

# HT502X 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	2016-08-12		初版 1. 256K flash+1K information,32k RAM; 2. flash/info 分区操作; 3. pmu 模块, Vsys 检测扩冲为 16 档可选, vbat 允许 5v 输入; 4. 增加随机数发生模块 5. 增加 3 个外部中断 (INT7/8/9) 6. 增加 1 路 ADCIN3 7. 红外调制增加 HRC 时钟源 8. 增加 ECC 模块
V1.1	2016-10-27		9. 去掉 ADCIN3 相关描述 10. PIN27\28 去掉 VLCD 描述 11. 双版块切换详见应用笔记 12. WDT 不受 JTAGWDTEN 控制, 详见 FAQ 13. 增加说明 TEST pin 上升沿仅复位 DP 模块 14. 增加 Vsys 可进入低功耗说明 15. 增加 ECC256 功能, 说明详见应用笔记 16. TX 使能不放出, 内部始终使能有效 17. 修改 DMA 相关描述 18. 修改 EMU 相关描述 19. 修改同步缓存相关描述 20. 修改 SAG\PEAK 相关描述, 修改 Peak、Sag 寄存器名, 使之对应其新设计的定义。 21. 修改 EMUIE 增加的中断标志寄存器, 使之与中断使能 bit 位对应。 22. 按照 B 版规格修改“引脚定义”关于模拟滤波 2us 的描述, 去掉 uart0~5 的模拟滤波 2us 描述 23. 修改 uart 章节, 去掉最大波特率 115200 描述, 增加最高波特率达到 1Mbps 以上。 24. 修改“引脚定义”说明: SW 模式下 PB13(SWIO)/PB15(SWCLK)开漏功能关闭。 25. 修改 EMU“同步缓存”描述为“随频采样 ADC 缓存”。 26. 修改 PMU 模块 VDETPCFG 相关低功耗下电源检测分时开启检测时间为固定值。
V1.20	2016-11-9		1. 修改 LCD 显示对比度 VRSEL 的值; 2. 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,增加 bit8FCSET2 描述。 3. adc 输入电压的极限最大值
V1.21	2017-5-22		1. lowFreq_ise1 位定义修改 2. Pstart/Qstar/Sstart/QPHSCAL 寄存器说修改

V1.22	2017-8-18		<ol style="list-style-type: none"> <li>1. ECC 章节 register 笔误, 及寄存器增加详细说明。</li> <li>2. 修改 Flash 分区框图 B9 区大小为 31K。</li> <li>3. EMUIE 修改过零丢失/Buffer Full 的位置</li> </ol>
V1.23	2018-4-17		<ol style="list-style-type: none"> <li>1. Timer 章节举例定时最大周期公式计算笔误修正</li> <li>2. EMUCTRL 寄存器位笔误修正</li> <li>3. 随频采样 ADC 描述修正</li> <li>4. 修正 2.5 小结写保护寄存器列表 GPIO 的功能复位值</li> <li>5. 修正 UART 波特率计算公式描述最大波特率到 1M</li> <li>6. 修正其他章节笔误</li> </ol>
V1.24	2018-6-14		<ol style="list-style-type: none"> <li>1. IO 写保护取消</li> <li>2. 引脚定义 SCANIN1 有误, 改为 JTAGWDTEN</li> <li>3. 删除 JTAGSTA 寄存器 bit2</li> </ol>
V1.25	2018-8-21		<ol style="list-style-type: none"> <li>1. 修正 ADCCON CFP 描述</li> <li>2. 细化校表时 HFCONST 计算描述</li> <li>3. 各章节寄存器定义格式修正统一及部分内容补充</li> <li>4. 修正 RANDSTR bit15 定义笔误</li> </ol>
V1.26	2019-3-15		<ol style="list-style-type: none"> <li>1. 添加 HT5029B 引脚图和封装</li> </ol>
V1.27	2019-5-8		<ol style="list-style-type: none"> <li>1. 统一 OVCyc 和 UdetCyc 为 OVCyc</li> <li>2. 修正捕获中断 CMPIE 为 CAPIE</li> <li>3. 修正 uart 波特率最大到 1Mbps 的描述, 最大到 115200bps</li> </ol>
V1.28	2019-7-29		<ol style="list-style-type: none"> <li>1. 添加 HT5029 管脚图、引脚定义和封装</li> </ol>
V1.29	2019-8-8		<ol style="list-style-type: none"> <li>1. 修正 RTC 模块 DFi 寄存器描述, 由 23bit 改为 21bit</li> <li>2. 添加完善 hold 和 sleep 功耗范围</li> <li>3. 修正 tbs 分时开启功耗说明、添加 adc 通道功耗说明</li> <li>4. 修正 uart 波特率最大到 1Mbps</li> </ol>
V1.30	2019-11-18		<ol style="list-style-type: none"> <li>1. 修正部分编辑错误</li> <li>2. 1.2.1 基本特点章节增加 48pin 封装描述</li> <li>3. 修改 HRC 频率描述错误</li> </ol>
V1.31	2020-04-21		<ol style="list-style-type: none"> <li>1. 修正引脚定义 SCANIN1</li> <li>2. 删除 HT5029B 引脚图和封装</li> </ol>
V1.32	2020-6-9	hma	<ol style="list-style-type: none"> <li>1. 修改 24.6 章节 lrc 参数频率范围描述</li> <li>2. 修正 IO 口驱动能力典型值描述</li> </ol>
V1.33	2020-7-17	hma	<ol style="list-style-type: none"> <li>1. 修正 rtc 章节概述 tout 分频和寄存器 tout 位个数</li> <li>2. 删除 gpio 章节概述中的 gpio 驱动描述, 详细见 DC 参数</li> </ol>
V1.34	2020-8-21	hshan	<ol style="list-style-type: none"> <li>1. 删除引脚定义 VLCD0-3“需外接滤波电容 4.7uF”说明</li> <li>2. 修正 CTRLBYFLASH 寄存器描述</li> </ol>
V1.35	2020-11-24	Hshan	<ol style="list-style-type: none"> <li>1. 修正 IO 口驱动能力典型值描述</li> <li>2. 修正 hold 模式常温下最大功耗值</li> <li>3. 修正数字引脚最大输入电压值</li> </ol>
V1.36	2022-2-7	Zwding/ ysyang	<ol style="list-style-type: none"> <li>1. 删除原 5.4 调试模块复位说明部分</li> <li>2. 补充推荐调试端口说明 4</li> <li>3. 修正 11.6 章节关于 LCD buf 个数描述</li> <li>4. 修正串口模块 SREL 寄存器设置值范围</li> </ol>

			<ol style="list-style-type: none"><li>5. 删除关于 SPCFG，替换为 SPICON</li><li>6. 修改无功相位补偿寄存器推荐值</li><li>7. 修改 PMU 模块中关于 POR_DET、LBOR_DET 歧义描述</li><li>8. 增加寄存器说明详细目录</li></ol>
V1.37	2022-5-17	ysyang	<ol style="list-style-type: none"><li>1. 增加丝印说明</li><li>2. 增加包装托盘装的尺寸示意图</li></ol>

## 目录

目录 .....	5
<b>1 HT502X 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 框图 .....	13
1.5 引脚排列 .....	14
1.6 引脚定义 .....	18
<b>2 存储器模块 .....</b>	<b>24</b>
2.1 概述 .....	24
2.2 存储器映射图 .....	25
2.3 FLASH 操作 .....	26
2.3.1 Flash 的读保护 .....	26
2.3.2 Code Flash 的操作说明 .....	26
2.3.3 256K Code Flash 操作说明 .....	27
2.3.4 Information Block 的操作说明 .....	28
2.4 FLASH 控制功能 .....	28
2.5 写保护寄存器列表 .....	30
2.6 特殊功能寄存器列表 .....	31
2.7 特殊功能寄存器说明 .....	32
<b>3 时钟单元 .....</b>	<b>41</b>
3.1 时钟分类 .....	41
3.2 时钟框图 .....	41
3.3 时钟停振检测框图 .....	43
3.4 时钟说明 .....	43
3.4.1 内部低频 RC 时钟 (Flrc) .....	43
3.4.2 内部高频 RC 时钟 (Fhrc) .....	43
3.4.3 外部低频晶振时钟 (Fosc) .....	43
3.4.4 内部 PLL 时钟 (Fpll) .....	44
3.4.5 时钟安全机制 .....	44
3.4.6 时钟异常状态处理 .....	45
3.5 特殊功能寄存器列表 .....	46
3.6 特殊功能寄存器说明 .....	47
<b>4 电源单元 .....</b>	<b>57</b>
4.1 概述 .....	57

4.2	框图 .....	58
4.3	电源单元详细功能说明 .....	59
4.3.1	电源切换 .....	59
4.3.2	电源实时监测 .....	59
4.3.3	内建 1.5V 电源 .....	59
4.3.4	BOR 检测功能(BOR_DET) .....	60
4.3.5	系统电源检测功能(VSYS_DET) .....	62
4.3.6	低电压检测功能(LVD_DET) .....	63
4.3.7	VSYS_DET, BOR_DET, LVD_DET 分时检测时序 .....	63
4.3.8	电池防钝化功能 .....	63
4.4	特殊功能寄存器列表 .....	64
4.5	特殊功能寄存器说明 .....	64
<b>5</b>	<b>调试支持 .....</b>	<b>72</b>
5.1	概况 .....	72
5.2	SW 引脚分布 .....	72
5.3	SW 引脚使用说明 .....	72
5.4	推荐调试端口 .....	73
<b>6</b>	<b>工作模式 .....</b>	<b>74</b>
6.1	芯片电源域分配 .....	74
6.2	工作模式 .....	74
6.3	睡眠模式 (SLEEP) .....	75
6.3.1	SLEEP 模式下各模块开关 .....	75
6.3.2	SLEEP 模式下的唤醒 .....	75
6.3.3	从 SLEEP 模式唤醒后的唤醒方式确认 .....	76
6.3.4	进入 SLEEP 模式 .....	76
6.4	待机模式 (HOLD) .....	77
6.4.1	进入 HOLD 模式 .....	77
6.4.2	模式转换图 .....	77
6.5	特殊功能寄存器列表 .....	78
6.6	特殊功能寄存器说明 .....	78
<b>7</b>	<b>GPIO 模块 .....</b>	<b>82</b>
7.1	概述 .....	82
7.2	芯片引脚结构说明 .....	83
7.3	芯片引脚描述 .....	83
7.4	I/O 端口基地址列表 .....	84
7.5	特殊功能寄存器说明 .....	85
<b>8</b>	<b>中断模块 .....</b>	<b>91</b>
8.1	中断向量说明 .....	91
8.2	中断使能和禁止 .....	92
8.2.1	中断使能和禁止相关寄存器列表 .....	93
8.2.2	中断使能和禁止相关寄存器说明 .....	93

8.3	外部中断特殊功能寄存器列表.....	94
8.4	外部中断特殊功能寄存器说明.....	95
<b>9</b>	<b>RESET 模块.....</b>	<b>99</b>
9.1	复位优先级.....	99
9.2	复位说明.....	100
9.2.1	上电复位.....	100
9.2.2	低电压检测复位.....	101
9.2.3	外部引脚复位.....	101
9.2.4	掉电复位.....	101
9.2.5	看门狗复位.....	102
9.2.6	软复位.....	102
9.2.7	调试复位.....	103
9.2.8	唤醒复位.....	103
9.3	特殊功能寄存器列表.....	104
9.4	特殊功能寄存器说明.....	104
<b>10</b>	<b>UART/7816 通讯模块.....</b>	<b>110</b>
10.1	功能说明.....	110
10.2	波特率计算.....	111
10.3	串口通讯模式说明.....	111
10.3.1	方式1.....	111
10.3.2	方式2.....	112
10.3.3	方式3.....	112
10.3.4	方式4.....	113
10.4	7816 接收和发送.....	114
10.4.1	7816 数据发送.....	114
10.4.2	7816 数据接收.....	114
10.4.3	7816 通讯示意图.....	115
10.5	特殊功能寄存器列表.....	118
10.6	特殊功能寄存器说明.....	119
<b>11</b>	<b>LCD 模块.....</b>	<b>126</b>
11.1	概述.....	126
11.2	LCD 与 GPIO 口复用表.....	126
11.3	LCD 框图.....	127
11.4	内部电阻分压驱动.....	128
11.5	输出波形.....	128
11.6	LCD 显示操作.....	132
11.7	特殊功能寄存器列表.....	132
11.8	特殊功能寄存器说明.....	132
<b>12</b>	<b>WDT 模块.....</b>	<b>137</b>
12.1	概述.....	137
12.2	工作模式.....	137

12.3	特殊功能寄存器列表.....	138
12.4	特殊功能寄存器说明.....	138
<b>13</b>	<b>定时器模块.....</b>	<b>140</b>
13.1	定时器单元概述.....	140
13.2	周期定时功能.....	140
13.3	PWM 功能.....	141
13.4	捕获功能.....	143
13.5	事件计数功能.....	143
13.6	中断功能.....	144
13.6.1	定时周期中断.....	144
13.6.2	捕获中断.....	144
13.6.3	比较中断.....	144
13.6.4	事件计数功能.....	144
13.7	特殊功能寄存器列表.....	145
13.8	特殊功能寄存器说明.....	145
<b>14</b>	<b>SPI 模块.....</b>	<b>150</b>
14.1	概述.....	150
14.2	详细功能说明.....	150
14.2.1	SPI 主要特征.....	150
14.2.2	SPI 接口模块框图.....	151
14.2.3	SPI 接口传输格式.....	151
14.2.4	主机模式传输格式.....	151
14.2.5	从机模式传输格式.....	152
14.2.6	中断功能.....	154
14.3	特殊功能寄存器列表.....	154
14.4	特殊功能寄存器说明.....	154
<b>15</b>	<b>I2C 模块.....</b>	<b>158</b>
15.1	概述.....	158
15.2	框图.....	158
15.3	功能描述.....	158
15.3.1	起始条件和停止条件.....	159
15.3.2	数据传输格式.....	159
15.3.3	时钟同步和数据仲裁.....	160
15.3.4	7 位地址格式.....	161
15.3.5	串行时钟生成.....	161
15.3.6	中断生成.....	161
15.3.7	传输模式.....	162
15.4	特殊功能寄存器列表.....	171
15.5	特殊功能寄存器说明.....	171
<b>16</b>	<b>RTC 模块.....</b>	<b>174</b>
16.1	概述.....	174

16.2	功能描述.....	174
16.3	时钟校正.....	174
16.4	RTC 补偿系数寄存器和 INFO FLASH 对应关系.....	175
16.5	时间和万年历.....	175
16.6	中断功能.....	176
16.7	RTC 指示寄存器读写流程.....	176
16.7.1	读取 RTC 指示寄存器流程.....	176
16.7.2	写入 RTC 指示寄存器流程.....	177
16.8	校时记录.....	178
16.9	辅助 RTC.....	178
16.10	特殊功能寄存器列表.....	178
16.11	特殊功能寄存器说明.....	179
<b>17</b>	<b>TBS 模块.....</b>	<b>201</b>
17.1	概述.....	201
17.2	功能描述.....	201
17.3	特殊功能寄存器列表.....	202
17.4	特殊功能寄存器说明.....	202
<b>18</b>	<b>AES&amp;GHASH&amp;RAND 模块.....</b>	<b>211</b>
18.1	AES&GHASH&RAND 概述.....	211
18.2	框图.....	211
18.3	特殊功能寄存器列表.....	211
18.4	特殊功能寄存器说明.....	213
<b>19</b>	<b>ECC256 模块.....</b>	<b>227</b>
19.1	概述.....	227
19.2	功能描述.....	227
19.3	ECC 运算加速模块.....	228
19.4	特殊功能寄存器列表.....	230
19.5	特殊功能寄存器说明.....	230
<b>20</b>	<b>DMA 功能.....</b>	<b>238</b>
20.1	概述.....	238
20.2	功能描述.....	238
20.3	DMA 数据请求源列表.....	238
20.4	DMA 数据传输说明.....	240
20.5	特殊功能寄存器列表.....	241
20.6	特殊功能寄存器说明.....	241
<b>21</b>	<b>按键扫描模块 (KEYSCAN).....</b>	<b>247</b>
21.1	概述.....	247
21.2	功能描述.....	247
21.3	特殊功能寄存器列表.....	248
21.4	特殊功能寄存器说明.....	248

<b>22</b>	<b>CORTEX-M0 内核简要说明.....</b>	<b>250</b>
22.1	概述 .....	250
22.2	系统定时器 SYSTICK .....	250
22.3	中断优先级说明.....	250
22.4	CMSIS 函数说明.....	250
<b>23</b>	<b>EMU 模块.....</b>	<b>252</b>
23.1	概述 .....	252
23.2	功能描述.....	252
23.2.1	模数转换器.....	253
23.2.2	ADC 采样输出和功率波形输出.....	253
23.2.3	有功功率、无功功率和视在功率.....	253
23.2.4	有效值.....	254
23.2.5	电压线频率.....	255
23.2.6	起动/潜动.....	255
23.2.7	功率反向指示.....	255
23.2.8	直流偏置校正.....	255
23.2.9	电能脉冲输出.....	256
23.2.10	窃电检测.....	257
23.2.11	中断源.....	258
23.2.12	PEAK / SAG 功能.....	259
23.2.13	OVI 过压过流检测功能.....	260
23.2.14	掉零线计量模式(Loss of Voltage).....	261
23.2.15	随频采样 ADC 波形缓存.....	261
23.2.16	Q 能量脉冲通道复用 P2.....	262
23.3	计量参数寄存器列表.....	263
23.4	计量参数寄存器说明.....	265
23.5	校表参数寄存器列表.....	277
23.6	校表参数寄存器说明.....	280
23.7	校表过程.....	319
<b>24</b>	<b>电气规格.....</b>	<b>322</b>
24.1	DC 参数.....	322
24.2	极限参数.....	323
24.3	功耗参数.....	323
24.4	计量 ADC 参数.....	325
24.5	TBS 模块 ADC 参数.....	326
24.6	内部 RC 模拟参数.....	328
<b>封装</b>	<b>.....</b>	<b>329</b>
	芯片封装.....	329
	丝印说明.....	331
	包装尺寸.....	332

# 1 HT502X 概述

## 1.1 简介

HT502X 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、256K flash、32K 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：2.2V~5.5V
- 工作温度范围：-40°C~85°C
- 无铅绿色封装：LQFP100：HT5023、LQFP80：HT5025、LQFP64：HT5027、LQFP48：HT5029B、LQFP48：HT5029

### 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，包含嵌套中断向量控制器 NVIC，以及可扩展的调试技术
- 存储器资源：256K Flash+1K Information Block、32K 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 两个电源内部切换功能
- 内置 9.8MHz 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，内部集成晶体所需电阻、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：外部晶振 32.768KHz 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±0.5 度
- LCD 驱动电阻分压方式，支持 4COM，6COM，8COM 的 LCD 显示，SEG 接口最多支持 54 个
- 最多支持 6 路 UART，其中 2 路复用硬件 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/解密加速模块，支持 AES256/192/128

- 内置 ECC 加密/解密加速模块，支持 ECC256/244/192
- 内置 3 个通道 DMA 功能
- 内置按键扫描功能，最大支持 4\*4 键盘扫描模式
- 支持 I2C, SPI, TIMER CC & PWM

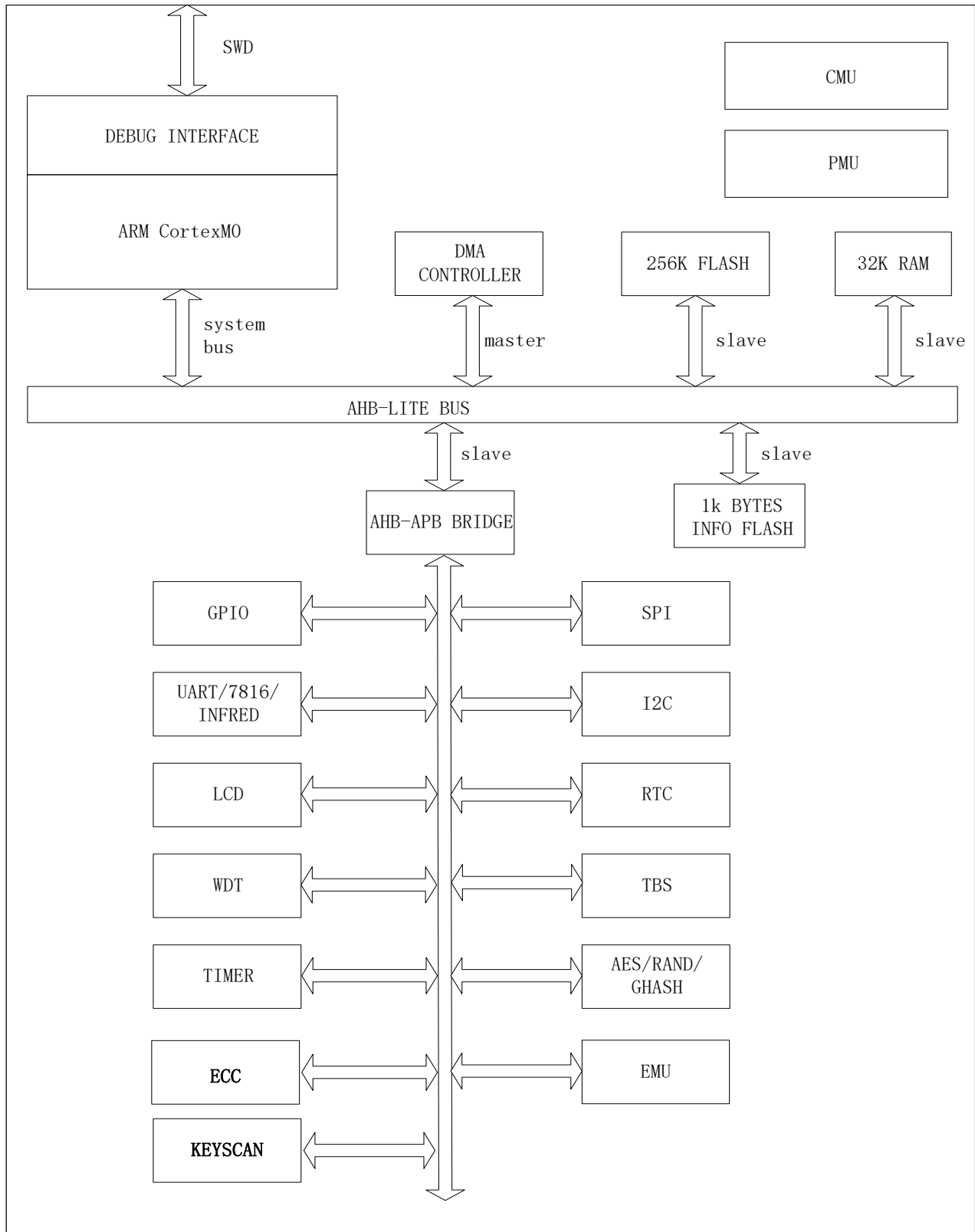
## 1.2.3 电能计量

- 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，用于掉零线、掉火线的防窃电计量，低功耗连续计量，系统功耗优于 650uA；支持常数计量模式，适用于电池辅助掉零线防窃电计量功能
- 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  $\pm 10\text{ppm}/^\circ\text{C}$

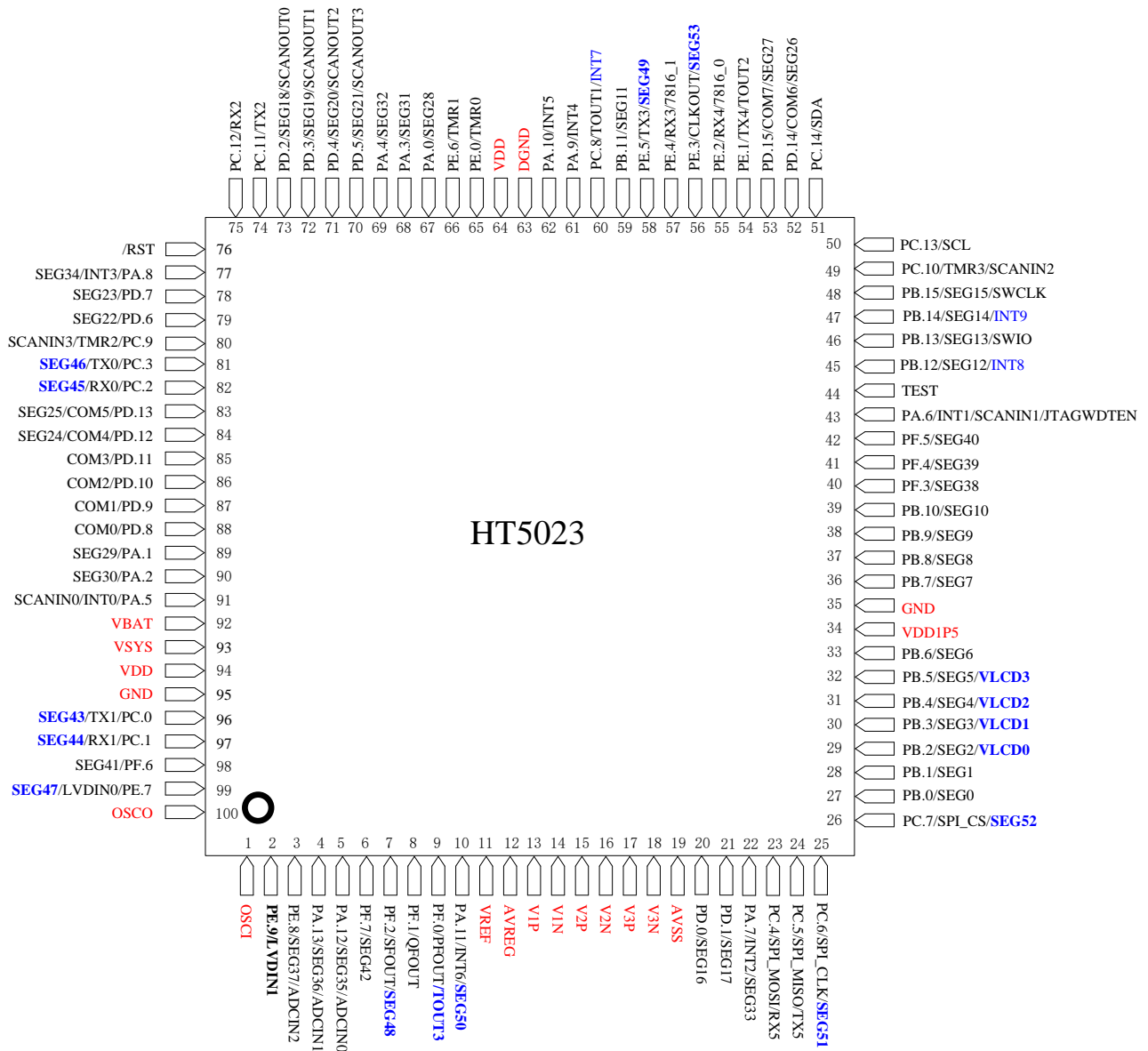
## 1.3 缩略语

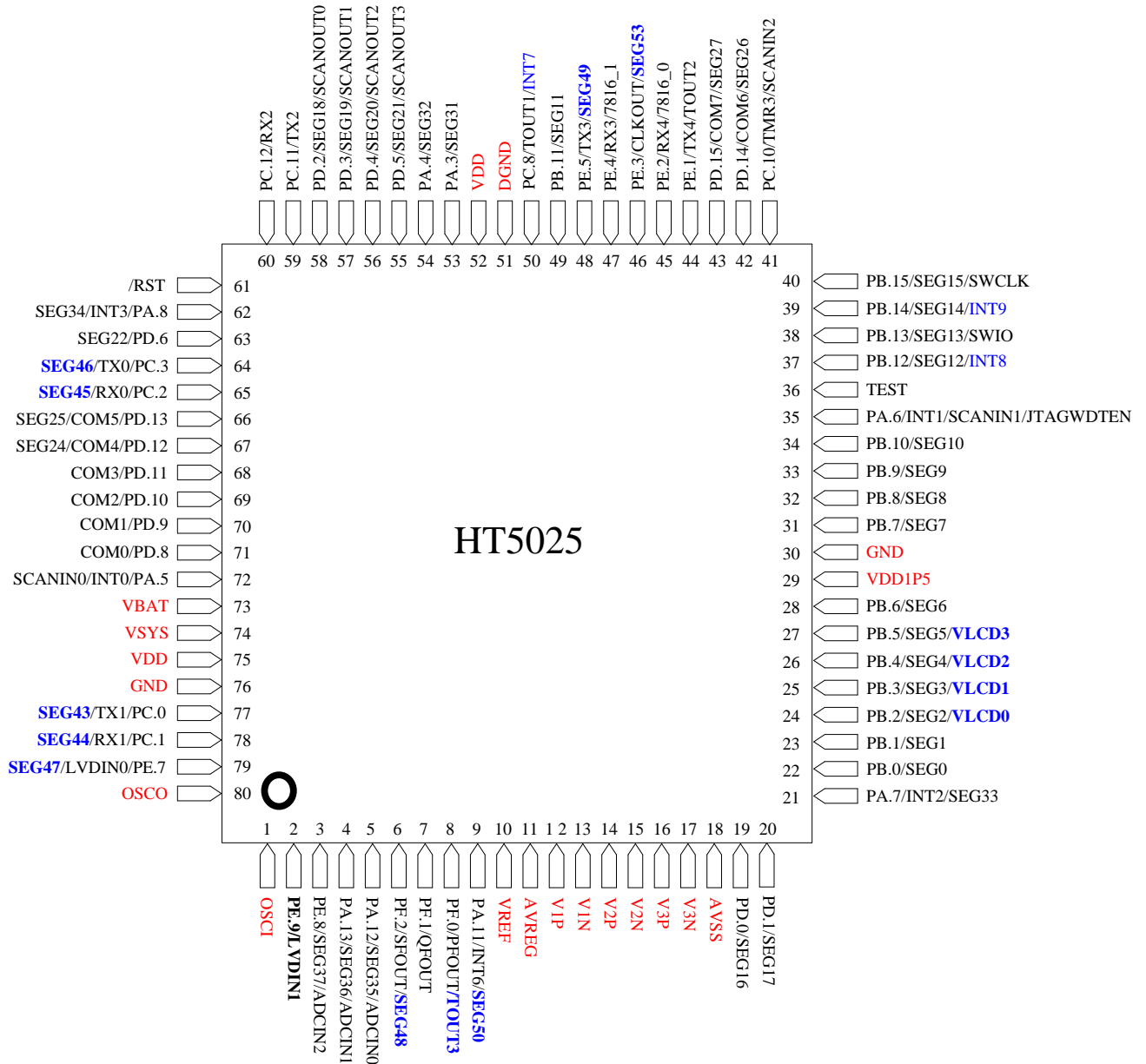
缩略语	英文原文	中文含义
WDT	Watch Dog Timer	看门狗
GPIO	General Purpose IO	通用 I/O
TBS	Temperature Battery Sensor	温度&电池传感器
LVD	Low Voltage Detect	低电压检测
POR	Power On Reset	上电复位
BOR	Brown Out Reset	掉电复位
WKR	Wakeup Reset	唤醒复位
EMU	Energy Mearsurment Unit	电能计量单元
PMU	Power Management Unit	系统电源管理单元
CMU	Clock Management Unit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

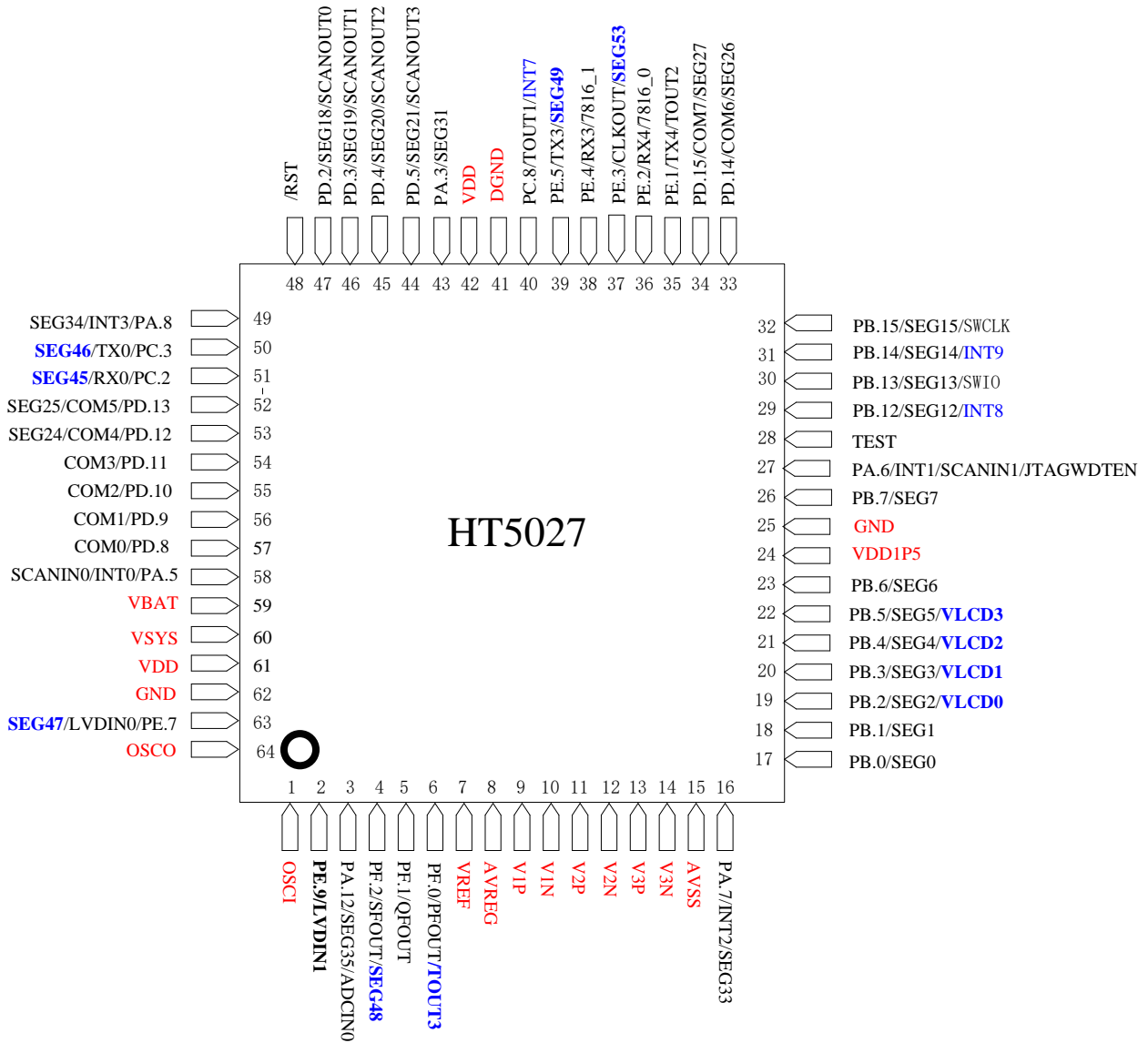
### 1.4 框图

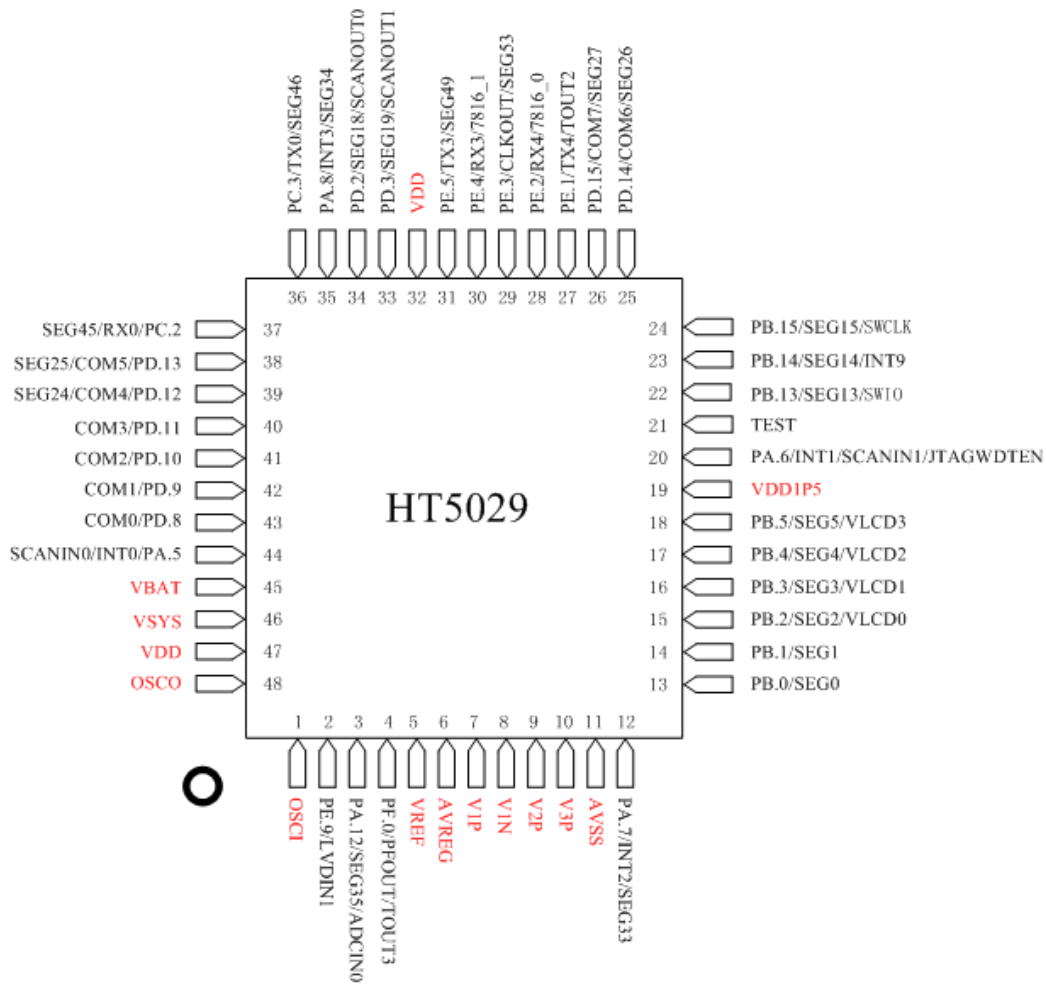


## 1.5 引脚排列









## 1.6 引脚定义

100 PIN	80 PIN	64 PIN	48 PIN (HT5029)	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	1	1	OSCI	I			32KHz 时钟输入
2	2	2	2	PE.9	I/O	LVDIN1		GPIO\外部电源检测
3	3			PE.8	I/O	SEG37	ADCIN2	GPIO\LCD_SEG 驱动输出\外部电压信号采样输入 2
4	4			PA.13	I/O	SEG36	ADCIN1	GPIO\LCD_SEG 驱动输出\外部电压信号采样输入 1
5	5	3	3	PA.12	I/O	SEG35	ADCIN0	GPIO\LCD_SEG 驱动输出\外部电压信号采样输入 0
6				PF.7	I/O	SEG42	SEG42	GPIO\LCD SEG 驱动输出
7	6	4		PF.2	I/O	SFOUT	SEG48	GPIO\电能脉冲输出(默认视在脉冲输出, 可通过 EMCON 配置为其它类型脉冲输出)\LCD)SEG 驱动输出 输出驱动 5mA
8	7	5		PF.1	I/O	QFOUT	QFOUT	GPIO\电能脉冲输出(默认无功脉冲输出, 可通过 EMCON 配置为其它类型脉冲输出) 输出驱动 5mA
9	8	6	4	PF.0	I/O	PFOUT	TOUT3	GPIO\电能脉冲输出(默认有功脉冲输出, 可通过 EMCON 配置为其它类型脉冲输出)\秒脉冲输出 输出驱动 5mA
10	9			PA.11	I/O	INT6	SEG50	GPIO\外部中断口(滤波 2us)\LCD_SEG 驱动输出, 输出驱动 5mA
11	10	7	5	VREF	OUT			1.185V 计量基准电压输出,需外接滤波电容 0.1uF&1uF
12	11	8	6	AVREG	P			2.8V 内部模拟电源输出, 需外接滤波电容 0.1uF&10uF
13	12	9	7	VIP	I			电流通道 1 模拟信号输入端(正), 内部 ESD 保护, 最大信号±800mVp
14	13	10	8	VIN	I			电流通道 1 模拟信号输入端(负), 内部 ESD 保护, 最大信

								号±800mVp
15	14	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
16	15	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
17	16	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
18	17	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
19	18	15	11	AVSS	G			模拟地
20	19			PD.0	I/O	SEG16		GPIO\LCD_SEG 驱动输出, 输出驱动 5mA
21	20			PD.1	I/O	SEG17	SEG17	GPIO\LCD_SEG 驱动输出
22	21	16	12	PA.7	I/O	INT2	SEG33	大功耗 GPIO\外部中断输入 (滤波 2us)\LCD_SEG 驱动输出 输出驱动 30mA
23				PC.4	I/O	SPI_MOSI	RX5	GPIO\SPI 驱动数据输入\串口接收端 输出驱动 5mA
24				PC.5	I/O	SPI_MISO	TX5	GPIO\SPI 驱动数据输出\串口发送端 输出驱动 5mA
25				PC.6	I/O	SPI_CLK	SEG51	GPIO\SPI 驱动时钟信号 \LCD_SEG 驱动输出 输出驱动 5mA
26				PC.7	I/O	SPI_CS	SEG52	GPIO SPI 驱动片选信号 \LCD_SEG 驱动输出 输出驱动 5mA
27	22	17	13	PB.0	I/O	SEG0		GPIO\LCD_SEG 驱动输出
28	23	18	14	PB.1	I/O	SEG1		GPIO\LCD_SEG 驱动输出
29	24	19	15	PB.2	I/O	SEG2	VLCD0	GPIO\LCD_SEG 驱动输出 \LCD 电阻分压模式 bias 电压输出
30	25	20	16	PB.3	I/O	SEG3	VLCD1	GPIO\LCD_SEG 驱动输出 \LCD 电阻分压模式 bias 电压输出
31	26	21	17	PB.4	I/O	SEG4	VLCD2	GPIO\LCD_SEG 驱动输出 \LCD 电阻分压模式 bias 电压

								输出
32	27	22	18	PB.5	I/O	SEG5	VLCD3	GPIO\LCD_SEG 驱动输出 \LCD 电阻分压模式 bias 电压输出
33	28	23		PB.6	I/O	SEG6	SEG6	GPIO\LCD_SEG 驱动输出
34	29	24	19	VDD1P5	P			内部 1.5V 输出, 需外接 0.1uF 滤波电容
35	30	25		GND	G			模拟地
36	31	26		PB.7	I/O	SEG7	SEG7	GPIO\LCD_SEG 驱动输出
37	32			PB.8	I/O	SEG8	SEG8	GPIO\LCD_SEG 驱动输出
38	33			PB.9	I/O	SEG9	SEG9	GPIO\LCD_SEG 驱动输出
39	34			PB.10	I/O	SEG10	SEG10	GPIO\LCD_SEG 驱动输出
40				PF.3	I/O	SEG38	SEG38	GPIO\LCD_SEG 驱动输出
41				PF.4	I/O	SEG39	SEG39	GPIO\LCD_SEG 驱动输出
42				PF.5	I/O	SEG40	SEG40	GPIO\LCD_SEG 驱动输出
43	35	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。
44	36	28	21	TEST	I			测试引脚, 滤波 2us 当此 PIN 先于 JTAGWDTEN 为 低电平状态时, 系统进入测试 模式。
45	37	29		PB.12	I/O	SEG12	INT8	GPIO\LCD_SEG 驱动输出\外 部中断输入
46	38	30	22	PB.13	I/O	SEG13	SWIO	GPIO\LCD_SEG 驱动输出\仿 真模式数据信号 SWIO
47	39	31	23	PB.14	I/O	SEG14	INT9	GPIO\LCD_SEG 驱动输出\外 部中断输入
48	40	32	24	PB.15	I/O	SEG15	SWCLK	GPIO\LCD_SEG 驱动输出\仿 真模式时钟信号 SWCLK
49	41			PC.10	I/O	TMR3	SCANIN2	GPIO\定时器模块外部端口\键 盘扫描列线扫描输入口线 输出驱动 5mA
50				PC.13	I/O	SCL	SCL	GPIO\IIC 模块时钟信号 IIC- CLK
51				PC.14	I/O	SDA	SDA	GPIO\IIC 模块数据信号 IIC-IO
52	42	33	25	PD.14	I/O	COM6	SEG26	GPIO\LCD_COM\LCD_SEG 驱动输出

53	43	34	26	PD.15	I/O	COM7	SEG27	GPIO\LCD_COM\LCD_SEG 驱动输出
54	44	35	27	PE.1	I/O	TX4	TOUT2	GPIO\串口发送端\秒脉冲输出 输出驱动 5mA
55	45	36	28	PE.2	I/O	RX4	7816_0	GPIO\串口接收端端\7816 驱动 端口
56	46	37	29	PE.3	I/O	CLKOUT	SEG53	GPIO\内部时钟分频输出端 (ESAM/CARD 时 钟 输 出)\LCD_SEG驱动输出
57	47	38	30	PE.4	I/O	RX3	7816_1	GPIO\串口接收端\ 7816驱动端 口
58	48	39	31	PE.5	I/O	TX3	SEG49	GPIO\串口发送端\LCD_SEG 驱动输出
59	49			PB.11	I/O	SEG11	SEG11	GPIO\LCD_SEG 驱动输出
60	50	40		PC.8	I/O	TOUT1	INT7	GPIO\秒脉冲输出 1\外部中断 输入
61				PA.9	I/O	INT4	INT4	GPIO\外部中断输出(滤波 2us)
62				PA.10	I/O	INT5	INT5	GPIO\外部中断输出(滤波 2us)
63	51	41		DGND	G			芯片数字地
64	52	42	32	VDD	P			内部电源输入, 提供内部数字 电源,必需由另一个 VDD 外部 连接供电
65				PE.0	I/O	TMR0	TMR0	GPIO\定时器模块外部端口
66				PE.6	I/O	TMR1	TMR1	GPIO\定时器模块外部端口
67				PA.0	I/O	SEG28	SEG28	GPIO\LCD_SEG 驱动输出
68	53	43		PA.3	I/O	SEG31	SEG31	GPIO\LCD_SEG 驱动输出
69	54			PA.4	I/O	SEG32	SEG32	GPIO\LCD_SEG 驱动输出
70	55	44		PD.5	I/O	SEG21	SCANOUT3	GPIO\LCD_SEG 驱动输出\键 盘扫描行扫描输出口线 输出驱动 5mA
71	56	45		PD.4	I/O	SEG20	SCANOUT2	GPIO\LCD_SEG 驱动输出\键 盘扫描行扫描输出口线 输出驱动 5mA
72	57	46	33	PD.3	I/O	SEG19	SCANOUT1	GPIO\LCD_SEG 驱动输出\键 盘扫描行扫描输出口线 输出驱动 5mA
73	58	47	34	PD.2	I/O	SEG18	SCANOUT0	GPIO\LCD_SEG 驱动输出\键 盘扫描行扫描输出口线 输出驱动 5mA
74	59			PC.11	I/O	TX2	TX2	GPIO\串口发送端

								输出驱动 5mA
75	60			PC.12	I/O	RX2	RX2	GPIO\串口接收端
76	61	48		/RST	I			复位信号（低电平有效）， 滤波 2us
77	62	49	35	PA.8	I/O	INT3	SEG34	大功耗 GPIO\外部中断输入(滤波 2us)\LCD_SEG 驱动输出 输出驱动 30mA
78				PD.7	I/O	SEG23	SEG23	GPIO\LCD_SEG 输出驱动 输出驱动 5mA
79	63			PD.6	I/O	SEG22	SEG22	GPIO\LCD_SEG 输出驱动 输出驱动 5mA
80				PC.9	I/O	TMR2	SCANIN3	GPIO\定时器模块外部端口\按钮扫描模块列扫描输入口线 输出驱动 5mA
81	64	50	36	PC.3	I/O	TX0	<b>SEG46</b>	GPIO\串口发送端\LCD_SEG 驱动输出
82	65	51	37	PC.2	I/O	RX0	<b>SEG45</b>	GPIO\串口接收端\LCD_SEG 驱动输出
83	66	52	38	PD.13	I/O	COM5	SEG25	GPIO\LCD_COM\LCD_SEG 驱动输出
84	67	53	39	PD.12	I/O	COM4	SEG24	GPIO\LCD_COM\LCD_SEG 驱动输出
85	68	54	40	PD.11	I/O	COM3	COM3	GPIO\LCD_COM 驱动输出
86	69	55	41	PD.10	I/O	COM2	COM2	GPIO\LCD_COM 驱动输出
87	70	56	42	PD.9	I/O	COM1	COM1	GPIO\LCD_COM 驱动输出
88	71	57	43	PD.8	I/O	COM0	COM0	GPIO\LCD_COM 驱动输出
89				PA.1	I/O	SEG29		GPIO\LCD_SEG 驱动输出
90				PA.2	I/O	SEG30		GPIO\LCD_SEG 驱动输出
91	72	58	44	PA.5	I/O	INT0	SCANIN0	GPIO\外部中断输入(滤波 2us)\ 键盘扫描列扫描输入口线
92	73	59	45	VBAT	P			电池供电输入
93	74	60	46	VSYS	P			系统电源输入
94	75	61	47	VDD	P			系统电源输出
95	76	62		GND	G			模拟地
96	77			PC.0	I/O	TX1	<b>SEG43</b>	大功耗 GPIO\串口发送端 \LCD_SEG 驱动输出 输出驱动 30mA
97	78			PC.1	I/O	RX1	<b>SEG44</b>	GPIO\串口接收端 \LCD_SEG 驱动输出
98				PF.6	I/O	SEG41	SEG41	GPIO\LCD_SEG 驱动输出

99	79	63		PE.7	I/O	LVDIN0	<b>SEG47</b>	GPIP\外部电源检测端口
100	80	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. 芯片引脚选择 GPIO 功能：

若方向寄存器配置为输出，开漏 OD 功能配置控制有效，上拉控制无效；

若方向寄存器配置为输入，开漏 OD 功能控制无效，上拉控制有效；

3. 复用功能为模拟功能：LVDIN<sub>x</sub>、LCD\_SEG\COM、POWIN、ADCIN<sub>x</sub>、VBAT,除此之外的其它复用功能为数字功能。

4. 芯片引脚选择复用功能：

复用功能的数字输出引脚都可配开漏功能(Open Drain)，上拉功能配置无效。

复用功能的数字输入引脚（除 RST/TEST/JTAGWDTEN，这三个引脚内部恒定上拉）都可配上拉功能，开漏功能；

若配置为模拟输入 OD 控制和上拉控制都无效；

5. 端口数据寄存器 PTDAT 说明及数据读取

1) 芯片引脚选择 GPIO 功能或复用数字功能

若方向寄存器配置为输出，PTDAT 读取值为寄存器设置值，不随外部 PIN 脚电平变化而变化；

若方向寄存器配置为输入，PTDAT 读取值为 PIN 脚状态值，反映外部 PIN 脚电平变化；

2) 芯片引脚选择复用模拟功能

PTDAT 相应 bit 位值，固定为 0

6. 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。

7. 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。

8. 两个 VDD 引脚外部需要连接在一起。

9. PB13/PB15 当软件配置为复用功能 2 时，为调试端口

10. PB13/PB15 上电后默认为调试端口，即复用功能 2

11. 当 TEST=0，且 JTAGWDTEN=0 时，则 PB13(SWIO)/PB15(SWCLK)恒定为 SW 调试端口，开漏功能关闭,配置开漏无效。

12. PE.0 上电后默认为复用功能 1，并且默认为 PFOUT 功能，PE.0 还可配置为复用功能 2 TOUT3（秒脉冲输出）；PF.1/PF.2 上电后默认为复用功能 1,并且分别默认为为 QFOUT/SFOUT 功能。

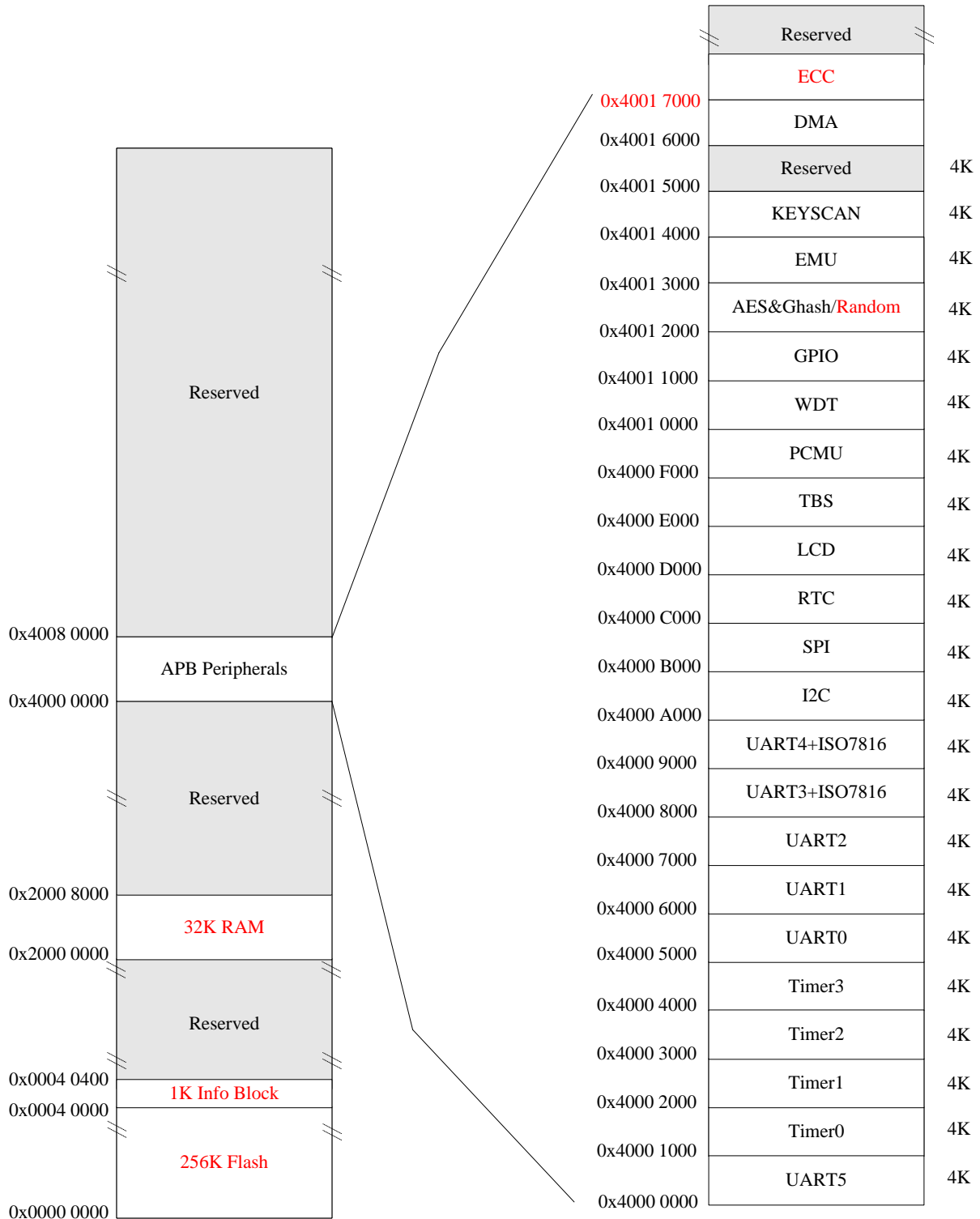
## 2 存储器模块

### 2.1 概述

HT502X 内置可编程高可靠 **256K Flash+1KInformationBlock** 和 **32KSRAM**。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：大小为 1K，1K 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，256K 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

对 256K CodeFlash 擦写可按照下图分成 11 个区块：

	Reserved
0x0004 0400	1K Info Block
0x0004 0000	CodeFlash_SectorB_1K
0x0003 FC00	CodeFlash_Sector9_31K
0x0003 8000	CodeFlash_Sector8_32K
0x0003 0000	CodeFlash_Sector7_32K
0x0002 8000	CodeFlash_Sector6_32K
0x0002 0000	CodeFlash_SectorA_1K
0x0001 8000	CodeFlash_Sector5_31K
0x0001 0000	CodeFlash_Sector4_32K
0x0000 8000	CodeFlash_Sector3_32K
0x0000 2000	CodeFlash_Sector2_24K
0x0000 0000	CodeFlash_Sector1_8K

对 CodeFlash 具体操作如下：

寄存器操作	对 Flash 的操作说明
FLASHLOCK = 0x7A68	可以对整个 256K CodeFlash 进行全擦写或页擦写，无需考虑任意 sector FSnLOCK 是否被解锁
<p>以下对 256K Flash 分区块进行解锁后，只能对解锁的区块进行页擦写，全擦无效。  <b>FLASHLOCK = 0x7A68 全擦写使能命令为最高覆盖等级，需要对 Flash 进行分区解锁操作时，FLASHLOCK 必需设置为非 0x7A68。</b></p>	
FS1LOCK = 0x7161	只能对 8K CodeFlash 0x0 0000 – 0x0 1FFF 进行页擦写
FS2LOCK = 0x7262	只能对 24K CodeFlash 0x0 2000- 0x0 7FFF 进行页擦写
FS3LOCK = 0x7363	只能对 32K CodeFlash 0x0 8000- 0x0 FFFF 进行页擦写

FS4LOCK = 0x7464	只能对 32K CodeFlash 0x10000- 0x1 7FFF 进行页擦写
FS5LOCK = 0x7565	只能对 31K CodeFlash 0x1 8000- 0x1 FBFF 进行页擦写
FSALOCK = 0x7A6A	只能对 1K CodeFlash 0x1 FC00 - 0x1 FFFF 进行页擦写
FS6LOCK = 0x7666	只能对 32K CodeFlash 0x2 0000- 0x2 7FFF 进行页擦写
FS7LOCK = 0x7767	只能对 32K CodeFlash 0x2 8000- 0x2 FFFF 进行页擦写
FS8LOCK = 0x7868	只能对 32K CodeFlash 0x3 0000- 0x3 7FFF 进行页擦写
FS9LOCK = 0x7969	只能对 31K CodeFlash 0x3 8000- 0x3 FBFF 进行页擦写
FSBLOCK = 0x7B6B	只能对 1K CodeFlash 0x3 FC00 - 0x3 FFFF 进行页擦写

### 2.3.3 256K Code Flash 操作说明

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，HT5X2X 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
```

(1) 对 256K 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
```

注：为保证字对齐操作：

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
```

(2) 对 256K 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
```

(3) 对 256K 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 256K 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
```

### 2.3.4 Information Block 的操作说明

Information Block 共 1K bytes (0x00040000~0x000403FF), 共 1 页, 1024 bytes/页, 其中存储有芯片出厂信息, 不建议进行擦写操作。

## 2.4 Flash 控制功能

HT5X2X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HT5X2X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

HT502X 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	AUTOR ELOAD	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
RTCST	=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	SYSCCLKCFG	0x0002	系统时钟配置寄存器（写保护）
0x08	JTAGSTA	0x0000	JTAG 状态寄存器（只读）
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）
0x1C	SYSCCLKDIV	0x0001	系统时钟分频寄存器（写保护）
0x24	CLKOUTSEL	0x0002	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	0x24A0	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
0x90	FLASHCON2	0x0000	Flash 访问控制寄存器 2（写保护）

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2（写保护）
0x204	MINR2	0x0000	分钟寄存器2（写保护）
0x208	HOURR2	0x0000	小时寄存器2（写保护）
0x20C	DAYR2	0x0001	天寄存器2（写保护）
0x210	MONTHR2	0x0001	月寄存器2（写保护）
0x214	YRR2	0x0000	年寄存器2（写保护）
0x218	WEEKR2	0x0001	星期寄存器2（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
0x08	JTAGSTA	R	0x0000	JTAG 状态寄存器
0x34	FLASHCON	R/W	0x0000	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
0x60	FS1LOCK	R/W	0x0000	FlashSector1 锁定寄存器
0x64	FS2LOCK	R/W	0x0000	FlashSector2 锁定寄存器
0x68	FS3LOCK	R/W	0x0000	FlashSector3 锁定寄存器
0x6C	FS4LOCK	R/W	0x0000	FlashSector4 锁定寄存器
0x70	FS5LOCK	R/W	0x0000	FlashSector5 锁定寄存器
0x74	FS6LOCK	R/W	0x0000	FlashSector6 锁定寄存器
0x78	FS7LOCK	R/W	0x0000	FlashSector7 锁定寄存器
0x7C	FS8LOCK	R/W	0x0000	FlashSector8 锁定寄存器
0x80	FS9LOCK	R/W	0x0000	FlashSector9 锁定寄存器
0x84	FSALOCK	R/W	0x0000	FlashSectorA 锁定寄存器
0x88	FSBLOCK	R/W	0x0000	FlashSectorB 锁定寄存器
0x90	FLASHCON2	R/W	0x0000	Flash 控制寄存器 2（写保护）

## 2.7 特殊功能寄存器说明

### WPREG(写保护寄存器)

<b>WPREG</b> (写保护寄存器)			<b>基地址: 0x4000F000</b>					
			<b>偏移地址: 00H</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>	WPREG[15:8]							
<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>	WPREG[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
WPREG[15:0]	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。 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。 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

### FLASHCON(Flash 控制寄存器 写保护)

<b>FLASHCON (写保护)</b> (Flash 控制寄存器)			<b>基地址: 0x4000F000</b>					
			<b>偏移地址: 34H</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	X	X	X	BUSY	FOP[1:0]	
<b>Write:</b>						X		
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
BUSY	<b>FLASH 忙标志位</b> (只读状态寄存器位, 写入无效) 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。

FOP[1:0]	<b>FLASH 操作模式选择</b>		
	<b>FOP1</b>	<b>FOP0</b>	<b>FLASH 操作模式</b>
	0	0	处于 Flash 只读模式
	0	1	对 FLASH 执行 Flash 写操作
	1	0	对 FLASH 执行 Flash 页擦除操作
	1	1	对 FLASH 执行 Flash 全擦除操作

## FLASHLOCK(Flash 锁定寄存器)

<b>FLASHLOCK</b> (Flash 锁定寄存器)			<b>基地址: 0x4000F000</b>					
			<b>偏移地址: 38H</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>	KEY[15:8]							
<b>Write:</b>	KEY[15:8]							
<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>	KEY[7:0]							
<b>Write:</b>	KEY[7:0]							
<b>Reset:</b>	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, 用户可擦写操作 FLASH。 写入非 0x7A68 数据后, FLASH 被锁定, 用户禁止擦写操作 FLASH。 默认值 0x0000 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>

## FS1LOCK(FlashSector1 锁定寄存器)

<b>FS1LOCK</b> (FlashSector1 锁定寄存器)			<b>基地址: 0x4000F000</b>					
			<b>偏移地址: 60H</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>	KEY[15:8]							
<b>Write:</b>	KEY[15:8]							
<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>	KEY[7:0]							
<b>Write:</b>	KEY[7:0]							

<b>Reset:</b>	0	0	0	0	0	0	0	0
---------------	---	---	---	---	---	---	---	---

位	功能描述
KEY[15:0]	<p><b>FlashSector1 锁定控制位</b></p> <p>对该寄存器写入 0x7161 后, 8K CodeFlash 0x00000 – 0x01FFF 被解锁, 用户可以对其进行页擦写操作。(注: 此时只能进行页擦除, 全擦无效)</p> <p>写入非 0x7161 数据后, CodeFlash 0x00000 – 0x01FFF 被锁定, 用户禁止擦写操作 FLASH。</p> <p>默认为锁定状态</p> <p>用户写入的是 0x7161, 读出值为 1; 写入的是非 0x7161, 读出值为 0</p>

### FS2LOCK(FlashSector2 锁定寄存器)

<b>FS2LOCK</b> (FlashSector2 锁定寄存器)			<b>基地址: 0x4000F000</b>					
			<b>偏移地址: 64H</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>	KEY[15:8]							
<b>Write:</b>	KEY[15:8]							
<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>	KEY[7:0]							
<b>Write:</b>	KEY[7:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
KEY[15:0]	<p><b>FlashSector2 锁定控制位</b></p> <p>对该寄存器写入 0x7262 后, 24K CodeFlash 0x02000 – 0x07FFF 被解锁, 用户可以对其进行页擦写操作。(注: 此时只能进行页擦除, 全擦无效)</p> <p>写入非 0x7262 数据后, CodeFlash 0x02000 – 0x07FFF 被锁定, 用户禁止写操作 FLASH。</p> <p>默认为锁定状态</p> <p>用户写入的是 0x7262, 读出值为 1; 写入的是非 0x7262, 读出值为 0</p>

### FS3LOCK(FlashSector3 锁定寄存器)

<b>FS3LOCK</b> (FlashSector3 锁定寄存器)			<b>基地址: 0x4000F000</b>					
			<b>偏移地址: 68H</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>	KEY[15:8]							

<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>	KEY[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
KEY[15:0]	<p><b>FlashSector3 锁定控制位</b></p> <p>对该寄存器写入 0x7363 后，32K CodeFlash 0x08000 – 0xFFFF 被解锁，用户可以对其进行页擦写操作。（注：此时只能进行页擦除，全擦无效）</p> <p>写入非 0x7363 数据后，CodeFlash 0x08000 – 0xFFFF 被锁定，用户禁止写操作 FLASH。</p> <p>默认为锁定状态</p> <p>用户写入的是 0x7363, 读出值为 1；写入的是非 0x7363, 读出值为 0</p>

### FS4LOCK(FlashSector4 锁定寄存器)

<b>FS4LOCK</b> (FlashSector4 锁定寄存器)		基地址: 0x4000F000 偏移地址: 6CH						
	<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>	KEY[15:8]							
<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>	KEY[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
KEY[15:0]	<p><b>FlashSector4 锁定控制位</b></p> <p>对该寄存器写入 0x7464 后，32K CodeFlash 0x10000 – 0x17FFF 被解锁，用户可以对其进行页擦写操作。（注：此时只能进行页擦除，全擦无效）</p> <p>写入非 0x7464 数据后，CodeFlash 0x10000 – 0x17FFF 被锁定，用户禁止写操作 FLASH。</p> <p>默认为锁定状态</p> <p>用户写入的是 0x7464, 读出值为 1；写入的是非 0x7464, 读出值为 0</p>

**FS5LOCK(FlashSector5 锁定寄存器)**

<b>FS5LOCK</b> (FlashSector5 锁定寄存器)			基地址: <b>0x4000F000</b> 偏移地址: <b>70H</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>	KEY[15:8]							
<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>	KEY[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
KEY[15:0]	<p><b>FlashSector5 锁定控制位</b></p> <p>对该寄存器写入 0x7565 后, 31K CodeFlash 0x18000–0x1FBFF 被解锁, 用户可以对其进行页擦写操作。(注: 此时只能进行页擦除, 全擦无效)</p> <p>写入非 0x7565 数据后, CodeFlash 0x18000 – 0x1FBFF 被锁定, 用户禁止写操作 FLASH。</p> <p>默认为锁定状态</p> <p>用户写入的是 0x7565, 读出值为 1; 写入的是非 0x7565, 读出值为 0</p>

**FS6LOCK(FlashSector6 锁定寄存器)**

<b>FS6LOCK</b> (FlashSector6 锁定寄存器)			基地址: <b>0x4000F000</b> 偏移地址: <b>74H</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>	KEY[15:8]							
<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>	KEY[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
KEY[15:0]	<p><b>FlashSector6 锁定控制位</b></p> <p>对该寄存器写入 0x7666 后, 32K CodeFlash 0x20000 – 0x27FFF 被解锁, 用户可以对其进行页擦写操作。(注: 此时只能进行页擦除, 全擦无效)</p> <p>写入非 0x7666 数据后, CodeFlash 0x20000 – 0x27FFF 被锁定, 用户禁止写操作</p>

	FLASH。 默认为锁定状态 用户写入的是 0x7666,读出值为 1；写入的是非 0x7666,读出值为 0
--	---

### FS7LOCK(FlashSector7 锁定寄存器)

<b>FS7LOCK</b> (FlashSector7 锁定寄存器)			基地址: 0x4000F000 偏移地址: 78H					
	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]	<b>FlashSector7 锁定控制位</b> 对该寄存器写入 0x7767 后, 32K CodeFlash 0x28000 – 0x2FFFF 被解锁, 用户可以进行页擦写操作。(注: 此时只能进行页擦除, 全擦无效) 写入非 0x7767 数据后, CodeFlash 0x28000 – 0x2FFFF 被锁定, 用户禁止写操作 FLASH。 默认为锁定状态 用户写入的是 0x7767,读出值为 1；写入的是非 0x7767,读出值为 0

### FS8LOCK(FlashSector8 锁定寄存器)

<b>FS8LOCK</b> (FlashSector8 锁定寄存器)			基地址: 0x4000F000 偏移地址: 7CH					
	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>FlashSector8 锁定控制位</b></p> <p>对该寄存器写入 0x7868 后, 32K CodeFlash 0x30000 – 0x37FFF 被解锁, 用户可以对其进行页擦写操作。(注: 此时只能进行页擦除, 全擦无效)</p> <p>写入非 0x7868 数据后, CodeFlash 0x30000 – 0x37FFF 被锁定, 用户禁止写操作 FLASH。</p> <p>默认为锁定状态</p> <p>用户写入的是 0x7868, 读出值为 1; 写入的是非 0x7868, 读出值为 0</p>

## FS9LOCK(FlashSector9 锁定寄存器)

FS9LOCK (FlashSector9 锁定寄存器)		基地址: 0x4000F000 偏移地址: 80H						
	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>FlashSector9 锁定控制位</b></p> <p>对该寄存器写入 0x7969 后, 31K CodeFlash 0x38000 – 0x3FBFF 被解锁, 用户可以对其进行页擦写操作。(注: 此时只能进行页擦除, 全擦无效)</p> <p>写入非 0x7969 数据后, CodeFlash 0x38000 – 0x3FBFF 被锁定, 用户禁止写操作 FLASH。</p> <p>默认为锁定状态</p> <p>用户写入的是 0x7969, 读出值为 1; 写入的是非 0x7969, 读出值为 0</p>

## FSALOCK(FlashSectorA 锁定寄存器)

FSALOCK (FlashSectorA 锁定寄存器)		基地址: 0x4000F000 偏移地址: 84H						
	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:	KEY[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
KEY[15:0]	<p><b>FlashSectorA 锁定控制位</b></p> <p>对该寄存器写入 0x7A6A 后，1K CodeFlash 0x1FC00 – 0x1FFFF 被解锁，用户可以对其进行页擦写操作。（注：此时只能进行页擦除，全擦无效）</p> <p>写入非 0x7A6A 数据后，CodeFlash 0x1FC00 – 0x1FFFF 被锁定，用户禁止写操作 FLASH。</p> <p>默认为锁定状态</p> <p>用户写入的是 0x7A6A, 读出值为 1；写入的是非 0x7A6A, 读出值为 0</p>

## FSBLOCK(FlashSectorB 锁定寄存器)

<b>FSBLOCK</b> (FlashSectorB 锁定寄存器)			<b>基地址: 0x4000F000</b> <b>偏移地址: 88H</b>					
	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>FlashSectorB 锁定控制位</b></p> <p>对该寄存器写入 0x7B6B 后，1K CodeFlash 0x3FC00 – 0x3FFFF 被解锁，用户可以对其进行页擦写操作。（注：此时只能进行页擦除，全擦无效）</p> <p>写入非 0x7B6B 数据后，CodeFlash 0x3FC00 – 0x3FFFF 被锁定，用户禁止写操作 FLASH。</p> <p>默认为锁定状态</p> <p>用户写入的是 0x7B6B, 读出值为 1；写入的是非 0x7B6B, 读出值为 0</p>

## FLASHCON2(Flash 控制寄存器 写保护)

<b>FLASHCON2 (写保护)</b>	<b>基地址: 0x4000F000</b>
------------------------	------------------------

(Flash 控制寄存器)			偏移地址: 90H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	CON[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CON[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CON[15:0]	<p><b>Flash 读写控制寄存器</b></p> <p>当 FLASHCON2= 0xCA53 时, Flash 处于写状态            当 FLASHCON2= 0xAC35 时, Flash 处于页擦状态            当 FLASHCON2 写其他任何值, Flash 处于读状态</p> <p>注: 写 FLASHCON2 没有全擦状态,此寄存器为数据传递功能, 最终作用于 FLASHCON 寄存器的相应功能, 读取值为 0, 如需判断 BUSY 状态, 请读取 FLASHCON 的 BUSY 标志位</p>

## 3 时钟单元

### 3.1 时钟分类

测试温度范围：-40°C~85°C

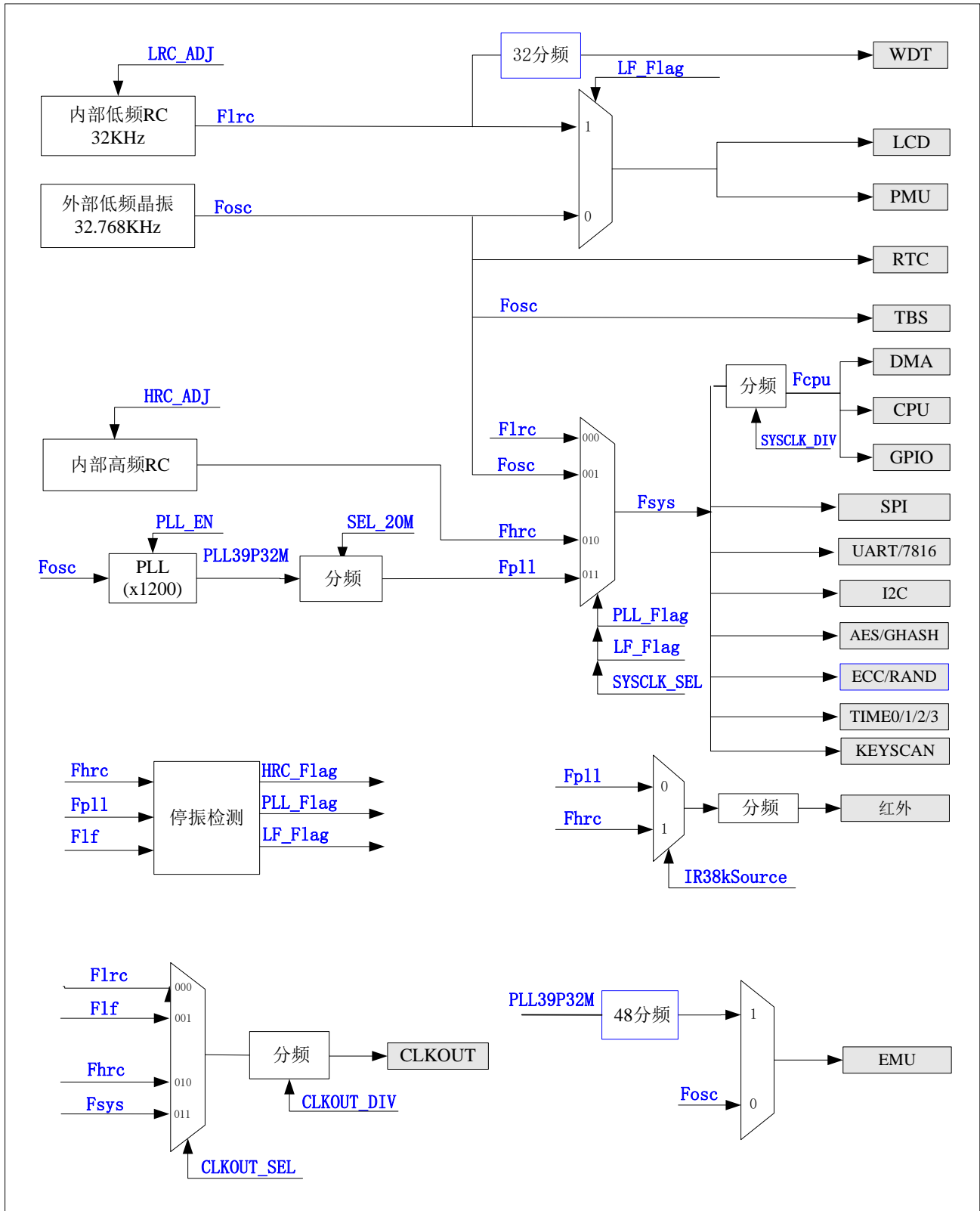
名称	频率	精度	功耗		
			MIN	TYP	MAX
内部低频 RC 时钟 (Flrc)	32KHz	13KHz~50KHz		1uA	
内部高频 RC 时钟 (Fhrc)	9.8MHz	±3%		130uA	
外部低频 OSC 晶振 (Fosc)	32.768KHz				1uA
内部 PLL (Fpll) 输出时钟	19.660800MHz/ 39.321600MHz			320uA	

注：内部高频 RC 时钟描述的频率是经过 HRCADJ 加载 Info Block 调整值之后的标称值，常温精度为±1%，全温度范围(-40°C~+85°C)为±3%

### 3.2 时钟框图

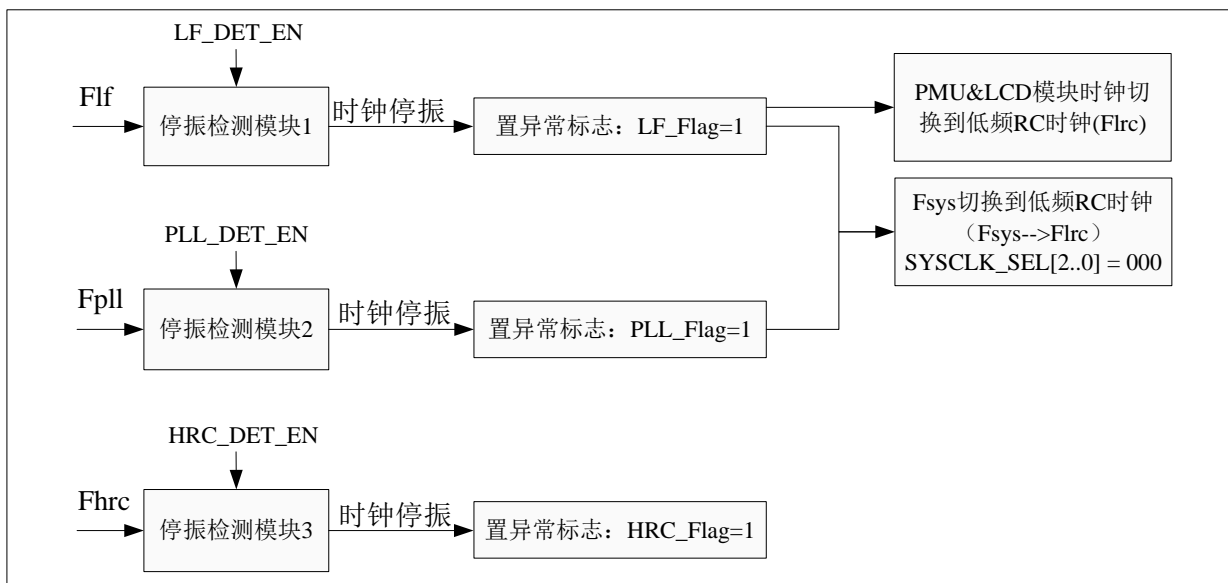
时钟符号说明：

- Flrc: 内部低频 RC 时钟(32KHz)，也作为看门狗时钟源。
- Fhrc: 内部高频RC时钟(9.8MHz)，系统复位后默认运行在Fhrc。
- Flf: 内部选择的低频时钟（32.768KHz），同Fosc时钟。
- Fosc: 外部低频OSC晶振时钟(32.768KHz)。
- Fpll: 内部PLL产生的高频时钟(39.321600MHz)或者PLL时钟2分频(19.660800MHz)，来源为Fosc。
- Fsys: 系统时钟，主要给各个外设提供时钟。
- Fcpu: 给CPU、GPIO、DMA、CRC提供时钟。



注：此图中，内部选择的低频时钟 Flf 为外部低频时钟 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 Flrc 振荡频率为 32KHz，提供给看门狗使用，可以选择该低频 RC 时钟 Flrc 作为系统时钟(SYSCLK\_SEL[2:0]=000)。

内部低频 RC 时钟 Flrc 可以通过寄存器 LRCADJ 校准，以提高时钟准确度。

#### 3.4.2 内部高频 RC 时钟 (Fhrc)

内部高频 RC 时钟 Fhrc 频率为 9.8MHz，可以选择该高频 RC 时钟 Fhrc 的分频输出作为系统时钟(SYSCLK\_SEL[2:0]=010)。

内部高频 RC 时钟 Fhrc 通过寄存器 HRCADJ 校准后，频率可达到 9.8MHz，常温精度可以达到 1%。

系统复位后，默认内部运行高频 RC 时钟 Fhrc (HRC\_EN = 1)，系统时钟选择内部高频 RC 时钟 Fhrc(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。

#### 3.4.3 外部低频晶振时钟 (Fosc)

芯片外接低功耗晶体振荡器，时钟频率 Fosc=32.768KHz，芯片内部集成了其震荡所需的电阻和电容，匹配电容约为 12.5pF。

### 3.4.4 内部 PLL 时钟 (Fpll)

内部 PLL 通过对外部低频晶振时钟 Fosc (32.768KHz) 倍频 (倍频值=1200), 可以对系统提供最高达 39.321600MHz 的高频时钟 Fpll, PLL 稳定输出后稳定标志位 PLL\_LOCK 置 1。在默认情况下 (SEL\_20M=0), PLL 输出 2 分频后的时钟, 即 Fpll 为 19.660800MHz。

### 3.4.5 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, 分别对内部低频时钟 Flf(来自 Fosc), PLL 输出时钟 Fpll, 和内部高频时钟 Fhrc 作检测。低频时钟 Flf 停振检测功能默认开启, 3 个停振检测模块均可以由用户软件关闭, 控制位分别为 LF\_DET\_EN, PLL\_DET\_EN, HRC\_DET\_EN。

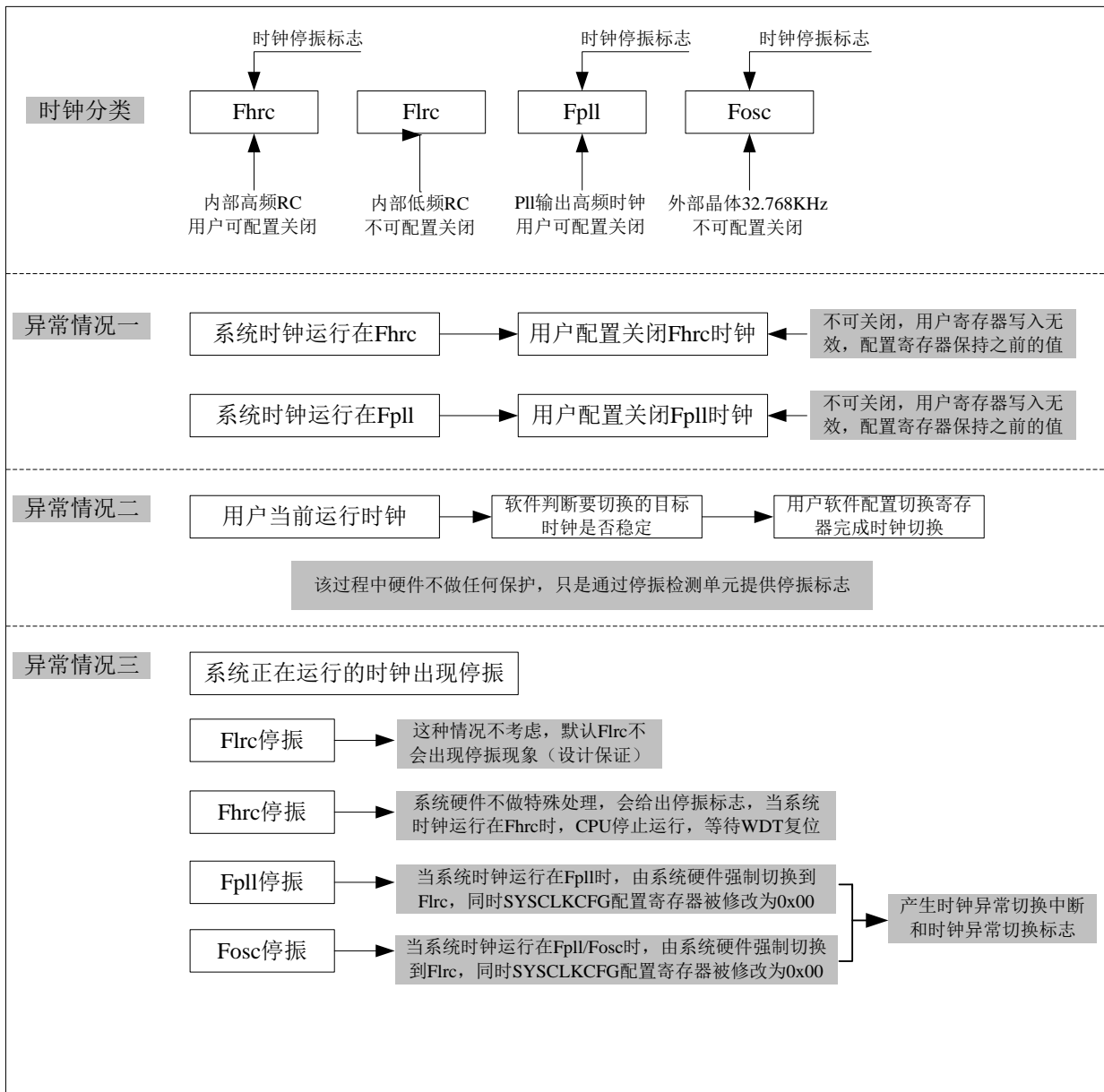
内部低频 RC 时钟 Flrc 默认不会停振 (设计保证), 作为时钟停振检测模块的时钟源。

当 Flf 停振检测开启检测到 Flf 停振, 系统给出时钟停振标志 LF\_FLAG; 如系统时钟 Fsys 选择 Flf 或 Fpll (Flf 为 Fpll 的时钟源) 时, 系统会由硬件强制将系统时钟 Fsys 切换到内部低频 RC 时钟 Flrc, 且产生中断 (NMI 中断), 同时将寄存器 SYSCLK\_SEL[2:0] 的值置为 000; 若系统时钟 Fsys 选择非 Flf 或非 Fpll 时, 系统时钟不会切换, 也不会发生 NMI 中断, 但停振标志会产生。

当 Fpll 停振检测开启检测到 Fpll 停振, 系统给出时钟停振标志 PLL\_FLAG; 如系统时钟 Fsys 选择 Fpll 时, 系统会由硬件强制将系统时钟 Fsys 切换到内部低频 RC 时钟 Flrc, 且产生中断 (NMI 中断), 同时将寄存器 SYSCLK\_SEL[2:0] 的值置为 000; 若系统时钟 Fsys 选择非 Fpll 时, 系统时钟不会切换, 也不会发生 NMI 中断, 但停振标志会产生。

当 Fhrc 停振检测开启检测到 Fhrc 停振, 系统给出标志位 HRC\_FLAG, 如系统时钟 Fsys 选择 Fhrc 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, 等待看门狗复位。

## 3.4.6 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	SYSCLKCFG	R/W	0x0002	系统时钟配置寄存器 (写保护)
0x08	JTAGSTA	R/W	0x0000	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	SYSCLKDIV	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	0x24A0	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/W	0x0000	Flash 控制寄存器 (写保护) (详情参见章节<2 存储器模块>)
0x38	FLASHLOCK	R/W	0x0000	Flash 锁定寄存器 (详情参见章节<2 存储器模块>)
0x3C	FLASHDLY	R/W	0x0000	<b>Flash 延迟取指寄存器</b>
0x60	FS1LOCK	R/W	0x0000	FlashSector1 锁定寄存器 (详情参见章节<2 存储器模块>)
0x64	FS2LOCK	R/W	0x0000	FlashSector2 锁定寄存器 (详情参见章节<2 存储器模块>)
0x68	FS3LOCK	R/W	0x0000	FlashSector3 锁定寄存器 (详情参见章节<2 存储器模块>)
0x6C	FS4LOCK	R/W	0x0000	FlashSector4 锁定寄存器 (详情参见章节<2 存储器模块>)
0x70	FS5LOCK	R/W	0x0000	FlashSector5 锁定寄存器 (详情参见章节<2 存储器模块>)
0x74	FS6LOCK	R/W	0x0000	FlashSector6 锁定寄存器 (详情参见章节<2 存储器模块>)
0x78	FS7LOCK	R/W	0x0000	FlashSector7 锁定寄存器 (详情参见章节<2 存储器模块>)
0x7C	FS8LOCK	R/W	0x0000	FlashSector8 锁定寄存器 (详情参见章节<2 存储器模块>)
0x80	FS9LOCK	R/W	0x0000	FlashSector9 锁定寄存器 (详情参见章节<2 存储器模块>)
0x84	FSALOCK	R/W	0x0000	FlashSectorA 锁定寄存器 (详情参见章节<2 存储器模块>)

0x88	FSBLOCK	R/W	0x0000	FlashSectorB 锁定寄存器 (详情参见章节<2 存储器模块>)
0x90	FLASHCON2	R/W	0x0000	Flash 控制寄存器 2 (写保护) (详情参见章节<2 存储器模块>)
0xF00	CHIPID	R	0x502x	芯片 ID 寄存器, 只读

## 3.6 特殊功能寄存器说明

### WPREG (写保护寄存器)

<b>WPREG</b> (写保护寄存器)			<b>基地址: 0x4000F000</b>					
			<b>偏移地址: 00H</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>	WPREG[15:8]							
<b>Write:</b>	WPREG[15:8]							
<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>	WPREG[7:0]							
<b>Write:</b>	WPREG[7:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
WPREG[15:0]	<p>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。 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。</p> <p>读该寄存器:</p> <p>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</p>

### SYSCLKCFG (系统时钟配置寄存器 写保护)

<b>SYSCLKCFG (写保护)</b> (系统时钟配置寄存器)			<b>基地址: 0x4000F000</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>	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>	WCLKE	X	X	X	X	SYSCLK_SEL[2:0]		
<b>Write:</b>	N	X	X	X	X	SYSCLK_SEL[2:0]		

Reset:	0	0	0	0	0	0	1	0
--------	---	---	---	---	---	---	---	---

位	功能描述																				
WCLKEN	<b>时钟配置寄存器写保护位</b> 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。																				
SYSCLK_SEL[2:0]	<b>系统时钟选择控制位:</b> <table border="1" style="margin: 10px auto;"> <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)

<b>JTAGSTA</b> (芯片状态指示寄存器)			<b>基地址: 0x4000F000</b>					
			<b>偏移地址: 08H</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>								
<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	X	FLASHS
<b>Write:</b>								TA
<b>Reset:</b>	0	0	0	0	0	0	0	X

位	功能描述
FLASHSTA	内部状态保留位 (只读)

**LRCADJ(LRC 时钟调整寄存器 写保护)**

<b>LRCADJ (写保护)</b> (LRC 时钟调整寄存器)			基地址: <b>0x4000F000</b> 偏移地址: <b>0CH</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>								
<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	LRC_ADJ[3:0]			
<b>Write:</b>								
<b>Reset:</b>	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

注: 为保证 LRC 输出频率的准确性, 建议客户只能加载 Info Block 存储的出厂测试值, 不要自行修改为其它值。

**HRCADJ(HRC 时钟调整寄存器 写保护)**

<b>HRCADJ (写保护)</b> (HRC 时钟调整寄存器)			基地址: <b>0x4000F000</b> 偏移地址: <b>10H</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>								
<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	HRC_ADJ[6:0]						
<b>Write:</b>								
<b>Reset:</b>	0	1	0	0	0	0	0	0

位	功能描述
HRC_ADJ[6:0]	HRC 输出频率调节控制位

注: 为保证 HRC 输出频率的准确性, 建议客户只能加载 Info Block 存储的出厂测试值, 不要自行修改为其它值。

**CLKSTA(时钟状态寄存器)**

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:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	PLL_LOCK	PLL_FLAG	HRC_FLAG	X	X	LF_FLAG
Write:			X	X	X			X
Reset:	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	PLL 时钟锁定状态标志 0: PLL 时钟未锁定。 1: PLL 时钟锁定正常。 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
PLL_FLAG	PLL 时钟 Fpll 停振标志 0: 正常。 1: 停振。
HRC_FLAG	内部高频 RC 时钟 Fhrc 停振标志 0: 正常。 1: 停振。
LF_FLAG	外部低频晶振时钟 Fosc 停振标志 0: 正常 1: 停振

**SYSCLKDIV(系统时钟分频寄存器 写保护)**

SYSCLKDIV (写保护) (系统时钟分频寄存器)			基地址: 0x4000F000 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	SEL_20	SYSCLK_DIV[2:0]		
Write:					M			
Reset:	0	0	0	0	0	0	0	1

	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 <b>注: 如果选择 CPU 时钟为 39.32M, 且系统时钟选择不分频(SYSCLK_DIV[2:0] = 000B), 则必须提前使能指令延迟取指功能 (FLASHDLY.FLASH_DLY=1)。</b>																																				
SYSCLK_DIV[2:0]	系统时钟分频得到 Fcpu, 设置如下: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="3">SYSCLK_DIV [2:0]</th> <th>内核时钟 (Fcpu)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Fsys</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Fsys/2(Default)</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Fsys/4</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Fsys/8</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Fsys/16</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Fsys/32</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Fsys/64</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Fsys/128</td> </tr> </tbody> </table>	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
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(CLKOUT 时钟选择寄存器 写保护)

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]	<b>CLKOUT 引脚时钟输出配置</b>						
	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			
<p>1, 用户可将芯片内部时钟源从 CLKOUT 引脚输出, 以观测内部时钟。</p> <p>2, 用户可用 CLKOUTDIV 寄存器将内部时钟分频后输出, 可作为外部设备的时钟源。</p>							

### CLKOUTDIV(CLKOUT 时钟分频寄存器 写保护)

<b>CLKOUTDIV (写保护)</b> (CLKOUT 时钟分频寄存器)			<b>基地址: 0x4000F000</b>					
			<b>偏移地址: 28H</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>								
<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	CLKOUT_DIV[3:0]			
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>位</b>	<b>功能描述</b>
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(内部模块使能控制寄存器 0 写保护)

<b>CLKCTRL0 (写保护)</b> (内部模块使能控制寄存器 0)			<b>基地址: 0x4000F000</b>					
			<b>偏移地址: 2CH</b>					
	<b>Bit23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>Bit16</b>
<b>Read:</b>	X	X	X	X	X	X	X	<b>ECC_EN</b>
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

	<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>	EMU_EN	AES_EN	1P5LBO	CLKOU	KEY_EN	OSC_SL	HRC_DE	PLL_DE
<b>Write:</b>	N		R_EN	T_EN		P	T_EN	T_EN
<b>Reset:</b>	0	0	1	0	0	1	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>	LF_DET	PLL_LO	HRC_EN	PLL_EN	I2C_EN	SPI_EN	LCD_EN	X
<b>Write:</b>	_EN	CK_EN						
<b>Reset:</b>	1	0	1	0	0	0	0	0

位	功能描述
ECC_EN	<b>ECC 模块使能信号</b> 0: 关闭 (default) 1: 打开
EMU_EN	<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 (用户不要修改该寄存器的默认值)</b> 0: 关闭 1: 打开 (default)
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 (default) 1: 使能
OSC_SLP	<b>OSC 低功耗使能控制信号 (用户不要修改该寄存器的默认值)</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 (default) 1: PLL 强制锁定

	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>HRC 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: 1.当用户切换至 HRC 时钟作为系统时钟时，必须先保证 HRC_EN 开启。 2.当用户选择 Fsys 为 Fpll 时，此时不能关闭 PLL_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: 1.当用户切换至 PLL 时钟作为系统时钟时，必须先保证 PLL_EN 开启。 2.当用户选择 Fsys 为 Fpll 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
SPI_EN	<b>SPI 模块时钟使能位</b> 0: 关闭 SPI 模块 (default) 1: 使能 SPI 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

## CLKCTRL1(内部模块使能控制寄存器 1 写保护)

<b>CLKCTRL1 (写保护)</b> (内部模块使能控制寄存器 1)		<b>基地址: 0x4000F000</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_RT	LRC_TF					UART5_	UART4_
<b>Write:</b>	C2_EN	REQ_RT	X	X	X	X	EN	7816_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_	UART2_	UART1_	UART0_	TMR3_E	TMR2_E	TMR1_E	TMR0_E
<b>Write:</b>	7816_EN	EN	EN	EN	N	N	N	N
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
LRC_RTC2_EN	<b>辅助 RTC 的工作使能位</b> 0: 关闭

	1: 开启
LRC_TFREQ_RTC2	辅助 RTC 的测频使能位 0: 关闭 1: 开启
UART5_EN	UART5 时钟使能位 0: 关闭 1: 使能
UART4_7816 _EN	UART4 时钟使能位 0: 关闭 1: 使能
UART3_7816 _EN	UART3 时钟使能位 0: 关闭; 1: 使能
UART2_EN	UART2 时钟使能位 0: 关闭 1: 使能
UART1_EN	UART1 时钟使能位 0: 关闭 1: 使能
UART0_EN	UART0 时钟使能位 0: 关闭 1: 使能
TMR3_EN	Timer3 时钟使能位 0: 关闭 1: 使能
TMR2_EN	Timer2 时钟使能位 0: 关闭 1: 使能
TMR1_EN	Timer1 时钟使能位 0: 关闭 1: 使能
TMR0_EN	Timer0 时钟使能位 0: 关闭 1: 使能

### FLASHDLY(Flash 控制寄存器 写保护)

<b>FLASHDLY (写保护)</b>		<b>基地址: 0x4000F000</b>						
<b>(Flash 控制寄存器)</b>		<b>偏移地址: 3CH</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	X	X	X	X	X	FLASH_

<b>Write:</b>								DLY
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
FLASH_DLY	<b>Flash 延迟取指控制</b> 0: 在 Flash 中取指令时不等待 1: 在 Flash 中取指令时等待 系统时钟 Fsys 选择 39.32M 时, 且系统时钟分频选择不分频 (SYSCLKDIV. SYSCLK_DIV[2:0]= 000B), 必须配置此控制位为 1。

## CHIPID(芯片版本寄存器)

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

位	功能描述
CHIPID[15: 0]	芯片 ID: 0x502x, x 为内部 ID 识别码为 0~F 中任意值 (只读)

## 4 电源单元

### 4.1 概述

PMU 为芯片的电源管理模块，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 LVDIN0 和 LVDIN1，当供电电压低于或高于设定阈值时可产生中断信号。
- 自动完成 VSYS 电源和 VBAT 电源的切换，电源状态在 PMUSTA 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 BOR 或 VSYS 检测中断信号
- 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
- 为芯片内部数字模块提供 1.5V 电源

4.2 框图

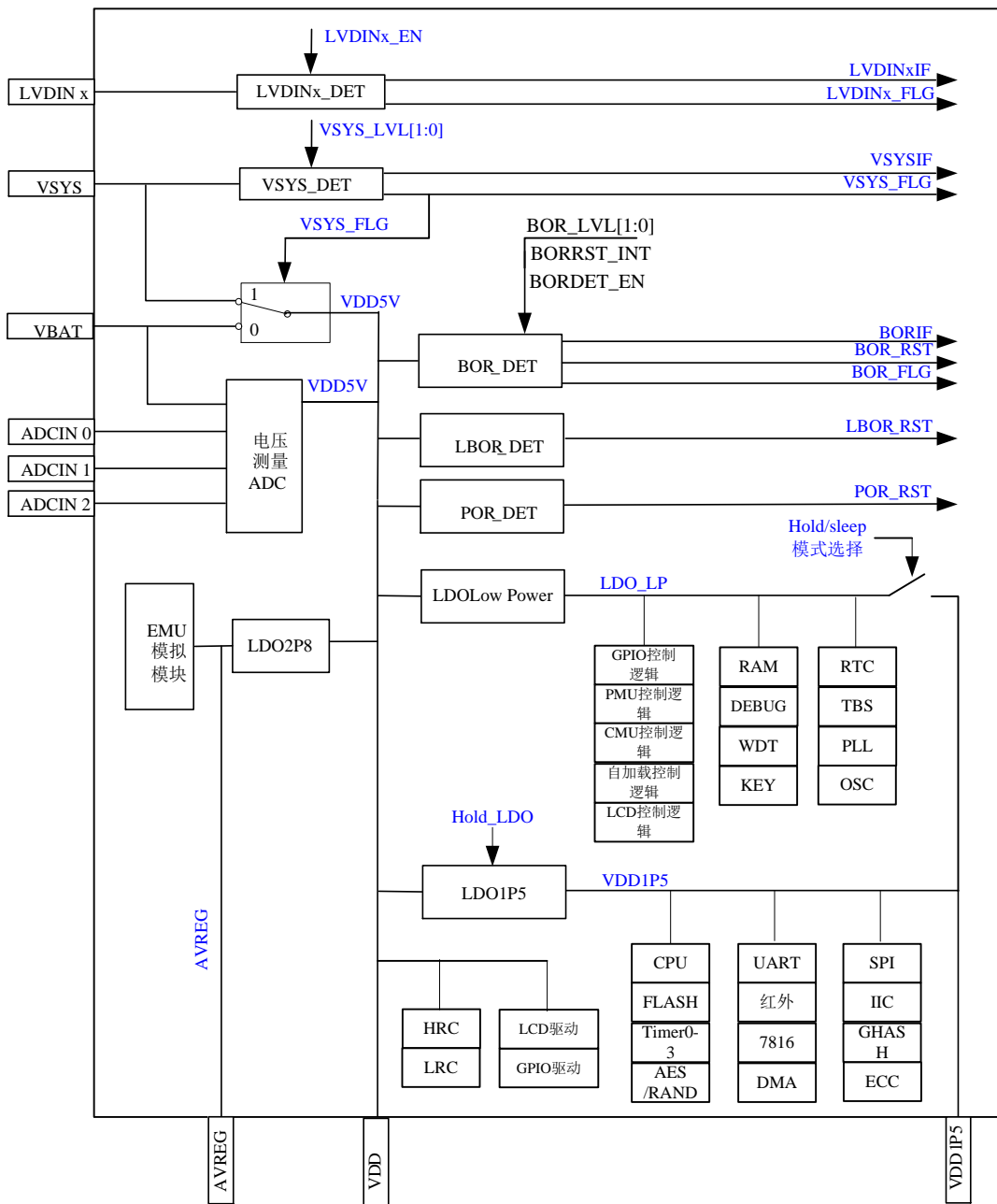


图 4-1 芯片内部供电框图

## 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，并将监测结果以中断信号和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 regulator 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)

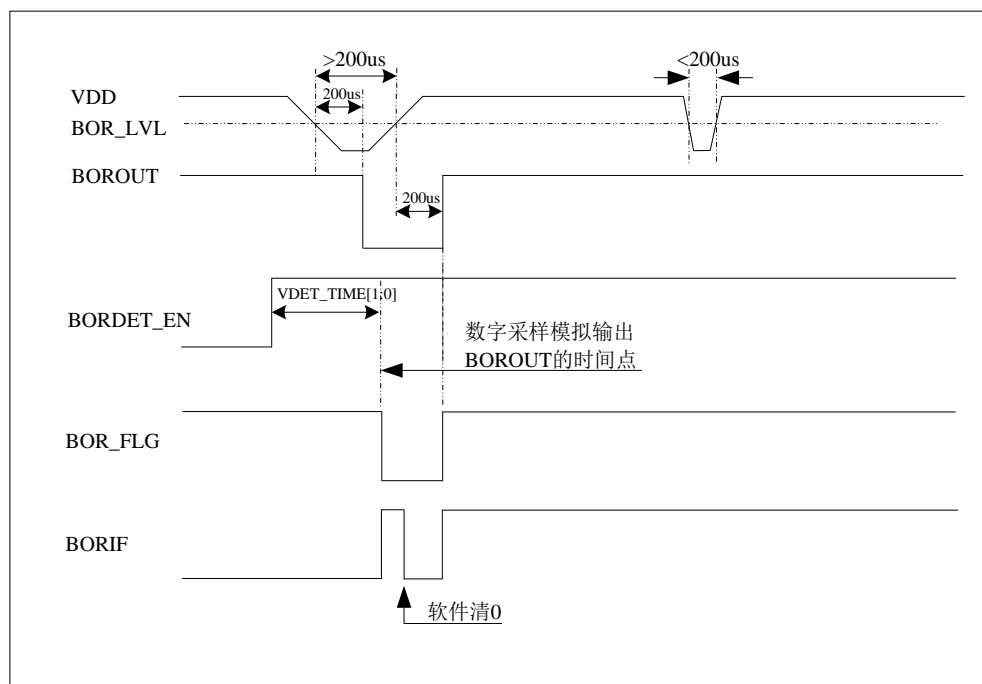
当 BOR 检测模块检测到系统电源 VDD 低于设定电压 Vborl 时, BOR 检测模块内部信号 BOROUT 输出低电平, 内部复位信号 IRST 也将变为低电平, 复位状态寄存器 RSTSR 的 BOR 标志位被置为 1。当掉电检测电路检测到系统电源 VDD 电压高于设定电压 Vborh 时, BOR 检测模块内部信号 BOROUT 输出高电平, 在该高电平持续的 1088 个 Flrc/Fosc 周期后, 内部复位信号 IRST 也变为高电平。

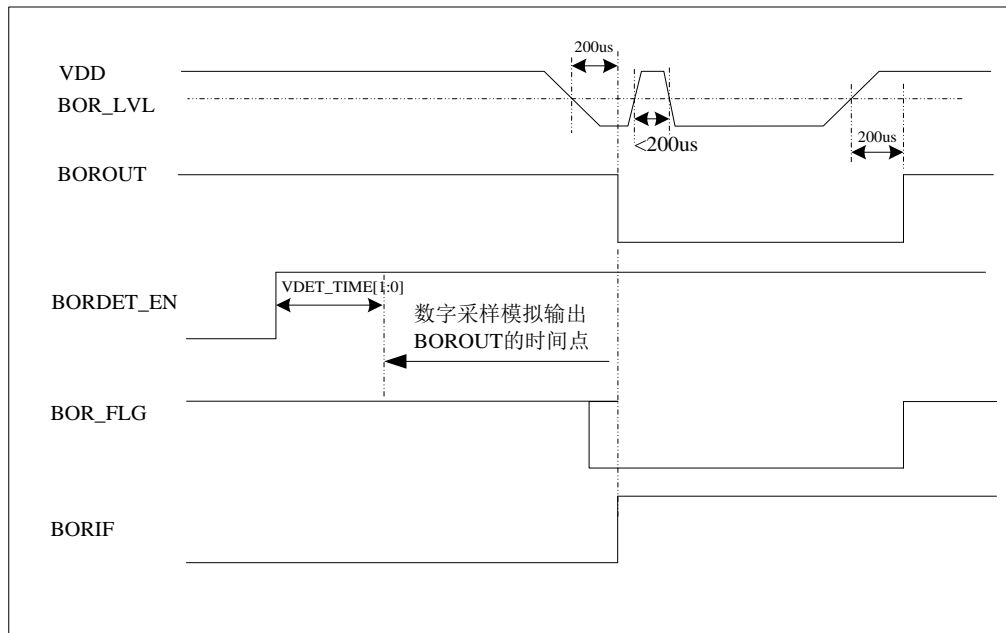
Vborh 和 Vborl 之间的迟滞为 200mV, BOR 模块的检测阈值可通过 VDETCFG 中的 BOR\_LVL[1:0] 设置。

掉电复位 BOR 产生时, 下面的事件将会发生:

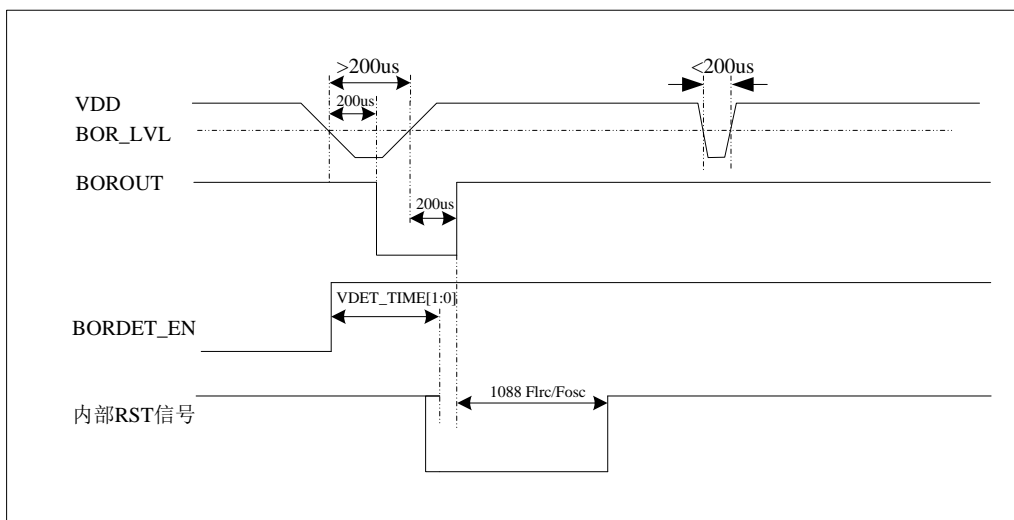
- 产生一个 BOR 脉冲
- 内部复位信号 IRST 有效
- 计数 1088 个 Flrc/Fosc 周期
- 复位状态寄存器 RSTSR 的掉电复位标志位 BOR 被设置为 1。
- CPU 从 0000H 开始执行程序

LBOR\_DET 与 BOR\_DET 的检测过程基本相同。



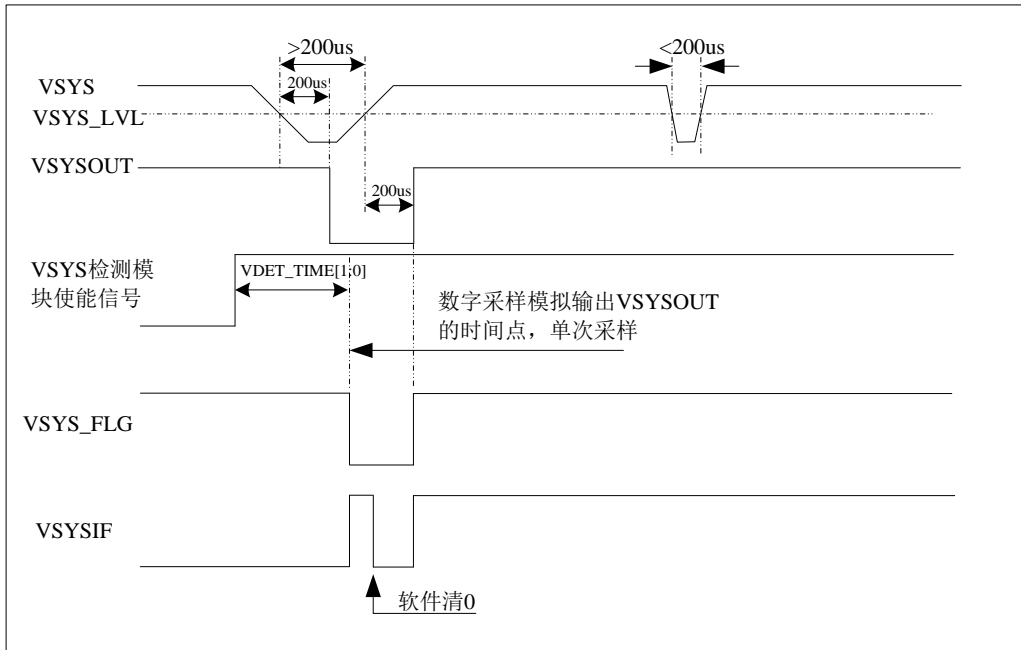


BOR 中断信号产生示意图

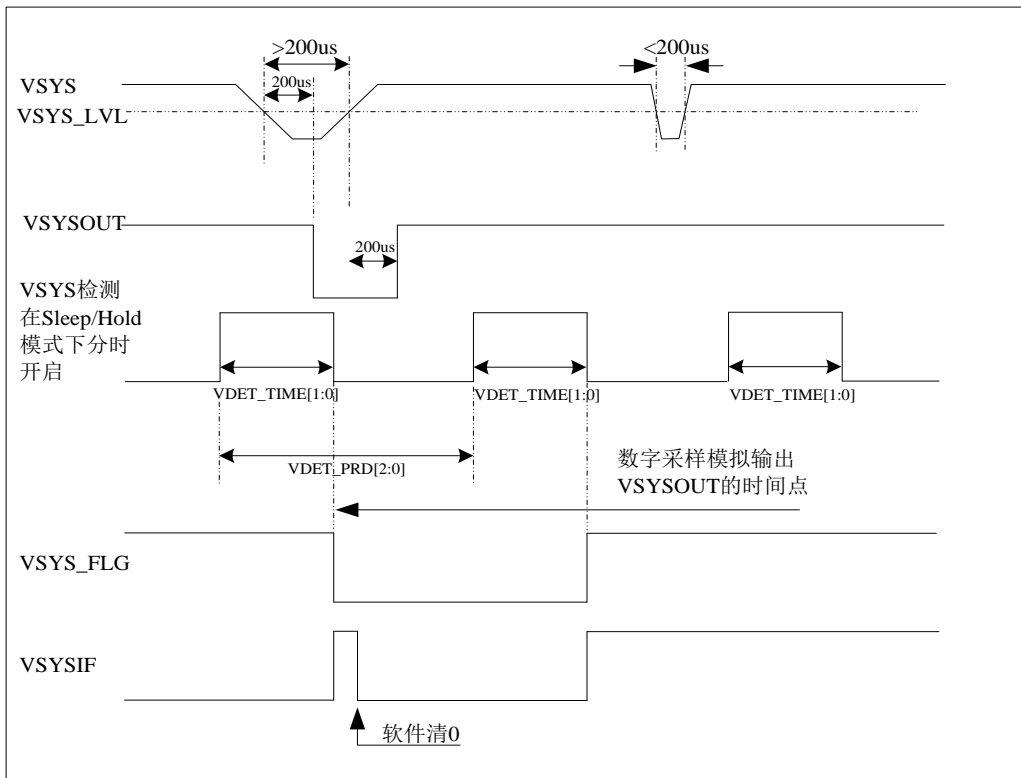


BOR 复位示意图

### 4.3.5 系统电源检测功能(VSYS\_DET)

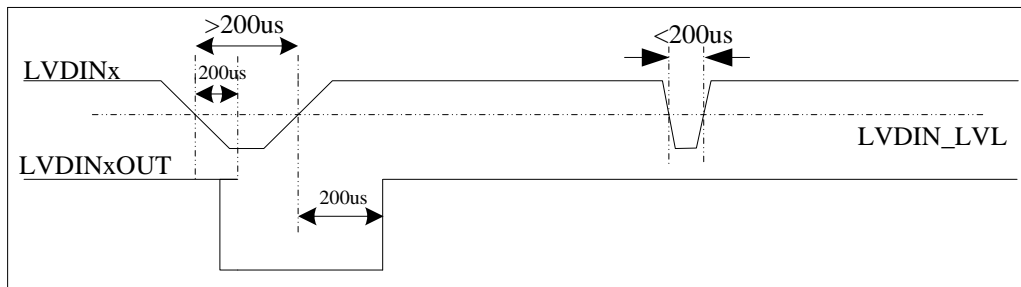


Normal 模式 Vsys 检测



Sleep/Hold 模式下 Vsys 分时检测

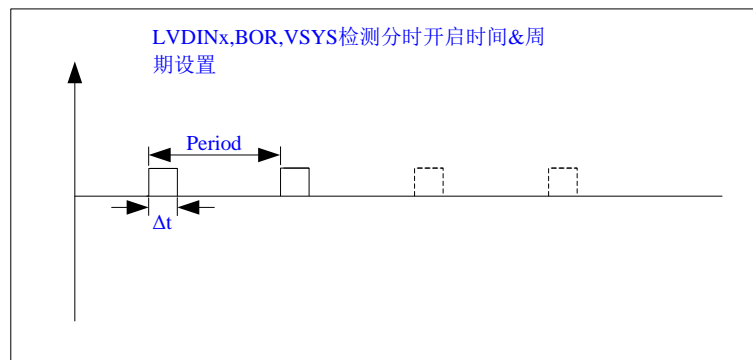
### 4.3.6 低电压检测功能(LVD\_DET)



LVDIN 检测信号示意图

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

系统在进入 Hold 或 Sleep 低功耗模式时，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：



VSYS\_DET、LVD\_DET、BOR\_DET 在 Sleep/Hold 下的分时检测时序图请参考<4.3.5 系统电源检测功能>章节，Sleep/Hold 模式下 Vsys 分时检测框图。

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。Δt 是分时检测开启 LVD\_DET、VSYS\_DET 和 BOR\_DET 采样时间，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 上会流过一定量的电流，用户可以配置 DISCHARGE\_CTRL 来控制这个防钝化电流的大小。

## 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	0x0089	电源检测阈值配置寄存器
0x08	VDETPCFG	R/W	0x0022	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	0x00000000	唤醒源标志寄存器

## 4.5 特殊功能寄存器说明

### PMUCON(PMU 配置寄存器 写保护)

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
<b>Read:</b>	DISCHARGE_EN	X	X	X	X	X	X	X
<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>	X	X	X	Hold_LD	LVDIN1	LVDIN0	BORRST	BORDET_EN
<b>Write:</b>				O	_EN	_EN		
<b>Reset:</b>	0	0	0	1	1	1	1	1

位	功能描述
DISCHARGE_EN	电池防钝化功能使能控制 0: 电池防钝化功能关闭 1: 电池防钝化功能开启
Hold_LDO	在 Hold 模式下选择打开/关闭 大功耗 LDO (默认打开) 0: 关闭大功耗 LDO 1: 打开大功耗 LDO (default)
LVDIN1_EN	LVD_DET 模块使能信号, 监测 LVDIN1 输入引脚 0: 关闭 LVDIN1_DET 模块 1: 开启 LVDIN1_DET 模块 (default)
LVDIN0_EN	LVD_DET 模块使能信号, 监测 LVDIN0 输入引脚

	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: VDD电压低/高于设定阈值时产生BOR中断 1: VDD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)

## VDETCFG(电源检测阈值配置寄存器)

<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>	DISCHA							
<b>Write:</b>	RGE_CT RL	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>	RESERV	X	VSYS_LVL[3:0]				BOR_LVL[1:0]	
<b>Write:</b>	ED							
<b>Reset:</b>	1	0	0	0	1	0	0	1

位	功能描述																																																							
DISCHARGE_CTRL	电池防钝化功能，电流大小选择： 0: 1mA 1: 2mA																																																							
Bit7	内部保留位，应用时保持为默认值 1																																																							
VSYS_LVL [3:0]	<b>VSYS_DET检测阈值控制位</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4">VSYS_LVL[3:0]</th> <th>检测电压</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>2.4V</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>2.6V</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2.8V (default)</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3.0V</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>3.2V</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>3.4V</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>3.6V</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>3.8V</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>4.0V</td></tr> <tr><td>1</td><td></td><td>0</td><td>1</td><td>4.2V</td></tr> </tbody> </table>	VSYS_LVL[3:0]				检测电压	0	0	0	0	2.4V	0	0	0	1	2.6V	0	0	1	0	2.8V (default)	0	0	1	1	3.0V	0	1	0	0	3.2V	0	1	0	1	3.4V	0	1	1	0	3.6V	0	1	1	1	3.8V	1	0	0	0	4.0V	1		0	1	4.2V
VSYS_LVL[3:0]				检测电压																																																				
0	0	0	0	2.4V																																																				
0	0	0	1	2.6V																																																				
0	0	1	0	2.8V (default)																																																				
0	0	1	1	3.0V																																																				
0	1	0	0	3.2V																																																				
0	1	0	1	3.4V																																																				
0	1	1	0	3.6V																																																				
0	1	1	1	3.8V																																																				
1	0	0	0	4.0V																																																				
1		0	1	4.2V																																																				

	1	0	1	0	4.4V
	1	0	1	1	4.6V
	1	1	0	0	4.8V
	1		0	1	5V
	1	1	1	X	5V
BOR_LVL[1:0]	<b>BOR_DET检测阈值控制位</b>				
	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)

<b>VDETPCFG</b> (电源检测周期配置寄存器)			基地址: <b>0x4000F400</b> 偏移地址: <b>08H</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>								
<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	RSRV	VDET_TIME[1:0]		VDET_PRD[2:0]		
<b>Write</b>								
<b>Reset:</b>	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]	<b>Hold&amp;Sleep 模式下 VSYS_DET, BOR_DET, LVDIN_DET 分时检测的时间设定</b>	
	VDET Time[1:0]	检测时间
	0 0	1068us(default)
	0 1	1068us
	1 0	1068us
	1 1	1068us

VDET_PRD[2:0]	<b>Hold&amp;Sleep 模式下 VSYS_DET,BOR_DET, LVDIN_DET 分时检测的周期设定</b>			
	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)

<b>PMUIE</b> (PMU 中断使能寄存器)			<b>基地址: 0x4000F400</b>					
			<b>偏移地址: 0CH</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>								
<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	LVDIN1I E	LVDIN0I E	BORIE	VSYSIE
<b>Write</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
LVDIN1IE	<b>LVDIN1检测中断使能位</b> 0: 关闭 1: 允许
LVDIN0IE	<b>LVDIN0检测中断使能位</b> 0: 关闭 1: 允许
BORIE	<b>BOR检测中断使能位</b> 0: 关闭 1: 允许
VSYSIE	<b>VSYS检测中断使能位</b> 0: 关闭 1: 允许

注：设置 PMU 的中断，需要同时使能 PMU 中断的内核函数才有效。

NVIC\_EnableIRQ(PMU\_IRQn);

### PMUIF(PMU 中断标志寄存器)

<b>PMUIF</b> (PMU 中断标志寄存器)			<b>基地址: 0x4000F400</b> <b>偏移地址: 10H</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>								
<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	LVDIN1I	LVDIN0I	BORIF	VSYSIF
<b>Write</b>					F	F		
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
LVDIN1IF	<b>LVDIN1检测中断标志位</b> 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，软件写0清0。
LVDIN0IF	<b>LVDIN0检测中断标志位</b> 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，软件写0清0。
BORIF	<b>BOR检测中断标志位</b> 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，软件写0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，软件写0清0。

注：该寄存器不能被 Wake\_UP 唤醒复位。

### PMUSTA(PMU 状态寄存器)

<b>PMUSTA</b> (PMU 状态寄存器)			<b>基地址: 0x4000F400</b> <b>偏移地址: 14H</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

	Bit7	6	5	4	3	2	1	Bit0
<b>Read</b>	X	X	X	X	LVDIN1_FLG	LVDIN0_FLG	BOR_FLG	VSYS_FLG
<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
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
LVDIN0_FLG	<b>LVDIN0 引脚电压状态</b> 0: 表示 LVDIN0 引脚电压小于 1.21V 阈值 1: 表示 LVDIN0 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])

注：该寄存器不能被 Wake\_UP 唤醒 RESET。

## WAKEIF(唤醒标志寄存器)

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
<b>Read:</b>	NMIWK IF	INT9WK IF	INT8WK IF	INT7WK IF	KEYWK IF	DMAW KIF	EMUWK IF	RSRV
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
<b>Read:</b>	X	SPIWKI F	I2CWKI F	RTCWKI F	TBSWKI F	TMR3W KIF	TMR2W KIF	TMR1W KIF
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	TMR0W KI	RX5WKI F	RX4WKI F	RX3WKI F	RX2WKI F	RX1WKI F	RX0WKI F	INT6WK IF
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	INT5WK	INT4WK	INT3WK	INT2WK	INT1WK	INT0WK	ARGEW	PMUWK

	IF	IF	IF	IF	IF	IF	KIF	IF
<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
NMIWKIF	<b>NMI唤醒标志</b> 当系统唤醒时发生了不可屏蔽中断 NMI，设置标志位为 1
INT9WKIF	<b>INT9唤醒标志</b> SLEEP/HOLD 模式下,INT9 唤醒发生时，设置标志位为 1
INT8WKIF	<b>INT8唤醒标志</b> SLEEP/HOLD 模式下,INT8 唤醒发生时，设置标志位为 1
INT7WKIF	<b>INT7唤醒标志</b> SLEEP/HOLD 模式下,INT7 唤醒发生时，设置标志位为 1
KEYWKIF	<b>按键扫描唤醒标志</b> SLEEP/HOLD 模式下,KEY 唤醒发生时，此位置为 1
DMAWKIF	<b>DMA唤醒标志</b> HOLD 模式下,DMA 唤醒发生时，此位置为 1
EMUWKIF	<b>EMU唤醒标志</b> HOLD 模式下,EMU 唤醒发生时，此位置为 1
RSRV	该位恒为 0
SPIWKIF	<b>SPI唤醒标志</b> HOLD 模式下,SPI 唤醒发生时，此位置为 1
I2CWKIF	<b>I2C唤醒标志</b> HOLD 模式下,I2C 唤醒发生时，此位置为 1
RTCWKIF	<b>RTC唤醒标志</b> SLEEP/HOLD 模式下 RTC 唤醒发生时，此位置为 1。 (具体是 RTC 哪个唤醒源，需要查看 RTCIF 寄存器)
TBSWKIF	<b>TBS唤醒标志</b> SLEEP/HOLD 模式下 TBS 唤醒发生时，此位置为 1。 (具体是 TBS 哪个唤醒源，需要查看 TBSIF 寄存器)
TMR3WKIF	<b>TMR3 唤醒标志</b> HOLD 模式下,TMR3 唤醒发生时，此位置为 1
TMR2WKIF	<b>TMR3 唤醒标志</b> HOLD 模式下,TMR2 唤醒发生时，此位置为 1
TMR1WKIF	<b>TMR3 唤醒标志</b> HOLD 模式下,TMR1 唤醒发生时，此位置为 1
TMR0WKIF	<b>TMR3 唤醒标志</b> HOLD 模式下,TMR0 唤醒发生时，此位置为 1
RX5WKIF	<b>RX5唤醒标志</b> SLEEP/HOLD 模式下,RX5 唤醒发生时，设置标志位为 1
RX4WKIF	<b>RX4唤醒标志</b> SLEEP/HOLD 模式下,RX4 唤醒发生时，设置标志位为 1
RX3WKIF	<b>RX3唤醒标志</b>

	SLEEP/HOLD模式下,RX3唤醒发生时, 设置标志位为1
RX2WKIF	<b>RX2唤醒标志</b> SLEEP/HOLD模式下,RX2唤醒发生时, 设置标志位为1
RX1WKIF	<b>RX1唤醒标志</b> SLEEP/HOLD模式下,RX1唤醒发生时, 设置标志位为1
RX0WKIF	<b>RX0唤醒标志</b> SLEEP/HOLD模式下,RX0唤醒发生时, 设置标志位为1
EXTI6WKIF	<b>INT6唤醒标志</b> SLEEP/HOLD模式下,INT6唤醒发生时, 设置标志位为1
EXTI5WKIF	<b>INT5唤醒标志</b> SLEEP/HOLD模式下,INT5唤醒发生时, 设置标志位为1
EXTI4WKIF	<b>INT4唤醒标志</b> SLEEP/HOLD模式下,INT4唤醒发生时, 设置标志位为1
EXTI3WKIF	<b>INT3唤醒标志</b> SLEEP/HOLD模式下,INT3唤醒发生时, 设置标志位为1
EXTI2WKIF	<b>INT2唤醒标志</b> SLEEP/HOLD模式下,INT2唤醒发生时, 设置标志位为1
EXTI1WKIF	<b>INT1唤醒标志</b> SLEEP/HOLD模式下,INT1唤醒发生时, 设置标志位为1
EXTI0WKIF	<b>INT0唤醒标志</b> SLEEP/HOLD模式下,INT0唤醒发生时, 设置标志位为1
<b>ARGEWKIF</b>	<b>AES/RAND/GHASH/ECC 唤醒标志</b> HOLD模式下, <b>AES/RAND/GHASH/ECC</b> 唤醒发生时, 此位置为1
PMUWKIF	<b>PMU唤醒标志</b> SLEEP/HOLD模式下,PMU唤醒发生时, 此位置为1 (具体那个PMU唤醒源需要查看PMUIF寄存器)

注: 1.Sleep 唤醒和 Hold 唤醒共用此标志位

2.**DMAWKIF /EMUWKIF/ SPIWKIF/I2CWKIF/TMR3WKIF/ TMR2WKIF/ TMR1WKIF/ TMR0WKIF/ARGEWKIF**, 只能在 Hold 下才能发生唤醒生效, sleep 模式不能唤醒。

3.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。该寄存器只能被 POR 复位。

## 5 调试支持

### 5.1 概况

HT502X 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

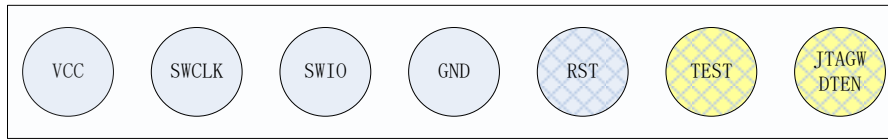
### 5.3 SW 引脚使用说明

管脚	软件配置为复用功能 2 TEST 为高电平	TEST=0, JTAGWDTEN=0
PB.13/ SEG13/ SWIO	SWIO	固定为 SWIO (软件配置不起作用, OD 关闭)
PB.15/ SEG15/ SWCLK	SWCLK	固定为 SWCLK (软件配置不起作用, OD 关闭)

注: TEST=0, JTAGWDTEN=1 情况下芯片进入测试模式, 用户应避免进入这种模式

## 5.4 推荐调试端口

推荐调试口配置如下：

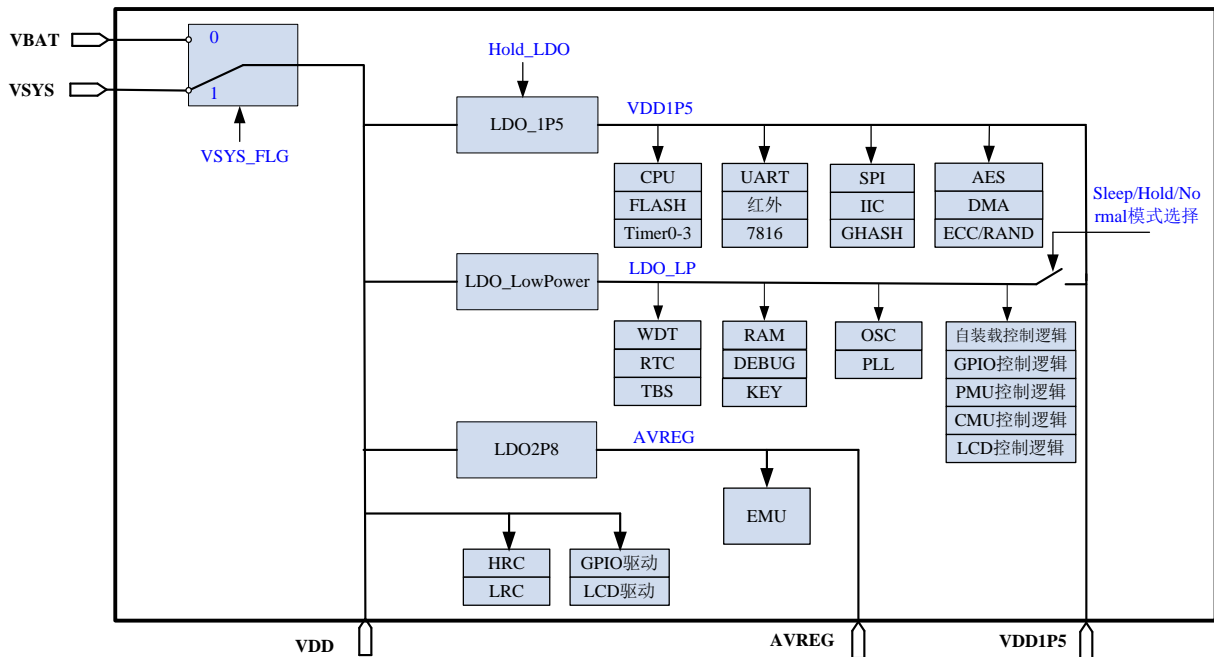


说明：

- 1, 前 4 个端口是程序下载必须需要的（如果目标板有供电，VCC 可以省去）；
- 2, RST 的作用是当 SW 状态机发生紊乱时，提供一种可靠的复位 SW 状态机的方式；
- 3, TEST 和 JTAGWDTEN 作用是当用户不小心将 SWCLK/SWIO 配置为其他功能（比如 GPIO 或者 SEG 功能），强制将 PB13/PB15 配置为对应的 SW 口。
- 4: 当 TEST 引脚从拉低变成拉高状态时，应让芯片发生一次上电复位或外部引脚复位。

## 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> 这种模式下，除了 PB13，PB15 被固定为 SW 接口外，其余与正常模式一致
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。 在正常 normal 模式运行，看门狗不能被禁止 <b>这种模式下，用户可以通过将 PB13/15 配置为复用功能 2（即 SW 口功能），进行调试模式操作。</b>
		<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：

	<pre>SCB-&gt;SCR = 0x0004; __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; __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

- 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LVD、BOR和VSYSD检测功能，则LVD\_DET、BOR\_DET和VSYSD\_DET模块会由硬件控制自动进入分时开启模式以降低功耗；
- 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寄存器配置为0x2400，CLKCTRL1寄存器全部清0
- 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，LDO\_LowPower与外部引脚VDD1P5连接的开关断开，VDD1P5无输出。

### 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（RSTSTA寄存器的WakeupRST和Sleep\_Flag位），如果这两个标志位为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3种中断源确定，可以查询PMUIF对应的位来确认是BORIF、VSYISIF、LVDINxIF，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、VDDIF、ADC0IF、ADC1IF、ADC2IF、VbatCMPIF、ADC0CMPIF、ADC1CMPIF，详细见TBS单元。

(4) WAKEIF的INTxWKIF（x=0~6）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=0~5）位为1，表示CPU是由相应的RX引脚信号引起了唤醒。

(6) WAKEIF的KEYWKIF位为1，表示CPU是由按键信号引起了唤醒。

### 6.3.4 进入 SLEEP 模式

SLEEP模式通过CortexM0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，LDO\_1P5 由用户控制开关的，数字的 LDO\_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工作正常，用户可配置开启大功耗大输出驱动能力的 LDO\_1P5 来适应其应用的需求，即使能 [PMUCON.Hold\\_LDO \(bit4\)](#)。

为了降低 HOLD 模式下的功耗，PMU 模块的 BOR\_DET, VSYS\_DET, LVDINx\_DET 模块由芯片硬件控制自动进入分时开启模式。

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，则芯片从 HOLD 模式下唤醒，并进入相应的中断处理程序。

在 HOLD 模式下，WDT 默认开启，WDT 计数溢出时，系统会发生 WDT 复位。

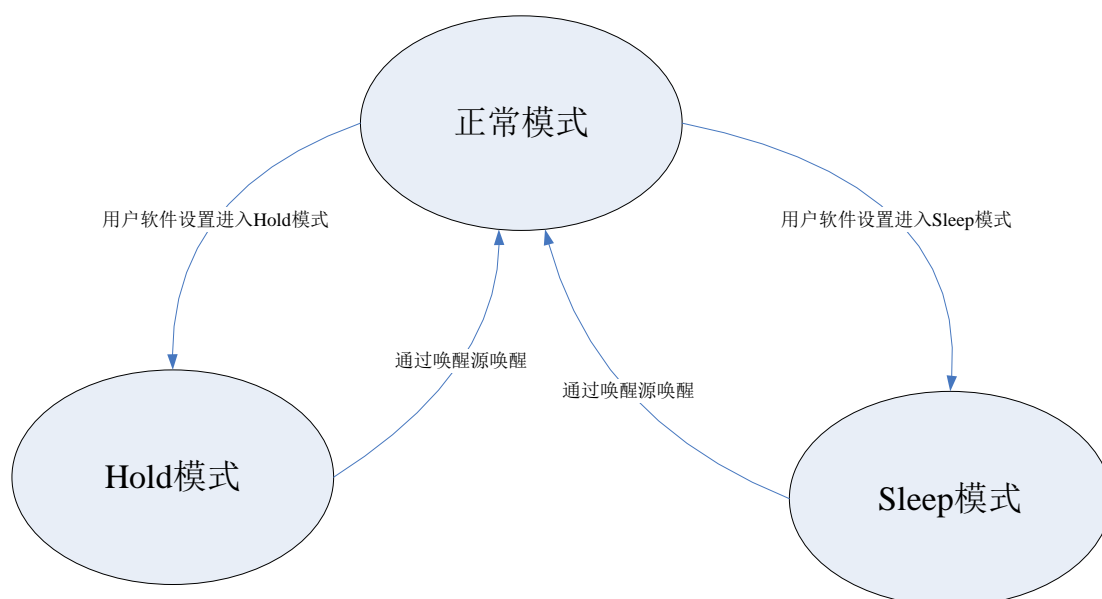
### 6.4.1 进入 HOLD 模式

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	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)

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:	X	X	X	X	X	SLEEPD	X	X
Write:	X	X	X	X	X	EPP	X	X
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

### WAKEIF(唤醒标志寄存器)

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:	NMIWK IF	INT9WK IF	INT8WK IF	INT7WK IF	KEYWK IF	DMAW KIF	EMUWK IF	SelfTest
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

	<b>Bit23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>Bit16</b>
<b>Read:</b>	X	SPIWKI F	I2CWKI F	RTCWKI F	TBSWKI F	TMR3W KIF	TMR2W KIF	TMR1W KIF
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0
	<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>	TMR0W KI	RX5WKI F	RX4WKI F	RX3WKI F	RX2WKI F	RX1WKI F	RX0WKI F	INT6WK IF
<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>	INT5WK IF	INT4WK IF	INT3WK IF	INT2WK IF	INT1WK IF	INT0WK IF	ARGEW KIF	PMUWK IF
<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
NMIWKIF	<b>NMI唤醒标志</b> SLEEP/HOLD 模式下,NMI 唤醒发生时, 设置标志位为 1
INT9WKIF	<b>INT9唤醒标志</b> SLEEP/HOLD模式下,INT9唤醒发生时, 设置标志位为1
INT8WKIF	<b>INT8唤醒标志</b> SLEEP/HOLD模式下,INT8唤醒发生时, 设置标志位为1
INT7WKIF	<b>INT7唤醒标志</b> SLEEP/HOLD模式下,INT7唤醒发生时, 设置标志位为1
KEYWKIF	<b>按键扫描唤醒标志</b> SLEEP/HOLD模式下,KEY唤醒发生时, 此位置为1
DMAWKIF	<b>DMA唤醒标志</b> HOLD模式下,DMA唤醒发生时, 此位置为1
EMUWKIF	<b>EMU唤醒标志</b> HOLD模式下,EMU唤醒发生时, 此位置为1
SelfTest	自测频中断, 不能唤醒, 该位恒为0
SPIWKIF	<b>SPI唤醒标志</b> HOLD模式下,SPI唤醒发生时, 此位置为1
I2CWKIF	<b>I2C唤醒标志</b> HOLD模式下,I2C唤醒发生时, 此位置为1
RTCWKIF	<b>RTC唤醒标志</b> SLEEP/HOLD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	<b>TBS唤醒标志</b> SLEEP/HOLD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)
TMR3WKIF	<b>TMR3 唤醒标志</b>

	HOLD模式下,TMR3唤醒发生时, 此位置为1
TMR2WKIF	<b>TMR3 唤醒标志</b> HOLD模式下,TMR2唤醒发生时, 此位置为1
TMR1WKIF	<b>TMR3 唤醒标志</b> HOLD模式下,TMR1唤醒发生时, 此位置为1
TMR0WKIF	<b>TMR3 唤醒标志</b> HOLD模式下,TMR0唤醒发生时, 此位置为1
RX5WKIF	<b>RX5唤醒标志</b> SLEEP/HOLD模式下,RX5唤醒发生时, 设置标志位为1
RX4WKIF	<b>RX4唤醒标志</b> SLEEP/HOLD模式下,RX4唤醒发生时, 设置标志位为1
RX3WKIF	<b>RX3唤醒标志</b> SLEEP/HOLD模式下,RX3唤醒发生时, 设置标志位为1
RX2WKIF	<b>RX2唤醒标志</b> SLEEP/HOLD模式下,RX2唤醒发生时, 设置标志位为1
RX1WKIF	<b>RX1唤醒标志</b> SLEEP/HOLD模式下,RX1唤醒发生时, 设置标志位为1
RX0WKIF	<b>RX0唤醒标志</b> SLEEP/HOLD模式下,RX0唤醒发生时, 设置标志位为1
EXTI6WKIF	<b>INT6唤醒标志</b> SLEEP/HOLD模式下,INT6唤醒发生时, 设置标志位为1
EXTI5WKIF	<b>INT5唤醒标志</b> SLEEP/HOLD模式下,INT5唤醒发生时, 设置标志位为1
EXTI4WKIF	<b>INT4唤醒标志</b> SLEEP/HOLD模式下,INT4唤醒发生时, 设置标志位为1
EXTI3WKIF	<b>INT3唤醒标志</b> SLEEP/HOLD模式下,INT3唤醒发生时, 设置标志位为1
EXTI2WKIF	<b>INT2唤醒标志</b> SLEEP/HOLD模式下,INT2唤醒发生时, 设置标志位为1
EXTI1WKIF	<b>INT1唤醒标志</b> SLEEP/HOLD模式下,INT1唤醒发生时, 设置标志位为1
EXTI0WKIF	<b>INT0唤醒标志</b> SLEEP/HOLD模式下,INT0唤醒发生时, 设置标志位为1
<b>ARGEWKIF</b>	<b>AES/RAND/GHASH/ECC 唤醒标志</b> HOLD模式下, <b>AES/RAND/GHASH/ECC</b> 唤醒发生时, 此位置为1
PMUWKIF	<b>PMU唤醒标志</b> SLEEP/HOLD模式下,PMU事件发生时将会产生PMU唤醒, 此位置为1 (具体那个PMU唤醒源需要查看PMUIF寄存器)

注:

- 1.Sleep 唤醒和 Hold 唤醒共用此标志位
- 2.**DMAWKIF /EMUWKIF/ SPIWKIF/I2CWKIF/TMR3WKIF/ TMR2WKIF/ TMR1WKIF/ TMR0WKIF/ARGEWKIF**, 只能 Hold 下才生效, sleep 模式不能唤醒。

3.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.

## 7 GPIO 模块

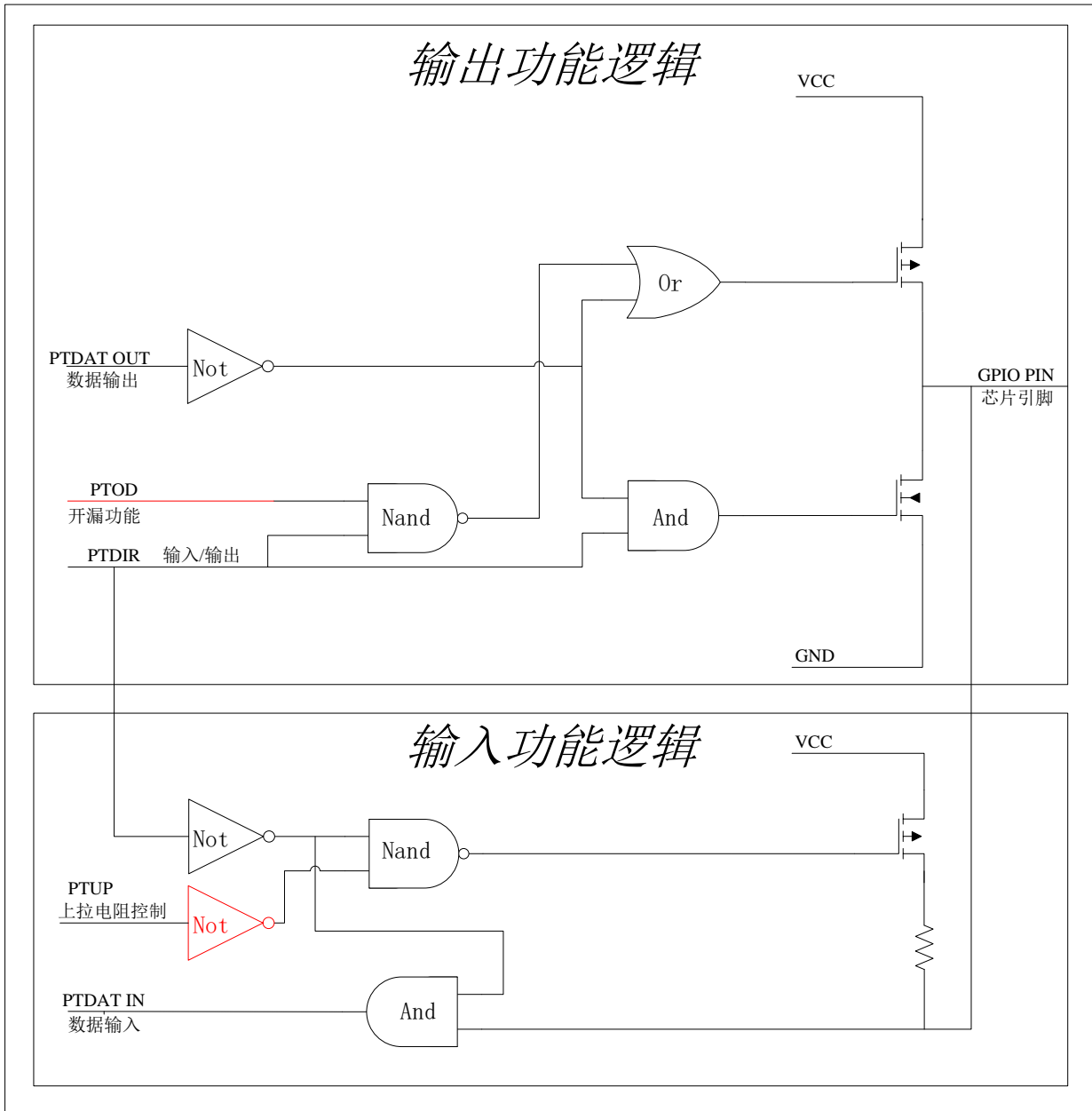
### 7.1 概述

HT5023提供PA[0..13], PB[0..15], PC[0..14], PD[0..15], PE[0..9], PF[0..7]并行端口, 支持79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。

HT5025提供PA[3..8], PA[11..13], PB[0..15], PC[0..3], PC8, PC[10..12], PD[0..6], PD[8..15], PE[1..5], PE[7..9], PF[0..2]并行端口, 支持59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。

HT5027 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。

## 7.2 芯片引脚结构说明



## 7.3 芯片引脚描述

详见章节<1.6 引脚定义>。

## 7.4 I/O 端口基地址列表

GPIO 模块寄存器基地址: 0x40011000(PA 端口); 0x40011200(PC 端口); 0x40011300(PD 端口); 0x40011400(PE 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
0x04	AFCFG	R/W	0x0000	端口功能配置寄存器 2
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: 0x40011100(PB 端口);				
偏移地址	名称	读写方式	复位值	功能描述
0x00	IOCFG	R/W	0xA000	端口功能配置寄存器 1
0x04	AFCFG	R/W	0xA000	端口功能配置寄存器 2
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

注:

PB15/SEG15/SWCLK 的复位默认功能配置为 SWCLK

PB13/SEG13/SWIO 的复位默认功能配置为 SWIO

即,HT\_GPIOB->IOCFG 复位默认值 0xA000

HT\_GPIOB->AFCFG 复位默认值 0xA000

GPIO 模块寄存器基地址: 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	0x0007	端口功能配置寄存器 1
0x04	AFCFG	R/W	0x0000	端口功能配置寄存器 2
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	<b>0xFFFF</b>	端口 OD 功能配置寄存器

注：

PF.0/PFOUT/TOUT3 的复位默认功能配置为 PFOUT

PF.1/QFOUT 的复位默认功能配置为 QFOUT

PF.2/SFOUT/SEG48 的复位默认功能配置为 SFOUT

即，HT\_GPIOF->IOCFG 复位默认值 0x0007

HT\_GPIOF->AFCFG 复位默认值 0x0000

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(端口功能配置寄存器 1)

<b>IOCFG</b> (端口功能配置寄存器 1)		<b>基地址： 0x40011000--0x40011400</b>						
		<b>偏移地址： 00H</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>	PT[15:8]							
<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>	PT[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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(端口功能配置寄存器 2)**

<b>AFCFG</b> (端口功能配置寄存器 2)			基地址: 0x40011000~0x40011400 偏移地址: 04H					
	<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>	PT[15:8]							
<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>	PT[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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)**

<b>PTDIR</b> (端口方向配置寄存器)			基地址: 0x40011000~0x40011400 偏移地址: 08H					
	<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>	PT[15:8]							
<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>	PT[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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)**

<b>PTUP</b> (端口上拉配置寄存器)			<b>基地址: 0x40011000–0x40011400</b>					
			<b>偏移地址: 0CH</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>	PT[15:8]							
<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>	PT[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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）

注:

1. 芯片引脚选择 GPIO 功能:

若方向寄存器配置为输出，上拉控制无效;

若方向寄存器配置为输入，上拉控制有效;

2. 芯片引脚选择复用功能:

复用功能的数字输出引脚上拉功能配置无效。

复用功能的数字输入引脚（除 RST/TEST/JTAGWDTEN，这三个引脚内部恒定上拉）都可配上拉功能;

若配置为模拟输入上拉控制无效;

3. 复用功能为模拟功能: LVDIN<sub>x</sub>、LCD\_SEG\COM、POWIN、ADCIN<sub>x</sub>、VBAT,除此之外的其它复用功能为数字功能。

**PTDAT(端口数据寄存器)**

<b>PTDAT</b> (端口数据寄存器)			<b>基地址: 0x40011000–0x40011400</b>					
			<b>偏移地址: 10H</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>	PT[15:8]							
<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>	PT[7:0]							
<b>Write:</b>								

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

注：端口数据寄存器 PTDAT 说明及数据读取

1. 芯片引脚选择 GPIO 功能或复用数字功能

若方向寄存器配置为输出，PTDAT 读取值为寄存器设置值，不随外部 PIN 脚电平变化而变化；

若方向寄存器配置为输入，PTDAT 读取值为 pad 状态值，反映外部 PIN 脚电平变化；

2. 芯片引脚选择复用模拟功能，PTDAT 相应 bit 位值，固定为 0

## PTSET(端口设置寄存器)

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)

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

<b>Read:</b>	X	X	X	X	X	X	X	X
<b>Write:</b>	PT[15:8]							
<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	X	X
<b>Write:</b>	PT[7:0]							
<b>Reset:</b>	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)

<b>PTTOG</b> (端口翻转寄存器)			<b>基地址: 0x40011000--0x40011400</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>	PT[15:8]							
<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	X	X
<b>Write:</b>	PT[7:0]							
<b>Reset:</b>	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)

<b>PTOD</b> (端口开漏配置寄存器)			<b>基地址: 0x40011000--0x40011400</b> <b>偏移地址: 20H</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>	PT[15:8]							
<b>Write:</b>	PT[15:8]							
<b>Reset:</b>	1	1	1	1	1	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>	PT[7:0]							
<b>Write:</b>								
<b>Reset:</b>	1	1	1	1	1	1	1	1

位	功能描述
PT[15:0]	<p>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）</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>

注：

1. 芯片引脚选择 GPIO 功能：

若方向寄存器配置为输出，开漏 OD 功能控制有效；

若方向寄存器配置为输入，开漏 OD 功能控制无效；

2. 芯片引脚选择复用功能：

复用功能的数字输出引脚都可配开漏功能(Open Drain)。

复用功能的数字输入引脚开漏功能无效；

若配置为模拟输入开漏 OD 控制无效；

3. 复用功能为模拟功能：LVDIN<sub>x</sub>、LCD\_SEG\COM、POWIN、ADCIN<sub>x</sub>、VBAT,除此之外的其它复用功能为数字功能。

## HDPORT(大电流端口配置寄存器)

<b>HDPORT</b> (大电流端口配置寄存器)		基地址： 0x40011500 偏移地址： 00H						
	<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	PA8HD	PA7HD	PA6HD	PC0HD
<b>Write:</b>								
<b>Reset:</b>	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>端口大电流驱动配置（分别对应 PA8,PA7,PA6,和 PC0 端口）</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 IRQn	中断使能	中断标志	功能描述
NMI	-14			硬件强制切换低频 RC 时钟 Flrc 中断
HardFault	-13			故障/异常触发中断
SVCcall	-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
以上为内核自带				
PMU	0	PMUIE.LVDIN0IE	PMUIF.LVDIN0IF	LVD检测中断
		PMUIE.LVDIN1IE	PMUIF.LVDIN1IF	LVD检测中断
		PMUIE.BORIE	PMUIF.BORIF	BOR检测中断
		PMUIE.VSYSIE	PMUIF.VSYSIF	VSYS检测中断
AES/ECC/RAND/GHASH	1	AESGHASHIE.AESIE	AESGHASHIF.AESIF	AES 中断
		AESGHASHIE.GHASHIE	AESGHASHIF.GHASHIF	GHASH 中断
		AESGHASHIE.RANDIE	AESGHASHIF.RANDIF	RAND 中断
		ECCCON.ECCIE	ECCSTA.ECCFLG	ECC 中断
EXTI0-6	2-8	EXTIE.RIE[0:6]	EXTIF.RIF[0:6]	外部输入引脚上升沿中断
		EXTIE.FIE[0:6]	EXTIF.FIF[0:6]	外部输入引脚下降沿中断
UART0-5 (UART 3 和 UART 4 有 7816 功能)	9-14	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)
TMR0-3	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.ADC1CMP IE	TBSIF.ADC0CMP IF	ADC1 比较中断
		TBSIE.ADC0CMP	TBSIF.ADC0CMP IF	ADC0 比较中断

		IE		
		TBSIE.VBATCMP IE	TBSIF.VBATCMP PIF	电池电压比较中断
		TBSIE.ADC2IE	TBSIF.ADC2IF	ADC 通道 2 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.VDDIE	TBSIF.VDDIF	电源电压 VDD 测量中断
		TBSIE.VBATIE	TBSIF.VBATIF	电池电压 VDD 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	I2C 中断
SPI	22		SPIF	SPI 传输完成中断
			MODF	SPI 模式故障中断
-	23	-	-	-
-	24	-	-	-
EMU	25	EMUIE	子中断标志 EMUIF 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
EXTI7-9	28-30	EXTIE2.RIE[2:0]	EXTIF2.RIF[2:0]	外部输入引脚上升沿中断
		EXTIE2.FIE[2:0]	EXTIF2.FIF[2:0]	外部输入引脚下降沿中断

注:

1. 只有 UART3 和 UART4 有 7816 功能, 因此只有这两个通讯端口有 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 22.4 CMSIS 函数说明。
3. NMI/HardFault/SVCall/PendSV/SysTick 为内核自带中断, 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)           //中断使能
void NVIC_DisableIRQ(IRQn_Type IRQn)         //中断关闭
void NVIC_SetPendingIRQ(IRQn_Type IRQn)      //设置中断挂起
void NVIC_ClearPendingIRQ(IRQn_Type IRQn)    //清除中断挂起
uint32_t NVIC_GetPendingIRQ(IRQn_Type IRQn) //获取中断挂起状态
void NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority) //中断优先级设置
uint32_t NVIC_GetPriority(IRQn_Type IRQn)     //中断优先级获取
    
```

## 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	IPR0-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 (中断使能配置寄存器)		寄存器地址: 0xE000E100							
		<b>Bit31...Bit0</b>							
<b>Read:</b>	SETENA[31:0]								
<b>Write:</b>	SETENA[31:0]								
<b>Reset:</b>	0	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 EXTI0  
 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							
		<b>Bit31...Bit0</b>							
<b>Read:</b>	CLRENA[31:0]								
<b>Write:</b>	CLRENA[31:0]								

<b>Reset:</b>	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。

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

<b>ICPR</b> (中断标志清零寄存器)	<b>寄存器地址: 0xE000E280</b>							
	<b>Bit31...Bit0</b>							
<b>Read:</b>	CLRPEND[31:0]							
<b>Write:</b>	CLRPEND[31:0]							
<b>Reset:</b>	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

<b>IPR0 --- IPR7</b> (中断优先级寄存器)	<b>寄存器地址: 0xE000E400--0xE000E417</b>							
	<b>Bit31...Bit0</b>							
<b>Read:</b>	Priority [31:0]							
<b>Write:</b>	Priority [31:0]							
<b>Reset:</b>	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 外部中断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	0x007F	引脚数字滤波器使能
<b>0x10</b>	<b>EXTIE2</b>	<b>R/W</b>	<b>0x0000</b>	<b>外部中断边沿配置寄存器 2</b>

0x14	EXTIF2	R/W	0x0000	外部中断标志寄存器 2
0x18	PINFLT2	R/W	0x0007	引脚数字滤波使能寄存器 2

## 8.4 外部中断特殊功能寄存器说明

### EXTIE(外部中断边沿配置寄存器)

<b>EXTIE</b> (外部中断边沿配置寄存器)		基地址: <b>0x40011800</b> 偏移地址: <b>00H</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	RIE[6:0]						
<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	FIE[6:0]						
<b>Write:</b>								
<b>Reset:</b>	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)

<b>EXTIF</b> (外部中断标志寄存器)		基地址: <b>0x40011800</b> 偏移地址: <b>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>	X	RIF[6:0]						
<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	FIF[6:0]						
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

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

### PINFLT(外部引脚滤波器配置寄存器)

<b>PINFLT</b> (外部引脚滤波器配置寄存器)		<b>基地址: 0x40011800</b> <b>偏移地址: 08H</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	RXFLT[5:0]					
<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	INTFLT[6:0]						
<b>Write:</b>								
<b>Reset:</b>	0	1	1	1	1	1	1	1

位	功能描述
RXFLT[5:0]	<b>RX 外部输入引脚数字滤波功能使能(RX0~RX5)</b> 0: 禁止 1: 使能
INTFLT[6:0]	<b>INT 外部输入引脚数字滤波功能使能(INT0~INT6)</b> 0: 禁止 1: 使能

注：1. RX0~5 和 INT0~6，除了数字滤波功能，内部还集成模拟滤波电路（2us），模拟滤波仅对下降沿有效，上升沿无效；数字滤波对上升沿和下降沿都有效；

2. 用户在使用 INT 外部中断引脚功能的时候，必须要将 PINFLT/PINFLT2 寄存器中的相应的引脚数字滤波功能打开。

### EXTIE2(外部中断边沿配置寄存器 2)

<b>EXTIE2</b> (外部中断边沿配置寄存器 2)		<b>基地址: 0x40011800</b> <b>偏移地址: 10H</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	RIE[2:0]		
<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	FIE[2:0]		
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
RIE[2:0]	INT7-9 外部输入引脚上升沿中断使能 0: 禁止 1: 使能
FIE[2:0]	INT7-9 外部输入引脚下降沿中断使能 0: 禁止 1: 使能

## EXTIF2(外部中断标志寄存器 2)

<b>EXTIF2</b> (外部中断标志寄存器 2)			<b>基地址: 0x40011800</b> <b>偏移地址: 14H</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	RIF[2:0]		
<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	FIF[2:0]		
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
RIF[2:0]	INT7-9 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[2:0]	INT7-9 外部输入引脚下降沿中断标志 0: 未产生中断 1: 产生中断

**PINFLT2(外部引脚滤波器配置寄存器 2)**

<b>PINFLT2</b> (外部引脚滤波器配置寄存器 2)			<b>基地址: 0x40011800</b> <b>偏移地址: 18H</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>								
<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	INTFLT[2:0]		
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	1	1	1

位	功能描述
INTFLT[2:0]	<b>INT7-9 外部输入引脚数字滤波功能使能</b> 0: 禁止 1: 使能

注：1. INT7~9，除了数字滤波功能，内部还集成模拟滤波电路（2us），模拟滤波仅对下降沿有效，上升沿无效；数字滤波对上升沿河下降沿都有效；

2. 用户在使用 INT 外部中断引脚功能的时候，必须要将 PINFLT/PINFLT2 寄存器中的相应的引脚数字滤波功能打开。

## 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 模块，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, RTC 模块的寄存器 3, LCD 相关寄存器 LCD_BUF[i]
2	低电压检测复位 (LBOR)		
3	外部引脚/RST 复位	二级	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, LCD 相关寄存器 LCD_BUF[i]
4	软件 POR/LBOR 复位		
5	掉电复位 (BOR)		
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 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
8	唤醒复位 (WakeUp Reset)		
9	软件复位 (Software reset)		

注:1.LRCADJ 会被 Watchdog 和 Debug Reset 复位

2.RSTSR 的 POR 复位标志和 LBOR 复位标志之间可以互相清除

3.所有的复位均会复位掉 CLKCTRL0 的 HRC\_EN 信号，复位之后系统跑 HRC (SYSCLK\_SEL[2:0]=010)。

## 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：

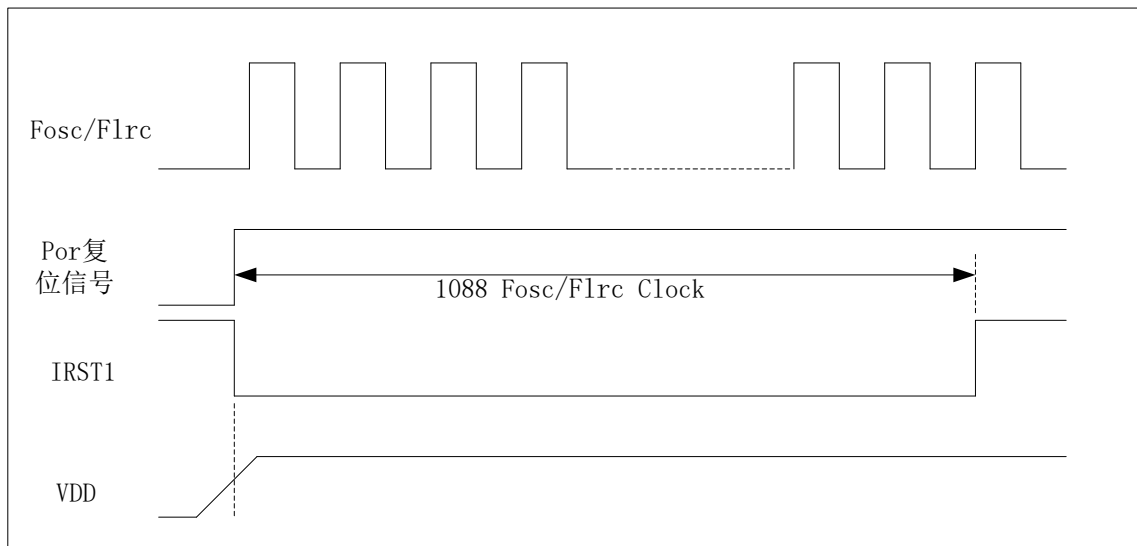
- 1) POR 和 LBOR 复位时，内部复位信号 IRST1 将保持有效，并保持 1088 个 Fosc/Flrc。
- 2) BOR 和外部 RST 复位时，内部复位信号 IRST2 将保持有效，并保持 1088 个 Fosc/Flrc。
- 3) WDT Reset 复位时，内部复位信号 IRST3 将保持有效，并保持 64 个 Fosc/Flrc。
- 4) SoftRese 和 Debug Reset 复位时，内部 IRST4 信号有效，并保持 64 个 Fosc/Flrc。
- 5) 只有 POR 复位会对 RTC 复位

复位预热计数的时钟源选择(Fosc/Flrc)，由 Fosc 停振检测模块(LF\_DET)的停振标志 LF\_FLAG 控制，LF\_FLAG=0，选择 Fosc 时钟；LF\_FLAG=1，选择 Flrc 时钟；若 Fosc 停振检测关闭，LF\_FLAG 固定为 0，复位计数时钟强制选择 Fosc，此时 Fosc 停振，会影响复位预热计数功能，建议系统正常运行打开 Fosc 停振检测，增强复位可靠性。

### 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 VDD 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：

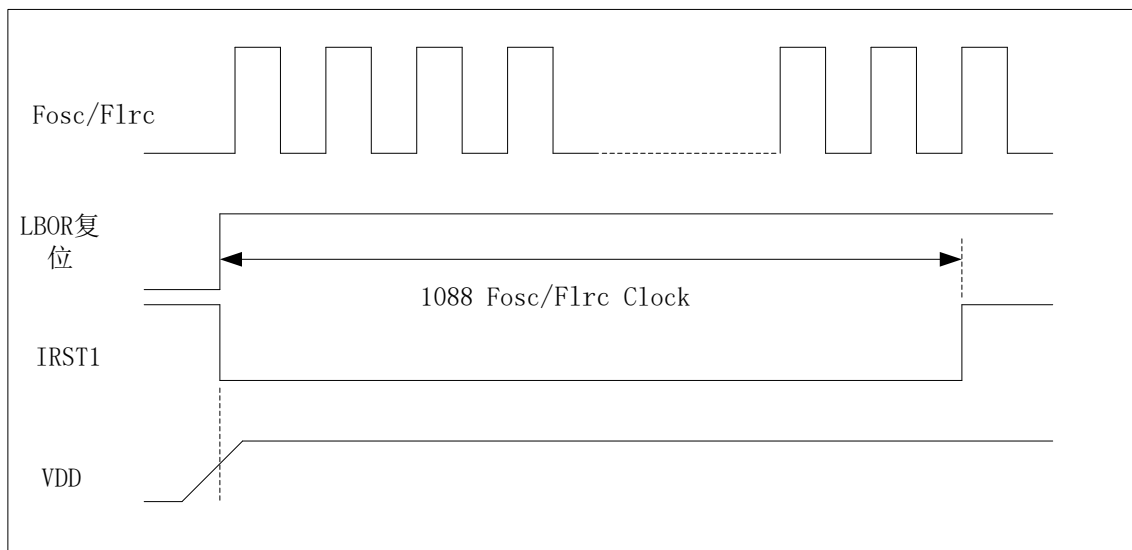
- POR 输出高电平
- 第一次 POR 复位会将 RTC 模块复位（RTC 的复位只能由 VRTC 的电源的上电 POR 复位）
- 内部复位信号 IRST1 有效
- 计数 1088 个 Fosc/Flrc
- 复位状态寄存器 RSTSTA 的上电复位标志位 POR 被设置为 1，其他 RSTSTA 为被清为 0。
- CPU 从地址 0000H 执行程序



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。

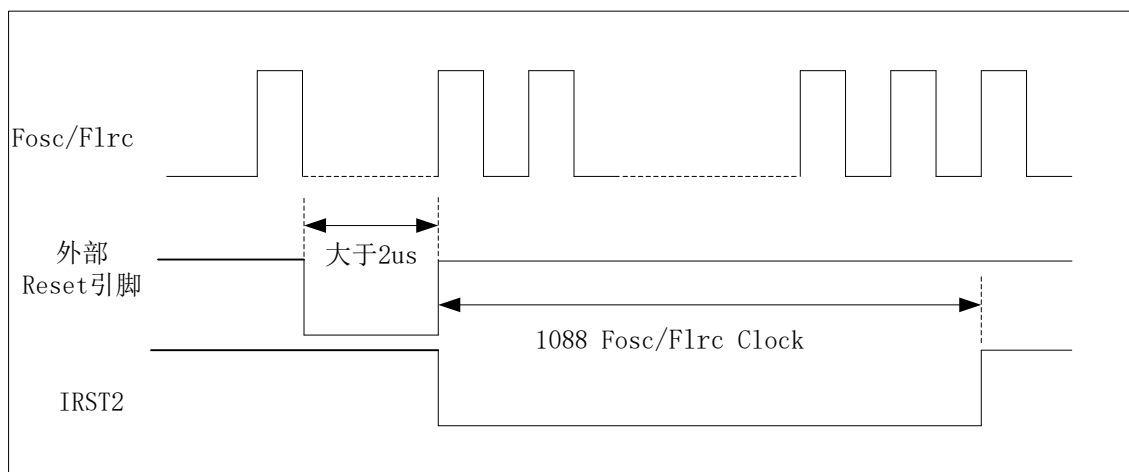


LBOR 复位说明

## 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

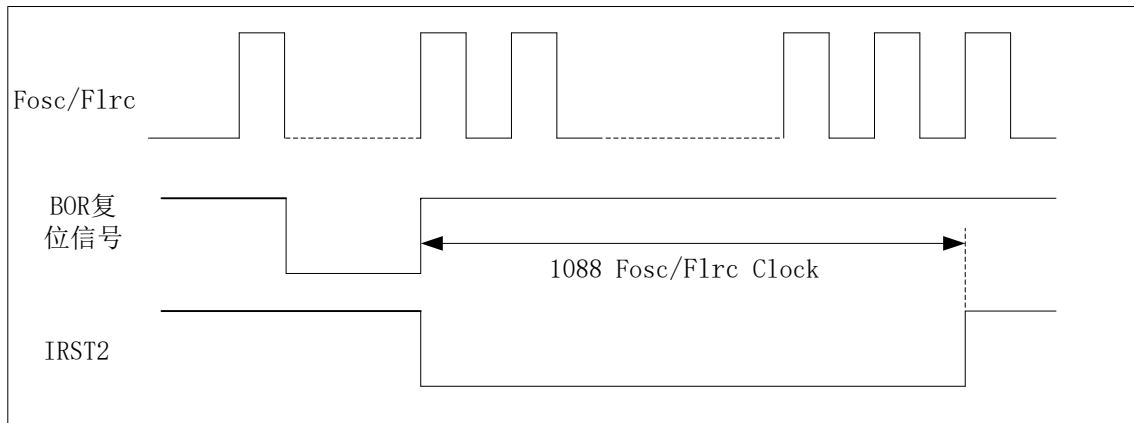
## 9.2.4. 掉电复位

当掉电检测电路检测到电源电压低于电压  $V_{bor1}$  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  $V_{borh}$  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：

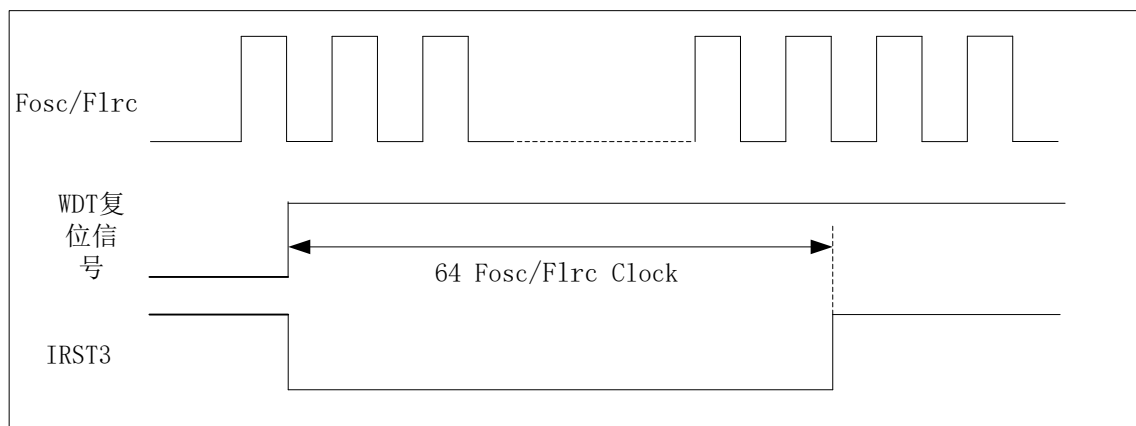
- 产生一个 BOR 脉冲
- 内部复位信号 IRST2 有效
- 计数 1088 个 Fosc/Flrc
- 复位状态寄存器 RSTSTA 的掉电复位标志位 BOR 被设置为 1，其他 RSTSTA 为被清为 0。
- CPU 从 0000H 开始执行程序



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 1088 个 Fosc/Flrc。

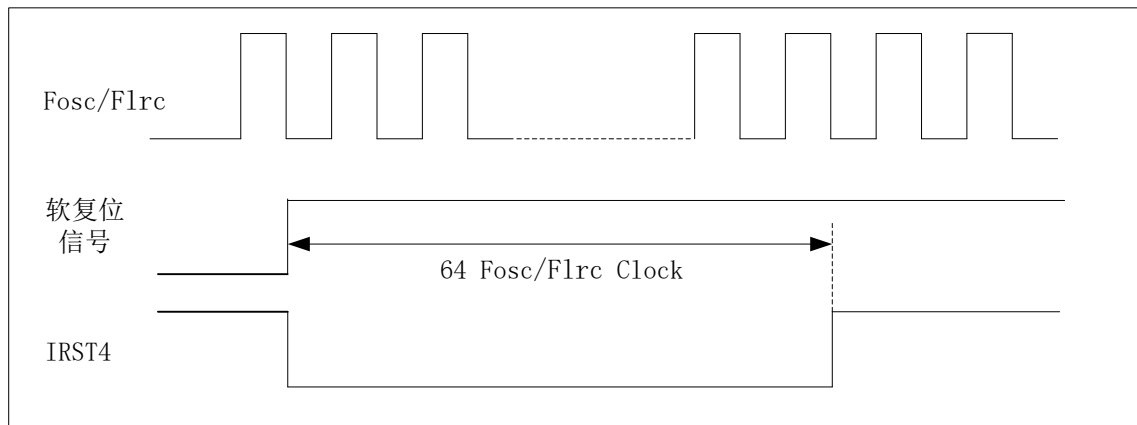


WDT 复位

## 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

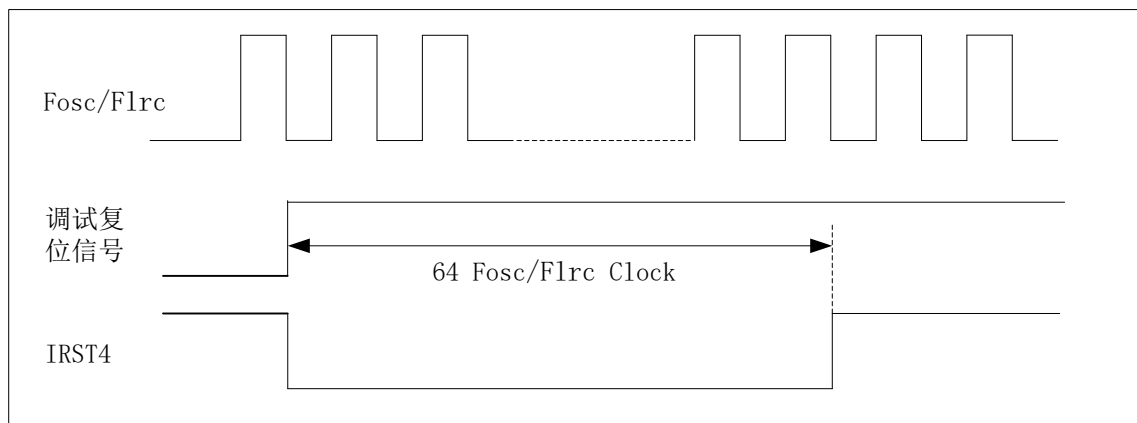
(Cortex-M0 自带功能)。



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。

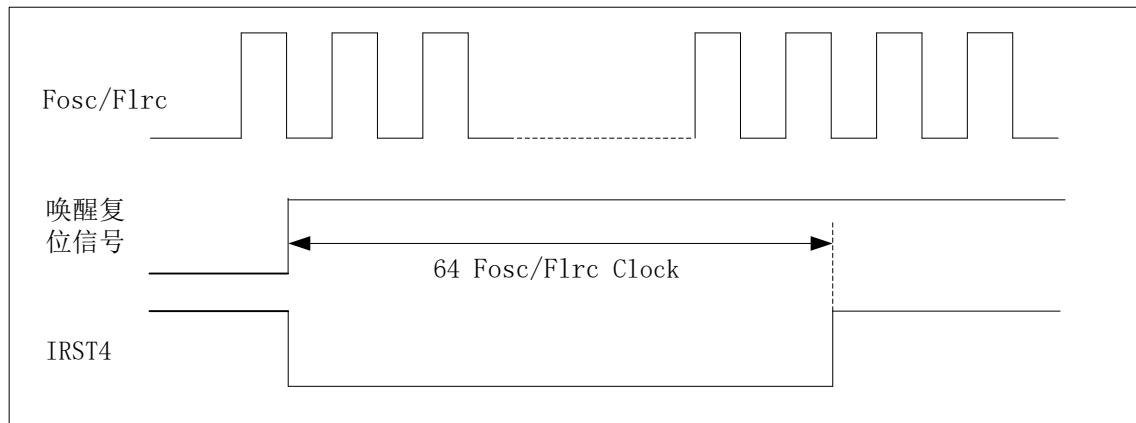


调试复位

## 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	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)

RSTSTA (复位标志寄存器)		基地地址: 0x4000F400 偏移地址: 30H						
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	Hold_Fla	Sleep_F	X	X	X	X	X	BORRST
<b>Write:</b>	g	lag						
<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>	DebugRS	SoftRS	ExtRST	FlashWR	Wakeup	WDTR	LBORRS	PORRST
<b>Write:</b>	T	T		_LBOR	RST	ST	T	
<b>Reset:</b>	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
--	--

注: 当 LBOR 和 POR 复位发生时, 会将其它复位标志全部进行复位;  
 除 LBOR 和 POR 两种复位之外的其它复位, 不会清除寄存器的已有的复位标志。

### WAKEIF(唤醒标志寄存器)

WAKEIF (唤醒标志寄存器)		基地址: 0x400F400 偏移地址: 18H						
	<b>Bit31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>Bit24</b>
<b>Read</b>	NMIWKI F	INT9WK IF	INT8WK IF	INT7WK IF	KEYWK IF	DMAWKI F	EMUWKI F	RSRV
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset</b>	0	0	0	0	0	0	0	0
	<b>Bit23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>Bit16</b>
<b>Read</b>	X	SPIWKIF	I2CWKI F	RTCWKI F	TBSWKI F	TMR3WK IF	TMR2WK IF	TMR1WK IF
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset</b>	0	0	0	0	0	0	0	0
	<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>	TMR0WK IF	RX5WKI F	RX4WKI F	RX3WKI F	RX2WKI F	RX1WKI F	RX0WKIF	INT6WKI F
<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>	INT5WKI F	INT4WK IF	INT3WK IF	INT2WK IF	INT1WK IF	INT0WKI F	ARGEWK IF	PMUWKI F
<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
NMIWKIF	NMI唤醒标志

	SLEEP/HOLD 模式下,NMI 唤醒发生时, 设置标志位为 1
<b>INT9WKIF</b>	<b>INT9唤醒标志</b> SLEEP/HOLD模式下,INT9唤醒发生时, 设置标志位为1
<b>INT8WKIF</b>	<b>INT8唤醒标志</b> SLEEP/HOLD模式下,INT8唤醒发生时, 设置标志位为1
<b>INT7WKIF</b>	<b>INT7唤醒标志</b> SLEEP/HOLD模式下,INT7唤醒发生时, 设置标志位为1
<b>KEYWKIF</b>	<b>按键扫描唤醒标志</b> SLEEP/HOLD模式下,KEY唤醒发生时, 此位置为1
<b>DMAWKIF</b>	<b>DMA唤醒标志</b> HOLD模式下,DMA唤醒发生时, 此位置为1
<b>EMUWKIF</b>	<b>EMU唤醒标志</b> HOLD模式下,EMU唤醒发生时, 此位置为1
<b>RSRV</b>	不能唤醒, 该位恒为0
<b>SPIWKIF</b>	<b>SPI唤醒标志</b> HOLD模式下,SPI唤醒发生时, 此位置为1
<b>I2CWKIF</b>	<b>I2C唤醒标志</b> HOLD模式下,I2C唤醒发生时, 此位置为1
<b>RTCWKIF</b>	<b>RTC唤醒标志</b> SLEEP/HOLD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)
<b>TBSWKIF</b>	<b>TBS唤醒标志</b> SLEEP/HOLD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)
<b>TMR3WKIF</b>	<b>TMR3 唤醒标志</b> HOLD模式下,TMR3唤醒发生时, 此位置为1
<b>TMR2WKIF</b>	<b>TMR3 唤醒标志</b> HOLD模式下,TMR2唤醒发生时, 此位置为1
<b>TMR1WKIF</b>	<b>TMR3 唤醒标志</b> HOLD模式下,TMR1唤醒发生时, 此位置为1
<b>TMR0WKIF</b>	<b>TMR3 唤醒标志</b> HOLD模式下,TMR0唤醒发生时, 此位置为1
<b>RX5WKIF</b>	<b>RX5唤醒标志</b> SLEEP/HOLD模式下,RX5唤醒发生时, 设置标志位为1
<b>RX4WKIF</b>	<b>RX4唤醒标志</b> SLEEP/HOLD模式下,RX4唤醒发生时, 设置标志位为1
<b>RX3WKIF</b>	<b>RX3唤醒标志</b> SLEEP/HOLD模式下,RX3唤醒发生时, 设置标志位为1
<b>RX2WKIF</b>	<b>RX2唤醒标志</b> SLEEP/HOLD模式下,RX2唤醒发生时, 设置标志位为1
<b>RX1WKIF</b>	<b>RX1唤醒标志</b> SLEEP/HOLD模式下,RX1唤醒发生时, 设置标志位为1
<b>RX0WKIF</b>	<b>RX0唤醒标志</b>

	SLEEP/HOLD模式下,RX0唤醒发生时, 设置标志位为1
EXTI6WKIF	<b>INT6唤醒标志</b> SLEEP/HOLD模式下,INT6唤醒发生时, 设置标志位为1
EXTI5WKIF	<b>INT5唤醒标志</b> SLEEP/HOLD模式下,INT5唤醒发生时, 设置标志位为1
EXTI4WKIF	<b>INT4唤醒标志</b> SLEEP/HOLD模式下,INT4唤醒发生时, 设置标志位为1
EXTI3WKIF	<b>INT3唤醒标志</b> SLEEP/HOLD模式下,INT3唤醒发生时, 设置标志位为1
EXTI2WKIF	<b>INT2唤醒标志</b> SLEEP/HOLD模式下,INT2唤醒发生时, 设置标志位为1
EXTI1WKIF	<b>INT1唤醒标志</b> SLEEP/HOLD模式下,INT1唤醒发生时, 设置标志位为1
EXTI0WKIF	<b>INT0唤醒标志</b> SLEEP/HOLD模式下,INT0唤醒发生时, 设置标志位为1
<b>ARGEWKIF</b>	<b>AES/RAND/GHASH/ECC 唤醒标志</b> HOLD模式下, <b>AES/RAND/GHASH/ECC</b> 唤醒发生时, 此位置为1
PMUWKIF	<b>PMU唤醒标志</b> SLEEP/HOLD模式下,PMU事件发生时将会产生PMU唤醒, 此位置为1 (具体那个PMU唤醒源需要查看PMUIF寄存器)

注: 1.Sleep 唤醒和 Hold 唤醒共用此标志位

2.DMAWKIF /EMUWKIF/ SPIWKIF/I2CWKIF/TMR3WKIF/ TMR2WKIF/ TMR1WKIF/ TMR0WKIF/ARGEWKIF, 只能 Hold 下才生效, sleep 模式不能唤醒。

3.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.

## AIRCR(应用中断与控制状态寄存器)

<b>AIRCR</b> (应用中断与控制状态寄存器)		<b>基地址: 0xE00ED00</b>						
		<b>偏移地址: 0CH</b>						
	<b>Bit31</b>	<b>30</b>	<b>29</b>	<b>28...11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>	
<b>Read:</b>	AIRCR [31:8]							
<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	SYSRES	RSRV	X
<b>Write:</b>						ETREQ		
<b>Reset:</b>	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
RSRV (BIT1)	请用户保持此位为 0。

## 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:

- 共六路 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, 最高波特率可达 1Mbps
- 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 UARTCON 配置串口工作在不同的工作模式, 列举如下:

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

6路UART的输出TX0~TX5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。

6 路 UART 的输入 PIN 配置为 RX0~RX5 复用功能 (详细见 GPIO 单元), 并使能 UART 中断 (NVIC\_EnableIRQ(UART0\_IRQn)), 可实现在 SLEEP 模式下的外部通信唤醒功能, 类似外部中断。

HOLD模式下UART唤醒, 为功能唤醒, 需开启UART模块, 接收/发送中断发生, 唤醒HOLD模式。

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
- 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：

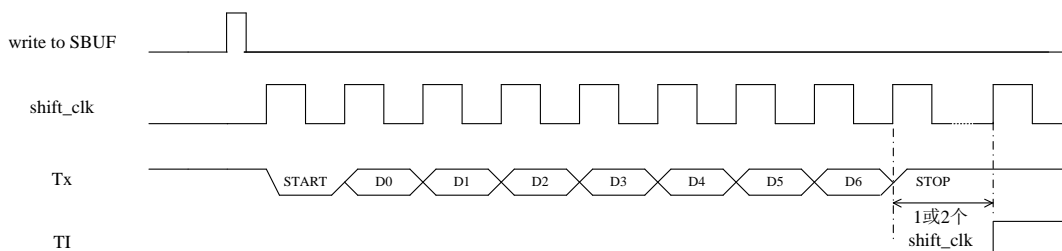
$$\text{波特率} = \frac{F_{\text{sys}}}{2 \times (SREL + 1)}$$

其中 SREL 是 16 位无符号数；F<sub>sys</sub> 是系统时钟。

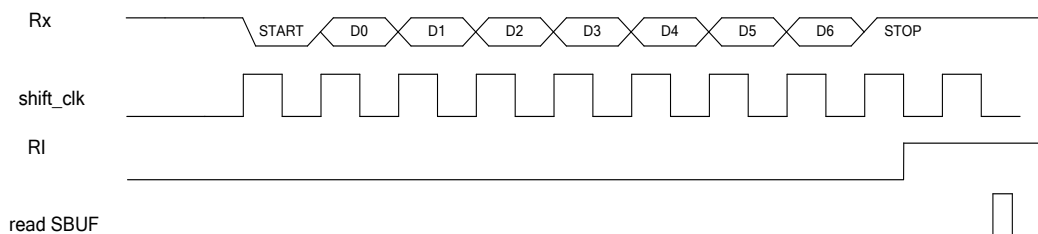
## 10.3 串口通讯模式说明

### 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，TX 引脚为数据发送端，RX 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

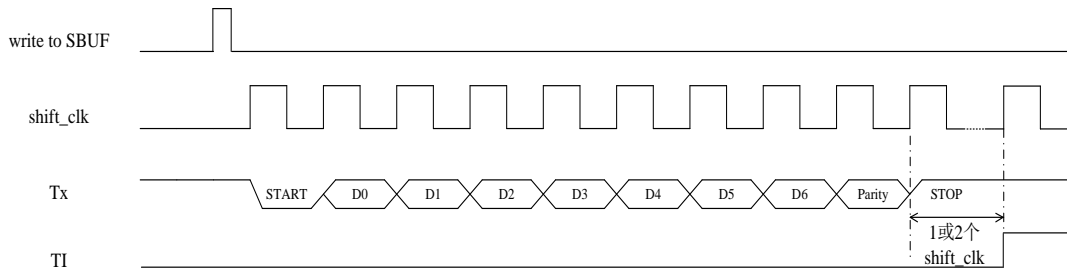


图示：方式 1 时串行接收数据时序

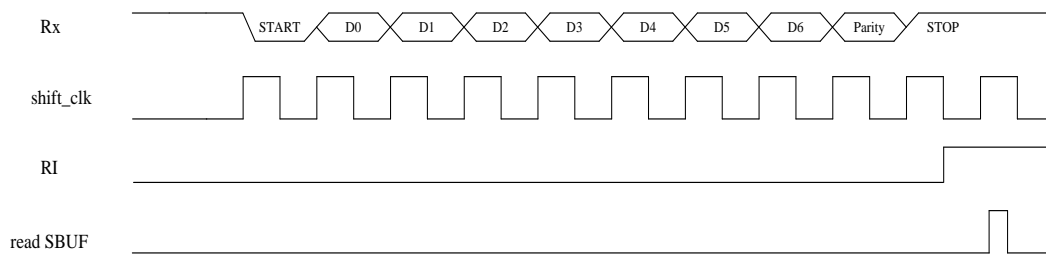
在方式 1 中，发送状态时，当一帧中最后一个数据发送完时，发送中断标志 TI 置“1”；接收状态时，接收完最后一个数据位时，接收中断标志 RI 置 1。

## 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)。TX 引脚为数据发送端，RX 引脚为数据接收端，其波形如下图所示：



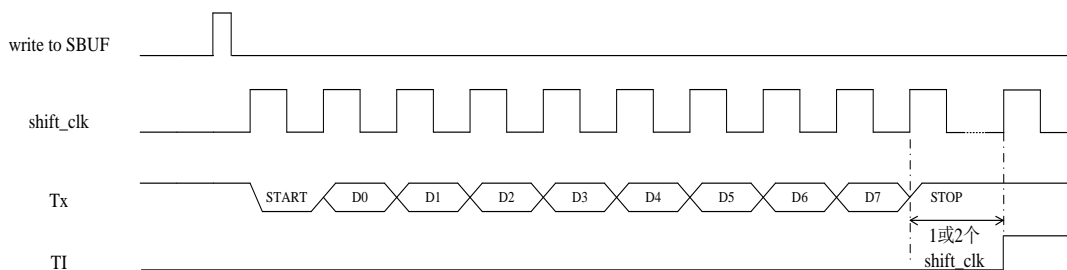
图示：方式 2 时串行发送数据时序



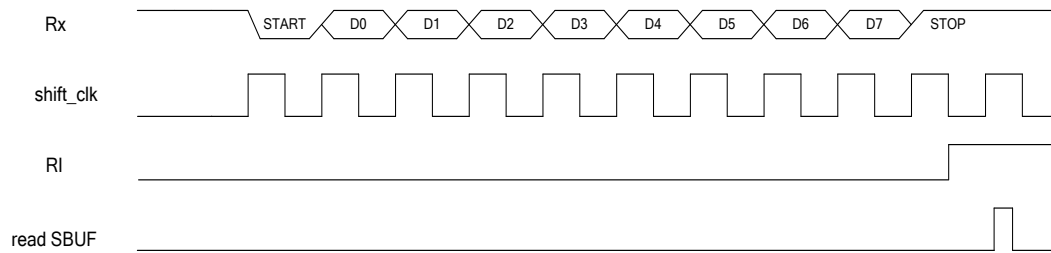
图示：方式 2 时串行接收数据时序

## 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，TX 引脚为数据发送端，RX 引脚为数据接收端，其波形如下图所示：



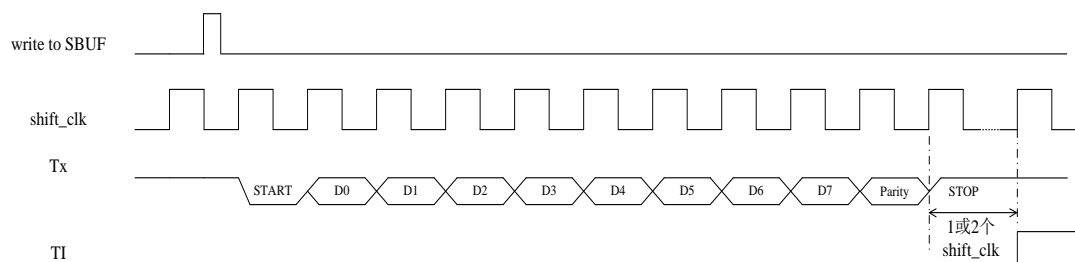
图示：方式 3 时串行发送数据时序



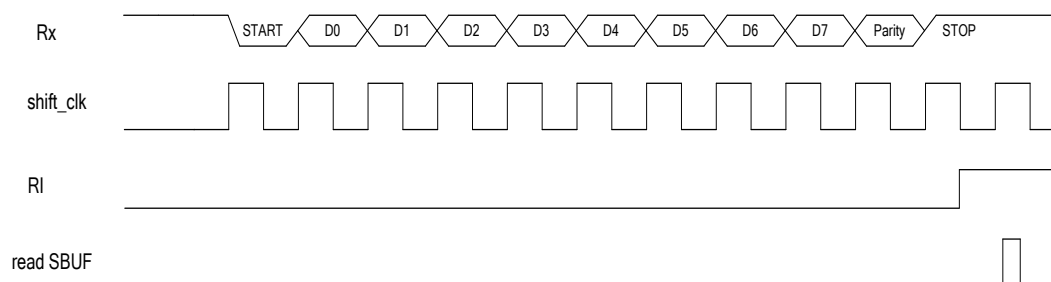
图示：方式 3 时串行接收数据时序

## 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）。TX 引脚为数据发送端，RX 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

## 10.47816 接收和发送

### 10.4.17816 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ACK信号; 如果ACK=0, TXERRSTAT置为“1”, 如果ACK=1, TXERRSTAT置为“0”; (第11ETU)
5. 处于发送等待状态; (第12-13ETU)
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ACK=1, 或禁止自动重发 (AUTOTXEN=0), 则UART接口回到IDLE 状态。  
如果ACK=0, 且使能自动重发 (AUTOTXEN=1), 则UART接口进入重发等待状态。
7. 重新开始发送一帧数据。

### 10.4.27816 数据接收

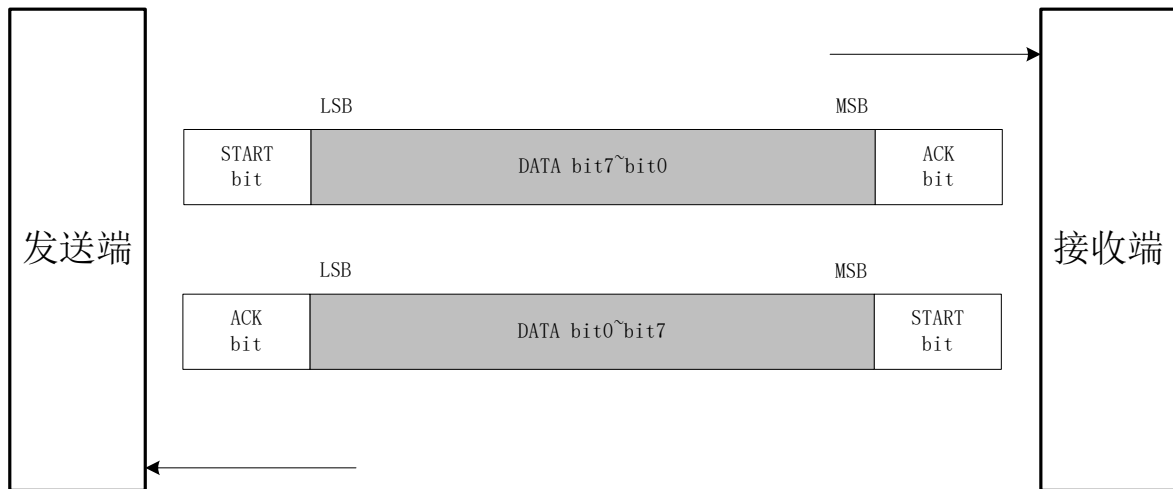
在IDLE状态下, 如果在7816端口(IO)上检测到下降沿, 即启动一次接收数据流程。该流程包括几个步骤, 每一步均需要一个或者几个ETU。

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. 验证校验位; (第10.5ETU)
5. (第10.5ETU) 向发送端发送ACK信号: 如果校验正确, 或者禁止自动重收 (AUTORXEN=0), 则发送1, 否则发送0。(ACK的宽度, 可以通过ACKLEN来配置)

校验位	AUTORXEN	7816IO
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”

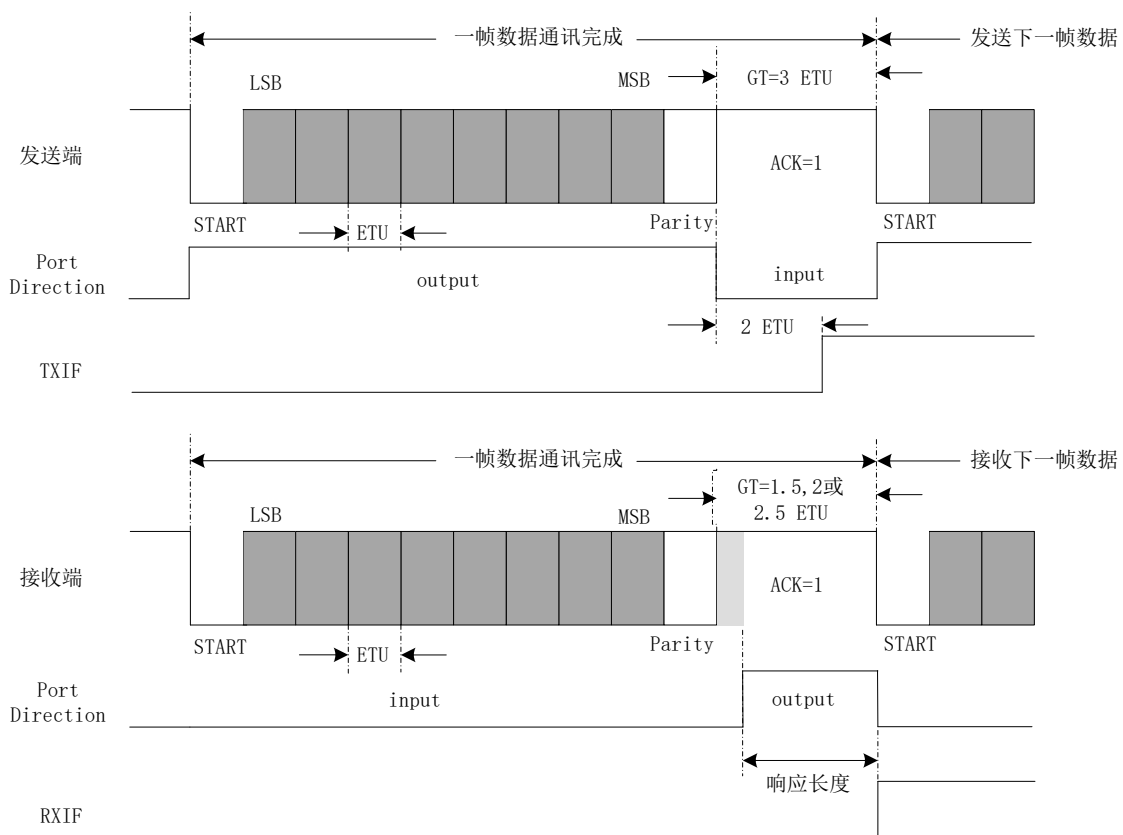
6. 此状态结束时, 回到 IDLE 状态, 产生接收中断。在中断中判断, 如果校验正确, 读取接收SBUF中的数据。(在接收的过程中, 如果程序有对SBUF3/4写入的动作, 该写入是无效的)

10.4.37816 通讯示意图

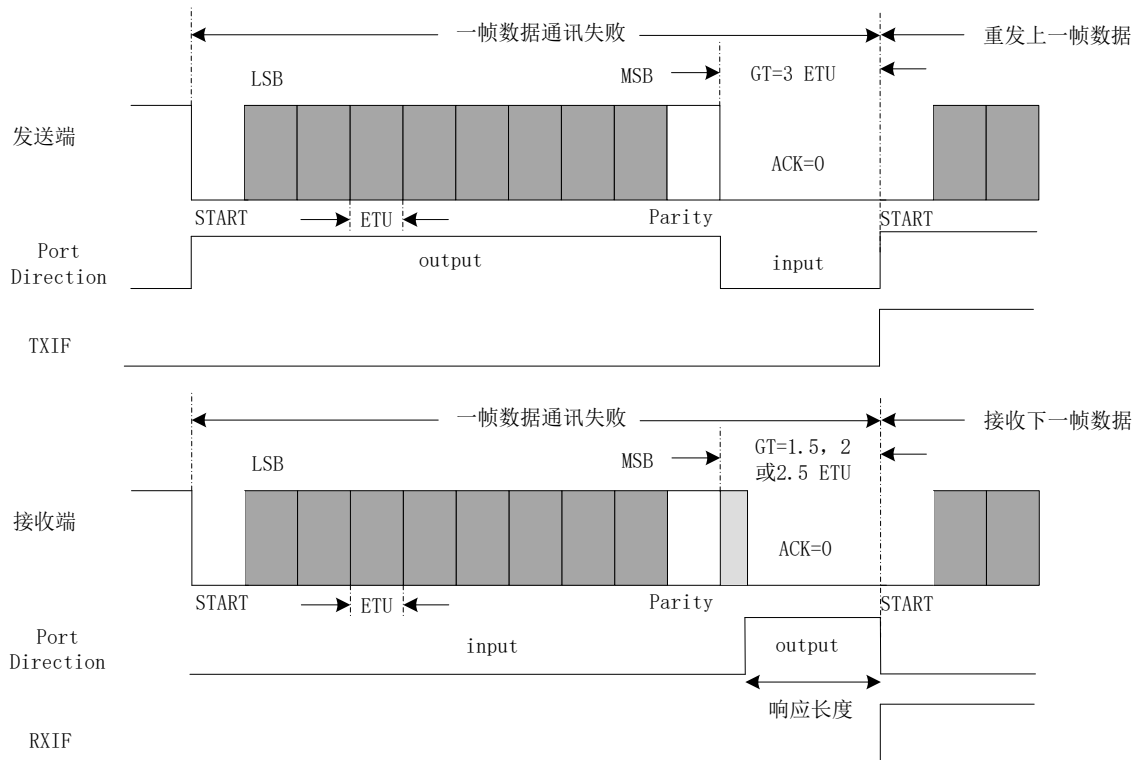


7816 通讯数据示意图

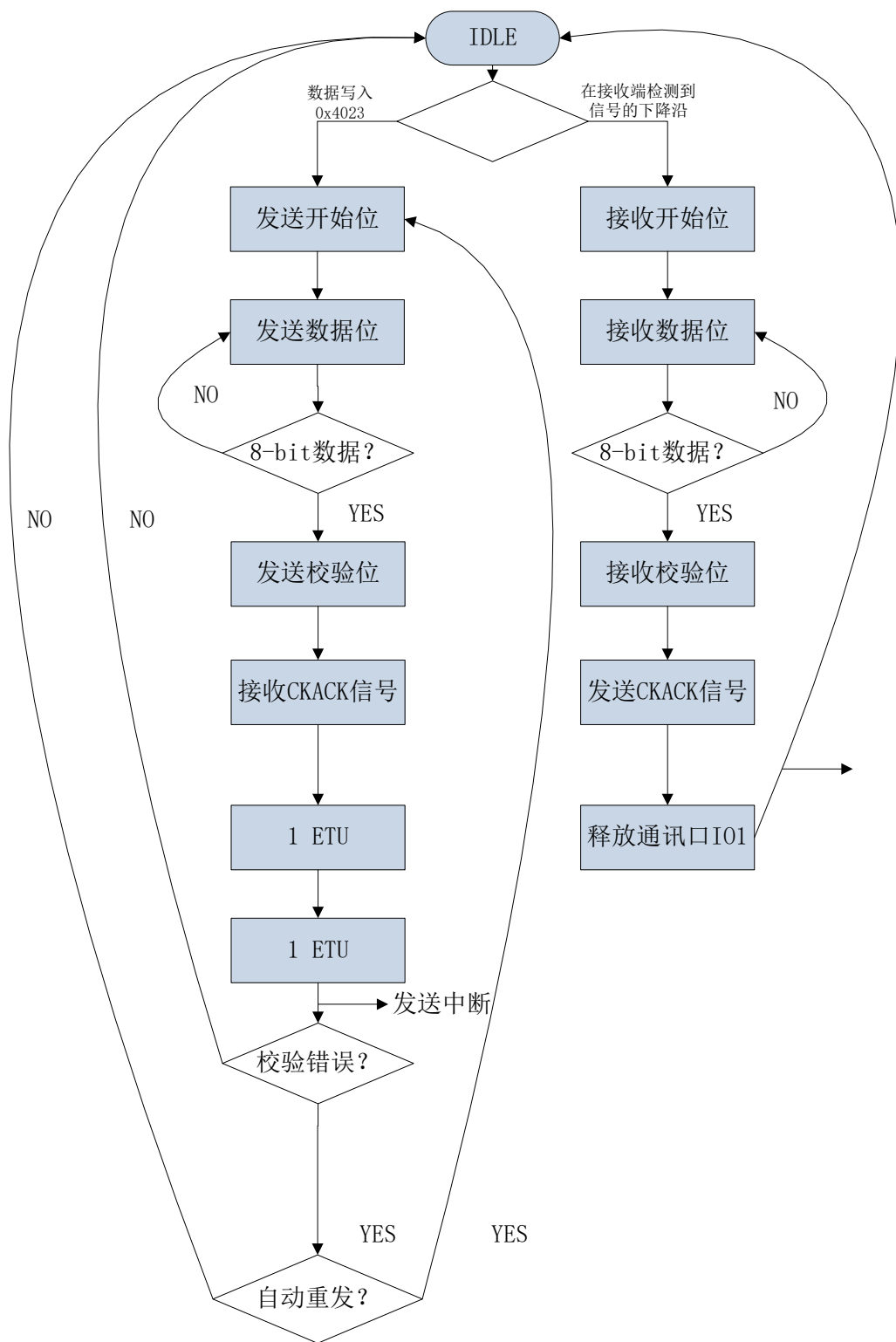
进行通信时，收发一个bit 需要的时间被定义为基础时间单位ETU（Elementary Time Unit）。发送端发送完一个数据帧后，接收端需要一定的时间对接收到的数据进行校验，发送端需要等待响应结果，然后再根据校验结果发送下一个数据帧或重发上一帧数据，即，连续发送两个数据帧之间的等待时间，该等待时间被定义为检测时间GT(Guarding Time)，一般， $1 GT = 3 ETU$ 。



收发数据成功时序



收发数据失败时序



ISO7816 功能状态图

## 10.5 特殊功能寄存器列表

UART模块寄存器基地址： 0x40005000(UART0端口); 0x40006000(UART1端口); 0x40007000(UART2端口); 0x40008000(UART3端口); 0x40009000(UART4端口); 0x40000000(UART5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	ISO7816 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	ISO7816 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：ISO7816 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)

MODESEL (串口功能选择寄存器)			基地址: 0x40005000—0x40009000; 0x40000000 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: ISO7816 功能

### UARTCON(UART 功能配置寄存器)

UARTCON (UART 功能配置寄存器)			基地址: 0x40005000—0x40009000; 0x40000000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	NEGFIX	UNEG	STOPSEL
Write:								L
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	UART 通讯中的逻辑锁定: 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	UART 通讯中的正逻辑或者是负逻辑 0: 正逻辑 (default)

	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

### ISO7816CON(ISO7816 功能配置寄存器)

ISO7816CON (ISO7816 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
<b>Read:</b>	X	X	X	X	X	X	7816PAR	AUTOR
<b>Write:</b>							ITY	XEN
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	AUTOT	REPTR1	REPTR0	ACKLE	ACKLE	PRDIE	RXIE	TXIE
<b>Write:</b>	XEN			N1	N0			
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
7816PARITY	奇偶校验选择位 1: 奇校验 0: 偶校验
AUTORXEN	自动重接收使能位 1: 使能 0: 禁止
AUTOTXEN	自动重发送使能位 1: 使能 0: 禁止
REPTR[1:0]	自动重收重发的次数上限 00: 0次 01: 1次 10: 2次 11: 3次
ACKLEN[1:0]	响应位的长度 00: 1bit 01: 1.5bit 10: 2bit 11: 2bit
PRDIE	ISO7816 上溢中断使能位 1: 使能 0: 禁止
RXIE	接收中断使能位 1: 使能 0: 禁止
TXIE	发送中断使能位 1: 使能 0: 禁止

### SREL(串口波特率发生寄存器)

SREL (串口波特率发生寄存器)		基地址: 0x40005000—0x40009000; 0x40000000 偏移地址: 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:								

<b>Reset:</b>	0	0	0	0	0	0	0	0
---------------	---	---	---	---	---	---	---	---

串口/ISO7816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2~65535之间的任一整数，最高波特率为1Mbps。波特率计算公式：

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

## SBUF(串口数据缓冲寄存器)

<b>SBUF</b> (串口数据缓冲寄存器)			<b>基地址: 0x40005000—0x40009000; 0x40000000</b> <b>偏移地址: 10H</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>								
<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>	SBUF[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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>UARTSTA</b> (UART 状态寄存器)			<b>基地址: 0x40005000—0x40009000; 0x40000000</b> <b>偏移地址: 14H</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>								
<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	PARITY	RXIF	TXIF
<b>Write:</b>								
<b>Reset:</b>	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	<b>接收中断标志</b> 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。

### ISO7816STA(ISO7816 状态寄存器)

<b>ISO7816STA</b> (ISO7816 状态寄存器)			<b>基地址: 0x40008000—0x40009000</b> <b>偏移地址: 18H</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>								
<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	RXERRS	TXERRS	PRDIF	RXIF	TXIF
<b>Write:</b>				TAT	TAT			
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
RXERRSTAT	<b>接收数据的状态</b> 1: 错误 0: 正确 写 0 清零，写 1 无效。
TXERRSTAT	<b>发送数据的状态</b> 1: 错误 0: 正确 写 0 清零，写 1 无效。
PRDIF	<b>ISO7816 接收上溢中断标志</b> 1: 产生上溢; 0: 未产生上溢; 写 0 清零，写 1 无效。
RXIF	<b>接收中断标志</b> 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。
--	--

## IRCON(红外调制控制寄存器)

<b>IRCON</b> (红外调制控制寄存器)			<b>基地址: 0x40005000—0x40009000; 0x40000000</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>	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	IR38kSource	IRLVL	IRTX
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
IR38kSource	<b>红外 38k 调制信号来源选择</b> 1: 红外 38k 调制信号来源于 HRC (经 HRCADJ 补偿之后未经分频的 HRC 时钟) 0: 红外 38k 调制信号来源于 PLL (未经分频的 PLL 时钟)
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

**IRDUTY(红外调制脉宽寄存器)**

<b>IRDUTY</b> (红外调制脉宽寄存器)			<b>基地址: 0x40005000—0x40009000; 0x40000000</b>					
			<b>偏移地址: 34H</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>								
<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]	调制波形占空比配置 00: 50% 01: 25% 10: 12.5% 11: 6.25%

# 11 LCD 模块

## 11.1 概述

HT5023 (100pin) 的 LCD 驱动单元最多可以支持 50 (Segment) \* 8 (Common) 输出。

HT5025 (80pin) 最多可以支持 39 (Segment) \* 8 (Common) 输出。

HT5027 (64pin) 最多可以支持 27 (Segment) \* 8 (Common) 输出。

主要特征如下:

- LCD 驱动模式: 内部电阻分压驱动
- 内部电阻分压驱动
  - 50 Seg \* 8 Com (100pin) | 39 Seg \* 8 Com (80pin) | 27 Seg \* 8 Com (64pin)
  - 52 Seg \* 6 Com (100pin) | 41 Seg \* 6 Com (80pin) | 29 Seg \* 6 Com (64pin)
  - 54 Seg \* 4 Com (100pin) | 43 Seg \* 4 Com (80pin) | 31 Seg \* 4 Com (64pin)
- 软件控制对比度可调 (电阻分压)
- LCD 充电方式可选: 大电流模式, 小电流模式和快速充电模式 (电阻分压)
- 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-SEG15	PB.0-PB.15	SEG0/PB.0 – SEG15/PB.15
SEG16-SEG17	PD.0-PD.1	SEG16/PD.0 – SEG17/PD.1
SEG18-SEG21	PD.2-PD.5	SCANOUT0/SEG18/PD.2-SCANOUT3/SEG21/PD.5
SEG22-SEG23	PD.6-PD.7	SEG22/PD.6-SEG23/PD.7
SEG28-SEG32	PA.0-PA.4	SEG28/PA.0-SEG32/PA.4
SEG33-SEG34	PA.7-PA.8	SEG33/INT2/PA.7-SEG34/INT3/PA.8
SEG35-SEG36	PA.12-PA.13	SEG35/ADCIN0/PA12-SEG36/ADCIN1/PA13
SEG37	PE.8	SEG37/ADCIN2/PE.8
SEG38-SEG42	PF.3-PF.7	SEG38/PF.3-SEG42/PF.7
SEG43-SEG46	PC.0-PC.3	SEG43/TX1/PC.0, SEG44/RX1/PC.1 SEG45/RX0/PC.2, SEG46/TX0/PC.3
SEG47	PE7	PE.7/LVDIN0/SEG47
SEG48	PF2	PF.2/SFOUT/SEG48

SEG49	PE5	PE.5/TX3/SEG49
SEG50	PA.11	PA.11/INT6/SEG50
SEG51	PC.6	PC.6/SPI_CLK/SEG51
SEG52	PC.7	PC.7/SPI_CS/SEG52
SEG53	PE.3	PE.3/CLKOUT/SEG53

80PIN 封装去除引脚: SEG23, SEG28-30, SEG38-42, SEG51-52

64PIN 封装去除引脚: SEG8-11, SEG16-17, SEG22, SEG32, SEG36-37, SEG43-44, SEG50

## 11.3 LCD 框图

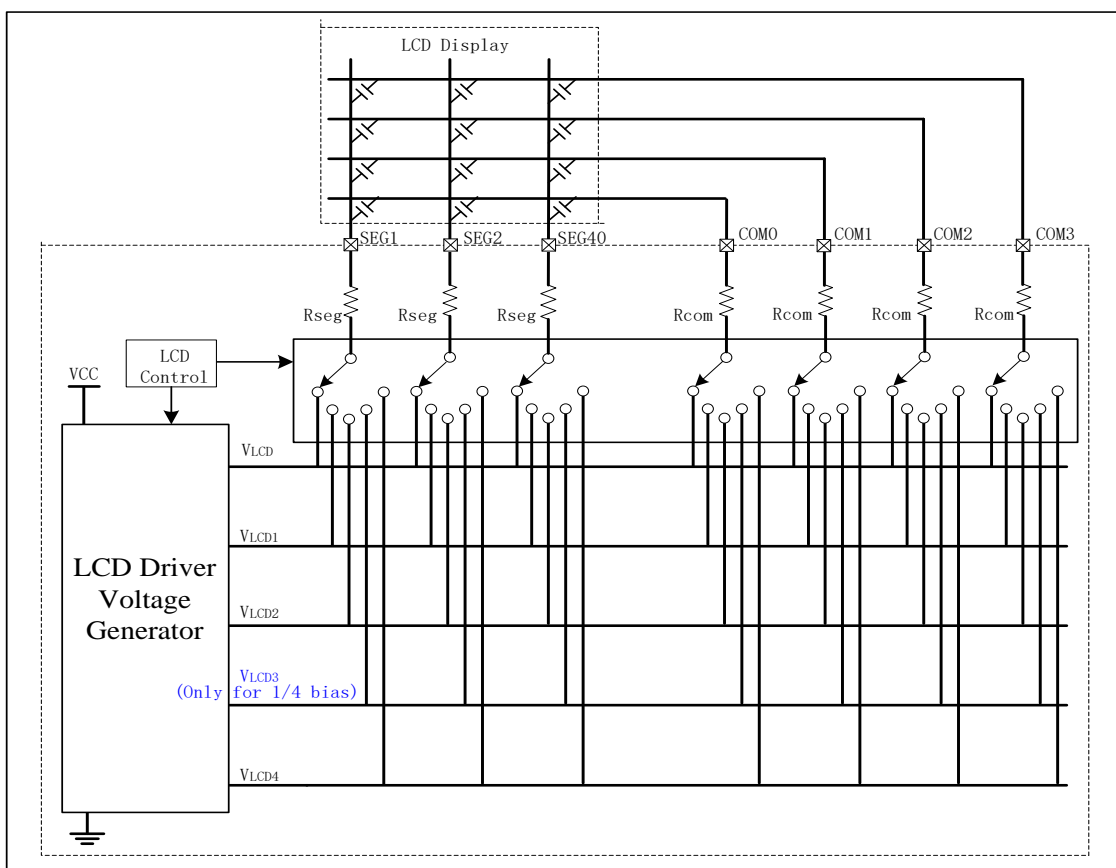
LCD 驱动单元采用 2 种工作方式: 1/3 Bias 和 1/4 Bias, 由 BIASCTL 位控制。LCD 的电源由 VLCD 提供, LCD 驱动电压 VLCD1、VLCD2、VLCD3、VLCD4 由内部的驱动电压发生模块生成, VLCD3 仅 1/4 Bias 有效。

DUTY[2:0]用于控制 LCD 波形 Duty, 参考第 11.5 输出波形章节。

LCD 单元的时钟源为 Fosc/Flrc, 由停振检测经过分频之后作为 LCD 波形扫描频率 Flcd, Flcd 可以通过寄存器 LCLK[2:0]进行配置。LCD 帧扫描频率 Ffrm=Flcd\*duty。如: 1/4 duty 时, 帧扫描频率 Ffrm=Flcd/4。详情参考特殊功能寄存器 LCDCLK 说明部分。

LCD 的显示数据放入 LCD BUF 中, 用于控制 LCD segment 的开关。当将某一段对应的 SEG 和 COM 控制位 (LCD\_BUF 对应位) 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, 内部驱动电压发生模块以及模拟电路被关闭, LCD 单元的时钟也被关闭。



LCD 模块功能框图

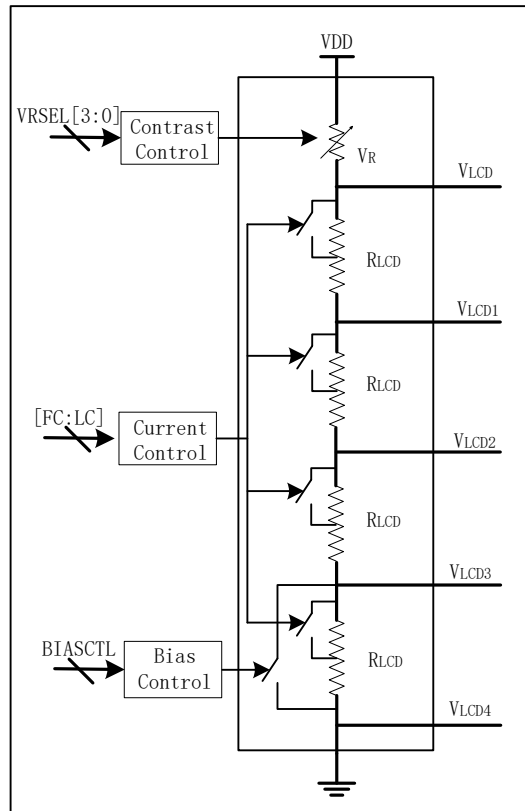
## 11.4 内部电阻分压驱动

LCD 驱动模式控制位可以选择内部电阻分压驱动模式。

此模式下，VLCD0、VLCD1、VLCD2、VLCD3 和 VLCD4 由内部电阻分压网络产生，最高档 VLCD0 $\leq$ VDD。

VRSEL[3:0]用于控制 LCD 对比度，参考特殊功能寄存器 LCDCON 相关 LCD 对比度部分 VRSEL 配置。

FC,LC 用于控制 LCD 充电模式。



LCD 内部电阻分压驱动

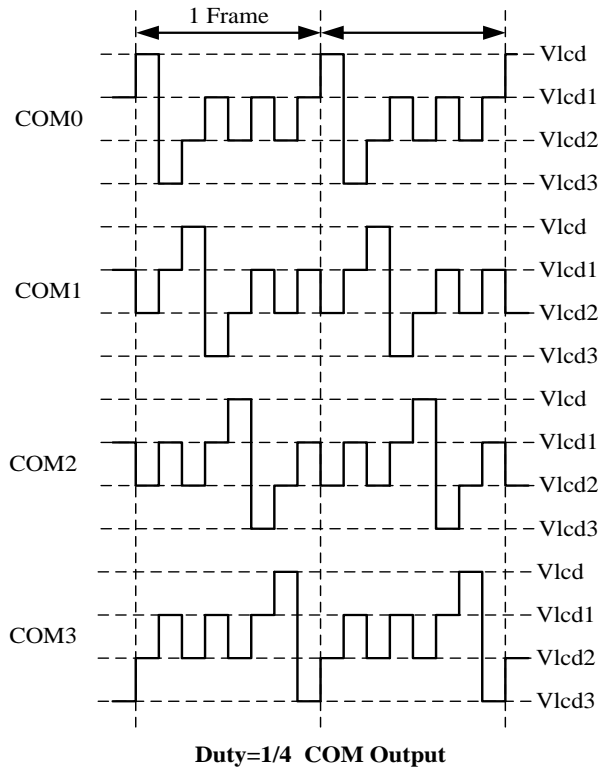
## 11.5 输出波形

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

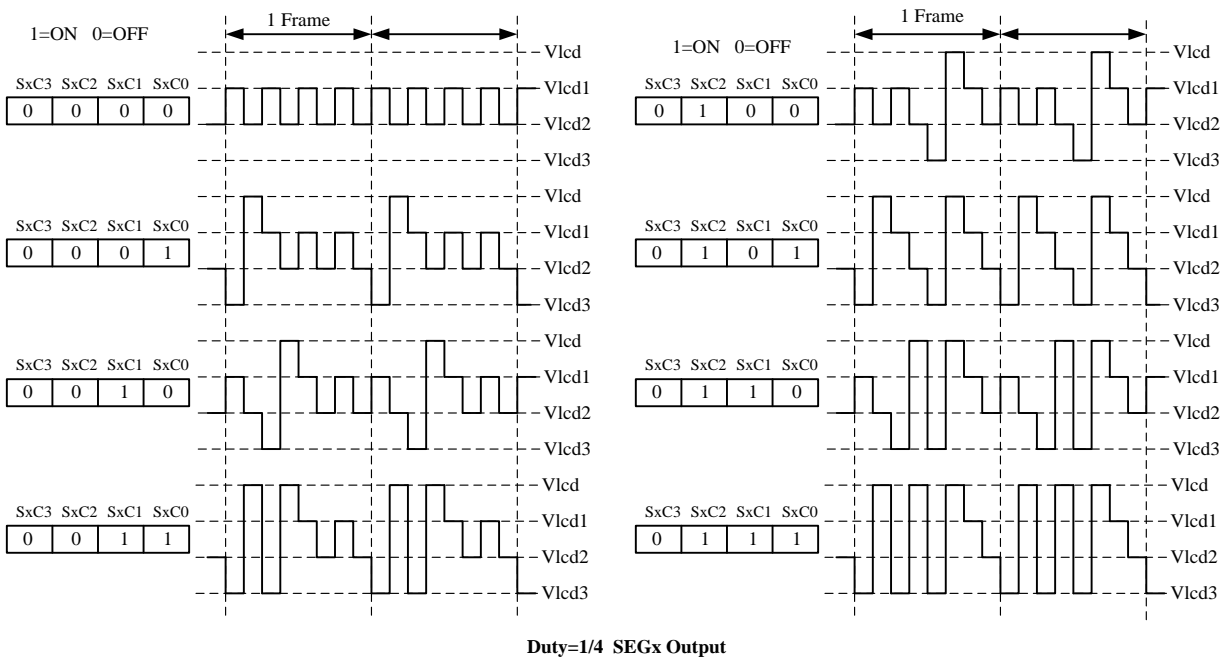
- DUTY[1:0]=00: 1/4 duty——COM0、COM1、COM2、COM3 被使用
- DUTY[1:0]=01: 1/6 duty——COM0 到 COM5 都被使用
- DUTY[1:0]=1x: 1/8 duty——COM0 到 COM7 都被使用

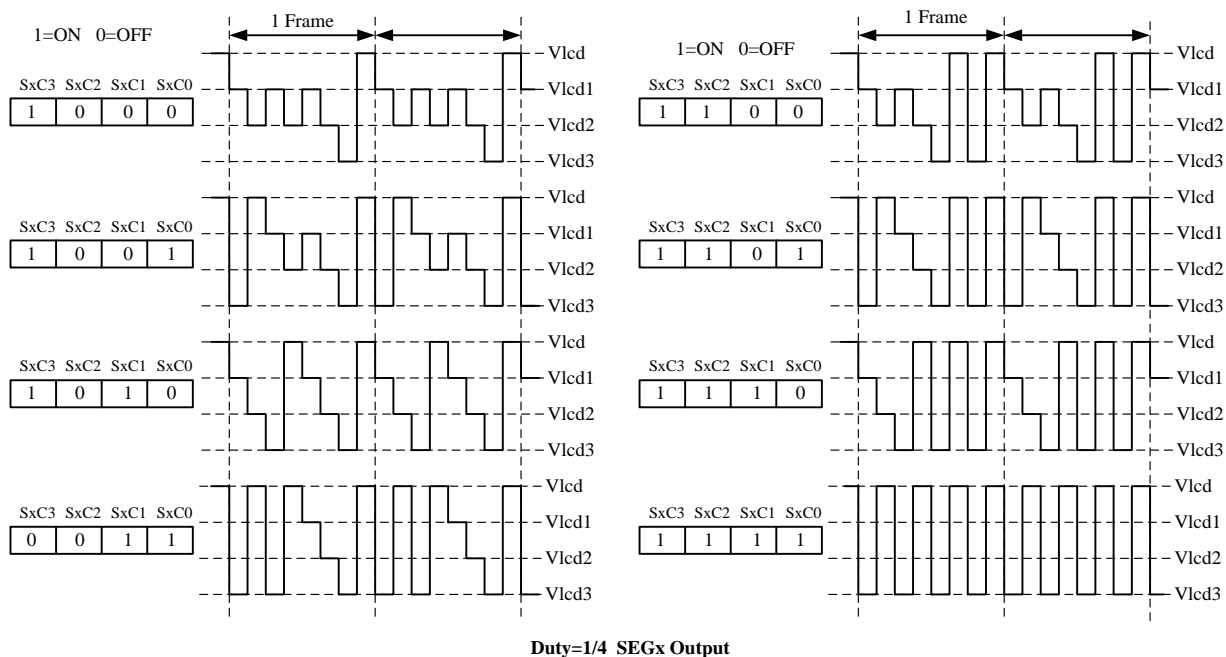
**COM 波形(1/3 Bias, 1/4 Duty):**

使用 COM0/1/2/3, 1 帧等于 4 个 LCD 波形时钟周期。

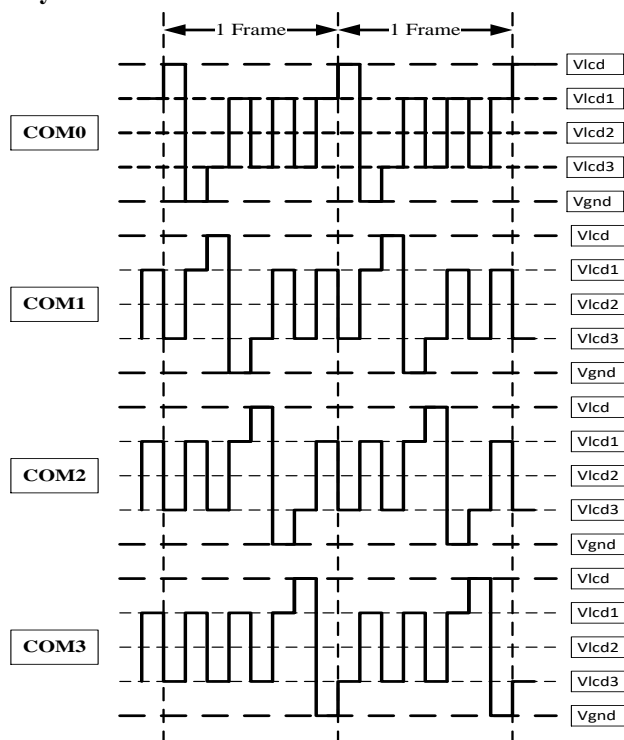


SEGMENT 输出波形(1/4 Duty):

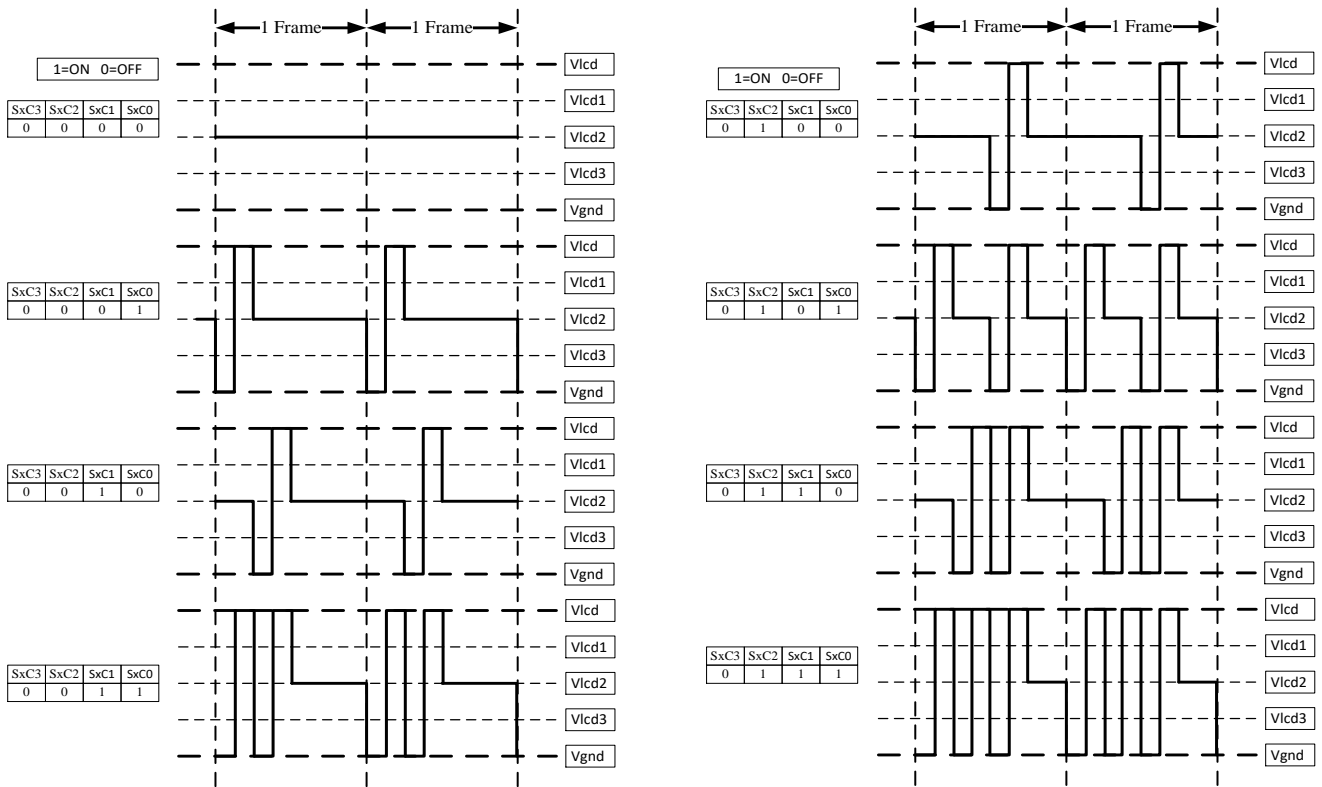




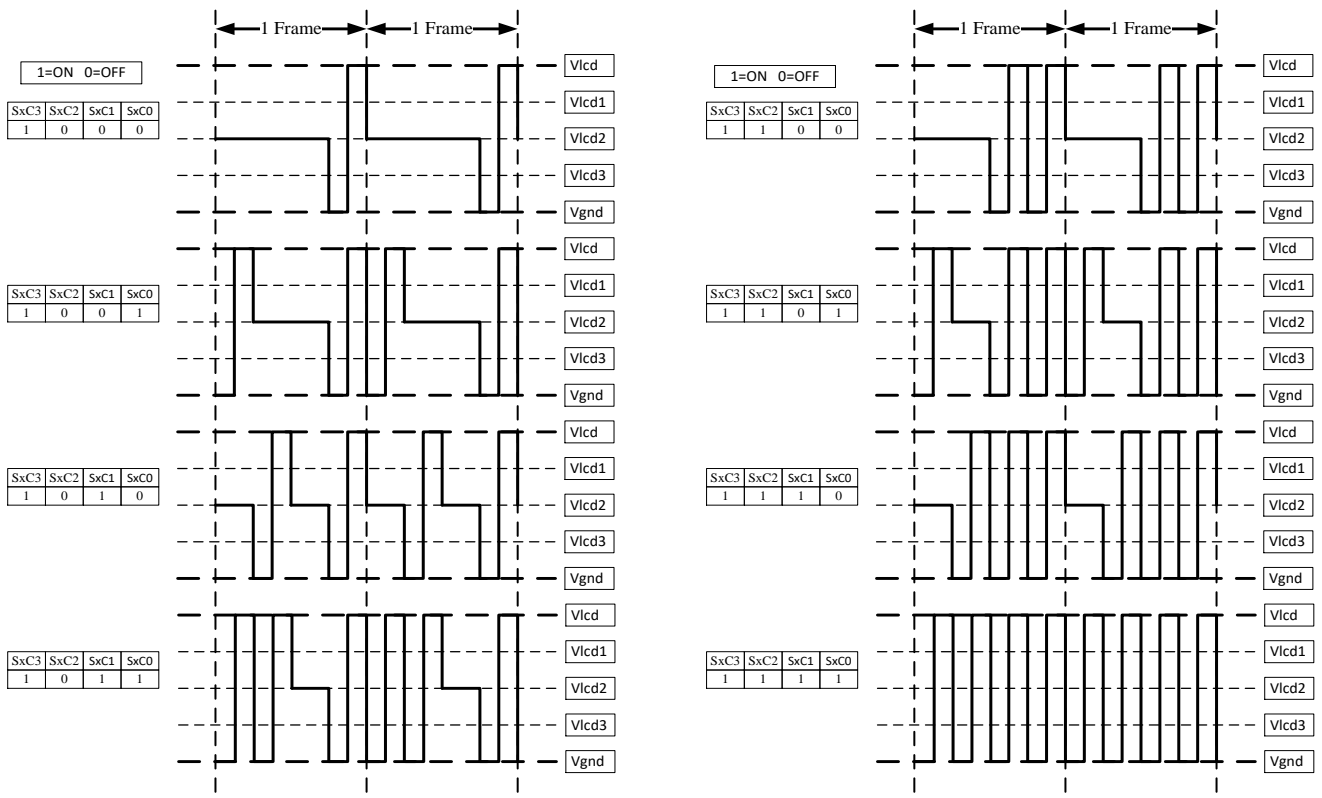
## COM 波形 (1/4Bias, 1/4Duty)



## SEGMENT 输出波形 (1/4Bias, 1/4Duty)



1/4Duty 1/4Bias SEGx Output



1/4Duty 1/4Bias SEGx Output

## 11.6 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。

54 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[41]对应 SEG41 (COM7—COM0)

地址 LCD\_BUF[42]对应 SEG42 (COM7—COM0)

地址 LCD\_BUF[43]对应 SEG43 (COM7—COM0)

地址 LCD\_BUF[44]对应 SEG44 (COM7—COM0)

地址 LCD\_BUF[45]对应 SEG45 (COM7—COM0)

地址 LCD\_BUF[46]对应 SEG46 (COM7—COM0)

.....

地址 LCD\_BUF[53]对应 SEG53 (COM7—COM0)

## 11.7 特殊功能寄存器列表

基地址: 0x4000D000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	LCDCCLK	R/W	0x0080	LCD 时钟频率选择寄存器
0x04	LCDCON	R/W	0x0089	LCD 驱动控制寄存器
0x0C	LCDOUT	R/W	0x0001	LCD 输出控制寄存器
0x10+i×4 (i=0~53)	LCD_BUF[i]	R/W	----	LCD 显示数据寄存器 此寄存器为 RAM 类型, 复位值为随机值

## 11.8 特殊功能寄存器说明

### LCDCCLK(LCD 时钟频率选择寄存器)

LCDCCLK (LCD 时钟频率选择寄存器)		基地址: 0x4000D000 偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	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>	BIASCT	X	X	DUTY[1:0]		LCLK[2:0]		
<b>Write:</b>	L							
<b>Reset:</b>	1	0	0	0	0	0	0	0

位	功能描述																
BIASCTL	<p><b>偏压驱动选择位</b></p> <p>1: 表示选择1/3 bias 偏压驱动 0: 表示选择1/4 bias 偏压驱动</p> <p>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。</p>																
DUTY[1:0]	<p><b>LCD的duty控制选择位</b></p> <table border="1"> <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(lf), F(lf)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 F(frm)=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。

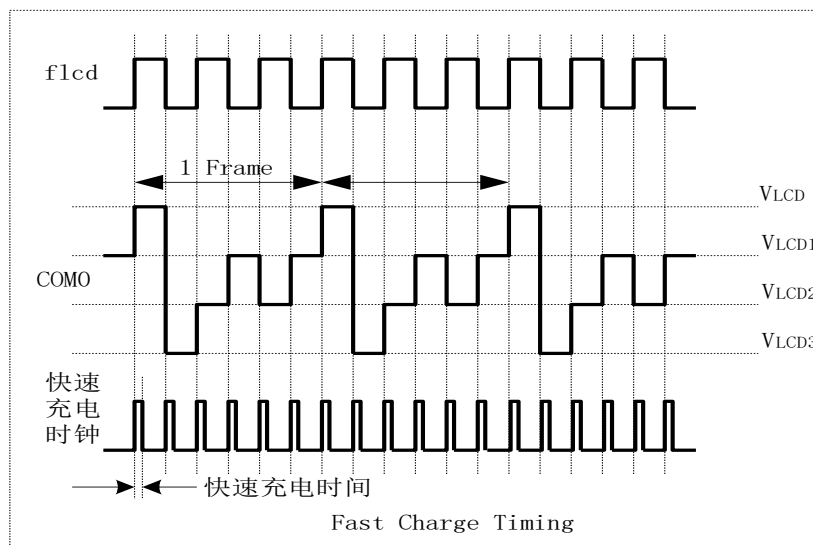
## LCDCON(LCD 驱动控制寄存器)

<b>LCDCON</b> (LCD 驱动控制寄存器)			<b>基地址: 0x4000D000</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>	X	X	X	X	X	X	X	FCSET2
<b>Write:</b>								

Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	FCSET[1:0]		FC	LC	VRSEL[3:0]			
Write:	FCSET[1:0]		FC	LC	VRSEL[3:0]			
Reset:	1	0	0	0	1	0	0	1

位	功能描述																				
FCSET2	<p>电阻分压结构控制位</p> <p>0: 保持原有的电阻分压结构 (default)</p> <p>1: 改善后电阻分压结构 (大电流模式的电流更大, 小电流模式的电流更小)</p> <p><b>改善后的结构, 选择快速充电模式, 显示效果更优, 功耗更小(详见 FAQ)</b></p>																				
FCSET[1:0]	<p>快速充电时间选择位</p> <table border="1"> <thead> <tr> <th>FCSET1</th> <th>FCSET0</th> <th>快速充电时间</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1/16个f lcd周期</td> </tr> <tr> <td>0</td> <td>1</td> <td>1/32个f lcd周期</td> </tr> <tr> <td>1</td> <td>0</td> <td>1/64个f lcd周期</td> </tr> <tr> <td>1</td> <td>1</td> <td>1/128个f lcd周期</td> </tr> </tbody> </table> <p>详见下图“快速充电时序图”</p>	FCSET1	FCSET0	快速充电时间	0	0	1/16个f lcd周期	0	1	1/32个f lcd周期	1	0	1/64个f lcd周期	1	1	1/128个f lcd周期					
FCSET1	FCSET0	快速充电时间																			
0	0	1/16个f lcd周期																			
0	1	1/32个f lcd周期																			
1	0	1/64个f lcd周期																			
1	1	1/128个f lcd周期																			
FC	<p>快速充电模式选择位</p> <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>23KΩ</b>。根据 LCD 差异, 可以选择 RLCD=<b>138KΩ</b>, 此时流过电阻网络的电流较小。</p> <p>在选择RLCD=<b>138KΩ</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>23KΩ</b>进行快速充电, 然后再切换到RLCD=<b>138KΩ</b>模式。快速充电时间可以通过FCSET[1:0]进行选择。</p>																				
LC	<p>慢速充电模式选择位</p> <p>和FC配合, 用于确定LCD充电模式</p> <table border="1"> <thead> <tr> <th>FC</th> <th>LC</th> <th>LCD充电模式</th> <th>RLCD默认结构阻值</th> <th>RLCD改善后结构阻值</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>0</td> <td>大电流充电模式</td> <td>RLCD=<b>23KΩ</b></td> <td>RLCD=<b>12KΩ</b></td> </tr> <tr> <td>0</td> <td>1</td> <td>小电流充电模式</td> <td>RLCD=<b>138KΩ</b></td> <td>RLCD=<b>700KΩ</b></td> </tr> <tr> <td>1</td> <td>1</td> <td colspan="3">短时间大电流, 快速充电模式</td> </tr> </tbody> </table>	FC	LC	LCD充电模式	RLCD默认结构阻值	RLCD改善后结构阻值	X	0	大电流充电模式	RLCD= <b>23KΩ</b>	RLCD= <b>12KΩ</b>	0	1	小电流充电模式	RLCD= <b>138KΩ</b>	RLCD= <b>700KΩ</b>	1	1	短时间大电流, 快速充电模式		
FC	LC	LCD充电模式	RLCD默认结构阻值	RLCD改善后结构阻值																	
X	0	大电流充电模式	RLCD= <b>23KΩ</b>	RLCD= <b>12KΩ</b>																	
0	1	小电流充电模式	RLCD= <b>138KΩ</b>	RLCD= <b>700KΩ</b>																	
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				1/3 bias 对比度	1/4 bias 对比度
		VRSEL3	VRSEL2	VRSEL1	VRSEL0	(% of VCC)	(% of VCC)
0	0	0	0	0	97.4	99.1	
0	0	0	1	0	92.4	94.0	
0	0	1	0	0	85.8	88.8	
0	0	1	1	0	80.0	84.2	
0	1	0	0	0	75.0	80.0	
0	1	0	1	0	70.6	76.2	
0	1	1	0	0	66.6	72.8	
0	1	1	1	0	63.2	69.6	
1	0	0	0	0	60.0	66.6	
1	0	0	1	0	57.2	64.0	
1	0	1	0	0	54.6	60.0	
1	0	1	1	0	52.2	59.2	
1	1	0	0	0	50.0	57.2	
1	1	0	1	0	48.0	55.2	
1	1	1	0	0	46.2	53.4	
1	1	1	1	0	44.4	51.6	



快速充电时序图

## LCDOUT(LCD 输出控制寄存器)

LCDOUT (LCD 输出控制寄存器)	基地址: 0x4000D000 偏移地址: 0CH
-------------------------	------------------------------

	<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	X	X	X	<b>OUTEN</b>
<b>Write:</b>	X	X	X	X	X	X	X	<b>OUTEN</b>
<b>Reset:</b>	0	0	0	0	0	0	0	1

位	功能描述
<b>OUTEN</b>	<b>LCD 输出使能</b> 0: LCD 输出关闭 1: LCD 输出打开 (default)

## 12 WDT 模块

### 12.1 概述

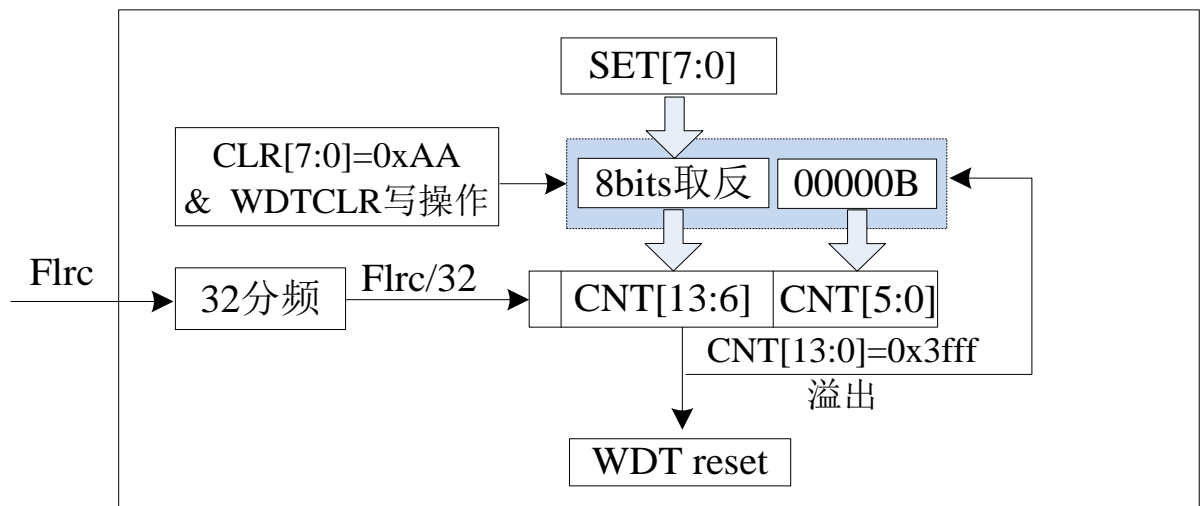
看门狗只有复位功能，没有中断功能，看门狗的开关会受到两个信号控制：

- (1)当芯片处于（TEST=0/JTAGWDTEN=0）（调试模式）时，看门狗一定被关闭；
- (2)当芯片处于 TEST=1 时，看门狗是否能正常运行受到低频 RC 的开关控制，只要低频 RC 正常运行，看门狗在非调试模式下永不关断。

Watchdog Timer 是一个特殊的定时器，其时钟为内部低频 RC 时钟 Flrc，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执行喂狗操作，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

功能框图如下：



### 12.2 工作模式

当正常模式时WDT均被使能；当TEST=0且JTAGWDTEN=0时（调试模式），关闭WDT计数，并且将WDTCNT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 时钟导致 WDT 被关闭时，需要同时对 WDTCLR 进行喂狗操作，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	WDTCNT	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)

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, 重新加载 WDT 内部计数器 WDTCNT 初值, 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> $WDT \text{ 溢出时间} = 64ms * (1 + SET[7:0])$ 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

### WDTCNT(WDT 计数寄存器)

WDTCNT (WDT 计数寄存器)		基地址: 0x40010000 偏移地址: 08H						
	Bit15	14	13	12	11	10	9	Bit8

<b>Read:</b>	X	X	CNT[13:0]					
<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>	CNT[7:0]							
<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
CNT[15:0]	<p><b>WDT 计数寄存器:</b></p> <p>指示当前 WDT 内部的计数值</p> <p>执行喂狗操作 WDTCLR = 0xAAxx，系统对 SET[7:0]8bits 取反,载入 CNT[13:6], CNT[5:0]被清零，向上重新开始计数，当 CNT[13:0]计到 0x3FFF，发生 WDT 复位</p>

## 13 定时器模块

### 13.1 定时器单元概述

HT502X 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 ( $F_{hrc}$ )，低频晶体时钟 ( $F_{osc}$ ) 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
2. PWM 功能
3. 捕获功能
4. 事件计数功能

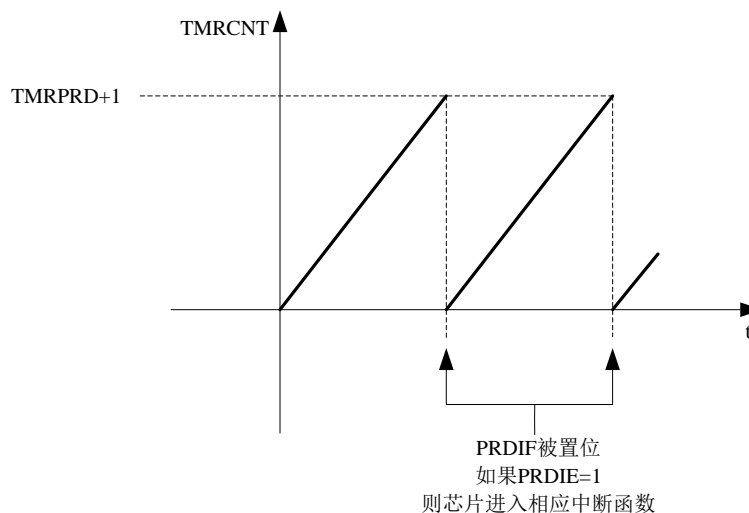
### 13.2 周期定时功能

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： $TMRCNT$ ， $TMRPRD$ 。

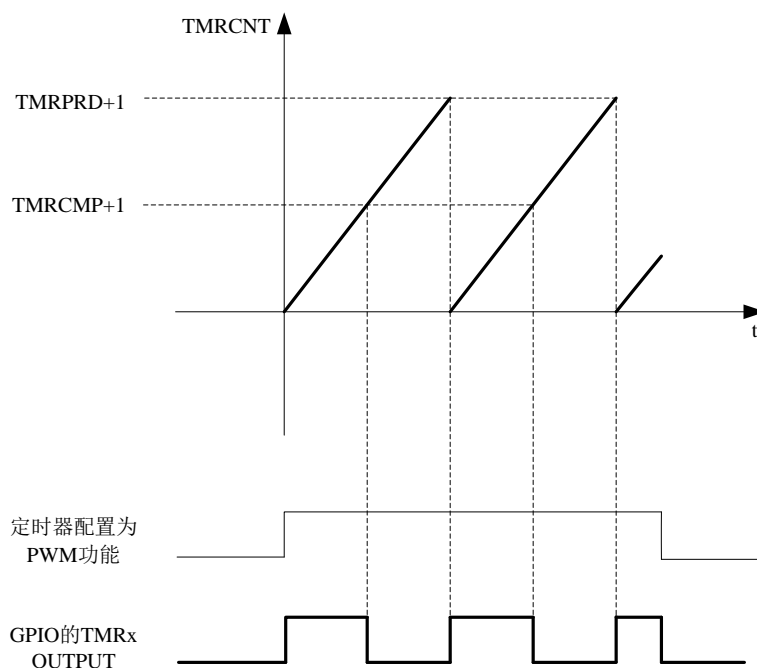


## 13.3 PWM 功能

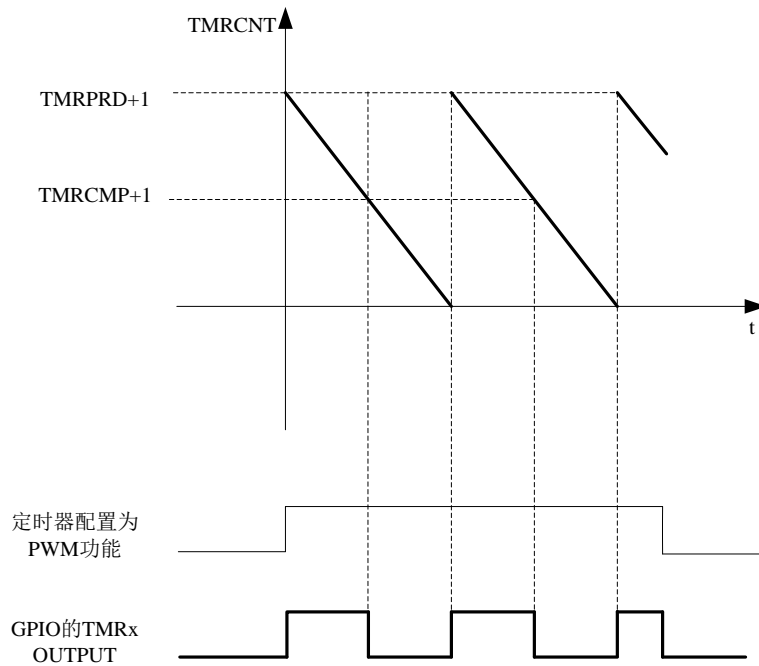
PWM 功能可通过寄存器 TMRCON.MODE[1:0]配置，同时需将对应的 GPIO 配置为 TMR 功能，配置成功后，相应的 TMR 引脚会输出 PWM 波形。PWM 的周期和占空比可通过寄存器 TMRPRD、TMRCMP 进行配置。

PWM 计数方式可以分为向上计数，向下计数以及中央对齐方式。

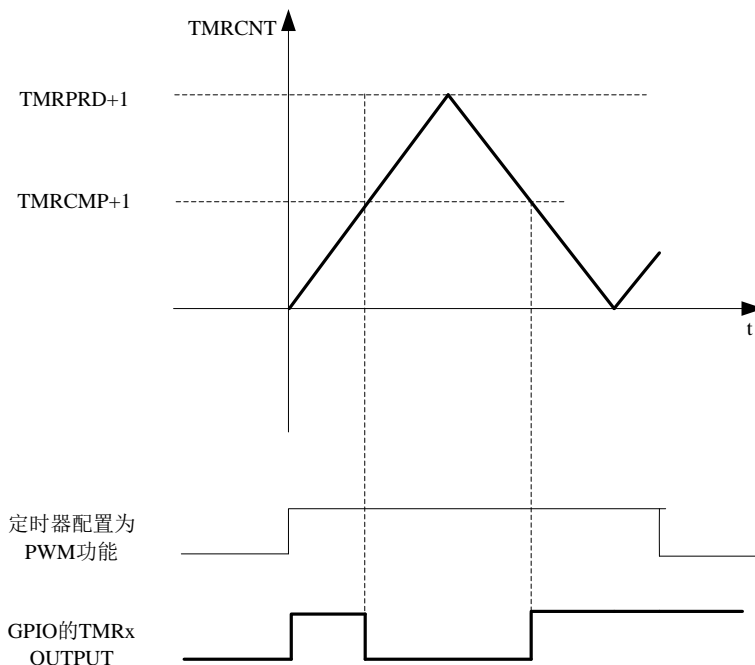
下图为向上计数方式。当使能计数器(CNTEN=1)之后，计数器开始从 0 计数，当计数寄存器(TMRCNT)的值等于设定的比较寄存器(TMRCMP+1)时，PWM 输出管脚发生电平翻转，同时置位比较中断标志 (CMPIF=1)。计数器继续向上计数，当计数寄存器(TMRCNT)的值等于设定的周期寄存器(TMRPRD+1)时，PWM 输出管脚再次发生电平翻转，同时置位周期定时中断标志 (PRDIF=1)。PWM 输出波形如下图所示：



下图为向下计数方式。当使能计数器 (CNTEN=1) 之后，计数器开始从 (TMRPRD+1) 向下计数，当计数寄存器 (TMRCNT) 的值等于设定的比较寄存器 (TMRCMP+1) 时，PWM 输出管脚发生电平翻转，同时置位比较中断标志 (CMPIF=1)。计数器继续向下计数，当计数寄存器 (TMRCNT) 的值等于 0 时，PWM 输出管脚再次发生电平翻转，同时置位周期定时中断标志 (PRDIF=1)。PWM 输出波形如下图所示：



下图为中央计数方式时，当计数寄存器（TMRcnt）从 0 开始向上计数，其值等于比较寄存器（TMRCMP+1）时，PWM 输出管脚发生电平翻转。计数器继续向上计数，当计数寄存器（TMRcnt）的值等于等于设定的周期寄存器（TMRPRD+1）时，置位周期定时中断标志（PRDIF=1），但 PWM 输出管脚不发生电平翻转，计数器从周期寄存器(TMRPRD+1)的值开始向下继续计数，当计数寄存器(TMRcnt)的值再次等于设定的比较寄存器（TMRCMP+1）时，PWM 输出管脚发生电平翻转。PWM 输出波形如下图所示：



假如系统时钟选择为 PLL 产生的 19.660800MHz 时钟，Timer 预分频寄存器（TMRDIV）默认值为 0（不分频），定时器 0 选择 PWM 模式，向上计数方式，初始电平为高电平。若要想 PWM 占空比（高电平时间与周期之比）为 30%，则  $(TMRCMP+1)/(TMRPRD+1)=30\%$ ，根据需要的周期值来确定 TMRPRD 寄存器的值。TMRPRD 能配置的最大周期定时时间为：

$(TMRPRD+1)/(19.660800M/(TMRDIV +1))=(0xFFFF+1)/(19.660800M/(0xFFFF+1))\approx 218.46ms$

TMRCMP, TMRPRD 如被修改, 均是在完成本次计数之后下一次生效。

功能主要相关寄存器: TMRCNT, TMRCMP, TMRPRD。

## 13.4 捕获功能

在输入捕获模式下, 假如设定上升沿检测, 当 TMR0~TMR3 管脚检测到上升沿, 计数寄存器(TMRCNT)的当前值被锁定到捕获数据寄存器(TMRCAP)中。当捕获事件发生时, 置位捕获中断标志(CAPIF=1), 如果使能捕获中断(CAPIE=1), 将产生捕获中断, 进入相应的捕获中断服务程序。

捕获过程中, 如果没有检测到沿, 当计数寄存器(TMRCNT)的值和设定的周期寄存器(TMRPRD+1)相等时会置位周期定时中断标志(PRDIF=1), 同时周期寄存器(TMRCNT)从0开始计数。如使能了周期定时中断(PRDIE=1), 会进入相应的周期中断服务程序。

捕获检测可以选择上升沿, 下降沿方式。

功能主要相关寄存器: TMRCAP, TMRCNT, TMRPRD。

## 13.5 事件计数功能

在输入事件计数模式下, 假如设定下降沿检测, 当 Timer0~Timer3 管脚上检测到一次下降沿, 计数寄存器(TMRCNT)的当前值加1。当计数寄存器(TMRCNT)的值等于设定的比较寄存器(TMRCMP+1)时, 会置位事件计数中断标志(ACIF=1), 同时计数寄存器(TMRCNT)自动清零, 重新开始计数。如果使能了事件计数中断(ACIE=1), 将产生事件计数中断, 进入相应的事件计数中断服务程序。

事件计数过程中, 在 TMRCMP>TMRPRD 情况下, 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), 计数寄存器(TMRCNT)继续计数直到等于设定的比较寄存器(TMRCMP+1), 如果使能了周期定时中断(PRDIE=1)会进入周期中断服务程序。

单次最大计数个数为 0xFFFF, 可以配合周期中断实现任意次数的组合。事件计数检测可以选择上升沿, 下降沿方式, 该功能可用于 MCU 发脉冲。

比较寄存器(TMRCMP)如被修改, 如果修改后的值小于当前计数寄存器(TMRCNT)的值, 则立刻触发事件计数中断, 同时计数寄存器(TMRCNT)清零, 重新开始计数; 如果修改后的值大于当前计数寄存器(TMRCNT)的值, 则继续本次计数。

功能主要相关寄存器: TMRCNT, TMRCMP, TMRPRD。

## 13.6 中断功能

### 13.6.1 定时周期中断

当计数寄存器（TMRCNT）的值等于设定的周期寄存器（TMRPRD+1）时，如果使能周期定时中断（PRDIE=1），则发生周期定时中断。此中断在任何功能模式下都可以产生。

### 13.6.2 捕获中断

当检测到外部输入信号相应沿时，如使能了捕获中断（CAPIE=1），则发生捕获中断。计数寄存器（TMRCNT）的值被锁定到捕获数据寄存器（TMRCAP）中。

### 13.6.3 比较中断

当计数寄存器（TMRCNT）的值等于设定的比较寄存器（TMRCMP+1）时，如使能了 PWM 比较中断（CMPIE=1），则发生比较中断。

### 13.6.4 事件计数功能

当检测到设定次数的外部输入信号相应沿时，如使能了事件计数中断（ACIE=1），则发生事件计数中断。

在输入事件计数模式下，假如设定下降沿检测，当 Timer0~Timer3 管脚上检测到一次下降沿，计数器的当前值加一。当计数器的值和设定的比较寄存器 TMRCMP 值相等时，相应的事件计数中断标志 TMRIF.3 被置 1，如果使能中断（TMRIE.3=1），将产生中断。产生事件计数中断标志后计数器 TMRCNT 自动清零，重新开始计数。

事件计数过程中，**在  $TMRCMP > TMRPRD$  情况下，当 TMRCNT 计数值增加到和 TMRPRD 相等时会产生周期中断标志**，如使能了周期中断会进入中断服务程序，之后 TMRCNT 继续计数直到等于 TMRCMP 值。单次最大计数个数为 0xFFFF，可以配合周期中断实现任意次数的组合。

事件计数检测可以选择上升沿，下降沿方式，该功能可用于 MCU 发脉冲。

功能主要相关寄存器：TMRCNT，TMRCMP。

TMRCMP 如被修改，如果修改后的值小于当前 TMRCNT 的值，则立刻触发事件计数中断，之后 TMRCNT 计数器清零，重新开始计数；如果修改后的值大于当前 TMRCNT 的值则继续本次计数。

注：Timer4/5 不具有此功能

## 13.7 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
0x0C	TMRCAP	R/*W	0x0000	捕获数据寄存器
0x10	TMRCNT	R/*W	0x0000	计数器寄存器
0x14	TMRCMP	R/W	0x0000	比较器寄存器
0x18	TMRIE	R/W	0x0000	中断使能寄存器
0x1C	TMRIF	R/W	0x0000	中断标志寄存器

## 13.8 特殊功能寄存器说明

### TMRCON(定时器控制寄存器)

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:	RESERV	PWMHL	PWMC[1:0]		CCMOD	MODE[1:0]		CNTE
Write:	ED				E			N
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RESERVED	内部保留位, 应用中固定为 0
PWMHL	PWM 初始电平选择: 0: 高电平 1: 低电平
PWMC[1:0]	PWM 工作模式选择: (PWM 计数方式) 00: 向上计数

	01: 向下计数 1X: 中央对齐
CCMODE	<b>捕获电平沿选择:</b> 0: 上升沿 1: 下降沿
MODE[1:0]	<b>Timer 功能选择:</b> 00: 事件计数 01: PWM 功能 10: 捕获功能 11: 周期定时功能 需将 GPIO 配置为 TMRx 功能
CNTEN	<b>计数器使能位:</b> 0: 关闭 1: 使能

### TMRDIV(预分频寄存器)

<b>TMRDIV</b> (预分频寄存器)			<b>基地址: 0x40001000--0x40004000</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>	TMRDIV[7:0]							
<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>	TMRDIV[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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)

### TMRPRD(周期寄存器)

<b>TMRPRD</b> (周期寄存器)			<b>基地址: 0x40001000--0x40004000</b>					
			<b>偏移地址: 08H</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>	TMRPRD[15:8]							
<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>	TMRPRD[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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。

### TMRCAP(捕获数据寄存器)

<b>TMRCAP</b> (捕获数据寄存器)			<b>基地址: 0x40001000--0x40004000</b> <b>偏移地址: 0CH</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>	TMRCAP[15:8]							
<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>	TMRCAP[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
TMRCAP[15:0]	当发生捕获事件时，当前计数器的值被存到该寄存器里

### TMRCNT(计数寄存器)

<b>TMRCNT</b> (计数寄存器)			<b>基地址: 0x40001000--0x40004000</b> <b>偏移地址: 10H</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>	TMRCNT[15:8]							
<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>	TMRCNT[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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

## TMRCMP(比较寄存器)

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

位	功能描述
TMRCMP[15:0]	比较寄存器有两个功能: (1)Timer 做 PWM 功能的时候,当计数器达到(TMRCMP+1)的设定值时,PWM 输出翻转,同时置位比较中断标志(CMPIF=1),如使能了 PWM 比较中断(CMPIE=1),则发生比较中断。 (2)Timer 做事件计数功能的时候,当计数寄存器(TMRCNT)的值等于设定的比较寄存器(TMRCMP+1)时,会置位事件计数标志(ACIF=1),同时计数寄存器(TMRCNT)会从 0 开始重新计数,如果使能了事件计数中断(ACIE=1),则芯片会产生事件计数中断。

## TMRIE(定时器中断使能寄存器)

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	ACIE	CMPIE	CAPIE	PRDIE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ACIE	事件计数中断使能 0: 关闭, 1: 使能
CMPIE	比较中断使能 0: 关闭, 1: 使能
CAPIE	捕获中断使能 0: 关闭, 1: 使能
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)

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	ACIF	CMPIF	CAPIF	PRDIF
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ACIF	事件计数中断标志 0: 未产生中断 1: 产生中断 (写 0 清 0)
CMPIF	比较中断标志 0: 未产生中断 1: 产生中断 (写 0 清 0)
CAPIF	捕获中断标志 0: 未产生中断 1: 产生中断 (写 0 清 0)
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 SPI 模块

### 14.1 概述

SPI 模块可以实现在 MCU 和外围设备（包含外部 MCU）之间的全双工同步串行通讯。这里提到的 MCU 或者外围设备必须包含 SPI 模块。包含下列特征：

- 全双工模式
- 三线同步传输
- 主机和从机模式
- 7 种主机波特率
- 从机时钟最高至  $f_{cpu}/4$
- 极性和相位可编程的串行时钟
- 写冲突处理机制
- 8 位数据传输，高字节在前，低字节在后
- 8 位从机选择接口，控制外部从机
- 与主机 CPU 的专用功能寄存器接口
- 无二义端口，标准的 SPI

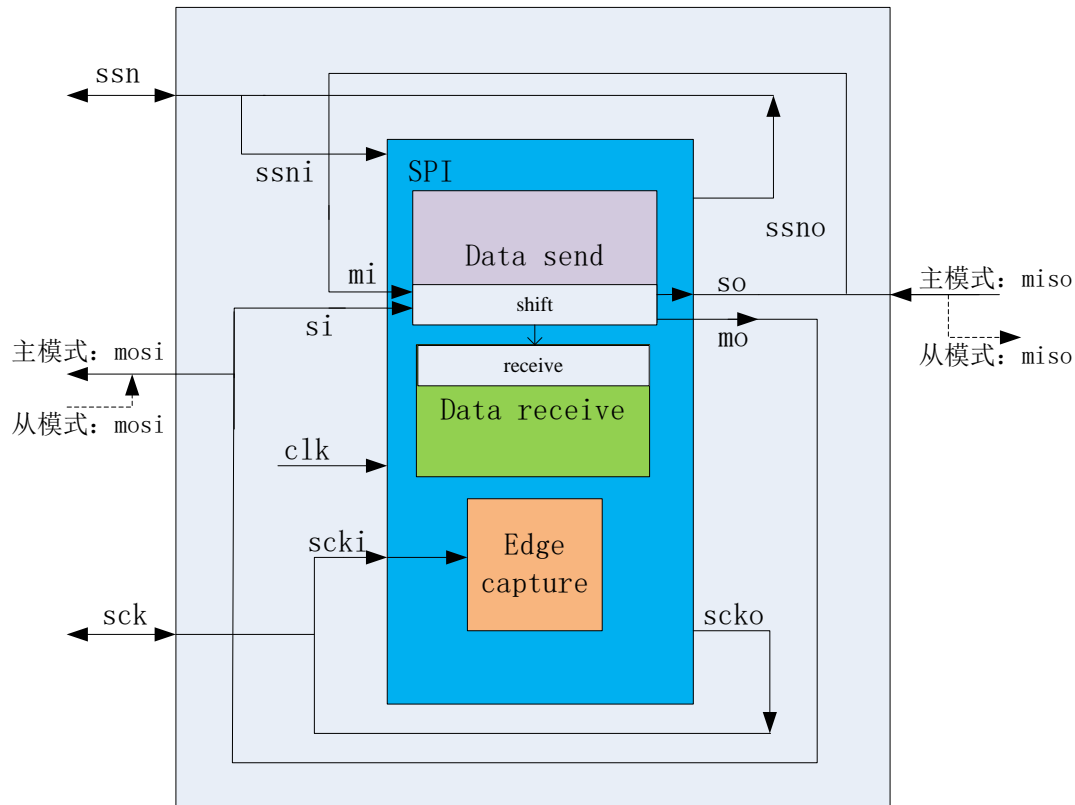
### 14.2 详细功能说明

串行外设接口(SPI)允许芯片与其他设备以半/全双工、同步、串行方式通信。此接口可以被配置成主模式，并为从设备提供通信时钟(SCK)。

#### 14.2.1 SPI 主要特征

- 3 线全双工同步传输
- 主模式或从模式操作
- 7 个主模式频率( $f_{cpu}$  的 2/4/8/16/32/64/128 分频)
- 在输入引脚 SPI\_CS 上的电平和下降沿侦测
- 可编程的时钟极性和相位
- 可触发中断的专用发送和接收标志

### 14.2.2 SPI 接口模块框图



### 14.2.3 SPI 接口传输格式

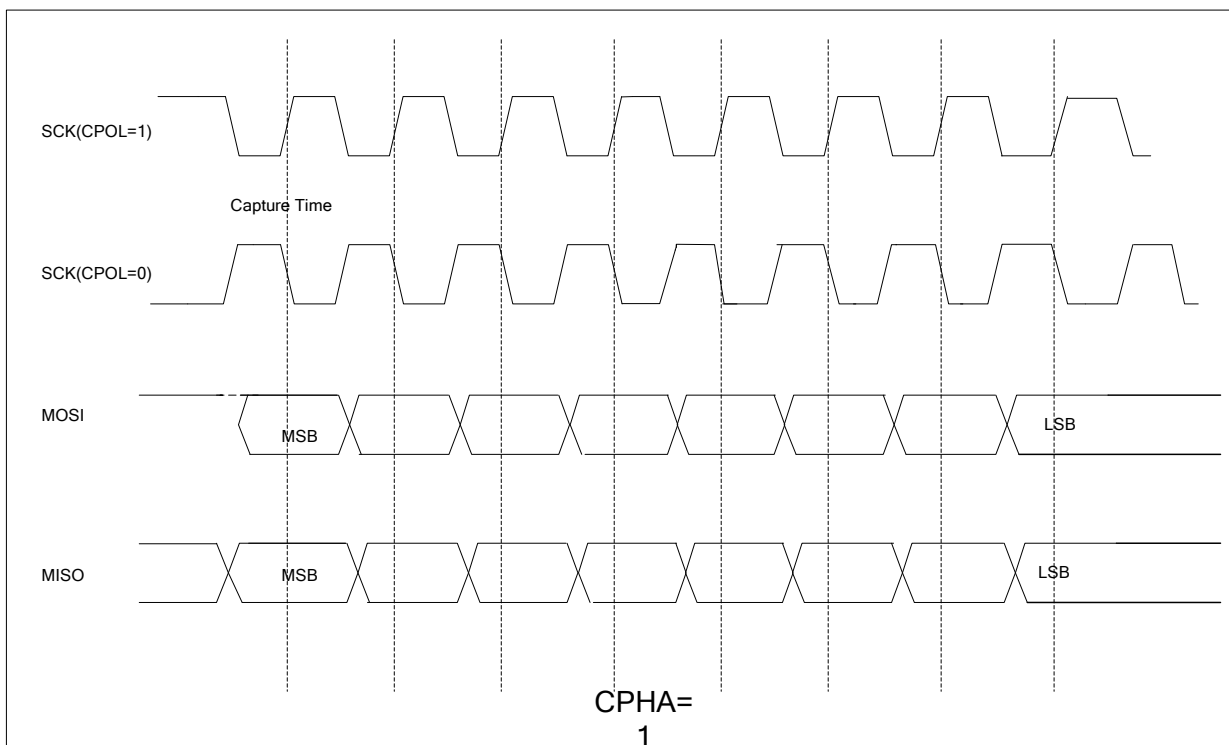
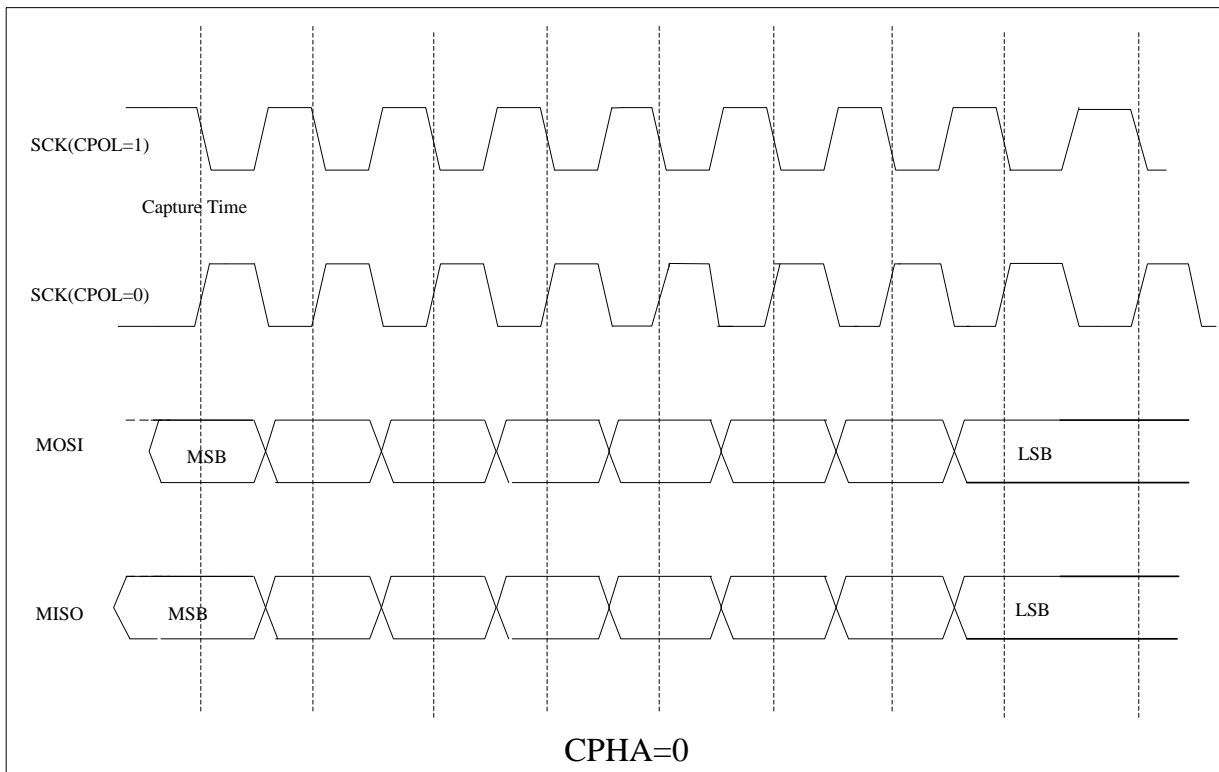
下图显示了数据传输的主要格式。根据 SPI 模块的设置，数据的每一位在主时钟（SCK）的上升沿（CPOL=0）或下降沿（CPOL=1）被传送。数据在主时钟（SCK）的下降沿（CPOL=0）或上升沿（CPOL=1）被接收。这适用于主模式或从模式的传输器/接收器，前提是 SCK 是传输过程中的主时钟。如果 CPHA 被置位，第一位（MSB）将在 SCK 的第一个动态沿时通过 MOSI/MISO 被发送。如果 CPHA 被清零，第一位（MSB）将在 SCK 的第一个动态沿之前半个周期被发送。

除此之外，输入数据在每一位传输一半时被取样，在这个时钟周期的相反的电平上，数据被移位到输出信号 MOSI 上。

### 14.2.4 主机模式传输格式

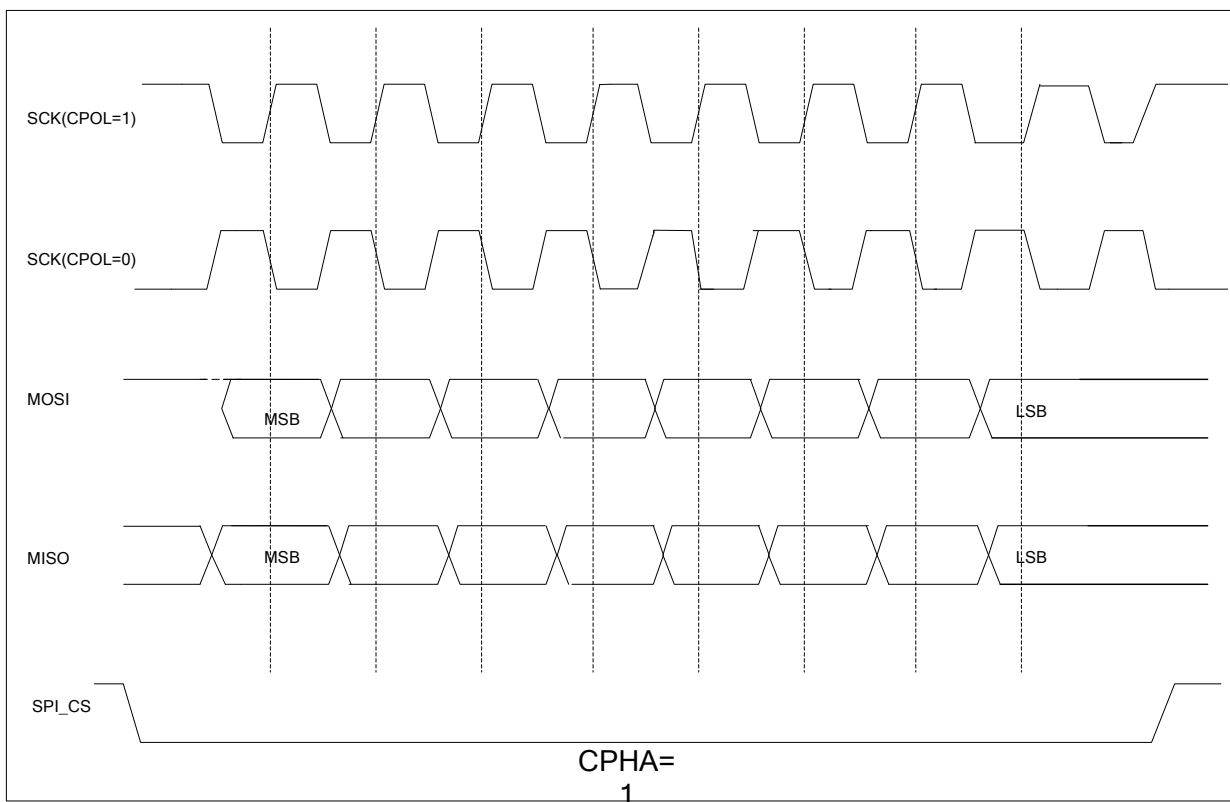
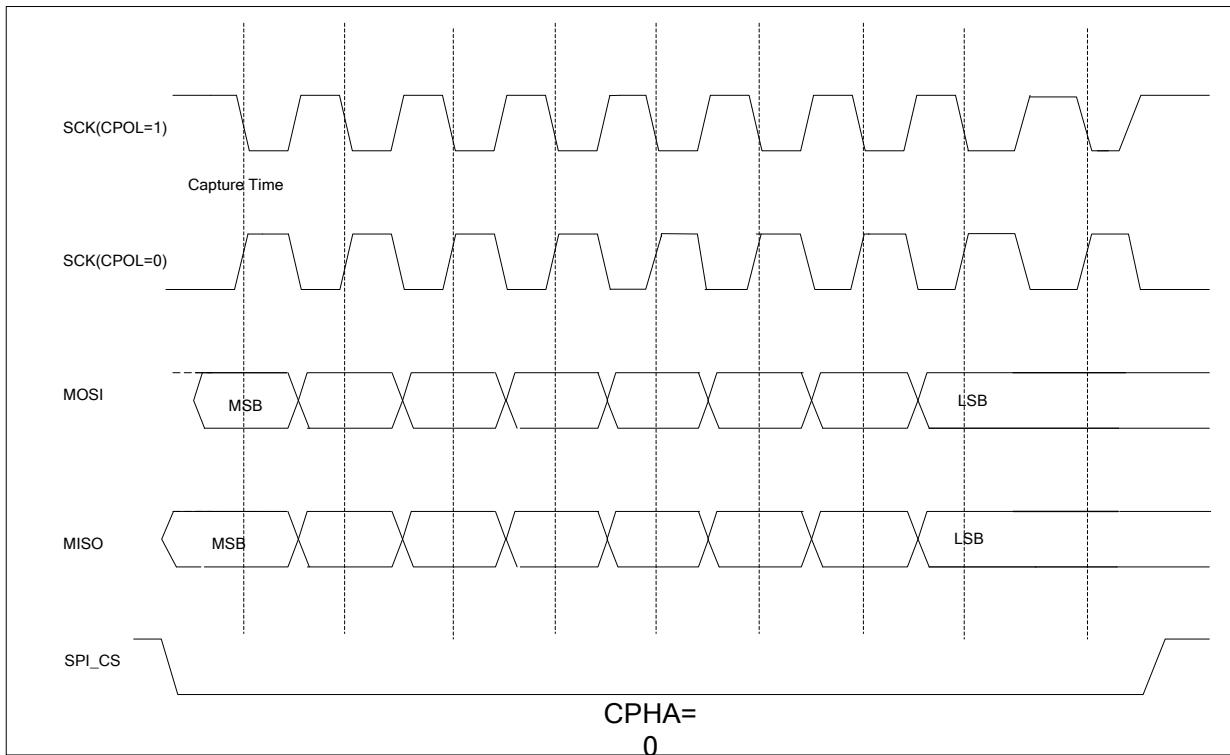
SPI 默认为主机模式。

在主机模式中，SPI 等待程序向寄存器 SPDAT 中写入数据。如果向 SPDAT 的写入动作完成，传输就开始。在时钟 SCK 的发送沿，数据被移位到输出引脚 MOSI 上。同时，从从机传送过来的另一字节的数据被移位到主机的输入引脚 MISO 上。



## 14.2.5 从机模式传输格式

首先，需要写寄存器 SPICON 中的 MSTR=0，以配置 SPI 进入从机模式。另外配置 SPI\_EN=1 以打开 SPI 模块使能。



从模式数据传输格式

在从机模式中，SPI 等待输入信号 SPI\_CS 的低电平，当抓到 SPI\_CS 的下降沿，传输开始，直到传输完成，SPI\_CS 都需要保持低电平状态。寄存器 SPCON 中 CPHA 的状态决定传输的开始位置，当 CPHA 被清零，从机必须在 SCK 信号的第一个下降沿之前开始传输；当 CPHA 被置位，从机会把 SCK 信号的

第一个下降沿做为传输的开始标志。

## 14.2.6 中断功能

名称	SPI 中断标志描述
SPIF	当传输完成，该标志位被硬件置位
MODF	当 SPI_CS 的状态与主从模式设置有冲突

## 14.3 特殊功能寄存器列表

SPI 模块寄存器基地址: 0x4000B000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	SPICON	R/W	0x0000	SPI 控制寄存器
0x04	SPISTA	R/W	0x0000	SPI 状态寄存器
0x08	SPIDAT	R/W	0x0000	SPI 数据寄存器
0x0C	SPISSN	R/W	0x00FF	SPI 从机选择寄存器

## 14.4 特殊功能寄存器说明

### SPICON(SPI 控制寄存器)

SPICON (SPI 控制寄存器)		基地址: 0x4000B000 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:	SSDIS	SPR[2:0]			CPHA	CPOL	MSTR	SPI_EN
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SSDIS	<b>SS控制位</b> 0: 在主/从模式中打开SPI_CS输入 1: 在主/从模式中关闭 SPI_CS 输入，该情况下不会产生 MODF 中断请求；在从模式中，若 CPHA=0，则该位无效

SPR[2:0]	<b>SPI时钟速率控制位(主模式时用)</b>		
	SPR[2:0]		
	0	0	0
	0	0	1
	0	1	0
	0	1	1
	1	0	0
	1	0	1
	1	1	0
	1	1	1
			Fsys/2
			Fsys/4
			Fsys/8
			Fsys/16
			Fsys/32
			Fsys/64
			Fsys/128
			不产生主时钟
CPHA	<b>时钟相位</b> 0: 表示高位 (MSB) 将在SCK的第一个动态沿之前半个周期被发送 1: 表示高位 (MSB) 将在SCK的第一个动态沿通过MOSI/MISO被发送		
CPOL	<b>时钟极性</b> 0: “sck”在空闲状态时被设置为低电平 1: “sck”在空闲状态时被设置为高电平		
MSTR	<b>SPI模式选择位</b> 0: 从机模式 1: 主机模式		
SPI_EN	<b>SPI使能位</b> 0: 关闭SPI模块 1: 打开 SPI 模块		

### SPISTA(SPI 状态寄存器)

<b>SPISTA (SPI 状态寄存器)</b>			<b>基地址: 0x4000B000</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>	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	SPIF	WCOL	SSERR	MODF
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
<b>位</b>	<b>功能描述</b>							
SPIF	<b>数据传输完成标志位</b> 当传输完成时由硬件置位；传输过程中由硬件复位，也可通过读寄存器“spsta”SPDAT来复位。							

WCOL	<b>写冲突标志位</b> 当写SPDAT冲突时由硬件置位；当传输完成无冲突发生时由硬件复位，也可通过访问寄存器“spsta”SPDAT复位。
SSERR	<b>同步从机错误标志位</b> 在接收完成前当SPI_CS输入有效时，被硬件置位；关闭SPI模块可清除该位（设置spen=0）。
MODF	<b>模式故障标志位</b> 当SPI_CS引脚状态与设置的模式有冲突时，硬件自动置位；当SPI_CS引脚恢复合适的电平状态时，硬件自动复位；也可以由软件读“spsta”寄存器来复位。

### SPIDAT(SPI 数据寄存器)

<b>SPIDAT</b> (SPI 数据寄存器)		<b>基地址: 0x4000B000</b>						
		<b>偏移地址: 08H</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>								
<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>	SPDAT[7:0]							
<b>Write:</b>	SPDAT[7:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
SPIDAT[7:0]	寄存器 SPIDAT 是“接收数据”寄存器的一个读/写缓冲。当向 SPIDAT 中写入数据，是直接写入移位寄存器中（没有传输缓冲）；从 SPIDAT 中读数据，返回的是接收缓冲中的数据，而非移位寄存器。

### SPISSN(SPI 从机选择寄存器)

<b>SPISSN</b> (SPI 从机选择寄存器)		<b>基地址: 0x4000B000</b>						
		<b>偏移地址: 0CH</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>								
<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	X	SSN0
<b>Write:</b>								

Reset:	1	1	1	1	1	1	1	1
--------	---	---	---	---	---	---	---	---

注：SPSSN是一个可读/写寄存器，它的每一位都可用来选择一个独立的外部SPI从机设备。

位	功能描述
SSN0	当芯片做 SPI 通讯的主机时，使用 SSN0 控制位可以控制芯片外部 SPI_CS 引脚的高低电平。 在使能 SPI 情况下（且为主机），写 1 拉高 CS，写 0 拉低 CS。

# 15 I2C 模块

## 15.1 概述

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。

## 15.2 框图

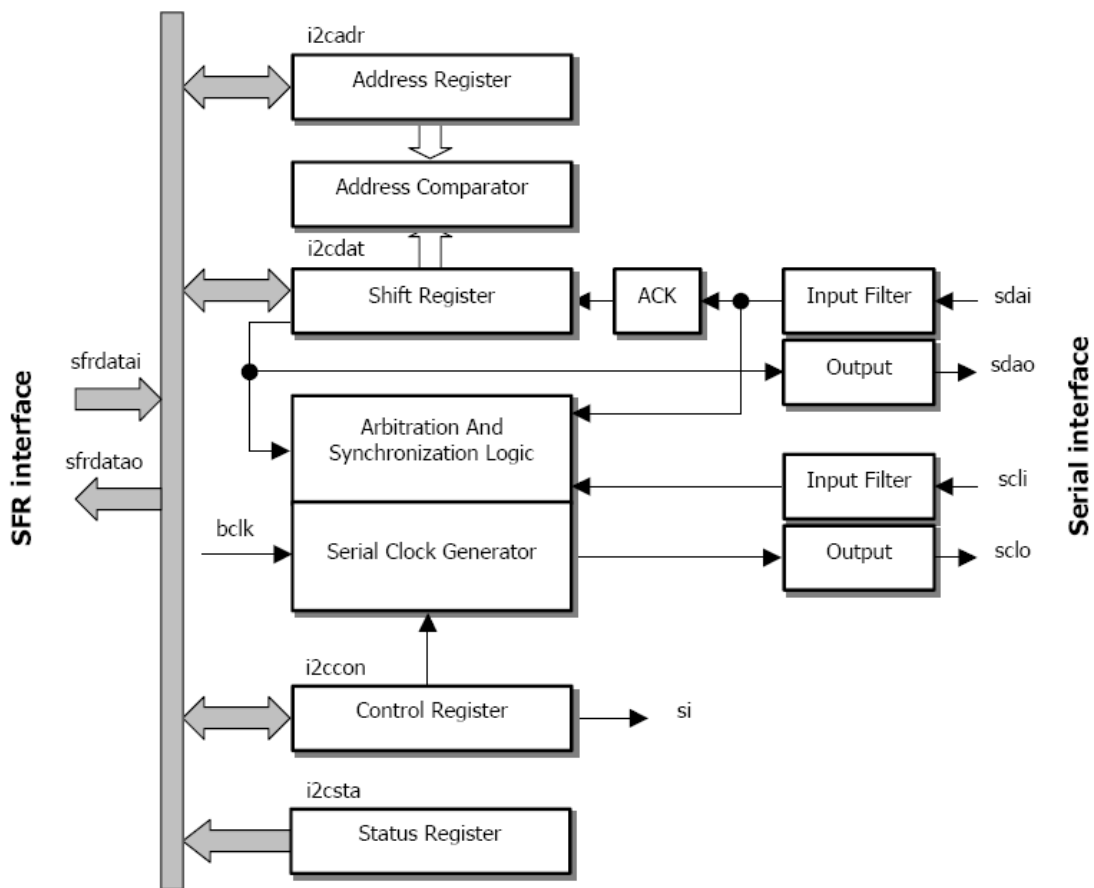


图 1 I2C 模块功能框图

## 15.3 功能描述

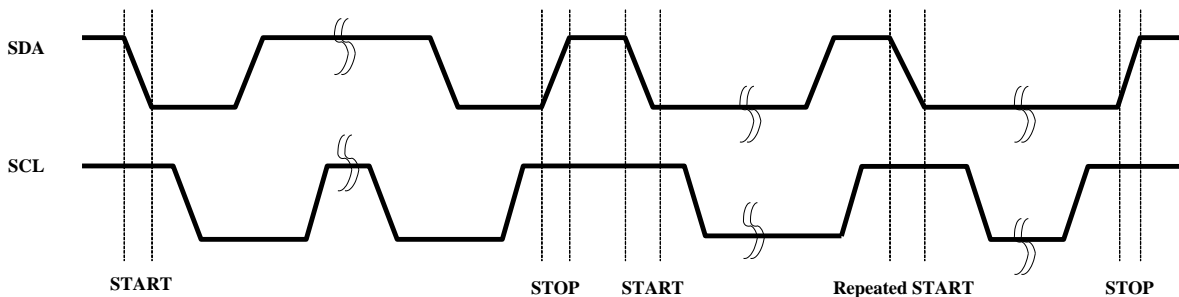
I2C 用两根线实现设备与总线之间的数据传输：串行时钟 SCL 和串行数据 SDA。每一个与总线相连的设备都有一个唯一的地址。支持主机模式(Master)和从机模式(Slave)，允许发送数据(Transmitter)和接收数据(Receiver)。I2C 是一个真正的多主机总线，它包含冲突侦测和仲裁机制，以防止多个主机同时开始数据传输时的数据丢失。

## 15.3.1 起始条件和停止条件

I2C 通讯协议定义了两个特殊的波形：起始条件和终止条件。

在时钟线为高电平时数据线的下降沿定义为起始条件；在时钟线为高电平时数据线的上升沿定义为终止条件。起始条件和终止条件均由主机发出。

主机可以发起和终结一次传输。当主机发送一个起始条件时开始一次传输，发送一个终止条件时结束本次传输。在起始条件和终止条件之间，总线定义为“忙碌”状态。其它主机不应该去试图发起传输。在“忙碌”状态下，如果主机再次发送起始条件，则定义为“重复起始条件”，表示主机希望不放弃总线的情况下开始一次新的传输。发送重复起始条件后，总线仍处于“忙碌”状态，一直到总线出现终止条件。鉴于重复起始条件和起始条件性质完全一致，除非特别声明，本文中采用起始条件来代替两者。



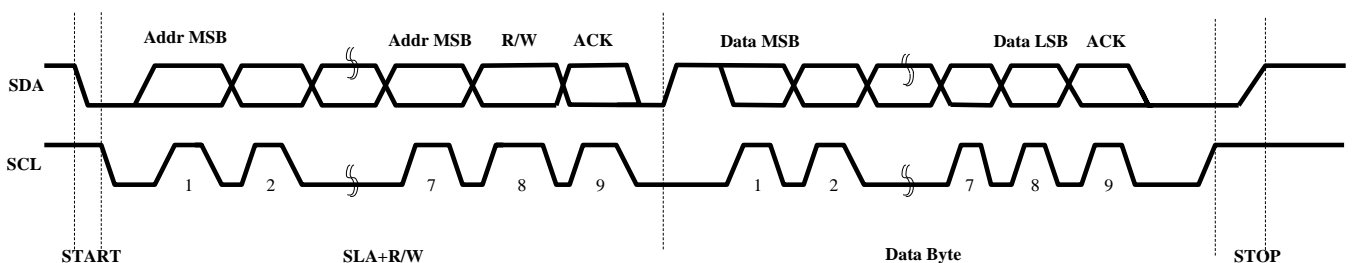
## 15.3.2 数据传输格式

数据传输中数据线上每一位的传输均需要时钟线上一个脉冲。在时钟高电平时数据线应保持恒定。但发送起始条件和终止条件时不需要遵守此规则。

I2C 采用从高到低逐位进行传输，所有数据包（包括地址包）均有 9 位组成，包括 1 个字节和一个应答位。主机负责发出时钟和起始及终止条件，接收者负责给出应答信号。

接收者通过第九个时钟脉冲处将数据线拉低发出“应答(ACK)”信号；或维持第九个脉冲处维持高电平表示“不应答(NACK)”信号。当接收方接收到最后一个字节，或因某种原因无法继续接收数据时，应回应“不应答(NACK)”信号。

一次传输通常包括一个起始条件，地址+读/写，一个或多个数据包和一个终止条件。仅包含起始条件和终止条件的数据格式是不合通讯规则的。



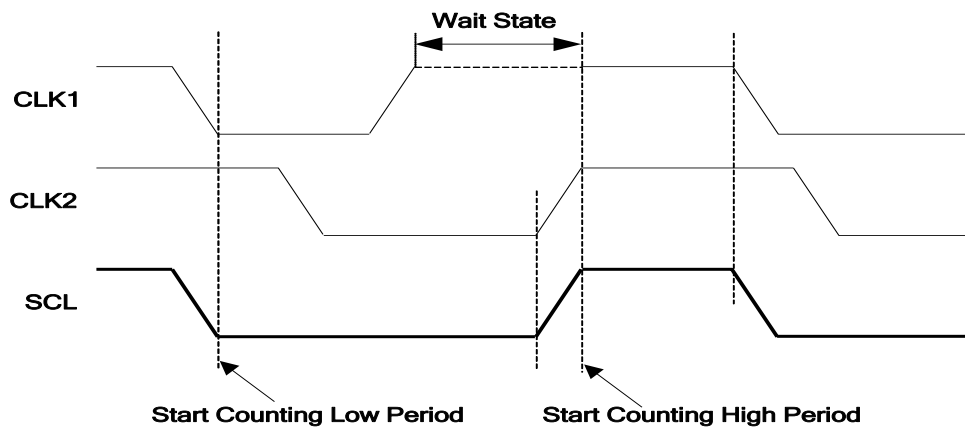
## 15.3.3 时钟同步和数据仲裁

当多个主机同时希望控制总线时，总线将依据“线与”原则决定时钟线高低电平。

### 时钟同步：

时钟线由高到低跳变将导致所有参与传输的器件开始低电平计时。每一个器件计时达到自己低电平要求时释放时钟线，在时钟线变为高电平之前进入高电平等待期；当所有器件均计满低电平周期时，时钟线才变为高电平。之后所有器件开始对高电平进行计时，第一个计满高电平周期的器件将拉低时钟线，进入下一个时钟周期。

这样，产生的同步 SCL 时钟的低电平周期由低电平时钟周期最长的器件决定，而高电平周期由高电平时钟周期最短的器件决定。



值得注意的是“线与”结构给主机和从机之间的握手信号提供了方便。当主机相对太快或从机需要处理其它事务时，从机可以通过拉低时钟线来拉长时钟线的低电平时间，从而降低通讯频率。从机可以拉长时钟线低电平周期但不会影响到时钟线高电平的周期。

### 数据仲裁

主机只有在总线处于“空闲”状态时才能开始一次传输。两个或多个主机可能在最小保持时间( $t_{HD:STA}$ )内同时发送起始条件，从而在总线上只看到一个起始条件。

由于发送起始条件的主机无法知道是否有其它主机在竞争总线，只能靠时钟高电平时对数据线的仲裁判断哪个主机占用总线。当有主机传输低电平时，传输高电平的主机将失去仲裁，断开它的数据输出级。失去仲裁的主机将继续发送时钟，直到当前传输字节发送完毕。当两个主机同时访问一个从机时，可能会顺利通过地址阶段，仲裁会继续比较数据位（如果是主机发送器）或者比较响应位（如果是主机接收器）。因此I2C 总线的地址和数据信息由赢得仲裁的主机决定，在仲裁过程中不会丢失信息

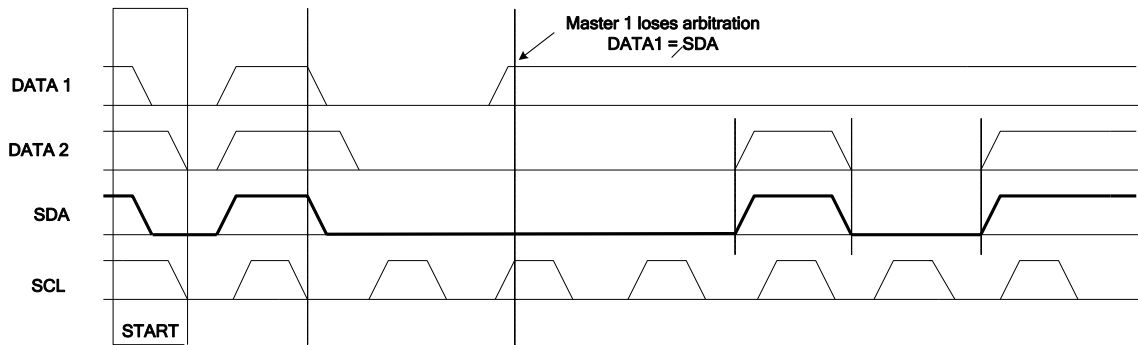
如果该主机同时开启了从机模式，在发送地址阶段失去仲裁后应检测线上的地址是否与自己相匹配；如果是对自己的访问，应立即切换到从机模式，接收信息。

在串行传输时，当重复起始条件或停止条件发送到I2C总线的时候，仲裁过程仍在进行。

仲裁不能在下面情况之间进行：

1. 重复起始条件和数据
2. 数据和终止条件
3. 终止条件和重复起始条件

从机不参与仲裁。

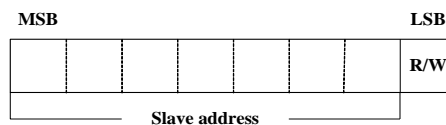


### 15.3.47 位地址格式

I2C 总线的寻址过程是通常在起始条件后的第一个字节决定了主机选择哪一个从机。从机的地址匹配单元检验所收到的地址是否与寄存器 I2CADR 中的 7 位地址相匹配。如果广播呼叫地址使能位 I2CADR[0]被置位，也将检测是否与广播呼叫地址 00H 相匹配。当地址匹配时，控制单元将产生合适的动作及相应的状态码。

第一个字节的头 7 位组成了从机地址，最低位 LSB 是第 8 位，它决定了报文的方向。第一个字节的最低位是“0”表示主机会写信息到被选中的从机，“1”表示主机会向从机读信息。

当发送了一个地址后，系统中的每个器件都在起始条件后将头 7 位与它自己的地址比较，如果一样，器件会认为它被主机寻址，至于是从机接收器还是从机发送器都由 R/W 位决定。



起始条件后第一字节

### 15.3.5 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]。

### 15.3.6 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。

## 15.3.7 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

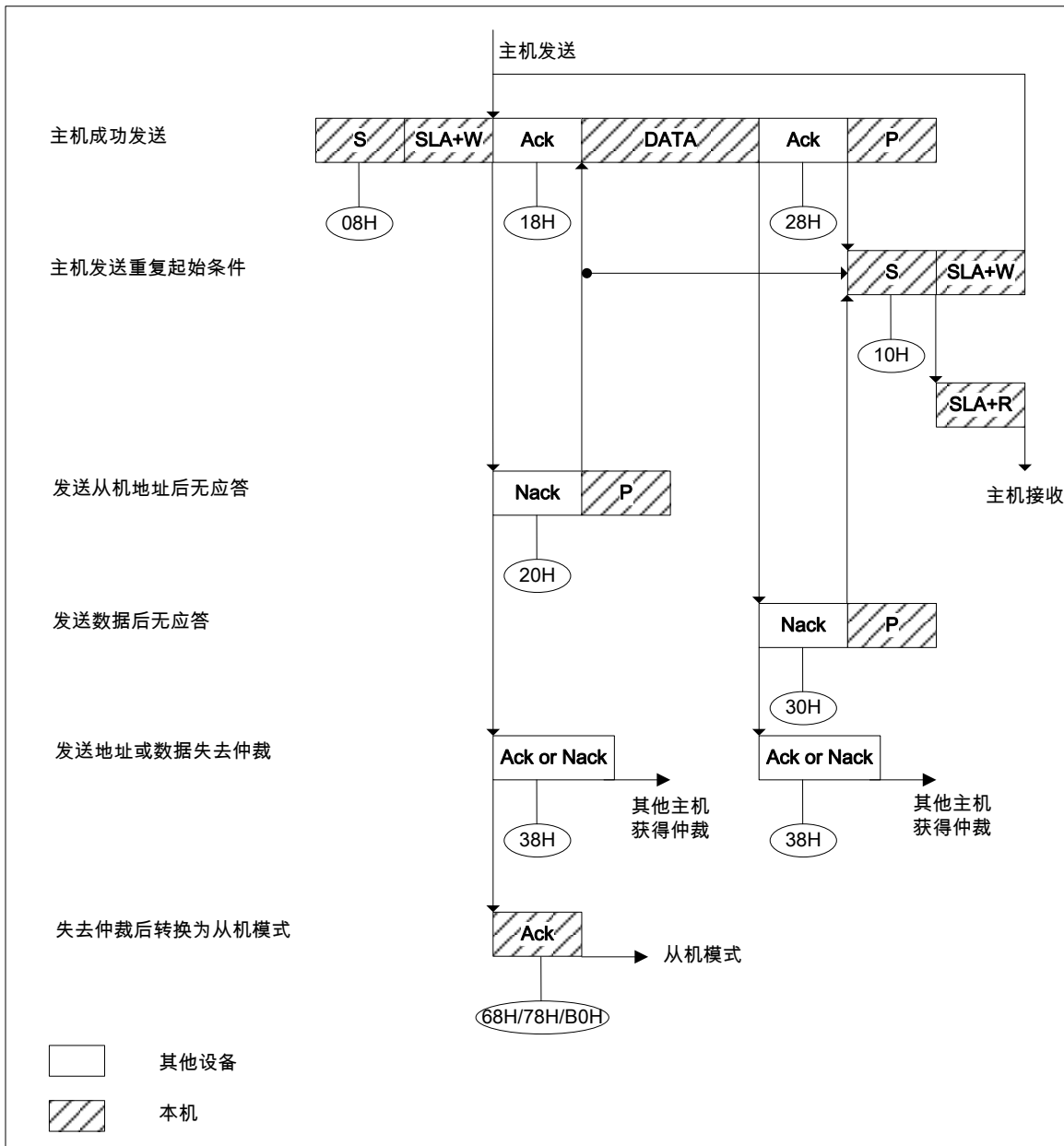
圆形用于表示中断标志已被置起。其中的数字表示当前状态寄存器 I2CSTA 中被掩去低三位的状态码。在 SI 被清除之前，I2C 通讯会暂停，应用软件必须决定是继续通讯还是终止当前传输。对每一个状态码，所需要的软件动作和随后的传输细节均有描述。

### 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 或者加载 SLA+R	X X	0 0	0 0	X X	同上 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 或无动作 或无动作 或无动作	0 1 0 1	0 0 1 1	0 0 0 0	X X X 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
20H	SLA+W 已被发送；	加载数据字节 或无动作	0 1	0 0	0 0	X X	数据字节将被发送；ACK 将被接收 重复起始条件将被发送

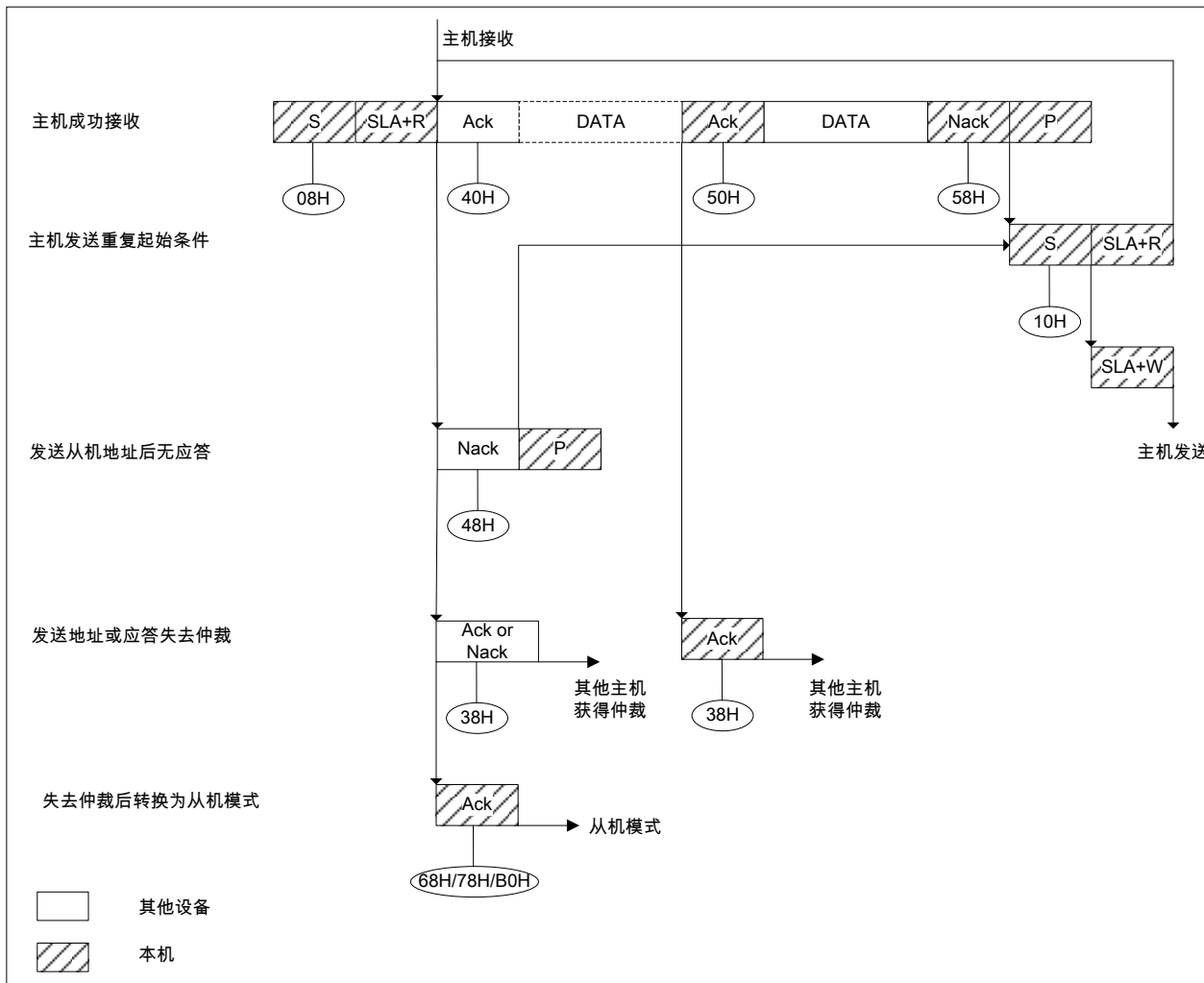
	“not ACK” 已被接收	或无动作 或无动作	0 1	1 1	0 0	X X	终止条件将被发送；sto 标志将被复位 起始条件被发送后将再发送一个终止条件；sto 标志位将被复位
28H	i2cdat 的数据字节已被发送；ACK 已被接收	加载数据字节 或无动作 或无动作 或无动作	0 1 0 1	0 0 1 1	0 0 0 0	X X X X	数据字节将被发送；将发送 ACK 字节 重复起始条件将被发送。 终止条件将被发送；sto 标志将被复位 起始条件被发送后将再发送一个终止条件；sto 标志将被复位
30H	i2cdat 的数据字节已被发送	数据字节 或无动作 或无动作 或无动作	0 1 0 1	0 0 1 1	0 0 0 0	X X X 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
38H	SLA+R/W 或数据字节仲裁失败	无动作 或无动作	0 1	0 0	0 0	X 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



## 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 或者 加载 SLA+W	X X	0 0	0 0	X X	同上 SLA+W 将被发送； I2C 将转换为“主接收器”模式
38H	“not ACK”位仲裁失败	无动作 或者无动作	0 1	0 0	0 0	X 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
40H	SLA+R 已被发送； ACK 已被接收	无动作 或者无动作	0 0	0 0	0 0	0 1	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”
48H	SLA+R 已被发送； “not ACK”已被接收	无动作 或无动作 或无动作	1 0 1	0 1 1	0 0 0	X X 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
50H	数据字节已被接收； 已返回 ACK	读取数据字节 或读取数据字节	0 0	0 0	0 0	0 1	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
58H	数据字节将被接收； 已返回“not ACK”	读取数据字节 或读取数据字节 或读取数据字节	1 0 1	0 1 1	0 0 0	X X 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



## 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。

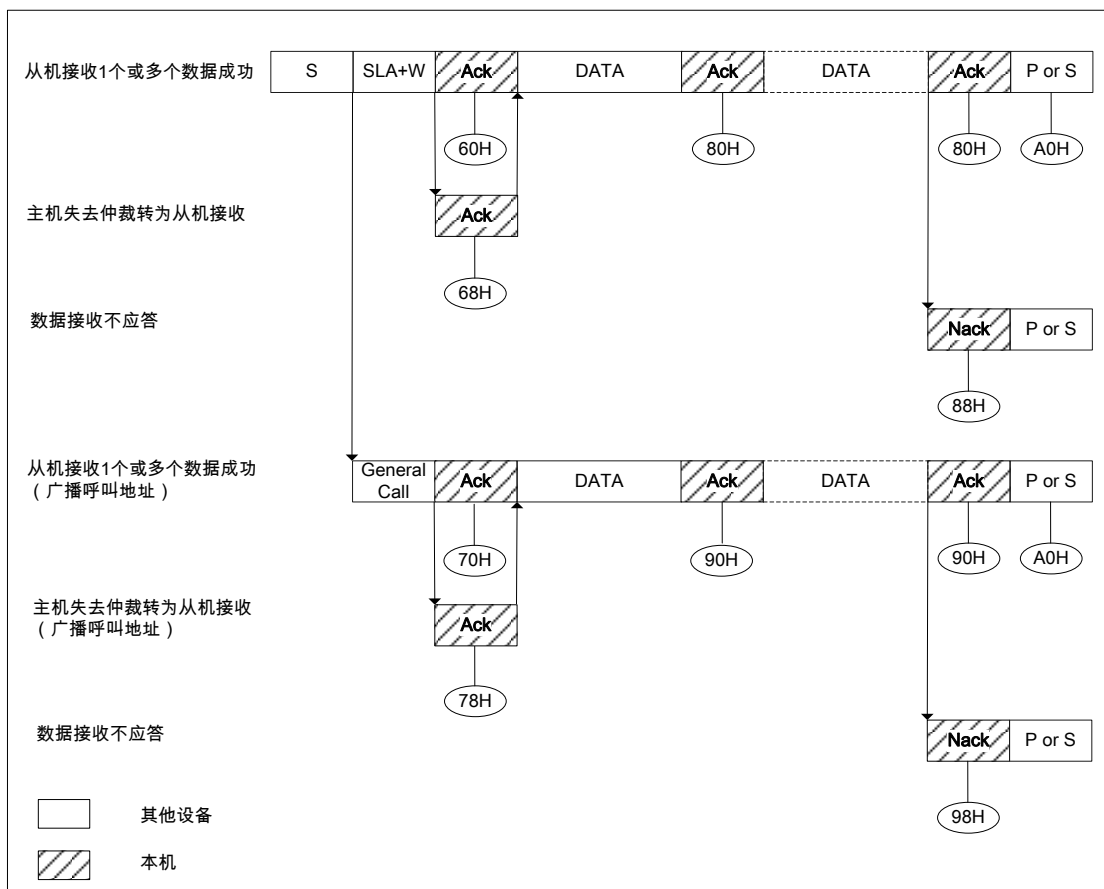
进入从机模式前，需设置从机地址，I2CADR 中 I2CADR[7..1]位为从机地址。如果 I2CADR[0]置位，从机也将响应广播呼叫地址(00H)；否则将不响应广播呼叫地址。

从机模式中，I2C 模块等待总线对本地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 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 从机发送模式:

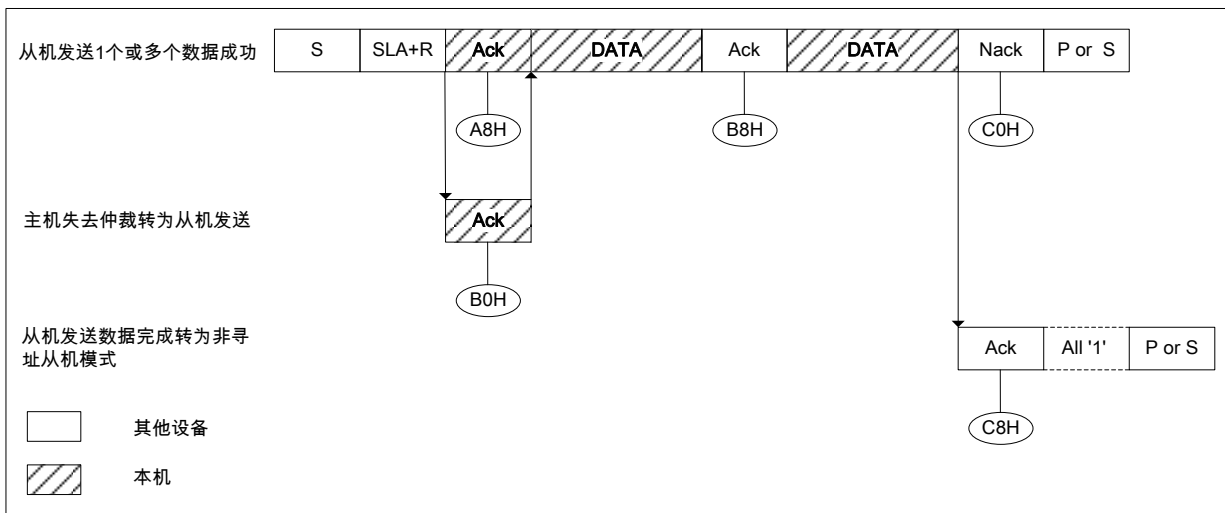
从机发送模式中，从机发送一系列数据到主机。

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 模块等待总线对本地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
B0H	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
C0H	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

## 15.4 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

## 15.5 特殊功能寄存器说明

### I2CDAT(I2C 数据寄存器)

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

寄存器I2CDAT是将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(I2C 地址寄存器)

I2CADR (I2C 地址寄存器)		基地址： 0x4000A000 偏移地址： 04H						
	Bit15	14	13	12	11	10	9	Bit8
<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
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	I2CADR[7:0]							

<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
I2CADR[7:1]	<b>I2C从机地址(7位)</b>
I2CADR[0]	<b>广播呼叫地址确认位</b> 当此位置1时，广播呼叫地址可以被响应，否则不能被响应。

## I2CCON(I2C 控制寄存器)

<b>I2CCON</b> (控制寄存器)		<b>基地址: 0x4000A000</b> <b>偏移地址: 08H</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	CR[9:3]						
<b>Write:</b>								
<b>Reset:</b>	0	1	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>	CR2	ENS1	STA	STO	SI	AA	CR[1:0]	
<b>Write:</b>								
<b>Reset:</b>	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=fsys/( CR[9:0]+1)/4
ENS1	<b>I2C使能位</b> 1: 打开I2C模块; 0: 关闭I2C模块;
STA	<b>开始标志位</b> 1: 检查I2C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I2C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；I2CADR[0]被置位的情况下接收到 <b>广播呼叫地址</b> ；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(I2C 状态寄存器)

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]	<p><b>I2C状态码</b></p> <p>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。</p>

## 16 RTC 模块

### 16.1 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）

### 16.2 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）
- TOUT PIN 可输出频率为 1Hz/2Hz/4Hz/8Hz 的方波信号
- 可输出每秒补偿的校验脉冲
- 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

### 16.3 时钟校正

由 RTC 模块读取 TPS 输出的温度值，根据低频晶体时钟 Fosc 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：

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

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)。

## 16.4 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（0x00040104~0x00040138），然后再写入到对应的 RTC 模块的寄存器中。

寄存器偏移地址 (RTC 模块寄存器基地址： 0x4000C000)	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 0x0FC0 地址的 bit2 为 1 时，上电后，系统自动加载 Info\_Flash 中的 RTC 补偿系数（0x00040104~0x00040138）到对应的寄存器中。
- 2.（0x00040000+0x13c）存放从（0x00040000+0x104）到（0x00040000+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，上电后也需要一同装载。
- 4.自动加载功能为系统发生某些复位，CPU 运行之前自动完成的，触发自动加载的复位：系统软复位 NVIC\_SystemReset(), WDT 复位，BOR 复位，外部引脚复位 PINRESET，POR 复位，软件 POR 复位（SoftPOR）。
- 5.自动装载以 32 位形式加载。

## 16.5 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。

## 16.6 中断功能

RTC 一共提供 8 种中中断源, 共用 MCU 的 IRQ-RTC 中断向量 12。RTC 的 8 种中断源由 RTCIE(AAH) 分别控制其使能。

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

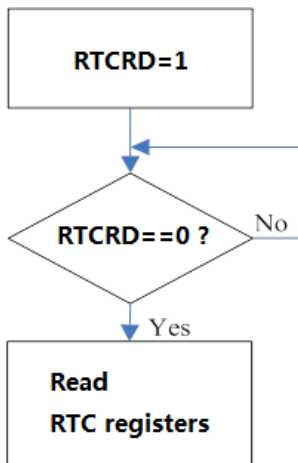
秒计数器  $SECR$  加 1 时, 产生一个秒中断, SECF 被置为 1。

对该位写 0 清标志。

## 16.7 RTC 指示寄存器读写流程

### 16.7.1 读取 RTC 指示寄存器流程

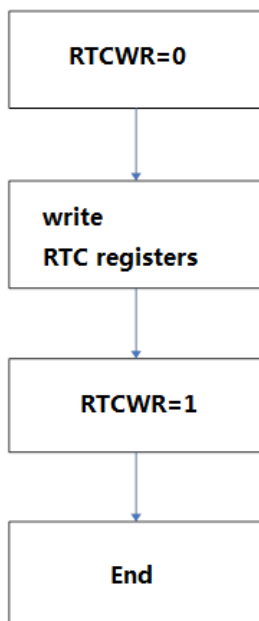
在用户需要读取 RTC 指示寄存器 (SECR, MINR, HRR, DAYR, MTHR, YRR, DOWR) 的时候, 用户应该按照以下流程操作:



注：辅助 RTC 读流程与上图一致，只是读控制寄存器为 RTCRD2

## 16.7.2 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

## 16.8 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 RTCCOUNT 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。

## 16.9 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 作为时钟源。当外部 OSC 停振时，系统时钟可切换至内部低频 RC，启用辅助 RTC 可获得持续计时时间。辅助 RTC 读写流程与第一套类似，其有单独的控制信号 RTCWR2 和 RTCRD2。

## 16.10 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	0x0001	日寄存器（写保护）
0x28	MONTHR	R/W	0x0001	月寄存器（写保护）
0x2C	YEARR	R/W	0x0000	年寄存器（写保护）
0x30	WEEKR	R/W	0x0001	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	0x0000	Mems 控制寄存器（保持默认值）
0x80	MCON23	R/W	0x0000	Mems 控制寄存器（保持默认值）
0x84	MCON45	R/W	0x8600	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

辅助 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

## 16.11 特殊功能寄存器说明

### RTCCON(RTC 控制寄存器)

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

<b>Read:</b>	X	RTC2EN	RTC1EN	X	TOUT[2:0]			AutoC
<b>Write:</b>								
<b>Reset:</b>	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[2:0]	<b>TOUT 输出频率</b>																																														
	<table border="1"> <thead> <tr> <th colspan="3">TOUT[2:0]</th> <th>TOUT</th> <th>TOUT (PLL 使能)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>32768Hz</td> <td>32768Hz</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>内部 RTC 1Hz</td> <td>由高频补偿 128Hz 后得到的 1Hz</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>内部 RTC 2Hz</td> <td>由高频补偿 128Hz 后得到的 2Hz</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>内部 RTC 4Hz</td> <td>由高频补偿 128Hz 后得到的 4Hz</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>内部 RTC 8Hz</td> <td>由高频补偿 128Hz 后得到的 8Hz</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>内部 RTC 1Hz</td> <td>内部 RTC 1Hz</td> </tr> </tbody> </table>		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
	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																																											
<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。																																															
AutoC																																															

## RTCIE(RTC 中断使能寄存器)

<b>RTCIE</b> (RTC 中断使能寄存器)			<b>基地址: 0x4000C000</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>	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>	ALMIE	RTC2IE	RTC1IE	MTHIE	DAYIE	HRIE	MINIE	SECIE
<b>Write:</b>								
<b>Reset:</b>	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	<b>RTC 定时器 2 中断使能位</b> 0: 关闭 1: 打开
RTC1IE	<b>RTC 定时器 1 中断使能位</b> 0: 关闭 1: 打开
MTHIE	<b>RTC 月中断使能位</b> 0: 关闭 1: 打开
DAYIE	<b>RTC 日中断使能位</b> 0: 关闭 1: 打开
HRIE	<b>RTC 小时中断使能位</b> 0: 关闭 1: 打开
MINIE	<b>RTC 分钟中断使能位</b> 0: 关闭 1: 打开
SECIE	<b>RTC 秒中断使能位</b> 0: 关闭 1: 打开

### RTCIF(RTC 中断标志寄存器)

<b>RTCIF</b> (RTC 中断标志寄存器)		<b>基地址: 0x4000C000</b> <b>偏移地址: 08H</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>								
<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>	ALMIF	RTC2IF	RTC1IF	MTHIF	DAYIF	HRIF	MINIF	SECIF
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

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

	0: 未产生中断 1: 产生中断, 写 0 清 0
RTC1IF	<b>RTC 定时器 1 中断标志位</b> 0: 未产生中断 1: 产生中断, 写 0 清 0
MTHIF	<b>RTC 月中断标志位</b> 0: 未产生中断 1: 产生中断, 写 0 清 0
DAYIF	<b>RTC 日中断标志位</b> 0: 未产生中断 1: 产生中断, 写 0 清 0
HRIF	<b>RTC 小时中断标志位</b> 0: 未产生中断 1: 产生中断, 写 0 清 0
MINIF	<b>RTC 分钟中断标志位</b> 0: 未产生中断 1: 产生中断, 写 0 清 0
SECIF	<b>RTC 秒中断标志位</b> 0: 未产生中断 1: 产生中断, 写 0 清 0

### ALMR(闹钟寄存器)

<b>ALMR</b> (闹钟寄存器)		<b>基地址: 0x4000C000</b> <b>偏移地址: 0CH</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	ALMH[5:0]				
<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	ALMM[5:0]					
<b>Write:</b>								
<b>Reset:</b>	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)**

<b>RTCTMR1</b> (RTC 定时器 1 寄存器)			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 10H</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>	CNT[15:8]							
<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>	CNT[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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>

**RTCTMR2(RTC 定时器 2 寄存器)**

<b>RTCTMR2</b> (RTC 定时器 2 寄存器)			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 14H</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>	CNT[15:8]							
<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>	CNT[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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</p>

=1\*0.0625S=0.0625S 的计时周期后，置位 RTC2IF 标志。

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。

## SECR(RTC 秒寄存器)

SECR (RTC 秒寄存器)			基地址： 0x4000C000 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	SEC[5:0]					
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SEC[5:0]	秒计数器： 可设范围：0-59。写入 0-59 以外的任何数值，寄存器无法写入成功。

## MINR(RTC 分寄存器)

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

位	功能描述
MIN[5:0]	分计数器： 可设范围：0-59。写入 0-59 以外的任何数值，寄存器无法写入成功。

**HOURR(RTC 时寄存器)**

<b>HOURR</b> (RTC 时寄存器)			基地址: <b>0x4000C000</b> 偏移地址: <b>20H</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>								
<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	HOUR[4:0]				
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	时计数器: 可设范围: 0-23。写入 0-23 以外的任何数值, 寄存器无法写入成功。

**DAYR(RTC 日寄存器)**

<b>DAYR</b> (RTC 日寄存器)			基地址: <b>0x4000C000</b> 偏移地址: <b>24H</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>								
<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	DAY[4:0]				
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DAY[4:0]	日计数器: 可设范围: 1-28/29/30/31。写入与年、月不匹配的任何数值, 寄存器无法写入成功。

**MONTHR(RTC 月寄存器)**

<b>MONTHR</b> (RTC 月寄存器)			基地址: <b>0x4000C000</b> 偏移地址: <b>28H</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>								
<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	MONTH[3:0]			
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
MONTH[3:0]	月计数器： 可设范围：1-12。写入 1-12 以外的任何数值，寄存器无法写入成功。

### YEARR(RTC 年寄存器)

<b>YEARR</b> (RTC 年寄存器)		<b>基地址： 0x4000C000</b> <b>偏移地址： 2CH</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>								
<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	YEAR[6:0]						
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	年计数器： 可设范围：0-99。写入 0-99 以外的任何数值，寄存器无法写入成功。

### WEEKR(RTC 周寄存器)

<b>WEEKR</b> (RTC 周寄存器)		<b>基地址： 0x4000C000</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>	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	WEEK[2:0]		
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
WEEK[2:0]	周计数器： 可设范围：1-7。写入 1-7 以外的任何数值，寄存器无法写入成功。

### RTCCNTH(RTC 校时次数寄存器高 16 位)

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

### RTCCNTL(RTC 校时次数寄存器低 16 位)

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

### RTCRD(RTC 读控制寄存器)

<b>RTCRD</b> (RTC 读控制寄存器)	<b>基地址:</b> 0x4000C000 <b>偏移地址:</b> 3CH
	<b>Bit15 14 13 12 11 10 9 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
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	X	X	X	X	X	X	X	READFL AG
<b>Write:</b>								
<b>Reset:</b>	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)

<b>RTCWR</b> (RTC 写控制寄存器)			<b>基地址: 0x4000C000</b> <b>偏移地址: 40H</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>								
<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	X	UPDATE
<b>Write:</b>								
<b>Reset:</b>	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)

<b>DFAH</b> (常数项高 7 位)			<b>基地址: 0x4000C000</b> <b>偏移地址: 50H</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>								
<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	DFx[22:16]						
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

**DFAL(常数项低 16 位)**

<b>DFAL</b> (常数项低 16 位)			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 54H</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>	DFx[15:8]							
<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>	DFx[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

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

**DFBH(常数项高 7 位)**

<b>DFBH</b> (常数项高 7 位)			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 58H</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>								
<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	DFx[22:16]						
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

**DFBL(常数项低 16 位)**

<b>DFBL</b> (常数项低 16 位)			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 5CH</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>	DFx[15:8]							
<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>	DFx[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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

### DFCH(常数项高 7 位)

<b>DFCH</b> (常数项高 7 位)			<b>基地址: 0x4000C000</b> <b>偏移地址: 60H</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>								
<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	DFx[22:16]						
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

### DFCL(常数项低 16 位)

<b>DFCL</b> (常数项低 16 位)			<b>基地址: 0x4000C000</b> <b>偏移地址: 64H</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>	DFx[15:8]							
<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>	DFx[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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)**

<b>DFDH</b> (常数项高 7 位)			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 68H</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>								
<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	DFx[22:16]						
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

**DFDL(常数项低 16 位)**

<b>DFDL</b> (常数项低 16 位)			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 6CH</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>	DFx[15:8]							
<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>	DFx[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>位</b>	<b>功能描述</b>
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)**

<b>DFEH</b> (常数项高 7 位)			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 70H</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>								
<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	DFx[22:16]						
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

**DFEL(常数项低 16 位)**

<b>DFEL</b> (常数项低 16 位)		<b>基地址: 0x4000C000</b> <b>偏移地址: 74H</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>	DFx[15:8]							
<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>	DFx[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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)**

<b>Toff</b> (温度偏置寄存器)		<b>基地址: 0x4000C000</b> <b>偏移地址: 78H</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>	DFx[15:8]							
<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>	DFx[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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)**

<b>DFiH</b> (常数项高 5 位)			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 88H</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>								
<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	DFi[20:16]				
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

**DFiL(常数项低 16 位)**

<b>DFiL</b> (常数项低 7 位)			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 8CH</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>	DFi[15:8]							
<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>	DFi[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DFi[22:0]	23 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**

<b>CTRLByFlash</b>			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 100H</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>	FLASH[7:0]							

<b>Write:</b>								
<b>Reset:</b>	X	X	X	X	X	X	X	X
	<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	Reseved	AUTOR ELOAD	LRC_CT RL	X
<b>Write:</b>								
<b>Reset:</b>	X	X	X	X	X	X	X	X

位	功能描述
FLASH[7:0]	代码空间加密位 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 <span style="background-color: yellow;">可以写</span>
AUTORELOAD	自动装载使能位 =1, 自动装载功能使能 =0, 自动装载功能屏蔽 该寄存器位只能读, 不能写
LRC_CTRL	低频 RC 控制位 =1: 打开低频 RC =0: 关闭低频 RC 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。 该寄存器位可读可写。

注: 其他位不可以修改, 保持默认值。该寄存器上电初始值等于 0XFC0 地址控制字的值

## LRCCOMMND(LRC 写关闭命令寄存器)

<b>LRCCOMMAND</b> (LRC 写关闭命令寄存器)		基地址: 0x4000C000 偏移地址: 104H						
	<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>	LRCCOMMAND[15:8]							
<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>	LRCCOMMAND[7:0]							
<b>Write:</b>								
<b>Reset:</b>	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)

<b>SECR2</b> (辅助 RTC 秒寄存器)			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 200H</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>								
<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	SEC[5:0]					
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
SEC[5:0]	辅助秒计数器（以 LRC 为时钟基准）： 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。

### MINR2(辅助 RTC 分寄存器)

<b>MINR2</b> (辅助 RTC 分寄存器)			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 204H</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>								
<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	MIN[5:0]					
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
MIN[5:0]	辅助分计数器： 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。

### HRR2(辅助 RTC 时寄存器)

<b>HRR2</b> (辅助 RTC 时寄存器)			基地址: 0x4000C000 偏移地址: 208H					
	<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	HOUR[4:0]				
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	辅助时计数器 (以 LRC 为时钟基准): 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>DAYR2</b> (辅助 RTC 日寄存器)			基地址: 0x4000C000 偏移地址: 20CH					
	<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	DAY[4:0]				
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DAY[4:0]	辅助日计数器 (以 LRC 为时钟基准): 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。

### MOUTHR2(辅助 RTC 月寄存器)

<b>MOUTHR2</b> (辅助 RTC 月寄存器)			基地址: 0x4000C000 偏移地址: 210H					
---------------------------------	--	--	-------------------------------	--	--	--	--	--

	<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	MONTH[3:0]			
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
MONTH[3:0]	辅助月计数器（以 LRC 为时钟基准）： 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>YEARR2</b> (辅助 RTC 年寄存器)		基地址: <b>0x4000C000</b> 偏移地址: <b>214H</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>								
<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	YEAR[6:0]						
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	辅助年计数器（以 LRC 为时钟基准）： 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)

<b>WEEKR2</b> (辅助 RTC 周寄存器)		基地址: <b>0x4000C000</b> 偏移地址: <b>218H</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>								
<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	WEEK[2:0]		
<b>Write:</b>								
<b>Reset:</b>	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)

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
<b>Read:</b>	X	X	X	X	X	X	X	CAL16
<b>Write:</b>								
<b>Reset:</b>	0	1	1	1	1	1	1	1
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	CAL15	CAL14	CAL13	CAL12	CAL11	CAL10	CAL9	CAL8
<b>Write:</b>								
<b>Reset:</b>	1	1	1	1	1	1	1	1
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
<b>Write:</b>								
<b>Reset:</b>	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

## RTC2RD2(辅助 RTC 读控制寄存器)

RTC2RD2 (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
<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
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	X	X	X	X	X	X	X	READFLAG
<b>Write:</b>								
<b>Reset:</b>	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

### RTCWR2(辅助 RTC 写控制寄存器)

<b>RTCWR2</b> (辅助 RTC 写控制寄存器)		<b>基地址: 0x4000C000</b> <b>偏移地址: 224H</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>								
<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	X	UPDATE
<b>Write:</b>								
<b>Reset:</b>	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\_LRV(RTC2 校正寄存器自动测试)

<b>FRE_LRC</b> (RTC2 校正寄存器自动测试)		<b>基地址: 0x4000C000</b> <b>偏移地址: 228H</b>						
	<b>Bit23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>Bit16</b>
<b>Read:</b>	X	X	X	X	X	X	X	LRC16
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
	<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>	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>

## 17 TBS 模块

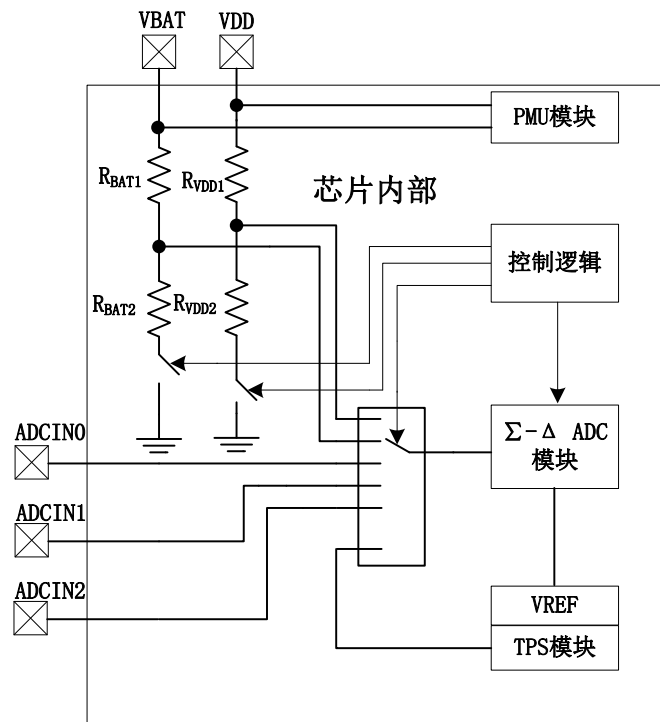
### 17.1 概述

芯片可以对温度和 5 路 ADC (VDD,VBAT,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。

### 17.2 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 (VDD,VBAT,ADCIN0,ADCIN1,ADCIN2)



VBAT 检测内阻为 30K ( $R_{BAT1}$  为 24K,  $R_{BAT2}$  为 6K), VDD 检测内阻为 42K ( $R_{VDD1}$  为 36K,  $R_{VDD2}$  为 6K)。ADCIN0,ADCIN1,ADCIN2 内部均无内阻。

### 17.3 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	VDDDAT	R/W	0x0000	电源电压测量输出值
1CH	ADC0DAT	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
2CH				
34H	ADC0CMP	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

### 17.4 特殊功能寄存器说明

#### TBSCON(TBS 设置寄存器)

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	ADC1C MPE <sub>n</sub>	ADC0C MPE <sub>n</sub>	X	Filter1	Filter0
Write:								
Reset:	0	0	0	0	0	0	0	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	RSRV	VbatCM PE <sub>n</sub>	ADC2En	ADC1En	ADC0En	VDDE <sub>n</sub>	VbatEn	TPSE <sub>n</sub>
Write:								
Reset:	0	0	0	0	0	0	0	1

位	功能描述
ADC1CMPE <sub>n</sub>	此位只在 ADC1En 使能的情况下才有效 ADC1CMPE <sub>n</sub> =0: 关闭 ADC1DAT 与 ADC1CMP 两个寄存器的比较 ADC1CMPE <sub>n</sub> =1: 使能 ADC1DAT 与 ADC1CMP 两个寄存器的比较
ADC0CMPE <sub>n</sub>	此位只在 ADC0En 使能的情况下才有效

	ADC0CMPEn=0: 关闭 ADC0DAT 与 ADC0CMP 两个寄存器的比较 ADC0CMPEn=1: 使能 ADC0DAT 与 ADC0CMP 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
<b>RSRV</b>	<b>建议保留此位为 0.</b>
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
VDDEn	VDDEn=0: VDD 测量关闭 VDDEn=1: VDD 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)

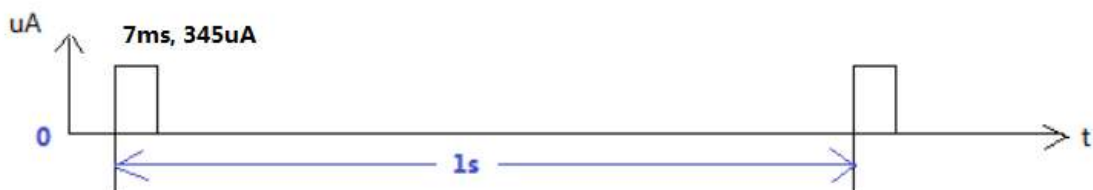
<b>TBSPRD</b> (TBS 打开周期设置寄存器)		基地址: <b>0x4000E000</b> 偏移地址: <b>04H</b>						
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	X	X	X	X	<b>ADC2P</b>	<b>ADC2P</b>	<b>ADC1P</b>	<b>ADC1P</b>
<b>Write:</b>					<b>RD1</b>	<b>RD0</b>	<b>RD1</b>	<b>RD0</b>
<b>Reset:</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>	<b>ADC0P</b>	<b>ADC0P</b>	VDDPR	VDDPR	VbatPRD	VbatPRD	TPSPRD	TPSPRD
<b>Write:</b>	<b>RD1</b>	<b>RD0</b>	D1	D0	1	0	1	0
<b>Reset:</b>	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
ADC0PRD[1..0]	ADC0PRD[1..0]=00b ADC 通道 0 1s 打开一次 ADC0PRD[1..0]=01b ADC 通道 0 4s 打开一次 ADC0PRD[1..0]=10b ADC 通道 0 8s 打开一次 ADC0PRD[1..0]=11b ADC 通道 0 32s 打开一次
VDDPRD[1..0]	VDDPRD [1..0]=00b VDD 检测 1s 打开一次 VDDPRD [1..0]=01b VDD 检测 4s 打开一次 VDDPRD [1..0]=10b VDD 检测 8s 打开一次 VDDPRD [1..0]=11b VDD 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)

<b>TBSIE</b> (TBS 中断使能寄存器)	<b>基地址: 0x4000E000</b> <b>偏移地址: 08H</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	ADC1C	ADC0C
<b>Write:</b>							MPIE	MPIE
<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>	RSRV	VbatC	ADC2IE	ADC1IE	ADC0IE	VDDIE	VbatIE	TPSIE
<b>Write:</b>		MPIE						
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
RSRV	建议保持此位为 0
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
VDDIE	VDDIE=0: VDD 测量中断关闭 VDDIE=1: VDD 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 中断标志寄存器)

<b>TBSIF</b> (TBS 中断标志寄存器)		<b>基地址: 0x4000E000</b> <b>偏移地址: 0CH</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	ADC1C	ADC0C
<b>Write:</b>							MPIF	MPIF
<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	VbatC	ADC2IF	ADC1IF	ADC0IF	VDDIF	VbatIF	TPSIF
<b>Write:</b>		MPIF						

<b>Reset:</b>	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
VDDIF	VDDIF=0: 未发生 VDD 测量中断 VDDIF=1: 发生 VDD 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>TMPDAT</b> (温度测量输出寄存器)		<b>基地址: 0x4000E000</b>						
		<b>偏移地址: 10H</b>						
	<b>Bit15...Bit0</b>							
<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]	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>VBATDAT</b> (电池电压测量输出寄存器)		<b>基地址： 0x4000E000</b>						
		<b>偏移地址： 14H</b>						
		<b>Bit15...Bit0</b>						
<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]	电池电压测量输出寄存器，16 位有符号数

电池电压计算公式：

$$\text{VBAT} = 0.0897 * \text{VBATDAT} + 2946.2662;$$

其中： VBAT为实际电池电压（ mV）

### VDDDAT(电源电压测量输出寄存器)

<b>VDDDAT</b> (电源电压测量输出寄存器)		<b>基地址： 0x4000E000</b>						
		<b>偏移地址： 18H</b>						
		<b>Bit15...Bit0</b>						
<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]	电源电压 VDD 测量输出寄存器，16 位有符号数

VDD 电压计算公式：

$$\text{VDD} = 0.0897 * \text{VDDDAT} + 2998.1189;$$

其中： VDD为实际VDD电压（mV）

**ADC0DAT(ADC0 测量输出寄存器)**

<b>ADC0DAT</b> (ADC0 测量输出寄存器)		<b>基地址: 0x4000E000</b>						
		<b>偏移地址: 1CH</b>						
	<b>Bit15...Bit0</b>							
<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 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:

$$VADCIN0 = 0.0128 * ADC0DAT + 425.5623;$$

其中: VADCIN0 为实际 ADC 测量电压 (mV)

**ADC1DAT(ADC1 测量输出寄存器)**

<b>ADC1DAT</b> (ADC1 测量输出寄存器)		<b>基地址: 0x4000E000</b>						
		<b>偏移地址: 20H</b>						
	<b>Bit15...Bit0</b>							
<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 1 测量输出寄存器, 16 位有符号数

ADC 通道 1 测量电压计算公式同通道 0。

**ADC2DAT(ADC2 测量输出寄存器)**

<b>ADC2DAT</b> (ADC2 测量输出寄存器)		<b>基地址: 0x4000E000</b>						
		<b>偏移地址: 24H</b>						
	<b>Bit15...Bit0</b>							
<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 测量电压计算公式同通道 0。

### VBATCMP(电池电压比较寄存器)

<b>VBATCMP</b> (电池电压比较寄存器)		<b>基地址: 0x4000E000</b>						
		<b>偏移地址: 28H</b>						
		<b>Bit15...Bit0</b>						
<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。

### ADC0CMP(ADC0 比较寄存器)

<b>ADC0CMP</b> (ADC0 测量比较寄存器)		<b>基地址: 0x4000E000</b>						
		<b>偏移地址: 34H</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>	DFx15	DFx14	DFx13	DFx12	DFx11	DFx10	DFx9	DFx8
<b>Write:</b>	0	0	0	0	0	0	0	0
<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>	DFx7	DFx6	DFx5	DFx4	DFx3	DFx2	DFx1	DFx0
<b>Write:</b>	0	0	0	0	0	0	0	0
<b>Reset:</b>	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 比较寄存器)

<b>ADC1CMP</b>	<b>基地址: 0x4000E000</b>
----------------	------------------------

(ADC1 测量比较寄存器)			偏移地址: 38H					
	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>								
<b>Reset:</b>	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。

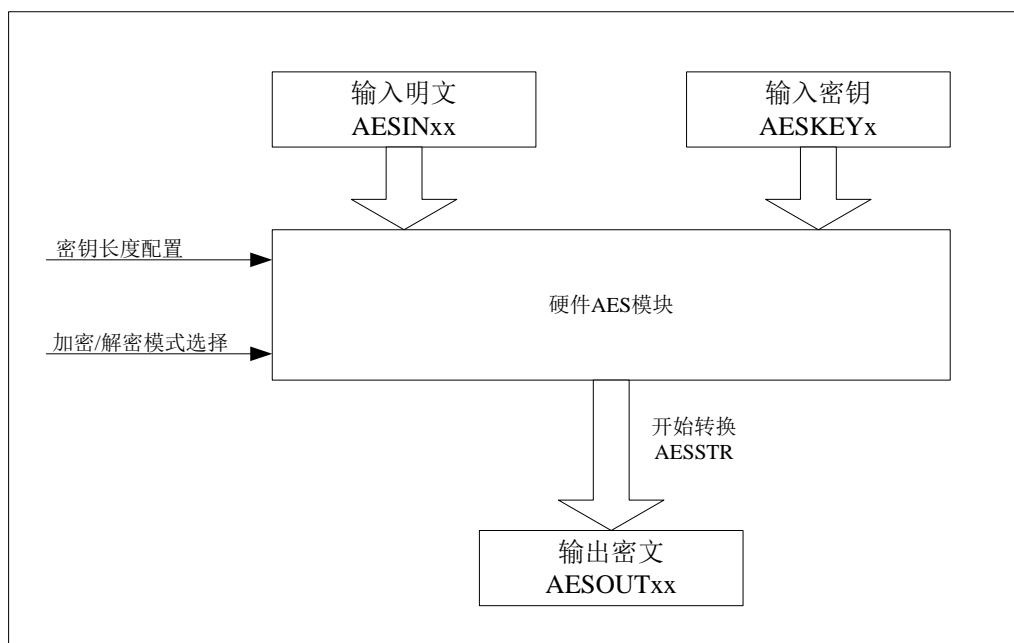
## 18 AES&Ghash&RAND 模块

### 18.1 AES&Ghash&RAND 概述

HT502X 内部集成了 AES 硬件加密/解密硬件加速器模块，符合 FIPS197 标准，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 EBC/CBC/CTR/GCM/GMAC 多种不同的加密/解密模式的时间。

HT502X 内部还集成了随机数发生器模块，一次产生 32 bits 随机数，可用于 ECC 加密中随机数的产生。

### 18.2 框图



### 18.3 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	0x0000	AES配置寄存器
04H	AESSTR	W	0x0000	AES启动命令寄存器
08H	AESFLG	R	0x0000	AES标志寄存器
0CH	AESINLL	R/W	0x00000000	AES输入待加/解密数据
10H	AESINML	R/W	0x00000000	AES输入待加/解密数据
14H	AESINHM	R/W	0x00000000	AES输入待加/解密数据
18H	AESINHH	R/W	0x00000000	AES输入待加/解密数据
1CH	AESOUTLL	R/W	0x00000000	AES输出已加/解密数据

20H	AESOUTML	R/W	0x00000000	AES输出已加/解密数据
24H	AESOUTHM	R/W	0x00000000	AES输出已加/解密数据
28H	AESOUTHH	R/W	0x00000000	AES输出已加/解密数据
2CH	AESKEY0	R/W	0x00000000	AES密钥寄存器
30H	AESKEY1	R/W	0x00000000	AES密钥寄存器
34H	AESKEY2	R/W	0x00000000	AES密钥寄存器
38H	AESKEY3	R/W	0x00000000	AES密钥寄存器
3CH	AESKEY4	R/W	0x00000000	AES密钥寄存器
40H	AESKEY5	R/W	0x00000000	AES密钥寄存器
44H	AESKEY6	R/W	0x00000000	AES密钥寄存器
48H	AESKEY7	R/W	0x00000000	AES密钥寄存器

**RAND 模块寄存器基地址：0x40012080**

偏移地址	名称	读写方式	复位值	功能描述
00H	RANDSTR	R/W	0x0000	真随机数启动命令寄存器
04H	RANDDAT	R/W	0x00000000	真随机数数据寄存器

**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	0x0000	GHASH启动命令寄存器
04H	GHASHFLG	R	0x0000	GHASH标志寄存器
08H	INPUT1LL	R/W	0x00000000	GHASH输入数据1寄存器
0CH	INPUT1ML	R/W	0x00000000	GHASH输入数据1寄存器
10H	INPUT1HM	R/W	0x00000000	GHASH输入数据1寄存器
14H	INPUT1HH	R/W	0x00000000	GHASH输入数据1寄存器
18H	INPUT2LL	R/W	0x00000000	GHASH输入数据2寄存器
1CH	INPUT2ML	R/W	0x00000000	GHASH输入数据2寄存器
20H	INPUT2HM	R/W	0x00000000	GHASH输入数据2寄存器
24H	INPUT2HH	R/W	0x00000000	GHASH输入数据2寄存器
28H	OUTPUTLL	R/W	0x00000000	GHASH输出数据寄存器
2CH	OUTPUTML	R/W	0x00000000	GHASH输出数据寄存器
30H	OUTPUTHM	R/W	0x00000000	GHASH输出数据寄存器
34H	OUTPUTHH	R/W	0x00000000	GHASH输出数据寄存器
38H	AESGHASHIE	R/W	0x0000	AES,Ghash中断使能寄存器
3CH	AESGHASHIF	R/W	0x0000	AES,Ghash中断标志寄存器

## 18.4 特殊功能寄存器说明

### AESCON(AES 配置寄存器)

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_MODE[1:0]		MODE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
KEY_MODE[1:0]	<b>密钥模式选择:</b> 00: AES-128, 密钥为 128bits(KEY0—KEY3)KEY0 为高位, KEY3 为低位 01: AES-192, 密钥为 192bits(KEY0—KEY5)KEY0 为高位, KEY5 为低位 10: AES-256, 密钥为 256bits(KEY0—KEY7)KEY0 为高位, KEY7 为低位 11: 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)

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)

<b>AESFLG</b> (AES 标志寄存器)		<b>基地址: 0x40012000</b> <b>偏移地址: 08H</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>								
<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	X	BUSY
<b>Write:</b>								X
<b>Reset:</b>	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>AESINLL</b> (AES 数据输入中低位)		<b>基地址: 0x40012000</b> <b>偏移地址: 0CH</b>						
	<b>Bit31...Bit0</b>							
<b>Read:</b>	DAT[31:0]							
<b>Write:</b>								
<b>Reset:</b>	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>AESINML</b>	<b>基地址: 0x40012000</b>
----------------	------------------------

<b>(AES 数据输入中低位)</b>		<b>偏移地址: 18H</b>						
	<b>Bit31...Bit0</b>							
<b>Read:</b>	DAT[31:0]							
<b>Write:</b>								
<b>Reset:</b>	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>AESINHM</b> <b>(AES 数据输入中高位)</b>		<b>基地址: 0x40012000</b>						
	<b>偏移地址: 10H</b>							
	<b>Bit31...Bit0</b>							
<b>Read:</b>	DAT[31:0]							
<b>Write:</b>								
<b>Reset:</b>	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

### AESINHH(AES 数据输入高位)

<b>AESINHH</b> <b>(AES 数据输入高位)</b>		<b>基地址: 0x40012000</b>						
	<b>偏移地址: 14H</b>							
	<b>Bit31...Bit0</b>							
<b>Read:</b>	DAT[31:0]							
<b>Write:</b>								
<b>Reset:</b>	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>AESOUTLL</b> <b>(AES 数据输出低位)</b>		<b>基地址: 0x40012000</b>						
	<b>偏移地址: 1CH</b>							

	<b>Bit31...Bit0</b>							
<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 输出已加密/解密数据 128bit 的 bit0-bit31

### AESOUTML(AES 数据输出中低位)

<b>AESOUTML</b> (AES 数据输出中低位)	<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 20H
	<b>Bit31...Bit0</b>
<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 输出已加密/解密数据 128bit 的 bit32-bit63

### AESOUTHM(AES 数据输出中高位)

<b>AESOUTHM</b> (AES 数据输出中高位)	<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 24H
	<b>Bit31...Bit0</b>
<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 输出已加密/解密数据 128bit 的 bit64-bit95

### AESOUTHH(AES 数据输出高位)

<b>AESOUTHH</b> (AES 数据输出高位)	<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 28H
	<b>Bit31...Bit0</b>

<b>Read:</b>	DAT[31:0]							
<b>Write:</b>								
<b>Reset:</b>	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>AESKEY0</b> (AES 密钥 0)	<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 2CH							
	<b>Bit31...Bit0</b>							
<b>Read:</b>	DAT[31:0]							
<b>Write:</b>								
<b>Reset:</b>	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>AESKEY1</b> (AES 密钥 1)	<b>基地址:</b> 0x40012000 <b>偏移地址:</b> 30H							
	<b>Bit31...Bit0</b>							
<b>Read:</b>	DAT[31:0]							
<b>Write:</b>								
<b>Reset:</b>	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

**AESKEY2(AES 密钥 2)**

<b>AESKEY2</b> (AES 密钥 2)		<b>基地址: 0x40012000</b>						
		<b>偏移地址: 34H</b>						
	<b>Bit31...Bit0</b>							
<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

**AESKEY3(AES 密钥 3)**

<b>AESKEY3</b> (AES 密钥 3)		<b>基地址: 0x40012000</b>						
		<b>偏移地址: 38H</b>						
	<b>Bit31...Bit0</b>							
<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

**AESKEY4(AES 密钥 4)**

<b>AESKEY4</b> (AES 密钥 4)		<b>基地址: 0x40012000</b>						
		<b>偏移地址: 3CH</b>						
	<b>Bit31...Bit0</b>							
<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>AESKEY5</b> (AES 密钥 5)	<b>基地址:</b>	<b>0x40012000</b>						
	<b>偏移地址:</b>	<b>40H</b>						
	<b>Bit31...Bit0</b>							
<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 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>AESKEY6</b> (AES 密钥 6)	<b>基地址:</b>	<b>0x40012000</b>						
	<b>偏移地址:</b>	<b>44H</b>						
	<b>Bit31...Bit0</b>							
<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 256bit 的 bit32-bit63, 该寄存器可以写入也可以读出

### AESKEY7(AES 密钥 7)

<b>AESKEY7</b> (AES 密钥 7)	<b>基地址:</b>	<b>0x40012000</b>						
	<b>偏移地址:</b>	<b>48H</b>						
	<b>Bit31...Bit0</b>							
<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 256bit 的 bit0-bit31, 该寄存器可以写入也可以读出

### RANDSTR(随机数启动控制寄存器)

RANDSTR (随机数启动控制寄存器)		基地址: 0x40012080 偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	RANDE	X	X	X	X	X	X	X
Write:	N							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	RESER	Start	X	X
Write:					VED			
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RANDEN	随机数模块使能位 0: 随机数模块关闭 1: 随机数模块使能 当 RANDEN 使能之后, 对 Start 写 1 才有效
RESERVED	内部状态位, 写操作无效, 读取值请忽略
Start	启动控制位 该寄存器位写 1 则启动真随机数生成器, 该控制位在随机数生成器生成随机数后由硬件清 0

### RANDDAT(随机数数据寄存器)

RANDDAT (随机数数据寄存器)		基地址: 0x40012080 偏移地址: 04H						
	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]	真随机数数据寄存器, 共 32 位

### GHASHSTR(GHASH 启动寄存器)

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

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:	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	X	BUSY
Write:	X	X	X	X	X	X	X	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)

INPUT1LL (数据 1 输入低位)		基地址: 0x40012100 偏移地址: 08H					
		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]	Ghash 输入计算数据 1 的 128bit 的 bit0-bit31

### INPUT1ML(数据 1 输入中低位)

<b>INPUT1ML</b> (数据 1 输入中低位)	<b>基地址:</b> 0x40012100
	<b>偏移地址:</b> 0CH
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	DAT[31:0]
<b>Reset:</b>	0 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>INPUT1MH</b> (数据 1 输入中高位)	<b>基地址:</b> 0x40012100
	<b>偏移地址:</b> 10H
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	DAT[31:0]
<b>Reset:</b>	0 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>INPUT1HH</b> (数据 1 输入高位)	<b>基地址:</b> 0x40012100
	<b>偏移地址:</b> 14H
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]

<b>Write:</b>								
<b>Reset:</b>	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>INPUT2LL</b> (数据 2 输入低位)	<b>基地址:</b> 0x40012100 <b>偏移地址:</b> 18H
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	DAT[31:0]
<b>Reset:</b>	0 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>INPUT2ML</b> (数据 2 输入中低位)	<b>基地址:</b> 0x40012100 <b>偏移地址:</b> 1CH
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	DAT[31:0]
<b>Reset:</b>	0 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>INPUT2MH</b> (数据 2 输入中高位)	<b>基地址:</b> 0x40012100 <b>偏移地址:</b> 20H
	<b>Bit31...Bit0</b>
<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]	Ghash 输入计算数据 1 的 128bit 的 bit64-bit95

### INPUT2HH(数据 2 输入高位)

<b>INPUT2HH</b> (数据 2 输入高位)	<b>基地址: 0x40012100</b> <b>偏移地址: 24H</b>
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	DAT[31:0]
<b>Reset:</b>	0 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>OUTPUTLL</b> (数据输出低位)	<b>基地址: 0x40012100</b> <b>偏移地址: 28H</b>
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	DAT[31:0]
<b>Reset:</b>	0 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>OUTPUTML</b> (数据输出中低位)	<b>基地址: 0x40012100</b> <b>偏移地址: 2CH</b>
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	DAT[31:0]
<b>Reset:</b>	0 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)

<b>OUTPUTMH</b> (数据输出中高位)		<b>基地址: 0x40012100</b>						
		<b>偏移地址: 30H</b>						
		<b>Bit31...Bit0</b>						
<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]	Ghash 输入计算数据 1 的 128bit 的 bit64-bit95

### OUTPUTHH(数据输出高位)

<b>OUTPUTHH</b> (数据输出高位)		<b>基地址: 0x40012100</b>						
		<b>偏移地址: 34H</b>						
		<b>Bit31...Bit0</b>						
<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]	Ghash 输入计算数据 1 的 128bit 的 bit96-bit127

### AESGHASHIE(AESGHASH 中断使能寄存器)

<b>AESGHASHIE</b> (AESGHASH 中断使能)		<b>基地址: 0x40012100</b>						
		<b>偏移地址: 38H</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	X	RANDIE	GHASHI E	AESIE
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
RANDIE	随机数中断使能
GHASHIE	GHASH 计算中断使能
AESIE	AES 计算中断使能

**AESGHASHIF(AESGHASH 中断标志寄存器)**

<b>AESGHASHIF</b> (AESGHASH 中断标志)			<b>基地址: 0x40012100</b> <b>偏移地址: 3CH</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>								
<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	RANDIF	GHASHI F	AESIF
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
RANDIF	随机数中断标志
GHASHIF	中断标志
AESIF	AES 计算中断标志

## 19 ECC256 模块

### 19.1 概述

HT502X 的 ECC256 模块为椭圆曲线(EC, Elliptic Curve)运算加速模块, 可显著提高基于椭圆曲线的加密协议的实现效率。本模块支持的加密协议或算法包括 ECDSA(EC Digital Signature Algorithm), ECDH (EC Deffie-Hellman) 及相关变种协议。

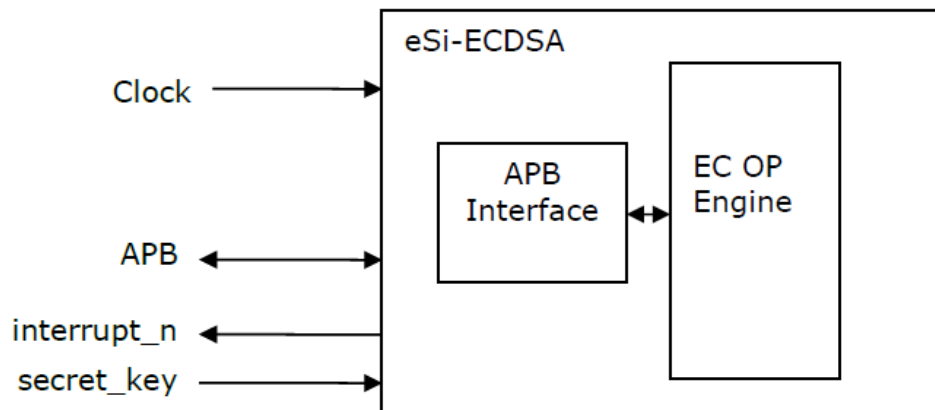
### 19.2 功能描述

- 支持基于简化 Weierstrass 方程 ( $y^2=x^3+ax+b \bmod p$ ) 的所有素域 GF(p)上椭圆曲线, 此椭圆曲线在 NIST, SEC2, Brainpool 等协议中被定义为常用椭圆曲线
- 支持 ECDSA 签名/认证运算
- 支持公钥是否在椭圆曲线上验证运算
- 支持椭圆曲线点运算
  - 点乘运算 (ECSM, EC Scalar Multiplication)
  - 点加运算 (ECA, EC Addition)
  - 倍加运算 (ECD, EC Doubling)
- 支持大数模运算
  - 模加运算 (MA, Modular Addition)
  - 模减运算 (MS, Modular Subtraction)
  - 模乘运算 (MM, Modular Multiplication)
  - 模除运算 (MD, Modular Division)
  - 模逆运算 (MI, Modular Inversion)
- 支持 ECC256, 且向下兼容 ECC244 和 ECC192
- 可阻挡 STA 攻击 (both Simple and Statistical Timing side channel Attacks) 和 SPA 攻击 (Simple Power Analysis attacks)

Clkctrl10 中的 AES\_EN 为 AES/RAND 共用使能控制位, Clkctrl10 新增控制位 ECC\_EN 独立控制 ECC 模块, 位置 bit 16 (Clkctrl10.<16>), 应用 ECC 模块前需使能此控制位。每次关闭再使能 ECC\_EN, ECC 模块会被复位。

Secret key 与内部 256-bits 寄存器相连, 用户将私钥存储于 Flash 地址 0x0003FC00~0x0003FC1F 中; 用户可配置 ECCCON.6<SE> 选择 Secretkey 或 KEYREG 作为私钥。每次自加载时, 则 0x0003FC00~0003FC1F 中的数据被加载到内部 256-bits 寄存器中, 低位存于 0x0003FC00, 高位存于 0003FC1F; 内部 256-bits 寄存器不开放, 用户无法操作。

### 19.3 ECC 运算加速模块



ECC 各运算寄存器配置:

加速模块输入/输出			
	运算模式	输入参数寄存器	输出结果寄存器
ECDSA 运算	签名 运算	PREG: 大素数 p AREG: 椭圆曲线参数 a PXREG: 椭圆曲线基点 x 坐标 PYREG: 椭圆曲线基点 y 坐标 SXREG: 基点的阶参数 KEYREG: 私钥 k SYREG: 随机大整数 MREG: HASH 后的消息摘要	RXREG: 签名结果 r 参数 RYREG: 签名结果 s 参数 ECCSTA: ECC 状态寄存器 [3:3]ECDSA_S = 0(若为 1, 重新获取随机大整数, 重新运算) [0:0]ECCFLG = 1
	认证 运算	PREG: 大素数 p AREG: 椭圆曲线参数 a PXREG: 椭圆曲线基点 x 坐标 PYREG: 椭圆曲线基点 y 坐标 SXREG: 基点的阶参数 KEYREG: 公钥 x 坐标 SYREG: 公钥 y 坐标 MREG: HASH 后的消息摘要 RXREG: 签名结果 r 参数 RYREG: 签名结果 s 参数	ECCSTA: ECC 状态寄存器 [0:0]ECCFLG = 1 [2:2]ECDSA_V = 1(认证通过) = 0(认证失败)
点在曲 线上验 证	公钥 验证	PREG: 大素数 p AREG: 椭圆曲线参数 a SYREG: 椭圆曲线参数 b PXREG: 公钥 x 坐标 PYREG: 公钥 y 坐标	ECCSTA: ECC 状态寄存器 [0:0]ECCFLG = 1 [4:4]PKV = 1(验证通过) = 0(验证失败)
ECC 点	点加运算	$P(X1,Y1) + S(X2,Y2) = R(X3,Y3)$	

运 算		PREG: 大素数 p AREG: 椭圆曲线参数 a PXREG: 椭圆曲线第 1 点 x 坐标 PYREG: 椭圆曲线第 1 点 y 坐标 SXREG: 椭圆曲线第 2 点 x 坐标 SYREG: 椭圆曲线第 2 点 y 坐标	RXREG: 点加结果 x 坐标 RYREG: 点加结果 y 坐标 (结果点是在椭圆曲线上) ECCSTA: ECC 状态寄存器 <b>[0:0]ECCFLG = 1</b>
	倍加运算	<b><math>2 * P(X1, Y1) = R(X3, Y3)</math></b> PREG: 大素数 p AREG: 椭圆曲线参数 a PXREG: 椭圆曲线第 1 点 x 坐标 PYREG: 椭圆曲线第 1 点 y 坐标	RXREG: 点加结果 x 坐标 RYREG: 点倍加结果 y 坐标 (结果点是在椭圆曲线上) ECCSTA: ECC 状态寄存器 <b>[0:0]ECCFLG = 1</b>
	点乘运算	<b><math>k * P(X1, Y1) = R(X3, Y3)</math></b> PREG: 大素数 p AREG: 椭圆曲线参数 a PXREG: 椭圆曲线点 x 坐标 PYREG: 椭圆曲线点 y 坐标 KEYREG: 标量系数 k	RXREG: 点加结果 x 坐标 RYREG: 点乘结果 y 坐标 (结果点是在椭圆曲线上) ECCSTA: ECC 状态寄存器 <b>[0:0]ECCFLG = 1</b>
标 量 模 运 算	模加运算	<b><math>PX + PY \pmod n = RX</math></b> PREG: 模运算参数 n PXREG: 第 1 个标量 PYREG: 第 2 个标量	RXREG: 模加结果标量 ECCSTA: ECC 状态寄存器 <b>[0:0]ECCFLG = 1</b>
	模减运算	<b><math>PX - PY \pmod n = RX</math></b> PREG: 模运算参数 n PXREG: 第 1 个标量 PYREG: 第 2 个标量	RXREG: 模减结果标量 ECCSTA: ECC 状态寄存器 <b>[0:0]ECCFLG = 1</b>
	模乘运算	<b><math>PX * PY \pmod n = RX</math></b> PREG: 模运算参数 n PXREG: 第 1 个标量 PYREG: 第 2 个标量	RXREG: 模乘结果标量 ECCSTA: ECC 状态寄存器 <b>[0:0]ECCFLG = 1</b>
	模除运算	<b><math>PY / PX \pmod n = RX</math></b> PREG: 模运算参数 n PXREG: 第 1 个标量 PYREG: 第 2 个标量	RXREG: 模除结果标量 ECCSTA: ECC 状态寄存器 <b>[0:0]ECCFLG = 1</b>
	模逆	<b><math>PX^{-1} \pmod n = RX</math></b>	

	运算	PREG: 模运算参数 n PXREG: 第 1 个标量	RXREG: 模逆结果标量 ECCSTA: ECC 状态寄存器 [0:0]ECCFLG = 1
--	----	---------------------------------	---

## 19.4 特殊功能寄存器列表

ECC 模块寄存器基地址: 0x40017000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	ECCCON	R/W	0x0000	ECC 模块控制寄存器
0x04	ECCSTA	R/W	0x0000	ECC 模块状态寄存器
0x08	PXREG	R/W	0x0000	基点 x 坐标寄存器
0x0C	PYREG	R/W	0x0000	基点 y 坐标寄存器
0x10	KEYREG	R/W	0x0000	私钥寄存器
0x14	AREG	R/W	0x0000	椭圆曲线参数 a
0x18	PREG	R/W	0x0000	大素数 p
0x1C	RXREG	R/W	0x0000	ECC 运算输出点 x 坐标
0x20	RYREG	R/W	0x0000	ECC 运算输出点 y 坐标
0x24	SXREG	R/W	0x0000	ECC 运算第二操作数 x 坐标
0x28	SYREG	R/W	0x0000	ECC 运算第二操作数 y 坐标
0x2C	MREG	R/W	0x0000	Hash 算法消息摘要 (for ECDSA)

## 19.5 特殊功能寄存器说明

### ECCCON(ECC 模块控制寄存器)

ECCCON (ECC 模块控制寄存器)			基地址: 0x40017000 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	SE	ECCIE	OP_SEL	OP_SEL	OP_SEL	OP_SEL	OP_STR
Write:				3	2	1	0	
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SE	私钥选择控制位

	1: 强制 Secret Key 作为私钥, 代替 KEYREG 寄存器中的值 0: KEYREG 寄存器值作为私钥
ECCIE	<b>ECC 中断使能控制 (中断向量与 AES 共用)</b> 1: 使能 ECC 中断 0: 关闭 ECC 中断
OP_SEL[3:0]	<b>运算模式选择位</b> 0000: 点乘运算 0001: 点加运算 0010: 倍加运算 0011: 模加运算 0100: 模减运算 0101: 模乘运算 0110: 模除运算 0111: 模逆运算 1000: ESDA 签名运算 1001: ESDA 认证运算 1010: 公钥曲线上验证 其他: 保留
OP_STR	<b>开始运算:</b> 1: 开始已设置的运算, 运算结束硬件清零 0: 运算完成

## ECCSTA(ECC 模块状态寄存器)

ECCSTA (ECC 模块状态寄存器)			基地址: 0x40017000 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	PKV	ECDSA_ S	ECDSA_ V	BUSY	ECCFLG
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PKV	公钥曲线上验证标志 0: 验证失败 1: 验证通过 此位需 ECCFLG=1, 即运算完成之后, 读取有效
ECDSA_S	ECDSA 签名重试标志

	0: 签名完成 1: 满足重试条件 此位需 ECCFLG=1, 即运算完成之后, 读取有效
ECDSA_V	<b>ECDSA 认证标志</b> 0: 认证失败 1: 认证通过 此位需 ECCFLG=1, 即运算完成之后, 读取有效
BUSY	<b>计算模块状态位</b> 0: 模块空闲 1: 模块忙碌
ECCFLG	<b>ECC 运算完成标志位</b> 0: 未完成 1: 已完成 Note: 写 0 清零此标志

## PXREG(基点 x 坐标寄存器)

PXREG (基点 x 坐标寄存器)	基地址: <b>0x40017000</b> 偏移地址: <b>08H</b>
	<b>Bit31...Bit0</b>
<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]	<b>基点 X 坐标数据入口 (4 bytes 分段)</b> <b>写入:</b> 位宽为 256bit 的数据, 写入 8 次寄存器, 从最低 4 个有效 bytes 开始 (自右向左写入, 寄存器 bit0 对应数据低位) 若实际位宽小于 256bit, 高位补零 <b>读取:</b> 4bytes 分段, 读取 8 次寄存器, 得到数据, 首先读取的为最高 4 个有效 bytes (自左向右读取, 寄存器 bit32 为数据高位, bit0 为低位) 若实际位宽小于 256bit, 读取的数据高位为 0

## PYREG(基点 y 坐标寄存器)

PYREG (基点 y 坐标寄存器)	基地址: <b>0x40017000</b> 偏移地址: <b>0CH</b>
	<b>Bit31...Bit0</b>

<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]	<p><b>基点 Y 坐标数据入口 (4 bytes 分段)</b></p> <p><b>写入:</b> 位宽为 256bit 的数据, 写入 8 次寄存器, 从最低 4 个有效 bytes 开始 (自右向左写入, 寄存器 bit0 对应数据低位) 若实际位宽小于 256bit, 高位补零</p> <p><b>读取:</b> 4bytes 分段, 读取 8 次寄存器, 得到数据, 首先读取的为最高 4 个有效 bytes (自左向右读取, 寄存器 bit32 为数据高位, bit0 为低位) 若实际位宽小于 256bit, 读取的数据高位为 0</p>

## KEYREG(私钥寄存器)

KEYREG (私钥寄存器)	<b>基地址:</b> 0x40017000
	<b>偏移地址:</b> 10H
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	DAT[31:0]
<b>Reset:</b>	0 0 0 0 0 0 0 0 0

位	功能描述
DAT[31:0]	<p><b>私钥寄存器数据入口 (4 bytes 分段)</b></p> <p><b>写入:</b> 位宽为 256bit 的数据, 写入 8 次寄存器, 从最低 4 个有效 bytes 开始 (自右向左写入, 寄存器 bit0 对应数据低位) 若实际位宽小于 256bit, 高位补零</p> <p><b>读取:</b> 4bytes 分段, 读取 8 次寄存器, 得到数据, 首先读取的为最高 4 个有效 bytes (自左向右读取, 寄存器 bit32 为数据高位, bit0 为低位) 若实际位宽小于 256bit, 读取的数据高位为 0</p>

## AREG(椭圆曲线参数 a)

AREG (椭圆曲线参数 a)	<b>基地址:</b> 0x40017000
	<b>偏移地址:</b> 14H
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]

<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	<p><b>椭圆曲线参数 a 数据入口 (4 bytes 分段)</b></p> <p><b>写入:</b> 位宽为 256bit 的数据, 写入 8 次寄存器, 从最低 4 个有效 bytes 开始 (自右向左写入, 寄存器 bit0 对应数据低位) 若实际位宽小于 256bit, 高位补零</p> <p><b>读取:</b> 4bytes 分段, 读取 8 次寄存器, 得到数据, 首先读取的为最高 4 个有效 bytes (自左向右读取, 寄存器 bit32 为数据高位, bit0 为低位) 若实际位宽小于 256bit, 读取的数据高位为 0</p>

### PREG(大参数 p)

PREG (大参数 p)	<b>基地址: 0x40017000</b>
	<b>偏移地址: 18H</b>
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	
<b>Reset:</b>	0 0 0 0 0 0 0 0 0

位	功能描述
DAT[31:0]	<p><b>大参数 p 数据入口 (4 bytes 分段)</b></p> <p><b>写入:</b> 位宽为 256bit 的数据, 写入 8 次寄存器, 从最低 4 个有效 bytes 开始 (自右向左写入, 寄存器 bit0 对应数据低位) 若实际位宽小于 256bit, 高位补零</p> <p><b>读取:</b> 4bytes 分段, 读取 8 次寄存器, 得到数据, 首先读取的为最高 4 个有效 bytes (自左向右读取, 寄存器 bit32 为数据高位, bit0 为低位) 若实际位宽小于 256bit, 读取的数据高位为 0</p>

### RXREG(ECC 运算输出点 x 坐标)

RXREG (ECC 运算输出点 x 坐标)	<b>基地址: 0x40017000</b>
	<b>偏移地址: 1CH</b>
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	

<b>Reset:</b>	0	0	0	0	0	0	0	0
---------------	---	---	---	---	---	---	---	---

位	功能描述
DAT[31:0]	<p><b>ECC 运算输出点 x 坐标数据入口（4 bytes 分段）</b></p> <p><b>写入：</b> 位宽为 256bit 的数据，写入 8 次寄存器，从最低 4 个有效 bytes 开始（自右向左写入，寄存器 bit0 对应数据低位）若实际位宽小于 256bit，高位补零</p> <p><b>读取：</b> 4bytes 分段，读取 8 次寄存器，得到数据，首先读取的为最高 4 个有效 bytes（自左向右读取，寄存器 bit32 为数据高位，bit0 为低位） 若实际位宽小于 256bit，读取的数据高位为 0</p>

### RYREG(ECC 运算输出点 y 坐标)

<b>RYREG</b> (ECC 运算输出点 y 坐标)	<b>基地址： 0x40017000</b> <b>偏移地址： 20H</b>
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	DAT[31:0]
<b>Reset:</b>	0 0 0 0 0 0 0 0 0

位	功能描述
DAT[31:0]	<p><b>ECC 运算输出点 y 坐标数据入口（4 bytes 分段）</b></p> <p><b>写入：</b> 位宽为 256bit 的数据，写入 8 次寄存器，从最低 4 个有效 bytes 开始（自右向左写入，寄存器 bit0 对应数据低位）若实际位宽小于 256bit，高位补零</p> <p><b>读取：</b> 4bytes 分段，读取 8 次寄存器，得到数据，首先读取的为最高 4 个有效 bytes（自左向右读取，寄存器 bit32 为数据高位，bit0 为低位） 若实际位宽小于 256bit，读取的数据高位为 0</p>

### SXREG(ECC 运算第二操作数 x 坐标)

<b>SXREG</b> (ECC 运算第二操作数 x 坐标)	<b>基地址： 0x40017000</b> <b>偏移地址： 24H</b>
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	DAT[31:0]
<b>Reset:</b>	0 0 0 0 0 0 0 0 0

位	功能描述
DAT[31:0]	<b>ECC 运算第二操作数 x 坐标数据入口 (4 bytes 分段)</b> <b>写入:</b> 位宽为 256bit 的数据, 写入 8 次寄存器, 从最低 4 个有效 bytes 开始 (自右向左写入, 寄存器 bit0 对应数据低位) 若实际位宽小于 256bit, 高位补零 <b>读取:</b> 4bytes 分段, 读取 8 次寄存器, 得到数据, 首先读取的为最高 4 个有效 bytes (自左向右读取, 寄存器 bit32 为数据高位, bit0 为低位) 若实际位宽小于 256bit, 读取的数据高位为 0

### SYREG(ECC 运算第二操作数 y 坐标)

SYREG (ECC 运算第二操作数 y 坐标)	<b>基地址: 0x40017000</b> <b>偏移地址: 28H</b>
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	
<b>Reset:</b>	0    0    0    0    0    0    0    0

位	功能描述
DAT[31:0]	<b>ECC 运算第二操作数 y 坐标数据入口 (4 bytes 分段)</b> <b>写入:</b> 位宽为 256bit 的数据, 写入 8 次寄存器, 从最低 4 个有效 bytes 开始 (自右向左写入, 寄存器 bit0 对应数据低位) 若实际位宽小于 256bit, 高位补零 <b>读取:</b> 4bytes 分段, 读取 8 次寄存器, 得到数据, 首先读取的为最高 4 个有效 bytes (自左向右读取, 寄存器 bit32 为数据高位, bit0 为低位) 若实际位宽小于 256bit, 读取的数据高位为 0

### MREG(Hash 算法消息摘要)

MREG (Hash 算法消息摘要)	<b>基地址: 0x40017000</b> <b>偏移地址: 2CH</b>
	<b>Bit31...Bit0</b>
<b>Read:</b>	DAT[31:0]
<b>Write:</b>	
<b>Reset:</b>	0    0    0    0    0    0    0    0

位	功能描述
DAT[31:0]	<b>Hash 算法消息摘要数据入口（4 bytes 分段）</b> <b>写入：</b> 位宽为 256bit 的数据，写入 8 次寄存器，从最低 4 个有效 bytes 开始（自右向左写入，寄存器 bit0 对应数据低位）若实际位宽小于 256bit，高位补零 <b>读取：</b> 4bytes 分段，读取 8 次寄存器，得到数据，首先读取的为最高 4 个有效 bytes（自左向右读取，寄存器 bit32 为数据高位，bit0 为低位） 若实际位宽小于 256bit，读取的数据高位为 0

注：传送信息经 HASH 算法之后，获得的信息摘要最多 256bit 可参与 ECDSA 运算，将信息摘要输入此寄存器作为参与 ECDSA 运算的参数。

## 20 DMA 功能

### 20.1 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。

### 20.2 功能描述

- DMA 模块共有 3 个独立的可配置通道
- 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
- DMA 与 CPU 之间交替占用总线
- DMA 支持最多 32 个外部数据请求源，每个通道还有一个软件请求源
- 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
- 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
- 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
- DMA 支持三种中断：传输完成中断，传输错误中断，块传输完成中断
- 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 CHNx<sub>BULKNUM</sub>[8:15]，达到循环次数后自动关闭相应通道

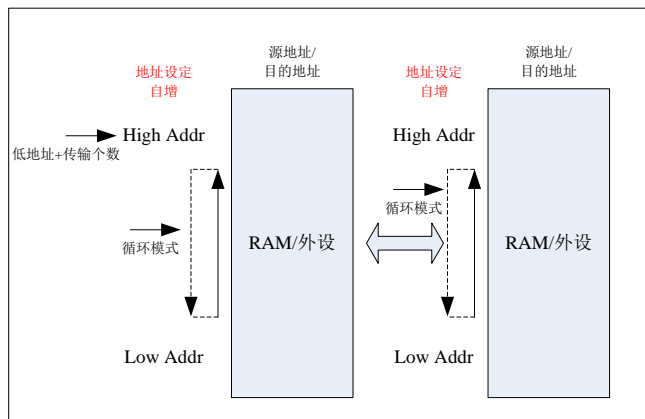
### 20.3 DMA 数据请求源列表

DMA_CTL[8...12]	说明
0	Soft request
1	UART0 发送
2	UART0 接收
3	UART1 发送
4	UART1 接收
5	UART2 发送
6	UART2 接收
7	UART3 发送
8	UART3 接收
9	UART4 发送
10	UART4 接收
11	UART5 发送

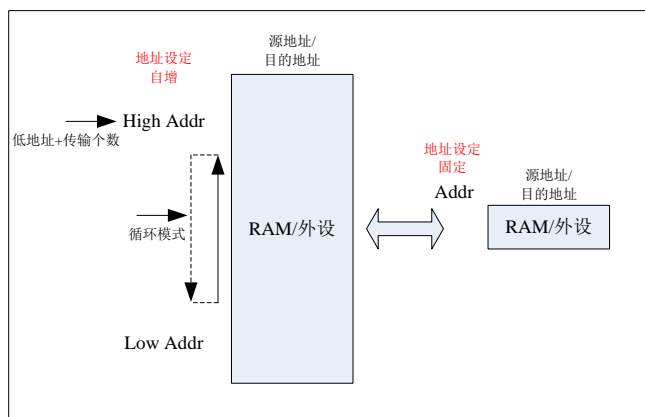
12	UART5 接收
13	ISO7816_0 发送
14	ISO7816_0 接收
15	ISO7816_1 发送
16	ISO7816_1 接收
17	SPI 发送
18	SPI 接收
19	I2C 发送
20	I2C 接收
21	EMU 电压波形数据
22	EMU 电流通路 1 波形数据
23	EMU 电流通路 2 波形数据
24	Timer0
25	Timer1
26	Timer2
27	Timer3
28	保留
29	保留
30	保留
31	保留
32	保留

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。

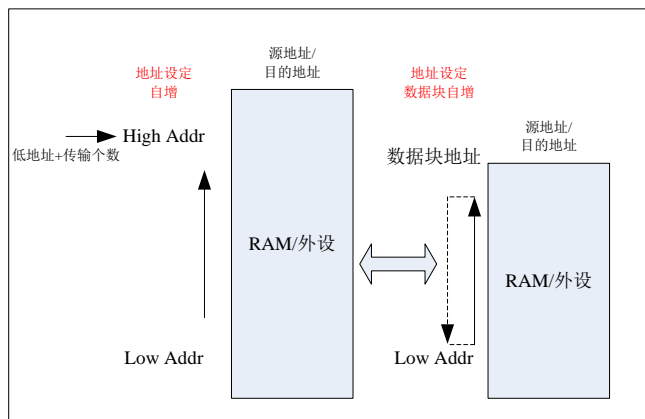
## 20.4DMA 数据传输说明



针对这两种传输方式还可以选择是否使用**块传输**，而块传输和单个传输的区别就是在**一个块传输的过程中不会被打断**



### 数据块自增模式



数据块自增的方式主要是可以实现两块不用大小的数据块之间的数据搬运

## 20.5 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	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

## 20.6 特殊功能寄存器说明

### DMAIE(DMA 中断使能寄存器)

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:	X	X	X	X	X	TEIE2	TEIE1	TEIE0
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	BCIE2	BCIE1	BCIE0	X	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 (仅块传输模式有效)

	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)

<b>DMAIF</b> (DMA 中断标志寄存器)		<b>基地址: 0x40016000</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>	X	X	X	X	X	TEIF2	TEIF1	TEIF0
<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	BCIF2	BCIF1	BCIF0	X	TCIF2	TCIF1	TCIF0
<b>Write:</b>								
<b>Reset:</b>	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]在每次块传输完成时置位, 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)

<b>CHNSTA</b> (DMA 状态寄存器)	<b>基地址: 0x40016000</b> <b>偏移地址: 08H</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>								
<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	BUSY2	BUSY1	BUSY0
<b>Write:</b>						X	X	X
<b>Reset:</b>	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)

<b>CHNCTL</b> (DMA 通道控制寄存器)		基地址: 0x4001600C, 0x40016024, 0x4001603C 偏移地址: 00H						
	<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	Request4	Request3	Request2	Request1	Request0	DESTIN_INC1
<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>	DESTIN_INC0	SOURC_INC1	SOURC_INC0	CYCLE	TMOD	PSIZE1	PSIZE0	DMA_CHNx_EN
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
Request[4...0]	触发数据源选择, 根据 20.3 章节所列 DMA 数据请求源列表来定
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
CYCLE	循环模式配置 0: 不循环模式

	1: 循环模式
TMOD	<b>传输模式配置</b> 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)
PSIZE[1...0]	<b>数据源的传送位数</b> 00: 8 位 01: 16 位 10: 32 位 11: 32bit
DMA_CHNx_EN	<b>DMA 通道使能(x=0\1\2)</b> 0: 禁止 1: 使能

## CHNSRC(DMA 通道源地址寄存器)

<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>					
	<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>	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8
<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>	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[15...0]	数据传输源地址寄存器

## CHNTAR(DMA 通道目的地址寄存器)

<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>					
	<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>	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8
<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>	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[15...0]	数据传输目的地址寄存器

## CHNCNT(DMA 通道传输数量寄存器)

<b>CHNCNT</b> (DMA 通道传输数量寄存器)		基地址: 0x4001600C, 0x40016024, 0x4001603C 偏移地址: 0CH						
	<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>	Num15	Num14	Num13	Num12	Num11	Num10	Num9	Num8
<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>	Num7	Num6	Num5	Num4	Num3	Num2	Num1	Num0
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
Num[15...0]	<b>DMA 数据传输个数设置寄存器</b> 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 通道以传输数据个数)

<b>CHNTCCNT</b> (DMA 通道已传输数据个数)		基地址: 0x4001600C, 0x40016024, 0x4001603C 偏移地址: 10H						
	<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>	Num15	Num14	Num13	Num12	Num11	Num10	Num9	Num8
<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>	Num7	Num6	Num5	Num4	Num3	Num2	Num1	Num0
<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
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)

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：范围 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

## 21 按键扫描模块（KEYSCAN）

### 21.1 概述

KEYSCAN 模块内建 4\*4 键盘自动扫描功能，包含 4 个列线扫描输入线 SCANIN0~3 引脚和 4 个行扫描输出线 SCANOUT0~4 引脚。用户如果需要使用按键扫描功能，则必须把对应引脚配置为 SCANIN<sub>x</sub> 或 SCANOUT<sub>x</sub> 复用功能。

### 21.2 功能描述

- 支持最大 4\*4 按键
- 支持 SLEEP\hold 模式下按键唤醒
- 仅支持下降沿触发
- 仅支持单键检测

若用户想要配置 4\*1 键盘，则（4SCANIN + 1 SCANOUT）或（1 SCANIN + 4 SCANOUT）的引脚配置为相应复用功能即可，如果配置 4\*2 键盘，则（4 SCANIN + 2 SCANOUT）或（2 SCANIN + 4 SCANOUT）的引脚配置为相应复用功能即可。

配置 CLKCTRL0 的 KEY\_EN=1，同时配置相应 I/O 口为 SCANIN/SCANOUT 功能，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 的低电平时间，则按键扫描中断标志置位，查看状态寄存器 KEYSTA，可定位具体的按键位置，若开启中断使能（NVIC\_EnableIRQ(KEY\_IRQn)），则发生按键中断，按键扫描中断号为 27。

被使用的 SCANIN 引脚，上拉电阻必须使能有效；被使用的 SCANOUT 引脚，开漏功能必须关闭。

## 21.3 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

## 21.4 特殊功能寄存器说明

### KEYSTA(KEYSCAN 状态寄存器)

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[15:0]	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)

KEYIF (KEY 标志寄存器)	基地址: 0x40014000 偏移地址: 04H
----------------------	------------------------------

	<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	X	KEYIF
<b>Write:</b>								
<b>Reset:</b>	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

## 22 Cortex-M0 内核简要说明

### 22.1 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。

### 22.2 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。

例如：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*0.1); //带入 Fcpu*timing = 19660800*0.1s
```

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

### 22.3 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

### 22.4 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 IRQn 的中断 (注: 需满足 $IRQn \geq 0$ , 对 $IRQn < 0$ 的中断, 没有中断使能位)
<code>void NVIC_DisableIRQ(IRQn_Type IRQn)</code>	屏蔽中断号为 IRQn 的中断 (注: 需满足 $IRQn \geq 0$ , 对 $IRQn < 0$ 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 IRQn 中断的挂起状态
<code>void NVIC_SetPendingIRQ(IRQn_Type IRQn)</code>	设置中断号为 IRQn 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 IRQn 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 IRQn 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 0x0—0x3)
<code>uint32_t NVIC_GetPriority(IRQn_Type IRQn)</code>	获取中断号为 IRQn 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: ticks 不能超过 0xFFFFFFFF)

注: 中断号 IRQn 详细见 8.1 中断向量说明

## 23 EMU 模块

### 23.1 概述

提供单相电能计量所需要的全部功能，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
- 同时提供两个通道有功功率，两个通道无功功率
- 电压、电流有效值
- 电压频率测量
- 有功功率、无功功率、视在功率波形数据
- 灵活的防窃电方案，窃电阈值通过寄存器可调
- 灵活的潜动与启动方案
- 直流偏置自动校正
- 多种能量计算模式，支持正反向电能单独计量
- 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
- 支持超低功耗灵活的断零线窃电计量与常数计量
- Q 通道能量脉冲可复用为 P2 能量脉冲
- 任意一路 ADC 随频采样波形缓存 BUFFER
- 提供 3 路 ADC 自动/手动温度补偿功能

### 23.2 功能描述

HT502X EMU 内含三路完全独立的二阶  $\Sigma\text{-}\Delta\text{ADC}$  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。

### 23.2.1 模数转换器

HT502X 内置三路完全独立的二阶  $\Sigma\text{-}\Delta$  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 的幅值进行相对于通道 1 的比例缩放，以用在自动防窃电等功能上。

### 23.2.2 ADC 采样输出和功率波形输出

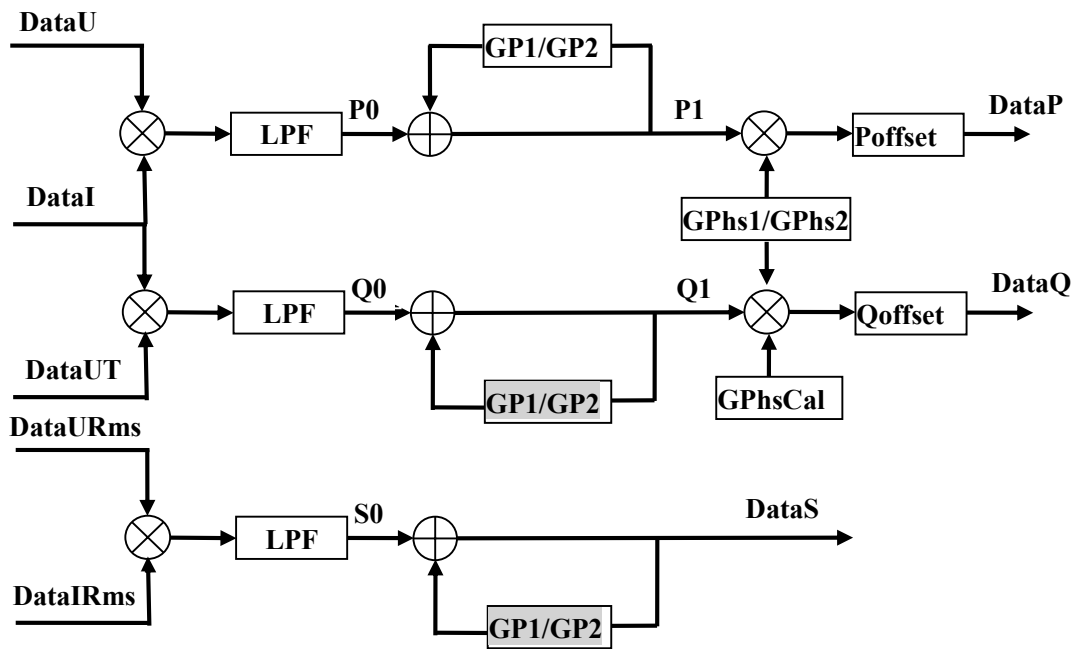
HT502X 将三路 ADC 输出的 22bit 波形数据 [SPLI1\(00H\)](#)、[SPLI2\(04H\)](#)、[SPLU\(08H\)](#) 开放给用户。同时也将作为电能累加的功率波形数据 [SPLP\(0CH\)](#)、[SPLQ\(10H\)](#) 开放给用户，波形采样数据可通过寄存器 [ADCCFG\(50H.WaveSel\)](#) 配置为高通前或高通后。

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。

### 23.2.3 有功功率、无功功率和视在功率

HT502X 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。



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。

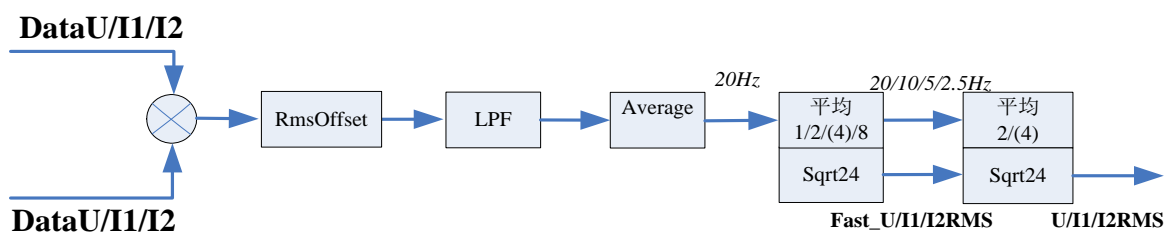
对于通道 1 和通道 2 的功率分别提供增益校正和相位校正，同时针对小信号的精度问题，提供了偏置校正来消除外界干扰。

HT502X 对增益校正做了优化，即 GP 寄存器对 PowerP/Q/S 同时产生作用，而原 GQ/GS 寄存器已无效。

### 23.2.4 有效值

HT502X 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%。



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。

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 ROSICTRL (94H) 的 LpCtrl[1:0]来设置，此时慢速有效值寄存器读出固定为 0。

### 23.2.5 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。

### 23.2.6 起动/潜动

HT502X 提供了灵活的潜动与启动功能，通过设置校表寄存器 P/Q/S Start(40H/44H/48H)，用户可以完成防潜动与起动的功能。用户同时可通过寄存器 EMUCTRL.StartSel (74H.Bit13)选择潜动启动方式。

**StartSel 未使能时 (Default)，启动/潜动根据 PQ 单独方式判断：**

|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。

**StartSel 使能时，启动/潜动根据 P|Q 联合方式判断：**

1) 当|P| 大于 PStart 或|Q|大于 QStart 两条件满足至少一个时，有功脉冲 PF 和 QF 均输出脉冲。相应 NOQLD1、NOPLD1 标志都为 0，或者 NOQLD2、NOPLD2 标志都为 0。

2) 当|P| 小于 PStart 且|Q|小于 QStart 时，PF 和 QF 不输出脉冲，NOQLD1、NOPLD1 标志都为 1，或者 NOQLD2、NOPLD2 标志都为 1。

HT502X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PQSStarScouceSel[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。

### 23.2.7 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。

### 23.2.8 直流偏置校正

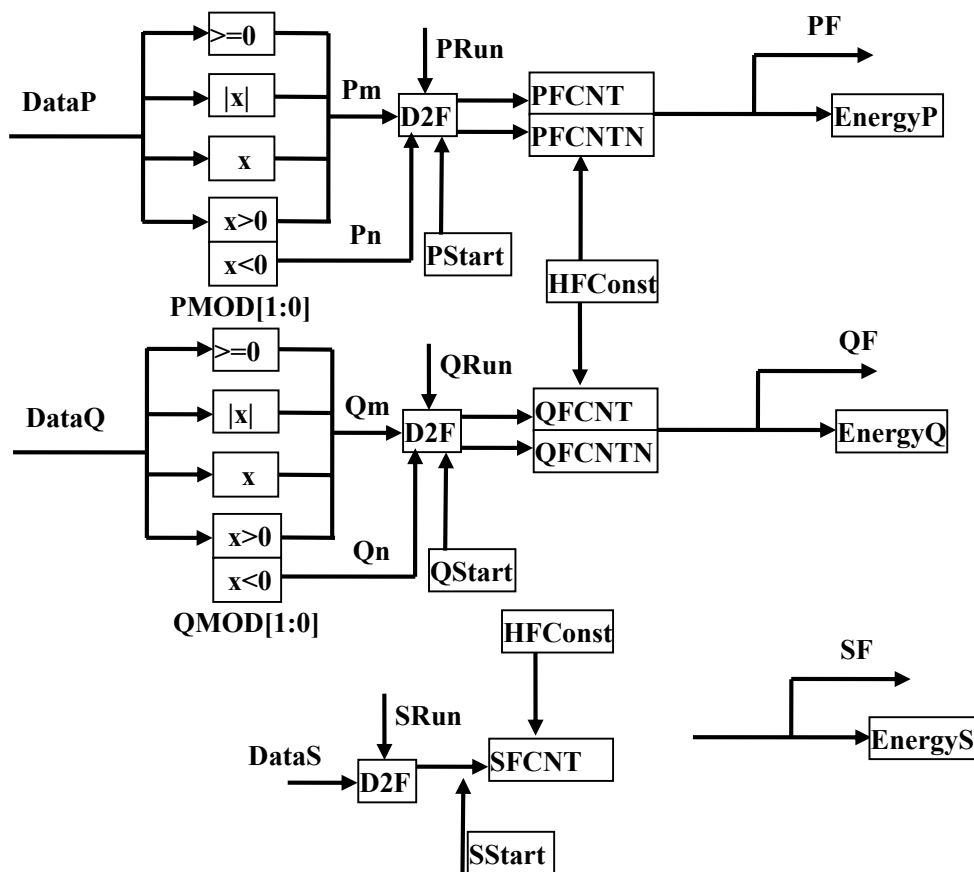
用户通过 CHNLCR(54H)的 HPFONU、HPFONI2、HPFONI1 可以分别控制电压和两路电流的高通滤波器环节是否打开。当这些位为 1 时，表示高通开启，反之高通关闭。

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(04H)、SPL\_U(08H)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。

## 23.2.9 电能脉冲输出

HT502X 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。



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、正反向分别计量四种累加方式。

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)、SFCNT(64H)、PFCNTN(118H)和 QFCNTN(11CH)。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 HFConst(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。

HT502X 开放了快速脉冲计数器 PFCNT/QFCNT/SFCNT/PFCNTN/QFCNTN，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。

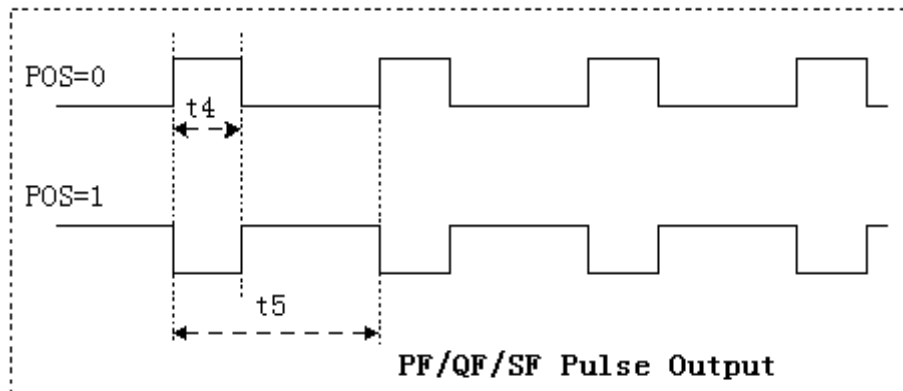
脉冲输出管脚 PF/QF/SF 是否输出和能量寄存器是否累加受到 EMCON(58H)的 PRun/QRun/SRun 以及

PQStart(40H)的控制。用户可以通过 POS (54H.Bit6) 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。

PFOUT/QFOUT/SFOUT 引脚默认功能分别为电能脉冲输出有功、无功、视在电能，也可通过 EMCON(58H)寄存器中的 CFxCFG 分别配置为输出不同的电能脉冲，同时可通过控制位 POS(54H.BIT6)选择 PF/QF/SF 引脚输出的有效电平。

用户也可配置使能 EMUCTRL.P2EN (74H.BIT11) 将第二电量模块 QFCNT 的脉冲输出复用为 P2 有功能量的脉冲输出，从而实现双通道的有功能量输出。

PF/QF/SF 输出满足下面时序关系：



脉冲输出波形

时序特性：

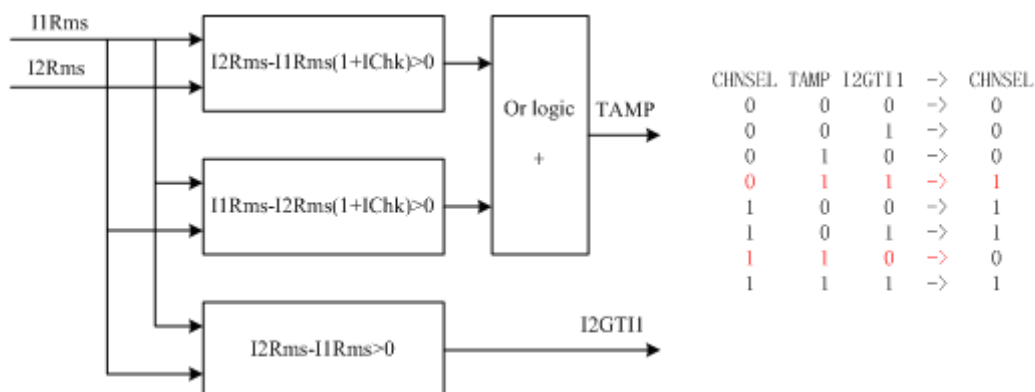
参数	指标	单位	测试条件及注释
t4	80	ms	PF/QF/SF 输出的高电平脉宽
t5	—	s	PF/QF/SF 输出的周期

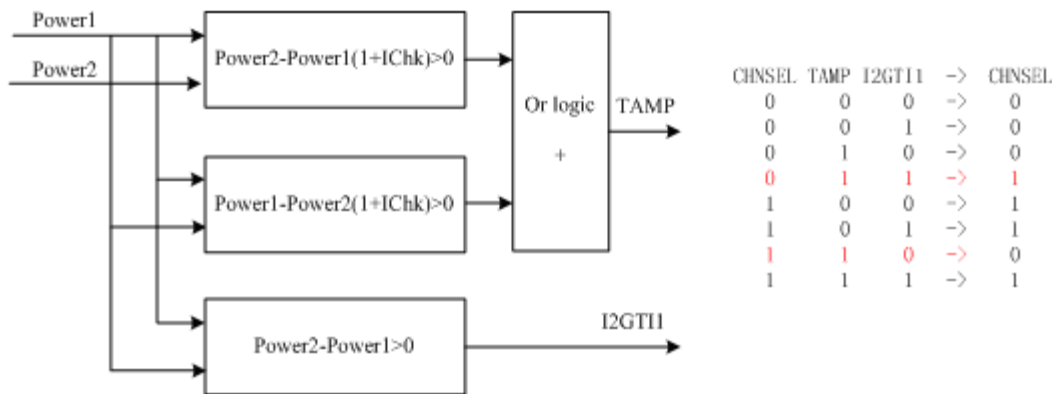
注意：当脉冲输出周期小于 180ms 时，脉冲以等 duty 形式输出。

HT502X 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。

## 23.2.10 窃电检测

可以通过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。





自动窃电方案

CHNSEL	TAMP	I2GTI1	->	CHNSEL
0	0	0	->	0
0	0	1	->	0
0	1	0	->	0
0	1	1	->	1
1	0	0	->	1
1	0	1	->	1
1	1	0	->	0
1	1	1	->	1

可以通过 FLTON(50H.5)设置是否开启自动窃电功能。FLTON=0 时,用户可以根据 CHNSEL(50H.4)进行通道选择; FLTON=1 时, 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。

I2GTI(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窃电阈值。

### 自动窃电设置步骤: (以电流窃电为例)

- 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) 根据需要的窃电阈值, 设置 ICHK(70H)。
- d) 根据需要检测窃电的最小电流或功率, 设置 IPTAMP(6CH)。
- e) 设置 FLTON(0x50H.5)为 1, 开启自动窃电功能。

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

## 23.2.11 中断源

HT502X 的 EMU 提供 17 个中断源, 在 EMU 中断使能寄存器 (EMUIE, 04H) 中定义: ZXlost\_IRQ、Buffer\_Full\_IRQ、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, 分别表示电压过零丢失中断请求、同步缓存数据存满中断请求、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

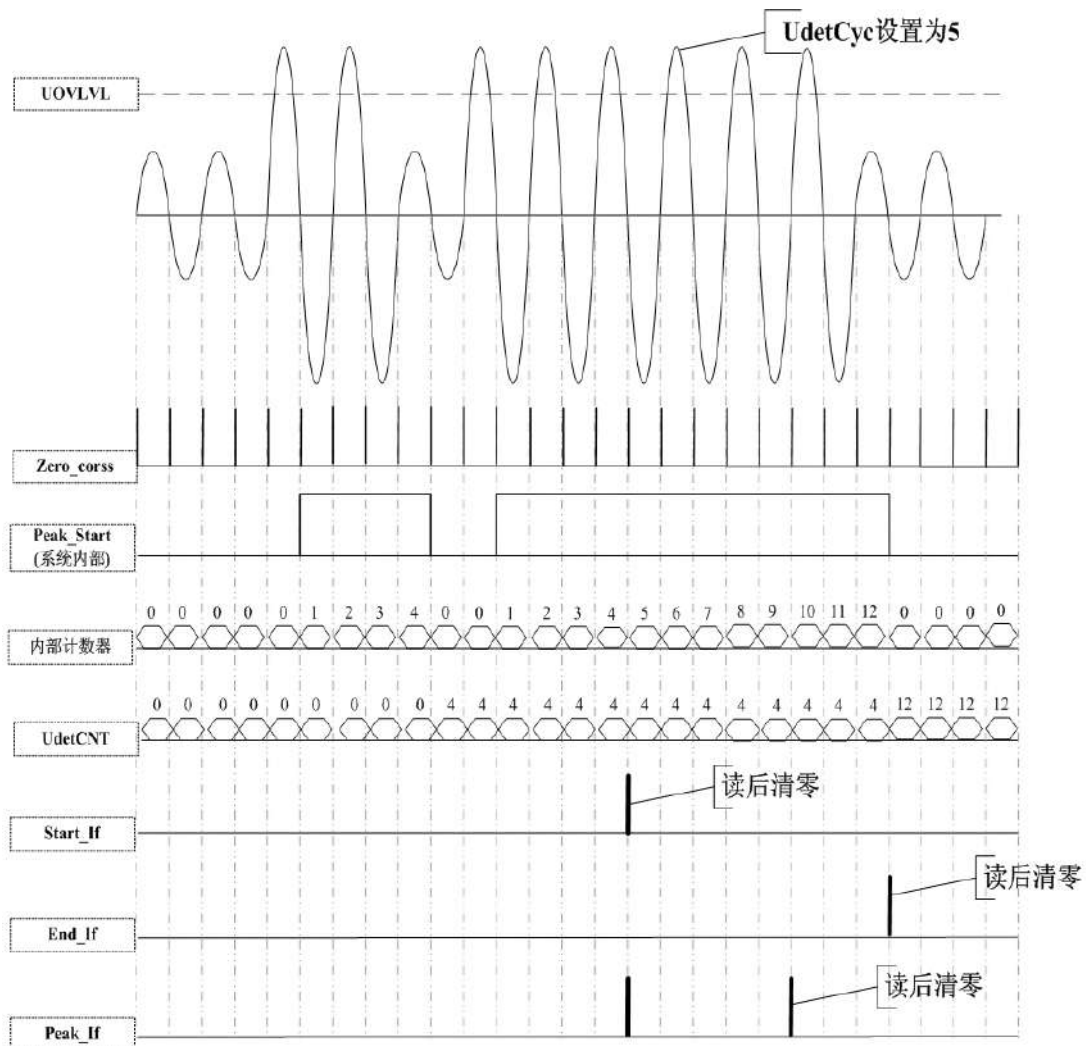
和 ZXD0 可以选择正向过零或者反向过零或者双向过零（详见 74H）。

EMU 17 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 后即可清除所有 EMU 中断标志。

## 23.2.12 PEAK / SAG 功能

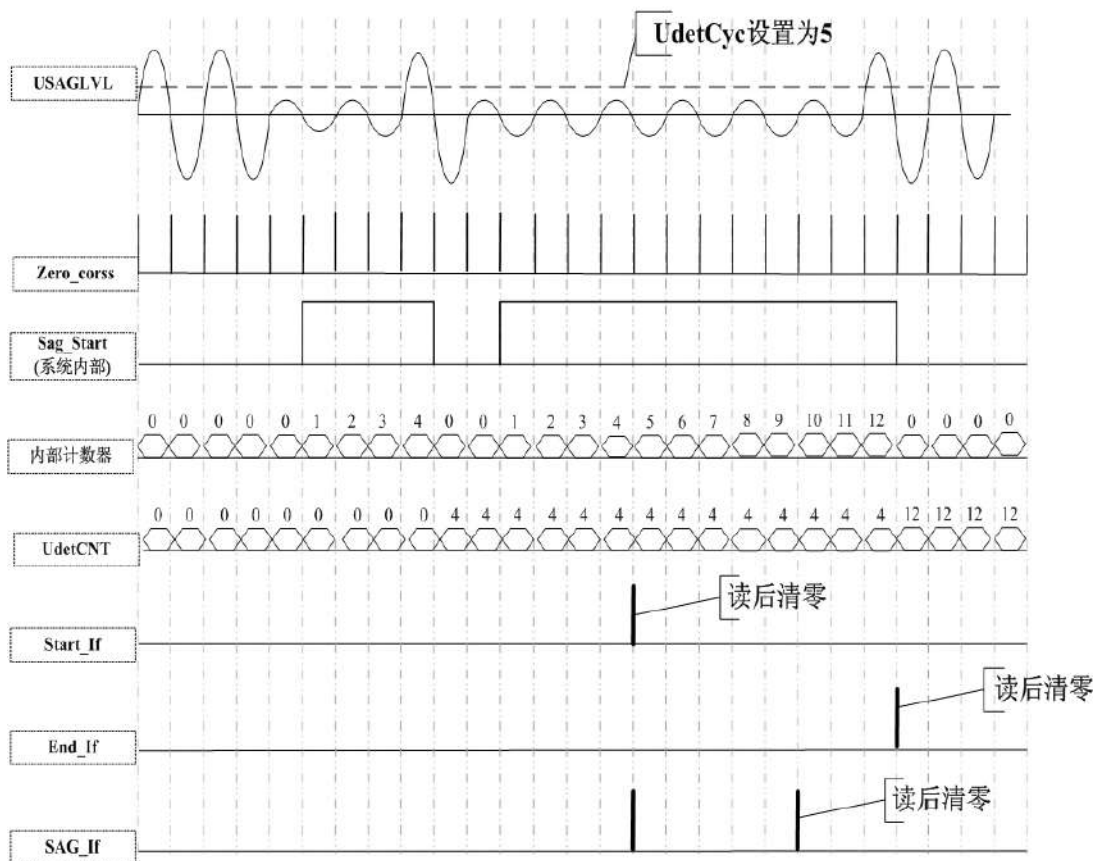
**PEAK 事件定义为：**系统对电压通道高通后数据以正负过零方式进行半周波检测峰值，同时将峰值更新至 Upd\_Half 寄存器（0x88）。当某次 Upd\_Half 绝对值大于 UOVLVL 寄存器（0xAC）设定的 PEAK 阈值，我们定义为 Peak\_Start 事件，系统内部的计数器开始对 Peak\_Start 事件进行计数，直至某次 Upd\_Half 绝对值小于设定阈值，将计数值更新至 UdetCNT，同时内部计数器清零。若内部的计数器计数大于等于 OVCyc 设定的周期数，则 Start\_If 标志位置 1（读后清零），同时 Peak\_If 标志位置 1（读后清零），直至某次 Upd\_Half 绝对值小于设定阈值，将计数更新至 UdetCNT，内部计数器清零，Start\_If 标志位清 0，同时 Peak\_If 标志位清 0，End\_If 置 1（读后清零）。PEAK 事件时序图如下图所示：



Peak 事件

**SAG 事件定义为：**系统对电压通道高通后数据以正负过零方式进行半周波检测峰值，同时将峰值更新至 Upd\_Half 寄存器 (0x88)。当某次 Upd\_Half 绝对值小于 USAGLVL 寄存器 (0xA4) 设定的 PEAK 阈值，我们定义为 Sag\_Start 事件，系统内部的计数器开始对 Sag\_start 事件进行计数，直至某次 Upd\_Half 绝对值大于设定阈值，将计数值更新至 UdetCNT，同时内部计数器清零。若内部的计数器计数大于等于设定的周期数 (OVCyc)，则 Start\_If 标志位置 1 (读后清零)，同时 Sag\_If 标志位置 1 (读后清零)，直至某次 Upd\_Half 绝对值大于设定阈值，将计数更新至 UdetCNT，内部计数器清零，Start\_If 标志位置 0，同时 Sag\_If 标志位置 0，End\_If 置 1 (读后清零)。

其中，电压通道的 SAG 事件和 PEAK 事件共用 UdetCNT，Start\_If，End\_If 寄存器。SAG 事件时序图如下图所示：



Sag 事件

### 23.2.13 OVI 过压过流检测功能

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。

#### 相关寄存器：

- 1) 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、IIOVIE、I2OVIE (04H)：置 1 可使能过压、过流中断输出。
- 4) OVI 状态标志位 UOVIF、IIOVIF、I2OVIF (08H)：读到 1 表示发生过压事件、过流事件，读后清 0；
- 5) 计量寄存器：电压波形点最值 Upd\_Half (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。

另该功能未设置专门的使能开关，复位后即保持工作状态，数据源取经高通后的电压、电流波形点。

### 23.2.14 掉零线计量模式(Loss of Voltage)

HT502X 在用户进入失压模式，为兼顾计量精度与功耗的矛盾，提供了几种可配置的掉零线计量功能：

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，可配置从有功电能输出引脚 (PF) 输出视在脉冲信号。

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，用于低频连续计量模式，常数计量模式不使用
- (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) GP：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连续计量模式，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，使能常数功率计量，芯片内部会一直用该常数功率累加发脉冲并累加电能，而系统可在保持 EMU 模块打开的情况下进入低功耗状态，直到下次唤醒可根据实际负载更新常数功率寄存器。

### 23.2.15 随频采样 ADC 波形缓存

HT502X 内置 128\*16bit 的随频采样 ADC 波形缓存 buffer，用于存放随频率变化的 ADC 同步采样数据，供用户做进一步的分析。通过配置校表参数寄存器 BufferStart (10CH) 指定需要缓存的通道与增益，并通过校表参数寄存 BufferCoff (110H) 调整采样率，将相应 ADC 波形数据保存到随频采样缓存 Buffer

中，写指针自动加 1，直到缓存 buffer 存满为止。存满可产生中断标志，设置 EMUIE 寄存器的 Buffer\_Full IE 使能 Buffer 满中断功能即可。在用户发送新的缓存命令之前，缓存的数据将会一直保持上一次的数据；若缓存未滿，又重新发送了启动缓存命令，命令会即刻执行，并从缓存 buff 的地址 0000H 开始重新保存数据。随频缓存功能不可以在低功耗计量模式下使用。

**缓存数据手动调整系数：**用户可通过手动校表寄存器的调整系数 BufferCoff (110H) 的方式，使 128\*16bit 的随频缓存数据调整为两个完整信号周波，即每周波 64 个点。输入信号范围 45Hz~65Hz。

**随频缓存数据的读取：**Buffer 数据存满后，用户可通过读计量寄存器 R\_Buffer (ACH) 取出数据。每读一次此寄存器，内部缓存地址指针自动加 1，直到缓存 Buffer 的值全部被读出。建议用户在启动了波形缓存功能后，必须等到 Buffer 存满之后才能进行读出操作（存满有中断标志）。如果在存入 Buffer 的过程中提前进行读操作，读出的数据有可能会出错（表现为第 1 个点错）。

**缓存的数据格式：**16bitADC 数据为补码格式，实际位数 Bit12-Bit0 是数据，Bit15-Bit13 是 3 个符号位。因此信号大小为  $\text{Buffer} / 2^{13} / \text{PGA}$ ，位数和 PGA 无关。

### 23.2.16 Q 能量脉冲通道复用 P2

HT502X 的无功电能模块可配置为第二通道计量的有功能量脉冲输出。通过配置寄存器 EMUCTRL (74H.BIT11) P2EN 使能之后，第二计量通道的有功电能 P2 代替了 QFCNT 的电能脉冲输出，从而实现双通道的有功能量同时输出。

### 23.3 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	SPLI1	R	3	0x000000	电流通道 1 的 ADC 采样数据，22bit
04H	SPLI2	R	3	0x000000	电流通道 2 的 ADC 采样数据，22bit
08H	SPLU	R	3	0x000000	电压通道的 ADC 采样数据 22bit
0CH	SPLP	R	4	0x000000	有功功率波形数据
10H	SPLQ	R	4	0x000000	无功功率波形数据
14H	FastRMSI1	R	3	0x000000	快速电流通道 1 的有效值
18H	FastRMSI2	R	3	0x000000	快速电流通道 2 的有效值
1CH	FastRMSU	R	3	0x000000	快速电压通道的有效值
20H	FreqU	R	2	0xFFFF	电压频率
24H	FastPowerP1	R	4	0x000000	快速第一路有功功率
28H	FastPowerQ1	R	4	0x000000	快速第一路无功功率
2CH	FastPowerP2	R	4	0x000000	快速第二路有功功率
30H	FastPowerQ2	R	4	0x000000	快速第二路无功功率
34H	FastPowerS1	R	4	0x000000	快速第一路视在功率
38H	FastPowerS2	R	4	0x000000	快速第二路视在功率
3CH	RMSI1	R	3	0x000000	慢速电流通道 1 的有效值
40H	RMSI2	R	3	0x000000	慢速电流通道 2 的有效值
44H	RMSU	R	3	0x000000	慢速电压通道的有效值
48H	PowerP1	R	4	0x000000	慢速第一路有功功率
4CH	PowerQ1	R	4	0x000000	慢速第一路无功功率
50H	PowerP2	R	4	0x000000	慢速第二路有功功率
54H	PowerQ2	R	4	0x000000	慢速第二路无功功率
58H	PowerS1	R	4	0x000000	慢速第一路视在功率
5CH	PowerS2	R	4	0x000000	慢速第二路视在功率
60H	EnergyP	R	3	0x000000	有功能量
64H	EnergyQ	R	3	0x000000	无功能量
68H	EnergyS	R	3	0x000000	视在能量
6CH	EnergyPC	R	3	0x000000	读后清 0 型有功能量
70H	EnergyQC	R	3	0x000000	读后清 0 型无功能量
74H	EnergySC	R	3	0x000000	读后清 0 型视在能量
78H	DC_UAverage	R	4	0x000000	直流电压平均值
7CH	DC_I1Average	R	4	0x000000	直流电流通道 1 平均值
80H	DC_I2Average	R	4	0x000000	直流电流通道 2 平均值
84H	Checksum	R	3	0x008EBA	校表寄存器校验和寄存器

88H	Upd_Half	R	3	0x000000	电压半波周期峰值寄存器, 19bit。
8CH	I1pd_Half	R	3	0x000000	I1 半波周期峰值寄存器, 19bit。
90H	I2pd_Half	R	3	0x000000	I2 半波周期峰值寄存器, 19bit。
94H	PFCNT_Photo	R	3	0x000000	PFCNT 快照
98H	QFCNT_Photo	R	3	0x000000	QFCNT 快照
9CH	SFCNT_Photo	R	3	0x000000	SFCNT 快照
A0H	AutoUgain	R	2	0x000000	U 通道温度自动补偿系数
A4H	AutoI1gain	R	2	0x000000	I1 通道温度自动补偿系数
A8H	AutoI2gain	R	2	0x000000	I2 通道温度自动补偿系数
ACH	R_Buffer	R	4	0x000000	ADC 缓存 buffer 读取数据寄存器
B0H	Reserved	R	4	0x000000	保留寄存器
B4H	Reserved	R	4	0x000000	保留寄存器
B8H	UdetCNT	R	3	0x000000	SAG/Peak 工况持续时间计数寄存器

## 23.4 计量参数寄存器说明

### ADC 波形寄存器

<b>SPLI1</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 00H</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

<b>SPLI2</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 04H</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

<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

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，其中 bit21 到 bit23 为扩展的符号位，bit20~bit0 为数据。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。

### 功率波形寄存器

<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

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。

## 快速有效值寄存器

<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

<b>FastRMSI2</b>	<b>基地址: 0x40013000</b>							
	<b>偏移地址: 18H</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

<b>FastRMSU</b>	<b>基地址: 0x40013000</b>							
	<b>偏移地址: 1CH</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

快速有效值寄存器是 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。

## 电压频率寄存器

<b>FreqU</b>	<b>基地址: 0x40013000</b>							
	<b>偏移地址: 20H</b>							

	<b>Bit15...Bit0</b>							
<b>Read:</b>	DAT[15:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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。

## 快速功率寄存器

<b>FastPowerP1</b>	<b>基地址： 0x40013000</b> <b>偏移地址： 24H</b>							
	<b>Bit31...Bit0</b>							
<b>Read:</b>	X							
<b>Write:</b>	DAT[31:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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

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

<b>FastPowerQ2</b>	<b>基地址： 0x40013000</b> <b>偏移地址： 30H</b>							
	<b>Bit31...Bit0</b>							
<b>Read:</b>	X							

<b>Write:</b>	DAT[31:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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

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

快速功率为 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。

设寄存器中的数据为 FastPowerP1，则供计算用的 Preg 为：

Preg=FastPowerP1 ; 如果 FastPowerP1<2<sup>31</sup>

Preg=FastPowerP1-2<sup>32</sup> ; 如果 FastPowerP1>=2<sup>31</sup>

用于显示的转换比例系数计算方法详见慢速功率寄存器描述。

## 慢速有效值寄存器

<b>RMSI1</b>	<b>基地址: 0x40013000</b> <b>偏移地址: 3CH</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

<b>RMSI2</b>	<b>基地址: 0x40013000</b> <b>偏移地址: 40H</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

<b>RMSU</b>	<b>基地址: 0x40013000</b>							
-------------	------------------------	--	--	--	--	--	--	--

		<b>偏移地址: 44H</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	

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在快速的基础上降低到 1/4 即 1.25Hz，可配置 EMUCTRL 寄存器配置为相应快速的 1/2，到最高 10Hz。

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。

### 慢速功率寄存器

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

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

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

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

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

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

慢速功率都为 32 位二进制补码格式，最高位为符号位，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。

设寄存器中的数据为 PowerP1，则供计算用的 Preg 为：

Preg=PowerP1 ; 如果 PowerP1<2<sup>31</sup>

Preg=PowerP1-2<sup>32</sup> ; 如果 PowerP1>=2<sup>31</sup>

设显示的有功功率为 P，转换系数为 Kpqs，则：

$P = \text{Preg} \times K_{pqs}$

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

$K_{pqs} = 1000/51673 = 0.01935$

当 PowerP1 读数为 0xFFFF4534 时，其代表的功率值为：

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

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

## 电能寄存器

<b>EnergyP</b>	<b>基地址: 0x40013000</b>							
	<b>偏移地址: 60H</b>							
	<b>Bit23...Bit0</b>							
<b>Read:</b>	X							
<b>Write:</b>	DAT[23:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>EnergyQ</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 64H</b>						
		<b>Bit23...Bit0</b>						
<b>Read:</b>	X							
<b>Write:</b>	DAT[23:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>EnergyS</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 68H</b>						
		<b>Bit23...Bit0</b>						
<b>Read:</b>	X							
<b>Write:</b>	DAT[23:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

EnergyP / EnergyQ / EnergyS 能量累加寄存器为读后不清 0 型能量寄存器，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

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

<b>EnergyPC</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 6CH</b>						
		<b>Bit23...Bit0</b>						
<b>Read:</b>	X							
<b>Write:</b>	DAT[23:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>EnergyQC</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 70H</b>						
		<b>Bit23...Bit0</b>						
<b>Read:</b>	X							
<b>Write:</b>	DAT[23:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>EnergySC</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 74H</b>						
		<b>Bit23...Bit0</b>						
<b>Read:</b>	X							
<b>Write:</b>	DAT[23:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

EnergyP / EnergyQ / EnergyS 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

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

## ADC 直流采样数据寄存器

<b>DC_UAverage</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 78H</b>						
		<b>Bit31...Bit0</b>						
<b>Read:</b>	X							
<b>Write:</b>	DAT[31:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>DC_I1Average</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 7CH</b>						
		<b>Bit31...Bit0</b>						
<b>Read:</b>	X							
<b>Write:</b>	DAT[31:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>DC_I2Average</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 80H</b>						
		<b>Bit31...Bit0</b>						
<b>Read:</b>	X							
<b>Write:</b>	DAT[31:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

直流平均值 DC\_Average 是 32 位有符号数，最高位为符号位。当 EMU 时钟频率为 819.2KHz 时，参数更新频率默认为 3.125Hz。配置 EMUCTRL [9:8] (74H) 调整更新速率可配最高 25Hz。

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。

## 参数校验和寄存器

<b>Checksum</b>		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 84H</b>						
		<b>Bit23...Bit0</b>						
<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，包含基地址 0x40013800，偏移地址 0CH---104H（计算不包含 5CH-64H 寄存器）的校表参数寄存器。

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。

## 峰值寄存器

Upd_Half		基地址: 0x40013000 偏移地址: 88H						
		Bit23...Bit0						
Read:	X							
Write:	DAT[23:0]							
Reset:	0	0	0	0	0	0	0	0

I1pd_Half		基地址: 0x40013000 偏移地址: 8CH						
		Bit23...Bit0						
Read:	X							
Write:	DAT[23:0]							
Reset:	0	0	0	0	0	0	0	0

I2pd_Half		基地址: 0x40013000 偏移地址: 90H						
		Bit23...Bit0						
Read:	X							
Write:	DAT[23:0]							
Reset:	0	0	0	0	0	0	0	0

峰值寄存器，每次 ADC 过零，将该寄存器更新，得到此半波时间内的 ADC 波形绝对值的最大值，该寄存器取自高通后的数据。为了和 ADC 的位数对齐，该寄存器有效位数为 19bit，即 Bit0~Bit17 为数据，Bit18~Bit23 均为扩展的符号位。数据采用二进制补码格式。

## 快速能量脉冲快照寄存器

PFCNT_Photo		基地址: 0x40013000 偏移地址: 94H						
		Bit15...Bit0						
Read:	DAT[15:0]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0

QFCNT_Photo		基地址: 0x40013000 偏移地址: 98H						
		Bit15...Bit0						

<b>Read:</b>	DAT[15:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>SFCNT_Photo</b>	<b>基地址: 0x40013000</b>							
	<b>偏移地址: 9CH</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	DAT[15:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

快速能量脉冲快照寄存器, 当用户读取有功能量寄存器时(不清零型能量寄存器/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 的值为 0x20, Hfconst 寄存器值为 0x40, 则 PFCNT\_Photo 表示的电能为:  $(0x20/0x40) / EC$ , 单位为 kWh。

QFCNT\_Photo SFCNT\_Photo 功能相同。

## 自动温补增益寄存器

<b>AutoUgain</b>	<b>基地址: 0x40012000</b>							
	<b>偏移地址: A0H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	X							
<b>Write:</b>	DAT[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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。

<b>AutoI1gain</b>	<b>基地址: 0x40012000</b>							
	<b>偏移地址: A4H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	X							
<b>Write:</b>	DAT[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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。

<b>AutoI2gain</b>	<b>基地址: 0x40012000</b>							
	<b>偏移地址: A8H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	X							
<b>Write:</b>	DAT[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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。

## 同步波形数据缓存寄存器

<b>R_Buffer</b>	<b>基地址: 0x40012000</b>							
	<b>偏移地址: ACH</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	X							
<b>Write:</b>	DAT[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

同步波形数据缓存寄存器，在启动缓存之后其数据才有效。缓存寄存器启动：校表寄存器 **BufferStar** (0100H) 写入 0xCCCX。ADC 同步采样数据存满产生缓存结束中断标志 (Buffer\_FullIF 置位)，用户可读取 R\_Buffer 获取缓存 buff 的数据。

R\_Buffer 存放的是首个 ADC 波形同步缓存数据，每读一次 R\_Buffer，内部计数器自动更新为下一个 ADC 波形数据地址。当读取次数达到 128，或者重新启动 ADC 波形数据缓存时，R\_Buffer 将回到缓存 buff 的首地址。

<b>Reserved</b>	<b>基地址: 0x40012000</b>							
	<b>偏移地址: B0H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	X							
<b>Write:</b>	DAT[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>Reserved</b>	<b>基地址: 0x40012000</b>							
	<b>偏移地址: B4H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	X							
<b>Write:</b>	DAT[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

## SAG/PEAK 工况持续时间计数寄存器

<b>UdetCNT</b>	<b>基地址: 0x40013000</b>							
	<b>偏移地址: B8H</b>							
	<b>Bit23...Bit0</b>							
<b>Read:</b>	X							
<b>Write:</b>	DAT[23:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

SAG/PEAK 工况持续时间计数寄存器，当电压通道 SAG/PEAK 检测使能，且检测到 SAG/PEAK 事件开始

建立， UdetCnt 以半周波数为单位计数，直到检测到退出 SAG/PEAK 事件，更新 UdetCNT 寄存器值。此寄存器记录的是 SAG/PEAK 事件持续的半周波数。

### 23.5 校表参数寄存器列表

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	EMUSR	R/W	2	0x0000	EMU 状态标志寄存器
04H	EMUIE	R/W	3	0x000000	EMU 中断使能寄存器
08H	EMUIF	R/W	3	0x000000	EMU 中断标志寄存器
0CH	GP1	R/W	2	0x0000	通道 1 的有功功率校正寄存器
10H	RSRV	R/W	2	0x0000	---
14H	RSRV	R/W	2	0x0000	---
18H	GPhs1	R/W	2	0x0000	通道 1 的相位校正寄存器
1CH	GP2	R/W	2	0x0000	通道 2 的有功功率校正寄存器
20H	RSRV	R/W	2	0x0000	---
24H	RSRV	R/W	2	0x0000	---
28H	GPhs2	R/W	2	0x0000	通道 2 的相位校正寄存器
2CH	QPhsCal	R/W	2	0x0000	无功相位补偿寄存器
30H	I2Gain	R/W	2	0x0000	电流通道 2 增益补偿寄存器
34H	I1Off	R/W	2	0x0000	I1 直流偏置校正寄存器
38H	I2Off	R/W	2	0x0000	I2 直流偏置校正寄存器
3CH	UOff	R/W	2	0x0000	U 通道直流偏置校正寄存器
40H	PStart	R/W	2	0x0040	起动功率设置寄存器
44H	QStart	R/W	2	0x0080	起动功率设置寄存器
48H	SStart	R/W	2	0x0090	起动功率设置寄存器
4CH	HFConst	R/W	2	0x0080	输出脉冲频率设置寄存器
50H	ADCCFG	R/W	2	0x0003	ADC 控制寄存器
54H	CHNLCR	R/W	2	0x0607	通道控制寄存器
58H	EMCON	R/W	2	0x1870	能量计量控制寄存器
5CH	PFCnt	R/W	2	0x0000	快速有功脉冲计数寄存器
60H	QFCnt	R/W	2	0x0000	快速无功脉冲计数寄存器
64H	SFCnt	R/W	2	0x0000	快速视在脉冲计数寄存器
68H	ADCCON	R/W	2	0x0000	ADC 通道增益选择配置寄存器
6CH	IPTAMP	R/W	2	0x0020	窃电检测域值寄存器
70H	ICLK	R/W	2	0x0010	窃电阈值设置寄存器
74H	EMUCTRL	R/W	2	0x0382	EMU 控制寄存器
78H	P1OFFSET	R/W	2	0x0000	通道 1 有功功率小信号偏置校正寄存器
7CH	P2OFFSET	R/W	2	0x0000	通道 2 有功功率小信号偏置校正寄存器
80H	Q1OFFSET	R/W	2	0x0000	通道 1 无功功率小信号偏置校正寄存器
84H	Q2OFFSET	R/W	2	0x0000	通道 2 无功功率小信号偏置校正寄存器

					器
88H	I1RMSOFFSET	R/W	2	0x0000	电流通道 1 有效值小信号校正寄存器
8CH	I2RMSOFFSET	R/W	2	0x0000	电流通道 2 有效值小信号校正寄存器
90H	URMSOFFSET	R/W	2	0x0000	电压通道有效值小信号校正寄存器
94H	RosiCtrl	R/W	2	0x4000	罗氏线圈使能控制寄存器
98H	ANA_control	R/W	2	0x27A6	模拟控制寄存器
9CH	UCONST	R/W	2	0x0000	失压计量电压寄存器
A0H	LpIdleTime	R/W	2	0x0000	低功耗计量分时空闲时间设置寄存器
A4H	USAGLVL	R/W	2	0x0000	电压跌落检测阈值寄存器
A8H	IpeakCyc	R/W	2	0x0020	电流 I1 I2 PEAK 检测周期寄存器
ACH	UOVLVL	R/W	2	0x0000	电压 PEAK 检测阈值寄存器
B0H	OVCyc	R/W		0x0001	电压 SAG\PEAK 检测周期寄存器
B4H	IOVLVL	R/W	2	0x0000	电流过流阈值寄存器
B8H	ZXILVL	R/W		0x0000	电流过零阈值寄存器
BCH	PDataCpH	R/W	2	0x0000	常数有功功率累加高 16 位寄存器
C0H	PDataCpL	R/W	2	0x0000	常数有功功率累加低 16 位寄存器
C4H	QDataCpH	R/W	2	0x0000	常数无功功率累加高 16 位寄存器
C8H	QDataCpL	R/W	2	0x0000	常数无功功率累加低 16 位寄存器
CCH	SDataCpH	R/W	2	0x0000	常数视在功率累加高 16 位寄存器
D0H	SDataCpL	R/W	2	0x0000	常数视在功率累加低 16 位寄存器
D4H	FilterCtrl	R/W	2	0x02F7	滤波器系数选择控制寄存器
D8H	TUgain	R/W	2	0x0000	电压通道手动温度补偿系数设置寄存器
DCH	TI1gain	R/W	2	0x0000	I1 通道手动温度补偿系数设置寄存器
E0H	TI2gain	R/W	2	0x0000	I2 通道手动温度补偿系数设置寄存器
E4H	UTCcoeffA	R/W	2	0x0000	U 通道自动温补 VrefGain 的二次系数设置寄存器
E8H	UTCcoeffB	R/W	2	0x0000	U 通道自动温补 VrefGain 的一次系数设置寄存器
ECH	UTCcoeffC	R/W	2	0x0000	U 通道自动温补 VrefGain 的常数项设置寄存器
F0H	I1TCcoeffA	R/W	2	0x0000	I1 通道自动温补 VrefGain 二次系数设置寄存器
F4H	I1TCcoeffB	R/W	2	0x0000	I1 通道自动温补 VrefGain 一次系数设置寄存器

F8H	I1TCcoeffC	R/W	2	0x0000	I1 通道自动温补 VrefGain 的常数项设置寄存器
FCH	I2TCcoeffA	R/W	2	0x0000	I2 通道自动温补 VrefGain 的二次系数设置寄存器
100H	I2TCcoeffB	R/W	2	0x0000	I2 通道自动温补 VrefGain 的一次系数设置寄存器
104H	I2TCcoeffC	R/W	2	0x0000	I2 通道自动温补 VrefGain 的常数项设置寄存器
以下寄存器不参与 Checksum 校表参数校验和计算:					
108H	LoadDataCp	R/W	2	0x0000	常数计量启动寄存器
10CH	BufferStart	R/W	2	0x0000	缓存数据启动寄存器
110H	BufferCoff	R/W	2	0x0100	缓存数据手动调整系数寄存器
114H	SRSTREG	R/W	1	0x00	EMU 软件复位寄存器
118H	PFCntN	R/W	2	0x0000	反向快速有功脉冲计数寄存器
11CH	QFCntN	R/W	2	0x0000	反向快速无功脉冲计数寄存器

## 23.6 校表参数寄存器说明

### EMUSR(EMU 状态寄存器)

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:	X	X	X	X	SFOF	PEOF	QEOF	CHSTS
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	<b>视在电能寄存器溢出标志</b> 1: 视在电能溢出 0: 未发生溢出 写 0 清零
PEOF	<b>有功电能寄存器溢出标志</b> 1: 有功电能溢出 0: 未发生溢出 写 0 清零
EOF	<b>无功电能寄存器溢出标志</b> 1: 无功电能溢出 0: 未发生溢出 写 0 清零
CHSTS	<b>计量通道状态标志</b> 1: 指示使用电流通道 2 计量 0: 指示使用电流通道 1 计量
TAMP	<b>窃电标志</b> 1: 表示发生窃电, 当 TampSel=0, 表示: $I1Rms > I2Rms * (1 + IChk)$ 或者 $I2Rms > I1Rms * (1 + IChk)$ ; 当 TampSel=1, 表示: $Power1 > Power2 * (1 + IChk)$ 或者 $Power2 > Power1 * (1 + IChk)$ ; 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。
I2GTI1	<b>通道 2 电流或功率大于通道 1 电流或功率标志</b> 1: 通道 2 电流或功率大于通道 1 当 TampSel=0, 表示: $I2Rms > I1Rms$ 当 TampSel=1, 表示: $Power2 > Power1$

	0: 通道 2 电流或功率小于通道 1 当 TampSel =0, 表示: I2Rms<I1Rms 当 TampSel =1, 表示: Power2<Power1
NoQLd2	<b>通道 2 无功功率潜动标志位</b> 1: 当无功功率 2 小于起动功率时, NoQLd1 被置为 1; 0: 当无功功率大于/等于起动功率时 NoQLd2 清为 0
NoPLd2	<b>通道 2 有功功率潜动标志位</b> 1: 当有功功率 2 小于起动功率时, NoPLd2 被置为 1; 0: 当有功功率大于/等于起动功率时 NoPLd2 清为 0
NoQLd1	<b>通道 1 无功功率潜动标志位</b> 1: 当无功功率 1 小于起动功率时, NoQLd1 被置为 1; 0: 当无功功率大于/等于起动功率时 NoQLd1 清为 0。
NoPLd1	<b>通道 1 有功功率潜动标志位</b> 1: 当有功功率 1 小于起动功率时, NoPLd1 被置为 1; 0: 当有功功率大于/等于起动功率时 NoPLd1 清为 0。
REVQ	<b>反向无功功率指示标识</b> 1: 当检测到负无功功率时, 该信号为 1 0: 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	<b>反向有功功率指示标识</b> 1: 当检测到负有功功率时, 该信号为 1; 0: 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 中断使能寄存器)

EMUIE		基地址: 0x40013800						
EMU 中断使能寄存器		偏移地址: 04H						
	Bit23	22	21	20	19	18	17	Bit16
<b>Read:</b>								
<b>Write:</b>	X	X	X	X	X		ZXlostIE	Buffer_Full IE
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>								
<b>Write:</b>	UdetIE	TAMPIE	DCaveUpdIE	SlowRmsUpdIE	FastRmsUpdIE	ZXI2IE	ZXI1IE	I2OVIE
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	PFIE	QFIE	SFIE	SPLIE	ZXIE	I1OVIE	UOVIE	SAGIE
<b>Write:</b>	X	X	X	X	X	X	X	X

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

位名称	描述
ZXlostIE	过零丢失中断使能 0: 禁止 1: 使能
Buffer_Full IE	缓存结束中断使能 0: 禁止 1: 使能
UdetIE	电压通道 SAG/PEAK 事件进入/退出中断使能 0: 禁止 1: 使能
TAMPIE	窃电发生中断使能 0: 禁止 1: 使能
DCaveUpdIE	直流平均值更新中断使能 0: 禁止 1: 使能
SlowRmsUpdIE	慢速有效值更新中断使能 0: 禁止 1: 使能
FastRmsUpdIE	快速有效值更新中断使能 0: 禁止 1: 使能
ZXI2IE	<b>I2</b> 发生用户指定的过零中断使能 0: 禁止 1: 使能
ZXI1IE	<b>I1</b> 发生用户指定的过零中断使能 0: 禁止 1: 使能
I2OVIE	<b>I2</b> 通道过流中断使能 0: 禁止 1: 使能
PFIE	<b>PF</b> 脉冲中断使能 0: 禁止 1: 使能
QFIE	<b>QF</b> 脉冲中断使能 0: 禁止 1: 使能
SFIE	<b>SF</b> 脉冲中断使能 0: 禁止 1: 使能
SPLIE	波形寄存器更新中断使能

	0: 禁止 1: 使能
ZXIE	<b>U 通道发生用户指定的过零方式中断使能</b> 0: 禁止 1: 使能
IIOVIE	<b>I1 通道过流中断使能</b> 0: 禁止 1: 使能
UOVIE	<b>过压中断使能</b> 0: 禁止 1: 使能
SAGIE	<b>U 通道 SAG 中断使能</b> 0: 禁止 1: 使能 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

### EMUIF(EMU 中断标志寄存器)

<b>EMUIF</b> EMU 中断标志寄存器			<b>基地址: 0x40013800</b> <b>偏移地址: 08H</b>					
	<b>Bit23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>Bit16</b>
<b>Read:</b>	X	X	X	X	X	ZXlostIF	Buffer_FullIF	Uend
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
	<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>	Ustart	TAMPiF	DCAveUpdIF	SlowRmsUpdIF	FastRmsUpdIF	ZXI2IF	ZXI1IF	I2OVIF
<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>	PFIF	QFIF	SFIF	SPLIF	ZXIF	IIOVIF	UOVIF	SAGIF
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位名称	描述
ZXlostIF	<b>过零丢失中断标志位</b> 1: 发生过零中断丢失 0: 未发生过零中毒丢失
Buffer_FullIF	<b>缓存结束中断标志位</b> 1: ADC 数据缓存结束中断 0: 未发生 ADC 数据缓存结束
Uend	<b>电压通道退出 SAG/PEAK 事件中断标志</b>

	<p>1: 退出 SAG/PEAK 事件中断发生</p> <p>0: 未发生退出 SAG/PEAK 事件</p> <p>读后清零</p>
Ustart	<p><b>电压通道进入 SAG/Peak 事件中断标志</b></p> <p>1: 进入 SAG/PEAK 事件中断发生</p> <p>0: 未发生进入 SAG/PEAK 事件</p> <p>读后清零</p>
TAMPIF	<p><b>当发生窃电时的中断标志</b></p> <p>1: 发生窃电</p> <p>0: 未发生窃电</p> <p>读后清零</p>
DCAveUpdIF	<p><b>直流平均值更新中断标志</b></p> <p>1: 直流平均值更新</p> <p>0: 未更新</p> <p>读后清零</p>
SlowRmsUpdIF	<p><b>慢速有效值更新中断标志</b></p> <p>1: 慢速有效值更新</p> <p>0: 未更新</p> <p>读后清零</p>
FastRmsUpdIF	<p><b>快速有效值更新中断标志</b></p> <p>1: 快速有效值更新</p> <p>0: 未更新</p> <p>读后清零</p>
ZXI2IF	<p><b>I2 通道发生用户指定的过零中断标志</b></p> <p>1: I2 通道发生用户指定的过零</p> <p>0: 未发生</p> <p>读后清零</p>
ZXI1IF	<p><b>I1 通道发生用户指定的过零中断标志</b></p> <p>1: I1 通道发生用户指定的过零</p> <p>0: 未发生</p> <p>读后清零</p>
I2OVIF	<p><b>I2 通道过流中断标志</b></p> <p>1: I2 通道发生过流中断</p> <p>0: 未发生过流</p> <p>I2Peak 持续 OVCyc 个半周波都超过 IOVLVL 阈值发生置位, 读后清零</p>
PFIF	<p><b>PF 脉冲中断标志</b></p> <p>1: PF 发出脉冲中断</p> <p>0: 未发生 PF 脉冲</p> <p>在 PF 发脉冲的开始沿置位, 读后清零</p>
QFIF	<p><b>QF 脉冲中断标志</b></p> <p>1: QF 发出脉冲中断</p> <p>0: 未发生 QF 脉冲</p> <p>在 QF 发脉冲的开始沿置位, 读后清零</p>

SFIF	<b>SF 脉冲中断标志</b> 1: SF 发出脉冲中断 0: 未发生 SF 脉冲 在 SF 发脉冲的开始沿置位, 读后清零
SPLIF	<b>波形寄存器更新中断标志</b> 1: SPL 波形寄存器发生数据更新 0: 未发生 读后清零
ZXIF	<b>U 通道发生用户指定方式过零中断标志</b> 1: 电压通道发生用户指定方式过零 0: 未发生 读后清零
IIOVIF	<b>I1 通道过流中断标志</b> 1: I1 通道发生过流中断 0: 未发生过流 I1pd_Half 持续 OVCyc 个半周波都超过 IOVLVL 阈值发生置位, 读后清零
UOVIF	<b>U 通道过压中断标志</b> 1: U 通道发生过压中断 0: 未发生过压 Upd_Half 持续 OVCyc 个半周波都超过 UOVLVL 阈值发生置位, 读后清零
SAGIF	<b>U 通道 SAG 事件中断标志</b> 1: SAG 事件发生 0: 未发生 持续电压跌落低于设定阈值并超出用户指定半周波数, 读后清零

## GP1(通道 1 有功功率校正寄存器)

<b>GP1</b>	<b>基地址: 0x40013800</b>							
<b>通道 1 有功功率校正寄存器</b>	<b>偏移地址: 0CH</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	GP1[15:0]							
<b>Write:</b>	GP1[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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:  $65536 + Pgain * 32768$

GP1 寄存器写入 Pgain 之后, 通道 1 的有功增益/无功增益/视在增益 (GP1/GQ1/GS1) 同时被校正。

### GPHS1(通道 1 相位校正寄存器)

<b>GPHS1</b> 通道 1 相位校正寄存器		<b>基地址: 0x40013800</b>						
		<b>偏移地址: 18H</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	GPHS1[15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

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:  $G_{phs1} = -Err\% * 32768 / 1.732$

如果 Err 为正数:  $G_{phs1} = 65536 - Err\% * 32768 / 1.732$

### GP2(通道 2 有功功率校正寄存器)

<b>GP2</b> 通道 2 有功功率校正寄存器		<b>基地址: 0x40013800</b>						
		<b>偏移地址: 1CH</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	GP2[15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

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%

$P_{gain} = -Err\% / (1 + Err\%)$

如果  $P_{gain}$  为正数, 则 GP2 的写入值为:  $P_{gain} * 32768$

如果  $P_{gain}$  为负数, 则 GP2 的写入值为:  $65536 + P_{gain} * 32768$

GP2 寄存器写入  $P_{gain}$  之后, 通道 2 的有功增益/无功增益/视在增益 (GP2/GQ2/GS2) 同时被校正。

### GPHS2(通道 2 相位校正寄存器)

<b>GPHS2</b> 通道 2 相位校正寄存器		<b>基地址: 0x40013800</b>						
		<b>偏移地址: 28H</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	GPHS2[15:0]							
<b>Write:</b>								
<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。具体的定义参见校表过程。

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(无功相位补偿寄存器)

<b>QPHSCal</b> 无功相位补偿寄存器	<b>基地址： 0x40013800</b>							
	<b>偏移地址： 2CH</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	QPHSCal[15:0]							
<b>Write:</b>	QPHSCal[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

该寄存器默认值 0000H。

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 0x0051， 可以实现准确的 90°移相。如果 OSR128 选择 128 时， QphsCal 为 0x0029。对应到 60Hz 输入信号 OSR128 选择 64 时， QphsCal 为 0x44。

## I2Gain(I2 增益补偿寄存器)

<b>I2Gain</b> I2 增益补偿寄存器	<b>基地址： 0x40013800</b>							
	<b>偏移地址： 30H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	I2Gain[15:0]							
<b>Write:</b>	I2Gain[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

I2 相对于 I1 的有效值增益校正,采用二进制补码形式，最高位为符号位。

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 I1rms，电流通道 2 有效值寄存器读数为 I2rms，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：**

则  $Gain = I1rms / I2rms - 1$

**如果选择功率防窃电方式：**

则  $Gain = Power1 / Power2 - 1$

如果  $Gain \geq 0$ ，  $I2Gain = Gain * 2^{15}$ ； 如果  $Gain < 0$ ，  $I2Gain = Gain * 2^{15} + 2^{16}$

### I1Off(I1 通道直流偏置校正寄存器)

<b>I1Off</b> I1 通道直流偏置校正寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 34H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	I1Off[15:0]							
<b>Write:</b>	I1Off[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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直流偏置校正寄存器。

### I2Off(I2 通道直流偏置校正寄存器)

<b>I2Off</b> I2 通道直流偏置校正寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 38H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	I2Off[15:0]							
<b>Write:</b>	I2Off[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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 04H 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。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。

### UOff(U 通道直流偏置校正寄存器)

<b>UOff</b> U 通道直流偏置校正寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 3CH</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	UOff[15:0]							
<b>Write:</b>	UOff[15:0]							

<b>Reset:</b>	0	0	0	0	0	0	0	0
---------------	---	---	---	---	---	---	---	---

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 08H 的值平均后，得到 UOff 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 UOff 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。

## PStart(有功功率启动设置寄存器)

<b>PStart</b> 有功功率启动设置寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 40H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	PQStart[15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

Default 值: 0x0040

PStart 是 16 位无符号数，做比较时，将其与 PowerP 绝对值的 bit8—bit23 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:

- 校表结束后，输入 Ib, Un。
- 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；
- 设写入 PStart 的值为 Y，假如要求 0.4%Ib 电表能够启动，则：  
 $Y < x2 * 0.4\%$   
一般推荐值为计算值的一半。

## QStart(无功功率启动设置寄存器)

<b>QStart</b> 无功功率启动设置寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 44H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	PQStart[15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

Default 值: 0x0080

QStart 是 16 位无符号数，做比较时，将其与 PowerQ 绝对值的 bit8—bit23 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:

- 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:  
 $Y < x2 * 0.4\%$   
一般推荐值为计算值的一半。

## SStart(视在功率启动设置寄存器)

<b>SStart</b> 视在功率启动设置寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 48H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	PQStart[15:0]							
<b>Write:</b>	PQStart[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

Default 值: 0x0090

SStart 是 16 位无符号数, 做比较时, 将其与 PowerS 值的 bit8—bit23 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:  
 $Y < x2 * 0.4\%$   
一般推荐值为计算值的一半。

HT502X 提供了两种起动潜动方式: 单独起动潜动 与联合起动潜动。具体参考 23.2.6 章节起动/潜动功能描述。

## HFCnst(高频脉冲常数设置寄存器)

<b>HFCnst</b> 高频脉冲常数设置寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 4CH</b>							
	<b>Bit15</b>	<b>Bit14...Bit0</b>						
<b>Read:</b>	X	HFCnst[14:0]						
<b>Write:</b>	X	HFCnst[14:0]						
<b>Reset:</b>	0	0	0	0	0	0	0	0

Default 值: 0x0080

HFCnst 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 HFCnst 的值, 那么就会有对应的 PF/QF/SF 脉冲输出。

**ADCCFG(计量 ADC 配置寄存器)**

<b>ADCCFG</b> 计量 ADC 配置寄存器			<b>基地址: 0x40013800</b> <b>偏移地址: 50H</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>	PQSStart	PQSStart	Vref_En	Add _CIADD	WaveSel	TampSrc Sel	IPTampS rcSel	CHNFix
<b>Write:</b>	SrcSel1	SrcSel0						
<b>Reset:</b>	0	0	1	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>	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												
PQSStartSrcSel[1:0]	<b>PQS 起动潜动比较源头控制位</b> <table border="1" data-bbox="507 869 1332 1041"> <tr> <td>PQSStartSrcSel1</td> <td>PQSStartSrcSel0</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>	PQSStartSrcSel1	PQSStartSrcSel0	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
PQSStartSrcSel1	PQSStartSrcSel0	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> 0: 关闭 Vref; 1: 使能 Vref;												
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 波形数据的源头选择控制位</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;												
TampSrcSel	<b>自动防窃电比较源头选择控制位</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;												
IPTampSrcSel	<b>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/功率值; 1: 选择阈值源头为快速有效值/功率值;												
CHNFix	<b>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>防窃电阈值判定依据</b> 0: 决定 IPTamp 寄存器是选择有效值做为防窃电阈值。												

	1: 决定 IPTamp 寄存器是选择功率做为防窃电阈值。																								
TampSel	<b>防窃电比较源头选择位</b> 0: 选择电流有效值做防窃电 1: 选择有功功率做防窃电																								
FLTON	<b>自动防窃电模块开关位</b> 0: 自动防窃电关闭 1: 自动防窃电开启																								
CHNSEL	<b>计量通道选择位</b> 0: 通道 1 计量 1: 通道 2 计量																								
CIADD	<b>单相三线累加模式选择位</b> 0: 单通道方式 1: 功率累加和模式																								
SPL[2:0]	<b>波形采样中断频率选择控制位</b> 当 Femu=819.2K 时, 选择的频率如下: <table border="1" data-bbox="507 869 1241 1128"> <thead> <tr> <th>SPL2</th> <th>SPL1</th> <th>SPL0</th> <th>波形采样频率</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1.6k Hz (femu/512)</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>3.2k Hz (femu/256)</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>6.4k Hz (femu/128)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>12.8k Hz (femu/64)</td> </tr> <tr> <td>1</td> <td>x</td> <td>X</td> <td>12.8k Hz (femu/64)</td> </tr> </tbody> </table> <p>注: 当 femu=409.6KHz 时, 选择的波形采样频率与上表相比等比例调整即可。</p>	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 (femu/64)	1	x	X	12.8k Hz (femu/64)
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 (femu/64)																						
1	x	X	12.8k Hz (femu/64)																						

注: 功率相加模式下, 每路通道用各自的校表参数数据。在 FLTON=1, 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	CHSTS	能量累加
1	X	X	指示自动防窃电的通道选择结果	根据 CHSTS 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

## CHNLCCR(计量通道控制寄存器)

<b>CHNLCCR</b> 计量通道控制寄存器		<b>基地址: 0x40013800</b> <b>偏移地址: 54H</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>	PDatacp	QDatacp	SDatacp	Ipeak_En	Udet_En	RmsLpfEn	PQLpfEn	Add_Dat
<b>Write:</b>	En	En	En					aCP
<b>Reset:</b>	0	0	0	0	0	1	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>	LPMode	POS	LP	UconstEn	PSSel	HPFONI	HPFONI	HPFON
<b>Write:</b>			FreqSel			2	1	U
<b>Reset:</b>	0	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> 1: 使能常数计量 0: 关闭常数计量 (default)
QDatacpEn	<b>无功功率常数计量使能位</b> 1: 使能常数计量 0: 关闭常数计量 (default)
SDatacpEn	<b>视在功率常数计量使能位</b> 1: 使能常数计量 0: 关闭常数计量 (default)
Ipeak_En	<b>电流通道过流检测使能位</b> 1:使能过流检测 0: 关闭过流检测
Udet_En	<b>电压通道 SAG/PEAK 检测使能</b> 1:使能 SAG/PEAK 检测 0: 关闭 SAG/PEAK 检测
RmsLpfEn	<b>有效值低通使能控制位</b> 1: 开启有效值通道的低通滤波器(default) 0: 关闭有效值通道的低通滤波器
PQLpfEn	<b>功率低通使能控制位</b> 1: 开启功率通道的低通滤波器(default) 0: 关闭功率通道的低通滤波器
Add_DataCP	<b>常数计量下, 能量累加速度选择位:</b> 1: 选择累加速度为 BS 位流速度 0: 选择累加速度为 32k 速度
LPMode	<b>EMU 低频计量模式配置位</b> 1: EMU 时钟频率切换到低频计量模式 0: femu 为高频计量模式, 根据 EMU_ClkCtrl 配置的高频计量频率
POS	<b>脉冲有效电平选择位</b> 1: 表示PF/QF/SF为低电平有效 0: 表示PF/QF/SF为高电平有效
LP FreqSel	<b>低功耗计量模式下时钟源选择位</b> 1: 表示选择225K时钟 (PLLOn) 0: 表示选择32K时钟 (OSC时钟, PLLOff)
UconstEn	<b>低功耗电压使能位</b>

	1: 使用Uconst寄存器和电流通道有效值计算视在功率 0: 使用电压通道有效值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> 1: PF口输出视在脉冲, SF口输出有功脉冲 0: PF口输出有功脉冲, SF口输出视在脉冲 此功能可适用于掉零线计量模式下, 在有功脉冲口输出视在脉冲的情况。
HPFONU	<b>U通道高通滤波器开关位</b> 1: 开启 0: 关闭
HPFONI2	<b>I2 高通滤波器开关位</b> 1: 开启 0: 关闭
HPFONI1	<b>I1 高通滤波器开关位</b> 1: 开启 0: 关闭

### EMUCON(能量累加控制寄存器)

<b>EMCON</b> 能量累加控制寄存器		<b>基地址: 0x40013800</b> <b>偏移地址: 58H</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>	CF1CFG	CF1CFG	CF2CFG	CF2CFG	CF3CFG	CF3CFG	AverPow	AverPow
<b>Write:</b>	0	0	0	1	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>	0	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

	1	1	P
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)
1	1	P	
AverPowerCal	<b>能量累加源头选择位</b> 1: 选择平均功率作为能量累加源头 0: 选择瞬时功率作为能量累加源头		
AverPowerSource	<b>当使能平均功率累加时 (AverPowerCal=1 时), 平均功率的源头选择:</b> 1: 选择快速平均功率作为平均功率累加源头 0: 选择慢速平均功率作为平均功率累加源头		
OSR128	<b>正常计量模式下 OSR 选择位:</b> 1: OSR =128 0: OSR=64 仅对正常计量模式起作用, 低功耗计量模式下无用		
SRun	<b>视在能量累加使能位</b> 1: 允许视在计量 0: 停止视在计量		
QRun	<b>无功能量累加使能位</b> 1: 允许无功计量 0: 停止无功计量		
PRun	<b>有功能量累加使能位</b> 1: 允许有功计量 0: 停止有功计量		
QMOD[1: 0]	<b>无功能量累加模式选择位</b>		
	<b>QMOD1</b>	<b>QMOD0</b>	<b>累加功率Qm Qn</b>
	0	0	Qm=DataQ Qn=0
	0	1	DataQ ≥ 0, Qm=DataQ; DataQ<0, Qm=0 Qn=0
	1	0	Qm= DataQ  Qn=0
1	1	DataQ ≥ 0, Qm=DataQ DataQ<0, Qn= DataQ	
PMOD[1: 0]	<b>有功能量累加模式选择位</b>		
	<b>PMOD1</b>	<b>PMOD0</b>	<b>累加功率Pm Pn</b>
	0	0	Pm=DataP Pn=0
	0	1	DataP ≥ 0, Pm=DataP;
	1	0	Pn=DataP

			DataP<0,Pm=0 Pn=0	率, 不累加负向有功功率
	1	0	Pm= DataP  Pn=0	脉冲能量按绝对值方式对有功功率累加
	1	1	DataP≥0, Pm=DataP DataP<0,Pn= -DataP	脉冲能量对正向、反向有功功率分别累加

### PFCnt/ QFCnt / SFCnt (快速脉冲计数寄存器)

<b>PFCnt</b> 快速有功脉冲计数寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 5CH</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	PFCNT[15:0]							
<b>Write:</b>	PFCNT[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>QFCnt</b> 快速无功脉冲计数寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 60H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	QFCNT[15:0]							
<b>Write:</b>	QFCNT[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>SFCnt</b> 快速视在脉冲计数寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 64H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	SFCNT[15:0]							
<b>Write:</b>	SFCNT[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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。

### ADCCON (ADC 通道增益控制寄存器)

<b>ADCCON</b> ADC 通道增益控制寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 68H</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	DGI3	DGI2	DGI1	DGI0	DGU1	DGU0
<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>	CFP1	CFP0	PGA3	PGA2	PGA1	PGA0	UPGA1	UPGA0
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述					
DGI[3:0]	<b>电流通道数字增益选择位</b>					
	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]	<b>电压通道数字增益选择位</b>					
	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]	<b>脉宽选择位</b>					
	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]	<b>电流通道模拟增益选择位</b>					
	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]	<b>电压通道模拟增益选择位</b>					
	UPGA1		UPGA0		电压通道增益	
	0		0		PGA=1	
	0		1		PGA=2	
	1		0		PGA=4	
	1		1		PGA=8	

## IPTAMP(窃电检测阈值寄存器)

<b>IPTAMP</b> 窃电检测阈值寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 6CH</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	IPTAMP[15:0]							
<b>Write:</b>	IPTAMP[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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**;

当选择窃电阈值为电流有效值时，对应有有效值寄存器的 **Bit19—Bit4**。

### 自动防窃电处理模块开启时:

若选择使用电流有效值做为防窃电阈值的判断，当通道 1 和 2 的电流有效值都低于 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。

若选择使用功率 P 的绝对值做为防窃电阈值的判断，当 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 (窃电阈值设置寄存器)

<b>IChk</b> 窃电阈值设置寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 70H</b>							
	<b>Bit7...Bit0</b>							
<b>Read:</b>	IChk[7:0]							
<b>Write:</b>	IChk[7:0]							
<b>Reset:</b>	0	0	0	1	0	0	0	0

窃电阈值电流寄存器采用二进制补码形式，表示范围(0,+1)，计算公式如下：

$$ICHK=ICK7*2^{(-1)} + ICK6*2^{(-2)} + ICK5*2^{(-3)} + \dots + ICK1*2^{(-7)} + ICK0*2^{(-8)}$$

当选择电流有效值作为防窃电判断源头时， $|I2Rms-I1Rms|/I1Rms$  或  $|I2Rms-I1Rms|/I2Rms$  大于 ICHK 时，置位窃电标志。

当选择功率作为防窃电判断源 $|PowerP2-PowerP1|/|PowerP1|$ 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	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1000 0000	0100 0000	0010 0000	0001 0000	0000 1000	0000 0100	0000 0010	0000 0001
0.5	0.25	0.125	0.0625	0.03125	0.015625	0.007813	0.003906

例如：Check Register=0x1A 时，窃电阈值为 0.0625+0.03125+0.007813=10.1563%

默认为：Check Register=0x10，也即 0.0625 或 6.25%。

### 开启自动防窃电后:

当选择电流有效值作为防窃电的源头时, I1 和 I2 两者之差比,即  $(|I2Rms-I1Rms|/I1Rms$  或者  $|I2Rms-I1Rms|/I2Rms)$  超过窃电阈值电流值, 则自动选择大的电流值参与功率计量, 同时 TAMP=1。如果 I2 大于 I1, 则将标志位 I2GTI1 置为 1, 否则标志位 I2GTI1 为 0。

当选择功率作为防窃电的源头时, PowerP1 和 PowerP2 两者之差比  $(|PowerP2-PowerP1|/|PowerP1|$  或  $|PowerP2-PowerP1|/|PowerP1|)$  超过窃电值, 则自动选择大的功率值参与功率计量, 同时 TAMP=1。

## EMUCTRL (EMU 和 ADC 控制寄存器)

EMUCTRL EMU 和 ADC 控制寄存器			基地址: 0x40013800 偏移地址: 74H					
	Bit15	14	13	12	11	10	9	Bit8
Read:				Vref_LP_En	P2EN	EMU_ClkCtrl	DC PRFCFG	DC PRFCFG
Write:	X	X	StartSel				1	0
Reset:	0	0	0	0	0	0	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	SlowPR	Adc_i1o	Adc_i2o	Adc_uon	Zxd1	Zxd0	PRFCFG	PRFCFG
Write:	MS	n	n				1	0
Reset:	1	0	0	0	0	0	1	0

位	功能描述																				
StartSel	<b>防潜方式选择位</b> 1: PQ 联合判断潜动, 即至少满足 $ P >PStar$ 和 $ Q >QStar$ 任一条件, PQ 都会启动计量 0:  P 根据 PStart 判断潜动,  Q 根据 QStart 判断潜动(default)																				
Vref_LP_En	<b>低功耗计量基准使能位</b> 1: 开启 0: 关闭 此位与 LDO_EMU_En、EMU_En、Vref_LP_En 共同起作用, 详细见下表: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>EMU_En</th> <th>LDO_EMU_En</th> <th>Vref_LP_En</th> <th>Vref_LP 输出</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>X</td> <td>0</td> </tr> <tr> <td>X</td> <td>0</td> <td>X</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	EMU_En	LDO_EMU_En	Vref_LP_En	Vref_LP 输出	0	X	X	0	X	0	X	0	1	1	0	0	1	1	1	1
EMU_En	LDO_EMU_En	Vref_LP_En	Vref_LP 输出																		
0	X	X	0																		
X	0	X	0																		
1	1	0	0																		
1	1	1	1																		
P2EN	<b>P2 替换 Q1 计量通道使能位</b> 1: P2 0: Q1																				
EMU_ClkCtrl	<b>EMU 时钟控制位:</b> 1: femu 为 409.6K (需要注意脉冲宽度)																				

	0: femu 为 819.2K															
DC PRFCFG[1:0]	<b>直流平均值更新速度控制寄存器</b> <table border="1"> <thead> <tr> <th>PRFCFG1</th> <th>PRFCFG0</th> <th>有效值更新速度 (EMU 频率 = 819.2kHz)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>25Hz</td> </tr> <tr> <td>0</td> <td>1</td> <td>12.5Hz</td> </tr> <tr> <td>1</td> <td>0</td> <td>6.25Hz</td> </tr> <tr> <td>1</td> <td>1</td> <td>3.125Hz (default)</td> </tr> </tbody> </table>	PRFCFG1	PRFCFG0	有效值更新速度 (EMU 频率 = 819.2kHz)	0	0	25Hz	0	1	12.5Hz	1	0	6.25Hz	1	1	3.125Hz (default)
PRFCFG1	PRFCFG0	有效值更新速度 (EMU 频率 = 819.2kHz)														
0	0	25Hz														
0	1	12.5Hz														
1	0	6.25Hz														
1	1	3.125Hz (default)														
SlowPRMS	<b>慢速有效值&amp;功率更新速度控制位</b> 1: 在快速的基础上降低到 1/4 (default) 0: 在快速的基础上降低到 1/2															
Adc_i1on	<b>I1_ADC 使能位</b> 1: 打开第一路电流 ADC (default) 0: 关闭第一路电流 ADC															
Adc_i2on	<b>I2_ADC 使能位</b> 1: 打开第二路电流 ADC 0: 关闭第二路电流 ADC(default)															
Adc_uon	<b>U_ADC 使能位</b> 1: 打开电压 ADC 0: 关闭电压 ADC(default)															
Zxd[1:0]	<b>电压电流过零中断方式选择控制位</b> <table border="1"> <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"> <thead> <tr> <th>PRFCFG1</th> <th>PRFCFG0</th> <th>有效值更新速度 (EMU 频率 = 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 频率 = 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 频率 = 819.2kHz)														
0	0	20Hz														
0	1	10Hz														
1	0	5Hz (default)														
1	1	2.5Hz														

## P1OFFSET(通道 1 有功偏执校正寄存器)

<b>P1OFFSET</b>	<b>基地址: 0x40013800</b>
通道 1 有功偏执校正寄存器	<b>偏移地址: 78H</b>
	<b>Bit15...Bit0</b>
<b>Read:</b>	P1OFFSET[15:0]
<b>Write:</b>	
<b>Reset:</b>	0 0 0 0 0 0 0 0

计量通道 1 有功功率校正寄存器采用二进制补码形式。P1OFFSETH 与 32 位寄存器 PowerP1 的 bit[18..3]

位对齐，公式如下：

$$P1\_offset = \frac{Preal * EC * HFConst * 2^{28} * (-Err\%)}{2.304 * 10^{10}}$$

### P2OFFSET(通道 2 有功偏执校正寄存器)

<b>P2OFFSET</b> 通道 2 有功偏执校正寄存器	<b>基地址： 0x40013800</b>							
	<b>偏移地址： 7CH</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	P2OFFSET[15:0]							
<b>Write:</b>	P2OFFSET[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

计量通道 2 有功功率校正寄存器采用二进制补码形式。P2OFFSETH 与 32 位寄存器 PowerP2 的 bit[18..3] 位对齐，公式如下：

$$P2\_offset = \frac{Preal * EC * HFConst * 2^{28} * (-Err\%)}{2.304 * 10^{10}}$$

### Q1OFFSET(通道 1 无功偏执校正寄存器)

<b>Q1OFFSET</b> 通道 1 无功偏执校正寄存器	<b>基地址： 0x40013800</b>							
	<b>偏移地址： 80H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	Q1OFFSET[15:0]							
<b>Write:</b>	Q1OFFSET[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

计量通道 1 无功功率校正寄存器采用二进制补码形式。Q1OFFSETH 与 32 位寄存器 PowerQ1 的 bit[18..3] 位对齐，公式如下：

$$Q1\_offset = \frac{Qreal * EC * HFConst * 2^{28} * (-Err\%)}{2.304 * 10^{10}}$$

### Q2OFFSET(通道 2 无功偏执校正寄存器)

<b>Q2OFFSET</b> 通道 2 无功偏执校正寄存器	<b>基地址： 0x40013800</b>							
	<b>偏移地址： 84H</b>							
	<b>Bit15...Bit0</b>							

<b>Read:</b>	Q2OFFSET[15:0]							
<b>Write:</b>	Q2OFFSET[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

计量通道 2 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 bit[18..3]位对齐，公式如下：

$$Q2\_offset = \frac{Q_{real} * EC * HFConst * 2^{28} * (-Err\%)}{2.304 * 10^{10}}$$

## I1RMSOFFSET(I1 有效值偏执校正寄存器)

<b>I1RMSOFFSET</b> I1 有效值偏执校正寄存器	<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 88H
	<b>Bit15...Bit0</b>
<b>Read:</b>	I1RMSOFFSET[15:0]
<b>Write:</b>	I1RMSOFFSET[15:0]
<b>Reset:</b>	0 0 0 0 0 0 0 0 0

I1 有效值校正寄存器采用二进制无符号数形式。计算公式为：

$$I1RMSOFFSET = (I1RMS^2) / (2^{15})$$

当输入信号为 0 的时候，多次读取 I1RMS，取平均值后，然后按照上面的公式计算。

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

## I2RMSOFFSET(I2 有效值偏执校正寄存器)

<b>I2RMSOFFSET</b> I2 有效值偏执校正寄存器	<b>基地址:</b> 0x40013800 <b>偏移地址:</b> 8CH
	<b>Bit15...Bit0</b>
<b>Read:</b>	I2RMSOFFSET[15:0]
<b>Write:</b>	I2RMSOFFSET[15:0]
<b>Reset:</b>	0 0 0 0 0 0 0 0 0

I2 有效值校正寄存器采用二进制无符号数形式。计算公式为：

$$I2RMSOFFSET = (I2RMS^2) / (2^{15})$$

当输入信号为 0 的时候，多次读取 I2RMS，取平均值后，然后按照上面的公式计算。

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

**URMSOFFSET(电压有效值偏执校正寄存器)**

<b>URMSOFFSET</b> 电压有效值偏执校正寄存器		<b>基地址: 0x40013800</b> <b>偏移地址: 90H</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	URMSOFFSET[15:0]							
<b>Write:</b>	URMSOFFSET[15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

电压有效值校正寄存器采用二进制无符号数形式。计算公式为：

$$URMSOFFSET = (URMS^2) / (2^{15})$$

当输入信号为 0 的时候，多次读取 URMS，取平均值后，然后按照上面的公式计算。

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

**Rosi\_ctrl register(罗氏线圈控制寄存器)**

<b>Rosi_ctrl register</b> 罗氏线圈控制寄存器		<b>基地址: 0x40013800</b> <b>偏移地址: 94H</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>	Auto_Tc	Clkdelay2	ClkDelay1	Clkdelay0	AdcI_ctrl	AdcI_ctrl	AdcI_ctrl	DC_En
<b>Write:</b>	_EN				2	1	0	
<b>Reset:</b>	0	1	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>	DC_S_E	XboFiter	LpIdleEn	LpStaCtrl			Rosi_i2_	Rosi_i1_
<b>Write:</b>	n	50Hz 60Hz			LpCtrl1	LpCtrl0	en	en
<b>Reset:</b>	0	0	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> 1: 自动温度补偿功能开启，开启后，在不同的温度点补偿值会在计量参数寄存器中给出当前温度点下的补偿值。 0: 自动温度补偿功能关闭 (default)
Clkdelay[2:0]	<b>模拟数字之间的 clkdelay</b>

Adci_ctr[2:0]	<p><b>ADC 工作电流控制位</b></p> <table border="1" data-bbox="406 235 1165 448"> <tr> <th>Adci_ctr[1]</th> <th>Adci_ctr[0]</th> <th>偏置电流</th> </tr> <tr> <td>0</td> <td>0</td> <td>10uA (default)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1.5 uA</td> </tr> <tr> <td>1</td> <td>0</td> <td>3 uA</td> </tr> <tr> <td>1</td> <td>1</td> <td>5 uA</td> </tr> </table> <p>ADC 的工作电流越大,相应 ADC 的功耗越大; 在低频 32K 计量模式下, 推荐用户配置为[01],用偏置电流 1.5uA.</p>	Adci_ctr[1]	Adci_ctr[0]	偏置电流	0	0	10uA (default)	0	1	1.5 uA	1	0	3 uA	1	1	5 uA
Adci_ctr[1]	Adci_ctr[0]	偏置电流														
0	0	10uA (default)														
0	1	1.5 uA														
1	0	3 uA														
1	1	5 uA														
DC_En	<p><b>直流计量使能位</b></p> <p>1: 使能直流平均值计算功能, 计量参数的直流平均值寄存器 (地址 78H\7CH\80H) 有效</p> <p>0: 关闭直流平均值计算功能, 计量参数的直流平均值寄存器 (地址 78H\7CH\80H) 无效</p>															
DC_S_En	<p><b>直流 S 使能控制位</b></p> <p>1: 使能直流 S 功能, 即两路 PowerS 均采用直流平均值 U I1 I2 (计量寄存器地址 78H\7CH\80H) 进行直接计算, S 脉冲与能量均为直流量</p> <p>0: 关闭直流 S 功能, 视在功率 PowerS 来源于 Urms 和 Irms 的相乘。</p>															
XboFiter 50Hz 60Hz	<p><b>陷波滤波器频率控制位</b></p> <p>1: 陷波中心频率为 60Hz</p> <p>0: 陷波中心频率为 50Hz (default)</p>															
LpIdleEn	<p><b>低功耗计量分时开启使能控制位:</b></p> <p>1: 低功耗计量为分时模式, 当 LpIdleTime = 0 时, 空闲时间为 1s</p> <p>0: 低功耗计量为连续模式</p>															
LpStaCtrl	<p><b>低功耗计量模式下, 稳定点数控制位</b></p> <p>1: 固定丢 1024 个点</p> <p>0: 固定丢 512 个点 (default)</p>															
LpCtrl[1:0]	<p><b>低功耗计量模式下, 平均时间控制位</b></p> <table border="1" data-bbox="406 1332 1316 1534"> <tr> <th>LpCtrl [1]</th> <th>LpCtrl [0]</th> <th>平均时间 (s)</th> </tr> <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> </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	<p><b>I2的罗氏线圈使能位</b></p> <p>1: 使能电流通道2罗氏线圈功能</p> <p>0: 关闭电流通道2罗氏线圈功能</p>															
Rosi_i1_en	<p><b>I1的罗氏线圈使能位</b></p> <p>1: 使能电流通道1罗氏线圈功能</p> <p>0: 关闭电流通道1罗氏线圈功能</p>															

**ANA\_control register(模拟控制寄存器)**

ANA_control register 模拟控制寄存器			基地址: 0x40013800 偏移地址: 98H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	ldo_emu	X	Vref_	Vref_	Vref_	Chop_	RSRV	Adc_Cho
Write:	_en		ctrl2	ctrl1	ctrl0	Vref_en		p_En
Reset:	0	0	1	0	0	1	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	uchop_en	RSRV	RSRV	RSRV	RSRV	RSRV	Vref_Ch	lowFreq_
Reset:	1	0	1	0	0	1	op_buf_e	isel
							n	

位	功能描述
ldo_emu_en	<b>EMU Analog 供电 LDO 使能控制位:</b> 1: 开启 EMU 的 LDO 0: 关闭 EMU 的 LDO (default) 进入低功耗模式不需要开启 EMU 模块可关闭此位, 或常数计量掉零线进入 hold 模式时关闭此位, 以降低功耗。
Vref_Ctrl[2:0]	<b>VrefTC 顶点调整</b> Default = 100
Chop_Vref_en	<b>Vref chop 使能控制位</b> 1: Vref chop on (default) 0: Vref chop off
RSRV	建议保持此位为 1.
Adc_Chop_En	<b>Adc chop 使能控制位</b> 1: 使能 Adc chop 0: 关闭 Adc chop (default)
uchop_en	电压采样的 chop 使能位 1: 使能 U 通道 chop 功能 (default) 0: 关闭 U 通道 chop 功能 建议在正常计量模式下保持 U 通道的 chop 功能开启, 在掉零线计量模式下, 因 U 通道关闭, 需关闭 U 通道的 chop 功能。
RSRV[BIT6,BIT2]	建议保持默认值配置: 01001
Vref_Chop_buf_en	<b>Vref chop buf使能控制位</b> 1: 开启 0: 关闭
lowFreq_isel	<b>LowFreq偏置电流控制位</b> 1: 偏置电流为5Ua 0: 偏置电流为10Ua (default)

## UCONST(失压计量电压常量寄存器)

<b>UCONST</b> 失压计量电压常量寄存器		<b>基地址: 0x40013800</b>						
		<b>偏移地址: 9CH</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	UCONST[15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

此寄存器采用二进制无符号数形式。该寄存器主要目的是替代 Urms 寄存器的值参与掉零线计量，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  $2^7$ ，就可以得到 Uconst 的值

## LpIdleTime(低功耗计量分时空闲时间设置)

<b>LpIdleTime</b> 低功耗计量分时空闲时间设置		<b>基地址: 0x40013800</b>						
		<b>偏移地址: A0H</b>						
		<b>Bit9...Bit0</b>						
<b>Read:</b>	LpIdleTime [9:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

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，再打开固定时间；

空闲时间可写范围 0---599，即空闲 1S--600S，再开固定时间。

注意在低功耗模式下，慢速有效值，慢速视在功率是没有值的。

当使能 LpIdleEn 低功耗分时模式之后，并进入低功耗计量模式，LpIdleTime(A0H)默认为 0 即 1S。当设置 A0H 为 0x0009 就是 10 秒。以此为例，在这种配置下，ADC 采样一次值，FastI1RMS 和 FastPowerS 寄存器的值保持 10 秒，并按照这个功率值去累加脉冲，然后再去采一次，再保持 10S 这样。

## USAGLVL(电压跌落检测阈值寄存器)

<b>USAGLVL</b> 电压跌落检测阈值寄存器		<b>基地址: 0x40013800</b>						
		<b>偏移地址: A4H</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	USAGLVL[15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

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 USAGLVL 相比较，在用户设定的 OVCyc 个半周期时间内，如果 ADC 波形都小于 USAGLVL 寄存器，

则在 OVCyc 的最后一个半周波，更新 SAG 标志。

计量参数寄存器 Upd\_Half(88H)。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：

$$\text{Upd\_Half} = \text{Urms} * 1.414 / 2^5$$

USAGLVL 阈值如果设成峰值的 80%，则：

$$\text{USAGLVL} = 0.8 * \text{Upd\_Half} / 2^3 = 0.8 * \text{Urms} * 1.414 / 2^8$$

## IpeakCyc(I1 I2 PEAK 检测周期设置)

<b>IpeakCyc</b> I1 I2 PEAK 检测周期设置		<b>基地址： 0x40013800</b>						
		<b>偏移地址： A8H</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	<b>IpeakCyc [15:0]</b>							
<b>Write:</b>	<b>IpeakCyc [15:0]</b>							
<b>Reset:</b>	0	0	0	0	0	0	0	0

该寄存器默认值为 0x0020。注意：

16 位无符号数，用于设定过流检测数据长度，即 IPeak 功能检测设定的数据长度后给出峰值。

1 个 LSB 对应半个周波。

## UOVLVL(电压 PEAK 检测阈值寄存器)

<b>UOVLVL</b> 电压 PEAK 检测阈值寄存器		<b>基地址： 0x40013800</b>						
		<b>偏移地址： ACH</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	<b>UOVLVL[15:0]</b>							
<b>Write:</b>	<b>UOVLVL[15:0]</b>							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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。

Upd\_Half 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。UOVLVL 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：

$$\text{Upd\_Half} = \text{Urms} * 1.414 / 2^5$$

UOVLVL 阈值如果设成峰值的 120%，则：

$$\text{UOVLVL} = 1.2 * \text{Upd\_Half} / 2^3 = 0.8 * \text{Urms} * 1.414 / 2^8$$

### OVCyc(电压 SAG\PEAK 检测周期寄存器)

OVCyc 电压 SAG\PEAK 检测周期寄存器		基地址: 0x40013800 偏移地址: B0H						
		Bit15...Bit0						
Read:	OVCyc[15:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

该寄存器默认值为 0x0001。注意:

16 位无符号数, 用于设定 Udet 检测数据长度, 即电压通道 Peak/SAG 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。

### IOVLVL(电流过流阈值寄存器)

IOVLVL 电流过流阈值寄存器		基地址: 0x40013800 偏移地址: B4H						
		Bit15...Bit0						
Read:	IOVLVL[15:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

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 IpeakCyc 内, 如果 ADC 波形超过 IOVLVL 寄存器, 则在 IpeakCyc 的最后一个半周波, 更新 UOVIF 标志。

Ipeak 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。IOVLVL 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:

$$I_{peak} = I_{rms} * 1.414 / 2^5$$

IOVLVL 阈值如果设成峰值的 120%, 则:

$$IOVLVL = 1.2 * I_{peak} / 2^3 = 0.8 * I_{rms} * 1.414 / 2^8$$

### ZXILVL(电流过零阈值寄存器)

ZXILVL 电流过零阈值寄存器		基地址: 0x40013800 偏移地址: B8H					
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	ZX15	ZX4	ZX13	ZC12...ZC3	ZC2	ZC1	ZC0

<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

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，则不输出电流用户指定的过零信号，内部输出恒为 0。

### PDataCpH(常数有功功率累加高 16 位)

<b>PDataCpH</b> 常数有功功率累加高 16 位	<b>基地址: 0x40013800</b>							
	<b>偏移地址: BCH</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	PDataCpH [15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

### PDataCpL(常数有功功率累加低 16 位)

<b>PDataCpL</b> 常数有功功率累加低 16 位	<b>基地址: 0x40013800</b>							
	<b>偏移地址: C0H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	PDataCpL [15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

常数计量下，常数有功功率的高、低 16 位组成 32 位常数视在功率，当用户使能常数计量功能后，由 PDATA CP 代替有功功率发脉冲和能量累加。

### QDataCpH(常数无功功率累加高 16 位)

<b>QDataCpH</b> 常数无功功率累加高 16 位	<b>基地址: 0x40013800</b>							
	<b>偏移地址: C4H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	QDataCpH [15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

**QDataCpL(常数无功功率累加低 16 位)**

<b>QDataCpL</b> 常数无功功率累加低 16 位		<b>基地址: 0x40013800</b>						
		<b>偏移地址: C8H</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	QDataCpL [15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

常数计量下，常数无功功率的高低 16 位，组成 32 位常数视在功率，当用户使能常数计量功能后，由 QDATA CP 代替无功功率发脉冲和能量累加；

**SDataCpH(常数视在功率累加高 16 位)**

<b>SDataCpH</b> 常数视在功率累加高 16 位		<b>基地址: 0x40013800</b>						
		<b>偏移地址: CCH</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	SDataCpH[15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

**SDataCpL(常数视在功率累加低 16 位)**

<b>SDataCpL</b> 常数视在功率累加低 16 位		<b>基地址: 0x40013800</b>						
		<b>偏移地址: D0H</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	SDataCpL [15:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

常数计量下，常数有功功率的高低 16 位，组成 32 位常数视在功率，当用户使能常数计量功能后，由 SDATA CP 代替有功功率发脉冲和能量累加；

**Filter control register(滤波器系数控制寄存器)**

<b>Filter control register</b> 滤波器系数控制寄存器		<b>基地址: 0x40013800</b>						
		<b>偏移地址: D4H</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	RosiPga Ctr1	RosiPga Ctr0	ZXsourc eSel	PahseShi ftCtr1	PahseShi ftCtr1
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	1	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	RosiCtr1	RosiCtr0	HpfCtl2	HpfCtl1	HpfCtl0	LpfCtl2	LpfCtl1	LpfCtl0
<b>Write:</b>								
<b>Reset:</b>	1	1	1	1	0	1	1	1

位	功能描述																																				
RosiPgaCtr[1:0]	<p><b>两路电流罗氏线圈增益控制档</b></p> <table border="1"> <thead> <tr> <th>RosiPgaCtr1</th> <th>RosiPgaCtr0</th> <th>Rosi 增益</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1 (default)</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> </tr> <tr> <td>1</td> <td>0</td> <td>4</td> </tr> <tr> <td>1</td> <td>1</td> <td>8</td> </tr> </tbody> </table>	RosiPgaCtr1	RosiPgaCtr0	Rosi 增益	0	0	1 (default)	0	1	2	1	0	4	1	1	8																					
RosiPgaCtr1	RosiPgaCtr0	Rosi 增益																																			
0	0	1 (default)																																			
0	1	2																																			
1	0	4																																			
1	1	8																																			
ZXsourceSel	<p><b>电流电压过零源头选择位</b></p> <p>1: 选择电压过零为 90 度移相后, 电流过零为经 rosi 后 0: 选择电压电流过零源头为高通过后</p>																																				
PahseShiftCtr[1:0]	<p><b>90 度移相滤波器系数控制档</b></p> <table border="1"> <thead> <tr> <th>PahseShiftCtr1</th> <th>PahseShiftCtr0</th> <th>移相滤波器档位</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>11</td> </tr> <tr> <td>0</td> <td>1</td> <td>12</td> </tr> <tr> <td>1</td> <td>0</td> <td>13(default)</td> </tr> <tr> <td>1</td> <td>1</td> <td>14</td> </tr> </tbody> </table>	PahseShiftCtr1	PahseShiftCtr0	移相滤波器档位	0	0	11	0	1	12	1	0	13(default)	1	1	14																					
PahseShiftCtr1	PahseShiftCtr0	移相滤波器档位																																			
0	0	11																																			
0	1	12																																			
1	0	13(default)																																			
1	1	14																																			
RosiCtr[1:0]	<p><b>两路电流罗氏线圈滤波器系数控制档</b></p> <table border="1"> <thead> <tr> <th>RosiCtr1</th> <th>RosiCtr0</th> <th>Rosi 滤波器档位</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>11</td> </tr> <tr> <td>0</td> <td>1</td> <td>12</td> </tr> <tr> <td>1</td> <td>0</td> <td>13</td> </tr> <tr> <td>1</td> <td>1</td> <td>14(default)</td> </tr> </tbody> </table>	RosiCtr1	RosiCtr0	Rosi 滤波器档位	0	0	11	0	1	12	1	0	13	1	1	14(default)																					
RosiCtr1	RosiCtr0	Rosi 滤波器档位																																			
0	0	11																																			
0	1	12																																			
1	0	13																																			
1	1	14(default)																																			
HpfCtl[2:0]	<p><b>高通系数选择控制位</b></p> <table border="1"> <thead> <tr> <th>HpfCtl2</th> <th>HpfCtl1</th> <th>HpfCtl0</th> <th>高通系数</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>4</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>5</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>6</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>7</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>8</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>9</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>10 (default)</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>11</td> </tr> </tbody> </table>	HpfCtl2	HpfCtl1	HpfCtl0	高通系数	0	0	0	4	0	0	1	5	0	1	0	6	0	1	1	7	1	0	0	8	1	0	1	9	1	1	0	10 (default)	1	1	1	11
HpfCtl2	HpfCtl1	HpfCtl0	高通系数																																		
0	0	0	4																																		
0	0	1	5																																		
0	1	0	6																																		
0	1	1	7																																		
1	0	0	8																																		
1	0	1	9																																		
1	1	0	10 (default)																																		
1	1	1	11																																		

LpfCtl[2:0]	低通系数选择控制位			
	HpfCtl2	HpfCtl1	HpfCtl0	低通系数
	0	0	0	4
	0	0	1	5
	0	1	0	6
	0	1	1	7
	1	0	0	8
	1	0	1	9
	1	1	0	10
1	1	1	11 (default)	

**TUgain(U 通道手动温度补偿系数设置)**

<b>TUgain</b> U 通道手动温度补偿系数设置	基地址: <b>0x40013800</b> 偏移地址: <b>D8H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	TUgain [15:0]							
<b>Write:</b>	TUgain [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

**TI1gain(I1 通道手动温度补偿系数设置)**

<b>TI1gain</b> I1 通道手动温度补偿系数设置	基地址: <b>0x40013800</b> 偏移地址: <b>DCH</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	TI1gain [15:0]							
<b>Write:</b>	TI1gain [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

**TI2gain(I2 通道手动温度补偿系数设置)**

<b>TI2gain</b> I2 通道手动温度补偿系数设置	基地址: <b>0x40013800</b> 偏移地址: <b>E0H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	TI2gain [15:0]							
<b>Write:</b>	TI2gain [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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。

## UTCcoeffA(U 通道自动温补 VrefGain 的二次系数设置)

自动温度补偿曲线系数寄存器:

<b>UTCcoeffA</b> U 通道自动温补 VrefGain 的二次系数设置		<b>基地址: 0x40013800</b>						
		<b>偏移地址: E4H</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	UTCcoeffA [15:0]							
<b>Write:</b>	UTCcoeffA [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

U 通道自动温度补偿曲线二次项系数。

## UTCcoeffB(U 通道自动温补 VrefGain 的一次系数设置)

<b>UTCcoeffB</b> U 通道自动温补 VrefGain 的一次系数设置		<b>基地址: 0x40013800</b>						
		<b>偏移地址: E8H</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	UTCcoeffB [15:0]							
<b>Write:</b>	UTCcoeffB [15:0]							
<b>Reset:</b>	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  $UTCcoeffB = Bit15 \text{ (符号)} \text{ bit14—bit13} + bit12 \text{ (符号)} \text{ bit11—bit0}$

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  $UTCoffB = 0 + -0.75 = -0.75$

如果  $UTCoffB$  值为  $0x2400$  则系数为：整数部分为 1，小数部分为  $0x400/2^{12} = 0.25$

则  $UTCoffB = 1 + 0.25 = 1.25$

### UTCcoffC(U 通道自动温补 VrefGain 的常数项系数设置)

<b>UTCcoffC</b> U 通道自动温补 VrefGain 的常数项系数设置		<b>基地址: 0x40013800</b>						
		<b>偏移地址: ECH</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	UTCcoffC [15:0]							
<b>Write:</b>	UTCcoffC [15:0]							
<b>Reset:</b>	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，TPSCODE 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：

$$AutoUgain = UTCcoffA/32768 * (TPSCODE/256)^2 + UTCcoffB * (TPSCODE/256) + UTCcoffC/32768$$

注意：UTCcoffB 分为整数部分和小数部分，各自都有符号位，UTCcoffC 也是有符号数，表示范围 -32768--32767。

### I1TCcoffA(I1 通道自动温补 VrefGain 的二次系数设置)

<b>I1TCcoffA</b> I1 通道自动温补 VrefGain 的二次系数设置寄存器		<b>基地址: 0x40013800</b>						
		<b>偏移地址: F0H</b>						
		<b>Bit15...Bit0</b>						
<b>Read:</b>	I1TCcoffA [15:0]							
<b>Write:</b>	I1TCcoffA [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。

### I1TCcoffB(I1 通道自动温补 VrefGain 的一次系数设置)

<b>I1TCcoffB</b>	<b>基地址: 0x40013800</b>
------------------	------------------------

<b>I1 通道自动温补 VrefGain 的一次系数设置寄存器</b>	<b>偏移地址: F4H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	I1TCcoeffB [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。

### I1TCcoeffC(I1 通道自动温补 VrefGain 的常数项系数设置)

<b>I1TCcoeffC</b> <b>I1 通道自动温补 VrefGain 的常数项系数设置寄存器</b>	<b>基地址: 0x40013800</b>							
	<b>偏移地址: F8H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	I1TCcoeffC [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。

### I2TCcoeffA(I2 通道自动温补 VrefGain 的二次系数设置)

<b>I2TCcoeffA</b> <b>I2 通道自动温补 VrefGain 的二次系数设置寄存器</b>	<b>基地址: 0x40013800</b>							
	<b>偏移地址: FCH</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	I2TCcoeffA [15:0]							
<b>Write:</b>								
<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。

### I2TCcoeffB(I2 通道自动温补 VrefGain 的一次系数设置)

<b>I2TCcoeffB</b> <b>I1 通道自动温补 VrefGain 的一次系数设置寄存器</b>	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 100H</b>							

	<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。

### I2TCcoeffC(I2 通道自动温补 VrefGain 的常数项系数设置)

<b>I2TCcoeffC</b> I1 通道自动温补 VrefGain 的常数项系数设置寄存器	<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>	I2TCcoeffC [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。

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。

### LoadDataCp(常数计量启动寄存器)

<b>LoadDataCp</b> 常数计量启动寄存器	<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>	LoadDataCp [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

当用户向 LoadDataCp 写入 0x00BC 后，常数计量 DATACpload 进入累加源头。

### BufferStart(缓存数据启动寄存器)

<b>BufferStart</b> 缓存数据启动寄存器	<b>基地址: 0x40013800</b> <b>偏移地址: 10CH</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>	BufferStar[15:8]							
<b>Write:</b>	BufferStar[15:8]							

<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>	BufferStar[7:4]				X	X	X	X
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

注意：默认值:0x0000 缓存启动寄存器，缓存启动之后，该寄存器重新复位为 0x0000。

BIT15--4 启动位写入 0CCCH 启动 buffer 缓存

位	功能描述															
bit[3:2]	Buf dec gain															
	<table border="1"> <thead> <tr> <th>Bit3</th> <th>Bit2</th> <th>增益</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> </tr> <tr> <td>1</td> <td>0</td> <td>4</td> </tr> <tr> <td>1</td> <td>1</td> <td>8</td> </tr> </tbody> </table>	Bit3	Bit2	增益	0	0	1	0	1	2	1	0	4	1	1	8
	Bit3	Bit2	增益													
	0	0	1													
	0	1	2													
1	0	4														
1	1	8														
buffer 源头选择																
<table border="1"> <thead> <tr> <th>Bit1</th> <th>Bit0</th> <th>源头选择</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>U</td> </tr> <tr> <td>0</td> <td>1</td> <td>I1</td> </tr> <tr> <td>1</td> <td>0</td> <td>I2</td> </tr> <tr> <td>1</td> <td>1</td> <td>U</td> </tr> </tbody> </table>	Bit1	Bit0	源头选择	0	0	U	0	1	I1	1	0	I2	1	1	U	
Bit1	Bit0	源头选择														
0	0	U														
0	1	I1														
1	0	I2														
1	1	U														

## BufferCoff(缓存数据手动调整系数寄存器)

<b>BufferCoff</b> 缓存数据手动调整系数寄存器	<b>基地址: 0x40013800</b> <b>偏移地址: 110H</b>
	<b>Bit8...Bit0</b>
<b>Read:</b>	BufferCoff [8:0]
<b>Write:</b>	
<b>Reset:</b>	0 0 0 0 0 0 0 0 0

当采样信号为 50Hz 时，此寄存器为 0x0100(Default)

缓存数据手动调整系数 bit8-0 写入范围 196—284，大于 284 取值 284 小于 196 取值 196

系数计算方法：为实际线频率。

$$\text{BufferCoff} = \frac{\text{EMUCLK}}{\text{Freq} \times 64}$$

EMUCLK 为 819.2KHz 或 409.6KHz,

Freq 为信号的线频率，频率范围为 45Hz~65Hz 之间，超出这个范围就不适用了。

用户可手动调整此系数，以获得缓存数据为两个整周波信号。

**SRSTREG(EMU 软件复位寄存器)**

<b>SRSTREG</b> EMU 软件复位寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 114H</b>							
	<b>Bit7...Bit0</b>							
<b>Read:</b>	SRSTREG [7:0]							
<b>Write:</b>	SRSTREG [7:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

当用户写入 0x55，复位所有校表参数寄存器，当用户写入 0xAA 复位整个 EMU 模块。

**PFCntN(反向快速有功脉冲计数寄存器)**

<b>PFCntN</b> 反向快速有功脉冲计数寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 118H</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	PFCntN [15:0]							
<b>Write:</b>	PFCntN [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

默认值为 0x0000

**QFCntN(反向快速无功脉冲计数寄存器)**

<b>QFCntN</b> 反向快速无功脉冲计数寄存器	<b>基地址: 0x40013800</b>							
	<b>偏移地址: 11CH</b>							
	<b>Bit15...Bit0</b>							
<b>Read:</b>	QFCntN [15:0]							
<b>Write:</b>	QFCntN [15:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

当快速脉冲计数寄存器 PFCntN/QFCntN 计数的值大于等于 HFconst 时，相应的 PF/QF 会有脉冲溢出，能量寄存器的值会相应的加 1。为了防止上下电时丢失电能，掉电时寄存器 PFCntN/QFCntN 可保存不足一个脉冲的能量值，MCU 可将此值读回并进行保存，然后在下次上电时 MCU 将这些值重新写入到 PFCntN/QFCntN 中去参与能量累加。

## 23.7 校表过程

### 第一步：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。

#### 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：

$$\text{HFCONST} = 0x0080 \times (1 + \text{Err}\%)$$

#### 举例说明：

电表表常数 EC 设置为 3200imp/KWh，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： $\text{HFCONST} = 0x0080 \times (1 + \text{Err}\%)$

计算得到： $\text{HFCONST} = 0x0080 \times (1 + 52.8\%) = 0x00C3$

将 0x00C3 写入校表寄存器 HFCONST（4CH），写入校正值之后，标准表显示误差应该在 10% 以内。

#### 方案二：

ADC 采样频率  $f_s$  为 12.8KHz 时（即  $F_{emu} = 819.2\text{KHz}$ ， $OSR = 64$ ， $f_s = F_{emu}/OSR$ ），HFConst 的计算公式如下：

$$\text{HFConst} = \frac{2.332 \times V_u \times V_i \times 10^{10}}{EC \times U_n \times I_b}$$

$V_u$ ：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

当 ADC 的采样频率  $f_s$  为其他值时，HFConst 按等比例变化即可，如  $f_s$  缩小一倍，HFConst 的值等比例缩小一倍。

#### 举例说明：

电表表常数 EC 设置为 3200imp/KWh，功率因数为 1， $f_s$  保持默认值为 12.8KHz。

$U_n$ （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， $V_i \times 16 = 28\text{mV}$

根据公式：

$$\text{HFConst} = \frac{2.332 \times V_u \times V_i \times 10^{10}}{EC \times U_n \times I_b}$$

计算得到：

$\text{HFConst} = 2.332 \times 0.22 \times 0.028 \times 10^{10} / (3200 \times 220 \times 5) = 0x0028$

将 0x0028 写入校表寄存器 HFCONST（4CH），写入后标准表的显示误差应该在 10% 以内。

### 第二步：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：

$$Pgain = \frac{-err\%}{1 + err\%}$$

如果  $Pgain \geq 0$ ，则  $GP1 = INT[Pgain \times 2^{15}]$

否则  $Pgain < 0$ ，则  $GP1 = INT[2^{16} + Pgain \times 2^{15}]$

将计算出值写入有功增益寄存器 GP1，芯片内部会自动将此值写入到无功增益 GQ1 和视在增益 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  $Pgain = -0.038 / (1 + 0.038) = -0.0366$

该数小于 0，需转换为补码，则  $-0.0366 \times 2^{15} + 2^{16} = 0xFB50H$

将 0xFB50H 写入 GP1(0CH)寄存器，无功和视在阻性增益也同时得到校正，即完成所有阻性增益校正。

### 第三步：第一通道相位校正

在阻性增益已经校正好之后，进行相位增益校正，需要将功率因素调整到 0.5L 状态处进行校正。

已知：0.5L 处标准表误差读数为  $err\%$

相位补偿公式：

$$\theta = \frac{-err\%}{1.732}$$

$$\text{如果 } \theta \geq 0, GPhs = \theta \times 2^{15}$$

$$\text{否则 } \theta < 0, GPhs = 2^{16} + \theta \times 2^{15}$$

举例说明：

校完阻性增益后，将功率因数调整为 0.5L，如果这时标准表显示的误差为 -0.4%，则

$$\theta = (-0.004) / 1.732 = 0.0023$$

$$Gphs1 = 0.0023 \times 2^{15} = 75.3$$

取整后为 0x4BH，写入相位校正寄存器 Gphs1(18H)即可。

### 第四步：电流通路 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 的有效值或功率寄存器值应该相等。

通过调整通道 2 增益校正寄存器 I2GAIN(30H)，使外部输入相同负载的情况下，两个通道有效值或功率寄存器的值一致。

输入同样额定电流，电流通路 1 有效值寄存器读数为  $I_{1rms}$ ，电流通路 2 有效值寄存器读数为  $I_{2rms}$ ，电流通路 1 有功功率  $Power1$ ，电流通路 2 有功功率  $Power2$ ，

如果选择比较电流防窃电方式：

$$\text{则 } Gain = I_{1rms} / I_{2rms} - 1$$

如果选择比较功率防窃电方式：

则  $Gain = Power1/Power2 - 1$

如果  $Gain \geq 0$ ,  $I2Gain = Gain \times 2^{15}$ ; 如果  $Gain < 0$ ,  $I2Gain = Gain \times 2^{15} + 2^{16}$

第五步：第二通道阻性增益校正、相位校正

第二通道阻性增益校正、相位校正与通道 1 类似。

第六步：Poffset 校正（小信号有功功率偏置校正）

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  $P_{real}$

根据公式计算：

$$P1\_offset = \frac{P_{real} * EC * HFConst * 2^{28} * (-Err\%)}{2.304 * 10^{10}}$$

**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  $I_b = 100\%$  时误差校正到 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 ( $P_{real}$  为标准表上的显示功率输出)

根据公式计算得到：

$$\begin{aligned} Poffset &= (P_{real} * EC * HFCONST * 2^{28} * (-Err\%)) / (2.3 * 10^{10}) \\ &= (55.02 * 3200 * 102 * 2^{28} * (-0.5\%)) / (2.3 * 10^{10}) \\ &= -1047.98 \end{aligned}$$

由于  $Poffset < 0$ ，所以写入寄存器  $P1offset$  的值为  $2^{16} + Poffset = 63444$

取整数 64488 即 0xFBE8，

将 0xFBE8 写入通道一小信号偏置寄存器  $P1offset(78H)$ 。

**第七步：Irms、Urms 和两个通道的功率显示转换系数校正**

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 RMSI1（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 就是电流有效值显示值转换系数，之后用户可根据读取的寄存器 RMSI1 的值乘这个 K 系数，则得到实际外部输入的电流。详见有效值输出章节和功率参数输出章节。

## 24 电气规格

### 24.1 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
Vih1	高电平输入电压	Reset 引脚, TEST 引脚	0.8VCC			V
Vih2		PC9,PC10,PE0,PE3,PE6	0.6VCC			V
Vih3		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
Vil	低电平输入电压	除了电源和地之外的所有引脚			0.2VCC	V
Ioh1 大电流 IO	高电平输出电流 1	VCC=5.0V I/O 口上电压 Voh 降低到 0.9VCC 测试引脚为: PA.6, PA.7, PA.8, PC.0 (需要开启大电流驱动)		10		mA
Iol1 大电流 IO	低电平输出电流 1	VCC=5.0V I/O 口上电压 Vol 升高到 0.1VCC 测试引脚为: PA.6, PA.7, PA.8, PC.0 (需要开启大电流驱动)		20		mA
Ioh2	高电平输出电流 2	I/O 口上电压 Voh 降低到 0.9VCC 测试引脚为: 除了 PA.6, PA.7, PA.8, PC.0, PE.9,PC.13,PC.14,PE.2, PE.4,PC.8,PA.9,PA.10, PE.0,PE.6,PC.12,PA.5 之外的其他 IO 口。		5(VCC=5.0V) 2.5(VCC=3.3V)		mA
Iol2	低电平输出电流 2	I/O 口上电压 Vol 降低到 0.1VCC 测试引脚为: 除了 PA.6, PA.7, PA.8, PC.0, PE.9,PC.13,PC.14,PE.2, PE.4,PC.8,PA.9,PA.10,		5(VCC=5.0V) 2.5(VCC=3.3V)		mA

		PE.0,PE.6,PC.12,PA.5 之外的其他 IO 口。				
Ioh3	高电平输出电流 3	VCC=5.0V I/O 口上电压 Voh 降低到 0.9VCC 测试引脚为： PE.9,PC.13,PC.14,PE.2, PE.4,PC.8,PA.9,PA.10, PE.0,PE.6,PC.12,PA.5		3(VCC=5.0V) 1.5(VCC=3.3V)		mA
Iol3	低电平输出电流 3	VCC=5.0V I/O 口上电压 Vio 降低到 0.1VCC 测试引脚为： PE.9,PC.13,PC.14,PE.2, PE.4,PC.8,PA.9,PA.10, PE.0,PE.6,PC.12,PA.5		3(VCC=5.0V) 1.5(VCC=3.3V)		mA

## 24.2 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	VCC	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	5.0	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)	ADC 输入引脚	-6000	+6000	V
		芯片所有的引脚	-8000	+8000	

注：VCC 输入的系统电压可承受 1.5 倍标称电压。

## 24.3 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	-45°C	2.2		uA

户在 Hold 模式下打开不同的数字模块而不同)	常温	1	3.9	11	
	85°C		34.3		
芯片处于 Sleep 模式下最低功耗	-45°C		1.9		uA
	常温	1	2.9	6.9	
	85°C		16.0		
芯片在低频 32768Hz 下运行, LCD 开启时的功耗	常温		48		uA
Hold 下, LCD 开启功耗	常温		12.2		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	常温		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	常温		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	常温		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	常温		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	常温		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	常温		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	常温		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	常温		8.76 /5.64		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	常温		5.78 /4.03		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	常温		3.63 /2.69		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	常温		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	常温		1.96 /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	常温		1.68 /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	常温		1.54 /0.39		mA
EMU 单路 ADC 功耗 (偏置电流 1.5uA 时)	常温		130		uA

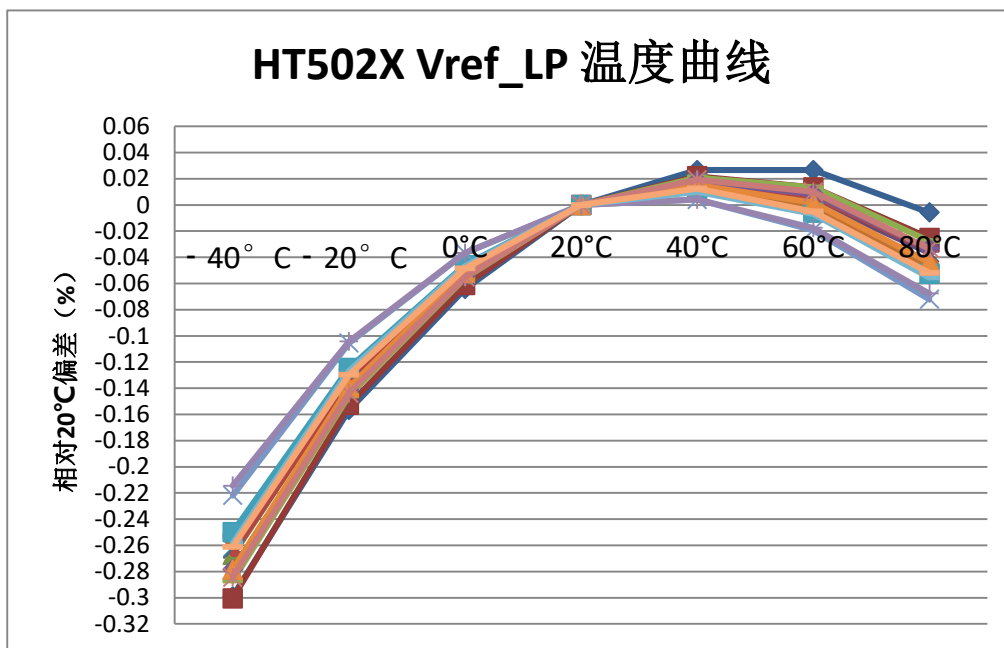
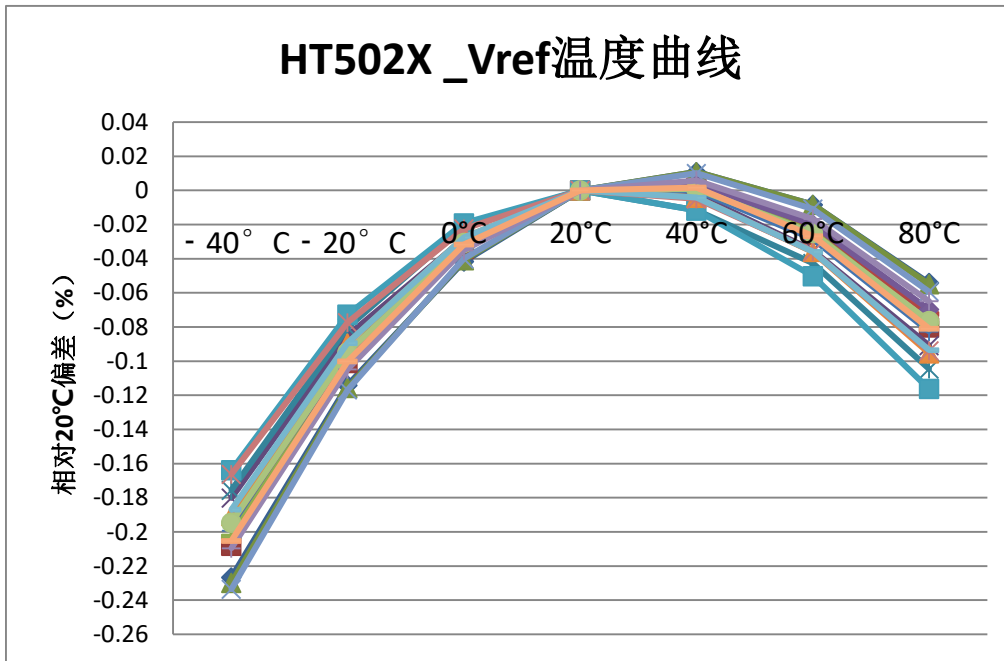
EMU 单路 ADC 功耗 (偏置电流 10uA 时)	常温		600		uA
-----------------------------	----	--	-----	--	----

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)		116		uA
低频 RC 模块功耗 (LRC)		0.9		uA
PLL 模块功耗		205		uA
LCD 模块功耗 (快速充电模式)		6.9		uA
LCD 模块功耗 (小电流模式)		8.3		uA
LCD 模块功耗 (大电流模式)		35.2		uA
BOR 模块功耗		8		uA
LVDIN 模块		7		uA

## 24.4 计量 ADC 参数

符号	参数说明	测试条件 (常温)	最小	典型	最大	单位
V3P/V3N	电压通道		0		±1000	mVp
V1P/V1N	电流 1 通道		0		±800	mVp
V2P/V2N	电流 2 通道		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	计量基准电压			1.20000		V
VrefTC	计量基准温度系数		10	15	20	ppm/°C
VrefLP	辅助计量基准电压			1.20000		V
VrefLPTC	辅助计量基准温度系数		15	20	30	ppm/°C

附：基准温度漂移曲线



## 24.5TBS 模块 ADC 参数

符号	参数说明	测试条件	最小	典型	最大	单位
VCC	工作电压		2.4		5.5	V
Iadc	工作电流			350		uA
Res	分辨率				16	Bit
ENOB	有效位			12		Bit
Vrefadc	参考电压	常温	-2%	1.258V	+2%	

	(内置)					
VREF TC	参考电压温度系数			100ppm		/°C
Fsample	工作频率			32.768k		Hz
Cin	输入电容			0.515		pF
Rin	输入电阻 (等效)			30M		Ω
ADCIN <sub>x</sub> 通道						
Vadcin	ADC 输入范围	Vrefadc=1.258V, 常温	0		800	mV
Resadcin	分辨率	理论		0.0128		mV/LSB
	计算公式	实际拟合	$VADCIN0 = 0.0128 * ADC0DAT + 425.5623;$ 其中: VADCIN0 为实际 ADC 测量电压 (mV)			mV
温度传感器 TPS 通道						
Adjtps	温度检测精度			+1		°C
Restps	分辨率	理论		0.00278		°C/LSB
	计算公式		$Tr = 12.9852 - TMPDAT * 0.0028$ 其中: Tr 为实际的温度 (°C)			°C
VBAT 通道						
Vbatin	VBAT 输入范围		1.0		5.5	V
Resvbat	分辨率	理论		0.08958		mV/LSB
Rinbat	VBAT 输入电阻 (内置)			220k		Ω
Div	分压系数 (内置)			1/7		
	计算公式	实际拟合	$VBAT = 0.0897 * VBATDAT + 2946.2662;$ 其中: VBAT 为实际电池电压 (mV)			mV
VDD 通道						
VDDin	VBAT 输入范围		2.4		5.5	V
Restps	分辨率	理论		0.08958		mV/LSB
Rinvdd	VBAT 输入电阻 (内置)			42k		Ω
Div	分压系数 (内置)			1/7		
	计算公式	实际拟合	$VDD = 0.0897 * VDDDAT + 2998.1189;$ 其中: VDD 为实际 VDD 电压 (mV)			mV

注: 使用 VBAT 和 VDD 通道, 工作电流为 ADC 工作电流和输入电阻对地电流之和

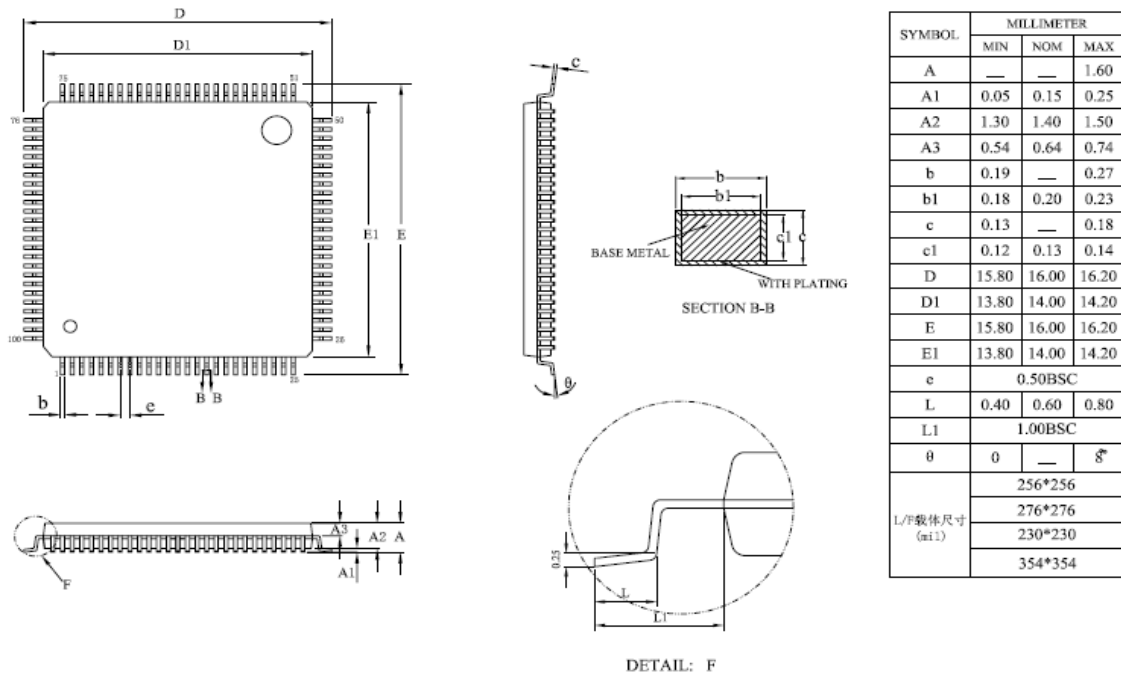
## 24.6 内部 RC 模拟参数

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
电源电压	VCC	@PVT	1.2	5	5.5	V
最低可操作电压		@TT		1		V
静态功耗	Icc			400	600	nA
LRC 全温度输出频率			13	32	50	KHz
上电最低工作电压		@VCC=3V		1.6(25°C) 1.3(-45°C) 1.3(85°C)		V
掉电最低工作电压		@VCC=3V		1.4(25°C) 1(-45°C) 1.1(85°C)		V

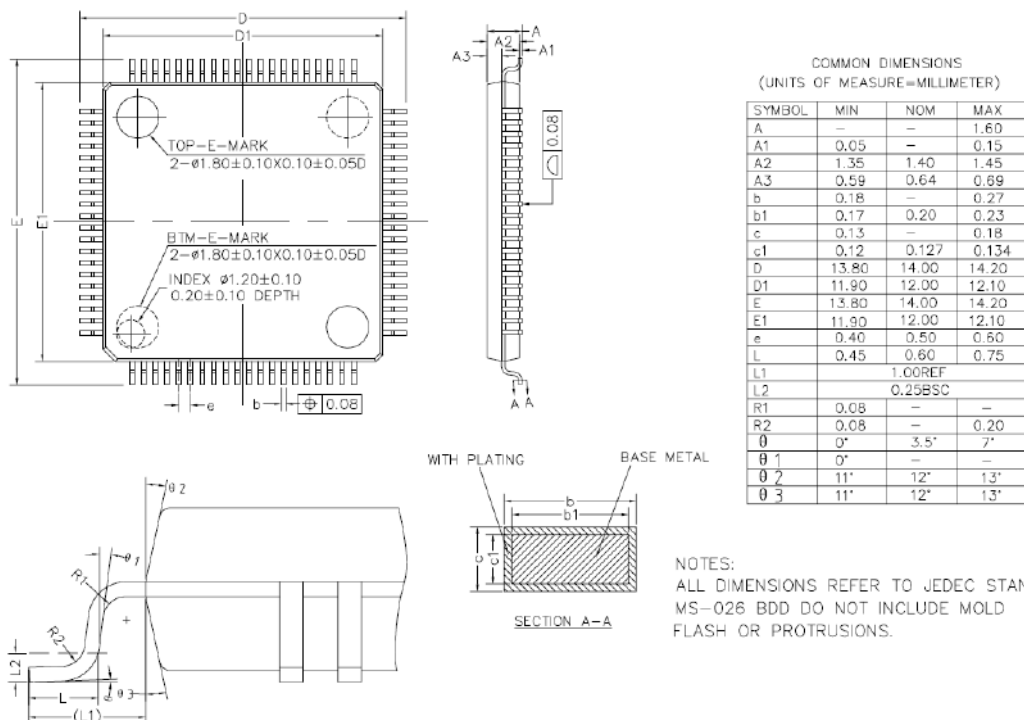
## 封装

### 芯片封装

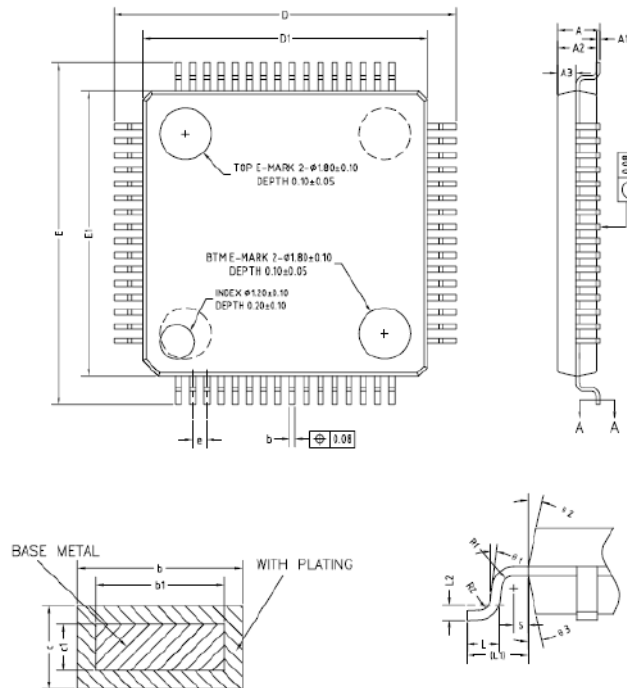
HT5023: LQFP100



HT5025: LQFP80



## HT5027: LQFP64

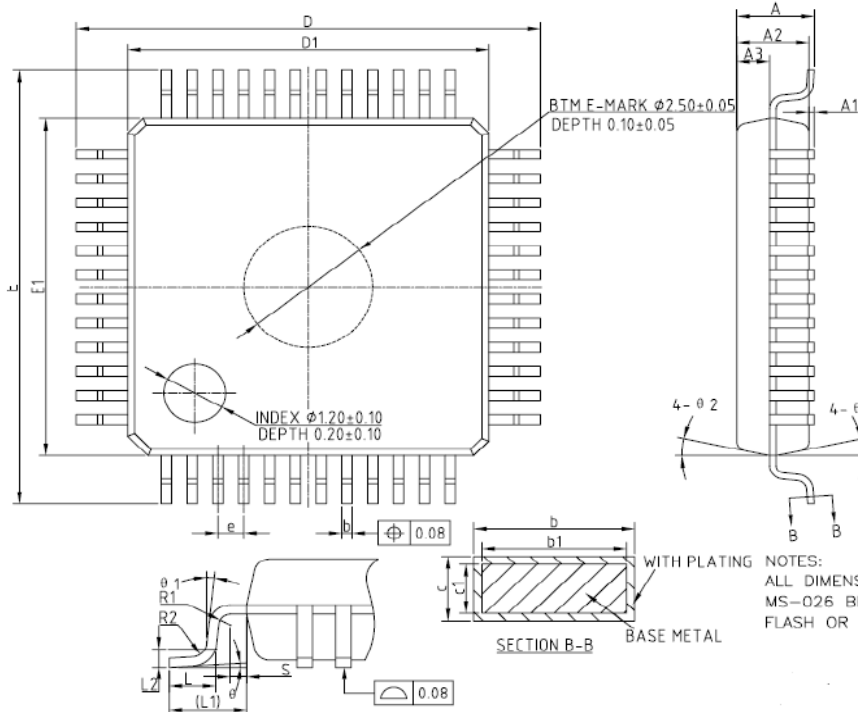


COMMON DIMENSIONS  
(UNITS OF MEASURE=MILLIMETER)

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.

## HT5029: LQFP48

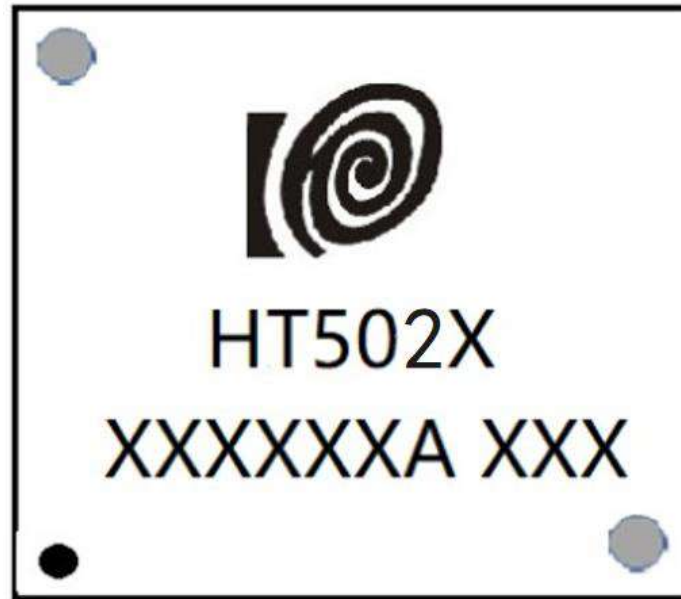


COMMON DIMENSIONS  
(UNITS OF MEASURE=MILLIMETER)

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	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
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 REFER TO JEDEC STANDARD  
MS-026 BSC DO NOT INCLUDE MOLD  
FLASH OR PROTRUSIONS.

## 丝印说明

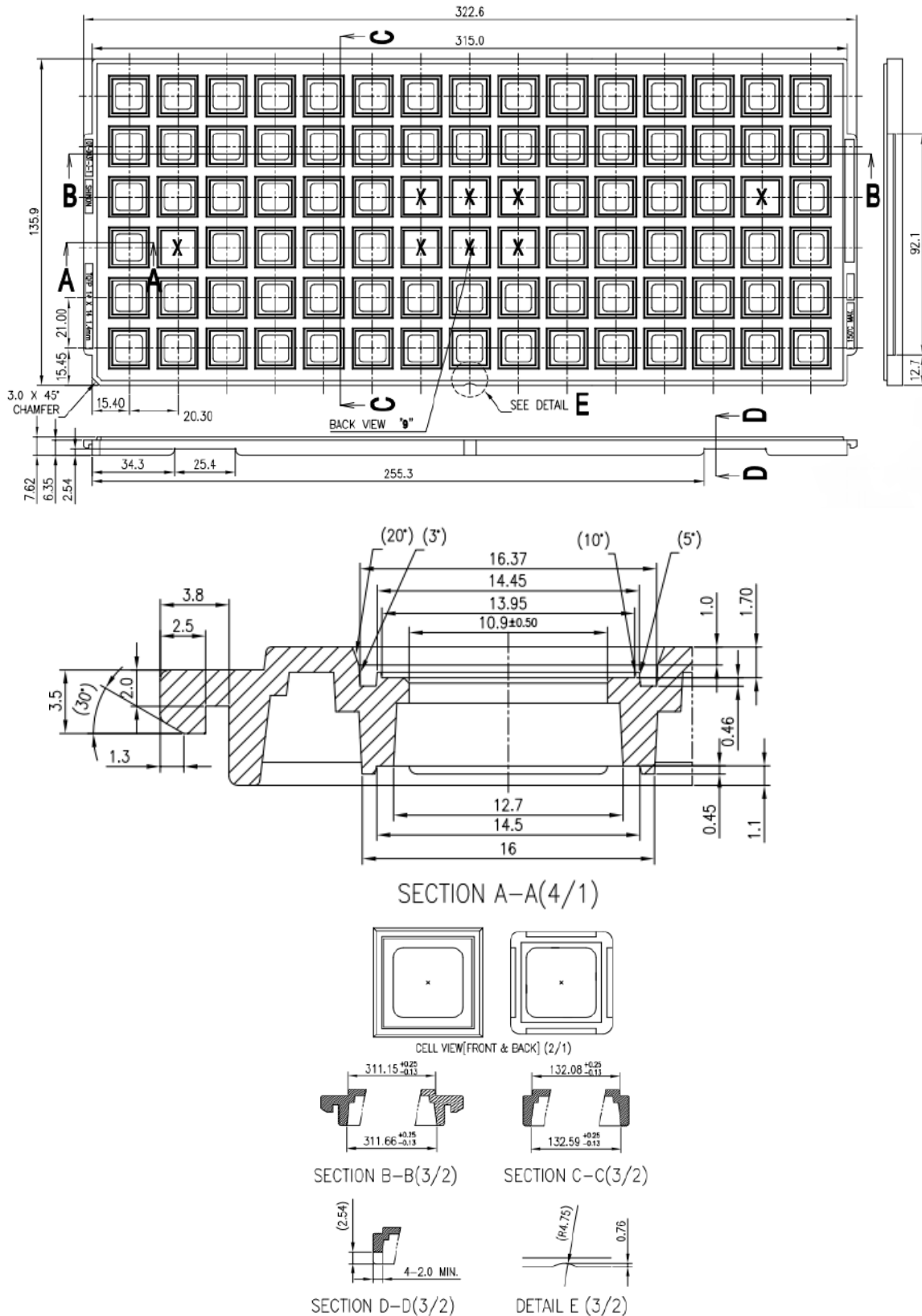


### 图样说明:

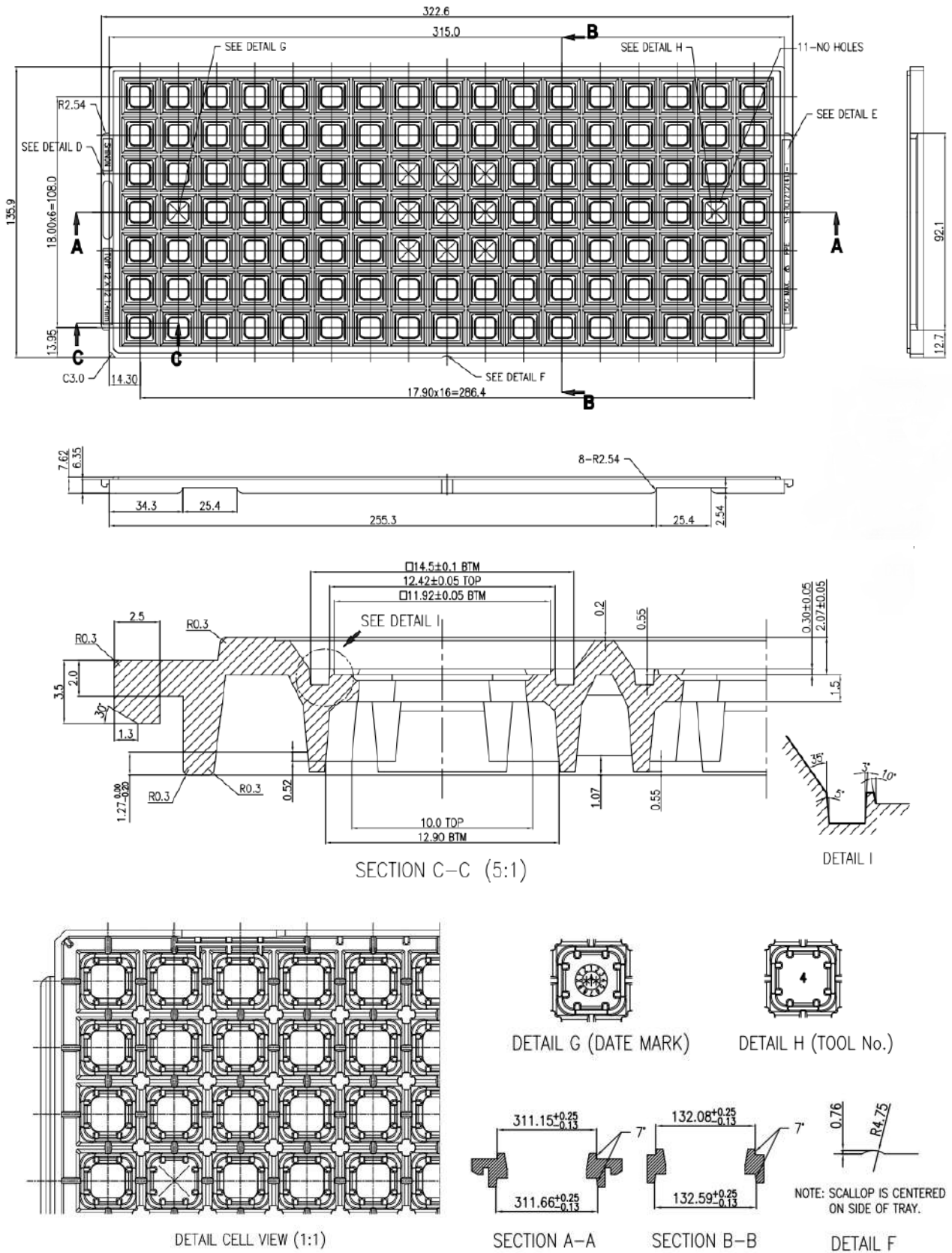
- 1) 第一行 logo 为钜泉商标。
- 2) 第二行字符为“产品名称”，如图“HT502X”为产品名称，本说明包含 HT5023/HT5025/HT5027 等产品。
- 3) 第三行左侧：“XXXXXX”前五位字符为产品批号，第六位“X”为版本号，“A”为供应商固定代码（图样此位为 A）。
- 4) 第三行右侧：“XXX”三个字符为 DATE CODE。

## 包装尺寸

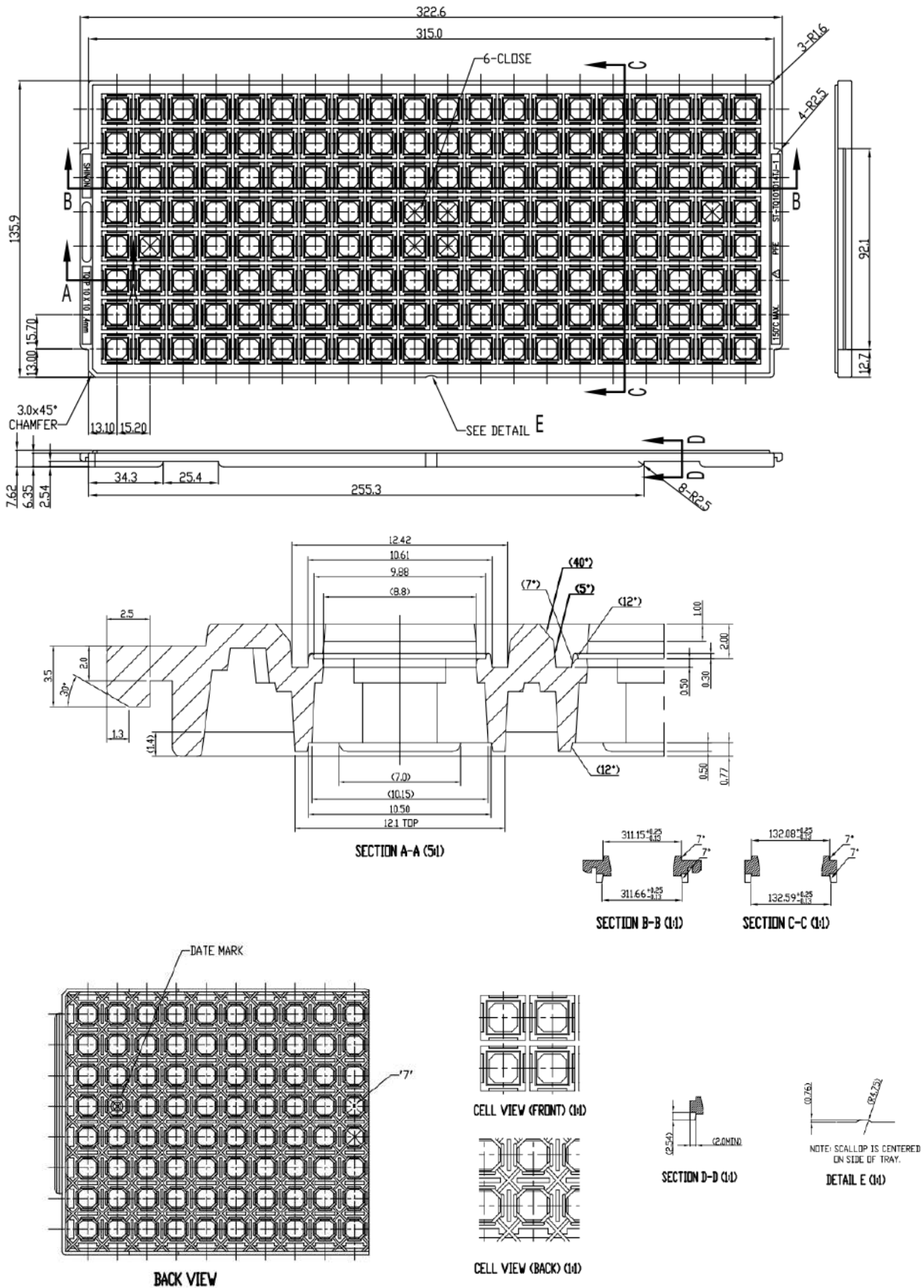
HT5023 tray 盘规格如下图:



HT5025 tray 盘规格如下图:



HT5027 tray 盘规格如下图:



HT5029 tray 盘规格如下图:

