

ATT7039

用户手册

版本更新记录					
文件编号	版本	日期	修订	更改理由	主要更改内容
ATT7039 用户手册	V0.1	2010/06/25	yyuan	初稿	
ATT7039 用户手册	V0.2	2010/07/10	yyuan	更新	1. 更改存储器寻址空间 2. 添加引脚分布图
ATT7039 用户手册	V0.3	2010/07/15	yyuan	添加系统框图；更改 EMUIE 地址	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

目 录

1	总体概况	9
1.1	简介	9
1.2	特点	9
1.2.1	基本特点	9
1.2.2	电能计量	9
1.2.3	处理器及外设	9
1.3	缩略语	10
1.4	系统框图	11
1.5	引脚分布	12
1.6	引脚定义	13
2	ATT7039 构架	15
2.1	概述	15
2.2	存储器	15
2.2.1	概述	15
2.2.2	内部数据存储器	15
2.3	指令系统	17
2.3.1	寻址方式	17
2.3.2	指令集	18
2.4	特殊功能寄存器	22
2.4.1	概述	22
2.4.2	寄存器写保护	23
2.4.3	CPU52 寄存器	24
2.5	模块框图	28
3	存储器控制单元	29
3.1	概述	29
3.2	ATT7039 FLASH特性	29
3.3	ATT7039 存储器映射	30
3.3.1	XRAM	30
3.3.2	Info Flash memory	30
3.3.3	Code Flash Memory	31
3.3.4	Data Flash Memory	31
3.3.5	程序拷贝至XRAM执行的方法	31
3.3.6	第一种映射模式	31
3.3.7	第二种映射模式	32
3.3.8	Flash 的写保护	33
3.3.9	Flash 的读保护	33
3.3.10	寄存器	33
4	电源管理单元	35

4.1	概述	35
4.2	详细功能说明	35
4.2.1	电源切换	35
4.2.2	系统工作电压低于阈值时发生复位	35
4.2.3	电池检测功能	35
4.3	寄存器	36
5	HOLD&SLEEP模式	41
5.1	概述	41
5.2	HOLD模式	41
5.3	SLEEP模式	41
5.4	特殊功能寄存器	42
6	时钟管理单元	46
6.1	概述	46
6.2	时钟系统框图	47
6.3	主要功能	47
6.3.1	低频振荡电路	47
6.3.2	高频振荡电路	47
6.4	系统时钟切换方法	48
6.4.1	概述	48
6.4.2	系统时钟cpu 从低频时钟fosc切换到高频时钟pll	48
6.4.3	系统时钟cpu 从高频时钟pll切换到低频时钟fosc	48
6.4.4	EMU的时钟是独立设置可控的	49
6.4.5	进入Hold模式	49
6.4.6	进入Sleep模式	49
6.4.7	时钟管理单元内部保护机制	49
6.4.8	外设时钟管理	49
6.5	特殊功能寄存器	50
7	时钟输出模块	52
7.1	概述	52
7.2	功能描述	52
7.3	特殊功能寄存器	52
8	系统复位	53
8.1	概述	53
8.2	功能说明	53
8.2.1	上电复位POR (Power on reset)	53
8.2.2	外部引脚复位 /RST (内部上拉30K电阻)	54
8.2.3	掉电复位BOR (Brownout Reset) & LBOR	54
8.2.4	Watchdog 复位WDTR	55
8.2.5	唤醒复位WKR	55
8.2.6	软复位	55

8.3	特殊功能寄存器.....	56
8.4	被分级复位的寄存器说明.....	56
9	WDT	58
9.1	概述	58
9.2	工作模式	58
9.3	特殊功能寄存器.....	58
10	中断系统.....	60
10.1	概述	60
10.2	中断列表	60
10.3	中断优先级	60
10.4	寄存器	61
10.4.1	中断使能寄存器.....	61
10.4.2	中断标志寄存器.....	63
10.4.3	中断优先级寄存器.....	63
10.5	中断处理	64
11	GPIO.....	65
11.1	概述	65
11.2	寄存器说明	65
11.2.1	GPIO输出复用配置寄存器.....	65
11.2.2	P0.....	69
11.2.3	P1.....	69
11.2.4	P2.....	70
11.2.5	P3.....	71
11.2.6	Port A.....	71
11.2.7	Port B.....	72
11.2.8	Port C	72
11.2.9	Port D.....	73
11.2.10	Port E.....	73
12	定时器.....	75
12.1	概述	75
12.2	模块原理框图.....	75
12.2.1	模式 0/1.....	75
12.2.2	模式 2	76
12.2.3	模式 3	77
12.3	定时器 0 详细说明.....	77
12.3.1	模式 0	77
12.3.2	模式 1	78
12.3.3	模式 2	78
12.3.4	模式 3	78
12.4	定时器 1 详细说明.....	79

12.4.1	模式0	79
12.4.2	模式1	79
12.4.3	模式2	79
12.4.4	模式3	80
12.5	定时器2 详细说明	80
12.5.1	概述	80
12.5.2	接口描述	80
12.5.3	模块结构图	80
12.5.4	定时器/计数器功能	80
12.5.5	比较器功能	81
12.5.6	捕获器功能	81
12.6	特殊功能寄存器	82
13	串口	88
13.1	波特率产生方式	88
13.2	串口0	89
13.2.1	模块框图	89
13.2.2	模式1	89
13.2.3	模式2	89
13.2.4	模式3	90
13.2.5	多处理器通讯	91
13.3	串口1	91
13.3.1	模块框图	91
13.3.2	波特率产生方式	91
13.3.3	模式A	91
13.3.4	模式B	92
13.3.5	多处理器通讯	93
13.4	特殊功能寄存器	93
14	红外模块	96
14.1	概述	96
14.2	特殊功能寄存器	96
16	电能计量	97
16.1	概述	97
16.2	功能描述	98
16.2.1	模数转换器	98
16.2.2	ADC采样输出和功率波形输出	98
16.2.3	有功功率、无功功率和视在功率	98
16.2.4	电压、电流有效值	99
16.2.5	电压频率输出	99
16.2.6	灵活的潜动/起动设置	100
16.2.7	功率反向指示	100
16.2.8	防窃电	100

16.2.9	直流偏置自动校正和直流偏置校正寄存器.....	101
16.2.10	能量寄存器和脉冲输出单元.....	101
16.2.11	中断系统.....	102
16.3	寄存器.....	103
16.3.1	特殊功能寄存器.....	103
16.3.2	间接寄存器.....	106
16.4	校表过程.....	125
17	KBI.....	127
17.1	概述.....	127
17.2	按键中断.....	127
17.2.1	按键中断产生.....	127
17.2.2	按键中断清除.....	127
17.3	低功耗模式.....	127
17.3.1	Hold模式.....	127
17.3.2	Sleep模式.....	127
17.4	特殊功能寄存器.....	127
18	LCD.....	130
18.1	概述.....	130
18.2	LCD与GPIO引脚复用.....	130
18.3	原理框图.....	130
18.4	输出波形.....	131
18.5	寄存器.....	134
19	TBS.....	138
19.1	概述.....	138
19.2	功能描述.....	138
19.3	功能详细说明.....	138
19.4	模块工作模式说明.....	139
19.5	测量误差.....	139
19.5.1	IC的温度测量.....	139
19.5.2	IC的电池电压测量.....	139
19.6	特殊功能寄存器.....	140
20	RTC.....	142
20.1	概述.....	142
20.2	功能描述.....	142
20.3	时钟校正.....	142
20.4	时间和万年历.....	143
20.5	寄存器写保护功能与复位源.....	143
20.6	中断功能.....	144
20.7	RTC指示寄存器读取流程.....	145
20.8	特殊功能寄存器.....	146

22	I2C	150
22.1	概述	150
22.2	结构框图	151
22.3	功能描述	151
22.3.1	操作模式	151
22.3.2	串行时钟生成	152
22.3.3	地址比较器	152
22.3.4	中断生成	152
22.4	特殊功能寄存器	152
23	JTAG	162
23.1	概述	162
23.2	介绍	162
24	电气特性	163
24.1	极限参数	163
24.2	DC参数	164
24.3	功耗参数	164
24.4	电能计量参数	165
24.5	ADC指标	166
24.6	ADC基准电压	166

1 总体概况

1.1 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 特点

1.2.1 基本特点

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

1.2.2 电能计量

- 有功电能测量误差小于 0.1%，动态范围大于 1000:1，支持 IEC62053-21、IEC62053-22 标准的精度要求
- 电流有效值测量误差小于 0.5%
- 电压有效值测量误差小于 0.5%
- 两路 ADC，支持单相两线制、单相三线制
- 提供 ADC 同步采样波形数据
- 电流采样通道具有 4 级增益可调，支持分流器和电流互感器直接接入
- 片内基准电压：1.25V ± 2% (温度系数 ± 50ppm/℃)
- 提供有功电能脉冲输出，并开放脉冲计数寄存器
- 提供多种电能累加方式选择
- 支持增益误差、相位误差的软件校表
- 支持防潜动功能，潜动阈值可灵活设置

1.2.3 处理器及外设

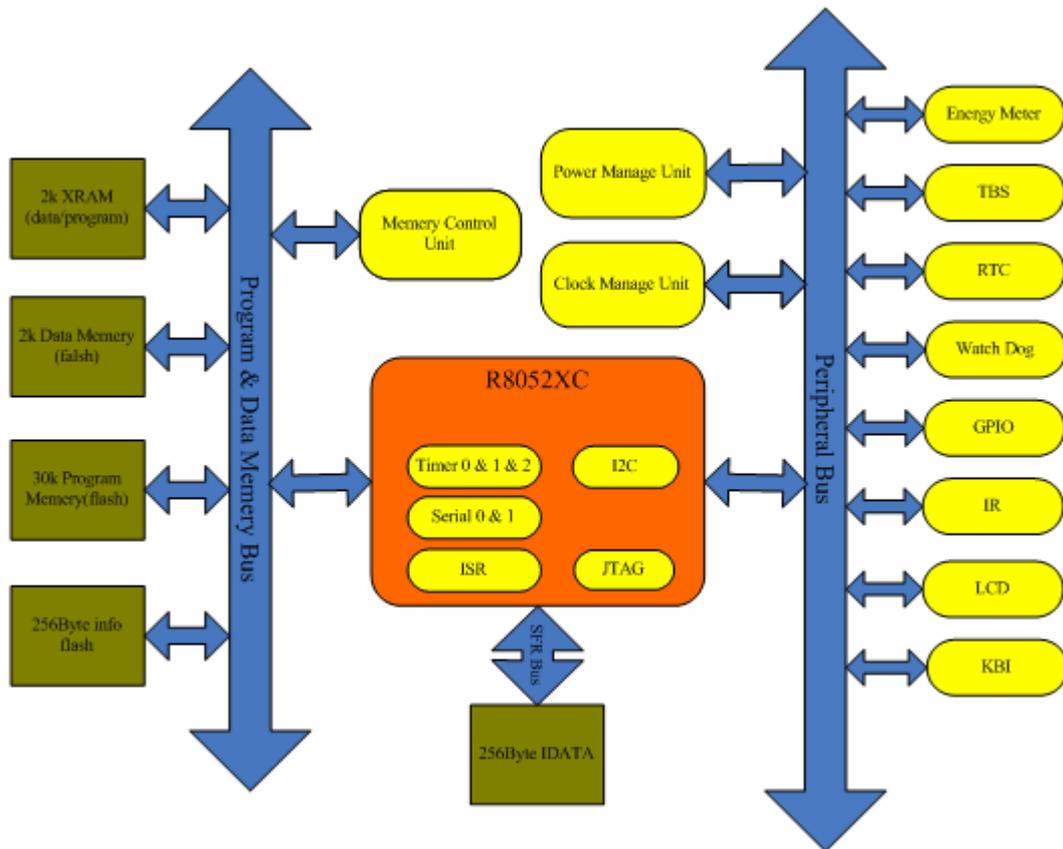
- 基于 8 位 CPU52 设计，具有 8052 兼容指令集和总线结构
- 单 Cycle 的 CPU
- 片内集成 PLL 倍频电路
- 片内集成丰富的存储器资源，包括 30K 的 FLASH 程序存储器，2K 的 Flash 数据存储器，擦除次数大于 20 万次；256 字节有保护操作的 Info FLASH 存储器；256 字节内部数据寄存器；4K 字节外部数据寄存器，其中 256 字节在掉电时，可由后备电源保持数据不丢失
- 具有电源监测功能，电源监测域值可设置，上下电可靠复位
- HOLD 模式下支持外部中断等多种唤醒方式
- SLEEP 模式下支持外部中断等多种唤醒方式
- 片内集成可永不关断的硬件看门狗电路
- 片内集成温度传感器和电池电压检测电路
- 片内集成按键、串行通讯、LCD、I2C 等外设

1.3 缩略语

表 1-1 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 60K 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
TBS	Temperature Battery Sensor	温度, 电池传感器
LVD	Low Voltage Detect	低电压检测
POR	Power On Reset	上电复位
BOR	Brown Out Reset	掉电复位
WKR	Wakeup Reset	唤醒复位
EMU	Energy Mearsurment Unit	电能计量单元
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

1.4 系统框图



1.5 引脚分布

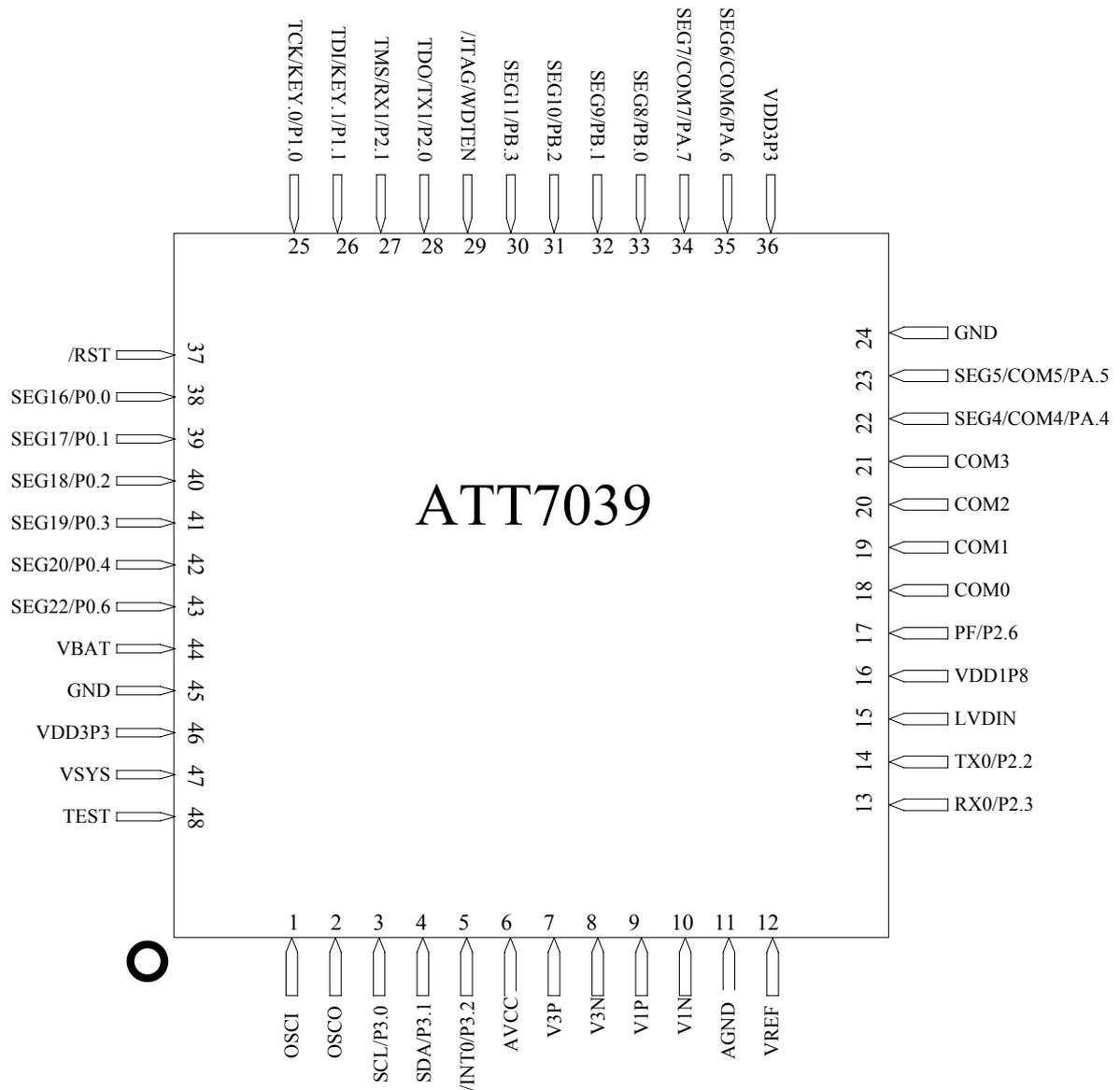


图 1-1 ATT7039 引脚分布图

1.6 引脚定义

表 1-2 ATT7039 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	INOUT	I2C_CLK/GPIO3.0
4	SDA_P31	INOUT	I2C_DATA/GPIO3.1
5	INT0_P32	INOUT	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	1.25V 基准输出端, 外接 10uF 电容
13	RX0_P23	INOUT	RX0/GPIO2.3
14	TX0_P22	INOUT	TX0/GPIO2.2
15	LVDIN	INPUT	VDCIN 检测输入端, 外接分压电阻
16	VDD1P8	POWER	1.8V 电压输出端, 外接 10uF 电容
17	PF_P26	INOUT	PF/GPIO2.6
18	COM0	OUTPUT	COM0
19	COM1	OUTPUT	COM1
20	COM2	OUTPUT	COM2
21	COM3	OUTPUT	COM3
22	SEG4_COM4_PA4	INOUT	SEG4/COM4/ GPIOA.4
23	SEG5_COM5_PA5	INOUT	SEG5/COM5/ GPIOA.5
24	GND	GND	GND
25	VDD3P3	Power	内部 3.3V 电压输出端, 接 10uF 电容
26	SEG6_COM6_PA6	INOUT	SEG6/COM6/ GPIOA.6
27	SEG7_COM7_PA7	INOUT	SEG7/COM7/ GPIOA.7
28	SEG8_PB0	INOUT	SEG8/GPIOB.0
29	SEG9_PB1	INOUT	SEG9/GPIOB.1
30	SEG10_PB2	INOUT	SEG10/GPIOB.2
31	SEG11_PB3	INOUT	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	INOUT	TDO/TX1/GPIO2.0
34	TMS_RX1_P21	INOUT	TDI/RX1/GPIO2.1
35	TDI_KEY1_P11	INOUT	TMS/KEY1/GPIO1.1
36	TCK_KEY0_P10	INOUT	TCK/KEY0/GPIO1.0

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

2 ATT7039 构架

2.1 概述

ATT7039 采用 8052 兼容的 CPU52，具有和 8052 兼容的体系架构。

CPU52 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，CPU52 可通过直接寻址的方式访问这些寄存器，控制系统工作。

2.2 存储器

2.2.1 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

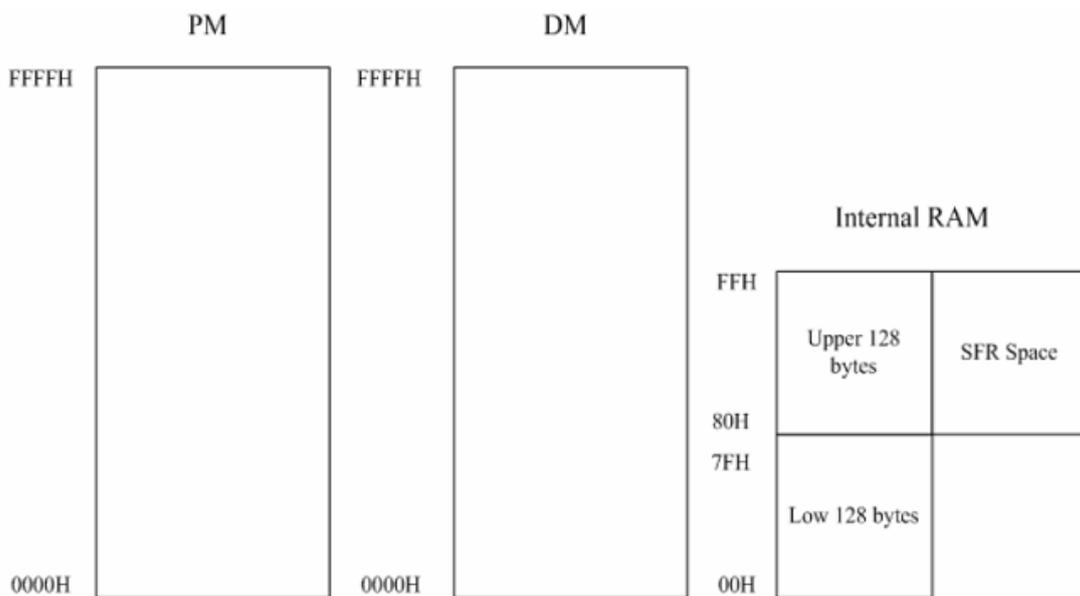


图 2-1 ATT7039 存储器地址空间

程序存储器、内部数据存储器和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 MOVC 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 MOVX 指令访问。

2.2.2 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。

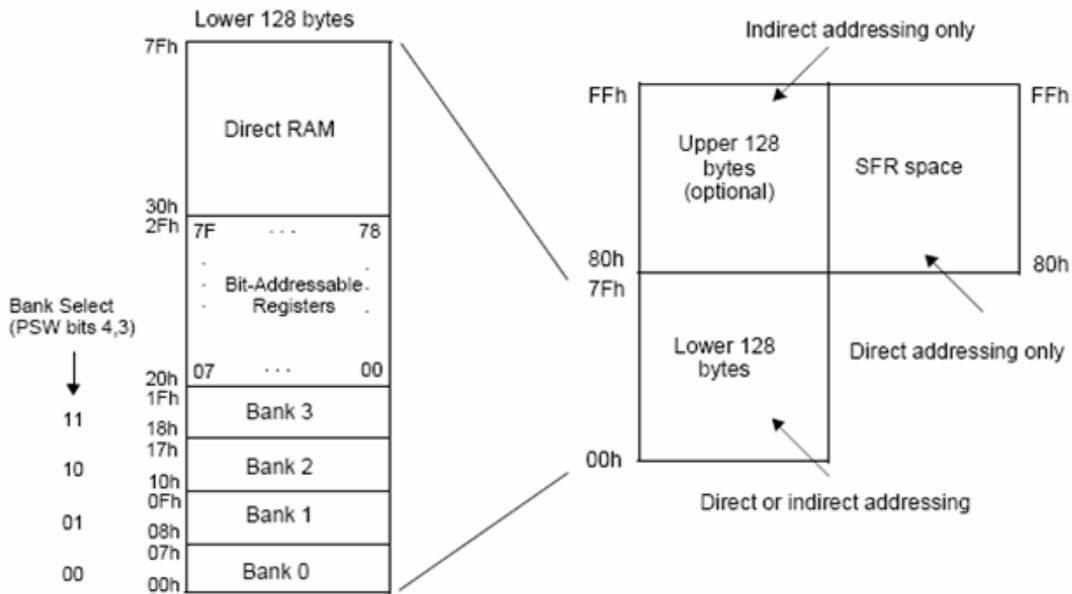


图 2-2 内部 RAM 的功能结构

表 2-1 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。

2.3 指令系统

2.3.1 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
```

2.3.2 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 2-3 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 2-4 ATT7039 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			
算术运算指令								
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	$A \leftarrow (A) + data$	√	√	√	√	2	2	24
ADDC A, Rn	$A \leftarrow (A) + (Rn) + (CY)$	√	√	√	√	1	2	38-3F
ADDC A, direct	$A \leftarrow (A) + (direct) + (CY)$	√	√	√	√	2	3	35
ADDC A, @Ri	$A \leftarrow (A) + ((Ri)) + (CY)$	√	√	√	√	1	4	36-37
ADDC A, #data	$A \leftarrow (A) + data + (CY)$	√	√	√	√	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	$bit \leftarrow 0$	×	×	×		2	4	C2
SETB C	$CY \leftarrow 1$	×	×	×	√	1	1	D3
SETB bit	$bit \leftarrow 1$	×	×	×		2	4	D2
CPL C	$CY \leftarrow \overline{(CY)}$	×	×	×	√	1	1	B3
CPL bit	$bit \leftarrow \overline{(bit)}$	×	×	×		2	4	B2
ANL C, bit	$CY \leftarrow (CY) \wedge (bit)$	×	×	×	√	2	3	82
ANL C, /bit	$CY \leftarrow (CY) \wedge \overline{(bit)}$	×	×	×	√	2	3	B0
ORL C, bit	$CY \leftarrow (CY) \vee (bit)$	×	×	×	√	2	3	72
ORL C, /bit	$CY \leftarrow (CY) \vee \overline{(bit)}$	×	×	×	√	2	3	A0
MOV C, bit	$CY \leftarrow (bit)$	×	×	×	√	2	3	A2
MOV bit, C	$bit \leftarrow (CY)$	×	×	×	×	2	4	92
ACALL addr11	$PC \leftarrow (PC)+2$, $SP \leftarrow (SP)+1$, $(SP) \leftarrow (PC)L$, $SP \leftarrow (SP)+1$, $(SP) \leftarrow (PC)H$, $PC_{10\sim 0} \leftarrow addr11$	×	×	×	×	2	4	11-F1
LCALL addr16	$PC \leftarrow (PC)+2$, $SP \leftarrow (SP)+1$, $(SP) \leftarrow (PC)L$, $SP \leftarrow (SP)+1$, $(SP) \leftarrow (PC)H$, $PC_{10\sim 0} \leftarrow addr16$	×	×	×	×	3	4	12
RET	$(PC)H \leftarrow ((SP))$, $SP \leftarrow (SP)-1$, $(PC)L \leftarrow ((SP))$, $SP \leftarrow (SP)-1$	×	×	×	×	1	5	22
RETI	$(PC)H \leftarrow ((SP))$, $SP \leftarrow (SP)-1$, $(PC)L \leftarrow ((SP))$, $SP \leftarrow (SP)-1$, 从中断中返回	×	×	×	×	2	5	32
AJMP addr11	$PC_{10\sim 0} \leftarrow addr11$	×	×	×	×	3	3	01-E1
LJMP addr16	$PC \leftarrow addr16$	×	×	×	×	2	4	02
SJMP rel	$PC \leftarrow PC+rel$	×	×	×	×	2	3	80
JC rel	$PC \leftarrow PC+2$, 若 $CY=1$, 则 $PC \leftarrow PC+rel$	×	×	×	×	2	3	40
JNC rel	$PC \leftarrow PC+2$, 若 $CY=0$, 则 $PC \leftarrow PC+rel$	×	×	×	×	2	3	50
JB bit, rel	$PC \leftarrow PC+3$, 若 $(bit)=1$, 则 $PC \leftarrow PC+rel$	×	×	×	×	3	3	20
JNB bit, rel	$PC \leftarrow PC+3$, 若 $(bit)=0$, 则 $PC \leftarrow PC+rel$	×	×	×	×	3	3	30
JBC bit, rel	$PC \leftarrow PC+3$, 若 $(bit)=1$, 则 $bit \leftarrow 0$, $PC \leftarrow (PC)+rel$	×	×	×	×	3	3	10
JMP @A+DPTR	$PC \leftarrow (A) + (DPTR)$	×	×	×	×	1	5	73

JZ rel	PC←PC+2 , 若 (A)=0 , PC←(PC)+rel	×	×	×	×	2	5	60
JNZ rel	PC←PC+2, 若 (A) 不等于 0, PC←(PC)+rel	×	×	×	×	2	5	70
CJNE A, direct, rel	PC← PC+3, 若 (A) 不等于 (direct), 则 PC←(PC)+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

2.4 特殊功能寄存器

2.4.1 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 CPU52 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	DDRC	DDRD	DDRP0	DDRP1	DDRP2	D7
C8	T2CON	WDTCON	CRCL	CRCH	TL2	TH2	P02CFG	P3CFG	CF
C0	IRCON	LEDPO	PTE	DDRE	LCDCLK	LDCDR	LCDCFG	KEYCFG	C7
B8	IEN1	IP1	SORELH	SIRELH	PTA	PTB	PTC	PTD	BF
B0	P3	PWMCR	PWMLDR	PWMHDR	RTCIF	EMUIF	KEYIF	WAKEIF	B7
A8	IEN0	IP0	SORELL	PECFG	EMUIE		RTCIE	KEYIE	AF
A0	P2	PLLCFG	PMSR	DIFF_CFG	VBAT_DATA	TEM_DATA	BWPR	RSTSR	A7

98	S0CON	S0BUF	IEN2	S1CON	S1BUF	SIRELL	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

2.4.2 寄存器写保护

表 2-5 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。

2.4.2.1 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。

2.4.2.2 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。

2.4.2.3 配置受保护寄存器示例程序

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;
```

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;
    
```

2.4.3 CPU52 寄存器

表 2-6 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 2-6 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	DPL0	0x00	数据指针寄存器 0 低 8 位
0x83	DPH0	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
0x87	PCON	0x30	波特率选择寄存器
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	0x01	时钟控制寄存器
0x92	DPS	0x00	DPTR 指针选择寄存器
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
0xA8	IEN0	0x00	中断使能寄存器
0xA9	IP0	0x00	中断优先级寄存器
0xAA	S0RELL	0x00	串口 0 重载寄存器低字节
0xB8	IEN1	0x80	中断使能寄存器
0xB9	IP1	0x00	中断优先级寄存器
0xBA	S0RELH	0x00	串口 0 重载寄存器高字节
0xBB	S1RELH	0x00	串口 1 重载寄存器高字节
0xC0	IRCON	0x00	中断请求控制寄存器
0xC8	T2CON	0x00	定时器 2 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
0xD0	PSW	0x00	程序状态字寄存器
0xD8	ADCON	0x40	串口 0 波特率选择寄存器
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 配置寄存器
0xE0	ACC	0x00	累加器
0xF0	B	0x00	B 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 2-7 Program Status Word (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			
PSW.7	CY: 进位标志。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。			
PSW.6	Ac: 辅助进位标志。在进行加法或减法操作而产生低 4 位数 (十进制的数字) 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。			
PSW.5	F0: 标志 0。是用户定义的一个状态标记, 可用软件置位或清零。			
PSW.4	RS1: 工作寄存器区选择控制位 1, 和 RS0 一起用以选择工作寄存器区			
PSW.3	RS0: 工作寄存器区选择控制位 0 <table border="1" style="margin-left: 20px;"> <tr> <td>RS1</td> <td>RS0</td> <td>Bank 选择</td> </tr> </table>	RS1	RS0	Bank 选择
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)	
PSW.2	OV: 溢出标志。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。			
PSW.1	F1: 标志 1, 是用户定义的一个状态标记, 可用软件置位或清零。			
PSW.0	P: 奇偶校验。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。它指示出堆栈顶部在内部 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

为加速数据的块搬移操作, CPU52 使用双数据指针。标准 8052 的数据指针是一个 16 位专用寄存器, 其高位字节寄存器用 DP_H 表示, 低位字节用 DP_L 表示, DPTR 主要用来存放 16 位地址, 当对外部数据存储单元寻址时, 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 DPTR0, 它位于 SFR 82H (DPL0) 和 83H (DPH0), 默认情况下, 数据指针使用 DPTR0。除此之外, 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 DPTR0 还是 DPTR1, 当 SEL=0, 使用 DPTR 的指令的数据指针用 DPL0 和 DPH0 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 和 UART1, 串行数据缓冲区 SBUF0 和 SBUF1 用于存放 UART0 和 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 2-8 Clock controller register (CKCON, 0x8EH)

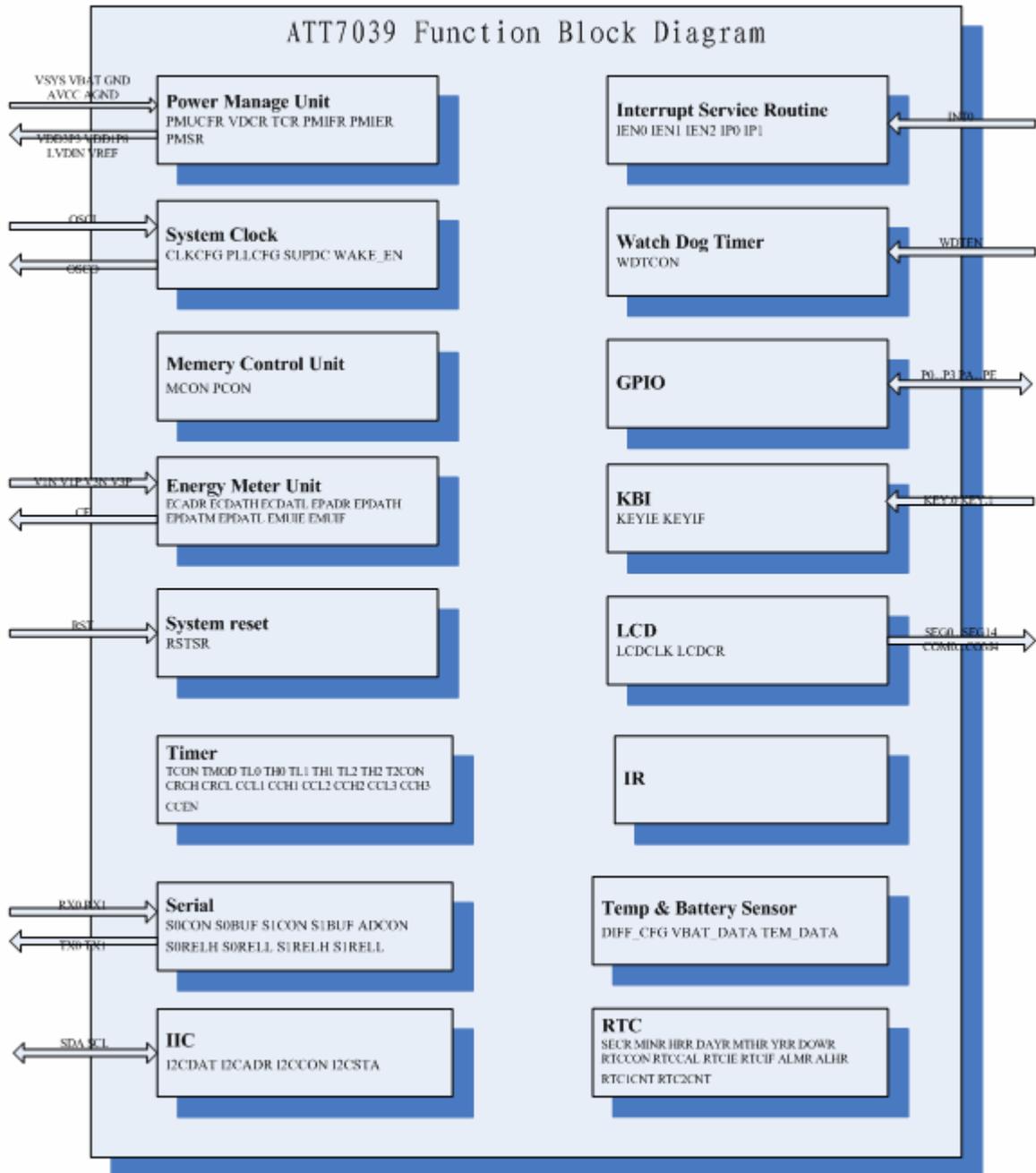
Clock controller register (CKCON)		Address: 8EH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	CKCON.7	CKCON.6	CKCON.5	CKCON.4	CKCON.3	CKCON.2	CKCON.1	CKCON.0

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

Table 39. CKCON register

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	-		

2.5 模块框图



各功能模块内标注有模块用到的 SFR 寄存器；
箭头表示与模块相关的引脚。

3 存储器控制单元

3.1 概述

ATT7039内置8bit可编程高可靠性32K bytes Flash Memory和2K XRAM。

Flash被擦除后的状态为0xFF，Flash在逻辑空间由3部分组成，30K Code Flash，2K Data Flash 和一个256 bytes info flash。

- 30K Code Flash用于用户存储程序，在Program Memory空间地址分配为0000H—77FFH。
- 2K 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—17FFH。
- 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—20FFH。

2K 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—0x07FF) 当芯片进入SLEEP模式后数据丢失。

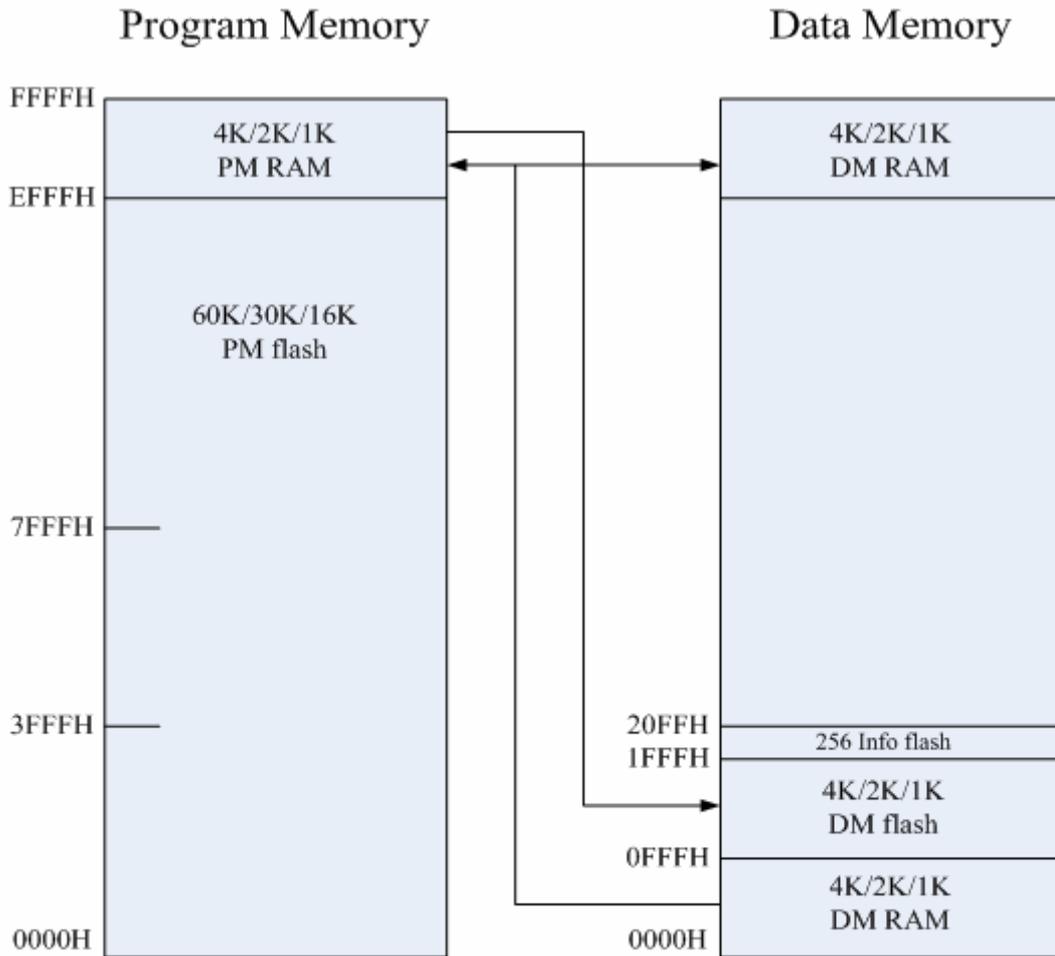
处于Data Memory 空间的XRAM可被映射到Program Memory空间的F000H—F7FFH地址范围 (由MCON寄存器的RAMMAP位控制)，通过此功能，客户可以在XRAM中执行用户程序。

存储空间分为两种映射模式，第一种映射模式的存在是为了与ATT7025兼容，第二种映射模式是ATT7039新模式。推荐使用第二种模式。

3.2 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)

3.3 ATT7039 存储器映射



3 configuration:

60K PM flash + 4K DM flash + 0.256K Info flash + 4K RAM

30K PM flash + 2K DM flash + 0.256K Info flash + 2K RAM

16K PM flash + 1K DM flash + 0.256K Info flash + 1K RAM

图3-1 UAM下PM和DM存储器映射 (ATT7039为30k PM flash的配置)

3.3.1 XRAM

XRAM 地址映射可以通过 RAMMAP 进行选择。

当 RAMMAP=0 时, RAM 地址空间为 DM 0000-07FFH, RAM 仅作为数据 RAM。而当 RAMMAP=1 时, RAM 地址空间为 F000H-F7FFH, 此时 RAM 既可以作为数据 RAM, 也可以作为程序 RAM, 也就是说地址空间 F000H-F7FFH 是一个数据与程序共用的地址空间。

3.3.2 Info Flash memory

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

3.3.3 Code Flash Memory

这部分 30K Flash 是用于存储程序，占用 PM 地址空间：0000H-7FFFH。分为 30 页，每页 1KB。

3.3.4 Data Flash Memory

数据存储器占用地址 1000H-17FFFH，采用 MOVX 指令访问。分为 2 页，每页 1KB。

3.3.5 程序拷贝至 XRAM 执行的方法

- 设置寄存器 MCON 的 RAMMAP 将 XRAM 作为 PM 使用，RAMMAP = 1，地址空间为 EFFFH~F7FFFH。
- 设置寄存器 MCON 的 MAPMOD 选择第一种映射模式，MAPMOD = 0
- 将需要拷贝的程序写入 F000H 起始的 2K PM (XRAM) 空间。
- 程序跳转到程序入口地址执行拷贝的程序。

3.3.6 第一种映射模式

3.3.6.1 flash whole chip 擦除操作

这种映射模式需要将以下操作 flash 的程序 copy 到 xram 中执行：

- 设置寄存器 MCON 使能 Flash 擦除和写入，PMLOCK = 1，RSLOCK = 1
- 设置寄存器 MCON 的 FOP1 和 FOP0 选择对 Flash whole chip 擦除，FOP[1:0]=11
- 设置寄存器 MCON 的 MAPMOD 选择第一种映射模式，MAPMOD = 0
- 将要擦除的 Flash 地址写入 DPTR，将要 flash 的数据（任意）送写入累加器 A，擦除 flash，等待 flash 擦除完成。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
```

说明：DPTR 地址落在 2000H——20FFFH 之间时，整个 flash 擦除

DPTR 地址落在 1000H——17FFFH 之间时，对 2K DM flash 操作

DPTR 地址落在 0000H——07FFFH 之间时，对 2K DM RAM 操作

3.3.6.2 flash 页擦除操作

这种映射模式需要将以下操作 flash 的程序 copy 到 xram 中执行：

- 设置寄存器 PCON 的 PMW 位选择对 PM 的 Flash 操作，PMW = 1（对 DM 或者 INFO 操作时，需要 PMW=0）
- 设置寄存器 MCON 使能 Flash 擦除和写入，PMLOCK = 1，RSLOCK = 1
- 设置寄存器 MCON 的 FOP1 和 FOP0 选择对 Flash 的页擦除操作，FOP[1:0]=10
- 设置寄存器 MCON 的 MAPMOD 选择第一种映射模式，MAPMOD = 0
- 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
```

3.3.6.3 flash 写操作

这种映射模式需要将以下操作 flash 的程序 copy 到 xram 中执行:

- 设置寄存器PCON的PMW位选择对PM的Flash操作, PMW = 1 (对DM或者**INFO**操作时, 需要PMW=0)
- 设置寄存器MCON使能 Flash擦除和写入, **PMLOCK = 1, RSLOCK = 1**
- 设置寄存器MCON的FOP1和FOP0选择对Flash的写操作, FOP[1:0]=01
- 设置寄存器MCON的MAPMOD选择第一种映射模式, MAPMOD = 0
- 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
```

3.3.6.4 flash 读操作

- 设置寄存器PCON的PMW位选择对PM的Flash操作, PMW = 1 (对DM或者**INFO**操作时, 需要PMW=0)
- 设置寄存器MCON的MAPMOD选择第一种映射模式, MAPMOD = 0
- 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0x2011的值到累加器A
```

3.3.7 第二种映射模式

3.3.7.1 flash whole chip 擦除操作

这种映射模式是在 flash 中运行程序对 flash 进行操作, 因此不能进行 whole chip 擦除操作, 否则操作 flash 的代码也被擦除。

3.3.7.2 flash 页擦除操作

这种映射模式可以在 flash 中运行程序来操作 flash, 操作方法如下:

- 设置寄存器PCON的PMW位选择对PM的Flash操作, PMW = 1 (对DM或者**INFO**操作时, 需要PMW=0)
- 设置寄存器MCON使能 Flash擦除和写入, **PMLOCK = 1, RSLOCK = 1**
- 设置寄存器MCON的FOP1和FOP0选择对Flash的页擦除操作, FOP[1:0]=10
- 设置寄存器MCON的MAPMOD选择第二种映射模式, MAPMOD = 1
- 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
```

3.3.7.3 flash 写操作

这种映射模式可以在 flash 中运行程序来操作 flash, 操作方法如下:

- 设置寄存器PCON的PMW位选择对PM的Flash操作，PMW = 1（对DM或者INFO操作时，需要PMW=0）
- 设置寄存器MCON使能 Flash擦除和写入，**PMLOCK = 1**， **RSLOCK = 1**
- 设置寄存器MCON的FOP1和FOP0选择对Flash的写操作，FOP[1:0]=01
- 设置寄存器MCON的MAPMOD选择第二种映射模式，MAPMOD = 1
- 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
    
```

3.3.7.4 flash 读操作

- 设置寄存器PCON的PMW位选择对PM的Flash操作，PMW = 1（对DM或者INFO操作时，需要PMW=0）
- 设置寄存器MCON的MAPMOD选择第二种映射模式，MAPMOD = 1。
- 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
    
```

3.3.8 Flash 的写保护

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

3.3.9 Flash 的读保护

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

3.3.10 寄存器

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

PMLOCK:

PMLOCK=1: 在 UAM 模式下，使能所有 Flash 的擦除/写操作。

PMLOCK=0: 在 UAM 模式下，PM Flash 处于只读模式，不能够被修改。

RSLOCK:

RSLOCK=1: 在 UAM 模式下，如果 PMLOCK=1，那么使能前 4K 的 PM Flash 的擦除/写操作。

RSLOCK=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。

FOP[1:0]:

FOP[1:0]=00: 处于 Flash 只读模式。

- FOP[1:0]=01: MOVX 将执行 Flash 写操作。
- FOP[1:0]=10: MOVX 将执行 Flash Page 擦除操作。
- FOP[1:0]=11: MOVX 将执行 Flash Whole Chip 擦除操作。

BUSY:

- BUSY=1: 表示 Flash 正在进行写/擦除操作
- BUSY=0: 表示 Flash 空闲, 可以进行操作

RAMMAP:

- RAMMAP=1: RAM 作为 Program Memory 使用, 地址空间为 EFFFH~F7FFH。
- RAMMAP=0: RAM 作为 Data Memory 使用, 地址空间为 0000H~07FFH。

MAPMOD:

- MAPMOD=1: 采用第二种映射方式。
- MAPMOD=0: 采用第一种映射方式。

WRFAIL:

- WRFAIL=1: 表示 Flash 在系统时钟切换时的写/擦除操作失败
- WRFAIL=0: 表示 Flash 的写/擦除操作正常

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

PMW:

- PMW=0: 使能数据存储器的写操作, 关闭程序存储器的写操作。
- PMW=1: 使能程序存储器的写操作, 关闭数据存储器的写操作。

4 电源管理单元

4.1 概述

PMU 为芯片的电源管理部分，主要有如下功能：

- 实时监测外部供电电源电压 (VSYS 和 VDC_IN)，当外部供电电压低 (高) 于设定阈值时，产生相应的中断信号。
- 通过 VSYS_DET 检测外部供电的 VSYS 电压，自动完成系统电源和电池电源的切换(电池 VBAT<->主电源 VSYS)，同时更新 PMSR 寄存中的状态位 VSYSS，当 VSYSS=0 时，表明当前系统由 VSYS 供电，当 VSYSS=1，表明当前系统由 VBAT 供电。
- 定量监测电池电压(VBAT)，当电池电源低于设定阈值时，产生相应的中断信号。
- 用独立的 BOR 模块监测芯片内部的工作电源 VDD，并产生相应的中断和状态信号。
- 对芯片内部的工作电压 VDD 进行监测，产生 BOR、POR、LBOR 复位信号。
- 控制 1.8V 逻辑供电电源的开关。

4.2 详细功能说明

4.2.1 电源切换

4.2.1.1 电源自动切换 (电池 VBAT<->主电源 VSYS)：

电源的切换为硬件切换，由VSYS_DET模块检测外部供电的VSYS电压，根据VSYS电压的状态，自动完成系统电源VSYS和电池电源VBAT的切换。在系统上电后，VSYS_DET检测功能是默认有效的。

4.2.1.2 能够指示当前供电状态

寄存器 PMSR 的 VSYSS 位给出 VDD 所连接的电源的状态。

4.2.1.3 电源切换时能够给出中断

电源切换的时候能够给出 PMU 中断。

4.2.2 系统工作电压低于阈值时发生复位

- POR：上电复位
- LBOR：掉电复位
- BOR：用户可以改变复位阈值，低于阈值后发生复位。
- BOR 复位功能
- BOR 中断功能

4.2.3 电池检测功能

通过内部的ADC对VBAT的电压进行检测来测试电池的使用情况，能够给出电池电压的具体的值。(具体见TBS部分)

4.3 寄存器

表 4-1 PMU 相关寄存器列表

地址	名称	复位值	功能描述
93H	PMUCFR	0x67H	PMU配置寄存器
94H	VDCR	0x78H	电压检测控制寄存器
95H	TCR	0x00H	测试控制寄存器
96H	PMIFR	0x00H	PMU中断标志寄存器
97H	PMIER	0x00H	PMU中断控制寄存器
A2H	PMSR	0x00H	PMU电源状态指示寄存器

表 4-2 PMU 配置寄存器 PMUCFR (93H)

PMUCFR Address: 93H								
	Bit7	6	5	4	3	2	1	Bit0
Read	X	TBSB_EN	TBST_EN	SEL_FTBS	TLVREF_EN	VDCIN_EN	BORRST	BOR_EN
Write	X							
Reset	0	1	1	0	0	1	1	1

位	功能描述
TBSB_EN	电池电压测量使能信号高电平有效 1: 测量电池电压。 0: 不测量电池电压。
TBST_EN	温度测量使能信号,高电平有效 1: 测量温度。 0: 不测量温度。
SEL_FTBS	TBS分时开启频率选择信号 1: ADC 分时打开, 每 0.5s 打开 0.5ms。 0: ADC 分时打开, 每 1s 打开 0.5ms。
TLVREF_EN	TLVREF_EN (LVREF 电压输出 TEST 控制信号) 1: LVREFO PIN 输出 LVREF 的电压, 用来测试。 0: LVREFO PIN 不能输出 LVREF 的电压。
VDCIN_EN	VDCIN_DET 整体使能控制信号 1: VDCIN_DET 模块开启 0: VDCIN_DET 模块关闭
BORRST	BORRST: BOR复位和中断选择位。 1: 表示VDD电压低于设定模式的Vbor1电压值, 将会立即产生BOR复位; 0: VDD电压下降降到低于设定阈值时或上升到高于设定阈值时产生BOR中断。
BOR_EN	BOR 整体使能控制信号 1: BOR 模块开启

0: BOR 模块关闭

表 4-3 VDCR (Vbat threshold register) (94H)

Vbat threshold register (VDCR)			Address: 94H					
	Bit7	6	5	4	3	2	1	Bit0
Read	X	lvref_ctl2	lvref_ctl1	lvref_ctl0	C_VSYS1	C_VSYS0	C_BOR1	C_BOR0
Write								
Reset:	0	1	1	1	1	0	0	0

注意: 该寄存器只能被POR, LBOR复位。

位	功能描述																																				
lvref_ctl[2:0]	基准调节信号与基准电压的对应表 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>LVREF_CTL2</th> <th>LVREF_CTL1</th> <th>LVREF_CTL0</th> <th>VO_LVREF</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>+53MV</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>+35MV</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>+18MV</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>+1MV</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>DEFAULT</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>-45MV</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>-31MV</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>-16MV</td></tr> </tbody> </table>	LVREF_CTL2	LVREF_CTL1	LVREF_CTL0	VO_LVREF	0	0	0	+53MV	0	1	0	+35MV	1	0	0	+18MV	1	1	0	+1MV	1	1	1	DEFAULT	0	0	1	-45MV	0	1	1	-31MV	1	0	1	-16MV
LVREF_CTL2	LVREF_CTL1	LVREF_CTL0	VO_LVREF																																		
0	0	0	+53MV																																		
0	1	0	+35MV																																		
1	0	0	+18MV																																		
1	1	0	+1MV																																		
1	1	1	DEFAULT																																		
0	0	1	-45MV																																		
0	1	1	-31MV																																		
1	0	1	-16MV																																		
C_VSYS[1:0]	C_VSYS0~C_VSYS1 为系统电压 VSYS 检测电路 4 档阈值的调节信号, 它们与阈值的对应关系: <table border="1" style="margin-left: 20px;"> <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</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	1	1	3.0																					
C_VSYS1	C_VSYS0	VTH_VSYS (V)																																			
0	0	2.4																																			
0	1	2.6																																			
1	0	2.8																																			
1	1	3.0																																			
C_BOR[1:0]	C_BOR [1:0]: BOR检测电压设置位。当VDD<BOR检测电压时, 产生BOR复位或者BOR中断。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>C_BOR1</th> <th>C_BOR0</th> <th>BOR检测电压</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>2.2 v模式</td></tr> <tr><td>0</td><td>1</td><td>2.4 v模式</td></tr> <tr><td>1</td><td>0</td><td>2.6 v模式</td></tr> <tr><td>1</td><td>1</td><td>2.8 v模式</td></tr> </tbody> </table>	C_BOR1	C_BOR0	BOR检测电压	0	0	2.2 v模式	0	1	2.4 v模式	1	0	2.6 v模式	1	1	2.8 v模式																					
C_BOR1	C_BOR0	BOR检测电压																																			
0	0	2.2 v模式																																			
0	1	2.4 v模式																																			
1	0	2.6 v模式																																			
1	1	2.8 v模式																																			

表 4-4 TCR (Lvref threshold register) (95H)

Lvref threshold register(TCR)		Address: 95H						
	Bit7	6	5	4	3	2	1	Bit0
Read	BOR_DIV	Stdby	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	在 Sleep mode 下 BOR 分时开启的控制位 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 这两个位来决定)。 0: 正常模式															
Stdby	在 hold mode 下 VREG 低功耗控制位 1: 低功耗模式 0: 正常模式 (只能在 Hold 下才能使能)															
Sel_AorD	数字滤波器和模拟滤波器选择 1: 选择数字滤波器 0: 选择模拟滤波器															
TO_VSYS[1:0]	TO_VSYS0~TO_VSYS1 用于在 sleep 下 VSYS_DET 分时检测时每个周期内 VSYS_DET 工作的时间: <table border="1" data-bbox="454 1102 1131 1473"> <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> <tr> <td>1</td> <td>1</td> <td>2100</td> </tr> </tbody> </table>	TO_VSYS1	TO_VSYS0	T(US)	0	0	524	0	1	1050	1	0	1570	1	1	2100
TO_VSYS1	TO_VSYS0	T(US)														
0	0	524														
0	1	1050														
1	0	1570														
1	1	2100														
T_VSYS[1:0]	T_VSYS0~T_VSYS1 用于在 sleep 下 VSYS_DET 分时检测的周期: <table border="1" data-bbox="454 1518 1131 1890"> <thead> <tr> <th>T_VSYS1</th> <th>T_VSYS0</th> <th>T(MS)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>33</td> </tr> <tr> <td>0</td> <td>1</td> <td>67</td> </tr> <tr> <td>1</td> <td>0</td> <td>134</td> </tr> <tr> <td>1</td> <td>1</td> <td>268</td> </tr> </tbody> </table>	T_VSYS1	T_VSYS0	T(MS)	0	0	33	0	1	67	1	0	134	1	1	268
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:

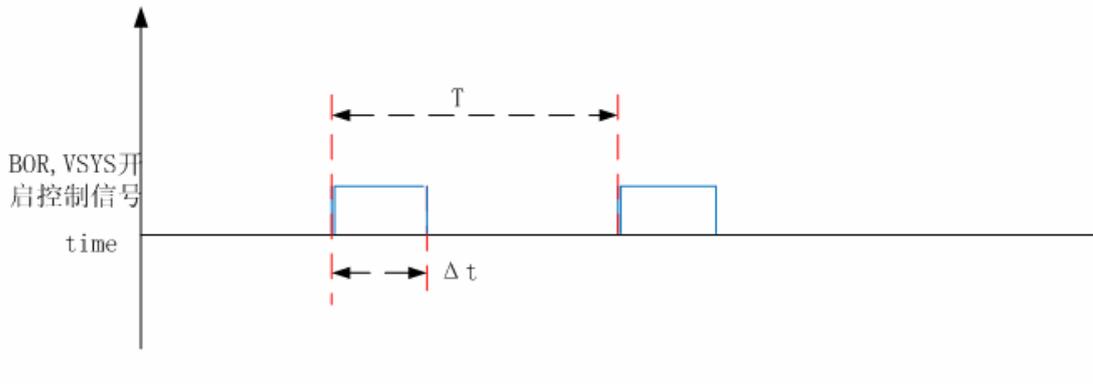


图 4-4 VSYS_DET, BOR_DET 分时检测时序图

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 Δt 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 Δt 的时间如上图所示。

表 4-5 PMIER PMU 中断使能寄存器(97H)

Power management interrupt enable register (PMIER)			Address: 97H					
	Bit7	6	5	4	3	2	1	Bit0
Read	VBATIE	TMPIE	X	X	x	BORIE	VSYSIE	VDCINIE
Write			0	0	0	0	0	0
Reset:	0	0	0	0	0	0	0	0

位	功能描述
VBATIE	VBATIE: Battery 电压测量中断使能信号 1: 允许Battery 电压测量中断 0: 不允许 Battery 电压测量中断
TMPIE	TMPIE: 温度测量中断使能信号 1: 允许温度测量中断 0: 不允许温度测量中断
BORIE	BOR中断使能位 1: enable BOR中断 0: disable BOR 中断
VSYSIE	VSYSDE, VSYS中断使能位 1: enable VSYS_monitor_INTR 0: disable VSYS_monitor_INTR
VDCINIE	VDCINDE, VDC_IN中断使能位 1: enable VDCIN_monitor_INTR 0: disable VDCIN_monitor_INTR

表 4-6 PMIFR PMU 中断标志寄存器(96H)

Power management interrupt Flag register (PMIFR)			Address: 96H					
	Bit7	6	5	4	3	2	1	Bit0
Read	VBATIF	TMPIF			x	BORIF	VSYISIF	VDCINIF
Write			X	X				
Reset:	0	0				0	0	0

注意：该寄存器不能被 Wake_UP 唤醒 RESET。

位	功能描述
VBATIF	VBATIF: Battery 电压测量中断源状态信号 1: 有Battery 电压测量中断标志位 0: 没有 Battery 电压测量中断标志位 该位写 0 清中断
TMPIF	TMPIF: 温度测量中断源状态信号 1: 有温度测量中断标志位 0: 没有温度测量中断标志位 该位写 0 清中断
BORIF	BOR中断源位 1: 系统产生BOR中断源 0: 无 BOR 中断源 该位写 0 清 0
VSYISIF	VSYISIF, VSYISIF_monitor_intr中断源位 1: 当VSYISIF输入电压下降到低于设定阈值时或上升到高于设定阈值时, VSYISIF置 1 0: 无VSYISIF_monitor_intr 该位写 0 清 0
VDCINIF	VDCINIF, VDCINIF_monitor_intr中断源位 1: 当VDCINIF输入电压下降到低于设定阈值时或上升到高于设定阈值时, VDCINIF置 1 0: 无 VDCINIF_monitor_intr 该位写 0 清 0

表 4-7 PMSR PMU 状态寄存器寄存器(A2H)

Power management Status Flag register (PMSR)			Address: A2H					
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	X	X	BORS	VSYSS	VDCINS
Write						X	X	X
Reset:	0	0	0	0	0			

注意：该寄存器不能被 Wake_UP 唤醒 RESET。

位	功能描述
BORS	Bor status (read only) 1: VDD 小于设定阈值 0: VDD 大于设定阈值
VSYS	系统电源状态 (read only) 1: VSYS 小于设定阈值 (VDD=VBAT) 0: VSYS 大于设定阈值 (VDD=VSYS)
VDCINS	Vdcin status (read only) 1: VDCIN 小于设定阈值 0: VDCIN 大于设定阈值

补充：在系统由 VSYS 供电时，不允许对 CMU 单元的 CLKCFG 进行配置进入 SLEEP MODE，如果在 VSYS 供电时，CLKCFG 进行配置进入 SLEEP MODE，就会出现 Soft Reset。

5 Hold&Sleep 模式

5.1 概述

为了更好地对芯片的功耗进行管理，ATT7039设置了HOLD，SLEEP两种节省功耗的模式。

5.2 Hold 模式

通过对写保护的CLKCFG寄存器进行操作，控制CPU进入Hold Mode。在Hold模式下，保持所有CPU的功能。Hold模式下CPU相连接的外设功能模块功能保持有效，只是CPU进入Standby状态，如果使能中断，CPU可以从中断中唤醒，开始从中断的ISR执行程序。

在进入HOLD模式后，各种外设模块保持进入HOLD模式前的状态。为了节省功耗。可以设置寄存器PMSR的Stdby=1，关闭产生1.8V的模块进入低功耗模式。

5.3 Sleep 模式

通过对写保护的CLKCFG寄存器进行操作，控制CPU进入Sleep Mode。进入SLEEP MODE后，为了节省功耗，关闭产生1.8V的模块。

在ATT7039进入Sleep后，可以被int0_n, int1_n, RX0, RX1, PMU, RTC, KEY这7个唤醒源唤醒。CPU从Sleep状态下唤醒，CPU不会执行中断服务程序。CPU从Sleep下唤醒等同复位，程序从复位地址0000H开始执行。只有在VBAT供电时，才能进入Sleep模式。

在SLEEP MODE 下：

Reset复位信号,是不可屏蔽的。(包括POR,BOR,LBOR,外部RESET PIN上产生的external Reset信号,以及内部的WDT复位)信号。当ATT7039进入SLEEP Mode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。

WDT唤醒复位信号,通过在SLEEP前写入WDT_SH=1,进入SLEEP后,WDT依然有效。WDT计数溢出时,系统发生WDT复位。通过在SLEEP前写入WDT_SH=0,进入SLEEP后,WDT无效(详细见WDT的SPEC章节)。

int0_n, int1_n, RX0, RX1, RTC, KEY, PMU/TBS唤醒源的使能信号,都可以利用软件单独控制。

5.4 特殊功能寄存器

表 5-1 Hold/Sleep SFR 寄存器列表

地址	名称	字节长度	功能描述
0x95	TCR	1	
0x9E	CLKCFG	1	
0x91	WAKE_EN	1	
0xA7	RSTSR	1	

表 5-2 TCR (Lvref threshold register) (95H)

Lvref threshold register (TCR)		Address: 95H						
	Bit7	6	5	4	3	2	1	Bit0
Read	BOR_DIV	Stdby	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	在 Sleep mode 下 BOR 分时开启的控制位 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: 正常模式															
Stdby	在 hold mode 下低功耗控制位(只能在 Hold 下才能使能) 1: 低功耗模式 0: 正常模式															
TO_VSYS[1:0]	TO_VSYS0~TO_VSYS1 用于在 sleep 下 VSYS_DET 分时检测时每个周期内 VSYS_DET 工作的时间: <table style="margin-left: auto; margin-right: auto;"> <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> <tr> <td>1</td> <td>1</td> <td>2100</td> </tr> </tbody> </table>	TO_VSYS1	TO_VSYS0	T (US)	0	0	524	0	1	1050	1	0	1570	1	1	2100
TO_VSYS1	TO_VSYS0	T (US)														
0	0	524														
0	1	1050														
1	0	1570														
1	1	2100														

T_VSYS[1:0]	T_VSYS0~T_VSYS1 用于在 sleep 下 VSYS_DET 分时检测的周期:															
	<table border="1"> <thead> <tr> <th>T_VSYS1</th> <th>T_VSYS0</th> <th>T (MS)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>33</td> </tr> <tr> <td>0</td> <td>1</td> <td>67</td> </tr> <tr> <td>1</td> <td>0</td> <td>134</td> </tr> <tr> <td>1</td> <td>1</td> <td>268</td> </tr> </tbody> </table>	T_VSYS1	T_VSYS0	T (MS)	0	0	33	0	1	67	1	0	134	1	1	268
T_VSYS1	T_VSYS0	T (MS)														
0	0	33														
0	1	67														
1	0	134														
1	1	268														

表 5-3 CLKCFG 时钟配置寄存器(9EH)

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

位	功能描述									
W_EN	<p>w_en位: 时钟配置寄存器模式写保护位</p> <p>如果用户要改系统的模式, 在向mode1和mode0 位写入新的值时, 必须同时将w_en位置1, 例如8 ‘b1xxxx10, 此时改变模式到hold, 该寄存器的其他位不会受到该写操作的影响; w_en位置0时, 可对该寄存器他位操作。(对该寄存器的写入要采取先打开写保护寄存器)</p>									
Mode[1:0]	<p>表 Mode[1:0]和工作模式对应表</p> <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>Mode</th> <th>Mode 1</th> <th>Mode 0</th> </tr> </thead> <tbody> <tr> <td>Hold</td> <td>1</td> <td>0</td> </tr> <tr> <td>Sleep</td> <td>1</td> <td>1</td> </tr> </tbody> </table> <p>(对该寄存器的写入要采取先打开写保护寄存器, 再写入 ‘b1xxx10xx), 才能设置系统进入Hold或者Sleep的节电模式。</p> <p>1. Mode[1..0]Default值为[00]。如果W_EN使能, 对该位置写[00], [01]系统产生Soft Reset。(如果在上电没有稳定的时候出现W_EN=1. Mode[1..0]=00的情况, 会造成系统的Soft Reset)。</p> <p>2. 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]进入SLEEP MODE, 就会出现Soft Reset。</p>	Mode	Mode 1	Mode 0	Hold	1	0	Sleep	1	1
Mode	Mode 1	Mode 0								
Hold	1	0								
Sleep	1	1								

表 5-4 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	INT1_EN	RX0_EN	RX1_EN	RTC_EN	KEY_EN	PMU/TBS_EN	X
Write:								
Reset:	0	0	0	0	0	0	0	0

注意: 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。

每一个 BIT 对应一个唤醒复位源，“0”代表该信号源在 Sleep 下的唤醒功能无效，“1”代表该信号源在 Sleep 下的唤醒功能有效。

INT0_EN=1, 表示 int0 信号唤醒功能有效。INT0_EN=0, 表示 int0 信号唤醒功能无效。

INT1_EN=1, 表示 int1 信号唤醒功能有效。INT1_EN=0, 表示 int1 信号唤醒功能无效。

RX0_EN=1, 表示 rx0 信号唤醒功能有效。RX0_EN=0, 表示 rx0 信号唤醒功能无效。

RX1_EN=1, 表示 rx1 信号唤醒功能有效。RX1_EN=0, 表示 rx1 信号唤醒功能无效。

RTC_EN=1, 表示 RTC 中断信号唤醒功能有效。RTC_EN=0, 表示 RTC 中断信号唤醒功能无效。

KEY_EN=1, 表示 KEY 信号唤醒功能有效。KEY_EN=0, 表示 KEY 信号唤醒功能无效。

PMU/TBS_EN=1, 表示 PMU/TBS 中断信号唤醒功能有效。PMU_EN=0, 表示 PMU/TBS 中断信号唤醒功能无效。

注意:

Sleep 下唤醒信号源: int0_n, int1_n, RX0, RX1, RTC, KEY, PMU/TBS。

要实现在 Sleep 下的唤醒功能, 进入 Sleep 前需进行以下功能配置的操作:

(1) (int0_n, int1_n, RX0, RX1)唤醒源的使能, 相应的 PIN 要配置为 int0, int1, RX0, RX1 功能 PIN (详见 GPIO 单元)。相应 PIN 上出现下降沿并保持不少于 4 个 fosc 的低电平时, 可以让 CPU 从 SLEEP 下唤醒。

(2) 配置相应的功能 PIN 为 KEY 功能, 并配置相应的在功能 PIN 的 KEY 功能上, 使能相应的 KEY 功能中断, 同时使能配置 WAKE_EN 中的 KEY_EN=1, 配置 KEY 唤醒使能。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) 设置 PMU/TBS 相应的功能有效和中断功能有效, 同时使能配置 WAKE_EN 中的 PMU/TBS_EN=1, 配置 PMU/TBS 唤醒使能。

(4) RTC 中对应中断源使能, 同时使能配置 WAKE_EN 中的 RTC_EN=1, 配置 RTC 唤醒使能。

CPU 从 Sleep 状态下唤醒, CPU 不会执行中断服务程序。CPU 从 Sleep 下唤醒等同复位, 程序从复位地址 0000H 开始执行。

表 5-5 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:	1	0	0	0	0	0	0	0

只有 POR rst, Lbor rst 和写 0 操作才能清除该寄存器。

1.POR Reset 发生后, 寄存器中的 POR BIT 置位。

2.LBOR Reset 发生后, 寄存器中的 LBOR BIT 置位。

详细的 Reset 后的状态在 Reset SPEC 单元中。

位	功能描述
WKR:	PDM复位唤醒标志。系统处于PDM下可被int0_n, int1_n, RX0, RX1, RTC, PMU/TBS, RTC唤醒复位, 唤醒复位发生时, 复位状态寄存器的WKR复位标志位WKR被设置为1。写零清零。

当系统处于SLEEP下可被int0_n, int1_n, RX0, RX1, RTC, KEY, PMU/TBS 等信号触发源,唤醒复位。唤醒复位发生时, 复位状态寄存器的WKR复位标志位WKR被设置为1。

表 5-6 WAKEIF 唤醒标志寄存器 (B7H)

WAKEIF		Address:						
	Bit7	6	5	4	3	2	1	Bit0
Read:	int0_n	int1_n	RX0	RX1	RTC	Key	PMU/TBS	
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RSTSR. 7	int0_n: int0唤醒标志。int0唤醒发生时, 设置标志位为1, 同时将其他唤醒标志位清零。(写零清零)。
RSTSR. 6	int1_n: int1唤醒标志。int1唤醒发生时, 设置标志位为1, 同时将其他唤醒标志位清零。读这个寄存器可以将这个寄存器清零(写零清零)。
RSTSR. 5	RX0: RX0唤醒标志。RX0唤醒发生时, 设置标志位为1, 同时将其他唤醒标志位清零。(写零清零)。
RSTSR. 4	RX1: RX1唤醒标志。RX1唤醒发生时, 设置标志位为1, 同时将其他唤醒标志位清零。(写零清零)。
RSTSR. 3	RTC: RTC中断唤醒标志。RTC中断发生时将会产生导致RTC唤醒, 唤醒状态寄存器的RTC唤醒标志位RTC被设置为1。写零清零。
RSTSR. 2	Key: 唤醒标志。Key上有输入时, 可唤醒芯片唤醒标志位Key被设置为1。写零清零。
RSTSR. 1	PMU/TBS: PMU/TBS中断唤醒标志。PMU/TBS中断发生时将会产生导致PMU唤醒, 唤醒状态寄存器的PMU唤醒标志位RTC被设置为1。写零清零。

注意:

- 1) RSTSR.3位置位 1, 表示是由RTC中断信号源头, 引起了Wake_UP唤醒。具体的中断信息 (RTC的8种中断源通过查询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) RSTSR.2位置位 1, 表示是由KEY中断信号源头, 引起了Wake_UP唤醒。具体的中断信息 (KEY的4种中断源通过查询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) RSTSR.1位置位 1, 表示是由PMU/TBS中断信号源头, 引起了Wake_UP唤醒。具体的中断信息 (PMU的3种中断源通过查询PMIFR对应的位来确认是BORIF, VSYSIF, VDCINIF, 电池电压检测 (VBATIF), 温度检测(TMPIF), 哪个唤醒源引起的唤醒, 详细见PMU单元)。

6 时钟管理单元

6.1 概述

系统时钟管理模块包含系统时钟生成和系统时钟管理两部分。

系统的时钟由两部分组成，一是低频晶振输出频率为 f_{osc} ，二是 PLL 输出频率 f_{pll} 。上电复位后，片上低频振荡电路开始工作，时钟来自片上低频晶振电路 f_{osc} 。经过 pll 后，产生 f_{pll} 高频时钟。

芯片外围单元 RTC、LCD、WDT、PMU、TBS 部分的时钟直接来自低频晶体振荡电路的输出 F_{osc} ，其他的外围单元 (SPI 和 I2C 单元等) 和处理器 CPU52 的时钟都来自系统时钟 f_{cpu} ， f_{cpu} 可配置为 F_{osc} 和 F_{pll} 两种。电能计量单元 EMU 的时钟来自于 f_{pll} ，通过分频寄存器进行分频后用于 EMU_CLK。红外 38K 模块时钟由 f_{pll} 提供。

OSC 产生 32.768KHz 的时钟，系统利用 PLL 把 32.768KHz 的时钟倍频到 5.505024MHz 或者 11.010048MHz。

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。

系统时钟 (f_{cpu}) 可以选择在低频时钟或高频时钟频率下工作，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} 。

低频时钟频率由外部晶振提供，产生 32768Hz 时钟，此时钟电路一值保证打开；高频时钟频率由 PLL 电路产生，主时钟可以提供四种时钟频率，11.010048MHz，5.505024MHz 及其二者的分频。具体的分频时钟，由 CLKCFG 的 PRIP[1:0] 位决定，PRION 位作为 PLL 电路使能控制位。

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。

从 OSC 时钟切换到 PLL 时钟时，使能 PLL 后需要等待 2ms，然后切到 PLL 时钟。

6.2 时钟系统框图

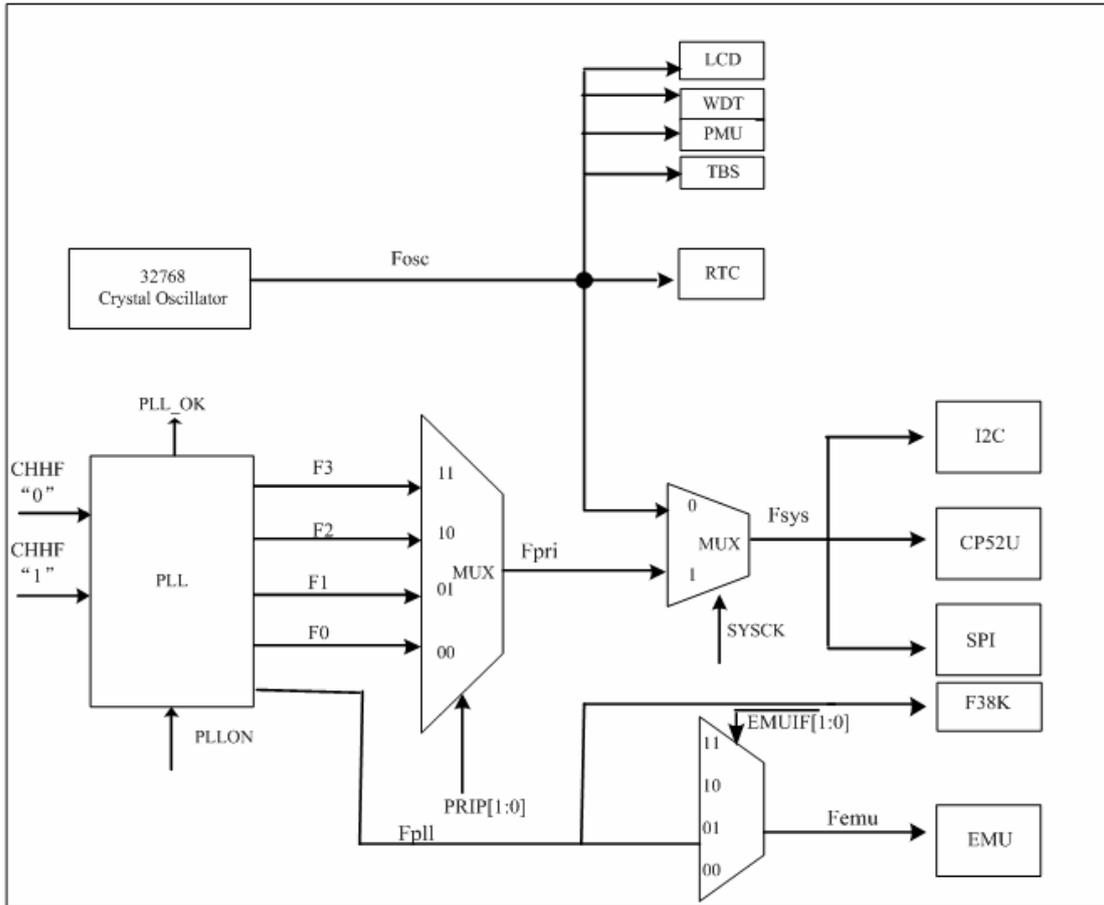


图 6-1

6.3 主要功能

6.3.1 低频振荡电路

上电复位后，缺省值 $SYSCK=0$ ，CPU 时钟默认为低频时钟 F_{osc} 。低频振荡器无使能信号，在任何情况下都打开。

6.3.2 高频振荡电路

6.3.2.1 高频输出功能

当 $SYSCK=1$ 时，高频时钟 f_{cpu} 由寄存器 PRIP [1:0]和 CHHF 决定，如下表所示：

当 $CHHF=1$ ：

PRIP [1:0]	f_{cpu} (MHz)
00	$F_0= 1.376256$
01	$F_1= 2.752512$

10	F2= 5.505024
11	F3= 11.010148
××	0

当 CHHF=0:

PRIP [1:0]	fcpu (MHz)
00	F0= 0.688128
01	F1= 1.376256
10	F2= 2.752512
11	F3= 5.505024

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

6.3.2.2 PLL 性能指标

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

6.4 系统时钟切换方法

6.4.1 概述

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fppll或其分频。

6.4.2 系统时钟fcpu 从低频时钟fosc 切换到高频时钟fppll

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 PRIP [1:0]确定高频时钟频率(例如 PRIP [1:0]=10)=>打开高频时钟电路 PRION=1 => 等待 2ms=> => 将 SYSCK 置 1,完成 fcpu 由 fosc 切换到 fppll。

6.4.3 系统时钟fcpu 从高频时钟fppll 切换到低频时钟fosc

6.4.3.1 系统时钟 Fsys 从高频时钟 fppll 切换到低频时钟 fosc 的流程

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 置为 0,完成 fcpu 由 fppll 切换到 fosc。

6.4.3.2 系统时钟 fcpu 在高频时钟 fppll 之间切换

此操作直接通过选择 PRIP [1:0]实现。

6.4.3.3 系统时钟 Fsys 在高频时钟 Fpri5.5M 和 11M 之间切换

可以通过对 PLLCFG 中的 CHHF 进行操作来选择 PLL 输出 5.5M 的高频时钟, 或者 11M 的高频时钟。“CHHF” Default 下为“0”。如果需要 CHHF 由“0”变为“1”(PLL 输出时钟由 5.5M 和 11M),或者由“1”

变为“0” (PLL 输出时钟由 11M 和 5.5M)。首先应该配置 CLKCFG 中的 SYSCK=0,切换到低频, 然后配置 PLLCFG 中的 CHHF “1” 或者 “0”, 通过查询 PLLCFG 中的 LOCK 位后切换到相应的高频时钟。

6.4.4 EMU 的时钟是独立设置可控的

EMU_CLK 直接来自于 fp11 时钟, 打开 EMU 的同时要打开 PLL 时钟, 通过设置 EMU 部分的时钟分频系数, 可以设置 EMU_CLK 为 900K, 450K, 225K. (fp11 选择为 5.505024MHZ)。

如果 CHHF=1, PLL 输出的 11.010048MHZ, 系统会自动二分频为 5.505024MHZ, 给与 fp11。如果 CHHF=0,PLL 输出的 5.505024MHZ, 直接给与 fp11。

6.4.5 进入 Hold 模式

通过对写保护的CLKCFG寄存器进行操作, 控制CPU进入Hold Mode。在Hold模式下, 保持所有CPU的功能。Hold模式下CPU相连接的外设功能模块功能保持有效, 只是CPU进入Standby状态, 如果使能中断, CPU可以从中断中唤醒, 开始从中断的ISR执行程序。

6.4.6 进入 Sleep 模式

通过对写保护的 CLKCFG 寄存器进行操作, 控制 CPU 进入 Sleep Mode。在 Sleep Mode 下, (可以被 int0_n, int1_n, RX0, RX1, PMU, RTC, KBI 唤醒复位)。

在 ATT7039 进入 Sleep 后的唤醒操作的功能, 中断信号 int0_n, int1_n, RX0, RX1, PMU, RTC,KBI, WDT,外部 Reset, 这 9 个中断源, 会造成芯片在 Sleep 模式下的唤醒复位), 除了外部 Reset 信号外, 都可以设置为可屏蔽和不可屏蔽。如果某个中断功能设置为可以屏蔽, 在 Sleep 模式下该中断信号不能使 CPU 唤醒。

只有在电池供电的情况下系统才可以进入 sleep 模式。

6.4.7 时钟管理单元内部保护机制

6.4.7.1 fp11 时钟

fcpu 的时钟必须要在PLL稳定后 (使能PLL后, 需要2mS延时), 才能由fosc切换至PLL。(保护机制

- 当 SYSCK=1, 表示选择了 fp11 作为系统时钟 fcpu, 此时不能将关闭高频 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, 而 SYSCK 清为 0, 选择 fosc 时钟作为系统时钟。

6.4.8 外设时钟管理

6.4.8.1 EMU 部分的时钟 Femu

EMU 的时钟 Femu 由高频 PLL 电路提供, 当 PLL 打开, 即 PRION=1 时, Femu 频率会根据高频振荡电路 fp11 输出频率, 直接用 emu_ctrl 寄存器的 Emu_lkctrl[1:0], 得到相应的输出频率 Femu, Femu 与 fp11 的对应关系如下表所示:

Emu_lkctrl [1:0]	系数 N	Fadc=fp11/N (KHz)	Femu
00	6	917.504	Fp11
01	12	458.752	Fp11/2
10	24	229.376	Fp11/4
11	24	229.376	Fp11/4

6.4.8.2 红外输出调制单元的时钟 Firf

当 PLL 打开, 即 PRION=1 时, $Firf=38.229\text{KHz}$ 。fp11 与 Firf 的关系如下表所示

fp11 (MHz)	系数 M	$Firf=fp11/M/18$ (KHz)
11.010048	16	38.4

6.5 特殊功能寄存器

表 6-1 CLKCFG 时钟配置寄存器(9EH)

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

位	功能描述									
W_EN	<p>w_en位: 时钟配置寄存器模式写保护位</p> <p>如果用户要改系统的模式, 在向mode1和mode0 位写入新的值时, 必须同时将w_en位置1, 例如: ‘b1xxxx10, 此时改变模式到hold, 该寄存器的其他位不会受到该写操作的影响; w_en位置0时, 可对该寄存器其它位操作。(对该寄存器的写入要采取先打开写保护寄存器)</p>									
SYSCK	<p>SYSCK用于选择系统时钟。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>SYSCK</td> <td>fcpu</td> </tr> <tr> <td>0</td> <td>fcpu=fosc</td> </tr> <tr> <td>1</td> <td>fcpu=fp11</td> </tr> </table>	SYSCK	fcpu	0	fcpu=fosc	1	fcpu=fp11			
SYSCK	fcpu									
0	fcpu=fosc									
1	fcpu=fp11									
Mode[1:0]	<p>表 Mode[1:0]和工作模式对应表</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Mode</td> <td>Mode 1</td> <td>Mode 0</td> </tr> <tr> <td>Hold</td> <td>1</td> <td>0</td> </tr> <tr> <td>Sleep</td> <td>1</td> <td>1</td> </tr> </table> <p>(对该寄存器的写入要采取先打开写保护寄存器, 再写入 ‘b1xxx10xx), 才能设置系统进入 Hold 或者 Sleep 的节电模式。</p> <p>1. Mode[1..0]Default值为[00]。如果W_EN使能, 对该位置写[00], [01].系统产生Soft Reset。(如果在上电没有稳定的时候出现W_EN=1. Mode[1..0]=00的情况, 会造成系统的Soft Reset)。</p> <p>2. 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]进入SLEEP MODE, 就会出现Soft Reset。</p>	Mode	Mode 1	Mode 0	Hold	1	0	Sleep	1	1
Mode	Mode 1	Mode 0								
Hold	1	0								
Sleep	1	1								

表 6-2 PLLCFG 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	0		0	1	1

位	功能描述																									
PRION	PRION=0, PLL 关闭。 PRION=1, PLL 打开。																									
CHHF	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]	PRIP[1:0]用于设置PLL的输出频率。 当 CHHF=1: <table border="1" data-bbox="550 510 1139 725" style="margin-left: 40px;"> <thead> <tr> <th>fpll</th> <th>PRIP1</th> <th>PRIP0</th> </tr> </thead> <tbody> <tr> <td>F0= 1. 376256M</td> <td>0</td> <td>0</td> </tr> <tr> <td>F1= 2. 752512M</td> <td>0</td> <td>1</td> </tr> <tr> <td>F2= 5. 505024M</td> <td>1</td> <td>0</td> </tr> <tr> <td>F3= 11. 010148M</td> <td>1</td> <td>1</td> </tr> </tbody> </table> 当 CHHF=0: <table border="1" data-bbox="550 766 1134 981" style="margin-left: 40px;"> <thead> <tr> <th>PRIP [1:0]</th> <th>fpll (MHz)</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>F0= 0.688128</td> </tr> <tr> <td>01</td> <td>F1= 1.376256</td> </tr> <tr> <td>10</td> <td>F2= 2.752512</td> </tr> <tr> <td>11</td> <td>F3= 5.505024</td> </tr> </tbody> </table>	fpll	PRIP1	PRIP0	F0= 1. 376256M	0	0	F1= 2. 752512M	0	1	F2= 5. 505024M	1	0	F3= 11. 010148M	1	1	PRIP [1:0]	fpll (MHz)	00	F0= 0.688128	01	F1= 1.376256	10	F2= 2.752512	11	F3= 5.505024
fpll	PRIP1	PRIP0																								
F0= 1. 376256M	0	0																								
F1= 2. 752512M	0	1																								
F2= 5. 505024M	1	0																								
F3= 11. 010148M	1	1																								
PRIP [1:0]	fpll (MHz)																									
00	F0= 0.688128																									
01	F1= 1.376256																									
10	F2= 2.752512																									
11	F3= 5.505024																									

注意: CLKCFG , PLLCFG 是写保护寄存器。只有打开写保护, 才能对该寄存器操作。
 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]**进入 SLEEP MODE , 就会出现 **Soft Reset**。

表 6-3 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	TBS_EN	LCD_EN	EMU_EN	I2C_EN	SPI_EN		
Write:							X	X
Reset:	0	0	0	0	0	0	0	0

说明:

- KBI_EN=1, Enable KBI模块。 KBI_EN=0, Disable KBI模块。
- TBS_EN=1, Enable TBS模块。 TBS_EN=0, Disable TBS模块。
- LCD_EN=1, Enable LCD 模块。 LCD_EN=0, Disable LCD 模块。
- EMU_EN=1, Enable EMU 模块。 EMU_EN=0, Disable EMU 模块。
- I2C_EN=1, Enable I2C 模块。 I2C_EN=0, Disable I2C 模块。
- SPI_EN=1, Enable SPI 模块。 SPI_EN=0, Disable SPI 模块。

注意:

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。

7 时钟输出模块

7.1 概述

ATT7039 有分频的时钟输出管脚，和 PE.4 复用，当 PIN 的 PE.4 被配置为时钟输出管脚 Clkout 时，可以输出 CPU 时钟频率的分频。

在 Sleep 模式下该分频不输出。

7.2 功能描述

CLKOUT pin 脚输出频率由 CLKOUT1 和 CLKOUT0 两位控制，共可以输出 4 种系统时钟分频频率。

CLKOUT1	CLKOUT0	CLKOUT输出频率
0	0	fcpu/2
0	1	fcpu/4
1	0	fcpu/8
1	1	fcpu/16

7.3 特殊功能寄存器

表 7-1 PWMCR PWM 控制寄存器 (B1H)

PWM Control Register (PWMCR)		Address: B1H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	CLKOUT1	CLKOUT0	IRTX1	IRTX0	PWMSEL	PWMCKS	PPOL	PWM_EN
Write:								
Reset:	0	0	0	0	0	0	0	0

CLKOUT1和CLKOUT0的配置结果如上表所示。

8 系统复位

8.1 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，第二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

8.2 功能说明

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，并保持为 128 个 fosc。

8.2.1 上电复位 POR (Power on reset)

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_{osc}
- 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。

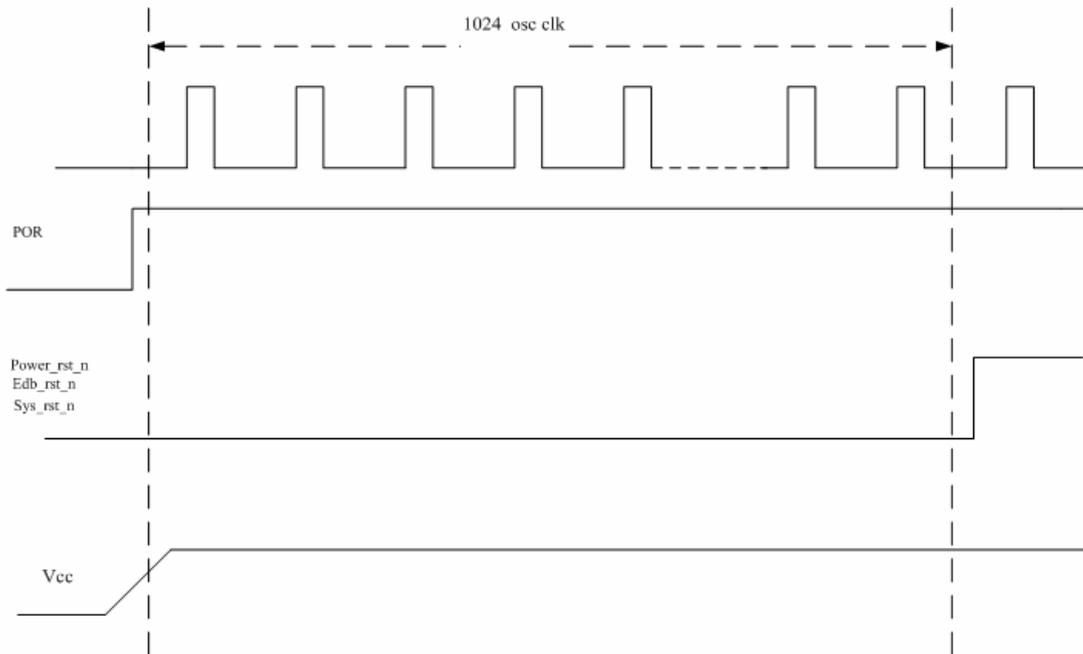
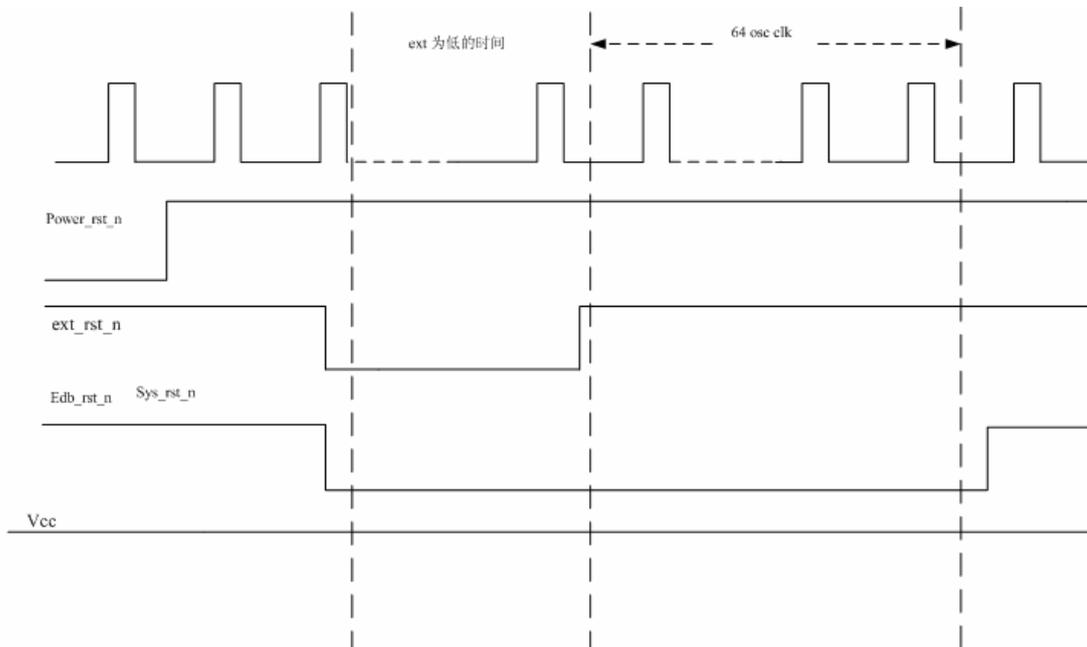


图 8-2 上电复位说明

8.2.2 外部引脚复位 /RST (内部上拉30K 电阻)

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 128 个 fosc。

如果/RST 低电平脉宽比 2us 窄, 系统不复位。



8.2.3 掉电复位 BOR (Brownout Reset) & LBOR

当掉电检测电路检测到电源电压低于电压 Vborl 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电压 V_{borh} 时，BOR 输出高电平，IRST 在 1024 个 f_{osc} 时间之后变为高电平。

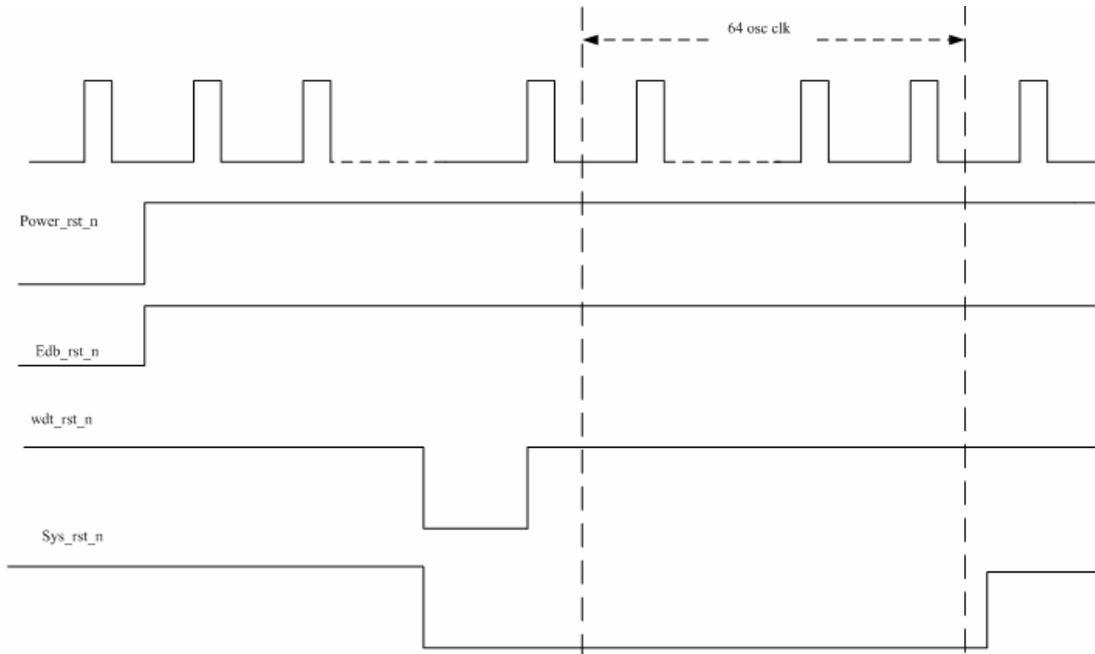
掉电复位 BOR 产生时，下面的事件将会发生：

- 产生一个 BOR 脉冲
- 内部复位信号 IRST 有效
- 计数 1024 个 f_{osc}
- 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

LBOR 与 BOR 过程基本相同，两者是与的关系。

8.2.4 Watchdog 复位 WDTR

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 128 个 f_{osc} 。



8.2.5 唤醒复位 WKR

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 128 个 f_{osc} 后，释放内部复位信号 IRST

8.2.6 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]进入 SLEEP MODE，就会出现 Soft Reset。（寄存器写保护关闭的情况下，对 Mode[1..0]写[11]或者[10]进入 HOLD 或者 SLEEP MODE 的操作都是无效的）。

8.3 特殊功能寄存器

表 8-1 复位标志寄存器(0xA7H, RSTSR)

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:	1	0	0	0	0	0	0	0

位	功能描述
RSTSR.7	POR: 上电复位标志。上电复位发生时, 设置标志POR位为1, 同时将其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)。
RSTSR.6	RST: 引脚/RST复位标志。外部复位引脚/RST出现比2us宽的低电平时, 将会产生RST PIN复位, 复位状态寄存器的复位标志位RST被设置为1。(写零清零)。
RSTSR.5	WDT: WDT溢出复位标志。WatchDog Timer溢出时将会产生导致WDT复位, 复位状态寄存器的WDT复位标志位WDT被设置为1。(写零清零)。
RSTSR.4	BOR: 掉电复位标志。如果AVCC电压低于设定模式的Vbor1电压值, 将会立即产生BOR复位, 复位状态寄存器的BOR复位标志位BOR被设置为1。(写零清零)。
RSTSR.3	WKR: PDM复位唤醒标志。系统处于PDM下可被int0_n, int1_n, RX0, RX1, RTC, WDT, RST PIN, PMU唤醒复位, 唤醒复位发生时, 复位状态寄存器的WKR复位标志位WKR被设置为1。(写零清零)。
RSTSR.2	LBOR: 低掉电复位标志。如果AVCC电压低于设定芯片工作的最低电压值, 将会立即产生LBOR复位, 复位状态寄存器的LBOR复位标志位LBOR被设置为1。(写零清零)。
RSTSR.1	Soft rst: 当软复位标志。当对一些寄存器的进行了一些不被允许的操作时, 就会产生软复位。如CLKCFG的mode位。
RSTSR.0	DE_RST: debug reset标志位, 当发生debug reset是改为置1。

只有 POR rst, Lbor rst 和读操作才能清除该寄存器。

8.4 被分级复位的寄存器说明

复位级别	复位源	需要置位的	不能复位的	需要清零的
1	POR	复位标志寄存器中的POR		其余的清0
	LBOR	复位标志寄存器中的LBOR位		其余的清0
2	外部RESET	复位标志寄存器中的RST	1、RTC的补偿寄存器和年月日、星期、时分秒寄存器。 2、BOR、LVREF、VSYSL的level寄存器 3、Jtag复位不能复位OCDS(jtag)	其余的清0
	BOR	复位标志寄存器中的BOR位		其余的清0
	DEBUG	复位标志寄存器中		其余的清0

	RESET	的DEBUG_RST位	模块	
3	WDT	复位标志寄存器中的WDT	1、R T C 的补偿寄存器和时年月日、星期、时分秒寄存器。	其余的清 0
	software reset	复位标志寄存器中的Soft rst位	2、BOR、LVREF、VSY S 的 l e v e l 寄存器。	其余的清 0
	PDM唤醒	复位标志寄存器中的WKR位	4、L C D 相关寄存器。 5、g p i o / l c d 复用状态 6、g p i o 的输入输出状态和方向控制寄存器 7、S U P D C	其余的清 0

9 WDT

9.1 概述

Watchdog Timer 是一个特殊的定时器，计时计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。

特征：

- 采用硬件狗设计
- SLEEP模式下WDT开启/关闭可选
- 可以通过外部pin JTAG_WDTEN进行控制

9.2 工作模式

和外部pin 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 module (0/off, 1/on)
0	0	0
0	1	1
1	0	1
1	1	1

在 Hold 和 Sleep Mode 下：

置寄存器 WDTCON 中的 WDT_SH=0,在 SLEEP MODE 或者 Hold MODE 下屏蔽掉 WDT 的 CLK, WDT 功能无效，同时并且将 WDT Counter 进行清零。

注意：当 WDT 被关闭，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 或者 HOLD Mode 下的状态
1	开启
0	关闭

9.3 特殊功能寄存器

表 9-1 WDTCON WDT 控制寄存器 (C9H)

WDTCON		Address: C9H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	WDT_S2	WDT_S1	WDT_S0	WDTEN	WDT_SH	WDTCLR2	WDTCLR1	WDTCLR0
Write:								
Reset:	1	0	0	0	1	0	0	0

注意：（该位置的置位在HOLD和SLEEP下有效的，才能够屏蔽WDT的CLK）。

WDT_S[2:0]用于选择 WDT 溢出时间：

WDT_S2	WDT_S1	WDT_S0	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 是 WDT 使能位，当 WDT 模块关闭后，计数器清零，控制寄存器值保持不变。往 WDTCLR[2:0] 写入数据 101，可以将 WDT 进行清零。写其他数据无效。

WDTSH:

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

注意：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。

10 中断系统

10.1 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_n 中断、定时器 0/1/2 中断、串行口 UART0/UART1 中断。

另外 5 个中断，利用 XC8051 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 或 IRQ_TPS)

10.2 中断列表

表 4-1 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
SPI, KBI	4	004BH	9	IEN1.1 KEYIE	KEYIF SPSTA
External Interrupt 1	5	0013H	2	IEN0.2	TCON.3
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
CC	12	006BH	13	IEN1.5	IRCON.5

10.3 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	SPI, KBI(EXT INT2)	
2	IP1.2 IP0.2	External Interrupt 1	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	CC(EXT INT6)	

10.4 寄存器

10.4.1 中断使能寄存器

表 4-2 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	EX1	ET0	EX0
Write:								
Reset:	0	0	0	0	0	0	0	0

说明:

EAL——全体中断使能

当 EAL=0, 关闭所有中断;

当 EAL=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——定时器 2 中断使能

当 ET2=0, 定时器 2 中断关闭;

当 ET2=1 并且 EAL=1, 定时器 2 中断使能;

ES0——串口 0 中断使能

当 ES0=0, 串口 0 中断关闭;

当 ES0=1 并且 EAL=1, 串口 0 中断使能;

ET1——定时器 1 溢出中断使能

当 ET1=0, 定时器 1 溢出中断关闭;

当 ET1=1 并且 EAL=1, 定时器 1 溢出中断使能;

EX1——外部中断 1 使能

当 EX1=0, 外部中断 1 关闭;

- 当 EX1=1 并且 EAL=1, 外部中断 1 使能;
- ET0——定时器 0 溢出中断使能
- 当 ET0=0, 定时器 0 溢出中断关闭;
- ET0=1 并且 EAL=1, 定时器 0 溢出中断使能;
- EX0——外部中断 0 使能
- 当 EX0=0, 外部中断 0 关闭;
- 当 EX0=1 并且 EAL=1, 外部中断 0 使能;

表 4-3 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	ECC	ERTC	EEMU	EPMU	EKBI_SPI	EI2C
Write:								
Reset:	0	0	0	0	0	0	0	0

说明:

- EXEN2——定时器 2 外部重载中断使能
- 当 EXEN2=0, 定时器 2 外部重载中断关闭
- 当 EXEN2=1 并且 EAL=1, 定时器 2 外部重载中断使能
- ECC——定时器 2 比较捕获中断使能
- 当 ECC =0, 定时器 2 比较捕获中断关闭
- 当 ECC =1 并且 EAL=1, 定时器 2 比较捕获中断使能
- ERTC——RTC 中断使能
- 当 ERTC =0, RTC 中断关闭
- 当 ERTC =1 并且 EAL=1, RTC 中断使能
- EEMU——EMU 中断使能
- 当 EEMU =0, EMU 中断关闭
- 当 EEMU =1 并且 EAL=1, EMU 中断使能
- EPMU——PMU 中断使能
- 当 EPMU =0, PMU 中断关闭
- 当 EPMU =1 并且 EAL=1, PMU 中断使能
- ESPI_KBI——SPI/KBI 中断使能
- 当 ESPI_KBI =0, SPI/KBI 中断关闭
- 当 ESPI_KBI =1 并且 EAL=1, SPI/KBI 中断使能
- EI2C——I2C 中断使能
- 当 EI2C =0, I2C 中断关闭
- 当 EI2C =1 并且 EAL=1, I2C 中断使能

表 4-4 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

说明:

ES1——串口 1 中断使能

当 ES1=0, 串口1中断关闭

当 ES1=1 并且 EAL=1, 串口1中断使能

表 4-5 EMU 中断使能寄存器 EMUIE (ACH)

EMU Interrupt Enable Register (EMUIE)			Address: ACH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PFIE	QFIE	SFIE	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。

表 4-7 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。

10.4.2 中断标志寄存器

具体内容见各章的特殊功能寄存器小节。

10.4.3 中断优先级寄存器

表 4-15IP0 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 4-16 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

10.5 中断处理

中断系统遵循下列两条基本规则：

1. 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；

2. 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 中清除 8 个中断标志。

11 GPIO

11.1 概述

提供 PA[7..0], PB[7..0], PC[7..0], PD[7..0], PE[4..0], P0[7..0], P1[7..0], P2[7..0], P3[7..0] 个并行端口, 支持 69 个双向 I/O 引脚, 可以分别配置成输入或者输出模式。作为输入方式时, 带有 Schmitt trigger 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。

11.2 寄存器说明

11.2.1 GPIO 输出复用配置寄存器

表 11-1 LCDCFG LCD 输出复用配置寄存器 (C6H)

LCDCFG		Address: C6H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	PDH	PDL	PTCH	PTCL	PTBH	PTBL	PTAH	PTAL
Write:								
Reset:	0	0	0	0	0	0	0	0

PTAL=0:	SEG0/PA0—SEG3/PA3 配置为标准 I/O:	PA0—PA3
PTAL=1:	SEG0/PA0—SEG3/PA3 配置为 LCD 输出:	SEG0—SEG3
PTAH=0:	SEG4/PA4—SEG7/PA7 配置为标准 I/O:	PA4—PA7
PTAH=1:	SEG4/PA4—SEG7/PA7 配置为 LCD 输出:	SEG4—SEG7 或者 COM4—COM7
PTBL=0:	SEG8/PB0—SEG11/PB3 配置为标准 I/O:	PB0—PB3
PTBL=1:	SEG8/PB0—SEG11/PB3 配置为 LCD 输出:	SEG8—SEG11
PTBH=0:	SEG12/PB4—SEG15/PB7 配置为标准 I/O:	PB4—PB7
PTBH=1:	SEG12/PB4—SEG15/PB7 配置为 LCD 输出:	SEG12—SEG15
PTCL=0:	SEG24/PC0—SEG27/PC3 配置为标准 I/O:	PC0—PC3
PTCL=1:	SEG24/PC0—SEG27/PC3 配置为 LCD 输出:	SEG24—SEG27
PTCH=0:	SEG28/PC4—SEG31/PC7 配置为标准 I/O:	PC4—PC7
PTCH=1:	SEG28/PC4—SEG31/PC7 配置为 LCD 输出:	SEG28—SEG31
PTDL=0:	SEG32/PD0—SEG35/PD3 配置为标准 I/O:	PD0—PD3
PTDL=1:	SEG32/PD0—SEG35/PD3 配置为 LCD 输出:	SEG32—SEG35
PTDH=0:	SEG36/PD4—SEG39/PD7 配置为标准 I/O:	PD4—PD7
PTDH=1:	SEG36/PD4—SEG39/PD7 配置为 LCD 输出:	SEG36—SEG39

注意:

1. 仅当配置 PTAH=1, 同时在 LCD 的单元部分配置 DUTY[1:0]=11: 1/8 duty——COM0 到 COM7 都被使用的时候。LCD 的 Duty 配置为 1/8 duty, PTAH 才能被配置为 COM4—COM7。

表 11-2 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	P245	P223	P201	P067	P045	P023	P001
Write:								
Reset:	0	0	0	0	0	0	0	0

P001=0: SEG16/P0.0—SEG17/P0.1 配置为标准 I/O: P0.0—P0.1
 P001=1: SEG16/P0.0—SEG17/P0.1 配置为 LCD 输出: SEG16—SEG17
 P023=0: SEG18/P0.2—SEG19/P0.3 配置为标准 I/O: P0.2—P0.3
 P023=1: SEG18/P0.2—SEG19/P0.3 配置为 LCD 输出: SEG18—SEG19
 P045=0: SEG20/P0.4—SEG21/P0.5 配置为标准 I/O: P0.4—P0.5
 P045=1: SEG20/P0.4—SEG21/P0.5 配置为 LCD 输出: SEG20—SEG21
 P067=0: SEG22/P0.6—SEG23/P0.7 配置为标准 I/O: P0.6—P0.7
 P067=1: SEG22/P0.6—SEG23/P0.7 配置为 LCD 输出: SEG22—SEG23

注意:

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, PTC, PTD 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。

P201=0: TD0/TX1/P2.0—TMS/RX1/P2.1 配置为 UART1: TX1—RX1
 P201=1: TD0/TX1/P2.0—TMS/RX1/P2.1 配置为标准 I/O: P2.0—P2.1
 P223=0: TX0/P2.2—RX0/P2.3 配置为 UART0: TX0—RX0
 P223=1: TX0/P2.2—RX0/P2.3 配置为标准 I/O: P2.2—P2.3
 P245=0: TMUXOUT/P2.4—SF/P2.5 配置为脉冲输出: TMUXOUT—SF
 P245=1: TMUXOUT/P2.4—SF/P2.5 配置为标准 I/O: P2.4—P2.5
 P267=0: PF/P2.6—QF/P2.7 配置为脉冲输出: PF—QF
 P267=1: PF/P2.6—QF/P2.7 配置为标准 I/O: P2.6—P2.7

注意:

在 /JTAG/WDTEN pin 被拉到低电平的时候, 使能 JTAG 调试功能, TD0/TX1/P2.0—TMS/RX1/P2.1, 两个 PIN 被配置使能为 TD0, TMS 功能。

如果 SUPDC 中的 EMU_EN=0, 关闭芯片的 EMU 单元。P245=0, P267=0, 相应的 PIN 被配置为 SF, PF, QF 的情况下, 这三个 PIN 输出为低电平。

表 11-3 KEYCFG P1 输出复用配置寄存器 (C7H)

KEYCFG		Address: C7H						
	Bit7	6	5	4	3	2	1	Bit0
Read:				SPI	KEY3	KEY2	KEY1	KEY0
Write:								
Reset:	0	0	0	0	0	0	0	0

KEY_x=0: P1.x/KEY_x 配置为标准 IO: P1.x
 KEY_x=1: P1.x/KEY_x 配置为键盘输入: KEY_x
 SPI=0: P1.X P1[7..4] 配置成 GPIO
 SPI=1: P1.X P1[7..4] 配置成硬件 SPI 接口功能有效

注意:

- 1) 只有在该位置配置了 P1[4..7]为 SPI 功能 PIN,同时在 SUPDC 寄存器中选择了 SPI_EN 使能 SPI 功能后。(表示使能硬件 SPI 功能)。
- 2) 只有在该位置配置了相应的 KEYx=1, 使能了该功能 PIN 的 KEY 功能, 同时在 SUPDC 寄存器中选择了 KBI_EN 使能 KBI 功能后。(才能使能相应的 PIN 上的 KEY 功能)。
- 3) 在/JTAG/WDTEN pin 被拉到低电平的时候, 使能 JTAG 调试功能后, TCK/KEY0/P1.0—TDI/KEY1/P1.1, 两个 PIN 被配置使能为 TCK,TDI。
- 4) 如果 SUPDC 中的 SPI_EN=0,关闭 SPI 功能模块。同时在 KEYFG 中又配置为 SPI=1, 配置为 SPI 总线功能接口/CS,SCLK,MOSI,MISO 功能 PIN 的时候, 该相应的功能 PIN 是默认为输入功能, 弱上拉。
- 5) 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 KEYx=1, 配置为 KEY 功能 PIN 的时候。对应的 KEY0,KEY1 PIN 默认为 Floating 状态的, 上升沿触发 KEY 中断。KEY2,KEY3 相应的功能 PIN 是默认为输入功能, 弱上拉, 下降沿触发 KEY 中断。(详细见 KBI 章节)。

表 11-4 P3CFG P3 输出复用配置 (CFH)

P3CFG		Address: CFH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	PWM	T2	T1	T0	INT1	INT0	I2C	X
Write:								
Reset:	0	0	0	0	0	0	0	0

I2C=0: P3.0/SCL, P3.1/SDA 配置为标准 IO: P3.0, P3.1

I2C=1: P3.0/SCL, P3.1/SDA 配置为 I2C 总线功能接口。

INT0=0: P3.2/INT0 配置为标准 IO: P3.2

INT0=1: P3.2/INT0 配置为: /INT0

INT1=0: P3.3/INT1 配置为标准 IO: P3.3

INT1=1: P3.3/INT1 配置为: /INT1

T0=0: P3.4/T0 配置为标准 IO: P3.4

T0=1: P3.4/T0 配置为: T0

T1=0: P3.5/T1 配置为标准 IO: P3.5

T1=1: P3.5/T1 配置为: T1

T2=0: P3.6/T2 配置为标准 IO: P3.6

T2=1: P3.6/T2 配置为: T2

PWM=0: TMS/P3.7/PWM 配置为标准 IO: P3.7

PWM=1: TMS/P3.7/PWM 配置为: PWM

注意:

- 1) 只有在 PWMCON 中使能了 PWM_EN=1, 同时配置 P3CFG 中的 PWM=1 为有效的情况下, 该 PIN 才能输出相应的 PWM 信号。
- 2) 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。
- 3) 如果 SUPDC 中的 PWM_EN=0, 关闭 PWM 功能模块。同时在 P3CFG 中又配置为 PWM=1, TMS/P3.7/PWM 配置为 PWM 功能输出 PIN 的时候, 该 PIN 在这种配置下输出低电平。

表 11-5 PECFG PE 口输出复用配置 (ABH)

PECFG		Address: ABH						
	Bit7	6	5	4	3	2	1	Bit0
Read:		TMUX1	TMUX0	CLKOUT	CC3	CC2	CC1	CC0
Write:								
Reset:	0	0	0	0	0	0	0	0

CC0 = 0: CC1/PE0 配置为标准 IO PE0

CC0 = 1: CC1/PE0 配置为第二功能 PIN CC0

CC1 = 0: CC1/PE1 配置为标准 IO PE1

CC1 = 1: CC1/PE1 配置为第二功能 PIN CC1

CC2 = 0: CC2/PE2 配置为标准 IO PE2

CC2 = 1: CC2/PE2 配置为第二功能 PIN CC2

CC3 = 0: CC3/PE3 配置为标准 IO PE3

CC3 = 1: CC3/PE3 配置为第二功能 PIN CC3

CLKOUT = 0: CLKOUT /PE4 配置为标准 IO PE4

CLKOUT = 1: CLKOUT /PE4 配置为第二功能 PIN CLKOUT

通过配置 TMUX[1:0]来实现 TMUXOUT/SF (PIN)不同的功能输出, 下面表格的配置来实现选择 TMUXOUT/SF 的不同的输出功能。

TMUX1	TMUX0	TMUXOUT/SF (PIN)
0	0	TOUT
0	1	SF
1	0	PF
1	1	QF

注意:

3. TMUXOUT 的输出功能在切换到 SF, PF, QF 的时候, 如果 SUPDC 中的 EMU_EN=0, 关闭芯片的 EMU 单元。TMUXOUT 对应的这三个输出信号时的状态为输出为低电平。

4. 以上 I/O 功能复用配置寄存器里面, 包括 LCDCFG, P02CFG, KEYCFG, P3CFG, PECFG 等, 需要增加写入操作的时候得写保护功能, 写保护密码密码是先写 0XCFH; 再写 0XDCH。属于 REG1 类型的写保护。写操作要求如下:

写 REG1 类型寄存器的操作:

```

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;
    
```

具体的详细介绍见 ATT7039 SPEC 中的写保护寄存器部分的介绍。

11.2.2 P0

P0 口是一个与 LCD 的 SEG16-SEG23 复用的 8 位并行端口。P0.0-P0.7 可以配置成直接驱动 LED 模式。

① P0 口数据寄存器

P0		Address: 80H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	P0.7	P0.6	P0.5	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

DDRPO		Address: D5H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	DDRPO7	DDRPO6	DDRPO5	DDRPO4	DDRPO3	DDRPO2	DDRPO1	DDRPO0
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:	LEDP07	LEDP06	LEDP05	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 DDRPOx 配置成输出模式时才有效。

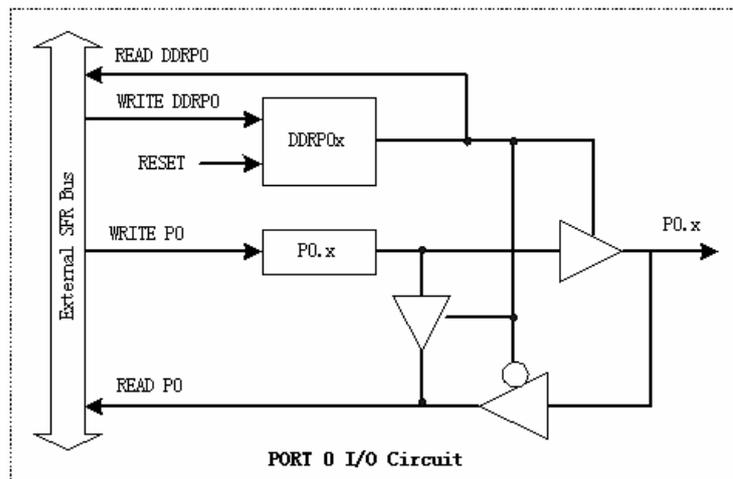


图 10-1

11.2.3 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:	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
Write:								
Reset:	u	u	u	u	u	u	u	u

② P1 口方向寄存器

DDRP1			Address: D6H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	DDRP17	DDRP16	DDRP15	DDRP14	DDRP13	DDRP12	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

11.2.4 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:	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
Write:								
Reset:	u	u	u	u	u	u	u	u

② P2 口方向寄存器

DDRP2			Address: D7H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	DDRP27	DDRP26	DDRP25	DDRP24	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

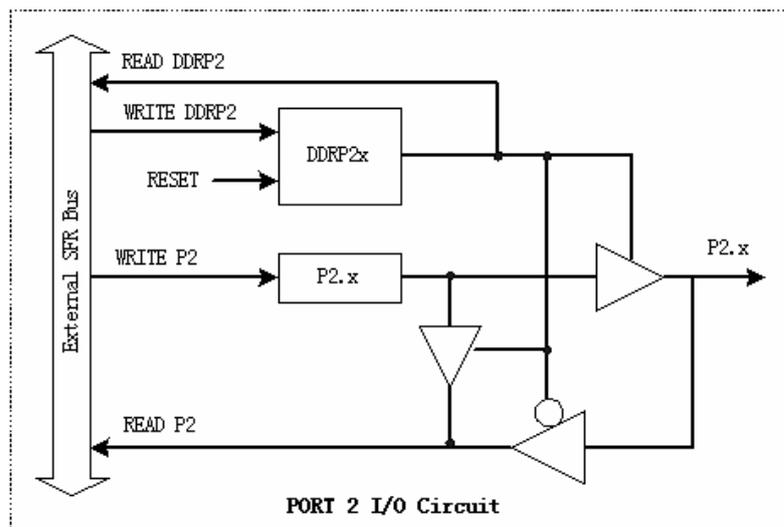


图 10-2

11.2.5 P3

P3 口是一个与 SCL/SDA/INT0/INT1/T0/T1/T2/PWM 复用的 8 位并行端口。

① P3 口数据寄存器

P3		Address: B0H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
Write:								
Reset:	u	u	u	u	u	u	u	u

② P3 口方向寄存器

DDRP3		Address: D9H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	DDRP37	DDRP36	DDRP35	DDRP34	DDRP33	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

11.2.6 Port A

PTA 口是一个与 LCD 的 SEG0-SEG7 复用的 8 位并行端口。

① PTA 口数据寄存器

PTA		Address: BCH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Write:								
Reset:	u	u	u	u	u	u	u	u

② PTA 口方向寄存器

DDRA		Address: D1H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
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

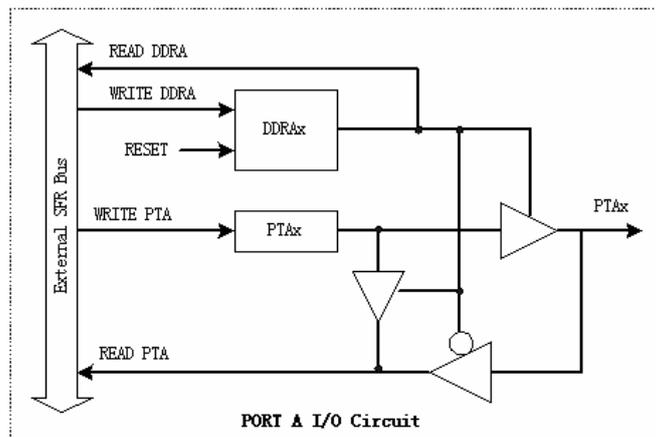


图 10-3

11.2.7 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:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Write:								
Reset:	u	u	u	u	u	u	u	u

② PTB 口方向寄存器

DDRB		Address: D2H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	DDRB7	DDRB6	DDRB5	DDRB4	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

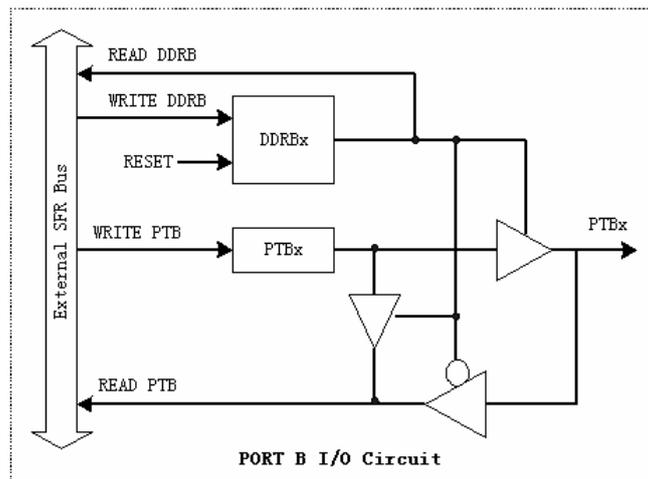


图 10-4

11.2.8 Port C

PTC 口是一个与 LCD 的 SEG23-SEG31 复用的 8 位并行端口。

① PTC 口数据寄存器

PTC		Address: BEH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	PTC7	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
Write:								
Reset:	u	u	u	u	u	u	u	u

② PTC 口方向寄存器

DDRC		Address: D3H						
	Bit7	6	5	4	3	2	1	Bit0
Read:								
Write:								
Reset:								

Read:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
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

11.2.9 Port D

PTD 口是一个与 LCD 的 SEG32-SEG39 复用的 8 位并行端口。

① PTD 口数据寄存器

PTD		Address: BFH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
Write:								
Reset:	u	u	u	u	u	u	u	u

② PTD 口方向寄存器

DDRD		Address: D4H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
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

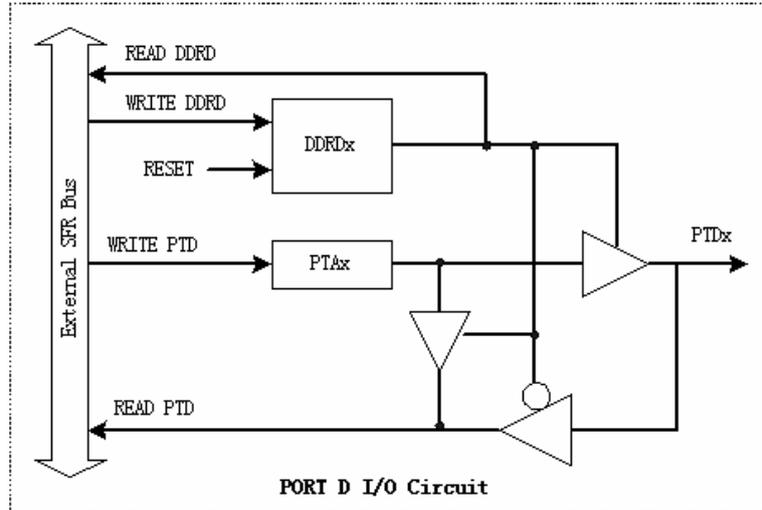


图 10-6

11.2.10 Port E

PTE 口是一个与 CC0, CC1, CC2, CC3, CLKOUT 复用的 5 位并行端口。

① PTE 口数据寄存器

PTE		Address: C2H						
	Bit7	6	5	4	3	2	1	Bit0
Read:				PTE4	PTE3	PTE2	PTE1	PTE0

Write:								
Reset:	u	u	u	u	u	u	u	u

② PTE 口方向寄存器

DDRE		Address: C3H						
	Bit7	6	5	4	3	2	1	Bit0
Read:				DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
Write:								
Reset:				0	0	0	0	0

以上的和 GPIO 相关的寄存器，包括第三优先级别的复位，(Debug Reset)在内的第二优先级的复位不能够复位该寄存器。(对 GPIO 部分所有的寄存器，下面的复位源不能复位该寄存器)。

12 定时器

12.1 概述

ATT7039 内部有三个 16 位可编程的定时器/计数器，定时器 T0、定时器 T1 和定时器 T2。ATT7039 既可作为定时器方式，又可作为计数器方式。

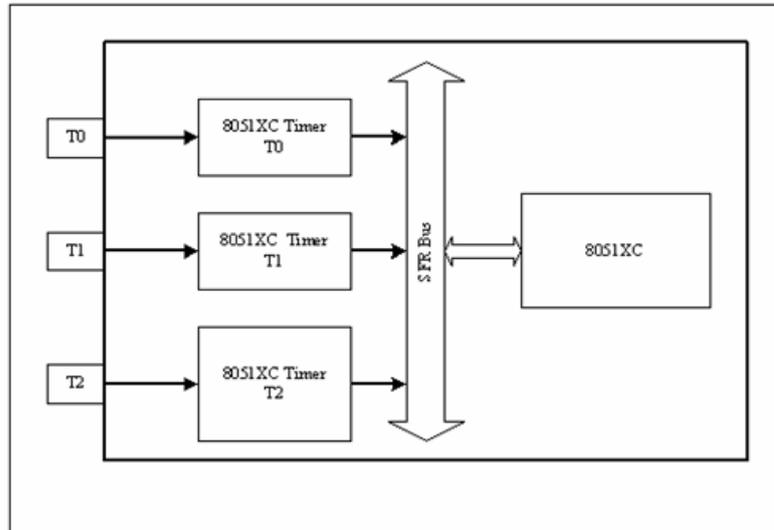


图 12-1 ATT7039 定时器/计数器框图

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

12.2 模块原理框图

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/计数器
- 模式 1: 16 位定时器/计数器
- 模式 2: 自动装入时间常数的 8 位定时器/计数器
- 模式 3: 1 个 8Bit 计数器和 1 个 8Bit 定时器
或 1 个 8bit 定时器和 1 个 8bit 定时器（只对 Timer0 有效）

12.2.1 模式 0/1

当 Timer0 作为 13 位定时器/计数器的时候，TL0 的低 3 位可以被忽略，不会发生变化。

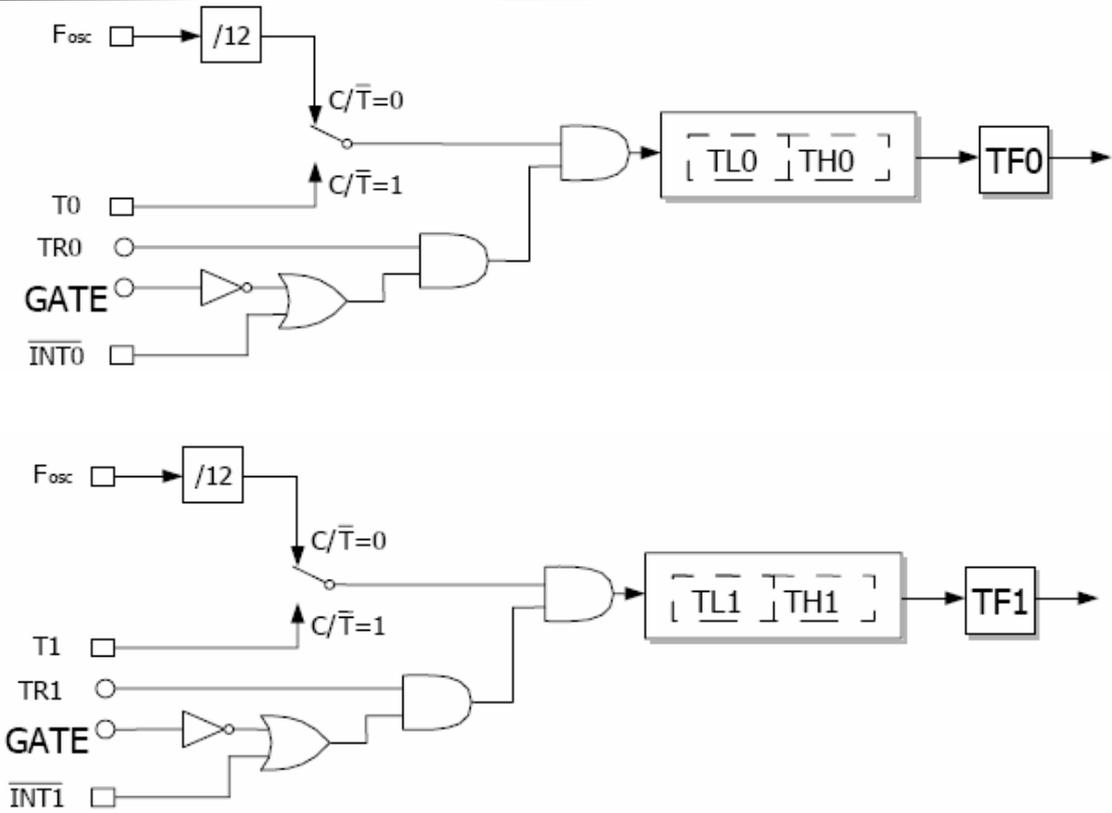


图 12-2

12.2.2 模式2

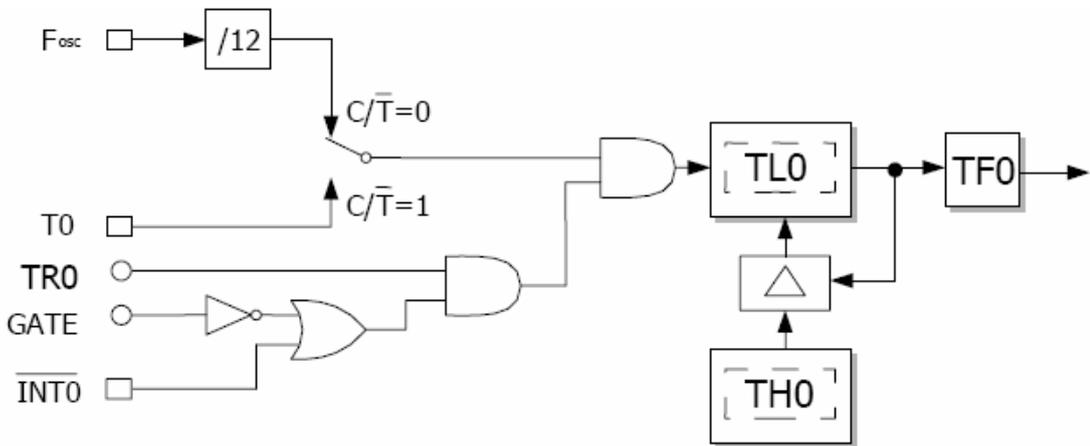


图 12-3

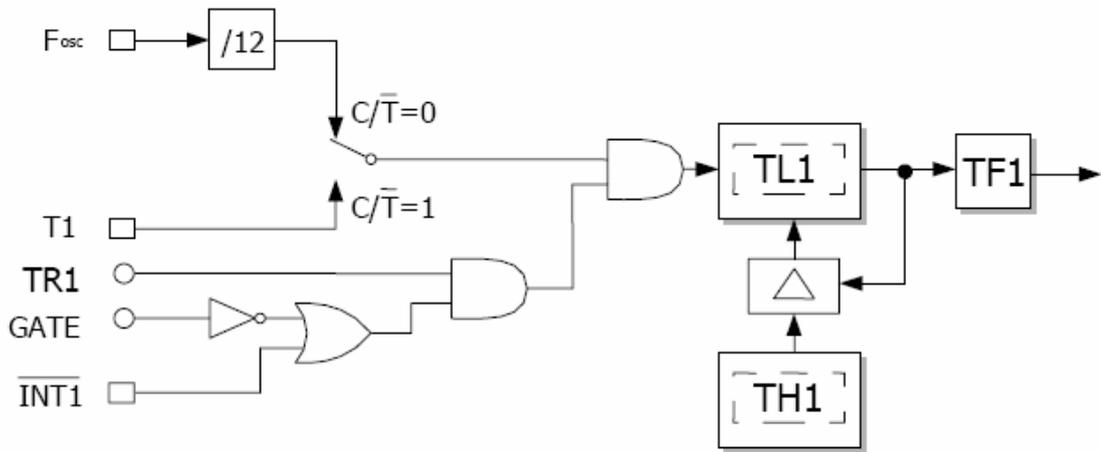


图 12-4

12.2.3 模式3

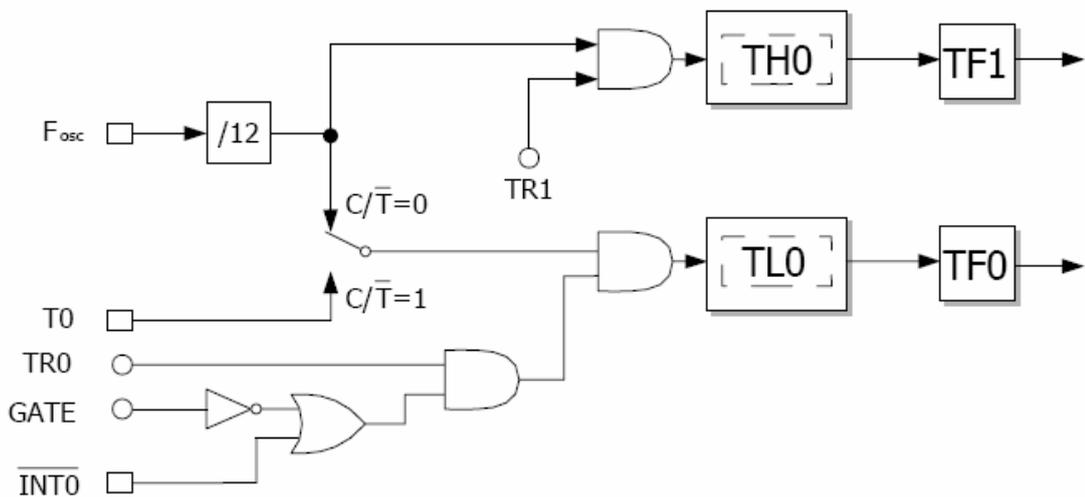


图 12-5

In mode 3 Timer 1 is stopped.

12.3 定时器 0 详细说明

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。

在计数器模式中，定时器 0 会检测 T0 pin 的信号，每个下降沿计数器加 1。计数器识别一个下降沿需要 2 个时钟周期，所以最大的输入计数频率是系统时钟的 1/2。虽然对时钟的占空比没有限制，但是为了确保能够识别到 0 或 1 的状态，输入信号应该至少保持 1 个系统时钟周期的稳定状态。

12.3.1 模式0

设置寄存器“tmod”的标志位 tmod[1:0]=00 可进入该模式。

在这种模式中，如果选择为定时器，寄存器中记录的是定时的输入时钟数；如果选择为计数器，寄存器中记录的是“t0”管脚的下降沿个数。清除标志位 tmod.2 可选择为定时器，反之选择为计数器。

定时器/计数器被分为两个 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（只有高 5 位是计数器的一部分）。这就使定时器/计数器成为一个 13 位的 counter，每 12 个时钟周期加 1，或者是外部输入信号“t0”的值由 1 变为 0 时加 1。当定时器/计数器 0 溢出时，“tcon.5”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。

定时器/计数器可被软件/硬件控制。标志位“tcon.4”可以控制定时器的停止（清 0）和运行（置 1）。如果“tmod.3”被使能，“int0”管脚的电平可以控制计数器的停止（低电平）和运行（高电平）。

12.3.2 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/计数器 0 是一个 16 位的计数器。

12.3.3 模式 2

设置寄存器“tmod”的标志位 tmod[1:0]=10 可进入该模式。

在这种模式中，如果选择为定时器，寄存器中记录的是定时的输入时钟数；如果选择为计数器，寄存器中记录的是“t0”管脚的下降沿个数。清除标志位 tmod.2 可选择为定时器，反之选择为计数器。

在这种模式下，只有低字节（“t10”）会每 12 个时钟周期加 1，或者是外部输入信号“t0”的值由 1 变为 0 时加 1。

在这种模式下，定时器/计数器是一个 8 位的重载定时器/计数器。当定时器/计数器 0 溢出时，“tcon.5”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”）到低字节（“t10”）中抓取。

定时器/计数器可被软件或者硬件控制。标志位“tcon.4”可以控制定时器的停止（清 0）和运行（置 1）。如果“tmod.3”被使能，“int0”管脚的电平可以控制计数器的停止（低电平）和运行（高电平）。

12.3.4 模式 3

设置寄存器“tmod”的标志位 tmod[1:0]=11 可进入该模式。

在这种模式中，如果选择为定时器，寄存器中记录的是定时的输入时钟数；如果选择为计数器，寄存器中记录的是“t0”管脚的下降沿个数。清除标志位 tmod.2 可选择为定时器，反之选择为计数器。

在这种模式下，低字节（“t10”）会每 12 个时钟周期加 1，或者是外部输入信号“t0”的值由 1 变为 0 时加 1。高字节（“th0”）会每 12 个时钟周期加 1。

当定时器/计数器低字节 0 溢出时，“tcon.5”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/计数器的高字节溢出时，“tcon[7]”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。

在这种模式中，定时器/计数器的低字节受“tcon.4”控制。高字节受“tcon.6”控制，当“tcon[6]”置位时，开始计数。

12.4 定时器 1 详细说明

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。

在计数器模式中，定时器 0 在检测到管脚“t1”有下降沿时加 1。因为它识别一个下降沿需要 2 个时钟周期，所以最大的输入计数频率是系统晶振频率的 1/2。虽然对时钟的占空比没有限制，但是为了确保能够识别到 0 或 1 的状态，输入信号应该至少保持 1 个系统时钟周期的稳定状态。

12.4.1 模式 0

设置寄存器“tmod”的标志位 tmod[5: 4]=00 可进入该模式。

在这种模式中，如果选择为定时器，寄存器中记录的是定时的输入时钟数；如果选择为计数器，寄存器中记录的是“t1”管脚的下降沿个数。清除标志位 tmod. 6 可选择为定时器，反之选择为计数器。

定时器/计数器 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（只有高 5 位是计数器的一部分）。这就使定时器/计数器 1 成为一个 13 位的 counter，每 12 个时钟周期加 1，或者是外部输入信号“t1”的值由 1 变为 0 时加 1。当定时器/计数器 1 溢出时，“tcon. 7”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。

定时器/计数器可被软件或者硬件控制。标志位“tcon. 6”可以控制定时器的停止（清 0）和运行（置 1）。如果“tmod. 7”被置位，“int1”管脚的电平可以控制计数器的停止（低电平）和运行（高电平）。

12.4.2 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/计数器 1 是一个 16 位的计数器。

12.4.3 模式 2

设置寄存器“tmod”的标志位 tmod[5: 4]=10 可进入该模式。

在这种模式中，如果选择为定时器，寄存器中记录的是定时的输入时钟数；如果选择为计数器，寄存器中记录的是“t1”管脚的下降沿个数。清除标志位 tmod. 6 可选择为定时器，反之选择为计数器。

在这种模式下，只有低字节（“t11”）会每 12 个时钟周期加 1，或者是外部输入信号“t1”的值由 1 变为 0 时加 1。

在这种模式下，定时器/计数器是一个 8 位的重载定时器/计数器。当低字节溢出时，“tcon[7]”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”）到低字节（“t11”）中抓取。

定时器/计数器可被软件或者硬件控制。标志位“tcon. 6”可以控制定时器的停止（清 0）和运行（置

1)。如果“tmod.7”被置位，“int1”管脚的电平可以控制计数器的停止（低电平）和运行（高电平）。

12.4.4 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）。

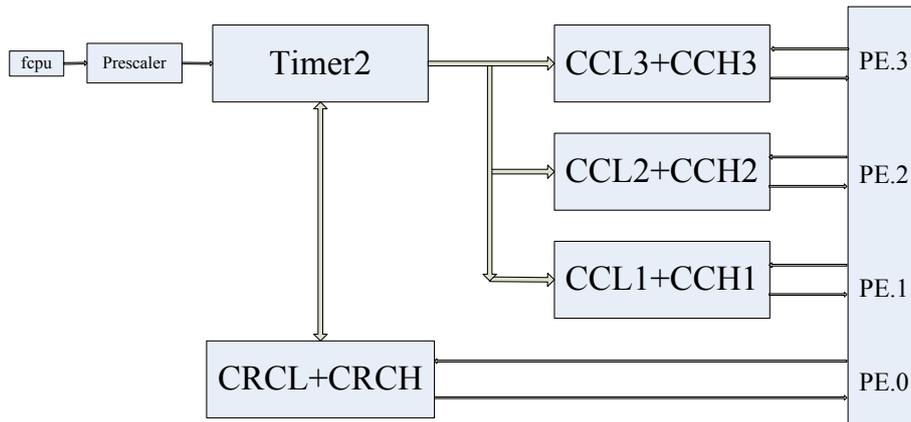
12.5 定时器 2 详细说明

12.5.1 概述

Timer2 是一个 16bit 的定时器/计数器，包含 CCU(Compare and Capture Unit)单元，可以实现比较/捕获功能。通过置寄存器 CCEN 来设置不同的比较/捕获模式。

12.5.2 接口描述

12.5.3 模块结构图



12.5.4 定时器/计数器功能

a) 定时器模式

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 个时钟周期 count 加 1；当标志位 t2ps=1 时，每 24 个时钟周期 count 加 1。

b) 计数器模式

设置寄存器 T2CON 的标志位 t2i0=0 和 t2i1=1 进入该模式。该模式中，定时器 2 对 t2 pin 的上升沿技术。最大计数率为时钟频率的 1/2。

c) 门控定时器模式

设置寄存器 T2CON 的标志位 t2i0=1 和 t2i1=1 进入该模式。与模式 a)相同，该模式中，有两个频率的时钟可以选择，每 12 或者 24 个时钟周期 count 加 1；同时受外部信号 t2 的控制。当 t2=0 时，Timer2 停止。

d) 重新载入

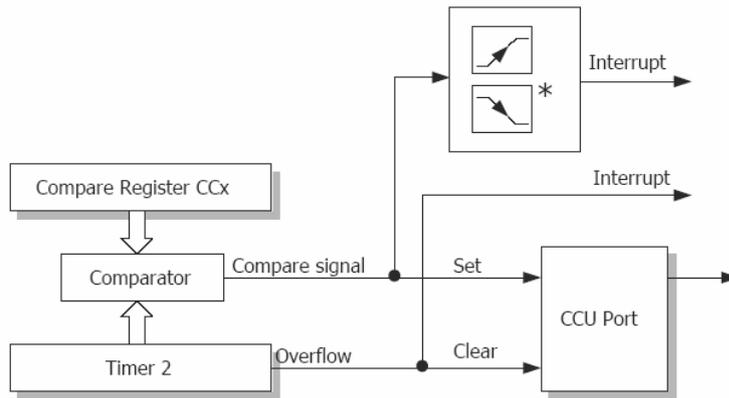
当 Timer2 溢出时，TH2/TL2 自动从 CRC 寄存器中载入 16 位数据。

12.5.5 比较器功能

CCU 单元由四组寄存器组成: crc、cc1、cc2、cc3, 每组寄存器都可以配置成比较器模式。在比较器模式中, 每组寄存器的值与 Timer2 的值比较, 比较的结果通过 PE 端口的低四位 (依次为 PE.0/PE.1/PE.2/PE.3) 输出。有两种比较模式, 可以通过寄存器 T2CON 的标志位 t2cm 选择。

a) 比较模式 0

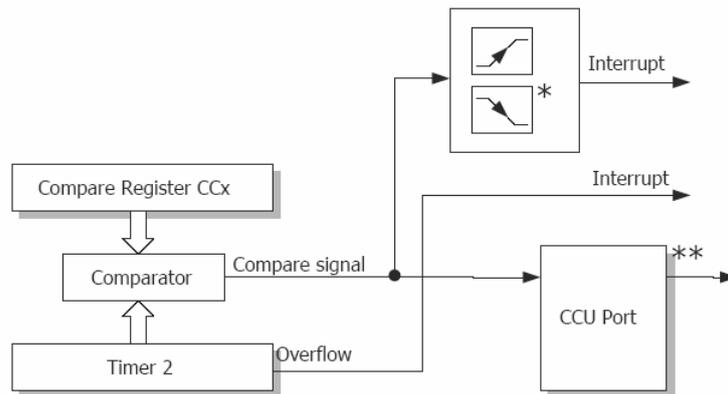
设置寄存器 T2CON 的标志位 t2cm=0 进入该模式。当 Timer2 的值和比较寄存器 crc/cc1/cc2/cc3 的值相同时, 比较输出由低电平转为高电平; 当 Timer2 溢出时, 比较输出又由高电平转为低电平。其中当 Timer2 的值和比较寄存器 crc 的值相同时, 还会触发中断请求 iex6(cc_intrrupt)。



* Only for CRC

b) 比较模式 1

设置寄存器 T2CON 的标志位 t2cm=1 进入该模式。该模式下, 寄存器位 PE.0/ PE.1/ PE.2/PE.3 可被软件赋值, 当 Timer2 的值和比较寄存器 crc/cc1/cc2/cc3 的值相同时, 寄存器位 PE.0/ PE.1/ PE.2/PE.3 的值被传送到外部接口。其中当 Timer2 的值和比较寄存器 crc 的值相同时, 还会触发中断请求 iex6(cc_intrrupt)。该模式下, timer2 的溢出, 不会改变比较器的外部输出信号。



* Only for CRC

** Shadow register is used

12.5.6 捕获器功能

CCU 单元的四组寄存器 crc、cc1、cc2、cc3 都可以配置成捕获器模式。在该模式中, 当有外部信号 (mode0) 或软件写操作 (mode1) 时, Timer2 的值将被保存到 CCU 寄存器中。

a) 捕获模式 0

以下条件满足时, 执行一次捕获操作:

输入信号 cc1 有上升沿 (Timer2 的值被捕获到寄存器 CCL1/CCH1 中)

输入信号 cc2 有上升沿 (Timer2 的值被捕获到寄存器 CCL2/CCH2 中)

输入信号 cc3 有上升沿 (Timer2 的值被捕获到寄存器 CCL3/CCH3 中)

输入信号 cc0 有上升/下降沿 (Timer2 的值被捕获到寄存器 CRCL/CRCH 中)

该模式下, 不会产生中断请求。

b) 捕获模式 1

当向每组 CCU 寄存器的低字节中写入任意值时, Timer2 的值将被捕获到对应的 CCU 寄存器中。该模式下, 不会产生中断请求。

12.6 特殊功能寄存器

表 12-1 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
0xCB	CRCH	0x00	自动装载寄存器高 8 位
0xC1	CCEN	0x00	定时器 2 比较/捕获使能寄存器
0xF6	CCL1	0x00	比较/捕获寄存器 1 低 8 位
0xF7	CCH1	0x00	比较/捕获寄存器 1 高 8 位
0xE6	CCL2	0x00	比较/捕获寄存器 2 低 8 位
0xE7	CCH2	0x00	比较/捕获寄存器 2 高 8 位
0xDE	CCL3	0x00	比较/捕获寄存器 3 低 8 位
0xDF	CCH3	0x00	比较/捕获寄存器 3 高 8 位

表 12-3 定时器 0、定时器 1 控制寄存器 (TCON 0x88H)

定时器 0、定时器 1 控制寄存器 (TCON)		Address: 88H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TCON.7	定时器1溢出中断标志 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。
TCON.6	定时器1运行控制位 清除该位，定时器1将停止运行
TCON.5	定时器0溢出中断标志 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。
TCON.4	定时器0运行控制位 清除该位，定时器0将停止运行
TCON.3	外部中断1标志 当外部中断int1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。
TCON.2	外部中断1类型控制位 TCON.2=1：外部中断1由输入引脚的下降沿触发； TCON.2=0：外部中断1由输入引脚的低电平触发；
TCON.1	外部中断0标志 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。
TCON.0	外部中断1类型控制位 TCON.0=1：外部中断1由输入引脚的下降沿触发； TCON.0=0：外部中断1由输入引脚的低电平触发；

表 12-2 定时器 0、定时器 1 模式寄存器(TMOD 0x89H)

定时器 0、定时器 1 模式寄存器 (TMOD)		Address: 89H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	GATE	C/\bar{T}	M1	M0	GATE	C/\bar{T}	M1	M0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMOD.7	定时器1门控位 GATE=1：只有在“int1”引脚为高电平和“tr1”=1时，定时器1才在“t1”引脚的下降沿计数； GATE=0：只要“tr1”=1时，定时器1就在“t1”引脚的下降沿计数；
TMOD.6	定时器1方式选择位 C/\bar{T} =1：计数器方式 C/\bar{T} =0：定时器方式
TMOD.5	定时器1模式控制位 如下表所示
TMOD.4	
TMOD.3	定时器0门控位

	GATE=1: 只有在“int0”引脚为高电平和“tr0”=1时, 定时器0才在“t0”引脚的下降沿计数; GATE=0: 只要“tr0”=1时, 定时器0就在“t0”引脚的下降沿计数;
TMOD.2	定时器0方式选择位 C/ \bar{T} =1: 计数器方式 C/ \bar{T} =0: 定时器方式
TMOD.1	定时器0模式控制位 如下表所示
TMOD.0	

M1	M0	方式	说明
0	0	0	13位计数器/定时器, TLO (TL1) 中的低5位和TH0(TH1)中的8位
0	1	1	16位计数器/定时器
1	0	2	8位计数器/定时器, 具有自动再装入功能, 装入在TLO(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: TLO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 2: 定时器 1 状态寄存器(8DH, TH1)

TH 1	Address: 8DH							
	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 2: 定时器 1 状态寄存器(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 2: 定时器 0 状态寄存器(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 2: 定时器 0 状态寄存器(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 12-5 定时器 2 控制寄存器 (T2CON 0xC8H)

定时器 2 控制寄存器 (T2CON)		Address: C8H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	T2PS	I3FR	I2FR	reserved	reserved	T2CM	T2I1	T2I0
Write:	x	x	x	x	x	x	x	x
Reset:	0	1	0	0	0	0	0	0

位	功能描述
T2CON.7	时钟选择位 t2ps = 0 – 定时器2的时钟选择为系统时钟的1/12; t2ps = 1 – 定时器2的时钟选择为系统时钟的1/24;
T2CON.6	捕获模式输入信号触发方式选择/外部中断“int3”触发方式选择 0 – 下降沿 1 – 上升沿
T2CON.5	外部中断“int2”触发方式选择 0 – 下降沿 1 – 上升沿
T2CON.4	保留
T2CON.3	保留
T2CON.2	定时器2比较模式选择 0 – 模式0 1 – 模式1
T2CON.1	定时器2输入选择(t2i1, t2i0) 00—定时器2中止工作
T2CON.0	01—输入时钟为f _{sys} /12 或者 f _{sys} /24 10—定时器2的计数由引脚“t2”的下降沿驱动 11—输入时钟由引脚“t2”确定, 为f _{sys} /12 或者 f _{sys} /24

表 2: 定时器 2 状态寄存器(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 2: 定时器 2 状态寄存器(0xCCH, 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。

定时器 2 比较/捕获使能寄存器 (0xC1, CCEN)

CCEN作为定时器2比较/捕获单元的功能配置寄存器，可以分别设置定时器2中四组比较器/捕获器的使能、触发模式。

位	介绍			类型
CCEN.7 CCEN.6	设置第3组寄存器CCL3/CCH3的比较/捕获模式			R/W
	CCEN.7	CCEN.6	描述	
	0	0	关闭比较/捕获功能	
	0	1	由cc3引脚的上升沿触发捕获	
	1	0	打开比较功能	
	1	1	向CCL3写操作以触发捕获	
CCEN.5 CCEN.4	设置第2组寄存器CCL2/CCH2的比较/捕获模式			
	CCEN.5	CCEN.4	描述	
	0	0	关闭比较/捕获功能	
	0	1	由cc2引脚的上升沿触发捕获	
	1	0	打开比较功能	
	1	1	向CCL2写操作以触发捕获	
CCEN.3 CCEN.2	设置第1组寄存器CCL1/CCH1的比较/捕获模式			
	CCEN.3	CCEN.2	描述	
	0	0	关闭比较/捕获功能	
	0	1	由cc1引脚的上升沿触发捕获	
	1	0	打开比较功能	
	1	1	向CCL1写操作以触发捕获	
CCEN.1 CCEN.0	设置第1组寄存器CRCL/CRCH的比较/捕获模式			
	CCEN.1	CCEN.0	描述	
	0	0	关闭比较/捕获功能	
	0	1	由cc0引脚的上升沿触发捕获	
	1	0	打开比较功能	
	1	1	向CRCL写操作以触发捕获	

表 2: 定时器控制寄存器(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 2: 定时器控制寄存器(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

比较/重载/捕获寄存器 CRCH/CRCL 是 16 位寄存器，对应定时器 2 的比较/重载/捕获工作模式；定时器 2 可同时提供 4 组比较/捕获器，除了上边的一组，其他三组比较/捕获寄存器分别为：CCL1、CCH1、CCL2、CCH2、CCL3、CCH3；其定义与 CRCH、CRCL 相同。

13 串口

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。

13.1 波特率产生方式

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 的方式 1、方式 3 与串口 1 的方式 A、方式 B 时，波特率是可变的。

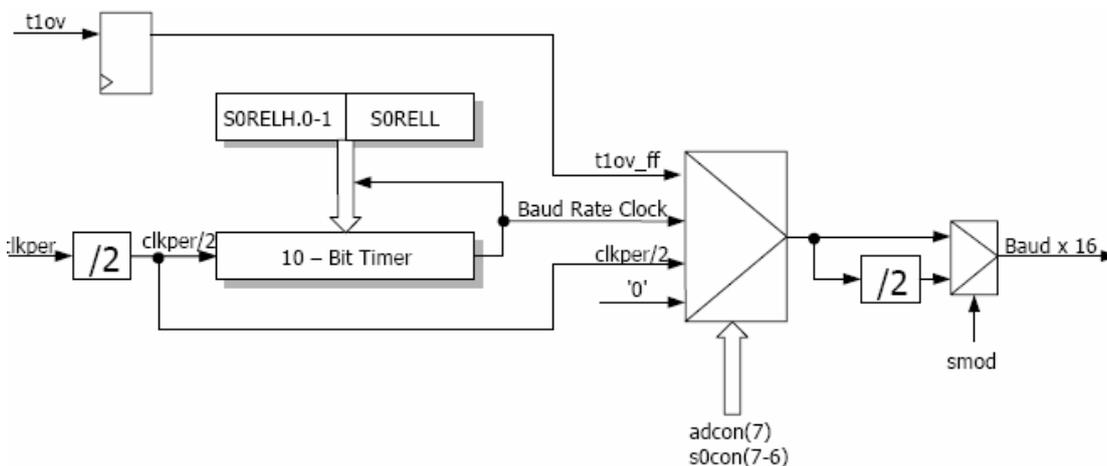
当 ADCON.7=1 时，波特率由波特率生成器和 SMOD 位的值确定：

$$\text{波特率} = \frac{2^{SMOD0} \times f_{cpu}}{64 \times (2^{10} - SOREL)}$$

其中 SMOD0 是 SFR PCON.7 的值；SOREL 是 10bit 无符号数，寄存器 SORELH 和 SORELL 的值；f_{cpu} 是系统时钟。

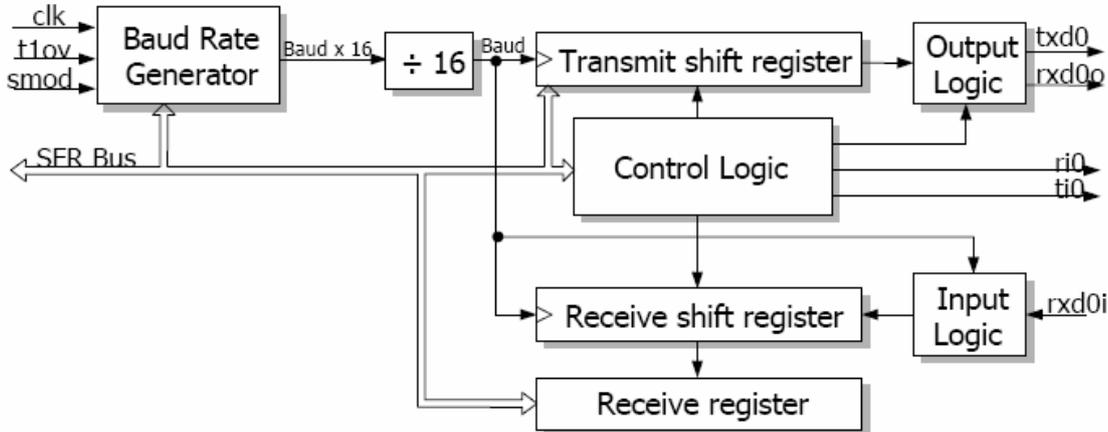
当 ADCON.7=0 时，波特率由定时器 1/定时器 2 的溢出速率和 SMOD 位的值确定：

$$\text{波特率} = \frac{2^{SMOD0}}{32} \times \text{Timer1 溢出速率}$$



13.2 串口 0

13.2.1 模块框图



13.2.2 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：



Figure 67. Serial0 transmission in mode 1

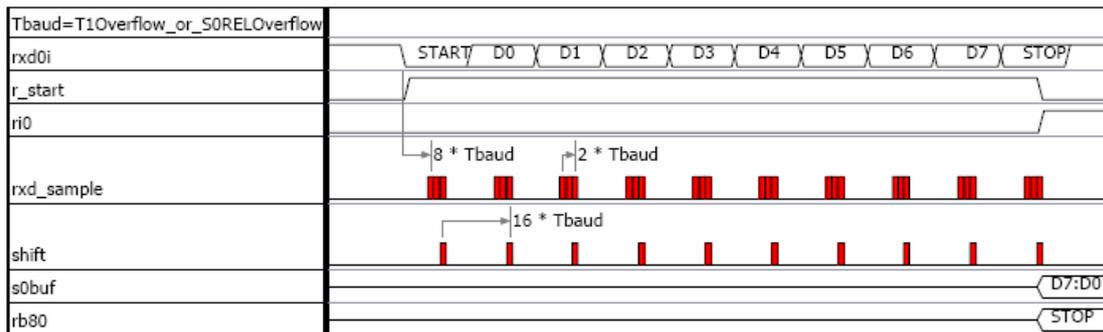


Figure 68. Serial0 reception in mode 1

图示：模式 1 时串口数据信息

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”。

13.2.3 模式 2

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：

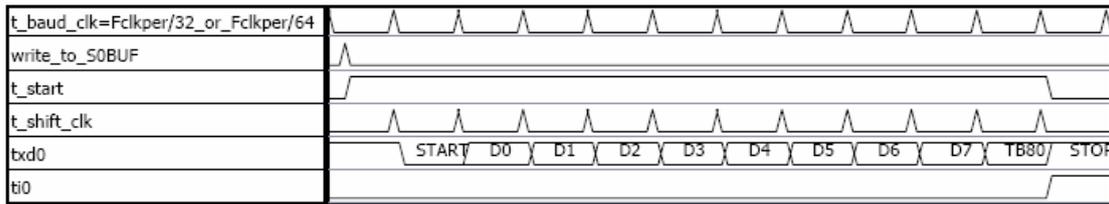


Figure 69. Serial0 transmission in mode 2

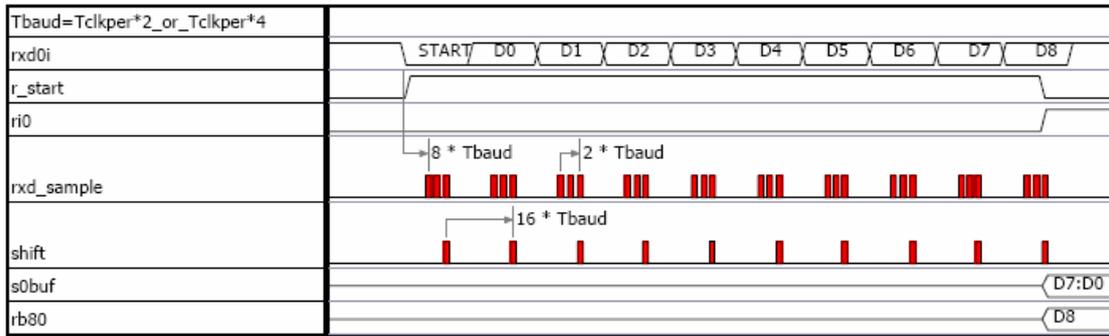


Figure 70. Serial0 reception in mode 2

图示：模式 2 时串口数据信息

在数据发送时，串口控制寄存器中的 TB8 作为第 9 位数据同时发送，发送完毕 TB8 被清除。当接收时，只有接收到第 9 位为 1 时才将串口的中断标志 RI 置“1”。按照模式 2 进行数据传送时不允许进行奇/偶校验，第 9 位数据可作为数据/地址标志位。

13.2.4 模式 3

模式 3 的数据传送方式和模式 2 相同，只是在数据接收时，无论第 9 位数据是 1 还是 0 都将串口的接收中断标志 RI 置“1”。按照模式 3 进行数据传送时允许奇偶校验，即当串口控制寄存器中的 REN=1 时，第 9 位数据为奇偶标志位。

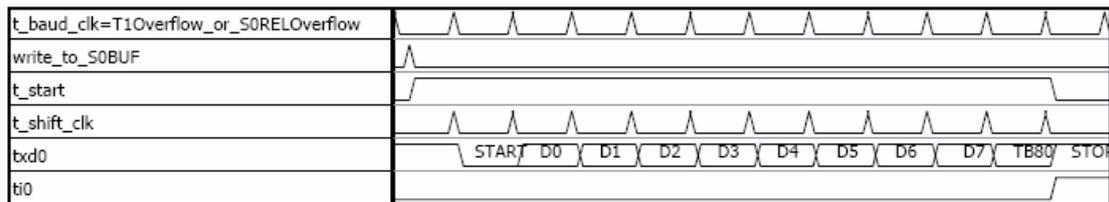


Figure 71. Serial0 transmission in mode 3

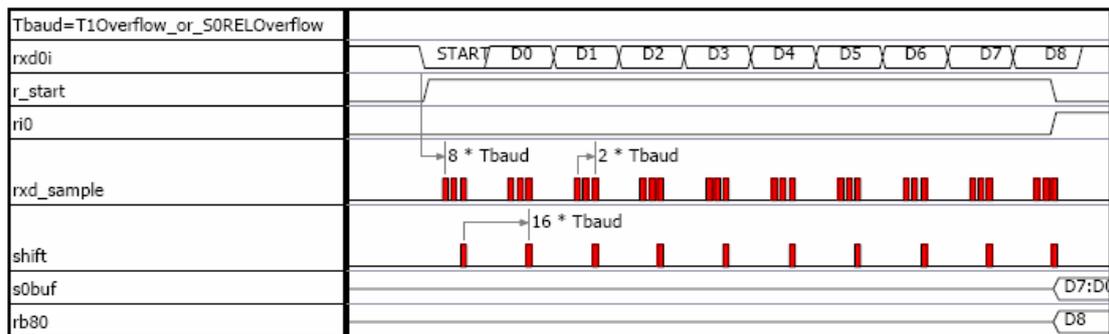


Figure 72. Serial0 reception in mode 3

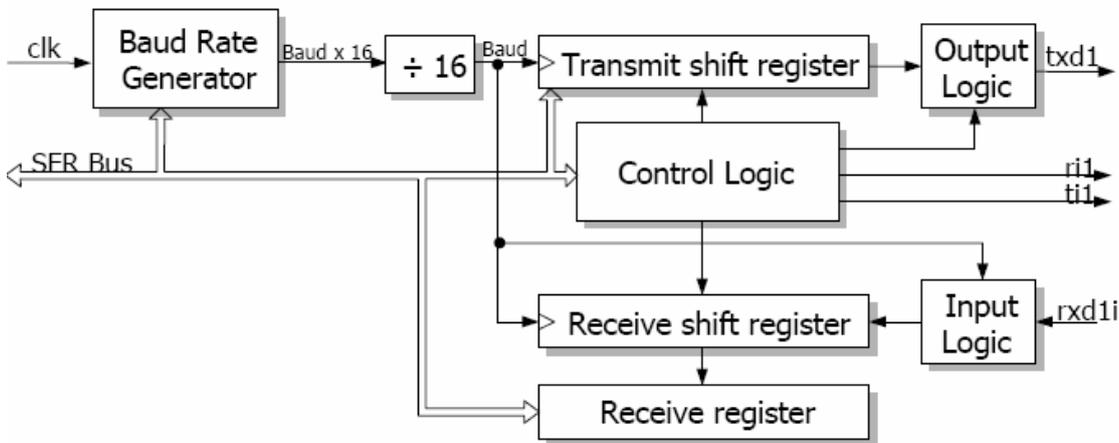
13.2.5 多处理器通讯

模式 2 和模式 3 具有多机通信的特殊功能。

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。在帧地址信息中，第 8 位为 1。从机的 SM2 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 清零，准备接收将传送的数据字节。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。当数据字节传送完毕后，被寻址的从机，其 RI 被激活，产生中断请求。响应中断后，由中断服务程序进行处理。

13.3 串口 1

13.3.1 模块框图



13.3.2 波特率产生方式

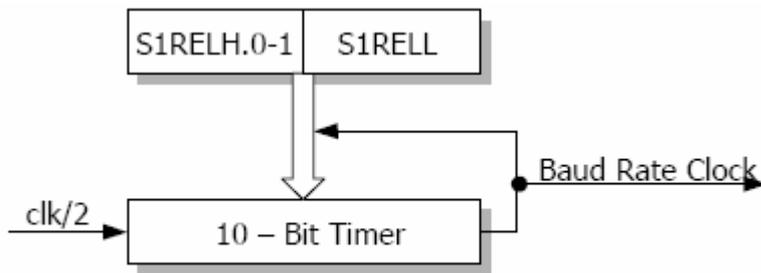


Figure 74. Serial1 Baud rate generation diagram

13.3.3 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“smod”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。

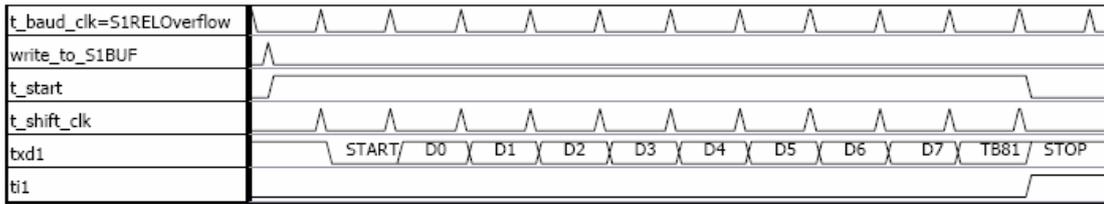


Figure 75. Serial1 transmission in mode A

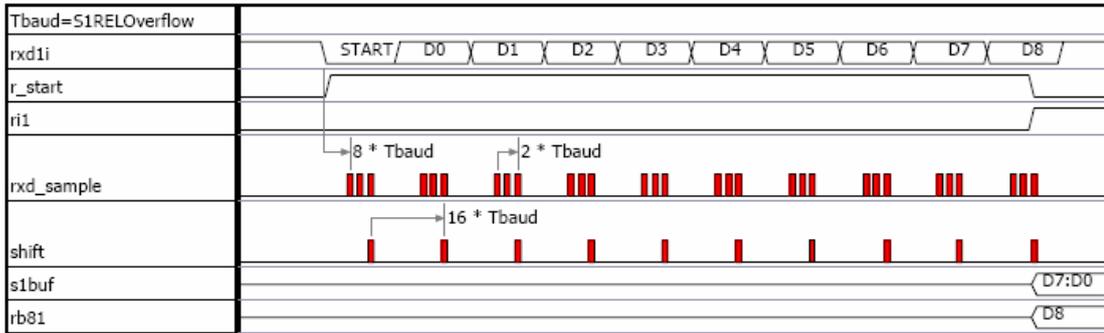


Figure 76. Serial1 reception in mode A

13.3.4 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“smod”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。

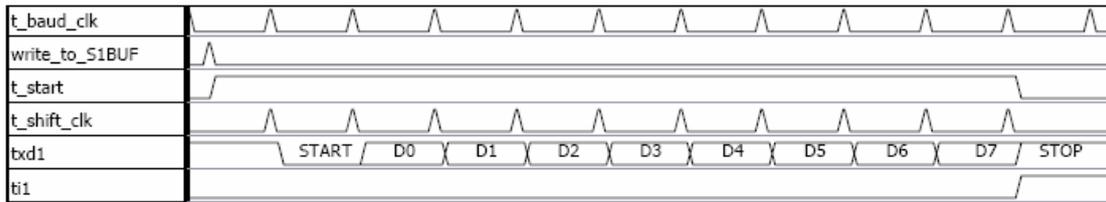


Figure 77. Serial1 transmission in mode B

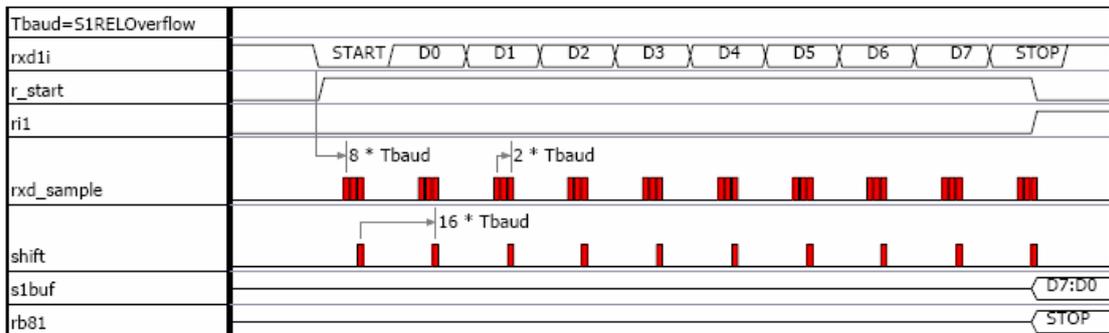


Figure 78. Serial1 reception in mode B

13.3.5 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。

13.4 特殊功能寄存器

表 13-7 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-见表 13-8
0x99	S0BUF	0x00	串口 0 数据缓冲寄存器
0x9b	S1CON	0x00	串口 1 控制寄存器-见表 13-9
0x9c	S1BUF	0x00	串口 1 数据缓冲寄存器
0x87	PCON	0x30	串口 0 波特率控制位-见表 13-10
0xD8	ADCON	0x40	串口 0 波特率控制位-见表 13-11
0xBA	S0RELH	0x00	串口 0 波特率发生器的重载值
0xAA	S0RELL	0x00	
0xBB	S1RELH	0x00	串口 1 波特率发生器的重载值
0x9D	S1RELL	0x00	
0xCE	P02CFG	0x00	P2P0 输出复用配置寄存器，写保护

表 13-8 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	TI_0	RI_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
SCON0.6	
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	发送数据位8 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	接收数据位8 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	发送中断标志 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	接收中断标志 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。

图 13-5

SM0_0	SM1_0	模式	描述	波特率
0	1	模式 1	8 位串行口	可变, 详细内容参考 11.3 波特率产生方式
1	0	模式 2	9 位串行口	可编程为系统频率的 1/32 或 1/64
1	1	模式 3	9 位串行口	可变, 详细内容参考 11.3 波特率产生方式

表 13-9 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:								
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 13-9 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		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
SCON1.7	串口1模式选择 sm = 0: 模式A sm = 1: 模式B
SCON1.6	保留
SCON1.5	多机通讯允许位
SCON1.4	允许接收位 由软件置位/复位; REN_1置位时允许接收。
SCON1.3	发送数据位8 由软件置位/复位; 在模式A中, 是被发送出去的第9数据位;

SCON1.2	接收数据位8 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；
SCON1.1	发送中断标志 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。
SCON1.0	接收中断标志 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 13-9 串口 1 缓冲寄存器(S1BUF 0x9CH)

串口 1 缓冲寄存器(S1BUF)		Address: 9CH						
	Bit7	6	5	4	3	2	1	Bit0
Read:								
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 13-9 Power Control Register(PCON 0x87H)

Register (PCON)		Address: 87H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	smod							
Write:								
Reset:	0	0	0	0	0	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。

表 13-9 Power Control Register(ADCON 0xD8H)

UART1 Control Register (ADCON)		Address: D8H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	bd							
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 11.3 波特率生成方式。

14 红外模块

14.1 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 下的外部唤醒通信功能。主要功能:

- 在ATT7039中能够单独的的实现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Baud Rate不超过2400bps)。

14.2 特殊功能寄存器

PWM Control Register (PWMCR)			Address: B1H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	CLKOUT1	CLKOUT0	IRTX1	IRTX0	PWMSEL	PWMCKS	PPOL	PWM_EN
Write:								
Reset:	0	0	0	0	0	0	0	0

IRTX1: Uart1 口的红外调制功能使能有效

IRTX1=1: 使能 Uart1 通讯口 TX1 功能 PIN 的红外调制电路模块。

IRTX1=0: 关闭 Uart1 通讯口 TX1 功能 PIN 的红外调制电路模块。

IRTX0: Uart0 口的红外调制功能使能有效

IRTX0=1: 使能 Uart0 通讯口 TX0 功能 PIN 的红外调制电路模块。

IRTX0=0: 关闭 Uart0 通讯口 TX0 功能 PIN 的红外调制电路模块。

注意: 只有 Prion=1, PLL 打开的情况下, 写 IRTX1=1, IRTX0=1 的红外调制使能功能才是有效的。(否则写这两位使能无效的, 红外调制没有时钟源)

15 电能计量

15.1 概述

ATT7039 提供单相电能计量所需要的全部功能，包括有功功率与有功电能、无功功率与无功电能、视在功率与视在电能、电压有效值、电流有效值及频率计算等，支持灵活的窃电方案和校表方案。

- 在动态范围 1000: 1 内有功误差小于 0.1%
- 无功精度远优于国标 2 级要求
- 有功功率与有功电能；无功功率与无功电能；视在功率与视在电能
- 电压、电流有效值
- 电压频率测量
- 三路 ADC 同步采样波形数据；有功功率、无功功率、视在功率波形数据
- 灵活的窃电方案，窃电阈值通过寄存器可调
- 灵活的潜动与启动方案
- 直流偏置自动校正
- 多种能量计算模式
- 开放快速脉冲计数寄存器，防止上下电时丢失电能
- 脉冲输出 PF/QF/SF 脉宽可选
- 支持单相三线制
- 过零中断检测信号
- 无功移相补偿
- 有效值偏置校正
- 有功功率 P，无功功率 Q 偏置校正

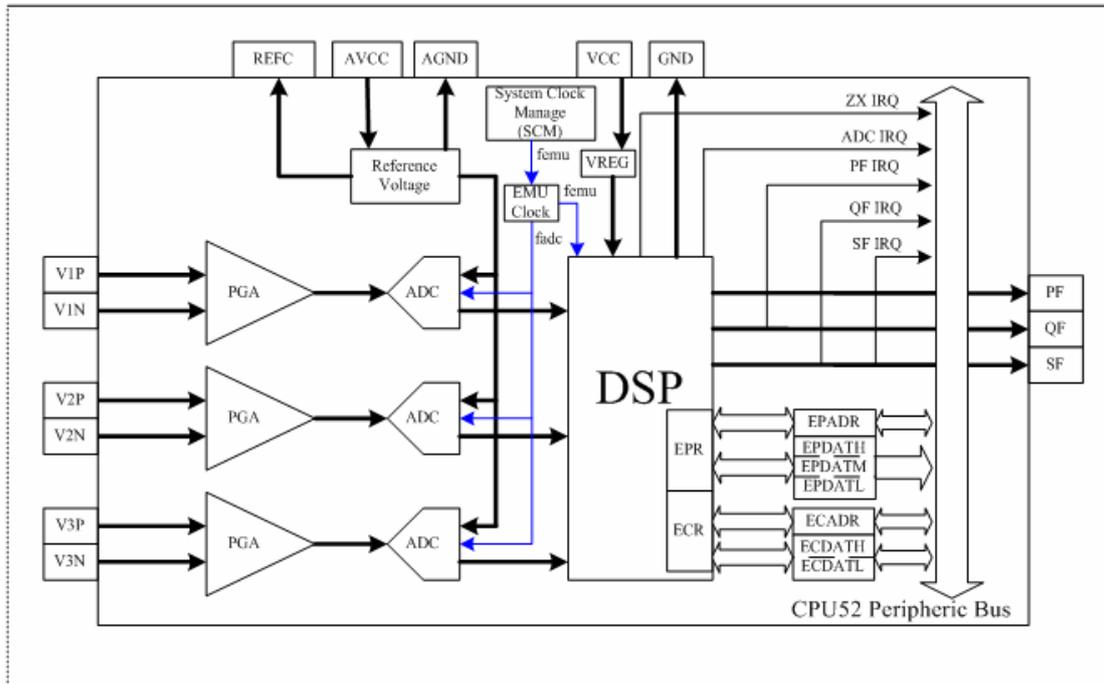


图 6-1 电能计量单元 (EMU)

15.2 功能描述

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

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

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。

15.2.1 模数转换器

ATT7039 有三路完全独立的两阶 Σ - Δ 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) 可以对三路 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) 可以对三路 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。

ADC2 (电流通道 2) 默认是关闭的，可以通过校表参数寄存器 EMU_Ctrl 单独选择关闭或者开启。

ADC2 同时提供增益校正寄存器 I2GAIN(4AH)，可以对 ADC2 的幅值进行比例缩放，可以用在自动防窃电等功能。

15.2.2 ADC 采样输出和功率波形输出

ATT7039 将三路 ADC 输出的 16bit 波形数据 Spl_I1(00H)、Spl_I2(01H)、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)、Spl_Q(04H)、Spl_S(05H) 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。

15.2.3 有功功率、无功功率和视在功率

ATT7039 提供有功功率、无功功率和视在功率输出寄存器。

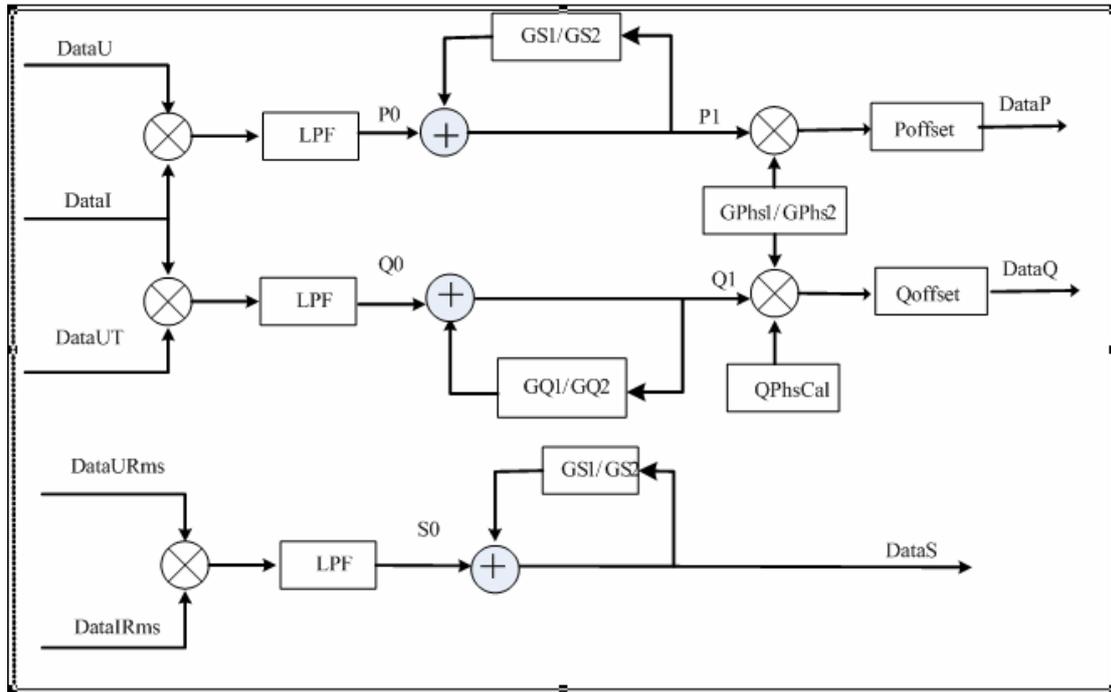


图 16-2 功率计算和补偿

有功功率通过电压、电流相乘经过低通滤波器后得到。

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

视在功率通过电压有效值与电流有效值相乘得到。

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

对电压移相 90 度的滤波器性能与输入信号的频率及 ADC 采样率相关，可以通过无功相位补偿寄存器 QphsCal(49H)对 90° 移相进行相位补偿。QphsCal默认为 0，在 femu 为 5.505024MHz 时，对应到 50Hz 输入信号可以实现准确的 90° 移相。

15.2.4 电压、电流有效值

ATT7039 同时输出两路电流和一路电压的有效值。

有效值可以保证 0.5% 的精度。

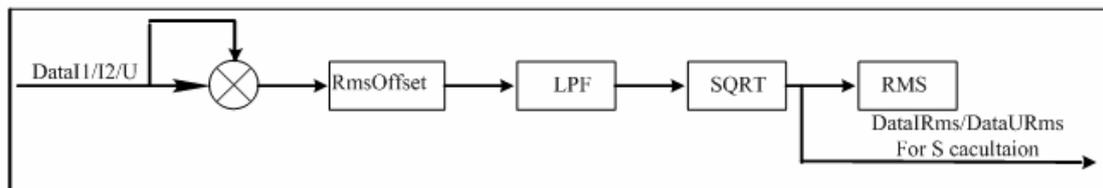


图 16-3

15.2.5 电压频率输出

通过对过零点计数的方式，提供电压频率输出。

频率的精度可以保证准确到 0.01Hz。

15.2.6 灵活的潜动/起动设置

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。

|P|/|Q|同时都小于PQStart时，SF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。

15.2.7 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。

这两位寄存器的更新与 PF、QF 的输出同步。

15.2.8 防窃电

可以通过防窃电模块对两路电流大小进行比较，选用较大的一路电流进行计量。

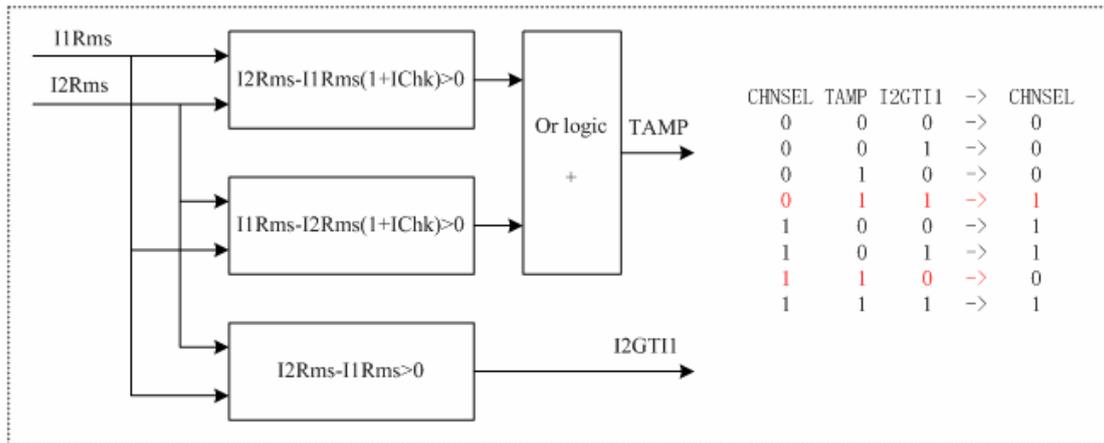


图 16-4 自动防窃电方案

可以通过 FLTON(52H.5) 设置是否开启自动防窃电功能。FLTON=0 时，用户可以根据 CHNSEL(51H.4) 进行通道选择；FLTON=1 时，防窃电单元根据用户的设置，自动选择相应的通道进行计量。

用户可以通过 CIADD(51H.3) 选择将两路电流先进行矢量加，然后再进行计量。

通过 ICHK(50H) 用户可以设置发生窃电的比例，比如可以设置 0x10H，表示两路电流有效值相差到 6.25% 时认为发生了窃电。

当两路的电流的有效值均小于 ITAMP(59H) 的值时，将默认选择通道 1 为计量单元，以防止小信号时噪声的干扰。

I2GTI1(51H.5) 为 0 表示 I1 大于 I2，为 1 时表示 I2 大于 I1。

TAMP(51H.6) 为 1 表示发生了窃电，即两路电流相比超过了设定的防窃电阈值。

自动防窃电设置步骤：

- 通过 EMU_Ctrl(5BH.5) 开启 ADC2 通道。
- 通过 I2GAIN(4AH) 对通道 2 的输出校正，保证同样的输入电流时，两个通道的有效值输出一致。
- 根据需要的防窃电阈值，设置 ICHK(50H)。
- 根据需要检测防窃电的最小电流，设置 ITAMP(59H)。
- 设置 FLTON(0x52H.5) 为 1，开启自动防窃电功能。

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

15.2.9 直流偏置自动校正和直流偏置校正寄存器

用户通过 CHNLCR(52H)的 HPFONU、HPFONI2、HPFONI1 可以分别控制电压和两路电流的高通滤波器环节是否加上。当这些位为 1 时，表示高通开启，反之高通关闭。

当高通滤波器关闭后，可以通过偏置校正寄存器 I1Off(4BH)、I2Off(4CH)、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_I2(01H)、SPL_U(02H)的多次平均值，设置相应的 OFFSET 值。

15.2.10 能量寄存器和脉冲输出单元

ATT7039 提供有功能量寄存器 ENERGY_P(0DH)、无功能量寄存器 ENERGY_Q(0EH)和视在能量寄存器 ENERGY_S(0FH)，同时提供相应的脉冲输出引脚 PF、QF 和 SF。

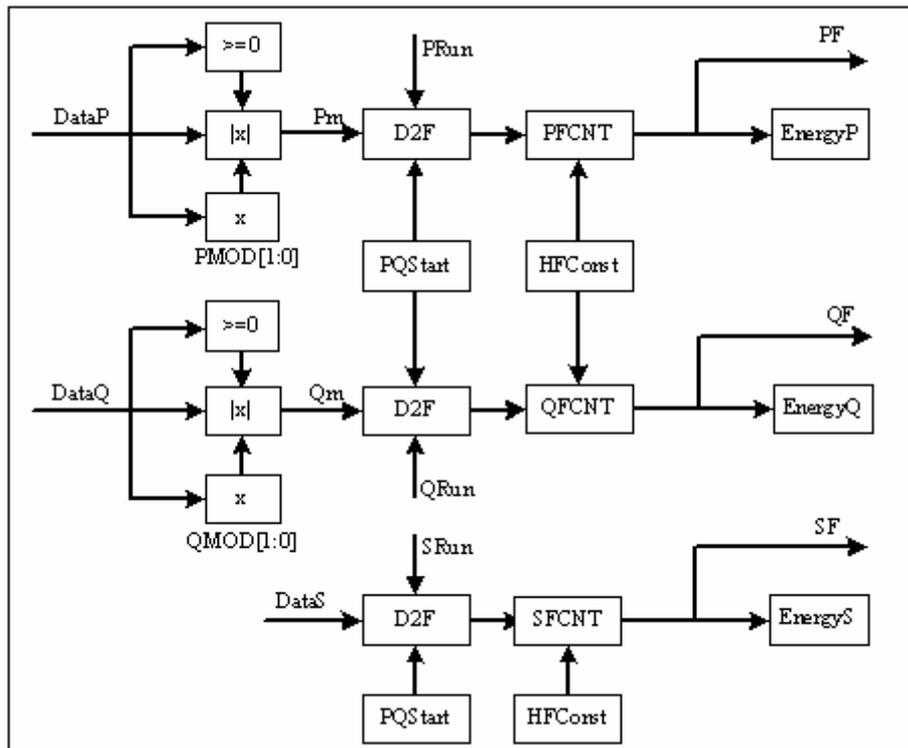


图 16-5 能量计量和脉冲输出

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)和 SFCNT(57H)。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/SFCNT，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/SF 和能量寄存器受到 EMCON(53H)的 PRun/QRun/SRun 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。

当 SRun=0 或者(|P|/|Q|同时都小于 PQStart)时，SF 不输出脉冲。

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

PF/QF/SF输出满足下面时序关系：

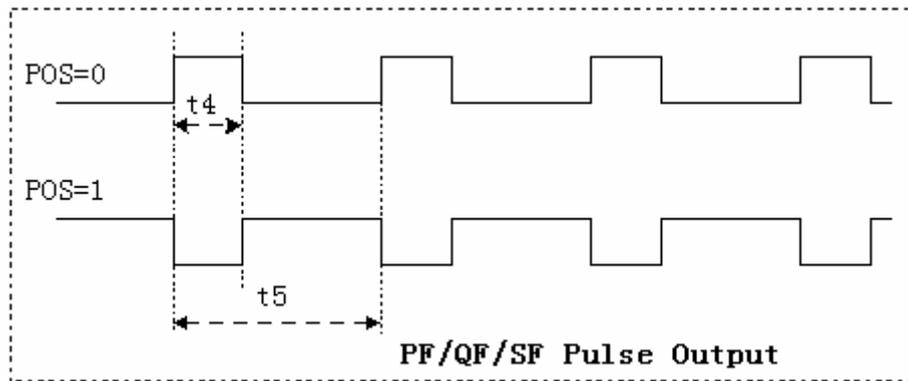


图 16-6 脉冲输出波形

时序特性：

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

注意：当脉冲输出周期小于 180ms 时，脉冲以等 duty 形式输出。

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。

15.2.11 中断系统

EMU 提供五个中断：PF_IRQ、QF_IRQ、SF_IRQ、ADC_IRQ 和 ZX_IRQ。

PF_IRQ：当有功脉冲输出时，EMU 向 CPU52 发出中断请求。

QF_IRQ：当有无功脉冲输出时，EMU 向 CPU52 发出中断请求。

SF_IRQ：当有视在脉冲输出时，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 五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。

15.3 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。

15.3.1 特殊功能寄存器

表 16-1 EMU SFR 寄存器列表

地址 (EPADR)	名称	字节长度	功能描述
0xE9	ECADR	1	EMU 校表地址寄存器
0xEA	ECDATH	1	EMU 校表高字节数据寄存器
0xEB	ECDATL	1	EMU 校表低字节数据寄存器
0xEC	EPADR	1	EMU 参数地址寄存器
0xED	EPDATH	1	EMU 参数高字节数据寄存器
0xEE	EPDATM	1	EMU 参数中字节数据寄存器
0xEF	EPDATL	1	EMU 参数低字节数据寄存器
0xAC	EMUIE	1	EMU 中断使能寄存器
0xB5	EMUIF	1	EMU 中断标志寄存器
0x9F	SUPDC	1	EMU 控制位(详细说明见 CMU SPEC)

表 16-1 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	TBS_EN	LCD_EN	EMU_EN	I2C_EN	SPI_EN		
Write:							X	X
Reset:	0	0	0	0	0	0	0	0

说明:

EMU_EN=1, Enable EMU 模块。EMU_EN=0, Disable EMU 模块。

注意:

EMU_EN 在 Battery 供电的情况下, EMU_EN=0, 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 16-2 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 16-3 ECDATH EMU Calibration Data High Byte Register (EAH)

EMU Calibration Data High Byte Register (ECDATH)		Address: EAH						
	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 16-4 EMU ECDATAL Calibration Data Low Byte Register (EBH)

EMU Calibration Data Low Byte Register (ECDATL)		Address: EBH						
	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 16-5 EPADR EMU Parameter Address Register (ECH)

EMU Parameter Address Register (EPADR)		Address: ECH						
	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 16-6 EPDATH EMU Parameter Data High Byte Register (EDH)

EMU Parameter Data High Byte Register (EPDATH)		Address: EDH						
	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 16-7 EPDATM EMU Parameter Data Middle Byte Register (EEH)

EMU Parameter Data Middle Byte Register (EPDATM)		Address: EEH						
	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 16-8 EPDATL EMU Parameter Data Low Byte Register (EFH)

EMU Parameter Data Low Byte Register (EPDATL)		Address: EFH						
	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 16-9 EMUIE EMU Interrupt Enable Register (ACH)

EMU Interrupt Enable Register (EMUIE)		Address: ACH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	PFIE	QFIE	SFIE	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。

EMU 中断产生，清除说明：

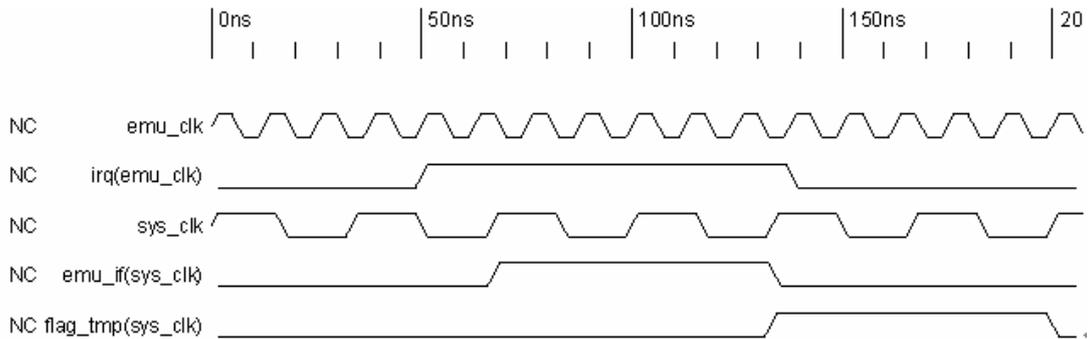


图 151-7 EMU 中断时序图 1

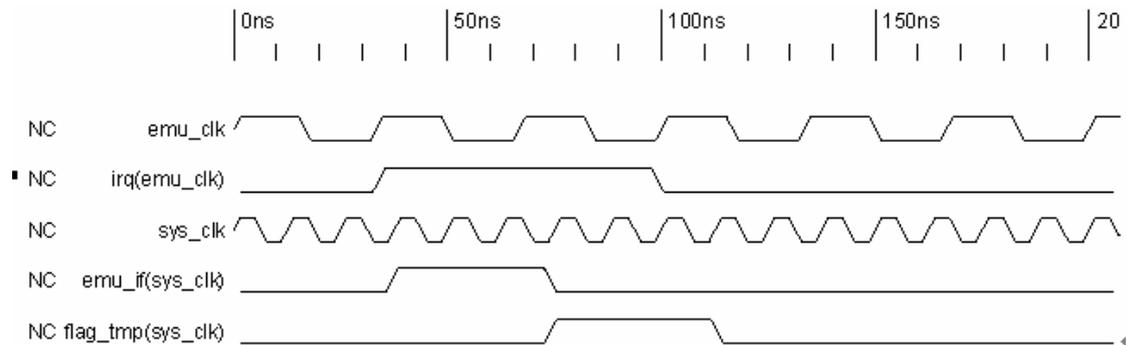


图 151-8 EMU 中断时序图 2

注意：

EMU 时钟频率大于 sys 时钟频率（EMU 中断时序图 1）/ EMU 时钟频率小于 sys 时钟频率（EMU 中断时序图 2）时，在 EMU 时钟域内产生中断 irq，irq 传递到 sys 时钟域产生 emu_if，如果 CPU 不响应中

断则所有信号保持；当 CPU 响应中断，写“0”清掉 emu_if，同时 flag_tmp 置“1”，当 irq 看到 flag_tmp 为“1”，irq 清“0”，如果 flag_tmp 看到 irq 为“0”，flag_tmp 清“0”。

表 16-10 EMUIF EMU Interrupt Flag Register (B5H)

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

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

15.3.2 间接寄存器

包括计量参数寄存器和校表参数寄存器，它们都是间接寄存器，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。

EPR 操作规则：

写地址寄存器 EPADR 时，相应地址的计量数据被更新。

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：

- 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；
- 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。

3. 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。

15.3.2.1 计量参数寄存器列表

表 16-12 EPR 寄存器列表(Read Only)

地址	名称	字节长度	功能描述
----	----	------	------

(EPADR)			
00H	Spl_I1	2	电流通道 1 的 ADC 采样数据
01H	Spl_I2	2	电流通道 2 的 ADC 采样数据
02H	Spl_U	2	电压通道的 ADC 采样数据
03H	Spl_P	3	有功功率波形数据
04H	Spl_Q	3	无功功率波形数据
05H	Spl_S	3	视在功率波形数据
06H	Rms_I1	3	电流通道 1 的有效值
07H	Rms_I2	3	电流通道 2 的有效值
08H	Rms_U	3	电压通道的有效值
09H	Freq_U	2	电压频率
0AH	Power_P	3	有功功率
0BH	Power_Q	3	无功功率
0CH	Power_S	3	视在功率
0DH	Energy_P	3	有功能量
0EH	Energy_Q	3	无功能量
0FH	Energy_S	3	视在能量
10H	Power_ppx	3	另一路有功功率
12H	Reserved	0	保留
13H	Reserved	0	保留
14H	Reserved	0	保留
15H	Reserved	0	保留
16H	Reserved	0	保留
17H	Reserved	0	保留
18H	Reserved	0	保留
19H	Reserved	0	保留
1AH	Reserved	0	保留
1BH	Reserved	0	保留
1CH	Reserved	0	保留
1DH	Reserved	0	保留
1EH	Reserved	0	保留
1FH	Reserved	0	保留

波形采样输出:

表 16-13 Spl_I1 Current Waveform Register (00H)

Current 1 Waveform Register (Spl_I1)			Address: 00H				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	SII_15	SII_14	SII_13	SII_12...SII_3	SII_2	SII_1	SII_0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

表 16-14 Spl_I2 Current Waveform Register (01H)

Current 2 Waveform Register (Spl_I2)			Address: 01H				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	SI2_15	SI2_14	SI2_13	SI2_12...SI2_3	SI2_2	SI2_1	SI2_0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

表 16-15 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 28kHz。

表 16-16 Spl_P Active Power Waveform Register (03H)

Active Power Waveform Register (Spl_P)			Address: 03H				
	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 16-17 Spl_Q Reactive Power Waveform Register (04H)

Reactive Power Waveform Register (Spl_Q)			Address: 04H				
	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

表 16-18 Spl_S Apparent Power Waveform Register (05H)

Apparent Power Waveform Register (Spl_S)			Address: 05H				
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	SS_23	SS_22	SS_21	SS_20...SS_3	SS_2	SS_1	SS_0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

功率波形采样值是二进制补码格式，24 位数据。其更新频率由 SPL（参见 ECR 寄存器 0x51H）确定。

最快可以到 28kHz。

有效值输出:

表 16-19 I1Rms Current 1 Rms Register (06H)

Current 1 Rms Register (I1Rms)		Address: 06H					
	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 16-20 I2Rms Current 2 Rms Register (07H)

Current 2 Rms Register (I2Rms)		Address: 07H					
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	I2S23	I2S22	I2S21	I2S20...I2S3	I2S2	I2S1	I2S0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

表 16-21 URms Voltage Rms Register (08H)

Voltage Rms Register (Urms)		Address: 08H					
	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。

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

$$I1rms = 65536 \times Kp = 1.3951A$$

有效值以 femu/1572864 的频率更新，其中 femu 为计量频率；例如，如果 femu=5.5MHz，那么有效值更新的频率为 3.5Hz，即每秒更新 3.5 次。

电压频率测量:

表 16-22 UFREQ Voltage Frequency Register (09H)

Voltage Frequency Register (UFREQ)		Address: 09H					
	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，如果计量频率选择 femu=5505024Hz，UFREQ=18350，那么测量到的实际频率为：

$$f = 5505024 / 6 / 18350 = 50.00\text{Hz}$$

电压频率测量值更新的周期为 0.7s。

功率参数输出：

表 16-23 PowerP Active Power Register (0AH)

Active Power Register (PowerP)		Address: 0AH					
	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 16-24 PowerQ Reactive Power Register (0BH)

Reactive Power Register (PowerQ)		Address: 0BH					
	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

表 16-25 PowerS Apparent Power Register (0CH)

Apparent Power Register (PowerS)		Address: 0CH					
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	SP23	SP22	SP21	SP20...SP3	SP2	SP1	SP0
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

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

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

设显示的有功功率为 P，转换系数为 KpqS

$$\text{则 } P = \text{Preg} \times KpqS$$

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：

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

其中 $\text{Preg} = \text{PowerP} - 2^{24} = -47820$

功率参数以 femu/1572864 的频率更新，其中 femu 为计量频率；例如，如果 femu=5.5MHz，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：

表 16-26 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 16-27 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

表 16-28 EnergyS Apparent Energy Register (0FH)

Apparent Energy Register(EnergyS)		Address: 0FH					
	Bit23	22	21	20 ... 3	2	1	Bit0
Read:	ES23	ES22	ES21	ES20...ES3	ES2	ES1	ES0
Write:	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0

电能参数是无符号数，EnergyP/EnergyQ/EnergyS 的寄存器值分别代表 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 16-29 PowerPPX PowerPPX Register (10H)

PPX Register		Address: 10H					
	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

不管防窃电打开还是关闭，该寄存器都为另一路电流的有功功率

15.3.2.2 校表参数寄存器列表

表 16-30 ECR 寄存器列表：(Read/Write)

地址 (ECADR)	名称	字节长 度	功能描述

40H	EMUSR	1	EMU 状态标志寄存器
41H	GP1	2	通道 1 的有功功率校正
42H	GQ1	2	通道 1 的无功功率校正
43H	GS1	2	通道 1 的视在功率校正
44H	GPhs1	2	通道 1 的相位校正
45H	GP2	2	通道 2 的有功功率校正
46H	GQ2	2	通道 2 的无功功率校正
47H	GS2	2	通道 2 的视在功率校正
48H	GPhs2	2	通道 2 的相位校正
49H	QPhsCal	2	无功相位补偿
4AH	I2Gain	2	电流通道 2 增益补偿
4BH	I1Off	2	电流通道 1 的偏置校正
4CH	I2Off	2	电流通道 2 的偏置校正
4DH	UOff	2	电压通道的偏置校正
4EH	PQStart	2	起动功率设置
4FH	HFConst	2	输出脉冲频率设置
50H	ICLK	1	窃电阈值设置
51H	ADCCFG	1	ADC 控制寄存器
52H	CHNLCR	1	通道控制信号
53H	EMCON	1	能量计量控制寄存器
54H	AutoDC	1	自动偏置校正
55H	PFCnt	2	快速有功脉冲计数
56H	QFCnt	2	快速无功脉冲计数
57H	SFCnt	2	快速视在脉冲计数
58H	ADCCON	1	ADC 通道增益选择
59H	ITAMP	2	窃电检测电流量值
5AH	DGAIN	1	通道数字增益
5BH	Emu_ctrl	1	Adc 单路控制信号, 过零中断控制信号, 时钟分频控制信号
5CH	P1OFFSET	1	通道 1 有功功率有效值校正参数, 为 8bit 补码
5DH	P2OFFSET	1	通道 2 有功功率有效值校正参数, 为 8bit 补码
5EH	Q1OFFSET	1	通道 1 无功功率有效值校正参数, 为 8bit 补码
5FH	Q2OFFSET	1	通道 2 无功功率有效值校正参数, 为 8bit 补码
60H	I1RMSOFFSET	1	通道 1 有效值补偿寄存器, 为 8bit 无符号数
61H	I2RMSOFFSET	1	通道 2 有效值补偿寄存器, 为 8bit 无符号数
62H	URMSOFFSET	1	电压通道 (通道 3) 有效值补偿寄存器, 为 8bit 无符号数
63H	Rosi_ctrl	1	罗氏线圈使能控制位
64H	UCONST	2	失压情况下参与计量的电压, 断相仿窃电

65H	Reserved	0	保留
-----	----------	---	----

表 16-30 EMUSR EMU Status Register(40H)

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

REVP: 反向有功功率指示标识信号, 当检测到负有功率时, 该信号为 1。当再次检测到正有功功率时, 该信号为 0。在 PF 发脉冲时更新该值。

REVQ: 反向无功功率指示标识信号, 当检测到负无功功率时, 该信号为 1。当再次检测到正无功功率时, 该信号为 0。在 QF 发脉冲时更新该值。

NoPLd: 当有功功率小于起动功率时, NoPLd 被置为 1; 当有功功率大于/等于起动功率时 NoPLd 清为 0。

NoQLd: 当无功功率小于起动功率时, NoQLd 被置为 1; 当无功功率大于/等于起动功率时 NoQLd 清为 0。

QEOF: 无功电能寄存器溢出标志, 写 0 清零

PEOF: 有功电能寄存器溢出标志, 写 0 清零

SEOF: 视在电能寄存器溢出标志, 写 0 清零

功率校正:

表 16-31 GP1 Active Power Gain 1 Register(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 16-32 GQ1 Reactive Power Gain 1 Register(42H)

Reactive Power Gain 1 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 16-33 GS1 Apparent Power Gain 1 Register(43H)

Apparent Power Gain 1 Register (GS1)			Address: 43H				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	GS1_15	GS1_14	GS1_13	GS1_12...GS1_3	GS1_2	GS1_1	GS1_0
Write:							
Reset:	0	0	0	0	0	0	0

表 16-34 GPhs1 Phase Calibration 1 Register(44H)

Phase Calibration 1 Register (GPhs1)			Address: 44H				
	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 16-35 Active Power Gain 2 Register(GP2 0x45)

Active Power Gain 2 Register (GP2)			Address: 45H				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	GP2_15	GP2_14	GP2_13	GP2_12...GP2_3	GP2_2	GP2_1	GP2_0
Write:							
Reset:	0	0	0	0	0	0	0

表 16-36 GQ2 Reactive Power Gain 2 Register(46H)

Reactive Power Gain 2 Register (GQ2)			Address: 46H				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	GQ2_15	GQ2_14	GQ2_13	GQ2_12...GQ2_3	GQ2_2	GQ2_1	GQ2_0
Write:							
Reset:	0	0	0	0	0	0	0

表 16-37 GS2 Apparent Power Gain 2 Register(47H)

Apparent Power Gain 2 Register (GS2)			Address: 47H				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	GS2_15	GS2_14	GS2_13	GS2_12...GS2_3	GS2_2	GS2_1	GS2_0
Write:							
Reset:	0	0	0	0	0	0	0

表 16-38 GPhs2 Phase Calibration 2 Register(48H)

Phase Calibration 2 Register (GPhs2)			Address: 48H				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	GPS2_15	GPS2_14	GPS2_13	GPS2_12...GPS2_3	GPS2_2	GPS2_1	GPS2_0
Write:							
Reset:	0	0	0	0	0	0	0

0x41H~0x48H寄存器都是二进制补码格式，最高位为符号位。具体的定义参见 校表过程

表 16-39 QPhsCal Reactive Power Phase Calibration Register(49H)

Reactive Power Phase Calibration Register (QPhsCal)			Address: 49H					
	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也采用二进制补码形式，最高位为符号位。

femu 为 5.5MHz 时，该寄存器写入 0。当 femu 为 2.7MHz 时，该寄存器写入 80H。

用户通过标准表在 0.5L 下测得无功精度误差为 Err，通过如下公式计算得到 Qphscal 值：

$$\text{Err} = \text{Qphscal} / (2^{15} * 1.732)$$

电流通道 2 增益设置：

表 16-40 I2Gain Current 2 Gain Register(4AH)

Current 2 Gain Register (I2Gain)			Address: 4AH				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	I2G15	I2G14	I2G13	I2G12...I2G3	I2G2	I2G1	I2G0
Write:							
Reset:	0	0	0	0	0	0	0

通道 2 电流增益寄存器采用二进制补码形式，最高位为符号位。定义参见 校表过程。

通道直流偏置校正寄存器：

表 16-41 I1Off Current 1 Offset Register(4BH)

Current 1 Offset Register (I1Off)			Address: 4BH				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	I1OS15	I1OS14	I1OS13	I1OS12...I1OS3	I1OS2	I1OS1	I1OS0
Write:							
Reset:	0	0	0	0	0	0	0

表 16-42 I2Off Current 2 Offset Register (4CH)

Current 2 Offset Register (I2Off)			Address: 4CH				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	I2OS15	I2OS14	I2OS13	I2OS12...I2OS3	I2OS2	I2OS1	I2OS0
Write:							
Reset:	0	0	0	0	0	0	0

表 16-43 UOff Voltage Offset Register(4DH)

Voltage Offset Register (UOff)			Address: 4DH				
	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 16-44 PQStart Start Power Threshold Setup Register(4EH)

Start Power Threshold Setup Register (PQStart)			Address: 4EH					
	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 16-45 HFConst High Frequency Impulse Const Register(4FH)

High Frequency Impulse Const Register (HFConst)			Address: 4FH					
	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。

窃电阈值设置:

表 16-46 IChk Check Current Rms Register(50H)

Check Current Rms Register (IChk)			Address: 50H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	ICK7	ICK6	ICK5	ICK4	ICK3	ICK2	ICK1	ICK0
Write:								
Reset:	0	0	0	1	0	0	0	0

窃电阈值电流寄存器采用二进制补码形式，表示范围(0,+1)。

$$ICK = ICK7 * 2^{-1} + ICK6 * 2^{-2} + ICK5 * 2^{-3} + \dots + ICK2 * 2^{-6} + ICK1 * 2^{-7} + ICK0 * 2^{-8}$$

默认为: 0.0625 也即 6.25%。

开启自动防窃电后，电流 1 和电流 2 两者相差比超过窃电阈值电流值，则自动选择大的电流值参与功率计量，同时 TAMP=1。如果电流 2 大于电流 1，则将标志位 I2GTI1 置为 1，否则标志位 I2GTI1 为 0。

ADC 采样:

表 16-47 ADCCFG ADC Config Register(51H)

ADC Config Register (ADCCFG)		Address: 51H	

	Bit7	6	5	4	3	2	1	Bit0
Read:	*	TAMP	I2GTI1	CHNSEL*	CIADD*	SPL2	SPL1	SPL0
Write:		X	X					
Reset:	0	0	0	0	0	0	0	0

TAMP: =1 表示发生窃电, $I1Rms > I2Rms * (1 + IChk)$ 或者 $I2Rms > I1Rms * (1 + IChk)$ 。

=0 表示未发生窃电, $I1Rms$ 与 $I2Rms$ 相差不到设定的 $IChk$ 范围。

I2GTI1: =1 表示 $I2Rms > I1Rms$; =0 表示 $I2Rms \leq I1Rms$ 。

CIADD: =1 表示两路电流相加模式, 与 CHNSEL 配合使用。电流相加模式下, 采用通道一的校表数据。在 $FLTON=1$ 时, 即开启防窃电模块时, CIADD 只读; 只有 $FLTON=0$ 时 CIADD 可读写。

CHNSEL: =0 表示当前选择通道一作为电流输入通道。

=1 表示当前选择通道二作为电流输入通道。

当 $FLTON=1$ 时, 即防窃电开启时, CHNSEL 的状态由防窃电比较器决定, CHNSEL 是一个只读状态位, 不能软件修改。

当 $FLTON=0$ 时, 即防窃电关闭时, CHNSEL 位和 CIADD 位可以进行读写, 满足以下逻辑关系。

表 15-2 anti-tap select

FLTON	CIADD	CHNSEL	工作模式	PPX
1	只读(恒为 0)	只读(内部决定)	防窃电模式	防窃电模式, 与正常工作模式有效电流输入相反
0	0	0	通道一为有效电流输入	通道一为有效电流输入
0	0	1	通道二为有效电流输入	通道二为有效电流输入
0	1	x	电流相加模式	电流相加模式

SPL[2:0]: 波形采样中断频率选择, 当 $femu=5.505024MHz$ 时, 选择的频率如下:

表 16-3 wave sample frequency select register

SPL2	SPL1	SPL0	波形采样频率
0	0	0	1.8k Hz ($femu/3072$)
0	0	1	3.6k Hz ($femu/1536$)
0	1	0	7.2k Hz ($femu/768$)
0	1	1	14.4k Hz ($femu/384$)
1	x	x	28.8k Hz ($femu/192$)

当 $femu=2.7648MHz/1.3824MHz/0.6912MHz$ 时, 选择的波形采样频率与上表相比等比例调整

即可。

通道设置:

表 16-48 CHNLCCR Current Channel Control Register(52H)

Current Channel Control Register (CHNLCCR)			Address: 52H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PPXEN	POS	FLTON	X	X	HPFONU	HPFONI2	HPFONI1
Write:								
Reset:	0	0	0	0	0	1	1	1

POS:

POS=0: 表示 PF/QF/SF 为高电平有效;

POS=1: 表示 PF/QF/SF 为低电平有效。

FLTON:

FLTON=0: 关闭防窃电处理模块

FLTON=1: 开启防窃电处理模块, 电流输入通道选择由防窃电模块决定。

HPFONI2/I1/U:

HPFON=1: 启动数字高通滤波器

HPFON=0: 关闭数字高通滤波器

能量计算:

表 16-49 EMCON Energy Measure Control Register(53H)

Energy Measure Control Register (EMCON)			Address: 53H					
	Bit7	6	5	4	3	2	1	Bit0
Read:		SRun	QRun	PRun	QMOD1	QMOD0	PMOD1	PMOD0
Write:								
Reset:	0	1	1	1	0	0	0	0

PMOD[1:0]: 有功能量累加方式选择

PMOD1	PMOD0	累加功率 Pm
0	0	Pm=DataP
0	1	DataP ≥ 0, Pm=DataP; DataP < 0, Pm=0
1	0	Pm= DataP
1	1	Pm=DataP

QMOD[1:0]: 无功能量累加方式选择

QMOD1	QMOD0	累加功率 Qm
0	0	Qm=DataQ
0	1	DataQ ≥ 0, Qm=DataQ; DataQ < 0, Qm=0
1	0	Qm= DataQ
1	1	Qm=DataQ

能量计算模式: 仅正向计量、绝对值计量、代数和计量等

PRun: 有功能量累加使能

PRun=0:停止计量; PRun=1: 允许计量
 QRun: 无功能量累加使能
 QRun=0:停止计量; QRun=1: 允许计量
 SRun: 视在能量累加使能
 SRun=0:停止计量; SRun=1: 允许计量

直流偏置校正:

表 16-50 AutoDC Auto Offset Calibration Register(54H)

Auto Offset Calibration Register (AutoDC)			Address: 54H					
	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 16-51 PFCNT Active Energy Counter Register(55H)

Active Energy Counter Register (PFCNT)			Address: 55H				
	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 16-52 QFCNT Reactive Energy Counter Register(56H)

Reactive Energy Counter Register (QFCNT)			Address: 56H				
	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

表 16-53 SFCNT Apparent Energy Counter Register(57H)

Apparent Energy Counter Register (SFCNT)			Address: 57H				
	Bit15	14	13	12 ... 3	2	1	Bit0
Read:	SFC15	SFC14	SFC13	SFC12...SFC3	SFC2	SFC1	SFC0
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 16-54 ADCCON ADC Channel Gain Register(58H)

ADC Channel Gain Register (ADCCON)	Address: 58H							
	Bit7	6	5	4	3	2	1	Bit0
Read:	CFP1	CFP0	PGA3	PGA2	PGA1	PGA0	UPGA1	UPGA0
Write:								
Reset:	0	0	1	0	1	0	0	0

CFP[1:0]: 脉宽选择寄存器, 即脉冲输出的 t4 参数, 见 PF/QF/SF 时序特性。

如果 femu=5.505024MHz, 那么:

CFP[1:0]	00	01	10	11
t4	90ms	90/2=45ms	90/4=22.5ms	90/8=11.25ms

如果 femu=2.7648MHz, 那么:

CFP[1:0]	00	01	10	11
t4	180ms	180/2=90ms	180/4=45ms	180/8=22.5ms

如果 femu=1.3824MHz, 那么:

CFP[1:0]	00	01	10	11
t4	360ms	360/2=180ms	360/4=90ms	360/8=45ms

如果 femu=0.6912MHz, 那么:

CFP[1:0]	00	01	10	11
t4	720ms	720/2=360ms	720/4=180ms	720/8=90ms

UPGA[1:0]:

UPGA1	UPGA0	电压通道增益
0	0	PGA=1
0	1	PGA=2
1	0	PGA=8
1	1	PGA=16

PGA[3:0]:

PGA1	PGA0	电流通道 1	PGA3	PGA2	电流通道 2
0	0	PGA=1	0	0	PGA=1
0	1	PGA=2	0	1	PGA=2
1	0	PGA=8	1	0	PGA=8

1	1	PGA=16	1	1	PGA=16
---	---	--------	---	---	--------

注：这里的电流、电压通道增益指的是 ADC 模拟部分的通道增益。

窃电检测电流阈值：

表 16-55 ITAMP Tamper Current Register(59H)

Tamper Current Register (ITAMP)		Address: 59H					
	Bit15	14	13	12...3	2	1	Bit0
Read:	ITAMP15	ITAMP14	ITAMP13	ITAMP12...ITAMP3	ITAMP2	ITAMP1	ITAMP0
Write:							
Reset:	0	0	0	0	0	0	0

格式同电流有效值寄存器，ITAMP[15:0]是高 16 位。防窃电处理模块开启时，当通道 1 和 2 的电流有效值都低于 ITAMP 时，始终选择通道 1 作为有效输入，TAMP、I2GTI1 和 CHNSEL 均为 0。

通道数字增益：

表 16-56 DGAIN Channel Digital Gain Register(5AH)

Channel Digital Gain Register (DGAIN)			Address: 5AH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	DGI3	DGI2	DGI1	DGI0	DGU1	DGU0
Write:	X	X						
Reset:	0	0	0	0	0	0	0	0

该寄存器类同通道增益选择，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。

DGU[1:0]:

DGU 1	DGU 0	电压数字增益
0	0	DG=1
0	1	DG=2
1	0	DG=4
1	1	DG=8

DGI[3:0]:

DGI1	DGI0	电流通道 1 数字增益	DGI3	DGI2	电流通道 2 数字增益
0	0	DG=1	0	0	DG=1
0	1	DG=2	0	1	DG=2
1	0	DG=4	1	0	DG=4
1	1	DG=8	1	1	DG=8

表 16-47 emu_ctrl clk and adc control (5BH)

clk and adc control Register (emu_ctrl)			Address: 5BH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	Apci_ctrl	Adc_i1on	Adc_i2on	Adc_uon	Zxd1	Zxd0	Emu_lkctrl	Emu_lkctrl
Write:							1	0
Reset:	1	1	0	1	0	0	0	0

Apci_ctrl :

'1': modulator select '10ua'

'0': banding the I to adc_clk

(adc_clk=900k, modulator select '10ua'; adc_clk=450k/270k,modulator select '5ua')

Adc_i1on: 打开第一路 adc

Adc_i2on: 打开第二路 adc

Adc_uon: 打开第三路 adc

默认情况下, 打开EMU即打开第一路第三路adc

- 1) DSP时钟来自CMU, 固定频率5.5M
- 2) ADC时钟由fadc提供, fadc是fdsp/6。

Emu_lkctrl1	Emu_lkctrl0	femu	fadc
0	0	femu	femu/6(900k)
0	1	Femu/2	femu/12(450k)
1	0	Femu/4	femu/24(225K)
1	1	Femu/4	fpri/24

ZXD1	ZXD0	description
0	0	正向过 0 中断
0	1	负向过 0 中断
1	x	双向过 0 中断

表 16-58 P1OFFSET Active Power offset 1 (5CH)

Power offset 1 (P1OFFSET)			Address: 5CH					
	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

有功功率通道1校正寄存器采用二进制补码形式

表 16-59 P2OFFSET Active Power offset 2 (5DH)

Power offset 2 (P2OFFSET)			Address: 5DH					
---------------------------	--	--	---------------------	--	--	--	--	--

	Bit7	6	5	4	3	2	1	Bit0
Read:	P2OFFSET							
Write:	7	6	5	4	3	2	1	0
Reset:	0	0	0	0	0	0	0	0

有功功率通道2校正寄存器采用二进制补码形式

表 16-60 Q1OFFSET Reactive Power offset 1 (5EH)

Reactive Power offset 1 (Q1OFFSET)		Address: 5EH						
	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

无功功率通道1校正寄存器采用二进制补码形式

表 16-61 Q2OFFSET Reactive Power offset 2 (5FH)

Reactive Power offset 2 (Q2OFFSET)		Address: 5FH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	Q2OFFSET	Q2OFFSET	Q2OFFSET	Q2OFFSET	Q2OFFSET	Q2OFFSET	Q2OFFSET	Q2OFFSET
Write:	7	6	5	4	3	2	1	0
Reset:	0	0	0	0	0	0	0	0

有功功率通道2校正寄存器采用二进制补码形式

表 16-62 I1RMSOFFSET I1RM offset (60H)

(I1RMSOFFSET)		Address: 60H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	I1RMSOFF	I1RMSOFF	I1RMSOFF	I1RMSOFF	I1RMSOFF	I1RMSOFF	I1RMSOFF	I1RMSOFF
Write:	SET7	SET6	SET5	SET4	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I1RMS的值，然后通过如下公式计算出I1RMS offset寄存器写入值。公式：

$$(I1RMS/2^{23})^2 * 2^{31} = I1RM\ offset$$

表 16-63 I2RMSOFFSET I2RM offset (61H)

(I2RMSOFFSET)		Address: 61H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2RMSOFF	I2RMSOFF	I2RMSOFF	I2RMSOFF	I2RMSOFF	I2RMSOFF	I2RMSOFF	I2RMSOFF
Write:	SET7	SET6	SET5	SET4	SET3	SET2	SET1	SET0
Reset:	0	0	0	0	0	0	0	0

电流通道2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I2RMS的值，然后通过如下公式计算出I2RMS offset寄存器写入值。公式：

$$(I_{RMS}/2^{23})^2 * 2^{31} = I_{2RMS} \text{ offset}$$

表 16-64 URMSOFFSET URM offset (62H)

(URMSOFFSET)			Address: 62H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	URMSOFF	URMSOFF	URMSOFF	URMSOFF	URMSOFF	URMSOFF	URMSOFF	URMSOFF
Write:	SET7	SET6	SET5	SET4	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

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

$$(URMS/2^{23})^2 * 2^{31} = URMS \text{ offset}$$

表 16-65 osi_ctrlr ROSI_CTRL(63H)

(rosi_ctrl)			Address: 63H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	Rosi_i2_en	Rosi_i1_en
Write:								
Reset:							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

表 16-66 UCONST (64H)

UCONST			Address: 64H				
	Bit19	18	17	16.....3	2	1	Bit0
Read:	UCONST15	UCONST14	UCONST13	UCONST3	UCONST2	UCONST1	UCONST0
Write:				UCONST12			
Reset:	0	0	0	0	0	0	0

失压情况下参与计量的电压寄存器16位，采用二进制无符号数形式。

该寄存器主要目的是替代Urms寄存器的值参与计量，用户也可以根据需要写入自己需要的值。该寄存器与Urms的对应关系如下：

URms是24位寄存器，其中最高位恒为0，即23位有效；而UCONST是16位有效位寄存器，所以在正常信号输入下读出来的寄存器URms值要右移7位，即除以 2^7 ，就可以得到Uconst的值。

15.4 校表过程

以标准表校验为例。

1. 高频脉冲常数设置

femu=5.505024MHz 时, HFConst 的计算公式如下

$$\text{HFConst} = 1.244 * \text{Vu} * \text{Vi} * 10^{11} / (\text{EC} * \text{Un} * \text{Ib})$$

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。

2. 第一通道有功、无功和视在增益校正

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

已知:

标准表上读出误差为 err

计算公式:

$$\text{Pgain} = \frac{-err}{1+err}$$

如果 Pgain ≥ 0, 则 GP1 = INT[Pgain * 2¹⁵]

否则 Pgain < 0, 则 GP1 = INT[2¹⁶ + Pgain * 2¹⁵]

同时将计算出的 GP1 的值写入到 GQ1 和 GS1。

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

4. 电流通道 2 增益校正 (做防窃电时必需)

做防窃电时, 需要对两个通道的电流有效值进行比较, 因而在同样电流输入下, 电流通道 1 与电流通道 2 的寄存器值应该相等。

通过电流通道 2 增益校正寄存器 I2GAIN(4AH), 使同样输入电流情况下, 二者寄存器的值一致。

假设同样输入额定电流, 电流通道 1 有效值寄存器读数为 I1rms, 电流通道 2 有效值寄存器读数为 I2rms, 则

$$\text{Gain} = I_{1\text{rms}}/I_{2\text{rms}} - 1$$

如果 $\text{Gain} \geq 0$, $I_2\text{Gain} = \text{Gain} * 2^{15}$

如果 $\text{Gain} < 0$, $I_2\text{Gain} = \text{Gain} * 2^{15} + 2^{16}$

5. 第二通道增益校正、相位校正
第二通道增益校正、相位校正与通道 1 类似。

简单举例:

假设计一块 220v (U_n)、5A (I_b) 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 $f_{\text{emu}} = 5.505024\text{MHz}/2$,

1. 计算 HFConst

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

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

$$\begin{aligned} \text{HFConst} &= [1.244 * V_u * V_i * 10^{11} / (\text{EC} * U_n * I_b)] / 2 \\ &= [1.244 * 0.220 * 0.028 * 10^{11} / (3200 * 220 * 5)] / 2 \\ &= 108.85 \end{aligned}$$

取整后 HFConst 为 0x6C(108)。将该值写入 HFConst(4FH) 寄存器即可。

2. 阻性增益校正

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

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

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

将 0x50FB 同时写入 GP1(41H)/GQ1(42H)/GS1(43H) 寄存器, 即完成阻性增益校正。

3. 相位校正

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

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

$$G_{\text{phs1}} = 0.0023 * 2^{15} = 75.3$$

取整后为 0x4BH, 写入角度校正寄存器 Gphs1(44H) 即可。

16 KBI

16.1 概述

键盘单元支持 2 个独立的可屏蔽的外部中断。

特征:

- 键盘输入引脚 KEY 内部 pull-high, 30K 电阻 (Key0, Key1 为 floating)
- Key0, Key1 上升沿有效模式
- 支持按键退出 SLEEP

通过 KEYIE0–KEYIE1 可以单独控制 P1 口是否作为键盘输入引脚。选择 P1[3..2] 口作为键盘输入引脚, 将使能内部上拉 30K 电阻。

16.2 按键中断

16.2.1 按键中断产生

触发类型是下降沿 (或 Key0, Key1 上升沿), 那么当有一个按键中断已经发生时, 其他的按键中断发生的时候, 相应的 KEYIF 置位。(在相应的 KEY 功能 PIN 上出现一个下降沿信号, 并保持 4 个 fosc 周期的低电平时间, 可以产生一个有效的中断标志置位)。

注意: 如果出现高电平, 并出现新的下降沿, 就以新的下降沿开始计时, 并保持 4 个 fosc 周期的低电平时间。

16.2.2 按键中断清除

对 KEYIFX 中对应的 KEYX 的中断标志位, 写 0 清中断标志。

16.3 低功耗模式

16.3.1 Hold 模式

Hold 模式不影响按键单元, 使能按键中断, 按键中断可以使 CPU51 退出 Hold 模式。

16.3.2 Sleep 模式

支持按键唤醒退出 SLEEP 模式。在相应的 KEY 功能 PIN 上出现一个下降沿信号 (或上升沿 Key0, Key1), 并保持 8 个 Fosc 周期的低电平 (Key0, Key1 触发为高电平) 时间, 可以产生一个有效的中断标志置位)

16.4 特殊功能寄存器

表 16-1 KEY SFR 寄存器列表

地址 (EPADR)	名称	字节长度	功能描述
0xAF	KEYIE	1	按键中断使能寄存器

0xB6	KEYIF	1	按键中断标志寄存器
0xB8	IEN1	1	中断使能寄存器, 详细内容参见 10.4.1 节

表 16-2 KEYIE 按键中断使能寄存器 (AFH)

KEY Interrupt Enable Register (KEYIE)	Address: AFH							
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	KEYIE3	KEYIE2	KEYIE1	KEYIE0
Write:								
Reset:	0	0	0	0	0	0	0	0

KEYIE_x=0: 当 KEY_x 选择作为按键输入, 屏蔽使能按键 KEY_x 中断.

KEYIE_x=1: 当 KEY_x 选择作为按键输入, 使能按键 KEY_x 中断.

位	功能描述
KEYIE3	KEY3 功能 PIN 中断功能使能有效 1:表示 KEY3 功能 PIN 中断有效 0: 表示 KEY3 功能 PIN 中断无效
KEYIE2	KEY2 功能 PIN 中断功能使能有效 1:表示 KEY2 功能 PIN 中断有效 0:表示 KEY2 功能 PIN 中断无效
KEYIE1	KEY1 功能 PIN 中断功能使能有效 1:表示 KEY1 功能 PIN 中断有效 0:表示 KEY1 功能 PIN 中断无效
KEYIE0	KEY0 功能 PIN 中断功能使能有效 1:表示 KEY0 功能 PIN 中断有效 0:表示 KEY0 功能 PIN 中断无效

表 16-3 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	KEYIF3	KEYIF2	KEYIF1	KEYIF0
Write:								
Reset:	0	0	0	0	0	0	0	0

注意: 该寄存器是不能被 Wake_UP Reset。

位	功能描述
KEYIF3	KEY3 功能 PIN 产生中断标志 1:表示产生中断标志 0:表示没有产生中断标志 写 0 清中断
KEYIF2	KEY2 功能 PIN 产生中断标志

	1:表示产生中断标志 0:表示没有产生中断标志 写 0 清中断
KEYIF1	KEY1 功能 PIN 产生中断标志 1:表示产生中断标志 0:表示没有产生中断标志 写 0 清中断
KEYIF0	KEY0 功能 PIN 产生中断标志 1:表示产生中断标志 0:表示没有产生中断标志 写 0 清中断

KEYIF_x=0: 当 KEY_x 选择作为按键输入, KEY_x 端口满足电平触发条件的时候, 对应的 Flag 置位。
 注意:

1. 当 KEY/GPIO 功能复用 PIN 配置为 KEY 功能的时候, 当 KEY[1..0] 的 KEY 功能 PIN 上出现下降沿并保持 4 个 fosc 的低电平时间, 相应的 KEYIF 出现置位, KEYIF[X]=1。如果对应的 KEYIF[X]=1 中断使能有效, EX2=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。
2. 在 SLEEP MODE 下, 当 KEY/GPIO 功能复用 PIN 配置为 KEY 功能的时候, 当 KEY[3..2] 的 KEY 功能 PIN 上出现下降沿 (或者 KEY[1..0] 的功能 PIN 上出现上升沿), 保持 8 个 fosc 的低 (高) 电平时间, 相应的 KEYIF 出现置位, 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。

17 LCD

17.1 概述

LCD 驱动单元最大可以支持 14 Segment * 4 Common 的输出模式。

特征:

- 软件可编程驱动方式。(具体要根据最终的 GPIO 引脚来定)
- LCD 驱动电压可选
- 1/3 Bias, 1/4 Bias
- Static、1/3、1/4、1/8 Duty 可选

17.2 LCD 与 GPIO 引脚复用

LCD 引脚定义	功能引脚定义	芯片引脚定义
COM0-COM3	--	COM0 - COM3
SEG0-SEG3	PA0-PA3	SEG0/PA0 - SEG3/PA3
SEG4-SEG7	PA4-PA7	SEG4/COM4/PA4 - SEG7/COM7/PA7
SEG8-SEG15	PB0-PB7	SEG8/PB0 - SEG15/PB7
SEG16-SEG23	P0_0-P0_7	SEG16/P0_0 - SEG23/P0_7
SEG24-SEG31	PD0-PD7	SEG24/PD0 - SEG31/PD7
SEG32-SEG39	PC. 0-PC. 7	SEG32/PC. 0 - SEG39/PC. 7

17.3 原理框图

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。

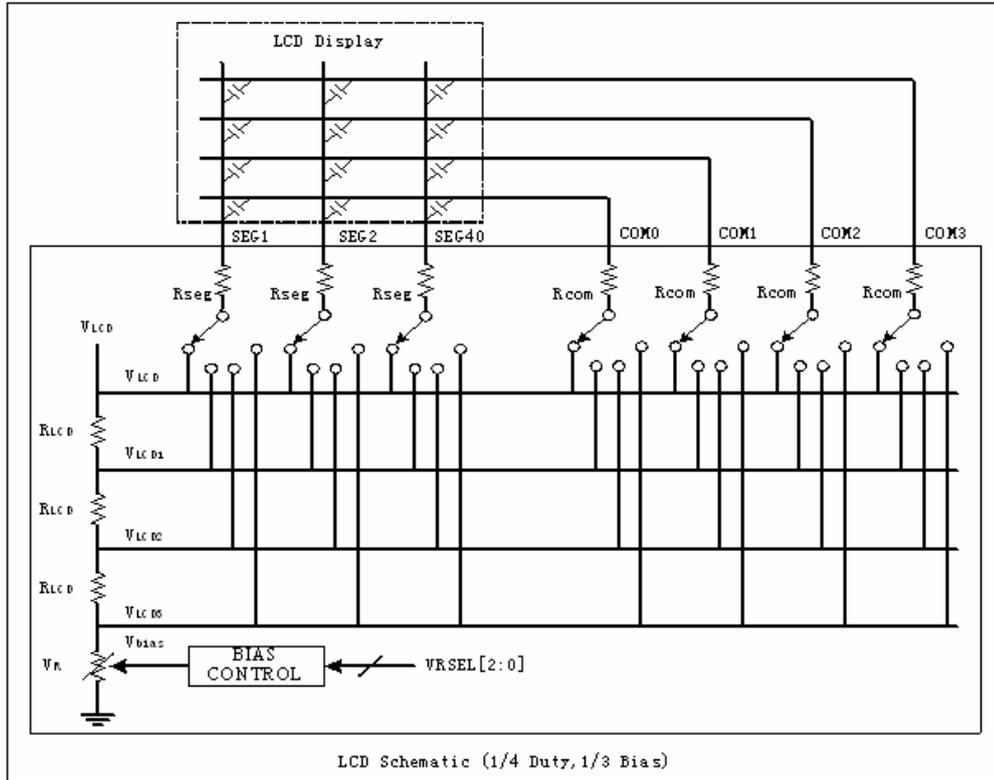


图 20 - 1

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。

17.4 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
- ◆ DUTY[1:0]=11: 1/8 duty——COM0 到 COM7 都被使用

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。

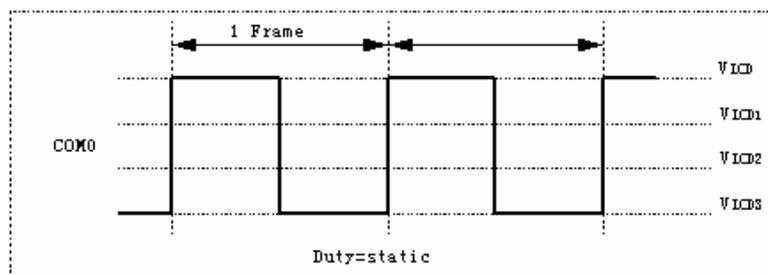


图 18-2

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。

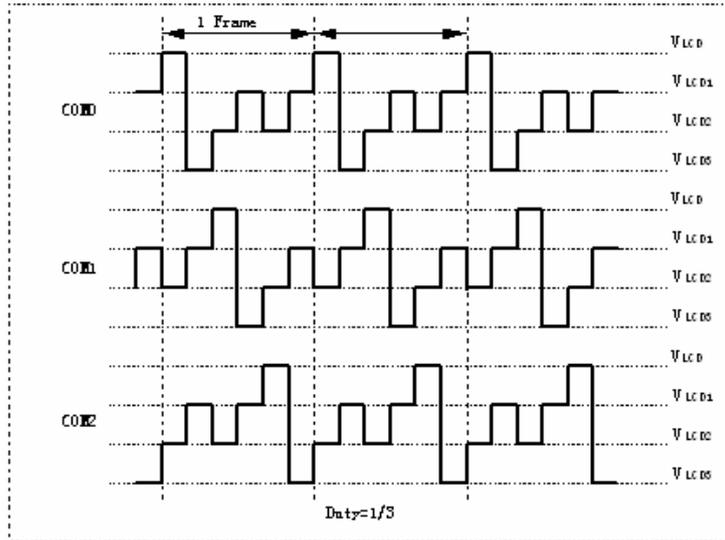


图 18-3

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。

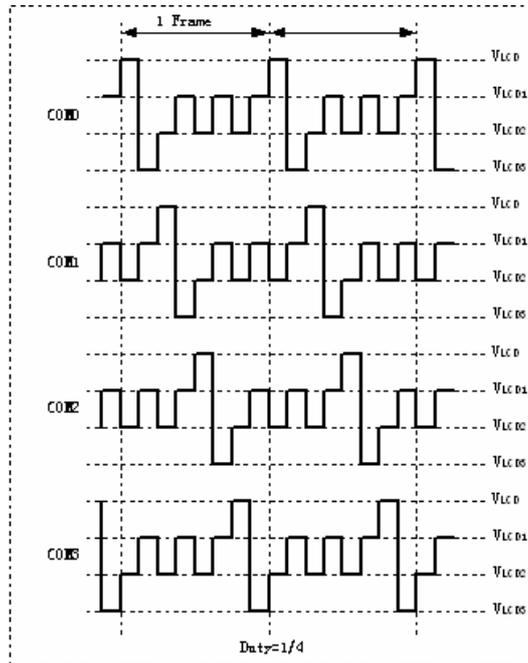


图 18-4

SEGMENT 输出波形:

① Static Duty

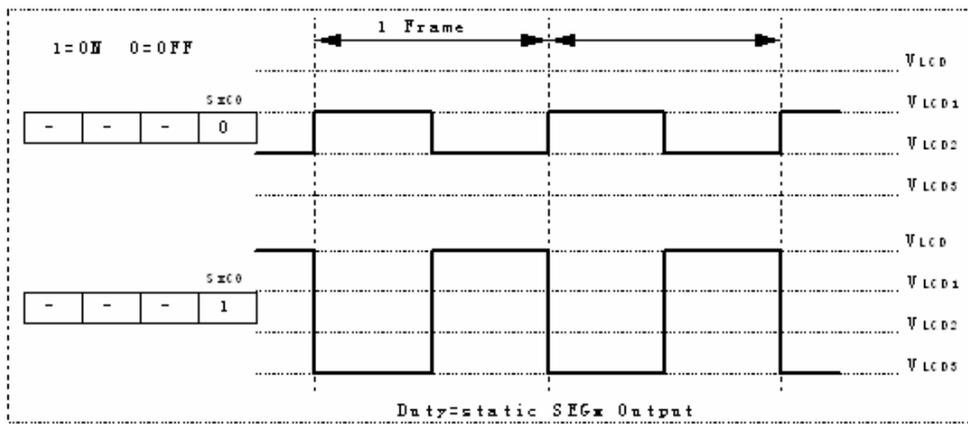


图 18-5

② 1/3 Duty

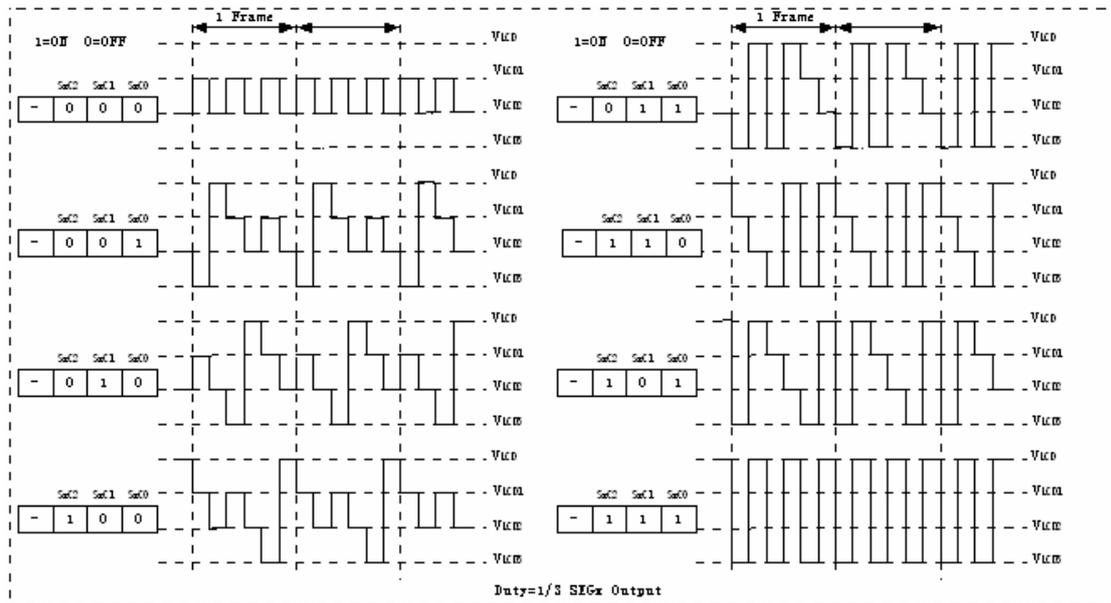


图 18-6

③ 1/4 Duty

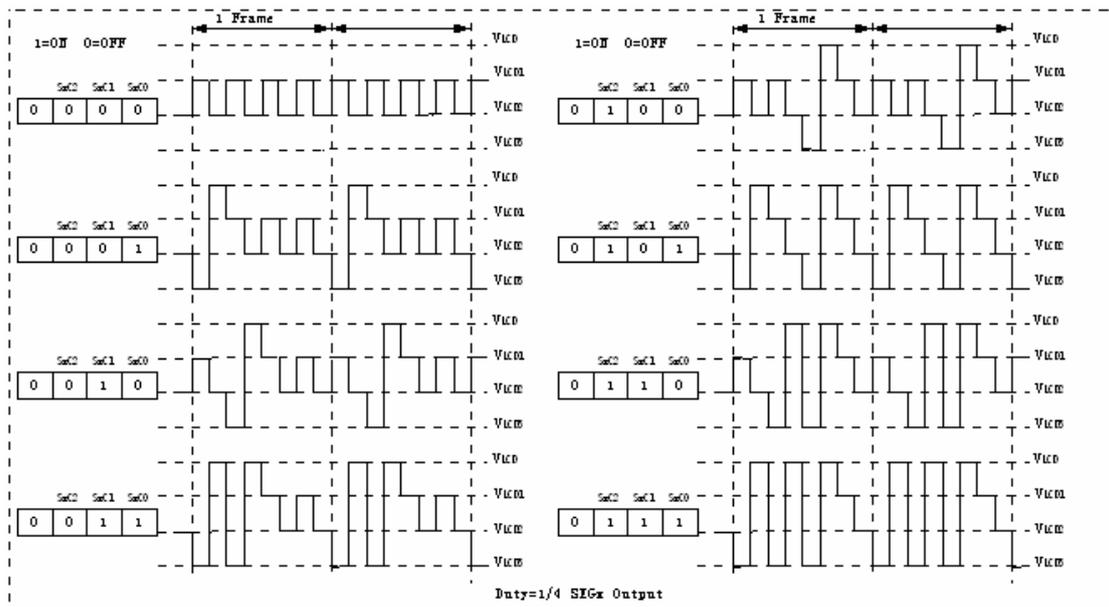


图 18-7

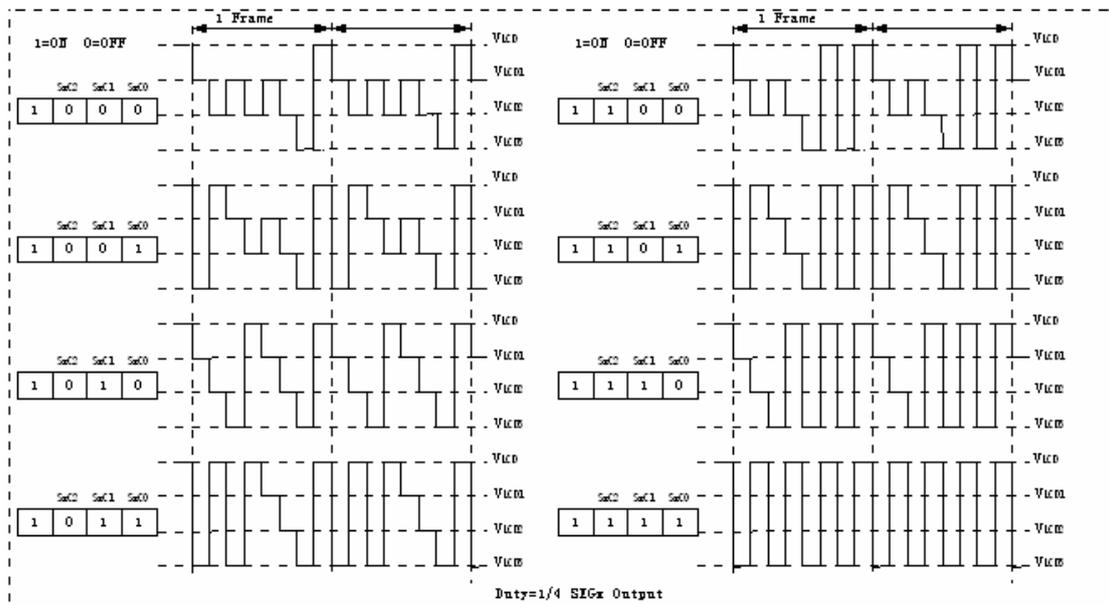


图 18-8

17.5 寄存器

1、LCD clock 寄存器

表 18-1 LCDCLK LCD Clock Register (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

DUTY[1:0]: LCD 的 duty 控制选择

LCLK[1:0]: LCD 扫描频率选择

BIASCTL: 1/3 bias 和 1/4 bias 选择信号。

BIASCTL=1, 为选择 1/3 bias 偏压驱动。

BIASCTL=0, 为选择 1/4 bias 偏压驱动。

为了更好地提高显示效果, 在设置 DUTY[1:0]=11: 1/8 duty——COM0 到 COM7 都被使用, 最好选择 1/4 bias 偏压驱动。

LCD 驱动电压:

LCD 电压来自 VLCD 引脚, 但是不能比 VCC 大, 即 $VLCD \leq VCC$ 。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。

LCD Duty:

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
- DUTY[1:0]=11: 1/8 duty——COM0 到 COM7 都被使用

LCD 波形扫描频率:

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。

LCD 帧扫描频率 $f_{frm} = f_{lcd} * duty$ 。

如: 1/3 duty 时, 帧扫描频率 $f_{frm} = f_{lcd} / 3$ 。

LCLK			Divide Ratio	f _{lcd} (Hz)	frame rate(Hz)			
LCLK2	LCLK1	LCLK0			Static	1/3	1/4	1/8
1	0	0	64	512	512	170.7	128	64
0	0	0	128	256	256	85.3	64	32
0	0	1	256	128	128	42.7	32	16
0	1	0	512	64	64	21.3	16	8
0	1	1	1024	32	32	10.7	8	4

2、LCDCR 寄存器

表 18-2 LCDCR LCD Control Register (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

FC: 快速充电模式选择。

LC: LCD 大电流驱动模式选择。

VRSEL[2:0]: LCD 显示对比度调节。

LCD 工作电流:

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

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

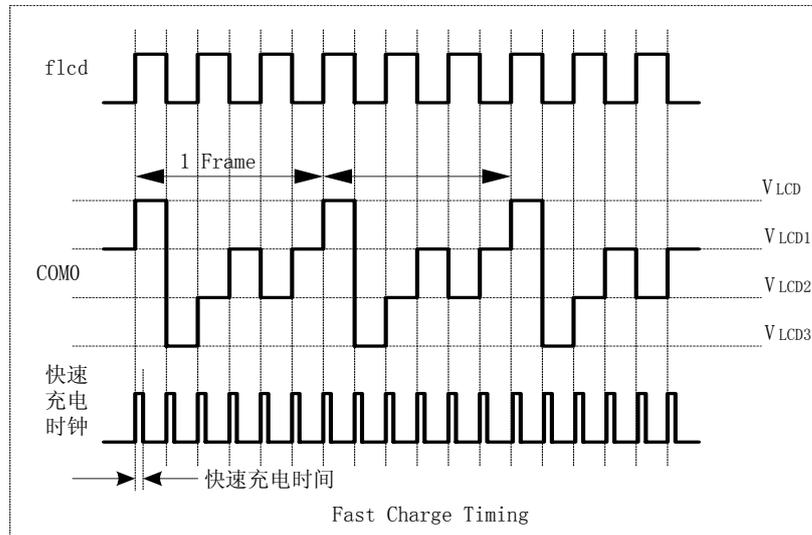
FCSET[1:0]: 快速充电脉宽选择。

FC: 快速充电模式选择。

LC: LCD 大电流驱动模式选择。

FC	LC	
X	0	RLCD=37KΩ
0	1	RLCD=146KΩ
1	1	快速充电模式

FCSET1	FCSET0	快速充电时间
0	0	1/8 个 flcd 周期
0	1	1/16 个 flcd 周期
1	0	1/32 个 flcd 周期
1	1	1/64 个 flcd 周期



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。

VRSEL2	VRSEL1	VRSEL0	Bias Volatge (% of VLCD)	
			1/3 bias	1/4 bias

0	0	0	0.1	0
0	0	1	2.2	1.7
0	1	0	4.3	3.3
0	1	1	6.3	4.8
1	0	0	8.3	6.4
1	0	1	10.1	7.8
1	1	0	11.9	9.2
1	1	1	13.6	10.6

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 3000H 对应 Seg0 (COM7—COM0)

地址 3001H 对应 Seg1 (COM7—COM0)

地址 3002H 对应 Seg2 (COM7—COM0)

地址 3027H 对应 Seg39 (COM7—COM0)

如把 SEG0 , COM7 点亮的操作:

MOV 3000H , 0X80;

如把 SEG0 , COM7 , COM6 都点亮的操作:

MOV 3000H , 0XC0;

18 TBS

18.1 概述

ATT7039 可以对温度和电池进行定量的测量，并将测量的结果保存在相应的寄存器中。用户可以根据测量的结果将当前的温度和电池电量在 LCD 上显示或者对 RTC 等模块进行补偿。

18.2 功能描述

TBS为芯片的温度测量部分，主要包括两部分功能：

- 1、测量 IC 基底的温度。
- 2、测试 VBAT 上的电压。
- 3、对于测量温度变化以及 VBAT PIN 上的电压超过了用户设置的的差值（阈值范围），产生相应的温度测量中断和电池电压测量中断。

18.3 功能详细说明

TBS 模块主要利用一个 8 位的 ADC 对温度和电池电压做分时的测量，自动比较前后两次测量得到的值之间的差异，如果大于用户设定的差值，新测量得到的值将被更新到相应的数据寄存器中，并产生相应的中断信号通知用户当前的数据已经被更新。

定量测量电池的电压，并将测量的结果保存在相应的寄存器中，并产生相应的中断信号

定量测量当前的温度，并将测量的结果保存在相应的寄存器中，并产生相应的中断信号

ATT7039 内置温度传感器（TBS），温度测量值存放于寄存器 TEM_DATA (0xA5H)中，电池电压测量值存放于寄存器 VBAT_DATA (0xA4H)中。

温度计算公式

温度转换公式为：

$$Tr = T_{off} - T_{data} * 0.689$$

其中 Tr 为实际的温度（供显示的温度）

T_{off} 为一个固定偏置，每个芯片的偏置需要分别校准。

T_{data} 由 TEM_DATA 转换而来：

$$TEM_DATA.7=0, T_{data}=(TEM_DATA[7:0])|(0x80);$$

$$TEM_DATA.7=1, T_{data}=(TEM_DATA[7:0])\&(0x7F)$$

温度偏置校正

在得到温度之前，必须得到温度偏置 T_{off}

$$\text{由 } Tr = T_{off} - T_{data} * 0.689$$

可知

$$T_{off} = Tr + T_{data} * 0.689$$

在恒温箱中得到环境温度 Tr，从寄存器中得到 T_{data}，即可算出 T_{off}。

温度计算举例

第一步计算 T_{off}，假设环境温度为 25℃，寄存器中读数为 FBH，如上提到的数据处理方法，得到 T_{data}=0x7B=123 则

$$T_{off} = 25 + 123 * 0.689 = 110$$

将 T_{off} 保存到 flash 或其他非易失 memory 中，在下次校准前该芯片的所有温度偏置均为该值。

第二步，根据寄存器 TEM_DATA 的值，得到显示的温度 Tr(测量的温度)

Tr(显示)	Toff	TEM_DATA	Tdata
25	110	7BH	123
-25	110	44H	196

电池电压计算公式

Vdata 由 VBAT_DATA 转换而来，由 Vdata 得到实际测量的电压 Vr:

VBAT_DATA.7=0, Vdata=(VBAT_DATA [7:0])|(0x80);

VBAT_DATA.7=1, Vdata=(VBAT_DATA [7:0])&(0x7F)

电池电压计算举例

由以上公式得到的 Vdata，和比例系数，可以得到电池测量电压值。

$V_r = V_{data} * 14.89(\text{mV})$

Vr(显示) (V)	VBAT_DATA	Vdata
3.648	75	245
3.559	6F	239

18.4 模块工作模式说明

时钟域：TBS的钟为32.768K的OSC clock

电压域：3.3V电压域

18.5 测量误差

18.5.1 IC 的温度测量

- 能够测量出 IC 的温度。
- 在室温下(+25°C)测量误差在±1°C。
- 在-40°C---80°C的范围内，测量误差为±4°C。
- 能够产生电池电压测量中断。

温度计量 0.689°C/LSB

测量温度范围 -40~80°C

18.5.2 IC 的电池电压测量

- 能够测试电池电压，在室温下(+25°C)测量误差在±15mV。
- 能够产生电池电压测量中断。

Battery 计量 14.89mV/LSB

测量 Battery 电压范围

0~3.8V

18.6 特殊功能寄存器

表 18-1TBS 相关寄存器列表

地址	名称	复位值	功能描述
A3H	DIFF_CFG		差值配置寄存器
A4H	VBAT_DATA		电池电压测量值
A5H	TEM_DATA		温度测量值

表 18-2 DIFF_CFG (different configure register) (A3H)

different configure register		Address: A3H						
	Bit7	6	5	4	3	2	1	Bit0
Read	x	Diff_b2	Diff_b1	Diff_b0	x	Diff_t2	Diff_t1	Diff_t0
Write								
Reset:	0	0	0	0	0	0	0	0

注意:

电池电压测量的的 1BIT 的分辨率是 14.89mV.

温度测量的的 1BIT 的分辨率是 0.689°C.

位	功能描述
Diff_b[2:0]	电池电压差值配置位：当前后两次测量大于设定的差值就将数据更新到 VBAT_DATA 寄存器。 110: 725mv 101: 604mv 100: 483mv 011: 362mv 010: 241mv 001: 121mv 000: 每次测量都更新 VBAT_DATA 的数据 注意: Default 下为000, 每测量一次就对VBAT_DATA进行一次数据更新
Diff_t[2:0]	温度差值配置位：当前后两次测量大于设定的差值就会将数据更新到 TMP_DATA 寄存器 111: 5.04°C 110: 4.32°C 101: 3.6°C 100: 2.88°C 011: 2.16°C 010: 1.44°C

	001: 0.72°C 000: 每次测量都更新 VBAT_DATA 的数据 注意: Default 下为 000, 每测量一次就对 TMP_DATA 进行一次数据更新
--	---

表 18-3 VBAT_DATA (Vbat data register) (A4H)

Vbat data register(VBAT_DATA)		Address: A4H						
	Bit7	6	5	4	3	2	1	Bit0
Read	VBAT_DA	VBAT_DA	VBAT_DAT	VBAT_DA	VBAT_DA	VBAT_DA	VBAT_DA	VBAT_D
Write	TA7	TA6	A5	TA4	TA3	TA2	TA1	ATA0
Reset:	x	x	x	x	x	x	x	x

位	功能描述
VBAT_DATA[7: 0]	VBAT_ADC 测量到的电压值 (read only)

表 18-4 TMP_DATA (temperature data register) (A5H)

Temperature data register(TMP_DATA)		Address: A5H						
	Bit7	6	5	4	3	2	1	Bit0
Read	TMP_DAT	TMP_DAT	TMP_DAT	TMP_DAT	TMP_DAT	TMP_DAT	TMP_DAT	TMP_D
Write	A7	A6	A5	A4	A3	A2	A1	ATA0
Reset:	x	x	x	x	x	x	x	x

位	功能描述
TMP_DATA[7: 0]	TMP_ADC 测量到的温度值 (read only)

19 RTC

19.1 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。

19.2 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、年和星期等
- 具有自动闰年闰月调整功能
- 1 个闹钟中断功能
- 2 个定时器周期性中断功能
- 5 个时间中断功能（秒、分、时、日、月）
- 可输出未校正的频率 1024/32768Hz
- 可输出校正后的频率 1/2/4/8/32/128Hz

19.3 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，校正结果能够保证每 20 秒内产生的时钟周期数与标准 32768Hz 每分钟内的时钟周期数一致。

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，计算需要补偿的数写入 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。

RTC 补偿为 20 秒补偿一次。对于 **RTCCAL (F9H)**，代表每次补偿加减 N 个低频时钟的时间（在相应要补偿的秒周期上）。若实际晶振小于标准时钟 32768Hz，则应该减少 N 个低频时钟的时间。反之，若实际晶振大于标准时钟 32768Hz，则应该增加 N 个低频时钟的时间。

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。

最小解析度为 $1/(32768*20)=1.525\text{ppm}$ 。

例如：

若实际测得晶振频率为 32767Hz，由于标准晶振为 32768Hz，芯片为 20s 补偿一次，根据公式，被补偿数 N 为：

$$(1/32767)*32768*19 + (1/32767)*(32768+N) = 20$$

$$N = (32767 - 32768) * 20$$

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 11101100

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 20-1 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	$\sim(1000_0000)+1=1000_0000=128$
0x81	1000_0001	减少	$\sim(1000_0001)+1=0111_1111=0x7f=127$
0x82	1000_0010	减少	$\sim(1000_0010)+1=0111_1110=0x7e=126$
0xaf	1010_1111	减少	$\sim(1010_1111)+1=0101_0001=0x51=81$
0xb0	1011_0000	减少	$\sim(1011_0000)+1=0101_0001=0x50=80$
0xfe	1111_1110	减少	$\sim(1111_1110)+1=0000_0010=0x02=2$
0xff	1111_1111	减少	$\sim(1111_1111)+1=0000_0001=0x01=1$

19.4 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。

19.5 寄存器写保护功能与复位源

表 20-2 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 Reset, BOR, Debug RST 复位	否	RTC 中断标志
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 CMU 单元中的写保护寄存器的设置）

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    0F9H, #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。

19.6 中断功能

RTC 一共提供 7 种中断源，共用 MCU 的 IRQ_RTC 中断向量。RTC 的 7 种中断源由 RTCIE（AAH）控制其使能，中断标志在 RTCIF（B4H）中。中断标志采用写 0 清 0 的方式。

具体的中断产生条件和中断清除步骤如下：

ALMF：闹钟中断标志

当小时和分钟与设定的闹钟匹配时，产生闹钟中断，ALMF 被置为 1。

对该位写 0 清标志。

RTC1F: RTC 定时器 1 中断标志

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

对该位写 0 清标志。

RTC2F: RTC 定时器 2 中断标志

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

对该位写 0 清标志。

MTHF: 月中断

月计数器 $MTHR$ 加 1 时，产生一个月中断， $MTHF$ 被置为 1。

对该位写 0 清标志。

DAYF: 日中断

日期计数器 $DAYR$ 加 1 时，产生一个日中断， $DAYF$ 被置为 1。

对该位写 0 清标志。

HRF: 小时中断

小时计数器 HRR 加 1 时，产生一个小时中断， HRF 被置为 1。

对该位写 0 清标志。

MINF: 分钟中断

分钟计数器 $MINR$ 加 1 时，产生一个分钟中断， $MINF$ 被置为 1。

对该位写 0 清标志。

SECF: 秒中断

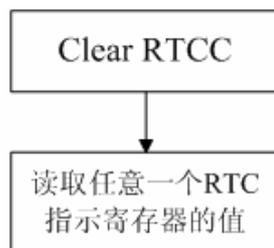
秒计数器 $SECR$ 加 1 时，产生一个秒中断， $SECF$ 被置为 1。

对该位写 0 清标志。

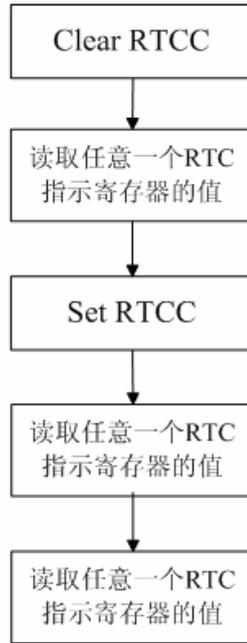
19.7 RTC 指示寄存器读取流程

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：



19.8 特殊功能寄存器

表 20-3 寄存器列表

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

表 20-4 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
--------	---	---	---	---	---	---	---	---

RTCC:

RTCC = 0: 开始读取 RTC 指示寄存器读数

RTCC = 1: 连续读取 RTC 指示寄存器读数

RTC2EN: RTC 定时器 2 使能

RTC2EN=0: RTC 定时器 2 被关闭

RTC2EN=1: RTC 定时器 2 被使能, 溢出产生 RTC2IF 标志。

RTC1EN: RTC 定时器 1 使能

RTC1EN=0: RTC 定时器 1 被关闭

RTC1EN=1: RTC 定时器 1 被使能, 溢出产生 RTC1IF 标志。

TOUTEN[1:0]:

TOUTEN[1:0]=00 时 TOUT 输出固定低电平 0; =01 时 TOUT 输出固定高电平 1, 而=10 或者 11 时 TOUT 按照 TOUT[2:0]配置输出。

TOUT[2:0]:

选择端口输出频率。

TOUT2	TOUT1	TOUT0	TOUT	TOUT2	TOUT1	TOUT0	TOUT
0	0	0	1Hz	1	0	0	32Hz
0	0	1	2Hz	1	0	1	128Hz
0	1	0	4Hz	1	1	0	1024Hz
0	1	1	8Hz	1	1	1	32.768KHz

表 20-5 RTCCAL RTC 校正寄存器 (写保护) (F9H)

RTC Calibration Register (RTCCAL)	Register	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

表 20-6 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	SECF
Write:								
Reset:	0	0	0	0	0	0	0	0

1: 使能中断; 0: 关闭中断。

表 20-7 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:	ALMF	RTC2IF	RTC1IF	MTHF	DAYF	HRF	MINF	SECF

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

表 20-8 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

允许写入 0-59 以外的数，但是闹钟中断将永远不会产生。

表 20-9 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

允许写入 0-23 以外的数，但是闹钟中断将永远不会产生。

表 20-10 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

最小分格为 1s，即最小可以每 1 秒产生一次中断，最大可以每 256 秒产生一次中断，当计数溢出时，置位 RTC2IF 标志。

RTC1N[7...0]用来表示一个 8BIT 的二进制的无符号整数，如果设置 RTC1N[7...0]=00H，表示 RTC 内部的秒表功能中断每经过 $(00H+1)*1S=1*1S=1S$ 的计时周期后，置位 RTC1IF 标志。

表 20-11 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

最小分格为 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。

RTC2N[7...0]用来表示一个 8BIT 的二进制的无符号整数，如果设置 RTC2N[7...0]=13H，表示 RTC 内部的秒表功能中断每经过 $(13H+1)*0.0625S=20*0.0625S=1.25S$ 的计时周期后，置位 RTC2IF 标志。

表 20-12 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

范围：0-59。写入 0-59 以外的任何数值，对该寄存器没有影响。

表 20-13 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

范围：0-59。写入 0-59 以外的任何数值，对该寄存器没有影响。

表 20-13 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

范围：0-23。写入 0-23 以外的任何数值，对该寄存器没有影响。

表 20-14 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

范围：1-28/29/30/31。写入与年、月不匹配的任何数值，对该寄存器没有影响。

表 20-15 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

范围：1-12。写入 1-12 以外的任何数值，对该寄存器没有影响。

表 20-16 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

有效范围：0-99，最大可写入 127。写入 0-127 以外的任何数值，对该寄存器没有影响。

表 20-17 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

范围：1-7。写入 1-7 以外的任何数值，对该寄存器没有影响。

20 I2C

20.1 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。

20.2 结构框图

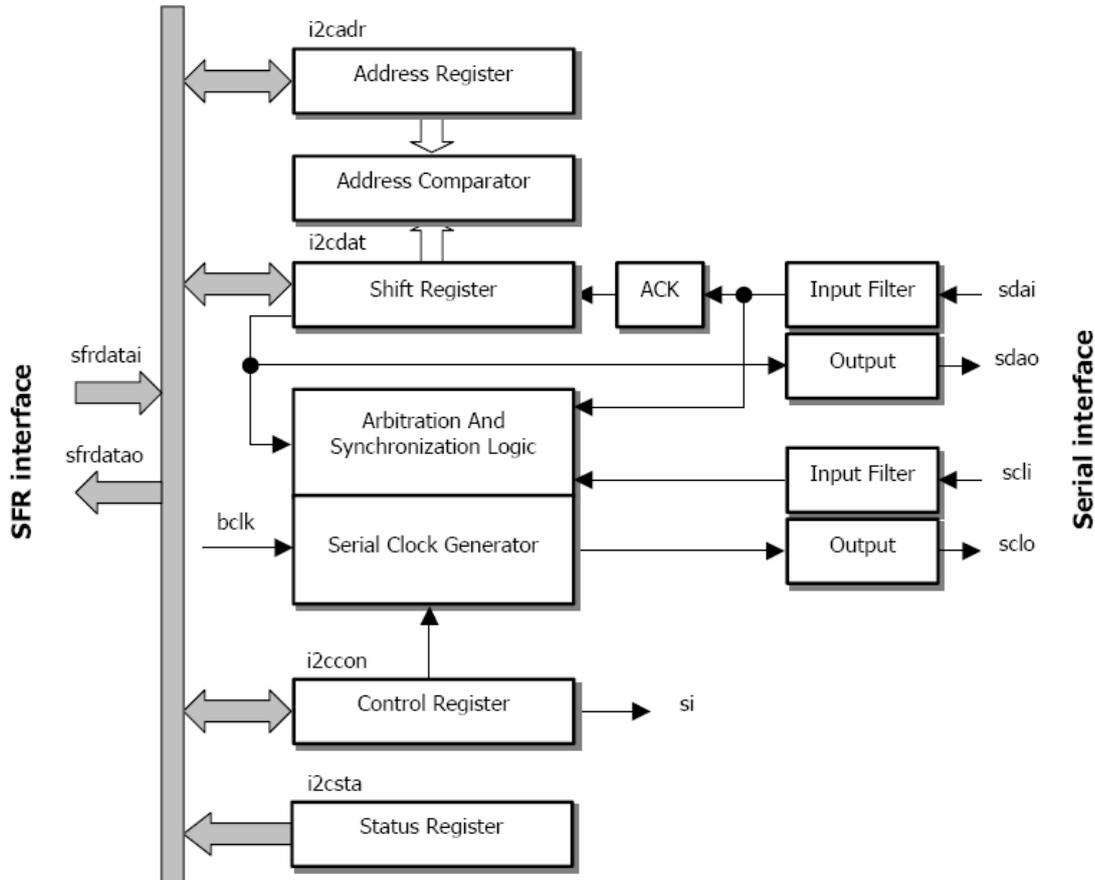


Figure 100. I2C block diagram

20.3 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，它包含冲突侦测和仲裁机制，以防止两个以上主机同时开始数据传输时的数据丢失。

20.3.1 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

20.3.2 串行时钟生成

当 I2C 处于主机模式时，可编程的时钟发生器提供“scl”时钟；当 I2C 处于从机模式时，时钟发生器被关闭，接收来自主机的时钟。时钟发生器的输出频率可以由寄存器“i2ccon”中的位“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。

20.3.3 地址比较器

20.3.4 中断生成

IIC 产生中断时，寄存器“i2ccon”中的标志位“si”会被置位。

20.4 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 22-1 寄存器列表

地址	名称	复位值	字节长度	功能描述
DAH	I2CDAT	00H	1	
DBH	I2CADR	00H	1	
DCH	I2CCON	00H	1	
DDH	I2CSTA	F8H	1	

表 22-2 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，是将要被传送到总线上的数据，或者是刚从总线上接收到的数据。寄存器“i2cdat”没有设置影子寄存器，也没有双缓存，所以当I2C中断发生时，MCU需要及时从它读取数据，以免数据丢失。

表 22-3 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 i2cadr.3 i2cadr.2 i2cadr.1	I2C从机地址 (7 位)
i2cadr.0	General Call Address Acknowledge 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 22-4 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	Clock rate bit 2
ENS1	I2C 使能位 ens1=0: 关闭IIC模块; ens1=1: 打开IIC模块;
STA	开始标志 sta=1: 检查IIC总线的状态, 如果空闲则生成开始信号;
STO	停止标志 sto=1: 当处于主机模式, 则向总线传输停止信号
SI	中断标志 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”; 该位必须被软件写“1”复位。
AA	生成应答标志 aa=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; aa=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	Clock rate bit 1
CR0	Clock rate bit 0

Table 120. I2C Clock rate bit settings

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 22-4 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:	0	0	0	0	0	0	0	0

位	功能描述
i2csta.7 i2csta.6 i2csta.5 i2csta.4 i2csta.3	I2C Status Code
i2csta.2 i2csta.1 i2csta.0	Not implemented, read as 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。

Table 121. I2C Status In Master Transmitter Mode

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

图 22-3

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 22-4

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 22-5

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 22-6

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	
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 22-7

Table 123. I2C Status In Slave Receiver Mode

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	
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 22-7

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

图 22-8

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 22-9

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 22-10

Table 124. I2C Status In Slave Transmitter Mode

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 22-11

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

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 22-12

21 JTAG

21.1 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。

21.2 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，不能进入低频状态。

22 电气特性

22.1 极限参数

测量项目	符号	最小	典型	最大	单位
数字电源电压	V_{CC}	-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}	--	--	± 5	mA
I/O 最大输入灌电流	I_{sunk}	--	--	± 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	°C
存储温度范围	T_{stg}	-65	--	150	°C

22. 2DC 参数

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 Ω
LBOR			1.9		2.1	
POR 启动电压		V_{POR}	0	--	100	mV
VDCIN (Analog Input)	Maximum Signal Levels	0		3.3	不超过 VDD3 P3	V
	Input Impedance (DC)			1		M Ω
	Low VDCIN Detection Threshold		1.12	1.18	1.22	
	VBAT Voltage Test Range	0		3.8		V
TBS	VBAT Voltage Measure		14.89mv/ LSB			
	Temperature Measure		0.689 $^{\circ}$ C/ LSB			
	TBS Conversion Delay		1			ms

注意：以上表格中提到的Vcc等于VDD3P3上的电压。(VCC=AVCC=3.3V，温度范围：-40 $^{\circ}$ C~+85 $^{\circ}$ C)

22. 3功耗参数

工作模式(三路 ADC OPEN)	功耗 Adci_ctrl = 1	功耗 Adci_ctrl = 0
	fs=900KHz	fs=450KHz

fsys=5.5MHz VCC=3.3V	5.59mA (关闭 1 路 ADC 会降低 0.78mA 的功耗) Fsys=5.5M P(adc)=0.78mA P(emu)= 1.2mA P(vref)= 0.45mA P(cpu)=1.53mA P(spi)=23uA P(i2c)=26uA P(lcd)=10uA P(Bor)=15uA	3.84mA (关闭 1 路 ADC 会降低 0.39mA 的功耗) Fsys=5.5M P(adc)=0.39mA P(emu)= 0.6mA P(vref)= 0.45mA P(cpu)=1.53mA P(spi)=23uA P(i2c)=26uA P(lcd)=10uA P(Bor)=15uA
fsys=2.7MHz, fs=900KHz, VCC=3.3V(3 路 ADC 都打开)	Adci_ctrl = 1 5.11 mA	Adci_ctrl = 0 3.35 mA
fsys=1.3MHz, fs=900KHz, VCC=3.3V(3 路 ADC 都打开)	Adci_ctrl = 1 4.87 mA	Adci_ctrl = 0 3.11 mA
PSM fsys=32KHz, VCC=3.3V, WithBOR, LCD ,RTCon	48uA	
Hold Mode With,RTC,VSYS,LCD on	37uA	
Sleep Mode With RTC,LCD on(调整 Vsys 不同的检测周期)	15.0uA	
Sleep Mode With RTC on(调整 Vsys 不同的检测周期)	5.0uA TCR 设置的分时开启时间最小	

22.4 电能计量参数

(V_{CC}=AV_{CC}=3.3V, 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			0.1%			常温 1000:1 的动态范围
有功电能测量带宽			14		kHz	
无功电能测量误差			0.3%			常温 1000:1 的动态范围
电压有效值测量误差			0.5%			
电压有效值测量带宽			14		kHz	
电流有效值测量误差			0.5%			

电流有效值测量带宽			14		kHz	
两个通道间的相位误差 PF=0.8 容性 PF=0.5 感性			±0.04 ±0.04		度 度	相位超前 37 度 相位滞后 60 度

22. 5ADC 指标

($V_{CC}=AV_{CC}=3.3V$, 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}			±800	mV	模拟通道差分输入
直流输入阻抗	Z_{DC}	400			k Ω	
信噪比	SNR		86		dB	
-3dB 带宽	B_{-3dB}		14		kHz	

22. 6ADC 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}		1.25		V	
温度系数	T_c		±25		ppm/ $^{\circ}C$	