

# ATT7039AU

# 用户手册

炬泉光电科技（上海）股份有限公司

Tel: 021-51035886  
Fax: 021-50277833  
Email: [sales@hitrendtech.com](mailto:sales@hitrendtech.com)  
Web: <http://www.hitrendtech.com>

### 版本更新说明

版本号	修改时间	修改内容
V0.1	2010/06/25	初稿
V0.2	2010/07/10	1. 更改存储器寻址空间 2. 添加引脚分布图
V0.3	2010/07/15	1. 在 1.4 节添加系统框图, 修改 2.5 节模块框图 2. 将有关 EMUIE 的地址由 0xAD 更改为 0xAC
V0.4	2010/11/25	1. 将寄存器 VDCR 的复位值由 0x78 更正为 0x79 2. 将寄存器 ADCCON 的复位值由 0x28 更正为 0x3C 3. TBS 模块“温度计算举例”中表格的第二行第三列的值由 7BH 更正为 FBH 4. 增加串口 1 的波特率计算公式; 列举串口常用波特率的配置值 5. 在校表参数寄存器列表中添加“默认值”项
V0.5	2011/2/27	1. 删除 ATT7035 中包含而 ATT7039 中不包含的内容; 2. 对寄存器 DPTR/PCON/ADCON 重新描述, 使之更严谨清晰; 3. 将系统复位-功能说明-2) 中“保持 128 个 fosc”修改为“保持 64 个 fosc”; 4. 在中断系统章节中添加寄存器 IRCON 的定义; 5. 在 EMU 章节中添加寄存器 Rosi_ctrl 的位定义; 6. 纠正寄存器 emu_ctrl 位 emu_1kctrl 的定义; 7. 修改 KBI 章节的内容描述, 使之更严谨清晰; 8. 纠正 LCD 与 GPIO 引脚复用表格中的错误; 9. 更改寄存器 I2CSTA 的默认值为 F8H; 10. 纠正其他地方笔误 17 处;
V0.6	2011/3/21	1. 删除 TBS 模块/删除时钟输出模块 2. 删除 Flash 的第一种映射模式 3. 对 SFR 寄存器整理, 删除 7039 中不再使用的寄存器、寄存器位 4. 去除电流通道 2; 去除无功功率, 无功电能, 视在功率, 视在电能; 去除防窃电的功能、电压电流夹角功能;
V0.61	2011/4/26	添加封装尺寸图
V0.62	2011/8/3	修改 xram 为 1k
V0.63	2011/4/26	更改为新模版格式
V1.00	2012-3-2	<b>针对 G 版功能升级, 进行修改:</b> 1, Pin25 为 QF/TOUT; 2, 内置 10M 偏置电阻, 外部无需 10M 电阻; 3, 寄存器 P02CFG 增加控制位: QF/TOUT; 4, 寄存器 MCON 位 RAMMAP 增加定义; 5, 可检测无功, 增加 QF 中断; 计量参数寄存器增加无功相关功率和能量寄存器, 无功校验增益配置寄存器;

		<ol style="list-style-type: none"><li>6, 增加校表校验和寄存器 Scheck;</li><li>7, ROSI_CTRL 寄存器增加 Clkdelay2, ClkDelay1 , Chop_VrefEn;</li><li>8, RTCTOUT 管脚输出秒信号,可以实现高频每秒补偿,修改寄存器 RTCCON;</li><li>9, KBI 可配置上升沿或者下降沿,修改寄存器 KEYIE;</li><li>10, 删除 hold 模式,修改寄存器 CLKCFG;</li><li>11, LCD KEYCFG 增加控制位 seg21、seg23,最大可 16segmentX4Com;</li><li>12, VDCR 寄存器增加: OSC 大电流/小电流模式, ADCChopper 使能信号, 去掉 LVref_O 调节选项。</li></ol>
--	--	---

## 目 录

<b>1</b>	<b>芯片概况</b>	<b>8</b>
1.1	芯片简介	8
1.2	芯片特性	8
1.2.1	基本特点	8
1.2.2	电能计量	8
1.2.3	处理器及外设	8
1.3	整体框图	9
1.4	引脚框图	10
1.5	引脚定义	11
1.6	缩略语	12
<b>2</b>	<b>电源与时钟</b>	<b>13</b>
2.1	电源管理	13
2.1.1	概述	13
2.1.2	框图	13
2.1.3	电源切换	14
2.1.4	电源实时监测	14
2.1.5	1.8V电源	14
2.1.6	模拟电源	14
2.1.7	Sleep 模式	14
2.1.8	特殊功能寄存器	15
2.2	系统复位	22
2.2.1	概述	22
2.2.2	框图	22
2.2.3	上电复位 (POR)	22
2.2.4	外部引脚复位	23
2.2.5	掉电复位 (BOR & LBOR)	23
2.2.6	电子狗复位	24
2.2.7	唤醒复位	24
2.2.8	软复位	24
2.2.9	特殊功能寄存器	25
2.2.10	被分级复位的寄存器说明	26
2.3	时钟管理	28
2.3.1	概述	28
2.3.2	框图	29
2.3.3	低频振荡电路	29
2.3.4	高频振荡电路	29
2.3.5	系统时钟切换	30
2.3.6	时钟管理单元内部保护机制	30

---

2.3.7	外设时钟管理.....	31
2.3.8	特殊功能寄存器.....	31
<b>3</b>	<b>控制单元 (MCU) .....</b>	<b>34</b>
3.1	MCU架构.....	34
3.1.1	概述.....	34
3.1.2	框图.....	35
3.1.3	存储器.....	36
3.1.4	指令系统.....	37
3.1.5	特殊功能寄存器.....	43
3.2	存储单元.....	50
3.2.1	概述.....	50
3.2.2	ATT7039 Flash特性.....	50
3.2.3	框图.....	51
3.2.4	存储器映射及类别说明.....	51
3.2.5	Flash 的操作.....	52
3.2.6	特殊功能寄存器.....	53
3.3	中断系统.....	55
3.3.1	概述.....	55
3.3.2	中断列表.....	55
3.3.3	中断优先级.....	55
3.3.4	特殊功能寄存器.....	56
3.3.5	中断处理.....	61
3.4	定时器.....	61
3.4.1	概述.....	61
3.4.2	框图.....	61
3.4.3	定时器工作模式.....	62
3.4.4	定时器0.....	62
3.4.5	定时器1.....	63
3.4.6	定时器2.....	64
3.4.7	特殊功能寄存器.....	65
<b>4</b>	<b>计量模块.....</b>	<b>69</b>
4.1	概述.....	69
4.2	框图.....	70
4.3	功能描述.....	70
4.3.1	模数转换器.....	70
4.3.2	ADC采样输出和功率波形输出.....	71
4.3.3	有功功率、无功功率和视在功率.....	71
4.3.4	有效值.....	71
4.3.5	电压线频率.....	71
4.3.6	起动/潜动.....	72
4.3.7	功率反向指示.....	72
4.3.8	直流偏置校正.....	72

---

4.3.9	电能脉冲输出.....	72
4.4	中断源.....	73
4.5	寄存器.....	74
4.5.1	特殊功能寄存器.....	74
4.5.2	间接寄存器.....	76
4.6	校表过程.....	90
<b>5</b>	<b>其他外围功能.....</b>	<b>93</b>
5.1	WDT.....	93
5.1.1	概述.....	93
5.1.2	工作模式.....	93
5.1.3	特殊功能寄存器.....	93
5.2	KBI.....	95
5.2.1	概述.....	95
5.2.2	按键中断.....	95
5.2.3	按键唤醒低功耗模式.....	95
5.2.4	特殊功能寄存器.....	95
5.3	LCD.....	97
5.3.1	概述.....	97
5.3.2	LCD与GPIO引脚复用.....	97
5.3.3	框图.....	97
5.3.4	输出波形.....	98
5.3.5	特殊功能寄存器.....	101
5.4	RTC.....	105
5.4.1	概述.....	105
5.4.2	功能描述.....	105
5.4.3	特殊功能寄存器.....	108
5.5	JTAG.....	114
5.5.1	概述.....	114
5.5.2	介绍.....	114
5.6	GPIO.....	115
5.6.1	概述.....	115
5.6.2	特殊功能寄存器.....	115
<b>6</b>	<b>通信接口.....</b>	<b>122</b>
6.1	UART.....	122
6.1.1	概述.....	122
6.1.2	串口0.....	122
6.1.3	串口1.....	125
6.1.4	特殊功能寄存器.....	128
6.2	红外模块.....	133
6.2.1	概述.....	133
6.2.2	特殊功能寄存器.....	133
6.3	I2C.....	134

---

6.3.1	概述.....	134
6.3.2	框图.....	134
6.3.3	功能描述.....	134
6.3.4	特殊功能寄存器.....	135
<b>7</b>	<b>电气规格.....</b>	<b>145</b>
7.1	绝对最大额定值.....	145
7.2	电器特性.....	145
7.2.1	DC参数.....	145
7.2.2	功耗参数.....	146
7.2.3	电能计量参数.....	146
7.2.4	ADC指标.....	146
7.2.5	ADC基准电压.....	147
<b>8</b>	<b>封装.....</b>	<b>147</b>
<b>9</b>	<b>典型应用.....</b>	<b>147</b>

## 1 芯片概况

### 1.1 芯片简介

ATT7039 是 ATT7037 的精简版本，片内集成单相计量、处理器、电源管理、时钟管理，PLL，JTAG 调试等功能。

### 1.2 芯片特性

#### 1.2.1 基本特点

- 工作电压范围：2.7V-3.6V
- 工作温度范围：-40°C~85°C
- 封装：LQFP-48L

#### 1.2.2 电能计量

- 动态范围 3000:1，有功电能误差小于 0.1%，支持 IEC62053-21、IEC62053-22 标准的精度要求
- 动态范围 3000:1，无功电能误差小于 0.1%，支持 IEC62053-23 标准的精度要求
- 电流/电压有效值测量误差小于 0.5%
- 提供两路 ADC 的原始采样数据和瞬时波形采样数据
- 电流和电压采样通道具有 4 级模拟/数字增益可调，支持分流器和互感器直接接入
- 片内基准电压：1.18V±2%（温度系数±25ppm/°C）
- 提供有功、无功电能脉冲输出，并开放快速脉冲计数寄存器，可保存掉电电能
- 提供多种电能累加方式
- 支持阻性功率增益误差、相位增益误差校验的软件校表方式
- 支持防潜动功能，启动功率可灵活设置

#### 1.2.3 处理器及外设

- 基于 8 位 R8051XC 设计，具有 8052 兼容指令集和总线结构
- 单 Cycle 的 CPU
- 片内集成 PLL 倍频电路
- 片内集成丰富的存储器资源，包括 32K 的 FLASH 程序存储器，其中 4K 可配制的 Flash 数据存储存储器，擦除次数大于 20 万次；256 字节带有写保护操作的 Info FLASH 存储器；256 字节内部数据寄存器；1K 字节外部数据寄存器，其中前 256 字节在掉电时，可由后备电源保持数据不丢失
- 具有电源监测功能，电源监测域值可设置，保证上下电能够可靠复位
- SLEEP 模式下支持外部中断等 6 种唤醒方式
- 片内集成可永不关断的硬件看门狗电路
- 片内集成 RTC 模块，可输出秒脉冲进行校验，实现每秒时钟补偿
- 片内集成按键、串行通讯、LCD、红外调制、I2C 等外设



1.3 整体框图

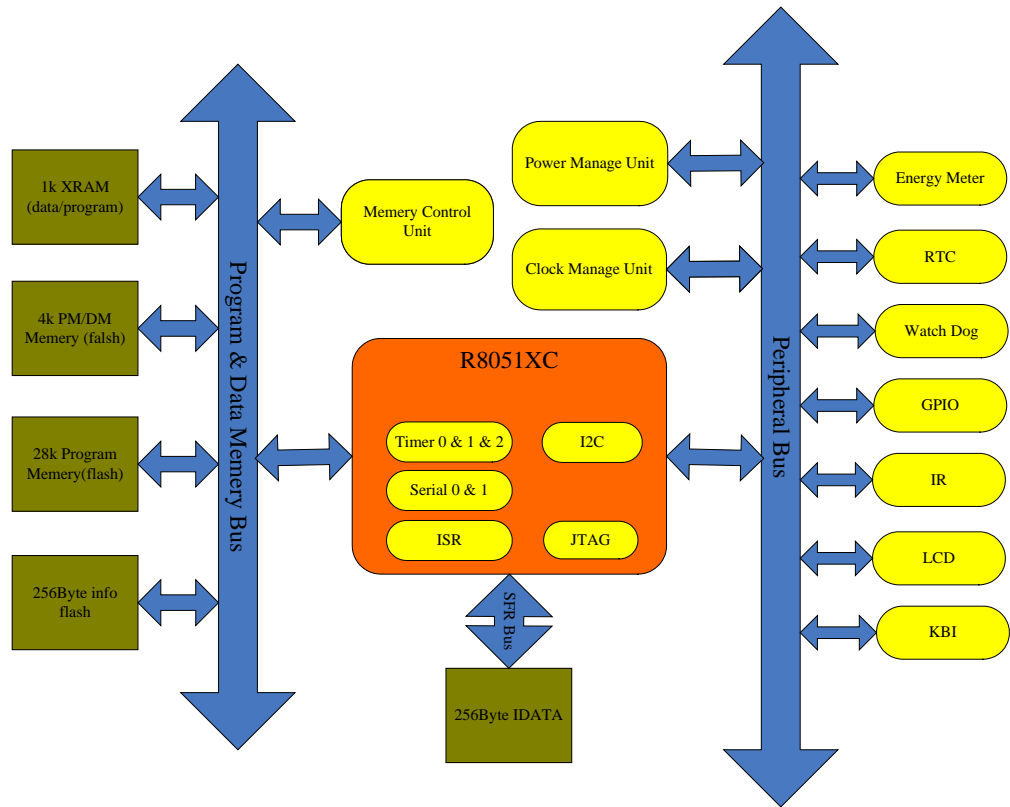


图 1-3-1 ATT7039 功能模块图

1.4 引脚框图

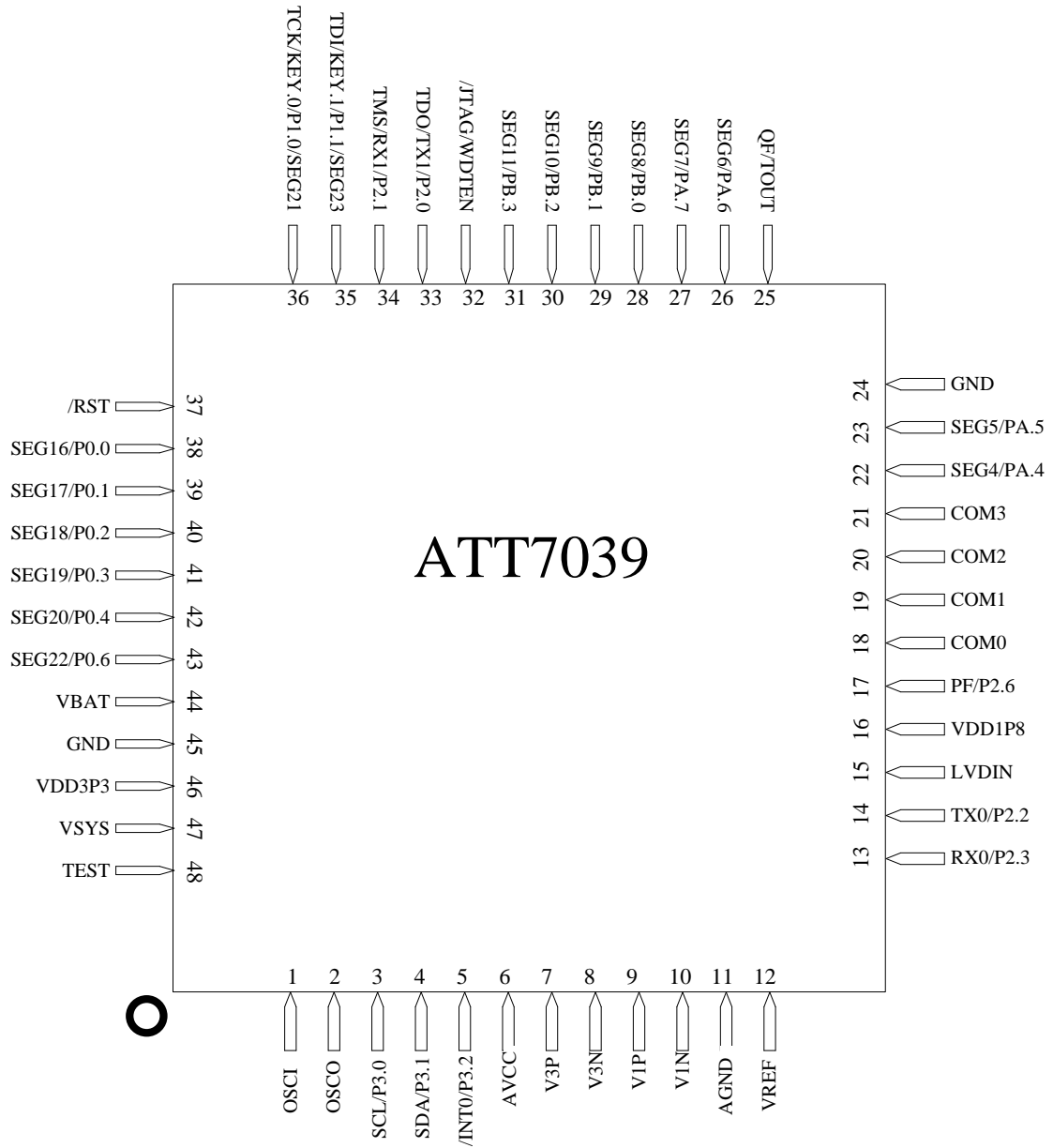


图 1-4-1 ATT7039 引脚分布图

## 1.5 引脚定义

表 1-5-1 引脚定义列表

Pin No.	标识	Pin type	功能描述
1	OSCI	INPUT	32k 晶振输入端
2	OSCO	INPUT	32k 晶振输出端
3	SCL_P30	IN/OUT	I2C_CLK/GPIO3.0
4	SDA_P31	IN/OUT	I2C_DATA/GPIO3.1
5	INT0_P32	IN/OUT	外部中断 0/GPIO3.2
6	AVCC	POWER	模拟电源输入端
7	V3P	INPUT	电压信号输入端（正）
8	V3N	INPUT	电压信号输入端（负）
9	V1P	INPUT	第一通道电流信号输入端（正）
10	V1N	INPUT	第一通道电流信号输入端（负）
11	AGND	GND	AGND
12	VREF	OUTPUT	1.18V 基准输出端，外接 10uF+0.1uF 电容
13	RX0_P23	IN/OUT	RX0/GPIO2.3
14	TX0_P22	IN/OUT	TX0/GPIO2.2
15	LVDIN	IN/PUT	LVDIN 检测输入端，外接分压电阻
16	VDD1P8	POWER	1.8V 电压输出端，外接 0.1uF 电容
17	PF_P26	IN/OUT	PF/ GPIO2.6
18	COM0	OUTPUT	COM0
19	COM1	OUTPUT	COM1
20	COM2	OUTPUT	COM2
21	COM3	OUTPUT	COM3
22	SEG4_COM4_PA4	IN/OUT	SEG4/COM4/ GPIOA.4
23	SEG5_COM5_PA5	IN/OUT	SEG5/COM5/ GPIOA.5
24	GND	GND	GND
25	QF/TOUT	OUTPUT	QF/TOUT 输出
26	SEG6_COM6_PA6	IN/OUT	SEG6/COM6/ GPIOA.6
27	SEG7_COM7_PA7	IN/OUT	SEG7/COM7/ GPIOA.7
28	SEG8_PB0	IN/OUT	SEG8/GPIOB.0
29	SEG9_PB1	IN/OUT	SEG9/GPIOB.1
30	SEG10_PB2	IN/OUT	SEG10/GPIOB.2
31	SEG11_PB3	IN/OUT	SEG11/GPIOB.3
32	JTAG_WDTEN	INPUT	JTAG_WDTEN 0: jtag 模式, WDT 关闭 1: normal 模式, WDT 使能
33	TDO_TX1_P20	IN/OUT	TDO/TX1/GPIO2.0
34	TMS_RX1_P21	IN/OUT	TDI/RX1/GPIO2.1
35	TDI_KEY1_P11_SEG23	IN/OUT	TMS/KEY1/GPIO1.1/SEG23
36	TCK_KEY0_P10_SEG21	IN/OUT	TCK/KEY0/GPIO1.0/SEG21

37	RST	INPUT	RESET
38	SEG16_P00	IN/OUT	SEG16/GPIO0.0
39	SEG17_P01	IN/OUT	SEG17/GPIO0.1
40	SEG18_P02	IN/OUT	SEG18/ GPIO0.2
41	SEG19_P03	IN/OUT	SEG19/ GPIO0.3
42	SEG20_P04	IN/OUT	SEG20/ GPIO0.4
43	SEG22_P06	IN/OUT	SEG22/ GPIO0.6
44	VBAT	POWER	VBAT 电压输入端
45	GND	GND	GND
46	VDD3P3	Power	内部 3.3V 电压输出端, 接 10uF+0.1uF 电容
47	VSYS	Power	系统 3.3V 电源输入端
48	TEST	INPUT	floating 或经 10K 电阻上拉

## 1.6 缩略语

表 1-6-1 缩略语

缩略语	英文原文	中文含义
SFR	Special Function Register	特殊功能寄存器
PM	Programe memory	指映射到 PM 空间的 32K 字节 Flash
DM	Data memory	指映射到 DM 空间的 4K 字节 Flash
WDT	Watch Dog Timer	看门狗
GPIO	General Purpose IO	通用 IO
POR	Power On Reset	上电复位
BOR	Brown Out Reset	掉电复位
WKR	Wakeup Reset	唤醒复位
EMU	Energy Mearsurment Unit	电能计量单元
PDM	Power Down Mode	掉电模式
PMU	Power Management Unit	系统电源管理
CMU	Clock Management Unit	系统时钟管理
KBI	Key Board Interface	键盘接口
UAM	User Application Mode	用户程序模式

## 2 电源与时钟

### 2.1 电源管理

#### 2.1.1 概述

PMU 为芯片的电源管理单元，功能如下：

- 监测系统电源 VSYS 和掉电检测输入端 LVDIN，当供电电压低或高于设定阈值时产生中断信号；
- 自动完成 VSYS 电源和 VBAT 电源的切换，电源状态在 PMSR 寄存器中的状态位 VSYSS 更新；
- 监测芯片内部的工作电源 VDD3P3，可以根据设定阈值产生中断信号；
- 监测芯片内部的工作电源 VDD3P3，可以根据设定阈值产生 BOR、LBOR、POR 复位信号；
- 为芯片内部数字模块提供 1.8V 电源，并由 VDD1P8 输出；

#### 2.1.2 框图

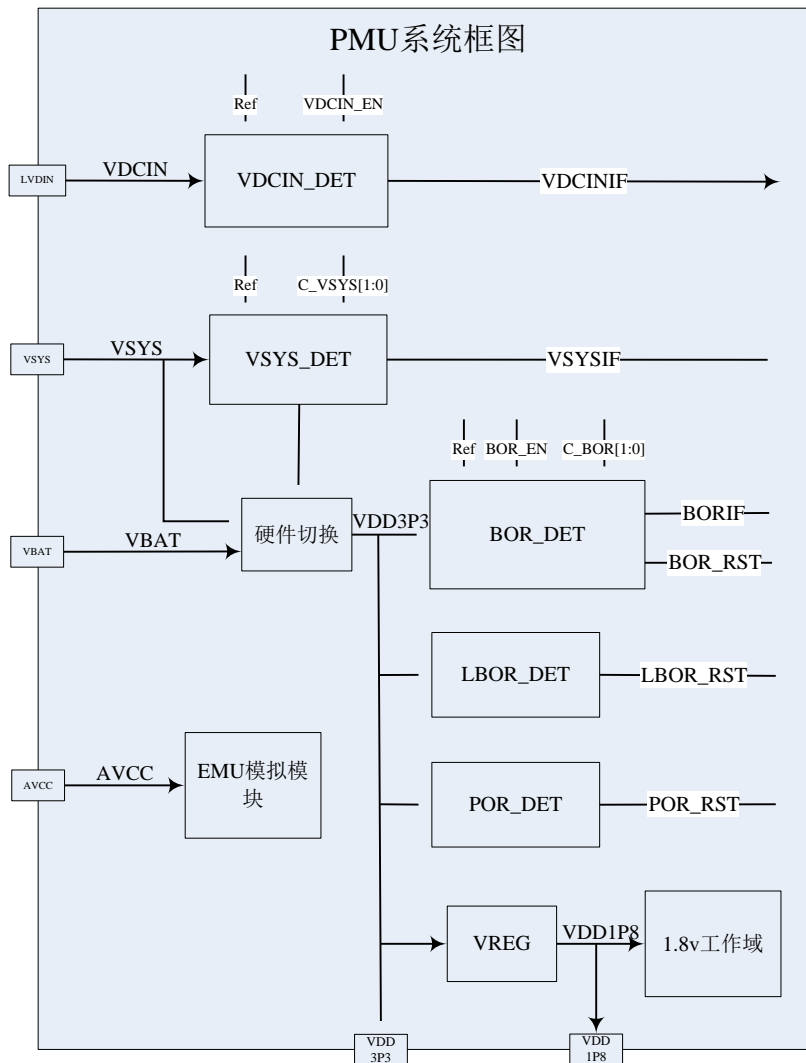


图 2-1-1 PMU 单元系统框图

### 2.1.3 电源切换

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

### 2.1.4 电源实时监测

PMU单元共设置五个子模块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，当电压低或高于1.185V时，置位LVDINIF标志位；如果使能了LVDINIE中断，就会响应产生LVDIN中断；
- **VSYS\_DET模块**：监测外部引脚VSYS的电压，当电压低或高于设定阈值时，置位VSYSIF标志位；如果使能VSYSIE中断，就会响应产生VSYS中断；阈值通过寄存器VDCR的位C\_VSYS[1:0]来设置；
- **BOR\_DET模块**：监测内部电源信号VDD3P3的电压，当电压低或高于设定阈值时，置位BORIF标志位；如果设置了BOR复位，即BORRST=1，将会立即产生BOR复位；如果没有设置BOR复位，即BORRST=0，但使能了BORIE中断，就会响应产生BOR中断；阈值通过寄存器VDCR的位C\_BOR[1:0]来设置；
- **LBOR\_DET模块**：监测内部电源信号VDD3P3的电压，当电压低于或高于阈值1.9V时，产生LBOR掉电复位；
- **POR\_DET模块**：监测内部电源信号VDD3P3的电压，当电压低或高于阈值0.3V时，产生POR上电复位；

### 2.1.5 1.8V 电源

芯片内部通过子模块VREG将3.3V电压调制成1.8V电压，供芯片内部的1.8V工作域使用，并通过引脚VDD1P8输出。在输出引脚VDD1P8上需要外接0.1uF电容，以协助芯片提供稳定的1.8V内部数字电源。

### 2.1.6 模拟电源

芯片通过引脚AVCC给电能计量EMU单元的模拟电路单独提供3.3V电源。

### 2.1.7 Sleep 模式

设置寄存器CLKCFG[1:0]=[11]（带写保护），可控制CPU进入SLEEP模式。只有在VBAT供电时，才能进入SLEEP模式。进入SLEEP模式后，芯片内部会自动关闭产生1.8V的模块。

#### 2.1.7.1 在SLEEP模式下的状况

- 所有CPU的功能关闭，EMU模块关闭；
- RTC模块由外部电池供电继续运行；
- BOR、LBOR模块和VSYS\_DET模块开启，用于检测上电复位和系统电压，其中BOR可控制为SLEEP模式下分时开启（详见TCR寄存器），VSYS\_DET模块可控制在SLEEP模式下分时开启（详见TCR寄存器），以降低功耗；
- WDT默认开启，在SLEEP模式下，WDT计数溢出时，系统会发生WDT复位，但可以配置WDT\_SH=0在sleep模式下关闭WDT（详见WDTCON寄存器）；
- 其它CPU相连接的外设功能模块均默认关闭；

- 进入SLEEP之前，如果配置SUPDC的LCD、TBS模块开启，在进入SLEEP模式后，即可实现LCD静态显示；
- 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SOC和外设的状态，防止漏电；

### 2.1.7.2 SLEEP 模式下的唤醒

在ATT7039进入SLEEP后，可以被INT0，RX0，RX1，PMU，RTC，KEY这6种唤醒源唤醒，唤醒源的使能信号都可以通过软件单独控制：

- 使能唤醒源功能有效；
- 使能相应的子中断有效，但不需要打开IENx中断向量占用的中断使能和总中断使能（EA=1）；
- 使能WAKE\_EN唤醒源；

CPU从SLEEP状态下唤醒等同复位，因不进入中断向量，不会执行中断服务程序，程序从复位地址0000H开始执行。在SLEEP模式下，Reset复位信号是不可被屏蔽的，包括POR，BOR，LBOR，外部RESET PIN上产生的外部复位信号，以及内部的WDT复位信号。当ATT7039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 下的唤醒功能，进入 Sleep 前需进行以下功能配置：**

- (1) **外部中断和红外唤醒：**相应的 PIN 要配置为 INT0，RX0，RX1 功能 PIN（详细见 GPIO 单元），配置 WAKE\_EN 使能相应的唤醒源。当相应 PIN 上出现下降沿并保持不少于 4 个 fosc 的低电平时间，可以实现 CPU 从 SLEEP 下唤醒。
- (2) **按键唤醒：**配置相应的功能 PIN 为 KEY 功能，使能相应的 KEY 功能子中断（仅配置 KEYIE 相应位中断使能），配置 WAKE\_EN 中的 KEY 唤醒使能（KEY\_EN=1）。当出现下降沿并保持不少于 8 个 fosc 的低电平时，可以让 CPU 从 SLEEP 下唤醒。
- (3) **RTC 唤醒：**使能对应 RTC 的子中断源（仅配置 RTCIE 相应位中断使能），配置 WAKE\_EN 中的 RTC 唤醒使能（RTC\_EN =1）。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 下唤醒。
- (4) **电源管理中断唤醒：**在 SUPDC 中设置 PMU 相应的功能有效，使能 PMU 对应的子中断源（仅配置 PMIER 相应位中断使能），配置 WAKE\_EN 中的 PMU 唤醒使能（PMU\_EN =1）。当电源检测超过阈值，可以让 CPU 从 SLEEP 下唤醒。

### 2.1.7.3 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WKR，如果置位来说明确实发生了唤醒复位，并能通过唤醒标志寄存器WAKEIF确定具体的唤醒源，其中：

- 1) WAKEIF bit3位为 1，表示由RTC子中断信号引起了Wake\_UP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中断引起的唤醒，详细见RTC单元。
- 2) WAKEIF bit2位 1，表示是由KEY子中断信号引起了Wake\_UP唤醒。具体的中断信息由KEY的4种中断源确定，可以查询KEYIF对应的位来确认是哪个KEY中断引起的唤醒，详细见KEY单元。
- 3) WAKEIF bit1位 1，表示是由PMU中断信号引起了Wake\_UP唤醒。具体的中断信息由PMU的3种中断源确定，可以查询PMIFR对应的位来确认是BORIF、VSYISIF、LVDINIF，详细见PMU单元。

## 2.1.8 特殊功能寄存器

表 2-1-1 PMU 相关寄存器列表

地址	名称	复位值	功能描述
93H	PMUCFR	0x07	PMU配置寄存器
94H	VDCR	0x19	电压检测控制寄存器
95H	TCR	0x00	测试控制寄存器
96H	PMIFR	0x00	PMU中断标志寄存器
97H	PMIER	0x00	PMU中断使能寄存器
A2H	PMSR	0x00	PMU状态指示寄存器

表 2-1-2 Hold/Sleep SFR 寄存器列表

地址	名称	复位值	功能描述
0x9E	CLKCFG	0x00	系统时钟配置寄存器
0x91	WAKE_EN	0x00	唤醒使能寄存器
0xA7	RSTSR	0x00	复位标志寄存器
0xB7	WAKEIF	0x00	唤醒标志寄存器

表 2-1-3 PMU 配置寄存器 PMUCFR (93H)

PMUCFR		Address: 93H default: 0x67						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	LVDIN_EN	BORRST	BOR_EN
Write:								
Reset:	0	0	0	0	0	1	1	1

位	功能描述
LVDIN_EN	<b>LVDIN_DET 模块使能信号, 监测 LVDIN PIN</b> 1: 开启 LVDIN_DET 模块(default) 0: 关闭 LVDIN_DET 模块
BORRST	<b>BOR复位/中断选择位</b> 1: VDD3P3电压低/高于VDCR[1..0]设定阈值时产生BOR复位(default) 0: VDD3P3电压低/高于设定阈值时产生BOR中断
BOR_EN	<b>BOR 模块使能信号</b> 1: 开启 BOR 模块(default) 0: 关闭 BOR 模块

表 2-1-4 VDCR 电源阈值控制寄存器(94H)

Votag Detect Control		Address: C2H default: 0x09						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	OSC_SLP	CHOP_ADC	C_VSYS1	C_VSYS0	C_BOR1	C_BOR0
Write:				_EN				
Reset:	0	0	0	1	1	0	0	1

位	功能描述
OSC_SLP	OSC 模式选择信号



	1: 大电流模式, OSC 振幅在此模式下增大, 建议 NORMAL 模式下设置为 1 0: 小电流模式, 建议 SLEEP 模式下设置为 0															
CHOP_ADC_EN	<b>ADC Chopper 使能信号</b> 1: 开启 ADC Chopper 模块 0: 关闭 ADC Chopper 模块															
C_VSYS[1:0]	<b>引脚供电电压VSYS检测阈值控制位</b> <table border="1"> <thead> <tr> <th>C_VSYS1</th> <th>C_VSYS0</th> <th>VTH_VSYS(V)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>2.4</td> </tr> <tr> <td>0</td> <td>1</td> <td>2.6</td> </tr> <tr> <td>1</td> <td>0</td> <td>2.8 (DEFAULT)</td> </tr> <tr> <td>1</td> <td>1</td> <td>3.0</td> </tr> </tbody> </table>	C_VSYS1	C_VSYS0	VTH_VSYS(V)	0	0	2.4	0	1	2.6	1	0	2.8 (DEFAULT)	1	1	3.0
C_VSYS1	C_VSYS0	VTH_VSYS(V)														
0	0	2.4														
0	1	2.6														
1	0	2.8 (DEFAULT)														
1	1	3.0														
C_BOR[1:0]	<b>内部工作电压VDD3P3检测阈值控制位</b> <table border="1"> <thead> <tr> <th>C_BOR1</th> <th>C_BOR0</th> <th>BOR检测电压(V)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>2.2</td> </tr> <tr> <td>0</td> <td>1</td> <td>2.4(default)</td> </tr> <tr> <td>1</td> <td>0</td> <td>2.6</td> </tr> <tr> <td>1</td> <td>1</td> <td>2.8</td> </tr> </tbody> </table>	C_BOR1	C_BOR0	BOR检测电压(V)	0	0	2.2	0	1	2.4(default)	1	0	2.6	1	1	2.8
C_BOR1	C_BOR0	BOR检测电压(V)														
0	0	2.2														
0	1	2.4(default)														
1	0	2.6														
1	1	2.8														

表 2-1-5 TCR 内部时序控制寄存器 (95H)

Timer Control Register			Address: 95H					
	Bit7	6	5	4	3	2	1	Bit0
Read	BOR_DIV	X	SeL_AorD	X	TO_VSYS1	TO_VSYS0	T_VSYS1	T_VSYS0
Write								
Reset:	0	0	0	0	0	0	0	0

位	功能描述												
BOR_DIV	<b>在 Sleep mode 下 BOR 分时开启的控制位</b> 1: 分时开启模式 (仅在 Sleep 下有效, 分时开启的检测时间和开启周期由 TO_VSYS0~TO_VSYS1 这两个 BIT 来决定)。 0: sleep mode 下, 如果 Bor-EN 使能, BOR 连续开启												
SeL_AorD	<b>内部数字滤波选择</b> 1: 开启内部数字滤波器, 可实现在高频不稳定的情况下强制锁定 PLL 输出 0: 关闭内部数字滤波器 (建议此位配置为 1)												
TO_VSYS[1:0]	<b>Sleep 模式下 VSYS_DET 分时检测时间设定</b> Sleep 模式, VSYS_DET 分时检测时, 每个周期内 VSYS_DET 工作时间为: <table border="1"> <thead> <tr> <th>TO_VSYS1</th> <th>TO_VSYS0</th> <th>T(US)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>524</td> </tr> <tr> <td>0</td> <td>1</td> <td>1050</td> </tr> <tr> <td>1</td> <td>0</td> <td>1570</td> </tr> </tbody> </table>	TO_VSYS1	TO_VSYS0	T(US)	0	0	524	0	1	1050	1	0	1570
TO_VSYS1	TO_VSYS0	T(US)											
0	0	524											
0	1	1050											
1	0	1570											

		1	1	2100
T_VSYS[1:0]	<b>Sleep 模式下 VSYS_DET 分时检测的周期设定</b>			
	T_VSYS1	T_VSYS0	T (MS)	
	0	0	33	
	0	1	67	
	1	0	134	
	1	1	268	

VSYS\_DET 在sleep mode下采用分时开启的方式工作:

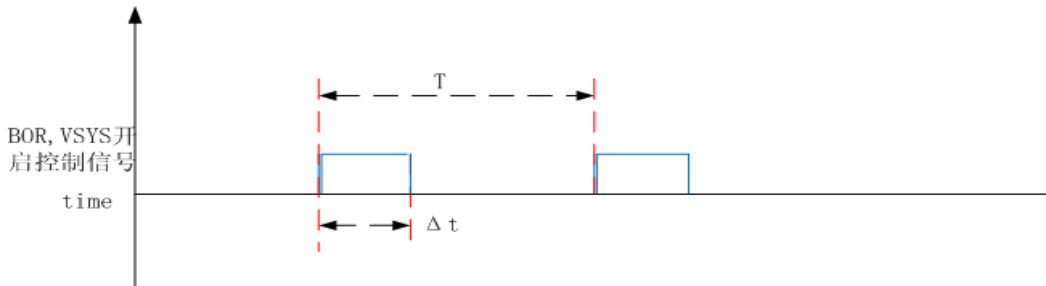


图 2-1-2 VSYS\_DET, BOR\_DET 分时检测时序图

#### VSYS在SLEEP模式下分时开启功能:

其中T是VSYS\_DET分时的周期, 通过TCR寄存器的T0\_VSYS和T1\_VSYS位控制。T和 $\Delta t$ 是分时检测时每个周期内VSYS\_DET工作的时间, 通过TCR寄存器的TO0\_VSYS和TO1\_VSYS位控制。

#### BOR模块在Sleep下的分时开启功能:

PMUCFR[BOR\_EN]置位为1, TCR[BOR\_DIV] 置位为1, BOR检测模块开启使能关闭的周期和检测时间和Vsys保持一致, 检测周期时间T由TCR来设置, 开启的 $\Delta t$ 的时间如上图所示。

表 2-1-6 PMIER PMU 中断使能寄存器(97H)

Power Management Interrupt Enable			Address: 97H					
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	X	reverd	BORIE	VSYSIE	LVDINIE
Write								
Reset:	0	0	0	0	0	0	0	0

注: 需要同时使能 IEN1.bit2, PMIER 使能的中断才有效。

位	功能描述
bit3	保持为 0
BORIE	<b>BOR中断使能位</b> 1: 允许BOR中断 0: 关闭 BOR 中断
VSYSIE	<b>VSYS_DET中断使能位</b> 1: 允许VSYS_DET中断

	0: 关闭 VSYS_DET 中断
LVDINIE	<b>LVDIN_DET中断使能位</b> 1: 允许LVDIN_DET中断 0: 关闭LVDIN_DET中断

表 2-1-7 PMIFR PMU 中断标志寄存器(96H)

Power Management Interrupt Flag		Address: 96H						
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	X	PLLIF	BORIF	VSYSIF	LVDINIF
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
PLLIF	<b>PLL失锁中断标志位</b> 当PLL失锁时该位置1; 软件写0清0;
BORIF	<b>BOR中断标志位</b> 当内部工作电压VDD3P3下降到低于设定阈值或上升到高于设定阈值时, 并且在BORRST=0的情况下, 该位置1; 软件写0清0;
VSYSIF	<b>VSYS_DET中断标志位</b> 当外部引脚VSYS电压下降到低于设定阈值或上升到高于设定阈值时, 该位置1; 软件写0清0;
LVDINIF	<b>LVDIN_DET中断标志位</b> 当外部引脚LVDIN电压下降到低于1.18V或上升到高于1.18V时, 该位置1; 软件写0清0;

表 2-1-8 PMSR PMU 状态寄存器寄存器(A2H)

Power Management Status Flag		Address: A2H						
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	X	X	BORS	VSYS	LVDINS
Write						X	X	X
Reset:	0	0	0	0	0	0	0	0

注: 该寄存器不能被 Wake\_UP 唤醒 RESET。

位	功能描述
BORS	<b>内部工作电压 VDD3P3 状态指示位</b> (只读, 系统默认阈值 2.4V) 1: 表示 VDD3P3 小于设定阈值 (VDCR[1..0]设定阈值) 0: 表示 VDD3P3 大于设定阈值
VSYS	<b>系统电源连接状态</b> (只读, 系统默认检测电压值为 2.8V) 1: 表示系统使用 VBAT 电源 (VDCR[3..2]设定阈值) 0: 表示系统使用 VSYS 电源
LVDINS	<b>外部引脚 LVDIN 电压状态</b> (只读) 1: 表示 LVDIN 小于 1.18V 阈值 0: 表示 LVDIN 大于 1.18V 阈值

表 2-1-9 CLKCFG 系统时钟配置寄存器(9EH)

CLKCFG		Address: 9EH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	W_EN	X	X	X	X	SYSCK	Mode1	Mode0
Write:								
Reset:	0	0	0	0	0	0	0	0

注：此寄存器带 REG0 型写保护，需连续对 BWPM 写入 CFH，BCH 才能改写此寄存器。

位	功能描述																			
W_EN	<b>时钟配置寄存器模式写保护位</b> 如果要改系统的模式，即在向mode1和mode0 位写入新的值时，必须同时将 w_en位置1，该寄存器的其它位不会受到该写操作的影响，w_en位置0时，可以对该寄存器其它位操作。																			
SYSCK	<b>系统时钟选择位</b> 1：表示系统时钟选择PLL倍频后的高频时钟，fcpu=fpll 0：表示系统时钟选择OSC产生的低频时钟，fcpu=fosc																			
Mode[1:0]	<b>系统工作模式选择</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>W_EN</th> <th>Mode 1</th> <th>Mode 0</th> <th>Mode</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0</td> <td rowspan="2">系统产生软件复位</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td><del>±</del></td> <td><del>±</del></td> <td><del>0</del></td> <td style="color: red;">进入HOLD模式</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>进入SLEEP模式</td> </tr> </tbody> </table> <p>注：在系统由VSYs供电时，不允许进入SLEEP MODE，如果在VSYs供电时，对Mode[1..0]写[11]，就会出现软件复位。</p>	W_EN	Mode 1	Mode 0	Mode	1	0	0	系统产生软件复位	1	0	1	<del>±</del>	<del>±</del>	<del>0</del>	进入HOLD模式	1	1	1	进入SLEEP模式
W_EN	Mode 1	Mode 0	Mode																	
1	0	0	系统产生软件复位																	
1	0	1																		
<del>±</del>	<del>±</del>	<del>0</del>	进入HOLD模式																	
1	1	1	进入SLEEP模式																	

表 2-1-10 WAKE\_EN 唤醒使能控制位(91H)

WAKE_EN		Address: 91H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	INT0_EN	X	RX0_EN	RX1_EN	RTC_EN	KEY_EN	PMU_EN	X
Write:								
Reset:	0	0	0	0	0	0	0	0

注：该寄存器只在芯片进入 SLEEP MODE 下唤醒有效，在其它模式下设置对系统不会产生任何影响。

位	功能描述
INT0_EN	<b>外部中断 0 唤醒使能位</b> 1：表示 INT0 信号唤醒功能有效； 0：表示 INT0 信号唤醒功能无效；
RX0_EN	<b>串口中断 0 唤醒使能位</b> 1：表示 RX0 中断信号唤醒功能有效； 0：表示 RX0 中断信号唤醒功能无效；
RX1_EN	<b>串口中断 1 唤醒使能位</b> 1：表示 RX1 中断信号唤醒功能有效； 0：表示 RX1 中断信号唤醒功能无效；

RTC_EN	<b>RTC 中断唤醒使能位</b> 1: 表示 RTC 中断信号唤醒功能有效; 0: 表示 RTC 中断信号唤醒功能无效;
KEY_EN	<b>按键中断唤醒使能位</b> 1: 表示 KEY 中断信号唤醒功能有效; 0: 表示 KEY 中断信号唤醒功能无效;
PMU_EN	<b>电源管理中断唤醒使能位</b> 1: 表示 PMU 中断信号唤醒功能有效; 0: 表示 PMU 中断信号唤醒功能无效;

表 2-1-11 RSTSR 复位标志寄存器(A7H)

RSTSR		Address: A7H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	POR	RST	WDT	BOR	WKR	LBOR	Soft rst	DE_RST
Write:								
Reset:	0	0	0	0	0	0	0	0

注: 只有 POR 复位、LBOR 复位和写 0 操作才能清除该寄存器。更多详细内容参考系统复位章节。当系统处于 SLEEP 下触发唤醒复位时, RSTSR 的 WKR 复位标志位被设置为 1。

表 2-1-12 WAKEIF 唤醒标志寄存器 (B7H)

WAKEIF		Address: B7H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	INT0	X	RX0	RX1	RTC	Key	PMU	X
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
INT0	<b>INT0 唤醒标志</b> INT0 唤醒发生时, 设置标志位为 1, 写零清零。
RX0	<b>RX0 唤醒标志</b> RX0 唤醒发生时, 设置标志位为 1, 写零清零。
RX1	<b>RX1 唤醒标志</b> RX1 唤醒发生时, 设置标志位为 1, 写零清零。
RTC	<b>RTC 中断唤醒标志</b> SLEEP 模式下 RTC 中断发生时将会产生 RTC 唤醒, 此位置为 1, 写零清零。
Key	<b>Key 唤醒标志</b> SLEEP 模式下 Key 上有输入时可唤醒芯片, 此位置为 1, 写零清零。
PMU	<b>PMU/TBS 中断唤醒标志</b> SLEEP 模式下 PMU/TBS 中断发生时将会产生 PMU 唤醒, 唤此位置为 1, 写零清零。

## 2.2 系统复位

### 2.2.1 概述

ATT7039 有 8 种复位方式：其中第一种和第二种为第一优先级，第三种到第五种为第二优先级，第六种到第八种为第三优先级，每个优先级中的复位时间一致。

- 1) 上电复位 (Power On Reset)
- 2) 低电压检测复位 (LBOR)
- 3) 外部引脚 /RST 复位
- 4) 掉电复位 (Brown Out Reset)
- 5) Debug reset
- 6) 看门狗复位
- 7) 软复位
- 8) PDM 唤醒复位

任何复位源产生复位时，同时内部复位信号 IRST 有效，CPU 的程序指针恢复到 0000H。IRST 也会将绝大部分寄存器恢复到缺省值，所有芯片外围单元都将处于复位状态。

- 1) POR, BOR 和 LBOR 会使内部复位信号 IRST 将保持有效，并保持 1024 个 fosc。
- 2) 外部 RST, WDT, Wake up reset 复位时，内部 IRST 信号有效，并保持 64 个 fosc。

### 2.2.2 框图

### 2.2.3 上电复位 (POR)

当电源第一次加到芯片上时，上电复位电路将会产生一个 POR 脉冲，指示发生上电。内部复位信号 IRST 保持为低电平，1024 个 fosc 后，IRST 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
- 内部复位信号 IRST 有效
- 计数 1024 个 f<sub>osc</sub>
- 复位状态寄存器 RSTSR 的上电复位标志位 POR 被设置为 1，其他 RSTSR 为被清为 0。
- CPU 从地址 0000H 执行程序

LBOR 在掉电后重新上电的复位过程与之相同。

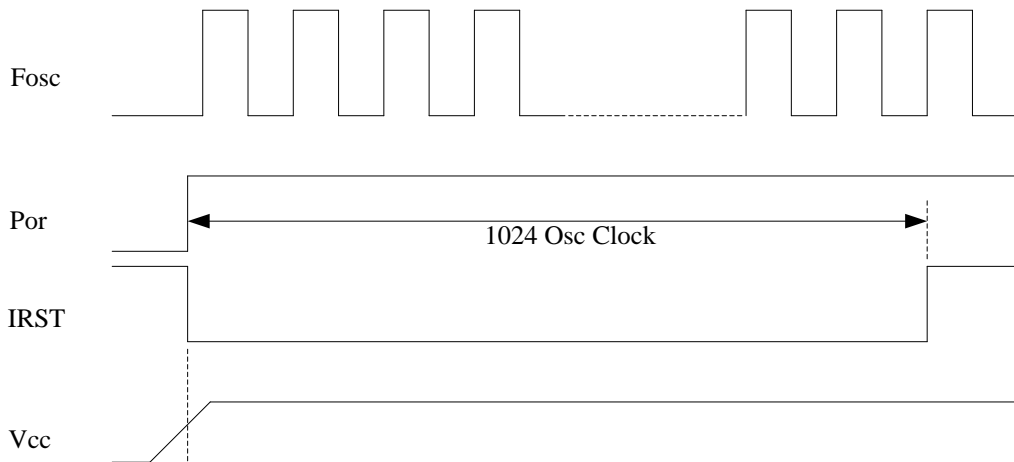


图 2-2-1 上电复位说明

### 2.2.4 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 IRST 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 IRST 有效脉宽为 64 个 fosc。

如果/RST 低电平脉宽比 2us 窄，系统不发生复位。

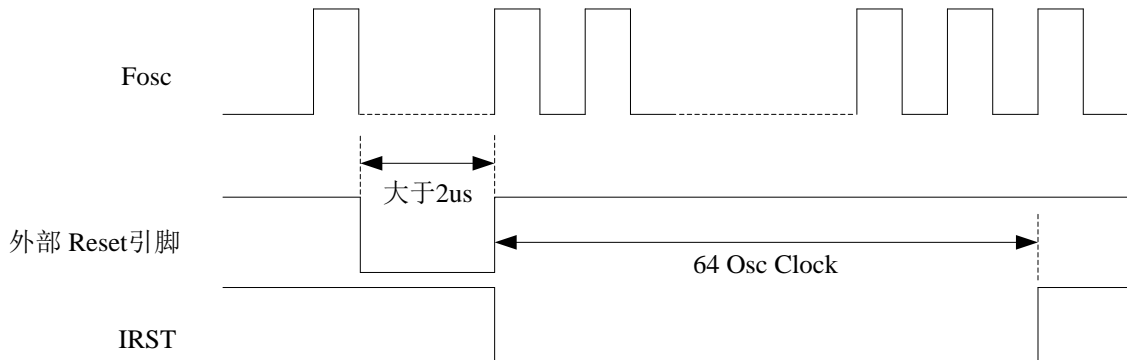


图 2-2-2 /RST 复位

### 2.2.5 掉电复位 (BOR& LBOR)

当掉电检测电路检测到电源电压低于设定电压  $V_{borl}$  时，BOR 输出低电平，内部复位信号 IRST 将变为低电平，复位状态寄存器 RSTSR 的 BOR 标志位被置为 1。当掉电检测电路检测到电源电压高于设定电压  $V_{borh}$  时，BOR 输出高电平，IRST 在 1024 个 fosc 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 IRST 有效
- 计数 1024 个 fosc
- 复位状态寄存器 RSTSR 的掉电复位标志位 BOR 被设置为 1，其他 RSTSR 为被清为 0。
- CPU0000H 开始执行程序

低电压掉电复位 LBOR 与 BOR 的过程基本相同，两者是与的关系。

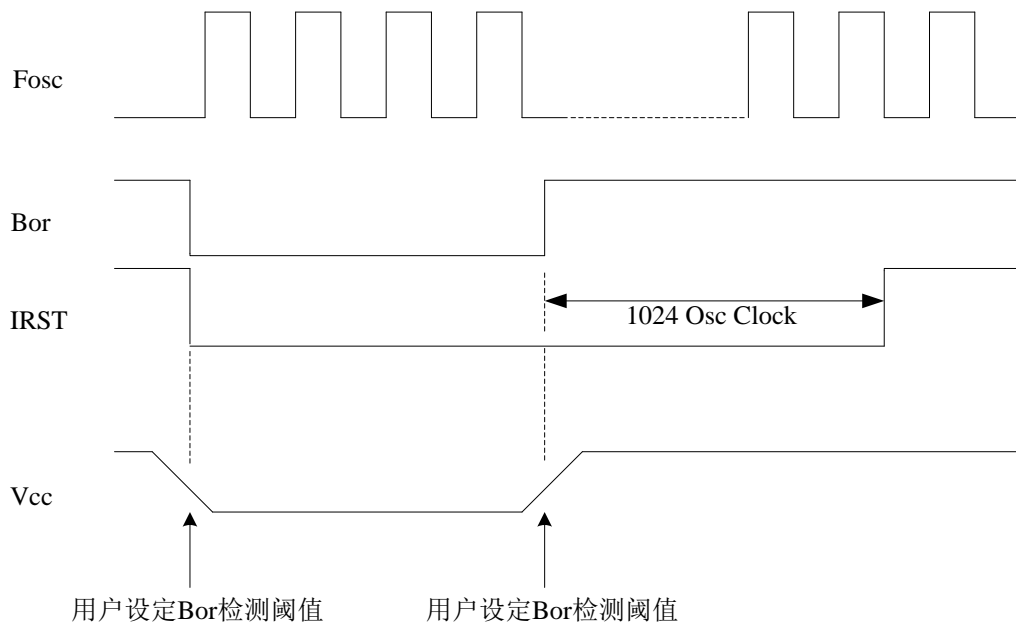


图 2-2-3 Bor 复位

### 2.2.6 电子狗复位

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

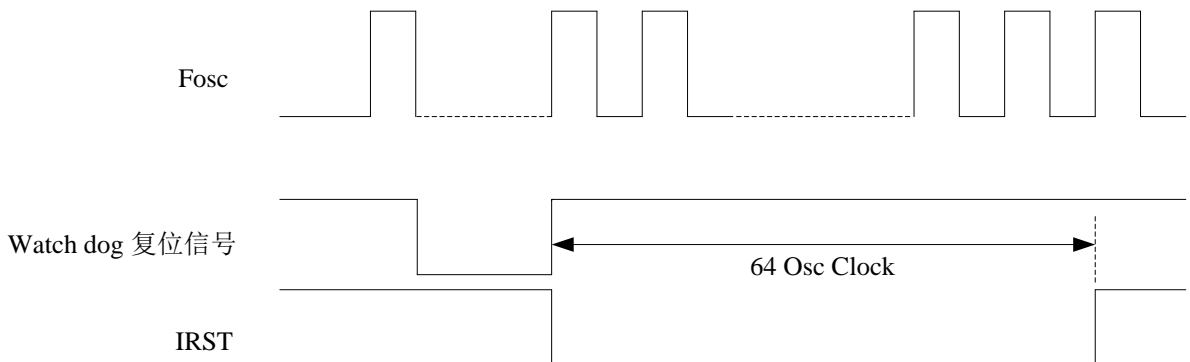


图 2-2-3 WDT 复位

### 2.2.7 唤醒复位

出现 PDM 唤醒事件时, 按照下面顺序执行:

- 内部复位信号 IRST 有效
- 复位状态寄存器 RSTSR 的掉电复位标志位 WKR 被设置为 1
- 计数 64 个 fosc 后, 释放内部复位信号 IRST

### 2.2.8 软复位

在系统由 VSYS 供电时, 不允许进入 SLEEP MODE, 如果 VSYS 供电时, 在打开寄存器写保护的情况下, 对 Mode[1..0]写[11], 就会出现 Soft Reset。



## 2.2.9 特殊功能寄存器

表 2-2-1 复位标志寄存器(0xA7H)

RSTSR			Address: A7H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	POR	RST	WDT	BOR	WKR	LBOR	Soft rst	DE_RST
Write:	x	x	x	x	x	x	x	x
Reset:	0	0	0	0	0	0	0	0

注：只有 POR 复位，LBOR 复位和读操作才能清除该寄存器。

位	功能描述
POR	<b>上电复位标志</b> 1: 发生上电复位，同时将其他复位标志位清零。读这个寄存器可以将这个寄存器包括POR标志清零(写零清零)。 0: 没有发生上电复位；
RST	<b>引脚/RST复位标志</b> 1: RST PIN复位，外部复位引脚/RST出现大于2us的低电平时产生(写零清零)； 0: 没有发生外部引脚/RST复位；
WDT	<b>WDT溢出复位标志</b> 1: WatchDog Timer溢出复位(写零清零)； 0: 没有发生WDT复位；
BOR	<b>掉电复位标志</b> 1: 产生 BOR复位，当VDD3P3低于VDCR[1..0]设定的电压值时，将会立即置位(写零清零)； 0: 没有发生BOR复位；
WKR	<b>PDM复位唤醒标志</b> 1: 当唤醒复位发生时，此位置为1(写零清零)。 0: 没有发生唤醒复位；
LBOR	<b>低掉电复位标志</b> 1: 产生LBOR复位，当VDD3P3低于芯片最低工作电压值1.9V，此位将会立即置为1(写零清零)； 0: 没有发生 LBOR 复位；
Soft rst	<b>当软复位标志</b> 1: 软复位，当对一些寄存器进行了不被允许的操作时，就会产生软复位（如CLKCFG的mode位误操作）。 0: 没有发生软复位；
DE_RST	<b>Debug reset标志位</b> 1: 发生Debug reset； 0: 没有发生 Debug reset；

## 2.2.10 被分级复位的寄存器说明

表 2-2-2 分级复位说明列表

复位级别	复位源	需要置位的	不能复位的
1	POR	复位标志寄存器中的 POR	
	LBOR	复位标志寄存器中的 LBOR 位	1、RTC 的补偿寄存器和年、月、日、星期、时分秒寄存器。
2	外部 RESET	复位标志寄存器中的 RST	2、RTC 的补偿寄存器和年、月、日、星期、时分秒寄存器。 3、BOR、LVREF、VSYS 的 level 寄存器
	BOR	复位标志寄存器中的 BOR 位	1、RTC 的补偿寄存器和年月日、星期、时分秒寄存器 2、BOR、LVREF、VSYS 的 level 寄存器，
	DEBUG RESET	复位标志寄存器中的 DEBUG_RST 位	1、RTC 的补偿寄存器和时年、月、日、星期、时分秒寄存器。 2、BOR、LVREF、VSYS 的 level 寄存器。 3、LCD 相关寄存器。 4、gpio/lcd 复用状态 5、gpio 的输入输出状态和方向控制寄存器 6、SUPDC
3	WDT	复位标志寄存器中的 WDT	1、RTC 的补偿寄存器和时年、月、日、星期、时分秒寄存器。 2、BOR、LVREF、VSYS 的 level 寄存器。 3、LCD 相关寄存器。 4、gpio/lcd 复用状态 5、gpio 的输入输出状态和方向控制寄存器 6、SUPDC 7、KEYIE(AFH), Wake_EN (91H)
	software reset	复位标志寄存器中的 Soft rst 位	1、RTC 的补偿寄存器和年、月、日、星期、时分秒寄存器 2、BOR、LVREF、VSYS 的 level 寄存器 3、LCD 相关寄存器 4、gpio/lcd 复用状态 5、gpio 的输入输出状态和方向控制寄存器 6、SUPDC 7、KEYIE(AFH), Wake_EN (91H)
	PDM 唤醒	复位标志寄存器中的 WKR 位	1、RTC 的补偿寄存器和年、月、日、星期、时分秒寄存器 2、BOR、LVREF、VSYS 的 level 寄存器 3、LCD 相关寄存器 4、gpio/lcd 复用状态 5、gpio 的输入输出状态和方向控制寄存器 6、SUPDC 7、KEYIE(AFH), Wake_EN (91H)

表 2-2-3 分级复位寄存器列表

复位级别	RSTSR复位源	不能被复位的寄存器	
一级	POR LBOR	(所有寄存器均被复位。注: LBOR不能复位RTC时间寄存器)	
二级	RST BOR DEBUG_RST T	RTC	RTCCAL, SECR, MINR, HRR, DAYR, MTHR, YRR, DOWR
		PMU	VDCR
DEBUG_RST 还不能复位OCDS模块			
三级	WDT Soft rst WKR	RTC	RTCCON, RTCCAL, RTCIE, RTCIF, ALMR, ALHR, RTC1CNT, RTC2CNT, SECR, MINR, HRR, DAYR, MTHR, YRR, DOWR
		KEY	KEYIE, KEYIF
		LCD	LCDCLK, LCDCR
		PMU	VDCR, PMIFR, PMSR
		GPIO	LCDCFG, P02CFG, KEYCFG, P3CFG, P0, LEDP0, DDRP0, P1, DDRP1, P2, DDRP2, P3, DDRP3, PTA, DDRA, PTB, DDRB
		CMU	SUPDC, WAKE_EN, RSTSR

注: SUPDC 的位 EMU\_\_EN 会被任何复位源复位

## 2.3 时钟管理

### 2.3.1 概述

- 系统时钟管理模块包含系统时钟生成和系统时钟控制两部分。
- 系统时钟  $f_{sys}$  有两种生成形式：一是低频晶振输出  $f_{osc}$  频率为 32KHz，二是 PLL 输出高频  $f_{pri}$  频率。由时钟配置寄存器 CLKCFG 的 SYSCK 位决定，复位后，系统时钟的缺省值为  $f_{osc}$ 。
- 上电复位后，片上低频振荡电路开始工作，OSC 产生 32.768KHz 的时钟，系统时钟来自片上低频晶振电路  $f_{osc}$ ，此时钟电路一直保持开启；
- 高频时钟频率由 PLL 电路产生，主时钟  $f_{pri}$  可以提供的时钟频率包括 11.010048MHz，5.505024MHz 及其二者的分频。PRION 位作为 PLL 电路使能控制位，当 PRION=1 打开 PLL 时，产生  $f_{pll}$  高频时钟，系统利用 PLL 把 32.768KHz 的时钟倍频到 5.505024MHz 或者 11.010048MHz（由 CHHI 决定）。从 OSC 时钟切换到 PLL 时钟，需要使能 PLL 后等待 2ms，才能切到 PLL 时钟。最终产生的高频时钟  $f_{pri}$  由 PLLCFG 的 PRIP[1:0] 位决定，可以输出 0.68MHz~11 MHz。
- 芯片外围单元 RTC、LCD、WDT、PMU 部分的时钟直接来自低频晶体振荡电路的输出  $f_{osc}$ ，外围单元 I2C 和处理器 R8051XC 的时钟都来自系统时钟  $f_{sys}$ ，即可选择低频时钟  $f_{osc}$ ，也可选择高频时钟  $f_{pri}$ 。电能计量单元 EMU 的时钟来自于  $f_{pll}$  分频后的固定频率 5.505024MHz。红外 38K 模块时钟由  $f_{pll}$  分频提供。
- 外部低频晶体振荡电路是为外部 32.768KHz 的晶体而设计的，OSCI 是晶体振荡电路的输入引脚，OSCO 是晶体振荡电路的输出引脚。上电复位后，外部低频晶体振荡电路开始工作，输出 32.768KHz 时钟，振荡电路的工作不受复位的影响，也不受系统运行模式的影响，外部低频晶体振荡电路提供 RTC 的时钟，也可作为系统节电模式的系统时钟源。
- 当 CPU52 改变系统时钟时，如果改变了 SYSCK 的状态，则必须等待 3 个  $f_{osc}$  以及 3 个  $f_{cpu}$  时钟后才能够正常切换。

### 2.3.2 框图

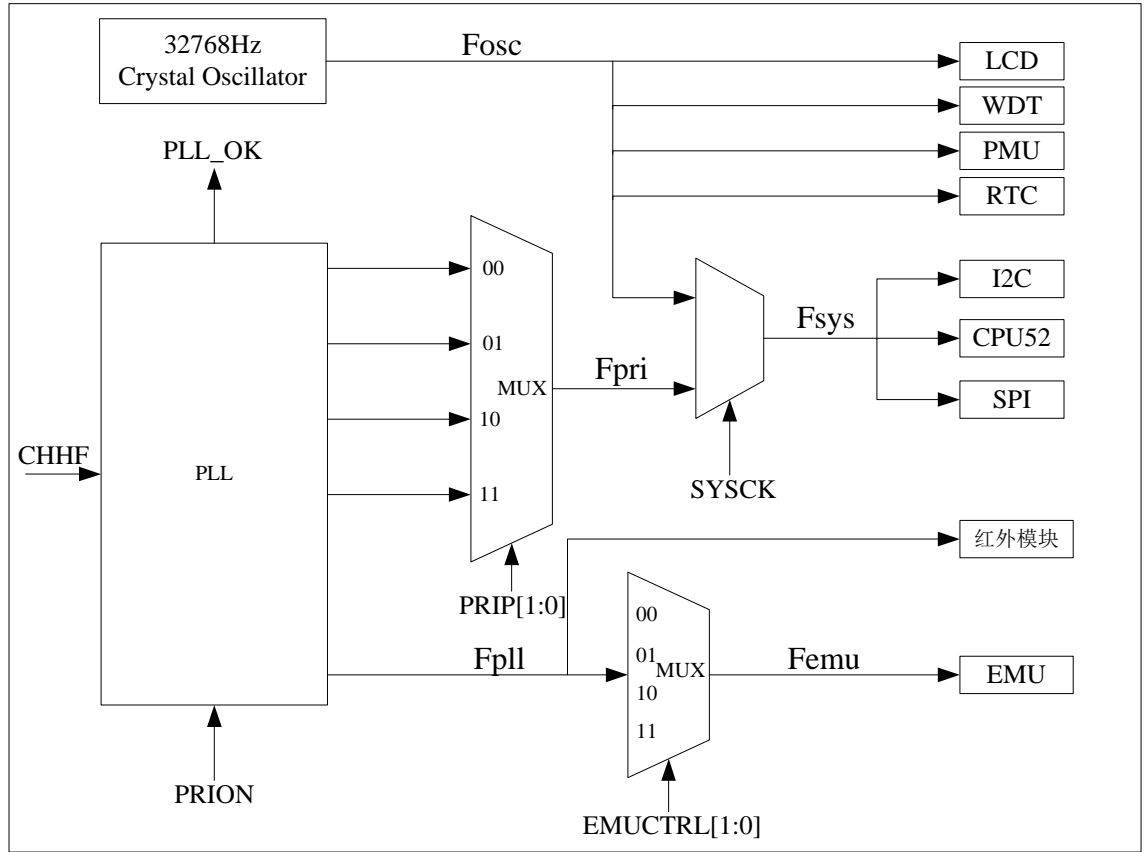


图 2-3-1

### 2.3.3 低频振荡电路

上电复位后，片上低频振荡电路开始工作，OSC 产生 32.768KHz 的时钟，OSCI 是晶体振荡电路的输入引脚，OSCO 是晶体振荡电路的输出引脚，系统时钟来自片上低频晶振电路 fosc，此时钟电路一直保持开启。低频振荡电路的工作不受复位的影响，也不受系统运行模式的影响。外部低频晶体振荡电路提供 RTC 的时钟，也可作为系统节电模式的系统时钟源。芯片外围单元 RTC、LCD、WDT、PMU 部分的时钟直接来自低频晶体振荡电路的输出 fosc。

### 2.3.4 高频振荡电路

高频时钟频率由 PLL 电路产生，主时钟 fpri 可以提供的时钟频率包括 11.010048MHz，5.505024MHz 及其二者的分频。PRION 位作为 PLL 电路使能控制位，当 PRION=1 打开 PLL 时，产生 fp11 高频时钟，系统利用 PLL 把 32.768KHz 的时钟倍频到 5.505024MHz 或者 11.010048MHz（由 CHHI 决定）。从 OSC 时钟切换到 PLL 时钟，需要使能 PLL 后等待 2ms，才能切到 PLL 时钟。最终产生的高频时钟 Fpri 由 PLLCFG 的 PRIP[1:0]位决定，可以输出 0.68MHz~11 MHz。

外围单元 I2C 单元和处理器 R8051XC 的时钟都来自 fsys，推荐应用在高频时钟 fpri。电能计量单元 EMU 的时钟来自于 fp11 分频后的固定频率 5.505024MHz。红外 38K 模块时钟由 fp11 分频提供。

### ● 高频输出功能

当 SYSCK=1 时，系统切换至高频时钟 fsys，fsys 由寄存器 PRIIP [1:0] 和 CHHF 共同决定：

	PRIIP [1:0]	fsys (MHz)		PRIIP [1:0]	fsys (MHz)
	CHHF=1	00		F0= 1.376256	CHHF=0
01		F1= 2.752512	01	F1= 1.376256	
10		F2= 5.505024	10	F2= 2.752512	
11		F3= 11.010148	11	F3= 5.505024	

上电复位后，CHHF=0, SYSCK=0, PRIIP [1:0]=11;

### ● PLL 性能指标

电压范围(V)	2.7~3.6
温度范围(°C)	-40~85°C
输入参考时钟	32.768KHz
输出时钟	11.010048MHZ 或者 5.505024MHZ

## 2.3.5 系统时钟切换

系统的时钟的选择是由CPU通过软件来进行控制的，系统复位后默认是fosc。fsys时钟可以通过软件设置SYSCK=1来选择fp11或其分频。

### 1) 系统时钟 fsys 从低频时钟 fosc 切换到高频时钟 fpri

此操作应避免在高频时钟稳定之前就将系统时钟切换到高频时钟，正确的操作流程是：系统时钟为低频时钟 => 打开写保护寄存器=> 通过选择 PRIIP [1:0]确定高频时钟频率（例如 PRIIP [1:0]=10）=>打开高频时钟电路 PRION=1 => 等待 2ms=>将 SYSCK 置 1，完成 fsys 由 fosc 切换到 fpri。

### 2) 系统时钟 fsys 从高频时钟 fpri 切换到低频时钟 fosc

当系统时钟选择高频时钟，此操作应避免在系统时钟切换到低频时钟之前就将高频时钟关掉，正确的操作流程是：系统时钟为高频时钟 =>打开写保护寄存器=> 将 SYSCK 写为 0，完成 fsys 由 fpri 切换到 fosc。

### 3) 系统时钟 fsys 在高频时钟 fp11 之间切换

此操作直接通过选择 PRIIP [1:0]实现，可实现系统时钟 fsys 在高频时钟 5.5M 和 11M 之间切换。

可以设置 PLLCFG 中的 CHHF 来选择 PLL 输出 5.5MHz 或者 11MHz 的高频时钟。首先应该配置 CLKCFG 中的 SYSCK=0，切换到低频，然后配置 PLLCFG 中的 CHHF 为“1”或者“0”。“CHHF”缺省值为“0”，如果需要 PLL 输出时钟由 5.5MHz 变为 11MHz，CHHF 由“0”变为“1”；或者 PLL 输出时钟由 11MHz 变为 5.5MHz，CHHF 由“1”变为“0”。通过查询 PLLCFG 中的 LOCK 位确认是否切换到相应的高频时钟。

## 2.3.6 时钟管理单元内部保护机制

fsys 的时钟必须在使能PLL后延时2ms、使PLL稳定之后才能由fosc切换至PLL。

- 当 SYSCK=1，表示选择了 fp11 作为系统时钟 fsys，此时不能关闭高频 PLL 电路，也就是说不能将 PRION 清为 0。
- 当 PRION 清为 0 时，不能将 SYSCK 设置为 1，也就是在高频 PLL 电路关闭的情况下不能选择 fp11 作为系统时钟。
- 复位后 PLL 高频电路默认关闭，PRION 置为 0，SYSCK 清为 0，选择 fosc 时钟作为系统时钟。

### 2.3.7 外设时钟管理

EMU 和红外调制单元的时钟直接来自于 PLL 后的固定频率  $f_{pll}=5.505024\text{MHz}$ ，必须先打开高频时钟，即  $PRION=1$ ，才能打开两者的时钟。如果  $CHHF=1$ ，PLL 输出的  $11.010048\text{MHz}$ ，系统会自动二分频提供  $f_{pll}$  为  $5.505024\text{MHz}$ ，如果  $CHHF=0$ ，PLL 输出的  $5.505024\text{MHz}$ ，直接给与  $f_{pll}$ 。

#### 2.3.7.1 EMU 部分的时钟 femu

EMU 的时钟 femu 由高频  $f_{pll}=5.505024\text{MHz}$  输出，直接配置 emu\_ctrl 寄存器的 Emu\_lkctrl[1:0]，得到相应的输出频率 femu；ADC 的采样时钟为 femu 的 6 分频；ADC 波形更新频率由 fadc 进行 32 分频获得，并可通过 ADCCFG 寄存器继续分频为更低的频率。对应关系如下表所示：

Emu_lkctrl [1:0]	femu	系数 N	fadc=femu/N (KHz)	fwave= fadc/32 (KHz)
00	$f_{pll}=5.505024\text{MHz}$	6	917.504	28.672
01	$f_{pll}/2$	12	458.752	14.336
10	$f_{pll}/4$	24	229.376	7.168
11	$f_{pll}/4$	24	229.376	7.168

#### 2.3.7.2 红外输出调制单元的时钟 Firf

当 PLL 打开，即  $PRION=1$  时， $f_{irf}=38.229\text{KHz}$ 。 $f_{pll}$  与  $f_{irf}$  的关系如下表所示

$f_{pll}$ (MHz)	系数 M	$f_{irf}=f_{pll}/M/18$ (KHz)
11.010048	16	38.4
5.505024MHz	8	38.4

### 2.3.8 特殊功能寄存器

表 2-3-2 时钟管理寄存器列表

地址	名称	复位值	功能描述
0x9E	CLKCFG	0x00	系统时钟配置寄存器
0xA1	PLLCFG	0x13	PLL 时钟配置寄存器
0x9F	SUPDC	0x00	内部模块使能控制寄存器
0xB1	PWMCR	0x00	PWM 控制寄存器

表 2-3-3 CLKCFG 系统时钟配置寄存器(9EH)

CLKCFG			Address: 9EH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	W_EN	X	X	X	X	SYSCK	Mode1	Mode0
Write:								
Reset:	0	0	0	0	0	0	0	0

注：此寄存器带 REG0 型写保护，需连续对 BWPM 写入 CFH，BCH 才能改写此寄存器。

位	功能描述																			
W_EN	<b>时钟配置寄存器模式写保护位</b> 如果要改系统的模式，即在向mode1和mode0 位写入新的值时，必须同时将w_en 位置1，该寄存器的其它位不会受到该写操作的影响，w_en位置0时，可以对该寄存器其它位操作。																			
SYSCK	<b>系统时钟选择位</b> 1: 表示系统时钟选择PLL倍频后的高频时钟，fcpu=fppll 0: 表示系统时钟选择OSC产生的低频时钟，fcpu=fosc																			
Mode[1:0]	<b>系统工作模式选择</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>W_EN</th> <th>Mode 1</th> <th>Mode 0</th> <th>Mode</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0</td> <td rowspan="2">系统产生软件复位</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td><del>1</del></td> <td><del>1</del></td> <td><del>0</del></td> <td style="color: red;">进入HOLD模式</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>进入SLEEP模式</td> </tr> </tbody> </table> <p>注：在系统由VSYSD供电时，不允许进入SLEEP MODE，如果在VSYSD供电时，对Mode[1..0]写[11]，就会出现软件复位。</p>	W_EN	Mode 1	Mode 0	Mode	1	0	0	系统产生软件复位	1	0	1	<del>1</del>	<del>1</del>	<del>0</del>	进入HOLD模式	1	1	1	进入SLEEP模式
W_EN	Mode 1	Mode 0	Mode																	
1	0	0	系统产生软件复位																	
1	0	1																		
<del>1</del>	<del>1</del>	<del>0</del>	进入HOLD模式																	
1	1	1	进入SLEEP模式																	

表 2-3-4 PLLCFG PLL 时钟配置寄存器 (A1H)

PLLCFG		Address: A1H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	PRION	reserved	reserved	reserved	X	CHHF	PRIP1	PRIP0
Write:								
Reset:	0	0	0	1	0	0	1	1

注：此寄存器是带 REG0 型写保护寄存器，需连续对 BWPM 写入 CFH, BCH 才能改写此寄存器。

位	功能描述																											
PRION	<b>PLL开启位</b> PRION=0, PLL 关闭。 PRION=1, PLL 打开。																											
reserved	bit[6..4]= [001]，保持默认状态。																											
CHHF	<b>PLL输出高频频率选择位</b> CHHF= “0”，PLL输出的时钟选择为5.505024MHZ CHHF= “1”，PLL输出的时钟选择为11.010048MHZ																											
PRIP [1:0]	<b>PLL输出频率设置</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>CHHF</th> <th>PRIP1</th> <th>PRIP0</th> <th>fppll (MHz)</th> </tr> </thead> <tbody> <tr> <td rowspan="4">1</td> <td>0</td> <td>0</td> <td>F0= 1.376256</td> </tr> <tr> <td>0</td> <td>1</td> <td>F1= 2.752512</td> </tr> <tr> <td>1</td> <td>0</td> <td>F2= 5.505024</td> </tr> <tr> <td>1</td> <td>1</td> <td>F3= 11.010148</td> </tr> <tr> <td rowspan="3">0</td> <td>0</td> <td>0</td> <td>F0= 0.688128</td> </tr> <tr> <td>0</td> <td>1</td> <td>F1= 1.376256</td> </tr> <tr> <td>1</td> <td>0</td> <td>F2= 2.752512</td> </tr> </tbody> </table>	CHHF	PRIP1	PRIP0	fppll (MHz)	1	0	0	F0= 1.376256	0	1	F1= 2.752512	1	0	F2= 5.505024	1	1	F3= 11.010148	0	0	0	F0= 0.688128	0	1	F1= 1.376256	1	0	F2= 2.752512
CHHF	PRIP1	PRIP0	fppll (MHz)																									
1	0	0	F0= 1.376256																									
	0	1	F1= 2.752512																									
	1	0	F2= 5.505024																									
	1	1	F3= 11.010148																									
0	0	0	F0= 0.688128																									
	0	1	F1= 1.376256																									
	1	0	F2= 2.752512																									



		1	1	F3= 5.505024
--	--	---	---	--------------

表 2-3-5 SUPDC 内部模块使能控制寄存器 (9FH)

SUPDC		Address: 9FH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	KBI_EN	X	LCD_EN	EMU_EN	I2C_EN	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
KBI_EN	<b>按键模块时钟使能位</b> 1: 使能 KBI 模块; 0: 关闭 KBI 模块;
LCD_EN	<b>LCD 模块时钟使能位</b> 1: 使能 LCD 模块; 0: 关闭 LCD 模块;
EMU_EN	<b>计量模块时钟使能位</b> 1: 使能 EMU 模块; 0: 关闭 EMU 模块;
I2C_EN	<b>I2C 模块时钟使能位</b> 1: 使能 I2C 模块; 0: 关闭 I2C 模块;

注意: 1, 此寄存器带 REG0 型写保护, 需连续对 BWPM 写入 CFH, BCH 才能改写此寄存器。  
2, 在 Battery 供电的情况下, EMU 固定关闭的, EMU 使能位 EMU\_EN 固定为 0, 写 1 无效。

---

## 3 控制单元 (MCU)

### 3.1 MCU 架构

#### 3.1.1 概述

ATT7039 采用 R8051XC 内核，具有和 8051 兼容的体系架构。

R8051XC 有两条总线：Memory 总线和 SFR 总线。Memory 总线用于在片内扩展程序和数据存储器，如扩展片内 ROM、Flash、XRAM 等。SFR(Special Function Register) 总线用于和片内的外设寄存器接口，除了工作寄存器 R0~R7、程序计数器 (PC) 和指令寄存器 (IR) 外，所有控制、配置和状态寄存器都映射到 SFR 空间，R8051XC 可通过直接寻址的方式访问这些寄存器，控制系统工作。

3.1.2 框图

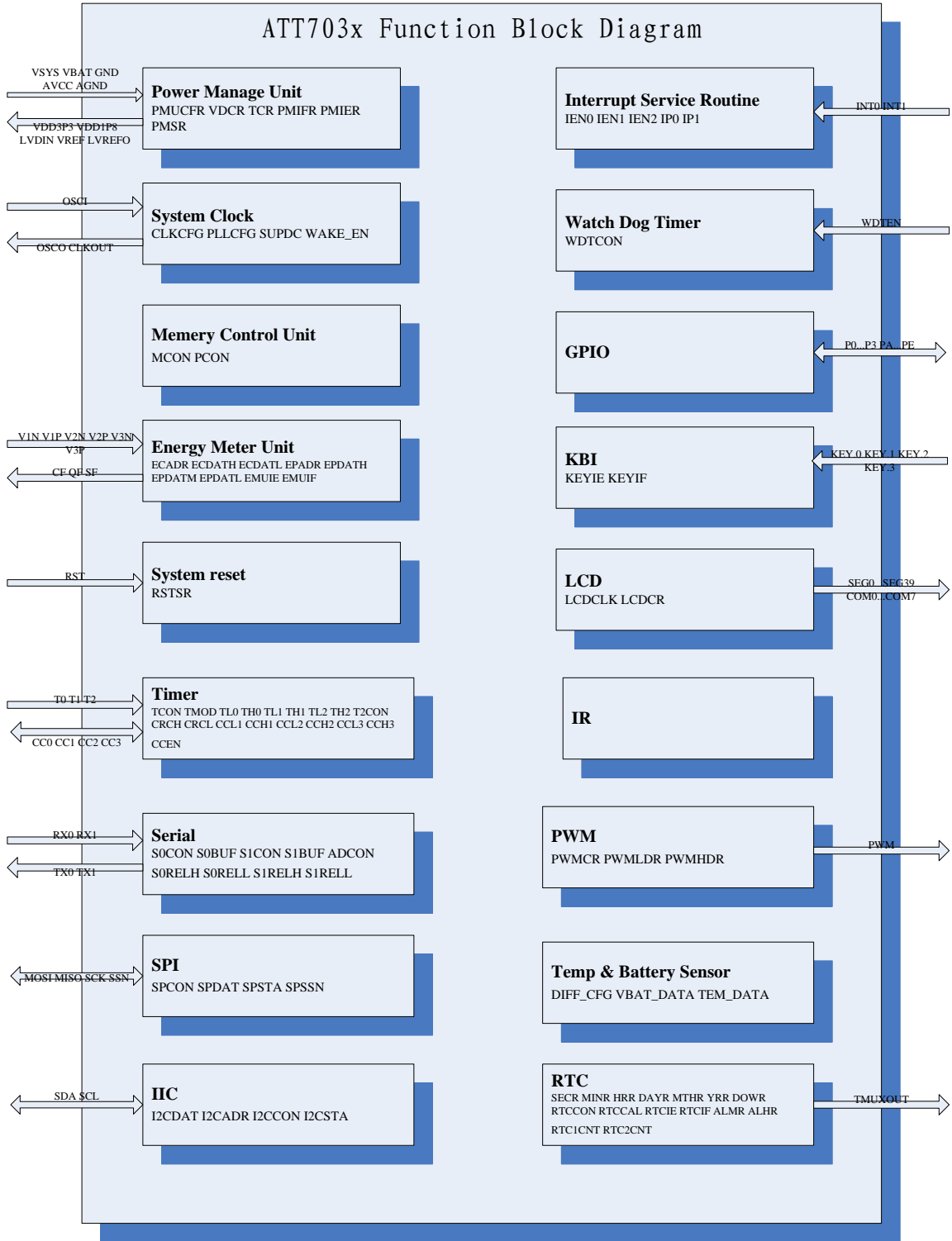


图 3-1-1

注：1、各功能模块内标注有该模块所定义的 SFR 寄存器；  
2、箭头表示与模块相关的引脚。

### 3.1.3 存储器

#### 3.1.3.1 概述

ATT7039 不支持片外扩展存储器，片内存储器逻辑上分为三个地址空间，如图 2-1 所示：

- 程序存储器 (PM)：寻址空间 0000H-FFFFH
- 内部数据存储器 (IRAM)：寻址空间 00H-FFH
- 扩展数据存储器 (DM)：寻址空间 0000H-FFFFH

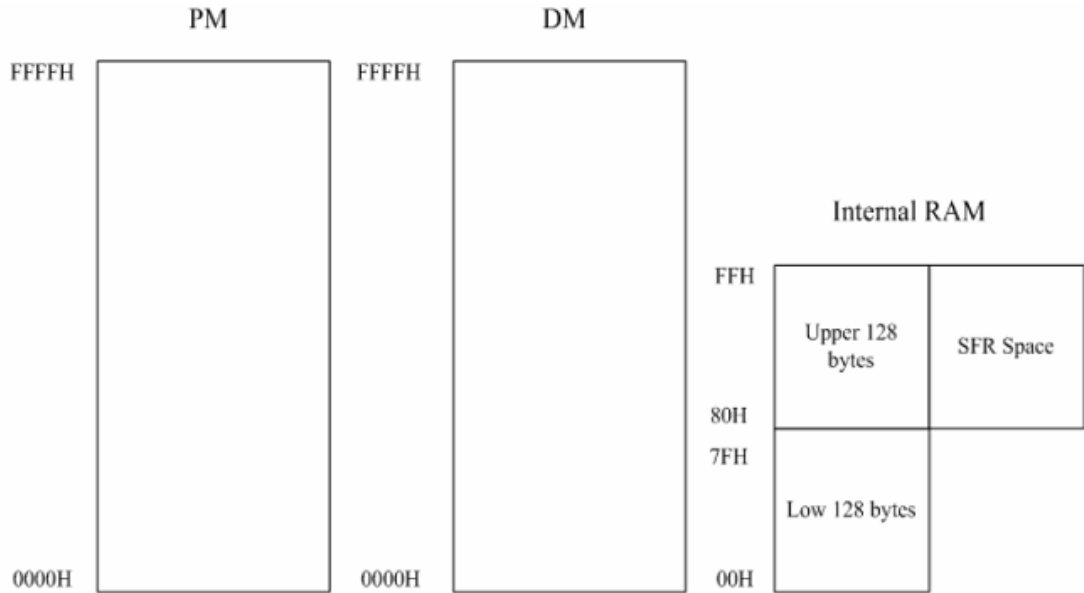


图 3-1-2 ATT7039 存储器地址空间

程序存储器、内部数据存储器 and 扩展数据存储器是分开的，分别有自己的寻址系统、控制信号和功能。程序存储器用来存放程序和一些常量，用 MOV<sub>C</sub> 指令访问；数据存储器用来存放程序运行时需要变量，内部数据存储器用 MOV 指令访问，扩展数据存储器用 MOV<sub>X</sub> 指令访问。

#### 3.1.3.2 内部数据存储器

内部数据存储器是最灵活的地址空间。它分为物理上独立且性质不同的三个区：00H~7FH 单元组成的 128 字节地址空间的 RAM 区；80H~FFH 空间组成的高 128 字节 RAM 区；128 字节地址空间的特殊功能寄存器区。

内部 RAM 区中不同的地址区域功能结构如图 2-2 所示。

其中 00H~1FH 共 32 个字节是四个通用工作寄存器区，每一个区有八个工作寄存器 R0~R7。每个区中 R0~R7 地址见表 2-1。当前程序使用的工作寄存器区是由状态字 PSW (SFR 0xD0H) 中的 RS1 和 RS0 来指示的。CPU 通过对 PSW 中 RS1 和 RS0 位内容的修改，就能任选一个工作寄存器区。这个特点使 CPU 具有快速现场保护的功能。如果用户程序不需要四个工作寄存器区，则不用的工作寄存器区单元可以作一般的 RAM 使用。KeilC 的 C 编译环境中经常使用某一组 R0~R7 传递参数，因此，**需要使用宏命令：**

```
#pragma NOAREGS
```

**防止寄存器组切换，避免出现参数传递错误。**

内部 RAM 的 20H~2FH 为位寻址区。位寻址区的每一位都可以视作软件触发器，由程序直接进行位处理。通常把各种程序状态标志、位控制变量设在位寻址区。同样，位寻址单元也可以作为一般的数据缓冲器使用。

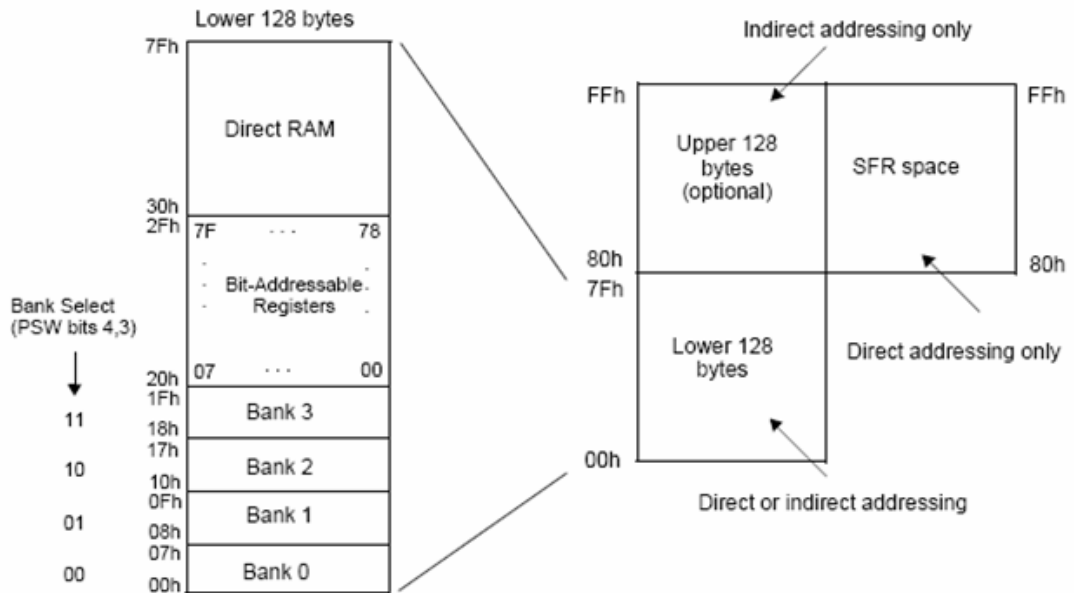


图 3-1-3 内部 RAM 的功能结构

表 3-1-1 内部 RAM 的通用工作寄存器区

0 区		1 区		2 区		3 区	
地址	寄存器	地址	寄存器	地址	寄存器	地址	寄存器
00H	R0	08H	R0	10H	R0	18H	R0
01H	R1	09H	R1	11H	R1	19H	R1
02H	R2	0AH	R2	12H	R2	1AH	R2
03H	R3	0BH	R3	13H	R3	1BH	R3
04H	R4	0CH	R4	14H	R4	1CH	R4
05H	R5	0DH	R5	15H	R5	1DH	R5
06H	R6	0EH	R6	16H	R6	1EH	R6
07H	R7	0FH	R7	17H	R7	1FH	R7

在实际的程序中需要堆栈以保存 CPU 的现场，堆栈原则上可以设在内部 RAM 的任意区域内，但一般设在 30H~FFH 的范围内。栈顶的位置由栈指针 SP 指出。

### 3.1.4 指令系统

#### 3.1.4.1 寻址方式

ATT7039 寻址方式和标准 8052 相同，有下列几种：

##### 1. 立即寻址

操作数包含在指令字节中，指令操作码后面字节的内容就是操作数本身，其数值由程序员在编制程序时指定。例如：

```
MOV A, #70H ; 70H->A
```

## 2. 直接寻址

在指令中含有操作数的直接地址，该地址指出了参与操作的数据所在的字节地址或位地址。直接寻址方式中的操作数存储的空间有三种：

- 1) 内部数据存储器 (IRAM) 的低 128 字节 (00H-7FH), 例如

```
MOV A, 70H ; (70H)→A
```

指令功能是把内部 RAM 70H 单元的内容送入累加器 A

- 2) 位地址空间, 例如

```
MOV C, 00H
```

- 3) SFR

SFR 只能用直接寻址的方式进行访问。例如：

```
MOV IE, #85H ; 立即数 85H→中断允许寄存器 IE
```

## 3. 寄存器寻址

由指令指出某一个寄存器的内容作为操作数。在这种寻址方式中，指令的操作码中包含了参加操作的寄存器的编号，寄存器寻址的工作寄存器是 R0-R7。例如：

```
INC R0 ; (R0)+1→R0, 对寄存器 R0 进行操作, 使其内容加 1
```

## 4. 寄存器间接寻址

由指令指出某一个寄存器的内容作为操作数的地址。在寄存器间接寻址中，存放在寄存器中的内容不是操作数，而是操作数所在的存储器单元的地址。

寄存器间接寻址只能用寄存器 R0 和 R1 作为地址指针来寻址内部 RAM(00H~FFH)中的数据。寄存器间接寻址也适用于访问外部 RAM，可使用 R0、R1 或者 DPTR 作为地址指针。寄存器间接寻址用符号“@”来表示。例如：

```
MOV A, @R0 ; (R0)→A
```

指令功能是把 R0 所指出的内部 RAM 单元中的内容送入累加器 A。

## 5. 基寄存器加变址寄存器间接寻址

这种寻址方式用于访问程序存储器中的数据表格，它把基寄存器 (DPTR 或 PC) 和变址寄存器 (A) 的内容作为无符号数相加形成 16 位地址，访问程序存储器中的数据表格。例如：

```
MOVC A, @A+DPTR ; 把 A+DPTR 所指的程序存储单元的内容→A
```

```
MOVC A, @A+PC ; 把 A+PC 所指的程序存储单元的内容→A
```

## 6. 相对寻址

这种寻址方式是以 PC 的内容作为基地址，加上指令中给定的偏移量所得的结果作为转移地址，它只适用于双字节转移指令。偏移量是带符号数，在+127~128 范围内，用 2 的补码表示。例如：

```
JC rel ; C=1, 跳转
```

### 3.1.4.2 指令集

ATT7039 指令集和标准工业 8052 指令集兼容，这种兼容性表现在指令的操作码、功能以及指令运行对标志位的影响相同。表 2-4 分类列出了 ATT7039 的指令集、字节数、机器周期等。

表 3-1-2 ATT7039 指令集列表符号说明

符号	功能
A	累加器
(A)	累加器内容
Rn	工作寄存器 R0-R7
(Rn)	工作寄存器的内容
Ri	i=0, 1, 数据指针 R0 或 R1
(Ri)	R0 或 R1 的内容
((Ri))	R0 或 R1 的指出的单元内容
@Ri	R0 或者 R1 指针指向的内部寄存器 (除了 MOVX 指令)
(X)	某一寄存器的内容
X	某一寄存器
((X))	某一寄存器指出的单元内容
direct	直接地址单元
(direct)	直接地址指出的单元内容
rel	相对偏移量, 带符号的 (2 的补码) 8 位偏移字节
bit	位地址
#data	8 位立即数
#data 16	16 位立即数
addr 16	16 位绝对地址
addr 11	页面地址
←	数据传送方向
∧	逻辑与
∨	逻辑或
⊕	逻辑异或
✓	对标志位产生影响
×	对标志位不产生影响

表 3-1-3 7039 指令集

助记符	功能	对标志的影响				字节数	机器周期	16 进制代码
		P	OV	Ac	CY			
<b>算术运算指令</b>								
ADD A, Rn	$A \leftarrow (A) + (Rn)$	✓	✓	✓	✓	1	2	28-2F
ADD A, direct	$A \leftarrow (A) + (\text{direct})$	✓	✓	✓	✓	2	3	25
ADD A, @Ri	$A \leftarrow (A) + ((Ri))$	✓	✓	✓	✓	1	4	26-27
ADD A, #data	$A \leftarrow (A) + \text{data}$	✓	✓	✓	✓	2	2	24
ADDC A, Rn	$A \leftarrow (A) + (Rn) + (CY)$	✓	✓	✓	✓	1	2	38-3F
ADDC A, direct	$A \leftarrow (A) + (\text{direct}) + (CY)$	✓	✓	✓	✓	2	3	35
ADDC A, @Ri	$A \leftarrow (A) + ((Ri)) + (CY)$	✓	✓	✓	✓	1	4	36-37
ADDC A, #data	$A \leftarrow (A) + \text{data} + (CY)$	✓	✓	✓	✓	2	2	34

SUBB A, Rn	$A \leftarrow (A) - (Rn) - (CY)$	√	√	√	√	1	2	98-9F
SUBB A, direct	$A \leftarrow (A) - (direct) - (CY)$	√	√	√	√	2	2	95
SUBB A, @Ri	$A \leftarrow (A) - ((Ri)) - (CY)$	√	√	√	√	1	4	96-97
SUBB A, #data	$A \leftarrow (A) - data - (CY)$	√	√	√	√	2	2	94
INC A	$A \leftarrow (A) + 1$	√	×	×	×	1	1	04
INC Rn	$Rn \leftarrow Rn + 1$	×	×	×	×	1	3	08-0F
INC direct	$direct \leftarrow direct + 1$	×	×	×	×	2	4	05
INC @Ri	$(Ri) \leftarrow ((Ri)) + 1$	×	×	×	×	1	5	06-07
INC DPTR	$DPTR \leftarrow (DPTR) + 1$					1	1	A3
DEC A	$A \leftarrow (A) - 1$	√	×	×	×	1	1	14
DEC Rn	$A \leftarrow (Rn) - 1$	×	×	×	×	2	3	18-1F
DEC direct	$A \leftarrow (direct) - 1$	×	×	×	×	1	4	15
DEC @Ri	$A \leftarrow ((Ri)) - 1$	×	×	×	×	1	5	16-17
MUL AB	Multiply A and B					1	4	A4
DIV	Divide A by B					1	4	84
DA A	对 A 进行十进制调整	√	√	√	√	1	1	D4
ANL A, Rn	$A \leftarrow (A) \wedge (Rn)$	√	×	×	×	1	2	58-5F
ANL A, direct	$A \leftarrow (A) \wedge (direct)$	√	×	×	×	2	3	55
ANL A, @Ri	$A \leftarrow (A) \wedge ((Rn))$	√	×	×	×	1	4	56-57
ANL A, #data	$A \leftarrow (A) \wedge data$	√	×	×	×	2	2	54
ANL direct, A	$direct \leftarrow (direct) \wedge A$	×	×	×	×	2	4	52
ANL direct, #data	$direct \leftarrow (direct) \wedge data$	×	×	×	×	3	4	53
ORL A, Rn	$A \leftarrow (A) \vee (Rn)$	√	×	×	×	1	2	48-4F
ORL A, direct	$A \leftarrow (A) \vee (direct)$	√	×	×	×	2	3	45
ORL A, @Ri	$A \leftarrow (A) \vee ((Ri))$	√	×	×	×	1	4	46-47
ORL A, #data	$A \leftarrow (A) \vee data$	√	×	×	×	2	2	44
ORL direct, A	$direct \leftarrow (direct) \vee A$	×	×	×	×	2	4	42
ORL direct, #data	$direct \leftarrow (direct) \vee data$	×	×	×	×	3	4	43
XRL A, Rn	$A \leftarrow (A) \oplus (Rn)$	√	×	×	×	1	2	68-6F
XRL A, direct	$A \leftarrow (A) \oplus (direct)$	√	×	×	×	2	3	65
XRL A, @Ri	$A \leftarrow (A) \oplus ((Ri))$	√	×	×	×	1	4	66-67
XRL A, #data	$A \leftarrow (A) \oplus data$	√	×	×	×	2	2	64
XRL direct, A	$direct \leftarrow (direct) \oplus A$	×	×	×	×	2	4	62
XRL direct, #data	$direct \leftarrow (direct) \oplus data$	×	×	×	×	3	4	63
CLR A	$A \leftarrow 0$	√	×	×	×	1	1	E4
CPL A	$A \leftarrow \overline{(A)}$	×	×	×	×	1	1	F4
SWAP A	A 半字节交换	×	×	×	×	1	1	C4
RL A	A 循环左移一位	×	×	×	×	1	1	23
RLC A	A 带进位循环左移一位	√	×	×	√	1	1	33
RR A	A 循环右移一位	×	×	×	×	1	1	03



RRC A	A 带进位循环右移一位	√	×	×	√	1	1	13
MOV A, Rn	$A \leftarrow (Rn)$	√	×	×	×	1	1	E8-EF
MOV A, direct	$A \leftarrow (\text{direct})$	√	×	×	×	2	3	E5
MOV A, @Ri	$A \leftarrow ((Ri))$	√	×	×	×	1	4	E6-E7
MOV A, #data	$A \leftarrow \text{data}$	√	×	×	×	2	2	74
MOV Rn, A	$Rn \leftarrow (A)$	×	×	×	×	1	1	F8-FF
MOV Rn, direct	$Rn \leftarrow (\text{direct})$	×	×	×	×	2	4	A8-AF
MOV Rn, #data	$Rn \leftarrow \text{data}$	×	×	×	×	2	2	78-7F
MOV direct, A	$\text{direct} \leftarrow (A)$	×	×	×	×	2	2	F5
MOV direct, Rn	$\text{direct} \leftarrow (Rn)$	×	×	×	×	2	3	88-8F
MOV direct, direct	$\text{direct} \leftarrow (\text{direct})$	×	×	×	×	3	4	85
MOV direct, @Ri	$\text{direct} \leftarrow ((Ri))$	×	×	×	×	2	5	86-87
MOV direct, #data	$\text{direct} \leftarrow \text{data}$	×	×	×	×	3	3	75
MOV @Ri, A	$((Ri)) \leftarrow (A)$	×	×	×	×	1	3	F6-F7
MOV @Ri, direct	$(Ri) \leftarrow (\text{direct})$	×	×	×	×	2	4	A6-A7
MOV @Ri, #data	$(Ri) \leftarrow \text{data}$	×	×	×	×	2	3	76-77
MOV DPTR, #data	$DPTR \leftarrow \text{data}$	×	×	×	×	3	3	90
MOVC A, @A+DPTR	$A \leftarrow ((A) + (DPTR))$	√	×	×	×	1	4	93
MOVC A, @A+PC	$A \leftarrow ((A) + (PC))$	√	×	×	×	1	4	83
MOVX A, @Ri	$A \leftarrow ((Ri) + P2)$	√	×	×	×	1	5-1 2	E2-E3
MOVX A, @DPTR	$A \leftarrow ((DPTR))$	√	×	×	×	1	4-1 1	E0
MOVX @Ri, A	$((Ri) + P2) \leftarrow A$	×	×	×	×	1	6-1 3	F2-F3
MOVX A, @DPTR	$A \leftarrow ((DPTR))$	×	×	×	×	1	5-1 2	F0
PUSH direct	$SP \leftarrow SP + 1, (SP) \leftarrow (\text{direct})$	×	×	×	×	2	4	C0
POP direct	$\text{direct} \leftarrow ((SP)), SP \leftarrow (SP) - 1$	×	×	×	×	2	3	D0
XCH A, Rn	$(A) \leftrightarrow (Rn)$	√	×	×	×	1	2	C8-CF
XCH A, direct	$(A) \leftrightarrow (\text{direct})$	√	×	×	×	2	3	C5
XCH A, @Ri	$(A) \leftrightarrow ((Ri))$	√	×	×	×	1	4	C6-C7
XCHD A, @Ri	$(A)_{0 \sim 3} \leftrightarrow ((Ri))_{0 \sim 3}$	√	×	×	×	1	5	D6-D7
CLR C	$CY \leftarrow 0$	×	×	×	√	1	1	C3
CLR bit	$\text{bit} \leftarrow 0$	×	×	×		2	4	C2
SETB C	$CY \leftarrow 1$	×	×	×	√	1	1	D3
SETB bit	$\text{bit} \leftarrow 1$	×	×	×		2	4	D2
CPL C	$CY \leftarrow \overline{(CY)}$	×	×	×	√	1	1	B3

CPL bit	$\text{bit} \leftarrow \overline{(\text{bit})}$	×	×	×		2	4	B2
ANL C, bit	$\text{CY} \leftarrow (\text{CY}) \wedge (\text{bit})$	×	×	×	√	2	3	82
ANL C, /bit	$\text{CY} \leftarrow (\text{CY}) \wedge \overline{(\text{bit})}$	×	×	×	√	2	3	B0
ORL C, bit	$\text{CY} \leftarrow (\text{CY}) \vee (\text{bit})$	×	×	×	√	2	3	72
ORL C, /bit	$\text{CY} \leftarrow (\text{CY}) \vee \overline{(\text{bit})}$	×	×	×	√	2	3	A0
MOV C, bit	$\text{CY} \leftarrow (\text{bit})$	×	×	×	√	2	3	A2
MOV bit, C	$\text{bit} \leftarrow (\text{CY})$	×	×	×	×	2	4	92
ACALL addr11	$\text{PC} \leftarrow (\text{PC}) + 2$ , $\text{SP} \leftarrow (\text{SP}) + 1$ , $(\text{SP}) \leftarrow (\text{PC})\text{L}$ , $\text{SP} \leftarrow (\text{SP}) + 1$ , $(\text{SP}) \leftarrow (\text{PC})\text{H}$ , $\text{PC}10 \sim 0 \leftarrow \text{addr}11$	×	×	×	×	2	4	11-F1
LCALL addr16	$\text{PC} \leftarrow (\text{PC}) + 2$ , $\text{SP} \leftarrow (\text{SP}) + 1$ , $(\text{SP}) \leftarrow (\text{PC})\text{L}$ , $\text{SP} \leftarrow (\text{SP}) + 1$ , $(\text{SP}) \leftarrow (\text{PC})\text{H}$ , $\text{PC}10 \sim 0 \leftarrow \text{addr}16$	×	×	×	×	3	4	12
RET	$(\text{PC})\text{H} \leftarrow ((\text{SP}))$ , $\text{SP} \leftarrow (\text{SP}) - 1$ , $(\text{PC})\text{L} \leftarrow ((\text{SP}))$ , $\text{SP} \leftarrow (\text{SP}) - 1$	×	×	×	×	1	5	22
RETI	$(\text{PC})\text{H} \leftarrow ((\text{SP}))$ , $\text{SP} \leftarrow (\text{SP}) - 1$ , $(\text{PC})\text{L} \leftarrow ((\text{SP}))$ , $\text{SP} \leftarrow (\text{SP}) - 1$ , 从中断中返回	×	×	×	×	2	5	32
AJMP addr11	$\text{PC}10 \sim 0 \leftarrow \text{addr}11$	×	×	×	×	3	3	01-E1
LJMP addr16	$\text{PC} \leftarrow \text{addr}16$	×	×	×	×	2	4	02
SJMP rel	$\text{PC} \leftarrow \text{PC} + \text{rel}$	×	×	×	×	2	3	80
JC rel	$\text{PC} \leftarrow \text{PC} + 2$ , 若 $\text{CY} = 1$ , 则 $\text{PC} \leftarrow \text{PC} + \text{rel}$	×	×	×	×	2	3	40
JNC rel	$\text{PC} \leftarrow \text{PC} + 2$ , 若 $\text{CY} = 0$ , 则 $\text{PC} \leftarrow \text{PC} + \text{rel}$	×	×	×	×	2	3	50
JB bit, rel	$\text{PC} \leftarrow \text{PC} + 3$ , 若 $(\text{bit}) = 1$ , 则 $\text{PC} \leftarrow \text{PC} + \text{rel}$	×	×	×	×	3	3	20
JNB bit, rel	$\text{PC} \leftarrow \text{PC} + 3$ , 若 $(\text{bit}) = 0$ , 则 $\text{PC} \leftarrow \text{PC} + \text{rel}$	×	×	×	×	3	3	30
JBC bit, rel	$\text{PC} \leftarrow \text{PC} + 3$ , 若 $(\text{bit}) = 1$ , 则 $\text{bit} \leftarrow 0$ , $\text{PC} \leftarrow (\text{PC}) + \text{rel}$	×	×	×	×	3	3	10
JMP @A+DPTR	$\text{PC} \leftarrow (\text{A}) + (\text{DPTR})$	×	×	×	×	1	5	73
JZ rel	$\text{PC} \leftarrow \text{PC} + 2$ , 若 $(\text{A}) = 0$ , $\text{PC} \leftarrow (\text{PC}) + \text{rel}$	×	×	×	×	2	5	60
JNZ rel	$\text{PC} \leftarrow \text{PC} + 2$ , 若 $(\text{A}) \neq 0$ , $\text{PC} \leftarrow (\text{PC}) + \text{rel}$	×	×	×	×	2	5	70
CJNE A, direct, rel	$\text{PC} \leftarrow \text{PC} + 3$ , 若 $(\text{A}) \neq$ $(\text{direct})$ , 则 $\text{PC} \leftarrow (\text{PC}) + \text{rel}$	×	×	×	×	3	5	B5

CJNE A, #d, rel	PC← PC+3, 若(A)不等于 data, 则 PC← (PC)+rel	×	×	×	×	3	4	B4
CJNE Rn, #d, rel	PC← PC+3, 若(Rn)不等于 data, 则 PC← (PC)+rel	×	×	×	×	3	4	B8-BF
CJNE @Ri, #d, rel	PC← PC+3, 若((Ri))不等于 d, 则 PC← (PC)+rel	×	×	×	×	3	6	B6-B7
DJNZ Rn, rel	PC← PC+2, Rn =(Rn)-1, 若(Rn)不等于 0, 则 PC← (PC)+rel	×	×	×	×	2	4	D8-DF
DJNZ direct, rel	PC← PC+2, direct =(direct)-1, 若(direct)不等于 0, 则 PC← (PC)+rel	×	×	×	×	3	5	D5
其他								
NOP	空操作	×	×	×	×	1	1	00

### 3.1.5 特殊功能寄存器

#### 3.1.5.1 概述

除了工作寄存器(R0~R7), 程序计数器(PC)和指令寄存器(IR)外, ATT7039 所有控制、配置和数据寄存器都以特殊功能寄存器(SFR)的形式出现, 它们在 SFR 空间 80H~FFH 范围内。SFR 表中 SFR bit 栏中包含“0”或者“1”的位写入值无效; 包含“-”的位没有定义位名称; 包含“\*”的位无定义, 用户不要对这些寄存器进行操作。表中黑色阴影部分标记的寄存器是 ATT7039 的 R8051XC 内部寄存器, 其他寄存器 ATT7039 的扩展寄存器。

Hex/ Bin	X000	X001	X010	X011	X100	X101	X110	X111	Bin/Hex
F8	RTCCON	RTCCAL	RTC1CNT	RTC2CNT	SECR	MINR	HRR	DAYR	FF
F0	B	ALMR	ALHR	MTHR	YRR	DOWR			F7
E8		ECADR	ECDATH	ECDATL	EPADR	EPDATH	EPDATM	EPDATL	EF
E0	ACC								E7
D8	ADCON	DDRP3	I2CDAT	I2CADR	I2CCON	I2CSTA			DF
D0	PSW	DDRA	DDRB			DDRP0	DDRP1	DDRP2	D7
C8	T2CON	WDTCON	CRCL	CRCH	TL2	TH2	P02CFG	P3CFG	CF
C0	IRCON	LEDP0			LCDCLK	LCDCR	LCDCFG	KEYCFG	C7
B8	IEN1	IP1	S0RELH	S1RELH	PTA	PTB			BF
B0	P3	PWMCR			RTCIF	EMUIF	KEYIF	WAKEIF	B7
A8	IEN0	IP0	S0RELL		EMUIE		RTCIE	KEYIE	AF
A0	P2	PLLCFG	PMSR				BWPR	RSTSR	A7
98	S0CON	S0BUF	IEN2	S1CON	S1BUF	S1RELL	CLKCFG	SUPDC	9F
90	P1	WAKE_EN	DPS	PMUCFR	VDCR	TCR	PMIFR	PMIER	97
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON		8F
80	P0	SP	DPL	DPH	DPL1	DPH1	MCON	PCON	87

### 3.1.5.2 寄存器写保护

表 3-1-4 Write protect Register (BWPR, 0xA6H)

Bit Write Protect Register (BWPR)		Address: A6H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	PASS7	PASS6	PASS5	PASS4	PASS3	PASS2	PASS1	PASS0
Write:								
Reset:	0	0	0	0	0	0	0	0

位写保护模式防止软件直接修改受密码保护的寄存器位。

#### 1) 受写保护的寄存器:

寄存器写保护分为两种类型: REG0 和 REG1。

受 REG0 类型写保护的寄存器有: CLKCFG, SUPDC, PLLCFG。

受 REG1 类型写保护的寄存器有: LCDCFG, P02CFG, KEYCFG, P3CFG, PECFG, RTCCAL, SECR, MINR, HRR, DAYR, MTHR, YRR, DOWR。

#### 2) 配置受保护寄存器的方法:

两种类型寄存器写保护操作类似, 都必须连续两次对 BWPR 写入密码, 才能对相应的寄存器进行操作, 只是设置的保护密码不一样。

REG0 的保护密码是先写 0xCF; 再写 0xBC。

REG1 的保护密码是先写 0xCF; 再写 0xDC。

对 BWPR 的两次写操作必须连续, 才能写被保护的寄存器, 否则 BWPR 清零, 写 REG0, REG1 无效。

写保护开启后, 如果没有写 0000000B 将写使能关闭, 那么在写使能开启后运行 30 条 CPU 指令后, 写使能自动被关闭, BWPR 将被复位为 00H。在写使能自动关闭之前如果再次将写使能开启, 那么可以重新计数运行 30 条 CPU 指令。如果进入 HOLD 或者 SLEEP 后, 该写保护寄存器使能自动关闭。

需要注意的是, 在线调试的时候, 如果对受写保护的寄存器赋值语句(包含打开写使能的语句)单步执行, 则两条语句间可能会超过 30 条 CPU 指令的时间, 从而使写使能自动关闭, 造成该赋值语句无效。

#### 3) 配置受保护寄存器示例程序:

例 1: 写 REG0 (CLKCFG)

```

CLR EA;
MOV BWPR , 11001111B; //可以写 Password
MOV BWPR , 10111100B; //OPEN Write Pro
写 REG0;
MOV BWPR , 00000000B; // CLOSE Write Pro
SETB EA;
    
```

例 2: 写 REG1 (RTC)

```

CLR EA;
MOV BWPR , 11001111B; //可以写 Password
MOV BWPR , 11011100B; //OPEN Write Pro
写 REG1;
MOV BWPR , 00000000B; // CLOSE Write Pro
SETB EA;
    
```

### 3.1.5.3 CPU52 寄存器

表 3-1-5 列出了 ATT7039 CPU52 寄存器的地址、名称、复位值和简单的功能描述,此外还列出了 CPU52 包含但标准 8052 中不包含的寄存器,用阴影部分标记。

表 3-1-5 CPU52 内部寄存器

地址	名称	默认值	功能描述
0x81	SP	0x07	堆栈指针寄存器
0x82	DPL	0x00	数据指针寄存器 0 低 8 位
0x83	DPH	0x00	数据指针寄存器 0 高 8 位
0x84	DPL1	0x00	数据指针寄存器 1 低 8 位
0x85	DPH1	0x00	数据指针寄存器 1 高 8 位
0x86	MCON	0x40	内存控制寄存器
0x87	PCON	0x00	P 控制寄存器
0x88	TCON	0x00	定时器控制寄存器
0x89	TMOD	0x00	定时器方式寄存器
0x8A	TL0	0x00	定时器 0 计数器低 8 位
0x8B	TL1	0x00	定时器 1 计数器低 8 位
0x8C	TH0	0x00	定时器 0 计数器高 8 位
0x8D	TH1	0x00	定时器 1 计数器高 8 位
0x8E	CKCON	0x71	时钟控制寄存器
0x90	P1	0x00	P1 口数据寄存器
0x91	WAKE_EN	0x00	唤醒使能控制位
0x92	DPS	0x00	DPTR 指针选择寄存器
0x93	PMUCFR	0x07	PMU 配置寄存器
0x94	VDCR	0x19	电压检测控制寄存器
0x95	TCR	0x00	测试控制寄存器
0x96	PMIFR	0x00	PMU 中断标志寄存器
0x97	PMIER	0x00	PMU 中断使能寄存器
0x98	S0CON	0x00	串口 0 控制寄存器
0x99	S0BUF	0x00	串口 0 数据缓冲寄存器
0x9A	IEN2	0x00	中断使能寄存器
0x9B	S1CON	0x00	串口 1 控制寄存器
0x9C	S1BUF	0x00	串口 1 数据寄存器
0x9D	S1RELL	0x00	串口 1 重载寄存器低字节
0x9E	CLKCFG	0x00	系统时钟配置寄存器
0x9F	SUPDC	0x00	内部模块使能控制寄存器
0xA0	P2	0x00	P2 口数据寄存器
0xA1	PLLCFG	0x00	PLL 时钟配置寄存器
0xA2	PMSR	0x00	PMU 状态指示寄存器
0xA6	BWPR	0x00	写保护寄存器
0xA7	RSTSR	0x00	复位标志寄存器
0xA8	IEN0	0x00	中断使能寄存器
0xA9	IPO	0x00	中断优先级寄存器

0xAA	S0RELL	0x00	串口 0 重载寄存器低字节
0xAC	EMUIE	0x00	EMU 中断使能寄存器
0xAE	RTCIE	0x00	RTC 中断使能寄存器
0xAF	KEYIE	0x30	按键中断控制寄存器
0xB0	P3	0x00	P3 口数据寄存器
0xB1	PWMCR	0x00	PWM 控制寄存器
0xB4	RTCIF	0x00	RTC 中断标志寄存器
0xB5	EMUIF	0x00	EMU 中断标志寄存器
0xB6	KEYIF	0x00	按键中断标志寄存器
0xB7	WAKEIF	0x00	唤醒标志寄存器
0xB8	IEN1	0x80	中断使能寄存器
0xB9	IP1	0x00	中断优先级寄存器
0xBA	S0RELH	0x03	串口 0 重载寄存器高字节
0xBB	S1RELH	0x03	串口 1 重载寄存器高字节
0xBC	PTA	0x00	PA 口数据寄存器
0xBD	PTB	0x00	PB 口数据寄存器
0xC0	IRCON	0x00	中断请求控制寄存器
0xC1	LEP0	0x00	P0 口 LED 控制寄存器
0xC4	LCDCLK	0x90	LCD 时钟频率选择寄存器
0xC5	LCDCR	0x00	LCD 驱动控制寄存器
0xC6	LCDCFG	0x00	LCD 输出复用配置寄存器
0xC7	KEYCFG	0x00	按键配置寄存器
0xC8	T2CON	0x00	定时器 2 控制寄存器
0xC9	WDTCON	0x00	WDT 控制寄存器
0xCA	CRCL	0x00	定时器 2 捕获寄存器低 8 位
0xCB	CRCH	0x00	定时器 2 捕获寄存器高 8 位
0xCC	TL2	0x00	定时器 2 计数器低 8 位
0xCD	TH2	0x00	定时器 2 计数器高 8 位
0xCE	P02CFG	0x00	P0 和 P2 输出复用配置寄存器
0xCF	P3CFG	0x00	P3 口配置寄存器
0xD0	PSW	0x00	程序状态字寄存器
0xD1	DDRA	0x00	PA 口输出方向寄存器
0xD2	DDRB	0x00	PB 口输出方向寄存器
0xD5	DDRP0	0x00	P0 口输出方向寄存器
0xD6	DDRP1	0x00	P1 口输出方向寄存器
0xD7	DDRP2	0x00	P2 口输出方向寄存器
0xD8	ADCON	0x40	串口 0 波特率选择寄存器
0xD9	DDRP3	0x00	P3 口输出方向寄存器
0xDA	I2CDAT	0x00	I2C 数据寄存器
0xDB	I2CADR	0x00	I2C 地址寄存器
0xDC	I2CCON	0x00	I2C 控制寄存器
0xDD	I2CSTA	0x00	I2C 状态寄存器

0xE0	ACC	0x00	累加器
0xE9	ECADR	0x00	EMU 校表地址寄存器
0xEA	ECDATH	0x00	EMU 校表高字节数据寄存器
0xEB	ECDATL	0x00	EMU 校表低字节数据寄存器
0xEC	EPADR	0x00	EMU 参数地址寄存器
0xED	EPDATH	0x00	EMU 参数高字节数据寄存器
0xEE	EPDATM	0x00	EMU 参数中字节数据寄存器
0xEF	EPDATL	0x00	EMU 参数低字节数据寄存器
0xF0	B	0x00	B 寄存器
0xF1	ALMR	0x00	闹钟的分钟设置
0xF2	ALHR	0x00	闹钟的小时设置
0xF3	MTHR	0x00	月寄存器
0xF4	YRR	0x00	年寄存器
0xF5	DOWR	0x00	周寄存器
0xF8	RTCCON	0x00	RTC 控制寄存器
0xF9	RTCCAL	0x00	RTC 时钟校正寄存器
0xFA	RTC1CNT	0x00	RTC 定时器 1 计数设置
0xFB	RTC2CNT	0x00	RTC 定时器 2 计数设置
0xFC	SECR	0x00	秒寄存器
0xFD	MINR	0x00	分钟寄存器
0xFE	HRR	0x00	小时寄存器
0xFF	DAYR	0x00	日寄存器

### 寄存器说明:

#### 1. ACC

累加器是一个最常用的专用寄存器。大部分单操作数指令的操作取自累加器。很多双操作数指令的一个操作数取自累加器。加、减、乘、除算术运算指令的运算结果都存放在累加器 A 或 AB 寄存器中。指令系统中用 A 作为累加器的助记符。

#### 2. B

在乘除指令中,用到 B 寄存器。乘法指令的两个操作数分别取自 A 和 B,其结果存放在 AB 寄存器中。除法指令中,被除数取自 A,除数取自 B,商数存放于 A,余数存放于 B。在其他指令中,B 寄存器可作为 RAM 中的一个单元来使用。

#### 3. PSW

程序状态字 PSW 是一个 8 位寄存器,它包含了程序状态信息。此寄存器的含义参见下表:

表 3-1-6 程序状态字 (PSW, 0xD0H)

Program Status Word (PSW)			Address: D0H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	CY	AC	F0	RS1	RS0	OV	F1	P
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述															
CY	<b>进位标志</b> 在执行某些算术和逻辑指令时，可以被硬件或软件复位或清零。在布尔处理机中它被认为是位累加器，其重要性相当于一般中央处理机中的累加器 A。															
AC	<b>辅助进位标志</b> 在进行加法或减法操作而产生低 4 位数（十进制的一个数字）向高 4 位数进位或借位时，Ac 将被置位，否则被清零。Ac 被用于 DAA 指令的十进制调整。															
F0	<b>标志 0</b> 是用户定义的一个状态标记，可用软件置位或清零。															
RS1	<b>工作寄存器区选择控制位 1</b> 和 RS0 一起用以选择工作寄存器区															
RS0	<b>工作寄存器区选择控制位 0</b> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>RS1</th> <th>RS0</th> <th>Bank 选择</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>区 0 (00H~07H)</td> </tr> <tr> <td>0</td> <td>1</td> <td>区 1 (08H~0FH)</td> </tr> <tr> <td>1</td> <td>0</td> <td>区 2 (10H~17H)</td> </tr> <tr> <td>1</td> <td>1</td> <td>区 3 (18H~1FH)</td> </tr> </tbody> </table>	RS1	RS0	Bank 选择	0	0	区 0 (00H~07H)	0	1	区 1 (08H~0FH)	1	0	区 2 (10H~17H)	1	1	区 3 (18H~1FH)
RS1	RS0	Bank 选择														
0	0	区 0 (00H~07H)														
0	1	区 1 (08H~0FH)														
1	0	区 2 (10H~17H)														
1	1	区 3 (18H~1FH)														
OV	<b>溢出标志</b> 当加法产生进位，减法产生借位，乘除产生溢出时，置 1。否则为 0。															
F1	<b>标志 1</b> 是用户定义的一个状态标记，可用软件置位或清零。															
P	<b>奇偶校验</b> 每个指令周期都由硬件来置位或清零，以表示累加器 A 中 1 的位数的奇偶数。若 1 的位数为奇数，则 P 置位，否则清 0。															

#### 4. SP

栈指针 SP 是一个 8 位专用寄存器。它指出堆栈顶部在内部 RAM 中的位置。系统复位后，SP 初始化为 07H，使得堆栈事实上由 08H 单元开始。考虑到 08H~1FH 单元分属于工作寄存器 1~3，若程序设计要用到这些区，则把 SP 的值改置更大的值。SP 的初值越小，堆栈深度就越深。堆栈指针的值可由软件改变，因此堆栈在内部 RAM 中的位置比较灵活。

除用软件改变 SP 值外，在执行 PUSH/POP、各种子程序调用、中断响应、子程序返回 (RET) 和中断返回 (RETI) 等指令时，SP 值将自动增加或减少。

#### 5. DPTR

标准 8052 的数据指针 DPTR 是一个 16 位专用寄存器，其高位字节寄存器用 DPH 表示，低位字节用 DPL 表示，DPTR 主要用来存放 16 位地址，当对外部数据存储器空间寻址时，可作为间接寄存器用。

ATT7039 包括一个同标准 8052 相同的数据指针 DPTR，它位于 SFR 82H (DPL) 和 83H (DPH)，默认情况下，数据指针使用 DPTR。除此之外，为加速数据的块搬移操作，ATT7039 增加了第二个数据指针 DPTR1，DPTR1 位于 SFR 84H (DPL1) 和 85H (DPH1)。DPS 寄存器 (SFR 92H) 的 SEL 位用来选择当前数据指针使用 DPTR 还是 DPTR1，当 SEL=0，使用 DPTR 的指令的数据指针用 DPL 和 DPH 作为数据指针；当 SEL=1，使用 DPTR 的指令的数据指针用 DPL1 和 DPH1。SEL 是 DPS 的第 0 位，DPS 的其他位无用。

所有和 DPTR 相关的指令使用 DPS 选择的数据指针。SEL 取反将导致数据指针切换，切换最快的方法是使用 INC DPS 指令，仅需要一条指令，就可使数据指针由源地址指向目的地址，当进行块数据搬移时，这样做节省了保存源地址和目的地址的代码和时间。当搬移大批量数据时，使用双数据指针的机制显



著地提高了代码的效率。

## 6. 串行数据缓冲区

ATT7039 有两个串口 UART0, UART1, 串行数据缓冲区 S0BUF, S1BUF 用于存放 UART0, UART1 欲发送或已接收的数据, 对于任一个 SBUF, 它实际上由两个独立的寄存器组成, 一个发送缓冲区, 另一个是接收缓冲区。当要发送的数据传送到 SBUF 时, 进的是发送缓冲区。当要从 SBUF 读数据时, 则取自接收缓冲区, 取走的是刚接收到的数据。

## 7. CKCON

程序和数据存储器的读写延迟控制寄存器。这个寄存器可以设置程序和数据存储器的读写的延迟时钟数, 以便适应不同的存储器。降低程序读取的延迟时间, 可以有效的增加程序执行速度。ATT7039 的程序和数据存储器均支持最短的读写延迟。

表 3-1-7 时钟控制寄存器 (CKCON, 0x8EH)

Clock controller register (CKCON)			Address: <b>8EH</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	CKCON.7	CKCON.6	CKCON.5	CKCON.4	CKCON.3	CKCON.2	CKCON.1	CKCON.0
Write:								
Reset:	0	1	1	1	0	0	0	1

Bit	Symbol	Description	Type
ckcon.7	-	not used, read as 0	R
ckcon.6	-	Program memory wait state control	R/W
ckcon.5	-		
ckcon.4	-		
ckcon.3	-	not used, read as 0	R
ckcon.2	-	External data memory stretch cycle control	R/W
ckcon.1	-		
ckcon.0	-		

## 3.2 存储单元

### 3.2.1 概述

ATT7039内置8bit可编程高可靠性32K bytes Flash Memory, 256 Byte Info Flash和1K XRAM。

Flash被擦除后的状态为0xFF, Flash在逻辑空间由2部分或3部分组成(根据RAMMAP寄存器位的选择), 若分成2部分则是由32K Flash (Program Memory) 和256 Bytes Info Flash (Data Memory)。其中32K Code Flash可以通过RAMMAP寄存器位分成28K Flash (Program Memory) 和4K Flash (Data Memory), 以及256 Bytes Info Flash (Data Memory), 这样逻辑空间就变成是3部分。

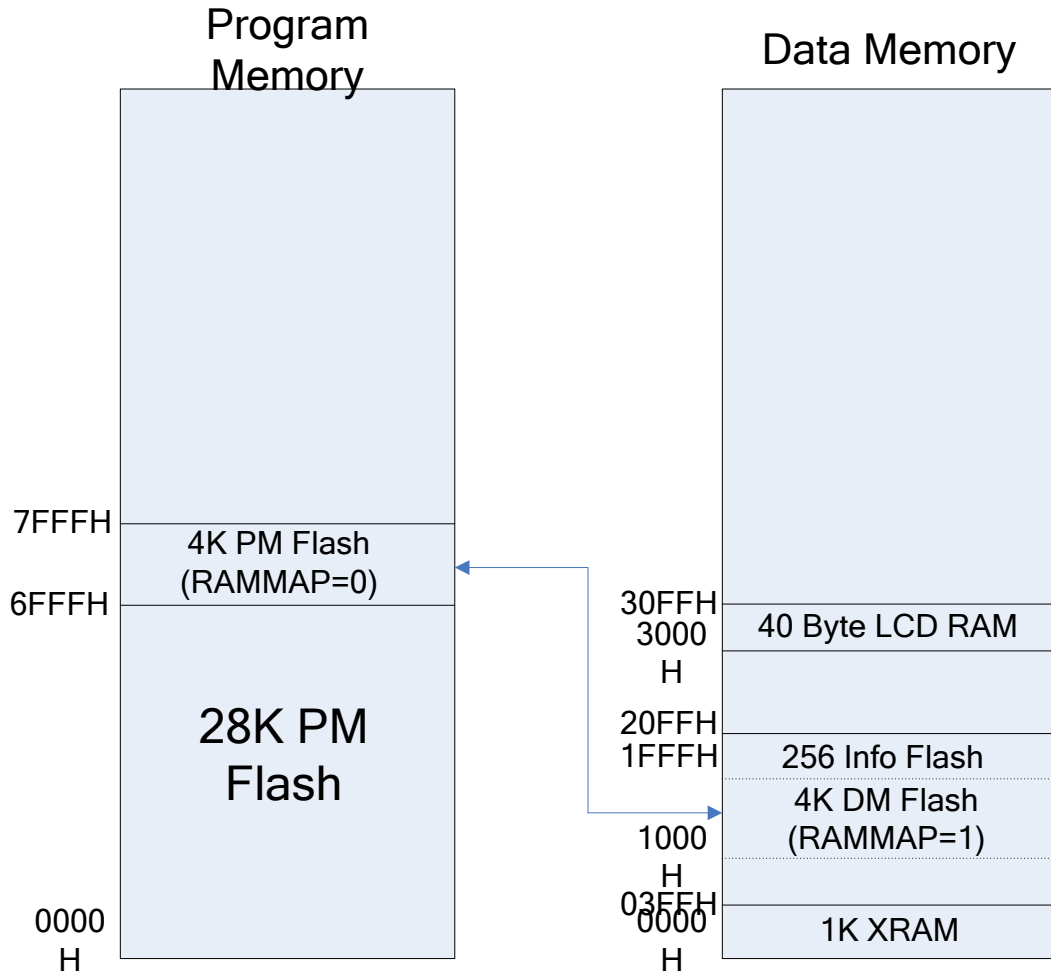
- 当用户配置Flash为32K时, 32K Flash用于用户存储程序, 地址分配为0000H—7FFFH。
- 当用户配置Flash为28K+4K时, 4K Data Flash用于存储当系统掉电时用户需要保持的数据, 在Data Memory空间地址分配为1000H—1FFFH。
- 256 Info flash可用于存储当系统掉电时用户需要保持的数据, 在Data Memory空间地址分配为2000—20FFFH。

1K XRAM 分为2个部分, 低256个字节的XRAM (0x0000—0x00FF) 在芯片进入SLEEP模式后不掉电, 仍然能够保存用户数据。其余的XRAM空间 (0x0100—0x03FF) 当芯片进入SLEEP模式后数据丢失。

### 3.2.2 ATT7039 Flash 特性

- Flash字节读取时间: 40ns
- Flash字节写时间: 30us (max)
- Flash页擦除时间: 20ms (max)
- Flash全擦除时间: 20ms (max)
- Code Memory: 1K bytes/page
- Data Memory: 1K bytes/page
- Information memory: 256 bytes/page
- 擦写次数: 200,000 次
- 数据保持时间: 20年 (min)

3.2.3 框图



RAMMAP=1: 28K PM flash + 4K DM flash + 0.256K Info flash + 1K XRAM  
 RAMMAP=0: 32K PM flash + 0.256K Info flash + 1K XRAM

图 3-2-1 UAM 下 PM 和 DM 存储器映射

RAMMAP	PM Flash	DM/PM Flash	XRAM	Info Flash
0	28K	4K PM	1K	256byte
1	28K	4K DM	1K	256byte

3.2.4 存储器映射及类别说明

1, XRAM

1024 字节的 XRAM 存储空间映射到 Data Memory 的地址 0000H-03FFH，即可通过寻址 Data Memory 地址 0000H-03FFH 来访问 1024 字节的 XRAM。

2, Info Flash memory

这部分的 256 字节 Flash 占用 DM 地址空间：2000H-20FFH。采用 MOVX 指令访问。这部分只包含一页，为 256 字节。

3, Code Flash Memory

32k 字节的 PM Flash 存储空间映射到 Program Memory 的地址 0000H-7FFFH。PM Flash 用于存储用户

程序，分为 32 页，每页 1KB。

#### 4, Data Flash Memory

4k 字节的 DM Flash 存储空间映射到 Data Memory 的地址 1000H-1FFFH。采用 MOVX 指令访问。分为 4 页，每页 1KB。

### 3.2.5 Flash 的操作

#### 3.2.5.1 Flash 页擦除

这种模式可以在 flash 中运行程序来操作 flash，操作方法如下：

- (1) 设置寄存器 PCON 的 PMW 为 1，选择对 PM 的 Flash 操作。对 DM 或者 INFO 操作时，需要 PMW 置 0。
- (2) 设置寄存器 MCON 使能 Flash 擦除和写入，**PMLOCK = 1, RSLOCK = 1**
- (3) 设置寄存器 MCON 的 FOP1 和 FOP0 选择对 Flash 的页擦除操作，FOP[1:0]=10
- (4) 设置寄存器 MCON 的 MAPMOD 为 1（对于 H1103，该位只能置 1）
- (5) 将要擦除的 Flash 页地址写入 DPTR，将要 flash 的数据送写入累加器 A，写 flash，等待 flash 操作完成。也就是如下代码：

```

        MOVX  @DPTR, A
    XXX:  MOV   A, MCON
        JB    ACC.4, XXX      ; 等待Flash的BUSY信号完成
        RET
    
```

#### 3.2.5.2 Flash 写操作

这种模式可以在 flash 中运行程序来操作 flash，操作方法如下：

- (1) 设置寄存器 PCON 的 PMW 为 1，选择对 PM 的 Flash 操作。对 DM 或者 INFO 操作时，需要 PMW 置 0。
- (2) 设置寄存器 MCON 使能 Flash 擦除和写入，**PMLOCK = 1, RSLOCK = 1**
- (3) 设置寄存器 MCON 的 FOP1 和 FOP0 选择对 Flash 的写操作，FOP[1:0]=01
- (4) 设置寄存器 MCON 的 MAPMOD 为 1（对于 H1103，该位只能置 1）
- (5) 将要写的 Flash 地址写入 DPTR，将要 flash 的数据送写入累加器 A，写 flash，等待 flash 操作完成。也就是如下代码：

```

        MOVX  @DPTR, A
    XXX:  MOV   A, MCON
        JB    ACC.4, XXX      ; 等待Flash的BUSY信号完成
        RET
    
```

#### 3.2.5.3 Flash 读操作

- (1) 设置寄存器 PCON 的 PMW 为 1，选择对 PM 的 Flash 操作。对 DM 或者 INFO 操作时，需要 PMW 置 0。
- (2) 设置寄存器 MCON 的 MAPMOD 为 1（对于 H1103，该位只能置 1）
- (3) 将要写的 Flash 地址写入 DPTR，读取 flash 内容至累加器 A。也就是如下代码：

```

        MOV   DPL, #11H      ; 用户需要读取的低8位地址
        MOV   DPH, #22H      ; 用户需要读取的高8位地址
        MOVX  A, @DPTR       ; 读取地址 0x2211 的值到累加器 A
    
```

#### 3.2.5.4 Flash 的读写保护

将 Flash 的 Info 部分的最后一个字节 (20FFH) 写 0 后，Flash 便处于读保护状态，只能擦除，无法读出。

设置寄存器 MOCN 的 PMLOCK, RSLOCK 位打开 Flash 的写保护功能。

### 3.2.6 特殊功能寄存器

表 3-3-1 Flash 操作模式控制寄存器列表

地址	名称	复位值	功能描述
0x86	MCON	0x00	内存控制寄存器
0x87	PCON	0x08	程序控制寄存器

表 3-2-2 MCON 内存控制寄存器

Memory Control Register (MCON)			Address: 86H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	WRFAIL	X	RAMMAP	BUSY	FOP1	FOP0	PMLOCK	RSLOCK
Write:				x				
Reset:	0	0	0	0	0	0	0	0

位	功能描述															
WRFAIL	<b>FLASH 操作失败标志位</b> 1: 表示 Flash 在系统时钟切换时的写/擦除操作失败; 0: 表示 Flash 的写/擦除操作正常。															
RAMMAP	<b>RAM 操作模式选择</b> 1: 表示 Flash 为 28K Program Memory + 4K Data Memory 0: 表示 Flash 为单独的 32K Proram Memory。															
BUSY	<b>FLASH 忙标志位</b> 1: 表示 Flash 正在进行写/擦除操作; 0: 表示 Flash 空闲, 可以进行操作。															
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>MOVX 将执行 Flash 写操作</td> </tr> <tr> <td>1</td> <td>0</td> <td>MOVX 将执行 Flash Page 擦除操作</td> </tr> <tr> <td>1</td> <td>1</td> <td>MOVX 将执行 Flash Whole Chip 擦除操作</td> </tr> </tbody> </table>	FOP1	FOP0	FLASH 操作	0	0	处于 Flash 只读模式	0	1	MOVX 将执行 Flash 写操作	1	0	MOVX 将执行 Flash Page 擦除操作	1	1	MOVX 将执行 Flash Whole Chip 擦除操作
FOP1	FOP0	FLASH 操作														
0	0	处于 Flash 只读模式														
0	1	MOVX 将执行 Flash 写操作														
1	0	MOVX 将执行 Flash Page 擦除操作														
1	1	MOVX 将执行 Flash Whole Chip 擦除操作														
PMLOCK	<b>PM Flash 锁定位</b> 1: 在 UAM 模式下, 使能所有 Flash 的擦除/写操作。(UAM: 用户程序模式) 0: 在 UAM 模式下, PM Flash 处于只读模式, 不能够被修改。															
RSLOCK	<b>前 4K PM Flash 锁定位</b> 1: 在 UAM 模式下, 如果 PMLOCK=1, 使能前 4K 的 PM Flash 的擦除/写操作; 0: 在 UAM 模式下, 即使 PMLOCK=1, 前 4K 的 PM Flash 也是处于只读模式, 不可修改。															

表 3-2-3 PCON 程序控制寄存器

P Control Register (PCON)		Address: 87H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	SMOD	X	X	PMW	TEST	X	X	X
Write:								
Reset:	0	0	0	0	1	0	0	0

位	功能描述
SMOD	<b>串口 0 的波特率加倍控制位</b> 0: 串口 0 的波特率由波特率生成器决定; 1: 串口 0 的波特率加倍 (详情参考串口 0 波特率生成公式)
PMW	<b>存储器写操作控制位</b> 0: 使能数据存储器 (DM) 的写操作, 关闭程序存储器 (PM) 的写操作; 1: 使能程序存储器 (PM) 的写操作, 关闭数据存储器 (DM) 的写操作。
TEST	内部测试位, 务必置为 1。

### 3.3 中断系统

#### 3.3.1 概述

ATT7039 中断系统支持 11 个中断，其中 6 个通用中断，包括外部引脚 INT0 中断，定时器 T0、T1、T2 中断以及串行口 UART0、UART1，保留了标准 8051 原有功能。

另外 5 个中断，利用 R8051XC 的 5 个扩展中断，分别为：

- 计量中断(IRQ\_EMU)
- RTC 中断(IRQ\_RTC)
- 按键中断(IRQ\_KBI)
- I2C 中断 (IRQ\_I2C)
- PMU 中断(IRQ\_PMU)

#### 3.3.2 中断列表

表 3-3-1 ATT7039 中断系统

ATT7039 中断	自然优先级	中断向量	中断向量号	中断使能 (EA=1)	中断标志
External Interrupt 0	0	0003H	0	IEN0.0	TCON.1
Serial 1 Interrupt	1	0083H	16	IEN2.0	S1CON.0 S1CON.1
I2C	2	0043H	8	IEN1.0	I2CCON.3
Timer 0 Interrupt	3	000BH	1	IEN0.1	TCON.5
KBI	4	004BH	9	IEN1.1 KEYIE	KEYIF SPSTA
PMU	6	0053H	10	IEN1.2	PMIFR
Timer 1 Interrupt	7	001BH	3	IEN0.3	TCON.7
EMU	8	005BH	11	IEN1.3 EMUIE	EMUIF
Serial 0 Interrupt	9	0023H	4	IEN0.4	S0CON.0 S0CON.1
RTC	10	0063H	12	IEN1.4 RTCIE	RTCIF
Timer 2 Interrupt	11	002BH	5	IEN0.5 IEN1.7	IRCON.6 IRCON.7

#### 3.3.3 中断优先级

ATT7039 可设定 4 个中断优先级，不支持对单个中断源的优先级进行调整，只能根据固定的中断向量组进行向量组的优先级调整。优先级的调整主要通过寄存器 IP0, IP1 来设置。

4 个中断优先级如下表所示：

IP1.X	IP0.X	优先级
0	0	Level 0( lowest )
0	1	Level 1

1	0	Level 2
1	1	Level 3( highest )

中断向量组与相应的中断控制位如下表所示：

组编号	组优先级控制位	组成员		
0	IP1.0 IP0.0	External Interrupt 0	Serial 1 Interrupt	I2C(EXT INT7)
1	IP1.1 IP0.1	Timer 0 Interrupt	KBI(EXT INT2)	
2	IP1.2 IP0.2	X	PMU(EXT INT3)	
3	IP1.3 IP0.3	Timer 1 Interrupt	EMU(EXT INT4)	
4	IP1.4 IP0.4	Serial 0 Interrupt	RTC(EXT INT5)	
5	IP1.5 IP0.5	Timer 2 Interrupt	X	

### 3.3.4 特殊功能寄存器

表 3-3-2 中断系统寄存器列表

地址	名称	复位值	功能描述
0xA8	IEN0	0x00	中断使能寄存器 0
0xB8	IEN1	0x00	中断使能寄存器 1
0x9A	IEN2	0x00	中断使能寄存器 2
0xAE	RTCIE	0x00	RTC 中断使能寄存器
0xAC	EMUIE	0x00	EMU 中断使能寄存器
0x97	PMIER	0x00	PMU 中断使能寄存器
0xAF	KEYIE	0x00	按键中断控制寄存器
0xC0	IRCON	0x00	中断标志寄存器
0xA9	IP0	0x00	优先级控制寄存器 0
0xB9	IP1	0x00	优先级控制寄存器 1

#### 3.3.4.1 中断使能寄存器

表 3-3-3 IEN0 中断使能寄存器 0 (A8H)

Interrupt Enable 0 Register (IEN0)			Address: A8H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	EAL	X	ET2	ES0	ET1	X	ET0	EX0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
EAL	全体中断使能位 0: 关闭所有中断; 1: 全体中断使能, 若要打开某一个中断, 还需要打开它对应的中断使能位;



ET2	<b>定时器 2 中断使能</b> 0: 定时器 2 中断关闭; 1: 并且 EAL=1, 定时器 2 中断使能;
ES0	<b>串口 0 中断使能</b> 0: 串口 0 中断关闭; 1: 并且 EAL=1, 串口 0 中断使能;
ET1	<b>定时器 1 溢出中断使能</b> 0: 定时器 1 溢出中断关闭; 1: 并且 EAL=1, 定时器 1 溢出中断使能;
ET0	<b>定时器 0 溢出中断使能</b> 0: 定时器 0 溢出中断关闭; 1: 并且 EAL=1, 定时器 0 溢出中断使能;
EX0	<b>外部中断 0 使能</b> 0: 外部中断 0 关闭; 1: 并且 EAL=1, 外部中断 0 使能;

表 3-3-4 IEN1 中断使能寄存器 1 (B8H)

Interrupt Enable 0 Register (IEN1)			Address: <b>B8H</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	ERTC	EEMU	EPMU	EKBI_SPI	EI2C
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ERTC	<b>RTC 中断使能位</b> 0: RTC 中断关闭 1: 并且 EAL=1, RTC 中断使能 (RTC 中断使能还需要配置 RTCIE 寄存器)
EEMU	<b>EMU 中断使能位</b> 0: EMU 中断关闭 1: 并且 EAL=1, EMU 中断使能 (EMU 中断使能还需要配置 EMUIE 寄存器)
EPMU	<b>PMU 中断使能位</b> 0: PMU 中断关闭 1: 并且 EAL=1, PMU 中断使能 (EMU 中断使能还需要配置 PMIER 寄存器)
EKBI	<b>KBI 中断使能位</b> 0: KBI 中断关闭 1: 并且 EAL=1, KBI 中断使能 (KBI 中断使能还需要配置 KEYIE 寄存器)
EI2C	<b>I2C 中断使能位</b> 0: I2C 中断关闭 1: 并且 EAL=1, I2C 中断使能

表 3-3-5 IEN2 中断使能寄存器 2 (9AH)

Interrupt Enable 0 Register (IEN2)			Address: 9AH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	ES1
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ES1	<b>ES1 中断使能位</b> 0: 串口 1 中断关闭 1: 并且 EAL=1, 串口 1 中断使能

表 3-3-6 RTC 中断使能寄存器 RTCIE (AEH)

RTC Interrupt Enable Register (RTCIE)			Address: AEH					
	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

1: 使能中断; 0: 关闭中断。

位	功能描述
ALMIE	<b>闹钟中断使能位</b> 0: 闹钟中断关闭 1: 闹钟中断使能
RTC2IE	<b>RTC 定时器 2 中断使能位</b> 0: RTC 定时器 2 中断关闭 1: RTC 定时器 2 中断使能
RTC1IE	<b>RTC 定时器 1 中断使能位</b> 0: RTC 定时器 1 中断关闭 1: RTC 定时器 1 中断使能
MTHIE	<b>月中断使能位</b> 0: 月中断关闭 1: 月中断使能
DAYIE	<b>日中断使能位</b> 0: 日中断关闭 1: 日中断使能
HRIE	<b>小时中断使能位</b> 0: 小时中断关闭 1: 小时中断使能
MINIE	<b>分钟中断使能位</b> 0: 分钟中断关闭 1: 分钟中断使能
SECIE	<b>RTC 秒中断使能位</b> 0: RTC 秒中断关闭 1: RTC 秒中断使能

表 3-3-7 EMU 中断使能寄存器 EMUIE (ACH)

EMU Interrupt Enable Register (EMUIE)		Address: <b>ACH</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:	PFIE	QFIE	X	SPLIE	ZXIE	0	0	0
Write:						x	x	x
Reset:	0	0	0	0	0	0	0	0

只有使能相应的中断位，0xB1H 的中断标志才能被置 1。

位	功能描述
PFIE	<b>有功脉冲中断使能位</b> 0: 有功脉冲中断关闭 1: 有功脉冲中断使能
QFIE	<b>无功脉冲中断使能位</b> 0: 无功脉冲中断关闭 1: 无功脉冲中断使能
SPLIE	<b>ADC 原始数据刷新中断使能位</b> 0: ADC 原始数据刷新中断关闭 1: ADC 原始数据刷新中断使能
ZXIE	<b>电压/电流过零中断使能位</b> 0: 电压/电流过零中断关闭 1: 电压/电流过零中断使能

表 3-3-8 PMIER PMU 中断使能寄存器(97H)

Power Management Interrupt Enable Register (PMIER)		Address: <b>97H</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	X	PLLIE	BORIE	VSYISIE	LVDINIE
Write								
Reset:	0	0	0	0	0	0	0	0

注：需要同时使能 IEN1.bit2，PMIER 使能的中断才会有效。

位	功能描述
PLLIE	<b>PLL 中断使能位</b> PLLIE=1: 使能 PLL 中断 PLLIE=0: 禁止 PLL 中断
BORIE	<b>BOR 中断使能位</b> 1: 允许 BOR 中断 0: 关闭 BOR 中断
VSYISIE	<b>VSYS_DET 中断使能位</b> 1: 允许 VSYS_DET 中断 0: 关闭 VSYS_DET 中断
LVDINIE	<b>LVDIN_DET 中断使能位</b> 1: 允许 LVDIN_DET 中断 0: 关闭 LVDIN_DET 中断

表 3-3-9 KEYIE 按键中断控制寄存器 (AFH)

KEY Interrupt Enable Register (KEYIE)			Address: AFH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	KEY1 SEL	KEY0 SEL	X	X	KEYIE1	KEYIE0
Write:								
Reset:	0	0	1	1	0	0	0	0

位	功能描述
KEYIE1	<b>KEY1 中断使能位</b> 1: 表示 KEY1 中断有效 0: 表示 KEY1 中断无效
KEYIE0	<b>KEY0 中断使能位</b> 1: 表示 KEY0 中断有效 0: 表示 KEY0 中断无效

### 3.3.4.2 中断标志寄存器

表 3-3-10 中断标志寄存器 IRCON (C0H)

IRCON			Address: C0H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	TF2	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TF2	<b>定时器 2 溢出中断标志位</b> 0: 表示定时器 2 溢出中断未发生 1: 表示定时器 2 溢出中断已发生 (写 0 清零)

### 3.3.4.3 中断优先级寄存器

表 3-3-11 IP0 优先级控制寄存器 0(A9H)

Interrupt Priority Control Register 0 (IP0)			Address: A9H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0
Write:								
Reset:	0	0	0	0	0	0	0	0

表 3-3-9 IP1 优先级控制寄存器 1(B9H)

Interrupt Priority Control Register 1 (IP1)			Address: B9H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0
Write:								
Reset:	0	0	0	0	0	0	0	0

### 3.3.5 中断处理

中断系统遵循下列两条基本规则：

第一：低优先级中断源可被高优先级中断源所中断，而高优先级中断源不能被同级或低优先级的中断源所中断；

第二：一种中断源不管是高优先级或低优先级，一旦得到响应，与它同级的中断源不能再中断它。

当同时收到几个同一优先级中断时，响应哪一个中断源取决于内部查询顺序。其优先级排列见上表中同级中断优先级列。值得指出的是，RTC 中断、EMU 中断、PMU 中断、KBI 中断都包含了若干个中断源。以 RTC 中断为例，RTC 中断标志寄存器包含：闹钟中断、秒定时中断、定时器 2 中断、定时器 1 中断、日中断、小时中断、分钟中断、秒中断 8 个中断标志，用户可以在 ISR 中通过软件查询的方式判断 RTC 中断源，并在 ISR 中在清除 8 个中断标志。

## 3.4 定时器

### 3.4.1 概述

- ATT7039 内部有三个 16 位可编程的定时器：定时器 T0、定时器 T1 和定时器 T2。
- 每一个定时器/计数器都有两个独立的 8 位寄存器组成

Timer0: TL0和TH0

Timer1: TL1和TH1

Timer2: TL2和TH2

### 3.4.2 框图

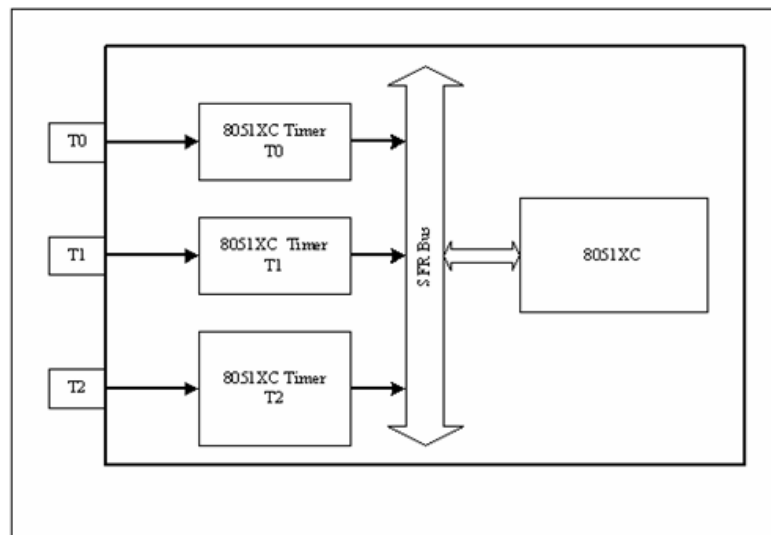


图 3-4-1 ATT7039 定时器/计数器框图

### 3.4.3 定时器工作模式

定时器 0 和定时器 1 都有 4 种工作模式，由 TMODE 和 TCON 控制。这四种工作模式是：

- 模式 0：13 位定时器
- 模式 1：16 位定时器
- 模式 2：自动装入时间常数的 8 位定时器
- 模式 3：两个 8bit 定时器（Timer1 在该模式下关闭）

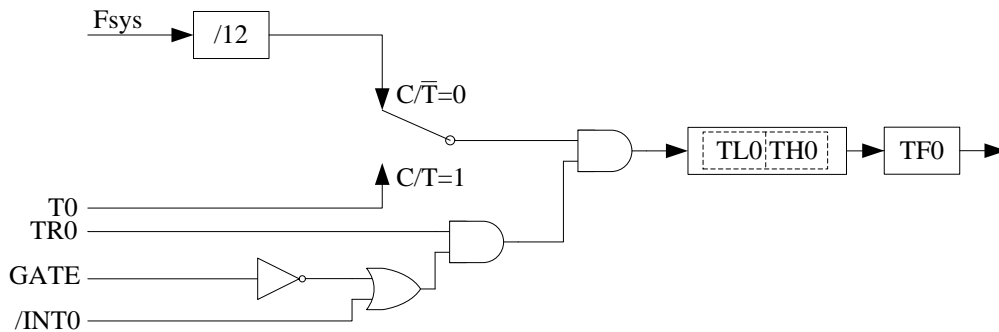
### 3.4.4 定时器 0

在定时器模式中，定时器 0 每 12 个时钟周期加 1，就是说，在每 12 个周期的时钟信号之后，它的计数会加 1。

#### 1, 模式 0

设置寄存器 TMOD 的标志位 TMOD[1:0]=00 可进入该模式，清除寄存器位 TMOD.2 可选择为定时器模式。

定时器被分为两个 8 位寄存器，低字节和高字节，低字节又被分为两部分：低 5 位和高 3 位（只有低 5 位是计数器的一部分）。这就使定时器成为一个 13 位的计数器，每 12 个时钟周期加 1，当定时器 0 溢出时寄存器位 TF0 置位，同时产生定时器溢出中断。程序进入中断后，该位被自动清零。



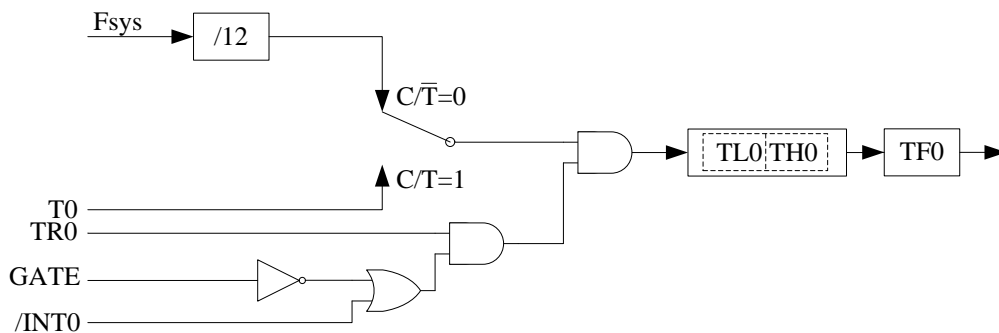
模式 0 下作定时器用时，其定时时间计算公式为：

$$Td = (2^{13} - X) \times 12 \times Tcpu$$

#### 2, 模式 1

设置寄存器 TMOD 的标志位 TMOD[1:0]=01 可进入该模式。

模式 1 与模式 0 的唯一区别就是低字节寄存器不再被分为两部分，整个低字节都用于计数器。在模式 1 中，定时器 0 是一个 16 位的计数器。



模式 1 下作定时器用时，其定时时间计算公式为：

$$Td = (2^{16} - X) \times 12 \times Tcpu$$

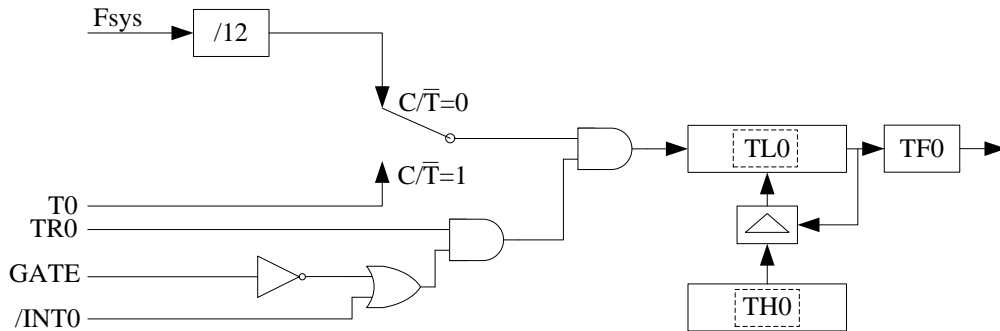
### 3, 模式 2

设置寄存器 TMOD 的标志位 TMOD[1:0]=10 可进入该模式。清除寄存器位 TMOD.2 可选择为定时器模式。

在这种模式下, 只有低字节寄存器 TL0 会每 12 个时钟周期加 1。定时器是一个 8 位的重载定时器。当定时器 0 溢出时, TF0 置位, 同时产生定时器溢出中断。程序进入中断后, 该位被自动清零。

当溢出发生时, 新的值将会从高字节 TH0 中抓取。

标志位 TR0 控制定时器的停止和运行。



模式 2 下作定时器用时, 其定时时间计算公式为:

$$Td = (2^8 - X) \times 12 \times Tcpu$$

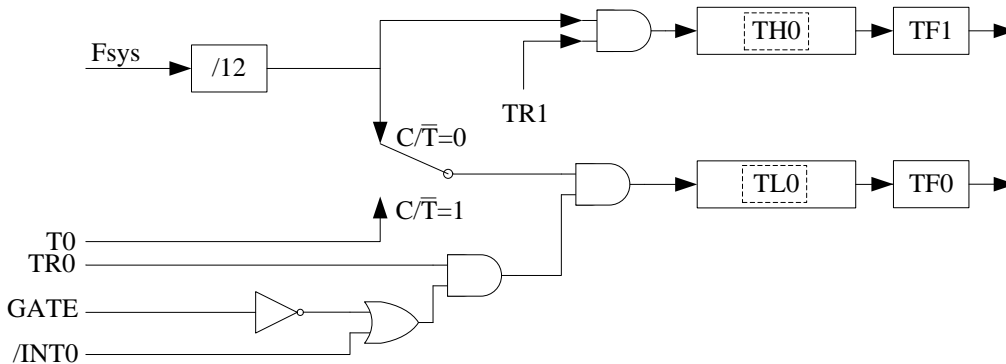
### 4, 模式 3

设置寄存器 TMOD 的标志位 TMOD[1:0]=11 可进入该模式。清除寄存器位 TMOD.2 可选择为定时器模式。

在这种模式下, 低字节 TL0 会每 12 个时钟周期加 1。高字节 TH0 会每 12 个时钟周期加 1。

当定时器低字节 0 溢出时, TF0 置位, 同时产生定时器溢出中断, 程序进入中断后, 该位被自动清零。当定时器的高字节溢出时, TCON.7 位, 同时产生定时器溢出中断, 程序进入中断后, 该位被自动清零。

在这种模式中, 定时器的低字节受 TR0 控制。高字节受 TR1 控制。



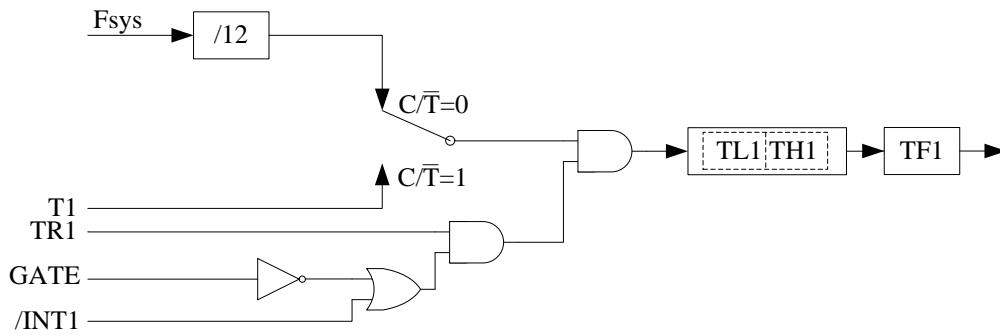
#### 3.4.5 定时器 1

在定时器模式中, 定时器 1 每 12 个时钟周期加 1, 就是说, 在每 12 个周期的时钟信号之后, 它的计数会加 1。

##### 1, 模式 0

设置寄存器 TMOD 的标志位 TMOD[5:4]=00 可进入该模式。清除寄存器位 TMOD.6 可选择为定时器。

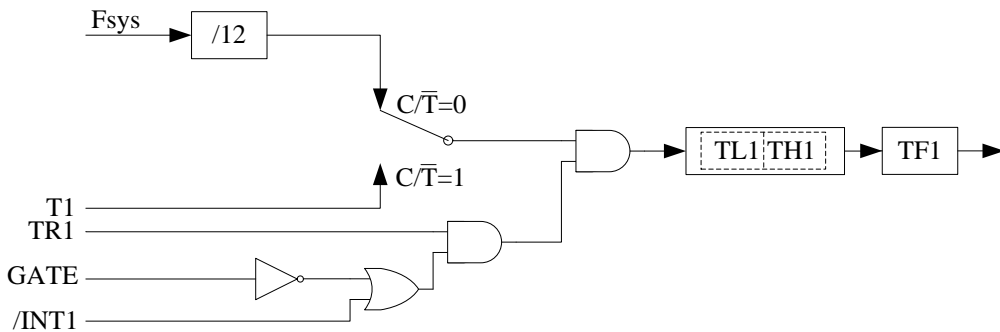
定时器 1 被分为两个 8 位寄存器, 低字节和高字节, 低字节又被分为两部分: 低 5 位和高 3 位 (只有低 5 位是计数器的一部分), 这就使定时器 1 成为一个 13 位的计数器, 每 12 个时钟周期加 1。当定时器 1 溢出时, TF1 置位, 同时产生定时器溢出中断。程序进入中断后, 该位被自动清零。



## 2, 模式 1

设置寄存器 TMOD 的标志位 TMOD[5: 4]=01 可进入该模式。

模式 1 与模式 0 的唯一区别就是低字节寄存器不再被分为低 5 位和高 3 位两部分，整个低字节都用于计数器，在模式 1 中，定时器 1 是一个 16 位的计数器。



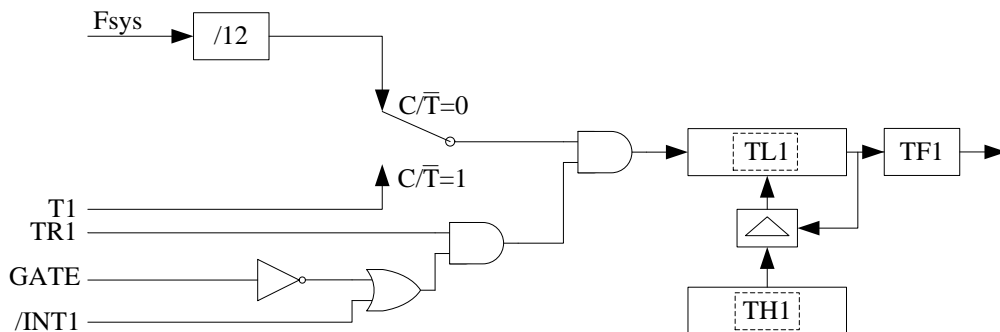
## 3, 模式 2

设置寄存器 TMOD 的标志位 TMOD[5: 4]=10 可进入该模式。清除寄存器位 TMOD.6 可选择为定时器。

在这种模式下，只有低字节 TL1 会每 12 个时钟周期加 1。在这种模式下，定时器是一个 8 位的重载定时器。当低字节溢出时，TF1 置位，同时产生定时器溢出中断。程序进入中断后，该位被自动清零。

当溢出发生时，新的值将会从高字节 TH1 中抓取。

标志位 TR1 可以控制定时器的停止和运行。



## 4, 模式 3

设置寄存器“TMOD”的标志位 TMOD [5: 4]=11 可进入该模式。

在这种模式下，定时器/计数器 1 被关闭（只有定时器/计数器 0 可在模式 3 下操作）。

### 3.4.6 定时器 2

Timer2 是一个 16bit 的定时器。



### 3.4.6.1 定时器功能

设置寄存器 T2CON 的标志位 T2I0=1 和 T2I1=0 进入该模式。定时器 2 有两个频率的时钟可以选择，当标志位 T2PS=0 时，每 12 个系统时钟计数器加 1；当标志位 T2PS=1 时，每 24 个系统时钟计数器加 1。

### 3.4.6.2 重载功能

设置寄存器 T2CON[4: 3]=10 设置定时器 2 为重载模式。当 Timer2 溢出时，TH2/TL2 自动从 CRC 寄存器中载入 16 位数据。

## 3.4.7 特殊功能寄存器

表 3-4-1 定时器 0 和定时器 1,定时器 2 寄存器列表

地址	名称	复位值	功能描述
0x88	TCON	0x00	定时器控制寄存器
0x89	TMOD	0x00	定时器方式寄存器
0x8A	TL0	0x00	定时器 0 计数器低 8 位
0x8B	TL1	0x00	定时器 1 计数器低 8 位
0x8C	TH0	0x00	定时器 0 计数器高 8 位
0x8D	TH1	0x00	定时器 1 计数器高 8 位
0xCC	TL2	0x00	定时器/计数器 2 低 8 位
0xCD	TH2	0x00	定时器/计数器 2 高 8 位
0xC8	T2CON	0x40	定时器 2 控制寄存器
0xCA	CRCL	0x00	自动装载寄存器低 8 位 增加 (T2)
0xCB	CRCH	0x00	自动装载寄存器高 8 位 增加 (T2)
0xC0	IRCON	0x00	中断标志寄存器

表 3-4-2 定时器 0、定时器 1 控制寄存器 (TCON 0x88H)

TCON		Address: 88H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	TF1	TR1	TF0	TR0	X	X	IE0	IT0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TF1	<b>定时器1溢出中断标志</b> 当定时器1溢出时，由硬件置位。 当进入中断处理程序后，由硬件自动清除；也可以由软件清除。
TR1	<b>定时器1运行控制位</b> 清除该位，定时器1将停止运行
TF0	<b>定时器0溢出中断标志</b> 当定时器0溢出时，由硬件置位。 当进入中断处理程序后，由硬件自动清除；也可以由软件清除。

TR0	<b>定时器0运行控制位</b> 清除该位，定时器0将停止运行
IE0	<b>外部中断0请求标志</b> 当外部中断int0产生时（电平触发或沿触发），由硬件置位。 当进入中断处理程序后，由硬件清除。
IT0	<b>外部中断0类型控制位</b> TCON.0=1：外部中断0由输入引脚的下降沿触发； TCON.0=0：外部中断0由输入引脚的低电平触发；

表 3-4-4 定时器 0、定时器 1 模式寄存器(TMOD 0x89H)

T0&T1 Mode register (TMOD)			Address: <b>89H</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	reserved	reserved	M1	M0	reserved	reserved	M1	M0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述																				
TMOD.7	Reserved, 保持为0																				
TMOD.6	Reserved, 保持为0																				
TMOD.5	定时器1模式控制位																				
TMOD.4	如下表所示																				
TMOD.3	Reserved, 保持为0																				
TMOD.2	Reserved, 保持为0																				
TMOD[1: 0]	定时器0模式控制位																				
	<table border="1"> <thead> <tr> <th>M1</th> <th>M0</th> <th>方式</th> <th>说明</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>13位定时器，TL0（TL1）中的低5位和TH0(TH1)中的8位</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>16位定时器</td> </tr> <tr> <td>1</td> <td>0</td> <td>2</td> <td>8位定时器，具有自动再装入功能，装入在TL0(TL1)溢出时发生，装入的值由TH0(TH1)提供。</td> </tr> <tr> <td>1</td> <td>1</td> <td>3</td> <td>定时器1停止操作，定时器0分成两个8位的独立计数器：TL0使用控制位tr0，溢出时将tf0置位；TH0使用控制位tr1，溢出时将tf1置位。</td> </tr> </tbody> </table>	M1	M0	方式	说明	0	0	0	13位定时器，TL0（TL1）中的低5位和TH0(TH1)中的8位	0	1	1	16位定时器	1	0	2	8位定时器，具有自动再装入功能，装入在TL0(TL1)溢出时发生，装入的值由TH0(TH1)提供。	1	1	3	定时器1停止操作，定时器0分成两个8位的独立计数器：TL0使用控制位tr0，溢出时将tf0置位；TH0使用控制位tr1，溢出时将tf1置位。
	M1	M0	方式	说明																	
	0	0	0	13位定时器，TL0（TL1）中的低5位和TH0(TH1)中的8位																	
	0	1	1	16位定时器																	
1	0	2	8位定时器，具有自动再装入功能，装入在TL0(TL1)溢出时发生，装入的值由TH0(TH1)提供。																		
1	1	3	定时器1停止操作，定时器0分成两个8位的独立计数器：TL0使用控制位tr0，溢出时将tf0置位；TH0使用控制位tr1，溢出时将tf1置位。																		

表 3-4-4 定时器 1 计数器高 8 位 (TH1)

TH 1			Address: <b>8DH</b>					
	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

表 3-4-5 定时器 1 计数器低 8 位 (8BH,TL1)

TL1			Address: 8BH					
	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

这两个寄存器存储了定时器1的状态值，TH1存储高字节，TL1存储低字节。

表 3-4-6 定时器 0 计数器高 8 位(8CH,TH0)

TH0			Address: 8CH					
	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

表 3-4-7 定时器 0 计数器低 8 位 (8AH,TL0)

TL0			Address: 8AH					
	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

这两个寄存器存储了定时器0的状态值，TH0存储高字节，TL0存储低字节

表 3-4-8 定时器 2 控制寄存器 (T2CON 0xC8H)

T2 control register (T2CON)			Address: C8H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	T2PS	X	X	T2OL1	T2OL0	X	T2I1	T2I0
Write:								
Reset:	0	1	0	0	0	0	0	0

位	功能描述									
T2PS	<b>时钟选择位</b> 0 – 定时器2的时钟选择为系统时钟的1/12; 1 – 定时器2的时钟选择为系统时钟的1/24;									
T2OL[1: 0]	<b>重载功能选择位</b> 当配置 T2CON[4:3]=10 时，溢出重载；其他情况不重载									
T2I[1: 0]	<b>定时器2输入选择位</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>T2I1</th> <th>T2I0</th> <th>T2工作模式</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>定时器2中止工作</td> </tr> <tr> <td>0</td> <td>1</td> <td>定时器模式：输入时钟为fsys/12 或者 fsys/24</td> </tr> </tbody> </table>	T2I1	T2I0	T2工作模式	0	0	定时器2中止工作	0	1	定时器模式：输入时钟为fsys/12 或者 fsys/24
T2I1	T2I0	T2工作模式								
0	0	定时器2中止工作								
0	1	定时器模式：输入时钟为fsys/12 或者 fsys/24								

表 3-4-9 定时器 2 计数器高 8 位(CDH,TH2)

TH2			Address: CDH					
	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

表 3-4-10 定时器 2 计数器低 8 位 (0CCH, TL2)

TL2			Address: CCH					
	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

这两个寄存器存储了定时器2的状态值，TH2存储高字节，TL2存储低字节。定时器2可以被配置成比较、捕获、重载模式。

表 3-4-12 定时器 2 自动装载高 8 位寄存器(CBH, CRCH)

CRCH			Address: CBH					
	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

表 3-4-13 定时器 2 自动装载低 8 位寄存器(CAH, CRCL)

CRCL			Address: CAH					
	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

重载寄存器 CRCH/CRCL 是 16 位寄存器，对应定时器 2 的重载工作模式；

表 3-4-14 中断标志寄存器 IRCON (C0H)

IRCON			Address: C0H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	x	TF2	x	x	x	x	x	x
Write:	x		x	x	x	x	x	x
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TF2	定时器 2 溢出中断标志位 0: 表示定时器 2 溢出中断未发生 1: 表示定时器 2 溢出中断已发生 (写 0 清零)

---

## 4 计量模块

### 4.1 概述

ATT7039 提供单相电能计量所需要有功功率与有功电能、无功功率与无功电能、电压有效值、电流有效值及频率参数等，支持软件校表方式。

- 在动态范围 3000: 1 内有功误差小于 0.1%
- 无功精度远优于国标 2 级要求
- 提供有功功率与有功电能，无功功率与无功电能
- 提供电压、电流有效值
- 提供电压频率测量值
- 提供两路 ADC 瞬时采样波形数据，以及有功功率、无功功率波形数据
- 提供灵活的潜动与启动方案
- 提供直流偏置自动校正
- 提供多种能量计算模式
- 开放快速脉冲计数寄存器，防止上下电时丢失电能
- 脉冲输出 PF/QF 脉宽可选
- 提供过零中断检测信号
- 提供无功移相补偿
- 提供有效值小信号偏置校正
- 提供有功功率 P，无功功率 Q 小信号偏置校正

## 4.2 框图

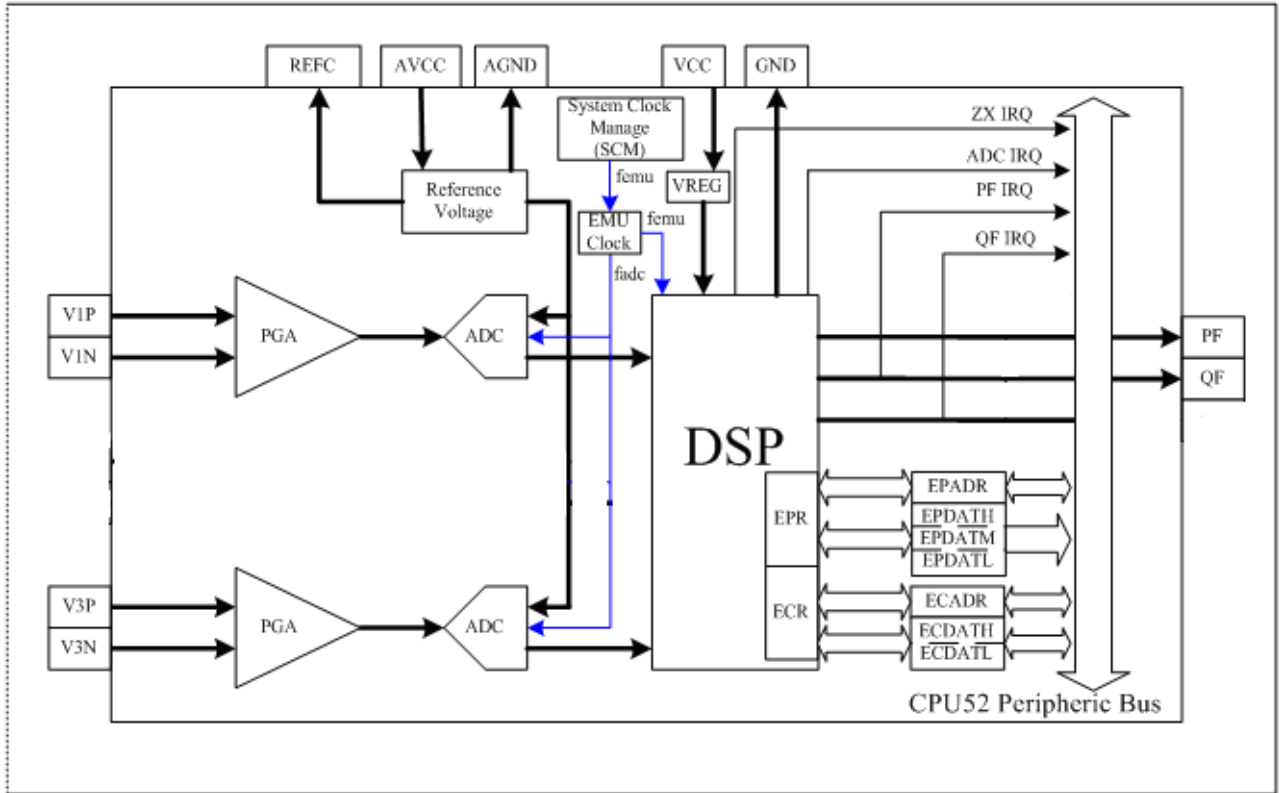


图 4-2-1 电能计量单元 (EMU)

## 4.3 功能描述

电能计量单元 EMU 包括两路完全独立的  $\Sigma$ - $\Delta$ ADC 以及数字信号处理部分。两路 ADC 完成电流信号和电压信号的采样，数字信号处理部分完成有功功率与有功电能、无功功率与无功电能、电压有效值、电流有效值及频率计算等计量功能。

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

femu 为 EMU 单元的时钟，ADC 的采样时钟 fadc 为 femu/6。

如果 femu=5.505024MHz，那么 fadc=917KHz。

### 4.3.1 模数转换器

ATT7039 有两路完全独立的两阶  $\Sigma$ - $\Delta$  ADC，每路 ADC 都有一个模拟增益放大器 (PGA)，内部有一个 1.25V 的高稳定度片内基准电压，每路 ADC 可以独立开关，通过寄存器 EMU\_Ctrl 进行设置。

模拟增益放大器完成输入差分信号的幅度放大，放大后的信号再送给 ADC 进行采样，在极小信号输入时能够保证测量的线性度。通过寄存器 ADCCON(58H)可以对两路 ADC 独立配置放大倍数，放大倍数分别为 1、2、8、16。

通过寄存器 DGAIN(5AH)可以对两路 ADC 采到的信号进行数字增益设置，共有 1、2、4、8 倍四种设置。在大信号不溢出的情况下，数字增益放大可以增加小信号计算的有效位数，进一步提高计量精度。

### 4.3.2 ADC 采样输出和功率波形输出

ATT7039 将两路 ADC 输出的 16bit 波形数据 Spl\_I1(00H)、Spl\_U(02H)开放给用户。同时也将作为电能累加的功率波形数据 Spl\_P(03H)开放给用户。

波形数据更新的频率为 femu/192、femu/384、femu/768、femu/1536、femu/3072，例如如果 femu 选择为 5.505MHz，那么波形数据更新频率可以选择为 28.6kHz、14.3kHz、7.15kHz、3.6kHz、1.8kHz。

注意：由于波形数据更新频率最快可以达到 28.6KHz，因此 ADC\_IRQ 的中断请求最快也可以达到 28.6KHz，而 CPU 的执行速度最快可以达到 11.010048MHz，用户在编程时需要注意 CPU 程序是否用足够的时间来响应中断。

### 4.3.3 有功功率、无功功率和视在功率

ATT7039 提供有功功率率输出寄存器。

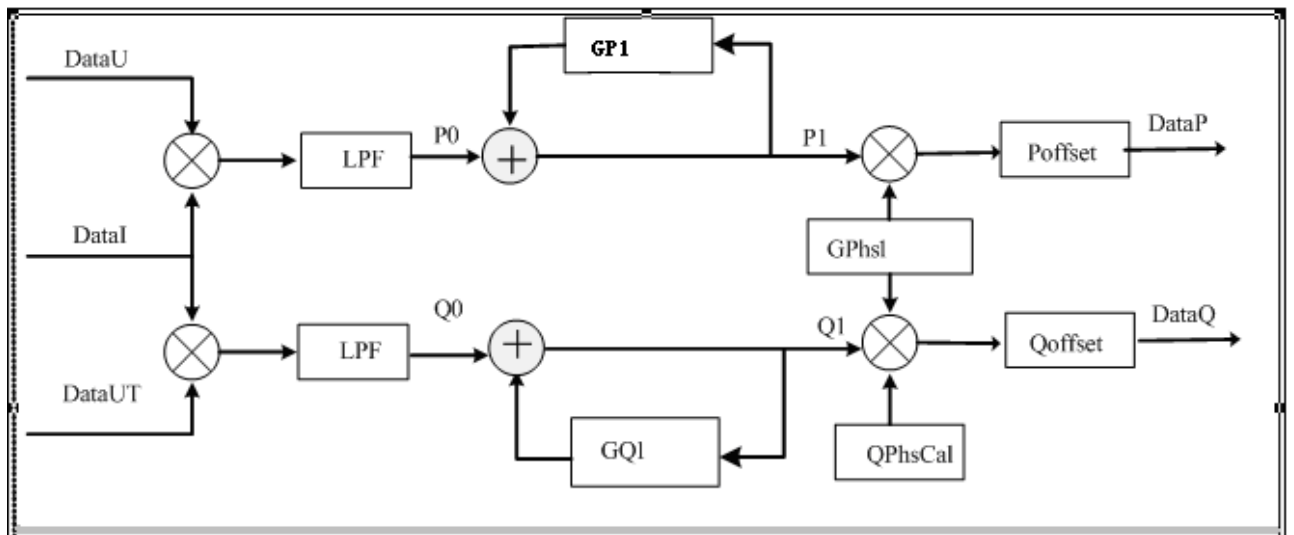


图 4-3-1 功率计算和补偿

有功功率通过电压、电流相乘经过低通滤波器后得到。

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

### 4.3.4 有效值

ATT7039 同时输出一路电流和一路电压的有效值。

有效值可以保证 0.5%的精度。

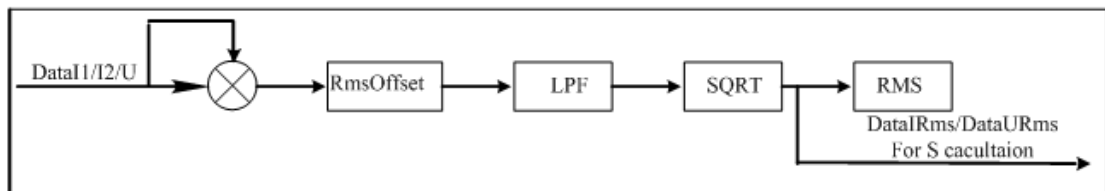


图 4-3-2

### 4.3.5 电压线频率

通过对电压采样信号过零点计数的方式，提供电压频率输出。

频率的精度可以保证准确到 0.01Hz。

### 4.3.6 起动潜动

通过设置寄存器 PQStart(4EH)，用户可以灵活的完成防潜动与起动的功能。

|P|小于 PQStart 时，PF 不输出脉冲。

|Q|小于 PQStart 时，QF 不输出脉冲。

另外状态寄存器 EMUSR(40H)的 NOQLD NOPLD 能够实时显示电能是否起动，方便用户对阈值的选取。

### 4.3.7 功率反向指示

通过状态寄存器 EMUSR(40H)的 REVQ、REVP 可以指示无功功率、有功功率是否反向。

REVQ、REVP 在 PF、QF 发脉冲的开始沿同步更新。

### 4.3.8 直流偏置校正

用户通过 CHNLCR(52H)的 HPFONU、HPFONI1 可以分别控制电压和电流的高通滤波器环节是否加上。当这些位为 1 时，表示高通开启，反之高通关闭。

高通滤波器关闭后，可以通过偏置校正寄存器 I1Off(4BH)、UOff(4DH)对采样到的数据进行偏置校正。用户可以手动或者自动完成偏置校正。

进行偏置校正时，需要将输入通道短接（即输入为 0）。

进行自动偏置校正时，在 AUTODC(54H)中，写入 0x01H，即可完成自动偏置校正，校完后，该寄存器变为 0x00H。新生成的校正值放在校正寄存器中。在进行自动偏置校正过程中（AUTODC=1），用户无法对校正寄存器进行操作。在 femu 为 5.505024MHz 时，自动偏置校正大约需要 0.6s。

进行手动偏置校正时，用户可以根据 ADC 采样数据 SPL\_I1(00H)、SPL\_U(02H)的多次平均值，设置相应的 OFFSET 值。

### 4.3.9 电能脉冲输出

ATT7039 提供有功能量寄存器 ENERGY\_P(0DH)、无功能量寄存器 ENERGY\_Q(0EH)，同时提供相应的脉冲输出引脚 PF、QF。

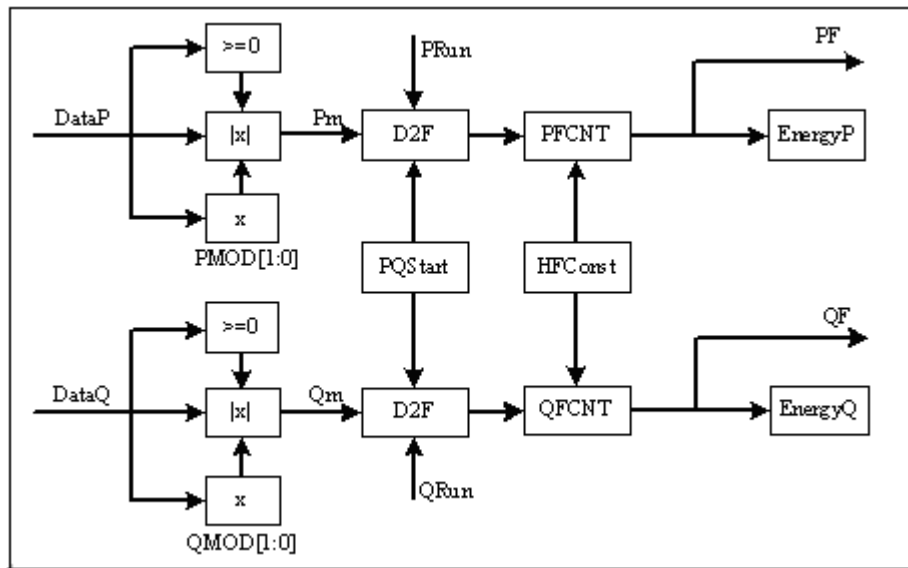


图 4-3-3 能量计量和脉冲输出

电能可以通过 EMCON(53H)的 QMOD、PMOD 选择正向计量、绝对值计量、代数和计量三种累加方



式。

内部功率值寄存器对功率进行累加，溢出后会发送一个溢出脉冲到快速脉冲寄存器 PFCNT(55H)、QFCNT(56H)。快速脉冲计数寄存器对溢出的次数进行累加计数。当快速脉冲寄存器中的计数绝对值大于等于输出脉冲频率设置寄存器 HFConst(4FH)的设置时，即发出一个 CF 脉冲，同时相应能量寄存器的值增加 1。

ATT7039 向用户开放了快速脉冲计数器 PFCNT/QFCNT，用户可以通过读写这些寄存器，防止下电时少计电能。

脉冲输出管脚 PF/QF 和能量寄存器受到 EMCON(53H)的 PRun/QRun 以及 PQStart(4EH)的控制。

当 PRun=0 或者|P|小于 PQStart 时，PF 不输出脉冲。

当 QRun=0 或者|Q|小于 PQStart 时，QF 不输出脉冲。

用户可以通过 POS (52H.6) 选择 PF/QF 的有效电平。POS 为 0 时，脉冲高电平有效；POS 为 1 时，脉冲低电平有效。

PF/QF 输出满足下面时序关系：

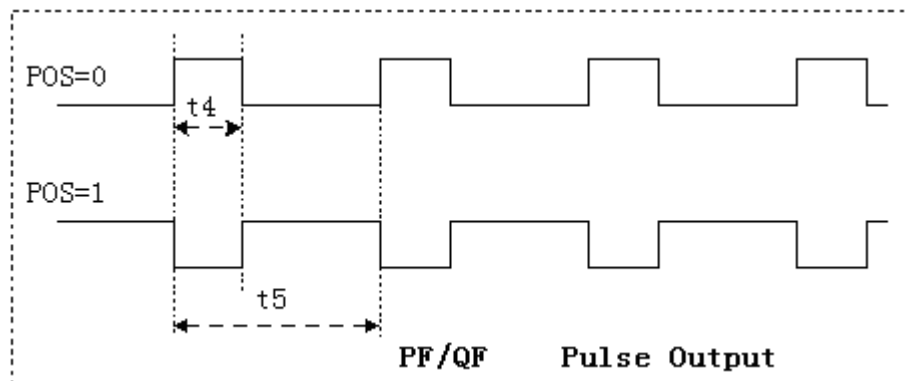


图 4-3-4 脉冲输出波形

时序特性：

参数	指标	单位	测试条件及注释
t4	90	ms	PF/QF 输出的高电平脉宽
t5	--	S	PF/QF 输出的周期

**注意：**当脉冲输出周期小于 180ms 时，脉冲以等 duty 形式输出。

ATT7039 可以实现输出脉冲宽度 (t4) 可调，通过 CFP[1:0](58H.[7:6])进行设置。

## 4.4 中断源

EMU 提供四个中断：PF\_IRQ、ADC\_IRQ 和 ZX\_IRQ、QF。

**PF\_IRQ：**当有有功脉冲输出时，EMU 向 CPU52 发出中断请求。

**ADC\_IRQ：**根据选定的采样频率，周期性向 CPU52 给出 ADC 中断请求。

**ZX\_IRQ：**当出现正向过零或者反向过零时，向 CPU52 发出过零中断。

通过寄存器 ZXD1 和 ZXD0(5BH)可以选择正向过零或者反向过零。

ZXD1=0, ZXD0=0: 表示选择正向过零点作为过零中断检测信号；

ZXD1=0, ZXD0=1: 表示选择负向过零点作为过零中断检测信号。

ZXD1=1, ZXD0=x: 表示选择双向过零点作为过零中断检测信号。

EMU 四个中断共用一个中断 IRQ\_EMU。通过 EMU 中断使能寄存器 EMUIE 和 EMU 中断标志寄存器 EMUIF，用户可以实现中断的控制和管理。

清除中断标志时，往 EMUIF 中相应的位写入 0 即可清除。

## 4.5 寄存器

EMU 包括两类寄存器，一类是 SFR 寄存器，即直接寄存器，用户可以通过 SFR 地址直接访问；另一类是计量参数和校表参数寄存器，是间接寄存器，用户需要通过直接寄存器间接访问。

### 4.5.1 特殊功能寄存器

表 4-13-1 EMU SFR 寄存器列表

地址	名称	复位值	功能描述
0x9F	SUPDC	0x00	内部模块使能控制寄存器
0xE9	ECADR	0x00	EMU 校表地址寄存器
0xEA	ECDATH	0x00	EMU 校表高字节数据寄存器
0xEB	ECDATL	0x00	EMU 校表低字节数据寄存器
0xEC	EPADR	0x00	EMU 参数地址寄存器
0xED	EPDATH	0x00	EMU 参数高字节数据寄存器
0xEE	EPDATM	0x00	EMU 参数中字节数据寄存器
0xEF	EPDATL	0x00	EMU 参数低字节数据寄存器
0xAC	EMUIE	0x00	EMU 中断使能寄存器 (IENE.3 需使能)
0xB5	EMUIF	0x00	EMU 中断标志寄存器

表 4-13-2 SUPDC 内部模块使能控制寄存器 (9FH)

SUPDC		Address: 9FH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	KBI_EN	X	LCD_EN	EMU_EN	I2C_EN	X		
Write:							X	X
Reset:	0	0	0	0	0	0	0	0

说明：EMU\_EN=1，使能EMU模块；EMU\_EN=0，关闭EMU模块。

注意：1，此寄存器带 REG0 型写保护，需连续对 BWPM 写入 CFH，BCH 才能改写此寄存器。

2，EMU 模块在 Battery 供电的情况下是固定关闭的。在 Battery 供电的情况下，写 EMU\_EN=1，是无法写进去的。

表 4-13-3 ECADR EMU Calibration Address Register (E9H)

EMU Calibration Address Register (ECADR)		Address: E9H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	ECADR7	ECADR6	ECADR5	ECADR4	ECADR3	ECADR2	ECADR1	ECADR0
Write:								
Reset:	0	0	0	0	0	0	0	0

表 4-13-4 ECDATH EMU Calibration Data High Byte Register (EAH)

EMU Calibration Data High Byte Register (ECDATH)			Address: <b>EAH</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	ECDATH7	ECDATH6	ECDATH5	ECDATH4	ECDATH3	ECDATH2	ECDATH1	ECDATH0
Write:								
Reset:	0	0	0	0	0	0	0	0

表 4-13-5 EMU ECDATAL Calibration Data Low Byte Register (EBH)

EMU Calibration Data Low Byte Register (ECDATL)			Address: <b>EBH</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	ECDATL7	ECDATL6	ECDATL5	ECDATL4	ECDATL3	ECDATL2	ECDATL1	ECDATL0
Write:								
Reset:	0	0	0	0	0	0	0	0

表 4-13-6 EPADR EMU Parameter Address Register (ECH)

EMU Parameter Address Register (EPADR)			Address: <b>ECH</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	EPADR7	EPADR6	EPADR5	EPADR4	EPADR3	EPADR2	EPADR1	EPADR0
Write:								
Reset:	0	0	0	0	0	0	0	0

表 4-13-7 EPDATH EMU Parameter Data High Byte Register (EDH)

EMU Parameter Data High Byte Register (EPDATH)			Address: <b>EDH</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	EPDATH7	EPDATH6	EPDATH5	EPDATH4	EPDATH3	EPDATH2	EPDATH1	EPDATH0
Write:	X	X	X	x	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

表 4-13-8 EPDATM EMU Parameter Data Middle Byte Register (EEH)

EMU Parameter Data Middle Byte Register (EPDATM)			Address: <b>EEH</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	EPDATM7	EPDATM6	EPDATM5	EPDATM4	EPDATM3	EPDATM2	EPDATM1	EPDATM0
Write:	X	X	X	x	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

表 4-13-9 EPDATL EMU Parameter Data Low Byte Register (EFH)

EMU Parameter Data Low Byte Register (EPDATL)			Address: <b>EFH</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	EPDATL7	EPDATL6	EPDATL5	EPDATL4	EPDATL3	EPDATL2	EPDATL1	EPDATL0
Write:	X	X	X	x	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

表 4-13-10 EMUIE EMU Interrupt Enable Register (ACH)

EMU Interrupt Enable Register (EMUIE)			Address: <b>ACH</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PFIE	QFIE	X	SPLIE	ZXIE	0	0	0
Write:						x	x	x
Reset:	0	0	0	0	0	0	0	0

注：同时需要使能 IENE.3，才能使设置的 EMU 中断使能有效。

位	功能描述
PFIE	<b>有功脉冲中断使能位</b> 1: 表示使能有功脉冲中断 0: 表示关闭有功脉冲中断
QFIE	<b>无功脉冲中断使能位</b> 1: 表示使能无功脉冲中断 0: 表示关闭无功脉冲中断
SPLIE	<b>ADC 原始数据刷新中断使能位</b> 1: 表示使能 ADC 原始数据刷新中断 0: 表示关闭 ADC 原始数据刷新中断
ZXIE	<b>电压/电流过零中断使能位</b> 1: 表示使能电压/电流中断 0: 表示关闭电压/电流中断

表 4-13-11 EMUIF EMU Interrupt Flag Register (B5H)

EMU Interrupt Flag Register (EMUIF)			Address: <b>B5H</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PFIF	QFIF	X	SPLIF	ZXIF	0	0	0
Write:						x	x	x
Reset:	0	0	0	0	0	0	0	0

各标志位对应的含义参见上表。

对中断标志写 0，可以将相应的中断标志清 0。如往 PFIF 写 0，则 PFIF 被清为 0。

#### 4.5.2 间接寄存器

间接寄存器包括计量参数寄存器和校表参数寄存器，描述如下：

##### 1. 计量参数只读寄存器 EPR: (Emu Parameter Register)

这些寄存器只能通过 EPADR 和 EPDATH/EPDATM/EPDATL 寄存器间接读取。

- 如果 EPR 寄存器为 3 字节的寄存器，那么 EPDATH/EPDATM/EPDATL 分别存放这 3 个字节的高、中、低位字节数据。
- 如果 EPR 寄存器为 2 字节的寄存器，那么 EPDATM/EPDATL 分别存放这 2 个字节的高、低位字节数据，而 EPDATH 为符号扩展位，即为 EPDATM.7 的扩展位。

##### 2. 校表参数设置寄存器 ECR: (Emu Calibration Register)

这些寄存器通过 ECADR 和 ECDATH/ECDATL 寄存器进行间接读写操作。

- 当 ECR 为 2 字节数据时，ECDATH 和 ECDATL 分别为 ECR 的高位和低位字节数据。
- 如果 ECR 为单字节数据，那么 ECDATL 为 ECR 的数据，而 ECDATH 字节数据被忽略。

**ECR 操作规则：**

- a、读 ECR 时，写地址到寄存器 ECADR，相应地址的 ECR 数据被放置到 ECDAT 中，供 CPU52 读取；
- b、写 ECR 时，先写地址寄存器 ECADR，然后写高字节数据 ECDATH（单字节数据可以忽略此操作），再写低字节数据 ECDATL。注意 16bit 写数据的操作顺序，当写 ECDATL 之后，16bit 数据就写入内部间接寄存器。

#### ECR 寄存器写保护：

只有当 EPADR=10100110 (0xA6H) 时，写 ECDATL 时，才能将 ECDAT 参数写到 ECR 寄存器中，否则写无效。

写保护打开后（即 EPADR=10100110），只要不改变 EPADR 寄存器的值，那么写保护打开就一直有效。

#### 4.5.2.1 计量参数寄存器列表

表 4-13-12 EPR 寄存器列表(Read Only)

地址 (EPADR)	名称	字节长度	缺省值	功能描述
00H	Spl_I1	2	0x0000	电流通道 1 的 ADC 采样数据
02H	Spl_U	2	0x0000	电压通道的 ADC 采样数据
03H	Spl_P	3	0x000000	有功功率波形数据
04H	Spl_Q	3	0x000000	无功功率波形数据
06H	Rms_I1	3	0x000000	电流通道 1 的有效值
08H	Rms_U	3	0x000000	电压通道的有效值
09H	Freq_U	2	0x0000	电压频率
0AH	Power_P1	3	0x000000	第一路有功功率
0BH	Power_Q1	3	0x000000	第一路无功功率
0DH	Energy_P	3	0x000000	有功能量
0EH	Energy_Q	3	0x000000	无功能量
12H	Energy_PC	3	0x000000	读后清 0 型有功能量寄存器
13H	Energy_QC	3	0x000000	读后清 0 型无功能量寄存器
15H	Scheck	3	0x000000	校验和寄存器

#### 波形采样输出寄存器：

表 4-13-13 Spl\_I1 Current Waveform Register (00H)

Current 1 Waveform Register (Spl_I1)			Address: 00H				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	SI1_15	SI1_14	SI1_13	SI1_12...SI1_3	SI1_2	SI1_1	SI1_0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

表 4-13-15 Spl\_U Voltage Waveform Register (02H)

Voltage Waveform Register (Spl_U)			Address: 02H				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	SU_15	SU_14	SU_13	SU_12...SU_3	SU_2	SU_1	SU_0

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

电流、电压波形采样值是二进制补码格式，为 16 位 ADC 的实际采样数据输出。其更新频率由 SPL（参见 ECR 寄存器 0x51H）确定。最快可以到 28.6kHz。

表 4-13-16 Spl\_P Active Power Waveform Register (03H)

Active Power Waveform Register (Spl_P)				Address: <b>03H</b>			
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	SP_23	SP_22	SP_21	SP_20...SP_3	SP_2	SP_1	SP_0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

表 4-13-17 Spl\_Q Reactive Power Waveform Register (04H)

Reactive Power Waveform Register (Spl_Q)				Address: <b>04H</b>			
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	SQ_23	SQ_22	SQ_21	SQ_20...SQ_3	SQ_2	SQ_1	SQ_0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

有效值输出寄存器:

表 4-13-19 I1Rms Current 1 Rms Register (06H)

Current 1 Rms Register (I1Rms)				Address: <b>06H</b>			
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	I1S23	I1S22	I1S21	I1S20...I1S3	I1S2	I1S1	I1S0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

表 4-13-21 URms Voltage Rms Register (08H)

Voltage Rms Register (Urms)				Address: <b>08H</b>			
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	US23	US22	US21	US20...US3	US2	US1	US0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

有效值 Rms 是 24 位的无符号数，最高位恒为 0。有效值以 femu/1572864 的频率更新，其中 femu=5.5MHz 为计量频率，那么有效值更新的频率为 3.5Hz，即每秒更新 3.5 次。

设寄存器读数为 RMSreg，实际的有效值为 RMS，转换系数为 Krms，则

$$RMS = RMSreg \times Krms$$

其中 Krms 为额定输入时额定值与相应寄存器的比值。

注：电压、两路电流的有效值计算都要各自的转换系数。

例：

设电流通道 1 输入额定 5A 电流时，RMSreg 的平均值为 0x039580 (234880)，则

$$Kp = 5/234880 = 2.1287466 \times 10^{-5}$$

当 RMSreg 的读数为 0x10000 (65536) 时，则实际的有效值 I1rms 为

$$I_{rms} = 65536 \times K_p = 1.3951A$$

### 电压频率测量寄存器:

表 4-13-22 UFREQ Voltage Frequency Register (09H)

Voltage Frequency Register (UFREQ)			Address: <b>09H</b>				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	Ufreq15	Ufreq14	Ufreq13	Ufreq12...Ufreq3	Ufreq2	Ufreq1	Ufreq0
Write:	X	X	X	X	X	X	X
Reset:	1	1	1	1	1	1	1

频率寄存器值是一个 16 位的无符号数，参数格式化公式为：

$$f = femu / 6 / UFREQ$$

例如，计量频率 femu=5505024Hz，UFREQ=18350，那么测量到的实际频率为：

$$f = 5505024 / 6 / 18350 = 50.00Hz。$$

电压频率测量值更新的周期为默认为 0.7s，当采样率加倍时，此寄存器更新周期减半。

### 功率参数输出寄存器:

表 4-13-23 PowerP Active Power Register (0AH)

Active Power Register (PowerP)			Address: <b>0AH</b>				
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	AP23	AP22	AP21	AP20...AP3	AP2	AP1	AP0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

表 4-13-24 PowerQ Reactive Power Register (0BH)

Reactive Power Register (PowerQ)			Address: <b>0BH</b>				
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	RP23	RP22	RP21	RP20...RP3	RP2	RP1	RP0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

功率参数 PowerP、PowerQ、PowerS 是二进制补码格式，24 位数据，其中最高位是符号位。

设寄存器中的数据为 PowerP，则供计算用的 Preg 为：

$$Preg = PowerP \quad \text{如果 } PowerP < 2^{23}$$

$$Preg = PowerP - 2^{24} \quad \text{如果 } PowerP \geq 2^{23}$$

设显示的有功功率为 P，转换系数为 Kpqs，则：

$$P = Preg \times Kpqs$$

Kpqs 为额定有功功率功率输入时，额定功率与 PowerP 读数的比值。

无功功率和视在功率做显示时的系数与有功功率的系数 Kpqs 相同。

例：

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

$$Kpqs = 1000 / 51673 = 0.01935$$

当 PowerP 读数为 0xFF4534 时，其代表的功率值为：

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

其中  $Preg = PowerP - 2^{24} = -47820$

功率参数以 femu/1572864 的频率更新，其中 femu 为计量频率，那么功率参数更新的频率为 3.5Hz，即每秒更新 3.5 次。

### 电能参数输出寄存器：

表 4-13-26 EnergyP Active Energy Register (0DH)

Active Energy Register (EnergyP)			Address: 0DH				
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	EP23	EP22	EP21	EP20...EP3	EP2	EP1	EP0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

表 4-13-27 EnergyQ Reactive Energy Register (0EH)

Reactive Energy Register (EnergyQ)			Address: 0EH				
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	EQ23	EQ22	EQ21	EQ20...EQ3	EQ2	EQ1	EQ0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

电能参数是 24 位无符号数，EnergyP/EnergyQ/EnergyS 寄存器值分别代表 PF/QF/SF 脉冲的累加个数；寄存器最小单位代表的能量为 1/EC kWh，其中 EC 为电表常数。

例：

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

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

表 4-13-31 读后清 0 型有功能量寄存器 EnergyPC (12H)

Active Energy Register (EnergyPC)			Address: 12H				
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	EP23	EP22	EP21	EP20...EP3	EP2	EP1	EP0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

表 4-13-32 读后清 0 型无功能量寄存器 EnergyQC (13H)

Reactive Energy Register (EnergyQC)			Address: 13H				
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	EQ23	EQ22	EQ21	EQ20...EQ3	EQ2	EQ1	EQ0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

表 4-13-34 参数和校验寄存器 Scheck (15H)



SumChecksum Register (Scheck)			Address: 15H				
	Bit23	22	21	20...3	2	1	Bit0
Read:	Scheck23	Scheck22	Scheck21	Scheck20..... Scheck3	Scheck2	Scheck1	Scheck0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

参数和校验寄存器为所有校表参数寄存器的和：41H--64H。55H, 56H 两个寄存器除外对所有的校验寄存器采用三字节的无符号数加法，两单字节寄存器高位补0。

#### 4.5.2.2 校表参数寄存器列表

表 4-13-35 ECR 寄存器列表 (Read/Write)

地址 (ECADR)	名称	字节长度	默认值	功能描述
40H	EMUSR	1	0x00	EMU 状态标志寄存器
41H	GP1	2	0x0000	通道 1 的有功功率增益校正寄存器
42H	GQ1	2	0x0000	通道 1 的无功功率增益校正寄存器
44H	GPhs1	2	0x0000	通道 1 的相位增益校正寄存器
49H	QPhsCal	1	0x00	无功相位补偿寄存器
4BH	I1Off	2	0x0000	电流通道 1 的直流偏置校正寄存器
4DH	UOff	2	0x0000	电压通道的直流偏置校正寄存器
4EH	PQStart	2	0x0040	起动功率设置寄存器
4FH	HFCnst	2	0x0080	高频脉冲常数设置寄存器
51H	ADCCFG	1	0x00	ADC 控制寄存器
52H	CHNLCR	1	0x05	通道控制信号寄存器
53H	EMCON	1	0xB0	能量累加控制寄存器
54H	AutoDC	1	0x00	直流自动偏置校正寄存器
55H	PFCnt	2	0x0000	快速有功脉冲计数寄存器
56H	QFCnt	2	0x0000	快速无功脉冲计数寄存器
58H	ADCCON	1	0x00	ADC 通道增益选择寄存器
5AH	DGAIN	1	0x00	通道数字增益寄存器
5BH	Emu_ctrl	1	0x40	EMU 控制寄存器
5CH	P1OFFSET	1	0x00	通道 1 有功功率补偿校正寄存器
5EH	Q1OFFSET	1	0x00	通道 1 无功功率补偿校正寄存器
60H	I1RMSOFFSET	2	0x0000	通道 1 有效值补偿校正寄存器
62H	URMSOFFSET	2	0x0000	电压通道有效值补偿校正寄存器
63H	Rosi_ctrl	1	0x02	罗氏线圈使能控制寄存器
65H	SRSTREG	1	0x00	校表参数复位寄存器

表 4-13-35 EMUSR EMU 状态标志寄存器 (40H)

EMU Status Register (EMUSR)			Address: 40H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	PEOF	QEOF	0	NoQLd	NoPLd	REVQ	REVP
Write:				X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PEOF	有功电能寄存器溢出标志 1: Energy_PC发生溢出, 写0清零; 0: Energy_PC没有溢出
QEOF	无功电能寄存器溢出标志 1: Energy_QC发生溢出, 写0清零; 0: Energy_QC没有溢出
NoQLd	无功潜动标志 1: 无功功率小于起动功率 0: 无功功率大于/等于起动功率
NoPLd	有功潜动标志 1: 有功功率小于起动功率 0: 有功功率大于/等于起动功率
REVQ	反向无功功率指示标识信号 1: 检测到负无功功率 0: 没有检测到负无功功率, 或者当再次检测到正无功功率时, 该信号为0。在QF发脉冲时的开始沿更新该值。
REVP	反向有功功率指示标识信号 1: 检测到负有功功率 0: 没有检测到负有功功率, 或者当再次检测到正有功功率时, 该信号为0。在PF发脉冲时的开始沿更新该值。

### 功率校正寄存器:

表 4-13-36 通道 1 有功功率增益校正寄存器 GP1 (41H)

Active Power Gain 1 Register (GP1)			Address: 41H				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	GP1_15	GP1_14	GP1_13	GP1_12...GP1_3	GP1_2	GP1_1	GP1_0
Write:							
Reset:	0	0	0	0	0	0	0

表 4-13-37 通道 1 无功功率增益校正寄存器 GQ1 (42H)

Reactive Power Gain1 Register (GQ1)			Address: 42H				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	GQ1_15	GQ1_14	GQ1_13	GQ1_12...GQ1_3	GQ1_2	GQ1_1	GQ1_0
Write:							
Reset:	0	0	0	0	0	0	0

表 4-13-39 通道 1 相位增益校正寄存器 GPhs1 (44H)

Phase Calibration1 Register (GPhs1)			Address: <b>44H</b>				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	GPS1_15	GPS1_14	GPS1_13	GPS1_12...GPS1_3	GPS1_2	GPS1_1	GPS1_0
Write:							
Reset:	0	0	0	0	0	0	0

表 4-13-44 无功相位补偿寄存器 QPhsCal (49H)

Reactive Power Phase Calibration Register (QPhsCal)			Address: <b>49H</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	QPC7	QPC6	QPC5	QPC4	QPC3	QPC2	QPC1	QPC0
Write:								
Reset:	0	0	0	0	0	0	0	0

无功相位补偿寄存器采用二进制补码形式，最高位为符号位。

当 femu 为 5.5MHz，电压频率为 50Hz 时，该寄存器保持默认值 0 即可，当电压频率为其它值时，需要按照下面的公式进行校正：

标准表在无功 0.5L 下（即 U、I 夹角为 30 度）测得无功精度误差为 Err%，通过如下公式计算得到 Qphscal 值：

$$Result = \frac{Err\% \times 32768}{1.732}$$

如果 Result ≥ 0，则 Qphscal = Result；

如果 Result < 0，则 Qphscal = 256 + Result；

#### 通道直流偏置校正寄存器：

表 40-13-46 电流通道 1 直流偏置校正寄存器 IIOff (4BH)

Current 1 Offset Register (IIOff)			Address: <b>4BH</b>				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	IIOS15	IIOS14	IIOS13	IIOS12...IIOS3	IIOS2	IIOS1	IIOS0
Write:							
Reset:	0	0	0	0	0	0	0

表 4-13-48 电压通道直流偏置校正寄存器 UOff (4DH)

Voltage Offset Register (UOff)			Address: <b>4DH</b>				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	UOS15	UOS14	UOS13	UOS12...UOS3	UOS2	UOS1	UOS0
Write:							
Reset:	0	0	0	0	0	0	0

0x4BH~0x4DH 寄存器是二进制补码格式，最高位为符号位。

其最小单位与 ADC 输出的 16 位数据的最小单位一致。通道直流偏置校正只在高通环节关闭后使用。

#### 潜动与启动：

表 4-13-49 起动功率设置寄存器 PQStart (4EH)

Start Power Threshold Setup Register (PQStart)			Address: <b>4EH</b>					
	Bit15	14	13	12 ... 7	6	5...2	1	Bit0
Read:	PQS15	PQS 14	PQS 13	PQS 12...PQS 7	PQS 6	PQS 5...PQS 2	PQS 1	PQS 0
Write:								
Reset:	0	0	0	0	1	0	0	0

PQStart 是 16 位无符号数，做比较时，将其作为低 16 位与 P/Q (PowerP 0x0AH / PowerQ 0x0BH，均为 24bit 有符号数) 的绝对值进行比较，以作起动的判断。

|P| 小于 PQStart 时，PF 不输出脉冲。

|Q| 小于 PQStart 时，QF 不输出脉冲。

(|P|/|Q| 同时都小于 PQStart) 时，SF 不输出脉冲。

### 脉冲频率：

表 4-13-50 高频脉冲常数设置寄存器 HFConst (4FH)

High Frequency Impulse Const Register (HFConst)			Address: <b>4FH</b>					
	Bit15	14	13	12 ... 8	7	6...2	1	Bit0
Read:	0	HFC14	HFC13	HFC12...HFC8	HFC7	HFC6...HFC2	HFC1	HFC0
Write:	X							
Reset:	0	0	0	0	1	0	0	0

HFConst 是 15 位无符号数，做比较时，将其作为低 15 位与快速脉冲计数寄存器 0x55H~0x57H 寄存器值的绝对值做比较，如果大于等于 HFConst 的值，那么就会有对应的 PF/QF/SF 脉冲输出。

HFConst 的默认值是 0x0080。

### ADC 控制寄存器：

表 4-13-52 ADC 控制寄存器 ADCCFG (51H)

ADC Config Register (ADCCFG)			Address: <b>51H</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	SPL2	SPL1	SPL0
Write:								
Reset:	0	0	0	1	0	0	0	0

位	功能描述			
SPL[2:0]	波形采样中断频率选择			
	当 femu=5.505024MHz，选择的频率如下表所述。			
	SPL2	SPL1	SPL0	波形采样频率
	0	0	0	1.79k Hz (femu/3072)
	0	0	1	3.58 kHz (femu/1536)
	0	1	0	7.168k Hz (femu/768)
	0	1	1	14.336k Hz (femu/384)
	1	x	x	28.67k Hz (femu/192)

**通道设置:**

表 4-13-55 通道控制寄存器 CHNLCR (0x52H)

Current Channel Control Register (CHNLCR)			Address: <b>52H</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	POS	X	X	X	HPFONU	X	HPFONII
Write:								
Reset:	0	0	0	0	0	1	0	1

位	功能描述
POS	<b>脉冲有效电平选择位</b> 1: 表示PF/QF/SF为低电平有效; 0: 表示PF/QF/SF为高电平有效。
HPFONU	<b>电压通道数字高通滤波使能位</b> 0: 关闭数字高通滤波器; 1: 启动数字高通滤波器;
HPFONII	<b>电流通道1数字高通滤波使能位</b> 0: 关闭数字高通滤波器; 1: 启动数字高通滤波器;

**能量计算:**

表 4-13-56 能量累加控制寄存器 EMCON (53H)

Energy Measure Control (EMCON)			Address: <b>53H</b> default: <b>0xF0</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PSRR_REG	X	QRun	PRun	QMOD1	QMOD0	PMOD1	PMOD0
Write:								
Reset:	1	0	1	1	0	0	0	0

位	功能描述							
PSRR_REG	<b>PSRR选择位</b> 1: 打开PSRR电源模块 0: 关闭PSRR电源模块							
QRun	<b>无功能量累加使能位</b> 1: 允许无功计量 0: 停止无功计量							
PRun	<b>有功能量累加使能位</b> 1: 允许有功计量 0: 停止有功计量							
QMOD[1: 0]	<b>无功能量累加模式选择位</b>							
	<table border="1"> <thead> <tr> <th>QMOD1</th> <th>QMOD0</th> <th>累加功率Qm</th> <th>无功能量累加方式选择</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Qm=DataQ</td> <td>脉冲能量按代数和方式对无功功率累加</td> </tr> </tbody> </table>	QMOD1	QMOD0	累加功率Qm	无功能量累加方式选择	0	0	Qm=DataQ
QMOD1	QMOD0	累加功率Qm	无功能量累加方式选择					
0	0	Qm=DataQ	脉冲能量按代数和方式对无功功率累加					

	0	1	$DataQ \geq 0, Qm = DataQ;$ $DataQ < 0, Qm = 0$	脉冲能量只累加正向无功功率, 不累加负向无功功率
	1	0	$Qm =  DataQ $	脉冲能量时按绝对值方式对无功功率累加
	1	1	$Qm = DataQ$	脉冲能量按代数和方式对无功功率累加
PMOD[1: 0]	<b>有功能量累加模式选择位</b>			
	PMOD1	PMOD0	累加功率Pm	有功能量累加方式选择
	0	0	$Pm = DataP$	脉冲能量按代数和方式对有功功率累加
	0	1	$DataP \geq 0, Pm = DataP;$ $DataP < 0, Pm = 0$	脉冲能量只累加正向有功功率, 不累加负向有功功率
	1	0	$Pm =  DataP $	脉冲能量按绝对值方式对有功功率累加
	1	1	$Pm = DataP$	脉冲能量按代数和方式对有功功率累加

直流偏置校正:

表 4-13-57 直流自动偏置校正寄存器 AutoDC (54H)

Auto Offset Calibration (AutoDC)			Address: <b>54H</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	0	0	0	0	AUTO
Write:	X	X	X	X	X	X	X	
Reset:	0	0	0	0	0	0	0	0

AUTO=1: 表示使能直流偏置校正, 校正结束时自动清为0。

快速脉冲计数器:

表 4-13-58 有功快速脉冲寄存器 PFCNT (55H)

Active Energy Counter Register (PFCNT)			Address: <b>55H</b>				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	PFC15	PFC14	PFC13	PFC12...PFC3	PFC2	PFC1	PFC0
Write:							
Reset:	0	0	0	0	0	0	0

表 4-13-59 无功快速脉冲寄存器 QFCNT (56H)

Reactive Energy Counter Register (QFCNT)			Address: <b>56H</b>				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	QFC15	QFC14	QFC13	QFC12...QFC3	QFC2	QFC1	QFC0
Write:							
Reset:	0	0	0	0	0	0	0

为了防止上下电时丢失电能，掉电时 MCU 将寄存器 PFCnt/QFCnt/SFCnt 值读回并进行保存，然后在下次上电时 MCU 将这些值重新写入到 PFCnt/QFCnt/SFCnt 中去。

当快速脉冲计数寄存器 PFCnt/QFCnt/SFCnt 计数的值大于等于 HFconst 时，相应的 PF/QF/SF 会有脉冲溢出，能量寄存器 0x0DH~0x0FH 寄存器的值会相应的加 1。

### 通道增益选择:

表 4-13-61 ADC 通道增益控制寄存器 ADCCON (58H)

ADC Channel Gain (ADCCON)			Address: 58H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	CFP1	CFP0	X	X	PGA1	PGA0	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述		
CFP[1:0]	脉宽选择位 即脉冲输出的t4参数，见PF/QF/SF时序特性，		
	CFP1	CFP0	脉宽 (femu=5.505024MHz)
	0	0	90ms
	0	1	45ms
	1	0	22.5ms
PGA[1: 0]	电流通道模拟增益选择位		
	PGA1	PGA0	电流通道1
	0	0	PGA=1
	0	1	PGA=2
	1	0	PGA=8
	1	1	PGA=16

### 通道数字增益:

表 4-13-63 通道数字增益寄存器 DGAIN (5AH)

Channel Digital Gain (DGAIN)			Address: 5AH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	X	X	DGI1	DGI0	DGI1	DGI0
Write:	X	X						
Reset:	0	0	0	0	0	0	0	0

注：该寄存器类似通道增益选择，但是其增益是通过放大 ADC 后的数字信号来实现的，放大倍率为 1/2/4/8。

位	功能描述		
DGI[1:0]	电流通道数字增益选择位		
	DGI1	DGI0	I1 数字增益
	0	0	DG=1
	0	1	DG=2

	1	0	DG=4
	1	1	DG=8
DGU[1:0]	电压通道数字增益选择位		
	DGU 1	DGU 0	电压通道数字增益
	0	0	DG=1
	0	1	DG=2
	1	0	DG=4
	1	1	DG=8

表 4-13-64 EMU 控制寄存器 emu\_ctrl (5BH)

clk and adc control (emu_ctrl)		Address: <b>5BH</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	Adc_ilon	X	Adc_uon	Zxd1	Zxd0	Emu_lkctrl1	Emu_lkctrl0
Write:								
Reset:	0	1	0	0	0	0	0	0

位	功能描述		
Adc_ilon	<b>I1_ADC使能位</b> 1: 打开第一路电流ADC 0: 关闭第一路电流ADC		
Adc_uon	<b>U_ADC使能位</b> 1: 打开电压ADC 0: 关闭电压ADC		
Zxd[1:0]	过零中断方式选择控制位		
	ZXD1	ZXD0	过零中断方式
	0	0	正向过0中断
	0	1	负向过0中断
	1	x	双向过0中断
Emu_lkctrl[1:0]	计量模块ADC时钟频率选择控制位		
	计量模块ADC时钟频率fadc由femu分频获得, femu为固定频率5.505024MHz		
	Emu_lkctrl1	Emu_lkctrl0	fadc
	0	0	femu/6=917.5kHz
	0	1	femu/12=458.8kHz
	1	x	femu/24=229.4kHz

表 4-13-65 通道 1 有功功率偏置校正寄存器 P1OFFSET (5CH)

Power offset 1 (P1OFFSET)		Address: <b>5CH</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:	P1OFFSET	P1OFFSET	P1OFFSET	P1OFFSET	P1OFFSET	P1OFFSET	P1OFFSET	P1OFFSET
Write:	7	6	5	4	3	2	1	0
Reset:	0	0	0	0	0	0	0	0

表 4-13-67 Q1OFFSET 通道 1 无功功率偏置校正寄存器 (5EH)



Reactive Power offset 1 (Q1OFFSET)			Address: <b>5EH</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	Q1OFFSET	Q1OFFSET	Q1OFFSET	Q1OFFSET	Q1OFFSET	Q1OFFSET	Q1OFFSET	Q1OFFSET
Write:	7	6	5	4	3	2	1	0
Reset:	0	0	0	0	0	0	0	0

功率偏置校正寄存器采用二进制补码形式

表 4-13-69 IIRMSOFFSET 通道 1 电流有效值偏置校正寄存器 (60H)

IIRMSOFFSET			Address: <b>60H</b>					
	Bit15	14	13	12.....4	3	2	1	Bit0
Read:	IIRMSOFF	IIRMSOFF	IIRMSOFF	IIRMSOFF	IIRMSOFF	IIRMSOFF	IIRMSOFF	IIRMSOFF
Write:	SET15	SET14	SET13	SET12...4	SET3	SET2	SET1	SET0
Reset:	0	0	0	0	0	0	0	0

电流通道 1 有效值校正寄存器采用二进制无符号数形式。

当输入信号为 0 的时候，读取 IIRMS 的值，然后通过如下公式计算出 IIRMS offset 寄存器写入值。

$$\text{公式: } IIRMSOFFSET = \frac{(IIRMS)^2}{2^{15}}$$

表 4-13-71 URMSOFFSET 电压有效值偏置校正寄存器 (62H)

URMSOFFSET			Address: <b>62H</b>					
	Bit15	14	13	12.....4	3	2	1	Bit0
Read:	URMSOFF	URMSOFF	URMSOFF	URMSOFF	URMSOFF	URMSOFF	URMSOFF	URMSOFF
Write:	SET15	SET14	SET13	SET12...4	SET3	SET2	SET1	SET0
Reset:	0	0	0	0	0	0	0	0

电压通道有效值校正寄存器采用二进制无符号数形式。校验方式同 IIRMSOFFSET。

表 4-13-72 ROSI\_CTRL 罗氏线圈控制寄存器(63H)

Rosi_ctrl register			Address: <b>63H</b> <b>default: 0x62</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	Reserved	Reserved	Reserved	X	X	X	Reserved	Rosi_i1_en
Write:								
Reset:	0	0	0	0	0	0	1	0

位	功能描述
Reserved	注：bit[7..5]必须设置为[000],bit[1]=1
Rosi_i1_en	电流通道1的罗氏线圈使能位 1：使能电流通道1罗氏线圈功能 0：关闭电流通道1罗氏线圈功能

注：罗氏线圈使能是打开内部滤波器，将罗氏线圈送进来的微分信号进行积分还原，故外部通道使用罗氏线圈，一定要开启这一通道的罗氏线圈使能。

表 4-13-74 EMU 复位寄存器 SRSTREG (65H)

Soft reset Register (SRSTREG)			Address: 65H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	SRST7	SRST 6	SRST 5	SRST 4	SRST 3	SRST 2	SRST 1	SRST 0
Write:								
Reset:	0	0	0	0	0	0	0	0

注：用于复位校表寄存器，当此寄存器写入 0x55 会导致校表寄存器发生复位，复位后该寄存器自动清 0。

## 4.6 校表过程

### 1. 高频脉冲常数设置(同一批表只需同样的 HFCONST)

通过 HFConst 寄存器将用户样表的误差精度调整到 15% 以内。有两种方式计算。

#### 方案一：

HFCONST 寄存器的默认值为 0x0080。

用户观察电表的初始误差为 Err，则按照下面公式将误差调整到 10% 以内：

$$\text{HFCONST} = 0x0080 * (1 + \text{Err})$$

#### 举例说明：

电表表常数 (EC) 设置为 3200，功率因数为 1，HFCONST 寄存器为默认值 0x0080，观察标准表上显示的误差为 62.5%。

$$\text{根据公式：HFCONST} = 0x0080 * (1 + \text{Err})$$

$$\text{计算得到：HFCONST} = 0x0080 * (1 + 62.5\%) = 0x00D0$$

将 0x00D0 写入 ATT7039 的 HFCONST (4FH) 寄存器：

实际：Write(0x4F, 0x00D0)，写入后标准表的显示误差应该在 10% 以内

#### 方案二：

femu=917.5kHz 时

$$\text{HFConst} = 1.168 * \text{Vu} * \text{Vi} * 10^{11} / (\text{EC} * \text{Un} * \text{Ib})$$

Vu：额定电压输入时，电压通道的电压（引脚上电压×放大倍数）

Vi：额定电流输入时，电流通道的电压（引脚上电压×放大倍数）

Un：额定输入的电压

Ib：额定输入的电流

EC：电表常数

femu 为其他值时，HFConst 按比例变化即可。

#### 举例说明：

假设设计一块 220v (Un)、5A (Ib) 额定输入、表常数为 3200 (EC) 的样表。

电流采样使用 350 微欧的锰铜，选择通道 1，模拟通道增益为 16 倍；

电压采用电阻分压输入，模拟通道增益为 1 倍，芯片引脚上电压值为 0.22v；

选择 femu=5.505024MHz。

根据公式解：

$$\text{Vu} = 0.22\text{v}$$

$$\text{Vi} = 5 * 0.00035 * 16 = 0.028\text{v}$$

$$\text{HFConst} = [1.168 * \text{Vu} * \text{Vi} * 10^{11} / (\text{EC} * \text{Un} * \text{Ib})]$$

$$= [1.168 * 0.220 * 0.028 * 10^{11} / (3200 * 220 * 5)]$$

=204

取整后 HFConst 为 0xCC(204)。将该值写入 HFConst(4FH)寄存器即可

实际: Write(0x4F, 0x00CC), 写入后标准表的显示误差应该在 10% 以内。

## 2. 第一通道有功、无功增益校正

只需要在额定输入、功率因数为 1 时根据有功计算。通常有功、无功和视在增益写入相同的值。

已知:

标准表上读出误差为 err

计算公式:

$$P_{gain} = \frac{-err}{1+err}$$

如果  $P_{gain} \geq 0$ , 则  $GP1 = INT[P_{gain} * 2^{15}]$

否则  $P_{gain} < 0$ , 则  $GP1 = INT[2^{16} + P_{gain} * 2^{15}]$

同时将计算出的 GP1 的值写入到 GQ1。

**举例说明:**

功率源上输出 220v、5A、功率因数为 1 的信号, 标准表上显示的误差为 3.8%

则  $P_{gain} = -0.038 / (1 + 0.038) = -0.0366$

该数小于 0, 需转换为补码, 则  $-0.0366 * 2^{15} + 2^{16} = 0xFB50H$

将 0xFB50H 同时写入 GP1(41H)/GQ1(42H)/GS1(43H)寄存器, 即完成阻性增益校正。

## 3. 第一通道相位校正

在增益已经校正好之后, 进行相位补偿。在功率因素 0.5L 处进行校正。

已知: 0.5L 处标准表误差读数为 err

相位补偿公式:

$$\theta = \frac{-err}{1.732}$$

如果  $\theta \geq 0$ ,  $GPhs = \theta * 2^{15}$

否则  $\theta < 0$ ,  $GPhs = 2^{16} + \theta * 2^{15}$

**举例说明:**

校完阻性增益后, 将功率因数改为 0.5L, 标准表显示的误差为 -0.4%, 则

$\theta = -(-0.004) / 1.732 = 0.0023$

$Gphs1 = 0.0023 * 2^{15} = 75.3$

取整后为 0x4BH, 写入角度校正寄存器 Gphs1(44H)即可。

## 4. Poffset 校正 (小信号有功功率校正)

在经过以上步骤之后, 用户在  $I_b = 100\%$  的时候电表误差校正到 0 附近;

观察小信号  $x\%I_b$  (5%, 2%) 点的电表误差为 Err%,

$x\%I_b$  点在阻性下读取标准表上输出的有功功率值 Preal

应用公式来计算:

---

$$P\_offset = \frac{Preal * EC * HFConst * 2^{23} * (-Err\%)}{1.0322 * 10^{11}}$$

**举例说明:**

额定电压 220V, 额定电流(Ib)5A, 表常数为 3200, 快速脉冲寄存器 (HFCONST) 读取为 0x66;

电表在 Ib = 100%时误差校正正在 0 附近, 观察小信号 5%点的电表误差为 0.5%;

从标准表上读取小信号 5%点的输出功率为 55.02 (Preal 为标准表上的显示功率输出)

根据公式计算得到:

$$\begin{aligned} Poffset &= (Preal * EC * HFCONST * 2^{23} * (-Err\%)) / (1.0322 * 10^{11}) \\ &= (55.02 * 3200 * 102 * 2^{23} * (-0.5\%)) / (1.0322 * 10^{11}) \\ &= -7.3 \end{aligned}$$

由于 Poffset < 0, 所以写入寄存器 P1offset 的值为  $2^8 + Poffset = 249$

取个整数 249(0xF9)

将 0xF9 写入 ATT7039 的 P1offset(5CH)寄存器:

## 5 其他外围功能

### 5.1 WDT

#### 5.1.1 概述

Watchdog Timer 是一个特殊的定时器，计时满预定时间则发出溢出脉冲，产生 WDTR 复位信号；在溢出脉冲发生前将 Watchdog Timer 清零，则不会发出 WDTR 复位。特点如下：

- 采用硬件狗设计
- SLEEP模式下WDT开启/关闭可选
- 可以通过外部引脚 JTAG\_WDTEN进行控制

#### 5.1.2 工作模式

WDTEN和外部引脚 JTAG\_WDTEN配合使用。当JTAG\_WDTEN拉高时，无论WDTEN位等于0还是等于1，WDT均被使能；当JTAG\_WDTEN拉低时，WDTEN=1，使能WDT计数，WDTEN=0，表示关闭WDT计数，并且将WDT Counter进行清零。

JTAG_WDTEN	WDTEN	WDT 模块 (0/off, 1/on)
0	0	0
0	1	1
1	0	1
1	1	1

在 Sleep 模式下：

当 WDTCON 中的 WDT\_SH=0，在 SLEEP 模式下屏蔽掉 WDT 的 CLK，WDT 功能无效，并且同时将 WDT Counter 进行清零。

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

WDT\_SH 的控制位对应到 WDT 的功能控制位：

WDT_SH	WDT 功能在 SLEEP Mode 下的状态
1	开启
0	关闭

#### 5.1.3 特殊功能寄存器

表 5-1-1 WDTCON WDT 控制寄存器 (C9H)

WDTCON			Address: C9H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	WDTS2	WDTS1	WDTS0	WDTEN	WDT_SH	WDTCLR2	WDTCLR1	WDTCLR0
Write:								
Reset:	1	0	0	0	1	0	0	0

注：该位置的置位在SLEEP下有效的，才能够屏蔽WDT的CLK。

位	功能描述			
WDTS[2:0]	<b>WDT 溢出时间选择位</b>			
	WDTS2	WDTS1	WDTS0	溢出时间
	0	0	0	31.25ms
	0	0	1	62.5ms
	0	1	0	125ms
	0	1	1	250ms
	1	0	0	500ms
	1	0	1	1.00s
	1	1	0	2.00s
	1	1	1	4.00s
WDTEN	<b>WDT使能位</b> 1: WDT模块打开 0: WDT模块关闭，当WDT模块关闭后，计数器清零，控制寄存器值保持不变。			
WDTSH	<b>WDT在sleep模式使能位</b>			
	WDT_SH	WDT 功能在SLEEP或者HOLD Mode下的状态		
	1	开启		
	0	关闭		
WDTCLR[2:0]	<b>WDT清零控制位</b> 往 WDTCLR[2:0]写入数据 101, 可以将 WDT 进行清零。写其他数据无效。			

注意：WDT 溢出时间即时生效，当用户先写入一个长的溢出时间，而后写入一个短的溢出时间，此时如果计数器计数已经超过这个短的溢出时间，即刻发生复位。

## 5.2 KBI

### 5.2.1 概述

键盘单元支持 2 个独立的可屏蔽的外部中断，当 P1[1..0] 端口被配置成键盘模式时，Key0, Key1 内部上拉 30K 电阻有效。

- 键盘输入引脚 KEY 内部上拉 30K 电阻
- Key0, Key1 都可以配置为上升沿或者下降沿触发有效
- 支持按键唤醒退出 SLEEP

### 5.2.2 按键中断

#### 5.2.2.1 按键中断产生

当 KEY[1: 0] 的功能 PIN 上出现下降沿或上升沿（由 KEYIE 配置），并保持 4 个  $f_{osc}$  的电平时间，相应的 KEYIF 置位，KEYIF[X]=1。如果对应的 KEYIF[X]=1 中断使能有效，EKBI\_SPI=1（IEN1.1, KEY 对应的扩展功能中断有效），EA=1（全局中断有效），就可以触发 KEY 中断。

#### 5.2.2.2 按键中断清除

对 KEYIF 中对应的 KEY<sub>x</sub> 的中断标志位，写 0 清中断标志。

### 5.2.3 按键唤醒低功耗模式

#### 5.2.3.1 按键唤醒 Sleep 模式

支持按键唤醒退出 SLEEP 模式。在 SLEEP MODE 下，当 KEY/GPIO 功能复用 PIN 配置为 KEY 功能的时候，当 KEY[1..0] 的功能 PIN 上出现下降沿上升沿（上升沿或者下降沿触发由 KEYIE 配置），保持 8 个  $f_{osc}$  的低（高）电平时间，相应的 KEYIF 置位，KEYIF [X] =1。如果对应的 KEYIF [X] =1 中断使能有效，就可以触发 KEY 唤醒。

### 5.2.4 特殊功能寄存器

表 5-3-1 KEY SFR 寄存器列表

地址	名称	复位值	功能描述
0xAF	KEYIE	0x30	按键中断控制寄存器
0xB6	KEYIF	0x00	按键中断标志寄存器
0xB8	IEN1	0x00	中断使能寄存器，（详细内容参见 3.3 中断系统）

表 5-3-2 KEYIE 按键中断控制寄存器（AFH）

KEY Interrupt Enable Register (KEYIE)			Address: AFH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	KEY1 SEL	KEY0 SEL	X	X	KEYIE1	KEYIE0
Write:								
Reset:	0	0	1	1	0	0	0	0

位	功能描述
KEY1SEL	<b>KEY1 中断触发方式选择位</b> 1: 表示 KEY1 功能上升沿有效 0: 表示 KEY1 功能下降沿有效
KEY0SEL	<b>KEY0 中断触发方式选择位</b> 1: 表示 KEY0 功能上升沿有效 0: 表示 KEY0 功能下降沿有效
KEYIE1	<b>KEY1 中断使能位</b> 1: 表示 KEY1 中断有效 0: 表示 KEY1 中断无效
KEYIE0	<b>KEY0 中断使能位</b> 1: 表示 KEY0 中断有效 0: 表示 KEY0 中断无效

表 5-3-3 KEYIF 按键中断标志寄存器 (B6H)

KEY Interrupt Flag Register (KEYIF)	Address: B6H							
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	KEYIF1	KEYIF0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
KEYIF1	<b>KEY1 中断标志位</b> 1: 表示产生中断标志 (写 0 清中断) 0: 表示没有产生中断标志
KEYIF0	<b>KEY0 中断标志位</b> 1: 表示产生中断标志 (写 0 清中断) 0: 表示没有产生中断标志

注: 1, 该寄存器是不能被唤醒复位的。

2, :当 KEYx 选择作为按键输入, KEYx 端口满足电平触发条件时, 对应的 Flag 置位。

5-3-4 IEN1 中断使能寄存器 1 (B8H)

Interrupt Enable 0 Register (IEN1)	Address: B8H							
	Bit7	6	5	4	3	2	1	Bit0
Read:	EXEN2	X	X	ERTC	EEMU	EPMU	EKBI	EI2C
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
EKBI	<b>KBI中断使能位</b> 1: KBI中断使能 0: KBI中断关闭



## 5.3 LCD

### 5.3.1 概述

LCD 驱动单元最多可以支持 16 (Segment) \* 4 (Common) 输出。主要特征如下:

- 软件可编程驱动方式。(具体要根据最终的 GPIO 引脚来定)
- 软件控制对比度可调
- LCD 驱动电压可选
- 1/3 Bias, 1/4 Bias 可选
- Static、1/3、1/4 可选

### 5.3.2 LCD 与 GPIO 引脚复用

表 5-4-1 LCD 引脚定义

LCD 引脚定义	功能引脚定义	芯片引脚定义
COM0-COM3	--	COM0 - COM3
SEG4-SEG7	PA4-PA7	SEG4/PA4 - SEG7/PA7
SEG8-SEG11	PB0-PB3	SEG8/PB0 - SEG11/PB3
SEG16-SEG20	P0_0-P0_4	SEG16/P0_0 - SEG20/P0_4
SEG22	P0_6	SEG22/P0_6
SEG21、SEG23	P1_0、P1_1	SEG21/P1_0、SEG23/P1_1

### 5.3.3 框图

LCD 驱动单元采用 1/3Bias 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。

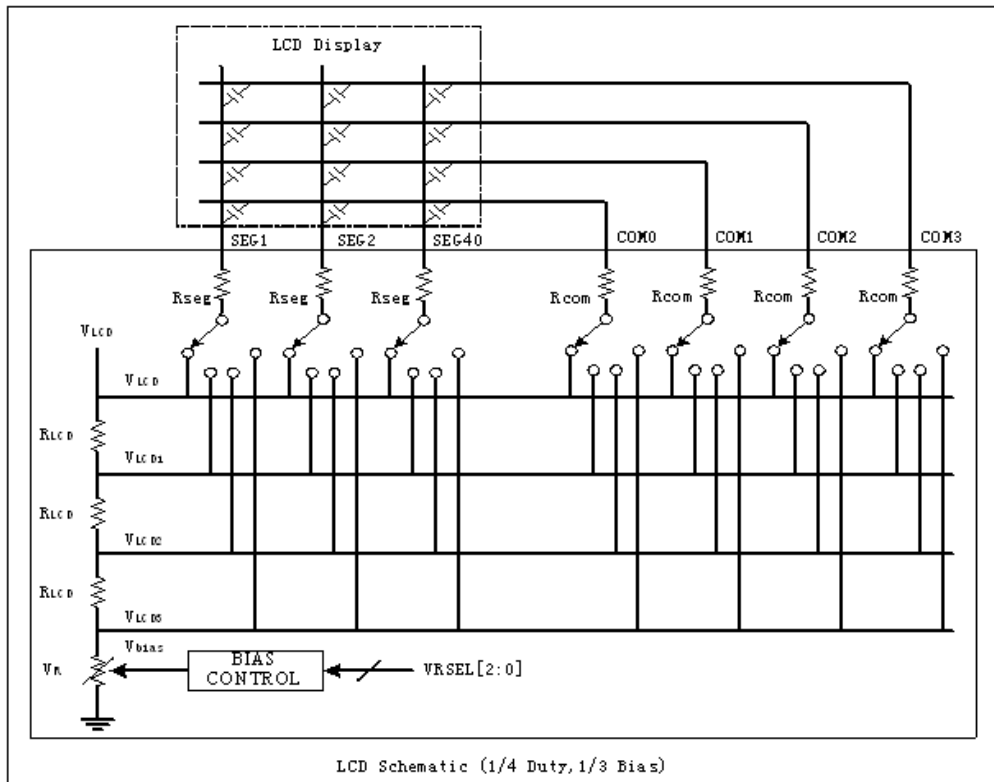


图 5-4-1 LCD 模块功能框图

LCD 的现实数据放在 LCD XRAM 中 (LBUF0—LBUF39)，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 写为 0，可以关闭 LCD 单元。LCD 关闭后，所有的 SEG 和 COM 都输出低电平，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。

### 5.3.4 输出波形

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

- ◆ DUTY[1:0]=00: Static——只用 COM0
- ◆ DUTY[1:0]=01: 1/3 duty——COM0、COM1、COM2 被使用
- ◆ DUTY[1:0]=10: 1/4 duty——COM0、COM1、COM2、COM3 被使用

#### LCD 驱动电压：

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

##### • VLCD3=Vbias

VLCD3 (Vbias) 用于控制 LCD 对比度，参考 LCD 对比度部分。

#### COM 输出波形：

##### ① Static Duty 输出波形

只使用一个 COM0，而 COM1/2/3 不使用。1 帧等于 LCD 波形时钟的周期。

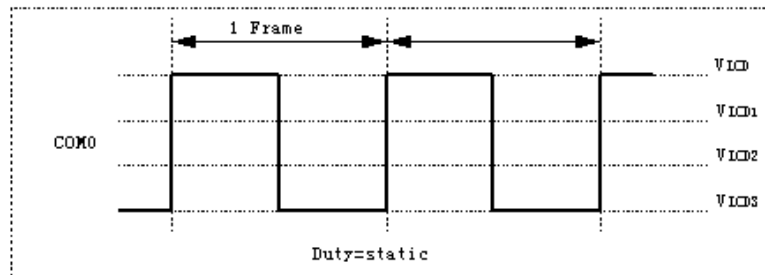


图 5-4-2

##### ② 1/3 Duty 输出波形

使用 COM0/1/2，COM3 没有被使用。1 帧等于 3 个 LCD 波形时钟周期。

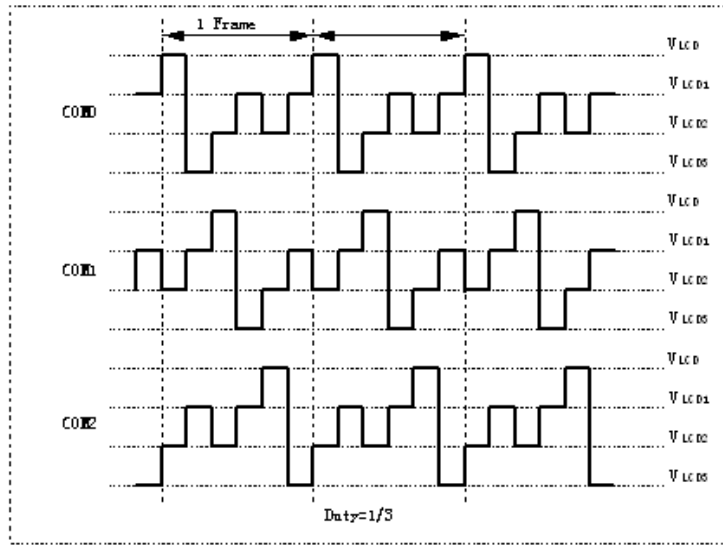


图 5-4-3

③ 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。

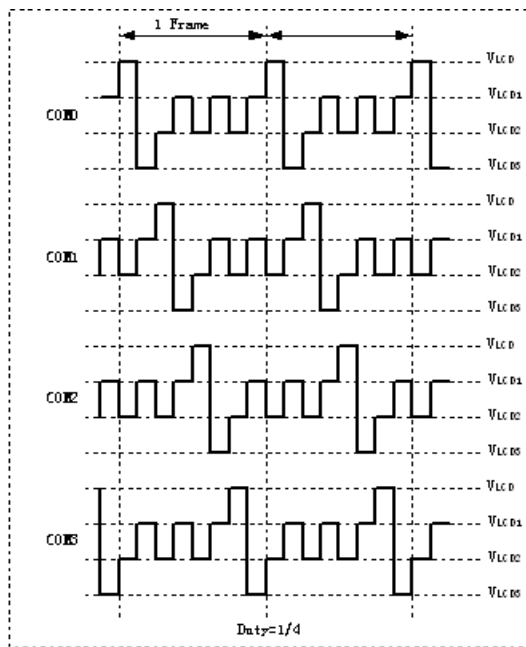


图 5-4-4

SEGMENT 输出波形:

- ① Static Duty

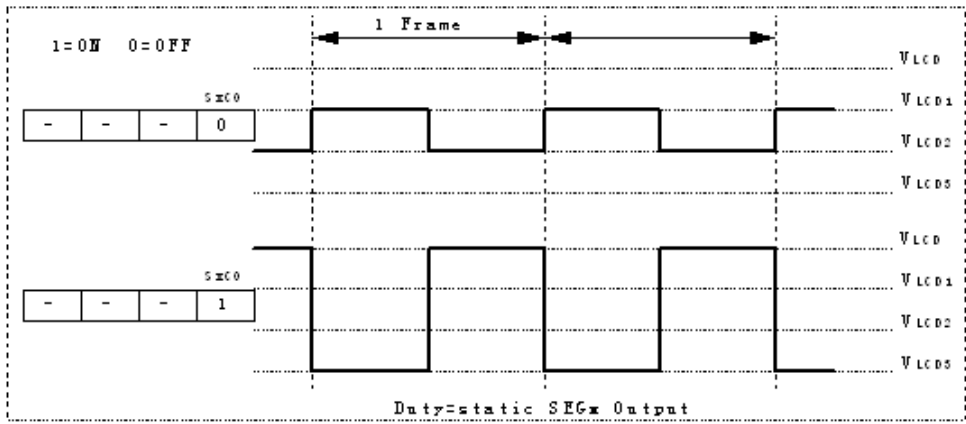


图 5-4-5

② 1/3 Duty

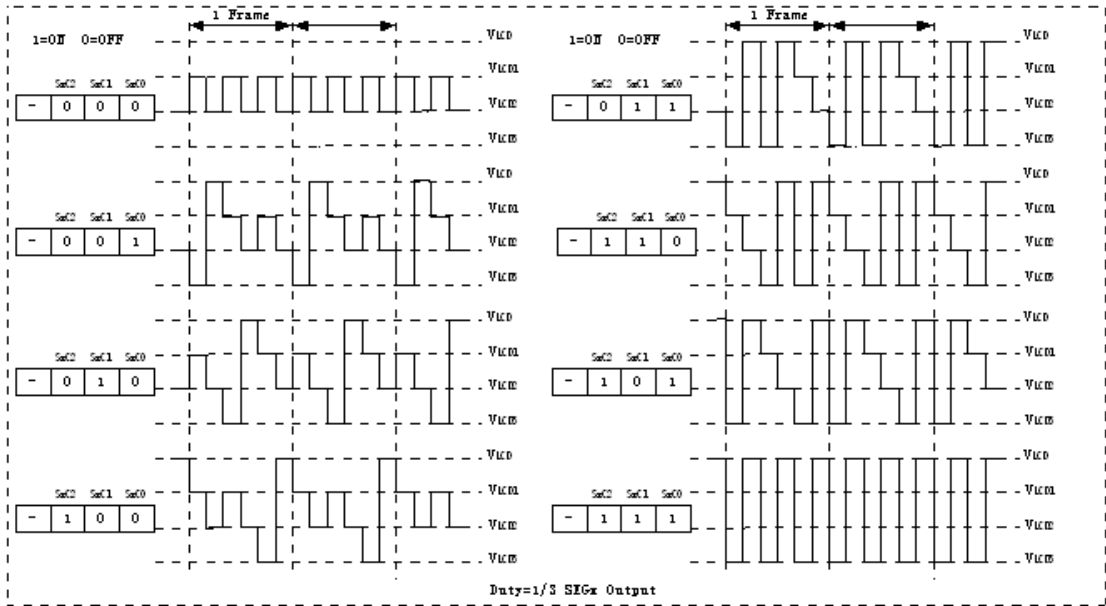


图 5-4-6

③ 1/4 Duty

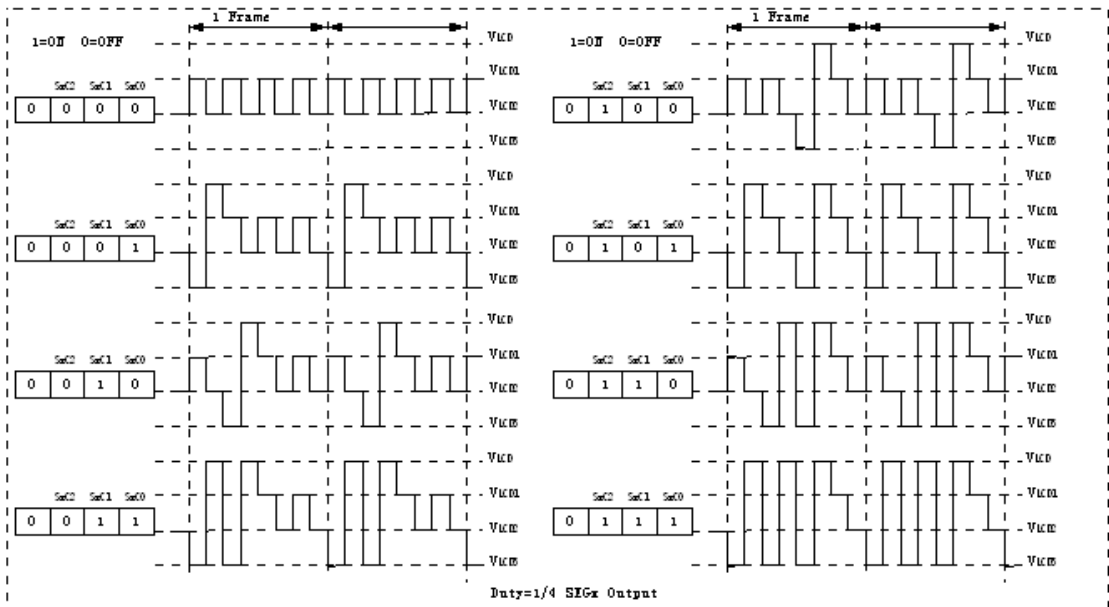


图 5-4-7

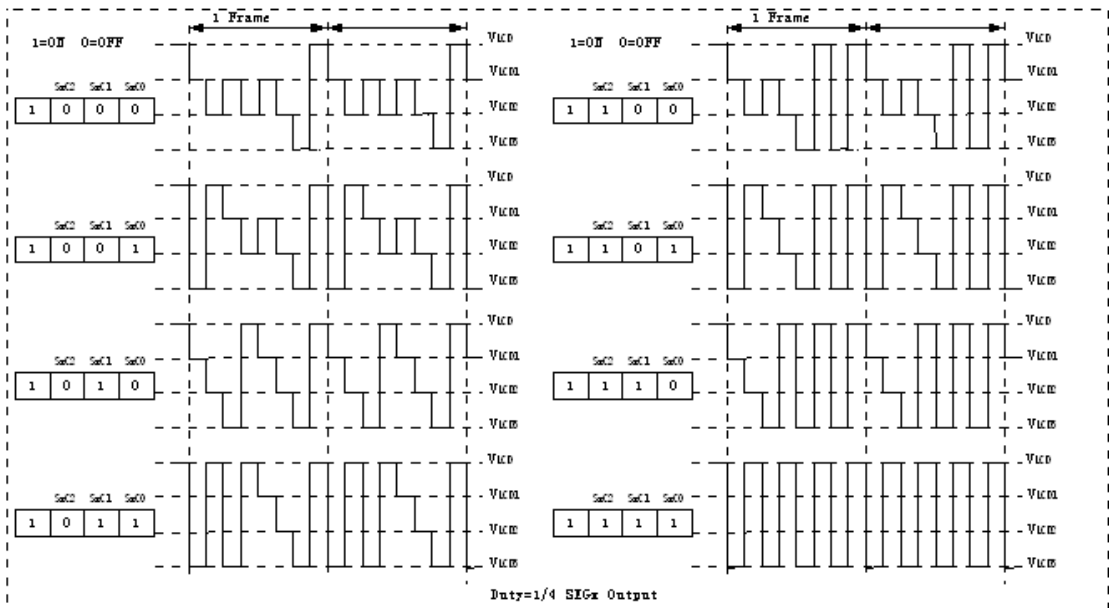


图 5-4-8

### 5.3.5 特殊功能寄存器

表 5-4-2 LCD SFR 寄存器列表

地址	名称	复位值	功能描述
0xC4	LCDCLK	0x90	LCD 时钟频率选择寄存器
0xC5	LCDCR	0x00	LCD 驱动控制寄存器
0xC6	LCDCFG	0x00	LCD 输出复用配置寄存器 (详见 5.8.2 GPIO)
0xCE	P02CFG	0x00	P0 和 P2 输出复用配置寄存器 (详见 5.8.2 GPIO)

表 5-4-2 LCDCLK LCD 时钟频率选择寄存器 (C4H)

LCD Clock Register (LCDCLK)			Address: C4H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	BIASCTL	0	0	DUTY1	DUTY0	LCLK2	LCLK1	LCLK0
Write:		x	x					
Reset:	1	0	0	1	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[1:0]	<b>LCD的duty控制选择位</b>					
	DUTY1	DUTY0	COMMON选择		LCD波形Duty	
	0	0	Com0		Static duty	
	0	1	Com0~Com2, 3Com		1/3 duty	
	1	0	Com0~Com3, 4Com		1/4 duty	
LCLK[2:0]	<b>LCD扫描频率选择位</b>					
	LCLK2	LCLK1	LCLK0	LCD波形扫描频率 flcd/Static 帧扫描频率	1/3Duty 帧扫描 频率	1/4Duty 帧扫描 频率
	1	0	0	512 Hz	170.7 Hz	128 Hz
	0	0	0	256 Hz	85.3 Hz	64 Hz
	0	0	1	128 Hz	42.7 Hz	32 Hz
	0	1	0	64 Hz	21.3 Hz	16 Hz
	0	1	1	32 Hz	10.7 Hz	8 Hz

注:

1, LCD 单元的时钟来自晶体振荡频率  $f_{osc}$ ,  $f_{osc}$  经过分频之后作为 LCD 波形扫描频率  $f_{lcd}$ , 波形扫描频率  $f_{lcd}$  可以通过寄存器 LCLK 进行配置, 当选择 Static Duty (即 com0) 时, Static Duty 帧扫描频率等于  $f_{lcd}$  的频率, 其它 Duty 情况下的帧扫描频率等于  $f_{lcd} * duty$ 。

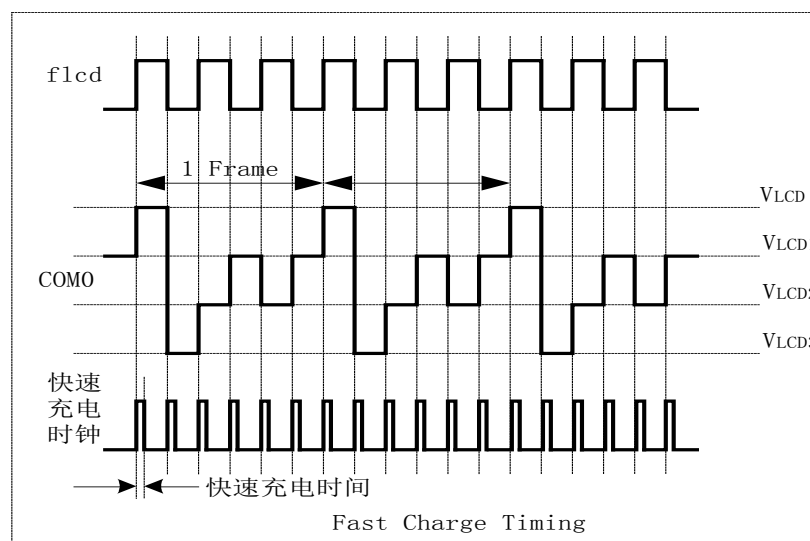
表 5-4-3 LCDCR LCD 驱动控制寄存器(C5H)

LCD Control Register (LCDCR)			Address: C5H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	FCSET1	FCSET0	FC	LC	0	VRSEL 2	VRSEL 1	VRSEL 0
Write:					x			
Reset:	0	0	0	0	0	0	0	0

位	功能描述		
FCSET[1:0]	<b>快速充电时间选择位</b>		
	FCSET1	FCSET0	快速充电时间

	0	0	1/8个f1cd周期	
	0	1	1/16个f1cd周期	
	1	0	1/32个f1cd周期	
	1	1	1/64个f1cd周期	
详见下图 5-4-9 快速充电时序图				
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			
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	
	X	0	RLCD=37K Ω，大电流充电模式	
	0	1	RLCD=146K Ω，小电流充电模式	
	1	1	短时间大电流，快速充电模式	
VRSEL[2:0]	<b>LCD显示对比度设置位</b>			
	VRSEL2	VRSEL1	VRSEL0	1/3 bias 对比度 (% of VLCD)
				1/4 bias 对比度 (% of VLCD)
	0	0	0	0.1
	0	0	1	2.2
	0	1	0	4.3
	0	1	1	6.3
	1	0	0	8.3
	1	0	1	10.1
	1	1	0	11.9
	1	1	13.6	

图 5-4-9 快速充电时序图



---

**LCD 工作电流:**

缺省状态下, LCD 分压网络的电阻 RLCD 为 37K  $\Omega$  (注: VLCD=3v 时)。根据 LCD 差异, 可以选择 RLCD=146K  $\Omega$ , 此时流过电阻网络的电流较小。

在选择 RLCD=146K  $\Omega$  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=37K  $\Omega$  进行快速充电, 然后再切换到 RLCD=146K  $\Omega$  模式。快速充电时间可以通过 FCSET[1:0] 进行选择。

**LCD 对比度:**

LCD 显示对比度可通过配置寄存器 VRSEL[2:0] 进行调整。VRSEL[2:0] 通过调整 Vbias 电压, 提供 8 级对比度调整范围。调整 Vbias 时, 即 VLCD3 被调整, 而 VLCD1 和 VLCD2 也被相应调整。

设置 VRSEL[2:0]= 000 时, 表示选择最大对比度, 而设置 VRSEL[2:0]= 111 时, 表示选择最小的对比度。

**LCD 的显示 RAM 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。

3000H – 3027H 共 40 Bytes XRAM 做为 LCD BUFF, 操作方式与一般 XRAM 相同, 同时与 LCD 的对应关系如下:

地址 3004H 对应 Seg4 (COM3—COM0)

地址 3005H 对应 Seg5 (COM3—COM0)

地址 3010H 对应 Seg16 (COM3—COM0)

如把 SEG4 , COM3 点亮的操作:

```
MOVX 3004H , 0X08;
```

如把 SEG16 , COM0, COM1 都点亮的操作:

```
MOVX 3010H , 0X03;
```



## 5.4 RTC

### 5.4.1 概述

RTC 单元可以提供实时时钟、日历等功能，具有自动闰年调整功能，支持闹钟功能和周期性中断功能。

RTC 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 所有的输出寄存器（时间&万年历）、RTC 时钟校正寄存器不会被复位，用以保持 RTC 的准确性。

### 5.4.2 功能描述

- 提供时钟和日历功能：输出寄存器包含秒、分、时、日、月、年和星期
- 具有自动闰年闰月调整功能
- 1 个闹钟中断功能
- 2 个定时器中断功能
- 5 个时间中断功能（秒、分、时、日、月）
- 可输出频率为 1/2/4/8/32/128/1024/32768Hz 的方波
- 可输出每秒补偿、20 秒补偿的校验脉冲

#### 5.4.2.1 时钟校正

RTC 功能需要外部的 32768Hz 晶振电路。由于晶体存在初始误差，在温度等外部条件变化时也会有相应的偏移，因而需要对其进行校准。

使用寄存器 **RTCCAL (F9H)** 可以对晶体产生的时钟进行校正。具体的校正需要通过 TOUT 管脚进行，配置 **TOUT[2:0] (F8H)** 为 111 使其输出未校正的晶振频率，用高精度的频率计测试其频率  $f_x$ ，计算需要补偿的数  $N$  写入 RTCCAL(F9H)。校正后的效果也可以通过 TOUT 管脚输出来判断，即把 TOUT[2:0] 配置为校正后的时钟输出。

RTCCAL 寄存器数值实际表示为 2 进制有符号数格式，最高位为符号位，范围是 -128 —— +127，01111111 表示+127，10000000 表示-128。每一个 LSB 对应的计时误差为 1.525ppm，即 1.525ppm/LSB。

当芯片工作在高频时钟域时(PLL 打开)，校正脉冲输出 TOUT 可选 1s 补偿一次（高频）或 20s 补偿一次（低频）；客户只需要写入低频补偿寄存器，高频 s 脉冲补偿参数会自动得到。

当芯片工作在低频时钟域时(PLL 关闭)，RTC 只能实现 20 秒补偿一次（低频补偿是以 32768Hz 晶体时钟为基准）。

时钟切换时不会影响内部 RTC 的精度。

补偿数  $N$  的计算公式为：

$$N = (f_x - 32768) \times 20$$

其中， $f_x$  为实际测试的 osc 频率值，在 32768Hz 附近；

若  $N > 0$ ,  $RTCCAL = N$

若  $N < 0$ ,  $RTCCAL = 256 - N$  （取补码）

例如：

若实际测得未校正晶振频率为 32767Hz，由于标准晶振为 32768Hz，则不论是高频 1 秒补偿一次还是低频 20 秒补偿一次，其补偿计算公式是完全一样的，即：

$$\begin{aligned} N &= (32767 - 32768) \times 20 \\ &= -20 \end{aligned}$$

计算得 N 为 -20，需要减少 20 个低频时钟周期，RTCCAL 寄存器中应写入 236，即 ECH(11101100)，无论是高频 1 秒补偿一次还是低频 20 秒补偿一次，RTCCAL 写入的值都是一样的。

#### 调校方法:

客户根据 TOUT 管脚的输出（此时选为 32768Hz 输出）确定 32768Hz 时钟的误差，得到写入此寄存器的校正值，保存在非易失性存储器中，系统上电后程序将该校正值写入此寄存器，然后硬件会根据 RTCCAL 中的值自动做出相应增减时钟的动作。

RTC 配置的实例以及对应的减少/增加的时钟个数:

表 5-6-1 RTCCAL 操作举例

RTCCAL	RTCCAL 二进制	增加/减少	个数
0x00	0000_0000	增加	0000_0000=0x00=00
0x01	0000_0001	增加	0000_0001=0x01=01
0x50	0101_0000	增加	0101_0000=0x50=80
0x51	0101_0001	增加	0101_0001=0x51=81
0x7f	0111_1111	增加	0111_1111=0x7F=127
0x80	1000_0000	减少	~(1000_0000)+1=1000_0000=128
0x81	1000_0001	减少	~(1000_0001)+1=0111_1111=0x7f=127
0x82	1000_0010	减少	~(1000_0010)+1=0111_1110=0x7e=126
0xaf	1010_1111	减少	~(1010_1111)+1=0101_0001=0x51=81
0xb0	1011_0000	减少	~(1011_0000)+1=0101_0000=0x50=80
0xfe	1111_1110	减少	~(1111_1110)+1=0000_0010=0x02=2
0xff	1111_1111	减少	~(1111_1111)+1=0000_0001=0x01=1

高频时可通过设置寄存器 RTCCON 的 TOUTEN[1:0]=11 使输出校验脉冲为每秒补偿一次，每秒补偿的脉冲精度与 20 秒补偿的脉冲精度相同。

#### 5.4.2.2 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。

#### 5.4.2.3 寄存器写保护功能与复位源

表 5-6-2 寄存器的复位说明

名称	复位源	写保护	功能描述
RTCCON	可以被复位	否	RTC 控制寄存器
RTCCAL	只能被 POR, LBOR 复位	是	RTC 时钟校正寄存器
RTCIE	可以被复位	否	RTC 中断使能配置
RTCIF	只能被 POR, LBOR, 外部	否	RTC 中断标志

	Reset, BOR, Debug RST 复位		
ALMR	可以被复位	否	闹钟的分钟设置
ALHR	可以被复位	否	闹钟的小时设置
RTC2CNT	可以被复位	否	RTC 定时器 2 计数设置
RTC1CNT	可以被复位	否	RTC 定时器 1 计数设置
SECR	只能被 POR, LBOR 复位	是	秒寄存器
MINR	只能被 POR, LBOR 复位	是	分钟寄存器
HRR	只能被 POR, LBOR 复位	是	小时寄存器
DAYR	只能被 POR, LBOR 复位	是	天寄存器
MTHR	只能被 POR, LBOR 复位	是	月寄存器
YRR	只能被 POR, LBOR 复位	是	年寄存器
DOWR	只能被 POR, LBOR 复位	是	星期寄存器

对于时间和万年历进行校正时，所有的时间计数器并不停止，如果修改相应的寄存器，则在此基础上继续累加。由于校正时并未停止时间计数，因而要考虑有可能产生的进位。

RTC 的输出寄存器（SECR、MINR、HRR、DAYR、MTHR、YRR、DOWR）只能被 POR, LBOR 复位，而且均有写保护。

打开写保护后，执行 32 个 cpu 指令周期的时间的限制，因而要确保该配置过程不能被中断等打断。推荐用户在使用时，**先将 EA 关闭，配置好后再将 EA 打开**。如果不能保证在规定的指令周期内操作完毕，可以将配置分为两个或者更多的步骤进行，只要在每个块之前重新做写保护打开的配置。（详细见 3.1.5.2 寄存器写保护）

比如，对秒输出寄存器（0F9H）操作如下。

```

CLR    EA                ;
MOV    BWPR, #0CFH      ; slect write enable mode
MOV    BWPR, #0DCH      ; enable RTC-proctec-reg write
MOV    0FCH, #010H      ; config RTC SECR
...
MOV    BWPR, #00H       ;close RTC reg writing;
                                ; may neglect, it will be closed in 32 cpu clk
SETB   EA
    
```

其中 BWPR 为 09AH。

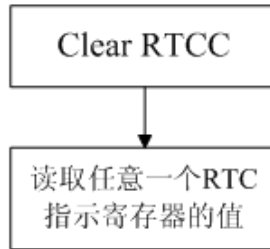
#### 5.4.2.4 中断功能

RTC 一共提供 8 种中断源，公用 MCU 的 IRQ-RTC 中断向量 12。RTC 的 8 种中断源由 RTCIE(AAH) 控制其使能；如果配置两个 RTC 定时器中断，需要同时打开相应 RTC 定时器使能（RTC CON），并配置定时中断时间（RTC1CNT、RTC2CNT）；如果配置闹钟中断（ALMIE=1），需要同时配置闹钟时间配置寄存器（ALMR、ALHR）；所有的 RTC 中断标志可以查询 RTCIF，中断标志写 0 清 0。

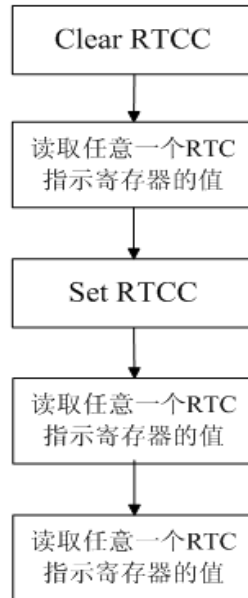
#### 5.4.2.5 RTC 指示寄存器读取流程

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

(1) 读取单个指示寄存器的值:



(2) 连续读取多个 RTC 指示寄存器的值:



### 5.4.3 特殊功能寄存器

表 5-6-3 寄存器列表

地址	名称	复位值	功能描述
F8H	RTCCON	0x00	RTC 控制寄存器
F9H	RTCCAL	0x00	RTC 时钟校正寄存器, 写保护
AEH	RTCIE	0x00	RTC 中断使能配置
B4H	RTCIF	0x00	RTC 中断标志
F1H	ALMR	0x00	闹钟的分钟设置
F2H	ALHR	0x00	闹钟的小时设置
FAH	RTC1CNT	0x00	RTC 定时器 1 计数设置
FBH	RTC2CNT	0x00	RTC 定时器 2 计数设置
FCH	SECR	0x00	秒寄存器, 写保护
FDH	MINR	0x00	分钟寄存器, 写保护
FEH	HRR	0x00	小时寄存器, 写保护
FFH	DAYR	0x00	天寄存器, 写保护
F3H	MTHR	0x00	月寄存器, 写保护
F4H	YRR	0x00	年寄存器, 写保护
F5H	DOWR	0x00	星期寄存器, 写保护

表 5-6-4 RTC 控制寄存器 (F8H)

RTC Control Register (RTCCON)			Address: F8H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	RTCC	RTC2EN	RTC1EN	TOUTEN1	TOUTEN0	TOUT2	TOUT1	TOUT0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述																																				
RTCC	<b>RTC 寄存器读取控制位</b> 1: 连续读取 RTC 指示寄存器读数 0: 开始读取 RTC 指示寄存器读数																																				
RTC2EN	<b>RTC 定时器 2 使能位</b> 1: RTC 定时器 2 被使能, 溢出产生 RTC2IF 标志 0: RTC 定时器 2 被关闭																																				
RTC1EN	<b>RTC 定时器 1 使能位</b> 1: RTC 定时器 1 被使能, 溢出产生 RTC1IF 标志 0: RTC 定时器 1 被关闭																																				
TOUTEN1[1:0]	<b>TOUT 输出信号控制</b> <table border="1" data-bbox="528 976 1422 1317"> <thead> <tr> <th>TOUTEN1</th> <th>TOUTEN0</th> <th>TOUT 输出信号</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>输出固定低电平 0</td> </tr> <tr> <td>0</td> <td>1</td> <td>输出固定高电平 1</td> </tr> <tr> <td>1</td> <td>0</td> <td>按照 TOUT[2:0]配置的频率输出方波信号, 输出的校验脉冲 20 秒补偿一次。</td> </tr> <tr> <td>1</td> <td>1</td> <td>按照 TOUT[2:0]配置的频率输出方波信号, 输出的校验脉冲在高频时每秒补偿一次, 低频时 20 秒补偿一次。</td> </tr> </tbody> </table>	TOUTEN1	TOUTEN0	TOUT 输出信号	0	0	输出固定低电平 0	0	1	输出固定高电平 1	1	0	按照 TOUT[2:0]配置的频率输出方波信号, 输出的校验脉冲 20 秒补偿一次。	1	1	按照 TOUT[2:0]配置的频率输出方波信号, 输出的校验脉冲在高频时每秒补偿一次, 低频时 20 秒补偿一次。																					
	TOUTEN1	TOUTEN0	TOUT 输出信号																																		
	0	0	输出固定低电平 0																																		
	0	1	输出固定高电平 1																																		
1	0	按照 TOUT[2:0]配置的频率输出方波信号, 输出的校验脉冲 20 秒补偿一次。																																			
1	1	按照 TOUT[2:0]配置的频率输出方波信号, 输出的校验脉冲在高频时每秒补偿一次, 低频时 20 秒补偿一次。																																			
TOUT[2:0]	<b>TOUT 输出方波频率配置</b> <table border="1" data-bbox="528 1357 1422 1744"> <thead> <tr> <th>TOUT2</th> <th>TOUT1</th> <th>TOUT0</th> <th>TOUT 输出方波信号频率</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1Hz</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>2Hz</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>4Hz</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>8Hz</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>32Hz</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>128Hz</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1024Hz</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>32.768KHz</td> </tr> </tbody> </table> <p>注: 1, 输出信号为占空比 50% 的方波;                      2, 当配置输出方波频率为 1Hz/2Hz/4Hz/8Hz/32Hz/128Hz 时, 输出经过 RTCCAL 补偿的信号;                      3, 当配置输出方波频率为 1024Hz /32.768KHz 时, 输出未经过补偿的信号;</p>	TOUT2	TOUT1	TOUT0	TOUT 输出方波信号频率	0	0	0	1Hz	0	0	1	2Hz	0	1	0	4Hz	0	1	1	8Hz	1	0	0	32Hz	1	0	1	128Hz	1	1	0	1024Hz	1	1	1	32.768KHz
	TOUT2	TOUT1	TOUT0	TOUT 输出方波信号频率																																	
	0	0	0	1Hz																																	
	0	0	1	2Hz																																	
	0	1	0	4Hz																																	
	0	1	1	8Hz																																	
	1	0	0	32Hz																																	
	1	0	1	128Hz																																	
1	1	0	1024Hz																																		
1	1	1	32.768KHz																																		

表 5-6-5 RTCCAL RTC 校正寄存器 (写保护) (F9H)

RTC Calibration Register (RTCCAL)			Address: F9H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
Write:								
Reset:	0	0	0	0	0	0	0	0

注：1，此寄存器带 REG1 型写保护，需连续对 BWPM 写入 CFH，DCH 才能改写此寄存器。

2，RTCCAL 数据为 8 位二进制补码格式，内部硬件会自动根据 RTCCAL 中的校正值自动对 OSC 晶体产生的时钟进行补偿校正，修正过的频率低于 1024Hz 的时钟信号可以通过 TOUT/P2.4 输出。

表 5-6-6 RTCIE RTC 中断使能寄存器 (AEH)

RTC Interrupt Enable Register (RTCIE)			Address: AEH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	ALMIE	RTC2IE	RTC1IE	MTHIE	DAYIE	HRIE	MINIE	SECIE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ALMIE	<b>闹钟中断使能位</b> 1: 使能中断; 0: 关闭中断
RTC2IE	<b>RTC 定时器 2 中断使能位</b> 1: 使能中断; 0: 关闭中断
RTC1IE	<b>RTC 定时器 1 中断使能位</b> 1: 使能中断; 0: 关闭中断
MTHIE	<b>RTC 月中断使能位</b> 1: 使能中断; 0: 关闭中断
DAYIE	<b>RTC 日中断使能位</b> 1: 使能中断; 0: 关闭中断
HRIE	<b>RTC 小时中断使能位</b> 1: 使能中断; 0: 关闭中断
MINIE	<b>RTC 分钟中断使能位</b> 1: 使能中断; 0: 关闭中断
SECIE	<b>RTC 秒中断使能位</b> 1: 使能中断; 0: 关闭中断

表 5-6-7 RTCIF RTC 中断标志寄存器 (B4H)

RTC Interrupt Flag register (RTCIF)			Address: B4H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	ALMIF	RTC2IF	RTC1IF	MTHIF	DAYIF	HRIF	MINIF	SECIF
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ALMIF	<b>闹钟中断标志位</b> 1: 当小时和分钟与设定的闹钟匹配时, 产生闹钟中断, 此位置 1, 对该位写 0 清标志; 0: 没有发生中断
RTC2IF	<b>RTC 定时器 2 中断标志位</b> 1: 如设置 RTC1CNT=X, 使能计数 RTC1EN 后, 经过(X+1)*1S 后, 该标志位置位 1., 对该位写 0 清标志; 0: 没有发生中断
RTC1IF	<b>RTC 定时器 1 中断标志位</b> 1: 如设置 RTC2CNT=X, 使能计数 RTC2EN 后, 经过(X+1)*1S 后, 该标志位置位 1., 对该位写 0 清标志; 0: 没有发生中断
MTHIF	<b>RTC 月中断标志位</b> 1: 月计数器 MTHR 加 1 时, 产生一个月中断, MTHF 被置为 1, 对该位写 0 清标志; 0: 没有发生中断
DAYIF	<b>RTC 日中断标志位</b> 1: 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; 0: 没有发生中断
HRIF	<b>RTC 小时中断标志位</b> 1: 小时计数器 HRR 加 1 时, 产生一个小时中断, HRF 被置为 1, 对该位写 0 清标志; 0: 没有发生中断
MINIF	<b>RTC 分钟中断标志位</b> 1: 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; 0: 没有发生中断
SECIF	<b>RTC 秒中断标志位</b> 1: 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; 0: 没有发生中断

表 5-6-8 ALMR 闹钟分钟设置寄存器 (F1H)

Alarm Minute Register (ALMR)			Address: F1H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	AM5	AM4	AM3	AM2	AM1	AM0
Write:	x	X						
Reset:	0	0	0	0	0	0	0	0

ALMR 为 6 位二进制无符号整数, 允许写入 0-59 以外的数, 但是闹钟中断将永远不会产生。

表 5-6-9 ALHR 闹钟小时设置寄存器 (F2H)

Alarm Hour Register (ALHR)			Address: F2H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	AH4	AH3	AH2	AH1	AH0
Write:	x	X	x					
Reset:	0	0	0	0	0	0	0	0

ALHR 为 5 位二进制无符号整数，允许写入 0-23 以外的数，但是闹钟中断将永远不会产生。

表 5-6-10 RTC1CNT RTC 定时器 1 寄存器 (FAH)

RTC1 Counter (RTC1CNT)			Address: FAH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	RTC1N7	RTC1N6	RTC1N5	RTC1N4	RTC1N3	RTC1N2	RTC1N1	RTC1N0
Write:								
Reset:	0	0	0	0	0	0	0	0

RTC1CNT 为 8 位二进制无符号整数，最小分辨率为 1s，即最小可以每 1 秒产生一次中断，最大可以每 256 秒产生一次中断，当计数溢出时，置位 RTC2IF 标志。

定时中断时间计算公式为：

$$T1 = (RTC1CNT + 1) \times 1S$$

单位：秒

例如：

如果设置  $RTC1N[7..0] = 00H$ ，表示 RTC 内部的秒表功能中断每经过  $(00H+1) \times 1S = 1 \times 1S = 1S$  的计时周期后，置位 RTC1IF 标志。

表 5-6-11 RTC2CNT RTC 定时器 2 寄存器 (FBH)

RTC2 Counter (RTC2CNT)			Address: FBH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	RTC2N7	RTC2N6	RTC2N5	RTC2N4	RTC2N3	RTC2N2	RTC2N1	RTC2N0
Write:								
Reset:	0	0	0	0	0	0	0	0

RTC2CNT 为 8 位二进制无符号整数，最小分辨率为 0.0625 秒，即最小可以每 0.0625 秒产生一次中断，最大可以每 16 秒产生一次中断。

定时中断时间计算公式为：

$$T2 = (RTC2CNT + 1) \times 0.0625S$$

单位：秒

例如：

如果设置  $RTC2N[7..0] = 13H$ ，表示 RTC 内部的秒表功能中断每经过  $(13H+1) \times 0.0625S = 20 \times 0.0625S = 1.25S$  的计时周期后，置位 RTC2IF 标志。



表 5-6-12 SECR RTC 秒寄存器 (写保护) (FCH)

Second Register (SECR)			Address: FCH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
Write:	x	X						
Reset:	0	0	0	0	0	0	0	0

SECR 为 6 位二进制无符号整数, 范围: 0-59。写入 0-59 以外的任何数值, 会写无效。

注: 写此寄存器需打开写保护: BWPR=0xCF; BWPR=0xDC;

表 5-6-13 MINR RTC 分钟寄存器 (写保护) (FDH)

Minute Register (MINR)			Address: FDH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0
Write:	x	X						
Reset:	0	0	0	0	0	0	0	0

MINR 为 6 位二进制无符号整数, 范围: 0-59。写入 0-59 以外的任何数值, 会写无效。

注: 写此寄存器需打开写保护: BWPR=0xCF; BWPR=0xDC;

表 5-6-14 HRR RTC 小时寄存器 (写保护) (FH)

Hour Register (HRR)			Address: FEH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	HR4	HR3	HR2	HR1	HR0
Write:	X	X	X					
Reset:	0	0	0	0	0	0	0	0

HRR 为 5 位二进制无符号整数, 范围: 0-23。写入 0-23 以外的任何数值, 会写无效。

注: 写此寄存器需打开写保护: BWPR=0xCF; BWPR=0xDC;

表 5-6-15 DAYR RTC 日寄存器 (写保护) (FFH)

Day Register (DAYR)			Address: FFH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	DAY4	DAY3	DAY2	DAY1	DAY0
Write:	x	X	x					
Reset:	0	0	0	0	0	0	0	0

DAYR 为 5 位二进制无符号整数, 范围: 1-28/29/30/31。写入与年、月不匹配的任何数值, 会写无效。

注: 写此寄存器需打开写保护: BWPR=0xCF; BWPR=0xDC;

表 5-6-16 MTHR RTC 月寄存器 (写保护) (F3H)

Month Register (MTHR)			Address: F3H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	0	MTH3	MTH2	MTH1	MTH0
Write:	x	X	x	x				
Reset:	0	0	0	0	0	0	0	0

MTHR 为 4 位二进制无符号整数，范围：1-12。写入 1-12 以外的任何数值，会写无效。

注：写此寄存器需打开写保护：BWPR=0xCF；BWPR=0xDC；

表 5-6-17 YRR RTC 年寄存器（写保护）(F4H)

Year Register (YRR)		Address: F4H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	YR6	YR5	YR4	YR3	YR2	YR1	YR0
Write:	X							
Reset:	0	0	0	0	0	0	0	0

YRR 为 7 位二进制无符号整数，有效范围：0-99，最大可写入 127。写入 0-127 以外的任何数值，会写无效。

注：写此寄存器需打开写保护：BWPR=0xCF；BWPR=0xDC；

表 5-6-18 DOWR RTC 星期寄存器（写保护）(F5H)

Day-Of-Week Register (DOWR)		Address: F5H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	0	0	DOW2	DOW1	DOW0
Write:	x	X	x	x	x			
Reset:	0	0	0	0	0	0	0	0

DOWR 为 3 位二进制无符号整数，范围：1-7。写入 1-7 以外的任何数值，会写无效。

注：写此寄存器需打开写保护：BWPR=0xCF；BWPR=0xDC；

## 5.5 JTAG

### 5.5.1 概述

ATT7039 支持标准的 JTAG 接口，用户可以通过引脚 TCK、TDI、TDO、TMS 实现在线编程和在线调试。

### 5.5.2 介绍

拉低外部引脚 JTAG\_WDTEN 可以控制芯片进入调试模式。

在调试模式，系统时钟始终为高频 1.37MHz，不能进入低频状态。

在调试模式，P2.0、P2.1、P1.0、P1.1 为 JTAG 功能，配置的 GPIO 功能或 UART1/KEY0/KEY1 功能无效，恢复正常模式后，系统自动转换为这些端口配置的功能。

## 5.6 GPIO

### 5.6.1 概述

ATT7039 提供 P0.6、P0[4..0]、P1[1..0]、P2.6、P[3..0]、P3[2..0]、PA[7..4]、PB[3..0] 共 24 个双向 GPIO 引脚，可以分别配置成输入或者输出模式。作为输入方式时，带有施密特触发器和 glitch filter 处理，内部上拉 30K 电阻。P1.0 和 P1.1 为 floating。

端口输入/输出模式配置寄存器相关位由 0 写为 1 时，即将端口由输入模式改为输出模式时，为避免在端口上产生毛刺，建议先写端口数据寄存器，然后再写端口方向寄存器。

### 5.6.2 特殊功能寄存器

表 5-6-1 SFR 寄存器列表

地址	名称	复位值	功能描述
0xC6	LCDCFG	00H	LCD 配置寄存器
0xC7	KEYCFG	00H	按键配置寄存器
0xCE	P02CFG	30H	P0 口 P2 口配置寄存器
0xCF	P3CFG	00H	P3 口配置寄存器
0x80	P0	00H	P0 口输出寄存器
0xD5	DDRP0	00H	P0 口输出方向寄存器
0xC1	LEDP0	00H	P0 口输出电流模式配置寄存器
0x90	P1	00H	P1 口输出寄存器
0xD6	DDRP1	00H	P1 口输出方向寄存器
0xA0	P2	00H	P2 口输出寄存器
0xD7	DDRP2	00H	P2 口输出方向寄存器
0xB0	P3	00H	P3 口输出寄存器
0xD9	DDRP3	00H	P3 口输出方向寄存器
0xBC	PTA	00H	PA 口输出寄存器
0xD1	DDRPA	00H	PA 口输出方向寄存器
0xBD	PTB	00H	PB 口输出寄存器
0xD2	DDRPB	00H	PB 口输出方向寄存器

#### 5.6.2.1 GPIO 输出复用配置寄存器

表 5-6-2 LCDCFG LCD 输出复用配置寄存器 (C6H)

LCDCFG		Address: C6H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	PTBL	PTAH	X
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PTBL	<b>PB 口低 4 位功能配置位</b> 1: SEG8/PB0—SEG11/PB3 配置为 LCD 输出 SEG8—SEG11

	0: SEG8/PB0—SEG11/PB3 配置为标准 I/O, PB0-PB3
PTAH	<b>PA 口高 4 位功能配置位</b> 1: SEG4/PA4—SEG7/PA7 配置为 LCD 输出, SEG4-SEG7 0: SEG4/PA4—SEG7/PA7 配置为标准 I/O, PA4-PA7

注: 1 此寄存器带 REG1 型写保护, 需连续对 BWPM 写入 CFH, DCH 才能改写此寄存器。

表 5-6-3 KEYCFG P1 输出复用配置寄存器 (C7H)

KEYCFG		Address: C7H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	SEG23	SEG21	KEY1	KEY0
Write:								
Reset:	0	0	0	0	0	0	0	0

KEY0	KEY1	SEG21	SEG23	功能说明
0	0	0	0	PIN35, PIN36 分别为 P1.1 和 P1.0
0	0	0	1	PIN35 为 SEG23 功能
0	0	1	0	PIN36 为 SEG21 功能
0	1	0	0	PIN35 为 KEY1 功能
1	0	0	0	PIN36 为 KEY0 功能
1	1	1	1	PIN35, PIN36 分别为 P1.1 和 P1.0
1	0	1	0	PIN35, PIN36 分别为 P1.1 和 P1.0
0	1	0	1	PIN35, PIN36 分别为 P1.1 和 P1.0

注意:

- 1, 如果 SUPDC 中的 KBI\_EN=0, 关闭 KBI 功能模块。如果 SUPDC 中的 KBI\_EN=1, 开启 KBI 功能模块, 同时在 KEYFG 中又配置为 KEY<sub>x</sub>=1, 则配置为 KEY 功能 PIN (详见 KBI 章节)。
- 2, 在 JTAG 模式 (WDTEN PIN 被拉到低电平), TCK/KEY0/P1.0/SEG21—TDI/KEY1/P1.1/SEG23, 两个 PIN 被配置使能为 TCK, TDI。
- 3, 此寄存器带 REG1 型写保护, 需连续对 BWPM 写入 CFH, DCH 才能改写此寄存器。

表 5-6-4 P02CFG P0 和 P2 输出复用配置寄存器 (CEH)

P02CFG		Address: CEH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	P267	QF/TOU T	P223	P201	P067	P045	P023	P001
Write:								
Reset:	0	0	1	1	0	0	0	0

位	功能描述
P267	<b>P2.6 端口功能配置位</b> 1: PF/P2.6 配置为标准 I/O, P2.6 0: PF/P2.6 配置为脉冲输出, PF
QF/TOU	<b>P25 端口功能配置位</b>

	QF/TOUT=0: 引脚 25 被配置为 QF 脉冲输出 QF/TOUT=1: 引脚 25 被配置为校准时钟 TOUT 输出
P223	<b>P2.2P2.3 端口功能配置位</b> 1: TX0/P2.2—RX0/P2.3 配置为标准 I/O, P2.2—P2.3 0: TX0/P2.2—RX0/P2.3 配置为 UART0, TX0—RX0
P201	<b>P2.0P2.1 端口功能配置位</b> 1: TD0/TX1/P2.0—TMS/RX1/P2.1 配置为标准 I/O, P2.0—P2.1 0: TD0/TX1/P2.0—TMS/RX1/P2.1 配置为 UART1, TX1—RX1
P067	<b>P0.6 端口功能配置位</b> 1: SEG22/P0.6 配置为 LCD 输出, SEG22 0: SEG22/P0.6 配置为标准 I/O, P0.6
P045	<b>P0.4 端口功能配置位</b> 1: SEG20/P0.4 配置为 LCD 输出, SEG20 0: SEG20/P0.4 配置为标准 I/O, P0.4
P023	<b>P0.2P0.3 端口功能配置位</b> 1: SEG18/P0.2—SEG19/P0.3 配置为 LCD 输出, SEG18—SEG19 0: SEG18/P0.2—SEG19/P0.3 配置为标准 I/O, P0.2—P0.3
P001	<b>P0.0P0.1 端口功能配置位</b> 1: SEG16/P0.0—SEG17/P0.1 配置为 LCD 输出, SEG16—SEG17 0: SEG16/P0.0—SEG17/P0.1 配置为标准 I/O, P0.0—P0.1

**注意:**

- 1, 只有在 SUPDC 中使能了 LCD\_EN=1, 同时配置 LCDCFG, P02CFG 为 LCD 使能有效的前提下, 以上配置的 I/O PIN 才配置为相应的 LCD 功能 PIN。如果在 SUPDC 中配置为 LCD\_EN=0, 关闭 LCD 功能。同时 PTA, PTB 又配置为 LCD 的功能输出口, 被配置为 LCD 输出口的 Com 和 Seg 都输出系统工作电压一致的 VCC 电平。
- 2, 如果 SUPDC 中的 EMU\_EN=0, 关闭芯片的 EMU 单元。而相应的 PIN 被配置为 PF, QF 的情况下, 这三个 PIN 输出为低电平。
- 3, 在 JTAG 模式 (WDTEN pin 被拉到低电平), TD0/TX1/P2.0—TMS/RX1/P2.1, 两个 PIN 被配置使能为 TD0、TMS 功能。
- 4, 此寄存器带 REG1 类型写保护, 需连续写入 BWPM 值为 CFH, DCH 才能改写此寄存器。

表 5-6-5 P3CFG P3 输出复用配置 (CFH)

P3CFG			Address: CFH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	INT0	I2C	X
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
INT0	<b>INT0 端口功能配置位</b> 1: P3.2/INT0 配置为, /INT0 0: P3.2/INT0 配置为标准 IO, P3.2
I2C	<b>I2C 端口功能配置位</b>

	1: P3.0/SCL, P3.1/SDA 配置为 I2C 总线功能接口 0: P3.0/SCL, P3.1/SDA 配置为标准 IO, P3.0, P3.1
--	--

注意:

- 1) 如果 SUPDC 中的 I2C\_EN=0, 关闭 I2C 功能模块。同时在 P3CFG 中又配置为 I2C=1, P3.0/SCL, P3.1/SDA 配置为 I2C 总线功能接口 SCL, SDA 功能 PIN 的时候, 是默认为输入功能, 弱上拉。
- 2) 此寄存器带 REG1 型写保护, 需连续对 BWPM 写入 CFH, DCH 才能改写此寄存器。

### 5.6.2.2 P0

#### ① P0 口数据寄存器

P0		Address: 80H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	P0.6	X	P0.4	P0.3	P0.2	P0.1	P0.0
Write:								
Reset:	u	u	u	u	u	u	u	u

#### ② P0 口方向寄存器

DDRP0		Address: D5H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	DDRP06	X	DDRP04	DDRP03	DDRP02	DDRP01	DDRP00
Write:								
Reset:	0	0	0	0	0	0	0	0

0: 配置为输入模式      1: 配置为输出模式

#### ③ P0 口 LED 控制寄存器

LEDP0		Address: C1H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	LEDP06	X	LEDP04	LEDP03	LEDP02	LEDP01	LEDP00
Write:								
Reset:	0	0	0	0	0	0	0	0

0: 配置为标准驱动 IO      1: 配置成 LED 驱动, 具有 10mA 的吸电流能力

注意: 只有在方向寄存器 DDRP0x 配置成输出模式时才有效。

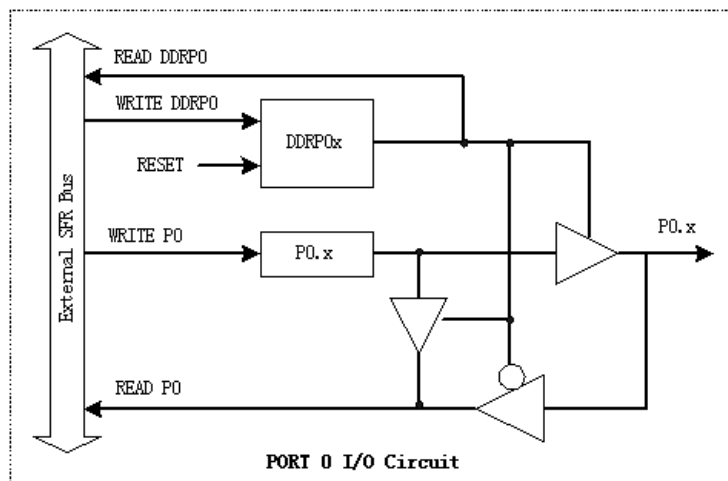


图 5-8 -1

### 5.6.2.3 P1

P1 口是一个与键盘的 KEY0-KEY3 以及和 SPI 功能复用的 8 位并行端口。

#### ① P1 口数据寄存器

P1			Address: 90H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	P1.1	P1.0
Write:								
Reset:	u	u	u	u	u	u	u	u

u: undefined, 未定义的

#### ② P1 口方向寄存器

DDRP1			Address: D6H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	DDRP11	DDRP10
Write:								
Reset:	0	0	0	0	0	0	0	0

0: 配置为输入模式      1: 配置为输出模式

### 5.6.2.4 P2

P2 口是一个与 RX0/TX0/RX1/TX1/TMUXOUT/PF/QF/SF 复用的 8 位并行端口。

#### ① P2 口数据寄存器

P2			Address: A0H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	P2.6	X	X	P2.3	P2.2	P2.1	P2.0
Write:								
Reset:	u	u	u	u	u	u	u	u

u: undefined, 未定义的

#### ② P2 口方向寄存器

DDRP2			Address: D7H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	DDRP26	X	X	DDRP23	DDRP22	DDRP21	DDRP20
Write:								
Reset:	0	0	0	0	0	0	0	0

0: 配置为输入模式      1: 配置为输出模式

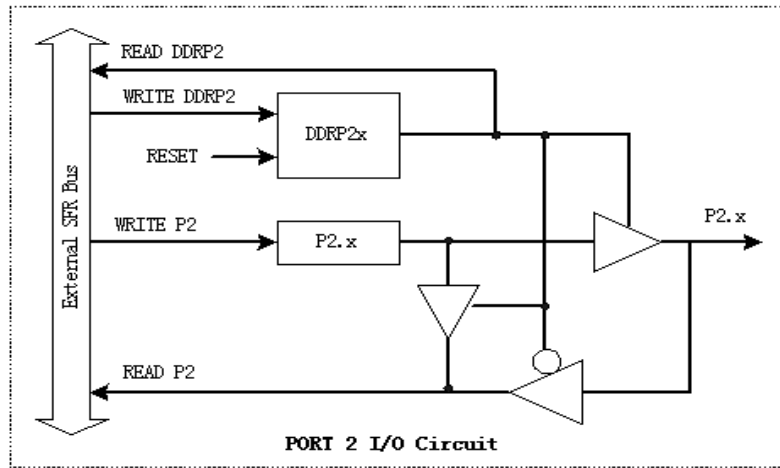


图 5-8-2

### 5.6.2.5 P3

#### ① P3 口数据寄存器

P3		Address: B0H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	P3.2	P3.1	P3.0
Write:								
Reset:	u	u	u	u	u	u	u	u

u: undefined, 未定义的

#### ② P3 口方向寄存器

DDRP3		Address: D9H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	DDRP32	DDRP31	DDRP30
Write:								
Reset:	0	0	0	0	0	0	0	0

0: 配置为输入模式

1: 配置为输出模式

### 5.6.2.6 Port A

#### ① PTA口数据寄存器

PTA		Address: BCH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	PTA7	PTA6	PTA5	PTA4	X	X	X	X
Write:								
Reset:	u	u	u	u	u	u	u	u

u: undefined, 未定义的

#### ② PA口方向寄存器

DDRA		Address: D1H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	DDRA7	DDRA6	DDRA5	DDRA4	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0



0: 配置为输入模式      1: 配置为输出模式

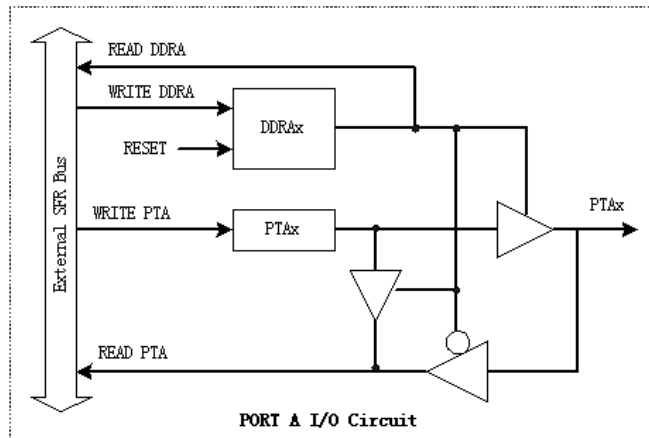


图 5-8 -3

### 5.6.2.7 Port B

PTB 口是一个与 LCD 的 SEG8-SEG15 复用的 8 位并行端口。

① PTB口数据寄存器

PTB		Address: BDH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	PTB3	PTB2	PTB1	PTB0
Write:								
Reset:	u	u	u	u	u	u	u	u

u: undefined, 未定义的

② PB口方向寄存器

DDRB		Address: D2H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	DDRB3	DDRB2	DDRB1	DDRB0
Write:								
Reset:	0	0	0	0	0	0	0	0

0: 配置为输入模式      1: 配置为输出模式

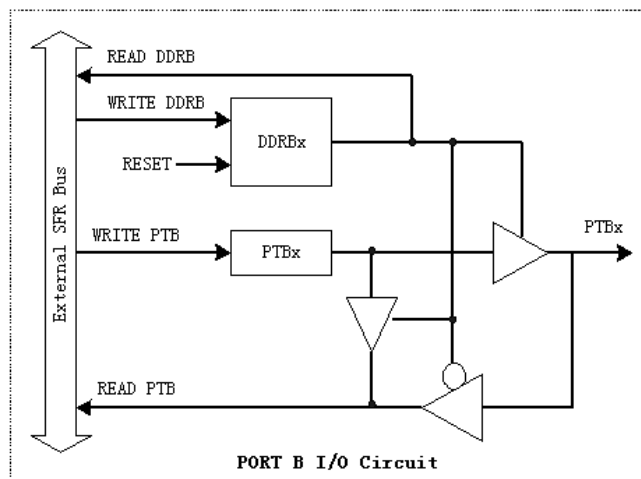


图 5-8 -4

## 6 通信接口

### 6.1 UART

#### 6.1.1 概述

ATT7039 支持两个串口：串口 0 和串口 1。

串口 0 提供一个灵活的全双工异步通信的接收器/发送器，支持三种异步工作方式：

- 方式 1：通过 TXD 发送或通过 RXD 接收 8 个数据位，波特率是可变的。
- 方式 2：通过 TXD 发送或通过 RXD 接收 9 个数据位，波特率可编程为振荡器频率的 1/32 或 1/64。
- 方式 3：通过 TXD 发送或通过 RXD 接收 9 个数据位，波特率是可变的。

串口 1 提供一个灵活的全双工异步通信的接收器/发送器，有两种工作模式：

- 方式 A：通过 TXD 发送或通过 RXD 接收 9 个数据位，波特率是可变的。
- 方式 B：通过 TXD 发送或通过 RXD 接收 8 个数据位，波特率是可变的。

#### 6.1.2 串口 0

##### 6.1.2.1 模块框图

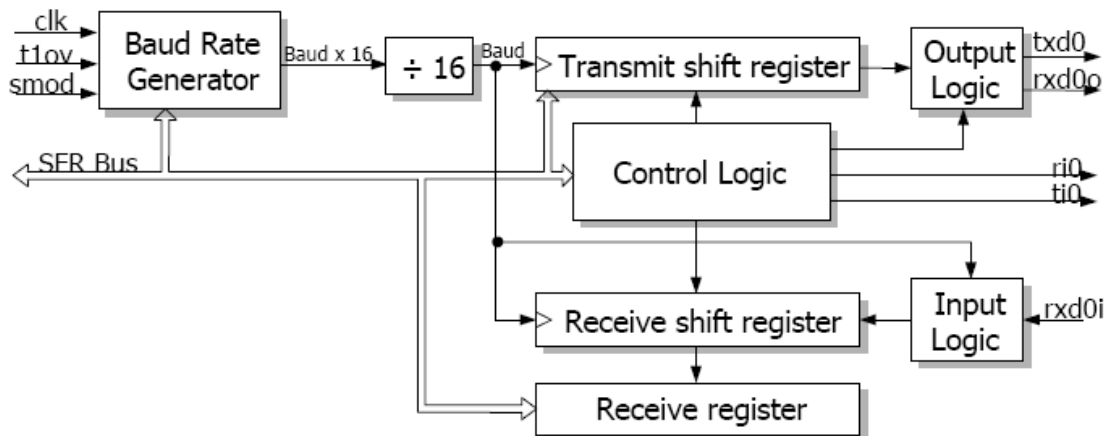


图 6-1-1

##### 6.1.2.2 波特率

在串口 0 的方式 2 时，波特率有两种选择：当 PCON.7(即 SMOD 位)=1 时，波特率为振荡器频率的 1/32；当 PCON.7(即 SMOD 位)=0 时，波特率为振荡器频率的 1/64。

在串口 0 的模式 1、模式 3 时，波特率是可变的。

当 BD (ADCON.7) =1 时，波特率由波特率生成器和 SMOD 位的值确定：

$$\text{波特率} = \frac{2^{SMOD} \times f_{cpu}}{64 \times (2^{10} - SOREL)}$$

其中 SMOD 是 SFR PCON.7 的值；SOREL 是 10bit 无符号数，寄存器 SORELH 和 SORELL 的值；f<sub>cpu</sub> 是系统时钟。

下表列举了常用波特率对应的 S0REL 值 (ADCON.7=1, PCON.7=1, fcpu=5.5M):

常用波特率	S0RELH	S0RELL
300	0x01	0xC3
600	0x02	0xE1
1200	0x03	0x71
2400	0x03	0xB8
4800	0x03	0xDC
9600	0x03	0xEE
14400	0x03	0xF4
19200	0x03	0xF7
38400	0x03	0xFC

当 BD (ADCON.7) = 0 时, 波特率由定时器 1/定时器 2 的溢出速率和 SMOD 位的值确定:

$$\text{波特率} = \frac{2^{\text{SMOD}}}{32} \times \text{Timer1 溢出速率}$$

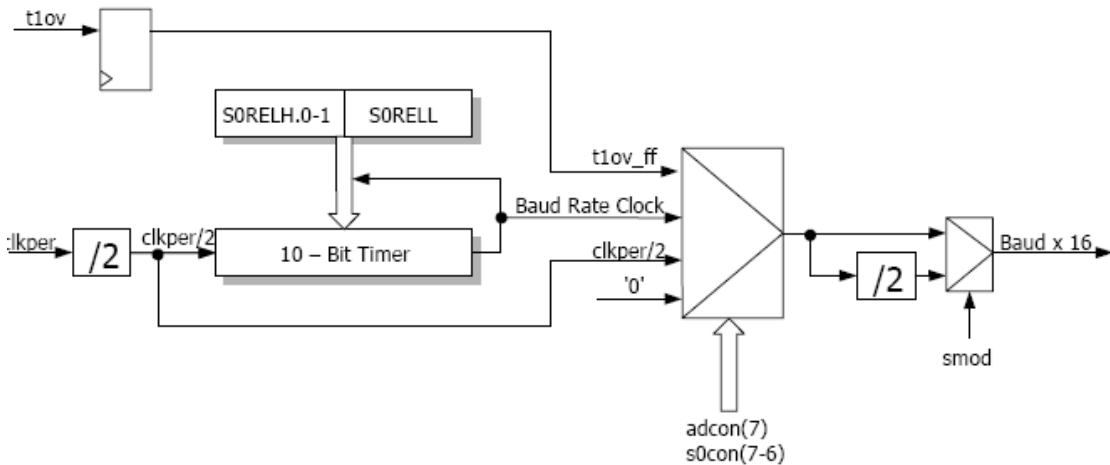


图 6-1-2

### 6.1.2.3 模式 1

模式 1 是一种标准的异步通信方式, 每帧包含 10 位数据信息: 1 位起始位 (0), 8 位数据位 (低位在前), 1 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:



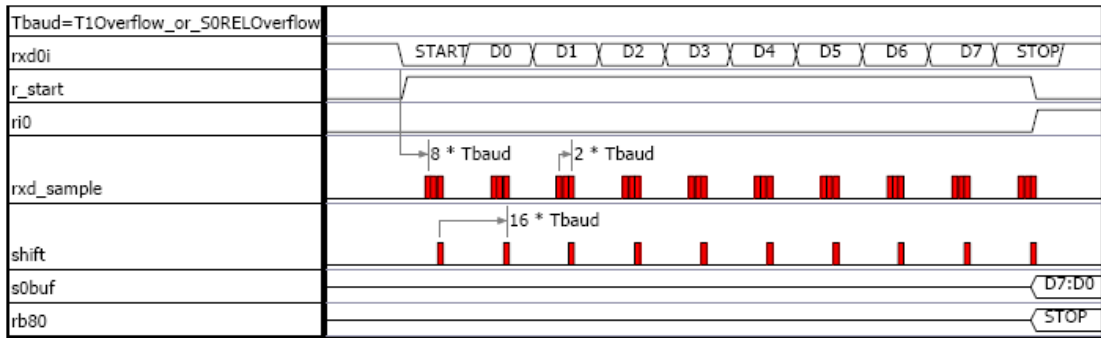


图 6-1-3 模式 1 时串行口数据信息

在模式 1 中，可以用中断方式进行数据通信。当一帧中最后一个数据发送完时，发送中断标志 TI 置“1”；同时，接收完最后一个数据位时，接收中断标志 RI 置“1”。

### 6.1.2.4 模式 2

模式 2 的波特率固定为系统时钟的 1/32 或 1/64（取决于 pcon 寄存器的位 smod）。

模式 2 是使用第 9 位数据的通信方式，每帧包含 11 位数据信息：1 位起始位（0），8 位数据位（低位在前），1 位可编程的第 9 位数据位，1 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：

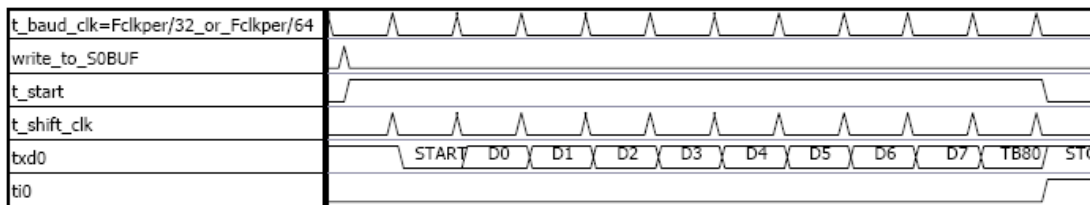


Figure 69. Serial0 transmission in mode 2

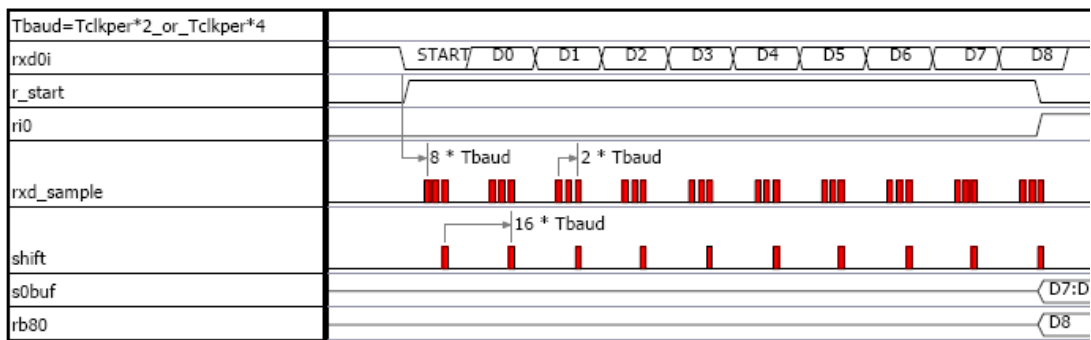


图 6-1-4 模式 2 时串行口数据信息

在数据发送时，串行口控制寄存器中的 TB8\_0 作为第 9 位数据同时发送，发送完毕 TB8\_0 被清除。

RXD0 引脚为数据输入端，当 RXD0 引脚的下降沿出现，开始接收。接收完成后，数据存放在 S0BUF 寄存器，第 9 位数据存放在寄存器 S0CON 的 RB8\_0 位。

数据收发时，第 9 位数据可作为奇/偶校验或数据/地址标志位使用。

### 6.1.2.5 模式 3

模式 3 的数据传送方式和模式 2 相同，只是在波特率选择上不同。按照模式 3 进行数据传送时允许奇偶校验。数据收发时，第 9 位数据可作为奇/偶校验或数据/地址标志位使用。

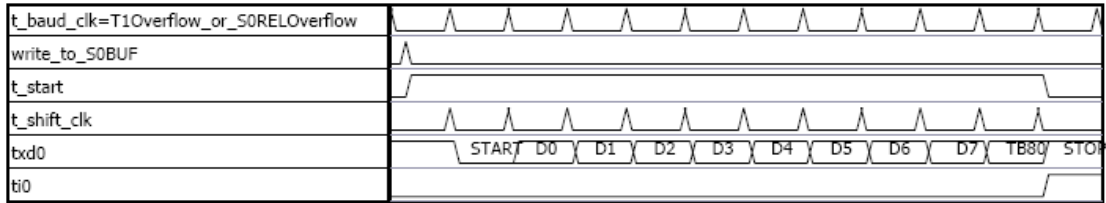


Figure 71. Serial0 transmission in mode 3

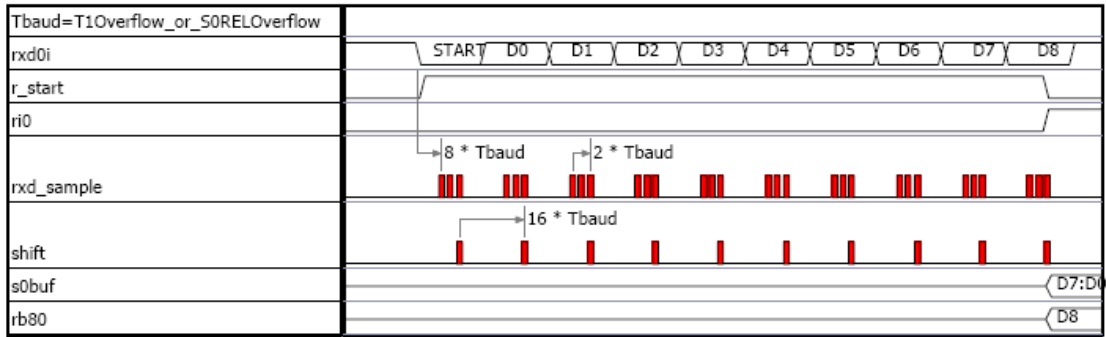


图 6-1-5

6.1.2.6 多处理器通讯

模式 2 和模式 3 中每字节接收 9 位数据的特性，可以实现多机通信的特殊功能。

当 S0CON 的位 SM2\_0 置 1 时，只有接受到的第 9 位数据（S0CON 的 RB8\_0）为 1 时，才会产生接收中断，否则不会有接收中断产生。

当主机要发送一批数据给几个从机之一时，首先发送一帧表明从机地址的地址字节信息，且在地址信息中第 9 位为 1。从机的 SM2\_0 都设定为 1，因此所有从机被中断而接收地址信息。当某从机的地址与所接收到的地址一致时，表明该从机被寻址，将其 SM2\_0 清零，准备接收将传送的数据字节。此后主机发送数据且保证第 9 位一直为 0，这样已被寻址的从机可接收数据，未被寻址的从机忽略之后所传送的数据字节。

6.1.3 串口 1

6.1.3.1 模块框图

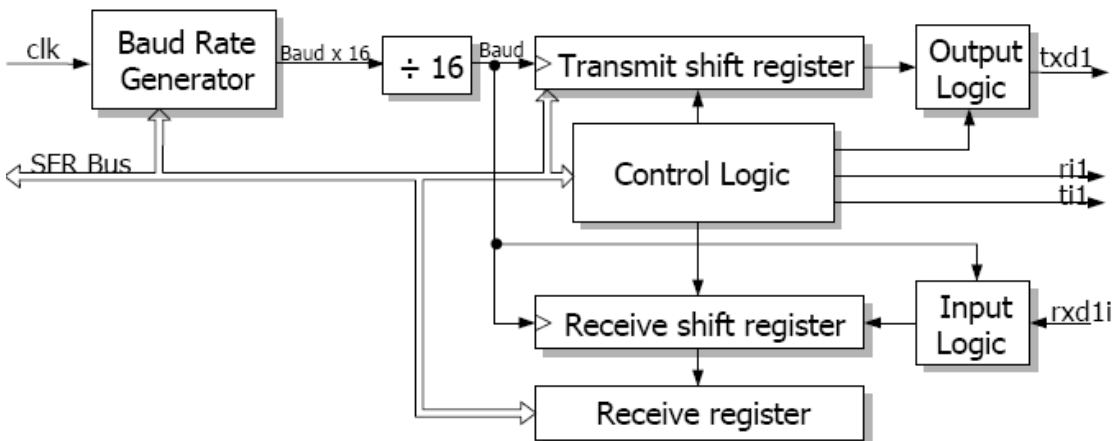


图 6-1-6

### 6.1.3.2 波特率

在串口 1 的方式 A、方式 B 中，波特率是可变的，采用内部波特率生成器，波特率计算公式为：

$$\text{波特率} = \frac{f_{cpu}}{32 \times (2^{10} - S1REL)}$$

其中 S1REL 是 10bit 无符号数，寄存器 S1RELH 和 S1RELL 的值；fcpu 是系统时钟。

下表列举了常用波特率对应的 S1REL 值（fcpu=5.5M）：

常用波特率	S1RELH	S1RELL
300	0x01	0xC3
600	0x02	0xE1
1200	0x03	0x71
2400	0x03	0xB8
4800	0x03	0xDC
9600	0x03	0xEE
14400	0x03	0xF4
19200	0x03	0xF7
38400	0x03	0xFC

串口 1 的波特率如下图所示：

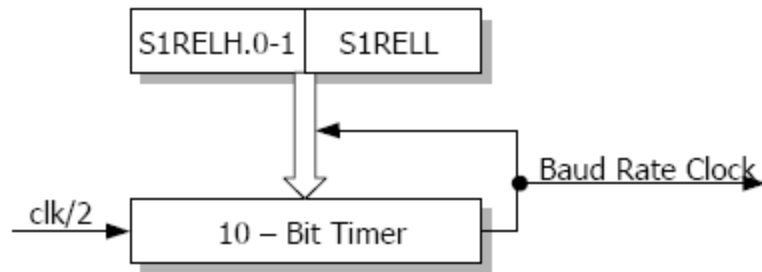


Figure 74. Serial1 Baud rate generation diagram

### 6.1.3.3 模式 A

模式 A 是一种异步通信方式，每帧包含 9 位数据信息，波特率可变。波特率生成器“S1RELH”“S1RELL”用于同步输入和输出传输。串口 1 的波特率不受寄存器“pcon”的“smode”位作用。

TXD1 引脚为数据输出端，当向寄存器“S1BUF”中写入数据后，传输开始。输出数据每帧包含 11 位数据信息：1 位起始位（0），8 位数据位（低位在前），1 位可编程的第 9 位数据位（来自寄存器“S1CON”的“tb81”位），1 位停止位（1）。

RXD1 引脚为数据输入端，当 RXD1 引脚的下降沿出现，开始接收。寄存器“S1BUF”完成接收后，输入数据才可继续，第 9 位数据存放在寄存器“S1CON”的“rb81”位。在接收期间，寄存器“S1BUF”“rb81”保持不变，直到接收完成。

数据收发时，第 9 位数据可作为奇/偶校验或数据/地址标志位使用。

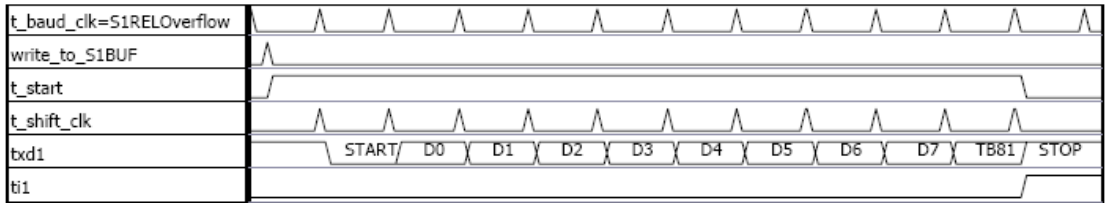


Figure 75. Serial1 transmission in mode A

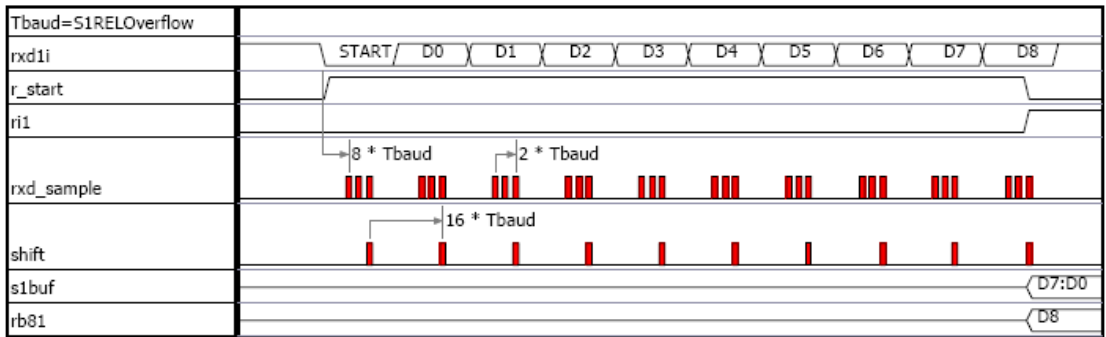


图 6-1-7

#### 6.1.3.4 模式 B

模式 B 是一种异步通信方式，每帧包含 8 位数据信息，波特率可变。波特率生成器“S1RELH”“S1RELL”用于同步输入和输出传输。波特率不受寄存器“PCON”的“sm0d”位作用。

TXD1 引脚为数据输出端，当向寄存器“S1BUF”中写入数据后，传输开始。输出数据每帧包含 10 位数据信息：1 位起始位（0），8 位数据位（低位在前），1 位停止位（1）。

RXD1 引脚为数据输入端，当 RXD1 引脚的下降沿出现，开始接收。寄存器“S1BUF”完成接收后，输入数据才可继续。在接收期间，寄存器“S1BUF”“rb81”保持不变，直到接收完成。

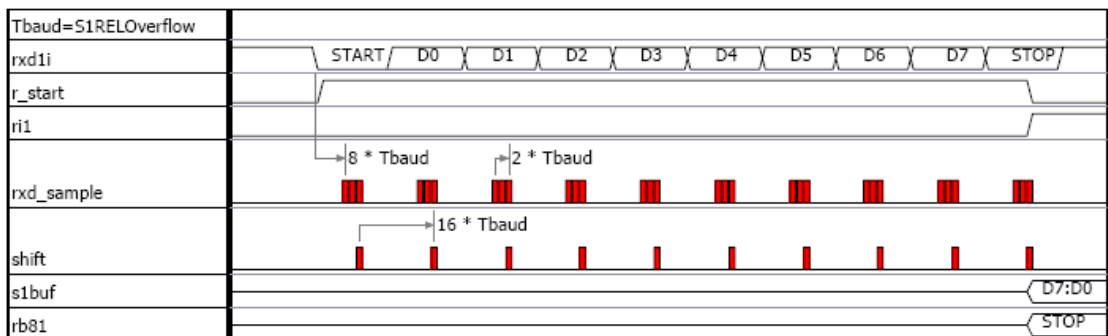
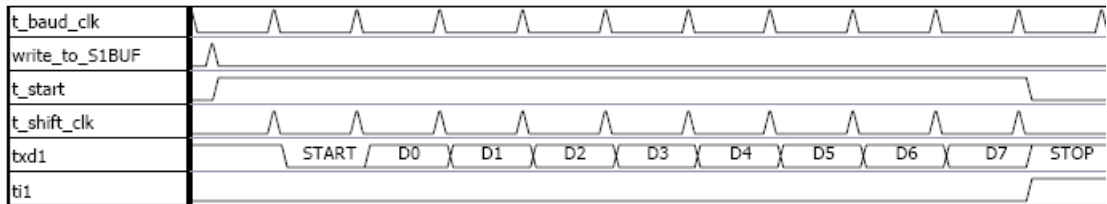


图 6-1-8

### 6.1.3.5 多处理器通讯

模式 A 中每字节接收 9 位数据的特性，可以实现多机通信的特殊功能。

若寄存器 S1CON 中的位”sm21”被置 1，接收中断只会在接收到的第 9 个字节是 1 的时候产生。否则，在接收的时候没有中断产生。

为了利用这个特性实现多处理器通讯，需要将处理器的”sm21”位设置为 1。主处理器传输从机地址，同时将第 9 位置 1，这样在所有的从机中触发接收中断。从处理器软件程序比较接收到的字节与自己的网络地址是否一致。如果一致，则清除”sm21”标志位，并且接收从主机传送过来的其他信息中，第 9 位一直保持为 0；其他的从机保持标志位”sm21”为 1，以忽略该主机传送过来的其他信息。

### 6.1.4 特殊功能寄存器

表 6-1-1 UART0 和 UART1 寄存器列表

地址	名称	复位值	功能描述
0x98	S0CON	0x00	串口 0 控制寄存器
0x99	S0BUF	0x00	串口 0 数据缓冲寄存器
0x9b	S1CON	0x00	串口 1 控制寄存器
0x9c	S1BUF	0x00	串口 1 数据缓冲寄存器
0x87	PCON	0x08	串口 0 波特率控制位
0xD8	ADCON	0x00	串口 0 波特率控制位
0xBA	S0RELH	0x03	串口 0 波特率发生器的重载值
0xAA	S0RELL	0xd9	
0xBB	S1RELH	0x03	串口 1 波特率发生器的重载值
0x9D	S1RELL	0x00	
0xCE	P02CFG	0x30	P2P0 输出复用配置寄存器，写保护
0xA8	IEN0	0x00	中断使能位 (Uart0)
0x9A	IEN2	0x00	中断使能位 (Uart1)

表 6-1-2 串口 0 控制寄存器(S0CON 0x98H)

串口 0 控制寄存器(S0CON)			Address: 98H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	SM0_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TL_0	RL_0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述				
SCON0.7	串行工作方式选择位:选择串行的三种工作方式之一				
	SM0_0	SM1_0	模式	描述	波特率
	0	1	模式1	8位串口	可变，详细内容参考波特率产生方式
	1	0	模式2	9位串口	可编程为系统频率的1/32或1/64
SCON0.6	1	1	模式3	9位串口	可变，详细内容参考波特率产生方式
SCON0.5	多机交互使能位				
SCON0.4	允许接收位 由软件置位/复位；REN_0置位时允许接收。				



SCON0.3	<b>发送数据位8</b> 由软件置位/复位；方式2、3中是被发出去的第9数据位。
SCON0.2	<b>接收数据位8</b> 方式2、3中是收到的第9数据位；方式1中若SM2_0=0，则为收到的停止位
SCON0.1	<b>发送中断标志</b> 方式0中，在发送的第8位刚结束时由硬件置位；其他方式时，在开始发送停止位时由硬件置位；必须由软件复位。
SCON0.0	<b>接收中断标志</b> 方式0中，在接收的第8位刚结束时由硬件置位；其他方式时，在接收停止位的中间时刻由硬件置位；必须由软件复位。

表 6-1-3 串口 0 缓冲寄存器(S0BUF 0x99H)

串口 0 缓冲寄存器(S0BUF)		Address: 99H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	S0BUF7	S0BUF6	S0BUF5	S0BUF4	S0BUF3	S0BUF2	S0BUF1	S0BUF0
Write:								
Reset:	0	0	0	0	0	0	0	0

对寄存器 S0BUF 写操作，则串口 0 将开始向外传输发送缓存数据；对寄存器 S0BUF 读操作，则串口 0 将从串行接收缓存中读取数据。

表 6-1-4 串口 1 控制寄存器(S1CON 0x9BH)

串口 1 控制寄存器(S1CON)		Address: 9BH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	SM	X	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SM	<b>串口1模式选择</b> sm = 0: 模式A sm = 1: 模式B
SM2_1	<b>多机通讯允许位</b>
REN_1	<b>允许接收位</b> 由软件置位/复位；REN_1置位时允许接收。
TB8_1	<b>发送数据位8</b> 由软件置位/复位； 在模式A中，是被发送出去的第9数据位；
RB8_1	<b>接收数据位8</b> 在模式A中，是收到的第9数据位； 在模式B中，若sm2_1 =0，则为收到的停止位；
TI_1	<b>发送中断标志</b> 方式A或B中，在开始发送停止位时由硬件置位；必须由软件复位。
RI_1	<b>接收中断标志</b> 方式A或B中，在接收停止位的中间时刻由硬件置位；必须由软件复位。

表 6-1-5 串口 1 缓冲寄存器(S1BUF 0x9CH)

串口 1 缓冲寄存器(S1BUF)		Address: <b>9CH</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:	S1BUF7	S1BUF6	S1BUF5	S1BUF4	S1BUF3	S1BUF2	S1BUF1	S1BUF0
Write:								
Reset:	0	0	0	0	0	0	0	0

对寄存器 S1BUF 写操作，则串口 1 将开始向外传输发送缓存数据；对寄存器 S1BUF 读操作，则串口 1 将从串行接收缓存中读取数据。

表 6-1-6 P Control Register(PCON 0x87H)

P Control Register (PCON)		Address: <b>87H</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:	smod	X	X	PWM	Reserved	X	X	X
Write:								
Reset:	0	0	0	0	1	0	0	0

SMOD: 串口 0 波特率控制位。SMOD=0, 波特率不受该位影响；SMOD=1, 波特率加倍。

Reserved: 该 Bit 位应该写为“0”。

表 6-1-7 AD Control Register(ADCON 0xD8H)

AD Control Register (ADCON)		Address: <b>D8H</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:	bd	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0

该寄存器的 Bd 位控制串口 0 的波特率生成方式，详细内容参见波特率生成方式。

表 6-1-8 串口 0 波特率重载寄存器(SORELL, AAH)

串口 0 缓冲寄存器(SORELL)		Address: <b>AAH</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:	SORELL7	SORELL6	SORELL5	SORELL4	SORELL3	SORELL2	SORELL1	SORELL0
Write:								
Reset:	0	0	0	0	0	0	0	0

表 6-1-9 串口 0 波特率重载寄存器(SORELH, BAH)

串口 0 缓冲寄存器(SORELH)		Address: <b>BAH</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:	SORELH7	SORELH6	SORELH5	SORELH4	SORELH3	SORELH2	SORELH1	SORELH0
Write:								
Reset:	0	0	0	0	0	0	0	0

对寄存器 SORELL 和 SORELH 写操作，通过配置寄存器改变 UART0 的通讯波特率。

表 6-1-10 串口 1 波特率重载寄存器(S1RELL, 9DH)

串口 1 缓冲寄存器(S1RELL)		Address: <b>9DH</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:	S1RELL7	S1RELL6	S1RELL5	S1RELL4	S1RELL3	S1RELL2	S1RELL1	S1RELL0
Write:								
Reset:	0	0	0	0	0	0	0	0

表 6-1-11 串口 1 波特率重载寄存器(S2RELH, BBH)

串口 1 缓冲寄存器(S1RELH)		Address: <b>BBH</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:	S1RELH7	S1RELH6	S1RELH5	S1RELH4	S1RELH3	S1RELH2	S1RELH1	S1RELH0
Write:								
Reset:	0	0	0	0	0	0	0	0

对寄存器 S1RELL 和 S1RELH 写操作，通过配置寄存器改变 UART1 的通讯波特率。

表 6-1-12 P02CFG P0 和 P2 输出复用配置寄存器 (CEH)

P02CFG		Address: <b>CEH</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:	P267	P245	P223	P201	P067	P045	P023	P001
Write:								
Reset:	0	0	1	1	0	0	0	0

位	功能描述
P223	<b>P2.2P2.3 端口功能配置位</b> 1: TX0/P2.2—RX0/P2.3 配置为标准 I/O, P2.2—P2.3 0: TX0/P2.2—RX0/P2.3 配置为 UART0, TX0—RX0
P201	<b>P2.0P2.1 端口功能配置位</b> 1: TD0/TX1/P2.0—TMS/RX1/P2.1 配置为标准 I/O, P2.0—P2.1 0: TD0/TX1/P2.0—TMS/RX1/P2.1 配置为 UART1, TX1—RX1

注：此寄存带 REG1 类型写保护，需连续写入 BWPM 值为 CFH, DCH 才能改写此寄存器。

表 6-1-13 IEN0 中断使能寄存器 0 (A8H)

Interrupt Enable 0 Register (IEN0)		Address: <b>A8H</b>						
	Bit7	6	5	4	3	2	1	Bit0
Read:	EAL	X	ET2	ES0	ET1	EX1	ET0	EX0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
EAL	<b>全体中断使能位</b> 0: 关闭所有中断; 1: 全体中断使能, 若要打开某一个中断, 还需要打开它对应的中断使能位;
ES0	<b>串口 0 中断使能</b> 0: 串口 0 中断关闭; 1: 并且 EAL=1, 串口 0 中断使能;

表 6-1-15 IEN2 中断使能寄存器 2 (9AH)

Interrupt Enable 0 Register (IEN2)			Address: <b>9AH</b>					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	ES1
Write:	X	X	X	X	X	X	X	ES1
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ES1	<b>ES1 中断使能位</b> 0: 串口 1 中断关闭 1: 并且 EAL=1, 串口 1 中断使能

## 6.2 红外模块

### 6.2.1 概述

ATT7039 有两个串行通讯接口, UART0, UART1。两个串口的输出 TX0/TX1 都可以调制成 38K 红外信号。两个串口的输入 RX0/RX1 也都可以配置作为外部中断的输入, 方便实现 SLEEP 下的外部通信唤醒功能。主要功能如下:

- 能够单独的实现UART0, UART1在芯片进入SLEEP后唤醒芯片的功能。(可以实现红外通信唤醒功能, 或者RS485通信唤醒)。
- 红外通信中38KHZ的调制信号是占空比50%的方波。(红外信号的最大波特率不超过2400bps)。

### 6.2.2 特殊功能寄存器

表 6-2-1 寄存器列表

地址	名称	复位值	字节长度	功能描述
B1H	PWMCR	0x00	1	PWM控制寄存器

表6-2-2 PWM控制寄存器

PWM Control Register (PWMCR)			Address: B1H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	IRTX1	IRTX0	X	X	X	X
Write:	X	X	IRTX1	IRTX0	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
IRTX1	<b>Uart1 口的红外调制功能选择位</b> 1: 使能 Uart1 通讯口 TX1 功能 PIN 的红外调制电路模块 0: 关闭 Uart1 通讯口 TX1 功能 PIN 的红外调制电路模块
IRTX0	<b>Uart0 口的红外调制功能选择位</b> 1: 使能 Uart0 通讯口 TX1 功能 PIN 的红外调制电路模块 0: 关闭 Uart0 通讯口 TX1 功能 PIN 的红外调制电路模块

## 6.3 I2C

### 6.3.1 概述

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

### 6.3.2 框图

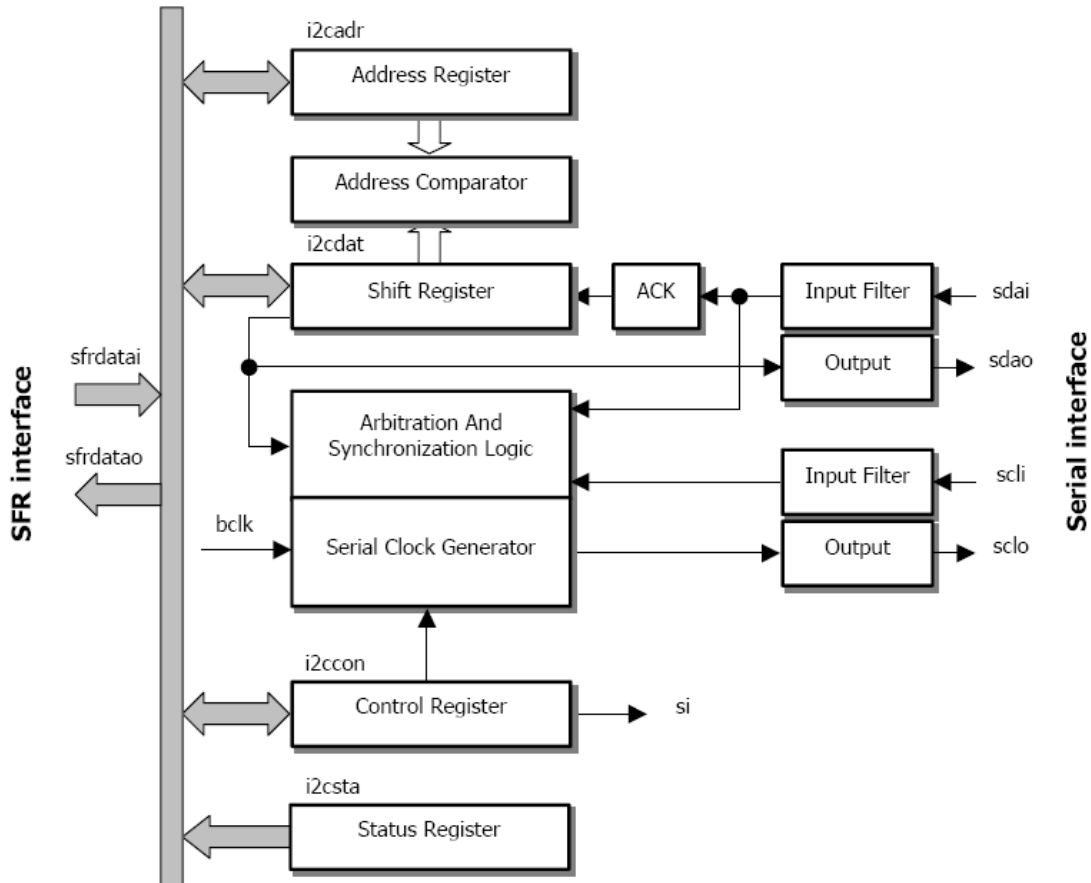


图 6-3-1 I2C 模块功能框图

### 6.3.3 功能描述

I2C 用两根线实现设备与总线之间的数据传输：“scl”（串行时钟线）和“sda”（串行数据线）。每一个与总线相连的设备都有一个可软件定义的唯一地址。I2C 是一个真正的多主机总线，它包含冲突检测和仲裁机制，以防止两个以上主机同时开始数据传输时的数据丢失。

#### 6.3.3.1 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 “sdao” 输出，串行时钟通过 “sclo” 输出
- 主机接收模式：串行数据通过 “sdai” 输入，串行时钟通过 “sclo” 输出
- 从机接收模式：串行数据通过 “sdai” 输入，串行时钟通过 “scli” 输入

- 从机发送模式：串行数据通过“sdao”输出，串行时钟通过“scli”输入

### 6.3.3.2 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“scli”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“cr0”、“cr1”、“cr2”控制。下表列出了在主机模式时“时钟发生器可能的输出频率，表中提到的时钟 bclk”与定时器 1 的溢出信号连接，就是说，I2C 的时钟可以由定时器 1 提供。

### 6.3.3.3 中断生成

IIC 产生中断时，寄存器“i2ccon”中的标志位“si”会被置位。

### 6.3.4 特殊功能寄存器

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：“i2ccon”（控制寄存器），“i2csta”（状态寄存器），“i2cdat”（数据寄存器），“i2cadr”（自身从机地址寄存器）。

表 6-3-1 寄存器列表

地址	名称	复位值	字节长度	功能描述
DAH	I2CDAT	00H	1	I2C数据寄存器
DBH	I2CADR	00H	1	I2C地址寄存器
DCH	I2CCON	00H	1	I2C控制寄存器
DDH	I2CSTA	F8H	1	I2C状态寄存器

表 6-3-2 I2CDAT I2C 数据寄存器 (DAH)

I2C Data Register (I2CDAT)			Address: DAH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2CDAT.7	I2CDAT.6	I2CDAT.5	I2CDAT.4	I2CDAT.3	I2CDAT.2	I2CDAT.1	I2CDAT.0
Write:								
Reset:	0	0	0	0	0	0	0	0

寄存器“i2cdat”包含一个字节的的数据，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。

表 6-3-3 I2CADR I2C 地址寄存器 (DBH)

I2C Address Register (I2CADR)			Address: DBH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2CADR.7	I2CADR.6	I2CADR.5	I2CADR.4	I2CADR.3	I2CADR.2	I2CADR.1	I2CADR.0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
i2cadr.7 i2cadr.6 i2cadr.5 i2cadr.4	I2C从机地址 (7 位)

i2cadr.3 i2cadr.2 i2cadr.1	
i2cadr.0	<b>呼叫地址确认位 General Call Address Acknowledge</b> 当此位置1时, 呼叫地址可以被识别, 否则不能被识别。 If this bit is set, the general call address is recognized; otherwise it is ignored.

寄存器“i2cadr”包含做为从机时的地址和“gc”标志位, gc标志位置1时使能对广播地址的识别。

表 6-3-4 I2CCON I2C 控制寄存器 (DCH)

I2C Control Register (I2CCON)		Address: DCH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CR2	<b>Clock rate bit 2</b>
ENS1	<b>I2C 使能位</b> 1: 打开IIC模块; 0: 关闭IIC模块;
STA	<b>开始标志位</b> 1: 检查IIC总线的状态, 如果空闲则生成开始信号; 0: 不会生成开始信号;
STO	<b>停止标志位</b> 1: 当处于主机模式, 则向总线传输停止信号 0: 不向总线传输停止信号;
SI	<b>中断标志位</b> 当进入25种IIC状态之一时, “si”由硬件置位, 唯一不置位的状态是“F8h”; 写“0”清0, 写“1”无影响。
AA	<b>生成应答标志位</b> 1: 应答在以下情况下被返回: 接收到自身作为从机的地址; gc被置位的情况下接收到地址呼叫; 主机接收模式下一个字节接收完成; 从机接收模式下一个字节接收完成; 0: 非应答在以下情况下被返回: 主机接收模式下一个字节接收完成; 从机接收模式下一个字节接收完成;
CR1	<b>Clock rate bit 1</b>
CR0	<b>Clock rate bit 0</b>



cr2	cr1	cr0	Bit frequency				Clk divided by	
			6 MHz	12 MHz	16 MHz	24 MHz		
0	0	0	23	47	63	92	256	
0	0	1	27	54	71	108	224	
0	1	0	31	63	83	124	192	
0	1	1	37	75	100	148	160	
1	0	0	6.25	12.5	17	25	960	
1	0	1	50	100	133	200	120	
1	1	0	100	200	266	400	60	
1	1	1	"bclk" input divided by 8					

寄存器“i2ccon”包含 I2C 模块的使能位“ens1”，时钟频率控制位（“cr0”，“cr1”，“cr2”），“开始”和“停止”信号的发送位，控制 ACK 信号的标志位“aa”，和中断标志位“si”。

表 6-3-5 I2CSTA I2C 状态寄存器 (DDH)

I2C Status Register (I2CSTA)			Address: DDH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	i2csta.7	i2csta.6	i2csta.5	i2csta.4	i2csta.3	i2csta.2	i2csta.1	i2csta.0
Write:								
Reset:	1	1	1	1	1	0	0	0

位	功能描述
i2csta.7	<b>I2C状态码</b> <b>I2C Status Code</b>
i2csta.6	
i2csta.5	
i2csta.4	
i2csta.3	
i2csta.2	忽略，始终读为“0”
i2csta.1	Not implemented, read as 0
i2csta.0	

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

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

I2C 主机发送模式状态描述如下:

Status code	Status of the I2C	Application software response to/from I2CDAT	Application software response to I2CCON				Next action taken by the I2C hardware
			sta	sto	si	aa	
08H	START condition has been transmitted	Load SLA+W	X	0	0	X	SLA+W will be transmitted ACK will be received
10H	Repeated START condition has been transmitted	Load SLA+W	X	0	0	X	As above
		Load SLA+R	X	0	0	X	SLA+R will be transmitted I2C will be switched to "master receiver" mode
18H	SLA+W has been transmitted; ACK has been received	Load data byte	0	0	0	X	Data byte will be transmitted; ACK will be received
		or no action	1	0	0	X	Repeated START will be transmitted;
		or no action	0	1	0	X	STOP condition will be transmitted; the "sto" flag will be reset
		or no action	1	1	0	X	STOP condition followed by a START condition will be transmitted; the "sto" flag will be reset
20H	SLA+W has been transmitted; "not ACK" has been received	Load data byte	0	0	0	X	Data byte will be transmitted; ACK will be received
		or no action	1	0	0	X	Repeated START will be transmitted
		or no action	0	1	0	X	STOP condition will be transmitted; the "sto" flag will be reset
		or no action	1	1	0	X	STOP condition followed by a START condition will be transmitted; the "sto" flag will be reset
28H	Data byte in i2cdat has been transmitted; ACK has been received	Load data byte	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		or no action	1	0	0	X	Repeated START will be transmitted
		or no action	0	1	0	X	STOP condition will be transmitted; the "sto" flag will be reset
		or no action	1	1	0	X	STOP condition followed by a START condition will be transmitted; sto flag will be reset

图 6-3-6 I2C 主机发送模式状态 (1)

Status code	Status of the I2C	Application software response					Next action taken by the I2C hardware
		to/from I2CDAT	sta	sto	si	aa	
30H	Data byte in i2cdat has been transmitted	Data byte	0	0	0	X	Data byte will be transmitted; ACK will be received
		or no action	1	0	0	X	Repeated START will be transmitted;
		or no action	0	1	0	X	STOP condition will be transmitted; sto flag will be reset
		or no action	1	1	0	X	STOP condition followed by a START condition will be transmitted; sto flag will be reset
38H	Arbitration lost in SLA+R/W or data bytes	No action	0	0	0	X	I2C bus will be released; the "not addressed slave" state will be entered
		or no action	1	0	0	X	A START condition will be transmitted when the bus becomes free

图 6-3-7 I2C 主机发送模式状态 (2)

I2C 主机接受模式状态描述如下:

**Table 122. I2C Status In Master Receiver Mode**

Status code	Status of the I2C	Application software response					Next action taken by the I2C hardware
		to/from I2CDAT	sta	sto	si	aa	
08H	START condition has been transmitted	Load SLA+R	X	0	0	X	SLA+R will be transmitted; ACK will be received
10H	Repeated START condition has been transmitted	Load SLA+R	X	0	0	X	As above
		Load SLA+W	X	0	0	X	SLA+W will be transmitted; I2C will be switched to "master transmitter" mode
38H	Arbitration lost in "not ACK" bit	No action	0	0	0	X	I2C bus will be released; I2C will enter a "slave" mode
		no action	1	0	0	X	A start condition will be transmitted when the bus becomes free
40H	SLA+R has been transmitted; ACK has been received	No action	0	0	0	0	Data byte will be received; not ACK will be returned
		no action	0	0	0	1	Data byte will be received; ACK will be returned

图 6-3-8 I2C 主机接受模式状态 (1)

Status code	Status of the I2C	Application software response				Next action taken by the I2C hardware	
		to/from I2CDAT	sta	sto	si		aa
48H	SLA+R has been transmitted; "not ACK" has been received	No action	1	0	0	X	Repeated START condition will be transmitted
		or no action	0	1	0	X	STOP condition will be transmitted; the "sto" flag will be reset
		or no action	1	1	0	X	STOP condition followed by START condition will be transmitted; the "sto" flag will be reset
50H	Data byte has been received; ACK has been returned	Read data byte	0	0	0	0	Data byte will be received; "not ACK" will be returned
		or read data byte	0	0	0	1	Data byte will be received; ACK will be returned
58H	Data byte has been received; "not ACK" has been returned	Read data byte	1	0	0	X	Repeated START condition will be transmitted
		or read data byte	0	1	0	X	STOP condition will be transmitted; the "sto" flag will be reset
		or read data byte	1	1	0	X	STOP condition followed by START condition will be transmitted; the "sto" flag will be reset

图 6-3-9 I2C 主机接受模式状态 (2)

I2C 从机接受模式状态描述如下:

Status code	Status of the I2C	Application software response				Next action taken by the I2C hardware	
		to/from I2CDAT	sta	sto	si		aa
60H	Own SLA+W has been received; ACK has been returned	No action	X	0	0	0	Data byte will be received and "not ACK" will be returned
		or no action	X	0	0	1	Data byte will be received and ACK will be returned
68H	Arbitration lost in SLA+R/W as master; own SLA+W has been received, ACK returned	No action	X	0	0	0	Data byte will be received and "not ACK" will be returned
		or no action	X	0	0	1	Data byte will be received and ACK will be returned
70H	General call address (00H) has been received; ACK has been returned	No action	X	0	0	0	Data byte will be received and "not ACK" will be returned
		or no action	X	0	0	1	Data byte will be received and ACK will be returned

图 6-3-10 I2C 从机接受模式状态 (1)

78H	Arbitration lost in SLA+R/W as master; general call address has been received, ACK returned	No action	X	0	0	0	Data byte will be received and "not ACK" will be returned
		or no action	X	0	0	1	Data byte will be received and ACK will be returned
80H	Previously addressed with own SLV address; DATA has been received; ACK returned	Read data byte	X	0	0	0	Data byte will be received and "not ACK" will be returned
		or read data byte	X	0	0	1	Data byte will be received and ACK will be returned
88H	Previously addressed with own SLA; DATA byte has been received; "not ACK" returned	Read data byte	0	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address
		or read data byte	0	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized
		or read data byte	1	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free
		or read data byte	1	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free
90H	Previously addressed with general call address; DATA has been received; ACK returned	Read data byte	X	0	0	0	Data byte will be received and "not ACK" will be returned
		or read data byte	X	0	0	1	Data byte will be received and ACK will be returned

图 6-3-11 I2C 从机接受模式状态 (2)

Status code	Status of the I2C	Application software response				Next action taken by the I2C hardware	
		to/from I2CDAT	to i2CCON				
			sta	sto	si	aa	
98H	Previously addressed with general call address; DATA has been received; ACK returned	Read data byte	0	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address
		or read data byte	0	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized
		or read data byte	1	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free
		or read data byte	1	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free
A0H	STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX	No action	0	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address
		or no action	0	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized
		or no action	1	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free
		or no action	1	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free

图 6-3-12 I2C 从机接受模式状态 (3)

I2C 从机发送模式状态描述如下：

Status code	Status of the I2C	Application software response to/from I2CDAT	Application software response to i2CCON				Next action taken by the I2C hardware
			sta	sto	si	aa	
A8H	Own SLA+R has been received; ACK has been returned	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
		or load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received
B0H	Arbitration lost in SLA+R/W as master; own SLA+R has been received; ACK has been returned	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
		or load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received
B8H	Data byte has been transmitted; ACK has been received	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
		or load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received
C0H	Data byte has been transmitted; not ACK has been received	No action	0	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address
		or no action	0	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized
		or no action	1	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free
		or no action	1	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free

图 6-3-13 I2C 从机发送模式状态 (1)

Status code	Status of the I2C	Application software response				Next action taken by the I2C hardware	
		to/from I2CDAT	to i2CCON				
			sta	sto	si	aa	
C8H	Last data byte has been transmitted; ACK has been received	No action	0	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address
		or no action	0	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized
		or no action	1	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free
		or no action	1	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free

图 6-3-14 I2C 从机发送模式状态 (2)

Table 125. I2C Status - miscellaneous states

Status code	Status of the I2C	Application software response				Next action taken by the I2C hardware	
		to/from I2CDAT	to i2CCON				
			sta	sto	si	aa	
F8H	No relevant state information available; si=0	No action	No action				Wait or proceed current transfer
00H	Bus error during MST or selected slave modes	No action	0	1	0	X	Only the internal hardware is affected in the "master" or "addressed slave" modes. In all cases, the bus is released and I2C is switched to the "not addressed slave" mode. The "sto" flag is reset.

图 6-3-15 I2C 复合状态



## 7 电气规格

### 7.1 绝对最大额定值

测量项目	符号	最小	典型	最大	单位
数字电源电压	$V_{DD3p3}$	-0.3	--	3.8	V
模拟电源电压	$AV_{CC}$	-0.3	--	3.8	V
I/O 最大输出拉电流	$I_{source}$	--	--	$\pm 5$	mA
I/O 最大输入灌电流	$I_{sunk}$	--	--	$\pm 10$	mA
数字输入电压相对于GND	$V_{IND}$	-0.3	--	$V_{CC}+0.3$	V
模拟输入电压相对于AGND	$V_{INA}$	-0.3	--	$AV_{CC}+0.3$	V
工作温度范围	$T_A$	-40	--	85	$^{\circ}C$
存储温度范围	$T_{stg}$	-65	--	150	$^{\circ}C$

### 7.2 电器特性

#### 7.2.1 DC 参数

注意：以下表格中提到的 $V_{CC}$ 等于 $V_{DD3P3}$ 上的电压。(VCC=AVCC=3.3V, 温度范围:  $-40^{\circ}C \sim +85^{\circ}C$ )

测量项目	符号	最小	典型	最大	单位	
输出高电平	$V_{OH}$	$V_{CC}-0.4$	--	$V_{CC}$	V	
输出低电平	$V_{OL}$	--	--	0.4	V	
输入高电平	$V_{IH}$	$0.7*V_{CC}$	--	$V_{CC}$	V	
输入低电平	$V_{IL}$	$V_{SS}$	--	$0.3*V_{CC}$	V	
I/O 上拉电阻	$R_{PU}$		30		k $\Omega$	
LBOR		1.9		2.1		
POR 启动电压	$V_{POR}$	0	--	100	mV	
LVDIN (Analog Input)	Maximum Signal Levels	0	3.3	$\leq V_{DD3P3}$	V	
	Input Impedance (DC)		1		M $\Omega$	
	Low Voltage Detection Input Threshold		1.12	1.18	1.22	V
	VBAT Voltage Test Range	0		3.8		V
TBS	VBAT Voltage Measure		15.48		mv/LSB	
	Temperature Measure		0.615		$^{\circ}C/LSB$	
	TBS Conversion Delay		1		ms	

### 7.2.2 功耗参数

工作模式 (三路 ADC OPEN)	功耗
$f_{sys}=5.5\text{MHz}$ $f_s=900\text{KHz}$ $V_{CC}=3.3\text{V}$	4.87mA 注：关闭 1 路 ADC 会降低 0.78mA 的功耗 $P(\text{adc})=0.78\text{mA}$ $P(\text{emu})=1.2\text{mA}$ $P(\text{vref})=0.45\text{mA}$ $P(\text{cpu})=1.9\text{mA}$ $P(\text{i2c})=22\mu\text{A}$ $P(\text{lcd})=26\mu\text{A}$ $P(\text{Bor})=15\mu\text{A}$
PSM $f_{sys}=32\text{KHz}$ , $V_{CC}=3.3\text{V}$ , With LCD, RTC on	46 $\mu\text{A}$
Sleep Mode With RTC, LCD on	16.0 $\mu\text{A}$
Sleep Mode With RTC on, TCR 设置的分时开启时间最小	4.0 $\mu\text{A}$

### 7.2.3 电能计量参数

( $V_{CC}=AV_{CC}=3.3\text{V}$ , 室温)

测量项目	符号	最小	典型	最大	单位	测试条件
有功电能测量误差	Err		0.1%			常温 3000:1 的动态范围
有功电能测量带宽	Bp		14		kHz	
无功电能测量误差	Err		0.1%			常温 3000:1 的动态范围
电压有效值测量误差	Urms		0.5%			常温 1000:1 的动态范围
电压有效值测量带宽	Bu		14		kHz	
电流有效值测量误差	Irms		0.5%			常温 1000:1 的动态范围
电流有效值测量带宽	Bi		14		kHz	
两个通道间的相位误差 PF=0.8 容性 PF=0.5 感性			$\pm 0.04$ $\pm 0.04$		度 度	相位超前 37 度 相位滞后 60 度

### 7.2.4 ADC 指标

( $V_{CC}=AV_{CC}=3.3\text{V}$ , 室温)

测量项目	符号	最小	典型	最大	单位	测试条件
最大信号电平	$V_{xn}$			$\pm 800$	mV	模拟通道差分输入
直流输入阻抗	$Z_{DC}$	400			$k\Omega$	
信噪比	SNR		86		dB	
-3dB 带宽	$B_{-3dB}$		14		kHz	

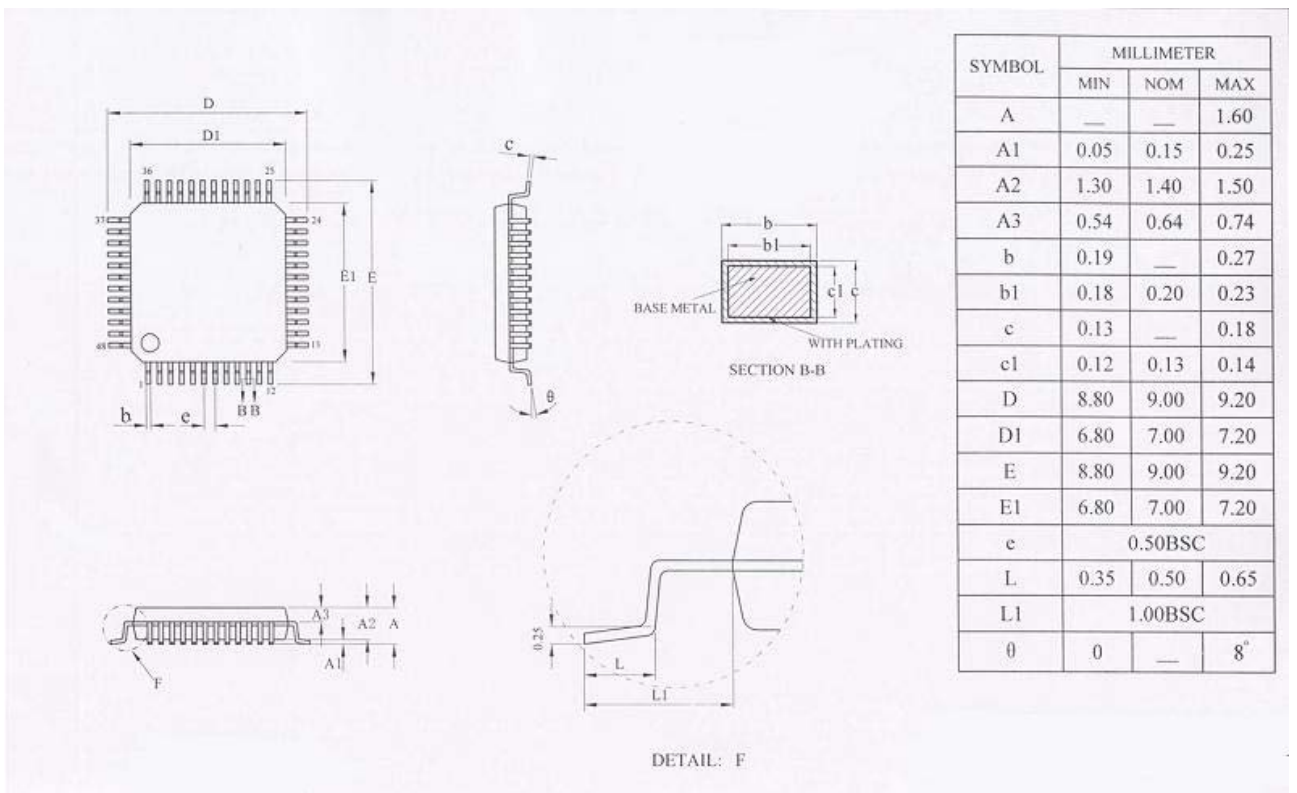
### 7.2.5 ADC 基准电压

( $V_{CC}=AV_{CC}=3.3V$ , 温度范围:  $-40^{\circ}C \sim +85^{\circ}C$ )

测量项目	符号	最小	典型	最大	单位	测试条件
输出电压	$V_{ref}$		1.185		V	
温度系数	$T_c$		$\pm 25$		ppm/ $^{\circ}C$	

## 8 封装

ATT7039: LQFP48, 7mm \*7mm \*1.4mm



## 9 典型应用

(详见 Demo 原理图)