

AT8PC712A

**ATW 8 位 RISC
12 位 ADC 及 CMP 单片机**

规格书 **C06**
December 6, 2013

本中文规格书内容若有不清楚或是与英文 R2.0 版本有差异争议之处，请以英文版本为主

版本控制

版本	日期	描述
C01	2013-07-23	初稿
C02	2013-08-13	修改CMP Table
C03	2013-08-21	修改ADCON_3(Register 32h)
C06	2013-12-6	1. 修改ADC控制寄存器3错误 2. 增加INT2/INT3控制位元说明

目录

【1】	AT8PC712A 产品简述.....	6
1.1	产品描述	6
1.2	特性	6
1.3	系统框图	7
1.4	引脚图	7
1.5	引脚描述	8
1.6	程序存储器.....	10
1.7	数据寄存器.....	10
1.8	数据寄存器位制定	11
1.9	操作配置图.....	13
【2】	微处理器单元	14
2.1	间接寻址定义.....	14
	地址 00H: INDF (间接寻址寄存器)	14
	地址 04H: FSR (间接寻址指针寄存器)	14
2.2	8 位程序计数器	15
	地址 02H: PCL (程序计数器低字节)	15
2.3	状态寄存器.....	16
	地址 03H: STATUS (状态寄存器)	16
2.4	General Purpose I/O ports.....	17
	2.4.1 PortA, PortB 口数据寄存器	17
	地址 05H: PORTA (Port A).....	17
	地址 06H: PORTB (Port B).....	17
	2.4.2 PortA, PortB 模式.....	18
	地址 08H: PAMODE0 (Port A 输入输出控制寄存器 0)	18
	地址 09H: PAMODE1 (Port A 输入输出控制寄存器 1)	18
	地址 0AH: PBMODE0 (Port b 输入输出控制寄存器 0)	18
	地址 0BH: PBMODE1 (Port B 输入输出控制寄存器 1)	18
2.5	看门狗定时器(WDT)	20
	地址 11h: WDT (看门狗定时器)	20
2.6	定时器 0 (Timer0)	22
	地址 0FH: TMOCON (TMO 控制寄存器)	22
	地址 0EH: TMO (Timer0 计数寄存器字节).....	23
	地址 10H: TMORLD (Timer0 比较或重置值寄存器字节)	23

【3】 电源和复位管理	24
地址 12H: PCON (电源控制寄存器)	24
【4】 时钟控制	26
地址 13H: CLKCFG (时钟配置寄存器)	26
【5】 系统工作模式	28
5.1 总览	28
【6】 PWM 描述	31
6.0 PWM / 蜂鸣器(Buzzer)总览	31
6.1 PWM0 寄存器	33
地址 14H: PWMOCON (PWM0 配置寄存器)	33
地址 15H: PWMOCR (PWM0 控制寄存器)	34
地址 16H: POTMLB (PWM0 计数寄存器低字节)	34
地址 17H: POTMRLDLB (PWM0 比较或重置值寄存器低字节)	35
6.2 PWM1 寄存器	36
地址 20H: PWM1CON (PWM1 配置寄存器)	36
地址 21H: PWM1CR (PWM1 控制寄存器)	37
地址 22H: P1TM (PWM1 计数寄存器低字节)	37
地址 23H: P1TMRLD (PWM1 比较或重置值寄存器低字节)	38
6.3 Buzzer 寄存器	38
地址 24H: BZS (Buzzer 输出选择寄存器)	38
6.4 PWM 范例程序	39
【7】 7+1 通道模数转换器(ADC)	41
7.1 ADC 寄存器	41
地址 28H: ADCON_1 (ADC 控制寄存器 1)	41
地址 29H: ADCON_2 (ADC 控制寄存器 2)	42
地址 30H: ADCHB (ADC 结果寄存器高字节)	43
地址 31H: ADCLB (ADC 结果寄存器低字节)	43
地址 29H: ADCON_3 (ADC 控制寄存器 3)	43
【8】 电压比较器	45
8.1 CMP 寄存器	45
地址 35H: CMPCON1 (比较器控制寄存器 1)	45
地址 36H: DACR1HB (DACR1 高字节寄存器)	47
地址 37H: DACR1LB (DACR1 低字节寄存器)	47
地址 38H: CMPCON2 (比较器控制寄存器 2)	47
地址 39H: DACR2HB (DACR2 高字节寄存器)	48

地址 3AH: DACR2LB (DACR2 低字节寄存器).....	48
8.2 比较器迟滞应用.....	49
【9】 中断.....	50
9.1 中断寄存器.....	51
地址 0B: INTEN (中断使能寄存器).....	51
地址 0CH: INTFLAG (中断标示位寄存器).....	51
地址 0DH: INTCON (中断控制寄存器).....	52
【10】 指令集.....	53
【11】 最大绝对额定值	55
【12】 DC 特性.....	55
12.1 概要 (工作温度 = 0 to 70 °C).....	55
12.2 GPIO 接口.....	56
12.3 ADC 规格.....	57
12.4 CMP 规格.....	57
【13】 封包框图.....	58
13.1 10-PIN MSOP.....	58

【1】 AT8PC712A 产品简述

1.1 产品描述

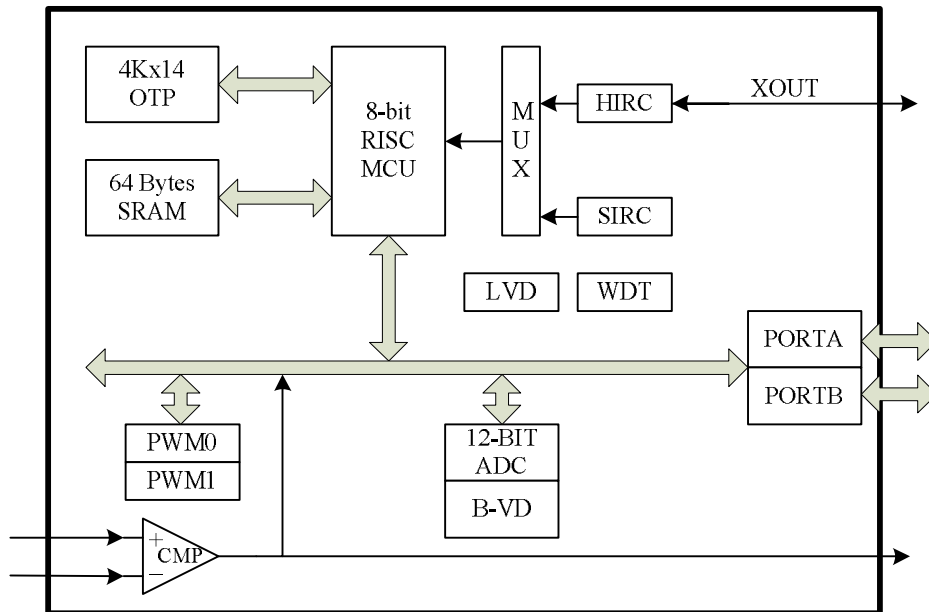
- ◆ AT8PC712A 是根据 AT8PC71A 进化延伸的优化产品。
- ◆ AT8PC712A 相较于 AT8PC71A 增加电压比较器功能。

CHIP	ROM	RAM	Clock		Stack	I/O	PWM	ADC/DAC	Timer			VREFH
			IRC	XTAL					Watch-dog	Timer-8	Timer-12/16	
AT8PC71A	1.5Kx14 OTP	64 Bytes	16 MHz	V	6	12	4(12Bi t)	5+1 ch / 1 ch	1	X	4(12Bi t)	2/3/4V
AT8PC712A	4.0Kx14 OTP	64 Bytes	16 MHz	X	6	8	1(16Bi t) 1(8Bi t)	7+1 ch / 0 ch	1	1(8Bi t)	1 (16Bi t) 1 (8Bi t)	2/3/4V

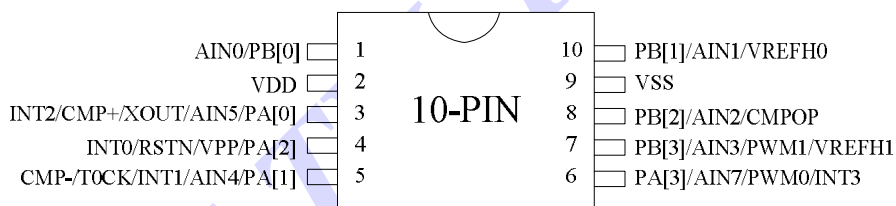
1.2 特性

- **8位RISC微处理器**
 - ◇ 每一个指令周期为一个时钟(1T)。
 - ◇ 4.0K x 14位的OTP程序存储器。
 - ◇ 6级堆栈缓存器。
 - ◇ 64 x 8位的数据存储器。
 - ◇ MCU工作频率可选择 FIRC / (2, 4, 8, 16) 的模式。
 - ◇ 一个8位计数器, 一个16位(PWM0)及一个8位(PWM1)的PWM输出
 - ◇ 看门狗定时器 (WDT)
- **系统时钟**
 - ◇ 内部振荡器
 - 内部RC (FIRC) 振荡器 = 16MHz。
 - 低频 / 绿色模式下, 内部RC振荡器 (FSIRC) = 32KHz。
- **8 个输入/输出**
 - ◇ 每个GPIO为双向I/O, 可编程为高阻抗输入、内置上拉、开源漏极模式输出或CMOS输出。
 - ◇ 每个GPIO具有较高的电流驱动能力: 15mA出力和入力电流(Normal Output)。
- **8 个中断来源**
 - ◇ 四个外部中断来源 INTO, INT1, INT2, INT3。
 - ◇ 六个内部中断来源 WDT、Timer0、PWM0、PWM1、CMP、ADC。
- **7+1 通道12位模数转换器 (12Bi t ADC)**
 - ◇ 支持连续转换模式或触发模式, 取得模数转换结果。
 - ◇ 支持7个外部模拟输入通道及1个内部1/4VDD通道。
 - ◇ 内建4组模数转换模块的参考电压(VDD, 4V, 3V, 2V)。
- **电源管理**
 - ◇ 上电复位2.4V(POR)
 - ◇ 掉电复位2.0V(PDR)
 - ◇ 三级低电压检测: LVDH(3.6V), LVDM(3.0V), LVDL(2.4V)
- **PWM 蜂鸣器(Buzzer) 功能**
 - ◇ 支持2个脉宽调制通道(PWM)
 - ◇ 可编程PWM0(Buzzer0)/PWM1(Buzzer1)的输出脚位。
- **电压比较器功能**
 - ◇ 内建低偏移量(offset)比较器。
 - ◇ 内建两组内部12位 DAC 比较电压 (DACR1/DACR2)。

1.3 系统框图



1.4 引脚图



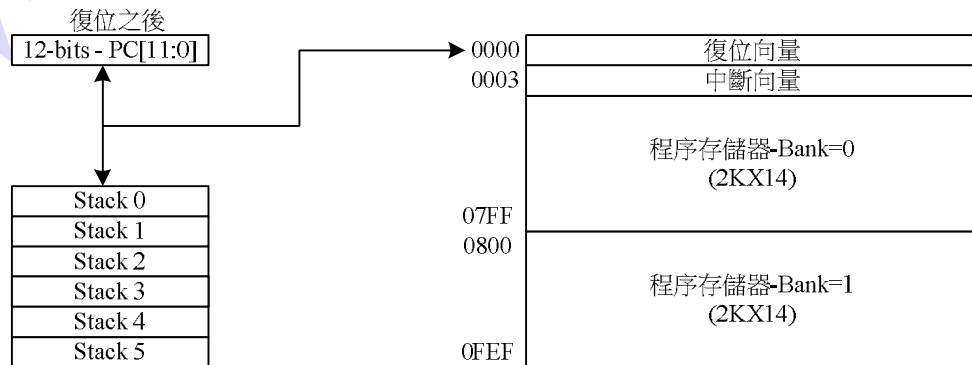
**AT8PC712A A-type
MSOP10**

1.5 引脚描述

引脚名称	类型	10-pin		说明
		P	-	
PB[0]	I/O	1	--	具双向 I/O 功能, 可编程上拉电阻, 高阻抗输入, 开源漏极输出或 CMOS 输出。输出的入力电流(Sink current) / 出力电流(Drive current) 最高可达 15mA。
AIN0	I			ADC 通道 0 输入口
VDD	P	2	--	Power Pin.
PA[0]	I/O	3	--	具双向 I/O 功能, 可编程上拉电阻, 高阻抗输入, 开源漏极输出或 CMOS 输出。输出的入力电流(Sink current) / 出力电流(Drive current) 最高可达 15mA。
AIN5	I			ADC 通道 5 输入口
CMP+	I			比较器正极端输入口
XOUT	O			HIRC 内部时钟输出口
INT2	I			外部中断 2 输入口, 下降沿或引脚变化触发(低电流)
PA[2]	I/O	4	--	具双向 I/O 功能, 可编程上拉电阻, 高阻抗输入, 开源漏极输出或 CMOS 输出。输出的入力电流(Sink current) / 出力电流(Drive current) 最高可达 15mA。
VPP	I			编程高压电源输入
RSTN	I			可当外部复位输入口, 低电平有效, 相关设置请参考 P.23(PCON)
INT0	I			外部中断 0 输入口, 下降沿或引脚变化触发(低电流)
PA[1]	I/O	5	--	具双向 I/O 功能, 可编程上拉电阻, 高阻抗输入, 开源漏极输出或 CMOS 输出。输出的入力电流(Sink current) / 出力电流(Drive current) 最高可达 15mA。
CMP-	I			比较器负极端输入口
AIN4	I			ADC 通道 4 输入口
TOCK	I			PWMO 外部时钟输入口
INT1	I			外部中断 1 输入口, 下降沿或引脚变化触发(低电流)
PA[3]	I/O	6	--	具双向 I/O 功能, 可编程上拉电阻, 高阻抗输入, 开源漏极输出或 CMOS 输出。输出的入力电流(Sink current) / 出力电流(Drive current) 最高可达 15mA。 上电为 100KΩ 上拉至 VDD, 可由软件可设置高阻状态。
AIN7	I			ADC 通道 7 输入口
PWMO	O			当 PWMO 使能时为 PWMO 输出口
INT3	I			外部中断 3 输入口, 下降沿或引脚变化触发(低电流)
PB[3]	I/O	7	--	具双向 I/O 功能, 可编程上拉电阻, 高阻抗输入, 开源漏极输出或 CMOS 输出。输出的入力电流(Sink current) / 出力电流(Drive current) 最高可达 15mA。
VREFH1	I			ADC 外部参考 1 电压输入口
AIN3	I			ADC 通道 3 输入口
PWM1	O			当 PWM1 使能时为 PWM1 输出口

引脚名称	类型	10-pin	-	说明
		P	--	
PB[2]	I/O	8	--	具双向 I/O 功能, 可编程上拉电阻, 高阻抗输入, 开源漏极输出或 CMOS 输出。输出的输入电流 (Sink current) / 出力电流 (Drive current) 最高可达 15mA。
CMPOP	O			比较器输出口
AIN2	I			ADC 通道 2 输入口
VSS	G	9	--	电源地
PB[1]	I/O	10	--	具双向 I/O 功能, 可编程上拉电阻, 高阻抗输入, 开源漏极输出或 CMOS 输出。输出的输入电流 (Sink current) / 出力电流 (Drive current) 最高可达 15mA。
AIN1	I			ADC 通道 1 输入口
VREFH0	I			ADC 外部参考 0 电压输入口

1.6 程序存储器



(地址 0FF0-0FFF 保留给配置使用)

1.7 数据寄存器

图 1.7-1 I/O 数据寄存器

地址	描述(F-plane)	地址	R-Plane
00h	INDF		
01h	--		
02h	PCL		
03h	STATUS		
04h	FSR		
05h	PORTA		
06h	PORTB		
07h	--		
08h	--	08h	* PAMODE0
09h	--	09h	* PAMODE1
0Ah	--	0Ah	* PBMODE0
0Bh	INTEN	0Bh	* PBMODE1
0Ch	INTFLAG		
0Dh	INTCON		
0Eh	TMO		
0Fh	TMOCON		
10h	TORLD		
11h	WDT		
12h	PCON		
13h	CLKCFG		
14h	PWMOCON		
15h	PWMOCR		
16h	POTMLB		
17h	POTMRDLB		
18h	POTMHB		
19h	PORDHB		
20h	PWM1CON		
21h	PWM1CR		
22h	P1TM		
23h	P1TMRLD		
24h	Buzzer		

地址	描述 (F-plane)	地址	R-Plane
25h-27h	--		
28h	ADCON_1		
29h	ADCON_2		
30h	ADCHB		
31h	ADCLB		
32h	ADCON_3		
33-34	--		
35h	CMPCON1		
36h	DACR1HB		
37h	DACR1LB		
38h	CMPCON2		
39h	DACR2HB		
3Ah	DACR2LB		
3B-3F	--		
40h 7Fh	SRAM 64 bytes		

* 执行 IOST/IOSTR 指令作存取

* IOST 指令用来将 ACC 内容写入 R-plane 寄存器, IOSTR 指令用来读取 R-plane 寄存器内容到 ACC

1.8 数据寄存器位制定

TABLE 1.7-2: 特殊功能寄存器列表

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
00h (R/W)	INDF	间接寻址寄存器 (不是实际存在的物理寄存器)							
01h (R/W)	--	--							
02h (R/W)	PCL	程序计数器 (PC) 低字节							
03h (R/W)	STATUS	--	ROMBK	FRP	/TO	/PD	Z	DC	C
04h (R/W)	FSR	间接寻址地址指针							
05h (R/W)	PORTA	--	--	--	--	PORTA[3]	PORTA[2]	PORTA[1]	PORTA[0]
06h (R/W)	PORTB	--	--	--	--	PORTB[3]	PORTB[2]	PORTB[1]	PORTB[0]
07h	--	--	--	--	--	--	--	--	--
08h	--	--	--	--	--	--	--	--	--
09h	--	--	--	--	--	--	--	--	--
0Ah	--	--	--	--	--	--	--	--	--
0Bh (R/W)	INTEN	GIE	--	--	INT3IE	INT2IE	INT1IE	INT0IE	TMOIE
0Ch (R/W)	INTFLAG	--	--	--	INT3IF	INT2IF	INT1IF	INT0IF	TMOIF
0Dh (R/W)	INTCON	SELINT1[2:0]			INT3CON	INT2CON	INT1CON	INT0CON	--
0Eh (R/W)	TMO	TMO[7:0] 8-bit real time clock/counter							
0Fh (R/W)	TMOCON	TMOEN	--	TMOPS[2:0]			TMOCKS[2:0]		
10h (R/W)	TMORLD	TMORLD[7:0]							
11h (R/W)	WDT	WDTE	WDTSL	WDTPS2	WDTPS1	WDTPS0	LVRSL	SPSEL	--
12h (R/W)	PCON	--	GRN_MD	LVDIS	RSTSL1	RSTSL0	LVDT36	LVDT24	LVDT30
13h (R/W)	CLKCFG	SELCLK2	SELCLK1	SELCLK0	CLKSW	ATSW	INCODES	EXOSEN	IRCN

This datasheet contains new product information. ATW Technology reserves the rights to modify the product specification without notice.
No liability is assumed as a result of the use of this product. No rights under any patent accompany the sales of the product.

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
14h (R/W)	PWMOCON	POINTSL	PODT2	PODT1	PODT0	POCKS1	POCKS0	POOSEL[1:0]	
15h (R/W)	PWMOCR	PWMOE	POOUTS	POTPS2	POTPS1	POTPS0	POTMEN	POTMIE	POTMIF
16h (R)	POTMLB	POTM[7:0]							
17h (R/W)	POTMRLDLB	POTMRLD[7:0]							
18h (R)	POTMHB	POTM[15:8]							
19h (R/W)	POTMRLDHB	POTMRLD[15:8]							
20h (R/W)	PWM1CON	P1INTSL	P1DT2	P1DT1	P1DT0	P1CKS1	P1CKS0	P1OSEL[1:0]	
21h (R/W)	PWM1CR	PWM1E	P1OUTS	P1TPS2	P1TPS1	P1TPS0	P1TMEN	P1TMIE	P1TMIF
22h (R/W)	P1TM	P1TM[7:0]							
23h (R/W)	P1TMRLD	P1TMRLD[7:0]							
24h (R/W)	Buzzer	BZS1	BZS0	--	--	--	--	--	--
25-27h	--	--	--	--	--	--	--	--	--
28h (R/W)	ADCON_1	ADCEN	ADCST	CHSEL[2:0]			--	ADCSR[1:0]	
29h (R/W)	ADCON_2	ADCI E	ADCI F	SVERFH	ADCNT	SELH	ADCTMS	SELVER1	SELVER0
30h (R)	ADCHB	ADCB[11]	ADCB[10]	ADCB[9]	ADCB[8]	ADCB[7]	ADCB[6]	ADCB[5]	ADCB[4]
31h (R)	ADCLB	--	--	--	--	ADCB[3]	ADCB[2]	ADCB[1]	ADCB[0]
32h (R/W)	ADCOM_3	INEN[7]	--	INEN[5]	INEN[4]	INEN[3]	INEN[2]	INEN[1]	INEN[0]
35h (R/W)	CMPCON1	CMPEN	COP1	CMP1E1	CMPRI F1	CMPFI F1	CMPMD[2:0]		
36h (R/W)	DACR1HB	DACR1[11:4]							
37h (R/W)	DACR1LB	--	--	--	--	DACR1[3:0]			
38h (R/W)	CMPCON2	CMPINT1	COP2	CMP1E2	CMPRI F2	CMPFI F2	CMPINT2	TOGSEL	--
39h (R/W)	DACR2HB	DACR2[11:4]							
3Ah (R/W)	DACR2LB	--	--	--	--	DACR2[3:0]			

注: -- = 未实现, 读取为 '0' ,

1.9 操作配置图

配置名称	描述	复位初值
SELXOUT[1:0]	选择 IRC 输出时钟	11
	SLEXOUT [1:0] 频率	
	11 Firc /1 = 16MHz	
	10 Firc /2 = 8MHz	
	01 Firc /4 = 4MHz	
00 Firc /8 = 2MHz		
Sel CpuClk[2:0]	选择 FCPU = FIRC 时钟	111
	Sel CpuClk[2:0] FCPU	
	1xx 1MHz	
	011 2MHz	
	010 4MHz	
-- --		
-- --		
INSWDY	内部时钟切换到外部时钟延迟时间 1 = 128us 延迟 0 = 4ms 延迟	1
WDTSEL	看门狗定时器溢出选择 WDTSEL[1:0] : 00 = 128 ms 01 = 512 ms 10 = 8 ms 11 = 32 ms	11
sel Por	上电复位延展时间 Sel Por[1:0]: 00 = 8 ms 01 = 16 ms 10 = 32 ms 11 = 64 ms	11
PROTECT	1: 无保护 ROM 程序码 0: 保护 ROM 程序码	1→ 无保护

【2】微处理器单元

2.1 间接寻址定义

INDF 不是一个实体的寄存器，它是根据 FSR 间接寻址指针寄存器来处里所选定的实体寄存器。当 FSR=xxh 时，读取 INDF 则会读取到 FSR 所指定寄存器的数据资料。

直接写入 INDF 是没有作用的。

FSR 寄存器的 0-6 位可选择 128 个寄存器(address : 00h-7fh)。

地址 00H: INDF (间接寻址寄存器)

位	7	6	5	4	3	2	1	0
名称	FSR 间接寻址指针所指向的内容							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

地址 04H: FSR (间接寻址指针寄存器)

位	7	6	5	4	3	2	1	0
名称	--	在间接寻址模式下用作定义地址用，只 7bit 有效						
读/写	--	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

例：如何使用间接寻址指令

(1) 写入 0xAA 数据到地址 7Fh 寄存器

```

MOV A      7Fh
MOV AR     FSR      ; 指针寄存器地址 = 7Fh
MOV A      AAh
MOV AR     INDF     ; 将数据 AAh 写入地址 = 7Fh 寄存器

```

(2) 读取地址 7Fh 寄存器到 ACC

```

MOV A      7Fh
MOV AR     FSR      ; 指针寄存器地址 = 7Fh
MOV R      INDF,A  ; 读取地址 7Fh 寄存器的数据，存入 ACC

```

2.2 8 位程序计数器

可透过 (ADDAR PCL, R) 指令的方式来建立列表, 但是 PCL 寄存器的位址最大为 FFh(256Byte), 因此须注意 (ADDAR PCL, R) 指令的执行结果, 是否会有大于 FFh 的情况发生。

地址 02H: PCL (程序计数器低字节)

位	7	6	5	4	3	2	1	0
名称	程序计数器低字节							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

例：建立列表

```

MOV R, TABLECNT, A ; 将 TABLECNT 存入 A 寄存器中
CALL TABLE1
.
.
TABLE1: ADDAR PCL, R ; PCL = PCL+TABLECNT
RET A 00h ; TABLECNT=0, 传回 00h
RET A 01H ; TABLECNT=1, 传回 01h
RET A 02H ; TABLECNT=2, 传回 02h
RET A 03H ; TABLECNT=3, 传回 03h
    
```

2.3 状态寄存器

状态寄存器的内容为逻辑运算处理单元（ALU）运算的结果及复位状态

Z, DC 或 C 位是根据逻辑运算处理单元（ALU）的结果设置为 0 或 1，并可以直接写 0 或 1。

T0, PD 位不可写。

地址 03H: STATUS (状态寄存器)

位	7	6	5	4	3	2	1	0
名称	--	ROMBK	FRP	/T0	/PD	Z	DC	C
读/写	R	R/W	R/W	R	R	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[6] : ROMBK

1 = 选择程序存储器为 Bank 1 (PC=0x800h-0xFEh)

0 = 选择程序存储器为 Bank 0 (PC=0x000h-0x7FFh)

位[5] : FRP

1: 依据外部引脚的状态(from PIN)来更新 I/O 数据资料 (PA/PB)。

0: 依据 PA/PB 寄存器(from register)资料, 来更新 I/O 状态。

位[4] : /T0

WDT 状态标示位

1 = CLRWDT 指令或 SLEEP 指令产生。

0 = 看门狗定时器溢出产生。

位[3] : /PD

电源下降状态标示位

1 = 开机复位或 CLRWDT 指令

0 = SLEEP 指令

位[2] : Z

零标示位

1 = 逻辑运算结果为零

0 = 逻辑运算结果不为零

位[1] : DC

辅助进位/借位标示位

ADDAR, ADDIA

1 = 运算结果低 4 位有进位

0 = 运算结果低 4 位没有进位

SUBAR, SUBIA

1 = 运算结果低 4 位没有借位

0 = 运算结果低 4 位有借位

位[0] : C

进位/借位标示

ADDAR, ADDIA, RLR

1 = 有进位

0 = 没有进位

SUBAR, ADDIA, RRR

1 = 没有借位

0 = 有借位

2.4 General Purpose I/O ports

Ports A 是 4 位的数据输入/输出寄存器，Ports B 是 4 位的数据输入/输出寄存器，在微处理器上每一脚位都可选择为外部中断输入口。

图 2-4.1 显示 GPIO 引脚方块。

图 2.4.3-1 每一引脚可独立配置为高阻抗输入口，内置上拉电阻输入口，开源极流输出或可选择出力电流强度输出口或入力电流输出口。

复位之后，GPIO 数据内容及控制寄存器将被清除，GPIO 为输入模式。

注意：IO 端口的电压最高不可超过 $VDD + 0.3V$ 。否则芯片容易受损。

2.4.1 PortA, PortB 口数据寄存器

地址 05H: PORTA (Port A)

位	7	6	5	4	3	2	1	0
名称	--	--	--	--	PORTA[3]	PORTA[2]	PORTA[1]	PORTA[0]
读/写	--	--	--	--	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7:0]: PORTA[3:0] 数据资料

1 = 引脚逻辑高电平

0 = 引脚逻辑低电平

地址 06H: PORTB (Port B)

位	7	6	5	4	3	2	1	0
名称	--	--	--	--	PORTB[3]	PORTB[2]	PORTB[1]	PORTB[0]
读/写	--	--	--	--	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7:0]: PORTB[3:0] 数据资料

1 = 引脚逻辑高电平

0 = 引脚逻辑低电平

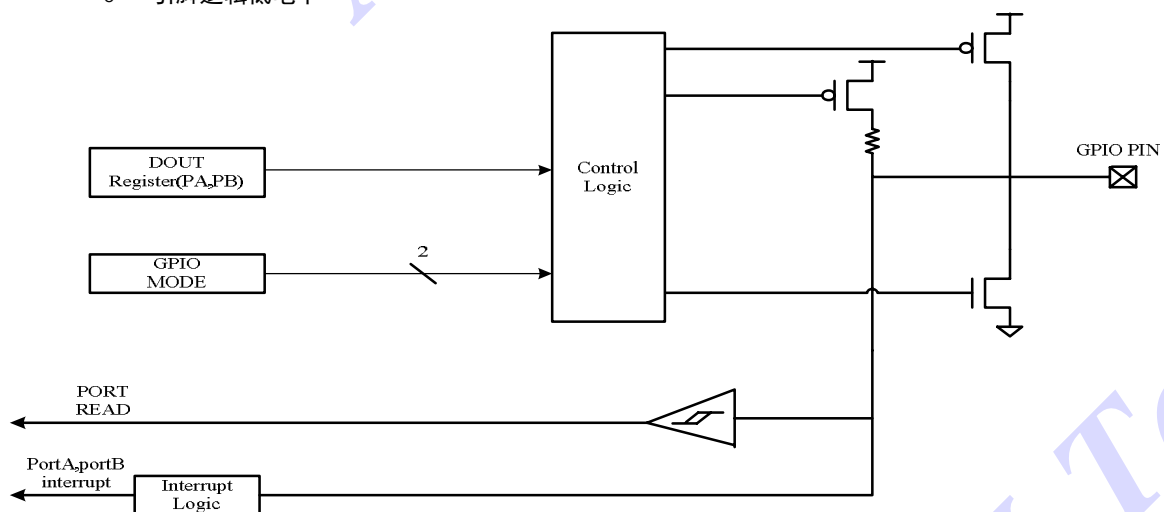


图 2.4-1 Block diagram of GPIO Port

2.4.2 PortA, PortB 模式

地址 08H: PAMODE0 (Port A 输入输出控制寄存器 0)

位	7	6	5	4	3	2	1	0
名称	--	--	--	--	PAMDO[3:0]			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	1	1	0	0

必须使用 IOST/IOSTR 指令存取

位[7:0] : PAMDO [3:0]

- 1 = PA 引脚 Mode0 为高电平
- 0 = PA 引脚 Mode0 为低电平

地址 09H: PAMODE1 (Port A 输入输出控制寄存器 1)

位	7	6	5	4	3	2	1	0
名称	--	--	--	--	PAMD1[3:0]			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

必须使用 IOST/IOSTR 指令存取

位[7:0] : PAMD1 [3:0]

- 1 = PA 引脚 Mode1 为高电平
- 0 = PA 引脚 Mode1 为低电平

Note: PA[2] : 预设状态为 100KΩ 上拉, 且可设为外部复位接脚(See P24 之位[4:3] : RSTSL[1:0])
 PA[3] : 预设状态为 100KΩ 上拉。

地址 0AH: PBMODE0 (Port b 输入输出控制寄存器 0)

位	7	6	5	4	3	2	1	0
名称	--	--	--	--	PBMD0[3:0]			
读/写	--	--	--	--	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

执行 IOST/IOSTR 指令存取

位[7:0] : PBMD0 [3:0]

- 1 = PB 引脚 Mode1 为高电平
- 0 = PB 引脚 Mode1 为低电平

地址 0BH: PBMODE1 (Port B 输入输出控制寄存器 1)

位	7	6	5	4	3	2	1	0
名称	--	--	--	--	PBMD1[3:0]			
读/写	--	--	--	--	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

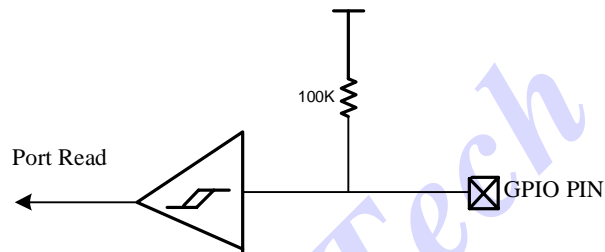
执行 IOST/IOSTR 指令存取

位[7:0] : PBMD1[3:0]

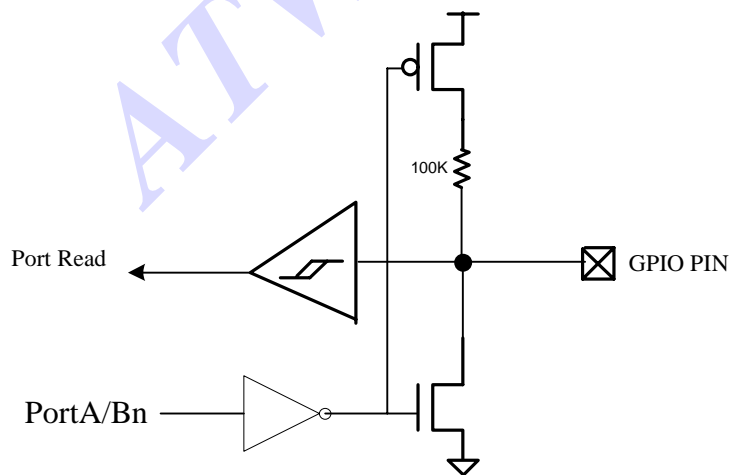
- 1 = PB 引脚 Mode1 为高电平
- 0 = PB 引脚 Mode1 为低电平

表 2.4.3-1 Port A and B 输出控制真值表

资料寄存器	模式 1	模式 0	输出驱动强度
1	0	0	高阻抗输入 (Input mode)
0			
1	0	1	上拉 (Input mode)
0			
1	1	0	上拉 (100KΩ)
0			入力 (15mA)
1	1	1	正常出力 (15mA)
0			入力 (15mA)



Mode1=0 and Mode0=1 上拉(输入模式)



Mode1=1 and Mode0=0 开源漏极模式

PS: 在此输出模式下, 当输出高电平时, 会有 100KΩ 上拉电阻。

图 2.4.3-1

2.5 看门狗定时器(WDT)

看门狗定时器(WDT)是一个不受其他电路控制的独立运行内建 RC 振荡器。在 SLEEP(休眠)模式, WDT 不会动作。在正常工作或是 Green Mode 的时候, 一旦 WDT 溢出将会导致 MCU 复位。

配置 WDTSEL[1:0]	VDD=5V~2.4V, SIR = 32KHz 看门狗定时器溢出时间
11	32 ms
10	8 ms
01	512 ms
00	128 ms

CLRWDT 指令可以清除 WDT, 使定时器重新配置, 以避免定时器溢出产生复位。

可设置 WDTE 位(0x08h Bit7)使看门狗定时器关闭。

WDT 的正常溢出周期为 32ms(无分频下), 在进入 Green Mode 时, 当 WDT 溢出时, MCU 将会被复位重置或执行下一个指令。

看门狗定时器的预分频见 表 2.5-1.

看门狗定时器方框图见 图 2.5-2.

地址 11h: WDT (看门狗定时器)

位	7	6	5	4	3	2	1	0
名称	WDTE	WDTSL	WDTPS[2:0]			LVRSL	SPSEL	--
读/写	R/W	R/W	R/W			R/W	R/W	--
复位初值	1	1	0	0	0	0	1	0

位[7] : WDTE

看门狗定时器使能位

- 1 = 使能
- 0 = 关闭

位[6] : WDTSL

看门狗溢出, MCU 动作选择位

- 1 = 如果看门狗溢出, MCU 被复位 (PC = 0x000h)
- 0 = 如果看门狗溢出, MCU 被唤醒且 PC = PC+1, 在绿色模式下有效

位[5:3] : WDTPS[2:0]

WDT 预分频位

WDTPS[2:0]	WDT 溢出时间 (配置选择为 32ms)
000	32ms * 1 = 32 ms
001	32ms * 2 = 64 ms
010	32ms * 4 = 128 ms
011	32ms * 8 = 256 ms
100	32ms * 16 = 512 ms
101	32ms * 32 = 1024 ms
110	32ms * 64 = 2048 ms
111	32ms * 128 = 4096 ms

表 2.5-1 WDT 预分频

位[2] : LVRSL

LVR 选择。

- 1 = 当 MCU 在绿色(Green)模式, LVR 讯号能唤醒 MCU 至正常模式(PC=PC+1)。
- 0 = 当 MCU 在绿色(Green)模式, LVR 讯号能复位 MCU(PC=0x000)。

位[1]: SPSEL

Slow IRC 时钟选择

1 : Slow IRC Clock 不随 VDD 变化。(在 Sleep Mode 下, 耗电低)

0 : Slow IRC Clock 随 VDD 变化

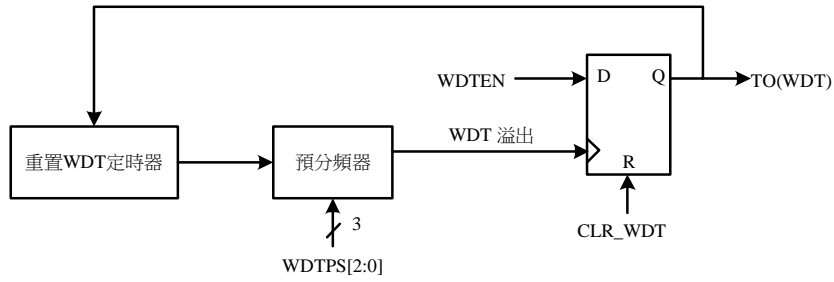


图 2.5-2 WDT 方框图

2.6 定时器 0 (Timer0)

定时器 0(Timer0)为 8 位定时器/计数器。

定时器 0 也可以当作一个中断事件发生计数器。定时器 0 方框图见图 2.6-1。

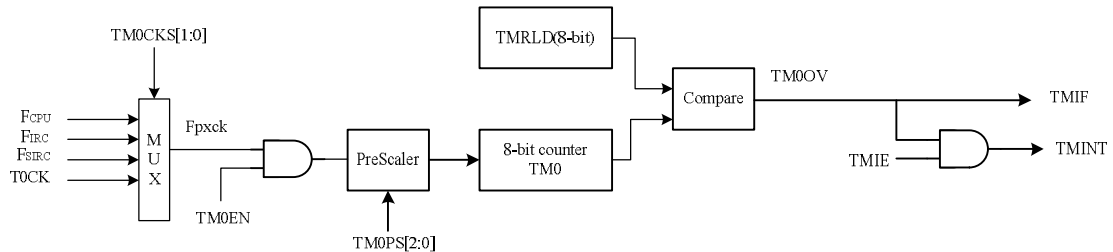


图 2.6-1 定时器 0 方框图

地址 0FH: TMOCON (TMO 控制寄存器)

位	7	6	5	4	3	2	1	0
名称	TMOEN	--	TMOPS[2:0]			TMOCKS[1:0]		--
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7] : TMOEN

TMO 模块使能位
1 = 使能
0 = 关闭

位[6] : 保留

位[5:3] : TMOPS[2:0]

TMO 定时器预分频如下

TMOPS [2:0]	TMO 模式
000	TMO 定时器时钟为 FT0ck /2
001	TMO 定时器时钟为 FT0ck /4
010	TMO 定时器时钟为 FT0ck /8
011	TMO 定时器时钟为 FT0ck /16
100	TMO 定时器时钟为 FT0ck /32
101	TMO 定时器时钟为 FT0ck /64
110	TMO 定时器时钟为 FT0ck /128
111	TMO 定时器时钟为 FT0ck /256

表 8-4 TMO 定时器时钟选择

位[2:1] : TMOCKS[1:0]

TMO 定时器时钟来源选择位

TMOCKS [1:0]	时钟来源选择 (FT0ck)
00	FCPU
01	FIRc (16Mhz)
10	Fsirc
11	TOCK(外部时钟)

地址 0EH: TMO (Timer0 计数寄存器字节)

位	7	6	5	4	3	2	1	0
名称	TMO[7:0]							
读/写	R	R	R	R	R	R	R	R
复位初值	0	0	0	0	0	0	0	0

位[7:0] : TMO[7:0]

Timer0 计数器值字节

地址 10H: TMORLD (Timer0 比较或重置值寄存器字节)

位	7	6	5	4	3	2	1	0
名称	TMORLD[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7:0] : TMORLD[7:0]

Timer0 计数对比值及溢位重置值 (Reload) 字节

【3】电源和复位管理

AT8PC712A 有五种类型的复位方式。发生复位的类型如下所示：

1. 上电(POR) / 掉电复位(PDR) (正常模式)
 - ◆ POR ($VDD \leq 2.4V$) and PDR ($VDD \leq 2.0V$)
2. 欠压复位
 - ◆ LVDT24 ($VDD \leq VLVR1 == 2.4V$ for $> 100ns$)
 - ◆ LVDT30 ($VDD \leq VLVR2 == 3.0V$ for $> 100ns$)
 - ◆ LVDT36 ($VDD \leq VLVR2 == 3.6V$ for $> 100ns$)
3. 看门狗复位
 - ◆ WDR (看门狗定时器溢出)
4. 外部复位(PA[2])
 - ◆ EXTR (低电平有效) (软件设定 RSTSL[1:0]=11, PA[2]转为外部 Reset 功能)
5. 掉电复位(PDRS) (睡眠模式下)
 - ◆ PDRS ($VDD \leq 1.3V$)

当有 POR, PDR, PDRS, WDR, LVDT24, LVDT3.0 或 EXTR 发生时, 芯片都将进入复位状态。

在复位状态下会发生以下的几种情况：

- ◆ 除状态寄存器 (03h) 外的所有寄存器复位后都还原。
- ◆ 只有在 POR, PDR 情况下复位时, 状态寄存器 (03h) 才会还原到原来的默认值。
- ◆ 在复位后, 程序计数器的开始地址位是 0x000

当 VDD 电压上升到大约 2.4V 及以上时, PORN 会被置为高电平, 此时芯片才开始复位。(参照图表 3.0-1)。

地址 12H: PCON (电源控制寄存器)

位	7	6	5	4	3	2	1	0
名称	--	GRN_MD	LVDIS	RSTSL[1:0]		LVDT36	LVDT24	LVDT30
读/写	R	R/W	R/W	R/W	R/W	R	R	R
复位初值	0	0	0	0	0	0	0	0

位[7] : 保留

位[6:5] : GRN_MD, LVDIS

{ GRN_MD, LVDIS }	描述
00	正常模式, 所有 LVD36, LVDT30, LVDT24 电路被使能 先切换 FCPU 到 FSIRC.
01	低频模式下, 所有 LVD36, LVDT30, LVDT24 电路被关闭 FCPU 使能 (FCPU = FSIRC)
10	绿色模式下, 所有 LVD36, LVDT30, LVDT24 电路被使能 FCPU 关闭
11	绿色模式下, 所有 LVD36, LVDT30, LVDT24 电路被关闭 FCPU 关闭

位[4:3] : RSTSL[1:0]*

RSTSL[1:0]	描述
00	POR, PDR, PDRS, WDR 可以复位芯片
01	POR, PDR, PDRS, WDR, LVDT3.0 ($VDD \leq 3.0V$) 可以复位芯片
10	POR, PDR, PDRS, WDR, LVDT2.4 ($VDD \leq 2.4V$) 可以复位芯片
11	POR, PDR, PDRS, EXTR(PA[2]) 可以复位芯片

*POR, PDR, PDRS 情况下复位时, RSTSL[1:0] 会还原到原来的默认值

*WDR, LVDT24, LVDT3.0 或 EXR 情况下复位时, RSTSL[1:0] 无法还原到原来的默认值

位[2] : LVDT36 (只能读取)
 低电压(3.6V)检测位
 1 = VDD 电压 > 3.6V。
 0 = VDD 电压 <= 3.6V。

位[1] : LVDT24
 低电压(2.4V)检测位
 1 = VDD 电压 > 2.4V。
 0 = VDD 电压 <= 2.4V。

位[0] : LVDT30
 低电压(3.0V)检测位
 1 = VDD 电压 > 3.0V。
 0 = VDD 电压 <= 3.0V。

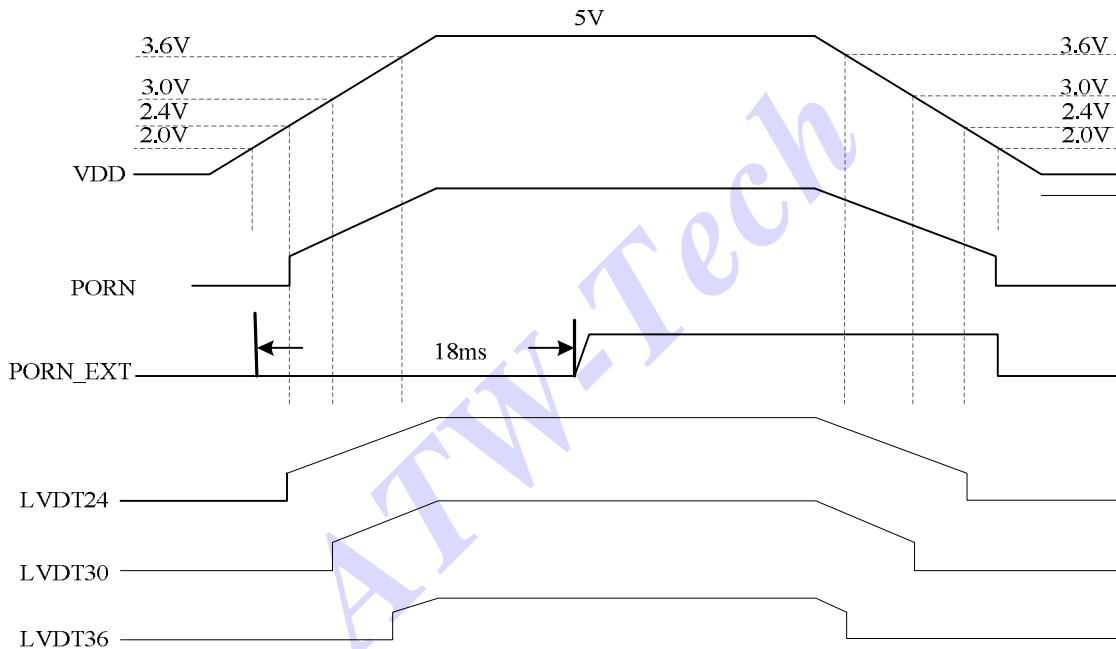


图 3.0-1 上电复位和 LVD24, LVDT30, LVDT36 时序图

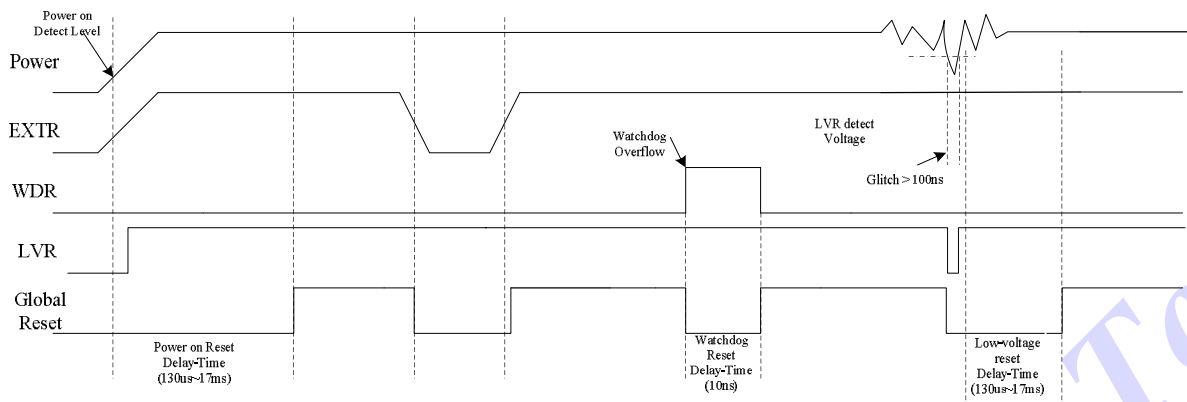


图 3.0-2 总体时序图

【4】时钟控制

MCU 工作频率范围如下图说明：(Fcpu 是 MCU 工作的时钟)

(1) 当 $VDD > 2.0V$ 时， $F_{cpu} \leq 8MHz$ (1T)

地址 13H: CLKCFG (时钟配置寄存器)

位	7	6	5	4	3	2	1	0
名称	SELCLK[2:0]			CLKSW	ATSW	INCODES	EXOSEN	IRCEN
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	1	1	0	1

位[7:5] : SELCLK[2:0]

选择 MCU 时钟 (FIRC=16Mhz)

SELCLK[2:0]	时钟选择 (Fsel) EXOSEN=0	时钟选择 (Fsel) EXOSEN=1
000	(FIRC)/4 = 4Mhz	TOCK/4
001	(FIRC)/16 = 1Mhz	TOCK/16
010	(FIRC)/8 = 2Mhz	TOCK/8
011	(FIRC)/2 = 8Mhz	TOCK/2
100	--	--
101	FSIRC	FSIRC

位[4] : CLKSW

CPU 时钟切

- 1 = CPU 时钟 (FCPU) 为 Fsel 时钟
- 0 = CPU 时钟 (FCPU) 为 Ffig 时钟

位[3] : ATSW

使能自动切换时钟。当 SELOPR=0 及 $VDD < 3.0V$ 。(复位时设定为使能状态)

- 1 = 使能自动切换 Fcpu 到 4Mhz。
- 0 = 关闭自动切换 Fcpu 到 4Mhz。

位[2] : INCODES

内部 IRC 时钟输出关闭

- 1 = 关闭内部 IRC 时钟输出。(复位时设定为禁能状态)
- 0 = 使能内部 IRC 时钟输出，内部 IRC 时钟输出到 XOUT 脚

位[1] : EXOSEN

外部震荡输入使能。(由 TOCK 输入)

- 1 = 使能外部震荡输入。Fcpu 时钟将会从 FIRC 模式切换到外部 TOCK/T1CK 模式
- 0 = 关闭外部震荡输入。

位[0] : IRCEN

内部 IRC 时钟使能。(复位时设定为使能状态)

- 1 = 使能内部 IRC 时钟
- 0 = 关闭内部 IRC 时钟

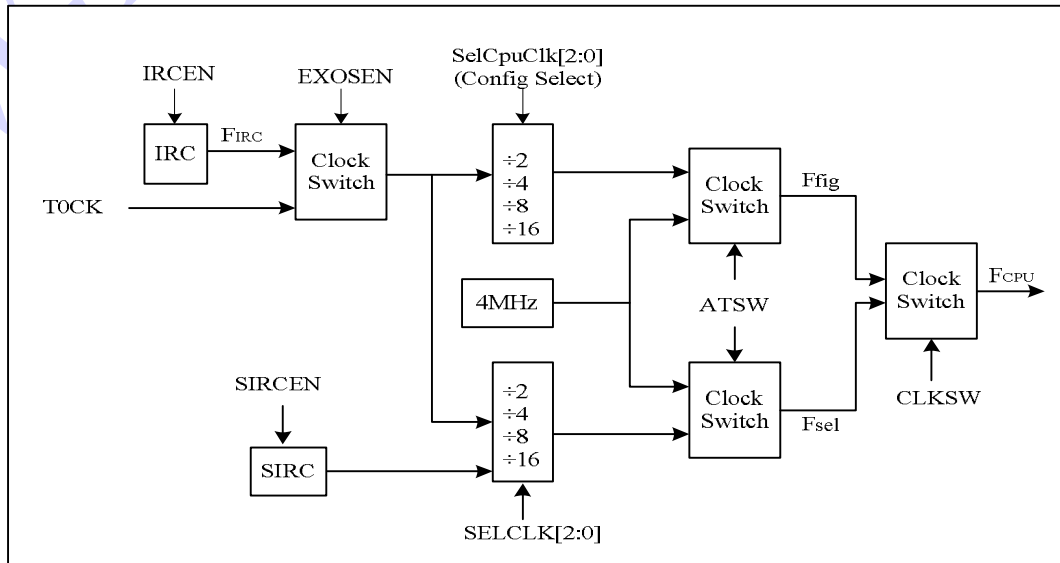


图 4.0-1 时钟切换方框图

例：时钟切换范例（当 selCpuClk == 001 时，MCU 时钟为 1Mhz）

将 MCU 时钟从 Ffig 切换为 Fsel，再切换为初始设定并切换为 8MHz

```

BCR    CLKCFG, SELCLK2_B
BCR    CLKCFG, SELCLK1_B
BSR    CLKCFG, SELCLK0_B    // 时钟频率选择为 Fsel (FIRC/16=1MHz)

BCR    CLKCFG, ATSW        // 清除 ATSW=0
BCR    CLKCFG, EXOSEN      // SET EXOSEN=0, Fcpu 切换为 Ffig

BCR    CLKCFG, SELCLK2_B
BSR    CLKCFG, SELCLK1_B
BSR    CLKCFG, SELCLK0_B    // 时钟频率选择为 Fsel (FIRC/2=8MHz)

BSR    CLKCFG, CLKSW       // Fcpu 切换为 Fsel =8Mhz (现在 Fcpu = 8Mhz)
    
```

例：时钟切换范例（当 selCpuClk == 3' b001 时 MCU 时钟为 1Mhz）

将 MCU 时钟从 Ffig 切换为 Fsel，再切换为初始设定并切换为 32KHz

```

BCR    CLKCFG, SELCLK2_B
BCR    CLKCFG, SELCLK1_B
BSR    CLKCFG, SELCLK0_B    // 时钟频率选择为 Fsel (FIRC/16=1MHz)

BCR    CLKCFG, ATSW        // 清除 ATSW=0
BCR    CLKCFG, EXOSEN      // SET EXOSEN=0, Fcpu 切换为 Ffig

BSR    CLKCFG, SELCLK2_B
BCR    CLKCFG, SELCLK1_B
BSR    CLKCFG, SELCLK0_B    // 时钟频率选择为 Fsel (FSIRC)

BSR    CLKCFG, CLKSW       // Fcpu 切换为 FSIRC 32KHz(现在 Fcpu = 32KHz)
    
```

【5】系统工作模式

5.1 总览

AT8PC712A 支持多种的低功耗的操作模式如下：

- (1) 正常模式
- (2) 低频模式
- (3) 绿色模式
- (4) 休眠模式

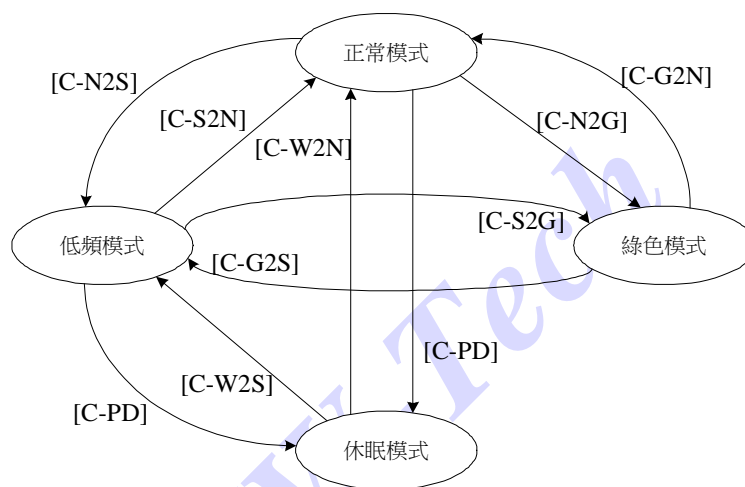


图 5.1-1 系统工作模式转换方框图

工作模式切换描述:

[C-PD]：正常模式或低频模式进入休眠模式情况

例：

```

Sleep          // Sleep 指令
NOP            // 务必要加上
NOP            // 务必要加上

```

[C-W2N]：唤醒休眠模式进入正常模式情况

- (1) INTO, INT1 中断唤醒
- (2) 外部复位发生

注：若从正常模式进入休眠模式，则当中断唤醒发生时系统将由休眠模式进入正常模式。

[C-W2S]：唤醒休眠模式进入低频模式情况

- (1) IN0, INT1 中断唤醒
- (2) LVR 或外部复位发生

注：若从低频模式进入休眠模式，则当中断唤醒发生时系统将由休眠模式进入低频模式。

[C-N2S]：正常模式进入低频模式情况

首先，软件需将 CPU 时钟从 FIRCF/FOSC 切换为 FSIRC。接着设置 LVDIS=1 (Reg. 12h.5)。

例:

```

BSR    CLKCFG, SELCLK2_B    // 设置 Fsel = FSIRC (SELCLK=3' b101)
BCR    CLKCFG, SELCLK1_B
BSR    CLKCFG, SELCLK0_B
BSR    CLKCFG, CLKSW_B      // 切换 CPU 时钟
NOP

BSR    PCON, LVDIS_B

```

[C-S2N]: 低频模式进入正常模式情况

软件将 CPU 时钟从 FSIRC 切换为 FIRC/FOSC。接着设置 LVDIS=0 (Reg. 12h. 5)。

例:

```

BCR    PCON, LVDIS_B
BCR    CLKCFG, CLKSW_B      // 切换 CPU 时钟

```

[C-N2G]: 正常模式进入绿色模式情况

绿色模式有两种类型。两者之间的差异在于由无关闭 LVD36、LVD24、LVD20 及内部高速 IRC。

范例程序为使能所有低电压检测及内部高速 IRC:

```

BSR    PCON, GRN_MD_B      // 设置 GRN_MD =1
NOP
NOP

```

范例程序为关闭所有低电压检测及内部高速 IRC:

```

MOVIA  0b01100000          // GREEN_MD 及 LVDIS 位同时间设置为 1
MOVAR  PCON
NOP
NOP

```

[C-G2N]: 绿色模式进入正常模式情况

- (1) PA, PB 中断唤醒
- (2) LVR 或外部复位发生
- (3) WDT 溢出中断
- (4) Timer 定时器溢出中断

[C-S2G]: 低频模式进入绿色模式情况

低频模式只能进入关闭低电压检测的绿色模式, 因为在低频模式下低电压检测是被关闭的, 设置 GREEN_MD (Reg. 12h. 6)位为 1 进入绿色模式

例:

```

BSR    PCON, GRN_MD_B
NOP
NOP

```

[C-G2S]: 绿色模式进入低频模式情况

- (1) PA, PB 中断唤醒
 - (2) LVR 或外部复位发生
 - (3) WDT 溢出中断
 - (4) Timer 定时器溢出中断
- 注:** 若系统从低频模式进入绿色模式, 则当唤醒发生时系统从绿色模式进入低频模式

工作模式描述

模式 Module	正常模式	低频模式	绿色模式	休眠模式 (掉电模式)
FOSC	运行 (EXOSEN = 1)	关闭或使能 (根据 EXOSEN)	关闭 (EXOSEN = 0 or 1)	关闭 (EXOSEN = 0 or 1)
FIRC	运行 (IRCEN = 1)	关闭或使能 (by IRCEN)	Disable (IRCEN = 0 or 1)	关闭 (IRCEN = 0 or 1)
FSIRC	运行 (SIRCEN = 1)	运行 (SIRCEN = 1)	运行 (SIRCEN = 1)	关闭 (SIRCEN = 0)
WDT	运行 (WDTE = 1, WDTSL=0)	运行 (WDTE = 1, WDTSL=0)	运行 (WDTE = 1, WDTSL=1)	关闭 (WDTE = 0, WDTSL=0)
内部中断	全部使能	全部使能	全部使能	全部使能
外部中断	全部使能	全部使能	全部使能	全部使能
唤醒事件	--	--	INT0, INT1 中断 Timer 定时器, WDT 定时器溢出, 复位	INT0, INT1 中断 复位
Sleep 指令	Non	non	non	执行
{Green, LVDIS} (In Reg-12h)	00	01	10 or 11	xx
MCU 程序计数器	运行	运行	停止	停止
MCU 时钟	FOSC or FIRC	FSIRC	没有时钟	没有时钟

【6】PWM 描述

6.0 PWM / 蜂鸣器(Buzzer)总览

PWM0 是 16 位脉宽控制模块。

PWM1 是 8 位脉宽控制模块。

AT8PC712A 支持两个通道 PWM0/PWM1 输出。PWM 输出时序如下 图 6-1。

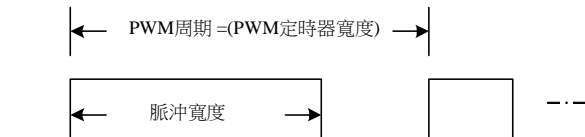


图 6-1 PWM 时序

PWM 周期宽度(Period):

$$\text{Period} = (2^{\text{Timer-bits}}) * (1/\text{FPXTPS})$$

PWM 脉冲宽度(Duty):

$$\text{脉冲宽度} = \text{PxTMRLD} * (1/\text{FPXTPS})$$

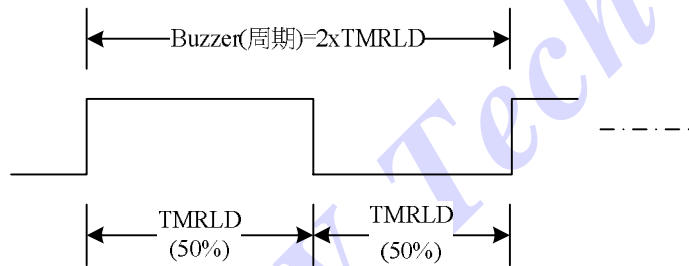


图 6-2 Buzzer 时序

$$\text{Buzzer(周期)} = 2 * (\text{PxTMRLD} + 1)$$

周期时间计算:

(A) PWM 模式下: (PWMXEN=1, PXTMEN=1, BZX=0)

$$\text{Period} = (2^{\text{Timer-bits}}) * (1/\text{FPXTPS})$$

例:

PxDT[2:0] = 000 (选择 8 位计数器)

PxCKS[2:0] = 001 (选择频率来源 Fpxck = FIRC = 16MHz)

PxTPS[2:0] = 001 (选择 PWM 频率 FPXTPS = Fpxck / 2)

PxTMRLD = 0fh = 15

$$\text{PWM 周期} = (2^3) * ((1/(FIRC / 2)) = 256 * (1/(16\text{MHz}/2)) = 32\mu\text{s}$$

$$\text{PWM 脉冲宽度} = 15 * ((1/(FIRC / 2)) = 15 * (1/(16\text{MHz}/2)) = 1.8765\mu\text{s}$$

(B) Timer 模式下: (PWMXEN=0, PXTMEN=1, BZX=0):

$$\text{周期} = \text{PxTMRLD} * (1/\text{FPXTPS})$$

例:

Fcpu = 1MHz

PxODT[2:0] = 101 (选择 10 位计数器),

PxCKS[2:0] = 000 (Fpxck = Fcpu = 1MHz)

PxTPS[2:0] = 000 (FPXTPS = Fpxck / 1)

PxTMRLD = 3E8 h

$$\text{Timer 溢出周期} = 3E8h * (1/\text{FPXTPS}) = 1000 * (1/1\text{MHz}) = 1000 * 1\mu\text{s} = 1\text{ms}$$

(C) Buzzer 模式下: (PWMXEN=0, PXTMEN=1, BZX=1)

$$\text{周期} = 2 * (\text{PxTMRD} + 1) * (1/\text{TPxTPS})$$

例:
 Fcpu = 1MHz
 PxDT[2:0] = 101 (选择 10 位计数器)
 PxCKS[2:0] = 000 (Fpxck = Fcpu = 1MHz)
 PxTPS[2:0] = 000 (FPxTPS = Fpxck/1)
 PxTMRD = 3E7h (11 1110 0011B(10Bit))

$$\text{Buzzer 周期} = 2 * (3E7h + 1) * (1/\text{TPxTPS}) = 2 * (1000) * (1/1\text{MHz}) = 2000 * 1\mu\text{s} = 2\text{ms} (0.5\text{KHz})$$

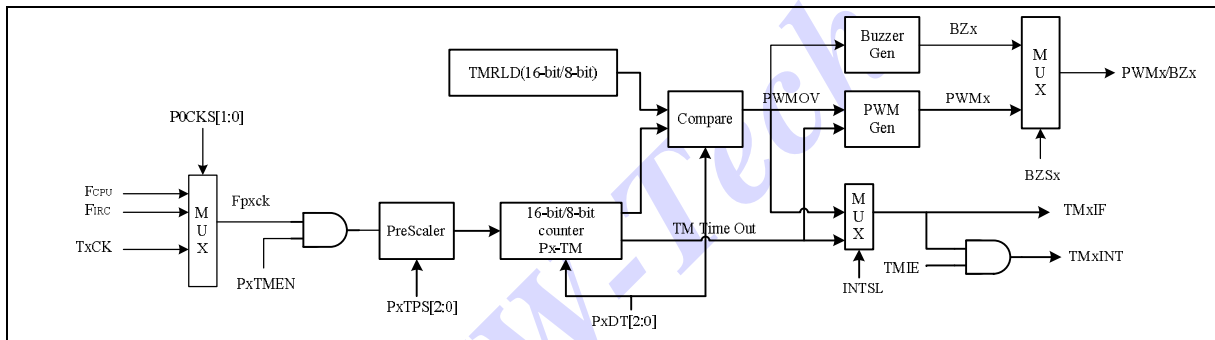


图 6-3 PWM0/ Buzzer0/ PWM1/ Buzzer1 方框图

6.1 PWM0 寄存器

地址 14H: PWMOCON (PWM0 配置寄存器)

位	7	6	5	4	3:2		1-0	
名称	POINTSL	PODT[2:0]			POCKS[1:0]		POOSEL[1:0]	
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7] : POINTSL

PWM0 中断选择

- 1 = PWM0 脉冲宽度(Duty)溢出产生中断
- 0 = PWM0 周期宽度(Period)溢出产生中断

位[6:4] : PODT[2:0]

PWM0 周期范围

表 6-3 PWM0 周期范围

PODT[2:0]	PWM0 周期范围		PWM0 定时器位
	高电平脉冲	低电平脉冲	
000	1~256	256~1	8 位
001	1~512	512~1	9 位
010	1~1024	1024~1	10 位
011	1~2048	2048~1	11 位
100	1~4096	4096~1	12 位
101	1~16384	16384~1	14 位
110	1~32768	32768~1	15 位
111	1~65536	65536~1	16 位

位[3:2] : POCKS[1:0]

PWM0 定时器时钟选择

POCKS[2:0]	时钟选择(Fp0ck)
00	FCPU
01	FIRC (16Mhz)
10	--
11	TOCK(外部时钟)

位[1:0] : POOSEL[1:0]

PWM0 / 蜂鸣器 0 输出选择

POOSE[2:0]	输出引脚选择
00	PA[3]
01	PA[0]
10	PA[1]
--	--

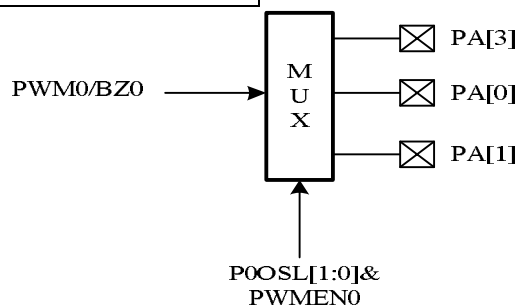


图 6-4 PWM0 / BZ0 引脚选择

地址 15H: PWMOCR (PWM 控制寄存器)

位	7	6	5	4	3	2	1	0
名称	PWMOEN	POOUTS	POTPS[2:0]			POTMEN	POTMIE	POTMIF
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7] : PWMOEN

PWM 模块使能
 1 = PWM 工作模式
 0 = Timer 工作模式

位[6] : POOUTS

PWM 脉冲宽度输出选择
 1 = PWM 脉冲宽度(Duty)时间内输出低电平
 0 = PWM 脉冲宽度(Duty)时间内输出高电平

位[5:3] : POTPS[2:0]

PWM 定时器预分频如下

POCKSL [2:0]	PWM 模式 (FPOTPS)
000	PWM 定时器时钟为 Fp0ck
001	PWM 定时器时钟为 Fp0ck /2
010	PWM 定时器时钟为 Fp0ck /4
011	PWM 定时器时钟为 Fp0ck /8
100	PWM 定时器时钟为 Fp0ck /16
101	PWM 定时器时钟为 Fp0ck /32
110	PWM 定时器时钟为 Fp0ck /64
111	PWM 定时器时钟为 Fp0ck /128

表 8-4 PWM 定时器时钟选择

位[2] : POTMEN

PWM 定时器使能位
 1 = 使能
 0 = 关闭

位[1] : POTMIE

PWM 定时器溢出中断使能位
 1 = 中断使能
 0 = 中断关闭

位[0] : POTMIF

PWM 定时器溢出位
 1 = PWM 定时器发生溢出, 写 0 清除此位
 0 = PWM 定时器未发生溢出

地址 16H: POTMLB (PWM 计数寄存器低字节)

位	7	6	5	4	3	2	1	0
名称	POTM[7:0]							
读/写	R	R	R	R	R	R	R	R
复位初值	0	0	0	0	0	0	0	0

位[7:0] : POTM[7:0]

PWM 计数器低 8 位字节

地址 17H: POTMRDLB (PWM 比较或重置值寄存器低字节)

位	7	6	5	4	3	2	1	0
名称	POTMRDL[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7:0] : POTMRDL[7:0]

PWM 计数对比值或溢位重置值(ReIoad)低 8 位字节

地址 18H: POTMHB (PWM 计数寄存器高字节)

位	7	6	5	4	3	2	1	0
名称	POTM [15:8]							
读/写	R	R	R	R	R	R	R	R
复位初值	0	0	0	0	0	0	0	0

位[7:4] : POTM[11:8]

PWM 计数器高 8 位字节

地址 19H: POTMRDLHB (PWM 比较或重置值寄存器高字节)

位	7	6	5	4	3	2	1	0
名称	POTMRDL [15:8]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7:0] : POTMRDL[15:8]

PWM 计数对比值或溢位重置值(ReIoad)高 8 位字节

6.2 PWM1 寄存器

地址 20H: PWM1CON (PWM1 配置寄存器)

位	7	6	5	4	3	2	1	0
名称	P1INTSL	P1DT[2:0]			P1CKS[1:0]		P10SEL[1:0]	
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7] : POINTSL

PWM1 中断选择

- 1 = PWM1 脉冲宽度(Duty)溢出产生中断
- 0 = PWM1 周期宽度(Period)溢出产生中断

位[6:4] : P1DT[2:0]

PWM1 周期范围

P1DT[1:0]	PWM1 周期范围		PWM1 定时器位
	高电平脉冲	低电平脉冲	
000	1~256	256~1	8 位
001	1~128	128~1	7 位
010	1~64	64~1	6 位
011	1~32	32~1	5 位
100	--	--	--
101	--	--	--
110	--	--	--
111	--	--	--

表 6-3 PWM1 周期范围

位[3:2] : P1CKS[1:0]

PWM1 定时器时钟选择

P1CKS[2:0]	时钟选择 (Fp1ck)
00	FCPU
01	FIRC(16Mhz)
10	--
11	TOCK(外部时钟)

位[1:0] : P10SEL[1:0]

PWM1/ Buzzer1 输出选择

P10SE [2:0]	输出引脚选择
00	PB[3]
01	PB[0]
10	PB[1]
11	PB[2]

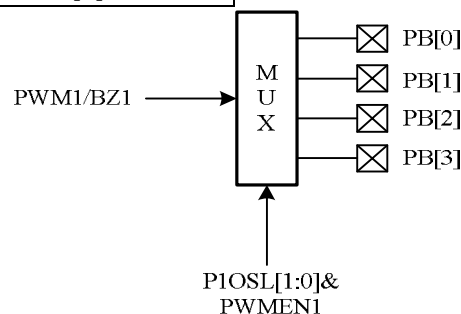


图 6-4 PWM1/ BZ1 引脚选择

地址 21H: PWM1CR (PWM1 控制寄存器)

位	7	6	5	4	3	2	1	0
名称	PWM1EN	P10UTS	P1TPS[2:0]			P1TMEN	P1TMIE	P1TMIF
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7] : PWM1EN

PWM1 模块使能
 1 = PWM 工作模式
 0 = Timer 工作模式

位[6] : P00UTS

PWM1 脉冲宽度输出选择
 1 = PWM1 脉冲宽度(Duty)时间内输出低电平
 0 = PWM1 脉冲宽度(Duty)时间内输出高电平

位[5:3] : P1TPS[2:0]

PWM1 定时器预分频如下

P1CKSL [2:0]	PWM1 模式 (Tp1TPS)
000	PWM1 定时器时钟为 Fp1ck
001	PWM1 定时器时钟为 Fp1ck /2
010	PWM1 定时器时钟为 Fp1ck /4
011	PWM1 定时器时钟为 Fp1ck /8
100	PWM1 定时器时钟为 Fp1ck /16
101	PWM1 定时器时钟为 Fp1ck /32
110	PWM1 定时器时钟为 Fp1ck /64
111	PWM1 定时器时钟为 Fp1ck /128

表 6-5 PWM0 定时器时钟选择

位[2] : P1TMEN

PWM1 定时器使能位
 1 = 使能
 0 = 关闭

位[1] : P1TMIE

PWM1 定时器溢出中断使能位
 1 = 中断使能
 0 = 中断关闭

位[0] : P1TMIF

PWM1 定时器溢出位
 1 = PWM1 定时器发生溢出, 写 0 清除此位
 0 = PWM1 定时器未发生溢出

地址 22H: P1TM (PWM1 计数寄存器低字节)

位	7	6	5	4	3	2	1	0
名称	P1TM[7:0]							
读/写	R	R	R	R	R	R	R	R
复位初值	0	0	0	0	0	0	0	0

位[7:0] : P1TM[7:0]

PWM1 计数器 8 位字节

地址 23H: P1TMRD (PWM1 比较或重置值寄存器低字节)

位	7	6	5	4	3	2	1	0
名称	P1TMRD[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7:0] : P1TMRD[7:0]

PWM1 计数比或溢位重置值(Reload)8 位字节

6.3 Buzzer 寄存器

地址 24H: BZS (Buzzer 输出选择寄存器)

位	7	6	5	4	3	2	1	0
名称	BZS1	BZS0	--					
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7] : BZS1

Buzzer 1 输出选择

1: 始能 buzzer1 输出到 PWM1 引脚(当 PWMEN1=1)

0: 关闭 buzzer1 输出

位[6] : BZS0

Buzzer 0 输出选择

1: 始能 buzzer0 输出到 PWM0 引脚 (当 PWMEN0=1)

0: 关闭 buzzer0 输出

6.4 PWM 范例程序

// 如何设置 PWMx 时钟(FpXck)来源

// 假设 Fcpu=4MHz , Nx=0

```
BCR      PWMxCON, PxCKS1_B      // Fpxck = Fcpu = 4MHz = 4MHz
BCR      PWMxCON, PxCKS0_B
```

// 设置 PWM 分频

```
BSR      PWMxCR, PxTPS2_B      // FpXTPS = 4MHz/32 = 125 KHz ;
BCR      PWMxCR, PxTPS1_B
BSR      PWMxCR, PxTPS0_B
```

(A) 8 位 PWM

// 假设 PWMx 周期宽度范围 8 位(范围 00h~ffh)

```
BCR      PWMxCON, PxDT2_B      // 设置 PWMx 周期位= 8Bit = 0~255
BCR      PWMxCON, PxDT1_B      // PWMx 周期 = 256 * (1/FpXTPS)
BCR      PWMxCON, PxDT0_B
```

// 设定 PWMx 脉冲宽度

```
MOVI A   0fh                  // 设置低字节 7-0 位初始值 (只能填入 00h~FFh)
MOVAR    PxTMRDLB             // 脉冲宽度 = 0fh(15) x (1/FpXTPS)
```

// 使能 PWMx

```
BSR      PWMxCR, PxTMI E_B     // PWMx 中断功能使能
BSR      PWMxCR, PWMxEN_B     // 选择为 PWMx 功能
BSR      PWMxCR, PxTMEN_B     // PWMx 使能
```

(B) 16 位 PWM

// 设定 PWMx 周期宽度范围 16 位(范围 0000h~ffffh)

```
BSR      PWMxCON, PxDT2_B     // 设置 PWMx 周期位= 16Bit = 0~65535
BSR      PWMxCON, PxDT1_B     // PWMx 周期 = 65536 * (1/FpXTPS)
BSR      PWMxCON, PxDT0_B
```

// 设定 PWMx 脉冲宽度

```
MOVI A   00h                  // 设置低 8 位字节初始值
MOVAR    PxTMRDLB             //
MOVI A   080h                 // 设置高 8 位字节初始值
MOVAR    PxRDHB               // 脉冲宽度 = 0800h(2048) x (1/FpXTPS)
```

// 使能 PWMx

```
BSR      PWMxCR, PxTMI E_B     // PWMx 中断功能使能
BSR      PWMxCR, PWMxEN_B     // 选择为 PWMx 功能
BSR      PWMxCR, PxTMEN_B     // PWMx 使能
```

(C) 8 位 PWM 切换至 16 位 PWM

```
// PWMx 周期范围 8 位 (范围 00h~ffh)
BCR      PWMxCON, PxDT2_B      // 设置 PWMx 周期位= 8Bit = 0~255
BCR      PWMxCON, PxDT1_B      // PWMx 周期 = 256 * (1/FpXTPS)
BCR      PWMxCON, PxDT0_B

MOVIA    0fh                    // 设置低字节 7~0 位初始值
MOVAR    PxTMRDLB              // 脉冲宽度 = 0fh(15) x (1/FpXTPS)

// 使能 PWMx
BSR      PWMxCR, PxTMI_E_B      // PWMx 中断功能使能
BSR      PWMxCR, PWMxEN_B       // 选择为 PWMx 功能
BSR      PWMxCR, PxTMEN_B       // PWMx 使能

// 切换到 PWMx 周期范围 16 位 (0000h~FFFFh)
MOVIA    0fh                    // 设置低 8 字节 [7~0] 位初始值
MOVAR    PxTMRDLB              //
MOVIA    00h                    // 设置高 8 字节 [15~8] 位初始值
MOVAR    PxTMRDLHB            // 脉冲宽度 = 000fh(15) x (1/FpXTPS)

BSR      PWMxCON, PxDT2_B       // 设置 PWMx 周期位=16 Bit = 0~65535
BSR      PWMxCON, PxDT1_B
BSR      PWMxCON, PxDT0_B       // PWMx 周期 = 65536 * (1/FpXTPS)
```

(D) 12 位 PWM 更新重置值

```
// PWM 周期范围=12 位 (范围 1~4096)
BSR      PWMxCON, PxDT2_B       // 设置 PWMx 周期位= 12Bit = 0~4095
BCR      PWMxCON, PxDT1_B       // PWMx 周期 = 4096 * (1/FpXTPS)
BCR      PWMxCON, PxDT0_B

// 载入重置值
MOVIA    007h                  // 设置低字节初始值
MOVAR    PxTMRDLB              //
MOVIA    10h                    // 设置高字节初始值
MOVAR    PxTMRDLHB

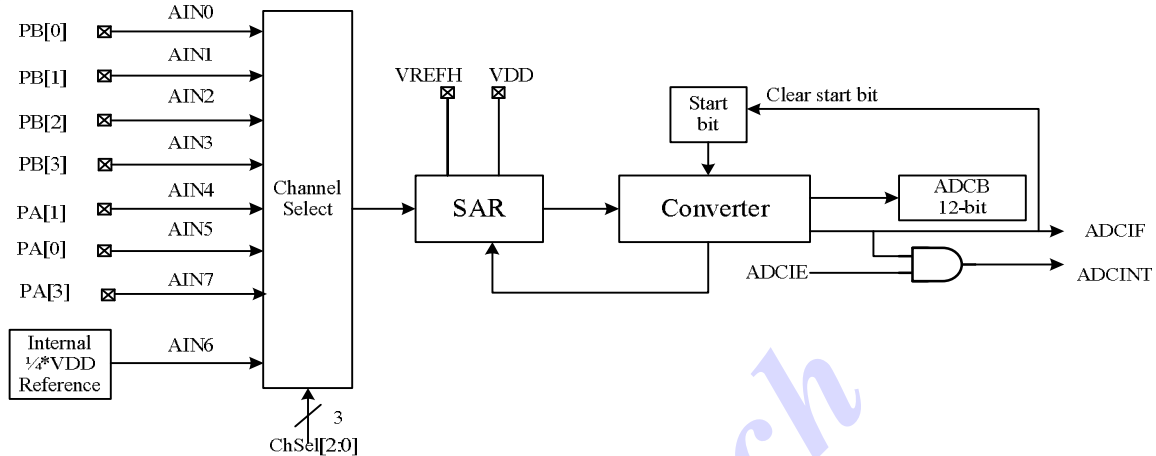
// PWM 使能
BSR      PWMxCR, PxTMI_E_B      // PWMx 中断功能使能
BSR      PWMxCR, PWMxEN_B       // 选择为 PWMx 功能
BSR      PWMxCR, PxTMEN_B       // PWMx 使能

// 更新重置值
MOVIA    02Ch                  // 设置低字节初始值
MOVAR    PORDLB                //
MOVIA    10h                    // 设置高字节初始值
MOVAR    PORDHB                // 脉冲宽度由 1007h 切换至 102Ch
```


【7】 7+1 通道模数转换器 (ADC)

ADC 模块支持 7 个模拟通道 (PB[0]-PB[3], PA[0], PA[1], PA[3]) 及 一个内部 $\frac{1}{4} VDD$ 参考通道 (可用作芯片电源电力监控)。

ADC 模块转换一个输入电压为 12 位数字数据



7.1 ADC 寄存器

地址 28H: ADCON_1 (ADC 控制寄存器 1)

位	7	6	5	4	3	2	1	0
名称	ADCEN	ADCST	CHSEL[2:0]			--	ADCSR[1:0]	
读/写	R/W	R/W	R/W	R/W	R/W	--	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7] : ADCEN

ADC 模块使能位

1 = ADC 使能

0 = ADC 关闭

位[6] : ADCST

1 = 开始模数转换, 当 ADC 转换结束, 此位被硬件清除为 0

0 = 停止模数转换。

位[5:3] : CHSEL[2:0]

ADC 输入通道选择位如下:

CHSEL[2:0]	输入通道
000	PB[0] (AIN0)
001	PB[1] (AIN1)
010	PB[2] (AIN2)
011	PB[3] (AIN3)
100	PA[1] (AIN4)
101	PA[0] (AIN5)
110	内部 $VDD/4$ (AIN6)
111	PA[3] (AIN7)

表 9-1 ADC 输入通道选择

位[2]: 保留。

位[1:0] : ADCSR[1:0]

ADC 时钟选择 (F_{adc} 最高不可超过 2Mhz)
 ADC 时钟分频选择 (F_{ADC})

ADCSR[1:0]	F _{ADC}
00	F _{ADCSEL} / 8
01	F _{ADCSEL} / 4
10	F _{ADCSEL} / 2
11	F _{ADCSEL} / 1

地址 29H: ADCON_2 (ADC 控制寄存器 2)

位	7	6	5	4	3	2	1	0
名称	ADCI E	ADCI F	SVREFH	ADCNT	SELH	ADCTMS	SELVER[1:0]	
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7] : ADCI E

ADC 中断使能位
 1 = 使能
 0 = 关闭

位[6] : ADCI F

ADC 转换完成中断标示位
 1 = ADC 转换完成, 写 0 清除此标示位
 0 = ADC 转换未完成或是完成。

位[5] : SVREFH

选择 VREFH 从外部或内部电压
 1 = 外部电压 (VREFH == PB[1] 或 PB[3] 引脚, 跟据 SELH 位[3]决定)
 0 = 内部电压 (VREFH == 内部参考电压 表 9-2)。

位[4] : ADCNT

ADC 样本模式
 1 = ADC 连续模式
 0 = ADC 触发模式, 触发根据 ADCST 位

位[3] : SELH

外部 VREFH 选择
 1 = 选择外部 VREFH 为 PB[1] 引脚
 0 = 选择外部 VREFH 为 PB[3] 引脚

位[2] : ADCTMS

ADC 转换时序
 1 = 固定转换时序
 0 = 非固定转换时序

位[1:0] : SELVER[1:0]

ADC 内部 VREFH 电压选择

SELVER[1:0]	VREFH
00	VDD
01	4V
10	3V
11	2V

表 9-2 ADC 内部 VREFH 电压选择

地址 30H: ADCHB (ADC 结果寄存器高字节)

位	7	6	5	4	3	2	1	0
名称	ADCB[11:4]							
读/写	R	R	R	R	R	R	R	R
复位初值	0	0	0	0	0	0	0	0

位[7:0] : ADCB[11:4] 只可读取

当 ADCEN=1, 为 ADCB[11:0] 中的高 8 位元字节

地址 31H: ADCLB (ADC 结果寄存器低字节)

位	7	6	5	4	3	2	1	0
名称	--				ADCB[3:0]			
读/写	R	R	R	R	R	R	R	R
复位初值	0	0	0	0	0	0	0	0

位[3:0] : ADCB[3:0] 只可读取

当 ADCEN=1, 为 ADCB[11:0] 中的低 4 位元字节

地址 29H: ADCON_3 (ADC 控制寄存器 3)

位	7	6	5	4	3	2	1	0
名称	INEN[7]	--	INEN[5]	INEN[4]	INEN[3]	INEN[2]	INEN[1]	INEN[0]
读/写	R/W	--	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7] : INEN[7]

PA[3] 输入电路始能 (AIN7).

1 = 关闭。(当 AIN7 为模拟讯号, 在进入 Sleep 模式时, 设定为关闭可省功耗)

0 = 使能

位[6] : 保留

位[5] : INEN[5]

PA[0] 输入电路始能 (AIN5).

1 = 关闭。(当 AIN7 为模拟讯号, 在进入 Sleep 模式时, 设定为关闭可省功耗)

0 = 使能

位[4] : INEN[4]

PA[1] 输入电路始能 (AIN4)

1 = 关闭。(当 AIN7 为模拟讯号, 在进入 Sleep 模式时, 设定为关闭可省功耗)

0 = 使能

位[3] : INEN[3]

PB[3] 输入电路始能 (AIN3)

1 = 关闭。(当 AIN3 为模拟讯号, 在进入 Sleep 模式时, 设定为关闭可省功耗)

0 = 使能

位[2] : INEN[2]

PB[2] 输入电路始能 (AIN2)

1 = 关闭。(当 AIN2 为模拟讯号, 在进入 Sleep 模式时, 设定为关闭可省功耗)

0 = 使能

位[1] : INEN[2]

PB[1] 输入电路始能 (AIN1)

1 = 关闭。(当 AIN1 为模拟讯号, 在进入 Sleep 模式时, 设定为关闭可省功耗)

0 = 使能

位[0] : INEN[0]

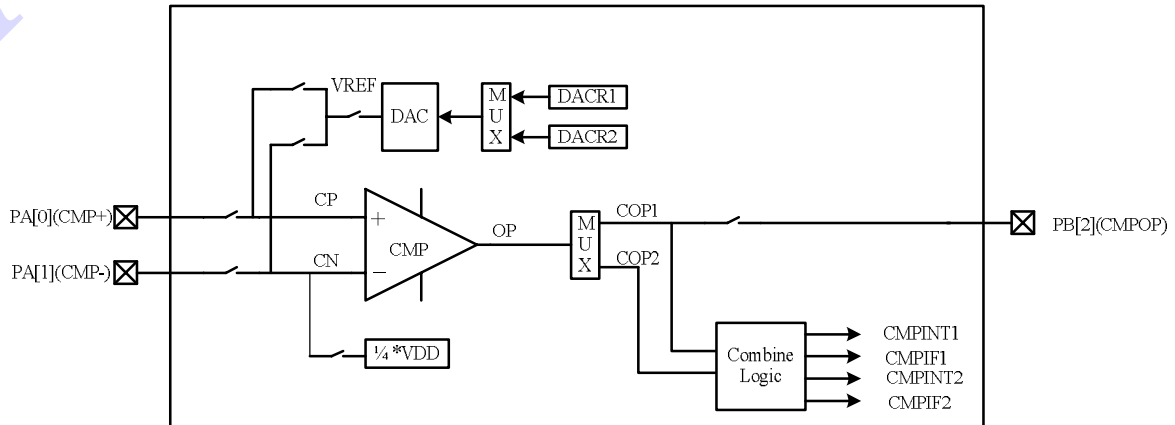
PB[0] 输入电路始能 (AIN0).

1 = 关闭。(当 AIN0 为模拟讯号, 在进入 Sleep 模式时, 设定关闭可省功耗)

0 = 使能

【8】电压比较器

AT8PC712A 内建一个低输入补偿(Low input offset) 的电压比较器。
方框图如下：



8.1 CMP 寄存器

地址 35H: CMPCON1 (比较器控制暨寄存器 1)

位	7	6	5	4	3	2	1	0
名称	CMPEN	COP1	CMPIE1	CMPRI F1	CMPFI F1	CMPMD[2:0]		
读/写	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7] : CMPEN

比较器 始能/ 关闭
1 = 始能
0 = 关闭

位[6] : COP1

比较器以 DACR1 为比对基准的输出讯号。
1 = CP > CN
0 = CP < CN

位[5] : CMPIE

比较器中断使能位 (以 DACR1 为比对基准)
1 = 使能比较器中断
0 = 关闭比较器中断

位[4] : CMPRI F1

CMPRI F 为比较器的中断标示位 (以 DACR1 为比对基准), 写 0 清除此标示位。
1 = 当比较器的输出讯号上升时, 标示位发生 (CP>CN)。
0 = 无标示位发生。

位[3] : CMPFI F1

CMPFI F 为比较器的中断标示位 (以 DACR1 为比对基准), 写 0 清除此标示位。
1 = 当比较器的输出讯号下降时, 标示位发生 (CP<CN)。
0 = 无标示位发生。

位[2:0]: CMPMD[2:0]

比较器操作模式如下

CMPEN	CMPMD[2:0]	描述			
		PA[0]	PA[1]	PB[2]	功能描述
0	XXX	GPI0	GPI0	GPI0	关闭比较器
1	111	CP	CN	COP	始能比较器 CP = PA[0] CN = PA[1] COP1 = PB[2]
1	101	CP	GPI0	COP	始能比较器 CP = PA[0] CN = DAC(VREF) COP1 = PB[2] (操作模式见 表 8-2)
1	011	GPI0	CN	COP	始能比较器 CP = DAC(VREF) CN = PA[1] COP1 = PB[2] (操作模式见 表 8-2)
1	110	CP	CN	GPI0	始能比较器 CP = PA[0] CN = PA[1]
1	100	CP	GPI0	GPI0	始能比较器 CP = PA[0] CN = DAC(VREF) (操作模式见 表 8-2)
1	010	GPI0	GPI0	GPI0	始能比较器 CP = DAC(VREF) CN = 1/4VDD (操作模式见 表 8-2)

表 8-1

ADCEN	TOGSEL	功能描述
0	0	CMP 的 DAC-1(DACR1) 单一操作模式 TMD-0
0	1	CMP 的 DAC-1(DACR1) 及 DAC-2(DACR2)交换操作模式 TMD-1
1	0	ADC 及 CMP 的 DAC-1(DACR1) 交换操作模式 TMD-2
1	1	ADC 及 CMP 的 DAC-1(DACR1) 及 DAC-2(DACR2) 交换操作模式 TMD-3

表 8-2 DACR 操作模式

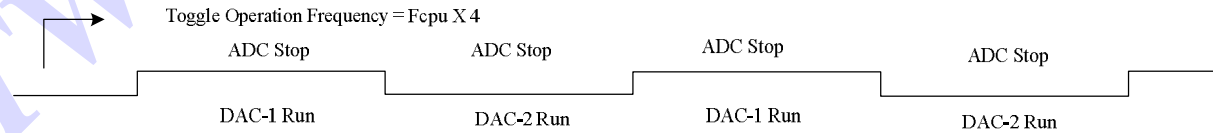
PS:

DACR1 可单独操作, 选择 TMD-0 & TMD-2。

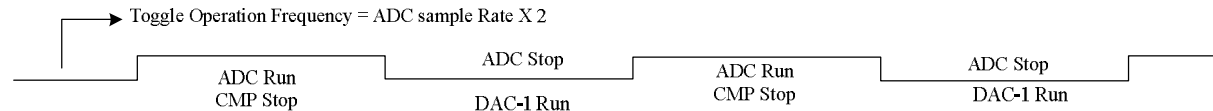
DACR2 不可单独操作, 需要与 DACR1 一块交互操作。请选择 TMD-1 & TMD-3。

有关比较器的使用, 可参考 C712A Application 的说明。

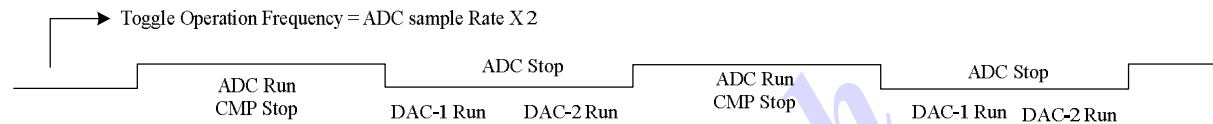
TMD-1 操作模式：



TMD-2 操作模式：



TMD-3 操作模式：



地址 36H: DACR1HB (DACR1 高字节寄存器)

位	7	6	5	4	3	2	1	0
名称	DACR1[11:4]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7:0] : DACR1[11:4]

DACR1 高字 8 位字节数据

地址 37H: DACR1LB (DACR1 低字节寄存器)

位	7	6	5	4	3	2	1	0
名称	--	--	--	--	DACR1[3:0]			
读/写	R	R	R	R	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[3:0] : DACR1[3:0]

DACR1 低 4 位字节数据

地址 38H: CMPCON2 (比较器控制寄存器 2)

位	7	6	5	4	3	2	1	0
名称	CMPINT1	COP2	CMPIE2	CMPRI F2	CMPFI F2	CMPINT2	TOGSEL	--
读/写	R/W	R	R/W	R/W	R/W	R/W	R/W	--
复位初值	0	0	0	0	0	0	0	0

位[7] : CMPINT1

1 = CMPINT1 状态变化中断

0 = CMPINT1 下降沿中断

位[6] : COP2

比较器以 DACR2 为比对基准的输出讯号。
 1 = CP > CN
 0 = CP < CN

位[5] : CMPIE2

比较器中断使能位 (以 DACR2 为比对基准)
 1 = 使能比较器 DACR2 的中断
 0 = 关闭比较器 DACR2 的中断

位[4] : CMPRIF2

CMPRIF2 为比较器的中断标示位 (以 DACR2 为比对基准), 写 0 清除此标示位。
 1 = 当比较器的输出讯号上升时, 标示位发生 (CP>CN)。
 0 = 无标示位发生。

位[3] : CMPFIF2

CMPFIF2 为比较器的中断标示位 (以 DACR2 为比对基准), 写 0 清除此标示位
 1 = 当比较器的输出讯号下降时, 标示位发生 (CP<CN)。
 0 = 无标示位发生。

位[2] : CMPINT2

1 = CMPINT2 状态变化中断
 0 = CMPINT2 下降沿中断

位[1] : TOGSEL

ADC 及 CMP 交换操作选择. (见 表 8-2)

位[0]: 保留

地址 39H: DACR2HB (DACR2 高字节寄存器)

位	7	6	5	4	3	2	1	0
名称	DACR2[11:4]							
读/写	R/W	R /W	R/W	R /W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7:0] : DACR2[11:4]

DACR2 高 8 位字节数据

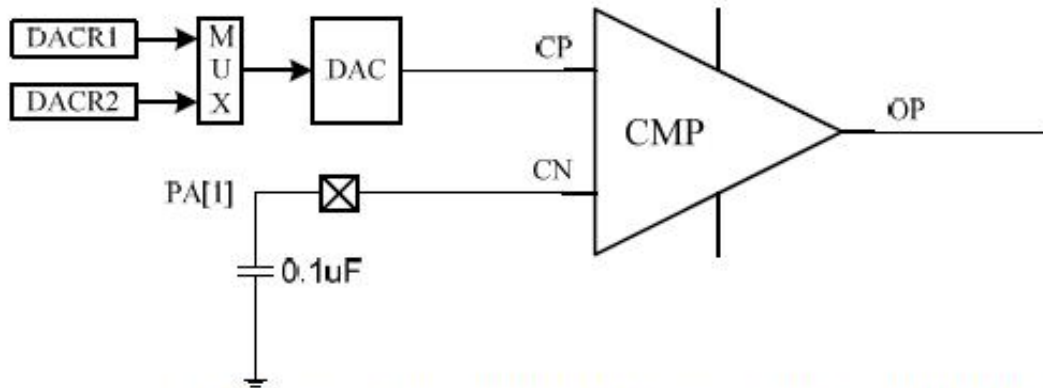
地址 3AH: DACR2LB (DACR2 低字节寄存器)

位	7	6	5	4	3	2	1	0
名称	--	--	--	--	DACR2[3:0]			
读/写	R	R	R	R	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

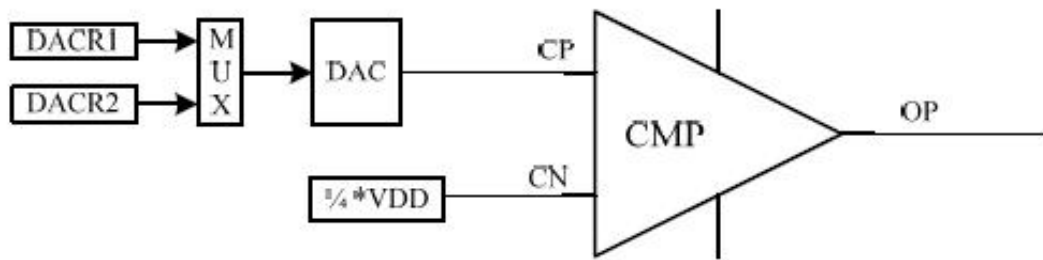
位[3:0] : DACR2[3:0]

DACR2 低 4 位字节数据

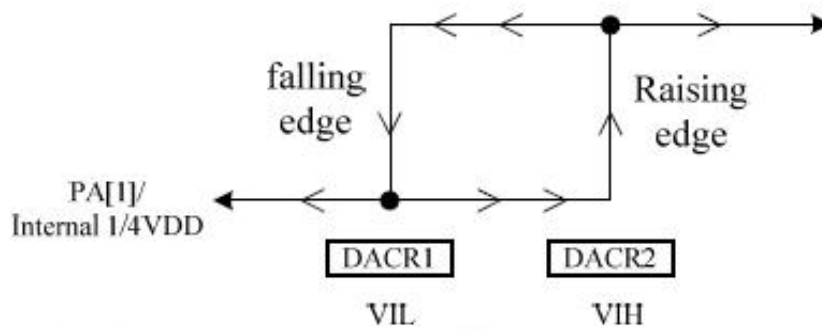
8.2 比较器迟滞应用



8.2-1 Compared External PA[1] Hysteresis application(TMD-1/TMD-3)



8.2-2 Compared Internal 1/4 VDD Hysteresis application(TMD-1/TMD-3)



8.2-3 Hysteresis Diagram of PA[1] or 1/4VDD

【9】 中断

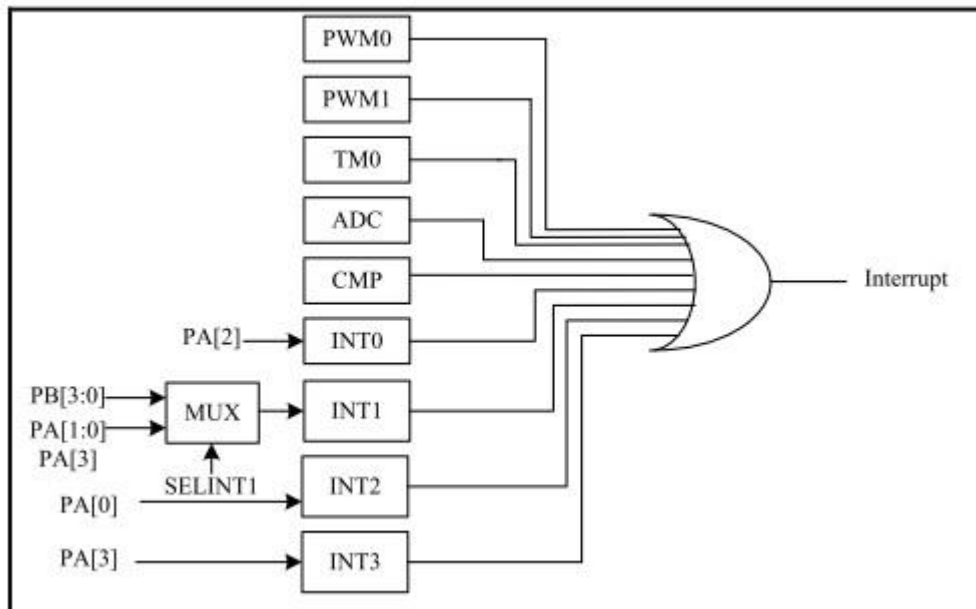
AT8PC712A 具备以下各种中断:

- (1) PTM0 / PTM1 中断
- (2) TMO 定时器中断
- (3) INT0, INT1, INT2 及 INT3 外部中断
- (4) ADC 样本完成中断
- (5) 比较器输出改变中断

AT8PC712A 的复位向量固定在 0x000h, 而中断向量在 0x003h.

总体中断使能位 GIE(0Bh-Bi t7), 可使能 (设置 1) 所有未被屏蔽的中断, 或禁能 (清除为 0) 所有中断。

个别的中断可以通过 INTEN 寄存器中相应的位来作使能/禁能操作。当 GIE 设置在使能位时, 各中断的优先权可由客户依据需求由软件来决定。



中断方框图

9.1 中断寄存器

地址 0B: INTEN (中断使能寄存器)

位	7	6	5	4	3	2	1	0
名称	GIE	--	--	INT3IE	INT2IE	INT1IE	INT0IE	TMOIE
读/写	R/W	R	R	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7] : GIE

总体中断使能位

1 = 使能

0 = 关闭

位[6:5] : 保留

位[4] : INT3IE

INT3 中断使能位

1 = 使能

0 = 关闭

位[3] : INT2IE

INT2 中断使能位

1 = 使能

0 = 关闭

位[2] : INT1IE

INT1 中断使能位

1 = 使能

0 = 关闭

位[1] : INT0IE

INT0 中断使能位

1 = 使能

0 = 关闭

位[0] : TMOIE

Timer 0 中断使能位

1 = 使能

0 = 关闭

地址 0CH: INTFLAG (中断标示位寄存器)

位	7	6	5	4	3	2	1	0
名称	--	--	--	INT3IF	INT2IF	INT1IF	INT0IF	TMOIF
读/写	R	R	R	R/W	R/W	R/W	R/W	R/W
复位初值	0	0	0	0	0	0	0	0

位[7:5] : 保留

位[4] : INT3IF

INT3 外部中断标示位, 写入 0 清除标示位

1 = INT3 中断发生[PA[3]].

0 = 未产生中断。

位[3] : INT2IF

INT2 外部中断标示位(INT2 下降沿), 写入 0 清除标示位
 1 = INT2 中断发生[PA[0]].
 0 = 未产生中断。

位[2] : INT1IF

INT1 外部中断标示位, 写入 0 清除标示位
 1 = INT1 中断发生请参考 SELINT1[2:0].
 0 = 未产生中断。

位[1] : INTOIF

INT0 外部中断标示位(INT0 下降沿), 写入 0 清除标示位
 1 = INT0 中断发生[PA[2]].
 0 = 未产生中断。

位[0] : TMOIF

Timer0 中断标示位, 写入 0 清除标示位
 1 = Timer0 计数器溢位及中断发生
 0 = 未产生中断。

地址 ODH: INTCON (中断控制寄存器)

位	7	6	5	4	3	2	1	0
名称	SELINT1[2]	SELINT1[1]	SELINT1[0]	INT3CON	INT2CON	INT1CON	INTOCON	--
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	--
复位初值	0	0	0	0	0	0	0	0

位[7:5] : SELINT1[2:0]

选择 INT1 中断引脚

SELINT1[2:0]	选择中断脚位
000	PA[1]
001	PA[0]
010	PA[3]
011	PB[0]
100	PB[1]
101	PB[2]
110	PB[3]

位[4] : INT3CON

1 = INT3 引脚变化发生中断
 0 = INT3 下降沿发生中断

位[3] : INT2CON

1 = INT2 引脚变化发生中断
 0 = INT2 下降沿发生中断

位[2] : INT1CON

1 = INT1 引脚变化发生中断
 0 = INT1 下降沿发生中断

位[1] : INTOCON

1 = INTO 引脚变化发生中断
 0 = INTO 下降沿发生中断

位 [0]: 保留

【10】 指令集

指令名称	描述	操作	周期	Status 所受影响	
BCR	R, bit	R 中的指定位清零	0 → R	1	-
BSR	R, bit	R 中的指定位设为 1	1 → R	1	-
BTRSC	R, bit	判断 R 中的指定位, 如果为 0 则跨越	Skip if R = 0	1/2 (1)	-
BTRSS	R, bit	判断 R 中的指定位, 如果为 1 则跨越	Skip if R = 1	1/2 (1)	-
NOP		无操作	No operation	1	-
CLRWDT		清除看门狗定时器	00h → WDT, 00h → WDT prescaler	1	\overline{TO} , \overline{PD}
SLEEP		进入睡眠模式	00h → WDT, 00h → WDT prescaler	1	\overline{TO} , \overline{PD}
RETURN		从子程序返回	Top of Stack → PC	2	-
RETFIE		从中断返回, 且设定 GIE 位	Top of Stack → PC, 1 → GIE	2	-
CLRA		清除 ACC	00h → ACC	1	Z
IOST	R	加载到 R-plane 寄存器	ACC → R-plane register	1	-
IOSTR	R	读取 R-plane 寄存器	R-plane register → ACC		
CLRR	R	清除寄存器 R	00h → R	1	Z
MOVAR	R	将 ACC 中的内容移到 R	ACC → R	1	-
MOVRR	R, d	将 R 中的内容移到目的地	R → dest	1	Z
DECR	R, d	R 内容减 1 后存到目的地	R - 1 → dest	1	Z
DECRSZ	R, d	R 内容减 1 后存到目的地, 若为 0 则跳跃	R - 1 → dest, Skip if result = 0	1/2 (1)	-
INCR	R, d	R 内容加 1 后存到目的地	R + 1 → dest	1	Z
INCRSZ	R, d	R 内容加 1 后存到目的地, 若为 0 则跳跃	R + 1 → dest, Skip if result = 0	1/2 (1)	-
ADDAR	R, d	R 和 ACC 相加后存到目的地	R + ACC → dest	1	C, DC, Z
SUBAR	R, d	R 和 ACC 相减后存到目的地	R - ACC → dest	1	C, DC, Z
ADCAR	R, d	R 和 ACC 及 Carry 相加后存到目的地	R + ACC + C → dest	1	C, DC, Z
SBCAR	R, d	R 和 ACC 相减后加 Carry 后存到目的地	R + \overline{ACC} + C → dest	1	C, DC, Z
ANDAR	R, d	ACC 和 R 作逻辑与后存到目的地	ACC and R → dest	1	Z
IORAR	R, d	ACC 和 R 作逻辑或后存到目的地	ACC or R → dest	1	Z
XORAR	R, d	ACC 和 R 作逻辑异或后存到目的地	R xor ACC → dest	1	Z
COMR	R, d	取 R 内容值补码, 存到目的地	\overline{R} → dest	1	Z
RLR	R, d	R 与 C 左移一个位后存到目的地	R<7> → C, R<6:0> → dest<7:1>, C → dest<0>	1	C
RRR	R, d	R 与 C 右移一个位后存到目的地	C → dest<7>, R<7:1> → dest<6:0>, R<0> → C	1	C
SWAPR	R, d	R 的高低位 4 位互换后存到目的地	R<3:0> → dest<7:4>, R<7:4> → dest<3:0>	1	-
MOVIA	I	加载立即值 I 到 ACC	I → ACC	1	-

指令名称	描述	操作	周期	Status 所受影响
ADDIA I	立即值 I 加上 ACC 后存入 ACC	$I + ACC \rightarrow ACC$	1	C, DC, Z
SUBIA I	立即值 I 减去 ACC 后存入 ACC	$I - ACC \rightarrow ACC$	1	C, DC, Z
ANDIA I	ACC 与立即值 I 相与后存入 ACC	$ACC \text{ and } I \rightarrow ACC$	1	Z
IORIA I	ACC 与立即值 I 相或后存入 ACC	$ACC \text{ or } I \rightarrow ACC$	1	Z
XORIA I	ACC 与立即值 I 相异或后存入 ACC	$ACC \text{ xor } I \rightarrow ACC$	1	Z
RETIA I	将立即值 I 存到 ACC 并从子程序返回	$I \rightarrow ACC,$ $\text{Top of Stack} \rightarrow PC$	2	-
CALL I	调用子程序	$PC + 1 \rightarrow \text{Top of Stack},$ $I \rightarrow PC$	2	-
GOTO I	无条件跳跃	$I \rightarrow PC$	2	-

注意:

1. 跳跃指令(GOTO/CALL) 及返回指令 (RETURN / RETIA / RETFIE) 及 判断指令 (除要跳跃到下一位址动作), 需要 2 个周期外, 其它指令都是 1 个周期。

2. Bit : 指定的位地址 (0-7)
 R : 数据寄存器地址 (0x00h ~ 0x7Fh)
 I : 常数数据
 ACC : 寄存器
 d : 目的地选择
 : =0 (将结果存到 ACC)
 : =1 (将结果存到寄存器 R)
 dest : 目标单元
 PC : 程序计数器
 PCHBUF : 程序计数器高位缓存器
 WDT : 看门狗定时器计数器
 GIE : 总体中断使能位
 \overline{TO} : 溢出标示位
 \overline{PD} : 电源关闭位
 C : 进位标示位
 DC : 辅助进位标示位
 Z : 零位标示位

【11】 最大绝对额定值

相关参数	条件	数值		单位
		最小	最大	
周围环境工作温度	-	-40	85	°C
存储温度	-	-40	150	°C
直流供应电压	相对于 VSS	2.4	5.5	V
供应电流	-	-	-	mA
所有引脚电压	相对于 VSS	-0.3	VDD+0.3V	V

【12】 DC 特性

12.1 概要 (工作温度 = 0 to 70 °C)

符号	参数	条件	值			单位
			最小	典型	最大	
VDD	工作电压	--	2.4	5.0	5.5	V
I _{dd1}	正常模式工作电流 典型 = (ADC 关闭)	Vdd=5V, 无 GPIO 负载 IRC 工作中, FCPU=16MHz		1.5		mA
	正常模式工作电流 典型 = (Disable ADC)	Vddc=3V, 无 GPIO 负载 IRC 工作中, FCPU=16MHz		1.5		mA
	正常模式工作电流 典型 = (Disable ADC)	Vdd=5V, 无 GPIO 负载 IRC 工作中, FCPU=1MHz		0.5		mA
	正常模式工作电流 典型 = (Disable ADC)	Vddc=3V, 无 GPIO 负载 IRC 工作中, FCPU=1MHz		0.5		mA
I _{dd2}	低频模式工作电流 典型 = (Disable ADC)	Vdd=5V, 无 GPIO 负载 SIRC 工作中, FCPU=32KHz		230		uA
	低频模式工作电流 典型 = (Disable ADC)	Vdd=3V, 无 GPIO 负载 SIRC 工作中, FCPU=16KHz		215		uA
I _{dd3}	绿色模式工作电流 (watch Dog Enable) 典型 = (Disable ADC)	Vdd=5V, 无 GPIO 负载 WDT = 32ms SIRC 工作中, FCPU=Disable		12		uA
	绿色模式工作电流 (watch Dog Enable) Typical = (Disable ADC)	Vdd=3V, 无 GPIO 负载 WDT = 32ms SIRC 工作中, FCPU=disable		6		uA
I _{dd4}	休眠模式工作电流 典型 = (Disable ADC)	Vdd=5V, 无 GPIO 负载 25 °C		1		uA
	休眠模式工作电流 典型 = (Disable ADC)	Vdd=3V, 无 GPIO 负载 25 °C		0.5		uA
	休眠模式工作电流 典型 = (Disable ADC)	Vdd=5V, 无 GPIO 负载 0-70 °C				
	休眠模式工作电流 典型 = (Disable ADC)	Vdd=3V, 无 GPIO 负载 0-70 °C				

This datasheet contains new product information. ATW Technology reserves the rights to modify the product specification without notice.
No liability is assumed as a result of the use of this product. No rights under any patent accompany the sales of the product.

符号	参数	条件	值			单位
			最小	典型	最大	
V _{LVDT}	低电压检测	VDT36, $\Delta = 150 \text{ mV}$	3.6- Δ		3.6	V
		VDT24, $\Delta = 150 \text{ mV}$	2.4- Δ		2.4	V
		VDT30, $\Delta = 150 \text{ mV}$	3.0- Δ		3.0	V
V _{POR}	电源启动复位	电源从 0V 到 VDD (VIH) (POR)		2.4		V
		电源从 VDD 到 0V (VIL) (PDR)		2		V
F _{CPU}	MCU 时钟频率	25 °C, VDD > 2.0V			8	MHz
						MHz
F _{IRC}	内部高频 RC 频率	VDD=5V, 25 °C, F _{IRC} 16Mhz $\Delta = 16 * (2\%)$	16 - Δ	16	16+ Δ	MHz
		VDD=2.2-5V, @25°C F _{IRC} = 16Mhz $\Delta = 16 * (2.5\%)$	16 - Δ	16	16+ Δ	MHz
F _{SIRC}	内部低频 RC 频率	VDD=2.2-5V, @25°C $\Delta = 32 * (10\%)$	32 - Δ	32	32+ Δ	KHz
T _{POREXT}	电源启动复位时间	SELPOR = 01 (16ms) VDD =5V	13		14	mS
		SELPOR = 01 (16ms) VDD =3V	23		27	mS
V _{INREFH}	内部参考电压	V _{INREFH} = 4V (Vdd=5V, 25 °C) $\Delta = 4V * (2\%)$	4- Δ	4	4+ Δ	V
		V _{INREFH} = 3V (Vdd=5V, 25 °C) $\Delta = 3V * (2\%)$	3- Δ	3	3+ Δ	V
		V _{INREFH} = 2V (Vdd=5V, 25 °C) $\Delta = 2V * (2\%)$	2- Δ	2	2+ Δ	V

12.2 GPIO 接口

符号	参数	条件	值			单位
			最小	标准	最大	
V _{up1}	GPIO 上拉电阻	VDD=5V	90		110	K Ω
V _{up2}	GPIO 上拉电阻	VDD=3V	90		110	K Ω
V _{OL}	输出低电压 (高驱动)	VDD=5V~3V, I _{OL} =15mA	-		0.2	V
		VDD=3V~2.4V, I _{OL} =15mA			0.25	V
V _{OH}	输出高电压	VDD=5~3V, I _{OH} =15mA	VDD-0.4V		-	V
		VDD=3V~2.4V, I _{OH} =11mA				V
V _{IL}	输入低电压	VDD=5V(全部 GPIO 输入)		0.3*VDD		V
		VDD=4V(全部 GPIO 输入)		0.3*VDD		V
		VDD=2.2V(全部 GPIO 输入)		0.3*VDD		V
V _{IH}	输入高电压	VDD=5V(全部 GPIO 输入)		0.7*VDD		V
		VDD=4V(全部 GPIO 输入)		0.7*VDD		V
		VDD=2.2V(全部 GPIO 输入)		0.7*VDD		V
V _{IL2}	输入低电压	复位引脚				V
V _{OH_vpp}	VPP 引脚输出高电压	VDD=5~2.4V, I _{OH} =15mA	VDD-0.7V			V
V _{OL_vpp}	VPP 引脚输出低电压	VDD=5~2.4V, I _{OL} <2mA			0.7V	V
V _{PH_vpp}	VPP 上拉电压	VDD=5V (VPP 输出)	3.5			V
		VDD=4V (VPP 输出)	2.6			V
		VDD=2.4V (VPP 输出)	1.4			V

This datasheet contains new product information. ATW Technology reserves the rights to modify the product specification without notice.
No liability is assumed as a result of the use of this product. No rights under any patent accompany the sales of the product.

12.3 ADC 规格

符号	参数	条件	值			单位
			最小	典型	最大	
ADCSR	ADC 取样率	-			32	KHz
DNL	非线性误差	分辨率 = 12 位 VDD=5.0V, VREFH = 3.3V FADCSR=64K				LSB
		分辨率 = 10 位 VDD=5.0V, VREFH = 3.3V FADCSR=64K				LSB
INL	线性误差	分辨率 = 12 位 VDD=5.0V, VREFH = 3.3V FADCSR=64K				LSB
		分辨率 = 10 位 VDD=5.0V, VREFH = 3.3V FADCSR=64K				LSB
GE	增益误差	分辨率 = 12 位 VDD=5.0V, VREFH = 3.3V FADCSR=64K			--	LSB
		分辨率 = 10 位 VDD=5.0V, VREFH = 3.3V FADCSR=64K			--	LSB
ADCRL	ADC 分辨率 Typical = bit	VDD=5.0V, VREFH = 3.3V FADCSR=64K	8	10	12	位
ADCRL	ADC 分辨率 Typical = bit	VDD=5.0V, VREFH = 内部 2V/3V/4V FADCSR=64K	8	9	11	位
					-	

12.4 CMP 规格

符号	参数	条件	值			单位
			最小	典型	最大	
V _{IO}	输入偏移(Offset)电压	VDD=5V~2.0V		+/-10		mV
V _{ICM}	输入一般模式电压	VDD=5V~2.0V	0		VDD	V

【13】 封包框图

13.1 10-PIN MSOP

