

# HT6X2X

# 用户手册

Tel: 021-51035886

Fax: 021-50277833

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

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

## 版本更新说明

版本号	修改时间	修改内容
V0.1	2014-12-25	初版
V1.0	2015-3-30	<ol style="list-style-type: none"> <li>1. ADC 管脚的输入范围和最大输入电压改为 800mV</li> <li>2. 芯片引脚结构说明图更改</li> </ol>
V1.1	2015-8-19	<ol style="list-style-type: none"> <li>1. 增加 AES、GHASH、RAND 模块概述</li> </ol>
V1.20	2015-8-29	<ol style="list-style-type: none"> <li>1, 电器规格: ADCBAT 输入最低 0</li> <li>2, POWIN 描述</li> </ol>
V1.30	2015-9-22	<ol style="list-style-type: none"> <li>1. 电源框图修改, 去掉 VCC 到 VRTC 之间的二极管</li> <li>2. 功耗参数修订: HRC、PLL 增加区分所有数字模块开启和关闭的功耗数据</li> </ol>
V1.31	2015-9-28	<ol style="list-style-type: none"> <li>1. RCCaliCON 增加 CLOCKIN 说明</li> </ol>
V1.32	2016-4-6	<ol style="list-style-type: none"> <li>1. 删除 MEMS 相关内容</li> </ol>
V1.33	2016-4-21	<ol style="list-style-type: none"> <li>1. 增加 5V tolerance IO 的说明</li> </ol>
V1.4	2016-05-16	<ol style="list-style-type: none"> <li>1. 增加 HT6027B 64PIN 封装双 SPI 同时更改引脚定义</li> <li>2. 增加 TPS 内部框图</li> <li>3. 更改 RTCRSTFLAG 定义的 POR 复位为 VRTC 电源域</li> </ol>
	2016-05-24	<ol style="list-style-type: none"> <li>1. LCD 章节添加 1/4bias 的 COM/SEG 波形说明, LCDCON_VRSEL[3: 0]]更新。</li> </ol>
V1.5	2016-08-09	<ol style="list-style-type: none"> <li>1. 修改 ADCIN、ADCBAT、VCC 计算公式</li> <li>2. ADCBAT pin 输入范围统一到 5.5V, 公式修改;</li> <li>3. INT \RX Sleep 唤醒状态描述修改;</li> <li>4. PIN 脚描述修改, 增加复用功能数字 IO 和模拟 IO 说明, 以及 OD 和上拉的应用说明;</li> <li>5. 寄存器默认值修正及说明;</li> <li>6. 7816 时序图及说明;</li> <li>7. 14.4 章节 HRC 频率描述笔误, 更改为 11MHz</li> <li>8. 时钟单元增加时钟源的起振时间描述</li> <li>9. Sleep 唤醒增加 UART 唤醒描述</li> <li>10. 外部数字滤波增加说明</li> <li>11. Toff 寄存器增加说明。</li> </ol>
V1.6	2016-11-07	<ol style="list-style-type: none"> <li>1 PMU 章节低功耗分时开启说明修改, VDET_TIME 时间修改</li> <li>2. VDD 更名为 VDD1P5, ADCBAT 更名为 ADCBAT</li> <li>3. 删除 Info Block 相关内容</li> <li>4. 修改电器参数 TTL 最低电平</li> <li>5. 去掉“管脚定义”里关于 RX0\RT1 “模拟滤波 2us”的描述, 在 UART 章节增加 RX0\RT1 最大波特率描述;</li> </ol>

V1.7	2016-11-14	<ol style="list-style-type: none"><li>1, LCDCON 增加 bit8 FCSET2, 修改充电模式分压电阻, 对比较度配置基准电压修改为 VCC 的百分比描述 ;</li><li>2, TBS 章节增加 ADCINX 输入阻抗描述、增加保证环境温度与外接低频 32K 时钟晶振的温度保持一致的描述;</li><li>3, 22.1DC 参数 IO 电流典型值修改为最小值, 增加最大值描述;</li><li>4, 23.1 增加过炉次数描述;</li><li>5, 21.1 增加说明 M0 不支持 MPU;</li><li>6, 18.1 增加说明: AES 加速器支持兼容 FIPS PUB 197, 详见应用笔记;</li><li>7, 18.3 增加 RAND 模块支持 FIPS-140-2, 以及随机数产生的工作方式描述。</li><li>8, 2.4 增加 Flash 安全性描述。</li><li>9, 2.1 增加 Flash 工作温度可保证系统在-40~+85 度正常工作描述。</li><li>10, 22.6 增加 ADC 分辨率测试条件描述。</li><li>11, 增加 15.3.4 传输模式, 包含 IIC 时序图。</li><li>12, 7.1 增加 I/O 防 ESD 电路描述。</li><li>13, 5.3 增加: 仿真口在上电/服务和启动之后的初始状态描述。</li><li>14, 22.6 增加 VCC 上升时间不影响复位的描述。</li><li>15, 22.3 增加各模块 3.3V 功耗数据。</li></ol>
------	------------	---

## 目 录

<b>1</b>	<b>HT6023/6025/6027 概述</b>	<b>10</b>
1.1	简介	10
1.2	框图	10
1.3	引脚排列	12
1.4	引脚定义	15
1.5	缩略语	21
<b>2</b>	<b>存储器模块</b>	<b>22</b>
2.1	概述	22
2.2	存储器映射图	22
2.3	FLASH 操作	24
2.3.1	Flash 的读保护	24
2.3.2	Code Flash 的操作说明	24
2.4	FLASH 控制功能	25
2.5	写保护寄存器说明	26
2.6	特殊功能寄存器列表	27
2.7	特殊功能寄存器说明	27
<b>3</b>	<b>时钟单元</b>	<b>30</b>
3.1	时钟分类	30
3.2	时钟框图	30
3.3	时钟停振检测框图	32
3.4	时钟说明	32
3.4.1	内部低频 RC 时钟 (Flrc)	32
3.4.2	内部高频 RC 时钟 (Fhrc)	32
3.4.3	外部低频晶振时钟 (Fosc)	32
3.4.4	内部 PLL 时钟 (Fpll)	33
3.4.5	时钟安全机制	33
3.4.6	时钟源的起振时间	33
3.4.7	时钟异常状态处理	34
3.5	特殊功能寄存器列表	35
3.6	特殊功能寄存器说明	35
<b>4</b>	<b>电源单元</b>	<b>45</b>
4.1	概述	45
4.2	框图	46
4.3	电源单元详细功能说明	47
4.3.1	电源切换	47
4.3.2	电源实时监测	47
4.3.3	内建 1.5V 电源	47

4.3.4	BOR 检测功能(BOR_DET).....	47
4.3.5	系统电源检测功能(VCC_DET) .....	50
4.3.6	低电压检测功能(LVDIN_DET).....	51
4.3.7	低功耗检测功能 (POW_DET) .....	51
4.3.8	VCC_DET, BOR_DET , LVDIN_DET 分时检测时序.....	51
4.3.9	电池钝化功能.....	52
4.4	特殊功能寄存器列表 .....	52
4.5	特殊功能寄存器说明 .....	53
<b>5</b>	<b>调试支持.....</b>	<b>60</b>
5.1	概况 .....	60
5.2	SW 引脚分布 .....	60
5.3	SW 口使用说明 .....	60
<b>6</b>	<b>工作模式.....</b>	<b>61</b>
6.1	工作模式 .....	61
6.2	睡眠模式 (SLEEP) .....	61
6.2.1	SLEEP 模式下各模块开关.....	61
6.2.2	SLEEP 模式下的唤醒.....	62
6.2.3	从 SLEEP 模式唤醒后的唤醒方式确认.....	62
6.2.4	进入 Sleep 模式.....	63
6.3	待机模式 (HOLD) .....	63
6.3.1	进入 Hold 模式.....	63
6.5	模式转换图.....	63
6.6	特殊功能寄存器列表 .....	64
6.7	特殊功能寄存器说明 .....	64
<b>7</b>	<b>GPIO 模块 .....</b>	<b>67</b>
7.1	概述 .....	67
7.2	芯片引脚结构说明 .....	68
7.3	I/O 端口基地址列表 .....	68
7.4	特殊功能寄存器说明 .....	69
<b>8</b>	<b>中断模块.....</b>	<b>74</b>
8.1	中断向量说明 .....	74
8.2	EXTI 中断说明 .....	75
8.3	特殊功能寄存器列表 .....	76
8.4	特殊功能寄存器说明 .....	76
<b>9</b>	<b>复位模块.....</b>	<b>80</b>
9.1	复位优先级 .....	80
9.2	复位说明 .....	80
9.2.1	上电复位.....	81
9.2.2	低电压检测复位.....	81

---

9.2.3	外部引脚复位.....	82
9.2.4	掉电复位.....	82
9.2.5	看门狗复位.....	83
9.2.6	软复位.....	83
9.2.7	调试复位.....	83
9.2.8	唤醒复位.....	84
9.3	特殊功能寄存器列表.....	84
9.4	特殊功能寄存器说明.....	85
<b>10</b>	<b>UART/7816 通讯模块.....</b>	<b>87</b>
10.1	功能说明.....	87
10.2	波特率计算.....	87
10.3	串口通讯模式说明.....	88
10.3.1	方式1.....	88
10.3.2	方式2.....	89
10.3.3	方式3.....	89
10.3.4	方式4.....	90
10.4	7816 接收和发送.....	91
10.4.1	7816 数据发送.....	91
10.4.2	7816 数据接收.....	91
10.4.3	7816 通讯示意图.....	92
10.5	特殊功能寄存器列表.....	95
10.6	特殊功能寄存器说明.....	95
<b>11</b>	<b>LCD 模块.....</b>	<b>103</b>
11.1	概述.....	103
11.2	LCD 与 GPIO 口复用表.....	103
11.3	LCD 框图.....	103
11.4	输出波形.....	104
11.5	LCD 显示操作.....	108
11.6	特殊功能寄存器列表.....	108
11.7	特殊功能寄存器说明.....	108
<b>12</b>	<b>WDT 模块.....</b>	<b>112</b>
12.1	概述.....	112
12.2	工作模式.....	112
12.3	特殊功能寄存器列表.....	112
12.4	特殊功能寄存器说明.....	113
<b>13</b>	<b>定时器模块.....</b>	<b>114</b>
13.1	定时器单元概述.....	114
13.2	周期性定时功能.....	114
13.3	PWM 功能.....	115
13.4	捕获功能.....	117

13.4.1	输入捕获模式	117
13.5	事件计数功能	117
13.6	中断功能	118
13.6.1	定时周期中断	118
13.6.2	捕获中断	118
13.6.3	比较中断	118
13.6.4	事件计数中断	118
13.7	特殊功能寄存器列表	119
13.8	特殊功能寄存器说明	119
<b>14</b>	<b>SPI 模块</b>	<b>124</b>
14.1	概述	124
14.2	详细功能说明	124
14.2.1	SPI 主要特征	124
14.2.2	SPI 模块框图	125
14.2.3	SPI 接口传输格式	125
14.2.4	主机模式传输格式	125
14.2.5	从机模式传输格式	127
14.2.6	中断功能	128
14.3	特殊功能寄存器列表	128
14.4	特殊功能寄存器说明	128
<b>15</b>	<b>I2C 模块</b>	<b>132</b>
15.1	概述	132
15.2	框图	132
15.3	功能描述	133
15.3.1	操作模式	133
15.3.2	串行时钟生成	133
15.3.3	中断生成	133
15.3.4	传输模式	133
15.4	特殊功能寄存器列表	141
15.5	特殊功能寄存器说明	141
<b>16</b>	<b>RTC 模块</b>	<b>144</b>
16.1	概述	144
16.2	功能描述	144
16.3	时钟校正	144
16.4	RTC 补偿系数寄存器说明	144
16.5	时间和万年历	145
16.6	中断功能	145
16.7	RTC 指示寄存器读写流程	146
16.7.1	读取 RTC 指示寄存器流程	146
16.7.2	写入 RTC 指示寄存器流程	146
16.8	校时记录	147

---

16.9	第二套 RTC	147
16.10	特殊功能寄存器列表	147
16.11	特殊功能寄存器说明	148
<b>17</b>	<b>TBS 模块</b>	<b>165</b>
17.1	概述	165
17.2	功能描述	165
17.3	特殊功能寄存器列表	166
17.4	特殊功能寄存器说明	166
<b>18</b>	<b>AES/GHASH/RAND/CRC 模块</b>	<b>176</b>
18.1	AES 概述	176
18.2	GHASH 概述	176
18.3	RAND 随机数概述	176
18.4	CRC 概述	176
18.5	特殊功能寄存器列表	177
18.6	特殊功能寄存器说明	178
<b>19</b>	<b>DMA 功能</b>	<b>192</b>
19.1	概述	192
19.2	功能描述	192
19.3	DMA 通道请求列表	192
19.4	DMA 数据传输说明	193
19.5	特殊功能寄存器列表	195
19.6	特殊功能寄存器说明	195
<b>20</b>	<b>HRC/LRC 自动校正</b>	<b>200</b>
20.1	概述	200
20.2	HRC 调整说明	200
20.3	LRC 调整说明	200
20.4	特殊功能寄存器列表	200
20.5	特殊功能寄存器说明	201
<b>21</b>	<b>CORTEX-M0 内核简要说明</b>	<b>204</b>
21.1	概述	204
21.2	系统定时器 SYSTICK	204
21.3	中断优先级说明	204
21.4	CMSIS 函数说明	204
<b>22</b>	<b>电气规格</b>	<b>206</b>
22.1	DC 参数	206
22.2	极限参数	206
22.3	功耗参数	207
22.4	外部 ADC 参数	209



---

22.5	ADCBAT 测试参数 .....	209
22.6	VCC 测试参数 .....	209
<b>23</b>	<b>封装 .....</b>	<b>210</b>
23.1	耐焊性 .....	210
23.2	封装图 .....	210

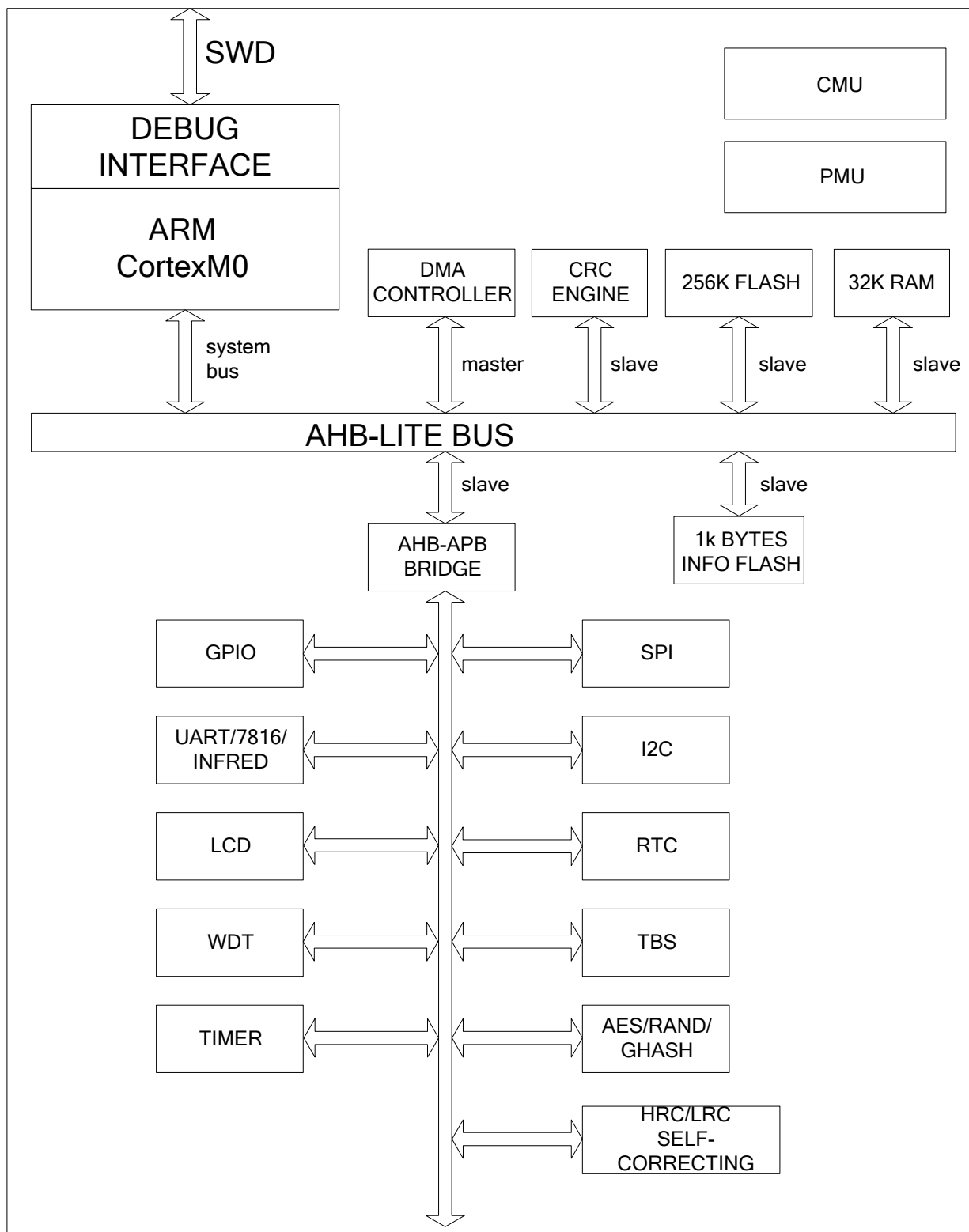
# 1 HT6023/6025/6027 概述

## 1.1 简介

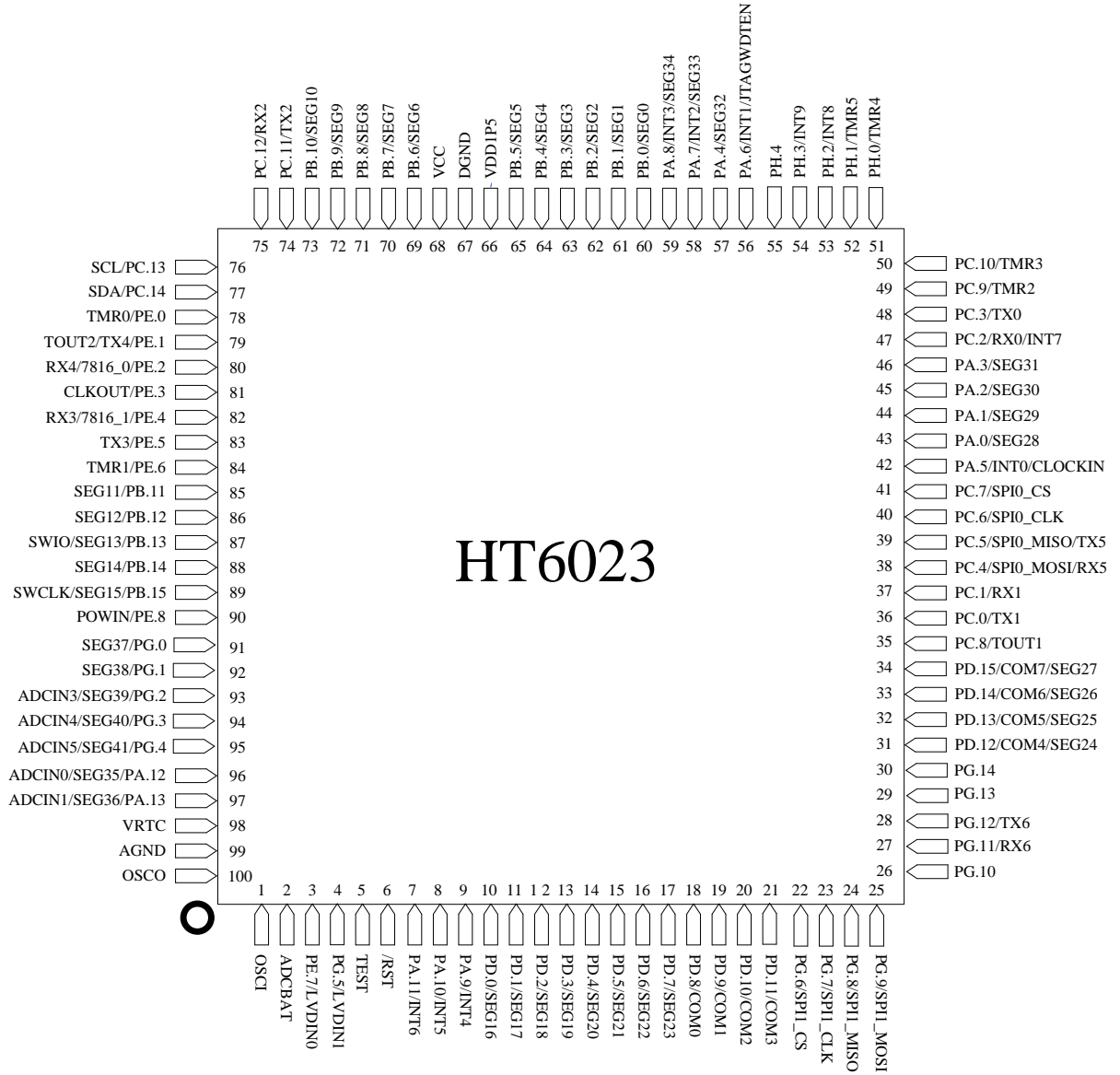
HT6X2X 系列是多功能、高性能、低功耗智能电表专用 256K 的 MCU 芯片，内部集成了 Cortex-M0 处理器、时钟管理、电源管理、硬件自动温度补偿 RTC、PLL、高频 RC、低频 RC、LCD 驱动等单元，以及 NVIC 和 DEBUG 调试功能。其中，支持每秒补偿机制的 RTC 单元，芯片以 32.768KHz 晶振时钟源作为 RTC 时钟源，通过芯片内部集成的时钟自动数字补偿单元，协助用户在无需其软件参与的情况下，实现 RTC 的自动补偿。

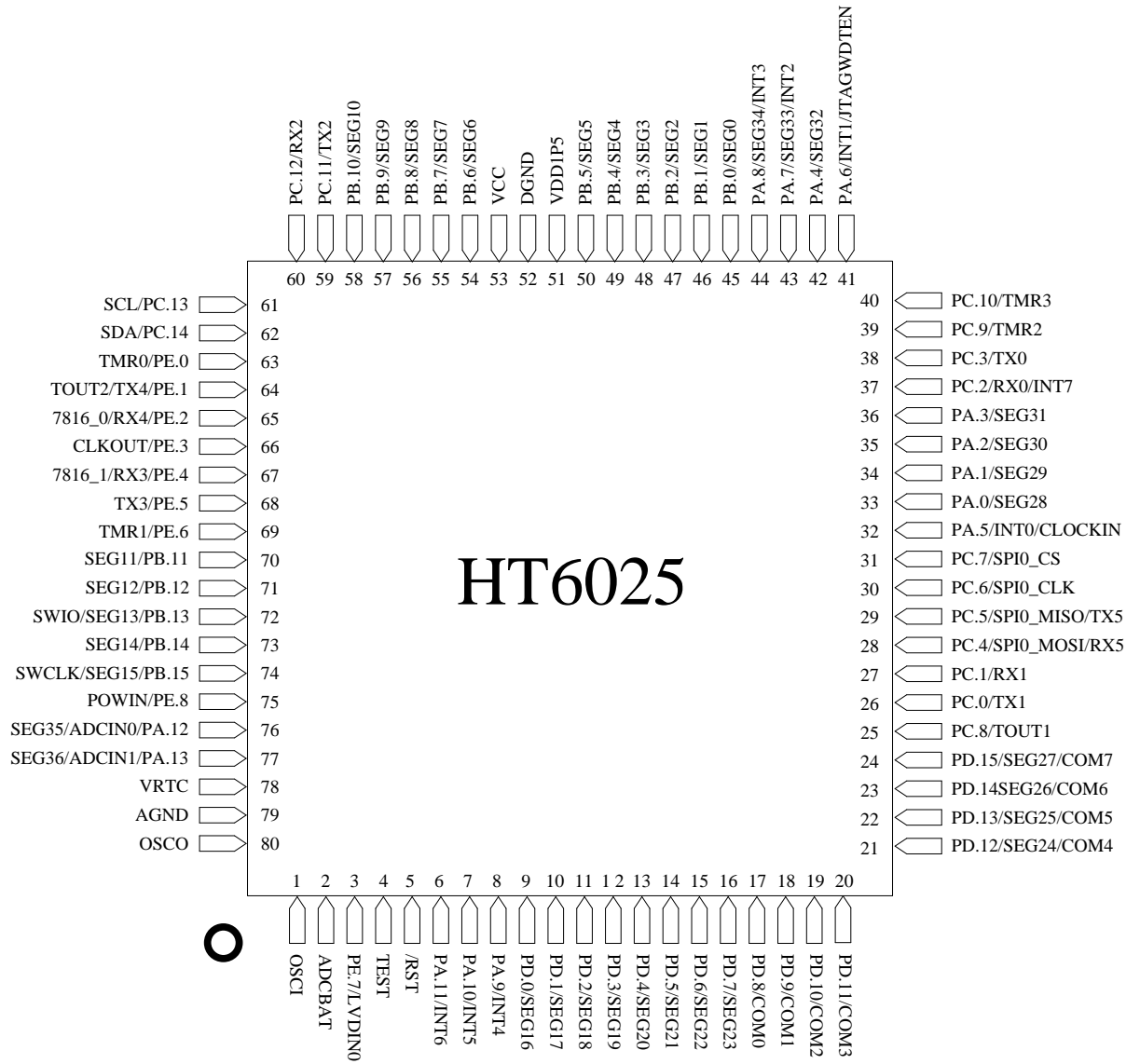
- 工作电压范围：2.0V~5.5V
- 工作温度范围：-40℃~85℃
- 采用 ARM Cortex-M0 CPU Core、256K Flash、32K SRAM
- 高速度：CPU 最高工作频率达到 44M（需使能指令预取功能）
- 低功耗：Hold 模式下最低功耗 4.3uA  
Sleep 模式下最低功耗 3.3uA
- RTC：支持每秒补偿机制
- RTC 补偿：RTC 内置曲线数字补偿，全温度范围 RTC 补偿无需用户软件参与
- LCD：支持 4COM，6COM，8COM 的 LCD 显示，SEG 接口最多支持 42 段（100 PIN）
- 供电方式：独立的 RTC 供电引脚，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：LQFP100：HT6023；LQFP80：HT6025；LQFP64：HT6027；

## 1.2 框图

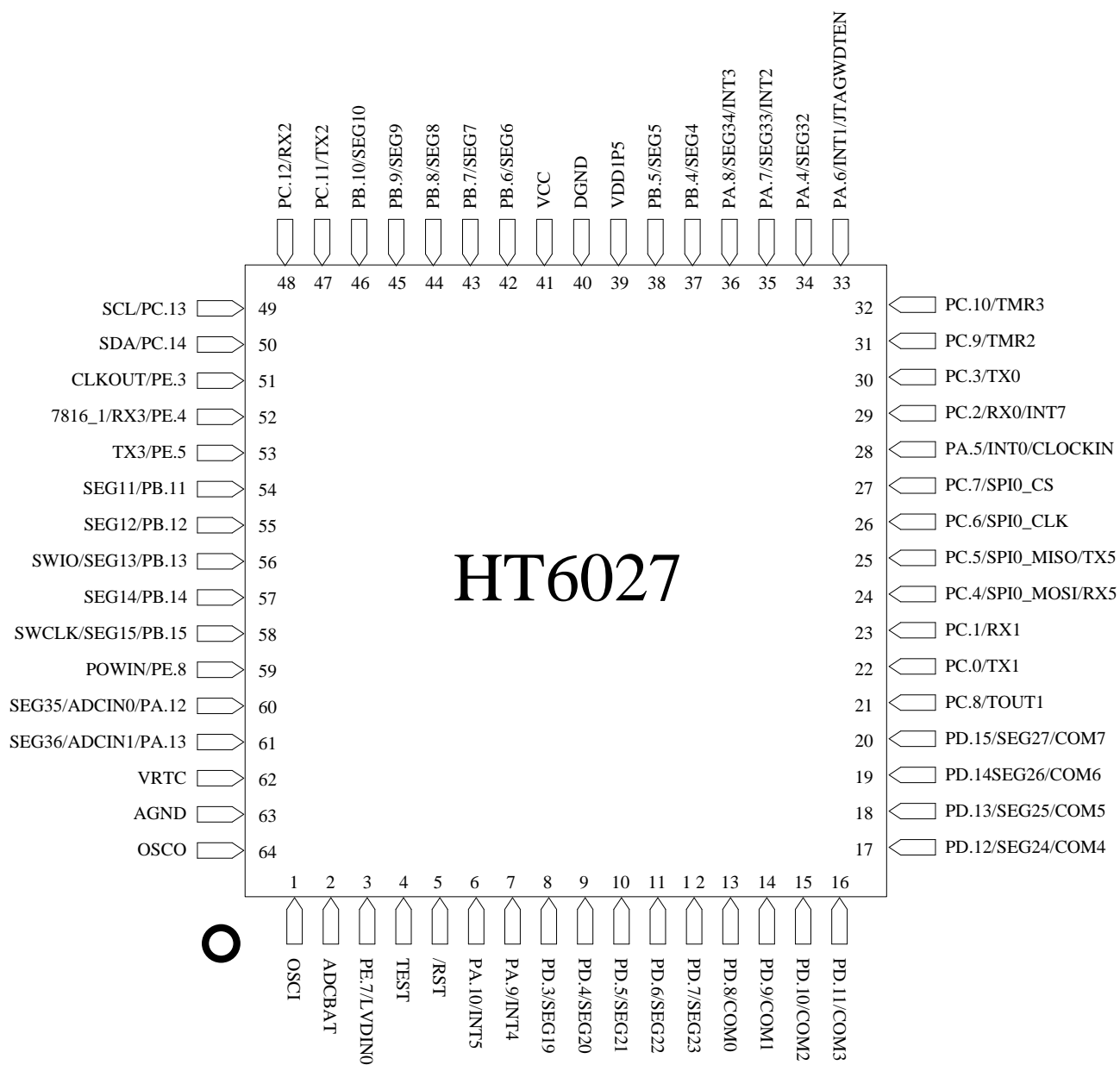


## 1.3 引脚排列

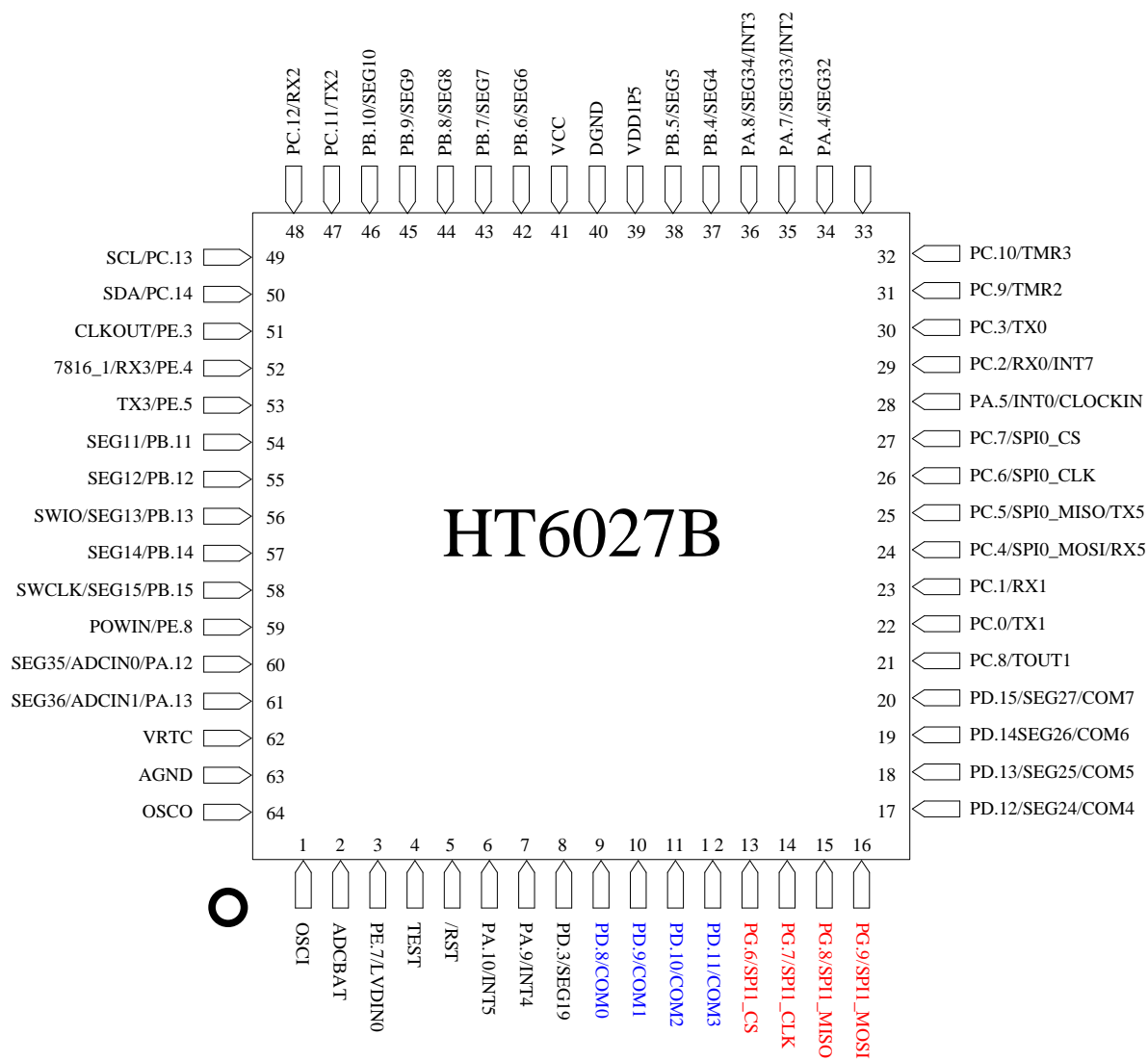




单 SPI 型号:



双 SPI 型号:



## 1.4 引脚定义

100 PIN	80 PIN	64 PIN	64 PIN (B)	标识	引脚类型	第一复用功能	第二复用功能	引脚说明
						IOCFG=1 AFCFG=0	IOCFG=1 AFCFG=1	
1	1	1	1	OSCI	I			低频晶振时钟输入
2	2	2	2	ADCBAT	I			电池电压采样输入
3	3	3	3	PE. 7	I/O	LVDINO		GPIO\外部电压检测 0, 比较电平 1.21V, 迟滞 150mV
4				PG. 5	I/O	LVDIN1		GPIO\外部电压检测 1, 比较电平 1.21V, 迟滞 150mV
5	4	4	4	TEST	I			测试引脚 (低电平有效, 内部上拉), 滤波 2us
6	5	5	5	/RST	I			复位信号 (低电平有效, 内部上拉), 滤波 2us
7	6			PA. 11	I/O	INT6		GPIO\外部中断 6, 滤波 2us
8	7	6	6	PA. 10	I/O	INT5		GPIO\外部中断 5, 滤波 2us
9	8	7	7	PA. 9	I/O	INT4		GPIO\外部中断 4, 滤波 2us
10	9			PD. 0	I/O	SEG16		GPIO\LCD 驱动 SEG 输出
11	10			PD. 1	I/O	SEG17		GPIO\LCD 驱动 SEG 输出
12	11			PD. 2	I/O	SEG18		GPIO\LCD 驱动 SEG 输出
13	12	8	8	PD. 3	I/O	SEG19		GPIO\LCD 驱动 SEG 输出
14	13	9		PD. 4	I/O	SEG20		GPIO\LCD 驱动 SEG 输出
15	14	10		PD. 5	I/O	SEG21		GPIO\LCD 驱动 SEG 输出
16	15	11		PD. 6	I/O	SEG22		GPIO\LCD 驱动 SEG 输出
17	16	12		PD. 7	I/O	SEG23		GPIO\LCD 驱动 SEG 输出
18	17	13	9	PD. 8	I/O	COM0		GPIO\LCD 驱动 COM 输出
19	18	14	10	PD. 9	I/O	COM1		GPIO\LCD 驱动 COM 输出
20	19	15	11	PD. 10	I/O	COM2		GPIO\LCD 驱动 COM 输出
21	20	16	12	PD. 11	I/O	COM3		GPIO\LCD 驱动 COM 输出
22			13	PG. 6	I/O	SPI1_CS		GPIO\SPI1_CS
23			14	PG. 7	I/O	SPI1_CLK		GPIO\SPI1_CLK
24			15	PG. 8	I/O	SPI1_MISO		GPIO\SPI1_MISO
25			16	PG. 9	I/O	SPI1_MOSI		GPIO\SPI1_MOSI



26				PG. 10	I/O			GPIO
27				PG. 11	I/O	RX6		GPIO\串口接收输入, 滤波 2us
28				PG. 12	I/O	TX6		GPIO\串口发送输出
29				PG. 13	I/O			GPIO
30				PG. 14	I/O			GPIO
31	21	17	17	PD. 12	I/O	COM4	SEG24	GPIO\LCD 驱动 COM\SEG 输出
32	22	18	18	PD. 13	I/O	COM5	SEG25	GPIO\LCD 驱动 COM 输出
33	23	19	19	PD. 14	I/O	COM6	SEG26	GPIO\LCD 驱动 COM 输出
34	24	20	20	PD. 15	I/O	COM7	SEG27	GPIO\LCD 驱动 COM 输出
35	25	21	21	PC. 8	I/O	TOUT1		GPIO\OSC 低频时钟或其分配时钟输出, TTL 电平输入
36	26	22	22	PC. 0	I/O	TX1		GPIO\串口发送输出, 可配置为大功耗驱动型 GPIO, Isink: 30mA Isource: 15mA
37	27	23	23	PC. 1	I/O	RX1		GPIO\串口接收输入
38	28	24	24	PC. 4	I/O	SPIO_MOSI	RX5	GPIO\SPIO_MOSI, 滤波 2us TTL 电平输入
39	29	25	25	PC. 5	I/O	SPIO_MISO	TX5	GPIO\SPIO_MISO, TTL 电平输入
40	30	26	26	PC. 6	I/O	SPIO_CLK		GPIO\SPIO_CLK, TTL 电平输入
41	31	27	27	PC. 7	I/O	SPIO_CS		GPIO\SPI_CS, TTL 电平输入
42	32	28	28	PA. 5	I/O	INT0	CLOCKIN	GPIO\外部中断输入 0\外部时钟输入, 滤波 2us
43	33			PA. 0	I/O	SEG28		GPIO\LCD 驱动 SEG 输出
44	34			PA. 1	I/O	SEG29		GPIO\LCD 驱动 SEG 输出
45	35			PA. 2	I/O	SEG30		GPIO\LCD 驱动 SEG 输出
46	36			PA. 3	I/O	SEG31		GPIO\LCD 驱动 SEG 输出
47	37	29	29	PC. 2	I/O	RX0	INT7	GPIO\串口接收输入
48	38	30	30	PC. 3	I/O	TX0		GPIO\串口发送输出,
49	39	31	31	PC. 9	I/O	TMR2		GPIO\PWM 输出或计数输入, TTL 电平输入
50	40	32	32	PC. 10	I/O	TMR3		GPIO\PWM 输出或计数输入, TTL 电平输入
51				PH. 0	I/O	TMR4		GPIO\PWM 输出, TTL 电平输入
52				PH. 1	I/O	TMR5		GPIO\PWM 输出, TTL 电平输入

53				PH. 2	I/O	INT8	INT8	GPIO\外部中断输入 8, TTL 电平输入, 滤波 2us
54				PH. 3	I/O	INT9	INT9	GPIO\外部中断输入 9, TTL 电平输入, 滤波 2us
55				PH. 4	I/O			GPIO, TTL 电平输入
56	41	33	33	PA. 6	I/O	INT1		<b>GPIO\外部中断输入 1, TEST=0 时, 该引脚功能为 JTAGWDTEN 功能, 此管脚可配 置为大功耗 GPIO, 滤波 2us Isink: 30mA Isource: 15mA</b>
57	42	34	34	PA. 4	I/O	SEG32		GPIO\LCD 驱动 SEG 输出
58	43	35	35	PA. 7	I/O	INT2	SEG33	GPIO\外部中断输入 2\LCD 驱 动 SEG 输出此管脚可配置为大 功耗 GPIO, 滤波 2us Isink: 30mA Isource: 15mA
59	44	36	36	PA. 8	I/O	INT3	SEG34	GPIO\外部中断输入 3\LCD 驱 动 SEG 输出此管脚可配置为大 功耗 GPIO, 滤波 2us Isink: 30mA Isource: 15mA
60	45			PB. 0	I/O	SEG0		GPIO\LCD 驱动 SEG 输出
61	46			PB. 1	I/O	SEG1		GPIO\LCD 驱动 SEG 输出
62	47			PB. 2	I/O	SEG2		GPIO\LCD 驱动 SEG 输出
63	48			PB. 3	I/O	SEG3		GPIO\LCD 驱动 SEG 输出
64	49	37	37	PB. 4	I/O	SEG4		GPIO\LCD 驱动 SEG 输出
65	50	38	38	PB. 5	I/O	SEG5		GPIO\LCD 驱动 SEG 输出
66	51	39	39	VDD1P5	P			内部 1.5V 输出, 需外接 0.1uF 滤波电容
67	52	40	40	DGND	G			芯片数字地
68	53	41	41	VCC	P			电源输入
69	54	42	42	PB. 6	I/O	SEG6		GPIO\LCD 驱动 SEG 输出
70	55	43	43	PB. 7	I/O	SEG7		GPIO\LCD 驱动 SEG 输出
71	56	44	44	PB. 8	I/O	SEG8		GPIO\LCD 驱动 SEG 输出
72	57	45	45	PB. 9	I/O	SEG9		GPIO\LCD 驱动 SEG 输出

73	58	46	46	PB. 10	I/O	SEG10		GPIO\LCD 驱动 SEG 输出
74	59	47	47	PC. 11	I/O	TX2		GPIO\串口发送输出 2
75	60	48	48	PC. 12	I/O	RX2		GPIO\串口接收输入 2, 滤波 2us
76	61	49	49	PC. 13	I/O	SCL		GPIO\IIC 驱动时钟信号, TTL 电平输入
77	62	50	50	PC. 14	I/O	SDA		GPIO\IIC 驱动数据信号, TTL 电平输入
78	63			PE. 0	I/O	TMR0		GPIO\PWM 输出或计数输入, TTL 电平输入
79	64			PE. 1	I/O	TX4	TOUT2	GPIO\串口发送输出 4\OSC 时 钟或其分频时钟输出,
80	65			PE. 2	I/O	RX4	7816_0	GPIO\串口接收输入 4\7816_0, 滤波 2us
81	66	51	51	PE. 3	I/O	CLKOUT		GPIO内部时钟源分频输出, ESAM/CARD_CLK, TTL电平输入
82	67	52	52	PE. 4	I/O	RX3	7816_1	GPIO\串口接收输入3\7816_1, 滤波2us
83	68	53	53	PE. 5	I/O	TX3		GPIO\串口发送输出 3,
84	69			PE. 6	I/O	TMR1		GPIO\PWM 输出或计数输入, TTL 电平输入
85	70	54	54	PB. 11	I/O	SEG11		GPIO\LCD 驱动 SEG 输出
86	71	55	55	PB. 12	I/O	SEG12		GPIO\LCD 驱动 SEG 输出
87	72	56	56	PB. 13	I/O	SEG13		GPIO\LCD 驱动 SEG 输出, 当系统进入调试模式, 此 PIN 作为 SW 通讯功能口: SWIO
88	73	57	57	PB. 14	I/O	SEG14		GPIO\LCD 驱动 SEG 输出
89	74	58	58	PB. 15	I/O	SEG15		GPIO\LCD 驱动 SEG 输出, 当系 统进入调试模式, 此 PIN 作为 SW 通讯功能口: SWCLK
90	75	59	59	PE. 8	I/O	POWIN		GPIO\低功耗检测输入, 若需使 用 POWIN 此功能, 硬件推荐外 接串联电阻 50 欧到 VCC
91				PG. 0	I/O	SEG37		GPIO\LCD 驱动 SEG 输出
92				PG. 1	I/O	SEG38		GPIO\LCD 驱动 SEG 输出
93				PG. 2	I/O	SEG39	ADCIN3	GPIO\LCD 驱动 SEG 输出\ADC 信 号输入
93				PG. 3	I/O	SEG40	ADCIN4	GPIO\LCD 驱动 SEG 输出\ADC 信 号输入
95				PG. 4	I/O	SEG41	ADCIN5	GPIO\LCD 驱动 SEG 输出\ADC 信

								号输入
96	76	60	60	PA. 12	I/O	SEG35	ADCIN0	GPIO\LCD 驱动 SEG 输出\ADC 信号输入
97	77	61	61	PA. 13	I/O	SEG36	ADCIN1	GPIO\LCD 驱动 SEG 输出\ADC 信号输入
98	78	62	62	VRTC	P			RTC 时钟电源输入
99	79	63	63	AGND	G			芯片模拟地
100	80	64	64	OSCO	0			外部低频时钟驱动输出

注：1. I=输入；O=输出；P=电源；G=地。

2. 芯片引脚选择 GPIO 功能：

若方向寄存器配置为输出，开漏 OD 功能配置控制有效，上拉控制无效；

若方向寄存器配置为输入，开漏 OD 功能控制无效，上拉控制有效；

3. 芯片引脚选择复用功能：

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

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

若配置为模拟输入 OD 控制和上拉控制都无效；

4. 复用功能为模拟功能：LVDINx、LCD\_SEG\COM、POWIN、ADCINx、ADCBAT,除此之外的其它复用功能为数字功能。

5. 端口数据寄存器 PTDAT 说明及数据读取

1) 芯片引脚选择 GPIO 功能或复用数字功能

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

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

2) 芯片引脚选择复用模拟功能

PTDAT 相应 bit 位值，固定为 0

6. PA. 6 引脚为一个特殊引脚，当 TEST=0 时，该引脚为输入 JTAGWDTEN 功能。

7. 普通管脚驱动能力为：Isink: 9mA Isource: 5mA；PC0、PA6-8 可配置成大电流驱动模式。

8. 具有 5V tolerance IO 的管脚共有 8 个，分别如下

100 PIN	80 PIN	64 PIN	标识	引脚类型	第一复用功能	第二复用功能	引脚说明
					IOCFG=1 AFCFG=0	IOCFG=1 AFCFG=1	
38	28	24	PC.4	I/O	SPI0_MOSI	RX5	滤波 2us，输出驱动 5mA TTL 电平输入

39	29	25	PC.5	I/O	SPI0_MISO	TX5	TTL 电平输入
40	30	26	PC.6	I/O	SPI0_CLK	SPI0_CLK	TTL 电平输入
41	31	27	PC.7	I/O	SPI0_CS	SPI0_CS	TTL 电平输入
49	39	31	PC.9	I/O	TMR2	TMR2	TTL 电平输入
50	40	32	PC.10	I/O	TMR3	TMR3	TTL 电平输入
76	61	49	PC.13	I/O	SCL	SCL	TTL 电平输入
77	62	50	PC.14	I/O	SDA	SDA	TTL 电平输入

## 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
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

---

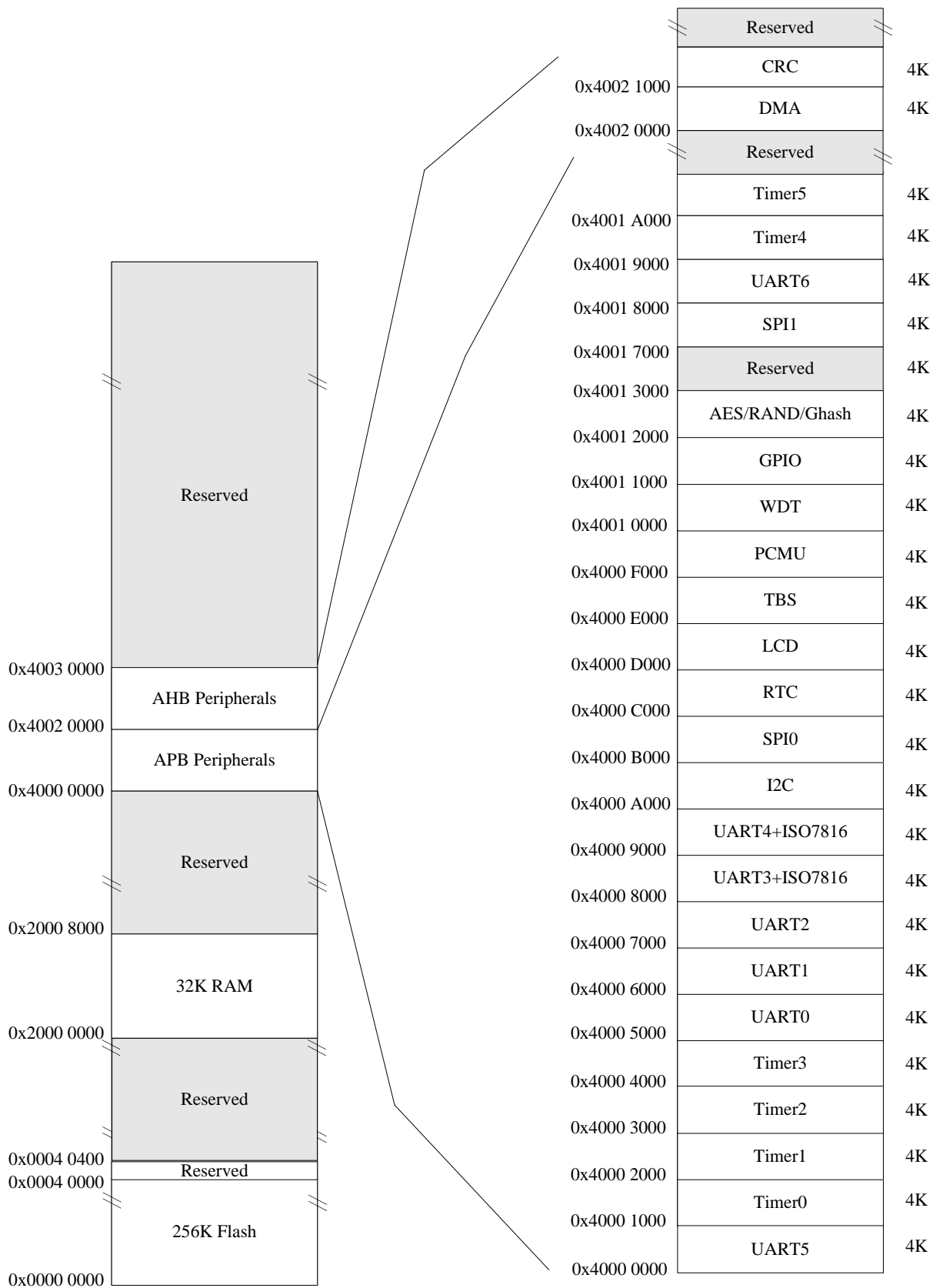
## 2 存储器模块

### 2.1 概述

HT6X2X 内置可编程高可靠 256K Flash 和 32K RAM。其中 Flash 具有读保护功能，可进行读、写、页擦除和全擦除操作，Flash 的特性如下：

- Flash 字节读取时间：40ns
- Flash 字节写时间：20us (max)
- Flash 页擦除时间：2ms (max)
- Flash 全擦除时间：10ms (max)
- Code Flash 页面大小：1K bytes/page
- 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℃）

### 2.2 存储器映射图



## 2.3 Flash 操作

### 2.3.1 Flash 的读保护

Flash 空间具有读保护功能，可以防止用户代码被读取。

将 Flash 的 00000FC1H 地址写入非 0FFH 的值后，开启读保护功能，256K Flash 空间的数据无法读出。（在线仿真时需要实时读取 Flash 内容，读保护下不能实现在线仿真）。

### 2.3.2 Code Flash 的操作说明

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 Flash 控制功能

为保证Flash的安全性，系统提供了可自加载的Flash控制字，可根据需求配置Flash Code加密。

H6X2X微控制器中Flash 存储器的0FC0H~0FC3H 区域为选项字节区域。当打开电源或从复位状态重启设备时，设备自动参考选项字节，并设置指定的功能。使用该产品时，必须使用选项字节设置以下几项功能。

- Sleep和Hold模式下看门狗开关
- Flash加密功能
- RTC补偿系数自动装载功能

CTRLBYFLASH Flash 控制功能说明			基地址: 0x4000C000 偏移地址: 100H					
Flash 地址	Bit7	6	5	4	3	2	1	Bit0
FC2H	RESERVE D	RESERVE D	RESERVE D	RESERVE D	WDT_EN [3]	WDT_EN [2]	WDT_EN [1]	WDT_EN [0]
Reset:	1	1	1	1	1	1	1	1
Flash 地址	Bit7	6	5	4	3	2	1	Bit0
FC1H	FLASH[7:0]							
Reset:	1	1	1	1	1	1	1	1

Flash 地址	Bit7	6	5	4	3	2	1	Bit0
FCOH	RESERVE D	x	RESERVE D	x	RTCST	AUTOREL OAD	RESERVE D	x
Reset:	1	0	1	0	1	1	1	0

位	功能描述
FC2H BIT[7:4]	内部控制位, 需保持 1111bit。
WDT_EN[3:0]	<b>看门狗使能</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 注: 需确保 FCOH 的 bit1 值为 1, 否则看门狗自动关闭。
FLASH[7:0]	<b>代码空间加密位</b> 如果 Flash[7:0]=0xFF, 则 Flash 不加密 其他: Flash 加密, 该寄存器位只能读, 不能写
FCOH BIT[7: 5]	内部控制位, 需保持 101bit
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>自动装载使能位</b> =1, 自动装载功能使能 =0, 自动装载功能屏蔽
FCOH BIT1	内部控制位, 需保持 1

注: 其他位不可以修改, 保持默认值。

## 2.5 写保护寄存器说明

被写保护的寄存器分布在 CMU、PMU 和 RTC 模块, 列表如下:

CMU模块寄存器基地址: 0x4000F000			
偏移地址	名称	复位值	功能描述
0x00	WPREG	0x0000	写保护控制寄存器
0x04	SYSCCLKCFG	0x0002	系统时钟配置寄存器 (写保护)
0x08	OSCADJ	0x0001	OSC 时钟电流偏置调整寄存器 (写保护)
0x0C	LRCADJ	0x0009	低频 RC 调整寄存器 (写保护)
0x10	HRCADJ	0x0040	高频 RC 调整寄存器 (写保护)
0x14	HRCDIV	0x0001	高频 RC 分频寄存器 (写保护)
0x1C	SYSCCLKDIV	0x0001	系统时钟分频寄存器 (写保护)
0x24	CLKOUTSEL	0x0002	CLKOUT 时钟选择寄存器 (写保护)
0x28	CLKOUTDIV	0x0000	CLKOUT 时钟分频寄存器 (写保护)
0x2C	CLKCTRL0	0x04E0	内部模块使能寄存器 0 (写保护)
0x30	CLKCTRL1	0x8000	内部模块使能寄存器 1 (写保护)

0x34	FLASHCON	0x0000	Flash 访问控制寄存器（写保护）
------	----------	--------	--------------------

**PMU 模块寄存器基地址：0x4000F400**

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

**RTC 模块寄存器基地址：0x4000C000**

偏移地址	名称	复位值	功能描述
0x18	SECR	0x0000	秒寄存器（写保护）
0x1C	MINR	0x0000	分寄存器（写保护）
0x20	HOURR	0x0000	时寄存器（写保护）
0x24	DAYR	0x0001	日寄存器（写保护）
0x28	MONTHR	0x0001	月寄存器（写保护）
0x2C	YEARR	0x0000	年寄存器（写保护）
0x30	WEEKR	0x0001	周寄存器（写保护）
0x200	SECR2	0x0000	第二套秒寄存器（写保护）
0x204	MINR2	0x0000	第二套分钟寄存器（写保护）
0x208	HOURR2	0x0000	第二套小时寄存器（写保护）
0x20C	DAYR2	0x0001	第二套天寄存器（写保护）
0x210	MONTHR2	0x0001	第二套月寄存器（写保护）
0x214	YRR2	0x0000	第二套年寄存器（写保护）
0x218	WEEKR2	0x0001	第二套星期寄存器（写保护）

## 2.6 特殊功能寄存器列表

**CMU模块寄存器基地址：0x4000F000**

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

## 2.7 特殊功能寄存器说明

WPREG (写保护寄存器)			基地址： 0x4000F000 偏移地址： 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	WPREG[15:8]							

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

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

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

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

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

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

## 3 时钟单元

### 3.1 时钟分类

测试温度范围：-40℃~85℃

名称	频率	精度	功耗		
			MIN	TYP	MAX
内部低频 RC 时钟 (Flrc)	32KHz 注 1	TBD		1uA	
内部高频 RC 时钟 (Fhrc)	11.01MHz 注 2	TBD		130uA	
外部低频 OSC 晶振 (Fosc)	32.768KHz			500nA	
内部 PLL (Fp11)	44.040192MHz/ 22.020096MHz			320uA	

注 1：内部低频 RC 时钟的误差范围在：13K~50K；

注 2：内部高频 RC 时钟在经过 HRCADJ 加载 Info-block 调整值之后，精度可达到 11.01MHz+-1%（常温）。

### 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 (11.01MHz)，系统复位后默认运行在 Fhrc。

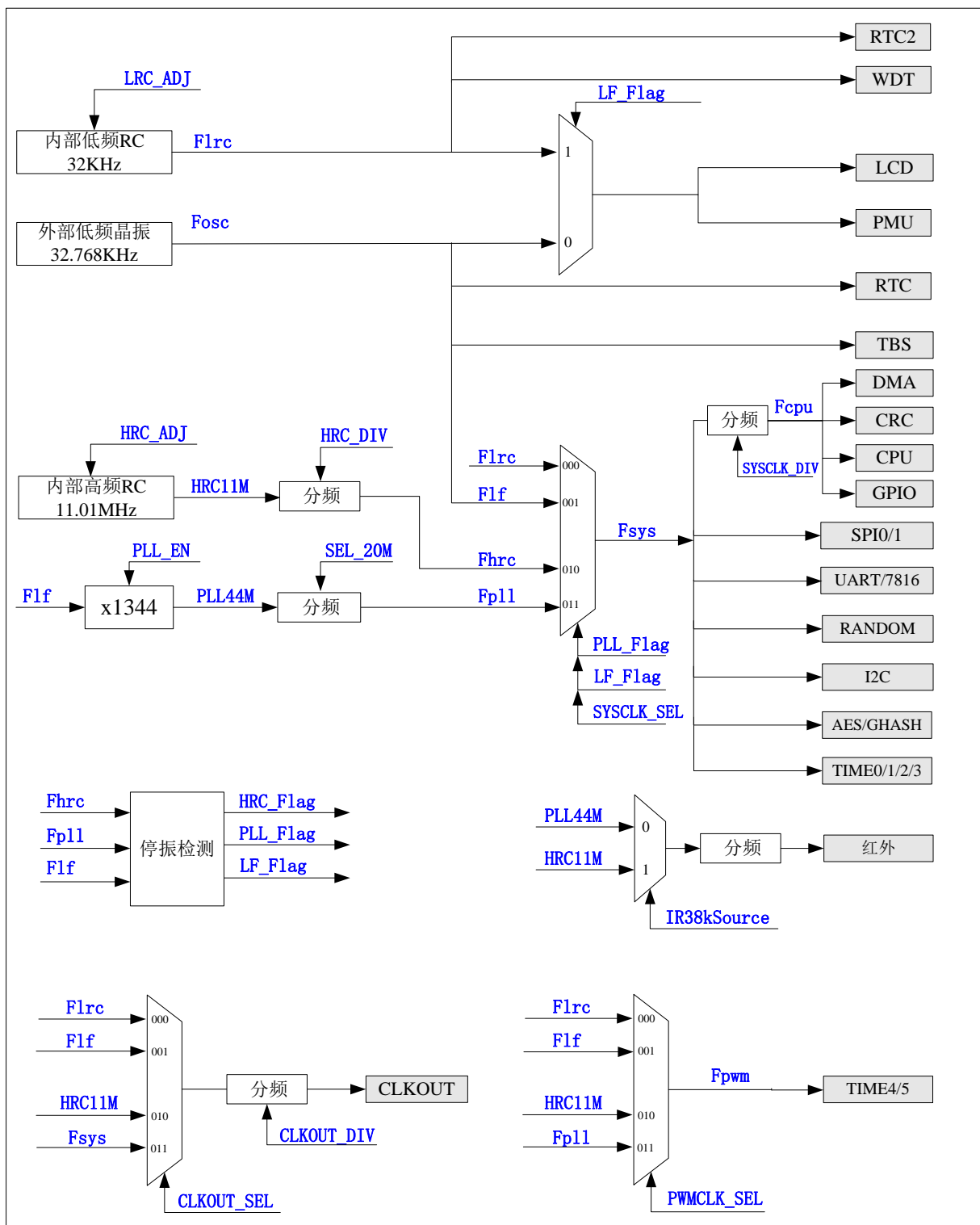
Fosc：外部低频 OSC 晶振时钟 (32.768KHz)。

Flf：内部选择的低频时钟 (32.768KHz)。

Fp11：内部 PLL 产生的高频时钟 (44.040192MHz) 或者 PLL 时钟 2 分频 (22.020096MHz)，来源为 Flf。

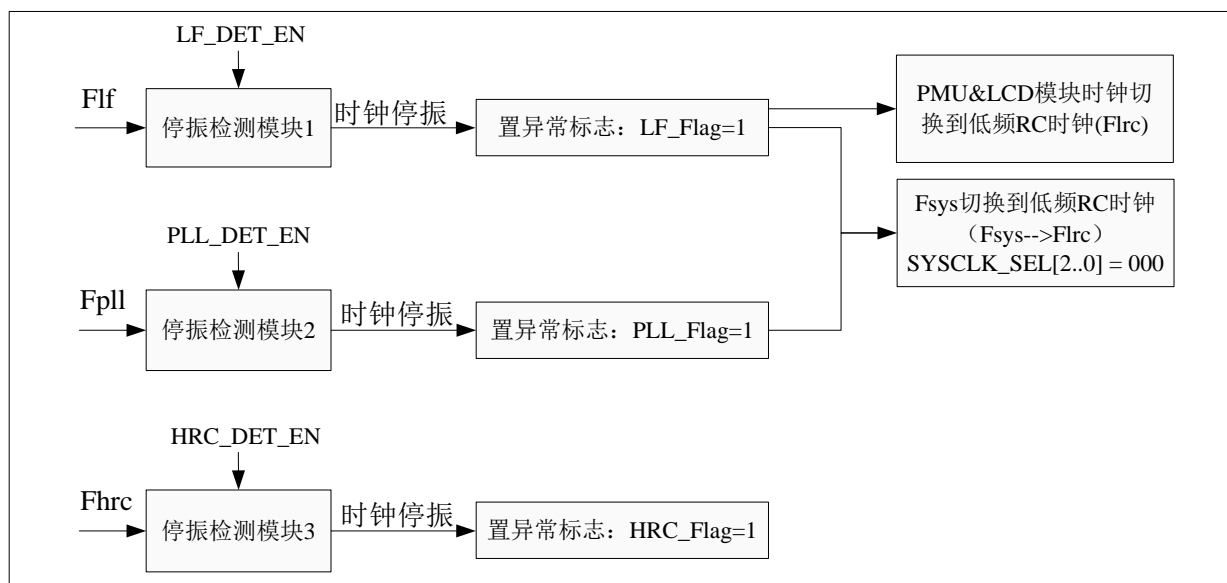
Fsys：系统时钟，主要给各个外设提供时钟

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 时钟停振检测框图



### 3.4 时钟说明

#### 3.4.1 内部低频 RC 时钟 (Flrc)

内部低频 RC 时钟振荡频率为 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 11.01MHz，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]）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)

HT6023/6025/6027 芯片外接低功耗晶体振荡器，时钟频率 F<sub>osc</sub>=32768Hz，以 F<sub>osc</sub> 作为系统的内部低频时钟 F<sub>lf</sub>，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 (Fp11)

内部 PLL 用于对内部低频时钟 F1f (32768Hz) 倍频 (倍频值=1344), 以对系统提供最高达 44.040192MHz 的高频时钟 Fp11。不过在默认情况下 (SEL\_20M=0), PLL 输出 2 分频后的时钟, 即 Fp11 为 22.020096 MHz。

如果用户选择 Fcpu 时钟为 44.040192MHz, 则必须先使能指令预取功能 (PreFetch\_EN=1)。PLL 稳定输出后会给出稳定标志位 PLL\_LOCK。

### 3.4.5 时钟安全机制

芯片内部集成有 3 个独立的时钟停振检测模块, 分别对内部低频时钟 F1f, PLL 输出时钟 Fp11, 和内部高频时钟 Fhrc 作检测。低频时钟 F1f 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 Flrc。

当对应的的时钟停振检测模块功能开启时, 内部低频时钟 F1f 发生停振, PLL 时钟 Fp11 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 F1f 停振, 系统给出时钟停振标志 LF\_FLAG, 如系统时钟 Fsys 选择 F1f 或 Fp11 (F1f 为 Fp11 的时钟源) 时, 系统会由硬件强制将系统时钟 Fsys 切换到内部低频 RC 时钟 Flrc, 且产生中断 (NMI 中断), 同时将寄存器 SYSCLK\_SEL[2:0] 的值置为 000。

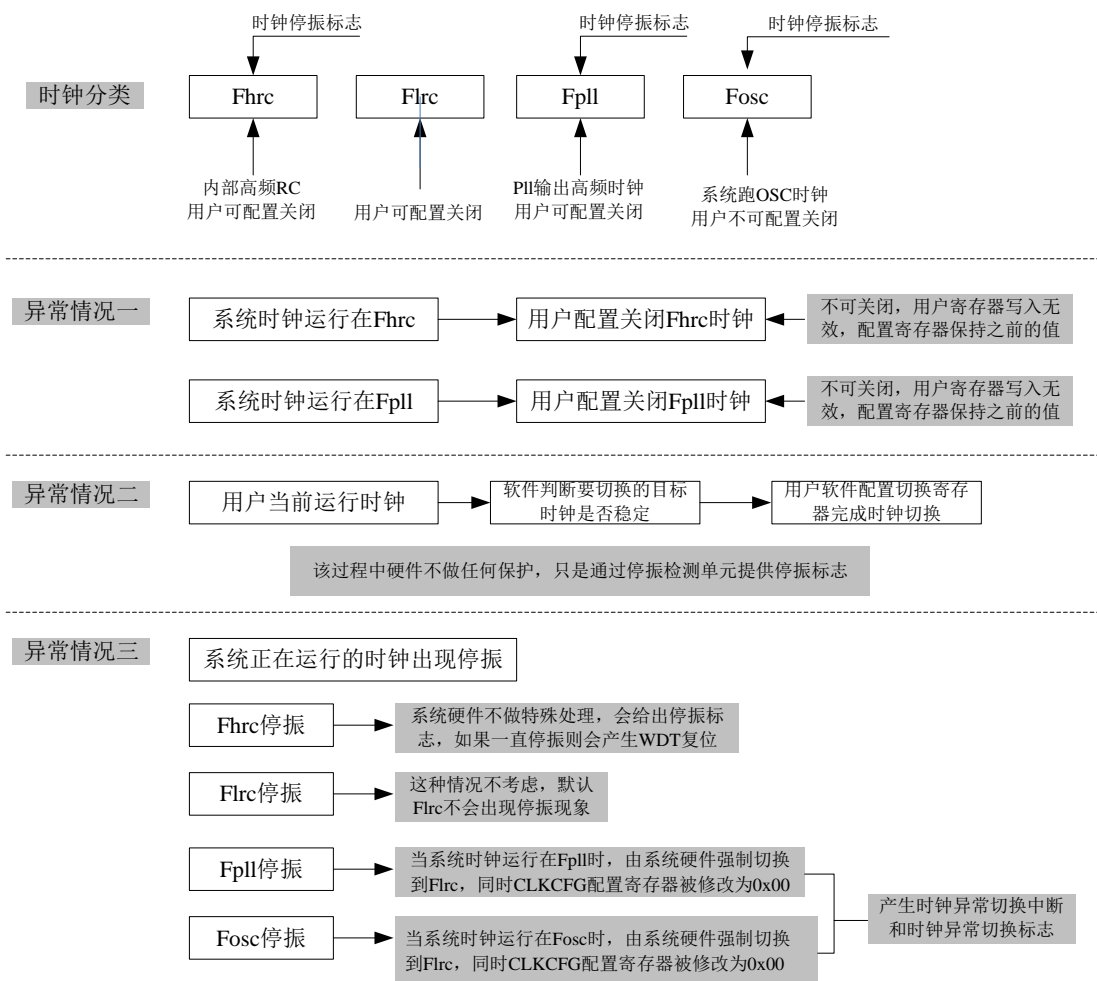
当停振检测模块检测到 Fp11 停振, 系统给出时钟停振标志 PLL\_FLAG, 如系统时钟 Fsys 选择 Fp11 时, 系统会由硬件强制将系统时钟 Fsys 切换到内部低频 RC 时钟 Flrc, 且产生中断 (NMI 中断), 同时将寄存器 SYSCLK\_SEL[2:0] 的值置为 000。

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。

### 3.4.6 时钟源的起振时间

时钟源	起振时间 or 稳定时间
LRC	25us
HRC	250us
LF (外部晶振 32.768KHz)	500ms
PLL	2ms

### 3.4.7 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。

### 3.5 特殊功能寄存器列表

CMU模块寄存器基地址：0x4000F000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	WPREG	R/W	0x0000	写保护控制寄存器
0x04	SYSLKCFG	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	0x00b5	LRC 时钟调整寄存器（写保护）
0x10	HRCADJ	R/W	0x0040	HRC 时钟调整寄存器（写保护）
0x14	HRCDIV	R/W	0x0001	HRC 时钟分频寄存器（写保护）
0x18	CLKSTA	R	0x0020	时钟状态寄存器（只读）
0x1C	SYSLKDIV	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	0x2CE0	内部模块使能寄存器 0（写保护）
0x30	CLKCTRL1	R/W	0x8000	内部模块使能寄存器 1（写保护）
0x3C	PREFETCH	R/W	0x0000	指令预取使能寄存器
0xF00	CHIPID	R/W	0x602x	芯片 ID 寄存器

### 3.6 特殊功能寄存器说明

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

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

SYSCLKCFG (写保护) (系统时钟配置寄存器)			基地址: 0x400F000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	WCLKEN	X	X	X	X	SYSCLK_SEL[2:0]		
Write:								
Reset:	0	0	0	0	0	0	1	0

位	功能描述																								
WCLKEN	<b>时钟配置寄存器写保护位</b> 如果用户要更改系统时钟选择, 必须同时将 WCLKEN 位置 1, 例如: ‘b1xxxxxxx’, 才可以对系统时钟选择位 SYSCLK_SEL[2:0] 进行写操作。																								
SYSCLK_SEL[2:0]	<b>系统时钟选择控制位:</b> <table border="1" style="margin: 10px auto;"> <thead> <tr> <th colspan="3">SYSCLK_SEL[2:0]</th> <th>系统时钟选择Fsys</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Flrc</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>F1f</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Fhrc (Default)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Fp11</td> </tr> <tr> <td>1</td> <td>X</td> <td>X</td> <td>保留</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>系统时钟可作为芯片内部硬件外设模块的时钟源, 经分频后也可作为 CPU 和 GPIO 的时钟源。</li> <li>复位后, 系统时钟默认为内部高频 RC 时钟 Fhrc (SYSCLK_SEL[2:0]=010)。</li> <li>当选择内部低频时钟 F1f 作为系统时钟时, 检测到低频时钟 F1f 发生停振, 此时系统会由硬件强制将 Fsys 切换到内部低速 RC 时钟 Flrc, 同时系统时钟控制位 SYSCLK_SEL[2:0] 被置为 000</li> <li>当选择 PLL 输出时钟 Fp11 作为系统时钟时, 检测到低频时钟 F1f) 或者 PLL 发生停振, 此时系统会由硬件强制将 Fsys 切换到内部低速 RC 时钟 Flrc, 同时系统时钟控制位 SYSCLK_SEL[2:0] 被置为 000。</li> <li>当选择 PLL 做系统时钟时, 必须先打开 PLL 时钟 (CLKCTRL0 的 bit4)</li> </ol>	SYSCLK_SEL[2:0]			系统时钟选择Fsys	0	0	0	Flrc	0	0	1	F1f	0	1	0	Fhrc (Default)	0	1	1	Fp11	1	X	X	保留
SYSCLK_SEL[2:0]			系统时钟选择Fsys																						
0	0	0	Flrc																						
0	0	1	F1f																						
0	1	0	Fhrc (Default)																						
0	1	1	Fp11																						
1	X	X	保留																						

JTAGSTA (芯片状态指示寄存器)			基地址: 0x400F000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	FLAG	X

Write:								
Reset:	0	0	0	0	0	0	0	1

位	功能描述
FLAG	该位用于指示芯片是否处于 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 (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:	LRC_ADJ[7:0]							
Write:	LRC_ADJ[7:0]							
Reset:	1	0	1	1	0	1	0	1

位	功能描述
LRC_ADJ[7:0]	LRC 输出频率调节控制位

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

位	功能描述
HRC_ADJ[5:0]	<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 11.01MHz。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 (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]	HRC 时钟分频设置:		
	HRC_DIV[1:0]		RC分频后的时钟Fhrc'
	0	0	Fhrc
	0	1	Fhrc/2 (Default)
	1	0	Fhrc/4
	1	1	Fhrc/8

CLKSTA (时钟状态寄存器)			基地址: 0x4000F000 偏移地址: 18H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	PLL_LOCK	PLL_FLAG	HRC_FLAG	X	X	LF_FLAG
Write:			X	X	X			X
Reset:	0	0	1	0	0	0	0	0

注: 此寄存器是只读状态寄存器

位	功能描述
PLL_LOCK	PLL 时钟锁定状态标志寄存器 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
PLL_FLAG	PLL 时钟 Fp11 停振标志 0: 正常。 1: 停振。
HRC_FLAG	内部高频 RC 时钟 Fhrc 停振标志

	0: 正常。 1: 停振。
LF_FLAG	内部低频时钟 F1f 停振标志 0: 正常 1: 停振

SYSCLKDIV (写保护) (系统时钟分频寄存器)			基地址: 0x4000F000 偏移地址: 1CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	SEL_20M	CPUCLK_DIV[2:0]		
Write:								
Reset:	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 $F_{p11}=22.020096\text{MHz}$ =1: PLL 输出时钟不分频, 即 $F_{p11}=44.040192\text{MHz}$																																				
CPUCLK_DIV[2:0]	<b>CPU 时钟分频设置:</b> <table border="1" data-bbox="502 1198 1077 1585"> <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 44M, 则必须提前使能指令预取功能 (PreFetch_EN=1)</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 (写保护) (CLKOUT 时钟选择寄存器)			基地址: 0x4000F000 偏移地址: 24H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0

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

位	功能描述																				
CLKOUT_SEL[2:0]	<p><b>CLKOUT 时钟输出引脚配置</b></p> <table border="1"> <thead> <tr> <th colspan="3">CLKOUT_SEL[2:0]</th> <th>CLKOUT时钟选择</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Flrc</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Flf</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Fhrc (Default)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Fsys</td> </tr> </tbody> </table> <p>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。</p>	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
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																		

CLKOUTDIV (写保护) (CLKOUT 时钟分频寄存器)		基地址: 0x4000F000 偏移地址: 28H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	CLKOUT_DIV[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0

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

CLKCTRL0 (写保护) (内部模块使能控制寄存器 0)		基地址: 0x4000F000 偏移地址: 2CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:			1P5LBOR	CLKOUT_	X	OSC_SLP	HRC_DET	PLL_DET
Write:	CRC_EN	ARG_EN	_EN	EN			_EN	_EN
Reset:	0	0	1	0	1	1	0	0



	Bit7	6	5	4	3	2	1	Bit0
<b>Read:</b>	LF_DET_	PLLLOCK	HRC_EN	PLL_EN	I2C_EN	SPI0_EN	LCD_EN	X
<b>Write:</b>	EN	_EN						
<b>Reset:</b>	1	1	1	0	0	0	0	0

位	功能描述
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
OSC_SLP	<b>OSC 低功耗使能控制位 (用户不要修改该寄存器的默认值)</b> 0: 大功耗 1: 小功耗 (default)
HRC_DET_EN	<b>HRC 时钟检测模块控制位</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 Fp11 时, 此时不能关闭 PLL_EN, 该寄存器位写入无效
I2C_EN	<b>I2C 模块时钟使能位</b> 0: 关闭 I2C 模块 (default) 1: 使能 I2C 模块
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_EN	<b>LCD 模块时钟使能位</b> 0: 关闭 LCD 模块 (default) 1: 使能 LCD 模块

<b>CLKCTRL1 (写保护)</b>		<b>基地址: 0x4000F000</b>						
<b>(内部模块使能控制寄存器 1)</b>		<b>偏移地址: 30H</b>						
	<b>Bit15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>Bit8</b>
<b>Read:</b>	LRC_RTC	SOFTWDT	SPI1_EN	UART6_E	TMR5_EN	TMR4_EN	UART5_E	UART4_7
<b>Write:</b>	2_EN	_EN		N			N	816_EN
<b>Reset:</b>	1	0	0	0	0	0	0	0
	<b>Bit7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Bit0</b>
<b>Read:</b>	UART3_7	UART2_E	UART1_E	UART0_E	TMR3_EN	TMR2_EN	TMR1_EN	TMRO_EN
<b>Write:</b>	816_EN	N	N	N				
<b>Reset:</b>	0	0	0	0	0	0	0	0

位	功能描述
LRC_RTC2_EN	<b>第二套 RTC 的工作使能位</b> 0: 关闭 1: 开启 (default)
SOFTWDT_EN	<b>调试模式下看门狗使能:</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	Timer4 时钟使能位 0: 关闭 1: 使能
UART5_EN	UART5 时钟使能位 0: 关闭 1: 使能
UART4_7816_EN	UART4 时钟使能位 0: 关闭 1: 使能
UART3_7816_EN	UART3 时钟使能位 0: 关闭; 1: 使能
UART2_EN	UART2 时钟使能位 0: 关闭 1: 使能
UART1_EN	UART1 时钟使能位 0: 关闭 1: 使能
UART0_EN	UART0 时钟使能位 0: 关闭 1: 使能
TMR3_EN	Timer3 时钟使能位 0: 关闭 1: 使能
TMR2_EN	Timer2 时钟使能位 0: 关闭 1: 使能
TMR1_EN	Timer1 时钟使能位 0: 关闭 1: 使能
TMRO_EN	Timer0 时钟使能位 0: 关闭 1: 使能

注: CRC/AES\_RAND\_GHASH/IIC/SPI0-1/UART0-6/TMR0-3 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 (写保护) (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	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	PreFetch_EN
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PreFetch_EN	指令预取使能控制位 =0: 禁止 =1: 使能 注: 若 CPU 时钟选择 44M 时钟, 则必须先使能此控制位。

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

位	功能描述
CHIPID[15...0]	0x602x

---

## 4 电源单元

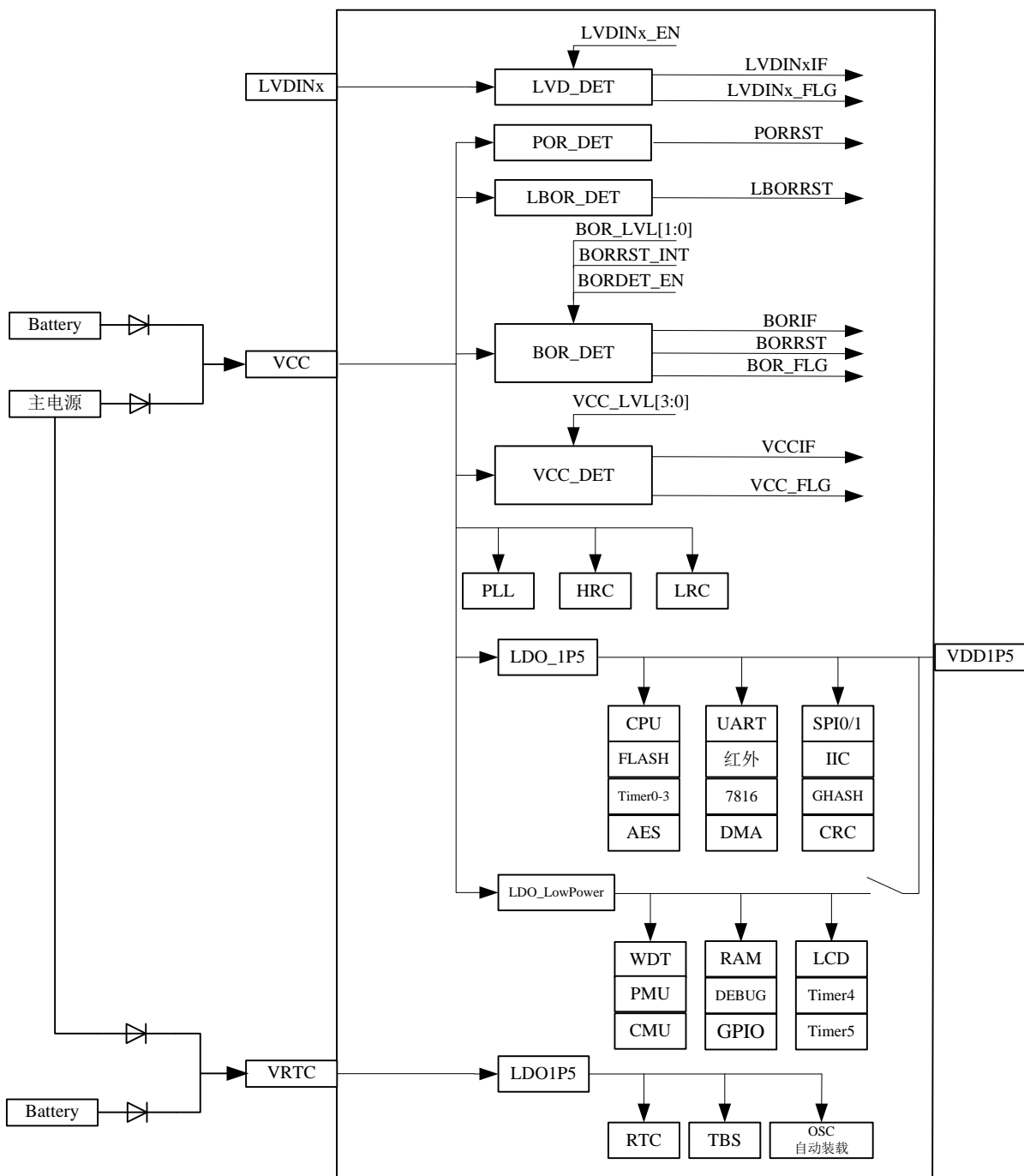
### 4.1 概述

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。
- 测量电池功耗, 给出报警标志。
- 测量电池电压 ADCBAT, 用于估算电池电量, 给出报警标志。
- 测量系统电源 VCC, 可用于调整 LCD 驱动的对比度等。
- 测量芯片引脚 ADCIN0, ADCIN1, ADCIN3, ADCIN4, ADCIN5 的电压。
- 为芯片内部数字模块提供 1.5V 电源

(注: VCC, ADCBAT, ADCIN0, ADCIN1, 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：LVDIN<sub>x</sub> 指 LVDINO 和 LVDIN1

## 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1.9V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

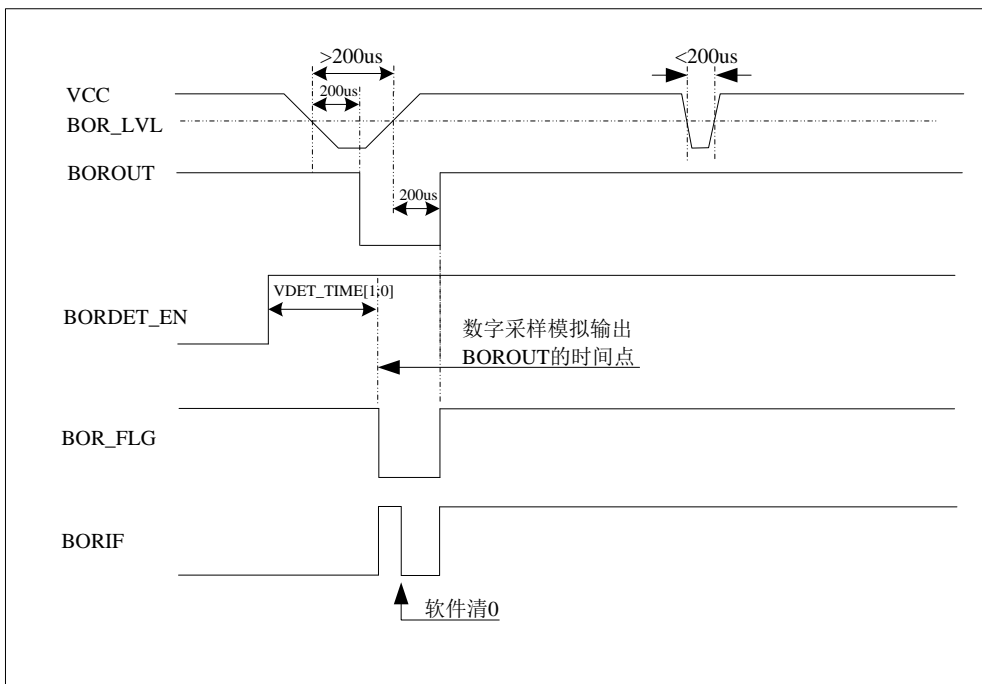
状态寄存器 RSTSTA 的 BOR 标志位被置为 1。当 BOR 检测模块检测到系统电源 VCC 电压高于设定电压  $V_{bor}$  时，BOR 检测模块内部信号 BOROUT 输出高电平，在该高电平持续的 1024 个 Flrc 周期后，内部复位信号 IRST 也变为高电平。

$V_{bor}$  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。

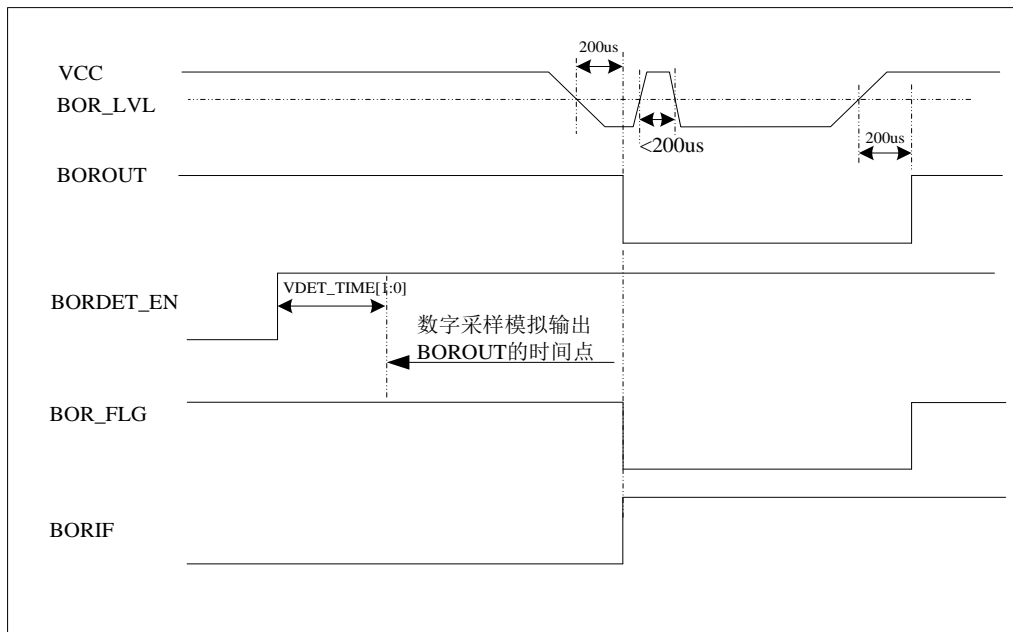
掉电复位 BOR 产生时，下面的事件将会发生：

- 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

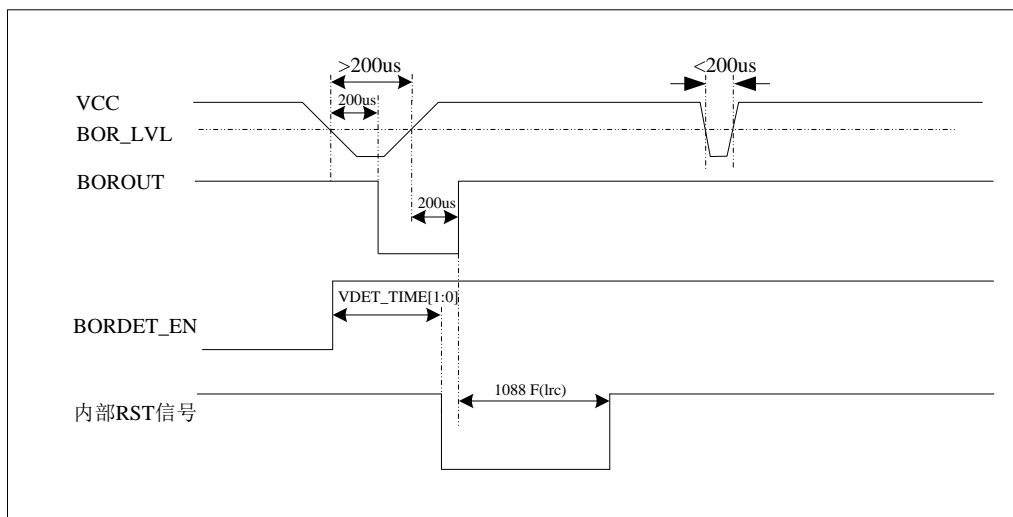
LBOR\_DET 与 BOR\_DET 的检测过程基本相同。





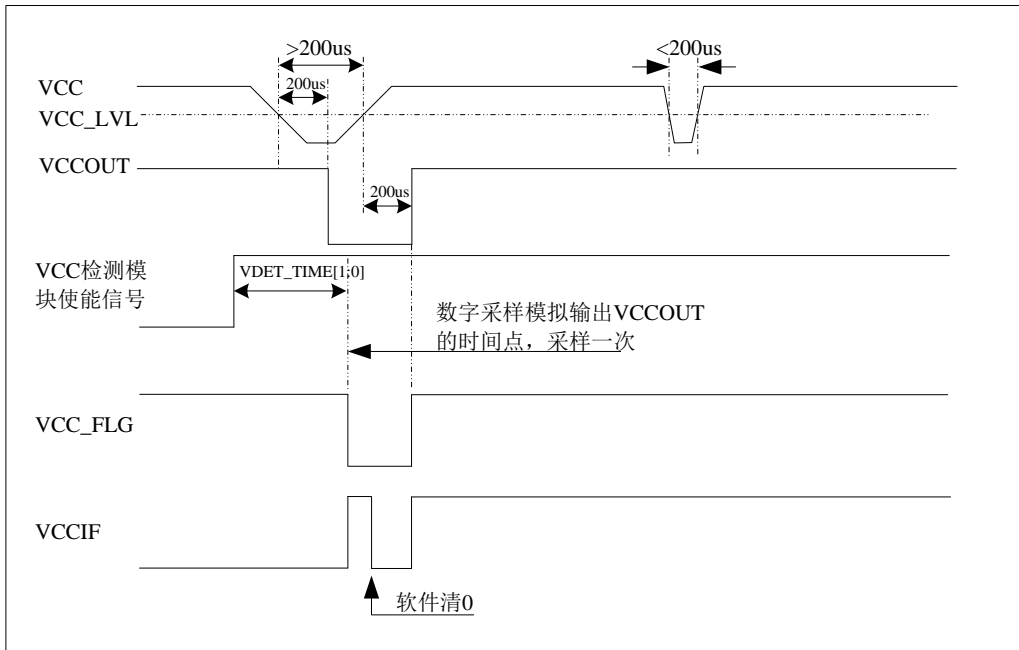


BOR 中断信号产生示意图

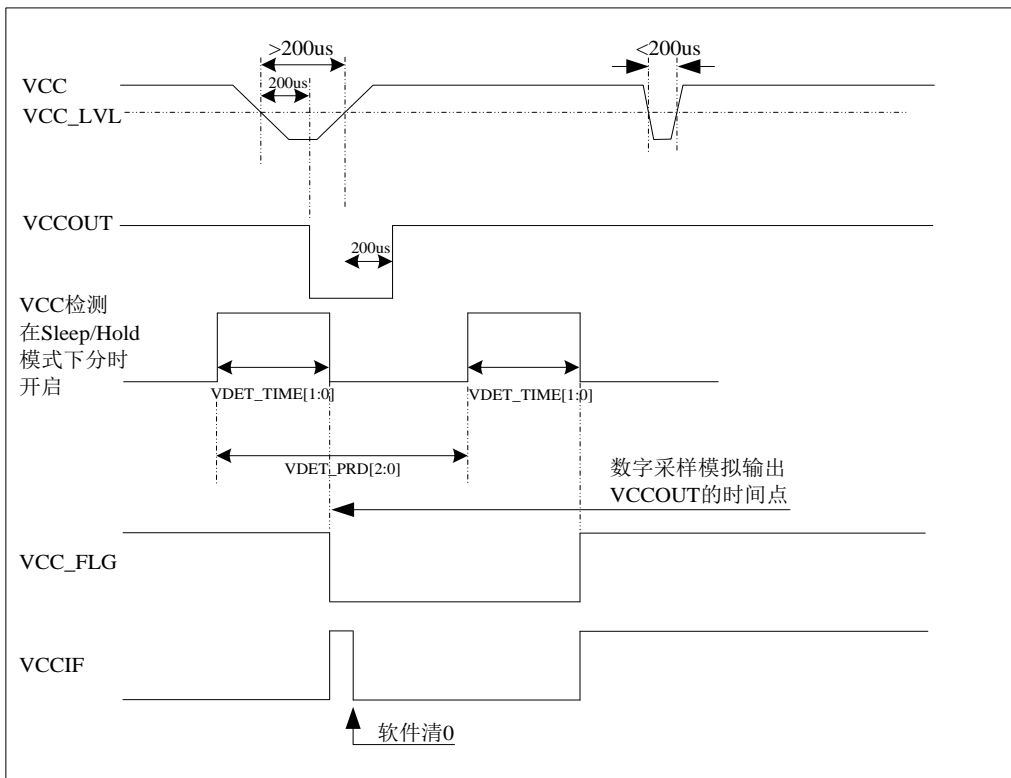


BOR 复位示意图

### 4.3.5 系统电源检测功能(VCC\_DET)

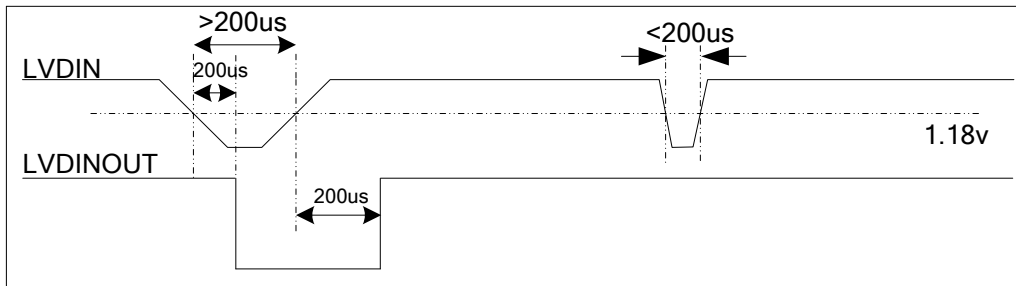


Normal模式Vcc检测



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 低功耗检测功能 (POW\_DET)

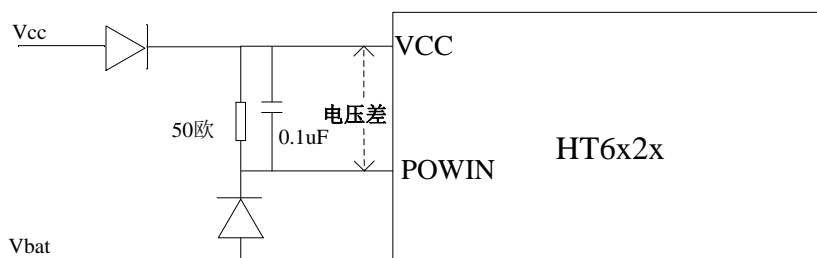
系统可以检测芯片在低功耗模式下运行过程中的功耗异常,当芯片 POWIN 引脚检测到的电压差(也就是下图中电阻上的电压差= $V_{PowIn}-V_{CC}$ ),高于用户设定的检测电压阈值(可由 POW\_LVL[1:0]设置),系统将产生相应的中断标志 POWIF,如果用户使能了中断 POWIE,则系统会产生相应的 POW 中断。

推荐用户串联的检测电阻为 50 欧姆,则对于不同检测阈值 1.25mV, 2.5mV, 3.75mV, 5.0mV, 6.25mV, 7.5mV, 8.75mV, 10mV, 相对应的实际检测电流为 25uA, 50uA, 75uA, 100uA, 125uA, 150uA, 175uA, 200uA。

由于检测模块本身的功耗为 100uA,所以在低功耗模式下采用分时开启的方式,1s 内开启 100us。功耗检测功能可由用户控制开启,控制位为寄存器 PMUCON 的 POWDET\_EN 位。

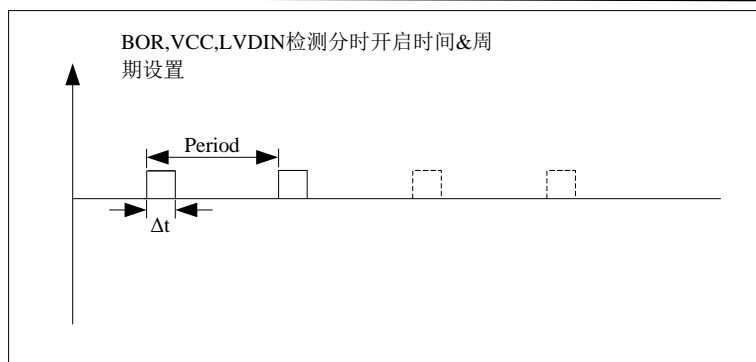
POWIN 引脚功能与 PE.8 复用,用户使用低功耗电流检测功能时需要将该 PE.8 引脚配置为 POWIN 复用功能。

在低功耗电流检测功能中用户还可以配置数字滤波功能来提高电流检测的准确度,通过配置寄存器 PDTFLT,用户可以得到更加准确的检测结果。



### 4.3.8 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。

### 4.3.9 电池钝化功能

当 ADCBAT Pin 连接外部电池时，如果置位 DISCHARGE\_EN 控制位，则使能了电池钝化功能，此时 ADCBAT Pin 上会流过相应电流，用户可以配置 DISCHARGE\_CTRL[1..0]来控制钝化电流大小。

## 4.4 特殊功能寄存器列表

PMU 模块寄存器基地址：0x4000F400				
偏移地址	名称	读写方式	复位值	功能描述
0x00	PMUCON	R/W	0x001f	PMU配置寄存器（写保护）
0x04	VDETCFG	R/W	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
0x1C	PDTFLT	R/W	0x07	低功耗电流检测数字滤波寄存器

## 4.5 特殊功能寄存器说明

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

位	功能描述
DISCHARGE_EN	<b>电池钝化功能使能:</b> 0: 电池钝化功能关闭 1: 电池钝化功能开启
POWDET_EN	<b>POW_DET 模块使能信号</b> 0: 关闭 POW_DET 模块 (default) 1: 开启 POW_DET 模块 <b>注: 该功能使能后, 将分时开启, 1 秒钟开启 100us</b>
Hold_LDO	<b>在 Hold 模式下选择打开/关闭 大功耗 LDO (默认打开)</b> 0: 关闭大功耗 LDO 1: 打开大功耗 LDO (default) <b>注: 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</b>
LVDIN1_EN	<b>LVDIN1_DET 模块使能信号, 监测 LVDIN1 输入引脚</b> 0: 关闭 LVDIN1_DET 模块 1: 开启 LVDIN1_DET 模块 (default)
LVDINO_EN	<b>LVDINO_DET 模块使能信号, 监测 LVDINO 输入引脚</b> 0: 关闭 LVDINO_DET 模块 1: 开启 LVDINO_DET 模块 (default)
BORRST	<b>BOR复位/中断选择位</b> 0: VCC电压低/高于设定阈值时产生BOR中断 1: VCC电压低于VDETCFG[1:0]设定阈值时产生BOR复位 (default)
BORDET_EN	<b>BOR_DET 模块使能信号</b> 0: 关闭 BOR_DET 模块 1: 开启 BOR_DET 模块 (default)

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

位	功能描述																																																
DISCHARGE_CTRL[1:0]	电池钝化功能, 电流大小选择 00: 1mA 01: 50uA 10: 2mA 11: 100uA																																																
POW_LVL [2:0]	POW_DET 检测阈值控制位 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="3">POW_LVL[2:0]</th> <th>检测电压 (R=50Ω)</th> <th>检测电流</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1.25mV</td> <td>25uA</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>2.5mV (default)</td> <td>50uA</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>3.75mV</td> <td>75uA</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>5mV</td> <td>100uA</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>6.25mV</td> <td>125uA</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>7.5mV</td> <td>150uA</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>8.75mV</td> <td>175uA</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>10mV</td> <td>200uA</td> </tr> </tbody> </table>				POW_LVL[2:0]			检测电压 (R=50Ω)	检测电流	0	0	0	1.25mV	25uA	0	0	1	2.5mV (default)	50uA	0	1	0	3.75mV	75uA	0	1	1	5mV	100uA	1	0	0	6.25mV	125uA	1	0	1	7.5mV	150uA	1	1	0	8.75mV	175uA	1	1	1	10mV	200uA
POW_LVL[2:0]			检测电压 (R=50Ω)	检测电流																																													
0	0	0	1.25mV	25uA																																													
0	0	1	2.5mV (default)	50uA																																													
0	1	0	3.75mV	75uA																																													
0	1	1	5mV	100uA																																													
1	0	0	6.25mV	125uA																																													
1	0	1	7.5mV	150uA																																													
1	1	0	8.75mV	175uA																																													
1	1	1	10mV	200uA																																													

VCC_LVL[3:0]	VCC_DET检测阈值控制位				
	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
	1	0	1	0	4.4V (default)
	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]	BOR_DET检测阈值控制位				
	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)			基地址: 0x4000F400 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	RESERVE	VDET_TIME[1:0]		VDET_PRD[2:0]		
Write			D					
Reset:	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			
	VDET_Time[1:0]		检测时间	
	0	0	336us (default)	
	0	1	397us	
	1	0	519us	
	1	1	1068us	
VDET_PRD[2:0]	Hold&Sleep 模式下 VCC_DET & BOR_DET & LVDIN_DET 分时检测的周期设定			
	VDET_PRD[2:0]			检测周期
	0	0	0	16.5ms
	0	0	1	33ms
	0	1	0	67ms (default)
	0	1	1	134ms
	1	0	0	268ms
	1	0	1	536ms
	1	1	0	1072ms
	1	1	1	2144ms

PMUIE (PMU 中断使能寄存器)		基地址: 0x4000F400 偏移地址: 0CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	POWIE	LVDIN1IE	LVDIN0IE	BORIE	VCCIE
Write					E	E		
Reset:	0	0	0	0	0	0	0	0

注: 需要同时使能 PMUIE 使能的中断才有效。

位	功能描述
POWIE	POW检测中断使能位 0: 关闭 1: 开启
LVDIN1IE	LVDIN1检测中断使能位 0: 关闭 1: 允许



LVDINOIE	<b>LVDINO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)			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	POWIF	LVDIN1I F	LVDINOI 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
POWIF	<b>POW 检测中断标志位:</b> 当内部ADC比较器监测POWIN电压与VCC电压差(芯片功耗)高于用户设定的阈值的时, 该位置1。 软件写0清0。
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。
LVDINOIF	<b>LVDINO检测中断标志位</b> 当外部引脚LVDINO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)			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	POW_FLG	LVDIN1_FLG	LVDINO_FLG	BOR_FLG	VCC_FLG
Write				X	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
POWT_FLG	<b>POWIN 引脚电压状态</b> 0: 表示 POWIN 引脚电压与 VCC 电压差小于设定阈值 (POW_LVL[2:0]) 1: 表示 POWIN 引脚电压与 VCC 电压差大于设定阈值 (POW_LVL[2:0])
LVDIN1_FLG	<b>LVDIN1 引脚电压状态</b> 0: 表示 LVDIN1 引脚电压小于 1.21V 阈值 1: 表示 LVDIN1 引脚电压大于 1.21V 阈值
LVDINO_FLG	<b>LVDINO 引脚电压状态</b> 0: 表示 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])

PDTFLT (POWER_DET_FILTER 低功耗检测滤波设置寄存器)			基地址: 0x4000F400 偏移地址: 1CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	X	Filter[3:0]			
Write								
Reset:	0	0	0	0	0	1	1	1

位	功能描述
Filter[3:0]	<b>低功耗检测滤波设置</b> 滤波时间为 $1 + \text{Filter}[3:0]$ ，也就是最长时间为 16 秒，Filter 的每一个值代表每秒一次的低功耗检测比较器的输出值，以用户设定 8 秒（默认值）为例： 只有当芯片内低功耗检测比较器连续监测 8 次都是相同的值（1 或者 0），才会更新 POW_FLG 标志。 只有当芯片内低功耗检测比较器连续监测 8 次都是 1，才会更新 POWIF 标志。

## 5 调试支持

### 5.1 概况

HT6X2X 采用的是 Cortex-M0 内核，该内核含有硬件调试模块。Cortex-M0 处理器支持以下调试特性：

- 程序的暂停、恢复以及单步执行；
- 访问处理器内核寄存器和特殊寄存器；
- 硬件断点（最多 4 个）；
- 软件断点（BKPT 指令）；
- 数据监视点（最多两个）；
- 动态存储器访问；
- 支持串行线（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/ SEG13/ SWIO
输出	SW 时钟	PB. 15/ SEG15/ SWCLK

### 5.3 SW 口使用说明

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

注：X 表示不可用

仿真口在上电/服务和启动之后的初始状态：

test 脚拉低，仿真口便被内部强制配置仿真功能，SWTCK 固定输入，SWIO 输出或输入（由执行指令决定，动态变化）

## 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，分别是 Sleep 模式和 Hold 模式。Sleep 模式和 Hold 模式最主要有以下两点区别：

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

### 6.2 睡眠模式（Sleep）

#### 6.2.1 SLEEP 模式下各模块开关

- 数字电源LD0\_1P5关闭，其供电的模块相应关闭；
- VRTC供电模块不关闭，RTC相关的晶振电路，TBS模块，分频补偿电路一直开启；
- 进入Sleep后，如果用户配置开启BOR和VCC检测功能，则BOR\_DET和VCC\_DET模块会由硬件分时开启以降低功耗；
- 在SLEEP模式下，看门狗可配开关。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2.4Flash控制功能。
- 进入SLEEP之前，如果配置LCD、TBS模块开启，在进入SLEEP模式后，即可实现LCD静态显示，温度和电池电压测量功能；

- 为降低SLEEP模式下的功耗，可以在进入SLEEP模式之前，配置GPIO的状态（详见GPIO章节），控制好芯片和外设的状态，防止通过GPIO往外部漏电；
- 如果用户期望在Sleep达到最低功耗：CLKCTRL0和CLKCTRL1寄存器全部清0
- 进入SLEEP 模式后，芯片内部会自动关闭LDO\_1P5（大功耗）输出，LDO\_LowPower（低功耗）保持输出1.5V。

## 6.2.2 SLEEP 模式下的唤醒

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

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

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

(1) **外部 INT 中断和串口 RX 引脚唤醒：**相应的 PIN 要配置为 INT 和 RX 功能（详见见 GPIO 单元），并使能外部中断和 UART 中断。当 INT 上升沿（或下降沿）中断使能相应 PIN 上出现上升沿（或下降沿），RX 中断使能相应 PIN 上出现下降沿，并保持低电平不少于 2 个 Fcpu 的时间，CPU 可从 SLEEP 模式下唤醒。注意，这里的 Fcpu 时钟为 CPU 时钟分频寄存器的输出时钟，如果用户在进入 Sleep 之前 CPU 时钟分频寄存器 SYSCLKDIV 的分频值过大（如 1/128），则会造成外部唤醒 PIN 上要给出很长时间（超过 2 个 Fcpu 时钟）的低电平信号，才能将芯片从 SLEEP 模式下唤醒。

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

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

(4) **UART 中断唤醒：**配置需要唤醒的 RX PIN 为 RX 功能，并配置相应管脚 RX 中断使能，打开相应内核中断：NVIC\_EnableIRQ(UARTx\_IRQn)；当 RX pin 出现有效的下降沿（不少于 2 个 Fcpu 的时间）唤醒信号，CPU 可从 sleep 模式唤醒。UART 唤醒不用使能 UART 模块。

## 6.2.3 从 SLEEP 模式唤醒后的唤醒方式确认

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

- 1) WAKEIF的RTCWKIF位为1，表示CPU是由RTC中断信号引起了唤醒。具体的中断源由RTC的8种中断源确定，可以查询RTCIF对应的位来确认是哪种RTC中断引起的唤醒，详见RTC单元。
- 2) WAKEIF的PMUWKIF位为1，表示CPU是由PMU中断信号引起了唤醒。具体的中断源由PMU的4种中断源确定，可以查询PMUIF对应的位来确认，详见PMU单元（电源单元）。

## 6.2.4 进入 Sleep 模式

Sleep模式通过CortexM0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）

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

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

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

在Hold模式下看门狗可配开关，Hold模式下看门狗配置详见2.4Flash控制功能

### 6.3.1 进入 Hold 模式

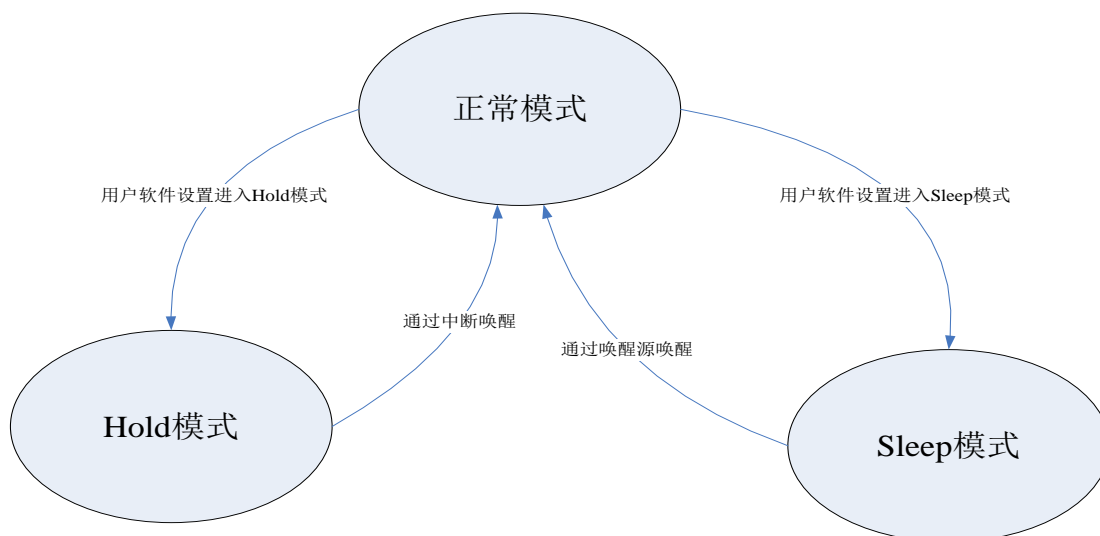
Hold模式通过CortexM0的系统自带指令WFI进入。

进入Hold指令如下：

```
SCR = 0x00;
__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。

## 6.5 模式转换图



## 6.6 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

## 6.7 特殊功能寄存器说明

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

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

WAKEIF (唤醒标志寄存器)			基地址: 0x4000F400 偏移地址: 18H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	X	INT9WKI	INT8WKI	INT7WKI	RX6WKIF	TIMER5	TIMER4	X
Write:		F	F	F				
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	RTCWKIF	TBSWKIF	X	X	X



Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	RX5WKIF	RX4WKIF	RX3WKIF	RX2WKIF	RX1WKIF	RX0WKIF	INT6WKIF
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	INT5WKIF	INT4WKIF	INT3WKIF	INT2WKIF	INT1WKIF	INT0WKIF	X	PMUWKIF
Write:	F	F	F	F	F	F		
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，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
INT9WKIF	<b>INT9唤醒标志</b> INT9唤醒发生时，此位置为1
INT8WKIF	<b>INT8唤醒标志</b> INT8唤醒发生时，此位置为1
INT7WKIF	<b>INT7唤醒标志</b> INT7唤醒发生时，此位置为1
RX6WKIF	<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。
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）
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
INT6WKIF	<b>INT6唤醒标志</b> INT6唤醒发生时，此位置为1
INT5WKIF	<b>INT5唤醒标志</b> INT5唤醒发生时，此位置为1
INT4WKIF	<b>INT4唤醒标志</b> INT4唤醒发生时，此位置为1
INT3WKIF	<b>INT3唤醒标志</b> INT3唤醒发生时，此位置为1
INT2WKIF	<b>INT2唤醒标志</b> INT2唤醒发生时，此位置为1
INT1WKIF	<b>INT1唤醒标志</b> INT1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
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

HT6023提供PA[0..13], PB[0..15], PC[0..14], PD[0..15], PE[0..8], PG[14..0], PH[4..0]并行端口, 支持90个双向I/O引脚。

HT6025提供PA[0..13], PB[0..15], PC[0..14], PD[0..15], PE[0..8]并行端口, 支持70个双向I/O引脚。

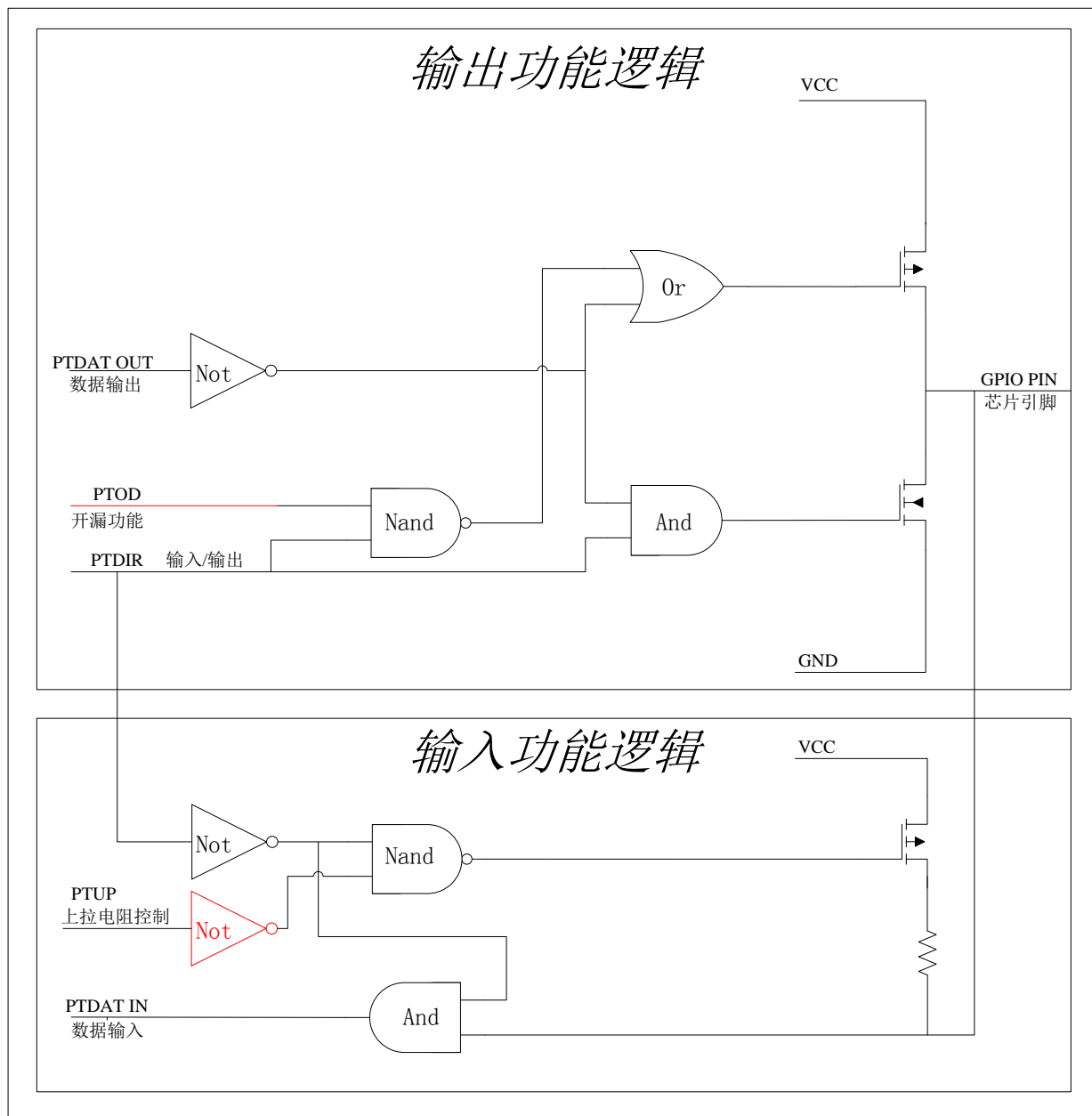
HT6027提供的I/O包括: PA[4..10, 12, 13], PB[4..15], PC[0..14], PD[3..15], PE[3..5, 7, 8], 支持54个双向I/O引脚。

每个I/O口输出都可配为推挽输出或者开漏输出; 若配为输入模式, 则输入上拉可配置, 输入上拉电阻为约90k。

各个I/O都具有5mA的输出驱动能力, 部分具备大驱动能力的I/O参见1.4引脚定义。

各个I/O均配置标准的防止ESD的二极管防护电路, 保证I/O的可靠性。

## 7.2 芯片引脚结构说明



## 7.3 I/O 端口基地址列表

GPIO 模块寄存器基地址:

0x40011000 (PA 端口);

0x40011100 (PB 端口);

0x40011200 (PC 端口);

0x40011300 (PD 端口);  
 0x40011400 (PE 端口);  
 0x40011600 (PG 端口);  
 0x40011700 (PH 端口);

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

GPIO 模块寄存器基地址: 0x40011500 (大电流端口);

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

## 7.4 特殊功能寄存器说明

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

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

注: 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

AFCFG (写保护) (端口功能配置寄存器 2)			基地址: 0x40011000--0x40011400 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8

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

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

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

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

PTUP (端口上拉配置寄存器)			基地址: 0x40011000—0x40011400 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PT[15:8]							
Write:	PT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PT[7:0]							
Write:	PT[7:0]							
Reset:	0	0	0	0	0	0	0	0

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

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

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

注：端口数据寄存器 PTDAT 说明及数据读取

1) 芯片引脚选择 GPIO 功能或复用数字功能

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

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

2) 芯片引脚选择复用模拟功能，PTDAT 相应 bit 位值，固定为 0

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

注：本寄存器只可写入。

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

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

注：本寄存器只可写入。

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

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

注：本寄存器只可写入。

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

PTOD	基地址: 0x40011000--0x40011400
------	-----------------------------



(端口开漏配置寄存器)			偏移地址: 20H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PT[15:8]							
Write:								
Reset:	1	1	1	1	1	1	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	PT[7:0]							
Write:								
Reset:	1	1	1	1	1	1	1	1

位	功能描述
PT[15:0]	端口开漏配置位（此寄存器只在对应端口配置为数字输出时才有效） 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）

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	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	PA8HD	PA7HD	PA6HD	PC0HD
Write:								
Reset:	0	0	0	0	0	0	0	0

注：该寄存器为单独的起始地址

位	功能描述
PA8HD, PA7HD, PA6HD, PC0HD	端口大电流驱动配置（分别对应 PA8, PA7, PA6, 和 PC0 端口） 0: 普通驱动能力 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. POWIE	PMUIF. POWIF	POWIN与VCC电压差检测中断
		PMUIE. LVDINOIE	PMUIF. LVDINOIF	LVDINO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
EXTIO-6	2-8	EXTIE. RIE[6:0]	EXTIF. RIF[6:0]	外部输入引脚上升沿中断
		EXTIE. FIE[6:0]	EXTIF. FIF[6:0]	外部输入引脚下降沿中断
UART0-5 (UART3 和 UART4 有 7816 功能)	9-14	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
		ISO7816CON. PRDIE	ISO7816STA. PRDIF	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)		
TMRO-3	15-18	TMRIE. CMPIE	TMRIF. CMPIF	比较中断
		TMRIE. CAPIE	TMRIF. CAPIF	捕获中断
		TMRIE. PRDIE	TMRIF. PRDIF	周期性溢出中断
TBS	19	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. ADCBATCMP	TBSIF. ADCBATCMPIF	电池电压比较中断

		IE		
		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. ADCBATIE	TBSIF. ADCBATIF	电池电压测量中断
		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
I2C	21		I2CCON. SI	I2C 传输中断
SPI0-1	22-23		SPISTA. SPIF	SPI 传输完成中断
			SPISTA. MODF	SPI 传输错误中断
SelfTestFreq	24	RCCaliIE. LRCIE	RCCaliIF. LRCIF	LRC 测频中断
		RCCaliIE. HRCIE	RCCaliIF. HRCIF	HRC 测频中断
TMR4-5	25-26	TMRIE. CMPIE	TMRIF. CMPIF	比较中断
		TMRIE. CAPIE	TMRIF. CAPIF	捕获中断
		TMRIE. PRDIE	TMRIF. 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	28-30	EXTIE2. RIE[2:0]	EXTIF2. RIF[2:0]	外部输入引脚上升沿中断
		EXTIE2. FIE[2:0]	EXTIF2. FIF[2:0]	外部输入引脚下降沿中断
DMA	31	DMAIE. TCIE	DMAIF. TCIF	DMA 传输完成中断
		DMAIE. BCIE	DMAIF. BCIF	DMA 块传输完成中断
		DMAIE. TEIE	DMAIF. TEIF	DMA 传输错误中断

注:

1. 只有 UART3 和 UART4 有 7816 功能, 因此只有该两个通讯端口有 7816 溢出中断, 7816 接收中断和 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, HT6023 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 功能前应先将对对应 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	0x0000	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	0x0000	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)		基地址: 0x40011800 偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	RIE[6:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	FIE[6:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RIE[6:0]	INT0-6 外部输入引脚上升沿使能 0: 禁止 1: 使能 注: 只有这些位置 1, 对应的中断标志才能置起来
FIE[6:0]	INT0-6 外部输入引脚下降沿使能 0: 禁止 1: 使能 注: 只有这些位置 1, 对应的中断标志才能置起来

EXTIF (外部中断标志寄存器)		基地址: 0x40011800 偏移地址: 04H	
----------------------	--	------------------------------	--

	Bit15	14	13	12	11	10	9	Bit8
Read:	RIF[6:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	FIF[6:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

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

PINFLT (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
Read:	RXFLT[6:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	INTFLT[6:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RXFLT[6:0]	RX0-6PIN 输入引脚数字滤波功能使能 0: 禁止 1: 使能
INTFLT[6:0]	INT0-6PIN 输入引脚数字滤波功能使能 0: 禁止 1: 使能

注: 用户在使用 INT 中断或 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。引脚数字滤波功能对相应 PIN 脚的其他功能 (GPIO 或其它复用功能) 无效。

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

Read:						RIE[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:						FIE[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RIE[2:0]	INT7-9 外部输入引脚上升沿中断使能 0: 禁止 1: 使能 注: 只有这些位置 1, 对应的中断标志才能置起来
FIE[2:0]	INT7-9 外部输入引脚下降沿中断使能 0: 禁止 1: 使能 注: 只有这些位置 1, 对应的中断标志才能置起来

EXTIF2 (外部中断标志寄存器 2)		基地址: 0x40011800 偏移地址: 14H						
	Bit15	14	13	12	11	10	9	Bit8
Read:						RIF[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:						FIF[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	0

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

PINFLT2 (引脚数字滤波使能寄存器 2)		基地址: 0x40011800 偏移地址: 18H						
	Bit15	14	13	12	11	10	9	Bit8

Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:						INTFLT[9:7]		
Write:						INTFLT[9:7]		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
INTFLT[9:7]	INT7-9PIN 输入引脚数字滤波功能使能 0: 禁止 1: 使能

注：用户在使用 INT 中断引脚功能的时候，必须要将 PINFLT2 寄存器中的相应的引脚数字滤波功能也打开。引脚数字滤波功能对相应 PIN 脚的其它功能（GPIO 或其它复用功能）无效。

## 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 RSTSR
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 RSTSR
4	掉电复位 (BOR)		2, PMU 模块的寄存器: PMUCON, VDETCFG, VDETPCFG 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
5	看门狗复位 (WatchDog)	三级	1, 复位状态寄存器 RSTSR
6	调试复位和软复位		2, PMU 模块的寄存器: PMUCON, VDETCFG, VDETPCFG 3, 唤醒标志寄存器 WAKEIF
7	唤醒复位 (WakeUp Reset)		4, GPIO 模块的寄存器: IOCFG, AFCFG, PTDIR, PTUP, PTDAT, PTOD 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, LCD 相关寄存器: LCDCLK, LCDCR, LCD_BUF[i] 8, CMU 相关寄存器: CLKCTRL0 (HRC_EN 和 PLL_EN 被复位), CLKCTRL1, LRCADJ, HRCADJ 9, RTC 模块的寄存器 10, TBS 模块的寄存器

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 和 BOR 复位时，内部复位信号 IRST 将保持有效，并保持 1088 个 Flf
- 2) 外部 RST, WDT Reset, SoftReset, Debug Reset, WakeUp Reset 复位时，内部 IRST 信号有效，并保持 64 个 Flrc。
- 3) 只有 VRTC 引脚的 POR/LBOR 复位会对 RTC 模块复位（可配，参见 2.4Flash 控制功能）

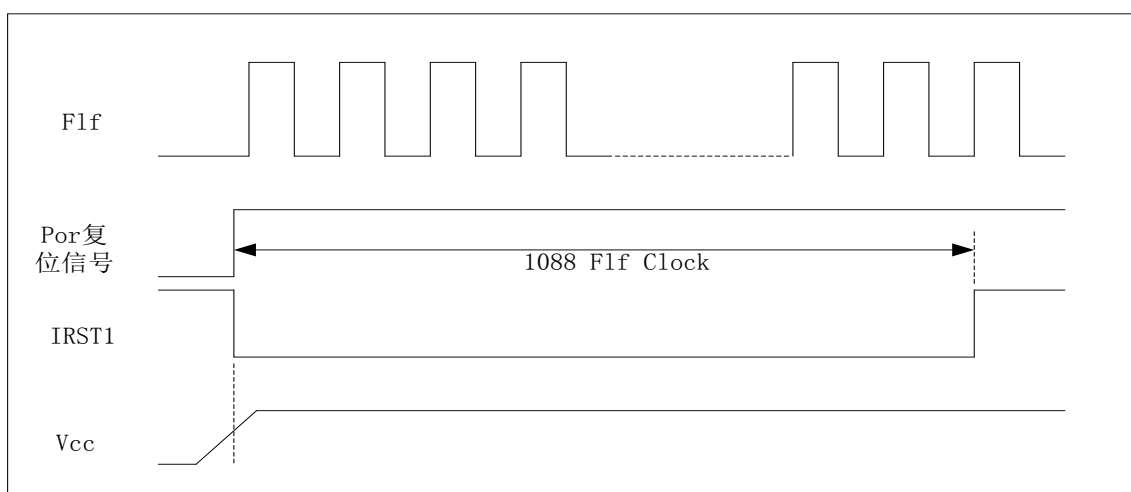


## 9.2.1 上电复位

当电源第一次加到芯片上时，上电复位电路检测电源电压  $V_{cc}$  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 F1f(注：如果不存在 F1f，系统自动切换到 F1rc)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：

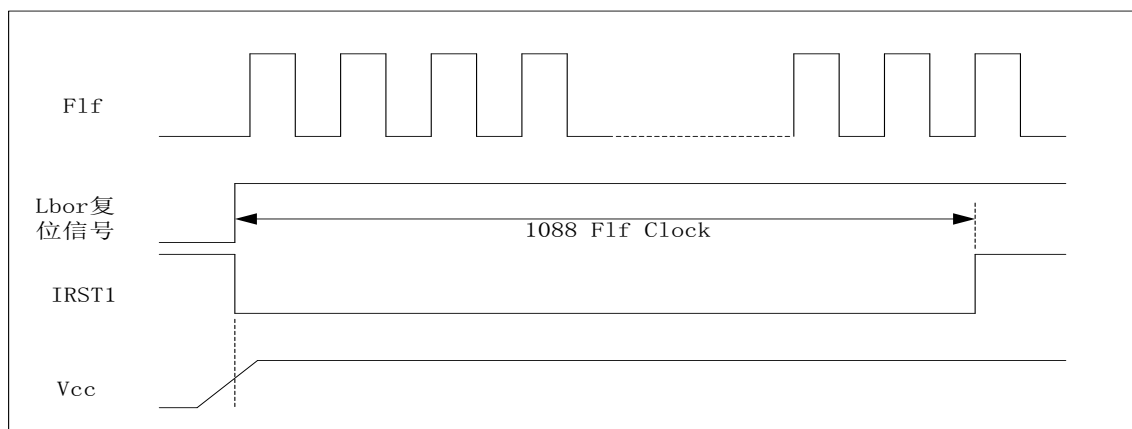
- 产生一个 POR 脉冲
- RTC 的复位只能由 VRTC 的电源的上电 POR 复位
- 内部复位信号 IRST 有效
- 计数 1088 个 F1f
- 复位状态寄存器 RSTSTA 的上电复位标志位 POR 被设置为 1，其他 RSTSTA 为被清为 0。
- CPU 从地址 0000H 执行程序



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。

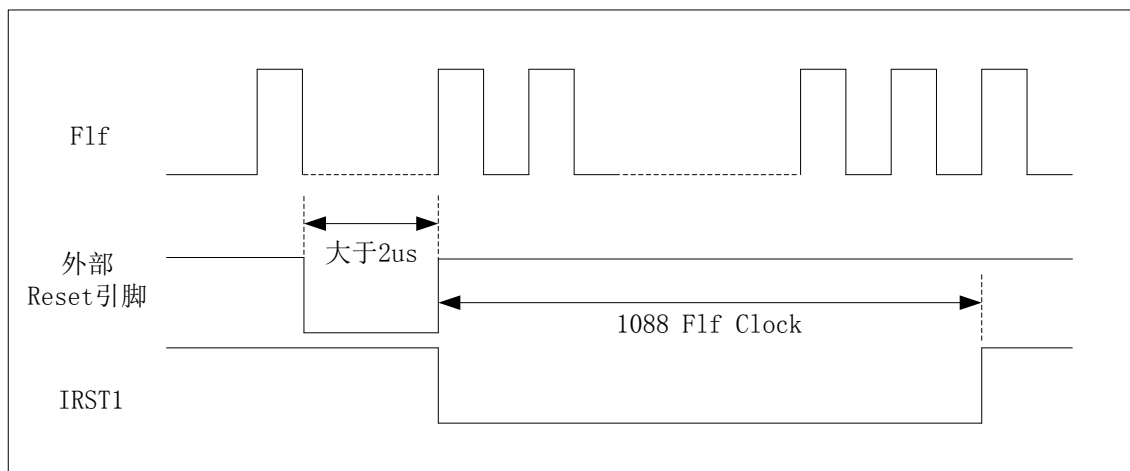


LBOR 复位说明

### 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 F1f。

如果/RST 低电平脉宽比 2us 窄，系统不发生复位。



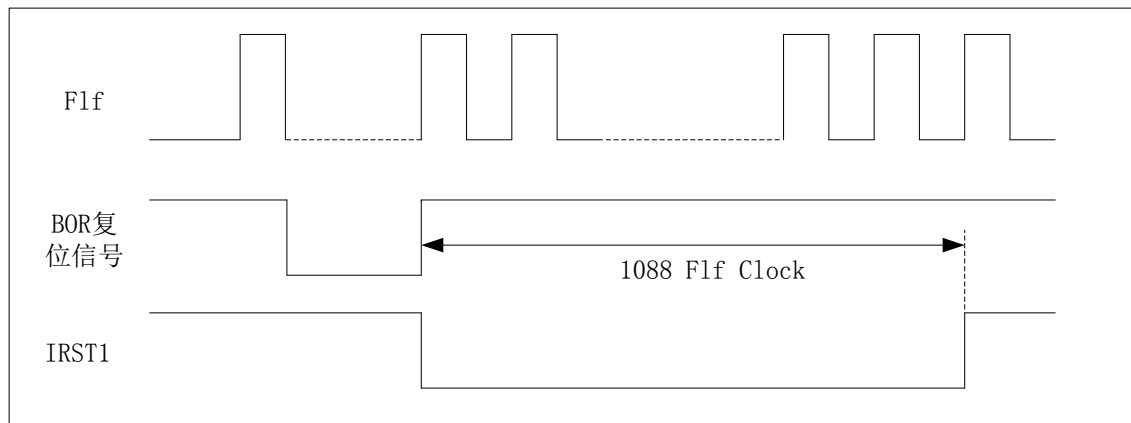
外部引脚复位说明

### 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 F1f 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：

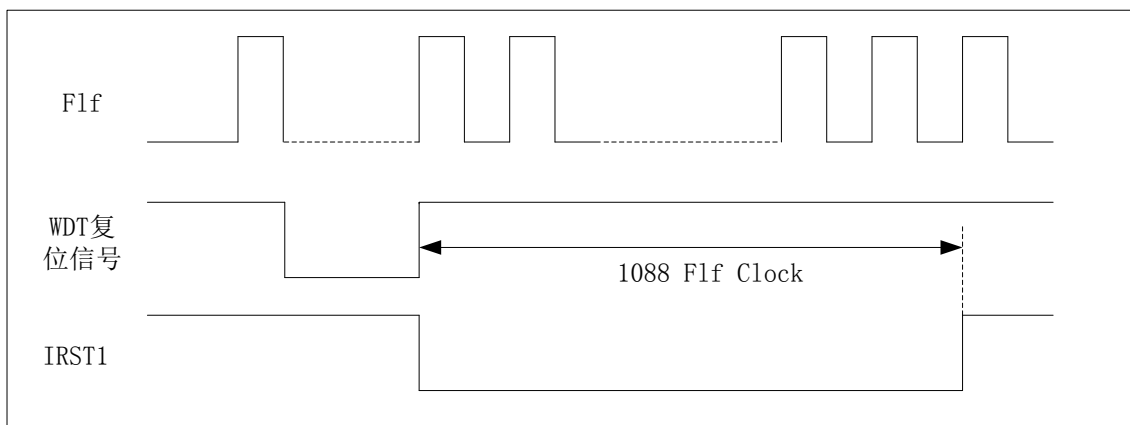
- 产生一个 BOR 脉冲
- 内部复位信号 IRST1 有效
- 计数 1088 个 F1f
- 复位状态寄存器 RSTSTA 的掉电复位标志位 BOR 被设置为 1，其他 RSTSTA 为被清为 0。
- CPU 从 0000H 开始执行程序



BOR 复位说明

## 9.2.5 看门狗复位

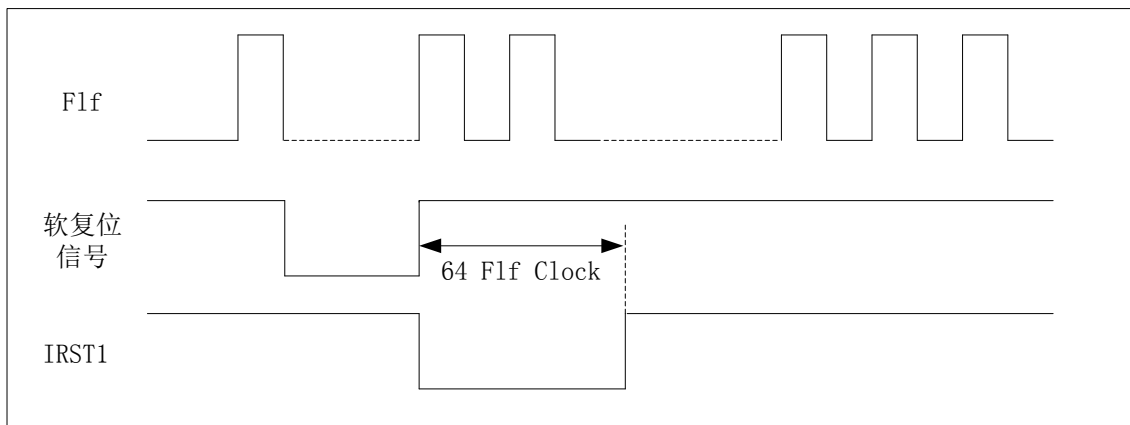
WatchDog Timer 溢出时将会产生导致内部复位 IRST1 有效, 复位状态寄存器的 WDT 复位标志位 WDT 被设置为 1。WDT 的复位脉宽为 1088 个 F1f。



WDT 复位

## 9.2.6 软复位

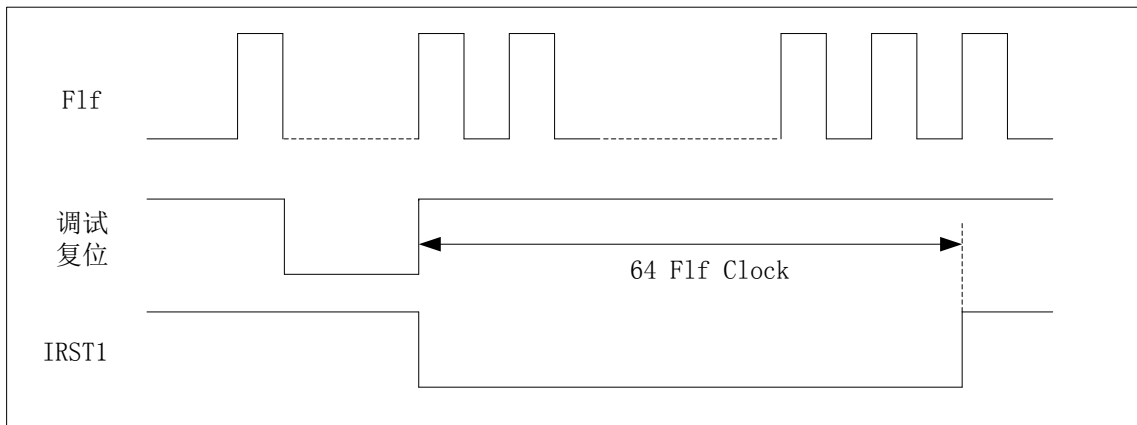
当向系统控制块的应用中断与控制状态寄存器寄存器 AIRCR 的 bit2 写入 1 时, 则产生一个软复位。推荐用户使用标准的 CMSIS 函数来操作软复位, 具体使用参见 19.4 CMSIS 函数说明。



软复位

## 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。

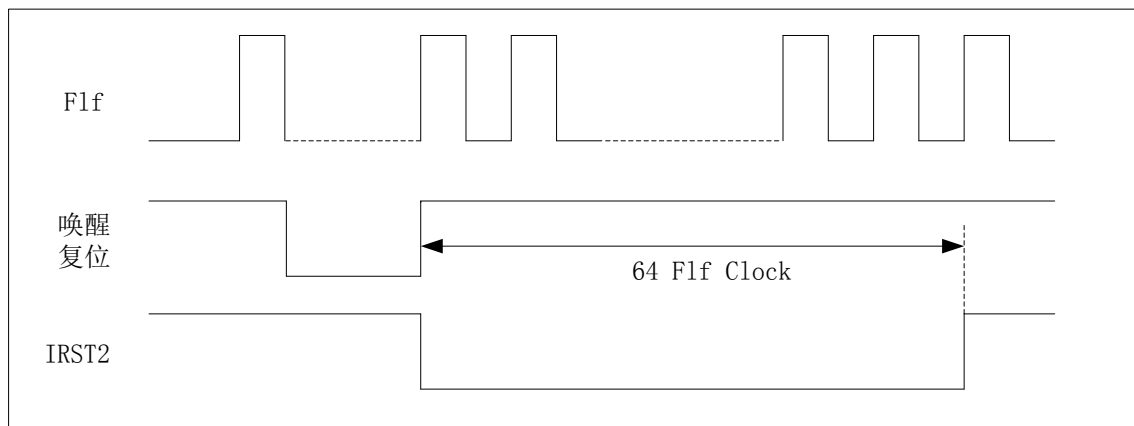


调试复位

### 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 F1f 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

基地值：0x4000F400（与 PMU 模块相同）				
偏移地址	名称	读写方式	复位值	功能描述
0x30	RSTSTA	R/W		复位标志寄存器

## 9.4 特殊功能寄存器说明

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

位	功能描述
Hold_flag	<b>Hold 模式下中断唤醒标志位</b> 0: 未发生 Hold 模式下中断唤醒标志位 1: 发生了 Hold 模式下中断唤醒标志位 写 0 清 0
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	<b>该标志位对用户无意义</b>
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。

## 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分别与两路7816接口复用, 通过MODESEL寄存器选择
- 波特率可软件设置
- 全双工通信口, 每个串口都可配置为红外调制输出, 调制信号的占空比可调, 调制极性可选
- 发送支持1个停止位或2个停止位
- 数据位宽支持7或8位
- 硬件自动完成奇偶校验, 数据接收完成的同时判断并提示奇偶校验错误, 给出标志。
- 接收/发送中断使能分别独立
- 每个串口都支持SLEEP或HOLD模式下的唤醒
- 红外38kHz调制信号可以来自PLL也可以来自HRC

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

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

芯片最多提供两路IS07816接口, 支持2个外部7816设备。

7816接口主要特点如下:

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

### 10.2 波特率计算

串口波特率由波特率生成器的值确定:

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

其中 SREL 是 16 位无符号数;  $F_{\text{sys}}$  是系统时钟。

注: RX0 和 RX1 不具有 2us 模拟滤波功能(详见 1.4 章节“引脚定义”), 因此可以支持较高波特

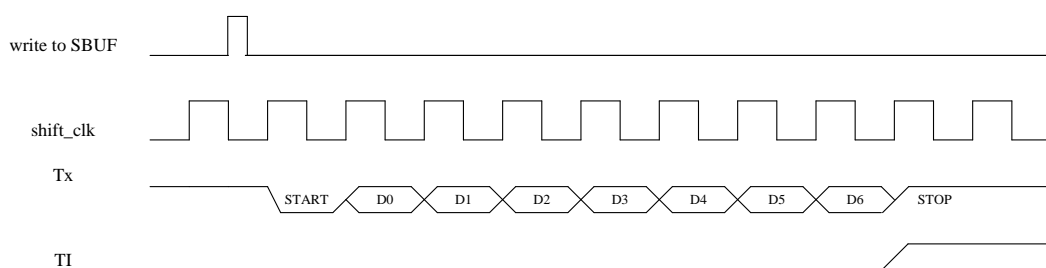
率通信，而其他 RX 引脚具有 2 $\mu$ s 模拟滤波功能，限制了通信速率，最大波特率只支持 115200bps。UART0/UART1 大波特率推荐配置如下：

波特率	SREL(十进制数)	波特率误差	UART0/UART1 收发状态
VCC=3.1V, Fsys = PLL 22MHz			
460800bps	23	0.466%	持续收发正常 (常温)
921600bps	11	0.487%	持续收发正常 (常温)
1000000bps	10	0.1%	持续收发正常 (常温)
1.5Mbps	6	3.4%	持续收发正常 (常温)
VCC=3.1V, Fsys = PLL 44MHz, PreFetch_EN=1 (指令预取功能)			
2.5Mbps	8	--	持续收发正常 (常温)

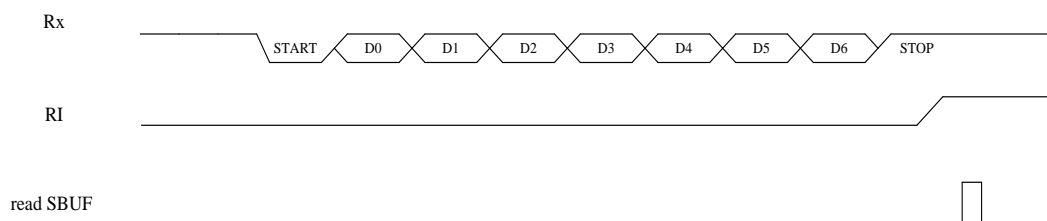
## 10.3 串口通讯模式说明

### 10.3.1 方式 1

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



图示：方式 1 时串行发送数据信息



图示：方式 1 时串行接收数据信息

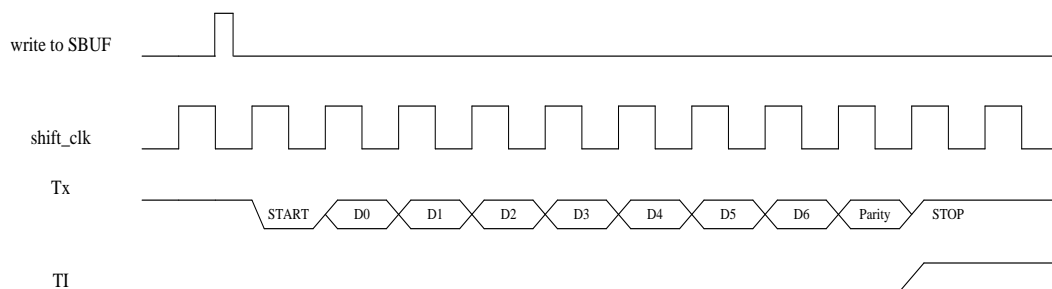
在方式 1 中，发送状态时，当一帧中最后一个数据发送完时，发送中断标志 TI 置“1”；接收状态



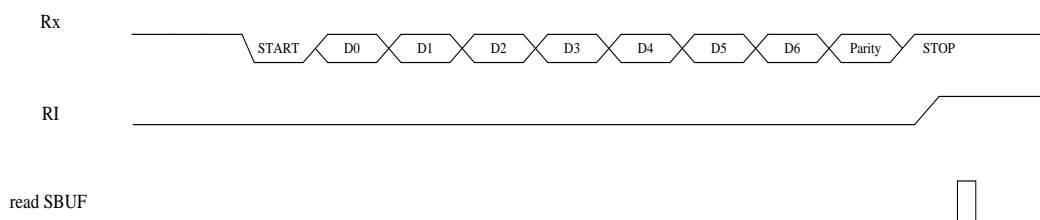
时，接收完最后一个数据位时，接收中断标志 RI 置 1。

### 10.3.2 方式 2

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



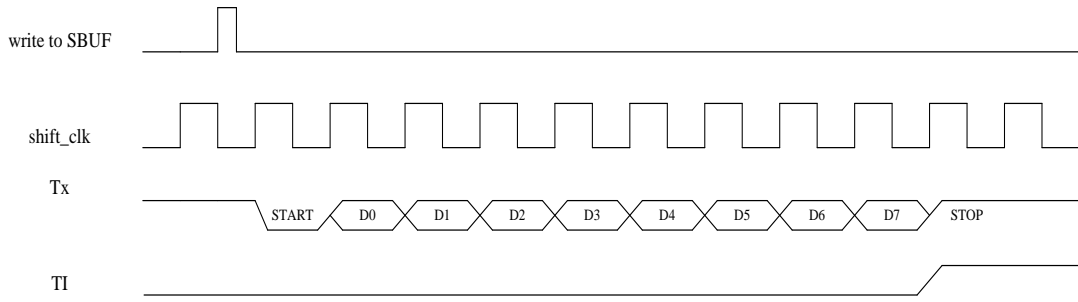
图示：方式 2 时串行发送数据信息



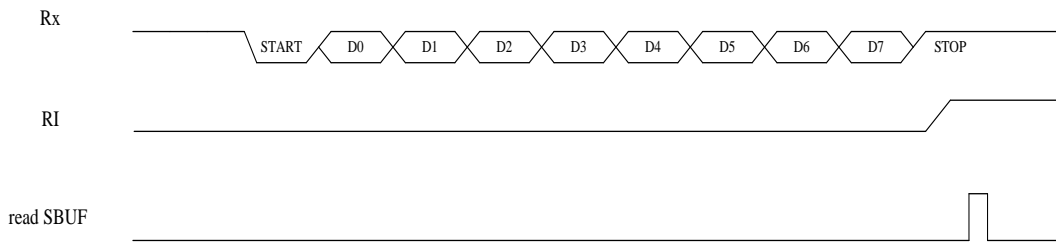
图示：方式 2 时串行接收数据信息

### 10.3.3 方式 3

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



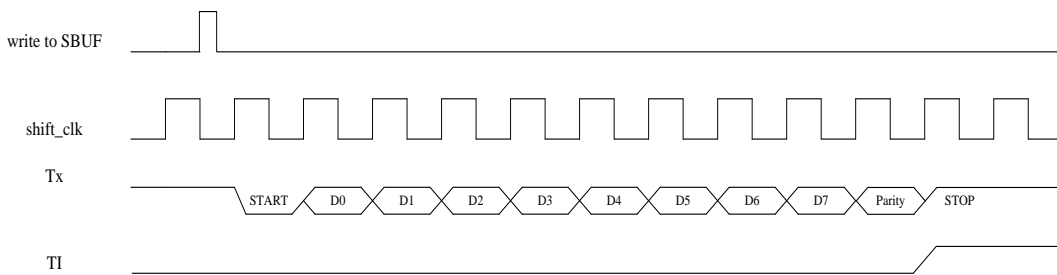
图示：方式 3 时串行发送数据信息



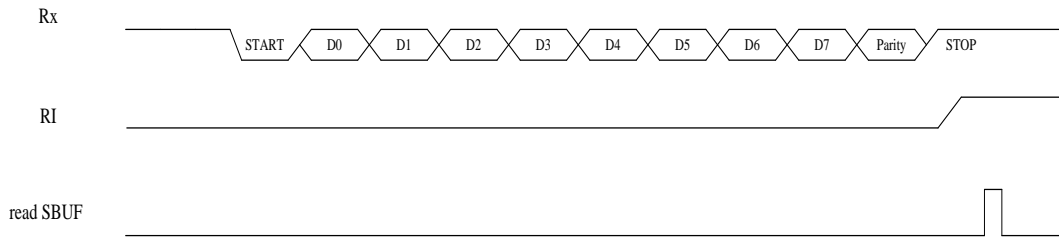
图示：方式 3 时串行接收数据信息

### 10.3.4 方式 4

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



图示：方式 4 时串行发送数据信息



图示：方式 4 时串行接收数据信息

## 10.4 7816 接收和发送

### 10.4.1 7816 数据发送

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

1. 发送起始位(0); (第 1ETU)
2. 发送 8bit 数据位; (第 2-9ETU)
3. 发送 1bit 校验位; (第 10ETU)
4. 读取接收到的 CKACK 信号; 如果 CKACK=0, TX\_PAR 置为“1”, 如果 CKACK=1, TX\_PAR 置为“0”; (第 11ETU)
5. 处于发送等待状态, 2 个 ETU;
6. 此时一个数据帧发送完成, SDIF=“1”, 如果 SDIE=1, 则此状态结束时, 产生发送中断。

如果 CKACK=1, 或禁止自动重发 (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。

### 10.4.2 7816 数据接收

在IDLE状态下，如果在接收端口(I0)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。

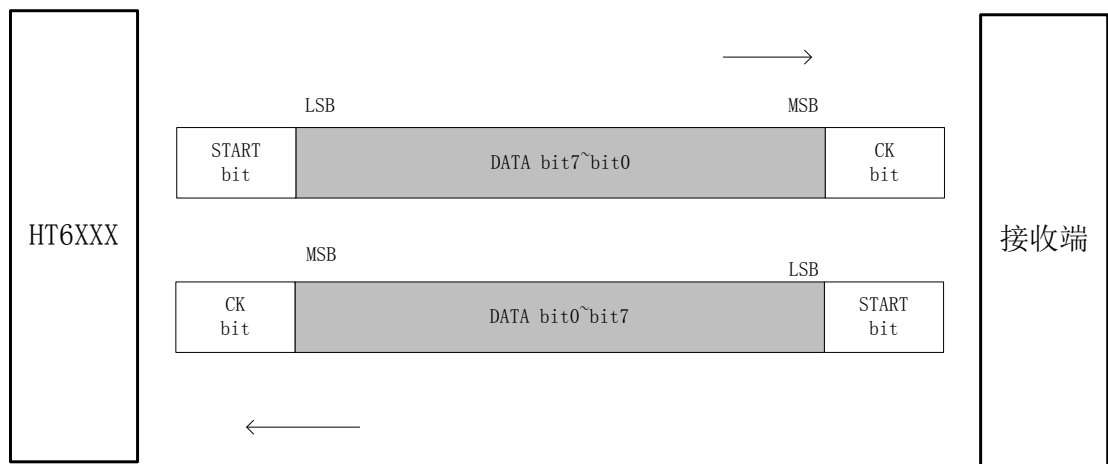
1. 接收起始位 (0); (第1ETU)
2. 接收 8bit 数据位; (第2-9ETU)
3. 接收 1bit 校验位; (第10ETU)
4. 向发送端发送 CKACK 信号。如果校验正确，或者禁止自动重收(AUTORXEN =0)，则发送 1，否则发送 0。(CKACK 的宽度，可以通过 ACKLEN[1:0]来配置)

校验位	AUTORXEN	7816I0
正确	“0” 禁止自动重收	“1”
正确	“1” 使能自动重收	“1”
错误	“0”	“0”

错误	“1”	“0”
----	-----	-----

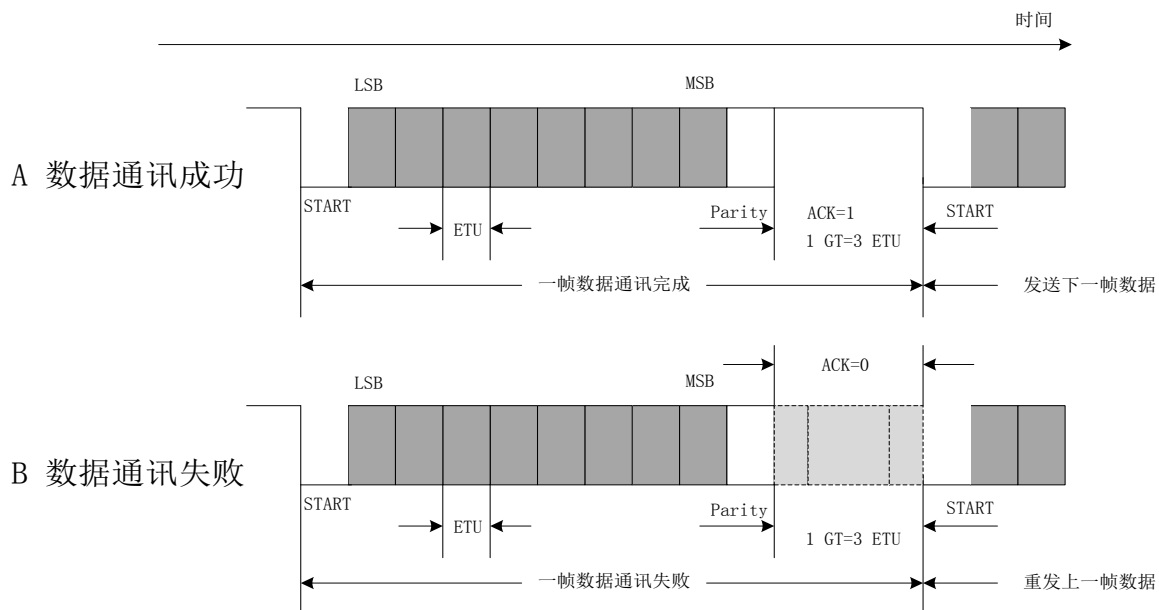
5. 此状态结束时，回到 IDLE 状态，产生接收中断。在中断中判断，如果校验正确，读取接收SBUF中的数据。（在接收的过程中，如果程序有对SBUF3/4写入的动作，该写入是无效的，需要等待）

### 10.4.3 7816 通讯示意图

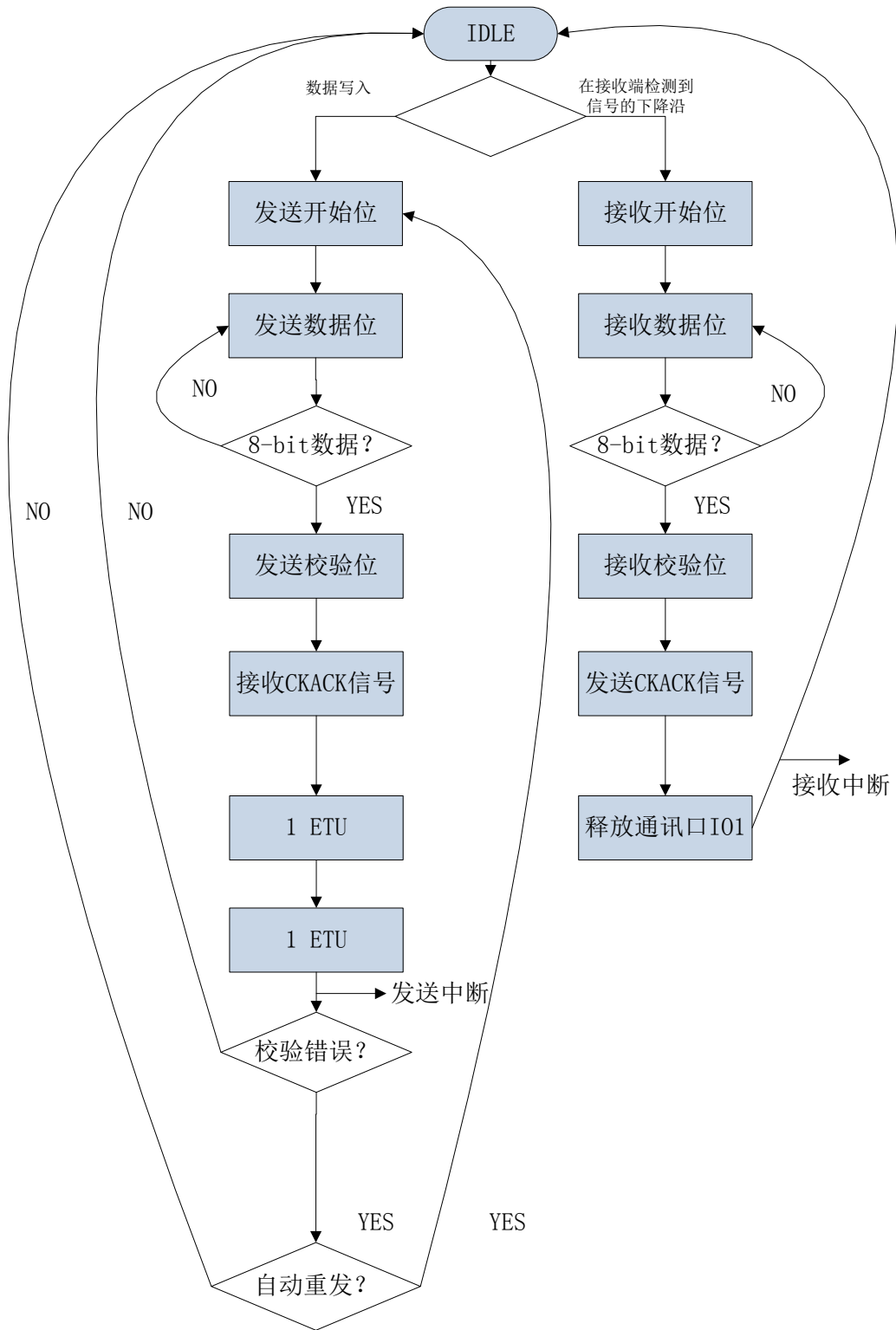


HT6XXX 和从机接收端通讯数据示意图

HT6XXX进行通信时，收发一个bit 的需要的的时间被定义为基础时间单位ETU (Elementary Time Unit)。发送端发送完一个数据帧后，接收端需要一定的时间对接收到的数据进行校验，然后再根据校验结果发送下一个数据帧或重发上一帧数据，即，连续发送两个数据帧之间的等待时间，该等待时间被定义为检测时间GT(Guarding Time)，一般， $1 GT = 3 ETU$ 。



HT6XXX 数据通讯发送示意图



## 10.5 特殊功能寄存器列表

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

## 10.6 特殊功能寄存器说明

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 1: 7816 功能

注：该寄存器设置 7816 功能只对 UART3 与 UART4 有作用

UARTCON (UART 功能配置寄存器)			基地址: 0x40005000—0x40009000; 0x40000000; 0x40018000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PRDIE	X	X	X	X	UNEGAUTO 0	UNEG	STOPSEL
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	LENSEL	PARITYSEL[1:0]		PARITYE N	RXIE	TXIE	RXEN	TXEN
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述															
PRDIE	<b>串口上溢中断使能控制位</b> 0: 上溢中断禁止 1: 上溢中断使能															
UNEGAUTO	<b>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)</b> 0: 由用户软件控制 UART 接收/发送通讯是正逻辑还是负逻辑, 通过 UNEG 控制位 1: 由硬件自动配置 UART 接收通讯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 (默认) 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="margin-left: 20px;"> <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	<b>UART 通讯停止位长度选择位</b> 1: 2bit 0: 1bit															
LENSEL	<b>UART 通讯数据长度选择位</b> 1: 7bit 0: 8bit															
PARITYSEL	<b>UART 奇偶校验选择位</b> 11: 固定为 1 00: 固定为 0 01: 奇校验 10: 偶校验															
PARITYEN	<b>UART 奇偶校验使能位</b>															



	1: 使能 0: 禁止
RXIE	UART 接收中断使能位 1: 使能 0: 禁止
TXIE	UART 发送中断使能位 1: 使能 0: 禁止
RXEN	UART 接收使能位 1: 使能 0: 禁止
TXEN	UART 发送使能位 1: 使能 0: 禁止

ISO7816CON (7816 功能配置寄存器)			基地址: 0x40005000—0x40009000; 0x40000000; 0x40018000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	7816PAR	AUTORXE
Write:							ITY	N
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	AUTOTXE	REPTR1	REPTRO	ACKLEN1	ACKLENO	PRDIE	RXIE	TXIE
Write:	N							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
7816PARITY	奇偶校验选择位 0: 偶校验 1: 奇校验
AUTORXEN	自动重接收使能位 1: 使能 0: 禁止
AUTOTXEN	自动重发送使能位 1: 使能 0: 禁止
REPTR[1:0]	自动重收重发的次数上限 00: 0 次 01: 1 次 10: 2 次 11: 3 次

ACKLEN[1:0]	响应位的长度 00: 1bit 01: 1.5bit 10/11: 2bit
PRDIE	上溢中断使能位 1: 使能 0: 禁止
RXIE	接收中断使能位 1: 使能 0: 禁止
TXIE	发送中断使能位 1: 使能 0: 禁止

SREL (串口波特率发生寄存器)		基地址: 0x40005000—0x40009000; 0x40000000; 0x40018000 偏移地址: 0CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:	SREL[15:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SREL[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

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

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

SBUF (串口数据缓冲寄存器)		基地址: 0x40005000—0x40009000; 0x40000000; 0x40018000 偏移地址: 10H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	ITDly[7..0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SBUF[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ITDly[7..0]	当读 SBUF 时，ITDly 无效，读出来总是 0 当写 SBUF 时，高 8 位含义如下： ITDly[7..0]：等待 (ITDly[7..0]*8BaudRate) 时间给出 DMA 请求 <b>注：此功能只针对 UART6，对其他 UART，不存在 ITDly 控制位</b>
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)		基地址：0x40005000—0x40009000；0x40000000；0x40018000 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	POLASTA	PRDIF	PARITY	RXIF	TXIF
Write:				X				
Reset:	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 1：负极性
PRDIF	<b>接收上溢中断标志</b> 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老的数据覆盖了 0：未产生上溢； 写 0 清零，写 1 无效。
PARITY	<b>接收时奇偶校验的状态</b> 1：错误 0：正确 写 0 清零，写 1 无效。
RXIF	<b>接收中断标志</b> 1：接收数据完成，可从寄存器 SBUF 中读出 0：接收数据还未完成 写 0 清零，写 1 无效。
TXIF	<b>发送中断标志</b> 1：发送完成 0：发送未完成 写 0 清零，写 1 无效。

ISO7816STA (7816 状态寄存器)			基地址: 0x40005000—0x40009000; 0x40000000; 0x40018000 偏移地址: 18H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	RXERRST	TXERRST	PRDIF	RXIF	TXIF
Write:				AT	AT			
Reset:	0	0	0	0	0	0	0	0

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

IRCON (红外调制控制寄存器)			基地址: 0x40005000—0x40009000; 0x40000000; 0x40018000 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	X	X	IR38kSource	IRLVL	IRTX
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
IR38kSource	<b>红外 38k 调制信号来源</b> 0: 红外 38k 调制信号来源于 PLL 1: 红外 38k 调制信号来源于 HRC
IRLVL	<b>红外调制输出极性选择</b> 1: 负极性。 0: 正极性。
IRTX	<b>红外调制功能使能控制</b> 1: 使能 TX 输出的红外调制功能。 0: 关闭 TX 输出的红外调制功能。

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

IRDUTY (红外调制脉宽寄存器)			基地址：0x40005000—0x40009000；0x40000000；0x40018000 偏移地址： 34H					
	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	IRDUTY[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

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

NEGDETLEN (逻辑正反自动判断时间设置)			基地址：0x40005000—0x40009000；0x40000000；0x40018000 偏移地址： 38H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	NEGDETLEN[15..8]							
Write:								
Reset:	0	0	0	0	0	0	0	0

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

位	功能描述
NEGDETLEN[0...15]	配置 485 逻辑正反自动判断的数据长度???

# 11 LCD 模块

## 11.1 概述

HT6023 的 LCD 驱动单元最多可以支持 38 (Segment) \* 8 (Common) 输出, HT6025 最多可以支持 33 (Segment) \* 8 (Common) 输出。HT6027 最多可以支持 22 (Segment) \* 8 (Common)。主要特征如下:

- 软件可编程驱动方式。(具体要根据最终的 GPIO 引脚来定)
- HT6023: 38SEG \* 8COM, 40SEG \* 6COM, 42SEG \* 4COM
- HT6025: 33SEG \* 8COM, 35SEG \* 6COM, 37SEG \* 4COM
- HT6027: 22SEG \* 8COM, 24SEG \* 6COM, 26SEG \* 4COM
- 软件控制对比度可调
- LCD 驱动电压可选
- 1/3 Bias, 1/4 Bias 可选
- 1/4、1/6、1/8 Duty 可选

## 11.2 LCD 与 GPIO 口复用表

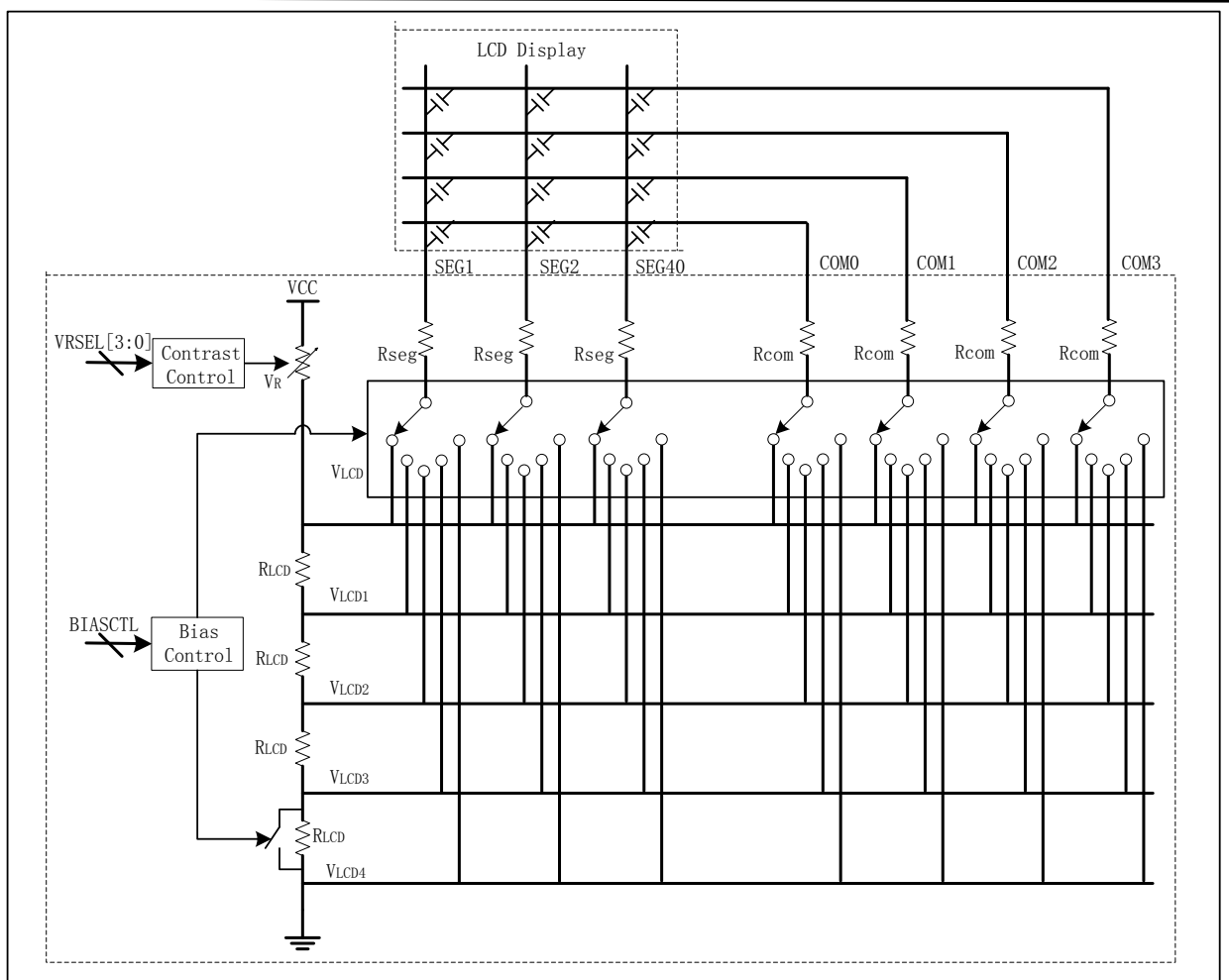
LCD 引脚定义	功能引脚定义	芯片引脚定义
COM0-COM3	PD. 8—PD. 11	PD. 8/COM0—PD. 11COM3
SEG0-SEG15	PB. 0—PB. 15	PB. 0/SEG0—PB. 15/SEG15
SEG16-SEG23	PD. 0—PD. 7	PD. 0/SEG16—PD. 7/SEG23
SEG24/COM4—SEG27/COM7	PD. 12—PD. 15	PD. 12/SEG24/COM4—PD. 15/SEG27/COM7
SEG28—SEG32	PA. 0—PA. 4	PA. 0/SEG28—PA. 4/SEG32
SEG33—SEG34	PA. 7—PA. 8	PA. 7/SEG33—PA. 8/SEG34
SEG35—SEG36	PA. 12—PA. 13	PA. 12/SEG35—PA. 13/SEG36
SEG37—SEG41	PG. 0—PG. 4	PG. 0/SEG37—PG. 4/SEG41

## 11.3 LCD 框图

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

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

VLCD(Vbias)用于控制 LCD 对比度, 参考 LCD 对比度部分。



LCD 的显示数据放在寄存器 LCD\_BUF 中，用于控制 LCD segment 的开关。当将某一段对应的 SEG 和 COM 都为 1 时，该段就会被点亮；否则不被点亮。

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

## 11.4 输出波形

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

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

LCD 驱动电压：

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

- $VLCD = VCC - V_{bias}$

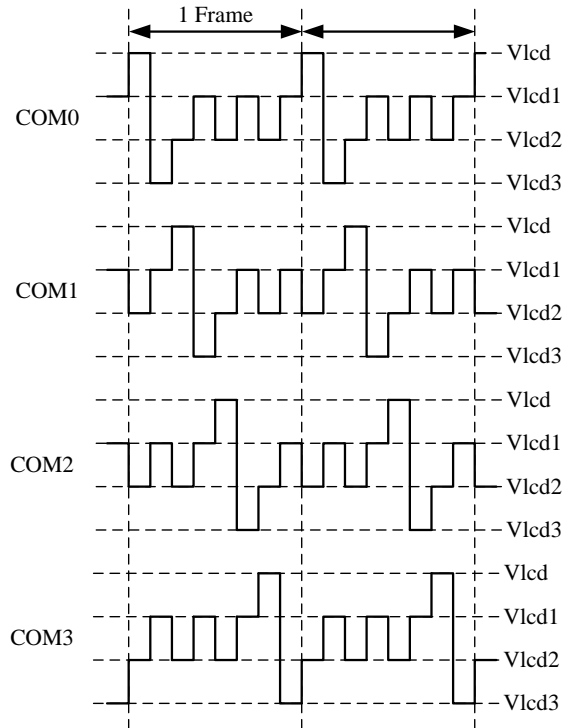
$V_{bias}$  用于控制 LCD 对比度，参考 LCD 对比度部分。

COM 输出波形：



## ① 1/4 Duty 输出波形

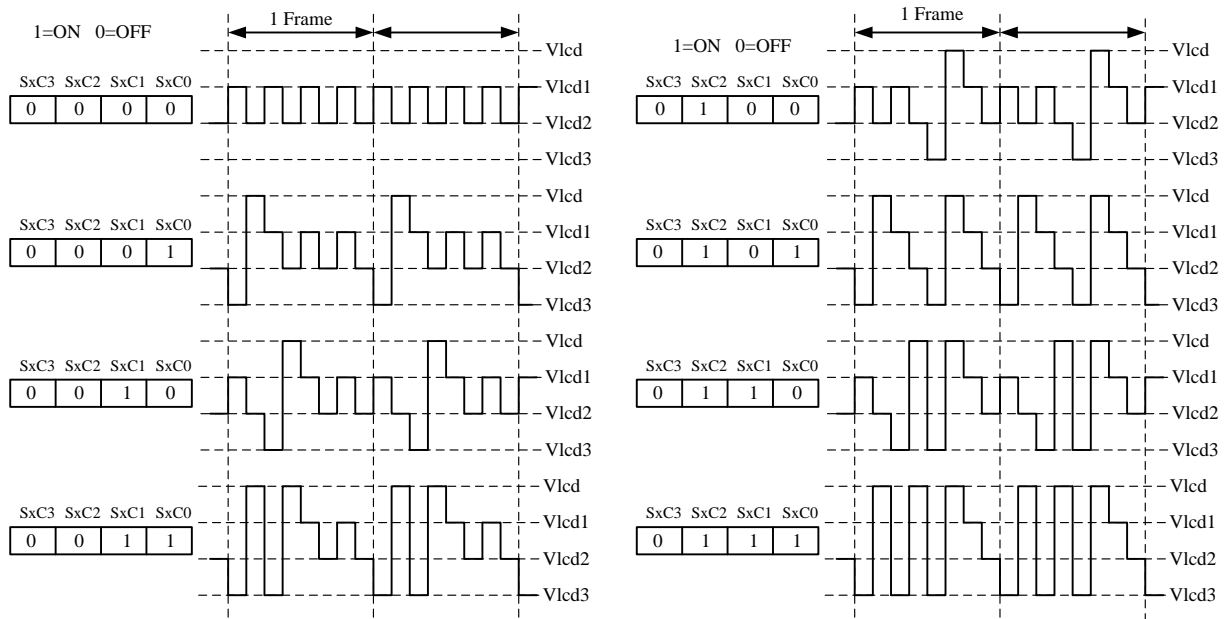
使用 COM0/1/2/3 都被使用。1 帧等于 4 个 LCD 波形时钟周期。



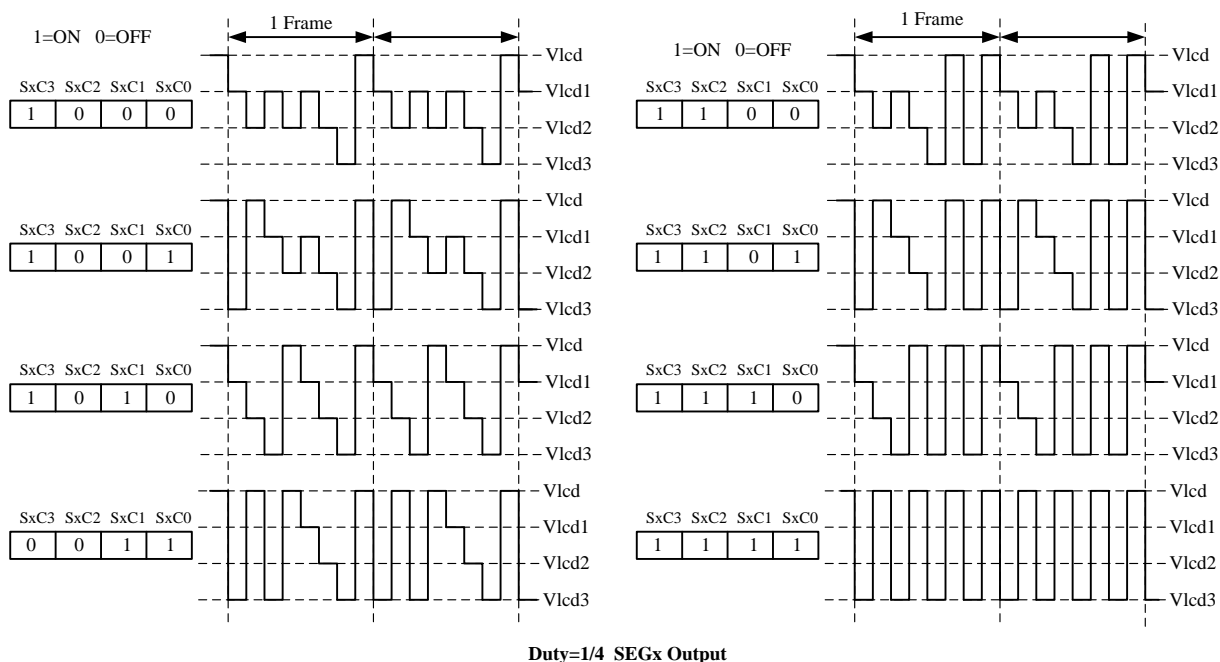
Duty=1/4 COM Output

## SEGMENT 输出波形:

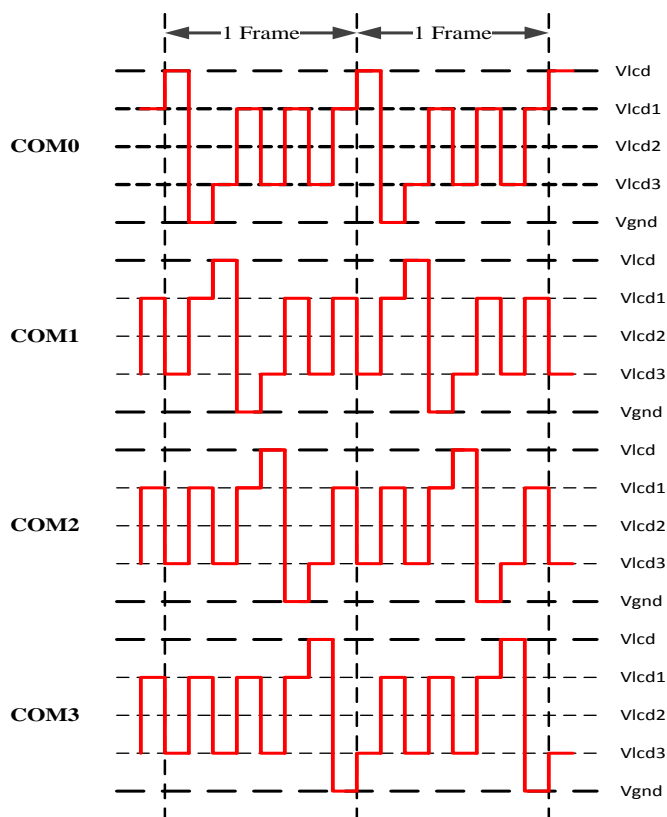
## ① 1/4 Duty



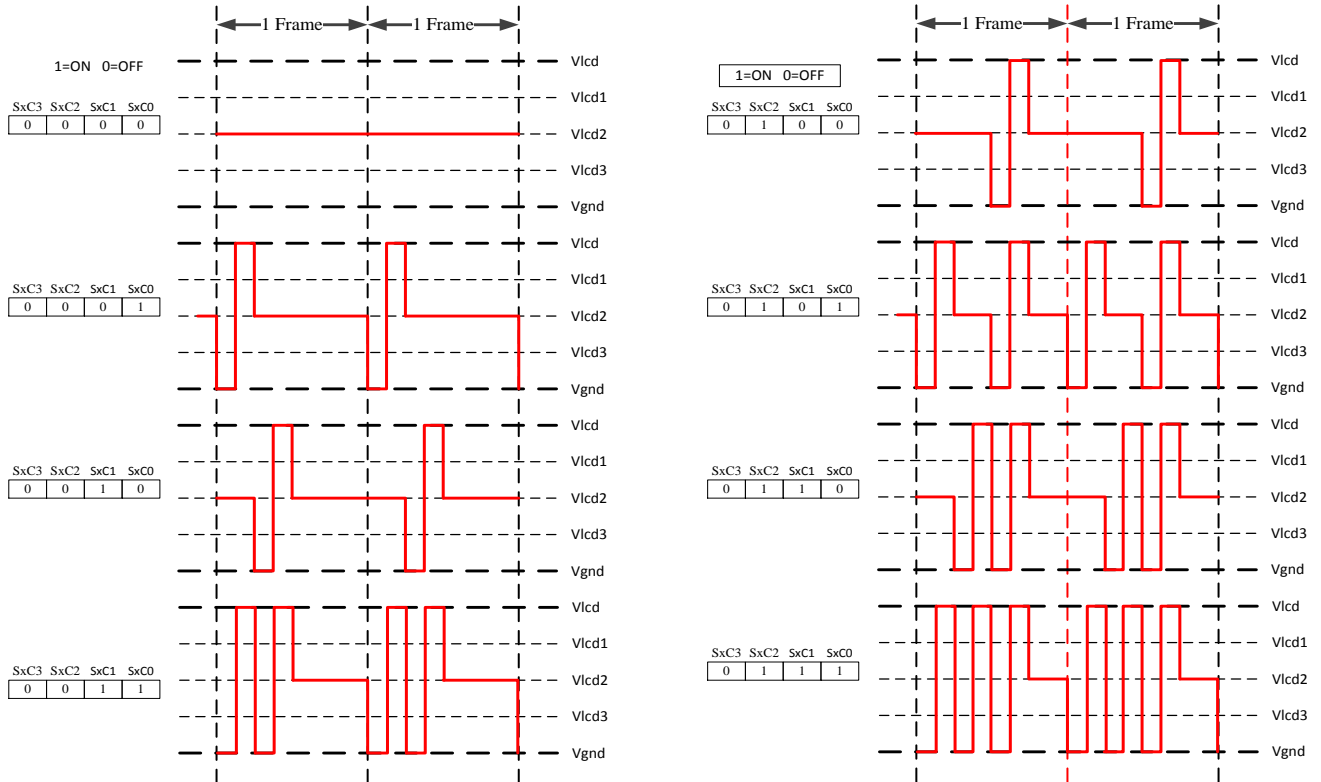
Duty=1/4 SEGx Output



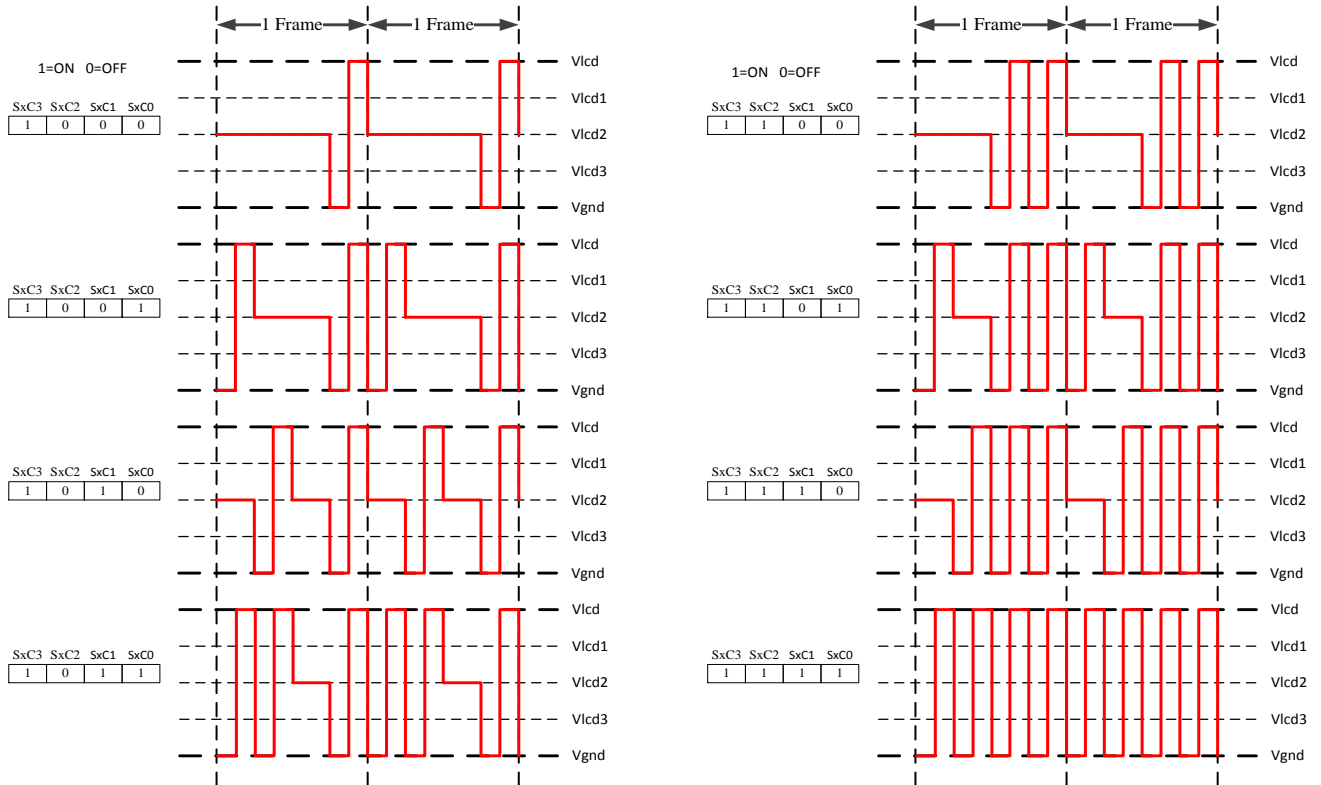
### COM 波形 (1/4Bias, 1/4Duty)



## SEGMENT 输出波形 (1/4Bias, 1/4Duty)



1/4Duty 1/4Bias SEGx Output



1/4Duty 1/4Bias SEGx Output

## 11.5 LCD 显示操作

LCD 的显示数据寄存器写功能:

LCD 单元的显示输出是采用往 LCD BUFF 内写入和对应得 SEG 输出的数据来输出显示的 LCD 段码。42Bytes 寄存器做为 LCD BUFF, 操作方式与一般寄存器相同, 同时与 LCD 的对应关系如下:

地址 LCD\_BUF[0]对应 SEG0 (COM7—COM0)

地址 LCD\_BUF[1]对应 SEG1 (COM7—COM0)

地址 LCD\_BUF[2]对应 SEG2 (COM7—COM0)

地址 LCD\_BUF[32]对应 SEG32 (COM7—COM0)

.....

地址 LCD\_BUF[40]对应 SEG40 (COM7—COM0)

地址 LCD\_BUF[41]对应 SEG41 (COM7—COM0)

## 11.6 特殊功能寄存器列表

基地址: 0x4000D000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	LCDCLK	R/W	0x0080	LCD 时钟频率选择寄存器
0x04	LCDCON	R/W	0x0089	LCD 驱动控制寄存器
0x10+i×4 (i=0~41)	LCD_BUF[i]	R/W	----	LCD 显示数据寄存器 (此寄存器为 RAM 类型, 复位值随机)

## 11.7 特殊功能寄存器说明

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

位	功能描述																
BIASCTL	<b>偏压驱动选择位</b> 1: 表示选择1/3 bias 偏压驱动 0: 表示选择1/4 bias 偏压驱动																
DUTY[2:0]	<b>LCD的duty控制选择位</b> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>DUTY1</th> <th>DUTY0</th> <th>COMMON选择</th> <th>Duty</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Com0~Com3</td> <td>1/4 duty</td> </tr> <tr> <td>0</td> <td>1</td> <td>Com0~Com5</td> <td>1/6 duty</td> </tr> <tr> <td>1</td> <td>X</td> <td>Com0~Com7</td> <td>1/8 duty</td> </tr> </tbody> </table>	DUTY1	DUTY0	COMMON选择	Duty	0	0	Com0~Com3	1/4 duty	0	1	Com0~Com5	1/6 duty	1	X	Com0~Com7	1/8 duty
DUTY1	DUTY0	COMMON选择	Duty														
0	0	Com0~Com3	1/4 duty														
0	1	Com0~Com5	1/6 duty														
1	X	Com0~Com7	1/8 duty														
LCLK[2:0]	<b>LCD扫描频率选择位:</b> LCD 单元的时钟来自低频时钟频率 F <sub>lf</sub> , F <sub>lf</sub> 经过分频之后作为 LCD 波形扫描频率 F <sub>lcd</sub> , F <sub>lcd</sub> 可以通过寄存器 LCLK[2:0]进行配置 (详见下表)。LCD 帧扫描频率 F <sub>frm</sub> =F <sub>lcd</sub> *duty。如: 1/4 duty 时, 帧扫描频率 F <sub>frm</sub> =F <sub>lcd</sub> /4。																

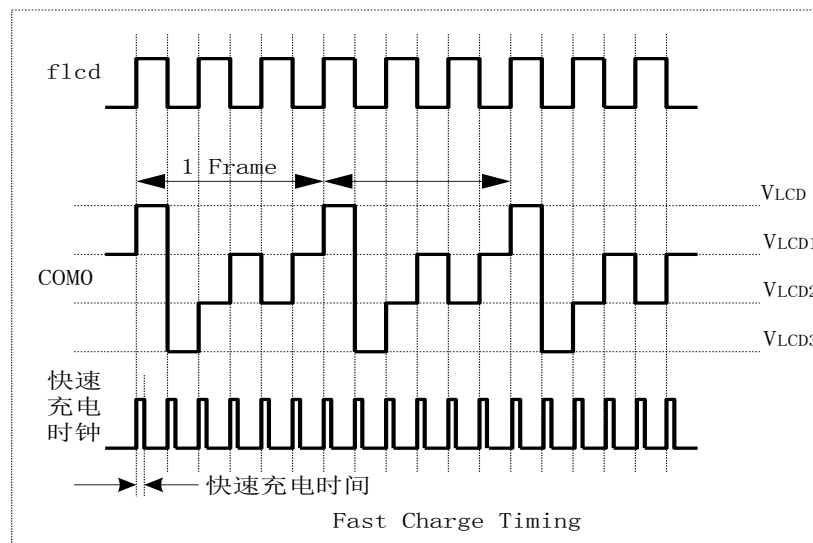
LCLK			Frame rate (Hz)		
LCLK2	LCLK1	LCLK0	1/4 duty	1/6 duty	1/8duty
1	0	0	128	85.3	64
0	0	0	64	42.7	32
0	0	1	32	21.3	16
0	1	0	16	10.7	8
0	1	1	8	5.3	4

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

LCD Control Register (LCDCON LCD 驱动控制寄存器)			基地址: 0x4000D000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	FCSET2
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	FCSET1	FCSET0	FC	LC	VRSEL3	VRSEL2	VRSEL 1	VRSEL 0
Write:								
Reset:	1	0	0	0	1	0	0	1

位	功能描述
FCSET2	<b>电阻分压结构控制位</b> 0: 保持原有的电阻分压结构 (default) 1: 改善后电阻分压结构 (大电流模式的电流更大, 小电流模式的电流更小) 改善后的结构, 选择快速充电模式, 显示效果更优, 功耗更小。

FCSET[1: 0]	<b>快速充电时间选择位</b>					
	FCSET1	FCSET0	快速充电时间			
	0	0	1/8个Flcd周期			
	0	1	1/16个Flcd周期			
	1	0	1/32个Flcd周期			
1	1	1/64个Flcd周期				
FC	<b>快速充电模式选择位</b>					
	和LC配合使用，用于确定LCD充电模式。					
	缺省状态下，LCD分压网络的电阻RLCD为24KΩ。根据LCD差异，可以选择RLCD=143KΩ，此时流过电阻网络的电流较小。					
	在选择RLCD=143KΩ时，设置控制位FC位1，可以选择快速充电模式，也就是说，每次LCD输出波形改变时，先选择RLCD=24KΩ进行快速充电，然后再切换到RLCD=143KΩ模式。快速充电时间可以通过FCSET[1:0]进行选择。					
LC	<b>慢速充电模式选择位</b>					
	和FC配合，用于确定LCD充电模式					
	FC	LC	LCD充电模式	RLCD默认结构阻值	RLCD改善后结构阻值	
	X	0	大电流充电模式	RLCD=24KΩ	RLCD=12KΩ	
	0	1	小电流充电模式	RLCD=143KΩ	RLCD=726KΩ	
1	1	短时间大电流，快速充电模式				
VRSEL[3: 0]	<b>LCD显示对比度设置位</b>					
	VRSEL3	VRSEL2	VRSEL1	VRSEL0	1/3 bias 对比度 (% of VCC)	1/4 bias 对比度 (% of VCC)
	0	0	0	0	97.4	99.1
	0	0	0	1	92.4	94.0
	0	0	1	0	85.8	88.8
	0	0	1	1	80.0	84.2
	0	1	0	0	75.0	80.0
	0	1	0	1	70.6	76.2
	0	1	1	0	66.6	72.8
	0	1	1	1	63.2	69.6
	1	0	0	0	60.0	66.6
	1	0	0	1	57.2	64.0
	1	0	1	0	54.6	60.0
	1	0	1	1	52.2	59.2
	1	1	0	0	50.0	57.2
	1	1	0	1	48.0	55.2
	1	1	1	0	46.2	53.4
	1	1	1	1	44.4	51.6



快速充电图示

## 12 WDT 模块

### 12.1 概述

HT6X2X 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

### 12.2 工作模式

不同工作模式下，WDT 的控制状态也所有区别，具体参见下表

工作模式	SOFTWDT_EN	HWDT_EN	WDT 模块
测试模式	X	X	X
调试模式	0	X	关闭
	1	X	开启，且可以随着 CPU 的停止而停止
正常模式	X	X	开启
Sleep/Hold 模式	X	0	关闭
	X	1	开启

注：1) X 表示无意义

2) 工作模式的定义详见 6.1 工作模式

3) SOFTWDT\_EN 为 CLKCTRL1 寄存器的 bit14

4) HWDT\_EN 由 Flash 0xFC2 地址内 WDT\_EN[3:0] 决定，具体参见 2.4Flash 控制功能

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

### 12.3 特殊功能寄存器列表

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



## 12.4 特殊功能寄存器说明

WDTCLR (WDT 喂狗与时间配置寄存器)			基地址: 0x40010000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	0	0	0	0	0	0	0	0
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。

WDCNT (WDT 计数寄存器)			基地址: 0x40010000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	CNT[15:8]							
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:	CNT[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

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

## 13 定时器模块

### 13.1 定时器单元概述

HT6X2X 共有 6 路定时器，定时器分为两种类型，定时器 1、2、3、4 具有所有功能，使用时钟源为  $F_{sys}$ ；定时器 4、5 只有简单的定时和 PWM 功能，使用的时钟源为  $F_{pwm}$ 。

定时器 0、1、2、3 的时钟源即系统时钟  $F_{sys}$ ，可根据  $SYSCLK\_SEL[2:0]$  选择为：内部低频 RC 时钟 ( $F_{lrc}$ )，内部高频 RC 时钟 ( $F_{hrc}$ )，内部低频时钟 ( $F_{lf}$ ) 和内部高频时钟 ( $F_{p11}$ )。

定时器 4、5 的时钟源  $F_{pwm}$  独立于系统时钟  $F_{sys}$ ，可根据  $PWMCLK\_SEL[2:0]$  选择为：内部低频 RC 时钟 ( $F_{lrc}$ )，内部高频 RC 时钟 ( $F_{hrc}$ )，内部低频时钟 ( $F_{lf}$ )，和内部高频时钟 ( $F_{p11}$ )。

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

定时器主要包括以下功能：

- 定时功能
- PWM 功能
- 捕获功能
- 事件计数功能

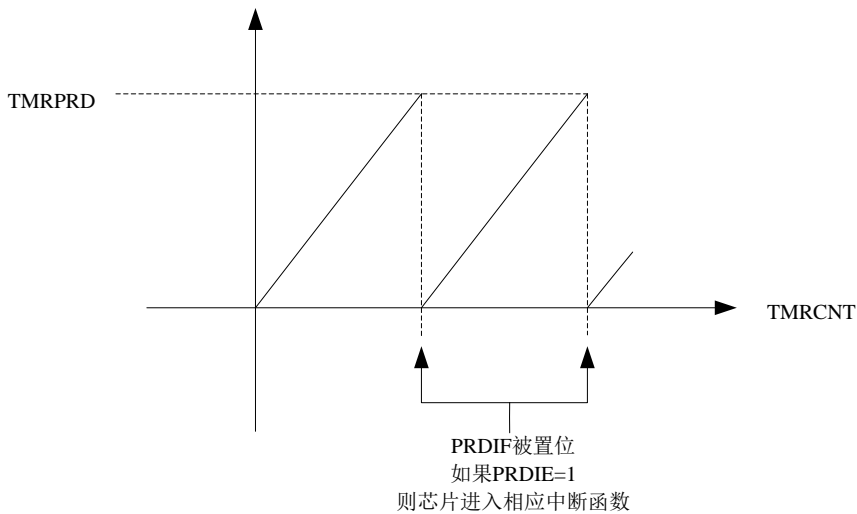
### 13.2 周期性定时功能

通用定时器的包括是一个 16 位计数器和周期寄存器。计数器的时钟由系统时钟 ( $F_{sys}$ ) 通过定时器单元内的一个预分频器 ( $TMRDIV$ ) 分频得到。当计数器寄存器  $TMRCNT$  等于周期寄存器 ( $TMRPRD$ ) 时会产生中断标志，如果使能相应的中断，则会触发定时器中断。

定时器的时钟源经选定后，可经预分频器  $TMRDIV$  对定时器时钟源进行分频，当使能计数器后定时器的计数器  $TMRCNT$  对预分频器后的时钟进行计数，当  $TMRCNT$  的值与周期寄存器  $TMRPRD$  的值相等时，产生周期中断标志，如果使能周期中断 ( $TMRIE.0$ )，则触发相应的周期中断函数。

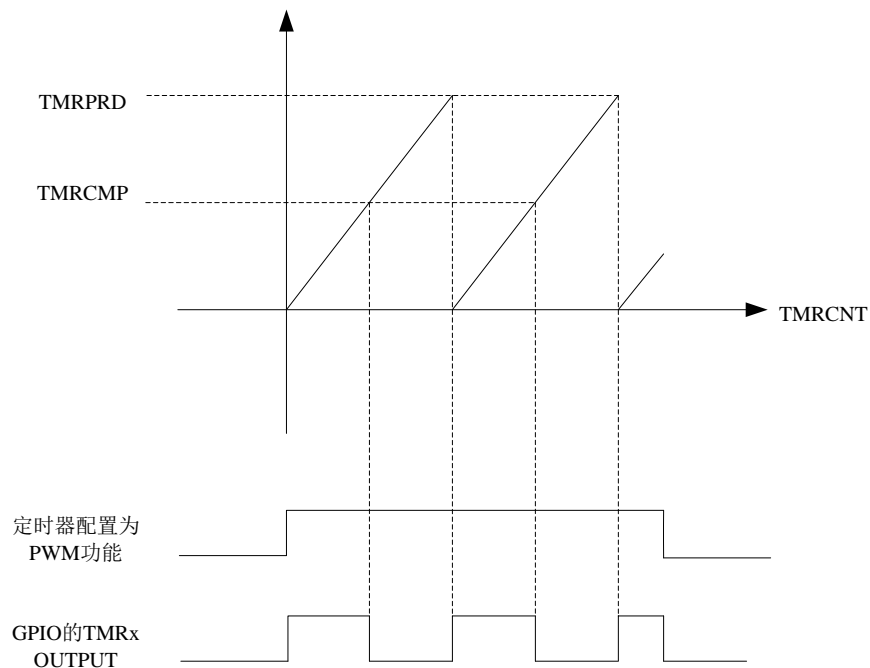
在周期中断标志产生后， $TMRCNT$  的值自动清零，然后重新开始计数。

功能主要相关寄存器： $TMRCNT$ ， $TMRPRD$ 。

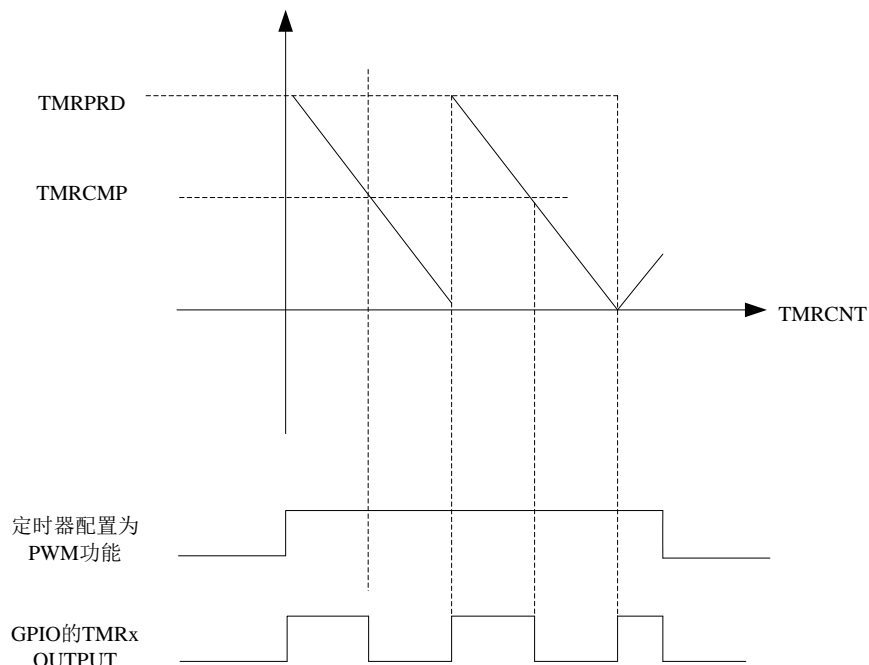


### 13.3 PWM 功能

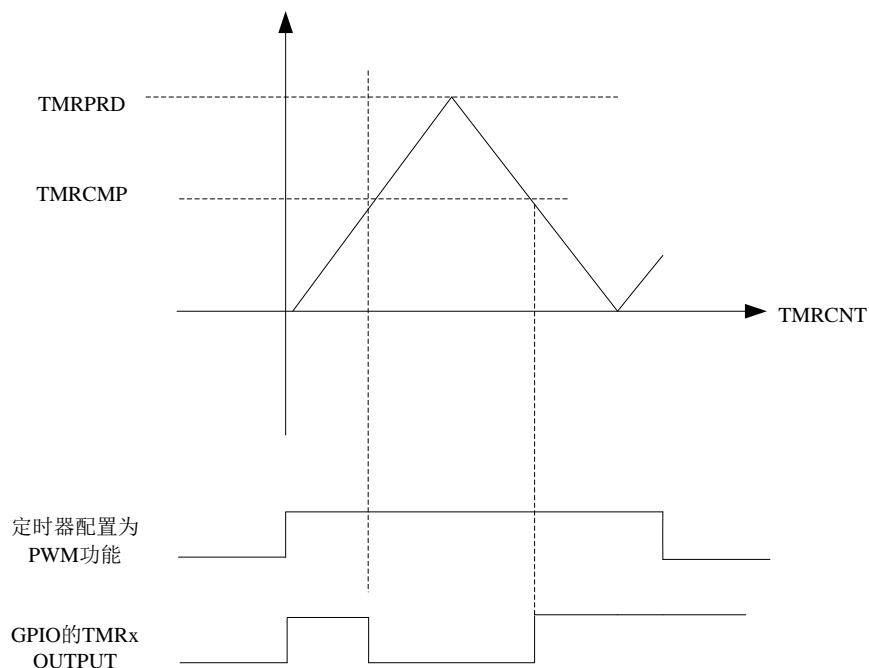
PWM 功能可通过寄存器 TMRCON.MODE[1..0]配置，同时需将对应的 GPIO 配置为 TMR 功能，配置成功后，相应的 GPIO 复用功能 TMR 引脚会输出波形。PWM 的周期和占空比可通过寄存器 TMRPRD, TMRCMP 进行配置。当使能了计数器 (CNTEN) 之后，计数器开始计数，当计数器 TMRcnt 的值等于比较寄存器 TMRCMP 的值时，PWM 输出管脚发生电平翻转，同时置位比较中断标志位。计数器继续计数，当计数器 TMRcnt 的值等于周期寄存器 TMRPRD 的值时，PWM 输出管脚再次发生电平翻转，同时置位周期中断标志位。PWM 输出波形如下图所示：



PWM 配置为向下计数方式时，当计数器 TMRCNT 从周期寄存器 TMRPRD 的值向下计数值等于比较寄存器 TMRCMP 的值时，PWM 输出管脚发生电平翻转。当计数器 TMRCNT 继续向下计数，计数值等于 0 时，PWM 输出管脚再次发生电平翻转。PWM 输出波形如下图所示：



PWM 配置为中央计数方式时，当计数器 TMRCNT 从周期寄存器 TMRPRD 的值向下计数值等于比较寄存器 TMRCMP 的值时，PWM 输出管脚发生电平翻转。当计数器 TMRCNT 继续向下计数，计数值等于 0 时，PWM 输出管脚再次发生电平翻转。PWM 输出波形如下图所示：



假如系统时钟选择为 PLL 产生的 22M 时钟，Timer 预分频寄存器 TMRDIV 默认值为 0 (不分频)，定时器 0 选择了 PWM 模式，向上计数方式，初始电平为高电平。若要想 PWM 占空比（高电平比上

周期)为30%。则  $TMRCMP/TMRPRD=30\%$ 。根据需要的周期值来确定 TMRPRD 寄存器的值。如果 Timer 预分频寄存器 TMRDIV 默认值为 0(不分频), TMRPRD 最大的周期时间为  $0xFFFF/(22M/(TMRDIV+1))=2.9789ms$

功能主要相关寄存器: TMRCNT, TMRCMP, TMRPRD。

## 13.4 捕获功能

### 13.4.1 输入捕获模式

在输入捕获模式下,假如设定上升沿检测,当 Timer0~Timer3 管脚上检测到上升沿,计数器的当前值被锁定到捕获寄存器中。当捕获事件发生时,相应的捕获中断标志 TMRIF.1 被置 1,如果使能中断 (TMRIF.1=1),将产生中断。

捕获过程中,如果没有检测到沿,在 TMRCNT 计数值和 TMRPRD 相等时会产生周期中断标志,如使能了周期中断会进入中断服务程序,之后 TMRCNT 从零开始计数。

捕获检测可以选择上升沿,下降沿方式。

功能主要相关寄存器: TMRCAP。

注: Timer4/5 不具有此功能

## 13.5 事件计数功能

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

事件计数过程中,如果没有检测到沿,在 TMRCNT 计数值和 TMRPRD 相等时会产生周期中断标志,如使能了周期中断会进入中断服务程序,之后 TMRCNT 继续计数直到 TMRCMP 值。单次最大计数个数为 0xffff,可以配合周期中断实现任意次数的组合。

事件计数检测可以选择上升沿,下降沿方式,该功能可用于 MCU 发脉冲。

功能主要相关寄存器: TMRCNT, TMRCMP。

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

注: Timer4/5 不具有此功能

## 13.6 中断功能

### 13.6.1 定时周期中断

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

### 13.6.2 捕获中断

当检测到外部输入信号相应沿时，如使能了捕获中断 TMRIE. CMPIE=1，则发生捕获中断。TMCNT 的值被锁定到 TMRCAP 中。

### 13.6.3 比较中断

当计数器的值 TMCNT 与比较寄存器的值 TMRCMP 相匹配时，如使能了 PWM 比较中断 TMRIE. CMPIE=1，则发生比较中断。

当计数器的值 TMCNT 与周期寄存器的值 TMRPRD 相等时，如使能了周期中断 TMRIE. PRDIE=1，也会发生周期中断。

### 13.6.4 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 TMRIE. ACIE=1，则发生事件计数中断。

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

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

事件计数检测可以选择上升沿，下降沿方式，该功能可用于 MCU 发脉冲。

功能主要相关寄存器：TMCNT，TMRCMP。

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

注：Timer4/5 不具有此功能

## 13.7 特殊功能寄存器列表

TMR 模块寄存器基地址:

0x40001000 (TMR0);

0x40002000 (TMR1);

0x40003000 (TMR2);

0x40004000 (TMR3);

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	控制寄存器
0x04	TMRDIV	R/W	0x0000	预分频寄存器
0x08	TMRPRD	R/W	0x0000	周期寄存器
0x0C	TMRCAP	R/*W	0x0000	捕获数据寄存器
0x10	TMRCNT	R/*W	0x0000	计数器寄存器
0x14	TMRCMP	R/W	0x0000	比较器寄存器
0x18	TMRIE	R/W	0x0000	中断使能寄存器
0x1C	TMRIF	R/W	0x0000	中断标志寄存器

## 13.8 特殊功能寄存器说明

TMRCON (定时器控制寄存器)			基地址: 0x40001000--0x40004000, 0x40019000, 0x4001A000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	PWMCLK_SEL[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	RESERVED	PWMHL	PWMC[1:0]		CCMODE	MODE[1:0]		CNTEN
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
---	------

PWMCLK_SEL[2:0]	<b>定时器 4、5 时钟选择控制位：</b> <table border="1" style="margin-left: 20px;"> <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>Flf</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Fhrc</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Fpll</td> </tr> </tbody> </table> 该时钟选择只对定时器 4、5 有用。	PWMCLK_SEL[2:0]			系统时钟选择Fsys	0	0	0	Flrc(Default)	0	0	1	Flf	0	1	0	Fhrc	0	1	1	Fpll
PWMCLK_SEL[2:0]			系统时钟选择Fsys																		
0	0	0	Flrc(Default)																		
0	0	1	Flf																		
0	1	0	Fhrc																		
0	1	1	Fpll																		
CMPMODE	<b>输出比较模式选择：</b> 0: 模式 0 1: 模式 1																				
PWMHL	<b>PWM 初始电平选择（当定时器被配置为 PWM 功能）：</b> 0: 高电平 1: 低电平																				
PWMC[1:0]	<b>PWM 工作模式选择（当定时器被配置为 PWM 功能）：</b> 00: 向上计数 01: 向下计数 1X: 中央对齐																				
CCMODE	<b>捕获/事件记录电平沿选择（当定时器被配置为捕获/事件记录功能）：</b> 0: 上升沿 1: 下降沿																				
MODE[1:0]	<b>Timer 功能选择：</b> 00: 事件记录 01: PWM 功能 10: 捕获功能 11: 周期定时功能																				
CNTEN	<b>计数器使能位：</b> 0: 关闭 1: 使能																				

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

位	功能描述
---	------



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

TMRPRD (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:	TMRPRD[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRPRD[7:0]							
Write:	TMRPRD[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMRPRD[15:0]	该寄存器是一个 16 的周期寄存器 计数的周期寄存器和 PWM 的周期寄存器都是该寄存器

TMRCAP (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:	TMRCAP[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRCAP[7:0]							
Write:	TMRCAP[7:0]							
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, 当前计数器的值被存到该寄存器里

TMRCNT (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:	TMRCNT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0

Read:	TMRCNT[7:0]							
Write:	TMRCNT[7:0]							
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)			基地址: 0x40001000--0x40004000, 0x40019000, 0x4001A000 偏移地址: 14H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRCMP[15:8]							
Write:	TMRCMP[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRCMP[7:0]							
Write:	TMRCMP[7:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMRCMP[15:0]	比较寄存器有两个功能: (1) Timer 做 PWM 功能的时候, 当计数器未达到 TMRPRD 周期设定之前, 而先达到 TMRCMP 的设定值的时候, TMRx 输出翻转。 (2) Timer 做事件计数功能的时候, 当设定了 TMRCMP 的值, TMRCNT 如果计数到与 TMRCMP 相同的时候, 会置位 TMRIF. 3(ACIF) 标志, 如果使能相应的中断 TMRIE. 3(ACIE), 则芯片会产生中断, 同时 TMRCNT 会从 0 开始重新计数。

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

位	功能描述
ACIE	事件计数中断使能 0: 关闭

	1: 使能
CMPIE	比较中断使能 0: 关闭 1: 使能
CAPIE	捕获中断使能 0: 关闭 1: 使能
PRDIE	定时周期中断使能 0: 关闭 1: 使能

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

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

## 14 SPI 模块

### 14.1 概述

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 和 PG. 9/SPI1\_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。包含下列特征：

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

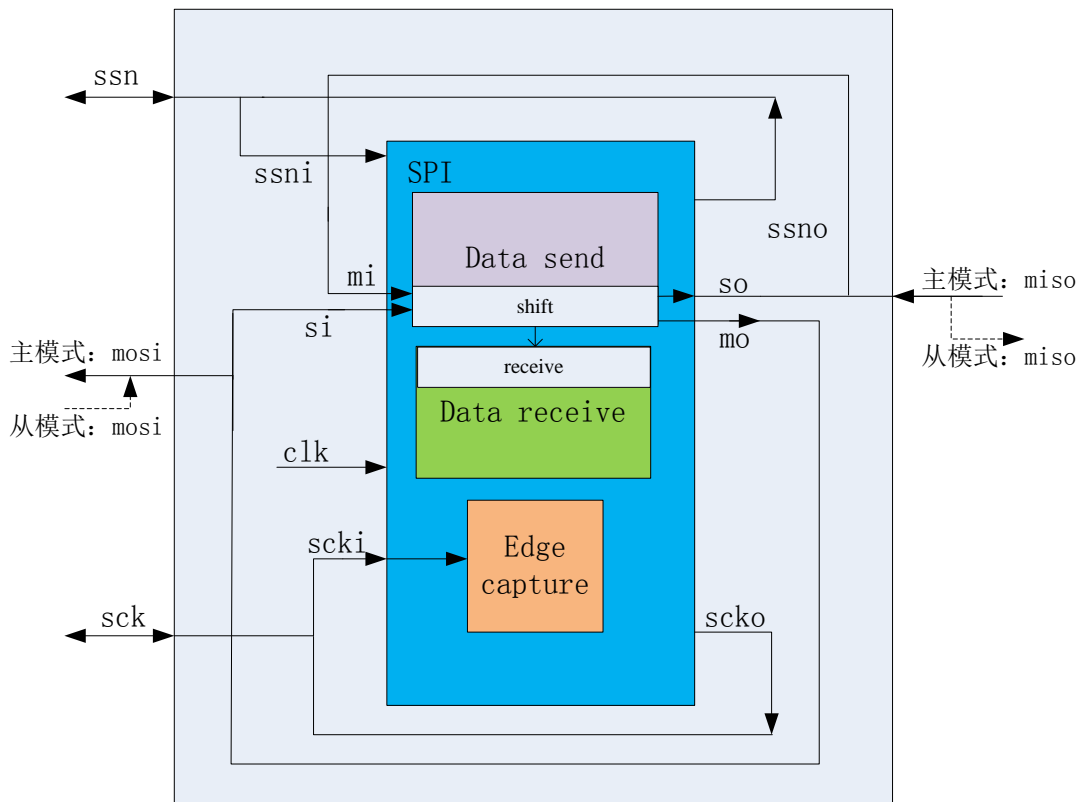
### 14.2 详细功能说明

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

#### 14.2.1 SPI 主要特征

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

## 14.2.2 SPI 模块框图



## 14.2.3 SPI 接口传输格式

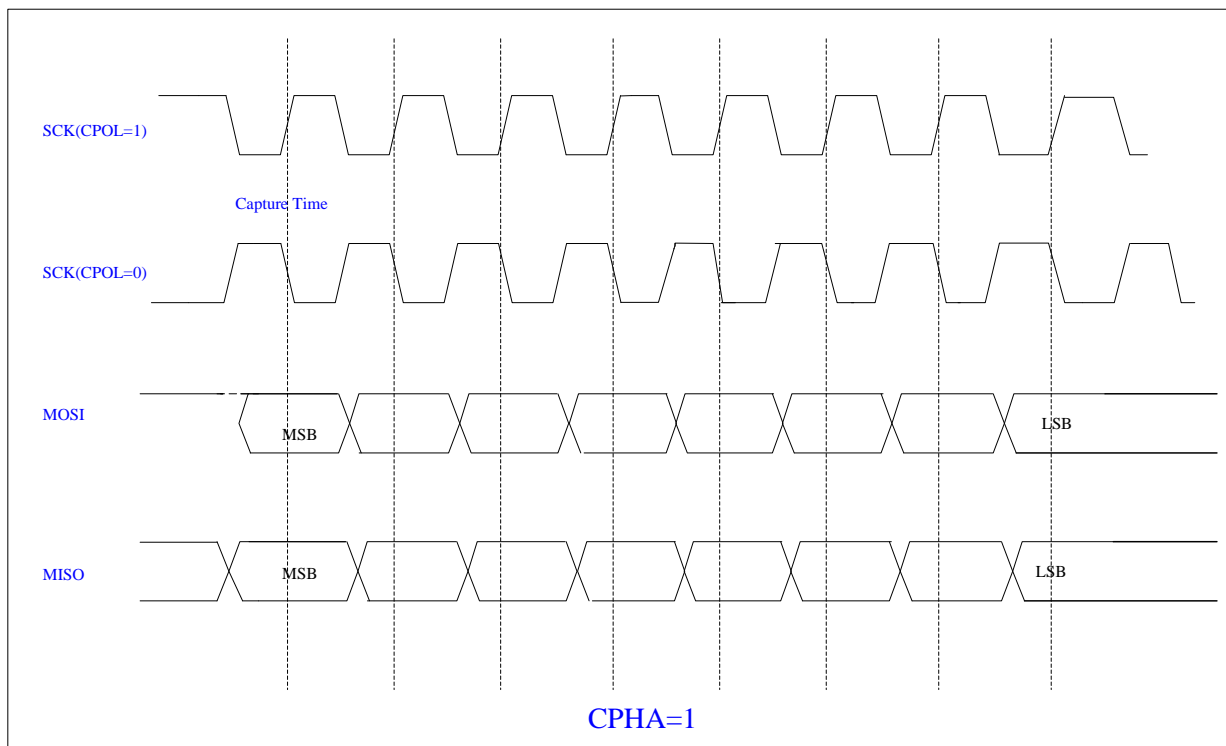
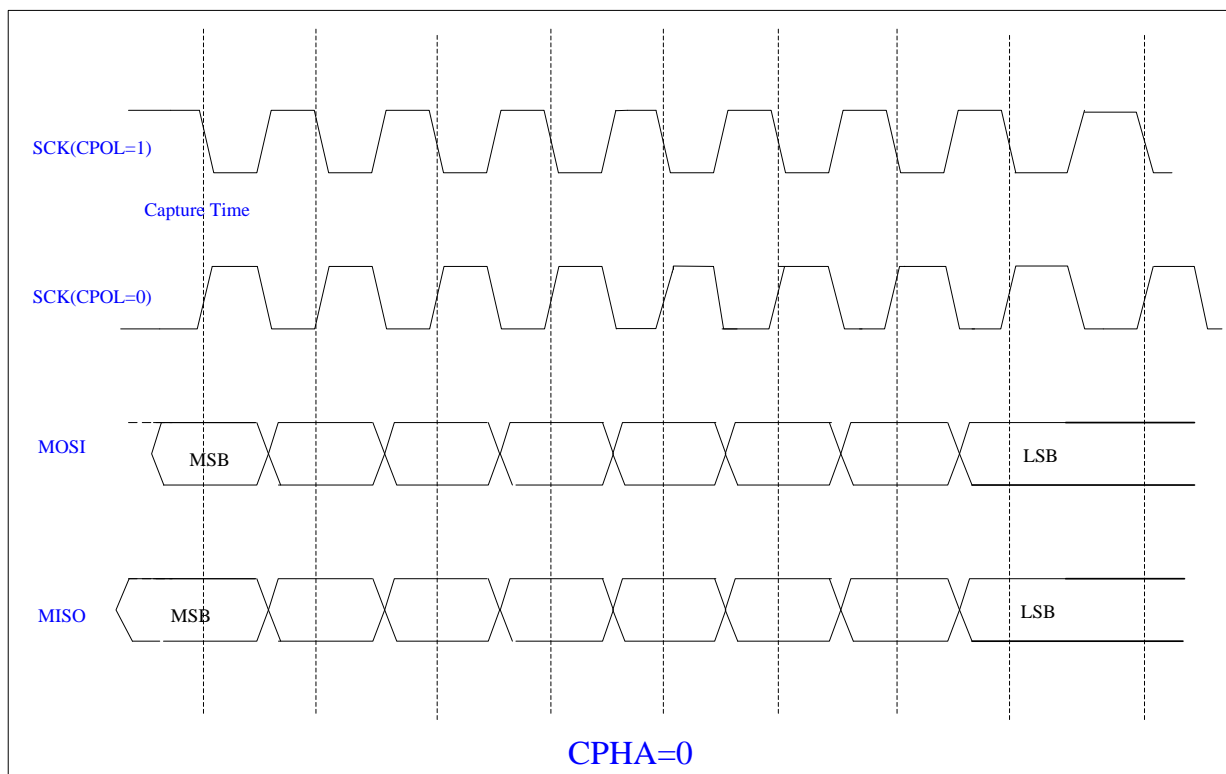
下图显示了数据传输的主要格式。根据 SPI 模块的设置，数据的每一位在主时钟（SCK）的上升沿（CPOL=0）或下降沿（CPOL=1）被传送。数据在主时钟（SCK）的下降沿（CPOL=1）或上升沿（CPOL=0）被接收。这适用于主模式或从模式的传输器/接收器，前提是 SCK 是传输过程中的主时钟。如果 CPHA 被置位，第一位（MSB）将在 SCK 的第一个动态沿时通过 MOSI/MISO 被发送。如果 CPHA 被清零，第一位（MSB）将在 SCK 的第一个动态沿之前半个周期被发送。

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

## 14.2.4 主机模式传输格式

SPI 默认为主机模式。

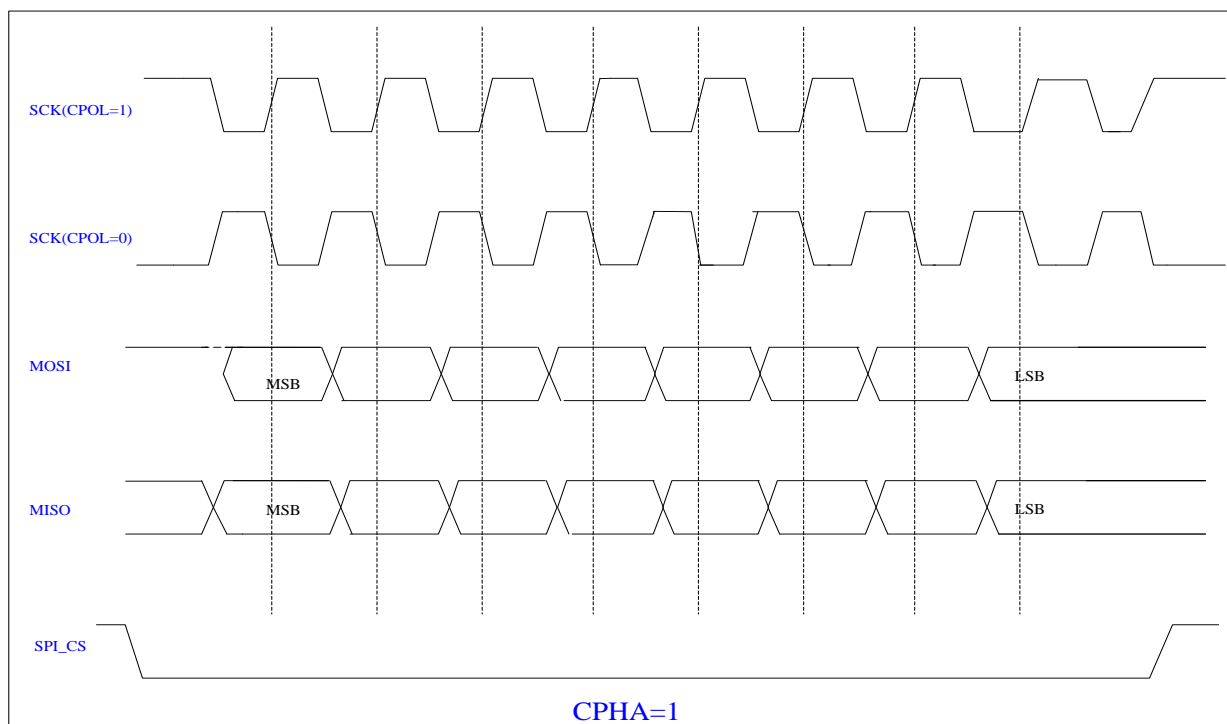
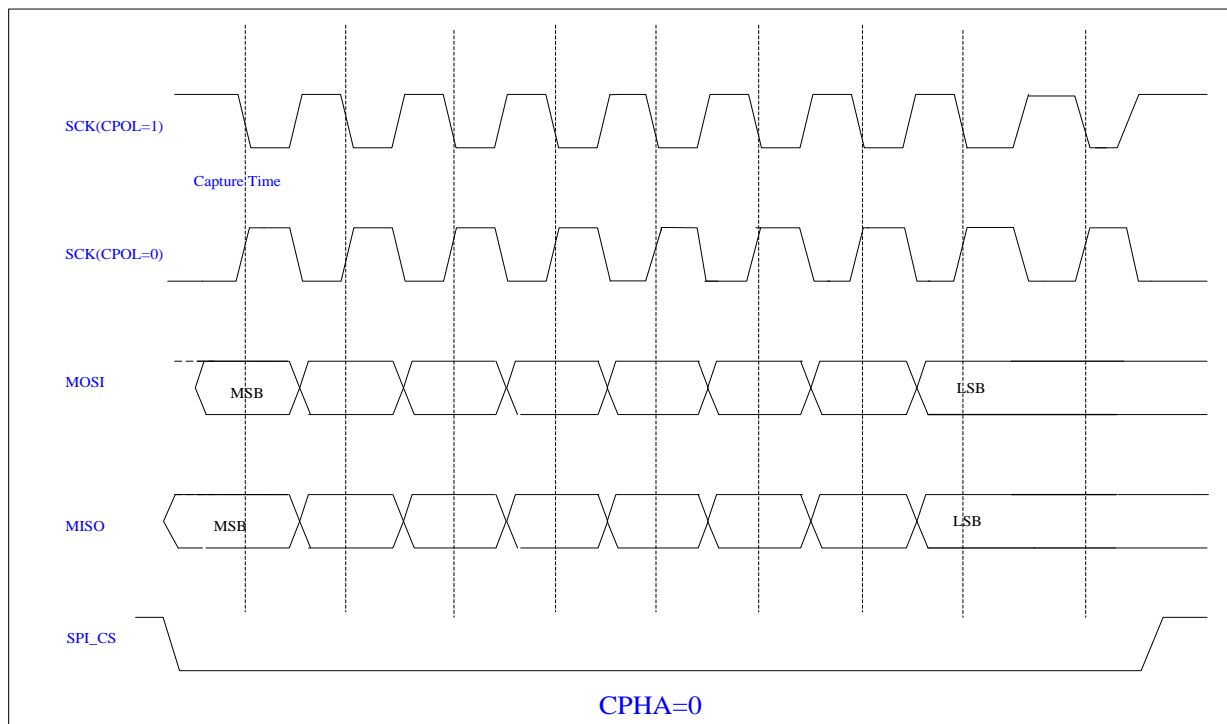
在主机模式中，SPI 等待程序向寄存器 SPIDAT 中写入数据。如果向 SPIDAT 的写入动作完成，传输就开始。在时钟 SCK 的发送沿，数据被移位到输出引脚 MOSI 上。同时，从从机传送过来的另一字节的数据被移位到主机的输入引脚 MISO 上。



主机模式数据传输格式

## 14.2.5 从机模式传输格式

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



从机模式数据传输格式

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

### 14.2.6 中断功能

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

## 14.3 特殊功能寄存器列表

SPI0 模块寄存器基地址：0x4000B000				
SPI1 模块寄存器基地址：0x40017000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	SPICON	R/W	0x0000	SPI 控制寄存器
0x04	SPISTA	R/W	0x0000	SPI 状态寄存器
0x08	SPIDAT	R/W	0x0000	SPI 数据寄存器
0x0C	SPISSN	R/W	0x00FF	SPI 从机选择寄存器

## 14.4 特殊功能寄存器说明

SPICON (控制寄存器)		基地址：0x4000B000, 0x40017000 偏移地址：00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SSDIS	SPR[2:0]			CPHA	CPOL	MSTR	SPI_EN
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SSDIS	SS控制位



	0: 在主/从模式中打开SPI_CS输入 1: 在主/从模式中关闭 SPI_CS 输入, 该情况下不会产生 MODF 中断请求; 在从模式中, 若 CPHA=0, 则该位无效																																				
SPR[2:0]	<b>SPI时钟速率控制位(主模式时用)</b> <table border="1" style="width: 100%; text-align: center;"> <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: 表示高位 (MSB) 将在SCK的第一个动态沿之前半个周期被发送 1: 表示高位 (MSB) 将在SCK的第一个动态沿通过被发送																																				
CPOL	<b>时钟极性</b> 0: SCK在空闲状态时被设置为低电平 1: SCK在空闲状态时被设置为高电平																																				
MSTR	<b>SPI模式选择位</b> 0: 从机模式 1: 主机模式																																				
SPI_EN	<b>SPI使能位</b> 0: 关闭SPI模块 1: 打开 SPI 模块																																				

SPISTA (状态寄存器)			基地址: 0x4000B000, 0x40017000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	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; 只能通过先读寄存器SPISTA, 然后读寄存器SPIDAT来清零;							
WCOL	<b>写冲突标志位</b>							

	当写SPIDAT冲突时由硬件置位；必须通过先读寄存器SPISTA，然后读寄存器SPIDAT来清零；
SSERR	<b>同步从机错误标志位</b> 在接收完成前当SPI_CS输入有效时，被硬件置位；关闭SPI模块可清除该位（设置SPI_EN=0）；
MODF	<b>模式故障标志位</b> 当SPI_CS引脚状态与设置的模式有冲突时，硬件自动置位；当SPI_CS引脚恢复合适的电平状态时，硬件自动复位；也可通过读寄存器SPISTA，然后读寄存器SPIDAT，然后该寄存器可被复位；

SPIDAT (数据寄存器)		基地址： 0x4000B000, 0x40017000 偏移地址： 08H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	CSB1	CSB0	CSA1	CSA0	ITDelay	ITDelay	ITDelay	ITDelay
Write:					3	2	1	0
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
CSB[1..0]	当读SPIDAT寄存器时，此2位无效（即读到的恒为0） 当写SPIDAT寄存器时，此2位含义如下： 00：发送数据前保持CS不变 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 注：此控制位只针对SPI1，SPI0无此控制位
CSA[1..0]	当读SPIDAT寄存器时，此2位无效（即读到的恒为0） 当写SPIDAT寄存器时，此2位含义如下： 00：发送数据后保持CS不变 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 注：此控制位只针对SPI1，SPI0无此控制位
ITDelay [3..0]	当读SPIDAT寄存器时，此4位无效（即读到的恒为0） 当写SPIDAT寄存器时，此4位含义如下： 发送数据完成后等待（Bit11-Bit8）*8 个SPICLK时间给出DMA请求 注：此控制位只针对SPI1，SPI0无此控制位

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)			基地址: 0x4000B000, 0x40017000 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	SSN0
Write:								
Reset:	1	1	1	1	1	1	1	1

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

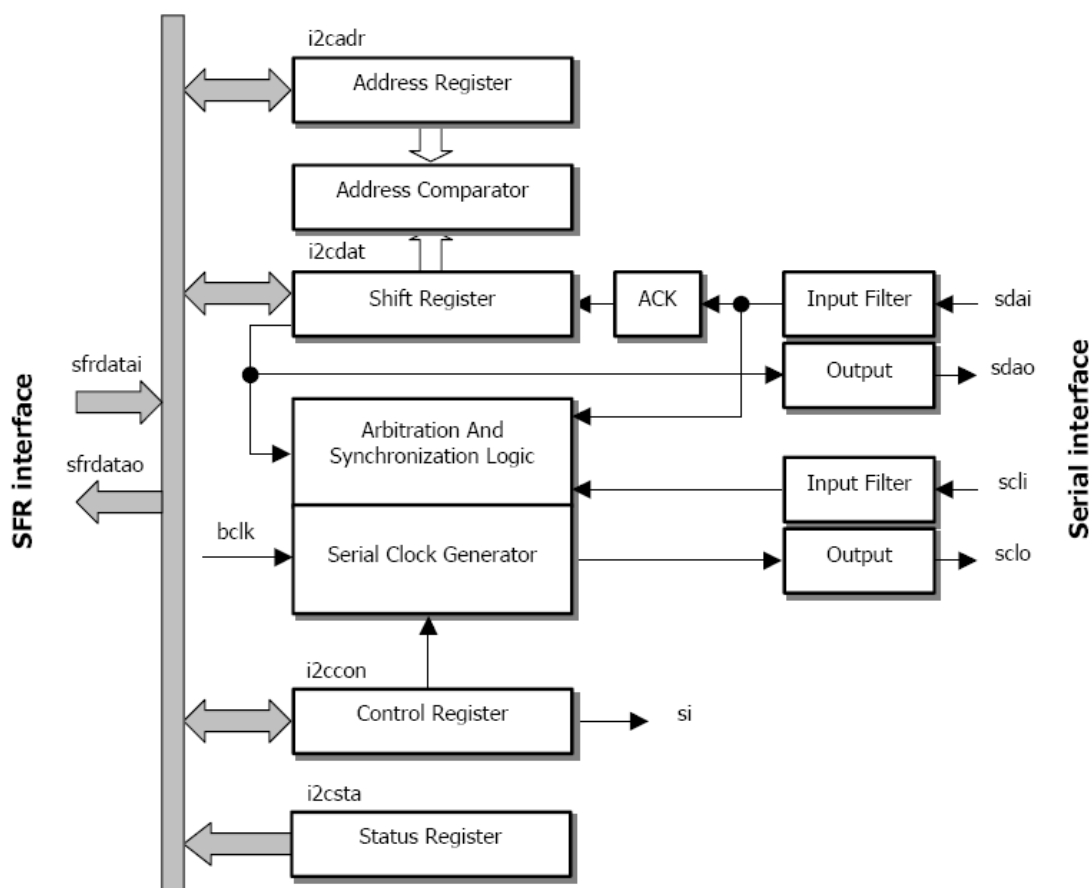
## 15 I2C 模块

### 15.1 概述

I2C 模块的芯片引脚为 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。

### 15.2 框图



I2C 模块功能框图

## 15.3 功能描述

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

### 15.3.1 操作模式

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

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

### 15.3.2 串行时钟生成

当 I2C 处于主机模式时，可编程的时钟发生器提供 SCL 时钟；当 I2C 处于从机模式时，时钟发生器被关闭，接收来自主机的时钟。时钟发生器的输出频率可以由寄存器 I2CCON 中的位 CR[9:0]控制，其中包含 I2CCON[0...1]，I2CCON[8...14]。

### 15.3.3 中断生成

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

IIC 产生中断时，寄存器 I2CCON 中的标志位 SI 会被置位。

### 15.3.4 传输模式

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

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

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

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

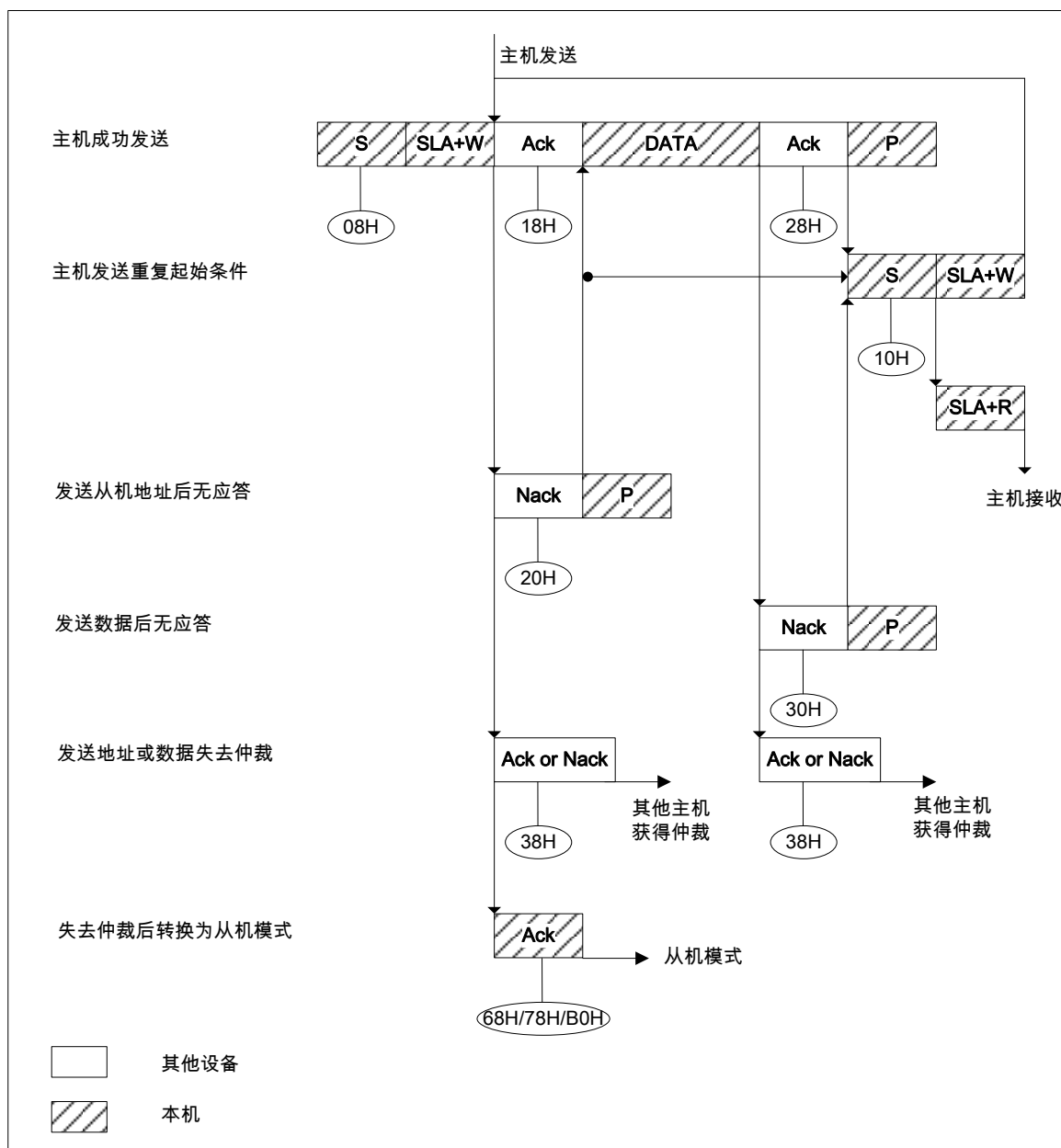
圆形用于表示中断标志已被置起。其中的数字表示当前状态寄存器 I2CSTA 中被掩去低三位的状态码。在 SI 被清除之前，I2C 通讯会暂停，应用软件必须决定是继续通讯还是终止当前传输。对每一个状态码，所需要的软件动作和随后的传输细节均有描述。

#### I2C 主机发送模式：

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

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

30H	i2cdat 的数据字节已被发送	数据字节或无动作或无动作或无动作	0	0	0	X	数据字节将被发送；ACK 将被接收 重复起始条件将被发送； 终止条件将被发送；sto 标志将被复位 起始条件被发送后将再发送一个终止条件；sto 标志将被复位
			1	0	0	X	
			0	1	0	X	
			1	1	0	X	
38H	SLA+R/W 或数据字节仲裁失败	无动作或无动作	0	0	0	X	I2C 总线将被释放；将进入“未寻址从机”状态； 当总线空闲时将发送一个起始条件
			1	0	0	X	

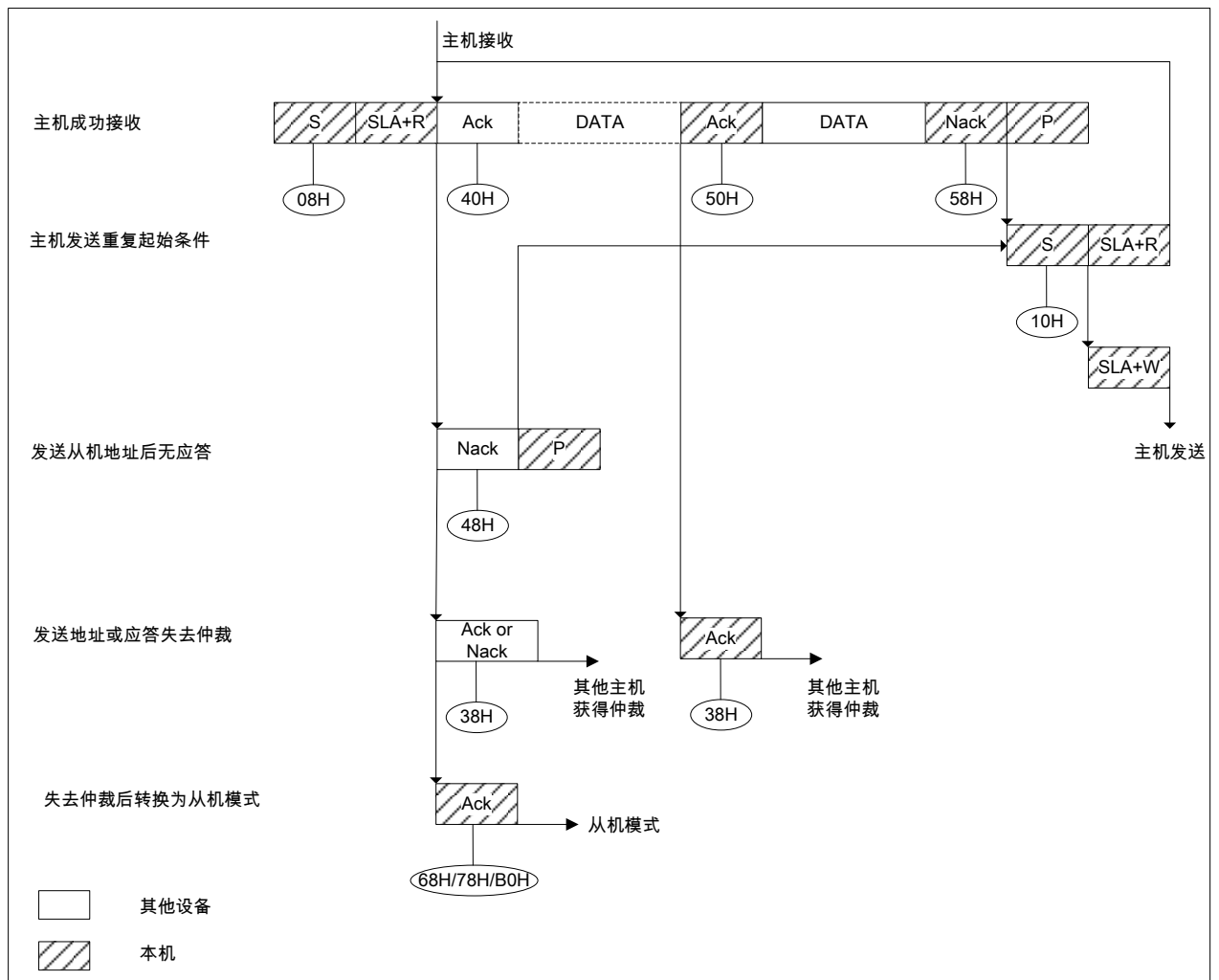


**I2C 主机接受模式：**

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

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





### I2C 从机接受模式:

从机接收模式中，从机从主机接收一系列数据。

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

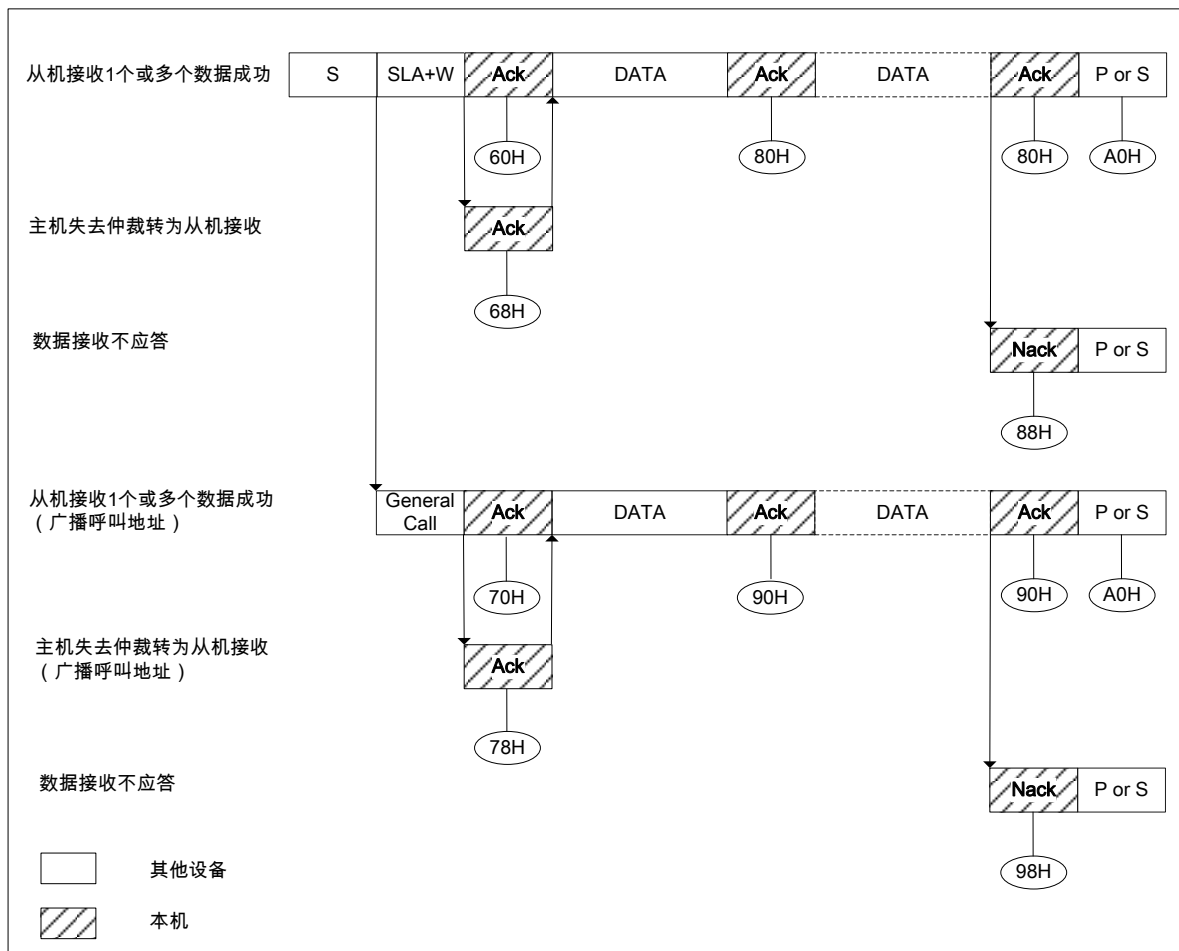
从机模式中，I2C 模块等待总线对本机地址或广播呼叫地址 (如果 I2CADR[0] 被置位) 的寻址。如果读写数据位是‘写’，则 I2C 进入从机接收模式，否则将进入从机发送模式。

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

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

	失败；自身的 SLA+W 已被接收，返回 ACK	或无动作	X	0	0	1	ACK” 数据字节将被接收并返回 ACK
70H	呼叫地址（00H）已被接收；已返回 ACK	无动作	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或无动作	X	0	0	1	数据字节将被接收并返回 ACK
78H	主机 SLA+R/W 仲裁失败；呼叫地址已被接收，返回 ACK	无动作	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或无动作	X	0	0	1	数据字节将被接收并返回 ACK
80H	预先写入自身 SLV 地址；DATA 字节已被接收；返回 ACK	读取数据字节	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或者 读取数据字节	X	0	0	1	数据字节将被接收并返回 ACK
88H	预先写入自身 SLA；DATA 字节已被接收；返回 “not ACK”	读取数据字节	0	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址
		或者 读取数据字节	0	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址
		或者 读取数据字节	1	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
		或者 读取数据字节	1	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
90H	预先写入呼叫地址；DATA 字节已被接收；返回 ACK	读取数据字节	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或者 读取数据字节	X	0	0	1	数据字节将被接收并返回 ACK
98H	预先写入呼叫地址；DATA 字节已被接收；返回 ACK	读取数据字节	0	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址
		或者 读取数据字节	0	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址
		或者 读取数据字节	1	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
		或者 读取数据字节	1	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
A0H	终止条件或重复起始条件在被配置为 SLV/REC 或 SLV/TRX 时被接收	无动作	0	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址
		或者 无动作	0	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址
		或者 无动作	1	0	0	0	切换为“未寻址从机”模式；不识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
		或者					

		无动作	1	0	0	1	切换为“未寻址从机”模式；识别自身从机地址或呼叫地址；当总线空闲时将发送一个起始条件
--	--	-----	---	---	---	---	--



### I2C 从机发送模式:

从机发送模式中，从机发送一系列数据到主机。

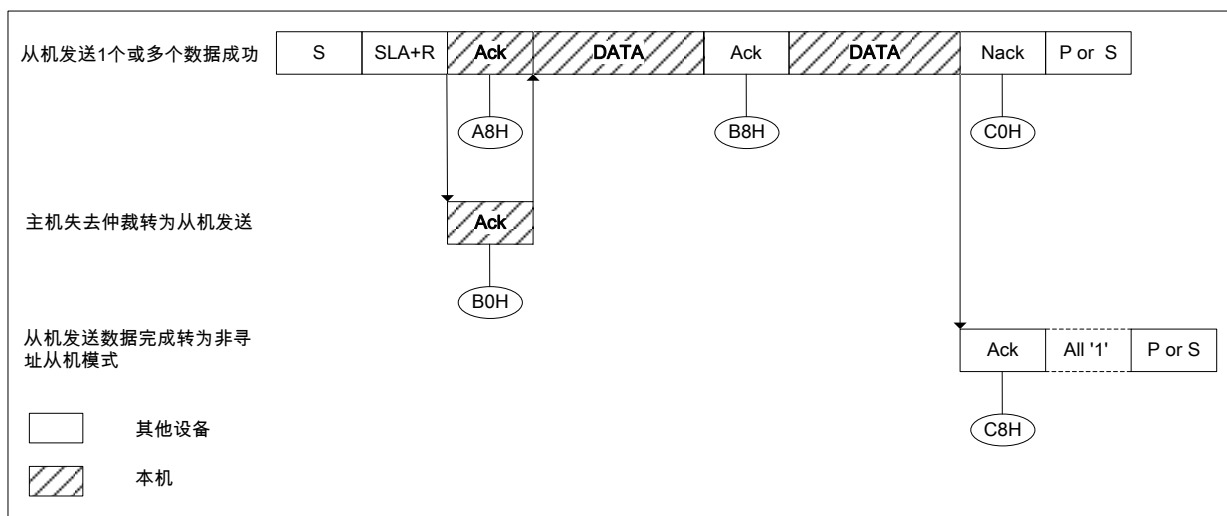
进入从机模式前，需设置从机地址，I2CADR 中 I2CADR[7..1] 位为从机地址。如果 I2CADR[0] 置位，从机也将响应广播呼叫地址 (00H)；否则将不响应广播呼叫地址。

从机模式中，I2C 模块等待总线对本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	加载数据字节	X	0	0	0	最后一个数据字节将被发送并接

	接收; 返回 ACK	或者 加载数据字节	X	0	0	1	收 ACK 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

## 15.4 特殊功能寄存器列表

微控制器与 I2C 组件的接口通过以下四个特殊功能寄存器来实现:

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

## 15.5 特殊功能寄存器说明

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

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

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

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

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

位	功能描述
CR[9:0]	I2C时钟频率控制位 $I2C\ CLOCK = F_{sys} / (CR[9:0] + 1) / 4$
ENS1	I2C 使能位 1: 使能IIC模块; 0: 关闭IIC模块;
STA	开始标志位 1: 检查IIC总线的状态，如果空闲则生成开始信号; 0: 不会生成开始信号;
STO	停止标志位

	1: 当处于主机模式, 则向总线传输停止信号 0: 不向总线传输停止信号;
SI	<b>中断标志位</b> 当进入25种IIC状态之一时, SI由硬件置位, 唯一不置位的状态是“F8H”; 写0清0, 写1无影响。
AA	<b>生成应答标志位</b> 1: 应答在以下情况下被返回: 接收到自身作为从机的地址; gc被置位的情况下接收到地址呼叫; 主机接收模式下一个字节接收完成; 从机接收模式下一个字节接收完成; 0: 非应答在以下情况下被返回: 主机接收模式下一个字节接收完成; 从机接收模式下一个字节接收完成;

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

## 16 RTC 模块

### 16.1 概述

RTC 单元提供实时时钟、日历功能，自动闰年调整，支持闹钟和周期性中断。  
RTC 模块在各种工作模式下都不会被关闭，在低功耗下仍然正常运行。  
RTC 输出寄存器、RTC 时钟校正寄存器不会被复位，以保持 RTC 的准确性。

### 16.2 功能描述

- 提供时钟和日历功能：输出寄存器包含秒、分、时、日、月、年和星期
- 具有自动闰年闰月调整功能
- 1 个 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

### 16.3 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  $DF_i$ ，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。 $DF_i$  计算公式如下：

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

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。

$DF_i$  每个 LSB 表示约 0.06ppm。

### 16.4 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。



## 16.5 时间和万年历

RTC 提供秒、分、时、日、月、年和星期输出寄存器。

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

## 16.6 中断功能

RTC 一共提供 8 种中断源，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 HOURR 加 1 时，产生一个小时中断，HRF 被置为 1。

对该位写 0 清标志。

**MINF**：分钟中断

分钟计数器 MINR 加 1 时，产生一个分钟中断，MINF 被置为 1。

对该位写 0 清标志。

**SECF**：秒中断

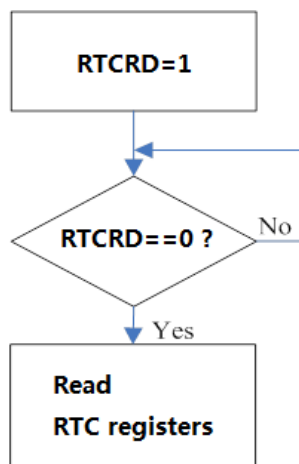
秒计数器 SECR 加 1 时，产生一个秒中断，SECF 被置为 1。

对该位写 0 清标志。

## 16.7 RTC 指示寄存器读写流程

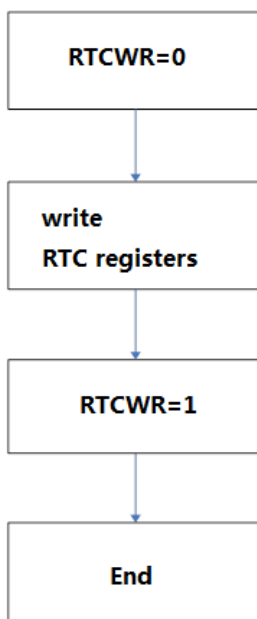
### 16.7.1 读取 RTC 指示寄存器流程

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



### 16.7.2 写入 RTC 指示寄存器流程

在用户需要更新 RTC 指示寄存器（SECR, MINR, HOURR, 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：



## 16.8 校时记录

校时次数增加以一次 RTCWR 置 1 为准，无论怎么更新 RTC 寄存器，只要 RTCWR 置 1，则 RTCNT 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 RTCWR 置 1，RTCNT 仍旧加 1。

## 16.9 第二套 RTC

系统提供第二套年/月/日/时/分/秒/周寄存器，由 LRC 驱动。这样可以在没有主时钟情况下，仍然可以获得较精确地计时时间。第二套 RTC 读写流程与第一套类似，只是其有单独的 RTCWR2 和 RTCRD2 寄存器。

## 16.10 特殊功能寄存器列表

RTC 模块寄存器基地址：0x4000C000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	RTCCON	R/W	0x0000	RTC 控制寄存器
0x04	RTCIE	R/W	0x0000	RTC 中断使能寄存器
0x08	RTCIF	R/W	0x0000	RTC 中断标志寄存器
0x0C	ALMR	R/W	0x0000	闹钟寄存器
0x10	RTCTMR1	R/W	0x0000	RTC 定时器 1 计数设置
0x14	RTCTMR2	R/W	0x0000	RTC 定时器 2 计数设置
0x18	SECR	R/W	0x0000	秒寄存器（写保护）
0x1C	MINR	R/W	0x0000	分寄存器（写保护）
0x20	HOURLR	R/W	0x0000	时寄存器（写保护）
0x24	DAYR	R/W	0x0001	日寄存器（写保护）
0x28	MONTHR	R/W	0x0001	月寄存器（写保护）
0x2C	YEARR	R/W	0x0000	年寄存器（写保护）
0x30	WEEKR	R/W	0x0001	周寄存器（写保护）
0x34	RTCCNTH	R/W	0x0000	校时次数寄存器高 16 位
0x38	RTCCNTL	R/W	0x0000	校时次数寄存器低 16 位
0x3C	RTCRD	R/W	0x0000	RTC 读控制寄存器
0x40	RTCWR	R/W	0x0000	RTC 写控制寄存器
0x50	DFAH	R/W	0x0000	RTC 硬件补偿常数项系数高位
0x54	DFAL	R/W	0x0000	RTC 硬件补偿常数项系数低位
0x58	DFBH	R/W	0x0000	RTC 硬件补偿一次项系数高位

0x5C	DFBL	R/W	0x0000	RTC 硬件补偿一次项系数低位
0x60	DFCH	R/W	0x0000	RTC 硬件补偿二次项系数高位
0x64	DFCL	R/W	0x0000	RTC 硬件补偿二次项系数低位
0x68	DFDH	R/W	0x0000	RTC 硬件补偿三次项系数高位
0x6C	DFDL	R/W	0x0000	RTC 硬件补偿三次项系数低位
0x70	DFEH	R/W	0x0000	RTC 硬件补偿四次项系数高位
0x74	DFEL	R/W	0x0000	RTC 硬件补偿四次项系数低位
0x78	Toff	R/W	0x0000	温度偏置寄存器
0x7C	MCON01	R/W	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 7 位
0x8C	DFiL	R/W	0x0000	频率误差寄存器低 16 位
0x90	RTCSTFLAG	R/W	0x0002	RTC 模块复位标志寄存器
0x94	RTCSTSET	W	0x0000	RTC 写复位寄存器
0x98	RTCChecksum	R/W	0x6A10	RTC 校验和寄存器
0x200	SECR2	R/W	0x0000	秒寄存器（写保护）
0x204	MINR2	R/W	0x0000	分钟寄存器（写保护）
0x208	HOURR2	R/W	0x0000	小时寄存器（写保护）
0x20C	DAYR2	R/W	0x0001	天寄存器（写保护）
0x210	MONTHR2	R/W	0x0001	月寄存器（写保护）
0x214	YRR2	R/W	0x0000	年寄存器（写保护）
0x218	WEEKR2	R/W	0x0001	星期寄存器（写保护）
0x21C	RTC2CAL	R/W	0x8000	第二套 RTC 计时调整寄存器
0x220	RTCRD2	R/W	0x0000	第二套 RTC 读控制寄存器
0x224	RTCWR2	R/W	0x0000	第二套 RTC 写控制寄存器

## 16.11 特殊功能寄存器说明

RTCCON (RTC 控制寄存器)			基地址: 0x4000C000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	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	RTC2EN	RTC1EN	TOUT[3:0]				AutoC
Write:	X	RTC2EN	RTC1EN	TOUT[3:0]				AutoC
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 DFiH/L 写无效 AutoC=1: RTC 手动补偿使能, 此时误差值由用户写入 DFiH/L (用户应先写 DFiH 再写 DFIL)。

TOUT[3:0]				TOUT	TOUT (PLL 使能)
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	32768Hz	32768Hz
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	

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

位	功能描述
ALMIE	RTC 闹铃中断使能位 0: 关闭 1: 打开
RTC2IE	RTC 定时器 2 中断使能位 0: 关闭 1: 打开
RTC1IE	RTC 定时器 1 中断使能位 0: 关闭 1: 打开
MTHIE	RTC 月中断使能位 0: 关闭 1: 打开
DAYIE	RTC 日中断使能位 0: 关闭 1: 打开
HRIE	RTC 小时中断使能位 0: 关闭 1: 打开
MINIE	RTC 分钟中断使能位 0: 关闭 1: 打开
SECIE	RTC 秒中断使能位 0: 关闭 1: 打开

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

位	功能描述
ALMIF	RTC 闹铃中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0

RTC2IF	RTC 定时器 2 中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
RTC1IF	RTC 定时器 1 中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
MTHIF	RTC 月中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
DAYIF	RTC 日中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
HRIF	RTC 小时中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
MINIF	RTC 分钟中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
SECIF	RTC 秒中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0

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

位	功能描述
ALMH[5:0]	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[6:0]	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。 说明: 当小时和分钟寄存器与闹钟寄存器中的值相匹配时 (且秒寄存器为 0), 才会产生中断。

RTCTMR1 (RTC 定时器 1 寄存器)			基地址: 0x4000C000 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	CNT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CNT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

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

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

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



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

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

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

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

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

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

DAYR (RTC 日寄存器)			基地址: 0x4000C000 偏移地址: 24H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	DAY[4:0]				
Write:								
Reset:	0	0	0	0	0	0	0	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。

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

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

位	功能描述
YEAR[6:0]	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)		基地址： 0x4000C000 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	X	X	WEEK[2:0]		
Write:								
Reset:	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。

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

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

位	功能描述
---	------

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

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

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

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

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

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

Reset:	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

注：Toff 仅用于 RTC 补偿的内部计算，对 TMPDAT 转换的显示温度值造成的误差较小，可忽略不计。详见 FAQ 描述。

MCON01 (RTC 校准控制寄存器)			基地址： 0x4000C000 偏移地址： 7CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	1	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
BIT15~BIT0	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)			基地址： 0x4000C000 偏移地址： 80H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	1	0	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	UPDATE
Write:								
Reset:	1	0	0	0	1	0	0	0

位	功能描述
BIT15~BIT0	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)			基地址： 0x4000C000 偏移地址： 84H					
	Bit15	14	13	12	11	10	9	Bit8

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

位	功能描述
BIT15~BIT0	RTC 校准控制寄存器。 请确保此寄存器值为 0x4488

DFiH (常数项高 7 位)		基地址: 0x4000C000 偏移地址: 88H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	DFx[22:16]						
Write:								
Reset:	0	0	0	0	0	0	0	0

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

位	功能描述
DFx[22:0]	23 位寄存器, 最高位为符号位。 当 AutoC 为 0 时, 此时用户对 DFiH/ DFiL 写无效。 当 AutoC 为 1 时, 手动补偿方式, 补偿值由用户写入 DFiH/ DFiL 寄存器。用户必须先写高位, 再写低位(即按照 DFiH—DFiL 顺序写寄存器)。 DFx 每一个 LSB 表示 1/512Hz(时钟源是 32768), 即约 0.06ppm/LSB

RTCSTFLAG (RTC 模块复位标志寄存器)			基地址: 0x4000C000 偏移地址: 90H					
	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	PORRST	SoftRST
Write:								
Reset:	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 与 TBS 模块发生软件复位的复位标志: 0: RTC 与 TBS 模块未发生软复位 1: RTC 与 TBS 模块发生了软复位 (写 0 清 0)

RTCSTSET (RTC 写复位寄存器)			基地址: 0x4000C000 偏移地址: 94H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Bit[15:0]	当向此寄存器写入 0xAAAA, 再写入 0x5555 时, RTC 与 TBS 模块发生复位, 发生复位后, RTCSTFLAG 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, 其他所有 RTC 寄存器和 TBS 寄存器都会复位。

RTCChecksum (RTC 校验和寄存器)			基地址: 0x4000C000 偏移地址: 98H					
	Bit15	14	13	12	11	10	9	Bit8

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

位	功能描述
Bit[15..0]	RTCChecksum = DFAH + DFAL + DFBH + DFBL + DFCH + DFCL + DFDH + DFDL + DFEH + DFEL + Toff + MCON01 + MCON23 + MCON45 注：所有被累积寄存器都当成 16 位无符号数

SECR2 (第二套 RTC 秒寄存器)			基地址: 0x4000C000 偏移地址: 200H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	SEC[5:0]					
Write:								
Reset:	0	0	0	0	0	0	0	0

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

MINR2 (第二套 RTC 分寄存器)			基地址: 0x4000C000 偏移地址: 204H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	MIN[5:0]					
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述



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

<b>HOURR2</b> (第二套 RTC 时寄存器)			<b>基地址:</b> 0x4000C000 <b>偏移地址:</b> 208H					
	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	HOUR[4:0]				
Write:								
Reset:	0	0	0	0	0	0	0	0

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

<b>DAYR2</b> (第二套 RTC 日寄存器)			<b>基地址:</b> 0x4000C000 <b>偏移地址:</b> 20CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	DAY[4:0]				
Write:								
Reset:	0	0	0	0	0	0	0	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。

<b>MONTHR2</b> (第二套 RTC 月寄存器)			<b>基地址:</b> 0x4000C000 <b>偏移地址:</b> 210H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0

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

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

YEARR2 (第二套 RTC 年寄存器)		基地址： 0x4000C000 偏移地址： 214H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	YEAR[6:0]						
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	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。

WEEKR2 (第二套 RTC 周寄存器)		基地址： 0x4000C000 偏移地址： 218H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	WEEK[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	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。

RT2CAL	基地址： 0x4000C000
--------	-----------------

(第二套 RTC 计时调整寄存器)		偏移地址: 21CH						
	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:	1	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	X
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Bit[16..0]	17 位无符号数, LRC 计时 1s 产生时间为: (RTC2CAL)/FLrc (FLrc 为低频 RC 的频率) 复位后默认值为 0x8000

RTC RD2 (第二套 RTC 读控制寄存器)		基地址: 0x4000C000 偏移地址: 220H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	READFLAG
Write:	X	X	X	X	X	X	X	G
Reset:	0	0	0	0	0	0	0	0

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

RTC WR2 (第二套 RTC 写控制寄存器)		基地址: 0x4000C000 偏移地址: 224H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	UPDATE

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

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

## 17 TBS 模块

### 17.1 概述

芯片可以对温度和 7 路 ADC (VCC, ADCBAT, ADCIN0, ADCIN1, ADCIN3, ADCIN4, ADCIN5) 进行定量的测量, 并将测量的结果保存在相应的寄存器中。用户可以根据测量的结果将当前的温度和电池电量在 LCD 上显示或者对 RTC 等模块进行补偿。

### 17.2 功能描述

TBS 主要包括两部分功能:

- 测量 IC 基底的温度
- 测试 7 路 ADC 电压 (VCC, ADCBAT, ADCIN0, ADCIN1, ADCIN3, ADCIN4, ADCIN5)

ADCBAT 检测内部有分压电阻 40K。ADCIN0、ADCIN1、ADCIN3、ADCIN4、ADCIN5 内部没有分压电阻。

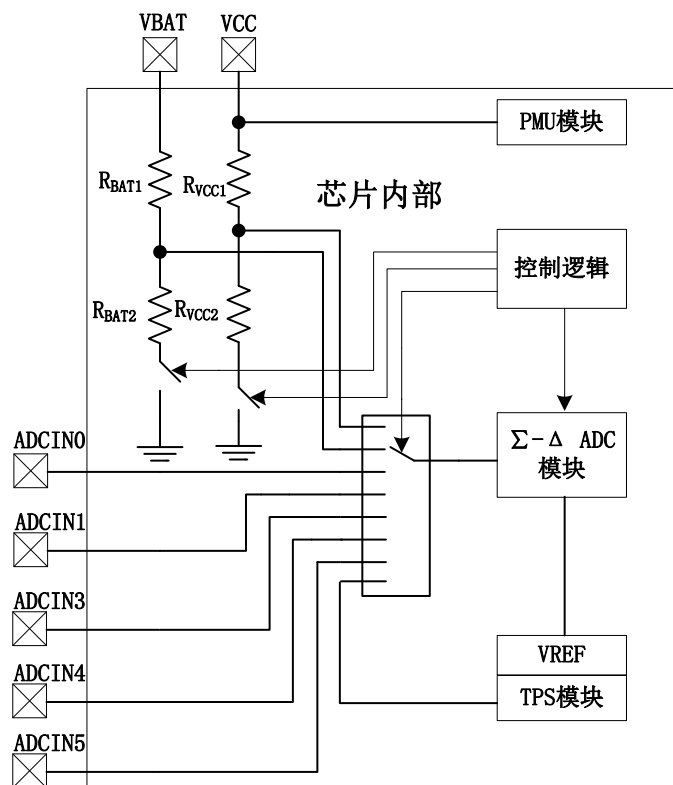


图 17.1 HT602X TBS 模块框图

Chip Num	ADCBAT 通道				VCC 通道			
	R <sub>BAT1</sub>	R <sub>BAT2</sub>	R <sub>BAT1</sub> :R <sub>BAT2</sub>	R <sub>BAT1</sub> + R <sub>BAT2</sub>	R <sub>VCC1</sub>	R <sub>VCC2</sub>	R <sub>VCC1</sub> : R <sub>VCC2</sub>	R <sub>VCC1</sub> + R <sub>VCC2</sub>
HT6X2X	37.5K	6.25K	6:1	43.75K	37.5K	6.25K	6:1	43.75K

ADCINX 输入阻抗: 30M

为保证环境温度与外接低频 32K 时钟晶振的温度保持一致，外接晶体应尽量靠近芯片管脚，并在保持环境温度稳定的情况下进行温度传感器的初值校正。温度传感器 TPS 初始误差出厂会进行测试，并存放在 Info Block 区域；同时，Info Block 提供的 RTC 补偿系数，是在全温度范围内由芯片内部 TPS 传感器读取值与晶体时钟对应数据拟合而成，已将温度差异因素考虑进去。

## 17.3 特殊功能寄存器列表

TBS 模块寄存器基地址: 0x4000E000				
偏移地址	名称	读写方式	复位值	功能描述
00H	TBSCON	R/W	0x6541	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	ADCBATDAT	R	0x0000	电池电压测量输出寄存器
14H	ADCODAT	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	VDRCMP	R/W	0x0000	电池电压比较寄存器
20H	TBSPRD	R/W	0x0000	TBS打开频率设置寄存器
24H	RESERVED	R/W	0x0200	该寄存器值用户不要修改
28H	VCCDAT	R	0x0000	电源电压测量输出值
2CH	ADCOCMP	R/W	0x0000	ADC通道0比较寄存器
30H	ADC3DAT	R	0x0000	ADC通道3测量输出值
34H	ADC4DAT	R	0x0000	ADC通道4测量输出值
38H	ADC5DAT	R	0x0000	ADC通道5测量输出值

## 17.4 特殊功能寄存器说明

TBSCON (TBS 设置寄存器)			基地址: 0x4000E000 偏移地址: 00H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	X	X	X	ADC5En	ADC4En
Write:								

Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADC3En	RESERVE	RESERVE	RESERVE	RESERVED		Filter[1:0]	
Write:		D	D	D				
Reset:	0	1	1	0	0	1	0	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	RESERVED		RESERVE	VCCEn	ADC1En	ADC0En	ADCBATE	TMPEEn
Write:			D				n	
Reset:	0	1	0	0	0	0	0	1

位	功能描述
ADC5En	<b>ADC 通道 5 测量功能控制位:</b> 0: 关闭 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 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 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
BIT[14: 10]	<b>RESERVED</b> 请用户务必保持默认 BIT 值配置: 11001
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
BIT[7:5]	<b>RESERVED</b> 请用户务必保持默认 BIT 值配置: 010
VCCEn	<b>VCC 电压测量功能控制位</b> 0: 关闭 1: 使能
ADC1En	<b>ADC 通道 1 测量功能控制位:</b> 0: 关闭 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 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
ADCBATEn	<b>电池电压测量功能控制位：</b> 0：关闭 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)		基地址： 0x4000E000 偏移地址： 04H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	ADC5IE	ADC4IE
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADC3IE	ADCOCMP IE	VCCIE	ADCBAT CMP IE	ADC1IE	ADC0IE	ADCBATI E	TMPIE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADC5IE	<b>ADC 通道 5 测量中断使能位</b> 0：关闭 1：打开
ADC4IE	<b>ADC 通道 4 测量中断使能位</b> 0：关闭 1：打开
ADC3IE	<b>ADC 通道 3 测量中断使能位</b> 0：关闭 1：打开
ADCOCMPIE	<b>ADC0 比较中断使能位</b> 0：关闭 1：打开 注：当 ADCODAT 大于 ADCOCMP 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 1：打开
ADCBATCMP IE	<b>ADCBATCMP：ADCBATDAT 小于 VDRCMP 时产生中断</b> ADCBATCMP IE = 0：中断关闭 ADCBATCMP IE = 1：中断打开 当 ADCBATDAT 低于 VDRCMP 时将产生中断，此功能只在使能了电池电压测量



	(ADCBATEn) 才有效
ADC1IE	ADC 通道 1 测量中断使能位 0: 关闭 1: 打开
ADC0IE	ADC 通道 0 测量中断使能位 0: 关闭 1: 打开
ADCBATIE	电池电压测量中断使能位 0: 关闭 1: 打开
TMPIE	温度测量中断使能位 0: 关闭 1: 打开

TBSIF (TBS 中断标志寄存器)			基地址: 0x4000E000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	ADC5IF	ADC4IF
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADC3IF	ADCOCMP IF	VCCIF	ADCBAT CMP IF	ADC1IF	ADC0IF	ADCBATI F	TMPIF
Write:								
Reset:	0	0	0	0	0	0	0	1

位	功能描述
ADC5IF	ADC 通道 5 测量中断标志位 0: 未产生中断 1: 产生中断
ADC4IF	ADC 通道 4 测量中断标志位 0: 未产生中断 1: 产生中断
ADC3IF	ADC 通道 3 测量中断标志位 0: 未产生中断 1: 产生中断
ADCOCMPIF	ADC0 比较中断标志位 0: 未产生中断 1: 产生中断 注: 当 ADCODAT 大于 ADCOCMP 时将产生标志
VCCIF	电源电压测量中断标志位 0: 未产生中断

	1: 产生中断
ADCBATCMPIF	<b>ADCBATCMPIF:</b> ADCBATCMPIF = 0: 未产生中断 ADCBATCMPIF = 1: 产生中断 <b>注: ADCBATDAT 小于 VDRCMP 时产生的中断标志</b>
ADC1IF	<b>ADC 通道 1 测量中断标志位</b> 0: 未产生中断 1: 产生中断
ADCOIF	<b>ADC 通道 0 测量中断标志位</b> 0: 未产生中断 1: 产生中断
ADCBATIF	<b>电池电压测量中断标志位</b> 0: 未产生中断 1: 产生中断
TMPIF	<b>温度测量中断标志位</b> 0: 未产生中断 1: 产生中断

<b>TMPDAT</b> (温度测量输出寄存器)			<b>基地址: 0x4000E000</b> <b>偏移地址: 0CH</b>					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	1	1	1	0	1	1	1	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	1	1	0	1	0	1	0	0

位	功能描述
DAT[15:0]	温度测量输出寄存器, 16 位有符号数

温度计算公式:

温度  $T_r = 12.9852 - TMPDAT * 0.0028$

其中:  $T_r$  为实际的温度 (°C)

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

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

电池电压计算公式：

$$\text{ADCBAT} = 0.1793 * \text{ADCBATDAT} + 1.3260;$$

其中：ADCBAT 为实际电池电压（mV）

ADC0DAT (ADC 通道 0 测量输出寄存器)	基地址：0x4000E000 偏移地址：14H
	Bit15...Bit0
Read:	DAT[15:0]
Write:	X X X X X X X X X
Reset:	0 0 0 0 0 0 0 0 0

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

ADC 通道 0 测量电压计算公式：

$$\text{VADCIN0} = 0.0258 * \text{ADC0DAT} + 4.7559;$$

其中：VADCIN0 为实际 ADC 测量电压（mV）

ADC1DAT (ADC 通道 1 测量输出寄存器)	基地址：0x4000E000 偏移地址：18H
	Bit15...Bit0
Read:	DAT[15:0]
Write:	X X X X X X X X X
Reset:	0 0 0 0 0 0 0 0 0

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

ADC 通道 1 测量电压公式同 ADC 通道 0。

$$\text{VADCIN1} = 0.0258 * \text{ADC1DAT} + 4.7559;$$

其中：VADCIN1 为实际 ADC 测量电压（mV）

ADCBATCMP (ADCBAT 比较寄存器)	基地址：0x4000E000 偏移地址：1CH
	Bit15...Bit0
Read:	DAT[15:0]
Write:	
Reset:	0 0 0 0 0 0 0 0 0

位	功能描述
---	------

DAT[15:0]	ADCBAT 比较寄存器，16 位有符号数。 注：当 ADCBATDAT 低于 ADCBATCMP 时将产生中断，此功能只在使能了电池电压测量 (ADCBATEn) 时才有效
-----------	--

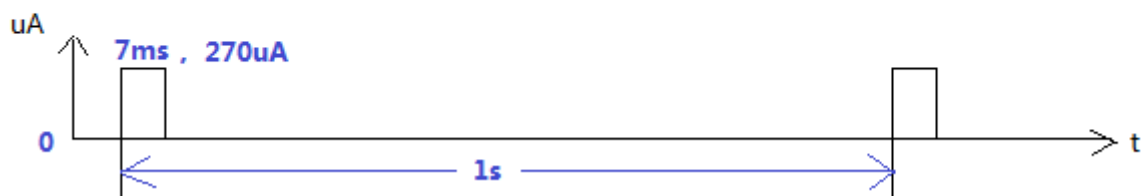
TBSPRD (TBS 测试寄存器)		基地址: 0x4000E000 偏移地址: 20H						
	Bit23	22	21	20	19	18	17	Bit16
Read:								ADC5PRD
Write:								[1]
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	ADC5PRD[	ADC4PRD[1:0]		ADC3PRD[1:0]		VCCPRD[1:0]		ADC1PRD
Write:	0]							[1]
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADC1PRD[	ADC0PRD[1:0]		ADCBATPRD[1:0]		TMPPRD[2:0]		
Write:	0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADC5PRD [1:0]	ADC 通道 5 测量周期设置: 00: 1 秒 01: 2 秒 10: 8 秒 11: 16 秒
ADC4PRD [1:0]	ADC 通道 4 测量周期设置: 00: 1 秒 01: 2 秒 10: 8 秒 11: 16 秒
ADC3PRD [1:0]	ADC 通道 3 测量周期设置: 00: 1 秒 01: 2 秒 10: 8 秒 11: 16 秒
VCCPRD [1:0]	电源电压测量周期设置: 00: 1 秒 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 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 01: 2 秒 10: 8 秒 11: 16 秒																		
ADCBATPRD[1:0]	<b>电池电压测量周期设置:</b> 00: 1 秒 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" data-bbox="470 967 1093 1393"> <thead> <tr> <th>TMPPRD [2:0]</th> <th>TBS (HT602X) 打开周期 (s)</th> </tr> </thead> <tbody> <tr><td>000</td><td>1/2</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 (HT602X) 打开周期 (s)	000	1/2	001	1	010	2	011	4	100	8	101	16	110	32	111	64
TMPPRD [2:0]	TBS (HT602X) 打开周期 (s)																		
000	1/2																		
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。  
 TPS 分时开启：每次开启的峰值功耗为 270uA，持续时间为 7ms，例如 1s 分时开启示意图如下：

**TPS 分时1s开启一次的 功耗&开启时间 示意图**



VCCDAT	基地址: 0x4000E000
--------	-----------------

(电源电压测量输出寄存器)	偏移地址: 28H							
	Bit15…Bit0							
Read:	DAT[15:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

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

VCC 电压计算公式:

$$VCC = 0.1798 * VCCDAT + 0.5943;$$

其中: Vcc 为实际 VCC 电压 (mV)

ADC0CMP (ADC0 比较寄存器)	基地址: 0x4000E000 偏移地址: 2CH							
	Bit15…Bit0							
Read:	DAT[15:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

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

ADC3DAT (ADC3 测量输出寄存器)	基地址: 0x4000E000 偏移地址: 30H							
	Bit15…Bit0							
Read:	DAT[15:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

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

ADC4DAT (ADC4 测量输出寄存器)	基地址: 0x4000E000 偏移地址: 34H							
	Bit15…Bit0							
Read:	DAT[15:0]							
Write:	X	X	X	X	X	X	X	X

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

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

ADC5DAT (ADC5 测量输出寄存器)		基地址: 0x4000E000 偏移地址: 38H						
		Bit15...Bit0						
Read:	DAT[15:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

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

## 18 AES/GHASH/RAND/CRC 模块

### 18.1 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 硬件加速器在 22M 的 PLL 时钟频率下的加密/解密时间小于 10us。详见应用笔记。

### 18.2 GHASH 概述

HT602x 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。

### 18.3 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，对于 HT6X2X 来说，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。

### 18.4 CRC 概述

CRC 模块含有可编程的多项式设置，支持常用的几个 CRC 标准。

- 支持三个常用的多项式：CRC-CCITT、CRC-16 和 CRC-32

CRC-CCITT:  $x^{16}+x^{12}+x^5+1$

CRC-16:  $x^{16}+x^{15}+x^2+1$

CRC-32:  $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

参数模型:

CRC-CCITT	CRC-16	CRC-32
Width: 16	Width: 16	Width: 32
Poly: 0x1021	Poly: 0x8005	Poly: 0x04C11DB7
Init: 0xFFFF	Init: 0x0000	Init: 0xFFFFFFFF
RefIn: False	RefIn: True	RefIn: True
RefOut: False	RefOut: True	RefOut: True
XorOut: 0x0000	XorOut: 0x0000	XorOut: 0xFFFFFFFF

- 接受任何字长的数据写入：8 位、16 位或 32 位



## 18.5 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	AESCFG	R/W	0000H	AES配置寄存器
04H	AESSTR	W	0000H	AES启动命令寄存器
08H	AESFLG	R/W	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/W	00000000H	AES输出已加/解密数据
20H	AESOUTML	R/W	00000000H	AES输出已加/解密数据
24H	AESOUTHM	R/W	00000000H	AES输出已加/解密数据
28H	AESOUTHH	R/W	00000000H	AES输出已加/解密数据
2CH	AESKEY0	R/W	00000000H	AES密钥寄存器
30H	AESKEY1	R/W	00000000H	AES密钥寄存器
34H	AESKEY2	R/W	00000000H	AES密钥寄存器
38H	AESKEY3	R/W	00000000H	AES密钥寄存器
3CH	AESKEY4	R/W	00000000H	AES密钥寄存器
40H	AESKEY5	R/W	00000000H	AES密钥寄存器
44H	AESKEY6	R/W	00000000H	AES密钥寄存器
48H	AESKEY7	R/W	00000000H	AES密钥寄存器

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	0000H	GHASH启动命令寄存器
04H	GHASHFLG	R/W	0000H	GHASH标志寄存器
08H	INPUT1LL	R/W	00000000H	GHASH输入数据1
0CH	INPUT1ML	R/W	00000000H	GHASH输入数据1
10H	INPUT1HM	R/W	00000000H	GHASH输入数据1
14H	INPUT1HH	R/W	00000000H	GHASH输入数据1
18H	INPUT2LL	R/W	00000000H	GHASH输入数据2
1CH	INPUT2ML	R/W	00000000H	GHASH输入数据2
20H	INPUT2HM	R/W	00000000H	GHASH输入数据2
24H	INPUT2HH	R/W	00000000H	GHASH输入数据2

28H	OUTPUTLL	R/W	00000000H	GHASH输出数据
2CH	OUTPUTML	R/W	00000000H	GHASH输出数据
30H	OUTPUTHM	R/W	00000000H	GHASH输出数据
34H	OUTPUTHH	R/W	00000000H	GHASH输出数据
38H	AESGHASHIE	R/W	00H	AES, Ghash中断使能位
3CH	AESGHASHIF	R/W	00H	AES, Ghash中断标志位

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

## 18.6 特殊功能寄存器说明

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

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

AESSTR (AES 启动寄存器)		基地址: 0x40012000 偏移地址: 04H	
Bit15...Bit0			

Read:	X							
Write:	DAT[15:0]							
Reset:	0	0	0	0	0	0	0	0

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

AESFLG (AES 标志寄存器)			基地址: 0x40012000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	BUSY
Write:								X
Reset:	0	0	0	0	0	0	0	0

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

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

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

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

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

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

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

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

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

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

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

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

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

AESOUTHM (AES 数据输出中高位)		基地址: 0x40012000 偏移地址: 24H						
	Bit31...Bit0							
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

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

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

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

AESKEY0 (AES 密钥 0)		基地址: 0x40012000 偏移地址: 2CH						
	Bit31...Bit0							
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

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

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

位	功能描述
DAT[31:0]	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)		基地址: 0x40012000 偏移地址: 34H						
	Bit31...Bit0							
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
DAT[31:0]	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)		基地址: 0x40012000 偏移地址: 38H						
	Bit31...Bit0							
Read:	DAT[31:0]							
Write:	DAT[31:0]							
Reset:	0	0	0	0	0	0	0	0

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

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

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

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

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

位	功能描述
DAT[31:0]	AES 密钥 192bit 的 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

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

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

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

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

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

RANDSTR (随机数启动控制寄存器)		基地址: 0x40012080 偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	RANDEN	X	X	X	X	BACKEN	BACKMOD E1	BACKMOD E0
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	Start	ModeSel 1	ModeSel 0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RANDEN	随机数模块使能位 0: 随机数模块关闭 1: 随机数模块使能 当 RANDEN 使能之后, 对 Start 写 1 才有效
BACKEN	随机数后处理使能 0: 处理关闭 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 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 该寄存器位写 1 则启动真随机数生成器, 该控制位在随机数生成器生成随机数后由硬件清 0
ModeSel[1..0]	真随机数模式选择控制: 00: 低频采高频模式 01: D 触发器不定态方式 10: 异或混合输出 11: 异或混合输出

RANDDAT	基地址: 0x40012080
---------	-----------------



(随机数数据寄存器)		偏移地址: 04H						
		Bit31…Bit0						
Read:	DAT[31:0]							
Write:								
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)		基地址: 0x40012100 偏移地址: 00H						
		Bit15…Bit0						
Read:	X							
Write:	DAT[15:0]							
Reset:	0	0	0	0	0	0	0	0

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

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

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

INPUT1LL	基地址: 0x40012100
----------	-----------------

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

位	功能描述
---	------

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

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

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

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

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

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

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

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

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

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

位	功能描述
RANDIE	随机数中断使能 0: 中断关闭 1: 中断使能
GHASHIE	GHASH 计算中断使能 0: 中断关闭 1: 中断使能
AESIE	AES 计算中断使能 0: 中断关闭 1: 中断使能

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	GHASHIF	AESIF
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RANDIF	随机数中断标志 0: 中断未产生

	1: 中断产生
GHASHIF	GHASH 计算中断标志 0: 中断未产生 1: 中断产生
AESIF	AES 计算中断标志 0: 中断未产生 1: 中断产生

CRCCON (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	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	MODE1	MODE0
Write:						RESET		
Reset:	0	0	0	0	0	0	0	0

位	功能描述												
RESET	CRC计算RESET位 复位CRC计算，但不改变当前模式，并根据当前模式设置CRCDAT寄存器； 只能对该位写‘1’，由硬件自动清零。												
MODE[1:0]	CRC计算模式控制位 <table border="1" style="margin-left: 20px;"> <tr> <th colspan="2">MODE[1:0]</th> <th>CRC计算模式</th> </tr> <tr> <td>0</td> <td>0</td> <td>CRC-CCITT</td> </tr> <tr> <td>0</td> <td>1</td> <td>CRC-16</td> </tr> <tr> <td>1</td> <td>X</td> <td>CRC-32</td> </tr> </table>	MODE[1:0]		CRC计算模式	0	0	CRC-CCITT	0	1	CRC-16	1	X	CRC-32
MODE[1:0]		CRC计算模式											
0	0	CRC-CCITT											
0	1	CRC-16											
1	X	CRC-32											

CRCDAT (数据寄存器)			基地址: 0x40021000 偏移地址: 04H					
	Bit31...Bit16							
Read:	DAT[31:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15...Bit0							
Read:	DAT[15:0]							
Write:								
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。

## 19 DMA 功能

### 19.1 概述

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

### 19.2 功能描述

- DMA 模块共有 3 个独立的可配置通道
- DMA 的传输可以外设到外设，外设到存储器，存储器到外设，存储器到存储器
- DMA 与 CPU 之间交替占用总线
- DMA 支持最多 64 个外部请求源，每个通道还有一个软件请求源
- 三个通道有优先级的设置，一旦一个通道的一次传输开始后不能被打断。当三个通道同时请求 DMA 传输时，优先级高的被响应。优先级低的通道只有在高优先级的通道传输完成后才能被响应
- 当两个通道选择了同一个外部请求源时，只有优先级高的通道响应此请求
- DMA 支持两种传输方式：一次请求传输一个数据、一次请求传输所有的数据
- DMA 支持三种中断：传输完成中断，传输错误中断，块传输中断
- 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 CHNxULKNUM[8:15]，达到循环次数后自动关闭相应通道

### 19.3 DMA 通道请求列表

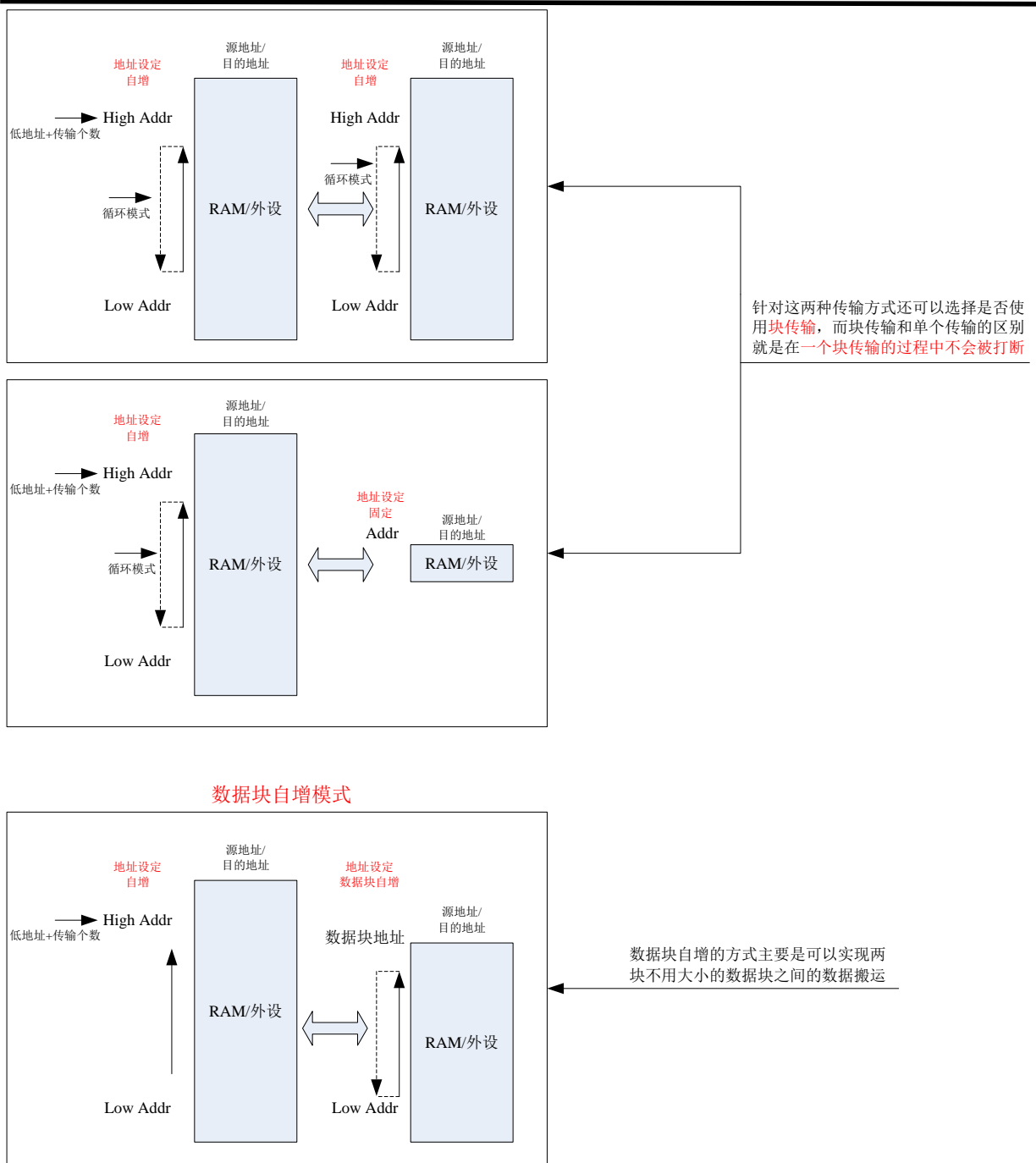
DMA_CTL[8...13]	说明
0	Soft request
1	UART0发送
2	UART0接收
3	UART1发送
4	UART1接收
5	UART2发送
6	UART2接收
7	UART3发送
8	UART3接收
9	UART4发送



10	UART4接收
11	UART5发送
12	UART5接收
13	IS07816_0发送
14	IS07816_0接收
15	IS07816_1发送
16	IS07816_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

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

## 19.4 DMA 数据传输说明



## 19.5 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
00H	DMAIE	R/W	0000H	中断使能寄存器
04H	DMAIF	R/W	0000H	通道中断标志
08H	CHNSTA	R/W	0000H	通道0状态寄存器
0CH	CHNOCTL	R/W	0000H	通道0控制寄存器
10H	CHNSRC	R/W	0000H	通道0源地址寄存器
14H	CHNOTAR	R/W	0000H	通道0目的地址寄存器
18H	CHNOCNT	R/W	0000H	通道0传输数量设置寄存器
1CH	CHNOTCCNT	R	0000H	通道0传输完成数据个数
20H	CHNOBULKNUM	R/W	0000H	通道0块传输个数设置
24H	CHN1CTL	R/W	0000H	通道1控制寄存器
28H	CHN1SRC	R/W	0000H	通道1源地址寄存器
2CH	CHN1TAR	R/W	0000H	通道1目的地址寄存器
30H	CHN1CNT	R/W	0000H	通道1传输数量设置寄存器
34H	CHN1TCCNT	R	0000H	通道1传输完成数据个数
38H	CHN1BULKNUM	R/W	0000H	通道1块传输个数设置
3CH	CHN2CTL	R/W	0000H	通道2控制寄存器
40H	CHN2SRC	R/W	0000H	通道2源地址寄存器
44H	CHN2TAR	R/W	0000H	通道2目的地址寄存器
48H	CHN2CNT	R/W	0000H	通道2传输数量设置寄存器
4CH	CHN2TCCNT	R	0000H	通道2传输完成数据个数
50H	CHN2BULKNUM	R/W	0000H	通道2块传输个数设置

## 19.6 特殊功能寄存器说明

DMAIE (DMA 中断使能寄存器)		基地址: 0x40020000 偏移地址: 00H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	TEIE2	TEIE1	TEIE0
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	BCIE2	BCIE1	BCIE0	X	TCIE2	TCIE1	TCIE0
Write:								
Reset:	0	0	0	0	0	0	0	0

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

DMAIF (DMA 中断标志寄存器)		基地址: 0x40020000 偏移地址: 04H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	TEIF2	TEIF1	TEIF0
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	BCIF2	BCIF1	BCIF0	X	TCIF2	TCIF1	TCIF0
Write:								
Reset:	0	0	0	0	0	0	0	0

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

CHNSTA (DMA 状态寄存器)		基地址: 0x40020000 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	BUSY2	BUSY1	BUSY0
Write:								
Reset:	0	0	0	0	0	0	0	0

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

CHNxCTL (DMA 通道控制寄存器)		基地址: 0x40020000 偏移地址: 0CH, 24H, 3CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	Channel	Channel	Channel	Channel	Channel	Channel	DESTIN_
Write:		5	4	3	2	1	0	INC1
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DESTIN_I	SOURC_I	SOURC_I	CYCLE	MODE	SIZE1	SIZE0	DMA_CHN
Write:	NC0	NC1	NC0					
Reset:	0	0	0	0	0	0	0	0

注: x 为 0,1,2

位	功能描述
Channel[5...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>目的地址增量模式</b> 00: 不增加 01: 增加 10: 数据块内循环增加 11: 数据块内循环增加
SOURC_INC[1...0]	<b>源地址地址增量模式</b> 00: 不增加 01: 增加 10: 数据块内循环增加 11: 数据块内循环增加
MODE	<b>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)
CYCLE	<b>循环模式</b> 0: 不循环模式 1: 循环模式

PSIZE[1...0]	MEMORY 的传送位数（外设的传送位数固定为 32bit）： 00: 8 位 01: 16 位 10: 32 位 11: 32bit
DMA_CHNOEN	DMA 通道使能 0: 禁止 1: 使能

CHNxSRC (DMA 通道源地址寄存器)		基地址: 0x40020000 偏移地址: 10H, 28H, 40H
	Bit31	Bit30 ... Bit1
Read:	ADDR31	ADDR30 ... ADDR1
Write:		
Reset:	0	0

注: x 为 0, 1, 2

位	功能描述
ADDR[31..0]	数据传输源地址寄存器

CHNxTAR (DMA 通道目的地址寄存器)		基地址: 0x40020000 偏移地址: 14H, 2CH, 44H
	Bit31	Bit30 ... Bit1
Read:	ADDR31	ADDR30 ... ADDR1
Write:		
Reset:	0	0

注: x 为 0, 1, 2

位	功能描述
ADDR[31..0]	数据传输目的地址寄存器

CHNxCNT (DMA 通道传输数量寄存器)			基地址: 0x40020000 偏移地址: 18H, 30H, 48H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num15	Num14	Num13	Num12	Num11	Num10	Num9	Num8
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num7	Num6	Num5	Num4	Num3	Num2	Num1	Num0
Write:								
Reset:	0	0	0	0	0	0	0	0

注: x 为 0, 1, 2

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

CHNxTCCNT (DMA 通道已传输数据个数)		基地址: 0x40020000 偏移地址: 1CH, 34H, 4CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num15	Num14	Num13	Num12	Num11	Num10	Num9	Num8
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num7	Num6	Num5	Num4	Num3	Num2	Num1	Num0
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

注: x 为 0, 1, 2

位	功能描述
Num[15...0]	<b>指示 DMA 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

CHNxBULKNUM (DMA 通道块传输设置寄存器)		基地址: 0x40020000 偏移地址: 20H, 38H, 50H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	CYCLE7	CYCLE6	CYCLE5	CYCLE4	CYCLE3	CYCLE2	CYCLE1	CYCLE0
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num7	Num6	Num5	Num4	Num3	Num2	Num1	Num0
Write:								
Reset:	0	0	0	0	0	0	0	0

注: x 为 0, 1, 2

位	功能描述
CYCLE[7...0]	<b>循环次数选择，当用户选择 DMA 传输为循环模式时起作用：</b> =0: 无限次循环 =0x01---0xFF: 为 0x01---0xFF 次循环
NUM[7...0]	<b>块传输数据个数选择，表示每一个数据块内有多少个数据：</b> 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

## 20 HRC/LRC 自动校正

### 20.1 概述

HRC/LRC 根据内部 32Hz 信号或者外部 1Hz 信号，能够自动调整 HRC/LRC 频率自动调整的过程（使能频率测试以及自动调整功能）：

- 第 1S（内部为第 1/32S），测得得到相应频率数据
- 第 2S（内部为第 2/32S），根据测得的数据，调整相应 RCADJ 寄存器（HRCADJ/LRCADJ）
- 第 3S（内部为第 3/32S），继续测频；第 4S（内部为第 4/32S）更新 RCADJ 寄存器，如此往复。
- 测频中断会在第 1s（1/32S），3s（3/32S），5s（5/32S）给出，即隔一个测频周期，而不是每个测频周期都会产生。

### 20.2 HRC 调整说明

HRCADJ 频率调整控制位总共有 7 位，最大 0x7F（对应最大频率），最小 0x00（对应最低频率），每个 LSB 约为 HRC 频率的 0.5%。HRC 中心频率为 11010048。

当 HRC 测到的频率与中心频率误差在 0.3% 以内，则不作调整。

### 20.3 LRC 调整说明

LRCADJ 频率调整控制位总共有 8 位，最大 0xFF（对应最大频率），最小 0x00（对应最低频率），每个 LSB 不确定。LRC 中心频率为 32768。

当 LRC 测到的频率与中心频率误差在 3% 以内，则不作调整。

### 20.4 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
0x60	RCCaliCON	R/W	0x0000	RC 校正配置寄存器
0x64	RCCaliIE	R/W	0x0000	RC 测频中断使能寄存器
0x68	RCCaliIF	R/W	0x0000	RC 测频中断标志寄存器
0x6C	HRCValue	R/W	0x000000	HRC 测频值 (25 位无符号数)
0x70	LRCValue	R/W	0x0000	LRC 测频值 (17 位无符号数)



## 20.5 特殊功能寄存器说明

RCCaliCON (RC 校正配置寄存器)			基地址: 0x4000F000 偏移地址: 60H					
	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	FreqSou	LRCStar	HRCStar	LRCAuto	HRCAuto
Write:				rce	t	t		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
FreqSource	<b>测频时钟源选择</b> 0: 选择内部 32Hz 为被测时钟源 1: 选择外部 1Hz 为被测时钟源 <b>注:</b> 当选择内部时钟源时, 无需配置 Tout 管脚, 也无需配置 RTCCON 寄存器。 当选择外部时钟源时, 需将对应管脚配置为 CLOCKIN 功能才行 (PA5 复用功能 2)。
LRCStart	<b>LRC 测频使能</b> 0: 关闭 LRC 对内部 32Hz 或者外部 1Hz 信号频率检测 1: 使能 LRC 对内部 32Hz 或者外部 1Hz 信号频率检测 <b>注:</b> 当使能后, 会在使能后第 1s (内部 1/32s), 3s (内部 3/32s), 5s (内部 5/32s) ... 给出相应中断标志及数据 如果时钟源是内部 32Hz, 则测到的频率值会被 X32 再放到相应的测频值寄存器 (HRCValue/ LRCValue)
HRCStart	<b>HRC 测频使能</b> 0: 关闭 HRC 对内部 32Hz 或者外部 1Hz 信号频率检测 1: 使能 HRC 对内部 32Hz 或者外部 1Hz 信号频率检测 <b>注:</b> 当使能后, 会在使能后第 1s (内部 1/32s), 3s (内部 3/32s), 5s (内部 5/32s) ... 给出相应中断标志及数据 如果时钟源是内部 32Hz, 则测到的频率值会被 X32 再放到相应的测频值寄存器 (HRCValue/ LRCValue)
LRCAuto	<b>LRC 自校正使能位</b> 0: 关闭 LRC 自动测试。此时用户可以自己写 LRCADJ 寄存器调整 LRC 频率 1: 使能 LRC 自动测试。此时用户写 LRCADJ 寄存器无效。(用户需同时使能 LRCCaliStart)
HRCAuto	<b>HRC 自校正使能位</b> 0: 关闭 HRC 自动测试。此时用户可以自己写 HRCADJ 寄存器调整 HRC 频率

	1: 使能 HRC 自动测试。此时用户写 HRCADJ 寄存器无效。(用户需同时使能 HRCCaliStart)
--	--

RCCaliIE (RC 测频中断使能寄存器)			基地址: 0x4000F000 偏移地址: 64H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
	0	0	0	0	0	0	0	0
Reset	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	LRCCali	HRCCali
Write:							IE	IE
Reset	0	0	0	0	0	0	0	0

位	功能描述
LRCIE	LRC 测频中断使能 0: 关闭 LRC 测频中断 1: 使能 LRC 测频中断
HRCIE	HRC 测频中断使能 0: 关闭 HRC 测频中断 1: 使能 HRC 测频中断

RCCaliIF (RC 测频中断标志寄存器)			基地址: 0x4000F000 偏移地址: 68H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
	0	0	0	0	0	0	0	0
Reset	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	LRCCaliIF	HRCCaliIF
Write:								
Reset	0	0	0	0	0	0	0	0

位	功能描述
LRCIF	LRC 测频中断标志 0: 未产生 LRC 测频中断 1: 产生 LRC 测频中断
HRCIF	HRC 测频中断标志 0: 未产生 HRC 测频中断

	1: 产生 HRC 测频中断
--	----------------

HRCValue (HRC 测频值)		基地址: 0x4000F000 偏移地址: 6CH						
	Bit31	30	29	28	27	26	25	Bit24
Read:	X	X	X	X	X	X	X	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

HRC 测频值, 25 位无符号数

LRCValue (LRC 测频值)		基地址: 0x4000F000 偏移地址: 70H						
	Bit23	22	21	20	19	18	17	Bit16
Read:	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:								
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0

LRC 测频值, 17 位无符号数

## 21 Cortex-M0 内核简要说明

### 21.1 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)。

### 21.2 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。

### 21.3 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

### 21.4 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

## 22 电气规格

### 22.1 DC 参数

符号	参数说明	测试条件	最小	典型	最大	单位
VCC	输入电源	输入电源引脚上的电压	2.0	5	5.5	V
Vih	高电平输入电压	Reset 引脚, TEST 引脚	0.8VCC			V
Vih	高电平输入电压	支持 TTL 输入管脚	0.55VCC \2.6V			V
Vih	高电平输入电压	除支持 TTL 输入管脚外其他普通管脚	0.7VCC			V
Vil	低电平输入电压	除了电源和地之外的所有引脚			0.2VCC	V
Ioh (大电流管脚)	高电平输出电流	VCC=5V I/O 口上电压 Vio 降低到 0.9VCC 测试引脚为: PA.6, PA.7, PA.8, PC.0	15		VCC/Gnd	mA
Iol (大电流管脚)	低电平输出电流	VCC=5V I/O 口上电压 Vio 升高到 0.1VCC 测试引脚为: PA.6, PA.7, PA.8, PC.0	30		VCC/Gnd	mA
Ioh (普通管脚)	高电平输出电流	VCC=5V I/O 口上电压 Vio 降低到 0.9VCC	5		VCC/Gnd	mA
Iol (普通管脚)	低电平输出电流	VCC=5V I/O 口上电压 Vio 升高到 0.1VCC	9		VCC/Gnd	mA

注：支持 TTL 和不支持 TTL 输入管脚参见 1.4 引脚定义

### 22.2 极限参数

符号	参数说明	测试条件	最小	最大	单位
VCC	输入电源	输入电源引脚上的电压	2.0	5.5	V
Vi	输入电压	所有的数字引脚	-0.3	5	V
Via	ADC 输入极限电压	测试 ADC 引脚输入电压, 高于此电压可能会导致该引脚损坏	-0.3	800	mV

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
Vesd	静态 ESD (HBM)	芯片所有的引脚	-8000	+6000	V

## 22.3 功耗参数

测试条件说明	最小	典型	最大	单位
<b>测试条件：</b> 以下功耗参数均为在 5V 供电下测试得到				
芯片处于 Hold 模式下最低功耗(具体功耗值会根据用户在 Hold 模式下打开不同的数字模块而不同)		5.0		uA
芯片处于 Sleep 模式下功耗		4.0		uA
芯片在低频 32768Hz 下运行功耗，LCD 开启，BOR 开启，LVDIN0/1 开启		70		uA
高频 RC 下运行，高频 RC 配置为 11M，CLKOUT 关闭，所有数字模块打开 \ 关闭		3.62 \2.49		mA
高频 RC 下运行，高频 RC 配置为 5.5M，CLKOUT 关闭，所有数字模块打开 \ 关闭		2.52 \1.37		mA
高频 RC 下运行，高频 RC 配置为 2.75M，CLKOUT 关闭，所有数字模块打开 \ 关闭		1.96 \0.81		mA
高频 RC 下运行，高频 RC 配置为 1.375M，CLKOUT 关闭，所有数字模块打开 \ 关闭		1.67 \0.52		mA
高频 RC 下运行，高频 RC 配置为 700K，CLKOUT 关闭，所有数字模块打开 \ 关闭		1.53 \0.38		mA
PLL 时钟下运行，CPU 运行 PLL 时钟 44M，所有数字模块打开 \ 关闭		12.69 \9.57		mA
PLL 时钟下运行，CPU 运行 PLL 时钟 22M，所有数字模块打开 \ 关闭		6.96 \5.16		mA
PLL 时钟下运行，CPU 运行 PLL 时钟 11M，所有数字模块打开 \ 关闭		4.75 \2.88		mA
PLL 时钟下运行，CPU 运行 PLL 时钟 5.5M，所有数字模块打开 \ 关闭		3.58 \1.67		mA
PLL 时钟下运行，CPU 运行 PLL 时钟 2.75M，所有数字模块打开 \ 关闭		3.00 \1.07		mA
PLL 时钟下运行，CPU 运行 PLL 时钟 1.375M，所有数字模块打开 \ 关闭		2.70 \0.76		mA

测试条件说明	最小	典型	最大	单位
<b>测试条件：</b> 以下功耗参数均为在 5V 供电下测试得到				
高频 RC 模块功耗 (HRC)		114.5		uA
低频 RC 模块功耗 (LRC)		0.3		uA
PLL 模块功耗		257		uA
RTC 模块功耗 (含 TBS 补偿)		1.9		uA
LCD 模块功耗 (快速充电模式)		14.6		uA
BOR 模块功耗		8		uA
LVDIN 模块		7		uA
VCC=3.6V 下测试, hold 模式下打开 LCD 模块的功耗		18.4		uA

模块处于低电压域中, 5V 和 3.3V 供电下功耗差异很小。

测试条件说明	最小	典型	最大	单位
<b>测试条件：</b> 以下功耗参数均为在 3.3V 供电下测试得到				
芯片处于 Hold 模式下最低功耗(具体功耗值会根据用户在 Hold 模式下打开不同的数字模块而不同)		3.3		uA
芯片处于 Sleep 模式下功耗		2.9		uA
芯片在低频 32768Hz 下运行功耗, LCD 开启, BOR 开启, LVDIN0/1 开启		48		uA
高频 RC 下运行, 高频 RC 配置为 11M, CLKOUT 关闭, 所有数字模块打开 \ 关闭		3.59 2.48		mA
高频 RC 下运行, 高频 RC 配置为 5.5M, CLKOUT 关闭, 所有数字模块打开 \ 关闭		2.49 1.36		mA
高频 RC 下运行, 高频 RC 配置为 2.75M, CLKOUT 关闭, 所有数字模块打开 \ 关闭		1.93 0.79		mA
高频 RC 下运行, 高频 RC 配置为 1.375M, CLKOUT 关闭, 所有数字模块打开 \ 关闭		1.65 0.51		mA
高频 RC 下运行, 高频 RC 配置为 700K, CLKOUT 关闭, 所有数字模块打开 \ 关闭		1.51 0.37		mA
PLL 时钟下运行, CPU 运行 PLL 时钟 44M, 所有数字模块打开 \ 关闭		12.63 9.54		mA
PLL 时钟下运行, CPU 运行 PLL 时钟 22M, 所有数字模块打开 \ 关闭		6.93 5.15		mA
PLL 时钟下运行, CPU 运行 PLL 时钟 11M, 所有数字模块打开 \ 关闭		4.72 2.86		mA
PLL 时钟下运行, CPU 运行 PLL 时钟 5.5M, 所有数字模块打开 \ 关闭		3.56 1.66		mA
PLL 时钟下运行, CPU 运行 PLL 时钟 2.75M, 所有数字模块打开 \ 关闭		2.97 1.06		mA
PLL 时钟下运行, CPU 运行 PLL 时钟 1.375M, 所有数字模块打开 \ 关闭		2.68 0.75		mA



测试条件说明	最小	典型	最大	单位
<b>测试条件：</b> 以下功耗参数均为在 3.3V 供电下测试得到				
高频 RC 模块功耗 (HRC)		113.3		uA
低频 RC 模块功耗 (LRC)		0.3		uA
PLL 模块功耗		255.5		uA
RTC 模块功耗 (含 TBS 补偿)		1.7		uA
LCD 模块功耗 (快速充电模式)		9.6		uA
BOR 模块功耗		6.2		uA
LVDIN 模块		7		uA
VCC=3.3V 下测试, hold 模式下打开 LCD 模块的功耗 (快速 1/64 充电模式)		13		uA

## 22.4 外部 ADC 参数

符号	参数说明	测试条件	最小	典型	最大	单位
Via	ADC 输入电压		0		800	mV
Fc	ADC 转换频率	VCC=5V	1/16		1	Hz
Res	分辨率	VCC=5V		0.18		mv/LSB

## 22.5 ADCBAT 测试参数

符号	参数说明	测试条件	最小	典型	最大	单位
Via	ADCBAT 输入电压		0		5	V
Fc	ADC 转换频率	VCC=5V	1/16		1	Hz
Res	分辨率	VCC=5V		0.17		mv/LSB

## 22.6 VCC 测试参数

符号	参数说明	测试条件	最小	典型	最大	单位
Via	VCC 输入电压		2.0		5.5	V
Fc	ADC 转换频率		1/16		1	Hz
Res	分辨率	满足 VCC 正常工作电压范围内, ADC 内部基准源保持恒定, 测试以内部基准源为准。		0.17		mv/LSB

VCC 电源上升时间——内部 POR 和 LBOR 互补控制可保证系统有效上电复位, 理论上对 VCC 上升时间无要求。

## 23 封装

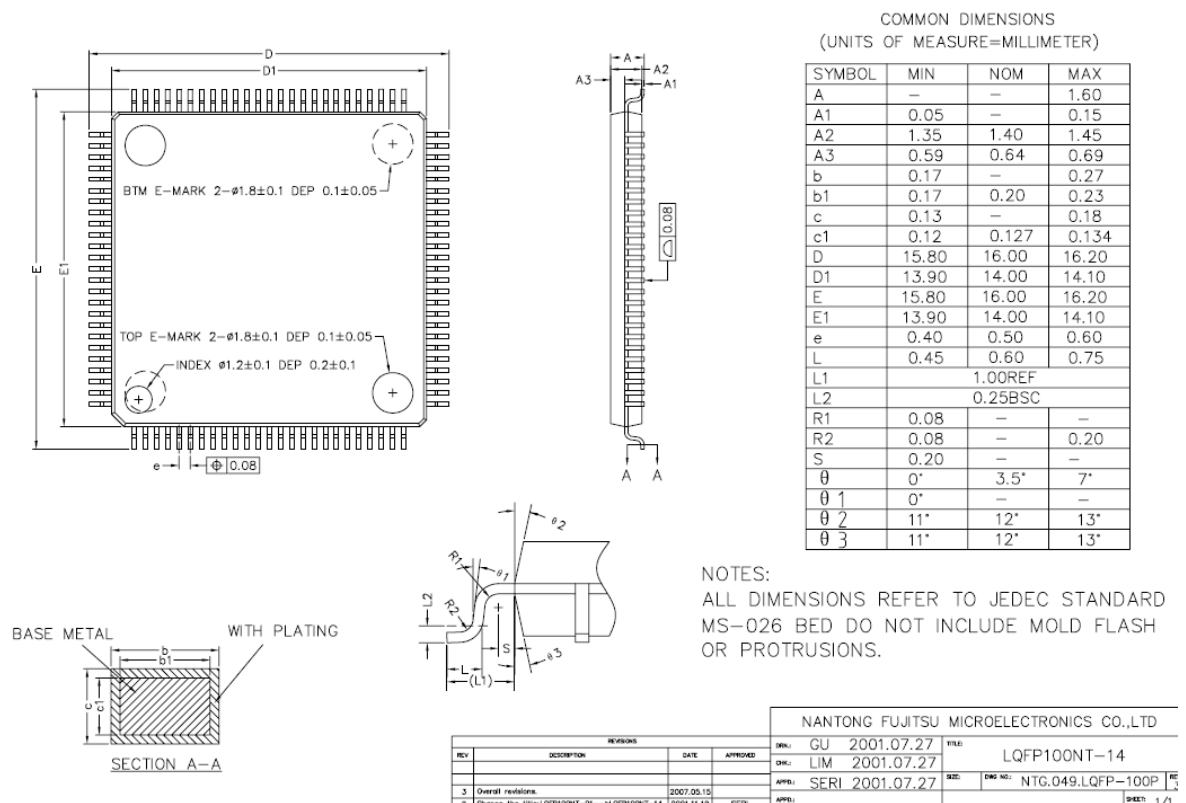
### 23.1 耐焊性

耐焊性：经过 260℃，不少于 5S 的回流焊，芯片能够保证正常工作。

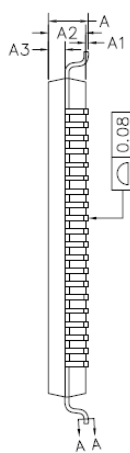
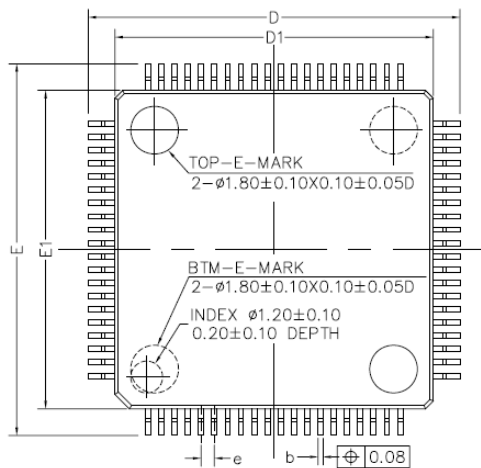
建议过锡炉次数：不超过 3 次。

### 23.2 封装图

HT6023: LQFP100

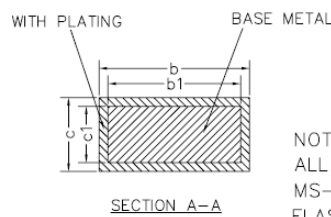
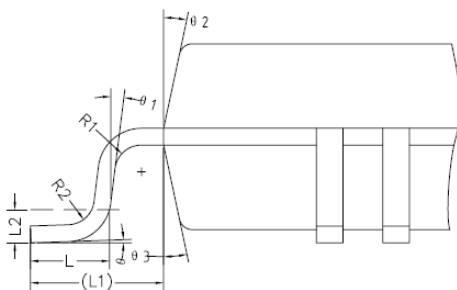


HT6025: LQFP80



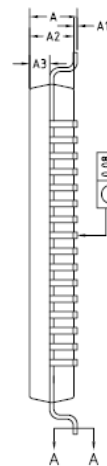
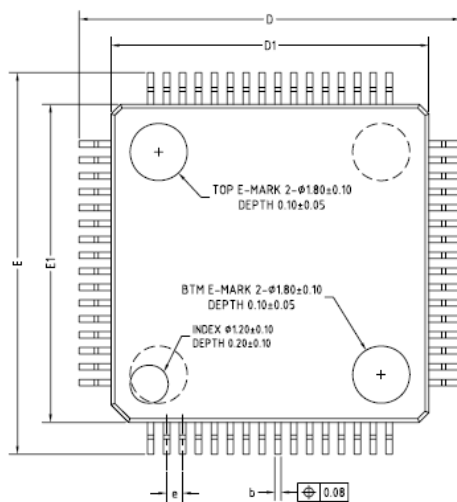
COMMON DIMENSIONS  
(UNITS OF MEASURE=MILLIMETER)

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°



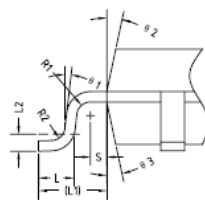
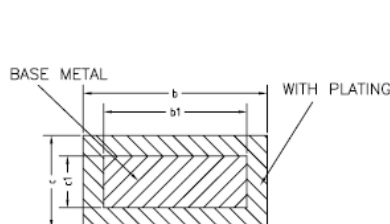
NOTES:  
ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026 BDD DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

## HT6027: LQFP64



COMMON DIMENSIONS  
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	-	-	1.60
A1	0.05	-	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	-	0.27
b1	0.17	0.20	0.23
c	0.13	-	0.18
c1	0.12	0.127	0.134
D	11.80	12.00	12.20
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
e	0.50BSC		
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
R1	0.08	-	-
R2	0.08	-	0.20
S	0.20	-	-
θ	0°	3.5°	7°
θ 1	0°	-	-
θ 2	11°	12°	13°
θ 3	11°	12°	13°



NOTES:  
ALL DIMENSIONS MEET JEDEC STANDARD MS-026 BCD DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.