

ATT7027 用户手册

版本	更新日期	备注
V1.0	2006-10-20	创建初稿
V2.0	2007-02-14	正式发布
V2.2	2008-01-17	版本升级
V2.21	2008-6-20	勘误

1、49 页

PDBOR	BORPDM	Normal mode 和PSM 下BOR模块的开关	PDM下BOR模 块的开关
1	1	关	关
1	0	关	关
0	1	开	开
0	0	开	关

2、117 页

SM0_0	SM1_0	方式	功能说明	波特率
0	0	0	不支持	—
0	1	1	8 位 UART	可变
1	0	2	9 位 UART	fcpu/64 或 fcpu/32
1	1	3	9 位 UART	可变

3、118 页

SM0_1	SM1_1	方式	功能说明	波特率
0	0	0	不支持	—
0	1	1	8 位 UART	可变
1	0	2	9 位 UART	fcpu/64 或 fcpu/32
1	1	3	9 位 UART	可变

注：红色是修改后的状态

注意：钜泉光电科技（上海）有限公司保留对该手册在任何时候的改动权限和解释权。请随时注意我公司的更新动态，该手册如有改动更新，我公司不另行通知。

Tel: 021-51035886

Fax: 021-50277833

Email: sales@hitrendtech.com

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

目录

1	总体概况.....	8
1.1	简介	8
1.2	特点	8
1.2.1	基本特点	8
1.2.2	电能计量	8
1.2.3	处理器及外设	8
1.3	缩略语	9
1.4	系统框图	10
1.5	引脚分布	10
1.6	引脚描述	11
2	电气特性.....	14
2.1	极限参数	14
2.2	DC参数	14
2.3	电能计量参数	15
2.4	ADC	15
2.5	ADC基准电压	15
3	ATT7027 结构.....	16
3.1	概述	16
3.2	存储器	16
3.2.1	概述	16
3.2.2	内部数据存储器	16
3.2.3	扩展存储器映射	18
3.2.4	程序存储器	19
3.2.5	数据存储器	19
3.2.6	寄存器	20
3.3	指令系统	20
3.3.1	寻址方式	20
3.3.2	指令集	21
3.3.3	指令时序	25
3.4	特殊功能寄存器	25
3.4.1	概述	25
3.4.2	CPU52 寄存器	28
3.4.3	扩展寄存器	30
4	中断系统.....	32
4.1	概述	32
4.2	寄存器	32
4.3	中断处理	36
4.4	中断屏蔽	36
4.5	中断优先级	36

4.6	中断响应时间	37
5	时钟.....	38
5.1	概述	38
5.2	功能描述	38
5.2.1	时钟振荡电路	38
5.2.2	系统时钟生成	39
5.2.3	针对电表应用的时钟方案	41
5.3	寄存器	41
6	系统复位.....	43
6.1	概述	43
6.2	功能描述	43
6.2.1	上电复位POR	43
6.2.2	外部引脚复位/RST	43
6.2.3	Watchdog复位WDTR	43
6.3	寄存器	44
7	电源系统.....	47
7.1	概述	47
7.2	功能描述	47
7.2.1	电源系统	47
7.2.2	BOR	47
7.2.3	LVD	48
7.3	寄存器	48
8	低功耗模式.....	52
8.1	概述	52
8.2	各工作模式的状态	52
8.2.1	节电模式（Power saving mode）	52
8.2.2	正常模式（Normal mode）	52
8.2.3	掉电模式(Power Down mode).....	53
8.3	各工作模式转换	54
8.3.1	Normal mode -> PSM	54
8.3.2	PSM-> Normal mode	55
8.3.3	PSM 或Normal mode-> PDM	55
8.3.4	PDM -> PSM	56
8.4	寄存器	57
9	电能计量.....	59
9.1	概述	59
9.2	功能描述	59
9.2.1	模数转换器	60
9.2.2	ADC采样输出和功率波形输出	60
9.2.3	有功功率、无功功率和视在功率	60

9.2.4	电压、电流有效值	61
9.2.5	电压频率输出	61
9.2.6	灵活的潜动/起动设置	61
9.2.7	功率反向指示	62
9.2.8	防窃电	62
9.2.9	直流偏置自动校正和直流偏置校正寄存器	62
9.2.10	能量寄存器和脉冲输出单元	63
9.2.11	中断系统	64
9.3	寄存器	64
9.3.1	SFR寄存器	64
9.3.2	间接寄存器	66
9.4	校表过程	79
10	FLASH	81
10.1	概述	81
10.2	功能描述	81
10.2.1	PS1 Flash 读操作	81
10.2.2	Flash mass 擦除	81
10.2.3	Flash 页擦除	82
10.2.4	Flash字节编程	82
10.2.5	PM Flash 写保护	83
10.3	寄存器	83
10.4	一种简单实用的写DATAFLASH的方法	85
11	定时器	86
11.1	概述	86
11.2	TIMER0 和TIMER1	86
11.2.1	寄存器	86
11.2.2	功能描述	88
11.3	TIMER2	89
11.3.1	寄存器	89
11.3.2	功能描述	90
11.4	定时器/计数器时钟控制	92
12	RTC	93
12.1	功能描述	93
12.1.1	功能概述	93
12.1.2	时钟校正	93
12.1.3	时间和万年历	94
12.1.4	寄存器写保护功能	94
12.1.5	中断功能	95
12.2	寄存器	96
12.3	低功耗模式	99
12.4	I/O信号	99

13	LCD DRIVER.....	100
13.1	概述	100
13.2	功能描述	100
13.2.1	引脚复用	100
13.2.2	LCD驱动电压	101
13.2.3	LCD DUTY	101
13.2.4	LCD波形扫描频率	103
13.2.5	LCD对比度	103
13.2.6	LCD工作电流	104
13.2.7	SEGMENT输出波形	105
13.3	寄存器	106
13.4	低功耗模式	108
14	温度传感器	109
14.1	功能描述	109
14.1.1	温度计算公式	109
14.1.2	温度偏置校正	109
14.1.3	温度计算举例	109
14.2	寄存器	109
14.3	低功耗模式	110
15	UART	111
15.1	概述	111
15.2	功能描述	111
15.2.1	方式 0	111
15.2.2	方式 1	111
15.2.3	方式 2	114
15.2.4	方式 3	115
15.2.5	多机通信	116
15.2.6	UART1 38K红外调制	116
15.2.7	红外输出调制单元的时钟f _{irf}	116
15.2.8	RX0/RX1 配置为/INT0 和/INT1 的输入	117
15.3	寄存器	117
16	I2C	120
16.1	概述	120
16.2	功能描述	121
16.3	寄存器	122
16.4	I/O信号	125
17	GPIO	127
17.1	概述	127
17.2	复用配置	127
17.3	P2 口	128

17.4	P3 口	128
17.5	PORT A 口	129
17.6	PORT B 口	129
17.7	PORT C 口	129
17.8	PDM模式下GPIO的状态	130
18	PWM	131
18.1	概述	131
18.2	功能描述	131
18.2.1	Tone生成器:	132
18.2.2	脉宽调制PWM	132
18.3	寄存器	133
19	编程模式	135
19.1	概述	135
19.2	功能描述	135
19.2.1	用户应用模式UAM	135
19.2.2	Flash编程模式FPM	135
19.2.3	复位后模式选择	136
20	片上ICE支持	137
20.1	概述	137
20.2	功能描述	137
20.3	寄存器	138
21	机械规格	140
22	附录A 使用超级终端下载程序	141
22.1	烧写程序	141

1 总体概况

1.1 简介

ATT7027 是一颗低功耗高性能单相电能计量 SOC 芯片，片内集成单相计量、处理器、LCD 驱动器和实时时钟等功能。为单相多功能电能表提供高度集成的单芯片方案。

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：无铅LQFP64

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
- 无功电能测量误差小于0.5%，动态范围大于1000:1，支持IEC62053-23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%
- 三路ADC，支持单相两线制、单相三线制，支持防窃电功能，窃电阈值可灵活设置
- 提供三路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±3%（温度系数±25ppm/℃）
- 提供有功、无功电能脉冲输出，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
- 片内集成高频低频时钟晶体振荡电路和专为看门狗设置的RC振荡器
- 片内集成丰富的存储器资源，包括24K+8K的FLASH程序存储器，其中8K还可作为数据寄存器，擦除次数大于50万次；64字节有保护操作的Info FLASH存储器；256字节内部数据寄存器；1K字节外部数据寄存器，其中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
- 低功耗设计：正常工作时，耗电小于6毫安；低速运行下耗电小于80微安；休眠状态下耗电小于3微安
- 休眠模式下支持外部中断、外部复位等多种唤醒方式
- 片内集成可永不关断的硬件看门狗电路
- 低功耗实时时钟单元实现万年历功能，可软件补偿，有写保护设计
- 片内集成LCD驱动电路，最多可以支持4×24段
- 片内集成温度传感器
- 片内集成串行通讯、I2C、PWM等外设
- 片内固化2K控制程序，能够实现在线调试和在线编程

1.3 缩略语

表 1-3 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
PS0	Program memory Section0	指映射到 PM 空间的 24K 字节 code flash memory+8K 字节 data flash memory
PS1	Program memory Section1	指映射到 PM 空间的 64 字节 information flash memory
WDT	Watch Dog Timer	看门狗
RTC	Real Time Clock	实时时钟
GPIO	General Purpose IO	通用 IO
TPS	Temperature Sensor	温度传感器
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 Measurement Unit	电能计量单元
PSM	Power Saving Mode	省电模式
PDM	Power Down Mode	掉电模式
SCM	System Clock Management	系统时钟管理
UAM	User Application Mode	用户程序模式
SFPM	Serial Flash Programming Mode	串行编程模式
PFPM	Parallel Flash Programming Mode	并行编程模式

1.4 系统框图

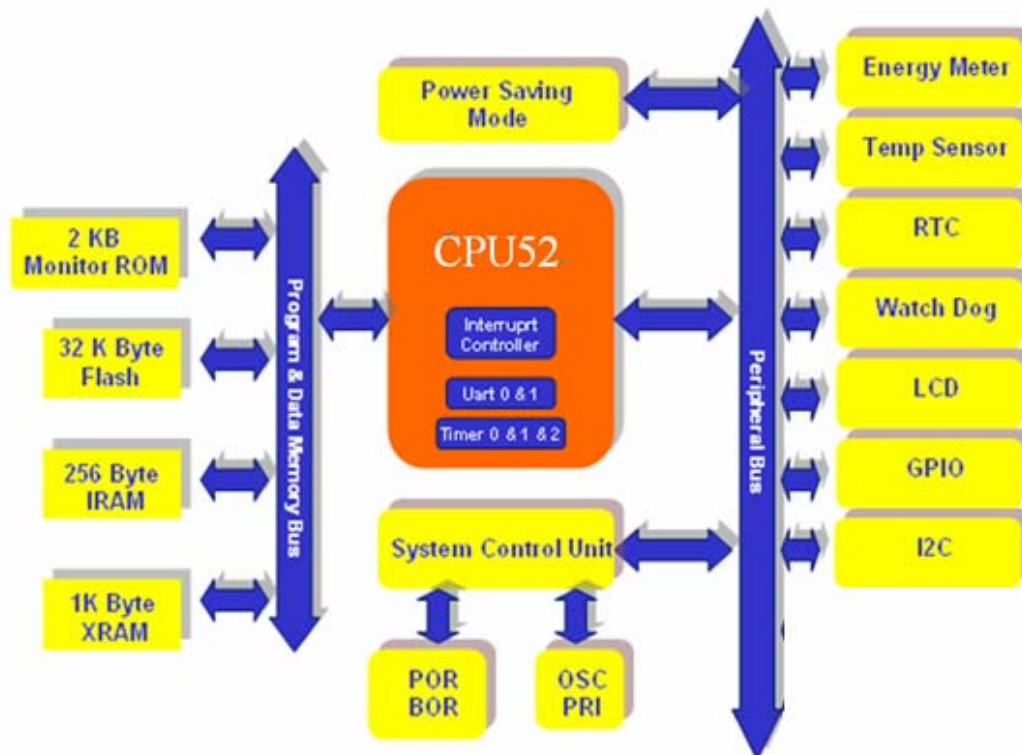


图 1-1 ATT7027 系统框图

1.5 引脚分布

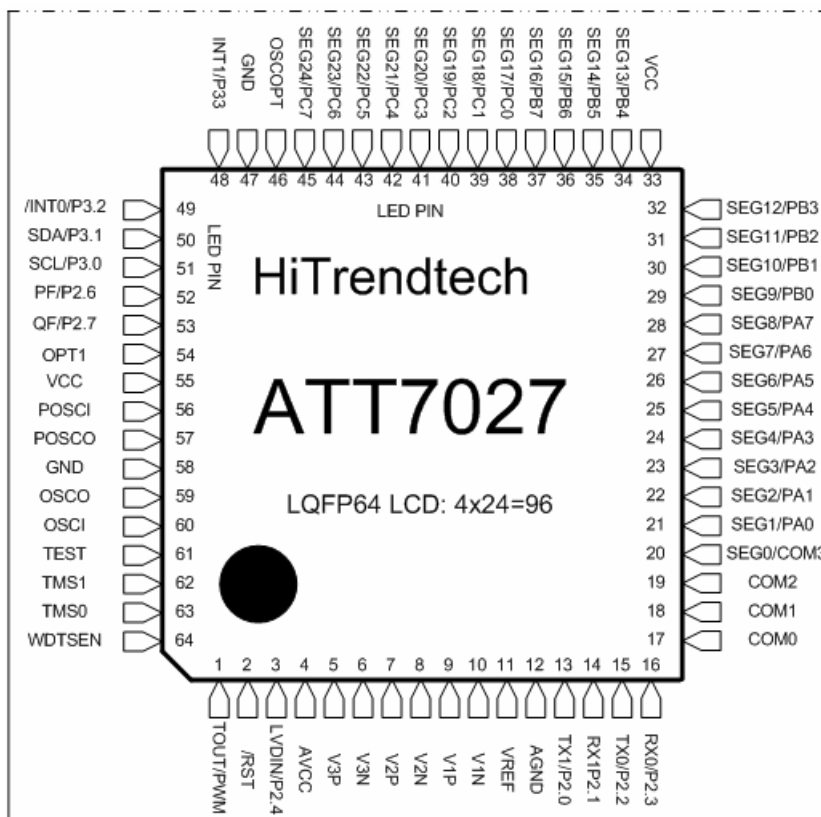


图 1-2 ATT7027 引脚分配

1.6 引脚描述

表 1-1 ATT7027 引脚描述

引脚	标识	特性	功能描述
1	TOUT	输出	由 RTC 时钟分频得到的实时钟校正输出
	PWM	输出	PWM 模块的输出端
2	/RST	输入	外部复位输入端，低电平有效；内部有 30K 上拉电阻
3	LVDIN	输入	LVD 模块的监测电压输入端
	P2.4	IO	通用 I/O 口
4	AVCC	电源	芯片模拟电源输入端，该引脚提供芯片模拟电路电源。正常应用范围：2.7V-3.6V
5	V3P	输入	电压信道的正模拟输入引脚，内部 ESD 保护电路，最大输入信号 $\pm 800\text{mV}$
6	V3N	输入	电压信道的负模拟输入引脚，内部 ESD 保护电路，最大输入信号 $\pm 800\text{mV}$
7	V2P	输入	电流信道 2 的正模拟输入引脚，内部 ESD 保护电路，最大输入信号 $\pm 800\text{mV}$
8	V2N	输入	电流信道 2 的负模拟输入引脚，内部 ESD 保护电路，最大输入信号 $\pm 800\text{mV}$
9	V1P	输入	电流信道 1 的正模拟输入引脚，内部 ESD 保护电路，最大输入信号 $\pm 800\text{mV}$
10	V1N	输入	电流信道 1 的负模拟输入引脚，内部 ESD 保护电路，最大输入信号 $\pm 800\text{mV}$
11	VREF	IO	参考电压，ADC 基准电压的输入输出引脚，外接 10uF 和 0.1uF。也可作为片外基准电压的输入端
12	AGND	电源	芯片模拟接地端。这是芯片模拟电路的接地参考点。该引脚应连接到 PCB 的模拟地。
13	TX1	输出	UART1 和红外的数据输出端
	P2.0	IO	通用 I/O 口
14	RX1	输入	UART1 和红外的数据输入端。内部无 30K 上拉电阻。外部做上拉处理。
	P2.1	IO	通用 I/O 口
15	TX0	输出	UART0 数据输出端
	P2.2	IO	通用 I/O 口
16	RX0	输入	UART0 数据输入端。内部有 30K 上拉电阻。
	P2.3	IO	通用 I/O 口
17	COM0	输出	LCD Common 驱动输出
18	COM1	输出	LCD Common 驱动输出
19	COM2	输出	LCD Common 驱动输出
20	SEG0	输出	LCD Segment 驱动输出
	COM3	输出	LCD Common 驱动输出
21	SEG1	输出	LCD Segment 驱动输出
	PA0	IO	通用 I/O 口
22	SEG2	输出	LCD Segment 驱动输出
	PA1	IO	通用 I/O 口
23	SEG3	输出	LCD Segment 驱动输出

	PA2	IO	通用 I/O 口
24	SEG4	输出	LCD Segment 驱动输出
	PA3	IO	通用 I/O 口
25	SEG5	输出	LCD Segment 驱动输出
	PA4	IO	通用 I/O 口
26	SEG6	输出	LCD Segment 驱动输出
	PA5	IO	通用 I/O 口
27	SEG7	输出	LCD Segment 驱动输出
	PA6	IO	通用 I/O 口
28	SEG8	输出	LCD Segment 驱动输出
	PA7	IO	通用 I/O 口
29	SEG9	输出	LCD Segment 驱动输出
	PB0	IO	通用 I/O 口
30	SEG10	输出	LCD Segment 驱动输出
	PB1	IO	通用 I/O 口
31	SEG11	输出	LCD Segment 驱动输出
	PB2	IO	通用 I/O 口
32	SEG12	输出	LCD Segment 驱动输出
	PB3	IO	通用 I/O 口
33	VCC	电源	芯片数字电源输入端
34	SEG13	输出	LCD Segment 驱动输出
	PB4	IO	通用 I/O 口
35	SEG14	输出	LCD Segment 驱动输出
	PB5	IO	通用 I/O 口
36	SEG15	输出	LCD Segment 驱动输出
	PB6	IO	通用 I/O 口
37	SEG16	输出	LCD Segment 驱动输出
	PB7	IO	通用 I/O 口
38	SEG17	输出	LCD Segment 驱动输出
	PC0	IO	通用 I/O 口
39	SEG18	输出	LCD Segment 驱动输出
	PC1	IO	通用 I/O 口
40	SEG19	输出	LCD Segment 驱动输出
	PC2	IO	通用 I/O 口
41	SEG20	输出	LCD Segment 驱动输出
	PC3	IO	通用 I/O 口
42	SEG21	输出	LCD Segment 驱动输出
	PC4	IO	通用 I/O 口
43	SEG22	输出	LCD Segment 驱动输出
	PC5	IO	通用 I/O 口
44	SEG23	输出	LCD Segment 驱动输出
	PC6	IO	通用 I/O 口
45	SEG24	输出	LCD Segment 驱动输出
	PC7	IO	通用 I/O 口
46	OSCOPT	输入	fosc选择引脚。OSCOPT=1, fosc选择源自外部晶体振荡电路；OSCOPT=0, fosc选择源自RC振荡器。fosc为RTC的时钟，在使用内部RTC时，推荐使用OSCOPT=1。
47	GND	电源	芯片数字接地端

48	/INT1	输入	外部中断 1，低电平有效，可配置为边沿触发或者电平触发
	P3.3	IO	通用 I/O 口
49	/INT0	输入	外部中断 0，低电平有效，可配置为边沿触发或者电平触发
	P3.2	IO	通用 I/O 口
50	SDA	IO	I2C 数据线
	P3.1	IO	通用 I/O 口
51	SCL	IO	I2C 时钟线
	P3.0	IO	通用 I/O 口
52	PF	输出	有功功率高频脉冲输出
	P2.6	IO	通用 I/O 口
53	QF	输出	无功功率高频脉冲输出
	P2.7	IO	通用 I/O 口
54	OPT1	输出	此引脚在应用时必须外接 10uf 电解电容和 0.1u 瓷片电容。
55	VCC	电源	芯片数字电源输入端
56	POSCI	输入	高频晶体振荡电路时钟输入，外接高频晶体
57	POSCO	输出	高频晶体振荡电路时钟输出，外接高频晶体
58	GND	电源	芯片数字接地端
59	OSCO	输出	低频晶体振荡电路时钟输出，外接 32.768Khz 低频晶体
60	OSCI	输入	低频晶体振荡电路时钟输入，外接 32.768Khz 低频晶体
61	TEST	输入	编程模式选择引脚，内部 30K 上拉
62	TMS1	输入	编程模式选择引脚，内部 30K 上拉
63	TMS0	输入	编程模式选择引脚，内部 30K 上拉
64	WDTSN	输入	WDT 使能输入端，高电平有效。内部有 30K 上拉电阻。

2 电气特性

2.1 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}	--	--	± 10	mA
I/O 最大输出灌电流	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

2.2 DC 参数

($V_{CC}=AV_{CC}=3.3V$, 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
输出高电平	V_{OH}	$V_{CC}-0.4$	--	--	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	
功耗 (Normal mode)	I_{ADD1}	5.5	--	5.6	mA	$f_{sys}=5.5MHz$, $f_{adc}=450KHz$, $V_{CC}=3.3V$
	I_{ADD2}	4.1	--	4.2	mA	$f_{sys}=1.3MHz$, $f_{adc}=230KHz$, $V_{CC}=3.3V$
功耗(Power Saving Mode)	I_{SDD1}	70	--	80	uA	$f_{sys}=32KHz$, $V_{CC}=3.3V$, With LCD on
	I_{SDD2}	65	--	75	uA	$f_{sys}=16KHz$, $V_{CC}=3.3V$, With LCD on
功耗(Power Down Mode)	I_{PDD1}	35	--	37	uA	With OSC, RTC, RC, LCD, LVD on
	I_{PDD2}	19	--	21	uA	With OSC, RTC, RC, LCD on
	I_{PDD3}	11	--	13	uA	With OSC, RTC, RC on
	I_{PDD4}	2	--	3	uA	With OSC, RTC on
I/O 上拉电阻	R_{PU}		30		k Ω	
POR 启动电压	V_{POR}	0	--	100	mV	

2.3 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

2.4 ADC

($V_{CC}=AV_{CC}=3.3V$, 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			$k\Omega$	
信噪比	SNR		86		dB	
-3dB 带宽	B_{-3dB}		14		kHz	

2.5 ADC 基准电压

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

测量项目	符号	最小	典型	最大	单位	测试条件
输出电压	V_{ref}		1.25		V	
温度系数	T_c		± 25		ppm/ $^{\circ}C$	

3 ATT7027 结构

3.1 概述

ATT7027 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。整个芯片架构如 [图 1-1](#)所示。

3.2 存储器

3.2.1 概述

ATT7027 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 3-1](#)所示：

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

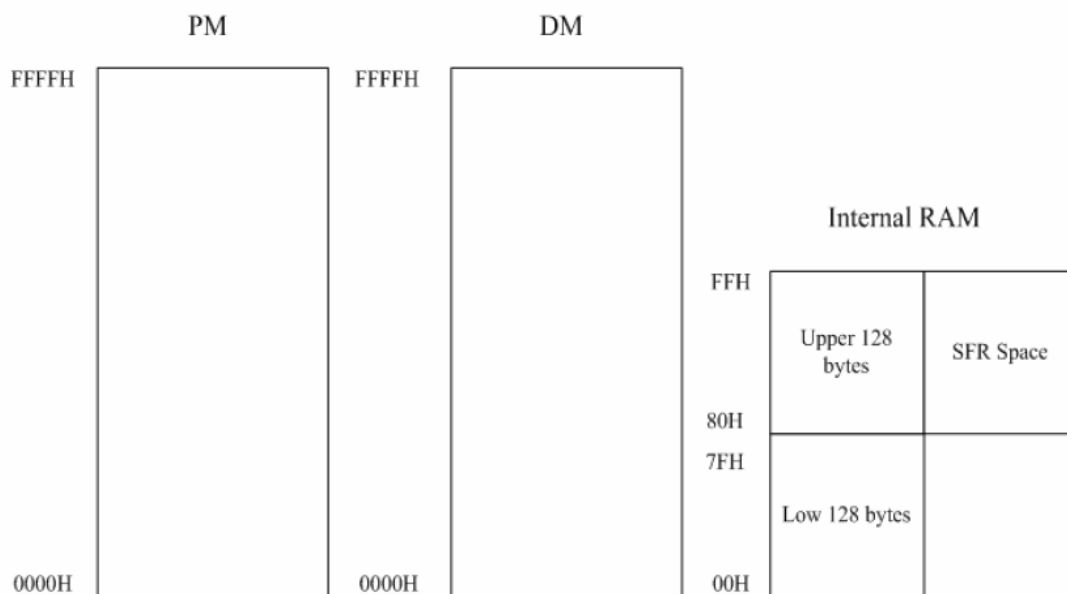


图 3-1 ATT7027 存储器地址空间

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始终要保留的常数，用 MOV_C 指令访问；数据存储器用来存放程序运行中需要的常数和变量，内部数据存储器用 MOV 指令访问，扩展数据存储器用 MOV_X 指令访问。

3.2.2 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几个区：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 字节的 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 3-2 所示。

其中 00H~1FH 共 32 个单元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 3-1。

当前程序使用的工作寄存器区是由程序状态字 PSW (SFR 0xD0H) 中的 RS1 和 RS0 来指示的，PSW 的状态和工作寄存器区对应关系见 表 3-9。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。

内部 RAM 的 20H~2FH 为位寻址区（见 表 3-2），这 16 个单元的每一位都有一个位地址，位地址范围为 00H~7FH。位寻址区的每一位都可以视作软件触发器，由程序直接进行位处理。通常把各种程序状态标志、位控制变量设在位寻址区。同样，位寻址单元也可以作为一般的数据缓冲器使用。

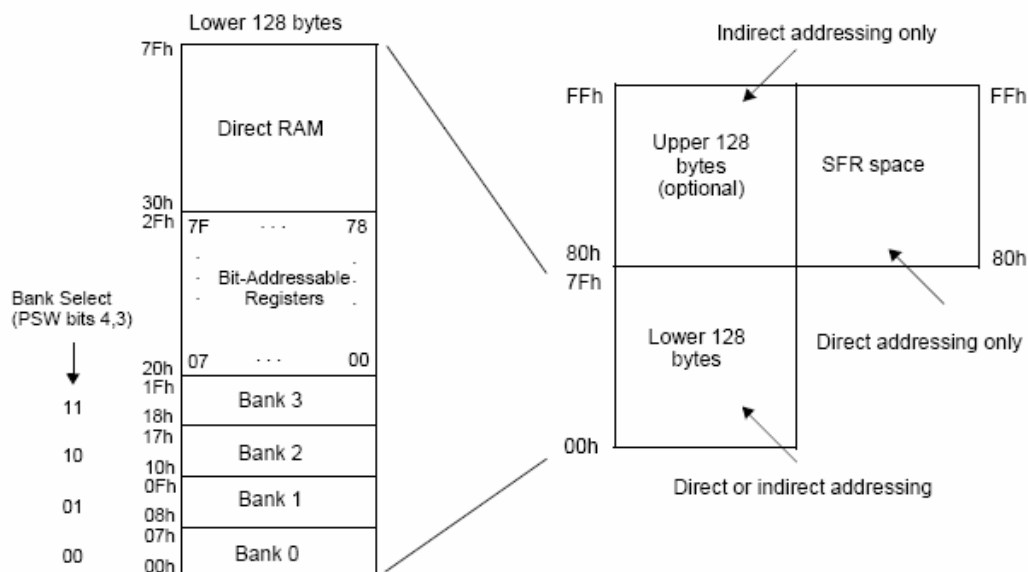


图 3-2 内部 RAM 的功能结构

表 3-1 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

表 3-2 内部 RAM 的位寻址区

字节地址	位地址							
	D7	D6	D5	D4	D3	D2	D1	D0
2FH	7F	7E	7D	7C	7B	7A	79	78
2EH	77	76	75	74	73	72	71	70

2DH	6F	6E	6D	6C	6B	6A	69	68
2CH	67	66	65	64	63	62	61	60
2BH	5F	5E	5D	5C	5B	5A	59	58
2AH	57	56	55	54	53	52	51	50
29H	4F	4E	4D	4C	4B	4A	49	48
28H	47	46	45	44	43	42	41	40
27H	3F	3E	3D	3C	3B	3A	39	38
26H	37	36	35	34	33	32	31	30
25H	2F	2E	2D	2C	2B	2A	29	28
24H	27	26	25	24	23	22	21	20
23H	1F	1E	1D	1C	1B	1A	19	18
22H	17	16	15	14	13	12	11	10
21H	0F	0E	0D	0C	0B	0A	09	08
20H	07	06	05	04	03	02	01	00

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

3.2.3 扩展存储器映射

ATT7027 片内的扩展的存储器资源包括：

- 24K-bytes 的 Code Flash memory，映射为 PM
- 8K-bytes 的 Data Flash memory，既可映射为 PM，又可映射为 DM
- 64-bytes Info Flash memory，映射为 PM
- 1K-bytes XRAM，既可映射为 PM，又可映射为 DM
- 2K-bytes Monitor ROM，映射为 PM

在用户模式（UAM）下，PM 和 DM 存储器映射如图 3-3 所示。

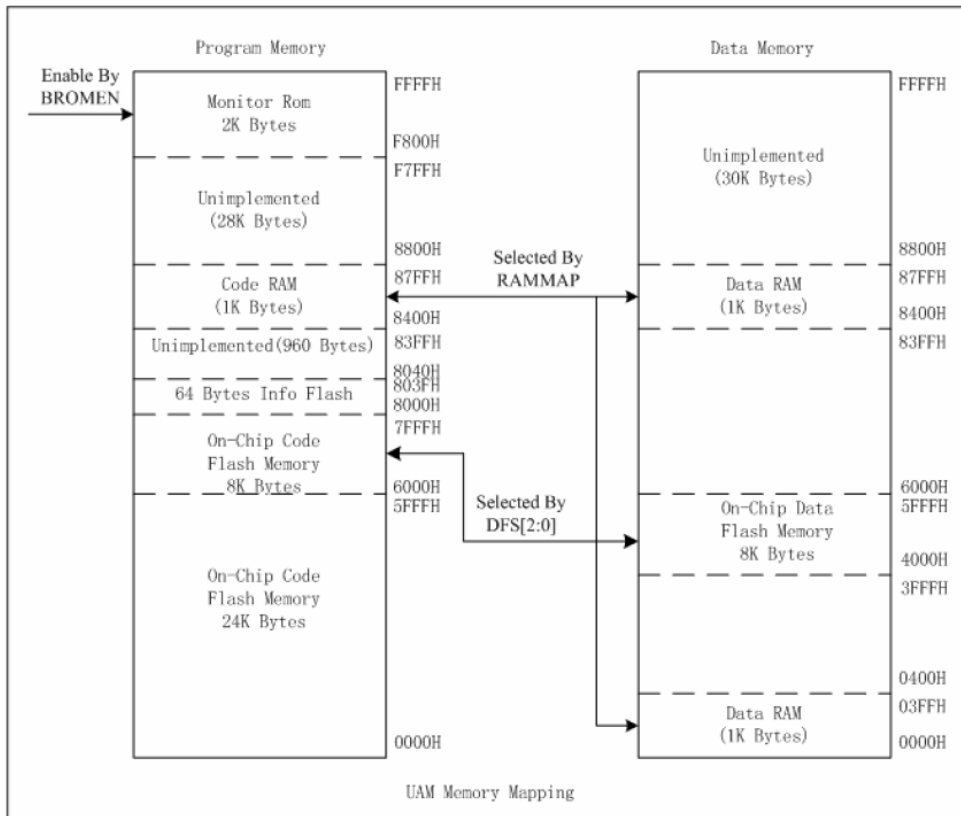


图 3-3 UAM 下 PM 和 DM 存储器映射

1. Monitor Rom

在用户模式下，Monitor Rom 地址空间为 PM: F800H-FFFFH；

Monitor Rom映射被BROMEN（表 3-3）控制。当BROMEN=1 时，UAM模式下可以读取Monitor Rom，即用户可以调用Monitor Rom中的程序，以支持用户程序编程IAP功能。当BROMEN=0 时，UAM模式下不可读取Monitor Rom，此时如果用户程序访问Monitor Rom地址，将返回 00H数据。

2. XRAM

XRAM地址映射可以通过RAMMAP（表 3-4）进行选择。

当 RAMMAP=0 时，RAM 地址空间为 DM 0000-03FFH，RAM 仅作为数据 RAM。而当 RAMMAP=1 时，RAM 地址空间为 8400H-87FFH，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 8400H-87FFH 是一个数据与程序共用的地址空间。

3. Info Flash memory (PS1)

这部分的 64 字节Flash占用PM地址空间(PS1)：8000H-803FH。

PS1 映射被PS1MAP(表 10-4)控制。当PS1MAP=0 时，PS1 不可直接通过地址访问，访问 8000H-803FH地址时，返回数据为 00H；当PS1MAP=1 时，PS1 被映射到 8000H-803FH 地址，可以通过地址 8000H-803FH直接读取/修改PS1 的内容。

4. Code Flash Memory

这部分 24K Flash是用于存储程序，占用PM地址空间(PS0)：0000H-5FFFH。

5. Data Flash Memory

这部分的 8K Flash通过DFS[2:0]（表 10-2）可以定义为PM和DM两类功能。

当 DFS[2:0]=111 时，8K-bytes Data Flash Memory 映射为 DM 地址空间：4000-5FFFH；

当 DFS[2:0]=0xx 时，Data Flash Memory 映射为 PM 地址空间(PS0)：6000-7FFFH；

复位状态，DFS[2:0]=000，Code Flash 和 Data Flash 组成连续的 32K 程序存储器，地址为 0000-7FFFH。

3.2.4 程序存储器

程序存储器用来存放固定程序和表格常数等。程序存储器以程序计数器PC做地址指针，通过 16 位地址总线，可寻址地址空间为 64K。如上节所述，ATT7027 程序存储器包括：2K 字节 Monitor ROM、64 字节 Info Flash、24K 字节 Code Flash、8K Data Flash(DFS[2:0]=0XX 时)、1K 字节 Code RAM(REMAP=1 时)。

Monitor Rom中的软件主要实现ICE功能，具体说明详见 片上ICE支持 章节。

Info Flash：系统配置用，读写擦除次数可达 50 万次，可用来存放出厂信息、表参数等。

Code Flash：存放用户程序。用户模式下，复位后，程序计数器 PC 的内容为 0000H，因此系统从 Code Flash 0000H 开始取指并执行程序。通常该单元中存放一条跳转指令，而用户程序从跳转地址开始存放。

Data Flash：DFS[2:0]=0XX 时可作程序存储器，和 Code Flash 组成连续的 32K 程序存储空间。

Code RAM：RAMMAP=1 时，XRAM 可作程序存储器。作为程序 RAM 时可以临时存放程序，CPU52 运行 RAM 程序时，可以支持将整个用户程序进行升级更新。

3.2.5 数据存储器

数据存储器用来存放程序的数据或者运算结果。如前所述，ATT7027 扩展数据存储器包括：8K 字节 Data Flash memory 和 1K 字节 XRAM。

Data Flash：DFS=111 时映射为数据存储器。Data Flash memory 是非易失性高可靠性存储器，擦写次数可达 500,000 次，在常温下数据保存时间达 100 年。在电表应用中可替代片外 E2PROM，用来存放电能计量数据。

Data RAM: RAMMAP=0 时, XRAM 可作数据存储器, 用来存放用户的临时数据。针对电表应用, XRAM 设计为两部分, 一部分 256 字节 XRAM, 映射为 0000H~00FFH (RAMMAP=0) 或者 8400H~84FFH (RAMMAP=1), 在 PDM 下不掉电, 可用来在 PDM 下存储用户一些重要现场数据; 另一部分 768 字节 XRAM, 映射为 0100H~03FFH (RAMMAP=0) 或者 8500H~87FFH (RAMMAP=1), 在 PDM 下掉电, 只能在节电模式和正常模式下使用。

3.2.6 寄存器

1. BROMEN 控制位 (写保护)

表 3-3 BROMEN 控制位(0xBEH, FMCFG.2)

FMCFG			Address: BEH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	DFS2	DFS1	DFS0	0	0	BROMEN	PMLOCK	RSLOCK
Write:				X	X			
Reset:	0	0	0	0	0	0	0	0

说明: BROMEN=1 时, UAM 模式下可以读取 Monitor Rom; 当 BROMEN=0 时, UAM 模式下不可读取 Monitor Rom, 此时如果用户程序访问 Monitor Rom 地址, 将返回 00H 数据。

2. RAMMAP 控制位

表 3-4 RAMMAP 控制位(0xF6H, MCON.0)

Memory Control Register (MCON)			Address: F6H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	0	0	0	0	RAMMAP
Write:		X	X	x	X	X	X	
Reset:		0	0	0	0	0	0	

说明: 当 REMAP=1 时, XRAM 作为程序存储器; 当 RAMMAP=0 时, XRAM 作为数据存储器。

3.3 指令系统

3.3.1 寻址方式

ATT7027 寻址方式和标准 8052 相同, 有下列几种:

1. 立即寻址

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

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

2. 直接寻址

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

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

MOV A, 70H ; (70H)->A

指令功能是把内部 RAM 70H 单元的内容送入累加器 A

2) 位地址空间, 例如

MOV C, 00H

3) SFR

SFR 只能用直接寻址的方式进行访问。例如:

MOV IE, #85H ; 立即数 85H->中断允许寄存器 IE

3. 寄存器寻址

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

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

4. 寄存器间接寻址

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

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

MOV A, @R0 ; ((R0))->A

指令功能是把 R0 所指出的内部 RAM 单元中的内容送入累加器 A

5. 基寄存器加变址寄存器间接寻址

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

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

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

6. 相对寻址

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

JC rel ; C=1, 跳转

3.3.2 指令集

ATT7027 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 3-5 定义了表 3-6 中用到的符号，表 3-6 分类列出了 ATT7027 的指令集、字节数、机器周期等。

表 3-5 ATT7027 指令集列表符号说明

符号	功能
A	累加器
(A)	累加器内容
Rn	工作寄存器 R0-R7
(Rn)	工作寄存器的内容
Ri	i=0, 1, 数据指针 R0 或 R1
(Ri)	R0 或 R1 的内容
((Ri))	R0 或 R1 的指出的单元内容
@Ri	R0 或者 R1 指针指向的内部寄存器（除了 MOVX 指令）
(X)	某一寄存器的内容
X	某一寄存器
((X))	某一寄存器指出的单元内容
direct	直接地址单元
(direct)	直接地址指出的单元内容
rel	相对偏移量，带符号的（2 的补码）8 位偏移字节

bit	位地址
#data	8 位立即数
#data 16	16 位立即数
addr 16	16 位绝对地址
addr 11	页面地址
←	数据传送方向
∧	逻辑与
∨	逻辑或
⊕	逻辑异或
✓	对标志位产生影响
×	对标志位不产生影响

表 3-6 ATT7027 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	C _Y			
算术运算指令								
ADD A, Rn	$A \leftarrow (A) + (Rn)$	√	√	√	√	1	1	28-2F
ADD A, direct	$A \leftarrow (A) + (direct)$	√	√	√	√	2	2	25
ADD A, @Ri	$A \leftarrow (A) + ((Ri))$	√	√	√	√	1	1	26-27
ADD A, #data	$A \leftarrow (A) + data$	√	√	√	√	2	2	24
ADDC A, Rn	$A \leftarrow (A) + (Rn) + (C_Y)$	√	√	√	√	1	1	38-3F
ADDC A, direct	$A \leftarrow (A) + (direct) + (C_Y)$	√	√	√	√	2	2	35
ADDC A, @Ri	$A \leftarrow (A) + ((Ri)) + (C_Y)$	√	√	√	√	1	1	36-37
ADDC A, #data	$A \leftarrow (A) + data + (C_Y)$	√	√	√	√	2	2	34
SUBB A, Rn	$A \leftarrow (A) - (Rn) - (C_Y)$	√	√	√	√	1	1	98-9F
SUBB A, direct	$A \leftarrow (A) - (direct) - (C_Y)$	√	√	√	√	2	2	95
SUBB A, @Ri	$A \leftarrow (A) - ((Ri)) - (C_Y)$	√	√	√	√	1	1	96-97
SUBB A, #data	$A \leftarrow (A) - data - (C_Y)$	√	√	√	√	2	2	94
INC A	$A \leftarrow (A) + 1$	√	×	×	×	1	1	04
INC Rn	$Rn \leftarrow Rn + 1$	×	×	×	×	1	1	08-0F
INC direct	$direct \leftarrow direct + 1$	×	×	×	×	2	2	05
INC @Ri	$(Ri) \leftarrow ((Ri)) + 1$	×	×	×	×	1	1	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	2	18-1F
DEC direct	$A \leftarrow (direct) - 1$	×	×	×	×	1	1	15
DEC @Ri	$A \leftarrow ((Ri)) - 1$	×	×	×	×	1	3	16-17
MUL AB	$AB \leftarrow (A) \cdot (B)$	√	√	×	√	1	5	A4
DIV AB	$AB \leftarrow (A) / (B)$	√	√	×	√	1	5	84
DA A	对 A 进行十进制调整	√	√	√	√	1	1	D4
逻辑运算指令								
ANL A, Rn	$A \leftarrow (A) \wedge (Rn)$	√	×	×	×	1	1	58-5F
ANL A, direct	$A \leftarrow (A) \wedge (direct)$	√	×	×	×	2	2	55
ANL A, @Ri	$A \leftarrow (A) \wedge ((Rn))$	√	×	×	×	1	1	56-57
ANL A, #data	$A \leftarrow (A) \wedge data$	√	×	×	×	2	2	54
ANL direct, A	$direct \leftarrow (direct) \wedge A$	×	×	×	×	2	2	52

ANL direct, #data	$\text{direct} \leftarrow (\text{direct}) \wedge \text{data}$	×	×	×	×	3	3	53
ORL A, Rn	$A \leftarrow (A) \vee (Rn)$	√	×	×	×	1	1	48-4F
ORL A, direct	$A \leftarrow (A) \vee (\text{direct})$	√	×	×	×	2	2	45
ORL A, @Ri	$A \leftarrow (A) \vee ((Ri))$	√	×	×	×	1	1	46-47
ORL A, #data	$A \leftarrow (A) \vee \text{data}$	√	×	×	×	2	2	44
ORL direct, A	$\text{direct} \leftarrow (\text{direct}) \vee A$	×	×	×	×	2	2	42
ORL direct, #data	$\text{direct} \leftarrow (\text{direct}) \vee \text{data}$	×	×	×	×	3	3	43
XRL A, Rn	$A \leftarrow (A) \oplus (Rn)$	√	×	×	×	1	1	68-6F
XRL A, direct	$A \leftarrow (A) \oplus (\text{direct})$	√	×	×	×	2	2	65
XRL A, @Ri	$A \leftarrow (A) \oplus ((Ri))$	√	×	×	×	1	1	66-67
XRL A, #data	$A \leftarrow (A) \oplus \text{data}$	√	×	×	×	2	2	64
XRL direct, A	$\text{direct} \leftarrow (\text{direct}) \oplus A$	×	×	×	×	2	2	62
XRL direct, #data	$\text{direct} \leftarrow (\text{direct}) \oplus \text{data}$	×	×	×	×	3	3	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
数据传送指令								
MOV A, Rn	$A \leftarrow (Rn)$	√	×	×	×	1	1	E8-EF
MOV A, direct	$A \leftarrow (\text{direct})$	√	×	×	×	2	2	E5
MOV A, @Ri	$A \leftarrow ((Ri))$	√	×	×	×	1	1	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	2	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	2	88-8F
MOV direct, direct	$\text{direct} \leftarrow (\text{direct})$	×	×	×	×	3	3	85
MOV direct, @Ri	$\text{direct} \leftarrow ((Ri))$	×	×	×	×	2	2	86-87
MOV direct, #data	$\text{direct} \leftarrow \text{data}$	×	×	×	×	3	3	75
MOV @Ri, A	$((Ri)) \leftarrow (A)$	×	×	×	×	1	1	F6-F7
MOV @Ri, direct	$(Ri) \leftarrow (\text{direct})$	×	×	×	×	2	2	A6-A7
MOV @Ri, #data	$(Ri) \leftarrow \text{data}$	×	×	×	×	2	2	76-77
MOV DPTR, #data	$\text{DPTR} \leftarrow \text{data}$	×	×	×	×	3	3	90
MOVC A, @A+DPTR	$A \leftarrow ((A) + (\text{DPTR}))$	√	×	×	×	1	3	93
MOVC A, @A+PC	$A \leftarrow ((A) + (\text{PC}))$	√	×	×	×	1	3	83
MOVB A, @Ri	$A \leftarrow ((Ri) + P2)$	√	×	×	×	1	2-9*	E2-E3
MOVB A, @DPTR	$A \leftarrow ((\text{DPTR}))$	√	×	×	×	1	2-9*	E0
MOVB @Ri, A	$((Ri) + P2) \leftarrow A$	×	×	×	×	1	2-9*	F2-F3
MOVB A, @DPTR	$A \leftarrow ((\text{DPTR}))$	×	×	×	×	1	2-9*	F0
PUSH direct	$\text{SP} \leftarrow \text{SP} + 1, (\text{SP}) \leftarrow (\text{direct})$	×	×	×	×	2	2	C0
POP direct	$\text{direct} \leftarrow ((\text{SP})), \text{SP} \leftarrow (\text{SP}) - 1$	×	×	×	×	2	2	D0
XCH A, Rn	$(A) \leftrightarrow (Rn)$	√	×	×	×	1	1	C8-CF
XCH A, direct	$(A) \leftrightarrow (\text{direct})$	√	×	×	×	2	2	C5

XCH A, @Ri	$(A) \leftrightarrow ((Ri))$	√	×	×	×	1	1	C6-C7
XCHD A, @Ri	$(A)_{0-3} \leftrightarrow ((Ri))_{0-3}$	√	×	×	×	1	1	D6-D7
位操作指令								
CLR C	$C_Y \leftarrow 0$	×	×	×	√	1	1	C3
CLR bit	$bit \leftarrow 0$	×	×	×		2	2	C2
SETB C	$C_Y \leftarrow 1$	×	×	×	√	1	1	D3
SETB bit	$bit \leftarrow 1$	×	×	×		2	2	D2
CPL C	$C_Y \leftarrow (\overline{C_Y})$	×	×	×	√	1	1	B3
CPL bit	$bit \leftarrow (\overline{bit})$	×	×	×		2	2	B2
ANL C, bit	$C_Y \leftarrow (C_Y) \wedge (bit)$	×	×	×	√	2	2	82
ANL C, /bit	$C_Y \leftarrow (C_Y) \wedge (\overline{bit})$	×	×	×	√	2	2	B0
ORL C, bit	$C_Y \leftarrow (C_Y) \vee (bit)$	×	×	×	√	2	2	72
ORL C, /bit	$C_Y \leftarrow (C_Y) \vee (\overline{bit})$	×	×	×	√	2	2	A0
MOV C, bit	$C_Y \leftarrow (bit)$	×	×	×	√	2	2	A2
MOV bit, C	$bit \leftarrow (C_Y)$	×	×	×	×	2	2	92
控制转移指令								
ACALL addr11	$PC \leftarrow (PC)+2, SP \leftarrow (SP)+1,$ $(SP) \leftarrow (PC)_L, SP \leftarrow (SP)+1,$ $(SP) \leftarrow (PC)_H, PC_{10-0} \leftarrow \text{addr11}$	×	×	×	×	2	3	11-F1
LCALL addr16	$PC \leftarrow (PC)+2, SP \leftarrow (SP)+1,$ $(SP) \leftarrow (PC)_L, SP \leftarrow (SP)+1,$ $(SP) \leftarrow (PC)_H, PC_{10-0} \leftarrow \text{addr16}$	×	×	×	×	3	4	12
RET	$(PC)_H \leftarrow ((SP)), SP \leftarrow (SP)-1,$ $(PC)_L \leftarrow ((SP)), SP \leftarrow (SP)-1$	×	×	×	×	1	4	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	3	32
AJMP addr11	$PC_{10-0} \leftarrow \text{addr11}$	×	×	×	×	3	4	01-E1
LJMP addr16	$PC \leftarrow \text{addr16}$	×	×	×	×	2	3	02
SJMP rel	$PC \leftarrow PC + \text{rel}$	×	×	×	×	2	3	80
JC rel	$PC \leftarrow PC+2$, 若 $C_Y=1$, 则 $PC \leftarrow PC + \text{rel}$	×	×	×	×	2	3	40
JNC rel	$PC \leftarrow PC+2$, 若 $C_Y=0$, 则 $PC \leftarrow PC + \text{rel}$	×	×	×	×	2	3	50
JB bit, rel	$PC \leftarrow PC+3$, 若 $(bit)=1$, 则 $PC \leftarrow PC + \text{rel}$	×	×	×	×	3	4	20
JNB bit, rel	$PC \leftarrow PC+3$, 若 $(bit)=0$, 则 $PC \leftarrow PC + \text{rel}$	×	×	×	×	3	4	30
JBC bit, rel	$PC \leftarrow PC+3$, 若 $(bit)=1$, 则 $bit \leftarrow 0, PC \leftarrow (PC) + \text{rel}$	×	×	×	×	3	4	10
JMP @A+DPTR	$PC \leftarrow (A) + (DPTR)$	×	×	×	×	1	3	73
JZ rel	$PC \leftarrow PC+2$, 若 $(A)=0$, $PC \leftarrow (PC) + \text{rel}$	×	×	×	×	2	3	60
JNZ rel	$PC \leftarrow PC+2$, 若 $(A) \neq 0$, $PC \leftarrow (PC) + \text{rel}$	×	×	×	×	2	3	70
CJNE A, direct, rel	$PC \leftarrow PC+3$, 若 $(A) \neq$ $(direct)$, 则 $PC \leftarrow (PC) + \text{rel}$	×	×	×	×	3	4	B5
CJNE A, #d, rel	$PC \leftarrow PC+3$, 若 $(A) \neq$ $data$, 则 $PC \leftarrow (PC) + \text{rel}$	×	×	×	×	3	4	B4

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

3.3.3 指令时序

ATT7027 的指令时序和标准工业 8052 有差异，机器周期和 8052 不同。如图 3-4 所示，ATT7027 的机器周期为 4 个 CPU 时钟振荡周期 C1、C2、C3、C4，根据指令的不同，每个时钟周期内 CPU 完成不同的操作，而标准 8052 的机器周期为 12 个振荡周期。对于大部分指令而言，ATT7027 指令执行速度比标准 8052 提高了 3 倍。

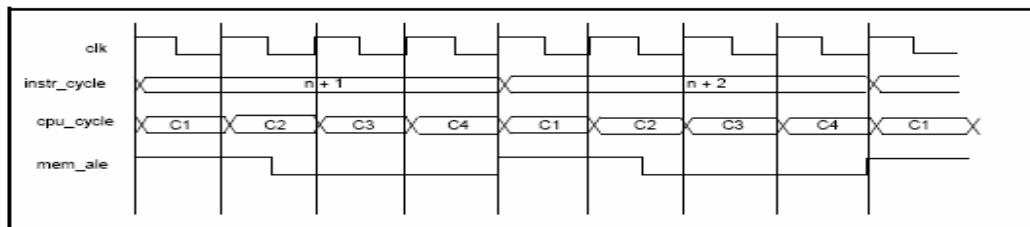


图 3-4 CPU 机器周期

ATT7027 某些指令的指令周期比标准 8052 长，标准 8052 除 MUL 和 DIV 指令外，其他所有指令都是 1-2 个机器周期，而 ATT7027 的某些指令的机器周期可达 3-5。例如，在标准 8052 中，执行指令 MOVX A, @DPTR 和 MOV direct, direct 都占有 2 个机器周期（24 个时钟振荡周期），对于 ATT7027，执行 MOVX A, @DPTR 占有 2 个机器周期（8 个时钟振荡周期），执行 MOV direct, direct 占用 3 个机器周期（12 个时钟振荡周期）。两条指令的执行都比标准 8052 快，但需要不同的时钟周期。

对于实时性强的事件，用户可通过 [表 3-6](#) 的机器周期一栏计算程序执行指令的时间。[表 3-6](#) 字节数一栏说明在执行指令期间 CPU 访问存储器的次数。在大多数情况下，字节数等于机器周期，但也有例外，如 DIV 和 MUL 指令的机器周期数大于指令字节数。

默认情况下，ATT7027 的定时器/计数器每 12 个时钟振荡周期增加 1，和标准 8052 相同。定时器也可设置为 4 个时钟振荡周期增加 1。

3.4 特殊功能寄存器

3.4.1 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，ATT7027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。[表3-7](#)列出这些SFR的助记符、地址和位名称。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ATT7025 CPU52内部寄存器，其他寄存器是ATT7025的扩展寄存器。

表3-7 ATT7027 特殊功能寄存器

地址	助记符	位名称							
		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x80	P0	*	*	*	*	*	*	*	*
0x81	SP	-	-	-	-	-	-	-	-
0x82	DPL0	-	-	-	-	-	-	-	-
0x83	DPH0	-	-	-	-	-	-	-	-
0x84	DPL1	-	-	-	-	-	-	-	-
0x85	DPH1	-	-	-	-	-	-	-	-
0x86	DPS	0	0	0	0	0	0	0	SEL
0x87	PCON	SMOD0	-	1	1	GF1	GF0	STOP	IDLE
0x88	TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
0x89	TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0
0x8A	TL0	-	-	-	-	-	-	-	-
0x8B	TL1	-	-	-	-	-	-	-	-
0x8C	TH0	-	-	-	-	-	-	-	-
0x8D	TH1	-	-	-	-	-	-	-	-
0x8E	CKCON	-	-	T2M	T1M	T0M	MD2	MD1	MD0
0x8F	SPC_FNC	0	0	0	0	0	0	0	WRS
0x90	P1	*	*	*	*	*	*	*	*
0x91	EXIF	IE5	IE4	IE3	IE2	1	0	0	0
0x92	MPAGE	-	-	-	-	-	-	-	-
0x93	PTA	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
0x94	PTB	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
0x95	PTC	PTC7	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
0x96	PTD	*	*	*	*	*	*	*	*
0x97	PTE	*	*	*	*	*	*	*	*
0x98	SCON0	SM0_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0
0x99	SBUF0	-	-	-	-	-	-	-	-
0x9A	BWPR	PASS4	PASS3	PASS2	PASS1	PASS0	BITPS	PMOD1	PMOD0
0x9B	DDRA	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
0x9C	DDRB	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
0x9D	DDRC	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
0x9E	DDRD	*	*	*	*	*	*	*	*
0x9F	DDRE	*	*	*	*	*	*	*	*
0xA0	P2	P2.7	P2.6	*	P2.4	P2.3	P2.2	P2.1	P2.0
0xA1	DDRP0	*	*	*	*	*	*	*	*
0xA2	DDRP1	*	*	*	*	*	*	*	*
0xA3	DDRP2	DDRP27	DDRP26	*	DDRP24	DDRP23	DDRP22	DDRP21	DDRP20
0xA4	DDRP3	*	*	*	*	DDRP33	DDRP32	DDRP31	DDRP30
0xA5	LEDP0	*	*	*	*	*	*	*	*
0xA6	LEDP1	*	*	*	*	*	*	*	*
0xA7	LEDP3	*	*	*	*	LEDP33	LEDP32	LEDP31	LEDP30
0xA8	IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
0xA9	EMUIE	PFIE	QFIE	SFIE	SPLIE	ZXIE	0	0	0
0xAA	RTCIE	ALMIE	SCNTIE	RTC2IE	RTC1IE	DAYIE	HRIE	MINIE	SECIE
0xAB	SCIE	0	0	0	I2CIE	SPTIE	SPRIE	SPERIE	0
0xAC	LVIE	0	0	0	0	0	0	LVDIE	BORIE

0xAD	KEYIE	*	*	*	*	*	*	*	*
0xAE	LEDC	LEDC7	LEDC6	LEDC5	LEDC4	LEDC3	LEDC2	LEDC1	LEDC0
0xAF	LEDE	*	*	*	*	*	*	*	*
0xB0	P3	*	*	*	*	P3.3	P3.2	P3.1	P3.0
0xB1	EMUIF	PFIF	QFIF	SFIF	SPLIF	ZXIF	0	0	0
0xB2	RTCIF	ALMF	SCNTF	RTC2F	RTC1F	DAYF	HRF	MINF	SECF
0xB3	SCIIF	I2RXIF	I2TXIF	0	I2ALIF	SPTEIF	SPRFIF	SPOVIF	SPMDIF
0xB4	LVIIF	0	0	0	0	0	0	LVDIF	BORIF
0xB5	LVDCON	LVDS3	LVDS2	LVDS1	LVDS0	0	0	0	0
0xB6	SYSSCR	LOCK	BOROUT	LVDOUT	0	WDTCLR3	WDTCLR2	WDTCLR1	WDTCLR0
0xB7	RSTSR	POR	RST	WDT	BOR	WKR	0	0	0
0xB8	IP	1	PS1	PT2	PS0	PT1	PX1	PT0	PX0
0xB9	LCDCFG	0	0	PTCH	PTCL	PTBH	PTBL	PTAH	PTAL
0xBA	P02CFG	P267	P245	P223	P201	0	0	0	0
0xBB	P3CFG	*	*	*	*	INT1	INT0	IRTX1	0
0xBC	BORCFG	PRION	WKRST	BORPDM	BORRST	BORVS1	BORVS0	PDM_RC	1
0xBD	CLKCFG	WDT2	WDT1	WDT0	WDTEN	PRIP1	PRIP0	SYSC1	SYSC0
0xBE	FMCFG	DFS2	DFS1	DFS0	0	0	BROMEN	PMLOCK	RSLOCK
0xBF	SUPDC	0	PDTPS	PDEMU	PDBOR	PDLVD	PDLCD	PDI2C	1
0xC0	SCON1	SM0_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1
0xC1	SBUF1	-	-	-	-	-	-	-	-
0xC2	LCDCLK	0	FCSET1	FCSET0	DUTY1	DUTY0	0	LCLK1	LCLK0
0xC3	LCDCR	0	0	FC	LC	LCCON3	LCCON2	LCCON1	LCCON0
0xC4	LADR	0	0	LADR5	LADR4	LADR3	LADR2	LADR1	LADR0
0xC5	LDAT	LDAT7	LDAT6	LDAT5	LDAT4	LDAT3	LDAT2	LDAT1	LDAT0
0xC6	RXCON	RXCON7	RXCON6	RXCON5	RXCON4	RXCON3	RXCON2	RXCON1	RXCON0
0xC7	PDMSR	PDMSR5	PDMSR4	PDMSR3	PDMSR2	PDMSR1	PDMSR0	PDC	PPDC
0xC8	T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
0xC9	TEMPDR	TPDR7	TPDR6	TPDR5	TPDR4	TPDR3	TPDR2	TPDR1	TPDR0
0xCA	RCAP2L	-	-	-	-	-	-	-	-
0xCB	RCAP2H	-	-	-	-	-	-	-	-
0xCC	TL2	-	-	-	-	-	-	-	-
0xCD	TH2	-	-	-	-	-	-	-	-
0xD0	PSW	CY	AC	F0	RS1	RS0	OV	F1	P
0xD8	EICON	SMOD1	1	EPFI	PFI	WDTI	0	0	0
0xE0	ACC	-	-	-	-	-	-	-	-
0xE8	EIE	1	1	1	EWD	EX5	EX4	EX3	EX2
0xF0	B	-	-	-	-	-	-	-	-
0xF1	RTCCON	0	TOUTEN1	TOUTEN0	RTCTEN	TESL	TOUT2	TOUT1	TOUT0
0xF2	RTCCAL	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
0xF3	SECCNT	0	0	SCNT5	SCNT4	SCNT3	SCNT2	SCNT1	SCNT0
0xF4	ALMR	0	0	AM5	AM4	AM3	AM2	AM1	AM0
0xF5	ALHR	0	0	0	AH4	AH3	AH2	AH1	AH0
0xF6	FCCON	BPSEL	0	0	0	0	0	0	RAMMAP
0xF7	KEYCR	*	*	*	*	*	*	*	*
0xF8	EIP	1	1	1	PWDI	PX5	PX4	PX3	PX2
0xF9	SECR	0	0	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
0xFA	MINR	0	0	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0
0xFB	HRR	0	0	0	HR4	HR3	HR2	HR1	HR0

0xFC	DAYR	0	0	0	DAY4	DAY3	DAY2	DAY1	DAY0
0xFD	MTHR	0	0	0	0	MTH3	MTH2	MTH1	MTH0
0xFE	YRR	0	YR6	YR5	YR4	YR3	YR2	YR1	YR0
0xFF	DOWR	0	0	0	0	0	DOW2	DOW1	DOW0

3.4.2 CPU52 寄存器

表 3-8 列出了 ATT7027 CPU52 寄存器的地址、名称、复位值和简单的功能描述，此外还列出了 CPU52 包含但标准 8052 中不包含的寄存器，用阴影部分标记。

表 3-8 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
0x86	DPS	0x00	数据指针选择寄存器
0x87	PCON	0x30	波特率选择寄存器
0x88	TCON	0x00	定时器控制寄存器 一见 表 11-3
0x89	TMOD	0x00	定时器方式寄存器 一见 表 11-2
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 一见 表 11-7
0x8F	SPC_FNC	0x00	DM/PM 选择寄存器 一见 表 10-3
0x91	EXIF	0x08	外部中断标志寄存器 一见 表 4-5
0x92	MPAGE	0x00	保留
0x98	SCON0	0x00	串口 0 控制寄存器 一见 表 15-8
0x99	SBUF0	0x00	串口 0 数据缓冲寄存器
0xA8	IE	0x00	中断使能寄存器 一见 表 4-3
0xB8	IP	0x80	中断优先级寄存器 一见 表 4-4
0xC0	SCON1	0x00	串口 1 控制寄存器 一见 表 15-9
0xC1	SBUF1	0x00	串口 1 数据缓冲寄存器
0xC8	T2CON	0x00	定时器 2 控制寄存器 一见 表 11-5
0xCA	RCAP2L	0x00	定时器 2 捕获寄存器低 8 位
0xCB	RCAP2H	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 一见 表 3-9
0xD8	EICON	0x40	扩展中断控制寄存器 一见 表 4-6
0xE0	ACC	0x00	累加器
0xE8	EIE	0xE0	外部中断使能寄存器 一见 表 4-7
0xF0	B	0x00	B 寄存器
0xF8	EIP	0xE0	外部中断优先级寄存器 一见 表 4-8

寄存器说明:

1. ACC

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

2. B

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

3. PSW

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

表 3-9 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	C _Y : 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(十进制的一个数字)向高 4 位数进位或借位时, Ac 将被置位, 否则被清零。Ac 被用于 DAA 指令的十进制调整。		
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		
	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，当对外部数据存储器空间寻址时，可作为间接寄存器用。

ATT7027 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，ATT7027 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 86H) 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

ATT7027 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. 定时器

ATT7027 有三个 16 位定时器 T0，T1 和 T2。它们各由两个独立的 8 位寄存器组成，共有 6 个独立的寄存器：TH0、TL0、TH1、TL1、TH2、TL2。可以对这六个寄存器寻址，但不能把 T0、T1、T2 当作一个 16 位寄存器用。

8. 其他 CPU52 寄存器

IP、IE、EIP、EIE、EXIF、TMOD、TCON、T2CON、RCAP2、SCON0、SCON1、CKCON 和 PCON 寄存器分别包含中断系统、定时器/计数器、串行口的控制和状态位，这些寄存器将在 [中断系统](#)、[定时器](#) 和 [UART](#) 章节中叙述。

3.4.3 扩展寄存器

除了 CPU52 内部的寄存器，ATT7027 通过 SFR 总线为系统控制和外设扩展了很多其他寄存器（[表 3-7](#) 阴影以外部分），这些寄存器统称扩展寄存器，包括：

- 存储器控制寄存器
详细说明参见 [存储器映射](#) 章节和 [FLASH](#) 章节。
- 系统复位寄存器
详细说明参见 [系统复位](#) 章节。
- 系统时钟管理寄存器
详细说明参见 [时钟](#) 章节和 [低功耗模式](#) 章节。
- 系统电源管理寄存器
详细说明参见 [电源系统](#) 章节和 [低功耗模式](#) 章节。
- 低电压检测寄存器
详细说明参见 [电源系统](#) 章节。
- 系统工作模式寄存器
详细说明参见 [低功耗模式](#) 章节。
- 电能计量寄存器
详细说明参见 [电能计量](#) 章节。
- GPIO 寄存器
详细说明参见 [GPIO](#) 章节。

- LCD 寄存器
详细说明参见 [LCD Driver](#) 章节
- RTC 寄存器
详细说明参见 [RTC](#) 章节
- I2C 寄存器
详细说明参见 [I2C](#) 章节
- PWM 寄存器
详细说明参见 [PWM](#) 章节
- TPS 寄存器
详细说明参见 [TPS](#) 章节
- 硬件断点寄存器
详细说明参见 [片上ICE支持](#) 章节
- 位写保护寄存器
位写保护模式防止软件直接修改受密码保护的寄存器位。

表 3-10 位写保护寄存器 (BWPR, 0x9AH)

Bit Write Protect Register (BWPR)			Address: 9AH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PASS4	PASS3	PASS2	PASS1	PASS0	BITPS	PMOD1	PMOD0
Write:						x		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
BWPR[7:3]	PASS[4:0]: 位写保护密码; PASS[4:0]=11000: 使能写 PMOD[1:0]操作。PASS[4:0]=10011: 开启所有受保护位的写使能。PASS[4:0]=10001: 开启 RTC 所有受保护寄存器的写使能。PASS[4:0]=10101: 关闭所有受保护位和受保护寄存器的写使能。
BWPR.2	BITPS: 位写保护状态标志; BITPS=1: 表示软件可以修改所有受保护的位; BITPS=0: 表示软件不可以修改所有受保护的位
BWPR[1:0]	PMOD[1:0]: 位写保护模式选择, PMOD[1:0]=11: 表示可以修改位写保护状态。PMOD[1:0]=00/01/10: 表示不可以修改位写保护状态

使用说明:

当 PMOD[1:0]=11 时, 写 PASS[4:0]=10011 可以使能所有受保护位的写操作, 这些受保护位允许被软件修改, 此时写使能处于开启状态; 而写 PASS[4:0]=10101 将关闭所有受保护位的写操作, 软件不可修改这些受保护的位, 此时写使能处于关闭状态。

写使能开启后, 如果没有写 PASS[4:0]将写使能关闭, 那么在写使能开启后的 128 个 fsys 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 128 个 fsys。

BITPS 表示当前写使能状态, 如果写使能被开启, 则 BITPS=1, 否则 BITPS=0。

PMOD[1:0] 不可以直接被修改, 也必须通过 PASS[4:0] 进行控制。只有当 PASS[4:0]=11000 时, 才能够修改 PMOD[1:0]的值, 否则写无效。

受保护的寄存器 ([表 3-7 红色字体部分](#)) 有:

1. RTC 寄存器: 包括 RTCCAL、SECR、MINR、HRR、DAYR、MTHR、YRR、DOWR;
2. 系统配置寄存器: 包括 BORCFG、CLKCFG、FMCFG;
3. 电源和时钟管理寄存器 SUPDC;
4. GPIO 复用配置寄存器: 包括 LCDCFG、P02CFG、P3CFG。

4 中断系统

4.1 概述

ATT7027 中断系统支持 12 个中断，其中 7 个通用中断，包括外部引脚 int0_n 和 int1_n 中断，定时器 T0、T1、T2 中断以及串行口 UART0、UART1，保留了标准 8052 原有功能。

另外 5 个中断，利用 CPU52 的 5 个扩展中断，分别为：

- 计量中断(IRQ_EMU)
- 实时钟中断(IRQ_RTC)
- I2C 中断(IRQ_SCI)
- LVD 或 BOR 中断(IRQ_LVI)
- 在线调试中断(IRQ_OCD)

中断系统概括如下表所示。

表 4-1 ATT7027 中断系统

ATT7027 中断源	CPU52中断 源	功能描述	同级 内中 断优 先级	中断向 量	中断使能 (EA=1)	中断标志
OCD	pfi	在线调试中断，详述见片上ICE支持章节	0	0033h	EICON.5 BPEN	EICON.4 BPACK
INT0	int0_n	外部引脚int_0中断	1	0003h	IE.0	-
T0	TF0	Timer0 中断	2	000Bh	IE.1	TF0
INT1	int1_n	外部引脚int_1中断	3	0013h	IE.2	-
T1	TF1	Timer1 中断	4	001Bh	IE.3	TF1
UART0	TI_0 or RI_0	UART0发送或接收	5	0023h	IE.4	TI_0 or RI_0
T2	TF2	Timer 2 中断	6	002Bh	IE.5	TF2
UART1	TI_1 or RI_1	UART1发送或接收	7	003Bh	IE.6	TI_1 or RI_1
EMU	int2	EMU中断，详述见EMU章节	8	0043h	EIE.0 EMUIE	EXIF.4 EMUIF
RTC	int3_n	RTC中断，详述见RTC章节	9	004Bh	EIE.1 RTCIE	EXIF.5 RTCIF
保留	int4	保留	10	0053h	EIE.2	EXIF.6
SCI	int5_n	I2C中断，详述见I2C章节	11	005Bh	EIE.3 SCIE	EXIF.7 SCIIF
LVI	wdti	LVD或BOR中断，详述见LVD和BOR章节	12	0063h	EIE.4 LVIE	EICON.3 LVIF

4.2 寄存器

表 4-2 中断系统寄存器列表

地址	名称	复位值	功能描述
0xA8	IE	0x00	中断允许寄存器一见 表4-3
0xB8	IP	0x00	中断优先级寄存器一见 表4-4
0x91	EXIF	0x08	外部中断标志寄存器一见 表4-5
0xD8	EICON	0x40	扩展中断控制寄存器一见 表4-6
0xE8	EIE	0xE0	扩展中断允许寄存器一见 表4-7

0xF8	EIP	0xE0	扩展中断优先级寄存器—见 表4-8
0xA9	EMUIE	0x00	EMU中断使能寄存器—见 表9-9
0xB1	EMUIF	0x00	EMU中断标志寄存器—见 表9-10
0xAA	RTCIE	0x00	RTC中断使能寄存器—见 表12-5
0xB2	RTCIF	0x00	RTC中断标志寄存器—见 表12-6
0xAB	SCIIE	0x00	SCI中断使能寄存器—见 表16-3
0xB3	SCIIF	0x08	SCI中断标志寄存器—见 表16-4
0xAC	LVIIE	0x00	LVI中断使能寄存器—见 表7-10
0xB4	LVIIF	0x00	LVI中断标志寄存器—见 表7-11
0xE3	HWBPCR	0x00	硬件断点控制寄存器—见 表22-4

1. 中断允许寄存器

表 4-3 Interrupt Enable Register (IE 0xA8H)

Interrupt Enable Register (IE)			Address: A8H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
IE.7	EA: CPU 中断允许标志。EA=0, CPU 禁止所有中断, 即 CPU 屏蔽所有的中断请求; EA=1, CPU 开放中断。但每个中断源的中断请求是允许还是被禁止, 还需由各自的允许位确定。
IE 6	ES1: 串行口 1 中断允许位。ES1=1, 允许串行口 1 中断; ES1=0, 禁止串行口 1 中断。
IE.5	ET2: 定时器 2 的溢出中断允许位。ET2=1, 允许 Timer2 中断; ET2=0, 禁止 Timer2 中断。
IE.4	ES0: 串行口 0 中断允许位。ES1=1, 允许串行口 0 中断; ES1=0, 禁止串行口 0 中断。
IE.3	ET1: 定时器 1 的溢出中断允许位。ET1=1, 允许 Timer1 中断; ET2=0, 禁止 Timer1 中断。
IE.2	EX1: 外部中断 1 中断允许位。EX1=1, 允许外部中断 1 中断; EX1=0, 禁止外部中断 1 中断。
IE.1	ET0: 定时器 0 的溢出中断允许位。ET0=1, 允许 Timer0 中断; ET0=0, 禁止 Timer0 中断。
IE.0	EX0: 外部中断 0 中断允许位。EX0=1, 允许外部中断 0 中断; EX0=0, 禁止外部中断 0 中断。

2. 中断优先级寄存器

表 4-4 Interrupt Priority Register (IP 0xB8H)

Interrupt Priority Register (IP)			Address: B8H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	1	PS1	PT2	PS0	PT1	PX1	PT0	PX0
Write:	X							
Reset:	0	0	0	0	0	0	0	0

位	功能描述
IP.7	保留。读为 1
IP.6	PS1: 串行口 1 中断优先级控制位。PS1=1, 串行口 1 定义为高优先级中断源; PS1=0, 串行口 1 定义为低优先级中断源。

IP.5	PT2: Timer2 中断优先级控制位。PT2=1, Timer2 定义为高优先级中断源; PT2=0, Timer2 定义为低优先级中断源。
IP.4	PS0: 串行口 0 中断优先级控制位。PS0=1, 串行口 0 定义为高优先级中断源; PS0=0, 串行口 0 定义为低优先级中断源。
IP.3	PT1: Timer1 中断优先级控制位。PT1=1, Timer1 定义为高优先级中断源; PT2=0, Timer1 定义为低优先级中断源。
IP.2	PX1: 外部中断 1 中断优先级控制位。PX1=1, 外部中断 1 定义为高优先级中断源; PX1=0, 外部中断 1 定义为低优先级中断源。
IP.1	PT0: Timer0 中断优先级控制位。PT0=1, Timer0 定义为高优先级中断源; PT0=0, Timer0 定义为低优先级中断源。
IP.0	PX1: 外部中断 0 中断优先级控制位。PX0=1, 外部中断 0 定义为高优先级中断源; PX0=0, 外部中断 0 定义为低优先级中断源。

3. 外部中断标志寄存器

表 4-5 External Interrupt Flag Register (EXIF 0x91H)

External Interrupt Flag Register (EXIF)			Address: 91H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	IF5	IF4	IF3	IF2	0	0	0	0
Write:					X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
EXIF.7	IF5: CPU52 外部中断 5 标志位。IF5=1 说明 CPU52 检测到 CPU52 int5_n 输入有一个下降沿跳变, 此跳变是由 I2C 中断产生的。在退出 I2C 中断服务程序前, 除了要软件清 SCIIF 外, IF5 也必须由软件清除。若将 IF5 软件置 1 将产生 I2C 中断。
EXIF.6	IF4: CPU52 外部中断 4 标志位。IF4=1 说明 CPU52 检测到 CPU52 int4 输入有一个上升沿跳变。该功能保留。
EXIF.5	IF3: IF3=1 说明 CPU52 检测到 CPU52 int3_n 有一个下降沿跳变, 此跳变是由 RTC 中断产生的。在退出 RTC 中断服务程序前, 除了要软件清 RTCIF 外, IF3 也必须由软件清除。若将 IF3 软件置 1 将产生 RTC 中断。
EXIF.4	IF2: IF2=1 说明 CPU52 检测到 CPU52 int2n 有一个上升沿跳变, 此跳变是由 EMU 中断产生的。在退出 EMU 中断服务程序前, 除了要软件清 EMUIF 外, IF2 也必须由软件清除。若将 IF2 软件置 1 将产生 EMU 中断。
EXIF.3	保留。读为 1
EXIF.2-0	保留。读为 0

4. 扩展中断控制寄存器

表 4-6 扩展中断控制寄存器(EICON 0xD8H)

EICON			Address: D8H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	SMOD1	1	EPFI	PFI	WDTI	0	0	0
Write:		X				X	X	X
Reset:	0	1	0	0	0	0	0	0

位	功能描述
EICON.7	SMOD1, 详述见 UART 章节。
EICON.6	保留。读为 1。
EICON.5	EPFI: CPU52 Power-fail 中断允许位。和 BPEN (HWBPCR.0) 断点使能位结合使用。当 EPFI=1 且 BPEN=1 时, 允许在线调试中断 OCD; 当 EPFI=0

	或 BPEN=0 时，禁止在线调试中断。
EICON.4	PFI: CPU52 Power-fail 中断标志位。PFI=1 说明 CPU52 检测到 CPU52 pfi 输入有一个上升沿跳变，此跳变是由 OCD 中断产生的。在退出 OCD 中断服务程序前，除了要软件清 BPACK 外，PFI 也必须由软件清除。若将 PFI 软件置 1 将产生 OCD 中断。
EICON.3	WDTI: CPU52 WDT 中断标志位。WDTI=1 说明 CPU52 检测到 CPU52 WDT 输入有一个上升沿跳变，此跳变是由 LVI 中断产生的。在退出 LVI 中断服务程序前，除了要软件清 LVIIF 外，WDTI 也必须由软件清除。若将 WDTI 软件置 1 将产生 LVI 中断。
EICON.2-0	保留。读为 0。

5. 扩展中断使能寄存器

表 4-7 扩展中断允许寄存器(EIE 0xE8H)

Extend Interrupt Enable Register (EIE)			Address: E8H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	1	1	1	EWDI	EX5	EX4	EX3	EX2
Write:	X	X	X					
Reset:	1	1	1	0	0	0	0	0

位	功能描述
EIE.7-5	保留，读为 1。
EIE.4	EWDI: CPU52 WDT 中断允许位。EWDI=0，禁止 LVI 中断；EWDI=1 且 LVIE 寄存器 BORIE=1 或 LVDIE=1 且 BORRST=0，使能 BOR 中断或 LVD 中断。
EIE.3	EX5: CPU52 int5_n 中断允许位。EX5=0，禁止 I2C 中断；EX5=1 且 SCIE 寄存器 I2CIE=1，使能 I2C 中断。
EIE.2	EX4: CPU52 int4 中断允许位。该功能保留。
EIE.1	EX3: CPU52 int3_n 中断允许位。EX3=0，禁止 RTC 中断；EX3=1 且 RTCIE 寄存器相应位置 1，使能 RTC 相应中断。
EIE.0	EX2: CPU52 int2 中断允许位。EX2=0，禁止 EMU 中断；EX2=1 且 EMUIE 相应位置 1，使能 EMU 相应中断。

6. 扩展中断优先级寄存器

表 4-8 扩展中断优先级寄存器(EIP 0xF8H)

Extend Interrupt Register(EIP)			Address: F8H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	1	1	1	PWDI	PX5	PX4	PX3	PX2
Write:	X	X	X					
Reset:	1	1	1	0	0	0	0	0

位	功能描述
EIP.7-5	保留，读为 1。
EIP.4	PWDI: CPU52 WDT 中断优先级控制位。PWDI=0，将 LVI 中断置为低优先级；PWDI=1，将 LVI 中断置为高优先级。
EIP.3	PX5: CPU52 int5_n 中断优先级控制位。PX5=0，将 I2C 中断置为低优先级；PX5=1，将 I2C 中断置为高优先级。
EIP.2	PX4: CPU52 int4 中断优先级控制位。PX4=0，将 int4 中断置为低优先级；PX4=1，将 int4 中断置为高优先级。该功能保留。
EIP.1	PX3: CPU52 int3_n 中断优先级控制位。PX3=0，将 RTC 中断置为低优先级；PX3=1，将 RTC 中断置为高优先级。

EIP.0	PX2: CPU52 int2 中断优先级控制位。PX2=0, 将 EMU 中断置为低优先级; PX2=1, 将 EMU 中断置为高优先级。
-------	--

7. 其他中断寄存器

其他中断寄存器将在后续相关章节叙述。

4.3 中断处理

CPU 在每个机器周期的 C4 时刻采样中断标志, 而在下一个机器周期对采样到的中断进行查询。如果在前一个机器周期的 C4 有中断标志, 则在查询周期内便会查询到并按优先级高低进行中断处理, 中断系统将控制系统转入相应的中断服务程序 (ISR)。下列三个条件中的任何一个都能封锁 CPU 对中断的响应。

1. CPU 正在处理同级或高一级的中断;
2. 现行的机器周期不是当前所执行指令的最后一个机器周期;
3. 当前正在执行的指令时返回 (RETI) 指令或是对 IE 或 IP 寄存器进行读/写的指令。

上述三个条件中, 第二条是保证把当前指令执行完, 第三条是保证如果在当前执行的是 RETI 指令或是对 IE、IP 进行访问的指令时, 必须至少再执行完一条指令之后才会响应中断。

中断查询在每个机器周期中重复执行, 所查询到的状态为前一个机器周期的 C4 时采样到的中断标志。这里要注意的是: 如果中断标志被置位, 但因上述条件之一的原因而未被响应, 上述封锁条件已撤销, 而中断标志位已不再存在 (已不再是置位状态) 时, 被拖延的中断就不再被响应, CPU 将丢弃中断查询的结果。

CPU 响应中断时, 先置响应的优先级激活触发器, 封锁同级和低级的中断。然后根据中断源的类别, 在硬件的控制下, 程序转向相应的向量入口单元, 执行中断服务程序。

当一个允许的中断发生时, 硬件调用中断服务程序时, 把程序计数器 PC 的内容压入堆栈 (但不能自动保存程序状态字 PSW 的内容), 同时把被响应的中断服务程序的入口地址装入 PC 中。13 个中断源服务程序的入口地址在 [表 4-1](#) 中列出。通常在中断入口程序处安排一条跳转指令, 以跳转到用户的服务程序入口。

如果没有更高优先级的中断发生, CPU 将执行完中断服务程序。中断服务程序的最后一条指令必须是中断返回指令 RETI。CPU 执行完这条指令后, 把响应中断时所置位的优先级激活触发器清 0, 然后从堆栈中弹出两个字节的內容 (断点地址) 装入程序计数器 PC 中, CPU 就从原来被中断处重新执行被中断的程序。

4.4 中断屏蔽

除了在线调试中断 (OCD), IE SFR 的 EA (IE.7) 是全局中断允许控制位。当 EA=1 时, 其他 12 个中断的每一个中断使能/屏蔽都由各自的中断使能位控制; 当 EA=0 时, 这 12 个中断被屏蔽。OCD 是唯一一个不受 EA 位控制的中断类型。不管 EA 的当前状态是 0 还是 1, 当 EPFI=1 且 BPEN=1 时, OCD 被使能。

[表 4-1](#) 列出了 ATT7027 中断系统的每一个中断的中断源、中断标志、中断允许和同级中断优先级。

4.5 中断优先级

ATT7027 中断系统具有三级优先级, OCD 具有最高的中断优先级, 其余 12 个中断由 IP 和 EIP 寄存器把各中断源的优先级分为高优先级和低优先级, 它们遵循下列两条基本规则:

1. OCD 不可被其他中断打断，其他 12 个中断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任何中断源所中断（除 OCD）；

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 [表 4-1](#) 中同级中断优先级列。

值得指出的是，RTC 中断、EMU 中断、SCI 中断、LVI 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 中清除中断 8 个中断标志。

4.6 中断响应时间

中断响应时间由 ATT7027 运行状态决定。最快的响应时间是 5 个机器周期：检测中断占用 1 个机器周期，执行到 ISR 的 LCALL 指令占用 4 个机器周期。当 ATT7027 执行 RETI 指令且中断返回后的程序跟着 MUL 或 DIV 指令时，中断响应时间最长，为 13 个机器周期，这 13 个机器周期为：检测中断占用 1 个机器周期，执行 RETI 占用 3 个机器周期，执行 MUL 或 DIV 占用 5 个机器周期，执行到 ISR 的 LCALL 指令占用 4 个机器周期。在这种情况下，中断响应时间是 $13 \times 4 = 52$ 个系统时钟周期（fsys）。

5 时钟

5.1 概述

ATT7027的时钟系统有如下特点：

- 内置三类时钟振荡电路；
- 低功耗设计，系统时钟可灵活配置
- 内建系统时钟切换保护机制
- 针对电表应用的时钟方案

5.2 功能描述

5.2.1 时钟振荡电路

ATT7027包含三类振荡电路：

- 内部RC振荡电路
- 32.768KHz低频晶体振荡电路
- 外部高频晶体振荡电路

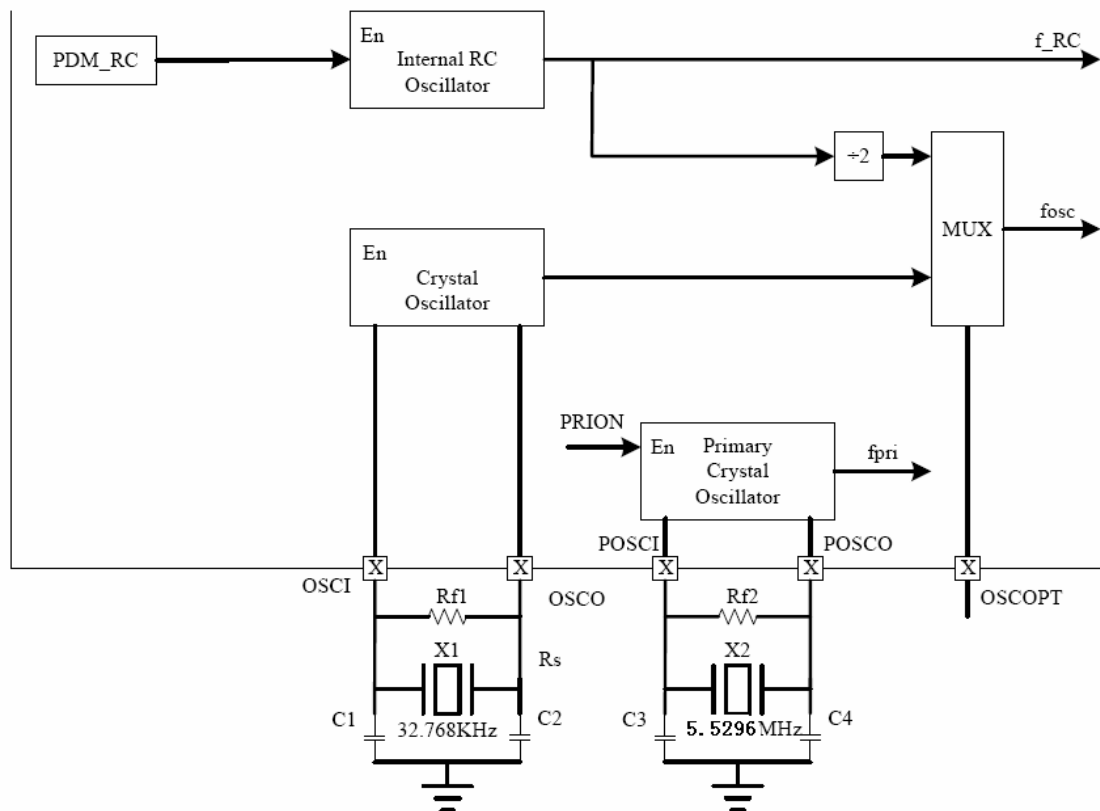


图5-1 时钟振荡电路框图

外部低频晶体振荡电路是为外部32.768KHz的晶体而设计的，如图5-1所示，OSC1是晶体振荡电路的输入引脚，OSC0是晶体振荡电路的输出引脚，典型应用电路下，X1推荐使用32.768KHz高精度晶体，C1、C2选择15PF±10% 的瓷片电容，Rf1选择10M精密电阻。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。

外部高频晶体振荡电路是为外部晶体11.0592MHz/5.5296MHz/2.7648MHz/1.3824MHz而设计的，如图5-1所示，POSCI引脚是晶体振荡电路的输入端，POSCO引脚是晶体振荡电路的输出端，典型应用电路下，X2推荐使用5.5296MHz晶体，C3、C4选择15PF $\pm$ 10% 的瓷片电容，Rf2选择10M精密电阻。外部高频晶体振荡电路是系统正常工作模式的时钟源。

内部RC振荡器主要作为片上WDT的时钟源，也可作为系统时钟时钟源fosc的输入。如图5-1所示，作为系统时钟源时，采用f_{RC}的二分频输出。通过引脚OSCOPT进行选择，默认OSCOPT内部上拉30K电阻到VCC，表示fosc直接源自外部32.768KHz的晶体振荡输入；当OSCOPT引脚为低电平，选择fosc源自f_{RC}，但是此时fosc为f_{RC}的二分频，其他操作完全等同外部晶振输入。

推荐当用户使用ATT7027 RTC功能时，为保证计时准确，使用外部晶体振荡作为fosc的时钟源；当用户不使用ATT7027 RTC功能时，可使用内部RC振荡作为fosc时钟源，可省掉外部晶体振荡电路。

5.2.2 系统时钟生成

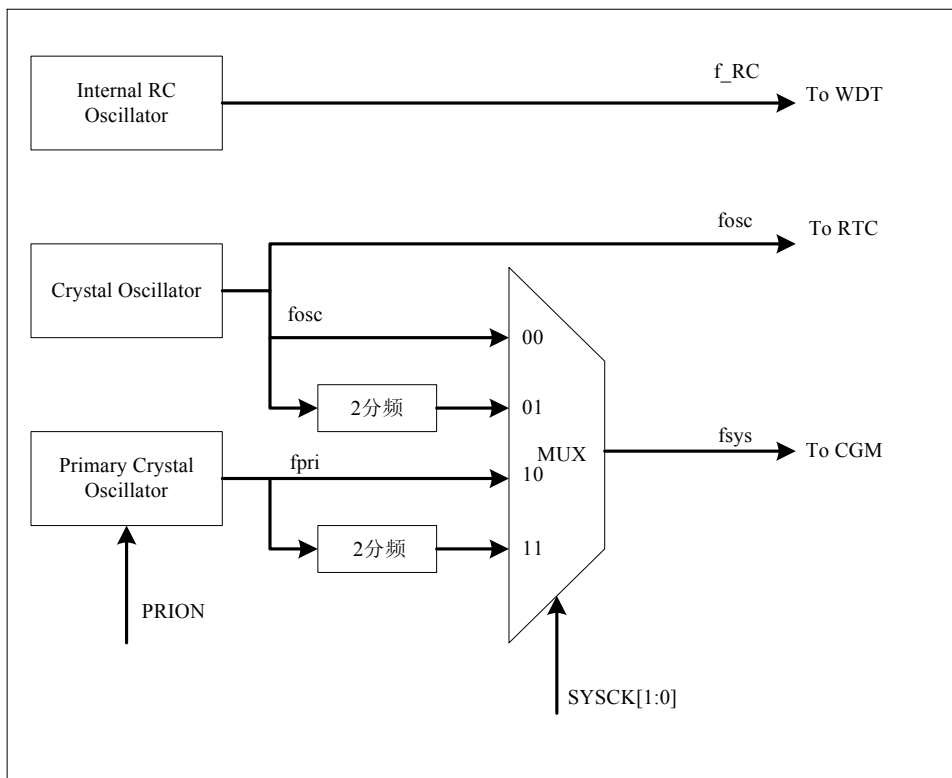


图 5-2 系统时钟生成的原理框图

图5-2是系统时钟生成的原理框图。如图所示，系统存在两个晶体振荡电路：低频振荡的输出频率为fosc和高频振荡的输出频率fpri。相应地，系统时钟fsys可以配置为fosc、fosc/2、fpri、fpri/2。

上电复位后，系统处于PSM，片上低频振荡电路开始工作，系统时钟来自片上低频晶振电路fosc，用户可通过配置SYSCK[1:0]使系统时钟为fosc、fosc/2；当SYSCK[1:0]=00时，fsys=fosc；当SYSCK[1:0]=01时，fsys=fosc/2。

5.2.2.1 低频系统时钟向高频系统时钟切换

复位后高频振荡电路默认关闭，PRION(表5-5)清零，而SYSCK1(表5-3)清为0，表示选择

fosc时钟作为系统时钟。为使计量模块工作在正常频率下，系统时钟需由低频时钟向高频时钟切换。低频系统时钟向高频系统时钟切换的过程如下：

1. 用户需根据外部高频晶振频率设置PRIP1和PRIP0寄存器，高频晶振输入频率主要有四个：11.0592MHz、5.5296MHz、2.7648MHz、1.3824MHz。相应地，PRIP1和PRIP0寄存器可按照表5-4设置。
2. 配置PRION=1，打开高频振荡电路。
3. 检测LOCK位(表5-6)是否为1，LOCK=1表示高频晶振已经稳定；LOCK=0表示高频晶振还在启动阶段或者没被开启。
4. 配置SYSCK[1:0](表5-3)使系统时钟为fpri或者fpri/2。当SYSCK[1:0]=10时，fsys=fpri；当SYSCK[1:0]=11时，fsys=fpri/2，时钟切换完成。

下面举例说明低频系统时钟向高频系统时钟切换的过程：

```
MOV    9AH, #0C3H
MOV    9AH, #98H          ; Disable Write protect
MOV    CLKCFG, #00H       ; PRIIP[1:0]=00
MOV    0BCH, #91H        ; PRION=1
LOCK:  MOV    A, 0B6H
JNB    ACC.7, LOCK        ; Judge if LOCK=1
MOV    9AH, #0C3H        ; Disable Write protect
MOV    9AH, #98H
MOV    0BDH, #02H        ; sysck[1:0]=10b, fsys switch to fpri
```

高频晶振输入频率主要有四个：11.0592MHz、5.5296MHz、2.7648MHz、1.3824MHz。那么可选用源自高频晶振的系统时钟fsys将有5个频率：11.0592MHz、5.5296MHz、2.7648MHz、1.3824MHz以及0.6912MHz。

5.2.2.2 内建保护功能

为保证时钟切换时系统运行稳定，芯片内建时钟切换保护机制。用户在进行时钟切换时需遵循以下规则：

1. 当PRION(表5-5)被设为1时，不能改变PRIP[1:0](表5-4)的设置。
2. 当SYSCK1(表5-3)=1，表示选择了fpri或者fpri/2作为系统时钟fsys，此时不能关闭高频振荡电路，也就是说不能将PRION清为0。
3. 当PRION清为0时，不能将SYSCK1设置为1，也就是在高频振荡电路关闭的情况下不能选择fpri或者fpri/2作为系统时钟。

5.2.2.3 高频系统时钟向低频系统时钟切换

当芯片由电池供电时，系统时钟需由高频时钟向低频时钟切换。高频系统时钟向低频系统时钟切换的过程如下：

1. 配置SYSCK[1:0]使系统时钟可以为fosc或者fosc/2。当SYSCK[1:0]=00时，fsys=fosc；当SYSCK[1:0]=01时，fsys=fosc/2；
2. 配置PRION=0，关闭高频振荡电路，系统时钟切换完成。

整个时钟切换的过程如下面程序所示：

```
MOV    9AH, #0C3H
MOV    9AH, #98H          ; Disable Write protect
MOV    0BDH, #0          ; sysck[1:0]=00b, fsys switch to fosc
MOV    9AH, #0C3H
MOV    9AH, #98H
```

MOV 0BCH, #01H ; prion=0

5.2.3 针对电表应用的时钟方案

1. RTC 的时钟

为保证RTC计时准确，RTC的时钟直接来自外部32.768KHz的低频晶体振荡电路，除非断电，RTC永远不会被关闭，模式切换（Nomal Mode、PSM、PDM之间的相互切换）和时钟切换对RTC时钟不产生影响；除此之外，RTC还能通过配置晶体补偿寄存器对低频晶体振荡电路进行数字补偿，消除由于温度漂移和电源纹波对低频晶体振荡电路产生的影响。所有的输出寄存器（时间&万年历）、晶体补偿寄存器不会被任何复位打扰，用以保持RTC的准确性。

2. WDT 的时钟

为保证系统可靠工作，芯片为WDT提供独立RC振荡时钟源。RC振荡器在CPU运行的节电模式和正常模式下始终工作，在PDM下，内部RC振荡器受PDM_RC位(表5-7)控制；当PDM_RC=1时，内部RC振荡在PDM下打开，WDT作为PDM唤醒源正常工作；当PDM_RC=0时，内部RC振荡在PDM下关闭，WDT在PDM下关闭。

3. EMU 部分的时钟 femu 和 fadc

EMU 部分的时钟来自系统时钟，可根据外接高频晶体频率自动调节。

1) 当系统时钟fsys源自外部低频晶体振荡频率时，即fsys=fosc或者fsys=fosc/2时，femu=fsys。在此频率下，EMU工作不正常。

2) 当系统时钟源自高频振荡电路输出时，即fsys=fpri或者fsys=fpri/2时，femu和fadc频率会根据高频振荡电路选项PRIP[1:0]进行自动调节。如下表所示：

表5-1 EMU时钟配置

PRIP[1:0]	SYSCK0	fpri	fsys	N	femu=fsys/N	fadc=femu/6
0	0	0	11.0592MHz	2	5.5296MHz	921.6 KHz
		1	11.0592MHz	1	5.5296MHz	921.6 KHz
0	1	0	5.5296MHz	1	5.5296MHz	921.6 KHz
		1	5.5296MHz	1	2.7648MHz	460.8 KHz
1	0	0	2.7648MHz	1	2.7648MHz	460.8 KHz
		1	2.7648MHz	1	1.3824MHz	230.4 KHz
1	1	0	1.3824MHz	1	1.3824MHz	230.4 KHz
		1	1.3824MHz	1	0.6912MHz	115.2 KHz

5.3 寄存器

表 5-2 系统时钟管理寄存器列表

地址	名称	复位值	功能描述
0xBD	CLKCFG	0x00	时钟配置寄存器一见表5-3，写保护
0xBC BORCFG.7	PRION	0	PRION控制位一见表5-5，保护
0xB6 SYSSCR.7	LOCK	0	LOCK状态位一见表5-6
0xBC BORCFG.1	PDM_RC	0	PDM_RC控制位一见表5-7，写保护

1. 时钟配置寄存器（写保护）

表 5-3 时钟配置寄存器（0xBDH, CLKCFG）

CLKCFG			Address: BDH					
Read:	Bit7	6	5	4	3	2	1	Bit0
Write:	WDS2	WDS1	WDS0	WDTEN	PRIP1	PRIP0	SYSCK1	SYSCK0

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

说明：PRIP[1:0]用于设置高频晶振的输入频率。

表5-4 PRIP[1:0]和fpri对应表

fpri	PRIP1	PRIP0
11.0592MHz	0	0
5.5296MHz	0	1
2.7648MHz	1	0
1.3824MHz	1	1

SYSCK[1:0]用于选择系统时钟。

SYSCK1	SYSCK0	fsys
0	0	fsys=fosc
0	1	fsys=fosc/2
1	0	fsys=fpri
1	1	fsys=fpri/2

2. PRION 控制位（写保护）

表 5-5 PRION 控制位（0xBCH, BORCFG.7）

BORCFG			Address: BCH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PRION	WKRST	BORPDM	BORRST	BORVS1	BORVS0	PDM_RC	1
Write:								X
Reset:	0	0	0	1	0	0	0	1

说明：PRION=1：开启高频晶振电路功能；PRION=0：关闭高频晶振电路功能。

3. LOCK 控制位

表 5-6 LOCK 控制位（0xB6H, SYSSCR.7）

SYSSCR			Address: B6H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	LOCK	BOROUT	LVDOUT	0	0	0	0	0
Write:	x	x	x	X	WDTCLR3	WDTCLR2	WDTCLR1	WDTCLR0
Reset:	0	0	0	0	0	0	0	0

说明：LOCK=1表示高频晶振已经稳定；LOCK=0表示高频晶振还在启动阶段或者没被开

4. PDM_RC 控制位（写保护）

表 5-7 PDM_RC 控制位

BORCFG			Address: BCH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PRION	WKRST	BORPDM	BORRST	BORVS1	BORVS0	PDM_RC	X
Write:								
Reset:	0	0	0	1	0	0	0	0

当 PDM_RC=1 时，PDM 模式下内部振荡电路开启；当 PDM_RC=0，PDM 模式下内部振荡电路关闭。

6 系统复位

6.1 概述

ATT7027支持5种复位方式：

- 上电复位POR(Power-On Reset)
- 外部引脚/RST复位
- Watchdog复位WDTR
- 掉电复位BOR(BrownOut Reset)
- PDM唤醒复位WKR

本节对POR、外部引脚/RST复位和WDTR进行说明。

任何复位源产生复位时，CPU52的程序指针恢复到0000H，所有芯片寄存器恢复到复位默认值，外部引脚电平恢复到复位状态。

6.2 功能描述

6.2.1 上电复位 POR

当电源加到芯片上时，将会发生上电复位。上电复位发生时，系统设置复位标志寄存器RSTSR标志位POR位为1，同时将其他复位标志位清零。读这个寄存器可以将这个寄存器包括POR标志清零(读后清零)。

6.2.2 外部引脚复位/RST

外部复位引脚/RST出现低电平时，将引起系统复位，复位状态寄存器的复位标志位RST被设置为1，读这个寄存器可以将这个寄存器包括RST标志清零(读后清零)。

外部/RST PIN内部设计有30K上拉电阻，上电后/RST引脚处于高电平，为保证系统可靠运行，用户在设计外围电路时，可在外部/RST引脚设计上拉电阻。

6.2.3 Watchdog 复位 WDTR

Watchdog Timer是一个特殊的定时器，按RC振荡器输出频率 f_{RC} 计时，计满预定时间则发出溢出脉冲，产生Watchdog复位。如果在溢出脉冲发生前将Watchdog Timer清零，则不会发出Watchdog复位。

ATT7027在提高WDT可靠性方面有如下特点：

- 采用硬件狗设计
- 内置RC振荡器为WDT提供独立时钟
- PDM模式下WDT开启/关闭可选

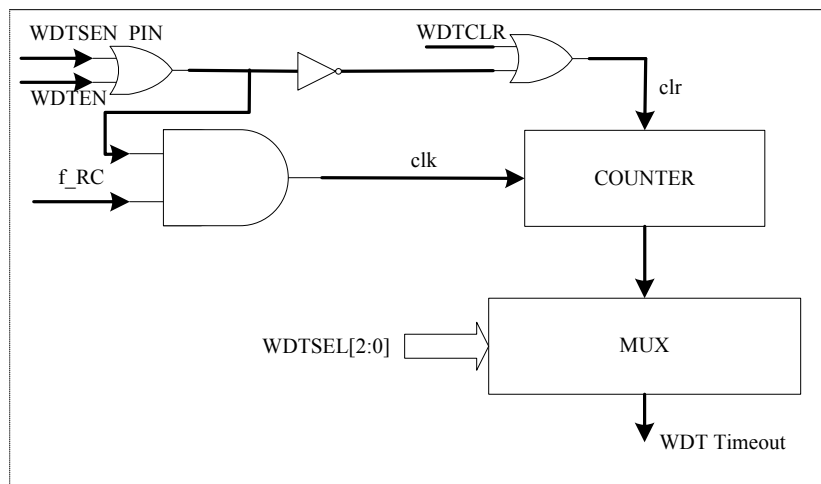


图6-1 WDT电路框图

图6-1是WDT电路框图。如图所示，WDT采用具有较高的可靠性硬件狗设计方法。WDT使能是通过SYSCR寄存器WDTEN位（[表6-4](#)）和外部WDTSEN PIN状态共同决定的。当WDTSEN PIN拉高时，无论WDTEN等于0还是等于1，WDT均被使能；当WDTSEN PIN拉低时，WDTEN=1，使能WDT，WDTEN=0，表示关闭WDT计数，并且将WDT Counter进行清零。复位后，WDTEN位默认状态为0，WDTSEN PIN的默认状态为高。通常情况下，用户在调试程序时，将WDTSEN PIN拉低，在批量生产时，将WDTSEN拉高。

通过设置WDTSEL[2:0]（[表6-3](#)）可选择WDT定时时间，定时时间分为8种可选：31.25ms、62.5ms、125ms、250ms、500ms、1.0s、2.0s、4.0s。WatchDog Timer溢出时将会导致系统复位，复位状态寄存器RSTSR（[表6-2](#)）的WDT复位标志位WDT被设置为1，读这个寄存器可以将这个寄存器包括WDT标志清零(读后清零)。

往WDTCLR[3:0]（[表6-5](#)）写入数据0110，可以将WDT清零。写其他数据无效。

通过在进入PDM模式前设置PDM_RC位（[表5-7](#)），PDM模式下，WDT的输入时钟RC振荡器可以选择被关闭（PDM_RC=0），也可以选择继续计数（PDM_RC=1）；

PDM下在WDT被关闭情况下（PDM_RC=0），WDT被清零，这样使得CPU52从PDM被唤醒后WDT从0开始计时。PDM下在WDT打开情况下，WDT继续计数，直到溢出产生WDT复位将CPU52从PDM中唤醒复位。

6.3 寄存器

表6-1 复位寄存器列表

地址	名称	复位值	功能描述
0xB7	RSTSR	0x80	复位标志寄存器—见 表6-2
0xBD CLKCFG.7-5	WDTSEL[2:0]	000	WDT溢出时间控制位—见 表6-3 ，写保护
0xBD CLKCFG.4	WDTEN	0	WDT使能位—见 表6-4 ，写保护
0xB6 SYSSCR.3-0	WDTCLR[3:0]	0000	WDT清零控制位—见 表6-5

1. 复位标志寄存器

表 6-2 复位标志寄存器(0xB7H, RSTSR)

RSTSR	Address: B7H
-------	--------------

	Bit7	6	5	4	3	2	1	Bit0
Read:	POR	RST	WDT	BOR	WKR	0	0	0
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(读后清零)。
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。读后清零。
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。读后清零。
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。读后清零。
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。读后清零。
RSTSR.2-0	保留。读为0。

2. WDT 溢出时间控制位 (写保护)

表 6-3 WDT 溢出时间控制位(0xBDH, CLKCFG.7-4)

CLKCFG	Address: BDH							
	Bit7	6	5	4	3	2	1	Bit0
Read:	WDTS2	WDTS1	WDTS0	WDTEN	PRIP1	PRIP0	SYSCK1	SYSCK0
Write:								
Reset:	0	0	0	0	0	0	0	0

WDTS[2:0]用于选择WDT溢出时间:

WDTS2	WDTS1	WDTS0	溢出时间
0	0	0	500ms
0	0	1	1.00s
0	1	0	2.00s
0	1	1	4.00s
1	0	0	31.25ms
1	0	1	62.5ms
1	1	0	125ms
1	1	1	250ms

3. WDT 使能位 (写保护)

表 6-4 WDT 使能位(0xBDH, CLKCFG.4)

CLKCFG	Address: BDH							
	Bit7	6	5	4	3	2	1	Bit0
Read:	WDTS2	WDTS1	WDTS0	WDTEN	PRIP1	PRIP0	SYSCK1	SYSCK0
Write:								
Reset:	0	0	0	0	0	0	0	0

说明: 和外部WDTS2EN PIN配合使用。当WDTS2EN PIN拉高时, 无论WDTEN等于0还是等于1, WDT均被使能; 当WDTS2EN PIN拉低时, WDTEN=1, 使能WDT, WDTEN=0, 表示关闭WDT计数, 并且将WDT Counter进行清零。

4. WDT 清零控制位

表 6-5 WDT 清零控制位(0xB6H, SYSSCR.3-0)

SYSSCR			Address: B6H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	LOCK	BOROUT	LVDOUT	0	0	0	0	0
Write:	x	x	x	x	WDTCLR3	WDTCLR2	WDTCLR1	WDTCLR0
Reset:	0	0	0	0	0	0	0	0

说明：往WDTCLR[3:0]写入数据0110，可以将WDT进行清零。写其他数据无效。

7 电源系统

7.1 概述

ATT7027 电源系统有如下特点：

- 采用外部 3.3V 单电源供电方式；
- 支持掉电复位功能，增加了系统电源可靠性；
- 支持低电压检测功能，实时监测电池电压。

7.2 功能描述

7.2.1 电源系统

ATT7027 的 VCC 和 GND 引脚是芯片数字部分电源输入和数字地，AVCC 和 AGND 引脚是芯片模拟部分电源输入和模拟地。ATT7027 采用外部单电源 3.3V 供电方式。

7.2.2 BOR

为增加系统电源可靠性，ATT7027 内置 BOR (Brown out reset) 单元，支持掉电复位功能；BOR 单元有如下特点：

- BOR 支持四档掉电电压域值选择；
- BOR 复位或中断可选；
- 低功耗设计，PDM 模式下 BOR 可选。

7.2.2.1 掉电复位

BOR 电路检测电源输入 AVCC 引脚的电压，当 AVCC 电压低于 V_{bor} 时强制系统复位，复位状态寄存器的 BOR 复位标志位 BOR (表 6-2) 被设置为 1，读这个寄存器可以将这个寄存器包括 BOR 标志清零 (读后清零)。

清 PDBOR (表 7-2) 位为 0，可以使能 BOR 检测电路。复位后，PDBOR=0，BOR 打开。

置 BORRST 位 (表 7-3) 为 1，BOR 作为掉电复位使用。此时，当 BOR 检测到 AVCC 低于 V_{bor1} 电压时生成 BOR Reset 信号，同时将 BOROUT (表 7-4) 置 1。

BOR 检测电压阈值可通过 BORVS[1:0] (表 7-3) 进行设置，可以设置 BOR 检测电压。上电复位后默认在 $VCC=2.15V$ 模式。如果要工作在其他模式，可以通过软件进行设置，但是每次上电复位后都必须设置。如果 AVCC 电压低于设定模式的 V_{bor} 电压值，将会立即产生复位信号，一直到当 AVCC 电压又高于 $V_{bor}+100mV$ ，AVCC 高于 $V_{bor}+100mV$ ，将 BOROUT 置 0。当 VCC 电压下降到接近 0V 时，将再次触发上电复位 POR。

7.2.2.2 BOR 中断

当 BORRST=0 时，BOR 作为 BOR 中断使用。如果使能 BOR 中断，每次 BOROUT 变化时 (包括 AVCC 下降小于 V_{bor} 和之后 AVCC 上升大于 $V_{bor}+100mV$) 都会产生一次 BOR 中断。使能 BOR 中断参见 BORIE 控制位说明 (表 7-5)。

当 BOR 中断发生时，系统将 BOR 中断标志位 BORIF 置 1，中断系统将控制系统转入 BOR 中断服务程序 (中断向量 0x0063H)。在退出 ISR 前，用户需通过软件清 BOR 中断，清中断过程参见 BORIF 控制位说明 (表 7-6)。

7.2.2.3 BOR 低功耗设计

为降低 PDM 下的芯片功耗，进入 PDM 模式前可以通过 PDBOR 控制位 (表 7-2) 和 BORPDM 控制位 (表 7-3) 将 BOR 电路关闭。如果进入 PDM 模式前 BOR 电路被开启，则 BOR 在 PDM 模式下继续工作，如果使能产生中断或者复位，会导致 CPU52 退出 PDM 模式。如果

BOR中断和BOR复位同时出现，则不会产生BOR中断，因为复位比中断的优先级更高。

7.2.3 LVD

ATT7027 内置 LVD 电路，实时检测 LVDIN 引脚 或者 AVCC 引脚的电压。在电表设计中，一般将电池电压通过分压电阻引入 LVDIN 引脚，利用 LVD 实时监测电池电压。

LVD 有如下特点：

- 可选择 LVD 中断；
- 可选择检测 LVDIN 引脚电压或 AVCC 电压；
- 可选择 LVD 检测电压点

LVD 使用说明如下：

PDLVD控制位(表 8-2)控制LVD的开关， PDLVD =0 表示开启LVD单元；PDLVD =1 表示关闭LVD单元；复位后，PDLVD=0，LVD打开。

LVDVS[3:0] (表 7-7) 设置LVD的域值，当LVDVS[3:0]=11XX时，LVD监测的是LVDIN 引脚上的输入电压，域值电压为 1.20V；当LVDVS[3:0]=0000~1011 时，LVD监测的是AVCC 的电压，域值电压从 2.10V到 3.50V分为 12 档。表 7-8 说明LVDVS[3:0]寄存器设置和LVD 电源监测域值的关系，注意当LVD开启，同时LVDVS[3:0]=11xx，选择LVDIN作为电源检测输入时，P2.4/LVDIN被配置成LVDIN功能，不再作为I/O端口。

当LVDVS[3:0] =0000~1011 时，AVCC降至低于LVD掉电阈值或者当LVDVS[3:0]=11XX 时，LVDIN PIN上的输入电压降至低于LVD掉电阈值时LVDOUT(表 7-9)=1；当LVDVS[3:0]=0000~1011 时，AVCC升至高于LVD掉电阈值或者当LVDVS[3:0]=11XX时，LVDIN PIN上的输入电压升至高于LVD掉电阈值，LVDOUT=0。

LVDIE (表 7-10) 使能LVD中断，LVDIE和EWDI (SFR 0xE8H, EIE.4) 结合使用，当EWDI=0 或LVDIE=0，禁止LVD中断；当EWDI=1 且LVDIE=1，使能LVD中断。如果使能LVD中断，那么每次LVDOUT变化时(包括AVCC下降或者LVDIN低于域值和上升高于域值两种情况)都会产生LVD中断。

当系统产生LVD中断时，系统将LVD中断标志位LVDIF(表7-11)置1。中断系统将控制系统转入LVD中断服务程序（中断向量0x0063H）。在退出ISR前，用户需通过软件清LVD中断，清中断过程参见LVDIF控制位说明。

7.3 寄存器

表 7-1 电源管理寄存器列表

地址	名称	复位值	功能描述
0xBF SUPDC.4	PDBOR	0	PDBOR控制位—见表7-2，写保护
0xBC	BORCFG	0x11	BOR配置寄存器—见表7-3，写保护
0xB6 SYSSCR.4	BOROUT	0	BOROUT标志位—见表7-4
0xAC LVIIE.0	BORIE	0	BOR中断允许位—见表7-5
0xB4 LVIIF.0	BORIF	0	BOR中断标志位—见表7-6
0xB5	LVDCON	0x00	LVD控制寄存器—见表7-7
0xB6 SYSSCR. 5	LVDOUT	0	LVDOUT标志位—见表7-9
0xAC LVIIE.1	LVDIE	0	LVD中断允许位—见表7-10

0xB4 LVIF.1	LVDIF	0	LVD中断标志位 一见 表7-11
----------------	-------	---	-----------------------------------

1. PDBOR 控制位（写保护）

表 7-2 PDBOR 控制位(0xBFH, SUPDC.4)

System Unit Power-Down Control Register (SUPDC)			Address: BFH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	PDTPS	PDEMU	PDBOR	PDLVD	PDLCD	PDI2C	保留
Write:	x							
Reset:	0	0	1	0	0	1	1	1

说明：PDBOR是BOR模块在Normal mode 和PSM 下的使能控制位。PDBOR=0：表示在Normal mode 和PSM 下开启BOR模块；PDBOR=1：表示在Normal mode 和PSM 下关闭BOR模块。最低位复位值为1，保留，不建议用户往该位写0。

2. BOR 配置寄存器（写保护）

表 7-3 BOR 配置寄存器(0xBCH)

BORCFG			Address: BCH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PRION	WKRST	BORPDM	BORRST	BORVS1	BORVS0	PDM_RC	1
Write:								x
Reset:	0	0	0	1	0	0	0	1

位	功能描述			
BORCFG.5	BORPDM: BORPDM是BOR模块在PDM下的使能控制位。BORPDM=0：表示在PDM下关闭BOR模块；BORPDM=1：表示在PDM下开启BOR模块。下表说明PDBOR和BORPDM对BOR模块在各种模式下打开或关断状态的控制。			
	PDBOR	BORPDM	Normal mode 和PSM 下BOR模块的开关	PDM下BOR模块的开关
	1	1	关	关
	1	0	关	关
	0	1	开	开
	0	0	开	关
BORCFG.4	BORRST: BOR复位和中断选择位。当BORRST=1，表示AVCC电压低于设定模式的Vborl电压值，将会立即产生BOR复位；当BORRST=0，表示AVCC电压低于设定模式的Vborl电压值或者AVCC升高过程中高于Vborh两种情况下，将会立即产生BOR中断。			
BORCFG.3-2	BORVS[1:0]: BOR检测电压设置位。当AVCC<BOR检测电压时，产生BOR复位或者BOR中断。			
	BORVS1	BORVS0	BOR检测电压	
	0	0	2.15 v模式	
	0	1	2.35 v模式	
	1	0	2.55 v模式	
	1	1	2.75 v模式	

3. BOROUT 标志位

表 7-4 BOROUT 标志位(0xB6H, SYSSCR.4)

SYSSCR			Address: B6H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	LOCK	BOROUT	LVDOUT	0	0	0	0	0
Write:	x	x	x	X	WDTCLR3	WDTCLR2	WDTCLR1	WDTCLR0

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

说明：BOROUT=1：表示AVCC低于BOR掉电阈值；BOROUT=0：表示AVCC高于BOR掉电阈值。

4. BOR 中断允许位

表 7-5 BOR 中断允许位(0xACh, LVIIE.0)

LVI Interrupt Enable Register (LVIIE)			Address: ACh					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	0	0	0	LVDIE	BORIE
Write:	X	X	X	X	X	X		
Reset:	0	0	0	0	0	0	0	0

说明：和 EWDI (SFR 0xE8H, EIE.4) 以及 BORRST 控制位结合使用，当 BORRST=1 或 EWDI=0 或 BORIE=0，禁止 BOR 中断；当 BORRST=0 且 EWDI=1 且 L BORIE=1，使能 BOR 中断。

5. BOR 中断标志位

表 7-6 BOR 中断标志位(0xB4H, LVIIF.0)

LVI Interrupt Flag Register (LVIIF)			Address: B4H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	0	0	0	LVDIF	BORIF
Write:	X	X	X	X	X	X		
Reset:	0	0	0	0	0	0	0	0

说明：当系统产生 BOR 中断时，系统将 BOR 中断标志位置 1。清 BOR 中断和 WDTI 标志位 (0xD8H SFR EICON.3) 结合使用，往 BORIF 写 1，可以将 BORIF 清 0。除了要软件清 BORIF 外，EICON.3 也必须由软件清除。若将 EICON.3 软件置 1 将产生 LVI 中断。

6. LVD 控制寄存器

表 7-7 LVD 控制寄存器(0xB5H, LVDCON)

LVDCON			Address: B5H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	LVDVS3	LVDVS2	LVDVS1	LVDVS0	0	0	0	0
Write:					x	x	x	x
Reset:	0	0	0	0	0	0	0	0

说明：LVDVS[3:0]设置 LVD 的域值。如表 7-8 所示：当 LVDVS[3:0]=11XX 时，LVD 监测的是 LVDIN PIN 上的输入电压，域值电压为 1.20V；当 LVDVS[3:0]=0000~1011 时，LVD 监测的是 AVCC 的电压，域值电压从 2.10V 到 3.50V 分为 12 档。

表 7-8 LVD 域值电压

LVDVS3	LVDVS2	LVDVS1	LVDVS0	LVD Voltage Level
0	0	0	0	AVCC = 2.10 v
0	0	0	1	AVCC = 2.20 v
0	0	1	0	AVCC = 2.30 v
0	0	1	1	AVCC = 2.40 v
0	1	0	0	AVCC = 2.50 v
0	1	0	1	AVCC = 2.65 v
0	1	1	0	AVCC = 2.80 v
0	1	1	1	AVCC = 2.90 v
1	0	0	0	AVCC = 3.05 v
1	0	0	1	AVCC = 3.20 v
1	0	1	0	AVCC = 3.35 v
1	0	1	1	AVCC = 3.50 v
1	1	X	X	LVDIN = 1.20 v

7. LVDOUT 标志位

表 7-9 LVDOUT 标志位 (0xB6H, SYSSCR.5)

SYSSCR			Address: B6H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	LOCK	BOROUT	LVDOUT	0	0	0	0	0
Write:	x	x	x	x	WDTCLR3	WDTCLR2	WDTCLR1	WDTCLR0
Reset:	0	0	0	0	0	0	0	0

说明：当 LVDVS[3:0]=0000~1011 时，AVCC 降至低于 LVD 掉电阈值，或者当 LVDVS[3:0]=11XX 时，LVDIN PIN 上的输入电压降至低于 LVD 掉电阈值，LVDOUT=1；当 LVDVS[3:0]=0000~1011 时，AVCC 升至高于 LVD 掉电阈值或者当 LVDVS[3:0]=11XX 时，LVDIN PIN 上的输入电压升至高于 LVD 掉电阈值，LVDOUT=0。

8. LVD 中断允许位

表 7-10 LVD 中断允许位(0xACH, LVIIE.1)

LVI Interrupt Enable Register (LVIIE)			Address: ACH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	0	0	0	LVDIE	BORIE
Write:	X	X	X	X	X	X		
Reset:	0	0	0	0	0	0	0	0

说明：和 EWDI (SFR 0xE8H, EIE.4) 结合使用，当 EWDI=0 或 LVDIE=0，禁止 LVD 中断；当 EWDI=1 且 LVDIE=1，使能 LVD 中断。

9. LVD 中断标志位

表 7-11 LVD 中断标志位(0xB4H, LVIIF.1)

LVI Interrupt Flag Register (LVIIF)			Address: B4H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	0	0	0	LVDIF	BORIF
Write:	X	X	X	X	X	X		
Reset:	0	0	0	0	0	0	0	0

说明：当系统产生 LVD 中断时，系统将 LVD 中断标志位 LVDIF 置 1。清 LVD 中断和 WDTI 标志位 (0xD8H SFR EICON.3) 结合使用，往 LVDIF 写 1，可以将 LVDIF 清 0。除了要软件清 LVDIF 外，WDTI 也必须由软件清除。若将 WDTI 软件置 1 将产生 LVI 中断。

8 低功耗模式

8.1 概述

为了降低系统功耗，ATT7027 提供三种运行模式：

- 正常模式 (Normal mode)
- 节电模式(Power saving mode)
- 掉电模式(Power Down mode)

正常模式应用于市电供电方式，停电时，用户应将运行模式切换至掉电模式，并采用电池供电，此时 CPU 停止工作。在市电掉电保存数据、PDM 唤醒复位、电池供电并且在需要 CPU 运行的情况下（如停电时抄表）等场合，用户应将运行模式切换至节电模式。

ATT7027 提供了模式间灵活和无缝的切换机制，各运行模式间的切换通过软件配置实现。

芯片功耗的降低主要取决于用户软件低功耗设计是否合理，通过软件关闭那些不需要工作的外围电路的电源和输入时钟，可以降低芯片的功耗。本节从应用的角度给用户软件低功耗设计一些指南。

8.2 各工作模式的状态

8.2.1 节电模式 (Power saving mode)

PSM 下的时钟管理：

1. PSM下，系统时钟来自片上低频晶振电路，用户可通过配置SYSCK[1:0]（[表 5-3](#)）使系统时钟为fosc、fosc/2。
2. PSM下，用户可通过设置PDEMU（[表 8-2](#)）选择打开或者关闭EMU和ADC的时钟，但打开EMU和ADC时钟后，EMU和ADC时钟源自系统时钟（fosc或者fosc/2），在此时钟下，EMU工作不正常，所以基于功耗和应用两方面的考虑，不推荐用户在PSM下打开EMU和ADC模块。复位后，PDEMU=1，EMU和3路ADC时钟关闭。
3. PSM下，用户可通过设置SUPDC寄存器（[表 8-2](#)）中的PDTPS、PDLCD、PDI2C控制位，选择打开或者关闭TPS、LCD、I2C的时钟。复位后，PDLCD、PDI2C控制位均为1，LCD、I2C的时钟关闭；PDTPS控制位为0，TPS默认开启。
4. PSM下，高频振荡电路默认关闭。建议用户只有在PSM切换到Normal模式前，才通过将PRION（[表 5-5](#)）配置为1，打开高频振荡电路。

PSM 下的电源管理：

PSM下，用户可通过设置SUPDC寄存器（[表 8-2](#)）PDTPS、PDEMU、PDBOR、PDLVD、PDLCD控制位，用户可选择打开或者关闭TPS、ADC、BOR、LVD、LCD模块电源。

复位后，PDEMU、PDLCD 均为1，ADC、LCD 电源关闭；PDTPS、PDBOR、PDLVD 均为0，TPS、BOR、LVD 电源打开；基于功耗和应用两方面的考虑，建议用户在 PSM 下打开 BOR，不打开 ADC，只有在使用时才打开 TPS、LCD、LVD 的电源。

典型测试条件下，f_{sys}=32.768KHz，V_{CC}=3.3V，LCD 打开，TPS、ADC、BOR、LVD、I2C 关闭，系统功耗可降至 70-80uA。

8.2.2 正常模式 (Normal mode)

Normal mode 应用于市电供电方式，因此是系统主要的工作模式，在 Normal mode 下，系统时钟由高频时钟提供，CPU 全速运行，计量模块正常工作。

Normal mode 下的时钟管理:

1. Normal mode下, 系统时钟来自片上高频晶振电路, 用户可通过配置SYSCK[1:0] ([表 5-3](#)) 使系统时钟为fpri、fpri/2。
2. 基于电能计量应用的考虑, 用户在系统处于Normal mode时, 必须通过设置PDEMU ([表 8-2](#)) =0 选择打开EMU和ADC时钟, 使EMU和ADC在Normal mode下正常工作。
3. Normal mode下, 用户可通过设置PDTPS、PDLCD、PDI2C控制位 ([表 8-2](#)), 选择打开或者关闭TPS、LCD、I2C的时钟。

Normal mode 下的电源管理:

1. 基于电能计量应用的考虑, 用户通过时钟切换进入Normal模式后, 必须通过PDEMU ([表 8-2](#)) 控制位将EMU和第 1、3 路ADC开启, 如果有防窃电的要求, 还要通过ADC2ON控制位 ([表 9-47](#)) 将第 2 路ADC开启, 确保EMU和ADC正常工作。
3. Normal mode下, 通过设置PDTPS、PDBOR、PDLVD、PDLCD控制位 ([表 8-2](#)), 用户可选择打开或者关闭TPS、BOR、LVD、LCD的电源。

典型测试条件下, $f_{sys}=5.5\text{MHz}$, $f_{adc}=450\text{KHz}$, $V_{CC}=3.3\text{V}$, 其他模块处于默认状态, 此时系统功耗可达 5~6mA。

8.2.3 掉电模式(Power Down mode)

市电停电时, 用户应将运行模式切换至掉电模式, 并采用电池供电。

PDM 下的时钟管理:

1. PDM 下, 系统时钟被停止, CPU52 (包括 UART 和 Timer) 停止工作, 模块时钟源于系统时钟的 EMU、ADC、I2C、TPS、38k 调制等由于系统时钟停止, 也停止工作。
2. PDM 下, 低频振荡电路仍保持工作, 由于 f_{osc} 给 RTC 提供时钟, RTC 正常工作。
3. 用户可通过进入PDM前PDM_RC控制位 ([表 5-7](#)) 配置RC振荡器为打开或者关闭, WDT 根据RC振荡器开关状态可选是否工作。
4. PDM下LCD是否工作取决于进入PDM前PDLCD ([表 8-2](#)) 配置情况。

PDM 下的电源管理:

1. PDM下, CPU52、Flash、XRAM、ROM、EMU、I2C、PWM、ICE (见 [片上ICE支持](#)) 等处于关闭状态。XRAM和EMU中的数据丢失, 所以用户程序需在系统掉电进入PDM前将XRAM和EMU重要数据保存到SRAM或者外部E2PROM中; 而CPU52、EMU、I2C、PWM、ICE (见 [片上ICE支持](#)) 的SFR等待唤醒复位后重新初始化。
2. RC 振荡器电源可通过进入 PDM 之前, PDM_RC 控制位配置为打开或者关闭, WDT 根据 RC 振荡器开关状态可选是否工作。
3. PDM下, BOR可通过进入PDM之前, PDBOR和BOR_PDM ([表 7-3](#)) 控制位配置为打开或者关闭。
4. PDM 下, TPS、LVD、LCD 电源通过进入 PDM 之前, PDTPS、PDLVD 和 PDLCD 控制位配置为打开或者关闭。
5. PDM 下, OSC、SRAM、POR、RTC、WDT 等供电正常。

典型测试条件下, 在进入 PDM 前, 在用户程序关闭所有可关闭的模块的条件下, 系统仅有 OSC 和 RTC 工作, 此时系统功耗降至 2-3uA。

8.3 各工作模式转换

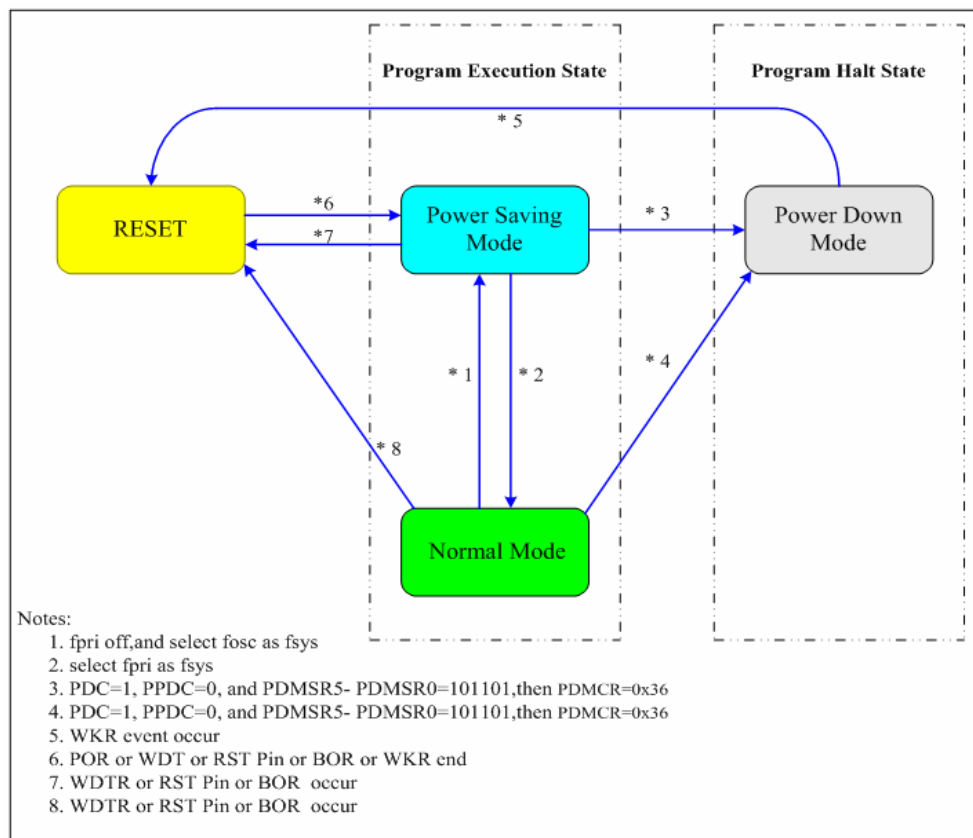


图 8-1 各模式的转换关系

8.3.1 Normal mode ->PSM

如 [图 8-2](#)所示，市电供电情况下，系统运行于Normal Mode，在这种模式下，用户程序需对市电进行实时监测，判断市电是否丢失。当检测到市电电源低于设置值时，系统电源自动切换由外部电池供电状态，用户可通过软件设置切换到PSM模式，或者通过软件设置直接切换到PDM。否则继续工作在Normal mode下。

如 [图 8-1](#)所示，Normal mode向PSM模式切换是通过系统时钟由高频时钟向低频时钟切换完成的。关于系统时钟由高频时钟向低频时钟切换的详细说明参见 [5.2.2.3 高频系统时钟向低频系统时钟切换](#) 章节。

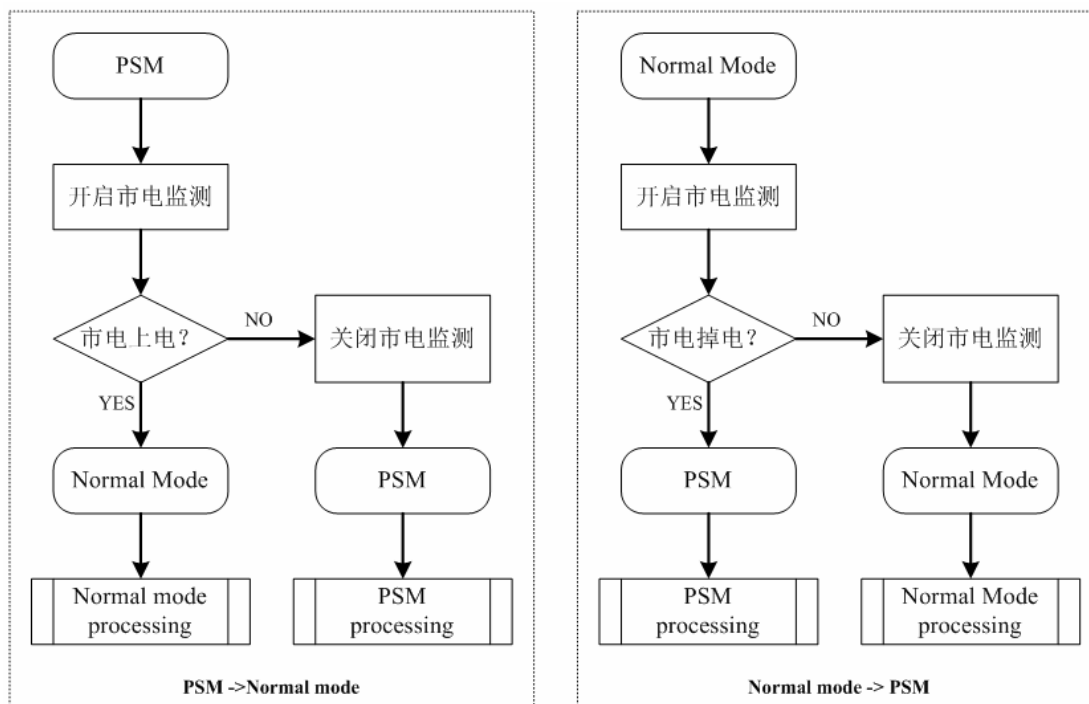


图 8-2 PSM<->Normal mode

8.3.2 PSM-> Normal mode

如 图 8-2 所示，PSM模式下，先开启市电监测电路对市电进行检测，判断市电是否上电。如果市电上电，则通过命令进入Normal模式；如果市电没有上电，则关闭市电监测电路，继续在PSM模式工作，处理相应的事件，然后退回到设定的PDM模式。

PSM向Normal模式切换是通过系统时钟由低频时钟向高频时钟切换完成的。关于系统时钟由低频时钟向高频时钟切换的详细说明参见 [5.2.2.1 低频系统时钟向高频系统时钟切换](#) 章节。

8.3.3 PSM 或 Normal mode-> PDM

如 图 8-3 所示，市电掉电时需保存现场数据，当市电掉电保存好EMU和XRAM中重要数据后，系统可由PSM切换到PDM，通过将PDMSR5- PDMSR0=101101，PDC和PPDC分别设置为 1 和 0，然后设置寄存器PDMCR=0x36，可以进入PDM模式。

当需要保存的数据量比较大时，为了缩短数据保存的时间，用户也可在高频时钟下保存数据，然后直接由 Normal mode 切换到 PDM，这个过程同 PSM 切换到 PDM 相似。

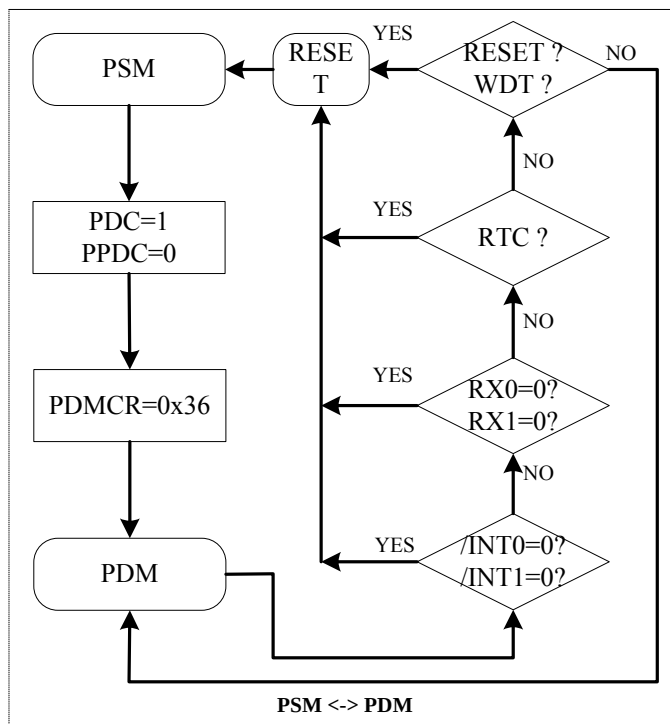


图 8-3 PSM<->PDM

8.3.4 PDM -> PSM

PDM 只能切换到 PSM，这种切换是通过唤醒复位的方式进行的，有多种唤醒源能产生 PDM 的唤醒复位，这些唤醒源包括：/INT0 PIN、/INT1 PIN、RX0 PIN、RX1 PIN、RST PIN、WDT 溢出复位、RTC 中断。

从 PDM 唤醒时，芯片将退出 PDM 模式；芯片退出 PDM 模式时，先将芯片复位，并置位 RSTSR 寄存器 WKR 标志位，CPU52 从 0000H 开始执行代码，除复位标志寄存器，其他寄存器恢复到复位状态，这个过程与 Power On Reset 相似。

用户在使用唤醒 PDM 模式应注意以下几点：

1. RX0 PIN、RX1 PIN 必须配置为 CPU52 的 /int0 或 /int1 输入才具有 PDM 唤醒复位功能（配置方法参见 [15.2.8 RX0/RX1 配置为 /INT0 和 /INT1 的输入](#) 章节），由于 /INT0 PIN、/INT1 PIN、RX0 PIN、RX1 PIN 共用了 CPU52 的 /int0 和 /int1 资源，所以这四种唤醒方式只能同时使用其中两种，这四种唤醒源的选择由 RXCON 寄存器（[表 15-13](#)）决定。
2. PDM 模式下 /INT0、/INT1、RX0 都处于高电平状态，RX1 PIN 需外加上拉电阻才能处于高电平。当 RXCON 决定的唤醒源引脚出现至少 100ns 的低电平时，芯片退出 PDM 模式。注意芯片被 /INT0 PIN、/INT1 PIN、RX0 PIN、RX1 PIN 唤醒时，CPU52 不会执行相应的中断的相应服务程序，此时，系统被复位，CPU52 从复位地址 0000H 开始执行。
3. 在使用 WDT 复位唤醒 PDM 时，注意在进入 PDM 前设置 PDM_RC=1，使 RC 振荡器在 PDM 下保持工作，WDT 才能在 PDM 下正常工作并产生溢出唤醒复位。
4. 由于 RTC 中断包含：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 个中断源。当使用 RTC 中断唤醒时，注意在进入 PDM 前，通过设置 RTCIE（[表 12-5](#)）寄存器屏蔽掉不需要的 RTC 中断源。

8.4 寄存器

表 8-1 运行模式相关寄存器列表

地址	名称	复位值	功能描述
0xBF	SUPDC	0x67	电源和时钟管理寄存器 一见表8-2, 写保护
0xC7	PDMSR	0x00	PDM状态寄存器 一见表8-3
0xCF	PDMCR	0x00	PDM控制寄存器 一见表8-4
0xB7 RSTSR.3	WKR	0	唤醒复位标志位 一见表8-5
0xBD	CLKCFG	0x00	时钟配置寄存器 一见表5-3, 写保护
0xBC BORCFG.7	PRION	0	PRION控制位 一见表5-5, 写保护
0xB6 SYSSCR.7	LOCK	0	LOCK状态位 一见表5-6
0xBC BORCFG. 5	BORPDM	0	BORPDM控制位 一见表7-3, 写保护
0xBC BORCFG. 1	PDM_RC	0	PDM_RC控制位 一见表5-7, 写保护

1. 电源和时钟管理寄存器（写保护）

表 8-2 电源和时钟管理寄存器(0xBFH, SUPDC)

System Unit Power-Down Control Register (SUPDC)			Address: BFH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	PDTPS	PDEMU	PDBOR	PDLVD	PDLCD	PDI2C	保留
Write:	X							
Reset:	0	0	1	0	0	1	1	1

位	功能描述
SUPDC.7	保留。读为0。
SUPDC.6	PDTPS: PDTPS =0表示开启温度传感器模块的时钟和电源; PDTPS =1表示关闭温度传感器模块的时钟和电源; 复位后, PDTPS=0, TPS开启。
SUPDC.5	PDEMU: PDEMU =0表示开启EMU计量单元的时钟以及3路ADC的时钟和电源; PDEMU =1表示关闭EMU计量单元的时钟以及3路ADC的时钟和电源; 复位后, PDEMU=1, EMU关闭。
SUPDC.4	PDBOR: PDBOR =0表示开启BOR单元的电源; PDBOR =1表示关闭BOR单元的电源; 复位后, PDBOR=0, BOR打开。
SUPDC.3	PDLVD: PDLVD =0表示开启LVD单元的电源; PDLVD =1表示关闭LVD单元的电源; 复位后, PDLVD=0, LVD打开。
SUPDC.2	PDLCD: PDLCD =0表示开启LCD单元的时钟和电源; PDLCD =1表示关闭LCD单元的时钟和电源; 复位后, PDLCD=1, LCD关闭。
SUPDC.1	PDI2C: PDI2C =0表示开启I2C单元的时钟; PDI2C =1表示关闭I2C单元的时钟; 复位后, PDI2C=1, I2C关闭。
SUPDC.0	保留: 保留复位值为1, 不建议用户往该位写0。

2. PDM 状态寄存器

表 8-3 PDM 状态寄存器(0xC7H, PDMSR)

Power Down Mode Status Register (PDMSR)			Address: C7H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PDMSR5	PDMSR4	PDMSR3	PDMSR2	PDMSR1	PDMSR0	PDC	PPDC
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PDMSR [7:2]	PDMSR5- PDMSR0: 当PDMSR5- PDMSR0=101101时, PDMSR寄存器为有效数据。 注意: 如果PDMSR5- PDMSR0不是有效数据, 此时写PDMCR=0x36, 不会造成系统进入掉电模式, 但是会被作为非法指令, 从而导致系统复位。
PDMSR[1:0]	PDC和PPDC: PDMSR5- PDMSR0=101101时, 通过将PDC和PPDC分别设置为1和0, 然后设置寄存器PDMCR=0x36, 可以进入PDM模式。 注意: 当PDMSR5- PDMSR0=101101时, 如果PDC和PPDC不为1和0, 设置寄存器PDMCR=0x36将导致不可预期的结果。

3. PDM 控制寄存器

表 8-4 PDM 控制寄存器(0xCFH, PDMCR)

Power Down Mode Control Register (PDMCR)			Address: CFH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PDMCR7	PDMCR6	PDMCR5	PDMCR4	PDMCR3	PDMCR2	PDMCR1	PDMCR0
Write:								
Reset:	0	0	0	0	0	0	0	0

说明: PDMSR 寄存器为有效数据时, 写 0x36 数据到 PDMCR 寄存器, 可以进入 PDM。

4. 唤醒复位标志位

表 8-5 唤醒复位标志位(0xB7H, RSTSR.3)

RSTSR			Address: B7H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	POR	RST	WDT	BOR	WKR	0	0	0
Write:	X	X	X	X	x	x	x	X
Reset:	1	0	0	0	0	0	0	0

说明: PDM 复位唤醒标志 WKR。系统处于 PDM 下可被 int0_n、int1_n、RX0、RX1、RTC、WDT、RST PIN 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。读后清零。

9 电能计量

9.1 概述

ATT7027 提供单相电能计量所需要的全部功能，包括有功功率与有功电能、无功功率与无功电能、视在功率与视在电能、电压有效值、电流有效值及频率计算等，支持灵活的防窃电方案和校表方案。

- 在动态范围 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
- 灵活的防窃电方案，窃电域值通过寄存器可调
- 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 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

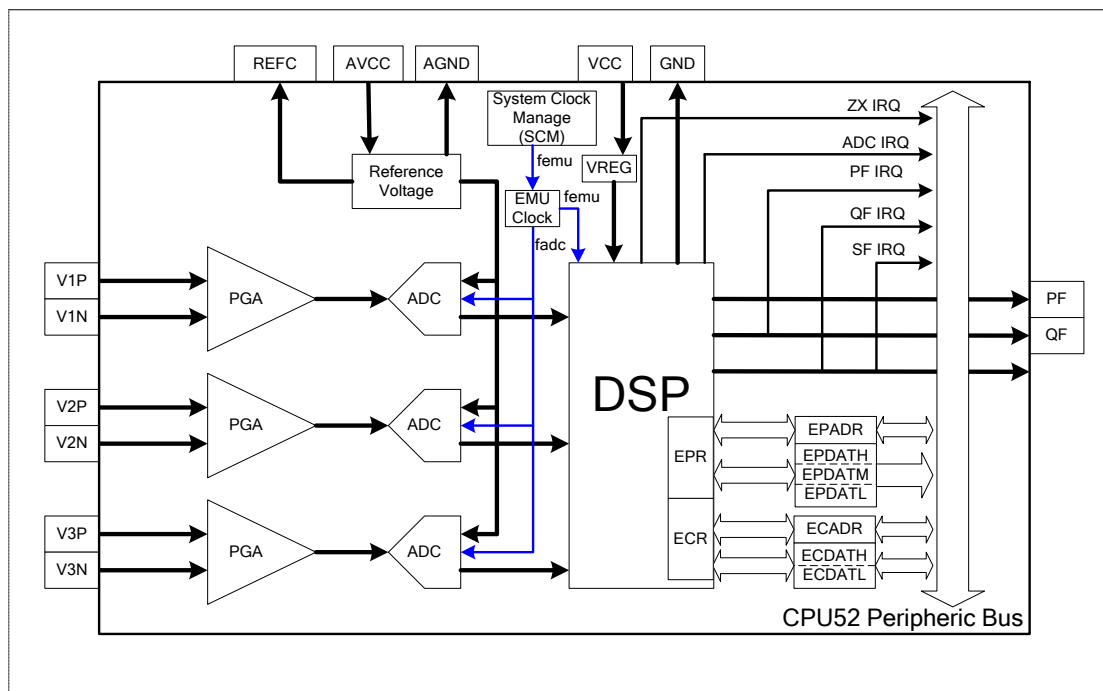


图 9-1 电能计量单元 (EMU)

9.2 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 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=5.5296MHz，那么 fadc=922KHz。

9.2.1 模数转换器

ATT7027 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。

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\(0x58H, 表 9-54\)](#)可以对三路ADC独立配置放大倍数，放大倍数分别为 1、8、16、32。

通过寄存器 [DGAIN\(0x5AH, 表 9-56\)](#)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 [ADC2ON\(0x51H.7, 表 9-47\)](#)单独选择关闭或者开启。

ADC2 同时提供增益校正寄存器 [I2GAIN\(0x4AH, 表 9-40\)](#)，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。

9.2.2 ADC 采样输出和功率波形输出

ATT7027 将三路ADC输出的 16bit波形数据 [Spl_I1\(0x00H, 表 9-13\)](#)、[Spl_I2\(0x01H, 表 9-14\)](#)、[Spl_U\(0x02H, 表 9-15\)](#)开放给用户。同时也将作为电能累加的功率波形数据 [Spl_P\(0x03H, 表 9-16\)](#)、[Spl_Q\(0x04H, 表 9-17\)](#)、[Spl_S\(0x05H, 表 9-18\)](#)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 5.5296MHz，那么波形数据更新频率可以选择为 28.8kHz、14.4kHz、7.2kHz、3.6kHz、1.8kHz。

注意：由于波形数据更新频率最快可以达到 28.8KHz，因此 ADC_IRQ 的中断请求最快也可以达到 28.8KHz，而 CPU 的执行速度最快可以达到 11.0592MHz，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。

9.2.3 有功功率、无功功率和视在功率

ATT7027 提供有功功率、无功功率和视在功率输出寄存器。

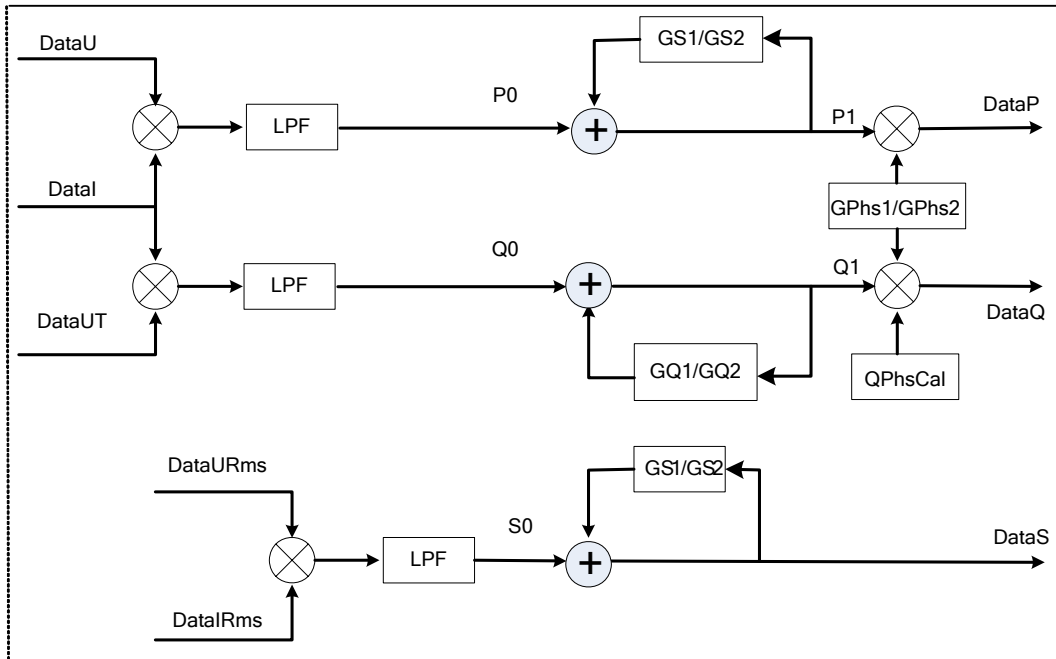


图 9-2 功率计算和补偿

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。

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\(0x49H, 表 9-39\)](#) 对 90° 移相进行相位补偿。QphsCal 默认为 0，在 femu 为 5.5296MHz 时，对应到 50Hz 输入信号可以实现准确的 90° 移相。

9.2.4 电压、电流有效值

ATT7027 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。

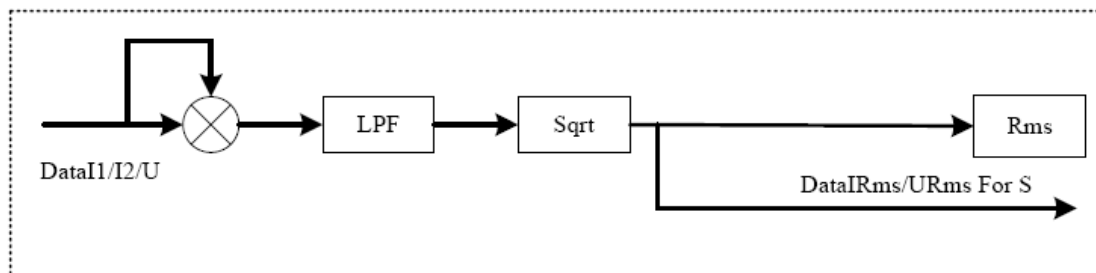


图 9-3 有效值计算

9.2.5 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。

9.2.6 灵活的潜动/起动设置

通过设置寄存器 [PQStart\(0x4EH, 表 9-44\)](#)，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，视在电能不再累加。

另外状态寄存器 [EMUSR\(0x40H, 表 9-30\)](#) 的 NOQLD NOPLD 能够实时显示电能是否启动, 方便用户对阈值的选取。

9.2.7 功率反向指示

通过状态寄存器 [EMUSR\(0x40H, 表 9-30\)](#) 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。

9.2.8 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。

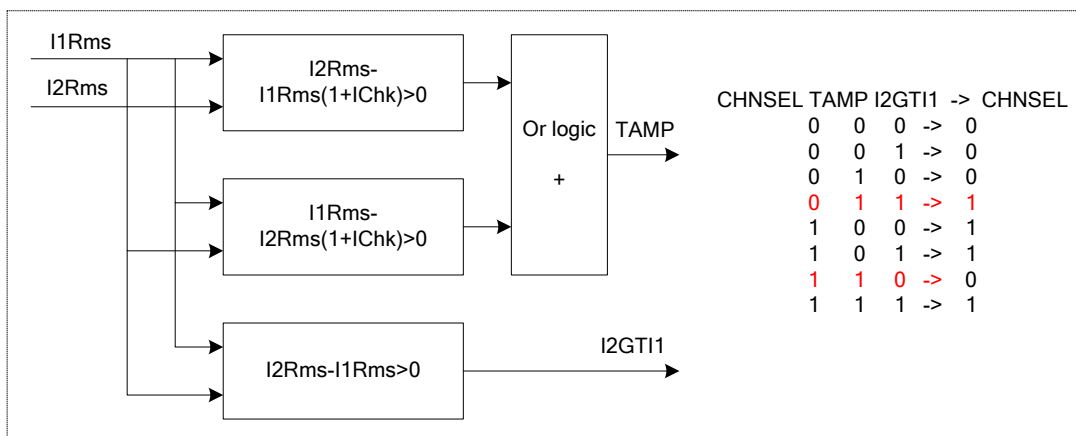


图 9-4 自动防窃电方案

可以通过 [FLTON \(0x52H.5, 表 9-48\)](#) 设置是否开启自动防窃电功能。FLTON=0 时, 用户可以根据 [CHNSEL\(0x51H.4, 表 9-47\)](#) 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\(0x51H.3, 表 9-47\)](#) 选择将两路电流先进行矢量加, 然后再进行计量。

通过 [ICHK\(0x50H, 表 9-50\)](#) 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\(0x59H, 表 9-55\)](#) 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。

[I2GTI\(0x51H.5, 表 9-47\)](#) 为 0 表示 I1 大于 I2, 为 1 时表示 I2 大于 I1。

[TAMP\(0x51H.6, 表 9-47\)](#) 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 [ADC2ON\(0x51H.7, 表 9-47\)](#) 开启 ADC2 通道。
- 通过 [I2GAIN\(0x4AH, 表 9-40\)](#) 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\(0x50H, 表 9-46\)](#)。
- 根据需要检测防窃电的最小电流, 设置 [ITAMP\(0x59H, 表 9-55\)](#)。
- 设置 [FLTON\(0x52H.5, 表 9-48\)](#) 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。

9.2.9 直流偏置自动校正和直流偏置校正寄存器

用户通过 [CHNLCCR\(0x52H, 表 9-48\)](#) 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 0 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 [IIOff\(0x4BH, 表 9-41\)](#)、[I2Off\(0x4CH, 表 9-42\)](#)、[UOff\(0x4DH, 表 9-43\)](#)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\(0x54H, 表 9-50\)](#)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 5.5296MHz 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\(0x00H, 表 9-13\)](#)、[SPL_I2\(0x01H, 表 9-14\)](#)、[SPL_U\(0x02H, 表 9-15\)](#)的多次平均值，设置相应的OFFSET值。

9.2.10 能量寄存器和脉冲输出单元

ATT7027 提供有功能量寄存器 [ENERGY_P\(0x0DH, 表 9-26\)](#)、无功能量寄存器 [ENERGY_Q\(0x0EH, 表 9-27\)](#)和视在能量寄存器 [ENERGY_S\(0x0FH, 表 9-28\)](#)，同时提供相应的脉冲输出引脚PF和QF。

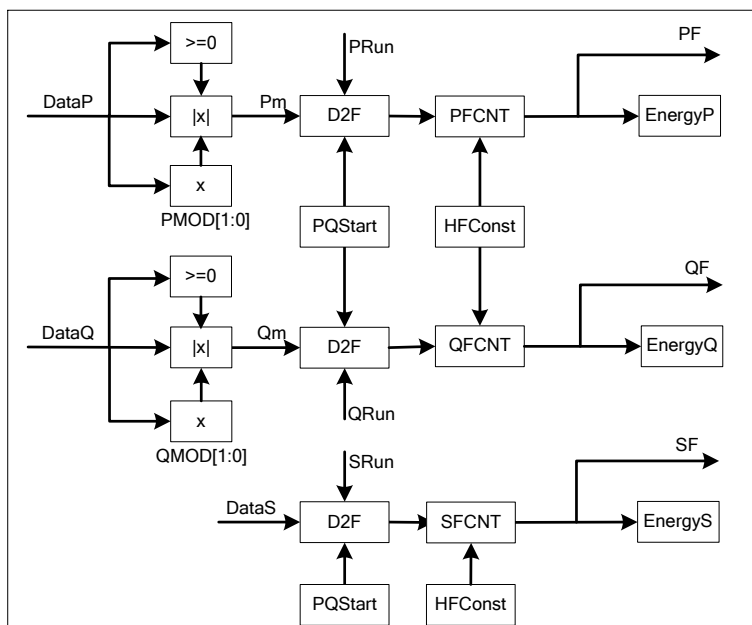


图 9-5 能量计量和脉冲输出

电能可以通过 [EMCON\(0x53H, 表 9-49\)](#)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\(0x55H, 表 9-51\)](#)、[QFCNT\(0x56H, 表 9-52\)](#)和 [SFCNT\(0x57H, 表 9-53\)](#)。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\(0x4FH, 表 9-45\)](#)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。

ATT7027 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和能量寄存器受到 [EMCON\(0x53H, 表 9-49\)](#)的PRun/QRun/SRun以及 [PQStart\(0x4EH, 表 9-44\)](#)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, 视在电能不再累加。

用户可以通过[POS \(0x52H.6, 表9-48\)](#) 选择PF/QF的有效电平。POS为0时, 脉冲高电平有效; POS为1时, 脉冲低电平有效。

PF/QF 输出满足下面时序关系:

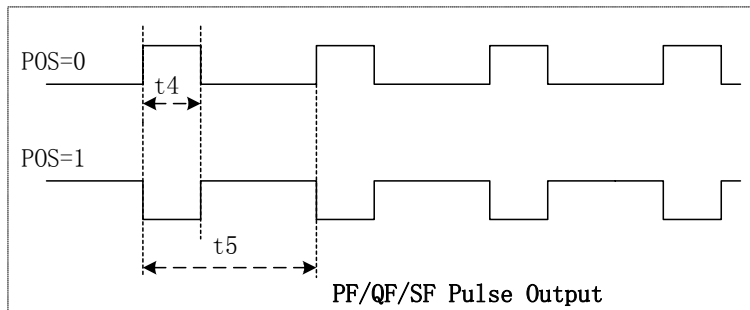


图 9-6 脉冲输出波形

时序特性:

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

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

ATT7027 可以实现输出脉冲宽度 (t4) 可调, 通过 [CFP\[1:0\]\(0x58H.\[7:6\], 表 9-54\)](#) 进行设置。

9.2.11 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

(注: ATT7027 没有视在脉冲输出 pin, 但是 SF_IRQ 仍然可以正常使用)。

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 [ZXD\(0x53H.7, 表 9-49\)](#) 位可以选择正向过零或者反向过零。

ZXD=0: 表示选择正向过零点作为过零中断检测信号;

ZXD=1: 表示选择负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\(0xA9H, 表 9-9\)](#) 和 EMU 中断标志寄存器 [EMUIF\(0xB1H, 表 9-10\)](#), 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 1 即可清除。

9.3 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。

9.3.1 SFR 寄存器

表 9-1 EMU SFR 寄存器列表

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

0xD9	ECADR	1	EMU 校表地址寄存器
0xDB	ECDATH	1	EMU 校表高字节数据寄存器
0xDA	ECDATL	1	EMU 校表低字节数据寄存器
0xDC	EPADR	1	EMU 参数地址寄存器
0xDF	EPDATH	1	EMU 参数高字节数据寄存器
0xDE	EPDATM	1	EMU 参数中字节数据寄存器
0xDD	EPDATL	1	EMU 参数低字节数据寄存器
0xA9	EMUIE	1	EMU 中断使能寄存器
0xB1	EMUIF	1	EMU 中断标志寄存器
0xBF	SUPDC	1	PDEMU 控制位

表 9-2 EMU Calibration Address Register (ECADR, 0xD9H)

EMU Calibration Address Register (ECADR)			Address: D9H					
	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 9-3 EMU Calibration Data High Byte Register (ECDATH, 0xDBH)

EMU Calibration Data High Byte Register (ECDATH)			Address: DBH					
	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 9-4 EMU Calibration Data Low Byte Register (ECDATL, 0xDAH)

EMU Calibration Data Low Byte Register (ECDATL)			Address: DAH					
	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 9-5 EMU Parameter Address Register (EPADR, 0xDCH)

EMU Parameter Address Register (EPADR)			Address: DCH					
	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 9-6 EMU Parameter Data High Byte Register (EPDATH, 0xDFH)

EMU Parameter Data High Byte Register (EPDATH)			Address: DFH					
	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 9-7 EMU Parameter Data Middle Byte Register (EPDATM, 0xDEH)

EMU Parameter Data Middle Byte Register (EPDATM)			Address: DEH					
	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 9-8 EMU Parameter Data Low Byte Register (EPDATL, 0xDDH)

EMU Parameter Data Low Byte Register (EPDATL)			Address: DDH					
	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 9-9 EMU Interrupt Enable Register (EMUIE, 0xA9H)

EMU Interrupt Enable Register (EMUIE)			Address: A9H					
	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 9-10 EMU Interrupt Flag Register (EMUIF, 0xB1H)

EMU Interrupt Flag Register (EMUIF)			Address: B1H					
	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 1，可以将中断标志清 0。

如往 PFIF 写 1，则 PFIF 被清为 0。

表 9-11 System Unit Power-Down Control Register (SUPDC, 0xBFH)

System Unit Power-Down Control Register (SUPDC)			Address: BFH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	PDTPS	PDEMU	PDBOR	PDLVD	PDLCD	PDI2C	保留
Write:	x							
Reset:	0	0	1	0	0	1	1	1

打开写保护后，PDEMU=0：表示使能 EMU 单元；PDEMU=1：表示关闭 EMU 单元。由关闭状态进入到开启状态，EMU 单元复位开始。

最低位保留，复位值为 1，不建议用户往该位写 0。

9.3.2 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：

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

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。

9.3.2.1 计量参数寄存器列表

表 9-12 EPR 寄存器列表(Read Only)

地址 (EPADR)	名称	字节长度	功能描述
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	Reserved	0	保留
11H	Reserved	0	保留
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 9-13 Current Waveform Register (Spl_I1 0x00H)

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

表 9-14 Current Waveform Register (Spl_I2 0x01H)

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 9-15 Voltage Waveform Register (Spl_U 0x02H)

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 9-16 Active Power Waveform Register (Spl_P 0x03H)

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 9-17 Reactive Power Waveform Register (Spl_Q 0x04H)

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 9-18 Apparent Power Waveform Register (Spl_S 0x05H)

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 9-19 Current 1 Rms Register (I1Rms 0x06H)

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 9-20 Current 2 Rms Register (I2Rms 0x07H)

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 9-21 Voltage Rms Register (URms 0x08H)

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=5.5296MHz，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 9-22 Voltage Frequency Register (UFREQ 0x09H)

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=5529600Hz，UFREQ=18350，那么测量到的实际频率为：

$$f = 5529600/6/18350 = 50.2Hz。$$

电压频率测量值更新的周期为 0.7s。

功率参数输出：

表 9-23 Active Power Register (PowerP 0x0AH)

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 9-24 Reactive Power Register (PowerQ 0x0BH)

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 9-25 Apparent Power Register (PowerS 0x0CH)

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

Preg=PowerP 如果 PowerP<2²³

Preg=PowerP-2²⁴ 如果 PowerP>=2²³

设显示的有功功率为 P，转换系数为 Kpqs

则 P=Preg×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*Preg=0.01935*(-47820)= -925.3 w

其中 Preg=PowerP-2²⁴=-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=5.5296MHz，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 9-26 Active Energy Register (EnergyP 0x0DH)

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 9-27 Reactive Energy Register (EnergyQ 0x0EH)

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 9-28 Apparent Energy Register (EnergyS 0x0FH)

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

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

9.3.2.2 校表参数寄存器列表

表 9-29 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	1	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	HFCnst	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	Reserved	0	保留
5CH	Reserved	0	保留
5DH	Reserved	0	保留
5EH	Reserved	0	保留
5FH	Reserved	0	保留

表 9-30 EMU Status Register(EMUSR 0x40H)

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 1 清零

PEOF: 有功电能寄存器溢出标志, 写 1 清零

SEOF: 视在电能寄存器溢出标志, 写 1 清零

功率校正:

表 9-31 Active Power Gain 1 Register(GP1 0x41H)

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 9-32 Reactive Power Gain 1 Register(GQ1 0x42H)

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 9-33 Apparent Power Gain 1 Register(GS1 0x43H)

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 9-34 Phase Calibration 1 Register(GPhs1 0x44H)

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 9-35 Active Power Gain 2 Register(GP2 0x45H)

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 9-36 Reactive Power Gain 2 Register(GQ2 0x46H)

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 9-37 Apparent Power Gain 2 Register(GS2 0x47H)

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 9-38 Phase Calibration 2 Register(GPhs2 0x48H)

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 9-39 Reactive Power Phase Calibration Register(QPhsCal 0x49H)

Reactive Power Phase Calibration Register (QPhsCal)			Address: 49H				
	Bit7	6	5	4	3	2	Bit0
Read:	QPC7	QPC6	QPC5	QPC4	QPC3	QPC2	QPC1
Write:							
Reset:	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。

电流通道 2 增益设置：

表 9-40 Current 2 Gain Register(I2Gain 0x4AH)

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 9-41 Current 1 Offset Register(I1Off 0x4BH)

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 9-42 Current 2 Offset Register(I2Off 0x4CH)

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 9-43 Voltage Offset Register(UOff 0x4DH)

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。

潜动与起动:

表 9-44 Start Power Threshold Setup Register(PQStart 0x4EH)

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，视在电能不再累加。

脉冲频率:

表 9-45 High Frequency Impulse Const Register(HFConst 0x4FH)

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 9-46 Check Current Rms Register(IChk 0x50H)

Check Current Rms Register (IChk)			Address: 50H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	ICHK7	ICHK6	ICHK5	ICHK4	ICHK3	ICHK2	ICHK1	ICHK0
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)。

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

默认为: 0.0625 也即 6.25%。

开启自动窃电后，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 9-47 ADC Config Register(ADCCFG 0x51H)

ADC Config Register (ADCCFG)			Address: 51H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	ADC2ON*	TAMP	I2GTI1	CHNSEL*	CIADD*	SPL2	SPL1	SPL0
Write:		X	X					
Reset:	0	0	0	0	0	0	0	0

ADC2ON: =1: 表示第二通道ADC开启; =0: 表示第二通道ADC关闭，ADC输出恒 0。在 [FLTON](#)=1 时，即开启防窃电模块时，ADC2ON只读；只有FLTON=0 时ADC2ON 可读写。

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 $I2Rms \leq I1Rms$ 。

注: TAMP 和 I2GTI1 标志位只在 FLTON=1 时有效, 否则保持复位值。

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。

FLTON	CIADD	CHNSEL	工作模式
1	只读(恒为 0)	只读(内部决定)	防窃电模式
0	0	0	通道一为有效电流输入
0	0	1	通道二为有效电流输入
0	1	X	电流相加模式

SPL[2:0]: 波形采样中断频率选择, 当 femu=5.5296MHz 时, 选择的频率如下:

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 9-48 Current Channel Control Register(CHNLCCR 0x52H)

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

POS:

POS=0: 表示 PF/QF 为高电平有效;

POS=1: 表示 PF/QF 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。

HPFON12/I1/U:

HPFON=0: 启动数字高通滤波器

HPFON=1: 关闭数字高通滤波器

能量计算:

表 9-49 Energy Measure Control Register(EMCON 0x53H)

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

ZXD:

ZXD=0: 表示选择正向过零点作为过零中断检测信号

ZXD=1: 表示选择负向过零点作为过零中断检测信号

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 9-50 Auto Offset Calibration Register(AutoDC 0x54H)

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 9-51 Active Energy Counter Register(PFCNT 0x55H)

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 9-52 Reactive Energy Counter Register(QFCNT 0x56H)

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 9-53 Apparent Energy Counter Register(SFCNT 0x57H)

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 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 9-54 ADC Channel Gain Register(ADCCON 0x58H)

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	0	0	0	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=5.5296MHz, 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=8
1	0	PGA=16
1	1	PGA=32

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=8	0	1	PGA=8
1	0	PGA=16	1	0	PGA=16
1	1	PGA=32	1	1	PGA=32

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 9-55 Tamper Current Register(ITAMP 0x59H)

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 9-56 Channel Digital Gain Register(DGAIN 0x5AH)

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

9.4 校表过程

以标准表校验为例。

1. 高频脉冲常数设置

femu=5.5296MHz 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\(0x4AH, 表 9-40\)](#)，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 I_{1rms} ，电流通道 2 有效值寄存器读数为 I_{2rms} ，则

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

$$\text{如果 } \text{Gain} \geq 0, I2\text{Gain} = \text{Gain} * 2^{15}$$

$$\text{如果 } \text{Gain} < 0, I2\text{Gain} = \text{Gain} * 2^{15} + 2^{16}$$

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 相似。

简单举例：

假设设计一块 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 $f_{emu} = 5.5296\text{MHz}/2$ ，

1. 计算 HFConst

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

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

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

取整后 HFConst 为 0x6C(108)。将该值写入 [HFConst\(0x4FH, 表 9-45\)](#) 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 $-0.0366 * 2^{15} + 2^{16} = 0xFB50H$

将 0xFB50H 同时写入 [GP1\(0x41H, 表 9-31\)](#)/[GQ1\(0x42H, 表 9-32\)](#)/[GS1\(0x43H, 表 9-33\)](#) 寄存器，即完成阻性增益校正。

3. 相位校正

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

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

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

取整后为 0x4BH，写入角度校正寄存器 [Gphs1\(0x44H, 表 9-34\)](#) 即可。

10 FLASH

10.1 概述

ATT7027 内置 8-bit 可编程高可靠性的 Flash memory。Flash memory 物理上由三部分组成：一个 8K data memory、一个 24K code memory 和一个 64 bytes information memory。Flash memory 的主要特点如下：

- 写入次数(Endurance): Data memory 和 Info memory 最少 500, 000 次写入次数；code memory 最少 20, 000 次写入次数
- 数据保持能力(Data Retention): 室温下 (25℃) 大于 100 年
- 访问时间: 50ns (max)
- 页可擦除:
 - 1) Data memory 64bytes/页
 - 2) Code memory 512bytes/页
 - 3) Information memory 64bytes/页
- 页擦除时间:
 - 1) Data memory: 2ms(min)
 - 2) Code memory: 20ms(min)
- 支持 Mass erase, Mass Erase 时间: 200ms (min)
- 支持字节编程, 字节编程时间: 40us (min)
- 支持程序存储器的写保护功能

10.2 功能描述

10.2.1 PS1 Flash 读操作

要对 [PS1](#) 进行读操作, 可以通过 [PS1MAP \(表 10-4\)](#) 进行选择。

当 [PS1MAP=0](#) 时, [PS1](#) 不可直接通过地址访问, 访问 8000H-803FH 地址时, 返回数据为 00H; 必须利用 [FCADR \(表 10-5\)](#) 和 [FCDAT \(表 10-6\)](#) 两个寄存器进行访问。在 [FCADR](#) 写入地址后, 通过读取 [FCDAT](#) 可以得到对应地址的数据。利用 [FCADR](#) 和 [FCDAT](#) 只能读取 8000H-803FH 地址 64 Bytes 的数据, 而且也仅提供只读功能。

当 [PS1MAP=1](#) 时, [PS1](#) 被 Mapping 到 8000H-803FH 地址, 可以通过地址 8000H-803FH 用 [MOVC](#) 指令读取/修改 [PS1](#) 的内容。

10.2.2 Flash mass 擦除

当 [FOP\[1:0\] \(表 10-4\) = 11](#) 时, 通过 [MOVX](#) 指令可完成 Mass 擦除, [MOVX](#) 指令提供地址。

Flash 物理上由三部分组成: 一个 8K data memory、一个 24K code memory 和一个 64 bytes information memory。相应地, MASS Erase 操作也分为几种情况:

1. [WRS \(表 10-3\) = 1](#), [MOVX](#) 指令指向 PM 空间, 且 [PMLOCK \(表 10-2\) = 1](#), [RSLOCK \(表 10-2\) = 1](#), 关闭 PM flash 写保护功能时:

1) 24K code memory 映射到 PM 0000H-5FFFH, [MOVX](#) 地址落在 PM 0000H-5FFFH 时, 只擦除 24K code memory 部分, 不擦除 8K data memory 和 64 bytes information memory。

2) [DFS\[2:0\] \(表 10-2\) = 0xx](#), 8K data memory 映射到 PM 6000H-7FFFH, 当 [MOVX](#) 地址落在 PM 6000H-7FFFH 时, 只擦除 8K data memory 部分, 不擦除 24K code flash 和 64 bytes information memory。

information flash。

3) PS1MAP (表 10-4) =1, 64 字节 information memory映射到PM 8000H-803FH, 当MOVX地址落在PM 8000H-803FH时, 擦除 8K data memory和 64 bytes information memory, 不擦除 24K code Flash。

2. WRS (表 10-3) =1, 但PMLOCK (表 10-2) =0 或者RSLOCK (表 10-2) =0, 打开 PM flash写保护功能, MASS Erase失效。

3. WRS (表 10-3) =0, MOVX指令指向DM空间。DFS[2:0] (表 10-2) =111, 8K data memory映射到DM 4000H-5FFFH, 当MOVX地址落在DM 4000H-5FFFH时, 只擦除 8K data memory部分, 不擦除 24K code memory和 64 bytes information memory。

Mass擦除开始后BUSY (表 10-4) 为高, Mass擦除完成后BUSY自动清为 0, Mass擦除完成后FOP[1:0] (表 10-4) 也自动恢复为 00 状态。Mass 擦除时间: 200ms (min)。

10.2.3 Flash 页擦除

当FOP[1:0] (表 10-4) =10 时, 通过MOVX指令可完成Page擦除, 其中Page地址由MOVX指令提供。

物理上, Flash 分页情况如下: Data memory 64bytes/页; Code memory 512bytes/页; Information memory 64bytes/页, 即 64bytes Information memory 可当作一页处理。相应地, 页擦除操作也分为几种情况:

1. WRS=1 (表 10-3), MOVX指令指向PM空间, 且PMLOCK (表 10-2) =1、RSLOCK (表 10-2) =1, 关闭flash 写保护功能时:

1) 24K code memory 映射到 PM 0000H-5FFFH, MOVX 地址落在 PM 0000H-5FFFH 时, 擦除 24K code memory 相应的页, 每页 512 bytes。

2) DFS[2:0] (表 10-2) =000 时, 8K data memory映射到PM 6000H-7FFFH, 当MOVX地址落在PM 6000H-7FFFH时, 擦除 8K data memory相应页, 每页 64 bytes。

3) PS1MAP (表 10-2) =1 时, 64 byte information memory映射到PM 8000H-803FH, 当MOVX地址落在PM 8000H-803FH时, 擦除 64 bytes information memory。

2. WRS=1, 但 PMLOCK=0, 打开 PM Flash 的写保护功能, Page Erase 失效。

3. WRS=1, PMLOCK=1, 但 RSLOCK=0, 打开 PM 前 4K 的写保护功能, PM 前 4K 的页不可擦除。其他页擦除和 1 相似。

4. WRS=0, MOVX 指令指向 DM 空间。DFS[2:0]=111, 8K data memory 映射到 DM 4000H-5FFFH, 当 MOVX 地址落在 DM 4000H-5FFFH 时, 擦除 8K data memory 相应页, 每页 64 bytes。

Page擦除开始后BUSY (表 10-4) 为高, Page擦除完成后BUSY自动清为 0; Page擦除完成后FOP[1:0] (表 10-4) 也自动恢复为 00 状态。注意: 多页擦除时, 对下页擦除时要重新设置FOP[1:0]。页擦除时间: Data memory/information memory: 2ms(min); Code memory: 20ms(min)。

10.2.4 Flash 字节编程

FOP[1:0] (表 10-4) =01 时, 采用MOVX指令完成对Flash的字节编程, 即可以采用MOVX @DPTR, A或者MOVX @Ri, A等指令完成写Flash的动作。Byte program操作也分为几种情况:

1. WRS=1 (表 10-3), MOVX指令指向PM空间, 且PMLOCK (表 10-2) =1, RSLOCK (表 10-2) =1, 关闭flash 写保护功能时:

1) 24K code memory 映射到 PM 0000H-5FFFH, MOVX 地址落在 PM 0000H-5FFFH

时，对 24K code memory 字节编程。

2) DFS[2:0] (表 10-2) = 000 时，8K data memory 映射到 PM 6000H-7FFFH，当 MOVX 地址落在 PM 6000H-7FFFH 时，对 8K data memory 相应字节编程。

3) PS1MAP (表 10-4) = 1 时，64 byte information memory 映射到 PM 8000H-803FH，当 MOVX 地址落在 PM 8000H-803FH 时，对相应字节编程。

2. WRS=1，但 PMLOCK=0，打开 PM Flash 的写保护功能，字节编程失效。

3. WRS=1，PMLOCK=1，但 RSLOCK=0，打开 PM 前 4K 的写保护功能，PM 前 4K 的页不可进行字节编程。其他空间字节编程和 1 相似。

4. WRS=0，MOVX 指令指向 DM 空间。DFS[2:0]=111，8K data memory 映射到 DM 4000H-5FFFH，当 MOVX 地址落在 DM 4000H-5FFFH 时，对 8K data memory 相应字节编程。

编程开始后 BUSY (表 10-4) 为高，编程完成后 BUSY 自动清为 0；编程完成后 FOP[1:0] 也自动恢复为 00 状态。注意：多字节编程时，对下一个字节编程时要重新设置 FOP[1:0]。字节编程时间：40us (min)。

10.2.5 PM Flash 写保护

PM Flash 的写保护功能通过配置 FMCFG (表 10-4) 的 PMLOCK 位和 RSLOCK 位实现。当 PMLOCK=1 时，在 UAM 模式下，使能 PM Flash 的擦除/写操作。当 PMLOCK=0，在 UAM 模式下，PM Flash 处于只读模式，不能够被修改。当 RSLOCK=1，在 UAM 模式下，如果 PMLOCK 同时为 1，那么使能前 4K 的 PM Flash 的擦除/写操作。当 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。注意：PMLOCK 和 RSLOCK 只对 PM Flash 有效，对 DM Flash 无效。

10.3 寄存器

表 10-1 FLASH 寄存器列表

地址	名称	复位值	功能描述
0xBE	FMCFG	0x00	Flash memory 配置寄存器—见表 10-2，写保护
0x8F	SPC_FNC	0x00	DM/PM 选择寄存器—见 表 10-3
0xE9	FMCON	0x00	Flash memory 控制寄存器—见表 10-4
0xEE	FCADR	0x00	PS1 Flash Address Register—见表 10-5
0xEF	FCDAT	0x00	PS1 Flash Data Register—见表 10-6

1. Flash memory 配置寄存器（写保护）

表 10-2 Flash memory 配置寄存器 (0xBEH, FMCFG)

FMCFG			Address: BEH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	DFS2	DFS1	DFS0	0	0	BROMEN	PMLOCK	RSLOCK
Write:				X	X			
Reset:	0	0	0	0	0	0	0	0

位	功能描述				
FMCFG[7:5]	DFS[2:0]: 当 DFS[2:0]=111 时，8K-bytes Data Flash Memory 映射为 DM 地址空间：4000-5FFFH；当 DFS[2:0]=0xx 时，Data Flash Memory 映射为 PM 地址空间：6000-7FFFH；复位状态，DFS[2:0]=000，Code Flash 和 Data Flash 组成连续的 32K 程序存储器，地址从 0000-7FFFH。				
	DFS2	DFS1	DFS0	大小分配	地址分配

				PM	DM	PM	DM
	0	x	x	32K	0K	0000-7FFF	0000
	1	1	1	24K	8K	0000-5FFF	4000-5FFF
	Others			x	x	Reserved	Reserved
注意：如果将 DFS[2:0]配置为 0xx 和 111 以外的值将导致不可预期结果。							
FMCFG.1	PMLOCK: PMLOCK=1, 在 UAM 模式下, 使能 PM 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。 注意：PMLOCK和RSLOCK只对PM Flash有效, 对DM Flash无效。						
FMCFG.0	RSLOCK: RSLOCK=1, 在 UAM 模式下, 如果 PMLOCK 同时为 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。 注意：PMLOCK 和 RSLOCK 只对PM Flash 有效, 对DM Flash 无效。						

2. DM/PM 选择寄存器

表 10-3 DM/PM 选择寄存器(0x8FH, SPC_FNC)

SPC_FNC			Address: 8FH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	0	0	0	0	WRS
Write:	X	X	X	X	X	X	X	
Reset:	0	0	0	0	0	0	0	

说明: 在对 Flash 进行 mass erase、page erase 和 byte programe 时, WRS 用以区分写入的地址是 DM flash 的, 还是 PM flash 的; WRS=0 表示可修改 DM 的 Flash, WRS=1 表示可修改 PM 的 Flash。

3. Flash memory 控制寄存器

表 10-4 Flash 访问控制寄存器 (0x8FH, SPC_FNC)

Flash Memory Control Register (FMCON)			Address: E9H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PS1MAP	FOP1	FOP0	0	0	BUSY	SPMOD	0
Write:				X	X	X	X	X
Reset:				0	0	0	0	0

位	功能描述
FMCON.7	PS1MAP: PS1MAP=1: PS1 Flash Memory 被映射到 8000H-803FH 地址空间; PS1MAP=0: PS1 只能通过寄存器 FCADR 和 FCDAT 进行读取操作。
FMCON[6:5]	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 mass 擦除操作。
FMCON[4:3]	保留。 对这两位进行写操作, 将导致不可预期结果。
FMCON.2	BUSY: BUSY=1: 表示 Flash 正在进行写/擦除操作, 忙; BUSY=0: 表示 Flash 空闲, 可以进行操作。
FMCON.1	见 片上ICE 章节
FMCON.0	保留。

4. PS1 Flash Address Register

表 10-5 PS1 Flash Address Register (0xEEH, FCADR)

PS1 Flash Address Register (FCADR)			Address: EEH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	FCADR7	FCADR6	FCADR5	FCADR4	FCADR3	FCADR2	FCADR1	FCADR0
Write:								
Reset:								

说明：当 PS1MAP=0 时，在 FCADR 写入地址后，通过读取 FCDAT 可以得到对应地址的数据，而且也仅提供只读功能。

5. PS1 Flash Data Register

表 10-6 PS1 Flash Data Register (0xEFH, FCDAT)

PS1 Flash Data Register (FCDAT)			Address: EFH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	FCDAT7	FCDAT6	FCDAT5	FCDAT4	FCDAT3	FCDAT2	FCDAT1	FCDAT0
Write:	X	X	X	x	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

说明：和 FCADR 结合使用。

10.4 一种简单实用的写 DATAFLASH 的方法

用户在电表开发中可以使用 ATT7025 Flash 的 8K Data Memory 替代传统电表中用以存放电能等参数的 EEPROM。程序涉及到 Data flash 的页擦除和字节写的操作，这里提供一种简单实用的方法，即通过调用 Monitor ROM 中对 flash 页擦除和字节写函数来完成。在 UAM 模式下这个函数的入口地址为 0xffd9h。将要操作的地址赋值给 DPTR，要写入的数据赋值给累加器 A。用户在调用此函数前注意要关闭全局中断。

下面给出汇编语言的示例程序：

汇编语言的页擦除和字节写操作的程序：

```

MOV    9AH,#0C3H           ; 允许操作被保护的寄存器
MOV    9AH,#9BH
ORL    0BEH,#04H           ; 允许系统调用 Monitor ROM 中的函数

PUSH   IE                  ; 保存 EA 现场
CLR    EA                  ; 关闭全局中断
MOV    8FH,#000H           ; 选择修改 DATA FLASH
MOV    DPTRH0,#40H         ; 指定要擦除的页的高地址
MOV    DPTRL0,#01H         ; 指定要擦除的页的低地址
ORL    0E9,#40H            ; 指定要对 DATA FLASH 进行页擦除操作
MOV    A,#0FFH
LCALL  0FFD9H              ; 调用操作函数进行页擦除

MOV    8FH,#000H           ; 选择修改 DATA FLASH
MOV    DPTRH0,#40H         ; 指定 DATA FLASH 的地址
MOV    DPTRL0,#01H
MOV    A,#0AAH             ; 指定要写入的内容
ORL    0E9,#20H            ; 允许执行 FLASH 写操作
LCALL  0FFD9H              ; 调用操作函数进行字节写操作
POP    IE                  ; 恢复 EA 现场

```

注：每一页的大小为 64 字节，任何落在该页的单个字节存储空间的地址都可以作为该页的地址；DATA FLASH 的地址放在 DPTR 中，要写入的内容放入 A 寄存器中。

11 定时器

11.1 概述

ATT7027 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。由于没有 T0/T1/T2 pin，所以 ATT7027 只能用作定时器方式，计数器功能没有开放。

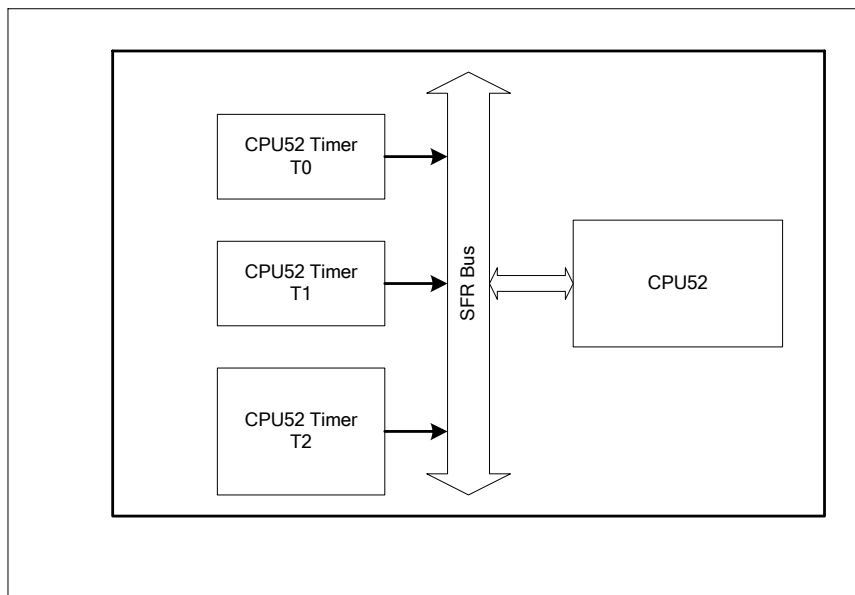


图 11-1 ATT7027 定时器框图

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

11.2 Timer0 和 Timer1

Timer0 和 Timer1 都有 4 种工作模式，由TMODE(表 11-2)和TCON(表 11-3)控制。这四种工作模式是：

- MODE0: 13 位定时器/计数器
- MODE1: 16 位定时器/计数器
- MODE2: 自动装入时间常数的 8 位定时器/计数器
- MODE3: 两个 8 位计数器（只对 Timer0 有效）

11.2.1 寄存器

表 11-1 Timer0 和 Timer1 寄存器列表

地址	名称	复位值	功能描述
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
0x88	TCON	0x00	定时器控制寄存器
0x89	TMOD	0x00	定时器方式寄存器

Timer0 和 Timer1 寄存器说明:

表 11-2 Timer0 and Timer1 Mode Register(TMODE 0x89H)

Timer0 and Timer1 Mode Register (TMODE)			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	GATE: Timer1 运行控制位, 用来确定外部中断请求引脚int1_n是否参与T1 操作控制。当GATE=0 时, 只要定时器控制寄存器的TR1 被置 1, T1 被允许开始计数; 当GATE=1 时, 不仅要TCON中的TR1 被置 1, 还需要int1_n引脚为高电平, 才允许计数 (参阅 图 11-2)。			
TMOD. 6	$C/\bar{T}$: 定时器方式或计数器选择方式选择位。当 $C/\bar{T}=0$, Timer1 是定时器, 定时器的时钟是 clk/4 还是 clk/12 由 T1M (CKCON.4) 的状态决定; 当 $C/\bar{T}=1$, Timer1 是计数器, 用来计 t1 引脚的脉冲个数。			
TMOD.5	M1: Timer1 模式选择位 1			
TMOD. 4	M0: Timer1 模式选择位 0			
	M1	M0	方式	说明
	0	0	0	13 位定时器/计数器
	0	1	1	16 位定时器/计数器
	1	0	2	自动装入时间常数的 8 位定时器/计数器
	1	1	3	停止工作
TMOD.3	GATE: Timer0 运行控制位, 用来确定外部中断请求引脚int0_n是否参与T0 操作控制。当GATE=0 时, 只要定时器控制寄存器的TR0 被置 1, T0 被允许开始计数; 当GATE=1 时, 不仅要TCON中的TR0 被置 1, 还需要int0_n引脚为高电平, 才允许计数 (参阅 图 11-2)。			
TMOD.2	$C/\bar{T}$: 定时器方式或计数器方式选择位。当 $C/\bar{T}=0$, Timer0 是定时器, 定时器的时钟是 clk/4 还是 clk/12 由 T0M(CKCON.3)的状态决定; 当 $C/\bar{T}=1$, Timer0 是计数器, 用来计 t0 引脚的脉冲个数。			
TMOD.1	M1: Timer0 模式选择位 1			
TMOD.0	M0: Timer0 模式选择位 0			
	M1	M0	方式	说明
	0	0	0	13 位定时器/计数器
	0	1	1	16 位定时器/计数器
	1	0	2	自动装入时间常数的 8 位定时器/计数器
	1	1	3	2 个 8 位计数器

表 11-3 Timer0 and Timer1 Control Register (TCON 0x88H)

Timer0 and Timer1 Control Register (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	TF1: Timer1 的溢出中断标志位。当 Timer1 计数溢出时由硬件自动置 1, 进入中断程序后硬件自动清零。

TCON.6	TR1: Timer1 的运行控制位。Timer1 的该位置 1 或清 0 用来启动计数或者停止计数。
TCON.5	TF0: Timer0 的溢出中断标志位。当 Timer0 计数溢出时由硬件自动置 1。
TCON.4	TR0: Timer0 的运行控制位。Timer0 的该位置 1 或清 0 用来启动计数或者停止计数。

11.2.2 功能描述

11.2.2.1 Mode0

当 M1M0 的设置 00 时，定时器选定为方式 0 工作。在这种方式下，16 位寄存器只用了 13 位，TL0 的高三位未用。由 TH0 的 8 位和 TL0 的低 5 位组成一个 13 位计数器。工作方式 0 的逻辑图如图 11-2 所示。

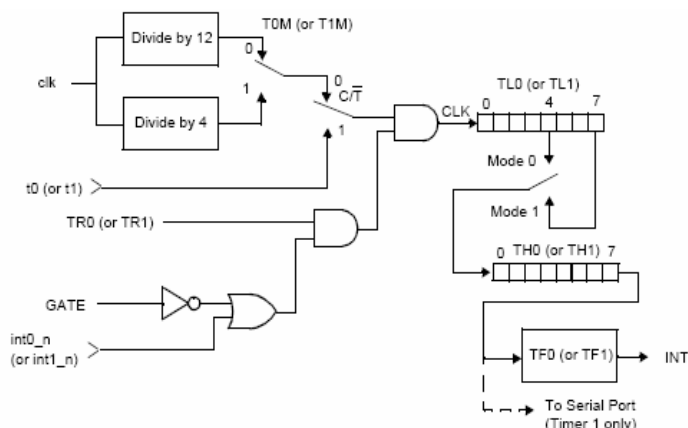


图 11-2 Timer0/1 方式 0 和方式 1

当 GATE=0 时，只要 TCON 中的 TR0 为 1，TL0 及 TH0 组成的 13 位计数器就开始计数；当 GATE=1 时和 TR0=1 时，TH0+TL0 是否计数取决于 Int0_n 引脚的信号，当 Int0_n 由 0 变 1 时，开始计数；当 Int0_n 由 1 变 0 时，停止计数；这样就可以用来测量 Int0_n 的脉宽。当 13 位计数器加 1 到全“1”以后，再加 1 就产生溢出。这时，置 TCON 的 TF0 位为 1，同时把计数器变为全“0”。

11.2.2.2 Mode1

方式 1 和方式 0 的工作相同，唯一的差别是 TH0 和 TL0 组成一个 16 位计数器。

11.2.2.3 Mode2

方式 2 把 TL0 配置成一个可以自动恢复初值（初始常数自动装入）的 8 位计数器，TH0 作为常数缓冲器。TH0 由软件预置值，当 TL0 产生溢出时，一方面使溢出标志 TF0 置 1，同时把 TH0 中的 8 位数据重新装入 TL0 中。工作方式 2 的逻辑图如图 11-3 所示。方式 2 常用于定时控制，还用作串口波特率发生器。

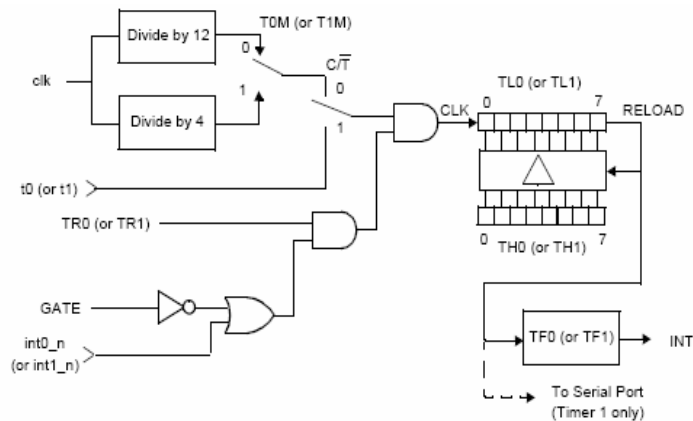


图 11-3 Timer0/1 方式 2

11.2.2.4 Mode3

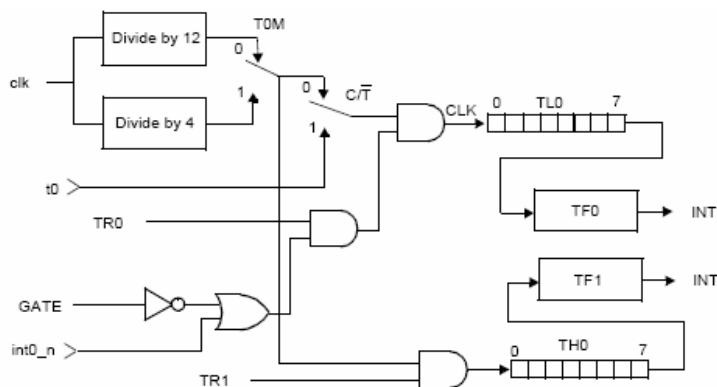


图 11-4 Timer0/1 方式 3

方式 3 对定时器 T0 和定时器 T1 是不相同的。若 T1 设置为方式 3，则停止工作。

当 T0 设置为方式 3 时，将使 TL0 和 TH0 成为两个相互独立的 8 位计数器，如图 11-4 所示。由图可见，TL0 利用了 T0 本身的一些控制（C/T、GATE、TH0、Int0_n），它的操作与方式 0 和方式 1 类似。而 TH0 控制了 T1 的中断。这时 T1 还可以设置为方式 0~2，用于任何不需要中断控制的场合，或用作串口波特率发生器。通常，当 T1 用作串行口波特率发生器时，T0 才定义为方式 3，以增加一个 8 位计数器。

11.3 Timer2

Timer2 有 3 种工作模式，由 T2CON(表 11-5)控制。这 3 种工作模式是：

- 16 位定时器/计数器
- 自动装入时间常数的 16 位定时器/计数器
- 波特率产生模式

11.3.1 寄存器

表 11-4 Timer0 和 Timer1 寄存器列表

地址	名称	复位值	功能描述
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	0x00	定时器 2 控制寄存器

0xCA	RCAP2L	0x00	自动装载寄存器低 8 位
0xCB	RCAP2H	0x00	自动装载寄存器高 8 位

表 11-5 Timer2 Control Register (T2CON 0xC8H)

Timer2 Control Register (T2CON)		Address: C8H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	TF2	0	RCLK	TCLK	0	TR2	$C/\bar{T}2$	0
Write:		x			x			x
Reset:	0	0	0	0	0	0	0	0

位	功能描述
T2CON.7	TF2: Timer2 的溢出中断标志位。当 Timer2 计数溢出时由硬件自动置 1。TF2 必须由软件清零。只有在 RCLK 和 TCLK 都为 0 时, TF2 能可以被置 1, 将 TF2 置 1 且 T2 中断打开, 将产生 T2 中断。
T2CON.6	保留。复位值为 0, 用户改变此位的状态将产生不可预期的结果。
T2CON.5	RCLK: 接收时钟标志。决定 Timer 1 或 Timer 2 是否用作串口 0 模式 1 或 3 接收数据的波特率发生器。当 RCLK=1 时, 选择 Timer2 的溢出作为接收时钟。
T2CON.4	TCLK: 发送时钟标志。决定 Timer 1 或 Timer 2 是否用作串口 0 模式 1 或 3 发送数据的波特率发生器。当 TCLK=1 时, 选择 Timer2 的溢出作为发送时钟。
T2CON.3	保留。复位值为 0, 用户改变此位的状态将产生不可预期的结果。
T2CON.2	TR2: Timer2 的运行控制位。Timer1 的该位置 1 或清 0 用来启动计数或者停止计数。
T2CON.1	$C/\bar{T}2$: 定时器方式或计数器方式选择位。当 $C/\bar{T}2=0$, Timer2 是定时器, 除了波特率产生模式, 定时器的时钟是 $clk/4$ 还是 $clk/12$ 由 CKCON.5 的状态决定, 在波特率产生模式, 定时器的时钟是 $clk/2$, 不由 CKCON.5 控制; 当 $C/\bar{T}2=1$, Time2 是计数器, 用来计 t2 引脚的脉冲个数。
T2CON.0	保留。复位值为 0, 用户改变此位的状态将产生不可预期的结果。

表 11-6 Timer2 模式控制总结

RCLK	TCLK	TR2	Timer2 模式
0	0	1	自动装入时间常数的 16 位定时器/计数器
1	x	1	波特率产生模式
x	1	1	波特率产生模式

11.3.2 功能描述

11.3.2.1 16 位定时器/计数器

图 11-5 说明 Timer2 16 位定时器/计数器模式。 $C/\bar{T}2$ 位决定 16 位定时器/计数器的时钟是来自系统时钟还是来自 t2 引脚的脉冲, 若来自系统时钟, Timer2 是定时器, 若来自 t2 引脚的脉冲, Timer2 是计数器。T2M(CKCON.5)位决定 16 位定时器的时钟来自 $clk/4$, 还是来自 $clk/12$ 。TR2 位决定是否启动 Timer2。当计数到 FFFFH 时, TF2 位置 1, 指示溢出发生。

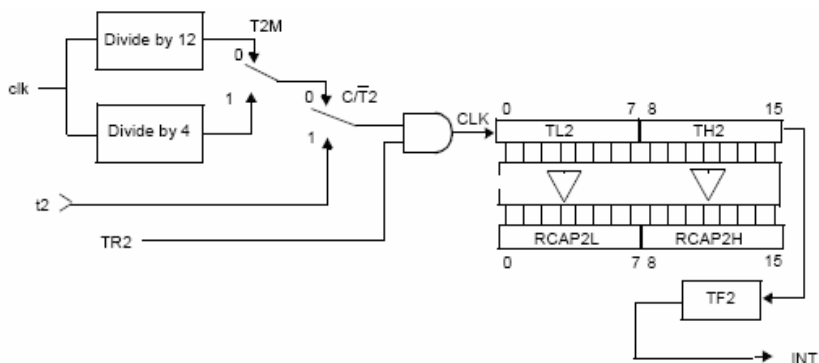


图 11-5 Timer2 16 位定时器/计数器模式

11.3.2.2 自动装入时间常数的 16 位定时器/计数器

图 11-6 说明 Timer2 自动装入时间常数的 16 位定时器/计数器模式，定时器/计数器的输入控制同 16 位定时器/计数器模式，当 TL2 和 TH2 计数值增加到 FFFFH 时，Timer2 将 TF2 置位同时将初始值重新装入 TL2 和 TH2 中，为此，程序必须预装载初始值到 RCAP2L 和 RCAP2H 寄存器。

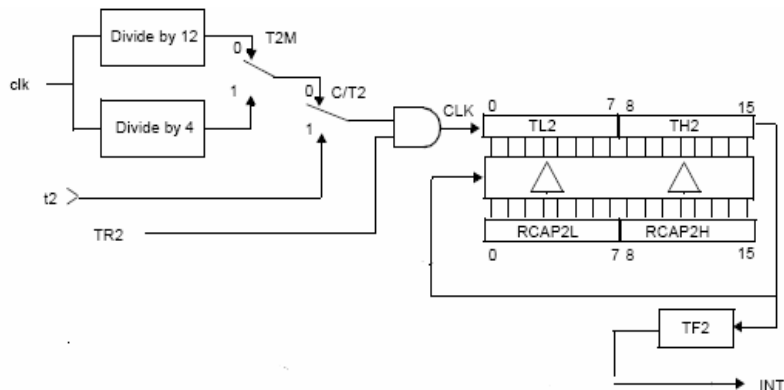


图 11-6 Timer2 自动装入时间常数的 16 位定时器/计数器模式

11.3.2.3 波特率产生模式

图 11-7 说明 Timer2 波特率产生模式。在 RCLK 或 TCLK 置 1 的情况下，Timer2 可作为串口 0 模式 1 或 3 的自动波特率发生器。波特率产生模式和自动装入时间常数的 16 位定时器/计数器模式相似，不同之处在于，自动装入时间常数的 16 位定时器/计数器模式定时器溢出产生 TF2 标志，而波特率产生模式定时器溢出将为串口 0 提供一个移位时钟作为接收或者发送时钟。和自动装入时间常数的 16 位定时器/计数器模式相同，溢出将导致在 RCAP2L 和 RCAP2H 寄存器的初始值移入 TL2 和 TH2 中。

值得指出的是，波特率产生模式的时钟是 $clk/2$ ，如果要用外部时钟源作为定时器的时钟，将 $C/\overline{T}2$ 置 1，并将外部时钟源接到 t2 引脚。

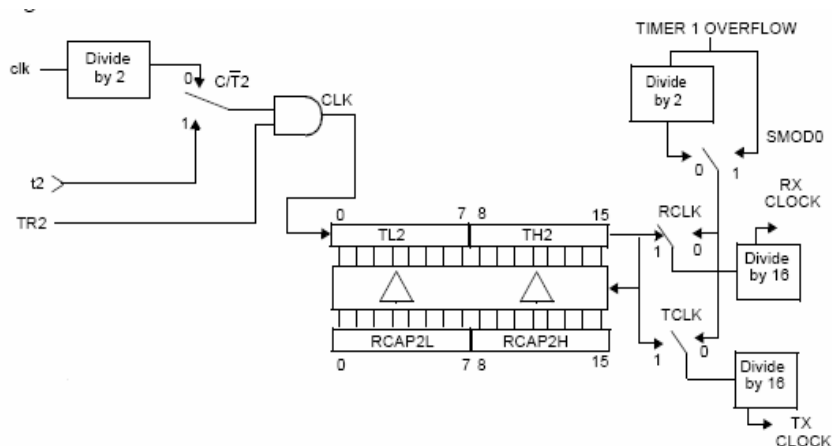


图 11-7 Timer2 波特率产生模式

11.4 定时器/计数器时钟控制

虽然 ATT7027 的机器周期是 4 个系统时钟周期, 但 ATT7027 默认的定时器时钟是 $clk/12$, 和标准 8052 相同, 这保证了将标准 8052 代码移植到 ATT7027 上时, 可以不用修改代码波特率。在程序需要更快的波特率时, 可通过设置时钟控制寄存器 CKCON(0x8Eh) 相应的位, 使 ATT7027 的定时器时钟变为 $clk/4$ 。CKCON(0x8Eh) 说明如下表所示:

表 11-7 Clock Control Register (CKCON 0x8EH)

Clock Control Register (CKCON)			Address: 8EH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	T2M	T1M	T0M	0	0	0
Write:	x	x				x	x	x
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CKCON.7, 6	保留。复位值为 0
CKCON.5	T2M: Timer2 时钟选择。当 T2M=0, Timer2 使用 $clk/12$ 作为时钟; 当 T2M=1, Timer2 使用 $clk/4$ 作为时钟; 波特率产生模式下, 此位失效, Timer2 使用 $clk/2$ 作为时钟
CKCON.4	T1M: Timer1 时钟选择。当 T1M=0, Timer1 使用 $clk/12$ 作为时钟; 当 T1M=1, Timer1 使用 $clk/4$ 作为时钟
CKCON.3	T0M: Timer0 时钟选择。当 T0M=0, Timer0 使用 $clk/12$ 作为时钟; 当 T0M=1, Timer0 使用 $clk/4$ 作为时钟
T2CON.2-0	保留。复位值为 0, 用户改变这些位的状态将产生不可预期的结果。

12 RTC

12.1 功能描述

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 永远不会被关闭，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。

12.1.1 功能概述

- 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
- 7 个周期性中断功能
- 1 个闹钟中断功能
- 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

12.1.2 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（0xF2H）可以对晶体产生的时钟进行校正，校正结果能够保证每分钟内产生的时钟周期数与标准 32768Hz 每分钟内产生的时钟周期数一致。

具体的校正需要通过 TOUT 管脚进行，配置 TOUT[2:0]（0xF1H）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(0xF2H)。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 补偿为每分钟做 5 次，相当于 12 秒补偿一次。（内部实显时并不是等间隔的补偿，但每分钟的补偿点是周期循环的）。

对于 RTCCAL（0xF2H），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）。增加时间，周期变长，代表 RTC 显示的时间变慢；反之，减少时间，RTC 显示的时间变快。

CAL7 为符号位，CAL7=0 为减少时间，N=[CAL6:CAL0]；CAL7=1 为增加时间，N= $\sim$ [CAL6:CAL0]+1。最小解析度为 $1/(32768*12)=1/0.393216\text{ ppm}=2.54\text{ ppm}$ 。

例如：如果配置 RTCCAL=02H，则表示每次补偿减 2 个低频时钟的时间，即相应周期显示的时间加快 $2/32768=61\mu\text{s}$ 。

如果配置 RTCCAL=0xaf=1010_1111，由于最高位为 1，表示要增加时间，增加的个数为低七位 0x2f 的取反加 1，为 0x51（即 81），也就是说相应周期显示的时间减慢 $81/32768=0.002471923828125\text{s}$ 。

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，然后硬件作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 12-1 RTCCAL 操作举例

RTCCAL	RTCCAL 二进制	增加/减少	个数
0x00	0_000_0000	减少	000_0000=0x00=00
0x01	0_000_0001	减少	000_0001=0x01=01
0x50	0_101_0000	减少	101_0000=0x50=80
0x51	0_101_0001	减少	101_0001=0x51=81
0x7f	0_111_1111	减少	111_1111=0x7f=127
0x80	1_000_0000	增加	~(000_0000)+1=000_0000=0x00=00
0x81	1_000_0001	增加	~(000_0001)+1=111_1111=0x7f=127
0x82	1_000_0010	增加	~(000_0010)+1=111_1110=0x7e=126
0xaf	1_010_1111	增加	~(010_1111)+1=101_0001=0x51=81
0xb0	1_011_0000	增加	~(011_0000)+1=101_0001=0x50=80
0xfe	1_111_1110	增加	~(111_1110)+1=000_0010=0x02=2
0xff	1_111_1111	增加	~(111_1111)+1=000_0001=0x01=1

12.1.3 时间和万年历

ATT7027 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。

注意星期需要初始化处理，经过初始化后，该寄存器以 7 天为一个周期循环，与天同时更新。

12.1.4 寄存器写保护功能

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）不可被复位，而且均被写保护。

打开写保护后，有 128 个 cpu clock 时间的限制，相当于最多执行 32 条单周期指令，因而要确保该配置过程不能被中断等过程打断。如果不能保证时，需要先将 EA 关闭，配置好后再将 EA 打开。如果不能保证在规定的机器周期内操作完毕，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。

比如，对秒输出寄存器（0F9H）操作如下。

```

CLR    EA                                ;
MOV    BWPR, #0C3H                      ; select write enable mode
MOV    BWPR, #08BH                      ; enable RTC-proctec-reg write
MOV    0F9H, #010H                     ; config RTC SECR
...                                         ; other RTC config operation,
MOV    BWPR, #0A8H                      ; close RTC reg writing;
                                           ; may neglect, it will be closed in 128 cpu clk

SETB   EA

```

其中 BWPR 为 09AH。

配置寄存器包括：RTCCON、RTCCAL、SECCNT、ALMR、ALHR、RTCIE、RTCIF。这些寄存器中，RTCCAL 不可被复位，而且也属于 RTC 模块的写保护寄存器；其他的配置寄存器属于正常的寄存器：可以被复位，复位后恢复到默认状态，而且不属于写保护寄存器。

12.1.5 中断功能

RTC 一共提供 8 种中断源，共用 MCU 的 IRQ_RTC 中断向量。RTC 的 8 种中断源由 RTCIE (AAH) 控制其使能，中断标志在 RTCIF (B2H) 中。

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。

读带有 ALMF 置为 1 的 RTCIF 寄存器，然后读闹钟小时设置寄存器 ALHR，可以将 ALMF 清 0。

SCNTF：秒定时中断标志

当秒计数到 SECCNT 设定的值时，产生秒计数溢出中断，SCNTF 被置为 1。

读带有 SCNTF 置为 1 的 RTCIF 寄存器，然后读秒计数溢出设置寄存器 SECCNT，可以将 SCNTF 清 0。

RTC2F：定时器 2 中断标志

每 0.25s 或者 0.0625s 产生一次定时器 2 中断，RTC2F 被设置为 1。

读带有 RTC2F 置为 1 的 RTCIF 寄存器，然后读月计数 MTHR，可以将 RTC2F 清 0。

RTC1F：定时器 1 中断标志

每 0.5s 或者 0.125s 产生一次定时器 1 中断，RTC1F 被设置为 1。

读带有 RTC1F 置为 1 的 RTCIF 寄存器，然后读年计数 YRR，可以将 RTC1F 清 0。

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。

读带有 DAYF 置为 1 的 RTCIF 寄存器，然后读日期计数器 DAYR，可以将 DAYF 清 0。

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。

读带有 HRF 置为 1 的 RTCIF 寄存器，然后读小时计数器 HRR，可以将 HRF 清 0。

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。

读带有 MINF 置为 1 的 RTCIF 寄存器，然后读分钟计数器 MINR，可以将 MINF 清 0。

SECF：秒中断

秒计数器 SECR 加 1 时，产生一个秒中断，SECF 被置为 1。

读带有 SECF 置为 1 的 RTCIF 寄存器，然后读秒计数器 SECR，可以将 SECF 清 0。

RTC 清中断注意事项：

RTC 使用低频晶振（32.768kHz）作为时钟，在清中断时，需要等待一个时钟周期后才能正确清除，即从中断产生，到能够清除中断标志，需要等待 30.6us。

正确的 RTC 清中断过程如下：

- 1.清除 RTC IF (0xB2) 的标志（需要保证 30.6us 的等待时间）；
- 2.清除 CPU INT3_N 的标志 EXIF.5。

清除 RTC IF 的所需等待时间有两种方法：

- 1.加入 nop 命令等待；
- 2.循环等待 RTC IF 标志，直到清除为止。

12.2 寄存器

表 12-2 寄存器列表

地址	名称	字节长度	功能描述
F1H	RTCCON	1	RTC 控制寄存器
F2H	RTCCAL	1	RTC 时钟校正寄存器，写保护
F3H	SECCNT	1	秒溢出设置
F4H	ALMR	1	闹钟分钟设置
F5H	ALHR	1	闹钟时钟设置
F9H	SECR	1	秒寄存器，写保护
FAH	MINR	1	分钟寄存器，写保护
FBH	HRR	1	小时寄存器，写保护
FCH	DAYR	1	天寄存器，写保护
FDH	MTHR	1	月寄存器，写保护
FEH	YRR	1	年寄存器，写保护
FFH	DOWR	1	星期寄存器，写保护
AAH	RTCIE	1	RTC 中断使能配置
B2H	RTCIF	1	RTC 中断标志
9AH	BWPR	1	RTC 和其他受保护寄存器的写保护配置寄存器

表 12-3 RTC 控制寄存器

RTC Control Register (RTCCON)			Address: F1H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	TOUTEN1	TOUTEN0	RTCTEN	TESL	TOUT2	TOUT1	TOUT0
Write:	X							
Reset:	0	0	0	0	0	0	0	0

RTCTEN: RTC 秒计数器使能

RTCTEN=0: 秒计数器被关闭

RTCTEN=1: 秒计数器被使能，溢出产生 SECCNT 中断

TESL: RTC 定时器 1/2 的时钟选择

TRSL=0: 定时器 1 中断选择为 0.5s；定时器 2 中断选择为 0.25s。

TRSL=1: 定时器 1 中断选择为 0.125s；定时器 2 中断选择为 0.0625s。

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 12-4 RTC 校正寄存器（写保护）

RTC (RTCCAL)	Calibration	Register	Address: F2H					
	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 12-5 RTC 中断允许寄存器

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

1: 使能中断; 0: 关闭中断。

表 12-6 RTC 中断标志寄存器

RTC Interrupt Flag register (RTCIF)	Address: B2H							
	Bit7	6	5	4	3	2	1	Bit0
Read:	ALMF	SCNTF	RTC2F	RTC1F	DAYF	HRF	MINF	SECF
Write:	x	X	x	x	x	X	x	x
Reset:	0	0	0	0	0	0	0	0

表 12-7 RTC 秒计数溢出设置寄存器

Second Counter Register (SECCNT)	Address: F3H							
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	SCNT5	SCNT4	SCNT3	SCNT2	SCNT1	SCNT0
Write:	x	X						
Reset:	0	0	u	u	u	U	u	u

表 12-8 闹钟分钟设置寄存器

Alarm Minute Register (ALMR)	Address: F4H							
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	AM5	AM4	AM3	AM2	AM1	AM0
Write:	x	X						
Reset:	0	0	u	u	u	U	u	u

允许写入 0-59 以外的数，但是闹钟中断将永远不会产生。

表 12-9 闹钟小时设置寄存器

Alarm Hour Register (ALHR)	Address: F5H							
	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	u	u	U	u	u

允许写入 0-23 以外的数，但是闹钟中断将永远不会产生。

表 12-10 RTC 秒寄存器（写保护）

Second Register (SECR)	Address: F9H							
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
Write:	x	X						
Reset:	0	0	u	u	u	U	u	u

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 12-11 RTC 分钟寄存器（写保护）

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

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 12-12 RTC 小时寄存器（写保护）

Hour Register (HRR)			Address: FBH					
	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	u	u	U	u	U

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 12-13 RTC 日寄存器（写保护）

Day Register (DAYR)			Address: FCH					
	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	u	u	U	u	u

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 12-14 RTC 月寄存器（写保护）

Month Register (MTHR)			Address: FDH					
	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	u	U	u	u

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 12-15 RTC 年寄存器（写保护）

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

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 12-16 RTC 星期寄存器（写保护）

Day-Of-Week Register (DOWR)			Address: FFH					
	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	U	u	u

范围：0-6。写入 0-6 以外的任何数值，对该寄存器没有影响。

表 12-17 RTC 写保护寄存器

Bit Write Protect Register (BWPR)			Address: 9AH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PASS4	PASS3	PASS2	PASS1	PASS0	BITPS	PMOD1	PMOD0
Write:						x		
Reset:	0	0	0	0	0	0	0	0

当 PMOD[1:0]=11 时，写 PASS[4:0]=10011 可以使能所有受保护位的写操作，允许所有受保护位被软件修改，此时写使能处于开启状态；而写 PASS[4:0]=10101 将关闭所有受保护位的写操作，软件不可修改这些受保护的位，此时写使能处于关闭状态。

写使能开启后，如果没有写 PASS[4:0]将写使能关闭，那么在写使能开启后的 128 个 fcpu 时钟周期后，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 128 个 fcpu 时钟周期。

BITPS 表示当前写使能状态，如果写使能被开启，则 BITPS=1，否则 BITPS=0。

PMOD[1:0] 不可以直接被修改，也必须通过 PASS[4:0] 进行控制。只有当 PASS[4:0]=11000 时，才能够修改 PMOD[1:0] 的值，否则写无效。

受保护的寄存器操作有：RTC 的 RTCCAL、SECR、MINR、HRR、DAYR、MTHR、YRR、DOWR。

PMOD[1:0]: 位写保护模式选择

PMOD[1:0]=11：表示可以修改位写保护状态

PMOD[1:0]=00/01/10：表示不可以修改位写保护状态

BITPS: 位写保护状态标志

BITPS=1：表示软件可以修改所有受保护的位

BITPS=0：表示软件不可以修改所有受保护的位

PASS[4:0]: 位写保护密码

PASS[4:0]=11000：使能写 PMOD[1:0] 操作。

PASS[4:0]=10011：开启所有受保护位的写使能。

PASS[4:0]=10001：开启 RTC 所有受保护寄存器的写使能。

PASS[4:0]=10101：关闭所有受保护位和受保护寄存器的写使能。

12.3 低功耗模式

RTC 在所有模式下均正常工作。

12.4 I/O 信号

RTC 具有实时时钟校正输出管脚 TOUT，与 PWM 信号共用，可用于测试校正前、后的 clock 信号。

13 LCD Driver

13.1 概述

ATT7027 的 LCD 驱动单元最多可以支持 25 个 Segment 和 4 个 Common 输出。特征：

- ◆ 软件可编程驱动方式
 - 24 Segments * 4 Commons (96)
 - 25 Segments * 3 Commons (75)
 - 25 Segments * 1 Commons (25)
- ◆ LCD 驱动电压可选
- ◆ 1/3 Bias
- ◆ Static、1/3、1/4 Duty 可选

13.2 功能描述

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。

LCD 数据寄存器 LBUF0—LBUF24（表 13-12），控制 LCD segment 的开关。当将 SxCy 位置为 1 时，对应的 SEGx 与 COMy 将显示；如果写 0 到 SxCy，则 SEGx 与 COMy 对应显示将关闭。

在不需要 LCD 显示时，LCD 单元能够被关闭。PDLCD(0xBFH.2, 表 13-7) 写为 0 时，LCD 被关闭。LCD 关闭后，所有的 SEG 和 COM 都驱动为 VCC 相同的电压，内部电阻分压网络以及模拟电路被关闭，LCD 单元内部时钟被屏蔽。

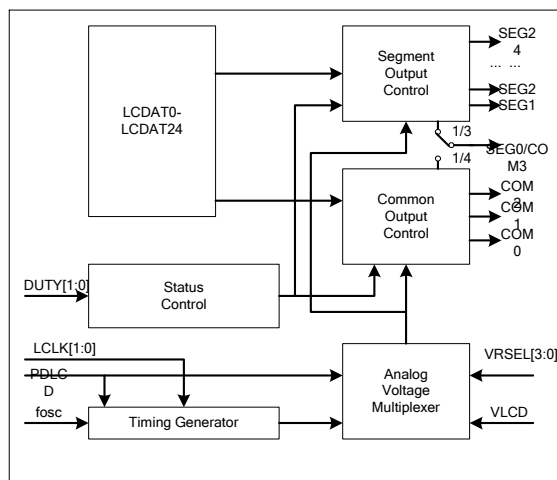


图 13-1 LCD 原理框图

13.2.1 引脚复用

表 13-1 LCD 引脚复用

LCD 引脚定义	功能引脚定义	芯片引脚定义
COM0-COM2	--	COM0 - COM2
COM3/SEG0	--	COM3/SEG0
SEG1-SEG8	PA0-PA7	SEG1/PA0 - SEG8/PA7
SEG9-SEG16	PB0-PB7	SEG9/PB0 - SEG16/PB7
SEG17-SEG24	PC0-PC7	SEG17/PC0 - SEG24/PC7

使用寄存器LCDCFG (0xB9H, 表 13-13)可以对LCD引脚复用进行配置。

13.2.2 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。

- VLCD=VCC
- VLCD1=2/3*(VLCD-Vbias)
- VLCD2=1/3*(VLCD-Vbias)
- VLCD3=Vbias

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

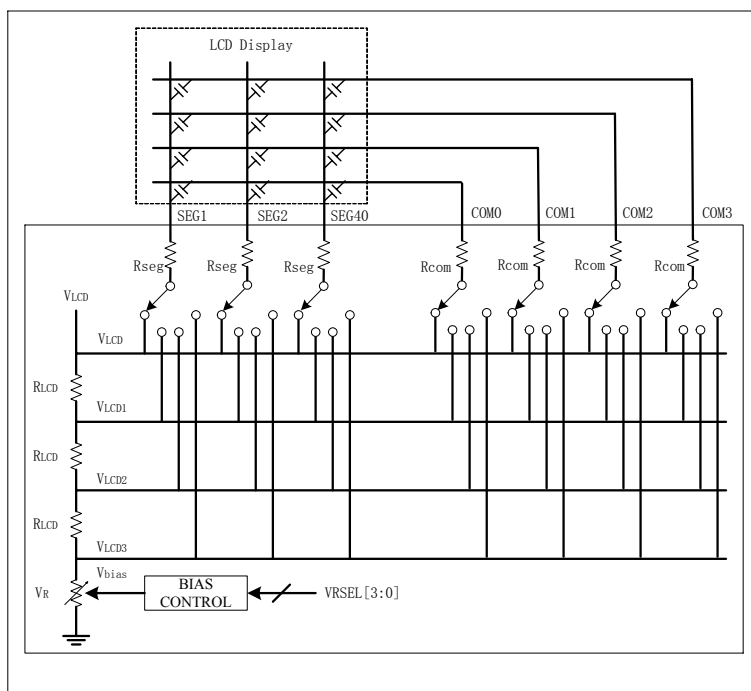


图 13-2 LCD 驱动原理图（1/4 Duty, 1/3 Bias）

13.2.3 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](0xC2H[4:3], 表 13-8)=00: Static——只用 COM0
- DUTY[1:0]=01: 1/3 duty——COM0、COM1、COM2 被使用
- DUTY[1:0]=1x: 1/4 duty——COM0、COM1、COM2、COM3 都被使用

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。

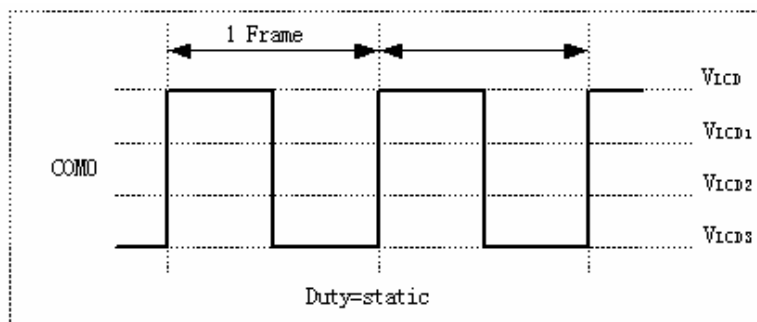


图 13-3 LCD Static Duty 输出波形

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。

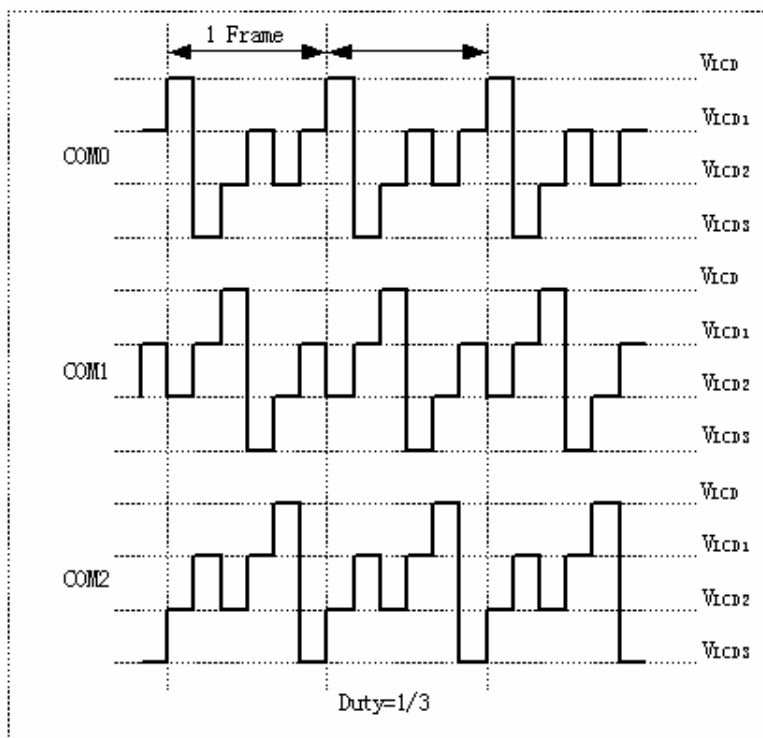


图 13-4 LCD 1/3 Duty 输出波形

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。

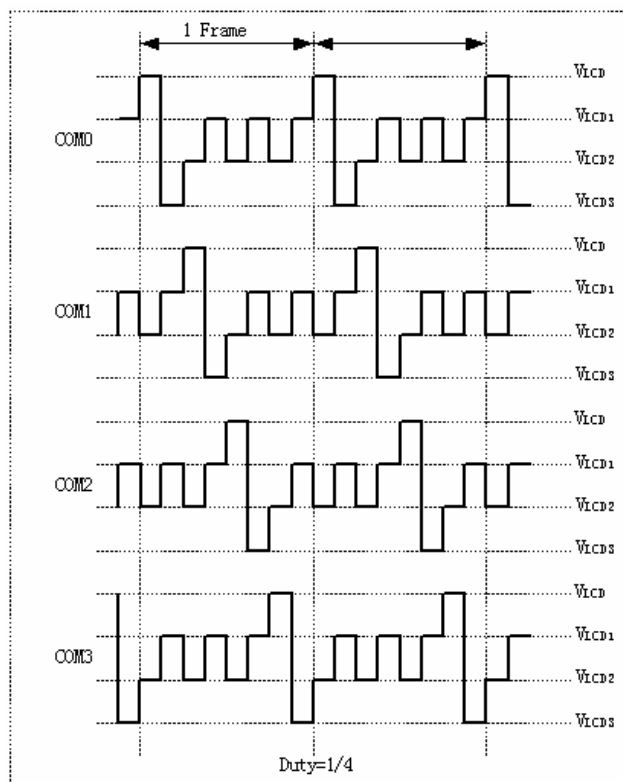


图 13-5 LCD 1/4 Duty 输出波形

13.2.4 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(0xC2H[1:0]，表 13-8)进行配置。

表 13-2 LCD 扫描频率和帧频

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

LCD 帧扫描频率 $ff_{rm} = f_{lcd} \times \text{duty}$ 。

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

13.2.5 LCD 对比度

LCD 显示对比度可通过配置寄存器 VRSEL[3:0]进行调整。VRSEL[3:0]通过调整 V_{bias} 电压，提供 16 级对比度调整范围。调整 V_{bias} 时，即 VLCD3 被调整，而 VLCD1 和 VLCD2 也被相应调整。

设置 VRSEL[3:0](0xC3H[3:0]，表 13-9)=0000 时，表示选择最大对比度，而设置 VRSEL[3:0]=1111 时，表示选择最小的对比度。

表 13-3 LCD 对比度设置

VRSEL3	VRSEL2	VRSEL1	VRSEL0	Bias Volatge (% of VLCD)
0	0	0	0	0.6
0	0	0	1	2.9
0	0	1	0	5.2

0	0	1	1	7.4
0	1	0	0	9.6
0	1	0	1	11.6
0	1	1	0	13.5
0	1	1	1	15.3
1	0	0	0	17.2
1	0	0	1	18.8
1	0	1	0	20.5
1	0	1	1	22.0
1	1	0	0	23.6
1	1	0	1	25.0
1	1	1	0	26.4
1	1	1	1	27.7

13.2.6 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，设置控制位LC(0xC3H.4, 表 13-9)为 1，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(0xC3H.5, 表 13-9)为 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] (0xC2H.[6:5], 表 13-8)进行选择。

表 13-4 快速充电模式设置

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

表 13-5 快速充电时间设置

FCSET1	FCSET0	快速充电时间
0	0	1/32 个 flcd 周期
0	1	1/64 个 flcd 周期
1	0	1/128 个 flcd 周期
1	1	不使用

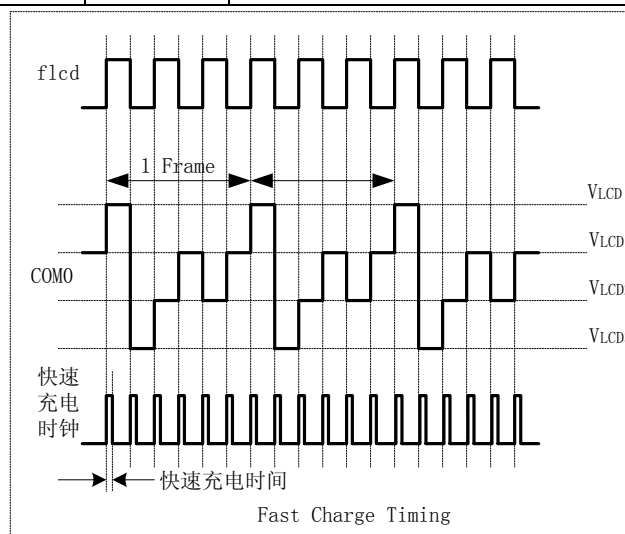


图 13-6 快速充电输出波形

13.2.7 SEGMENT 输出波形

1. Static Duty

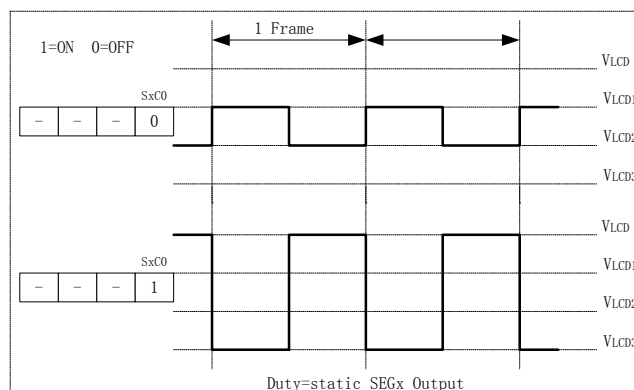


图 13-7 Static SEGMENT 输出波形

2. 1/3 Duty

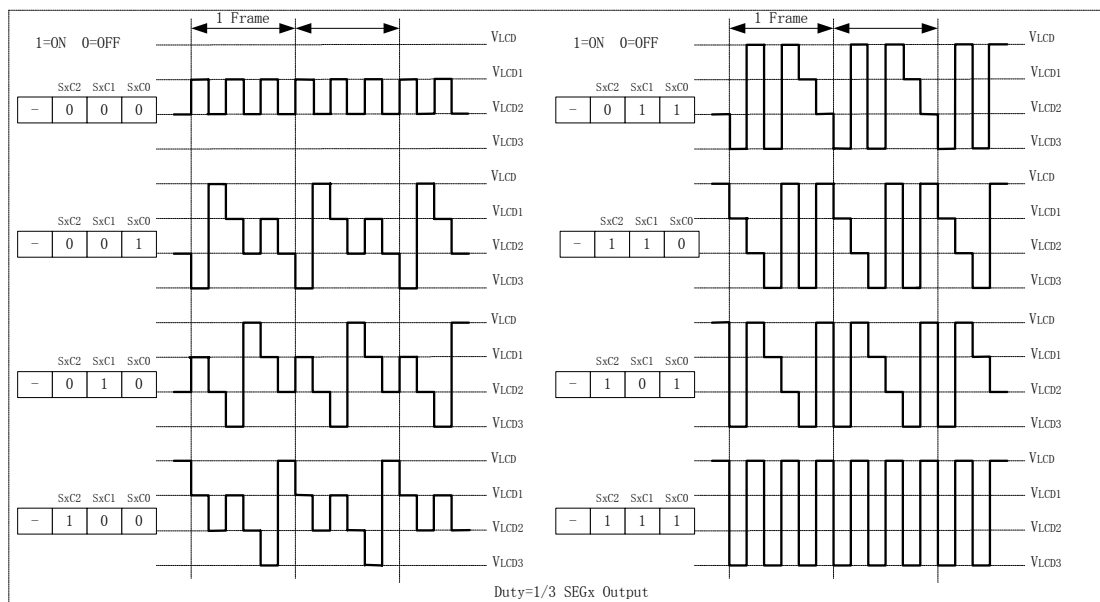


图 13-8 1/3 Duty SEGMENT 输出波形

3. 1/4 Duty

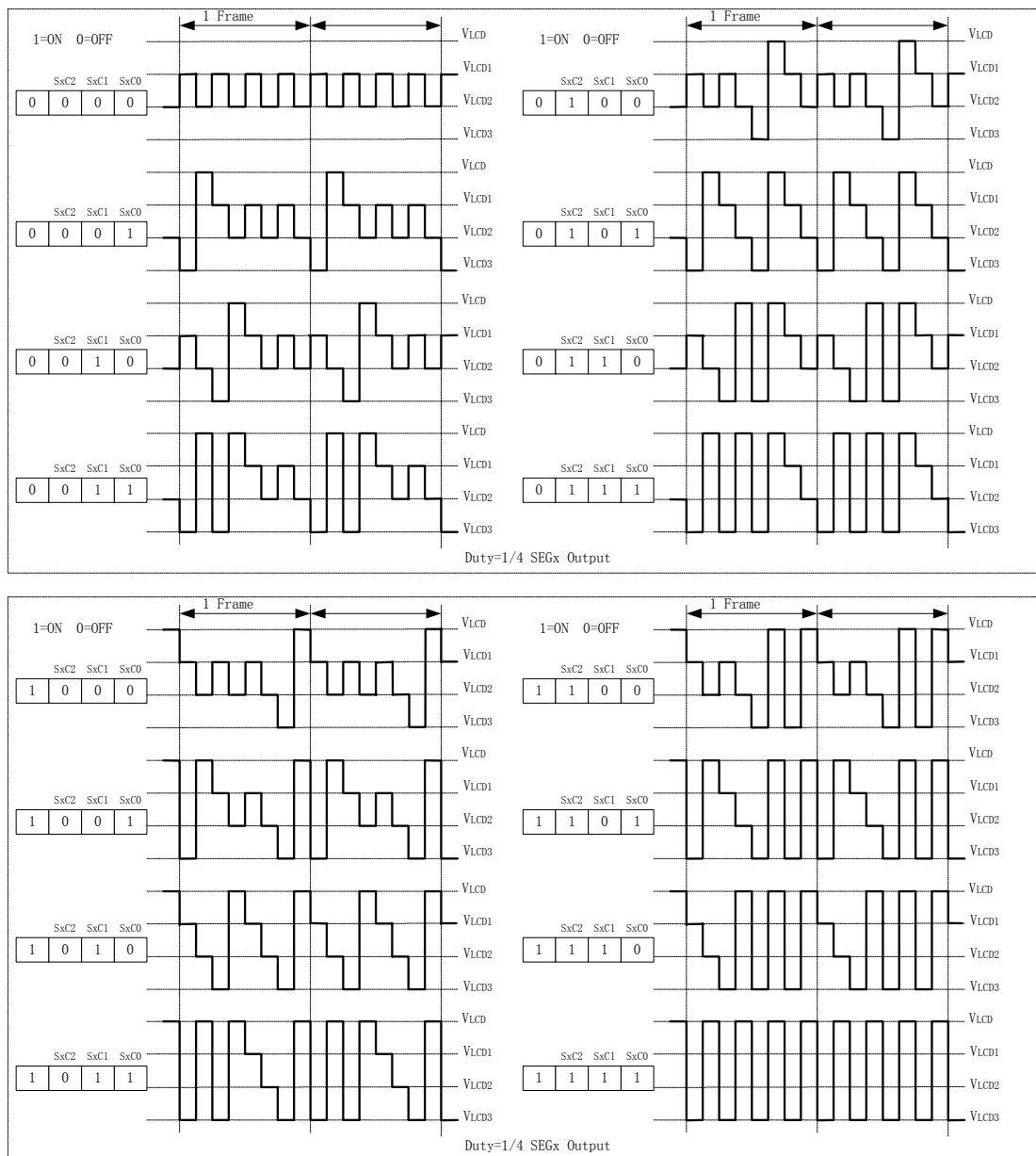


图 13-9 1/4 Duty SEGMENT 输出波形

13.3 寄存器

表 13-6 LCD 寄存器列表

地址(EPADR)	名称	字节长度	功能描述
BFH	PDLCD	1	PDLCD控制位 一见 表 13-7, 写保护
C2H	LCDCLK	1	LCD时钟寄存器 一见 表 13-8
C3H	LCDCR	1	LCD控制寄存器 一见 表 13-9
C4H	LADR	1	LCD地址寄存器 一见 表 13-10
C5H	LDAT	1	LCD数据寄存器 一见 表 13-11
00H-18H	LBUFx	1*25	LCD缓冲数据寄存器 一见 表 13-12
B9H	LCDCFG	1	LCD输出复用配置 一见 表 13-13, 写保护

LCD 寄存器说明:

表 13-7 PDLCD 控制位 (SUPDC 0xBFH)

System Unit Power-Down Control Register (SUPDC)			Address: BFH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	PDTPS	PDEMU	PDBOR	PDLVD	PDLCD	PDI2C	保留
Write:	x							
Reset:	0	0	1	0	0	1	1	1

PDLCD: LCD 单元使能控制位。

打开写保护后, PDLCD=0: 开启 LCD 单元; PDLCD=1: 关闭 LCD 单元。

最低位保留, 复位值为 1, 不建议用户往该位写 0。

表 13-8 LCD Clock Register (LCDCLK 0xC2H)

LCD Clock Register (LCDCLK)			Address: C2H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	FCSET1	FCSET0	DUTY1	DUTY0	0	LCLK1	LCLK0
Write:	x					x		
Reset:	0	0	0	1	0	0	0	0

FCSET[1:0]: 快速充电脉宽选择

DUTY[1:0]: LCD 的 duty 控制选择

LCLK[1:0]: LCD 扫描频率选择

表 13-9 LCD Control Register (LCDCR 0xC3H)

LCD Control Register (LCDCR)			Address: C3H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	FC	LC	VRSEL 3	VRSEL 2	VRSEL 1	VRSEL 0
Write:	X	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[3:0]: LCD 显示对比度调节

表 13-10 LCD Address Register (LADR 0xC4H)

LCD Address Register (LADR)			Address: C4H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	LADR5	LADR4	LADR3	LADR2	LADR1	LADR0
Write:	x	x						
Reset:	0	0	0	0	0	0	0	0

写地址寄存器 LADR, 会根据 LADR[5:0]的索引内容, 将 LBUF_x(x=LADR[5:0])中的数据保存到 LDAT 寄存器中。

表 13-11 LCD Data Register (LDAT 0xC5H)

LCD Data Register (LDAT)			Address: C5H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	LDAT7	LDAT6	LDAT5	LDAT4	LDAT3	LDAT2	LDAT1	LDAT0
Write:								
Reset:	0	0	0	0	0	0	0	0

写数据寄存器 LDAT, 则会根据 LADR[5:0]的索引内容, 将 LDAT 数据写入到 LBUF_x中去(x=LADR[5:0])。

表 13-12 LCD Buffer Data Register (LBUF_x 00H~18H)

LCD Buffer Data Register (LBUF _x)			Address: 00H-18H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	0	SxC3	SxC2	SxC1	SxC0

Write:	x	x	x	x				
Reset:	0	0	0	0	u	u	u	u

LCD 内部缓冲数据寄存器。

表 13-13 LCD 输出复用配置(LCDCFG B9H)

LCDCFG		Address: B9H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	保留	保留	PTCH	PTCL	PTBH	PTBL	PTAH	PTAL
Write:								
Reset:	0	0	0	0	0	0	0	0

此寄存器是写保护寄存器。打开写保护后，

PTAL=0:	SEG1/PA0—SEG4/PA3 配置为标准 I/O:	PA0-PA3
PTAL=1:	SEG1/PA0—SEG4/PA3 配置为 LCD 输出:	SEG1-SEG4
PTAH=0:	SEG5/PA4—SEG8/PA7 配置为标准 I/O:	PA4-PA7
PTAH=1:	SEG5/PA4—SEG8/PA7 配置为 LCD 输出:	SEG5-SEG8
PTBL=0:	SEG9/PB0—SEG12/PB3 配置为标准 I/O:	PB0-PB3
PTBL=1:	SEG9/PB0—SEG12/PB3 配置为 LCD 输出:	SEG9-SEG12
PTBH=0:	SEG13/PB4—SEG16/PB7 配置为标准 I/O:	PB4-PB7
PTBH=1:	SEG13/PB4—SEG16/PB7 配置为 LCD 输出:	SEG13-SEG16
PTCL=0:	SEG17/PC0—SEG20/PC3 配置为标准 I/O:	PC0-PC3
PTCL=1:	SEG17/PC0—SEG20/PC3 配置为 LCD 输出:	SEG17-SEG20
PTCH=0:	SEG21/PC4—SEG24/PC7 配置为标准 I/O:	PC4-PC7
PTCH=1:	SEG21/PC4—SEG24/PC7 配置为 LCD 输出:	SEG21-SEG24
Bit[7:6]:	保留，复位值为 0，不建议用户修改该值。	

13.4 低功耗模式

PSM 模式:

PSM 模式下，LCD 可以继续工作。如果 PSM 模式不需要 LCD，将 LCD 使能控制位 PDLCD 置为 1，即可关闭 LCD。PDLCD 置 1 会使 LCD 单元处于 POWER DOWN 状态。

PDM 模式:

PDM 模式下，LCD 可以继续工作，这取决于进入 PDM 前 PDLCD 位的状态。如果进入 PDM 前 PDLCD 位为 0，并且 LCD 输出复用配置选择为 LCD 输出，那么进入 PDM 模式后 LCD 仍能正常显示，但是数据不再更新；如果进入 PDM 前 PDLCD 位为 1，那么 LCD 单元关闭。

14 温度传感器

14.1 功能描述

ATT7025 内置温度传感器 (TPS)，温度测量值存放于寄存器 TEMPDR (0xC9H)中。

14.1.1 温度计算公式

温度转换公式为：

$$Tr = T_{off} - T_{data} * 0.625$$

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 TEMPDR 转换而来：

$$TEMPDR.7 = 0, T_{data} = TEMPDR[6:0]$$

$$TEMPDR.7 = 1, T_{data} = -(\sim TEMPDR[6:0]) \text{ (按位取反)}$$

14.1.2 温度偏置校正

在得到温度之前，必须得到温度偏置 T_{off}

$$\text{由 } Tr = T_{off} - T_{data} * 0.625$$

可知

$$T_{off} = Tr + T_{data} * 0.625$$

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}。

14.1.3 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 10H，则

$$T_{off} = 25 + 16 * 0.625 = 35$$

将 T_{off} 保存到 flash 或其他非易失 memory 中，在下一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 TEMPDR 的值，得到显示的温度 Tr

Tr(显示)	T _{off}	TEMPDR	T _{data}
25	35	10H	16
35	35	0	0
-44	35	7FH	127
114	35	80H	-127
44	35	F0H	-15

14.2 寄存器

表 14-1 Temperature Sensor Data Register (TEMPDR 0xC9H)

Temperature Sensor Data register (TEMPDR)			Address: C9H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	TPDR7	TPDR6	TPDR5	TPDR4	TPDR3	TPDR2	TPDR1	TPDR0
Write:	x	x	x	x	x	x	x	x
Reset:	0	0	0	0	0	0	0	0

表 14-2 PDTPS 控制位(SUPDC 0xBFH)

System Unit Power-Down Control Register (SUPDC)			Address: BFH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	PDTPS	PDEMU	PDBOR	PDLVD	PDLCD	PDI2C	PDSPi
Write:	x							
Reset:	0	0	1	0	0	1	1	1

打开写保护后，PDTPS=0：表示开启温度传感器；PDTPS=1：表示关闭温度传感器。

14.3 低功耗模式

温度传感可以在所有模式下运行，用户可以根据需要开启和关闭该模块。

15 UART

15.1 概述

ATT7027 支持两个串行口 UART0 和 UART1，特征如下：

- 除了不支持 Mode0，UART0 和 UART1 保留标准 8052 两个串口的其他全部功能
- UART1 输出 TX1 支持 38K 红外调制
- 在应用中，UART0 一般作为调试口和 RS485 通信口，UART1 作为 38K 红外信号调制口
- 两个串口的输入 RX0/RX1 也都可以配置作为中断/INT0 和/INT1 的输入，方便实现自动波特率功能和 PDM 下红外唤醒功能

图 15-1 是 UART0 和 UART1 的功能框图

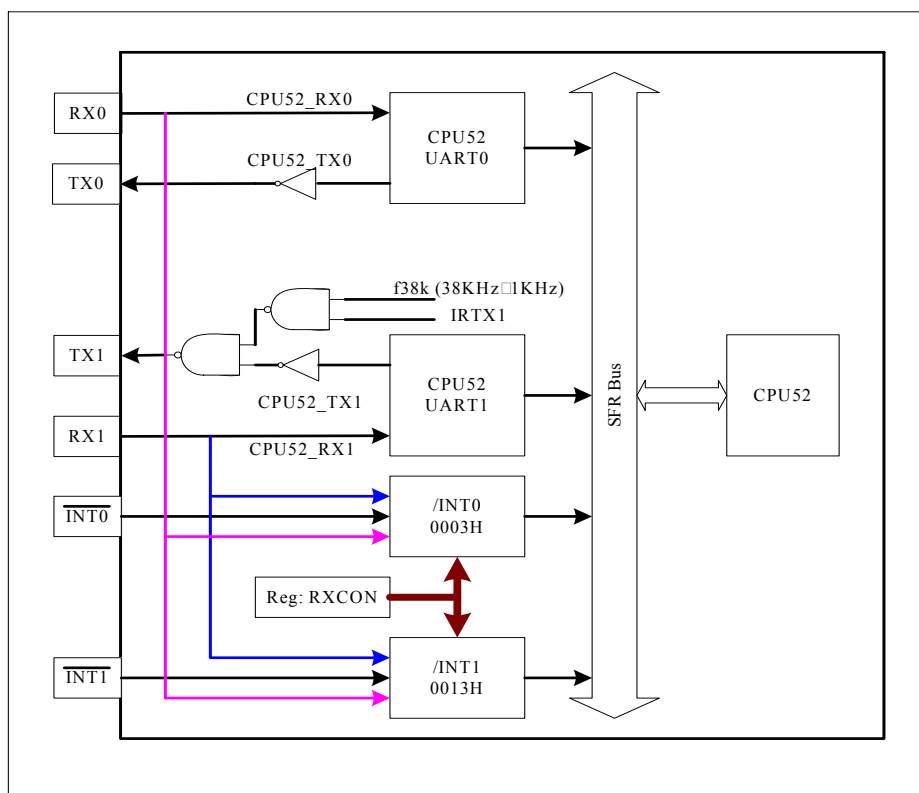


图 15-1 UART0 和 UART1 的功能框图

15.2 功能描述

15.2.1 方式 0

UART0和UART1均不支持方式0

15.2.2 方式 1

方式 1 提供波特率可变的全双工 8 位异步通信。传送一帧信息为 10 位：即 1 位起始位（0）、8 位数据位（低位在先）和 1 位停止位（1）。波特率可变，取决于定时器 1 或 2 的溢出速率。对于接收操作，停止位接收到 RB8_0(或 RB8_1) 中。

15.2.2.1 方式 1 波特率

方式 1 的波特率由定时器的溢出产生。串口 0 可使用定时器 1 或者定时器 2 作为波特率

发生器，串口 1 只能用定时器 1 作为波特率发生器。两个串口如果同时用定时器 1 产生波特率，则两个串口的波特率相同。如果串口 0 用定时器 2 产生波特率而串口 1 用定时器 1 产生波特率，两个串口的波特率可设置的不同。

当定时器计数到最大值（Timer1 FFH，Timer2 FFFFH），就产生一个时钟到波特率发生电路，这个时钟 16 分频产生波特率。

当使用定时器 1，SMOD0(或 SMOD1)位用以选择 Timer1 的翻转速率是否被除以 2。所以当使用定时器 1 时，波特率的由下式决定：

$$\text{波特率} = \frac{2^{\text{SMODx}}}{32} \times \text{Timer1 溢出值}$$

其中 SMOD0 是 SFR PCON.7；SMOD1 是 SFR EICON.7

当使用定时器 2 时，波特率由下式决定：

$$\text{波特率} = \frac{\text{Timer2 溢出值}}{16}$$

虽然可使用 Timer1 的任何一种模式作为波特率发生器，但用户最好使用模式 2（初始常数自动装入），Timer1 的初值存储在 TH1 寄存器中，由此 Timer1 的波特率公式可化为：

$$\text{波特率} = \frac{2^{\text{SMODx}}}{32} \times \frac{f_{\text{sys}}}{12 \times (256 - \text{TH1})}$$

当 SFR CKCON 的 TIM=1 时，Timer1 的时钟为 clk/4，式中系数 12 应换为系数 4。由上述可推导出在已知波特率且 TM1=0 的情况下，TH1 装入值的公式为：

$$\text{TH1} = 256 - \frac{2^{\text{SMODx}} \times f_{\text{sys}}}{384 \times \text{波特率}}$$

当需要更低波特率时，用户也可使用 Timer1 的模式 1 通过中断后重置 TH 和 TL 的方式产生串口所需波特率。

表 15-1 不同波特率下 Timer1 模式 2 TH1 的装入值

波特率	SMODx	C/T	Timer1 模式	Fsys=11.0592MHz 下 TH1 值
57.6Kb/s	1	0	2	0xFFH
19.2 Kb/s	1	0	2	0xFDH
9.6 Kb/s	1	0	2	0xFAH
4.8 Kb/s	1	0	2	0xF4H
2.4 Kb/s	1	0	2	0xE8H
1.2 Kb/s	1	0	2	0xD0H

当使用 Timer2 作为波特率发生器时，设置 Timer2 模式为自动装入时间常数的 16 位定时器/计数器模式，并使能 TCLK、RCLK 位。TCLK=1 表示选择 Timer2 作为发送方向的波特率产生器，RCLK=1 表示选择 Timer2 作为接收方向的波特率产生器。Timer2 的 16 位重装初值存储在 RCAP2H、RCAP2L 寄存器中，所以当使用 Timer2 时，波特率的由下式决定：

$$\text{波特率} = \frac{f_{\text{sys}}}{32 \times (65536 - \text{RCAP2H}, \text{RCAP2L})}$$

式中 16 位无符号数 RCAP2H, RCAP2L 是寄存器 RCAP2H 和 RCAP2L 的内容。设置 RCLK=1 或 TCLK=1 自动使 Timer2 的时钟=clk/2, 这个时钟 16 分频产生波特率, 所以波特率式中分母系数为 32。

由上式可推导出在已知波特率的情况下, RCAP2H 和 RCAP2L 装入值的公式为:

$$\text{RCAP2H, RCAP2L} = 65536 - \frac{f_{\text{sys}}}{32 \times \text{波特率}}$$

表 15-2 不同波特率下 Timer2 RCAP2H 和 RCAP2L 的装入值

波特率	C/T	Fsys=11.0592MHz 下 RCAP2H	Fsys=11.0592MHz 下 RCAP2L
57.6Kb/s	0	0xFFH	0xFAH
19.2 Kb/s	0	0xFFH	0xEEH
9.6 Kb/s	0	0xFFH	0xDCH
4.8 Kb/s	0	0xFFH	0xB8H
2.4 Kb/s	0	0xFFH	0x70H
1.2 Kb/s	0	0xFEH	0xE0H

15.2.2.2 方式 1 发送

图 15-2 说明方式 1 发送时序。在方式 1 中, CPU 执行任何一条以 SBUF 为目标寄存器的指令, 就启动发送。先把起始位输出到 txd, 然后把移位寄存器的输出位送到 txd。接着发出第一个移位脉冲 (SHIFT), 使数据右移一位, 并从左端补入 0。当发送完数据位时, 置位中断标志 TI。

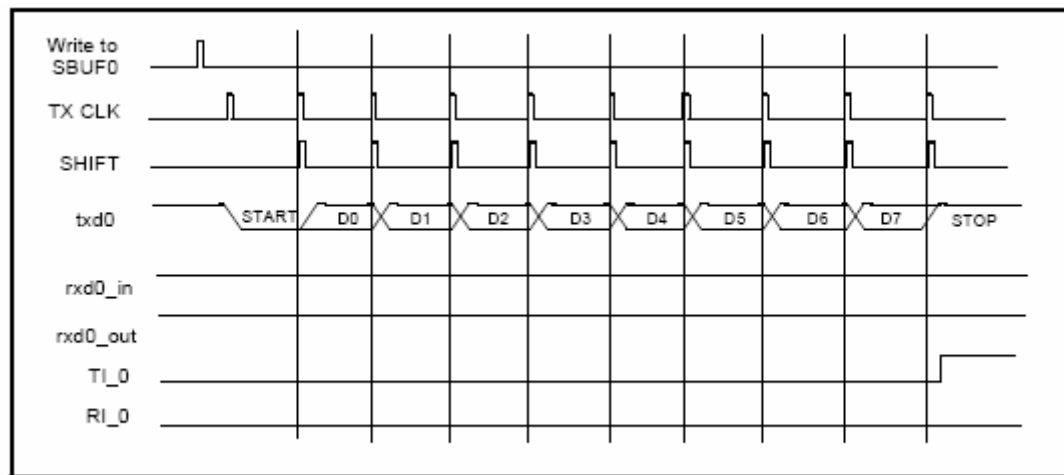


图 15-2 方式 1 发送时序

15.2.2.3 方式 1 接收

图 15-3 说明方式 1 接收时序。串行口以方式 1 输入时, 当检测到 RXD 引脚上由 1 到 0 的跳变时开始接收过程, 并复位内部 16 分频计数器, 以实现同步。计数器的 16 个状态把 1 位时间等分成 16 份, 并在第 7、8、9 个计数状态时采样 RXD 电平, 因此每位数值采样三次, 当接收到的三个值中至少有两个值相同时, 这两个相同值才被确认接收。这样可排除噪声干扰。如果检测到起始位的值不是 0, 则复位接收电路, 并重新寻找另一个 1 到 0 的跳变。当检测到起始位有效时, 才把它移入移位寄存器并开始接收本帧的其余部分。一帧信息也是 10 位: 即 1 位起始位、8 位数据位 (先低位)、1 位停止位。在起始位到达移位寄存器的最

左位时，它使控制电路进行最后一次移位。在产生最后一次移位脉冲时能满足下列两个条件。
1) RI=0; 2)接收到的停止位为 1 或 SM2=0 时，停止位进入 RB8，8 位数据进入 SBUF，且置位中断标志 RI。如果上述两个条件中任何一个不满足，将丢失接收帧。中断标志 RI 必须由用户在中断服务程序中清 0。通常串行口以方式 1 工作时，SM2 置位为“0”。

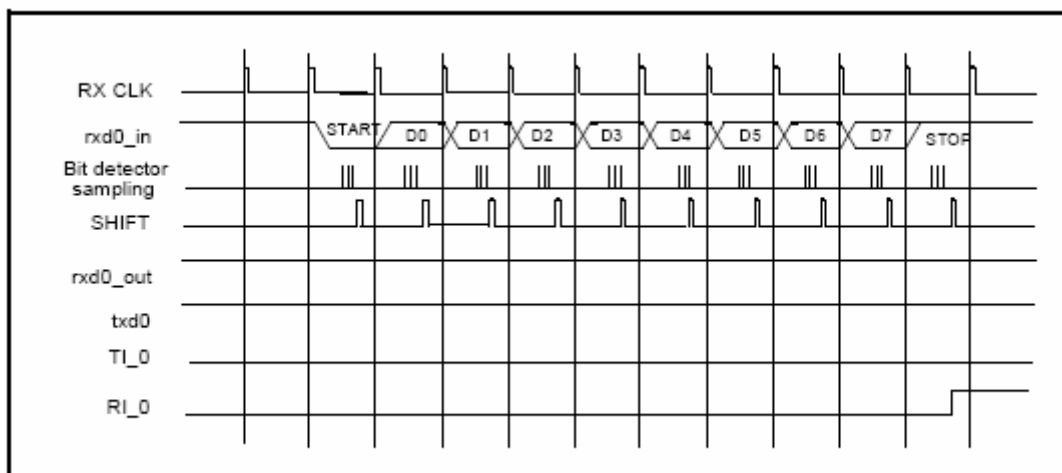


图 15-3 方式 1 接收时序

15.2.3 方式 2

串行口工作在方式 2 时，被定义为 9 位异步全双工通信接口。发送和接收一帧信息都是 11 位：1 位起始位（0）、8 位数据位（低位在先）、1 位可编程位（即第 9 位）和 1 位停止位（1）。方式 2 波特率是固定的，为 $f_{sys}/32$ 或者 $f_{sys}/64$ ，由 SMOD0(SMOD1)决定，波特率公式为：

$$\text{波特率} = \frac{2^{SMODx} \times f_{sys}}{64}$$

15.2.3.1 方式 2 发送

图 15-4 说明方式 2 发送时序。方式 2 的发送过程是由执行任何一条以 SBUF 作为目的寄存器的指令来启动的。由“Write to SBUF”信号把 8 位数据装入 SBUF，同时还把 TB8 装入发送移位器的第 9 位位置上（可由软件把 TB8 赋予 0 或 1），并通知发送控制器要求进行一次发送。发送开始，把一个起始位（0）发到 txd 端。经过一位时间后，数据由移位寄存器送到 txd 端。通过第一位数据，出现第一个移位脉冲。在第一次移位时，把一个停止位“1”由控制位的停止位送入移位寄存器的第 9 位。此后，每次移位，把 0 送入第 9 位。因此，当 TB8 的内容移到移位寄存器的输出位置时，其左面一位是停止位“1”，再往左的所有位全为 0。这种状态由零检测器检测到后，就通知发送控制器作最后一次移位，然后置 TI=1，请求中断。第 9 位数据（即 SCON 中的 TB8 值）由软件置位或清零，可以作为数据的奇偶校验位，也可以作为多机通信中的地址、数据标志位。如果把 TB8 作为奇偶校验位，可以在发送中断服务程序中，在数据写入 SBUF 之前，先将数据的奇偶位写入 TB8。

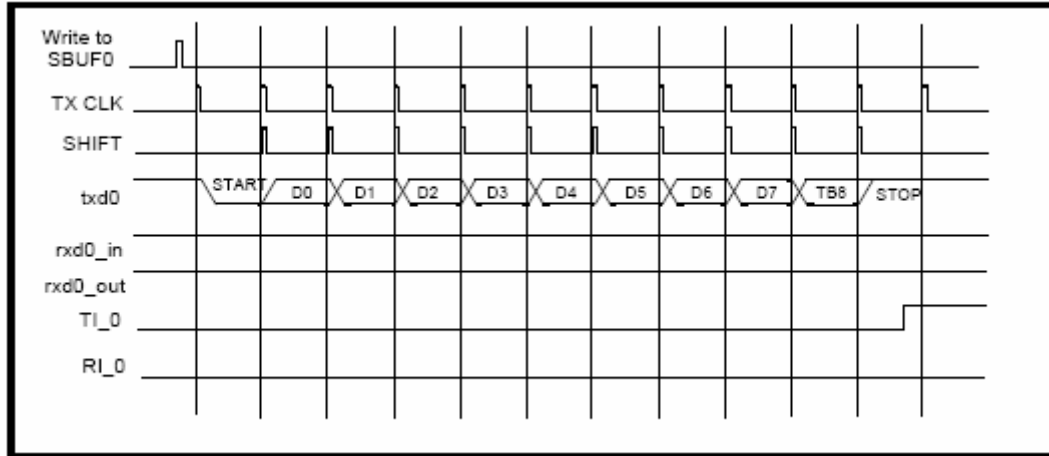


图 15-4 方式 2 发送时序

15.2.3.2 方式 2 接收

图 15-5 说明方式 2 接收时序。方式 2 的接收过程与方式 1 类似。数据从 rxd 端输入，接收过程由 rxd 端检测到负跳变时开始（CPU 对 rxd 不断采样，采样速率为所建立的波特率的 16 倍）。当检测到负跳变，16 分频计数器就立即复位，同时把 1FFH 写入输入移位寄存器。计数器的 16 个状态把一位时间分为 16 份，在每一位的第 7、8、9 个状态，位检测器对 rxd 端的值进行采样。如果所接收到的起始位不是 0，则复位接收电路，等待另一个负跳变的到来。若起始位有效（=0），则起始位移入输入移位寄存器，并开始接收这一帧的其余位。当起始位 0 移到最左面时，通知接收控制器进行最后一次移位。把 8 位数据装入接收缓冲区，第 9 位数据装入 SCON 中的 RB8，并置中断标志 RI=1。数据装入接收缓冲器和 RB8，并置位 RI。只在产生最后一个移位脉冲时，并且要满足下列条件：1) RI=0, SM2=0；2) 接收到的第 9 位数据为 1 时，才会进行。如果不满足上述条件接收到的信息就会丢失，而且中断标志 RI 不置 1。

注意与方式 1 的区别。在方式 2 中装入的 RB8 是第 9 位数据，而不是停止位。所接收到的 RB8 的值可用于多机处理。

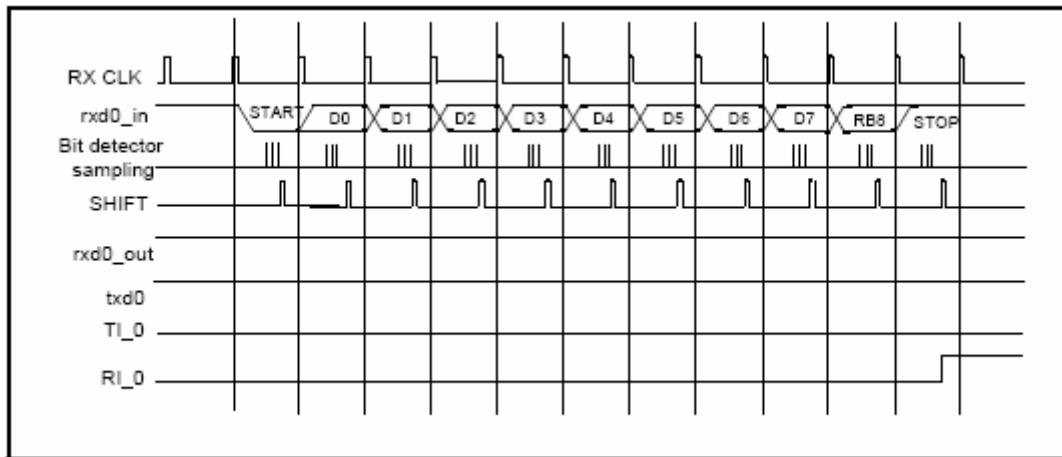


图 15-5 方式 2 接收时序

15.2.4 方式 3

串行口工作在方式 3 时，被定义为 9 位异步全双工通信接口。发送和接收一帧信息都是

11 位：1 位起始位（0）、8 位数据位（低位在先）、1 位可编程位（即第 9 位）和 1 位停止位（1）。方式 3 的发送和接收过程和方式 2 相同，方式 3 的波特率可变，波特率的产生机制和方式 1 相同，也就是说，方式 3 是方式 1 和方式 2 的结合。

15.2.5 多机通信

如前所述，串行口以方式 2 或方式 3 接收时，若 SM2 为 1，则只有当接收器收到的第 9 位数据为 1 时，数据才装入接收缓冲器，并将中断标志 RI 置“1”，向 CPU 发中断申请；如果接收到的第 9 位数据为 0，则不产生中断标志，信息将丢失。而当 SM2 为 0 时，则接收到一个数据字节后，不管第 9 位数据是 1 还是 0，都产生中断标志（RI=1），接收到的数据装入接收缓冲器。利用这个特点，可实现多机通信。

从机系统由从机的初始化程序将串行口编程为方式 2 或方式 3 接收，且置 SM2 为 1。当主机要发送一数据块给从机时，先送出一地址字节，以辨认目标从机。地址字节与数据字节可用第 9 位来区别，发出地址信息时第 9 位为 1，发数据（包括命令）信息时第 9 位为 0。当主机发送地址时，各从机的串行口接收到第 9 位信息（RB8）为 1，则把中断标志 RI 置“1”。这样，使每一台从机都检查一下所接收到的主机发送来的字节是否与本站相符。若为本机地址，则清除 SM2，并准备接收即将到来的数据。没有被寻址的从机，则保持 SM2=1 状态，这些从机将不理睬进入到串行口的数据字节。在主机发送数据时，各从机串行口接收到 RB8 为 0 信息。只有被寻址的从机（已清 SM2）激活中断标志 RI，接收主机的数据（或执行主机的命令），实现和主机的信息传送。其余从机因 SM2 不等于零，且第 9 位 RB8 为 0，不满足接收数据条件，即将所接收到的数据丢失。

15.2.6 UART1 38K 红外调制

UART1 的输出引脚 TX1 支持 38K 红外调制。由图 15-1 所示，38K 红外调制信号实际上是 38K 时钟信号 f38k 和 CPU52 TX1 信号进行或运算的输出。即当 CPU52 TX1=1 时，UART1 TX1 引脚输出为 1；当 CPU52 TX1=0 时，UART1 TX1 引脚输出 f38k 信号。

IRTX1(SFR P3CFG.1)用来控制 UART1 的输出引脚 TX1 38K 红外调制的启停。当 IRTX1=1 时，串口 1 使能 TX1 进行 38K 红外信号调制；当 IRTX1=0 时，串口 1 禁止 TX1 进行 38K 红外信号调制。

f38k 由系统时钟分频产生，值得指出的是，只有当系统时钟来自外部高频晶体振荡时钟 11.0592MHz/5.5296MHz/2.7648MHz/1.3824MHz 时，f38k 才输出 38k 时钟信号，而当系统时钟来自外部低频晶体振荡时钟或者内部 RC 振荡器时，f38k 输出的时钟信号频率小于 38k（详见时钟管理章节），此时若启动 38K 红外调制功能，将导致调制失败。

在使用红外通信时，UART1 的输入引脚 RX1 外接红外解调器可用来接收已经过 38K 红外解调后的信号，因外接红外解调器等效为一个兆级电阻接地，为保证系统在 pdm 模式下 RX1 口不漏电，RX1 引脚没有内部上拉电阻。当用户将 UART1 作为红外通信口时，需外接上拉电阻。

15.2.7 红外输出调制单元的时钟 firf

1. 系统时钟源自 fosc 时，即 fsys=fosc 或者 fsys=fosc/2 时，红外输出调制模块时钟 firf=fsys/36，红外 f38k=firf/4，在此频率下，f38k 输出不等于 38kHz，红外输出不正常。

2. 当 fsys 源自高频振荡电路输出频率时，firf 频率会根据 PRIP[1:0] 选项自动调整，如下表所示，由表知，f38k 输出等于 38kHz，红外输出正常。

表 15-6 红外输出时钟配置

PRIP[1:0]		SYSCK 0	fpri	fsys	firf=fsys/18	IRN	f38k=firf/IRN
0	0	0	11.0592MHz	11.0592MHz	614.4 KHz	16	38.4 KHz
		1	11.0592MHz	5.5296MHz	307.2 KHz	8	38.4 KHz
0	1	0	5.5296MHz	5.5296MHz	307.2 KHz	8	38.4 KHz
		1	5.5296MHz	2.7648MHz	153.6 KHz	4	38.4 KHz
1	0	0	2.7648MHz	2.7648MHz	153.6 KHz	4	38.4 KHz
		1	2.7648MHz	1.3824MHz	76.8 KHz	2	38.4 KHz
1	1	0	1.3824MHz	1.3824MHz	76.8 KHz	2	38.4 KHz
		1	1.3824MHz	0.6912MHz	38.4 KHz	1	38.4 KHz

15.2.8 RX0/RX1 配置为/INT0 和/INT1 的输入

为实现PDM下红外唤醒功能，RX0/RX1 可配置为/INT0 和/INT1 的输入，如 [图 15-1](#)所示。配置SFR RXCON(0xC6H, [表 15-13](#))可实现这个功能。当RXCON不等于 10XXX101 时，CPU52 的/INT0 和/INT1 输入信号直接是芯片引脚/INT0 和/INT1。当RXCON=10XXX101 时，CPU52 的/INT0 和/INT1 输入信号可来自RX0, RX1, INT0 和/INT1。

在使用 UART1 进行红外通信时，通过配置 SFR RXCON,将 RX1 配置为 CPU52 的/INT0 或/INT1, 可实现红外唤醒功能。因 PDM 模式下 RX0, RX1 唤醒实际上利用 CPU52 的/INT0 和/INT1 输入，所以 PDM 模式 RX0, RX1, /INT0 和/INT1 唤醒功能不能同时实现，只能同时实现其中两种。

15.3 寄存器

表 15-7 UART0 和 UART1 寄存器列表

地址	名称	复位值	功能描述
0x98	SCON0	0x00	串口 0 控制寄存器-见表 15-8
0x99	SBUF0	0x00	串口 0 数据缓冲寄存器
0xC0	SCON1	0x00	串口 1 控制寄存器-见表 15-9
0xC1	SBUF1	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 15-10
0xD8	SMOD1	0x40	串口 1 波特率控制位-见表 15-11
0xBB	P3CFG	0x00	P3 输出复用配置寄存器，写保护-见表 15-12
0xC6	RXCON	0x00	RX 控制寄存器-见表 15-13
0xD8	EICON	0x40	扩展中断控制寄存器-见 表 4-6

1. 串口 0 控制寄存器

表 15-8 UART0 Control Register(SCON0 0x98H)

UART0 Control Register (SCON0)			Address: 98H					
Read:	Bit7	6	5	4	3	2	1	Bit0
Write:	SM0_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0
Reset:	0	0	0	0	0	0	0	0

位	功能描述				
SCON0.7	SM0_0: 串口 0 模式选择位 1				
SCON0.6	SM1_0: 串口 0 模式选择位 0				
	SM0_0	SM1_0	方式	功能说明	波特率
	0	0	0	不支持	—

	0	1	1	8 位 UART	可变	
	1	0	2	9 位 UART	fcpu/64 或 fcpu/32	
	1	1	3	9 位 UART	可变	
SCON0.5	SM2_0: 串口0允许方式2和3的多机通信使能位。在方式2或3中, 若SM2_0置为1, 且接收到的第9位数据 (RB8_0) 为0, 则接收中断标志RI_0不会被激活。在方式1中, 若SM2_0=1, 则只有收到有效的停止位时才会激活RI。					
SCON0.4	REN_0: 串口0允许串行接收位。由软件置位或清零, REN_0=1, 允许接收; REN_0=0, 禁止接收					
SCON0.3	TB8_0: 串口0在方式2、3中要发送的第9位数据, 由软件置位或者清零					
SCON0.2	RB8_0: 串口0在方式2、3中已接收的第9位数据, 由软件置位或者清零。在方式1中, 若SM2_0=0, RB8_0是接收到的停止位。					
SCON0.1	TI_0: 串口0发送中断标志位。在发送停止位的开始时由硬件置位, TI_0=1时申请中断, CPU响应中断后, 发送下一帧数据。在任何方式中, 该位都必须由软件清零。					
SCON0.0	RI_0: 串口0接收中断标志位。在接收到停止位的中间时刻由硬件置位, RI_0=1时申请中断, 要求CPU取走数据。但在方式1中, 当SM2_0=1时, 若未接收到有效的停止位, 则不会对RI置位。在任何工作方式中, 该位都必须由软件清零。					

2. 串口 1 控制寄存器

表 15-9 UART1 Control Register(SCON1 0xC0H)

UART1 Control Register (SCON1)		Address: C0H						
	Bit7	6	5	4	3	2	1	Bit0
Read:	SM0_1	SM1_1	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	SM0_1: 串口 1 模式选择位 1				
SCON1.6	SM1_1: 串口 1 模式选择位 0				
	SM0_1	SM1_1	方式	功能说明	波特率
	0	0	0	不支持	—
	0	1	1	8 位 UART	可变
	1	0	2	9 位 UART	fcpu/64 或 fcpu/32
	1	1	3	9 位 UART	可变
SCON1.5	SM2_1: 串口1允许方式2和3的多机通信使能位。在方式2或3中, 若SM2_1置为1, 且接收到的第9位数据 (RB8_1) 为0, 则接收中断标志RI_1不会被激活。在方式1中, 若SM2_1=1, 则只有收到有效的停止位时才会激活RI_1。				
SCON1.4	REN_1: 串口1允许串行接收位。由软件置位或清零, REN_1=1, 允许接收; REN_1=0, 禁止接收				
SCON1.3	TB8_1: 串口1在方式2、3中要发送的第9位数据, 由软件置位或者清零				
SCON1.2	RB8_1: 串口1在方式2、3中已接收的第9位数据, 由软件置位或者清零。在方式1中, 若SM2_1=0, RB8_1是接收到的停止位。				
SCON1.1	TI_1: 串口1发送中断标志位。在发送停止位开始时由硬件置位, TI_1=1时申请中断, CPU响应中断后, 发送下一帧数据。在任何方式中, 该位都必须由软件清零。				

SCON1.0	RI_1: 串口1接收中断标志位。在接收到停止位的中间时刻由硬件置位, RI_1=1时申请中断, 要求CPU取走数据。但在方式1中, 当SM2_1=1时, 若未接收到有效的停止位, 则不会对RI置位。在任何工作方式的, 该位都必须由软件清零。
---------	---

3. 串口 0 波特率控制位

表 15-10 串口 0 波特率控制位 (SMOD0)

PCON			Address: 87H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	SMOD0	0	1	1	GF1	GF0	STOP	IDLE
Write:		X	X	X				
Reset:	0	0	1	1	0	0	0	0

说明: SMOD0 位控制串口 0 的波特率是否加倍。当 SMOD0=1, 串口 0 的波特率加倍。

4. 串口 1 波特率控制位

表 15-11 串口 1 波特率控制位(SMOD1)

EICON			Address: D8H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	SMOD1	1	EPFI	PFI	WDTI	0	0	0
Write:		X				X	X	X
Reset:	0	1	0	0	0	0	0	0

说明: SMOD1 位控制串口 1 的波特率是否加倍。当 SMOD1=1, 串口 1 的波特率加倍。

5. 串口 1 允许 TX1 38K 调制位 (写保护)

表 15-12 串口 1 允许 TX1 38K 调制位(IRTX1)

P3CFG			Address: BBH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	T2	T1	T0	INT1	INT0	IRTX1	0
Write:	X							X
Reset:	0	0	0	0	0	0	0	0

说明: IRTX1 控制输出是否进行 38K 调制。当 IRTX1=1 时, 串口 1 使能 TX1 进行 38K 红外信号调制; 当 IRTX1=0 时, 串口 1 禁止 TX1 进行 38K 红外信号调制。

6. RX 控制寄存器

表 15-13 RX Control Register(RXCON 0xC6H)

RXCON			Address: C6H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	RXCON7	RXCON6	RXCON5	RXCON4	RXCON3	RXCON2	RXCON1	RXCON0
Write:								
Reset:	0	0	0	0	0	0	0	0

说明: 当 RXCON 不等于 10XXX101 时, CPU52 的/INT0 和/INT1 输入信号直接是芯片引脚 /INT0 和/INT1。当 RXCON=10XXX101 时, CPU52 的/INT0 和/INT1 输入信号可来自 RX0, RX1, INT0 和/INT1, 下表列出了选择配置关系:

RXCON[7:0]	CPU52 中断/INT0 输入信号	CPU52 中断/INT1 输入信号
= 10 000 101	/INT0	/INT1
= 10 001 101	RX0	/INT1
= 10 010 101	RX1	/INT1
= 10 011 101	RX0	RX1
= 10 100 101	/INT0	/INT1
= 10 101 101	/INT0	RX0
= 10 110 101	/INT0	RX1
= 10 111 101	RX1	RX0

不等于 10 xxx 101	/INT0	/INT1
----------------	-------	-------

16 I2C

16.1 概述

ATT7027 的 I2C 有如下特征：

- 软件可控制 Acknowledge 位的生成
- 中断方式实现多字节传输
- 呼叫地址识别中断
- 自动检测 R/W 位和选择发送、接收模式
- 检测起始 START、重复起始 RESTART 和停止 STOP 信号
- 主模式自动生成 START、STOP 条件
- 主模式无应答 No-ACK 检测
- 8 种主模式时钟选择
- 自动识别接收应答 Acke 位
- 支持 7 位地址格式

SDA 和 SCL 总线时序和电气参数请参考 I2C 总线规范。

16.2 功能描述

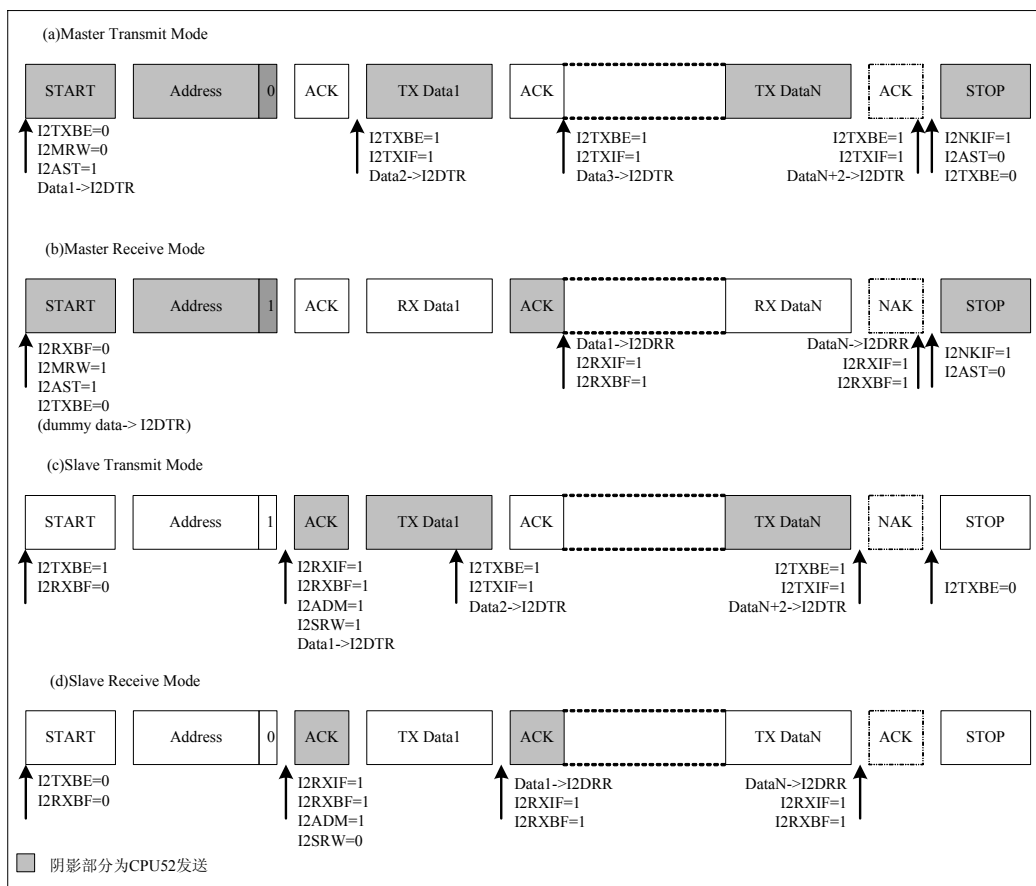


图 16-1 主从传输模式

从上图可以看出，I2C传输模式分为四种：主发送、主接收、从发送、从接收，用户可以根据需要灵活选择。

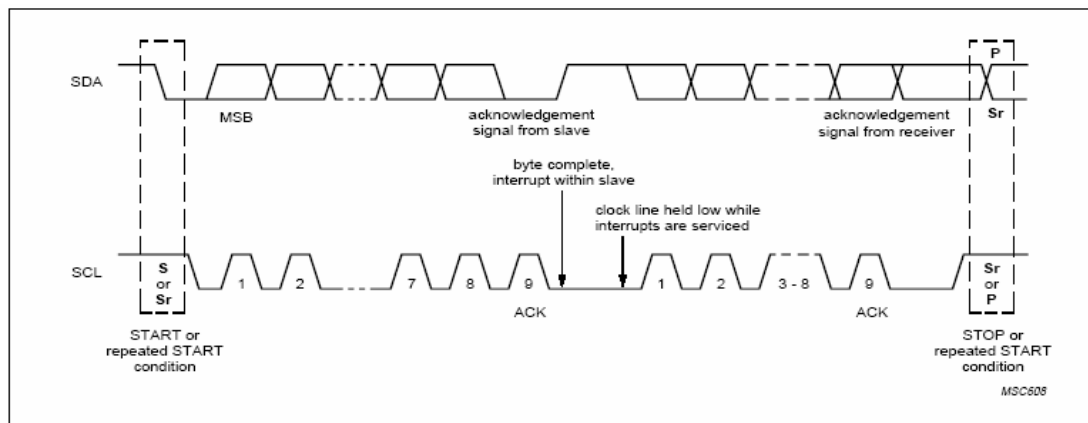


图16-2 I2C总线数据传输时序图

上图是I2C总线的时序图，分别对应主模式发送的START、ADDRESS、ACK、TxDATA、STOP等状态。

更详细的时序图可以参阅I2C总线规范。

16.3 寄存器

表 17-1 I2C 寄存器列表

地址(EPADR)	名称	字节长度	功能描述
BFH	PDI2C	1	PDI2C控制位 一见 表 16-3,写保护
D3H	I2ADR	1	I2C地址寄存器 一见 表 16-2
ABH	SCIIE	1	I2C中断使能寄存器 一见 表 16-4
B3H	SCIIF	1	I2C中断标志寄存器 一见 表 16-5
D4H	I2MCR	1	I2C控制寄存器 一见 表 16-6
D5H	I2SR	1	I2C状态寄存器 一见 表 16-8
D6H	I2DTR	1	I2C数据发送寄存器 一见 表 16-9
D7H	I2DRR	1	I2C数据接收寄存器 一见 表 16-10

寄存器说明:

表 16-2 I2C Address Register (I2ADR 0xD3H)

I2C Address Register (I2ADR)			Address: D3H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2AD6	I2AD5	I2AD4	I2AD3	I2AD2	I2AD1	I2AD0	EXTAD
Write:								
Reset:	1	0	1	0	0	0	0	0

I2AD[6:0]: (设备地址)

从模式下代表从机地址，主模式下表示呼叫地址。

EXTAD: (扩展地址)

被设置为 1 时从模式地址进行扩展。当 EXTAD=1 时，I2C 可以识别呼叫地址有：I2AD[6:0]、0000 000 以及 0001 100 这三类地址。

表 16-3 PDI2C 控制位(SUPDC 0xBFH)

System Unit Power-Down Control Register (SUPDC)			Address: BFH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	PDTPS	PDEMU	PDBOR	PDLVD	PDLCD	PDI2C	保留
Write:	x							
Reset:	0	0	1	0	0	1	1	1

PDI2C: (I2C 使能控制)

打开写保护后，PDI2C 被清为 0 时，表示使能 I2C 单元，这时 P3.0/SCL 和 P3.1/SDA 被配置为 SCL 和 SDA，作为 I2C 的 I/O 引脚。而当 PDI2C 被设置为 1 时，表示 I2C 单元被关闭，所有标志及控制位恢复到 RESET 状态。

最低位保留，复位值为 1，不建议用户往该位写 0。

表 16-4 SCI Interrupt Enable Register (SCIIE 0xABH)

SCI Interrupt Enable Register (SCIIE)			Address: ABH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	I2CIE	SPTIE	SPRIE	SPERIE	0
Write:	x	x	x					X
Reset:	0	0	0	0	0	0	0	0

I2CIE: (I2C 中断使能)

被设置为 1 时，I2TXIF、I2RXIF 和 I2NKIF 标志位被使能产生 I2C 中断。当 I2CIE 被清为 0 时，这些标志将不会产生中断给 CPU52。

表 16-5 SCI Interrupt Flag Register (SCIIF 0xB3H)

SCI Interrupt Flag Register (SCIIF)			Address: B3H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2RXIF	I2TXIF	I2NKIF	0	SPTEIF	SPRFIF	SPOVIF	SPMDIF
Write:	x	x		x	x	x	x	x
Reset:	0	0	0	0	0	0	0	0

I2RXIF: (接收中断标志)

在一个新接收的数据放置在数据接收寄存器I2DRR(0xD7H, 表 16-10)中时I2RXIF被设置为 1。一旦I2DRR存放了接收数据,其他的接收数据就不能再放置到I2DRR中,直到CPU52从I2DRR读取接收数据使得I2RXBF(0xD5H.0, 表 16-8)被清为 0。

在 I2CIE=1 时, 该位可以产生中断给 CPU52。I2RXIF 可以被软件写 0 清除或者复位清除, 或者 PDI2C 设置为 1。

I2TXIF: (发送中断标志)

数据传输寄存器I2DTR中的数据被输出完后,标志I2TXIF被置为 1,此时新的数据可以被写入到I2DTR(0xD6H, 表 16-9)中去。

在 I2CIE=1 时, 该位可以产生中断给 CPU52。I2TXIF 可以被软件写 0 清除或者复位清除, 或者 PDI2C 设置为 1。

I2NKIF: (无应答中断标志)

这位只在主模式下才能被设置为 1, 满足下面条件:

发送一个数据字节或者呼叫地址后,没有应答信号 ACK 被检测到。这个标志也清 I2AST 位。如果 I2CIE=1 时, I2NKIF 将产生中断给 CPU52。I2NKIF 可以通过软件写 0 清除或者复位清除。

表 16-6 I2C Master Control Register (I2MCR 0xD4H)

I2C Master Control Register (I2MCR)			Address: D4H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2TXAK	I2RPS	I2BB	I2AST	I2MRW	I2BR2	I2BR1	I2BR0
Write:			x					
Reset:	0	0	0	0	0	0	0	0

I2TXAK: (发送应答位使能)

被设置为 1, I2C 单元在接收到 8 个数据位后在第 9 个时钟时不发送应答位 ACK; 当 I2TXAK 被清为 0 时在第 9 个时钟时发送应答位 ACK。

I2RPS: (重复起始使能)

被设置为 1, 主模式传输时(I2AST=1)使能重复起始信号 RESTART。I2RPS 位在重复起始信号 RESTART 发送完后由硬件自动清为 0, 或者 I2AST 被清为 0 时 I2RPS 也自动清为 0。

I2BB: (总线忙标志)

检测到起始条件之后, I2BB 被设置为 1, 表示总线忙(bus busy); 在检测到停止条件时, I2BB 被清除为 0, 表示总线空闲(bus idle)。

I2AST: (主模式控制位)

这位被设置为 1 以初始化主模式传输。

在主模式下, I2C 单元在 SDA 和 SCL 上生成起始条件, 后面接着发送 I2ADR 中的呼叫地址。当前字节传输结束时, I2NKIF 被设置为 1 或者软件清除 I2AST 位时, I2C 单元生成停止条件到总线上。

I2MRW: (主模式读写位)

设置 I2AST 为 1 进入主模式时, I2MRW 作为读写 R/W 位跟随呼叫地址被发送出去。I2MRW 决定后面的数据传输方向。I2MRW=1 表示处于主接收模式; I2MRW=0, 表示主发送模式。

I2BR[2:0]: (波特率选择)

表 16-7 主模式下主时钟的频率

I2BR2	I2BR1	I2BR0	Baud Rate (fi2c=fsys)
0	0	0	fi2c/8
0	0	1	fi2c/16
0	1	0	fi2c/32
0	1	1	fi2c/64
1	0	0	fi2c/128
1	0	1	fi2c/256
1	1	0	fi2c/512
1	1	1	fi2c/1024

表 16-8 I2C Status Register (I2SR 0xD5H)

I2C Status Register (I2SR)			Address: D5H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	I2ADM	I2SRW	I2RXAK	0	I2TXBE	I2RXBF
Write:	X	X	x	x	x	x	x	x
Reset:	0	0	0	0	1	0	1	0

I2ADM: (地址匹配标志)

当 I2DRR 寄存器中的数据是一个呼叫地址，这个呼叫地址与 I2C 地址寄存器 I2ADR 设置的地址或者它的扩展地址(EXTAD=1)相匹配时，I2ADM 被设置为 1。

I2ADM=1: 表示接收地址与 I2ADR 匹配

I2ADM=0: 表示接收地址与 I2ADR 不匹配

I2SRW: (从模式读写标志)

在从模式时 I2SRW 指示数据传输的方向。在从主机接收到呼叫地址后，I2SRW 被更新。

当呼叫主机正在从 I2C 读取数据时，I2SRW 被设置为 1；当呼叫主机正在向 I2C 写入数据时，I2SRW 被设置为 0。

I2SRW=1: 从模式传输

I2SRW=0: 从模式接收

I2RXAK: (接收应答标志)

I2RXAK=0，表示往总线上发送完 8 位数据之后，接收到一个应答信号。当 I2RXAK 被设置为 1 时，表示在第 9 个时钟没有检测到应答信号；I2C 将释放 SDA 信号线，以便主机生成停止条件或者重复起始条件。

I2TXBE: (发送缓冲空)

这个标志指示数据传输寄存器 I2DTR 的状态。

当 CPU52 写数据到 I2DTR，I2TXBE 将被清除为 0。在一个字节传输完成后，I2DTR 变为空时，I2TXBE 将置为 1。

I2RXBF: (接收缓冲满)

这个标志指示数据接收寄存器 I2DRR 的状态。

当 CPU52 从 I2DRR 读取数据时，I2RXBF 将被清为 0。当从总线接收一个数据保存到 I2DRR 寄存器中时，I2RXBF 被设置为 1 表示 I2DRR 满。

表 16-9 I2C Data Transmit Register (I2DTR 0xD6H)

I2C Data Transmit Register(I2DTR)			Address: D6H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2TD7	I2TD6	I2TD5	I2TD4	I2TD3	I2TD2	I2TD1	I2TD0
Write:								
Reset:	1	1	1	1	1	1	1	1

当 I2C 单元被使能时(PDI2C=0)，被写入到这个寄存器的数据的操作，主要取决于主模式和从模式的状态。

从模式下，满足下面条件时，I2DTR 数据将被发送到输出电路：

- 检测到一个匹配的呼叫地址(I2ADM=1)，同时呼叫主机请求数据(I2SRW=1)
- 或者，前一个数据被发送完，接收主机返回的一个应答位(I2RXAK=0)

如果呼叫主机不返回一个应答位(I2RXAK=1)，I2C 单元将释放 SDA，以便主机生成一个停止条件或者重复起始条件。I2DTR 中的数据将不被发送出去，直到下一次主机呼叫。标志 I2TXBE 保留为 0 状态。

主模式下，满足下面条件时，I2DTR 数据将被发送到输出电路：

• 设置主传输模式(I2MRW=0)，并且呼叫地址已经被发送后，接收到一个应答位(I2RXAK=0)

- 或者，前一个数据被发送完，接收从机返回的一个应答位(I2RXAK=0)

如果从机不返回一个应答位(I2RXAK=1)，主机将生成停止条件或者重复起始条件。I2DTR 寄存器中的数据将不被发送到总线。输出缓冲空闲标志 I2TXBE 保留为 0。

表 16-10 I2C Data Receive Register (I2DRR 0xD7H)

I2C Data Receive Register (I2DRR)			Address: D7H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2RD7	I2RD6	I2RD5	I2RD4	I2RD3	I2RD2	I2RD1	I2RD0
Write:	x	x	x	x	x	x	x	x
Reset:	0	0	0	0	0	0	0	0

当 I2C 单元使能后(PDI2C=0)，只读寄存器 I2DRR 也取决与主模式还是从模式。

从模式下，当地址匹配标志 I2ADM=1 时，I2DRR 中数据表示从主机接收到的呼叫地址；而当 I2ADM=0 时，I2DRR 数据表示上一次接收的数据。

主模式下，表示上一次接收到的数据。

当 I2DRR 被 CPU52 读取时，接收缓冲满标志 I2RXBF 被清为 0，下一次接收到的数据将被存放到 I2DRR 中。每次新数据保存到 I2DRR 中时 I2RXIF 中断标志被置为 1，表示新数据能够从 I2DRR 中得到。

16.4 I/O 信号

I2C 的对外接口（SDA 和 SCL）采用线与方式，芯片外部管脚需要接一个上拉电阻。具体连接如下图。

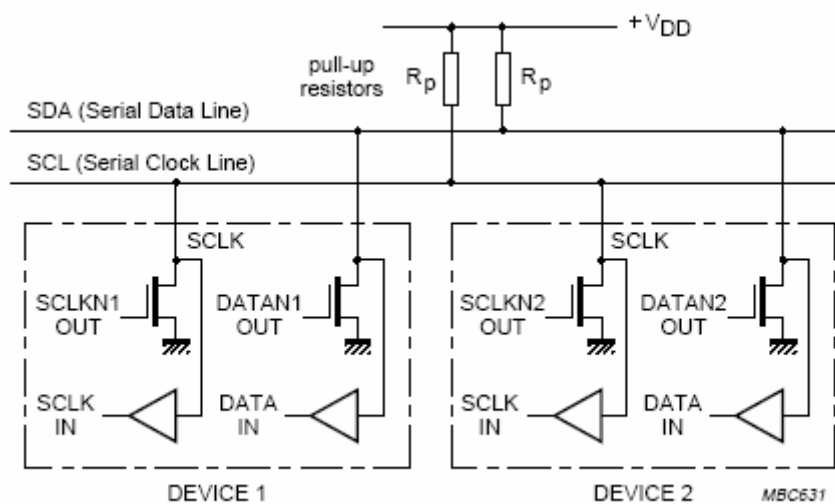


图16-3 SDA和SCL电气连接示意图

17 GPIO

17.1 概述

ATT7027 支持 35 个双向 I/O 引脚，但在电表应用中基本上配置成 LCD 输出、外部中断、UART、I2C 等功能。

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。

17.2 复用配置

表 17-1 LCD 输出复用配置寄存器 (LCDCFG 0xB9H)

LCDCFG			Address: B9H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	保留	保留	PTCH	PTCL	PTBH	PTBL	PTAH	PTAL
Write:								
Reset:	0	0	0	0	0	0	0	0

此为写保护寄存器，打开写保护后：

PTAL=0:	SEG1/PA0—SEG4/PA3 配置为标准 I/O:	PA0-PA3
PTAL=1:	SEG1/PA0—SEG4/PA3 配置为 LCD 输出:	SEG1-SEG4
PTAH=0:	SEG5/PA4—SEG8/PA7 配置为标准 I/O:	PA4-PA7
PTAH=1:	SEG5/PA4—SEG8/PA7 配置为 LCD 输出:	SEG5-SEG8
PTBL=0:	SEG9/PB0—SEG12/PB3 配置为标准 I/O:	PB0-PB3
PTBL=1:	SEG9/PB0—SEG12/PB3 配置为 LCD 输出:	SEG9-SEG12
PTBH=0:	SEG13/PB4—SEG16/PB7 配置为标准 I/O:	PB4-PB7
PTBH=1:	SEG13/PB4—SEG16/PB7 配置为 LCD 输出:	SEG13-SEG16
PTCL=0:	SEG17/PC0—SEG20/PC3 配置为标准 I/O:	PC0-PC3
PTCL=1:	SEG17/PC0—SEG20/PC3 配置为 LCD 输出:	SEG17-SEG20
PTCH=0:	SEG21/PC4—SEG24/PC7 配置为标准 I/O:	PC4-PC7
PTCH=1:	SEG21/PC4—SEG24/PC7 配置为 LCD 输出:	SEG21-SEG24
Bit[7:6]:	保留，复位值为 0，不建议用户修改该值。	

表 17-2 P0 和 P2 输出复用配置寄存器 (P02CFG 0xBAH)

P02CFG			Address: BAH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	P267	P245	P223	P201	保留	保留	保留	保留
Write:								
Reset:	0	0	0	0	0	0	0	0

此为写保护寄存器，打开写保护后：

Bit[3:0]:	保留，复位值为 0，不建议用户修改该值。	
P201=0:	TX1/P2.0—RX1/P2.1 配置为 UART1:	TX1-RX1
P201=1:	TX1/P2.0—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:	LVDIN/P2.4 配置为 LVDIN:	LVDIN
P245=1:	LVDIN/P2.4 配置为标准 I/O:	P2.4
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

表 17-3 P3 输出复用配置寄存器 (P3CFG 0xBBH)

P3CFG			Address: BBH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	保留	保留	保留	保留	INT1	INT0	IRTX1	0
Write:								X
Reset:	0	0	0	0	0	0	0	0

此为写保护寄存器，打开写保护后

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
 Bit[7:4]: 保留，默认值为 0，不建议用户修改该值

17.3 P2 口

P2 口是一个与 RX0/TX0/RX1/TX1/LVDIN/PF/QF 复用的 7 位并行端口。

表 17-4 P2 口数据寄存器(P2 0xA0H)

P2			Address: A0H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	P2.7	P2.6	保留	P2.4	P2.3	P2.2	P2.1	P2.0
Write:								
Reset:	u	u	u	u	u	u	u	u

表 17-5 P2 口方向寄存器(DDRP2 0xA3H)

DDRP2			Address: A3H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	DDRP27	DDRP26	保留	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

17.4 P3 口

P3 口是一个与 SCL/SDA/INT0/INT1 复用的 4 位并行端口。P3.0-P3.3 可以配置成直接驱动 LED 模式。

表 17-6 P3 口数据寄存器(P3 0xB0H)

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

表 17-7 P3 口方向寄存器(DDRP3 0xA4H)

DDRP3			Address: A4H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	保留	保留	保留	保留	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

表 17-8 P3 口 LED 控制寄存器(LED P3 0xA7H)

LEDP3			Address: A7H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	保留	保留	保留	保留	LEDP33	LEDP32	LEDP31	LEDP30
Write:								
Reset:	0	0	0	0	0	0	0	0

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 DDRP3x 配置成输出模式时才有效; 以上三个寄存器的 Bit[7:4] 是保留位, 不建议用户修改默认值。

17.5 Port A 口

Port A 口是一个与 LCD 的 SEG1-SEG8 复用的 8 位并行端口。

表 17-9 PTA 口数据寄存器(PTA 0x93H)

PTA			Address: 93H					
	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

表 17-10 PTA 口方向寄存器(DDRA 0x9BH)

DDRA			Address: 9BH					
	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

17.6 Port B 口

Port B 口是一个与 LCD 的 SEG9-SEG16 复用的 8 位并行端口。

表 17-11 PTB 口数据寄存器(PTB 0x94H)

PTB			Address: 94H					
	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

表 17-12 PTB 口方向寄存器(DDRB 0x9CH)

DDRB			Address: 9CH					
	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

17.7 Port C 口

Port C 口是一个与 LCD 的 SEG17-SEG24 复用的 8 位并行端口。PTC0-PTC7 可以配置成直接驱动 LED 模式。

表 17-13 PTC 口数据寄存器(PTC 0x95H)

PTC			Address: 95H					
	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

表 17-14 PTC 口方向寄存器(DDRC 0x9DH)

DDRC			Address: 9DH					
	Bit7	6	5	4	3	2	1	Bit0
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

表 17-15 PTC 口 LED 控制寄存器(LEDC 0xAEH)

LEDC		Address: AEH						
	Bit7	6	5	4	3	2	1	Bit0
Read:	LEDC7	LEDC6	LEDC5	LEDC4	LEDC3	LEDC2	LEDC1	LEDC0
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 DDRCx 配置成输出模式时才有效。

17.8 PDM 模式下 GPIO 的状态

在 PDM 下，GPIO 的状态除了 LCD 驱动口和 PF、QF 口之外，均为输入状态，内部有上拉电阻，这时 IO 口的状态不受方向寄存器的控制。RX1 也为输入状态，但其内部没有上拉电阻。LCD 驱动口如果在进入 PDM 前设置成 GPIO 功能，那么进入 PDM 模式后为输入状态，并且内部有上拉电阻。PF、QF 在 PDM 下为输出，输出为低电平。

18 PWM

18.1 概述

ATT7027 的 PWM 支持两种工作模式：Tone 生成器和脉宽调制 PWM。用户通过配置寄存器可以产生 Tone 波形以及 PWM 波形。

18.2 功能描述

PWM 支持两种工作模式：Tone 生成器和脉宽调制 PWM。

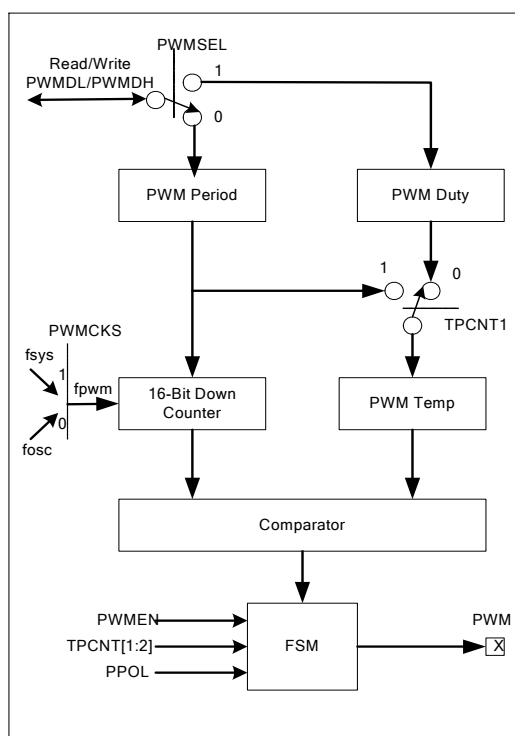


图 18-1 PWM 原理框图

工作模式选择：

TPCNT2	TPCNT1	PWMEN	Mode
X	X	0	Disable(Default)
0	0	1	PWM
0	1	1	Tone-Square
1	1	1	Tone-Staircase

PWMEN(0xE5H.7, 表 18-1)用于使能或者关闭PWM/Tone功能。当PWMEN=1 时，该模块电路被使能，可以配置为Tone生成器，也可以配置为PWM，这主要由TPCNT1(0xE5H.0, 表 18-1)决定，此时引脚PWM/TOUT被配置为PWM输出引脚。而当PWMEN=0 时，这个模块电路被完全关闭。

TPCNT1=0 时，配置为 PWM 模式，PWM 周期由 PWM Period 寄存器决定；而 PWM 的 Duty 由 PWM Duty 寄存器决定。

TPCNT1=1 时，配置为Tone模式，可以生成方波或者阶梯波。当TPCNT2(0xE5H.1, 表 18-1)=0 时，生成方波，Duty 为 50%；而TPCNT2=1 时，生成阶梯波。

18.2.1 Tone 生成器:

支持两种 Tone 模式：方波模式和阶梯波模式。

方波模式支持两个电平输出：0V 和 VCC；而阶梯波模式支持三个电平输出：0V、高阻和 VCC。在阶梯波模式下，每个半周期的后 1/4 输出为高阻。

Tone 频率公式：

$$\text{ToneFrequency} = \frac{1}{2 \times (\text{PWMPeriod}[15:0] + 1) \times T_{\text{PWM}}} = \frac{f_{\text{PWM}}}{2 \times (\text{PWMPeriod}[15:0] + 1)}$$

方波波形图：(PWMPeriod[15:0]=1)

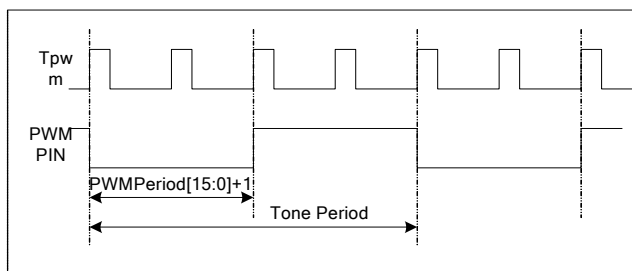


图 18-2 PWM 方波波形图

阶梯波波形图：(Period<3 时输出等 duty 的方波)

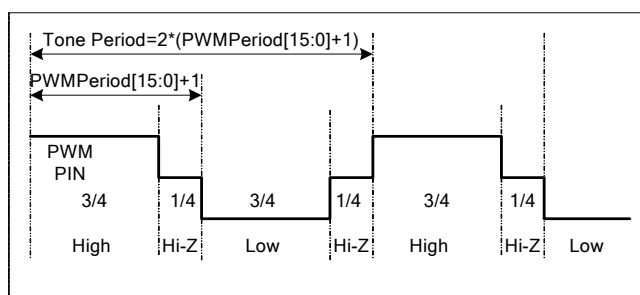


图 18-3 PWM 阶梯波波形图

18.2.2 脉宽调制 PWM

PWM Period(0xE5H.5, 表 18-1)寄存器控制PWM的周期，PWM Duty寄存器控制Duty宽度(脉宽)。可以通过控制位PPOL选择On Duty或者Off Duty。

如果 PPOL=1，选择 Off Duty；而 PPOL=0，则选择 On Duty。

PPOL	Condition	Duty Cycle
0	$0 \leq \text{Duty} < \text{Period}$	Intermediate Value
0	$\text{Duty} \geq \text{Period}$	100%(Always outputs high)
1	$0 \leq \text{Duty} < \text{Period}$	Intermediate Value
1	$\text{Duty} \geq \text{Period}$	0%(Logic '0')

当 PPOL=0 时：选择 ON Duty

$$\text{PWMFrequency} = \frac{1}{(\text{PWMPeriod}[15:0] + 1) \times T_{\text{PWM}}} = \frac{f_{\text{PWM}}}{\text{PWMPeriod}[15:0] + 1}$$

$$\text{PWM ON Period} = (\text{PWMDuty}[15:0] + 1) \times T_{\text{PWM}}$$

$$\text{Duty Cycle} = \frac{PWMDuty[15:0] + 1}{PWMPeriod[15:0] + 1}$$

而当 PPOL=1 时，选择 Off Duty

$$\text{Duty Cycle} = \frac{PWMPeriod[15:0] - PWMDuty[15:0]}{PWMPeriod[15:0] + 1}$$

PWM 波形图：（PPOL=1，PWM Period=5，PWM Duty=1）

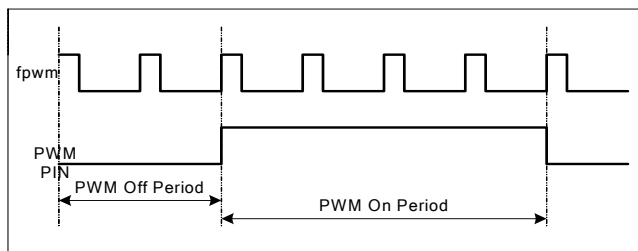


图 18-4 PWM 波形图

18.3 寄存器

表 18-1 PWM 控制寄存器 (PWMCON 0xE5H)

PWM Control Register (PWMCON)			Address: E5H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PWMEN	0	PPOL	0	PWMSEL	PWMCKS	TPCNT2	TPCNT1
Write:		X		X				
Reset:	0	0	0	0	0	0	0	0

PWMEN: PWM/Tone 使能控制

PWMEN=1: 使能 PWM/Tone 电路模块

PWMEN=0: 关闭 PWM/Tone 电路模块

PPOL: Duty 极性选择

PPOL=0: On Duty。

PPOL=1: Off Duty。

PWMSEL: PWM 寄存器选择，选择哪个寄存器被 PWMDL/PWMDH 读/写。

PWMSEL=0: 选择 PWM Period 寄存器。

PWMSEL=1: 选择 PWM Duty 寄存器。

PWMCKS: PWM/Tone 时钟选择

PWMCKS=0: 选择 $f_{pwm}=f_{osc}$

PWMCKS=1: 选择 $f_{pwm}=f_{sys}$

TPCNT[2:1]: 控制 PWM 的工作模式

TPCNT2	TPCNT1	PWMEN	Mode
X	X	0	Disable(Default)
0	0	1	PWM
0	1	1	Tone-Square
1	1	1	Tone-Staircase

表 18-2 PWM 低字节数据寄存器 (PWMDL 0xE6H)

PWM Low Data Register (PWMDL)			Address: E6H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PWMD7	PWMD6	PWMD5	PWMD4	PWMD3	PWMD2	PWMD1	PWMD0
Write:								

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

表 18-3 PWM 高字节数据寄存器 (PWMDH 0xE7H)

PWM High Data Register (PWMDH)			Address: E7H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PWMD15	PWMD14	PWMD13	PWMD12	PWMD11	PWMD10	PWMD9	PWMD8
Write:								
Reset:	0	0	0	0	0	0	0	0

PWMD[15:0]是用于读/写 PWM Period 和 PWM Duty 两个寄存器的操作寄存器。当 PWMSEL=0 时，读/写 PWMD[15:0]就是读写 PWM Period 寄存器；而 PWMSEL=1 时，则是读写 PWM Duty 寄存器。

19 编程模式

19.1 概述

ATT7027 芯片工作模式主要有：

- 用户应用模式 UAM (User Application Mode)
- 串行编程模式 SFPM (Serial Flash Programming Mode)
- 并行编程模式 PFPM (Parallel Flash Programming Mode)

工作模式通过模式选择引脚：TEST/TMS1/TMS0(内部 30K 上拉)来选择，Table 说明模式选择引脚和工作模式的对应关系：

表 19-1 编程模式选择

TEST	TMS1	TMS0	Work Mode
1	1	1	UAM 模式
1	0	1	SFPM 模式
1	1	0	PFPM 模式
1	0	0	Reserved
0	X	X	

19.2 功能描述

19.2.1 用户应用模式 UAM

这是提供给用户正常应用的模式，复位后程序从Flash的0000H地址开始运行。UAM模式下，如果使能Monitor Rom，即BROMEN=1，那么用户程序可以调用Monitor Rom中程序，以支持IAP (In-Application Programming) 和OCD(On-Chip Debug)功能。（参考[存储器映射](#)部分说明）

19.2.2 Flash 编程模式 FPM

FPM模式主要为了支持Flash的编程用，支持串行和并行编程。复位后，CPU52从0000H地址开始运行Monitor Rom程序，根据设定的编程模式，选择串行编程或者并行编程模式。

串行编程模式可以支持在系统编程ISP (In-System Programming)，利用芯片UART0进行串行编程，利用的资源最少。图19-1是串行编程模式的示意图：

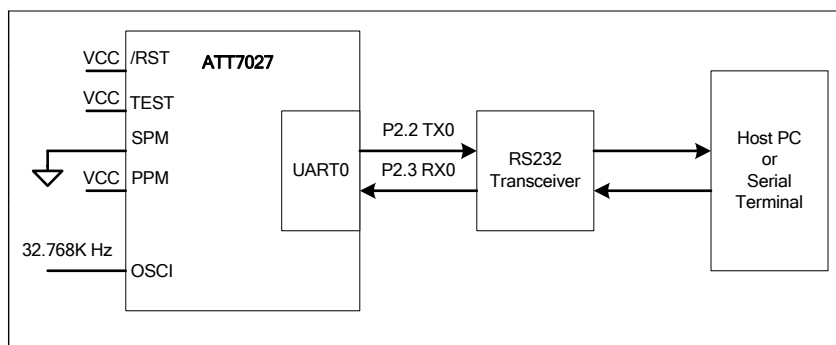


图 19-1 SPFM 连接图

并行编程模式利用PA/PB/PC以及P2.4/ P2.6/P2.7、P3.0-P3.2等30个I/O端口完成，这个模式的编程速度快，可用于量产时编程。图19-2是并行编程模式的示意图。

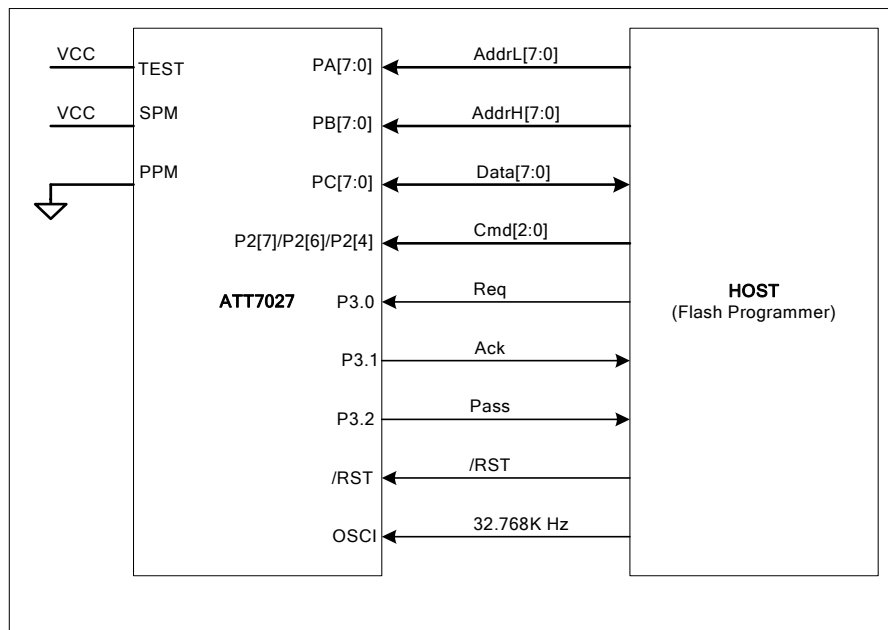


图 19-2 PPFM 连接图

19.2.3 复位后模式选择

系统复位完成后，根据不同的配置，芯片进入不同的工作模式。

主要包括两种工作模式：用户应用模式UAM(User Application Mode)和Flash编程模式FPM(Flash Programming Mode)。

UAM模式主要是提供客户用，是芯片的主要工作模式，运行Flash程序，从0000H地址开始执行。

FPM模式是供系统开发之用，提供在线编程ISP以及在线调试OCD之用。先运行ROM中的Monitor程序，从地址0000H开始执行。主要依靠引脚TEST/TMS0/TMS1进行选择(表19-1)。

NOTES: Hiterndtech为用户提供了支持OCD和IAP功能的仿真器和Flash编程器，这部分描述详见《ATT7027 仿真器使用说明》和《ATT7027 Flash Programmer 使用说明》。

20 片上 ICE 支持

20.1 概述

ATT7027的ICE支持三部分功能：

- OCD (On-Chip Debug)
- ISP (In-System Programming)
- IAP (In-Application Programming)

ICE主要通过Monitor Rom中的软件实现，但也需要部分硬件支持，ATT7027提供的片上ICE硬件支持包括：模式选择、硬件断点和Monitor Rom。

20.2 功能描述

ICE主要通过Monitor Rom中的软件实现，但也需要部分硬件支持，这些硬件支持包括：

1. 模式选择

芯片提供两种工作模式：

1) 用户应用模式UAM: User Application Mode。这是提供给客户正常应用的模式，复位后程序从Flash的0000H地址开始运行。UAM模式下，如果使能Monitor Rom，BROMEN ([表20-6](#)) =1，那么用户程序可以调用Monitor Rom中程序，以支持IAP和OCD功能。

2) Flash编程模式FPM: Flash Programming Mode。FPM模式主要为了支持Flash的编程用，支持串行和并行编程。复位后，Flash和Monitor Rom地址空间重新映射，CPU52从0000H地址开始运行Monitor Rom程序，根据设定的编程模式，选择串行编程或者并行编程模式。串行编程模式可以支持在系统编程ISP，利用芯片UART0进行串行编程，利用的资源最少。并行编程模式利用PA/PB/PC以及P2.4/ P2.6/P2.7、P3.0-P3.2等30个I/O端口完成，这个模式的编程速度快，可用于量产时编程。

当系统处于FPM模式下，SPMOD状态位 ([表20-7](#)) 指示系统选择了SFPM模式还是PFPM模式。

2. 硬件断点支持

ATT7027支持两个硬件断点，每个断点可以选择设置在PM地址或者在DM地址。断点中断IRQ_OCD的中断向量为0033H。

硬件断点的寄存器包括：HWBPL ([表20-2](#))、HWBPH ([表20-3](#))、HWBPCR ([表20-4](#)) 以及BPSEL控制位 ([表20-5](#))。

3. Monitor Rom

在FPM模式下，CPU52从0000H地址开始运行Monitor Rom，首先检测设置是串行编程模式还是并行编程模式，然后进入不同的子程序，完成对编程命令的分析和编程操作。

在UAM模式下，CPU52从Flash的0000H开始运行程序，如果允许访问Monitor Rom，那么程序调用Monitor Rom中的IAP子程序以及OCD服务程序。

IAP程序支持Flash的Write/Erase等操作。

OCD程序支持所有内部寄存器/存储器的读写操作、单步、断点设置等Debug操作。OCD程序利用UART0与外部主机进行数据交换。

注意：当需要调用Monitor Rom程序前，需先使能BOOT ROM，这是通过设置BROMEN控制位（表20-6）实现的。

20.3 寄存器

表20-1 ICE寄存器列表

地址	名称	复位值	功能描述
0xE1	HWBPL	0x00	硬件断点低位地址寄存器—见表22-2
0xE2	HWBPH	0x00	硬件断点高位地址寄存器—见表22-3
0xE3	HWBPCR	0x00	硬件断点控制寄存器—见表22-4
0xF6 MCON.7	BPSEL	0	断电选择控制位—见表22-5
0xBE FMCFG.2	BROMEN	0	BOOT ROM 使能控制位—见表22-6，写保护
0xE9 FMCON.1	SPMOD	0	FPM模式状态位—见表22-7

1. 硬件断点低位地址寄存器

表 20-2 硬件断点低位地址寄存器(0xE1H, HWBPL)

Hardware BreakPoint Low Address Register (HWBPL)			Address: E1H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	BPL7	BPL6	BPL5	BPL4	BPL3	BPL2	BPL1	BPL0
Write:								
Reset:	0	0	0	0	0	0	0	0

说明：断点低地址；数据被写入到BPSEL选择的断点的低位地址。

2. 硬件断点高位地址寄存器

表 20-3 硬件断点高位地址寄存器(0xE2H, HWBPH)

Hardware BreakPoint High Address Register (HWBPH)			Address: E2H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	BPH7	BPH6	BPH5	BPH4	BPH3	BPH2	BPH1	BPH0
Write:								
Reset:	0	0	0	0	0	0	0	0

说明：断点高地址；数据被写入到BPSEL选择的断点的高位地址。

3. 硬件断点控制寄存器

表 20-4 硬件断点控制寄存器(0xE3H, HWBPCR)

Hardware Breakpoint Control Register (HWBPCR)			Address: E3H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	0	0	0	0	0	0	PMSEL	BPEN
Write:	BPACK	X	X	x	X	X		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
HWBPCR.7	BPACK: 断点中断应答。BPACK写0，没有任何影响。BPACK写1，清除由BPSEL选择的断点中断。
HWBPCR[6:2]	保留。读为0。

HWBPCR.1	PMSEL: 断点设置选择。PMSEL=0: 表示断点设置在Program Memory地址空间。PMSEL=1: 表示断点设置在Data Memory地址空间。
HWBPCR.0	BPEN: 断点使能。BPEN=0: 表示关闭断点功能。BPEN=1: 表示开启断点功能。

4. BPSEL 控制位

表20-5 断电选择控制位(0xF6H MCON.7)

Memory Control Register (MCON)			Address: F6H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	BPSEL	0	0	0	0	0	0	RAMMAP
Write:		X	X	x	X	X	X	
Reset:	0	0	0	0	0	0	0	0

BPSEL: 断点地址选择。对BPSEL写操作时: BPSEL=0: 表示选择断点0; BPSEL=1: 表示选择断点1。对BPSEL读操作时: BPSEL表示是断点0还是断点1产生断点中断, 同时自动选择相应断点地址。BPSEL=0: 表示断点0产生中断。BPSEL=1: 表示断点1产生中断。

5. BROMEN 控制位 (写保护)

表20-6 BOOT ROM 使能控制位(0xBEH FMCFG.2)

FMCFG			Address: BEH					
	Bit7	6	5	4	3	2	1	Bit0
Read:	DFS2	DFS1	DFS0	0	0	BROMEN	PMLOCK	RSLOCK
Write:				X	X			
Reset:	0	0	0	0	0	0	0	0

BROMEN: Boot Rom使能控制; 当BROMEN=1时, 用户程序可以调用boot Rom程序。当BROMEN=0时, 用户程序不可以调用boot Rom程序。

6. SPMOD 控制位

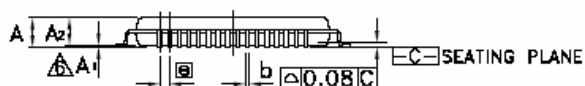
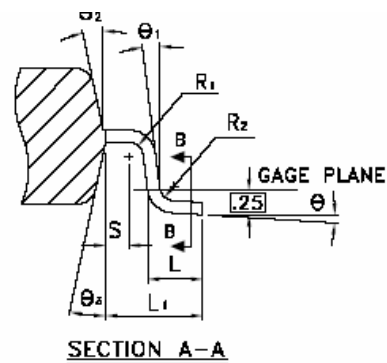
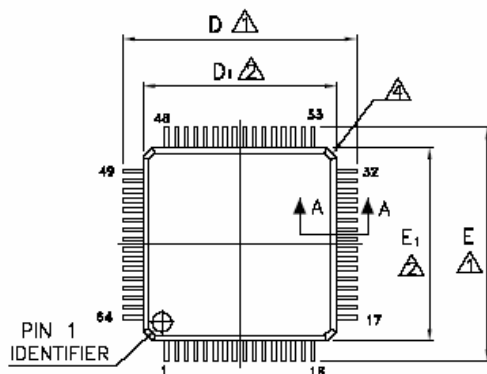
表20-7 FPM模式状态位(0xF6H MCON.7)

Flash Memory Control Register (FMCON)			Address: E9H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	PS1MAP	FOP1	FOP0	MMA	FMA	BUSY	SPMOD	0
Write:				X		X		X
Reset:	0	0	0	0	0	0	0	0

SPMOD: SPMOD=0: 指示系统选择了并行的 FPM 模式。SPMOD=1: 指示系统选择了串行的 FPM 模式。

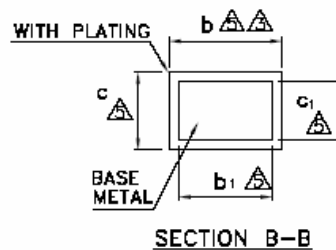
21 机械规格

ATT7027: 无铅 LQFP64



NOTE :

- △ TO BE DETERMINED AT SEATING PLANE $\square\square$.
- △ DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
D1 AND E1 ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMATCH.
- △ DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.
DAMBAR CAN NOT BE LOCATED ON THE LOWER RADIUS OF THE FOOT.
- △ EXACT SHAPE OF EACH CORNER IS OPTIONAL.
- △ THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.10 mm AND 0.25 mm FROM THE LEAD TIP.
- △ A1 IS DEFINED AS THE DISTANCE FROM THE SEATING PLANE TO THE LOWEST POINT OF THE PACKAGE BODY.
- 7. CONTROLLING DIMENSION : MILLIMETER.
- 8. REFERENCE DOCUMENT : JEDEC MS-026 , BCD.



Symbol	Dimension in mm			Dimension in inch		
	Min	Norm	Max	Min	Norm	Max
A	—	—	1.60	—	—	0.063
A ₁	0.05	—	0.15	0.002	—	0.006
A ₂	1.35	1.40	1.45	0.053	0.055	0.057
b	0.17	0.22	0.27	0.012	0.015	0.018
b ₁	0.17	0.20	0.23	0.012	0.014	0.016
c	0.09	—	0.20	0.004	—	0.008
c ₁	0.09	—	0.16	0.004	—	0.006
D	12.00 BSC			0.472 BSC		
D ₁	10.00 BSC			0.394 BSC		
E	12.00 BSC			0.472 BSC		
E ₁	10.00 BSC			0.394 BSC		
⌀	0.50 BSC			0.020 BSC		
L	0.45	0.60	0.75	0.018	0.024	0.030
L ₁	1.00 REF			0.039 REF		
R ₁	0.08	—	—	0.003	—	—
R ₂	0.08	—	0.20	0.003	—	0.008
S	0.20	—	—	0.008	—	—
θ	0°	3.5°	7°	0°	3.5°	7°
θ ₁	0°	—	—	0°	—	—
θ ₂	12°TYP			12°TYP		
θ ₃	12°TYP			12°TYP		

22 附录 A 使用超级终端下载程序

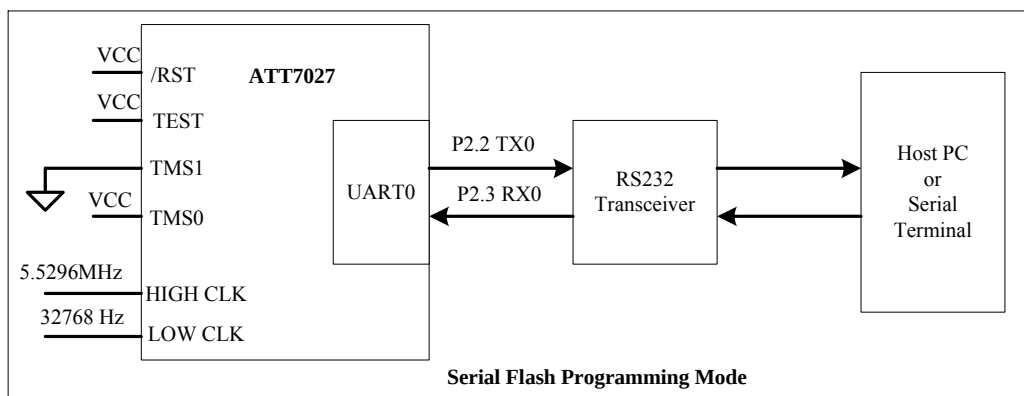
利用 UART0, 可以直接通过串口对 ATT7027 进行程序烧写操作, 不需要其他辅助电路。软件可以使用 windows 自带的超级终端调试程序, 当然也可以使用其他串口助手或者自己编写的串口调试程序。

需要注意的是, 由于是利用 UART0 进行编程、调试, 因而芯片必须有高频时钟支持 (外部必须有 1MHz 以上的晶振电路, 或者外灌 1MHz 以上的方波信号)。

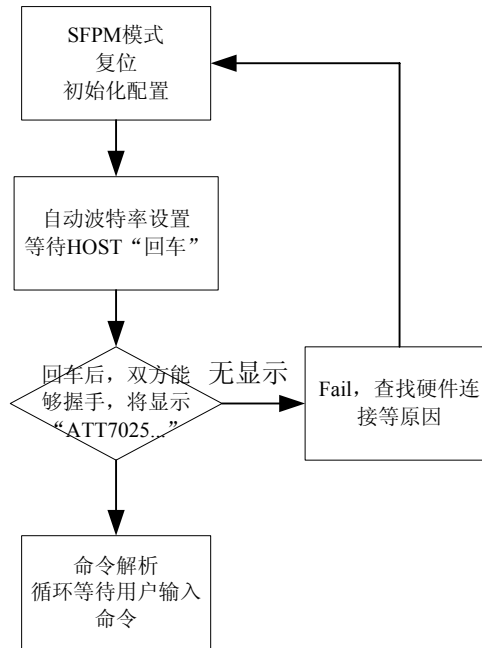
22.1 烧写程序

借助 ATT7027 的串行编程模式(SFPM), 可以很容易将生成的 hex 文件 load 到 ATT7027 的 flash 中。

串行编程模式与用户正常模式 (UAM) 的硬件设置区别为, 它的模式选择中 TMS1 由接高改为接低电平即可。



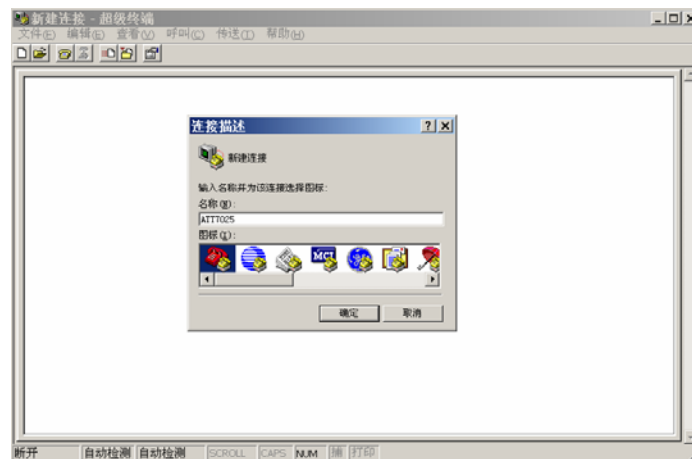
串行编程模式的流程参见下图。



将ATT7027通过RS232电平转换电路与PC连接，将模式配为SFPM，然后进行上位机操作：

第一步：设置超级中断

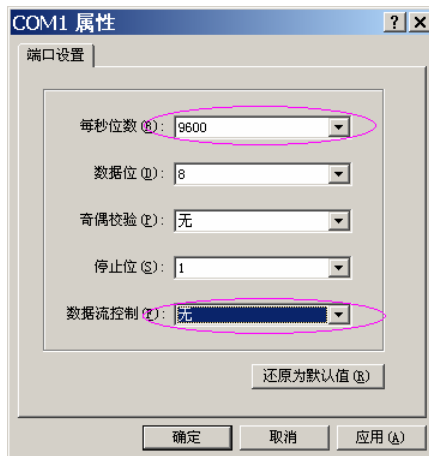
打开 所有程序->附件->通讯->超级终端，前面的一些可以随便写，到新建连接里面，输入自己的命名，比如ATT7027



按确定，选择相应的COM口，



按确定后，选择 9600、8bit、无奇偶、1bitstop，数据流控制选择“无”（默认“硬件”）

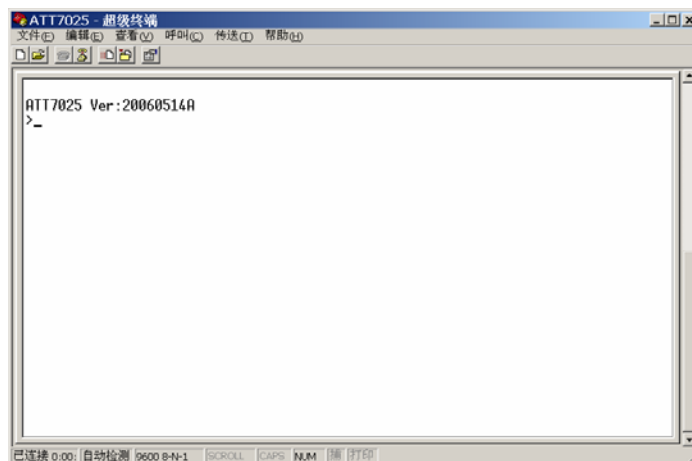


再按“确定”，超级中断配置完毕。其中波特率可以设为其他的值，比如4800、19200等。

第二步：确认握手（ATT7027进行自动波特率设置）

在上面的超级终端界面，在键盘上敲“回车”（Enter）

将会显示如下字样，’>’为命令提示符。



如果没有上面字样产生，请查找硬件连接等原因，然后重新来一次。

第三步：擦除&load hex文件。

- 1、在命令符后输入“M0000”，不用回车，等待出现“ok”
- 2、在命令符后输入“M8000”，不用回车，等待出现“ok”
- 3、在命令符后输入“E”，其目的是不回显键盘输入的字符
- 4、在键盘上输入“L”，然后选择 传送->发送文本文件，选择“*.”，然后选择你的hex文件即可
- 5、过程中可以看到“.”，表示过程正常。
- 6、最后以T结束

注：

A 步骤4中的“L”输入后，在屏幕上不会显示该字符，此时不要输入其他字符。

B 上面所有的命令必须为大写字母

C 步骤5、6过程中，不能出现’.’‘T’以外的字符，否则未烧写成功。

D 由于是与串口进行通讯，如果敲错命令，不能用'BACKSPACE'等键进行修改，可以reset或者将当次命令敲完，等待下一次提示符'>'出现，再继续操作。

第四步：转为UAM模式，运行程序

将模式改为正常模式（UAM模式），复位后，程序即从0x0000开始运行。