

# HT762X

# 用户手册

钜泉光电科技（上海）股份有限公司

Tel: 021-51035886

Fax: 021-50277833

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

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

多功能、高性能、低功耗电能计量 SOC 芯片，内部集成了 Cortex-M0 处理器、时钟管理、电源管理、硬件自动温度补偿 RTC、误差自监测单元、电能计量单元

## MCU 功能

- **内核：ARM 32 位的 Cortex-M0 CPU**
  - 最高 39.3MHz 工作频率
- **存储器**
  - 256K Flash
  - 80K SRAM
- **时钟、复位和电源管理**
  - 1.8~5.5V 供电和 I/O 引脚
  - 上电/掉电复位 (POR/LBOR)
  - 内建出厂校准的 9.8MHz 的 RC 振荡器
  - 内建 32KHz 的 RC 振荡器
  - 产生 CPU 和系统时钟的 PLL
  - 带低功耗高精度的 32768Hz OSC 振荡器
- **RTC (独立 VRTC 供电)**
  - -40°C~85°C 范围内，时钟精度在 ±5ppm 内
  - VRTC: 2~5.5V RTC 具有自动温度补偿 1~2V 保持温度补偿数据
- **低功耗 (睡眠、待机模式)**
  - Sleep 模式: 3.7uA
  - Hold 模式: 11.1uA
- **Sigma-delta ADC**
  - 温度传感器: -40°C~85°C 范围内，误差小于 1°C
  - ADC 输入范围: 0~840mV
  - 快速触发功能和注入模式功能
  - 最高正常工作时钟可达 1Mhz
  - 可支持外部 13 路 ADC 输入支持 8 路测温端子
- **DMA**
  - 13 个独立可配置通道
  - 支持的外设: UART、ISO7816、SPI、I2C、Timer、SRAM、Soft request、ADC、EMU、EMU interface
- **FFT 计算单元 (22 位)**
  - 支持 64、128、256、512、1024 点计算
- **数据帧打包模块 FRAMPACK**
  - 支持 EMU 数据自动组包功能
  - 与 SPI0\2 配合自动连续发送功能
  - 支持 CRC、累加和校验功能
- **GPIO**
  - 默认高阻态
  - 部分 IO 支持 5v tolerance
  - 支持 TTL 电平
  - 具有输入滤波功能
- **调试模式**
  - 串行单线调试 (SWD)
- **多达 6 个 16 位定时器**
  - 定时功能、PWM 功能、捕获功能、事件计数功能
  - Timer4/5 支持低功耗下工作支持无内部时钟事件计数功能
- **多达 11 个通信接口**
  - I2C 接口数: 1
  - UART 接口数: 7 (两路支持 7816 功能)
  - SPI 接口数: 4 (SPI0\2 具有 32\*8bit FIFO)
- **WDT**
  - 正常模式和低功耗运行模式下，WDT 不可关闭。Sleep/Hold 模式下可软件关闭 WDT。
- **多达 10 路外部中断**
- **内置 AES/GHASH/RAND/CRC 模块**

## ■ ADC 和计量参数

- 7 路  $\Sigma$ - $\delta$ ADC，内置 PGA 可选。其中第七路 ADC 最大可到 **32** 倍。
- 8000:1 动态范围，12.8k/25.6k 采样率可选
- 内置 Vref 温度系数在 -40°C ~ 80°C 温度范围，典型值 5ppm。
- 提供 A/B/C/合 相的全波/基波/谐波有功、无功、视在功率寄存器
- 提供 7 路通道的全波电压电流、基波电压电流、谐波电压电流有效值
- 提供电压、电流矢量和有效值，**真线电压**有效值寄存器
- 提供 A/B/C 分相和合相 电压频率，全波、基波功率因数
- 提供 7 路 ADC 的电压/电流角度寄存器，全波/基波可选
- 提供 7 路 ADC 通道采样的瞬时采样数据、直流数据
- 支持各种状态指示和相应中断
- 支持三相全失压检测

## ■ 误差自监测

- 提供电压、电流、相位和 Vref 监测
- 多频点支持

## ■ 电能部分

- 芯片共 24 路电能计量模块。每路均提供正、反向电能计量通道。即共 48 个电能寄存器
- 24 路电能均支持自定义常数功率输入
- 芯片提供 5 个脉冲输出口，可任选输出 6 路合相电能的**只正/只反/正反向**的电能脉冲输出。
- 提供二级电能脉冲机制，可使电能寄存器

精确到 1/N CF 脉冲

- 谐波电能具备独立的 EC 常数
- 支持**正反电能频繁切换**

## ■ 校表功能

- 提供 7 路 ADC 通道的移采样点相位校正 (**可达 10 度**)，相位分段校正
- 提供各通道 Gain 与通道 TGain 增益
- 全波基波谐波各自独立的 GP/Gphs 与 PowerOffset 校表参数
- 提供自动温度补偿功能

## ■ 电能质量 (纯硬件支持)

- 基于**锁相环的工频同步**系统
- 提供电压暂升、电压暂降、电压 Interrupt 事件的相关功能配置
- 提供电流过流事件检测的相关功能配置
- IEC 定义的电压**真闪变算法**计算模块。
- IEC 定义的**真谐波/间谐波算法**支持
- 提供灵活可配的 ADC 数据 Channel Remap，专用 DMA 与硬件 FFT、PACK 功能
- 提供连续 **Goertzel 基波复数**输出，方便电压/电流不平衡计算
- 提供 7 路 ADC 通道的半波峰值、半波有效值等参数
- IEEE1459 视在功率支持

## ■ 其他

- 支持电压之间、电流之间接错线的**采样通道任意互换、反向配置**
- 内置 CT 二次侧检测模块
- Vref 可外接
- 罗氏线圈

## 版本更新说明

版本号	修改时间	修改人	修改内容
V0.0	2020-09-24	smduan	初版 更新部分管脚命名
V0.1	2020-11-15	Chyang	1. 增加 EMU 计量相关等说明； 2. 增加不具备误差自监测功能的型号 HT7625S；
V0.2	2021-01-08	xyyuan	1. 修改 EMU 计量单元的部分描述 2. 删除 64Pin 封装的相关描述 3. 规范寄存器列表 4. 添加电气规格
V0.3	2021-04-15	Chyang	1. 增加型号 HT7627S 的管脚定义和封装图； 2. 修改计量 ADC 管脚的输入阻抗
V0.31	2021-07-08	Hshan	1. 统一页边距 2. 半波峰值和过零章节描述修改 3. 勘误
V0.32	2021-07-12	Hshan	1. 勘误 PJ.0,PJ.1 PTOD 说明 2. 修订目录编辑错误
V0.33	2021-07-14	Hshan	1. 部分编辑问题 2. 勘误 VTP Buffer 输入电压范围 0.3V-2.6V 3. 修订页脚不统一问题
V0.34	2021-08-05	Hshan	1. 修订半波周期峰值寄存器描述 2. 补充合相功率寄存器为分相功率寄存器之和的一半的说明 3. 修订 25.1 章节 Vi 的输入范围为：0~VCC 4. 部分编辑错误 5. 修订计量校表过程
V0.35	2021-10-19	Hshan	1. 修订部分编辑错误
V0.36	2021-10-26	Hshan	1.修订 I2C 主机接收模式状态 40H 描述错误
V0.4	2021-12-14	Hshan	1. 部分编辑错误修订 2. 线电压计算公式修订 3. 有效值偏置寄存器校正计算公式修订 4. 补充 VTP 计算公式
V0.5	2022-05-16	Hshan	1. FRAMPACK 模块框图修订 2. 增加 EMU TDVREF 计算公式 3. 修订 FIFOIF 寄存器描述错误 4. 修订 FilterCon 寄存器位描述 5. 修订 FFTCFG.FFMODE 位描述 6. 修订 CFCFG1,CFCFG2 寄存器位命名序号 7. 修订部分编辑错误
V0.6	2022-06-06	Hshan	1. 新增 RTCTESTCTRL 寄存器说明
V0.7	2022-06-20	Hshan	1. 修订 EMU 寄存器基地址和 Offset

V0.8	2022-07-05	Hshan	<ol style="list-style-type: none"> <li>1. 修订 CFCFG_IOx 与引脚 PFOUTx 电能脉冲输出对应关系</li> <li>2. EMU 时钟框图中增加 EMUCLK 标识</li> <li>3. 增加电压频率计算公式举例</li> <li>4. 补充 60Hz 系统下基波过零 FundZCDelay 推荐值</li> <li>5. 修订 GP 计算公式编辑错误</li> <li>6. 修订 PJ 口的驱动能力为 1.5mA</li> <li>7. 修订起动方式命名与寄存器描述一致</li> <li>8. 新增 Pstart 计算公式，并举例</li> </ol>
V0.9	2022-07-18	Hshan	<ol style="list-style-type: none"> <li>1. 修订谐波正向有功电能寄存器名称</li> <li>2. 有效值起动，增加电流转换系数说明</li> <li>3. FIFOSTA 寄存器统一替换为 FIFOIF</li> <li>4. 补充寄存器 Amp_THO,Lock_THO,TED_Amnt 推荐值</li> <li>5. 删除 LBORCFG 寄存器</li> </ol>
V1.0	2023-05-06	Aqzhao	<ol style="list-style-type: none"> <li>1. 修改 ADC 测量电压计算公式</li> <li>2. 新增 TBSTEST 寄存器说明</li> </ol>
V1.1	2023-12-25	Jzwang	<ol style="list-style-type: none"> <li>1. 删除 EMU 配置寄存器中 Bit4 UacSel 位</li> <li>2. 修订 CFCFG2 寄存器中 CFCFG_IOx 的关系定义表格</li> <li>3. 修订 ADCINx 计算公式</li> </ol>

## 目 录

<b>1</b>	<b>HT762X 概述</b> .....	<b>16</b>
1.1	简介 .....	16
1.2	框图 .....	17
1.3	引脚排列.....	18
1.4	引脚定义.....	22
1.5	缩略语 .....	26
<b>2</b>	<b>存储器模块</b> .....	<b>28</b>
2.1	概述 .....	28
2.2	存储器映射图.....	29
2.3	FLASH 操作 .....	30
2.3.1	Flash 的读保护.....	30
2.3.2	Code Flash 的操作说明.....	30
2.4	INFORMATION BLOCK 的操作说明.....	31
2.5	FLASH 控制功能 .....	31
2.6	特殊功能寄存器列表.....	33
2.7	特殊功能寄存器说明.....	33
2.7.1	WPREG (写保护寄存器) .....	33
2.7.2	FLASHCON (Flash 控制寄存器, 写保护) .....	34
2.7.3	FLASHLOCK (Flash 锁定寄存器) .....	34
2.7.4	INFOLOCK (Information Block 锁定寄存器) .....	35
<b>3</b>	<b>时钟单元</b> .....	<b>36</b>
3.1	时钟分类.....	36
3.2	时钟框图.....	36
3.3	时钟选择机制说明.....	38
3.3.1	HRC 19.6M 高频方案.....	38
3.3.2	RTC 模块时钟选择机制.....	38
3.4	时钟说明.....	38
3.4.1	内部低频 RC 时钟 (Flrc) .....	38
3.4.2	内部高频 RC 时钟 (Fhrc) .....	38
3.4.3	外部低频晶振时钟 (Fosc) .....	38
3.4.4	内部 PLL 时钟 (Fpll) .....	38
3.4.5	时钟源的起振时间 .....	39
3.5	时钟停振与锁定算法说明.....	39
3.5.1	OSC 停振检测算法.....	39
3.5.2	PLL 锁定算法.....	40
3.5.3	时钟安全机制.....	41
3.5.4	时钟异常状态处理 .....	42
3.6	特殊功能寄存器列表.....	42
3.7	特殊功能寄存器说明.....	43

<b>4</b>	<b>电源单元</b>	<b>60</b>
4.1	概述	60
4.2	框图	61
4.3	电源单元详细功能说明	62
4.3.1	电源切换	62
4.3.2	电源实时监测	62
4.3.3	内建 1.5V 电源	62
4.3.4	BOR 检测功能(BOR_DET)	62
4.3.5	系统电源检测功能(VCC_DET)	65
4.3.6	低电压检测功能(LVDIN_DET)	66
4.3.7	VCC_DET, BOR_DET , LVDIN_DET 分时检测时序	66
4.4	特殊功能寄存器列表	66
4.5	特殊功能寄存器说明	67
<b>5</b>	<b>调试支持</b>	<b>73</b>
5.1	概况	73
5.2	SW 引脚分布	73
5.3	SW 口使用说明	73
<b>6</b>	<b>工作模式</b>	<b>75</b>
6.1	工作模式	75
6.2	睡眠模式 (SLEEP)	75
6.2.1	SLEEP 模式概述	75
6.2.2	SLEEP 模式下的唤醒源	76
6.2.3	查询 SLEEP 模式唤醒源	76
6.2.4	进入 SLEEP 模式的方式	77
6.3	待机模式 (HOLD)	77
6.3.1	HOLD 模式概述	77
6.3.2	HOLD 模式下的唤醒源	77
6.3.3	进入 HOLD 模式的方式	77
6.4	模式转换图	78
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
7.2.1	芯片引脚基本结构图	83
7.2.2	5V Tolerance IO 引脚结构图	84
7.3	关于高阻状态的说明	84
7.4	IO 的功能配置说明	85
7.5	复用功能说明	86
7.6	I/O 端口基地址列表	87
7.7	特殊功能寄存器说明	87

<b>8</b>	<b>中断模块</b>	<b>95</b>
8.1	中断向量说明	95
8.2	EXTI 中断说明	100
8.3	特殊功能寄存器列表	100
8.4	特殊功能寄存器说明	101
<b>9</b>	<b>复位模块</b>	<b>105</b>
9.1	复位优先级	105
9.2	复位说明	106
9.2.1	上电复位	106
9.2.2	低电压检测复位	106
9.2.3	外部引脚复位	107
9.2.4	掉电复位	107
9.2.5	看门狗复位	108
9.2.6	软复位	108
9.2.7	调试复位	108
9.2.8	唤醒复位	109
9.3	特殊功能寄存器列表	109
9.4	特殊功能寄存器说明	109
<b>10</b>	<b>UART/7816 通讯模块</b>	<b>112</b>
10.1	功能说明	112
10.2	波特率计算	112
10.3	串口通讯模式说明	112
10.3.1	方式 1	113
10.3.2	方式 2	113
10.3.3	方式 3	114
10.3.4	方式 4	114
10.4	红外调制	114
10.5	7816 接收和发送	115
10.5.1	7816 数据发送	115
10.5.2	7816 数据接收	115
10.5.3	7816 通讯示意图	116
10.6	特殊功能寄存器列表	119
10.7	特殊功能寄存器说明	119
<b>11</b>	<b>WDT 模块</b>	<b>127</b>
11.1	概述	127
11.2	工作模式	127
11.3	特殊功能寄存器列表	128
11.4	特殊功能寄存器说明	128
<b>12</b>	<b>定时器模块</b>	<b>130</b>
12.1	定时器(TMR)单元概述	130

12.2	定时器(TMR)时钟选择 .....	130
12.3	周期定时功能.....	130
12.4	PWM 功能.....	131
12.5	捕获功能.....	134
12.5.1	双沿捕获 (仅 TMR4/5 支持) .....	134
12.6	单次计数功能 (仅 TMR4/5 支持) .....	135
12.6.1	单次定时 .....	135
12.6.2	单脉冲输出 .....	135
12.6.3	单次捕获 .....	136
12.7	事件计数功能.....	137
12.8	扩展触发功能 (仅 TMR4/5 支持) .....	137
12.8.1	TMR4/5 事件计数或捕获输入扩展及信号数字滤波.....	137
12.8.2	TMR4/5 外部异步脉冲计数功能.....	138
12.8.3	TMR4/5 外部触发使能功能 .....	138
12.9	中断功能.....	139
12.9.1	周期定时中断 .....	139
12.9.2	捕获中断 .....	139
12.9.3	比较中断 .....	139
12.9.4	事件计数中断 .....	139
12.10	特殊功能寄存器列表.....	140
12.11	特殊功能寄存器说明.....	140
<b>13</b>	<b>SPI 模块.....</b>	<b>149</b>
13.1	概述 .....	149
13.2	详细功能说明.....	149
13.2.1	SPI 模块框图.....	150
13.2.2	SPI 接口传输格式.....	151
13.2.3	主机模式传输格式 .....	152
13.2.4	从机模式传输格式 .....	153
13.2.5	快速从机模式 (仅 SPI0/SPI2 支持) .....	156
13.2.6	FIFO 模式 (仅 SPI0/SPI2 支持) .....	156
13.2.7	中断功能.....	156
13.3	特殊功能寄存器列表.....	157
13.4	特殊功能寄存器说明.....	157
<b>14</b>	<b>I2C 模块 .....</b>	<b>167</b>
14.1	概述 .....	167
14.2	框图 .....	167
14.3	功能描述.....	168
14.3.1	串行时钟生成.....	168
14.3.2	中断生成.....	168
14.3.3	传输模式.....	168
14.4	特殊功能寄存器列表.....	176
14.5	特殊功能寄存器说明.....	177

<b>15</b>	<b>RTC 模块</b>	<b>180</b>
15.1	概述	180
15.2	修改寄存器 RTCCHECKSUM 说明	180
15.3	RTC 校时方式	180
15.4	功能描述	180
15.5	时钟校正	181
15.6	RTC 补偿系数寄存器说明	181
15.7	时间和万年历	181
15.8	RTC 时间戳功能	181
15.9	中断功能	182
15.10	RTC 指示寄存器读写流程	183
15.10.1	读取 RTC 指示寄存器流程	183
15.10.2	写入 RTC 指示寄存器流程	183
15.11	校时记录	184
15.12	RTC 备份寄存器	184
15.13	寄存器复位的相关说明	184
15.14	特殊功能寄存器列表	185
15.15	特殊功能寄存器说明	188
<b>16</b>	<b>TBS 模块</b>	<b>239</b>
16.1	概述	239
16.2	功能描述	239
16.2.1	基本功能	239
16.2.2	模块结构框图	240
16.2.3	模块时钟	240
16.3	工作模式	241
16.3.1	固定分时开启模式	241
16.3.2	实时触发模式	241
16.3.3	注入工作模式	241
16.4	测温端子功能说明	242
16.4.1	功能说明	242
16.4.2	功能指标	243
16.5	特殊功能寄存器列表	243
16.6	特殊功能寄存器说明	244
<b>17</b>	<b>AES/GHASH/RAND/CRC 模块</b>	<b>266</b>
17.1	AES 概述	266
17.2	GHASH 概述	266
17.3	RAND 随机数概述	266
17.4	CRC 概述	266
17.5	特殊功能寄存器列表	268
17.6	特殊功能寄存器说明	270
<b>18</b>	<b>DMA 功能</b>	<b>295</b>

18.1	概述 .....	295
18.2	功能描述.....	295
18.3	DMA 通道请求列表 .....	296
18.4	DMA 通用通道数据传输说明 .....	298
18.5	EMU-FFT 专用通道数据传输说明.....	300
18.6	特殊功能寄存器列表.....	302
18.7	特殊功能寄存器说明.....	305
<b>19</b>	<b>EMU INTERFACE 模块 .....</b>	<b>369</b>
19.1	概述 .....	369
19.2	使用流程.....	371
19.2.1	FFT 数据接口配置使用 .....	371
19.2.2	非侵入数据接口配置使用.....	371
19.3	FFT 数据接口.....	372
19.3.1	16bits mode 时的 FFT 数据接口地址分配.....	372
19.3.2	24bits mode 时的 FFT 数据接口地址分配.....	372
19.3.3	32bits mode 时的 FFT 数据接口地址分配 MSB.....	373
19.3.4	32bits mode 时的 FFT 数据接口地址分配 LSB.....	373
19.4	非侵入数据接口.....	374
19.4.1	16bits mode 时的地址分配.....	374
19.4.2	24bits mode 时的地址分配.....	374
19.4.3	32bits mode 时的地址分配 MSB .....	374
19.4.4	32bits mode 时的地址分配 LSB.....	375
19.5	特殊功能寄存器列表.....	375
19.6	特殊功能寄存器说明.....	375
<b>20</b>	<b>FFT 模块 .....</b>	<b>382</b>
20.1	概述 .....	382
20.2	功能描述.....	382
20.3	FFT 使用说明.....	383
20.3.1	FFT 数据配置 .....	383
20.3.2	FFT 输入输出 .....	384
20.3.3	FFT 内存访问.....	385
20.3.4	FFT 停止和复位.....	385
20.4	特殊功能寄存器列表.....	386
20.5	特殊功能寄存器说明.....	387
<b>21</b>	<b>非侵入数据帧打包模块 FRAMPACK.....</b>	<b>399</b>
21.1	概述 .....	399
21.1.1	非侵入数据帧打包模块.....	399
21.1.2	帧打包与 SPI 发送.....	401
21.1.3	发送间隔.....	402
21.2	特殊功能寄存器列表.....	402
21.3	特殊功能寄存器说明.....	403

22	EMU 三相计量单元.....	419
22.1	概述 .....	419
22.2	系统结构.....	420
22.3	计量单元功能.....	421
22.3.1	工作模式 .....	421
22.3.2	模数转换器 .....	421
22.3.3	采样波形输出 .....	421
22.3.4	功率 .....	422
22.3.5	有效值测量 .....	425
22.3.6	频率 .....	426
22.3.7	角度 .....	426
22.3.8	功率因数 .....	427
22.3.9	多功能滤波器设计 .....	427
22.3.10	半波峰值 .....	428
22.3.11	过零 .....	428
22.3.12	电能模块 .....	429
22.3.13	自动温度补偿 .....	432
22.3.14	相位分段 .....	433
22.3.15	移采样点相位校正 .....	434
22.3.16	低频计量模式 .....	434
22.4	检测与电能质量 .....	435
22.4.1	起动的潜动 .....	435
22.4.2	电压暂升 PEAK/暂降 SAG .....	436
22.4.3	电压短时中断 INT .....	438
22.4.4	电流过流事件检测 .....	439
22.4.5	反向指示 .....	440
22.4.6	失压指示 .....	441
22.4.7	电压电流错序指示 .....	441
22.4.8	采样通道互换 .....	441
22.4.9	全失压比较器设计 .....	441
22.4.10	互感器二次侧检测 .....	442
22.4.11	负载电流快速变化 .....	442
22.4.12	闪变 .....	443
22.4.13	三相不平衡度 .....	443
22.4.14	谐波/间谐波分析 .....	443
22.5	EMU 数据接口 .....	444
22.5.1	工频同步系统简介 .....	444
22.5.2	EMU-DMA 设计 .....	444
22.6	计量参数寄存器列表 .....	446
22.7	计量参数寄存器说明 .....	453
22.7.1	波形寄存器 .....	453
22.7.2	功率寄存器 .....	454
22.7.3	有效值寄存器 .....	454

22.7.4	矢量和有效值寄存器.....	455
22.7.5	线电压有效值寄存器.....	456
22.7.6	电压频率寄存器.....	456
22.7.7	功率因数寄存器.....	456
22.7.8	角度寄存器.....	457
22.7.9	电能寄存器.....	457
22.7.10	快速脉冲计数寄存器.....	458
22.7.11	ADC 采样直流数据寄存器.....	460
22.7.12	半波周期峰值寄存器.....	460
22.7.13	瞬时有效值平方寄存器.....	461
22.7.14	Peak 相关事件寄存器.....	461
22.7.15	SAG 相关事件寄存器.....	461
22.7.16	UINT 事件相关寄存器.....	462
22.7.17	电流过流事件相关寄存器.....	462
22.7.18	自动温补增益寄存器.....	463
22.7.19	半波有效值数据.....	463
22.7.20	DFT 基波运算结果寄存器.....	463
22.7.21	电压闪变 Flicker 寄存器.....	463
22.7.22	校验和寄存器.....	464
22.8	校表寄存器列表.....	465
22.9	校表寄存器说明.....	474
22.9.1	移采样点相位校正.....	474
22.9.2	ADC 直流偏置校正.....	475
22.9.3	通道温度补偿增益.....	475
22.9.4	通道增益.....	476
22.9.5	数字移位放大.....	476
22.9.6	功率增益校正.....	477
22.9.7	PQ 方式相位校正.....	477
22.9.8	功率偏置校正.....	478
22.9.9	有效值偏置校正.....	478
22.9.10	常数自定义功率寄存器 1~24.....	479
22.9.11	电能偏置功率寄存器.....	479
22.9.12	常数自定义电压.....	480
22.9.13	过零点数配置寄存器.....	480
22.9.14	起动潜动阈值.....	481
22.9.15	AVG 平均模块.....	482
22.9.16	高频脉冲 HFConst.....	482
22.9.17	电能脉冲分辨率 DIV.....	483
22.9.18	采样通道互换与反向.....	484
22.9.19	模拟 PGA 放大配置.....	485
22.9.20	ADC 配置寄存器.....	486
22.9.21	ADC 相关功能开关寄存器.....	487
22.9.22	EMU 配置寄存器.....	489
22.9.23	Rosi 与常数电压寄存器.....	490

22.9.24	常规滤波器系数配置.....	491
22.9.25	起动机配置.....	492
22.9.26	模拟参数配置寄存器.....	493
22.9.27	全失压比较器配置.....	494
22.9.28	低频计量模式配置.....	496
22.9.29	电能相关配置寄存器.....	497
22.9.30	闪变功能配置.....	503
22.9.31	基波谐波使能配置.....	504
22.10	状态标注与中断说明.....	505
22.10.1	潜动状态标志.....	505
22.10.2	功率/电能反向标志.....	506
22.10.3	功率因数过小标志.....	507
22.10.4	信号状态标志.....	507
22.10.5	Peak/Sag/UINT/IOV 状态标志.....	508
22.10.6	EMU 中断使能与中断标志.....	509
22.10.7	eWatching 与工频同步系统中断.....	515
22.11	工频同步系统配置.....	516
22.12	校表过程.....	523
22.12.1	功率法校表.....	523
22.12.2	脉冲法校表.....	528
<b>23</b>	<b>EWU 误差自监测.....</b>	<b>532</b>
<b>24</b>	<b>CORTEX-M0 内核简要说明.....</b>	<b>533</b>
24.1	概述.....	533
24.2	系统定时器 SysTick.....	533
24.3	中断优先级说明.....	533
24.4	CMSIS 函数说明.....	533
<b>25</b>	<b>电气规格.....</b>	<b>535</b>
25.1	极限参数.....	535
25.2	功耗参数.....	535
25.2.1	模块功耗.....	535
25.2.2	运行功耗.....	536
25.3	DC/AC 参数.....	536
25.3.1	存储器.....	536
25.3.2	GPIO.....	537
25.3.3	CMU.....	538
25.3.4	PMU.....	539
25.3.5	WDT.....	541
25.3.6	TIMER.....	541
25.3.7	AES/GHASH/RAND/CRC/ECC.....	541
25.3.8	TPS&ADC.....	542
25.3.9	RTC.....	543
25.3.10	计量 ADC 参数.....	544

---

26	封装.....	546
26.1	耐焊性 .....	546
26.2	封装图 .....	546

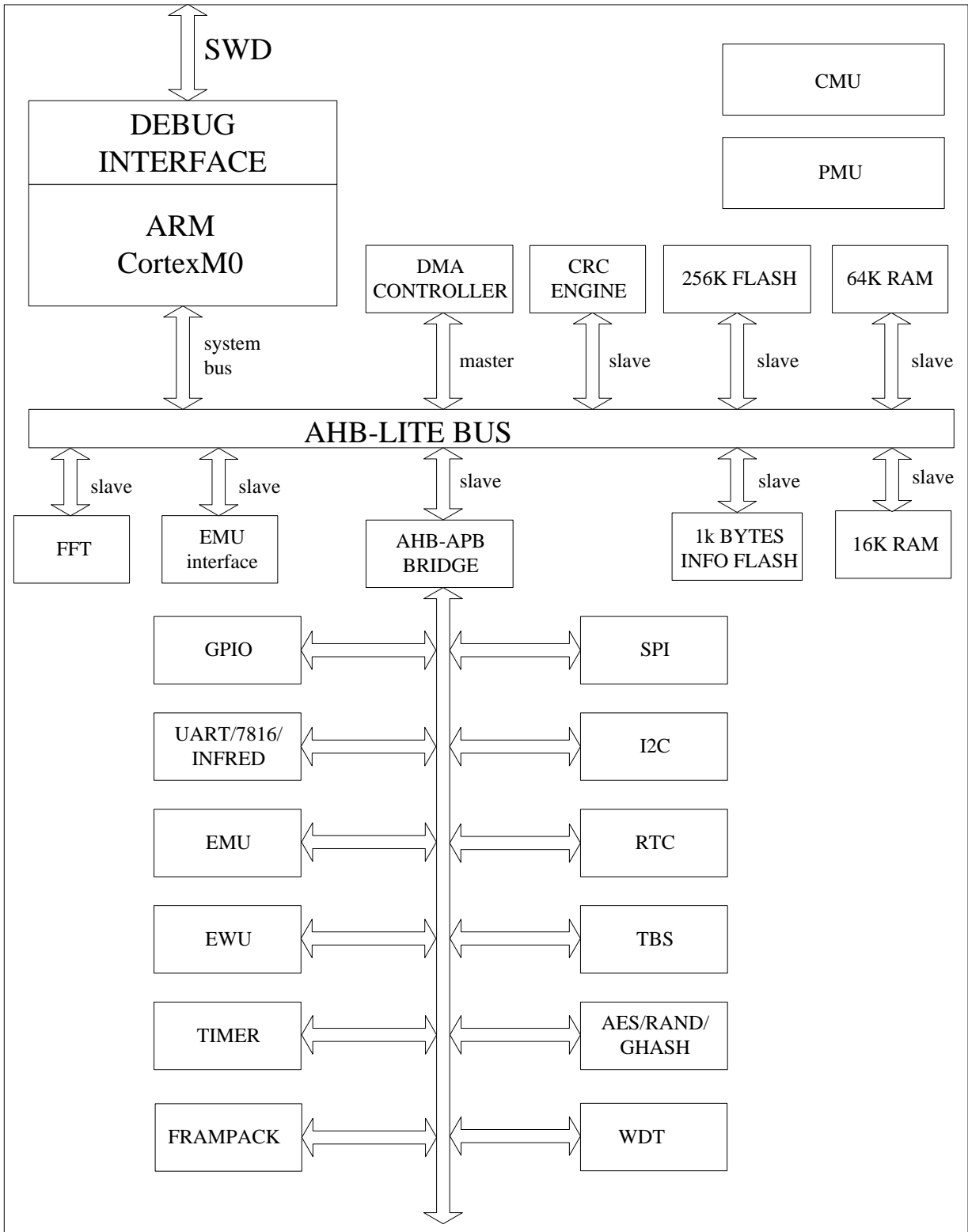
# 1 HT762X 概述

## 1.1 简介

HT762X 系列是多功能、高性能、低功耗电能计量 SoC 芯片，内部集成了 Cortex-M0 处理器、时钟管理、电源管理、硬件自动温度补偿 RTC、PLL、高频 RC、低频 RC，以及 NVIC 和 DEBUG 调试功能。其中，支持每秒补偿机制的 RTC 单元，芯片以 32.768KHz 晶振时钟源作为 RTC 时钟源，通过芯片内部集成的时钟自动数字补偿单元，协助用户在无需其软件参与的情况下，实现 RTC 的自动补偿。

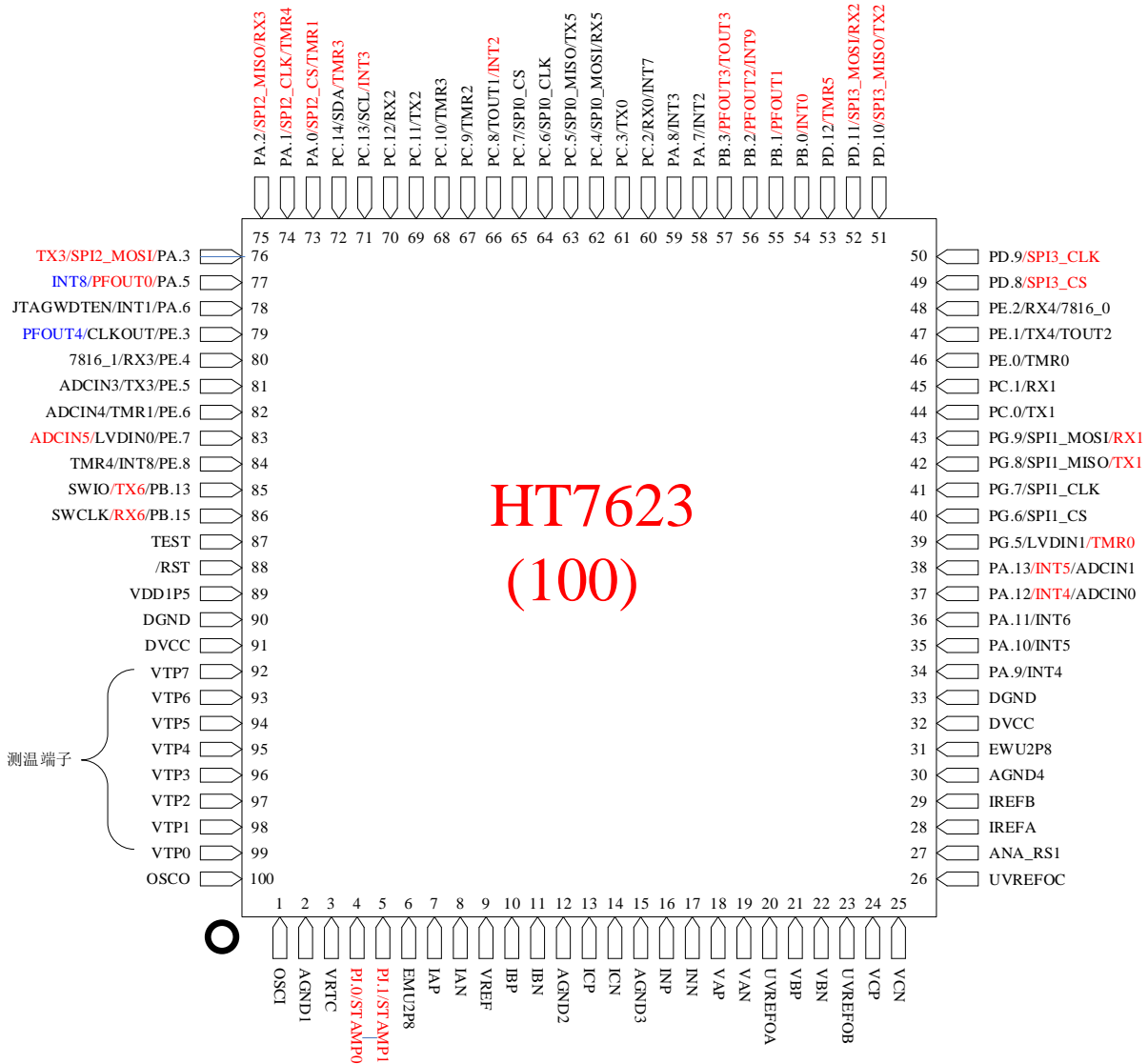
- 工作电压范围：2.0V~5.5V
- 工作温度范围：-40°C~85°C
- 采用 ARM Cortex-M0 CPU Core、256K Flash、80K SRAM
- 高速度：CPU 最高工作频率达到 39.3M
- 低功耗：Hold 模式下最低功耗 11.1uA  
Sleep 模式下最低功耗 3.7uA
- RTC：支持每秒补偿机制
- RTC 补偿：RTC 内置曲线数字补偿，全温度范围 RTC 补偿无需用户软件参与
- 供电方式：独立的 RTC 供电引脚，在芯片内除 RTC 模块之外的所有模块不供电的情况下，RTC 模块仍然可以保持独立工作
- 高精度温度传感器：-40 度 ~ +85 度 温度范围内，温度传感器一致性优于正负 1 度
- 2 路硬件 7816 功能
- 正常模式下，WDT 模块不可关闭，保证系统可靠运行。在 sleep/hold 模式下，可软件关闭 WDT 模块
- 采用绿色封装：(HT7623) LQFP100; (HT7625) LQFP80; (HT7625S) LQFP80; (HT7627S) LQFP64

1.2 框图

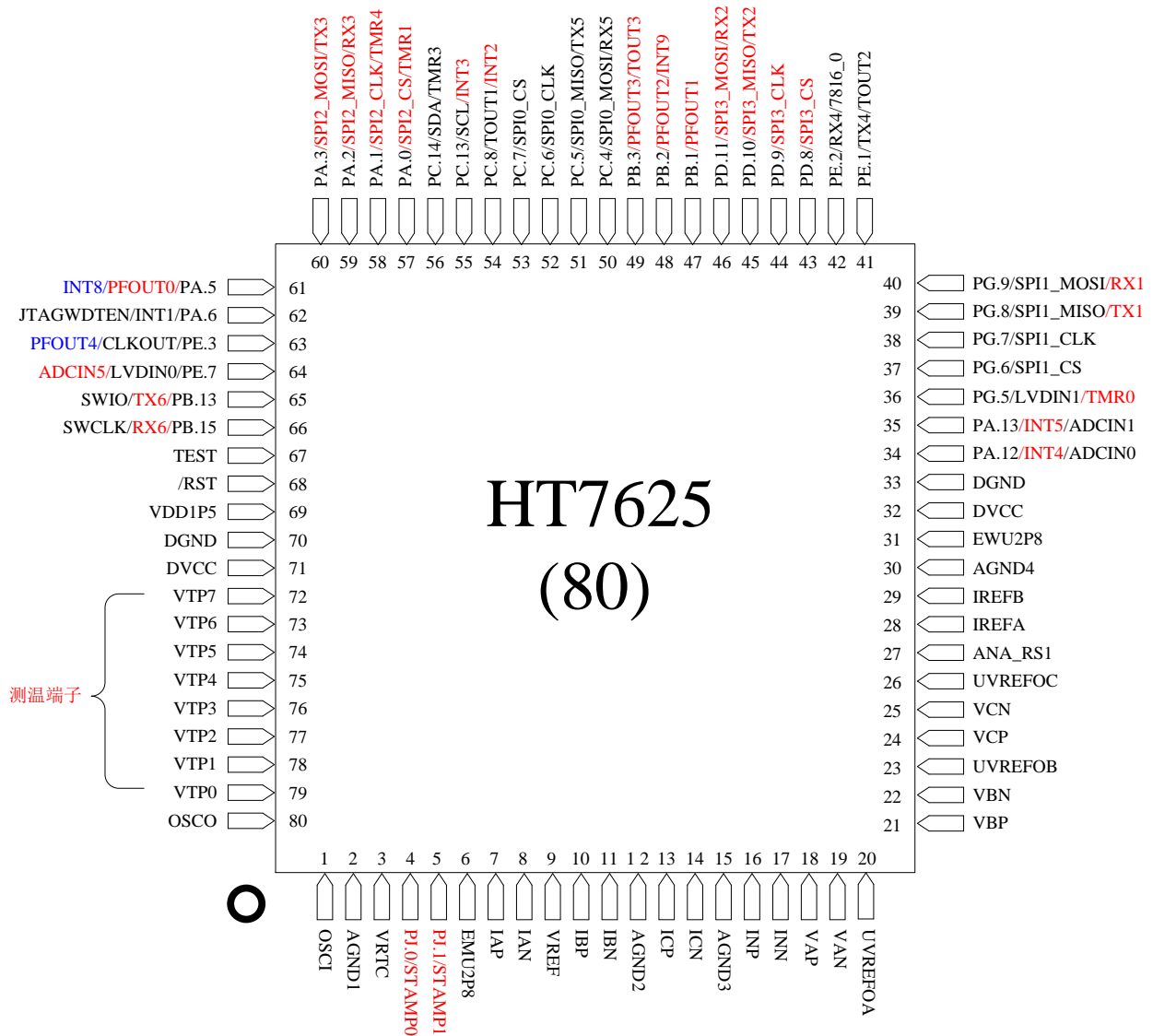


## 1.3 引脚排列

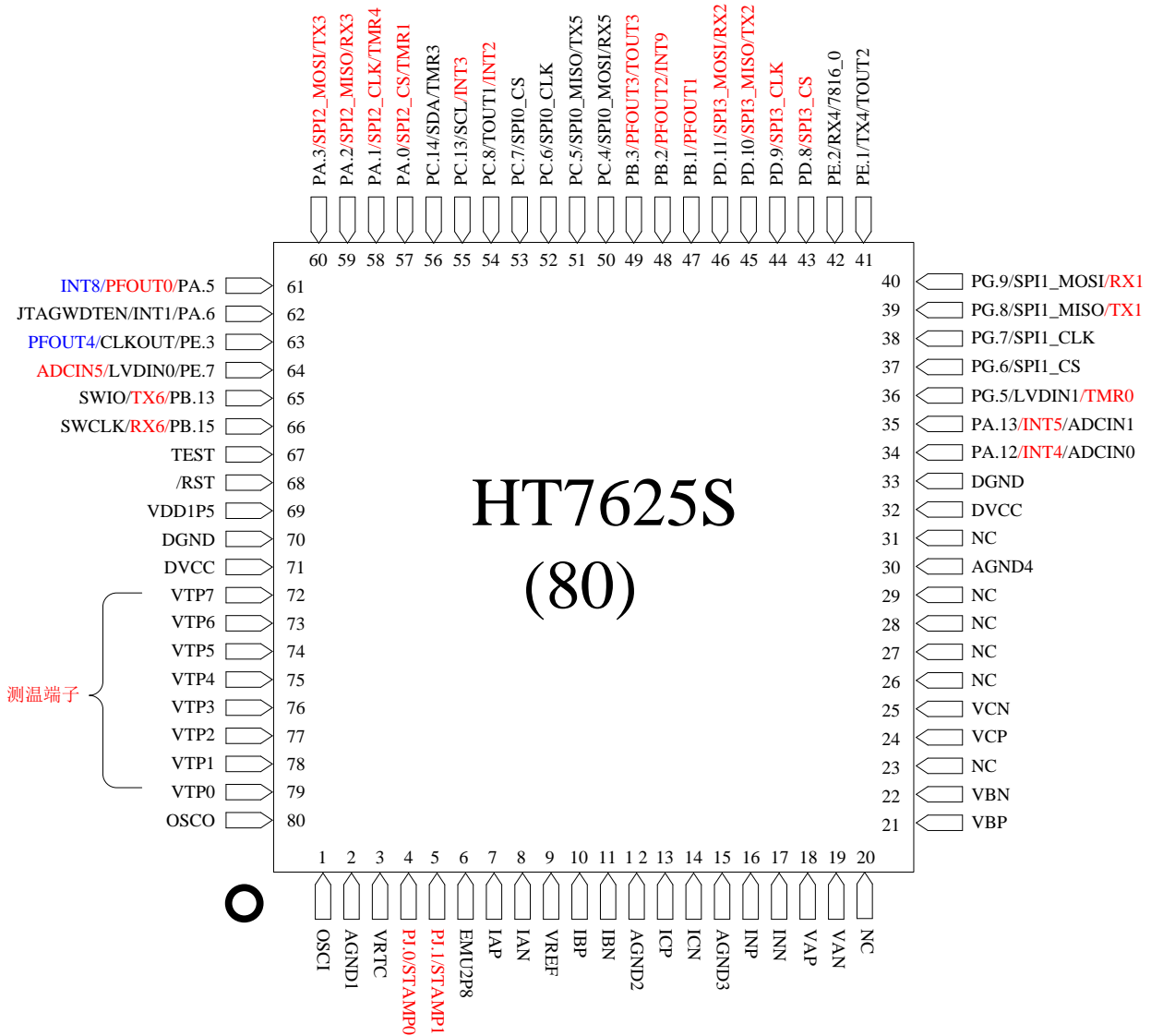
100pin:

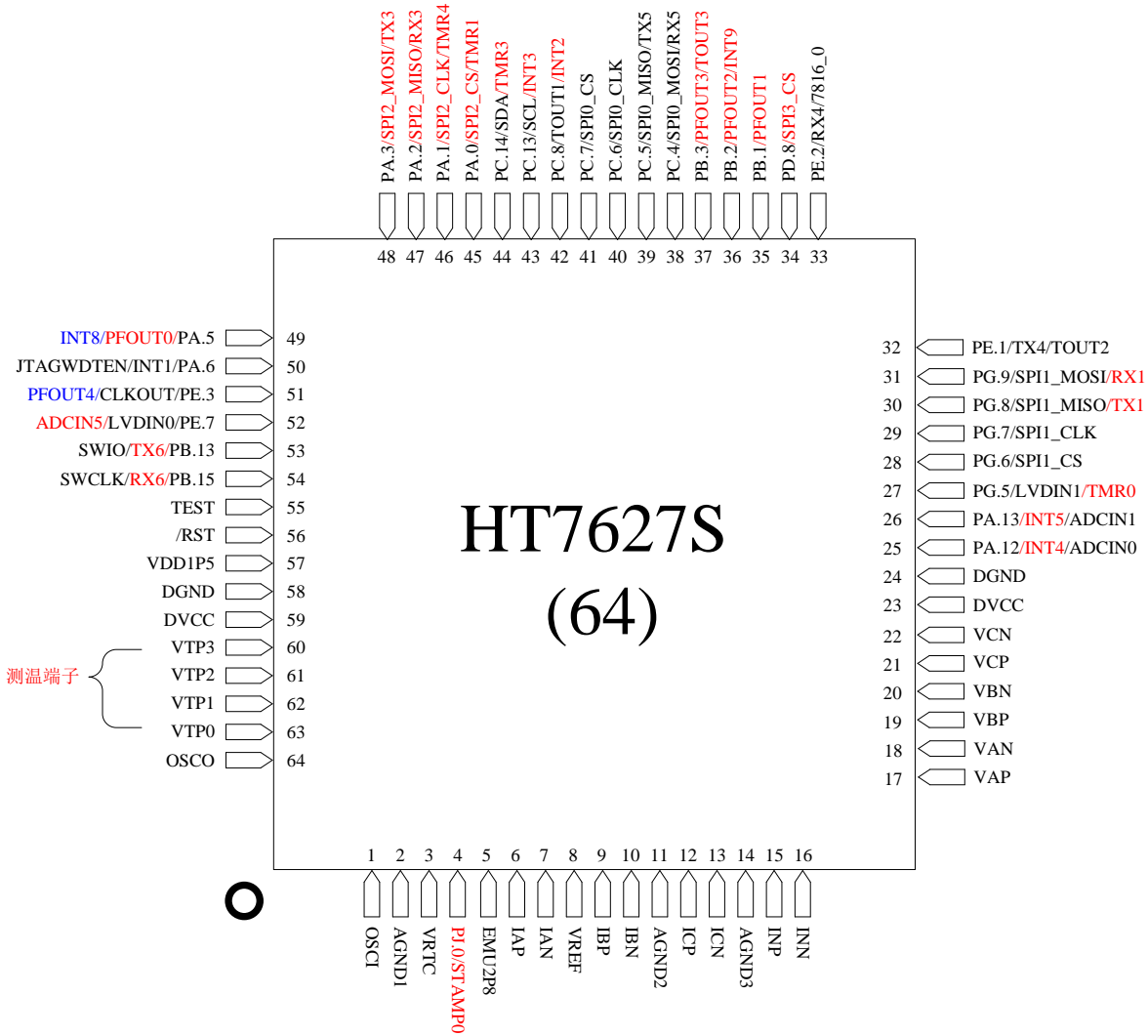


80pin: HT7625



HT7625S





注: HT7625S HT7627S 无 EWU 误差自监测相关功能。

## 1.4 引脚定义

100 PIN	80 PIN	64 PIN (S)	标识	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	OSCI	I	X	X	32K 外部时钟输入
2	2	2	AGND1	G	X	X	芯片模拟地
3	3	3	VRTC	PI	X	X	时钟电源输入，需外接滤波电容 1uF&0.1uF
4	4	4	PJ.0	I/O	STAMP0	X	时间戳信号 输入滤波 2us (GPIO 低电平和高电平输出电 流大小均为 1.5mA)
5	5	-	PJ.1	I/O	STAMP1	X	时间戳信号 输入滤波 2us (GPIO 低电平和高电平输出电 流大小均为 1.5mA)
6	6	5	EMU2P8	PO	X	X	模拟 EMU 2.8V 电源输出，需外接 10uF&0.1uF 电容滤波
7	7	6	IAP	I	X	X	电流通道 A 模拟信号输入端 (正)
8	8	7	IAN	I	X	X	电流通道 A 模拟信号输入端 (负)
9	9	8	VREF	O	X	X	计量基准电压输出，需外接滤波电容 0.1uF&1uF
10	10	9	IBP	I	X	X	电流通道 B 模拟信号输入端 (正)
11	11	10	IBN	I	X	X	电流通道 B 模拟信号输入端 (负)
12	12	11	AGND2	G	X	X	芯片模拟地
13	13	12	ICP	I	X	X	电流通道 C 模拟信号输入端 (正)
14	14	13	ICN	I	X	X	电流通道 C 模拟信号输入端 (负)
15	15	14	AGND3	G	X	X	芯片模拟地
16	16	15	INP	I	X	X	电流通道模拟信号输入端 (正)
17	17	16	INN	I	X	X	电流通道模拟信号输入端 (负)
18	18	17	VAP	I	X	X	电压通道 A 模拟信号输入端 (正)
19	19	18	VAN	I	X	X	电压通道 A 模拟信号输入端 (负)
20	20	-	UVREFOA	O	X	X	自校正电压基准输出 A
21	21	19	VBP	I	X	X	电压通道 B 模拟信号输入端 (正)
22	22	20	VBN	I	X	X	电压通道 B 模拟信号输入端 (负)
23	23	-	UVREFOB	O	X	X	自校正电压基准输出 B
24	24	21	VCP	I	X	X	电压通道 C 模拟信号输入端 (正)
25	25	22	VCN	I	X	X	电压通道 C 模拟信号输入端 (负)

26	26	-	UVREFOC	O	X	X	自校正电压基准输出 C
27	27	-	ANA_RS1	O	X	X	eWatching 基准电流外接 12.5kΩ 高精度电阻
28	28	-	IREFA	O	X	X	eWatching 自校准电流基准输出 A
29	29	-	IREFB	O	X	X	eWatching 自校准电流基准输出 B
30	30	-	AGND4	G	X	X	芯片模拟地
31	31	-	EWU2P8	PO	X	X	自校准 eWatching 电源输出, 需外接滤波电容 10uF&0.1uF
32	32	23	DVCC	PI	X	X	芯片数字电源, 需外接滤波电容 10uF&0.1uF
33	33	24	DGND	G	X	X	芯片数字地
34	-	-	PA.9	I/O	INT4	X	外部中断 4 输入滤波 2us
35	-	-	PA.10	I/O	INT5	X	外部中断 5 输入滤波 2us
36	-	-	PA.11	I/O	INT6	X	外部中断 6 输入滤波 2us
37	34	25	PA.12	I/O	INT4	ADCIN0	外部中断 4/ADC 输入通道 0
38	35	26	PA.13	I/O	INT5	ADCIN1	外部中断 5/ADC 输入通道 1
39	36	27	PG.5	I/O	LVDIN1	TMR0	外部低电压检测 1/PWM 主输出_捕获输入_事件输入
40	37	28	PG.6	I/O	SPI1_CS	X	SPI1 片选
41	38	29	PG.7	I/O	SPI1_CLK	X	SPI1 时钟
42	39	30	PG.8	I/O	SPI1_MISO	TX1	SPI1 主输入从输出/UART1 发送
43	40	31	PG.9	I/O	SPI1_MOSI	RX1	SPI1 主输出从输入/UART1 接收
44	-	-	PC.0	I/O	TX1	X	UART1 发送 支持大电流驱动
45	-	-	PC.1	I/O	RX1	X	UART1 接收
46	-	-	PE.0	I/O	TMR0	X	PWM 主输出_捕获输入_事件输入 TTL 电平输入
47	41	32	PE.1	I/O	TX4	TOUT2	UART4 发送/RTC 秒时钟输出
48	42	33	PE.2	I/O	RX4	7816_0	UART4 接收/7816_IO 输入滤波 2us
49	43	34	PD.8	I/O	SPI3_CS	X	SPI3 片选
50	44	-	PD.9	I/O	SPI3_CLK	X	SPI3 时钟
51	45	-	PD.10	I/O	SPI3_MISO	TX2	SPI3 主输入从输出/UART2 发送
52	46	-	PD.11	I/O	SPI3_MOSI	RX2	SPI3 主输出从输入/UART2 接收
53	-	-	PD.12	I/O	TMR5	X	PWM 输出_捕获输入_事件输入
54	-	-	PB.0	I/O	INT0	X	外部中断 0
55	47	35	PB.1	I/O	PFOUT1	X	电能脉冲输出 1

56	48	36	PB.2	I/O	PFOUT2	INT9	电能脉冲输出 2/外部中断 9
57	49	37	PB.3	I/O	PFOUT3	TOUT3	电能脉冲输出 3/RTC 秒时钟输出 3
58	-	-	PA.7	I/O	INT2	X	外部中断 2 输入滤波 2us 支持大电流驱动
59	-	-	PA.8	I/O	INT3	X	外部中断 3 输入滤波 2us 支持大电流驱动
60	-	-	PC.2	I/O	RX0	INT7	UART0 接收/外部中断 7
61	-	-	PC.3	I/O	TX0	X	UART0 发送
62	50	38	PC.4	I/O	SPI0_MOSI	RX5	SPI0 主输出从输入/UART5 接收 输入滤波 2us TTL 电平输入 5V Tolerance IO
63	51	39	PC.5	I/O	SPI0_MISO	TX5	SPI0 主输入从输出/UART5 发送 TTL 电平输入 5V Tolerance IO
64	52	40	PC.6	I/O	SPI0_CLK	X	SPI0 时钟 TTL 电平输入 5V Tolerance IO
65	53	41	PC.7	I/O	SPI0_CS	X	SPI0 片选 TTL 电平输入 5V Tolerance IO
66	54	42	PC.8	I/O	TOUT1	INT2	RTC 秒时钟输出 1/外部中断 2
67	-	-	PC.9	I/O	TMR2	X	PWM 主输出_捕获输入_事件输入 TTL 电平输入 5V Tolerance IO
68	-	-	PC.10	I/O	TMR3	X	PWM 主输出_捕获输入_事件输入 TTL 电平输入 5V Tolerance IO
69	-	-	PC.11	I/O	TX2	X	UART2 发送
70	-	-	PC.12	I/O	RX2	X	UART2 接收 输入滤波 2us
71	55	43	PC.13	I/O	SCL	INT3	I2C 时钟/外部中断 3 TTL 电平输入 5V Tolerance IO
72	56	44	PC.14	I/O	SDA	TMR3	I2C 数据/PWM 主输出_捕获输入_事件输入 TTL 电平输入 5V Tolerance IO
73	57	45	PA.0	I/O	SPI2_CS	TMR1	SPI2 片选/PWM 主输出_捕获输入_事件输入
74	58	46	PA.1	I/O	SPI2_CLK	TMR4	SPI2 时钟/PWM 输出_捕获输入_事件输入

75	59	47	PA.2	I/O	SPI2_MISO	RX3	SPI2 主输入从输出/UART3 接收
76	60	48	PA.3	I/O	SPI2_MOSI	TX3	SPI2 主输出从输入/UART3 发送
77	61	49	PA.5	I/O	PFOUT0	INT8	电能脉冲输出 0/外部中断 8 输入滤波 2us
78	62	50	PA.6	I/O	INT1	JTAGWDT EN <sup>(注 3)</sup>	外部中断 1 /TEST=0 时, 该引脚功能为 JTAGWDTEN 输入滤波 2us 支持大电流驱动
79	63	51	PE.3	I/O	CLKOUT	PFOUT4	时钟输出/电能脉冲输出 4 TTL 电平输入
80	-	-	PE.4	I/O	RX3	7816_1	UART3 接收/7816_IO 输入滤波 2us
81	-	-	PE.5	I/O	TX3	ADCIN3	UART3 发送/ADC 输入通道 3
82	-	-	PE.6	I/O	TMR1	ADCIN4	PWM 主输出_捕获输入_事件输入/ADC 输入 通道 4 TTL 电平输入
83	64	52	PE.7	I/O	LVDIN0	ADCIN5	外部低电压检测 0/ADC 输入通道 5
84	-	-	PE.8	I/O	INT8	TMR4	外部中断 8/PWM 输出_捕获输入_事件输入
85	65	53	PB.13	I/O	TX6	SWIO	UART6 发送/SW 通讯_SWIO
86	66	54	PB.15	I/O	RX6	SWCLK	UART6 接收/SW 通讯_SWCLK
87	67	55	TEST	I	X	X	测试引脚 (低电平有效, 内部上拉 10K, 含 2us 滤波)
88	68	56	/RST	I	X	X	复位信号 (低电平有效, 内部上拉 88K, 含 2us 滤波)
89	69	57	VDD1P5	PO	X	X	内部数字 1.5V 输出, 需外接 0.1uF 电容滤波
90	70	58	DGND	G	X	X	芯片数字地
91	71	59	DVCC	PI	X	X	芯片数字电源, 需外接滤波电容 10uF&0.1uF
92	72	-	VTP7	I	X	X	测温端子通道 7
93	73	-	VTP6	I	X	X	测温端子通道 6
94	74	-	VTP5	I	X	X	测温端子通道 5
95	75	-	VTP4	I	X	X	测温端子通道 4
96	76	60	VTP3	I	X	X	测温端子通道 3
97	77	61	VTP2	I	X	X	测温端子通道 2
98	78	62	VTP1	I	X	X	测温端子通道 1
99	79	63	VTP0	I	X	X	测温端子通道 0
100	80	64	OSCO	O	X	X	32K 外部时钟输出

注:

1. I=输入; O=输出; PI=电源输入; PO=电源输出; G=地。
2. 红色字体——新增或修改的复用功能;

X——无复用功能

3. JTAGWDTEN 功能无法通过寄存器配置来实现，仅当 TEST=0 时 PA.6 固定为 JTAGWDTEN 功能。
4. 双向 IO 为高阻功能时，输入信号固定零，数字输入/输出无效，只有模拟功能有效。
5. 双向 IO 为 GPIO 功能：
 

若方向寄存器配置为输出，开漏功能(Open Drain)配置有效，上拉功能配置无效，滤波功能配置无效。

若方向寄存器配置为输入，上拉功能配置有效，滤波功能配置有效，开漏功能(Open Drain)配置无效。
6. 双向 IO 为复用功能：
 

复用功能为数字输出时，开漏功能(Open Drain)配置有效，上拉功能配置无效，滤波功能配置无效。

复用功能为数字输入时，上拉功能配置有效，滤波功能配置有效，开漏功能(Open Drain)配置无效。

若为模拟功能，数字输入和输出功能配置无效。
7. 双向 IO 具有模拟复用功能：LVDIN<sub>x</sub>(x=0/1)、ADCIN<sub>x</sub>(x=0/1/3/4/5)。
8. 端口数据寄存器 PTDAT 说明及数据读取规则：
  - 1) 引脚选择 GPIO 功能：
 

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

若方向寄存器配置为输入，PTDAT 读取值为引脚状态值，反映引脚电平状态。
  - 2) 引脚选择复用数字功能：
 

若复用为数字输出功能，且 PTDIR 为 0，则 PTDAT 读取值恒为 0，不随引脚电平变化而变化；

若复用为数字输出功能，且 PTDIR 为 1，则 PTDAT 读取值恒为寄存器设置值，不影响引脚实际输出；

若复用为数字输入功能，且 PTDIR 为 0，则 PTDAT 读取值为引脚状态值，反映引脚电平状态；

若复用为数字输入功能，且 PTDIR 为 1，则 PTDAT 读取值为寄存器设置值，不随引脚电平变化而变化。
  - 3) 引脚选择模拟功能，则该引脚对应的 PTDAT 读取值固定为 0。
9. PA.6 引脚为一个特殊引脚：
 

当 TEST=0 时，该引脚为 JTAGWDTEN 输入功能；

当 TEST=0，且 JTAGWDTEN=0 时，PB.13 和 PB.15 强制为 SW 通讯的 SWIO 和 SWCLK 功能。
10. SWIO、SWCLK 默认为第二复用功能，IOCFG 和 AFCFG 对应 bit 位默认为 1。
11. PJ.0/PJ.1 工作在 VRTC 电压域，默认为高阻态，只有 VRTC 的 POR 复位有效。
12. VCC=5V 时，
 

GPIO 若配为输入上拉，上拉电阻约为 88kΩ；

/RST 引脚固定为输入上拉，上拉电阻约为 88kΩ；

TEST 引脚固定为输入上拉，上拉电阻为 10kΩ。
13. HT7625S 无误差自监测功能，相关管脚为 NC pin（包括：UVREFOA/ UVREFOB/ UVREFOC/ ANA\_RS1/ IREFA/ IREFB/ EWU2P8）。

## 1.5 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

缩略语	英文原文	中文含义
BOR	Brown Out Reset	掉电复位
WKR	Wakeup Reset	唤醒复位
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
EMU	Energy Metering Units	电能计量单元
EWU	eWatching Units	误差在线监测单元

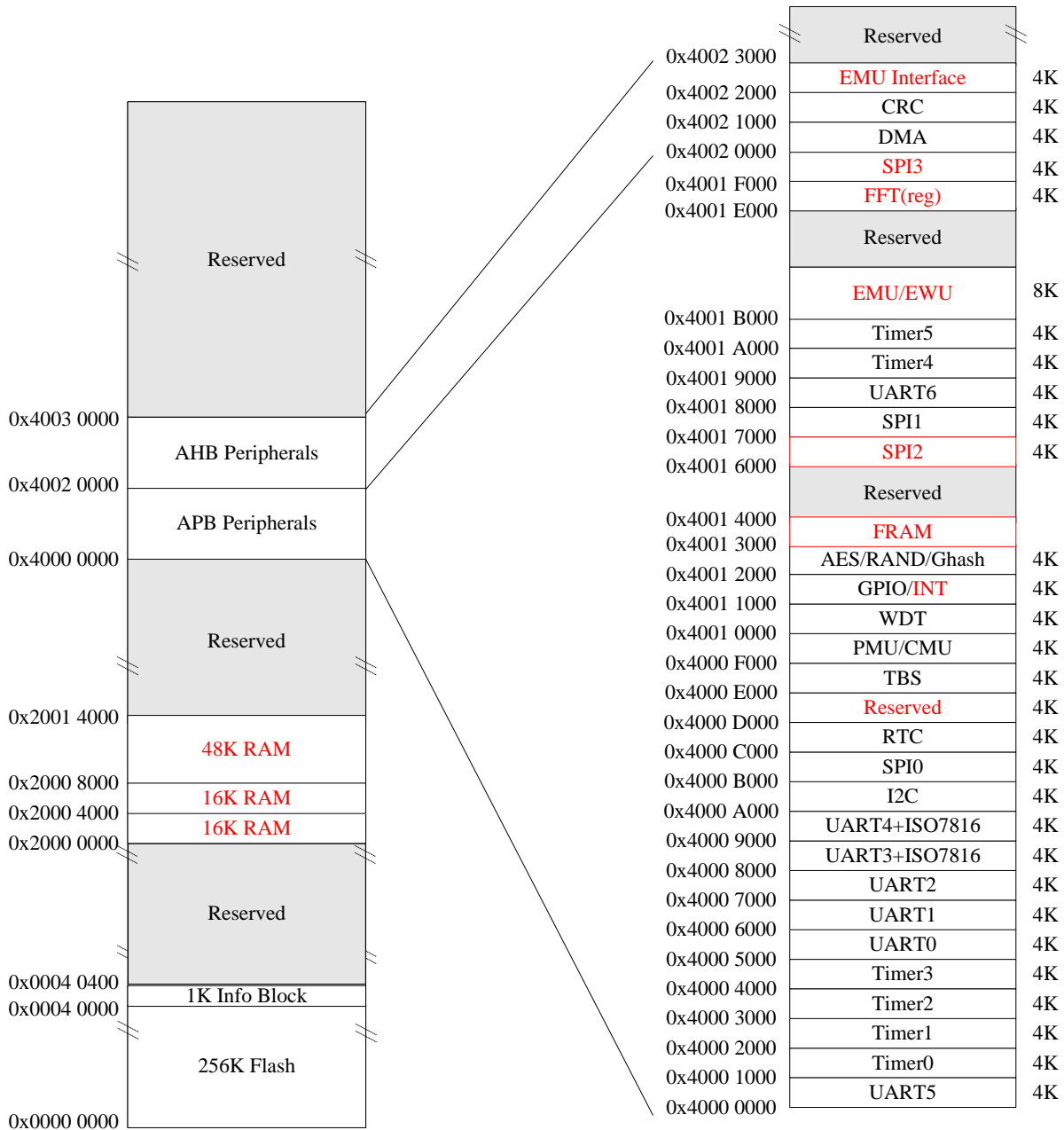
## 2 存储器模块

### 2.1 概述

内置可编程高可靠 256K Flash+ 1K Information Block 和 80KRAM。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
- 擦写次数：100,000 次
- 数据保持时间：20 年 (min)
- 操作温度：-40 度到+105 度（可以保证系统的正常工作温度为-40~85°C）

## 2.2 存储器映射图



注:

- 1) 当使用 FFT 时，会将高 48K (0x20008000-0x20014000) 的 RAM 作为专用 RAM，建议用户谨慎访问此段 RAM 空间，并将堆栈空间配置在前 32K 地址中；当不使用 FFT 时，高 48K 的 RAM 可作为通用 RAM 使用。
- 2) Reset 模块寄存器、唤醒标志寄存器的基地址为 0x4000F400 (同 PMU)。

## 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 0xFF 的值后，开启读保护功能，256K Flash 空间的数据无法读出。在线仿真时需要读取 Flash 内容，导致在线仿真部分功能受限。

注：Flash 加密时，可以通过 SWD 接口读出 0~256 字节的数据。

### 2.3.2 Code Flash 的操作说明

256K Code Flash 可以执行写/页擦除/全擦除操作，说明如下（伪代码举例，后同）。

1. 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，支持字节操作，半字操作，字操作。

宏定义方式：

```
#define M8(adr) (*(uint8_t *) (adr))
```

```
#define M16(adr) (*(uint16_t *) (adr))
```

```
#define M32(adr) (*(uint32_t *) (adr))
```

以上宏定义实现对 Flash 地址 addr 的取址

2. 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
```

```
//当进行字(32bit)写入时， prog_address 以 4 为单位递增
```

```
//当进行半字(16bit)写入时， prog_address 以 2 为单位递增
```

```
//当进行字节 (8bit)写入时， prog_address 以 1 为单位递增
```

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

```
;
```

3. 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
;

4. 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
    
```

## 2.4 Information Block 的操作说明

将 Flash 的 00000FC1H 地址写入非 0xFF 的值后，256K Flash 被加密；Information Block 区不受影响，依旧可以读出。

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

```

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

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

Information Block 共 1024bytes（0x00040000~0x000403FF），分 1 页，1kByte/页，其中地址 0x00040100~0x000401FF 存储有芯片出厂信息，用户若需要修改 information block 中的数据，应先把芯片出厂信息等数据备份再进行“擦—改—写”操作。

## 2.5 Flash 控制功能

微控制器 Auto-Load 时序修改为读取 Flash 三次，根据三次读取的结果选择实际加载的值：

- 1) 三次读取的值均不同时，加载第三次读取的值；
- 2) 前两次一致，第三次不同，则加载第二次读取的值；除此之外的情况（含三次读取的值均不同，即情况 1）加载第三次读取的值。

AutoLoadSTA 用于指示 AutoLoad 是否被正确加载，只有连续三次读取 Flash 的值一致才会被判定为正确

加载, 置起标志位。AutoLoadSTA 置 0 时, 表示三次读取的 Flash 的值不一致, 但仍会按照规则加载 Auto-Load 的值。

微控制器中Flash存储器的FC0H~FC3H区域为选项字节区域。当系统上电或从复位状态重启时, 设备自动装载选项字节, 设置指定的功能。用户需设置以下几项功能:

- LRC开关功能
- RTC补偿系数自动装载功能
- POR/LBOR复位RTC功能
- Flash加密功能
- Sleep/Hold模式下WDT使能功能

Flash选项字控制作用如下:

Flash控制选项字节		地址: 0000FC0H						
	Bit31	30	29	28	27	26	25	Bit24
Read:	X	X	X	X	X	X	X	X
Write:								
	Bit23	22	21	20	19	18	17	Bit16
Read:	SRAMRDFLH[3:0]				WDT_EN[3:0]			
Write:								
	Bit15	14	13	12	11	10	9	Bit8
Read:	FLASH[7:0]							
Write:								
	Bit7	6	5	4	3	2	1	Bit0
Read:	X		X		RTC	AUTORE	LRC_CTR	Reserved
Write:					RST	LOAD	L	

位	功能描述
SRAMRDFLH[3:0]	SRAMRDFLH[3:0] =1111b 时, 程序在 SRAM 空间运行,对 Flash main block 读取不加密 SRAMRDFLH[3:0] !=1111b时, 程序在SRAM空间运行, 对Flash main block读取加密, 读取值为固定0x55555555; 此时Flash擦写状态如下: 1) SRAM加密打开, 此时通过SWD口对0-8K Flash不可页擦和写(用户程序可页擦和写0-8K), 但可进行全擦操作, 可以页擦和写9K-255K。 2) 不通过SWD接口, 而通过用户程序, 即使SRAM加密有打开, 0-255K也可以全擦、页擦和写。
WDT_EN[3:0]	<b>Sleep和Hold模式看门狗使能位</b> 0101b: 看门狗在Sleep和Hold模式下关闭 其他: 看门狗在Sleep和Hold模式下开启
FLASH[7:0]	<b>Flash加密位</b> 0xFF: Flash不加密 其他: Flash加密
RTCST	<b>RTC复位控制位</b> 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

AUTORELOAD	<b>RTC补偿系数装载使能位</b> 1: RTC补偿系数装载功能使能 0: RTC补偿系数装载功能屏蔽
LRC_CTRL	<b>LRC控制位</b> 1: LRC不能被关闭 0: LRC可以被关闭

## 2.6 特殊功能寄存器列表

CMU模块寄存器基地址: 0x4000F000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	WPREG	R/W	0x0000	写保护控制寄存器
0x34	FLASHCON	R/W	0x00	Flash 控制寄存器 (写保护)
0x38	FLASHLOCK	R/W	0x0000	Flash 锁定寄存器
0x50	INFOLOCK	R/W	0x0000	Information Block 锁定寄存器

## 2.7 特殊功能寄存器说明

### 2.7.1 WPREG (写保护寄存器)

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

位	功能描述
WPREG[15:0]	1. 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。 2. 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。 3. 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

### 2.7.2 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> 0: 表示 Flash 空闲, 可以进行操作。 1: 表示 Flash 正在进行写/擦除操作。 只读状态寄存器位, 写入无效。															
FOP[1:0]	<b>FLASH 操作模式选择</b> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>FOP1</th> <th>FOP0</th> <th>FLASH 操作</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>处于 Flash 只读模式</td> </tr> <tr> <td>0</td> <td>1</td> <td>对 STR/STRH 所指 FLASH 区执行 Flash 写操作</td> </tr> <tr> <td>1</td> <td>0</td> <td>对 STR/STRH 所指 FLASH 区执行 Flash 页擦除操作</td> </tr> <tr> <td>1</td> <td>1</td> <td>对 STR/STRH 所指 FLASH 区执行 Flash 全擦除操作</td> </tr> </tbody> </table>	FOP1	FOP0	FLASH 操作	0	0	处于 Flash 只读模式	0	1	对 STR/STRH 所指 FLASH 区执行 Flash 写操作	1	0	对 STR/STRH 所指 FLASH 区执行 Flash 页擦除操作	1	1	对 STR/STRH 所指 FLASH 区执行 Flash 全擦除操作
FOP1	FOP0	FLASH 操作														
0	0	处于 Flash 只读模式														
0	1	对 STR/STRH 所指 FLASH 区执行 Flash 写操作														
1	0	对 STR/STRH 所指 FLASH 区执行 Flash 页擦除操作														
1	1	对 STR/STRH 所指 FLASH 区执行 Flash 全擦除操作														

### 2.7.3 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>								
<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]	<b>Flash 锁定控制位</b> 对该寄存器写入 0x7A68 后, FLASH 被解锁, 用户可以写操作 FLASH。 写入非 0x7A68 数据后, FLASH 被锁定, 用户禁止写操作 FLASH。 默认为锁定状态, Flash 不可执行写/页擦除/全擦除 操作

用户写入的是 0x7A68, 读出值为 1; 写入的是非 0x7A68, 读出值为 0
---

## 2.7.4 INFOLOCK (Information Block 锁定寄存器)

<b>INFOLOCK</b> (Information Block 锁定寄存器)			基地址: 0x4000F000 偏移地址: 50H					
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	KEY[15:8]							
<b>Write:</b>	KEY[15:8]							
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<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>Information Flash 锁定控制位</b></p> <p>对该寄存器写入 0xF998 后, Information block 被解锁, 用户可以写操作 Information FLASH。</p> <p>写入非 0xF998 数据后, <b>Information</b> block 被锁定, 用户禁止写操作 InformationFLASH。</p> <p>默认为锁定状态, <b>Flash</b> 不可执行写/页擦除操作</p> <p>用户写入的是 0xF998, 读出值为 1; 写入的是非 0xF998, 读出值为 0</p>

## 3 时钟单元

### 3.1 时钟分类

测试温度范围：-40°C~85°C

名称	频率	精度	功耗		
			MIN	TYP	MAX
内部低频 RC 时钟 (Flrc)	32.768KHz 注 1	<16%			
内部高频 RC 时钟 (Fhrc)	19.6MHz 注 2	<3%			
外部低频 OSC 晶振 (Fosc)	32.768KHz				
内部 PLL (Fpll)	39.3216MHz	<300ppm			

注 1：内部低频 RC 时钟的范围在：27KHz~38KHz；

注 2：内部高频 RC 时钟在经过 HRCADJ 加载 Info-block 调整值之后，常温精度可达到±1%，全温度范围精度±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(19.6MHz)，系统复位后默认运行在Fhrc。

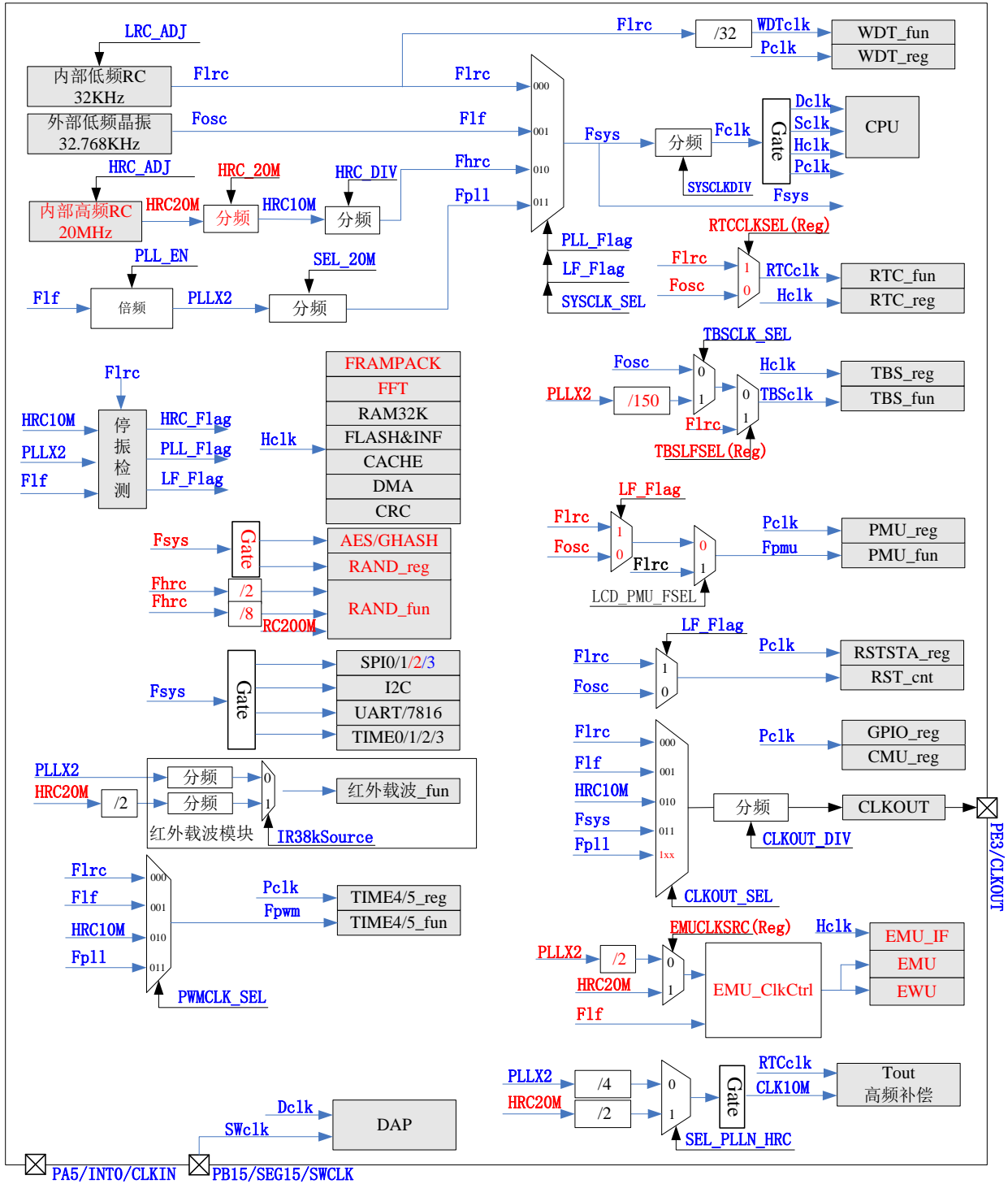
Fosc： 外部低频OSC晶振时钟(32.768KHz)。

Flf： 内部选择的低频时钟（32.768KHz）。

Fpll： 内部 PLL 产生的高频时钟(39.3216MHz)，来源为 Flf。

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 = 外部低频时钟 Fosc。

## 3.3 时钟选择机制说明

### 3.3.1 HRC 19.6M 高频方案

内部高频 RC 时钟频率为 19.6MHz，可以选择该高频 RC 时钟预分频寄存器控制位（控制位为 HRC\_20M 详见时钟框图）作为 9.8MHz 或 19.6MHz 时钟源（默认选择 9.8MHz），其分频输出时钟（分频设置位为 HRC\_DIV[1:0]）作为系统时钟(SYSCLK\_SEL[2:0]=010)

系统复位后，系统时钟默认选择内部高频 RC 时钟(SYSCLK\_SEL[2:0]=010)。HRC\_20M 控制位默认为 0（即选择 9.6MHz 作为时钟源）。

### 3.3.2 RTC 模块时钟选择机制

RTC 模块时钟可通过寄存器 LFCLKCFG 配置为 LRC 时钟。

## 3.4 时钟说明

### 3.4.1 内部低频 RC 时钟（Flrc）

内部低频 RC 时钟振荡频率为 32KHz，提供给看门狗使用，可以选择该低频 RC 时钟作为系统时钟(SYSCLK\_SEL[2:0]=000)。

### 3.4.2 内部高频 RC 时钟（Fhrc）

内部高频 RC 时钟频率为 19.6MHz，可以选择该高频 RC 时钟的分频输出时钟（分频设置位为 HRC\_DIV[1:0]和 HRC\_20M）作为系统时钟(SYSCLK\_SEL[2:0]=010)。

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

当选择内部高频 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）

外接低功耗晶体振荡器，时钟频率 Fosc=32768Hz，以 Fosc 作为系统的内部低频时钟 Flf，芯片内部集成了其震荡所需的电阻和电容。

### 3.4.4 内部 PLL 时钟（Fpll）

内部 PLL 用于对外部低频晶振时钟 Fosc（32.768KHz）倍频（倍频值=1200），以对系统提供最高达 39.3216MHz 的高频时钟 Fpll，PLL 稳定输出后会给出稳定标志位 PLL\_LOCK。在默认情况下（SEL\_20M=0），

PLL 输出 2 分频后的时钟，即  $F_{pll}$  为 19.660800MHz。

如果用户选择  $F_{cpu}$  时钟为 39.3216MHz，则 2T 取指自动打开，客户可根据需求配置 3T 取指功能。

### 3.4.5 时钟源的起振时间

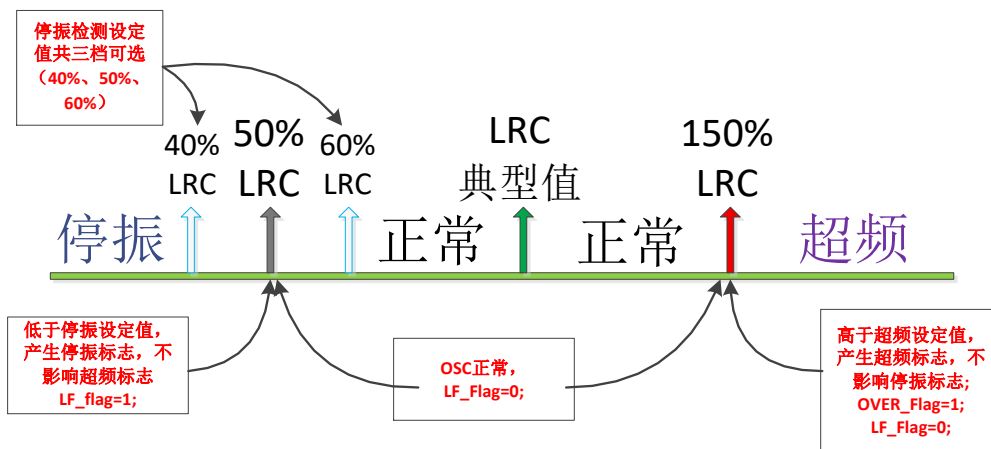
时钟源	起振时间 or 稳定时间
LRC	60us
HRC	6us
LF (外部晶振 32.768KHz)	700ms
PLL	2ms

## 3.5 时钟停振与锁定算法说明

### 3.5.1 OSC 停振检测算法

使用 OSC 统计 LRC（内部低频 RC 时钟源）时钟频率，如果 OSC 频率在设定范围内则 OSC 正常运行且  $LF\_flag$  清 0，低于设定值设定范围则 OSC 停振且  $LF\_flag$  置 1，高出设定值范围则 OSC 时钟超频且  $OVER\_flag$  置 1。具体停振检测原理如下：

- 1、首先使用 OSC 分频后的时钟（64 或 128 分频）统计 LRC 时钟个数。
- 2、如果统计的 OSC 时钟频率大于停振检测设定值，则 OSC 停振锁定周期数加 1，否则 OSC 停振锁定周期数减 1。停振锁定周期数默认为 32，最大值为  $LOCKH+1$ ，最小值为  $LOCKL-1$ 。
- 3、OSC 停振锁定周期数达到最大值  $LOCKH+1$  时，停振标志清零，停振锁定周期数达到最小值  $LOCKL-1$  时，停振标志置 1。
- 4、如果统计的 OSC 时钟频率大于设定值（固定为 LRC 的 150%），则超频标志置 1；反之超频标志清 0。
- 5、具有加速停振检测功能，OSC 经过 128 或 64 分频后的时钟为  $F_1$ ，512 个 LRC 时钟未检测到  $F_1$  时钟翻转，则立刻给出停振标志和停振状态。



**快速停振检测功能说明：**如果 OSC 停振锁定周期个数大于  $LOCKH$  条件满足，测得 OSC 频率低于  $Stop\_Limit$  设定值时（即  $flag\_cnt$  减 1），则产生停振标志  $LF\_Flag=1$ 。该功能可快速检测到 OSC 停振。

**辅助停振标志位 (AUXLF\_Flag):** OSC 停振锁定周期个数等于 LOCKH+1 后, 如果测得 OSC 频率低于 Stop\_Limit 设定值 (即 flag\_cnt 减 1), 则该位置 1。该位写 0 清 0。该位不需要快速停振检测功能使能, 但需要停振检测模块使能, 该模块才工作。

### 3.5.2 PLL 锁定算法

用 PLL 时钟的 16 分频 ( $32768 \times 1200 / 16 = 2457600$ ) 采样 OSC 32768Hz 时钟, 采用滑动连续采样方式, 每次采样值与设置值比较, 保留最近的 N 次比较结果 (N 由 Sumctr[1:0] 寄存器设定)。

OSC	倍频数	PLL 频率(OSC*倍频数)	采样时钟(PLL/16)	采样值中心值 (PLL/16/32768)
32768	1200	39321600	2457600	75

设置值为  $(75 \pm X)$ , X 由寄存器 Tosclen[1:0] 设定, 75 为每次采样理论值。

每次采样值在设置值范围内, 则此次采样值为有效; 若采样值超出设置值范围, 则此次采样值为无效; 当最近 N 次采样中, 3/4 采样值为有效, 则置位锁定标志;

当最近 N 次采样中, 3/4 采样值为无效, 则清除锁定标志。

采样次数(N)	标志置位条件	标志清除条件
16	$\geq 12$ 次有效采样值	$\leq 4$ 次有效采样值
32	$\geq 24$ 次有效采样值	$\leq 8$ 次有效采样值
64	$\geq 48$ 次有效采样值	$\leq 16$ 次有效采样值
128	$\geq 96$ 次有效采样值	$\leq 32$ 次有效采样值

举例如下:

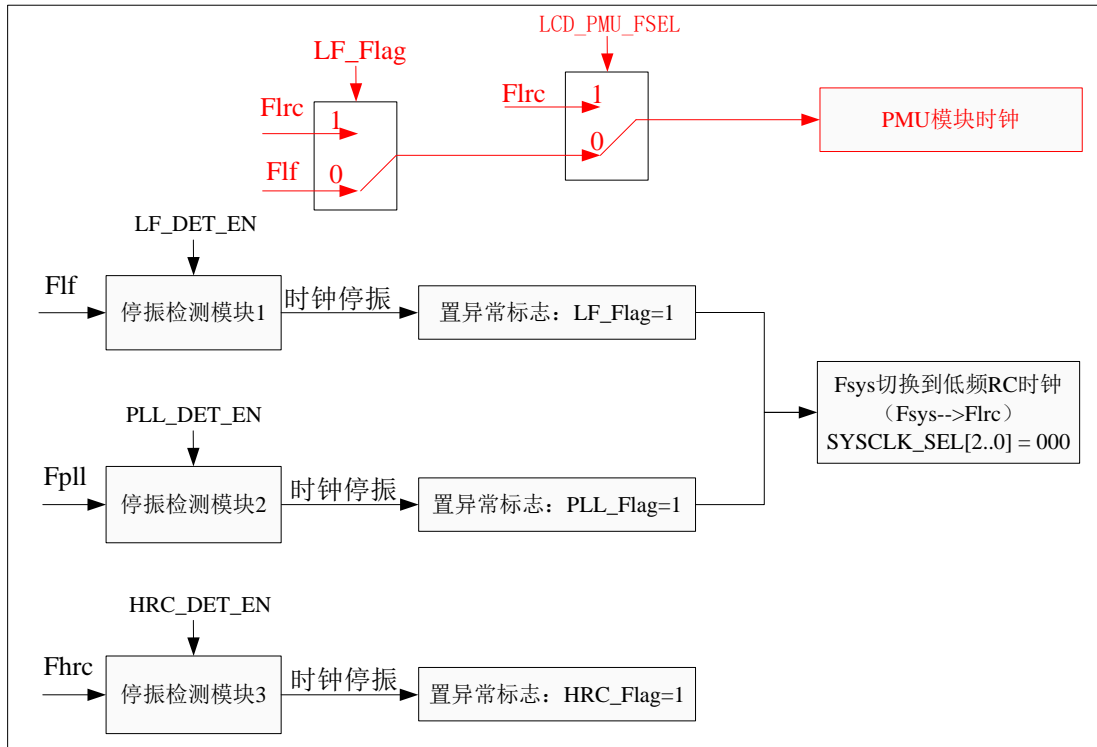
设定采样次数 N=16, 采样值范围为  $75 \pm 2$  (X=2)

则在连续滑动采样过程中,

如果 16 次采样值中, 12 次采样值有效 (采样值在 73-77 (含 73 和 77)), 则置位标志位;

如果 16 次采样值中, 12 次采样值无效 (采样值 < 73 或 采样值 > 77), 则清除标志位;

### 3.5.3 时钟安全机制



时钟停振检测框图

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 Ff，PLL 输出时钟 Fpll，和内部高频时钟 Fhrc 作检测。低频时钟 Ff 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。

时钟停振检测模块的时钟源为内部低频 RC 时钟 Ffrc。

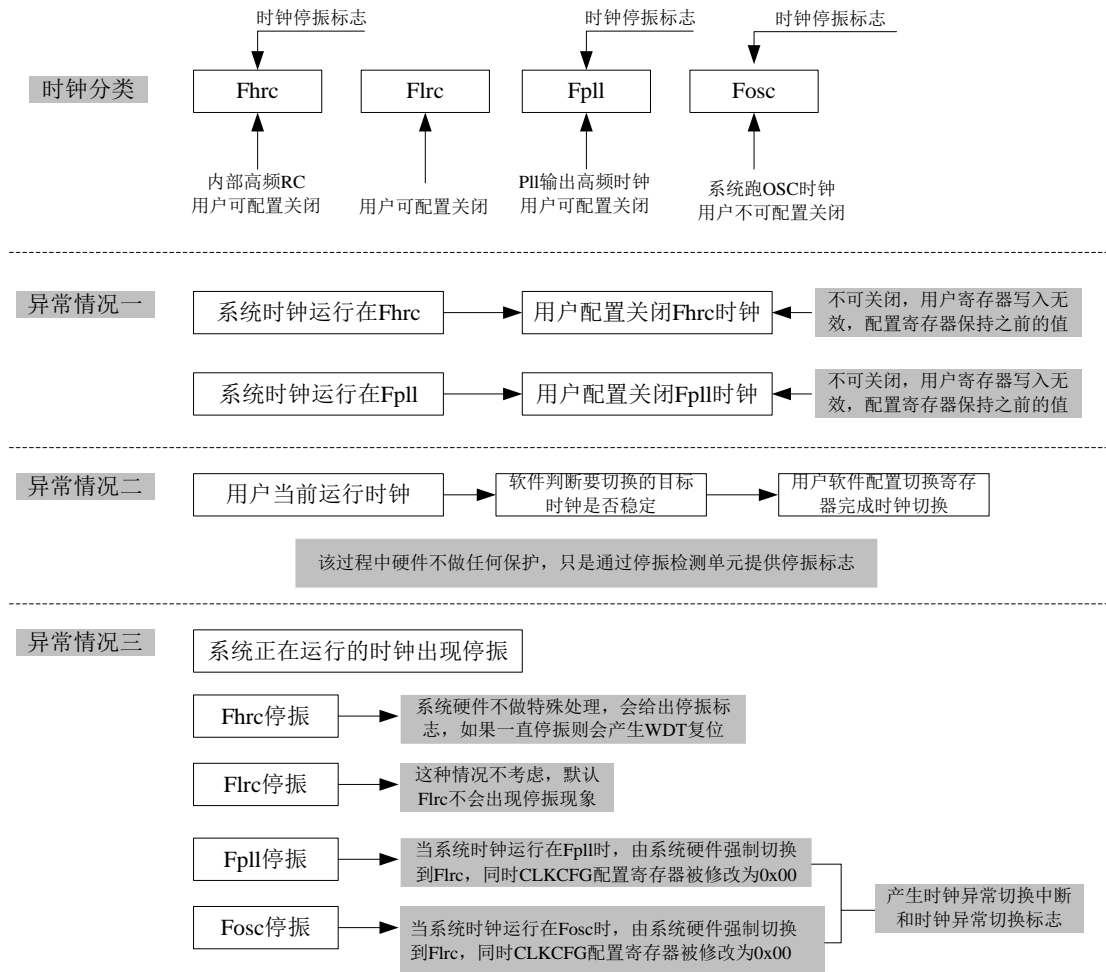
当对应的的时钟停振检测模块功能开启时，内部低频时钟 Ff 发生停振，PLL 时钟 Fpll 发生停振，或内部高频 RC 时钟 Fhrc 发生停振，都会产生相应的时钟故障标志（LF\_FLAG, PLL\_FLAG, HRC\_FLAG）。

当停振检测模块检测到 Ff 停振，系统给出时钟停振标志 LF\_FLAG，如系统时钟 Fsys 选择 Ff 或 Fpll (Ff 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 Ffrc，且产生中断（NMI 中断），同时将寄存器 SYSCLK\_SEL[2:0] 的值置为 000。

当停振检测模块检测到 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 Ffrc，且产生中断（NMI 中断），同时将寄存器 SYSCLK\_SEL[2:0] 的值置为 000。

当停振检测模块检测到 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。

### 3.5.4 时钟异常状态处理



1. 系统运行于低频时钟 Flf 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 Flf 时, 如 Flf 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 Flf 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。

### 3.6 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
0x04	SYSCLKCFG	R/W	0x0002	系统时钟配置寄存器（写保护）
0x08	JTAGSTA	R	0x0001	芯片调试状态指示寄存器
0x0C	LRCADJ	R/W	0x00A5	LRC 时钟调整寄存器（写保护）
0x10	HRCADJ	R/W	0x1840	HRC 时钟调整寄存器（写保护）
0x14	HRCDIV	R/W	0x0001	HRC 时钟分频寄存器（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	0x002CE0	内部模块使能寄存器 0（写保护）
0x30	CLKCTRL1	R/W	0x008000	内部模块使能寄存器 1（写保护）
0x3C	PREFETCH	R/W	0x0260	指令预取使能寄存器
0x40	Reserved	R/W	0x0200	保留寄存器，须保持为默认值
0x200	LFDETCFG	R/W	0x0020	LF 停振检测配置寄存器（写保护）
0x204	HRCCLKCFG	R/W	0x0000	HRC 时钟配置寄存器(写保护)
0x208	LFCLKCFG	R/W	0x0000	低频时钟源配置寄存器（写保护）
0x20C	OSCSTA	R/W	0x0000	OSC 状态寄存器
0xF00	CHIPID	R	0x5320	芯片 ID 寄存器
0xF10	PLLLOCK_PRE	R/W	0x0000	PLL 锁定条件设置寄存器（写保护）
0xF30	FLTCTR	R/W	0x0000	时钟滤波控制寄存器

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

## SYCLKCFG (系统时钟配置寄存器, 写保护)

SYCLKCFG (写保护) (系统时钟配置寄存器)		基地址: 0x4000F000 偏移地址: 04H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	WCLKE	X	X	X	X	SYSCLK_SEL[2:0]		
Write:	N							
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-top: 10px;"> <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>Ffrc</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Ffif</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Ffrc (default)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Fpll</td> </tr> <tr> <td>1</td> <td>X</td> <td>X</td> <td>保留, 请勿配置</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 Ffrc (SYSCLK_SEL[2:0]=010)。</li> <li>当选择内部低频时钟 Ffif 作为系统时钟时, 检测到低频时钟 Ffif 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 Ffrc, 同时系统时钟控制位 SYSCLK_SEL[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 Ffif) 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 Ffrc, 同时系统时钟控制位 SYSCLK_SEL[2:0] 被置为 000。</li> </ol>	SYSCLK_SEL[2:0]			系统时钟选择 Fsys	0	0	0	Ffrc	0	0	1	Ffif	0	1	0	Ffrc (default)	0	1	1	Fpll	1	X	X	保留, 请勿配置
SYSCLK_SEL[2:0]			系统时钟选择 Fsys																						
0	0	0	Ffrc																						
0	0	1	Ffif																						
0	1	0	Ffrc (default)																						
0	1	1	Fpll																						
1	X	X	保留, 请勿配置																						

	5. 当选择 PLL 做系统时钟时，必须先打开 PLL 时钟（CLKCTRL0 的 bit4）
--	---

### JTAGSTA（芯片状态指示寄存器）

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

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

注: bit0 的值默认为 1, 用户无须更改该位。

### LRCADJ（LRC 时钟调整寄存器，写保护）

LRCADJ (写保护) (LRC 时钟调整寄存器)			基地址: 0x4000F000 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	LRCADJ[7:0]							
Write:	LRCADJ[7:0]							
Reset:	1	0	1	0	0	1	0	1

位	功能描述
LRCADJ[7:0]	LRC 输出频率调节控制位

## HRCADJ (HRC 时钟调整寄存器, 写保护)

HRCADJ (写保护) (HRC 时钟调整寄存器)			基地址: 0x4000F000 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	HRCADJ[13:8]					
Write:								
Reset:	0	0	0	1	1	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	HRCADJ[6:0]						
Write:								
Reset:	0	1	0	0	0	0	0	0

位	功能描述
HRCADJ[13:8]	HRCADJ[13:8]存储于 Info flash 地址 0x00040140。
HRCADJ[6:0]	<b>HRC 输出频率调节控制位</b> 芯片出厂时, HRC 已经过校准, 校准之后频率为 9.8MHz。HRCADJ 调整值存储于 Info flash 地址 0x00040140。默认配置下, 芯片可正常启动运行。如果加载 Info flash 相应的调整值到 HRCADJ, 可提高 HRC 频率的准确性, 详见 FAQ

## HRCDIV (HRC 时钟分频寄存器, 写保护)

HRCDIV (写保护) (HRC 时钟分频寄存器)			基地址: 0x4000F000 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:	X	X	X	X	X	X	HRC_DIV[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	1

位	功能描述
---	------

HRC_DIV[1:0]	<b>HRC 时钟分频设置:</b>		
	<b>HRC_DIV[1:0]</b>		<b>RC分频后的时钟Fhrc'</b>
	0	0	Fhrc
	0	1	Fhrc/2 (default)
	1	0	Fhrc/4
	1	1	Fhrc/8

## CLKSTA (时钟状态寄存器)

<b>CLKSTA</b> (时钟状态寄存器)		<b>基地址: 0x4000F000</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	PLL_LOCK	PLL_FLAG	HRC_FLAG	X	X	LF_FLAG
<b>Write:</b>			X	X	X			X
<b>Reset</b>	0	0	0	0	0	0	0	0

注: 此寄存器是只读状态寄存器

位	功能描述
PLL_LOCK	<b>PLL 时钟锁定状态标志寄存器</b> 0: PLL 时钟锁定异常 1: PLL 时钟锁定正常 注: 用来指示芯片内部 PLL 的工作稳定状态, 用户也可以打开 PLL 后等待 4ms 来判断其稳定。关闭 PLL_EN, PLL_LOCK 立刻清零。 注 2: 由于 PLL_LOCK 通过 PLL 时钟更新, 当 PLL 无时钟时, PLL_LOCK 不更新, 保持不变。
PLL_FLAG	<b>PLL 时钟 Fpll 停振标志</b> 0: 正常 1: 停振
HRC_FLAG	<b>内部高频 RC 时钟 Fhrc 停振标志</b> 0: 正常 1: 停振
LF_FLAG	<b>内部低频时钟 Flf 停振标志</b> 0: 正常 1: 停振

**SYSCCLKDIV (系统时钟分频寄存器, 写保护)**

<b>SYSCCLKDIV (写保护)</b> (系统时钟分频寄存器)			<b>基地址: 0x4000F000</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>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	X	X	X	X	SEL_20	CPUCLK_DIV[2:0]		
<b>Write:</b>					M			
<b>Reset:</b>	0	0	0	0	0	0	0	1

位	功能描述																																				
SEL_20M	PLL输出时钟预分频寄存器控制位: 0: PLL 输出时钟 2 分频, 即 Fpll=19.660800MHz (default) 1: PLL 输出时钟不分频, 即 Fpll=39.3216MHz																																				
CPUCLK_DIV[2:0]	<b>CPU 时钟分频设置:</b> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="3">CPUCLK_DIV [2:0]</th> <th>CPU时钟选择(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> <p>注: 如果选择 CPU 时钟为 39.3M, 则 2T 取指自动打开, 客户可根据需求配置 3T 取指功能。</p>	CPUCLK_DIV [2:0]			CPU时钟选择(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
CPUCLK_DIV [2:0]			CPU时钟选择(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)**

<b>CLKOUTSEL (写保护)</b> (CLKOUT 时钟选择寄存器)			<b>基地址: 0x4000F000</b> <b>偏移地址: 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	X	X	CLKOUT_SEL[2:0]		
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	1	0

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

### CLKOUTDIV (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

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

### 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	SEL_PL
<b>Write:</b>								LN_HRC
<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>	CRC_EN	ARG_E	1P5LBO	CLKOU	Reserved	Reserved	HRC_DE	PLL_DE
<b>Write:</b>		N	R_EN	T_EN			T_EN	T_EN
<b>Reset:</b>	0	0	1	0	1	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	SPIO_EN	Reserved	LCD_P
<b>Write:</b>								
<b>Reset:</b>	1	1	1	0	0	0	0	0

位	功能描述
SEL_PLLN_HRC	<b>RTC 高频补偿时钟选择</b> 0: PLL(default) 1: HRC
CRC_EN	<b>CRC 模块时钟使能位</b> 0: 关闭(default) 1: 打开
ARG_EN	<b>AES/RAND/GHASH 模块时钟使能位</b> 0: 关闭 (default) 1: 打开
1P5LBOR_EN	<b>LDO 内部 LBOR 使能位 (用户不要修改这个寄存器位)</b> 0: 关闭 1: 打开 (default)
CLKOUT_EN	<b>CLKOUT 使能信号</b> 0: 关闭 (default) 1: 打开
HRC_DET_EN	<b>HRC 时钟检测模块控制位</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 时钟检测模块控制位</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 时钟检测模块控制位</b> 0: 关闭 LF 时钟停振检测单元 1: 使能 LF 时钟停振检测单元 (default)
PLL_LOCK_EN	<b>PLL 强制锁定使能控制位</b> 0: PLL 不强制锁定 1: PLL 强制锁定 (default) 注意: 当使能该位且 PLL_EN=1 时, 相应的 PLL 锁定标志位

	PLL_LOCK 固定为 1, 反之锁定标志位 PLL_LOCK 将反应 PLL 的实际工作状态。
HRC_EN	<b>HF RC 时钟振荡器使能位</b> 0: 关闭高频 RC 时钟模块; 1: 使能高频 RC 时钟模块; (default) 注意: 当用户选择 Fsys 为 Fhrc 时, 此时不能关闭 HRC_EN, 该寄存器位写入无效
PLL_EN	<b>PLL 模块时钟使能位</b> 0: 关闭 PLL 模块 (default) 1: 使能 PLL 模块 注意: 1. 当用户切换至 PLL 时钟作为系统时钟时, 必须先打开 PLL_EN。 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
SPIO_EN	<b>SPIO 模块时钟使能位</b> 0: 关闭 SPIO 模块 (default) 1: 使能 SPIO 模块
LCD_PMU_FSEL	<b>PMU 模块时钟源选择控制位</b> 0: PMU 模块时钟源由 LF_Flag 停振标志控制 (LF_Flag=0: OSC 时钟, =1: LRC 时钟) (default) 1: PMU 模块时钟源为 LRC 时钟

## CLKCTRL1 (内部模块使能控制寄存器 1, 写保护)

CLKCTRL1 (写保护) (内部模块使能控制寄存器 1)		基地址: 0x4000F000 偏移地址: 30H						
	Bit31	30	29	28	27	26	25	Bit24
Read:	X	X	X	X	X	PACK_EN	FFT_EN	EMUIF_EN
Write:						N		EN
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	EMUCLKSRC	SPI3_EN	X	SPI2_EN	EMU_EN	X	Reserved
Write:						N		
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	Reserved	SOFTWDT_EN	SPI1_EN	UART6_EN	TMR5_EN	TMR4_EN	UART5_EN	UART4_7816_EN
Write:					N	N		
Reset:	1	0	0	0	0	0	0	0

	Bit7	6	5	4	3	2	1	Bit0
<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
PACK_EN	<b>帧打包模块时钟使能</b> 0: 关闭 (default) 1: 打开
FFT_EN	<b>FFT 模块时钟使能</b> 0: 关闭 (default) 1: 打开
EMUIF_EN	<b>EMU Interface 模块时钟使能</b> 0: 关闭 (default) 1: 打开
EMUCLKSRC	<b>EMU 时钟选择控制位</b> 0: EMU 时钟源为 PLL (default) 1: EMU 时钟源为 HRC 注: EMU 时钟建议优先选择 PLL, 如果 PLL 或者 OSC 停振再切换到 HRC。
SPI3_EN	<b>SPI3 时钟使能位</b> 0: 关闭 1: 使能
SPI2_EN	<b>SPI2 时钟使能位</b> 0: 关闭 1: 使能
EMU_EN	<b>EMU 模块时钟使能</b> 0: 关闭 (default) 1: 打开
SOFTWDT_EN	<b>调试模式下看门狗使能位</b> 0: 调试模式下看门狗关闭 (default) 1: 调试模式下看门狗打开
SPI1_EN	<b>SPI1 时钟使能位</b> 0: 关闭 1: 使能
UART6_EN	<b>UART6 时钟使能位</b> 0: 关闭 1: 使能
TMR5_EN	<b>Timer5 时钟使能位</b> 0: 关闭 1: 使能
TMR4_EN	<b>Timer4 时钟使能位</b> 0: 关闭 1: 使能

UART5_EN	<b>UART5 时钟使能位</b> 0: 关闭 1: 使能
UART4_7816_EN	<b>UART4 时钟使能位</b> 0: 关闭 1: 使能
UART3_7816_EN	<b>UART3 时钟使能位</b> 0: 关闭 1: 使能
UART2_EN	<b>UART2 时钟使能位</b> 0: 关闭 1: 使能
UART1_EN	<b>UART1 时钟使能位</b> 0: 关闭 1: 使能
UART0_EN	<b>UART0 时钟使能位</b> 0: 关闭 1: 使能
TMR3_EN	<b>Timer3 时钟使能位</b> 0: 关闭 1: 使能
TMR2_EN	<b>Timer2 时钟使能位</b> 0: 关闭 1: 使能
TMR1_EN	<b>Timer1 时钟使能位</b> 0: 关闭 1: 使能
TMR0_EN	<b>Timer0 时钟使能位</b> 0: 关闭 1: 使能

注：CRC/AES\_RAND\_GHASH/I2C/SPI0-3/UART0-6/TMR0-5 这些模块需先配置对应的时钟使能位，才可以读写模块内寄存器。

## PREFETCH（指令预取使能控制寄存器）

PREFETCH （指令预取使能控制寄存器）			基地址： 0x4000F000 偏移地址： 3CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	Reserved	2T_EN
Write:								
Reset:	0	0	0	0	0	0	1	0

	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	3T_EN	Reserved	Reserved	Reserved	X	Reserved	Reserved	Reserved
<b>Write:</b>								
<b>Reset:</b>	0	1	1	0	0	0	0	0

位	功能描述
Reserved	该寄存器所有保留控制位请保持为默认值
2T_EN	<b>CPU 2T 取指控制位</b> 0: 关闭 Flash access 2T (default) 1: 使能 Flash access 2T 注: 1.如果 3T_EN 和 2T_EN 同时开启, 则使能 Flash access 3T 2.Fcpu=39.3MHz, Flash Access 2T 自动生效
3T_EN	<b>CPU 3T 取指控制位</b> 0: 关闭 Flash access 3T (default) 1: 使能 Flash access 3T

### LFDETCFG (LF 停振检测配置寄存器, 写保护)

LFDETCFG (写保护) (LF 停振检测配置寄存器)	基地址: 0x4000F000 偏移地址: 200H							
	<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	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
<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>	QuickSto	Stop_Limit[1:0]		Reserved		Stop_Num[1:0]		DETCL
<b>Write:</b>	p							KDIV
<b>Reset:</b>	0	0	1	0	0	0	0	0

位	功能描述								
QuickStop	<b>快速停振检测控制位</b> =0: 关闭 (default) =1: 使能 注: 该位使能且 OSC 停振锁定周期个数大于 LOCKH, 如果测得 OSC 频率低于 Stop_Limit 设定值, 则产生停振标志 LF_Flag=1。								
Stop_Limit[1:0]	<b>OSC 停振检测设定值</b> <table border="1" data-bbox="391 1832 986 2000"> <tr> <td>Stop_Limit[1:0]</td> <td>LRC 频率百分比</td> </tr> <tr> <td>00</td> <td>40% LRC</td> </tr> <tr> <td>01 (default)</td> <td>50% LRC</td> </tr> <tr> <td>10</td> <td>60% LRC</td> </tr> </table>	Stop_Limit[1:0]	LRC 频率百分比	00	40% LRC	01 (default)	50% LRC	10	60% LRC
Stop_Limit[1:0]	LRC 频率百分比								
00	40% LRC								
01 (default)	50% LRC								
10	60% LRC								

	11	切换为旧模式停振检测规则															
	注：如果测得 OSC 频率低于 Stop_Limit 设定值，则停振锁定周期个数减 1，否则停振锁定周期个数加 1。																
Stop_Num[1:0]	<b>OSC 停振锁定周期个数设定</b> <table border="1" style="width: 100%;"> <tr> <td>Stop_Num[1:0]</td> <td>LOCKL</td> <td>LOCKH</td> </tr> <tr> <td>00 (default)</td> <td>28</td> <td>36</td> </tr> <tr> <td>01</td> <td>24</td> <td>40</td> </tr> <tr> <td>10</td> <td>20</td> <td>44</td> </tr> <tr> <td>11</td> <td>16</td> <td>48</td> </tr> </table>		Stop_Num[1:0]	LOCKL	LOCKH	00 (default)	28	36	01	24	40	10	20	44	11	16	48
Stop_Num[1:0]	LOCKL	LOCKH															
00 (default)	28	36															
01	24	40															
10	20	44															
11	16	48															
	注：锁定周期与 DETCLKDIV 选择有关，DETCLKDIV 选择 0，OSC 停振锁定周期为：OSC/64；DETCLKDIV 选择 1，OSC 停振锁定周期为：OSC/128。停振检测周期个数等于 LOCKL-1，则 LF_FLAG 建立，周期个数等于 LOCKH+1，则 LF_FLAG 清 0。																
DETCLKDIV	<b>检测时钟分频选择位</b> =0: 64 (default) =1: 128																

注：唤醒复位、调试和软复位、看门狗复位不能复位 LFDETCFG, BOR、EXRST、LBOR、POR、自定义软复位可以复位该寄存器。

## HRCCLKCFG (HRC 时钟配置寄存器, 写保护)

HRCCLKCFG (写保护)		基地址: 0x4000F000						
(HRC 时钟配置寄存器)		偏移地址: 204H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	Reserve	Reserve	Reserve	Reserve	Reserve	Reserved	HRC_20M
Write:		d	d	d	d	d		
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Reserve	Reserve	Reserve	Reserve	Reserve	Reserve	Reserved	Reserved
Write:	d	d	d	d	d	d		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
HRC_20M	<b>HRC 输出时钟预分频寄存器控制位</b> =0:HRC 输出时钟 2 分频 (default) =1:HRC 输出时钟不分频

注：唤醒复位、调试和软复位、看门狗复位 BOR、EXRST、LBOR、POR、自定义软复位可以复位该寄存器。

**LFCLKCFG (低频时钟源配置寄存器, 写保护)**

LFCLKCFG (写保护) (低频时钟源配置寄存器)			基地址: <b>0x4000F000</b> 偏移地址: <b>208H</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>	<b>X</b>	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
<b>Write:</b>								
<b>Reset:</b>	<b>0</b>	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>	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TBSLFSE	RTCCLK
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	<b>0</b>	0	0

位	功能描述
TBSLFSEL	TBS 模块时钟配置位 =0: 由 TBSCLKSEL 选择时钟 (default) =1: LRC 注: TBSCLKSEL=1,选择 PLL 时钟源, TBSCLKSEL=0 选择 OSC 时钟源。
RTCCLKSEL	RTC 模块时钟配置位 =0: OSC (default) =1: LRC

注: 唤醒复位、调试和软复位、看门狗不能复位 LFCLKCFG, BOR、EXRST、LBOR、POR、自定义软复位可以复位该寄存器。

**OSCSTA (OSC 状态寄存器)**

OSCSTA (OSC 状态寄存器)			基地址: <b>0x4000F000</b> 偏移地址: <b>20CH</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>	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
<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>	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	AUXL	OVER_Flag
<b>Write:</b>								F_Flag
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
AUXLF_Flag	OSC 辅助停振标志位

	0: 表示 OSC 未发生停振 1: 表示 OSC 发生停振。 注: OSC 停振锁定周期个数大于 LOCKH 后, 如果测得 OSC 频率低于 Stop_Limit 设定值, 则该位置 1。该位写 0 清 0。
OVER_Flag	<b>OSC 超频标志位</b> 0: 表示 OSC 未发生超频 1: 表示 OSC 发生超频。

注:

1. 唤醒复位、调试和软复位、看门狗复位不能复位 OSCSTA, BOR、EXRST、LBOR、POR、自定义软复位可以复位该寄存器。
2. OSCSTA 为状态寄存器, 因此不受新增功能写保护控制位和新增功能生效控制位的影响。

## CHIPID (芯片版本寄存器)

<b>CHIPID</b> (芯片版本寄存器)		基地址: 0x4000F000 偏移地址: F00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	CHIPID[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	1	0	1	0	0	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	CHIPID[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	1	0	0	0	0	0

位	功能描述
CHIPID[15:0]	0x5320

## PLLLOCK\_PRE (PLL 锁定条件设置寄存器, 写保护)

PLLLOCK_PRE (写保护) (PLL 锁定条件设置寄存器)		基地址: 0x4000F000 偏移地址: F10H						
	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	Tosclenct	Tosclenct	X	X	Sumctr[1]	Sumctr[0]

Write:			r[1]	r[0]			1	1
Reset:	0	0	0	0	0	0	0	0

位	功能描述															
Tosclenctr[1:0]	<p>每个 OSC 周期内有效的 PLL 锁定的判断值的设置： PLL 时钟为 32768*75*16,</p> <table border="1"> <thead> <tr> <th>Tosclen [1:0]</th> <th>Tosclen-min</th> <th>Tosclen-max</th> </tr> </thead> <tbody> <tr> <td>00 (default)</td> <td>73</td> <td>77</td> </tr> <tr> <td>01</td> <td>71</td> <td>79</td> </tr> <tr> <td>10</td> <td>69</td> <td>81</td> </tr> <tr> <td>11</td> <td>67</td> <td>83</td> </tr> </tbody> </table> <p>此值用来与每个 OSC 周期内 pll 16 分频时钟的个数比对。如果 pll 时钟个数一个 OSC 周期内累加值位于 min,max 之间, 则认为此 OSC 周期内 PLL 趋于锁定状态。</p>	Tosclen [1:0]	Tosclen-min	Tosclen-max	00 (default)	73	77	01	71	79	10	69	81	11	67	83
Tosclen [1:0]	Tosclen-min	Tosclen-max														
00 (default)	73	77														
01	71	79														
10	69	81														
11	67	83														
Sumctr[1:0]	<p>PLL 锁定的 OSC 周期个数设定</p> <table border="1"> <thead> <tr> <th>Sumctr[1:0]</th> <th>Cntsum</th> </tr> </thead> <tbody> <tr> <td>00 (default)</td> <td>16</td> </tr> <tr> <td>01</td> <td>32</td> </tr> <tr> <td>10</td> <td>64</td> </tr> <tr> <td>11</td> <td>128</td> </tr> </tbody> </table>	Sumctr[1:0]	Cntsum	00 (default)	16	01	32	10	64	11	128					
Sumctr[1:0]	Cntsum															
00 (default)	16															
01	32															
10	64															
11	128															

## FLTCTR (时钟滤波控制寄存器)

FLTCTR (时钟滤波控制寄存器)		基地址: 0x4000F000 偏移地址: F30H						
	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:	PLL_CTR[1:0]		Reserved		LRC_CTR[1:0]		OSC_CTR[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PLL_CTR[1:0]	PLL 时钟滤波控制位 00: 无 (default) 01: 低 10: 中 11: 中
Reserved	保留位, 用户须保持为默认值

LRC_CTR[1:0]	LRC 时钟滤波控制位 00: 无 (default) 01: 低 10: 中 11: 高
OSC_CTR[1:0]	OSC 时钟滤波控制位 00: 无 (default) 01: 低 10: 中 11: 高

注：唤醒复位、调试和软复位、看门狗复位 BOR、EXRST、LBOR、POR、自定义软复位可以复位该寄存器。

## 4 电源单元

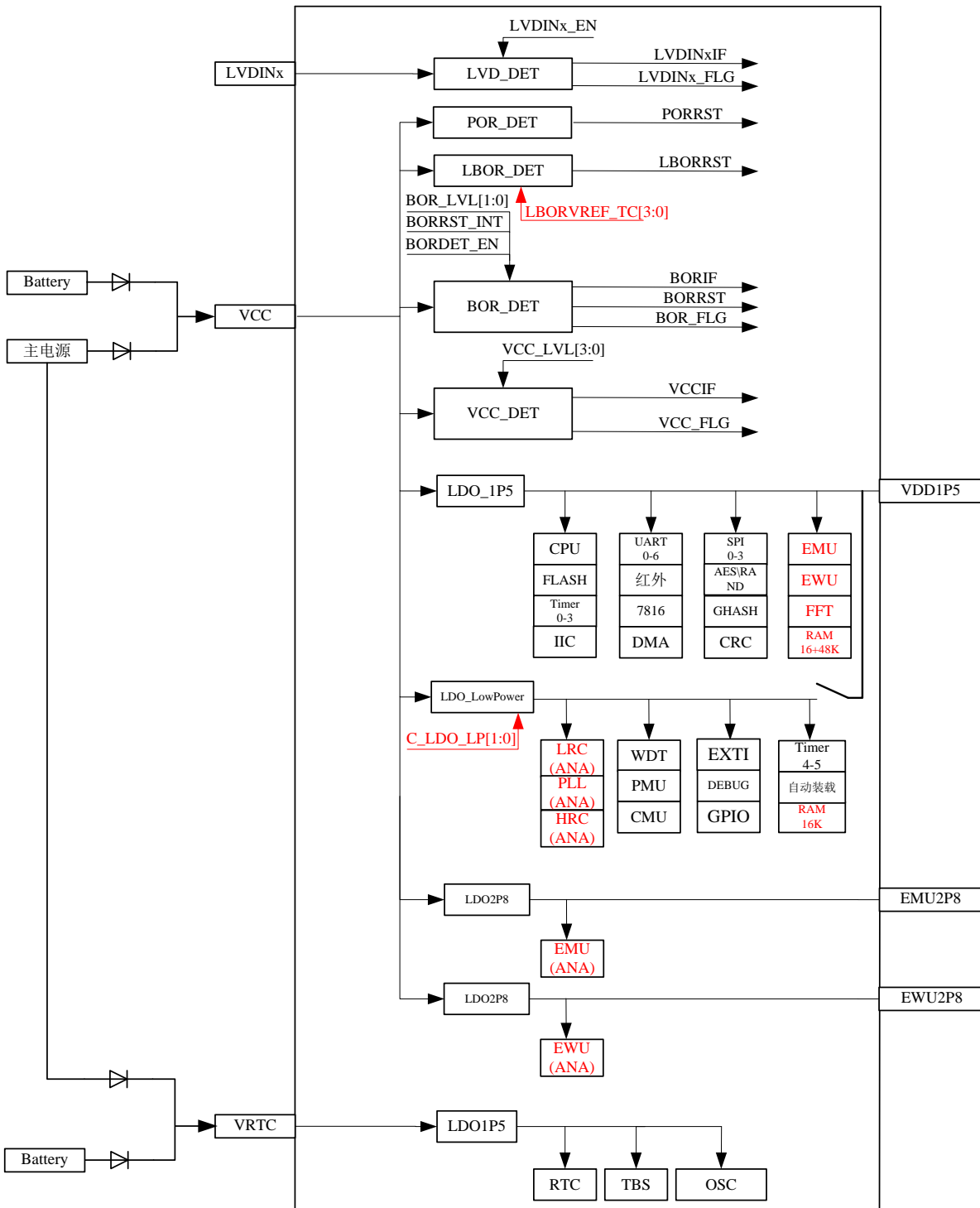
### 4.1 概述

PMU 为芯片的电源管理单元，功能如下：

- 监测低电压检测输入端 LVDIN0, LVDIN1, 当供电电压低于或高于设定阈值时产生中断信号。
- 监测系统电源 VCC, 可以根据设定阈值产生 BOR、LBOR、POR 复位信号。
- 测量芯片 ADCIN0, ADCIN1, **TDVREF**, ADCIN3, ADCIN4, ADCIN5 的电压。
- 为芯片内部数字模块提供 1.5V 电源

（注：VCC, ADCIN0, ADCIN1, **TDVREF**, ADCIN3, ADCIN4, ADCIN5 测量值寄存器在 TBS 单元）

### 4.2 框图



芯片内部供电框图

注:

- 1) LVDIN<sub>x</sub> 指 LVDIN0、LVDIN1
- 2) 内部 LP\_SWOFFN 信号控制 LDO\_1P5 和 LDO\_LowPower 之间的开关, LP\_SWOFFN=1, 开关闭合,

**LP\_SWOFFN=0**，开关断开。**Normal** 和 **Hold** 模式下，该开关保持闭合，**Sleep** 模式下该开关断开。若已开启了 **PLL** 或 **HRC** 时钟，则此开关保持闭合，**LDO\_1P5** 保持供电。

## 4.3 电源单元详细功能说明

### 4.3.1 电源切换

芯片供电电源的切换由芯片外部电路来完成。一般将5V主电源和3.6V电池通过两个二极管并联后输入到芯片的VCC和VRTC引脚，来实现电源的无缝切换。

### 4.3.2 电源实时监测

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

- **LVDIN\_DET**模块：  
监测外部引脚LVDIN的电压，当电压低于或高于1.21V时，置位LVDINIF标志位；如果使能了LVDINIE中断，就会产生LVDIN中断。
- **VCC\_DET**模块：  
监测系统电源VCC的电压，当电压低于或高于设定阈值时，置位VCCIF标志位；如果使能VCC检测中断(VCCIE)，就会产生VCC中断；检测阈值可通过寄存器VDETCFG的位VCC\_LVL[3:0]来设置。
- **BOR\_DET**模块：  
监测系统电源VCC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BOR模块产生中断信号(BORRST=0)，同时使能BOR检测中断(BORIE)，就会产生BOR中断；如果设置BOR模块产生复位信号(BORRST=1)，当电压低于设定阈值时，将会立即产生BOR复位。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VCC的电压，当电压低于阈值1.7V时，产生LBOR掉电复位。
- **POR\_DET**模块：  
监测系统电源VCC的电压，当电压上升到阈值0.3V时，产生POR上电复位。

### 4.3.3 内建 1.5V 电源

芯片内部通过子模块 VREG 将 VCC 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 VDD1P5 输出。在输出引脚 VDD1P5 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)

如果用户设置 BOR 模块产生复位信号 (BORRST=1)，当 BOR 检测模块检测到系统电源 VCC 低于设定电压  $V_{bor}$  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

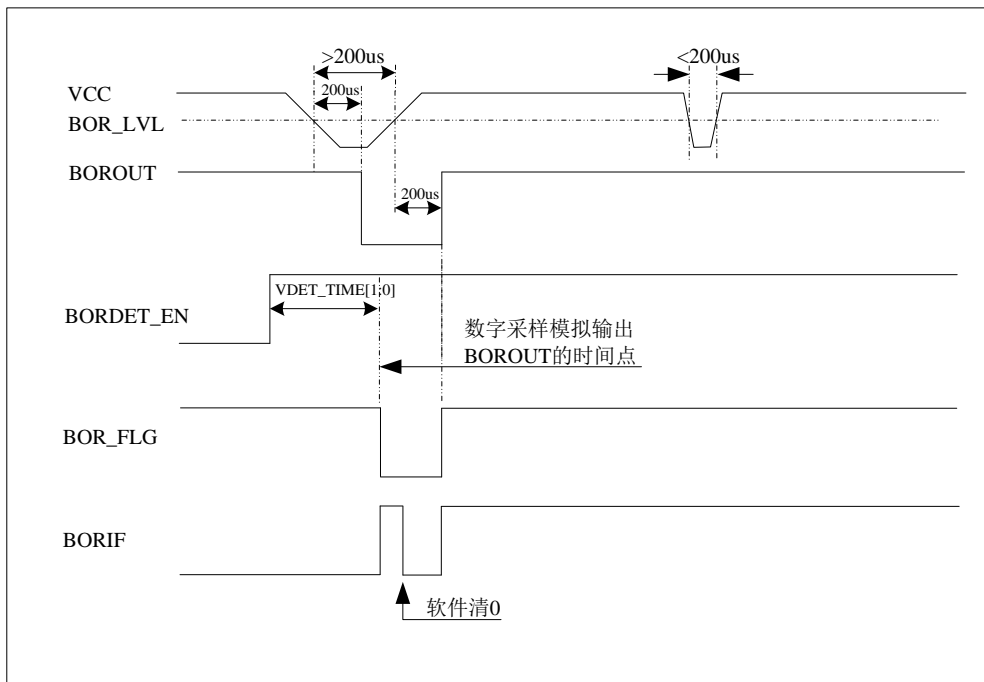
态寄存器 RSTSTA 的 BOR 标志位被置为 1。当 BOR 检测模块检测到系统电源 VCC 电压高于设定电压  $V_{bor}$  时，BOR 检测模块内部信号 BOROUT 输出高电平，在该高电平持续的 1024 个 Flrc 周期后，内部复位信号 IRST 也变为高电平。

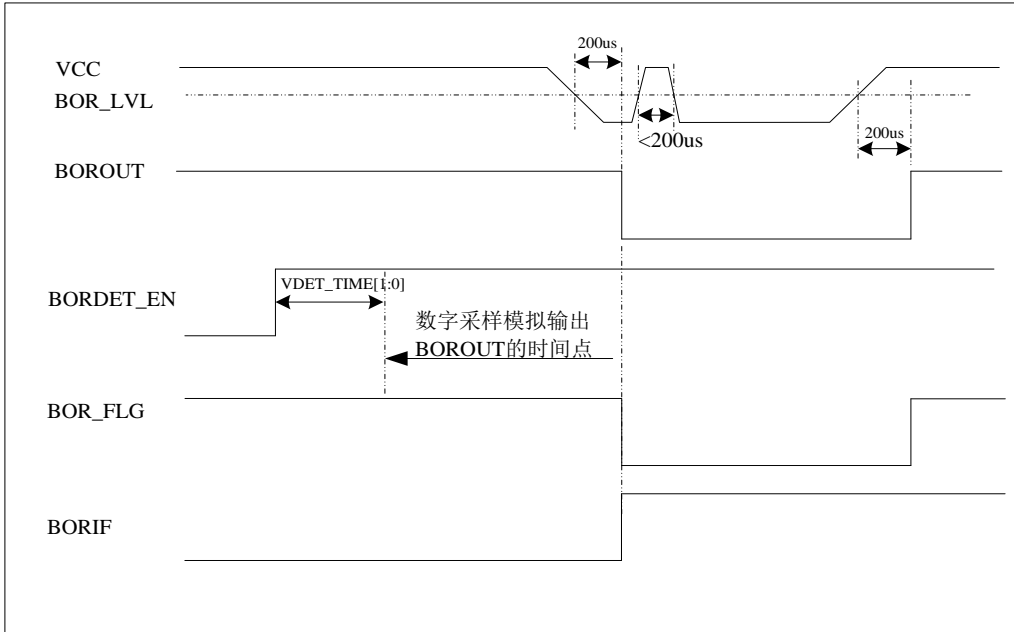
$V_{bor}$  具有迟滞特性，迟滞电压为 200mV，BOR 模块的检测阈值  $V_{bor}$  可通过 VDETCFG 中的 BOR\_LVL[1:0] 设置。

掉电复位 BOR 产生时，下面的事件将会发生：

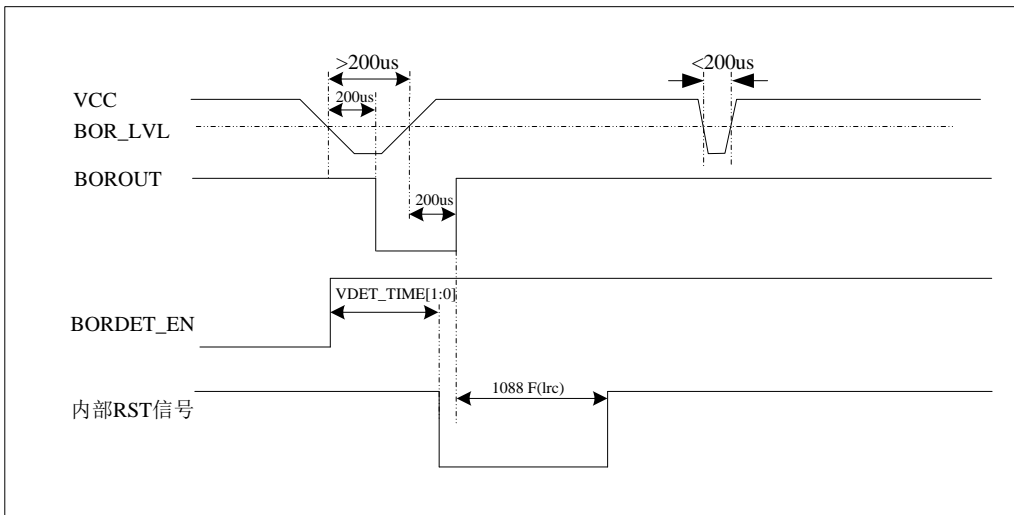
- 产生一个 BOR 脉冲
- 内部复位信号 IRST 有效
- 计数 1024 个 Flrc
- 复位状态寄存器 RSTSTA 的掉电复位标志位 BOR 被设置为 1。
- CPU 从 0000H 开始执行程序

LBOR\_DET 与 BOR\_DET 的检测过程基本相同。



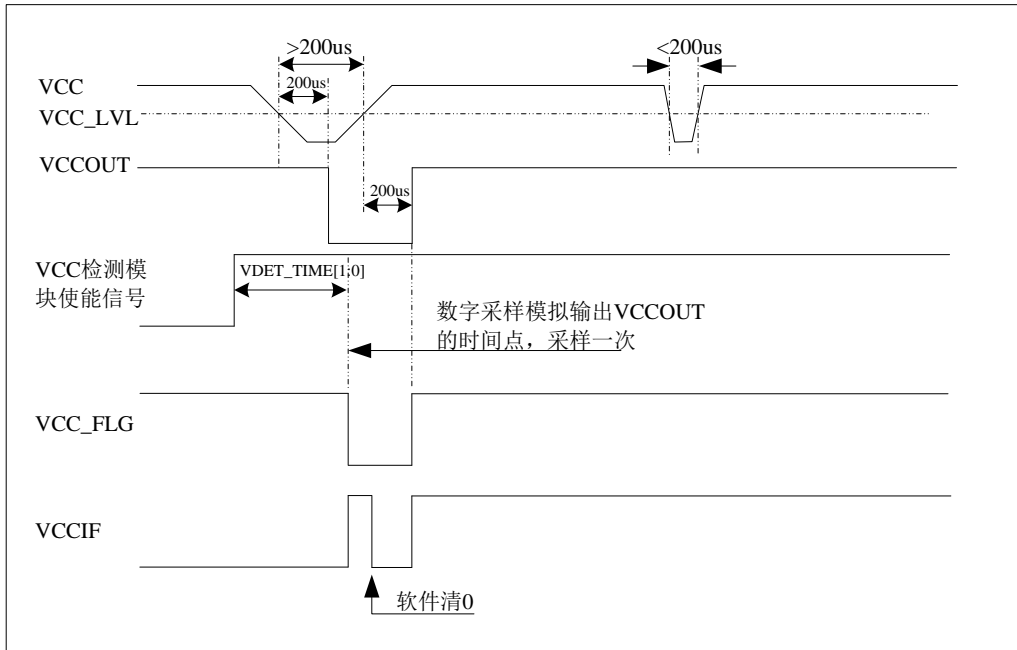


BOR 中断信号产生示意图

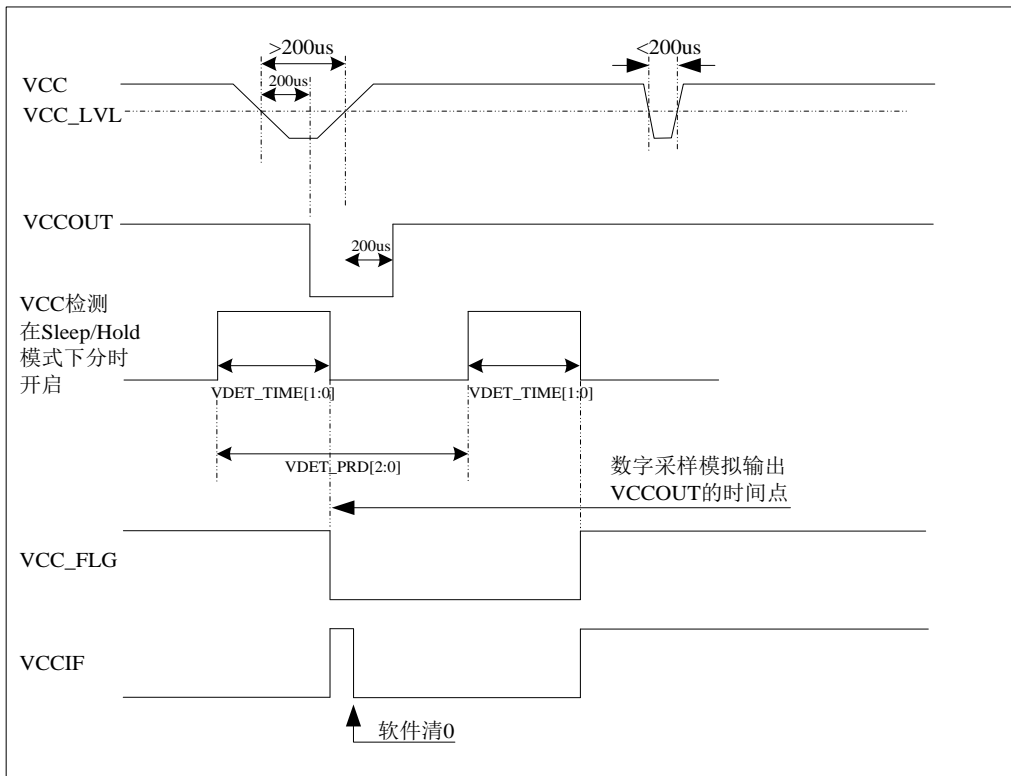


BOR 复位示意图

### 4.3.5 系统电源检测功能(VCC\_DET)

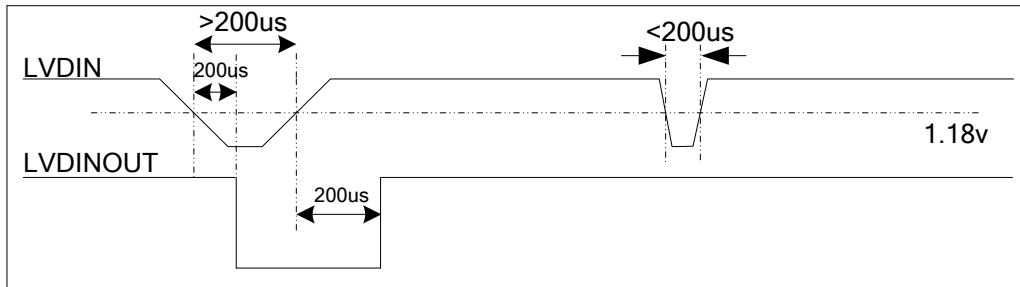


Normal模式Vcc检测



Sleep模式下Vcc分时检测

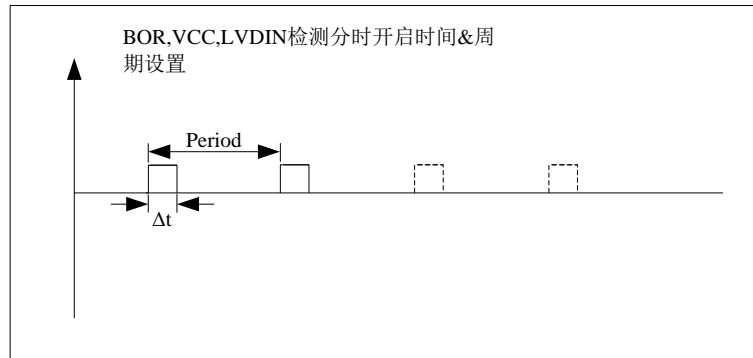
### 4.3.6 低电压检测功能(LVDIN\_DET)



LVDIN 检测信号示意图

### 4.3.7 VCC\_DET, BOR\_DET , LVDIN\_DET 分时检测时序

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



其中Period为VCC\_DET & BOR\_DET& LVDIN\_DET检测模块分时开启的周期,可通过VDETPCFG寄存器的VDET\_PRD[2:0]位设置。 $\Delta t$ 是分时检测时每个周期内VCC\_DET & BOR\_DET& LVDIN\_DET工作的时间,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:

正常上电运行时, VCC\_DET & BOR\_DET& LVDIN\_DET 模块是连续开启的,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。

当进入低功耗模式时, VCC\_DET & BOR\_DET& LVDIN\_DET 模块自动进入分时开启模式,检测系统电源 VCC 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再唤醒。

## 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
0x04	VDETCFG	R/W	0x0069	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
0x104	LDOVREFCFG	R/W	0x0080	LDOVREF配置寄存器（写保护）

## 4.5 特殊功能寄存器说明

### PMUCON（PMU 配置寄存器）

PMUCON (PMU 配置寄存器)		基地址: 0x400F400 偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	Reserved	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	Reserved	HOLD_L	LVDIN1	LVDIN0	BORRST	BORDET
Write:				DO	_EN	_EN		T_EN
Reset:	0	0	0	1	1	1	1	1

位	功能描述
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) 注: 1) 当用户需要在 Hold 模式下达到最低功耗时, 可以将该大功耗 LDO 关闭, 届时芯片自动切换使用低驱动能力低功耗的 LDO。 2) 进入 HOLD 模式时, 若已开启了 PLL 或 HRC 时钟, 则大功耗 LDO 会保持开启, 此控制位无效。
LVDIN1_EN	LVDIN1_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	LVDIN0_DET 模块使能信号, 监测 LVDIN0 输入引脚 0: 关闭 LVDIN0_DET 模块 1: 开启 LVDIN0_DET 模块 (default)
BORRST	BOR复位/中断选择位 0: VCC电压低/高于设定阈值时产生BOR中断

	1: VCC电压低于VDETCFG[1:0]设定阈值时产生BOR复位 (default)
BORDET_EN	<b>BOR_DET</b> 模块使能信号 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)		基地址: <b>0x4000F400</b> 偏移地址: <b>04H</b>						
	Bit15	14	13	12	11	10	9	Bit8
Read:	Reserved		X	X	X	X	X	Reserved
Write:	Reserved		X	X	X	X	X	Reserved
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Reserved		VCC_LVL[3:0]				BOR_LVL[1:0]	
Write:	Reserved		VCC_LVL[3:0]				BOR_LVL[1:0]	
Reset:	0	1	1	0	1	0	0	1

位	功能描述																																																																																			
VCC_LVL[3:0]	<b>VCC_DET</b> 检测阈值控制位																																																																																			
	<table border="1"> <thead> <tr> <th colspan="4">VCC_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</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>0</td><td>0</td><td>1</td><td>4.2V</td></tr> <tr><td><b>1</b></td><td><b>0</b></td><td><b>1</b></td><td><b>0</b></td><td><b>4.4V (default)</b></td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>4.6V</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>4.8V</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>5V</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>X</td><td>5V</td></tr> </tbody> </table>				VCC_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	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	0	1	4.2V	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>4.4V (default)</b>	1	0	1	1	4.6V	1	1	0	0	4.8V	1	1	0	1	5V	1	1	1	X	5V
	VCC_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																																																																															
	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	0	1	4.2V																																																																															
	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>4.4V (default)</b>																																																																															
	1	0	1	1	4.6V																																																																															
1	1	0	0	4.8V																																																																																
1	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	2.2V
	0	1	2.0V (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	RESERVED	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										
RESERVED	该寄存器位用户不要修改, 无意义										
VDET_TIME [1:0]	Hold&Sleep 模式下 VCC_DET & BOR_DET & LVDIN_DET 分时检测的时间设定 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>VDET_Time[1:0]</th> <th>检测时间</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td><b>1120us (default)</b></td> </tr> <tr> <td>0 1</td> <td><b>1120us</b></td> </tr> <tr> <td>1 0</td> <td><b>1120us</b></td> </tr> <tr> <td>1 1</td> <td><b>1120us</b></td> </tr> </tbody> </table>	VDET_Time[1:0]	检测时间	0 0	<b>1120us (default)</b>	0 1	<b>1120us</b>	1 0	<b>1120us</b>	1 1	<b>1120us</b>
VDET_Time[1:0]	检测时间										
0 0	<b>1120us (default)</b>										
0 1	<b>1120us</b>										
1 0	<b>1120us</b>										
1 1	<b>1120us</b>										

VDET_PRD[2:0]	<b>Hold&amp;Sleep 模式下 VCC_DET &amp; BOR_DET &amp; 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
	<b>0</b>	<b>1</b>	<b>0</b>	<b>67ms(default)</b>
	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)		基地址: <b>0x4000F400</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	Reserved	LVDIN1I E	LVDIN0I E	BORIE	VCCIE
<b>Write</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

注: 需要同时使能 PMUIE 使能的中断才有效。

位	功能描述
LVDIN1IE	<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
VCCIE	<b>VCC检测中断使能位</b> 0: 关闭 1: 允许

**PMUIF (PMU 中断标志寄存器)**

PMUIF (PMU 中断标志寄存器)		基地址: 0x4000F400 偏移地址: 10H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	X	LVDIN1I F	LVDIN0I F	BORIF	VCCIF
Write								
Reset:	0	0	0	0	0	0	0	0

注: 该寄存器不能被 Wake\_UP 唤醒复位。

位	功能描述
LVDIN1IF	<b>LVDIN1检测中断标志位</b> 当外部引脚LVDIN1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LVDIN0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VCC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。
VCCIF	<b>VCC检测中断标志位</b> 当系统电源VCC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。

**PMUSTA (PMU 状态寄存器寄存器)**

PMUSTA (PMU 状态寄存器寄存器)		基地址: 0x4000F400 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	X	X	X	X	LVDIN1 _FLG	LVDIN0 _FLG	BOR_FL G	VCC_FL G
Write					X	X	X	X

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

注：该寄存器为只读寄存器。

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 LVDIN 引脚电压小于 1.21V 阈值 1: 表示 LVDIN 引脚电压大于 1.21V 阈值
BOR_FLG	<b>工作电压 VCC 电压状态</b> 0: 表示 VCC 小于设定阈值 (BOR_LVL[3:0]) 1: 表示 VCC 大于设定阈值 (BOR_LVL[3:0])
VCC_FLG	<b>工作电压 VCC 电压状态</b> 0: 表示 VCC 小于设定阈值 (VCC_LVL[3:0]) 1: 表示 VCC 大于设定阈值 (VCC_LVL[3:0])

## LDOVREFCFG (LDOVREF 配置寄存器, 写保护)

<b>LDOVREFCFG (写保护)</b>		<b>基地址: 0x4000F400</b>						
<b>LDOVREF 配置寄存器</b>		<b>偏移地址: 104H</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>	<b>C_LDO_LP[1:0]</b>		X	X	X	X	X	X
<b>Write</b>								
<b>Reset:</b>	1	0	0	0	0	0	0	0

位	功能描述
C_LDO_LP[1:0]	<b>LowPowerLDO 电压配置寄存器</b> 10: 1.48V (default)

注：该寄存器只可被一级复位源 LBOR/POR 复位。

## 5 调试支持

### 5.1 概况

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；
- 支持串行线（SW）调试协议；

### 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/TX6/SWIO
输出	SW 时钟	PB.15/RX6/SWCLK

### 5.3 SW 口使用说明

工作模式	管脚功能		
	PA.6/INT1/JTAGWDTEN	PB.13/TX6/SWIO	PB.15/RX6/SWCLK
测试模式	X	X	X
调试模式	JTAGWDTEN	SWIO	SWCLK
正常模式	PA.6/INT1	PB.13/TX6	PB.15/RX6

注：X 表示不可用

仿真口在上电/服务和启动之后的初始状态：

1、PB.13/PB.15 默认 SW 功能(PB.13 为 SWIO 功能引脚，PB.15 为 SWCLK 功能引脚)，TEST 和 JTAG 不拉低，可进入调试模式。

2、TEST 引脚拉低，PB.15 强制配置为 SWCLK 输入，PB.13 强制配置为 SWIO 输出或输入（由执行指令决定，动态变化），PA.6 强制配置为 JTAGWDTEN（控制看门狗的使能）。PA.6/PB.13/PB.15 的 GPIO 寄存器可正常操作，但寄存器与之对应的功能失效，仅有调试仿真功能。

3、TEST 引脚拉高，芯片进入正常模式，PA.6/PB.13/PB.15 的功能由寄存器配置决定，即寄存器与之相对应的功能生效。



## 6 工作模式

### 6.1 工作模式

芯片共有五种模式：测试模式，调试模式，正常模式，SLEEP 模式，HOLD 模式

TEST	JTAGWDTEN	工作模式
0	1	<b>测试模式</b>
0	0	<b>调试模式</b> 此模式主要用来仿真下载程序
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，CPU 正常工作。
		<b>SLEEP 模式</b> 在正常模式下，CPU通过执行如下指令进入SLEEP模式： SCB->SCR = 0x0004; __WFI();
		<b>HOLD 模式</b> 在正常模式下，CPU通过执行如下指令进入HOLD模式： SCB->SCR = 0x0000; __WFI();

芯片在正常模式下可以通过软件配置进入两种低功耗模式(SLEEP 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 模式唤醒后产生唤醒复位，而 HOLD 模式唤醒后则是接着原来运行的代码继续运行
- **SLEEP/HOLD 模式下，HRC 或 PLL 时钟使能，大功耗 LDO 保持常开且 LP\_SWOFFN 保持闭合。**

### 6.2 睡眠模式 (SLEEP)

#### 6.2.1 SLEEP 模式概述

- LDO\_1P5 (大功耗) 自动关闭，其供电的模块相应关闭 (**HRC和PLL时钟未使能**)，由LDO\_LowPower (低功耗) 为部分数字供电。
- VRTC电压域下的模块供电不关闭。
- RTC相关的晶振电路，分频补偿电路一直开启。
- 进入SLEEP后，则BOR\_DET、LVDINx(x=0/1)和VCC\_DET模块由硬件分时逻辑控制，以降低功耗。
- 在SLEEP模式下，看门狗开关可配。如果使能看门狗，看门狗计数溢出时，系统会发生WDT复位。SLEEP模式下看门狗配置详见Flash控制功能。
- TBS工作模式 (分时开启、实时触发、注入模式) 与芯片工作模式无关，建议使用分时开启模式。

- 为降低SLEEP模式下的功耗，可以在进入SLEEP模式之前，根据外设的状态配置GPIO（详见GPIO章节），防止通过GPIO漏电。
- 如果用户期望在SLEEP达到最低功耗：通过CLKCTRL0和CLKCTRL1寄存器，关闭非必需模块的时钟。
- LRC 默认开启状态，可通过软件配置关闭。

## 6.2.2 SLEEP 模式下的唤醒源

CPU从SLEEP状态下唤醒后产生唤醒复位，内核被复位，因此不进入中断向量，不会执行中断服务程序，程序从地址0x00000000开始执行。

在SLEEP模式下，所有复位信号是不可被屏蔽的（软复位除外），即系统进入SLEEP模式后，如果有效复位信号产生，芯片发生复位动作，程序从地址0x00000000开始执行。

要实现在 SLEEP 模式下的唤醒功能，必需先使能对应的 NVIC 中断。SLEEP 模式具有以下唤醒源：

- 1) **RTC 中断(含时间戳)唤醒：**配置 RTC 模块中断使能 (NVIC\_EnableIRQ(RTC\_IRQn))，并且使能相应 RTC 的子中断源，当使能的 RTC 中断产生时，则会产生 RTC 中断唤醒。
- 2) **TBS 中断唤醒：**配置 TBS 模块中断使能 (NVIC\_EnableIRQ(TBS\_IRQn))，并且使能相应 TBS 的子中断源，当使能的 TBS 中断产生时，则会产生 TBS 中断唤醒。
- 3) **PMU 中断唤醒：**配置 PMU 模块中断使能 (NVIC\_EnableIRQ(PMU\_IRQn))，并且使能相应 PMU 的子中断源，当使能的 PMU 中断产生时，则会产生 PMU 中断唤醒。
- 4) **TMR4/5 中断唤醒：**配置 TMR4/5 模块中断使能 (NVIC\_EnableIRQ(TMRx\_IRQn))，并且使能相应 TMR4/5 的子中断源，当使能的 TMR4/5 中断产生时，则会产生 TMR4/5 中断唤醒。
- 5) **外部 INT 中断：**相应的 PIN 要配置为 INT 复用功能（详见 GPIO 单元），并使能相应外部中断模块 (NVIC\_EnableIRQ(EXTIx\_IRQn)) 和相应触发沿中断。当 INT 引脚出现有效触发沿（软件配置决定），并保持沿后电平大于 2 个 Fcpu，则会产生外部 INT 中断唤醒。
- 6) **RX 中断唤醒：**相应的 PIN 要配置为 RX 复用功能（详见 GPIO 单元），并使能相应 UART 模块中断 (NVIC\_EnableIRQ(UARTx\_IRQn))。当 RX 引脚出现下降沿，并保持低电平大于 2 个 Fcpu，则会产生 RX 中断唤醒。UART 唤醒不用使能 UART 模块时钟。

注：Fcpu 时钟为 CPU 时钟，如果用户在进入 SLEEP 之前 CPU 时钟分频寄存器 SYSCLKDIV 的分频值设置过大（如 1/128），则单个 Fcpu 时钟较长，导致芯片从 SLEEP 模式下唤醒时间较长。

## 6.2.3 查询 SLEEP 模式唤醒源

从SLEEP模式唤醒后，可以查询复位标志寄存器（RSTSTA寄存器的WakeupRST位），如果该位为1，则说明发生了唤醒复位。通过唤醒标志寄存器WAKEIF和配置的模块中断使能情况，以确定具体的唤醒源，其中可供查询的唤醒标志为PMUWKIF、INTxWKIF、RXxWKIF、TBSWKIF、RTCWKIF、TIMER4WKIF、TIMER5WKIF。例如：WAKEIF的RTCWKIF位为1，表示CPU是由RTC中断信号引起的唤醒，具体的中断源详见RTC模块说明。

## 6.2.4 进入 SLEEP 模式的方式

SLEEP模式通过内核的系统自带指令WFI进入。进入SLEEP指令如下：

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

在仿真调试状态下，执行上述指令，CPU停止，但未进SLEEP模式。

## 6.3 待机模式（HOLD）

### 6.3.1 HOLD 模式概述

在 HOLD 模式下，CPU 时钟停止，所有的外设能够继续运行并可在外部中断或事件发生时唤醒 CPU。**HRC 和 PLL 均未使能条件下**，为进一步降低功耗，可通过 HOLD\_LDO 控制位关闭 LDO\_1P5（大功耗），由 LDO\_LowPower（低功耗）为数字供电。

由于 LDO\_LowPower（低功耗）低输出驱动能力，无法满足所有功能模块运行的功耗需求。用户可根据实际应用，配置开启 LDO\_1P5（大功耗），以提高输出驱动能力。

为了降低 HOLD 模式下的功耗，BOR\_DET、LVDINx 和 VCC\_DET 模块由芯片硬件分时逻辑控制。根据外设的状态配置 GPIO（详见 GPIO 章节），防止通过 GPIO 漏电。

如果系统在进入 HOLD 模式之前配置了模块中断使能，在进入 HOLD 模式后产生相应的中断，则会导致芯片从 HOLD 模式下唤醒，并响应相应的中断。

RTC 相关的晶振电路，分频补偿电路一直开启。TBS 工作模式（分时开启、实时触发、注入模式）与芯片工作模式无关，建议使用分时开启模式。

LRC默认开启状态，可通过软件配置关闭。

看门狗可配开关，HOLD模式下看门狗配置详见Flash控制功能。

### 6.3.2 HOLD 模式下的唤醒源

CPU从HOLD状态下唤醒后，内核接着原来运行的代码继续运行并响应相应的中断。

在HOLD模式下，所有复位信号是不可被屏蔽的（唤醒复位除外），即系统进入HOLD模式后，如果有效复位信号产生，芯片发生复位动作，程序从地址0x00000000开始执行。

在总中断打开的前提下，所有外设模块中断可唤醒CPU（根据实际配置模块中断使能和中断标志建立的情况决定）。

### 6.3.3 进入 HOLD 模式的方式

HOLD模式通过内核的系统自带指令WFI进入。

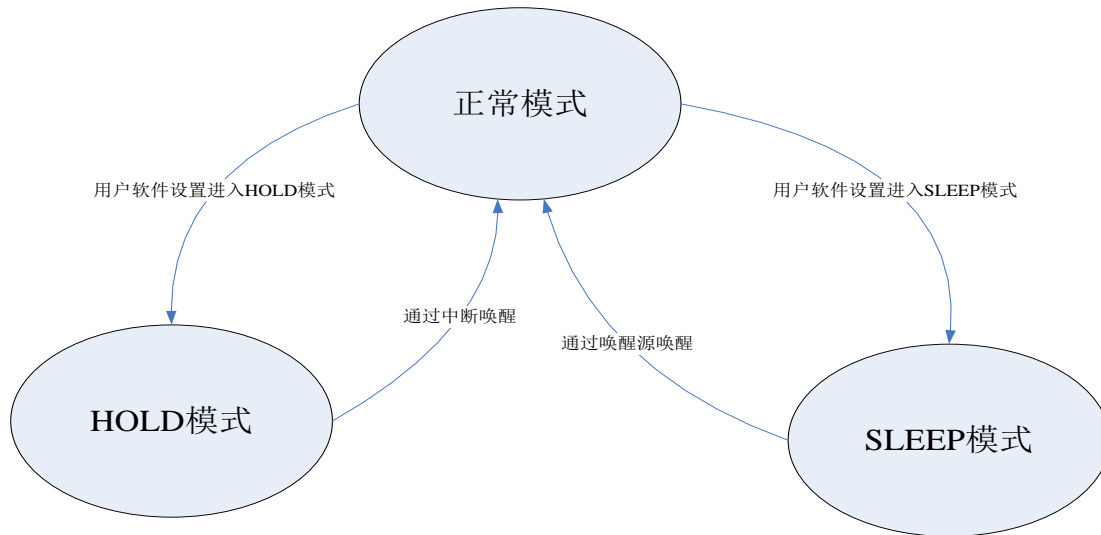
进入HOLD指令如下：

```
SCR = 0x0000;
```

\_\_WFI());

在仿真调试状态下，执行上述指令，CPU停止，芯片进入HOLD模式。

## 6.4 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

基地址：0x4000F400				
偏移地址	名称	读写方式	复位值	功能描述
0x18	WAKEIF	R/W	0x0000	唤醒源标志寄存器

## 6.6 特殊功能寄存器说明

### SCR（系统控制寄存器）

SCR (系统控制寄存器)		基地址： 0xE000ED00 偏移地址： 10H						
	Bit31	30	29	28	27	26	25	Bit24
Read:	SCR[31:24]							
Write:	SCR[31:24]							

Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	SCR[23:16]							
Write:	SCR[23:16]							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	SCR[15:8]							
Write:	SCR[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	Reserved	X	SLEEPD EEP	Reserved	X
Write:	X	X	X	Reserved	X	SLEEPD EEP	Reserved	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SLEEPDEEP	0: 在执行 WFI 指令后, 芯片进入 HOLD 模式 1: 在执行 WFI 指令后, 芯片进入 SLEEP 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:	DMAWK	SPI2WK	SPI3WK	INT789	RX6WKI	TIMER5	TIMER4	SELFTE
Write:	KIF	IF	IF	WKIF	F	WKIF	WKIF	STWKIF
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	SPI1WK	SPI0WK	I2CWKI	RTCWKI	TBSWKI	TIM3W	TIM2W	TIM1W
Write:	IF	IF	F	F	F	KIF	KIF	KIF
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	TIM0W	RX5WKI	RX4WKI	RX3WKI	RX2WKI	RX1WKI	RX0WKI	EWUW
Write:	KIF	F	F	F	F	F	F	KIF
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	EMUWK	INT456	Reserved	FFTFRA	INT123	INT0WK	AESWKI	PMUWK
Write:	IF	WKIF	Reserved	MWKIF	WKIF	IF	F	IF
Reset:	0	0	0	0	0	0	0	0

注: Sleep 唤醒和 Hold 唤醒共用此标志位

该寄存器为只读寄存器, 它永远会保持上一次导致芯片唤醒的唤醒源头, 当一个新的唤醒事件产生时候, 由

硬件产生新的唤醒源标志，同时将之前的唤醒标志清 0

位	功能描述
DMAWKIF	<b>DMA唤醒标志</b> DMA唤醒发生时，此位置为1
SPI2WKIF	<b>SPI2唤醒标志</b> SPI2唤醒发生时，此位置为1
<b>SPI3WKIF</b>	<b>SPI3唤醒标志</b> SPI3唤醒发生时，此位置为1
<b>INT789WKIF</b>	<b>INT789唤醒标志</b> INT789唤醒发生时，此位置为1
<b>RX6WKIF</b>	<b>RX6唤醒标志</b> RX6唤醒发生时，此位置为1
TIMER5WKIF	<b>TIMER5唤醒标志</b> SLEEP/HOLD模式下TIMER5中断发生时将会产生TIMER5唤醒，此位置为1。
TIMER4WKIF	<b>TIMER4唤醒标志</b> SLEEP/HOLD 模式下 TIMER4 中断发生时将会产生 TIMER4 唤醒，此位置为 1。
SELFTTESTWKIF	<b>SELFTTEST唤醒标志</b> SELFTTEST唤醒发生时，此位置为1
SPI1WKIF	<b>SPI1唤醒标志</b> SPI1唤醒发生时，此位置为1
SPI0WKIF	<b>SPI0唤醒标志</b> SPI0唤醒发生时，此位置为1
I2CWKIF	<b>I2C唤醒标志</b> 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RTC那个唤醒源头需要查看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TBS那个唤醒源头需要查看TBSIF寄存器）
TIM3WKIF	<b>TIM3唤醒标志</b> TIM3 唤醒发生时，此位置为 1
TIM2WKIF	<b>TIM2唤醒标志</b> TIM2 唤醒发生时，此位置为 1
TIM1WKIF	<b>TIM1唤醒标志</b> TIM1 唤醒发生时，此位置为 1
TIM0WKIF	<b>TIM0唤醒标志</b> TIM0 唤醒发生时，此位置为 1
RX5WKIF	<b>RX5唤醒标志</b> RX5唤醒发生时，此位置为1
RX4WKIF	<b>RX4唤醒标志</b> RX4唤醒发生时，此位置为1

RX3WKIF	<b>RX3唤醒标志</b> RX3唤醒发生时，此位置为1
RX2WKIF	<b>RX2唤醒标志</b> RX2唤醒发生时，此位置为1
RX1WKIF	<b>RX1唤醒标志</b> RX1唤醒发生时，此位置为1
RX0WKIF	<b>RX0唤醒标志</b> RX0唤醒发生时，此位置为1
EWUWKIF	<b>EWU唤醒标志</b> EWU唤醒发生时，此位置为1
EMUWKIF	<b>EMU唤醒标志</b> EMU唤醒发生时，此位置为1
INT456WKIF	<b>INT456唤醒标志</b> INT456唤醒发生时，此位置为1
FFTFRAMWKIF	<b>FFTFRAM唤醒标志</b> FFT模块和FRAMPACK模块唤醒发生时，此位置为1
INT123WKIF	<b>INT123唤醒标志</b> INT123唤醒发生时，此位置为1
INT0WKIF	<b>INT0唤醒标志</b> INT0唤醒发生时，此位置为1
AESWKIF	<b>AES唤醒标志</b> AES 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（具体哪个PMU唤醒源需要查看PMUIF寄存器。）

## 7 GPIO 模块

### 7.1 概述

HT762X提供PA[0..3,5..13], PB[0..3,13,15], PC[0..14], PD[8..12], PE[0..8], PG[5..9], PJ[0..1]并行端口, 支持55个双向I/O引脚。

除PB.13和PB.15以外的双向IO默认态为高阻态, PB13、PB15默认态为SWD功能。

特定GPIO支持, 上电初始化状态进行趋向性设计, 即电源电压未达到逻辑生效电压(LDO未建立)仍能保持高阻状态。

关于IO的详细状态详见章节“IO状态说明”。

每个双向IO口输出都可配为推挽输出或者开漏输出, 输入都可配为上拉输入或浮空输入, 输入上拉电阻为约88k。都可配置为高阻模式。详细规则参见章节“IO的功能配置说明”。

具有复用模拟功能: LVDIN<sub>x</sub>(x=0/1)、ADCIN<sub>x</sub>(x=0/1/3/4/5),除此之外的其它复用功能为数字功能。

各个IO均配置标准的防止ESD的二极管防护电路, 保证IO的可靠性。

各个IO都具有5mA的输出驱动能力, 部分具备大电流驱动的IO参见“支持大电流驱动的IO”。

部分管脚支持TTL电平, 详见引脚定义章节的说明;

部分管脚支持5V Tolerance功能, 详见引脚定义章节的说明。

当双向IO功能为数字输出时, 开漏功能(Open Drain)配置有效, 上拉功能配置无效, 滤波功能配置无效。

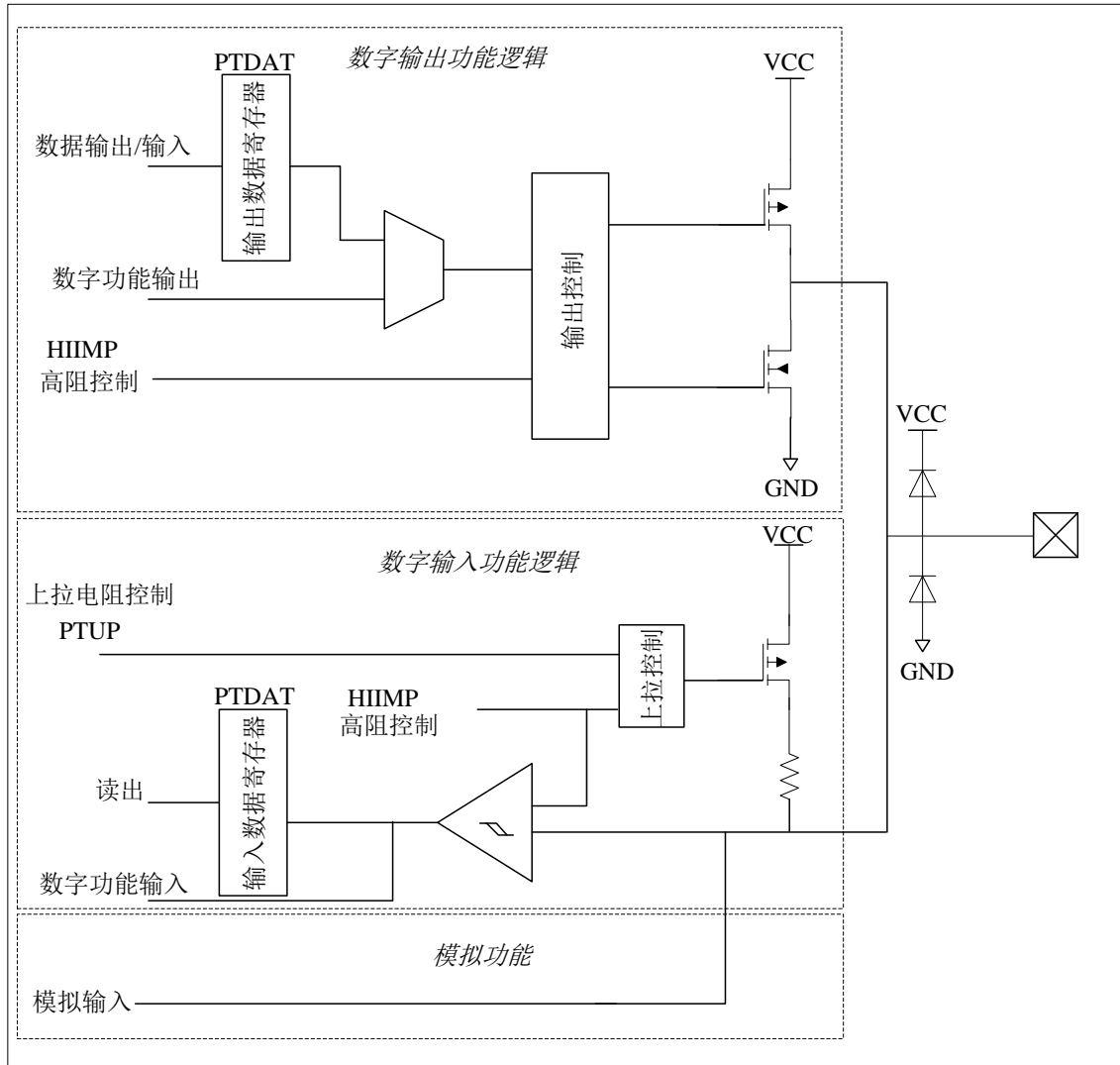
当双向IO功能为数字输入时, 上拉功能配置有效, 滤波功能配置有效。开漏功能(Open Drain)配置无效。

若配置为模拟输入, 开漏控制和上拉控制都无效;

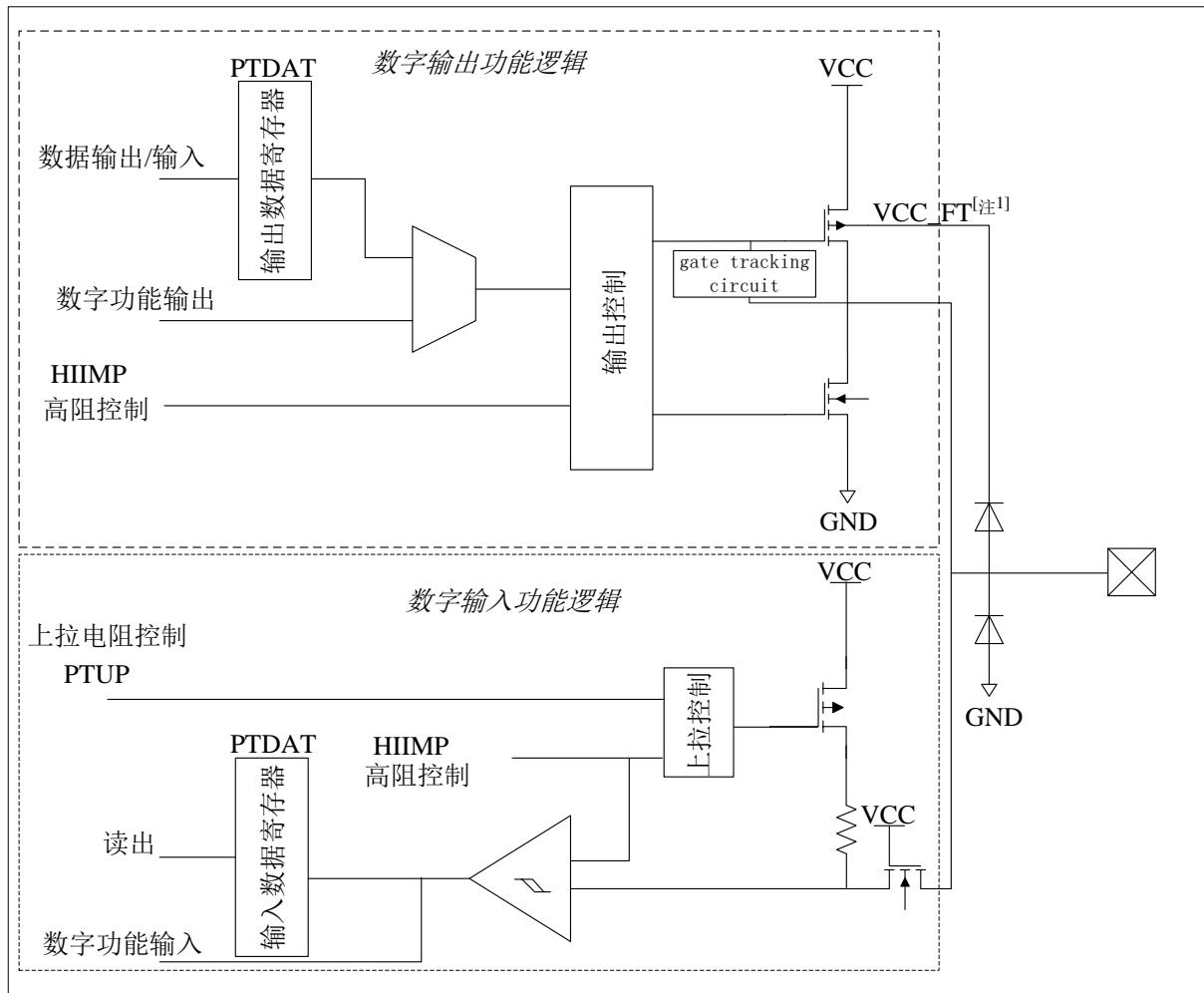
若配置为高阻功能, 双向IO输入信号固定为零, 数字输入/输出无效, 只有模拟功能有效, 详见章节“关于高阻状态的说明”。

## 7.2 芯片引脚结构说明

### 7.2.1 芯片引脚基本结构图



## 7.2.2 5V Tolerance IO 引脚结构图



注:

- 1) VCC\_FT 仅针对 5V Tolerance 引脚，与 VCC 不同。
- 2) 5V Tolerance 引脚为纯数字引脚，不具有模拟复用功能。

## 7.3 关于高阻状态的说明

HT762X 的 GPIO 具有高阻状态，关于高阻状态时，GPIO 具有以下特征：

1. 双向 IO 为高阻功能时，数字输入功能无效，输入信号固定零，上拉和输入滤波控制无效，上拉电阻关闭；数字输出无效，开漏功能(Open Drain)配置无效，只模拟功能有效。其 IO 对应的寄存器 (IOCFG、AFCFG、PTDIR、PTUP、PTDAT、PTSET、PTCLR、PTTOG、PTOD、FILT、HDPORT) 的控制位可配置，但功能无效。
2. 除 PB.13 和 PB.15 以外的双向 IO 默认态和复位态<sup>[注]</sup>为高阻态。
3. 当端口从 GPIO 输入配置为高阻功能时，其 IO 对应的寄存器中，PTDAT 的控制位将固定为 0，其他寄存器 (IOCFG、AFCFG、PTDIR、PTUP、PTSET、PTCLR、PTTOG、PTOD、FILT、HDPORT)

的控制位将保持配置为高阻前的状态，但不会生效；

当端口从 GPIO 输出配置为高阻功能时，其 IO 对应的寄存器（IOCFG、AFCFG、PTDIR、PTUP、PTDAT、PTSET、PTCLR、PTTOG、PTOD、FILT、HDPORT）的控制位将保持配置为高阻前的状态，但不会生效；

当端口从复用功能配置为高阻功能时，其 IO 对应的寄存器中，PTDIR 和 PTDAT 无意义，其他寄存器（IOCFG、AFCFG、PTUP、PTSET、PTCLR、PTTOG、PTOD、FILT、HDPORT）的控制位将保持配置为高阻前的状态，但不会生效；

4. 具有模拟复用功能的 IO，信号由引脚直接连接至模拟模块内（呈现高阻态），模拟功能始终有效。即使没有将相应 IO 复用为相应的模拟功能，IO 仍然可以作为模拟引脚使用。具有模拟功能的 IO 配置为高阻态时，建议把对应的模拟模块使能关闭。

注：关于复位态的说明：

- 1) 双向 IO 的复位态指发生了 BOR 复位、外部引脚/RST 复位、自定义软复位；
- 2) VRTC 电压域下的双向 IO 仅受 VRTC 的 POR 复位影响，不会被 VCC 电压域下的复位影响；
- 3) POR 和 LBOR 复位期间，双向 IO 由模拟逻辑控制，不属于本条说明的描述范围。

具有模拟复用功能的 IO 如下：

100 PIN	80 PIN	64 PIN	标识	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
37	34	24	PA.12	I/O	INT4	ADCIN0	外部中断 4/ADC 输入通道 0
38	35	25	PA.13	I/O	INT5	ADCIN1	外部中断 5/ADC 输入通道 1
39	36	26	PG.5	I/O	LVDIN1	TMR0	外部低电压检测 1/PWM 主输出_捕获输入_事件输入
81	-	-	PE.5	I/O	TX3	ADCIN3	UART3 发送/ADC 输入通道 3
82	-	-	PE.6	I/O	TMR1	ADCIN4	PWM 主输出_捕获输入_事件输入/ADC 输入通道 4 TTL 电平输入
83	64	54	PE.7	I/O	LVDIN0	ADCIN5	外部低电压检测 0/ADC 输入通道 5

## 7.4 IO 的功能配置说明

寄存器控制说明：

IO 功能		HIIPM	IOCFG	AFCFG	PTDIR	PTUP	PTOD	HDPOR T <sup>[注]</sup>	FILT
高阻 模式	高阻模式	1	X	X	X	X	X	X	X
GPIO 模式	IO 输入	0	0	X	0	0/1	X	X	0/1
	IO 输出	0	0	X	1	X	0/1	0/1	X

复用功能模式	数字输入	0	1	0/1	X	0/1	X	X	0/1
	数字输出	0	1	0/1	X	X	0/1	0/1	X
	模拟功能	0	1	0/1	X	X	X	X	X

说明:

0: 需配置为 0;

1: 需配置为 1;

0/1: 可配置为 0 或 1, 且配置生效;

X: 可配置为 0 或 1, 但配置无效。

注: HDPORIT 寄存器仅针对支持大电流驱动的 IO 有配置位, 有效范围参见章节“支持大电流驱动的 IO”。

## 7.5 复用功能说明

部分复用功能支持映射到两个管脚, 若仅有一个管脚被配置为该复用功能, 则该管脚的复用功能直接生效, 若两个管脚均被配置为该复用功能, 会遵循以下规则:

- 1) 若目标复用功能是输出属性, 则复用功能内部连接, 两个管脚上的复用功能均生效;
- 2) 若目标复用功能是输入属性, 则采取优先级的方式判断, 优先级高的管脚复用功能生效, 优先级低的管脚仅保持该复用功能的属性, 但复用功能不生效;
- 3) 若目标复用功能同时具有输入和输出属性, 则其作为输出功能使用时, 按照规则 1) 进行判定, 作为输入功能使用时, 按照规则 2) 进行判定;
- 4) 若将复用功能已经生效的优先级高的管脚修改为其他功能, 则优先级低的管脚复用功能将被激活, 但复用功能内部模块接收到的输入信号是连续的, 并不会察觉输入信号的管脚已经进行了切换;
- 5) 同时具有输入和输出属性的复用功能, 其低优先级的管脚输入输出属性跟随实际复用功能进行切换;
- 6) 具有多管脚映射的复用功能中, 仅具有输出属性的复用功能有: TX1~3;
- 7) 优先级如下表所示:

No.	复用功能	优先级高	优先级低
1	INT2	PA.7	PC.8
2	INT3	PA.8	PC.13
3	INT4	PA.9	PA.12
4	INT5	PA.10	PA.13
5	INT8	PE.8	PA.5
6	RX1	PC.1	PG.9
7	RX2	PC.12	PD.11
8	RX3	PE.4	PA.2
9	TMR0	PE.0	PG.5
10	TMR1	PE.6	PA.0
11	TMR3	PC.10	PC.14
12	TMR4	PE.8	PA.1
13	TX1	PC.0	PG.8
14	TX2	PC.11	PD.10
15	TX3	PE.5	PA.3

## 7.6 I/O 端口基地址列表

<b>GPIO 模块寄存器基地址:</b>				
0x40011000(PA 端口);				
0x40011100(PB 端口);				
0x40011200(PC 端口);				
0x40011300(PD 端口);				
0x40011400(PE 端口);				
0x40011600(PG 端口);				
0x40011900(PJ 端口);				
偏移地址	名称	读写方式	复位值	功能描述
0x00	IOCFG	R/W	0x0000	复用功能生效配置寄存器
0x04	AFCFG	R/W	0x0000	复用功能选择寄存器
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	端口置位寄存器 (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	端口开漏功能配置寄存器
0x24	FILT	R/W	0xFFFF	输入模拟滤波配置寄存器 (部分)
0x28	HIIPM	R/W	0xFFFF	高阻配置寄存器

注：寄存器复位值与最大封装芯片该组管脚有效数有关，如 PA14、PA15，对于这类端口相应的数据位如 IOCFG[15:14]，是不可读写的。

<b>HDPORT 模块寄存器基地址: 0x40011500(大电流端口);</b>				
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

## 7.7 特殊功能寄存器说明

### IOCFG (复用功能生效配置寄存器)

<b>IOCFG</b> (复用功能生效配置寄存器)			<b>基地址: 0x40011000--0x40011400, 0x40011600, 0x40011900</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>	IOCFG[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>	IOCFG[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
IOCFG[15:0]	端口 IO 功能配置位 0: 对应的端口配置为 GPIO (default) 1: 对应的端口配置为复用功能 PIN

注:

1. 有些端口没有使用如 PA14, PA15, 对于这类端口相应的数据位如 IOCFG[15:14], 是不可读写的;
2. PB.13、PB.15 的对应 bit 位默认为 1, 选择第二复用功能 SWD。

## AFCFG (复用功能选择寄存器)

<b>AFCFG</b> (复用功能选择寄存器)		基地址: 0x40011000--0x40011400, 0x40011600, 0x40011900 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>	AFCFG[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>	AFCFG[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
AFCFG[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 (default) 1: 复用功能 2

注: PB.13、PB.15 的对应 bit 位默认为 AFCFG=1, 选择第二复用功能 SWD;

## 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, 0x40011600, 0x40011900 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>	PTDIR[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>	PTDIR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
PTDIR[15:0]	端口方向配置位（此寄存器只在对应端口配置为 <b>GPIO</b> 功能时才有效） 0: 输入（default） 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)	基地址: 0x40011000--0x40011400, 0x40011600, 0x40011900 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>	PTUP[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>	PTUP[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
PTUP[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（default） 1: 禁止上拉（浮空）

注：数字输入含配置为输入模式的 GPIO 和具有数字输入属性的复用功能。

### PTDAT（端口数据寄存器）

<b>PTDAT</b> (端口数据寄存器)	基地址: 0x40011000--0x40011400, 0x40011600, 0x40011900 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>	PTDAT[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>	PTDAT[7:0]							
<b>Write:</b>	PTDAT[7:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
PTDAT[15:0]	端口数据位（此寄存器只在对应端口配置为 GPIO 功能，或含有数字输入属性的复用功能时才有效） 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 说明及数据读取规则：

- 1) 引脚选择 GPIO 功能：
  - 若方向寄存器配置为输出，PTDAT 读取值为寄存器设置值，不随引脚电平变化而变化；
  - 若方向寄存器配置为输入，PTDAT 读取值为引脚状态值，反映引脚电平状态。
- 2) 引脚选择复用数字功能：
  - 若复用为数字输出功能，且 PTDIR 为 0，则 PTDAT 读取值恒为 0，不随引脚电平变化而变化；
  - 若复用为数字输出功能，且 PTDIR 为 1，则 PTDAT 读取值恒为寄存器设置值，不影响引脚实际输出；
  - 若复用为数字输入功能，且 PTDIR 为 0，则 PTDAT 读取值为引脚状态值，反映引脚电平状态；
  - 若复用为数字输入功能，且 PTDIR 为 1，则 PTDAT 读取值为寄存器设置值，不随引脚电平变化而变化。
- 3) 引脚选择模拟功能，则该引脚对应的 PTDAT 读取值固定为 0。

## PTSET（端口置位寄存器）

<b>PTSET</b> (端口置位寄存器)	基地址：0x40011000--0x40011400, 0x40011600, 0x40011900 偏移地址：14H							
	<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>	PTSET[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>	PTSET[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
PTSET[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 PTDAT 中对应的值）
--	---

## PTCLR（端口复位寄存器）

<b>PTCLR</b> (端口复位寄存器)			基地址: 0x40011000--0x40011400, 0x40011600, 0x40011900 偏移地址: 18H					
	<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>	PTCLR[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>	PTCLR[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
PTCLR[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 PTDAT 中对应的值）

## PTTOG（端口翻转寄存器）

<b>PTTOG</b> (端口翻转寄存器)			基地址: 0x40011000--0x40011400, 0x40011600, 0x40011900 偏移地址: 1CH					
	<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>	PTTOG[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>	PTTOG[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
PTTOG[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 PTDAT 中对应的值）

## PTOD（端口开漏功能配置寄存器）

<b>PTOD</b> (端口开漏功能配置寄存器)			基地址: 0x40011000--0x40011400, 0x40011600, 0x40011900 偏移地址: 20H					
	<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>	PTOD[15:8]							
<b>Write:</b>								
<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>	PTOD[7:0]							
<b>Write:</b>								
<b>Reset:</b>	1	1	1	1	1	1	1	1

位	功能描述
PTOD[15:0]	端口开漏功能配置位（此寄存器只在对应端口配置为数字输出时才有效） 0: 开漏功能使能（开漏输出，输出高为浮空，输出低为低） 1: 开漏功能无效（推挽输出，输出高为高，输出低为低）（default）

注：数字输出含配置为输出模式的 GPIO 和具有数字输出属性的复用功能。

**PJ.0,PJ.1** PTOD 的逻辑是反的。0-开漏功能无效，1-开漏功能使能。

## FILT（输入模拟滤波配置寄存器）

<b>FILT</b> (输入模拟滤波配置寄存器)			基地址: 0x40011000--0x40011400, 0x40011600, 0x40011900 偏移地址: 24H					
	<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>	FILT[15:8]							
<b>Write:</b>								
<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>	FILT[7:0]							
<b>Write:</b>								
<b>Reset:</b>	1	1	1	1	1	1	1	1

位	功能描述
FILT[15:0]	端口模拟 2us 滤波配置位 0: 模拟滤波无效 1: 模拟滤波有效（default）  注意仅部分 PAD 有此滤波控制功能，控制位对应 GPIO 相应位。 不同 GPIO 组对应的复位值不同，仅具有滤波功能的端口复位值为 1，其余为 0； 具有此功能的 IO 列表如下：

	PA.5、PA.6、PA.7、PA.8、PA.9、PA.10、PA.11 PC.4、PC.12 PE.2、PE.4 <b>PJ.0、PJ.1</b>  模拟滤波仅针对数字输入功能的下降沿有效。
--	---

## HIIPM（高阻配置寄存器）

HIIPM (高阻配置寄存器)		基地址: 0x40011000--0x40011400, 0x40011600, 0x40011900 偏移地址: 28H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	HIIPM[15:8]							
Write:								
Reset:	1	1	1	1	1	1	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	HIIPM[7:0]							
Write:								
Reset:	1	1	1	1	1	1	1	1

位	功能描述
HIIPM[15:0]	端口高阻配置位 0: 高阻关闭 1: 高阻开启 (default)

注: PB.13、PB.15 的对应 bit 位默认为 HIIPM=0, 高阻关闭, 选择第二复用功能 SWD;

## HDPORT（大电流端口配置寄存器）

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

注: 该寄存器为单独的起始地址

位	功能描述
PA8HD	<b>PA.8 端口大电流驱动配置</b> 0: 普通驱动能力 (default) 1: 超大电流驱动能力
PA7HD	<b>PA.7 端口大电流驱动配置</b> 0: 普通驱动能力 (default) 1: 超大电流驱动能力
PA6HD	<b>PA.6 端口大电流驱动配置</b> 0: 普通驱动能力 (default) 1: 超大电流驱动能力
PC0HD	<b>PC.0 端口大电流驱动配置</b> 0: 普通驱动能力 (default) 1: 超大电流驱动能力

## 8 中断模块

### 8.1 中断向量说明

系统中断	中断号	中断使能	中断标志	功能描述
NMI	-14			硬件强制切换低频 RC 中断
HardFault	-13			故障/异常触发中断
SVCall	-5			软件触发中断
PendSV	-2			软件触发中断
SysTick	-1			系统定时器周期中断
以上为内核自带				
PMU	0	PMUIE.LVDIN0IE	PMUIF.LVDIN0IF	LVDIN0检测中断
		PMUIE.LVDIN1IE	PMUIF.LVDIN1IF	LVDIN1检测中断
		PMUIE.BORIE	PMUIF.BORIF	BOR检测中断
		PMUIE.VCCIE	PMUIF.VCCIF	VCC检测中断
AES	1	AESGHASHIE. RANDIE	AESGHASHIF. RANDIF	随机数中断标志
		AESGHASHIE. GHASHIE	AESGHASHIF. GHASHIF	GHASH计算中断标志
		AESGHASHIE. AESIE	AESGHASHIF. AESIF	AES计算中断标志
EXTI0	2	EXTIE.RIE[0]	EXTIF.RIF[0]	外部输入引脚上升沿中断
		EXTIE.FIE[0]	EXTIF.FIF[0]	外部输入引脚下降沿中断
EXTI1-3	3	EXTIE.RIE[3:1]	EXTIF.RIF[3:1]	外部输入引脚上升沿中断
		EXTIE.FIE[3:1]	EXTIF.FIF[3:1]	外部输入引脚下降沿中断
FFT	4	FFTIE.FFT_IE	FFTIF.FFT_IF	FFT 计算完成中断
		FFTIE.FFT_DONE_IE	FFTIF.FFT_DONE_IF	FFT 配置通道计算完成中断
		EIIE.EI_FFTIE	EIIF.EI_FFTIF	FFT 数据接口更新中断
		EIIE.EI_NISIE	EIIF.EI_NISIF	非侵入数据接口更新中断
		FRAMIE.FRAME_DONE IE	FRAMIF.FRAME_DON EIF	帧自动发送完成中断
		FRAMIE.FIFOEMPTYI E	FRAMIF.FIFOEMPTY IF	EMU 数据缓存 FIFO 空中断
		FRAMIE.FIFOFULLIE	FRAMIF.FIFOFULLIF	EMU 数据缓存 FIFO 满中断
		FRAMIE.HEADIE	FRAMIF.HEADIF	帧头传输完成中断
		FRAMIE.ENDIE	FRAMIF.ENDIF	帧尾传输完成中断
Reserved1	5			
EXTI4-6	6	EXTIE.RIE[6:4]	EXTIF.RIF[6:4]	外部输入引脚上升沿中断
		EXTIE.FIE[6:4]	EXTIF.FIF[6:4]	外部输入引脚下降沿中断
EMU	7	EMUIE1.AIIP_PosCFIE	EMUIF1.AIIP_PosCFIF	全波有功通路 CF 正向脉冲中断

EMUIE1.AllQ_PosCFIE	EMUIF1.AllQ_PosCFIF	全波无功通路 CF 正向脉冲中断
EMUIE1.S_PosCFIE	EMUIF1.S_PosCFIF	视在 通路 CF 正向脉冲中断
EMUIE1.FundP_PosCFIE	EMUIF1.FundP_PosCFIF	基波有功通路 CF 正向脉冲中断
EMUIE1.FundQ_PosCFIE	EMUIF1.FundQ_PosCFIF	基波无功通路 CF 正向脉冲中断
EMUIE1.HarP_PosCFIE	EMUIF1.HarP_PosCFIF	谐波有功通路 CF 正向脉冲中断
EMUIE1.AllP_NegCFIE	EMUIF1.AllP_NegCFIF	全波有功通路 CF 反向脉冲中断
EMUIE1.AllQ_NegCFIE	EMUIF1.AllQ_NegCFIF	全波无功通路 CF 反向脉冲中断
EMUIE1.S_NegCFIE	EMUIF1.S_NegCFIF	视在 通路 CF 反向脉冲中断
EMUIE1.FundP_NegCFIE	EMUIF1.FundP_NegCFIF	基波有功通路 CF 反向脉冲中断
EMUIE1.FundQ_NegCFIE	EMUIF1.FundQ_NegCFIF	基波无功通路 CF 反向脉冲中断
EMUIE1.HarP_NegCFIE	EMUIF1.HarP_NegCFIF	谐波有功通路 CF 反向脉冲中断
EMUIE1.ZCUaIE	EMUIF1.ZCUaIF	A 相电压过零中断
EMUIE1.ZCUbIE	EMUIF1.ZCUbIF	B 相电压过零中断
EMUIE1.ZCUcIE	EMUIF1.ZCUcIF	C 相电压过零中断
EMUIE1.ZCIaIE	EMUIF1.ZCIaIF	A 相电流过零中断
EMUIE1.ZCIbIE	EMUIF1.ZCIbIF	B 相电流过零中断
EMUIE1.ZCIcIE	EMUIF1.ZCIcIF	C 相电流过零中断
EMUIE1.ZCInIE	EMUIF1.ZCInIF	零线电流过零中断
EMUIE1.CMPaIE	EMUIF1.CMPaIF	电流通道模拟比较器 A 相中断
EMUIE1.CMPbIE	EMUIF1.CMPbIF	电流通道模拟比较器 B 相中断
EMUIE1.CMPcIE	EMUIF1.CMPcIF	电流通道模拟比较器 C 相中断
EMUIE1.ZCUaLostIE	EMUIF1.ZCUaLostIF	A 相电压过零丢失中断
EMUIE1.ZCUbLostIE	EMUIF1.ZCUbLostIF	B 相电压过零丢失中断
EMUIE1.ZCUcLostIE	EMUIF1.ZCUcLostIF	C 相电压过零丢失中断
EMUIE1.EnergyOVIE	EMUIF1.EnergyOVIF	电能溢出中断
EMUIE1.SPLUpdateIE	EMUIF1.SPLUpdateIF	SPL 寄存器的更新中断
EMUIE1.PowerUpdateIE	EMUIF1.PowerUpdateIF	Power 寄存器的更新中断
EMUIE1.RMSUpdateIE	EMUIF1.RMSUpdateIF	RMS 寄存器的更新中断
EMUIE1.RMSTUpdateIE	EMUIF1.RMSTUpdateIF	RMST 寄存器的更新中断
EMUIE1.RMSUtOVIE	EMUIF1.RMSUtOVIF	RMS_Ut 与 RMSUt_THO 比较中断
EMUIE1.RMSItOVIE	EMUIF1.RMSItOVIF	RMS_It 与 RMSIt_THO 比较中

				断
		EMUIE2.PeakUaIE	EMUIF2.PeakUaIF	A 相发生 Peak 事件中断
		EMUIE2.PeakUbIE	EMUIF2.PeakUbIF	B 相发生 Peak 事件中断
		EMUIE2.PeakUcIE	EMUIF2.PeakUcIF	C 相发生 Peak 事件中断
		EMUIE2.SagUaIE	EMUIF2.SagUaIF	A 相发生 SAG 事件中断
		EMUIE2.SagUbIE	EMUIF2.SagUbIF	B 相发生 SAG 事件中断
		EMUIE2.SagUcIE	EMUIF2.SagUcIF	C 相发生 SAG 事件中断
		EMUIE2.UaEndIE	EMUIF2.UaEndIF	A 相退出 Peak 或 SAG 事件中断
		EMUIE2.UbEndIE	EMUIF2.UbEndIF	B 相退出 Peak 或 SAG 事件中断
		EMUIE2.UcEndIE	EMUIF2.UcEndIF	C 相退出 Peak 或 SAG 事件中断
		EMUIE2.INTUaIE	EMUIF2.INTUaIF	A 相发生 INT 事件中断
		EMUIE2.INTUbIE	EMUIF2.INTUbIF	B 相发生 INT 事件中断
		EMUIE2.INTUcIE	EMUIF2.INTUcIF	C 相发生 INT 事件中断
		EMUIE2.INTUIE	EMUIF2.INTUIF	ABC 三相都发生 INT 事件中断
		EMUIE2.UaINTEndIE	EMUIF2.UaINTEndIF	A 相退出 INT 事件中断
		EMUIE2.UbINTEndIE	EMUIF2.UbINTEndIF	B 相退出 INT 事件中断
		EMUIE2.UcINTEndIE	EMUIF2.UcINTEndIF	C 相退出 INT 事件中断
		EMUIE2.IaOVIE	EMUIF2.IaOVIF	A 相发生 OverFlow 事件中断
		EMUIE2.IbOVIE	EMUIF2.IbOVIF	B 相发生 OverFlow 事件中断
		EMUIE2.IcOVIE	EMUIF2.IcOVIF	C 相发生 OverFlow 事件中断
		EMUIE2.InOVIE	EMUIF2.InOVIF	零线发生 OverFlow 事件中断
		EMUIE2.IaOVEndIE	EMUIF2.IaOVEndIF	A 相退出 OverFlow 事件中断
		EMUIE2.IbOVEndIE	EMUIF2.IbOVEndIF	B 相退出 OverFlow 事件中断
		EMUIE2.IcOVEndIE	EMUIF2.IcOVEndIF	C 相退出 OverFlow 事件中断
		EMUIE2.InOVEndIE	EMUIF2.InOVEndIF	零线退出 OverFlow 事件中断
		EMUIE2.VpupdUaIE	EMUIF2.VpupdUaIF	A 相电压通道的半波峰值更新中断
		EMUIE2.VpupdUbIE	EMUIF2.VpupdUbIF	B 相电压通道的半波峰值更新中断
		EMUIE2.VpupdUcIE	EMUIF2.VpupdUcIF	C 相电压通道的半波峰值更新中断
		EMUIE2.VpupdIaIE	EMUIF2.VpupdIaIF	A 相电流通道的半波峰值更新中断
		EMUIE2.VpupdIbIE	EMUIF2.VpupdIbIF	B 相电流通道的半波峰值更新中断
		EMUIE2.VpupdIcIE	EMUIF2.VpupdIcIF	C 相电流通道的半波峰值更新中断
		EMUIE2.VpupdInIE	EMUIF2.VpupdInIF	零线电流通道的半波峰值更新中断
EWU	8	EWUIE.HpRmsIE	EWUIF.HpRmsIF	半波有效值更新中断
		EWUIE.FlickerUpdIE	EWUIF.FlickerUpdIF	Flicker 结果更新中断
UART0-5	9-14	UARTCON.RXIE	UARTSTA.RXIF	UART 接收中断

(UART3 和 UART4 有 7816 功能)		UARTCON.TXIE	UARTSTA.TXIF	UART 发送中断
		UARTCON.PRDIIE	UARTSTA.PRDIIF	UART 溢出中断
		ISO7816CON.PRDIIE	ISO7816STA.PRDIIF	7816 溢出中断 (UART3, UART4)
		ISO7816CON.RXIE	ISO7816STA.RXIF	7816 接收中断 (UART3, UART4)
		ISO7816CON.TXIE	ISO7816STA.TXIF	7816 发送中断 (UART3, UART4)
TMR0-3	15-18	TMRIE.ACIE	TMRIF.ACIF	事件计数中断
		TMRIE.CMPIE	TMRIF.CMPIF	比较中断
		TMRIE.CAPIE	TMRIF.CAPIF	捕获中断
		TMRIE.PRDIIE	TMRIF.PRDIIF	周期性溢出中断
TBS	19	TBSIE.VTP7IE	TBSIF.VTP7IF	测温端子通道 7 测量中断
		TBSIE.VTP6IE	TBSIF.VTP6IF	测温端子通道 6 测量中断
		TBSIE.VTP5IE	TBSIF.VTP5IF	测温端子通道 5 测量中断
		TBSIE.VTP4IE	TBSIF.VTP4IF	测温端子通道 4 测量中断
		TBSIE.VTP3IE	TBSIF.VTP3IF	测温端子通道 3 测量中断
		TBSIE.VTP2IE	TBSIF.VTP2IF	测温端子通道 2 测量中断
		TBSIE.VTP1IE	TBSIF.VTP1IF	测温端子通道 1 测量中断
		TBSIE.VTP0IE	TBSIF.VTP0IF	测温端子通道 0 测量中断
		TBSIE.ADC5IE	TBSIF.ADC5IF	ADC 通道 5 测量中断
		TBSIE.ADC4IE	TBSIF.ADC4IF	ADC 通道 4 测量中断
		TBSIE.ADC3IE	TBSIF.ADC3IF	ADC 通道 3 测量中断
		TBSIE.ADC0CMPIE	TBSIF.ADC0CMPIF	ADC0 比较中断
		TBSIE.VCCIE	TBSIF.VCCIF	电源电压测量中断
		TBSIE.TDVREFCMPIE	TBSIF.TDVREFCMPIF	TDVERFCMP 比较中断
		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.TDVREFIE	TBSIF.TDVREFIF	EMU 模块 VREF 测量中断
TBSIE.TMPIE	TBSIF.TMPIF	温度测量中断		
RTC	20	RTCIE.ALMIE	RTCIF.ALMIF	RTC 闹铃中断
		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
		RTCTS0IE.TS0OVIE	RTCTS0IF.TS0OVIF	RTC 时间戳事件 STAMP0 计数溢出中断
		RTCTS0IE.TS0CMPIE	RTCTS0IF.TS0CMPIF	RTC 时间戳事件 STAMP0 计数

				比较中断
		RTCTS0IE.TS0IE	RTCTS0IF.TS0IF	RTC 时间戳 STAMP0 触发中断
		RTCTS1IE.TS10VIE	RTCTS1IF.TS10VIF	RTC 时间戳事件 STAMP1 计数溢出中断
		RTCTS1IE.TS1CMPIE	RTCTS1IF.TS1CMPIF	RTC 时间戳事件 STAMP1 计数比较中断
		RTCTS1IE.TS1IE	RTCTS1IF.TS1IF	RTC 时间戳 STAMP1 触发中断
I2C	21		I2CCON.SI	I2C 中断标志位
SPI0	22		SPI0STA.SPIF	SPI0 数据传输完成中断
			SPI0STA.MODF	SPI0 模式故障中断
		FIFOIE.TXFIFO_DONE_IE	FIFOIF.TXFIFO_DONE_IE	SPI0 FIFO 数据发送完成中断使能位
		FIFOIE.RXT_IE	FIFOIF.RXT_IF	SPI0 接收 FIFO 大于触发阈值中断
		FIFOIE.TXT_IE	FIFOIF.TXT_IF	SPI0 发送 FIFO 小于等于触发阈值中断
		FIFOIE.RXO_IE	FIFOIF.RXO_IF	SPI0 接收 FIFO 溢出中断
		FIFOIE.TXO_IE	FIFOIF.TXO_IF	SPI0 发送 FIFO 溢出中断
		FIFOIE.RXF_IE	FIFOIF.RXF_IF	SPI0 接收 FIFO 满中断
		FIFOIE.TXF_IE	FIFOIF.TXF_IF	SPI0 发送 FIFO 满中断
		FIFOIE.RXNE_IE	FIFOIF.RXNE_IF	SPI0 接收 FIFO 非空中断
FIFOIE.TXE_IE	FIFOIF.TXE_IF	SPI0 发送 FIFO 空中断		
SPI1	23		SPI1STA.SPIF	SPI1 数据传输完成中断
			SPI1STA.MODF	SPI1 模式故障中断
SelfTestFreq	24	Reserved		
TMR4-5	25-26	TMRIE.TRIGIE	TMRIE.TRIGIF	外部触发中断
		TMRIE.ACIE	TMRIE.ACIF	事件计数中断
		TMRIE.CMPIE	TMRIE.CMPIF	比较中断
		TMRIE.CAPIE	TMRIE.CAPIF	捕获中断
		TMRIE.PRDIE	TMRIE.PRDIF	周期性溢出中断
UART6	27	UARTCON.RXIE	UARTSTA.RXIF	UART 接收中断
		UARTCON.TXIE	UARTSTA.TXIF	UART 发送中断
		UARTCON.PRDIE	UARTSTA.PRDIF	UART 溢出中断
EXTI7-9	28	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
SPI3	29		SPI3STA.SPIF	SPI3 数据传输完成中断
			SPI3STA.MODF	SPI3 模式故障中断
SPI2	30		SPI2STA.SPIF	SPI2 数据传输完成中断
			SPI2STA.MODF	SPI2 模式故障中断
		FIFOIE.TXFIFO_DONE_IE	FIFOIF.TXFIFO_DONE_IE	SPI2 FIFO 数据发送完成中断使能位

		FIFOIE.RXT_IE	FIFOIF.RXT_IF	SPI2 接收 FIFO 大于触发阈值中断
		FIFOIE.TXT_IE	FIFOIF.TXT_IF	SPI2 发送 FIFO 小于等于触发阈值中断
		FIFOIE.RXO_IE	FIFOIF.RXO_IF	SPI2 接收 FIFO 溢出中断
		FIFOIE.TXO_IE	FIFOIF.TXO_IF	SPI2 发送 FIFO 溢出中断
		FIFOIE.RXF_IE	FIFOIF.RXF_IF	SPI2 接收 FIFO 满中断
		FIFOIE.TXF_IE	FIFOIF.TXF_IF	SPI2 发送 FIFO 满中断
		FIFOIE.RXNE_IE	FIFOIF.RXNE_IF	SPI2 接收 FIFO 非空中断
		FIFOIE.TXE_IE	FIFOIF.TXE_IF	SPI2 发送 FIFO 空中断
DMA	31	DMAIE2.PEIE <sub>x</sub>	DMAIF2.PEIF <sub>x</sub>	DMA 包传输中断 (x=12)
		DMAIE2.TEIE <sub>x</sub>	DMAIF2.TEIF <sub>x</sub>	DMA 传输错误中断 (x=12)
		DMAIE2.BCIE <sub>x</sub>	DMAIF2.BCIF <sub>x</sub>	DMA 块传输中断 (x=12)
		DMAIE1.TCIE <sub>x</sub>	DMAIF1.TCIF <sub>x</sub>	DMA 传输结束中断 (x=8~11)
		DMAIE1.BCIE <sub>x</sub>	DMAIF1.BCIF <sub>x</sub>	DMA 块传输中断 (x=8~11)
		DMAIE1.TEIE <sub>x</sub>	DMAIF1.TEIF <sub>x</sub>	DMA 传输错误中断 (x=8~11)
		DMAIE.TCIE <sub>x</sub>	DMAIF.TCIF <sub>x</sub>	DMA 传输结束中断 (x=0~7)
		DMAIE.BCIE <sub>x</sub>	DMAIF.BCIF <sub>x</sub>	DMA 块传输中断 (x=0~7)
		DMAIE.TEIE <sub>x</sub>	DMAIF.TEIF <sub>x</sub>	DMA 传输错误中断 (x=0~7)

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 接收中断和 7816 发送中断。
2. 对于中断号大于等于 0 的中断, 每个中断都有一个对应的中断使能信号, 具体配置参见 21.4 CMSIS 函数说明。
3. NMI/HardFault/SVCALL/PendSV/SysTick 为内核自带, 这些中断没有专门的中断使能控制位。

## 8.2 EXTI 中断说明

EXTI 中断即为外部 INT 口管脚中断, 共有 10 个 INT 口, 即 INT0-INT9, 每个 INT 口都可以配置为上升沿触发和下降沿触发。用户在使用 INT 功能前应先将对应用 IO 口配置为 INT 复用功能。

## 8.3 特殊功能寄存器列表

基地址: 0x40011800				
偏移地址	名称	读写方式	复位值	功能描述
0x00	EXTIE	R/W	0x0000	外部中断边沿配置寄存器
0x04	EXTIF	R/W	0x0000	外部中断标志寄存器
0x08	PINFLT	R/W	0x007F	引脚数字滤波使能寄存器
0x10	EXTIE2	R/W	0x0000	外部中断边沿配置寄存器 2
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 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]	<b>INT0-6 外部输入引脚上升沿使能</b> 0: 禁止 (default) 1: 使能
FIE[6:0]	<b>INT0-6 外部输入引脚下降沿使能</b> 0: 禁止 (default) 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 (default) 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 (default) 1: 产生中断

## PINFLT (引脚数字滤波使能寄存器)

PINFLT (引脚数字滤波使能寄存器)		基地址: 0x40011800 偏移地址: 08H						
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	X	RXFLT[6:0]						
<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	INTFLT[6:0]						
<b>Write:</b>		X	X	X	X	X	X	X
<b>Reset:</b>	0	1	1	1	1	1	1	1

位	功能描述
RXFLT[6:0]	<b>RX0-6PIN 输入引脚数字滤波功能使能</b> 0: 禁止 (default) 1: 使能
INTFLT[6:0]	<b>INT0-6PIN 输入引脚数字滤波功能使能</b> 0: 禁止 1: 使能 (default)

注:

1. INT 中断引脚对应的数字滤波功能固定打开, 无法关闭, 仅在对应引脚复用为 INT 功能时有效 (GPIO 或其它复用功能时无效)。
2. 用户在使用的 RX 中断引脚功能的时候, 必须要将 PINFLT 寄存器中的相应的引脚数字滤波功能也打开。其数字滤波功能仅在对应引脚复用为 RX 功能时有效 (GPIO 或其它复用功能时无效)。

## EXTIE2 (外部中断边沿配置寄存器 2)

EXTIE2 (外部中断边沿配置寄存器 2)		基地址: 0x40011800 偏移地址: 10H						
	Bit15	14	13	12	11	10	9	Bit8

<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]	<b>INT7-9 外部输入引脚上升沿中断使能</b> 0: 禁止 (default) 1: 使能
FIE[2:0]	<b>INT7-9 外部输入引脚下降沿中断使能</b> 0: 禁止 (default) 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)		基地址: <b>0x40011800</b> 偏移地址: <b>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]	<b>INT7-9 外部输入引脚上升沿中断标志</b> 0: 未产生中断 (default) 1: 产生中断
FIF[2:0]	<b>INT7-9 外部输入引脚下降沿中断标志</b> 0: 未产生中断 (default) 1: 产生中断

**PINFLT2 (引脚数字滤波使能寄存器 2)**

<b>PINFLT2</b> (引脚数字滤波使能寄存器 2)			基地址: <b>0x40011800</b> 偏移地址: <b>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[9:7]		
<b>Write:</b>						X	X	X
<b>Reset:</b>	0	0	0	0	0	1	1	1

位	功能描述
INTFLT[9:7]	<b>INT7-9PIN 输入引脚数字滤波功能使能</b> 0: 禁止 1: 使能 (default)

注: INT 中断引脚对应的数字滤波功能固定打开, 无法关闭, 仅在对应引脚复用为 INT 功能时有效。

## 9 复位模块

### 9.1 复位优先级

芯片共有 8 种复位方式，可分三种复位优先级。

下表中的复位主要是指芯片中除了 RTC 模块之外的所有模块的复位。

RTC 模块为独立的，RTC 模块的复位只会被其独立的 VRTC 引脚电源的上电 POR 而复位。

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 RSTSTA 2, <b>RTC 模块 (包括 RTC-stamp) 的寄存器 (只有 VRTC 的 POR 可复位)</b> 3, TBS 模块的寄存器 (只有 VRTC 的 POR 可复位)
2	低电压掉电复位 (LBOR)		
3	外部引脚 /RST 复位	二级	1, 复位状态寄存器 RSTSTA 2, PMU 模块的寄存器: PMUCON, VDETCFG, VDETPCFG, PMUIF, PMUSTA, <b>LDOVREFCFG</b> 3, 唤醒标志寄存器 WAKEIF 4, RTC 模块的寄存器 5, TBS 模块的寄存器 6, WDT 模块的寄存器: WDTCNT
4	自定义软复位		
5	掉电复位 (BOR)		
6	看门狗复位 (WatchDog)	三级	1, 复位状态寄存器 RSTSTA 2, PMU 模块的寄存器: PMUCON, VDETCFG, VDETPCFG, PMUIF, PMUSTA <b>LDOVREFCFG</b> 3, 唤醒标志寄存器 WAKEIF 4, GPIO 模块的寄存器: IOCFG, AFCFG, PTDIR, PTUP, PTOD, FILT, HIIPM, HDPORT 5, EXTI 模块寄存器: EXTIE <sub>x</sub> , EXTIF <sub>x</sub> , PINFLT <sub>x</sub> 6, Timer4/5 模块寄存器 7, CMU 相关寄存器: CLKCTRL0 (HRC_EN 和 PLL_EN 被复位), CLKCTRL1, LRCADJ, HRCADJ, PLLLOCKPRE 8, RTC 模块的寄存器 9, TBS 模块的寄存器 10, Fosc 停振检测模块的寄存器: LFDETCFG, OSCSTA, CLKSTA 11, 低频时钟选择寄存器: LFCLKCFG 12, WDT 模块的寄存器: WDTCNT (除调试复位和看门狗复位)
7	调试复位和软复位		
8	唤醒复位 (WakeUp Reset)		

注 1: RSTSTA 的 POR 复位标志和 LBOR 复位标志之间可以互相清除

注 2: LRCADJ 会被 Watchdog 和 Debug Reset 复位

注 3: 软复位 (SoftReset) 属于 ARM Cortex M0 内核自带复位, 不能复位所有寄存器

## 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, LBOR, 外部 RST, 自定义软复位和 BOR 复位时, 内部复位信号 IRST 将保持有效, 并保持 1088 个 Fosc/Flrc

2) WDT Reset, WakeUp Reset 复位时, 内部 IRST 信号有效, 并保持 64 个 Fosc/Flrc。

3) 只有 VRTC 引脚的 POR 复位会对 RTC、RTC-stamp 和 TBS 模块复位 (可配, 参见 2.5Flash 控制功能)。

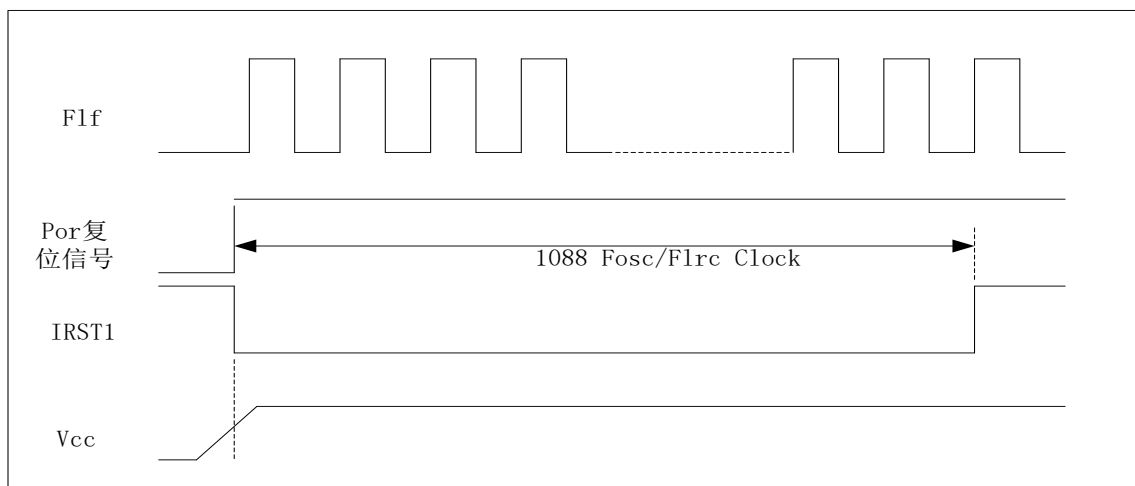
复位预热计数的时钟源选择 (Fosc/Flrc), 由 Fosc 停振检测模块 (LF\_DET) 的停振标志 LF\_FLAG 控制, LF\_FLAG=0, 选择 Fosc 时钟; LF\_FLAG=1, 选择 Flrc 时钟; 若 Fosc 停振检测关闭, LF\_FLAG 固定为 0, 复位计数时钟强制选择 Fosc, 此时 Fosc 停振, 会影响复位预热计数功能, 建议系统正常运行打开 Fosc 停振检测, 增强复位可靠性。

### 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 Vcc 上升到阈值 0.3V 时, POR 输出高电平, 指示发生上电。内部复位信号 IRST 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:

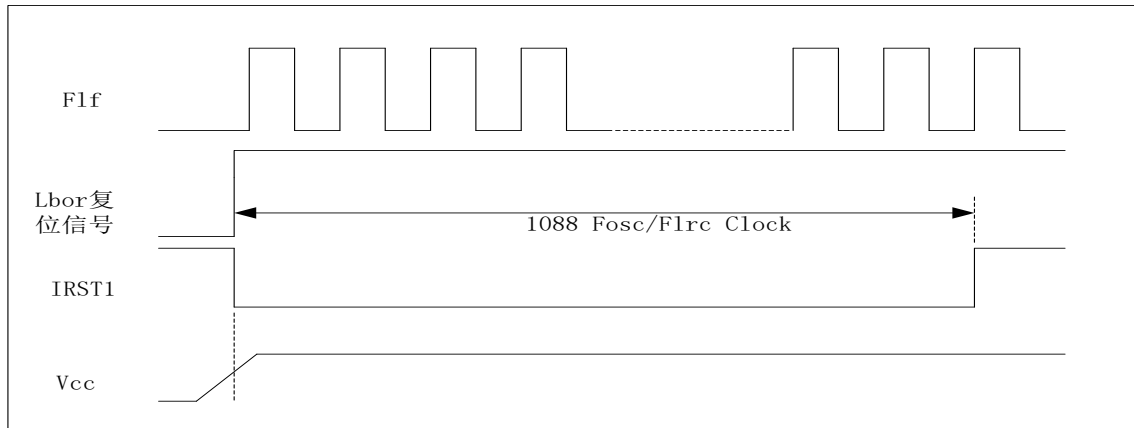
- 产生一个 POR 脉冲
- RTC 的复位只能由 VRTC 的电源的上电 POR 复位
- 内部复位信号 IRST 有效
- 计数 1088 个 Fosc/Flrc
- 复位状态寄存器 RSTSTA 的上电复位标志位 POR 被设置为 1, 其他 RSTSTA 为被清为 0。
- CPU 从地址 0000H 执行程序



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。

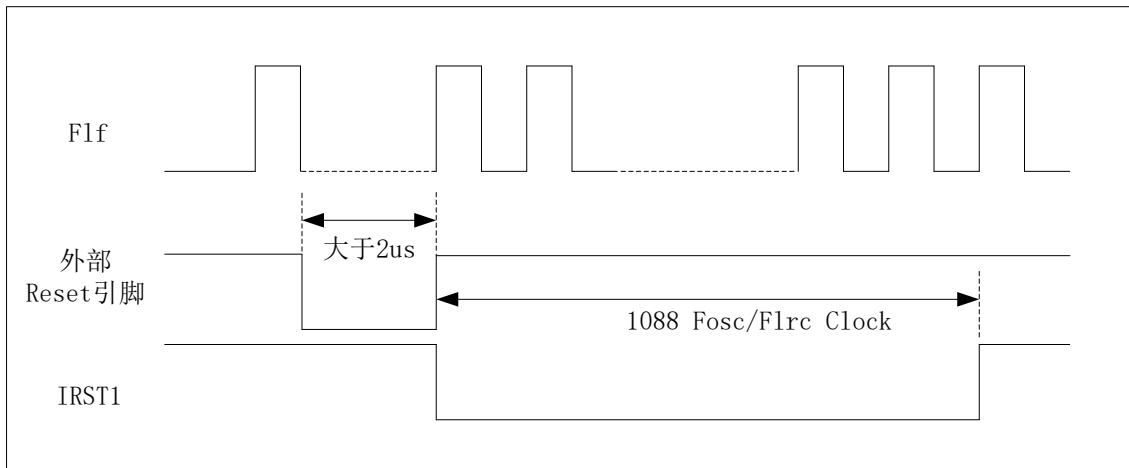


LBOR 复位说明

### 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 IRST1 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 IRST1 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

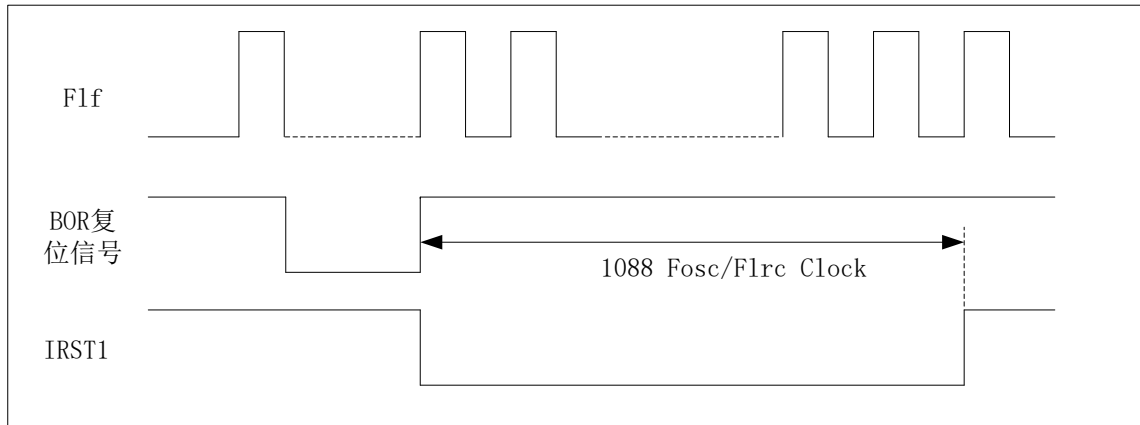
### 9.2.4 掉电复位

当掉电检测电路检测到电源电压低于电压 Vbor 时，BOR 输出低电平，内部复位信号 IRST1 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 Vbor 时，BOR 输出高电平，IRST1 在 1088 个 Flf 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 IRST1 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。

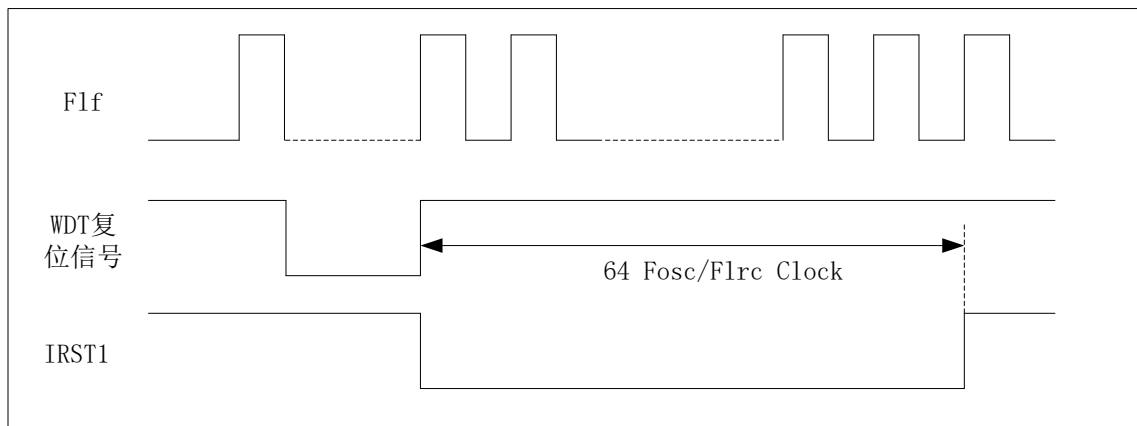
- CPU 从 0000H 开始执行程序



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 IRST1 有效，复位状态寄存器的 WDT 复位标志位 WDT 被设置为 1。WDT 的复位脉宽为 64 个 Fosc/Flrc。



WDT 复位

## 9.2.6 软复位

当向系统控制块的应用中断与控制状态寄存器寄存器 AIRCR 的 bit2 写入 1 时，则产生一个软复位。推荐用户使用标准的 CMSIS 函数来操作软复位，具体使用参见 24.4 CMSIS 函数说明。

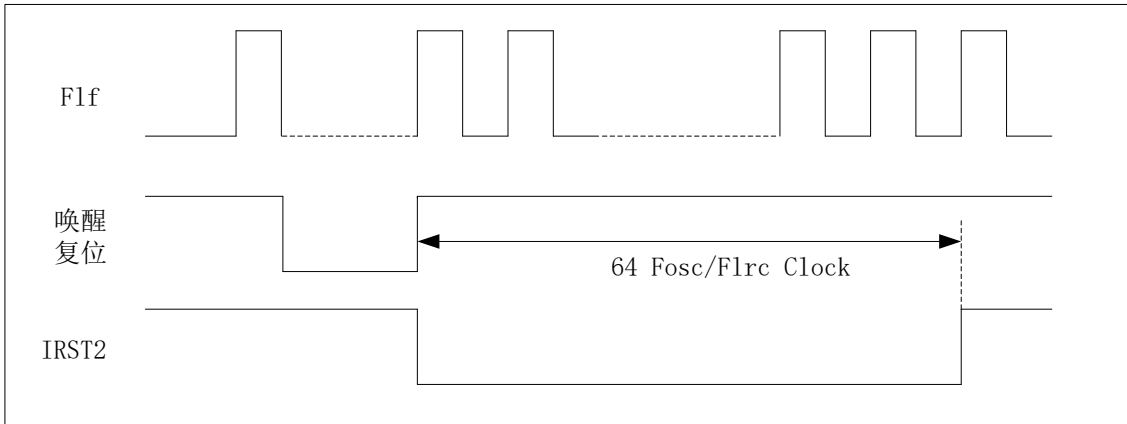
## 9.2.7 调试复位

这种复位只在 JTAG 调试状态下才有可能产生。

## 9.2.8 唤醒复位

出现唤醒事件时，按照下面顺序执行：

- 内部复位信号 IRST2 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 IRST2



唤醒复位

## 9.3 特殊功能寄存器列表

基地址：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	0x0011	复位标志寄存器

## 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)			基地址： 0x4000F400 偏移地址： 30H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Hold_Fla	Sleep_F	X	X	X	X	X	BORRST
Write:	g	lag						
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DebugRS	SoftRS	ExtRST	RESERV	Wakeup	WDTR	LBORRS	PORRST

<b>Write:</b>	T	T		ED	RST	ST	T	
<b>Reset:</b>	0	0	0	1	0	0	0	1

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
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
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
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
RESERVED	该标志位对用户无意义
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
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
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
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
--------------------------

注 1: LBOR 复位和 POR 复位发生, 置对应复位标志, 并清除其它复位标志; 除 LBOR 和 POR, 其它复位发生, 仅置对应复位标志, 不会清除其它复位标志。

注 2: LBOR 复位和 POR 复位改进设计, 提高可靠性, 用户在进行复位标志判断时建议使用 (LBORRST|PORRST), 可保证软件兼容性。

## 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:

- 共七路UART
- UART3、UART4支持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115200bps
- 全双工通信口
- 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
- 具有奇偶校验功能并可给出校验标志
- 具有接收/发送中断使能
- 每路UART都可以调制成38K红外信号，调制信号源可选PLLX2/HRC10M。调制信号的占空比可调，调制极性可选，最大波特率不超过2400bps
- 支持在 SLEEP 和 HOLD 模式，RX 引脚可配置唤醒触发源

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:

- UART3、UART4支持7816接口复用，通过MODESEL寄存器选择
- 波特率设置与UART波特率设置相同
- 响应位长度支持1、1.5或2bit，支持奇偶校验功能
- 支持收发数据状态查询，并给出相应标志
- 具有接收/发送中断使能
- 支持错误重收发功能和重收发次数设置

### 10.2 波特率计算

串口波特率计算公式如下:

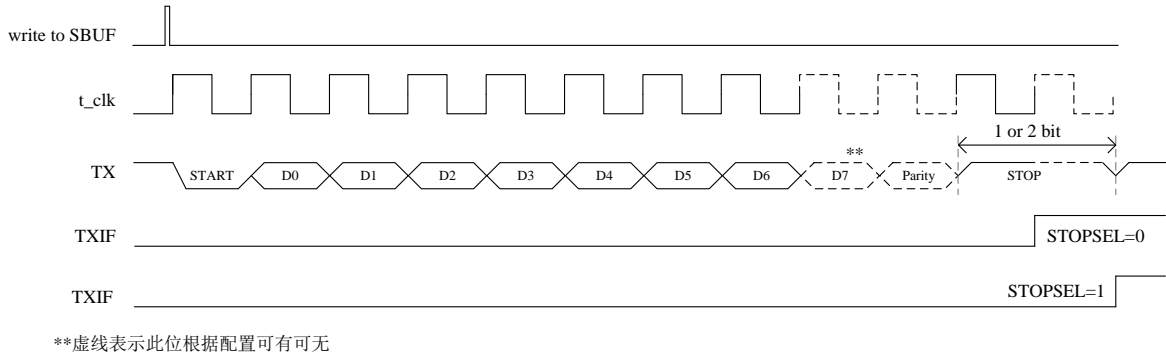
$$\text{波特率} = \frac{F_{sys}}{2 \times (SREL + 1)}$$

其中 SREL 是 16 位无符号数（不可配置为 0 和 1）；F<sub>sys</sub> 是系统时钟。

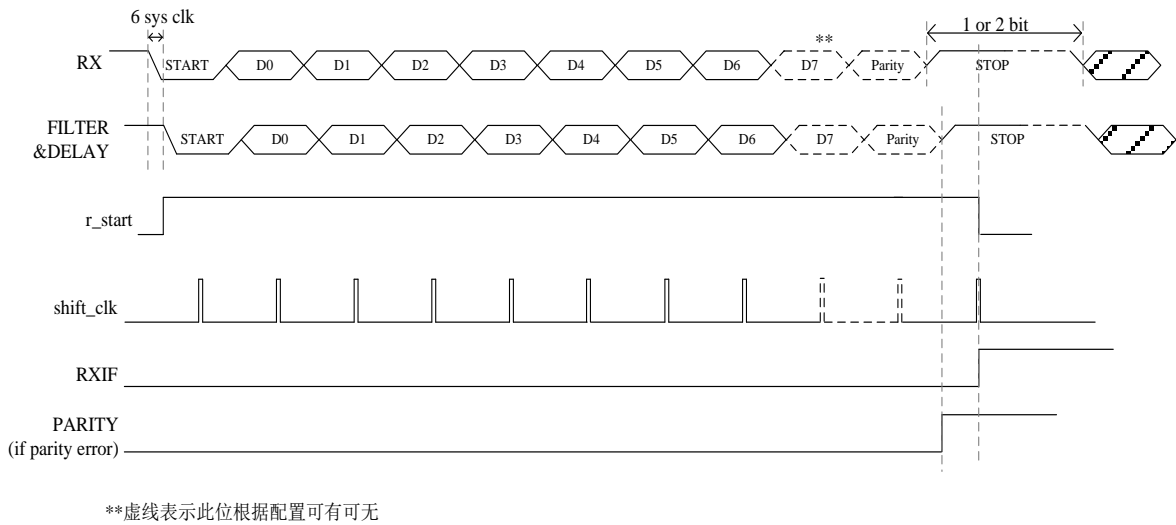
注：如果需要支持较高通信速率时，需要关闭对应 IO 的模拟滤波功能。

### 10.3 串口通讯模式说明

TX 引脚为数据发送端，当一帧中停止位发送完成时，发送中断标志 TXIF 置“1”，如下图所示:



RX 引脚为数据接收端，接收完停止位采样点后，接收中断标志 RXIF 置“1”，如下图所示：



### 10.3.1 方式 1

方式 1 每帧包含 9 或 10 位数据信息：1 位起始位，7 位数据位（低位在前），1 或 2 位停止位。一帧数据波形如下图所示：



图示：方式 1 时单帧数据时序

### 10.3.2 方式 2

方式 2 每帧包含 10 或 11 位数据信息：1 位起始位，7 位数据位（低位在前），1 位奇偶校验位，1 或 2 位停止位，一帧数据波形如下图所示：



图示：方式 2 时单帧数据时序

### 10.3.3 方式 3

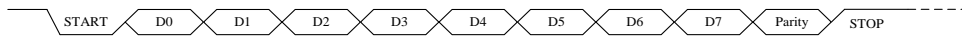
方式 3 每帧包含 10 或 11 位数据信息：1 位起始位，8 位数据位（低位在前），1 或 2 位停止位，一帧数据波形如下图所示：



图示：方式 3 时单帧数据时序

### 10.3.4 方式 4

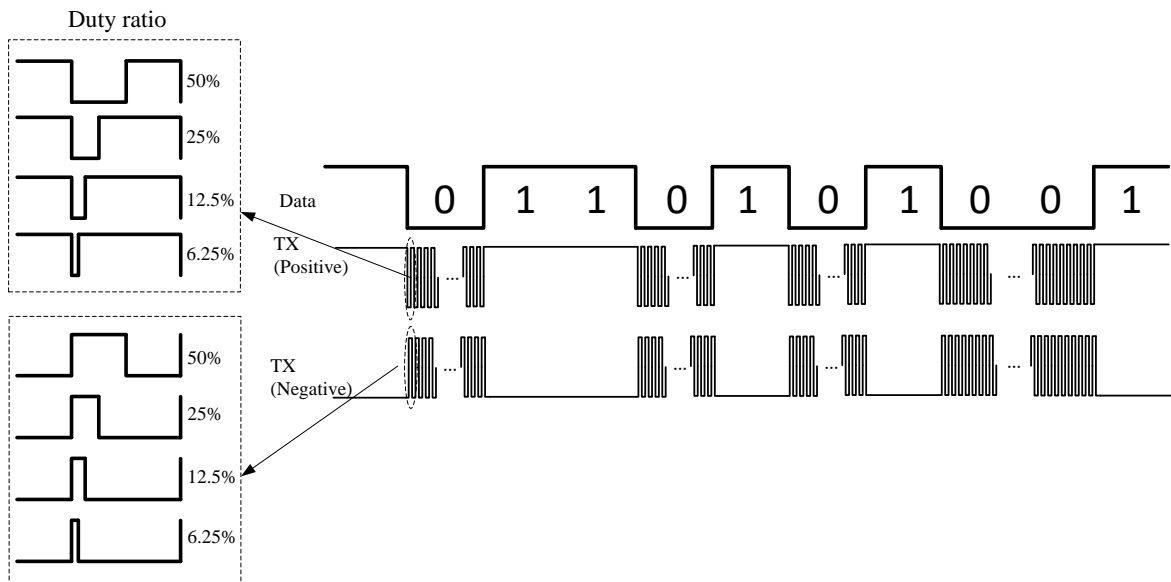
方式 4 每帧包含 11 或 12 位数据信息：1 位起始位，8 位数据位（低位在前），1 位奇偶校验位，1 或 2 位停止位，一帧数据波形如下图所示：



图示：方式 4 时单帧数据时序

## 10.4 红外调制

发送器可配置为红外调制输出，开启 IRTX 红外调制功能使能控制可打开红外调制功能，红外调制极性可选，调制信号的占空比可调，最大波特率不超过 2400bps。调制信号如下图所示：



红外调制波形示意图

## 10.5 7816 接收和发送

### 10.5.1 7816 数据发送

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

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

### 10.5.2 7816 数据接收

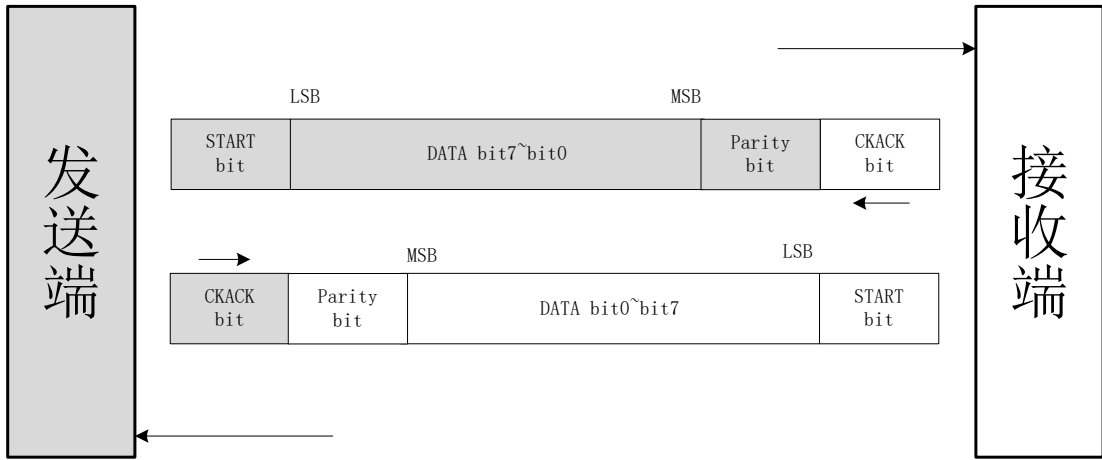
在 IDLE 状态下, 如果在接收端口(IO)上检测到下降沿, 即启动一次接收数据流程。该流程包括几个步骤, 每一步均需要一个或者几个 ETU。

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

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

10.5.37816 通讯示意图

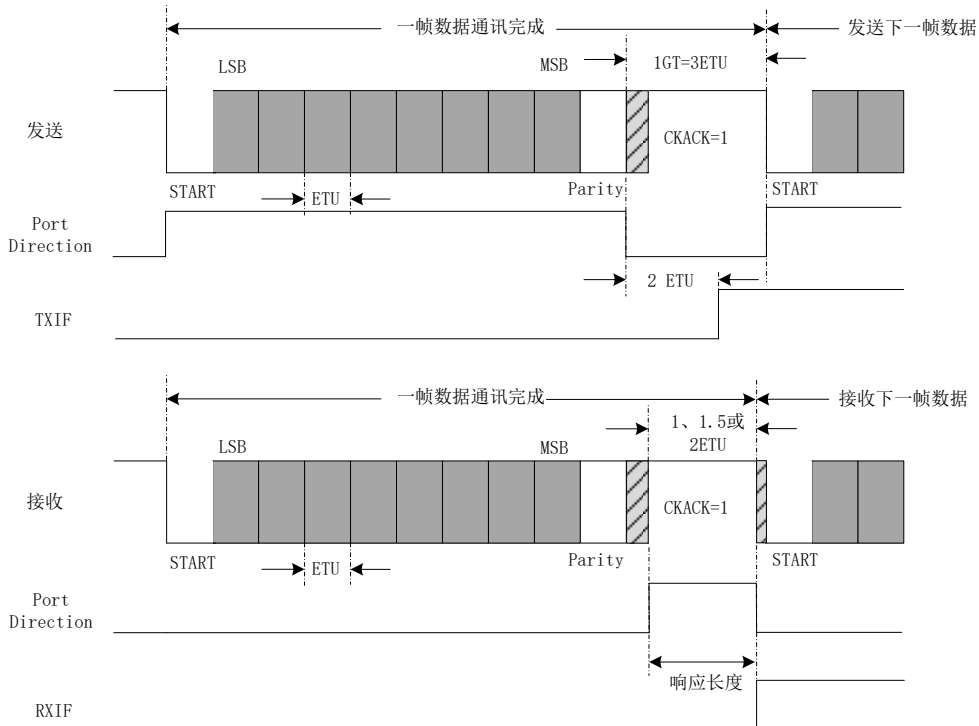


7816 通讯数据示意图

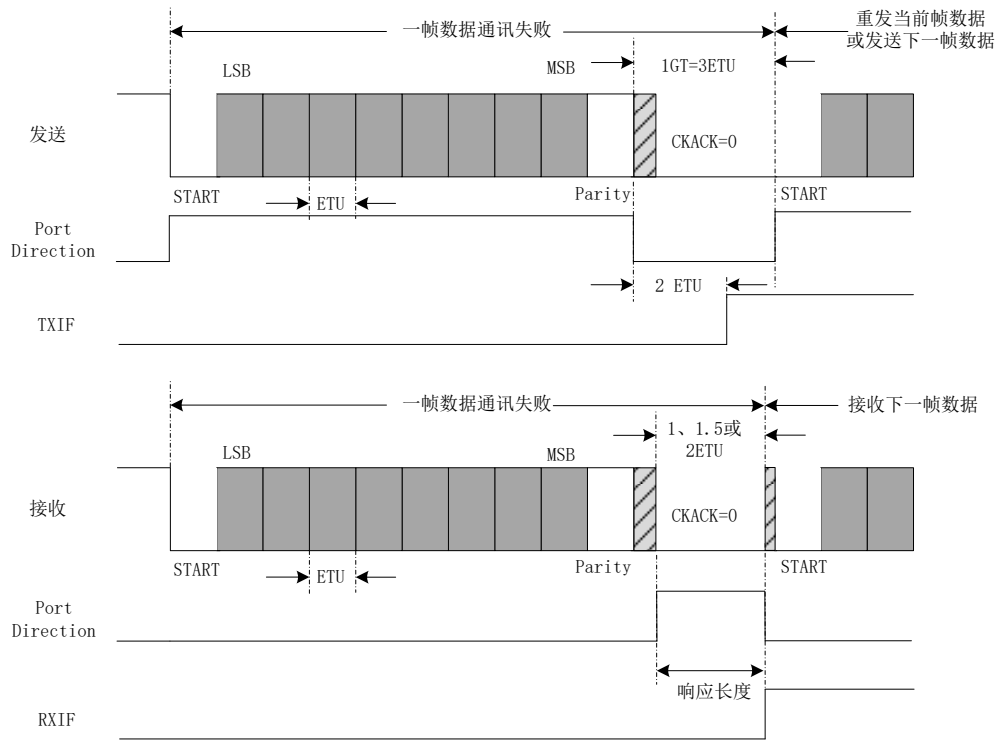
进行 7816 通信时，收发一个 bit 的时间被定义为基础时间单位 ETU (Elementary Time Unit)，连续发送两个数据帧之间的等待时间，该等待时间被定义为检测时间 GT(Guarding Time)，一般  $1 GT = 3 ETU$ 。

发送端发送完一个数据帧后，接收端需要一定的时间对接收到的数据进行校验，然后再根据校验结果返回 CKACK 信号给发送端。发送端根据 CKACK 信号和 AUTOTXEN 配置决定发送下一个数据帧或重发当前帧数据。

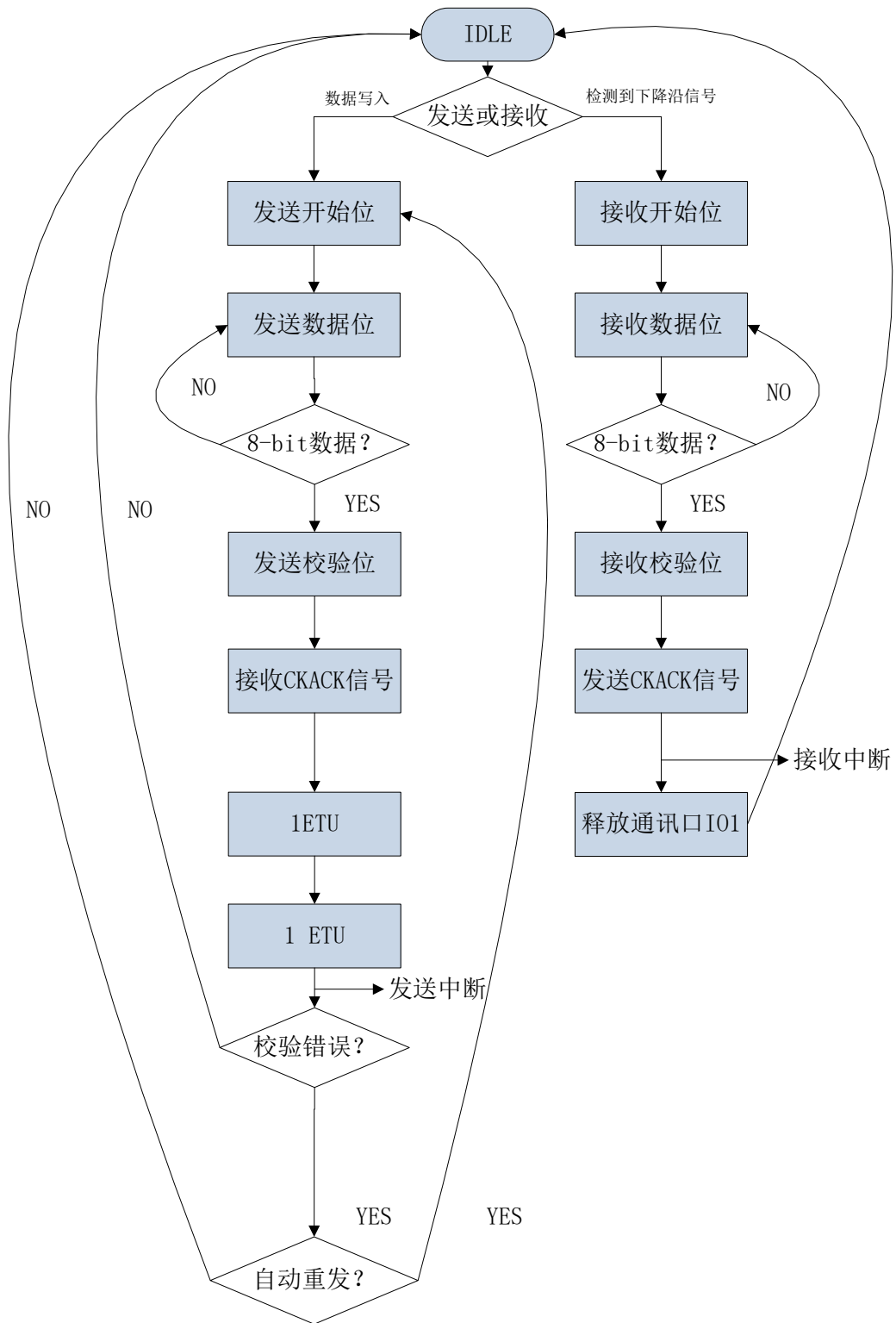
发送端收到接收端返回的数据帧时，根据接收的数据进行校验，然后再根据校验结果发送 CKACK 信号给接收端。



收发数据成功时序



收发数据失败时序



## 10.6 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); 0x40018000(UART6 端口);				
偏移地址	名称	读写方式	复位值	功能描述
0x00	MODESEL	R/W	0x0000	串口功能选择寄存器
0x04	UARTCON	R/W	0x0000	UART 功能配置寄存器
0x08	ISO7816CON	R/W	0x0000	7816 功能配置寄存器
0x0C	SREL	R/W	0x0000	串口波特率发生寄存器
0x10	SBUF	R/W	0x0000	串口数据缓冲寄存器
0x14	UARTSTA	R/W	0x0000	UART 状态寄存器
0x18	ISO7816STA	R/W	0x0000	7816 状态寄存器
0x30	IRCON	R/W	0x0000	红外调制控制寄存器
0x34	IRDUTY	R/W	0x0000	红外调制脉宽调整寄存器
0x38	NEGDETLN	R/W	0x00	485 逻辑正反自动判断数据长度设置

注：7816 的相关寄存器只对 UART3 与 UART4 的基地址有效

## 10.7 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; 0x40018000 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 (default) 1: 7816 功能
--	------------------------------------

## UARTCON (UART 功能配置寄存器)

<b>UARTCON</b> (UART 功能配置寄存器)		基地址: <b>0x40005000—0x40009000; 0x40000000; 0x40018000</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>	PRDIE	X	X	X	X	UNEGAUTO	UNEG	STOPSEL
<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>	LENSEL	PARITYSEL[1:0]		PARITYEN	RXIE	TXIE	RXEN	TXEN
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述															
PRDIE	串口上溢中断使能控制位 0: 上溢中断禁止 (default) 1: 上溢中断使能															
UNEGAUTO	UART 通讯中的正反逻辑判断 (该逻辑自动判断只针对数据接收) 0: 由用户软件控制 UART 接收是正逻辑还是负逻辑, 通过 UNEG 控制位 (default) 1: 由硬件自动配置 UART 接收是正逻辑还是负逻辑, UNEG 控制位无效															
UNEG	当 UNEGAUTO=0 时, 用户手动配置 UART 通讯中接收和发送是正逻辑或者是负逻辑 0: 正逻辑 (default) 1: 负逻辑 当 UNEGAUTO=1 时, 该寄存器位只能控制 UART 通讯中发送是正逻辑或者负逻辑 <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th>UNEGAUTO</th> <th>UNEG</th> <th>UART 发送和接收的逻辑状态</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>UART 发送, 接收都是正逻辑</td> </tr> <tr> <td>0</td> <td>1</td> <td>UART 发送, 接收都是负逻辑</td> </tr> <tr> <td>1</td> <td>0</td> <td>UART 接收根据外部 RX 默认状态高低自动判断逻辑 UART 发送是正逻辑</td> </tr> <tr> <td>1</td> <td>1</td> <td>UART 接收根据外部 RX 默认状态高低自动判断逻辑 UART 发送是负逻辑</td> </tr> </tbody> </table>	UNEGAUTO	UNEG	UART 发送和接收的逻辑状态	0	0	UART 发送, 接收都是正逻辑	0	1	UART 发送, 接收都是负逻辑	1	0	UART 接收根据外部 RX 默认状态高低自动判断逻辑 UART 发送是正逻辑	1	1	UART 接收根据外部 RX 默认状态高低自动判断逻辑 UART 发送是负逻辑
UNEGAUTO	UNEG	UART 发送和接收的逻辑状态														
0	0	UART 发送, 接收都是正逻辑														
0	1	UART 发送, 接收都是负逻辑														
1	0	UART 接收根据外部 RX 默认状态高低自动判断逻辑 UART 发送是正逻辑														
1	1	UART 接收根据外部 RX 默认状态高低自动判断逻辑 UART 发送是负逻辑														
STOPSEL	UART 通讯停止位长度选择位 0: 1bit (default) 1: 2bit															
LENSEL	UART 通讯数据长度选择位 0: 8bit (default) 1: 7bit															
PARITYSEL[1:0]	UART 奇偶校验选择位															

	00: 固定为 0 (default) 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 (default) 1: 使能
RXIE	<b>UART 接收中断使能位</b> 0: 禁止 (default) 1: 使能
TXIE	<b>UART 发送中断使能位</b> 0: 禁止 (default) 1: 使能
RXEN	<b>UART 接收使能位</b> 0: 禁止 (default) 1: 使能
TXEN	<b>UART 发送使能位</b> 0: 禁止 (default) 1: 使能

### ISO7816CON (7816 功能配置寄存器)

ISO7816CON (7816 功能配置寄存器)		基地址: 0x40008000—0x40009000 偏移地址: 08H						
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	X	X	X	X	X	X	7816PARITY	AUTORXEN
<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>	AUTOTXE	REPTR[1:0]		ACKLEN[1:0]		PRDIE	RXIE	TXIE
<b>Write:</b>	N							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
7816PARITY	<b>奇偶校验选择位</b> 0: 偶校验 (default) 1: 奇校验
AUTORXEN	<b>自动重接收使能位</b> 0: 禁止 (default) 1: 使能
AUTOTXEN	<b>自动重发送使能位</b>

	0: 禁止 (default) 1: 使能
REPTR[1:0]	自动重收重发的次数上限 00: 0 次 (default) 01: 1 次 10: 2 次 11: 3 次
ACKLEN[1:0]	响应位的长度 00: 1bit (default) 01: 1.5bit 10: 2bit 11: 保留位, 请勿配置
PRDIE	上溢中断使能位 0: 禁止 (default) 1: 使能
RXIE	接收中断使能位 0: 禁止 (default) 1: 使能
TXIE	发送中断使能位 0: 禁止 (default) 1: 使能

## SREL (串口波特率发生寄存器)

<b>SREL</b> (串口波特率发生寄存器)		基地址: 0x40005000—0x40009000; 0x40000000; 0x40018000 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>	SREL[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>	SREL[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
SREL[15:0]	串口波特率发生寄存器是一个 16 位的波特率分频系数, 其值建议设置为 0x0002~0xFFFF 之间的任一整数。波特率计算公式: $\text{波特率} = \frac{F_{sys}}{2 \times (SREL + 1)}$

**SBUF（串口数据缓冲寄存器）**

<b>SBUF</b> (串口数据缓冲寄存器)			基地址: 0x40005000—0x40009000; 0x40000000; 0x40018000 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>	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

位	功能描述
SBUF[7:0]	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)			基地址: 0x40005000—0x40009000; 0x40000000; 0x40018000 偏移地址: 14H					
	<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	POLASTA	PRDIF	PARITY	RXIF	TXIF
<b>Write:</b>				X				
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
POLASTA	<b>485 极性判断状态位</b> 0: 正极性 (default) 1: 负极性
PRDIF	<b>接收上溢中断标志</b> 0: 未产生上溢 (default) 1: 产生上溢, 也就是外部输入到 SBUF 中的数据, 还未被及时读出, 然后又有新的数据传输过来把旧的数据覆盖了

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

## ISO7816STA (7816 状态寄存器)

ISO7816STA (7816 状态寄存器)			基地址: 0x40008000—0x40009000 偏移地址: 18H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	RXERRSTAT	TXERRSTAT	PRDIF	RXIF	TXIF
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RXERRSTAT	接收数据的状态 0: 正确 (default) 1: 错误 写 0 清零，写 1 无效。
TXERRSTAT	发送数据的状态 0: 正确 (default) 1: 错误 写 0 清零，写 1 无效。
PRDIF	接收上溢中断标志 0: 未产生上溢 (default) 1: 产生上溢 写 0 清零，写 1 无效。
RXIF	接收中断标志

	0: 接收数据还未完成 (default) 1: 接收数据完成, 可从寄存器 SBUF 中读出 写 0 清零, 写 1 无效。
TXIF	发送中断标志 0: 发送未完成 (default) 1: 发送完成 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)			基地址: 0x40005000—0x40009000; 0x40000000; 0x40018000 偏移地址: 30H					
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	X	X	X	X	X	X	X	X
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	X	X	X	X	X	IR38kSource	IRLVL	IRTX
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
IR38kSource	红外 38k 调制信号来源 0: 红外 38k 调制信号来源于 PLLX2 (default) 1: 红外 38k 调制信号来源于 HRC10M
IRLVL	红外调制输出极性选择 0: 正极性 (default) 1: 负极性
IRTX	红外调制功能使能控制 0: 关闭 TX 输出的红外调制功能 (default) 1: 使能 TX 输出的红外调制功能

注意: 只有 IR38kSource 选择的时钟使能的情况下, 红外调制功能才是有效的。

## IRDUTY (红外调制脉宽寄存器)

<b>IRDUTY</b> (红外调制脉宽寄存器)			基地址: 0x40005000—0x40009000; 0x40000000; 0x40018000 偏移地址: 34H					
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	X	X	X	X	X	X	X	X

<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	X	X	X	X	X	X	IRDUTY[1:0]	
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
IRDUTY[1:0]	调制波形占空比配置 00: 50% (default) 01: 25% 10: 12.5% 11: 6.25%

### NEGDETLEN (485 逻辑正反自动判断时间设置)

NEGDETLEN (485 逻辑正反自动判断时间设置)			基地址: 0x40005000—0x40009000; 0x40000000; 0x40018000 偏移地址: 38H					
	<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>	NEGDETLEN[15:8]							
<b>Write:</b>	NEGDETLEN[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>	NEGDETLEN[7:0]							
<b>Write:</b>	NEGDETLEN[7:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
NEGDETLEN[15:0]	配置 485 逻辑正反自动判断的数据长度 注: 1 LSB 为串口单 bit 数据长度

# 11 WDT 模块

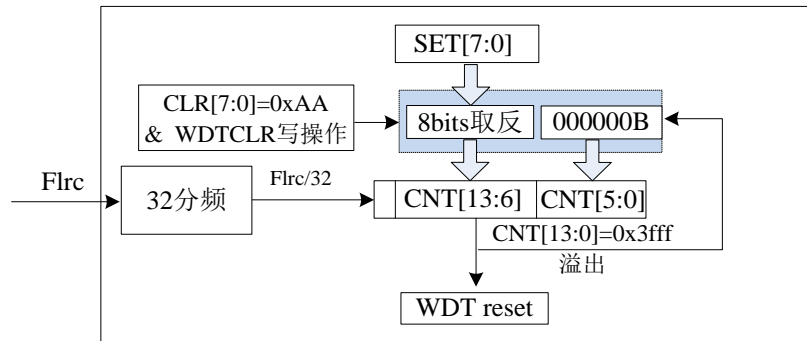
## 11.1 概述

有一个硬看门狗，其由内部低频 RC 时钟驱动，当计时器计满预定时间则发出溢出脉冲，产生 WDT 复位信号。在溢出脉冲发生前将 Watchdog Timer 清零，则不会发出 WDT 复位。特点如下：

- 采用硬件狗设计
- SLEEP或者HOLD模式下WDT开启/关闭可选
- 调试模式下可关闭，且随着CPU停止而停止，方便调试使用

执行喂狗操作（WDTCLR = 0xAAxx），系统对寄存器 **WDTCLR** 控制位 SET[7:0]8bits 取反,载入计数器 **WDTCNT** 的 CNT[13:6]位, CNT[5:0]被清零，向上重新开始计数，当计数器 CNT[13:0]计到 0x3FFF，发生 WDT 复位。

功能框图如下：



## 11.2 工作模式

不同工作模式下，WDT 的控制状态也有所区别，具体参见下表

工作模式	SOFTWDT_EN	JTAGWDTEN	WDT_EN[3:0]	WDT 模块
测试模式	X	1	X	X
调试模式	0	0	X	关闭
	1	0	X	开启，且可以随着 CPU 的停止而停止
正常模式	X	X	X	开启
Sleep/Hold 模式	X	X	0101b	关闭
	X	X	非 0101b	开启

注：1) X 表示无意义

2) 工作模式的定义详见工作模式说明

3) SOFTWDT\_EN 为 CLKCTRL1 寄存器的 bit14

4) WDT\_EN[3:0]由 Flash 地址 0x00000FC2 的数据决定，具体参见 Flash 控制功能

5) WDT 时钟由 LRC 驱动，如果关闭了 LRC，则 WDT 也不会工作。

6) 在调试模式下 (JTAGWDTEN=0&TEST=0)，可通过 SOFTWDT\_EN 控制位配置软狗是否开启，SOFTWDT\_EN=1 时，开启软狗功能，SOFTWDT\_EN=0 时，关闭软狗功能；软件看门狗与硬件看门狗的都有各自独立的 WDCNT 寄存器。

## 11.3 特殊功能寄存器列表

WDT 模块寄存器基地址：0x40010000				
偏移地址	名称	读写方式	复位值	功能描述
0x04	WDTCLR	R/W	0x0040	看门狗喂狗与时间配置寄存器
0x08	WDCNT	R	0x3000	看门狗计数寄存器（只读）

## 11.4 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，清除 WDT 内部计数器 WDCNT，写入其他值无效 该高 8bit 只能写入，不能读取，读出值永远为 0
SET[7:0]	<b>WDT 溢出时间设置：</b> WDT 溢出时间=64ms * (1+SET[7:0]) SET[7:0]为 8 位无符号数，由上面公式可以得出，最短的定时时间为 64ms，最长为 16384ms。默认为 4160ms。

**WDTCNT (WDT 计数寄存器)**

<b>WDTCNT</b> (WDT 计数寄存器)			基地址: <b>0x40010000</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	CNT[13:8]					
<b>Write:</b>			X	X	X	X	X	X
<b>Reset:</b>	0	0	1	1	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[13:0]	<b>WDT 计数寄存器:</b> 指示当前 WDT 内部的计数值

## 12 定时器模块

### 12.1 定时器(TMR)单元概述

共有 6 路定时器(TMR)，分为两种类型: TMR 0~3 使用时钟源为 Fsys; TMR 4、5 使用的时钟源为 Fpwm，可使用户在切换系统时钟时保持 TMR 时钟不变。

TMR 0~3 的时钟源为系统时钟 (Fsys)，可根据 SYSCLK\_SEL[2:0]选择为: 内部低频 RC 时钟 (Flrc)，内部高频 RC 时钟 (Fhrc)，内部低频时钟 (Flf) 和 PLL 时钟 (Fpll)。

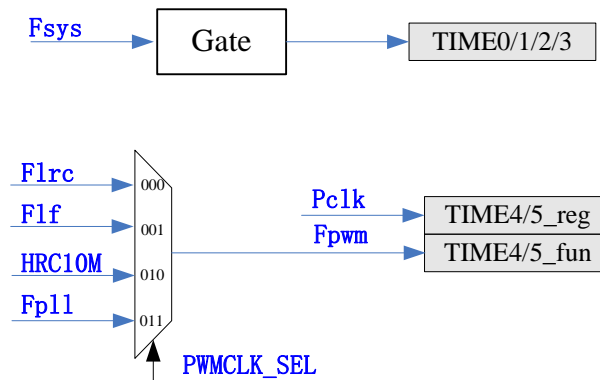
TMR4、5 的时钟源 Fpwm 独立于系统时钟 Fsys，可根据 PWMCLK\_SEL[2:0]选择为: 内部低频 RC 时钟 (Flrc)，内部高频 RC 时钟 (HRC10M)，内部低频时钟 (Flf) 和 PLL 时钟 (Fpll)，外部引脚输入时钟 (Fexclk)。

所有 TMR 单元相关寄存器的配置，都需要在 CLKCTRL1 寄存器中使能相应的 TMR 模块后才能配置。

TMR 主要包括以下功能:

- 1.周期定时功能
- 2.PWM 功能
- 3.捕获功能
- 4.事件计数功能
- 5.单次计数功能 (仅 TMR4/5 支持)
- 6.扩展触发功能 (仅 TMR4/5 支持)

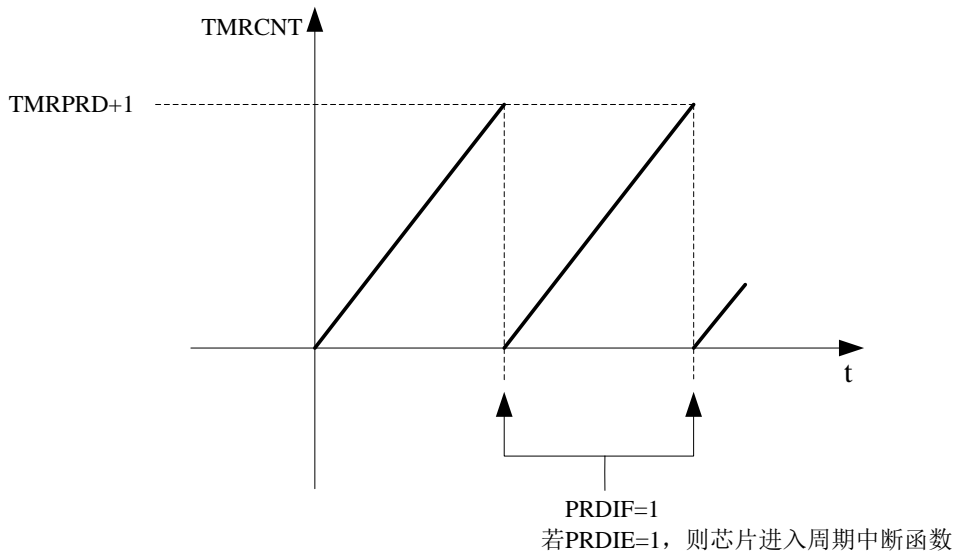
### 12.2 定时器(TMR)时钟选择



### 12.3 周期定时功能

通用 TMR 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 (Fsys) 通过 TMR 单元内的预分频器 (TMRDIV) 分频得到，当使能计数器 (CNTEN=1) 后，TMR 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 TMR 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 0，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：TMRCON，TMRCNT，TMRPRD。

## 12.4 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。

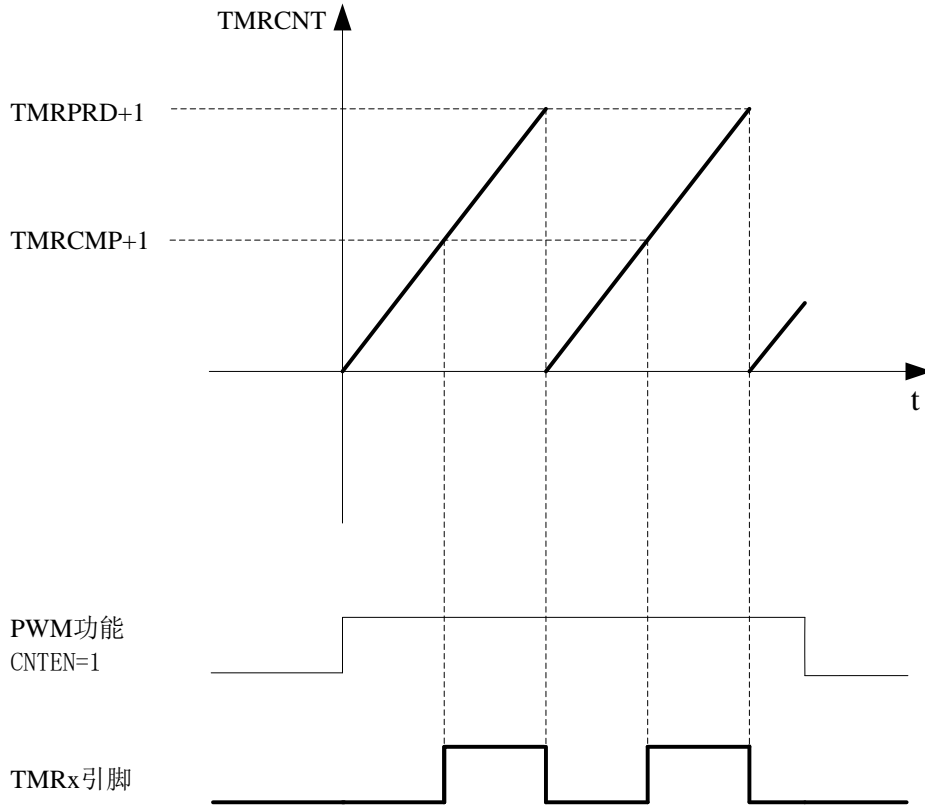
在 PWM 使用过程中 (CNTEN=1) 修改 PWM 输出极性，TMR0`3 立即生效，TMR4/5 在时钟同步后立即生效。建议用户在使用时不要修改 PWM 输出极性。

在 PWM 使用过程中 (CNTEN=1) 修改 TMRCMP 和 TMRPRD 时，TMR0`3 在下一个周期更新生效，TMR4/5 在时钟同步后的下一个周期更新生效。

PWM 计数方式分为向上计数、向下计数和中央计数方式。

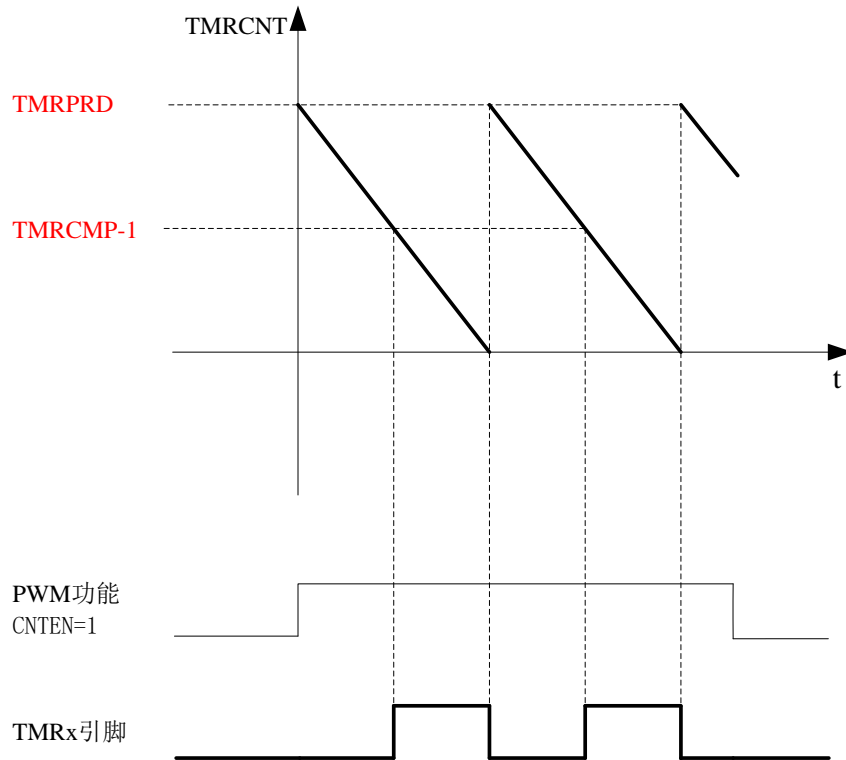
### 1.向上计数方式

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：



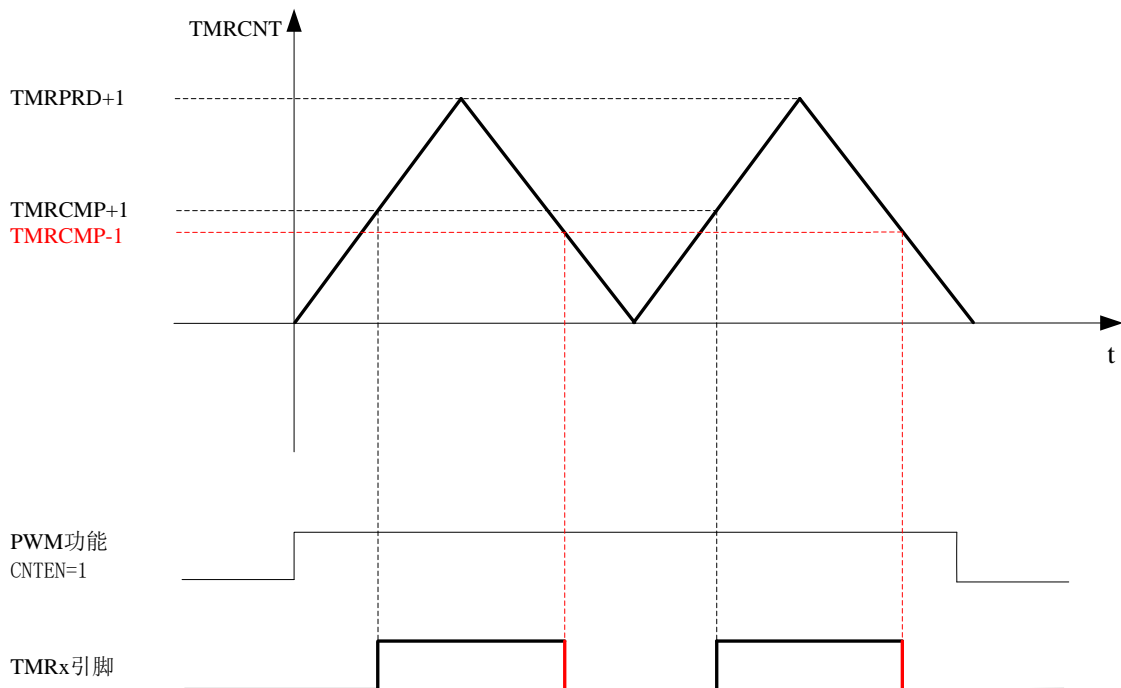
## 2. 向下计数方式

当使能计数器 (CNTEN=1) 之后, 计数器开始从 (TMRPRD) 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:



### 3.中央计数方式

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 19.6608 MHz 时钟，TMR 预分频寄存器（TMRDIV）默认值为 0（不分频），TMR0 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  $(\text{TMRCMP}+1) / (\text{TMRPRD}+1) = 30\%$ ，根据需要的周期值来确定 TMRPRD 寄存器的值，TMRPRD 所能配置的最大周期定时时间为：

$$(\text{TMRPRD}+1) / (19.6608 \text{ M} / (\text{TMRDIV} + 1)) = 0\text{xFFFF} / (19.6608 \text{ M} / (0+1)) \approx 3.33\text{ms}$$

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：TMRCON, TMRCNT, TMRCMP, TMRPRD。

## 12.5 捕获功能

在输入捕获模式下，假如设定上升沿检测，当 TMR0~TMR5 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。

产生捕获事件时，TMR4/5 捕获数据寄存器值 TMRCAP 会保持不变，直到 CAPIF 标志被清除。软件在读取 TMRCAP 值时，要先读 TMRCAP，再清除 CAPIF 标志位。

捕获检测可以选择上升沿，下降沿方式，以及双沿捕获（仅 TMR4/5 支持）。

功能主要相关寄存器：TMRCON, TMRCAP, TMRCNT, TMRPRD。

### 12.5.1 双沿捕获（仅 TMR4/5 支持）

实现双沿捕获模式，需要使能 TMRCON.CDMODE=1，配合 TMRCON.CCMODE 可以实现捕获真值表如下

CDMODE	CCMODE	
0	0	上升沿捕获
0	1	下降沿捕获
1	0	双沿捕获，上升沿开始
1	1	双沿捕获，下降沿开始

在双沿捕获模式下，若使能双沿捕获清零功能 TMRCON.CDCLEAR=1，则在每次有效捕获后，自动将计数清零 TMRCNT=0；

第一次触发也会将当前的 TMRCNT 捕获至 TMRCAP 寄存器，使用时需要软件处理判断。

在双沿捕获模式下，计数器的计数值 TMRCNT 在每次到达设定周期值 TMRPRD 或发生捕获事件时，自动清零；其真值表如下所示：

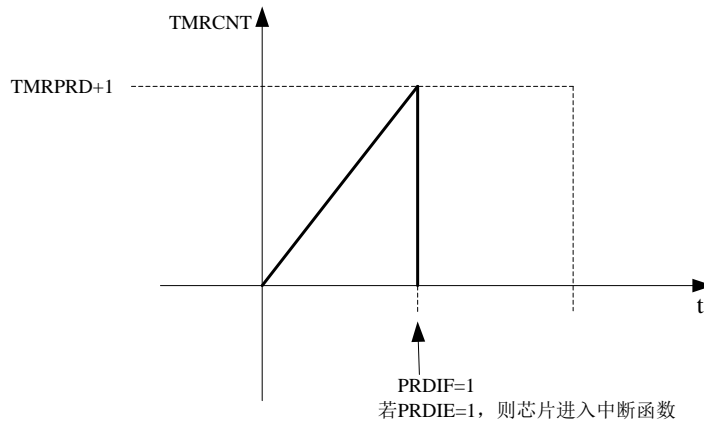
CDMODE	CCMODE	CDCLEAR	捕获事件	TMRCNT 清零
0	0	X	上升沿捕获	TMRCNT >= TMRPRD
0	1	X	下降沿捕获	TMRCNT >= TMRPRD
1	0	0	双沿捕获，上升沿开始	TMRCNT >= TMRPRD
1	0	1	双沿捕获，上升沿开始	TMRCNT >= TMRPRD or 捕获事件

1	1	0	双沿捕获, 下降沿开始	$\text{TMRcnt} \geq \text{TMRPRD}$
1	1	1	双沿捕获, 下降沿开始	$\text{TMRcnt} \geq \text{TMRPRD}$ or 捕获事件

## 12.6 单次计数功能（仅 TMR4/5 支持）

### 12.6.1 单次定时

单次定时功能是周期定时功能的一个特例，需要先使能单次计数功能控制位  $\text{TMR\_ONCE}=1$ ，再使能计数器（ $\text{CNTEN}=1$ ）之后，计数器开始从 0 计数，当计数寄存器（ $\text{TMRcnt}$ ）的值等于设定的周期寄存器（ $\text{TMRPRD}+1$ ）时，置位周期定时中断标志（ $\text{PRDIF}=1$ ），如果使能定时中断（ $\text{PRDIE}=1$ ），则会触发 TMR 中断，进入相应的中断服务程序；同时计数器自动重载值 0，并且计数器停止计数、 $\text{CNTEN}$  清 0；在下次将计数器使能  $\text{CNTEN}$  置 1 之前，计数器将保持停止。



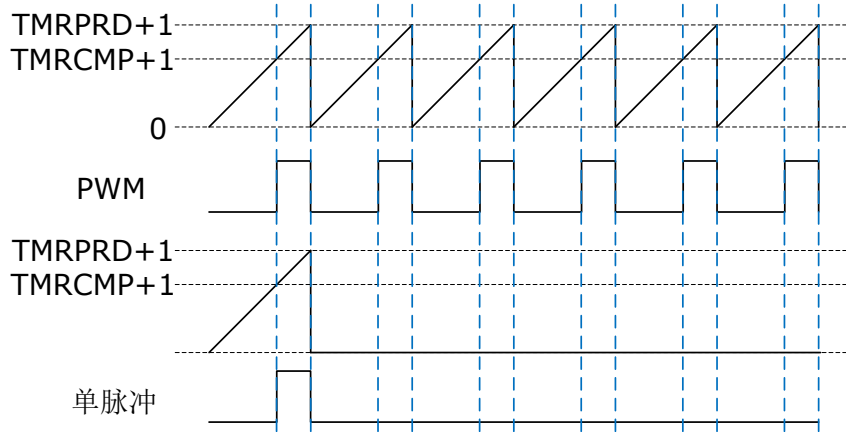
### 12.6.2 单脉冲输出

单脉冲功能是 PWM 功能的一个特例，需要先使能单次计数功能控制位  $\text{TMRCFG.TMR\_ONCE}=1$ ，再使能计数器  $\text{TMRCON.CNTEN}=1$ ；并且只有当  $\text{TMRcmp}$  和  $\text{TMRPRD}$  初始值不同时，才能正确产生一个脉冲；启动前（TMR 等待启动时），必须保证  $0 < \text{TMRcmp} < \text{TMRPRD}$ 。

**注：**单脉冲输出只在向上或向下计数模式下有效，在中央计数模式时无效。

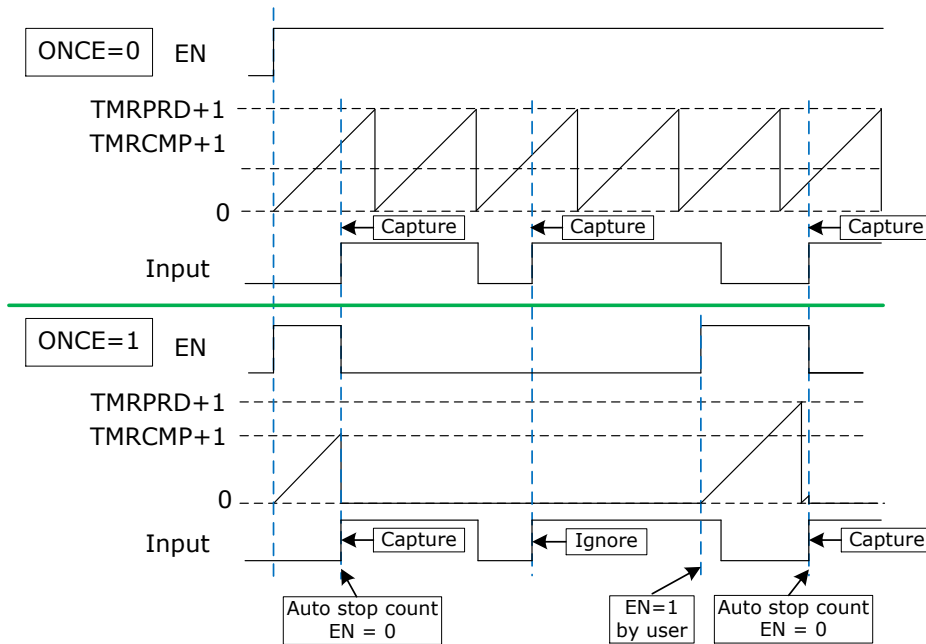
例如，在向上计数模式下，计数器开始从 0 计数，当计数寄存器（ $\text{TMRcnt}$ ）的值等于设定的比较寄存器（ $\text{TMRcmp}+1$ ）时，PWM 输出管脚发生电平翻转，同时置位比较中断标志（ $\text{CMPIF}=1$ ）。计数器继续向上计数，当计数寄存器（ $\text{TMRcnt}$ ）的值等于设定的周期寄存器（ $\text{TMRPRD}+1$ ）时，PWM 输出管脚再次发生电平翻转，同时置位周期定时中断标志（ $\text{PRDIF}=1$ ），计数器自动重载值 0，并且计数器停止计数、 $\text{CNTEN}$  清 0；在下次将计数器使能  $\text{CNTEN}$  置 1 之前，计数器将保持停止。

向上计数时 PWM 输出和单脉冲输出对比如下图所示：



### 12.6.3 单次捕获

捕获模式同时开启单次计数  $\text{TMRCFG.TMR\_ONCE}=1$ ，则捕获到一个有效触发沿到来时，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。单次捕获完成的同时会停止计数，使能关闭并保持当前的捕捉数据至捕获数据寄存器 TMRCAP，直到下一次重新使能后，计数器重新计数，并再次进行边沿捕捉。



双沿捕获和单次模式联合控制情况如下表所示

双沿捕获， $\text{TMRCFG.TMR\_ONCE}=0$

	CDCLEAR	TMRCNT	TMRCAP	CAPIF
CNTEN	X	开始计数	0	0
首次有效事件发生	0	不清零	更新	置1
	1	清零	更新	置1
非首次事件发生	0	不清零	更新	置1
	1	清零	更新	置1

双沿捕获, TMRCFG.TMR\_ONCE=1

	CDCLEAR	TMRCNT	TMRCAP	CAPIF	CNTEN
CNTEN	X	开始计数	0	0	X
首次有效事件发生	0	不清零	更新	置 1	1
	1	清零	更新	置 1	1
第二次有效事件发生	0	不清零	更新	置 1	0
	1	清零	更新	置 1	0
第三次及之后事件发生	0	保持	保持	保持	0
	1	保持	保持	保持	0

## 12.7 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 TMR0~5 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 0, 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, 若配置了 TMRCMP>TMRPRD, 当计数寄存器 (TMRCNT) 的值等于设定的周期寄存器 (TMRPRD) 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), 如果使能了比较中断(CMPIE=1)会进入比较中断服务程序。

上述事件计数应用 (TMRCMP>TMRPRD) 时, 当 TMRCNT=TMRPRD 时, PRDIF 标志置起; 但如果此时不再发生新的触发事件 (即维持 TMRCNT=TMRPRD 状态), PRDIF 会不断置起, 如果使能中断则反复进入中断, 直到新的事件到来, 使 TMRCNT>TMRPD 后 PRDIF 标志才能被清除。

建议使用方式为:

(1) 使用事件计数功能时, 配置 TMRCMP<TMRPRD

(2) 若特殊应用需要配置 TMRCMP>TMRPRD 时, 则当 TMRCNT=TMRPRD, PRDIF 标志置起产生中断时, 进入中断服务程序处理时, 增加清零 TMRCNT 后清除 PRDIF 标志的操作。

在事件计数模式下, 若需要修改 TMRCMP、TMRPRD, 建议软件操作应先关闭计数器(CNTEN=0), 修改完后再开启计数器(CNTEN=1)。

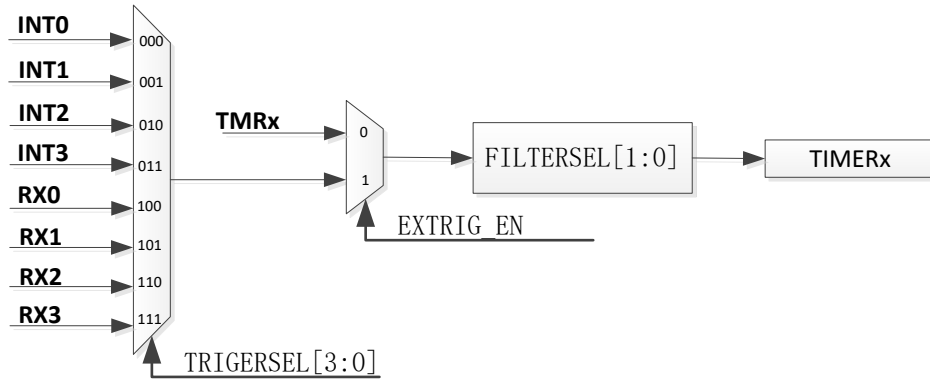
事件计数检测可以选择上升沿, 下降沿方式, 该功能可用于 MCU 发脉冲。

功能主要相关寄存器: TMRCON, TMRCNT, TMRCMP, TMRPRD。

## 12.8 扩展触发功能 (仅 TMR4/5 支持)

### 12.8.1 TMR4/5 事件计数或捕获输入扩展及信号数字滤波

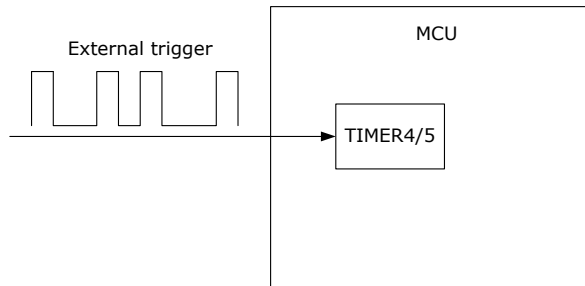
当 TMR4/5 选择事件计数或捕获功能时, 可选择 INT0-3 与 RX0-3 作为其输入信号源, 例如扩展输入信号源使能打开 (EXTRIG\_EN =1), 扩展输入信号源选择 RX0, 假如 CCMODE 设定下降沿检测且 MODE 选择事件计数功能, 当 TMR4/5 检测到一次下降沿, 计数寄存器 (TMRCNT) 的当前值加 1。



TMR4/5 具有输入信号源数字滤波功能，其滤波器共四档可选择，分别为 0、2、5、11 个 Fpwm 时钟个数。

### 12.8.2 TMR4/5 外部异步脉冲计数功能

当 TMR4/5 的时钟选择配置为外部输入时钟 Fexclk 时，TMR4/5 可以不需要内部时钟，使用外部输入时钟进行计数（上升沿或下降沿触发），时钟信号从外部 TMR4/5 管脚输入（引脚需配置为 TMR 复用功能）；如下图所示：



TMR4/5 的异步脉冲计数功能使用，需配置以下寄存器控制位：

- TMR4/5 为周期定时模式 (MODE[1:0]=11b)
- 选择事件计数触发沿为上升沿或下降沿 (CCMODE=0 或 1)
- 选择 TMR4/5 的时钟为外部时钟 (PWMCLK\_SEL[2:0]=110b 或 111b, Fexclk)

当 TMR4/5 工作在上图所示的独立脉冲计数模式时，周期中断可以正常生效（需使能相应的周期中断），产生中断可以唤醒芯片；唤醒后，可读出当前累计的计数 TMRCNT 值（周期中断产生后，TMRCNT 自动清零从 0 重新开始计数，此时 **实际 TMRCNT 值=读出 CNT 值+周期**）。

**注：**

该模式下，使能后的前 7 个脉冲，TMR4/5 会利用外部脉冲作为时钟进行内部配置，在第 8 个脉冲开始计数。在该模式下，建议外部输入脉冲频率不大于 1MHz。

### 12.8.3 TMR4/5 外部触发使能功能

TMR4/5 的触发模式可以在所选的触发事件产生时启动 TMR，通过配置外部使能控制位 TMRTRIG\_EN 使能该功能；使能 TMRTRIG\_EN=1 后，计数器会自动停止计数并清零 TMRCNT，TMR 模块将进入等待触发状态。触发事件产生后，TMR 模块根据用户预先配置的功能运行（定时、PWM、脉冲输出等）。该功能可以在未唤醒 CPU 情况下，由外部触发信号使能运行 TMR 模块。

外部触发信号源可以通过 TMR 触发信号选择寄存器 TMRTRIGSEL[3:0]配置选择;可以选择 INT0、INT3、RX0、RX3、RTCTMR2\_PRD (RTCTMR2 的定时中断)、TMR5\_CMP (TMR5 的比较中断)、TOUT; 触发方式可配置, 可选择为: 上升沿触发、下降沿触发或上升/下降沿触发。

注: 在使用以下信号作为外部使能触发源时, 如 RTCTMR2\_PRD (RTCTMR2 的定时中断)、TMR5\_CMP (TMR5 的比较中断), 由于内部中断信号是从0到1(上升沿), 此时触发上升沿/下降沿控制位 TRIGMODE[1:0]配置只有上升沿有效, 其余配置无效, 即此时只有信号上升沿会触发 TMR 使能。

## 12.9 中断功能

### 12.9.1 周期定时中断

当计数寄存器 (TMRCNT) 的值等于设定的周期寄存器 TMRPRD+1 (向上计数) 或 TMRPRD-1 (向下计数) 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。

### 12.9.2 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 (CMPIE=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。

### 12.9.3 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 (向上计数) 或 TMRCMP-1 (向下计数) 时, 如使能了 PWM 比较中断 (CMPIE=1), 则发生比较中断。

### 12.9.4 事件计数中断

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。

## 12.10 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);				
0x40019000(TMR4);				
0x4001A000(TMR5);				
偏移地址	名称	读写方式	复位值	功能描述
0x00	TMRCON	R/W	0x0000	定时器控制寄存器
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	计数寄存器
0x14	TMRCMP	R/W	0x0000	比较寄存器
0x18	TMRIE	R/W	0x0000	定时器中断使能寄存器
0x1C	TMRIF	R/W	0x0000	定时器中断标志寄存器
0x28	TMRTRIGERCFG	R/W	0x0000	定时器扩展输入配置寄存器
0x2C	TMRCFG	R/W	0X0000	定时器功能配置寄存器
0x30	TMREXTRIGCFG	R/W	0x0000	定时器触发信号选择寄存器

## 12.11 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、0x40019000、0x4001A000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	CDCLE AR	CDMOD E	PWMCLK_SEL[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	RESERV ED	PWMHL	PWMC[1:0]		CCMOD E	MODE[1:0]		CNTEN
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
---	------

CDCLEAR	<p><b>双沿捕获模式清零控制位：（仅 TMR4/5 支持）</b>                  0：双沿捕获后，计数器不清零（default）                  1：双沿捕获后，计数器清零                  注：只对双沿捕获生效，CDMODE=0 时无效；</p>																								
CDMODE	<p><b>双沿捕获模式控制位：（仅 TMR4/5 支持）</b>                  0：单边沿捕获（default）                  1：双边沿捕获</p> <table border="1" data-bbox="424 488 1034 701"> <thead> <tr> <th>CDMODE</th> <th>CCMODE</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>上升沿捕获</td> </tr> <tr> <td>0</td> <td>1</td> <td>下降沿捕获</td> </tr> <tr> <td>1</td> <td>0</td> <td>双沿捕获，上升沿开始</td> </tr> <tr> <td>1</td> <td>1</td> <td>双沿捕获，下降沿开始</td> </tr> </tbody> </table>	CDMODE	CCMODE		0	0	上升沿捕获	0	1	下降沿捕获	1	0	双沿捕获，上升沿开始	1	1	双沿捕获，下降沿开始									
CDMODE	CCMODE																								
0	0	上升沿捕获																							
0	1	下降沿捕获																							
1	0	双沿捕获，上升沿开始																							
1	1	双沿捕获，下降沿开始																							
PWMCLK_SEL[2:0]	<p><b>TMR4/5 时钟选择控制位：</b></p> <table border="1" data-bbox="424 745 1353 1003"> <thead> <tr> <th colspan="3">PWMCLK_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（default）</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>HRC10M</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Fpll</td> </tr> <tr> <td>1</td> <td>X</td> <td>X</td> <td>Fexclk</td> </tr> </tbody> </table> <p>该时钟选择只对 TMR4/5 有用。</p>	PWMCLK_SEL[2:0]			系统时钟选择Fsys	0	0	0	Flrc（default）	0	0	1	Fosc	0	1	0	HRC10M	0	1	1	Fpll	1	X	X	Fexclk
PWMCLK_SEL[2:0]			系统时钟选择Fsys																						
0	0	0	Flrc（default）																						
0	0	1	Fosc																						
0	1	0	HRC10M																						
0	1	1	Fpll																						
1	X	X	Fexclk																						
RESERVED	内部保留位，应用中固定为 0																								
PWMHL	<p><b>PWM 初始电平选择：</b>（每次写入极性与原有极性比较，不同则修改，相同不处理）                  0：高电平                  1：低电平</p>																								
PWMC[1:0]	<p><b>PWM 工作模式选择（PWM 计数方式）：</b>                  00：向上计数                  01：向下计数                  1x：中央对齐</p>																								
CCMODE	<p><b>捕获/事件计数电平沿选择（当定时器配置为捕获/事件计数功能）：</b>                  0：上升沿                  1：下降沿</p>																								
MODE[1:0]	<p><b>TMR 功能选择：</b>                  00：事件计数功能                  01：PWM 功能                  10：捕获功能                  11：周期定时功能                  注：需将对应 GPIO 配置为 TMRx 功能</p>																								
CNTEN	<p><b>计数器使能：</b>                  0：关闭                  1：使能</p>																								

## TMRDIV (预分频寄存器)

<b>TMRDIV</b> (预分频寄存器)			基地址: 0x40001000--0x40004000、0x40019000、0x4001A000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRDIV[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRDIV[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

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

## TMRPRD (周期寄存器)

<b>TMRPRD</b> (周期寄存器)			基地址: 0x40001000--0x40004000、0x40019000、0x4001A000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRPRD[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRPRD[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMRPRD[15:0]	该寄存器是一个 16 位的周期寄存器 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。  $\text{Period} = \frac{(TMRPRD[15:0] + 1)}{F/(TMRDIV[15:0] + 1)}$

## TMRCAP（捕获数据寄存器）

<b>TMRCAP</b> (捕获数据寄存器)			基地址: 0x40001000--0x40004000、0x40019000、0x4001A000 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRCAP[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRCAP[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMRCAP[15:0]	当发生捕获事件时，当前计数寄存器（TMRCNT）的值被存到该寄存器里 注：对于 TMR4/5，在双沿捕获时，CAPIF 会将捕获数据寄存器 TMRCAP 和双沿捕获状态位 CCSTA 一起锁定，直至 CAPIF 被软件清零，才能触发下一次捕获状态和数据更新

## TMRCNT（计数寄存器）

<b>TMRCNT</b> (计数寄存器)			基地址: 0x40001000--0x40004000、0x40019000、0x4001A000 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRCNT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRCNT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMRCNT[15:0]	计数器当前的计数值

## TMRCMP（比较寄存器）

<b>TMRCMP</b>	基地址: 0x40001000--0x40004000、0x40019000、0x4001A000
---------------	---

(比较寄存器)			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)TMR 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)TMR 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)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、0x40019000、0x4001A000 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	TRIGIE	ACIE	CMPIE	CAPIE	PRDIE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TRIGIE	外部触发中断使能: (仅 TMR4/5 支持) 0: 关闭 1: 使能
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	周期定时中断使能 0: 关闭 1: 使能

## TMRIF（定时器中断标志寄存器）

<b>TMRIF</b> (定时器中断标志寄存器)		基地址: 0x40001000--0x40004000、0x40019000、0x4001A000 偏移地址: 1CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	CCSTA
Write:								X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	TRIGIF	ACIF	CMPIF	CAPIF	PRDIF
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CCSTA	<b>捕获触发状态标志: (仅 TMR4/5 支持)</b> 0: 当前捕获数据为下降沿捕获 1: 当前捕获数据为上升沿捕获 注: 对于 TMR4/5, 在双沿捕获时, CAPIF 会将该状态位和捕获数据寄存器 TMRCAP 一起锁定, 直至 CAPIF 被软件清零, 才能触发下一次捕获状态和数据更新
TRIGIF	<b>外部触发中断标志: (仅 TMR4/5 支持)</b> 0: 未产生中断 1: 产生中断
ACIF	<b>事件计数中断标志</b> 0: 未产生中断 1: 产生中断 (写 0 清 0)
CMPIF	<b>比较中断标志</b> 0: 未产生中断 1: 产生中断 (写 0 清 0)
CAPIF	<b>捕获中断标志</b> 0: 未产生中断 1: 产生中断 (写 0 清 0)

PRDIF	周期定时中断标志 0: 未产生中断 1: 产生中断 (写 0 清 0)
-------	---

### TMRTRIGERCFG (定时器触发源配置寄存器)

<b>TMRTRIGERCFG</b> (定时器触发源配置寄存器)			<b>基地址: 0x40019000、0x4001A000</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>								TMR_M
<b>Write:</b>	X	X	X	X	X	X	Reserved	ODE_SE L
<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>	FILTERSEL[1:0]		EXTRIG	X	X	TRIGERSEL[2:0]		
<b>Write:</b>			_EN					
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述																				
TMR_MODE_SEL	<b>TMR4/5 异步读取 TMRCNT 模式控制位</b> 0: 不使用同步逻辑进行读取 (default) 1: 使用同步逻辑进行读取 注: 建议用户使用时, 将该控制位配置为 1。																				
FILTERSEL[1:0]	输入信号源滤波选择位: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="2">FILTERSEL[1:0]</th> <th>滤波时钟个数 (Fpwm)</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0(default)</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">5</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">11</td> </tr> </tbody> </table> 注: 滤波时钟源为 Fpwm。该输入信号源滤波选择位只对 TMR4/5 有用。	FILTERSEL[1:0]		滤波时钟个数 (Fpwm)	0	0	0(default)	0	1	2	1	0	5	1	1	11					
FILTERSEL[1:0]		滤波时钟个数 (Fpwm)																			
0	0	0(default)																			
0	1	2																			
1	0	5																			
1	1	11																			
EXTRIG_EN	扩展输入源使能控制位 0: 关闭 (default) 1: 使能 注: 只有该位使能时, TRIGERSEL 选择扩展输入源才生效。该扩展输入源使能控制位只对 TMR4/5 有用。																				
TRIGERSEL[2:0]	定时器 4、5 扩展输入源选择控制位: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="3">TRIGERSEL[2:0]</th> <th>输入源</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">INT0 (default)</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">INT1</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">INT2</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">INT3</td> </tr> </tbody> </table>	TRIGERSEL[2:0]			输入源	0	0	0	INT0 (default)	0	0	1	INT1	0	1	0	INT2	0	1	1	INT3
TRIGERSEL[2:0]			输入源																		
0	0	0	INT0 (default)																		
0	0	1	INT1																		
0	1	0	INT2																		
0	1	1	INT3																		

	1	0	0	RX0
	1	0	1	RX1
	1	1	0	RX2
	1	1	1	RX3

注：该扩展触发源选择只对 TMR4/5 有用。

注：该寄存器只对 TMR4/5 生效。

### TMRCFG（定时器功能配置寄存器）

<b>TMRCFG</b> (定时器功能配置寄存器)			基地址： 0x40019000、0x4001A000 偏移地址： 2CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	TMR_ ONCE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMR_ONCE	<b>TMR 单次计数功能控制位：</b> =0：不使能（default） =1：使能

注：该寄存器只对 TMR4/5 生效。

### TMREXTRIGCFG（定时器触发信号选择寄存器）

<b>TMREXTRIGCFG</b> (定时器触发信号选择寄存器)			基地址： 0x40019000、0x4001A000 偏移地址： 30H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRTRI	X	X	X	X	X	TRIGMODE[1: 0]	
Write:	G_EN							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	TMRTRIGSEL[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述																																																		
TMRTRIG_EN	<p><b>外部触发使能控制位：</b>                      =0：不使能（default）                      =1：使能</p> <p>注：当 TMRTRIG_EN=0 不使能时，CNTEN=1，计数器自动计数；当 TMRTRIG_EN=1 使能时，CNTEN=1，计数器不自动计数，等待外部触发源到来触发计数。使能 TMRTRIG_EN=1 后，计数器会自动停止计数并清零 TMRCNT</p>																																																		
TRIGMODE[1: 0]	<p><b>触发电平边沿选择：</b>                      00：上升沿                      01：下降沿                      1x：上升/下降沿</p>																																																		
TMRTRIGSEL[3:0]	<p><b>定时器触发信号源选择控制位：</b></p> <table border="1"> <thead> <tr> <th colspan="4">TMRTRIGSEL[3:0]</th> <th>触发源</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>INT0（default）</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>INT3</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>RX0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>RX3</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>INT0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>RTCTMR2_PRD（RTCTMR2 的定时中断）</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>INT0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>TMR5_CMP（TMR5 的比较中断）</td> </tr> <tr> <td>1</td> <td>X</td> <td>X</td> <td>X</td> <td>TOUT</td> </tr> </tbody> </table> <p>注：只有在 TMRTRIG_EN 位使能，该寄存器配置才起作用。                      注：使用以下信号作为外部使能触发源时，只有信号上升沿会触发 TMR 使能：RTCTMR2_PRD（RTCTMR2 的定时中断）、TMR5_CMP（TMR5 的比较中断）</p> <p>注：该寄存器只对 TMR4/5 生效。</p>	TMRTRIGSEL[3:0]				触发源	0	0	0	0	INT0（default）	0	0	0	1	INT3	0	0	1	0	RX0	0	0	1	1	RX3	0	1	0	0	INT0	0	1	0	1	RTCTMR2_PRD（RTCTMR2 的定时中断）	0	1	1	0	INT0	0	1	1	1	TMR5_CMP（TMR5 的比较中断）	1	X	X	X	TOUT
TMRTRIGSEL[3:0]				触发源																																															
0	0	0	0	INT0（default）																																															
0	0	0	1	INT3																																															
0	0	1	0	RX0																																															
0	0	1	1	RX3																																															
0	1	0	0	INT0																																															
0	1	0	1	RTCTMR2_PRD（RTCTMR2 的定时中断）																																															
0	1	1	0	INT0																																															
0	1	1	1	TMR5_CMP（TMR5 的比较中断）																																															
1	X	X	X	TOUT																																															

## 13 SPI 模块

### 13.1 概述

SPI0 模块的芯片引脚为 PC.7/SPI0\_CS、PC.6/SPI0\_CLK、PC.4/SPI0\_MOSI/RX5 和 PC.5/SPI0\_MISO/TX5。

SPI1 模块的芯片引脚为 PG.6/SPI1\_CS、PG.7/SPI1\_CLK、PG.8/SPI1\_MISO/TX1 和 PG.9/SPI1\_MOSI/RX1。

SPI2 模块的芯片引脚为 PA.0/SPI2\_CS/TMR1、PA.1/SPI2\_CLK/TMR4、PA.2/SPI2\_MISO/RX3 和 PA.3/SPI2\_MOSI/TX3。

SPI3 模块的芯片引脚为 PD.8/SPI3\_CS、PD.9/SPI3\_CLK、PD.10/SPI3\_MISO、PD.11/SPI3\_MOSI。

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。

SPI 模块包含下列特征：

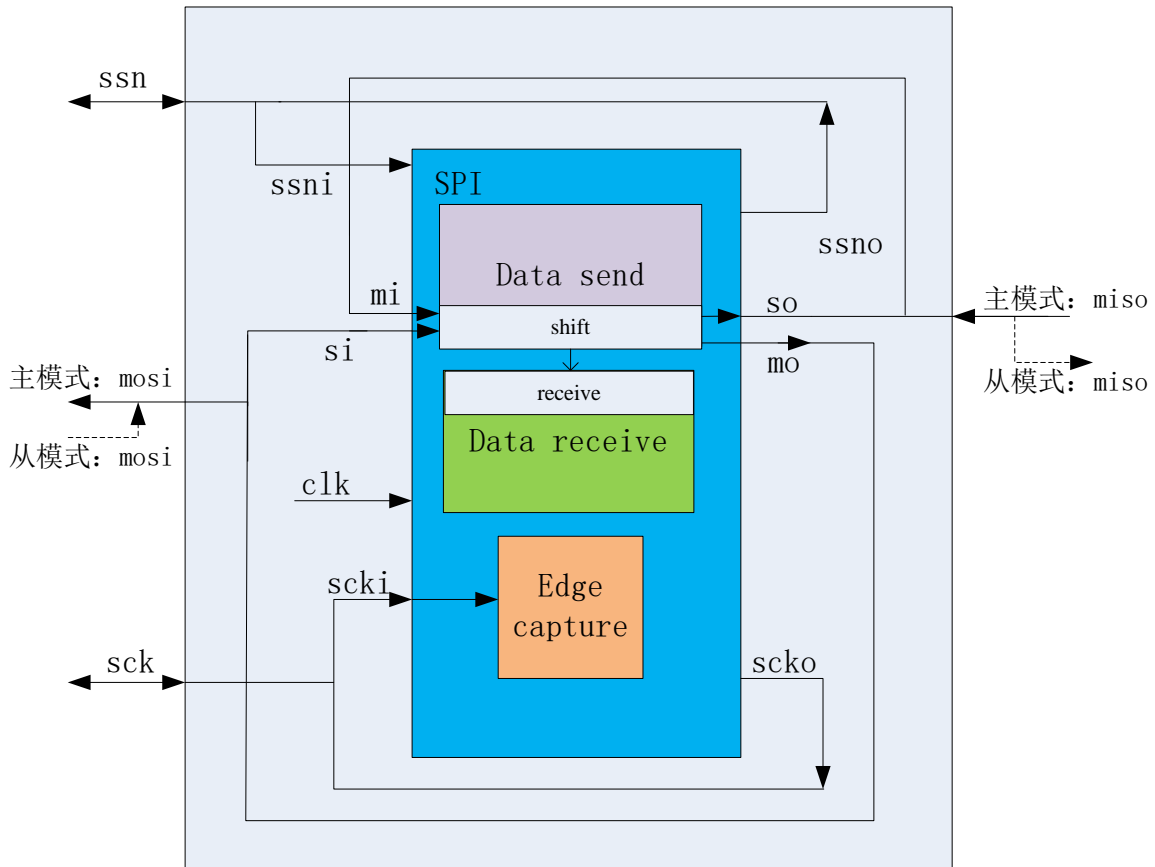
- 全双工模式
- 三线同步传输
- 主机和从机模式
- 7 种主机波特率
- 从机时钟最高至  $F_{sys}/2$ （需使能从机快速模式）
- 极性和相位可编程的串行时钟
- 写冲突处理机制
- 8 位数据传输，可配置 MSB 在前或 LSB 在前
- 8 位从机选择接口，控制外部从机
- 与主机 CPU 的专用功能寄存器接口
- 无二义端口，标准的 SPI
- SPI0/SPI2 具有独立的 TX、RXFIFO（TXFIFO、RXFIFO 分别 32\*8bit）

### 13.2 详细功能说明

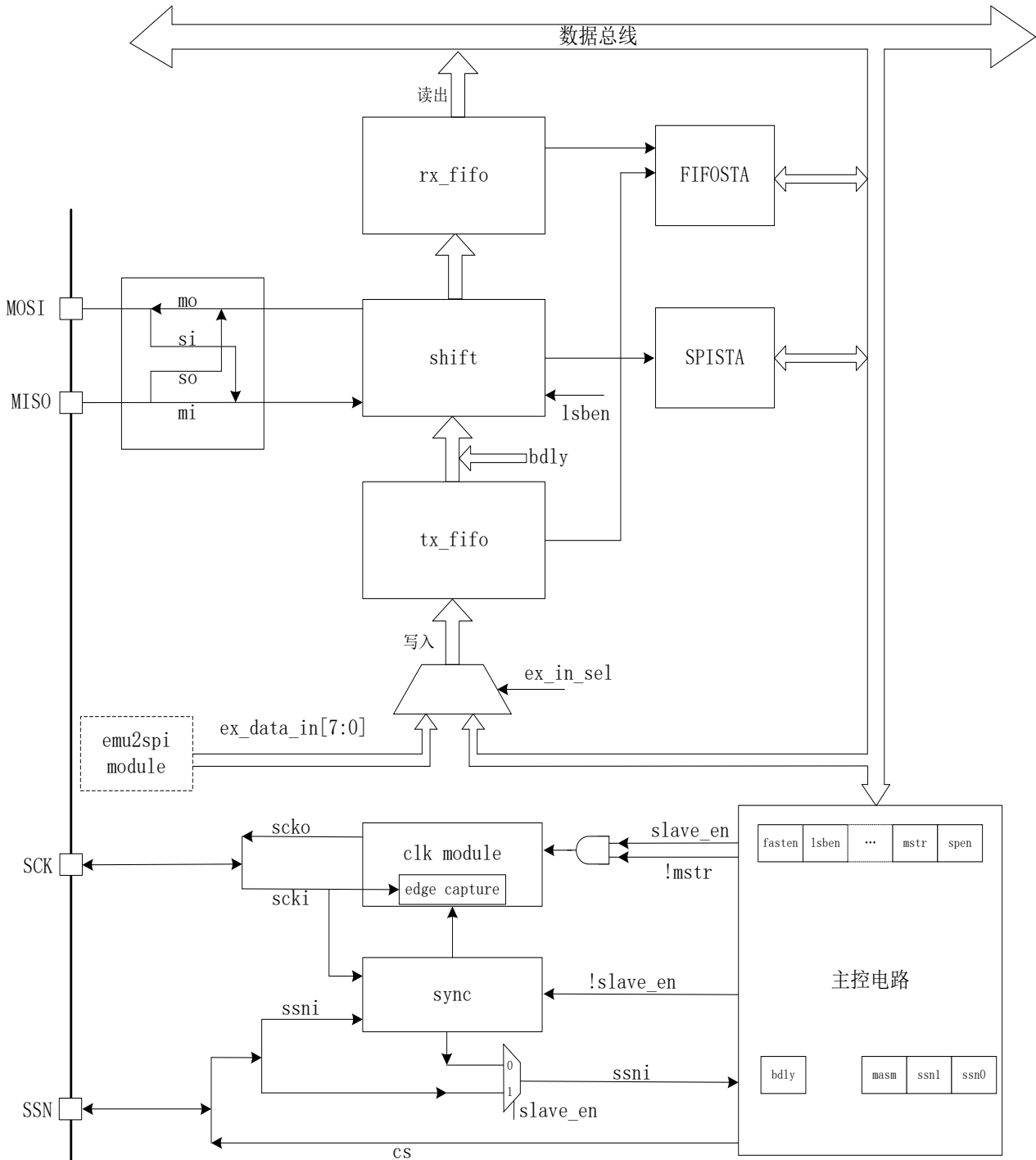
串行外设接口(SPI)允许芯片与其他设备以半/全双工、同步、串行方式通信。此接口可以被配置成主模式，并为从设备提供通信时钟(SCK)。

### 13.2.1 SPI 模块框图

#### 13.2.1.1 普通 SPI (SPI1/SPI3)



13.2.1.2 带 FIFO 的 SPI (SPI0/SPI2)



13.2.2 SPI 接口传输格式

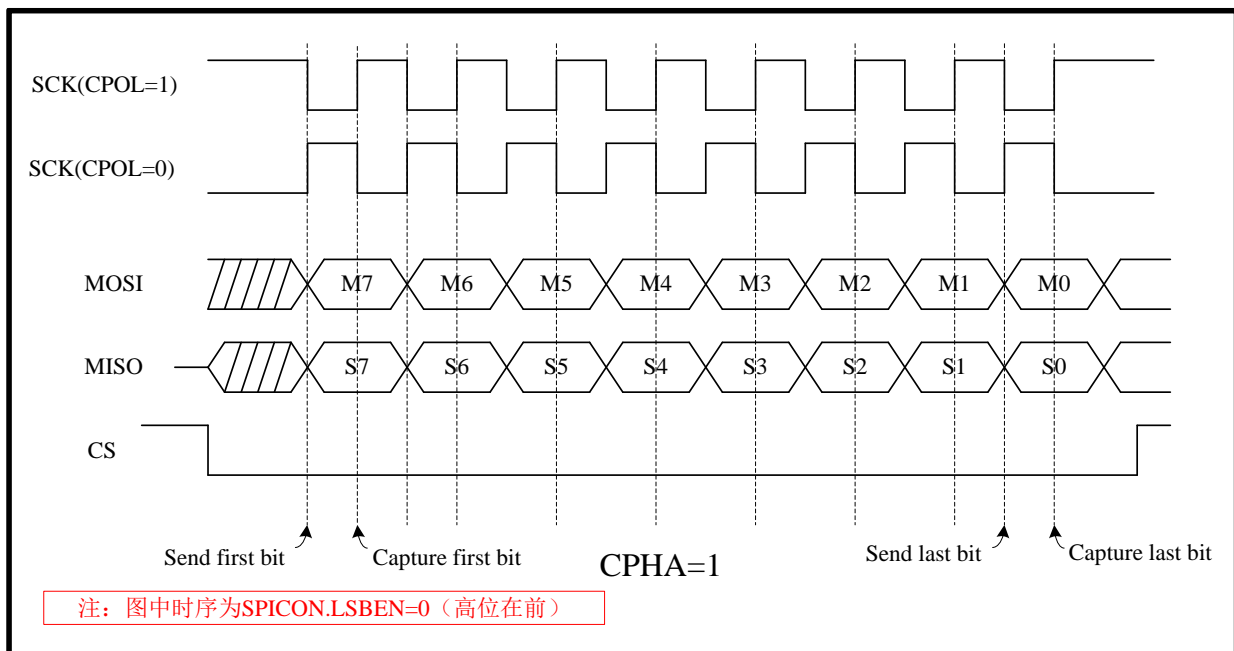
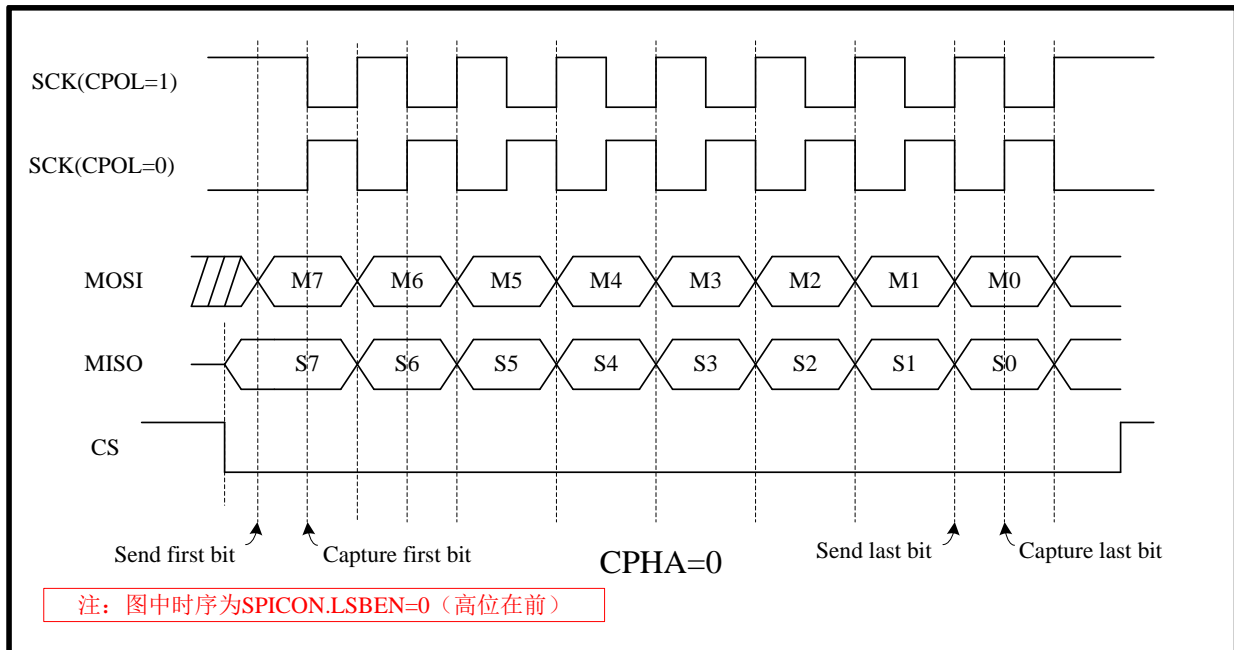
根据 SPI 模块的设置，数据的每一位在主时钟（SCK）的上升沿（CPOL=0）或下降沿（CPOL=1）发送；在主时钟（SCK）的下降沿（CPOL=1）或上升沿（CPOL=0）进行接收采样。这适用于主模式或从模式的传

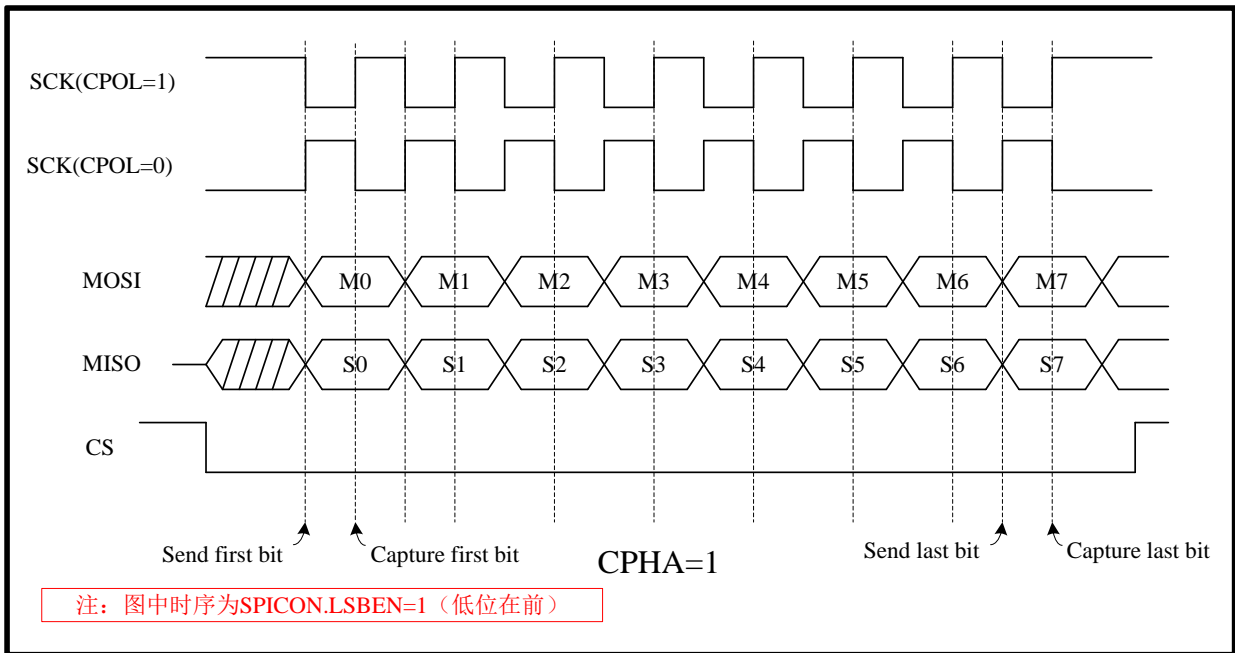
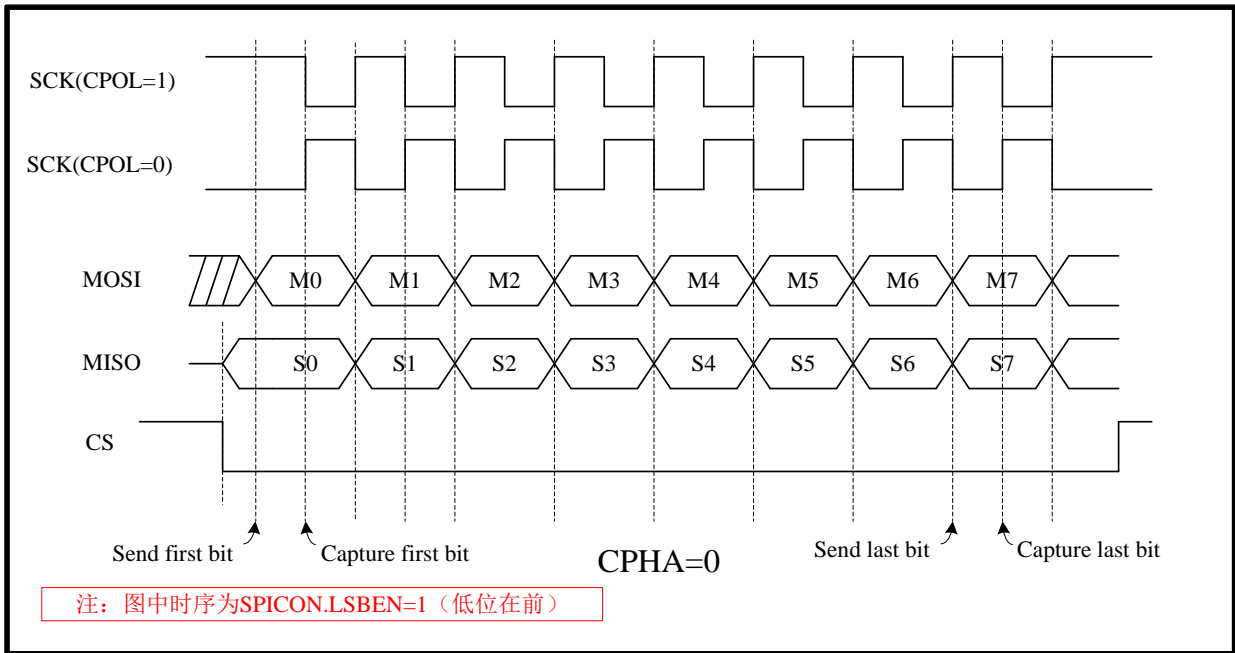
输器/接收器，时钟 SCK 在传输过程中由 SPI 主机提供。

若 CPHA=0，数据的第一个 bit 位将在 SCK 的第一个动态沿进行接收采样。若 CPHA=1，数据的第一个 bit 位将在 SCK 的第二个动态沿进行接收采样。

### 13.2.3 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 SPIDAT 中写入数据。如果向 SPIDAT 的写入动作完成，传输就开始。传输时序示意图如下所示：

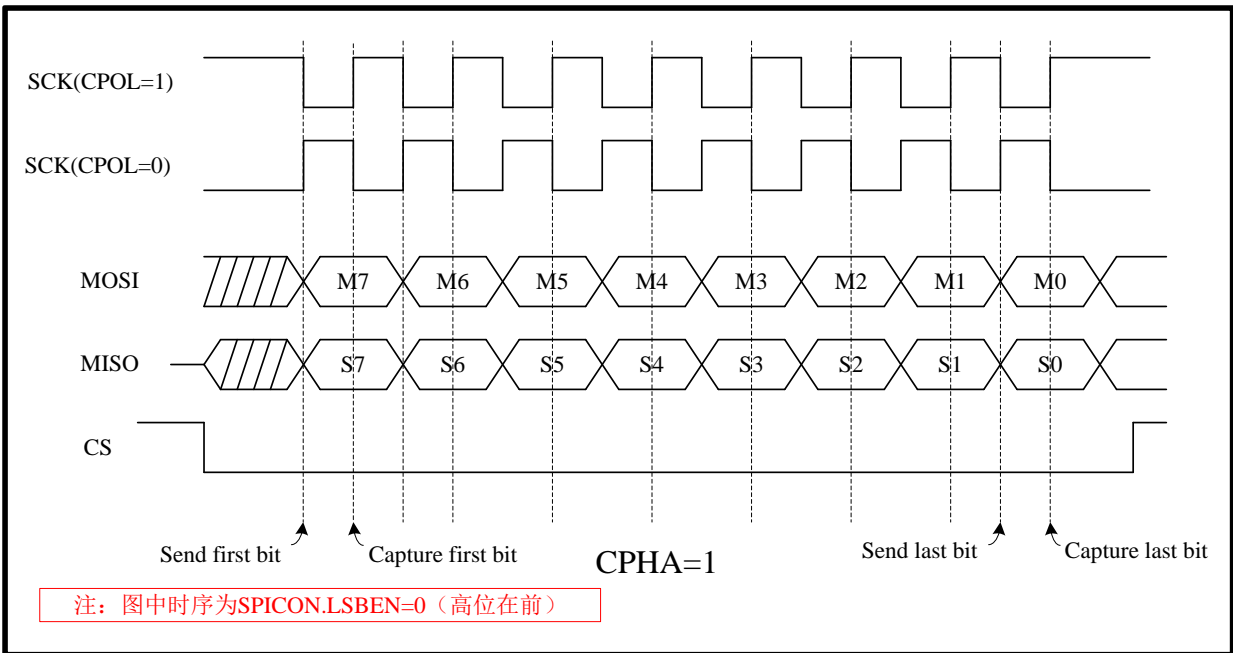
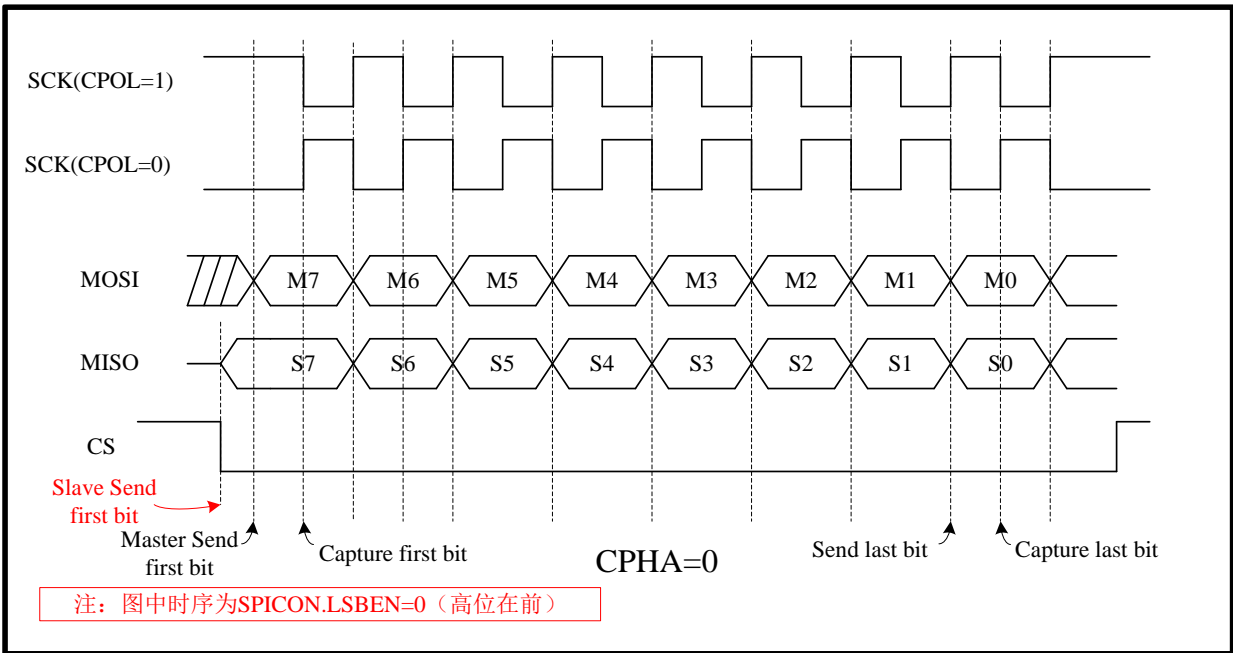


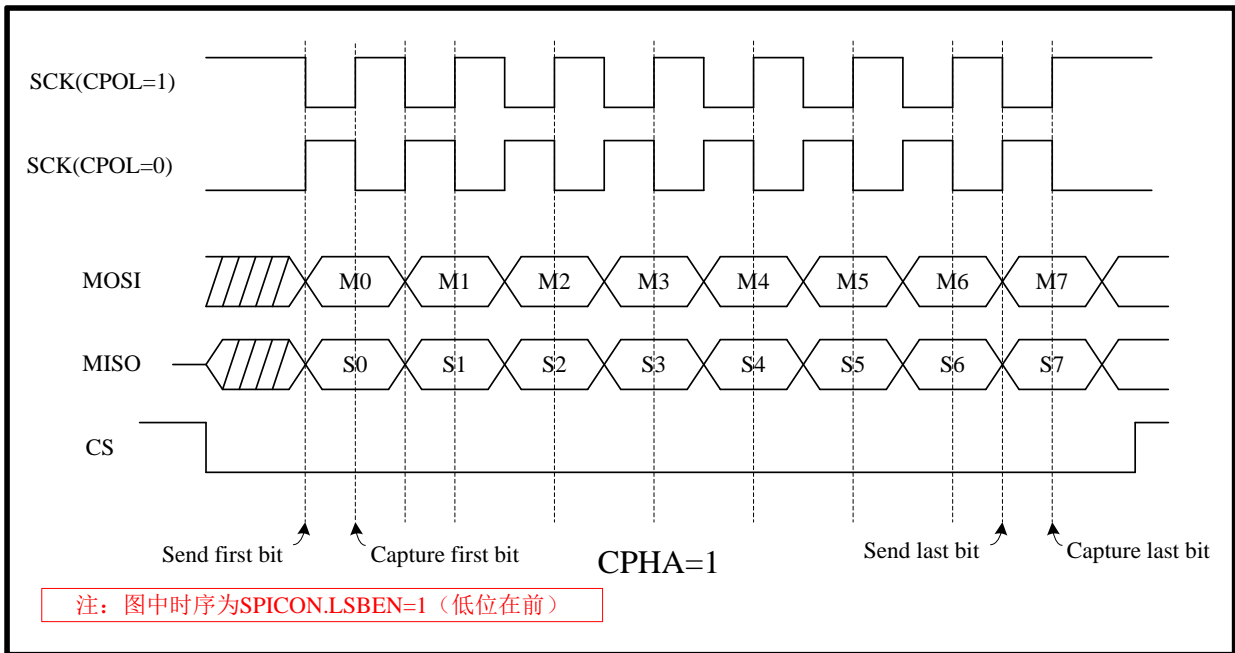
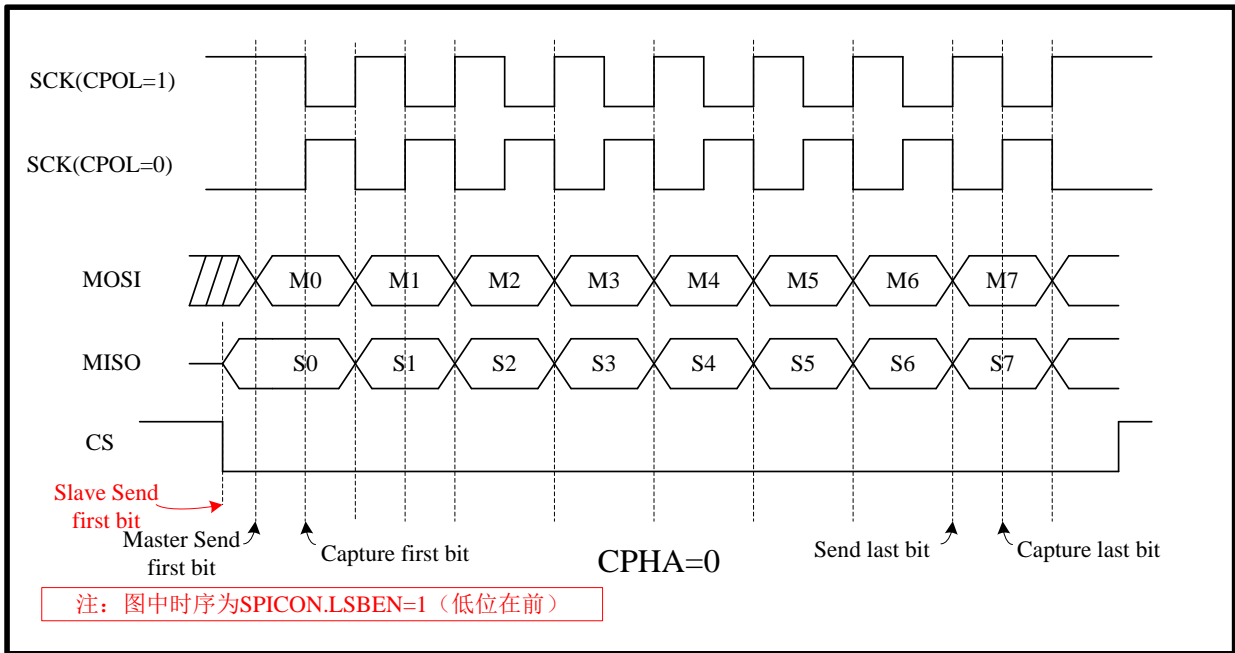


注：根据 SPICON 寄存器中的 LSBEN 位，输出数据位时可以 MSB 在先也可以 LSB 在先。

### 13.2.4 从机模式传输格式

通过将寄存器 SPICON.MSTR 写 0，可将 SPI 配置为从机模式。从机传输时序图如下所示：





**注：根据 SPICON 寄存器中的 LSBEN 位，输出数据位时可以 MSB 在先也可以 LSB 在先。**

在从机模式中，SPI 等待输入信号 SPI\_CS 的低电平，当检测到 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=0 时，从机在 SCK 信号的第一个边沿之前开始传输数据的第一个 bit 位（SPI\_CS 的下降沿）；当 CPHA=1 时，从机会在 SCK 信号的第一个边沿开始传输数据的第一个 bit 位。

注：

- 1) SPI 从机模式在 CPHA=0 时，不支持 SPI\_CS 一直被拉低的连续传输，因为从机数据的第一个 bit 位发送取决于检测到 SPI\_CS 的下降沿（使用时注意）。

### 13.2.5 快速从机模式（仅 SPI0/SPI2 支持）

快速从机模式为带 FIFO 的快速从机模式，必须开启 FIFO 功能，并配置 FIFOCON.FS\_EN=1。同样此模式下，可能会因受到毛刺影响导致数据采样错误，使用时需注意。

**注意：带 FIFO 的快速从机模式在使用时，需先关闭 SPI\_EN，配置相关功能再重新使能 SPI\_EN，同时需注意关闭重新打开 FIFOEN 或写 FIFORST 复位 FIFO。**

### 13.2.6 FIFO 模式（仅 SPI0/SPI2 支持）

在 FIFO 模式下有以下注意事项

- 当 FIFO\_EN 使能时，SPIF 的中断及标志会被屏蔽；在使用时根据 FIFO 相关中断标志进行应用处理。
- TXFIFO\_EN=1 时，SPI 发送无法配合 DMA 使用；RXFIFO\_EN=1 时，SPI 接收无法配合 DMA 使用。
- SPISSN.SSN1 对 FIFO 模式也生效，但在 FIFO 模式下，不能配置 CSDLY.CSB 和 CSDLY.CSA。
- 字与字之间延时期间的 CS 控制位 HLCS，只在 CS 由硬件自动控制时生效。

注：

FIFO 模式下的自动 CS 控制，默认在发送前的  $(1/2SPICLK+2SYSCLK)$  进行拉低，当 FIFO 数据全都发完时，自动拉高 CS。

主机 FIFO 模式下 (TXFIFO\_EN=1)，可以通过配置 BDLY[15:0] 来控制发送数据每个 byte 之间的延时（以 SPI 模块所使用的时钟为单位），FIFOCON.HLCS 的值决定在发送字与字之间的延时期间 SPI\_CS 的高低电平（HLCS=0，延时期间 SPI\_CS=0；HLCS=1，延时期间 SPI\_CS=1）。

### 13.2.7 中断功能

名称	SPI 中断标志描述	SPI 支持
SPIF	当传输完成，该标志位由硬件置位	SPI0、SPI1、SPI2、SPI3
MODF	当 SPI_CS 的状态与主从模式设置有冲突	SPI0、SPI1、SPI2、SPI3
RXT_IF	接收 FIFO 数据大于等于触发阈值，该标志置位	SPI0、SPI2
TXT_IF	发送 FIFO 数据小于等于触发阈值，该标志置位	SPI0、SPI2
RXO_IF	接收 FIFO 溢出，该标志置位	SPI0、SPI2
TXO_IF	发送 FIFO 溢出，该标志置位	SPI0、SPI2
RXF_IF	接收 FIFO 满，该标志置位	SPI0、SPI2
TXF_IF	发送 FIFO 满，该标志置位	SPI0、SPI2
RXNE_IF	接收 FIFO 非空，该标志置位	SPI0、SPI2
TXE_IF	发送 FIFO 空，该标志置位	SPI0、SPI2

### 13.3 特殊功能寄存器列表

SPI 模块寄存器基地址： 0x4000B000 (SPI0 模块)； 0x40017000 (SPI1 模块)； 0x40016000 (SPI2 模块)； 0x4001F000 (SPI3 模块)；				
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
0x10	SPIDIV	R/W	0x0010	SPI 分频辅助寄存器
0x14	CSDLY	R/W	0x0000	SPI cs/delay 控制寄存器
0x30	FIFOCON	R/W	0x02000100	SPI FIFO 控制寄存器
0x34	FIFOIE	R/W	0x0000	SPI FIFO 中断使能寄存器
0x38	FIFOIF	R/W	0x0001	SPI FIFO 状态寄存器
0x3C	BDLY	R/W	0x0000	字间隔配置寄存器
0x40	RXFIFOSTA	R/W	0x00	接收 FIFO 状态寄存器
0x44	TXFIFOSTA	R/W	0x00	发送 FIFO 状态寄存器

### 13.4 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, 0x40017000, 0x40016000, 0x4001F000 偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	Reserved	LSBEN
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
Reserved	该位必须保持为0

LSBEN	<b>SPI低位在前/高位在前控制位</b> 0: SPI发送接收时高位在前, 即 MSB (default) 1: SPI 发送接收时低位在前, 即 LSB																																				
SSDIS	<b>SS控制位</b> 0: 在主/从模式中打开SPI_CS功能 1: 在主/从模式中关闭 SPI_CS 功能, 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> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="3">SPR[2:0]</th> <th>SPI时钟速率</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Fsys/2</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Fsys/4</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Fsys/8</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Fsys/16</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Fsys/32</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Fsys/64</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Fsys/128</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>不产生主时钟</td> </tr> </tbody> </table>	SPR[2:0]			SPI时钟速率	0	0	0	Fsys/2	0	0	1	Fsys/4	0	1	0	Fsys/8	0	1	1	Fsys/16	1	0	0	Fsys/32	1	0	1	Fsys/64	1	1	0	Fsys/128	1	1	1	不产生主时钟
SPR[2:0]			SPI时钟速率																																		
0	0	0	Fsys/2																																		
0	0	1	Fsys/4																																		
0	1	0	Fsys/8																																		
0	1	1	Fsys/16																																		
1	0	0	Fsys/32																																		
1	0	1	Fsys/64																																		
1	1	0	Fsys/128																																		
1	1	1	不产生主时钟																																		
CPHA	<b>时钟相位</b> 0: 表示数据的第一位将在SCK的第一个动态沿被采样 1: 表示数据的第一位将在SCK的第二个动态沿被采样																																				
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</b> 0: 关闭SPI模块 1: 打开 SPI 模块																																				

## SPISTA (SPI 状态寄存器)

<b>SPISTA</b> (SPI 状态寄存器)			基地址: 0x4000B000, 0x40017000, 0x40016000, 0x4001F000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	SPIF	WCOL	SSERR	MODF

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

位	功能描述
SPIF	<b>数据传输完成标志位</b> 当传输完成时由硬件置位；可通过先读寄存器SPISTA，再读SPIDAT来清除该标志位。 这位写零清零；
WCOL	<b>写冲突标志位</b> 当写SPIDAT冲突时由硬件置位；可通过先读寄存器SPISTA，再读SPIDAT来清除该标志位。 这位写零清零；
SSERR	<b>同步从机错误标志位</b> 在接收完成前，SPI_CS被拉高，硬件置位；关闭SPI模块可清除该位（设置spen=0）。 这位写零清零；
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“SPISTA”寄存器来复位。 这位写零清零；

## SPIDAT（SPI 数据寄存器）

<b>SPIDAT</b> （数据寄存器）			基地址： 0x4000B000, 0x40017000, 0x40016000, 0x4001F000 偏移地址： 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SPIDAT[7:0]							
Write:	SPIDAT[7:0]							
Reset:	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 (从机选择寄存器)**

<b>SPISSN</b> (从机选择寄存器)			基地址: <b>0x4000B000, 0x40017000, 0x40016000, 0x4001F000</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>	MSAM1	MSAM0	Reserved	Reserved	Reserved	Reserved	SSN1	SSN0
<b>Write:</b>								
<b>Reset:</b>	1	1	1	1	1	1	1	1

位	功能描述
Reserved	内部保留位, 对用户无意义, 更改其值不产生任何影响。
MSAM[1:0]	主机延后采样控制位 11: 0 Fsysclk (default) 10: 1 Fsysclk 01: 2 Fsysclk 00: 3 Fsysclk
SSN1	<b>SPIDAT 控制 SPI_CS 使能控制</b> 1: 软件控制 SPI_CS 状态 (default) 0: 硬件自动控制 SPI_CS 状态 <b>问题: 当 SSN1=0 时, 软件仍然可以控制 SPI_CS</b>
SSN0	<b>SPI_CS 主模式引脚高低电平控制位</b> 1: 拉高 SPI_CS (default) 0: 拉低 SPI_CS 当芯片做 SPI 通讯的主机时, 在使能 SPI 情况下, 使用 SSN0 控制位可以控制芯片外部 SPI_CS 引脚的高低电平。

**CSDLY (SPI cs/delay 控制寄存器)**

<b>CSDLY</b> (SPI cs/delay 控制寄存器)			基地址: <b>0x4000B000, 0x40017000, 0x40016000, 0x4001F000</b> 偏移地址: <b>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>	CSB[1:0]		CSA[1:0]		ITDelay[3:0]			

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

位	功能描述
CSB[1:0]	00: 发送数据前保持CS不变 (default) 01: 发送数据前提前2个SPICLK拉低CS (即先拉低CS, 等待2个SPICLK延迟后再发送数据) 10: 发送数据前提前4个SPICLK拉低CS 11: 发送数据前提前8个SPICLK拉低CS
CSA[1:0]	00: 发送数据后保持CS不变 (default) 01: 发送数据后延迟2个SPICLK拉高CS 10: 发送数据后延迟4个SPICLK拉高CS 11: 发送数据后延迟8个SPICLK拉高CS
ITDelay[3:0]	发送数据完成后等待ITDelay[3:0]*8 个SPICLK时间给出DMA请求

注: CSB、CSA 控制位的配置, 需要先配置 SPISSN->SSN1=0, 才能生效。

## FIFOCON (FIFO 控制寄存器)

FIFOCON (FIFO 控制寄存器)		基地址: 0x4000B000, 0x40016000 偏移地址: 30H						
	Bit31	30	29	28	27	26	25	Bit24
<b>Read:</b>	X	X	X	X	X	X	CTR[1:0]	
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	1	0
	Bit23	22	21	20	19	18	17	Bit16
<b>Read:</b>	Reserved	Reserved	Reserved	RXFIFO_TL[4:0]				
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	Reserved	Reserved	Reserved	TXFIFO_TL[4:0]				
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	1
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	EX_IN_	HLCS	RXFIFO	TXFIFO	X	FS_EN	RXFIFO	TXFIFO
<b>Write:</b>	SEL		_RST	_RST			_EN	_EN
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
CTR[1:0]	<b>SPI快速从机模式2滤波控制位</b> 00: 0ns 01: 1.5ns

	10: 3ns (default) 11: 4.5ns
RXFIFO_TL[4:0]	<b>接收FIFO触发阈值设定控制位</b> 配置接收FIFO的阈值为 ( <b>RXFIFO_TL[4:0]</b> ), 当RXFIFO内数据大于所配置阈值时, 相应的状态标志会置起 注: 数字内部实际保留了RXFIFO_TL[4:0]的宽度, 最大可配置为31, >31的值按 (RXFIFO_TL[4:0])生效。
TXFIFO_TL[4:0]	<b>发送FIFO触发阈值设定控制位</b> 配置发送FIFO的阈值为 ( <b>TXFIFO_TL[4:0]</b> ), 当TXFIFO内数据小于等于所配置阈值时, 相应的状态标志会置起。(建议用户不要配置为0) 注: 数字内部实际保留了TXFIFO_TL[4:0]的宽度, 最大可配置为31, >31的值按 (TXFIFO_TL[4:0])生效。
EX_IN_SEL	<b>FIFO的数据来源选择控制位</b> 0: 来自APB写入 (default) 1: 来自外部端口 <b>注: SPI0/2的外部端口为打包模块, 在使用时SPI0和SPI2此位不能同时写1。</b>
HLCS	CS 在 BDLY 延时期期间拉高拉低控制 0: 拉低 SPI_CS (default) 1: 拉高 SPI_CS 注: 只在 CS 由硬件自动控制时生效。
RXFIFO_RST	写1复位RXFIFO, 自动清0
TXFIFO_RST	写1复位TXFIFO, 自动清0
FS_EN	<b>带FIFO快速从机使能位,</b> 0: 不使能 (default) 1: 使能 此模式在从机模式下生效, 且带有fifo
RXFIFO_EN	<b>接收FIFO</b> 0: 接收FIFO不使能 (default) 1: 接收FIFO使能
TXFIFO_EN	<b>发送FIFO</b> 0: 发送FIFO不使能 (default) 1: 发送FIFO使能

注: 当 FIFO\_EN 使能时, SPIF 的中断和标志会被屏蔽; 在使用 FIFO 模式时, 需同时收发应同时使能 RXFIFO\_EN 和 TXFIFO\_EN。

## FIFOIE (FIFO 中断使能寄存器)

<b>FIFOIE</b> (FIFO 中断使能寄存器)	基地址: <b>0x4000B000, 0x40016000</b> 偏移地址: <b>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>								TXFIFO
<b>Write:</b>	X	X	X	X	X	X	X	_DONE_ IE
<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>							RXNE_I	
<b>Write:</b>	RXT_IE	TXT_IE	RXO_IE	TXO_IE	RXF_IE	TXF_IE	E	TXE_IE
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
TXFIFO_DONE_IE	<b>FIFO数据发送完成中断使能位</b> 0: FIFO数据发送完成中断不使能 (default) 1: FIFO数据发送完成中断使能
RXT_IE	<b>接收FIFO大于触发阈值中断</b> 0: 接收FIFO阈值触发中断不使能 (default) 1: 接收FIFO阈值触发中断使能
TXT_IE	<b>发送FIFO小于等于触发阈值中断</b> 0: 发送FIFO阈值触发中断不使能 (default) 1: 发送FIFO阈值触发中断使能
RXO_IE	<b>接收FIFO溢出中断</b> 0: 接收FIFO溢出中断不使能 (default) 1: 接收FIFO溢出中断使能
TXO_IE	<b>发送FIFO溢出中断</b> 0: 发送FIFO溢出中断不使能 (default) 1: 发送FIFO溢出中断使能
RXF_IE	<b>接收FIFO满中断</b> 0: 接收FIFO满中断不使能 (default) 1: 接收FIFO满中断使能
TXF_IE	<b>发送FIFO满中断</b> 0: 发送FIFO满中断不使能 (default) 1: 发送FIFO满中断使能
RXNE_IE	<b>接收FIFO非空中断</b> 0: 接收FIFO非空中断不使能 (default) 1: 接收FIFO非空中断使能
TXE_IE	<b>发送FIFO空中断</b> 0: 发送FIFO空中断不使能 (default) 1: 发送FIFO空中断使能

注：当 FIFO\_EN 使能时，SPIF 的中断及标志会被屏蔽。在使用时根据 FIFO 相关中断标志进行应用处理。

**FIFOIF (FIFO 中断标志寄存器)**

<b>FIFOIF</b> (FIFO 中断标志寄存器)		基地址: 0x4000B000, 0x40016000 偏移地址: 38H						
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	X	X	X	X	X	X	X	TXFIFO_DONE_IF
<b>Write:</b>								X
<b>Reset:</b>	0	0	0	0	0	0	0	1
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	RXT_IF	TXT_IF	RXO_IF	TXO_IF	RXF_IF	TXF_IF	RXNE_I F	TXE_IF
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	1

位	功能描述
TXFIFO_DONE_IF	<b>FIFO数据发送完成中断标志位 (写0清0)</b> 0: FIFO数据发送未完成 (default) 1: FIFO数据发送完成 注: 当发送FIFO空, 且移位寄存器中的数据也全部移除后, 该标志会置起。
RXT_IF	<b>接收FIFO大于触发阈值中断标志位</b> 0: 接收FIFO数据不大于触发阈值 1: 接收FIFO数据大于触发阈值
TXT_IF	<b>发送FIFO小于等于触发阈值中断标志位</b> 0: 发送FIFO数据大于触发阈值 1: 发送FIFO数据小于等于触发阈值
RXO_IF	<b>接收FIFO溢出中断标志位 (写0清0)</b> 0: 接收FIFO未溢出 1: 接收FIFO溢出
TXO_IF	<b>发送FIFO溢出中断标志位 (写0清0)</b> 0: 发送FIFO未溢出 1: 发送FIFO溢出
RXF_IF	<b>接收FIFO满中断标志位</b> 0: 接收FIFO未满 1: 接收FIFO满
TXF_IF	<b>发送FIFO满中断标志位</b> 0: 发送FIFO未满 1: 发送FIFO满
RXNE_IF	<b>接收FIFO非空中断标志位</b> 0: 接收FIFO空 1: 接收FIFO非空

TXE_IF	发送FIFO空中断标志位 0: 发送FIFO非空 1: 发送FIFO空 (default)
--------	---

注：当 FIFO\_EN 使能时，SPIF 的中断会被屏蔽。在使用时根据 FIFO 相关中断标志进行应用处理。

## BDLY（字间隔配置寄存器）

<b>BDLY</b> (字间隔配置寄存器)		基地址: 0x4000B000, 0x40016000 偏移地址: 3CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:	BDLY[15:8]							
Write:	BDLY[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	BDLY[7:0]							
Write:	BDLY[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
BDLY[15:0]	字间隔延时配置位 FIFO 模式下，主机发送时每个字节之间的延时控制，延时 BDLY[15:0]个 Fsys 时钟

## RXFIFOSTA（接收 FIFO 状态寄存器）

<b>RXFIFOSTA</b> (接收 FIFO 状态寄存器)		基地址: 0x4000B000, 0x40016000 偏移地址: 40H						
	Bit7	6	5	4	3	2	1	Bit10
Read:	Reserved		RXFIFO_CNT[5:0]					
Write:	Reserved		X					
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RXFIFO_CNT[5:0]	RXFIFO 数据计数状态位 RXFIFO_CNT[5:0]反映当前 RXFIFO 内数据字节数

## TXFIFOSTA (发送 FIFO 状态寄存器)

<b>TXFIFOSTA</b> (发送 FIFO 状态寄存器)			基地址: <b>0x4000B000, 0x40016000</b> 偏移地址: <b>44H</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>	Reserved	Reserved	TXFIFO_CNT[5:0]					
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
TXFIFO_CNT[5:0]	TXFIFO 数据计数状态位
0]	TXFIFO_CNT[5:0]反映当前 TXFIFO 内数据字节数

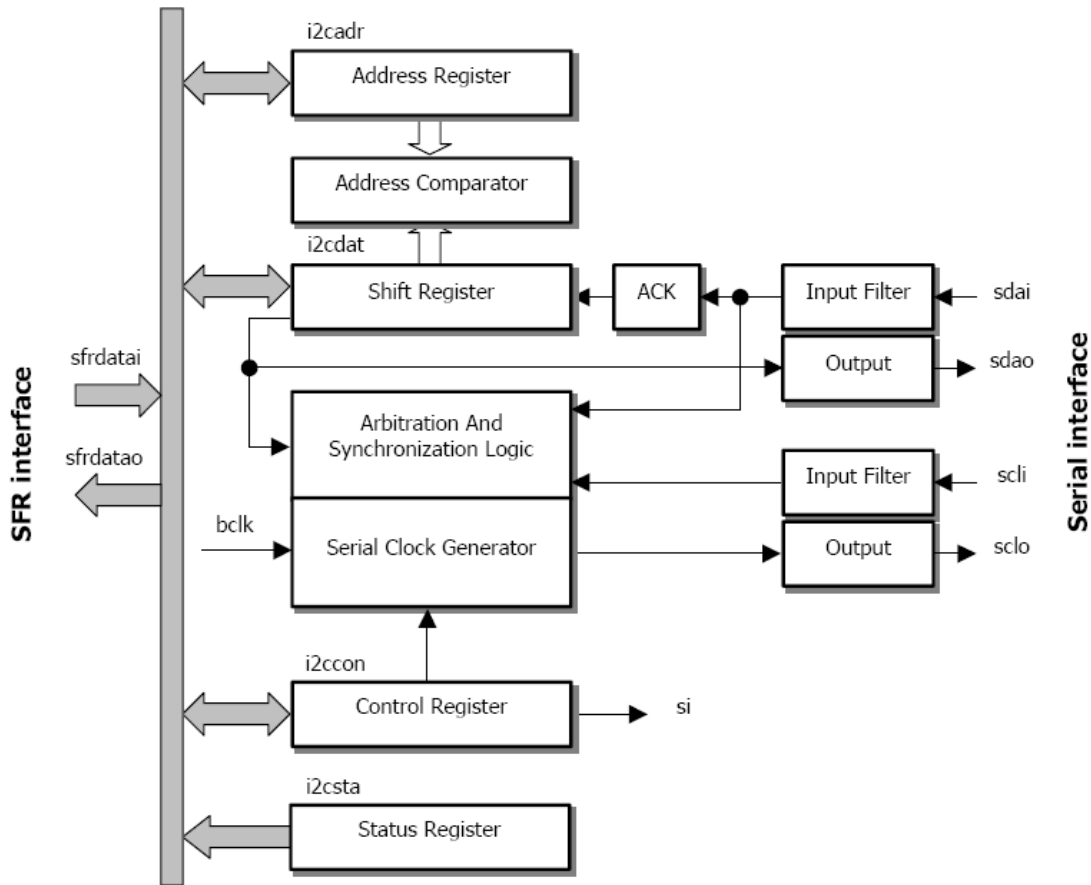
# 14 I2C 模块

## 14.1 概述

I2C 模块的芯片引脚为 PC.13/SCL 和 PC.14/SDA。

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

## 14.2 框图



I2C 模块功能框图

## 14.3 功能描述

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

### 14.3.1 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[9:0]控制,其中包含 I2CCON[1:0], I2CCON[14:7]。

### 14.3.2 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。

I2C 产生中断时，寄存器 I2CCON 中的标志位 SI 会被置位。

### 14.3.3 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，快速模式可达 400kbit/s 的速率。它可以下边四种模式工作：

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

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：

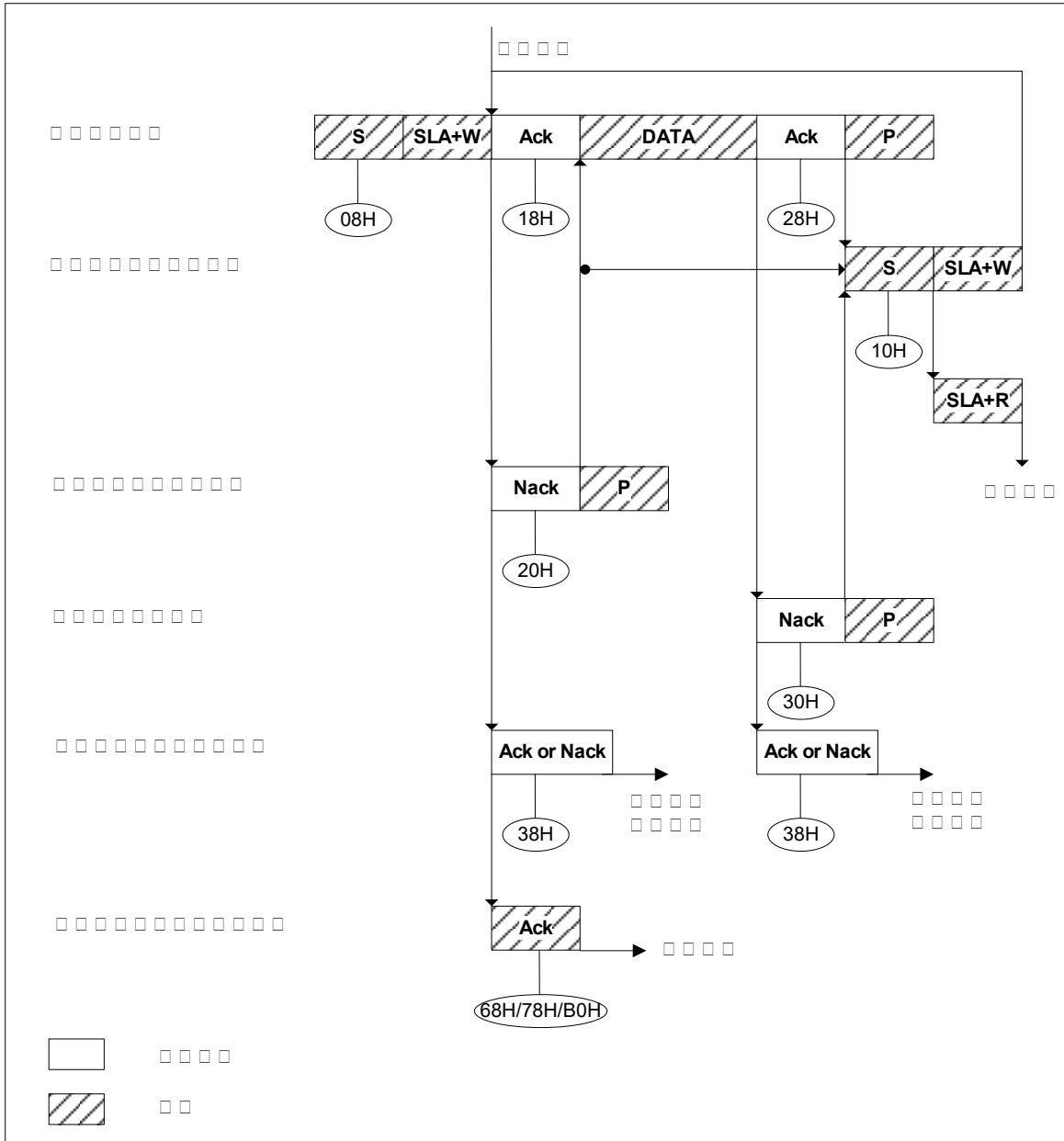
- S : 开始条件
- Rs : 重复开始条件
- R : 读控制位
- W : 写控制位
- A : 应答位
- $\bar{A}$  : 无应答位
- DATA : 8 位数据
- P : 终止条件
- SLA : 从机地址

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； “not 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

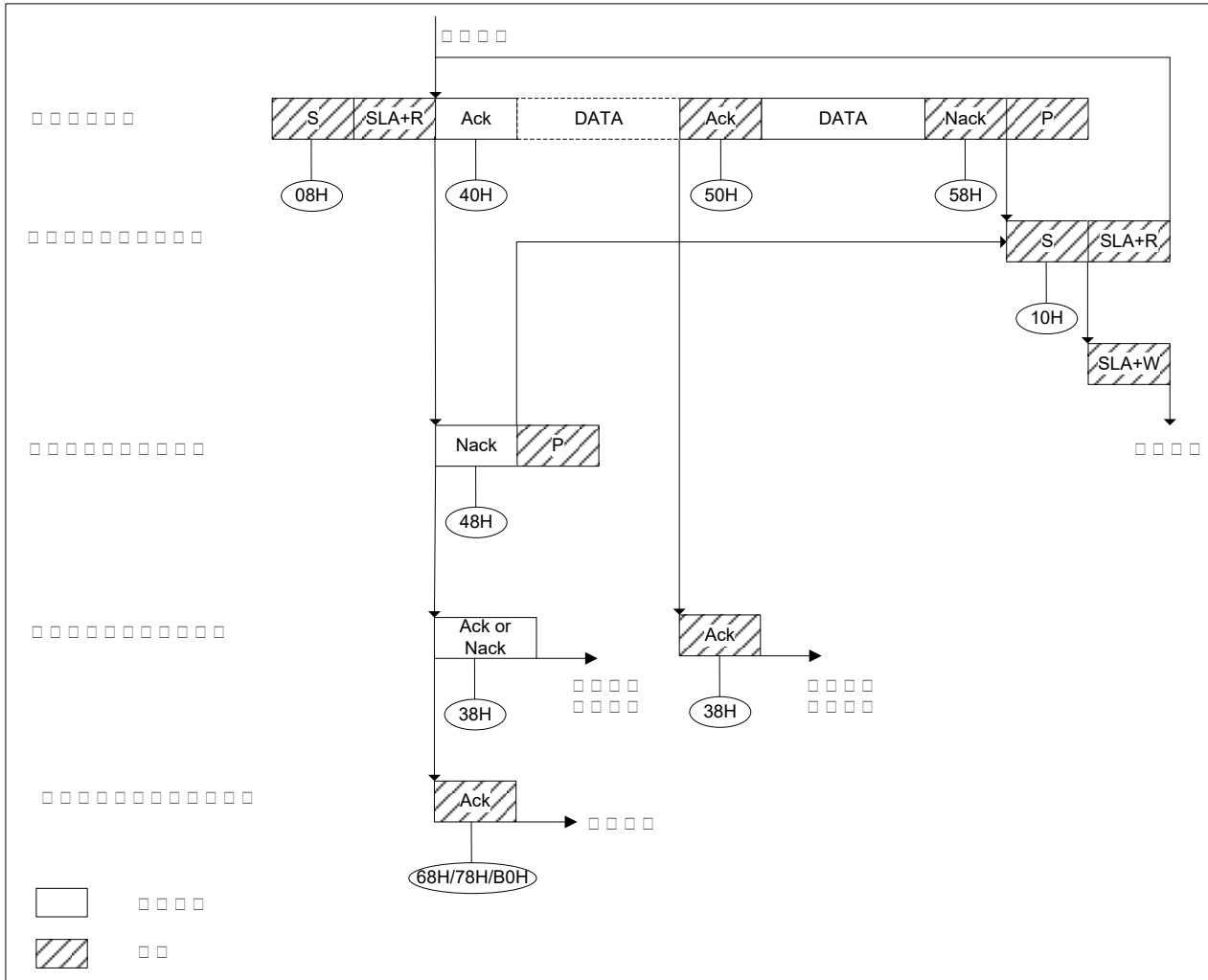


### I2C 主机接收模式:

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	0	0	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	0	0	0	X	I2C 总线将被释放；I2C 将会进入“从机”模式
		或者无动作	1	0	0	X	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	0	0	0	0	数据字节将被接收；将返回“not ACK”
		或者无动作	0	0	0	1	数据字节将被接收；将返回“ACK”
48H	SLA+R 已被发送； “not ACK”已被接收	无动作	1	0	0	X	重复起始条件将被发送
		或无动作	0	1	0	X	终止条件将被发送；sto 标志将被复位
		或无动作	1	1	0	X	起始条件被发送后将再发送一个终止条件；sto 标志将被复位
50H	数据字节已被接收； 已返回 ACK	读取数据字节	0	0	0	0	数据字节将被接收；将返回“not ACK”
		或读取数据字节	0	0	0	1	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	1	0	0	X	重复起始条件将被发送
		或读取数据字节	0	1	0	X	终止条件将被发送；sto 标志将被复位
		或读取数据字节	1	1	0	X	起始条件被发送后将再发送一个终止条件；sto 标志将被复位



### I2C 从机接收模式:

从机接收模式中，从机从主机接收一系列数据。

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

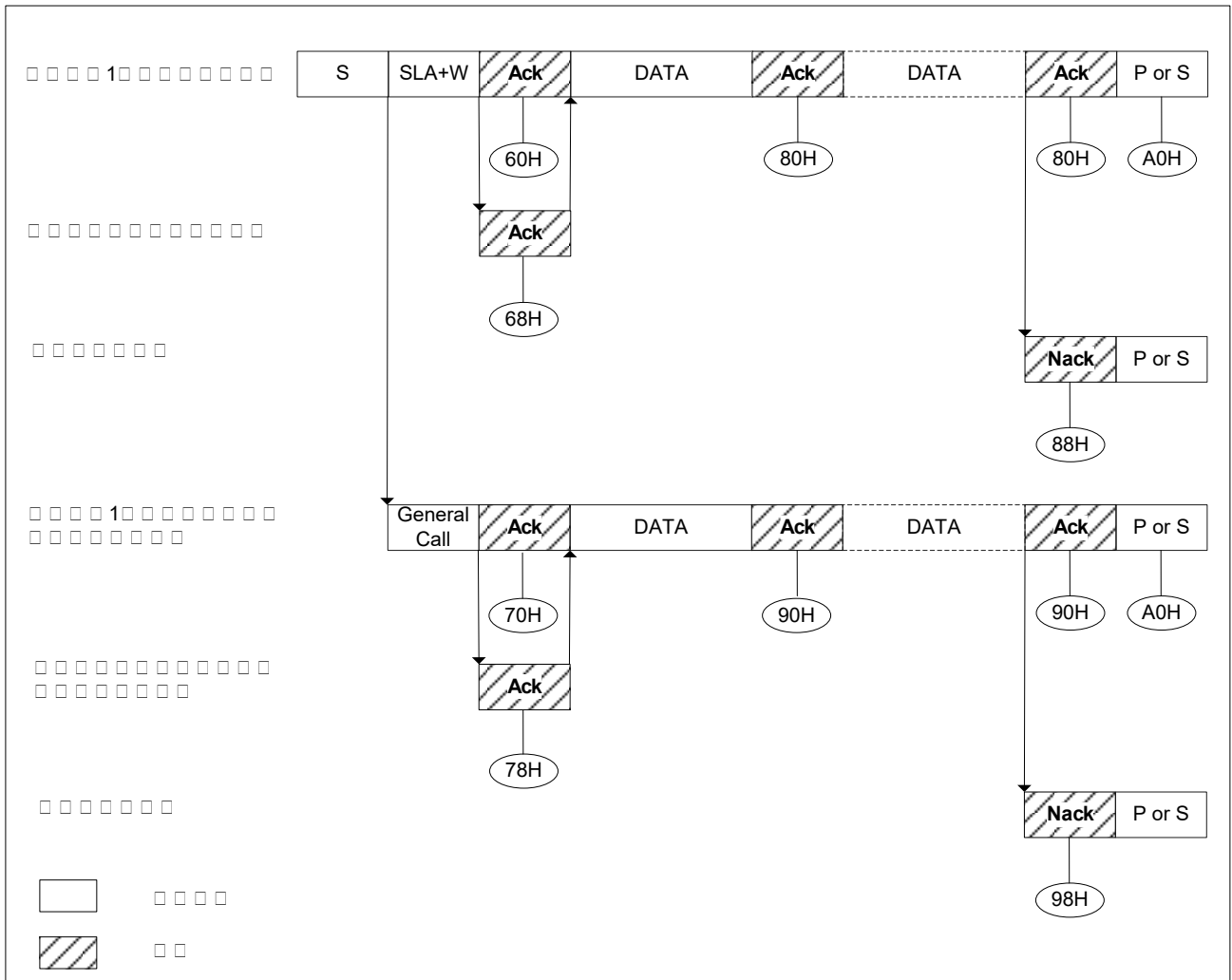
从机模式中，I2C 模块等待总线对本机地址或广播呼叫地址(如果 I2CADR[0]被置位)的寻址。如果读写数据位是‘写’，则 I2C 进入从机接收模式，否则将进入从机发送模式。

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

状态代码	I2C 状态	应用程序配置				I2C 硬件响应	
		I2CDAT	I2CCON				
			sta	sto	si		aa
60H	自身的 SLA+W 已被接收； 已返回 ACK	无动作	X	0	0	0	数据字节将被接收并返回“not ACK”
		或无动作	X	0	0	1	数据字节将被接收并返回 ACK
68H	主机 SLA+R/W 仲裁失败；自身的 SLA+W 已被接	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

	收, 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
A0 H	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

							身从机地址或呼叫地址;当总线空闲时将发送一个起始条件
--	--	--	--	--	--	--	----------------------------



### I2C 从机发送模式:

从机发送模式中，从机发送一系列数据到主机。

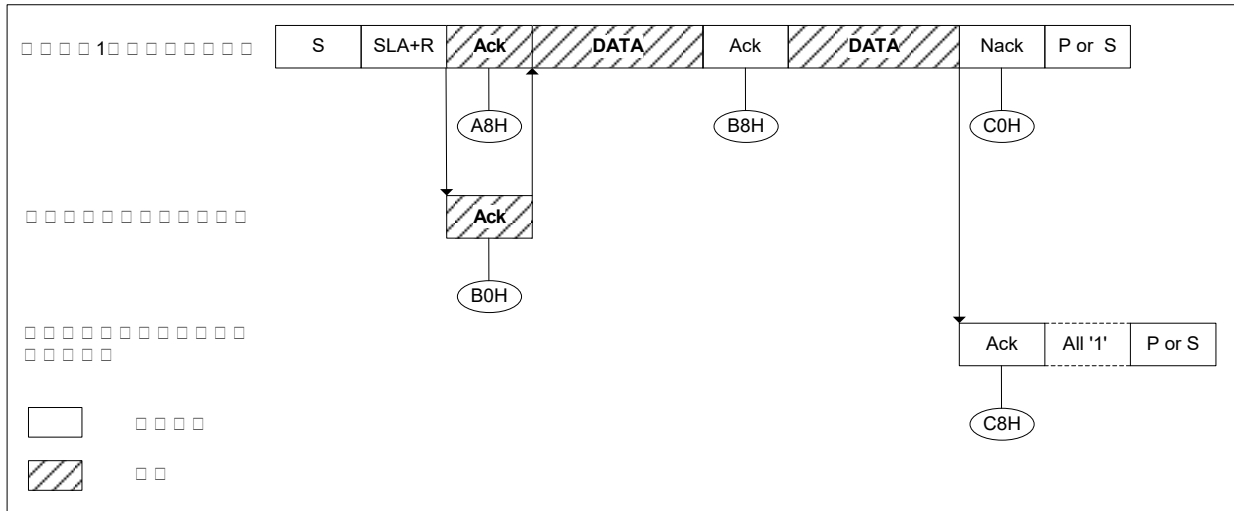
进入从机模式前，需设置从机地址，I2CADR 中 I2CADR[7..1]位为从机地址。如果 I2CADR[0]置位，从机也将响应广播呼叫地址(00H)；否则将不响应广播呼叫地址。

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

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

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

	ACK	加载数据字节	X	0	0	1	数据字节将被发送；ACK 将被接收
BOH	主机 SLA+R 仲裁失败；自身 SLA+R 已被接收；返回 ACK	加载数据字节	X	0	0	0	最后一个数据字节将被发送并接收 ACK
		或者 加载数据字节	X	0	0	1	数据字节将被发送；ACK 将被接收
B8H	数据字节已被发送；ACK 已被接收	加载数据字节	X	0	0	0	最后一个数据字节将被发送并接收 ACK
		或者 加载数据字节	X	0	0	1	数据字节将被发送；ACK 将被接收
COH	数据字节已被发送；“not ACK”已被接收	无动作	0	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址
		或者无动作	0	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址
		或者无动作	1	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
		或者无动作	1	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
C8H	最后一个数据字节已被发送；ACK 已被接收	无动作	0	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址
		或者无动作	0	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址
		或者无动作	1	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
		或者无动作	1	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件


**I2C 复合状态:**

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

## 14.4 特殊功能寄存器列表

微控制器与 I2C 组件的接口通过以下四个特殊功能寄存器来实现:

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

## 14.5 特殊功能寄存器说明

### I2CDAT (I2C 数据寄存器)

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

位	功能描述
I2CDAT[7:0]	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 (地址寄存器)

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

位	功能描述
I2CADR[7:0]	<b>bit[7:1]:I2C从机地址(7位)</b> <b>bit0: 呼叫地址确认位</b> 当此位置1时, 呼叫地址可以被识别, 否则不能被识别。

## I2CCON（控制寄存器）

<b>I2CCON</b> (控制寄存器)			基地址: <b>0x4000A000</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>	<b>X</b>	<b>CR[9:3]</b>						
<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>	<b>CR[2]</b>	<b>ENS1</b>	<b>STA</b>	<b>STO</b>	<b>SI</b>	<b>AA</b>	<b>CR[1:0]</b>	
<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> $I2C\ CLOCK = F_{sys} / (CR[9:0] + 1) / 4$
ENS1	<b>I2C 使能位</b> 1: 使能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; gc被置位的情况下接收到地址呼叫; 主机接收模式下一个字节接收完成; 从机接收模式下一个字节接收完成; 0: 非应答在以下情况下被返回: 主机接收模式下一个字节接收完成; 从机接收模式下一个字节接收完成;

## I2CSTA（状态寄存器）

<b>I2CSTA</b> （状态寄存器）			基地址: <b>0x4000A000</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>	I2CSTA[4:0]					X	X	X
<b>Write:</b>								
<b>Reset:</b>	1	1	1	1	1	0	0	0

位	功能描述
I2CSTA[4:0]	I2C状态码

寄存器 I2CSTA 反映 I2C 模块的实时状态。这个寄存器的低三位始终为 0。总共有 26 种可能的状态。当进入 25 种状态的其中一种时，都会产生中断；唯一一种不产生中断的情况是状态 F8H。

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

## 15 RTC 模块

### 15.1 概述

RTC 单元提供实时时钟、日历功能，自动闰年调整，支持闹钟和周期性中断。

RTC 模块在各种工作模式下都不会被关闭，在低功耗下仍然正常运行。

RTC 输出寄存器、RTC 时钟校正寄存器不会被复位，以保持 RTC 的准确性。相关寄存器请参考寄存器复位的相关说明。

RTC 模块的时钟源可选择 OSC 时钟 (Default) 和 LRC 时钟。

RTC 时间戳 (Timer Stamp) 功能，2 个时间戳触发端口 (STAMP0/1)；可记录触发事件及时间 (time&date)，内建事件计数器及时间缓存寄存器。最大可记录 6 次触发事件发生的时间。

### 15.2 修改寄存器 RTCchecksum 说明

Autoload 分为两级加载，第一级加载配置信息 (如看门狗配置、flash\sram 加密、RTC 加载使能等)，第二级为 RTC 系数自加载，如果第一级 autoload 配置 RTC 加载使能，则第二级加载生效。第二级自加载功能：从 info 地址 0x40104~0x4013B 加载到 RTC->DFAL~RTC->MCON45 寄存器中，加载过程中并累加计算校验和 RTCchecksum。例如：0x40104~0x40107 存放 DFAL 系数，需要加载到 RTC->DFAL 寄存器后， $RTCchecksum += RTC->DFAL$ 。0x40108~0x4010B 存放 DFAH 系数，需要加载到 RTC->DFAH 寄存器后， $RTCchecksum += RTC->DFAH$ 。

旧版本 RTCchecksum 寄存器是按照十六位累加的，高出 16 位数据忽略。新版本需要按照 32 位累加。

### 15.3 RTC 校时方式

- 默认校时方式

RTC 的秒、分、时、日、月、年和星期时，写入生效时间发生在下一次跳秒时沿，即，写入日历参数时，RTC 秒、分、时、日、月、年和星期不能立即更新，会延时 0~1 秒时间。

- 立即校时方式

具有 RTC 校准立即清零控制功能，控制位在寄存器 CALENDARCLR 中，该功能使能后，在 RTC 校准日历时，日历参数立即生效并清零秒以下的 ms 计时，从写入时刻，RTC 日历在修改后的时间基础上立刻正常累加。

### 15.4 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 RTC 闹铃中断功能
- 2 个定时器周期性中断功能

- 5 个时间中断功能（秒、分、时、日、月）
- 可输出频率为 1/2/8/16/32/64/128/32768Hz 的方波
- 可输出每秒补偿的校验脉冲
- 增加只读寄存器，用于保存校时的次数
- **2 个时间戳功能**

## 15.5 时钟校正

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

$$DFi = (DFA + DFB * (TMPDAT - Toff) + DFC * (TMPDAT - Toff)^2 + DFD * (TMPDAT - Toff)^3 + DFE * (TMPDAT - Toff)^4 + 0x02) >> 2$$

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 表示约 0.06ppm。

## 15.6 RTC 补偿系数寄存器说明

RTC 的补偿系数寄存器在芯片复位时（除唤醒复位）会自动装载，详见 FAQ。

## 15.7 时间和万年历

RTC 提供秒、分、时、日、月、年和星期输出寄存器。

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

## 15.8 RTC 时间戳功能

使用 RTC 时间戳功能，需要配置对应的外部 GPIO 复用选择 STAMP0/1 功能。

设置 RTCTSxCON 寄存器的 TS1EN 或 TS0EN 控制位，开启或关闭 STAMP1 或 STAMP0 事件触发源；若发生相应的触发信号，则内部 RTC 时间和日期将被自动存储到缓存寄存器，并置起 TSOIF/TS1IF 标志。若开启相应中断使能，则发生 RTC 中断。

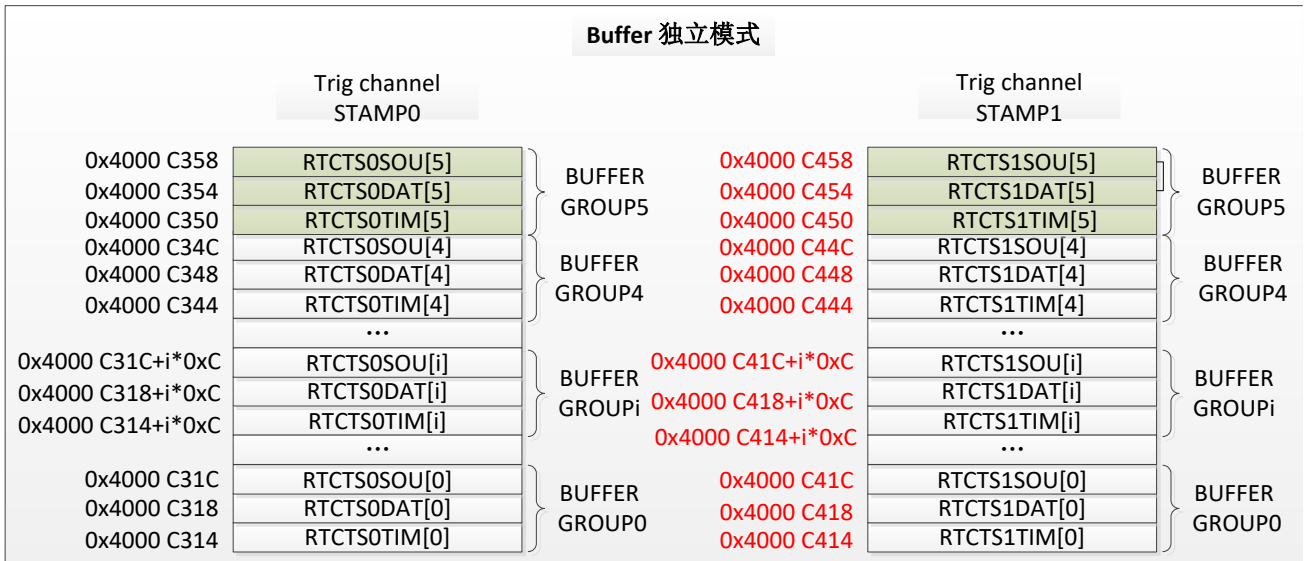
内建 6 组数据缓存寄存器，每组 3 个寄存器，分别存储时间（time）/日期/触发源信息。**最多可记录 6 次事件信息**。对应的寄存器如下：

RTCTSxTIM[i](x=0\1) 寄存器存储当前触发事件的时间（time）；

RTCTSxDAT[i](x=0\1) 寄存器存储当前触发事件的日期（date）；

RTCTSxSOU[i](x=0\1) 寄存器存储当前触发事件的触发源信息（source）。

0~5 组寄存器存储 STAMP 的触发事件，前 5 组存储最先发生的 5 次触发事件，第 6 组存储最近发生的 1 次事件；



例如：

发生 20 次事件，第 0~4 组寄存器存储 STAMP 发生的前 5 次事件，第 5 组寄存器存储 STAMP 发生的第 20 次事件；

每次触发对应的事件计数器加 1，内部计数器为 8bit，若溢出，相应的溢出标志置 1；若开启相应中断，则发生 RTC 中断。对应的寄存器如下：

RTCTSxCNT(x=0\1)寄存器存储发生有效触发事件的次数 (number)；

设置 RTCTSxCON(x=0\1)寄存器的 **TS1POL[1:0]**和 **TS0POL[1:0]**控制位，可以选择 **STAMP1** 和 **STAMPO** 引脚上信号触发沿（上升沿/下降沿/双沿）。

设置 RTCTSxCON(x=0\1)寄存器的 **TS1CMPEN/TS0CMPEN** 控制位，可开启时间戳计数比较功能，若 RTCTSxCNT(x=0\1) 事件计数器大于等于 RTCTSxCMP(x=0\1) 中的设置值，则计数比较中断标志位 **TS0IF/TS1IF** 置 1，若中断使能开启，则发生 RTC 中断。

## 15.9 中断功能

RTC 一共提供 8 种中断源，共用 MCU 的 IRQ-RTC 中断向量，向量号 20。RTC 的 8 种中断源由 RTCIE 控制其使能。

具体的中断产生条件和中断清除步骤如下：

**ALMF**: RTC 闹铃中断标志

当小时和分钟与设定的闹钟匹配时，产生 RTC 闹铃中断，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 RTCTMR1=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 RTCTMR2=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 MONTHR 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 HOU RR 加 1 时，产生一个小时中断，HRF 被置为 1。  
对该位写 0 清标志。

**MINF:** 分钟中断

分钟计数器 MINR 加 1 时，产生一个分钟中断，MINF 被置为 1。  
对该位写 0 清标志。

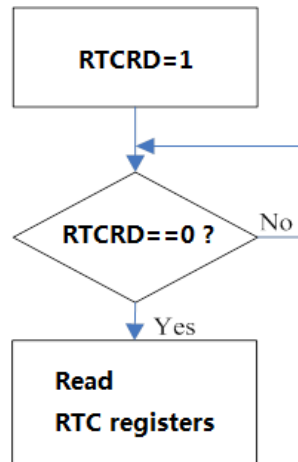
**SECF:** 秒中断

秒计数器 SECR 加 1 时，产生一个秒中断，SECF 被置为 1。  
对该位写 0 清标志。

## 15.10 RTC 指示寄存器读写流程

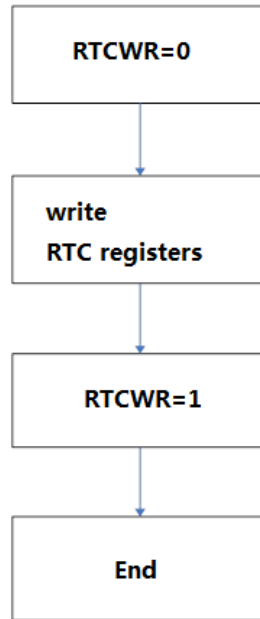
### 15.10.1 读取 RTC 指示寄存器流程

在用户需要读取 RTC 指示寄存器（SECR, MINR, HOU RR, DAYR, MONTHR, YEARR, WEEKR）的时候，用户应该按照以下流程操作：



### 15.10.2 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, HOU RR, DAYR, MONTHR, YEARR, WEEKR）的时候，用户应该按照以下流程操作，请一次性写入所有 7 个 RTC 指示寄存器，并且按照年、月、日、时、分、秒、周的顺序操作：



## 15.11 校时记录

校时次数增加以一次 RTCWCR 置 1 为准，无论怎么更新 RTC 寄存器，只要 RTCWCR 置 1，则 RTCCNT 加 1。

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

## 15.12 RTC 备份寄存器

为了方便用户存储所需要的数据信息，提供 128bytes 存储空间大小共计 32 个备份寄存器。

**注：RTC 备份寄存器为 SRAM，所以无固定复位值，掉电后再上电为随机值。**

## 15.13 寄存器复位的相关说明

RTC 模块 RTC 软复位不影响的寄存器列表：

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
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
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	0x2000	RTC 校准控制寄存器
0x80	MCON23	R/W	0x0588	RTC 校准控制寄存器
0x84	MCON45	R/W	0x4488	RTC 校准控制寄存器
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

## 15.14 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	0x0000	校时次数寄存器高 16 位
0x38	RTCCNTL	R	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	0x2000	RTC 校准控制寄存器
0x80	MCON23	R	0x0588	RTC 校准控制寄存器
0x84	MCON45	R	0x4488	RTC 校准控制寄存器
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
0x90	RTCRSTFLAG	R/W	0x0002	RTC 模块复位标志寄存器
0x94	RTCRSTSET	W	0x0000	RTC 写复位寄存器
0x98	RTCChecksum	R	0x00006A10	RTC 校验和寄存器
0xF0	Reserved	R/W	0x0000	保留寄存器, 请保持默认值
0xF4	Reserved	R/W	0x0001	保留寄存器, 请保持默认值
0xF8	Reserved	R	0x0000	保留寄存器, 请保持默认值
0xFC	Reserved	R	0x0000	保留寄存器, 请保持默认值
0x100	CTRLByFlash	R/W	0x00FFFA2	自加载结果寄存器
0x108	AUTOLOADS TA	R	0x0001	自加载状态寄存器
0x10C	RTCTESTCTL	R/W	0x0008	RTC 测试控制寄存器
0x110	CALENDARCL	R/W	0x0000	万年历校准清零寄存器
0x300	RTCTS0CON	R/W	0x0000	RTC 时间戳 0 控制寄存器
0x304	RTCTS0CMP	R/W	0x0000	RTC 时间戳 0 事件比较寄存器
0x308	RTCTS0CNT	R	0x0000	RTC 时间戳 0 事件计数寄存器
0x30C	RTCTS0IE	R/W	0x0000	RTC 时间戳 0 中断使能寄存器
0x310	RTCTS0IF	R/W	0x0000	RTC 时间戳 0 中断标志寄存器
0x314	RTCTS0TIM0	R	0x0000	RTC 时间戳 0 时间缓存寄存器 0
0x318	RTCTS0DAT0	R	0x0000	RTC 时间戳 0 日期缓存寄存器 0
0x31C	RTCTS0SOU0	R	0x0000	RTC 时间戳 0 触发源缓存寄存器 0
0x320	RTCTS0TIM1	R	0x0000	RTC 时间戳 0 时间缓存寄存器 1
0x324	RTCTS0DAT1	R	0x0000	RTC 时间戳 0 日期缓存寄存器 1
0x328	RTCTS0SOU1	R	0x0000	RTC 时间戳 0 触发源缓存寄存器 1

0x32C	RTCTS0TIM2	R	0x0000	RTC 时间戳 0 时间缓存寄存器 2
0x330	RTCTS0DAT2	R	0x0000	RTC 时间戳 0 日期缓存寄存器 2
0x334	RTCTS0SOU2	R	0x0000	RTC 时间戳 0 触发源缓存寄存器 2
0x338	RTCTS0TIM3	R	0x0000	RTC 时间戳 0 时间缓存寄存器 3
0x33C	RTCTS0DAT3	R	0x0000	RTC 时间戳 0 日期缓存寄存器 3
0x340	RTCTS0SOU3	R	0x0000	RTC 时间戳 0 触发源缓存寄存器 3
0x344	RTCTS0TIM4	R	0x0000	RTC 时间戳 0 时间缓存寄存器 4
0x348	RTCTS0DAT4	R	0x0000	RTC 时间戳 0 日期缓存寄存器 4
0x34C	RTCTS0SOU4	R	0x0000	RTC 时间戳 0 触发源缓存寄存器 4
0x350	RTCTS0TIM5	R	0x0000	RTC 时间戳 0 时间缓存寄存器 5
0x354	RTCTS0DAT5	R	0x0000	RTC 时间戳 0 日期缓存寄存器 5
0x358	RTCTS0SOU5	R	0x0000	RTC 时间戳 0 触发源缓存寄存器 5
0x400	RTCTS1CON	R/W	0x0000	RTC 时间戳 1 控制寄存器
0x404	RTCTS1CMP	R/W	0x0000	RTC 时间戳 1 事件比较寄存器
0x408	RTCTS1CNT	R	0x0000	RTC 时间戳 1 事件计数寄存器
0x40C	RTCTS1IE	R/W	0x0000	RTC 时间戳 1 中断使能寄存器
0x410	RTCTS1IF	R/W	0x0000	RTC 时间戳 1 中断标志寄存器
0x414	RTCTS1TIM0	R	0x0000	RTC 时间戳 1 时间缓存寄存器 0
0x418	RTCTS1DAT0	R	0x0000	RTC 时间戳 1 日期缓存寄存器 0
0x41C	RTCTS1SOU0	R	0x0000	RTC 时间戳 1 触发源缓存寄存器 0
0x420	RTCTS1TIM1	R	0x0000	RTC 时间戳 1 时间缓存寄存器 1
0x424	RTCTS1DAT1	R	0x0000	RTC 时间戳 1 日期缓存寄存器 1
0x428	RTCTS1SOU1	R	0x0000	RTC 时间戳 1 触发源缓存寄存器 1
0x42C	RTCTS1TIM2	R	0x0000	RTC 时间戳 1 时间缓存寄存器 2
0x430	RTCTS1DAT2	R	0x0000	RTC 时间戳 1 日期缓存寄存器 2
0x434	RTCTS1SOU2	R	0x0000	RTC 时间戳 1 触发源缓存寄存器 2
0x438	RTCTS1TIM3	R	0x0000	RTC 时间戳 1 时间缓存寄存器 3
0x43C	RTCTS1DAT3	R	0x0000	RTC 时间戳 1 日期缓存寄存器 3
0x440	RTCTS1SOU3	R	0x0000	RTC 时间戳 1 触发源缓存寄存器 3
0x444	RTCTS1TIM4	R	0x0000	RTC 时间戳 1 时间缓存寄存器 4
0x448	RTCTS1DAT4	R	0x0000	RTC 时间戳 1 日期缓存寄存器 4
0x44C	RTCTS1SOU4	R	0x0000	RTC 时间戳 1 触发源缓存寄存器 4
0x450	RTCTS1TIM5	R	0x0000	RTC 时间戳 1 时间缓存寄存器 5
0x454	RTCTS1DAT5	R	0x0000	RTC 时间戳 1 日期缓存寄存器 5
0x458	RTCTS1SOU5	R	0x0000	RTC 时间戳 1 触发源缓存寄存器 5
0xE00+0x0 4*x (x=0~31)	RTCBACKUP x	R/W	0x0000	第 x 个 RTC 备份寄存器

## 15.15 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
Read:	X	RTC2EN	RTC1EN	TOUT[3:0]				AutoC
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RTC2EN	<b>RTC 定时器 2 使能位</b> RTC2EN=0: RTC 定时器 2 被关闭 RTC2EN=1: RTC 定时器 2 被使能, 溢出产生 RTC2IF 标志。
RTC1EN	<b>RTC 定时器 1 使能位</b> RTC1EN=0: RTC 定时器 1 被关闭 RTC1EN=1: RTC 定时器 1 被使能, 溢出产生 RTC1IF 标志。
TOUT[3:0]	<b>TOUT 输出频率说明见下表</b>
AutoC	<b>RTC 自动补偿控制位</b> AutoC=0: RTC 自动补偿使能, 此时芯片内部会根据 DFA/B/C/D/E 和温度传感器的值自动计算当前误差值。用户此时对 DF <sub>I</sub> H/L 写无效 AutoC=1: RTC 手动补偿使能, 此时误差值由用户写入 DF <sub>I</sub> H/L (用户应先写 DF <sub>I</sub> H 再写 DF <sub>I</sub> L)。

TOUT[3:0]				TOUT	TOUT (PLL 使能)
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	<b>RTC 模块时钟</b>	
0	0	1	1	未经高频补偿的 1Hz	高频补偿得到的 1Hz
0	1	0	0	未经高频补偿的 2Hz	高频补偿得到的 2Hz
0	1	0	1	未经高频补偿的 4Hz	高频补偿得到的 4Hz
0	1	1	0	未经高频补偿的 8Hz	高频补偿得到的 8Hz
0	1	1	1	未经高频补偿的 16Hz	高频补偿得到的 16Hz
1	0	0	0	未经高频补偿的 32Hz	高频补偿得到的 32Hz
1	0	0	1	未经高频补偿的 64Hz	高频补偿得到的 64Hz
1	0	1	0	未经高频补偿的 128Hz	高频补偿得到的 128Hz

其他	Reserved
----	----------

说明：高频补偿后的 TOUT 输出精度要求为 0.06ppm；  
未经高频补偿的 TOUT 输出精度要求为 30.5ppm。

## RTCIE（RTC 中断使能寄存器）

<b>RTCIE</b> (RTC 中断使能寄存器)		基地址： 0x4000C000 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>	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>RTC 闹铃中断使能位</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)**

RTCIF (RTC 中断标志寄存器)			基地址: 0x4000C000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
<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>	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>RTC 闹铃中断标志位</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)			基地址: <b>0x4000C000</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	ALMH[4:0]				
<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]	RTC 闹铃中断小时设置, 允许写入 0-23 以外的数, 但是 RTC 闹铃中断将永远不会产生。
ALMM[5:0]	RTC 闹铃中断分钟设置, 允许写入 0-59 以外的数, 但是 RTC 闹铃中断将永远不会产生。 注: 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)			基地址: <b>0x4000C000</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>	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]	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。 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。

注：当定时器到达设定值时，如果用户没有关闭定时器，则定时器将从 0 开始重新计数。

## RTCTMR2 (RTC 定时器 2 寄存器)

<b>RTCTMR2</b> (RTC 定时器 2 寄存器)		基地址: <b>0x4000C000</b> 偏移地址: <b>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]	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。 CNT[15:0] 用来表示一个 16BIT 的二进制的无符号整数, 如果设置 CNT[15:0] = 00H, 表示 RTC 内部的秒表功能中断每经过 (00H+1)*0.0625S = 1*0.0625S = 0.0625S 的计时周期后, 置位 RTC2IF 标志。 注：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)

<b>SECR</b> (RTC 秒寄存器)		基地址: <b>0x4000C000</b> 偏移地址: <b>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	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]	秒计数器:

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

### MINR (RTC 分寄存器)

<b>MINR</b> (RTC 分寄存器)			<b>基地址: 0x4000C000</b>					
			<b>偏移地址: 1CH</b>					
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	X	X	X	X	X	X	X	X
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	X	X	MIN[5:0]					
<b>Write:</b>	X	X	MIN[5:0]					
<b>Reset:</b>	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，对该寄存器没有影响。

### HOURR (RTC 时寄存器)

<b>HOURR</b> (RTC 时寄存器)			<b>基地址: 0x4000C000</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>	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	HOUR[4:0]				
<b>Write:</b>	X	X	X	HOUR[4:0]				
<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，对该寄存器没有影响。

**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	1

位	功能描述
DAY[4:0]	<b>日计数器:</b> 可设范围: 1-28/29/30/31。写入与年、月不匹配的任何数值, 对该寄存器没有影响。 注: (1) 写入大于 31 的数据 & DAYR.bit5 = 0 时, DAYR 的值为写入值的低 4 位; (2) 写入大于 31 的数据 & DAYR.bit5 = 1 时, 对该寄存器写入无效。

**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	1

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

## YEARR (RTC 年寄存器)

<b>YEARR</b> (RTC 年寄存器)			基地址: <b>0x4000C000</b> 偏移地址: <b>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, 对该寄存器没有影响。

## WEEKR (RTC 周寄存器)

<b>WEEKR</b> (RTC 周寄存器)			基地址: <b>0x4000C000</b> 偏移地址: <b>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	1

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, 对该寄存器没有影响。 注: 其它数值也可以写入, 取低 3 位。

## RTCCNTH (RTC 校时次数寄存器高 16 位)

<b>RTCCNTH</b>	基地址: <b>0x4000C000</b>
----------------	------------------------

(RTC 校时次数寄存器高 16 位)			偏移地址: 34H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	RTCCNTH[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	RTCCNTH[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RTCCNTH[15:0]	校时次数高 16 位, 与 RTCCNTL[15:0]构成 32 位寄存器, 用于记录校时次数。该寄存器只读。

### RTCCNTL (RTC 校时次数寄存器低 16 位)

RTCCNTL (RTC 校时次数寄存器低 16 位)			基地址: 0x4000C000 偏移地址: 38H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	RTCCNTL[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	RTCCNTL[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RTCCNTL[15:0]	校时次数低 16 位, 与 RTCCNTH[15:0]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)

RTCRD (RTC 读控制寄存器)			基地址: 0x4000C000 偏移地址: 3CH					
	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>	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“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)			基地址: <b>0x4000C000</b> 偏移地址: <b>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“RTC 指示寄存器读写流程”

### DFAH (RTC 硬件补偿常数项系数高位)

<b>DFAH</b> (RTC 硬件补偿常数项系数高位)			基地址: <b>0x4000C000</b> 偏移地址: <b>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	DFA[22:16]						
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

**DFAL (RTC 硬件补偿常数项系数低位)**

<b>DFAL</b> (RTC 硬件补偿常数项系数低位)			基地址: <b>0x4000C000</b> 偏移地址: <b>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>	DFA[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>	DFA[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DFA[22:0]	23 位寄存器, 最高位为符号位。 RTC 硬件补偿常数项系数

**DFBH (RTC 硬件补偿一次项系数高位)**

<b>DFBH</b> (RTC 硬件补偿一次项系数高位)			基地址: <b>0x4000C000</b> 偏移地址: <b>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	DFB[22:16]						
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

**DFBL (RTC 硬件补偿一次项系数低位)**

<b>DFBL</b> (RTC 硬件补偿一次项系数低)			基地址: <b>0x4000C000</b> 偏移地址: <b>5CH</b>					
---------------------------------	--	--	--	--	--	--	--	--

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

位	功能描述
DFB[22:0]	23 位寄存器，最高位为符号位。 RTC 硬件补偿一次项系数

### DFCH (RTC 硬件补偿二次项系数高位)

DFCH (RTC 硬件补偿二次项系数高位)			基地址: 0x4000C000 偏移地址: 60H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X							
Write:								
Reset:	0	0	0	0	0	0	0	0

### DFCL (RTC 硬件补偿二次项系数低位)

DFCL (RTC 硬件补偿二次项系数低位)			基地址: 0x4000C000 偏移地址: 64H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	DFC[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DFC[7:0]							

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

位	功能描述
DFC[22:0]	23 位寄存器，最高位为符号位。 RTC 硬件补偿二次项系数

### DFDH (RTC 硬件补偿三次项系数高位)

<b>DFDH</b> (RTC 硬件补偿三次项系数高位)			基地址: 0x4000C000 偏移地址: 68H					
	<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	DFD[22:16]						
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

### DFDL (RTC 硬件补偿三次项系数低位)

<b>DFDL</b> (RTC 硬件补偿三次项系数低位)			基地址: 0x4000C000 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>	DFD[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>	DFD[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DFD[22:0]	23 位寄存器，最高位为符号位。 RTC 硬件补偿三次项系数

**DFEH (RTC 硬件补偿四次项系数高位)**

<b>DFEH</b> (RTC 硬件补偿四次项系数高位)			基地址: 0x4000C000 偏移地址: 70H					
	<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	DFE[22:16]						
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

**DFEL (RTC 硬件补偿四次项系数低位)**

<b>DFEL</b> (RTC 硬件补偿四次项系数低位)			基地址: 0x4000C000 偏移地址: 74H					
	<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>	DFE[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>	DFE[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DFE[22:0]	23 位寄存器, 最高位为符号位。 RTC 硬件补偿四次项系数

**Toff (温度偏置寄存器)**

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

位	功能描述
Toff [15:0]	16 位有符号数，用户不需操作该寄存器

### MCON01（RTC 校准控制寄存器）

<b>MCON01</b> (RTC 校准控制寄存器)			<b>基地址: 0x4000C000</b> <b>偏移地址: 7CH</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>	MCON01[15:8]							
<b>Write:</b>	X	X	X	X	X	X	X	X
<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>	MCON01[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
MCON01[15:0]	RTC 校准控制寄存器。 请确保此寄存器值为 0x2000

### MCON23（RTC 校准控制寄存器）

<b>MCON23</b> (RTC 校准控制寄存器)			<b>基地址: 0x4000C000</b> <b>偏移地址: 80H</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>	MCON23[15:8]							
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	1	0	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>	MCON23[7:0]							
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	1	0	0	0	1	0	0	0

位	功能描述
MCON23[15:0]	RTC 校准控制寄存器。 请确保此寄存器值为 0x0588

### MCON45（RTC 校准控制寄存器）

MCON45 (RTC 校准控制寄存器)			基地址: 0x4000C000 偏移地址: 84H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	MCON45[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	1	0	0	0	1	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	MCON45[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	1	0	0	0	1	0	0	0

位	功能描述
MCON45[15:0]	RTC 校准控制寄存器。 请确保此寄存器值为 0x4488

### DFiH（频率误差寄存器高 5 位）

DFiH (频率误差寄存器高 5 位)			基地址: 0x4000C000 偏移地址: 88H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	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	DFi[20:16]				
Write:	X	X	X					
Reset:	0	0	0	0	0	0	0	0

**DFiL (频率误差寄存器低 16 位)**

<b>DFiL</b> (频率误差寄存器低 16 位)		基地址: <b>0x4000C000</b> 偏移地址: <b>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[20:0]	21 位寄存器, 最高位为符号位。 当 AutoC 为 0 时, 此时用户对 DFiH/ DFiL 写无效。 当 AutoC 为 1 时, 手动补偿方式, 补偿值由用户写入 DFiH/ DFiL 寄存器。 用户必须先写高位, 再写低位(即按照 DFiH—DFiL 顺序写寄存器)。 DFi 每一个 LSB 表示 1/512Hz(时钟源是 32768), 即约 0.06ppm/LSB

**RTCRSTFLAG (RTC 模块复位标志寄存器)**

<b>RTCRSTFLAG</b> (RTC 模块复位标志寄存器)		基地址: <b>0x4000C000</b> 偏移地址: <b>90H</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	PORRST	SoftRST
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	1	0

位	功能描述
PORRST	VRTC 管脚发生 PORRST 复位的标志: 0: VRTC 电源未发生 POR 复位 1: VRTC 电源发生了 POR 复位 (写 0 清 0)
SoftRST	RTC、 <b>STAMP</b> 与 TBS 模块发生软件复位的复位标志: 0: RTC、 <b>STAMP</b> 与 TBS 模块未发生软复位 1: RTC、 <b>STAMP</b> 与 TBS 模块发生了软复位 (写 0 清 0)

## RTCRSTSET (RTC 写复位寄存器)

<b>RTCRSTSET</b> (RTC 写复位寄存器)			基地址: <b>0x4000C000</b> 偏移地址: <b>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>	X	X	X	X	X	X	X	X
<b>Write:</b>	RTCRSTSET[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>	RTCRSTSET[7:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
RTCRSTSET [15:0]	当向此寄存器写入 0xAAAA, 再写入 0x5555 时, RTC、 <b>STAMP</b> 与 TBS 模块发生复位, 发生复位后, RTCRSTFLAG 寄存器中 SoftReset 会置 1。 除了 RTC 计时寄存器、校时次数寄存器以及 DFXH/DFXL (X=A, B, C, D, E,i), Toff, MCON01,MCON23,MCON45, <b>GPIOJ 相关寄存器</b> ,其他所有 <b>STAMP 寄存器</b> 、RTC 寄存器和 TBS 寄存器都会复位。

## RTCChecksum (RTC 校验和寄存器)

<b>RTCChecksum</b> (RTC 校验和寄存器)			基地址: <b>0x4000C000</b> 偏移地址: <b>98H</b>					
	<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>	RTCCHK[31:24]							
<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>	RTCCHK[23:16]							
<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>	RTCCHK[15:8]							
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	1	1	0	1	0	1	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	RTCCHK[7:0]							
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	1	0	0	0	0

位	功能描述
RTCCHK[31:0]	$RTCChecksum = DFAH + DFAL + DFBH + DFBL + DFCH + DFCL + DFDH + DFDL + DFEH + DFEL + Toff + MCON01 + MCON23 + MCON45$ 注：所有被累积寄存器按 16 位无符号数累加,结果保留 32 位无符号数

### CTRLByFlash（自加载结果寄存器）

CTRLByFlash (自加载结果寄存器)		基地址: 0x4000C000 偏移地址: 100H						
	Bit31	30	29	28	27	26	25	Bit24
Read:	X	X	X	X	X			
Write:					X			
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	sramRDflh[3:0]				WDT_EN[3:0]			
Write:	X	X	X	X	X	X	X	X
Reset:	1	1	1	1	1	1	1	1
	Bit15	14	13	12	11	10	9	Bit8
Read:	FLASH[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	1	1	1	1	1	1	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	X		X		RTCRST	AUTOR ELOAD	LRC_CT RL	Reserved
Write:						X		
Reset:	1	0	1	0	0	0	1	0

位	功能描述
sramRDflh[3:0]	sramRDflh[3:0] =1111 时，程序在 SRAM 空间运行,对 Flash main block 读取不加密 sramRDflh[3:0] !=1111时，程序在SRAM空间运行，对Flash main block读取加密，读取值为固定55555555；此时Flash擦写状态如下： 1) SRAM加密打开，此时通过SWD口对0-8K flash不可页擦和写（用户程序可页擦和写0-8K），但可进行全擦操作，可以页擦和写9K-255K。 2) 不通过SWD接口，而通过用户程序，即使SRAM加密有打开，0-255K也可以全擦、页擦和写。
WDT_EN[3:0]	Sleep 和 Hold 模式看门狗使能位 =0101: 看门狗在 Sleep 和 Hold 模式下关闭 =其他: 看门狗在 Sleep 和 Hold 模式下开启
FLASH[7:0]	Flash 加密位

	=0xFF: Flash 不加密 =其他: Flash 加密
RTCST	RTC 复位控制位 =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
AUTORELOAD	RTC 补偿系数装载使能位 =1: RTC 补偿系数装载功能使能 =0: RTC 补偿系数装载功能屏蔽
LRC_CTRL	LRC 使能位 =1: 打开 LRC =0: 关闭 LRC
Reserved	可读写, 无功能

## AUTOLOADSTA (自加载状态寄存器)

AUTOLOADSTA (自加载状态寄存器)			基地址: 0x4000C00 偏移地址: 108H					
	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	AUTOL OADSU CIF
Write:								X
Reset:	0	0	0	0	0	0	0	1

位	功能描述
AUTOLOADSU CIF	自加载成功标志位 0: 失败 1: 成功 (default) 注: AUTOLOADSUCIF 置 1, 表示自动装载成功即加载值与配置值一致。

注: AUTOLOADSTA 为状态寄存器, 因此不受新增功能写保护控制位和新增功能生效控制位的影响。

## RTCTESTCTRL (RTC 测试控制寄存器)

RTCTESTCTRL (RTC 测试控制寄存器)	基地址: 0x4000C00 偏移地址: 10CH
------------------------------	------------------------------

	Bit15	14	13	12	11	10	9	Bit8
Read:	RTCTEST_WP[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	FLT_OSC_VRTC[1:0]		Reserved	Reserved	Reserved	Reserved
Write:								
Reset:	0	0	0	0	1	0	0	0

位	功能描述
RTCTEST_WP[7:0]	RTC 测试控制寄存器写保护控制位 =0xA5,写保护关闭,可写,读返回值为0. =其他值,写保护开启,不可写。(default)
Reserved Bit[7:6]	请用户保持默认值 00b
FLT_OSC_VRTC[1:0]	VRTC 电压域下 OSC 时钟滤波控制位 =00, 0ns(default) =01, 1.5ns =10, 3.0ns =11, 4.5ns
Reserved Bit[3:0]	请用户保持默认值 1000b

## CALENDARCLR (万年历校准清零寄存器)

CALENDARCLR (万年历校准清零寄存器)		基地址: 0x4000C000 偏移地址: 110H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	CALENDARCLR_WP[7:0]							
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	WRCLDCLR
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CALENDARCLR_WP[7:0]	万年历校准清零寄存器低 8 位写保护控制位 = 0xA5,写保护关闭,可写,读返回值为0 =其他,写保护开启,不可写 (default)
WRCLDCLR	万年历校准清零控制位 =0: 关闭(default) =1: 开启

注：只有该位使能后，RTC 校准日历写入后，RTC 从校准时刻正常累加。  
该位关闭时，保持默认校准功能，即写入万年历后，下次跳秒才生效。

注：操作 CALENDARCLR 寄存器需要 CALENDARCLR\_WP[7:0]写入 0xA5。例如 WRCLDCLR =1 操作，CALENDARCLR=0xA501,写入成功后，CALENDARCLR 寄存器值为 0x0001。

## RTCTS0CON (RTC 时间戳 0 控制寄存器)

RTCTS0CON (RTC 时间戳 0 控制寄存器)		基地址: 0x4000C000 偏移地址: 300H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	TSCLR	X	Reserved	X	X	X	TSOFILT[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	TSOPOL[1:0]		X	TSOCMP EN	X	TSOEN
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TSCLR	时间戳事件计数及缓存清零控制 写 1 清零 RTCTSCNT, RTCTSTIM[i], RTCTSDAT[i], RTCTSSOU[i] 寄存器，清零结束硬件清零此标志位。
TSOFILT[1:0]	<b>STAMP0 数字滤波选择</b> 00:512clk (default) 01:4096clk 10:16384clk 11:32768clk
TSOPOL[1:0]	时间戳触发源 STAMP0 触发沿选择 00: 上升沿触发 01: 下降沿触发 10: 双沿触发 11: 双沿触发
TSOCMPEN	时间戳引脚 STAMP0 计数比较功能使能 0: 关闭 1: 开启
TSOEN	RTC 时间戳引脚 STAMP0 使能控制 0: 关闭 1: 开启

**RTCTS0CMP (RTC 时间戳 0 事件比较寄存器)**

<b>RTCTS0CMP</b> (RTC 时间戳 0 事件比较寄存器)			基地址: 0x4000C000 偏移地址: 304H					
	<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>	TS0CMP[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
TS0CMP[7:0]	RTC 时间戳引脚 STAMP0 触发事件比较寄存器

**RTCTS0CNT (RTC 时间戳 0 事件计数寄存器)**

<b>RTCTS0CNT</b> (RTC 时间戳 0 事件计数寄存器)			基地址: 0x4000C000 偏移地址: 308H					
	<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>	TS0CNT[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
TS0CNT[7:0]	RTC 时间戳引脚 STAMP0 触发事件计数器, 只读

注: 寄存器溢出后置起标志位, 并从 0 开始重新计数。

**RTCTS0IE (RTC 时间戳 0 中断使能寄存器)**

<b>RTCTS0IE</b>	基地址: 0x4000C000
-----------------	-----------------

(RTC 时间戳 0 中断使能寄存器)			偏移地址: 30CH					
	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	TS0OVI	X	TS0CMP	X	TS0IE
Write:				E		IE		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TS0OVIE	RTC 时间戳事件 STAMP0 计数溢出中断使能位 0: 关闭 1: 打开
TS0CMPIE	RTC 时间戳事件 STAMP0 计数比较中断使能位 0: 关闭 1: 打开
TS0IE	RTC 时间戳 STAMP0 触发中断使能位 0: 关闭 1: 打开

### RTCTS0IF (RTC 时间戳 0 中断标志寄存器)

RTCTS0IF (RTC 时间戳 0 中断标志寄存器)			基地址: 0x4000C000 偏移地址: 310H					
	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	TS0OVI	X	TS0CMP	X	TS0IF
Write:				F		IF		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TS0OVIF	RTC 时间戳事件 STAMP0 计数溢出中断标志位 0: 未发生中断 1: 发生中断 (写 0 清 0)
TS0CMPIF	RTC 时间戳事件 STAMP0 计数比较中断标志位

	0: 未发生中断 1: 发生中断 (写 0 清 0)
TSOIF	<b>RTC 时间戳 STAMP0 触发中断标志位</b> 0: 未发生中断 1: 发生中断 (写 0 清 0)

### RTCTS0TIM0 (RTC 时间戳 0 时间缓存寄存器 0)

<b>RTCTS0TIM0</b> (RTC 时间戳 0 时间缓存寄存器 0)			基地址: 0x4000C000 偏移地址: 314H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	HOUR[4:0]				
Write:				X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	MIN[5:0]					
Write:			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	SEC[5:0]					
Write:			X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	时计数器缓存: 范围: 0-23。只读
MIN[5:0]	分计数器缓存: 范围: 0-59。只读
SEC[5:0]	秒计数器缓存: 范围: 0-59。只读

### RTCTS0DAT0 (RTC 时间戳 0 日期缓存寄存器 0)

<b>RTCTS0DAT0</b> (RTC 时间戳 0 日期缓存寄存器 0)			基地址: 0x4000C000 偏移地址: 318H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	YEAR[6:0]						

<b>Write:</b>		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>	X	X	X	X	MONTH[3:0]			
<b>Write:</b>					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	DAY[4:0]				
<b>Write:</b>				X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	年计数器缓存： 范围：0-99。只读
MONTH[3:0]	月计数器缓存： 范围：1-12。只读
DAY[4:0]	日计数器缓存： 范围：1-28/29/30/31。只读

### RTCTS0SOU0(RTC 时间戳 0 触发源缓存寄存器 0)

<b>RTCTS0SOU0</b> (RTC 时间戳 0 触发源缓存寄存器 0)			基地址： 0x4000C000 偏移地址： 31CH					
	<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	X
<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>	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	SPOL	Reserved
<b>Write:</b>							X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
<b>SPOL</b>	当前事件触发源极性（只读）： 0：上升沿 1：下降沿

**RTCTS0TIM1 (RTC 时间戳 0 时间缓存寄存器 1)**

<b>RTCTS0TIM1</b> (RTC 时间戳 0 时间缓存寄存器 1)			基地址: <b>0x4000C000</b> 偏移地址: <b>320H</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	HOUR[4:0]				
<b>Write:</b>				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>	X	X	MIN[5:0]					
<b>Write:</b>			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	SEC[5:0]					
<b>Write:</b>			X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	时计数器缓存: 范围: 0-23。只读
MIN[5:0]	分计数器缓存: 范围: 0-59。只读
SEC[5:0]	秒计数器缓存: 范围: 0-59。只读

**RTCTS0DAT1 (RTC 时间戳 0 日期缓存寄存器 1)**

<b>RTCTS0DAT1</b> (RTC 时间戳 0 日期缓存寄存器 1)			基地址: <b>0x4000C000</b> 偏移地址: <b>324H</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	YEAR[6:0]						
<b>Write:</b>		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>	X	X	X	X	MONTH[3:0]			

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

位	功能描述
YEAR[6:0]	年计数器缓存： 范围：0-99。只读
MONTH[3:0]	月计数器缓存： 范围：1-12。只读
DAY[4:0]	日计数器缓存： 范围：1-28/29/30/31。只读

### RTCTS0SOU1(RTC 时间戳 0 触发源缓存寄存器 1)

<b>RTCTS0SOU1</b> (RTC 时间戳 0 触发源缓存寄存器 1)			基地址： 0x4000C000 偏移地址： 328H					
	<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	X
<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>	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	SPOL	Reserved
<b>Write:</b>							X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
<b>SPOL</b>	当前事件触发源极性（只读）： 0: 上升沿 1: 下降沿

**RTCTS0TIM2 (RTC 时间戳 0 时间缓存寄存器 2)**

<b>RTCTS0TIM2</b> (RTC 时间戳 0 时间缓存寄存器 2)			基地址: 0x4000C000 偏移地址: 32CH					
	<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	HOUR[4:0]				
<b>Write:</b>				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>	X	X	MIN[5:0]					
<b>Write:</b>			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	SEC[5:0]					
<b>Write:</b>			X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	时计数器缓存: 范围: 0-23。只读
MIN[5:0]	分计数器缓存: 范围: 0-59。只读
SEC[5:0]	秒计数器缓存: 范围: 0-59。只读

**RTCTS0DAT2 (RTC 时间戳 0 日期缓存寄存器 2)**

<b>RTCTS0DAT2</b> (RTC 时间戳 0 日期缓存寄存器 2)			基地址: 0x4000C000 偏移地址: 330H					
	<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	YEAR[6:0]						
<b>Write:</b>		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>	X	X	X	X	MONTH[3:0]			
<b>Write:</b>					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	DAY[4:0]				
<b>Write:</b>				X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	年计数器缓存: 范围: 0-99。只读
MONTH[3:0]	月计数器缓存: 范围: 1-12。只读
DAY[4:0]	日计数器缓存: 范围: 1-28/29/30/31。只读

### RTCTS0SOU2(RTC 时间戳 0 触发源缓存寄存器 2)

<b>RTCTS0SOU2</b> (RTC 时间戳 0 触发源缓存寄存器 2)			基地址: 0x4000C000 偏移地址: 334H					
	Bit23	22	21	20	19	18	17	Bit16
<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
	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	SPOL	Reserved
<b>Write:</b>							X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
SPOL	当前事件触发源极性 (只读): 0: 上升沿 1: 下降沿

### RTCTS0TIM3 (RTC 时间戳 0 时间缓存寄存器 3)

<b>RTCTS0TIM3</b> (RTC 时间戳 0 时间缓存寄存器 3)		基地址: 0x4000C000 偏移地址: 338H	
--	--	-------------------------------	--

器 3)								
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	HOUR[4:0]				
Write:				X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	MIN[5:0]					
Write:			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	SEC[5:0]					
Write:			X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	时计数器缓存： 范围：0-23。只读
MIN[5:0]	分计数器缓存： 范围：0-59。只读
SEC[5:0]	秒计数器缓存： 范围：0-59。只读

### RTCTS0DAT3 (RTC 时间戳 0 日期缓存寄存器 3)

RTCTS0DAT3 (RTC 时间戳 0 日期缓存寄存器 3)			基地址： 0x4000C000 偏移地址： 33CH					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	YEAR[6:0]						
Write:		X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	MONTH[3:0]			
Write:					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	DAY[4:0]				
Write:				X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
---	------

YEAR[6:0]	年计数器缓存： 范围：0-99。只读
MONTH[3:0]	月计数器缓存： 范围：1-12。只读
DAY[4:0]	日计数器缓存： 范围：1-28/29/30/31。只读

### RTCTS0SOU3(RTC 时间戳 0 触发源缓存寄存器 3)

<b>RTCTS0SOU3</b> (RTC 时间戳 0 触发源缓存寄存器 3)			基地址： 0x4000C000 偏移地址： 340H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	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	SPOL	Reserved
Write:							X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SPOL	当前事件触发源极性（只读）： 0：上升沿 1：下降沿

### RTCTS0TIM4 (RTC 时间戳 0 时间缓存寄存器 4)

<b>RTCTS0TIM4</b> (RTC 时间戳 0 时间缓存寄存器 4)			基地址： 0x4000C000 偏移地址： 344H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	HOUR[4:0]				
Write:				X	X	X	X	X
Reset:	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>	X	X	MIN[5:0]					
<b>Write:</b>			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	SEC[5:0]					
<b>Write:</b>			X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	时计数器缓存： 范围：0-23。只读
MIN[5:0]	分计数器缓存： 范围：0-59。只读
SEC[5:0]	秒计数器缓存： 范围：0-59。只读

### RTCTS0DAT4 (RTC 时间戳 0 日期缓存寄存器 4)

<b>RTCTS0DAT4</b> (RTC 时间戳 0 日期缓存寄存器 4)			基地址： 0x4000C000 偏移地址： 348H					
	<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	YEAR[6:0]						
<b>Write:</b>		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>	X	X	X	X	MONTH[3:0]			
<b>Write:</b>					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	DAY[4:0]				
<b>Write:</b>				X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	年计数器缓存： 范围：0-99。只读
MONTH[3:0]	月计数器缓存： 范围：1-12。只读
DAY[4:0]	日计数器缓存：

范围：1-28/29/30/31。只读

**RTCTS0SOU4(RTC 时间戳 0 触发源缓存寄存器 4)**

<b>RTCTS0SOU4</b> (RTC 时间戳 0 触发源缓存寄存器 4)			基地址： 0x4000C000 偏移地址： 34CH					
	<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	X
<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>	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	SPOL	Reserved
<b>Write:</b>							X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
<b>SPOL</b>	当前事件触发源极性（只读）： 0：上升沿 1：下降沿

**RTCTS0TIM5（RTC 时间戳 0 时间缓存寄存器 5）**

<b>RTCTS0TIM5</b> (RTC 时间戳 0 时间缓存寄存器 5)			基地址： 0x4000C000 偏移地址： 350H					
	<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	HOUR[4:0]				
<b>Write:</b>				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>	X	X	MIN[5:0]					
<b>Write:</b>			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	SEC[5:0]					
<b>Write:</b>			X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	时计数器缓存： 范围：0-23。只读
MIN[5:0]	分计数器缓存： 范围：0-59。只读
SEC[5:0]	秒计数器缓存： 范围：0-59。只读

### RTCTS0DAT5 (RTC 时间戳 0 日期缓存寄存器 5)

<b>RTCTS0DAT5</b> (RTC 时间戳 0 日期缓存寄存器 5)			基地址： 0x4000C000 偏移地址： 354H					
	<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	YEAR[6:0]						
<b>Write:</b>		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>	X	X	X	X	MONTH[3:0]			
<b>Write:</b>					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	DAY[4:0]				
<b>Write:</b>				X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	年计数器缓存： 范围：0-99。只读
MONTH[3:0]	月计数器缓存： 范围：1-12。只读
DAY[4:0]	日计数器缓存： 范围：1-28/29/30/31。只读

**RTCTS0SOU5 (RTC 时间戳 0 触发源缓存寄存器 5)**

<b>RTCTS0SOU5</b> (RTC 时间戳 0 触发源缓存寄存器 5)			基地址: <b>0x4000C000</b> 偏移地址: <b>358H</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	X
<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>	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	SPOL	Reserved
<b>Write:</b>							X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
SPOL	当前事件触发源极性 (只读): 0: 上升沿 1: 下降沿

**RTCTS1CON (RTC 时间戳 1 控制寄存器)**

<b>RTCTS1CON</b> (RTC 时间戳 1 控制寄存器)			基地址: <b>0x4000C000</b> 偏移地址: <b>400H</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>	TSCLR	X	Reserved	X	X	X	TS1FILT[1:0]	
<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	TS1POL[1:0]		X	TS1CMP EN	X	TS1EN
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
TSCLR	时间戳事件计数及缓存清零控制 写 1 清零 RTCTSCNT, RTCTSTIM[i], RTCTSDAT[i], RTCTSSOU[i] 寄存器, 清

	零结束硬件清零此标志位。
TS1FILT[1:0]	<b>STAMP1 数字滤波选择</b> 00: 512clk 01: 4096clk 10:16384clk 11:32768clk
TS1POL[1:0]	<b>时间戳触发源 STAMP1 触发沿选择</b> 00: 上升沿触发 01: 下降沿触发 10: 双沿触发 11: 双沿触发
TS1CMPEN	<b>时间戳引脚 STAMP1 计数比较功能使能</b> 0: 关闭 1: 开启
TS1EN	<b>RTC 时间戳引脚 STAMP1 使能控制</b> 0: 关闭 1: 开启

### RTCTS1CMP (RTC 时间戳 1 事件比较寄存器)

<b>RTCTS1CMP</b> (RTC 时间戳 1 事件比较寄存器)			<b>基地址: 0x4000C000</b> <b>偏移地址: 404H</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>	TS1CMP[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
TS1CMP[7:0]	RTC 时间戳引脚 STAMP1 触发事件比较寄存器

### RTCTS1CNT (RTC 时间戳 1 事件计数寄存器)

<b>RTCTS1CNT</b> (RTC 时间戳 1 事件计数寄存器)	<b>基地址: 0x4000C000</b> <b>偏移地址: 408H</b>
---	---

器)								
	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:	TS1CNT[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TS1CNT[7:0]	RTC 时间戳引脚 STAMP1 触发事件计数器，只读

注：寄存器溢出后置起标志位，并从 0 开始重新计数。

### RTCTS1IE (RTC 时间戳 1 中断使能寄存器)

RTCTS1IE (RTC 时间戳 1 中断使能寄存器)			基地址: 0x4000C000 偏移地址: 40CH					
	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	TS1OVI	X	TS1CMP	X	TS1IE
Write:	X	X	X	E		IE		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TS1OVI	RTC 时间戳事件 STAMP1 计数溢出中断使能位 0: 关闭 1: 打开
TS1CMP	RTC 时间戳事件 STAMP1 计数比较中断使能位 0: 关闭 1: 打开
TS1IE	RTC 时间戳 STAMP1 触发中断使能位 0: 关闭 1: 打开

**RTCTS1IF (RTC 时间戳 1 中断标志寄存器)**

<b>RTCTS1IF</b> (RTC 时间戳 1 中断标志寄存器)			<b>基地址: 0x4000C000</b> <b>偏移地址: 410H</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	TS1OVI	X	TS1CMP	X	TS1IF
<b>Write:</b>				F		IF		
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
TS1OVI	<b>RTC 时间戳事件 STAMP1 计数溢出中断标志位</b> 0: 未发生中断 1: 发生中断 (写 0 清 0)
TS1CMPIF	<b>RTC 时间戳事件 STAMP1 计数比较中断标志位</b> 0: 未发生中断 1: 发生中断 (写 0 清 0)
TS1IF	<b>RTC 时间戳 STAMP1 触发中断标志位</b> 0: 未发生中断 1: 发生中断 (写 0 清 0)

**RTCTS1TIM0 (RTC 时间戳 1 时间缓存寄存器 0)**

<b>RTCTS1TIM0</b> (RTC 时间戳 1 时间缓存寄存器 0)			<b>基地址: 0x4000C000</b> <b>偏移地址: 414H</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	HOUR[4:0]				
<b>Write:</b>				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>	X	X	MIN[5:0]					
<b>Write:</b>			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	SEC[5:0]					

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

位	功能描述
HOUR[4:0]	时计数器缓存： 范围：0-23。只读
MIN[5:0]	分计数器缓存： 范围：0-59。只读
SEC[5:0]	秒计数器缓存： 范围：0-59。只读

### RTCTS1DAT0 (RTC 时间戳 1 日期缓存寄存器 0)

<b>RTCTS1DAT0</b> (RTC 时间戳 1 日期缓存寄存器 0)			<b>基地址： 0x400C000</b> <b>偏移地址： 418H</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	YEAR[6:0]						
<b>Write:</b>		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>	X	X	X	X	MONTH[3:0]			
<b>Write:</b>					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	DAY[4:0]				
<b>Write:</b>				X	X	X	X	X
<b>Reset:</b>				0	0	0	0	0

位	功能描述
YEAR[6:0]	年计数器缓存： 范围：0-99。只读
MONTH[3:0]	月计数器缓存： 范围：1-12。只读
DAY[4:0]	日计数器缓存： 范围：1-28/29/30/31。只读

**RTCTS1SOU0(RTC 时间戳 1 触发源缓存寄存器 0)**

<b>RTCTS1SOU0</b> (RTC 时间戳 1 触发源缓存寄存器 0)			<b>基地址: 0x4000C000</b> <b>偏移地址: 41CH</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	X
<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>	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	SPOL	Reserved
<b>Write:</b>							X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
<b>SPOL</b>	当前事件触发源极性 (只读): 0: 上升沿 1: 下降沿

**RTCTS1TIM1 (RTC 时间戳 1 时间缓存寄存器 1)**

<b>RTCTS1TIM1</b> (RTC 时间戳 1 时间缓存寄存器 1)			<b>基地址: 0x4000C000</b> <b>偏移地址: 420H</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	HOUR[4:0]				
<b>Write:</b>				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>	X	X	MIN[5:0]					
<b>Write:</b>			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	SEC[5:0]					
<b>Write:</b>			X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	时计数器缓存： 范围：0-23。只读
MIN[5:0]	分计数器缓存： 范围：0-59。只读
SEC[5:0]	秒计数器缓存： 范围：0-59。只读

### RTCTS1DAT1 (RTC 时间戳 1 日期缓存寄存器 1)

<b>RTCTS1DAT1</b> (RTC 时间戳 1 日期缓存寄存器 1)		基地址： 0x4000C000 偏移地址： 424H						
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	YEAR[6:0]						
Write:		X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	MONTH[3:0]			
Write:					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	DAY[4:0]				
Write:				X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	年计数器缓存： 范围：0-99。只读
MONTH[3:0]	月计数器缓存： 范围：1-12。只读
DAY[4:0]	日计数器缓存： 范围：1-28/29/30/31。只读

### RTCTS1SOU1(RTC 时间戳 1 触发源缓存寄存器 1)

<b>RTCTS1SOU1</b> (RTC 时间戳 1 触发源缓存寄存器 1)	基地址： 0x4000C000 偏移地址： 428H
---	-------------------------------

器 1)								
	Bit23	22	21	20	19	18	17	Bit16
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
	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	SPOL	Reserved
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SPOL	当前事件触发源极性（只读）： 0: 上升沿 1: 下降沿

## RTCTS1TIM2（RTC 时间戳 1 时间缓存寄存器 2）

RTCTS1TIM2 （RTC 时间戳 1 时间缓存寄存器 2）			基地址： 0x4000C000 偏移地址： 42CH					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	HOUR[4:0]				
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	MIN[5:0]					
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	SEC[5:0]					
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	时计数器缓存： 范围：0-23。只读
MIN[5:0]	分计数器缓存：

	范围：0-59。只读
SEC[5:0]	秒计数器缓存： 范围：0-59。只读

## RTCTS1DAT2 (RTC 时间戳 1 日期缓存寄存器 2)

<b>RTCTS1DAT2</b> (RTC 时间戳 1 日期缓存寄存器 2)			<b>基地址： 0x4000C000</b> <b>偏移地址： 430H</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	YEAR[6:0]						
<b>Write:</b>		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>	X	X	X	X	MONTH[3:0]			
<b>Write:</b>					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	DAY[4:0]				
<b>Write:</b>				X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	年计数器缓存： 范围：0-99。只读
MONTH[3:0]	月计数器缓存： 范围：1-12。只读
DAY[4:0]	日计数器缓存： 范围：1-28/29/30/31。只读

## RTCTS1SOU2(RTC 时间戳 1 触发源缓存寄存器 2)

<b>RTCTS1SOU2</b> (RTC 时间戳 1 触发源缓存寄存器 2)			<b>基地址： 0x4000C000</b> <b>偏移地址： 434H</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	X
<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>	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	SPOL	Reserved
<b>Write:</b>							X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
<b>SPOL</b>	当前事件触发源极性（只读）： 0: 上升沿 1: 下降沿

### RTCTS1TIM3（RTC 时间戳 1 时间缓存寄存器 3）

<b>RTCTS1TIM3</b> (RTC 时间戳 1 时间缓存寄存器 3)			<b>基地址: 0x4000C000</b> <b>偏移地址: 438H</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	HOUR[4:0]				
<b>Write:</b>				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>	X	X	MIN[5:0]					
<b>Write:</b>			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	SEC[5:0]					
<b>Write:</b>			X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	时计数器缓存： 范围：0-23。只读
MIN[5:0]	分计数器缓存： 范围：0-59。只读
SEC[5:0]	秒计数器缓存： 范围：0-59。只读

### RTCTS1DAT3

<b>RTCTS1DAT3</b> (RTC 时间戳 1 日期缓存寄存器 3)			<b>基地址: 0x4000C000</b> <b>偏移地址: 43CH</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	YEAR[6:0]						
<b>Write:</b>		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>	X	X	X	X	MONTH[3:0]			
<b>Write:</b>					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	DAY[4:0]				
<b>Write:</b>				X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	年计数器缓存: 范围: 0-99。只读
MONTH[3:0]	月计数器缓存: 范围: 1-12。只读
DAY[4:0]	日计数器缓存: 范围: 1-28/29/30/31。只读

### RTCTS1SOU3(RTC 时间戳 1 触发源缓存寄存器 3)

<b>RTCTS1SOU3</b> (RTC 时间戳 1 触发源缓存寄存器 3)			<b>基地址: 0x4000C000</b> <b>偏移地址: 440H</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	X
<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>	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	SPOL	Reserved
<b>Write:</b>							X	X

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

位	功能描述
<b>SPOL</b>	当前事件触发源极性（只读）： 0: 上升沿 1: 下降沿

### RTCTS1TIM4（RTC 时间戳 1 时间缓存寄存器 4）

<b>RTCTS1TIM4</b> (RTC 时间戳 1 时间缓存寄存器 4)			<b>基地址: 0x4000C000</b> <b>偏移地址: 444H</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	HOUR[4:0]				
<b>Write:</b>				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>	X	X	MIN[5:0]					
<b>Write:</b>			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	SEC[5:0]					
<b>Write:</b>			X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	时计数器缓存： 范围：0-23。只读
MIN[5:0]	分计数器缓存： 范围：0-59。只读
SEC[5:0]	秒计数器缓存： 范围：0-59。只读

### RTCTS1DAT4（RTC 时间戳 1 日期缓存寄存器 4）

<b>RTCTS1DAT4</b> (RTC 时间戳 1 日期缓存寄存器 4)			<b>基地址: 0x4000C000</b> <b>偏移地址: 448H</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	YEAR[6:0]						
<b>Write:</b>		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>	X	X	X	X	MONTH[3:0]			
<b>Write:</b>					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	DAY[4:0]				
<b>Write:</b>				X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	年计数器缓存： 范围：0-99。只读
MONTH[3:0]	月计数器缓存： 范围：1-12。只读
DAY[4:0]	日计数器缓存： 范围：1-28/29/30/31。只读

### RTCTS1SOU4(RTC 时间戳 1 触发源缓存寄存器 4)

<b>RTCTS1SOU4</b> (RTC 时间戳 1 触发源缓存寄存器 4)			<b>基地址: 0x4000C000</b> <b>偏移地址: 44CH</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	X
<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>	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	SPOL	Reserved
<b>Write:</b>							X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
<b>SPOL</b>	当前事件触发源极性（只读）： 0：上升沿

1: 下降沿

**RTCTS1TIM5 (RTC 时间戳 1 时间缓存寄存器 5)**

<b>RTCTS1TIM5</b> (RTC 时间戳 1 时间缓存寄存器 5)			<b>基地址: 0x4000C000</b> <b>偏移地址: 450H</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	HOUR[4:0]				
<b>Write:</b>				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>	X	X	MIN[5:0]					
<b>Write:</b>			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	SEC[5:0]					
<b>Write:</b>			X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HOUR[4:0]	时计数器缓存: 范围: 0-23。只读
MIN[5:0]	分计数器缓存: 范围: 0-59。只读
SEC[5:0]	秒计数器缓存: 范围: 0-59。只读

**RTCTS1DAT5(RTC 时间戳 1 触发源缓存寄存器 5)**

<b>RTCTS1DAT5</b> (RTC 时间戳 1 日期缓存寄存器 5)			<b>基地址: 0x4000C000</b> <b>偏移地址: 454H</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	YEAR[6:0]						
<b>Write:</b>		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>	X	X	X	X	MONTH[3:0]			
<b>Write:</b>					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	DAY[4:0]				
<b>Write:</b>				X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	年计数器缓存： 范围：0-99。只读
MONTH[3:0]	月计数器缓存： 范围：1-12。只读
DAY[4:0]	日计数器缓存： 范围：1-28/29/30/31。只读

### RTCTS1SOU5(RTC 时间戳 1 触发源缓存寄存器 5)

<b>RTCTS1SOU5</b> (RTC 时间戳 1 触发源缓存寄存器 5)			<b>基地址： 0x4000C000</b> <b>偏移地址： 458H</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	X
<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>	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	SPOL	Reserved
<b>Write:</b>							X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
<b>SPOL</b>	当前事件触发源极性（只读）： 0：上升沿 1：下降沿

**RTCBACKUP<sub>x</sub>(x =0~31)** (第 x 个 RTC 数据备份寄存器)

<b>RTCBACKUP<sub>x</sub>(x =0~31)</b> (第 x 个 RTC 数据备份寄存器)		基地址: <b>0x4000C000</b> 偏移地址: <b>E00H+04H*x</b>						
	<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>	<b>RTCBACKUP<sub>x</sub>[31:24]</b>							
<b>Write:</b>								
<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>	<b>RTCBACKUP<sub>x</sub>[23:16]</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>	<b>RTCBACKUP<sub>x</sub>[15:8]</b>							
<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>	<b>RTCBACKUP<sub>x</sub>[7:0]</b>							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
<b>RTCBACKUP<sub>x</sub>[31:0]</b>	第 x 个 RTC 数据备份寄存器 (x=0~31)

## 16 TBS 模块

### 16.1 概述

芯片可以对环境温度（TPS）、电源电压（VCC）、EMU 模块 VREF（TDVREF）以及 5 路外部 ADC（ADCIN0/1/3/4/5）进行定量的测量，并将测量结果保存到相应的寄存器中。

芯片包含 8 路可以调节 PGA 的外部测温端子通道（VTP0/1/2/3/4/5/6/7），与测温端子配合使用，可以测量电能表端子温度，并将测试结果保存到相应的寄存器中。用户可以根据测量结果监测电能表端子的温度。

### 16.2 功能描述

#### 16.2.1 基本功能

- 测量 IC 基底的温度（TPS）
- 测量电源电压（VCC）
- 测量 EMU 模块 VREF（TDVREF）
- 测量 5 路 ADC 电压（ADCIN0/1/3/4/5）
- 测量电能表端子温度(VTP0/1/2/3/4/5/6/7)

## 16.2.2 模块结构框图

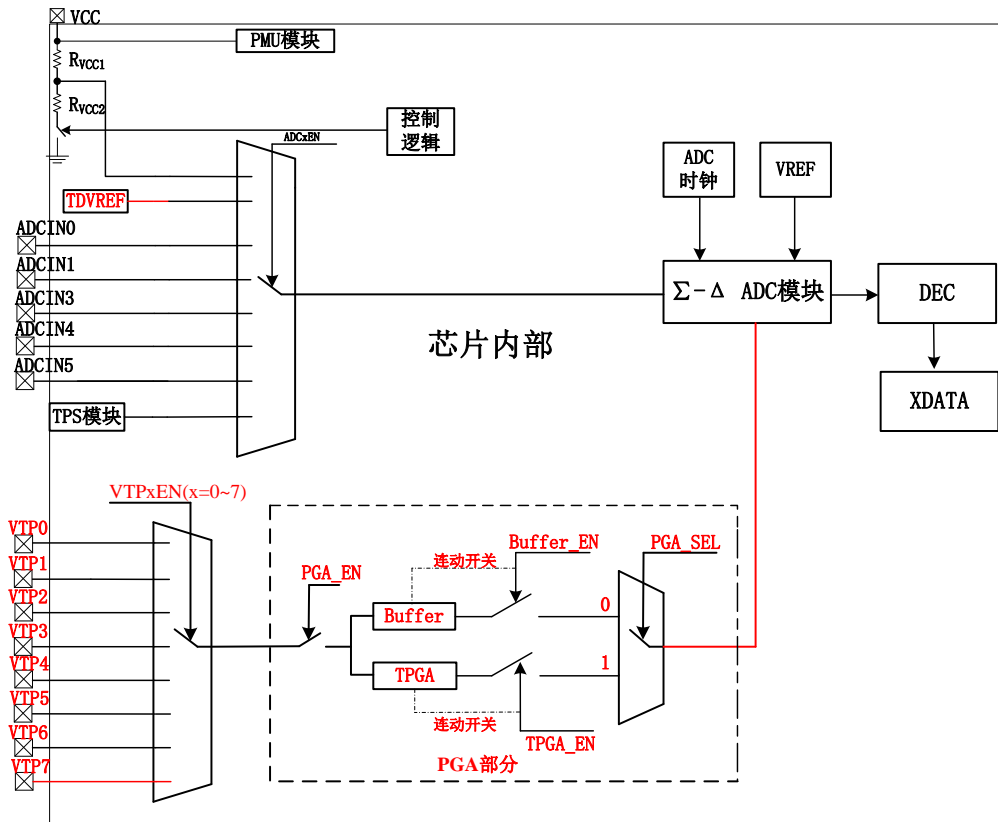


图 1.1 TBS 模块结构框图

ChipNum	VCC 通道			
	R <sub>vcc1</sub>	R <sub>vcc2</sub>	R <sub>vcc1</sub> : R <sub>vcc2</sub>	R <sub>vcc1</sub> + R <sub>vcc2</sub>
	37.5K	6.25K	6:1	43.75K

注：（1）VCC 检测内阻约为 43.75K，ADCIN0/1/3/4/5 内部均无电阻分压；

（2）IC 的 ADC 通道的输入阻抗约为 8.5M@32k；

（3）IC 的 VTP 通道的输入阻抗无穷大；

（4）TDVREF 通道采样时，ADC 模块内部会自动做分压处理；

（5）Buffer\_EN 和 TPGA\_EN 的开关取决于 PGA\_SEL 的选择，两者不会同时闭合。当选择 Buffer 时，TPGA\_EN 断开，选择 TPGA 时，Buffer\_EN 断开。

## 16.2.3 模块时钟

TBS 模块时钟包括 Fosc、Flrc 或 PLLX2 三种时钟选择源，由 TBS 模块下 TBSCON 寄存器工作时钟源选择控制位 TBSCLK\_SEL 以及 CMU 模块下 LFCLKCFG 寄存器 TBS 模块时钟配置位 TBSLFSSEL 共同决定。

- Fosc 时钟 (default)
- Flrc 时钟

- PLLX2 分频时钟
- TBSLFSEL=0, TBS 模块时钟由 TBSCLK\_SEL 决定
  - 当 TBSCLK\_SEL = 0 时, TBS 模块时钟选择为 Fosc 时钟; TBSCLK\_SEL = 1 时, TBS 模块时钟选择为 PLLX2 时钟的 150 分频, 时钟频率为 262.144KHz。
- TBSLFSEL=1, TBS 模块时钟选择为 Flrc 时钟, TBSCLK\_SEL 控制无效

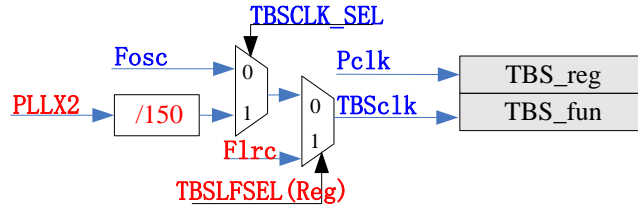


图 1.2 TBS 模块时钟框图

## 16.3 工作模式

TBS 模块包括以下三种工作模式:

- 固定分时开启模式
- 实时触发模式
- 注入工作模式

### 16.3.1 固定分时开启模式

为了降低 TBS 模块工作时的功耗, TBS 模块工作时, 默认配置为固定分时开启模式, 该模式的特点如下:

- 优先级最低, 实时触发或注入模式开启均会打断固定分时开启逻辑;
- 各通道按照固定时序分时开启, 一个采样周期内, 各通道开启时刻固定不变;
- 分时开启时间固定

### 16.3.2 实时触发模式

为了提高 ADC 的采样速率, 可将待测通道配置为实时触发功能。

- 实施触发优先级高于固定分时开启, 低于注入工作模式
- 实时触发操作方式: 向实时触发控制寄存器 bit[15:0]软件置 1, 代表发送实时触发请求, 实时触发被响应且数据转换完成此位自动清 0。

### 16.3.3 注入工作模式

为了满足更高速采样的应用需求, 增加注入工作模式, 从以下方面提高 ADC 采样转换速率。

- 提高 ADC 采样时钟速率

ADC 采样时钟源选择 PLLX2 时钟时，可通过 INJECTCLK\_SEL[2:0]分频得到，ADC 的采样时钟速率设置范围为 153.6K~1228.8KHz。

- 增加通道强制固定开启

注入工作模式下，可选择强制 ADC 的某通道一直处于开启状态，每个 OSR 可输出采样数据，以提高采样转换速率。

#### 使用说明：

1. 以上提高 ADC 采样速率的方式，仅当进入注入工作模式才有效；
2. 注入工作模式下，开启 TPS 测量时，TPS 输出滑动滤波控制有效；
3. 注入工作模式开启后，ADC 前 3 个采样转换数据无效，第 4 个数据才有效；
4. 固定分时开启、实时触发和注入模式三者之间说明
  - a) 注入模式进入方式：TBS 注入模式控制寄存器的 INJECTMODE\_EN 置 1，才可进入 ADC 注入模式；
  - b) 若分时开启、实时触发和注入模式同时开启，注入模式优先级最高，优先被响应；需要等到退出注入模式后，固定分时开启和实时触发模式才有效，实时触发优先级高于固定分时开启，需要等到实时触发转换完成后，才继续响应固定分时开启转换。

5. 注入工作模式的时钟选择是由以下几个控制位共同决定的，具体详见下表：

TBSLF SEL	TBSCLK_SEL	INJECTMODE _EN	INJECTCLK _EN	INJECTCLK_SEL [2:0]	TBS 时钟频率 (KHz)
1	X	X	X	X	Flrc
0	0	X	X	X	Fosc=32.768
0	1	0	X	X	262
0	1	1	0	X (REG=100)	246(default)
0	1	1	1	REG	PLLX2/(32*(REG+1))

## 16.4 测温端子功能说明

为了满足电能表具有电压与电流端子铜条温度监测功能，根据温度值实施报警或跳闸等策略，确保用电安全。增加 8 路测温端子专用通道，与测温端子配合使用，实时监测电能表端子温度。

### 16.4.1 功能说明

测温端子包括以下基本功能：

- 8 路测温端子通道可选
- 2 种 PGA 增益可选
- TPGA 档位可配置

- 独立的中断使能控制位与标志位，共用 TBS 中断源
- 8 路测温端子通道具有独立的数据寄存器
- 仅支持固定分时开启和实时触发模式

## 16.4.2 功能指标

端子温度监测应满足要求：

1. +25 摄氏度-+150 摄氏度之间的测量误差不超过±5 摄氏度；
2. 任一个测温端子两次测温之间的时间间隔不超过 5 秒

## 16.5 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	0x10006541	TBS设置寄存器
04H	TBSIE	R/W	0x0000	TBS中断使能寄存器
08H	TBSIF	R/W	0x0001	TBS中断标志寄存器
0CH	TMPDAT	R	0xEED4	温度测量输出寄存器
10H	TDVREFDAT	R	0x0000	EMU模块VREF测试输出寄存器
14H	ADC0DAT	R	0x0000	ADC通道0测量输出寄存器
18H	ADC1DAT	R	0x0000	ADC通道1测量输出寄存器
1CH	TDVREFCMP	R/W	0x0000	EMU模块VREF比较寄存器
20H	TBSPRD	R/W	0x000000	TBS打开频率设置寄存器
24H	Reserved	R/W	0x00010200	保留寄存器，请保持为默认值
28H	VCCDAT	R	0x0000	电源电压测量输出寄存器
2CH	ADC0CMP	R/W	0x0000	ADC通道0比较寄存器
30H	ADC3DAT	R	0x0000	ADC通道3测量输出寄存器
34H	ADC4DAT	R	0x0000	ADC通道4测量输出寄存器
38H	ADC5DAT	R	0x0000	ADC通道5测量输出寄存器
40H	TRIREQ	R/W	0x0000	TBS实时触发控制寄存器
48H	TBSINJECTCTRL	R/W	0x0400	TBS注入模式控制寄存器
4CH	VTPPRD	R/W	0x0000	测温端子打开频率设置寄存器
50H	VTP0DAT	R	0x0000	测温端子通道0测量输出寄存器
54H	VTP1DAT	R	0x0000	测温端子通道1测量输出寄存器
58H	VTP2DAT	R	0x0000	测温端子通道2测量输出寄存器
5CH	VTP3DAT	R	0x0000	测温端子通道3测量输出寄存器
60H	VTP4DAT	R	0x0000	测温端子通道4测量输出寄存器

64H	VTP5DAT	R	0x0000	测温端子通道5测量输出寄存器
68H	VTP6DAT	R	0x0000	测温端子通道6测量输出寄存器
6CH	VTP7DAT	R	0x0000	测温端子通道7测量输出寄存器

## 16.6 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						
	Bit31	30	29	28	27	26	25	Bit24
Read:	PGA_EN	PGA_SE L	TPGA[1:0]		X	X	VTP7EN	VTP6EN
Write:								
Reset:	0	0	0	1	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	VTP5EN	VTP4EN	VTP3EN	VTP2E N	VTP1EN	VTP0EN	ADC5E N	ADC4E N
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADC3E N	Reserved	Reserved	Reserve d	Reserved		Filter[1:0]	
Write:								
Reset:	0	1	1	0	0	1	0	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	Reserved bit[7:6]		TBSCLK_ SEL	VCCEN	ADC1E N	ADC0E N	TDVREF EN	TMPEN
Write:								
Reset:	0	1	0	0	0	0	0	1

位	功能描述
PGA_EN	<b>PGA 模块使能控制位</b> 0: PGA 模块关闭 (default) 1: PGA 模块打开 注: PGA_EN 为 PGA 部分总开关
PGA_SEL	<b>PGA 选择控制位</b> 0: Buffer (default) 1: TPGA 注: (1) 该位需与PGA_EN配合使用, 当PGA_EN=1时, 该位配置才有意义。 (2) Buffer_EN/TPGA_EN与PGA_SEL是绑定的, 当PGA_SEL=0, Buffer_EN同时开启; 当PGA_SEL=1, TPGA_EN同时开启。 (3) PGA_SEL=0, 选择Buffer, gain=1; (适用于0.3V~2.4V输入)

PGA_SEL=1, 选择 TPGA, gain=4; (适用于 0~0.3V 输入)	
TPGA[1:0]	<b>测温端子 TPGA 档位选择位</b> 00: 2 倍 01: 4 倍 (default) 10: 8 倍 11: 16 倍 注: 该控制位需与 PGA_SEL 控制位配合使用, 仅当 PGA_SEL=1 时, 选择 TPGA 时该位配置才有意义。
VTP7EN	<b>测温端子通道 7 功能控制位</b> 0: 关闭 (default) 1: 使能
VTP6EN	<b>测温端子通道 6 功能控制位</b> 0: 关闭 (default) 1: 使能
VTP5EN	<b>测温端子通道 5 功能控制位</b> 0: 关闭 (default) 1: 使能
VTP4EN	<b>测温端子通道 4 功能控制位</b> 0: 关闭 (default) 1: 使能
VTP3EN	<b>测温端子通道 3 功能控制位</b> 0: 关闭 (default) 1: 使能
VTP2EN	<b>测温端子通道 2 功能控制位</b> 0: 关闭 (default) 1: 使能
VTP1EN	<b>测温端子通道 1 功能控制位</b> 0: 关闭 (default) 1: 使能
VTP0EN	<b>测温端子通道 0 功能控制位</b> 0: 关闭 (default) 1: 使能
ADC5EN	<b>ADC 通道 5 测量功能控制位:</b> 0: 关闭 (default) 1: 使能 注: 如使用该功能, 用户需在 GPIO 模块将对应 PIN 配置为 ADCIN5 才行
ADC4EN	<b>ADC 通道 4 测量功能控制位:</b> 0: 关闭 (default) 1: 使能 注: 如使用该功能, 用户需在 GPIO 模块将对应 PIN 配置为 ADCIN4 才行
ADC3EN	<b>ADC 通道 3 测量功能控制位:</b> 0: 关闭 (default) 1: 使能

	注：如使用该功能，用户需在 GPIO 模块将对应 PIN 配置为 ADCIN3 才行
Reserved	保留位，需保留为 1
Reserved	保留位，需保留为 1
Reserved	保留位，需保留为 0
Reserved	保留位，需保留为默认值
Filter[1:0]	<b>ADC 输出滤波控制位（平均次数，只针对温度测量）：</b> 00: 1 次（不推荐） 01: 2 次 (default) 10: 4 次 11: 8 次
Reserved bit[7:6]	<b>RESERVED</b> 请用户务必保持默认 bit 值配置：01b
TBSCLK_SEL	<b>TBS 工作时钟源选择控制位</b> 0: 选择 OSC 时钟 (default) 1: 选择 PLL 时钟的 <b>150</b> 分频
VCCEN	<b>VCC 电压测量功能控制位</b> 0: 关闭 (default) 1: 使能
ADC1EN	<b>ADC 通道 1 测量功能控制位：</b> 0: 关闭 (default) 1: 使能 注：如使用该功能，用户需在 GPIO 模块将对应 PIN 配置为 ADCIN1 才行
ADC0EN	<b>ADC 通道 0 测量功能控制位：</b> 0: 关闭 (default) 1: 使能 注：如使用该功能，用户需在 GPIO 模块将对应 PIN 配置为 ADCIN0 才行
TDVREFEN	<b>EMU 模块 VREF 测量功能控制位</b> 0: 关闭 (default) 1: 使能
TMPEN	<b>温度测量功能控制位：</b> 0: 关闭 1: 使能 (default)

## TBSIE（TBS 中断使能寄存器）

<b>TBSIE</b> （TBS 中断使能寄存器）			基地址： <b>0x4000E000</b> 偏移地址： <b>04H</b>					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	X	X	X	VTP7IE	VTP6IE
Write:								
Reset:	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>	VTP5IE	VTP4IE	VTP3IE	VTP2IE	VTP1IE	VTP0IE	ADC5IE	ADC4IE
<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>	ADC3IE	ADC0C MPIE	VCCIE	TDVREF CMPIE	ADC1IE	ADC0IE	TDVREF IE	TMPIE
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
VTP7IE	测温端子通道 7 测量中断使能位 0: 关闭 (default) 1: 打开
VTP6IE	测温端子通道 6 测量中断使能位 0: 关闭 (default) 1: 打开
VTP5IE	测温端子通道 5 测量中断使能位 0: 关闭 (default) 1: 打开
VTP4IE	测温端子通道 4 测量中断使能位 0: 关闭 (default) 1: 打开
VTP3IE	测温端子通道 3 测量中断使能位 0: 关闭 (default) 1: 打开
VTP2IE	测温端子通道 2 测量中断使能位 0: 关闭 (default) 1: 打开
VTP1IE	测温端子通道 1 测量中断使能位 0: 关闭 (default) 1: 打开
VTP0IE	测温端子通道 0 测量中断使能位 0: 关闭 (default) 1: 打开
ADC5IE	ADC 通道 5 测量中断使能位 0: 关闭 (default) 1: 打开
ADC4IE	ADC 通道 4 测量中断使能位 0: 关闭 (default) 1: 打开
ADC3IE	ADC 通道 3 测量中断使能位 0: 关闭 (default) 1: 打开

ADC0CMPIE	<b>ADC0 比较中断使能位</b> 0: 关闭 (default) 1: 打开 注: 当 ADC0DAT 大于等于 ADC0CMP 时将产生中断, 此功能只在使能了 ADC0 测量(ADC0En)时才有效
VCCIE	<b>电源电压测量中断使能位</b> 0: 关闭 (default) 1: 打开
TDVREFCMPIE	<b>TDVREF 比较中使能位</b> 0: 中断关闭 (default) 1: 中断打开 注: 当 TDVREFDAT 小于 TDVREFCMP 时将产生中断, 此功能只在使能了 EMU 模块 VREF 测量(TDVREFEN)才有效
ADC1IE	<b>ADC 通道 1 测量中断使能位</b> 0: 关闭 (default) 1: 打开
ADC0IE	<b>ADC 通道 0 测量中断使能位</b> 0: 关闭 (default) 1: 打开
TDVREFIE	<b>EMU 模块 VREF 测量中断使能位</b> 0: 关闭 (default) 1: 打开
TMPIE	<b>温度测量中断使能位</b> 0: 关闭 (default) 1: 打开

## TBSIF (TBS 中断标志寄存器)

TBSIF (TBS 中断标志寄存器)			基地址: 0x4000E000 偏移地址: 08H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	X	X	X	VTP7IF	VTP6IF
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	VTP5IF	VTP4IF	VTP3IF	VTP2IF	VTP1IF	VTP0IF	ADC5IF	ADC4IF
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADC3IF	ADC0C MPIF	VCCIF	TDVREF CMPIF	ADC1IF	ADC0IF	TDVREF IF	TMPIF
Write:								

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

位	功能描述
VTP7IF	测温端子通道 7 测量中断标志位 0: 未产生中断 (default) 1: 产生中断
VTP6IF	测温端子通道 6 测量中断标志位 0: 未产生中断 (default) 1: 产生中断
VTP5IF	测温端子通道 5 测量中断标志位 0: 未产生中断 (default) 1: 产生中断
VTP4IF	测温端子通道 4 测量中断标志位 0: 未产生中断 (default) 1: 产生中断
VTP3IF	测温端子通道 3 测量中断标志位 0: 未产生中断 (default) 1: 产生中断
VTP2IF	测温端子通道 2 测量中断标志位 0: 未产生中断 (default) 1: 产生中断
VTP1IF	测温端子通道 1 测量中断标志位 0: 未产生中断 (default) 1: 产生中断
VTP0IF	测温端子通道 0 测量中断标志位 0: 未产生中断 (default) 1: 产生中断
ADC5IF	ADC 通道 5 测量中断标志位 0: 未产生中断 (default) 1: 产生中断
ADC4IF	ADC 通道 4 测量中断标志位 0: 未产生中断 (default) 1: 产生中断
ADC3IF	ADC 通道 3 测量中断标志位 0: 未产生中断 (default) 1: 产生中断
ADC0CMPIF	ADC0 比较中断标志位 0: 未产生中断 (default) 1: 产生中断 注: 当 ADC0DAT 大于等于 ADC0CMP 时将产生标志
VCCIF	电源电压测量中断标志位 0: 未产生中断 (default) 1: 产生中断

<b>TDVREFCMPPIF</b>	<b>TDVREF 比较中断标志位</b> 0: 未产生中断 (default) 1: 产生中断 注: <b>TDVREFDAT</b> 小于 <b>TDVREFCMP</b> 时产生的中断标志
<b>ADC1IF</b>	<b>ADC 通道 1 测量中断标志位</b> 0: 未产生中断 (default) 1: 产生中断
<b>ADC0IF</b>	<b>ADC 通道 0 测量中断标志位</b> 0: 未产生中断 (default) 1: 产生中断
<b>TDVREFIF</b>	<b>EMU 模块 VREF 测量中断标志位</b> 0: 未产生中断 (default) 1: 产生中断
<b>TMPIF</b>	<b>温度测量中断标志位</b> 0: 未产生中断 1: 产生中断 (default)

## TMPDAT (温度测量输出寄存器)

<b>TMPDAT</b> (温度测量输出寄存器)			基地址: <b>0x4000E000</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>	TMPDAT[15:8]							
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	1	1	1	0	1	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>	TMPDAT[7:0]							
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	1	1	0	1	0	1	0	0

位	功能描述
TMPDAT[15:0]	温度测量输出寄存器, 16 位有符号数 注:随着芯片供电降低, 内部信号 <b>TMP_2VFLAG</b> 由 1 变为 0 时, <b>TMPDAT</b> 保持原有值不变。

温度计算公式:

$$\text{温度 } Tr = 12.9852 - \text{TMPDAT} * 0.002828$$

其中:  $Tr$ 为实际的温度 (°C)

**TDVREFDAT (EMU 模块 VREF 测量输出寄存器)**

<b>TDVREFDAT</b> (EMU 模块 VREF 测量输出寄存器)			基地址: <b>0x4000E000</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>	TDVREFDAT[15:8]							
<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>	TDVREFDAT[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
TDVREFDAT [15:0]	EMU 模块 VREF 测量输出寄存器, 16 位有符号数。

**TDVREF** 计算公式:

$$TDVREF = 0.04959 * TDVREFDAT - 20.99917$$

TDVREF 为实测量电压 (mv)

**ADC0DAT (ADC 通道 0 测量输出寄存器)**

<b>ADC0DAT</b> (ADC 通道 0 测量输出寄存器)			基地址: <b>0x4000E000</b> 偏移地址: <b>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>	ADC0DAT[15:8]							
<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>	ADC0DAT[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
ADC0DAT[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:

$$VADCIN0 = 0.02461 * ADC0DAT + 0.11918;$$

其中: VADCIN0 为实际 ADC 测量电压 (mV)

**ADC1DAT (ADC 通道 1 测量输出寄存器)**

<b>ADC1DAT</b> (ADC 通道 1 测量输出寄存器)			基地址: <b>0x4000E000</b> 偏移地址: <b>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>	ADC1DAT[15:8]							
<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>	ADC1DAT[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
ADC1DAT[15:0]	ADC 通道 1 测量输出寄存器, 16 位有符号数

ADC 通道 1 测量电压公式同 ADC 通道 0。

**TDCVREFCMP (EMU 模块 VREF 比较寄存器)**

<b>TDVREFCMP</b> (EMU 模块 VREF 比较寄存器)			基地址: <b>0x4000E000</b> 偏移地址: <b>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>	<b>TDVREFCMP[15:8]</b>							
<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>	<b>TDVREFCMP[7:0]</b>							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
<b>TDVREFCMP</b> [15:0]	<b>TDVREF</b> 比较寄存器, 16 位有符号数。 注: 当 <b>TDVREFDAT</b> 小于 <b>TDVREFCMP</b> 时将产生中断, 此功能只在使能了 <b>EMU 模块 VREF 测量(TDVREFEN)</b> 时才有效

**TBSPRD (TBS 打开频率设置寄存器)**

TBSPRD (TBS 打开频率设置寄存器)			基地址: 0x4000E000 偏移地址: 20H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	X	X	X	X	ADC5PR D[1]
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADC5PR D[0]	ADC4PRD[1:0]		ADC3PRD[1:0]		VCCPRD[1:0]		ADC1PR D[1]
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADC1PR D[0]	ADC0PRD[1:0]		TDVERFPRD[1:0]		TMPPRD[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADC5PRD [1:0]	<b>ADC 通道 5 测量周期设置</b> 00: 1 秒 (default) 01: 2 秒 10: 8 秒 11: 16 秒
ADC4PRD [1:0]	<b>ADC 通道 4 测量周期设置</b> 00: 1 秒 (default) 01: 2 秒 10: 8 秒 11: 16 秒
ADC3PRD [1:0]	<b>ADC 通道 3 测量周期设置</b> 00: 1 秒 (default) 01: 2 秒 10: 8 秒 11: 16 秒
VCCPRD [1:0]	<b>电源电压测量周期设置</b> 00: 1 秒 (default) 01: 2 秒 10: 8 秒 11: 16 秒
ADC1PRD [1:0]	<b>ADC 通道 1 测量周期设置</b> 00: 1 秒 (default) 01: 2 秒 10: 8 秒

	11: 16 秒																		
ADC0PRD[1:0]	<b>ADC 通道 0 测量周期设置</b> 00: 1 秒 (default) 01: 2 秒 10: 8 秒 11: 16 秒																		
TDVERFPRD[1:0]	<b>EMU 模块 VREF 测量周期设置</b> 00: 1 秒 (default) 01: 2 秒 10: 8 秒 11: 16 秒																		
TMPPRD[2:0]	<b>温度测量周期设置</b> TPS 打开频率如下表所示: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>TMPPRD [2:0]</th> <th>TBS 打开周期(s)</th> </tr> </thead> <tbody> <tr><td>000</td><td>1/2 (default)</td></tr> <tr><td>001</td><td>1</td></tr> <tr><td>010</td><td>2</td></tr> <tr><td>011</td><td>4</td></tr> <tr><td>100</td><td>8</td></tr> <tr><td>101</td><td>16</td></tr> <tr><td>110</td><td>32</td></tr> <tr><td>111</td><td>64</td></tr> </tbody> </table>	TMPPRD [2:0]	TBS 打开周期(s)	000	1/2 (default)	001	1	010	2	011	4	100	8	101	16	110	32	111	64
TMPPRD [2:0]	TBS 打开周期(s)																		
000	1/2 (default)																		
001	1																		
010	2																		
011	4																		
100	8																		
101	16																		
110	32																		
111	64																		

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。

## TBSTEST (保留寄存器)

TBSTEST (保留寄存器)			基地址: 0x4000E000 偏移地址: 24H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	X	X	X	X	X	X	X	Reserved
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	Reserved[23:16]							
Write:	Reserved[23:16]							
Reset:	0	0	0	0	0	0	0	1
	Bit15	14	13	12	11	10	9	Bit8
Read:	Reserved[15:12]				X	Reserved[10:8]		
Write:	Reserved[15:12]					Reserved[10:8]		
Reset:	0	0	0	0	0	0	1	0
	Bit7	6	5	4	3	2	1	Bit0

Read:	Reserved[7:5]			X	X	Reserved[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Reserved	保留寄存器，该寄存器所有控制位请保持为默认值

## VCCDAT（电源电压测量输出寄存器）

<b>VCCDAT</b> (电源电压测量输出寄存器)			基地址: <b>0x4000E000</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>
Read:	VCCDAT[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
Read:	VCCDAT[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
VCCDAT[15:0]	电源电压测量输出寄存器，16位有符号数

VCC 电压计算公式:

$$VCC = 0.1785 * VCCDAT - 30.9252;$$

其中：VCC 为实际芯片电源电压（mV）

## ADC0CMP（ADC0 比较寄存器）

<b>ADC0CMP</b> (ADC0 比较寄存器)			基地址: <b>0x4000E000</b> 偏移地址: <b>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>
Read:	ADC0CMP[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
Read:	ADC0CMP[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADC0CMP[15:0]	ADC0 测量比较寄存器，16 位有符号数 注：当 ADC0DAT 大于等于 ADC0CMP 时将产生标志

### ADC3DAT（ADC3 测量输出寄存器）

ADC3DAT (ADC3 测量输出寄存器)		基地址: 0x4000E000 偏移地址: 30H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADC3DAT[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADC3DAT[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADC3DAT[15:0]	ADC3 测量输出寄存器，16 位有符号数

测量电压公式同 ADC 通道 0。

### ADC4DAT（ADC4 测量输出寄存器）

ADC4DAT (ADC4 测量输出寄存器)		基地址: 0x4000E000 偏移地址: 34H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADC4DAT[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADC4DAT[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADC4DAT[15:0]	ADC4 测量输出寄存器，16 位有符号数

测量电压公式同 ADC 通道 0。

## ADC5DAT (ADC5 测量输出寄存器)

ADC5DAT (ADC5 测量输出寄存器)			基地址: 0x4000E000 偏移地址: 38H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADC5DAT[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADC5DAT[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADC5DAT[15:0]	ADC5 测量输出寄存器, 16 位有符号数

测量电压公式同 ADC 通道 0。

## TRIREQ (TBS 实时触发控制寄存器)

TRIREQ (TBS 实时触发控制寄存器)			基地址: 0x4000E000 偏移地址: 40H					
	Bit15	14	13	12	11	10	9	Bit8
Write:	Vtp7tri	Vtp6tri	Vtp5tri	Vtp4tri	Vtp3tri	Vtp2tri	Vtp1tri	Vtp0tri
Read:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Write:	Adc5tri	Adc4tri	Adc3tri	vcctri	Adc1tri	Adc0tri	tdvreftri	tmptri
Read:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Vtp7tri	测温端子通道 7 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
Vtp6tri	测温端子通道 6 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效

Vtp5tri	测温端子通道 5 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
Vtp4tri	测温端子通道 4 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
Vtp3tri	测温端子通道 3 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
Vtp2tri	测温端子通道 2 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
Vtp1tri	测温端子通道 1 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
Vtp0tri	测温端子通道 0 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
Adc5tri	Adc5 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
Adc4tri	Adc4 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
Adc3tri	Adc3 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
vcctri	vcc 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
Adc1tri	Adc1 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
Adc0tri	Adc0 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
tdvreftri	EMU 模块 VREF 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效
tmptri	tmp 实时触发控制位 1: 实时触发请求, 转换完成自动清零 0: 此位写零无效

**TBSINJECTCTRL (TBS 注入模式控制寄存器)**

<b>TBSINJECTCTRL</b> (TBS 注入模式控制寄存器)			基地址: <b>0x4000E000</b> 偏移地址: <b>48H</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	INJECTCLK_EN	X	X	X	INJECTCLK_SEL[2:0]		
<b>Write:</b>								
<b>Reset:</b>	0	0	0	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>	X	INJECTCHEL_SEL[2:0]			X	X	X	INJECTMODE_EN
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
INJECTCLK_EN	注入模式分频时钟控制使能位 (仅 TBS 模块选择 PLLX2 时钟源有效) 0: 关闭 (default) 1: 打开
INJECTCLK_SEL [2:0]	注入模式时钟分频控制位 (仅 TBS 模块选择 PLL 时钟源有效) 时钟频率计算公式为: $INJECTCLK = PLL / (32 * (REG + 1))$ ; 其中 REG 可设置范围为: 000b~111b, <b>default 值为 100b</b> 注: TBSCLK_SEL、INJECTMODE_EN 和 INJECTCLK_EN 同时置 1 时, 注入模式分频时钟控制位才有效。
INJECTCHEL_SEL [2:0]	注入模式通道选择位 000: tmp (default) 001: <b>tdvref</b> 010: adc0 011: adc1 100: vcc 101: adc3 110: adc4 111: adc5
INJECTMODE_EN	ADC 注入模式使能控制位 0: 关闭 (default) 1: 打开

**VTPPRD (测温端子打开频率设置寄存器)**

<b>VTPPRD</b> (测温端子打开频率设置寄存	基地址: <b>0x4000E000</b> 偏移地址: <b>4CH</b>
--------------------------------	--

器)									
	Bit15	14	13	12	11	10	9	Bit8	
Read:	VTP7PRD[1:0]		VTP6PRD[1:0]		VTP5PRD[1:0]		VTP4PRD[1:0]		
Write:									
Reset:	0	0	0	0	0	0	0	0	
	Bit7	6	5	4	3	2	1	Bit0	
Read:	VTP3PRD[1:0]		VTP2PRD[1:0]		VTP1PRD[1:0]		VTP0PRD[1:0]		
Write:									
Reset:	0	0	0	0	0	0	0	0	

位	功能描述
VTP7PRD[1:0]	测温端子通道 7 测量周期设置 00: 1 秒 (default) 01: 2 秒 10: 8 秒 11: 16 秒
VTP6PRD[1:0]	测温端子通道 6 测量周期设置 00: 1 秒 (default) 01: 2 秒 10: 8 秒 11: 16 秒
VTP5PRD[1:0]	测温端子通道 5 测量周期设置 00: 1 秒 (default) 01: 2 秒 10: 8 秒 11: 16 秒
VTP4PRD[1:0]	测温端子通道 4 测量周期设置 00: 1 秒 (default) 01: 2 秒 10: 8 秒 11: 16 秒
VTP3PRD[1:0]	测温端子通道 3 测量周期设置 00: 1 秒 (default) 01: 2 秒 10: 8 秒 11: 16 秒
VTP2PRD[1:0]	测温端子通道 2 测量周期设置 00: 1 秒 (default) 01: 2 秒 10: 8 秒 11: 16 秒
VTP1PRD[1:0]	测温端子通道 1 测量周期设置 00: 1 秒 (default)

	01: 2秒 10: 8秒 11: 16秒
VTPOPRD[1:0]	测温端子通道 0 测量周期设置 00: 1秒 (default) 01: 2秒 10: 8秒 11: 16秒

### VTPODAT (测温端子通道 0 测量输出寄存器)

VTPODAT (测温端子通道 0 测量输出寄存器)			基地址: 0x4000E000 偏移地址: 50H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	VTPODAT[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	VTPODAT[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
VTPODAT[15:0]	测温端子通道 0 测量输出寄存器, 16 位有符号数

测温端子VTP0计算公式:

$$VTP0 = 0.09843 * VTPODAT + 2.55787$$

其中: VTP0为实际的电压值 (mv)

### VTP1DAT (测温端子通道 1 测量输出寄存器)

VTP1DAT (测温端子通道 1 测量输出寄存器)			基地址: 0x4000E000 偏移地址: 54H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	VTP1DAT[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	VTP1DAT[7:0]							
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>位</b>	<b>功能描述</b>
VTP1DAT[15:0]	测温端子通道 1 测量输出寄存器，16 位有符号数

测试公式同VTP0通道

### VTP2DAT（测温端子通道 2 测量输出寄存器）

VTP2DAT (测温端子通道 2 测量输出寄存器)	<b>基地址: 0x4000E000</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>	VTP2DAT[15:8]							
<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>	VTP2DAT[7:0]							
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>位</b>	<b>功能描述</b>
VTP2DAT[15:0]	测温端子通道 2 测量输出寄存器，16 位有符号数

测试公式同VTP0通道

### VTP3DAT（测温端子通道 3 测量输出寄存器）

VTP3DAT (测温端子通道 3 测量输出寄存器)	<b>基地址: 0x4000E000</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>	VTP3DAT[15:8]							
<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>	VTP3DAT[7:0]							
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>位</b>	<b>功能描述</b>
VTP3DAT[15:0]	测温端子通道 3 测量输出寄存器，16 位有符号数

测试公式同VTP0通道

### VTP4DAT（测温端子通道 4 测量输出寄存器）

VTP4DAT (测温端子通道 4 测量输出寄存器)			<b>基地址: 0x4000E000</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>	VTP4DAT[15:8]							
<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>	VTP4DAT[7:0]							
<b>Write:</b>	X	X	X	X	X	X	X	X
<b>Reset:</b>	0	0	0	0	0	0	0	0

<b>位</b>	<b>功能描述</b>
VTP4DAT[15:0]	测温端子通道 4 测量输出寄存器，16 位有符号数

测试公式同VTP0通道

### VTP5DAT（测温端子通道 5 测量输出寄存器）

VTP5DAT (测温端子通道 5 测量输出寄存器)			<b>基地址: 0x4000E000</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>	VTP5DAT[15:8]							
<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>	VTP5DAT[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
VTP5DAT[15:0]	测温端子通道 5 测量输出寄存器，16 位有符号数

测试公式同VTP0通道

### VTP6DAT（测温端子通道 6 测量输出寄存器）

VTP6DAT (测温端子通道 6 测量输出寄存器)			基地址: 0x4000E000 偏移地址: 68H					
	<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>	VTP6DAT[15:8]							
<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>	VTP6DAT[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
VTP6DAT[15:0]	测温端子通道 6 测量输出寄存器，16 位有符号数

测试公式同VTP0通道

### VTP7DAT（测温端子通道 7 测量输出寄存器）

VTP7DAT (测温端子通道 7 测量输出寄存器)			基地址: 0x4000E000 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>	VTP7DAT[15:8]							
<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>	VTP7DAT[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
VTP7DAT[15:0]	测温端子通道 7 测量输出寄存器，16 位有符号数

测试公式同VTP0通道

## 17 AES/GHASH/RAND/CRC 模块

### 17.1 AES 概述

AES 硬件加速器可以支持客户使用 AES 算法用于数据包的加密/解密，该 AES 算法与 FIPS PUB 197 兼容，同时密钥长度支持 128bit, 192bit, 256bit 三种形式，AES 硬件加速器运行在 19.66M 的 PLL 时钟频率下的加密/解密时间小于 10us。详见应用笔记。

### 17.2 GHASH 概述

提供了一个独立的硬件 Ghash 运算单元用于 AES-GCM 这种加密和认证模式，配合用户的软件，可以加速 GCM 算法中的身份认证过程。

### 17.3 RAND 随机数概述

RAND 模块符合 FIPS-140-2 标准，RAND 单独使用亚稳态真随机数发生器、单独使用低频采高频真随机数发生器或者采用两者共同工作，产生操作数使用异或的方式，对真随机数发生器所产生的真随机数序列使用异或链的方式进行处理，且使用以真随机数序列为种子的伪随机数发生器产生随机数与真随机数序列异或的方式，即使用异或周期序列的方式。

真随机数功能在信息安全系统中有很好的应用，其真随机数功能由 2 部分组成，真随机数发生器产生真随机序列，数字算法会对真随机序列进行后处理，通过这两个部分最终实现产生真随机数的功能。

### 17.4 CRC 概述

CRC 模块含有可编程的多项式设置，支持常用的若干 CRC 标准。

- 支持三个常用的多项式（CRCCON\_MODE[1:0]）
- 支持初始化预置值（CRCINIT）
- 支持输入数据每个字节按位反转（CRCCON\_RefIn）
- 支持异或输出前，整个数据按 CRC 数据宽度（Width）按位反转（CRCCON\_RefOut）
- 支持输出数据异或操作选择（CRCCON\_XorOut）
- 支持输入数据大小端选择（CRCCON\_ENDIAN）
- 接受三种字长的数据写入：8 位、16 位或 32 位
  - 8 位数据写入：1 cpu clk
  - 16 位数据写入：2 cpu clk
  - 32 位数据写入：4 cpu clk

支持参数模型:

Name	Width	Poly	Init	RefIn	RefOut	XorOut	Generator polynomial
CRC-16/IBM	16	0x8005	0x0000	TRUE	TRUE	0x0000	$x^{16}+x^{15}+x^2+1$
CRC-16/MAXIM	16	0x8005	0x0000	TRUE	TRUE	0xFFFF	$x^{16}+x^{15}+x^2+1$
CRC-16/USB	16	0x8005	0xFFFF	TRUE	TRUE	0xFFFF	$x^{16}+x^{15}+x^2+1$
CRC-16/MODBUS	16	0x8005	0xFFFF	TRUE	TRUE	0x0000	$x^{16}+x^{15}+x^2+1$
CRC-16/CCITT	16	0x1021	0x0000	TRUE	TRUE	0x0000	$x^{16}+x^{12}+x^5+1$
CRC-16/CCITT-FALSE	16	0x1021	0xFFFF	FALSE	FALSE	0x0000	$x^{16}+x^{12}+x^5+1$
<b>CRC-16/X25</b>	<b>16</b>	<b>0x1021</b>	<b>0xFFFF</b>	<b>TRUE</b>	<b>TRUE</b>	<b>0xFFFF</b>	<b><math>x^{16}+x^{12}+x^5+1</math></b>
CRC-16/XMODEM	16	0x1021	0x0000	FALSE	FALSE	0x0000	$x^{16}+x^{12}+x^5+1$
CRC-32	32	0x04C11DB7	0xFFFFFFF	TRUE	TRUE	0xFFFFFFF	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$
CRC-32/MPEG-2	32	0x04C11DB7	0xFFFFFFF	FALSE	FALSE	0x0000	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

注: **Width:** 数据宽度

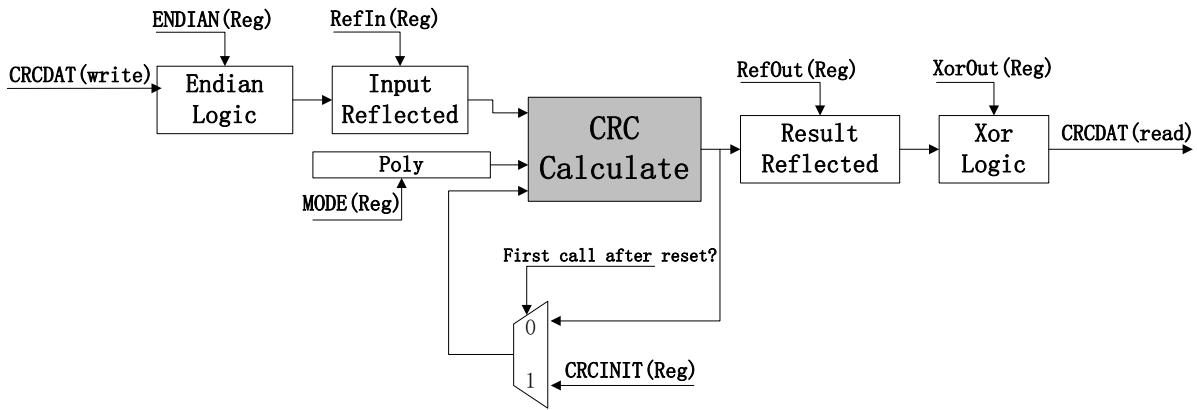
**Poly:** 多项式生成项的简写, 以 16 进制表示。例: CRC-16/25 生成多项式为  $x^{16}+x^{12}+x^5+1$ , 完整的生成项是 0x11021, 忽略最高位的"1", 多项式简写为 0x1021

**Init:** 初始化预置值

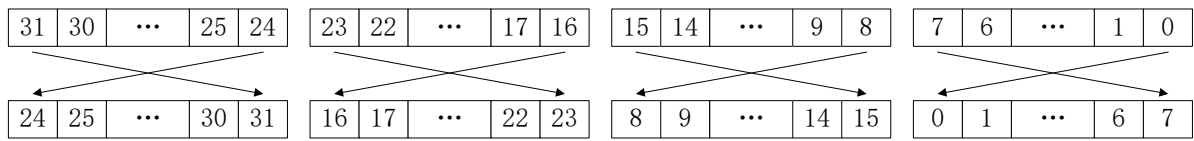
**RefIn:** 输入数据每个字节按位颠倒, TRUE 表示每个字节计算前先颠倒, 即 bit0 为最高位, bit7 为最低位; FALSE 表示不颠倒

**RefOut:** 输出数据顺序颠倒, TRUE 表示计算结束后, 计算结果先颠倒, 再进入 XorOut 处理; FALSE 表示不颠倒。(注: 输出数据在 Width 宽度上按位颠倒)

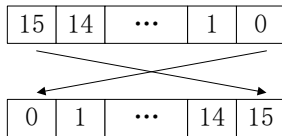
**XorOut:** 与所选模式位宽 (Width) 相同的值 (全零或全 1), 全 1 表示 CRCCON\_XorOut 置 1, 此时结果按位取反; 全 0 表示 CRCCON\_XorOut 清 0, 此时结果不操作



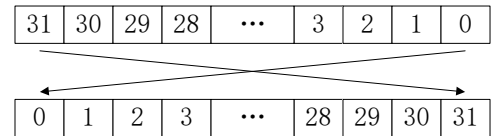
CRC 计算流程图



RefIn 处理示意图



RefOut 处理示意图 1: MODE=0 或 MODE=1



RefOut 处理示意图 2: MODE=2 或 MODE=3

## 17.5 特殊功能寄存器列表

AES 模块寄存器基地址: 0x40012000				
偏移地址	名称	读写方式	复位值	功能描述
00H	AESCON	R/W	0000H	AES配置寄存器
04H	AESSTR	W	0000H	AES启动命令寄存器
08H	AESFLG	R	0000H	AES标志寄存器
0CH	AESINLL	R/W	00000000H	AES输入待加/解密数据
10H	AESINML	R/W	00000000H	AES输入待加/解密数据
14H	AESINHM	R/W	00000000H	AES输入待加/解密数据
18H	AESINHH	R/W	00000000H	AES输入待加/解密数据
1CH	AESOUTLL	R	00000000H	AES输出已加/解密数据
20H	AESOUTML	R	00000000H	AES输出已加/解密数据
24H	AESOUTHM	R	00000000H	AES输出已加/解密数据
28H	AESOUTHH	R	00000000H	AES输出已加/解密数据
2CH	AESKEY0	R/W	00000000H	AES密钥寄存器
30H	AESKEY1	R/W	00000000H	AES密钥寄存器

34H	AESKEY2	R/W	00000000H	AES密钥寄存器
38H	AESKEY3	R/W	00000000H	AES密钥寄存器
3CH	AESKEY4	R/W	00000000H	AES密钥寄存器
40H	AESKEY5	R/W	00000000H	AES密钥寄存器
44H	AESKEY6	R/W	00000000H	AES密钥寄存器
48H	AESKEY7	R/W	00000000H	AES密钥寄存器

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	0000H	真随机数启动命令寄存器
04H	RANDDAT	R/W	00000000H	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
0CH	INPUT1ML	R/W	0x00000000	GHASH输入数据1
10H	INPUT1HM	R/W	0x00000000	GHASH输入数据1
14H	INPUT1HH	R/W	0x00000000	GHASH输入数据1
18H	INPUT2LL	R/W	0x00000000	GHASH输入数据2
1CH	INPUT2ML	R/W	0x00000000	GHASH输入数据2
20H	INPUT2HM	R/W	0x00000000	GHASH输入数据2
24H	INPUT2HH	R/W	0x00000000	GHASH输入数据2
28H	OUTPUTLL	R	0x00000000	GHASH输出数据
2CH	OUTPUTML	R	0x00000000	GHASH输出数据
30H	OUTPUTHM	R	0x00000000	GHASH输出数据
34H	OUTPUTHH	R	0x00000000	GHASH输出数据
38H	AESGHASHIE	R/W	0x00	AES,Ghash中断使能位
3CH	AESGHASHIF	R/W	0x00	AES,Ghash中断标志位

CRC 模块寄存器基地址: 0x40021000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	CRCCON	R/W	0x0000	CRC 控制寄存器
0x04	CRCDAT	R/W	0x0000FFFF	CRC 数据寄存器
0x08	CRCINIT	R/W	0x0000FFFF	CRC 初始化种子寄存器

## 17.6 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]	密钥长度选择: 00: AES-128, 密钥为 128bits(KEY0—KEY3)KEY0 为高位, KEY3 为低位 (default) 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	编码/解码模式选择控制: 0: 加密 (default) 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	14	13	12	11	10	9	Bit8
Read:	X							
Write:	DAT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X							
Write:	DAT[7:0]							
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 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。 该控制寄存器只能写入，不能读取。该寄存器读出永远为 0
-----------	--

## 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	<b>AES 加密/解密状态位:</b> 0: AES 加密/解密完成 1: AES 加密/解密未完成 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</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>	DAT[31:24]							
<b>Write:</b>								
<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>	DAT[23:16]							
<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>	DAT[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>	DAT[7:0]							
<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> (AES 数据输入中低位)		<b>基地址: 0x40012000</b> <b>偏移地址: 10H</b>						
	<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>	DAT[31:24]							
<b>Write:</b>								
<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>	DAT[23:16]							
<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>	DAT[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>	DAT[7:0]							
<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>基地址: 0x40012000</b>
----------------	------------------------

(AES 数据输入中高位)			偏移地址: 14H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

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

### AESINHH (AES 数据输入高位)

AESINHH (AES 数据输入高位)			基地址: 0x40012000 偏移地址: 18H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

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

### AESOUTLL (AES 数据输出低位)

AESOUTLL (AES 数据输出低位)			基地址: 0x40012000 偏移地址: 1CH					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0

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

### AESOUTML (AES 数据输出中低位)

AESOUTML (AES 数据输出中低位)			基地址: 0x40012000 偏移地址: 20H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:	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>	DAT[15:8]							
<b>Write:</b>	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>	DAT[7:0]							
<b>Write:</b>	X							
<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: 0x40012000</b>					
			<b>偏移地址: 24H</b>					
	<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>	DAT[31:24]							
<b>Write:</b>	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>	DAT[23:16]							
<b>Write:</b>	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>	DAT[15:8]							
<b>Write:</b>	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>	DAT[7:0]							
<b>Write:</b>	X							
<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)**

AESOUTHH (AES 数据输出高位)			基地址: 0x40012000 偏移地址: 28H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0

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

**AESKEY0 (AES 密钥 0)**

AESKEY0 (AES 密钥 0)			基地址: 0x40012000 偏移地址: 2CH					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0

	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	DAT[7:0]							
<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: 0x40012000</b> <b>偏移地址: 30H</b>					
	<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>	DAT[31:24]							
<b>Write:</b>								
<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>	DAT[23:16]							
<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>	DAT[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>	DAT[7:0]							
<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>	<b>基地址: 0x40012000</b>
----------------	------------------------

(AES 密钥 2)			偏移地址: 34H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:	DAT[31:24]							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:	DAT[23:16]							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:	DAT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							
Write:	DAT[7:0]							
Reset:	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)

AESKEY3 (AES 密钥 3)			基地址: 0x40012000 偏移地址: 38H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:	DAT[31:24]							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:	DAT[23:16]							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:	DAT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							

<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 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</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>	DAT[31:24]							
<b>Write:</b>	DAT[31:24]							
<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>	DAT[23:16]							
<b>Write:</b>	DAT[23:16]							
<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>	DAT[15:8]							
<b>Write:</b>	DAT[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>	DAT[7:0]							
<b>Write:</b>	DAT[7:0]							
<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: 0x40012000</b> <b>偏移地址: 40H</b>					
	<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>	DAT[31:24]							

<b>Write:</b>								
<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>	DAT[23:16]							
<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>	DAT[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>	DAT[7:0]							
<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 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)

AESKEY6 (AES 密钥 6)			基地址: 0x40012000 偏移地址: 44H					
	<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>	DAT[31:24]							
<b>Write:</b>								
<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>	DAT[23:16]							
<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>	DAT[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>	DAT[7:0]							
<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 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: 0x40012000</b>						
		<b>偏移地址: 48H</b>						
	<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>	DAT[31:24]							
<b>Write:</b>								
<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>	DAT[23:16]							
<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>	DAT[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>	DAT[7:0]							
<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 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)

<b>RANDSTR</b> (随机数启动控制寄存器)		<b>基地址: 0x40012080</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>	RANDE N	X	X	X	X	BACKE N	BACKMODE[1:0]	
<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	Start	Reserved	
<b>Write:</b>								
<b>Reset:</b>	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 (default) 1: 随机数模块使能 当 RANDEN 使能之后, 对 Start 写 1 才有效
BACKEN	随机数后处理使能 0: 处理关闭 (default) 1: 处理使能 (必须打开)
BACKMODE[1:0]	随机数后处理模式选择 00: LFSR 方式 (default) 01: 伪随机混合方式 10: 上面两种方式的和 11: 上面两种方式的和
Start	启动控制位 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
Reserved	

## 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	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:	DAT[31:24]							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:	DAT[23:16]							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:	DAT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							
Write:	DAT[7:0]							
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)

<b>GHASHSTR</b> (GHASH 启动寄存器)		基地址: 0x40012100 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							
<b>Write:</b>	DAT[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							
<b>Write:</b>	DAT[7:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

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

### GHASHFLG (GHASH 标志寄存器)

<b>GHASHFLG</b> (GHASH 标志寄存器)		基地址: 0x40012100 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	BUSY
<b>Write:</b>								X
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
BUSY	乘法运算状态位 0: GHASH 乘法运算完成 1: GHASH 乘法运算未完成 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	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

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

**INPUT1ML (数据 1 输入中低位)**

INPUT1ML (数据 1 输入中低位)			基地址: 0x40012100 偏移地址: 0CH					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8

<b>Read:</b>	DAT[15:8]							
<b>Write:</b>	DAT[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>	DAT[7:0]							
<b>Write:</b>	DAT[7:0]							
<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 bit32-bit63

### INPUT1HM (数据 1 输入中高位)

<b>INPUT1HM</b> (数据 1 输入中高位)		<b>基地址: 0x40012100</b> <b>偏移地址: 10H</b>						
	<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>	DAT[31:24]							
<b>Write:</b>	DAT[31:24]							
<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>	DAT[23:16]							
<b>Write:</b>	DAT[23:16]							
<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>	DAT[15:8]							
<b>Write:</b>	DAT[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>	DAT[7:0]							
<b>Write:</b>	DAT[7:0]							
<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

### INPUT1HH (数据 1 输入高位)

<b>INPUT1HH</b> (数据 1 输入高位)		<b>基地址: 0x40012100</b> <b>偏移地址: 14H</b>						
--------------------------------	--	--	--	--	--	--	--	--

	<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>	DAT[31:24]							
<b>Write:</b>								
<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>	DAT[23:16]							
<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>	DAT[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>	DAT[7:0]							
<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: 0x40012100</b> <b>偏移地址: 18H</b>						
	<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>	DAT[31:24]							
<b>Write:</b>								
<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>	DAT[23:16]							
<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>	DAT[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>	DAT[7:0]							
<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 bit0-bit31

### INPUT2ML (数据 2 输入中低位)

INPUT2ML (数据 2 输入中低位)		基地址: 0x40012100 偏移地址: 1CH						
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

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

### INPUT2HM (数据 2 输入中高位)

INPUT2HM (数据 2 输入中高位)		基地址: 0x40012100 偏移地址: 20H						
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:								
Reset:	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>	DAT[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>	DAT[7:0]							
<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 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</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>	DAT[31:24]							
<b>Write:</b>								
<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>	DAT[23:16]							
<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>	DAT[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>	DAT[7:0]							
<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

## OUTPUTLL (数据输出低位)

<b>OUTPUTLL</b>	<b>基地址: 0x40012100</b>
-----------------	------------------------

(数据输出低位)			偏移地址: 28H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0

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

## OUTPUTML (数据输出中低位)

OUTPUTML (数据输出中低位)			基地址: 0x40012100 偏移地址: 2CH					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

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

### OUTPUTHM (数据输出中高位)

OUTPUTHM (数据输出中高位)			基地址: 0x40012100 偏移地址: 30H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0

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

### OUTPUTHH (数据输出高位)

OUTPUTHH (数据输出高位)			基地址: 0x40012100 偏移地址: 34H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:	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>	DAT[15:8]							
<b>Write:</b>	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>	DAT[7:0]							
<b>Write:</b>	X							
<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

### AESGASHIE (AESGHASH 中断使能)

<b>AESGASHIE</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>								
<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	GHASHIE	AESIE
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
RANDIE	<b>随机数中断使能</b> 0: 中断关闭 (default) 1: 中断使能
GHASHIE	<b>GHASH 计算中断使能</b> 0: 中断关闭 (default) 1: 中断使能
AESIE	<b>AES 计算中断使能</b> 0: 中断关闭 (default) 1: 中断使能

**AESGHASHIF (AESGHASH 中断标志)**

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

位	功能描述
RANDIF	<b>随机数中断标志</b> 0: 中断未产生 1: 中断产生
GHASHIF	<b>GHASH 计算中断标志</b> 0: 中断未产生 1: 中断产生
AESIF	<b>AES 计算中断标志</b> 0: 中断未产生 1: 中断产生

**CRC CON (控制寄存器)**

CRC CON (控制寄存器)		基地址: 0x40021000 偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	ENDIAN
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	XorOut	RefOut	RefIn	X	X	MODE[1:0]	
Write:						RESET		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ENDIAN	<b>CRC 计算大小端选择位</b> 0: 输入数据为大端模式 (default)

	<b>1: 输入数据为小端模式</b>												
XorOut	<b>输出数据按位异或控制</b> 0: 输出数据不操作 (default) 1: 输出数据按位取反操作												
RefOut	<b>输出数据顺序颠倒控制</b> 0: 输出数据顺序不颠倒 (default) 1: 输出数据顺序颠倒, 按所选 CRC 模式数据宽度按位颠倒												
RefIn	<b>输入数据顺序颠倒控制</b> 0: 输入数据顺序不颠倒 (default) 1: 输入数据每个字节按位颠倒												
RESET	<b>CRC计算RESET位</b> 1: 复位CRC计算, 并根据当前模式设置CRCDAT寄存器 此位写1不会改变当前模式 只能对该位写‘1’, 由硬件自动清零。												
MODE[1:0]	<b>CRC计算多项式选择位</b> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="2">MODE[1:0]</th> <th>生成多项式 (简写)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Poly:0x1021 (default)</td> </tr> <tr> <td>0</td> <td>1</td> <td>Poly:0x8005</td> </tr> <tr> <td>1</td> <td>X</td> <td>Poly:0x04C11DB7</td> </tr> </tbody> </table>	MODE[1:0]		生成多项式 (简写)	0	0	Poly:0x1021 (default)	0	1	Poly:0x8005	1	X	Poly:0x04C11DB7
MODE[1:0]		生成多项式 (简写)											
0	0	Poly:0x1021 (default)											
0	1	Poly:0x8005											
1	X	Poly:0x04C11DB7											

## CRCDAT (数据寄存器)

CRCDAT (数据寄存器)		基地址: 0x40021000 偏移地址: 04H						
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:	DAT[31:24]							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:	DAT[23:16]							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:	DAT[15:8]							
Reset:	1	1	1	1	1	1	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							
Write:	DAT[7:0]							
Reset:	1	1	1	1	1	1	1	1

位	功能描述
DAT[31:0]	<b>CRC数据寄存器位</b> 写入该寄存器时，作为数据输入寄存器； 读取该寄存器时，返回CRC计算的结果。

## CRCINIT（CRC 初始化种子寄存器）

CRCINIT (CRC 初始化种子寄存器)		基地址: 0x40021000 偏移地址: 08H						
	Bit31	30	29	28	27	26	25	Bit24
Read:	DAT[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	DAT[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DAT[15:8]							
Write:								
Reset:	1	1	1	1	1	1	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	DAT[7:0]							
Write:								
Reset:	1	1	1	1	1	1	1	1

位	功能描述
DAT[31:0]	<b>CRC初始化数据寄存器</b> 若模式选择CRC-CCITT或CRC-16，则低16位（DAT[15:0]）有效，高16位无效 若模式选择CRC-32，则32位（DAT[31:0]）有效

## 18 DMA 功能

### 18.1 概述

DMA 模块提供高速的数据传输在外设和 RAM 之间或者 RAM 和 RAM 之间，在 DMA 数据搬运的过程中不需要 CPU 的参与，但是访问同一个从设备时会和 CPU 交替占用系统总线。在使能了 SRAM 加密时，DMA 不支持访问 FLASH。

### 18.2 功能描述

- DMA 模块有 13 个独立的可配置通道，通道 0~11 为通用通道，通道 12 为 EMU-FFT 专用通道
- DMA 的传输可以外设到外设，外设到 RAM，RAM 到外设，RAM 到 RAM
- 当访问同一个从设备时，DMA 与 CPU 之间交替占用总线
- DMA 支持最多 128 个外部请求源，每个通道还有一个软件请求源
- 每个传输通道的优先级可用软件编程（4 个级别：非常高、高、中、低）
- 在软件优先级相同的情况下，默认编号小的 DMA 通道具有更高的优先级，即 DMA\_CHN0 > DMA\_CHN1 > ... > DMA\_CHN11；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
- 对源和目标访问可设递增、递减或非增量寻址（地址递增/递减基于源和目标传输宽度）
- DMA 支持两种传输方式：一次请求传输一个数据、一次请求传输整个数据块
- DMA 支持三种中断：传输完成中断，传输错误中断，块传输中断
- 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\_BULK\_NUM[15:8]，达到循环次数后自动关闭相应通道

注：

当低优先级通道正在响应的请求时，当前请求的数据未传输完成，更高优先级的通道也无法抢占响应，需在低优先级通道完成本次请求的数据传输后，才可响应

在数据传输过程中，不能更改 DMA 相关配置寄存器，否则会产生不可预期的错误。

### 18.3 DMA 通道请求列表

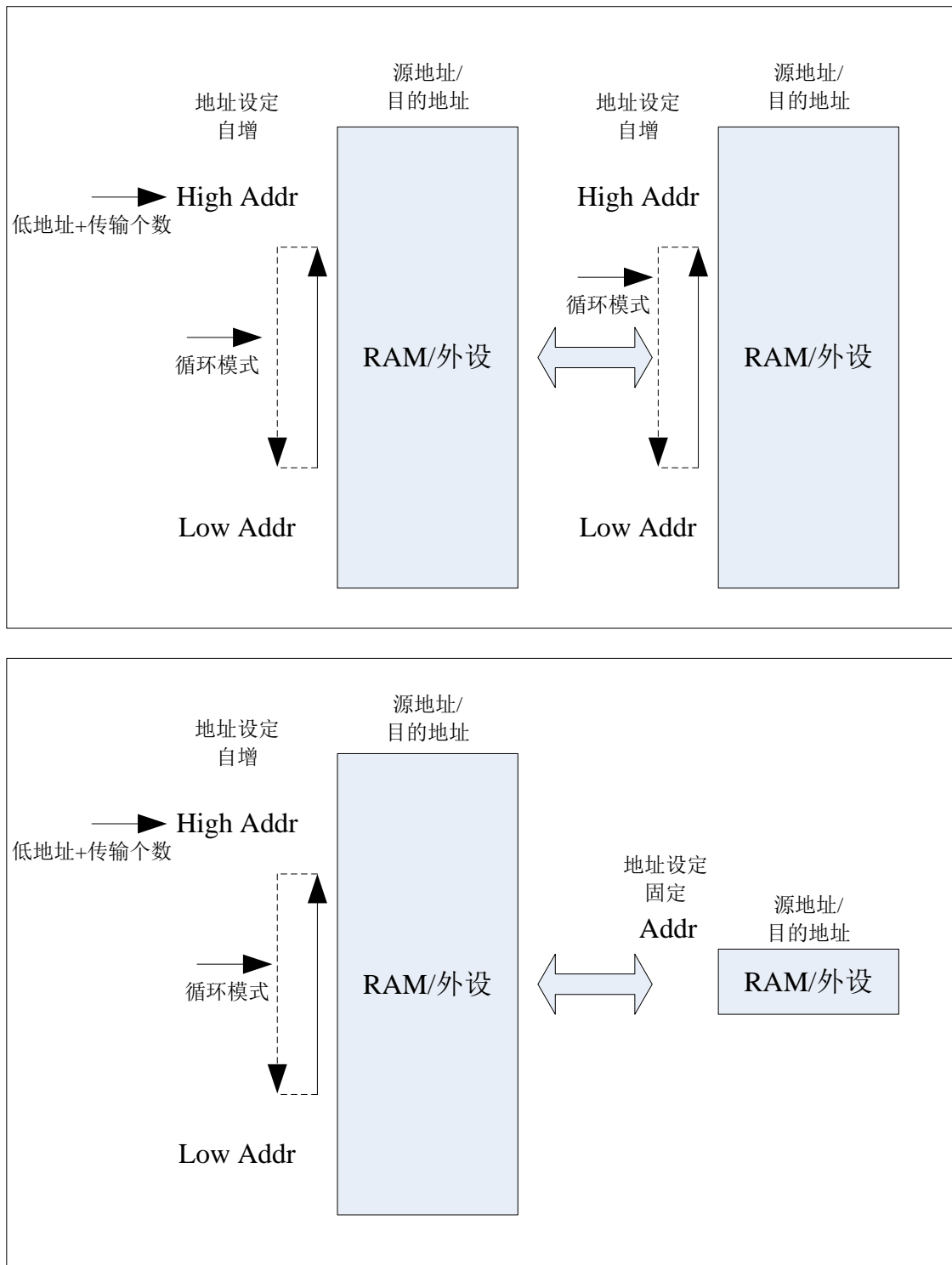
Channel[6:0]	说明
0	Soft request
1	UART0发送
2	UART0接收
3	UART1发送
4	UART1接收
5	UART2发送
6	UART2接收
7	UART3发送
8	UART3接收
9	UART4发送
10	UART4接收
11	UART5发送
12	UART5接收
13	ISO7816_0发送
14	ISO7816_0接收
15	ISO7816_1发送
16	ISO7816_1接收
17	SPI0发送
18	SPI0接收
19	I2C发送
20	I2C接收
21	Reserved
22	Reserved
23	Reserved
24	Timer0
25	Timer1
26	Timer2
27	Timer3
28	Timer4
29	Timer5
30	UART6发送
31	UART6接收
32	SPI1发送
33	SPI1接收
34	SPI2发送
35	SPI2接收
36	Reserved
37	Reserved
38	ADC0转换完成

39	ADC1转换完成
40	Reserved
41	ADC3转换完成
42	ADC4转换完成
43	ADC5转换完成
44	Reserved
45	Reserved
46	Reserved
47	Reserved
48	Reserved
49	Reserved
50	SPI3发送
51	SPI3接收
52	EMU电压通道SPL_Ua波形采样数据更新
53	EMU电压通道SPL_Ub波形采样数据更新
54	EMU电压通道SPL_Uc波形采样数据更新
55	EMU电流通道SPL_Ia波形采样数据更新
56	EMU电流通道SPL_Ib波形采样数据更新
57	EMU电流通道SPL_Ic波形采样数据更新
58	EMU电流通道SPL_In波形采样数据更新
59	EMU电压通道a闪变数据更新
60	EMU电压通道b闪变数据更新
61	EMU电压通道c闪变数据更新
62	EMU Interface数据准备完成（非侵入）
63	EMU Interface数据准备完成（FFT）
64-127	Reserved

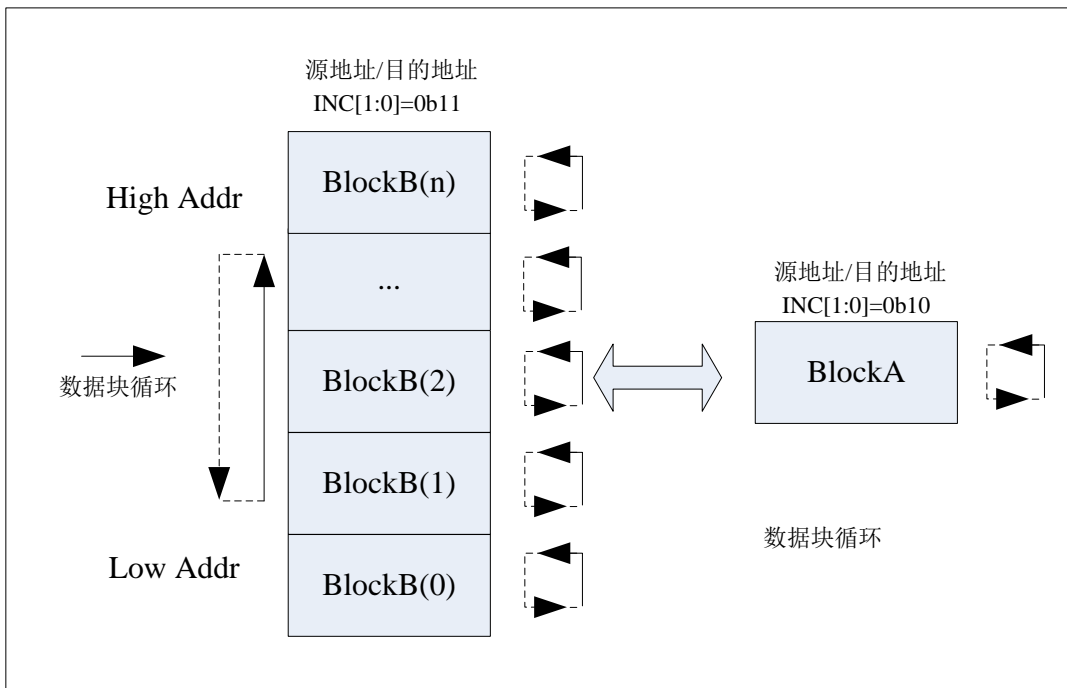
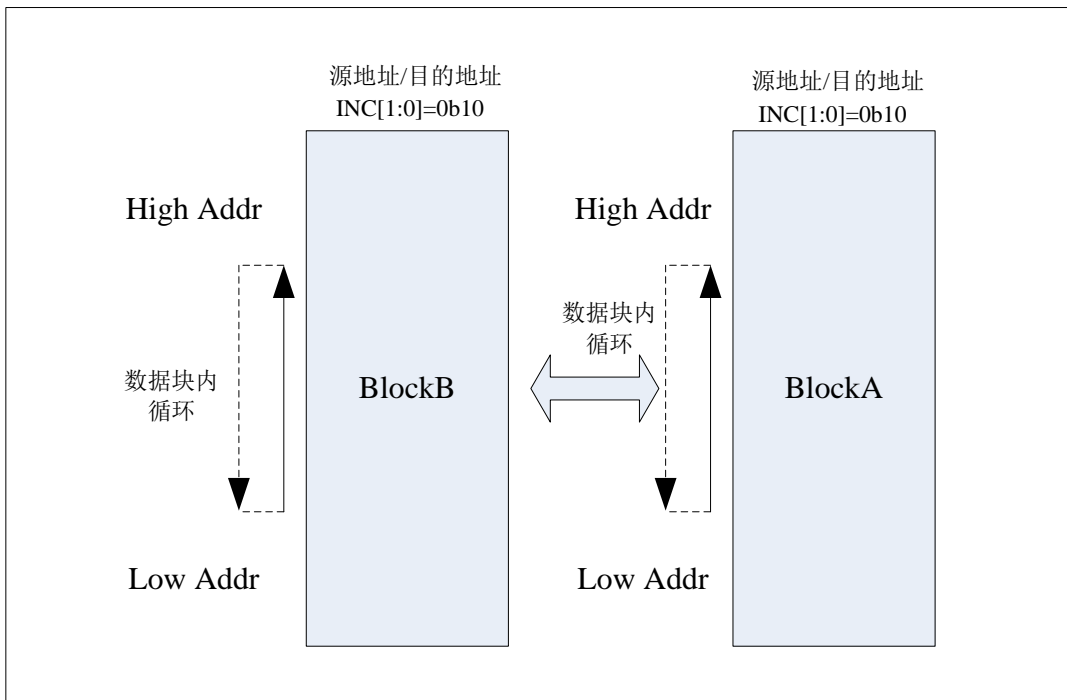
注：

- 1) 当选择DMA发送或接收串口数据（UART）时，UART对应的TXIF/RXIF会置位。
- 2) 请求源52-58的数据可以在EMU模块下选择：原始采样数据或瞬时有效值平方
- 3) 请求源63仅为EMU-FFT专用通道12使用，且通道12仅有63一个请求源可生效。

### 18.4 DMA 通用通道数据传输说明



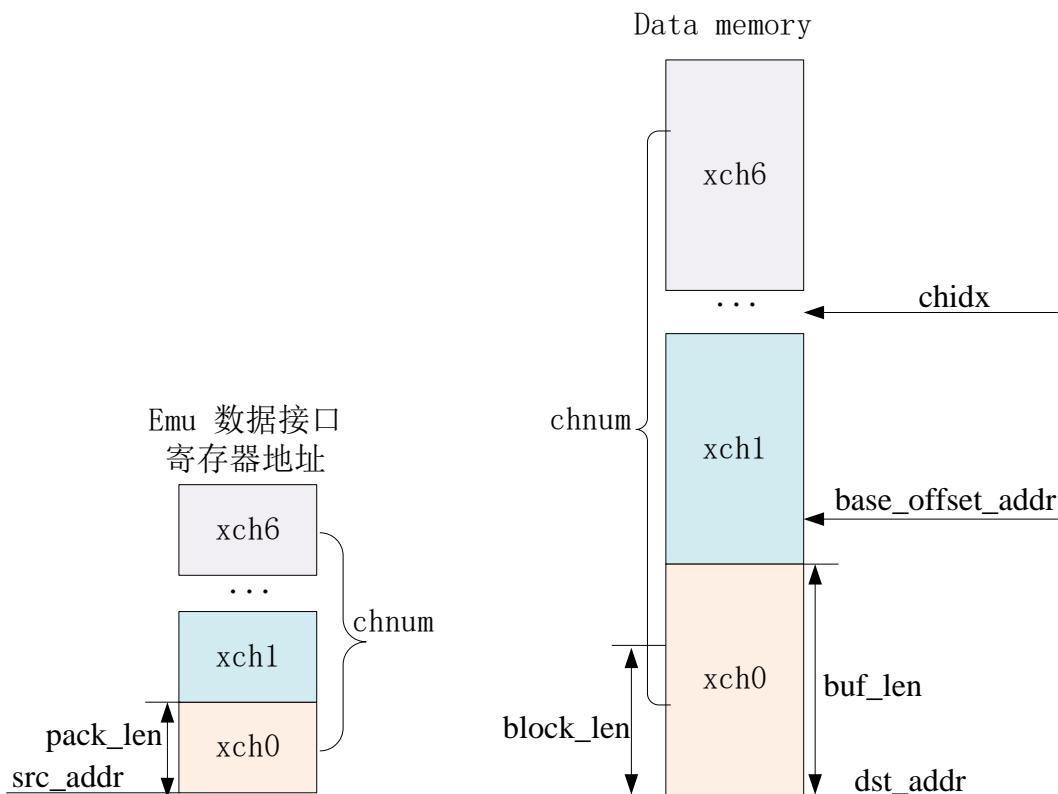
DMA数据传输示意图



DMA数据块传输示意图

## 18.5 EMU-FFT 专用通道数据传输说明

EMU-FFT 专用通道 12 可以完成将 EMU interface 上的 FFT 数据接口（最大 7 个接口地址）分别传输至目标 RAM 的各个内存块中，但不支持其他模块的数据传输。



专用通道 12 的 DMA 数据传输如上图所示。EMU-FFT 专用通道 12 在使用时不同于普通 DMA 通道，需结合 EMU、EMU interface 和 FFT 的配置进行使用。使用配置流程如下

1) 配置源地址：

配置 CHN12SRC.ADDR[31:0] 为对应的 EMU 接口数据的起始基地址 src\_addr；

2) 配置有效通道数及数据包长度：

配置 CHN12SIZE.CHN\_NUM[2:0]=x (x=1~7)，确定要从 EMU interface 的接口地址处传输 x (x=1~7) 个通道的数据；CHN\_NUM[2:0]=0 时，数字内部作为值=7 处理。

EMU 模块下 SYNC\_CFG.SYNC\_DMA\_bw\_mode[1:0] (0x4001CC00, bit 16~17) 控制 DMA 传输数据的打包数量 pack\_len。

EMU 模块 SYNC_CFG. SYNC_DMA_bw_mode[1:0]	DMA 传输打包数量 (pack_len)
00: 16bit	1
01: 24bit	3
10: 32bit	1
11: 32bit	1

3) 配置目的地址：

配置 CHN12TARx.ADDR[31:0] (x=0~6，取决于最大有效通道个数 CHN12SIZE.CHN\_NUM[2:0])，

设定每个 EMU 数据接口通道的传输目标基地址 dst\_addr0~ dst\_addr6;

4) 配置目的地址的数据空间大小:

配置 CHN12SIZE.BUFSIZE[15:0]设定每个传输目标地址的空间大小 buf\_len; 所有生效的目的地址对应的数据空间大小均一致, 使用时需提前计算好, 避免冲突;

5) 配置触发中断或触发 FFT 计算的数据量大小:

配置 CHN12SIZE.INTSIZE[10:0]设定每个有效通道传输多少数据后产生中断 (对应每通道传输的 word 数), 即图中的 block\_len; 当配置 CHN12CTL.FFTTRIG\_EN=1 时, 会触发 FFT 进行计算;

**注: 配合 FFT 模块进行自动多通道 FFT 计算时, 需保证目的地址的配置连续, 即每个通道的数据空间之间没有间隔和重叠。**

根据 FFT 数据采样点个数配置 block\_len 的数值

FFT 数据采样点 个数	16bits mode (fft_length*1/2)	24bits mode (fft_length*3/4)	32bits mode (fft_length)
<b>64</b>	32	48	64
<b>128</b>	64	96	128
<b>256</b>	128	192	256
<b>512</b>	256	384	512
<b>1024</b>	512	768	1024

## 18.6 特殊功能寄存器列表

DMA 模块寄存器基地址: 0x40020000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	DMAIE	R/W	0x0000	中断使能寄存器
0x04	DMAIF	R/W	0x0000	DMA中断标志
0x0C	CHN0CTL	R/W	0x0000	通道0控制寄存器
0x10	CHN0SRC	R/W	0x00000000	通道0源地址寄存器
0x14	CHN0TAR	R/W	0x00000000	通道0目的地址寄存器
0x18	CHN0CNT	R/W	0x0000	通道0传输数量设置寄存器
0x1C	CHN0TCCNT	R	0x0000	通道0传输完成数据个数
0x20	CHN0BULKNUM	R/W	0x0000	通道0块传输个数设置
0x24	CHN1CTL	R/W	0x0000	通道1控制寄存器
0x28	CHN1SRC	R/W	0x00000000	通道1源地址寄存器
0x2C	CHN1TAR	R/W	0x00000000	通道1目的地址寄存器
0x30	CHN1CNT	R/W	0x0000	通道1传输数量设置寄存器
0x34	CHN1TCCNT	R	0x0000	通道1传输完成数据个数
0x38	CHN1BULKNUM	R/W	0x0000	通道1块传输个数设置
0x3C	CHN2CTL	R/W	0x0000	通道2控制寄存器
0x40	CHN2SRC	R/W	0x00000000	通道2源地址寄存器
0x44	CHN2TAR	R/W	0x00000000	通道2目的地址寄存器
0x48	CHN2CNT	R/W	0x0000	通道2传输数量设置寄存器
0x4C	CHN2TCCNT	R	0x0000	通道2传输完成数据个数
0x50	CHN2BULKNUM	R/W	0x0000	通道2块传输个数设置
0x54	CHN3CTL	R/W	0x0000	通道3控制寄存器
0x58	CHN3SRC	R/W	0x00000000	通道3源地址寄存器
0x5C	CHN3TAR	R/W	0x00000000	通道3目的地址寄存器
0x60	CHN3CNT	R/W	0x0000	通道3传输数量设置寄存器
0x64	CHN3TCCNT	R	0x0000	通道3传输完成数据个数
0x68	CHN3BULKNUM	R/W	0x0000	通道3块传输个数设置
0x6C	CHN4CTL	R/W	0x0000	通道4控制寄存器
0x70	CHN4SRC	R/W	0x00000000	通道4源地址寄存器
0x74	CHN4TAR	R/W	0x00000000	通道4目的地址寄存器
0x78	CHN4CNT	R/W	0x0000	通道4传输数量设置寄存器
0x7C	CHN4TCCNT	R	0x0000	通道4传输完成数据个数
0x80	CHN4BULKNUM	R/W	0x0000	通道4块传输个数设置
0x84	CHN5CTL	R/W	0x0000	通道5控制寄存器
0x88	CHN5SRC	R/W	0x00000000	通道5源地址寄存器
0x8C	CHN5TAR	R/W	0x00000000	通道5目的地址寄存器
0x90	CHN5CNT	R/W	0x0000	通道5传输数量设置寄存器
0x94	CHN5TCCNT	R	0x0000	通道5传输完成数据个数

0x98	CHN5BULKNUM	R/W	0x0000	通道5块传输个数设置
0x9C	CHN6CTL	R/W	0x0000	通道6控制寄存器
0xA0	CHN6SRC	R/W	0x00000000	通道6源地址寄存器
0xA4	CHN6TAR	R/W	0x00000000	通道6目的地址寄存器
0xA8	CHN6CNT	R/W	0x0000	通道6传输数量设置寄存器
0xAC	CHN6TCCNT	R	0x0000	通道6传输完成数据个数
0xB0	CHN6BULKNUM	R/W	0x0000	通道6块传输个数设置
0xB4	CHN7CTL	R/W	0x0000	通道7控制寄存器
0xB8	CHN7SRC	R/W	0x00000000	通道7源地址寄存器
0xBC	CHN7TAR	R/W	0x00000000	通道7目的地址寄存器
0xC0	CHN7CNT	R/W	0x0000	通道7传输数量设置寄存器
0xC4	CHN7TCCNT	R	0x0000	通道7传输完成数据个数
0xC8	CHN7BULKNUM	R/W	0x0000	通道7块传输个数设置
0x100	DMAIE1	R/W	0x0000	中断使能寄存器1
0x104	DMAIF1	R/W	0x0000	DMA中断标志1
0x10C	CHN8CTL	R/W	0x0000	通道8控制寄存器
0x110	CHN8SRC	R/W	0x00000000	通道8源地址寄存器
0x114	CHN8TAR	R/W	0x00000000	通道8目的地址寄存器
0x118	CHN8CNT	R/W	0x0000	通道8传输数量设置寄存器
0x11C	CHN8TCCNT	R	0x0000	通道8传输完成数据个数
0x120	CHN8BULKNUM	R/W	0x0000	通道8块传输个数设置
0x124	CHN9CTL	R/W	0x0000	通道9控制寄存器
0x128	CHN9SRC	R/W	0x00000000	通道9源地址寄存器
0x12C	CHN9TAR	R/W	0x00000000	通道9目的地址寄存器
0x130	CHN9CNT	R/W	0x0000	通道9传输数量设置寄存器
0x134	CHN9TCCNT	R	0x0000	通道9传输完成数据个数
0x138	CHN9BULKNUM	R/W	0x0000	通道9块传输个数设置
0x13C	CHN10CTL	R/W	0x0000	通道10控制寄存器
0x140	CHN10SRC	R/W	0x00000000	通道10源地址寄存器
0x144	CHN10TAR	R/W	0x00000000	通道10目的地址寄存器
0x148	CHN10CNT	R/W	0x0000	通道10传输数量设置寄存器
0x14C	CHN10TCCNT	R	0x0000	通道10传输完成数据个数
0x150	CHN10BULKNUM	R/W	0x0000	通道10块传输个数设置
0x154	CHN11CTL	R/W	0x0000	通道11控制寄存器
0x158	CHN11SRC	R/W	0x00000000	通道11源地址寄存器
0x15C	CHN11TAR	R/W	0x00000000	通道11目的地址寄存器
0x160	CHN11CNT	R/W	0x0000	通道11传输数量设置寄存器
0x164	CHN11TCCNT	R	0x0000	通道11传输完成数据个数
0x168	CHN11BULKNUM	R/W	0x0000	通道11块传输个数设置
0x200	DMAIE2	R/W	0x0000	中断使能寄存器2
0x204	DMAIF2	R/W	0x0000	DMA中断标志2

0x208	CHN12CTL	R/W	0x007E00	通道12控制寄存器
0x20C	CHN12SRC	R/W	0x40022020	通道12源地址寄存器
0x210	CHN12TAR0	R/W	0x00000000	通道12目的地址寄存器0
0x214	CHN12TAR1	R/W	0x00000000	通道12目的地址寄存器1
0x218	CHN12TAR2	R/W	0x00000000	通道12目的地址寄存器2
0x21C	CHN12TAR3	R/W	0x00000000	通道12目的地址寄存器3
0x220	CHN12TAR4	R/W	0x00000000	通道12目的地址寄存器4
0x224	CHN12TAR5	R/W	0x00000000	通道12目的地址寄存器5
0x228	CHN12TAR6	R/W	0x00000000	通道12目的地址寄存器6
0x22C	CHN12SIZE	R/W	0x0000	通道12长度控制寄存器
0x230	CHN12CLR	W	0x0000	通道12清除使能寄存器
0x238	CHN12EMUCH0OFFSET	R	0x0000	搬运EMU通道0数据偏移量寄存器
0x23C	CHN12IDX	R	0x0000	通道12当前搬移数据的通道ID寄存器
0x240	CHN12INTCNT	R	0x0000	通道12已完成数据搬移个数寄存器
0x244	CHN12PACKCNT	R	0x0000	当前emu的chx发送的是包中第几个数据寄存器
0x248	CHN12STARTOFFSET	R/W	0x0000	目的地址起始地址偏移量寄存器

## 18.7 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: 0x40020000 偏移地址: 00H						
	Bit31	30	29	28	27	26	25	Bit24
Read:	X	TEIE7	BCIE7	TCIE7	X	TEIE6	BCIE6	TCIE6
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	TEIE5	BCIE5	TCIE5	X	TEIE4	BCIE4	TCIE4
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	TEIE3	BCIE3	TCIE3	X	TEIE2	BCIE2	TCIE2
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	TEIE1	BCIE1	TCIE1	X	TEIE0	BCIE0	TCIE0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TCIE7	通道 7 传输结束中断使能 0: 禁止 1: 使能
BCIE7	通道 7 块传输中断使能 0: 禁止 1: 使能
TEIE7	通道 7 传输错误中断使能 0: 禁止 1: 使能

注: DMA 通道 0-6 对应使能位功能与通道 7 一致;

### DMAIF (DMA 中断标志寄存器)

DMAIF (DMA 中断标志寄存器)		基地址: 0x40020000 偏移地址: 04H	
------------------------	--	------------------------------	--

	<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>	X	TEIF7	BCIF7	TCIF7	X	TEIF6	BCIF6	TCIF6
<b>Write:</b>								
<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	TEIF5	BCIF5	TCIF5	X	TEIF4	BCIF4	TCIF4
<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>	X	TEIF3	BCIF3	TCIF3	X	TEIF2	BCIF2	TCIF2
<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	TEIF1	BCIF1	TCIF1	X	TEIF0	BCIF0	TCIF0
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
TCIF7	通道 7 传输结束中断标志 0: 未产生中断 1: 产生中断
BCIF7	通道 7 块传输完成中断标志 0: 未产生中断 1: 产生中断
TEIF7	通道 7 传输错误中断标志 0: 未产生中断 1: 产生中断

注：DMA 通道 0-6 对应标志位功能与通道 7 一致；

## CHN0CTL (DMA 通道 0 控制寄存器)

<b>CHN0CTL</b> (DMA 通道 0 控制寄存器)		基地址: 0x40020000 偏移地址: 0CH						
	<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>	DESTIN_	SOURC_	X	X	X	X	PRI[1:0]	
<b>Write:</b>	DIR	DIR						
<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>	Channel[6:0]							DESTIN
<b>Write:</b>								_INC[1]
<b>Reset:</b>	0	0	0	0	0	0	0	0

	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	DESTIN_	SOURC_INC[1:0]		CYCLE	MODE	SIZE[1:0]		CHN0E
<b>Write:</b>	INC[0]							N
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DESTIN_DIR	<b>DMA 通道 0 目的地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
SOURC_DIR	<b>DMA 通道 0 源地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
PRI[1:0]	<b>DMA 通道 0 优先级配置寄存器</b> 00: 低优先级 (default) 01: 中优先级 10: 高优先级 11: 最高优先级
Channel[6:0]	触发通道选择, 具体参见 DMA 通道请求列表来定
DESTIN_INC[1:0]	<b>DMA 通道 0 目的地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 SOURC_DIR 控制, 默认递增
SOURC_INC[1:0]	<b>DMA 通道 0 源地址地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
CYCLE	<b>DMA 通道 0 循环模式</b> 0: 不循环模式 1: 循环模式
MODE	<b>DMA 通道 0 传输模式</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)
SIZE[1:0]	<b>DMA 通道 0 MEMORY 的传送位数 (外设的传送位数固定为 32bit):</b> 00: 8 位 01: 16 位 10: 32 位 11: 32 位
CHN0EN	<b>DMA 通道 0 使能</b> 0: 禁止

1: 使能

**CHN0SRC (DMA 通道 0 源地址寄存器)**

CHN0SRC (DMA 通道 0 源地址寄存器)			基地址: 0x40020000 偏移地址: 10H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 0 数据传输源地址寄存器

**CHN0TAR (DMA 通道 0 目的地址寄存器)**

CHN0TAR (DMA 通道 0 目的地址寄存器)			基地址: 0x40020000 偏移地址: 14H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:								
Reset:	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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 0 数据传输目的地址寄存器

### CHN0CNT (DMA 通道 0 传输数量寄存器)

<b>CHN0CNT</b> (DMA 通道 0 传输数量寄存器)			基地址: 0x40020000 偏移地址: 18H					
	<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>	Num[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>	Num[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

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

### CHN0TCNT (DMA 通道 0 已传输数据个数)

<b>CHN0TCNT</b> (DMA 通道 0 已传输数据个数)			基地址: 0x40020000 偏移地址: 1CH					
	<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>	Num[15:8]							
<b>Write:</b>	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:	Num[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Num[15:0]	指示 DMA 通道 0 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

### CHN0BULKNUM (DMA 通道 0 块传输设置寄存器)

CHN0BULKNUM (DMA 通道 0 块传输设置寄存器)			基地址: 0x40020000 偏移地址: 20H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	CYCLE[7:0]							
Write:	CYCLE[7:0]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num[7:0]							
Write:	Num[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CYCLE[7:0]	通道 0 循环次数选择，当用户选择 DMA 传输为循环模式时起作用： =0: 无限次循环 =0x01---0xFF: 为 0x01---0xFF 次循环
Num[7:0]	通道 0 块传输数据个数选择，表示每一个数据块内有多少个数据： 0—255 个 如果用户设置的是块传输，那么该寄存器则表示每一个数据块内有多少个数据

### CHN1CTL (DMA 通道 1 控制寄存器)

CHN1CTL (DMA 通道 1 控制寄存器)			基地址: 0x40020000 偏移地址: 24H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	DESTIN_	SOURC_	X	X	X	X	PRI[1:0]	
Write:	DIR	DIR	X	X	X	X	PRI[1:0]	

<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>	Channel[6:0]							DESTIN _INC[1]
<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>	DESTIN_ INC[0]	SOURC_INC[1:0]		CYCLE	MODE	SIZE[1:0]		CHN1E N
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DESTIN_DIR	<b>DMA 通道 1 目的地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
SOURC_DIR	<b>DMA 通道 1 源地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
PRI[1:0]	<b>DMA 通道 1 优先级配置寄存器</b> 00: 低优先级 (default) 01: 中优先级 10: 高优先级 11: 最高优先级
Channel[6:0]	触发通道选择, 具体参见 DMA 通道请求列表来定
DESTIN_INC[1:0]	<b>DMA 通道 1 目的地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
SOURC_INC[1:0]	<b>DMA 通道 1 源地址地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
CYCLE	<b>DMA 通道 1 循环模式</b> 0: 不循环模式 1: 循环模式
MODE	<b>DMA 通道 1 传输模式</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)
SIZE[1:0]	<b>DMA 通道 1 MEMORY 的传送位数 (外设的传送位数固定为 32bit):</b> 00: 8 位

	01: 16 位 10: 32 位 11: 32 位
CHN1EN	<b>DMA 通道 1 使能</b> 0: 禁止 1: 使能

### CHN1SRC (DMA 通道 1 源地址寄存器)

<b>CHN1SRC</b> (DMA 通道 1 源地址寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>28H</b>					
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							
<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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 1 数据传输源地址寄存器

### CHN1TAR (DMA 通道 1 目的地址寄存器)

<b>CHN1TAR</b> (DMA 通道 1 目的地址寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>2CH</b>					
	<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>	ADDR[31:24]							
<b>Write:</b>								

Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:	ADDR[23:16]							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR[15:8]							
Write:	ADDR[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR[7:0]							
Write:	ADDR[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 1 数据传输目的地址寄存器

### CHN1CNT (DMA 通道 1 传输数量寄存器)

CHN1CNT (DMA 通道 1 传输数量寄存器)		基地址: 0x40020000 偏移地址: 30H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num[15:8]							
Write:	Num[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num[7:0]							
Write:	Num[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Num[15:0]	DMA 通道 1 数据传输个数设置寄存器 最大设置到 65535 个传输数据 如果用户设置的是块传输, 那么该寄存器则表示用户需要传输的数据块个数

### CHN1TCNT (DMA 通道 1 已传输数据个数)

CHN1TCNT	基地址: 0x40020000
----------	-----------------

(DMA 通道 1 已传输数据个数)			偏移地址: 34H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Num[15:0]	指示 DMA 通道 1 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

### CHN1BULKNUM (DMA 通道 1 块传输设置寄存器)

CHN1BULKNUM (DMA 通道 1 块传输设置寄存器)			基地址: 0x40020000 偏移地址: 38H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	CYCLE[7:0]							
Write:	CYCLE[7:0]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num[7:0]							
Write:	Num[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CYCLE[7:0]	通道 1 循环次数选择, 当用户选择 DMA 传输为循环模式时起作用: =0: 无限次循环 =0x01----0xFF: 为 0x01----0xFF 次循环
Num[7:0]	通道 1 块传输数据个数选择, 表示每一个数据块内有多少个数据: 0 — 255 个 如果用户设置的是块传输, 那么该寄存器则表示每一个数据块内有多少个数据

**CHN2CTL (DMA 通道 2 控制寄存器)**

CHN2CTL (DMA 通道 2 控制寄存器)		基地址: 0x40020000 偏移地址: 3CH						
	Bit23	22	21	20	19	18	17	Bit16
Read:	DESTIN_	SOURC_	X	X	X	X	PRI[1:0]	
Write:	DIR	DIR						
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	Channel[6:0]							DESTIN
Write:								_INC[1]
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DESTIN_	SOURC_INC[1:0]		CYCLE	MODE	SIZE[1:0]		CHN2E
Write:	INC[0]							N
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DESTIN_DIR	<b>DMA 通道 2 目的地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
SOURC_DIR	<b>DMA 通道 2 源地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
PRI[1:0]	<b>DMA 通道 2 优先级配置寄存器</b> 00: 低优先级 (default) 01: 中优先级 10: 高优先级 11: 最高优先级
Channel[6:0]	触发通道选择, 具体参见 DMA 通道请求列表来定
DESTIN_INC[1:0]	<b>DMA 通道 2 目的地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
SOURC_INC[1:0]	<b>DMA 通道 2 源地址地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增

CYCLE	<b>DMA 通道 2 循环模式</b> 0: 不循环模式 1: 循环模式
MODE	<b>DMA 通道 2 传输模式</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)
SIZE[1:0]	<b>DMA 通道 2 MEMORY 的传送位数 (外设的传送位数固定为 32bit):</b> 00: 8 位 01: 16 位 10: 32 位 11: 32 位
CHN2EN	<b>DMA 通道 2 使能</b> 0: 禁止 1: 使能

### CHN2SRC (DMA 通道 2 源地址寄存器)

CHN2SRC (DMA 通道 2 源地址寄存器)			基地址: 0x40020000 偏移地址: 40H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:	ADDR[31:24]							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:	ADDR[23:16]							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR[15:8]							
Write:	ADDR[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR[7:0]							
Write:	ADDR[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 2 数据传输源地址寄存器

**CHN2TAR (DMA 通道 2 目的地址寄存器)**

<b>CHN2TAR</b> (DMA 通道 2 目的地址寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>44H</b>					
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							
<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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 2 数据传输目的地址寄存器

**CHN2CNT (DMA 通道 2 传输数量寄存器)**

<b>CHN2CNT</b> (DMA 通道 2 传输数量寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>48H</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>	Num[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>	Num[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
---	------

Num[15:0]	<b>DMA 通道 2 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
-----------	---

### CHN2TCNT (DMA 通道 2 已传输数据个数)

<b>CHN2TCNT</b> (DMA 通道 2 已传输数据个数)		基地址: 0x40020000 偏移地址: 4CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Num[15:0]	指示 DMA 通道 2 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

### CHN2BULKNUM (DMA 通道 2 块传输设置寄存器)

<b>CHN2BULKNUM</b> (DMA 通道 2 块传输设置寄存器)		基地址: 0x40020000 偏移地址: 50H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	CYCLE[7:0]							
Write:	CYCLE[7:0]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num[7:0]							
Write:	Num[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CYCLE[7:0]	通道 2 循环次数选择，当用户选择 DMA 传输为循环模式时起作用： =0: 无限次循环

	=0x01----0xFF: 为 0x01----0xFF 次循环
Num[7:0]	通道 2 块传输数据个数选择, 表示每一个数据块内有多少个数据: 0 — 255 个 如果用户设置的是块传输, 那么该寄存器则表示每一个数据块内有多少个数据

### CHN3CTL (DMA 通道 3 控制寄存器)

CHN3CTL (DMA 通道 3 控制寄存器)		基地址: 0x40020000 偏移地址: 54H						
	Bit23	22	21	20	19	18	17	Bit16
Read:	DESTIN_	SOURC_	X	X	X	X	PRI[1:0]	
Write:	DIR	DIR						
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	Channel[6:0]							DESTIN
Write:								_INC[1]
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DESTIN_	SOURC_INC[1:0]		CYCLE	MODE	SIZE[1:0]		CHN3E
Write:	INC[0]							N
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DESTIN_DIR	<b>DMA 通道 3 目的地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
SOURC_DIR	<b>DMA 通道 3 源地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
PRI[1:0]	<b>DMA 通道 3 优先级配置寄存器</b> 00: 低优先级 (default) 01: 中优先级 10: 高优先级 11: 最高优先级
Channel[6:0]	触发通道选择, 具体参见 DMA 通道请求列表来定
DESTIN_INC[1:0]	<b>DMA 通道 3 目的地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
SOURC_INC[1:0]	<b>DMA 通道 3 源地址地址增减模式</b>

	00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
CYCLE	<b>DMA 通道 3 循环模式</b> 0: 不循环模式 1: 循环模式
MODE	<b>DMA 通道 3 传输模式</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)
SIZE[1:0]	<b>DMA 通道 3 MEMORY 的传送位数 (外设的传送位数固定为 32bit):</b> 00: 8 位 01: 16 位 10: 32 位 11: 32 位
CHN3EN	<b>DMA 通道 3 使能</b> 0: 禁止 1: 使能

### CHN3SRC (DMA 通道 3 源地址寄存器)

CHN3SRC (DMA 通道 3 源地址寄存器)			基地址: 0x40020000 偏移地址: 58H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:	ADDR[31:24]							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:	ADDR[23:16]							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR[15:8]							
Write:	ADDR[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR[7:0]							
Write:	ADDR[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 3 数据传输源地址寄存器

### CHN3TAR (DMA 通道 3 目的地址寄存器)

CHN3TAR (DMA 通道 3 目的地址寄存器)		基地址: 0x40020000 偏移地址: 5CH						
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:	ADDR[31:24]							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:	ADDR[23:16]							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR[15:8]							
Write:	ADDR[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR[7:0]							
Write:	ADDR[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 3 数据传输目的地址寄存器

### CHN3CNT (DMA 通道 3 传输数量寄存器)

CHN3CNT (DMA 通道 3 传输数量寄存器)		基地址: 0x40020000 偏移地址: 60H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num[15:8]							
Write:	Num[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num[7:0]							

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

位	功能描述
Num[15:0]	<b>DMA 通道 3 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

### CHN3TCNT (DMA 通道 3 已传输数据个数)

<b>CHN3TCNT</b> (DMA 通道 3 已传输数据个数)			基地址: 0x40020000 偏移地址: 64H					
	<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>	Num[15:8]							
<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>	Num[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
Num[15:0]	指示 DMA 通道 3 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

### CHN3BULKNUM (DMA 通道 3 块传输设置寄存器)

<b>CHN3BULKNUM</b> (DMA 通道 3 块传输设置寄存器)			基地址: 0x40020000 偏移地址: 68H					
	<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>	CYCLE[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>	Num[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
CYCLE[7:0]	通道 3 循环次数选择，当用户选择 DMA 传输为循环模式时起作用： =0: 无限次循环 =0x01---0xFF: 为 0x01---0xFF 次循环
Num[7:0]	通道 3 块传输数据个数选择，表示每一个数据块内有多少个数据： 0 — 255 个 如果用户设置的是块传输，那么该寄存器则表示每一个数据块内有多少个数据

## CHN4CTL (DMA 通道 4 控制寄存器)

CHN4CTL (DMA 通道 4 控制寄存器)		基地址: 0x40020000 偏移地址: 6CH						
	Bit23	22	21	20	19	18	17	Bit16
Read:	DESTIN_	SOURC_	X	X	X	X	PRI[1:0]	
Write:	DIR	DIR						
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	Channel[6:0]							DESTIN_
Write:								_INC[1]
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DESTIN_	SOURC_INC[1:0]		CYCLE	MODE	SIZE[1:0]		CHN4E
Write:	INC[0]							N
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DESTIN_DIR	<b>DMA 通道 4 目的地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
SOURC_DIR	<b>DMA 通道 4 源地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
PRI[1:0]	<b>DMA 通道 4 优先级配置寄存器</b> 00: 低优先级 (default) 01: 中优先级 10: 高优先级 11: 最高优先级
Channel[6:0]	触发通道选择，具体参见 DMA 通道请求列表来定
DESTIN_INC[1:0]	<b>DMA 通道 4 目的地址增减模式</b> 00: 固定

	01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
SOURC_INC[1:0]	<b>DMA 通道 4 源地址地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
CYCLE	<b>DMA 通道 4 循环模式</b> 0: 不循环模式 1: 循环模式
MODE	<b>DMA 通道 4 传输模式</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)
SIZE[1:0]	<b>DMA 通道 4 MEMORY 的传送位数 (外设的传送位数固定为 32bit):</b> 00: 8 位 01: 16 位 10: 32 位 11: 32 位
CHN4EN	<b>DMA 通道 4 使能</b> 0: 禁止 1: 使能

### CHN4SRC (DMA 通道 4 源地址寄存器)

CHN4SRC (DMA 通道 4 源地址寄存器)			基地址: 0x40020000 偏移地址: 70H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:	ADDR[31:24]							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:	ADDR[23:16]							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR[15:8]							
Write:	ADDR[15:8]							
Reset:	0	0	0	0	0	0	0	0

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

位	功能描述
ADDR[31:0]	通道 4 数据传输源地址寄存器

### CHN4TAR (DMA 通道 4 目的地址寄存器)

CHN4TAR (DMA 通道 4 目的地址寄存器)			基地址: 0x40020000 偏移地址: 74H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 4 数据传输目的地址寄存器

### CHN4CNT (DMA 通道 4 传输数量寄存器)

CHN4CNT (DMA 通道 4 传输数量寄存器)			基地址: 0x40020000 偏移地址: 78H					
	Bit15	14	13	12	11	10	9	Bit8

<b>Read:</b>	Num[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>	Num[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
Num[15:0]	<b>DMA 通道 4 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

### CHN4TCNT (DMA 通道 4 已传输数据个数)

<b>CHN4TCNT</b> (DMA 通道 4 已传输数据个数)			基地址: 0x40020000 偏移地址: 7CH					
	<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>	Num[15:8]							
<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>	Num[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
Num[15:0]	指示 DMA 通道 4 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

### CHN4BULKNUM (DMA 通道 4 块传输设置寄存器)

<b>CHN4BULKNUM</b> (DMA 通道 4 块传输设置寄存器)			基地址: 0x40020000 偏移地址: 80H					
	<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>	CYCLE[7:0]							
<b>Write:</b>								

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

位	功能描述
CYCLE[7:0]	通道 4 循环次数选择, 当用户选择 DMA 传输为循环模式时起作用: =0: 无限次循环 =0x01----0xFF: 为 0x01----0xFF 次循环
Num[7:0]	通道 4 块传输数据个数选择, 表示每一个数据块内有多少个数据: 0—255 个 如果用户设置的是块传输, 那么该寄存器则表示每一个数据块内有多少个数据

### CHN5CTL (DMA 通道 5 控制寄存器)

CHN5CTL (DMA 通道 5 控制寄存器)		基地址: 0x40020000 偏移地址: 84H						
	Bit23	22	21	20	19	18	17	Bit16
Read:	DESTIN_	SOURC_	X	X	X	X	PRI[1:0]	
Write:	DIR	DIR						
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	Channel[6:0]							DESTIN_
Write:								INC[1]
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DESTIN_	SOURC_INC[1:0]		CYCLE	MODE	SIZE[1:0]		CHN5E
Write:	INC[0]							N
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DESTIN_DIR	<b>DMA 通道 5 目的地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
SOURC_DIR	<b>DMA 通道 5 源地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
PRI[1:0]	<b>DMA 通道 5 优先级配置寄存器</b> 00: 低优先级 (default) 01: 中优先级

	10: 高优先级 11: 最高优先级
Channel[6:0]	触发通道选择, 具体参见 DMA 通道请求列表来定
DESTIN_INC[1:0]	<b>DMA 通道 5 目的地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
SOURC_INC[1:0]	<b>DMA 通道 5 源地址地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
CYCLE	<b>DMA 通道 5 循环模式</b> 0: 不循环模式 1: 循环模式
MODE	<b>DMA 通道 5 传输模式</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)
SIZE[1:0]	<b>DMA 通道 5 MEMORY 的传送位数 (外设的传送位数固定为 32bit):</b> 00: 8 位 01: 16 位 10: 32 位 11: 32 位
CHN5EN	<b>DMA 通道 5 使能</b> 0: 禁止 1: 使能

### CHN5SRC (DMA 通道 5 源地址寄存器)

<b>CHN5SRC</b> (DMA 通道 5 源地址寄存器)			<b>基地址: 0x40020000</b>					
			<b>偏移地址: 88H</b>					
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							
<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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 5 数据传输源地址寄存器

### CHN5TAR (DMA 通道 5 目的地址寄存器)

<b>CHN5TAR</b> (DMA 通道 5 目的地址寄存器)			<b>基地址: 0x40020000</b> <b>偏移地址: 8CH</b>					
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							
<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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 5 数据传输目的地址寄存器

**CHN5CNT (DMA 通道 5 传输数量寄存器)**

<b>CHN5CNT</b> (DMA 通道 5 传输数量寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>90H</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>	Num[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>	Num[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
Num[15:0]	<b>DMA 通道 5 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

**CHN5TCNT (DMA 通道 5 已传输数据个数)**

<b>CHN5TCNT</b> (DMA 通道 5 已传输数据个数)			基地址: <b>0x40020000</b> 偏移地址: <b>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>	Num[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>	Num[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
Num[15:0]	<b>指示 DMA 通道 5 已经传输完成的数据个数</b> 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

**CHN5BULKNUM (DMA 通道 5 块传输设置寄存器)**

<b>CHN5BULKNUM</b> (DMA 通道 5 块传输设置寄存器)		基地址: 0x40020000 偏移地址: 98H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	CYCLE[7:0]							
Write:	CYCLE[7:0]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num[7:0]							
Write:	Num[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CYCLE[7:0]	通道 5 循环次数选择, 当用户选择 DMA 传输为循环模式时起作用: =0: 无限次循环 =0x01----0xFF: 为 0x01----0xFF 次循环
Num[7:0]	通道 5 块传输数据个数选择, 表示每一个数据块内有多少个数据: 0—255 个 如果用户设置的是块传输, 那么该寄存器则表示每一个数据块内有多少个数据

**CHN6CTL (DMA 通道 6 控制寄存器)**

<b>CHN6CTL</b> (DMA 通道 6 控制寄存器)		基地址: 0x40020000 偏移地址: 9CH						
	Bit23	22	21	20	19	18	17	Bit16
Read:	DESTIN_	SOURC_	X	X	X	X	PRI[1:0]	
Write:	DIR	DIR						
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	Channel[6:0]							DESTIN_
Write:	Channel[6:0]							INC[1]
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DESTIN_	SOURC_INC[1:0]		CYCLE	MODE	SIZE[1:0]		CHN6E
Write:	INC[0]	SOURC_INC[1:0]		CYCLE	MODE	SIZE[1:0]		N
Reset:	0	0	0	0	0	0	0	0

位	功能描述
---	------

DESTIN_DIR	<b>DMA 通道 6 目的地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
SOURC_DIR	<b>DMA 通道 6 源地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
PRI[1:0]	<b>DMA 通道 6 优先级配置寄存器</b> 00: 低优先级 (default) 01: 中优先级 10: 高优先级 11: 最高优先级
Channel[6:0]	触发通道选择, 具体参见 DMA 通道请求列表来定
DESTIN_INC[1:0]	<b>DMA 通道 6 目的地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
SOURC_INC[1:0]	<b>DMA 通道 6 源地址地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
CYCLE	<b>DMA 通道 6 循环模式</b> 0: 不循环模式 1: 循环模式
MODE	<b>DMA 通道 6 传输模式</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)
SIZE[1:0]	<b>DMA 通道 6 MEMORY 的传送位数 (外设的传送位数固定为 32bit):</b> 00: 8 位 01: 16 位 10: 32 位 11: 32 位
CHN6EN	<b>DMA 通道 6 使能</b> 0: 禁止 1: 使能

**CHN6SRC (DMA 通道 6 源地址寄存器)**

<b>CHN6SRC</b> (DMA 通道 6 源地址寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>A0H</b>					
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							
<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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 6 数据传输源地址寄存器

**CHN6TAR (DMA 通道 6 目的地址寄存器)**

<b>CHN6TAR</b> (DMA 通道 6 目的地址寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>A4H</b>					
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							
<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>	ADDR[15:8]							
<b>Write:</b>								

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

位	功能描述
ADDR[31:0]	通道 6 数据传输目的地址寄存器

### CHN6CNT (DMA 通道 6 传输数量寄存器)

CHN6CNT (DMA 通道 6 传输数量寄存器)			基地址: 0x40020000 偏移地址: A8H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num[15:8]							
Write:	Num[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num[7:0]							
Write:	Num[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Num[15:0]	DMA 通道 6 数据传输个数设置寄存器 最大设置到 65535 个传输数据 如果用户设置的是块传输, 那么该寄存器则表示用户需要传输的数据块个数

### CHN6TCNT (DMA 通道 6 已传输数据个数)

CHN6TCNT (DMA 通道 6 已传输数据个数)			基地址: 0x40020000 偏移地址: ACH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num[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
Num[15:0]	指示 DMA 通道 6 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

## CHN6BULKNUM (DMA 通道 6 块传输设置寄存器)

<b>CHN6BULKNUM</b> (DMA 通道 6 块传输设置寄存器)			基地址: 0x40020000 偏移地址: B0H					
	<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>	CYCLE[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>	Num[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
CYCLE[7:0]	通道 6 循环次数选择，当用户选择 DMA 传输为循环模式时起作用： =0: 无限次循环 =0x01---0xFF: 为 0x01---0xFF 次循环
Num[7:0]	通道 6 块传输数据个数选择，表示每一个数据块内有多少个数据： 0—255 个 如果用户设置的是块传输，那么该寄存器则表示每一个数据块内有多少个数据

## CHN7CTL (DMA 通道 7 控制寄存器)

<b>CHN7CTL</b> (DMA 通道 7 控制寄存器)			基地址: 0x40020000 偏移地址: B4H					
	<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>	DESTIN_	SOURC_	X	X	X	X	PRI[1:0]	
<b>Write:</b>	DIR	DIR						
<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>	Channel[6:0]							DESTIN

<b>Write:</b>								_INC[1]
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	DESTIN_	SOURC_INC[1:0]		CYCLE	MODE	SIZE[1:0]		CHN7E
<b>Write:</b>	INC[0]							N
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DESTIN_DIR	<b>DMA 通道 7 目的地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
SOURC_DIR	<b>DMA 通道 7 源地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
PRI[1:0]	<b>DMA 通道 7 优先级配置寄存器</b> 00: 低优先级 (default) 01: 中优先级 10: 高优先级 11: 最高优先级
Channel[6:0]	触发通道选择, 具体参见 DMA 通道请求列表来定
DESTIN_INC[1:0]	<b>DMA 通道 7 目的地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
SOURC_INC[1:0]	<b>DMA 通道 7 源地址地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
CYCLE	<b>DMA 通道 7 循环模式</b> 0: 不循环模式 1: 循环模式
MODE	<b>DMA 通道 7 传输模式</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)
SIZE[1:0]	<b>DMA 通道 7 MEMORY 的传送位数 (外设的传送位数固定为 32bit):</b> 00: 8 位 01: 16 位 10: 32 位 11: 32 位

CHN7EN	<b>DMA 通道 7 使能</b> 0: 禁止 1: 使能
--------	--------------------------------------

### CHN7SRC (DMA 通道 7 源地址寄存器)

<b>CHN7SRC</b> (DMA 通道 7 源地址寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>B8H</b>					
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							
<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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 7 数据传输源地址寄存器

### CHN7TAR (DMA 通道 7 目的地址寄存器)

<b>CHN7TAR</b> (DMA 通道 7 目的地址寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>BCH</b>					
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							

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

位	功能描述
ADDR[31:0]	通道 7 数据传输目的地址寄存器

### CHN7CNT (DMA 通道 7 传输数量寄存器)

<b>CHN7CNT</b> (DMA 通道 7 传输数量寄存器)	基地址: 0x40020000 偏移地址: C0H							
	<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>	Num[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>	Num[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
Num[15:0]	DMA 通道 7 数据传输个数设置寄存器 最大设置到 65535 个传输数据 如果用户设置的是块传输, 那么该寄存器则表示用户需要传输的数据块个数

### CHN7TCNT (DMA 通道 7 已传输数据个数)

<b>CHN7TCNT</b> (DMA 通道 7 已传输数据个数)	基地址: 0x40020000 偏移地址: C4H							
	<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>	Num[15:8]							
<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>	Num[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
Num[15:0]	指示 DMA 通道 7 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

### CHN7BULKNUM (DMA 通道 7 块传输设置寄存器)

<b>CHN7BULKNUM</b> (DMA 通道 7 块传输设置寄存器)			基地址: 0x40020000 偏移地址: C8H					
	<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>	CYCLE[7:0]							
<b>Write:</b>	CYCLE[7:0]							
<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>	Num[7:0]							
<b>Write:</b>	Num[7:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
CYCLE[7:0]	通道 7 循环次数选择，当用户选择 DMA 传输为循环模式时起作用： =0: 无限次循环 =0x01----0xFF: 为 0x01----0xFF 次循环
Num[7:0]	通道 7 块传输数据个数选择，表示每一个数据块内有多少个数据： 0 — 255 个 如果用户设置的是块传输，那么该寄存器则表示每一个数据块内有多少个数据

### DMAIE1 (DMA 中断使能寄存器 1)

<b>DMAIE1</b> (DMA 中断使能寄存器 1)	基地址: 0x40020000 偏移地址: 100H
----------------------------------	-------------------------------

	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	X	TEIE11	BCIE11	TCIE11	X	TEIE10	BCIE10	TCIE10
<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	TEIE9	BCIE9	TCIE9	X	TEIE8	BCIE8	TCIE8
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
TCIE11	通道 11 传输结束中断使能 0: 禁止 1: 使能
BCIE11	通道 11 块传输中断使能 0: 禁止 1: 使能
TEIE11	通道 11 传输错误中断使能 0: 禁止 1: 使能

注：DMA 通道 8-10 对应使能位功能与通道 11 一致；

## DMAIF1 (DMA 中断标志寄存器 1)

DMAIF1 (DMA 中断标志寄存器 1)		基地址: 0x40020000 偏移地址: 104H						
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	X	TEIF11	BCIF11	TCIF11	X	TEIF10	BCIF10	TCIF10
<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	TEIF9	BCIF9	TCIF9	X	TEIF8	BCIF8	TCIF8
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
TCIF11	通道 11 传输结束中断标志 0: 未产生中断 1: 产生中断
BCIF11	通道 11 块传输完成中断标志 0: 未产生中断 1: 产生中断

TEIF11	<b>通道 11 传输错误中断标志</b> 0: 未产生中断 1: 产生中断
--------	--

注：DMA 通道 8-10 对应标志位功能与通道 11 一致；

## CHN8CTL (DMA 通道 8 控制寄存器)

<b>CHN8CTL</b> (DMA 通道 8 控制寄存器)		基地址: <b>0x40020000</b> 偏移地址: <b>10CH</b>						
	Bit23	22	21	20	19	18	17	Bit16
<b>Read:</b>	DESTIN_	SOURC_	X	X	X	X	PRI[1:0]	
<b>Write:</b>	DIR	DIR						
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	Channel[6:0]							DESTIN_
<b>Write:</b>								_INC[1]
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	DESTIN_	SOURC_INC[1:0]		CYCLE	MODE	SIZE[1:0]		CHN8E
<b>Write:</b>	INC[0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DESTIN_DIR	<b>DMA 通道 8 目的地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
SOURC_DIR	<b>DMA 通道 8 源地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
PRI[1:0]	<b>DMA 通道 8 优先级配置寄存器</b> 00: 低优先级 (default) 01: 中优先级 10: 高优先级 11: 最高优先级
Channel[6:0]	触发通道选择, 具体参见 DMA 通道请求列表来定
DESTIN_INC[1:0]	<b>DMA 通道 8 目的地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增

SOURC_INC[1:0]	<b>DMA 通道 8 源地址地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
CYCLE	<b>DMA 通道 8 循环模式</b> 0: 不循环模式 1: 循环模式
MODE	<b>DMA 通道 8 传输模式</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)
SIZE[1:0]	<b>DMA 通道 8 MEMORY 的传送位数 (外设的传送位数固定为 32bit):</b> 00: 8 位 01: 16 位 10: 32 位 11: 32 位
CHN8EN	<b>DMA 通道 8 使能</b> 0: 禁止 1: 使能

### CHN8SRC (DMA 通道 8 源地址寄存器)

CHN8SRC (DMA 通道 8 源地址寄存器)			基地址: 0x40020000 偏移地址: 110H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:	ADDR[31:24]							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:	ADDR[23:16]							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR[15:8]							
Write:	ADDR[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR[7:0]							
Write:	ADDR[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 8 数据传输源地址寄存器

### CHN8STAR (DMA 通道 8 目的地址寄存器)

CHN8STAR (DMA 通道 8 目的地址寄存器)			基地址: 0x40020000 偏移地址: 114H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 8 数据传输目的地址寄存器

### CHN8CNT (DMA 通道 8 传输数量寄存器)

CHN8CNT (DMA 通道 8 传输数量寄存器)			基地址: 0x40020000 偏移地址: 118H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0

<b>Read:</b>	Num[7:0]							
<b>Write:</b>	Num[7:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
Num[15:0]	<b>DMA 通道 8 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

### CHN8TCNT (DMA 通道 8 已传输数据个数)

<b>CHN8TCNT</b> (DMA 通道 8 已传输数据个数)			基地址: 0x40020000 偏移地址: 11CH					
	<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>	Num[15:8]							
<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>	Num[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
Num[15:0]	指示 DMA 通道 8 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

### CHN8BULKNUM (DMA 通道 8 块传输设置寄存器)

<b>CHN8BULKNUM</b> (DMA 通道 8 块传输设置寄存器)			基地址: 0x40020000 偏移地址: 120H					
	<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>	CYCLE[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>	Num[7:0]							
<b>Write:</b>								

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

位	功能描述
CYCLE[7:0]	通道 8 循环次数选择, 当用户选择 DMA 传输为循环模式时起作用: =0: 无限次循环 =0x01----0xFF: 为 0x01----0xFF 次循环
Num[7:0]	通道 8 块传输数据个数选择, 表示每一个数据块内有多少个数据: 0—255 个 如果用户设置的是块传输, 那么该寄存器则表示每一个数据块内有多少个数据

## CHN9CTL (DMA 通道 9 控制寄存器)

CHN9CTL (DMA 通道 9 控制寄存器)		基地址: 0x40020000 偏移地址: 124H						
	Bit23	22	21	20	19	18	17	Bit16
<b>Read:</b>	DESTIN_	SOURC_	X	X	X	X	PRI[1:0]	
<b>Write:</b>	DIR	DIR						
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	Channel[6:0]							DESTIN
<b>Write:</b>								_INC[1]
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	DESTIN_	SOURC_INC[1:0]		CYCLE	MODE	SIZE[1:0]		CHN9E
<b>Write:</b>	INC[0]							N
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DESTIN_DIR	<b>DMA 通道 9 目的地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
SOURC_DIR	<b>DMA 通道 9 源地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
PRI[1:0]	<b>DMA 通道 9 优先级配置寄存器</b> 00: 低优先级 (default) 01: 中优先级 10: 高优先级 11: 最高优先级
Channel[6:0]	触发通道选择, 具体参见 DMA 通道请求列表来定
DESTIN_INC[1:0]	<b>DMA 通道 9 目的地址增减模式</b>

]	00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
SOURC_INC[1:0]	<b>DMA 通道 9 源地址地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
CYCLE	<b>DMA 通道 9 循环模式</b> 0: 不循环模式 1: 循环模式
MODE	<b>DMA 通道 9 传输模式</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)
SIZE[1:0]	<b>DMA 通道 9 MEMORY 的传送位数 (外设的传送位数固定为 32bit):</b> 00: 8 位 01: 16 位 10: 32 位 11: 32 位
CHN9EN	<b>DMA 通道 9 使能</b> 0: 禁止 1: 使能

### CHN9SRC (DMA 通道 9 源地址寄存器)

<b>CHN9SRC</b> (DMA 通道 9 源地址寄存器)		基地址: <b>0x40020000</b> 偏移地址: <b>128H</b>						
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							
<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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 9 数据传输源地址寄存器

### CHN9TAR (DMA 通道 9 目的地址寄存器)

<b>CHN9TAR</b> (DMA 通道 9 目的地址寄存器)			基地址: 0x40020000 偏移地址: 12CH					
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							
<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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 9 数据传输目的地址寄存器

### CHN9CNT (DMA 通道 9 传输数量寄存器)

<b>CHN9CNT</b> (DMA 通道 9 传输数量寄存器)			基地址: 0x40020000 偏移地址: 130H					
--------------------------------------	--	--	-------------------------------	--	--	--	--	--

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

位	功能描述
Num[15:0]	<b>DMA 通道 9 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

### CHN9TCNT (DMA 通道 9 已传输数据个数)

<b>CHN9TCCNT</b> (DMA 通道 9 已传输数据个数)		基地址: 0x40020000 偏移地址: 134H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Num[15:0]	指示 DMA 通道 9 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

### CHN9BULKNUM (DMA 通道 9 块传输设置寄存器)

<b>CHN9BULKNUM</b> (DMA 通道 9 块传输设置寄存器)		基地址: 0x40020000 偏移地址: 138H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	CYCLE[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>	Num[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
CYCLE[7:0]	通道 9 循环次数选择，当用户选择 DMA 传输为循环模式时起作用： =0: 无限次循环 =0x01----0xFF: 为 0x01----0xFF 次循环
Num[7:0]	通道 9 块传输数据个数选择，表示每一个数据块内有多少个数据： 0 — 255 个 如果用户设置的是块传输，那么该寄存器则表示每一个数据块内有多少个数据

## CHN10CTL (DMA 通道 10 控制寄存器)

<b>CHN10CTL</b> (DMA 通道 10 控制寄存器)		基地址: 0x40020000 偏移地址: 13CH						
	<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>	DESTIN_	SOURC_	X	X	X	X	PRI[1:0]	
<b>Write:</b>	DIR	DIR						
<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>	Channel[6:0]							DESTIN_
<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_	SOURC_INC[1:0]		CYCLE	MODE	SIZE[1:0]		CHN10E
<b>Write:</b>	INC[0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DESTIN_DIR	<b>DMA 通道 10 目的地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
SOURC_DIR	<b>DMA 通道 10 源地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
PRI[1:0]	<b>DMA 通道 10 优先级配置寄存器</b> 00: 低优先级 (default)

	01: 中优先级 10: 高优先级 11: 最高优先级
Channel[6:0]	触发通道选择, 具体参见 DMA 通道请求列表来定
DESTIN_INC[1:0]	<b>DMA 通道 10 目的地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
SOURC_INC[1:0]	<b>DMA 通道 10 源地址地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
CYCLE	<b>DMA 通道 10 循环模式</b> 0: 不循环模式 1: 循环模式
MODE	<b>DMA 通道 10 传输模式</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)
SIZE[1:0]	<b>DMA 通道 10 MEMORY 的传送位数 (外设的传送位数固定为 32bit):</b> 00: 8 位 01: 16 位 10: 32 位 11: 32 位
CHN10EN	<b>DMA 通道 10 使能</b> 0: 禁止 1: 使能

## CHN10SRC (DMA 通道 10 源地址寄存器)

<b>CHN10SRC</b> (DMA 通道 10 源地址寄存器)			<b>基地址: 0x40020000</b>					
			<b>偏移地址: 140H</b>					
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							

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

位	功能描述
ADDR[31:0]	通道 10 数据传输源地址寄存器

### CHN10TAR (DMA 通道 10 目的地址寄存器)

<b>CHN10TAR</b> (DMA 通道 10 目的地址寄存器)		基地址: 0x40020000 偏移地址: 144H						
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							
<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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 10 数据传输目的地址寄存器

**CHN10CNT (DMA 通道 10 传输数量寄存器)**

<b>CHN10CNT</b> (DMA 通道 10 传输数量寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>148H</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>	Num[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>	Num[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
Num[15:0]	<b>DMA 通道 10 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

**CHN10TCNT (DMA 通道 10 已传输数据个数)**

<b>CHN10TCNT</b> (DMA 通道 10 已传输数据个数)			基地址: <b>0x40020000</b> 偏移地址: <b>14CH</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>	Num[15:8]							
<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>	Num[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
Num[15:0]	<b>指示 DMA 通道 10 已经传输完成的数据个数</b> 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

**CHN10BULKNUM (DMA 通道 10 块传输设置寄存器)**

<b>CHN10BULKNUM</b> (DMA 通道 10 块传输设置寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>150H</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>	CYCLE[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>	Num[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
CYCLE[7:0]	通道 10 循环次数选择, 当用户选择 DMA 传输为循环模式时起作用: =0: 无限次循环 =0x01----0xFF: 为 0x01----0xFF 次循环
Num[7:0]	通道 10 块传输数据个数选择, 表示每一个数据块内有多少个数据: 0—255 个 如果用户设置的是块传输, 那么该寄存器则表示每一个数据块内有多少个数据

**CHN11CTL (DMA 通道 11 控制寄存器)**

<b>CHN11CTL</b> (DMA 通道 11 控制寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>154H</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>	DESTIN_	SOURC_	X	X	X	X	PRI[1:0]	
<b>Write:</b>	DIR	DIR						
<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>	Channel[6:0]							DESTIN_
<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_	SOURC_INC[1:0]		CYCLE	MODE	SIZE[1:0]		CHN11E N
<b>Write:</b>	INC[0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DESTIN_DIR	DMA 通道 11 目的地址增减方向选择:

	0: 地址递增模式 (default) 1: 地址递减模式
SOURC_DIR	<b>DMA 通道 11 源地址增减方向选择:</b> 0: 地址递增模式 (default) 1: 地址递减模式
PRI[1:0]	<b>DMA 通道 11 优先级配置寄存器</b> 00: 低优先级 (default) 01: 中优先级 10: 高优先级 11: 最高优先级
Channel[6:0]	触发通道选择, 具体参见 DMA 通道请求列表来定
DESTIN_INC[1:0]	<b>DMA 通道 11 目的地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
SOURC_INC[1:0]	<b>DMA 通道 11 源地址地址增减模式</b> 00: 固定 01: 递增/递减 10: 数据块内循环递增/递减 11: 以整个数据块为单位递增/递减 注: 递增/递减选择由控制位 DESTIN_DIR 控制, 默认递增
CYCLE	<b>DMA 通道 11 循环模式</b> 0: 不循环模式 1: 循环模式
MODE	<b>DMA 通道 11 传输模式</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)
SIZE[1:0]	<b>DMA 通道 11 MEMORY 的传送位数 (外设的传送位数固定为 32bit):</b> 00: 8 位 01: 16 位 10: 32 位 11: 32 位
CHN11EN	<b>DMA 通道 11 使能</b> 0: 禁止 1: 使能

## CHN11SRC (DMA 通道 11 源地址寄存器)

CHN11SRC	基地址: 0x40020000
----------	-----------------

(DMA 通道 11 源地址寄存器)			偏移地址: 158H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 11 数据传输源地址寄存器

### CHN11TAR (DMA 通道 11 目的地址寄存器)

CHN11TAR (DMA 通道 11 目的地址寄存器)			基地址: 0x40020000 偏移地址: 15CH					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR[7:0]							
Write:								

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

位	功能描述
ADDR[31:0]	通道 11 数据传输目的地址寄存器

### CHN11CNT (DMA 通道 11 传输数量寄存器)

<b>CHN11CNT</b> (DMA 通道 11 传输数量寄存器)			基地址: 0x40020000 偏移地址: 160H					
	<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>	Num[15:8]							
<b>Write:</b>	Num[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>	Num[7:0]							
<b>Write:</b>	Num[7:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
Num[15:0]	<b>DMA 通道 11 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

### CHN11TCNT (DMA 通道 11 已传输数据个数)

<b>CHN11TCNT</b> (DMA 通道 11 已传输数据个数)			基地址: 0x40020000 偏移地址: 164H					
	<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>	Num[15:8]							
<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>	Num[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
Num[15:0]	指示 DMA 通道 11 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

## CHN11BULKNUM (DMA 通道 11 块传输设置寄存器)

CHN11BULKNUM (DMA 通道 11 块传输设置寄存器)		基地址: 0x40020000 偏移地址: 168H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	CYCLE[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CYCLE[7:0]	通道 11 循环次数选择，当用户选择 DMA 传输为循环模式时起作用： =0: 无限次循环 =0x01----0xFF: 为 0x01----0xFF 次循环
Num[7:0]	通道 11 块传输数据个数选择，表示每一个数据块内有多少个数据： 0 — 255 个 如果用户设置的是块传输，那么该寄存器则表示每一个数据块内有多少个数据

## DMAIE2 (DMA 中断使能寄存器 2)

DMAIE2 (DMA 中断使能寄存器 2)		基地址: 0x40020000 偏移地址: 200H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	TEIE12
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	BCIE12	X	X	X	PEIE12

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

位	功能描述
TEIE12	通道 12 输错误中断使能 0: 禁止 1: 使能
BCIE12	通道 12 块传输中断使能 0: 禁止 1: 使能
PEIE12	通道 12 包传输中断使能（完成 1 次所配置通道的 pack 包传输中断） 0: 禁止 1: 使能

## DMAIF2（DMA 中断标志寄存器 2）

DMAIF2 (DMA 中断标志寄存器 2)		基地址: 0x40020000 偏移地址: 204H						
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	X	X	X	X	X	X	X	TEIF12
<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	BCIF12	X	X	X	PEIF12
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
BCIF12	通道 12 块传输完成中断标志 0: 未产生中断 1: 产生中断
TEIF12	通道 12 传输错误中断标志 0: 未产生中断 1: 产生中断
PEIF12	通道 12 包传输中断标志（完成 1 次所配置通道的 pack 包传输中断） 0: 未产生中断 1: 产生中断

## CHN12CTL (通道 12 控制寄存器)

CHN12CTL (通道 12 控制寄存器)			基地址: 0x40020000 偏移地址: 208H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	X	X	X	PRI[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	Channel[6:0]							X
Write:								
Reset:	0	1	1	1	1	1	1	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	FFTTRI	CHN12E
Write:							G_EN	N
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PRI[1:0]	<b>DMA 通道 12 优先级配置寄存器</b> 00: 低优先级 (default) 01: 中优先级 10: 高优先级 11: 最高优先级
Channel[6:0]	触发通道选择, 具体参见 DMA 通道请求列表来定 通道 12 为专用通道, 只支持 emu 的 FFT 数据完成请求, Channel[6:0]配置成 63。
FFTTRIG_EN	传输指定数量的数据之后, 如果该位置 1, 将触发 FFT 进行计算。 0: 禁止 1: 使能
CHN12EN	<b>DMA 通道 12 使能</b> 0: 禁止 1: 使能

## CHN12SRC (DMA 通道 12 源地址寄存器)

CHN12SRC (DMA 通道 12 源地址寄存器)			基地址: 0x40020000 偏移地址: 20CH					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							

<b>Write:</b>								
<b>Reset:</b>	0	1	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>	ADDR[23:16]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	1	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>	ADDR[15:8]							
<b>Write:</b>								
<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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	1	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 12 数据传输源地址寄存器(0x40022020)

### CHN12TAR0 (DMA 通道 12 目的地址寄存器 0)

<b>CHN12TAR0</b> (DMA 通道 12 目的地址寄存器)			<b>基地址: 0x40020000</b> <b>偏移地址: 210H</b>					
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							
<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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 12 数据传输目的地址寄存器 0

### CHN12TAR1 (DMA 通道 12 目的地址寄存器 1)

CHN12TAR1 (DMA 通道 12 目的地址寄存器 1)			基地址: 0x40020000 偏移地址: 214H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 12 数据传输目的地址寄存器 1

### CHN12TAR2 (DMA 通道 12 目的地址寄存器 2)

CHN12TAR2 (DMA 通道 12 目的地址寄存器 2)			基地址: 0x40020000 偏移地址: 218H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:								
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>	ADDR[23:16]							
<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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 12 数据传输目的地址寄存器 2

### CHN12TAR3 (DMA 通道 12 目的地址寄存器 3)

<b>CHN12TAR3</b> (DMA 通道 12 目的地址寄存器 3)	基地址: 0x40020000 偏移地址: 21CH							
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							
<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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 12 数据传输目的地址寄存器 3

**CHN12TAR4 (DMA 通道 12 目的地址寄存器 4)**

CHN12TAR4 (DMA 通道 12 目的地址寄存器 4)			基地址: 0x40020000 偏移地址: 220H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADDR[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADDR[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 12 数据传输目的地址寄存器 4

**CHN12TAR5 (DMA 通道 12 目的地址寄存器 5)**

CHN12TAR5 (DMA 通道 12 目的地址寄存器 5)			基地址: 0x40020000 偏移地址: 224H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	ADDR[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	ADDR[23:16]							

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

位	功能描述
ADDR[31:0]	通道 12 数据传输目的地址寄存器 5

### CHN12TAR6 (DMA 通道 12 目的地址寄存器 6)

<b>CHN12TAR6</b> (DMA 通道 12 目的地址寄存器 6)			基地址: <b>0x40020000</b> 偏移地址: <b>228H</b>					
	<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>	ADDR[31:24]							
<b>Write:</b>								
<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>	ADDR[23:16]							
<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>	ADDR[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>	ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
ADDR[31:0]	通道 12 数据传输目的地址寄存器 6

**CHN12SIZE (通道 12 长度控制寄存器)**

CHN12SIZE(通道 12 长度控制寄存器)		基地址: 0x40020000 偏移地址: 22CH						
	Bit31	30	29	28	27	26	25	Bit24
Read:	X	CHN_NUM[2:0]			X	INTSIZE[10:8]		
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	INTSIZE[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	BUFSIZE[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	BUFSIZE[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CHN_NUM[2:0]	emu 采样数据中需要被传输的通道数量 配置 CHN12SIZE.CHN_NUM[2:0]=x (x=1~7), 确定要从 EMU 接口地址处传输 x (x=1~7) 个通道的数据; CHN_NUM[2:0]=0 时, 数字内部作为值=7 处理。
INTSIZE[10:0]	dma 传输指定数量的数据之后, 触发中断。单位为 word。这个根据 fft 长度来决定。
BUFSIZE[15:0]	data memory 中用于存放一个通道数据所开辟的 memory 长度。单位为 word。

**CHN12CLR (清除通道 12 使能寄存器)**

CHN12CLR(清除通道 12 使能寄存器)		基地址: 0x40020000 偏移地址: 230H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X							CLR_EN

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

位	功能描述
CLR_EN	该位写 1,通道 12 的 CHN12CTL 的位 CHN12EN 清 0; 该寄存器只写寄存器,读该寄存器无意义。

注: 当 DMA 响应了 EMU 的请求, 写 1 清 0, 不会立即生效; 一包数据传输完成之后, CHN12EN=0, 如果 DMA 没响应 EMU 的请求, 写 1 清 0, CHN12EN=0 立即生效。

### CHN12EMUOFFSET (搬运 EMU 数据偏移量寄存器)

CHN12EMUOFFSET (搬运 EMU 通道 0 数据偏移量寄存器)			<b>基地址: 0x40020000</b> <b>偏移地址: 238H</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>	EMUOFFSET[15:8]							
<b>Write:</b>	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>	EMUOFFSET[7:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
EMUOFFSET[15:0]	当前 DMA 搬运了 EMU 的数据偏移量寄存器

### CHN12IDX(通道 12 当前搬移数据的通道 ID 寄存器)

CHN12IDX(通道 12 当前搬移数据的通道 ID 寄存器)			<b>基地址: 0x40020000</b> <b>偏移地址: 23CH</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							
<b>Write:</b>	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						CHIDX[2:0]	
<b>Write:</b>	X						X	
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
CHIDX[2:0]	读寄存器该位:表示当前正在搬移的数据属于哪个 emu 通道的数据

### CHN12INTCNT(通道 12 已完成数据搬移个数寄存器)

CHN12INTCNT(通道 12 已完成数据搬移个数寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>240H</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					INT_CNT[10:8]		
<b>Write:</b>						X		
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	INT_CNT[7:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
INT_CNT[10:0]	指示当前通道 12 已完成数据搬移个数计数值 注: 该计数会随 DMA 的搬运数据累加, 反映当前的搬运数据个数 (word)。从 0 累加至 INTSIZE[10:0]-1 后自动清零, DMA 继续搬运时, 从 0 继续累加。

### CHN12PACKCNT (当前 emu 的 chx 是包中第几个数据寄存器)

CHN12PACKCNT (当前 emu 的 chx 是包中第几个数据寄存器)			基地址: <b>0x40020000</b> 偏移地址: <b>244H</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							
<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						PACK_LEN[1:0]	
<b>Write:</b>							X	
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
PACK_LEN[1:0]	当前 emu 的 chx 发送的是包中第几个数据。

**CHN12STARTOFFSET(通道 12 目的地址起始地址偏移寄存器)**

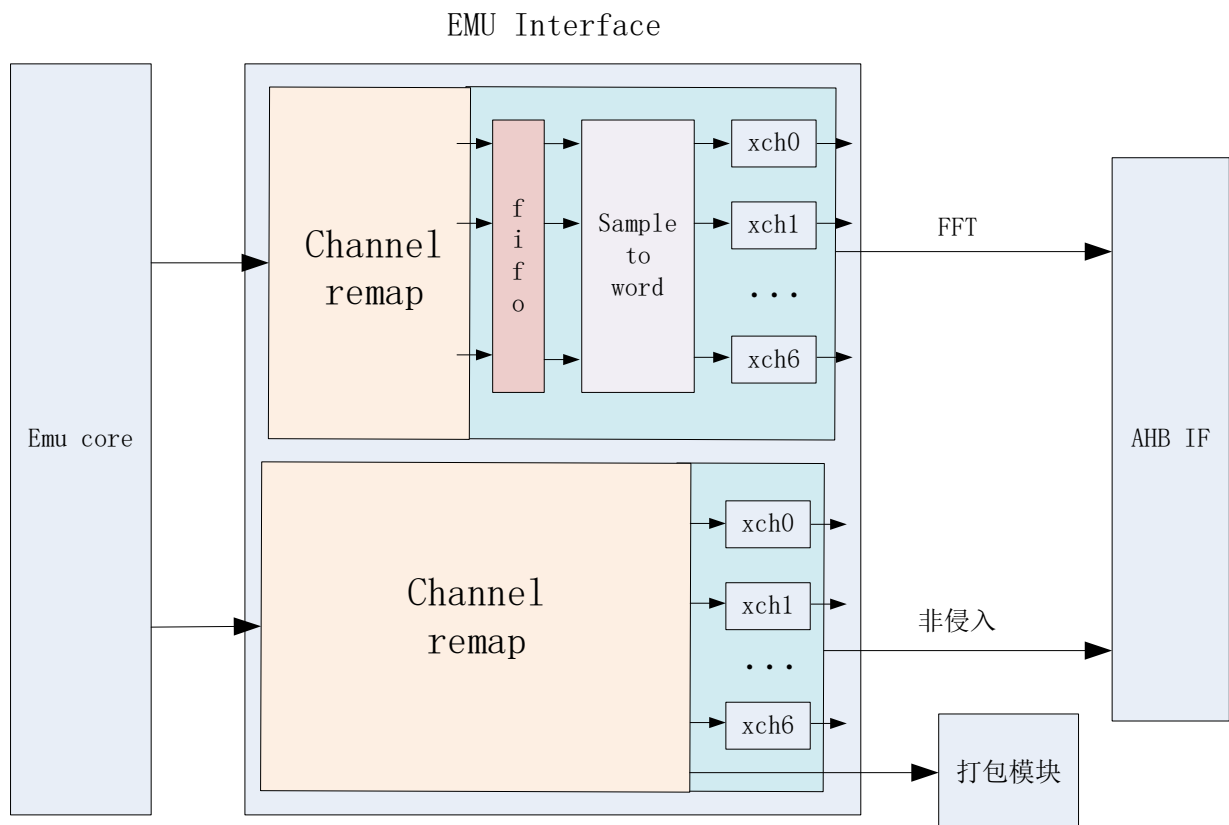
CHN12STARTOFFSET(通道 12 目的地址起始地址偏移寄存器)		基地址: <b>0x40020000</b> 偏移地址: <b>248H</b>						
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	STARTOFFSET[15:8]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	STARTOFFSET[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
STARTOFFSET[15:0]	目的地址起始地址相对目的地址基址偏移量寄存器,DMA 搬运的起始地址是 <b>CHN12TARi(i=0,1,2,3,4,5,6,7)+ STARTOFFSET</b> 值。 注: 当 DMA 完成 CHN12SIZE.INTSIZE[10:0]所配置的数据长度传输后, 硬件会自动更新此寄存器的值, 用于提示用户新数据的起始偏移位置。

## 19 EMU Interface 模块

### 19.1 概述

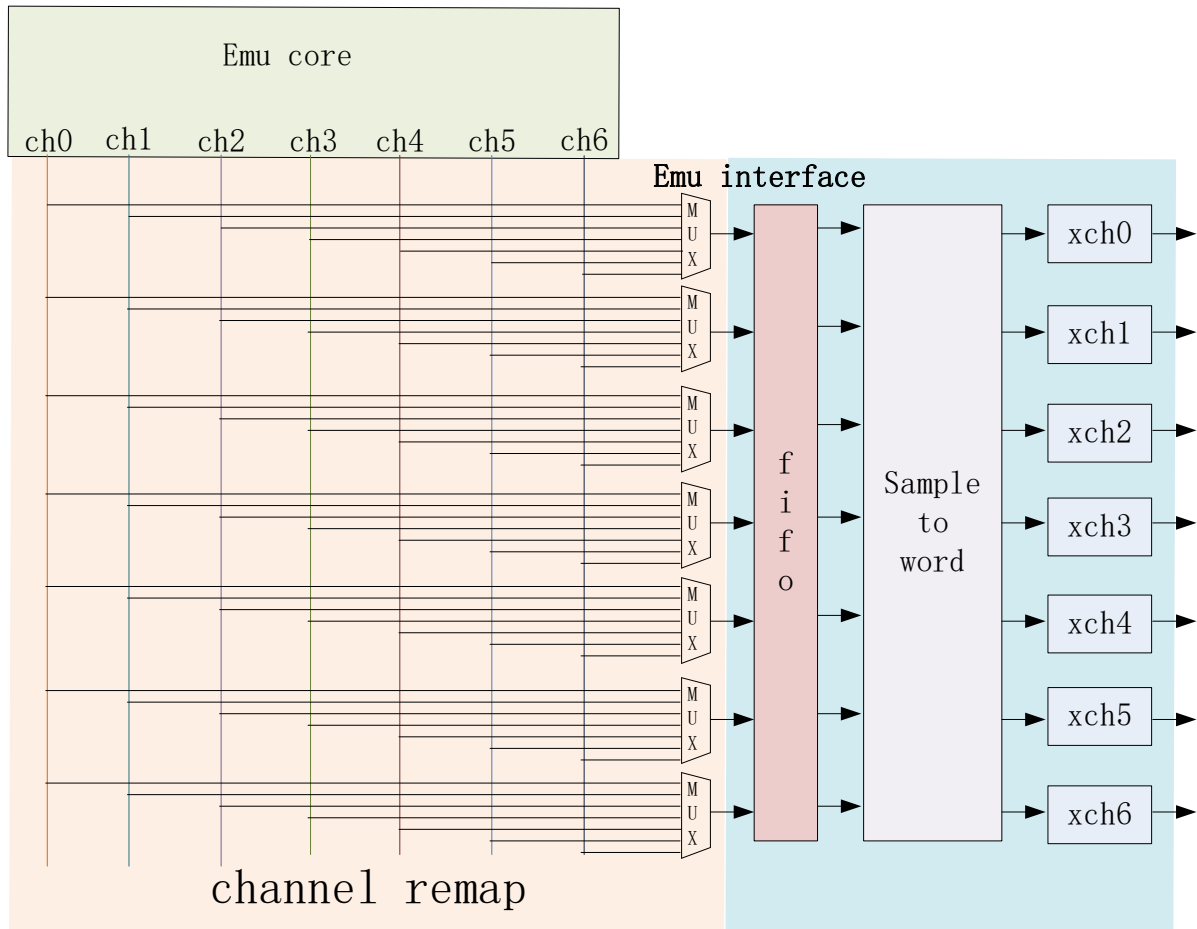
EMU Interface 模块将经过工频同步系统的 EMU ADC 采样数据进行通道重映射及打包等处理后，提供特定格式的数据，给到 FFT 模块和非侵入数据帧打包模块。模块框图如下所示



EMU interface 模块提供一路用于 FFT 的专用数据接口：FFT 数据接口每个通道具有 8word 深度的 FIFO 用于缓冲数据，并且具有根据输出数据宽度对数据进行拼接处理功能，详见后续 19.3 FFT 数据接口相关说明。

EMU interface 模块提供一路用于非侵入相关应用的专用数据接口：非侵入数据从 EMU 输出后经 remap 后直接通过独立数据总线输出至打包模块，同时也会更新在非侵入的数据接口上。**当使用打包模块进行非侵入数据进行处理传输时，读取非侵入端口上的数据不会影响打包模块获得的数据；**当不使用打包模块进行非侵入数据处理传输时，用户可使用 DMA 或软件读取接口上的数据自行组包传输。

数据通道重映射可以实现将 EMU 原始采样数据进行任意顺序组合排列输出，内部结构如下图所示（仅 FFT 接口具有 FIFO 和 sample to word）。



EMU Interface 的数据接口有效通道数可 1-7 自由配置，当接口数据准备好时，会给出 IRQ 信号，可以产生中断或触发 DMA 进行搬运。在读取时需注意，当读取最后一个有效通道的最后一个数据时，整个接口上的数据会被更新；建议按照接口的地址分配依次读取，避免数据读取错误。

注：

- 1) FFT 接口上的数据在未及时读取时，后续数据会在 FIFO 中缓存，当 FIFO 发生溢出时，会导致后续数据无法写入而丢失；非侵入接口上的数据在未及时读取时，后续数据会在接口上覆盖旧的数据。
- 2) **FFT 接口上的数据读取更新**：若 FIFO 中还有数据，则会立即将新的数据更新至端口；若 FIFO 中最后一个数据被读取，则接口上会保持为最后一个数据，直到在下一个有效数据写入后才更新为新的数据。
- 3) **非侵入接口上的数据读取更新**：读取后接口上数据会直到在下一个有效数据写入后才更新为新的数据。
- 4) FFT 接口数据会根据配置的有效通道数进行更新，例如，配置 FFT 数据接口有效通道数为 3，则每次会更新 FFT 输出数据接口 0~2，输出数据接口 3~6 一直保持固定值，不会更新。（FFT 数据接口未使能的输出通道值为随机值）
- 5) 非侵入接口数据每次更新均会同步更新所有输出数据接口上的值，更新的有效数据取决于 EMU 模块配置的有效输出通道。

## 19.2 使用流程

### 19.2.1 FFT 数据接口配置使用

1. FFT 数据接口建议配合专用 DMA 通道使用，FFT 接口有效输出通道数由 DMA 模块下 CHN12SIZE.CHN\_NUM[2:0]控制(0x4002022C, bit 28~30);FFT 数据接口位宽由 EMU 模块下 SYNC\_CFG. SYNC\_DMA\_bw\_mode[1:0]控制 (0x4001CC00, bit 16~17) ; 当配置成 32bit mode (SYNC\_CFG. SYNC\_DMA\_bw\_mode=10 和 11) 时, 可选择配置 FFT 数据接口高对齐使能位 (仅在 32bits 模式下生效) EICON.EI\_FFTHA。

EMU 模块 SYNC_CFG. SYNC_DMA_bw_mode[1:0]	EMU_IF 端口数 据宽度
00: 16bit	16bit
01: 24bit	24bit
10: 32bit	32bit
11: 32bit	32bit

2. 配置 FFT 数据接口输出通道 n 数据源选择寄存器 (FFT 数据接口输出通道 n 选择 EMU 采样通道 CHx 的数据) EI\_FFTCFG. FFT\_CHNSEL0, EI\_FFTCFG. FFT\_CHNSEL1, EI\_FFTCFG. FFT\_CHNSEL2, EI\_FFTCFG. FFT\_CHNSEL3, EI\_FFTCFG. FFT\_CHNSEL4, EI\_FFTCFG. FFT\_CHNSEL5, EI\_FFTCFG. FFT\_CHNSEL6。
3. 可配置 FFT 数据接口更新中断使能 EIIE. EI\_FFTE=1。
4. FFT 数据接口使能 EICON.EI\_FFTE=1。

注：由于 FFT 数据接口 FIFO 溢出会导致数据丢失，因此在配合 FFT 及 DMA 使用时，建议在硬件自动 FFT 模式下，先配置并使能 FFT 模块，再配置并使能 DMA，最后配置并使能 EMU interface；在软件使能 FFT 模式下，先配置 FFT、DMA、EMU interface 模块，再依次使能 DMA 和 EMU interface，最后在 DMA 完成后触发软件使能 FFT 模块。

### 19.2.2 非侵入数据接口配置使用

1. 配置非侵入数据接口位宽 EICON.EI\_NISMODE；当配置成 32bit mode (EICON.EI\_NISMODE==10/11) 时, 可选择配置非侵入数据接口高对齐使能位 (仅在 32bits 模式下生效) EICON.EI\_NISHA。
2. 配置 非侵入数据接口输出通道 n 数据源选择寄存器(非侵入数据接口输出通道 n 选择 EMU 数据通道 CHx 的数据)EI\_NISCFG.NIS\_CHNSEL0,EI\_NISCFG.NIS\_CHNSEL1,EI\_NISCFG.NIS\_CHNSEL2,EI\_NISCFG.NIS\_CHNSEL3, EI\_NISCFG.NIS\_CHNSEL4, EI\_NISCFG.NIS\_CHNSEL5, EI\_NISCFG.NIS\_CHNSEL6。
3. 可配置非侵入数据接口更新中断使能 EIIE. EI\_NISIE=1。
4. 非侵入数据接口使能 EICON.EI\_NISEN=1。

注：由于非侵入数据接口没有 FIFO，未及时读取会导致数据被后续数据覆盖而丢失，因此在应用时，建议先配置并使能 SPI 和打包模块或 DMA，然后配置并使能 EMU interface。

## 19.3 FFT 数据接口

### 19.3.1 16bits mode 时的 FFT 数据接口地址分配

在 16bit 模式下，FFT 数据接口会取 EMU 数据的高 16bits 并将 2 次采样数据映射至对应的端口地址上，然后给出 IRQ 信号。

如下表所示，若使能了 7 个有效通道数据接口，可从每个端口地址一次读出 2 个采样数据，在读取 offset 在 0x38 地址上的 32bit 数据时，整个接口上的数据才会更新。

offset	channel	word			
		Byte3	Byte2	Byte1	Byte0
0x20	xch0	sample N+1		sample N	
0x24	xch1	sample N+1		sample N	
0x28	xch2	sample N+1		sample N	
0x2C	xch3	sample N+1		sample N	
0x30	xch4	sample N+1		sample N	
0x34	xch5	sample N+1		sample N	
0x38	xch6	sample N+1		sample N	

### 19.3.2 24bits mode 时的 FFT 数据接口地址分配

在 24bit 模式下，FFT 数据接口会将 4 次采样数据组合映射至对应的端口地址上，然后给出 IRQ 信号。

如下表所示，若使能了 7 个有效通道数据接口，需从每个端口地址读 3 次共计读出 4 个采样数据，在读取 offset 在 0x70 地址上的 32bit 数据时，整个接口上的数据才会更新。

offset	channel	word			
		Byte3	Byte2	Byte1	Byte0
0x20	xch0	sample N+1	sample N		
0x24		sample N+2		sample N+1	
0x28		sample N+3			sample N+2
0x2C	xch1	sample N+1	sample N		
0x30		sample N+2		sample N+1	
0x34		sample N+3			sample N+2
0x38	xch2	sample N+1	sample N		
0x3c		sample N+2		sample N+1	
0x40		sample N+3			sample N+2
0x44	xch3	sample N+1	sample N		
0x48		sample N+2		sample N+1	
0x4C		sample N+3			sample N+2
0x50	xch4	sample N+1	sample N		

0x54		sample N+2	sample N+1	
0x58		sample N+3		sample N+2
0x5C	xch5	sample N+1	sample N	
0x60		sample N+2	sample N+1	
0x64	xch6	sample N+3		sample N+2
0x68		sample N+1	sample N	
0x6C		sample N+2	sample N+1	
0x70		sample N+3		sample N+2

### 19.3.3 32bits mode 时的 FFT 数据接口地址分配 MSB

在 32bit 高对齐模式(EICON.EI\_FFTHA=1)下, FFT 数据接口将 EMU 采样数据映射至对应的端口地址的高 24bit 上, 低 8bit 补 0, 然后给出 IRQ 信号。

如下表所示, 若使能了 7 个有效通道数据接口, 可从每个端口地址读出 1 个采样数据, 在读取 offset 在 0x38 地址上的 32bit 数据时, 整个接口上的数据才会更新。

offset	channel	word			
		Byte3	Byte2	Byte1	Byte0
0x20	xch0	sample N			补 0
0x24	xch1	sample N			补 0
0x28	xch2	sample N			补 0
0x2C	xch3	sample N			补 0
0x30	xch4	sample N			补 0
0x34	xch5	sample N			补 0
0x38	xch6	sample N			补 0

### 19.3.4 32bits mode 时的 FFT 数据接口地址分配 LSB

在 32bit 低对齐模式(EICON.EI\_FFTHA=0)下, FFT 数据接口将 EMU 采样数据映射至对应的端口地址的低 24bit 上, 高 8bit 补符号位, 然后给出 IRQ 信号。

如下表所示, 若使能了 7 个有效通道数据接口, 可从每个端口地址读出 1 个采样数据, 在读取 offset 在 0x38 地址上的 32bit 数据时, 整个接口上的数据才会更新。

offset	channel	word			
		Byte3	Byte2	Byte1	Byte0
0x20	xch0	补符号位	sample N		
0x24	xch1	补符号位	sample N		
0x28	xch2	补符号位	sample N		
0x2C	xch3	补符号位	sample N		
0x30	xch4	补符号位	sample N		
0x34	xch5	补符号位	sample N		
0x38	xch6	补符号位	sample N		

## 19.4 非侵入数据接口

### 19.4.1 16bits mode 时的地址分配

在 16bit 模式下，非侵入数据接口将 EMU 采样数据的高 16bits 映射至对应的端口地址上，然后给出 IRQ 信号。建议此模式下，使用 16bits 位宽进行访问。如下表所示，使能 7 个有效通道数据接口，可从每个端口地址读出 16bits 的采样数据。

offset	Half word	
	Byte1	Byte0
0x80	xch0	
0x82	xch1	
0x84	xch2	
0x86	xch3	
0x88	xch4	
0x8A	xch5	
0x8C	xch6	

### 19.4.2 24bits mode 时的地址分配

在 24bit 模式下，非侵入数据接口将 EMU 采样数据依次映射至对应的端口地址上，然后给出 IRQ 信号。如下表所示，使能 7 个有效通道时的接口上 24bit 数据会依次拼接。

offset	word			
	Byte3	Byte2	Byte1	Byte0
0x80	xch1	xch0		
0x84	xch2		xch1	
0x88	xch3			xch2
0x8C	xch5	xch4		
0x90	xch6		xch5	
0x94	0			xch6

### 19.4.3 32bits mode 时的地址分配 MSB

在 32bit 高对齐模式下，非侵入数据接口将 EMU 采样数据映射至对应的端口地址的高 24bit 上，低 8bit 补 0，然后给出 IRQ 信号。如下表所示，使能 7 个有效通道数据接口，可从每个端口地址读出 32bits 的采样数据。

offset	word			
	Byte3	Byte2	Byte1	Byte0
0x80	xch0			补 0
0x84	xch1			补 0
0x88	xch2			补 0

<b>0x8C</b>	xch3	补 0
<b>0x90</b>	xch4	补 0
<b>0x94</b>	xch5	补 0
<b>0x98</b>	xch6	补 0

### 19.4.4 32bits mode 时的地址分配 LSB

在 32bit 低对齐模式下，非侵入数据接口将 EMU 采样数据映射至对应的端口地址的低 24bit 上，高 8bit 补符号位，然后给出 IRQ 信号。如下表所示，使能 7 个有效通道数据接口，可从每个端口地址读出 32bits 的采样数据。

offset	word			
	Byte3	Byte2	Byte1	Byte0
<b>0x80</b>	补符号位	xch0		
<b>0x84</b>	补符号位	xch1		
<b>0x88</b>	补符号位	xch2		
<b>0x8C</b>	补符号位	xch3		
<b>0x90</b>	补符号位	xch4		
<b>0x94</b>	补符号位	xch5		
<b>0x98</b>	补符号位	xch6		

## 19.5 特殊功能寄存器列表

EMU Interface 模块寄存器基地址： 0x40022000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	EICON	R/W	0x7000	EMU Interface 控制寄存器
0x04	EIIE	R/W	0x0000	EMU Interface 中断使能寄存器
0x08	EIIF	R/W	0x0000	EMU Interface 中断标志寄存器
0x0C	EL_FFTCFG	R/W	0x0000	FFT 数据接口配置寄存器
0x10	EL_NISCFG	R/W	0x0000	非侵入数据接口配置寄存器
0x20+0x04*i	FFT_DATAi			FFT 数据接口输出数据寄存器 i(i=0~20)
0x80+0x04*i	NIS_DATAi			非侵入数据接口输出数据寄存器 i(i=0~6)

## 19.6 特殊功能寄存器说明

### EICON (EMU Interface 控制寄存器)

EICON	基地址： 0x4002 2000
-------	------------------

(EMU Interface 控制寄存器)		偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	Reserved			EI_NISMODE[1:0]		EI_NISHA	EI_NISEN
Write:		Reserved			Reserved		EI_FFTHA	EI_FFTE
Reset:	0	1	1	1	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	EI_FFTR	Reserved			Reserved		EI_FFTHA	EI_FFTE
Write:	ST	Reserved			Reserved		EI_FFTHA	EI_FFTE
Reset:	0	0	0	0	0	0	0	0

位	功能描述
EI_NISMODE[1:0]	非侵入数据接口位宽配置位 00: 16bits mode 01: 24bits mode 1x: 32bits mode
EI_NISHA	非侵入数据接口高对齐使能位 (仅在32bits模式下生效) 0: 非侵入数据接口低位对齐 1: 非侵入数据接口高位对齐
EI_NISEN	非侵入数据接口使能位 0: 不使能非侵入数据接口 (default) 1: 使能非侵入数据接口
EI_FFTRST	FFT数据接口复位控制位 写 1 复位 FFT 数据接口状态机, 硬件自动清 0.
EI_FFTHA	FFT数据接口高对齐使能位 (仅在32bits模式下生效) 0: FFT数据接口低位对齐 1: FFT 数据接口高位对齐
EI_FFTE	FFT数据接口使能位 0: 不使能FFT数据接口 (default) 1: 使能 FFT 数据接口

## EIIE (EMU Interface 中断使能寄存器)

EIIE (EMU Interface 中断使能寄存器)		基地址: 0x4002 2000 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

<b>Read:</b>	X	X	X	X	X	X	EI_NISI	EI_FFTI
<b>Write:</b>							E	E
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
EI_NISIE	<b>非侵入数据接口更新中断使能位</b> 0: 不使能非侵入数据接口更新中断 1: 使能非侵入数据接口更新中断
EI_FFTIE	<b>FFT 数据接口更新中断使能位</b> 0: 不使能 FFT 数据接口更新中断 1: 使能 FFT 入数据接口更新中断

### EIIF (EMU Interface 中断标志寄存器)

<b>EIIF (EMU Interface 中断标志寄存器)</b>	<b>基地址: 0x4002 2000</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	EI_NISI	EI_FFTI
<b>Write:</b>							F	F
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
EI_NISIF	<b>非侵入数据接口更新中断标志位 (写0清0)</b> 0: 非侵入数据接口上的数据未更新 1: 非侵入数据接口上的数据更新
EI_FFTIF	<b>FFT 数据接口更新中断标志位 (写0清0)</b> 0: FFT 数据接口上的数据未更新 1: FFT 数据接口上的数据更新

### EI\_FFTCFG (EMU Interface 模块 FFT 数据接口配置寄存器)

<b>EI_FFTCFG (EMU Interface 模块 FFT 数据接口配置寄存器)</b>	<b>基地址: 0x4002 2000</b>
	<b>偏移地址: 0CH</b>

	<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>	X	X	X	X	X	FFT_CHNSEL6[2:0]		
<b>Write:</b>								
<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	FFT_CHNSEL5[2:0]			X	FFT_CHNSEL4[2:0]		
<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>	X	FFT_CHNSEL3[2:0]			X	FFT_CHNSEL2[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	FFT_CHNSEL1[2:0]			X	FFT_CHNSEL0[2:0]		
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
FFT_CHNSEL6[2:0]	<b>FFT 数据接口输出通道 6 数据源选择寄存器</b> FFT数据接口输出通道6选择EMU采样通道CHx的数据，x= FFT_CHNSEL6[2:0] 000: 输出通道6选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;
FFT_CHNSEL5[2:0]	<b>FFT 数据接口输出通道 5 数据源选择寄存器</b> FFT数据接口输出通道5选择EMU采样通道CHx的数据，x= FFT_CHNSEL5[2:0] 000: 输出通道选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;
FFT_CHNSEL4[2:0]	<b>FFT 数据接口输出通道 4 数据源选择寄存器</b> FFT数据接口输出通道4选择EMU采样通道CHx的数据，x= FFT_CHNSEL4[2:0] 000: 输出通道选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;
FFT_CHNSEL3[2:0]	<b>FFT 数据接口输出通道 3 数据源选择寄存器</b> FFT数据接口输出通道3选择EMU采样通道CHx的数据，x= FFT_CHNSEL3[2:0] 000: 输出通道选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;
FFT_CHNSEL2[2:0]	<b>FFT 数据接口输出通道 2 数据源选择寄存器</b> FFT数据接口输出通道2选择EMU采样通道CHx的数据，x= FFT_CHNSEL2[2:0] 000: 输出通道选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;
FFT_CHNSEL1[2:0]	<b>FFT 数据接口输出通道 1 数据源选择寄存器</b> FFT数据接口输出通道1选择EMU采样通道CHx的数据，x= FFT_CHNSEL1[2:0] 000: 输出通道选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;
FFT_CHNSEL0[2:0]	<b>FFT 数据接口输出通道 0 数据源选择寄存器</b>

2:0]	FFT数据接口输出通道0选择EMU采样通道CHx的数据，x= FFT_CHNSEL0[2:0] 000: 输出通道选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;
------	---

注意事项：采样通道 CH0-CH6 分别代表 Ua,Ub,Uc,Ia,Ib,Ic,In

## EI\_NISCFG (EMU Interface 模块非侵入数据接口配置寄存器)

EI_NISCFG (EMU Interface 模块非侵入数据接口配置寄存器)		基地址: 0x4002 2000 偏移地址: 10H						
	Bit31	30	29	28	27	26	25	Bit24
<b>Read:</b>	X	X	X	X	X	NIS_CHNSEL6[2:0]		
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
<b>Read:</b>	X	NIS_CHNSEL5[2:0]			X	NIS_CHNSEL4[2:0]		
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	X	NIS_CHNSEL3[2:0]			X	NIS_CHNSEL2[2:0]		
<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	NIS_CHNSEL1[2:0]			X	NIS_CHNSEL0[2:0]		
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
NIS_CHNSEL6[2:0]	<b>非侵入数据接口输出通道 6 数据源选择寄存器</b> 非侵入数据接口输出通道6选择EMU采样通道CHx的数据，x= NIS_CHNSEL6[2:0] 000: 输出通道6选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;
NIS_CHNSEL5[2:0]	<b>非侵入数据接口输出通道 5 数据源选择寄存器</b> 非侵入数据接口输出通道5选择EMU采样通道CHx的数据，x= NIS_CHNSEL5[2:0] 000: 输出通道选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;
NIS_CHNSEL4[2:0]	<b>非侵入数据接口输出通道 4 数据源选择寄存器</b> 非侵入数据接口输出通道4选择EMU采样通道CHx的数据，x= NIS_CHNSEL4[2:0] 000: 输出通道选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;
NIS_CHNSEL3[2:0]	<b>非侵入数据接口输出通道 3 数据源选择寄存器</b> 非侵入数据接口输出通道3选择EMU采样通道CHx的数据，x= NIS_CHNSEL3[2:0]

	000: 输出通道选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;
NIS_CHNSEL2[2:0]	<b>非侵入数据接口输出通道 2 数据源选择寄存器</b> 非侵入数据接口输出通道2选择EMU采样通道CHx的数据, x= NIS_CHNSEL2[2:0] 000: 输出通道选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;
NIS_CHNSEL1[2:0]	<b>非侵入数据接口输出通道 1 数据源选择寄存器</b> 非侵入数据接口输出通道1选择EMU采样通道CHx的数据, x= NIS_CHNSEL1[2:0] 000: 输出通道选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;
NIS_CHNSEL0[2:0]	<b>非侵入数据接口输出通道 0 数据源选择寄存器</b> 非侵入数据接口输出通道0选择EMU采样通道CHx的数据, x= NIS_CHNSEL0[2:0] 000: 输出通道选择EMU采样通道CH0的数据 (default) 111: 数字内部处理成同 000;

注意事项: 采样通道 CH0-CH6 分别代表 Ua,Ub,Uc,Ia,Ib,Ic,In

### FFT\_DATAi (FFT 数据接口输出数据寄存器 i, i=0~20)

<b>FFT_DATAi (FFT 数据接口输出数据寄存器 i, i=0~20)</b>		<b>基地址: 0x4002 2000</b>						
		<b>偏移地址: 20H+04H*i (i=0~20)</b>						
	<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>	FFT_DATAi[31:24]							
<b>Write:</b>								
<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>	FFT_DATAi[23:16]							
<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>	FFT_DATAi[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>	FFT_DATAi[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
FFT_DATAi[31:0]	FFT数据接口输出数据寄存器i, i=0~20

**NIS\_DATAi (非侵入数据接口输出数据寄存器 i, i=0~6)**

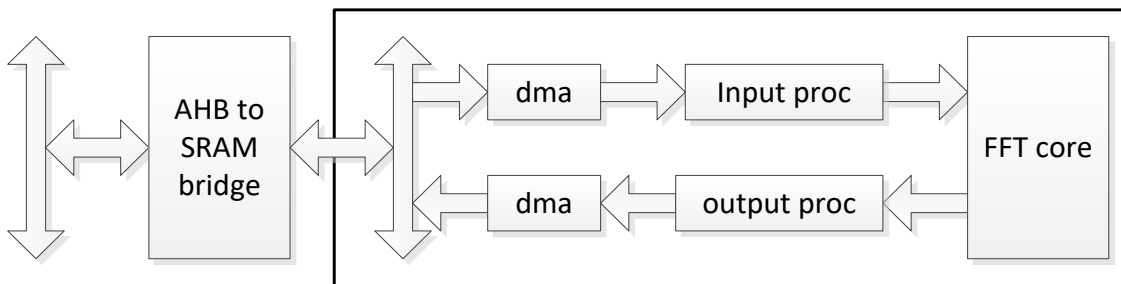
NIS_DATAi (FFT 数据接口输出数据寄存器 i, i=0~6)			基地址: 0x4002 2000 偏移地址: 80H+04H*i (i=0~6)					
	Bit31	30	29	28	27	26	25	Bit24
Read:	NIS_DATAi[31:24]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	NIS_DATAi[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	NIS_DATAi[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	NIS_DATAi[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
NIS_DATAi[31:0] ]	非侵入数据接口输出数据寄存器i, i=0~6

## 20 FFT 模块

### 20.1 概述

FFT 模块支持 64, 128, 256, 512 点的 22 位宽 FFT 计算。FFT 具有 AHB 总线的 master 接口，可以从指定存储空间读取数据计算，并将结果输出至指定存储空间。

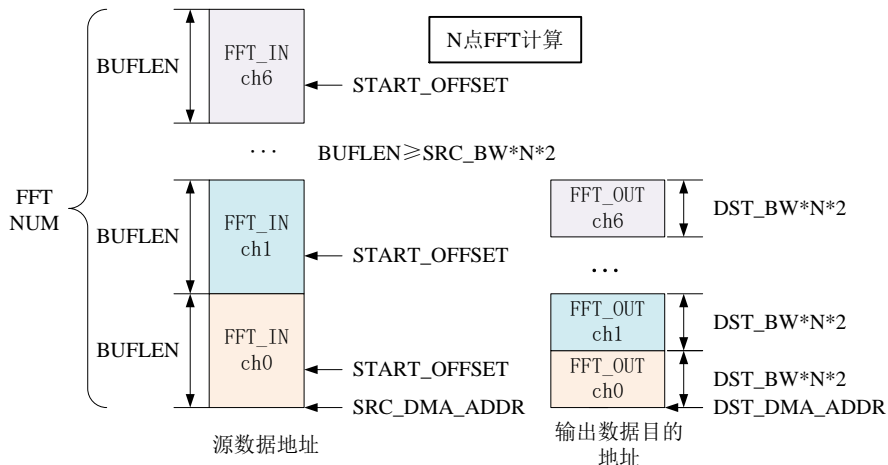


### 20.2 功能描述

- 支持22位宽 FFT 计算
- 支持64, 128, 256, 512点复数输入 FFT 计算
- 支持128, 256, 512, 1024点的实数输入 FFT 计算
- 输入数据接口支持16、24、32位位宽
- 输出数据支持16、32位位宽
- 输入数据支持右移控制
- 输出数据支持自动左移

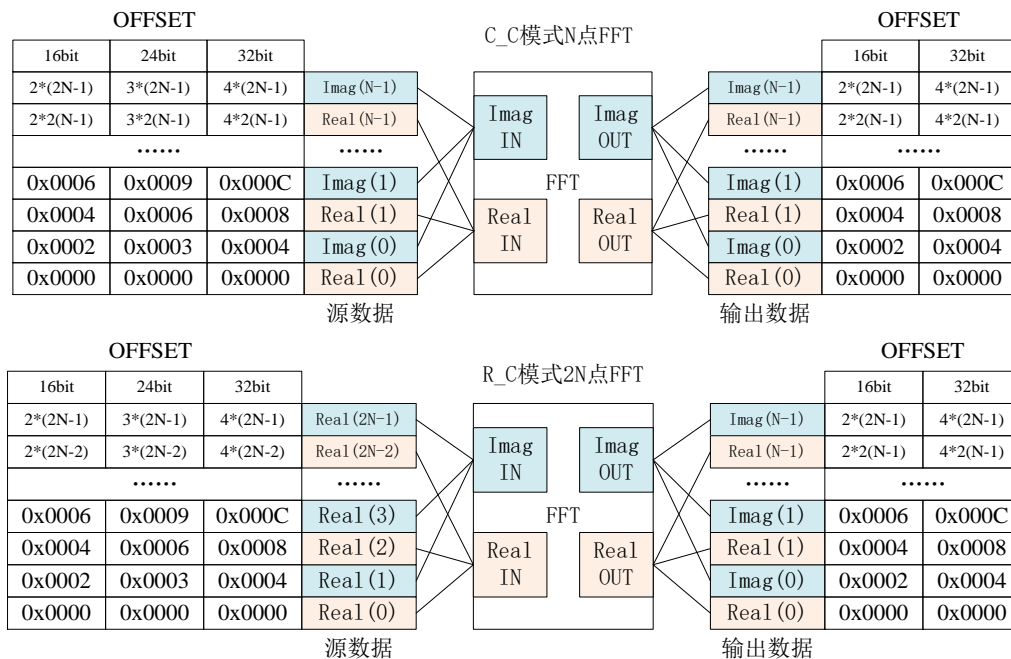
## 20.3 FFT 使用说明

### 20.3.1 FFT 数据配置



FFT 最大支持从源数据地址读取 7 个通道的数据进行计算（NUM\_FFTS[2:0]配置 FFT 计算的次数），源数据基地址（FFT\_SRC\_DMA\_ADDR[31:0]）配置了所有通道数据内存空间的起始地址，起始偏移地址（FFT\_START\_OFFSET[31:0]）配置了所有通道数据计算时的起始地址。

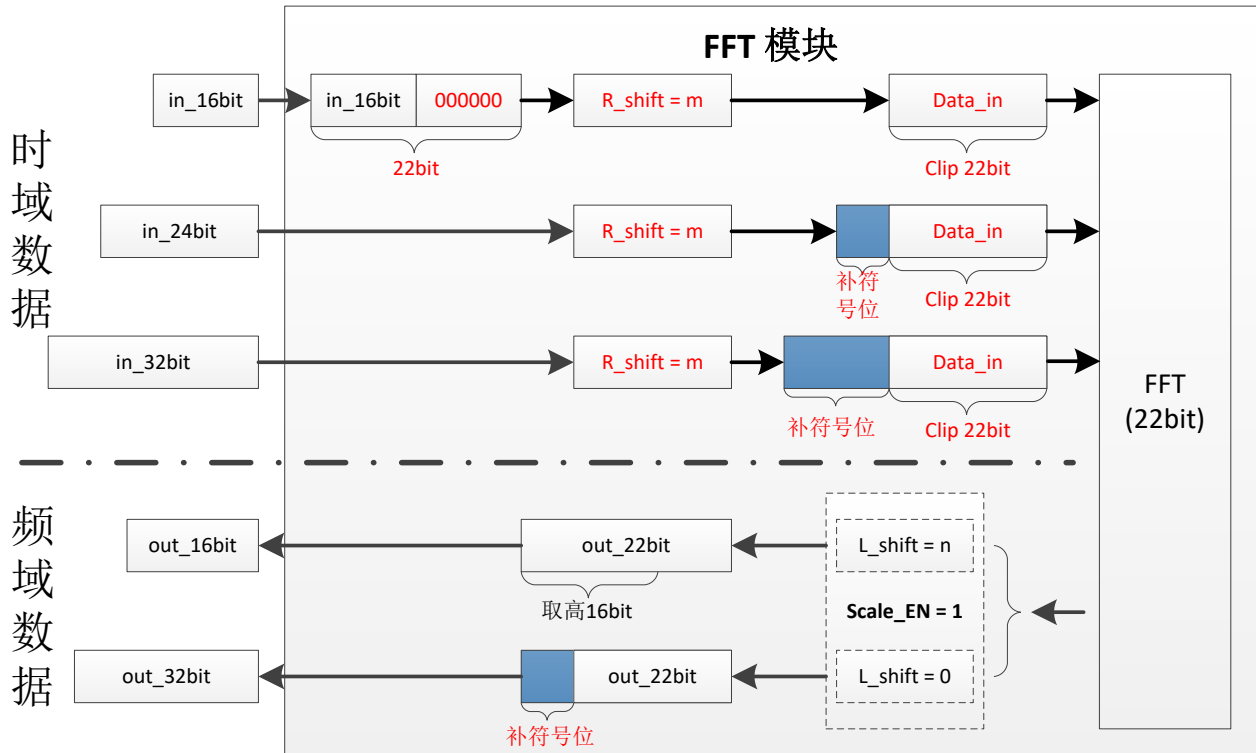
FFT 在计算时，自动按照以上配置依次从各个通道的源数据起始地址读取足够计算的数据（N 点 FFT，N 由寄存器 FFTMODE[1:0]决定）并计算，然后将计算结果输出至所设定的目标存储空间内。输出数据的起始目的地址由寄存器 FFT\_DST\_DMA\_ADDR[31:0]配置决定，每个通道的输出数据长度由输出数据宽度（DST\_BW）和 FFT 计算的点数（N）决定。



如上图所示，FFT 模块支持 2N 点全实数输入的 FFT 计算。使用时注意输入输出的数据缓冲空间大小满足当前所配置的 FFT 计算所需要的大小。

## 20.3.2 FFT 输入输出

在使用时需注意输入数据的位宽（FFT 实际有效位宽 22bit），在 24bit 和 32bit 输入时，通过源地址数据右移控制寄存器进行右移，避免溢出。



当发生溢出错误时，FFT 不会停止，但会产生相应的溢出错误标志（寄存器 FFTERR），当使能了 FFT\_IE=1 时，溢出错误也会触发中断并使 FFT\_IF 置 1，用户可以查询 FFTERR 得到 FFT 计算发生溢出的错误状态。

### 20.3.2.1 FFT 输入

输入位宽为 16bit 数据时，FFT 模块会将 16bit 数据作为 22bit 位宽数据的高 16 位（低位补 0，即左移 6 位），而后再将此 22bit 位宽的数据根据所配置的右移进行处理。输入位宽为 24/32bit 数据时，则 FFT 模块直接根据所配置的右移进行处理。

### 20.3.2.2 FFT 输出

FFT 输出数据仅支持 16bit 和 32bit 两种格式，且在 R\_C（实数输入，复数输出）的  $2N$  点 FFT 模式下，只输出前一半的结果数据（后一半结果数据与前一半对称）。

16bit 位宽输出时，FFT 将 22bit 计算结果的低 6bit 进行四舍五入处理后，再将高 16bit 输出；32bit 位宽输出时，FFT 将 22bit 结果高位补符号位后输出。

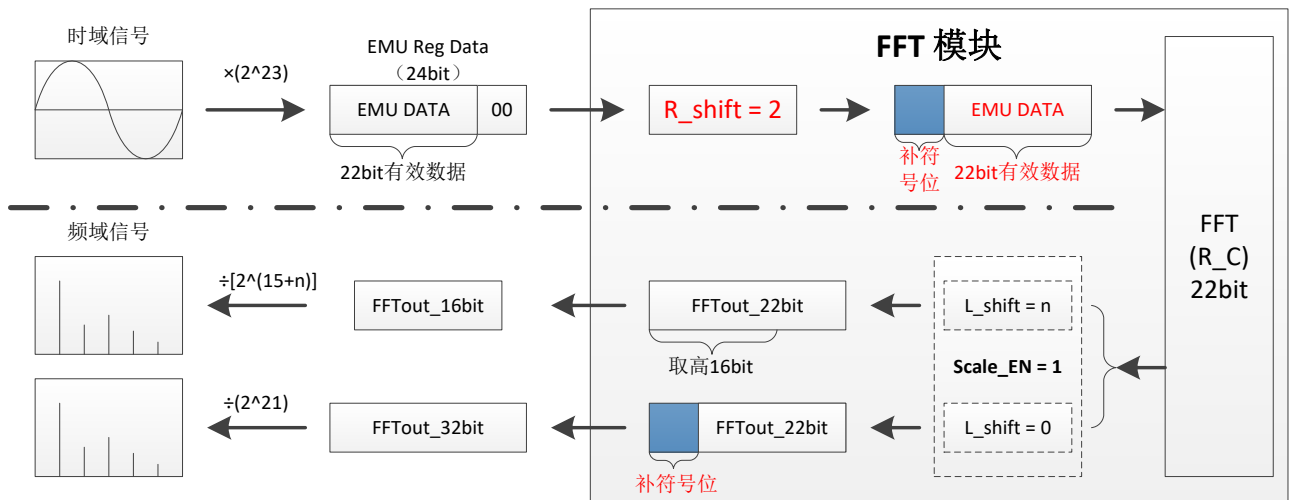
在 16bit 输出模式下，可启用输出自动比例调节功能来提高输出数据的精度。当该功能使能后（FFT\_DST\_DMA\_CTL.SCALE\_EN=1=1），在 FFT 模块将结果数据输出到 RAM 之前，会根据数据的大小进行判断，移除多余的符号位，保留更多的有效数据位后输出，并将左移位数更新至对应通道的左移状态寄存器中。当使用该功能时，需根据所配置的 FFT 计算确定基波所在位置并填入 FFT\_DST\_DMA\_CTL.FUNDIDX

寄存器，FFT 模块会将基波单独进行左移处理，其他谐波统一左移处理。

注：

- 1) 32bit 位宽输出已是全精度输出，故 32bit 位宽输出时自动左移功能无效，但当在 32bit 位宽输出模式下使能了 FFT\_DST\_CTL. SCALE\_EN=1 时，输出自动左移状态寄存器 FFT\_DST\_FUNDSCALE 和 FFT\_DST\_NONFUNDSCALE 仍然会有非 0 值输出，使用时忽略这两个寄存器的值即可。
- 2) 通常 FFT 模块的相应状态寄存器不会自动清除，需要软件操作或在下一次计算时更新。

### 20.3.2.3 输入输出数据使用说明举例



- 1) EMU 寄存器输出为 24bit 高位对齐格式（时域信号放大  $2^{23}$  倍），实际有效位为 22bit 有符号数，在经过 EMU interface 处理后输入至 FFT 模块时，将右移配置为 2。（如上图所示，实际进行 FFT 计算的数据为对应时域信号为放大  $2^{21}$  倍后结果）
- 2) FFT（R\_C 模式）计算出 22bit 位宽的结果，根据输出位宽和是否使能自动比例调节进行左移处理
- 3) 16bit 位宽取 22bit 的高 16 位输出，32bit 位宽在高位补符号位后输出
- 4) 16bit 输出数据除以  $2^{(15+n)}$  后为原始时域信号所对应的频域信号（n 为自动左移位数）；32bit 输出数据除以  $2^{21}$  后为原始时域信号所对应的频域信号

### 20.3.3 FFT 内存访问

FFT 模块内的 DMA 在访问对应数据空间时，具有最高优先级，当 FFT\_DMA，DMA，CPU 三者同时对同一块 RAM 进行访问时，始终由 FFT\_DMA 优先访问，而后 DMA 和 CPU 会交替优先级进行访问。

### 20.3.4 FFT 停止和复位

FFT 模块在运行完一次触发的计算后会自动停止，相关状态（如输入输出的 DMA 偏移地址等）均会保持，下次触发时会根据配置在之前基础上继续进行计算。

将 FFTCFG.ABORT 写 1，可以中止并复位 FFT 模块。ABORT 操作会停止并复位 FFT 模块。

**当新的计算需要重新配置 FFT 时，需首先将 FFTCFG.ABORT 写 1 以停止和复位 FFT 状态机，然后写入新的配置，再启动新的 FFT 计算。**

注：

在硬件触发模式下，使用 ABORT 时，建议软件按顺序配合进行以下操作

- 1) 关闭 DMA12 至 FFT 输入数据空间的通路，或关闭数据源头；
- 2) Reset DMA12；
- 3) ABORT FFT。

停止 FFT 后，若要重新启动 FFT 时，建议软件按顺序进行操作

- 1) Restart FFT；
- 2) Restart DMA12；
- 3) 开启 DMA12 至 FFT 输入数据空间的通路，并启用数据源。

## 20.4 特殊功能寄存器列表

FFT 模块寄存器基地址：0x4001E000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	FFTCFG	R/W	0x0000	FFT配置寄存器
0x04	FFTIE	R/W	0x0000	FFT中断使能寄存器
0x08	FFT_SRC_DMA_CTL	R/W	0x0000	FFT源数据配置寄存器
0x0C	FFT_SRC_DMA_ADDR	R/W	0x00000000	FFT源数据基地址配置寄存器
0x10	FFT_BUFLLEN	R/W	0x00000000	FFT源数据空间大小配置寄存器（每通道）
0x14	FFT_START_OFFSET	R/W	0x00000000	FFT源数据起始偏移量寄存器
0x18	FFT_DST_DMA_CTL	R/W	0x0000	FFT输出数据控制寄存器
0x1C	FFT_DST_FUNDSCALE	R	0x0000	FFT输出基波左移位寄存器
0x20	FFT_DST_NONFUNDSCALE	R	0x0000	FFT输出非基波左移位寄存器
0x24	FFT_DST_DMA_ADDR	R/W	0x0000	FFT输出数据目的地址寄存器
0x28	FFT_NUMS	R	0x0000	FFT剩余计算数量状态寄存器
0x2C	FFTIF	R	0x0000	FFT状态标志寄存器
0x30	FFTERR	R	0x0000	FFT数据溢出状态寄存器
0x34	FFT_DMA_CNT	R	0x00000000	FFT DMA 计数状态寄存器

## 20.5 特殊功能寄存器说明

### FFTCFG (FFT 配置寄存器)

FFTCFG (FFT 配置寄存器)			基地址: 0x4001E000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	SOFT_EN	INV	ABORT
Write:						N		
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	NUM_FFTS[2:0]			NORM_SQRTN	IODTYPE	FFTMODE[1:0]		SOFT_START
Write:					E			
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SOFT_EN	<b>FFT 软件启动使能控制位</b> 0: 不使能软件启动 (default) 1: 使能软件启动 注: 当使能软件启动时, 硬件自动触发功能将被屏蔽, 只能由软件写入 SOFT_START=1 启动 FFT。
INV	<b>FFT 计算控制位</b> 0: FFT (default) 1: IFFT 注: IFFT 不可选择 R_C 模式。
ABORT	<b>FFT 中止控制位</b> 软件写 1 中止 FFT, FFT 内部状态机复位, 硬件自动清 0
NUM_FFTS[2:0]	<b>FFT 计算次数配置位</b> 软件写入, 每次触发 FFT 计算 NUM_FFTS[2:0]次。
NORM_SQRTN	<b>FFT 归一化因子配置位</b> 0: 归一化因子为 1/n (default) 1: 归一化因子为 1/sqrt(n) 注: n 由 FFT 计算点数配置位 FFTMODE[1:0]决定。
IODTYPE	<b>FFT 输入数据类型配置位</b> 0: C_C, 复数输入, 复数输出 (default) 1: R_C, 实数输入, 复数输出 注: R_C 模式不支持 IFFT 计算。
FFTMODE[1:0]	<b>FFT 计算点数配置位</b> C_C 模式: 虚数点 00: 64 点 (default) 01: 128 点

	10: 256 点 11: 512 点 R_C 模式: 实数点 00: 128 点 (default) 01: 256 点 10: 512 点 11: 1024 点
<b>SOFT_START</b>	<b>FFT 软件启动控制位</b> 当 SOFT_EN 功能开启时, 该位写 1 启动 FFT, FFT 计算完成硬件自动清 0。 注: SOFT_EN=0 时, 该位写 1 将会保持 1, 可软件写 0、ABORT 清 0 或硬件触发 FFT 计算完成后清 0。

### FFTIE (FFT 中断使能寄存器)

<b>FFTIE (FFT 中断使能寄存器)</b>		基地址: <b>0x4001E000</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	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	FFT_DO	FFT_IE
<b>Write:</b>							NE_IE	
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
<b>FFT_DONE_IE</b>	<b>FFT 配置通道计算完成中断使能位 (当前所有配置通道)</b> 0: 不使能 FFT 配置通道计算完成中断 1: 使能 FFT 配置通道计算完成中断
<b>FFT_IE</b>	<b>FFT 计算完成中断使能位 (单次)</b> 0: 不使能 FFT 计算完成中断 1: 使能 FFT 计算完成中断

### FFT\_SRC\_DMA\_CTL (FFT 源数据配置寄存器)

<b>FFT_SRC_DMA_CTL (FFT 源数据配置寄存器)</b>		基地址: <b>0x4001E000</b> 偏移地址: <b>08H</b>						
	<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>	X	X	SRC_BW[1:0]		R_SHIFT6[3:0]			

<b>Write:</b>								
<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>	R_SHIFT5[3:0]				R_SHIFT4[3:0]			
<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>	R_SHIFT3[3:0]				R_SHIFT2[3:0]			
<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>	R_SHIFT1[3:0]				R_SHIFT0[3:0]			
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
SRC_BW[1:0]	<b>FFT 输入数据位宽选择位</b> 00: 16bit (default) 01: 24bit 1x: 32bit 注: FFT 有效位宽 22bit, 当选择 16bit 位宽时, 输入数据高位对齐至 FFT 输入端口
R_SHIFT6[3:0]	<b>FFT 源数据通道 6 右移控制位</b> 000: 右移 0 位 (default) 将源数据右移 R_SHIFT6[3:0]位再进行计算
R_SHIFT5[3:0]	<b>FFT 源数据通道 5 右移控制位</b> 000: 右移 0 位 (default) 将源数据右移 R_SHIFT6[3:0]位再进行计算
R_SHIFT4[3:0]	<b>FFT 源数据通道 4 右移控制位</b> 000: 右移 0 位 (default) 将源数据右移 R_SHIFT6[3:0]位再进行计算
R_SHIFT3[3:0]	<b>FFT 源数据通道 3 右移控制位</b> 000: 右移 0 位 (default) 将源数据右移 R_SHIFT6[3:0]位再进行计算
R_SHIFT2[3:0]	<b>FFT 源数据通道 2 右移控制位</b> 000: 右移 0 位 (default) 将源数据右移 R_SHIFT6[3:0]位再进行计算
R_SHIFT1[3:0]	<b>FFT 源数据通道 1 右移控制位</b> 000: 右移 0 位 (default) 将源数据右移 R_SHIFT6[3:0]位再进行计算
R_SHIFT0[3:0]	<b>FFT 源数据通道 0 右移控制位</b> 000: 右移 0 位 (default) 将源数据右移 R_SHIFT6[3:0]位再进行计算

**FFT\_SRC\_DMA\_ADDR (FFT 源数据基地址配置寄存器)**

<b>FFT_SRC_DMA_ADDR (FFT 源数据基地址配置寄存器)</b>		<b>基地址: 0x4001E000</b>						
		<b>偏移地址: 0CH</b>						
	<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>	SRC_BASE_ADDR[31:24]							
<b>Write:</b>								
<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>	SRC_BASE_ADDR[23:16]							
<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>	SRC_BASE_ADDR[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>	SRC_BASE_ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
SRC_BASE_ADDR[31:0]	<b>FFT 源数据基地址配置位</b> FFT 计算源数据基地址, 当使能多通道 FFT 计算时, 该地址为 FFT 通道 0 的起始地址。

**FFT\_BUFLLEN (FFT 每个通道源数据空间大小配置寄存器)**

<b>FFT_BUFLLEN (FFT 每个通道源数据空间大小配置寄存器)</b>		<b>基地址: 0x4001E000</b>						
		<b>偏移地址: 10H</b>						
	<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>	BUFLLEN[31:24]							
<b>Write:</b>								
<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>	BUFLLEN[23:16]							
<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>	BUFLEN[15:8]							
<b>Write:</b>	BUFLEN[15:8]							
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	BUFLEN[7:0]							
<b>Write:</b>	BUFLEN[7:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
BUFLEN[31:0]	FFT 每个通道源数据空间大小配置位 配置每个 FFT 计算通道的 buffer 空间大小, 单位 word

### FFT\_START\_OFFSET (FFT 源数据起始偏移量寄存器)

<b>FFT_START_OFFSET (FFT 源数据起始偏移量寄存器)</b>	基地址: <b>0x4001E000</b> 偏移地址: <b>14H</b>							
	<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>	START_OFFSET[31:24]							
<b>Write:</b>	START_OFFSET[31:24]							
<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>	START_OFFSET[23:16]							
<b>Write:</b>	START_OFFSET[23:16]							
<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>	START_OFFSET[15:8]							
<b>Write:</b>	START_OFFSET[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>	START_OFFSET[7:0]							
<b>Write:</b>	START_OFFSET[7:0]							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
START_OFFSET[31:0]	<b>FFT 源数据地址起始偏移量配置位</b> 配置每个 FFT 计算通道的起始计算偏移地址, 当该配置位不为 0 时, FFT 从 buffer 的该偏移地址开始读取源数据进行计算。 <b>注: 该寄存器只有在 FFT 模块为停止状态时写入有效 (即第一次使用或 ABORT 后), 当从停止状态 (第一次使用或 ABORT 后) 触发 FFT 计算时, 会自动加载寄存器最新写入的 offset 值; 从非停止状态触发 FFT 计算时, 会从上轮计算后的偏移量开始新一轮计算。</b>

读取该寄存器返回当前 FFT 内部的 offset 值，需要注意的是，由于只有当触发 FFT 计算时，硬件才会加载最新的值，因此软件并不一定能读取到实时的值，读到的结果通常为上一次计算后内部的 offset 值。

### FFT\_DST\_DMA\_CTL (FFT 输出数据控制寄存器)

FFT_DST_DMA_CTL (FFT 输出数据控制寄存器)			基地址: 0x4001E000 偏移地址: 18H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	X	X	X	X	X	X	FUNDIDX[9:8]	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	FUNDIDX[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
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:	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SCALE_	DST_B
Write:	X	X	X	X	X	X	EN	W
Reset:	0	0	0	0	0	0	0	0

位	功能描述
FUNDIDX[9:0]	基波所在位置配置位 表示基波在结果中所在位置的偏移量
SCALE_EN	结果自动比例调节使能控制位 0: 不使能 1: 使能 注: 使能后, FFT 会将计算结果自动进行比例调节 (左移放大) 后再输出, 可提高 16bit 输出的精度, 仅 16bit 位宽输出时有效。
DST_BW	FFT 计算结果输出数据位宽配置位 0: 16bit 1: 32bit

**FFT\_DST\_FUNDSCALE (FFT 输出基波左移位寄存器)**

FFT_DST_FUNDSCALE (FFT 输出基波左移位寄存器)		基地址: 0x4001E000 偏移地址: 1CH							
	Bit31	30	29	28	27	26	25	Bit24	
Read:	X	X	X	X	FUNSCALE_SHIFT6[3:0]				
Write:					X				
Reset:	0	0	0	0	0	0	0	0	
	Bit23	22	21	20	19	18	17	Bit16	
Read:	FUNSCALE_SHIFT5[3:0]				FUNSCALE_SHIFT4[3:0]				
Write:	X				X				
Reset:	0	0	0	0	0	0	0	0	
	Bit15	14	13	12	11	10	9	Bit8	
Read:	FUNSCALE_SHIFT3[3:0]				FUNSCALE_SHIFT2[3:0]				
Write:	X				X				
Reset:	0	0	0	0	0	0	0	0	
	Bit7	6	5	4	3	2	1	Bit0	
Read:	FUNSCALE_SHIFT1[3:0]				FUNSCALE_SHIFT0[3:0]				
Write:	X				X				
Reset:	0	0	0	0	0	0	0	0	

位	功能描述
FUNSCALE_S HIFT6[3:0]	<b>FFT 输出数据通道 6 基波左移状态位</b> 000: 左移 0 位 (default) FFT 输出数据左移了 FUNSCALE_SHIFT6[3:0]位。
FUNSCALE_S HIFT5[3:0]	<b>FFT 输出数据通道 5 基波左移状态位</b> 000: 左移 0 位 (default) FFT 输出数据左移了 FUNSCALE_SHIFT5[3:0]位。
FUNSCALE_S HIFT4[3:0]	<b>FFT 输出数据通道 4 基波左移状态位</b> 000: 左移 0 位 (default) FFT 输出数据左移了 FUNSCALE_SHIFT4[3:0]位。
FUNSCALE_S HIFT3[3:0]	<b>FFT 输出数据通道 3 基波左移状态位</b> 000: 左移 0 位 (default) FFT 输出数据左移了 FUNSCALE_SHIFT3[3:0]位。
FUNSCALE_S HIFT2[3:0]	<b>FFT 输出数据通道 2 基波左移状态位</b> 000: 左移 0 位 (default) FFT 输出数据左移了 FUNSCALE_SHIFT2[3:0]位。
FUNSCALE_S HIFT1[3:0]	<b>FFT 输出数据通道 1 基波左移状态位</b> 000: 左移 0 位 (default) FFT 输出数据左移了 FUNSCALE_SHIFT1[3:0]位。
FUNSCALE_S HIFT0[3:0]	<b>FFT 输出数据通道 0 基波左移状态位</b> 000: 左移 0 位 (default)

	FFT 输出数据左移了 FUNSCALE_SHIFT0[3:0]位。
--	------------------------------------

### FFT\_DST\_NONFUNDSCALE (FFT 输出非基波左移位寄存器)

<b>FFT_DST_NONFUNDSCALE</b> (FFT 输出非基波左移位寄存器)		<b>基地址: 0x4001E000</b> <b>偏移地址: 20H</b>							
	<b>Bit31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>	<b>Bit24</b>
<b>Read:</b>	X	X	X	X	NONFUNSCALE_SHIFT6[3:0]				
<b>Write:</b>					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>16</b>	<b>Bit16</b>
<b>Read:</b>	NONFUNSCALE_SHIFT5[3:0]				NONFUNSCALE_SHIFT4[3:0]				
<b>Write:</b>	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>8</b>	<b>Bit8</b>
<b>Read:</b>	NONFUNSCALE_SHIFT3[3:0]				NONFUNSCALE_SHIFT2[3:0]				
<b>Write:</b>	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>0</b>	<b>Bit0</b>
<b>Read:</b>	NONFUNSCALE_SHIFT1[3:0]				NONFUNSCALE_SHIFT0[3:0]				
<b>Write:</b>	X				X				
<b>Reset:</b>	0	0	0	0	0	0	0	0	

位	功能描述
NONFUNSCALE_SHIFT6[3:0]	<b>FFT 输出数据通道 6 非基波左移状态位</b> 000: 左移 0 位 (default) FFT 输出数据左移了 FUNSCALE_SHIFT6[3:0]位。
NONFUNSCALE_SHIFT5[3:0]	<b>FFT 输出数据通道 5 非基波左移状态位</b> 000: 左移 0 位 (default) FFT 输出数据左移了 FUNSCALE_SHIFT5[3:0]位。
NONFUNSCALE_SHIFT4[3:0]	<b>FFT 输出数据通道 4 非基波左移状态位</b> 000: 左移 0 位 (default) FFT 输出数据左移了 FUNSCALE_SHIFT4[3:0]位。
NONFUNSCALE_SHIFT3[3:0]	<b>FFT 输出数据通道 3 非基波左移状态位</b> 000: 左移 0 位 (default) FFT 输出数据左移了 FUNSCALE_SHIFT3[3:0]位。
NONFUNSCALE_SHIFT2[3:0]	<b>FFT 输出数据通道 2 非基波左移状态位</b> 000: 左移 0 位 (default) FFT 输出数据左移了 FUNSCALE_SHIFT2[3:0]位。

NONFUNSCALE_SHIFT1[3:0]	<b>FFT 输出数据通道 1 非基波左移状态位</b> 000: 左移 0 位 (default) FFT 输出数据左移了 FUNSCALE_SHIFT1[3:0]位。
NONFUNSCALE_SHIFT0[3:0]	<b>FFT 输出数据通道 0 非基波左移状态位</b> 000: 左移 0 位 (default) FFT 输出数据左移了 FUNSCALE_SHIFT0[3:0]位。

### FFT\_DST\_DMA\_ADDR (FFT 输出数据目的地址寄存器)

<b>FFT_DST_DMA_ADDR (FFT 输出数据目的地址寄存器)</b>		<b>基地址: 0x4001E000</b>						
		<b>偏移地址: 24H</b>						
	<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>	DST_ADDR[31:24]							
<b>Write:</b>								
<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>	DST_ADDR[23:16]							
<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>	DST_ADDR[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>	DST_ADDR[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DST_ADDR[31:0]	<b>FFT 输出数据目的地址配置位</b> 配置 FFT 计算输出数据的目的地址, 当有多个通道 FFT 进行计算时, 输出结果会按通道顺序依次存放。

### FFT\_NUMS (FFT 剩余计算数量状态寄存器)

<b>FFTNUMS (FFT 剩余计算数量状态寄存器)</b>	<b>基地址: 0x4001E000</b>
	<b>偏移地址: 28H</b>

	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	FFT_NUMS[2:0]		
Write:						X		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
FFT_NUMS[2:0]	<p><b>FFT 剩余计算数量状态位</b></p> <p>软件读取该状态位获得当前剩余未完成 FFT 计算次数值, 硬件在计算时会自动更新此数值。</p> <p>触发 FFT 开始计算时, 自动加载 FFTCFG.NUM_FFTS[2:0], 每计算一次 FFT, FFT_NUMS[2:0]值减 1, 当 NUM_FFTS[2:0]=0 时, FFT 停止。</p>

注: 读取该位可以判断 FFT 模块的运行状态, FFT\_NUMS[2:0]=0 则 FFT 停止, FFT\_NUMS[2:0]不为 0 则 FFT 模块正在运行。

## FFTIF (FFT 中断标志寄存器)

FFTIF (FFT 中断标志寄存器)		基地址: 0x4001E000 偏移地址: 2CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	FFT_DO NE_IF	FFT_IF
Write:							X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
FFT_DONE_IF	<p><b>FFT 配置通道计算完成中断标志位</b></p> <p>0: FFT 配置通道计算未完成</p> <p>1: FFT 配置通道计算完成</p> <p>完成当前配置所有通道的 FFT 后该标志会置起, 写 1 清 0。</p>
FFT_IF	<p><b>FFT 计算完成中断标志位</b></p> <p>0: FFT 计算未完成</p> <p>1: FFT 计算完成</p>

完成一次 FFT 后该标志会置起，写 1 清 0。

## FFTERR (FFT 数据溢出状态寄存器)

FFTERR (FFT 数据溢出状态寄存器)			基地址: 0x4001E000 偏移地址: 30H					
	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	clipout	clipin	Reserved	Reserved	clipe
Write:						X	X	
Reset:	0	0	0	0	0	0	0	0

位	功能描述
clipout	<b>数据输出比例调节溢出状态标志位 (左移)</b> 0: 输出数据在进行比例调节时未溢出 1: 输出数据在进行比例调节时溢出 写 1 清 0
clipin	<b>数据输入比例调节溢出状态标志位 (右移)</b> 0: 输入数据在进行比例调节时未溢出 1: 输入数据在进行比例调节时溢出 写 1 清 0
clipe	<b>FFT 第一级运算溢出状态标志位</b> 0: FFT 第一级运算时未溢出 1: FFT 第一级运算时溢出 写 1 清 0

## FFT\_DMA\_CNT (FFT DMA 计数状态寄存器)

FFT_DMA_CNT (FFT DMA 计数状态寄存器)			基地址: 0x4001E000 偏移地址: 34H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	DMA_OUT_CNT[15:8]							
Write:	X							
Reset:	0	0	0	0	0	0	0	0

	Bit23	22	21	20	19	18	17	Bit16
<b>Read:</b>	DMA_OUT_CNT[7:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
<b>Read:</b>	DMA_IN_CNT[15:8]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	DMA_IN_CNT[7:0]							
<b>Write:</b>	X							
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
DMA_OUT_CNT[15:0]	<b>FFT 输出 DMA 剩余计数状态位 (单位 word)</b> 读取该状态为, 可以得到 FFT 输出 DMA 在传输数据时, 所剩余数据的计数值
DMA_IN_CNT[15:0]	<b>FFT 输入 DMA 剩余计数状态位 (单位 word)</b> 读取该状态为, 可以得到 FFT 输入 DMA 在传输数据时, 所剩余数据的计数值

## 21 非侵入数据帧打包模块 FRAMPACK

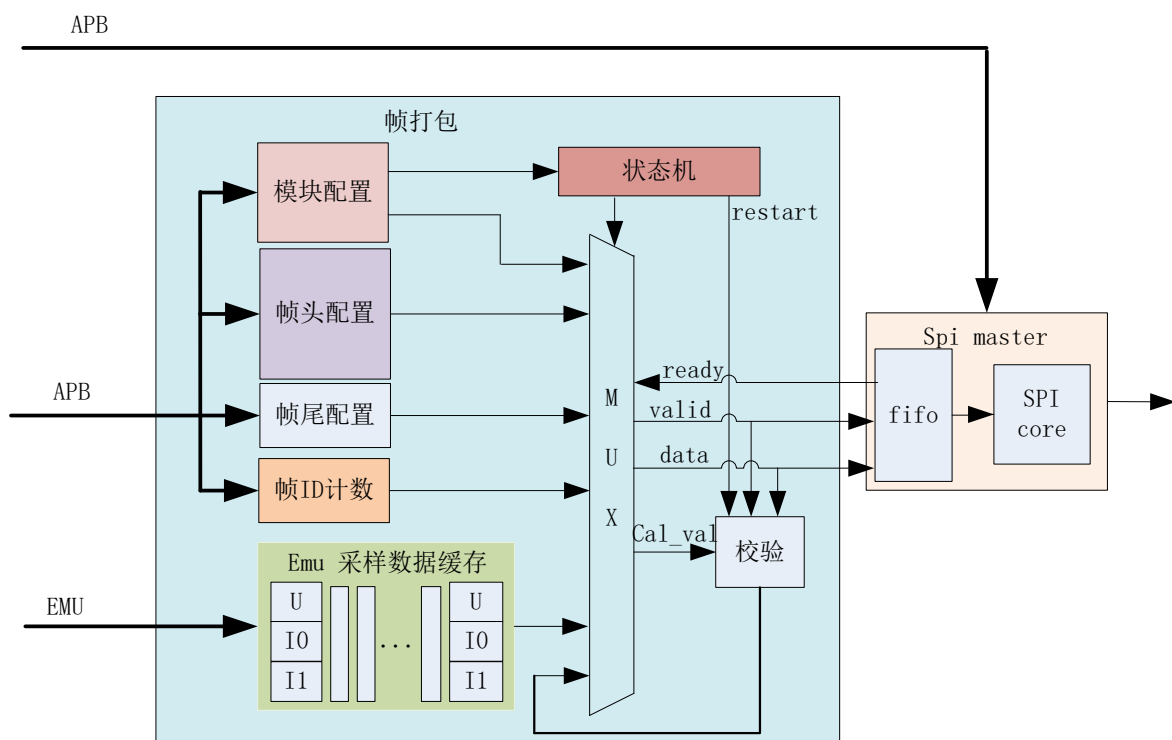
### 21.1 概述

非侵入数据帧打包模块可实现将 EMU 的数据自动读取并组包后通过 SPI0/2 发送，用户可根据需求进行相关功能配置

- 非侵入数据帧打包模块具有独立的使能位
- 帧头信息 HEAD A 和 HEAD B 分别 1~16 字节可配置
- 帧尾信息 1 字节可配置
- 校验方式可选 CRC 或是校验和，并在填充数据时硬件自动计算
- CRC 支持 CRC16/32，校验和 support 1、2、4 字节可配置
- 帧数据的长度 1~1024 个采样组数据可配置
- 帧 ID 支持 1~2 字节可配置，硬件可自动计算并填充
- 帧打包模块可选自动控制 SPI0/SPI2 的 CS
- 每个信号通道独立 32byte 深度 FIFO

#### 21.1.1 非侵入数据帧打包模块

数据帧打包模块主要分成配置模块、帧 ID 计数模块、emu 采样数据缓存模块、校验数据产生模块和状态控制模块。



## 1. 模块配置

- 1) 非侵入数据帧打包模块支持 24bit 和 16bit 位宽的 EMU 数据传输，配置 SDA\_CFG=0，选取 24bit 位宽的 EMU 采样数据，配置 SDA\_CFG=1，选取 16bit 位宽的 EMU 采样数据（高 16 位）。
- 2) EMU 的各通道数据输出顺序可以随意组合配置，通过 COM\_SEL[2:0]可配置输出通道，最大支持 7 路 EMU 数据，通过 SEQSEL 寄存器可以配置各个 EMU 采样数据通道从不同的输出通道序列上输出。
- 3) 不同于打包模块使能的关闭，操作模块中止控制位 ABORT 写 1 可实现模块在发送完当前数据帧后停止功能。

## 2. 帧头配置

- 1) 通过 FTLA\_CFG[3:0]和 FTLB\_CFG[3:0]可以分别配置帧头 HEADA 和 HEADB 的有效字节长度。HEADA 和 HEADB 的帧头的使用顺序，始终从 HA0(或 HB0)开始，依次递增至 HA<sub>x</sub>(或 HB<sub>x</sub>)，x 为 FTLA\_CFG[3:0]（或 FTLB\_CFG[3:0]）配置生效的值。
- 2) 帧头 A 配置的为不参与校验的帧头部分，帧头 B 配置的为参与校验的帧头部分。
- 3) 帧头发送完成后会给出帧头传输完成中断标志（HEADIF=1），此时模块进入帧数据的发送模式，用户可根据此标志写入下一次传输的帧头部分以及配置下一次发送的数据长度 SAMPLE\_NUM[15:0]。

## 3. 帧 ID 计数配置

- 1) 帧 ID 长度可通过配置 ID\_SEL 选择 1byte 或 2byte，并且可以通过 ID\_ADDR[3:0]选择 ID 在帧头 HEADB 中的位置。
- 2) 帧 ID 的清除有两种方式，①帧 ID 计数器累加至溢出自动清 0，②对清除 ID 计数控制寄存器 ID\_CLR 写 1，将在前帧发送完成后，帧 ID 值被清 0。
- 3) 通过读取帧 ID 寄存器 FRAM\_ID[15:0]，可得到当前实时的帧 ID；写该寄存器，在下一帧发送时新写入的帧 ID 生效。

## 4. 帧尾配置

帧尾支持 1byte 长度，写 STOP 寄存器来配置。

当帧尾发送完成后，会给出帧尾传输完成中断标志位（ENDIF=1）

## 5. EMU 采样数据缓存

暂存 EMU 数据，避免在发送帧头和帧尾等无法及时发送采样数据时造成数据丢失。

读取 FRAMSTA 寄存器的 FIFOSTA 状态位，可以查看 EMU 数据缓存 FIFO 状态，FRAMIF 寄存器的 FIFOFULLIF 和 FIFOEMPTYIF 中断标志位可以给出内部 FIFO 的满、空状态，当对应的中断使能打开时，可以触发中断，进入中断服务程序。

## 6. 校验

通过 CRC\_SEL 配置可以选择使用累加和或 CRC；校验位的长度可由 CLEN\_SEL[1:0]配置，支持 1 字节、2 字节、4 字节；当选择 CRC 时，通过 CRC\_MODE[1:0]可选择 CRC16/CRC32。

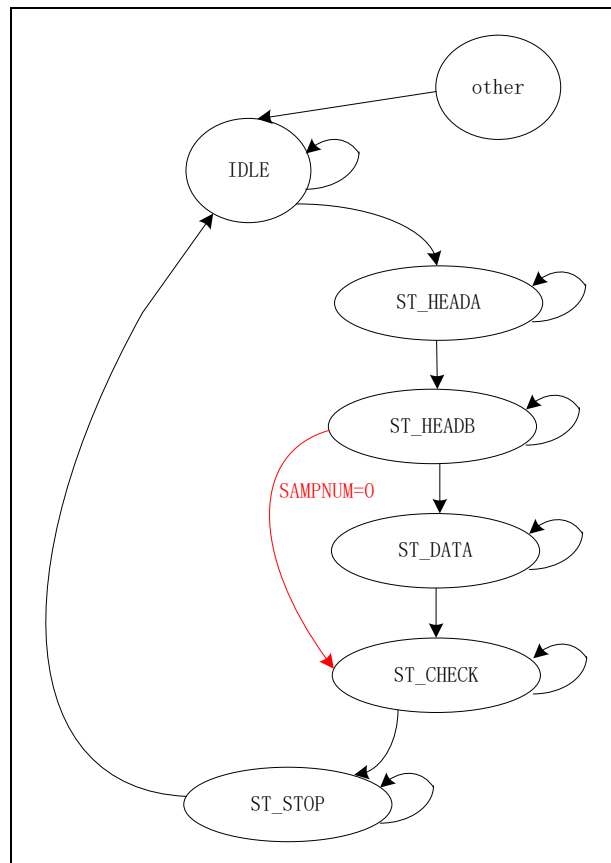
校验由硬件自动完成并发送，在每帧发送完后会自动清 0 并准备好重新计算下一帧。

注：

1) 打包模块在使用时，如需更改配置，建议在当前帧发送完成后，停止、复位模块并重新配置后再使能模块。

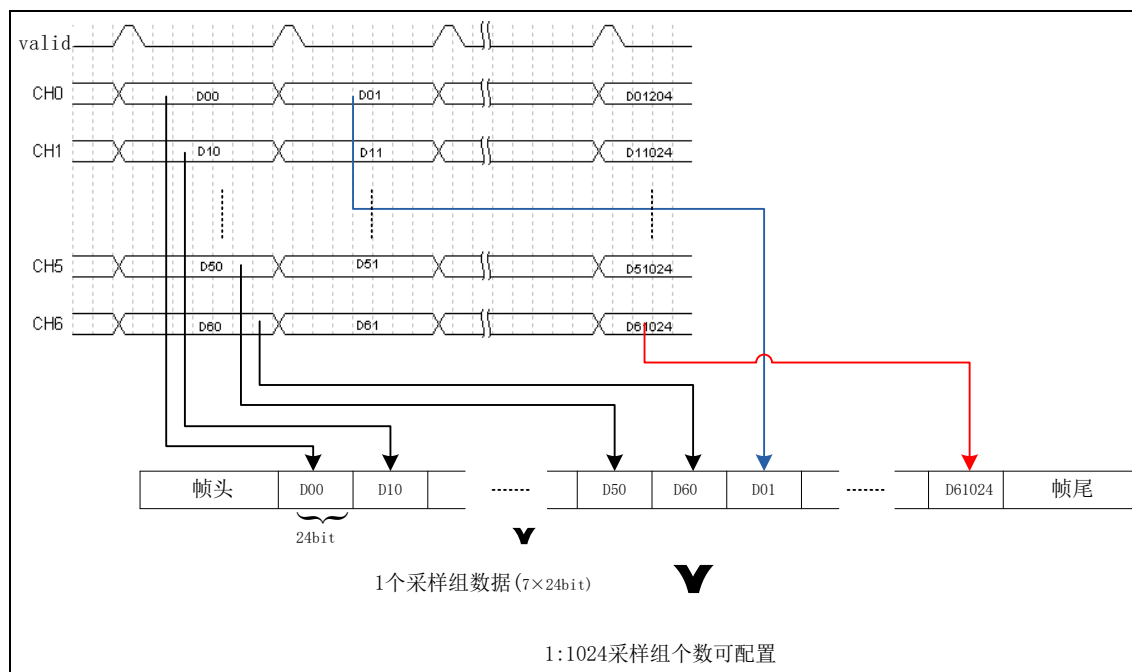
2) 帧打包模块的内部多个参数（帧 ID 计数，CRC 校验结果等）均基于帧发送结束自动清除（状态机回到 IDLE 状态），建议使用过程中软件使用超时判定，当帧打包模块未能在预期时间内给出相应的标志时，可重启模块使其恢复默认状态。

如下图所示，配置 SAMPLE\_NUM[15:0]=0 时，打包模块会跳过 DATA 数据处理状态。



### 21.1.2 帧打包与 SPI 发送

配置帧打包模块并使能后，模块会根据 EMU 接口数据的更新，读取 EMU 采样数据，并按照配置好的帧格式通过 SPI 向外发送。帧数据长度可通过 SAMPLE\_NUM[15:0]配置设定采样组的数据。



### 21.1.3 发送间隔

在帧打包模块配合 SPI2 进行自动组包发送时，可通过 FDLY[15: 0]配置帧与帧之间的空闲间隔，以确保留出足够的时间供从机进行数据分析处理。

## 21.2 特殊功能寄存器列表

FRAMPACK 模块寄存器基地址：0x40013000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	FRAMCON	R/W	0x0100	帧功能配置寄存器
0x04	HLCFG	R/W	0x0000	帧头长度配置寄存器
0x08	FCRCCFG	R/W	0x0000	CRC配置寄存器
0x0C	FCRCINIT	R/W	0x0000FFFF	CRC初始化种子寄存器
0x10	IDCFG	R/W	0x0000	帧ID配置寄存器
0x14	FRAMIE	R/W	0x0000	帧打包中断使能寄存器
0x18	FRAMIF	R/W	0x0008	帧打包中断标志寄存器
0x20	SAMPNUM	R/W	0x0000	EMU数据采样组个数配置寄存器
0x24	HEADAO	R/W	0x00000000	帧头A0配置寄存器
0x28	HEADAO	R/W	0x00000000	帧头A1配置寄存器
0x2C	HEADAO	R/W	0x00000000	帧头A2配置寄存器
0x30	HEADAO	R/W	0x00000000	帧头A3配置寄存器
0x34	HEADB0	R/W	0x00000000	帧头B0配置寄存器
0x38	HEADB1	R/W	0x00000000	帧头B1配置寄存器
0x3C	HEADB2	R/W	0x00000000	帧头B2配置寄存器
0x40	HEADB3	R/W	0x00000000	帧头B3配置寄存器
0x44	STOPF	R/W	0x0000	帧尾配置寄存器
0x48	FRAMIDR	R/W	0x0000	帧ID计数寄存器
0x4C	FRAMSTA	R	0x0000	帧状态寄存器
0x50	FDLY	R/W	0x0000	帧间隔延时配置寄存器
0x54	FAUTONUM	R/W	0x0000	帧自动发送数配置寄存器

## 21.3 特殊功能寄存器说明

### FRAMCON（帧功能配置寄存器）

FRAMCON (帧功能配置寄存器)			基地址: 0x40013000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	DATAOUT_DLY[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	SDA_CF	COM_SEL[2:0]			ENDSEL	SPI_SEL	ABORT	PACKEN
Write:	G							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DATAOUT_DLY[1:0]	<b>输出数据延时控制位</b> 00: 输出数据不延时 01: 输出数据延时 1 个 clk (default) 10: 输出数据延时 2 个 clk 11: 输出数据延时 3 个 clk
SDA_CFG	<b>传输通道数据位宽配置位</b> 0: 输出 EMU 的 24bit 数据 (default) 1: 输出 EMU 的高 16bit 数据
COM_SEL[2:0]	<b>数据传输通道数配置位</b> 000: 设定输出通道支持 6 路 (default) 001: 设定输出通道支持 1 路 010: 设定输出通道支持 2 路 011: 设定输出通道支持 3 路 100: 设定输出通道支持 4 路 101: 设定输出通道支持 5 路 110: 设定输出通道支持 6 路 111: 设定输出通道支持 7 路
ENDSEL	<b>EMU 传输数据大小端选择控制位</b> 0: 小端模式 (default) 1: 大端模式
SPI_SEL	<b>打包模块数据发送选择:</b> 0: 数据往 spi0 传送 1: 数据往 spi2 传送
ABORT	<b>模块中止控制位:</b> 写 1 中止模块, 发送完当前帧后停止模块, 寄存器配置不变, 内部状态机回到 IDLE。

	写 0 模块继续运行。 <b>注：模块 ABORT 后保持停止状态，不再接收数据，也不会向外发送数据，但内部 FIFO 不会被清除。</b>
PACKEN	打包模块使能： 0：不使能（default） 1：使能 <b>注：打包模块使能关闭后，会清除内部状态及 FIFO，但不会清除寄存器配置。</b>

## HLCFG（帧头长度配置寄存器）

HLCFG (帧头长度配置寄存器)			基地址： 0x40013000 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:	FTLA_CFG[3:0]				FTLB_CFG[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
FTLA_CFG[3:0]	<b>帧头 A 长度配置寄存器 (Fram Top A Legnthconfig)</b> 最小为 1，只发送固定长度帧头，由此配置长度,不参与校验计算 0000: 长度为 1 (default) 0001: 长度为 2 ... 1111: 长度为 16 生效长度为 (FTLA_CFG[3:0]+1) 字节，起始字节为 HA0[7:0]，
FTLB_CFG[3:0]	<b>帧头 B 长度配置寄存器 (Fram Top B Legnthconfig)</b> 最小为 1，只发送固定长度帧头，由此配置长度，参与校验计算 0000: 长度为 1 0001: 长度为 2 ... 0100: 长度为 5,... 1111: 长度为 16 生效长度为 (FTLB_CFG[3:0]+1) 字节，起始字节为 HB0[7:0]

注：1) 两部分帧头的有效数据及组包顺序，始终从 HA0（或 HB0）开始，依次递增至 HA<sub>x</sub>（或 HB<sub>x</sub>），x 为 FTLA\_CFG[3:0]（或 FTLB\_CFG[3:0]）配置生效的值。

2) 帧头 A 配置的为不参与校验的帧头部分，帧头 B 配置的为参与校验的帧头部分

**FCRCCFG (CRC 配置寄存器)**

FCRCCFG (CRC 配置寄存器)			基地址: 0x40013000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:								CHECK_ENDSEL
Write:	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:	XorOut	RefOut	RefIn	CLEN_SEL[1:0]		CRC_MODE[1:0]		CRC_SEL
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述												
CHECK_ENDSEL	校验数据输出大小端控制位 0: 小端输出 (default) 1: 大端输出												
XorOut	输出数据按位异或控制 0: 输出数据不操作 (default) 1: 输出数据按位取反操作												
RefOut	输出数据顺序颠倒控制 0: 输出数据顺序不颠倒 (default) 1: 输出数据顺序颠倒, 按所选 CRC 模式数据宽度按位颠倒												
RefIn	输入数据顺序颠倒控制 0: 输入数据顺序不颠倒 (default) 1: 输入数据每个字节按位颠倒												
CLEN_SEL[1:0]	CRC/CHECKSUM 位长度配置位 00: 4byte (default) 01: 1byte 10: 2byte 11: 4byte <b>注: 选择 CRC 校验时, 此长度配置依然生效。</b>												
CRC_MODE[1:0]	CRC 校验计算多项式配置位 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="2">MODE[1:0]</th> <th>生成多项式 (简写)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>CRC16: Poly=0x1021 (default)</td> </tr> <tr> <td>0</td> <td>1</td> <td>CRC16: Poly=0x8005</td> </tr> <tr> <td>1</td> <td>X</td> <td>CRC32: Poly=0x04C11DB7</td> </tr> </tbody> </table> <b>注: 只有在 CRC_SEL 选择 CRC 校验时才生效</b>	MODE[1:0]		生成多项式 (简写)	0	0	CRC16: Poly=0x1021 (default)	0	1	CRC16: Poly=0x8005	1	X	CRC32: Poly=0x04C11DB7
MODE[1:0]		生成多项式 (简写)											
0	0	CRC16: Poly=0x1021 (default)											
0	1	CRC16: Poly=0x8005											
1	X	CRC32: Poly=0x04C11DB7											
CRC_SEL	CRC/CHECKSUM 校验选择配置位 0: 选择累加和 (default) 1: 选择 CRC 校验值												

**FCRCINIT (CRC 初始化种子寄存器)**

<b>FCRCINIT</b> (CRC 初始化种子寄存器)		基地址: <b>0x40013000</b> 偏移地址: <b>0CH</b>						
	<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>	FCRCINIT[31:24]							
<b>Write:</b>								
<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>	FCRCINIT[23:16]							
<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>	FCRCINIT[15:8]							
<b>Write:</b>								
<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>	FCRCINIT[7:0]							
<b>Write:</b>								
<b>Reset:</b>	1	1	1	1	1	1	1	1

位	功能描述
FCRCINIT[31:0]	CRC 初始化数据寄存器 若模式选择 CRC-16, 则低 16 位 (FCRCINIT[15:0]) 有效, 高 16 位无效 若模式选择 CRC-32, 则 32 位 (FCRCINIT[31:0]) 有效

**IDCFG (帧 ID 配置寄存器)**

<b>IDCFG</b> (帧 ID 配置寄存器)		基地址: <b>0x40013000</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>	ID_ADDR[3:0]				X	X	ID_SEL	ID_CLR
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
---	------

ID_ADDR[3:0]	<b>帧 ID 地址配置寄存器 (Fram Top Legnthconfig)</b> 配置此寄存器，确定帧 ID 在帧头中的位置：		
	ID_ADDR[3:0]	ID_SEL = 0	ID_SEL = 1
	0000	ID 位置在 HB0[7:0]	ID[15:8]位置在 HB1[7:0] ID[7:0]位置在 HB0[7:0]
	0001	ID 位置在 HB1[7:0]	ID[15:8]位置在 HB3[7:0] ID[7:0]位置在 HB2[7:0]
	0010	ID 位置在 HB2[7:0]	ID[15:8]位置在 HB5[7:0] ID[7:0]位置在 HB4[7:0]
	0011	ID 位置在 HB3[7:0]	ID[15:8]位置在 HB7[7:0] ID[7:0]位置在 HB6[7:0]
	0100	ID 位置在 HB4[7:0]	ID[15:8]位置在 HB9[7:0] ID[7:0]位置在 HB8[7:0]
	0101	ID 位置在 HB5[7:0]	ID[15:8]位置在 HB11[7:0] ID[7:0]位置在 HB10[7:0]
	0110	ID 位置在 HB6[7:0]	ID[15:8]位置在 HB13[7:0] ID[7:0]位置在 HB12[7:0]
	0111	ID 位置在 HB7[7:0]	ID[15:8]位置在 HB15[7:0] ID[7:0]位置在 HB14[7:0]
	1000	ID 位置在 HB8[7:0]	
	1001	ID 位置在 HB9[7:0]	
	1010	ID 位置在 HB10[7:0]	
	1011	ID 位置在 HB11[7:0]	
	1100	ID 位置在 HB12[7:0]	
	1101	ID 位置在 HB13[7:0]	
	1110	ID 位置在 HB14[7:0]	
1111	ID 位置在 HB15[7:0]		
注：当对应位置被配置为帧 ID 时，对应位的寄存器数据将不再有效，实际模块加载的帧 ID 由 FRAMIDR 寄存器决定。			
ID_SEL	帧 ID 长度选择 0: 1byte 1: 2byte		
ID_CLR	清除 ID 计数控制位 写 1 清除 ID 计数，完成后硬件自动清 0 注：写 1 生效，在当前帧发送完成后，帧 ID 值被清 0，且硬件自动将 ID_CLR 位清 0。当帧 ID 值计满 255（或 65535）后，也会自动清 0。		

## FRAMIE（帧打包中断使能寄存器）

<b>FRAMIE</b> （帧打包中断使能寄存器）		基地址： 0x40013000 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:	X	X	X	FRAM_D	FIFOEMP	FIFOFU	HEADI	ENDIE
Write:				ONEIE	TYIE	LLIE	E	

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

位	功能描述
FRAM_DONEI E	帧自动发送完成中断使能位 0: 不使能 (default) 1: 使能 注: 只有当 FAUTONUM.AUTONUM[15:0]≠0 时, 自动发送完成中断及标志才能生效
FIFOEMPTYIE	EMU 数据缓存 FIFO 空中断使能位 0: 不使能 (default) 1: 使能
FIFOFULLIE	EMU 数据缓存 FIFO 满中断使能位 0: 不使能 (default) 1: 使能
HEADIE	帧头传输完成中断使能位 0: 不使能 (default) 1: 使能
ENDIE	帧尾传输完成中断使能位 0: 不使能 (default) 1: 使能

## FRAMIF (帧打包中断标志寄存器)

FRAMIF (帧打包中断标志寄存器)		基地址: 0x40013000 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	FRAM_D ONEIF	FIFOEMP TYIF	FIFOFU LLIF	HEADIF	ENDIF
Write:								
Reset:	0	0	0	0	1	0	0	0

位	功能描述
FRAM_DONEI F	帧自动发送完成中断标志位(写 0 清 0) 0: 自动发送帧数据未完成 (default) 1: 自动发送帧数据未完成 注: 只有当 FAUTONUM.AUTONUM[15:0]≠0 时, 自动发送完成中断及标志才能生效

FIFOEMPTYIF	<b>EMU 数据缓存 FIFO 空中断标志位</b> 0: EMU 数据缓存 FIFO 非空 1: EMU 数据缓存 FIFO 空
FIFOFULLIF	<b>EMU 数据缓存 FIFO 满中断标志位</b> 0: EMU 数据缓存 FIFO 非满 1: EMU 数据缓存 FIFO 满
HEADIF	<b>帧头传输完成中断标志位(写 0 清 0)</b> 0: 帧头未传输完成 1: 帧头传输完成 注: 使能 HEADIE, 帧头部分发送至 SPI 后会产生相应的中断标志
ENDIF	<b>帧尾传输完成中断标志位(写 0 清 0)</b> 0: 帧尾未传输完成 1: 帧尾传输完成 注: 使能 ENDIE, 帧尾发送至 SPI 后会产生相应的中断标志

## SAMPNUM (EMU 采样组数量配置寄存器)

<b>SAMPNUM</b> (EMU 采样组数量配置寄存器)			基地址: <b>0x40013000</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>	SAMPLE_NUM[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>	SAMPLE_NUM[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
SAMPLE_NUM[15:0]	一帧数据域中, 有多少个 EMU 输入数据采样组个数。

$$len_{fram} = len_{head} + len_{checksum} + len_{end} + chn * n * sample_{num}$$

$len_{fram}$ : 帧的总长度

$len_{head}$ : 帧头长度

$len_{checksum}$ : 检验和长度, 由 CLEN\_SEL[1:0]配置

**len<sub>end</sub>**: 结束符长度, 1 字节

**chn**: 启用传输的 EMU 通道数, 由 COM\_SEL[2:0]配置

**n**: 1 个 EMU 有效数据长度, 由 SDA\_CFG 配置

**sample<sub>num</sub>**: 一帧数据域中, 打包多少个 EMU 数据采样组的数据, 由 SAMPLE\_NUM[15:0]配置

## HEAD A0 (帧头 A0 配置寄存器)

HEAD A0 (帧头 A0 配置寄存器)		基地址: 0x40013000 偏移地址: 24H						
	Bit31	30	29	28	27	26	25	Bit24
Read:	HA3[7:0]							
Write:	HA3[7:0]							
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	HA2[7:0]							
Write:	HA2[7:0]							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	HA1[7:0]							
Write:	HA1[7:0]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	HA0[7:0]							
Write:	HA0[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
HA3[7:0]	帧头 A3 数据寄存器, 存放 1byte 数据
HA2[7:0]	帧头 A2 数据寄存器, 存放 1byte 数据
HA1[7:0]	帧头 A1 数据寄存器, 存放 1byte 数据
HA0[7:0]	帧头 A0 数据寄存器, 存放 1byte 数据

注: 帧头 HEAD A 数据寄存器使用顺序为 HA0, HA1, ..., HA<sub>x</sub>, x 为 x 为 FTLA\_CFG[3:0]配置生效的值。

## HEAD A1 (帧头 A1 配置寄存器)

HEAD A1 (帧头 A1 配置寄存器)			基地址: 0x40013000 偏移地址: 28H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	HA7[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	HA6[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	HA5[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	HA4[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
HA7[7:0]	帧头 A7 数据寄存器, 存放 1byte 数据
HA6[7:0]	帧头 A6 数据寄存器, 存放 1byte 数据
HA5[7:0]	帧头 A5 数据寄存器, 存放 1byte 数据
HA4[7:0]	帧头 A4 数据寄存器, 存放 1byte 数据

注: 帧头 HEAD A 数据寄存器使用顺序为 HA0, HA1, ..., HA<sub>x</sub>, x 为 x 为 FTLA\_CFG[3:0]配置生效的值。

## HEAD A2 (帧头 A2 配置寄存器)

HEAD A2 (帧头 A2 配置寄存器)			基地址: 0x40013000 偏移地址: 2CH					
	Bit31	30	29	28	27	26	25	Bit24
Read:	HA11[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	HA10[7:0]							
Write:								
Reset:	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>	HA9[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>	HA8[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HA11[7:0]	帧头 A11 数据寄存器, 存放 1byte 数据
HA10[7:0]	帧头 A10 数据寄存器, 存放 1byte 数据
HA9[7:0]	帧头 A9 数据寄存器, 存放 1byte 数据
HA8[7:0]	帧头 A8 数据寄存器, 存放 1byte 数据

注: 帧头 HEADA 数据寄存器使用顺序为 HA0, HA1, ..., HA<sub>x</sub>, x 为 x 为 FTLA\_CFG[3:0]配置生效的值。

### HEADA3 (帧头 A3 配置寄存器)

<b>HEADA3</b> (帧头 A3 配置寄存器)			<b>基地址: 0x40013000</b> <b>偏移地址: 30H</b>					
	<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>	HA15[7:0]							
<b>Write:</b>								
<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>	HA14[7:0]							
<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>	HA13[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>	HA12[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HA15[7:0]	帧头 A15 数据寄存器, 存放 1byte 数据
HA14[7:0]	帧头 A14 数据寄存器, 存放 1byte 数据
HA13[7:0]	帧头 A13 数据寄存器, 存放 1byte 数据

HA12[7:0]	帧头 A12 数据寄存器, 存放 1byte 数据
-----------	---------------------------

注: 帧头 HEAD A 数据寄存器使用顺序为 HA0, HA1, ..., HA<sub>x</sub>, x 为 x 为 FTLA\_CFG[3:0]配置生效的值。

## HEADB0 (帧头 B0 配置寄存器)

<b>HEADB0</b> (帧头 B0 配置寄存器)		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 34H</b>						
	<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>	HB3[7:0]							
<b>Write:</b>								
<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>	HB2[7:0]							
<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>	HB1[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>	HB0[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HB3[7:0]	帧头 B3 数据寄存器, 存放 1byte 数据
HB2[7:0]	帧头 B2 数据寄存器, 存放 1byte 数据
HB1[7:0]	帧头 B1 数据寄存器, 存放 1byte 数据
HB0[7:0]	帧头 B0 数据寄存器, 存放 1byte 数据

注: 帧头 HEAD B 数据寄存器使用顺序为 HB0, HB1, ..., HB<sub>x</sub>, x 为 x 为 FTLB\_CFG[3:0]配置生效的值。

## HEADB1 (帧头 B1 配置寄存器)

<b>HEADB1</b> (帧头 B1 配置寄存器)		<b>基地址: 0x40013000</b>						
		<b>偏移地址: 38H</b>						
	<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>	HB7[7:0]							
<b>Write:</b>								
<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>	HB6[7:0]							
<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>	HB5[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>	HB4[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HB7[7:0]	帧头 B7 数据寄存器, 存放 1byte 数据
HB6[7:0]	帧头 B6 数据寄存器, 存放 1byte 数据
HB5[7:0]	帧头 B5 数据寄存器, 存放 1byte 数据
HB4[7:0]	帧头 B4 数据寄存器, 存放 1byte 数据

注: 帧头 HEADB 数据寄存器使用顺序为 HB0, HB1, ..., HBx, x 为 x 为 FTLB\_CFG[3:0]配置生效的值。

## HEADB2 (帧头 B2 配置寄存器)

<b>HEADB2</b> (帧头 B2 配置寄存器)		基地址: <b>0x40013000</b> 偏移地址: <b>3CH</b>						
	<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>	HB11[7:0]							
<b>Write:</b>								
<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>	HB10[7:0]							
<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>	HB9[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>	HB8[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
HB11[7:0]	帧头 B11 数据寄存器, 存放 1byte 数据
HB10[7:0]	帧头 B10 数据寄存器, 存放 1byte 数据
HB9[7:0]	帧头 B9 数据寄存器, 存放 1byte 数据
HB8[7:0]	帧头 B8 数据寄存器, 存放 1byte 数据

注: 帧头 HEADB 数据寄存器使用顺序为 HB0, HB1, ..., HBx, x 为 x 为 FTLB\_CFG[3:0]配置生效的值。

## HEADB3 (帧头 B3 配置寄存器)

HEADB3 (帧头 B3 配置寄存器)			基地址: 0x40013000 偏移地址: 40H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	HB15[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	HB14[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	HB13[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	HB12[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
HB15[7:0]	帧头 B15 数据寄存器, 存放 1byte 数据
HB14[7:0]	帧头 B14 数据寄存器, 存放 1byte 数据
HB13[7:0]	帧头 B13 数据寄存器, 存放 1byte 数据
HB12[7:0]	帧头 B12 数据寄存器, 存放 1byte 数据

注: 帧头 HEADB 数据寄存器使用顺序为 HB0, HB1, ..., HBx, x 为 x 为 FTLB\_CFG[3:0]配置生效的值。

## STOPF (帧尾配置寄存器)

STOPF (帧尾配置寄存器)		基地址: 0x40013000 偏移地址: 44H					
--------------------	--	------------------------------	--	--	--	--	--

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

位	功能描述
STOP[7:0]	结束符寄存器，存放 1byte 数据

## FRAMIDR (帧 ID 寄存器)

<b>FRAMIDR</b> (帧 ID 寄存器)	基地址: 0x40013000 偏移地址: 48H							
	<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>	FRAM_ID[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>	FRAM_ID[7:0]							
<b>Write:</b>								
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
FRAM_ID[15:0]	帧 ID 寄存器 写该寄存器，在下一帧发送时新写入的帧 ID 生效，读取该寄存器，返回当前实时的帧 ID。

注：寄存器 FRAMIDR 发生写入动作时值才会生效，后续如果寄存器 FRAMIDR 没有新的写入动作，帧 ID 在内部会自动加 1。

## FRAMSTA (帧打包状态寄存器)

<b>FRAMSTA</b> (帧打包状态寄存器)	基地址: 0x40013000 偏移地址: 4CH							
	<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	STATE[2:0]		
<b>Write:</b>						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	FFIFOSTA[1:0]		
Write:	X								
Reset:	0	0	0	0	0	0	0	0	

位	功能描述
STATE[2:0]	内部工作状态标志位（对应时序图和状态机查看状态） IDLE :0x0 ST_HEADA :0x1 ST_HEADB :0x2 ST_DATA :0x3 ST_CHECK :0x4 ST_STOP :0x5 DEFAULT :other
FFIFOSTA[1:0]	EMU 数据缓存 FIFO 状态寄存器 只读，FIFO 内部数据深度 0~3 注：FFIFOSTA[1:0]表示当前 FIFO 内部缓存了 0~3 组 EMU 数据。

### FDLY（帧间隔配置寄存器）

<b>FDLY</b> (帧间隔配置寄存器)		基地址: <b>0x40013000</b> 偏移地址: <b>50H</b>						
	Bit15	14	13	12	11	10	9	Bit8
Read:	FDLY[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	FDLY[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
FDLY[15:0]	帧间隔延时配置位 帧打包发送时帧与帧之间的延时控制，延时 FDLY[15:0]个 Fsys 时钟 注：帧间延时期间会自动拉高 CS，下一帧开始前自动拉低 CS

### FAUTONUM（帧自动发送数量配置寄存器）

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

(帧自动发送数量配置寄存器)		偏移地址: 54H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	AUTONUM[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	AUTONUM[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
AUTONUM[15:0]	<p><b>帧自动发送数量配置位</b></p> <p>帧打包模块自动连续发送 AUTONUM[15:0]+1 个数据帧后，停止模块，FRAM_DONEIF 标志置起，若使能了 FRAM_DONEIE，则会产生完成中断。</p> <p>注：寄存器由软件写入配置帧的总个数，读取该寄存器的获得当前已发送的帧个数。仅当 AUTONUM[15:0]≠0 时，自动发送生效。</p>

## 22 EMU 三相计量单元

### 22.1 概述

#### ADC 和计量参数

- 7路  $\Sigma$ - $\delta$ ADC，内置 PGA 可选。其中第七路 ADC 最大可到 32 倍
- 8000:1 动态范围内非线性 < 0.1%，12.8k/25.6k 采样率可选
- 内置 1.2V 基准电压，温度系数在 -40°C ~ 80°C 温度范围，典型值 5ppm
- 提供 A/B/C/合相的全波/基波/谐波有功、无功、视在功率寄存器
- 提供全波/基波 RMS、PQS 两种视在功率、能量计量
- 提供 7 路通道的全波电压电流、基波电压电流、谐波电压电流有效值
- 提供电压、电流矢量和有效值，真线电压有效值寄存器
- 提供 A/B/C 分相和合相电压频率，全波、基波功率因数
- 提供 7 路 ADC 的电压/电流角度寄存器，全波/基波可选
- 提供 7 路 ADC 通道采样的瞬时采样数据、直流数据
- 支持各种状态指示和相应中断
- 支持三相全失压检测

#### 电能部分

- 芯片共 24 路电能计量通路。每路均提供正、反向电能计量通道。即共 48 个电能寄存器
- 24 路电能均支持自定义常数功率输入
- 芯片提供 5 个脉冲输出口，可任选输出 6 路合相电能的只正/只反/正反向的电能脉冲输出
- 多种灵活的起动潜动方式及阈值可选，有效值/功率可选，源头可选，基波谐波可选等
- 提供有功、无功、视在快速脉冲计数
- 提供二级电能脉冲机制，可使电能寄存器精确到 1/N CF 脉冲
- 谐波电能具备独立的 EC 常数
- 支持正负总谐波电能计量，谐波电能计算由硬件加速引擎完成，可实现不间断连续计算
- 支持正反**电能频繁切换**

#### 校表功能

- 提供 7 路 ADC 通道的移采样点相位校正（可达 10 度），相位分段校正
- 提供各通道 Gain 与通道 AllGain 增益
- 全波基波谐波各自独立的 GP/Gphs 与功率 Offset 校表参数
- 提供自动温度补偿功能

#### 电能质量（纯硬件支持）

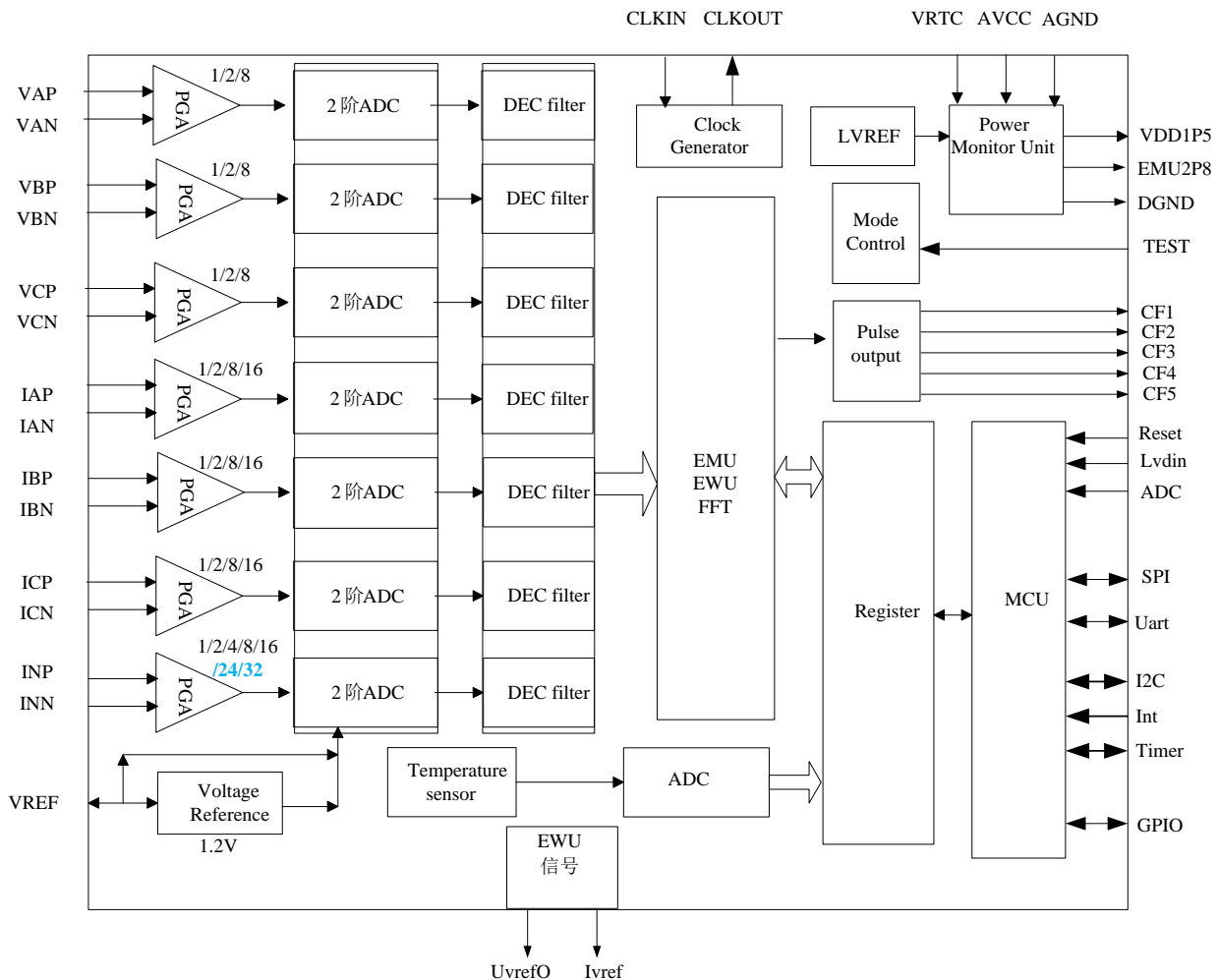
- 基于锁相环的工频同步系统
- 提供电压骤升、电压暂降、电压中断 Interrupt 事件的相关功能配置
- 提供电流过流事件检测的相关功能配置

- IEC 定义的电压真闪变算法计算模块
- IEC 定义的真谐波/间谐波算法支持
- 提供灵活可配的 ADC 数据 Channel Remap，专用 DMA 与硬件 FFT、PACK 功能
- 提供连续 Goertzel 基波复数输出，方便电压/电流不平衡计算
- 提供 7 路 ADC 通道的半波峰值、半波有效值等参数
- IEEE1459 视在功率支持

### 其他

- 支持电压之间、电流之间接错线的采样通道任意互换、反向配置
- 内置 CT 二次侧检测模块
- Vref 可外接
- 罗氏线圈

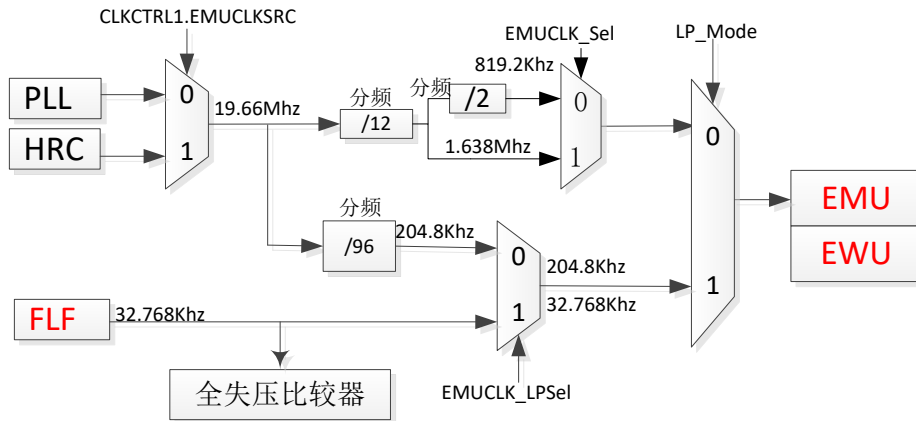
## 22.2 系统结构



## 22.3 计量单元功能

### 22.3.1 工作模式

EMU 三相计量单元提供两种计量模式：正常计量模式(Normal Mode)、低功耗低频计量模式(LP Mode) 模式切换的时钟结构如下：



### 22.3.2 模数转换器

三相计量单元内置 7 路完全独立的 22Bit 二阶  $\Sigma$ - $\Delta$  ADC。系统时钟 819.2K / 1.6384M 可选，OSR64/128 可选，采用双端差分信号输入，输入最大正弦信号(满量程)峰值是 900mV。

每路 ADC 都有其模拟增益放大器 (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。通过寄存器可以对 7 路 ADC 独立配置模拟放大倍数。其中电压通道由 PGACON.PGAU 同时对三路电压信号进行配置，可选 1/2/8 倍

火线电流通道 Ia/Ib/Ic 采样信号由 PGACON.PGA<sub>Ia/Ib/Ic</sub> 进行放大配置，可选 1/2/8/16 倍。

零线电流通道 In 采样信号支持锰铜采样，由 PGACON.PGA<sub>In</sub> 进行放大配置，可选 1/2/4/8/16/24/32 倍。

EMU 单元内部有一个 1.2V 的高稳定度片内基准电压 Vref，典型值为 8PPM/°C，也可通过寄存器配置 Vref 的温度特性曲线，最优配置可在 -40°C~80°C 温度范围内达典型值 5PPM/°C。

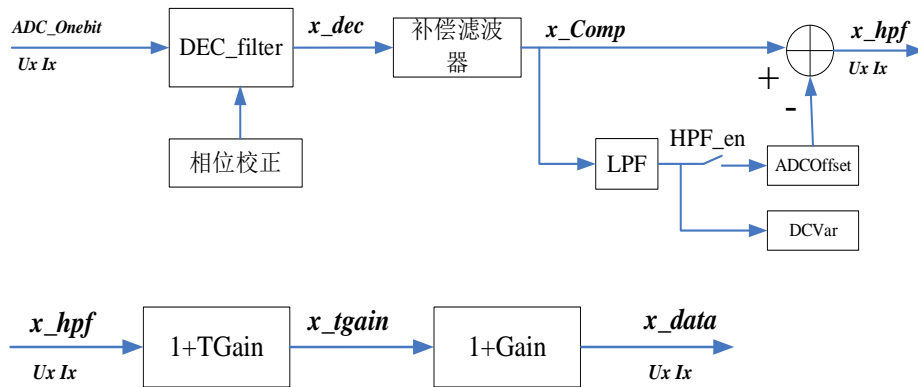
EMU 单元提供了可以对 7 路 ADC 采样信号进行数字移位放大设置，7 路 ADC 通道均有 1、2、4、8 倍四种设置。在大信号不溢出的情况下，数字增益放大可以增加小信号计算的有效位数，从而进一步提高计量精度。

为降低功耗，7 路 ADC 默认均为关闭状态。因处于不同的电压域，每路 ADC 的模拟部分和数字部分均有其独立的使能位。用户在使能某通道 ADC 时，需同时打开其 ADCE<sub>n</sub>\_ANA\_UI<sub>x</sub> 与 ADCEN\_UI<sub>x</sub>。

### 22.3.3 采样波形输出

EMU 单元将 7 路 ADC 输出的波形数据 SPL\_Ua/Ub/Uc、SPL\_Ia/Ib/Ic/In 开放给计量参数寄存器 00~18H。存放格式为 3Bytes(Bit23 符号+Bit22~0 数据)

EMU 单元具备自动消除直流 Offset 的功能。用户不需操作各通道的 ADCOffset 寄存器即完成偏置校正。功能框图如下图：



如图， $x_{Dec}$  或者  $x_{hpf}$  或者  $x_{data}$  节点的数据都可以通过寄存器 `ADCCFG.WaveSel[1:0]` 配置选择波形数据源给到 SPL 波形参数寄存器。经过高通的数据会经过二级 Gain 的增益后用于后续功率/有效值/电能相关计算。

ADCOffset 的数据会被系统实时刷新，用户也可以关闭高通，写入固定的 ADCOffset 用于后续运算。同时，计量单元也会把直流数据实时更新到直流参数计量寄存器中 DCVar。

### 22.3.4 功率

EMU 单元计量参数直接提供如下 28 种功率寄存器：

功率 \ 相	A 相	B 相	C 相	合相
全波有功功率	All_Pa	All_Pb	All_Pc	All_Pt
全波无功功率	All_Qa	All_Qb	All_Qc	All_Qt
全波视在功率	All_Sa	All_Sb	All_Sc	All_St
基波有功功率	Fund_Pa	Fund_Pb	Fund_Pc	Fund_Pt
基波无功功率	Fund_Qa	Fund_Qb	Fund_Qc	Fund_Qt
基波视在功率	Fund_Sa	Fund_Sb	Fund_Sc	Fund_St
谐波有功功率	Har_Pa	Har_Pb	Har_Pc	Har_Pt

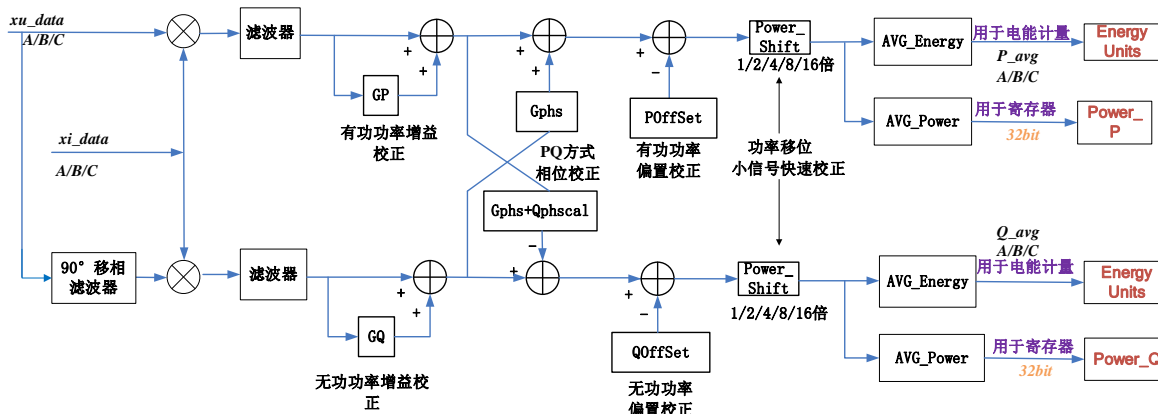
且每一相均提供各自独立的增益校正、相位校正 PowerOffset 偏置校正。其校表寄存器与相应功率的对应关系如下：

校正	功率	A 相	B 相	C 相
全波增益校正	有功功率	All_GPa	All_GPb	All_GPc
	无功功率	All_GQa	All_GQb	All_GQc
	视在功率	All_GSa	All_GSb	All_GSc
全波相位校正	分段 0	All_GphsA0	All_GphsB0	All_GphsC0
	分段 1	All_GphsA1	All_GphsB1	All_GphsC1
	分段 2	All_GphsA2	All_GphsB2	All_GphsC2
全波偏置校正	有功功率	All_PowerOffset_Pa	All_PowerOffset_Pb	All_PowerOffset_Pc
	无功功率	All_PowerOffset_Qa	All_PowerOffset_Qb	All_PowerOffset_Qc
	有功功率	Fund_GPa	Fund_GPb	Fund_GPc

基波增益校正	无功功率	Fund_GQa	Fund_GQb	Fund_GQc
	视在功率	Fund_GSa	Fund_GSb	Fund_GSc
基波相位校正	分段 0	All_GphsA0	Fund_GphsB0	Fund_GphsC0
	分段 1	Fund_GphsA1	Fund_GphsB1	Fund_GphsC1
	分段 2	Fund_GphsA2	Fund_GphsB2	Fund_GphsC2
基波偏置校正	有功功率	Fund_PowerOffset_Pa	Fund_PowerOffset_Pb	Fund_PowerOffset_Pc
	无功功率	Fund_PowerOffset_Qa	Fund_PowerOffset_Qb	Fund_PowerOffset_Qc
谐波增益校正	有功功率	Har_GPa	Har_GPb	Har_GPc
谐波偏置校正	有功功率	Har_PowerOffset_Pa	Har_PowerOffset_Pb	Har_PowerOffset_Pc

### 22.3.4.1 分相有功功率/无功功率

各通道采样数据经过高通并各自的温度补偿增益、通道增益以后进行如下处理：



如图所示，A/B/C 分相的电压数据与对应分相的电流数据相乘，经过滤波器、增益、PQ 方式相位校准、小信号 Poffset 校准，并提供加速脉冲的功率移位后计算出瞬时有功功率。

同理，A/B/C 分相电压数据经 90°Hilbert 移相滤波器后经上述运算得到瞬时无功功率。

计量单元提供可设置的平均模块寄存器：

AVG\_Power，用于对瞬时功率进行设定平均点数的平均后得到平均功率寄存器。

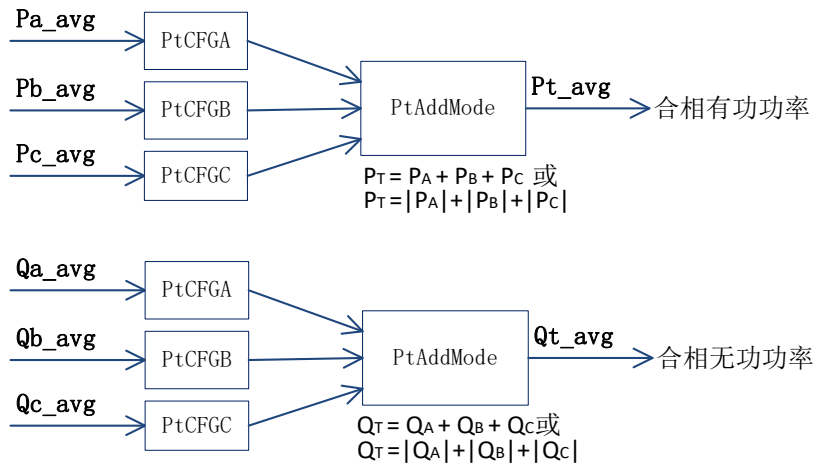
AVG\_Energy，对瞬时功率进行设定点数的平均后，得到平均功率作用于电能模块的功率源头。

用户可通过修改 AVG\_Energy 寄存器的平均点数值来满足不同的试验环境与应用场景。

### 22.3.4.2 合相有功功率/无功功率

在计算出各分相的有功功率和无功功率以后，各分相的平均功率代数和(默认)即为合相功率。EMU 单元提供了寄存器控制：

1. EMUCFG.PtCFGx (x = A, B, C)，可直接选择某分相是否参与合相功率的计算，是否取反后再参与合相功率的计算
2. EnergyCon.PtAddMode，选择各分相功率带符号相加计算合相功率（也即代数和），或选择各分相功率取绝对值相加计算合相功率。



三相四线制、三相三线制配置：

三相四线制的合相功率为： $P_T = P_A + P_B + P_C$ ；  $Q_T = Q_A + Q_B + Q_C$ ；

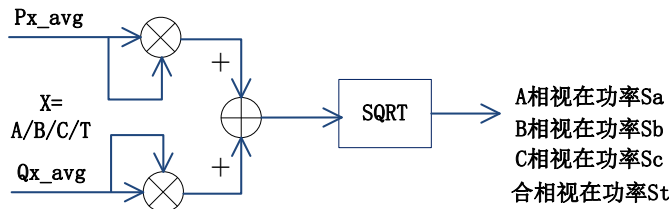
如果用户系统为三相三线模式，则需要 EMUCFG.PtCFGB 寄存器配为 B 相不参与计算。此时合相功率  $P_T = P_A + P_C$ ；  $Q_T = Q_A + Q_C$ ；

如果某分相的功率，处于潜动状态，那么该分相功率将以 0 参与合相计算。

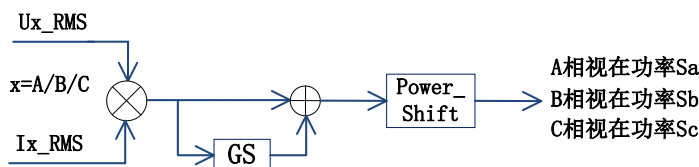
### 22.3.4.3 视在功率

EMU 单元提供两种视在功率计算方式：PQS 方式和 RMS 方式。可通过 EMUCFG.PowerSCal\_Sel 选择。

PQS 视在功率（公式一）： $S = \sqrt{P^2 + Q^2}$ ；

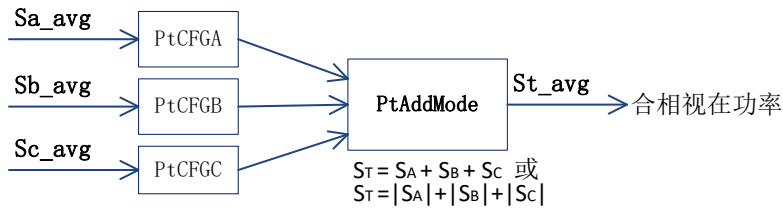


RMS 视在功率（公式二）： $S = U_{rms} * I_{rms}$ ，



其中 PQS 方式视在功率的校正，由有功和无功的校正决定。RMS 方式提供单独的 GS 增益校正，并和有功无功共用 Power\_Shift。

合相视在功率  $S_t = S_a + S_b + S_c$  即：



## 22.3.5 有效值测量

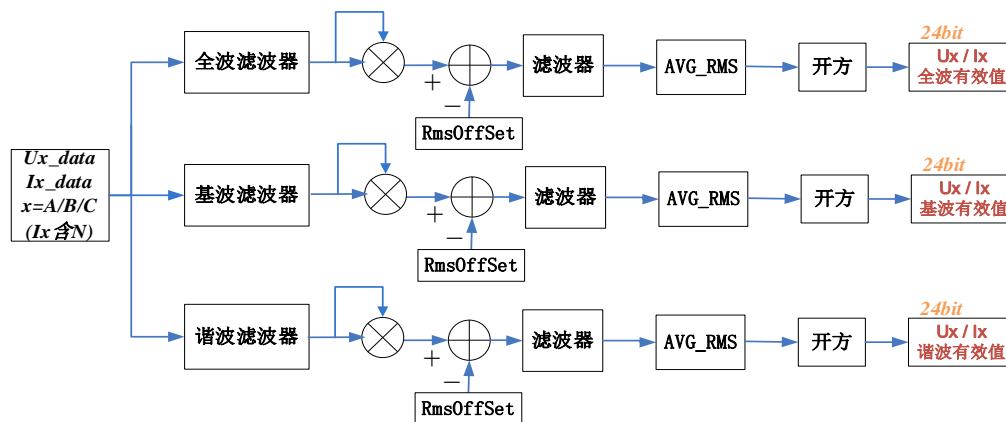
EMU 单元提供 7 路 ADC 的全波、基波、谐波，电压、电流有效值，线电压有效值，电压、电流矢量和有效值。

(注：芯片的工频同步系统也提供半波有效值，详见半波有效值与工频同步系统介绍)

有效值 \ 通道	UA	UB	UC	IA	IB	IC	IN
全波有效值	All_RMS_ Ua	All_RMS_ Ub	All_RMS_ Uc	All_RMS _Ia	All_RMS _Ib	All_RMS _Ic	All_RMS _In
基波有效值	Fund_RMS _Ua	Fund_RMS _Ub	Fund_RMS _Uc	Fund_RM S_Ia	Fund_RM S_Ib	Fund_RM S_Ic	Fund_RM S_In
谐波有效值	Har_RMS_ Ua	Har_RMS_ Ub	Har_RMS_ Uc	Har_RMS _Ia	Har_RMS _Ib	Har_RMS _Ic	Har_RMS _In
线电压有效值 (全波或基波)	RMS_UAB	RMS_UBC	RMS_UAC				
矢量和有效值 (全波或基波)	RMS_Ut			RMS_It			

### 22.3.5.1 电压电流有效值

电压、电流有效值计算，同功率计算同数据源头，即各通道采样数据经过高通并各自的温度补偿增益、通道增益以后进行自乘并后续处理。

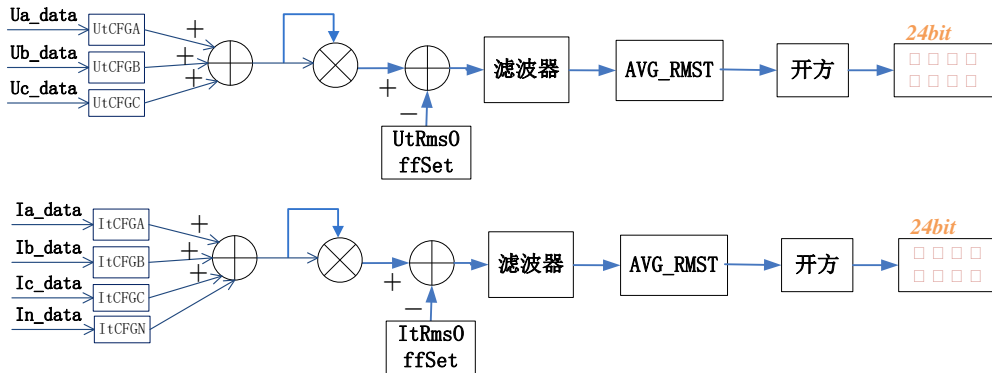


高通与通道 Gain 后的数据自乘以后过低通，并进行 RMSOffset 校正得到瞬时有效值平方，经过 AVG\_RMS 平均模块并开平方以后给到有效值寄存器更新。

EMU 计量单元提供了自乘并 RMSOffset 校正后的节点数据存入寄存器

注意功率寄存器和有效值寄存器分时使用各自的平均模块，功率使用 AVG\_Power，有效值使用 AVG\_RMS。

### 22.3.5.2 矢量和有效值



EMU 单元提供电压/电流矢量和寄存器，将 U 各通道/I 各通道的高通并各自 Gain 之后的瞬时数据进行加法/减法操作，经过独立于常规有效值的 AVG\_RMST 设置电压/电流矢量和寄存器更新频率。

UtCFGx 和 ItCFGx 的作用为选择某一相数据：是否参加矢量和运算，或者是否取反后参加矢量和运算。如果用户未使用零线通道电流采样，通过寄存器 ItCFGN 把零线设置为不参与矢量和运算即可。

用户也可通过 UtFund\_En 和 ItFund\_En 选择矢量和有效值寄存器是来自全波数据 / 基波数据。

如果用户系统处于三相三线模式，需要手动操作 UtCFGB/ItCFGB，使 B 相不参与计算即可。

### 22.3.5.3 线电压有效值

EMU 单元提供线电压有效值寄存器。用户可以通过 FundLineU\_En 选择线电压有效值为全波线电压或者基波线电压。

当系统处于三相三线模式时，用户可通过寄存器 UacSel 选择是否使用 Ua-Uc 代替 Ub 计算 B 相电压有效值。

### 22.3.6 频率

通过电压采样信号过零方式得到分相、合相频率，FreqUa/FreqUb/FreqUc/FreqU，频率分辨率约为 0.001Hz。基准信号过零无效时(SignalStatus.ZCValid\_Ua/ZCValid\_Ub/ZCValid\_Uc)，不更新频率值。

电压采样信号过零数据源可通过 ADCCFG.bit19. ZcSrSel 选择全波或者基波过零。

### 22.3.7 角度

通过采样信号过零方式计算三路电压、四路电流共计七路信号与角度基准信号之间的角度，提供七个角度寄存器(PhaseAngle\_Ua/PhaseAngle\_Ub/PhaseAngle\_Uc/ PhaseAngle\_Ia/ PhaseAngle\_Ib/ PhaseAngle\_Ic/ PhaseAngle\_In)，基准信号选择位AngleBaseSel可以将七路信号中的任意一路配置为基准信号，分辨率约为 0.01

度。

基准信号可通过AutoAngBase选择固定基准信号，也可根据选择基准信号是否有效自动切换。可通过SignalStatus.AngleBaseFlag显示当前基准信号选择状态。当AutoAngBase = 0 时，按照AngBaseSel的配置选择基准信号，并且不进行自动切换；AutoAngBase=1时，按照AngBaseSel的配置选择基准信号，并且能够在所选基准信号过零无效时自动切换到其他通道（In除外），切换的条件是当前配置的基准信号过零无效，并且根据切换优先级判断，选择优先级高且通道过零有效的通道，切换的优先级是 $U_a > U_b > U_c > I_a > I_b > I_c$ ，在当前所有通道都过零无效，此时基准信号不进行切换，继续进行过零是否有效判断，一旦有某路过零有效，则基准信号自动切换过去。

通道过零有效状态可通过过零有效状态标志显示(SignalStatus.ZCValid\_Ua/ZCValid\_Ub/ZCValid\_Uc/ZCValid\_Ia/ZCValid\_Ib/ZCValid\_Ic/ ZCValid\_In)，基准信号过零无效时，所有角度值不更新；测试通道信号过零无效时，测试通道角度值不更新，采样信号过零数据源可通过ADCCFG.bit19.ZcSrSel选择全波或者基波过零。

### 22.3.8 功率因数

提供全波分相、合相功率因数，基波分相、合相功率因数。功率因数计算方法用有功功率/视在功率得到。分相、合相视在功率计算方式可通过EMUCFG.bit3.PowerSCal\_Sel选择：PQS方式和RMS方式。

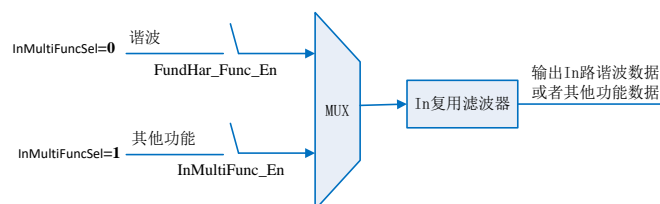
### 22.3.9 多功能滤波器设计

EMU 计量单元，提供多功能滤波器设计。

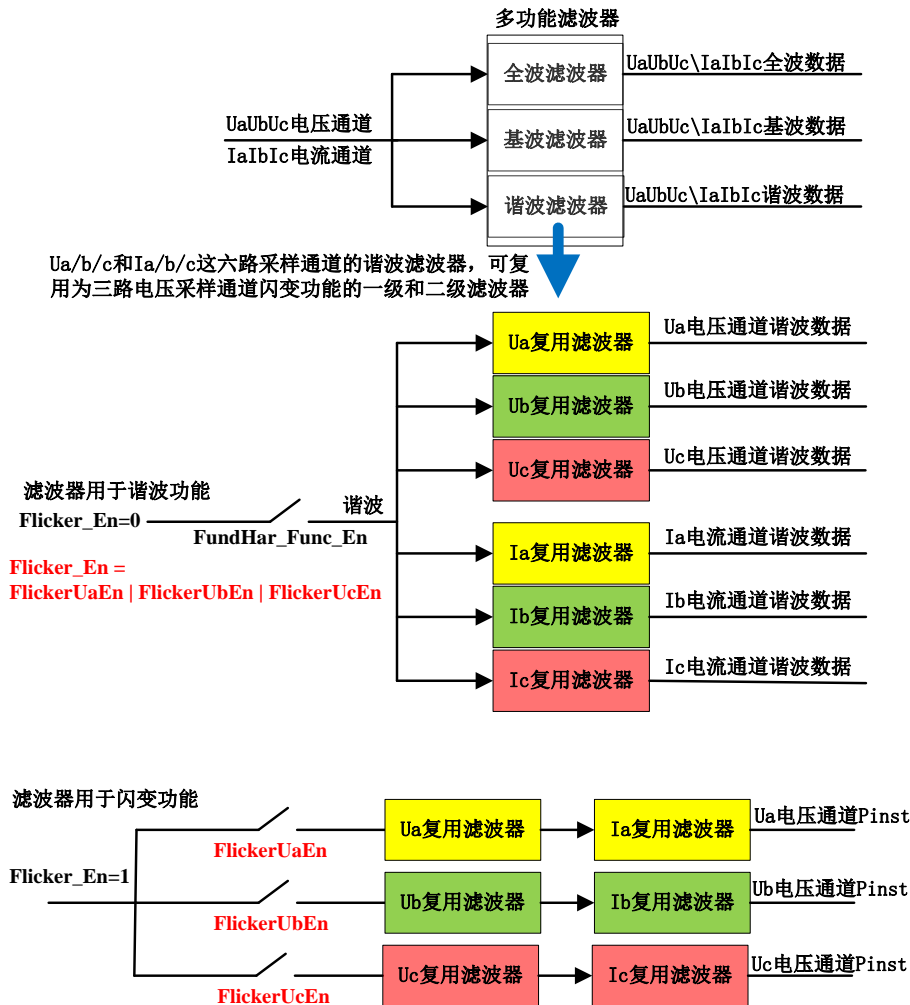
EMU 计量单元的  $U_a/U_b/U_c$ 、 $I_a/I_b/I_c$ 、In 这 7 路 ADC 采样通道，每路采样通道在高频并各自通道增益后，均有三组滤波器。即总 21 个滤波器。

这三组滤波器，常规功能设计为分别用于该通道数据的全波、基波、谐波滤波器。其中，7 路 ADC 通道，全波滤波器使用同一组系数，基波滤波器使用同一组系数，谐波滤波器使用同一组系数。另外谐波滤波器组还可以被复用为闪变滤波器用于实现闪变检测。

关于 In 零线通道：In 通道除了全波、基波、谐波滤波器的功能以外，In 通道的谐波滤波器还可以复用为多功能滤波器，用于其他自定义的功能。



$U_a/U_b/U_c/I_a/I_b/I_c$  谐波滤波器可通过设置FlickerCfg.FlickerUaEn/FlickerUbEn/ FlickerUcEn选择用于闪变功能，只要有一个FlickerUxEn = 1，6路谐波滤波器都用于闪变功能，关系图如下：



用户可通过  $FundHarCFG.bit0.FundHar\_Func\_En$  寄存器配置，选择使能或者关闭基波谐波功能。当  $FundHar\_Func\_En = 1$  时，可得到 7 路基波谐波有效值，基波有功、无功、视在功率，谐波有功功率。基波谐波能量可通过能量通道累加使能位  $CFxRun$  使能计算得到。否则当  $FundHar\_Func\_En = 0$ ，则关闭基波谐波功能。

谐波可通过  $FundHarCFG.bit2.Har\_AlgMode$  选择计算方式， $Har\_AlgMode = 1$ ，谐波滤波器方式， $Har\_AlgMode = 0$  全波-基波计算。当闪变功能使能后，谐波只能使用全波-基波计算。

通过调用钜泉提供的  $HT7x2x.lib$  文件中的函数，可以对多功能滤波器进行配置。

### 22.3.10 半波峰值

EMU 单元取 7 路 ADC 每个通道的高通数据经过全波或基波滤波器 ( $ADCCFG.ZcSrSel$ ) 后每两个过零点之间绝对值的最大值作为半波峰值，实时更新到半波峰值寄存器。

### 22.3.11 过零

EMU 单元用于过零判断的数据源头可选全波数据或基波数据(寄存器  $ZcSrSel$ )。

过零欠时事件(过零的时间过短): 某 ADC 通道, 如果连续两个双向过零点 a 和 b 之间的点数小于 ZCNumMin 时, 认为发生过零欠时事件。当发生过零欠时事件时, 系统会认为第二个过零点 b 无效, 直到有后续过零点 c 与第一个过零点 a 之间的点数大于等于 ZCNumMin, 才当做正常的过零事件处理。

过零超时事件(过零的时间过长): 如果从过零点 a 后, 连续 ZCNumMax 都未再次发生双向过零, 则认为发生过零超时事件, 此时系统将会强制发生一次过零事件并认为该路发生半周波过零丢失事件, 给出相应的过零丢失标志。但是系统内部将过零超时事件当作正常的过零事件来处理。

取高通数据经过全波或基波滤波器 (ADCCFG.ZcSrSel) 后每两个过零点之间绝对值的最大值作为半波峰值, 实时更新到寄存器。

过零有效:

HT762X 的电压过零阈值 UZCLVL 和电流过零阈值 IZCLVL 均可设, 当某通路基波数据连续 (NumFund<<5) 个点的绝对值的最大值大于过零阈值时, 认为过零有效, 过零有效标志 ZCValid\_Ux/Ix 置位, 否则过零无效。

过零有效的作用范围:

1. 过零中断, 系统只有在过零有效时才进行过零方向的判断

通过 ZCd[1:0] 可选正向过零、反向过零、双向过零, 数据源通过 ZCINTSrc\_Sel[1:0] 可选高通前、高通后、基波滤波器之后的数据。

2. 频率、角度、相序

当过零无效时, 频率、角度、相序的结果均不可靠

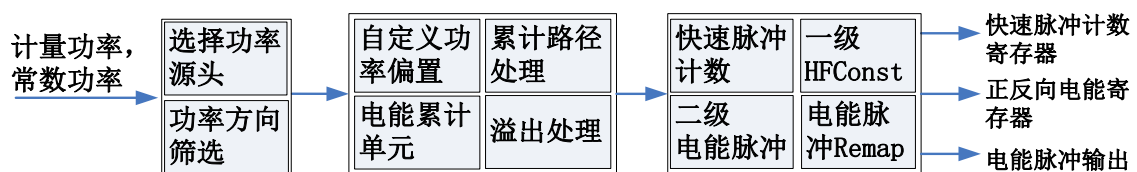
不受过零有效影响的功能:

半波峰值更新, 不受过零有效影响

Peak/Sag/UINT/IOV 功能, 不受过零有效影响

## 22.3.12 电能模块

EMU 单元计量参数直接提供 28 种功率寄存器, 并把功率接入到 EMU 提供的 24 路电能计量。每一路电能通路都有正、反两分支电能计量通道。



### 22.3.12.1 电能通路的功率源

功率与电能通道的接入关系如下:

电能通道	前端采样功率	对应电能 (正反)	备注
1	全波 A 相有功	All_EnergyPos/Neg_Pa	(1) 该 24 路电能通路的源头, 均支持常数自定义功率输入功能。 (2) 9~12 路视在功率不存在反向功率与电能, 因此 9~12 路的反向电能通路的用途为
2	全波 B 相有功	All_EnergyPos/Neg_Pb	
3	全波 C 相有功	All_EnergyPos/Neg_Pc	
4	全波合相有功	All_EnergyPos/Neg_Pt	

5	全波 A 相无功	All_EnergyPos/Neg_Qa	负常数自定义功率。 (3) 9~12 路电能通路源头, 可以通过寄存器 EnergySSrc_Sel 选择源头功率为全波功率或者基波功率。 (4) 1~20 均不存在累加模块的 AccuOffset, 仅谐波的 21~24 通路具备。 (5) 谐波不具备 Judge 谐波 (6) 谐波通道不允许使用瞬时功率计量电能 (7) 只有 4,8,12,16,20,24 这六个合相电能通路才可以发脉冲, 其他电能通路终点是电能寄存器。 (8) 六路合相电能通路(每各通路均包含正反向两支路)将经过脉冲 Remap 功能后从五个电能脉冲口发出电能脉冲
6	全波 B 相无功	All_EnergyPos/Neg_Qb	
7	全波 C 相无功	All_EnergyPos/Neg_Qc	
8	全波合相无功	All_EnergyPos/Neg_Qt	
9	A 相视在	Energy_Sa/Neg_SaConst	
10	B 相视在	Energy_Sb/Neg_SbConst	
11	C 相视在	Energy_Sc/Neg_ScConst	
12	合相视在	Energy_St/Neg_StConst	
13	基波 A 相有功	Fund_EnergyPos/Neg_Pa	
14	基波 B 相有功	Fund_EnergyPos/Neg_Pb	
15	基波 C 相有功	Fund_EnergyPos/Neg_Pc	
16	基波合相有功	Fund_EnergyPos/Neg_Pt	
17	基波 A 相无功	Fund_EnergyPos/Neg_Qa	
18	基波 B 相无功	Fund_EnergyPos/Neg_Qb	
19	基波 C 相无功	Fund_EnergyPos/Neg_Qc	
20	基波合相无功	Fund_EnergyPos/Neg_Qt	
21	谐波 A 相有功	Har_EnergyPos/Neg_Pa	
22	谐波 B 相有功	Har_EnergyPos/Neg_Pb	
23	谐波 C 相有功	Har_EnergyPos/Neg_Pc	
24	谐波合相有功	Har_EnergyPos/Neg_Pt	

能量单元的输入为瞬时功率、平均功率(默认)、常数自定义功率。可通过寄存器 PowerSrcCFG1、PowerSrcCFG2 做源头选择。

用户可通过 PassMode 寄存器对上述 6 组电能(各电能通路的 A/B/C/合为 1 组)做第一级功率数据分拣。对接入电能通路的功率做: 全部通过 / 只正功率通过 / 只负功率通过 的第一级功率分拣。

### 22.3.12.2 电能累计模式

功率经过第一级分拣后, 会进入该电能累计模式的分拣并对功率进行电能的累计。该模块共提供了四种能量累计模式。

模式	说明
Mode0	通道正分支代数和和累计模式, 此时只使用正向电能计量的分支进行电能累计。该模式前提下: PowerOVSel=0, 功率在累加器 SUMP 中累计, 不论以正向或反向的溢出形式累计出一个电能脉冲时((PFCnt ++/--)= abs(HFConst)), 结果均送到正向电能寄存器 EnergyPos。 PowerOVSel=1, 功率在累加器 SUMP 中累计, 累计出一个电能脉冲时: 累计正向溢出的电能, 送到正向电能寄存器。累计反向溢出的电能, 送到反向电能寄存器 EnergyNeg。

<b>Mode1</b>	<p>通道正分支绝对值累计模式，此时只使用正向电能计量的分支进行电能累计。</p> <p>此模式下不存在负向溢出的电能。功率在累加器 <b>SUMP</b> 中累计，结果均送到正向电能寄存器 <b>EnergyPos</b></p>
<b>Mode2</b>	<p>通道正反向电能累计模式，此时同时使用正、反向电能计量的分支进行电能累计。</p> <p>功率输入如果是正功率，则直接在电能正向累计的分支进行累计电能(功率在累加器 <b>SUMP</b> 中累计)，并把累计溢出的电能结果送到正向电能寄存器 <b>EnergyPos</b></p> <p>功率输入如果是负功率，则直接在电能反向累计的分支进行累计电能。功率在累加器 <b>SUMN</b> 中累计，并把累计溢出的电能结果送到反向电能寄存器 <b>EnergyNeg</b></p>
<b>Mode3</b>	<p>通道正反向电能累计模式，该模式下同时使用正、反向电能计量的分支进行电能累计。在此 <b>Mode3</b> 的前提下，用户可配置 <b>Judge_En</b> 来判断当前信号状态。有如下三种能量路径：</p> <p><b>Judge_En=0</b>：不使能当前信号状态判断，功率均在累加器 <b>SUMP</b> 中累加。<b>SUMP</b> 累加器若正向溢出，则 <b>PFCntP+1</b>，当 <b>PFCntP</b> 累计到和 <b>HFConst</b> 相等则送到正向电能寄存器 <b>EnergyPos</b>。<b>SUMP</b> 累加器若负向溢出，则 <b>PFCntN+1</b>，当 <b>PFCntN</b> 累计到和 <b>HFConst</b> 相等则送到反向电能寄存器 <b>EnergyNeg</b></p> <p><b>Judge_En=1</b>：使能当前信号状态判断。</p> <p>如果状态标志寄存器 <b>P90_Power</b> 为 0，说明此时信号状态正常。此时的电能路径将和 <b>Mode2</b> 相同。</p>

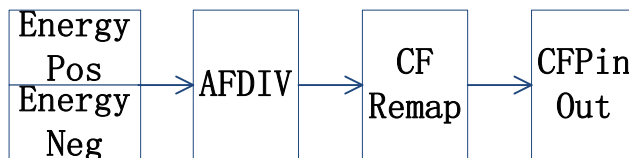
如果状态标志寄存器 P90\_Power 为 1, 说明信号处于极小功率因数状态, 此时的电能路径将和 Mode0 & PowerOVSel=1 相同。

注: 电能的累计模式, 对不同的应用场景有多种分支处理。一般推荐使用 Mode0 & PowerOVSel=1。如果需要使用正反向电能模式, 则 Mode2 和 Mode3 下的路径再根据需求进行选择。

### 22.3.12.3 电能脉冲输出

电能模块支持提高电能分辨率的设计。即当功率在选定的电能模式累计出一个相关电能以后, 电能寄存器会加 1。如果电能寄存器的值累加到等于 AFDIV 时, 电能单元才会发一个脉冲给输出 Pin。在默认状态下, 电能分辨率 AFDIV 为 1, 此时电能寄存器和电能输出脉冲 Pin 为同步关系。

另由于计量单元有 6 路合相电能寄存器并脉冲输出, 且每一路合相电能均有 正电能、负电能、正负电能(默认) 三个电能脉冲输出选项。而芯片的 GPIO 只提供 5 个电能脉冲 Pin。因此电能模块设计了 CF Remap 功能, 用户可通过 CFCFG1/2 能量脉冲控制寄存器来选择某个 CFPin 输出合相 6 路电能(每路三种脉冲输出)共计 18 种电能脉冲的任意一个。



6 路电能脉冲分别为:

All Pt	P1	N1	P1	N1
All Qt	P2	N2	P2	N2
St	P3	N3	P3	N3
Fund Pt	P4	N4	P4	N4
Fund Qt	P5	N5	P5	N5
Har Pt	P6	N6	P6	N6

用户可以通过 POS 选择电能脉冲的有效电平。POS 为 0 时, 脉冲高电平有效; POS 为 1 时, 脉冲低电平有效。脉冲输出满足下面时序关系:



### 22.3.13 自动温度补偿

EMU 单元提供自动温度补偿功能, 补偿原理为: 在计量芯片的采样系统中, 锰铜/采样电阻/分压电阻/芯片基准电压都具备各自的温度曲线。且系统中的不同器件的温度曲线趋势也是不同的。芯片提供一种一元二次

补偿的机制：根据芯片的温度传感器模块提供的数据，与用户计算的一元二次补偿的各项系数，自动计算出当前温度下的各器件温度偏移所引起的增益总量，并填入 7 路 ADC 的各通道 TGain 寄存器进行温度补偿。当 ADCCFG.bit15.Auto\_Tc\_EN = 1 时，7 路通道都打开自动温度补偿功能。Auto\_Tc\_EN = 0 时，手动温度补偿，设置 Tgain\_U / Tgain\_I / Tgain\_In 进行温度补偿。

模拟曲线公式： $y = ax^2 + bx + c$ ，可设置二次项系数、一次项系数、常数项系数

Ua/Ub/Uc 通道： TU\_CCOFF\_A / TU\_CCOFF\_B / TU\_CCOFF\_C

Ia/Ib/Ic 通道： TI\_CCOFF\_A / TI\_CCOFF\_B / TI\_CCOFF\_C

In 通道： TIn\_CCOFF\_A / TIn\_CCOFF\_B / TIn\_CCOFF\_C

自动温度补偿曲线一次项系数 Tx\_CCOFF\_B 说明：

Tx\_CCOFF\_B 的 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 Tx\_CCOFF\_B = Bit15(符号 bit)bit14~bit13+ bit12(符号 bit)bit11~bit0。

举例：

Tx\_CCOFF\_B = 0x1400，则系数为：整数部分为 0，小数部分为  $-(2^{13}-0x1400)/2^{12} = -0.75$ ，则 Tx\_CCOFF\_B =  $0 + -0.75 = -0.75$ ；

Tx\_CCOFF\_B = 0x2400，则系数为：整数部分为 1，小数部分为  $0x400/2^{12} = 0.25$ ，则 Tx\_CCOFF\_B =  $1 + 0.25 = 1.25$ 。

实际内部计算公式：

$$\text{AutoTgain} = \text{INT}[\text{Tx\_CCOFF\_A} / 32768 * (\text{TPSCODE} / 256)^2 + (\text{Tx\_CCOFF\_B}' / 4096 + \text{Tx\_CCOFF\_B}'') * (\text{TPSCODE} / 256) + \text{Tx\_CCOFF\_C} / 32768]$$

注意：

TPSCODE：有符号，16bit，由 TBS 模块温度传感器寄存器值 TMP 取高 10 位，即 TMP 寄存器值 &0xffc0。

Tx\_CCOFF\_A：有符号，表示范围(-16384,16384)，转换成小数的系数绝对值设置不能超过 0.5。

Tx\_CCOFF\_B'：有符号，Tx\_CCOFF\_B 的小数部分。

Tx\_CCOFF\_B''：有符号，Tx\_CCOFF\_B 的整数部分。

Tx\_CCOFF\_C：有符号，16bit，表示范围-32768~32767。

举例如下：

TMP 寄存器值为 0x3fff，TPSCODE = 0x3fc0，Tx\_CCOFF\_A=0x0AA0，Tx\_CCOFF\_B =0x1400，Tx\_CCOFF\_C =0x5612，则计算出来，如下：

$$\text{AutoTgain} = \text{INT}[\text{Tx\_CCOFF\_A} / 32768 * (\text{TPSCODE} / 256)^2 + (\text{Tx\_CCOFF\_B}' + \text{Tx\_CCOFF\_B}'') * (\text{TPSCODE} / 256) + \text{Tx\_CCOFF\_C} / 32768]$$

$$= \text{INT}[0x0AA0 / 32768 * (0x3fc0 / 256)^2 + (0 + -0.75) * (0x3fc0 / 256) + 0x5612 / 32768]$$

$$= \text{INT}[0x0AA0 / 32768 * (0x3fc0 / 256)^2 + (0 + -0.75) * (0x3fc0 / 256) + 0x5612 / 32768]$$

$$= 290$$

$$= 0x0122$$

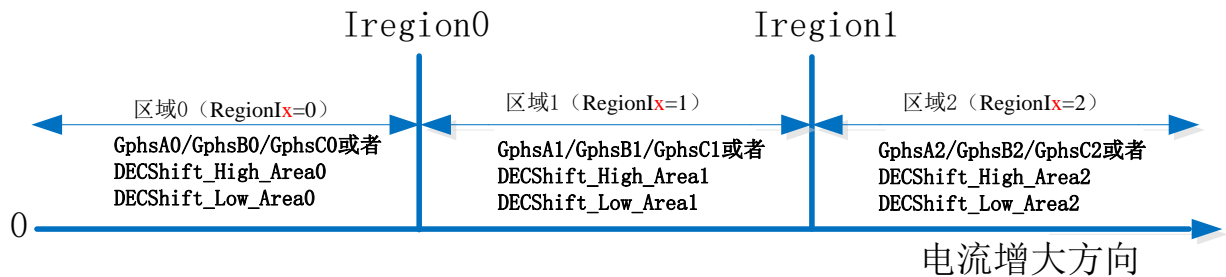
## 22.3.14 相位分段

为了更好的满足外部互感器特性，相位补偿可分为 3 段，使用相位分段寄存器 Iregion0 和 Iregion1 设置电流分段点，分段需要增加滞回 Iregion0Hys 和 Iregion1Hys，用有效值和 Iregion 对比，来选择不同的相位补偿寄存器参与相位校正。

相位校正方式有两种：PQ 方式和移采样点方式，不建议两个方式同时使用，推荐只使用其中一种相位

校正方式，不使用的这种方式相位校正相关寄存器保持复位值或者设置为全 0。

可通过 ADCCON.bit8.Phase\_Sel 可选择全波或者基波电流有效值用于相位分段校正。各相根据选择的电流有效值与 Iregionx 和迟滞 IregionxHys 的组合对比，更新状态标志 SignalStatus 的 bit10~15 各相相位分段状态 RegionIx，说明目前相位分段所处区域。



RegionIa 状态表示 Ua/Ia 的 DECSHift 和 A 相全波基波 Gphs 选择；RegionIb 状态表示 Ub/Ib 的 DECSHift 和 B 相全波基波 Gphs 选择；RegionIc 状态表示 Uc/Ic 的 DECSHift 和 C 相全波基波 Gphs 选择。

当 RegionIx=0 时，对应相可使用 Gphsx0 或者 DECSHift\_High\_Area0 和 DECSHift\_Low\_Area0 参与相位校正；当 RegionIx=1 时，对应相可使用 Gphsx1 或者 DECSHift\_High\_Area1 和 DECSHift\_Low\_Area1 参与相位校正；当 RegionIx=2 时，对应相可使用 Gphsx2 或者 DECSHift\_High\_Area2 和 DECSHift\_Low\_Area2 参与相位校正。

零线电流使用移采样点相位校正方式，根据内部 RegionIn 判断采用哪个区域的 DECSHift\_High\_Areax 和 DECSHift\_Low\_Areax。

### 22.3.15 移采样点相位校正

EMU 单元提供了支持大角度初始误差(可达 10°以上)的移采样点方式相位校正。

由于外部硬件导致 U 和 I 有一定的角差，校正的目的为通过移动 onebit 的点数进行相位调整。

以 emuclk=1.638M 为例，假设采样电网为准确 50Hz，

一信号周期采样点数为  $1638400/50 = 32768$  个，一周期为 360°，因此移动 1 个 onebit 将会移动相位： $1/32768 * 360 = 0.011^\circ$ ，而移动 1 个 dec 将会移动  $0.011^\circ * OSR = 1.406^\circ$  (设 OSR=128)，等效为移动 128 个 onebit 的结果。

在 0.5L 理想情况下，移动 U 的 1 个 Onebit，将会使电压相对原电压滞后 1 个 Onebit，带来的误差为  $(\cos(60 - 0.011) - \cos(60)) / \cos(60) \approx 0.03325\%$

同样地，移动 I 的 1 个 Onebit，将会使电流相对原电流滞后 1 个 Onebit，带来的误差为  $(\cos(60 - 0.011) - \cos(60)) / \cos(60) \approx 0.03325\%$

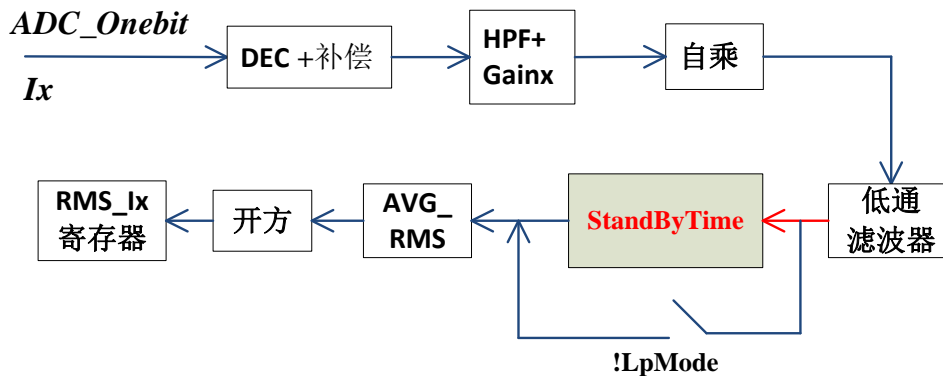
相关寄存器详见 DecShift。

### 22.3.16 低频计量模式

EMU 单元提供低频计量模式，能够使 EMU 计量时钟工作于低频模式(204.8K / 32K)，并在该模式下提供相应状态机设计，从而保证快速获取有效值的精度。(详见计量单元工作模式介绍)

实测在低频计量模式下，可实现在开启 ADC 的时刻，40mS 达到 1%精度误差，而 60mS 可达到 0.1%的

精度误差。



正常模式，EMU 模块的时钟源可选 PLL、HRC。

低频计量模式，EMU 模块的时钟源可选 fosc，或者把 Femu 切换为 204.8KHz。且在 204.8K 或 32K 下的 OSR 固定死为 64，不可选。

在低功耗模式下（正常模式也可用，一般用途低频计量模式下）UconstEn=1，使能常数电压 UCONST 和电流通道 RMS 计算视在功率，除了在低功耗计量应用外，正常模式也可开启常数电压计量功能。

## 22.4 检测与电能质量

钜泉提供完整的电能质量分析相关标准库函数，用户可直接调用实现。

### 22.4.1 起动潜动

EMU 单元提供三个起动阈值寄存器：PStart，QStart，HarStart。

起动阈值判断源头可选：电流有效值/功率，瞬时数据/平均数据，基波数据/全波数据。其中全波数据/基波数据使用 PStart，QStart。谐波数据单独使用 HarStart。起动数据源的选择，默认均为使用平均数据。

全波、基波起动：

#### (1) 功率起动方式 1：P,Q 单独判断

|PowerP|根据 PStart 判断潜动，|PowerQ|根据 QStart 判断潜动(default)，即如果|PowerP|> PStart，有功起动；|PowerQ|>QStart 则无功起动，有功与无功各自独立。

PowerP 指的是 All\_P 和 Fund\_P，PowerQ 指的是 All\_Q 和 Fund\_Q

#### (2) 功率起动方式 2：P|Q 方式判断

PQ 联合判断潜动，即如果满足|PowerP|>PStart 和|PowerQ|>QStart 任一条件，有功和无功都会同时起动其通路计量。

#### (3) 有效值起动方式 1：即基波有效值方式

如果各相（A/B/C 相）基波电流有效值 Fund\_RMSIa/Ib/Ic >PStart，则该相的全波/基波，有功无功视在功率全都起动计量。（注意用的是平均后的有效值。瞬时有效值不可去参与启动潜动）

#### (4) 有效值起动方式 2：即全波有效值方式

如果各相（A/B/C 相）全波电流有效值有 All\_RMSIa/Ib/Ic >PStart 则：该相的全波，有功无功视在功率全都

启动计量。(注意用的是平均后的有效值。瞬时有效值不去参与启动潜动)。

注意基波的有功无功视在功率是否启动计量，仍然使用各相(A/B/C相)基波电流有效值  $Fund\_RMSIa/Ib/Ic > PStart$  判断。

#### 视在功率启动:

对于视在功率的启动，有效值方式不管1和2，都是完全和基波全波有功、无功同步的。

而功率方式不论1和2，视在都选用P|Q的方式，即只要P或者Q有一个启动，S就会启动。(All\_P或All\_Q有一个启动，则启动All\_S，Fund\_P或Fund\_Q有一个启动，则启动Fund\_S)

#### 谐波功率启动:

HarStart\_CFG 可选有效值/功率，=1选择功率，=0选择有效值，但是都必须是平均值后，不可以选择瞬时功率。

功率方式:  $|Har\_P| > HarStart$ ，则I通道谐波功率开始启动计量;

有效值方式:  $Har\_RMS\_I > HarStart$ ，则I通道谐波功率开始启动计量。

#### 合相功率的启动潜动:

如果某个分相的功率处于潜动状态(注意应该是潜动标志，而并非功率值小于启动阈值，因为有可能是有效值方式使功率处于潜动状态)，那么该相的分相功率不去参与合相功率的计算。

另外，合相电能本身并没有启动潜动的概念，主要指的三个分相参与电能累加的条件，先是参与合相功率的计算，然后合相功率直接去电能计量。

芯片提供各分相，全波/基波/谐波，有功/无功/视在的潜动标志Noload寄存器。

## 22.4.2 电压暂升 PEAK/暂降 SAG

根据数据源选择位ADCCON.bit14.PeakSagSrc选择对半波峰值或者半波有效值进行peak/sag判断，选择数据源更新后才进行peak/sag判断。以半周波为单位，每一个半周波进行一次判别，判断是对选择的数据源取绝对值后与peak/sag检测阈值进行比较。

PEAK和SAG事件开始中断标志独立 PeakUaIF/PeakUbIF/PeakUcIF/SagUaIF/SagUbIF/SagUcIF，结束中断标志各相共用一个 UaEndIF/UbEndIF/UcEndIF。

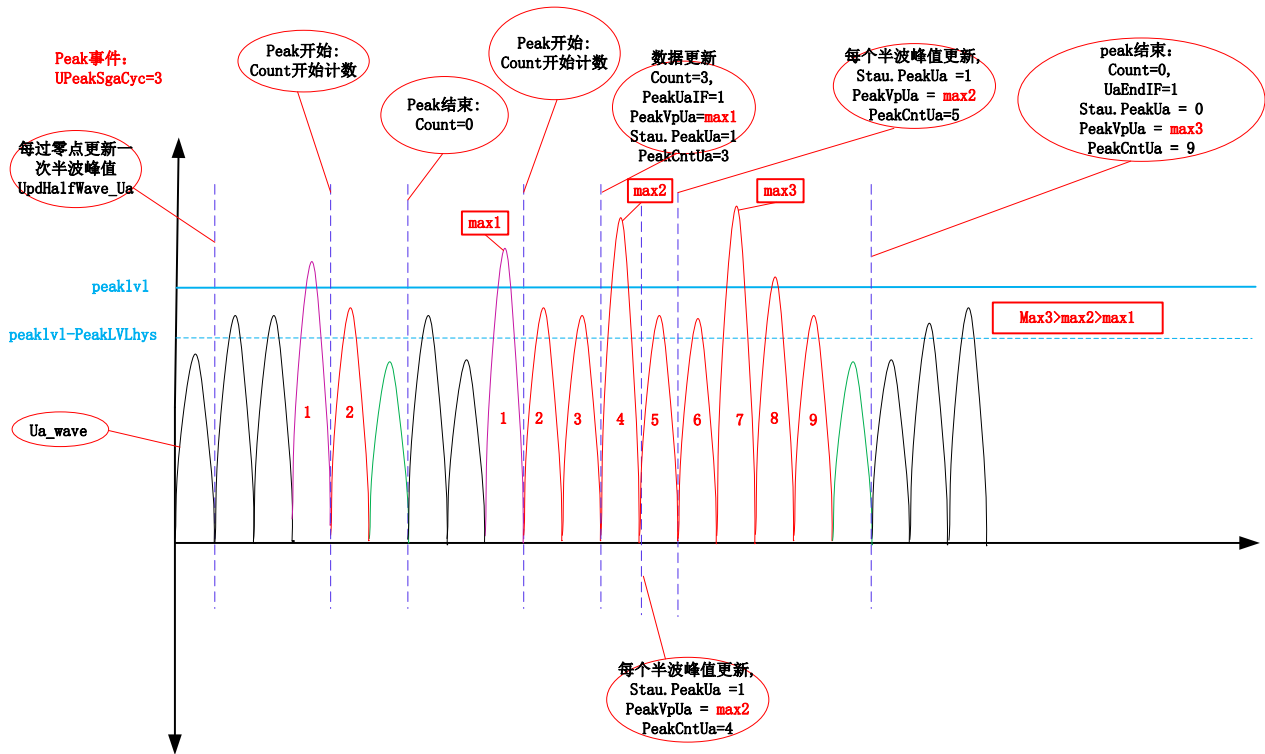
#### 1) PEAK事件检测描述:

当ADCCON.bit13.PeakEn = 0，功能关闭，清状态标志寄存器PeakSagIntIovStau.PeakUx、期间最大值寄存器PeakVpUx和期间半波周数PeakCntUx。

当ADCCON.bit13.PeakEn = 1，功能开启，某路电压第一个半周波检测信号数据源的绝对值大于PeakLVL，记为Peak事件判断开始，只要半周波检测信号的绝对值一直大于等于  $(PeakLVL - PeakLVLHys)$ ，则一直处于Peak状态，当Peak状态持续的半周波计数Count大于等于UpeakSagCyc设定的半周波数，则判定发生Peak事件，置起PeakUxIF中断标志，置起状态标志PeakSagIntIovStau.PeakUx = 1，更新期间最大值寄存器PeakVpUx和期间半波个数PeakCntUx，之后每半个周波持续比较，若半周波检测信号数据源的绝对值大于等于  $(PeakLVL - PeakLVLHys)$ ，则每个半周波更新一次PeakSagIntIovStau.PeakUx、PeakVpUx和PeakCntUx。当半周波检测信号数据源的绝对值小于  $(PeakLVL - PeakLVLHys)$ 时，就退出Peak事件，置起UxEndIF退出中断标志，PeakSagIntIovStau.PeakUx = 0。Peak事件结束后，期间最大值寄存器PeakVpUx和期间半波个数PeakCntUx寄存器保持最后一次更新值不变，直到再一次发生Peak事件，持续发生UpeakSagCyc个半周波，更新新的值于期间最大值寄存器PeakVpUx和期间半波个数PeakCntUx寄存器中。

例如：如下图所示，以Ua通道选择半波峰值为判断数据源，PeakEn = 1，当检测Ua半波峰值的绝对值大

于PeakLVL，第一次发生peak事件，但半波个数不满足个数大于等于UpeakSagCyc，又检测Ua半波峰值小于(PeakLVL-PeakLVLHys)，此时不更新数据，再检测Ua半波峰值的绝对值大于PeakLVL，当持续UpeakSagCyc个半周波都满足peak条件，置起中断标志PeakUaIF = 1, PeakSagIntIovStau.PeakUA = 1，更新期间最大值PeakVpUa = max1, PeakCntUa = 3，每半个周波持续检测，更新PeakVpUa, PeakSagIntIovStau.PeakUA, PeakCntUa,直到半波峰值的绝对值小于(PeakLVL-PeakLVLHys)，退出peak状态，置起结束中断标志UaEndIF = 1, PeakCntUa = 9, PeakVpUa = max3, PeakSagIntIovStau.PeakUA = 0。



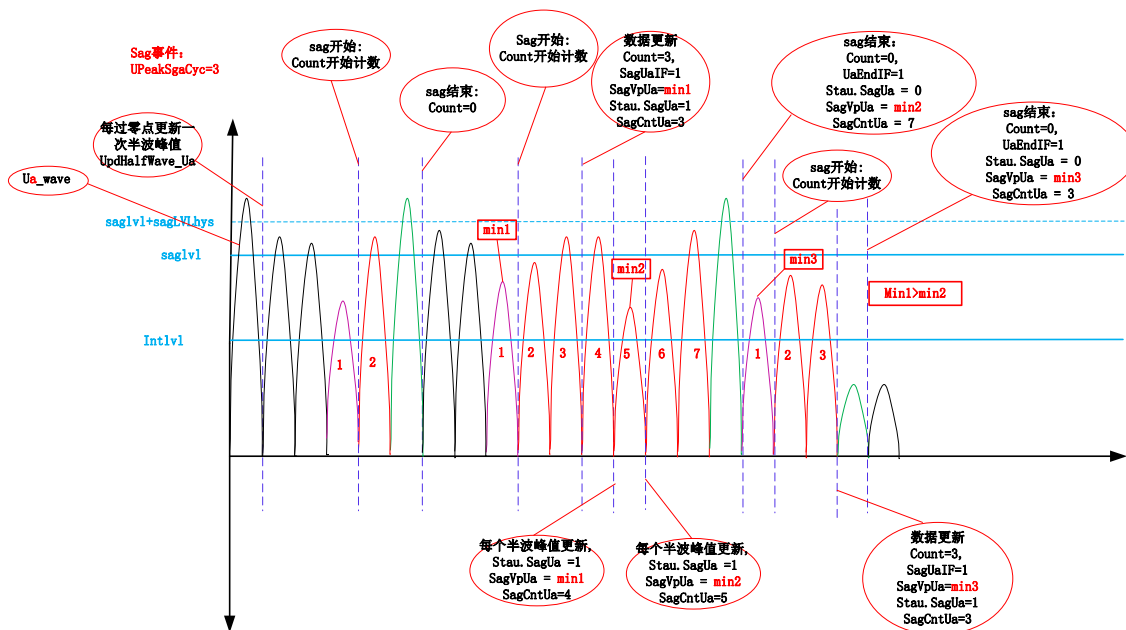
## 2) SAG 事件检测描述:

当 ADCCON.bit12.SagEn = 0，功能关闭，清状态标志寄存器 PeakSagIntIovStau.SagUx、期间最小值寄存器 SagVpUx 和期间半波周数 SagCntUx。

当 ADCCON.bit12.SagEn = 1，功能开启，某路电压波形采样值第一个半周波检测信号的绝对值小于 SagLVL 且大于 IntLVL，记为 Sag 事件判断开始，只要半周波检测信号的绝对值一直小于等于 (SagLVL+SagLVLHys)且大于等于 IntLVL，则一直处于 Sag 状态，当 Sag 状态持续的半周波计数值 Count 大于等于 UpeakSagCyc 设定的半周波数，则判定发生 Sag 事件，置起中断标志 SagUxIF，置起状态标志 PeakSagIntIovStau.SagUx = 1，更新期间最小峰值寄存器 SagVpUx 和 SagCntUx，之后每个半波持续比较，若半周波检测信号的绝对值小于等于 (SagLvl+SagLvlHys)且大于 IntLVL，则每个半周波更新一次 PeakSagIntIovStau.SagUx、SagVpUx 和期间半周波个数 SagCntUx。当半周波检测信号的绝对值大于 (SagLVL+SagLVLHys)或者小于 IntLVL 时，就退出 Sag 事件，置起 UxEndIF 退出中断标志，PeakSagIntIovStau.SagUx = 0。Sag 事件结束后，期间最小值寄存器 SagVpUx 和期间半周波个数 SagCntUx 寄存器保持最后一次更新值不变，直到再一次发生 Sag 事件，持续发生 UpeakSagCyc 个半周波，更新新的值于期间最小峰值寄存器 SagVpUx 和期间半波个数 SagCntUx 寄存器中。

例如：如下图所示，以Ua通道选择半波峰值为判断数据源，SagEn = 1，当检测Ua半波峰值的绝对值小于 SagLVL，第一次发生Sag事件，但半波个数不满足个数大于等于UpeakSagCyc，又检测信号大于 (SagLVL+SagLVLHys)，此时不更新数据，再检测Ua半波峰值的绝对值小于SagLVL，当持续UpeakSagCyc个

半周波都满足Sag条件，置起中断标志SagUaIF = 1, PeakSagIntIovStau.SagUA = 1, 更新期间最小值SagVpUa = min1, SagCntUa = 3, 每半个周波持续检测，更新SagVpUa, PeakSagIntIovStau.SagUA, SagCntUa, 直到半波峰值大于(SagLVL+SagLVLHys), 退出Sag状态，置起结束中断标志UaEndIF = 1, SagCntUa = 7, SagVpUa = min2, PeakSagIntIovStau.SagUA = 0。然后检测Ua半波峰值的绝对值小于SagLVL, 当持续UpeakSagCyc个半周波都满足Sag条件，置起中断标志SagUaIF = 1, PeakSagIntIovStau.SagUA = 1, 更新期间最小值SagVpUa = min3, SagCntUa = 3, 检测Ua半波峰值的绝对值小于IntLVL, 退出Sag状态，置起结束中断标志UaEndIF = 1, SagCntUa = 3, SagVpUa = min3, PeakSagIntIovStau.SagUA = 0。



注：电压暂升/暂降，用户可直接调用钜泉标准库函数实现。

## 22.4.3 电压短时中断 INT

根据数据源选择位ADCCON.bit14.PeakSagSrc选择对半波峰值或者半波有效值进行INT判断,选择数据源更新后才进行INT判断。以半周波为单位，每一个半周波进行一次判别，判断是对选择的数据源取绝对值后与INT检测阈值进行比较。

INT事件提供分相开始、结束中断INTUaIF/INTUbIF/INTUcIF、UaINTEndIF/UbINTEndIF/UcINTEndIF，三相都发生INT中断INTUIF，用户需要保证在某相UxINTEndIF产生以后,清零该相的INTUxIF标志，以规避INTUIF中断被错误置起的可能性。

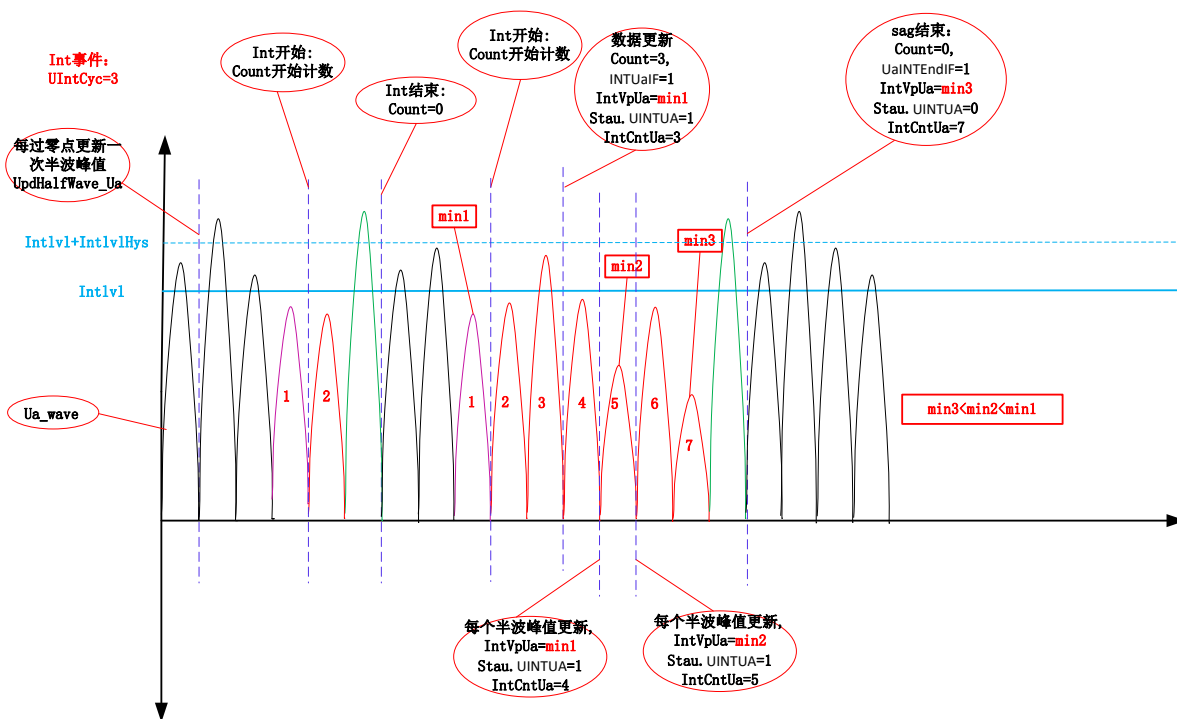
### INT 事件检测描述:

当 ADCCON.bit11.IntEn = 0, 功能关闭，清状态标志寄存器 PeakSagIntIovStau.UINTUx、期间最小值寄存器 IntVpUx 和期间半波个数 IntCntUx。

当 ADCCON.bit11.IntEn = 1, 功能开启，某路电压波形采样值第一个半周波检测信号的绝对值小于IntLVL, 记为INT事件判断开始，只要半周波检测信号的绝对值一直小于等于(IntLVL+IntLVLHys), 则一直处于INT状态，当INT状态持续的半周波计数值Count大于等于UIntCyc设定的半周波数，则判定发生INT事件，置起中断标志INTUxIF, 置起状态标志PeakSagIntIovStau.UINTUx = 1, 更新期间最大峰值寄存器IntVpUx和IntCntUx, 之后每个半波持续比较，若半周波检测信号的绝对值小于等于(IntLVL+IntLVLHys),

则每个半周波更新一次  $\text{PeakSagIntIovStau.UINTUx}$ 、 $\text{IntVpUx}$  和期间半周波个数  $\text{IntCntUx}$ 。当半周波检测信号的绝对值大于  $(\text{IntLVL} + \text{IntLVLHys})$  时，就退出 INT 事件，置起  $\text{UxINTEndIF}$  退出中断标志， $\text{PeakSagIntIovStau.UINTUx} = 0$ 。INT 事件结束后，期间最小值寄存器  $\text{IntVpUx}$  和期间半周波数  $\text{IntCntUx}$  寄存器保持最后一次更新值不变，直到再一次发生 INT 事件，持续发生  $\text{UIntCyc}$  个半周波，更新新的值于期间最小值寄存器  $\text{IntVpUx}$  和期间半周波个数  $\text{IntCntUx}$  寄存器中。

例如：如下图所示，以  $\text{Ua}$  通道选择半波峰值为判断数据源， $\text{IntEn} = 1$ ，当检测  $\text{Ua}$  半波峰值的绝对值小于  $\text{IntLVL}$ ，第一次发生 INT 事件，但半波个数不满足个数大于等于  $\text{UIntCyc}$ ，又检测信号大于  $(\text{IntLVL} + \text{IntLVLHys})$ ，此时不更新数据，再检测  $\text{Ua}$  半波峰值的绝对值小于  $\text{IntLVL}$ ，当持续  $\text{UIntCyc}$  个半周波都满足 INT 条件，置起中断标志  $\text{INTUaIF} = 1$ ， $\text{PeakSagIntIovStau.UINTUa} = 1$ ，更新期间最小值  $\text{IntVpUa} = \text{min1}$ ， $\text{IntCntUa} = 3$ ，每半个周波持续检测，更新  $\text{IntVpUa}$ ， $\text{PeakSagIntIovStau.UINTUa}$ ， $\text{IntCntUa}$ ，直到半波峰值大于  $(\text{IntLVL} + \text{IntLVLHys})$ ，退出 INT 状态，置起结束中断标志  $\text{UaINTEndIF} = 1$ ， $\text{IntCntUa} = 7$ ， $\text{IntVpUa} = \text{min2}$ ， $\text{PeakSagIntIovStau.UINTUa} = 0$ 。



注：电压 INT，用户可直接调用钜泉标准库函数实现。

## 22.4.4 电流过流事件检测

根据数据源选择位  $\text{ADCCON.bit14.PeaKsagSrc}$  选择对半波峰值或者半波有效值进行过流判断，选择数据源更新后才进行过流判断。以半周波为单位，每一个半周波进行一次判别，判断是对选择的数据源取绝对值后与过流检测阈值进行比较。

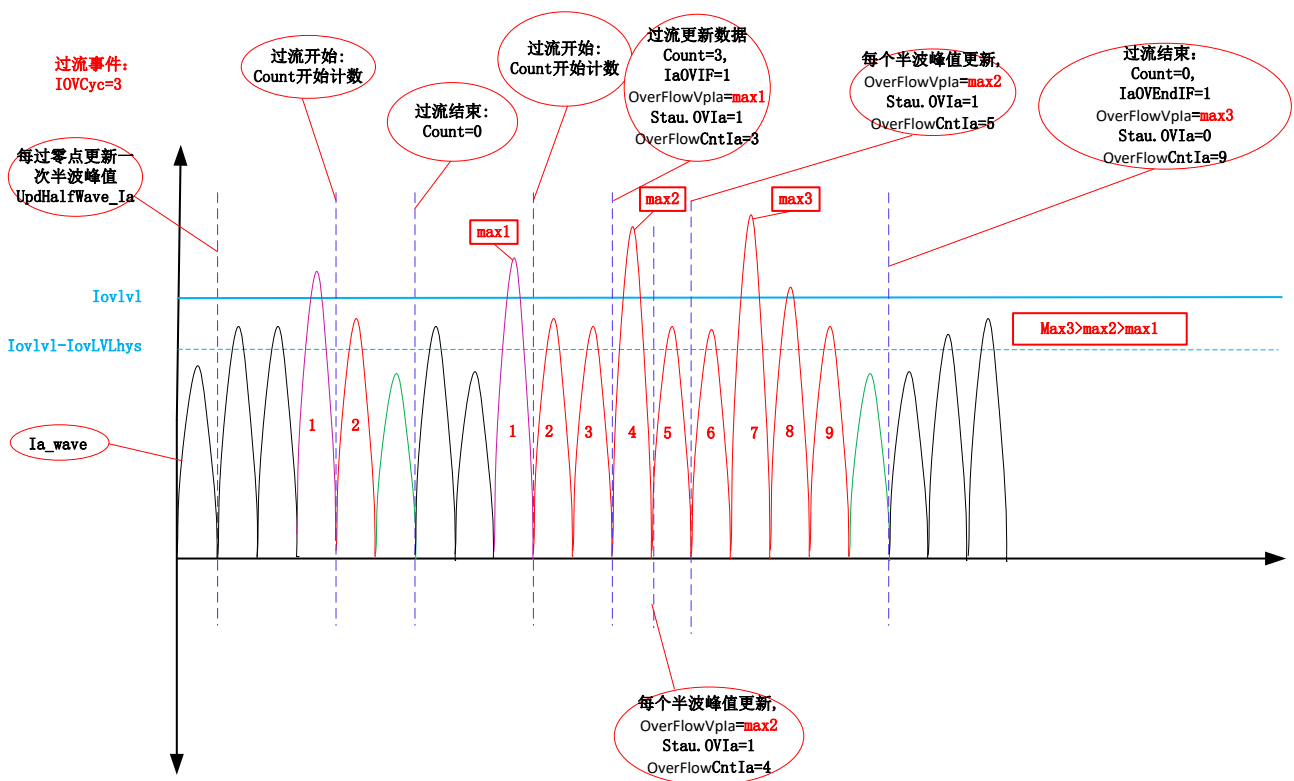
### 过流事件检测描述：

当  $\text{ADCCON.bit10.IovEn} = 0$ ，功能关闭，清状态标志寄存器  $\text{PeakSagIntIovStau.OVIx}$ 、期间最大峰值寄存器  $\text{OverFlowVpIx}$  和期间半周波数  $\text{OverFlowCntIx}$ 。

当  $\text{ADCCON.bit10.IovEn} = 1$ ，功能开启，某路电流第一个半周波检测信号数据源的绝对值大于  $\text{IOVLVL}$ ，

记为过流事件判断开始，只要半周波检测信号的绝对值一直大于等于 $(IOVLVL-IOVLVLHys)$ ，则一直处于过流状态，当过流状态持续的半周波计数值Count大于等于IOVCyc设定的半周波数，则判定发生过流事件，置起IxOVIF中断标志，置起状态标志PeakSagIntIovStau.OVIx = 1，更新期间最大值寄存器OverFlowVpIx和期间半波个数OverFlowCntIx，之后每半个周波持续比较，若半周波检测信号数据源的绝对值大于等于 $(IOVLVL-IOVLVLHys)$ ，则每个半周波更新一次PeakSagIntIovStau.OVIx、OverFlowVpIx和OverFlowCntIx。当半周波检测信号数据源的绝对值小于 $(IOVLVL-IOVLVLHys)$ 时，就退出过流事件，置起IxOVerEndIF退出中断标志，PeakSagIntIovStau.OVIx = 0。过流事件结束后，期间最大峰值寄存器OverFlowVpIx和期间半波个数OverFlowCntIx寄存器保持最后一次更新值不变，直到再一次发生过流事件，持续发生IOVCyc个半周波，更新新的值于期间最大值寄存器OverFlowVpIx和期间半波周数OverFlowCntIx寄存器中。

例如：如下图所示，以Ia通道选择半波峰值为判断数据源，IovEn = 1，当检测Ia半波峰值的绝对值大于IOVLVL，第一次发生过流事件，但半波个数不满足个数大于等于IOVCyc，又检测Ia半波峰值小于 $(IOVLVL-IOVLVLHys)$ ，此时不更新数据，再检测Ia半波峰值的绝对值大于IOVLVL，当持续IOVCyc个半周波都满足过流条件，置起中断标志IaOVIF=1，PeakSagIntIovStau.OVIa = 1，更新期间最大值OverFlowVpIa = max1，OverFlowCntIa = 3，每半个周波持续检测，更新OverFlowVpIa，PeakSagIntIovStau.OVIa，OverFlowCntIa，直到半波峰值小于 $(IOVLVL-IOVLVLHys)$ ，退出过流状态，置起结束中断标志IaOVerEndIF = 1，OverFlowCntIa = 9，OverFlowVpIa = max3，PeakSagIntIovStau.OVIa = 0。



## 22.4.5 反向指示

计量单元同时提供 功率反向标志 和 电能反向标志寄存器。

提供 ABC 相与合相的全波有功、无功功率，基波有功、无功功率和谐波有功功率的状态指示标志 REV\_Power，当检测到负功率时，该信号为 1；当检测到正有功功率时，该信号为 0。

提供 ABC 相与合相的全波有功、无功能量，视在能量，基波有功、无功能量和谐波有功能量的 CFCnt 累计溢出大于 HFConst 的方向 REV\_Energy, CFCnt 累计溢出大于 HFConst 或者小于(-HFConst)的时刻触发比较，并不是 CFCnt 累加的过程状态，当检测到能量大于等于 HFConst 时，该信号为 0；当检测到能量小于等于(-HFConst)时，该信号为 1。

## 22.4.6 失压指示

计量单元提供 A/B/C 相 某相失压的状态检测。即当某相电压有效值小于失压阈值时触发该标志。

ABC 相失压状态指示标志 SignalStatus.VoltageFailA/VoltageFailB/VoltageFailC，当检测到某相电压有效值小于失压阈值(FailVoltage $\ll$ 8)时，该对应相 VoltageFailx 信号为 1，当检测到某相电压有效值大于失压阈值(FailVoltage $\ll$ 8)时，该对应相 VoltageFailx 信号为 0。

## 22.4.7 电压电流错序指示

提供 ABC 相电压、电流错序状态指示标志 IOrder/ UOrder。

错序的定义：

当检测到电压 ABC 相错序时，UOrder 信号为 1，否则为正序 UOrder = 0；当检测到电流 ABC 相错序时，IOrder 信号为 1，否则为正序 IOrder = 0。

## 22.4.8 采样通道互换

为防止用户 A/B/C 三相电压之间、A/B/C/N 电流(含零线电流)之间 接错线。比如 A/B 火线电压接反，A 相电流的进出接反等等。计量单元提供了采样通道互换功能。

支持通道互换、通道数据信号反向功能（通道指的是三路电压通道和四路电流通道）。其中，三路电压之间可以互相切换，四路电流之间可以互相切换，电压和电流通道之间不能互相切换。

另外，计量单元还支持 ANSI 标准下的多种接法。

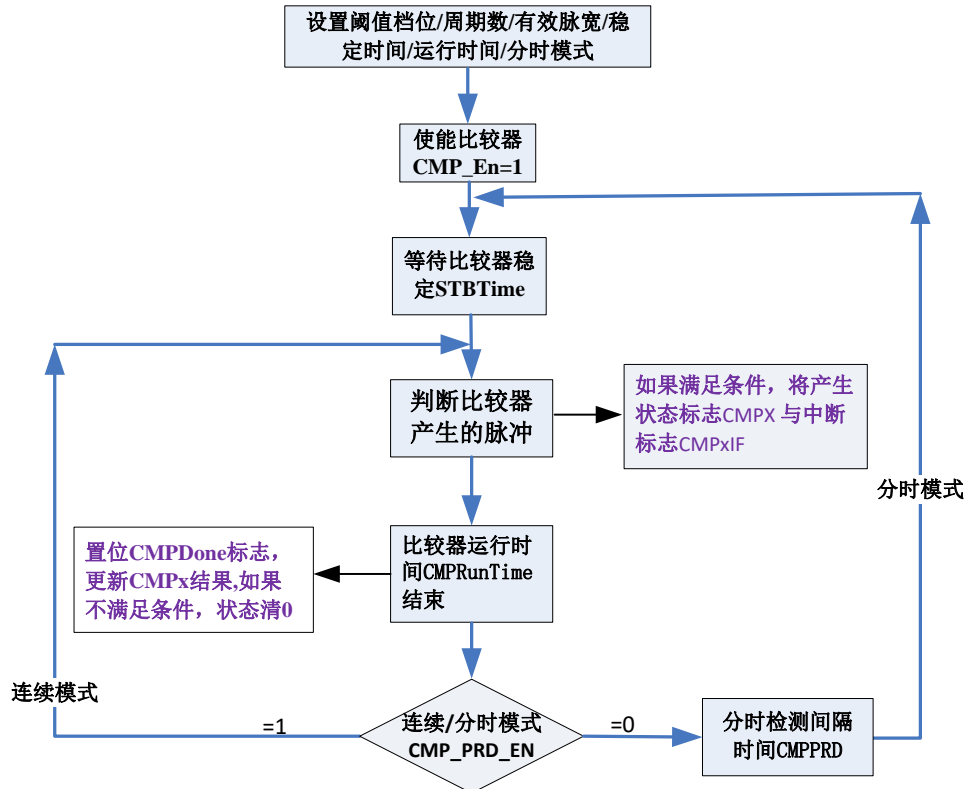
## 22.4.9 全失压比较器设计

全失压的定义：

当 ABC 三相电压都低于临界电压的情况下，如果此时检测到任意一相的负荷电流大于 5% 额定电流，就认为进入了全失压状态。由于三相电压尽失，此时电表会进入电池等的辅助供电模式，对检测电流有效值的整体功耗会有严格的要求。

当检测到任意一相电压大于等于临界电压，或者检测的所有相负荷电流都不再大于 5% 额定电流，全失压状态解除。

HT762X 芯片提供全失压硬件比较器并完整的检测状态机，用于连续/分时检测 ABC 三相的负荷电流。



全失压比较器分时模式下分时开启时间通过  $CMP\_PRD[7:0]$  可设，范围：1s~250s；比较器开启后运行时间通过  $CMPRunTime[7:0]$  也可设，范围：39ms~780ms，运行结束产生比较完成标志  $CMPDone$ 。

全失压比较器开启需要一段建立时间，以满足 LDO/Vref/滤波器等的稳定时间。分时模式每次开启都需要时间稳定，(连续模式只有第一次开启需要时间稳定) 稳定时间通过  $STBTime[5:0]$  可设，范围：0ms~63ms。

全失压比较器阈值可通过  $VcmpLVL[2:0]$  进行选择档位，也可通过寄存器进行有效脉宽设置；另外可通过  $VOLNUM[3:0]$  设定比较器连续检测的正弦波周期数，只有两个有效脉宽间隔不大于 35ms 时，才认为脉宽连续，因此电流信号与比较器比较产生的脉宽宽度和连续脉宽数满足设定的脉宽和周期数时，就可认为处于全失压状态。

HT762X 提供 A\B\C 三相的全失压状态标志以及中断标志，当进入全失状态后，Hold 模式下可唤醒 MCU。

详细功能与相关配置、数据，需参照该功能单独文档。

## 22.4.10 互感器二次侧检测

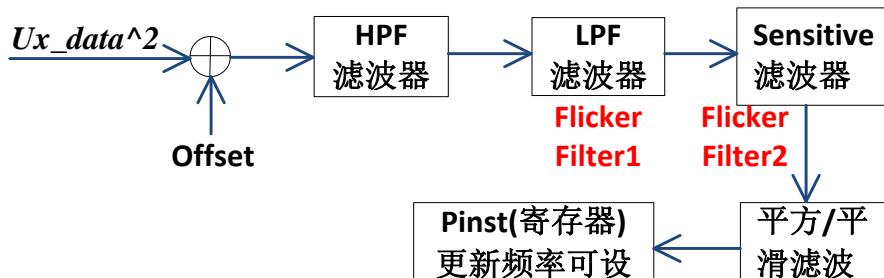
详见互感器二次侧检测相关文档

## 22.4.11 负载电流快速变化

详见负载电流快速变化相关文档

## 22.4.12 闪变

电压闪变是指人眼对灯光亮度变化所引起的刺激的不稳定感，即人眼灯光亮度变化的不适感。EMU 计量单元集成了完整的硬件闪变的设计。



闪变功能所需的滤波器 Filter1 和 Filter2 需要和基波谐波功能、CT 二次侧检测功能复用滤波器。详见 **多功能滤波器** 章节

**注：闪变完整方案，用户可直接调用钜泉标准库函数实现。**

## 22.4.13 三相不平衡度

三相电压在幅值上不同或者相位差不是  $120^\circ$ ，或者两者兼有，定义为电压不平衡。其三相电力系统中三相不平衡的程度，定为不平衡度。

不平衡度的分析方法为电压、电流负序基波分量或零序基波分量与正序基波分量的方均根值百分比表示。

EMU 单元的工频同步系统，直接提供了 7 路 ADC 信号的基波实部/虚部信息，可供用户按照对称分量法的原理计算不平衡度。用户也可以通过工频同步系统  $\rightarrow$  FFT 路径，对 10 周期时间间隔连续对基波成分进行相关分析。另外，EMU 单元也提供了全波线电压有效值 / 基波电压有效值，全波/基波 角度寄存器 供用户分析。

**注：不平衡度，用户可直接调用钜泉标准库函数实现。**

## 22.4.14 谐波/间谐波分析

EMU 单元提供如下几种谐波分析方法：

### 1. 谐波滤波器

通过基波谐波功能的使能开关，能获取到总谐波的功率/有效值等寄存器参数，并可接入电能通路以计算总谐波电能。

不过谐波滤波器并不能区分分次谐波，因此谐波滤波器适合做总谐波相关参数。

### 2. 工频同步系统—FFT

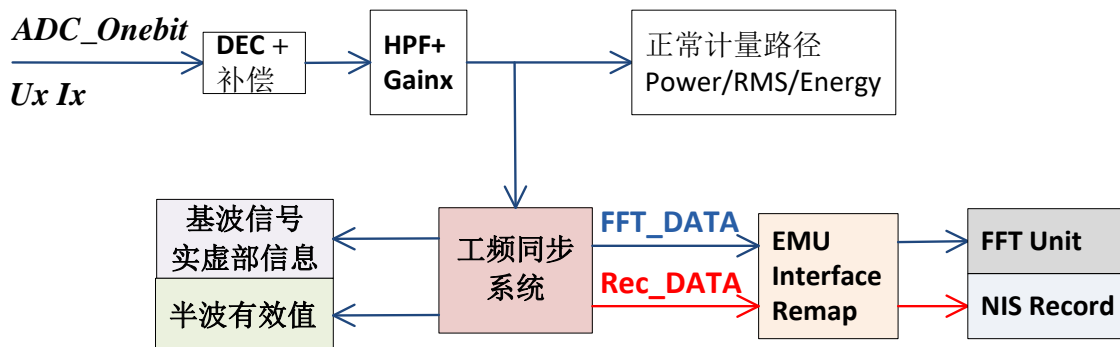
HT762X 集成了硬件 FFT 功能，用户可将工频同步系统实时输出的 FFT\_DATA 接入 FFT 模块，即可从寄存器获取到 FFT 的计算结果（复数形式）。对 FFT 的结果进行处理即可分析出各次谐波/间谐波的幅值/相位等信息。然后以常数自定义功率的形式接入到电能通路做电能计量，即可完成谐波/间谐波的数据参数、精度分析、分次谐波电能计量。

注：谐波/间谐波，用户可直接调用钜泉标准库函数实现。

## 22.5 EMU 数据接口

### 22.5.1 工频同步系统简介

计量单元提供采样信号的基于锁相环算法的工频同步系统，能够精确锁定采样信号频率，并将 7 路 ADC 采样原始信号自动转换成  $2^N$ /周波 的工频同步采样数据。



工频同步系统输出如下四组数据：

1. 提取出 7 路 ADC 采样数据的基波信号实部与虚部信号寄存器  
FundValue\_UxReal & FundValue\_UxImage (x = A/B/C);  
FundValue\_IxReal & FundValue\_IxImage (x = A/B/C/N);
2. 7 路 ADC 信号的半波有效值，更新周期为 10mS 一次(50Hz 系统)  
HpRMS\_Ux (x = A/B/C) & HpRMS\_Ix (x = A/B/C/N)  
为进一步提供半波有效值精度，工频同步系统提供了寄存器配置为半波更新的全波有效值。
3. 给 FFT 路径输出的四种数据  
128 点/周波，256 点/周波，1024 点/10 周波(用于 50Hz 系统)和 1024 点/12 周波(用于 60Hz 系统)
4. 给 Record 路径输出的四种数据  
32 点/周波，64 点/周波，128 点/周波，256 点/周波

工频同步系统的应用注意事项：

1. 必须至少开启一个电压通道即 ADCUa | Ub | Uc 至少开启一个，否则将无法完成锁频。
2. 系统提供可以加速系统收敛的相关寄存器。但是收敛越快，系统稳定性也会下降。目前寄存器提供的初值已经优化调试，不需要再另行配置，使用默认值让系统完成自动收敛即可。

### 22.5.2 EMU-DMA 设计

#### 1. 通用 DMA :

EMU 单元为通用 DMA 功能提供三组数据源：

- (1) 采样波形数据，SPL\_Ua~In，地址 000~018H

---

(2) 瞬时有效值的平方数据, QuartRMSIns, 地址 2F8~310H

(3) 闪变数据, Pinst, 地址 3E4H ~ 3ECH

为节省 DMA 通道资源, 设计为上述 1 和 2 的 DMA Request MUX 关系, 选择共用 7 个 DMA 通道, 通过 Bit.DMASrc\_Sel 选择是 1 或 2 的 Request, Default 为 SPL。

**注意:** SPL 波形寄存器的更新频率受 SPL[2:0]的影响。QuartRMSIns 的 Request 频率为 EMUCLK/OSR

## 2. 工频同步系统 DMA:

ADC 采样数据经过 EMU 单元的工频同步系统后, 经 EMUInterface 模块进行 Channel Remap, 并经专用 DMA 送至 FFT 模块进行频域分析。

该部分 DMA 设计可参考 EMU Interface 与 FFT 模块章节。

## 22.6 计量参数寄存器列表

EMU 模块计量寄存器基地址: 0x 4001 B000 全部为 ReadOnly					
计量寄存器: 0x4001B000					
偏移地址	名称	读写方式	有效字长	复位值	功能描述
000H	SPL_Ua	RO	22bit	--	A 相电压通道 ADC 采样数据 SPL 格式高对齐低补 0 Bit[23]符号 + Bit[22:2]数据 + Bit[1:0]补 0
004H	SPL_Ub	RO	22bit	--	B 相电压通道 ADC 采样数据
008H	SPL_Uc	RO	22bit	--	C 相电压通道 ADC 采样数据
00CH	SPL_Ia	RO	22bit	--	A 相电流通道 ADC 采样数据
010H	SPL_Ib	RO	22bit	--	B 相电流通道 ADC 采样数据
014H	SPL_Ic	RO	22bit	--	C 相电流通道 ADC 采样数据
018H	SPL_In	RO	22bit	--	零线电流通道 ADC 采样数据
01CH	All_Power_Pa	RO	32bit	--	A 相全波有功功率 Q1.31 Bit0-Bit30 数据, Bit31 为符号位
020H	All_Power_Pb	RO	32bit	--	B 相全波有功功率 Q1.31
024H	All_Power_Pc	RO	32bit	--	C 相全波有功功率 Q1.31
028H	All_Power_Pt	RO	32bit	--	合相全波有功功率 Q1.31
02CH	All_Power_Qa	RO	32bit	--	A 相全波无功功率 Q1.31
030H	All_Power_Qb	RO	32bit	--	B 相全波无功功率 Q1.31
034H	All_Power_Qc	RO	32bit	--	C 相全波无功功率 Q1.31
038H	All_Power_Qt	RO	32bit	--	合相全波无功功率 Q1.31
03CH	All_Power_Sa	RO	32bit	--	A 相全波视在功率 Q1.31
040H	All_Power_Sb	RO	32bit	--	B 相全波视在功率 Q1.31
044H	All_Power_Sc	RO	32bit	--	C 相全波视在功率 Q1.31
048H	All_Power_St	RO	32bit	--	合相全波视在功率 Q1.31
04CH	All_RMS_Ua	RO	23bit	--	A 相全波电压有效值 Q0.23 所有 RMS 寄存器 Bit0-Bit22 数据, Bit23 恒为 0
050H	All_RMS_Ub	RO	23bit	--	B 相全波电压有效值 Q0.23
054H	All_RMS_Uc	RO	23bit	--	C 相全波电压有效值 Q0.23
058H	All_RMS_Ia	RO	23bit	--	A 相全波电流有效值 Q0.23
05CH	All_RMS_Ib	RO	23bit	--	B 相全波电流有效值 Q0.23
060H	All_RMS_Ic	RO	23bit	--	C 相全波电流有效值 Q0.23
064H	All_RMS_In	RO	23bit	--	零线全波电流有效值 Q0.23 注: AllRMSIn 可以复用为 CT 二次侧检测功能的结果存储 CTdetEn=1 时, 该寄存器复用为 CT 二次侧

					的数据。格式为 22Bit 低对齐存放
068H	RMS_Ut	RO	23bit	--	电压矢量和有效值-全波/基波共用可选 <b>Q0.23</b>
06CH	RMS_It	RO	23bit	--	电流矢量和有效值-全波/基波共用可选 <b>Q0.23</b>
070H	RMS_UAB	RO	23bit	--	AB 线电压有效值-全波/基波共用可选 <b>Q0.23</b>
074H	RMS_UBC	RO	23bit	--	BC 线电压有效值-全波/基波共用可选 <b>Q0.23</b>
078H	RMS_UAC	RO	23bit	--	AC 线电压有效值-全波/基波共用可选 <b>Q0.23</b>
07CH	FreqUa	RO	23bit	0x7FFFFFFF	A 相电压频率 <b>Q0.23</b>
080H	FreqUb	RO	23bit	0x7FFFFFFF	B 相电压频率 <b>Q0.23</b>
084H	FreqUc	RO	23bit	0x7FFFFFFF	C 相电压频率 <b>Q0.23</b>
088H	FreqU	RO	23bit	0x7FFFFFFF	合相电压频率 <b>Q0.23</b>
08CH	All_PFa	RO	24	--	A 相全波功率因数 <b>Q1.23</b>
090H	All_PFb	RO	24	--	B 相全波功率因数 <b>Q1.23</b>
094H	All_PFc	RO	24	--	C 相全波功率因数 <b>Q1.23</b>
098H	All_PFt	RO	24	--	合相全波功率因数 <b>Q1.23</b>
09CH	PhaseAngle_Ua	RO	24bit	--	A 相电压角度-全波/基波共用可选
0A0H	PhaseAngle_Ub	RO	24bit	--	B 相电压角度-全波/基波共用可选
0A4H	PhaseAngle_Uc	RO	24bit	--	C 相电压角度-全波/基波共用可选
0A8H	PhaseAngle_Ia	RO	24bit	--	A 相电流角度-全波/基波共用可选
0ACH	PhaseAngle_Ib	RO	24bit	--	B 相电流角度-全波/基波共用可选
0B0H	PhaseAngle_Ic	RO	24bit	--	C 相电流角度-全波/基波共用可选
0B4H	PhaseAngle_In	RO	24bit	--	零线电流角度-全波/基波共用可选
0B8H	All_EnergyPos_Pa	RO	24	--	A 相全波正向有功电能 <b>Q0.24</b>
0BCH	All_EnergyPos_Pb	RO	24	--	B 相全波正向有功电能 <b>Q0.24</b>
0C0H	All_EnergyPos_Pc	RO	24	--	C 相全波正向有功电能 <b>Q0.24</b>
0C4H	All_EnergyPos_Pt	RO	24	--	合相全波正向有功电能 <b>Q0.24</b>
0C8H	All_EnergyPos_Qa	RO	24	--	A 相全波正向无功电能 <b>Q0.24</b>
0CCH	All_EnergyPos_Qb	RO	24	--	B 相全波正向无功电能 <b>Q0.24</b>
0D0H	All_EnergyPos_Qc	RO	24	--	C 相全波正向无功电能 <b>Q0.24</b>
0D4H	All_EnergyPos_Qt	RO	24	--	合相全波正向无功电能 <b>Q0.24</b>
0D8H	Energy_Sa	RO	24	--	A 相视在电能-全波/基波共用可选 <b>Q0.24</b>
0DCH	Energy_Sb	RO	24	--	B 相视在电能-全波/基波共用可选 <b>Q0.24</b>
0E0H	Energy_Sc	RO	24	--	C 相视在电能-全波/基波共用可选 <b>Q0.24</b>
0E4H	Energy_St	RO	24	--	合相视在电能-全波/基波共用可选 <b>Q0.24</b>
0E8H	All_CFCntPos_Pa	RO	20	--	A 相全波正向有功快速脉冲计数 格式为: Bit0-Bit19 数据, Bit20-23 补充为符号位
0ECH	All_CFCntPos_Pb	RO	20	--	B 相全波正向有功快速脉冲计数
0F0H	All_CFCntPos_Pc	RO	20	--	C 相全波正向有功快速脉冲计数
0F4H	All_CFCntPos_Pt	RO	20	--	合相全波正向有功快速脉冲计数
0F8H	All_CFCntPos_Qa	RO	20	--	A 相全波正向无功快速脉冲计数

0FCH	All_CFCntPos_Qb	RO	20	--	B 相全波正向无功快速脉冲计数
100H	All_CFCntPos_Qc	RO	20	--	C 相全波正向无功快速脉冲计数
104H	All_CFCntPos_Qt	RO	20	--	合相全波正向无功快速脉冲计数
108H	CFCntPos_Sa	RO	20	--	A 相视在快速脉冲计数（视在无反） 全波/基波共用可选
10CH	CFCntPos_Sb	RO	20	--	B 相视在快速脉冲计数（视在无反） 全波/基波共用可选
110H	CFCntPos_Sc	RO	20	--	C 相视在快速脉冲计数（视在无反） 全波/基波共用可选
114H	CFCntPos_St	RO	20	--	合相视在快速脉冲计数（视在无反） 全波/基波共用可选
118H	All_EnergyNeg_Pa	RO	24	--	A 相全波反向有功电能 <b>Q0.24</b>
11CH	All_EnergyNeg_Pb	RO	24	--	B 相全波反向有功电能 <b>Q0.24</b>
120H	All_EnergyNeg_Pc	RO	24	--	C 相全波反向有功电能 <b>Q0.24</b>
124H	All_EnergyNeg_Pt	RO	24	--	合相全波反向有功电能 <b>Q0.24</b>
128H	All_EnergyNeg_Qa	RO	24	--	A 相全波反向无功电能 <b>Q0.24</b>
12CH	All_EnergyNeg_Qb	RO	24	--	B 相全波反向无功电能 <b>Q0.24</b>
130H	All_EnergyNeg_Qc	RO	24	--	C 相全波反向无功电能 <b>Q0.24</b>
134H	All_EnergyNeg_Qt	RO	24	--	合相全波反向无功电能 <b>Q0.24</b>
138H	All_CFCntNeg_Pa	RO	20	--	A 相全波反向有功快速脉冲计数
13CH	All_CFCntNeg_Pb	RO	20	--	B 相全波反向有功快速脉冲计数
140H	All_CFCntNeg_Pc	RO	20	--	C 相全波反向有功快速脉冲计数
144H	All_CFCntNeg_Pt	RO	20	--	合相全波反向有功快速脉冲计数
148H	All_CFCntNeg_Qa	RO	20	--	A 相全波反向无功快速脉冲计数
14CH	All_CFCntNeg_Qb	RO	20	--	B 相全波反向无功快速脉冲计数
150H	All_CFCntNeg_Qc	RO	20	--	C 相全波反向无功快速脉冲计数
154H	All_CFCntNeg_Qt	RO	20	--	合相全波反向无功快速脉冲计数
158H	EnergyNeg_SaConst	RO	24	--	A 相视在通路的反向电能 <b>Q0.24</b> 注:视在不存在负数, 因此这个功率用途只有使能常数功率并且为负
15CH	EnergyNeg_SbConst	RO	24	--	B 相视在通路的反向电能 <b>Q0.24</b>
160H	EnergyNeg_ScConst	RO	24	--	C 相视在通路的反向电能 <b>Q0.24</b>
164H	EnergyNeg_StConst	RO	24	--	合相视在通路的反向电能 <b>Q0.24</b>
168H	CFCntNeg_Sa	RO	20	--	A 相视在通路的反向快速脉冲计数 注:视在不存在负数, 因此这个功率用途只有使能常数功率并且为负
16CH	CFCntNeg_Sb	RO	20	--	B 相视在通路的反向快速脉冲计数
170H	CFCntNeg_Sc	RO	20	--	C 相视在通路的反向快速脉冲计数
174H	CFCntNeg_St	RO	20	--	合相视在通路的反向快速脉冲计数
178H	Fund_Power_Pa	RO	32bit	--	A 相基波有功功率 <b>Q1.31</b>
17CH	Fund_Power_Pb	RO	32bit	--	B 相基波有功功率 <b>Q1.31</b>
180H	Fund_Power_Pc	RO	32bit	--	C 相基波有功功率 <b>Q1.31</b>
184H	Fund_Power_Pt	RO	32bit	--	合相基波有功功率 <b>Q1.31</b>

188H	Fund_Power_Qa	RO	32bit	--	A 相基波无功功率 <b>Q1.31</b>
18CH	Fund_Power_Qb	RO	32bit	--	B 相基波无功功率 <b>Q1.31</b>
190H	Fund_Power_Qc	RO	32bit	--	C 相基波无功功率 <b>Q1.31</b>
194H	Fund_Power_Qt	RO	32bit	--	合相基波无功功率 <b>Q1.31</b>
198H	Fund_Power_Sa	RO	32bit	--	A 相基波视在功率 <b>Q1.31</b>
19CH	Fund_Power_Sb	RO	32bit	--	B 相基波视在功率 <b>Q1.31</b>
1A0H	Fund_Power_Sc	RO	32bit	--	C 相基波视在功率 <b>Q1.31</b>
1A4H	Fund_Power_St	RO	32bit	--	合相基波视在功率 <b>Q1.31</b>
1A8H	Fund_RMS_Ua	RO	23bit	--	A 相基波电压有效值 <b>Q0.23</b>
1ACH	Fund_RMS_Ub	RO	23bit	--	B 相基波电压有效值 <b>Q0.23</b>
1B0H	Fund_RMS_Uc	RO	23bit	--	C 相基波电压有效值 <b>Q0.23</b>
1B4H	Fund_RMS_Ia	RO	23bit	--	A 相基波电流有效值 <b>Q0.23</b>
1B8H	Fund_RMS_Ib	RO	23bit	--	B 相基波电流有效值 <b>Q0.23</b>
1BCH	Fund_RMS_Ic	RO	23bit	--	C 相基波电流有效值 <b>Q0.23</b>
1C0H	Fund_RMS_In	RO	23bit	--	零线基波电流有效值 <b>Q0.23</b>
1C4H	Fund_PFa	RO	24		A 相基波功率因数 <b>Q1.23</b>
1C8H	Fund_PFb	RO	24		B 相基波功率因数 <b>Q1.23</b>
1CCH	Fund_PFc	RO	24		C 相基波功率因数 <b>Q1.23</b>
1D0H	Fund_PFt	RO	24		合相基波功率因数 <b>Q1.23</b>
1D4H	Fund_EnergyPos_Pa	RO	24	--	A 相基波正向有功电能 <b>Q0.24</b>
1D8H	Fund_EnergyPos_Pb	RO	24	--	B 相基波正向有功电能 <b>Q0.24</b>
1DCH	Fund_EnergyPos_Pc	RO	24	--	C 相基波正向有功电能 <b>Q0.24</b>
1E0H	Fund_EnergyPos_Pt	RO	24	--	合相基波正向有功电能 <b>Q0.24</b>
1E4H	Fund_EnergyPos_Qa	RO	24	--	A 相基波正向无功电能 <b>Q0.24</b>
1E8H	Fund_EnergyPos_Qb	RO	24	--	B 相基波正向无功电能 <b>Q0.24</b>
1ECH	Fund_EnergyPos_Qc	RO	24	--	C 相基波正向无功电能 <b>Q0.24</b>
1F0H	Fund_EnergyPos_Qt	RO	24	--	合相基波正向无功电能 <b>Q0.24</b>
1F4H	Fund_CFCntPos_Pa	RO	20	--	A 相基波正向有功快速脉冲计数
1F8H	Fund_CFCntPos_Pb	RO	20	--	B 相基波正向有功快速脉冲计数
1FCH	Fund_CFCntPos_Pc	RO	20	--	C 相基波正向有功快速脉冲计数
200H	Fund_CFCntPos_Pt	RO	20	--	合相基波正向有功快速脉冲计数
204H	Fund_CFCntPos_Qa	RO	20	--	A 相基波正向无功快速脉冲计数
208H	Fund_CFCntPos_Qb	RO	20	--	B 相基波正向无功快速脉冲计数
20CH	Fund_CFCntPos_Qc	RO	20	--	C 相基波正向无功快速脉冲计数
210H	Fund_CFCntPos_Qt	RO	20	--	合相基波正向无功快速脉冲计数
214H	Fund_EnergyNeg_Pa	RO	24	--	A 相基波反向有功电能 <b>Q0.24</b>
218H	Fund_EnergyNeg_Pb	RO	24	--	B 相基波反向有功电能 <b>Q0.24</b>
21CH	Fund_EnergyNeg_Pc	RO	24	--	C 相基波反向有功电能 <b>Q0.24</b>
220H	Fund_EnergyNeg_Pt	RO	24	--	合相基波反向有功电能 <b>Q0.24</b>
224H	Fund_EnergyNeg_Qa	RO	24	--	A 相基波反向无功电能 <b>Q0.24</b>
228H	Fund_EnergyNeg_Qb	RO	24	--	B 相基波反向无功电能 <b>Q0.24</b>
22CH	Fund_EnergyNeg_Qc	RO	24	--	C 相基波反向无功电能 <b>Q0.24</b>

230H	Fund_EnergyNeg_Qt	RO	24	--	合相基波反向无功电能 <b>Q0.24</b>
234H	Fund_CFCntNeg_Pa	RO	20	--	A 相基波反向有功快速脉冲计数
238H	Fund_CFCntNeg_Pb	RO	20	--	B 相基波反向有功快速脉冲计数
23CH	Fund_CFCntNeg_Pc	RO	20	--	C 相基波反向有功快速脉冲计数
240H	Fund_CFCntNeg_Pt	RO	20	--	合相基波反向有功快速脉冲计数
244H	Fund_CFCntNeg_Qa	RO	20	--	A 相基波反向无功快速脉冲计数
248H	Fund_CFCntNeg_Qb	RO	20	--	B 相基波反向无功快速脉冲计数
24CH	Fund_CFCntNeg_Qc	RO	20	--	C 相基波反向无功快速脉冲计数
250H	Fund_CFCntNeg_Qt	RO	20	--	合相基波反向无功快速脉冲计数
254H	Har_Power_Pa	RO	32	--	A 相谐波有功功率 Q1.31
258H	Har_Power_Pb	RO	32	--	B 相谐波有功功率 Q1.31
25CH	Har_Power_Pc	RO	32	--	C 相谐波有功功率 Q1.31
260H	Har_Power_Pt	RO	32	--	合相谐波有功功率 Q1.31
264H	Har_RMS_Ua	RO	23	--	A 相谐波电压有效值 <b>Q0.23</b>
268H	Har_RMS_Ub	RO	23	--	B 相谐波电压有效值 <b>Q0.23</b>
26CH	Har_RMS_Uc	RO	23	--	C 相谐波电压有效值 <b>Q0.23</b>
270H	Har_RMS_Ia	RO	23	--	A 相谐波电流有效值 <b>Q0.23</b>
274H	Har_RMS_Ib	RO	23	--	B 相谐波电流有效值 <b>Q0.23</b>
278H	Har_RMS_Ic	RO	23	--	C 相谐波电流有效值 <b>Q0.23</b>
27CH	Har_RMS_In	RO	23	--	零线谐波电流有效值 <b>Q0.23</b>
280H	Har_EnergyPos_Pa	RO	24	--	A 相谐波正向有功电能 <b>Q0.24</b>
284H	Har_EnergyPos_Pb	RO	24	--	B 相谐波正向有功电能 <b>Q0.24</b>
288H	Har_EnergyPos_Pc	RO	24	--	C 相谐波正向有功电能 <b>Q0.24</b>
28CH	Har_EnergyPos_Pt	RO	24	--	合相谐波正向有功电能 <b>Q0.24</b>
290H	Har_CFCntPos_Pa	RO	20	--	A 相谐波正向有功快速脉冲计数
294H	Har_CFCntPos_Pb	RO	20	--	B 相谐波正向有功快速脉冲计数
298H	Har_CFCntPos_Pc	RO	20	--	C 相谐波正向有功快速脉冲计数
29CH	Har_CFCntPos_Pt	RO	20	--	合相谐波正向有功快速脉冲计数
2A0H	Har_EnergyNeg_Pa	RO	24	--	A 相谐波反向有功电能 <b>Q0.24</b>
2A4H	Har_EnergyNeg_Pb	RO	24	--	B 相谐波反向有功电能 <b>Q0.24</b>
2A8H	Har_EnergyNeg_Pc	RO	24	--	C 相谐波反向有功电能 <b>Q0.24</b>
2ACH	Har_EnergyNeg_Pt	RO	24	--	合相谐波反向有功电能 <b>Q0.24</b>
2B0H	Har_CFCntNeg_Pa	RO	20	--	A 相谐波反向有功快速脉冲计数
2B4H	Har_CFCntNeg_Pb	RO	20	--	B 相谐波反向有功快速脉冲计数
2B8H	Har_CFCntNeg_Pc	RO	20	--	C 相谐波反向有功快速脉冲计数
2BCH	Har_CFCntNeg_Pt	RO	20	--	合相谐波反向有功快速脉冲计数
2C0H	DCVar_Ua	RO	22Bit	--	A 相电压通道 ADC 采样直流数据 <b>DCVar</b> 的格式为 <b>Q3.21</b> Bit0-Bit20 数据, Bit21 为符号位, Bit22~23 补充为符号位
2C4H	DCVar_Ub	RO	22Bit	--	B 相电压通道 ADC 采样直流数据
2C8H	DCVar_Uc	RO	22Bit	--	C 相电压通道 ADC 采样直流数据

2CCH	DCVar_Ia	RO	22Bit	--	A 相电流通道 ADC 采样直流数据
2D0H	DCVar_Ib	RO	22Bit	--	B 相电流通道 ADC 采样直流数据
2D4H	DCVar_Ic	RO	22Bit	--	C 相电流通道 ADC 采样直流数据
2D8H	DCVar_In	RO	22Bit	--	零线电流通道 ADC 采样直流数据
2DCH	UpdHalfWave_Ua	RO	22Bit	--	A 相电压通道半波峰值 Bit0-Bit20 数据, Bit21 为符号位, Bit22~23 补充为符号位
2E0H	UpdHalfWave_Ub	RO	22Bit	--	B 相电压通道半波峰值
2E4H	UpdHalfWave_Uc	RO	22Bit	--	C 相电压通道半波峰值
2E8H	UpdHalfWave_Ia	RO	22Bit	--	A 相电流通道半波峰值
2ECH	UpdHalfWave_Ib	RO	22Bit	--	B 相电流通道半波峰值
2F0H	UpdHalfWave_Ic	RO	22Bit	--	C 相电流通道半波峰值
2F4H	UpdHalfWave_In	RO	22Bit	--	零线电流通道半波峰值
2F8H	QuartRmsIns_Ua	RO	32	--	Ua 通道瞬时有效值平方
2FCH	QuartRmsIns_Ub	RO	32	--	Ub 通道瞬时有效值平方
300H	QuartRmsIns_Uc	RO	32	--	Uc 通道瞬时有效值平方
304H	QuartRmsIns_Ia	RO	32	--	Ia 通道瞬时有效值平方
308H	QuartRmsIns_Ib	RO	32	--	Ib 通道瞬时有效值平方
30CH	QuartRmsIns_Ic	RO	32	--	Ic 通道瞬时有效值平方
310H	QuartRmsIns_In	RO	32	--	In 通道瞬时有效值平方
314H	PeakVpUa	RO	22		若 Peak 事件判定数据源选择半波峰值, 则 Peak 结束后写入的是 A 相电压最大峰值。 (最大半波峰值的 Bit23/22 补符号位) (若数据源选择半波有效值, 则事件结束后写入的是 A 相电压最大半波有效值)同理 Sag/Int/OV 数据源为半波峰值: Bit0-Bit20 数据, Bit21~23 补充为符号位 数据源为半波有效值: Bit0-Bit22 数据, Bit23 恒为 0
318H	PeakVpUb	RO	22		Peak 结束后 B 相电压最大峰值
31CH	PeakVpUc	RO	22		Peak 结束后 C 相电压最大峰值
320H	PeakCntUa	RO	24		peak 结束后 A 相持续半周数 Q0.24
324H	PeakCntUb	RO	24		peak 结束后 B 相持续半周数 Q0.24
328H	PeakCntUc	RO	24		peak 结束后 C 相持续半周数 Q0.24
32CH	SagVpUa	RO	22		SAG 结束后 A 相电压最小峰值
330H	SagVpUb	RO	22		SAG 结束后 B 相电压最小峰值
334H	SagVpUc	RO	22		SAG 结束后 C 相电压最小峰值
338H	SagCntUa	RO	24		sag 结束后 A 相持续半周数 Q0.24
33CH	SagCntUb	RO	24		sag 结束后 B 相持续半周数 Q0.24
340H	SagCntUc	RO	24		sag 结束后 C 相持续半周数 Q0.24
344H	IntVpUa	RO	22		INT 结束后 A 相电压最小峰值
348H	IntVpUb	RO	22		INT 结束后 B 相电压最小峰值

34CH	IntVpUc	RO	22		INT 结束后 C 相电压最小峰值
350H	IntCntUa	RO	24		INT 结束后 A 相持续半周数 Q0.24
354H	IntCntUb	RO	24		INT 结束后 B 相持续半周数 Q0.24
358H	IntCntUc	RO	24		INT 结束后 C 相持续半周数 Q0.24
35CH	OverFlowVpIa	RO	22		电流过流结束后 A 相大峰值
360H	OverFlowVpIb	RO	22		电流过流结束后 B 相最大峰值
364H	OverFlowVpIc	RO	22		电流过流结束后 C 相最大峰值
368H	OverFlowVpIn	RO	22		电流过流结束后零线最大峰值
36CH	OverFlowCntIa	RO	24		电流过流结束后 A 相持续半周数 Q0.24
370H	OverFlowCntIb	RO	24		电流过流结束后 B 相持续半周数 Q0.24
374H	OverFlowCntIc	RO	24		电流过流结束后 C 相持续半周数 Q0.24
378H	OverFlowCntIn	RO	24		电流过流结束后零线持续半周数 Q0.24
37CH	AutoTgain_U	RO	16		U 采样通道的温度增益(自动补偿开启后计算的 Tgain 增益值)
380H	AutoTgain_I	RO	16		I 采样通道的温度增益(自动补偿开启后计算的 Tgain 增益值)
384H	AutoTgain_In	RO	16		In 采样通道的温度增益(自动补偿开启后计算的 Tgain 增益值)
388H	HpRMS_Reg_Ua	R	23		半波有效值数据,来自工频同步系统 HpRMS 的格式为 Q0.23, 和 All_RMS 的数据格式是一样的。
38CH	HpRMS_Reg_Ub	R	23		
390H	HpRMS_Reg_Uc	R	23		
394H	HpRMS_Reg_Ia	R	23		
398H	HpRMS_Reg_Ib	R	23		
39CH	HpRMS_Reg_Ic	R	23		
3A0H	HpRMS_Reg_In	R	23		
3A4H	DFT_Value_UaReal	R	24		<b>DFT 结果, Real &amp; image</b> <b>DFT 呈现结果为 3Byte 带符号, 还原公式:</b> <b>Real/2<sup>22</sup>(负数则 (Real-2<sup>24</sup>)/2<sup>22</sup>)取模取</b> <b>相位, 注意是除以 2<sup>22</sup></b>
3A8H	DFT_Value_UaImage	R	24		
3ACH	DFT_Value_UbReal	R	24		
3B0H	DFT_Value_UbImage	R	24		
3B4H	DFT_Value_UcReal	R	24		
3B8H	DFT_Value_UcImage	R	24		
3BCH	DFT_Value_IaReal	R	24		
3C0H	DFT_Value_IaImage	R	24		
3C4H	DFT_Value_IbReal	R	24		
3C8H	DFT_Value_IbImage	R	24		
3CCH	DFT_Value_IcReal	R	24		
3D0H	DFT_Value_IcImage	R	24		
3D4H	DFT_Value_InReal	R	24		
3D8H	DFT_Value_InImage	R	24		

3DCH	Reserved				
3E0H	Reserved				
3E4H	FlickerUaPinst	R	32		A 相电压通道闪变
3E8H	FlickerUbPinst	R	32		B 相电压通道闪变
3ECH	FlickerUcPinst	R	32		C 相电压通道闪变
3F0H	<b>Checksum1</b>	RO	32	<b>0x1CDEB1E</b>	32Bit 无符号, 参与该校验的校表参数寄存器相加。范围: Base0x4001C000 Offset: <b>000~104H,加入 7 个 ADCOffset</b> 110H~1C0H,250H~2D4H,2E4H~348H,358H,36CH,370H
3F4H	<b>Checksum2</b>	RO	32	<b>0x1DF3F13A</b>	32Bit 无符号, 参与该校验的校表参数寄存器相加。范围为各类滤波器。客户可暂不使用。
3F8H	<b>Checksum3</b>	RO	32	<b>0x0037DA40</b>	32Bit 无符号, EWU/工频同步系统功能模块的配置参数校验和, 范围: A08H~A3CH, 工频同步系统不参与校验和
3FCH	<b>Reserved</b>	RO	32		
400H	Noload	<b>RO</b>	24		启动潜动状态
404H	REV_Power	<b>RO</b>	24		功率方向标志
408H	REV_Energy	<b>RO</b>	24		能量方向标志
40CH	P90_Power	<b>RO</b>	16		电能通道功率的 90 度标志
410H	SignalStatus	<b>RO</b>	24		采样信号与系统状态寄存器
414H	PeakSagIntIovStau	<b>RO</b>	16		PeakSagINT 事件状态标志寄存器

#### 备注格式定义:

寄存器列表里面的描述会后缀一个 **Q n.m** 的格式, 意为符号数 + 有效位数。

例 **Q1.15** 即为该数据为 16Bit, 其中有 1Bit 符号位, 15Bit 有效数据。**Q0.23** 意为数据 23Bit, 没有符号位, 23Bit 有效数据。

## 22.7 计量参数寄存器说明

### 22.7.1 波形寄存器

偏移地址	000H	004H	008H	00CH	010H	014H	018H
寄存器	SPL_Ua	SPL_Ub	SPL_Uc	SPL_Ia	SPL_Ib	SPL_Ic	SPL_In
Bit	Bit[23:0]						
功能描述	波形寄存器有效位数为 22 位补码格式, Bit[23]符号 + Bit[22-2]数据 + Bit[1:0]补 0 波形寄存器的更新速度由 SPL[2:0]的 3 个 bit 控制, 数据源头可通过 WaveSel 配置为高通前或高通后或通道增益后的数据。 计算公式: $((SPL \gg 2) / 2^{21} / PGA) * ADCFullScale$ , 即为外部 Pin 的模拟采样实时信号						

(if (SPL>>2)大于等于  $2^{21}$  则为负数, 用(SPL>>2)- $2^{22}$  替代计算实时采样信号)

## 22.7.2 功率寄存器

全波

偏移地址	01CH	020H	024H	028H	02CH
寄存器	All_Power_Pa	All_Power_Pb	All_Power_Pc	All_Power_Pt	All_Power_Qa
偏移地址	030H	034H	038H	03CH	040H
寄存器	All_Power_Qb	All_Power_Qc	All_Power_Qt	All_Power_Sa	All_Power_Sb
偏移地址	044H	048H			
寄存器	All_Power_Sc	All_Power_St			
Bit	Bit[31:0]				
功能描述	全波功率寄存器为 32bit 补码格式, 最高位为符号位。 更新速度受控于 AVG_Power 寄存器, 即 EMUCLK/OSR/AVG_Power 就是功率寄存器的更新频率 (基波和谐波功率更新频率也是受控于 AVG_Power) 计算公式: 以 S 为例, $(All\_Power\_S/2^{31}) * ADCFullScaleU * ADCFullScaleI$ 即为输入信号端电压和电流通道的采样信号所计算的视在功率。(注意采样信号和真实强电电压、电流还有采样系统的传递系数) 注意: 合相功率寄存器的值等于分相功率寄存器之和的一半, 即: $Power\_xt = (Power\_xa + Power\_xb + Power\_xc) / 2$				

基波

偏移地址	178H	17CH	180H	184H
寄存器	Fund_Power_Pa	Fund_Power_Pb	Fund_Power_Pc	Fund_Power_Pt
偏移地址	188H	18CH	190H	194H
寄存器	Fund_Power_Qa	Fund_Power_Qb	Fund_Power_Qc	Fund_Power_Qt
偏移地址	198H	19CH	1A0H	1A4H
寄存器	Fund_Power_Sa	Fund_Power_Sb	Fund_Power_Sc	Fund_Power_St
Bit	Bit[31:0]			
功能描述	基波功率寄存器为 32bit 补码格式, 最高位为符号位。 详见全波功率寄存器			

谐波

偏移地址	254H	258H	25CH	260H
寄存器	Har_Power_Pa	Har_Power_Pb	Har_Power_Pc	Har_Power_Pt
Bit	Bit[31:0]			
功能描述	谐波有功功率寄存器为 32bit 补码格式, 最高位为符号位。 详见全波功率寄存器			

## 22.7.3 有效值寄存器

全波

偏移地址	04CH	050H	054H	058H
寄存器	All_RMS_Ua	All_RMS_Ub	All_RMS_Uc	All_RMS_Ia
偏移地址	05CH	060H	064H	
寄存器	All_RMS_Ib	All_RMS_Ic	All_RMS_In	
Bit	Bit[23:0]			
功能描述	全波有效值寄存器最高位恒为 0，有效位数 23Bit 的无符号数。更新频率由 AVG_RMS 寄存器配置，即 EMUCLK/OSR/AVG_RMS 寄存器的更新频率（基波和谐波有效值也是受控该 AVG_RMS）。 计算公式： $(All\_RMS/2^{23}/PGA)*ADCFullScale$ ，即为外部 Pin 模拟信号的有效值。			

## 基波

偏移地址	1A8H	1ACH	1B0H	1B4H
寄存器	Fund_RMS_Ua	Fund_RMS_Ub	Fund_RMS_Uc	Fund_RMS_Ia
偏移地址	1B8H	1BCH	1C0H	
寄存器	Fund_RMS_Ib	Fund_RMS_Ic	Fund_RMS_In	
Bit	Bit[23:0]			
功能描述	基波有效值寄存器最高位恒为 0，有效位数 23Bit 的无符号数。 详见全波有效值寄存器			

## 谐波

偏移地址	264H	268H	26CH	270H
寄存器	Har_RMS_Ua	Har_RMS_Ub	Har_RMS_Uc	Har_RMS_Ia
偏移地址	274H	278H	27CH	
寄存器	Har_RMS_Ib	Har_RMS_Ic	Har_RMS_In	
Bit	Bit[23:0]			
功能描述	谐波有效值寄存器最高位恒为 0，有效位数 23Bit 的无符号数。 详见全波有效值寄存器			

## 22.7.4 矢量和有效值寄存器

偏移地址	068H	06CH
寄存器	RMS_Ut	RMS_It
Bit	Bit[23:0]	
功能描述	电压电流通道的矢量和有效值寄存器的数据格式与全波有效值寄存器相同，最高位恒为 0，有效位数 23Bit 的无符号数 矢量和计算的全波和基波源头由 EMUCFG 的 UtFund_En 和 ItFund_En 选择，通道是否参与运算由 ItCfgX 和 UtCfgX 配置 计算公式： $(RMS\_Ut/2^{23})*ADCFullScale*2$ 为电压矢量和有效值(unit: mv)。 $(RMS\_It/2^{23})*ADCFullScale*2$ 为电流矢量和有效值(unit: mv)。 注意需要再乘 2 才是真实矢量和数值	

### 22.7.5 线电压有效值寄存器

偏移地址	070H	074H	078H
寄存器	RMS_UAB	RMS_UCB	RMS_UAC
Bit	Bit[23:0]		
功能描述	线电压有效值寄存器的数据格式与全波有效值寄存器相同，最高位恒为 0，有效位数 23Bit 的无符号数 线电压计算的全波和基波源头由 EMUCFG 的 FundLineU_En 选择 计算公式: $RMS\_UAB/2^{23}/PGA*ADCFullScale$		

### 22.7.6 电压频率寄存器

偏移地址	07CH	080H	084H	088H
寄存器	FreqUa	FreqUb	FreqUc	FreqU
Bit	Bit[23:0]			
功能描述	分相电压频率和合相电压频率寄存器，最高位恒为 0，有效位数 23bit 无符号 至少一路分相频率有效时则合相电压频率有效，否则均为复位值 0x7FFFFFFF；电压频率 FreqU 等于其中一路分相电压的有效频率，优先级顺序 A>B>C；分相电压过零有效时计算电压频率，否则为复位值 注意采样率变化时需要修改 ZXLostCFG 的 ZCNumMin 和 ZCNumMax 防止频率计算错误 计算公式：信号频率=(EMUCLK/OSR)/(FreqUx/128) 举例：以 FreqUa 寄存器为例，假设 EMUCLK=1638.4KHz,OSR=128，读取 FreqUa 为 0x8000，则根据计算公式可得信号频率： $1638400/128/(32768/128)=50Hz$			

### 22.7.7 功率因数寄存器

全波

偏移地址	08CH	090H	094H	098H
寄存器	All_PFa	All_PFb	All_PFc	All_PFt
Bit	Bit[23:0]			
功能描述	全波功率因数寄存器为 24bit 补码格式，最高位为符号位。 功率因数由对应的 Power_P/S 计算，表示范围[-1,1) 计算公式：功率因数=All_PF/2 <sup>23</sup>			

基波

偏移地址	1C4H	1C8H	1CCH	1D0H
寄存器	Fund_PFa	Fund_PFb	Fund_PFc	Fund_PFt
Bit	Bit[23:0]			
功能描述	基波功率因数寄存器为 24bit 补码格式，最高位为符号位。 详见全波功率因数寄存器			

## 22.7.8 角度寄存器

偏移地址	09CH	0A0H	0A4H	0A8H
寄存器	PhaseAngle_Ua	PhaseAngle_Ub	PhaseAngle_Uc	PhaseAngle_Ia
偏移地址	0ACH	0B0H	0B4H	
寄存器	PhaseAngle_Ib	PhaseAngle_Ic	All_Power_In	
Bit	Bit[23:0]			
功能描述	角度寄存器为 24bit 补码格式，最高位为符号位。 对应通道的过零有效才计算角度否则复位为 0；角度基准信号选择在 EMUCFG 的 AngleBaseSel[2:0]，基准信号的角度寄存器保持为 0，其他通道与基准通道角度均以基准信号为基准；自动和手动切换的控制位在 EMUCFG 的 AutoAngBase；自动切换时在高优先级基准无效时切换低优先级基准，优先级 Ua>Ub>Uc>Ia>Ib>Ic>In；角度表示范围[-180, 180) 计算公式：角度=PhaseAngle/ 2 <sup>23</sup> * 180			

## 22.7.9 电能寄存器

全波正向

偏移地址	0B8H	0BCH	0C0H	0C4H
寄存器	All_EnergyPos_Pa	All_EnergyPos_Pb	All_EnergyPos_Pc	All_EnergyPos_Pt
偏移地址	0C8H	0CCH	0D0H	0D4H
寄存器	All_EnergyPos_Qa	All_EnergyPos_Qb	All_EnergyPos_Qc	All_EnergyPos_Qt
偏移地址	0D8H	0DCH	0E0H	0E4H
寄存器	Energy_Sa	Energy_Sb	Energy_Sc	Energy_St
Bit	Bit[23:0]			
功能描述	全波正向电能寄存器为 24bit 无符号数，其中视在电能计量通道是全波基波复用，控制位在 EnergySSrc_Sel 电能寄存器默认为读后清零（可配置 EnergyRC_En=0 为读后不清零）。注意电能寄存器和计量发出的脉冲之间还有二级脉冲关系 AFDIV，也即电能累计 AFDIV 个以后才会发一个脉冲。由于 AFDIV 默认为 1，即默认状态下，电能和脉冲是同步关系的。 为防止配置为读后不清零时电能长时间出现累计溢出的现象，提供一个 EnergyOVIF 总标志位，同时提供 EMUIF_EnergyPOVIF 和 EMUIF_EnergyNOVIF 的总共 48 个能量通道溢出标志位			

全波反向

偏移地址	118H	11CH	120H	124H
寄存器	All_EnergyNeg_P a	All_EnergyNeg_P b	All_EnergyNeg_P c	All_EnergyNeg_Pt
偏移地址	128H	12CH	130H	134H
寄存器	All_EnergyNeg_Q a	All_EnergyNeg_Q b	All_EnergyNeg_Q c	All_EnergyNeg_Q t

Bit	Bit[23:0]
功能描述	全波反向电能寄存器，格式同全波正向电能寄存器

反向视在电能

偏移地址	158H	15CH	160H	164H
寄存器	EnergyNeg_SaCon nst	EnergyNeg_SbCon nst	EnergyNeg_ScCon nst	EnergyNeg_StCon st
Bit	Bit[23:0]			
功能描述	格式同全波正向电能寄存器，24bit 无符号 视在功率不存在负数，用于使能常数功率并且常数功率为负时累计电能			

基波电能寄存器（正反向）

偏移地址	1D4H	1D8H	1DCH	1E0H
寄存器	Fund_EnergyPos_ Pa	Fund_EnergyPos_ Pb	Fund_EnergyPos_ Pc	Fund_EnergyPos_ Pt
偏移地址	1E4H	1E8H	1ECH	1F0H
寄存器	Fund_EnergyPos_ Qa	Fund_EnergyPos_ Qb	Fund_EnergyPos_ Qc	Fund_EnergyPos_ Qt
偏移地址	214H	218H	21CH	220H
寄存器	Fund_EnergyNeg_ Pa	Fund_EnergyNeg_ Pb	Fund_EnergyNeg_ Pc	Fund_EnergyNeg_ Pt
偏移地址	224H	228H	22CH	230H
寄存器	Fund_EnergyNeg_ Qa	Fund_EnergyNeg_ Qb	Fund_EnergyNeg_ Qc	Fund_EnergyNeg_ Qt
Bit	Bit[23:0]			
功能描述	基波电能寄存器为 24bit 无符号数 详见全波电能寄存器			

谐波电能寄存器

偏移地址	280H	284H	288H	28CH
寄存器	Har_EnergyPos_P a	Har_EnergyPos_P b	Har_EnergyPos_P c	Har_EnergyPos_Pt
偏移地址	2A0H	2A4H	2A8H	2ACH
寄存器	Har_EnergyNeg_P a	Har_EnergyNeg_P b	Har_EnergyNeg_P c	Har_EnergyNeg_P t
Bit	Bit[23:0]			
功能描述	谐波电能寄存器为 24bit 无符号数 详见全波电能寄存器			

## 22.7.10 快速脉冲计数寄存器

全波正向快速脉冲计数寄存器

偏移地址	0E8H	0ECH	0F0H	0F4H
寄存器	All_CFCntPos_Pa	All_CFCntPos_Pb	All_CFCntPos_Pc	All_CFCntPos_Pt
偏移地址	0F8H	0FCH	100H	104H
寄存器	All_CFCntPos_Qa	All_CFCntPos_Qb	All_CFCntPos_Qc	All_CFCntPos_Qt
偏移地址	108H	10CH	110H	114H
寄存器	CFCntPos_Sa	CFCntPos_Sb	CFCntPos_Sc	CFCntPos_St
Bit	Bit[23:0]			
功能描述	全波正向 CF 通路快速脉冲计数寄存器，格式为：Bit0-Bit19 数据，Bit20~23 补符号 将电能累计模块溢出的 CFxCnt 开放出来供用户读取以观察累计过程（只可读，不可以写）			

## 全波反向快速脉冲计数寄存器

偏移地址	138H	13CH	140H	144H
寄存器	All_CFCntNeg_Pa	All_CFCntNeg_Pb	All_CFCntNeg_Pc	All_CFCntNeg_Pt
偏移地址	148H	14CH	150H	154H
寄存器	All_CFCntNeg_Q a	All_CFCntNeg_Q b	All_CFCntNeg_Q c	All_CFCntNeg_Qt
Bit	Bit[23:0]			
功能描述	格式同全波正向 CF 通路快速脉冲计数寄存器			

## 反向视在电能快速脉冲计数寄存器

偏移地址	168H	16CH	170H	174H
寄存器	CFCntNeg_Sa	CFCntNeg_Sb	CFCntNeg_Sc	CFCntNeg_St
Bit	Bit[23:0]			
功能描述	格式同全波正向快速脉冲计数寄存器，Bit0-Bit19 数据，Bit20~23 补符号 视在电能反向 CF 通路快速脉冲计数寄存器			

## 基波电能快速脉冲计数寄存器（正反向）

偏移地址	1F4H	1F8H	1FCH	200H
寄存器	Fund_CFCntPos_ Pa	Fund_CFCntPos_ Pb	Fund_CFCntPos_ Pc	Fund_CFCntPos_ Pt
偏移地址	204H	208H	20CH	210H
寄存器	Fund_CFCntPos_ Qa	Fund_CFCntPos_ Qb	Fund_CFCntPos_ Qc	Fund_CFCntPos_ Qt
偏移地址	234H	238H	23CH	240H
寄存器	Fund_CFCntNeg_ Pa	Fund_CFCntNeg_ Pb	Fund_CFCntNeg_ Pc	Fund_CFCntNeg_ Pt
偏移地址	244H	248H	24CH	250H
寄存器	Fund_CFCntNeg_ Qa	Fund_CFCntNeg_ Qb	Fund_CFCntNeg_ Qc	Fund_CFCntNeg_ Qt
Bit	Bit[23:0]			
功能描述	基波电能 CF 通路快速脉冲计数寄存器，格式为：Bit0-Bit19 数据，Bit20~23 补符号 详见全波正向快速脉冲计数寄存器			

## 谐波有功快速脉冲计数寄存器

偏移地址	290H	294H	298H	29CH
寄存器	Har_CFCntPos_Pa	Har_CFCntPos_Pb	Har_CFCntPos_Pc	Har_CFCntPos_Pt
偏移地址	2B0H	2B4H	2B8H	2BCH
寄存器	Har_CFCntNeg_P a	Har_CFCntNeg_P b	Har_CFCntNeg_P c	Har_CFCntNeg_Pt
Bit	Bit[23:0]			
功能描述	谐波电能 CF 通路快速脉冲计数寄存器，格式为：Bit0-Bit19 数据，Bit20~23 补符号 详见全波正向快速脉冲计数寄存器			

### 22.7.11 ADC 采样直流数据寄存器

偏移地址	2C0H	2C4H	2C8H	2CCH
寄存器	DCVar_Ua	DCVar_Ub	DCVar_Uc	DCVar_Ia
偏移地址	2D0H	2D4H	2D8H	
寄存器	DCVar_Ib	DCVar_Ic	DCVar_In	
Bit	Bit[23:0]			
功能描述	ADC 采样直流数据的 Bit0-Bit20 数据，Bit21 ~23 补充为符号位 源头取自高通前，有效位数与波形采样寄存器相同 计算公式： $DCVar/2^{21}/PGA * ADCFullScale$ 即为外部 Pin 的模拟采样实时信号			

### 22.7.12 半波周期峰值寄存器

偏移地址	2DCH	2E0H	2E4H	2E8H
寄存器	UpdHalfWave_Ua	UpdHalfWave_Ub	UpdHalfWave_Uc	UpdHalfWave_Ia
偏移地址	2ECH	2F0H	2F4H	
寄存器	UpdHalfWave_Ib	UpdHalfWave_Ic	UpdHalfWave_In	
Bit	Bit[23:0]			
功能描述	半波峰值寄存器有效位数为 22bit，Bit0~Bit20 为数据，Bit21~Bit23 全都为符号位。 每次 ADC 正反向过零，将该寄存器更新，得到此半波时间内的 ADC 波形的最大值，该寄存器取自高通后的数据。 计算公式：U 为例， $(RMS\_U/4)*1.414 = HalfWave\_U$			

### 22.7.13 瞬时有效值平方寄存器

偏移地址	2F8H	2FCH	300H	304H
寄存器	QuartRmsIns_Ua	QuartRmsIns_Ub	QuartRmsIns_Uc	QuartRmsIns_Ia
偏移地址	308H	30CH	310H	
寄存器	QuartRmsIns_Ib	QuartRmsIns_Ic	QuartRmsIns_In	
Bit	Bit[23:0]			
功能描述	瞬时有效值平方寄存器即把 ADC 各通道的波形数据自乘后的瞬时结果放到寄存器，数据格式 32Bit 无符号，其更新频率为 CLK/OSR。 芯片提供 EMUCFG.SQRMS_Src1 来选择该寄存器的数据，是自乘后的滤波器前数据还是滤波器后数据。			

### 22.7.14 Peak 相关事件寄存器

#### Peak 事件期间最值寄存器

偏移地址	314H	318H	31CH
寄存器	PeakVpUa	PeakVpUb	PeakVpUc
Bit	Bit[23:0]		
功能描述	Peak 事件发生期间的最值即最大半波峰值，每半周波更新，由 ADCCON 的 PeakSagSrc 选择数据源头 数据源为半波峰值时：Bit0-Bit20 数据，Bit21~23 补充为符号位 数据源为半波有效值时：Bit0-Bit22 数据，Bit23 恒为 0		

#### Peak 事件持续时间寄存器

偏移地址	320H	324H	328H
寄存器	PeakCntUa	PeakCntUb	PeakCntUc
Bit	Bit[23:0]		
功能描述	PEAK 工况持续时间计数寄存器，24 位无符号位，记录的是 PEAK 事件持续的半周波数。每半波更新一次，退出事件保持最后一次更新值不变		

### 22.7.15 SAG 相关事件寄存器

#### SAG 期间最值寄存器

偏移地址	32CH	330H	334H
寄存器	SagVpUa	SagVpUb	SagVpUc
Bit	Bit[23:0]		
功能描述	SAG 事件发生期间的最值即最小半波峰值，每半周波更新，由 ADCCON 的 PeakSagSrc 选择数据源头 数据源为半波峰值时：Bit0-Bit20 数据，Bit21~23 补充为符号位 数据源为半波有效值时：Bit0-Bit22 数据，Bit23 恒为 0		

## SAG 事件持续时间寄存器

偏移地址	338H	33CH	340H
寄存器	SagCntUa	SagCntUb	SagCntUc
Bit	Bit[23:0]		
功能描述	SAG 工况持续时间计数寄存器，24 位无符号位，记录的是 SAG 事件持续的半周波数。每半波更新一次，退出事件保持最后一次更新值不变		

## 22.7.16 UINT 事件相关寄存器

## Int 期间最值寄存器

偏移地址	344H	348H	34CH
寄存器	IntVpUa	IntVpUb	IntVpUc
Bit	Bit[23:0]		
功能描述	Int 事件发生期间的最值，由 ADCCON 的 PeakSagSrc 选择数据源头 数据源为半波峰值时：Bit0-Bit20 数据，Bit21~23 补充为符号位 数据源为半波有效值时：Bit0-Bit22 数据，Bit23 恒为 0		

## Int 事件持续时间寄存器

偏移地址	350H	354H	358H
寄存器	IntCntUa	IntCntUb	IntCntUc
Bit	Bit[23:0]		
功能描述	INT 工况持续时间计数寄存器，24 位无符号位，记录的是 INT 事件持续的半周波数。每半波更新一次，退出事件保持最后一次更新值不变		

## 22.7.17 电流过流事件相关寄存器

## OV 期间最值寄存器

偏移地址	35CH	360H	364H	368H
寄存器	OverFlowVpIa	OverFlowVpIb	OverFlowVpIc	OverFlowVpIn
Bit	Bit[23:0]			
功能描述	IOV 事件发生期间的最值，由 ADCCON 的 PeakSagSrc 选择数据源头 数据源为半波峰值时：Bit0-Bit20 数据，Bit21~23 补充为符号位 数据源为半波有效值时：Bit0-Bit22 数据，Bit23 恒为 0			

## OV 事件持续时间寄存器

偏移地址	36CH	370H	374H	378H
寄存器	IntCntUa	IntCntUb	IntCntUc	
Bit	Bit[23:0]			
功能描述	IOV 工况持续时间计数寄存器，24 位无符号位，记录的是 OV 事件持续的半周波数。每半波更新一次，退出事件保持最后一次更新值不变			

### 22.7.18 自动温补增益寄存器

偏移地址	37CH	380H	384H
寄存器	AutoTgain_U	AutoTgain_I	AutoTgain_In
Bit	Bit[15:0]		
功能描述	格式同通道增益校正寄存器，16bit 补码格式，最高位为符号位 自动温度补偿功能开启后，该寄存器的值表示在当前温度点的 U/I/In 通道温度自动补偿值。该寄存器由计量单元根据自动温补的 ABC 三项系数自动计算出的 Gain 值		

### 22.7.19 半波有效值数据

偏移地址	388H	38CH	390H	394H
寄存器	HpRMS_Reg_Ua	HpRMS_Reg_Ub	HpRMS_Reg_Uc	HpRMS_Reg_Ia
偏移地址	398H	39CH	3A0H	
寄存器	HpRMS_Reg_Ib	HpRMS_Reg_Ic	HpRMS_Reg_In	
Bit	Bit[23:0]			
功能描述	半波有效值数据，来自工频同步系统，与 All_RMS 的数据格式是一致			

### 22.7.20 DFT 基波运算结果寄存器

偏移地址	3A4H	3A8H	3ACH	3B0H
寄存器	DFT_Value_UaReal	DFT_Value_UaImage	DFT_Value_UbReal	DFT_Value_UbImage
偏移地址	3B4H	3B8H	3BCH	3C0H
寄存器	DFT_Value_UcReal	DFT_Value_UcImage	DFT_Value_IaReal	DFT_Value_IaImage
偏移地址	3C4H	3C8H	3CCH	3D0H
寄存器	DFT_Value_IbReal	DFT_Value_IbImage	DFT_Value_IcReal	DFT_Value_IcImage
偏移地址	3D4H	3D8H		
寄存器	DFT_Value_InReal	DFT_Value_InImage		
Bit	Bit[23:0]			
功能描述	DFT 结果的 Real & image，24bit 有符号 计算公式：Real/2 <sup>22</sup> (负数则(Real-2 <sup>24</sup> )/2 <sup>22</sup> )取模取相位，注意是除以 2 <sup>22</sup>			

### 22.7.21 电压闪变 Flicker 寄存器

偏移地址	3E4H	3E8H	3ECH
寄存器	FlickerUaPinst	FlickerUbPinst	FlickerUcPinst

Bit	Bit[31:0]
功能描述	闪存寄存器为 32bit 补码格式，最高位为符号位

## 22.7.22 校验和寄存器

### Checksum1 寄存器

偏移地址	3F0H
寄存器	Checksum1
Bit	Bit[31:0]
功能描述	32Bit 无符号，参与该校验的校表参数寄存器相加。 校验范围 000~034H，054~104H，110H~1C0H，250H~2D4H， 2E4H~348H,358H，36CH，370H 共计算 167 个寄存器

### Checksum2 寄存器

偏移地址	3F4H
寄存器	Checksum2
Bit	Bit[31:0]
功能描述	32Bit 无符号，参与该校验的校表参数寄存器相加。 用户可暂不使用。

### Checksum3 寄存器

偏移地址	3F8H
寄存器	Checksum3
Bit	Bit[31:0]
功能描述	32Bit 无符号，参与该校验的校表参数寄存器相加。 校验范围 A08H~A3CH

## 22.8 校表寄存器列表

EMU 模块校表寄存器总基地址: 0x 4001 C000					
偏移地址	名称	读写方式	有效长度 Bit	复位值	功能描述
<b>EMU 校表功能相关寄存器 0x 4001 C000 + Offset 000H</b>					
000H	DecShift_UaH	R/W	12	0x0000	Ua 通道移动 Onebit 相位校正大拍(4bit*3 段) 总结结构为: (4Bit High + 7Bit Low)*3 段 由于相位分三段, Decshift 寄存器存放其实为 UaH 和 UaL 的拼接方式, 见说明
004H	DecShift_UaL	R/W	21	0x000000	Ua 通道移动 Onebit 相位校正小拍(7bit*3 段)
008H	DecShift_UbH	R/W	12	0x0000	Ub 通道移动 Onebit 相位校正大拍
00CH	DecShift_UbL	R/W	21	0x000000	Ub 通道移动 Onebit 相位校正小拍
010H	DecShift_UcH	R/W	12	0x0000	Uc 通道移动 Onebit 相位校正大拍
014H	DecShift_UcL	R/W	21	0x000000	Uc 通道移动 Onebit 相位校正小拍
018H	DecShift_IaH	R/W	12	0x0000	Ia 通道移动 Onebit 相位校正大拍
01CH	DecShift_IaL	R/W	21	0x000000	Ia 通道移动 Onebit 相位校正小拍
020H	DecShift_IbH	R/W	12	0x0000	Ib 通道移动 Onebit 相位校正大拍
024H	DecShift_IbL	R/W	21	0x000000	Ib 通道移动 Onebit 相位校正小拍
028H	DecShift_IcH	R/W	12	0x0000	Ic 通道移动 Onebit 相位校正大拍
02CH	DecShift_IcL	R/W	21	0x000000	Ic 通道移动 Onebit 相位校正小拍
030H	DecShift_InH	R/W	12	0x0000	In 通道移动 Onebit 相位校正大拍
034H	DecShift_InL	R/W	21	0x000000	In 通道移动 Onebit 相位校正小拍
038H	ADCOffset_Ua	R/W	22	0x000000	Ua 通道 ADC 直流偏置校正, 其和 SPL 的数据长度对齐(低对齐), 均为 22Bit ADCOffset 格式和 SPL_XX (SPL 内部数据, 并非寄存器所列的存储格式) 完全一致。即低位对齐
03CH	ADCOffset_Ub	R/W	22	0x000000	Ub 通道 ADC 直流偏置校正 Q1.21
040H	ADCOffset_Uc	R/W	22	0x000000	Uc 通道 ADC 直流偏置校正 Q1.21
044H	ADCOffset_Ia	R/W	22	0x000000	Ia 通道 ADC 直流偏置校正 Q1.21
048H	ADCOffset_Ib	R/W	22	0x000000	Ib 通道 ADC 直流偏置校正 Q1.21
04CH	ADCOffset_Ic	R/W	22	0x000000	Ic 通道 ADC 直流偏置校正 Q1.21
050H	ADCOffset_In	R/W	22	0x000000	In 通道 ADC 直流偏置校正 Q1.21
054H	Tgain_U	R/W	16	0x0000	U 采样通道的温度增益(三路电压共用, 用于补偿温度变化等因素导致的 U 通道整体偏移) Q1.15
058H	Tgain_I	R/W	16	0x0000	I 采样通道的温度增益(三路电流共用, 用于补偿温度变化等因素导致的 I 通道整体偏移)

					Q1.15
05CH	Tgain_In	R/W	16	0x0000	In 零线电流采样通道专用温度增益 Q1.15
060H	Gain_Ua	R/W	16	0x0000	Ua 采样通道增益 Q1.15
064H	Gain_Ub	R/W	16	0x0000	Ub 采样通道增益 Q1.15
068H	Gain_Uc	R/W	16	0x0000	Uc 采样通道增益 Q1.15
06CH	Gain_Ia	R/W	16	0x0000	Ia 采样通道增益 Q1.15
070H	Gain_Ib	R/W	16	0x0000	Ib 采样通道增益 Q1.15
074H	Gain_Ic	R/W	16	0x0000	Ic 采样通道增益 Q1.15
078H	Gain_In	R/W	16	0x0000	In 采样通道增益 Q1.15
07CH	Digital_gain	R/W	14	0x0000	数字增益，每通道 2Bit 分别控制放大为 1/2/4/8 倍。数字增益为左移位数 DGUa/DGUb/DGUc/DGIa/DGIb/DGIc/ DGIn
080H	All_GPa	R/W	16	0x0000	A 相全波有功功率增益 Q1.15
084H	All_GPb	R/W	16	0x0000	B 相全波有功功率增益 Q1.15
088H	All_GPc	R/W	16	0x0000	C 相全波有功功率增益 Q1.15
08CH	All_GQa	R/W	16	0x0000	A 相全波无功功率增益 Q1.15
090H	All_GQb	R/W	16	0x0000	B 相全波无功功率增益 Q1.15
094H	All_GQc	R/W	16	0x0000	C 相全波无功功率增益 Q1.15
098H	All_GSa	R/W	16	0x0000	A 相全波视在功率增益 Q1.15
09CH	All_GSb	R/W	16	0x0000	B 相全波视在功率增益 Q1.15
0A0H	All_GSc	R/W	16	0x0000	C 相全波视在功率增益 Q1.15
0A4H	All_GphsA0	R/W	16	0x0000	A 相全波相位校正 0 Q1.15
0A8H	All_GphsA1	R/W	16	0x0000	A 相全波相位校正 1 Q1.15
0ACH	All_GphsA2	R/W	16	0x0000	A 相全波相位校正 2 Q1.15
0B0H	All_GphsB0	R/W	16	0x0000	B 相全波相位校正 0 Q1.15
0B4H	All_GphsB1	R/W	16	0x0000	B 相全波相位校正 1 Q1.15
0B8H	All_GphsB2	R/W	16	0x0000	B 相全波相位校正 2 Q1.15
0BCH	All_GphsC0	R/W	16	0x0000	C 相全波相位校正 0 Q1.15
0C0H	All_GphsC1	R/W	16	0x0000	C 相全波相位校正 1 Q1.15
0C4H	All_GphsC2	R/W	16	0x0000	C 相全波相位校正 2 Q1.15
0C8H	Reserved				Reserved
0CCH	All_PowerOffset_Pa	R/W	24	0x000000	A 相全波有功功率偏置校正（功率 Offset 为 24bit -Q1.23） PowerOffset 和 PowerP/Q 是低对齐关系即对应 PowerP/Q 的 bit[23:0]
0D0H	All_PowerOffset_Pb	R/W	24	0x000000	B 相全波有功功率偏置校正 Q1.23
0D4H	All_PowerOffset_Pc	R/W	24	0x000000	C 相全波有功功率偏置校正 Q1.23
0D8H	All_PowerOffset_Qa	R/W	24	0x000000	A 相全波无功功率偏置校正 Q1.23
0DCH	All_PowerOffset_Qb	R/W	24	0x000000	B 相全波无功功率偏置校正 Q1.23
0E0H	All_PowerOffset_Qc	R/W	24	0x000000	C 相全波无功功率偏置校正 Q1.23
0E4H	All_RmsOffset_Ua	R/W	24	0x000000	Ua 通道全波电压有效值偏置校正 Q0.24 Bit0-Bit23 数据,无符号

					自乘后直接低位对齐相减： <b><math>RMS = \sqrt{Um - RMSOffset * 2^7}</math></b>
0E8H	All_RmsOffset_Ub	R/W	24	0x000000	Ub 通道全波电压有效值偏置校正 <b>Q0.24</b>
0ECH	All_RmsOffset_Uc	R/W	24	0x000000	Uc 通道全波电压有效值偏置校正 <b>Q0.24</b>
0F0H	All_RmsOffset_Ia	R/W	24	0x000000	Ia 通道全波电压有效值偏置校正 <b>Q0.24</b>
0F4H	All_RmsOffset_Ib	R/W	24	0x000000	Ib 通道全波电压有效值偏置校正 <b>Q0.24</b>
0F8H	All_RmsOffset_Ic	R/W	24	0x000000	Ic 通道全波电压有效值偏置校正 <b>Q0.24</b>
0FCH	All_RmsOffset_In	R/W	24	0x000000	In 通道全波电压有效值偏置校正 <b>Q0.24</b>
100H	RMSOffset_Ut	R/W	24	0x000000	合相电压有效值偏置校正, (可通过 Bit 选 All/Fund) <b>Q0.24</b>
104H	RMSOffset_It	R/W	24	0x000000	合相电流有效值偏置校正, (可通过 Bit 选 All/Fund) <b>Q0.24</b>
108H	Reserved				Reserved
10CH	Reserved				Reserved
<b>基波与谐波校准寄存器</b>					
110H	Fund_GPa	R/W	16	0x0000	A 相基波有功功率增益 <b>Q1.15</b>
114H	Fund_GPb	R/W	16	0x0000	B 相基波有功功率增益 <b>Q1.15</b>
118H	Fund_GPc	R/W	16	0x0000	C 相基波有功功率增益 <b>Q1.15</b>
11CH	Fund_GQa	R/W	16	0x0000	A 相基波无功功率增益 <b>Q1.15</b>
120H	Fund_GQb	R/W	16	0x0000	B 相基波无功功率增益 <b>Q1.15</b>
124H	Fund_GQc	R/W	16	0x0000	C 相基波无功功率增益 <b>Q1.15</b>
128H	Fund_GSa	R/W	16	0x0000	A 相基波视在功率增益 <b>Q1.15</b>
12CH	Fund_GSb	R/W	16	0x0000	B 相基波视在功率增益 <b>Q1.15</b>
130H	Fund_GSc	R/W	16	0x0000	C 相基波视在功率增益 <b>Q1.15</b>
134H	Fund_GphsA0	R/W	16	0x0000	A 相基波相位校正 0 <b>Q1.15</b>
138H	Fund_GphsA1	R/W	16	0x0000	A 相基波相位校正 1 <b>Q1.15</b>
13CH	Fund_GphsA2	R/W	16	0x0000	A 相基波相位校正 2 <b>Q1.15</b>
140H	Fund_GphsB0	R/W	16	0x0000	B 相基波相位校正 0 <b>Q1.15</b>
144H	Fund_GphsB1	R/W	16	0x0000	B 相基波相位校正 1 <b>Q1.15</b>
148H	Fund_GphsB2	R/W	16	0x0000	B 相基波相位校正 2 <b>Q1.15</b>
14CH	Fund_GphsC0	R/W	16	0x0000	C 相基波相位校正 0 <b>Q1.15</b>
150H	Fund_GphsC1	R/W	16	0x0000	C 相基波相位校正 1 <b>Q1.15</b>
154H	Fund_GphsC2	R/W	16	0x0000	C 相基波相位校正 2 <b>Q1.15</b>
158H	<b>Reserved</b>				
15CH	Fund_PowerOffset_Pa	R/W	24	0x000000	A 相基波有功功率偏置校正 <b>Q1.23</b>
160H	Fund_PowerOffset_Pb	R/W	24	0x000000	B 相基波有功功率偏置校正 <b>Q1.23</b>
164H	Fund_PowerOffset_Pc	R/W	24	0x000000	C 相基波有功功率偏置校正 <b>Q1.23</b>
168H	Fund_PowerOffset_Qa	R/W	24	0x000000	A 相基波无功功率偏置校正 <b>Q1.23</b>
16CH	Fund_PowerOffset_Qb	R/W	24	0x000000	B 相基波无功功率偏置校正 <b>Q1.23</b>
170H	Fund_PowerOffset_Qc	R/W	24	0x000000	C 相基波无功功率偏置校正 <b>Q1.23</b>
174H	Fund_RmsOffset_Ua	R/W	24	0x000000	Ua 通道基波电压有效值偏置校正 <b>Q0.24</b>
178H	Fund_RmsOffset_Ub	R/W	24	0x000000	Ub 通道基波电压有效值偏置校正 <b>Q0.24</b>

17CH	Fund_RmsOffset_Uc	R/W	24	0x000000	Uc 通道基波电压有效值偏置校正 Q0.24
180H	Fund_RmsOffset_Ia	R/W	24	0x000000	Ia 通道基波电流有效值偏置校正 Q0.24
184H	Fund_RmsOffset_Ib	R/W	24	0x000000	Ib 通道基波电流有效值偏置校正 Q0.24
188H	Fund_RmsOffset_Ic	R/W	24	0x000000	Ic 通道基波电流有效值偏置校正 Q0.24
18CH	Fund_RmsOffset_In	R/W	24	0x000000	In 通道基波电流有效值偏置校正 Q0.24
190H	Har_GPa	R/W	16	0x0000	A 相谐波有功功率增益 Q1.15
194H	Har_GPb	R/W	16	0x0000	B 相谐波有功功率增益 Q1.15
198H	Har_GPc	R/W	16	0x0000	C 相谐波有功功率增益 Q1.15
19CH	Har_PowerOffset_Pa	R/W	24	0x000000	A 相谐波有功功率偏置校正 Q1.23
1A0H	Har_PowerOffset_Pb	R/W	24	0x000000	B 相谐波有功功率偏置校正 Q1.23
1A4H	Har_PowerOffset_Pc	R/W	24	0x000000	C 相谐波有功功率偏置校正 Q1.23
1A8H	Har_RmsOffset_Ua	R/W	24	0x000000	Ua 通道谐波电压有效值偏置校正 Q0.24
1ACH	Har_RmsOffset_Ub	R/W	24	0x000000	Ub 通道谐波电压有效值偏置校正 Q0.24
1B0H	Har_RmsOffset_Uc	R/W	24	0x000000	Uc 通道谐波电压有效值偏置校正 Q0.24
1B4H	Har_RmsOffset_Ia	R/W	24	0x000000	Ia 通道谐波电流有效值偏置校正 Q0.24
1B8H	Har_RmsOffset_Ib	R/W	24	0x000000	Ib 通道谐波电流有效值偏置校正 Q0.24
1BCH	Har_RmsOffset_Ic	R/W	24	0x000000	Ic 通道谐波电流有效值偏置校正 Q0.24
1C0H	Har_RmsOffset_In	R/W	24	0x000000	In 通道谐波电流有效值偏置校正 Q0.24
1C4H	PowerConst_All_Pa	R/W	32	0x00000000	第 1 电能通路常数功率 <b>PowerConst 均为 Q1.31</b> <b>Bit0-Bit30 数据, Bit31 为符号位</b> 注: 由于用于电能计算的功率是 Power 寄存器的一半。所以 PowerConst 用户写入应为 Power 寄存器一半。
1C8H	PowerConst_All_Pb	R/W	32	00000000	第 2 电能通路常数功率
1CCH	PowerConst_All_Pc	R/W	32	00000000	第 3 电能通路常数功率
1D0H	PowerConst_All_Pt	R/W	32	00000000	第 4 电能通路常数功率
1D4H	PowerConst_All_Qa	R/W	32	00000000	第 5 电能通路常数功率
1D8H	PowerConst_All_Qb	R/W	32	00000000	第 6 电能通路常数功率
1DCH	PowerConst_All_Qc	R/W	32	00000000	第 7 电能通路常数功率
1E0H	PowerConst_All_Qt	R/W	32	00000000	第 8 电能通路常数功率
1E4H	PowerConst_Sa	R/W	32	00000000	第 9 电能通路常数功率
1E8H	PowerConst_Sb	R/W	32	00000000	第 10 电能通路常数功率
1ECH	PowerConst_Sc	R/W	32	00000000	第 11 电能通路常数功率
1F0H	PowerConst_St	R/W	32	00000000	第 12 电能通路常数功率
1F4H	PowerConst_Fund_Pa	R/W	32	00000000	第 13 电能通路常数功率
1F8H	PowerConst_Fund_Pb	R/W	32	00000000	第 14 电能通路常数功率
1FCH	PowerConst_Fund_Pc	R/W	32	00000000	第 15 电能通路常数功率
200H	PowerConst_Fund_Pt	R/W	32	00000000	第 16 电能通路常数功率
204H	PowerConst_Fund_Qa	R/W	32	00000000	第 17 电能通路常数功率
208H	PowerConst_Fund_Qb	R/W	32	00000000	第 18 电能通路常数功率
20CH	PowerConst_Fund_Qc	R/W	32	00000000	第 19 电能通路常数功率

210H	PowerConst_Fund_Qt	R/W	32	00000000	第 20 电能通路常数功率
214H	PowerConst_Har_Pa	R/W	32	00000000	第 21 电能通路常数功率
218H	PowerConst_Har_Pb	R/W	32	00000000	第 22 电能通路常数功率
21CH	PowerConst_Har_Pc	R/W	32	00000000	第 23 电能通路常数功率
220H	PowerConst_Har_Pt	R/W	32	00000000	第 24 电能通路常数功率
224H	AccuOffset_Har_Pa_Pos	R/W	32	00000000	A 相谐波正向电能偏置功率 <b>注: AccuOffset_P 只可以写正</b> Bit0-Bit30 数据, Bit31 为符号位, 对齐方式和 PowerConst 一致
228H	AccuOffset_Har_Pa_Neg	R/W	32	00000000	A 相谐波反向电能偏置功率 <b>注: AccuOffset_N 只可以写负!</b>
22CH	AccuOffset_Har_Pb_Pos	R/W	32	00000000	B 相谐波正向电能偏置功率
230H	AccuOffset_Har_Pb_Neg	R/W	32	00000000	B 相谐波反向电能偏置功率
234H	AccuOffset_Har_Pc_Pos	R/W	32	00000000	C 相谐波正向电能偏置功率
238H	AccuOffset_Har_Pc_Neg	R/W	32	00000000	C 相谐波反向电能偏置功率
23CH	AccuOffset_Har_Pt_Pos	R/W	32	00000000	合相谐波正向电能偏置功率
240H	AccuOffset_Har_Pt_Neg	R/W	32	00000000	合相谐波反向电能偏置功率
244H	UconstA	R/W	23	00000000	A 相常数电压通道 ALL_RMS_Ua 计算 A 相视在功率并累加电能 (只有全波)
248H	UconstB	R/W	23	00000000	B 相常数电压通道 ALL_RMS_Ub 计算 B 相视在功率并累加电能 (只有全波)
24CH	UconstC	R/W	23	00000000	C 相常数电压通道 ALL_RMS_Uc 计算 C 相视在功率并累加电能 (只有全波)
250H	<b>ZXLostCFG</b>	<b>R/W</b>	<b>24</b>	0xA080C0	<b>ZCNumMin / ZCNumMax / NumFund</b>
254H	PStart	R/W	16	0x0040	起动潜动阈值设置寄存器 P/S/HarStart 格式与对齐关系: Bit0-Bit15 数据, 无符号。与 PowerP/Q/S 绝对值的 bit8—bit23 进行比较, 与 Fund_RMS/All_RMS 值的 bit0—bit15 低位对齐进行比较。即若选择功率方式, 则将该 Pstart 左移 8bit 与 Power 对比。若选择有效值方式, 则将该 Pstart 直接与 All/Fund_RMS 对比
258H	QStart	R/W	16	0x0080	起动功率设置寄存器
25CH	HarStart	R/W	16	0x0040	谐波起动设置寄存器
260H	AVG_Energy	R/W	14	0x0100	能量通路平均模块点数设置 <b>该寄存器上限 0x2C88, 下限 0x002D</b> <b>写入大于 2C88 强制等于 2C88, 小于 002D 强制等于 002D</b>
264H	AVG_PowerREG	R/W	14	0x0A00	功率通路平均模块点数设置 50Hz 的 10 个周波, 即 5Hz 更新频率. <b>该寄</b>

					寄存器上限 0x2C88, 下限 0x002D
268H	AVG_RMS	R/W	14	0x0A00	分相有效值通路平均模块点数设置, 所有分相有效值(全/基/谐波共用)该寄存器上限 0x2C88, 下限 0x002D
26CH	AVG_RMST	R/W	14	0x0A00	合相有效值通路平均模块点数设置 该寄存器上限 0x2C88, 下限 0x002D
270H	HFCConstAF	R/W	20	0x001000	全波/基波电能高频脉冲 HFCConst Bit0-Bit19 为数据位, 无符号。 Bit20-Bit23 恒 0 该寄存器 用户不允许写 0 和 1
274H	HFCConstHar	R/W	20	0x001000	谐波电能高频脉冲 HFCConst 同 HFCConstAF
278H	AFDIV	R/W	15	0x0001	全波/基波电能脉冲分辨率, 无符号有效 15Bit, 即 Bit0-14 位数据, Bit15 恒 0 寄存器可读写 0, 但是寄存器写入 0 内部强制为 1 运算
27CH	HarDIV	R/W	15	0x0001	谐波电能脉冲分辨率, 无符号有效 15Bit, 即 Bit0-14 位数据, Bit15 恒 0 同 AFDIV
280H	ChannelSel	R/W	23	0x000000	采样通道互换与反向寄存器
284H	PGACON	R/W	11	0x0000	7 路采样通道的模拟 PGA 控制
288H	ADCCFG	R/W	23	0x000307	系统时钟, 源头等 ADC 配置
28CH	ADCCON	R/W	23	0x0000	ADC 开关, 事件的功能开关等
290H	EMUCFG	R/W	23	0x000000	计量相关配置选择
294H	FilterCon	R/W	21	0x04D0C0	Rosi 与常数电压寄存器
298H	FilterCFG	R/W	19	0x056DB5	常规滤波器系数配置
29CH	StartCFG	R/W		0x000000	起动作配置
2A0H	ANACFG1	R/W		0x0C411B	模拟参数配置寄存器
2A4H	ANACFG2	R/W		0x04707F	模拟参数配置寄存器
2A8H	Reserved	R/W			
2ACH	ULostCMPCFG	R/W		0x00200E	全失压比较器配置
2B0H	ULostPRDCFG	R/W		0x0A403C	全失压比较器分时模式配置
2B4H	FailVoltage	R/W	16	0x0300	和 All_RMSU 的 Bit[23: 8]对齐
2B8H	LPMoDeCON	R/W	8	0x50	低功耗模式控制寄存器
2BCH	CFxRun	R/W	24	0x000000	电能通路的能量累加独立使能
2C0H	EnergyCon	R/W	24	0x000400	计算电能的功率选择
2C4H	PowerSrcCFG1	R/W	24	0x000000	选择用于电能累加的功率
2C8H	PowerSrcCFG2	R/W	24	0x000000	选择用于电能累加的功率
2CCH	PowerMoDeCFG	R/W	24	0x03F000	电能累计模块与功率分选四种模式
2D0H	CFCFG1	R/W	24	0x000000	能量脉冲选择寄存器 1
2D4H	CFCFG2	R/W		0x000000	能量脉冲选择寄存器 2
2D8H	Reserved				暂无用途
2DCH	Reserved				暂无用途

2E0H	PowerConstLoad	R/W	8	00	常数功率的 Load 同步指令对 24 路生效, 只有正确指令才能加载常数功率 (前提使能常数功率): Load 指令为 0xBC
2E4H	UPeakSagCyc	R/W	16	0x0001	sag/peak 持续半周波数设置
2E8H	UIntCyc	R/W	16	0x0001	电压 Int 持续半周波数设置
2ECH	IOVCyc	R/W	16	0x0020	过流持续半周波数设置
2F0H	PeakLVL	R/W	16	0x3090	Upeak 阈值对齐关系 Bit0-15 位数据, 无符号, 当数据源选择半波峰值(22bit Q1.21)时, 阈值 LVL 与 ABS(HalfVp)[21:6]比较; 当选择半波有效值(24bit Q0.23)时, 阈值 LVL 与 ABS(HalfRms)[23:8]比较
2F4H	SagLVL	R/W	16	0x27C0	Usag 阈值对齐关系 Bit0-15 位数据, 无符号, 当数据源选择半波峰值(22bit Q1.21)时, 阈值 LVL 与 ABS(HalfVp)[21:6]比较; 当选择半波有效值(24bit Q0.23)时, 阈值 LVL 与 ABS(HalfRms)[23:8]比较
2F8H	IntLVL	R/W	16	0x0460	UINT 阈值对齐关系 Bit0-15 位数据, 无符号, 当数据源选择半波峰值(22bit Q1.21)时, 阈值 LVL 与 ABS(HalfVp)[21:6]比较; 当选择半波有效值(24bit Q0.23)时, 阈值 LVL 与 ABS(HalfRms)[23:8]比较
2FCH	IOVLVL	R/W	16	0x1200	电流过流 OVerflow 阈值 Bit0-15 位数据, 无符号, 当数据源选择半波峰值(22bit Q1.21)时, 阈值 LVL 与 ABS(HalfVp)[21:6]比较; 当选择半波有效值(24bit Q0.23)时, 阈值 LVL 与 ABS(HalfRms)[23:8]比较
300H	PeakLVLHys	R/W	16	0x00F9	Upeak 迟滞
304H	SagLVLHys	R/W	16	0x00CB	Usag 迟滞
308H	IntLVLHys	R/W	16	0x0020	UINT 迟滞
30CH	IOVLVLHys	R/W	16	0x0060	电流过流 OVerflow 迟滞
310H	UZCLVL	R/W	21	0x004600	电压通道过零阈值 和基波滤波器数据 M 点后最大值的绝对值直接全 Bit 对齐 注:电压和电流过零阈值是和基波滤波器之后的数据做比对的, 但是该节点数据并没提供寄存器。用户应用时可以在直接基波的前提下, 把半波峰值寄存器作为设定阈值的对比数据。位数一样的
314H	IZCLVL	R/W	21	0x000300	电流通道过零阈值 和基波滤波器数据 M 点后最大值的绝对值直接全 Bit 对齐
318H	TU_CCOFF_A	R/W	16	0x0000	自动温补系数 U 二次项
31CH	TU_CCOFF_B	R/W	16	0x0000	自动温补系数 U 一次项

320H	TU_CCOFF_C	R/W	16	0x0000	自动温补系数 U 常数项
324H	TI_CCOFF_A	R/W	16	0x0000	自动温补系数 I 二次项
328H	TI_CCOFF_B	R/W	16	0x0000	自动温补系数 I 一次项
32CH	TI_CCOFF_C	R/W	16	0x0000	自动温补系数 I 常数项
330H	TIn_CCOFF_A	R/W	16	0x0000	自动温补系数 In 二次项
334H	TIn_CCOFF_B	R/W	16	0x0000	自动温补系数 In 一次项
338H	TIn_CCOFF_C	R/W	16	0x0000	自动温补系数 In 常数项
33CH	Iregion0	R/W	24	0x7FFFFFFF	相位分段校正电流阈值 0Q0.23 和 RMS 有效值直接全 Bit 对齐
340H	Iregion1	R/W	24	0x7FFFFFFF	相位分段校正电流阈值 1Q0.23 和 RMS 有效值直接全 Bit 对齐
344H	Iregion0Hys	R/W	24	0x000000	相位分段校正电流阈值 0 的滞回区间
348H	Iregion1Hys	R/W	24	0x000000	相位分段校正电流阈值 1 的滞回区间
34CH	DECReset	R/W	8	00	DEC 滤波器清零寄存器,用于重置 DEC 滤波器。写 0x6A 可重置 DEC 滤波器, 写其他任意值均不操作, 读出为 0
350H	EMUWPREG	R/W	16	0x0000	EMU 的部分写保护, 该寄存器: 写入 0xA55A: 只开启 EMU&EWU&工频同步系统(不开启 Deccomp/基波谐波滤波器部分)的写使能, 读出为 1; 即 DEC 补偿/基波谐波滤波器 不可写, 其他均可写 写入 0xA5BC: 只开启 Deccomp/基波谐波滤波器部分(不开启 EMU&EWU&工频同步系统)的写使能, 读出为 2; 即 DEC 补偿/基波谐波滤波器可写, 其他均不可写 写入其他值无效, 所有寄存器均不可写入。 读出为 0
354H	SRSTREG	R/W	8	0x00	软件复位寄存器 写 0x55 用于复位校表寄存器即基地址 0x4001C000~D000 之内排布的所有寄存器配置, 包括了 EMU/eWatching/工频同步系统/全失压。 写 0xAA 则复位整个 EMU 模块即包括上述 EMU/eWatching/工频同步系统/全失压的寄存器与内部系统状态。 写入其他命令不动作。
358H	FlickerCfg		16	0x00F8	闪变配置
35CH	Reserved				Reserved
360H	Reserved				
364H	Reserved				
368H	FlickerReset	R/W	8	0	用于复位 Flicker 功能模块, 写 0x69 复位 Flicker, 写其他任意值均不操作, 读出为 0
36CH	RMSUt_THO	R/W	23	0x032000	电压矢量和阈值寄存器, 23Bit 无符号和

					RMS_Ut 等位对比, 如果 RMS_Ut 大于该值则触发 RMSUtoVIF 中断标志
370H	RMSIt_THO	R/W	23	0x0004A0	电压矢量和阈值寄存器, 23Bit 无符号和 RMS_It 等位对比, 如果 RMS_It 大于该值则触发 RMSItOVIF 中断标志
374H	Reserved				
378H	Reserved				
37CH	Reserved				
380H	EMUIE1	R/W	32	0x00000000	EMU 中断使能 1
384H	EMUIE2	R/W	24	0x00000000	EMU 中断使能 2-PeakSagIntIOV 事件专用
388H	EMUIF1	R/W	32	----	EMU 中断标志 1
38CH	EMUIF2	R/W	24	----	EMU 中断标志 2-PeakSagIntIOV 事件专用
390H	EMUIF_EnergyPOVIF	R/W	24	----	该 48 个电能溢出中断事件“或”起来运算, 只要有一个溢出就会触发 EMUIE1 的 EnergyOVIF
394H	EMUIF_EnergyNOVIF	R/W	24	----	
<b>基波谐波功能配置</b>					
71CH	FundHarCFG	R/W	16	0x0000	基波谐波功能配置
<b>EWU eWatching 在线监控功能模块 0x4001 CA00</b>					
功能暂不开放, 用户可直接调用钜泉 lib 库函数实现					
00H	EWUIE	R/W	24	0x0000	EWU 中断使能寄存器
04H	EWUIF	R/W	24	----	EWU 中断标志寄存器
<b>工频同步系统 0x4001 CC00</b>					
00H	SYNC_CFG	R/W	18	0x012000	SYNC 模块配置寄存器
04H	SFO_Value	R/W	24	0x00000000	SFO 数据, Q1.23 格式
08H	REC_CFG	R/W	13	0x0000	用于非侵入数据的路径相关配置
0CH	DFT_CFG	R/W	6	0x00	模块产出 DFT 基波信息数据的配置
10H	SYNC_DMA_rShift	R/W	28	0x00000000	SYNC 数据移位处理
14H	HpRMS_Ctl_U	R/W	24	0x00000000	电压半波有效值的校正
18H	HpRMS_Ctl_I	R/W	32	0x00000000	电流半波有效值的校正
<b>同频同步系统的锁频功能 0x4001 CE00</b>					
00H	PLL_CTRL	R/W	5	0x00	
04H	PLL_Gain	R/W	32	0x00000000	
08H	PLL_Acc	R/W	31	0x00000000	
0CH	SFO_EST	R/W	24	0x00000000	
10H	LOCK_THO	R/W	23	0x00000000	
14H	AMP_THO	R/W	25	0x00000000	
18H	TED_Amnt	R/W	8	0x00	

## 22.9 校表寄存器说明

### 22.9.1 移采样点相位校正

DECShift UaH~InH 移采样点相位校正寄存器			基地址: 0x4001 C000 偏移地址: 000/008/010/018/020/028/030					
	Bit15	14	13	12	11	10	9	Bit8
Read:					DECShift_High_Area2			
Write:								
Reset:					0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DECShift_High_Area1				DECShift_High_Area0			
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述
DECShift_High_Area	该寄存器须和 DECShift_Low_Area 拼接完成移采样点相位校正寄存器 Bit 与移动对应大拍数关系: 0000-> 0 ; 0001 -> 1 ; 0010 -> 2 ; 0011 -> 3 ; 0100 -> 4 ; 0101 -> 5 ; 0110 -> 6 ; 0111 -> 7 ; 1000 -> 8 ; 1001 -> 9 ; 1010 -> 0 ; 1011 -> 0 ; 1100 -> 0 ; 1101 -> 0 ; 1110 -> 0 ; 1111 -> 0 ;

DECShift UaL~InL 移采样点相位校正寄存器			基地址: 0x4001 C000 偏移地址: 004/00C/014/01C/024/02C/034					
	Bit23	22	21	20	19	18	17	Bit16
Read:					DECShift_Low_Area2			
Write:								
Reset:				0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	DECShift_Low_Area2		DECShift_Low_Area1					
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Area1		DECShift_Low_Area0					
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述

DECSHift_Low_Area	以 Ua 通道 Onebit 的 Shift 为例，如上寄存器拼接过程为： Iregion 切割成三段的相位校正： 第一段：DECSHift_High_Area0 (4Bit) + DECSHift_Low_Area0 (7Bit) 第二段：DECSHift_High_Area1 (4Bit) + DECSHift_Low_Area1 (7Bit) 第三段：DECSHift_High_Area2 (4Bit) + DECSHift_Low_Area2 (7Bit) 组成的 DECSHift_Area0 用于校正该相位段。
-------------------	--

## 22.9.2 ADC 直流偏置校正

ADCOffset_Ua/Ub/Uc		基地址： 0x4001 C000						
ADCOffset_Ia/Ib/Ic/In		偏移地址： 038~050H						
ADC 直流偏置校正寄存器								
	Bit23	22	21	20	19	18	17	Bit16
Read:				ADCOffset[21:16]				
Write:								
Reset:				0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADCOffset[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADCOffset[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述
ADCOffset	该寄存器的值，只有在高通关闭时才可以写，否则会立刻以 EMUCLK/OSR 的速度被 DEC 过 LPF 数据刷掉。字长和 SPL 一样，都是 22Bit $SPLUa' = SPLUa - AdcOffset\_Ua \ll 2$ DC_Var 参数：更新速度为 $fs = EMUCLK/OSR$ 当 HPF_EN = 1，即高通使能，零漂等参数是实时送到 DC_Var 计量参数寄存器的。同时也会以 FS 更新的频率参与零漂校正。当用于直流应用时，HPF_EN = 0，此时 ADCOffset 写入的值则刷进内部数据。

## 22.9.3 通道温度补偿增益

Tgain_U/I/In		基地址： 0x4001 C000						
通道温度补偿增益		偏移地址： 054/058/05C						
	Bit15	14	13	12	11	10	9	Bit8
Read:	Tgain_U/I/In[15:8]							

Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Tgain_U/I/In[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述
Tgain	三路电压共用一个温度补偿增益 UGain 三路火线电流共用一个温度补偿增益 IGain 零线电流专用一个温度补偿增益 InGain Tgain 在 EMU 算法节点上位于高通之后，通道 Gain 之前。该寄存器写值会直接影响有效值/功率/电能等参数。 16 位有符号数，最高位为符号位，校正公式均为如下 $Data' = Data * (1 + Gain / 32768)$ 或 $Data' = Data * (1 + (Gain - 65536) / 32768)$

## 22.9.4 通道增益

<b>Gain_Ua/Ub/Uc/Ia/Ib/Ic/In</b>			<b>基地址: 0x4001 C000</b>					
<b>通道增益</b>			<b>偏移地址: 060/064/068/06C/070/074/078</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Gain [15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Gain [7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述
Gain	7 路 ADC 通道都有各自独立的通道增益 通道 Gain 在 EMU 算法节点上位于高通并通道温度补偿 TGain 之后。该寄存器写值会直接影响有效值/功率/电能等参数。 16 位有符号数，最高位为符号位，校正公式均为如下 $Data' = Data * (1 + Gain / 32768)$ 或 $Data' = Data * (1 + (Gain - 65536) / 32768)$

## 22.9.5 数字移位放大

<b>Digital_gain</b>	<b>基地址: 0x4001 C000</b>
<b>数字移位放大</b>	<b>偏移地址: 07C</b>

	Bit15	14	13	12	11	10	9	Bit8
Read:			DGUc1	DGUc0	DGUb1	DGUb0	DGUa1	DGUa0
Write:								
Reset:			0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DGIIn1	DGIIn0	DGIc1	DGIc0	DGIb1	DGIb0	DGIa1	DGIa0
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述
DG	数字 Digital 的放大倍数 00/01/10/11：对应 1 倍/2 倍/4 倍/8 倍

## 22.9.6 功率增益校正

功率增益校正	A 相	偏移地址	B 相	偏移地址	C 相	偏移地址
全波有功	All_GPa	080H	All_GPb	084H	All_GPc	088H
全波无功	All_GQa	08CH	All_GQb	090H	All_GQc	094H
全波视在	All_GSa	098H	All_GSb	09CH	All_GSc	0A0H
基波有功	Fund_GPa	110H	Fund_GPb	114H	Fund_GPc	118H
基波无功	Fund_GQa	11CH	Fund_GQb	120H	Fund_GQc	124H
基波视在	Fund_GSa	128H	Fund_GSb	12CH	Fund_GSc	130H
谐波有功	Har_GPa	190H	Har_GPb	194H	Har_GPc	198H

位名称	描述
GP / GQ / GS	<p>功率增益寄存器 16bit 有符号，bit15 为符号位。分别作用于各自功率寄存器与的增益校正公式：以 P 为例，<math>P' = P * (1 + GP / 32768)</math> 或者 <math>P' = P * (1 + (GP - 65536) / 32768)</math>，其中 P 为校正前寄存器数值，P' 为校正后数值，Q 和 S 同理</p> <p>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> <p><math>P_{gain} = -Err\% / (1 + Err\%)</math></p> <p>如果 P<sub>gain</sub> 为正数，则 GP 的写入值为：P<sub>gain</sub> * 32768</p> <p>如果 P<sub>gain</sub> 为负数，则 GP 的写入值为：65536 + P<sub>gain</sub> * 32768</p>

## 22.9.7 PQ 方式相位校正

功率相位校正	A 相	偏移地址	B 相	偏移地址	C 相	偏移地址
全波功率	All_GphsA0	0A4H	All_GphsB0	0B0H	All_GphsC0	0BCH
	All_GphsA1	0A8H	All_GphsB1	0B4H	All_GphsC1	0C0H
	All_GphsA2	0ACH	All_GphsB2	0B8H	All_GphsC2	0C4H

基波功率	All_GphsA0	134H	Fund_GphsB0	140H	Fund_GphsC0	14CH
	Fund_GphsA1	138H	Fund_GphsB1	144H	Fund_GphsC1	150H
	Fund_GphsA2	13CH	Fund_GphsB2	148H	Fund_GphsC2	154H
位名称	描述					
Gphs	PQ 方式相位校正寄存器为 16bit 有符号, bit15 为符号位。分别作用于全波/基波的有功、无功功率的相位校正。 相位校正必须和相位分段功能配合使用。如果用户不使用相位分段, 则给三段 Gphs 写同样的值即可。 校正公式: $\text{PowerP}' = \text{PowerP} + \text{PowerQ} * \text{Gphs} / 32768$ $\text{PowerQ}' = \text{PowerQ} - \text{PowerP} * \text{Gphs} / 32768$ 其中 P 和 Q 为校正前寄存器数值, P'和 Q'为校正后数值, 外加信号 0.5L 时校正。					

## 22.9.8 功率偏置校正

功率偏置校正	A 相	偏移地址	B 相	偏移地址	C 相	偏移地址
全波功率	All_PowerOffset_Pa	0CCH	All_PowerOffset_Pb	0D0H	All_PowerOffset_Pc	0D4H
	All_PowerOffset_Qa	0D8H	All_PowerOffset_Qb	0DCH	All_PowerOffset_Qc	0E0H
基波功率	Fund_PowerOffset_Pa	15CH	Fund_PowerOffset_Pb	160H	Fund_PowerOffset_Pc	164H
	Fund_PowerOffset_Qa	168H	Fund_PowerOffset_Qb	16CH	Fund_PowerOffset_Qc	170H
基波功率	Har_PowerOffset_Pa	19CH	Har_PowerOffset_Pb	1A0H	Har_PowerOffset_Pc	1A4H
位名称	描述					
PowerOffset	PowerOffset 和 PowerP/Q 是低对齐关系即对应 PowerP/Q 的 bit[23:0] 校正公式: (全波 A 相有功为例) $\text{All\_Power\_Pa}' = \text{All\_Power\_Pa} - \text{All\_PowerOffset\_Pa}$					

## 22.9.9 有效值偏置校正

通道	全波	偏移地址	基波	偏移地址	谐波	偏移地址
UA	All_RmsOffset_Ua	0E4H	Fund_RmsOffset_Ua	174H	Har_RmsOffset_Ua	1A8H
UB	All_RmsOffset_Ub	0E8H	Fund_RmsOffset_Ub	178H	Har_RmsOffset_Ub	1ACH
UC	All_RmsOffset_Uc	0ECH	Fund_RmsOffset_Uc	17CH	Har_RmsOffset_Uc	1B0H
IA	All_RmsOffset_Ia	0F0H	Fund_RmsOffset_Ia	180H	Har_RmsOffset_Ia	1B4H
IB	All_RmsOffset_Ib	0F4H	Fund_RmsOffset_Ib	184H	Har_RmsOffset_Ib	1B8H

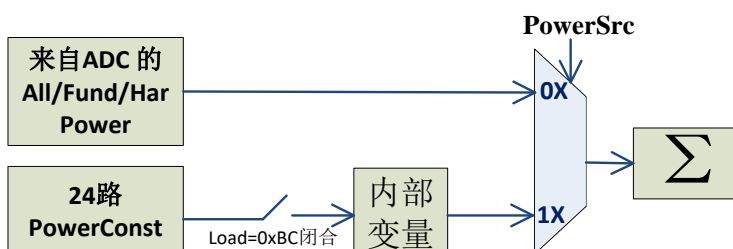
IC	All_RmsOffset_Ic	0F8H	Fund_RmsOffset_Ic	188H	Har_RmsOffset_Ic	1BCH
IN	All_RmsOffset_In	0FCH	Fund_RmsOffset_In	18CH	Har_RmsOffset_In	1C0H
电压矢量和	RMSOffset_Ut	100H	RMSOffset_Ut	100H	(全波/基波共用电压矢量和有效值偏置校正)	
电流矢量和	RMSOffset_It	104H	RMSOffset_It	104H	(全波/基波共用电流矢量和有效值偏置校正)	
位名称	描述					
RMSOffset	RMSOffset 寄存器为无符号数据。其作用于各路信号自乘后直接减去该寄存器。					

已知：输入信号为 0 的时候，读取有效寄存器的值 rms

计算公式： $RMSOffset = (rms^2) / (2^7)$ 。

## 22.9.10 常数自定义功率寄存器 1~24

通道	偏移地址
第 X 路电能通路的前端 常数自定义功率 PowerConst X = 1 ~ 24	$1C4H + 004H * (X - 1)$ 即 1C4H ~ 220H

位名称	描述
PowerConst	<p>PowerConst 寄存器共 24 个，即第 1~24 电能通路的前端输入，均可选择为用户自定义的常数功率形式。</p> <p>格式同功率寄存器，Bit0-Bit30 数据，Bit31 为符号位。</p> <p>注 和计量参数 Power 寄存器的关系： PowerConst 寄存器的对齐格式为计量参数的 Power 寄存器 &gt;&gt; 1。</p> <p>24 路 PowerConst 的功能逻辑图如下：</p> 

## 22.9.11 电能偏置功率寄存器

通道	偏移地址
谐波通道专用的电能偏置功率 A/B/C/合相 均有各自的电能偏置功率	224H ~ 240H

位名称	描述

AccuOffset	电能偏置功率共有 8 个，并且只存在于谐波电能通路的 21~24 这四条电能通路的正反向两个分支上。分别为： A 相正向、A 相反向，B 相正向、B 相反向，C 相正向、C 相反向，合相正向、合相反向 AccuOffset 的格式为：Bit0-Bit30 数据，Bit31 为符号位，对齐方式和 PowerConst 一致 正向 AccuOffset 只允许写正值，反向 AccuOffset 只允许写负值。
------------	---

## 22.9.12 常数自定义电压

<b>UconstA/B/C</b>		<b>基地址： 0x4001 C000</b>		
常数电压有效值		<b>偏移地址： 244/248/24C</b>		
	Bit23	22~1		Bit0
Read:	<b>Uconst [23:0]</b>			
Write:				
Reset:	0	..000..		0
<b>位名称</b>		<b>描述</b>		
Uconst		A/B/C 各相常数电压有效值的作用为，替代三相电压 ADC 通道计算出的有效值 All_RMS_Ux，去参与计算各相的视在功率寄存器并累计视在电能(只全波)		

## 22.9.13 过零点配置寄存器

<b>ZXLostCFG</b>		<b>基地址： 0x4001 C000</b>						
		<b>偏移地址： 250</b>						
	Bit23	22	21	20	19	18	17	Bit16
Read:	NumFund[5:0]						ZCNumMin[8:7]	
Write:								
Reset:	1	0	1	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ZCNumMin[6:0]							ZCNumMax[8]
Write:								
Reset:	1	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ZCNumMax[7:0]							
Write:								
Reset:	1	1	0	0	0	0	0	0

<b>位名称</b>	<b>描述</b>
ZCNumMax	用户需保证 ZCNumMin < ZCNumMax 过零超时事件点数
ZCNumMin	过零欠时事件点数

NumFund	过零有效事件判断点数																																
备注 1	用户注意： <ol style="list-style-type: none"> <li>1) 过零有效还需组合 FundZCDelay, ZCNumMin, ZCNumMax, NumFund 这四个寄存器。</li> <li>2) 过零有效作为频率、相序、角度等计算的精确度依据。需严格按照推荐值填写。过零对 SAG、PEAK、UINT、IOV 也会有间接影响。</li> <li>3) 当用户关于过零点相关有特殊应用需求，可调整该寄存器组合。</li> <li>4) 在 25.6K 的采样率下(1.6384M/OSR64)，ZCNumMin, ZCNumMax 一定要配置为各乘以 2，以防过零超时和欠时的误触发</li> </ol>																																
备注 2	用户推荐：用户可根据 Fs 值与计量系统频率填写如下值(十进制)： 注 $F_s = \text{emuclk} / \text{OSR}$ ，例 $F_s = 12800 = 1.6384\text{M} / \text{OSR}128$ 或 $819.2\text{K} / \text{OSR}64$ <table border="1" style="margin-top: 10px;"> <thead> <tr> <th rowspan="3">Fs 频率</th> <th colspan="4">FundZCDelay</th> <th rowspan="3">ZCNumMin</th> <th rowspan="3">ZCNumMax</th> <th rowspan="3">NumFund</th> </tr> <tr> <th colspan="2">过零使用全波数据</th> <th colspan="2">过零使用基波数据</th> </tr> <tr> <th>50Hz</th> <th>60Hz</th> <th>50Hz</th> <th>60Hz</th> </tr> </thead> <tbody> <tr> <td>12800</td> <td>77</td> <td>75</td> <td>102</td> <td>89</td> <td>64</td> <td>192</td> <td>40</td> </tr> <tr> <td>25600</td> <td>175</td> <td>168</td> <td>210</td> <td>180</td> <td>128</td> <td>384</td> <td>40</td> </tr> </tbody> </table>	Fs 频率	FundZCDelay				ZCNumMin	ZCNumMax	NumFund	过零使用全波数据		过零使用基波数据		50Hz	60Hz	50Hz	60Hz	12800	77	75	102	89	64	192	40	25600	175	168	210	180	128	384	40
Fs 频率	FundZCDelay				ZCNumMin	ZCNumMax				NumFund																							
	过零使用全波数据		过零使用基波数据																														
	50Hz	60Hz	50Hz	60Hz																													
12800	77	75	102	89	64	192	40																										
25600	175	168	210	180	128	384	40																										

## 22.9.14 起动潜动阈值

PStart/ QStart/HarStart 起动潜动阈值设置寄存器	基地址： <b>0x4001 C000</b> 偏移地址： <b>254/258/25C</b>
	<b>Bit15...Bit0</b>
Read:	PQStart[15:0]
Write:	
Reset:	<b>PStart: 0x0040; QStart: 0x0080; SStart: 0x0040</b>

位名称	描述
PStart QStart HarStart	<p><b>PStart</b> Default 值：0x0040; <b>QStart</b> Default 值：0x0080; <b>SStart</b> Default 值：0x0040</p> <p>16 位无符号数，Bit0-Bit15 数据，</p> <p>P 格式与对齐关系：</p> <p>功率方式与 PowerP/Q/S 绝对值的 bit8—bit23 进行比较，有效值方式与 Fund_RMS/All_RMS 值的 bit0—bit15 低位对齐进行比较。</p> <p>即若选择功率方式，则将该 Pstart 左移 8bit 与 Power 对比。若选择有效值方式，则将该 Pstart 直接与 All/Fund_RMS 对比</p>

举例：标准源施加额定电压  $U_n$ , 额定电流  $I_b$ , 功率因数 1.0, 启动电流系数为  $z\%$ , 一般推荐客户预留启动值余量，额外乘以 0.6 倍数。

如果为功率启动方式则：

$$PStart = \frac{0.6 \times z\% \times U_n \times I_b \times HFConst \times EC}{343.323 \times 2^8}$$

如果为电流有效值启动方式则：

$$P_{Start} = \frac{0.6 \times z\% \times IB}{K_{Irms}}$$

### 22.9.15 AVG 平均模块

AVG_Energy AVG_PowerREG AVG_RMS AVG_RMST		基地址: 0x4001 C000 偏移地址: 260 / 264 / 268 / 26CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:	x	x	AVG[13:8]					
Write:								
Reset:	0	0	0	0	x	0	x	x
	Bit7	6	5	4	3	2	1	Bit0
Read:	AVG[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述
AVG	寄存器上限 0x2C88, 下限 0x002D 写入大于 0x2C88 强制等于 0x2C88, 小于 0x002D 强制等于 0x 002D AVG_Energy: 瞬时功率经过该平均模块后得到平均功率, 接入 24 路电能模块进行电能累计, 默认值 0x100. AVG_PowerREG: 瞬时功率经过该平均模块后得到平均功率, 接入计量参数: 功率寄存器, 默认值 0xA00. AVG_RMS: 各通道采样数据自乘后的瞬时数据经过该平均模块后得到平均数据, 最终结果接入计量参数: 有效值寄存器, 默认值 0xA00. AVG_RMST: 各通道采样数据做矢量和运算并自乘后的瞬时数据经过该平均模块后得到平均数据, 最终结果接入计量参数: 矢量和有效值寄存器, 默认值 0xA00. 例 EMUCLK=819.2kHz, OSR=64 时, 寄存器更新频率: EMUCLK/OSR/ AVG_PowerREG=5Hz 而同时用于电能累计的平均功率的频率: EMUCLK/OSR/ AVG_Energy=50Hz
备注:	用户需先确定要平均的点数再开启 ADC, 在正常计量过程中尽量不要频繁切换 AVG 模块值。

### 22.9.16 高频脉冲 HFConst

HFConstAF HFConstHar		基地址: 0x4001 C000 偏移地址: 270 / 274H						
	Bit23	22	21	20	19	18	17	Bit16

Read:					HFConst[19:16]			
Write:					HFConst[19:16]			
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	HFConst[15:8]							
Write:	HFConst[15:8]							
Reset:	0	0	0	1	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	HFConst[7:0]							
Write:	HFConst[7:0]							
Reset:	0	0	0	0	0	0	0	0

位名称	描述
<b>HFConst</b>	20Bit 无符号, 默认值为 0x1000 HFConst 格式为:Bit0-Bit19 为数据位, 无符号, Bit20~Bit23 恒 0。 HFConstAF 用于全波和基波电能通路的快速脉冲计数 CFxCnt 比较。当全波和基波电能通路的 CFxCnt 累计溢出等于 HFConstAF 时则发一个电能脉冲给后级电能寄存器与二级脉冲单元。 HFConstHar 只用于谐波电能通路的快速脉冲计数 CFxCnt 比较。当谐波电能通路的 CFxCnt 累计溢出等于 HFConstHar 时则发一个电能脉冲给后级电能寄存器与二级脉冲单元。

### 22.9.17 电能脉冲分辨率 DIV

<b>AFDIV</b>	<b>基地址: 0x4001 C000</b>							
<b>HarDIV</b>	<b>偏移地址: 278</b>							
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	DIV[14:8]						
Write:		DIV[14:8]						
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DIV[7:0]							
Write:	DIV[7:0]							
Reset:	0	0	0	0	0	0	0	1

位名称	描述
<b>DIV</b>	电能分辨率寄存器。即二级 HFConst, 默认为 0x0001 无符号数; 有效 15Bit, 即 Bit0-14 位数据, Bit15 恒 0 默认状态下, 电能寄存器和电能脉冲是同步的: 即电能寄存器+1 的同时, 电能脉冲也会发一个到后级脉冲输出口。 当设置 AFDIV / HarDIV 为 N, 则电能寄存器每次+1 直到累加到 N 时, 电能

	脉冲才会发一个到后级脉冲口。其中 AFDIV 用于全波基波电能的分辨率扩展，HarDIV 用于谐波电能的分辨率扩展 电能分辨率 DIV，用户不可写入 0。
--	--

### 22.9.18 采样通道互换与反向

ChannelSel		基地址: 0x4001 C000 偏移地址: 280						
	Bit23	22	21	20	19	18	17	Bit16
Read:		ANSI_Mode		UcRev	UbRev	UaRev	InRev	IcRev
Write:								
Reset:		0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	IbRev	IaRev	UcSel		UbSel		UaSel	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	InSel		IcSel		IbSel		IaSel	
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述
IxSel/UxSel	IaSel: 2Bit; 00/01/10/11 Ia 通道选择 Ia/Ib/Ic/In IbSel: 2Bit; 00/01/10/11 Ib 通道选择 Ib/Ic/In/Ia IcSel: 2Bit; 00/01/10/11 Ic 通道选择 Ic/In/Ia/Ib InSel: 2Bit; 00/01/10/11 In 通道选择 In/Ia/Ib/Ic UaSel: 2Bit; 00/01/10/11 Ua 通道选择 Ua/Ub/Uc/Ua UbSel: 2Bit; 00/01/10/11 Ub 通道选择 Ub/Uc/Ua/Ub UcSel: 2Bit; 00/01/10/11 Uc 通道选择 Uc/Ua/Ub/Uc
IxRev/UxRev	IaRev: 1Bit; 0/1 Ia 通道 DEC 之后的数据乘以 1/-1 IbRev: 1Bit; 0/1 Ib 通道 DEC 之后的数据乘以 1/-1 IcRev: 1Bit; 0/1 Ic 通道 DEC 之后的数据乘以 1/-1 InRev: 1Bit; 0/1 In 通道 DEC 之后的数据乘以 1/-1 UaRev: 1Bit; 0/1 Ua 通道 DEC 之后的数据乘以 1/-1 UbRev: 1Bit; 0/1 Ub 通道 DEC 之后的数据乘以 1/-1 UcRev: 1Bit; 0/1 Uc 通道 DEC 之后的数据乘以 1/-1
ANSI_Mode	00 通道不变 01 $U_b = -U_a - U_c$ , 10 $U_b = -U_a$ , 11 $U_a = U_a - U_b$ 同时 $U_c = U_c - U_b$ 注意 ANSI_Mode 控制的通道切换的节点在 ADC 的 Gain 之后

备注	三路电压之间可以互相切换，四路电流之间可以互相切换，但是电压和电流通道之间不能互相切换。
----	--

## 22.9.19 模拟 PGA 放大配置

PGA CON		基地址: 0x4001 C000						
模拟 PGA 倍数配置		偏移地址: 284						
	Bit15	14	13	12	11	10	9	Bit8
Read:						PGAU[1:0]		PGAIn[2]
Write:								
Reset:						0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PGAIn[1:0]		PGA Ic[1:0]		PGA Ib[1:0]		PGA Ia[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述
PGAU[1:0]	三路电压通道 ADC PGA 控制信号; default=00 00 -> x 1 01 -> x 2 10 -> x 8 11 -> x 8
PGA Ia/Ib/Ic[1:0]	三路火线电流通道 ADC PGA 控制信号; default=00 00 -> x 1 01 -> x 2 10 -> x 8 11 -> x 16
PGA In[2:0]	零线电流通道 ADC PGA 控制信号; default=000 000 -> x 1 001 -> x 4 010 -> x 8 011 -> x 16 100 -> x 32 101 -> x 2 110 -> x 4 111 -> x 24
备注	电流 PGA Ia/Ib/Ic/In (PGA CON 寄存器) 只有 In 通道才有 3Bit 配置可到 24/32 倍。PGA Ua/Ub/Uc 三路电压共用一个 PGAU 控制

## 22.9.20 ADC 配置寄存器

ADCCFG		基地址: 0x4001 C000 偏移地址: 288						
	Bit23	22	21	20	19	18	17	Bit16
Read:					ZcSrSel	FreqSel	MODE3P3	DMASrc_Sel
Write:								
Reset:					0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	Auto_Tc_	WaveSel[1:0]		ZCd[1:0]		SPL[2:0]		
Write:	EN							
Reset:	0	0	0	0	0	0	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	ZCINTSrc_Sel[1:0]		EMUCIk_Sel	Deccomp	OSR128	HPFON_In	HPFON_I	HPFON_U
Write:				p_EN				
Reset:	0	0	0	0	0	1	1	1

位名称	描述
HPFON_U	三路U通道高通滤波器开关位 0: 关闭; 1: 开启 (default)
HPFON_I	三路 I 通道高通滤波器开关位 0: 关闭; 1: 开启 (default)
HPFON_In	零线电流 In 通道高通滤波器开关位 0: 关闭; 1: 开启 (default)
OSR128	正常计量模式下 OSR 选择位: 0: OSR=64 (default); 1: OSR =128
Deccomp_EN	DEC 补偿滤波器使能 0 : 不使能 (default) ; 1 : 使能 DEC 补偿滤波器的作用是用于补偿外部 RC 以及 DEC 滤波器对于高频信号的衰减, 补偿滤波器系数寄存器可配, 七路补偿滤波器受同一个使能位和同一套系数控制
EMUCIk_Sel	EMU 时钟控制位: 0: femu 为 819.2K (Default) 1: femu 为 1.6384M
ZCINTSrc_Sel [1:0]	过零中断事件的数据源选择 00: 高通后的数据(Default) 01: 高通前的数据 1X: 基波滤波器之后的数据
SPL[2:0]	SPL 波形更新频率选择控制位 (Default 12.8k) (该 Bit 涉及非侵入应用, 一定注意波形更新频率) 当 Femu=819.2KHz、OSR=64 或 Femu=1.6384MHz、OSR=128 时, 选择的

	频率如下： <table border="1" style="margin-left: 20px;"> <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> 注：当 femu=1.6384MHz、OSR=64 时，选择的波形采样频率在上表的基础上乘以 2。	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)																						
ZCd[1:0]	电压电流过零中断方式选择控制位 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>ZCD1</th> <th>ZCD0</th> <th>电压电流过零中断方式</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>正向过零中断</td> </tr> <tr> <td>0</td> <td>1</td> <td>负向过零中断</td> </tr> <tr> <td>1</td> <td>X</td> <td>双向过零中断</td> </tr> </tbody> </table>	ZCD1	ZCD0	电压电流过零中断方式	0	0	正向过零中断	0	1	负向过零中断	1	X	双向过零中断												
ZCD1	ZCD0	电压电流过零中断方式																							
0	0	正向过零中断																							
0	1	负向过零中断																							
1	X	双向过零中断																							
WaveSel[1:0]	ADC 波形数据(SPL 数据)的源头选择控制位 00：选择 ADC 波形数据为未经高通的原始数据；(Default) 01：选择 ADC 波形数据为经过高通的数据； 1X：选择 ADC 波形数据为 U/Gain 后面；																								
Auto_Tc_EN	自动温度补偿控制位 0：自动温度补偿功能关闭 (default) 1：自动温度补偿功能开启，开启后芯片会根据已设定的自动温补增益公式在不同温度点补偿计量通道的 Tgain（自动，用户也可手动补偿，但需要关闭该功能）。																								
DMA Src_Sel	EMU 通用 DMA Request Source 源选择 Bit， 0：为 SPL 通道数据的 Request 信号，注意是波形寄存器，其受控 SPL[2:0] 影响 1：QuartRmsIns 通道 Request 信号																								
MODE3P3	0：系统保持为三相四线模式，即不使能三相三线模式(Default) 1：系统配置为三相三线模式																								
FreqSel	系统频率选择控制位 0：选择当前计量系统为 50Hz (Default) 1：选择当前计量系统为 60Hz																								
ZcSrSel	过零数据源和半波峰值数据源 选择位 0：全波的数据 (Default) 1：基波的数据																								
备注	三路电压共用一个高通开关，三路火线电流共用一个高通开关																								

### 22.9.21 ADC 相关功能开关寄存器

ADCCON	基地址： 0x4001 C000 偏移地址： 28C
--------	-------------------------------

	Bit23	22	21	20	19	18	17	Bit16
Read:		ADCEn_A	ADCEn_A	ADCEn_A	ADCEn_A	ADCEn_A	ADCEn_A	ADCEn_A
Write:		NA_Uc	NA_Ub	NA_Ua	NA_In	NA_Ic	NA_Ib	NA_Ia
Reset:		0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	Reserved	PeakSagSrc	PeakEn	SagEn	IntEn	IovEn	OSR_CT2	Phase_Sel
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CTdetEn	ADC_En_Uc	ADC_En_Ub	ADC_En_Ua	ADC_En_In	ADC_En_Ic	ADC_En_Ib	ADC_En_Ia
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述
ADC_En_Ia/Ib/Ic/In /Ua/Ub/Uc	Ia/Ib/Ic/In的ADC通道使能位(数字部分) 0: 关闭 (Default) ,1: 开启 注意该ADC_En_Ix/Ux作用于数字设计部分的Gate与相关使能。该寄存器并不能控制模拟部分的ADC转换器模块的使能! ADC转换模块的使能开关, 由Bit22~Bit16的ADCEn_ANA_Ux/Ix独立控制。即: ADC_En_Ix/Ux 作用于数字, ADCEn_ANA_Ux/Ix作用于模拟, 二者独立。 <b>用户使用时必须把ADC_En和ADCEn_ANA同时打开!</b>
CTdetEn	CT 二次侧检测功能使能 =0: 关闭 CT 二次侧检测功能, In 通道用于检测零线电流; =1: 开启CT二次侧检测功能 CTdetEn=1时, 计量参数寄存器064H的All_RMS_In寄存器存放, 复用为CT二次侧的数据。格式为22Bit低对齐, 并取 1024点绝对值的平均值
Phase_Sel	相位分段校正电流有效值源头选择 0: 全波电流有效值 1: 基波电流有效值
OSR_CT2	CT 二次侧的 OSR 0: 16 (Default) 1: 32
PeakEn SagEn IntEn IovEn	电压通道 Peak/Sag/INT 功能使能, 电流通道 OverFlowing 使能 0: 不使能(Default) 1: 使能
PeakSagSrc	Peak、Sag、INT、IOV 这四种事件的数据源头 0: 选择半波峰值做数据源头 (default) 1: 选择半波有效值做数据源头

ADCEn_ANA_Ia/Ib/Ic/In/Ua/Ub/Uc	Bit22~Bit16 ADCEn_ANA_Ux/Ix 独立控制 ADC 转换模块的使能开关
备注:	1.在 In 复用为 CT2 功能时(CT2En=1),In 的 DECShift 需要写 0。

## 22.9.22 EMU 配置寄存器

EMUCFG		基地址: 0x4001 C000 偏移地址: 290						
	Bit23	22	21	20	19	18	17	Bit16
Read:		SQRMS_S	AngleBaseSel[2:0]			AutoAngBase	PtCfgC	
Write:		rc						
Reset:		0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	PtCfgB		PtCfgA		UtCfgC	UtCfgB	UtCfgA	ItCfgN
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ItCfgC	ItCfgB	ItCfgA	Reserved	PowerSCal_Sel	FundLineU_En	UtFund_En	ItFund_En
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述
ItFund_En	合相电流有效值源头选择 0: 选择全波电流计算合相有效值 (Default) 1: 选择基波电流计算合相有效值
UtFund_En	合相电压有效值源头选择 0: 选择全波电压计算合相有效值 (Default) 1: 选择基波电压计算合相有效值
FundLineU_En	线电压有效值寄存器计算源选择: 0: 选择全波电压信号计算线电压 (Default) 1: 选择基波电压信号计算线电压
PowerSCal_Sel	视在功率计算方式选择: 0: PQS 方式计算视在功率 1: RMS 方式计算视在功率
ItCfgA/ItCfgB/ ItCfgC/ItCfgN	选择电流通道参与合相电流有效值的计算 0: 参与 (Default) 1: 不参与
UtCfgA/ UtCfgB/UtCfgC	选择电压通道参与合相电压有效值的计算 0: 参与 (Default) 1: 不参与

PtACfg/ PtBCfg/PtCCfg	选择某单相功率参与合相功率的计算(包括全/基/谐波, 有功无功视在) 00: 直接参与 01: 功率值取反后参与 1X: 不参与
AutoAngBase	基准信号自动切换位 0: 基准信号手动选择 1: 基准信号自动切换 只有在 AngBaseSel 配置的通道过零无效时才自动切换, 自动切换时 AngBaseSel[2:0] 寄存器的值不更新, 自动切换的优先级从高到低为 ua/ub/uc/ia/ib/ic。 系统会把当前所有的基准信号通道放到信号状态 SignalStatus 寄存器 Bit.AngleBaseFlag
AngleBaseSel[2:0] ]	角度基准信号选择位 000/001/010/011/100/101/11X : ua/ub/uc/ia/ib/ic/in
SQRMS_Src	瞬时有效值平方参数寄存器的数据来源 0: 全波通道数据自乘后(Default) 1: 基波通道数据自乘后
备注	1) 合相电压电流有效值 (计量) 寄存器, 基波和全波有效值共用。 2) 合相电压电流有效值的更新速度由 AVG_RMST 单独控制 3) 功率寄存器值更新的前提: 该相的 ADC U 和 I 同时打开, 比如 ABC 的 U 开关全关, ABCN 的 I 开关全开, 那么这个时候 ABCN 的电流有效值有更新的, 但是功率都不更新。 4) AutoAngBase 如果选择为 0 手动选择; 则按照 AngleBaseSel[2:0]作为参考基准信号 如果选择为 1, 则内部自动切换基准信号。

### 22.9.23 Rosi 与常数电压寄存器

FilterCon		基地址: 0x4001 C000 偏移地址: 294							
	Bit23	22	21	20	19	18	17	Bit16	
Read:				FundZCDelay[8:4]					
Write:									
Reset:				0	0	1	0	0	
	Bit15	14	13	12	11	10	9	Bit8	
Read:	FundZCDelay[3:0]				UconstC_En	UconstB_En	UconstA_En	Reserved	
Write:									
Reset:	1	1	0	1	0	0	0	0	
	Bit7	6	5	4	3	2	1	Bit0	
Read:	RmsLpf_En	PowerLpf_En	RosiShift[1:0]		Rosi_En_In	Rosi_En_Ic	Rosi_En_Ib	Rosi_En_Ia	
Write:									

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

位	功能描述															
Rosi_En_Ia/Ib/Ic/In	Ia/Ib/Ic/In的罗氏线圈使能位 0: 关闭电流通道Ia/Ib/Ic/In罗氏线圈功能(default) 1: 使能电流通道Ia/Ib/Ic/In罗氏线圈功能															
RosiShift[1:0]	两路电流罗氏线圈增益控制档															
	<table border="1"> <thead> <tr> <th>RosiShift[1]</th> <th>RosiShift[0]</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>	RosiShift[1]	RosiShift[0]	Rosi 增益	0	0	1 (default)	0	1	2	1	0	4	1	1	8
	RosiShift[1]	RosiShift[0]	Rosi 增益													
	0	0	1 (default)													
	0	1	2													
1	0	4														
1	1	8														
RmsLpf_En	有效值低通使能控制位 0: 关闭有效值通道的低通滤波器 1: 开启有效值通道的低通滤波器(default)															
PowerLpf_En	功率低通使能控制位 0: 关闭功率通道的低通滤波器 1: 开启功率通道的低通滤波器(default)															
UconstA/B/C_En	常数电压有效值使能位 0: 不使能(default) 1: 使能 当使能相应的Uconstx_En时, 该相将以Uconst设定的值和IRMS相乘去计算视在功率。注意只作用于全波视在。															
FundZCDelay	FundZCDelay和ZCNumMin, ZCNumMax NumFund组合应用于过零有效															
备注 1	<p>UConst、ADCEN和PowerP/Q/S的Gating关系如下: (x为A/B/C的唯一一种) (Uconstx_En   ADC_En_Ux) &amp; ADC_En_Ix 作为PowerPx/Qx/Sx等x相参数的时钟控制信号。</p> <p>RMSUx只受控于其ADC_En_Ux, RMSIx只受控于其ADC_En_Ix。</p> <p>Clear关系: Px/Qx的寄存器clear受(ADC_En_Ux &amp; ADC_En_Ix)决定, 即只要U和I有一个ADC_EN=0, 则Px/Qx寄存器就会被Clear为0。</p> <p>但是Sx寄存器只受ADC_En_Ix决定, 即只要ADC_EN_Ix=1, 不管ADC_En_U是否为1, Sx寄存器均不会被Clear为0</p>															
备注 2	用户在使用UConst时, 需要先将StartCFG.Bit[1:0]配置为11即有效值起动方式2, 同时需要将视在功率的计算方式配置为RMS方式, 否则将无法获得UConst*IRMS形式的合相视在电能。															

## 22.9.24 常规滤波器系数配置

<b>FilterCFG</b> 滤波器系数控制寄存器	<b>基地址: 0x4001 C000</b> <b>偏移地址: 298</b>
--------------------------------	---

	Bit23	22	21	20	19	18	17	Bit16
Read:						PH90Ctrl[1:0]		LPFCtrl[11]
Write:								
Reset:					0	1	0	1
	Bit15	14	13	12	11	10	9	Bit8
Read:	LPFCtrl[10:3]							
Write:								
Reset:	0	1	1	0	1	1	0	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	LPFCtrl[2:0]			RosiCtrl[1:0]		HpfCtrl[2:0]		
Write:								
Reset:	1	0	1	1	0	1	0	1

位	功能描述
HpfCtrl[2:0]	高通滤波器系数 默认‘101’
RosiCtrl[1:0]	罗氏功能滤波器系数 默认‘10’
LPFCtrl[11:0]	低通滤波器系数 默认‘101101101101’
PH90Ctrl[1:0]	基波无功90度移相滤波器系数 默认‘10’
备注	Normal模式下使用时需保持滤波器寄存器为默认，低频模式下使用时需参考HT762X低频计量模式的相关推荐配置

## 22.9.25 起动潜动配置

<b>StartCFG</b>		<b>基地址： 0x4001 C000</b>						
		<b>偏移地址： 29C</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:					HarStart_CFG	StartSrc_CFG	Start_CFG[1:0]	
Write:								
Reset:					0	0	0	0

位	功能描述
Start_CFG[1:0]	起动潜动定义：(Default 00) 00 : 功率起动方式1 01 : 功率起动方式2 10 : 有效值起动方式1 11 : 有效值起动方式2
StartSrc_CFG	0 : 平均功率(Default),

	1 : 瞬时功率 (注意电流方式不可选瞬时)
HarStart_CFG	谐波起动潜动可选功率/有效值 0 : 功率(default) 1 : 有效值

## 22.9.26 模拟参数配置寄存器

ANACFG1			基地址: 0x4001 C000 偏移地址: 2A0					
	Bit23	22	21	20	19	18	17	Bit16
Read:		Vref_En	ldo_emu_en	UADC_BIAS<2:0>			CLKCfg[4:3]	
Write:								
Reset:		0	0	0	1	1	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	CLKCfg[2:0]			Reserved				
Write:								
Reset:	0	1	0	0	0	0	0	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	Reserved		PGA_BIAS<2:0>			IADC_BIAS<2:0>		
Write:								
Reset:	0	0	0	1	1	0	1	1

位	功能描述
IADC_BIAS[2:0]	电流通道的 ADC Bias 调节位, default='011' 000-> 4uA 001->6uA 010->8uA 011->10uA(Default) 1xx->12uA
PGA_BIAS[2:0]	PGA Bias 调节位, default='011' 000-> 5uA 001->10uA 010->15uA 011->20uA (Default) 1xx->25uA
CLKCfg[4:0]	时钟调节位, Default = '00010'
UADC_BIAS[2:0]	电压通道 ADC BIAS 调节位, default='011' 000-> 4uA 001->6uA 010->8uA 011->10uA(Default)

	1xx->12uA
ldo_emu_en	EMU计量单元的LDO控制位， <b>EMU功能的开启，该Bit必须先打开</b> 1：开启 0：关闭(Default)
Vref_En	高精度VREF的EN信号， <b>ADC的Vref功能的开启，该Bit必须先打开</b> 1：开启 0：关闭(Default)

ANACFG2			基地址：0x4001 C000 偏移地址：2A4					
	Bit23	22	21	20	19	18	17	Bit16
Read:					Iref_En	TC2ND_EN	Reserved	Reserved
Write:								
Reset:					0	1	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	Reserved	Reserved	Reserved					
Write:	Reserved	Reserved	Reserved					
Reset:	0	1	1	1	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Vref_Ctrl[2:0]			VrefChop_En	InChop_En	IChop_En	UChop_En	AMPChop_En
Write:	Vref_Ctrl[2:0]			VrefChop_En	InChop_En	IChop_En	UChop_En	AMPChop_En
Reset:	0	1	1	1	1	1	1	1

位	功能描述
AMPChop_En	PGA Chopper Enable信号，默认为‘1’，开启
UChop_En	U ADC Chopper Enable信号，默认为‘1’，U ADC Chopper开启
IChop_En	I ADC Chopper Enable信号，默认为‘1’，I ADC Chopper开启
InChop_En	In ADC Chopper Enable信号，默认为‘1’，InADC Chopper开启
VrefChop_En	VREF Chopper EN信号，Default=1开启
Vref_Ctrl[2:0]	VRef TC code调节，default=‘011’
TC2ND_En	2阶Vref 调整信号，Default=‘1’
Iref_En	高精度Iref的En信号， <b>ADC的Iref功能的开启，该Bit必须先打开</b> 1：开启 0：关闭(Default)

## 22.9.27 全失压比较器配置

ULostCMPCFG 全失压比较器配置寄存器			基地址：0x4001 C000 偏移地址：2AC					
	Bit15	14	13	12	11	10	9	Bit8

Read:	CMP_PRD	DUTY[6:0]						
Write:	_EN							
Reset:	0	0	1	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	VcmpLVL[2:0]			VOLNUM[3:0]				CMP_EN
Write:								
Reset:	0	0	0	0	1	1	1	0

位	功能描述
CMP_EN	CMP 比较器模块使能位: 0: 不使能; 1: 使能比较检测模块
VOLNUM[3:0]	设定比较器比较的正弦波的周期数为 VOLNUM+1  默认值为 7, 实际检测 7+1=8 个周期 (单周期正常信号为 20ms 时间长度)
VcmpLVL[2:0]	LOSTU 档位选择信号, default '000' 000->0.942mV 001->1.414mV 010->1.791mV 011->3.140mV 100->4.488mV 101->5.837mV 110->7.185mV 111->9.542mV
DUTY[6:0]	设定最小有效脉宽时长系数 = Round(DUTY/32768Hz)  Default=32, 设定最小脉宽时长为1ms。最大填满7个Bit即 127 / 32768 = 3.8Ms
CMP_PRD_EN	比较器的分时/连续模式选择开关。分时模式最短间隔 1s。 0: 分时模式(default) ; 1: 连续模式  打开比较器使能开关 CMP_EN 后, 配置分时模式, 则比较器按照 CMPPRD 的时间去定时开启比较器。比较器本身开启后运行时间定为 CMPRunTime[7:0]。在此期间内用户可以直接手动写 CMP_EN 关闭比较器, 如果不手动写, CMPRunTime[7:0]时间到了以后比较器也会自行关闭。然后比较器等待 CMPPRD 时间再去开启。  如果设置了连续模式, 比较器开启后就不再自动关闭, 直到用户手动关闭 CMP_En

ULostPRDCFG 全失压比较器分时寄存器			基地址: 0x4001 C000 偏移地址: 2B0					
	Bit23	22	21	20	19	18	17	Bit16
Read:	STBTime[5:0]							
Write:								
Reset:			0	0	1	0	1	0

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

位	功能描述
CMPPRD[7:0]	比较器分时检测周期设定 8bits, 单位为s 比较器分时模式开启CMP_PRD_EN=1时有效, 间隔时间为(CMPPRD+1)秒. =0间隔1s, 最大间隔为256s。CMPPRD更新需等待当前周期结束后生效。(默认值为60秒, 即0x3F)
CMPRunTime[7:0]	比较器在打开后的运行时间设定 运行时间=round( (CMPRunTime) /256 ) CMPRunTime的可写范围0~255, CMPRunTime就是从0~1S 每一个Bit代表3.9个mS即128个32768 clk Default=64, 即64*3.9mS = 250mS
STBTime[5:0]	每次CMP_EN打开, 先丢STBTime[5:0]的时间。 STBTime[5:0]定义: LSB分辨率为32个32768 clk即1mS, 稳定时间可设范围0~63mS Default值001010 即10mS 注意: STBTime[5:0]并不包含在CMPRunTime[7:0]内! 二者为时间衔接关系
备注 2	用户约束: 全失压比较器的分时功能, 只能在MCU Hold下使用, Normal模式不要使用分时模式, 否则会有问题。 用户约束: 全失压比较器模式下, 基准必须切换到LVref

## 22.9.28 低频计量模式配置

<b>LPMoDeCON</b>		<b>基地址: 0x4001 C000</b>						
<b>EMU 低功耗模式控制寄存器</b>		<b>偏移地址: 2B8</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:	StandByTime[3:0]				X	LP_VREF_SEL	EMUCLK_LPSEL	LPMoDe
Write:								
Reset:	0	1	0	1	0	0	0	0

位	功能描述
StandByTime	低功耗模式下, 开启的固定丢点数设定对应固定丢点数为:

	StandByTime[3:0]				丢的点数
	0	0	0	0	不丢点
	0	0	0	1	4个点
	0	0	1	0	8个点
	0	0	1	1	16个点
	0	1	0	0	32个点
	0	1	0	1	64个点(Default)
	0	1	1	0	128个点
	0	1	1	1	256个点
	1	0	0	0	512个点
	1	0	0	1	1024个点
1	X	1	X	不丢点	
LPMode	低功耗模式使能 Bit 只有该 Bit 置 1, 相关功能的配置(emu 低频时钟等)才可以有效, 否则那些功能配置位可写但是不无效。				
EMUCLK_LPSEL	低功耗模式下选择使用的 EMUCLK 0: 选择 EMU 时钟为 204.8K ; 1: 选择 EMU 时钟为 32K				
LP_VREF_SEL	计量使用LVREF选择信号, Default= 0 0: 选择高精度Vref 1: 选择LVref				
备注	1. 低功耗模式下, 电能累加模块速度不再是408K,而是切换成低功耗模式下所使用的时钟204.8K或32K 2. LPMode 的作用: 只有在 LPMode=1 的前提下: (1) 电能累计模块的速度: 切换成低功耗模式下所使用的二分频时钟 204.8K/2=102.4k 或 32K/2 = 16k (2) OSR 的选择固定死为 64 (LPMode=1, OSR128Bit 只能为 0, 不可选 128) (3) StandByTime 的丢点才可以生效。(LPMode=0, 丢点数永为 0) 3. 不和低功耗模式 LPMode 绑在一起的(应用注意保证) (1) Analog 下各种 PGA/ADC 等的 BIAS 之类的节省功耗的配置 (2) Vref 和 LVref 的切换, 用户需要注意: 切换 Vref 和 ADC 的关系, 不要先关 Vref 再关 ADC (3) Uconst 不合 LPMode 绑一起, 可单独使用。				

## 22.9.29 电能相关配置寄存器

电能通道的分组	电能各通道	前端采样功率	对应电能(正反分支)
1	1	全波 A 相有功	All_EnergyPos/Neg_Pa
	2	全波 B 相有功	All_EnergyPos/Neg_Pb

	3	全波 C 相有功	All_EnergyPos/Neg_Pc
	4	全波合相有功	All_EnergyPos/Neg_Pt
2	5	全波 A 相无功	All_EnergyPos/Neg_Qa
	6	全波 B 相无功	All_EnergyPos/Neg_Qb
	7	全波 C 相无功	All_EnergyPos/Neg_Qc
	8	全波合相无功	All_EnergyPos/Neg_Qt
3	9	A 相视在	Energy_Sa/Neg_SaConst
	10	B 相视在	Energy_Sb/Neg_SbConst
	11	C 相视在	Energy_Sc/Neg_ScConst
	12	合相视在	Energy_St/Neg_StConst
4	13	基波 A 相有功	Fund_EnergyPos/Neg_Pa
	14	基波 B 相有功	Fund_EnergyPos/Neg_Pb
	15	基波 C 相有功	Fund_EnergyPos/Neg_Pc
	16	基波合相有功	Fund_EnergyPos/Neg_Pt
5	17	基波 A 相无功	Fund_EnergyPos/Neg_Qa
	18	基波 B 相无功	Fund_EnergyPos/Neg_Qb
	19	基波 C 相无功	Fund_EnergyPos/Neg_Qc
	20	基波合相无功	Fund_EnergyPos/Neg_Qt
6	21	谐波 A 相有功	Har_EnergyPos/Neg_Pa
	22	谐波 B 相有功	Har_EnergyPos/Neg_Pb
	23	谐波 C 相有功	Har_EnergyPos/Neg_Pc
	24	谐波合相有功	Har_EnergyPos/Neg_Pt

CFxRun		基地址: 0x4001 C000 偏移地址: 2BC						
	Bit23	22	21	20	19	18	17	Bit16
Read:	CFRun_Har_Pt	CFRun_Har_Pc	CFRun_Har_Pb	CFRun_Har_Pa	CFRun_Fund_Qt	CFRun_Fund_Qc	CFRun_Fund_Qb	CFRun_Fund_Qa
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	CFRun_Fund_Pt	CFRun_Fund_Pc	CFRun_Fund_Pb	CFRun_Fund_Pa	CFRun_St	CFRun_Sc	CFRun_Sb	CFRun_Sa
Write:	nd_Pt	nd_Pc	nd_Pb	nd_Pa				
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CFRun_All_Pt	CFRun_All_Pc	CFRun_All_Pb	CFRun_All_Pa	CFRun_All_Pt	CFRun_All_Pc	CFRun_All_Pb	CFRun_All_Pa
Write:	l_Qt	l_Qc	l_Qb	l_Qa	l_Pt	l_Pc	l_Pb	ll_Pa
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Bit0:CFRun_All_Pa	第1 路全波有功电能通路的能量累加独立使能位 0: 停止第1路的电能累加 1: 允许第1路的电能累加

Bit1-23	分别表示第2-24路电能通路的能量累加独立使能位
---------	--------------------------

EnergyCon		基地址: 0x4001 C000 偏移地址: 2C0							
	Bit23	22	21	20	19	18	17	Bit16	
Read:	PassMode_Har_P		PassMode_Fund_Q		PassMode_Fund_P		PassMode_S		
Write:	PassMode_Har_P		PassMode_Fund_Q		PassMode_Fund_P		PassMode_S		
Reset:	0	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8	
Read:	PassMode_All_Q		PassMode_All_P		EnergySSr	EnergyRC	PtAddMod	POS	
Write:	PassMode_All_Q		PassMode_All_P		c_Sel	_En	e	POS	
Reset:	0	0	0	0	0	1	0	0	
	Bit7	6	5	4	3	2	1	Bit0	
Read:	CFP[1:0]		Reserved			Power_Shift[2:0]			
Write:	CFP[1:0]		Reserved			Power_Shift[2:0]			
Reset:	0	0	0	0	0	0	0	0	0

位名称	描述															
Power_Shift	脉冲(功率)加倍功能---不包括谐波 000:1 倍 001:2 倍 010:4 倍 011:8 倍 100:16 倍 101/110/111 :1 倍															
POS	脉冲有效电平选择位 0: 表示脉冲为高电平有效 1: 表示脉冲为低电平有效															
CFP[1:0]	<table border="1"> <thead> <tr> <th>CFP1</th> <th>CFP0</th> <th>脉宽 (femu=819.2KHz)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>80ms</td> </tr> <tr> <td>0</td> <td>1</td> <td>40ms</td> </tr> <tr> <td>1</td> <td>0</td> <td>20ms</td> </tr> <tr> <td>1</td> <td>1</td> <td>160ms</td> </tr> </tbody> </table>	CFP1	CFP0	脉宽 (femu=819.2KHz)	0	0	80ms	0	1	40ms	1	0	20ms	1	1	160ms
CFP1	CFP0	脉宽 (femu=819.2KHz)														
0	0	80ms														
0	1	40ms														
1	0	20ms														
1	1	160ms														
PtAddMode	分相去计算合相功率的相加方式 0 : 分相功率带符号相加计算合相功率 (代数和相加) 1 : 分相功率取绝对值后相加计算合相功率 (绝对值相加)															
EnergyRC_En	所有电能寄存器是否配为读后清零型 0: 读后不清零 1: 读后清零(Default)															
EnergySSrc_Sel	视在电能选择 Bit 0: 选择全波视在功率计算视在电能 1: 选择基波视在功率计算视在电能															

PassMode	功率选通模式:对功率符号判断 00 全通模式: 正负不做判断, 均通过 (Default) 01 只正模式: 只正通过 1X 只负模式: 只负通过 注意电能分6组, PassMode的配置能够控制该组下的ABCT四种功率
----------	--

PowerSrcCFG1			基地址: 0x4001 C000 偏移地址: 2C4					
	Bit23	22	21	20	19	18	17	Bit16
Read:	PowerSrc_St		PowerSrc_Sc		PowerSrc_Sb		PowerSrc_Sa	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	PowerSrc_All_Qt		PowerSrc_All_Qc		PowerSrc_All_Qb		PowerSrc_All_Qa	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PowerSrc_All_Pt		PowerSrc_All_Pc		PowerSrc_All_Pb		PowerSrc_All_Pa	
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述
PowerSrc	选择用于电能累加的功率: 00 平均功率, Default 01 瞬时功率, 1X 常数功率

PowerSrcCFG2			基地址: 0x4001 C000 偏移地址: 2C8					
	Bit23	22	21	20	19	18	17	Bit16
Read:	PowerSrc_Har_Pt		PowerSrc_Har_Pc		PowerSrc_Har_Pb		PowerSrc_Har_Pa	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	PowerSrc_Fund_Qt		PowerSrc_Fund_Qc		PowerSrc_Fund_Qb		PowerSrc_Fund_Qa	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PowerSrc_Fund_Pt		PowerSrc_Fund_Pc		PowerSrc Fund_Pb		PowerSrc_Fund_Pa	
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述
PowerSrc	选择用于电能累加的功率： 00 平均功率，Default 01 瞬时功率， 1X 常数功率

PowerModeCFG 功率分栋模块的四种模式		基地址：0x4001 C000 偏移地址： 2CC							
	Bit23	22	21	20	19	18	17	Bit16	
Read:	JudgeEn_	JudgeEn_	JudgeEn_	JudgeEn_	JudgeEn_	JudgeEn_	JudgeEn_	PowerOV	PowerOV
Write:	Har	Fund_Q	Fund_P	S	All_Q	All_P	Sel_HarP	Sel_Fund	Q
Reset:	0	0	0	0	0	0	1	1	
	Bit15	14	13	12	11	10	9	Bit8	
Read:	PowerOV	PowerOV	PowerOV	PowerOV	AccuMode_Har_P		AccuMode_Fund_Q		
Write:	Sel_FundP	Sel_S	Sel_AllQ	Sel_AllP					
Reset:	1	1	1	1	0	0	0	0	
	Bit7	6	5	4	3	2	1	Bit0	
Read:	AccuMode_Fund_P		AccuMode_S		AccuMode_All_Q		AccuMode_All_P		
Write:									
Reset:	0	0	0	0	0	0	0	0	

位名称	描述
AccuMode	电能累计模式： 00 Mode0 01 Mode1 10 Mode2 11 Mode3
PowerOVSel	配合PowerAccuModeCFG使用 在模式0的前提下 0: 正反电能均累加到正向能量寄存器 1: 正反电能分别累计到正向电能寄存器和反向电能寄存器
Judge90En	用来设定24路电能各自通道的是否过小功率因数判断使能 注意只用在正反向模式 注意视在功率和谐波有功功率JudgeEn并无实际意义

CFCFG1 能量脉冲控制寄存器		基地址：0x4001 C000 偏移地址： 2D0							
	Bit23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit16	
Read:	X	X	X	CFCFG_IO2[4:0]					

Write:								
Reset:				0	0	0	0	0
	Bit15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit8
Read:	X	X	X	CFCFG_IO1[4:0]				
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
Read:	X	X	X	CFCFG_IO0[4:0]				
Write:								
Reset:				0	0	0	0	0

<b>CFCFG2</b> 能量脉冲控制寄存器			基地址: 0x4001 C000 偏移地址: 2D4					
	Bit15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit8
Read:	X	X	X	CFCFG_IO4[4:0]				
Write:								
Reset:				0	0	0	0	0
	Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
Read:	X	X	X	CFCFG_IO3[4:0]				
Write:								
Reset:				0	0	0	0	0

位名称	描述																														
	<p>该寄存器控制CF的5路CF Pin分别发什么电能脉冲出来.注意6路电能和最终发出的5路CF Pin的对应关系</p> <p>每路CF电能通道都有P正向电能, N反向电能, P和N或运算发出来。默认是发或运算之后(如果是低有效,发的是与 运算)的电能。可以配制成P, 也可配成N, 也可以配成别的CF通道电能。</p> <p>6路电能分别为: AllPt AllQt St FundPt FundQt HarPt所产生的, 现指定发脉冲的功率为:</p> <table border="1" data-bbox="427 1541 999 1756"> <tr> <td>All Pt</td> <td>P1</td> <td>N1</td> <td>P1</td> <td>N1</td> </tr> <tr> <td>All Qt</td> <td>P2</td> <td>N2</td> <td>P2</td> <td>N2</td> </tr> <tr> <td>St</td> <td>P3</td> <td>N3</td> <td>P3</td> <td>N3</td> </tr> <tr> <td>Fund Pt</td> <td>P4</td> <td>N4</td> <td>P4</td> <td>N4</td> </tr> <tr> <td>Fund Qt</td> <td>P5</td> <td>N5</td> <td>P5</td> <td>N5</td> </tr> <tr> <td>Har Pt</td> <td>P6</td> <td>N6</td> <td>P6</td> <td>N6</td> </tr> </table> <p>CFCFG_IOx的关系定义如下</p> <p>(Default状态下PFOUT0-4引脚分别发的电能是: 全波有功/全波无功/基波有功/谐波正/谐波负)</p>	All Pt	P1	N1	P1	N1	All Qt	P2	N2	P2	N2	St	P3	N3	P3	N3	Fund Pt	P4	N4	P4	N4	Fund Qt	P5	N5	P5	N5	Har Pt	P6	N6	P6	N6
All Pt	P1	N1	P1	N1																											
All Qt	P2	N2	P2	N2																											
St	P3	N3	P3	N3																											
Fund Pt	P4	N4	P4	N4																											
Fund Qt	P5	N5	P5	N5																											
Har Pt	P6	N6	P6	N6																											

CFCFG_I01[4:0]	CF1 Pin	CFCFG_I02[4:0]	CF2 Pin	CFCFG_I03[4:0]	CF3 Pin
0 0 0 0 0	P1   N1	0 0 0 0 0	P2   N2	0 0 0 0 0	P4   N4
0 0 0 0 1	P1	0 0 0 0 1	P2	0 0 0 0 1	P4
0 0 0 1 0	N1	0 0 0 1 0	N2	0 0 0 1 0	N4
0 0 0 1 1	P2   N2	0 0 0 1 1	P1   N1	0 0 0 1 1	P1   N1
0 0 1 0 0	P2	0 0 1 0 0	P1	0 0 1 0 0	P1
0 0 1 0 1	N2	0 0 1 0 1	N1	0 0 1 0 1	N1
0 0 1 1 0	P3   N3	0 0 1 1 0	P3   N3	0 0 1 1 0	P2   N2
0 0 1 1 1	P3	0 0 1 1 1	P3	0 0 1 1 1	P2
0 1 0 0 0	N3	0 1 0 0 0	N3	0 1 0 0 0	N2
0 1 0 0 1	P4   N4	0 1 0 0 1	P4   N4	0 1 0 0 1	P3   N3
0 1 0 1 0	P4	0 1 0 1 0	P4	0 1 0 1 0	P3
0 1 0 1 1	N4	0 1 0 1 1	N4	0 1 0 1 1	N3
0 1 1 0 0	P5   N5	0 1 1 0 0	P5   N5	0 1 1 0 0	P5   N5
0 1 1 0 1	P5	0 1 1 0 1	P5	0 1 1 0 1	P5
0 1 1 1 0	N5	0 1 1 1 0	N5	0 1 1 1 0	N5
0 1 1 1 1	P6   N6	0 1 1 1 1	P6   N6	0 1 1 1 1	P6   N6
1 0 0 0 0	P6	1 0 0 0 0	P6	1 0 0 0 0	P6
1 0 0 0 1	N6	1 0 0 0 1	N6	1 0 0 0 1	N6

CFCFG_I04[4:0]	CF4 Pin	CFCFG_I05[4:0]	CF5 Pin
0 0 0 0 0	P6	0 0 0 0 0	N6
0 0 0 0 1	P6   N6	0 0 0 0 1	P6   N6
0 0 0 1 0	N6	0 0 0 1 0	P6
0 0 0 1 1	P1   N1	0 0 0 1 1	P1   N1
0 0 1 0 0	P1	0 0 1 0 0	P1
0 0 1 0 1	N1	0 0 1 0 1	N1
0 0 1 1 0	P2   N2	0 0 1 1 0	P2   N2
0 0 1 1 1	P2	0 0 1 1 1	P2
0 1 0 0 0	N2	0 1 0 0 0	N2
0 1 0 0 1	P3   N3	0 1 0 0 1	P3   N3
0 1 0 1 0	P3	0 1 0 1 0	P3
0 1 0 1 1	N3	0 1 0 1 1	N3
0 1 1 0 0	P4   N4	0 1 1 0 0	P4   N4
0 1 1 0 1	P4	0 1 1 0 1	P4
0 1 1 1 0	N4	0 1 1 1 0	N4
0 1 1 1 1	P5   N5	0 1 1 1 1	P5   N5
1 0 0 0 0	P5	1 0 0 0 0	P5
1 0 0 0 1	N5	1 0 0 0 1	N5

### 22.9.30 闪变功能配置

FlickerCfg 闪变配置寄存器		基地址: 0x4001 C000 偏移地址: 358						
	Bit15	14	13	12	11	10	9	Bit8
Read:		Reserved	Flickerbpf_lshift			Flickerhpf_lshift		
Write:								
Reset:		0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0

Read:	FlickerUpdRate					FlickerUc En	FlickerUb En	FlickerUaEn
Write:								
Reset:	1	1	1	1	1	0	0	0

位名称	描述
FlickerUaEn FlickerUbEn FlickerUcEn	电压闪变功能使能位， 当使能时，选择闪变功能所需要的滤波器，为谐波滤波器。 即ABC三相电压通道的谐波滤波器
FlickerUpdRate	Pinst数据更新速率，可选51.6Hz~1.6KHz 更新频率为1600/ FlickerUpdRate 即如果需要100Hz的更新速度，该寄存器需写10000
Flickerhpf_lshift	闪变功能第一级滤波器放大设置
Flickerbpf_lshift	闪变功能第二级滤波器放大设置
备注	使用Pinst寄存器计算闪变，需要丢弃前10秒钟的数据以保持最优结果性能。

### 22.9.31 基波谐波使能配置

FundHarCFG		基地址： 0x4001 C000 偏移地址： 71C						
	Bit15	14	13	12	11	10	9	Bit8
Read:						InHarFilterShift[2:0]		
Write:								
Reset:						0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	InMultiFunc_E	InMultiFuncSel	HarFilter_Shift[2:0]			Har_Alg Mode	All_En	FundHar_Func_En
Write:	n							
Reset:	0	0	0	0	0	0	0	0

位名称	描述
FundHar_Func_En	基波谐波功能的使能开关 0: Default, 基波谐波功能不使能 1: 基波谐波功能的使能, 注: 基波滤波器本身是常开的, 使能的作用为后续数据的计算
All_En	全波通路使能开关, Default 为 0, 即 Default 状态为把全波滤波器旁路, 不使能
Har_AlgMode	谐波通路的算法选择 Har_AlgMode0: 选择全波减基波的方式计算谐波参数(Default), Har_AlgMode 1: 选择谐波滤波器实现谐波计算参数
HarFilter_Shift[2:0]	谐波滤波器方式, 倍数=2^shifit,全波-基波模式, 倍数=2^shifit, 且

	shift $\geq$ 5 时, 倍数=1 倍 即: 当 Har_AlgMode=0 时, 000: 1 倍; 001:2 倍; 010:4 倍; 011 :8 倍; 100: 16 倍; 101/110/111 :1 倍; 当 Har_AlgMode=1 时, 000: 1 倍; 001:2 倍; 010:4 倍; 011 :8 倍; 100: 16 倍; 101: 32; 110: 64; 111:128 倍
InMultiFuncSel	In 通道谐波滤波器功能选择 该 Bit 专门用于 In 通道的用于谐波功能滤波器的扩展, 0: 为 InHar 滤波器, Default 1:其他功能 (需要用户手动切换系数)
InMultiFunc_En	In 通道的 Har 滤波器, 用于其他功能时的使能位, 配合 InMultiFuncSel 使用
InHarFilterShift[2:0]	In 通道的 Har 滤波器, 当 Har_AlgMode=0 时, 000: 1 倍; 001:2 倍; 010:4 倍; 011 :8 倍; 100: 16 倍; 101/110/111 :1 倍; 当 Har_AlgMode=1 时, 000: 1 倍; 001:2 倍; 010:4 倍; 011 :8 倍; 100: 16 倍; 101: 32; 110: 64; 111:128 倍

## 22.10 状态标注与中断说明

### 22.10.1 潜动状态标志

NoLoad 潜动状态标志寄存器			基地址: 0x4001B000 偏移地址: 400					
	Bit23	22	21	20	19	18	17	Bit16
Read:				NoLoad_H arPc	NoLoad_H arPb	NoLoad_H arPa	NoLoad_F undSc	NoLoad_F undSb
Write:								
Reset:								
	Bit15	14	13	12	11	10	9	Bit8
Read:	NoLoad_ FundSa	NoLoad_F undQc	NoLoad_ FundQb	NoLoad_F undQa	NoLoad_F undPc	NoLoad_F undPb	NoLoad_F undPa	NoLoad_A llSc
Write:								
Reset:								
	Bit7	6	5	4	3	2	1	Bit0
Read:	NoLoad_ AllSb	NoLoad_ AllSa	NoLoad_ AllQc	NoLoad_A llQb	NoLoad_A llQa	NoLoad_A llPc	NoLoad_A llPb	NoLoad_A llPa
Write:								
Reset:								

位名称	描述
NoLoad_Power	对应功率是否处于潜动状态 视在功率的启动取决于P/Q 只要相应的P/Q功率启动了, 那么视在功率也就启动了。(可以去计量电能, 能不能计量还要看视在电能通道选的是All/Fund)

该寄存器提供的潜动状态，因此S的状态应该为P&Q  
寄存器提供的是潜动状态，启动为0，潜动状态为1

## 22.10.2 功率/电能反向标志

REV_Power 功率方向标志寄存器			基地址: 0x4001B000 偏移地址: 404					
	Bit23	22	21	20	19	18	17	Bit16
Read:					Rev_HarP	Rev_HarP	Rev_HarP	Rev_HarP
Write:					t	c	b	a
Reset:								
	Bit15	14	13	12	11	10	9	Bit8
Read:	Rev_Fund	Rev_Fund	Rev_Fund	Rev_Fun	Rev_Fund	Rev_Fund	Rev_Fund	Rev_Fund
Write:	Qt	Qc	Qb	dQa	Pt	Pc	Pb	Pa
Reset:								
	Bit7	6	5	4	3	2	1	Bit0
Read:	Rev_AllQt	Rev_AllQ	Rev_AllQ	Rev_All	Rev_AllPt	Rev_AllPc	Rev_AllPb	Rev_AllPa
Write:		c	b	Qa				
Reset:								

位名称	描述
Rev_Power	各功率的反向标志 注 在功率寄存器更新时产生，来源是平均功率的符号。

REV_Energy 能量方向标志寄存器			基地址: 0x4001B000 偏移地址: 408					
	Bit23	22	21	20	19	18	17	Bit16
Read:	REV_Har	REV_Har	REV_Har	REV_Ha	REV_Fun	REV_Fun	REV_Fund	REV_Fund
Write:	EPt	EPc	EPb	rEPa	dEQt	dEQc	EQb	EQa
Reset:								
	Bit15	14	13	12	11	10	9	Bit8
Read:	REV_Fun	REV_Fun	REV_Fun	REV_Fu	REV_ESt	REV_ESc	REV_ESb	REV_ESa
Write:	dEPt	dEPc	dEPb	ndEPa				
Reset:								
	Bit7	6	5	4	3	2	1	Bit0
Read:	REV_AllE	REV_AllE	REV_AllE	REV_All	REV_AllE	REV_AllE	REV_AllE	REV_AllE
Write:	Qt	Qc	Qb	EQa	Pt	Pc	Pb	Pa
Reset:								

位名称	描述
Rev_EnergyP	各电能通路的反向标志

	<p>该标志在电能寄存器变化时更新，即来源是电能通道的CFxCnt累计溢出大于HFConst并Energy的方向。</p> <p>由于通路电能有正向和反向两个电能寄存器，定义电能通路的反向标志为正反两个电能的或运算。（注意是CFxCnt累计溢出大于HFConst的时刻触发比较，并不是CFxCnt的过程状态）</p> <p>如果视在电能通道选用了自定义常数功率，视在电能寄存器就有被反向溢出的可能性，为保持统一，也提供反向电能标志。请注意区分</p> <p>注：在电能没有出来的时候，指示的是上次出电能的状态维持。</p>
--	---

### 22.10.3 功率因数过小标志

P90_Power 电能通道功率因数过小标志		基地址：0x4001B000 偏移地址： 40C						
	Bit15	14	13	12	11	10	9	Bit8
Read:	P90_Fund	P90_Fund	P90_Fund	P90_Fund	P90_Fun	P90_Fun	P90_Fund	P90_Fund
Write:	EQt	EQc	EQb	EQa	dEPt	dEPc	EPb	EPa
Reset:								
	Bit7	6	5	4	3	2	1	Bit0
Read:	P90_AllIE	P90_AllIE	P90_AllIE	P90_AllIE	P90_All	P90_AllIE	P90_AllIEP	P90_AllIEP
Write:	Qt	Qc	Qb	Qa	EPt	Pc	b	a
Reset:								

位名称	描述
P90_Power	当前电能通道为P，处于功率因数过小状态 当前电能通道为Q，处于功率因数过小状态 1~24路电能中，视在和谐波通路不存在处于功率因数过小状态

### 22.10.4 信号状态标志

SignalStatus 信号状态标志寄存器		基地址：0x4001B000 偏移地址： 410						
	Bit31	30	29	28	27	26	25	Bit24
Read:							AngleBaseFlag[2:1]	
Write:								
Reset:								
	Bit23	22	21	20	19	18	17	Bit16
Read:	AngleBase	ZCValid_I	ZCValid_I	ZCValid_I	ZCValid_I	ZCValid_I	ZCValid_U	ZCValid_Ua
Write:	Flag[0]	n	c	b	Ia	Uc	b	
Reset:								
	Bit15	14	13	12	11	10	9	Bit8

Read:	RegionIc		RegionIb		RegionIa		UOrder	IOrder
Write:								
Reset:								
	Bit7	6	5	4	3	2	1	Bit0
Read:	Mod3P3Sta	VoltageFai	VoltageFai	VoltageFai	CMPC	CMPB	CMPA	CMPDone
Write:	tus	IC	IB	IA				
Reset:								

位名称	描述
CMPDone	比较器比较完成标识( CMPRunTime[7:0]) 为1表示全失压预判结束， 为0表示全失压预判正在进行
CMPA\B\C	分别指示对应A\B\C三相，在上一次比较器期间的比较输出状态。当有效脉宽数大于设定VOLNUM时，该状态标志置1. 这个标志会持续到比较器的下次开启。如果在比较器工作的CMPRunTime[7:0]内，都没有发生有效脉宽数大于设定VOLNUM，那么就会在CMPRunTime[7:0]时间到了将标志置0。
VoltageFailA/B/C	= 1 表示 A/B/C 相失压，Urms <(FailVoltage<<8)设置阈值
Mod3P3Status	=1 当前系统为三相三线模式； =0 当前系统为三相四线模式
IOrder	电流错序
UOrder	电压错序
RegionIa/Ib/Ic	各相电流有效值，所处的相位分段 00: 电流处于区域 0； 01: 电流处于区域 1 1X: 电流处于区域 2
ZCValid_Ux/Ix	各相电压电流过零有效标志 1:有效， 0: 无效
AngleBaseFlag	角度基准参考信号状态标志 000/001/010/011/100/101/11X : ua/ub/uc/ia/ib/ic/in 注意该 Bit 为组合 AutoAngBase 与 AngleBaseSel[2:0]功能。 如果角度手动切换，则把选择的参考信号放到该标志。如果选择自动切换，则把系统选择的参考信号放到该标志

## 22.10.5 Peak/Sag/UINT/IOV 状态标志

PeakSagIntIovStau		基地址: 0x4001B000						
PEAK/SAG/INT/IOV 标志		偏移地址: 414						
	Bit15	14	13	12	11	10	9	Bit8
Read:				OVIN	OVIC	OVIB	OVIA	UINTUC
Write:								
Reset:								
	Bit7	6	5	4	3	2	1	Bit0
Read:	UINTUB	UINTUA	SagUC	SagUB	SagUA	PeakUC	PeakUB	PeakUA
Write:								

Reset:								
--------	--	--	--	--	--	--	--	--

位名称	描述
PeakUA/UB/UC	Peak 事件发生后即 Ustart 和 PeakIF 置位后，每半个周波更新一次这个标志。直到 Uend 的中断这个标志位就清零。
SagUA/UB/UC	同 Peak
UINTUA/UB/UC	同 Peak
OVIa/IB/IC/IN	同 Peak，电流通道过流事件

### 22.10.6 EMU 中断使能与中断标志

EMUIE1 EMU 中断使能寄存器 1		基地址: 0x4001 C000 偏移地址: 380						
	Bit31	30	29	28	27	26	25	24
Read:	RMSItO	RMSUt	RMSTUp	RMSUpd	PowerUp	SPLUpdat	Energy	ZCUcLo
Write:	VIE	OVIIE	dateIE	ateIE	dateIE	eIE	OVIIE	stIE
Reset:	0	0		0	0	0	0	0
	23	22	21	20	19	18	17	16
Read:	ZCUBLo	ZCUaLo	CMPcIE	CMPbIE	CMPaIE	ZCInIE	ZCICIE	ZCIbIE
Write:	stIE	stIE						
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:								
Write:	ZCIaIE	ZCUcIE	ZCUBIE	ZCUaIE	HarP_Ne	FundQ_N	FundP_N	S_NegC
Reset:	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	AllQ_Ne	AllP_Ne	HarP_Pos	FundQ_P	FundP_P	S_PosCFI	AllQ_P	AllP_Pos
Write:	gCFIE	gCFIE	CFIE	osCFIE	osCFIE	E	osCFIE	CFIE
Reset:	0	0	0	0	0	0	0	0

位名称	描述 EMU 中断使能寄存器 1 - EMUIE1
AllP_PosCFIE	全波有功通路 CF 正向脉冲中断使能
AllQ_PosCFIE	全波无功通路 CF 正向脉冲中断使能
S_PosCFIE	视在通路 CF 正向脉冲中断使能
FundP_PosCFIE	基波有功通路 CF 正向脉冲中断使能
FundQ_PosCFIE	基波无功通路 CF 正向脉冲中断使能
HarP_PosCFIE	谐波有功通路 CF 正向脉冲中断使能
AllP_NegCFIE	全波有功通路 CF 反向脉冲中断使能
AllQ_NegCFIE	全波无功通路 CF 反向脉冲中断使能

S_NegCFIE	视在通路 CF 反向脉冲中断使能
FundP_NegCFIE	基波有功通路 CF 反向脉冲中断使能
FundQ_NegCFIE	基波无功通路 CF 反向脉冲中断使能
HarP_NegCFIE	谐波有功通路 CF 反向脉冲中断使能
ZCUaIE	A 相电压过零中断使能
ZCUbIE	B 相电压过零中断使能
ZCUCIE	C 相电压过零中断使能
ZCIaIE	A 相电流过零中断使能
ZCIbIE	B 相电流过零中断使能
ZCIcIE	C 相电流过零中断使能
ZCInIE	零线电流过零中断使能
CMPaIE	电流通道模拟比较器 A 相中断使能
CMPbIE	电流通道模拟比较器 B 相中断使能
CMPcIE	电流通道模拟比较器 C 相中断使能
ZCUaLostIE	A 相电压过零丢失中断使能
ZCUbLostIE	B 相电压过零丢失中断使能
ZCUCLostIE	C 相电压过零丢失中断使能
EnergyOVIE	24 路共 48 个电能寄存器有某个寄存器发生了电能溢出中断使能 注意该中断是对 48 个电能寄存器溢出的总使能, 任何一个电能溢出都可以产生 EnergyOVIF, 用户需要去二级状态判断哪个电能发生溢出了
SPLUpdateIE	SPL 寄存器的更新中断使能
PowerUpdateIE	Power 寄存器的更新中断使能
RMSUpdateIE	RMS 寄存器的更新中断使能
RMSTUpdateIE	RMST 寄存器的更新中断使能
RMSUtOVIE	RMS_Ut 大于 RMSUt_THO 则触发 RMSUtOVIF 中断
RMSItOVIE	RMS_It 大于 RMSIt_THO 则触发 RMSItOVIF 中断

EMUIE2		基地址: 0x4001 C000						
EMU 中断使能寄存器 2		偏移地址: 384						
	Bit31	30	29	28	27	26	25	24
Read:	X	VpupdIn	VpupdIcI	VpupdIbI	VpupdIaI	VpupdU	VpupdUb	VpupdUa
Write:		IE	E	E	E	cIE	IE	IE
Reset:	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Read:	InO	IcOVEn	IbOVEnd	IaOVEnd	InOVIE	IcOVIE	IbOVIE	IaOVIE
Write:	VEndIE	dIE	IE	IE				
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	UcINTE	UbINTE	UaINTE	INTUIE	INTUcIE	INTUbi	INTUaIE	UcEndIE
Write:	ndIE	ndIE	dIE			E		
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0

Read:	UbEndIE	UaEndIE	SagUcIE	SagUbIE	SagUaIE	PeakUcIE	PeakUbIE	PeakUaIE
Write:								
Reset:	0	0	0	0	0	0	0	0

位名称	描述 EMU 中断使能寄存器 2 –EMUIE2
PeakUaIE	A 相发生 Peak 事件中断使能
PeakUbIE	B 相发生 Peak 事件中断使能
PeakUcIE	C 相发生 Peak 事件中断使能
SagUaIE	A 相发生 SAG 事件中断使能
SagUbIE	B 相发生 SAG 事件中断使能
SagUcIE	C 相发生 SAG 事件中断使能
UaEndIE	A 相退出 Peak 或 SAG 事件中断使能
UbEndIE	B 相退出 Peak 或 SAG 事件中断使能
UcEndIE	C 相退出 Peak 或 SAG 事件中断使能
INTUaIE	A 相发生 INT 事件中断使能
INTUbIE	B 相发生 INT 事件中断使能
INTUcIE	C 相发生 INT 事件中断使能
INTUIE	ABC 三相都发生 INT 事件中断使能
UaINTEndIE	A 相退出 INT 事件中断使能
UbINTEndIE	B 相退出 INT 事件中断使能
UcINTEndIE	C 相退出 INT 事件中断使能
IaOVIE	A 相发生 OverFlow 事件中断使能
IbOVIE	B 相发生 OverFlow 事件中断使能
IcOVIE	C 相发生 OverFlow 事件中断使能
InOVIE	零线发生 OverFlow 事件中断使能
IaOVEndIE	A 相退出 OverFlow 事件中断使能
IbOVEndIE	B 相退出 OverFlow 事件中断使能
IcOVEndIE	C 相退出 OverFlow 事件中断使能
InOVEndIE	零线退出 OverFlow 事件中断使能
Vpupd	各通道的半波峰值更新中断使能

EMUIF1 EMU 中断标志寄存器		基地址: 0x4001 C000 偏移地址: 388						
	Bit31	30	29	28	27	26	25	24
Read:	RMSItO	RMSUt	RMSTUp	RMSUpd	PowerUp	SPLUpdat	Energy	ZCUcLo
Write:	VIF	OVIF	dateIF	ateIF	dateIF	eIF	OVIF	stIF
Reset:	0	0		0	0	0	0	0
	23	22	21	20	19	18	17	16
Read:	ZCUBLo	ZCUaLo	CMPcIF	CMPbIF	CMPaIF	ZCInIF	ZCICIF	ZCIBIF
Write:	stIF	stIF						
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8

Read:								
Write:	ZCIaIF	ZCUcIF	ZCUBIF	ZCUaIF	HarP_NegCFIF	FundQ_NegCFIF	FundP_NegCFIF	S_NegCFIF
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	AllQ_NegCFIF	AllP_NegCFIF	HarP_PosCFIF	FundQ_PosCFIF	FundP_PosCFIF	S_PosCFIF	AllQ_PosCFIF	AllP_PosCFIF
Write:								
Reset:	0	0	0	0	0	0	0	0

注：所有中断标志都是写1清0

位名称	描述 EMU 中断标志寄存器 1 - EMUIF1
AllP_PosCFIF	全波有功通路 CF 正向脉冲中断标志
AllQ_PosCFIF	全波无功通路 CF 正向脉冲中断标志
S_PosCFIF	视在通路 CF 正向脉冲中断标志
FundP_PosCFIF	基波有功通路 CF 正向脉冲中断标志
FundQ_PosCFIF	基波无功通路 CF 正向脉冲中断标志
HarP_PosCFIF	谐波有功通路 CF 正向脉冲中断标志
AllP_NegCFIF	全波有功通路 CF 反向脉冲中断标志
AllQ_NegCFIF	全波无功通路 CF 反向脉冲中断标志
S_NegCFIF	视在通路 CF 反向脉冲中断标志
FundP_NegCFIF	基波有功通路 CF 反向脉冲中断标志
FundQ_NegCFIF	基波无功通路 CF 反向脉冲中断标志
HarP_NegCFIF	谐波有功通路 CF 反向脉冲中断标志
ZCUaIF	A 相电压过零中断标志
ZCUBIF	B 相电压过零中断标志
ZCUcIF	C 相电压过零中断标志
ZCIaIF	A 相电流过零中断标志
ZCIbIF	B 相电流过零中断标志
ZCIcIF	C 相电流过零中断标志
ZCInIF	零线电流过零中断标志
CMPaIF	电流通道模拟比较器 A 相中断标志
CMPbIF	电流通道模拟比较器 B 相中断标志
CMPcIF	电流通道模拟比较器 C 相中断标志
ZCUaLostIF	A 相电压过零丢失中断标志
ZCUBLosIF	B 相电压过零丢失中断标志
ZCUcLostIF	C 相电压过零丢失中断标志
EnergyOVIF	24 路共 48 个电能寄存器有某个寄存器发生了电能溢出中断标志 注意该中断是对 48 个电能寄存器溢出的总标志，任何一个电能溢出都可以产生 EnergyOVIF，用户需要去二级状态判断哪个电能发生溢出了
SPLUpdateIF	SPL 寄存器的更新中断标志
PowerUpdateIF	Power 寄存器的更新中断标志
RMSUpdateIF	RMS 寄存器的更新中断标志
RMSTUpdateIF	RMST 寄存器的更新中断标志

RMSUtOVIF	RMS_Ut 大于 RMSUt_THO 则触发 RMSUtOVIF 中断标志
RMSItOVIF	RMS_It 大于 RMSIt_THO 则触发 RMSItOVIF 中断标志

EMUIF2 EMU 中断标志寄存器		基地址: 0x4001 C000 偏移地址: 38C						
	Bit31	30	29	28	27	26	25	24
Read:		VpupdIn	VpupdIcI	VpupdIbI	VpupdIaI	VpupdUcI	VpupdUbI	VpupdUaI
Write:		IF	F	F	F	F	bIF	IF
Reset:	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Read:	InOVEnd	IcOVEn	IbOVEnd	IaOVEnd	InOVIF	IcOVIF	IbOVIF	IaOVIF
Write:	IF	dIF	IF	IF				
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	UcINTE	UbINTE	UaINTE	INTUaI	INTUcI	INTUbI	INTUaI	UcEndIF
Write:	ndIF	ndIF	dIF				F	
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	UbEndIF	UaEndIF	SagUcIF	SagUbIF	SagUaIF	PeakUcIF	PeakUbI	PeakUaI
Write:							F	F
Reset:	0	0	0	0	0	0	0	0

位名称	描述 EMU 中断标志寄存器 2 –EMUIF2
PeakUaIF	A 相发生 Peak 事件中断标志
PeakUbIF	B 相发生 Peak 事件中断标志
PeakUcIF	C 相发生 Peak 事件中断标志
SagUaIF	A 相发生 SAG 事件中断标志
SagUbIF	B 相发生 SAG 事件中断标志
SagUcIF	C 相发生 SAG 事件中断标志
UaEndIF	A 相退出 Peak 或 SAG 事件中断标志
UbEndIF	B 相退出 Peak 或 SAG 事件中断标志
UcEndIF	C 相退出 Peak 或 SAG 事件中断标志
INTUaIF	A 相发生 INT 事件中断标志
INTUbIF	B 相发生 INT 事件中断标志
INTUcIF	C 相发生 INT 事件中断标志
INTUIF	ABC 三相都发生 INT 事件中断标志(INTUaIF& INTUaIF& INTUaIF 与操作) 用户需要保证在某相 Uend 产生以后写清该相的 INT 标志, 以规避 INT 总中断被错误置起的可能性
UaINTEndIF	A 相退出 INT 事件中断标志
UbINTEndIF	B 相退出 INT 事件中断标志
UcINTEndIF	C 相退出 INT 事件中断标志
IaOVIF	A 相发生 OverFlow 事件中断标志

IbOVIF	B 相发生 OverFlow 事件中断标志
IcOVIF	C 相发生 OverFlow 事件中断标志
InOVIF	零线发生 OverFlow 事件中断标志
IaOVEndIF	A 相退出 OverFlow 事件中断标志
IbOVEndIF	B 相退出 OverFlow 事件中断标志
IcOVEndIF	C 相退出 OverFlow 事件中断标志
InOVEndIF	零线退出 OverFlow 事件中断标志
Vpupd	各通道的半波峰值更新中断使能

EMUIF_EnergyPOVIF 电能寄存器溢出中断寄存器			基地址: 0x4001 C000 偏移地址: 390					
	Bit23	22	21	20	19	18	17	Bit16
Read:	OV_Har	OV_Har	OV_Har	OV_Har	OV_Fun	OV_Fun	OV_Fun	OV_Fun
Write:	EPt_POSIF	EPc_POSIF	EPb_POSIF	EPa_POSIF	dEQt_POSIF	dEQc_POSIF	dEQb_POSIF	dEQa_POSIF
Reset:								
	Bit15	14	13	12	11	10	9	Bit8
Read:	OV_Fun	OV_Fun	OV_Fun	OV_Fun	OV_ESt	OV_ESc	OV_ESb	OV_ESa
Write:	dEPt_POSIF	dEPc_POSIF	dEPb_POSIF	dEPa_POSIF	POSIF	_POSIF	_POSIF	_POSIF
Reset:								
	Bit7	6	5	4	3	2	1	Bit0
Read:	OV_AIIE	OV_AIIE	OV_AIIE	OV_AIIE	OV_AIIE	OV_AIIE	OV_AIIE	OV_AIIE
Write:	Qt_POSIF	Qc_POSIF	Qb_POSIF	Qa_POSIF	Pt_POSIF	Pc_POSIF	Pb_POSIF	Pa_POSIF
Reset:								

位名称	描述 全部从属于 EnergyOVIF
OV_Energy_POS	24路电能通路的Positive正向电能寄存器溢出 写1清零

EMUIF_EnergyNOVIF 电能寄存器溢出中断寄存器			基地址: 0x4001 C000 偏移地址: 394					
	Bit23	22	21	20	19	18	17	Bit16
Read:	OV_Har	OV_Har	OV_Har	OV_Har	OV_Fun	OV_Fun	OV_Fun	OV_Fun
Write:	EPt_NEGIF	EPc_NEGIF	EPb_NEGIF	EPa_NEGIF	dEQt_NEGIF	dEQc_NEGIF	dEQb_NEGIF	dEQa_NEGIF
Reset:								
	Bit15	14	13	12	11	10	9	Bit8
Read:	OV_Fun	OV_Fun	OV_Fun	OV_Fun	OV_ESt	OV_ESc	OV_ESb	OV_ESa
Write:	dEPt_NEGIF	dEPc_NEGIF	dEPb_NEGIF	dEPa_NEGIF	NEGIF	_NEGIF	_NEGIF	_NEGIF
Reset:								

	Bit7	6	5	4	3	2	1	Bit0
Read:	OV_AllE	OV_AllE	OV_AllE	OV_AllE	OV_AllE	OV_AllE	OV_AllE	OV_AllE
Write:	Qt_NEGI F	Qc_NEG IF	Qb_NEG IF	Qa_NEG IF	Pt_NEGI F	Pc_NEGI F	Pb_NEG IF	Pa_NEGI F
Reset:								

位名称	描述
OV_Energy_NEG	全部从属于 EnergyOVIF 24路电能通路的Negative反向电能寄存器溢出 写1清零

## 22.10.7 eWatching 与工频同步系统中断

EWUIE EWU 中断使能寄存器		基地址: 0x4001CA00 偏移地址: 00H						
	Bit23	22	21	20	19	18	17	Bit16
Read:						FlickerU pdIE	Reserved	HpRmsI E
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	Reserved							
Write:	Reserved							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Reserved							
Write:	Reserved							
Reset:	0	0	0	0	0	0	0	0

位名称	描述
HpRmsIE	半波有效值更新中断使能
FlickerUpdIE	闪变数据 Pinst 更新中断使能

EWUIF EWU 中断标志寄存器		基地址: 0x4001CA00 偏移地址: 04H						
	Bit23	22	21	20	19	18	17	Bit16
Read:						FlickerU pdIF	Reserved	HpRmsI F
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	Reserved							
Write:	Reserved							

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

位名称	描述
HpRmsIF	半波有效值更新中断标志
FlickerUpdIF	闪变数据 Pinst 更新中断标志

## 22.11 工频同步系统配置

SYNC_CFG		基地址: 0x4001CC00 偏移地址: 00H						
							Bit17	Bit16
Read:							Sync_DMA_bw_mod	
Write:							e	
Reset:							0	1
	Bit15	14	13	12	11	10	9	Bit8
Read:	RmsUpd	sfo_en	SYNC_d	SYNC_D	SYNC_Rate		SYNCO	SYNC_I
Write:	Hp		islpf	FTEn			UT_En	nEn
Reset:	0	0	1	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SYNC_I	SYNC_I	SYNC_I	SYNC_U	SYNC_U	SYNC_U	SYNC_R	SYNCSy
Write:	cEn	bEn	aEn	cEn	bEn	aEn	reset	s_En
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SYNCSys_En	工频同步系统滤波器使能位 0: Disable(Default) ; 1: Enable
SYNC_Reset	工频同步系统 Sync 模块复位 0: Disable(Default) ; 1: Enable
SYNC_UaEn	SYNC 使能输出: Channel Ua 0: Disable(Default) ; 1: Enable
SYNC_UbEn	SYNC 使能输出: Channel Ub 0: Disable(Default) ;

	1: Enable
SYNC_UcEn	SYNC 使能输出: Channel Uc 0: Disable(Default); 1: Enable
SYNC_IaEn	SYNC 使能输出: Channel Ia 0: Disable(Default); 1: Enable
SYNC_IbEn	SYNC 使能输出: Channel Ib 0: Disable(Default); 1: Enable
SYNC_IcEn	SYNC 使能输出: Channel Ic 0: Disable(Default); 1: Enable
SYNC_InEn	SYNC 使能输出: Channel In 0: Disable(Default); 1: Enable
SYNCOUT_En	SYNC 的结果是否给输出给 FFT 计算路径的使能 0:Disable(Default); 1:Enable
SYNC_Rate	SYNC 给 FFT 路径输出的四种数据源选择 00: 256 点每周波 01: 128 点每周波 10: 1024 点每 10 个周波 11: 1024 点每 12 个周波 (适用 60Hz 系统)
SYNC_DFTEn	工频同步系统是否计算实时输出基波数据 0: Disable(Default); 1: Enable
SYNC_dislpf	工频同步系统的 SYNC 低通滤波器关闭 0: Enable (Default) 注意默认是 LPF 开启的; 1: Disable
sfo_en	工频同步系统的 SFO 使能: 0: SFO Bypass, 1: 使能
RmsUpdHp	全波 RMS 计算每半周波更新一次的功能使能 Bit 0: 半波有效值每半个周波更新一次(Default) 1: 全波有效值每半个周波更新一次
SYNC_DMA_bw_mode	00/01/10/11: 16/24/32/32 SynCout 数据给到 EMU InterFace 的数据格式 数据格式配置: 00: 16Bit 数据, 22Bit 位宽的 SynCoutData 取高 16Bit 补充到高 2Byte, 低 Byte 00 01: 24Bit 数据(Default), SynCoutData 高对齐填满 3Bytes, 低 2Bit 补 0 10/11: 32Bit 数据, 同 01 的 24Bit
备注	在 SYNC_DMA_BW_MODE = 16 位数据输出时, 因为是低位对齐, 为了不截

	位，需要同时配置 SYNC_DMA_rShift，如果要完全不截位，所有通道的 SYNC_DMA_rShift 需要配置为 6 在实际应用中，可以根据每个通道 RMS 值，配置适当的 SYNC_DMA_rShift 来保证足够的动态范围
--	---

SFO_Value			基地址: 0x4001CC00 偏移地址: 04H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	SFO_Value[23:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	SFO_Value[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SFO_Value[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SFO_Value	SFO 数据, Q1.23 格式, 可以配置也可以由工频同步系统自动产生填充。 注意受控于 PLL_CTRL.Bit.Manual_sfo 自动加载或者手动填写

REC_CFG 非侵入/记录数据输出			基地址: 0x4001CC00 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:				REC_lpf	REC_sig	REC_RateSelect		REC_InE
Write:				En	nalSrc			n
Reset:				0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	REC_IcE	REC_IbE	REC_IaE	REC_Uc	REC_Ub	REC_Ua	REC_res	REC_ena
Write:	n	n	n	En	En	En	et	ble
Reset:	0	0	0	0	0	0	0	0

位	功能描述
REC_enable	同频同步的 Record 数据功能总使能 0:Disable(Default); 1:Enable
REC_reset	同频同步的 Record 复位 0:Disable(Default); 1:Enable
REC_UaEn	Channel Ua 是否记录该通道 0:Disable(Default); 1:Enable
REC_UbEn	Channel Ub 是否记录该通道 0:Disable(Default); 1:Enable
REC_UcEn	Channel Uc 是否记录该通道 0:Disable(Default); 1:Enable

REC_IaEn	Channel Ia 是否记录该通道 0:Disable(Default) ; 1:Enable
REC_IbEn	Channel Ib 是否记录该通道 0:Disable(Default) ; 1:Enable
REC_IcEn	Channel Ic 是否记录该通道 0:Disable(Default) ; 1:Enable
REC_InEn	Channel In 是否记录该通道 0:Disable(Default) ; 1:Enable
REC_RateSelect	给非侵入等数据记录的路径，可设置为更新周期 00: 256/每信号周波 Default 01: 128/每信号周波 10: 64/每信号周波 11: 32/每信号周波
REC_signalSrc	给非侵入等数据记录的源选择： 0: 从工频同步系统输出 1: 从 EMU 通路 Gain 后（旁路整个同步系统）
REC_lpfEn	给非侵数据路径的 LPF 是否使能 0: disable LPF 1: enable LPF 注：该 bit12 只有在 REC_CFG.REC_RateSelect=00b(每周波 256 点)时有效，默认 bit12=0 关闭 LPF 当 REC_CFG.REC_RateSelect!=00b(每周波 128, 64, 32 点)，LPF 强制打开的，该位无效的，且不能反应 LPF 的真实状态。

DFT_CFG		基地址：0x4001CC00 偏移地址： 0CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:								
Write:								
Reset:								
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	DFT_LPF_I			X	DFT_LPF_U		
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DFT_LPF_U	工频同步系统产生基波数据，电压结果的低通系数。
DFT_LPF_I	工频同步系统产生基波数据，电流结果的低通系数。

SYNC_DMA_rShift		基地址：0x4001CC00 偏移地址： 10H						
	Bit31	30	29	28	27	26	25	Bit24
Read:					SYNC_DMA_Rshift_In			
Write:								
Reset:					0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	SYNC_DMA_Rshift_Ic				SYNC_DMA_Rshift_Ib			
Write:								

Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	SYNC_DMA_Rshift_Ia				SYNC_DMA_Rshift_Uc			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SYNC_DMA_Rshift_Ub				SYNC_DMA_Rshift_Ua			
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SYNC_DMA_Rshift	工频同步系统输出给 EMU Interface 的数据右移处理

HpRMS_Ctl_U		基地址: 0x4001CC00 偏移地址: 14H						
	Bit31	30	29	28	27	26	25	Bit24
Read:								
Write:								
Reset:								
	Bit23	22	21	20	19	18	17	Bit16
Read:		HpRMS_Offset_Uc						HpRMS_UcEnable
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:		HpRMS_Offset_Ub						HpRMS_UbEnable
Write:	X							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:		HpRMS_Offset_Ua						HpRMS_UaEnable
Write:	X							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
HpRMS_Ctl_U	各电压通道半波有效值的校正

HpRMS_Ctl_I		基地址: 0x4001CC00 偏移地址: 18H						
	Bit31	30	29	28	27	26	25	Bit24
Read:	X	HpRMS_Offset_In						HpRMS_

Write:								InEnable
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	HpRMS_Offset_Ic						HpRMS_IcEnable
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	HpRMS_Offset_Ib						HpRMS_IbEnable
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	HpRMS_Offset_Ia						HpRMS_IaEnable
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
HpRMS_Ct_I	各电流通道半波有效值的校正

PLL_CTRL		基地址: 0x4001CE00 偏移地址: 00H						
	Bit7	6	5	4	3	2	1	Bit0
Read:		PLL_locked	PLL_manual_sfo	PLL_source		PLL_reset	PLL_freeze	PLL_enable
Write:		X						
Reset:								

位	功能描述
PLL_enable	工频同步系统自动锁频功能使能 0: Disable(Default); 1: Enable
PLL_freeze	工频同步系统冻结成功标志可读可写 0: 未冻结 1: 冻结
PLL_reset	锁频功能复位 写入 1 则复位 PLL 模块, 但是该寄存器读出仍为 0
PLL_source	锁频功能所使用的电压信号源: 00/01/10/11: UA/Ub/UC/UA --Default 00
PLL_manual_sfo	SFO 是自动/手动产生 0: 自动 1: 手动
PLL_locked	工频同步系统锁定成功标志 只读 0: 未锁定 1: 锁定

PLL_Gain		基地址: 0x4001CE00 偏移地址: 04H						
	Bit31 ~ Bit16							
Read:	PLL_Gain2							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15 ~ Bit0							
Read:	PLL_Gain1							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PLL_Gain	锁频功能增益, 建议写 0x40003000

PLL_Acc		基地址: 0x4001CE00 偏移地址: 08H						
	Bit31 ~ Bit0							
Read:	PLL_Acc							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PLL_ACC	锁频功能 ACC 参数配置

SFO_EST		基地址: 0x4001CE00 偏移地址: 0CH						
	Bit31 ~ Bit0							
Read:	SFO_EST							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SFO_EST	SFO 估计值的瞬时值

LOCK_THO		基地址: 0x4001CE00 偏移地址: 10H						
	Bit31 ~ Bit0							
Read:	lock_threshold							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
LOCK_THO	PLL 锁频阈值, 建议写 0x00000FA0

AMP_THO	基地址: 0x4001CE00 偏移地址: 14H							
	Bit31 ~ Bit0							
Read:	AMP_THO							
Write:	AMP_THO							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
AMP_THO	锁频功能生效的信号阈值, 建议写 0x00000064

TED_Amnt	基地址: 0x4001CE00 偏移地址: 18H							
	Bit7 ~ Bit0							
Read:	TED_rShiftAmnt							
Write:	TED_rShiftAmnt							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TED_Amnt	建议写 0x04

## 22.12 校表过程

### 22.12.1 功率法校表

注 1:

由于脉冲法是直接读取机台误差计算校表参数, 校准精度要比功率法更高。但是小信号时出脉冲较慢, 因此脉冲法相比功率法也会更慢。用户可按需求使用下述两种方法。

注 2:

校正完通道 gain 之后, 为保证功率寄存器更新正确值, 需要等待至少 400ms, 再进行其他校表步骤。

功率法校表

功率源施加额定电压、电流, 功率因数为 0.5L 条件下, 根据计量芯片内部寄存器有功、无功功率值、有效值和标准表 (功率表) 有功、无功值、有效值实现精度校正。

步骤如下:

### 22.12.1.1 通道 Gain 的确定

目的：当  $U_a/U_b/U_c$  三相施加相等的电压(例 220V)时，通过通道 Gain 的校准，使芯片三路电压输入通道采集到相同的数据。从而消除三相采样通道之间的元器件的差异。同理三路电流。

通道 Gain 的确定有两种方法。一、表间一致。二、表内一致。

用户可根据需求选择通道 Gain 的确定方法。

#### (1) 表间一致

先计算出三相 U 通道、三相 I 通道的标准值。然后调整 Gain $U_a/U_b/U_c$  把三相电压通道信号校准为和计算的标准值相等。调整 Gain $I_a/I_b/I_c$  把三相电流通道信号校准为和计算的标准值相等。

标准值  $U_{rms\_std}$ 、 $I_{rms\_std}$  的计算：

$$U_{rms\_std} = \frac{U_n * 2^{23}}{0.93 * K_u}$$

$$I_{rms\_std} = \frac{I_B * R_{ct} * 2^{23}}{0.93 * Trans}$$

Un	三相额定电压
IB	三相额定电流
Ku	电压通道采样分压变比
Trans	电流通道 CT 互感器变比
Rct	电流通道 CT 后端的采样电阻

例：电表用大小电阻串对电压进行分压采样，6 个 200K $\Omega$  大电阻+1 个 1.2K $\Omega$  小电阻，那么  $K_u = (1.2K\Omega + 6 * 200K\Omega) / 1.2K\Omega = 1001$

电表用 CT 互感器变比 5A:2mA 规格进行电流采样，CT 后端的采样电阻值为两个 3.9R。那么  $Trans = 5A / 2mA = 2500$  ;  $R_{ct} = 3.9R * 2 = 7.8$

电表的  $U_n = 220V$ ， $I_B = 5A$

由此计算

$$U_{rms\_std} = 220 * 2^{23} / (0.93 * 1001) = 1982419$$

$$I_{rms\_std} = 5 * 7.8 * 2^{23} / (0.93 * 2500) = 140712$$

在计算得  $U_{rms\_std}$  和  $I_{rms\_std}$  以后，做如下校准：

在  $U_a/U_b/U_c = U_N$ ， $I_a/I_b/I_c = I_B$  负载点：进行  $U_{Gain\_A,B,C}$  和  $I_{Gain\_A,B,C}$  的校正。

A 相为例：

读出 A 相电压、电流有效值寄存器值  $All\_RMS\_U_a$ 、 $All\_RMS\_I_a$ (需要 10 次平均)分别为  $U_{rms\_a}$ 、 $I_{rms\_a}$ 。

$$\begin{cases} U_{gain\_a} = \left( \frac{U_{rms\_std}}{U_{rms\_a}} - 1 \right) * 2^{15}, U_{rms\_std} > U_{rms\_a} \\ U_{gain\_a} = \left( \frac{U_{rms\_std}}{U_{rms\_a}} - 1 \right) * 2^{15} + 2^{16}, U_{rms\_std} < U_{rms\_a} \end{cases}$$

$$\begin{cases} I_{gain\_a} = \left( \frac{I_{rms\_std}}{I_{rms\_a}} - 1 \right) * 2^{15}, I_{rms\_std} > I_{rms\_a} \\ I_{gain\_a} = \left( \frac{I_{rms\_std}}{I_{rms\_a}} - 1 \right) * 2^{15} + 2^{16}, I_{rms\_std} < I_{rms\_a} \end{cases}$$

将求得的  $U_{gain\_a}$  和  $I_{gain\_a}$  转换成 16 进制填入  $Gain\_U_a$ 、 $Gain\_I_a$  寄存器中。

B 相、C 相校正方法和 A 相一致。

#### (2) 表内一致

把 Ua/Ia 的采样结果作为标准值。通过调整 GainUb/GainUc 把 Ub/Uc 通道信号校准为和 Ua 采样结果相等。调整 GainIb/GainIc 把 Ib/Ic 通道信号校准为和 Ia 采样结果相等。

该方式下，

标准源输出三相 Un、Ib，相位角度为 0.5L（例 Un=220V、Ib=5A），此时读出三相电流有效值以及电压有效值寄存器值分别为 All\_RMS\_Ua/b/c、All\_RMS\_Ia/b/c。每个参数需要 10 次并取平均值，以提高数据计算精度。

通过如下方式计算 Gain:

$$\begin{cases} U_{gain\_b} = \left(\frac{Urms\_a}{Urms\_b} - 1\right) * 2^{15}, Urms\_a > Urms\_b \\ U_{gain\_b} = \left(\frac{Urms\_a}{Urms\_b} - 1\right) * 2^{15} + 2^{16}, Urms\_a < Urms\_b \end{cases}$$

$$\begin{cases} I_{gain\_b} = \left(\frac{Irms\_a}{Irms\_b} - 1\right) * 2^{15}, Irms\_a > Irms\_b \\ I_{gain\_b} = \left(\frac{Irms\_a}{Irms\_b} - 1\right) * 2^{15} + 2^{16}, Irms\_a < Irms\_b \end{cases}$$

将求得的 Ugain\_b 和 Igain\_b 转换成 16 进制填入寄存器 Gain\_Ub、Gain\_Ib 中。

C 相校正方法和 B 相一致。

### 22.12.1.2 HFConst 的确定

Vu: 额定电压输入时，电压通道的电压（引脚上电压×放大倍数）

Vi: 额定电流输入时，电流通道的电压（引脚上电压×放大倍数）

Un: 额定输入的电压

IB: 额定输入的电流

EC: 电表常数

$$HFConst = 9.1 * Vu * Vi * 10^{11} / (EC * Un * IB)$$

例：目前电压通道电阻串分压：7 个 330K+1 个 1.2K；

目前电流通道 CT 变比 5A:2mA，采样电阻为两个 3.9R

EC = 3200imp/kwh、Un=220V、IB = 5A

则计算

$$Vu = 1.2K / (1.2K + 7 * 330K) * 220V = 0.114$$

$$Vi = 5A * 1 / 2500(CT) * 2 * 3.9R = 0.0156$$

$$HFConst = 9.1 * 10^{11} * Vu * Vi / (EC * Un * IB) = 460 = 0x01CC，填入 HFConstAF 寄存器$$

注：根据外围硬件调整 HFCONST 值，可以适当加上阈值

### 22.12.1.3 电压、电流、功率转换系数的确定

表台加  $U_n$   $I_b$ ，读出计量芯片的电压有效值  $Rms\_U$ 、电流有效值  $Rms\_I$ ，由于经过了通道 Gain 的增益校正，因此三相电压都具备相同的转换系数，同理电流。

计算：

(1) 电压转换系数:  $K_{rmsU} = U_n / Rms\_U$

该系数与电压有效值寄存器的值相乘即得到输入的电压 (V)

(2) 电流转换系数:  $K_{rmsI} = I_B / Rms\_I$

该系数与电流寄存器的值相乘即得到输入的电流值 (A)

(3) 功率转换系数

$$Kpower = \frac{343.323}{HFConst * EC}$$

读取寄存器的值  $PowerP\_Register$  乘以系数  $Kpower$ ，得到输入的功率值  $Preal$  (W)，即  $Power\_Reg * Kpower = Preal$ 。

### 22.12.1.4 GP/Gphs 校正

如下 a.b.c 三步

a. 先进行相位校正

(1) PQ 方式校正相位：

表台加 A,B,C 三相的  $U_n=220V$ 、 $I_b=5A$ 、 $0.5L$  相位负载点，读取表台标准表的有功功率  $P_{std}$ 、无功功率值  $Q_{std}$ ，读取 A,B,C 三相的有功、无功功率寄存器值  $PowerP$ 、 $PowerQ$ （注意：分别是 A,B,C 三相的功率寄存器）。

注：要保证表台源输出稳定，读取方式为三相有功无功寄存器轮询读取，间隔要短，且三相各自的有功、无功寄存器值读 10 次取平均。

按如下公式计算角度偏差误差： $\theta$  (A 相为例)

$$\theta = \frac{P_{std} * PowerQ - Q_{std} * PowerP}{PowerP * P_{std} + PowerQ * Q_{std}}$$

如果  $\theta \geq 0$ ， $GPhs = \theta \times 2^{15}$ 。否则， $GPhs = 2^{16} + \theta \times 2^{15}$ ，

填入  $All\_GphsA$  相位校正寄存器，且  $All\_GphsA0/A1/A2$  均需写入相同值。

同理 B 相、C 相。

(2) 移采样点方式校正相位：（相位切换时响应更快，但校正分辨率不及 PQ 方式）

（初始误差不超过  $\pm 4.25\%$ ，即不超过一个大拍）

EMUCLK	OSR	公式
819.2K		如果 $\theta$ 为负， $\theta * (-1.732) / 0.000665$ 填入电流移采样校正

		Decshift_IxL(x = A/B/C) 如果 $\theta$ 为正, $\theta*(1.732) / 0.000665$ 填入电压移采样校正 Decshift_UxL(x = A/B/C)
1.638M		如果 $\theta$ 为负, $\theta*(-1.732) / 0.0003325$ 填入电流移采样校正 Decshift_IxL(x = A/B/C) 如果 $\theta$ 为正, $\theta*(1.732) / 0.0003325$ 填入电压移采样校正 Decshift_UxL(x = A/B/C)

b. 计算相位校正后的功率值  $P', Q'$

分别计算出 A,B,C 三相相位校正后的有功无功输入值:

$$P' = (PowerP + PowerQ * \theta) * Kpower$$

$$Q' = (PowerQ - PowerP * \theta) * Kpower$$

c. 功率增益校正

以 A 相为例:

$$P_{gain} = \frac{P_{std}}{P'} - 1$$

如果  $P_{gain} \geq 0$ , 则  $GP1 = INT[P_{gain} \times 215]$

否则  $P_{gain} < 0$ , 则  $GP1 = INT[216 + P_{gain} \times 215]$

将计算出值写入有功增益寄存器 All\_GPa, 一般情况下, All\_GPa/All\_GQa/All\_GSa 写一样的值即可。

同理 B 相、C 相

### 22.12.1.5 小信号 OFFSET 校正

(1) 有功功率 Poffset 校正

建议可将 5%Ib 点功率值作为小信号 POFFSET 的校正点。

表台加 A,B,C 三相的  $U_n$  5%Ib 信号, 相位为有功 1.0。为提高精度, 建议至少循环连续读取 10 次左右计量芯片有功功率寄存器和表台标准有功功率值(注意: 分别是 A,B,C 三相的功率寄存器), 分别取平均得到计量芯片功率寄存器平均值 P 和表台标准表平均功率值 Pstd, 计算功率小信号 POFFSET 值。

$$P_{offset} = P_{std} / Kpower - P$$

P: 芯片内部功率寄存器循环多次读取得到的平均值;

Pstd: 表台标准表有功功率循环多次读取得到的平均值;

Kpower: 功率转换系数 Kpower;

计算的 Poffset 值转成十六进制, 并取反 直接填入 All\_PowerOffset\_Pa/b/c 寄存器。

(2) 无功功率 Qoffset 校正(如不需要无功可不校准)

可将 5%Ib 点功率值作为小信号 QOFFSET 的校正点。

表台加 Un 5%Ib 无功 1.0, 为提高精度, 建议至少循环连续读取 20 次左右计量芯片无功功率寄存器和表台标准无功功率值, 分别取平均得到计量芯片功率寄存器平均值 Q 和表台标准表平均功率值 Qstd, 计算功率小信号 QOFFSET 值。

$$Q_{offset} = Q_{std} / K_{power} - Q$$

计算的 Qoffset 值转成十六进制, 并取反 填入 All\_PowerOffset\_Qa/b/c 寄存器寄存器。

Q: 芯片内部功率寄存器循环多次读取得到的平均值;

Qstd: 表台标准表有功功率循环多次读取得到的平均值;

Kpower: 功率转换系数 Kpower;

## 22.12.2 脉冲法校表

### 22.12.2.1 通道 Gain 的确定

目的: 当 Ua/Ub/Uc 三相施加相等的电压(例 220V)时, 通过通道 Gain 的校准, 使芯片三路电压输入通道采集到相同的数据。从而消除三相采样通道之间的元器件的差异。同理三路电流。

通道 Gain 的确定有两种方法。一、表间一致。二、表内一致。

用户可根据需求选择通道 Gain 的确定方法。

(1) 表间一致

先计算出三相 U 通道、三相 I 通道的标准值。然后调整 GainUa/Ub/Uc 把三相电压通道信号校准为和计算的标准值相等。调整 GainIa/Ib/Ic 把三相电流通道信号校准为和计算的标准值相等。

标准值 Urms\_std、Irms\_std 的计算:

$$U_{rms\_std} = \frac{U_n * 2^{23}}{0.93 * K_u}$$

$$I_{rms\_std} = \frac{I_B * R_{ct} * 2^{23}}{0.93 * Trans}$$

Un	三相额定电压
IB	三相额定电流
Ku	电压通道采样分压变比
Trans	电流通道 CT 互感器变比
Rct	电流通道 CT 后端的采样电阻

例: 电表用大小电阻串对电压进行分压采样, 6 个 200KΩ 大电阻+1 个 1.2KΩ 小电阻, 那么 Ku = (1.2KΩ+6\*200KΩ)/1.2KΩ = 1001

电表用 CT 互感器变比 5A:2mA 规格进行电流采样, CT 后端的采样电阻值为两个 3.9R。那么 Trans = 5A/2mA = 2500 ; Rct = 3.9R \* 2 = 7.8

电表的 Un=220V, IB = 5A

由此计算

$$U_{rms\_std} = 220 * 2^{23} / (0.93 * 1001) = 1982419$$

$$I_{rms\_std} = 5 * 7.8 * 2^{23} / (0.93 * 2500) = 140712$$

在计算得  $Urms\_std$  和  $Irms\_std$  以后，做如下校准：

在  $Ua/Ub/Uc=UN$ ， $Ia/Ib/Ic=IB$  负载点：进行  $UGain\_A,B,C$  和  $IGain\_A,B,C$  的校正。

A 相为例：

读出 A 相电压、电流有效值寄存器值  $All\_RMS\_Ua$ 、 $All\_RMS\_Ia$ (需要 10 次平均)分别为  $Urms\_a$ 、 $Irms\_a$ 。

$$\begin{cases} Ugain\_a = \left(\frac{Urms\_std}{Urms\_a} - 1\right) * 2^{15}, Urms\_std > Urms\_a \\ Ugain\_a = \left(\frac{Urms\_std}{Urms\_a} - 1\right) * 2^{15} + 2^{16}, Urms\_std < Urms\_a \end{cases}$$

$$\begin{cases} Igain\_a = \left(\frac{Irms\_std}{Irms\_a} - 1\right) * 2^{15}, Irms\_std > Irms\_a \\ Igain\_a = \left(\frac{Irms\_std}{Irms\_a} - 1\right) * 2^{15} + 2^{16}, Irms\_std < Irms\_a \end{cases}$$

将求得的  $Ugain\_a$  和  $Igain\_a$  转换成 16 进制填入  $Gain\_Ua$ 、 $Gain\_Ia$  寄存器中。

B 相、C 相校正方法和 A 相一致。

## (2) 表内一致

把  $Ua/Ia$  的采样结果作为标准值。通过调整  $GainUb/GainUc$  把  $Ub/Uc$  通道信号校准为和  $Ua$  采样结果相等。调整  $GainIb/GainIc$  把  $Ib/Ic$  通道信号校准为和  $Ia$  采样结果相等。

该方式下，

标准源输出三相  $Un$ 、 $Ib$ ，相位角度为  $0.5L$ （例  $Un=220V$ 、 $Ib=5A$ ），此时读出三相电流有效值以及电压有效值寄存器值分别为  $All\_RMS\_Ua/b/c$ 、 $All\_RMS\_Ia/b/c$ 。每个参数需要 10 次并取平均值，以提高数据计算精度。

通过如下方式计算  $Gain$ ：

$$\begin{cases} Ugain\_b = \left(\frac{Urms\_a}{Urms\_b} - 1\right) * 2^{15}, Urms\_a > Urms\_b \\ Ugain\_b = \left(\frac{Urms\_a}{Urms\_b} - 1\right) * 2^{15} + 2^{16}, Urms\_a < Urms\_b \end{cases}$$

$$\begin{cases} Igain\_b = \left(\frac{Irms\_a}{Irms\_b} - 1\right) * 2^{15}, Irms\_a > Irms\_b \\ Igain\_b = \left(\frac{Irms\_a}{Irms\_b} - 1\right) * 2^{15} + 2^{16}, Irms\_a < Irms\_b \end{cases}$$

将求得的  $Ugain\_b$  和  $Igain\_b$  转换成 16 进制填入寄存器  $Gain\_Ub$ 、 $Gain\_Ib$  中。

C 相校正方法和 B 相一致。

## 22.12.2.2 HFConst 的确定

$Vu$ ：额定电压输入时，电压通道的电压（引脚上电压×放大倍数）

$Vi$ ：额定电流输入时，电流通道的电压（引脚上电压×放大倍数）

$Un$ ：额定输入的电压

$IB$ ：额定输入的电流

$EC$ ：电表常数

$$HFConst = 9.1 * Vu * Vi * 10^{11} / (EC * Un * IB)$$

例：目前电压通道电阻串分压：7 个 330K+1 个 1.2K；  
目前电流通道 CT 变比 5A:2mA，采样电阻为两个 3.9R

$$EC = 3200\text{imp/kwh}、U_n=220\text{V}、IB = 5\text{A}$$

则计算

$$V_u = 1.2\text{K}/(1.2\text{K}+7*330\text{K})*220\text{V} = 0.114$$

$$V_i = 5\text{A} * 1/2500(\text{CT})*2*3.9\text{R} = 0.0156$$

$\text{HFConst} = 9.1*10^{11}*V_u*V_i/(EC*U_n*IB) = 460 = 0x01\text{CC}$ ，填入 HFConstAF 寄存器

注：根据外围硬件调整 HFCONST 值，可以适当加上阈值

### 22.12.2.3 电压、电流、功率转换系数的确定

表台加  $U_n I_b$ ，读出计量芯片的电压有效值  $R_{ms\_U}$ 、电流有效值  $R_{ms\_I}$ ，由于经过了通道 Gain 的增益校正，因此三相电压都具备相同的转换系数，同理电流。

计算：

(1)电压转换系数:  $K_{rmsU} = U_n / R_{ms\_U}$

该系数与电压有效值寄存器的值相乘即得到输入的电压 (V)

(2)电流转换系数:  $K_{rmsI} = I_B / R_{ms\_I}$

该系数与电流寄存器的值相乘即得到输入的电流值 (A)

(3)功率转换系数

$$\text{Kpower} = \frac{343.323}{\text{HFConst} * EC}$$

读取寄存器的值 PowerP\_Register 乘以系数 Kpower，得到输入的功率值 Preal (W)，即  $\text{Power\_Reg} * \text{Kpower} = \text{Preal}$ 。

### 22.12.2.4 GP/Gphs 校正

已知：A 相为例，额定输入、功率因数为 1.0 时标准表上读出误差为 err%

计算公式：

$$\text{Pgain} = \frac{-err\%}{1 + err\%}$$

如果  $\text{Pgain} \geq 0$ ，则  $\text{GP}_a = \text{INT}[\text{Pgain} * 2^{15}]$

否则  $\text{Pgain} < 0$ ，则  $\text{GP}_a = \text{INT}[2^{16} + \text{Pgain} * 2^{15}]$

将计算出值写入有功增益寄存器 All\_GP<sub>a</sub>，一般情况下，All\_GP<sub>a</sub>/All\_GQ<sub>a</sub>/All\_GS<sub>a</sub> 写一样的值即可。同理 B 相、C 相。

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。相位校正有

两种方式可选择：PQ 方式和移采样点方式。

#### 22.12.2.4.1 PQ 方式校正相位：

在芯片内部 0.5L 相位夹角时，有功和无功功率的计算公式如下：

$$\text{PowerP}' = \text{PowerP} + \text{PowerQ} \times \text{GPhs}/2^{15}$$

$$\text{PowerQ}' = \text{PowerQ} - \text{PowerP} \times \text{GPhs}/2^{15}$$

让  $\theta = \text{GPhs}/2^{15}$  作为需要校正的相位误差。

已知：0.5L 处标准表误差读数为 err%

相位补偿公式：

$$\theta = \frac{-\text{err}\%}{1.732}$$

如果  $\theta \geq 0$ ， $\text{GPhs} = \theta \times 2^{15}$ 。否则， $\text{GPhs} = 2^{16} + \theta \times 2^{15}$

填入 All\_GphsA 相位校正寄存器，且 All\_GphsA0/A1/A2 均需写入相同值。同理 B 相、C 相。

#### 22.12.2.4.2 移采样点方式校正相位：（校正精度比 PQ 方式低）

（初始误差不超过±4.25%，即不超过一个大拍）

EMUCLK	OSR	公式
819.2K		如果 err% 为正，err% / 0.0665 填入电流移采样校正 如果 err% 为负，-err% / 0.0665 填入电压移采样校正
1.638M		如果 err% 为正，err% / 0.03325 填入电流移采样校正 如果 err% 为负，-err% / 0.03325 填入电压移采样校正

#### 22.12.2.5 小信号 Poffset 校正

Poffset 的校准是在小信号 5%IB 电流时进行的。

$$\text{Poffset} = \text{Preal} * \text{EC} * \text{HFConst} * (\text{err}) / 343.323$$

其中 Preal 为功率源显示的当前真实功率。

err 为当前读取到的误差，例 err=0.1% 则应该以 0.001 参与计算

计算的 Poffset 值转成十六进制，直接填入 All\_PowerOffset\_Pa/b/c 寄存器。

## 23 EWU 误差自监测

芯片提供高稳定性的电压、电流信号源，多频点输出，可多路同时监测。  
用户可直接调用钜泉库函数即可实现。

详细见单独的说明文档。

## 24 Cortex-M0 内核简要说明

### 24.1 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。

M0 不支持内存保护模块(MPU)。

### 24.2 系统定时器 SysTick

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

比如 Fcpu 为 11010048Hz，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 (11010048/10);

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

### 24.3 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

### 24.4 CMSIS 函数说明

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

CMSIS 函数	函数说明
----------	------

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

注: 中断号 `IRQn` 详细见 8.1 中断向量说明

## 25 电气规格

### 25.1 极限参数

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	2.0		5.5	V
Vi	输入电压	所有的数字引脚	-0.3		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		3	V
Vib	ADCBAT 输入极限电压	测试 ADCBAT 引脚输入电压, 高于此电压可能会导致该引脚损坏	-0.3		5	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
VESDhbm	静态 ESD (HBM)	芯片所有的引脚	-4000		+4000	V
VESDmm	MM 静态 ESD				400	V

### 25.2 功耗参数

#### 25.2.1 模块功耗

符号	参数说明	测试条件	最小	典型	最大	单位
Ivccdet	VCC 检测模块功耗	常温, VCC=5.0V		<b>17</b>		uA
Ibor	BOR 检测模块功耗	常温, VCC=5.0V		<b>6.8</b>		uA
Ilvdin	LVDIN 检测模块功耗	常温, VCC=5.0V, 单个通道开启		<b>5.6</b>		uA
Ihrc	HRC 功耗	常温, VCC=5.0V		<b>129</b>		uA
Ilrc	LRC 功耗	-45°C~85°C, VCC=5.0V	<b>0.4</b>	<b>0.5</b>	<b>0.6</b>	uA
Ipll	PLL 功耗	常温, VCC=5.0V		<b>246</b>		uA
Irtc	RTC 功耗 (包括 TPS 分时)			<b>1.3</b>		uA

## 25.2.2 运行功耗

符号	参数说明	测试条件	最小	典型	最大	单位
Ihold	Hold 功耗			<b>11.1</b>		uA
Isleep	Sleep 功耗			<b>3.7</b>		uA
Irunhrc	HRC 下运行功耗	高频 RC 配置为 19.6M, CLKOUT 关闭, 所有数字模块打开 (EMU 开启) \打开 (EMU 未开启) \关闭		<b>17.33</b> <b>/9.44</b> <b>/6.79</b>		mA
		高频 RC 配置为 9.8M, CLKOUT 关闭, 所有数字模块打开 (EMU 开启) \打开 (EMU 未开启) \关闭		<b>12.92</b> <b>/5.18</b> <b>/3.51</b>		mA
Irunpll	PLL 下运行功耗	CPU 运行 PLL 时钟 39.32M, 所有数字模块打开 (EMU 开启, CACHE 开启) \打开 (EMU 关闭, CACHE 开启) \打开 (EMU 关闭, CACHE 关闭) \关闭		<b>23.51</b> <b>/16.43</b> <b>/11.93</b> <b>/10.71</b>		mA
		CPU 运行 PLL 时钟 19.66M, 所有数字模块打开 (EMU 开启, CACHE 开启) \打开 (EMU 关闭, CACHE 开启) \打开 (EMU 关闭, CACHE 关闭) \关闭		<b>18.04</b> <b>/10.17</b> <b>/7.65</b> <b>/6.91</b>		mA
Irunosc	Crystal32k 下运行功耗			<b>700</b>		nA

## 25.3 DC/AC 参数

### 25.3.1 存储器

符号	参数说明	测试条件	最小	典型	最大	单位
	Flash 空间大小			<b>256K</b>		bytes
	Information Block 空间大小			<b>1K</b>		bytes
	Ram 空间大小			<b>80K</b>		bytes
	Flash 字节读取时间			<b>40</b>		ns
	Flash 字节写时间				<b>20</b>	us
	Flash 页擦除时间				<b>2</b>	ms
	Flash 全擦除时间				<b>10</b>	ms
	Code Flash 页面大小			<b>1k</b>		bytes/page

Information	Block 页面大小			1k		bytes/page
	擦写次数		100,000			次
	数据保持时间		20			years
	操作温度		-40		105	°C

### 25.3.2 GPIO

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, BOR 关闭	2.0	3.3	5.5	V
		输入电源引脚上的电压, BOR 开启	2.0	3.3	5.5	V
Vih1	高电平输入电压	Reset 引脚, TEST 引脚	0.8VCC			V
Vih2		支持 TTL 输入管脚	0.6VCC*1			V
Vih3		除了电源和地, 除了 Reset, TEST, TTL 输入引脚之外的所有引脚	0.7VCC			V
Vil1	低电平输入电压	Reset 引脚, TEST 引脚			0.2VCC	V
Vil2		除了电源和地, 除了 Reset, TEST 之外的所有引脚			0.2VCC	V
Ioh1 大电流 IO	高电平输出电流	VCC=5V I/O 口上输出高电压 Voh 降低到 0.9VCC 测试引脚为: 支持 source 大电流驱动的 IO *2	12			mA
Ioh2		VCC=5.0V I/O 口上输出高电压 Voh 降低到 0.9VCC 除了支持 source 大电流驱动 GPIO, 以及 PJx*4 引脚以外的所有 GPIO 引脚*2	5			mA
Iol1 大电流 IO	低电平输出电流	VCC=5.0V I/O 口上输出低电压 Vol 升高到 0.1VCC 测试引脚为: 支持大电流驱动的所有 GPIO 引脚*2	30			mA
Iol2		VCC=5.0V I/O 口上输出低电压 Vol 降低到 0.1VCC	5			mA

		除了支持大电流驱动的 GPIO, 以及 PJx*4 引脚以外的所有 GPIO 引脚*2				
Rpull	输入上拉电阻	VCC=5.0V, 除 test 其余 GPIO 引脚		88		kΩ
		VCC=5.0V,test 引脚		10		kΩ
Iif	输入漏电流	IO 配置为输入, 上拉关闭, 引脚接 VCC 或 GND			0.1	uA
Iod	开漏输出漏电流	IO 配置为输出高, 开漏开启, 引脚接 GND			0.1	uA

注:

\*1 支持 TTL 输入管脚参见章节“引脚定义”;

\*2 支持大电流驱动的 IO 参见章节“支持大电流驱动的 IO”。

\*3 部分引脚支持 5v Tolerance 功能, 参见章节“引脚定义”。

\*4 PJx 包含 PJ0,PJ1。低电平和高电平输出电流大小均为 1.5mA。

### 25.3.3 CMU

符号	参数说明	测试条件	最小	典型	最大	单位
Fpll	内部 PLL 时钟频率			39.321 6		MHz
	C-C jitter	常温		469		ps
	PLL 锁定时间					ms
	PLL 环路倍频比			1344		
	PLL 工作电压	工作温度-40°C~85°C	1.5		5.5	V
	PLL 模块功耗					uA
Fosc	外部 32.768K 晶体时钟频率			32.768	*1	KHz
	Fosc 起振时间			500		ms
	OSC 工作电压	工作温度-40°C~85°C	1.5		5.5	V
	OSC 模块功耗			0.7		uA
Fhrc	内部高频 RC 时钟频率			20	*2	MHz
	HRC 频率可调范围	寄存器 HRCADJ 调整, 7bit 控制位				MHz
	HRC C-C jitter	常温		140		ps
	HRC 起振时间			1.15		ms
	HRC 工作电压	工作温度-40°C~85°C	2.0		5.5	V
	HRC 模块功耗			129.5		uA
Flrc	内部低频 RC 时钟频率	工作温度-40°C~85°C	20	32	60	KHz

	起振时间			41.15		us
	LRC 工作电压	工作温度-40°C~85°C	1.5		5.5	V
	LRC 模块功耗			0.5		uA

\*1: 内封晶体类芯片, 晶体型号 SC-32S, 详细参数见相关规格书。

### 25.3.4 PMU

符号	参数说明	测试条件	最小	典型	最大	单位
POR	上电复位阈值			1		V
	最慢上电速率			1		V/ms
LBOR	LBOR 复位阈值	上电翻转电压		1.769	2.1	V
		下点翻转电压	1.2	1.648	1.95	V
		迟滞电压		124		mV
	最快上电速率			14		V/ms
	功耗			150		nA
VCC_DET	VCC 检测阈值电压	常温, 中心值, VCC_LVL[3:0] = 0000		2.406		V
		迟滞电压		187		mV
		常温, 中心值, VCC_LVL[3:0] = 0001		2.605		V
		迟滞电压		192		mV
		常温, 中心值, VCC_LVL[3:0] = 0010		2.839		V
		迟滞电压		203		mV
		常温, 中心值, VCC_LVL[3:0] = 0011		3.017		V
		迟滞电压		217		mV
		常温, 中心值, VCC_LVL[3:0] = 0100		3.229		V
		迟滞电压		221		mV
		常温, 中心值, VCC_LVL[3:0] = 0101		3.402		V
		迟滞电压		228		mV
		常温, 中心值, VCC_LVL[3:0] = 0110		3.661		V
		迟滞电压		238		mV
		常温, 中心值, VCC_LVL[3:0] = 0111		3.814		V
		迟滞电压		243		mV
		常温, 中心值, VCC_LVL[3:0] = 1000		4.056		V

		迟滞电压		<b>252</b>		mV	
		常温, 中心值, VCC_LVL[3:0] = 1001		<b>4.290</b>		V	
		迟滞电压		<b>261</b>		mV	
		常温, 中心值, VCC_LVL[3:0] = 1010		<b>4.438</b>		V	
		迟滞电压		<b>264</b>		mV	
		常温, 中心值, VCC_LVL[3:0] = 1011		<b>4.644</b>		V	
		迟滞电压		<b>268</b>		mV	
		常温, 中心值, VCC_LVL[3:0] = 1100		<b>4.816</b>		V	
		迟滞电压		<b>271</b>		mV	
		常温, 中心值, VCC_LVL[3:0] = 1101/1110/1111		<b>5.053</b>		V	
		迟滞电压		<b>273</b>		mV	
	VCC_DET 抗噪滤波	-45°C~85°C	<b>150</b>	<b>250</b>	<b>500</b>	us	
	VCC 检测模块功耗	常温, VCC=5.0V		<b>13.4</b>		uA	
BOR_DET	BOR 检测阈值电压	常温, 中心值, BOR_LVL[1:0] = 00		<b>2.226</b>		V	
		迟滞电压		<b>139</b>		mV	
		常温, 中心值, BOR_LVL [1:0] = 01		<b>2.023</b>		V	
		迟滞电压		<b>126</b>		mV	
		常温, 中心值, BOR_LVL [1:0] = 10		<b>2.828</b>		V	
		迟滞电压		<b>180</b>		mV	
		常温, 中心值, BOR_LVL [1:0] = 11		<b>2.622</b>		V	
		迟滞电压		<b>166</b>		mV	
		BOR 抗噪滤波	-45°C~85°C	<b>150</b>	<b>250</b>	<b>500</b>	us
		BOR 检测模块功耗	常温, VCC=5.0V		<b>6.8</b>		uA
LVD_DET	低电压检测阈值	2.0V<VCC<5.5V,常温		<b>1.211</b>		V	
		迟滞电压		<b>136</b>		mV	
	LVDIN_DET 抗噪滤波	-45°C~85°C		<b>180</b>		us	
	LVDIN 检测模块功耗	常温, VCC=5.0V,单个通道开启		<b>2.3</b>		uA	
LDO_1P5	内部 1.5v 稳压源输出	-45°C~85°C, VCC=2.0~5.5v		<b>1.511</b>		V	

	驱动电流	-45°C~85°C, VCC=2.0~5.5v		<b>162.9</b>		mA
	外置负载电容			<b>0.1</b>		uF
	静态功耗			<b>5</b>		uA
LDO_LP	内部低功耗稳压源输出	-45°C~85°C, VCC=2.0~5.5v		<b>1.414</b>		V
	外置负载电容			<b>0.1</b>		uF
	静态功耗			<b>0.6</b>		uA

### 25.3.5 WDT

符号	参数说明	测试条件	最小	典型	最大	单位
	WDT 定时时间	最小时间 *1		<b>64</b>		ms
		最大时间		<b>16384</b>		ms
	WDT 误差	同 LRC 精度	<b>-39.0</b>		<b>83.1</b>	%
	功耗				<b>1</b>	uA

\*1: WDT 时钟源固定为内部LRC 时钟

### 25.3.6 TIMER

符号	参数说明	测试条件	最小	典型	最大	单位
	定时器位宽			<b>16</b>		Bit
	时钟分频		<b>1</b>		<b>65536</b>	
	PWM 周期		<b>1</b>		<b>65536</b>	Ftmer clk
	PWM 占空比		<b>1</b>		<b>65536</b>	Ftmer clk

### 25.3.7 AES/GHASH/RAND/CRC/ECC

符号	参数说明	测试条件	最小	典型	最大	单位
	AES 密钥长度		<b>128</b>	<b>192</b>	<b>256</b>	bit
	AES 加速器运算时间	Fsys=PLL22M, 单次加/解密			<b>10</b>	us
	Ghash 乘法加速器运算位宽			<b>128</b>		bit
	Rand 发生器输出位宽			<b>32</b>		bit
	CRC 支持写入数据位宽		<b>8</b>	<b>16</b>	<b>32</b>	bit
	CRC 运算时间	8bit 数据写入		<b>1</b>		Fcpu clk
		16bit 数据写入		<b>2</b>		Fcpu clk
		32bit 数据写入		<b>4</b>		Fcpu clk

	CRC 支持多项式	CRC-16/IBM	$x^{16}+x^{15}+x^2+1$		
		CRC-16/MAXIM	$x^{16}+x^{15}+x^2+1$		
		CRC-16/USB	$x^{16}+x^{15}+x^2+1$		
		CRC-16/MODBUS	$x^{16}+x^{15}+x^2+1$		
		CRC-16/CCITT	$x^{16}+x^{12}+x^5+1$		
		CRC-16/CCITT-FALSE	$x^{16}+x^{12}+x^5+1$		
		CRC-16/X25	$x^{16}+x^{12}+x^5+1$		
		CRC-16/XMODEM	$x^{16}+x^{12}+x^5+1$		
		CRC-32	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$		
		CRC-32/MPEG-2	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$		
	ECC 数据位宽				256 bit
	ECDSA 签名时间	系统时钟为 PLL20M		30.4	ms
	ECDSA 认证时间	系统时钟为 PLL20M		59.8	ms

### 25.3.8 TPS&ADC

符号	参数说明	测试条件	最小	典型	最大	单位
VCC	工作电压		2		5.5	V
Iadc	工作电流			200		uA
Res	分辨率				16	Bit
ENOB	有效位			12		Bit
Vrefadc	参考电压 (内置)	常温	-2%	1.213V	+2%	
VREF TC	参考电压温度系数			129ppm		/°C
Fsample	工作频率			32.768k		Hz
Cin	输入电容			0.141		pF
Rin	输入电阻(等效)			10M		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		0.0255		mV/LSB
	计算公式	实际拟合	VADCIN <sub>x</sub> = 0.02461*ADCxDAT+ 0.11918; 其中: VADCIN <sub>x</sub> 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		1	°C

Restps	分辨率	理论		0.002828		°C/LSB
	计算公式		$Tr = 12.9852 - TMPDAT * 0.002828$ 其中: $Tr$ 为实际的温度 (°C)			°C
VCC 通道						
VCCin	VCC 输入范围		2.1		5.5	V
Restps	分辨率	理论		0.1785		mV/LSB
Rinvdd	VCC 输入电阻 (内置)			43.75k		Ω
Div	分压系数 (内置)			1/6		
	计算公式	实际拟合	$VCC = 0.1785 * VCCDAT - 30.9252$ ; 其中: $VCC$ 为实际 VCC 电压 (mV)			mV
VTP 通道 (Buffer)						
VCCin	VTP 输入范围		0.3		2.6	V
Restps	分辨率	理论		0.09843		mV/LSB
	计算公式	实际拟合	$VTPx = 0.09843 * VTPxDAT + 2.55787$ 其中: $VTPx$ 为实际 $VTPx$ 引脚采样电压 (mV)			mV
Rin	输入电阻 (等效)			∞		Ω
VTP 通道 (TPGA)						
VCCin	VTP 输入范围		0		0.3	V
Restps	分辨率	理论		0.02461		mV/LSB
	增益选择		2	4	16	
	计算公式	实际拟合	$VTPx = 0.02461 * VTPxDAT + 1.94214$ 其中: $VTPx$ 为实际 $VTPx$ 引脚采样电压 (mV)			mV
Rin	输入电阻 (等效)			∞		Ω

注: 使用 VCC 通道, 工作电流为 ADC 工作电流和输入电阻对地电流之和

### 25.3.9 RTC

符号	参数说明	测试条件	最小	典型	最大	单位
	RTC 工作电压		2.0		5.5	V
	Vrtc 功耗			1.3		uA
	内部高频补偿精度	PLL 时钟开启		0.06		ppm
	低频补偿精度			30		ppm
	单点校正精度	误差系数为平均系数, 多功能烧写器进行单点常温校正		6	10	ppm

		-45~85°C			
	TOUT 输出频率			参见 下表	

附：TOUT 输出配置表

TOUT[3:0]				TOUT	TOUT (PLL 使能)
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	RTC 模块时钟	
0	0	1	1	未经高频补偿的 1Hz	高频补偿得到的 1Hz
0	1	0	0	未经高频补偿的 2Hz	高频补偿得到的 2Hz
0	1	0	1	未经高频补偿的 4Hz	高频补偿得到的 4Hz
0	1	1	0	未经高频补偿的 8Hz	高频补偿得到的 8Hz
0	1	1	1	未经高频补偿的 16Hz	高频补偿得到的 16Hz
1	0	0	0	未经高频补偿的 32Hz	高频补偿得到的 32Hz
1	0	0	1	未经高频补偿的 64Hz	高频补偿得到的 64Hz
1	0	1	0	未经高频补偿的 128Hz	高频补偿得到的 128Hz
其他				Reserved	

### 25.3.10 计量 ADC 参数

参数	最小值	典型值	最大值	单位	测试条件
<b>电能计量参数</b>					
有功电能计量误差		0.1%			常温 8000:1
无功电能计量误差		0.1%			常温 8000:1
电压有效值计量误差		0.1%			1000:1
		0.5%			8000:1
电流有效值计量误差		0.1%			1000:1
		0.5%			8000:1
频率测量误差		2		mHz	40Hz~75Hz
<b>ADC(Σ-Δ)参数</b>					
输入信号范围 (ADC FullScale)		±930		mVp	Ia/Ib/Ic/Ua/Ub/Uc
		±1080		mVp	In
输入阻抗		220		kΩ	EMUCLK=819.2KHz Ua/Ub/Uc
		115			EMUCLK=819.2KHz Ia/Ib/Ic
		75		kΩ	EMUCLK=819.2KHz In
		110		kΩ	EMUCLK=1.638MHz

					Ua/Ub/Uc
		57			EMUCLK=1.638MHz Ia/Ib/Ic/
		37		kΩ	EMUCLK=1.638MHz In
信噪比(SNR)		80		dB	@80%*VFS
带宽(-3dB)		6 12		kHz	EMU 时钟 819.2k EMU 时钟 1.638M
基准电压 Vref		1.178		V	常温
基准电压温度系数		8		ppm/°C	-40°C~85°C
<b>ADC(Σ-Δ)功耗 (常温)</b>					
三路 Ua Ub Uc 通道 ADC 开启		2.49		mA	
三路 Ia Ib Ic 通道 ADC 开启		2.4		mA	
In 通道 ADC 开启		1.62		mA	
7 路 ADC 数字控制逻辑总功耗		0.49		mA	
<b>工作电压</b>					
数字电源电压	3.0/4.5	3.3/5.0	3.6/5.5	V	
模拟电源电压	3.0/4.5	3.3/5.0	3.6/5.5	V	
CF 脉冲口驱动电流		5	8	mA	3.3V/5V
工作温度范围	-40		85	°C	

## 26 封装

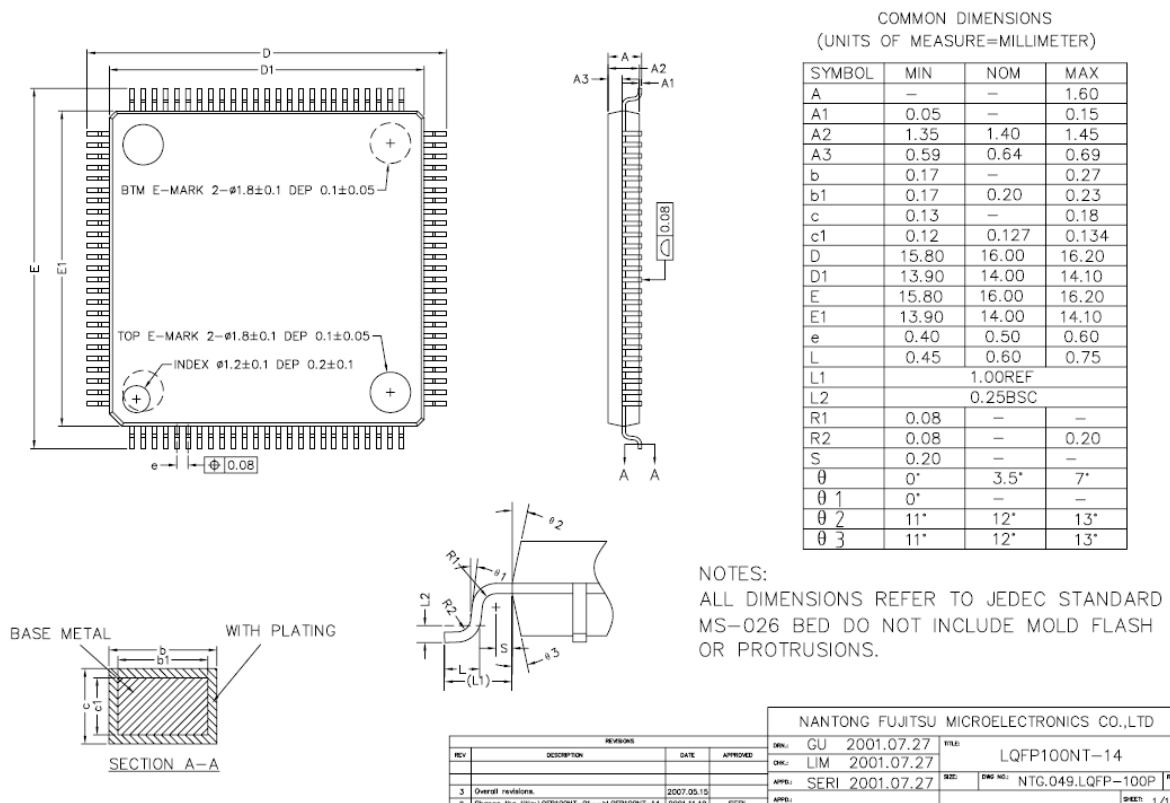
### 26.1 耐焊性

耐焊性：经过 260°C，不少于 5S 的回流焊，芯片能够保证正常工作。

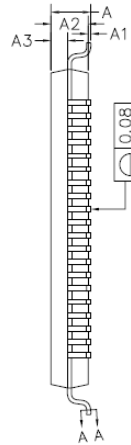
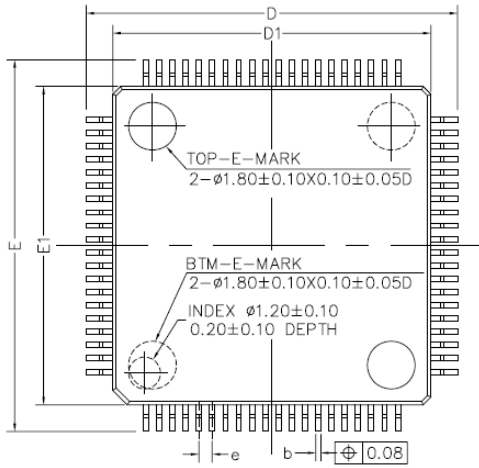
建议过锡炉次数：不超过 3 次。

### 26.2 封装图

HT7623 (100pin) : LQFP100

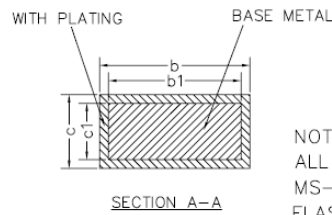
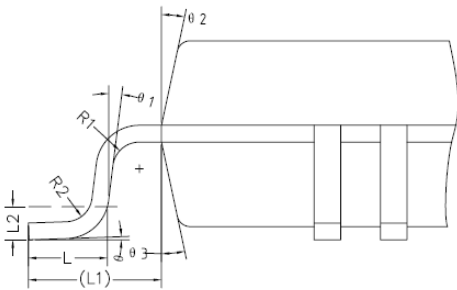


HT7625 HT7625S (80pin) : LQFP80



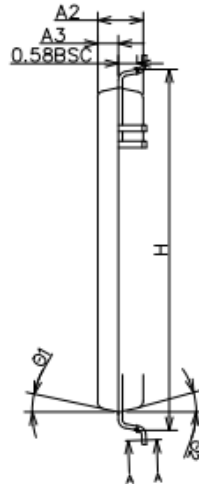
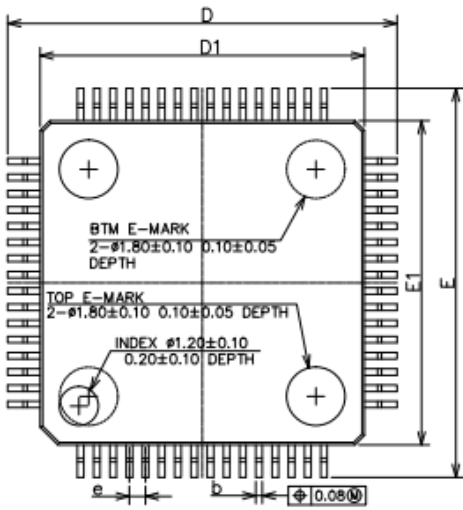
COMMON DIMENSIONS  
(UNITS OF MEASURE=MILLIMETER)

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	13.80	14.00	14.20
D1	11.90	12.00	12.10
E	13.80	14.00	14.20
E1	11.90	12.00	12.10
e	0.40	0.50	0.60
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
R1	0.08	—	—
R2	0.08	—	0.20
θ	0°	3.5°	7°
θ 1	0°	—	—
θ 2	11°	12°	13°
θ 3	11°	12°	13°



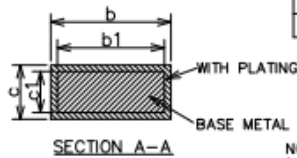
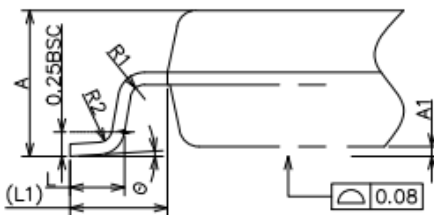
NOTES:  
ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026 BDD DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

## HT7627S (64pin): 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	0.117	0.127	0.137
D	11.95	12.00	12.05
D1	9.90	10.00	10.10
E	11.95	12.00	12.05
E1	9.90	10.00	10.10
e	0.40	0.50	0.60
H	11.09	11.13	11.17
L	0.53	—	0.70
L1	1.00REF		
R1	0.15REF		
R2	0.13REF		
θ	0°	3.5°	7°
θ1	11°	12°	13°
θ2	11°	12°	13°



NOTES:  
1. ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026 BCD DO NOT INCLUDE MOLD FLASH,GATE BURR OR PROTRUSION.

LEAD FORM PART