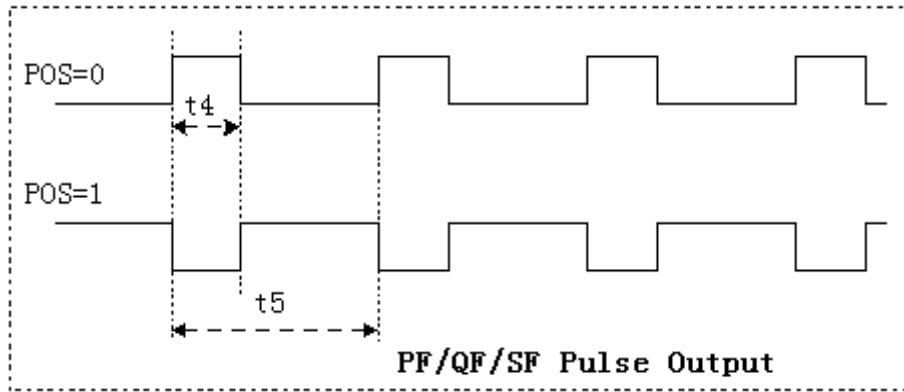


图 22-2-4 脉冲输出波形

时序特性:

参数	指标	单位	测试条件及注释
t4	80	ms	PF/QF/SF输出的高电平脉宽
t5	—	s	PF/QF/SF输出的周期

注意：当脉冲输出周期小于 180ms 时，脉冲以等 duty 形式输出。

HT500X可以实现输出脉冲宽度（t4）可调，通过CFP[1:0]( 68H.[7:6])进行设置。

## 21.2.10 窃电检测

可以通过防窃电模块对两路电流或者两路功率大小进行比较，选用较大的一路电流或功率进行计量。

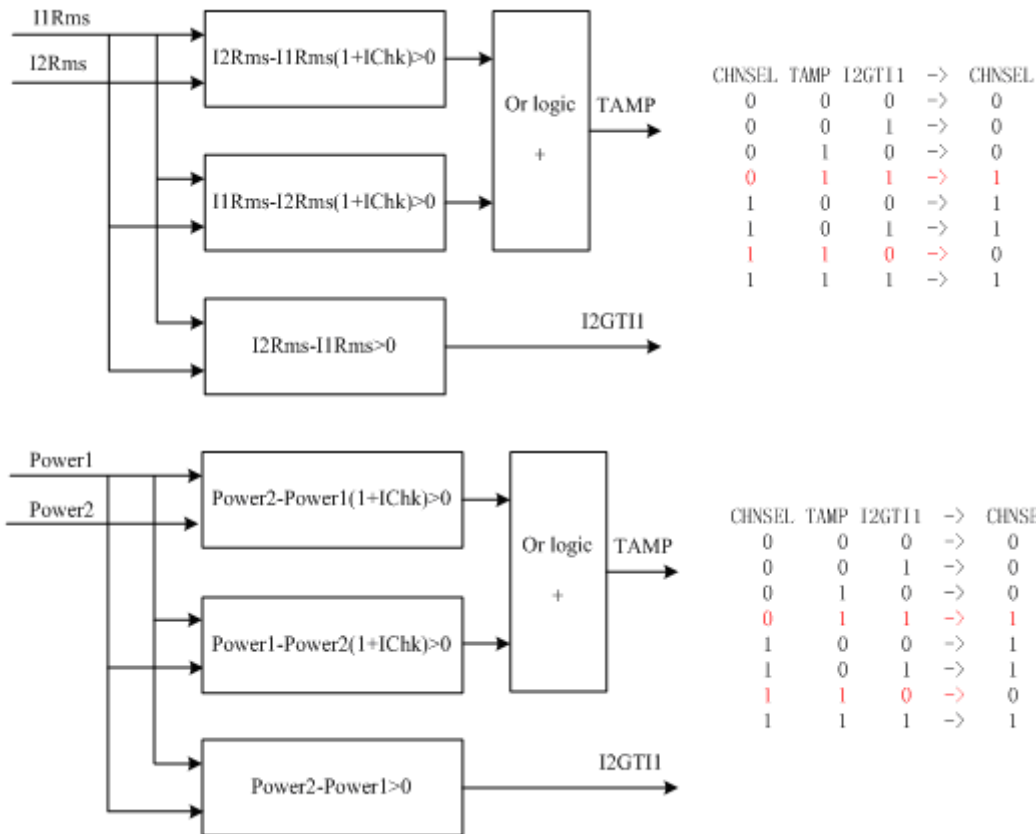


图 22-2-5 自动防窃电方案

可以通过 FLTON (50H.5) 设置是否开启自动防窃电功能。FLTON=0 时, 用户可以根据 CHNSEL(50H.4) 进行通道选择; FLTON=1 时, 防窃电单元根据用户的设置, 自动选择相应的通道进行计量。

通过 ICHK(70H) 用户可以设置发生窃电的比例, 比如可以设置 0x10, 表示两路电流有效值相差到 6.25% 时认为发生了窃电。

当两路电流的有效值 (或者功率值) 均小于 IPTAMP(6CH) 的值时, 增加寄存器位 CHNFix(50H.8) 可选:

- (1) 默认选择通道 1 为计量单元
- (2) 不切换, 选择之前的计量通道。

I2GT1(00H.6) 为 0 表示 I1 大于 I2, 为 1 时表示 I2 大于 I1。

TAMP(00H.7) 为 1 表示发生了窃电, 即两路电流相比超过了设定的防窃电阈值。

#### 自动防窃电设置步骤: (以电流防窃电为例)

- a) 通过 EMUCTRL (74H.5) 开启电流通道 2ADC。
- b) 通过 I2GAIN(30H) 对电流通道 2 的输出校正, 保证同样的输入电流时, 两个通道的有效值输出一致。
- c) 根据需要的防窃电阈值, 设置 ICHK(70H)。
- d) 根据需要检测防窃电的最小电流或功率, 设置 IPTAMP(6CH)。
- e) 设置 FLTON(0x50H.5) 为 1, 开启自动防窃电功能。

在自动防窃电功能打开后, CHNSEL 和 CIADD 处于只读状态, 通道选择由防窃电的结果决定, 可以通过寄存器位 CHNSEL/TAMP/I2GTI1 查看防窃电状态。

## 21.2.11 中断源

HT500X 的 EMU 提供 15 个中断源, 在 EMU 中断使能寄存器 (EMUIE, 04H) 中定义: Tamp\_IRQ、DCUpdate\_IRQ、SlowRmsUpdate\_IRQ、FastRmsUpdate\_IRQ、ZXI2\_IRQ、ZXI1\_IRQ、I2OV\_IRQ、PF\_IRQ、QF\_IRQ、SF\_IRQ、SPL\_IRQ、ZXU\_IRQ、I1OV\_IRQ、UOV\_IRQ 和 SAG\_IRQ, 分别表示窃电中断请求、直流有效值更新中断请求、慢速有效值更新中断请求、快速有效值更新中断请求、I2 过零中断请求、I1 过零中断请求、I2 过流中断请求、有功脉冲中断请求、无功脉冲中断请求、视在脉冲中断请求、ADC 原始数据刷新中断请求、电压过零中断请求、I1 过流中断请求、电压过压中断请求和电压跌落中断请求。过零方式通过 ZXD1 和 ZXD0 可以选择正向过零或者反向过零或者双向过零 (详见 74H)。

EMU 15 个中断共用一个中断向量 IRQ\_EMU。通过 EMU 中断使能寄存器 EMUIE 和 EMU 中断标志寄存器 EMUIF, 用户可以实现中断的控制和管理。

用户读取 EMUIF 后即可清除中断标志。

## 21.2.12 SAG 功能描述

电压跌落检测(SAG), 此模块工作在正常计量模式下, 属于电能质量范畴。该功能使用的数据是高通后的数据。

以半周波为单位(2 个过零点之间), 找到 ADC 采样值绝对值的最大值。SAG 事件的判据为: 电压波形采样值绝对值的峰值的高 16 位小于 SAG 阈值设置寄存器 SAGLVL(A4H) 的设置值, 且持续时间为 SAG 长度设置寄存器 USagCyc (A8H) 设定的半周波数, 则判定电压暂降即 SAG 事件发生。当 USagCyc=0x0000 时, 关闭 SAG 功能。

校表寄存器:

- 1) 使能 SAG 功能 (Sag\_En), 观测长度寄存器 USagCyc (A8H): 以半周波为单位;
- 2) 阈值设置寄存器 SAGLVL(A4H): 给定的比较阈值, 与 ADC 采样值高 16bit 对齐;
- 3) SAG 中断使能位 EMUIE.0 (04H): 置 1 可使能 SAG 中断输出。

参数寄存器:

- 1) SAG 状态标志位 EMUIF.0 (08H): 读到 1 表示发生 SAG, 读后清 0;
- 2) 电压波形点最值 UPeak (88H): 波形点的最大峰值, 有符号数, 该寄存器为 24bit, Bit21—Bit23 为符号位。

### 21.2.13 OVI 功能描述

过压过流检测(OVI), 此模块工作在正常计量模式下, 属于电能质量范畴。该功能使用的数据是高通后的数据, 如果高通关闭, 则选择 ADCOFFSET 后的数据。

以半周波为单位(2 个过零点之间), 找到 ADC 采样值绝对值的最大值。OVI 事件的判据为: 电压或者电流波形采样值绝对值的峰值的高 16 位大于电压或电流峰值检测阈值设置寄存器 UOVLVL(ACH)或 IOVLVL(B4H)的设置值, 且持续时间为 OVCyc 长度设置寄存器 OVCyc (B0H) 设定的半周波数, 则判定电压过压或者电流过流事件发生。当 OVCyc=0x0000 时, 关闭 OVI 功能。

校表寄存器:

- 1) 使能 OVI 功能 (OvI\_En), 观测长度寄存器 OVCyc (B0H): 以半周波为单位;
- 2) 阈值设置寄存器 UOVLVL(ACH)或 IOVLVL(B4H): 给定的比较阈值, 与 ADC 采样值高 16bit 对齐;
- 3) OVI 中断使能位 UOVIE、I1OVIE、I2OVIE (04H): 置 1 可使能过压、过流中断输出。

参数寄存器:

- 1) OVI 状态标志位 UOVIF、I1OVIF、I2OVIF (08H): 读到 1 表示发生过压、过流, 读后清 0;
- 2) 电压波形点最值 UPeak (88H)、I1Peak (8CH)、I2Peak (90H): 波形点的最大峰值, 有符号数, 该寄存器为 24bit, Bit21—Bit23 为符号位。

### 21.2.14 掉零线计量模式(Loss of Voltage)

HT500X 在用户进入失压模式, 为兼顾计量精度与功耗的矛盾, 提供了几种计量方式用于掉零线计量。

1. 芯片内部使用寄存器 UCONST 替代电压有效值信号与电流有效值相乘计算视在功率, 视在电能, 从有功电能输出引脚 (PF) 输出脉冲信号。

在失压模式下, 用户需要配置的寄存器如下:

- (1) UCONST: 用于替代电压有效值的电压常数
- (2) HFCONST: 用于在 100%Ib 点对视在电能脉冲输出进行精度校正 (粗调)
- (3) SGAIN: 用于在 100%Ib 点对视在电能脉冲输出进行精度校正 (细调)
- (4) ADCx\_EN: 用于选择开启哪一路电流通道 (控制失压模式下的功耗)
- (5) Channel\_Sel: 用于选择使用哪一路电流通道计量电能
- (6) SRUN: 用于打开视在电能计量
- (7) PSSel: 用于配置视在电能脉冲从有功电能脉冲引脚 (PF) 输出

2. 芯片内部提供低功耗计量模式, 可配置 EMU 时钟为 204.8KHz 或 32KHz, 详细应用可见应用笔记;

3 芯片内部支持自定义常数计量模式，当用户写入常数功率寄存器后，使能常数计量，芯片内部会一直用该常数功率累加发脉冲并累加电能：

## 21.3 特殊功能寄存器列表

### 21.3.1 计量参数寄存器

#### 21.3.1.1 计量参数寄存器列表

EMU 模块计量寄存器基地址： 计量寄存器：0x40013000					
偏移地址	名称	读写方式	有效字长	复位值	功能描述
00H	<a href="#">SPLI1</a>	R	3	0x000000	电流通道 1 的 ADC 采样数据
04H	<a href="#">SPLI2</a>	R	3	0x000000	电流通道 2 的 ADC 采样数据
08H	<a href="#">SPLU</a>	R	3	0x000000	电压通道的 ADC 采样数据
0CH	<a href="#">SPLP</a>	R	4	0x00000000	有功功率波形数据
10H	<a href="#">SPLQ</a>	R	4	0x00000000	无功功率波形数据
14H	<a href="#">FastRMSI1</a>	R	3	0x000000	快速电流通道 1 的有效值
18H	<a href="#">FastRMSI2</a>	R	3	0x000000	快速电流通道 2 的有效值
1CH	<a href="#">FastRMSU</a>	R	3	0x000000	快速电压通道的有效值
20H	<a href="#">FreqU</a>	R	2	0x0000	电压频率
24H	<a href="#">FastPowerP1</a>	R	4	0x00000000	快速第一路有功功率
28H	<a href="#">FastPowerQ1</a>	R	4	0x00000000	快速第一路无功功率
2CH	<a href="#">FastPowerP2</a>	R	4	0x00000000	快速第二路有功功率
30H	<a href="#">FastPowerQ2</a>	R	4	0x00000000	快速第二路无功功率
34H	<a href="#">FastPowerS1</a>	R	4	0x00000000	快速第一路视在功率
38H	<a href="#">FastPowerS2</a>	R	4	0x00000000	快速第二路视在功率
3CH	<a href="#">RMSI1</a>	R	3	0x000000	慢速电流通道 1 的有效值
40H	<a href="#">RMSI2</a>	R	3	0x000000	慢速电流通道 2 的有效值
44H	<a href="#">RMSU</a>	R	3	0x000000	慢速电压通道的有效值
48H	<a href="#">PowerP1</a>	R	4	0x00000000	慢速第一路有功功率
4CH	<a href="#">PowerQ1</a>	R	4	0x00000000	慢速第一路无功功率
50H	<a href="#">PowerP2</a>	R	4	0x00000000	慢速第二路有功功率
54H	<a href="#">PowerQ2</a>	R	4	0x00000000	慢速第二路无功功率
58H	<a href="#">PowerS1</a>	R	4	0x00000000	慢速第一路视在功率

5CH	<a href="#">PowerS2</a>	R	4	0x00000000	慢速第二路视在功率
60H	<a href="#">EnergyP</a>	R	3	0x000000	有功能量
64H	<a href="#">EnergyQ</a>	R	3	0x000000	无功能量
68H	<a href="#">EnergyS</a>	R	3	0x000000	视在能量
6CH	<a href="#">EnergyPC</a>	R	3	0x000000	读后清 0 型有功能量
70H	<a href="#">EnergyQC</a>	R	3	0x000000	读后清 0 型无功能量
74H	<a href="#">EnergySC</a>	R	3	0x000000	读后清 0 型视在能量
78H	<a href="#">DC_UAverage</a>	R	4	0x00000000	直流电压平均值
7CH	<a href="#">DC_I1Average</a>	R	4	0x00000000	直流电流通道 1 平均值
80H	<a href="#">DC_I2Average</a>	R	4	0x00000000	直流电流通道 2 平均值
84H	<a href="#">Checksum</a>	R	3	0x000000	校验和寄存器, 这个寄存器来确保校表寄存器没有发生改变
88H	<a href="#">Upeak</a>	R	3	0x000000	电压波形峰值寄存器, 19bit, 和 ADC 位数一致
8CH	<a href="#">I1peak</a>	R	3	0x000000	I1 电流波形峰值寄存器, 19bit, 和 ADC 位数一致
90H	<a href="#">I2peak</a>	R	3	0x000000	I2 电流波形峰值寄存器, 19bit, 和 ADC 位数一致
94H	<a href="#">PFCNT_Photo</a>	R	2	0x0000	PFCNT 快照
98H	<a href="#">QFCNT_Photo</a>	R	2	0x0000	QFCNT 快照
9CH	<a href="#">SFCNT_Photo</a>	R	2	0x0000	SFCNT 快照
A0H	<a href="#">AutoUgain</a>	R	2	0x0000	U 通道温度自动补偿系数
A4H	<a href="#">AutoI1gain</a>	R	2	0x0000	I1 通道温度自动补偿系数
A8H	<a href="#">AutoI2gain</a>	R	2	0x0000	I2 通道温度自动补偿系数

### 21.3.1.2 计量寄存器说明

SPLI1		基地址: 0x40013000					
		偏移地址: 00H					
	Bit23...Bit0						
Read:	DAT[23:0]						
Write:	X						
Reset:	0	0	0	0	0	0	0

SPLI2		基地址: 0x40013000					
		偏移地址: 04H					
	Bit23...Bit0						
Read:	DAT[23:0]						
Write:	X						



Reset:	0	0	0	0	0	0	0	0
--------	---	---	---	---	---	---	---	---

<b>SPLU</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 08H</b>						
		<b>Bit23...Bit0</b>						
<b>Read:</b>	DAT[23:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

波形寄存器的更新速度由时钟配置寄存器 FreqCFG. [2: 0]的 3 个 bit 控制。此 3 个寄存器有效位数为 22 位, bit21 为符号位, 同时该符号位扩展到 24 位, 也就是说读出的数据位中 bit23—bit21 都是符号位。ADC 波形寄存器可通过 **ADCCFG** (50H.WaveSel) 配置为高通前或高通后的数据。数据采用二进制补码格式。

<b>SPLP</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 0CH</b>						
		<b>Bit31...Bit0</b>						
<b>Read:</b>	DAT[31:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>SPLQ</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 10H</b>						
		<b>Bit31...Bit0</b>						
<b>Read:</b>	DAT[31:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

功率波形寄存器的更新速度由时钟配置寄存器 FreqCFG. [2: 0]的 3 个 bit 控制。此 2 个寄存器有效位数为 32 位, bit31 为符号位。ADC 波形寄存器可通过 **ADCCFG** (50H.WaveSel) 配置为高通前或高通后的数据。数据采用二进制补码格式。

<b>FastRMSI1</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 14H</b>						
		<b>Bit23...Bit0</b>						
<b>Read:</b>	DAT[23:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

FastRMSI2		<b>基地址:</b> 0x40013000 <b>偏移地址:</b> 18H						
	Bit23...Bit0							
<b>Read:</b>	DAT[23:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

FastRMSU		<b>基地址:</b> 0x40013000 <b>偏移地址:</b> 1CH						
	Bit23...Bit0							
<b>Read:</b>	DAT[23:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

快速有效值 Rms 是 24 位的无符号数，最高位恒为 0。当 EMU 时钟频率为 819.2KHz 时，参数更新频率默认为 5Hz，可配置 EMUCTRL[1: 0]到最高 20Hz。

如果用户需要在小信号时得到更加准确的有效值寄存器值，则需要通过 IIRMSOFFSET 和 I2RMSOFFSET 这两个寄存器对有效值进行零漂校正。

FreqU		<b>基地址:</b> 0x40013000 <b>偏移地址:</b> 20H						
	Bit15...Bit0							
<b>Read:</b>	DAT[15:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

频率值是一个 16 位的无符号数，参数格式化公式为:

$$Frequency = \frac{femu}{(UFREQ \times 2)}$$

其中:

femu—— 计量模块工作频率，默认 819.2KHz

例如，如果 EMU 时钟（femu）选择为 819.2KHz，寄存器 UFREQ=8192，

那么测量到的实际频率为：Frequency=819.2KHz/2/8192=50Hz。

FastPowerP1		<b>基地址:</b> 0x40013000 <b>偏移地址:</b> 24H						
	Bit31...Bit0							

Read:	X							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

<b>FastPowerQ1</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 28H</b>						
		Bit31...Bit0						
Read:	X							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

<b>FastPowerP2</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 2CH</b>						
		Bit31...Bit0						
Read:	X							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

<b>FastPowerQ2</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 30H</b>						
		Bit31...Bit0						
Read:	X							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

<b>FastPowerS1</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 34H</b>						
		Bit31...Bit0						
Read:	X							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

<b>FastPowerS2</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 38H</b>						
		Bit31...Bit0						
Read:	X							
Write:	DAT[31:0]							

Reset:	0	0	0	0	0	0	0	0
--------	---	---	---	---	---	---	---	---

<b>RMSI1</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 3CH</b>						
		Bit23...Bit0						
<b>Read:</b>	DAT[23:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>RMSI2</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 40H</b>						
		Bit23...Bit0						
<b>Read:</b>	DAT[23:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>RMSU</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 44H</b>						
		Bit23...Bit0						
<b>Read:</b>	DAT[23:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

慢速有效值 Rms 是 24 位的无符号数，最高位恒为 0。当 EMU 时钟频率为 819.2KHz 时，参数更新频率默认为 1.25Hz。配置 EMUCTRL [7]调整更新速率，可对应到快速有效值更新速率的 1/2 或 1/4。

如果用户需要在小信号时得到更加准确的有效值寄存器值，则需要通过 I1RMSOFFSET 和 I2RMSOFFSET 这两个寄存器对有效值进行零漂校正。

<b>PowerP1</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 48H</b>						
		Bit31...Bit0						
<b>Read:</b>	X							
<b>Write:</b>	DAT[31:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

PowerQ1		<b>基地址:</b> 0x40013000 <b>偏移地址:</b> 4CH						
		Bit31...Bit0						
Read:	X							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

PowerP2		<b>基地址:</b> 0x40013000 <b>偏移地址:</b> 50H						
		Bit31...Bit0						
Read:	X							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

PowerQ2		<b>基地址:</b> 0x40013000 <b>偏移地址:</b> 54H						
		Bit31...Bit0						
Read:	X							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

PowerS1		<b>基地址:</b> 0x40013000 <b>偏移地址:</b> 58H						
		Bit31...Bit0						
Read:	X							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

PowerS2		<b>基地址:</b> 0x40013000 <b>偏移地址:</b> 5CH						
		Bit31...Bit0						
Read:	X							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

注意:

快速/慢速功率都为 32 位二进制补码格式，最高位为符号位，快速功率寄存器参数默认更新频率为 5Hz（EMU 时钟频率为 819.21KHz，配置 EMU 时钟为 409.6KHz 时则为 2.5Hz），可通过配置 EMUCTRL[1: 0] 到最高 20Hz。

慢速功率寄存器参数默认频率为 1.25Hz（EMU 时钟频率为 819.21KHz，配置 EMU 时钟为 409.6KHz 时则为

0.625Hz)，通过配置 EMUCTRL [7]调整更新速率，可对应到快速有效值更新速率的 1/2 或 1/4。

两路功率参数 FastPowerP1/P2、FastPowerQ1/Q2、FastPowerS1/S2、PowerP1/P2、PowerQ1/P2、PowerS1/S2 是二进制补码格式，32 位数据，其中最高位是符号位。

设寄存器中的数据为 PowerP1，则供计算用的 Preg 为：

Preg=PowerP1 ; 如果 PowerP1<2^31

Preg=PowerP1-2^32 ; 如果 PowerP1>=2^31

设显示的有功功率为 P，转换系数为 Kpqs，则：

$P = \text{Preg} \times \text{Kpqs}$

Kpqs 为额定有功功率输入时，额定功率与 PowerP1 读数的比值。

无功功率和视在功率做显示时的系数与有功功率的系数 Kpqs 相同。

例：

输入 1000w 有功功率，PowerP1 读数平均为 0x00C9D9(51673)，则

$\text{Kpqs} = 1000/51673 = 0.01935$

当 PowerP1 读数为 0xFFFF4534 时，其代表的功率值为：

$P = \text{Kpqs} * \text{Preg} = 0.01935 * (-47820) = -925.3 \text{ w}$

其中  $\text{Preg} = \text{PowerP1} - 2^32 = -47820$

EnergyP		基地址： 0x40013000						
		偏移地址： 60H						
		Bit23...Bit0						
Read:	X							
Write:	DAT[18:0]							
Reset:	0	0	0	0	0	0	0	0

注意：

该能量累加寄存器为读后不清 0，该寄存器的最小单位代表的能量为 1/EC kWh。

例：脉冲常数 EC 为 3200imp/kWh，寄存器读数为 0x001000（4096）时，其代表的能量为

$E = 4096/3200 = 1.28 \text{ kWh}$

EnergyQ		基地址： 0x40013000						
		偏移地址： 64H						
		Bit23...Bit0						
Read:	X							
Write:	DAT[18:0]							
Reset:	0	0	0	0	0	0	0	0

EnergyS		基地址： 0x40013000						
		偏移地址： 68H						

	Bit23...Bit0							
Read:	X							
Write:	DAT[23:0]							
Reset:	0	0	0	0	0	0	0	0

EnergyPC	基地址: 0x40013000 偏移地址: 6CH							
	Bit23...Bit0							
Read:	X							
Write:	DAT[23:0]							
Reset:	0	0	0	0	0	0	0	0

注意:

该能量累加寄存器为读后清 0，该寄存器的最小单位代表的能量为 1/EC kWh。

例：脉冲常数EC为3200imp/kWh，寄存器读数为0x001000（4096）时，其代表的能量为

$$E = 4096 / 3200 = 1.28 \text{ kWh}$$

EnergyQC	基地址: 0x40013000 偏移地址: 70H							
	Bit23...Bit0							
Read:	X							
Write:	DAT[23:0]							
Reset:	0	0	0	0	0	0	0	0

EnergySC	基地址: 0x40013000 偏移地址: 74H							
	Bit23...Bit0							
Read:	X							
Write:	DAT[23:0]							
Reset:	0	0	0	0	0	0	0	0

DC_UAverage	基地址: 0x40013000 偏移地址: 78H							
	Bit31...Bit0							
Read:	X							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

DC_I1Average		<b>基地址:</b> 0x40013000 <b>偏移地址:</b> 7CH						
		Bit31...Bit0						
<b>Read:</b>	X							
<b>Write:</b>	DAT[31:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

DC_I2Average		<b>基地址:</b> 0x40013000 <b>偏移地址:</b> 80H						
		Bit31...Bit0						
<b>Read:</b>	X							
<b>Write:</b>	DAT[31:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

直流平均值 DC\_Average 是 32 位的数。当 EMU 时钟频率为 819.2KHz 时，参数更新频率默认为 8Hz。配置 EMUCTRL [9:8] (74H) 调整更新速率为 1、2、4、8Hz。

如果用户需要在小信号时得到更加准确的直流平均值，则需要通过直流 Uoff、I1off、I2off 寄存器进行零漂校正。

Checksum		<b>基地址:</b> 0x40013000 <b>偏移地址:</b> 84H						
		Bit23...Bit0						
<b>Read:</b>	X							
<b>Write:</b>	DAT[23:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

参数和校验寄存器为所有校表参数寄存器的和，基地址 0x40013800，偏移地址 0CH---104H（计算不包含 5CH-64H 寄存器）。

Scheck 寄存器的计算方式为:

对所有的校验寄存器采用三字节的无符号数加法，两/单字节寄存器高位补 0。

若配置过校表寄存器，此校验和寄存器随即更新，等待时间为 2us 左右，可忽略不计。

Upeak		<b>基地址:</b> 0x40013000 <b>偏移地址:</b> 88H						
		Bit24...Bit0						
<b>Read:</b>	X							
<b>Write:</b>	DAT[24:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

I1peak		<b>基地址:</b> 0x40013000						
--------	--	------------------------	--	--	--	--	--	--



		偏移地址: 8CH						
		Bit24...Bit0						
Read:	X							
Write:	DAT[24:0]							
Reset:	0	0	0	0	0	0	0	0

I2peak		基地址: 0x40013000						
		偏移地址: 90H						
		Bit24...Bit0						
Read:	X							
Write:	DAT[24:0]							
Reset:	0	0	0	0	0	0	0	0

注意:

峰值寄存器, 和 OVI 功能相关, 当使能 OVI 功能且经过用户指定的 ADC 半周波后, 将该寄存器更新, 得到此段时间内的 ADC 波形最大值(绝对值), 该寄存器取自高通后的数据, 该寄存器为 19bit. Bit18 为符号位, 同时该符号位扩展到 24 位, 也就是说读出的数据位中 bit23—bit18 都是符号位。数据采用二进制补码格式。

PFCNT_Photo		基地址: 0x40013000						
		偏移地址: 94H						
		Bit15...Bit0						
Read:	X							
Write:	DAT[15:0]							
Reset:	0	0	0	0	0	0	0	0

QFCNT_Photo		基地址: 0x40013000						
		偏移地址: 98H						
		Bit15...Bit0						
Read:	X							
Write:	DAT[15:0]							
Reset:	0	0	0	0	0	0	0	0

SFCNT_Photo		基地址: 0x40013000						
		偏移地址: 9CH						
		Bit15...Bit0						
Read:	X							
Write:	DAT[15:0]							
Reset:	0	0	0	0	0	0	0	0

注意:

快速能量脉冲快照寄存器, 当用户读取有功能量寄存器时 (不清零能量寄存器/清零型能量寄存器), 芯片会

在 5 个 EMU CLK 时间内把 PFCNT 抓取放入 PFCNT\_Photo 寄存器，以提高能量寄存器的分辨率。

例如：读取 PFCNT\_Photo 到的值为 0x20，Hfconst 寄存器值为 0x40，则 PFCNT\_Photo 表示的电能为：

$(0x20/0x40) / EC \text{ kWh}$

QFCNT\_Photo SFCNT\_Photo 功能相同。

AutoUgain	<b>基地址： 0x40012000</b> <b>偏移地址： A0H</b>							
	Bit15...Bit0							
Read:	X							
Write:	DAT[15:0]							
Reset:	0	0	0	0	0	0	0	0

注意：

当自动温度补偿功能开启后，该寄存器的值表示在不同温度点的 U 通道温度自动补偿值。

AutoIlgain	<b>基地址： 0x40012000</b> <b>偏移地址： A4H</b>							
	Bit15...Bit0							
Read:	X							
Write:	DAT[15:0]							
Reset:	0	0	0	0	0	0	0	0

注意：

当自动温度补偿功能开启后，该寄存器的值表示在不同温度点的 I1 通道温度自动补偿值。

AutoI2gain	<b>基地址： 0x40012000</b> <b>偏移地址： A8H</b>							
	Bit15...Bit0							
Read:	X							
Write:	DAT[15:0]							
Reset:	0	0	0	0	0	0	0	0

注意：

当自动温度补偿功能开启后，该寄存器的值表示在不同温度点的 I2 通道温度自动补偿值。

## 21.3.2 校表参数寄存器

### 21.3.2.1 校表寄存器列表

EMU 模块校表寄存器基地址： 校表寄存器：0x40013800					
偏移地址	名称	读写方式	有效字长	复位值	功能描述
00H	<a href="#">EMUSR</a>	R/W	2	0x0000	EMU 状态标志寄存器
04H	<a href="#">EMUIE</a>	R/W	2	0x0000	EMU 中断使能寄存器
08H	<a href="#">EMUIF</a>	R/W	2	0x0000	EMU 中断标志寄存器
0CH	<a href="#">GP1</a>	R/W	2	0x0000	通道 1 的有功功率校正
10H	<a href="#">GQ1</a>	R/W	2	0x0000	通道 1 的无功功率校正
14H	<a href="#">GS1</a>	R/W	2	0x0000	通道 1 的视在功率校正
18H	<a href="#">GPhs1</a>	R/W	2	0x0000	通道 1 的相位校正
1CH	<a href="#">GP2</a>	R/W	2	0x0000	通道 2 的有功功率校正
20H	<a href="#">GQ2</a>	R/W	2	0x0000	通道 2 的无功功率校正
24H	<a href="#">GS2</a>	R/W	2	0x0000	通道 2 的视在功率校正
28H	<a href="#">GPhs2</a>	R/W	2	0x0000	通道 2 的相位校正
2CH	<a href="#">QPhsCal</a>	R/W	2	0x0000	无功相位补偿
30H	<a href="#">I2Gain</a>	R/W	2	0x0000	电流通道 2 增益补偿
34H	<a href="#">I1Off</a>	R/W	2	0x0000	电流通道 1 的直流偏置校正
38H	<a href="#">I2Off</a>	R/W	2	0x0000	电流通道 2 的直流偏置校正
3CH	<a href="#">UOff</a>	R/W	2	0x0000	电压通道的直流偏置校正
40H	<a href="#">PStart</a>	R/W	2	0x0040	起动功率设置，16 位无符号数，与有功功率寄存器 32 位值的高 24 位的绝对值相比，超过时，累加有功能量；
44H	<a href="#">QStart</a>	R/W	2	0x0080	起动功率设置，16 位无符号数，与无功功率寄存器 32 位值的高 24 位的绝对值相比，超过时，累加无功能量；
48H	<a href="#">SStart</a>	R/W	2	0x0090	起动功率设置，16 位无符号数，与视在功率寄存器 32 位值的高 24 位的绝对值相比，超过时，累加视在能量；
4CH	<a href="#">HFConst</a>	R/W	2	0x0080	输出脉冲频率设置
50H	<a href="#">ADCCFG</a>	R/W	2	0x0003	ADC 控制寄存器
54H	<a href="#">CHNLCR</a>	R/W	2	0x0607	通道控制寄存器
58H	<a href="#">EMCON</a>	R/W	2	0x1870	能量计量控制寄存器
5CH	<a href="#">PFCnt</a>	R/W	2	0x0000	快速有功脉冲计数
60H	<a href="#">QFCnt</a>	R/W	2	0x0000	快速无功脉冲计数
64H	<a href="#">SFCnt</a>	R/W	2	0x0000	快速视在脉冲计数

68H	<a href="#">ADCCON</a>	R/W	2	0x0000	ADC 通道增益选择（包括数字和模拟增益）
6CH	<a href="#">IPTAMP</a>	R/W	2	0x0020	窃电检测域值
70H	<a href="#">ICLK</a>	R/W	2	0x0010	窃电阈值设置
74H	<a href="#">EMUCTRL</a>	R/W	2	0x0382	EMU 控制寄存器
78H	<a href="#">P1OFFSET</a>	R/W	2	0x0000	通道 1 有功功率小信号偏置校正
7CH	<a href="#">P2OFFSET</a>	R/W	2	0x0000	通道 2 有功功率小信号偏置校正
80H	<a href="#">Q1OFFSET</a>	R/W	2	0x0000	通道 1 无功功率小信号偏置校正
84H	<a href="#">Q2OFFSET</a>	R/W	2	0x0000	通道 2 无功功率小信号偏置校正
88H	<a href="#">I1RMSOFFSET</a>	R/W	2	0x0000	电流通道 1 有效值小信号校正寄存器
8CH	<a href="#">I2RMSOFFSET</a>	R/W	2	0x0000	电流通道 2 有效值小信号校正寄存器
90H	<a href="#">URMSOFFSET</a>	R/W	2	0x0000	电压通道有效值小信号校正寄存器
94H	<a href="#">ROSICTRL</a>	R/W	2	0x4000	罗氏线圈使能控制位
98H	<a href="#">ANA_control</a>	R/W	2	0x27A6	模拟控制寄存器
9CH	<a href="#">UCONST</a>	R/W	2	0x0000	失压情况下参与计量的电压，断相窃电
A0H	<a href="#">LpIdleTime</a>	R/W	2	0x0000	低功耗计量模式下，分时模式空闲时间设置寄存器
A4H	<a href="#">USAGLVL</a>	R/W	2	0x0000	电压跌落检测阈值寄存器
A8H	<a href="#">USagCyc</a>	R/W	2	0x0020	电压跌落检测 ADC 半周波数设置寄存器
ACH	<a href="#">UOVLVL</a>	R/W	2	0x0000	电压峰值检测阈值寄存器
B0H	<a href="#">OvCyc</a>	R/W		0x0001	峰值检测 ADC 半周波数设置寄存器
B4H	<a href="#">IOVLVL</a>	R/W	2	0x0000	电流峰值阈值寄存器
B8H	<a href="#">ZXILVL</a>	R/W		0x0000	电流过零阈值寄存器
BCH	<a href="#">PDataCpH</a>	R/W	2	0x0000	常数有功功率寄存器高 16bit，用户需要写完高低 16bit，再在 LoadDataCp 写 0x00BC 后，才会进行常数计量。
C0H	<a href="#">PDataCpL</a>	R/W	2	0x0000	常数有功功率寄存器低 16bit
C4H	<a href="#">QDataCpH</a>	R/W	2	0x0000	常数无功功率寄存器高 16bit，用户需要写完高低 16bit，再在 LoadDataCp 写 0x00BC 后，才会进行常数计量。
C8H	<a href="#">QDataCpL</a>	R/W	2	0x0000	常数无功功率寄存器低 16bit
CCH	<a href="#">SDataCpH</a>	R/W	2	0x0000	常数视在功率寄存器高 16bit，用户需要写完高低 16bit，再在 LoadDataCp 写 0x00BC 后，才会进行常数计量。
D0H	<a href="#">SDataCpL</a>	R/W	2	0x0000	常数功率寄存器低 16bit
D4H	<a href="#">FilterCtrl</a>	R/W	2	0x02F7	滤波器系数选择控制
D8H	<a href="#">TUgain</a>	R/W	2	0x0000	电压通道手动温度补偿系数
DCH	<a href="#">TI1gain</a>	R/W	2	0x0000	电流 I1 通道手动温度补偿系数
E0H	<a href="#">TI2gain</a>	R/W	2	0x0000	电流 I2 通道手动温度补偿系数
E4H	<a href="#">UTCcoeffA</a>	R/W	2	0x0000	电压通道自动温补 Ugain 的二次系数

E8H	<a href="#">UTCcoeffB</a>	R/W	2	0x0000	电压通道自动温补 Ugain 的一次系数
ECH	<a href="#">UTCcoeffC</a>	R/W	2	0x0000	电压通道自动温补 Ugain 的常数项
F0H	<a href="#">I1TCcoeffA</a>	R/W	2	0x0000	电流通道 1 自动温补 I1gain 的二次系数
F4H	<a href="#">I1TCcoeffB</a>	R/W	2	0x0000	电流通道 1 自动温补 I1gain 的一次系数
F8H	<a href="#">I1TCcoeffC</a>	R/W	2	0x0000	电流通道 1 自动温补 I1gain 的常数项
FCH	<a href="#">I2TCcoeffA</a>	R/W	2	0x0000	电流通道 2 自动温补 I2gain 的二次系数
100H	<a href="#">I2TCcoeffB</a>	R/W	2	0x0000	电流通道 2 自动温补 I2gain 的一次系数
104H	<a href="#">I2TCcoeffC</a>	R/W	2	0x0000	电流通道 2 自动温补 I2gain 的常数项
108H	<a href="#">LoadDataCp</a>	R/W	2	0x0000	当用户向该寄存器写 0x00BC 时, 常数功率值 P/Q/SDATACP load 进入累加源头;
10CH	Reserved	R/W	2	0x0000	保留寄存器
110H	Reserved	R/W	2	0x0000	保留寄存器
114H	<a href="#">SRSTREG</a>	R/W	1	0x00	软件复位寄存器 (写 0x55 用于复位校表寄存器, 写 0xAA 用于复位 EMU 模块)

### 21.3.2.2 校表寄存器说明

EMUSR (EMU 状态寄存器)		基地址: 0x40013800 偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:					SFOF	PEOF	QEOF	Chanelstatus
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TAMP	I2GTI1	NoQLd2	NoPLd2	NoQLd1	NoPLd1	REVQ	REVP
Write:		X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SFOF	视在电能寄存器溢出标志, 写 0 清零
PEOF	有功电能寄存器溢出标志, 写 0 清零
QEOF	无功电能寄存器溢出标志, 写 0 清零
Chanelstatus	计量通道状态标志。(0: 指示使用电流通道 1 计量 1: 指示使用电流通道 2 计量)
TAMP	窃电标志

	<p>0: 表示未发生窃电, I1Rms 与 I2Rms 或者 Power1 与 Power2 相差不到设定的 IChk 范围。</p> <p>1: 表示发生窃电,</p> <p>当TampSel =0, 表示: <math>I1Rms &gt; I2Rms * (1 + IChk)</math> 或者 <math>I2Rms &gt; I1Rms * (1 + IChk)</math></p> <p>当TampSel =1, 表示: <math>Power1 &gt; Power2 * (1 + IChk)</math> 或者 <math>Power2 &gt; Power1 * (1 + IChk)</math></p>
I2GTI1	<p><b>通道2电流或功率大于通道1电流或功率标志</b></p> <p>0: I2Rms &lt; I1Rms 或 Power2 &lt; Power1</p> <p>    当TampSel =0, 表示: I2Rms &lt; I1Rms</p> <p>1: I2Rms &gt; I1Rms 或 Power2 &gt; Power1</p> <p>    当TampSel =0, 表示: I2Rms &gt; I1Rms</p> <p>    当TampSel =1, 表示: Power2 &gt; Power1</p> <p>当TampSel =1, 表示: Power2 &lt; Power1</p>
NoQLd2	当无功功率 2 小于起动功率时, NoQLd2 被置为 1; 当无功功率大于/等于起动功率时 NoQLd2 清为 0
NoPLd2	当有功功率 2 小于起动功率时, NoPLd2 被置为 1; 当有功功率大于/等于起动功率时 NoPLd2 清为 0
NoQLd1	当无功功率 1 小于起动功率时, NoQLd1 被置为 1; 当无功功率大于/等于起动功率时 NoQLd1 清为 0。
NoPLd1	当有功功率 1 小于起动功率时, NoPLd1 被置为 1; 当有功功率大于/等于起动功率时 NoPLd1 清为 0。
REVQ	反向无功功率指示标识信号, 当检测到负无功功率时, 该信号为 1。当再次检测到正无功功率时, 该信号为 0。在 QF 发脉冲时更新该值。
REVP	反向有功功率指示标识信号, 当检测到负有功功率时, 该信号为 1。当再次检测到正有功功率时, 该信号为 0。在 PF 发脉冲时更新该值。

注: 公用标识 (REVP、REVQ、SEOF、PEOF、QEOF) 表示当前有效计量通道的状态。

EMUIE (EMU 状态寄存器)		基地址: 0x40013800 偏移地址: 04H						
	Bit15	14	13	12	11	10	9	Bit8
Read:								
Write:	X	TAMPIE	DCavreageUpdateIE	SlowRmsUpdateIE	FastRmsUpdateIE	ZXI2IE	ZXI1IE	I2OVIE
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PFIE	QFIE	SFIE	SPLIE	ZXIE	I1OVIE	UOVIE	SAGIE
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位名称	描述
TAMPIE	窃电发生中断使能 (0: 禁止 1: 使能)
DCavreageUpdateIE	直流平均值更新中断使能 (0: 禁止 1: 使能)

SlowRmsUpdateIE	慢速电流电压有效值更新时的中断使能 (0: 禁止 1: 使能)
FastRmsUpdateIE	快速电流电压有效值更新时的中断使能 (0: 禁止 1: 使能)
ZXI2IE	电流通道 2 发生用户指定的过零中断使能 (0: 禁止 1: 使能)
ZXI1IE	电流通道 1 发生用户指定的过零中断使能 (0: 禁止 1: 使能)
I2OVIE	电流 I2 通道峰值超过阈值的中断使能 (0: 禁止 1: 使能)
PFIE	PF 发脉冲时的中断使能 (0: 禁止 1: 使能)
QFIE	QF 发脉冲时的中断使能 (0: 禁止 1: 使能)
SFIE	SF 发脉冲时的中断使能 (0: 禁止 1: 使能)
SPLIE	波形寄存器更新时的中断使能 (0: 禁止 1: 使能)
ZXIE	电压发生用户指定的过零方式时的中断使能 (0: 禁止 1: 使能)
I1OVIE	电流 I1 通道峰值超过阈值的中断使能 (0: 禁止 1: 使能)
UOVIE	电压峰值超过阈值的中断使能 (0: 禁止 1: 使能)
SAGIE	电压跌落低于设定阈值并时间超出用户指定半周波数时的中断使能 (0: 禁止 1: 使能)

EMUIF (EMU 状态寄存器)		基地址: 0x40013800 偏移地址: 08H						
	Bit15	14	13	12	11	10	9	Bit8
Read:			DCavrea	SlowR	FastRms	ZXI2IF	ZXI1IF	I2OVIF
Write:	X	TAMPIF	geUpdate IF	msUpdat eIF	UpdateIF			
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PFIF	QFIF	SFIF	SPLIF	ZXIF	I1OVIF	UOVIF	SAGIF
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位名称	描述
TAMPIF	当发生窃电时的中断标志, 读后清零;
DCavreageUpdateIF	直流平均值更新时的中断标志, 读后清零;
SlowRmsUpdateIF	慢速电流电压有效值更新时的中断标志, 读后清零;
FastRmsUpdateIF	快速电流电压有效值更新时的中断标志, 读后清零;
ZXI2IF	电流通道 2 发生用户指定的过零中断标志, 读后清零;
ZXI1IF	电流通道 1 发生用户指定的过零中断标志, 读后清零;
I2OVIF	电流 I2 通道峰值超过阈值的中断标志, 读后清零;
PFIF	PF 发脉冲时的中断标志, 读后清零;
QFIF	QF 发脉冲时的中断标志, 读后清零;
SFIF	SF 发脉冲时的中断标志, 读后清零;
SPLIF	波形寄存器更新时的中断标志, 读后清零;
ZXIF	电压发生用户指定的过零方式时的中断标志, 读后清零;
I1OVIF	电流 I1 通道峰值超过阈值的中断标志, 读后清零;

UOVIF	电压峰值超过阈值的中断标志，读后清零；
SAGIF	电压跌落低于设定阈值并时间超出用户指定半周波数时的中断标志，读后清零；

GP1	<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 0CH							
	Bit15...Bit0							
<b>Read:</b>	GP1[15:0]							
<b>Write:</b>	GP1[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

该寄存器为 16 位有符号数，最高位为符号位。

功率因数为 1 的情况下，用户在校表过程中测得的误差为: Err%

$$Pgain = -Err\% / (1 + Err\%)$$

如果 Pgain 为正数，则 GP1 的写入值为:  $Pgain * 32768$

如果 Pgain 为负数，则 GP1 的写入值为:  $65536 + Pgain * 32768$

GQ1	<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 10H							
	Bit15...Bit0							
<b>Read:</b>	GQ1[15:0]							
<b>Write:</b>	GQ1[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

16 位有符号数，最高位为符号位。

用户校表过程中，一般 GQ1 写入值与 GP1 计算后的写入值相同。

GS1	<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 14H							
	Bit15...Bit0							
<b>Read:</b>	GS1[15:0]							
<b>Write:</b>	GS1[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

16 位有符号数，最高位为符号位。

用户校表过程中，GS1 写入值与 GP1 计算后的写入值相同。

GPHS1	<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 18H							
-------	--	--	--	--	--	--	--	--



	Bit15…Bit0							
Read:	GPHS1[15:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

注意:

PQ 方式校相位计算公式如下:

用户在信号输入为阻性的时候通过 GP 寄存器将输出误差校正到 0 附近

将信号输入调整为 0.5L, 此时观察误差为 Err%

如果 Err 为负数:

$$Gphs1 = -Err\% * 32768 / 1.732$$

如果 Err 为正数:

$$Gphs1 = 65536 - Err\% * 32768 / 1.732$$

GP2	基地址: 0x40013800 偏移地址: 1CH							
	Bit15…Bit0							
Read:	GP2[15:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

注意:

与 GP1 的计算公式相同。

GQ2	基地址: 0x40013800 偏移地址: 20H							
	Bit15…Bit0							
Read:	GQ2[15:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

注意:

用户在校正过程中, 一般与 GP2 的写入值相同。

GS2	基地址: 0x40013800 偏移地址: 24H							
	Bit15…Bit0							
Read:	GS2[15:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

注意:

用户在校正过程中, 一般与 GP2 的写入值相同。

GPHS2		<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 28H						
		Bit15...Bit0						
<b>Read:</b>	GPHS2[15:0]							
<b>Write:</b>	GPHS2[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

0CH~28H 寄存器都是二进制补码格式，最高位为符号位。具体的操作方式可参考校表过程。

注意:

PQ 方式校相位计算公式如下:

用户在信号输入为阻性的时候通过 GP 寄存器将输出误差校正到 0 附近

将信号输入调整为 0.5L，此时观察误差为 Err%

如果 Err 为负数:

$$Gphs2 = -Err\% * 32768 / 1.732$$

如果 Err 为正数:

$$Gphs2 = 65536 - Err\% * 32768 / 1.732$$

QPHSCal		<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 2CH						
		Bit15...Bit0						
<b>Read:</b>	QPHSCal[15:0]							
<b>Write:</b>	QPHSCal[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

无功相位补偿寄存器也采用二进制补码形式，最高位为符号位。

QphsCal 默认为 0x0000，Femu 为 819.2KHz，对应到 50hz 输入信号 OSR128 (EMCON bit7) 选择 64 时，QphsCal 为 0x005E，可以实现准确的 90 移相。对应到 60Hz 输入信号 OSR128 选择 64 时，QphsCal 为 0x44。

I2Gain		<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 30H						
		Bit15...Bit0						
<b>Read:</b>	I2Gain[15:0]							
<b>Write:</b>	I2Gain[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

电流通道 2 增益校正,采用二进制补码形式，最高位为符号位。定义参见校表过程。

I10ff		<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 34H						
		Bit15...Bit0						
<b>Read:</b>	I10ff[15:0]							
<b>Write:</b>	I10ff[15:0]							

<b>Reset:</b>	0	0	0	0	0	0	0	0
---------------	---	---	---	---	---	---	---	---

注意:

在测量直流信号的情况下使用,先关闭高通,当输入通道信号为0的时候,连续几次读寄存器 00H 的值平均后,得到 I1Off 寄存器的值,用户得到后将这个值存储起来,每次重新上电将之前存储的 I1Off 值重新写入寄存器。

该寄存器与 ADC 输出数据的 bit[19]~bit[4]位对齐。

该寄存器主要目的是当用户期望测试直流信号时,将芯片内部高通关闭,I1/I2/U 必须一起关高通,否则会引入相位误差。通过这个寄存器校正外部输入信号为0时的 ADC 零漂,一般来说用户测试交流信号不需要配置该寄存器。

<b>I20ff</b>		<b>基地址: 0x40013800</b>						
		<b>偏移地址: 38H</b>						
		Bit15...Bit0						
<b>Read:</b>	I20ff[15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

在测量直流信号的情况下使用,先关闭高通,当输入通道信号为0的时候,连续几次读寄存器 01H 的值平均后,得到 I2Off 寄存器的值,用户得到后将这个值存储起来,每次重新上电将之前存储的 I2Off 值重新写入寄存器。

该寄存器与 ADC 输出数据的 bit[19]~bit[4]位对齐。

<b>U0ff</b>		<b>基地址: 0x40013800</b>						
		<b>偏移地址: 3CH</b>						
		Bit15...Bit0						
<b>Read:</b>	U0ff[15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

在测量直流信号的情况下使用,先关闭高通,当输入通道信号为0的时候,连续几次读寄存器 02H 的值平均后,得到 U0ff 寄存器的值,用户得到后将这个值存储起来,每次重新上电将之前存储的 U0ff 值重新写入寄存器。

该寄存器与 ADC 输出数据的 bit[19]~bit[4]位对齐。

通道直流偏置校正只在高通环节关闭后使用。I1/I2/U 必需一起关高通,否则会引入相位误差。

<b>PStart</b>		<b>基地址: 0x40013800</b>						
		<b>偏移地址: 40H</b>						
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12...7</b>	<b>6</b>	<b>5...2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	PS15	PS 14	PS 13	PS 12...PS 7	PS 6	PS 5...PS 2	PS 1	PS 0
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	1	0	0	0

Default 值: 0x0040

注意:

PStart 是 16 位无符号数, 做比较时, 将其与 PowerP 绝对值的 bit8—bit31 进行比较, 以作起动判断。  
|P| 小于 PStart 时, PF 不输出脉冲。同时将 REVP 反向标志清 0。

应用方式:

- 1, 校表结束后, 输入 Ib, Un。
- 2, 读出 PowerP 的值为 32bit, 取其高 24bit 为 x1,  
如果 x1 为正数,  $x2 = x1$ ;  
如果 x1 为负数, 取其原码为 x2;
- 3, 设写入 PStart 的值为 Y, 假如要求 0.4%Ib 电表能够启动, 则:  
 $Y = x2 * 0.2\%$

QStart		<b>基地址: 0x40013800</b> <b>偏移地址: 44H</b>						
	Bit15	14	13	12 ... 8	7	6...2	1	Bit0
Read:	QS15	QS 14	QS 13	QS 12...QS 8	QS 7	QS 6...QS 2	QS 1	QS 0
Write:								
Reset:	0	0	0	0	1	0	0	0

Default 值: 0x0080

注意:

QStart 是 16 位无符号数, 做比较时, 将其与 PowerQ 绝对值的 bit8—bit31 进行比较, 以作起动判断。  
|Q| 小于 QStart 时, QF 不输出脉冲。同时将 REVQ 反向标志清 0。

应用方式:

- 1, 校表结束后, 输入 Ib, Un。
- 2, 读出 PowerQ 的值为 32bit, 取其高 24bit 为 x1,  
如果 x1 为正数,  $x2 = x1$ ;  
如果 x1 为负数, 取其原码为 x2;
- 3, 设写入 QStart 的值为 Y, 假如要求 0.4%Ib 电表能够启动, 则:  
 $Y = x2 * 0.2\%$

SStart		<b>基地址: 0x40013800</b> <b>偏移地址: 48H</b>							
	Bit15	14	13	12 ... 8	7	6...5	4	3...1	Bit0
Read:	SS15	SS 14	SS 13	SS 12...SS 8	SS 7	SS 6...SS 5	SS 4	SS3...SS 1	SS 0
Write:									
Reset:	0	0	0	0	1	0	1	0	0

Default 值: 0x0090

注意:

SStart 是 16 位无符号数, 做比较时, 将其与 PowerS 值的 bit8—bit31 进行比较, 以作起动判断。  
|S| 小于 SStart 时, SF 不输出脉冲。

应用方式:

- 1, 校表结束后, 输入 Ib, Un。
- 2, 读出 PowerS 的值为 32bit, 取其高 24bit 为 x2,

3, 设写入 SStart 的值为 Y, 假如要求 0.4%Ib 电表能够启动, 则:

$$Y = x^2 * 0.2 \%$$

HFConst		<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 4CH						
	Bit15	Bit14...Bit0						
<b>Read:</b>	X	HFConst[14:0]						
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	1	0	0	0

注意:

HFConst 是 15 位无符号数, 做比较时, 将其作为低 15 位与快速脉冲计数寄存器 0x6FH~0x71H 寄存器值的绝对值做比较, 如果大于等于 HFConst 的值, 那么就会有对应的 PF/QF/SF 脉冲输出。

Default 值: 0x0080

ADC Config Register (ADCCFG)		<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 50H						
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	PQSStart SourceSe	PQSStart SourceSe	Vref_En	Add _CIADD	WaveSel	TampSou rceSel	IPtampS ource_Se	CHNFix
<b>Write:</b>	11	10					1	
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	IPtamp_ Sel	TampSel	FLTON	CHNSEL	CIADD	SPL2	SPL1	SPL0
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	1	1

位名称	描述																				
PQSStartSourceSel[1:0]	<b>PQS 起动潜动比较源头控制位:</b> <table border="1"> <tr> <td>PQSStartSourceSel1</td> <td>PQSStartSourceSel0</td> <td>PQS 起动潜动源头</td> </tr> <tr> <td>0</td> <td>0</td> <td>快速功率</td> </tr> <tr> <td>0</td> <td>1</td> <td>慢速功率</td> </tr> <tr> <td>1</td> <td>X</td> <td>瞬时功率</td> </tr> </table>	PQSStartSourceSel1	PQSStartSourceSel0	PQS 起动潜动源头	0	0	快速功率	0	1	慢速功率	1	X	瞬时功率								
PQSStartSourceSel1	PQSStartSourceSel0	PQS 起动潜动源头																			
0	0	快速功率																			
0	1	慢速功率																			
1	X	瞬时功率																			
Vref_En	<b>Vref 使能控制位:</b> 与 LDO_EMU_En、EMU_En、ADC_On 共同起作用。详细见下表 <table border="1"> <tr> <td>LDO_EMU_En</td> <td>EMU_en</td> <td>Vref_En</td> <td>Adc_on</td> <td>Vref 实际输出</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>0</td> </tr> <tr> <td>X</td> <td>0</td> <td>X</td> <td>X</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>	LDO_EMU_En	EMU_en	Vref_En	Adc_on	Vref 实际输出	0	X	X	X	0	X	0	X	X	0	1	1	0	0	0
LDO_EMU_En	EMU_en	Vref_En	Adc_on	Vref 实际输出																	
0	X	X	X	0																	
X	0	X	X	0																	
1	1	0	0	0																	

	1	1	1	0	1
	1	1	1	1	1
Add_CIADD	<b>单相三线模式下累加模式控制位</b> = 0, 绝对值累加方式 = 1, 代数和累加方式				
WaveSel	<b>ADC 波形采样数据的源头选择</b> = 0, 选择 ADC 波形数据为未经高通的原始数据 = 1, 选择 ADC 波形数据为经过高通的数据				
TampSourceSel	<b>自动防窃电通道间比较源头选择:</b> =0, 选择比较源头为慢速有效值或慢速功率值 =1, 选择比较源头为快速有效值或快速功率值				
IPtampSource_Sel	<b>自动防窃电阈值判断源头的快速/慢速选择位</b> =0, 选择阈值源头为慢速有效值或慢速功率值 =1, 选择阈值源头为快速有效值或快速功率值				
CHNFix	<b>自动防窃电小信号通道切换配置位</b> =0, 在两路电流通道都降低到 IPTAMP 以下的时候, 选择固定第一电流通道计量 =1, 在两路电流通道都降低到 IPTAMP 以下的时候, 不切换通道, 保持前一个通道计量				
IPtamp_Sel	<b>自动防窃电阈值判定依据</b> =0, 决定 IPtamp 寄存器选择有效值做为防窃电阈值 =1, 决定 IPtamp 寄存器选择有功功率做为防窃电阈值				
TampSel	<b>防窃电比较方式选择</b> =0, 选择电流有效值做防窃电比较 =1, 选择有功功率做防窃电				
FLTON	自动防窃电模块开关 (0: 自动防窃电关闭 1: 自动防窃电开启), 详细见后表				
CHNSEL	选择通道计量 (0: 选择通道 1 计量 1: 选择通道 2 计量), 详细见后表				
CIADD	单相三线累加模式选择 (0: 单通道方式 1: 电流累加和模式)				
SPL[2:0]	波形采样频率选择控制寄存器, 详细见下表				

注意:

电流相加模式下, 每路通道用各自的校表参数数据。在 FLTON=1 时, 即开启自动防窃电模块时, CIADD 和 CHNSEL 可以读写, 但是无效; 只有 FLTON=0 时 CIADD 和 CHNSEL 可有效读写。

输入信号			输出信号	
FLTON	CIADD	CHNSEL	Chanelstatus	能量累加
1	X	X	指示自动防窃电的通道选择结果	根据 Chanelstatus 来决定采用哪一路功率参与计量

0	0	0	0	选择通道 1 参与计量 (default)
0	0	1	1	选择通道 2 参与计量
0	1	x	0	单相三线模式

### 自动防窃电判断依据设置说明

TampSel	IPtamp_Sel	TampSourceSel	IPtampSource_Sel	自动防窃电的判断依据
0	0	0	0	IPTAMP(6CH)基于慢速有效值寄存器比较是否超过窃电阈值，ICLK(70H)比较两路慢速有效值差值是否超过设定的百分比；
0	0	0	1	IPTAMP(6CH)基于快速有效值寄存器比较是否超过窃电阈值，ICLK(70H)比较两路慢速有效值差值是否超过设定的百分比；
0	0	1	0	IPTAMP(6CH) 基于慢速有效值，ICLK(70H)基于快速有效值
0	0	1	1	IPTAMP(6CH) 基于快速有效值，ICLK(70H)基于快速有效值
0	1	0	0	IPTAMP(6CH) 基于慢速功率，ICLK(70H)基于慢速有效值
0	1	0	1	IPTAMP(6CH) 基于快速功率，ICLK(70H)基于慢速有效值
0	1	1	0	IPTAMP(6CH) 基于慢速功率，ICLK(70H)基于快速有效值
0	1	1	1	IPTAMP(6CH) 基于快速功率，ICLK(70H)基于快速有效值
1	0	0	0	IPTAMP(6CH) 基于慢速有效值，ICLK(70H)基于慢速功率
1	0	0	1	IPTAMP(6CH) 基于快速有效值，ICLK(70H)基于慢速功率
1	0	1	0	IPTAMP(6CH) 基于慢速有效值，

				ICCHK(70H)基于快速功率
1	0	1	1	IPTAMP(6CH) 基于快速有效值， ICCHK(70H)基于快速功率
1	1	0	0	IPTAMP(6CH) 基于慢速功率， ICCHK(70H)基于慢速功率
1	1	0	1	IPTAMP(6CH) 基于快速功率， ICCHK(70H)基于慢速功率
1	1	1	0	IPTAMP(6CH) 基于慢速功率， ICCHK(70H)基于快速功率
1	1	1	1	IPTAMP(6CH) 基于快速功率， ICCHK(70H)基于快速功率

SPL[2:0]: 波形采样中断频率选择, 当 Femu=819.2K 时, 选择的频率如下:

SPL2	SPL1	SPL0	波形采样频率
0	0	0	1.6k Hz (femu/512)
0	0	1	3.2k Hz (femu/256)
0	1	0	6.4k Hz (femu/128)
0	1	1	12.8k Hz (fadc/64)
1	X	x	12.8k Hz (fadc/64)

注意:

当 femu=409.6KHz 时, 选择的波形采样频率与上表相比等比例下调即可。

Current Channel Control Register (CHNLCR 通道控制寄存器)			基地址: 0x40013800 偏移地址: 54H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PDatacp En	QDatacpE n	SDatacpE n	Sag_E n	Ovl_ En	RmsLpfE n	PQLpfEn	Add_Data CP
Write:								
Reset:	0	0	0	0	0	1	1	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	LPMode	POS	LP FreqSel	Uconst En	PSSel	HPFONI2	HPFONI1	HPFONU



Write:								
Reset:	0	0		0	0	1	1	1

位	功能描述
PDatacpEn	<b>有功功率常数计量使能位</b> 0: 关闭常数计量 (default) 1: 使能常数计量
QDatacpEn	<b>无功功率常数计量使能位</b> 0: 关闭常数计量 (default) 1: 使能常数计量
SDatacpEn	<b>视在功率常数计量使能位</b> 0: 关闭常数计量 (default) 1: 使能常数计量
Sag_En	<b>电压 SAG 功能使能控制:</b> 0: 关闭电压 SAG 功能; 1: 使能电压 SAG 功能;
Ovl_En	<b>过压过流功能使能控制位:</b> 0: 关闭过压过流检测功能; 1: 使能过压过流检测功能;
RmsLpfEn	<b>RmsLpf 使能控制位:</b> =0, 关闭有效值通道的低通滤波器 =1, 开启有效值通道的低通滤波器 (default)
PQLpfEn	<b>PQLpf 使能控制位:</b> =0, 关闭功率通道的低通滤波器 =1, 开启功率通道的低通滤波器 (default)
Add_DataCP	<b>常数计量下, 能量累加速度选择位:</b> = 0, 选择累加速度为 32k 速度 = 1, 选择累加速度为 BS 位流速度
LPMode	<b>低功耗模式使能控制位:</b> =0, EMU 为正常计量模式 (default) =1, EMU 时钟频率切换到低频, 进入低功耗模式
POS	<b>脉冲有效电平选择位</b> 0: 表示PF/QF/SF为高电平有效 1: 表示PF/QF/SF为低电平有效
LP FreqSel	<b>低功耗计量模式下时钟选择位</b> = 0: 表示选择32K时钟 (OSC时钟, PLL Off) = 1: 表示选择204.8K时钟 (PLL On)
UconstEn	<b>低功耗计量模式下计算视在功率的电压源头选择位</b> 0: 使用电压通道有效值和电流通道有效值计算视在功率 1: 使用Uconst寄存器和电流通道有效值计算视在功率
PSSel	<b>脉冲输出类型选择位</b> 0: PF输出有功脉冲, SF输出视在脉冲 1: PF 输出视在脉冲, SF输出有功脉冲

	此功能适用于低功耗计量模式下，用视在功率替代有功功率，在同一脉冲输出口输出有功和视在脉冲的情况。
HPFONI2	电流通道2高通滤波器开关（0：关闭 1：开）
HPFONI1	电流通道1高通滤波器开关（0：关闭 1：开）
HPFONU	电压通道高通滤波器开关（0：关闭 1：开）

Energy Measure Control (EMCON 能量累加控制寄存器)			基地址： 0x40013800 偏移地址： 58H					
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	CF1CFG	CF1CFG	CF2CFG	CF2CFG	CF3CFG	CF3CFG	AverPow	AverPow
<b>Write:</b>	1	0	1	0	1	0	erCal	erSource
<b>Reset:</b>	0	0	0	1	1	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	OSR128	SRun	QRun	PRun	QMOD1	QMOD0	PMOD1	PMOD0
<b>Write:</b>								
<b>Reset:</b>	<b>0</b>	1	1	1	0	0	0	0

位	功能描述		
CF1CFG[1:0]	<b>CF1 口输出配置控制位:</b>		
	CF1CFG1	CF1CFG 0	CF1
	0	0	P (default)
	0	1	Q
	1	0	S
CF2CFG[1:0]	<b>CF2 口输出配置控制位:</b>		
	CF2CFG1	CF2CFG 0	CF2
	0	0	P
	0	1	Q (default)
	1	0	S
CF3CFG[1:0]	<b>CF3 口输出配置控制位:</b>		
	CF3CFG1	CF3CFG 0	CF3
	0	0	P
	0	1	Q
	1	0	S (default)
AverPowerCal	<b>能量累加源头选择位</b>		
	=0, 选择瞬时功率作为能量累加源头		

	=1, 选择平均功率作为能量累加源头			
AverPowerSource	当使能平均功率累加时 (AverPowerCal =1 时), 平均功率的源头选择: =0, 选择慢速平均功率作为平均功率累加源头 =1, 选择快速平均功率作为平均功率累加源头			
OSR128	正常计量模式下过采样率 OSR 选择位: (仅对正常计量模式起作用, 低功耗计量模式下无用) =0, OSR =64 =1, OSR =128			
SRun	视在能量累加使能位 0: 停止视在计量 1: 允许视在计量			
QRun	无功能量累加使能位 0: 停止无功计量 1: 允许无功计量			
PRun	有功能量累加使能位 0: 停止有功计量 1: 允许有功计量			
QMOD[1: 0]	无功能量累加模式选择位			
	QMOD1	QMOD0	累加功率Qm	无功能量累加方式选择
	0	0	$Q_m = \text{DataQ}$	脉冲能量按代数和方式对无功功率累加
	0	1	$\text{DataQ} \geq 0, Q_m = \text{DataQ};$ $\text{DataQ} < 0, Q_m = 0$	脉冲能量只累加正向无功功率, 不累加负向无功功率
	1	0	$Q_m =  \text{DataQ} $	脉冲能量时按绝对值方式对无功功率累加
PMOD[1: 0]	有功能量累加模式选择位			
	PMOD1	PMOD0	累加功率Pm	有功能量累加方式选择
	0	0	$P_m = \text{DataP}$	脉冲能量按代数和方式对有功功率累加
	0	1	$\text{DataP} \geq 0, P_m = \text{DataP};$ $\text{DataP} < 0, P_m = 0$	脉冲能量只累加正向有功功率, 不累加负向有功功率
	1	0	$P_m =  \text{DataP} $	脉冲能量按绝对值方式对有功功率累加
1	1	$P_m = \text{DataP}$	脉冲能量按代数和方式对有功功率累加	

PFCnt	基地址:	0x40013800
	偏移地址:	5CH
Bit15...Bit0		

Read:	PFCNT[15:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

<b>QFCnt</b>		<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 60H						
		Bit15···Bit0						
Read:	QFCNT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

<b>SFCnt</b>		<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 64H						
		Bit15···Bit0						
Read:	SFCNT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

注意:

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

当快速脉冲计数寄存器 PFCnt/QFCnt/SFCnt 计数的值大于等于 HFconst 时, 相应的 PF/QF/SF 会有脉冲溢出, 能量寄存器的值会相应的加 1。

<b>ADC Channel Gain (ADCCON</b>		<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 68H						
<b>ADC 通道增益控制寄存器)</b>								
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
Read:	0	0	DGI3	DGI2	DGI1	DGI0	DGU1	DGU0
Write:	X	X						
Reset:	0	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
Read:	CFP1	CFP0	PGA3	PGA2	PGA1	PGA0	UPGA1	UPGA0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述					
DGI[3:0]	电流通道数字增益选择位					
	DGI1	DGI0	I1 数字增益	DGI3	DGI2	I2 数字增益
	0	0	DG=1	0	0	DG=1

	0	1	DG=2	0	1	DG=2
	1	0	DG=4	1	0	DG=4
	1	1	DG=8	1	1	DG=8
DGU[1:0]	电压通道数字增益选择位					
	DGU 1		DGU 0		电压通道数字增益	
	0		0		DG=1	
	0		1		DG=2	
	1		0		DG=4	
1		1		DG=8		
CFP[1:0]	脉宽选择位 即脉冲输出的t4参数，见PF/QF/SF时序特性，					
	CFP1	CFP0	脉宽 (femu=819.2KHz)			
	0	0	80ms			
	0	1	40ms			
	1	0	20ms			
1	1	10ms				
PGA[3: 0]	电流通道模拟增益选择位					
	PGA1	PGA0	电流通道1	PGA3	PGA2	电流通道2
	0	0	PGA=1	0	0	PGA=1
	0	1	PGA=2	0	1	PGA=2
	1	0	PGA=8	1	0	PGA=8
1	1	PGA=16	1	1	PGA=16	
UPGA[1:0]	电压通道模拟增益选择位					
	UPGA1		UPGA0		电压通道增益	
	0		0		PGA=1	
	0		1		PGA=2	
	1		0		PGA=4	
1		1		PGA=8		

<b>IPTAMP</b>		基地址: 0x40013800						
		偏移地址: 6CH						
		Bit15...Bit0						
Read:	IPTAMP[15:0]							
Write:								
Reset:	0	0	0	0	0	0	1	0

注意：该寄存器默认值为 0x0020。

格式同电流有效值寄存器或功率寄存器，IPTAMP[15:0]是中间 16 位的电流有效值寄存器或功率寄存器。

当选择窃电阈值为功率时：对应功率寄存器的 **Bit27—Bit12**

当选择窃电阈值为有效值时：对应有效值寄存器的 **Bit19—Bit4**

自动防窃电处理模块开启时：

若选择使用快速或慢速电流有效值做为防窃电阈值的判断，当通道 1 和 2 的快速或慢速电流有效值都

低于 IPTAMP 时，选择通道 1 作为有效输入或者保持之前的计量通道，TAMP、I2GTI1 均为 0。

若选择使用快速或慢速功率 P 的绝对值做为防窃电阈值的判断，当快速或慢速 PowerP1 和 PowerP2 都低于 IPTAMP 时，选择通道 1 做为有效输入或者保持之前的计量通道，TAMP、I2GTI1 均为 0。

IChk		基地址: 0x40013800						
		偏移地址: 70H						
		Bit7...Bit0						
Read:	IChk[7:0]							
Write:								
Reset:	0	0	0	1	0	0	0	0

注意:

通道间窃电相差百分比阈值寄存器采用二进制补码形式，表示范围(0,+1)。

$$ICKK=ICK7*2^{(-1)} + ICK6*2^{(-2)} + ICK5*2^{(-3)} + \dots + ICK1*2^{(-7)} + ICK0*2^{(-8)}$$

当 $|I2Rms-I1Rms|/I1Rms$  或者 $|PowerP2-PowerP1|/PowerP1$  大于 ICHK 时，置位窃电标志。

当且仅当 Check Register 的某一个 Bit 为 1 时，对应的阈值如下表:

Bit7	0.5
Bit6	0.25
Bit5	0.125
Bit4	0.0625
Bit3	0.03125
Bit2	0.015625
Bit1	0.007813
Bit0	0.003906

例如: Check Register=0x1A 时，窃电阈值为 0.0625+0.03125+0.007813=10.1563%

默认为: 0.0625 也即 6.25%。

开启自动防窃电后，当选择快速或慢速电流有效值作为防窃电比较的源头时，快速或慢速电流 1 和电流 2 两者之差百分比 ( $|I2Rms-I1Rms|/I1Rms$ ) 超过设定值，则自动选择大的电流值参与功率计量，同时 TAMP=1。如果电流 2 大于电流 1，则将标志位 I2GTI1 置为 1，否则标志位 I2GTI1 为 0。

当选择快速或慢速功率作为防窃电比较的源头时，快速或慢速功率 PowerP1 和功率 PowerP2 两者之差百分比 ( $|PowerP2-PowerP1|/PowerP1$ ) 超过设定值，则自动选择大的功率值参与功率计量，同时 TAMP=1。

EMU and adc control (EMUCTRL EMU 控制寄存器)		基地址: 0x40013800						
		偏移地址: 74H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	EMU_ClkCtrl	DC	DC
Write:							PRFCFG1	PRFCFG0
Reset:	0	0	0	0	0	0	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	SlowPR	Adc_ilo	Adc_i2o	Adc_uon	Zxd1	Zxd0	PRFCFG1	PRFCFG0
Write:	MS	n	n					
Reset:	1	0	0	0	0	0	1	0

<b>位</b>	<b>功能描述</b>																	
EMU_ClkCtrl	<b>EMU 时钟控制位:</b> = 0, EMUclk 为 819.2K = 1, EMUclk 为 409.6K;需要注意脉冲宽度																	
DC PRFCFG[1:0]	<b>直流平均值更新速度控制寄存器</b> <table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th>DC PRFCFG1</th> <th>DC PRFCFG0</th> <th>直流平均值更新速度 (EMU = 819.2kHz)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1Hz</td> </tr> <tr> <td>0</td> <td>1</td> <td>2 Hz</td> </tr> <tr> <td>1</td> <td>0</td> <td>4 Hz</td> </tr> <tr> <td>1</td> <td>1</td> <td>8 Hz (default)</td> </tr> </tbody> </table>			DC PRFCFG1	DC PRFCFG0	直流平均值更新速度 (EMU = 819.2kHz)	0	0	1Hz	0	1	2 Hz	1	0	4 Hz	1	1	8 Hz (default)
DC PRFCFG1	DC PRFCFG0	直流平均值更新速度 (EMU = 819.2kHz)																
0	0	1Hz																
0	1	2 Hz																
1	0	4 Hz																
1	1	8 Hz (default)																
SlowPRMS	<b>慢速有效值功率更新速度控制位</b> =0, 在快速的基础上降低到 1/2 =1, 在快速的基础上降低到 1/4 (Default)																	
Adc_i1on	<b>I1_ADC 使能位</b> 0: 关闭第一路电流 ADC(Default) 1: 打开第一路电流 ADC																	
Adc_i2on	<b>I2_ADC 使能位</b> 0: 关闭第二路电流 ADC (Default) 1: 打开第二路电流 ADC																	
Adc_uon	<b>U_ADC 使能位</b> 0: 关闭电压 ADC (默认关闭) 1: 打开电压 ADC																	
Zxd[1:0]	<b>电压电流过零中断方式选择控制位</b> <table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th>ZXD1</th> <th>ZXD0</th> <th>电压 电流过零中断方式</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>正向过0中断</td> </tr> <tr> <td>0</td> <td>1</td> <td>负向过0中断</td> </tr> <tr> <td>1</td> <td>x</td> <td>双向过0中断</td> </tr> </tbody> </table>			ZXD1	ZXD0	电压 电流过零中断方式	0	0	正向过0中断	0	1	负向过0中断	1	x	双向过0中断			
ZXD1	ZXD0	电压 电流过零中断方式																
0	0	正向过0中断																
0	1	负向过0中断																
1	x	双向过0中断																
PRFCFG[1:0]	<b>快速有效值功率更新速度控制位</b> <table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th>PRFCFG1</th> <th>PRFCFG0</th> <th>有效值更新速度 (EMU = 819.2kHz)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>20Hz</td> </tr> <tr> <td>0</td> <td>1</td> <td>10Hz</td> </tr> <tr> <td>1</td> <td>0</td> <td>5Hz (default)</td> </tr> <tr> <td>1</td> <td>1</td> <td>2.5Hz</td> </tr> </tbody> </table>			PRFCFG1	PRFCFG0	有效值更新速度 (EMU = 819.2kHz)	0	0	20Hz	0	1	10Hz	1	0	5Hz (default)	1	1	2.5Hz
PRFCFG1	PRFCFG0	有效值更新速度 (EMU = 819.2kHz)																
0	0	20Hz																
0	1	10Hz																
1	0	5Hz (default)																
1	1	2.5Hz																

<b>P1OFFSET</b>		<b>基地址: 0x40013800</b>					
		<b>偏移地址: 78H</b>					
		Bit15...Bit0					
<b>Read:</b>		P1OFFSET[15:0]					
<b>Write:</b>							
<b>Reset:</b>	0	0	0	0	0	0	0

注意:

有功功率通道 1 校正寄存器采用二进制补码形式。  
P1OFFSETH 与 32 位寄存器 PowerP1 的低 16 位对齐。

$$P\_offset = \frac{P_{real} * EC * HFConst * 2^{31} * (-Err\%)}{2.304 * 10^{10}}$$

详细见校表过程。

<b>P2OFFSET</b>	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 7CH</b>							
	Bit15···Bit0							
<b>Read:</b>	P2OFFSET[15:0]							
<b>Write:</b>	P2OFFSET[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

有功功率通道 2 校正寄存器采用二进制补码形式。  
P2OFFSETH 与 32 位寄存器 PowerP2 的低 16 位对齐。

$$P\_offset = \frac{P_{real} * EC * HFConst * 2^{31} * (-Err\%)}{2.304 * 10^{10}}$$

<b>Q1OFFSET</b>	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 80H</b>							
	Bit15···Bit0							
<b>Read:</b>	Q1OFFSET[15:0]							
<b>Write:</b>	Q1OFFSET[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

无功功率通道 1 校正寄存器采用二进制补码形式。  
Q1OFFSETH 与 32 位寄存器 PowerQ1 的 bit[17..2]位对齐。

$$Q1\_offset = \frac{Q_{real} * EC * HFConst * 2^{29} * (-Err\%)}{2.304 * 10^{10}}$$

<b>Q2OFFSET</b>	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 84H</b>							
	Bit15···Bit0							
<b>Read:</b>	Q2OFFSET[15:0]							
<b>Write:</b>	Q2OFFSET[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

无功功率通道 2 校正寄存器采用二进制补码形式。  
Q2OFFSETH 与 32 位寄存器 PowerQ2 的低 16 位对齐。

$$Q2\_offset = \frac{Q_{real} * EC * HFConst * 2^{31} * (-Err\%)}{2.304 * 10^{10}}$$



<b>I1RMSOFFSET</b>	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 88H</b>							
	Bit15…Bit0							
<b>Read:</b>	I1RMSOFFSET [15:0]							
<b>Write:</b>	I1RMSOFFSET [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

电流通道 1 有效值校正寄存器采用二进制无符号数形式。

计算公式为:

当输入信号为 0 的时候, 多次读取 I1RMS, 取平均值后, 然后按照下面的公式计算。

$$I1RMSOFFSET = (I1RMS^2) / (2^{15})$$

如果外部噪声很大, 则会导致通过上述公式计算出的 I1RMSOFFSET 超限, 此时就只能通过用户自行写软件来去除板级过大的噪声, 该寄存器不能完全消除这种零漂噪声

<b>I2RMSOFFSET</b>	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 8CH</b>							
	Bit15…Bit0							
<b>Read:</b>	I2RMSOFFSET [15:0]							
<b>Write:</b>	I2RMSOFFSET [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

电流通道 2 有效值校正寄存器采用二进制无符号数形式。

计算公式为:

当输入信号为 0 的时候, 多次读取 I2RMS, 取平均值后, 然后按照下面的公式计算。

$$I2RMSOFFSET = (I2RMS^2) / (2^{15})$$

如果外部噪声很大, 则会导致通过上述公式计算出的 I2RMSOFFSET 超限, 此时就只能通过用户自行写软件来去除板级过大的噪声, 该寄存器不能完全消除这种零漂噪声

<b>URMSOFFSET</b>	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 90H</b>							
	Bit15…Bit0							
<b>Read:</b>	URMSOFFSET [15:0]							
<b>Write:</b>	URMSOFFSET [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

电压通道有效值校正寄存器采用二进制无符号数形式。

计算公式为:

当输入信号为 0 的时候, 多次读取 URMS, 取平均值后, 然后按照下面的公式计算。

$$URMSOFFSET = (URMS^2) / (2^{15})$$

如果外部噪声很大，则会导致通过上述公式计算出的 URMSOFFSET 超限，此时就只能通过用户自行写软件来去除板级过大的噪声，该寄存器不能完全消除这种零漂噪声

ROSICTRL (罗氏线圈控制寄存器)		基地址: 0x40013800 偏移地址: 94H						
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	Auto_Tc	Clkdelay	ClkDelay1	Clkdela	Adci_ctrl	Adci_ctrl	Adci_ctrl	DC_En
<b>Write:</b>	_EN	2						
<b>Reset:</b>	0	1	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	DC_S_En	XboFiter	LpIdleEn	LpStaCt	LpCtrl1	LpCtrl0	Rosi_i2_	Rosi_i1_
<b>Write:</b>		n						
<b>Reset:</b>	0	0	0	0			0	0

注：罗氏线圈使能是打开内部滤波器，将罗氏线圈送进来的微分信号进行积分还原，故外部通道使用罗氏线圈，一定要开启这一通道的罗氏线圈使能。

位	功能描述																																				
Auto_Tc_EN	<b>自动温度补偿控制位</b> 0: 自动温度补偿功能关闭 (default) 1: 自动温度补偿功能开启，注意，开启后，在不同的温度点下，EMU 会在计量参数寄存器中给出当前温度点下的补偿值																																				
Clkdelay[2:0]	<b>模拟数字之间的 clk delay，请保持默认值。</b>																																				
Adci_ctr[2:0]	<b>ADC 工作电流控制位(电流越大,相应 ADC 的功耗增加)</b> =000 (默认)，正常计量模式下，用偏置电流 10 uA; 在低功耗模式下，推荐用户配置为[001],用偏置电流 1.5uA. <table border="1" data-bbox="391 1249 1364 1637"> <thead> <tr> <th>Adci_ctr[2]</th> <th>Adci_ctr[1]</th> <th>Adci_ctr[0]</th> <th>偏置电流</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>10uA</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1.5 uA</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>3 uA</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>5 uA</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>7.5 uA</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>11.5 uA</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>12.5 uA</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>15.5 uA</td></tr> </tbody> </table>	Adci_ctr[2]	Adci_ctr[1]	Adci_ctr[0]	偏置电流	0	0	0	10uA	0	0	1	1.5 uA	0	1	0	3 uA	0	1	1	5 uA	1	0	0	7.5 uA	1	0	1	11.5 uA	1	1	0	12.5 uA	1	1	1	15.5 uA
Adci_ctr[2]	Adci_ctr[1]	Adci_ctr[0]	偏置电流																																		
0	0	0	10uA																																		
0	0	1	1.5 uA																																		
0	1	0	3 uA																																		
0	1	1	5 uA																																		
1	0	0	7.5 uA																																		
1	0	1	11.5 uA																																		
1	1	0	12.5 uA																																		
1	1	1	15.5 uA																																		
DC_En	<b>DC_En 直流平均值功能使能位:</b> = 0, 关闭直流平均值计算功能 = 1, 使能直流平均值计算功能																																				
DC_S_En	<b>DC_S_En 直流 S 视在功率能量使能控制位;</b> = 0, 关闭直流 S 视在功能 = 1, 使能直流 S 视在功能，两路 PowerS 均采用直流平均值 U I1 I2 计算，S 视在脉冲与能量均为直流																																				
XboFiter 50Hz	<b>直流计量时，陷波滤波器频率控制位（交流计量时，保持默认即可）:</b> = 0 ,陷波中心频率为 50Hz																																				

60Hz	= 1,陷波中心频率为 60Hz															
LpIdleEn	低功耗计量分时开启使能控制位： = 0, 低功耗计量为连续模式 = 1, 低功耗计量为分时模式，当 LpIdleTime = 0 时，空闲时间为 1s															
LpStaCtrl	低功耗计量模式下，稳定点数控制位 = 0, 固定丢 512 个点；(default) = 1, 固定丢 1024 个点；															
LpCtrl[1:0]	低功耗计量模式下，平均时间控制位 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>LpCtrl [1]</th> <th>LpCtrl [0]</th> <th>平均时间 (s)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0.5</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>2</td> </tr> <tr> <td>1</td> <td>1</td> <td>4</td> </tr> </tbody> </table>	LpCtrl [1]	LpCtrl [0]	平均时间 (s)	0	0	0.5	0	1	1	1	0	2	1	1	4
LpCtrl [1]	LpCtrl [0]	平均时间 (s)														
0	0	0.5														
0	1	1														
1	0	2														
1	1	4														
Rosi_i2_en	电流通道2的罗氏线圈使能位 0: 关闭电流通道2罗氏线圈功能 1: 使能电流通道2罗氏线圈功能															
Rosi_i1_en	电流通道1的罗氏线圈使能位 0: 关闭电流通道1罗氏线圈功能 1: 使能电流通道1罗氏线圈功能															

<b>ANA_control register</b> (模拟控制寄存器)			基地址: 0x40013800 偏移地址: 98H					
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	ldo_emu	lowfreq_	Vref_	Vref_	Vref_	Chop_	Reserved	Adc_Cho
<b>Write:</b>	_en	en	ctrl2	ctrl1	ctrl0	Vref_en		
<b>Reset:</b>	0	0	1	0	0	1	1	1
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	uchop_en	Reserved	Reserved	Reserved	Reserved	Reserved	Vref_Ch	lowVref_
<b>Write:</b>		4	3	2	1	0	op_buf_e	
<b>Reset:</b>	1	0	1	0	0	1	1	0

注意，默认值为 27A6H，当用户需要使能 EMU 内部计量，必须配置为 A7A6H。

位	功能描述
ldo_emu_en	EMU Analog 供电 LDO 使能控制位： = 0, 关闭LDO, 仅用于常数计量模式下使用，以降低功耗；(default) = 1, 开启LDO;
lowfreq_en	低功耗模式下，Lvref参与计量控制位 当LvrefEn = 0, 使用Vref作为计量的基准

	= 1, 使用 Lvref 作为计量的基准
Vref_Ctrl[2:0]	<b>Vref_Ctrl[2: 0], Vref TC 顶点调整 code</b> <b>Default = 100;</b>
Chop_Vref_en	<b>Vref chop 控制位:</b> = 0, Vref chop off = 1, Vref chop on; (default)
Reserved	内部保留寄存器, 保持默认值
Adc_ChopEn	<b>Adc chop 控制位</b> = 0, 关闭 Adc chop, = 1, 使能 Adc chop, 建议开启 (default)
uchop_en	<b>电压通道 chop 控制位:</b> =0: 关闭电压通道 chop; =1: 使能电压通道 chop; 建议开启 (default)
Reserved[4:2]	内部保留寄存器, 保持默认值 <b>Default = 010</b>
Reserved[1:0]	内部保留寄存器, 保持默认值 <b>Default = 01</b>
Vref_Chop_buf_en	<b>Vref chop buf使能控制位:</b> = 0, 关闭 = 1, 开启 (default)
lowVref_isel	<b>Lvref偏置电流控制位, 当使能Lvref作为计量的基准时, 可通过偏置电流设置来调整ADC的功耗:</b> = 0, 偏置电流为10ua (default) = 1, 偏置电流为5ua

<b>UCONST</b>	<b>基地址: 0x40013800</b>
	<b>偏移地址: 9CH</b>
	Bit15...Bit0
<b>Read:</b>	<b>UCONST [15:0]</b>
<b>Write:</b>	
<b>Reset:</b>	0    0    0    0    0    0    0    0    0

注意:

电压常量寄存器采用二进制无符号数形式。

失压情况下低功耗计量参与计量的电压寄存器 16 位, 采用二进制无符号数形式。

该寄存器主要目的是替代 Urms 寄存器的值参与计量, 用户也可以根据需要写入自己需要的值。该寄存器与 Urms 的对应关系如下:

URms 是 24 位寄存器, 其中最高位恒为 0, 即 23 位有效; 而 UCONST 是 16 位有效位寄存器, 所以在正常信号输入下读出来的寄存器 URms 值要右移 7 位, 即除以  $2^7$ , 就是 Uconst 的值

<b>LpIdleTime</b>	<b>基地址: 0x40013800</b>
	<b>偏移地址: A0H</b>
	Bit10...Bit0
<b>Read:</b>	<b>LpIdleTime [10:0]</b>

Write:								
Reset:	0	0	0	0	0	0	0	0

注意:

低功耗计量模式下, 分时模式的空闲时间设置寄存器, 默认为 0, 即空闲 1S, 再开固定时间; 空闲时间可写范围 0---599, 即空闲 1S--600S, 再开固定时间。

<b>USAGLVL</b>	<b>基地址: 0x40013800</b> <b>偏移地址: A4H</b>							
	Bit15...Bit0							
Read:	USAGLVL [15:0]							
Write:	USAGLVL [15:0]							
Reset:	0	0	0	0	0	0	0	0

注意:

该寄存器为无符号数据, 其 16bit 和 19bit 的 ADC 波形寄存器高对齐, ADC 波形寄存器的绝对值和 USAGLVL 相比较, 在用户设定的 SagCyc 内, 如果 ADC 波形都小于 USAGLVL 寄存器, 则在 SagCyc 的最后一个半周波, 更新 SAG 标志。

Upeak 寄存器, 22bit 有符号数据, 采用二进制补码格式, 与波形采样数据对齐。USAGLVL 寄存器, 16bit 无符号数据, 与波形采样采样的绝对值高对齐。

有效值和峰值之间存在 1.414 倍的关系。所以有如下关系:

$$U_{peak} = U_{rms} * 1.414 / 2^5$$

USAGLVL 阈值如果设成峰值的 80%, 则:

$$USAGLVL = 0.8 * U_{peak} / 2^3 = 0.8 * U_{rms} * 1.414 / 2^8$$

<b>USagCyc</b>	<b>基地址: 0x40013800</b> <b>偏移地址: A8H</b>							
	Bit15...Bit0							
Read:	SagCyc [15:0]							
Write:	SagCyc [15:0]							
Reset:	0	0	0	0	0	0	0	0

注意:

16 位无符号数, 用于设定 SAG 检测数据长度, 即 SAG 功能检测设定的数据长度后给出峰值。1LSB 对应半个周波。

<b>UOVLVL</b>	<b>基地址: 0x40013800</b> <b>偏移地址: ACH</b>							
	Bit15...Bit0							
Read:	UOVLVL [15:0]							
Write:	UOVLVL [15:0]							
Reset:	0	0	0	0	0	0	0	0

## 注意

该寄存器为无符号数据,其 16bit 和 22bit 的 ADC 波形寄存器高对齐,ADC 波形寄存器的绝对值和 UOVLVL 相比较, 在用户设定的 OvCyc 内, 如果 ADC 波形超过 UOVLVL 寄存器, 则在 OvCyc 的最后一个半周波, 更新 UOVIF 标志, 同时在用户设定的 OVCyc 时间后, 更新 Upeak 值。

Upeak 寄存器, 19bit 有符号数据, 采用二进制补码格式。UOVLVL 寄存器, 16bit 无符号数据, 与波形采样的绝对值高对齐。

有效值和峰值之间存在 1.414 倍的关系。所以有如下关系:

$$Upeak = Urms * 1.414 / 2^5$$

UOVLVL 阈值如果设成峰值的 120%, 则:

$$UOVLVL = 1.2 * Upeak / 2^3 = 1.2 * Urms * 1.414 / 2^8$$

<b>OvCyc</b>	<b>基地址: 0x40013800</b>							
	<b>偏移地址: B0H</b>							
	Bit15...Bit0							
<b>Read:</b>	OvCyc [15:0]							
<b>Write:</b>	OvCyc [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

## 注意:

16 位无符号数, 用于设定 UPeak/IPeak 检测数据长度, 即 Peak 值在检测设定的数据长度后给出峰值。1 个 LSB 对应半个周波, 当检测值超过设定的阈值后, 给出中断标志。

<b>IOVLVL</b>	<b>基地址: 0x40013800</b>							
	<b>偏移地址: B4H</b>							
	Bit15...Bit0							
<b>Read:</b>	IOVLVL [15:0]							
<b>Write:</b>	IOVLVL [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

## 注意:

该寄存器为无符号数据, 其 16bit 和 22bit 的 ADC 波形寄存器高对齐, ADC 波形寄存器的绝对值和 IOVLVL 相比较, 在用户设定的 OvCyc 内, 如果 ADC 波形超过 IOVLVL 寄存器, 则在 OvCyc 的最后一个半周波, 更新 IOVIF 标志, 同时在用户设定的 OVCyc 时间后, 更新 Ipeak 值。

Ipeak 寄存器, 19bit 有符号数据, 采用二进制补码格式, 与波形采样数据对齐。IOVLVL 寄存器, 16bit 无符号数据, 与波形采样的绝对值高对齐。

有效值和峰值之间存在 1.414 倍的关系。所以有如下关系:

$$Ipeak = Irms * 1.414 / 2^5$$

IOVLVL 阈值如果设成峰值的 120%, 则:

$$IOVLVL = 1.2 * Ipeak / 2^3 = 1.2 * Irms * 1.414 / 2^8$$

<b>ZXILVL</b>	<b>基地址: 0x40013800</b>							
	<b>偏移地址: B8H</b>							
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12 ... 3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>	

<b>Read:</b>	ZX15	ZX4	ZX13	ZX12...ZX3	ZX2	ZX1	ZX0
<b>Write:</b>							
<b>Reset:</b>	0	0	0	0	0	0	0

注意:

电流有效值与 **ZXILVL** 相比较。**ZXILVL** 对应 IRMS 的低 16 位 Bit15...bit0。

过零电流域值设置寄存器，当电流有效值小于用户设定的电流过零域值设置寄存器的时候，则不输出电流过零信号，内部输出恒为 0。

<b>PDataCpH</b>	<b>基地址: 0x40013800</b>						
	<b>偏移地址: BCH</b>						
	Bit15...Bit0						
<b>Read:</b>	IChk[7:0]						
<b>Write:</b>							
<b>Reset:</b>	0	0	0	0	0	0	0

<b>PDataCpL</b>	<b>基地址: 0x40013800</b>						
	<b>偏移地址: COH</b>						
	Bit15...Bit0						
<b>Read:</b>	IChk[7:0]						
<b>Write:</b>							
<b>Reset:</b>	0	0	0	0	0	0	0

注意:

常数计量下，常数有功功率的高低 16 位，组成 32 位常数有功功率，当用户使能常数计量功能并 LoadDataCp 后，由 PDATAACP 代替有功功率发脉冲和能量累加；

<b>QDataCpH</b>	<b>基地址: 0x40013800</b>						
	<b>偏移地址: C4H</b>						
	Bit15...Bit0						
<b>Read:</b>	IChk[7:0]						
<b>Write:</b>							
<b>Reset:</b>	0	0	0	0	0	0	0

<b>QDataCpL</b>	<b>基地址: 0x40013800</b>						
	<b>偏移地址: C8H</b>						
	Bit15...Bit0						
<b>Read:</b>	IChk[7:0]						
<b>Write:</b>							
<b>Reset:</b>	0	0	0	0	0	0	0

注意:

常数计量下，常数无功功率的高低 16 位，组成 32 位常数无功功率，当用户使能常数计量功能并 LoadDataCp 后，由 QDATAACP 代替无功功率发脉冲和能量累加；

<b>SDataCpH</b>		<b>基地址:</b> 0x40013800 <b>偏移地址:</b> CCH						
	Bit15...Bit0							
<b>Read:</b>	IChk[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>SDataCpL</b>		<b>基地址:</b> 0x40013800 <b>偏移地址:</b> DOH						
	Bit15...Bit0							
<b>Read:</b>	IChk[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

常数计量下, 常数视在功率的高低 16 位, 组成 32 位常数视在功率, 当用户使能常数计量功能并 LoadDataCp 后, 由 SDATACP 代替视在功率发脉冲和能量累加;

<b>Filter control register</b> (滤波器系数控制寄存器)		<b>基地址:</b> 0x40013800 <b>偏移地址:</b> D4H						
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	X	X	X	RosiPga	RosiPga	ZXsourc	PahseShi	PahseShi
<b>Write:</b>	X	X	X	Ctr1	Ctr0	eSel	ftCtr1	ftCtr1
<b>Reset:</b>	0	0	0	0	0	0	1	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	RosiCtr1	RosiCtr0	HCtl2	HCtl1	HCtl0	LCtl2	LCtl1	LCtl0
<b>Write:</b>								
<b>Reset:</b>	1	1	1	1	0	1	1	1

位	功能描述
RosiPgaCtr[1:0]	两路电流罗氏线圈增益控制档 1 (default) 2 4 8
ZXsourceSel	电流电压过零源头选择: = 0, 选择电压电流过零源头为高通后; = 1, 选择电压过零为 90 度移相后, 电流过零为经 rosi 后;
PahseShiftCtr[1:0]	90 度移相滤波器系数控制档 11 12 13 14(default)
RosiCtr[1:0]	两路电流罗氏线圈滤波器系数控制档 11 12 13 14(default)
HCtl[2:0]	内部系数 1 选择控制位 当 HCtl1[2:0] = 000--111 时, 相应内部系数 1 为 4--11; 默认值为 110, 对应内部系数 1 为 10;



LCt1[2:0]	内部系数 2 选择控制位 当 LCt1[3:0] = 000--111 时，相应内部系数 2 为 4--11； 默认值为 111，对应内部系数 2 为 11；
-----------	---

<b>TUgain</b>	<b>基地址:</b> 0x40013800 <b>偏移地址:</b> D8H							
	Bit15...Bit0							
<b>Read:</b>	IChk[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>TI1gain</b>	<b>基地址:</b> 0x40013800 <b>偏移地址:</b> DCH							
	Bit15...Bit0							
<b>Read:</b>	IChk[15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>TI2gain</b>	<b>基地址:</b> 0x40013800 <b>偏移地址:</b> E0H							
	Bit15...Bit0							
<b>Read:</b>	TI2gain [15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意:

温度手动补偿寄存器 TUgain, TI1gain, TI2gain。用于用户手动补偿温度对三路通道的影响。

对 3 路 ADC 增加了手动增益校正,主要针对 VREF 和外围由于温度的变化引起 ADC 发生的变化。

举例

已知: 由于 Vref 和外围变化导致的 U 通道有效值变化 err%, 注意是有效值的 err。

计算公式:  $gain = (-err\%)/(1 + err\%)$

如果  $gain \geq 0$ , 则  $TUgain = INT[gain * 2^{15}]$

否则  $gain < 0$ , 则  $TUgain = INT[2^{16} + gain * 2^{15}]$

TI1gain 和 TI2gain 手动补偿寄存器的计算方式与 TUgain 一样。

自动温度补偿曲线系数寄存器

<b>UTCcoeffA</b>	<b>基地址:</b> 0x40013800 <b>偏移地址:</b> E4H							
	Bit15...Bit0							

Read:	UTCcoeffA[15:0]								
Write:									
Reset:	0	0	0	0	0	0	0	0	0

U 通道自动温度补偿曲线二次项系数。

<b>UTCcoeffB</b>		<b>基地址: 0x40013800</b>							
		<b>偏移地址: E8H</b>							
		Bit15...Bit0							
Read:	UTCcoeffB [15:0]								
Write:									
Reset:	0	0	0	0	0	0	0	0	0

U 通道自动温度补偿曲线一次项系数。

Bit 15—bit13 为一次系数的整数部分，最高位为符号位，可以表示正负 3（应该为 -4 --- +3 范围），bit12—bit0 表示一次系数的小数部分，最高位为符号位，可表示正负 1。

所以 UTCcoeffB = Bit15（符号）bit14—bit13 + bit12（符号）bit11—bit0

举例如下：

如果 UTCcoeffB 值为 0x1400 则系数为：整数部分为 0，小数部分为  $-(2^{13}-0x1400)/2^{12} = -0.75$   
 则 UTCcoeffB = 0 + -0.75 = -0.75

如果 UTCcoeffB 值为 0x2400 则系数为：整数部分为 1，小数部分为  $0x400/2^{12} = 0.25$   
 则 UTCcoeffB = 1 + 0.25 = 1.25

<b>UTCcoeffC</b>		<b>基地址: 0x40013800</b>							
		<b>偏移地址: ECH</b>							
		Bit15...Bit0							
Read:	UTCcoeffC[15:0]								
Write:									
Reset:	0	0	0	0	0	0	0	0	0

U 通道自动温度补偿曲线常数项。

自动温度补偿使能后，TPScore 值为芯片内部 TPS 寄存器的高 10 位。

TPS code 为二进制补码格式。

自动温度电压补偿输出值为：

A0H	AutoUgain	R	2	0x000000	U 通道温度自动补偿系数
-----	-----------	---	---	----------	--------------

实际内部计算公式：

$$\text{AutoUgain} = \text{UTCcoeffA}/32768 * (\text{TPSCODE}/4)^2 + \text{UTCcoeffB} * (\text{TPSCODE}/4) + \text{UTCcoeffC}/32768$$

注意： UTCcoeffB 分为整数部分和小数部分，各自都有符号位，UTCcoeffC 也是有符号数，表示范围 -32768---32767。

<b>I1TCcoeffA</b>	<b>基地址: 0x40013800</b>							
-------------------	------------------------	--	--	--	--	--	--	--

		<b>偏移地址:</b> FOH						
		<b>Bit15…Bit0</b>						
<b>Read:</b>	I1TCcoeffA [15:0]							
<b>Write:</b>	I1TCcoeffA [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

I1 通道自动温度补偿曲线二次项系数。

<b>I1TCcoeffB</b>		<b>基地址:</b> 0x40013800						
		<b>偏移地址:</b> F4H						
		<b>Bit15…Bit0</b>						
<b>Read:</b>	I1TCcoeffB [15:0]							
<b>Write:</b>	I1TCcoeffB [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

I1 通道自动温度补偿曲线一次项系数。

Bit 15—bit13 为一次系数的整数部分，最高位为符号位，可以表示正负 3（应该为 -4 --- +3 范围），bit12—bit0 表示一次系数的小数部分，最高位为符号位，可表示正负 1。

<b>I1TCcoeffC</b>		<b>基地址:</b> 0x40013800						
		<b>偏移地址:</b> F8H						
		<b>Bit15…Bit0</b>						
<b>Read:</b>	I1TCcoeffC [15:0]							
<b>Write:</b>	I1TCcoeffC [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

I1 通道自动温度补偿曲线常数项。

<b>I2TCcoeffA</b>		<b>基地址:</b> 0x40013800						
		<b>偏移地址:</b> FCH						
		<b>Bit15…Bit0</b>						
<b>Read:</b>	I2TCcoeffA [15:0]							
<b>Write:</b>	I2TCcoeffA [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

I2 通道自动温度补偿曲线二次项系数。

<b>I2TCcoeffB</b>		<b>基地址:</b> 0x40013800						
		<b>偏移地址:</b> 100H						
		<b>Bit15…Bit0</b>						
<b>Read:</b>	I2TCcoeffB [15:0]							
<b>Write:</b>	I2TCcoeffB [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

I2 通道自动温度补偿曲线一次项系数。

Bit 15—bit13 为一次系数的整数部分，最高位为符号位，可以表示正负 3（应该为 -4 --- +3 范围），bit12—bit0 表示一次系数的小数部分，最高位为符号位，可表示正负 1。

<b>I2TCcoeffC</b>		<b>基地址: 0x40013800</b>						
		<b>偏移地址: 104H</b>						
		<b>Bit15…Bit0</b>						
<b>Read:</b>	I2TCcoeffC [15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

I1 通道自动温度补偿曲线常数项。

注：两路电流的自动温度补偿系数计算方式与电压通道一样。

<b>LoadDataCp</b>		<b>基地址: 0x40013800</b>						
		<b>偏移地址: 108H</b>						
		<b>Bit15…Bit0</b>						
<b>Read:</b>	LoadDataCp [15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

进入常数计量模式后，当用户向 LoadDataCp 写入 0x00BC 后，常数功率值 P/Q/SDATACP load 进入累加源头。

<b>SRSTREG</b>		<b>基地址: 0x40013800</b>						
		<b>偏移地址: 114H</b>						
		<b>Bit15…Bit0</b>						
<b>Read:</b>	IChk [15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意：

当用户写入 0x55 后，复位所有校表参数寄存器；当用户写入 0xAA 后，复位整个 EMU 模块。

## 21.4 校表过程

### 1. 高频脉冲常数设置（同一批表只需同样的 HFCONST）

通过 HFConst 寄存器将用户样表的误差精度调整到 10% 以内。有两种方式计算。

#### 21.4.1 方案一

HFCONST 寄存器的默认值为 0x0080

用户观察电表的初始误差为 Err%，则按照下面公式将误差调整到 10 以内：

$$\text{HFCONST} = 0x0080 * (1 + \text{Err}\%)$$

举例说明：

电表表常数（EC）设置为 3200，功率因数为 1，HFCONST 寄存器为默认值 0x0080，观察标准表上显

示的误差为 52.8%。

根据公式： $HFCNST = 0x0080 * (1 + Err\%)$

计算得到： $HFCNST = 0x0080 * (1 + 52.8\%) = 0x00C3$

将 0x00C3 写入 HT500X 的 HFCNST (4CH) 寄存器：

写入后标准表的显示误差应该在 10% 以内

## 21.4.2 方案二

ADC 采样率  $f_s = 12.8\text{KHz}$  时

$HFCNST = 2.332 * V_u * V_i * 10^{10} / (EC * U_n * I_b)$

$V_u$ : 额定电压输入时, 电压通道的电压 (引脚上电压 × 放大倍数)

$V_i$ : 额定电流输入时, 电流通道的电压 (引脚上电压 × 放大倍数)

$U_n$ : 额定输入的电压

$I_b$ : 额定输入的电流

EC: 电表常数

$f_s$  为其他值时, HFCNST 按比例变化即可。

### 举例说明:

电表表常数 (EC) 设置为 3200, 功率因数为 1。

$U_n$  (额定电压) 为 220V,  $I_b$  (额定电流) 为 5A,  $V_u$  (电压通道的电压) 为 0.22V

$V_i$  (电流通道的电压) 为 1.75mV, 内部电流通道 16 倍增益,  $V_i * 16 = 28\text{mV}$

根据公式： $HFCNST = 2.332 * V_u * V_i * 10^{10} / (EC * U_n * I_b)$  计算得到

$HFCNST = 2.332 * 0.22 * 0.028 * 10^{10} / (3200 * 220 * 5) = 0x0028$

将 0x0028 写入 HT500X 的 HFCNST (4CH) 寄存器：

写入后标准表的显示误差应该在 10% 以内

## 2. 第一通道有功、无功和视在增益校正

只需要在额定输入、功率因数为 1 时根据有功计算。通常有功、无功和视在增益写入相同的值。

已知：

标准表上读出误差为  $err\%$

计算公式：

$$P_{gain} = \frac{-err\%}{1 + err\%}$$

如果  $P_{gain} \geq 0$ , 则  $GP1 = \text{INT}[P_{gain} * 2^{15}]$

否则  $P_{gain} < 0$ , 则  $GP1 = \text{INT}[2^{16} + P_{gain} * 2^{15}]$

同时将计算出的 GP1 的值写入到 GQ1 和 GS1。

### 举例说明:

功率源上输出 220v、5A、功率因数为 1 的信号, 标准表上显示的误差为 3.8%

则  $P_{gain} = -0.038 / (1 + 0.038) = -0.0366$

该数小于 0, 需转换为补码, 则  $-0.0366 * 2^{15} + 2^{16} = 0x\text{FB50H}$

将 0x\text{FB50H} 同时写入 GP1(0CH)/GQ1(10H)/GS1(14H)寄存器, 即完成阻性增益校正。

### 3. 第一通道相位校正

在增益已经校正好之后，进行相位补偿。在功率因素 0.5L 处进行校正。

已知：0.5L 处标准表误差读数为  $err\%$

相位补偿公式：

$$\theta = \frac{-err\%}{1.732}$$

如果  $\theta \geq 0$ ， $GPhs = \theta * 2^{15}$

否则  $\theta < 0$ ， $GPhs = 2^{16} + \theta * 2^{15}$

#### 举例说明：

校完阻性增益后，将功率因数改为 0.5L，标准表显示的误差为-0.4%，则

$$\theta = -(-0.004)/1.732=0.0023$$

$$Gphs1=0.0023*2^{15}=75.3$$

取整后为 0x4BH，写入角度校正寄存器 Gphs1(18H)即可。

### 4. 电流通道 2 增益校正（做防窃电时必需）

做防窃电时，需要对两个通道的电流有效值或者功率进行比较，因而在同样电流输入下，通道 1 与通道 2 的寄存器值应该相等。

通过通道 2 增益校正寄存器 I2GAIN(30H)，使输入同样电流情况下，二者寄存器的值一致。

假设输入同样额定电流，电流通道 1 有效值寄存器读数为  $I1_{rms}$ ，电流通道 2 有效值寄存器读数为  $I2_{rms}$ ，

电流通道 1 有功功率  $Power1$ ，电流通道 2 有功功率  $Power2$

#### 如果选择电流防窃电方式：

$$\text{则 Gain} = I1_{rms}/I2_{rms} - 1$$

#### 如果选择功率防窃电方式：

$$\text{则 Gain} = Power1/Power2 - 1$$

如果  $Gain \geq 0$ ， $I2Gain = Gain * 2^{15}$ ；如果  $Gain < 0$ ， $I2Gain = Gain * 2^{15} + 2^{16}$

### 5. 第二通道增益校正、相位校正

第二通道增益校正、相位校正与通道 1 类似。

### 6. Poffset 校正（小信号有功功率校正）

在经过以上步骤之后，用户在  $I_b = 100\%$  的时候电表误差校正到 0 附近；

观察小信号  $x\%I_b$ （5%，2%）点的电表误差为  $Err\%$ ，

$x\%I_b$  点在阻性下读取标准表上输出的有功功率值  $Preal$

应用公式来计算：

$$P\_offset = \frac{Preal * EC * HFConst * 2^{31} * (-Err\%)}{2.3 * 10^{10}}$$

#### 举例说明：

额定电压 220V，额定电流( $I_b$ )5A，表常数为 3200，快速脉冲寄存器（HFCONST）读取为 0x66；

电表在  $I_b = 100\%$  时误差校正为 0 附近，观察小信号 5% 点的电表误差为 0.5%；  
从标准表上读取小信号 5% 点的输出功率为 55.02 (Preal 为标准表上的显示功率输出)  
根据公式计算得到：

$$\begin{aligned} Poffset &= (\text{Preal} * EC * HFCONST * 2^{31} * (-Err\%)) / (2.3 * 10^{10}) \\ &= (55.02 * 3200 * 102 * 2^{31} * (-0.5\%)) / (2.3 * 10^{10}) \\ &= -2091.4 \end{aligned}$$

由于  $Poffset < 0$ ，所以写入寄存器 P1offset 的值为  $2^{16} + Poffset = 63444$   
取个整数 63444 (0xF7D4)  
将 0xF7D4 写入 HT500X 的 P1offset(78H) 寄存器。

7, IRMS 增益、URMS 增益和两个通道的功率增益转换系数校正  
这些参数没有相应的寄存器，需要由用户根据需要自行计算获取。  
举例说明：

以电流通道 1 有效值为例，电流通道 1 标准台输出 5A 电流有效值，电流通道 1 有效值寄存器 RMS\_I1 (3CH) 的值读取得 0x03BA55，如果用户希望在液晶上显示出 5A，则需要自行计算两者之间的转换系数如下： $K = 5 / 0x03BA55 = 2.046 * 10^{-5}$   
这里的 K 就是转换系数，之后用户根据读取的 RMS\_I1 的值乘这个 K，则得到正确的电流显示值。  
详见有效值输出章节和功率参数输出章节。

## 22 电气规格

### 22.1 DC 参数

符号	参数说明	测试条件	最小	典型	最大	单位
VCC	输入电源	输入电源引脚上的电压	2.2	3.3	5.5	V
Vih	高电平输入电压	Reset 引脚, TEST 引脚	0.8VCC			V
Vih	高电平输入电压	PC9,PC10,PE0,PE3,PE6	0.6VCC			V
Vih	高电平输入电压	除了电源和地, 除了 Reset, TEST, PC9,PC10,PE0,PE3, PE6 之外的所有引脚	0.7VCC			V
Vih	低电平输入电压	除了电源和地之外的所有引脚			0.2VCC	V
Ioh	高电平输出电流	VCC=3.3V I/O 口上电压 Vio 降低到 0.9VCC 测试引脚为: PA.6, PA.7, PA.8, PC.0	10			mA
Iol	低电平输出电流	VCC=3.3V I/O 口上电压 Vio 升高到 0.1VCC 测试引脚为: PA.6, PA.7, PA.8, PC.0	20			mA
Ioh	高电平输出电流	VCC=3.3V I/O 口上电压 Vio 降低到 0.9VCC 测试引脚为: PD.4,PD.5,PD.6,PD.7 PC.5,PC.6,PC.7,PC.8,PC.9, PC.10,PC.11,PE.1	5			mA
Iol	低电平输出电流	VCC=3.3V I/O 口上电压 Vio 降低到 0.9VCC 测试引脚为: PD.4,PD.5,PD.6,PD.7 PC.5,PC.6,PC.7,PC.8,PC.9, PC.10,PC.11,PE.1	5			mA
Ioh	高电平输出电流	VCC=3.3V I/O 口上电压 Vio 降低到 0.9VCC	3			mA



		测试引脚为： 除了 PA.6,PA.7,PA.8,PC.0, PD.4,PD.5,PD.6,PD.7 PC.5,PC.6,PC.7,PC.8,PC.9, PC.10,PC.11,PE.1				
Iol	低电平输出电流	VCC=3.3V I/O 口上电压 Vio 降低到 0.9VCC 测试引脚为： 除了 PA.6,PA.7,PA.8,PC.0, PD.4,PD.5,PD.6,PD.7 PC.5,PC.6,PC.7,PC.8,PC.9, PC.10,PC.11,PE.1	3			mA

## 22.2 极限参数

符号	参数说明	测试条件	最小	最大	单位
VCC	输入电源	输入电源引脚上的电压	2.2	5.5	V
Vi	输入电压	所有的数字引脚	0	5.5	V
Via	ADC 输入极限电压	测试 ADC 引脚输入电压，高于此电压可能会导致该引脚损坏	0	3.0	V
Vib	VBAT 输入极限电压	测试 VBAT 引脚输入电压，高于此电压可能会导致该引脚损坏	0	3.6	V
Idd	输入电流	VCC 电源引脚		50	mA
Iss	地上电流	所有 GND 引脚		50	mA
Tstg	存储温度	芯片的极限存储温度	-65	+150	°C
Vesd	静态 ESD (HBM)	芯片所有的引脚	-6000	+4000	V

注：VCC 输入的系统电压可承受 1.5 倍标称电压。

## 22.3 功耗参数

测试条件说明	最小	典型	最大	单位
<b>测试条件：</b> 以下功耗参数均为在 3.3V 供电下测试得到				
芯片处于 Hold 模式下最低功耗（具体功耗值会根据用户在 Hold 模式下打开不同的数字模块而不同）		3.7	8	uA
芯片处于 Sleep 模式下最低功耗		2.9	7	uA

芯片在低频 32768Hz 下运行，LCD 开启时的功耗		48		uA
高频 RC 下运行，高频 RC 配置为 9.5M，CLKOUT 关闭，所有数字模块打开 /关闭		2.85 /1.52		mA
高频 RC 下运行，高频 RC 配置为 8M，CLKOUT 关闭，所有数字模块打开 /关闭		2.19 /0.86		mA
高频 RC 下运行，高频 RC 配置为 4.8M，CLKOUT 关闭，有数字模块打开 /关闭		1.75 /0.53		mA
高频 RC 下运行，高频 RC 配置为 2.4M，CLKOUT 关闭，有数字模块打开 /关闭		1.42 /0.35		mA
高频 RC 下运行，高频 RC 配置为 1.2M，CLKOUT 关闭，有数字模块打开 /关闭		1.25 /0.27		mA
高频 RC 下运行，高频 RC 配置为 600K，CLKOUT 关闭，有数字模块打开 /关闭		1.17 /0.23		mA
高频 RC 下运行，高频 RC 配置为 300K，CLKOUT 关闭，有数字模块打开 /关闭		1.13 /0.21		mA
PLL 时钟下运行，CPU 运行 PLL 时钟 39.32M，所有数字模块打开/关闭		7.34 /4.64		mA
PLL 时钟下运行，CPU 运行 PLL 时钟 19.66M，所有数字模块打开/关闭		4.51 /3.03		mA
PLL 时钟下运行，CPU 运行 PLL 时钟 9.83M，所有数字模块打开 /关闭		3.20 /1.69		mA
PLL 时钟下运行，CPU 运行 PLL 时钟 4.915M，所有数字模块打开 /关闭		2.53 /1.00		mA
PLL 时钟下运行，CPU 运行 PLL 时钟 2.45M，所有数字模块打开 /关闭		2.19 /0.66		mA
PLL 时钟下运行，CPU 运行 PLL 时钟 1.22M，所有数字模块打开 /关闭		2.02 /0.48		mA
PLL 时钟下运行，CPU 运行 PLL 时钟 620k，所有数字模块打开 /关闭		1.94 /0.39		mA
EMU 单路 ADC 功耗（偏置电流 1.5uA 时）		130		uA
EMU 单路 ADC 功耗（偏置电流 10uA 时）		600		uA

测试条件说明	最小	典型	最大	单位
<b>测试条件：</b> 以下功耗参数均为在 3.3V 供电下测试得到				
高频 RC 模块功耗（HRC）		114		uA
低频 RC 模块功耗（LRC）		0.9		uA
PLL 模块功耗		197		uA
RTC 模块功耗（含 TBS 补偿）		1.2		uA
LCD 模块功耗（1/64 快速充电模式）		4.3		uA
BOR 模块功耗		8		uA
LVDIN 模块		7		uA

## 22.4 计量 ADC 参数

符号	参数说明	测试条件	最小	典型	最大	单位
Via	ADC 输入电压		0		±800	mVp
Fc	ADC 转换频率	VCC=3.3V	1/16		1	Hz
Res	分辨率	VCC=3.3V		0.012		mv/LSB
Vref TC	计量基准温度系数			10		ppm/°C

## 22.5 VBAT 测试参数

符号	参数说明	测试条件	最小	典型	最大	单位
Via	VBAT 输入电压		1		3.6	V
Fc	ADC 转换频率	VCC=3.3V	1/16		1	Hz
Res	分辨率	VCC=3.3V		0.064		mv/LSB

## 22.6 VCC 测试参数

符号	参数说明	测试条件	最小	典型	最大	单位
Via	VCC 输入电压		2.2		5.5	V
Fc	ADC 转换频率		1/16		1	Hz
Res	分辨率			0.089		mv/LSB

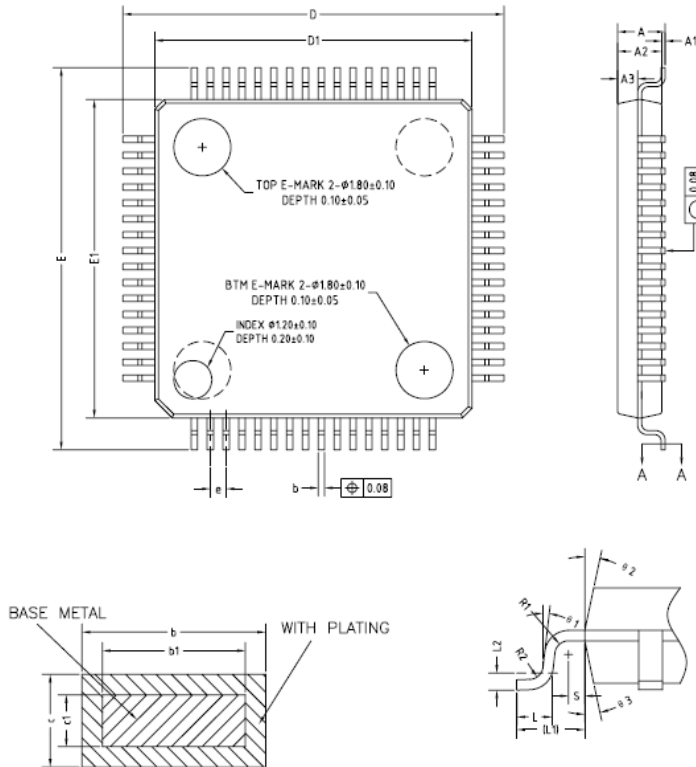
## 22.7 IO 口性能指标

	-40°C	27°C	90°C
Isink@0.1VCC	7mA	5mA	4mA
	14mA	10mA	8mA
	56mA	40mA	32mA

Isource@0.9VCC	5mA	4mA	3.5mA
	10mA	8mA	7mA
	15mA	12mA	10.5mA
总 Isource 不能大于 50mA,总 Isink 不能大于 120mA。			

## 23 封装

### 23.1 HT5007:LQFP64

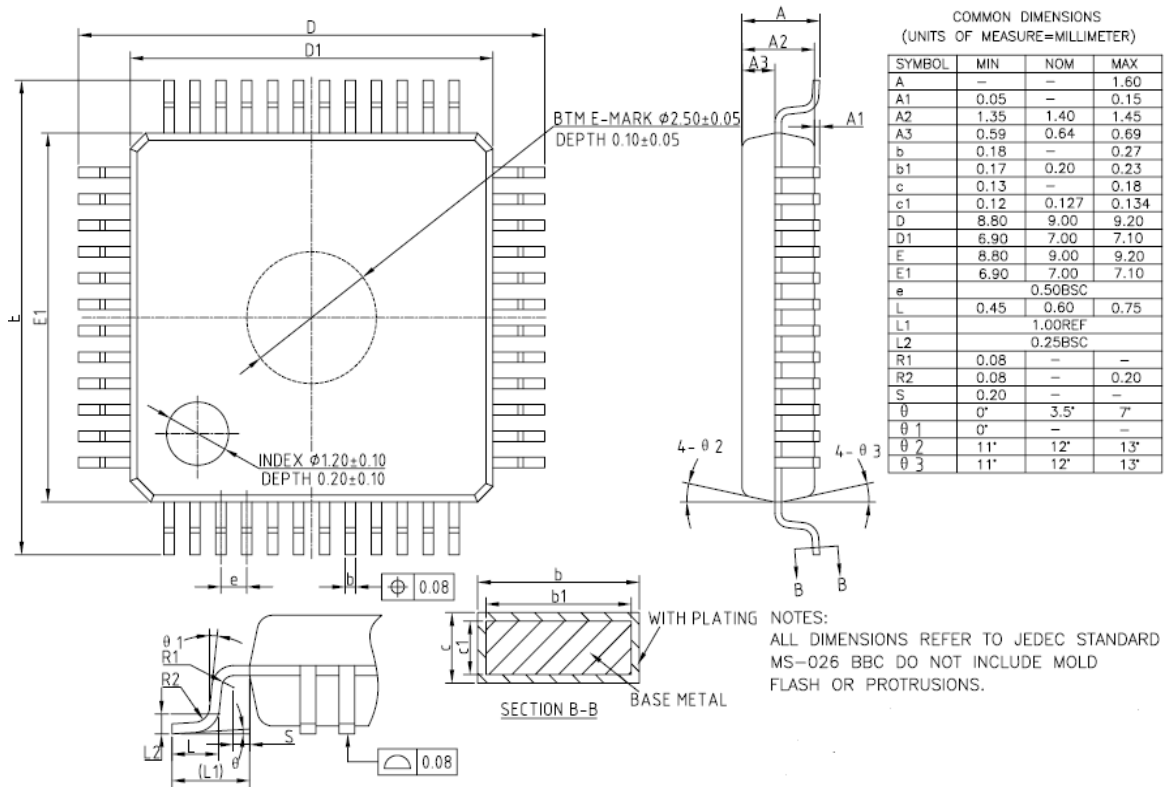


COMMON DIMENSIONS  
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	—	0.27
b1	0.17	0.20	0.23
c	0.13	—	0.18
c1	0.12	0.127	0.134
D	11.80	12.00	12.20
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
e	0.50BSC		
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
R1	0.08	—	—
R2	0.08	—	0.20
S	0.20	—	—
$\theta$	0°	3.5°	7°
$\theta$ 1	0°	—	—
$\theta$ 2	11°	12°	13°
$\theta$ 3	11°	12°	13°

NOTES:  
ALL DIMENSIONS MEET JEDEC STANDARD  
MS-026 BCD DO NOT INCLUDE MOLD  
FLASH OR PROTRUSIONS.

## 23.2 HT5009:LQFP48



## X-ON Electronics

Largest Supplier of Electrical and Electronic Components

*Click to view similar products for* [Power Management Specialised - PMIC category:](#)

*Click to view products by* [HiTrend Tech manufacturer:](#)

Other Similar products are found below :

[P9145-I0NQGI](#) [SLG7NT4192VTR](#) [AS3729B-BWLM](#) [TLE92623QXV33XUMA1](#) [LNBH25SPQR](#) [ADP5080ACBZ-1-RL](#)  
[MC32PF3000A6EP](#) [MB39C831QN-G-EFE2](#) [MAX9959DCCQ+D](#) [MAX1932ETC+T](#) [MAX1856EUB+T](#) [STNRG011TR](#)  
[IRPS5401MXI03TRP](#) [S6AE102A0DGN1B200](#) [MMPF0100FDAEP](#) [MCZ33903DS5EK](#) [S6AE101A0DGNAB200](#) [MCZ33903DS3EK](#)  
[NCP6924CFCHT1G](#) [MAX17117ETJ+](#) [L9916](#) [L9915-CB](#) [MCZ33904D5EK](#) [MCZ33905DS3EK](#) [MMPF0100FCANES](#) [MCZ33905DD3EK](#)  
[MMPF0100FBANES](#) [WM8325GEFL/V](#) [MCZ33903DP5EK](#) [MCZ33905DS5EK](#) [MCZ33903D3EK](#) [MCZ33903DD5EK](#) [ADN8835ACPZ-R7](#)  
[MCZ33903DP5EKR2](#) [MCZ33903D5EK](#) [MCZ33903DD3EK](#) [MMPF0100FAAZES](#) [SLG7NT4198V](#) [MIC5164YMM](#) [P9180-00NHGI](#)  
[NCP6914AFCAT1G](#) [TLE9261QX](#) [TEA1998TS/1H](#) [MAX881REUB+T](#) [TLE9262QX](#) [TLE8880TN](#) [MAX8520ETP+T](#) [SLG7NT4083V](#)  
[ADP1031ACPZ-1-R7](#) [ADP1031ACPZ-2-R7](#)