

# AT8BM84A

---

**22 I/O 22+2-通道 ADC 8 位 MTP-Based 单片机**

**Version 1.2**

**Dec. 11, 2025**

---

ATW TECHNOLOGY CO. reserves the right to change this document without prior notice. Information provided by ATW is believed to be accurate and reliable. However, ATW makes no warranty for any errors which may appear in this document. Contact ATW to obtain the latest version of device specifications before placing your orders. No responsibility is assumed by ATW for any infringement of patent or other rights of third parties which may result from its use. In addition, ATW products are not authorized for use as critical components in life support devices/systems or aviation devices/systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of ATW.

## 改版记录

版本	日期	内容描述	修正页
1.0	2025/05/22	初始版本。	-
1.1	2025/09/03	<ol style="list-style-type: none"><li>编辑和重排格式以保持清晰和准确。</li><li>修改封装脚位名称。</li><li>更新 26 章节电气特性</li></ol>	<p>- 12~14 76</p>
1.2	2025/12/11	<ol style="list-style-type: none"><li>更新 ICP/OCD/ISP 名称。</li><li>更新图 29 PWM 方向在接近 100% 占空比时变化的示例</li><li>更新 LVD/CMP。</li><li>删除第 1.5 和 1.6 节的描述。</li><li>更新推荐工作电压。</li></ol>	<p>- 62 71~76 - 138</p>

## 目 录

<b>1. 产品特性</b>	<b>10</b>
1.1 功能特性	10
1.2 系统框图	11
1.3 引脚图	12
1.4 引脚说明	15
1.5 片上仿真 (OCD)	18
1.5.1 概述	18
1.5.2 OCD限制	19
1.6 系统内程序 (ISP)	19
1.6.1 概述	19
1.6.2 ISP限制	19
<b>2. 存储器组织</b>	<b>20</b>
2.1 程序存储器	20
2.1.1 程序存储寄存器	21
2.2 数据存储器	21
2.2.1 数据存储器列表	23
2.2.2 数据存储寄存器	27
2.2.3 特殊功能寄存器	28
2.3 累加器 (ACC)	31
2.3.1 概述	31
2.3.2 ACC 应用	31
<b>3. I/O Ports</b>	<b>32</b>
3.1 概述	32
3.1.1 I/O引脚结构框图	33
3.2 I/O端口寄存器	37
3.2.1 AWUCON (PortA唤醒控制寄存器)	37
3.2.2 BCDWUCON (PortB/C/D 唤醒控制寄存器)	38
3.2.3 PORTACON30 / PORTACON74 / PORTBCON30 / PORTBCON74 / PORTCCON30 / PORTCCON74 (端口属性控制寄存器)	38
3.2.4 PORTA (PORTA 数据寄存器)	39

3.2.5	PORTB (PORTB 数据寄存器) .....	40
3.2.6	PORTC (PORTC 数据寄存器) .....	40
3.2.7	IOSTA (PORTA I/O控制寄存器) .....	40
3.2.8	IOSTB (PORTB I/O控制寄存器) .....	40
3.2.9	IOSTC (PORTC I/O控制寄存器) .....	41
3.2.10	PxCON (端口模拟引脚控制寄存器) .....	41
<b>4.</b>	<b>定时器 0 (Timer0) .....</b>	<b>42</b>
4.1	概述 .....	42
4.2	定时器 0 控制寄存器 .....	43
4.2.1	T0MD 寄存器 .....	43
4.2.2	TMR0 (定时器 0 寄存器) .....	44
<b>5.</b>	<b>定时器 1 / 定时器 4 / 定时器 5 .....</b>	<b>45</b>
5.1	概述 .....	45
5.2	定时器 1 / 定时器 4 / 定时器 5 控制寄存器 .....	47
5.2.1	TMRxL (定时器低字节寄存器) .....	47
5.2.2	TMRxH (定时器高字节寄存器) .....	47
5.2.3	TxCR1 (定时器控制寄存器) .....	48
5.2.4	TxCR2 (定时器控制寄存器 2) .....	49
<b>6.</b>	<b>PWM .....</b>	<b>50</b>
6.1	PWM1 .....	50
6.1.1	概述 .....	50
6.2	PWM2 .....	51
6.2.1	概述 .....	51
6.3	PWM3 .....	51
6.3.1	概述 .....	51
6.4	PWM4 .....	52
6.4.1	概述 .....	52
6.5	PWM5 .....	53
6.5.1	概述 .....	53
6.6	PWM 周期 .....	53
6.7	PWM 占空比 .....	53
6.8	CCP 模式 .....	54

6.8.1	模块配置	54
6.8.2	CCP I/O 配置	54
6.8.3	捕捉模式（仅在CCP1 中可用）	55
6.8.4	比较模式（仅在CCP1 中可用）	56
6.8.5	增强型PWM模式	56
6.8.6	PWM操作设置	57
6.8.7	增强型PWM特性	57
6.8.8	半桥模式（可用于CCP1 和CCP2）	59
6.8.9	全桥模式（可用于CCP1）	60
6.8.10	全桥模式（可用于CCP1）	61
6.8.11	启动注意事项	62
6.8.12	电源控制模式下的操作	62
6.8.13	复位影响	63
6.9	PWM控制寄存器	63
6.9.1	PWMxCON (PWM控制寄存器)	63
6.9.2	PWMDBx (PWM死区寄存器)	64
6.9.3	PWM1DUTY (PWM 1 占空比寄存器)	64
6.9.4	PWM2DUTY (PWM 2 占空比寄存器)	65
6.9.5	PWM3DUTY (PWM 3 占空比寄存器)	65
6.9.6	PWM4DUTY (PWM 4 占空比寄存器)	65
6.9.7	PWM5DUTY (PWM 5 占空比寄存器)	66
6.9.8	CCPxCON (CCPx控制寄存器)	66
7.	蜂鸣器 (Buzzer)	68
7.1	概述	68
7.2	BZ1CR (Buzzer1 控制寄存器)	68
8.	IR (红外) 载波	69
8.1	概述	69
8.2	IRCR (红外控制寄存器)	69
9.	RFC	70
9.1	RFC (RFC控制寄存器)	70
10.	低电压检测 (LVD)	71
10.1	概述	71
10.2	LVDCON (LVD控制寄存器)	71

<b>11. 电压比较器 (CMP) .....</b>	<b>73</b>
11.1 概述 73	
11.2 比较器参考电压 (Vref) .....	73
11.3 比较器控制寄存器.....	75
11.3.1 CMPCON (比较器控制寄存器) .....	75
11.3.2 CMPCR (比较器电压选择控制寄存器) .....	75
<b>12. 模拟-数字转换器 (ADC) .....</b>	<b>77</b>
12.1 概述 77	
12.2 ADC参考电压 .....	77
12.3 ADC模拟输入通道 .....	78
12.4 ADC时钟 (ADCLK)、采样时钟 (SHCLK) 和位数选择 .....	79
12.5 ADC操作过程 .....	80
12.6 ADC数据格式 .....	80
12.7 ADC控制寄存器.....	81
12.7.1 ADMD (ADC模式寄存器) .....	81
12.7.2 ADDL (ADC LSB输出寄存器) .....	82
12.7.3 ADDH (ADC输出数据寄存器) .....	82
12.7.4 ADCON1 (ADC控制寄存器1) .....	82
12.7.5 ADJMD (ADC模拟引脚寄存器) .....	83
12.7.6 ADCR (采样脉冲和ADC位数寄存器) .....	83
<b>13. 看门狗 (WDT) .....</b>	<b>84</b>
13.1 概述 84	
<b>14. 中断 .....</b>	<b>85</b>
14.1 概述 85	
14.1.1 Timer0 上溢中断.....	85
14.1.2 Timer1 下溢中断.....	85
14.1.3 Timer4 下溢中断.....	85
14.1.4 Timer5 下溢中断.....	86
14.1.5 Timer5 下溢 / CCP中断.....	86
14.1.6 WDT 超时中断.....	86
14.1.7 PA/PB/PC 输入状态改变中断.....	86
14.1.8 外部中断0.....	86

14.1.9 外部中断 1.....	86
14.1.10 外部中断 2.....	86
14.1.11 LVD 中断.....	86
14.1.12 比较器输出状态改变中断.....	86
14.1.13 ADC 转换结束中断.....	86
14.1.14 串行接口模式中断.....	86
<b>14.2 中断控制寄存器 .....</b>	<b>87</b>
14.2.1 INTCON (中断控制寄存器) .....	87
14.2.2 PIR1 (中断标志寄存器) .....	87
14.2.3 PIE1 (中断使能寄存器 1) .....	88
14.2.4 PIR2 (中断标志寄存器 2) .....	89
14.2.5 PIE2 (中断使能寄存器 2) .....	89
14.2.6 PCON (Power 寄存器) .....	90
14.2.7 INTEDG (外部中断控制寄存器) .....	91
<b>15. 振荡器配置 .....</b>	<b>92</b>
15.1 概述 92	
15.2 OSCCR (振荡器控制寄存器) .....	94
<b>16. 工作模式 .....</b>	<b>95</b>
16.1 概述 95	
16.2 正常模式.....	96
16.3 慢速模式.....	96
16.4 待机模式.....	96
16.5 睡眠模式.....	97
16.6 唤醒稳定时间.....	97
16.7 工作模式概述.....	98
<b>17. 复位 .....</b>	<b>98</b>
17.1 概述 98	
<b>18. SPI 模式 .....</b>	<b>100</b>
18.1 概述 100	
18.2 串行时钟极性和相位 .....	102
18.3 SPI 错误条件 .....	103

18.4 SPI控制寄存器	103
18.4.1 SIMDR (串行接口模式数据寄存器)	103
18.4.2 SIMCR (串行接口模式控制寄存器)	103
18.4.3 SPCR (SPI控制和状态寄存器)	104
<b>19. I<sup>2</sup>C 模式</b>	<b>105</b>
19.1 概述	105
19.2 I <sup>2</sup> C 模式协议	106
19.3 I <sup>2</sup> C 模式操作	106
19.4 仲裁机制	107
19.5 I <sup>2</sup> C 控制寄存器	108
19.5.1 MADR (I <sup>2</sup> C模式地址寄存器)	108
19.5.2 MFDR (I <sup>2</sup> C模式频率寄存器)	108
19.5.3 MCR (I <sup>2</sup> C模式控制寄存器)	109
19.5.4 MSR (I <sup>2</sup> C模式状态寄存器)	109
<b>20. 通用异步收发器 (UART)</b>	<b>111</b>
20.1 概述	111
20.2 UART 控制寄存器	111
20.2.1 DLL (波特率除法锁存LSB寄存器)	111
20.2.2 DLH (波特率除法锁存MSB寄存器)	111
20.2.3 LCR (行控制寄存器)	112
20.2.4 LSR (行状态寄存器)	113
20.2.5 THR/RBR (发送保持寄存器/接收缓冲寄存器)	113
<b>21. LCD</b>	<b>114</b>
21.1 概述	114
21.2 LCD控制寄存器	115
21.2.1 LCDCON0 (LCD控制寄存器0)	115
21.2.2 LCDCON1 (LCD控制寄存器1)	115
<b>22. 片上仿真 (OCD)</b>	<b>117</b>
22.1 概述	117
22.2 OCD限制	117
<b>23. 系统内程序 (ISP)</b>	<b>117</b>

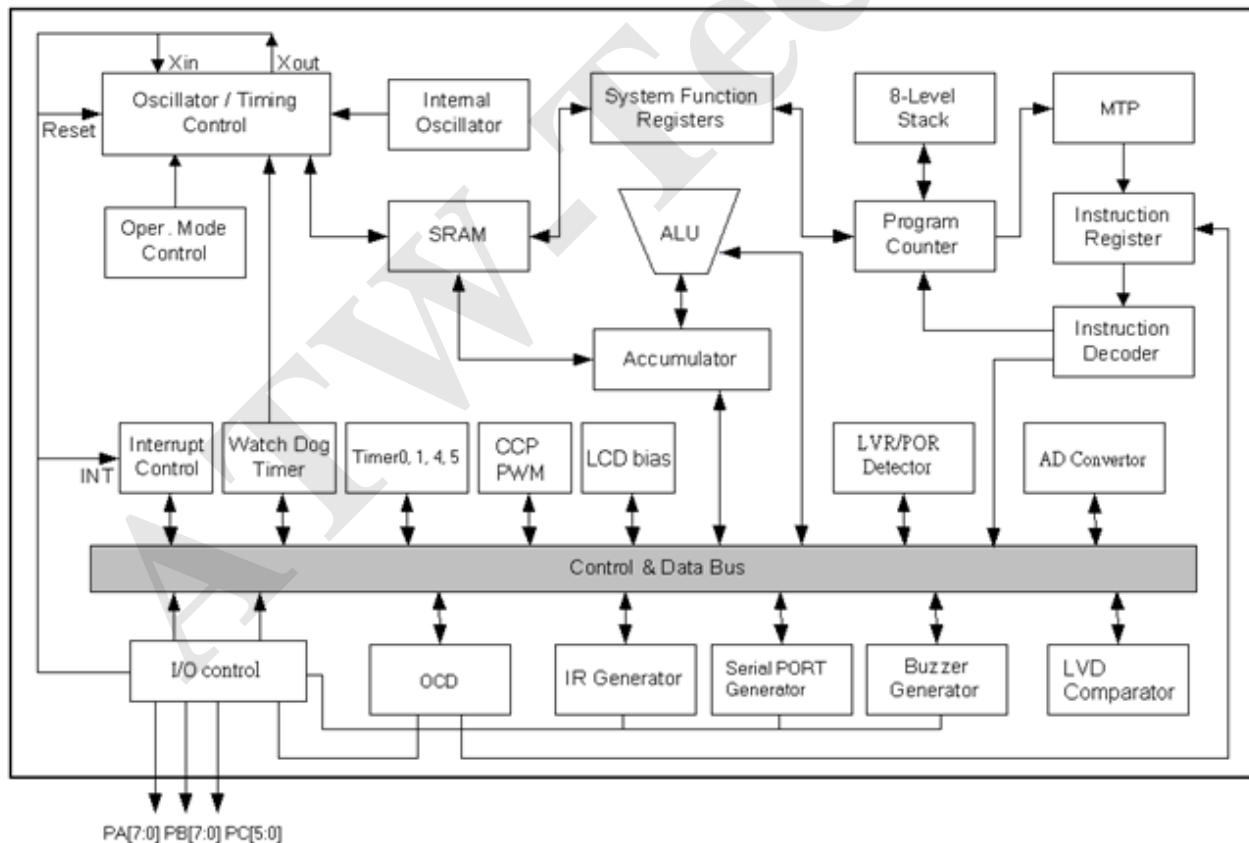
23.1 概述	117
23.2 ISP限制	117
<b>24. 指令设置</b>	<b>118</b>
<b>25. 配置表</b>	<b>131</b>
<b>26. 电气特性</b>	<b>133</b>
26.1 最大绝对值	133
26.2 直流电气特性	133
26.3 比较器/ LVD电气特性	134
26.4 OSC电气特性	135
26.5 ADC电气特性	135
26.6 特性曲线图	136
26.6.1 $I_{HRC}$ 的频率与 $V_{DD}$ 曲线图	136
26.6.2 $I_{HRC}$ 的频率与温度曲线图	136
26.6.3 $I_{LRC}$ 的频率与 $V_{DD}$ 曲线图	137
26.6.4 $I_{LRC}$ 的频率与温度曲线图	137
26.6.5 上拉电阻与 $V_{DD}$ 曲线图	138
26.6.6 下拉电阻与 $V_{DD}$ 曲线图	138
26.7 推荐工作电压	138
<b>27. 封装尺寸</b>	<b>139</b>
27.1 24 引脚QFN	139
27.2 24 引脚SSOP	139
27.3 20 引脚TSSOP	140
27.4 16 引脚SOP (150 mil)	140
<b>28. 订购信息</b>	<b>141</b>

## 1. 产品特性

### 1.1 功能特性

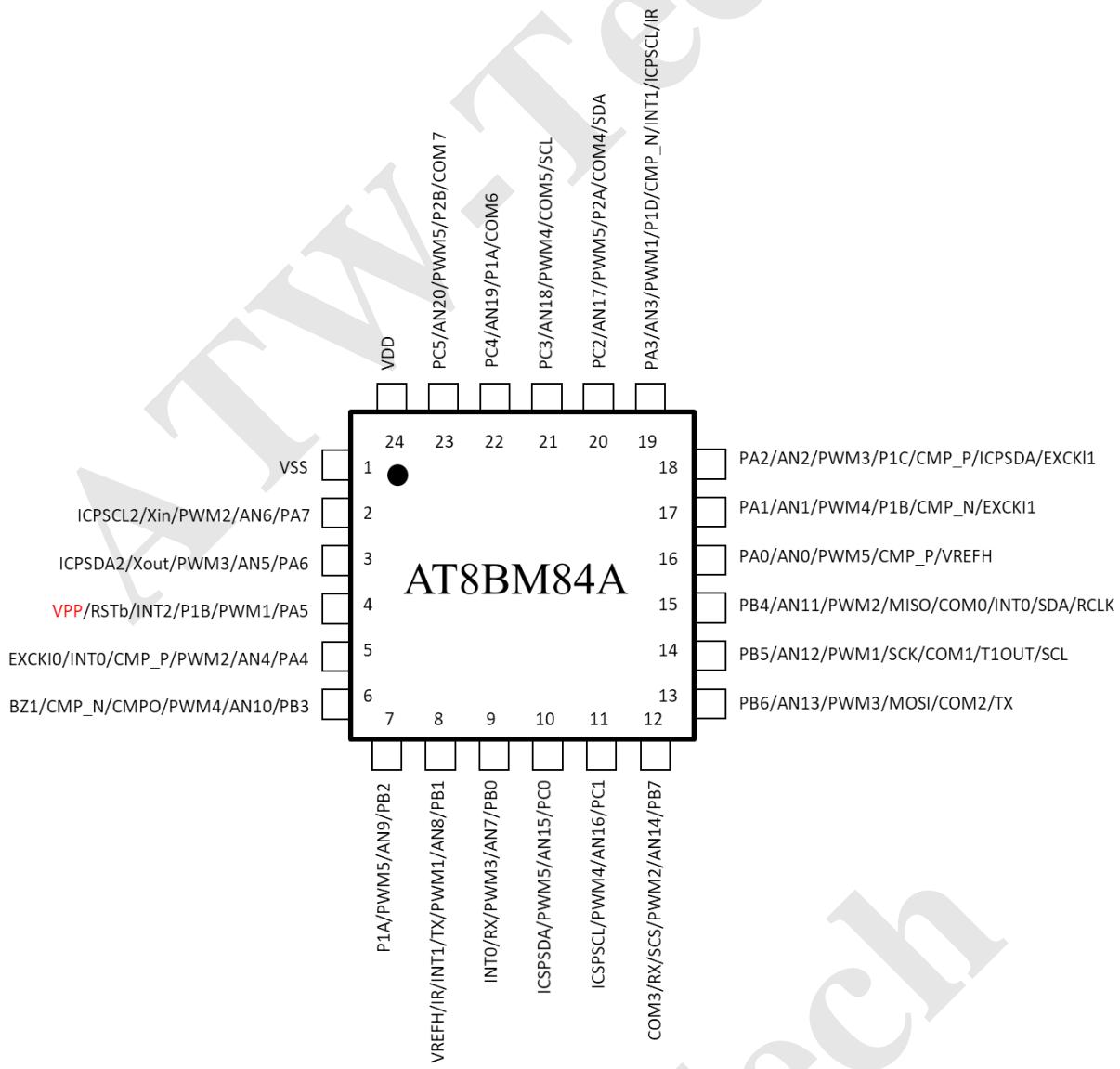
- 指令系统
  - 系统时钟最高支持 24MHz/2T
- 内存
  - 4Kx16 bits MTP。
  - 336 bytes SRAM。
- 5 种震荡方式
  - I\_HRC-内部高速震荡: 24MHz
  - E\_HXT-外部高速震荡: 24MHz
  - E\_XT-外部震荡: 455K~6MHz
  - I\_LRC-内部低速震荡: 32.768KHz
  - E\_LXT-外部低速震荡: 32.768KHz
- 内置WDT 定时器
- 中断源
  - PA/PB/PC口电平变化中断
  - 外部中断INT0/INT1/INT2
  - 其他中断
- 定时器
  - 8 位定时器Timer0 可选外部 32.768KHz 或内部 32.768KHz。
  - 10 位定时器 Timer1/Timer4/Timer5
- 低压复位功能(LVR)
  - 1.8V/2.0V/2.4V/2.7V/3.0/3.3V/3.6V
- GPIO
  - 最多 22 个GPIO
  - 所有数字功能可分配到任意GPIO
  - 均支持上/下拉电阻功能
- 低电压检测(LVD)
  - 内建 16 阶准确的低电压侦测电路功能
- 通信模块
  - 1xSPI
  - 1xI<sup>2</sup>C
  - 1xUART
- 工作电压范围
  - 3.0V ~ 5.5V @ 24MHz/2T
  - 1.6V ~ 5.5V @ 4MHz/4T
- 工作温度
  - 40°C~85°C
- 蜂鸣器驱动
  - 50%占空比, 频率可自由设置
- 增强型PWM
  - 五通道PWM(1/2/3/4/5)
  - 支持独立/互补/同步/成组模式
  - 支持CCP 死区互补模式
  - 1 组全桥(FB)或 2 组半桥(HB)模式
- 高精度 12 位ADC
  - 所有GPIO(22I/Os)均支持ADC通道
  - 参考电压可选择(2V/3V/4V/VDD)
  - 可支持外部电压检测(PA0 或PB1)
  - 支持硬件触发自动转换功能
- 一路模拟比较器
  - 负端可选择内部 0.6V 基准电压或Vref
  - 正端选择内部Vref 或外部电压(PA1 / PB3 / PA3)
- 内建二线控制的除错仿真电路(On Chip Debug)。
- 支持二线带电程序升级 (In-System Programming)(ISP)。

## 1.2 系统框图

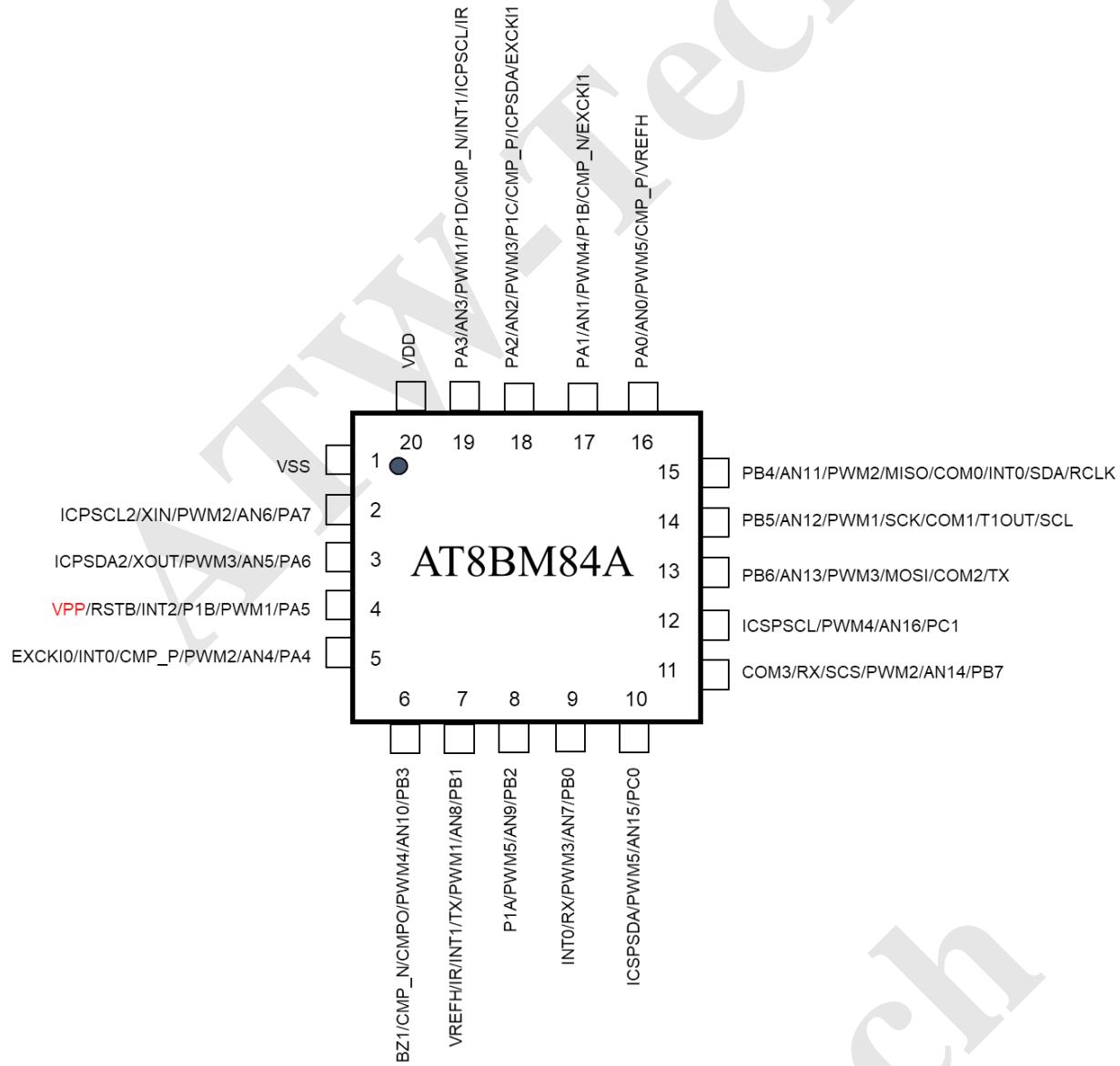


## 1.3 引脚图

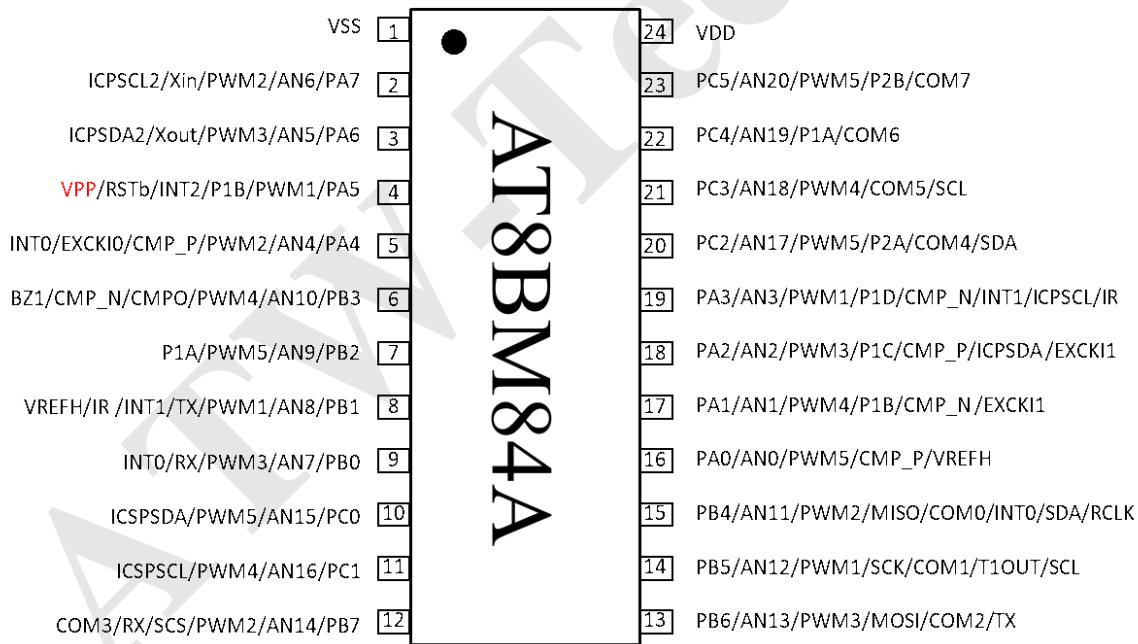
AT8BM84A提供以下封装类型: QFN24、QFN20、SSOP24、TSSOP20、SOP16 和SOP8。



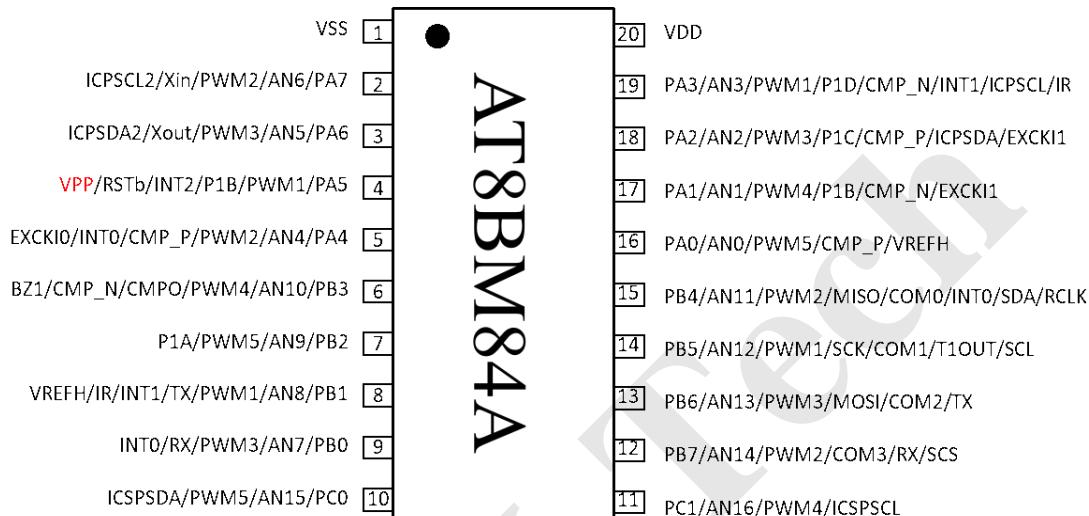
QFN20 封裝



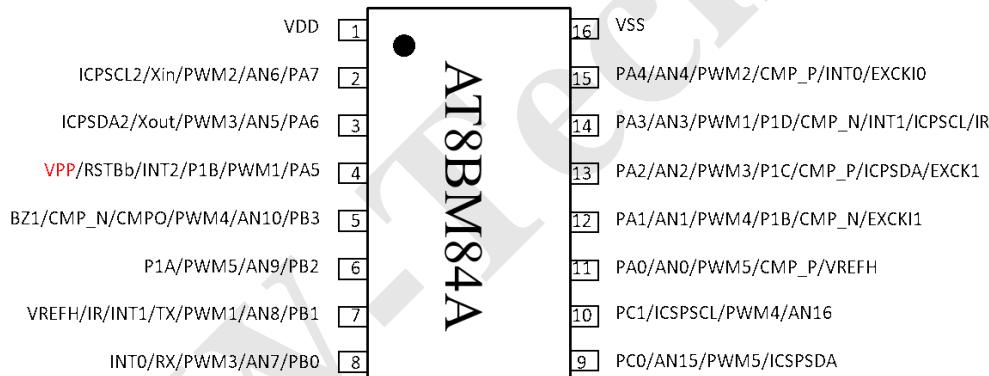
## SSOP24 封裝



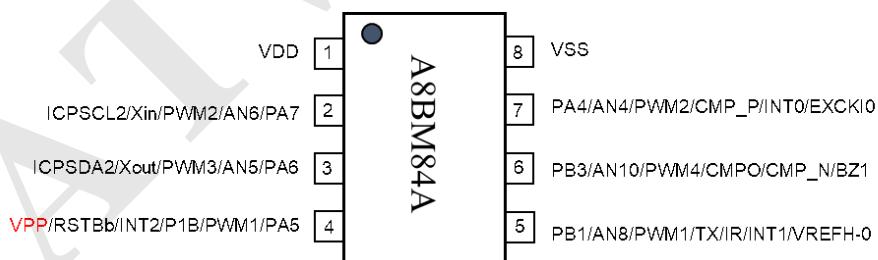
## TSSOP20 封裝



## SOP16 封装



## SOP8 封装



## 1.4 引脚说明

引脚名	I/O	描述
PA0 / AN0 / PWM5 / CMP_P / VREFH	I/O	PA0 是一个双向I/O引脚。 AN0 是ADC模拟输入引脚。 PWM5 是PWM输出引脚。 CMP_P是比较器正向输入引脚。 VREFH是ADC高参考输入引脚。
PA1 / AN1 / PWM4 / P1B / CMP_N / EXCKI1	I/O	PA1 是一个双向I/O引脚。 AN1 是ADC模拟输入引脚。 PWM4 是PWM输出引脚。 P1B是CCP输出引脚。 CMP_N是比较器反向输入引脚。 EXCKI1 是Timer4/5 的时钟源。
PA2 / AN2 / PWM3 / P1C / CMP_P / ICPSDA / EXCKI1	I/O	PA2 是一个双向I/O引脚。 AN2 是ADC模拟输入引脚。 PWM3 是PWM输出引脚。 P1C是CCP输出引脚。 CMP_P是比较器正向输入引脚。 ICPSDA 是ICP/OCD SDA 引脚。 EXCKI1 是Timer4/5 的时钟源。

引脚名	I/O	描述
PA3 / AN3 / PWM1 / P1D / CMP_N / INT1 / ICPSCL / IR	I/O	PA3 是一个双向I/O引脚。 AN3 是ADC模拟输入引脚。 PWM1 是PWM输出引脚。 P1D是CCP输出引脚。 CMP_N是比较器反向输入引脚。 INT1是外部中断引脚。 ICPSCL是ICP/OCD SCL引脚。 IR模式开启时，PA3 是IR载波输出，输出高灌电流。
PA4 / AN4 / PWM2 / CMP_P / INT0 / EXCKI0	I/O	PA4 是一个双向I/O引脚。 AN4 是ADC模拟输入引脚。 PWM2 是PWM输出引脚。 CMP_P是比较器正向输入引脚。 INT0 是外部中断引脚。 EXCKI0 是Timer0/1 的时钟源。
PA5 / PWM1 / P1B / INT2 / RSTb / VPP	I/O	PA5 是一个双向I/O引脚。 PWM1 是PWM输出引脚。 P1B是CCP输出引脚。 INT2 是外部中断引脚。 PA5 可作为复位引脚RSTb。 VPP烧录引脚，它也可使IC进入MTP烧录模式。
PA6 / AN5 / PWM3 / Xout / ICPSDA2	I/O	PA6 是一个双向I/O引脚。 AN5 是ADC模拟输入引脚。 PWM3 是PWM输出引脚。 PA6 可作为外部晶振的输出引脚Xout。 ICPSDA2 是ICP SDA引脚。
PA7 / AN6 / PWM2 / Xin / ICPSCL2	I/O	PA7 是一个双向I/O引脚。 AN6 是ADC模拟输入引脚。 PWM2 是PWM输出引脚。 PA7 可作为外部晶振的输入引脚Xin。 ICPSCL2 是ICP SCL引脚。
PB0 / AIN7 / PWM3 / RX / INT0	I/O	PB0 是一个双向I/O引脚。 AIN7 是ADC模拟输入引脚。 PWM3 是PWM输出引脚。 RX是UART RX引脚。 INT0 是外部中断引脚。
PB1 / AN8 / PWM1 / TX / IR / VREFH / INT1	I/O	PB1 是一个双向I/O引脚。 AN9 是ADC模拟输入引脚。 PWM1 是PWM输出引脚。 TX是UART TX引脚。 IR模式开启时，PB1 是IR载波输出，输出正常灌电流。 VREFH是ADC高参考输入引脚。 INT1 是外部中断引脚。

引脚名	I/O	描述
PB2 / AN9 / PWM5 / P1A	I/O	PB2 是一个双向I/O引脚。 AN9 是ADC模拟输入引脚。 PWM5 是PWM输出引脚。 P1A是CCP输出引脚。
PB3 / AN10 / PWM4 / CMPO / CMP_N / BZ1	I/O	PB3 是一个双向I/O引脚。 AN10 是ADC模拟输入引脚。 PWM4 是PWM输出引脚。 CMPO是比较器输出引脚。 CMP_N是比较器反向输入引脚。 BZ1 是蜂鸣器输出引脚。
PB4 / AN11 / PWM2 / MISO / INT0 / SDA / RCLK / COM0	I/O	PB4 是一个双向I/O引脚。 AN11 是ADC模拟输入引脚。 PWM2 是PWM输出引脚。 当AT8BM84A作从机时， MISO是SPI输出引脚。 INT0 是外部中断引脚。 SDA是I <sup>2</sup> C数据引脚。 RCLK是UART波特率输入引脚。 PB4 可输出 LCD 1/2 VDD COM电压。
PB5 / AN12 / PWM1 / SCK / T1OUT / SCL / COM1	I/O	PB5 是一个双向I/O引脚。 AN12 是ADC模拟输入引脚。 PWM1 是PWM输出引脚。 SCK是SPI时钟输入引脚。 Timer1 匹配输出引脚，当Timer1 发生下溢时， T1OUT切换。 SCL是I <sup>2</sup> C时钟输入引脚。 PB5 可输出 LCD 1/2 VDD COM1 电压。
PB6 / AN13 / PWM3 / MOSI / TX/ COM2	I/O	PB6 是一个双向I/O引脚。 AN13 是ADC模拟输入引脚。 PWM3 是PWM输出引脚。 当AT8BM84A作从机时， MOSI是SPI输入引脚。 TX是UART TX引脚。 PB6 可输出 LCD 1/2 VDD COM2 电压。
PB7 / AN14 / PWM2 / SCS / RX / COM3	I/O	PB7 是一个双向I/O引脚。 AN14 是ADC模拟输入引脚。 PWM2 是PWM输出引脚。 SCS是SPI 使能引脚。 RX是UART RX引脚。 PB7 可输出 LCD 1/2 VDD COM3 电压。
PC0 / AN15 / PWM5 / ICSPSDA	I/O	PC0 是一个双向I/O引脚。 AN15 是ADC模拟输入引脚。 PWM5 是PWM输出引脚。 ICSPSDA是ICP/OCD/ISP数据引脚。

引脚名	I/O	描述
PC1 / AN16 / PWM4 / ICSPSCL	I/O	PC1 是一个双向I/O引脚。 AN16 是ADC模拟输入引脚。 PWM4 是PWM输出引脚。 ICSPSCL是ICP/OCD/ISP时钟引脚。
PC2 / AN17 / PWM5 / P2A / SDA/ COM4	I/O	PC2 是一个双向I/O引脚。 AN17 是ADC模拟输入引脚。 PWM5 是PWM输出引脚。 P2A是CCP输出引脚。 SDA是I <sup>2</sup> C SDA引脚。 PC2 可输出 LCD 1/2 VDD COM4 电压。
PC3 / AN18 / PWM4 / SCL/ COM5	I/O	PC3 是一个双向I/O引脚。 AN18 是ADC模拟输入引脚。 PWM4 是PWM输出引脚。 SCL是I <sup>2</sup> C SCL引脚。 PC3 可输出 LCD 1/2 VDD COM5 电压。
PC4 / AN19 / P1A / COM6	I/O	PC4 是一个双向I/O引脚。 AN19 是ADC模拟输入引脚。 P1A是CCP输出引脚。 PC4 可输出 LCD 1/2 VDD COM6 电压。
PC5 / AN20 / PWM5 / P2B / COM7	I/O	PC5 是一个双向I/O引脚。 AN20 是ADC模拟输入引脚。 PWM5 是PWM输出引脚。 P2B是CCP输出引脚。 PC5 可输出 LCD 1/2 VDD COM7 电压。
VDD	P	电源正端。
VSS	P	电源负端。

P=电源; I/O= 双向的;:

## 1.5 片上仿真 (OCD)

### 1.5.1 概述

AT8BM84A内嵌片上调试仿真功能(OCD)，为开发人员提供了一种低成本的调试用户代码的方法，OCD提供了程序流程控制的调试能力，包括 3 个硬件地址断点、1 个条件寄存器中断、单步、自由运行和内存访问命令。OCD系统使用两线串行接口ICPSCL和ICPSDA，在目标设备和控制调试器主机之间建立通信。ICPSDA是用于调试数据传输的输入/输出引脚，ICPSCL是用于与ICPSDA同步的输入引脚。AT8BM84A还使用ICPSCL和ICPSDA作为控制引脚来写入和读取它。

AT8BM84A作为OCD空间共享ROM地址 0XF00~0xFFFF。如果用户使用超过 0XEFF 的程序空间。OCD功能将被禁用。

### 1.5.2 OCD 限制

AT8BM84A是一个功能齐全的微控制器，在其有限的I/O引脚上多路复用多个功能。必须牺牲一些设备功能来为OCD系统提供资源。OCD有以下限制：

1、ICPSCL/ICPSDA引脚物理上位于同一引脚 PC1/PC0 或 PA3/PA2 或 PA7/PA6 上。因此，它的I/O功能和共享的多功能都无法仿真。

2、系统时钟不能关闭，因为OCD使用该时钟来监视其内部状态：当系统处于睡眠模式时，由于设备的某些部分可能没有时钟，所以执行ram/寄存器访问是无效的。读访问可能返回无用的信息，或者写访问可能不会成功。但以下访问不受系统停止的影响：读取当前程序地址、当前PCL、当前中断条件和当前睡眠状态。

## 1.6 系统内程序 (ISP)

### 1.6.1 概述

基于MTP ROM的存储器可以通过“系统内程序”(ISP)进行编程。如果产品是终端用户手中需要固件更新的终端产品，硬件编程方式会使重复编程变得困难和不方便，ISP方法使其变得容易和可能，而无需从系统中移除AT8BM84A。

有两个信号引脚 ICSPSDA(PC0)和ICSPSCK(PC1)，参与ISP功能。ICSPSDA是数据输入和输出引脚。ICSPSCK是时钟输入引脚，用于同步编程时AT8BM84A输入或输出的数据。用户应该在电路板上留下这两个引脚加上VDD和GND引脚，以使ISP成为可能。

ATW为AT8BM84A提供了ISP工具(NYISP)，用户可以通过ATW编程器轻松执行ISP。

### 1.6.2 ISP 限制

为避免漏电流，两个信号引脚ICSPSDA和ICSPSCK必须设置为输入拉高I/O。ISP模式不允许客户使用非编程设备制造电路板。

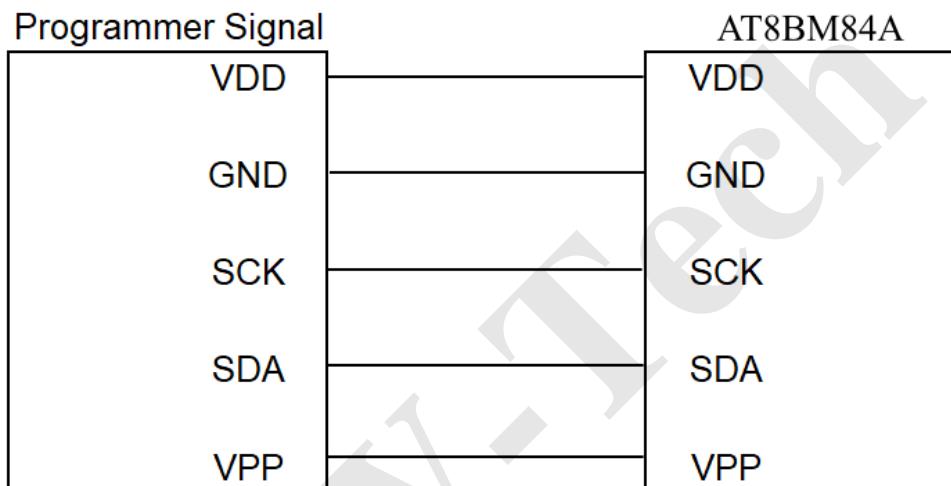


图 1 ISP编程的典型连接

## 2. 存储器组织

AT8BM84A 存储器分为两类：分别是程序存储器和数据存储器。

### 2.1 程序存储器

AT8BM84A程序存储器空间是 4K。因此，12 位的程序计数器（PC）可以访问程序存储器的任何地址。

程序内存的一些位置被保留为中断入口。上电复位地址位于 0x000，软件中断地址位于 0x001，内部和外部硬件中断地址位于 0x004。

AT8BM84A提供指令 LGOTO和 LCALL去访问程序空间的 4K 地址。（注：在增强型指令集版本中，LGOTO 和LCALL指令能够访问 4K 程序空间的任何地址。）

当执行LCALL指令或中断导致分支时，下一个ROM地址写入堆栈的顶部。而当执行RET或RETIE指令时，堆栈顶部的数据会被读取并加载到程序计数器。

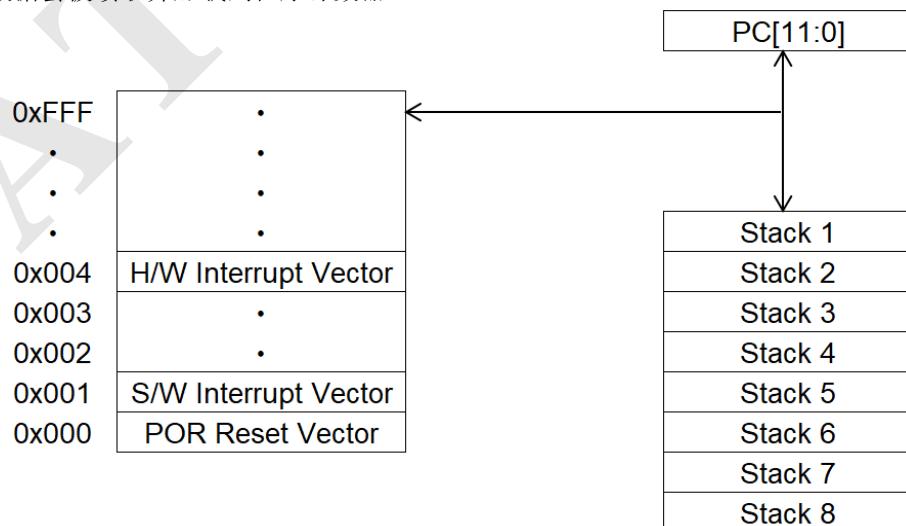


图 2 程序存储器对应地址

## 2.1.1 程序存储寄存器

### 2.1.1.1 PCL (程序计数器低字节)

寄存器PCL是12位PC的最低有效字节( LSB )。除部分指令直接改变PC外，指令执行后PCL将加1。PC的高字节，即PC[11:8]，是不能直接访问的。PC[11:8]的更新必须通过寄存器PCHBUF完成。

当写入PCL时，PC[7:0]是写入的数据。PC[11:8]来自PCHBUF[3:0]。

对于LGOTO指令，PC[10:0]来自指令字节。PC[11]来自PCHBUF[3]。

对于LCALL指令，PC[10:0]来自指令字节。PC[11]来自PCHBUF[3]。

此外，下一个PC地址，即PC+1，将推到堆栈顶部。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCL	0x2	PCL[7:0]							
读/写属性	读/写								
初始值	0x00								

### 2.1.1.2 PCHBUF (程序计数器高字节)

当写入PCL时，PC[11:8]来自PCHBUF[3:0]。对于LGOTO / LCALL指令，PC[11]来自PCHBUF[3]。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCHBUF	0xA					PCHBUF[3:0]			
读/写属性	写								
初始值	x	x	x	x	x	0			

Bit 7:4 未生效的。

Bit 3:0 PCHBUF[3:0]: PC的第11~8位缓存器。

## 2.2 数据存储器

数据存储器被划分为4个bank，其中包含特殊功能寄存器(SFR)和通用寄存器(GPR)。每个bank的前32个位置存放SFR，而作为静态随机存储器(SRAM)类型的GPR则位于每个bank的最后96个位置。由于AT8BM84A的bank 1/2/3的最后16个位置未生效，因此其通用寄存器的最大数量为336个。

数据存储器有两个寻址方式：直接寻址方式和间接寻址方式。

数据存储器的间接寻址方式如下图所描述。这种间接寻址方式是通过访问寄存器INDF而来。即访问INDF SFR时，bank选择由IRP(STATUS[7])和FSR[7]决定，位置选择来自FSR[6:0]。

数据存储器的直接寻址方式如下图所描述。bank选择直接由指令操作码[15:14]决定。而位置选择则直接从指令操作码[6:0]开始。

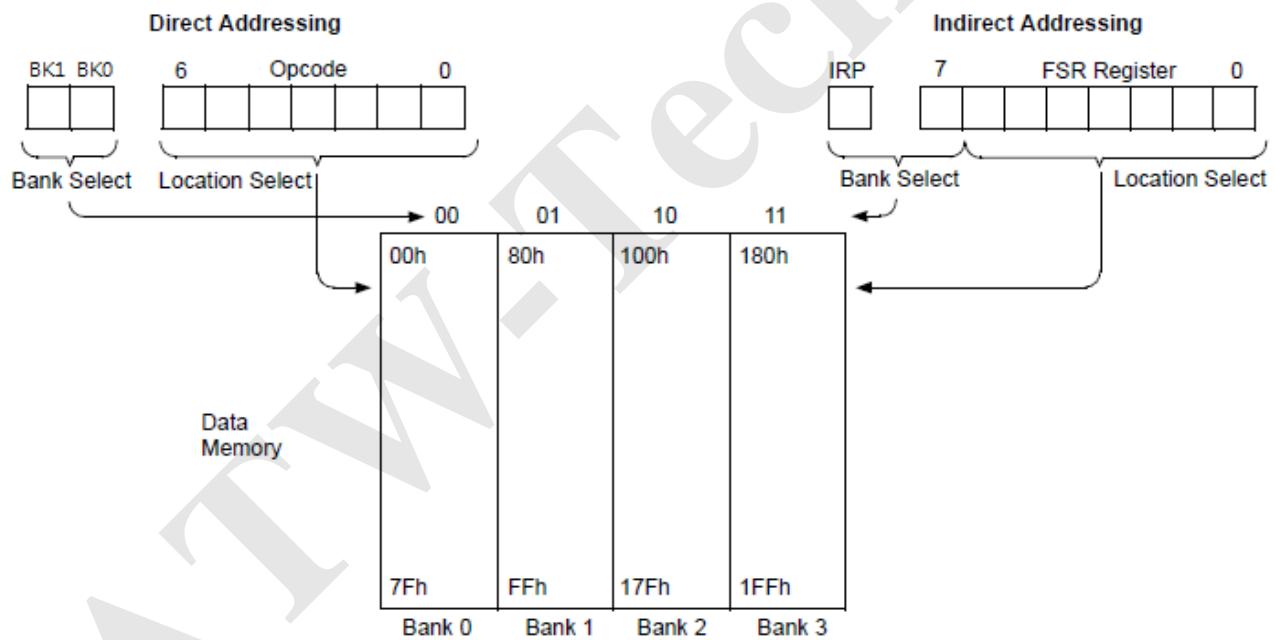


图 3 间接寻址方式/直接寻址方式存取数据存储器

; 通过直接寻址方式写数据内存

MOVIA 0xA5  
MOVAR 0x1A0 ; 将数据 0xA5 写入 Bank3 0x20

; 通过间接寻址方式写数据内存

BSR STATUS,7 ; 通过设置 STATUS[7]和Set FSR[7]选择Bank3  
MOVIA 0x80 | 0x20  
MOVAR FSR ; FSR[6:0]访问地址 0x1A0 (Bank3 0x20)  
MOVIA 0xA5  
MOVAR INDF ; 将数据 0xA5 写入 Bank3 0x20

数据存储器可以通过算术指令和数据移动指令等通用指令访问。SFR占用Bank 0~3 的 0x0 ~ 0x1F地址。GPR物理占用Bank 0 的 0x20 ~ 0x7F地址和Bank 1~3 的 0x20 ~ 0x6F地址。Bank 1~3 的访问地址 0x70~0x7F实际上将访问Bank0 的地址 0x70~0x7F。

下表描述了AT8BM84A寄存器名称和SFR的地址映射。

## 2.2.1 数据存储器列表

Address	Name	Address	Name	Address	Name	Address	Name
00h	INDF	80h	INDF	100h	INDF	180h	INDF
01h	TMR0	81h	TMR0	101h	TMR0	181h	TMR0
02h	PCL	82h	PCL	102h	PCL	182h	PCL
03h	STATUS	83h	STATUS	103h	STATUS	183h	STATUS
04h	FSR	84h	FSR	104h	FSR	184h	FSR
05h	PORTA	85h	IOSTA	105h	PORTA	185h	IOSTA
06h	PORTB	86h	IOSTB	106h	PORTB	186h	IOSTB
07h	PORTC	87h	IOSTC	107h	PORTC	187h	IOSTC
08h		88h		108h		188h	
09h	OSCCR	89h	LVDCON	109h	ADMD	189h	RFC
0Ah	PCHBUF	8Ah	PCHBUF	10Ah	PCHBUF	18Ah	PCHBUF
0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	INTCON
0Ch	PIR1	8Ch	PIE1	10Ch	ADDL	18Ch	LCDCON0
0Dh	PIR2	8Dh	PIE2	10Dh	ADDH	18Dh	LCDCON1
0Eh		8Eh		10Eh	ADCON1	18Eh	
0Fh	TMRxL	8Fh	PCON	10Fh	ADJMD	18Fh	
10h	TMRxH	90h	IRCR	110h	PxCON	190h	
11h	TxCR1	91h	PWM1DUTYL	111h	ADCR	191h	
12h	TxCR2	92h	PWM1DUTYH	112h	SIMDR	192h	
13h	BZ1CR	93h	PWM2DUTYL	113h	SIMCR	193h	
14h	AWUCON	94h	PWM2DUTYH	114h	SPCR	194h	
15h	BCDWUCON	95h	PWM3DUTYL	115h	MADR	195h	
16h	PORTACON30	96h	PWM3DUTYH	116h	MFDR	196h	
17h	PORTACON74	97h	PWM4DUTYL	117h	MCR	197h	
18h	PORTBCON30	98h	PWM4DUTYH	118h	MSR	198h	
19h	PORTBCON74	99h	PWM5DUTYL	119h	DLL	199h	
1Ah	PORTCCON30	9Ah	PWM5DUTYH	11Ah	DLH	19Ah	
1Bh	PORTCCON74	9Bh	INTEDG	11Bh	LCR	19Bh	
1Ch		9Ch	CMPCON	11Ch	LSR	19Ch	
1Dh	SELCON	9Dh	CMPPCR	11Dh	THR/RBR	19Dh	
1Eh	PSxCV	9Eh	TBHP	11Eh	PWMDB	19Eh	
1Fh	PWMxCON	9Fh	TBHD	11Fh	CCPCON	19Fh	
20h	General Purpose Register 80 Bytes	A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h	General Purpose Register 80 Bytes
6Fh		EFh		16Fh		1EFh	
70h	General Purpose Register 16 Bytes (Common)	F0h	Mapped in Bank0	170h	Mapped in Bank0	1F0h	Mapped in Bank0
7Fh		FFh		17Fh		1FFh	

表 2-1 SFR地址映射

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value of POR
00h	INDF	INDF7	INDF6	INDF5	INDF4	INDF3	INDF2	INDF1	INDF0	xxxx xxxx
01h	TMR0	TMR07	TMR06	TMR05	TMR04	TMR03	TMR02	TMR01	TMR00	xxxx xxxx
02h	PCL	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	0000 0000
03h	STATUS	IRP	BK1	BK0	/TO	/PD	Z	DC	C	0001 1xxx
04h	FSR	FSR7	FSR6	FSR5	FSR4	FSR3	FSR2	FSR1	FSR0	xxxx xxxx
05h	PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	xxxx xxxx
06h	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx xxxx
07h	PORTC	-	-	PC5	PC4	PC3	PC2	PC1	PC0	xxxx xxxx
08h	-	-	-	-	-	-	-	-	-	-
09h	OSCCR	-	-	-	XSPD_STP	OPMD1	OPMD0	STPHOSC	SELHOSC	xxxx 0001
0Ah	PCHBUF	-	-	-	-	PCHBUF3	PCHBUF2	PCHBUF1	PCHBUF0	0000 0000
0Bh	INTCON	GIE	PEIE	T0IE	INT0IE	PABCIE	T0IF	INT0IF	PABCIF	0000 0000
0Ch	PIR1	INT2IF	INT1IF	WDTIF	-	-	T5IF/CCPIF	T4IF	T1IF	000x x000
0Dh	PIR2	ADIF	LVDIF	CMPIF	-	SIMIF	LSRIF	TXIF	RXIF	000x 0010
0Eh	-	-	-	-	-	-	-	-	-	-
0Fh	TMRxL	TMRx7	TMRx6	TMRx5	TMRx4	TMRx3	TMRx2	TMRxx	TMRx0	xxxx xxxx
10h	TMRxH	-	-	-	-	-	-	TMRx9	TMRx8	xxxx xxxx
11h	TxCR1	-	-	TmxOE	VFSELx	TMx_HCK	TxOS	TxRL	TxEN	xxxx xxxx
12h	TxCR2	-	-	TxCS	TxCE	/PSxEN	PSxSEL2	PSxSEL1	PSxSEL0	xxxx xxxx
13h	BZ1CR	BZ1EN	-	-	-	BZ1FSEL3	BZ1FSEL2	BZ1FSEL1	BZ1FSEL0	0xxx1111
14h	AWUCON	WUPA7	WUPA6	WUPA5	WUPA4	WUPA3	WUPA2	WUPA1	WUPA0	0000 0000
15h	BCDWUCON	WUPBCD7	WUPBCD6	WUPBCD5	WUPBCD4	WUPBCD3	WUPBCD2	WUPBCD1	WUPBCD0	0000 0000
16h	PORTACON30	PA3C1	PA3C0	PA2C1	PA2C0	PA1C1	PA1C0	PA0C1	PA0C0	0000 0000
17h	PORTACON74	PA7C1	PA7C0	PA6C1	PA6C0	PA5C1	PA5C0	PA4C1	PA4C0	0000 0000
18h	PORTBCON30	PB3C1	PB3C0	PB2C1	PB2C0	PB1C1	PB1C0	PB0C1	PB0C0	0000 0000
19h	PORTBCON74	PB7C1	PB7C0	PB6C1	PB6C0	PB5C1	PB5C0	PB4C1	PB4C0	0000 0000
1Ah	PORTCCON30	PC3C1	PC3C0	PC2C1	PC2C0	PC1C1	PC1C0	PC0C1	PC0C0	0000 0000
1Bh	PORTCCON74	-	-	-	-	PC5C1	PC5C0	PC4C1	PC4C0	0000 0000
1Ch	-	-	-	-	-	-	-	-	-	-
1Dh	SELCON	CCPSEL1	CCPSLE0	HBTM	BCDWU_SEL1	BCDWU_SEL0	TPSEL2	TPSEL1	TPSEL0	0xxx x000
1Eh	PSxCV	PSxCV7	PSxCV6	PSxCV5	PSxCV4	PSxCV3	PSxCV2	PSxCV1	PSxCV0	1111 1111
1Fh	PWMxCON							PWMxOEN	PWMxOAL	xxxx xxxx

表 2-1 Bank0 SFR bit 映射 (接上页)

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value of POR
80h	INDF	INDF7	INDF6	INDF5	INDF4	INDF3	INDF2	INDF1	INDF0	xxxx xxxx
81h	T0MD	LCKTM0	T0EN	T0CS	T0CE	PS0WDT	PS0SEL2	PS0SEL1	PS0SEL0	0111 1111
82h	PCL	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	0000 0000
83h	STATUS	IRP	BK1	BK0	/TO	/PD	Z	DC	C	0001 1xxx
84h	FSR	FSR7	FSR6	FSR5	FSR4	FSR3	FSR2	FSR1	FSR0	xxxx xxxx
85h	IOSTA	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0	1111 1111
86h	IOSTB	IOPB7	IOPB6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0	1111 1111
87h	IOSTC	-	-	IOPC5	IOPC4	IOPC3	IOPC2	IOPC1	IOPC0	xx11 1111
88h	-	-	-	-	-	-	-	-	-	-
89h	LVDCON	-	LVOUT	-	-	LVDS3	LVDS2	LVDS1	LVDS0	xxxx 1111
8Ah	PCHBUF	-	-	-	-	PCHBUF3	PCHBUF2	PCHBUF1	PCHBUF0	0000 0000
8Bh	INTCON	GIE	PEIE	T0IE	INT0IE	PABCIE	T0IF	INT0IF	PABCIF	0000 0000
8Ch	PIE1	INT2IE	INT1IE	WDTIE	-	-	T5IE/CCPIE	T4IE	T1IE	000x x000
8Dh	PIE2	ADIE	LVDIE	CMPIE	-	SIMIE	LSRIE	TXIE	RXIE	000x 0000
8Eh	-	-	-	-	-	-	-	-	-	-
8Fh	PCON	WDTEN	-	LVDEN	-	LVREN	-	-	-	1x0x 1xxx
90h	IRCR	IR OSC358M	-	-	-	-	IRCSEL	IRF57K	IREN	0xxx x000
91h	PWM1DUTYL	PWM1 DUTY7	PWM1 DUTY6	PWM1 DUTY5	PWM1 DUTY4	PWM1 DUTY3	PWM1 DUTY2	PWM1 DUTY1	PWM1 DUTY0	xxxx xxxx
92h	PWM1DUTYH	-	-	-	-	-	-	PWM1 DUTY9	PWM1 DUTY8	xxxx xxxx
93h	PWM2DUTYL	PWM2 DUTY7	PWM2 DUTY6	PWM2 DUTY5	PWM2 DUTY4	PWM2 DUTY3	PWM2 DUTY2	PWM2 DUTY1	PWM2 DUTY0	xxxx xxxx
94h	PWM2DUTYH	-	-	-	-	-	-	PWM2 DUTY9	PWM2 DUTY8	xxxx xxxx
95h	PWM3DUTYL	PWM3 DUTY7	PWM3 DUTY6	PWM3 DUTY5	PWM3 DUTY4	PWM3 DUTY3	PWM3 DUTY2	PWM3 DUTY1	PWM3 DUTY0	xxxx xxxx
96h	PWM3DUTYH	-	-	-	-	-	-	PWM3 DUTY9	PWM3 DUTY8	xxxx xxxx
97h	PWM4DUTYL	PWM4 DUTY7	PWM4 DUTY6	PWM4 DUTY5	PWM4 DUTY4	PWM4 DUTY3	PWM4 DUTY2	PWM4 DUTY1	PWM4 DUTY0	xxxx xxxx
98h	PWM4DUTYH	-	-	-	-	-	-	PWM4 DUTY9	PWM4 DUTY8	xxxx xxxx
99h	PWM5DUTYL	PWM5 DUTY7	PWM5 DUTY6	PWM5 DUTY5	PWM5 DUTY4	PWM5 DUTY3	PWM5 DUTY2	PWM5 DUTY1	PWM5 DUTY0	xxxx xxxx
9Ah	PWM5DUTYH	-	-	-	-	-	-	PWM5 DUTY9	PWM5 DUTY8	xxxx xxxx
9Bh	INTEDG	INT2EDGE	EIS2	EIS1	EIS0	INT1G1	INT1G0	INT0G1	INT0G0	0000 0101
9Ch	CMPCON	CMPEN	BIASEN	-	-	-	-	CMPOUT	CMPOE	00xx xxxx
9Dh	CMPCR	-	RBIAS_H	RBIAS_L	CMP_INV	PS1	PS0	NS1	NS0	x000 1100
9Eh	TBHP	-	-	-	TBHP4	TBHP3	TBHP2	TBHP1	TBHP0	xxxx xxxx
9Fh	TBHD	TBHD7	TBHD6	TBHD5	TBHD4	TBHD3	TBHD2	TBHD1	TBHD0	xxxx xxxx

表 2-1 Bank1 SFR bit 映射 (接上页)

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value of POR
100h	INDF	INDF7	INDF6	INDF5	INDF4	INDF3	INDF2	INDF1	INDF0	xxxx xxxx
101h	TMR0	TMR07	TMR06	TMR05	TMR04	TMR03	TMR02	TMR01	TMR00	xxxx xxxx
102h	PCL	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	0000 0000
103h	STATUS	IRP	BK1	BK0	/TO	/PD	Z	DC	C	0001 1xxx
104h	FSR	FSR7	FSR6	FSR5	FSR4	FSR3	FSR2	FSR1	FSR0	xxxx xxxx
105h	PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	xxxx xxxx
106h	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx xxxx
107h	PORTC	-	-	PC5	PC4	PC3	PC2	PC1	PC0	xxxx xxxx
108h	-	-	-	-	-	-	-	-	-	-
109h	ADMD	ADEN	START	GCHS	CHS4	CHS3	CHS2	CHS1	CHS0	0000 0000
10Ah	PCHBUF	-	-	-	-	PCHBUF3	PCHBUF2	PCHBUF1	PCHBUF0	0000 0000
10Bh	INTCON	GIE	PEIE	T0IE	INT0IE	PABCIE	T0IF	INT0IF	PABCIF	0000 0000
10Ch	ADDL	ADDL7	ADDL6	ADDL5	ADDL4	ADDL3	ADDL2	ADDL1	ADDL0	xxxx xxxx
10Dh	ADDH	ADDH7	ADDH6	ADDH5	ADDH4	ADDH3	ADDH2	ADDH1	ADDH0	xxxx xxxx
10Eh	ADCON1	EVHENB	-	EOC	ADFM	ADCK1	ADCK0	VHS1	VHS0	0xxx 0011
10Fh	ADJMD	-	-	ADJ_SIGN	ADJ4	ADJ3	ADJ2	ADJ1	ADJ0	xx00 0000
110h	PxCON	PxCON7	PxCON6	PxCON5	PxCON4	PxCON3	PxCON2	PxCON1	PxCON0	0000 0000
111h	ADCR	PxSEL1	PxSEL0	-	-	SHCKS1	SHCKS0	ADCR1	ADCR0	00xx 1010
112h	SIMDR	SIMD7	SIMD6	SIMD5	SIMD4	SIMD3	SIMD2	SIMD1	SIMD0	xxxx xxxx
113h	SIMCR	SIMC1 (SPE)	SIMC0 (MEN)	MSTA	SSB_PAD_EN	RX_PAD_EN	TX_PADEN	RCLK_PAD_EN	UREN	0000 0000
114h	SPCR	SPIF	WCOL	-	MODF	CPOL	CKEG	SPR1	SPR0	00x0 0000
115h	MADR	MAD7	MAD6	MAD5	MAD4	MAD3	MAD2	MAD1	-	0000 000x
116h	MFDR	-	-	-	FD4	FD3	FD2	FD1	FD0	Xxx0 0000
117h	MCR	-	-	-	MTX	TXAK	-	-	-	xxx0 0xxx
118h	MSR	MCF	MAAS	MBB	MAL	-	SRW	MIF	RXAK	1000 x001
119h	DLL	DLL7	DLL6	DLL5	DLL4	DLL3	DLL2	DLL1	DLL0	0000 0000
11Ah	DLH	DLH7	DLH6	DLH5	DLH4	DLH3	DLH2	DLH1	DLH0	0000 0000
11Bh	LCR	LOOP	SBRK	PSTUCK	PEVEN	PREN	STPS	WL1	WL0	0000 0000
11Ch	LSR	-	TSRE	THRE	BRKINT	FERR	PERR	OERR	READY	X110 0000
11Dh	THR/RBR	URD7	URD6	URD5	URD4	URD3	URD2	URD1	URD0	xxxx xxxx
11Eh	PWMDBx	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	xxxx xxxx
11Fh	CCPxCON	PWMxM1	PWMxM0	FBCH1	FBCH0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	0000 0000

表 2-1 Bank2 SFR bit 映射 (接上页)

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value of POR
180h	INDF	INDF7	INDF6	INDF5	INDF4	INDF3	INDF2	INDF1	INDF0	xxxx xxxx
181h	T0MD	LCKTM0	T0EN	T0CS	T0CE	PS0WDT	PS0SEL2	PS0SEL1	PS0SEL0	0011 1111
182h	PCL	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	0000 0000
183h	STATUS	IRP	BK1	BK0	/TO	/PD	Z	DC	C	0001 1xxx
184h	FSR	FSR7	FSR6	FSR5	FSR4	FSR3	FSR2	FSR1	FSR0	xxxx xxxx
185h	IOSTA	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0	1111 1111
186h	IOSTB	IOPB7	IOPB6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0	1111 1111
187h	IOSTC	-	-	IOPC5	IOPC4	IOPC3	IOPC2	IOPC1	IOPC0	xxxx xx11
188h	-	-	-	-	-	-	-	-	-	-
189h	RFC	RFCEN				PSEL3	PSEL2	PSEL1	PSEL0	0xxx 0000
18Ah	PCHBUF					PCHBUF3	PCHBUF2	PCHBUF1	PCHBUF0	0000 0000
18Bh	INTCON	GIE	PEIE	T0IE	INT0IE	PABCIE	T0IF	INT0IF	PABCIF	0000 0000
18Ch	LCDCON0	LCDEN						LCDR1	LCDR0	0xxx xx00
18Dh	LCDCON1	P7COM	P6COM	P5COM	P4COM	P3COM	P2COM	P1COM	P0COM	0000 0000
18Eh	-	-	-	-	-	-	-	-	-	-
18Fh	-	-	-	-	-	-	-	-	-	-
190h	-	-	-	-	-	-	-	-	-	-
191h	-	-	-	-	-	-	-	-	-	-
192h	-	-	-	-	-	-	-	-	-	-
193h	-	-	-	-	-	-	-	-	-	-
194h	-	-	-	-	-	-	-	-	-	-
195h	-	-	-	-	-	-	-	-	-	-
196h	-	-	-	-	-	-	-	-	-	-
197h	-	-	-	-	-	-	-	-	-	-
198h	-	-	-	-	-	-	-	-	-	-
199h	-	-	-	-	-	-	-	-	-	-
19ah	-	-	-	-	-	-	-	-	-	-
19bh	-	-	-	-	-	-	-	-	-	-
19ch	-	-	-	-	-	-	-	-	-	-
19dh	-	-	-	-	-	-	-	-	-	-
19eh	-	-	-	-	-	-	-	-	-	-
19fh	-	-	-	-	-	-	-	-	-	-

表 2-1 Bank3 SFR bit 映射 (接上页)

## 2.2.2 数据存储寄存器

### 2.2.2.1 INDF (间接寻址寄存器)

间接寻址寄存器 (INDF) 并不是真的存在, 而是以间接寻址模式来使用。任何访问INDF的指令实际上都是访问寄存器IRP和FSR所指向的寄存器。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
INDF	0x0								INDF[7:0]	
读/写属性		读/写								
初始值		xxxxxxxx								

### 2.2.2.2 FSR (数据指针寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSR	0x4	FSR[7:0]							
读/写属性	读/写								
初始值		0	X	X	X	X	X	X	X

Bit 7 **FSR[7]**: 在间接寻址模式下, bank选择bit 0。

Bit 6:0 **FSR[6:0]**: 从指定Bank数据存储器的 128 个寄存器中选择一个。

### 2.2.2.3 TBHP (表格指针高字节寄存器)

当指令CALLA, GOTOA或TABLEA被执行时, 此目标地址由TBHP[3:0]和ACC构成。ACC是PC[11:0]的低字节, TBHP[3:0]是PC[11:0]的高字节。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBHP	0x9e	-	-	-	TBHP4	TBHP3	TBHP2	TBHP1	TBHP0
读/写属性	读/写								
初始值		X	X	X	X	X	X	X	X

### 2.2.2.4 TBHD (表格数据高字节寄存器)

当指令TABLEA被执行后, ROM表格的数据高字节内容被加载到TBHD[7:0]寄存器, ROM表格的数据低字节内容则被加载到ACC。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBHD	0x9F	TBHD7	TBHD6	TBHD5	TBHD4	TBHD3	TBHD2	TBHD1	TBHD0
读/写属性	读								
初始值		X	X	X	X	X	X	X	X

## 2.2.3 特殊功能寄存器

### 2.2.3.1 STATUS (状态寄存器)

寄存器STATUS包含算术指令的结果和引起复位的原因。

STATUS寄存器如下表所示, 包括:

- ALU的算术状态。
- 系统复位状态。
- 存储bank为数据存储器选择位。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	0x3	IRP	BK1	BK0	/TO	/PD	Z	DC	C
读/写属性	读/写								
初始值		0	X	X	1	1	X	X	X

Bit 7 **IRP**: 在间接寻址模式下, IRP和FSR[7]用来选择Bank。

IRP, FSR[7]=00, 选择bank0。

IRP, FSR[7]=01, 选择bank1。

IRP, FSR[7]=10, 选择bank2。

IRP, FSR[7]=11, 选择bank3。

Bit 6:5 **BK[1:0]:** 在直接寻址模式下, BK1 和BK0 用来选择Bank。

BK1, BK0=00, 选择bank0。

BK1, BK0=01, 选择bank1。

BK1, BK0=10, 选择bank2。

BK1, BK0=11, 选择bank3。

Bit 4 **/TO:** 看门狗超时标志位。

1 = 上电后或执行CLRWDT或SLEEP指令后。

0 = 发生WDT超时。

Bit 3 **/PD:** 睡眠模式标志位。

1 = 上电后或执行CLRWDT指令后。

0 = 执行SLEEP指令后。

Bit 2 **Z:** 零位。

1 = 逻辑运算的结果为零。

0 = 逻辑运算的结果不为零。

Bit 1 **DC:** 半进位/半借位标志位。

1 = 加法运算低四位有进位或减法运算时没有向高四位借位。

0 = 加法运算低四位无进位或减法运算时有向高四位借位。

Bit 0 **C:** 进位/借位标志位。

1 = 加法运算有进位或减法运算无借位。

0 = 加法运算无进位或减法运算有借位。

(\*1) 能被SLEEP指令清零。

(\*2) 能被CLRWDT指令设置。

### 2.2.3.2 SELCON (选择控制寄存器)

这个寄存器包含映射虚拟SFR和物理SFR的控制信号，以节省SFR空间。虚拟SFR和物理SFR的关系如下图所示：SFRx为虚拟寄存器，SFR0 / SFR1为物理寄存器。当SELCON = 0 时，读取或写入SFRx将实际访问SFR0。

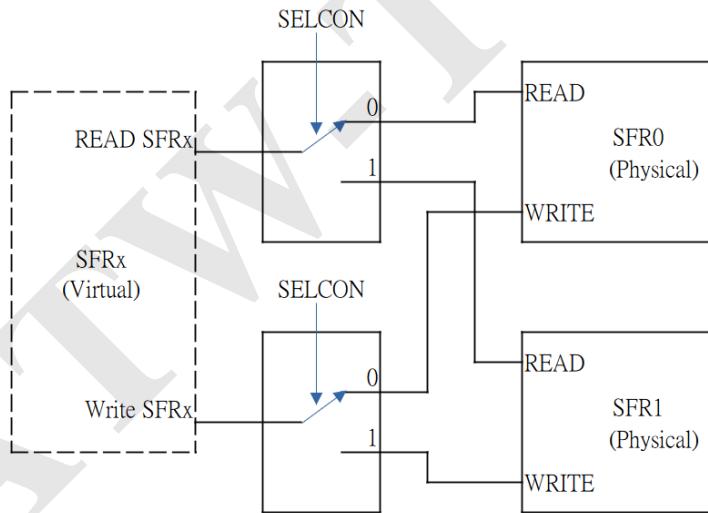


图 4 虚拟SFR的操作

当SELCON=1 时，读取或写入SFRx将实际访问SFR1。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SELCON	0x1D	CCPSEL	-	-	BCDWUSEL[1:0]		TPSEL[2:0]		
读/写属性		读/写	-	-	读/写		读/写		
初始值(note*)		0	X	X	00		000		

Bit 7:6 **CCPSEL:** 虚拟SFR CCPxCON 和 PWMDBx的映射。

0 = CCPxCON和PWMDBx映射到CCP1CON和PWMDB1。 (初始值)

1 = CCPxCON和PWMDBx映射到CCP2CON和PWMDB2。

Bit 5 未生效的。

Bit 4:3 **BCDWUSEL:** 虚拟SFR BCDWUCON 映射。

00 = BCDWUCON 映射到BWUCON。 (初始值)

01 = BCDWUCON 映射到CWUCON。

Bit 2:0 **TPSEL:** 虚拟SFR TMRxL/TMRxH/TxCR1/TxCR2/PWMxCON映射。

001 = TMRxL / TMRxH / TxCR1 / TxCR2 / PWMxCON映射到TMR1L / TMR1H / T1CR1 / T1CR2 / PWM1CON (初始值)

010 = PWMxCON 映射到 PWM2CON

011 = PWMxCON 映射到 PWM3CON

100 = TMRxL / TMRxH / TxCR1 / TxCR2 / PWMxCON 映射到 TMR4L / TMR4H / T4CR1 / T4CR2 / PWM4CON

101 = TMRxL / TMRxH / TxCR1 / TxCR2 / PWMxCON 映射到 TMR5L / TMR5H / T5CR1 / T5CR2 / PWM5CON

SELCON	虚拟SFR	SELCON值	物理SFR
CCPSEL	CCPxCON	0	CCP1CON
		1	CCP2CON
	PWMDBx	0	PWMDB1
		1	PWMDB2
BCDWUSEL	BCDWUCON	00	BWUCON
		01	CWUCON
TPSEL	TMRxL	001	TRM1L
		100	TMR4L
		101	TMR5L
	TMRxH	001	TMR1H
		100	TMR4H
		101	TMR5H
	TxCR1	001	T1CR1
		100	T4CR1
		101	T5CR1
	TxCR2	001	T1CR2
		100	T4CR2
		101	T5CR2
	PWMxCON	001	PWM1CON
		010	PWM2CON
		011	PWM3CON
		100	PWM4CON
		101	PWM5CON

## 2.3 累加器 (ACC)

### 2.3.1 概述

ALU是一个 8 位宽的算术逻辑单元，执行MCU中的所有算术和逻辑操作。它可以对数据进行加法、减法、移位和逻辑运算。ALU还控制状态标志（在STATUS寄存器中），状态标志反映操作的结果。

ACC寄存器是一个 8 位寄存器，用于存储ALU操作的结果。它不是数据内存的一部分，而是位于CPU内，专用于ALU在计算期间使用。因此，它不能直接被寻址，只能通过提供的特定指令来访问。

### 2.3.2 ACC 应用

; 用直接寻址方式写数据内存

MOVR 0x20,ACC ; 将GPR 0x20 数据移至ACC

MOVAR STATUS ; 将ACC数据写入STATUS

### 3. I/O Ports

这些I/O端口的一些引脚与设备上的外围设备特性的备用功能进行多路复用。一般来说，当外围设备启用时，该引脚可能不会用作通用I/O引脚。

#### 3.1 概述

AT8BM84A提供3组I/O端口：分别是PORTA，PORTB和PORTC，PORTA是一个8位宽的双向端口。相应的数据方向寄存器是IOSTA。PORTB是一个8位宽的双向端口。相应的数据方向寄存器是IOSTB。PORTC是一个6位宽的双向端口。相应的数据方向寄存器是IOSTC。

当一个I/O引脚配置为输入引脚时，它可能有Pull-High电阻或Pull-down电阻。当一个I/O引脚配置为输出引脚时，它可以是Push-Pull输出或Open-drain输出。每个I/O引脚有2位SFR寄存器，即PA0C[1:0]~PA7C[1:0]，PB0C[1:0]~PB7C[1:0]和PC0C[1:0]~PC5C[1:0]，用于设置这些属性。当I/O引脚处于输入模式时，这个2位寄存器用于设置Pull-High或Pull-down属性。当I/O引脚处于输出模式时，这些2位寄存器用于设置Push-Pull或Open-drain属性，更详细的描述见下表：

2位SFR寄存器属性	输入	输出
00	Floating	Push-Pull
01	Pull-Down	Push-Pull
10	Pull-High	Open-Drain
11	-	Open-Drain

表 3-1 I/O 端口特性总结

根据配置字设置，每个引脚的输入电流（sink current）可以是一般灌电流（Normal）（25mA@VDD=5V），大灌电流（large）（40mA@VDD=5V）或极大灌电流（ultra-large）（80mA @VDD =5V）。当I/O配置为输出口，根据配置字设置，每个引脚的驱动电流（drive current）可以是一般输出电流（normal）（15mA @V<sub>DD</sub> =5V）或大电流（large）（30mA @VDD =5V）。PA4~PA7进一步提供超大驱动电流（45mA@VDD =5V）。

查看下表的Drive/sink电流模式设置：

Drive Mode	Sink Mode	Note
Normal	Normal	All Pin
Large	Large	All Pin
Large	Ultra-Large	PA[3:0], PB[7:0], PC[5:0]
Ultra-Large	Ultra-Large	PA[7:4] Only

表 3-2 Drive 和 Sink 电流模式

### 3.1.1 IO 引脚结构框图

OUT\_EN: 设定引脚属性为输出。  
WR\_DATA: 将数据写入引脚。  
RD\_DATA: 读取引脚状态。  
RD\_TYPE: 选择读取引脚或数据锁存器。  
PULLUP\_ENB: 开启内部  $100\text{K}\Omega$  上拉电阻。  
PULLDOWN\_EN: 开启内部下拉电阻。  
OD\_EN: 设定引脚为开漏极模式。  
WU\_RN: 开启引脚变化中断。  
PADIF: 引脚变化中断标志。  
ADCH\_SEL: 开启引脚到ADC通道输入。  
CMPCH\_SEL: 开启引脚到比较器通道输入。

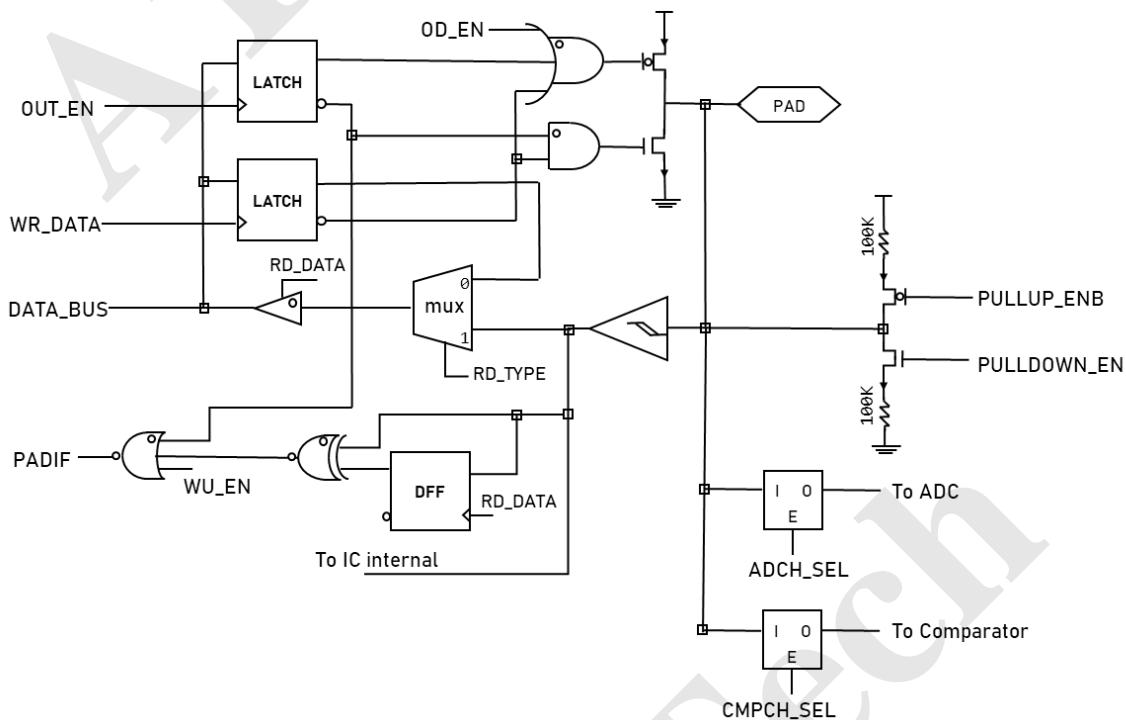


图 5 PA0~PA4, PB3 引脚结构框图

**OUT\_EN:** 设定引脚属性为输出。

**WR\_DATA:** 将数据写入引脚。

RD\_DATA: 读取引脚状态。

**RD\_TYPE:** 选择读取引脚或数据锁存器。

PULLUP\_ENB: 开启内部  $100\text{K}\Omega$  上拉串

PULLDOWN\_EN: 开启内部下拉

OD EN: 设定引脚为开漏极模式

WU RN: 开启引脚变化中断。

**PADIF:** 引脚变化中断标志。

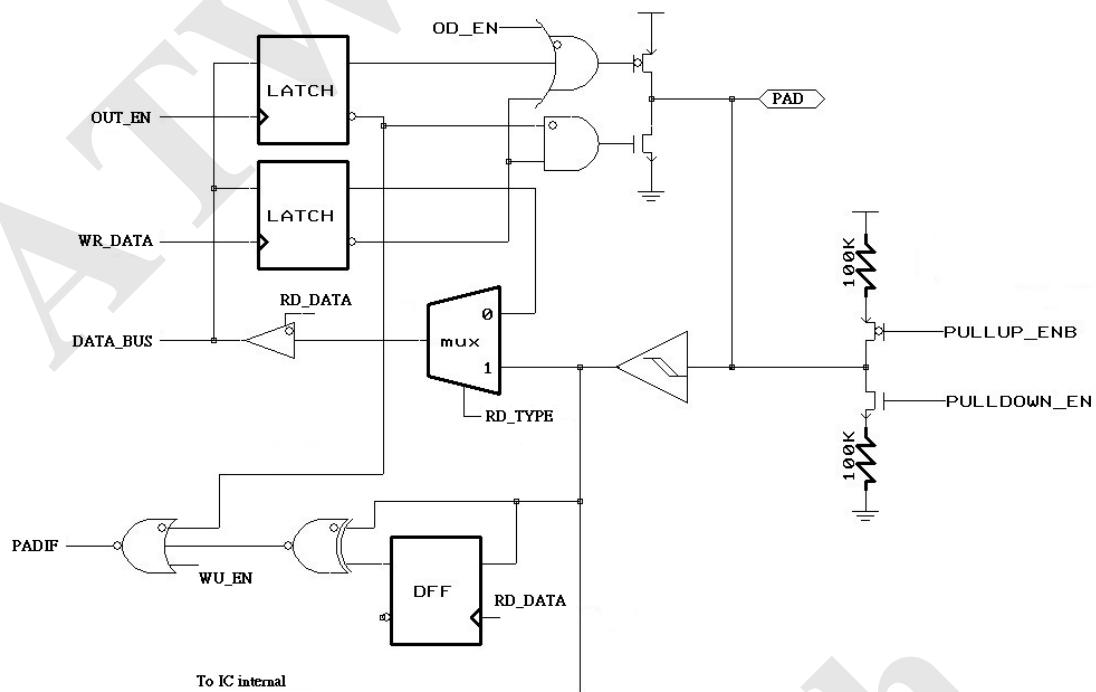


图 6 PA5 引脚结构框图

OUT\_EN: 设定引脚属性为输出。  
WR\_DATA: 将数据写入引脚。  
RD\_DATA: 读取引脚状态。  
RD\_TYPE: 选择读取引脚或数据锁存器。  
PULLUP\_ENB: 开启内部  $100\text{K}\Omega$  上拉电阻。  
PULLDOWN\_EN: 开启内部下拉电阻。  
OD\_EN: 设定引脚为开漏极模式。  
WU\_RN: 开启引脚变化中断。  
PADIF: 引脚变化中断标志。  
ADCH\_SEL: 开启引脚到ADC通道输入。  
XTL\_EN: 开启引脚作为XTAL引脚。

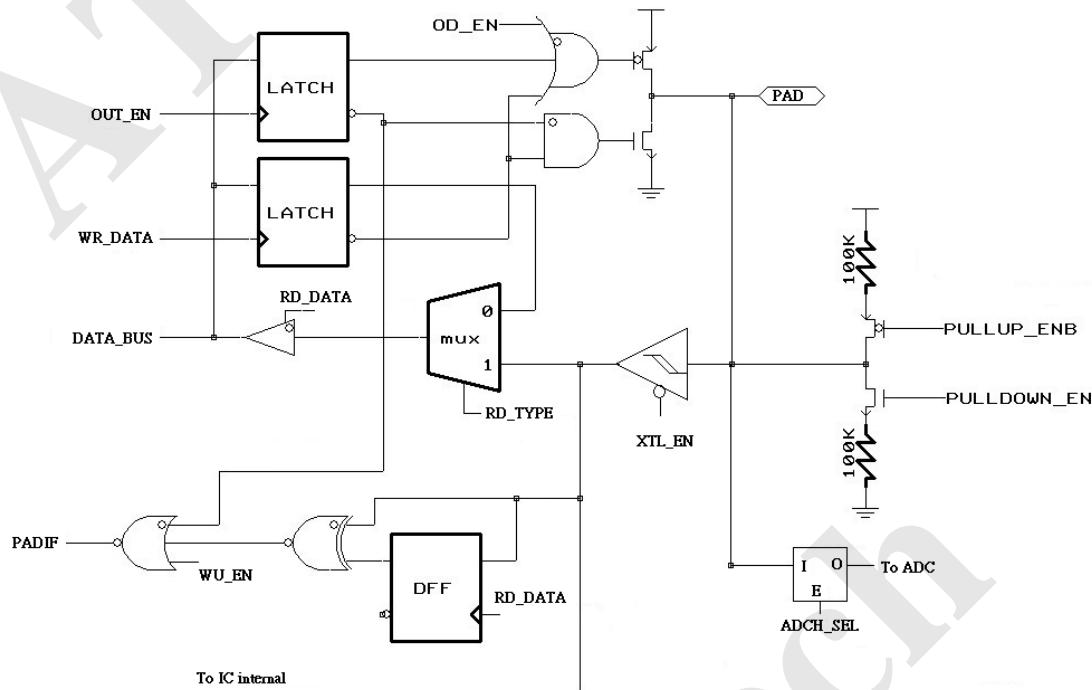


图 7 PA6, PA7 引脚结构框图

OUT\_EN: 设定引脚属性为输出。  
WR\_DATA: 将数据写入引脚。  
RD\_DATA: 读取引脚状态。  
RD\_TYPE: 选择读取引脚或数据锁存器。  
PULLUP\_ENB: 开启内部  $100\text{K}\Omega$  上拉电阻。  
PULLDOWN\_EN: 开启内部下拉电阻。  
OD\_EN: 设定引脚为开漏极模式。  
WU\_RN: 开启引脚变化中断。  
PADIF: 引脚变化中断标志。  
ADCH\_SEL: 开启引脚到ADC通道输入。  
COM\_EN: 开启PAD到  $1/2 \text{ VDD}$  电压输出。  
VDD\*0.5:  $1/2 \text{ VDD}$  电压。

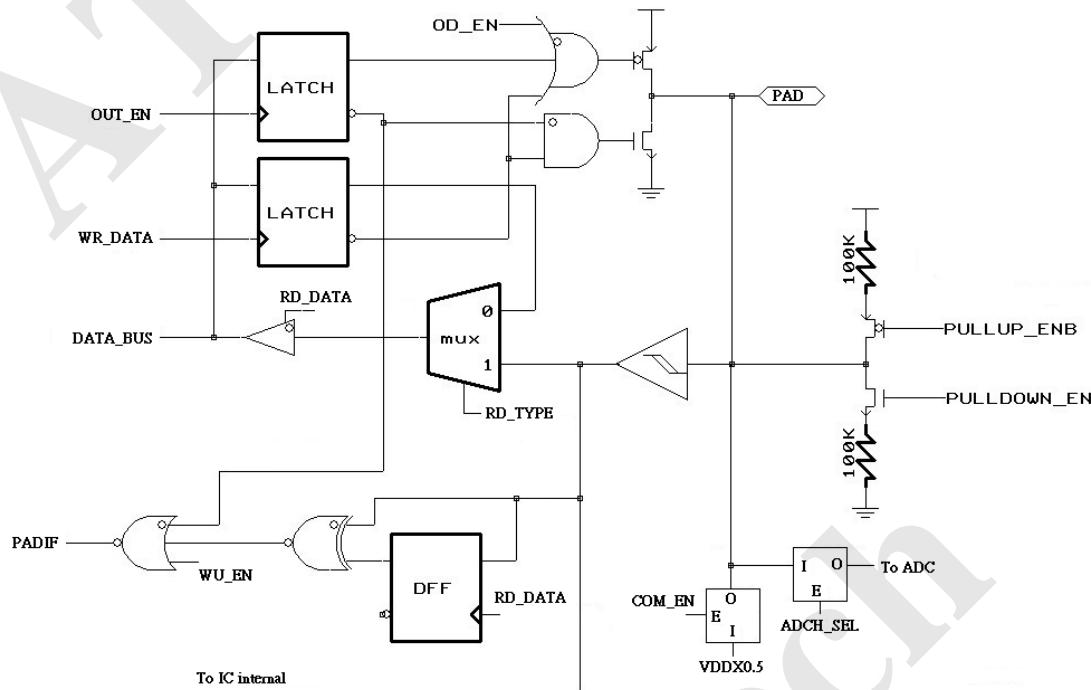


图 8 PB4~PB7, PC2~PC5 引脚结构框图

OUT\_EN: 设定引脚属性为输出。  
 WR\_DATA: 将数据写入引脚。  
 RD\_DATA: 读取引脚状态。  
 RD\_TYPE: 选择读取引脚或数据锁存器。  
 PULLUP\_ENB: 开启内部  $100\text{K}\Omega$  上拉电阻。  
 PULLDOWN\_EN: 开启内部下拉电阻。  
 OD\_EN: 设定引脚为开漏极模式。  
 WU\_RN: 开启引脚变化中断。  
 PADIF: 引脚变化中断标志。  
 ADCH\_SEL: 开启引脚到ADC通道输入。

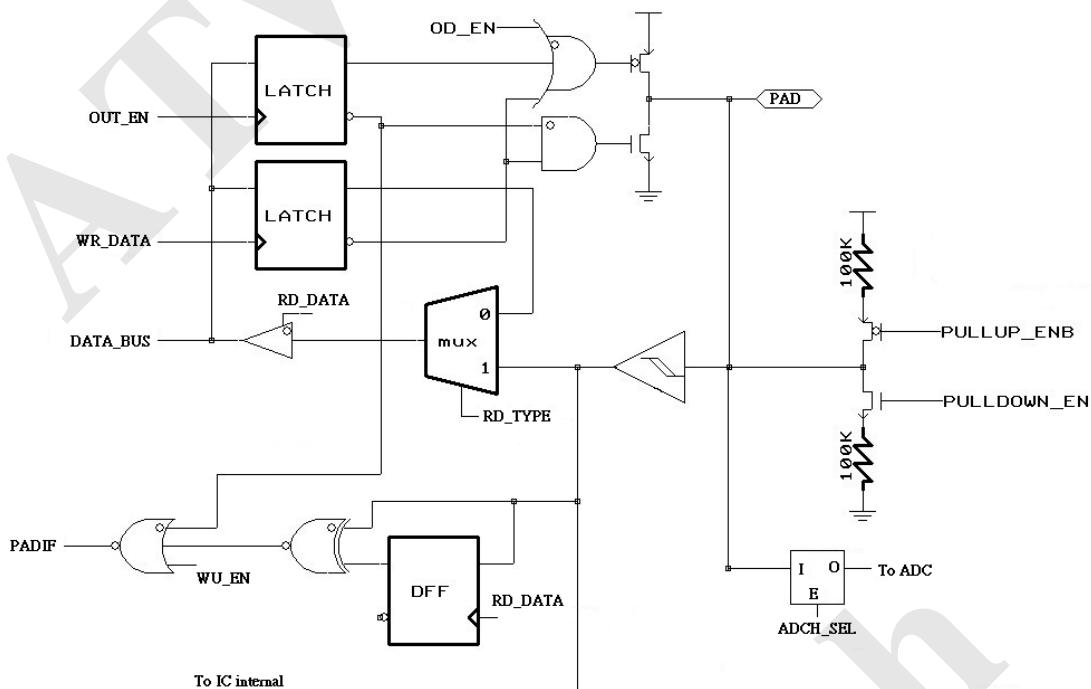


图 9 PB0~PB2, PC0~1 引脚结构框图

### 3.2 I/O端口寄存器

#### 3.2.1 AWUCON (PortA 唤醒控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AWUCON	0x14	WUPA7	WUPA6	WUPA5	WUPA4	WUPA3	WUPA2	WUPA1	WUPA0
读/写属性		读/写							
初始值		0	0	0	0	0	0	0	0

Bit 7:0      **WUPAx**: 开启/关闭 PAx 唤醒功能,  $0 \leq x \leq 7$ 。

1 = 开启 PAx 唤醒功能。

0 = 关闭 PAx 唤醒功能。

### 3.2.2 BCDWUCON (PortB/C/D 唤醒控制寄存器)

根据BCDWUSEL[1:0]的值，访问虚拟SFR BCDWUCON相当于访问物理SFR BWUCON或CWUCON。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BCDWUCON	0x15	WUPBCD7	WUPBCD6	WUPBCD5	WUPBCD4	WUPBCD3	WUPBCD2	WUPBCD1	WUPBCD0
读/写属性		读/写							
初始值		0	0	0	0	0	0	0	0

Bit 7:0 **WUPBCDx**: 开启/关闭 PB<sub>x</sub> 或 PC<sub>x</sub> 唤醒功能, 0 ≤ x ≤ 7

对于 BCDWUSEL[1:0]=00

1 = 开启 PB<sub>x</sub> 唤醒功能。

0 = 关闭 PB<sub>x</sub> 唤醒功能。

对于 BCDWUSEL[1:0]=01

1 = 开启 PC<sub>x</sub> 唤醒功能。

0 = 关闭 PC<sub>x</sub> 唤醒功能。

### 3.2.3 PORTACON30 / PORTACON74 / PORTBCON30 / PORTBCON74 / PORTCCON30 / PORTCCON74 (端口属性控制寄存器)

如果IO脚位为输入时，这些寄存器用于设置对应IO脚位Pull-High或Pull-down属性。如果IO脚位为输出时，这些寄存器用于设置对应IO脚位Push-Pull或Open-drain属性。具体见下表。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTACON30	0x16	PA3C[1:0]		PA2C[1:0]		PA1C[1:0]		PA0C[1:0]	
读/写属性		读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值		0	0	0	0	0	0	0	0

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTACON74	0x17	PA7C[1:0]		PA6C[1:0]		PA5C[1:0]		PA4C[1:0]	
读/写属性		读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值		0	0	0	0	0	0	0	0

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTBCON30	0x18	PB3C[1:0]		PB2C[1:0]		PB1C[1:0]		PB0C[1:0]	
读/写属性		读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值		0	0	0	0	0	0	0	0

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTBCON74	0x19	PB7C[1:0]		PB6C[1:0]		PB5C[1:0]		PB4C[1:0]	
读/写属性		读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值		0	0	0	0	0	0	0	0

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTCCON30	0x1A	PC3C[1:0]		PC2C[1:0]		PC1C[1:0]		PC0C[1:0]	
读/写属性		读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值		0	0	0	0	0	0	0	0

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTCCON74	0x1B	-		-		PC5C[1:0]		PC4C[1:0]	
读/写属性		-	-	-	-	读/写	读/写	读/写	读/写
初始值		X	X	X	X	0	0	0	0

Bit 7:0

**PAxC[1:0]: PORTA 属性设置,  $0 \leq x \leq 7$ 。**

PAxC[1:0]	(IOPAx=1) input	(IOPAx=0) output
00	Floating	Push-Pull
01	Pull-Down	Push-Pull
10	Pull-High	Open-Drain
11	-	Open-Drain

**PBxC[1:0]: PORTB 属性设置,  $0 \leq x \leq 7$ 。**

PBxC[1:0]	(IOPBx=1) input	(IOPBx=0) output
00	Floating	Push-Pull
01	Pull-Down	Push-Pull
10	Pull-High	Open-Drain
11	-	Open-Drain

**PCxC[1:0]: PORTC 属性设置,  $0 \leq x \leq 5$ 。**

PCxC[1:0]	(IOPCx=1) input	(IOPCx=0) output
00	Floating	Push-Pull
01	Pull-Down	Push-Pull
10	Pull-High	Open-Drain
11	-	Open-Drain

### 3.2.4 PORTA (PORTA 数据寄存器)

读取PortA时, 若特定引脚被配置为输入引脚, 将得到该引脚输入状态。然而, 若该引脚被配置为输出引脚, 依据配置选项RD\_OPT, 得到该引脚的状态或相对应的输出数据锁存值。当写入PORTA时, 数据被写入PA口的输出数据锁存器中。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA	0x5	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
读/写属性	读/写								
初始值	xxxxxxxx								

### 3.2.5 PORTB (PORTB 数据寄存器)

读取PORTB时, 若特定引脚被配置为输入引脚, 将得到该引脚输入状态。然而, 若该引脚被配置为输出引脚, 依据配置选项RD\_OPT, 得到该引脚的状态或相对应的输出数据锁存值。当写入PORTB时, 数据被写入PORTB的输出数据锁存器中。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTB	0x6	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
读/写属性	读/写								
初始值	XXXXXXXX								

### 3.2.6 PORTC (PORTC 数据寄存器)

读取PORTC时, 若特定引脚被配置为输入引脚, 将得到该引脚输入状态。然而, 若该引脚被配置为输出引脚, 依据配置选项RD\_OPT, 得到该引脚的状态或相对应的输出数据锁存值。当写入PORTC时, 数据被写入PORTC的输出数据锁存器中。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTC	0x7	-	-	PC5	PC4	PC3	PC2	PC1	PC0
读/写属性	读/写								
初始值	XXXXXXXX								

### 3.2.7 IOSTA (PORTA I/O 控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOSTA	0x85	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0
读/写属性	读/写								
初始值	1 1 1 1 1 1 1 1 1								

Bit 7:0      **IOPAx**: PAx I/O 模式选择,  $0 \leq x \leq 7$ 。

1 = PAx 为输入模式。

0 = PAx 为输出模式。

### 3.2.8 IOSTB (PORTB I/O 控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOSTB	0x86	IOPB7	IOPB6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0
读/写属性	读/写								
初始值	1 1 1 1 1 1 1 1 1								

Bit 7:0      **IOPBx**: PBx I/O 模式选择,  $0 \leq x \leq 7$ 。

1 = PBx 为输入模式。

0 = PBx 为输出模式。

## 3.2.9 IOSTC (PORTC I/O 控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOSTC	0x87	-	-	IOPC5	IOPC4	IOPC3	IOPC2	IOPC1	IOPC0
读/写属性	-	-	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值	x	x	1	1	1	1	1	1	1

Bit 7:6 未生效的。

Bit 5:0 **IOPCx**: PCx I/O 模式选择,  $0 \leq x \leq 5$ 。

1 = PCx 为输入模式。

0 = PCx 为输出模式。

## 3.2.10 PxCON (端口模拟引脚控制寄存器)

根据 PxSEL[1:0] 的值, 访问虚拟 SFR PxCON 相当于访问物理 SFR PACON、PBCON 或 PCCON。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxCON	0x110	PxCON7	PxCON6	PxCON5	PxCON4	PxCON3	PxCON2	PxCON1	PxCON0
读/写属性	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值	0	0	0	0	0	0	0	0	0

当 PxSEL=00, PxCON=PACON,

Bit 7:0 **PACON[7:0]**: PORTA 模拟引脚选择。

1 = PAi 为纯模拟引脚, 为了省电关闭输入缓存,  $0 \leq i \leq 7$ 。

0 = PAi 可做模拟或数字引脚,  $0 \leq i \leq 7$ 。

当 PxSEL=01, PxCON=PBCON,

Bit 7:0 **PBCON[7:0]**: PORTB 模拟引脚选择。

1 = PBi 为纯模拟引脚, 为了省电关闭输入缓存,  $0 \leq i \leq 7$ 。

0 = PBi 可做模拟或数字引脚,  $0 \leq i \leq 7$ 。

当 PxSEL=10, PxCON=PCCON,

Bit 7:0 **PCCON[7:0]**: PORTC 模拟引脚选择。

1 = PCi 为纯模拟引脚, 为了省电关闭输入缓存,  $0 \leq i \leq 5$ 。

0 = 可做模拟或数字引脚,  $0 \leq i \leq 5$ 。

## 4. 定时器 0 (Timer0)

### 4.1 概述

定时器 0 是 8 位上数定时器，由寄存器 T0EN (T0MD[6]) 开启/关闭。写入定时器 0 将会设定其初始值，读取定时器 0 时则会显示目前的计数数值。

定时器 0 的时钟源可由寄存器 T0CS (T0MD[5]) 与 LCK\_TM0 (T0MD[7]) 所决定，可以来自指令时钟、外部时钟输入引脚 EX\_CKIO 或低速时钟。当 T0CS 为 0，指令时钟会被选择当作定时器 0 时钟源。当 T0CS 为 1 且 LCK\_TM0 为 0，EX\_CKIO 会被当作定时器 0 时钟源。当 T0CS 为 1 且 LCK\_TM0 为 1 (并且定时器 0 必须设置为 1)，会选择低频振荡 I\_LRC / E\_LXT 当作定时器 0 时钟源 (I\_LRC 或 E\_LXT，取决于配置字)。

汇总成表格如下：

定时器 0 时钟源	T0CS	LCKTM0	定时器 0 来源	低频振荡
指令时钟	0	X	X	X
EX_CKIO	1	0	X	X
		X	0	
E_LXT	1	1	1	1
I_LRC	1	1	1	0

表 4-1 定时器 0 时钟源控制摘要

寄存器 T0CE (T0MD[4]) 可决定 EX\_CKIO 引脚或 I\_LRC 的时钟触发沿选择。当 T0CE 为 1 时，EX\_CKIO 引脚或 I\_LRC 的下降沿信号将让定时器 0 计数加一。当 T0CE 为 0，EX\_CKIO 引脚或 I\_LRC 的上升沿信号将让定时器 0 计数加一。当使用 I\_LRC 作为定时器 0 的时钟源时，建议使用预分频器 0 并且分频比设置为 4 以上，否则可能会使计数丢失。 **预分频器 0 的频率至少要比  $F_{INST}$  的频率慢 2 倍。**

在向 Timer0 提供时钟源之前，如果寄存器 PS0WDT (T0MD[3]) 为 0，Timer0 时钟源可以由预分频器 0 所分频，预分频器 0 会被指定到 Timer0，且会在 PS0WDT 设为 0 时清除 Timer0 与预分频器 0。寄存器 PS0SEL[2:0] 决定预分频器 0 的预分频比，其数值从 1:2 到 1:256。

当定时器 0 上溢时，寄存器 T0IF (INTCON[2]) 将会设定为 1，以表明定时器 0 发生上溢中断。如果寄存器 T0IE (INTCON[5]) 与 GIE 都设定为 1，会发生中断的请求并执行中断服务程序。直到程序写入 0 到 T0IF，T0IF 才会被清除。

定时器 0 与 WDT 的结构框图如下图：

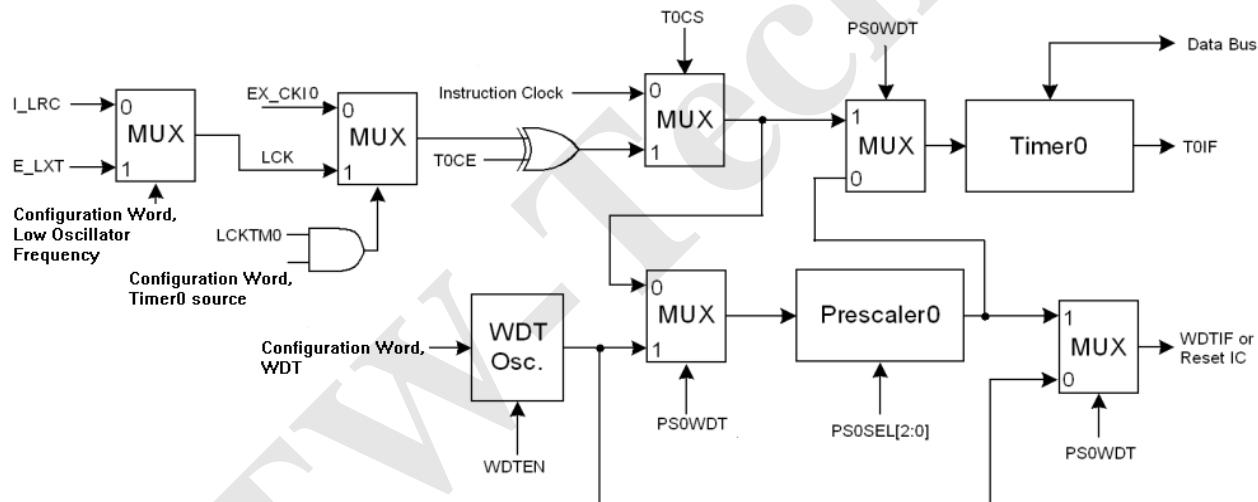


图 10 Timer0 和 WDT结构框图

## 4.2 定时器 0 控制寄存器

### 4.2.1 T0MD 寄存器

T0MD是一个可读/可写寄存器，它只能被指令T0MD / T0MDR访问。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T0MD	0x81	LCKTM0	T0EN	T0CS	T0CE	PS0WDT	PS0SEL[2:0]		
读/写属性	读/写								
初始值(note*)	0	1	1	1	1	111			

当T0CS=0 时，选择指令时钟 $F_{INST}$  作为定时器 0 时钟源。

Bit 7 **LCKTM0**: 定时器 0 时钟源选择。

当T0CS=1 时，定时器 0 时钟源可选择为低频振荡。

1 = 低振荡器频率(I\_LRC或E\_LXT，取决于配置字低振荡器频率)输出取代引脚EX\_CKIO 作为定时器 0 时钟源。

0 = .选择引脚EX\_CKIO 上的外部时钟作为定时器 0 时钟源。

**注意：有关定时器 0 时钟源选择的详细说明，请参考定时器 0 章节。**

Bit 6 **T0EN**: 开启/关闭定时器 0。

1 = 开启定时器 0。

0 = 关闭定时器 0。

Bit 5 **T0CS**: 定时器 0 时钟源选择。

1 = 选择EX\_CKIO 引脚或低频振荡I\_LRC / E\_LXT。

0 = 选择指令时钟 $F_{INST}$ 。

Bit 4 **T0CE**: 定时器 0 外部时钟源触发沿选择。

1 = EX\_CKIO 发生上升沿信号时定时器 0 加一。

0 = EX\_CKIO 发生下降沿信号时定时器 0 加一。

**注意: T0CE应用在低频振荡作为定时器 0 时钟源条件。**

Bit 3 **PS0WDT:** 预分频器 0 分配选择。

1 = 预分频器 0 被分配到WDT。

0 = 预分频器 0 被分配到定时器 0。

**注意: 在使能看门狗或定时器0中断前, 要先设定PS0WDT和PS0SEL[2:0], 否则复位或中断可能导致错误触发。**

Bit 2:0 **PS0SEL[2:0]:** 选择预分频器 0 的预分频比 (Dividing Rate)。由预分频器 0 决定的预分频比被分配给定时器 0 或 WDT。当预分频器 0 被分配给WDT, 预分频比取决于选择哪种超时机制。

PS0SEL[2:0]	预分频比选项		
	PS0WDT=0 (Timer0)	PS0WDT=1 (WDT Reset)	PS0WDT=1 (WDT Interrupt)
000	1:2	1:1	1:2
001	1:4	1:2	1:4
010	1:8	1:4	1:8
011	1:16	1:8	1:16
100	1:32	1:16	1:32
101	1:64	1:32	1:64
110	1:128	1:64	1:128
111	1:256	1:128	1:256

#### 4.2.2 TMR0 (定时器 0 寄存器)

当读取TMR0 寄存器时, 会得到定时器 0 目前计数数值。

当写入TMR0 寄存器时, 会改变定时器 0 目前计数数值。

通过设置OPTION寄存器与配置字 (Configuration Word) , 定时器 0 时钟源可以从指令时钟FINST、外部时钟EX\_CKIO 或低频振荡器I\_LRC/E\_LXT中选择一个。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR0	0x1	TMR0[7:0]							
读/写属性		读/写							
初始值		xxxxxxxx							

## 5. 定时器 1 / 定时器 4 / 定时器 5

### 5.1 概述

定时器 1、定时器 4 和定时器 5 具有相似的结构。它们都是带有分频器的 10 位向下计数定时器，分频比可编程。但它们在其他方面也有所不同。下表列出了定时器 1、定时器 4 和定时器 5 功能的比较。

Timer0 clock source	Timer1	Timer4	Timer5	Note
计数器	10-bit			
向上或向下计数	向下计数			
计数模式	单次计数模式和连续计数模式			
重载模式	重载 0x3ff 或寄存器			
预分频器分频比	8 个选择			
高速振荡时钟输入	是			
外部时钟	EX_CK10	EX_CK11	EX+CK11	上升/下降沿
PWM (用于定时器框架)	PWM1/2/3	PWM4	PWM5	
定时器切换输出	是	否	否	
Buzzer (频率源)	是	否	否	
CCP1 比较定时器	否	低字节	高字节	
CCP1 捕捉定时器	否	低字节	高字节	
CCP1 全桥	否	否	是	
CCP1 半桥	否	否	是	
CCP2 半桥	否	是	否	死区控制
RFC	是	否	否	

表 5-1 定时器 0 时钟源控制摘要

定时器 1/4/5 操作由虚拟SFR TMRxL / TMRxH / TxCR1 / TxCR2 和TPSEL[2:0]控制。TPSEL[2:0]用于控制这些虚拟SFR到物理存在的定时器SFR的映射。例如，当TPSEL[2:0]=4 时，虚拟SFR TMRxL / TMRxH / TxCR1 / TxCR2 / TxPSC将映射到SFR TMR4L / TMR4H / T4CR1 / T4CR2 / T4PSC。这意味着对这些虚拟SFR的访问将实际访问定时器 4 的SFR。对于AT8BM84A，TPSEL可以是 1、4 或 5，分别对应定时器 1、定时器 4 和定时器 5。

**注意：在本节中，通用引用用于寄存器和位名称相同，除了一个“x”变量表示该项与特定定时器模块的关联。为了清晰起见，所有的模块操作描述都是通用的，并且同样适用于所有的定时模块。如有必要时，计时器之间的差异会指出。**

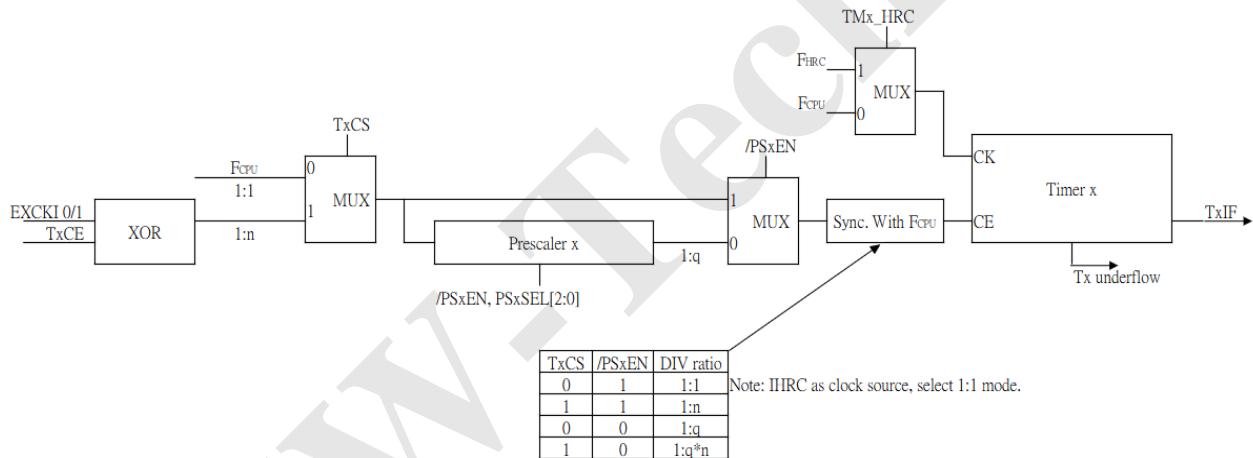


图 11 定时器 1/4/5 的结构框图

定时器  $x$  的时钟源可以通过  $\text{TMx\_HRC}$  ( $\text{TxCR1[3]}$ ) 选择。当  $\text{TMx\_HRC}=0$  时，指令时钟 ( $\text{F}_{\text{CPU}}$ ) 被选择作为定时器  $x$  的时钟源，否则 HRC 时钟输出 ( $\text{F}_{\text{HRC}}$ ) 被选择作为定时器  $x$ 。

定时器  $x$  时钟源分频比由  $\text{TxCS}$  和预分频器  $x$  选择，如上图所示。当  $\text{TxCS}=0$  时，基准分频比为 1。当  $\text{TxCS}=1$  时，基准分频比为外部时钟 (EXCKI0 或 EXCKI1) 周期除以 CPU 周期 ( $\text{F}_{\text{CPU}}$ )。当  $/\text{PSxEN}=0$  时，预分频器  $x$  开启，由寄存器  $\text{PS1SEL}[2:0]$  ( $\text{T1CR2}[2:0]$ ) 决定的预分频器  $x$  分频比为  $1:2 \sim 1:256$ 。当  $/\text{PSxEN}=1$  时，分频器  $x$  关闭，分频器  $x$  分频比为 1:1。最终的定时器  $x$  时钟源分频比是预分频比乘以基准分频比。需要注意的是，由于 HRC 时钟 ( $\text{F}_{\text{HRC}}$ ) 的频率高于 CPU 时钟 ( $\text{F}_{\text{CPU}}$ ) 的频率，当定时器  $x$  时钟源为 HRC 时钟输出 ( $\text{F}_{\text{HRC}}$ ) 时，最终的定时器  $x$  时钟源分频比必须为 1，即设置  $\text{TxCS}=0$ ,  $/\text{PSxEN}=1$ 。

注意，定时器 1 和定时器 4/定时器 5 的外部时钟输入是不同的。定时器 1 的外部时钟是 EXCKI0 (PA4)，定时器 4 和定时器 5 的外部时钟是 EXCKI1 (PA1 或 PA2 可选)。当选择 EXCKI0 或 EXCKI1 作为定时器时钟源时，使定时器  $x$  递减的有效沿由寄存器  $\text{TxCE}$  ( $\text{TxCR2[4]}$ ) 决定。当  $\text{TxCE}$  为 1 时， $\text{EXCKIx}$  上的下降沿信号使定时器  $x$  递减。当  $\text{TxCE}$  为 0 时， $\text{EXCKIx}$  上的上升沿信号使定时器  $x$  递减。

定时器的操作可以通过寄存器  $\text{TxEN}$  ( $\text{TxCR1[0]}$ ) 来开启或关闭。

10 位定时器的 MSB 2 位来自  $\text{TMRxH}[1:0]$ ，而 LSB 8 位来自  $\text{TMRxL}[7:0]$ 。

当读取寄存器  $\text{TMRxH}$  时，它将得到 10 位下数定时器  $x$  的当前 MSB 值，即  $\text{TMRx}[9:8]$ 。当读取寄存器  $\text{TMRxL}$  时，会得到 10 位下数定时器  $x$  的当前 LSB 值，即  $\text{TMRx}[7:0]$ 。当写入寄存器  $\text{TMRxH}$  时，它将数据写入  $\text{TMRx}[9:8]$  重新加载寄存器，即  $\text{TMRxLD}[9:8]$ 。当写入寄存器  $\text{TMRxL}$  时，它将数据写入  $\text{TMRx}[7:0]$  重新加载寄存器，即  $\text{TMRxLD}[7:0]$ 。

定时器提供两种工作模式：一种是单次计数模式 (One-Shot mode)，另一种是连续计数模式 (Non-Stop mode)。当寄存器  $\text{TxOS}$  ( $\text{TxCR1[2]}$ ) 为 1 时，选择单次计数模式。从寄存器  $\text{TMRx}[9:0]$  中的初始值到 0x00，定时器  $x$  将倒数一次，即发生了下溢。当寄存器  $\text{TxOS}$  ( $\text{TxCR1[2]}$ ) 为 0 时，选择连续计数模式。当发生下溢时，有两个选择用来开始下一个下数，这是由寄存器  $\text{TxRL}$  ( $\text{TxCR1[1]}$ ) 决定的。当  $\text{TxRL}$  为 1 时，定时器处于自动加载模式，存储在寄存器  $\text{TMRxLD}[9:0]$  上的初始值将被恢复，并从这个初始值开始下一个下数。当  $\text{TxRL}$  为 0 (连续计数模式) 时，定时器  $x$  将从 0x3FF 开始下一个下数。

定时器  $x$  在自动加载模式下 ( $\text{TxOS}=0$ ,  $\text{TxRL}=1$ ) 采用双缓冲机制：当  $\text{TxEN}=1$  时，定时器  $x$  加载寄存器  $\text{TMRxLD}[9:0]$  不会上传到定时器 1 计数器 ( $\text{TMRx}[9:0]$ )，直到定时器  $x$  溢出。然而，当  $\text{TxEN}=0$  时，无论定

时器 x 是在单次模式，自动加载模式还是连续模式，在写TMRxL动作发生后，定时器 x 重新加载寄存器 TMRxLD[9:0]将立即上传到定时器 x 计数器 (TMRx[9:0])。

当定时器 x 下溢时，寄存器TxIF (分别为T5IF、T4IF和T1IF的PIR[2:0]) 置 1，表示发生定时器 x 下溢事件。如果寄存器TxIE (分别为T5IE、T4IE和T1IE的PIE[2:0]) 和GIE都为 1，则会发生中断请求，并执行中断服务程序。在软件将 0 写入TxIF之前，TxIF不会被清除。

定时器的时序图如下图所示。

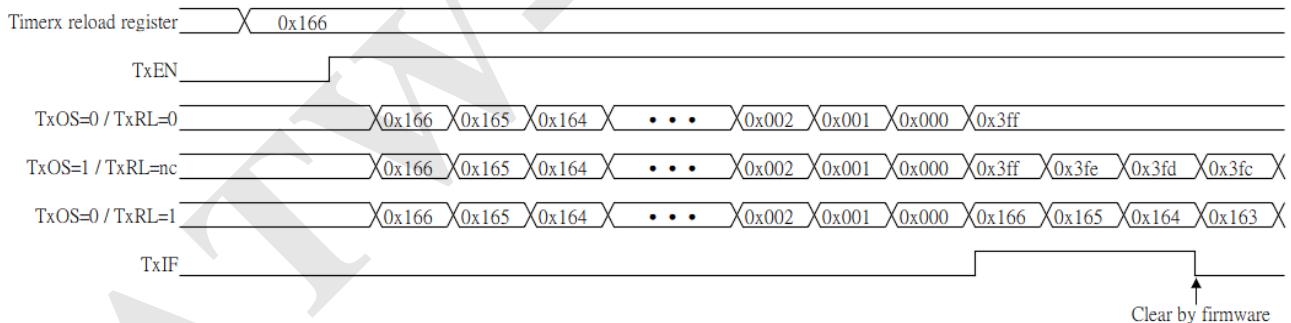


图 12 定时器 1 / 4 / 5 时序图

## 5.2 定时器 1 / 定时器 4 / 定时器 5 控制寄存器

### 5.2.1 TMRxL (定时器低字节寄存器)

根据TPSEL[2:0]的值。访问虚拟SFR TMRxL相当于访问物理SFR TMR1L、TMR4L或TMR5L，

当读取寄存器TMRxL和TPSEL[2:0]=001时，它将获得 10 位的下数定时器 1 的当前LSB值，即TMR1[7:0]。当写入寄存器TMRxL和TPSEL[2:0]=001时，它将数据写入TMR1L重新加载寄存器，如果T1EN=0，它也将写入TMR1[7:0]当前内容。这些操作也适用于TPSEL[2:0] = 100 (定时器 4) 或 101 (定时器 5) 条件。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMRxL	0xf	TMRx[7:0]							
读/写属性		读/写							
初始值		XXXXXXXX							

### 5.2.2 TMRxH (定时器高字节寄存器)

根据TPSEL[2:0]的当前值，访问虚拟SFR TMRxH相当于访问物理SFR TMR1H、TMR4H或TMR5H。

当读取寄存器TMRxH和TPSEL[2:0]=001时，它将获得 10 位的下数定时器 1 的当前MSB值，即TMR1[9:8]。当写入寄存器TMRxH和TPSEL[2:0]=001时，如果T1EN=0，则将TMR1H重新加载寄存器中的数据写入TMR1[9:8]当前内容。当写入寄存器TMRxH和TPSEL[2:0]=001时，将数据写入TMR1H重载寄存器。这些操作也适用于TPSEL[2:0] = 100 (定时器 4) 或 101 (定时器 5) 条件。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMRxH	0x10	-	-	-	-	-	-	TMRx[9:8]	
读/写属性		-	-	-	-	-	-	读/写	
初始值		X	X	X	X	X	X	XX	

## 5.2.3 TxCR1 (定时器控制寄存器)

根据TPSEL[2:0]的值, 访问虚拟SFR TxCR1 相当于访问物理SFR T1CR1、T4CR1 或T5CR1。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TxCR1	0x11	-	-	TMxOE	VFSELx	TMx_HRC	TxOS	TxRL	TxEN
读/写属性	-	-	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值	x	x	0	0	0	0	0	0	0

该寄存器用于配置定时器 x 功能。 (x= 1, 4 或 5)

Bit 7:6 未生效的。

Bit 5 **TMxOE:** 开启/关闭定时器 x 匹配输出, 当定时器 x 发生下溢时, TxOUT切换输出。

(\*1), (\*2)

1 = TxOUT 输出至PB5。

0 = PB5 为 GPIO。

Bit 4 **VFSELx:** 定时器 x 特殊时钟源选择 (\*3)

1 = 定时器 x 和 PWM时钟源是一种特殊的高速振荡时钟。

0 = 定时器 x 和PWM时钟源的选择取决于T1CS寄存器位。

Bit 3 **TMx\_HRC:** 定时器 x 时钟源选择。

1 = 定时器 x 和重载PWM的时钟源是高频振荡时钟。

0 = 定时器 x 和重载PWM的时钟源的选择取决于T1CS寄存器位。

Bit 2 **TxOS:** 当下溢发生时, 设置定时器 x 的工作模式。

1 = 单次计数模式 (One-Shot mode), 定时器 x 从初始值到 0x00, 计数一次。

0 = 连续计数模式 (Non-Stop mode), 下溢后, 定时器 x 将保持向下计数

定时器 x 下数功能		
TxOS	TxRL	
0	0	定时器 x 从重载值下数到0x00。 当下溢发生, 0x3FF被重载至定时器 x 并继续下数。 .
0	1	定时器 x 从重载的数值下数到0x00。 当下溢发生, 定时器 x 从重载值被重新载入初值并继续下数。
1	x	定时器 x 从初始值下数到0x00。 当下溢发生, 定时器 x 停止下数。

Bit 1 **TxRL:** 当选择连续计数模式 (TxOS=0) 时, 配置定时器 x 向下计数方式。

1 = 定时器 x 初始值从重载寄存器TMRx[9:0]重新加载。

0 = 当下溢发生时, 定时器 x 从 0x3FF开始连续计数。

Bit 0 **TxEN:** 开启/关闭定时器 x 。

1 = 开启定时器 x。

0 = 关闭定时器 x。

**注意:**

1. 对于 AT8BM84A, 只有定时器 1 具有定时器匹配输出 T1OUT。
2. 如果 T1OUT 和 PWM1 都配置为输出到 PB5, 则 T1OUT 的输出优先级高于 PWM1。
3. VFSELx 比 TMx\_HRC 有更高的优先级。

## 5.2.4 TxCR2 (定时器控制寄存器 2)

根据TPSEL[2:0]的值, 访问虚拟SFR TxCR2 相当于访问物理SFR T1CR2、T4CR2 或T5CR2。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TxCR2	0x12	-	-	TxCS	TxCE	/PSxEN	PSxSEL[2:0]		
读/写属性	-	-	读/写	读/写	读/写	读/写	读/写		
初始值	x	x	x	x	x	x	x		

该寄存器用于配置定时器 x 功能。 (x=1, 4 或 5)

Bit 7:6 未生效的。

Bit 5 **TxCS:** 定时器 x 时钟源选项。

1 = 选择EX\_CKIx引脚作为外部时钟输入。

0 = 选择指令时钟FINST。

Bit 4 **TxCE:** 定时器 x 外部时钟触发沿选项。

1 = EX\_CKIx引脚是下降沿时定时器 x 减一。

0 = EX\_CKIx引脚是上升沿时定时器 x 减一。

Timer	EXCKIx	Note
0	PA4	
1	PA4	
4	PA1	External clock option=0
	PA2	External clock option=1
5	PA1	External clock option=0
	PA2	External clock option=1

Bit 3 **/PSxEN:** 关闭/开启定时器 x 预分频器。

1 = 关闭定时器 x 预分频器。

0 = 开启定时器 x 预分频器。

Bit 2:0 **PSxSEL[2:0]:** 定时器 x 预分频器的预分频比选项。

PSxSEL[2:0]	预分频比
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

**注意: 在 /PS1EN=1 时总要设置 PS1SEL[2:0], 否则中断可能会发生误触发。**

## 6. PWM

AT8BM84A能够同时输出 5 个 10 位分辨率的PWM波形。PWM1、PWM2 和 PWM3 共用同一定时器框架(定时器 1)，而 PWM4 和 PWM5 各有独立的定时器框架 (定时器 4 和定时器 5)。PWM输出引脚由配置字决定，如下表所示。

配置字	00	01	10	11	注意
PWM1_PAD	PB5	PB1	PA3	PA5	
PWM2_PAD	PB4	PA4	PB7	PA7	When M62D_PAD=0, PB3 is replaced by PA4
PWM3_PAD	PA6	PA2	PB6	PB0	
PWM4_PAD	PC3	PB3	PA1	PC1	When M62D_PAD=0, PB3 is replaced by PA4
PWM5_PAD	PB2	PC0	PC5	PA0	When PC5_PWM=0, PC5 is replaced by PC2

表 6-1 PWM输出引脚配置

### 6.1 PWM1

#### 6.1.1 概述

PWM1 输出可以在PB5, PB1, PA3 或PA5 (由配置选择) 其中一个I/O引脚上获得。通过依次设置TPSEL[2:0]=001 和PWMxOEN=1 来启用PWM1。PWM1 将自动成为输出引脚。PWM1 输出的有效状态由寄存器PWMxOAL决定。当在TPSEL[2:0]=001 时将PWMxOAL设置为 1 时，PWM1 输出为低电平。当TPSEL[2:0]=001 且PWMxOAL为 0 时，PWM1 输出高电平。此外，PWM1 的占空比和帧率都是可编程的。占空比由寄存器PWM1DUTY[9:0] (PWM1DUTYH[1:0]和PWM1DUTYL[7:0]) 决定。当PWM1DUTY为 0 时，PWM1 输出将一直无效。当PWM1DUTY为 0x3FF 时，PWM1 将输出 1023/1024 的占空比。帧率由定时器 1 的周期决定。(对于自动加载模式，定时器 1 的周期为TMR1LD[9:0]) PWM1DUTY值必须小于或等于定时器 1 的周期。

当 PWM1 开启 (PWMxOEN=1) 时，AT8BM84A PWM 占空比的更新采用双缓冲机制：用户写 PWM1DUTYL 和 PWM1DUTYH 直到定时器 1 溢出才生效。这意味着新的 PWM1 占空比只能在下一个定时器 1 周期可用。

当 PWM1 被禁用时，先写 PWM1DUTY[9:8] MSB 2 位 (PWM1DUTYH[1:0])，然后写 PWM1DUTYL[7:0]，这样确保第一个定时器周期获得正确的 PWM 占空比。

PWM1 的结构框图如下图所示。

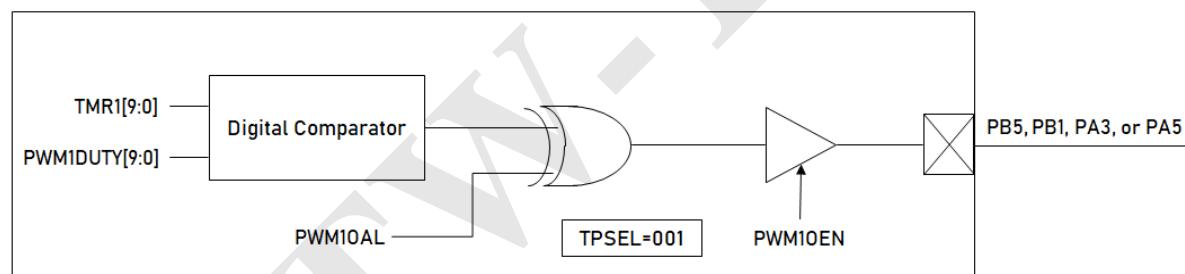


图 13 PWM1 的结构框图

## 6.2 PWM2

### 6.2.1 概述

PWM2 输出可以在PB4, PA4, PB3, PB7 或PA7 (由配置选择) 其中一个I/O引脚上获得。通过依次设置TPSEL[2:0]=002 和PWMxOEN=1 开启PWM2。PWM2 将自动成为输出引脚。PWM2 输出的有效状态由寄存器PWMxOAL决定。当在TPSEL[2:0]=001 时将PWMxOAL设置为 1 时, PWM2 输出为低电平。当TPSEL[2:0]=002 且PWMxOAL为 0 时, PWM2 输出高电平。此外, PWM2 的占空比和帧率都是可编程的。占空比由寄存器PWM2DUTY[9:0] (PWM2DUTYH[1:0]和PWM2DUTYL[7:0]) 决定。当PWM2DUTY为 0 时, PWM2 输出将一直无效。当PWM2DUTY为 0x3FF 时, PWM2 将输出 1023/1024 的占空比。帧率由定时器 1 周期决定。(对于自动加载模式, 定时器 1 周期为TMR2LD[9:0]) PWM2DUTY值必须小于或等于定时器 1 周期。

当 PWM2 开启 (PWMxOEN=1) 时, AT8BM84A PWM 占空比的更新采用双缓冲机制: 用户写 PWM2DUTYL 和 PWM2DUTYH 直到定时器 1 溢出才生效。这意味着新的 PWM2 占空比只能在下一个定时器 1 周期可用。

当 PWM2 被禁用时, 先写 PWM2DUTY[9:8] MSB 2 位 (PWM2DUTYH[1:0]), 然后写 PWM2DUTYL[7:0], 这样确保第一个定时器周期获得正确的 PWM 占空比。

PWM2 的结构框图如下图所示:

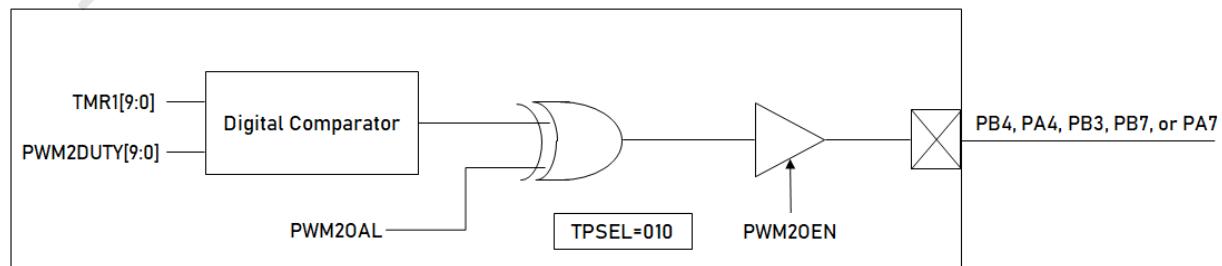


图 14 PWM2 的结构框图

## 6.3 PWM3

### 6.3.1 概述

PWM3 输出可以在PA6, PA2, PB6 或PB0 (由配置选择) 其中一个I/O引脚上获得。通过依次设置TPSEL[2:0]=003 和 PWMxOEN=1 开启PWM3。PWM3 将自动成为输出引脚。PWM3 输出的有效状态由寄存器PWMxOAL决定。当在TPSEL[2:0]=001 时将PWMxOAL设置为 1 时, PWM3 输出为低电平。当TPSEL[2:0]=003 且PWMxOAL为 0 时, PWM3 输出高电平。此外, PWM3 的占空比和帧率都是可编程的。占空比由寄存器PWM3DUTY[9:0] (PWM3DUTYH[1:0]和PWM3DUTYL[7:0]) 决定。当PWM3DUTY为 0 时, PWM3 输出将一直无效。当PWM3DUTY为 0x3FF 时, PWM3 将输出 1023/1024 的占空比。帧率由定时器 1 周期决定。(对于自动加载模式, 定时器 1 周期为TMR3LD[9:0]) PWM3DUTY值必须小于或等于定时器 1 周期。

当 PWM3 开启 (PWMxOEN=1) 时, AT8BM84A PWM 占空比的更新采用双缓冲机制: 用户写 PWM3DUTYL 和 PWM3DUTYH 直到定时器 1 溢出才生效。这意味着新的 PWM3 占空比只能在下一个定时器 1 周期可用。

当PWM3 被禁用时, 先写PWM3DUTY[9:8] MSB 2 位 (PWM3DUTYH[1:0]), 然后写PWM3DUTYL[7:0], 这样确保第一个定时器周期获得正确的PWM 占空比。

PWM3 的结构框图如下图所示。

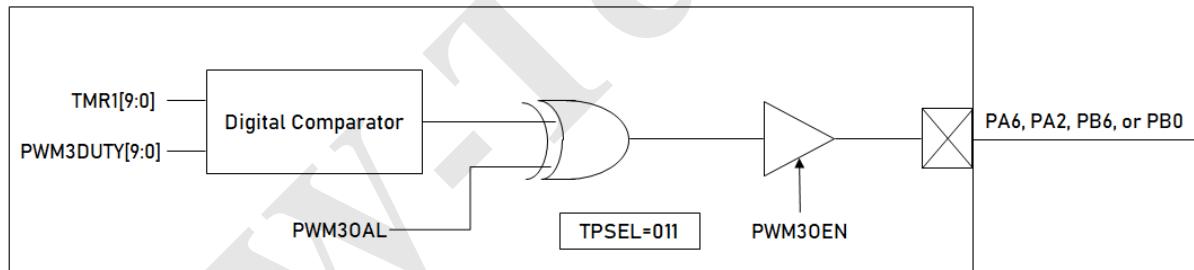


图 15 PWM3 的结构框图

## 6.4 PWM4

### 6.4.1 概述

PWM4 输出可以在PC3, PB3, PA4, PA1 或 PC1 (由配置选择) 其中一个I/O引脚上获得。通过依次设置TPSEL[2:0]=003 和 PWMxOEN=1 开启PWM4。PWM4 将自动成为输出引脚。PWM4 输出的有效状态由寄存器PWMxOAL决定。当在TPSEL[2:0]=001 时将PWMxOAL设置为1时, PWM4 输出为低电平。当TPSEL[2:0]=003 且PWMxOAL为0时, PWM4 输出高电平。此外, PWM4 的占空比和帧率都是可编程的。占空比由寄存器PWM4DUTY[9:0] (PWM4DUTYH[1:0]和PWM4DUTYL[7:0]) 决定。当PWM4DUTY 为0 时, PWM4 输出将一直无效。当PWM4DUTY 为0x3FF 时, PWM4 将输出 1023/1024 的占空比。帧率由定时器 4 的周期决定。(对于自动加载模式, 定时器 4 的周期为TMR3LD[9:0]) PWM4DUTY值必须小于或等于定时器 4 的周期。

当PWM4 开启 (PWMxOEN=1) 时, AT8BM84A PWM 占空比的更新采用双缓冲机制: 用户写 PWM4DUTYL 和 PWM4DUTYH 直到定时器 4 溢出才生效。这意味着新的PWM4 占空比只能在下一个定时器 4 周期可用。

当PWM4 被禁用时, 先写PWM4DUTY[9:8] MSB 2 位 (PWM4DUTYH[1:0]), 然后写PWM4DUTYL[7:0], 这样确保第一个定时器周期获得正确的PWM 占空比。

PWM4 的结构框图如下图所示。

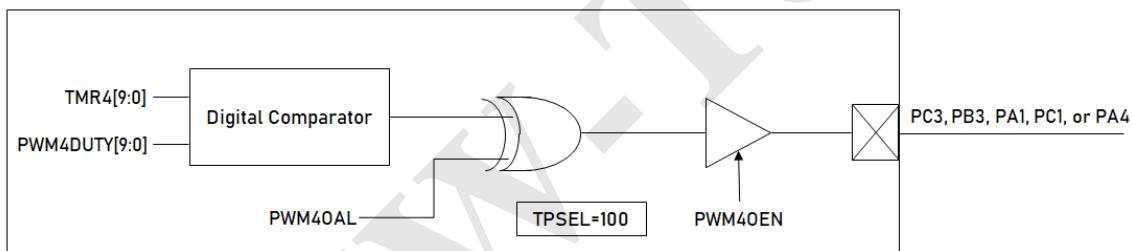


图 16 PWM4 的结构框图

## 6.5 PWM5

### 6.5.1 概述

PWM5 输出可以在PB2, PC0, PC5, PC2 或 PA0 (由配置选择) 其中一个I/O引脚上获得。通过依次设置TPSEL[2:0]=003 和 PWMxOEN=1 开启PWM5。PWM5 将自动成为输出引脚。PWM5 输出的有效状态由寄存器PWMxOAL决定。当在TPSEL[2:0]=001 时将PWMxOAL设置为1时, PWM5 输出为低电平。当TPSEL[2:0]=003 且PWMxOAL为0时, PWM5 输出高电平。此外, PWM5 的占空比和帧率都是可编程的。占空比由寄存器PWM5DUTY[9:0] (PWM5DUTYH[1:0]和PWM5DUTYL[7:0]) 决定。当PWM5DUTY 为0 时, PWM5 输出将一直无效。当PWM5DUTY 为0x3FF 时, PWM5 将输出 1023/1024 的占空比。帧率由定时器5 周期决定。(对于自动加载模式, 定时器5 周期为TMR3LD[9:0]) PWM5DUTY值必须小于或等于定时器5 周期。

当PWM5 开启 (PWMxOEN=1) 时, AT8BM84A PWM 占空比的更新采用双缓冲机制: 用户写 PWM5DUTYL 和 PWM5DUTYH 直到定时器5 溢出才生效。这意味着新的PWM5 占空比只能在下一个定时器5 周期可用。

当PWM5 被禁用时, 先写PWM5DUTY[9:8] MSB 2 位 (PWM5DUTYH[1:0]), 然后写PWM5DUTYL[7:0], 这样确保第一个定时器周期获得正确的PWM 占空比。

PWM5 的结构框图如下图所示。

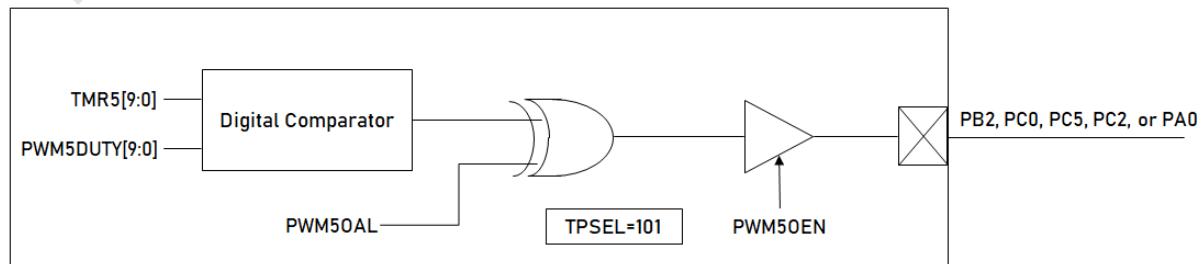


图 17 PWM5 的结构框图

## 6.6 PWM 周期

PWM周期通过写入PRx = [TMRxH[1:0], TMRxL[7:0]]寄存器 (x=1,4,5) 来指定。

PWM周期的计算公式如下:

$$\text{PWM 周期} = [(PRx)+1] * T_{CPU CYCLE} * (TMRx \text{ 时钟周期}) - PWM \text{ 时钟选择 } F_{INST}$$

$$\text{PWM 周期} = [(PRx)+1] * T_{HRC CYCLE} - PWM \text{ 时钟选择 } V_{HRC} \text{ 或 } HRC$$

PWM频率定义为 1/[PWM周期]。

## 6.7 PWM 占空比

PWM占空比 (PDx) 通过写入PWMxDUTYH[1:0], PWMxDUTYL[7:0] (PDx, x=1~5) 寄存器位来指定。最高可达 10-bit分辨率。以下公式用于计算PWM占空比 (百分比或时间)

$$\text{PWM 占空比}(\%) = PDx / (PRx+1) - PWM \text{ 时钟选择指令时钟}$$

$$\text{PWM 占空比(秒)} = PDx * T_{CPU CYCLE} * (TMRx \text{ 时钟周期}) - PWM \text{ 时钟选择 } F_{INST}$$

$$\text{PWM 占空比(秒)} = PDx * T_{HRC CYCLE} - PWM \text{ 时钟选择 } V_{HRC} \text{ 或 } HRC$$

PDx可以在任何时候写入, 但是直到TMRx溢出 (即周期结束) 之后, 占空比才锁存到PDx\_LH中。

PDx / PWMxDUTYH, PWMxDUTYL寄存器用于双缓冲PWM占空比。这种双缓冲对于无故障PWM操作至关重要。

给定PWM频率的最大PWM分辨率（位）如下公式：

$$\text{PWM Resolution (max)} = \log(\text{PRx}+1)/\log(2)$$

注意：如果PWM占空比大于PWM周期，则不产生PWM输出。

PWM 频率	3.906 KHz	7.81 kHz	62.5 kHz	125 kHz	250 kHz	500 kHz
定时器预分频器	1:2	1:4	1:1	1:1	1:1	1:1
PRx 值	3FFh	FFh	7Fh	3Fh	1Fh	0Fh
最大分辨率（位）	10	8	7	6	5	4

表 6-2 在 16MHz/2T 同步模式下的典型 PWM 频率和分辨率

## 6.8 CCP 模式

AT8BM84A CCP模块的主要特点包括：

- 一个 16 位输入捕捉模块，用于输入脚边沿事件：每次上升/每次下降/第 4 次上升/第 16 次上升。
- 一个 16 位输出比较模块，具有多个输出选项：设置输出/复位输出/触发中断。
- 增强型 PWM：3 对带死区控制的半桥。
- 增强型 PWM：一个全桥（正向和反向）模式。

### 6.8.1 模块配置

AT8BM84A CCP模块启用时，它们利用和征用定时器和PWM寄存器，如PWM5DUTY, PWM4DUTY, TMR5 和TMR4。因此，原始的定时器或PWM功能可能变得不可用。应注意避免启用与CCP操作冲突的其他与定时器相关功能。表 6-3 总结了各种CCP模式及其占用的寄存器。

CCP 模式	被征用的 PWM 寄存器	被征用的定时器寄存器
捕捉	捕捉寄存器 (16-bit) = {PWM5DUTYL[7:0], PWM4DUTYL[7:0]}	{TMR5L[7:0], TMR4L[7:0]}
比较	比较寄存器 (16-bit) = {PWM5DUTYL[7:0], PWM4DUTYL[7:0]}	{TMR5L[7:0], TMR4L[7:0]}
PWM(HB/FB)	PWM 占空比寄存器 (10-bit) = {PWM5DUTYH[1:0], PWM5DUTYL[7:0]}	{TMR5H[1:0], TMR5L[7:0]}
PWM(HB)	PWM 占空比寄存器 (10-bit) = {PWM4DUTYH[1:0], PWM4DUTYL[7:0]}	{TMR4H[1:0], TMR4L[7:0]}

表 6-3 AT8BM84A CCP 定时器和 PWM 寄存器资源

**注意：AT8BM84A CCP1 是全功能的，而 CCP2, CCP3 只实现 HB 功能。**

### 6.8.2 CCP I/O 配置

CCP模块不控制AT8 引脚的方向。CCP模块可以有捕捉模式的输入，可以有比较模式的输出，也可以有多达四个PWM输出（Px A到Px D）。有效的输出取决于所选择的CCP操作模式。引脚分配总结在表 2 中。通过设置PM<1:0>和CCPxM<3:0>位来选择合适的CCP模式，并且引脚方向控制IOST寄存器还必须初始化。

建议使用CCP模块的启动顺序如下：

1. 通过向CCPxM<3:0> (CCPxCON<3:0>)写入 0000'来禁用该模块。
2. 通过设置IOST寄存器初始化引脚方向，通过IO端口寄存器初始化引脚状态。
3. 设置相关定时器模式，初始化相关定时器值。
4. 将任何初始值写入CCP和相关寄存器。对于增强型PWM模式，很好的配置CCPxCON<7:6>。
5. 通过将适当的模式选择值写入CCPxM<3:0>来启用该模块。

### 6.8.3 捕捉模式（仅在 CCP1 中可用）

在捕捉模式下，当事件发生在CCP引脚（PB2）上时，捕捉寄存器成对捕捉所选定时器寄存器的 16 位值。事件被定义为以下条件之一：

- 每一次下降沿
- 每一次上升沿
- 每 4 次的上升沿
- 每 16 次的上升沿

事件由CCP 模式选择位选择，CCPxM<3:0> (CCPxCON<3:0>)。当捕捉发生时，CCP中断请求标志位（CCPIF）被设置。（必须在软件中清除。）如果在读取捕捉的数据寄存器中的值之前发生另一次捕捉，则旧捕捉值将被新捕捉值覆盖。

图 18 显示了捕捉模式的结构框图。

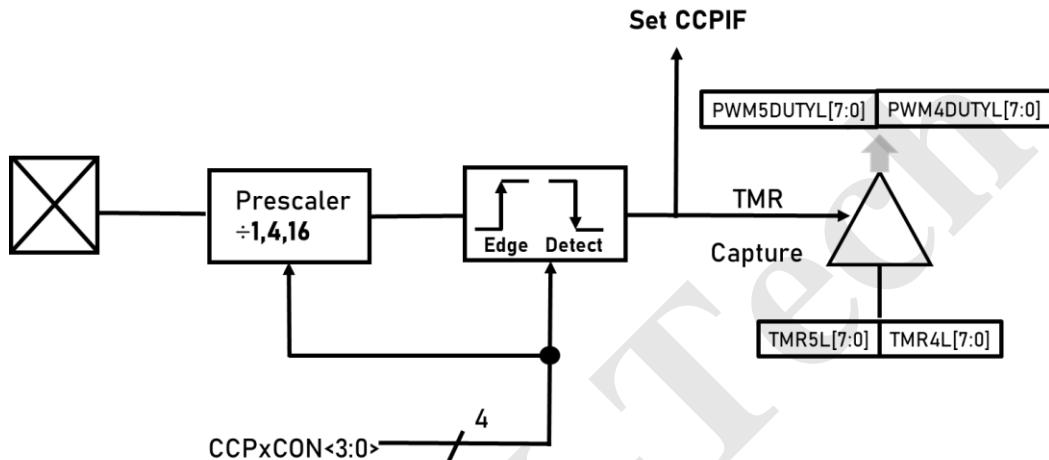


图 18 捕捉模式结构框图

在捕捉模式下，相应的CCP引脚（PB2）应该通过设置相应的IOST方向位来配置作为输入脚。

要用于捕捉功能，所选定时器必须初始化并在定时器模式或同步计数器模式下运行。

**注意，CCP定时器是一个向下计数定时器。每个CCP模块选择使用的定时器如表 6-3 所示。**

### 6.8.4 比较模式 (仅在 CCP1 中可用)

在比较模式下, 16 位比较数据值不断与所选定时器的寄存器的值进行比较。当匹配发生时, CCPx引脚可以为:

- 高驱动
- 低驱动
- 切换输出 (由高到低或由低到高)
- 保持不变 (即反映I/O锁存器的状态)

当CCP模块开启时, 该模块不会自动配置引脚作为输出脚。

若要用于比较功能, 所选定时器必须在定时器模式或同步计数器模式下运行。在异步计数器模式下, 比较操作可能无法工作。注意, CCP定时器以向下计数模式运行。

两个CCP模块都配备了一个特殊事件触发器。这是在比较模式下生成的内部硬件信号, 用于触发其他模块的操作。通过选择比较特殊事件触发模式 (CCPxM<3:0> = 1011) 来启用特殊事件触发。

对于任何一个CCP模块, 特殊事件触发器都会重置当前分配给模块的定时器资源的定时器寄存器值。这允许CCPRx寄存器作为任一定时器的可编程周期寄存器。

当选择生成软件中断模式 (CCPx <3:0> = 1010) 时, CCPx引脚不受影响。如果开启, 只产生一个CCP中断, 并且设置CCPxIE位。

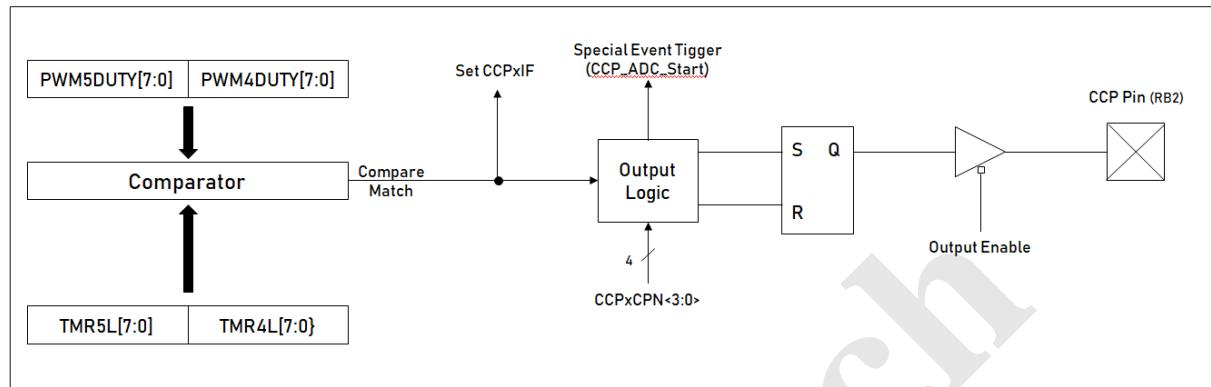


图 19 比较模式结构框图

### 6.8.5 增强型 PWM 模式

AT8BM84A CCP模块在CCP模块中包括增强型PWM模式, 它提供了第 6 章中描述的PWM的另一种选择。CCP增强型PWM模式将在 6.7.6 节中进行描述。

在CCP脉宽调制 (PWM) 模式下, CCP引脚产生高达 10 位分辨率的PWM输出。图 20 显示了该模块在 PWM模式下的简化结构框图。

PWM输出 (图 21) 具有时基 (周期) 和输出保持高电平的时间 (占空比)。PWM的频率是周期的倒数 (1/周期)。

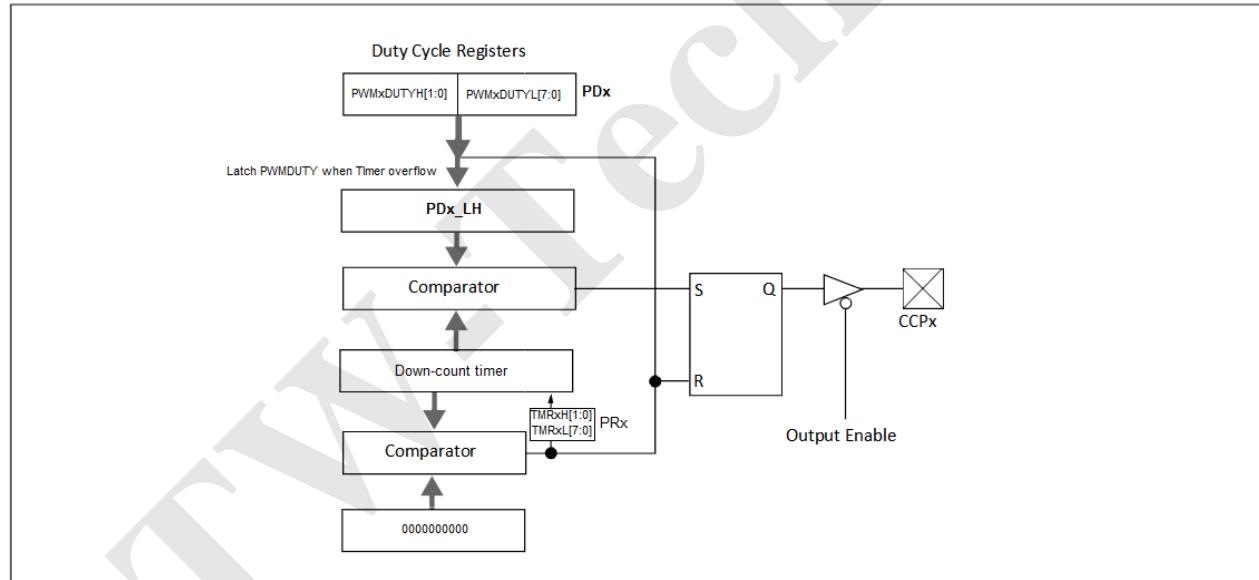


图 20 增强型 PWM 结构框图

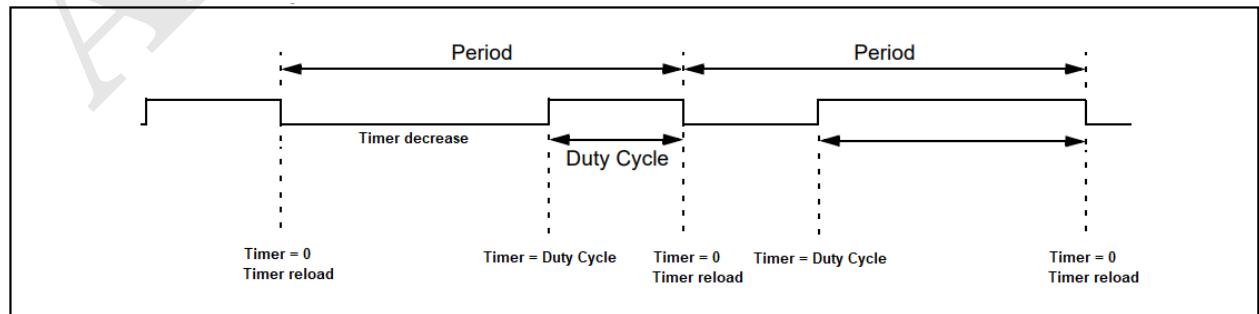


图 21 增强型 PWM 输出

### 6.8.6 PWM 操作设置

为PWM操作配置CCP模块：

- 设置正确的引脚方向和初始值。
- 通过写入PRx寄存器设置PWM周期。
- 通过写入PDx寄存器设置PWM占空比。
- 设置TMRx MODE，预分频值，然后通过写入TxCR1开启Timerx。
- 为PWM操作配置CCPxCON寄存器。

### 6.8.7 增强型 PWM 特性

在CCP模块中，增强型PWM模式可以在多达四个不同的输出引脚上产生PWM信号，分辨率高达 10 位。它可以通过四种不同的PWM输出模式做到这一点：

1. 单个PWM模式（可用于CCP1），此功能与第 6 章节中描述的PWM基本相同。
2. 带死区控制的半桥PWM模式（可用于CCP1 和 CCP2）。
3. 全桥PWM，正向模式（可用于CCP1）。
4. 全桥PWM，反向模式（可用于CCP1）。

要选择增强型 PWM 模式，必须适当设置  $PWMxM<1:0>$  bits（ $CCPxCON<7:6>$ ）。

PWM输出与I/O引脚多路复用，并指定为Px A, Px B, Px C和Px D。PWM引脚的极性是可配置的，通过在CCPxCON寄存器中设置适当的CCPxM位来选择。为每个增强型PWM模式提供引脚分配。

图 22 提供了增强型PWM模块的简化框图示例。

图 23 提供了各种CCP模式下的波形。

**注意：为了防止在PWM首次启用时产生不完整的波形，CCP模块在产生PWM信号之前先等待，直到一个新的PWM周期开始。**

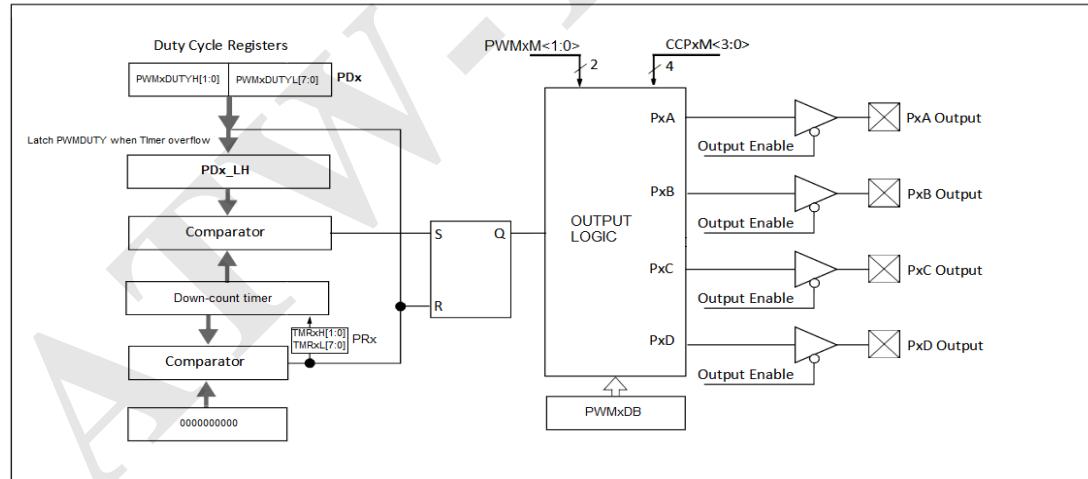


图 22 增强型 PWM 模式的简化框图

ECCP Mode	PWMxM<1:0>	Px A		Px B		Px C		Px D	
CCPx		CCP2	CCP3	CCP1	CCP2	CCP1	CCP2	CCP1	CCP2
单个	00	PB2	-	-	-	-	-	-	-
半桥 (死区)	01	PB2/ PC4/ PA7	PC2	PA5/ PA1/ PA6	PC5	-	-	-	-
全桥，正向	10	PB2/ PC4/ PA7	-	PA5/ PA1/ PA6	-	PA2/ PA2/ PA5	-	PA3/ PA3/ PA4	-
全桥，反向	11	PB2/ PC4/ PA7	-	PA5/ PA1/ PA6	-	PA2/ PA2/ PA5	-	PA3/ PA3/ PA4	-

表 6-4 AT8BM84A CCP输出引脚配置

**注意：**

1. 引脚通过选项选择。
2. 任何未被增强型 PWM 模式使用的引脚都可用于备用引脚功能。

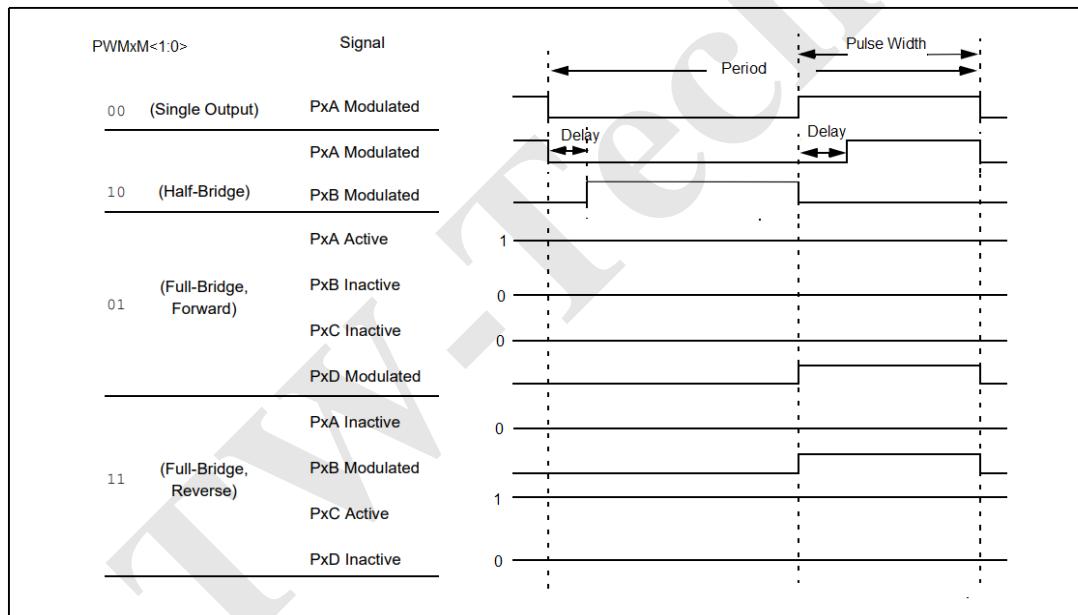


图 23 增强型 PWM 输出关系 (高有效状态) 示例

### 6.8.8 半桥模式 (可用于 CCP1 和 CCP2)

在半桥模式下，两个引脚用作输出来驱动推-拉负载。PWM输出信号在PxA引脚上输出，而互补PWM输出信号在PxB引脚上输出（见图 24）。该模式可用于半桥应用，也可用于全桥应用，其中四个功率开关用两个PWM信号调制（图 25）。

在半桥模式下，可编程死区延迟可用于防止半桥功率器件中的击穿电流。SFR PWMxDB寄存器的DB<7:0>位的值设置了在输出被激活之前PWM输入时钟周期的数量。

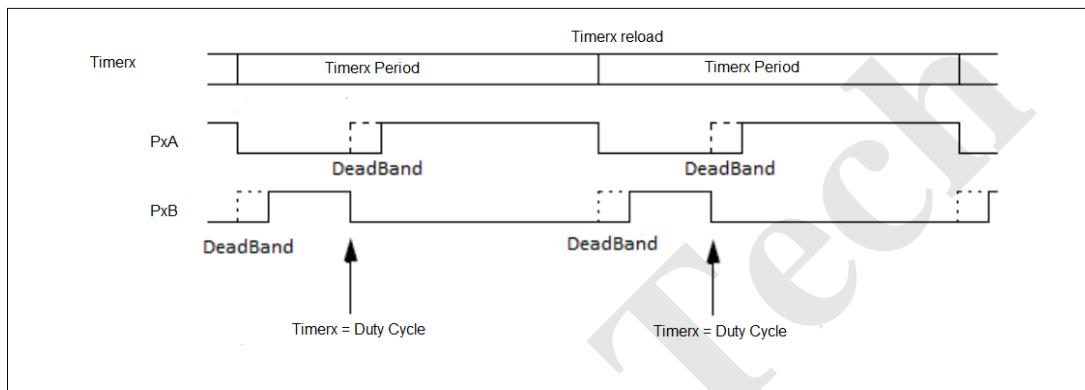


图 24 半桥 PWM 输出示例

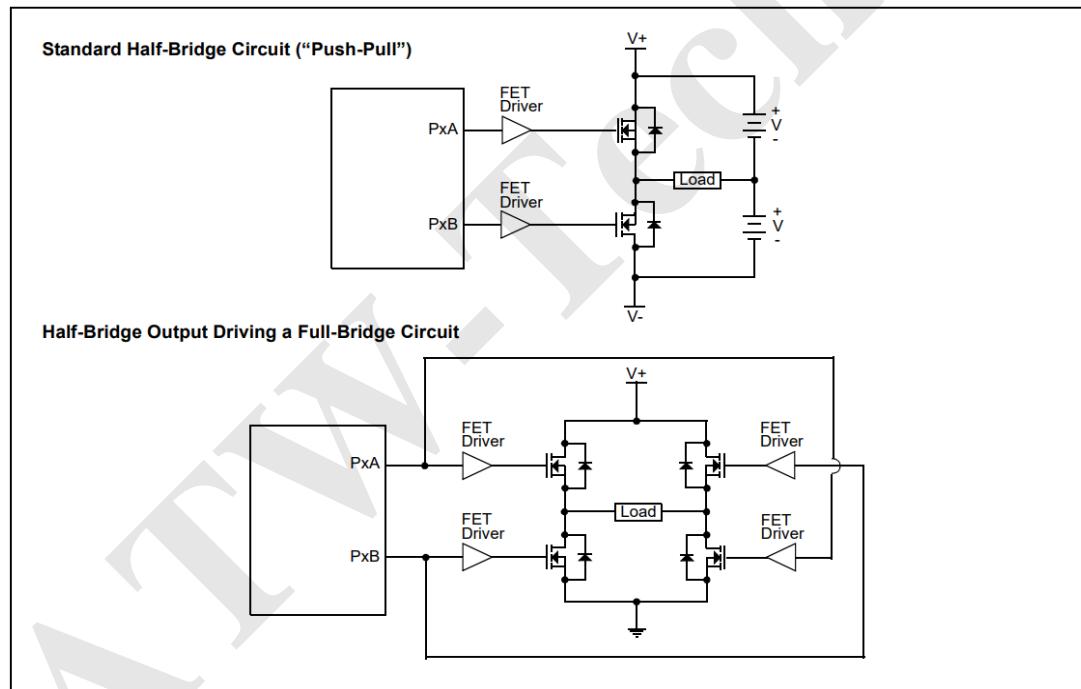


图 25 半桥应用示例

### 6.8.9 全桥模式（可用于 CCP1）

在全桥模式下，所有四个引脚都用作输出。图 26 提供了一个全桥应用的示例。

在正向模式下，Px A引脚被驱动到其有效状态，Px D引脚被调制，而Px B和Px C引脚被驱动到其非有效状态，如图 27 所示。

在反向模式下，Px C引脚被驱动到其有效状态，Px B引脚被调制，而Px A和Px D引脚被驱动到其非有效状态，如图 27 所示。

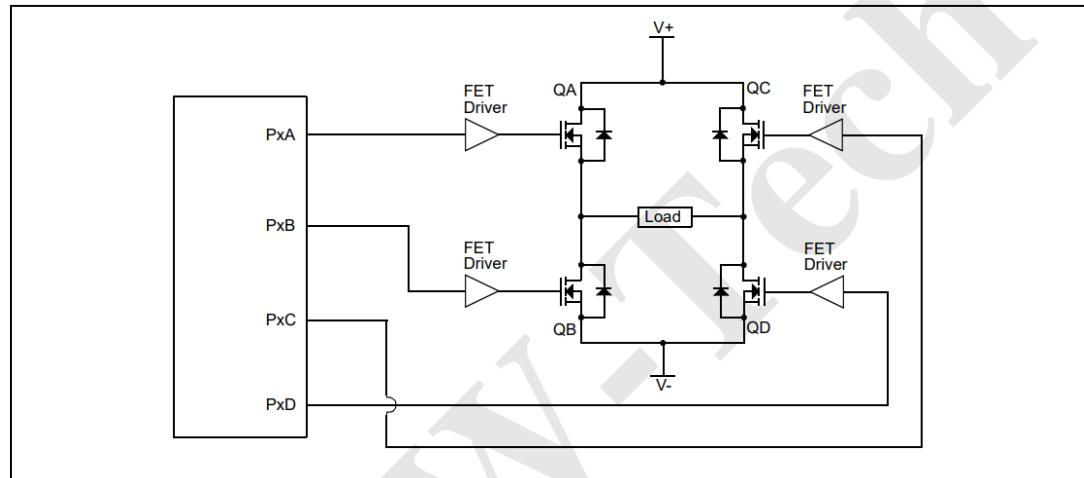


图 26 全桥应用的示例

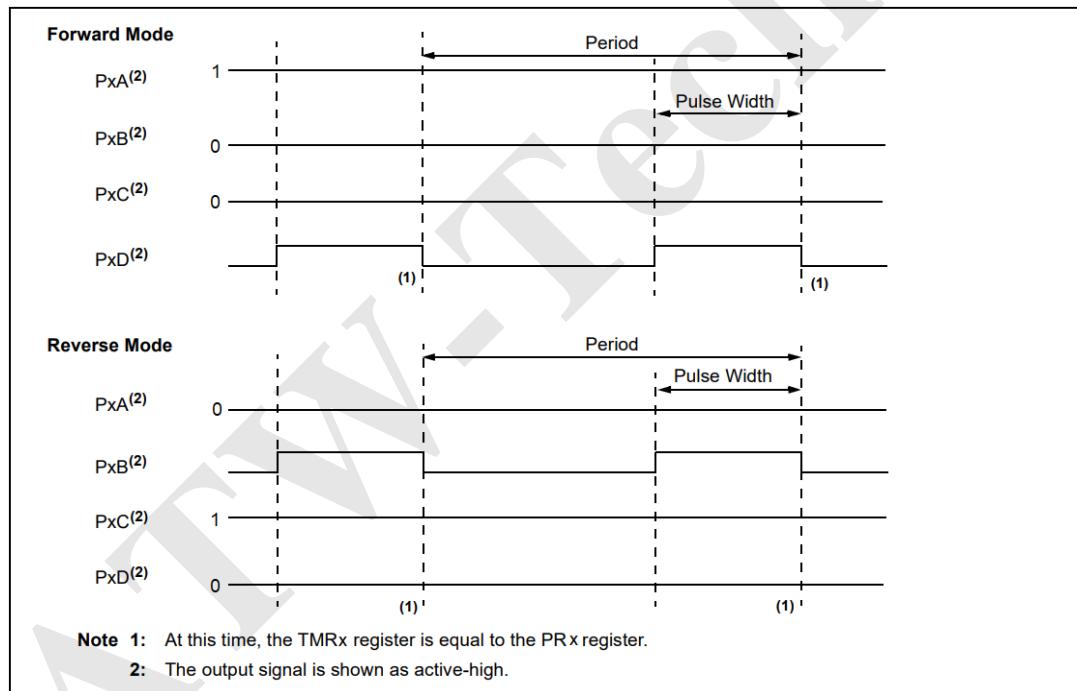


图 27 全桥PWM输出示例

### 6.8.10 全桥模式（可用于 CCP1）

在全桥模式下，CCPxCON寄存器中的PWMxM1位允许用户控制正向 / 反向。当应用固件改变这个方向控制位时，模块将在下一个PWM周期改变到新的方向。

在软件中通过更改CCPxCON寄存器的PWMxM1位来启动方向更改。以下序列发生在当前PWM周期结束之前：

PWM调制在下一个周期开始时恢复。为了说明这个序列，请看图 28。

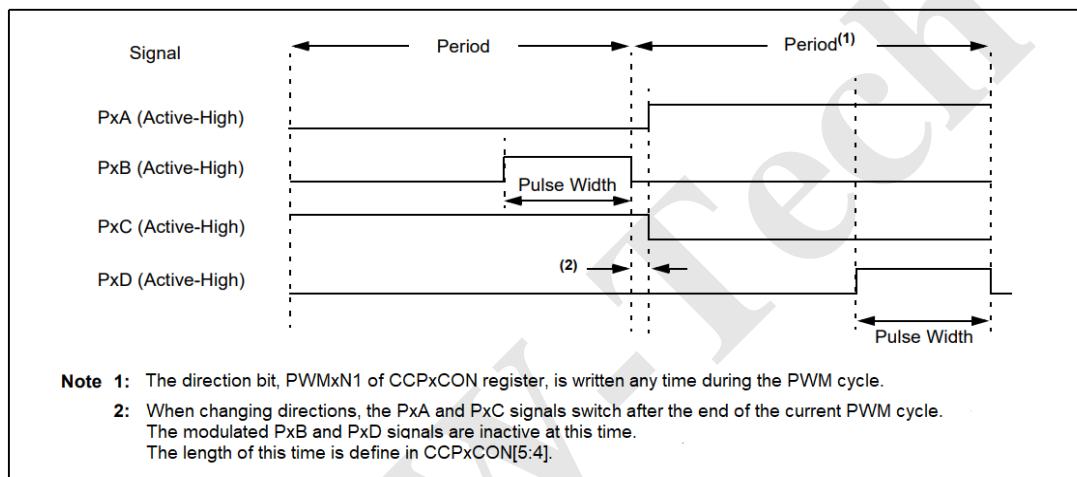


图 28 PWM方向改变示例

全桥模式不提供死区延迟。由于一次只调制一个输出，因此通常不需要死区延迟。有一种情况需要死区延迟。当以下两个条件同时满足时，就会出现这种情况：

1. 当输出占空比等于或接近 100% 时，PWM输出的方向发生改变。
2. 电源开关（包括电源器件和驱动电路）的断开时间大于导通时间。

图 29 显示了 PWM 方向在接近 100% 占空比时从反向到正向变化的示例。在本例中，在  $t_1$  时刻， $PxA$  和  $PxD$  输出变为有效，而  $PxC$  输出变为非有效。由于电源器件的断开时间比导通时间长，因此在  $t$  段时间内会有击穿电流流过电源器件  $QC$  和  $QD$ 。电源器件  $QA$  和  $QB$  也会出现同样的现象，因为 PWM 方向由反向变为正向。

如果应用需要在高占空比下改变 PWM 方向，消除击穿电流的两种可能解决方案是：

1. 在改变方向之前为一个 PWM 周期减少 PWM 占空比。
2. 使用开关驱动器，可以使开关关闭的速度比开关打开的速度快。也可能存在其他防止击穿电流的选项。

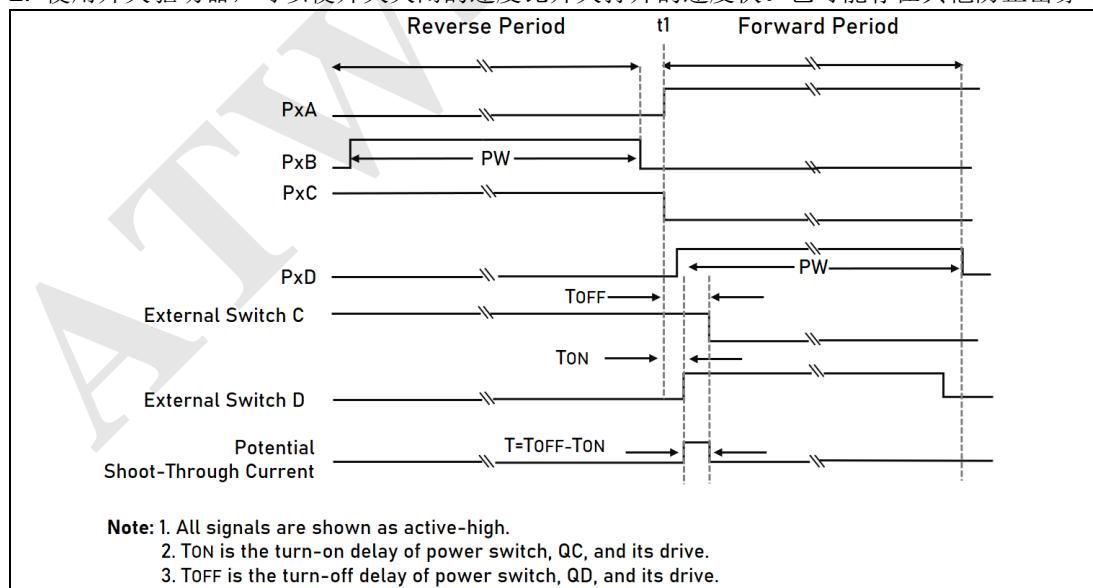


图 29 PWM 方向在接近 100% 占空比时变化的示例

### 6.8.11 启动注意事项

当使用任何 PWM 模式时，应用硬件必须在 PWM 输出引脚上使用适当的外部上拉和/或下拉电阻。

**注意：当微控制器从Reset中释放出来时，所有的I/O引脚都处于高阻抗状态。外部电路必须保持电源开关器件处于OFF状态，直到微控制器驱动具有适当信号电平的I/O引脚或激活PWM输出。**

CCPxCON 寄存器的 PWMxM<1:0>位允许用户选择每对 PWM 输出引脚（ $PxA/PxC$  和  $PxB/PxD$ ）的 PWM 输出信号是高有效还是低有效。PWM 输出极性必须在 PWM 引脚输出驱动器启用之前选择。不建议在启用 PWM 引脚输出驱动器时更改极性配置，因为这可能会导致应用电路损坏。

当 PWM 模块初始化时， $PxA$ 、 $PxB$ 、 $PxC$  和  $PxD$  输出锁存器可能不处于适当的状态。在增强型 PWM 模式的同时开启 PWM 引脚输出驱动器可能会对应用电路造成损坏。增强型 PWM 模式必须在正确的输出模式下启用，并在启用 PWM 引脚输出驱动器之前完成一个完整的 PWM 周期。当第二个 PWM 周期开始时，一个完整的 PWM 周期的完成由 INTF 寄存器的 TMRxIF 位表示。

### 6.8.12 电源控制模式下的操作

睡眠模式下，关闭所有时钟源。计时器不会增加，模块的状态也不会改变。如果 CCPx 引脚正在驱动一个值，它将继续驱动该值。当设备唤醒时，它将从这个状态继续。

在待机模式下，主时钟将继续为模块提供时钟，不会改变。

### 6.8.13 复位影响

任何复位事件将强制所有I/O端口进入输入模式，CCP寄存器进入复位状态。

## 6.9 PWM控制寄存器

### 6.9.1 PWMxCON (PWM 控制寄存器)

根据TPSEL[2:0]的值，访问虚拟SFR PWMxCON相当于访问物理SFR PWM1CON、PWM2CON、PWM3CON、PWM4CON或PWM5CON。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMxCON	0x1f	-	-	-	-	-	-	PWMxOEN	PWMxOAL
读/写属性	-	-	-	-	-	-	-	读/写	读/写
初始值	x	x	x	x	x	x	x	0	0

Bit 7:2 未生效的。

#### 对于TPSEL [2:0]=001

Bit 1 **PWMxOEN:** 开启/关闭PWM1 输出。

1 = PWM1 由PAD输出。

0 = PAD是一般IO脚。

Bit 0 **PWMxOAL:** 定义PWM1 输出有效状态。

1 = PWM1 输出低有效。

0 = PWM1 输出高有效。

#### 对于TPSEL [2:0]=010

Bit 1 **PWMxOEN:** 开启/关闭PWM2 输出。

1 = PWM2 由PAD输出。

0 = PAD是一般IO脚。

Bit 0 **PWMxOAL:** 定义PWM2 输出有效状态。

1 = PWM2 输出低有效。

0 = PWM2 输出高有效。

#### 对于TPSEL [2:0]=011

Bit 1 **PWMxOEN:** 开启/关闭PWM3 输出。

1 = PWM3 由PAD输出。

0 = PAD是一般IO脚。

Bit 0 **PWMxOAL:** 定义PWM3 输出有效状态。

1 = PWM3 输出低有效。

0 = PWM3 输出高有效。

#### 对于TPSEL [2:0]=100

Bit 1 **PWMxOEN:** 开启/关闭PWM4 输出。

1 = PWM4 由PAD输出。

0 = PAD是一般IO脚。

Bit 0 **PWMxOAL:** 定义PWM4 输出有效状态。

1 = PWM4 输出低有效。

0 = PWM4 输出高有效。

**For TPSEL [2:0]=101**

Bit 1 **PWMxOEN:** 开启/关闭PWM5 输出。

1 = PWM5 由PAD输出。

0 = PAD是一般IO脚。

Bit 0 **PWMxOAL:** 定义PWM5 输出有效状态。

1 = PWM5 输出低有效。

0 = PWM5 输出高有效。

### 6.9.2 PWMDBx (PWM 死区寄存器)

根据CCPSEL[1:0] 的值, 访问虚拟SFR PWMDBx相当于访问物理SFR PWMDB1 或PWMDB2 或PWMDB3。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDBx	0x11e	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
读/写属性		读/写							
初始值		0	0	0	0	0	0	0	0

对于CCPSEL=0

Bit 7:0 **CCP1 半桥输出模式的PWM延迟计数:** FOSC或CPU的数量  
P1A转换和P1B转换之间的周期。

对于CCPSEL=1

Bit 7:0 **CCP2 半桥输出模式的PWM延迟计数:** FOSC或CPU的数量  
P2A转换和P2B转换之间的周期。

### 6.9.3 PWM1DUTY (PWM 1 占空比寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DUTYL	0x91	PWM1DUTY[7:0]							
读/写属性		写							
初始值		xxxxxxxx							

Bit 7:0 **PWM1DUTY[7:0]:** PWM1 占空比数据LSB 8 位。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DUTYH	0x92	-	-	-	-	-	-	PWM1DUTY[9:8]	
读/写属性		-	-	-	-	-	-	写	
初始值		x	x	x	x	x	x	xx	

Bit 7:2 未生效的。

Bit 1:0 **PWM1DUTY[9:8]**: PWM1 占空比数据MSB 2 位。 (PWM1DUTY[9:0])6.9.4 **PWM2DUTY (PWM 2 占空比寄存器)**

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2DUTY <sub>L</sub>	0x93								
读/写属性									写
初始值									XXXXXXXX

Bit 7:0 **PWM2DUTY[7:0]**: PWM2 占空比数据LSB 8 位。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2DUTY <sub>H</sub>	0x94	-	-	-	-	-	-		PWM2DUTY[9:8]
读/写属性		-	-	-	-	-	-		写
初始值		X	X	X	X	X	X		XX

Bit 7:2 未生效的。

Bit 1:0 **PWM2DUTY[9:8]**: PWM2 占空比数据MSB 2 位。 (PWM2DUTY[9:0])6.9.5 **PWM3DUTY (PWM 3 占空比寄存器)**

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM3DUTY <sub>L</sub>	0x95								
读/写属性									写
初始值									XXXXXXXX

Bit 7:0 **PWM3DUTY[7:0]**: PWM3 占空比数据LSB 8 位。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM3DUTY <sub>H</sub>	0x96	-	-	-	-	-	-		PWM3DUTY[9:8]
读/写属性		-	-	-	-	-	-		写
初始值		X	X	X	X	X	X		XX

Bit 7:2 未生效的。

Bit 1:0 **PWM3DUTY[9:8]**: PWM3 占空比数据MSB 2 位。 (PWM3DUTY[9:0])6.9.6 **PWM4DUTY (PWM 4 占空比寄存器)**

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM4DUTY <sub>L</sub>	0x97								
读/写属性									写
初始值									XXXXXXXX

Bit 7:0 **PWM4DUTY[7:0]**: PWM4 占空比数据LSB 8 位。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM4DUTY <sub>H</sub>	0x98	-	-	-	-	-	-		PWM4DUTY[9:8]
读/写属性		-	-	-	-	-	-		写
初始值		X	X	X	X	X	X		XX

Bit 7:2

未生效的。

Bit 1:0

**PWM4DUTY[9:8]:** PWM4 占空比数据MSB 2 位。 (PWM4DUTY[9:0])**6.9.7 PWM5DUTY (PWM 5 占空比寄存器)**

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM5DUTYL	0x99	PWM5DUTY[7:0]							
读/写属性		写							
初始值		xxxxxxxx							

Bit 7:0

**PWM5DUTY[7:0]:** PWM5 占空比数据LSB 8 位。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM5DUTYH	0x9A	-	-	-	-	-	-	PWM5DUTY[9:8]	
读/写属性		-	-	-	-	-	-	写	
初始值		x	x	x	x	x	x	xx	

Bit 7:2

未生效的。

Bit 1:0

**PWM5DUTY[9:8]:** PWM5 占空比数据MSB 2 位。 (PWM5DUTY[9:0])**6.9.8 CCPxCON (CCPx 控制寄存器)**

根据CCPSEL的值, 访问虚拟SFR CCPxCON相当于访问物理SFR CCP1CON或CCP2CON。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCPxCON	0x11F	PWMx M1	PWMx M0	FBCH1	FBCH0	CCPx M3	CCPx M2	CCPx M1	CCPx M0
读/写属性	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值	0	0	0	0	0	0	0	0	0

当 CCPSEL = 0

Bit 7:6 **PWMxM[1:0]:**

当 CCPxM[3:2] = 00/01/10 时, PB2 是捕捉输入脚, 比较输出脚。

当 CCPxM[3:2] = 11, PWMxM[1:0]时,

00 = PWM单一输出。

01 = PWM 全桥正向输出。

10 = PWM 半桥输出。

11 = PWM 全桥反向输出。

Bit 5:4 **FBCH[1:0] :** 全带变向间隙。

00 = 1 个CPU周期。

01 = 4 个CPU周期。

1x = 16 个CPU周期。

Bit 3:0 **CCPxM[3:0]:** CCP模式选择。

0000 = OFF

0010 = 比较模式, 在匹配时切换输出。

0100 = 捕捉模式, 捕捉每一次下降沿。

- 0101 = 捕捉模式，捕捉每一次上升沿。  
0110 = 捕捉模式，每 4 次上升沿捕捉一次。  
0111 = 捕捉模式，每 16 次上升沿捕捉一次。  
1000 = 比较模式，在匹配时设置输出。  
1001 = 比较模式，在匹配时清除输出。  
1010 = 比较模式，在匹配时中断。  
1011 = 比较模式，触发特殊事件。  
1100 = PWM模式，P1A/P1C高有效，P1D/P1B高有效。  
1101 = PWM模式，P1A/P1C高有效，P1D/P1B低有效。  
1110 = PWM模式，P1A/P1C低有效，P1D/P1B高有效。  
1111 = PWM模式，P1A/P1C低有效，P1D/P1B低有效。

**当 CCPSEL = 1**

Bit 7:6

**PWMxM[1:0]:**

- 当 CCPxM[3:2] = 00/01/10, N.C.  
当 CCPxM[3:2] = 11, PWMxM[1:0]时，  
00 = N.C.  
01 = N.C.  
10 = N.C.  
11 = PWM 半桥输出。

Bit 3:0

**CCPxM[3:0]: CCP模式选择。**

- 1100 = PWM模式，P2A高有效，P2B高有效。  
1101 = PWM模式，P2A高有效，P2B低有效。  
1110 = PWM模式，P2A低有效，P2B高有效。  
1111 = PWM模式，P2A低有效，P2B低有效。

## 7. 蜂鸣器 (Buzzer)

### 7.1 概述

当寄存器BZ1EN (BZ1CR1[7]) 设置为 1 时, Buzzer1 输出 (BZ1) 可在I/O引脚PB3 上获得。PB3 将自动成为输出引脚。BZ1 的频率可以从定时器 1 输出或预分频器 1 输出中得到, 分频比由寄存器BZ1FSEL[3:0] (BZ1CR[3:0]) 决定。当BZ1FSEL[3]为 0 时, 选择预分频器 1 输出, 产生BZ1 输出。当BZ1FSEL[3]为 1 时, 选择定时器 1 输出, 产生BZ1 输出。分频率可从 1:2 到 1:256, 以产生各种频率。Buzzer1 的结构框图如下图所示。

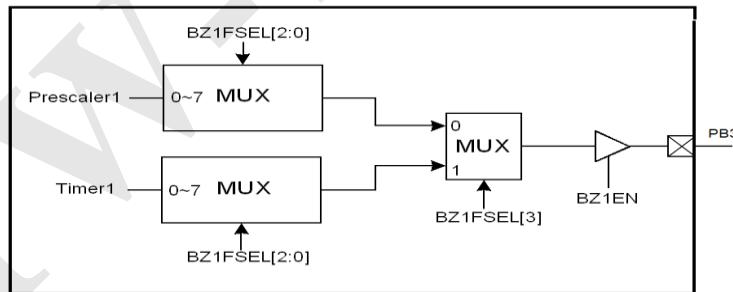


图 30 Buzzer1 结构框图

### 7.2 BZ1CR (Buzzer1 控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BZ1CR	0x13	BZ1EN	-	-	-	BZ1FSEL[3:0]			
读/写属性		写	-	-	-	写			
初始值		0	x	x	x	1	1	1	1

Bit 7      **BZ1EN:** 开启/关闭BZ1 输出。

1 = 开启 Buzzer1。

0 = 关闭 Buzzer1。

Bit 6:4      未生效的。

Bit 3:0      **BZ1FSEL[3:0]:** BZ1 输出频率选择。

BZ1FSEL[3:0]	BZ1 频率选择	
	时钟源	预分频比
0000	预分频器 1 输出	1:2
0001		1:4
0010		1:8
0011		1:16
0100		1:32
0101		1:64
0110		1:128
0111		1:256
1000	定时器 1 输出	定时器 1 bit 0
1001		定时器 1 bit 1
1010		定时器 1 bit 2
1011		定时器 1 bit 3
1100		定时器 1 bit 4
1101		定时器 1 bit 5
1110		定时器 1 bit 6
1111		定时器 1 bit 7

## 8. IR (红外) 载波

### 8.1 概述

根据配置字选择IR载波引脚为PB1或PA3。将寄存器IREN (IRCR[0]) 设置为1后生成IR载波。当IREN设为1时，IR载波引脚自动成为输出引脚。当IREN清除为0时，PB1或PA3将成为配置时的通用I/O引脚。

红外载波频率由寄存器IRF57K (IRCR[1]) 选择。当IRF57K为1时，IR载波频率为57KHz。当IRF57K为0时，IR载波频率为38KHz。由于红外载波频率来源于高频系统振荡FHOSC，所以在使用外部晶振时，有必要指定系统振荡的频率。寄存器IROSC358M (IRCR[7]) 用于向AT8BM84A提供此信息。当IROSC358M为1时，外部晶振频率为3.58MHz，当IROSC358M为0时，外部晶振频率为455KHz。当采用内部高频振荡时，该寄存器将被忽略，并为红外模块提供4MHz时钟。

根据红外载波引脚输出数据可选择红外载波的有效状态（极性）。当寄存器位IRCSEL (IRCR[2]) 为1，且红外引脚输出数据为0时，该引脚产生红外载波。当寄存器位IRCSEL (IRCR[2]) 为0时，且红外引脚输出数据为1时，该引脚产生红外载波。红外载波的极性如下图所示。

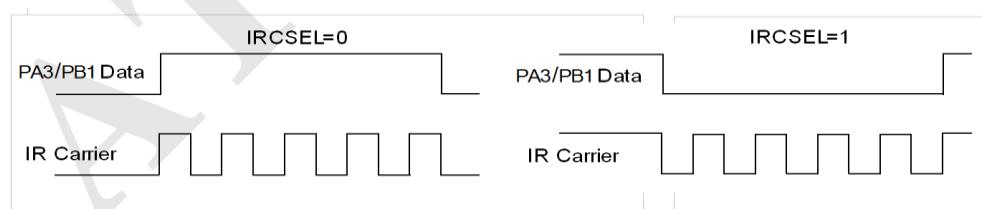


图 31 红外载波极性与输出数据

IREN Register	IR_PAD Option	IR function	IR Pad
0	X	OFF	-
1	1	ON	PA3
1	0	ON	PB1

表 8-1 红外载波选项

### 8.2 IRCR (红外控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCR	0x90	IROSC358M	-	-	-	-	IRCSEL	IRF57K	IREN
读/写属性		写	-	-	-	-	写	写	写
初始值	0	x	x	x	x	x	0	0	0

Bit 7 **IROSC358M:** 当使用外部晶体时，根据使用的晶体类型确定此位。如果使用内部高频振荡，则忽略该位。

1 = 晶振频率为3.58MHz。

0 = 晶振频率为455KHz。

Bit 6:3 未生效的。

Bit 2 **IRCSEL:** 红外载波极性选择。

1 = 当I/O引脚数据为0时产生红外载波。

0 = 当I/O引脚数据为1时产生红外载波。

Bit 1 **IRF57K:** 红外载波频率的选择。

1 = 红外载波频率为57KHz。

0 = 红外载波频率为38KHz。

Bit 0 **IREN:** 开启/关闭红外载波输出。

1 = 开启红外载波输出。

0 = 关闭红外载波输出。

**注意:**

1. 只有高频振荡 ( $F_{Hosc}$ ) (见 OSC 章节) 可以用作红外时钟源。

2. 不同振荡类型的分频比。

## 9. RFC

### 9.1 RFC (RFC控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFC	0x189	RFCEN	-	-	-	PSEL[3:0]			
读/写属性		读/写	-	-	-	读/写			
初始值		0	x	x	x	0			

Bit 7 **RFCEN:** 开启/关闭RFC功能。

1 = 开启RFC功能。

0 = 关闭RFC功能。

Bit 6:4 未生效的。

Bit 3:0 **PSEL[3:0]:** 选择RFC引脚。

PSEL[3:0]	RFC PAD
0000	PA0
0001	PA1
0010	PA2
0011	PA3
0100	PA4
0101	PA5
0110	PA6
0111	PA7
1000	PB0
1001	PB1
1010	PB2
1011	PB3
1100	PB4
1101	PB5
1110	PB6
1111	PB7

## 10. 低电压检测 (LVD)

### 10.1 概述

AT8BM84A低电压检测 (LVD) 内置精确带隙基准，精确检测VDD电平。如果LVDEN (寄存器PCON[5]) =1, CMP\_INV (寄存器CMPCR[4])=0, 且VDD电压值低于由RBIAS\_H, RBIAS\_L, LVDS[3:0]选择的LVD电压, 如下表所示, 则LVD输出变低。如果LVD中断被启用, LVD中断标志将是高, 如果GIE=1, 它将强制程序执行中断服务程序。此外, LVD的实际状态输出可以通过寄存器LVDCON[6]轮询。如果CMP\_INV (寄存器CMPCR[4])=1, 则在VDD电压值高于LVD电压的情况下, LVD中断标志为高。

LVD框图如下:

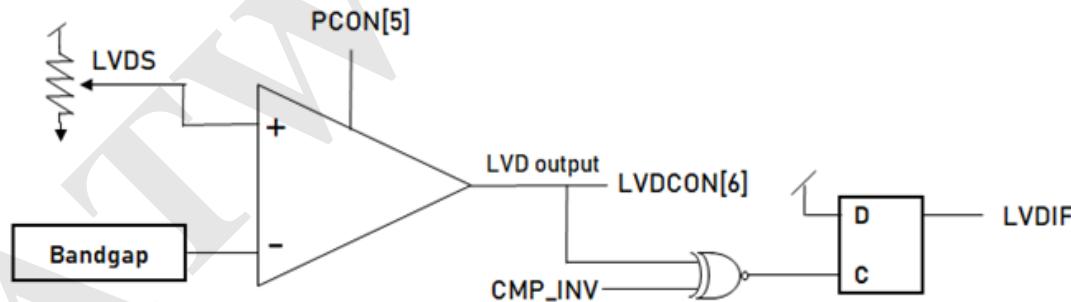


图 32 LVD结构框图

### 10.2 LVDCON (LVD控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDCON	0x89	-	LVDOUT	-	-	LVDS3	LVDS2	LVDS1	LVDS0
读/写属性		-	读	-	-	读/写	读/写	读/写	读/写
初始值		x	1	x	x	1	1	1	1

Bit 7 未生效的。

Bit 6 **LVDOUT**: 低压检测输出, 只读。

Bit 5:4 未生效的。

Bit 3:0 **LVDS[3:0]**: 选择LVD电压。

**注意:**

1. **LVD 的迟滞电压 (由低到高) 约为 0.1V。**

2. 在电池充电应用中 (检测电压由低到高), **LVD 电压选择表如下:**

RBIAS_H	RBIAS_L	LVDS[3:0]	VDD ↓	VDD ↑
0	1	0110	4.84V	(4.84+0.1) V
0	1	0101	4.54V	(4.54+0.1) V
1	1	1011	4.39V	(4.39+0.1) V
0	0	1111	4.15V	(4.15+0.1) V
0	0	1110	4.05V	(4.05+0.1) V
0	0	1101	3.90V	(3.90+0.1) V
0	0	1100	3.75V	(3.75+0.1) V
0	0	1011	3.60V	(3.60+0.1) V
0	1	0011	3.50V	(3.50+0.1) V
0	0	1010	3.45V	(3.45+0.1) V

RBIAS_H	RBIAS_L	LVDS[3:0]	VDD ↓	VDD ↑
0	0	1001	3.30V	(3.30+0.1) V
0	0	1000	3.15V	(3.15+0.1) V
0	1	0010	3.05V	(3.05+0.1) V
0	0	0111	3.00V	(3.00+0.1) V
0	0	0110	2.90V	(2.90+0.1) V
0	0	0101	2.80V	(2.80+0.1) V
0	1	0001	2.64V	(2.64+0.1) V
0	0	0100	2.60V	(2.60+0.1) V
1	0	1110	2.52V	(2.52+0.1) V
0	1	0000	2.46V	(2.46+0.1) V
0	0	0011	2.40V	(2.40+0.1) V
1	0	1100	2.33V	(2.33+0.1) V
1	0	1011	2.24V	(2.24+0.1) V
0	0	0010	2.20V	(2.20+0.1) V
1	0	1001	2.05V	(2.05+0.1) V
0	0	0001	2.00V	(2.00+0.1) V
0	0	0000	1.90V	(1.90+0.1) V

LVD控制流程如下：

- 步骤 1：通过LVDS[3:0]选择LVD电压
- 步骤 2：设置CMPCR = 0x0A
- 步骤 3：设置PCON[5]=1（启用LVD）
- 步骤 4：通过PCON1[6]检查LVD状态

**注意：如果LVD电压LVDS[3:0]被改变，用户必须等待至少 50us (@FHOSC=1MHz) 才能通过LVDCON[6]获得正确的LVD状态。**

例：

```

MOVIA      LVD_4P15V
MOVAR      LVDCON          ; Select LVD voltage
MOVIA      RBias_High_Dis | RBias_Low_Dis | CMPFINV_Dis | 0x0A
MOVAR      CMPCR           ; Set CMPCR (CMP_INV=0; CMPCR[3:0]=0xA)
BSR        PCON,LVDEN      ; Enable LVD

L_MAIN_LOOP:
CLRWDT
BTRSC      LVDCON,LVDOUT ; Clear watch dog
BSR        PORTB,0          ; If VDD < LVD, PB0=0, otherwise PB0=1
BTRSS      LVDCON,LVDOUT
BCR        PORTB,0
LGOTO     L_MAIN_LOOP

```

## 11. 电压比较器 (CMP)

### 11.1 概述

AT8BM84A提供1套具有各种模拟比较模式的电压比较器和内部参考电压。比较器非反相输入和反相输入可与GPIO共享。内部参考电压只能按路径发送到比较器的反相输入。

CMPEN(寄存器CMPCON[7])用于启用和禁用比较器。当CMPEN=0(默认)时, 比较器被禁用。当CMPEN=1时, 比较器启用。在睡眠模式下, 比较器被自动禁用。

比较器结构如下图所示:

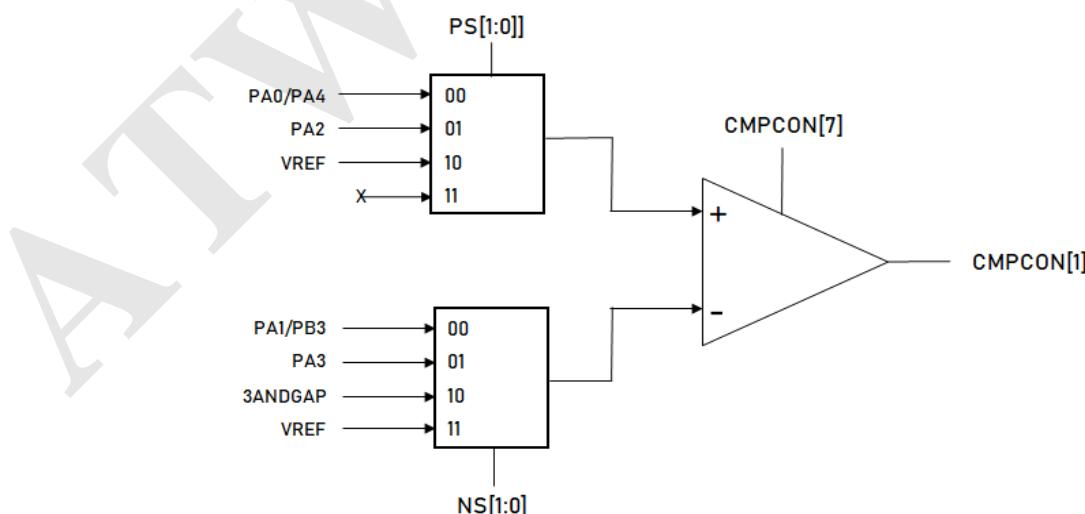


图 33 比较器硬件连接

### 11.2 比较器参考电压 (Vref)

内部参考电压Vref由串联电阻构成, 提供不同等级的参考电压。RBIAS\_H和RBIAS\_L用于选择Vref的最大值和最小值, LVDS[3:0]用于选择16个电压中的一个。

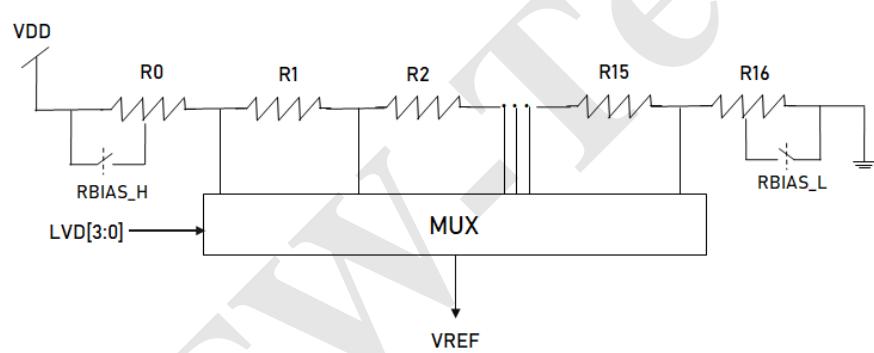


图 34 Vref硬件连接

Vref由RBIAS\_H, RBIAS\_L和LVDS[3:0]决定。LVDS[3:0]用于从16个参考电压中选择一个, 如下表所示。

RBIAS[H:L]	LVDS[3:0]	V <sub>IN</sub> =V <sub>ref</sub>	Unit
10	0000	0.508*VDD	
10	0001	0.483*VDD	
10	0010	0.439*VDD	
11	0000	0.420*VDD	
10	0011	0.402*VDD	
11	0001	0.390*VDD	
10	0100	0.371*VDD	
10	0101	0.345*VDD	
10	0110	0.333*VDD	
10	0111	0.322*VDD	
10	1000	0.306*VDD	
00	0001	0.300*VDD	
10	1001	0.292*VDD	
10	1010	0.280*VDD	
00	0010	0.273*VDD	
10	1011	0.268*VDD	
10	1100	0.257*VDD	
10	1101	0.247*VDD	
10	1110	0.238*VDD	
10	1111	0.233*VDD	
01	0001	0.227*VDD	
11	0110	0.213*VDD	
00	0110	0.207*VDD	
11	0111	0.200*VDD	
00	1000	0.190*VDD	
00	1001	0.182*VDD	
00	1010	0.174*VDD	
00	1011	0.167*VDD	
00	1100	0.160*VDD	
00	1101	0.154*VDD	
00	1110	0.148*VDD	
11	1011	0.137*VDD	
01	0101	0.132*VDD	
01	0110	0.124*VDD	
01	0111	0.116*VDD	
01	1000	0.106*VDD	
11	1110	0.102*VDD	
01	1001	0.096*VDD	
01	1010	0.088*VDD	
01	1011	0.080*VDD	
01	1100	0.072*VDD	
01	1101	0.065*VDD	
01	1110	0.059*VDD	
01	1111	0.055*VDD	

表 11-1 参考电压V<sub>ref</sub>选择表

获得比较器输出结果的方法有两种：一种是通过寄存器轮询，另一种是通过探测输出引脚。

比较器输出可以通过CMPOUT（寄存器CMPCON[1]）轮询。

要在输出引脚探测比较器输出，将CMPOE（寄存器CMPCON[0]）设置为1，则PB3将是比较器输出的实时状态。需要注意的是，当CMPOE=1时，如果PWM3功能是启用状态，则该功能将被禁用。

### 11.3 比较器控制寄存器

#### 11.3.1 CMPCON (比较器控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCON	0x9C	CMPEN	BIASEN	-	-	-	-	CMPOUT	CMPOE
读/写属性		读/写	读/写	-	-	-	-	读	读/写
初始值		0	0	x	x	xx	x		0

Bit 7 **CMPEN:** 开启/关闭比较器。

1 = 开启比较器。

0 = 关闭比较器。

Bit 6 **BIASEN:** 开启/关闭比较器/ LVD模块偏置。

1 = 开启比较器/ LVD模块偏置。

0 = 关闭比较器/ LVD模块偏置。

Bit 5:2 未生效的。

Bit 1 **CMPOUT:** 比较器输出状态, 只读。

Bit 0 **CMPOE:** 开启/关闭比较器输出到PB3 引脚。

1 = 开启比较器输出到PB3 引脚。

0 = 关闭比较器输出到PB3 引脚。

**注意: 比较器输出到 PB3 引脚的优先级高于 buzzer1 输出。**

#### 11.3.2 CMPCR (比较器电压选择控制寄存器)

名称	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCR	0x9D	-	RBIAS_H	RBIAS_L	CMP_INV	PS1	PS0	NS1	NS0
读/写属性		-				读/写			
初始值		x	0	0	0	1	1	0	0

Bit 7 未生效的。

Bit 6:5 **RBIAS[H:L]:** 设置相应的电压基准电平。

Bit 4 **CMPF\_INV:** 比较器反向输出控制位。

Bit 3:2 **PS[1:0]:** 比较器正向输入选择。

PS[1:0]	正向输入
00	PA0
01	PA2
10	Vref
11	---

Bit 1:0 **NS[1:0]:** 比较器反向输入选择。

NS[1:0]	反向输入
00	PA1
01	PA3
10	Bandgap (0.6V)
11	Vref

**Example: (For P2P Mode)**

```

MOVIA    PA0_Input | PA1_Input
MOVAR    IOSTA
MOVIA    RBias_High_Dis | RBias_Low_Dis| CMPFINV_Dis
MOVAR    CMPCR
BSR      CMPCON,CMPEN
BCR      PIR2, CMPIF
BSR      PIE2, CMPIE
BSR      INTCON, PEIE
BSR      INTCON,GIE

L_MAIN_LOOP:
CLRWDT
BTRSC    CMPCON,CMPOUT
BSR      PORTB,7
BTRSS    CMPCON,CMPOUT
BCR      PORTB,7
LGOTO    L_MAIN_LOOP

ISR:
BTRSS    PIR2, CMPIF      ; Check comparator interrupt flag
LGOTO    ISR_Exit
MOVIA    0x40
XORAR    PORTB,1          ; PB6 toggle output, when compare occur
BCR      PIR2, CMPIF      ; Clear comparator interrupt flag

ISR_Exit:
RETIE

```

**Example: (For P2V Mode)**

```

MOVIA    PA0_Input
MOVAR    IOSTA
MOVIA    RBias_High_En | RBias_Low_Dis| CMPFINV_Dis
MOVAR    CMPCR
BSR      CMPCON,CMPOE
BSR      CMPCON,CMPEN
BCR      PIR2, CMPIF
BSR      PIE2, CMPIE
BSR      INTCON, PEIE
BSR      INTCON,GIE

L_MAIN_LOOP:
CLRWDT
BTRSC    CMPCON,CMPOUT
BSR      PORTB,7
BTRSS    CMPCON,CMPOUT
BCR      PORTB,7
LGOTO    L_MAIN_LOOP

ISR:
BTRSS    PIR2, CMPIF      ; Check comparator interrupt flag
LGOTO    ISR_Exit
MOVIA    0x40
XORAR    PORTB,1          ; PB6 toggle output, when compare occur
BCR      PIR2, CMPIF      ; Clear comparator interrupt flag

ISR_Exit:
RETIE

```

## 12. 模拟-数字转换器 (ADC)

### 12.1 概述

AT8BM84A提供22+2通道12位SAR ADC，将模拟信号转换为12位数字信号。ADC高参考电压是可选的。它们可以是来自PA0, PB1的外部电压，或内部产生的电压VDD, 4V, 3V或2V。模拟输入可选择来自于模拟信号输入引脚PA0~PA4、PA6~PA7、PB0~PB7、PC0~PC5，来自于内部生成的 $1/4 * VDD$ 或VSS。ADC时钟ADCLK可选择Fcpu/1, Fcpu/2, Fcpu/8或Fcpu/16。采样脉冲宽度可以选择作为ADCLK\*1, ADCLK\*2, ADCLK\*4或ADCLK\*8。ADC运行前设置ADEN=1。然后设置START (ADMD [6])=1, ADC将开始将模拟信号转换为数字信号。EOC=0表示ADC正在处理中。EOC=1表示ADC转换结束。如果ADIE=1并且全局中断被启用，则ADC中断将在EOC由低变高后发出。结构框图如下图所示。

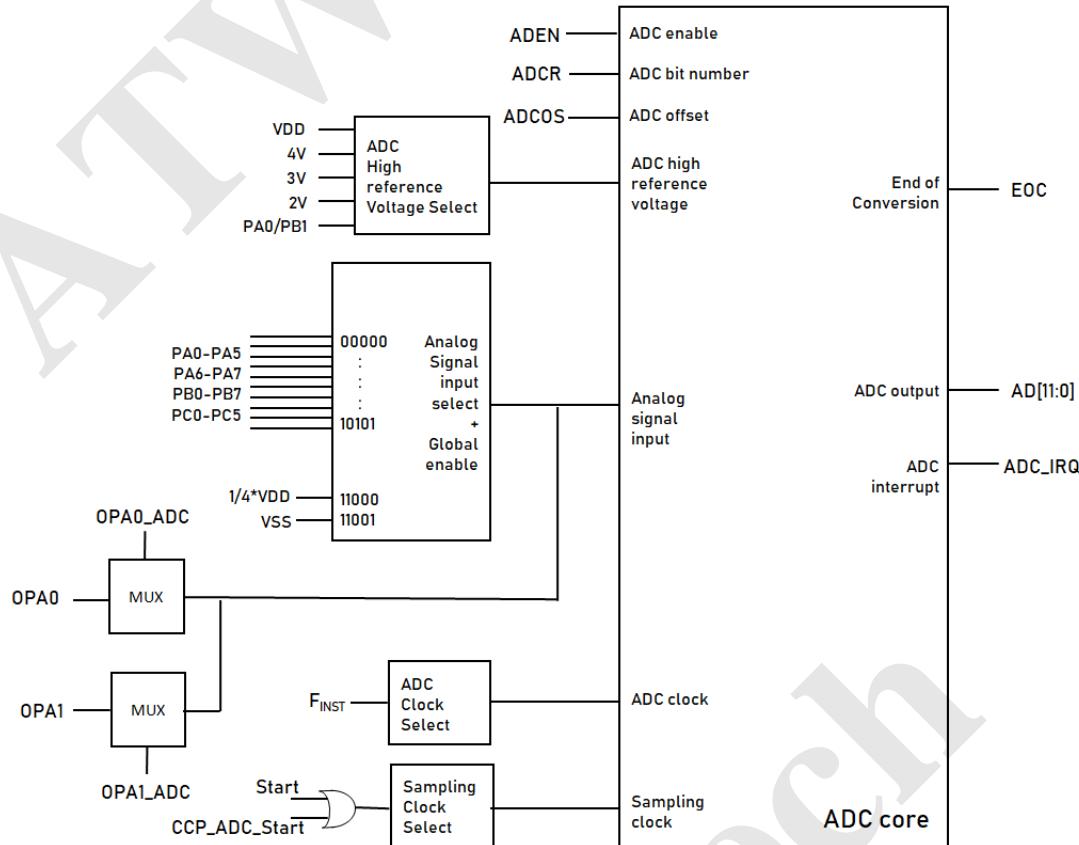


图 35 ADC结构框图

### 12.2 ADC参考电压

ADC内置5个高参考电压源，由ADVREFH寄存器控制（见下表）。这些高参考电压源是外部电压源（PA0或PB1可选）和四个内部电压源（VDD, 4V, 3V, 2V）。当EVHENB位为“1”时，ADC参考电压来自外部电压源。在这种模式下，PA0或PB1必须是介于VDD和2V之间的电压。如果EVHENB位为0，则ADC参考电压来自VHS[1:0]选择的内部电压源。如果VHS[1:0]为“11”，则ADC参考电压为VDD。如果VHS[1:0]为“10”，则ADC参考电压为4V。如果VHS[1:0]为“01”，则ADC参考电压为3V。如果VHS[1:0]为“00”，则ADC参考电压为2V。对于内部参考电压为4V/3V/2V的应用，VDD不能低于所选的内部电压4V、3V或2V。ADC输入电压范围是从VSS到高参考电压。

EVHENB	VHS[1:0]	参考电压
1	xx	PA0 or PB1 (by option)
0	11	VDD
0	10	4V
0	01	3V
0	00	2V

表 12-1 ADC参考电压选择

### 12.3 ADC模拟输入通道

ADC输入引脚与数字I/O引脚共享。将模拟信号连接到这些引脚可能会导致I/O引脚的额外电流泄漏。在断电模式下，上述漏电流将是一个大问题。将“1”写入PxCON寄存器位，将相关的PAx/PBx/PCx（由PxSEL[1:0]选择）引脚配置为纯模拟输入引脚，以避免电流泄漏，一旦设置就不能作为正常I/O使用。

除了设置PxCON寄存器位外，所选的模拟输入引脚必须设置为输入模式，并且必须关闭内部上拉/下拉，否则可能影响模拟输入电平。

ADC使用CHS[4:0]选择模拟输入源。

ADCEN	GCHS	CHS[4:0]	ADC模拟输入源
0	x	XXXXX	x
x	0	XXXXX	x
1	1	00000	PA0
1	1	00001	PA1
1	1	00010	PA2
1	1	00011	PA3
1	1	00100	PA4
1	1	00101	PA5
1	1	00110	PA6
1	1	00111	PA7
1	1	01000	PB0
1	1	01001	PB1
1	1	01010	PB2
1	1	01011	PB3
1	1	01100	PB4
1	1	01101	PB5
1	1	01110	PB6
1	1	01111	PB7
1	1	10000	PC0
1	1	10001	PC1
1	1	10010	PC2
1	1	10011	PC3
1	1	10100	PC4
1	1	10101	PC5
1	1	11000	1 / 4 * VDD
1	1	11001	VSS

表 12-2 ADC模拟输入源选择

## 12.4 ADC时钟 (ADCLK)、采样时钟 (SHCLK) 和位数选择

ADC时钟 (ADCLK)、采样脉宽 (SHCLK) 和转换位数的选择影响转换速度和转换精度。ADCLK是ADC的基准时钟。在SAR ADC的工作过程中，位操作与ADCLK同步。SHCLK是模拟信号采样时间的持续时间，较大的SHCLK会更接近模拟信号的原始电平，但会减慢ADC的转换速度。亦然。ADC可以根据ADCR[1:0]寄存器位选择不同的转换位数。有 2 bit 数字可供选择，分别是 12-bit、10-bit 和 8-bit。转换位数越少，ADC的转换速率越快，但有效的ADC 位数就越少。转换位数越多，转换速率越慢，但转换精度越高。

ADC时钟来自Fcpu，可从ADCK[1:0]中选择。

ADCK[1:0]	ADC时钟
00	F <sub>CPU</sub> /16
01	F <sub>CPU</sub> /8
10	F <sub>CPU</sub> /1
11	F <sub>CPU</sub> /2

表 12-3 ADC时钟选择

采样时钟宽度来自ADCLK，可从SHCK[1:0]中选择。

SHCK[1:0]	采样时钟
00	1 ADCLK
01	2 ADCLK
10	4 ADCLK
11	8 ADCLK

表 12-4 ADC采样时钟选择

ADC位数选择来自ADCR[1:0]。

ADCR[1:0]	转换位数
00	8-bit
01	10-bit
1x	12-bit

表 12-5 转换位数选择

ADC转换时间从START (ADC转换开始) 到EOC=1 (ADC转换结束)。持续时间取决于ADC分辨率、ADC时钟速率和采样时钟宽度。

**ADC转换时间≈采样时钟宽度+ (ADC 位数+ 2)\*ADCLK宽度**

下表是ADC转换时间和转换速率的一些示例。

Bit No.	ADC 时钟	SHCLK	转换时间 (ADCLK No.)	Fcpu=2MHz		Fcpu=250K	
				时间	速率	时间	速率
12	Fcpu/16	8 ADCLK	22	176us	5.68kHz	1408us	710Hz
12	Fcpu/1	1 ADCLK	15	7.5us	133.3kHz	60us	16.7kHz
10	Fcpu/1	1 ADCLK	13	6.5us	153.8kHz	52us	19.2kHz
8	Fcpu/1	1 ADCLK	11	5.5us	181.8kHz	44us	22.7kHz

表 12-6 ADC转换时间

## 12.5 ADC操作过程

设置ADC时钟（ADCLK）、采样时钟宽度（SHCLK）、转换位数（ADCR）、ADC高参考电压（ADVREFH）、选择输入通道和PACON或PBCON相关位。然后设置ADEN=1。

设置ADEN=1后，ADC工作之前至少要等待 256us（ADC内部偏置稳定时间）。将START写入1以启动ADC转换会话。ADC处理期间EOC=0。轮询EOC=1或在ADC转换结束时等待ADC中断。

## 12.6 ADC数据格式

ADC 12Bit数据输出形式[ADDH+ADDL]。用户可以使用ADFM设置ADC输出数据[11:0]。下图为AT8BM84A在不同转换位数下的A/D结果数据格式。

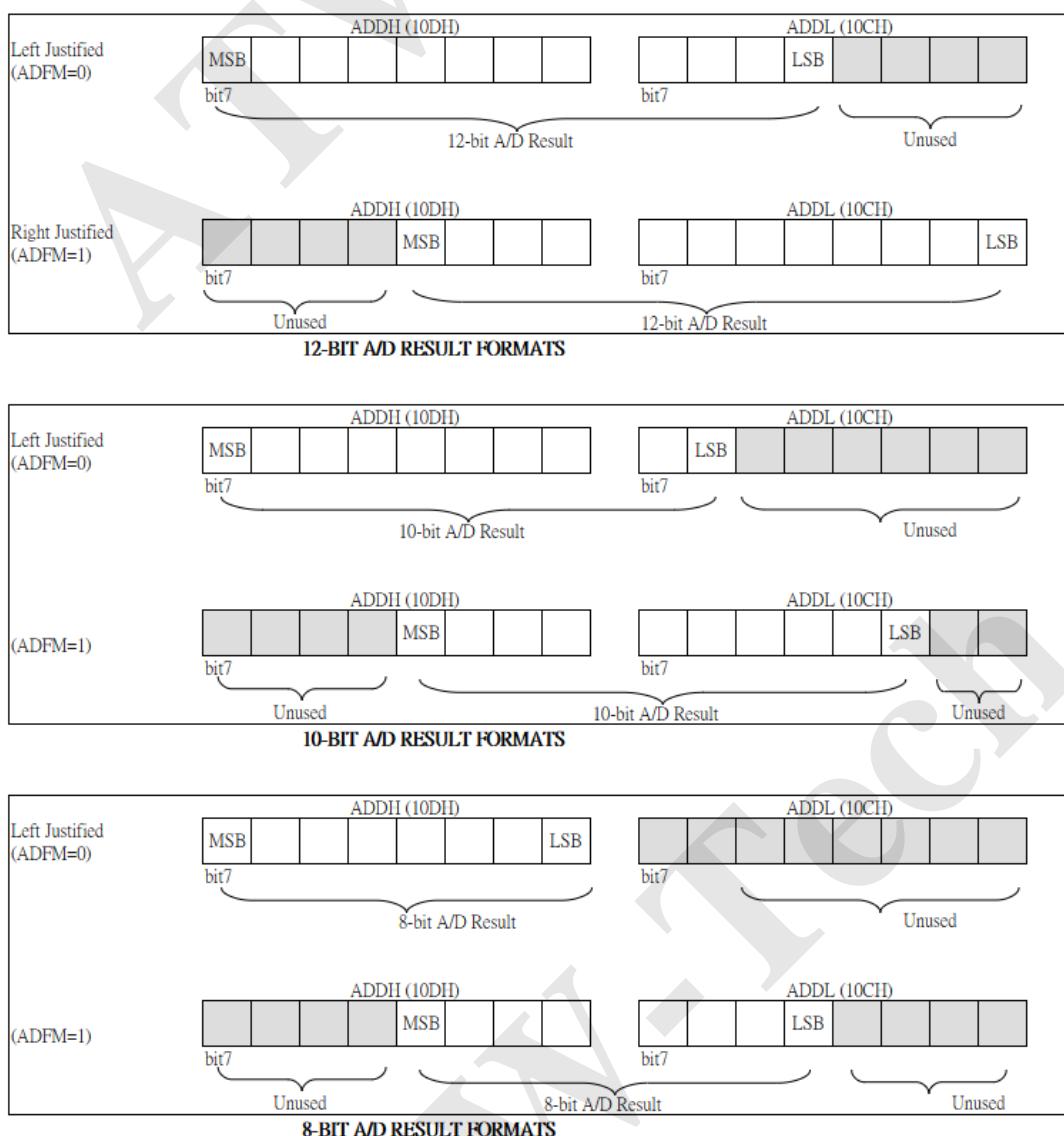


图 36 ADFM设置

## 12.7 ADC控制寄存器

## 12.7.1 ADMD (ADC 模式寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADMD	0x109	ADEN	START	GCHS	CHS4	CHS3	CHS2	CHS1	CHS0
读/写属性		读/写	写	读	读/写	读/写	读/写	读/写	读/写
初始值		0	0	0	0	0	0	0	0

Bit 7      **ADEN:** ADC开启位。

1 = ADC开启。

0 = ADC关闭。

Bit 6      **START:** 启动ADC转换会话。

将 1 写入此位将启动ADC转换。这个位是只写的。读这个位会得到 0。

Bit 5      **GCHS:** ADC全局通道选择位。

1 = 开启所有ADC输入通道。

0 = 关闭所有ADC通道。

Bit 4:0      **CHS[4:0]:** 比较器输出反向控制位。

00000=选择PA0 作为ADC输入。

00001=选择PA1 作为ADC输入。

00010=选择PA2 作为ADC输入。

00011=选择PA3 作为ADC输入。

00100=选择PA4 作为ADC输入。

00110=选择PA6 作为ADC输入。

00111=选择PA7 作为ADC输入。

01000=选择PB0 作为ADC输入。

01001=选择PB1 作为ADC输入。

01010=选择PB2 作为ADC输入。

01011=选择PB3 作为ADC输入。

01100=选择PB4 作为ADC输入。

01101=选择PB5 作为ADC输入。

01110=选择PB6 作为ADC输入。

01111=选择PB7 作为ADC输入。

10000=选择PC0 作为ADC输入。

10001=选择PC1 作为ADC输入。

10010=选择PC2 作为ADC输入。

10011=选择PC3 作为ADC输入。

10100=选择PC4 作为ADC输入。

10101=选择PC5 作为ADC输入。

11100=选择 1/4 VDD作为ADC输入。

11101=选择VSS作为ADC输入。

## 12.7.2 ADDL (ADC LSB 输出寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDL	0x10c								ADDL[7:0]
读/写属性								读	
初始值									XXXXXXXX

Bit 7:0 **ADDL[7:0]**: 低字节ADC数据缓冲区。

## 12.7.3 ADDH (ADC 输出数据寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDH	0x10d								ADDH[7:0]
读/写属性								读	
初始值									XXXXXXXX

Bit 7:0 **ADDH[7:0]**: 高字节ADC数据缓冲区。

## 12.7.4 ADCON1 (ADC 控制寄存器 1)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	0x10e	EVHENB	-	EOC	ADFM	ADCK1	ADCK0	VHS1	VHS0
读/写属性		读/写	-	写	读/写	读/写	读/写	读/写	读/写
初始值		0	x	1	0	0	0	1	1

Bit 7 **EVHENB**: ADC参考高电压 (VREFH) 选择控制位。

1 = ADC参考高压由外部引脚PA0 提供。

0 = ADC参考高压是内部产生的, 所选电压取决于VHS1~0。

Bit 6 未生效的。

Bit 5 **EOC**: ADC状态位, 只读。

1 = ADC转换结束, ADC数据在ADR和ADD中是可用的。

0 = ADC正在进行中。

Bit 4 **ADFM**: ADC结果格式选择位。

1 = 右对齐。

0 = 左对齐。

Bit 3:2 **ADCK[1:0]**: ADC时钟选择。

11 = ADC clock=FCPU/2

10 = ADC clock=FCPU/1

01 = ADC clock=FCPU/8

00 = ADC clock=FCPU/16

Bit 1:0 **VHS[1:0]**: ADC内部基准高压选择位。

11 = VREFH=VDD

10 = VREFH=4V

01 = VREFH=3V,

00 = VREFH=2V

## 12.7.5 ADJMD (ADC 模拟引脚寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADJMD	0x10f	-	-	ADJ_SIGN	ADJ4	ADJ3	ADJ2	ADJ1	ADJ0
读/写属性	-	-	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值	x	x	0	0	0	0	0	0	0

Bit 7:6 未生效的。

Bit 5 **ADJ\_SIGN**: ADC调校标志位。

1 = ADC数据增加。

0 = ADC数据减小。

Bit 4:0 **ADJ[4:0]**: ADC调校补偿。

11111 = 补偿 16mV

00000 = 补偿 0mV

## 12.7.6 ADCR (采样脉冲和 ADC 位数寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCR	0x111	PxSEL1	PxSEL0	-	-	SHCK1	SHCK0	ADCR1	ADCR0
读/写属性	读/写	读/写	-	-	读/写	读/写	读/写	读/写	读/写
初始值	0	0	x	x	1	0	1	0	0

Bit 7:6 **PxSEL[1:0]**: 选择虚拟SFR PxCON寄存器到PACON、PBCON或PCCON的映射位。

10 = PORTC通过虚拟SFR PxCON模拟引脚设置 (PCCON)。

01 = PORTB通过虚拟SFR PxCON模拟引脚设置 (PBCON)。

00 = PORTA通过虚拟SFR PxCON模拟引脚设置 (PACON)。

Bit 5:4 未生效的。

Bit 3:2 **SHCK[1:0]**: ADC采样脉冲宽度选择。

11 = 8 个ADC clock。

10 = 4 个ADC clock。

01 = 2 个ADC clock。

00 = 1 个ADC clock。

Bit 1:0 **ADCR[1:0]**: ADC转换位数选择。

1x = 12-bit ADC.

01 = 10-bit ADC.

00 = 8-bit ADC.

## 13. 看门狗 (WDT)

### 13.1 概述

AT8BM84A中有一个自由运行的振荡器被WDT所使用。由于该振荡器与其它振荡电路无关，故在待机模式和睡眠模式中WDT仍能继续工作。

WDT能被配置字开启或关闭。当WDT被配置字开启时，在程序运行过程中，WDT仍然可以通过WDTEN位（寄存器PCON[7]）来控制。此外，WDT超时后可由另一个配置字决定复位AT8BM84A或发出的中断请求。同时，在WDT超时后，寄存器/TO（STATUS[4]）位将被清除为0。

WDT超时的时基可以是3.5毫秒、15毫秒、60毫秒或250毫秒，由两个配置字决定。如果将预分频器0分配给WDT，则可以延长超时周期。通过将1写入寄存器PS0WDT位，预分频器0将分配给WDT。预分频器0对WDT的分频比由寄存器PS0SEL[2:0]位决定，而且取决于WDT的超时机制。分频比从1:1到1:128，如果WDT超时，将使AT8BM84A复位。分频比从1:2到1:256，如果WDT超时，将使AT8BM84A中断。

当预分频器0分配给WDT时，执行CLRWDT指令将清除WDT、预分频器0。并设置/TO标志位为1。

如果用户选择WDT中断机制，在WDT超时后，寄存器WDTIF（PIR[5]）位将设置为1。如果寄存器WDTIE（PIE[5]）位和GIE位都设置为1，则可能产生中断请求。直到程序将0写入WDTIF，WDTIF才会被清除为0。

## 14. 中断

### 14.1 概述

AT8BM84A提供两种中断：一个是软件中断，另一个是硬件中断。

软件中断是由INT指令的执行引起的。有 14 个硬件中断：

- Timer0 上溢中断。
- Timer1 下溢中断。
- Timer4 下溢中断。
- Timer5 下溢 / CCP中断。
- WDT 超时中断。
- PA/PB/PC 输入状态改变中断。
- 外部中断 0。
- 外部中断 1。
- 外部中断 2。
- LVD中断。
- 比较器输出状态改变中断。
- ADC转换结束中断。
- SIM中断（串行接口模式中断）。

GIE是全局中断使能标志。它必须为 1 才能启用硬件中断功能。

同时，GIE由AT8BM84A自动清零。软件中断的中断服务程序的最后一条指令必须是RETIE。执行此指令会将GIE设为 1，并返回原来的执行顺序。

当发生任何硬件中断时，相应的中断标志位将被设置为 1。在软件将 0 写入此位之前，此位不会被清除。因此，用户可以通过轮询相应的中断标志位来获取导致硬件中断的事件信息。注意，只有当相应的中断使能位设置为 1 时，才会读取相应的中断标志。如果相应的中断使能位设置为 1，并且GIE也为 1，则会发生硬件中断，从 0x004 取出下一条指令。同时，AT8BM84A将自动清除寄存器位GIE。

#### 14.1.1 Timer0 上溢中断

Timer0 上溢（从 0x00 到 0xFF）将设置寄存器位T0IF。如果T0IE和GIE设置为 1，这个中断请求将被处理。

#### 14.1.2 Timer1 下溢中断

Timer1 下溢（从 0x3FF 到 0x00）将设置寄存器位T1IF。如果T1IE和GIE设置为 1，这个中断请求将被处理。

#### 14.1.3 Timer4 下溢中断

Timer4 下溢（从 0x3FF 到 0x00）将设置寄存器位T4IF。如果T4IE和GIE设置为 1，这个中断请求将被处理。

#### 14.1.4 Timer5 下溢中断

Timer5 下溢（从 0x3FF 到 0x00）将设置寄存器位 T5IF。如果 T5IE 和 GIE 设置为 1，这个中断请求将被处理。

#### 14.1.5 Timer5 下溢 / CCP 中断

Timer5 下溢（从 0x3FF 到 0x00）将设置寄存器位 T5IF。如果 T5IE 和 GIE 设置为 1，这个中断请求将被处理。

#### 14.1.6 WDT 超时中断

当 WDT 超时时，配置字选择 WDT 超时将产生中断请求，它将设置寄存器位 WDTIF。如果 WDTIE 和 GIE 设置为 1，这个中断请求将被处理。

#### 14.1.7 PA/PB/PC 输入状态改变中断

当将 PA<sub>x</sub>, 0 ≤ x ≤ 7, PB<sub>y</sub>, 0 ≤ y ≤ 7, PC<sub>z</sub>, 0 ≤ z ≤ 5, 配置为输入引脚，并且对应的寄存器位 WUPA<sub>x</sub>, WUPB<sub>x</sub> 设置为 1 时，在所选 I/O 引脚上的电平变化将设置寄存器位 PABCIF。如果 PABCIE 和 GIE 被设置为 1，这个中断请求将被处理。注意当 PA3、PA4 或 PA5 同时被设置为电平变化中断和外部中断时，外部中断使能 EIS0, EIS1 或 EIS2=1 将会禁止 PA3、PA4 或 PA5 的电平变化操作。

#### 14.1.8 外部中断 0

根据 EIS0=1 和 INTEDG 的配置，I/O 引脚 PA4 上选择的有效边沿将设置寄存器位 INT0IF，如果 INT0IE 和 GIE 设置为 1，则该中断请求将被处理。

#### 14.1.9 外部中断 1

根据 EIS1=1 和 INTEDG 的配置，I/O 引脚 PA3 上选择的有效边沿将设置寄存器位 INT1IF，如果 INT1IE 和 GIE 设置为 1，则该中断请求将被处理。

#### 14.1.10 外部中断 2

根据 EIS2=1 和 INTEDG 的配置，I/O 引脚 PA5 上选择的有效边沿将设置寄存器位 INT2IF，如果 INT2IE 和 GIE 设置为 1，则该中断请求将被处理。

#### 14.1.11 LVD 中断

当 VDD 电压低于 LVD 电压时，LVD 标志由高变低，并设置寄存器位 LVDIF=1。如果 LVDIE 和 GIE 设置为 1，这个中断请求将被处理。

#### 14.1.12 比较器输出状态改变中断

每当比较器输出状态发生变化时，就会触发比较器中断。如果 CMPIE 和 GIE 设置为 1，该中断请求将被处理。注意，在比较器中断发生之前，需要读取寄存器 OSCCR 来清除先前的比较器输出状态差异。

#### 14.1.13 ADC 转换结束中断

当 ADC 转换结束信号发出时，ADC 中断被触发。如果 ADIE 和 GIE 被设置为 1，这个中断请求将被处理。

#### 14.1.14 串行接口模式中断

当 SPI 模式的 SPIF 或 I<sup>2</sup>C 模式的 MIF 发出时，SIM 中断被触发。如果 SIMIE 和 GIE 设置为 1，这个中断请求将被处理。

## 14.2 中断控制寄存器

### 14.2.1 INTCON (中断控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCON	0xB	GIE	PEIE	TOIE	INT0IE	PABCIE	TOIF	INT0IF	PABCIF
读/写属性		读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值		0	0	0	0	0	0	0	0

Bit 7 **GIE:** 全局中断使能位。

1 = 打开所有中断。

0 = 关闭所有中断。

Bit 6 **PEIE:** 外围中断使能位。外围中断被定义为除了timer0 中断、外部 0 中断和pad中断之外的所有中断。

1 = 打开所有外围中断。

0 = 关闭所有外围中断。

Bit 5 **TOIE:** Timer0 溢出中断使能位。

1 = 打开TMR0 中断。

0 = 关闭TMR0 中断。

Bit 4 **INT0IE:** 外部中断 0 使能位。

1 = 打开外部中断 0。

0 = 关闭外部中断 0。

Bit 3 **PABCIE:** PORTA/PORPB/PORPC 变化中断使能位。

1 = 打开PORTA/PORPB/PORPC 变化中断。

0 = 关闭PORTA/PORPB/PORPC 变化中断。

Bit 2 **TOIF:** Timer0 中断标志位。

1 = TMR0 溢出。

0 = TMR0 未溢出。

Bit 1 **INT0IF:** 外部中断 0 标志位。

1 = 外部中断 0 发生。

0 = 外部中断 0 未发生。

Bit 0 **PABCIF:** PORTA/PORPB/PORPC 变化中断标志位。

1 = PORTA/PORPB/PORPC 状态发生改变。

0 = PORTA/PORPB/PORPC 状态未发生改变。

### 14.2.2 PIR1 (中断标志寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIR1	0xC	INT2IF	INT1IF	WDTIF	-	-	T5IF/C CPIF	T4IF	T1IF
读/写属性		读/写	读/写	读/写	-	-	读/写	读/写	读/写
初始值(note*)		0	0	0	x	x	0	0	0

Bit 7 **INT2IF:** 外部中断 2 标志位。

1 = 外部中断 2 发生 (INT2IF必须通过软件清零)。

0 = 外部中断 2 未发生。

Bit 6 **INT1IF:** 外部中断 1 标志位。

1 = 外部中断 1 发生 (INT1IF必须通过软件清零)。

0 = 外部中断 1 未发生。

Bit 5 **WDTIF:** WDT超时中断标志位。

		1 = WDT超时中断发生 (WDTIF必须通过软件清零)
		0 = WDT超时中断未发生。
Bit 4:3		未生效的。
Bit 2		<b>T5IF/CCPIF:</b> Timer5 或 CCP中断标志。
		1 = Timer5 或 CCP中断发生 (T5IF/CCPIF必须通过软件清零)
		0 = Timer5 或 CCP中断未发生。
Bit 1		<b>T4IF:</b> Timer4 中断标志。
		1 = Timer4 中断发生 (T4IF必须通过软件清零)
		0 = Timer4 中断未发生。
Bit 0		<b>T1IF:</b> Timer1 中断标志。
		1 = Timer1 中断发生 (T1IF必须通过软件清零)
		0 = Timer1 中断未发生。

**注意：当对应的 INT<sub>E</sub> 位未使能时，读中断标志为 0。**

#### 14.2.3 PIE1 (中断使能寄存器 1)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIE1	0x8C	INT2IE	INT1IE	WDTIE	-	-	T5IE/CCPIE	T4IE	T1IE
读/写属性	读/写	读/写	读/写	-	-	读/写	读/写	读/写	读/写
初始值	0	0	0	x	x	0	0	0	0

Bit 7	<b>INT2IE:</b> 外部中断 2 使能位。 1 = 外部中断 2 开启。 0 = 外部中断 2 关闭。
Bit 6	<b>INT1IE:</b> 外部中断 1 使能位。 1 = 外部中断 1 开启。 0 = 外部中断 1 关闭。
Bit 5	<b>WDTIF:</b> WDT超时中断使能位。 1 = WDT超时中断开启。 0 = WDT超时中断关闭。
Bit 4:3	未生效的。
Bit 2	<b>T5IE/CCPIE:</b> Timer5 或 CCP中断使能位。 1 = Timer5 或 CCP中断开启。 0 = Timer5 或 CCP中断关闭。
Bit 1	<b>T4IE:</b> Timer4 下溢中断使能位。 1 = Timer4 下溢中断开启。 0 = Timer4 下溢中断关闭。
Bit 0	<b>T1IF:</b> Timer1 下溢中断使能位。 1 = Timer1 下溢中断开启。 0 = Timer1 下溢中断关闭。

## 14.2.4 PIR2 (中断标志寄存器 2)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIR2	0xD	ADIF	LVDIF	CMPIF	-	SIMIF	LSRIF	TXIF	RXIF
读/写属性		读/写	读/写	读/写	-	读	读	读	读
初始值 (注意*)		0	0	0	x	0	0	1	0

Bit 7 **ADIF:** ADC转换结束中断标志位。

1 = ADC转换结束中断发生 (ADIF必须通过软件清零)

0 = ADC转换结束中断未发生。

Bit 6 **LVDIF:** LVD中断标志位。

1 = LVD中断发生 (LVDIF必须通过软件清零)

0 = LVD中断未发生。

Bit 5 **CMPIF:** 比较器中断标志位。

1 = 比较器中断发生 (CMPIF必须通过软件清零)

0 = 比较器中断未发生。

Bit 4 未生效的。

Bit 3 **SIMIF:** 串口中断标志位 (I<sup>2</sup>C模式或SPI模式)。

1 = 串口中断发生 (SIMIF必须通过软件清零)

0 = 串口中断未发生。

如果SIMCR.MEM=1, SIMIF显示MIF的状态。

如果SIMCR.SPE=1, SIMIF显示SPIF的状态。

Bit 2 **LSRIF:** LSR中断标志位。

1 = 行状态中断发生 (LSRIF必须通过软件清零)

( BKINT=1 或 FERR=1 或 PERR=1 或 OERR=1 )

0 = 行状态中断未发生。

Bit 1 **TXIF:** 发送保持寄存器 (THR) 空标志位。

TXIF显示LSR的状态。

这个位表示控制器已准备好接收一个新字符进行传输。当一个字符从发送保持寄存器 (THR) 转移到发送移位寄存器 (TBR) 时, 它被设置为逻辑 1。向TBR写入数据将清除此标志。

Bit 0 **RXIF:** 数据就绪标志位。

RXIF显示LSR.RREADY的状态。

每当接收到一个完整的传入字符并将其传输到接收缓冲寄存器 (RBR) 时, 它被设置为逻辑 1。读取RBR数据将清除此标志。

**注意: 当对应的 INTE 位未使能时, 读中断标志 (ADIF, LVDIF, CMPIF) 为 0。**

## 14.2.5 PIE2 (中断使能寄存器 2)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIE2	0x8D	ADIE	LVDIE	CMPIE	-	SIMIE	LSRIE	TXIE	RXIE
读/写属性		读/写	读/写	读/写	-	读/写	读/写	读/写	读/写
初始值		0	0	0	x	0	0	0	0

Bit 7	<b>ADIE:</b> ADC转换结束中断使能位。 1 = ADC转换结束中断开启。 0 = ADC转换结束中断关闭。
Bit 6	<b>LVDIE:</b> LVD中断使能位。 1 = LVD中断开启。 0 = LVD中断关闭。
Bit 5	<b>CMPIE:</b> 比较器中断使能位。 1 = 比较器中断开启。 0 = 比较器中断关闭。
Bit 4	未生效的。
Bit 3	<b>SIMIE:</b> 串口中断使能位 (I <sup>2</sup> C模式或SPI模式)。 1 = 串口中断开启 (I <sup>2</sup> C模式或SPI模式)。 如果SIMCR.MEM=1, I <sup>2</sup> C中断开启。 如果SIMCR.SPE=1, SPI中断开启。 0 = 串口中断关闭。
Bit 2	<b>LSRIE:</b> LSR中断使能位。 1 = 接收行状态中断开启。 0 = 接收行状态中断关闭。
Bit 1	<b>TXIE:</b> 发送保持寄存器 (THR) 空中断使能位。 1 = 发送保持寄存器 (THR) 空中断开启。 0 = 发送保持寄存器 (THR) 空中断关闭。
Bit 0	<b>RXIE:</b> 完全接收 1byte中断使能位。 1 = 完全接收 1byte中断开启。 0 = 完全接收 1byte中断关闭。

#### 14.2.6 PCON (Power 寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	0x8f	WDTEN	-	LVDEN	-	LVREN	-	-	-
读/写属性	读/写	-	读/写	-	读/写	-	-	-	-
初始值	1	x	0	x	1	x	x	x	x

Bit 7	<b>WDTEN:</b> 开启/关闭 WDT。 1 = 开启 WDT。 0 = 关闭 WDT。
Bit 6	未生效的。
Bit 5	<b>LVDEN:</b> 开启/关闭 LVD。 1 = 开启 LVD。 0 = 关闭 LVD。
Bit 4	未生效的。
Bit 3	<b>LVREN:</b> 开启/关闭 LVR。 1 = 开启 LVR。 0 = 关闭 LVR。
Bit 2:0	未生效的。

## 14.2.7 INTEDG (外部中断控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTEDG	0x9B	INT2EDGE	EIS2	EIS1	EIS0	INT1G1	INT1G0	INT0G1	INT0G0
读/写属性		读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值		0	0	0	0	0	1	0	1

Bit 7 **INT2EDGE:** INT2 边沿触发选择位。

1 = 上升沿触发。

0 = 下降沿触发。

Bit 6 **EIS2:** 外部中断 2 引脚选择位。

1 = PA5 是外部中断 2 引脚。

0 = PA5 选择为GPIO。

Bit 5 **EIS1:** 外部中断 1 引脚选择位。

1 = PB1 或 PA3 是外部中断 1 引脚。

0 = PB1 或 PA3 是GPIO。

Bit 4 **EIS0:** 外部中断 0 引脚选择位。

1 = PB0, PB4 或 PA4 是外部中断 0 引脚。

0 = PB0, PB4 或 PA4 是GPIO。

Bit 3:2 **INT1G[1:0]:** INT1 边沿触发选择位。

11 = 上升沿 / 下降沿触发。

10 = 下降沿触发。

01 = 上升沿触发。

00 = 保留的。

Bit 1:0 **INT0G[1:0]:** INT0 边沿触发选择位。

11 = 上升沿 / 下降沿触发。

10 = 下降沿触发。

01 = 上升沿触发。

00 = 保留的。

## 15. 振荡器配置

### 15.1 概述

由于AT8BM84A是双时钟IC，因此可以选择高速振荡( $F_{Hosc}$ )和低速振荡( $F_{Losc}$ )作为系统振荡( $F_{osc}$ )。可用于 $F_{Hosc}$ 的振荡有内部高速RC振荡( $I_{HRC}$ )、外部高速石英振荡( $E_{HXT}$ )和外部石英振荡( $E_{XT}$ )。可用于 $F_{Losc}$ 的振荡有内部低速RC振荡( $I_{LRC}$ )和外部低速石英振荡( $E_{LXT}$ )。

根据配置字设置，当采用外部晶体( $E_{HXT}$ 、 $E_{XT}$ 或 $E_{LXT}$ )进行高速振荡或低速振荡时，PA7作为晶体输入引脚( $Xin$ )，PA6作为晶体输出引脚( $Xout$ )。当系统振荡选择 $I_{HRC}$ 或 $I_{LRC}$ 模式，不选择 $E_{HXT}$ 、 $E_{XT}$ 或 $E_{LXT}$ 模式时，如果使能配置字，在PA7上可以观察到指令时钟。

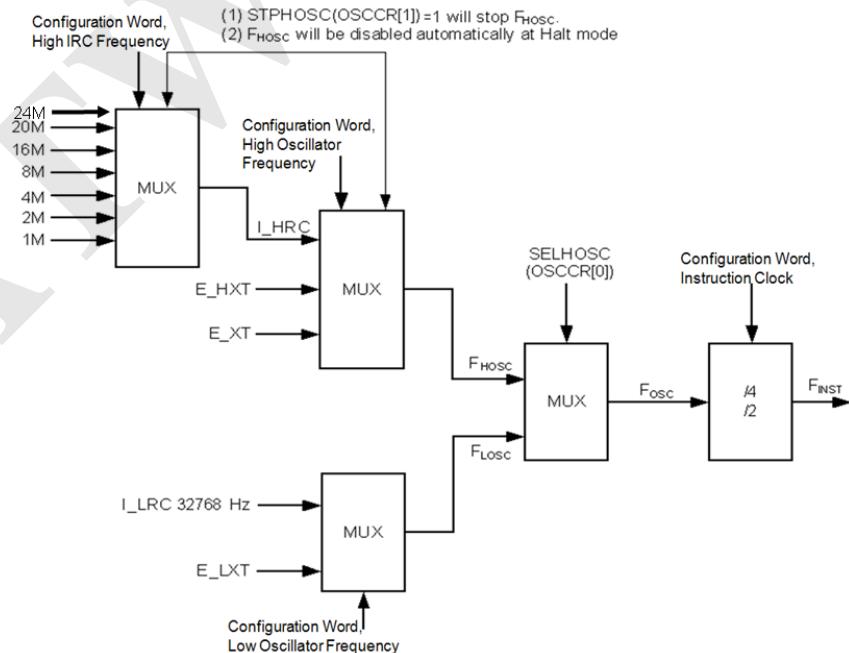


图 37 AT8BM84A的振荡器配置

有两个配置字来决定哪个振荡器将被用作 $F_{Hosc}$ 。当选择 $I_{HRC}$ 作为 $F_{Hosc}$ 时， $I_{HRC}$ 输出频率由三个配置字决定，可以是1M、2M、4M、8M、16M或20MHz。此外，外部晶振引脚PA6和PA7可以用作I/O引脚。另一方面，根据配置字的设置，PA7可以作为指令时钟的输出引脚。如果 $F_{Hosc}$ 需要使用频率为8MHz~20MHz的外部晶体，推荐使用 $E_{HXT}$ 。如果 $F_{Hosc}$ 需要外部晶体，频率范围为455KHz~6MHz，建议使用 $E_{XT}$ 。当采用 $E_{HXT}$ 或 $E_{XT}$ 时，PA6/PA7不能作为I/O引脚。它们必须用作晶体输出引脚和输入引脚。PA6为晶体输出引脚( $Xout$ )，PA7为晶体输入引脚( $Xin$ )。

有一个配置字来决定哪个振荡器将被用作 $F_{Losc}$ 。当选择 $I_{LRC}$ 时，其频率以32768Hz为中心。如果 $F_{Losc}$ 需要外部晶体，选择 $E_{LXT}$ ，只允许用32768Hz晶体。当采用 $E_{LXT}$ 时，PA6/PA7不能作为I/O引脚。它们必须用作晶体输出引脚和输入引脚。PA6为晶体输出引脚( $Xout$ )，PA7为晶体输入引脚( $Xin$ )。

$F_{Hosc}$ 和 $F_{Losc}$ 的双时钟组合如下所示。

No.	$F_{Hosc}$	$F_{Losc}$
1	$I_{HRC}$	$I_{LRC}$
2	$E_{HXT}$ or $E_{XT}$	$I_{LRC}$
3	$I_{HRC}$	$E_{LXT}$

表 15-1 双时钟组合

当E\_HXT、E\_XT或E\_LXT作为振荡之一时，将晶体或谐振器连接到Xin和Xout上提供振荡。此外，为了提供可靠的振荡，建议如下图连接一个电阻和两个电容，参考晶体或谐振器的规格，采用合适的C1或C2值。

C1和C2的推荐值如下表所示。

振荡模式	晶体频率(Hz)	C1, C2 (pF)
E_HXT	16M	5 ~ 10
	10M	5 ~ 30
	8M	5 ~ 20
E_XT	4M	5 ~ 30
	1M	5 ~ 30
	455K	10 ~ 100
E_LXT	32768	5 ~ 30

表 15-2 不同类型晶体振荡的C1 和C2 推荐值

对于2个时钟CPU周期模式下的20MHZ谐振器，C2电容必须使用18pF。

为了获得精确稳定的32.768kHz频率，选择合适的C1和C2值非常重要。您需要将C1 / C2电容与您选择的特定晶体相匹配。每个晶体数据表都列出了负载电容(CL)，C1和C2值是用以下公式选择的：

$$C1 = C2 = 2 * CL - Cbt$$

其中Cbt为AT8BM84A晶体内置电容，约为5pF。例如，晶体CL=12.5pF时，建议设置C1=C2=20pF。

在25°C商业条件下，I\_HRC的精度为±1%。

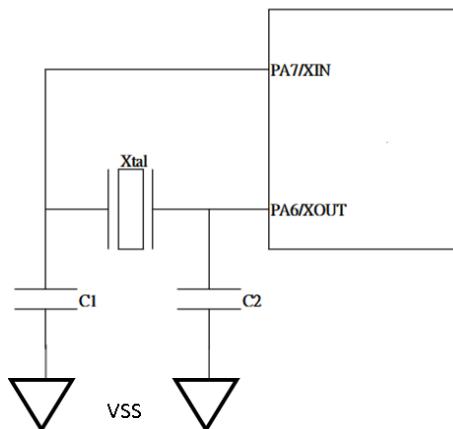


图 38 外部晶体振荡连接

根据寄存器位SELHOSC (OSCCR[0]) 的值，可以选择F<sub>Hosc</sub>或F<sub>Losc</sub>作为系统振荡F<sub>osc</sub>。当SELHOSC为1时，选择F<sub>Hosc</sub>作为F<sub>osc</sub>。当SELHOSC为0时，选择F<sub>Losc</sub>作为F<sub>osc</sub>。一旦确定了F<sub>osc</sub>，根据配置字的值，指令时钟F<sub>INST</sub>可以是F<sub>osc</sub>/2或F<sub>osc</sub>/4。

## 15.2 OSCCR (振荡器控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCR	0x9	-	-	-	XTL_BKUP	OPMD[1:0]	STPHOSC	SELHOSC	
读/写属性		-	-	-	读/写	读/写	读/写	读/写	
初始值		x	x	x	0	00	0	1	

Bit 7:5 未生效的。

Bit 4 **XTL\_BKUP:** 写 1 来停止晶体 32.768K 加速功能, 只写。

Bit 3:2 **OPMD[1:0]:** 选择操作模式。

OPMD[1:0]	操作模式
00	正常模式
01	睡眠模式
10	待机模式
11	保留

Bit 1 **STPHOSC:** 关闭/开启高频率振荡器 ( $F_{Hosc}$ )。

1 =  $F_{Hosc}$  会停止振荡并被关闭。

0 =  $F_{Hosc}$  保持振荡。

Bit 0 **SELHOSC:** 系统振荡器选择 ( $F_{osc}$ )。

1 =  $F_{osc}$  是高频振荡器 ( $F_{Hosc}$ )。

0 =  $F_{osc}$  是低频振荡器 ( $F_{Losc}$ )。

**注意: STPHOSC 不能与 SELHOSC 或 OPMD 同时更改。当 SELHOSC=1 时, STPHOSC 不能与 OPMD 同时更改。**

## 16. 工作模式

## 16.1 概述

AT8BM84A提供四种工作模式，以适应各种应用，节省功耗。这些操作模式包括正常模式、慢速模式、待机模式和睡眠模式。正常模式为高速运行模式。慢速模式为低速模式，以节省功耗。在待机模式下，AT8BM84A将停止除了Timer0/Timer1/Timer4/Timer5/WDT之外的几乎所有操作，以便定期唤醒。在睡眠模式下，AT8BM84A将睡眠，直到外部事件或WDT触发IC唤醒。四种工作模式的框图如下图所示。

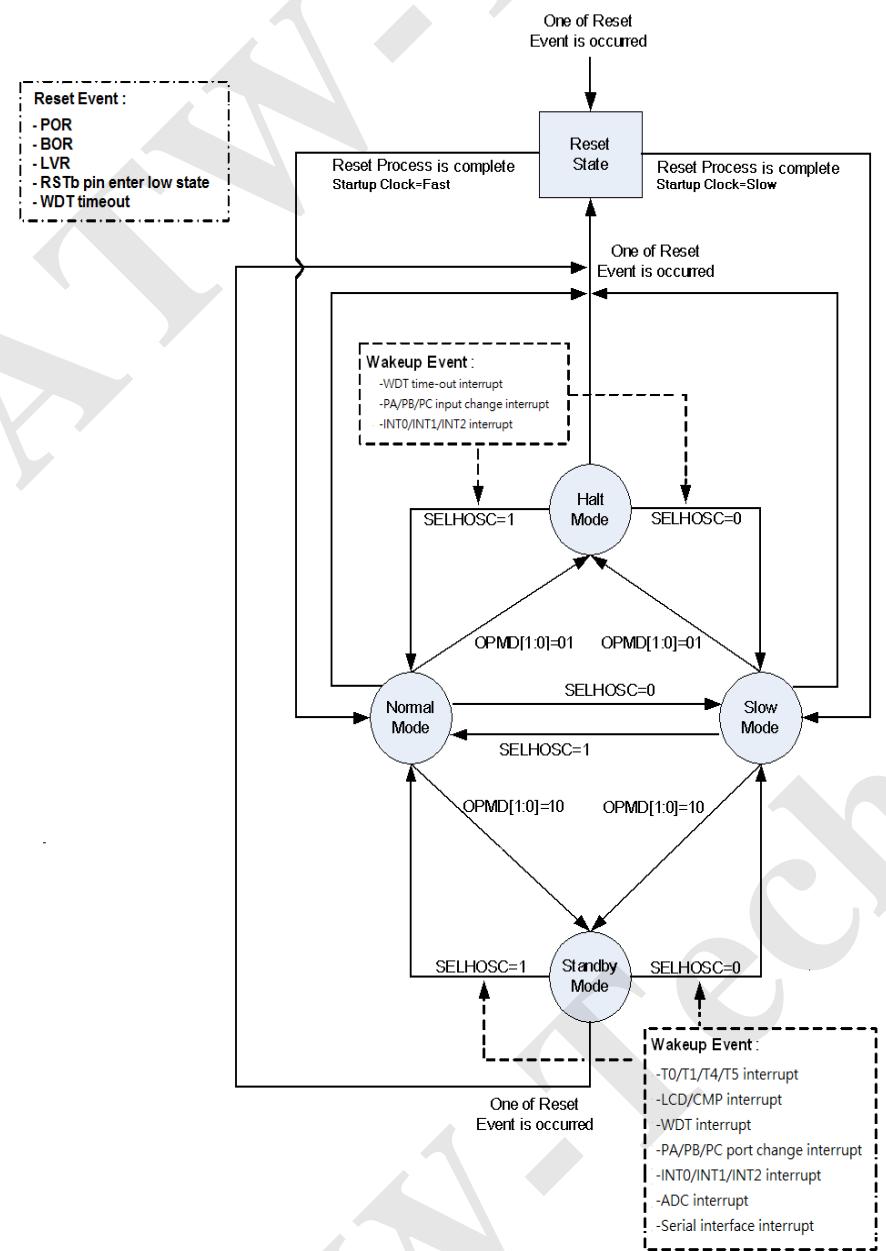


图 39 四种工作模式

## 16.2 正常模式

当复位事件发生并完成复位过程后, AT8BM84A将在正常模式或慢速模式下开始执行程序。复位过程后选择哪种模式由Startup Clock配置字决定。如果启动时钟为Fast, AT8BM84A进入正常模式, 如果启动时钟为Slow, AT8BM84A进入慢速模式。在正常工作模式下, 为了提供最高的性能, 选择F<sub>Hosc</sub>作为系统振荡, 其功耗在四种工作模式中最大。上电或复位触发释放后, 复位过程完成后, AT8BM84A进入正常模式。

- 指令执行以F<sub>Hosc</sub>为基础, 且所有外围模块可根据相应的模块使能位激活。
- F<sub>losc</sub>仍然运行。
- IC可以通过写 0 到寄存器位SELHOSC (OSCCR[0]) 切换到慢速模式。
- IC可以通过编程寄存器位OPMD[1:0] (OSCCR[3:2]) 来切换到待机模式或睡眠模式。
- 对于实时时钟应用, AT8BM84A可以在正常模式下运行, 同时低频振荡时钟连接到Timer0 时钟。这可以通过将LCKTMO 设置为 1 并将相应的配置字Timer0 源设置为 1 来实现。

## 16.3 慢速模式

AT8BM84A将通过写 0 到寄存器位SELHOSC进入慢速模式。在慢速模式下, 选择F<sub>losc</sub>作为系统振荡, 以节省功耗, 但仍保持IC运行。但是, F<sub>Hosc</sub>不会被AT8BM84A自动关闭。因此, 用户可以在慢速模式下写 0 到寄存器位STPHOSC (OSCCR[1]), 以进一步降低功耗。但是需要注意的是, 禁止同时进入慢速模式和停止F<sub>Hosc</sub>, 必须先进入慢速模式, 然后关闭F<sub>Hosc</sub>, 否则程序可能会被中止。

- 指令执行以F<sub>losc</sub>为基础, 所有外围模块可以根据相应的模块使能位激活。
- F<sub>losc</sub>可以通过向寄存器位STPHOSC写入 1 来关闭。
- IC可以通过编程寄存器位OPMD[1:0], 切换到待机模式或睡眠模式。
- IC可以通过将 1 写入SELHOSC, 切换到正常模式。

## 16.4 待机模式

AT8BM84A通过写入 10b 到寄存器位OPMD[1:0]进入待机模式。然而, 在待机模式下, AT8BM84A不会自动关闭F<sub>Hosc</sub>, 用户必须进入慢速模式并将 1 写入寄存器位STPHOSC以停止F<sub>Hosc</sub>振荡。大多数AT8BM84A外围模块被关闭, 但如果寄存器位T0EN/T1EN/T4EN/T5EN设置为 1, 定时器仍然可以激活。因此AT8BM84A可以在Timer0/Timer1/Timer4/Timer5 失效后唤醒。Timer0/Timer1/Timer4/Timer5 的有效时间由寄存器TMR0/TMR1[9:0]/TMR4[9:0]/TMR5[9:0]、FINST和其他配置决定。

- 指令执行停止, 一些外围模块根据相应的模块使能位激活。
- F<sub>losc</sub> 可以通过向寄存器位STPHOSC写入 1 来关闭。
- F<sub>losc</sub> 仍然运行。
- 如果发生以下中断, IC可以从待机模式唤醒。(a) Timer0/Timer1/Timer4/Timer5(上溢/下溢)中断, (b) WDT 超时中断, (c) PA/PB/PC输入改变中断, (d) INT外部中断, (e) LVD中断, (f) 比较器输出状态改变中断 (g) ADC转换结束中断。
- 从待机模式唤醒后, 如果SELHOSC=1, IC将返回到正常模式, 如果SELHOSC=0, IC将返回到慢速模式。
- 不建议同时切换振荡器模式(正常转慢/慢转正常)并进入待机状态。

用户应在待机模式指令后立即添加NOP指令。

```
; Into Standby Mode Method:  
MOVIA 0x08 | C_FHOSC_Sel  
MOVAR OSCCR          ; Set OSCCR Register to Enters Standby Mode  
NOP                 ; NOP Instruction After the Standby Mode
```

## 16.5 睡眠模式

AT8BM84A通过执行SLEEP指令或将 01b 写入寄存器位OPMD[1:0]进入睡眠模式。进入睡眠模式后，寄存器位 /PD (STATUS[3]) 清除为 0，寄存器位 / TO (STATUS[4]) 设置为 1，WDT清除，但继续运行。

在睡眠模式下，所有外围模块都被关闭，指令停止执行，AT8BM84A只能通过一些特定的事件唤醒。因此，睡眠模式是AT8BM84A提供的最省电的模式。

- 指令执行停止，所有外围模块被关闭。
- FHOSC和FLOSC都会自动关闭。
- 如果发生以下中断，IC可以从睡眠模式唤醒：(a) WDT超时中断，(b) PA/PB输入改变中断或 (c) INT或外部中断。
- 从睡眠模式唤醒后，如果SELHOSC=1，IC将返回到正常模式，如果SELHOSC=0，IC将返回到慢速模式。

**注意：用户可以在同一指令中更改STPHOSC并进入睡眠模式。**

- 不建议同时切换振荡器模式（正常转慢或慢转正常）并进入睡眠模式。

## 16.6 唤醒稳定时间

睡眠模式的唤醒稳定时间由配置字决定：高速振荡频率或低速振荡频率。若选择E\_HXT, E\_XT 或 E\_LXT其中一种作为系统振荡时钟来源，其睡眠模式的唤醒等待时间为  $512*Fosc$ ，若没有选择外部振荡器作为系统振荡时钟来源，其睡眠模式的唤醒等待时间为  $16*Fosc$ ，由于待机模式下FHOSC或FLOSC仍在运行，因此无需为待机模式唤醒稳定时间。

在AT8BM84A进入待机模式或睡眠模式之前，用户可以执行指令BSR INTCON.GIE。在此条件下，AT8BM84A将跳转到地址 0x004，以便在唤醒后执行中断服务程序。如果在进入待机模式或睡眠模式之前执行BCR INTCON.GIE指令，在唤醒后将会执行下一条指令。**用户应在睡眠中唤醒的SLEEP指令后写一个NOP。**

; Into Sleep Mode Method 1

```
MOVIA    C_Halt_Mode | C_FHOSC_Sel
MOVAR    OSCCR          ; Set OSCCR Register to Enters Halt Mode
NOP      ; NOP Instruction After the Sleep Mode
```

; Into Sleep Mode Method 2

```
SLEEP      ; Execute instruction to Enters Halt Mode
NOP      ; NOP Instruction After the Sleep Mode
```

## 16.7 工作模式概述

四种工作模式概述如下：

模式	正常模式	慢速模式	待机模式	睡眠模式
$F_{HOSC}$	使能	STPHOSC	STPHOSC	关闭
$F_{LOSC}$	使能	使能	使能	关闭
指令执行	执行	执行	停止	停止
Timer0/1/4/5	TxEN	TxEN	TxEN	关闭
WDT	选择和 WDTEN	选择和 WDTEN	选择和 WDTEN	选择和 WDTEN
其它硬件	硬件使能位	硬件使能位	硬件使能位	全部关闭
Wake-up Source	-	-	- 定时器 0/1/4/5 上溢/下溢 - WDT超时 - PA/PB/PC输入状态改变 - 外部中断 INT0/1/2 - LVD中断 - 比较器中断 - ADC模数转换结束	- WDT超时 - PA/PB/PC 输入状态改变 - 外部中断INT0/1/2

表 16-1 工作模式概述

## 17. 复位

### 17.1 概述

AT8BM84A可通过以下 4 种方式复位：

- 当VDD检测到上升沿时，发生上电复位（POR）。
- 当VDD电压低于预设的LVR电压时，发生LVR复位。
- RSTb引脚为低电平状态。
- WDT超时复位。

此外，所有寄存器如果初始值未知时，寄存器初始值将会被初始化或保持不变。状态位/TO和/PD可以根据复位事件来初始化。/TO和/PD的值及其相关的事件概述如下。

事件	/TO	/PD
POR, LVR	1	1
非睡眠模式时发生RSTb复位	不变	不变
睡眠模式时发生RSTb复位	1	1
非睡眠模式时发生WDT复位	0	1
睡眠模式时发生WDT复位	0	0
执行SLEEP 指令	1	0
执行CLRWDT 指令	1	1

表 17-1 /TO和/PD值和相关事件概述

复位事件发生后, AT8BM84A将会开始复位进程。无论采用什么样的振荡器, 它将等待一定的周期使振荡稳定。这个周期被称为上电复位时间, 它由三位配置字决定, 这个时间可能是 600us, 4.5ms, 18ms, 72ms 或 288 ms。振荡器稳定后, 上电复位后, 在开始执行程序前, AT8BM84A将等待更多的振荡启动时间(OST)。若前一个上电复位时间为 600us 时, 则  $F_{osc}$  的时钟周期  $OST=1$ , 若前一个上电复位时间为 4.5ms, 18ms, 72ms 或 288ms 时, 则  $F_{osc}$  的时钟周期  $OST=16$ 。

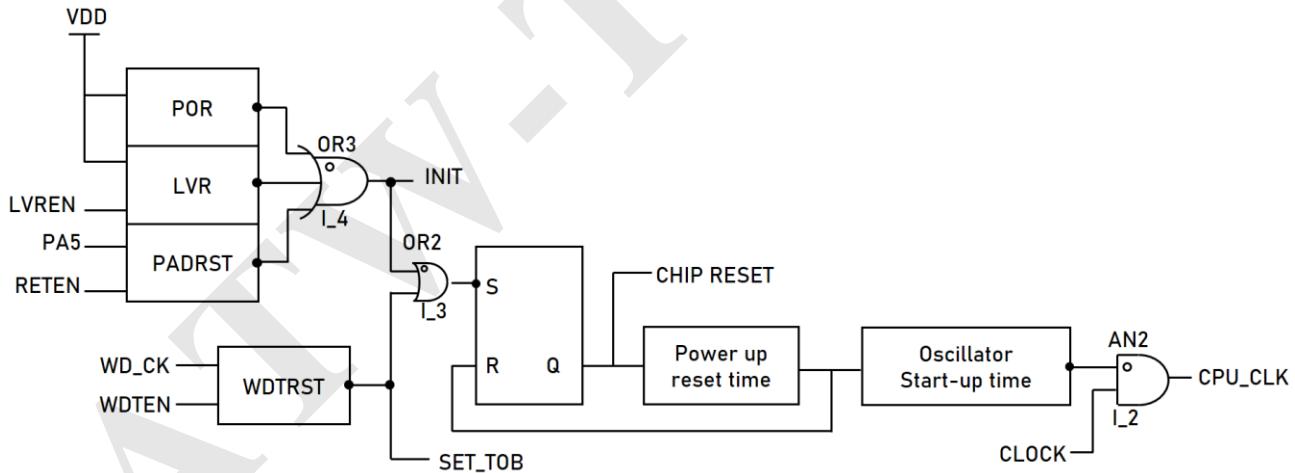


图 40 芯片复位电路框图

PA5 可作为由配置字决定的外部复位输入。当对PA5 输入低有效信号时, 会使AT8BM84A进入复位过程。对于 VDD 上电缓慢的情况, 建议使用 RSTb 复位, 如下图所示。

- 建议 R 阻值不大于 40KΩ。
- R1 值 = 100Ω ~ 1KΩ 时, 将阻止过大电流, ESD 或电气过载信号进入复位引脚。
- 二极管 D 使电容 C 能在 VDD 断电时快速放电。

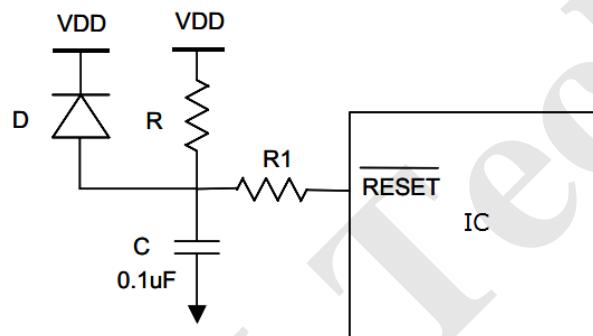


图 41 外部上电复位硬件连接图

## 18. SPI 模式

### 18.1 概述

SPI 包含以下特性:

- 全双工操作。
- 4 个可编程的主模式频率。
- 具有可编程极性和相位的串行时钟
- 传输结束中断标志
- 写入冲突错误标志。
- 总线冲突错误标志。

AT8BM84A将通过写入寄存器SIMCR[7:6]=10 进入SPI模式。如果寄存器SIMCR[4]=1, IC包含四线SPI接口, 如果寄存器SIMCR[4]=0, IC包含三线SPI接口。

当设备可作主机也可作从机时, 它的通信是以被动/主动形式作双向输出。如果寄存器SIMCR[5]=1, 设备为主机, 如果SIMCR[5]=0, 则为从机。

SPI 接口是全双工同步串行数据连接。它是一个四线接口, 引脚名称分别为SCK, MISO, MOSI 和SSBEN。当选择三线接口时, SSBEN省略, 无论主机还是从机都是开启的。

只有主机 SPI 才能发起传输。当主机 SPI 写入数据到 SPI 数据寄存器 (SIMDR) 时, 软件开始传输。寄存器SIMDR不会从传输的 SPI 中缓存数据。在连续时钟源的控制下, 数据写入SIMDR会直接进入转移寄存器并开始立即传输。在循环 8 个连续时钟后, SPI 标志 (SPIF) 设置起来时, 传输结束。在SPI 标志 (SPIF) 设置起来的同时, 转移进主机 SPI 的数据从接收设备传输到SIMDR。SIMDR通过 SPI 缓存接收到的数据。在主机SPI 发送下一个字节之前, 软件必须通过读SPCR来清零SPIF位然后读或写SIMDR。

从机SPI, 在连续时钟的控制下, 数据从主机 SPI 进入转移寄存器。当从机 SPI 的一个字节进入转移寄存器后, 它将传输给SIMDR。为了防止溢出, 在另一个字节进入转移寄存器并准备传输给SIMDR之前, 从机软件必须在SIMDR中读取字节。

先读SPCR, 然后再读或写SIMDR会清零SPIF和WCOL。

先读SPCR, 然后写SPCR会清零MODF。

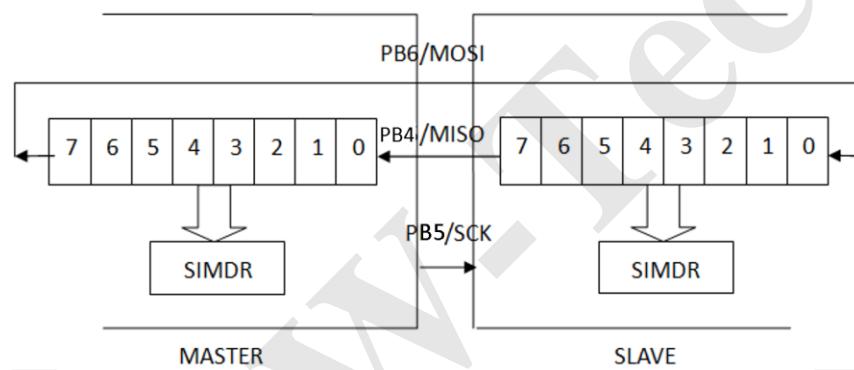


图 42 单个主机/从机

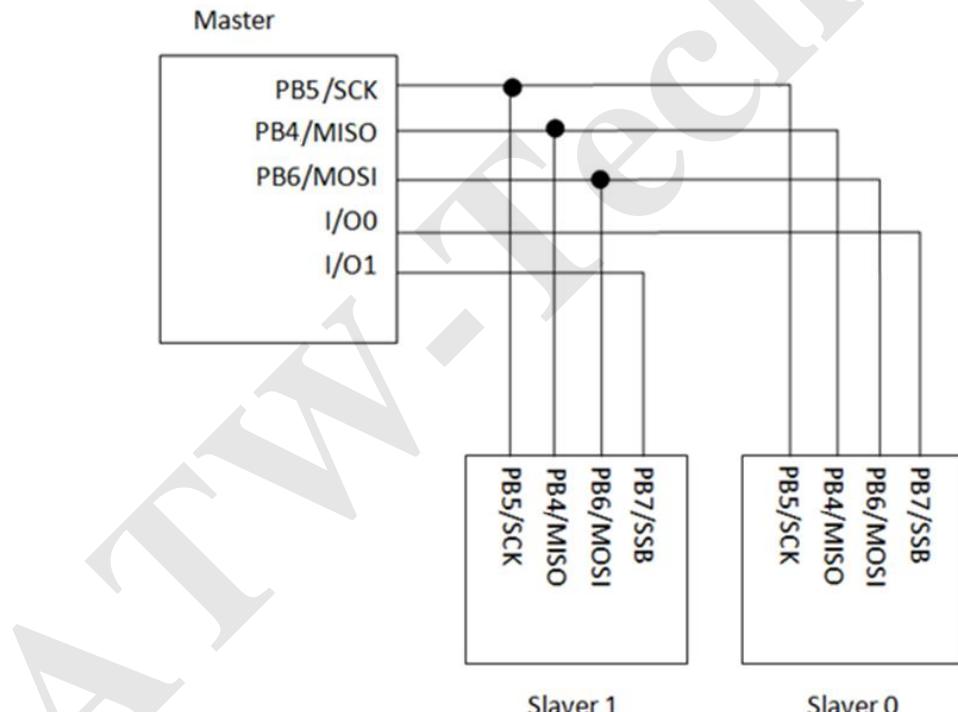


图 43 单个主机和多个从机

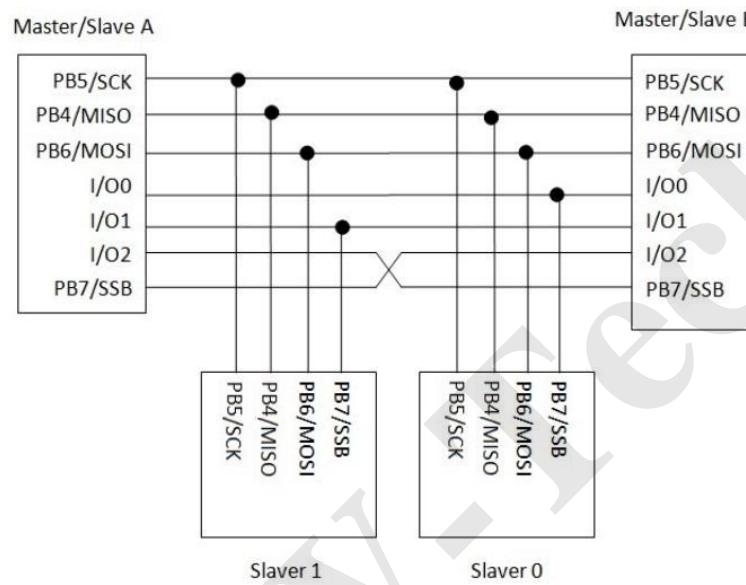


图 44 多个主机和多个从机

## 18.2 串行时钟极性和相位

根据CPOL (SPCR[3]) 位和CKEG (SPCR[2]) 位的配置，有四种类型可以适应不同的外设串行通信需求。

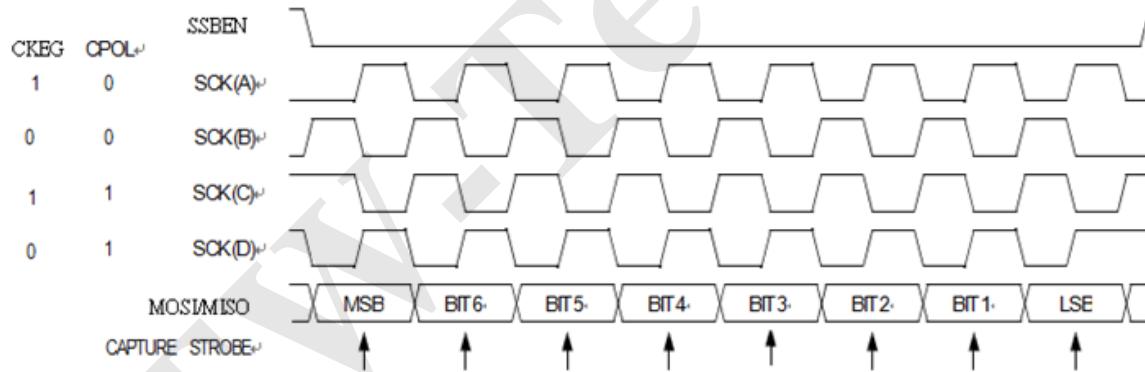


图 45 显示了CPOL和CKEG位如何影响时钟/数据时序

有四种不同的 SPI 主时钟速率，按SPCR[1:0] (SPR[1:0]) 配置，SPR[1:0]=00/01/10/11 将选择系统时钟/2、系统时钟/4、系统时钟/16、系统时钟/32。

用户必须先设置SPCR[3:0]，然后通过设置SIMCR[7:6]=10 开启 SPI 模块，否则会发生硬件错误。

下图是SPI 流程图。

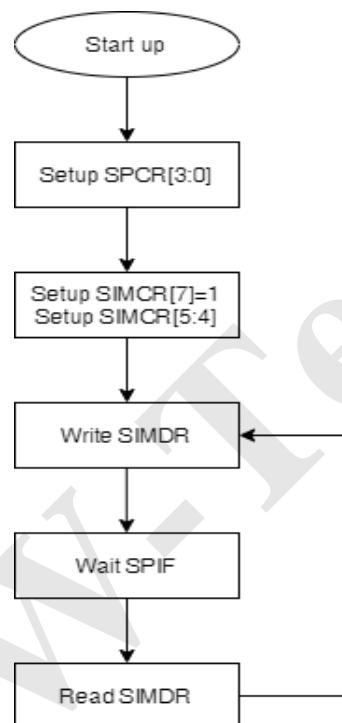


图 46 SPI 接口流程图

### 18.3 SPI错误条件

这些条件造成系统错误：

- (1) 在主机模式下, 如果PB5/SSBEN 为逻辑 0, MODF(模式错误)将会发生。
- (2) 在传输过程中写入SIMDR会导致写入冲突错误, 并在SPCR中设置WCOL位。这个错误不会影响之前写入的字节的传输, 但是导致该错误的字节会丢失。
- (3) 在下一个进入的字节设置SPIF位之前, 读取SIMDR失败。

### 18.4 SPI控制寄存器

#### 18.4.1 SIMDR (串行接口模式数据寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SIMDR	0x112	SIMD7	SIMD6	SIMD5	SIMD4	SIMD3	SIMD2	SIMD1	SIMD0
读/写属性	读/写								
初始值	x	x	x	x	x	x	x	x	x

Bit 7:0 **SIMDR[7:0]:** 8bit 串行接口模式数据寄存器。

如果I<sup>2</sup>C模式开启, 它们是 I<sup>2</sup>C模式数据寄存器。

如果SPI 模式开启, 它们是 SPI 模式数据寄存器。

#### 18.4.2 SIMCR (串行接口模式控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SIMCR	0x113	SIMC1	SIMC0	MSTA	SSB_PAD	RX_PA DEN	TX_PA DEN	RCLK_PADEN	UREN
读/写属性	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值	0	0	0	0	0	0	0	0	0

Bit 7 **SIMC1:** SPI 模式开启位。

1 = SPI 模式开启。

0 = SPI 模式关闭。

Bit 6 **SIMC0:** I<sup>2</sup>C 模式开启位。

1 = I<sup>2</sup>C 模式开启。

0 = I<sup>2</sup>C 模式关闭。

Bit 5 **MSTA:** 主机/从机模式选择 (包括I<sup>2</sup>C模式和SPI模式)。

1 = 选择主机模式。

0 = 选择从机模式。

Bit 4 **SSB\_PAD:** 决定PB7 的引脚是否共享为SSBEN, 此功能在SPI模式下使用。

1 = PB7 用作SSBEN。

0 = PB7 不用作SSBEN (此时, 不管是主机模式还是从机模式, 始终是开启状态)。

Bit 3 **RX\_PADEN:** UART端口RXPAD。

1 = PB7 或PB0 用作UART RX信号输入。

0 = PB7 或PB0 用作一般IO口。

Bit 2 **TX\_PADEN:** UART端口TXPAD。

1 = PB6 或PB1 用作UART TX信号输出。

0 = PB6 或PB1 用作一般IO口。

- Bit 1      **RCLK\_PDEN:** UART端口 RCLK\_PDEN。  
     1 = PB4 用作UART RX CKT时钟输入，频率=波特率\*/16。  
     0 = PB4 用作一般IO口。
- Bit 0      **UREN:** UART端口开启。  
     1 = UART端口开启。  
     0 = UART端口关闭。

#### 18.4.3 SPCR (SPI 控制和状态寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCR	0x114	SPIF	WCOL	-	MODF	CPOL	CKEG	SPR[1:0]	
读/写属性		读/写	读/写	-	读/写	读/写	读/写	读/写	读/写
初始值		0	0	x	0	0	0	0	0

- Bit 7      **SPIF:** SPI 标志。  
     1 = 传输完成。  
     0 = 传输失败。  
     先读 SPCR，然后读或写SIMDR，会清除SPIF 和 WCOL。
- Bit 6      **WCOL:** 写入冲突标志。  
     1 = 对SIMDR写入无效。  
     0 = 对SIMDR写入有效。  
     先读 SPCR，然后读或写SIMDR，会清除SPIF 和 WCOL。
- Bit 5      未生效的。
- Bit 4      **MODF:** 模式故障标志。  
     1 = 当MSTR位设置时，SSBEN被拉低。  
     0 = 当MSTR位设置时，SSBEN不会被拉低。  
     先读 SPCR，然后写SPCR，会清除MODF。
- Bit 3      **CPOL:** SPI 时钟极性选择位。  
     1 = 传输之间逻辑 1 的SCK引脚，空闲时为高电平。  
     0 = 传输之间逻辑 0 的SCK引脚，空闲时为低电平。
- Bit 2      **CKEG:** SPI SCK时钟边沿选择。  
     当 CPOL = 1  
     1 = SCK空闲时为高电平，下降沿捕捉数据。  
     0 = SCK空闲时为高电平，上升沿捕捉数据。  
     当 CPOL = 0  
     1 = SCK空闲时为低电平，下降沿捕捉数据。  
     0 = SCK空闲时为低电平，上升沿捕捉数据。
- Bit 1:0     **SPR[1:0]:** SPI 时钟速率选择位。

SPR[1:0]	SPI 时钟速率
00	System Clock /2
01	System Clock /4
10	System Clock /16
11	System Clock /32

## 19. I<sup>2</sup>C 模式

### 19.1 概述

AT8BM84A通过写SIMCR[7:6]=01进入I<sup>2</sup>C模式，I<sup>2</sup>C模式的标准是一个双线总线。这种双线总线最大限度地减少了设备之间的互连，并消除了对地址解码器的需要，具有PCB路由少和经济的硬件结构的优势。

I<sup>2</sup>C模式适用于需要在多个设备之间进行短距离通信的应用。

I<sup>2</sup>C模式具有以下特点：

- 多主机操作
- 32个软件可编程串行时钟频率
- 软件可选择的应答位
- 逐字节传输中断
- 仲裁失败中断
- 呼叫地址识别中断
- 产生/检测开始，停止和应答信号
- 重复START信号产生
- 总线忙线检测

下图为I<sup>2</sup>C模式的功能结构图：

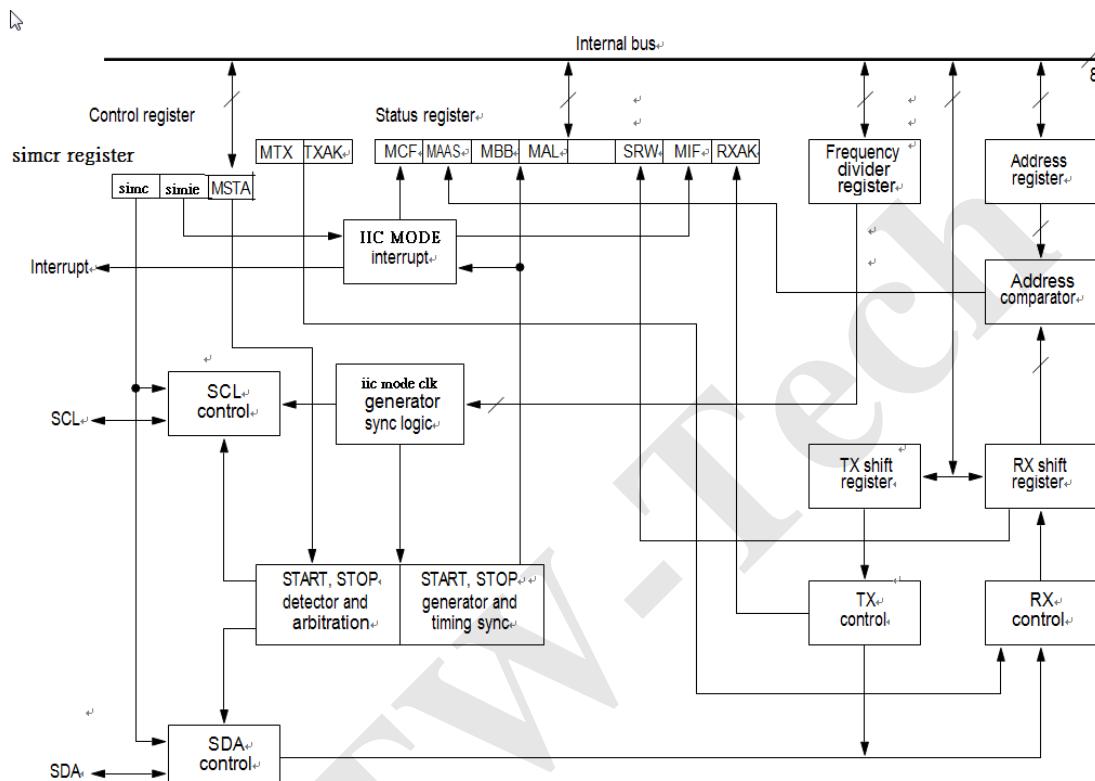


图 47 I<sup>2</sup>C接口结构框图

下图为I<sup>2</sup>C操作示例。一台主机设备和一台从机设备。SDA和SCK需要拉高电阻。

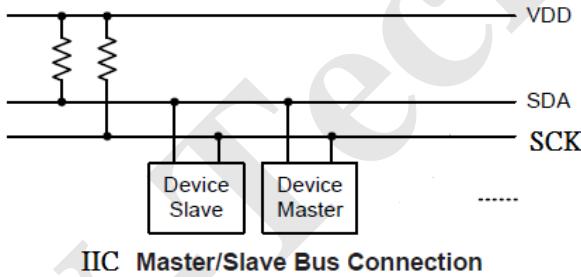


图 48 I<sup>2</sup>C总线连接

## 19.2 I<sup>2</sup>C 模式协议

一个标准的通信由四个部分组成，它们是：

- (1) 开始
- (2) 从机地址传输
- (3) 数据传输
- (4) 停止。

下图显示了SDA/SCK：

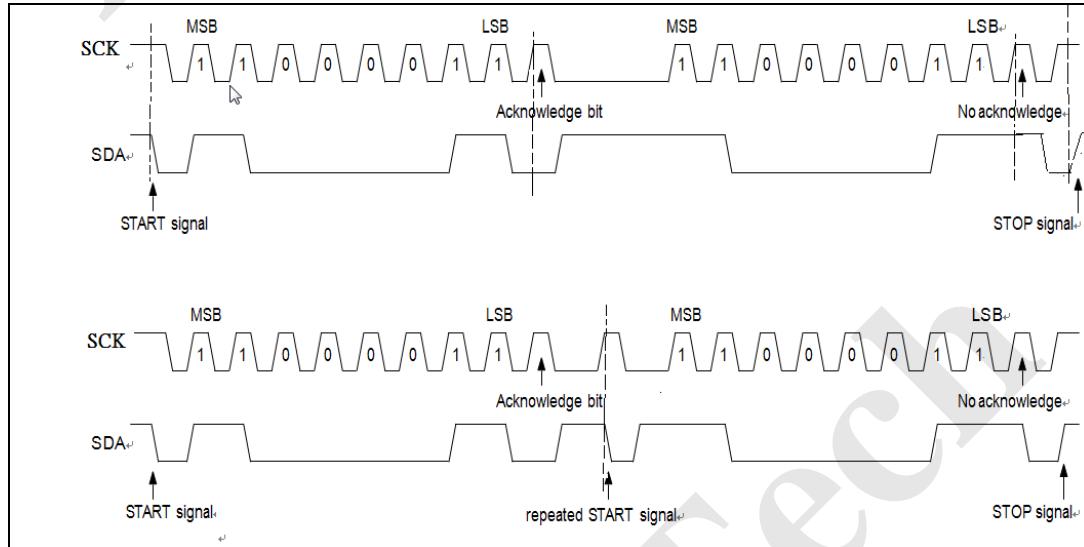


图 49 I<sup>2</sup>C总线传输信号图

## 19.3 I<sup>2</sup>C 模式操作

AT8BM84A 将通过写入SIMCR[7:6]=01 进入I<sup>2</sup>C模式。如果SIMCR[5](MSTA)=1，设备进入主机模式；如果SIMCR[5](MSTA)=0，设备进入从机模式。

复位后，SIMCR[5]被清除。当这个位从 0 变为 1 时，总线上产生一个START信号，并选择主机模式。当这个位从 1 变为 0 时，产生一个STOP信号，操作模式从主机切换为从机。

MCR[4]=1，I<sup>2</sup>C 模式设置为发送模式，否则MCR[4]=0，I<sup>2</sup>C模式设置为接收模式。

每次，我们想要TX/RX数据，首先我们去清除MSR[1](MIF)。然后如果MAL=1 则清除MSTA为 0。

在主机/TX模式下，发送 8 位数据后，将读取MSR[0]来检查是否接收第 9 个时钟发送确认。如果接收到确认位，则继续写入SIMDR，否则向终端通信产生STOP信号。

在主机/RX模式下，设置TXAK=0，接收数据，在第9个时钟位发送确认信号。如果从SIMDR读取了最后2个Byte的数据，则设置TXAK=1。如果从SIMDR读取了最后1个Byte，则生成STOP信号。

在从机模式下，检查MSR[6] (MASS)是否为第1个字节。如果MSR[6](MASS)=1，则继续检查SRW是高还是低。如果SRW高，则将MCR[4](MTX)=1设置为进入TX模式并向SIMDR写入数据；如果SRW低，则将MCR[4](MTX)=0设置为RX模式并虚拟读取SIMDR。

在从机/TX模式下，发送8位数据后，读取MSR[0](RXAK)，检查是否收到第9个时钟发送确认。如果接收确认位，则继续写SIMDR，否则设置RX模式和虚拟读取SIMDR。

下图显示建议流程：

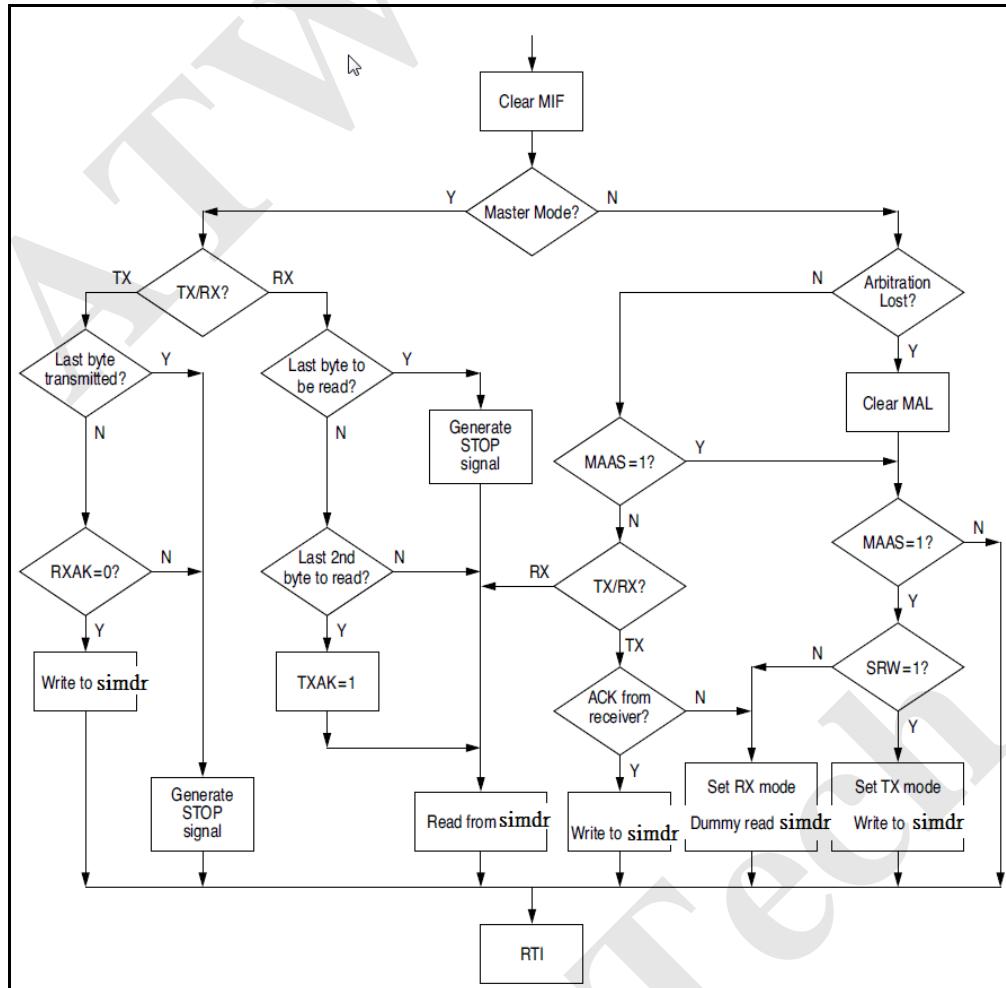


图 50 I<sup>2</sup>C模式设置流程

#### 19.4 仲裁机制

该接口电路是一个真正的多主机系统，允许多个主机连接。如果两个或更多的主机试图同时控制总线，一个时钟同步过程决定总线时钟。时钟低周期等于主机中最长的时钟低周期，而时钟的高周期是所有主机中最短的。数据仲裁程序决定数据的优先级。如果主机发送逻辑“1”，而其他主机发送逻辑“0”，则主机将失去仲裁，失去的主机将立即切换到从机接收模式，停止数据和时钟输出。从主机模式到从机模式的转换不会生成STOP条件。与此同时，硬件将设置一个软件位来表示仲裁失败。

19.5 I<sup>2</sup>C 控制寄存器19.5.1 MADR (I<sup>2</sup>C 模式地址寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MADR	0x115	MAD7	MAD6	MAD5	MAD4	MAD3	MAD2	MAD1	-
读/写属性		读/写	-						
初始值		0	0	0	0	0	0	0	x

Bit 7:1 **MAD[7:1]**: MAD[7:1]是I<sup>2</sup>C模式的从机地址位。

Bit 0 未生效的。

19.5.2 MFDR (I<sup>2</sup>C 模式频率寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MFDR	0x116	-	-	-	FD4	FD3	FD2	FD1	FD0
读/写属性		-	-	-	读/写	读/写	读/写	读/写	读/写
初始值		x	x	x	0	0	0	0	0

Bit 7:5 未生效的。

Bit 4:0 **FD[4:0]**: FD[4:0]被用于时钟频率选择，串行位时钟频率等于CPU时钟除以下表所示的分频器。

例: CPU CLOCK=1MHz, FD[4:0]=2, I<sup>2</sup>C MODE CLK FREQ=1MHz/28=35.7KHz

FD[4:0]	DIVIDER	FD[4:0]	DIVIDER
00000	22	10000	352
00001	24	10001	384
00010	28	10010	448
00011	34	10011	544
00100	44	10100	704
00101	48	10101	768
00110	56	10110	896
00111	68	10111	1088
01000	88	11000	1408
01001	96	11001	1536
01010	112	11010	1792
01011	136	11011	2176
01100	176	11100	2816
01101	192	11101	3072
01110	224	11110	3584
01111	272	11111	4352

19.5.3 MCR (I<sup>2</sup>C 模式控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MCR	0x117	-	-	-	MTX	TXAK	-	-	-
读/写属性		-	-	-	读/写	读/写	-	-	-
初始值		x	x	x	0	0	x	x	x

Bit 7:5 未生效的。

Bit 4 **MTX:** I<sup>2</sup>C 模式的发送/接收模式选择。

1 = 发送模式。

0 = 接收模式。

Bit 3 **TXAK:** I<sup>2</sup>C 模式的应答信号。

1 = 不发送应答信号。

0 = 在第 9 个时钟发送应答信号。

Bit 2:0 未生效的。

19.5.4 MSR (I<sup>2</sup>C 模式状态寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MSR	0x118	MCF	MAAS	MBB	MAL	-	SRW	MIF	RXAK
读/写属性		读	读	读	读/写	-	读	读/写	读
初始值		1	0	0	0	x	0	0	1

Bit 7 **MCF:** 数据发送成功。

1 = 1byte 发送成功。

0 = 1byte 正在发送。

Bit 6 **MAAS:** 从机地址。

1 = 当前从机地址。

0 = 不是当前从机地址。

设置 MAAS 时, MIF (I<sup>2</sup>C 模式中断) 位也会被设置。

如果 MAAS=1, 则 CPU 需要检查 SRW 位, 并且相应地设置其 MTX 位。

Bit 5 **MBB:** 总线忙。

1 = 总线忙。

0 = 总线闲。

Bit 4 **MAL:** 仲裁失败。

1 = 主机模式总线仲裁失败。

0 = 主机模式总线仲裁成功。

在主机发送模式下, 当 I<sup>2</sup>C 模式主机仲裁失败时, MAL 被置 1。

MAL 被置 1 时, MIF 也会置 1, 此位必须由软件清除。

Bit 3 未生效的。

Bit 2 **SRW:** 从机读/写选择。

1 = 呼叫主机，向从机读数据。

0 = 呼叫主机，向从机写数据。

MAAS 置 1 后，读/写从主机发送的呼叫地址的命令位被锁存到此 SRW 位中。通过检查该位，设备可以通过配置 I<sup>2</sup>C 模式控制寄存器的 MTX 位来选择从机发送/接收模式。

Bit 1 **MIF:** I<sup>2</sup>C 中断标志。

1 = I<sup>2</sup>C 中断发生。

0 = I<sup>2</sup>C 中断没有发生。

如果 SIMIE 等于 1，MIF 被置 1，产生如下中断事件：

- 1) 完成 1byte 数据传输。它被设置在第 9 个时钟的下降沿- MCF 设置。
- 2) 从机模式下，主机呼叫地址与从机地址匹配。- MAAS 设置。
- 3) 总线仲裁失败。- MAL 设置。

该位必须由中断程序中的软件清除。

Bit 0 **RXAK:** 接收应答信号。

1 = 没有检测到应答信号。

0 = 接收第 8 位数据后检测到应答信号。

**注意：MIF 和 MAL 位是软件可清除的，而其他位是只读的。**

## 20. 通用异步收发器 (UART)

### 20.1 概述

可编程异步通信接口(UART)的功能模块提供数据格式化和控制串口通信通道。

该功能具有选择、读/写、中断和总线接口逻辑功能，允许在 8 位双向并行数据总线系统上传输数据。通过适当的格式化和错误检查，该功能模块可以发送和接收串行数据，支持异步操作。

- 全双缓冲。
- 异步操作。
- 独立控制发送，工作状态以及接收中断。
- 可编程数据字节长（5~8 位），奇偶校验和停止位。
- 奇偶校验，溢出和帧错误检查。
- 可编程波特率发生器允许任何参考时钟除以 1 到 (216~1)，并产生一个内部的  $16 \times$  时钟。
- 错误启动位检测。
- 自动断路产生及检测。
- 内部诊断能力。

发射部分由发射保持寄存器(THR)和发射移位寄存器(TSR)组成。当THR或TSR为空时，写入THR将把数据总线(DIN 7-0)的内容传输到传输保持寄存器。这个写操作应该在设置发送保持寄存器(THR)为空(THRE)时完成。

这个寄存器包含封包的接收数据。在开始位的下降边沿上，接收部分开始其操作。如果RXDATA在起始位的中间样本仍然很低，那么起始位是有效的，从而防止接收端数据包一个错误的数据字符。

行控制寄存器用于指定数据通信格式。可以通过写入行状态寄存器(LSR)中相应的位来改变中断特性、奇偶校验、停止位和字节长度。

### 20.2 UART 控制寄存器

#### 20.2.1 DLL (波特率除法锁存 LSB 寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DLL	0x119	DLL7	DLL6	DLL5	DLL4	DLL3	DLL2	DLL1	DLL0
读/写属性		读/写							
初始值		0	0	0	0	0	0	0	0

Bit 7:0      **DLL[7:0]**: 波特率除法器低 8 位。

#### 20.2.2 DLH (波特率除法锁存 MSB 寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DLH	0x11A	DLH7	DLH6	DLH5	DLH4	DLH3	DLH2	DLH1	DLH0
读/写属性		读/写							
初始值		0	0	0	0	0	0	0	0

Bit 7:0      **DLH[7:0]**: 波特率除法器高 8 位。

**Baud rate=I\_HRC / [16x(N)]**, N={**DLH[7:0], DLL[7:0]**}

## 20.2.3 LCR (行控制寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCR	0x11b	LOOP	SBRK	PSTUCK	PEVEN	PREN	STPS	WL1	WL0
读/写属性		读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值		0	0	0	0	0	0	0	0

Bit 7

**LOOP:** 循环回测试使能位。

1 = 发射器移位输出数据被循环回移入接收器移位寄存器。

0 = 关闭循环回测试。

Bit 6

**SBRK:** 设定休息时间。

1 = 串行输出被强制至间隔（逻辑 0）状态，无论其它传输器是否在传输数据，都将保持这个状态。

Bit 5

**PSTUCK:** 固定奇偶校验。

1 = 奇偶校验位被发送，然后被接收器检测为逻辑 0。

0 = 奇偶校验位为逻辑 1。

Bit 4

**PEVEN:** 奇偶位选择。

1 = 发送或检查偶数位。

0 = 发送或检查奇数位。

Bit 3

**PREN:** 校验使能位。

1 = 在串行数据的最后一个数据位和停止位之间生成奇偶校验位（传输数据）或检查（接收数据）。

0 = 无奇偶校验。

Bit 2

**STPS:** 结束位数量选择。

STPS	字长	结束位数
0	X	1
1	5	1.5
1	6,7,8	2

Bit 1:0

**WL[1:0]:** 字长选择位。

WL[1:0]	字长位数
00	每字符 5 位
01	每字符 6 位
10	每字符 7 位
11	每字符 8 位

## 20.2.4 LSR (行状态寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LSR	0x11c	-	TSRE	THRE	BKINT	FERR	PERR	OVERR	READY
读/写属性	-	读	读	读/写	读/写	读/写	读/写	读/写	读/写
初始值	x	1	1	0	0	0	0	0	0

Bit 7 未生效的。

Bit 6 **TSRE**: 发送移位寄存器 (TSR) 为空。

THR 寄存器和 TSR 移位寄存器均为空时, 该位设置为逻辑 1, 只要包含数据字符, 它就会重置为逻辑 0。写 TBR 会将 TSRE 清除。

Bit 5 **THRE**: 发送保持寄存器 (THR) 空标志。

这个位表示控制器已准备好接受一个新字符进行传输。当一个字符从发送保持寄存器 (THR) 转移到移位寄存器时, 它被设置为逻辑 1。向 THR 写入数据将清除 THRE。

Bit 4 **BKINT**: Break 中断标志。

每当接收器数据输入保持在间隔状态 (逻辑 0) 超过全字传输时间时, 该位被设置为逻辑 1。读取 LSR 会将 BKINT 清除。

Bit 3 **FERR**: 帧错误标志。

此位表示接收的字符没有有效的停止位。当检测到最后一个数据位或奇偶校验位后的停止位为零位时, 它被设置为逻辑 1。读取 LSR 会将 FERR 清除。

Bit 2 **PERR**: 奇偶校验错误标志。

1 = 检测奇偶校验错误。

0 = 无奇偶校验错误。

读取 LSR 会将 PERR 清除。

Bit 1 **OVERR**: 超时运行错误标志。

1 = 超时运行发生。

0 = 超时运行未发生。

读取 LSR 会将 OVERR 清除。

Bit 0 **READY**: 数据就绪标志位。

每当接收到一个完整的传入字符并将其传输到接收缓存寄存器时, 它被设置为逻辑 1。

读取 RBR 数据将清除此标志。

## 20.2.5 THR/RBR (发送保持寄存器/接收缓冲寄存器)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
THR/RBR	0x11d	URD7	URD6	URD5	URD4	URD3	URD2	URD1	URD0
读/写属性	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值	x	x	x	x	x	x	x	x	x

Bit 7:0 **URD[7:0]**: 如果给这个寄存器赋值, 则为 UART 发送数据寄存器。

如果读取这个寄存器的值, 数据为 UART 接收数据寄存器。

## 21. LCD

### 21.1 概述

AT8BM84A有一个内置的LCD驱动模块。当LCDEN设置为1时，相应的SFR LDCCON1中的PxCOM位设置为1时，LCD驱动模块用于驱动  $\frac{1}{2}$  VDD偏置到8个COM端口之一。8个COM端口和对应的PxCOM如下表所示：

COM PORT	SFR PxCOM
PB4	P0COM
PB5	P1COM
PB6	P2COM
PB7	P3COM
PC2	P4COM
PC3	P5COM
PC4	P6COM
PC5	P7COM

表 21-1 LCD COM 端口选择

AT8BM84A的SEG端口可以是任何可以输出VDD和VSS电压电平的IO引脚。

一个完整的LCD波形周期包含两个帧，分别为，P帧和N帧，如下图所示为3个COM的LCD波形。

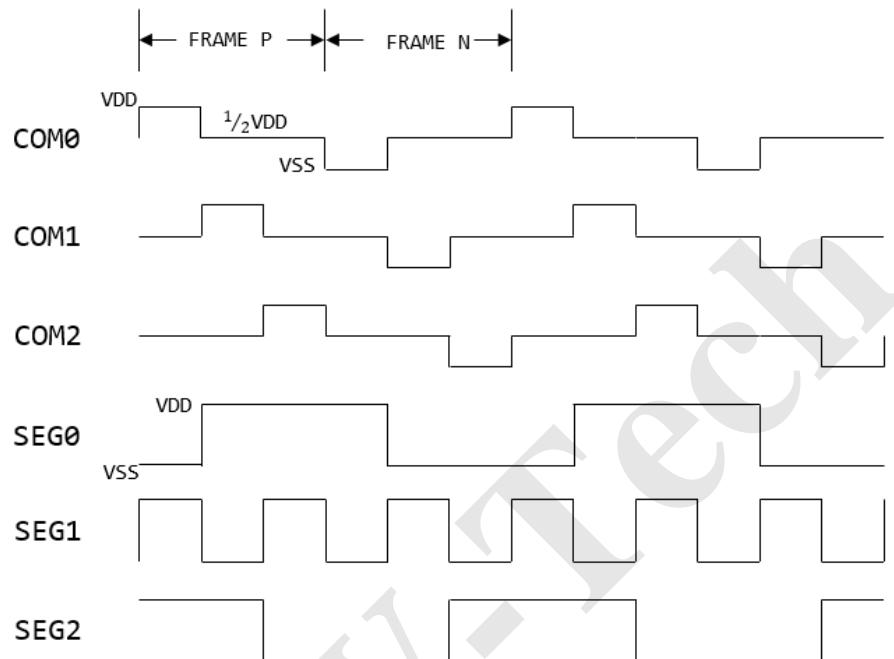


图 51 LCD波形

要点亮对应的COM/SEG，请将P帧中的SEG端口设置为VSS，COM端口设置为VDD，同样，将N帧中的SEG端口设置为VDD，COM端口设置为VSS。具体LCD端口设置请参见下表：

PADS	Light		Dark	
	Frame P	Frame N	Frame P	Frame N
COM	PxCOM=0 mode=output output=VDD	PxCOM=0 mode=output output=VSS	PxCOM=0 mode=output output=VDD	PxCOM=0 mode=output output=VSS
SEG	mode=output output=VSS	mode=output output=VDD	mode=output output=VSS	mode=output output=VDD

表 21-2 LCD端口设置

## 21.2 LCD控制寄存器

### 21.2.1 LCDCON0 (LCD 控制寄存器 0)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON0	0x18c	LCDEN	-	-	-	-	-	LCDRS1	LCDRS0
读/写属性	读/写	-	-	-	-	-	-	读/写	读/写
初始值	0	x	x	x	x	x	x	0	0

Bit 7 **LCDEN**: LCD 功能使能位。

1 = LCD 功能开启。

0 = LCD 功能关闭。

Bit 6 :2 未生效的。

Bit 1 :0 **LCDRS[1:0]**:  $\frac{1}{2}$  VDD 电流选择。

11 = 电流=100  $\mu$ A

10 = 电流=50  $\mu$ A

01 = 电流=16  $\mu$ A

00 = 电流=8  $\mu$ A

### 21.2.2 LCDCON1 (LCD 控制寄存器 1)

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON1	0x18d	P7COM	P6COM	P5COM	P4COM	P3COM	P2COM	P1COM	P0COM
读/写属性	读/写								
初始值	0	0	0	0	0	0	0	0	0

Bit 7 **P7COM**:  $\frac{1}{2}$  LCD 偏置选择。

1 = PC5 输出  $\frac{1}{2}$  VDD。

0 = PC5 是一般 IO 脚。

Bit 6 **P6COM**:  $\frac{1}{2}$  LCD 偏置选择。

1 = PC4 输出  $\frac{1}{2}$  VDD。

0 = PC4 是一般 IO 脚。

Bit 5 **P5COM**:  $\frac{1}{2}$  LCD 偏置选择。

1 = PC3 输出  $\frac{1}{2}$  VDD。

0 = PC3 是一般 IO 脚。

Bit 4	<b>P4COM:</b> 1/2 LCD 偏置选择。 1 = PC2 输出 1/2 VDD。 0 = PC2 是一般 IO 脚。
Bit 3	<b>P3COM:</b> 1/2 LCD 偏置选择。 1 = PB7 输出 1/2 VDD。 0 = PB7 是一般 IO 脚。
Bit 2	<b>P2COM:</b> 1/2 LCD 偏置选择。 1 = PB6 输出 1/2 VDD。 0 = PB6 是一般 IO 脚。
Bit 1	<b>P1COM:</b> 1/2 LCD 偏置选择。 1 = PB5 输出 1/2 VDD。 0 = PB5 是一般 IO 脚。
Bit 0	<b>P0COM:</b> 1/2 LCD 偏置选择。 1 = PB4 输出 1/2 VDD。 0 = PB4 是一般 IO 脚。

**注意:** 当  $PxCOM=1$  时, 对应的端口输出缓冲区将自动关闭。

## 22. 片上仿真 (OCD)

### 22.1 概述

AT8BM84A内嵌片上调试仿真功能(OCD)，为开发人员提供一种低成本的调试用户代码的方法。OCD提供了程序流程控制的调试能力，包括3个硬件地址断点、1个条件寄存器断点、单步、自由运行和内存访问命令。OCD系统不占用内存映射中的任何位置，也不共享任何片上外围设备。

OCD系统使用SCL和SDA两线串行接口，在目标设备和控制调试器主机之间建立通信。SDA是用于调试数据传输的输入/输出引脚，SCL是用于与SDA同步的输入引脚。AT8BM84A也使用SCL和SDA作为控制引脚来写入和读取它。

### 22.2 OCD限制

AT8BM84A是一个功能齐全的微控制器，在其有限的I/O引脚上多路复用多个功能。必须牺牲一些设备功能来为OCD系统提供资源。OCD有以下限制：

1、SCL/SDA引脚物理上位于同一引脚 PC1/PC0 或 PA3/PA2 上。因此，它的I/O功能和共享的多功能都无法仿真。

2、系统时钟不能关闭，因为OCD使用该时钟来监视其内部状态：当系统处于睡眠模式时，由于设备的某些部分可能没有时钟，所以执行ram/寄存器访问是无效的。读访问可能返回无用的信息，或者写访问可能不会成功。但以下访问不受系统停止的影响：读取当前程序地址、当前PCL、当前中断条件和当前睡眠状态。

## 23. 系统内程序 (ISP)

### 23.1 概述

系统内程序 (In-System Program, ISP) 是在芯片安装到电路板上后更新产品固件的一种方法。AT8BM84A基于MTP的存储器可以通过ISP使用三个引脚进行编程。它简化了升级代码的过程，甚至在产品已经部署到最终用户之后，并且很容易在制造过程中校准系统，从而减少了上市时间。

板上编程：ISP无需从电路板上移除AT8BM84A芯片，即可实现代码更新，方便终端用户升级系统和制造商提前准备生产计划。包括选项在内的整个代码可以在字段中重写。

在制造过程中校准系统：存储在MTP中的校准或定制参数也可以在制造或测试的最后阶段使用ISP更新。

串行编程专用引脚：

- ISPSDA：输入和输出数据。
- ISPSCK：时钟输入引脚，确保在编程期间数据同步。
- ISPVPP：VPP输入引脚，电压高。
- VSS：参考GND。

这三个引脚需要与VDD $\geq 3.5V$ 和GND引脚正确连接，才能开启ISP。

开启ISP功能：为了开启ISP功能，必须在NYIDE上勾选“ISP选项”。

ATW ISP工具：ATW为AT8BM84A提供了一个编程工具NYISP，使ISP可以轻松、友好地进行代码更新。

### 23.2 ISP限制

为避免漏电流，三个信号引脚 ISPSDA和ISPSCK和VPP必须设置为输入拉高I/O。ISP模式不允许客户使用非编程设备制造电路板。

## 24. 指令设置

AT8BM84A 为各种应用程序提供了 40 个功能强大的指令。

指令	助记符		说明	周期数	影响标志	指令	助记符		说明	周期数	影响标志
	1	2					1	2			
<b>算术指令</b>											
ANDAR	R	d	dest = ACC & R	1	Z	ADDAR	R	d	dest = R + ACC	1	Z, DC, C
IORAR	R	d	dest = ACC   R	1	Z	SUBAR	R	d	dest = R + (~ACC)	1	Z, DC, C
XORAR	R	d	dest = ACC $\oplus$ R	1	Z	ADCAR	R	d	dest = R + ACC + C	1	Z, DC, C
ANDIA	i		ACC = ACC & i	1	Z	SBCAR	R	d	dest = R + (~ACC) + C	1	Z, DC, C
IORIA	i		ACC = ACC   i	1	Z	ADDIA	i		ACC = i + ACC	1	Z, DC, C
XORIA	i		ACC = ACC $\oplus$ i	1	Z	SUBIA	i		ACC = i + (~ACC)	1	Z, DC, C
RRR	R	d	Rotate right R	1	C	ADCIA	i		ACC = i + ACC + C	1	Z, DC, C
RLR	R	d	Rotate left R	1	C	SBCIA	i		ACC = i + (~ACC) + C	1	Z, DC, C
BSR	R	bit	Set bit in R	1	-	CLRA			Clear ACC	1	Z
BCR	R	bit	Clear bit in R	1	-	CLRR			Clear R	1	Z
INCR	R	d	Increase R	1	Z						
DECR	R	d	Decrease R	1	Z						
COMR	R	d	dest = ~R	1	Z						
<b>条件指令</b>											
BTRSC	R	bit	Test bit in R, skip if clear	1 or 2	-	NOP			No operation	1	-
BTRSS	R	bit	Test bit in R, skip if set	1 or 2	-	SLEEP			Go into Halt mode	1	/TO, /PD
INCRSZ	R	d	Increase R, skip if 0	1 or 2	-	CLRWDT			Clear Watch-Dog Timer	1	/TO, /PD
DECRSZ	R	d	Decrease R, skip if 0	1 or 2	-	RET			Return from subroutine	2	-
<b>数据传送指令</b>											
MOVAR	R		Move ACC to R	1	-	RETIE			Return from interrupt and enable interrupt	2	-
MOVR	R	d	Move R	1	Z	RETIA	i		Return, place immediate in ACC	2	-
MOVIA	i		Move immediate to ACC	1	-	LCALL	adr		Call subroutine(4K)	2	-
SWAPR	R	d	Swap halves R	1	-	LGOTO	adr		unconditional branch(4K)	2	-
TABLEA			Read ROM	2	-						

表 24-1 指令表

- ACC:** 累加器。
- adr:** 地址。.
- bit:** 8 位寄存器 R 的 bit 地址。
- C:** 进位/借位。  
C=1, 加法指令有进位, 减法指令无借位。  
C=0, 加法指令无进位, 减法指令有借位。
- d:** 目标。  
若d=“0” , 结果存入ACC。  
若d=“1” , 结果存入R寄存器。
- DC:** 数字进位标志。
- dest:** 目标。
- i:** 8 位立即数。
- PC:** 程序计数器。
- PCHBUF:** 程序计数器高字节。
- /PD:** 睡眠标志位。  
/PD=1, 上电或CLRWDT指令执行后。  
/PD=0, SLEEP指令执行后。
- Prescaler:** 预分频器。
- R:** R页面特殊功能寄存器, R值为 0x0~0x7F。
- T0MD:** T0MD寄存器。
- TBHP:** 表格指针高字节寄存器。
- TBHD:** 表格数据高字节寄存器。
- /TO:** 看门狗超时标志位。  
/TO=1, 上电或执行 CLRWDT 或 SLEEP 指令后。  
/TO=0, 看门狗超时。
- WDT:** 看门狗计时器。
- Z:** 清零标志。

ADCAR	Add ACC and R with Carry	ADDAR	Add ACC and R
语法	ADCAR R, d	语法	ADDAR R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$	操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$R + ACC + C \rightarrow dest$	操作	$ACC + R \rightarrow dest$
状态影响	Z, DC, C	状态影响	Z, DC, C
说明	ACC和R带进位加法: 若d="0", 结果存入ACC; 若d="1", 结果存入"R"。	说明	ACC和R加法: 若d="0", 结果存入ACC; 若d="1", 结果存入"R"。
周期	1	周期	1
举例	ADCAR R, d 执行指令前: ACC=0x12, R=0x34, C=1, d=1, 执行指令后: R=0x47, ACC=0x12, C=0.	举例	ADDAR R, d 执行指令前: ACC=0x12, R=0x34, C=1, d=1, 执行指令后: R=0x46, ACC=0x12, C=0.
ADCI A	Add ACC and Immediate with Carry	ADDIA	Add ACC and Immediate
语法	ADCI A i	语法	ADDIA i
操作数	$0 \leq i < 255$	操作数	$0 \leq i < 255$
操作	$ACC + i + C \rightarrow ACC$	操作	$ACC + i \rightarrow ACC$
状态影响	Z, DC, C	状态影响	Z, DC, C
说明	ACC和8位立即数带进位加法, 结果存入ACC。	说明	ACC和8位立即数加法, 结果存入ACC。
周期	1	周期	1
举例	ADCI A i 执行指令前: ACC=0x12, i=0x34, C=1, 执行指令后: ACC=0x47, C=0.	举例	ADDIA i 执行指令前: ACC=0x12, i=0x34, C=1, 执行指令后: ACC=0x46, C=0.

ANDAR	AND ACC and R	BCR	Clear Bit in R
语法	ANDAR R, d	语法	BCR R, bit
操作数	$0 \leq R \leq 511$ $d = 0, 1.$	操作数	$0 \leq R \leq 511$ $0 \leq bit \leq 7$
操作	ACC & R dest	操作	$0 R[bit]$
状态影响	Z	状态影响	--
说明	ACC和R做“AND”运算；若d=“0”，结果存入ACC；若d=“1”，结果存入“R”。	说明	将R寄存器的bit位清 0。
周期	1	周期	1
举例	ANDAR R, d 执行指令前： ACC=0x5A, R=0xAF, d=1. 执行指令后： R=0x0A, ACC=0x5A, Z=0.	举例	BCR R, B2 执行指令前： R=0x5A, B2=0x3, 执行指令后： R=0x52.

ANDIA	AND Immediate with ACC	BSR	Set Bit in R
语法	ANDIA i	语法	BSR R, bit
操作数	$0 \leq i < 255$	操作数	$0 \leq R \leq 511$ $0 \leq bit \leq 7$
操作	ACC & i→ACC	操作	$1 \rightarrow R[bit]$
状态影响	Z	状态影响	--
说明	ACC和 8 位立即数做“AND”运算。结果存入ACC。	说明	将R寄存器的bit位置 1。
周期	1	周期	1
举例	ANDIA i 执行指令前： ACC=0x5A, i=0xAF, 执行指令后： ACC=0x0A, Z=0.	举例	BSR R, B2 执行指令前： R=0x5A, B2=0x2, 执行指令后： R=0x5E.

BTRSC	Test Bit in R and Skip if Clear	CLRA	Clear ACC
语法	BTRSC R, bit	语法	CLRA
操作数	$0 \leq R \leq 511$ $0 \leq \text{bit} \leq 7$	操作数	--
操作	Skip next instruction, if $R[\text{bit}] = 0$ .	操作	$00h \rightarrow \text{ACC}$ $1 \rightarrow Z$
状态影响	--	状态影响	Z
说明	位判断指令, 为“0”则跳过下一条指令。	说明	ACC清零, Z标志位置“1”。
周期	1 or 2(skip)	周期	1
举例	BTRSC R, B2 指令 1 指令 2 执行指令前: $R=0x5A, B2=0x2$ , 执行指令后: 由于 $R[B2]=0$ , 则指令 1 不执行, 程序直接从指令 2 开始执行。	举例	CLRA 执行指令前: $ACC=0x55, Z=0$ . 执行指令后: $ACC=0x00, Z=1$ .

BTRSS	Test Bit in R and Skip if Set	CLRR	Clear R
语法	BTRSS R, bit	语法	CLRR R
操作数	$0 \leq R \leq 511$ $0 \leq \text{bit} \leq 7$	操作数	$0 \leq R \leq 511$
操作	Skip next instruction, if $R[\text{bit}] = 1$ .	操作	$00h \rightarrow R$ $1 \rightarrow Z$
状态影响	--	状态影响	Z
说明	位判断指令, 为“1”则跳过下一条指令。	说明	寄存器R清零, Z标志位置“1”。
周期	1 or 2(skip)	周期	1
举例	BTRSS R, B2 指令 2 指令 3 执行指令前: $R=0x5A, B2=0x3$ , 执行指令后: 由于 $R[B2]=1$ , 则指令 2 不执行, 直接从指令 3 开始执行。	举例	CLRR R 执行指令前: $R=0x55, Z=0$ . 执行指令后: $R=0x00, Z=1$ .

CLRWDT	Clear Watch-Dog Timer	DECR	Decrease R
语法	CLRWDT	语法	DECR R, d
操作数	--	操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	00h→WDT, 00h→WDT prescaler 1→/TO 1→/PD	操作	$R - 1 \rightarrow dest$
状态影响	/TO, /PD	状态影响	Z
说明	清WDT计数器和预分频器; /TO和 /PD标志位置“1”。	说明	$R - 1$ , 若 $d = 0$ , 结果存入ACC; 若 $d = 1$ , 结果存入R。
周期	1	周期	1
举例	CLRWDT 执行指令前: /TO=0 执行指令后: /TO=1	举例	DECR R, d 执行指令前: $R = 0x01, d = 1, Z = 0.$ 执行指令后: $R = 0x00, Z = 1.$

COMR	Complement R	DECRSZ	Decrease R, Skip if 0
语法	COMR R, d	语法	DECRSZ R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$	操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$\sim R \rightarrow dest$	操作	$R - 1 \rightarrow dest$ , Skip if result = 0
状态影响	Z	状态影响	--
说明	R寄存器取反, 结果存入d; $d = 0$ , 结果存入ACC; $d = 1$ , 结果存入R。	说明	R先-1, 若 $d = 0$ , 结果存入ACC; 若 $d = 1$ , 结果存入R, 若结果为“0” 则跳过下一条指令, 改为执行NOP 指令, 因此结果为“0”时要执行两个 周期。
周期	1	周期	1 or 2(skip)
举例	COMR, d 执行指令前: $R = 0xA6, d = 1, Z = 0.$ 执行指令后: $R = 0x59, Z = 0.$	举例	DECRSZ R, d 指令 2 指令 3 执行指令前: $R = 0x1, d = 1, Z = 0.$ 执行指令后: $R = 0x0, Z = 1$ , 操作结果为 0, 指令 2 被跳过。

INCR	Increase R	IORAR	OR ACC with R
语法	INCR R, d	语法	IORAR R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$	操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$R + 1 \rightarrow \text{dest.}$	操作	$\text{ACC}   R \rightarrow \text{dest}$
状态影响	Z	状态影响	Z
说明	R+ 1, 若d="0", 结果存入ACC; 若d="1", 结果存入R。	说明	ACC和R做“OR”运算, 若d="0", 结果存入ACC; 若d="1", 结果存入R。
周期	1	周期	1
举例	INCR R, d 执行指令前: R=0xFF, d=1, Z=0. 执行指令后: R=0x00, Z=1.	举例	IORAR R, d 执行指令前: R=0x50, ACC=0xAA, d=1, Z=0. 执行指令后: R=0xFA, ACC=0xAA, Z=0.

INCRSZ	Increase R, Skip if 0	IORIA	OR Immediate with ACC
语法	INCRSZ R, d	语法	IORIA i
操作数	$0 \leq R \leq 511$ $d = 0, 1.$	操作数	$0 \leq i < 255$
操作	$R + 1 \rightarrow \text{dest.}$ Skip if result = 0	操作	$\text{ACC}   i \rightarrow \text{ACC}$
状态影响	--	状态影响	Z
说明	R先+ 1, 若d="0", 结果存入ACC; 若d="1", 结果存入R。若结果为“0” 则跳过下一条指令 (执行 NOP 指令)。	说明	ACC和 8 位立即数做“OR”运算, 结果存入ACC。
周期	1 or 2(skip)	周期	1
举例	INCRSZ R, d 指令 2, 指令 3. 执行指令前: R=0xFF, d=1, Z=0. 执行指令后: R=0x00, Z=1. 因结果为 0, 程序跳过指令 2。	举例	IORIA i 执行指令前: i=0x50, ACC=0xAA, Z=0. 执行指令后: ACC=0xFA, Z=0.

LCALL	Call Subroutine	MOVAR	Move ACC to R
语法	LCALL adr	语法	MOVAR R
操作数	$0 \leq adr \leq 4095$	操作数	$0 \leq R \leq 511$
操作	$PC + 1 \rightarrow$ Top of Stack, adr $\rightarrow PC[11:0]$	操作	ACC $\rightarrow$ R
状态影响	--	状态影响	--
说明	长调用子程序。首先将PC+1压入栈顶，然后将12位立即数载入PC[11:0]。	说明	ACC赋值给R。
周期	2	周期	1
举例	LCALL SUB 执行指令前： PC=A0. Stack level=1 执行指令后： PC=address of SUB, Stack[1]= A0+1, Stack pointer =2.	举例	MOVAR R 执行指令前： R=0x55, ACC=0xAA. 执行指令后： R=0xAA, ACC=0xAA.

LGOTO	Unconditional Branch	MOVIA	Move Immediate to ACC
语法	LGOTO adr	语法	MOVIA i
操作数	$0 \leq adr \leq 4095$	操作数	$0 \leq i < 255$
操作	adr $\rightarrow PC[11:0]$ .	操作	i $\rightarrow$ ACC
状态影响	--	状态影响	--
说明	无条件长跳转，12位立即数写入PC[11:0]。	说明	8位立即数赋值给ACC。
周期	2	周期	1
举例	LGOTO Level 执行指令前： PC=A0. 执行指令后： PC=address of Level.	举例	MOVIA i 执行指令前： i=0x55, ACC=0xAA. 执行指令后： ACC=0x55.

MOV R	Move R to ACC or R	RETIE	Return from Interrupt and Enable Interrupt Globally
语法	MOV R R, d	语法	RETIE
操作数	$0 \leq R \leq 511$ $d = 0, 1.$	操作数	--
操作	$R \rightarrow \text{dest}$	操作	Top of Stack $\rightarrow$ PC $1 \rightarrow \text{GIE}$
状态影响	Z	状态影响	--
说明	R 赋值给 d, 若 d="0", 结果存入 ACC; 若 d="1", 结果存入寄存器 R。指令执行后, 通过状态标位 Z 检查 R 是否为 0。	说明	中断返回, 栈顶地址载入 PC 同时使能中断。
周期	1	周期	2
举例	MOV R, d 执行指令前: R=0x0, ACC=0xAA, Z=0, d=0. 执行指令后: R=0x0, ACC=0x00, Z=1.	举例	RETIE 执行指令前: GIE=0, Stack level=2. 执行指令后: GIE=1, PC=Stack[2], Stack pointer=1.

NOP	No Operation
语法	NOP
操作数	--
操作	No operation.
状态影响	--
说明	空操作。
周期	1
举例	NOP 执行指令前: PC=A0 执行指令后: PC=A0+1

RETIA	Return with Data in ACC
语法	RETIA i
操作数	$0 \leq i < 255$
操作	$i \rightarrow \text{ACC}$ , Top of Stack $\rightarrow$ PC
状态影响	--
说明	带参数返回: 8 位立即数赋值给 ACC, 栈顶地址载入 PC。
周期	2
举例	RETIA i 执行指令前: Stack pointer = 2. i=0x55, ACC=0xAA. 执行指令后: PC=Stack[2], Stack pointer = 1. ACC=0x55.

RET	Return from Subroutine
语法	RET
操作数	--
操作	Top of Stack→PC
状态影响	--
说明	子程序返回，栈顶载入PC。
周期	2
举例	RET 执行指令前： Stack level=2. 执行指令后： PC=Stack[2], Stack level=1.

RRR	Rotate Right R Through Carry
语法	RRR R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$C \rightarrow \text{dest}[7]$ , $R[7:1] \rightarrow \text{dest}[6:0]$ , $R[0] \rightarrow C$
状态影响	
说明	带进位R循环右移：若d=“0”，结果存入ACC；若d=“1”，结果存入R。
周期	1
举例	RRR R, d 执行指令前： R=0xA5, d=1, C=0. 执行指令后： R=0x52, C=1.

RLR	Rotate Left R Through Carry
语法	RLR R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$C \rightarrow \text{dest}[0]$ , $R[7] \rightarrow C$ , $R[6:0] \rightarrow \text{dest}[7:1]$

状态影响	
说明	带进位R循环左移：若d=“0”，结果存入ACC；若d=“1”，结果存入R。
周期	1
举例	RLR R, d 执行指令前： R=0xA5, d=1, C=0. 执行指令后： R=0x4A, C=1.

SBCAR	Subtract ACC and Carry from R	SBCIA	Subtract ACC and Carry from Immediate
语法	SBCAR R, d	语法	SBCIA i
操作数	$0 \leq R \leq 511$ $d = 0, 1.$	操作数	$0 \leq i < 255$
操作	$R + (\sim ACC) + C \rightarrow dest$	操作	$i + (\sim ACC) + C \rightarrow dest$
状态影响	$Z, DC, C$	状态影响	$Z, DC, C$
说明	$R$ 和ACC带借位减法, 若 $d=0$ , 结果存入ACC; 若 $d=1$ , 结果存入R。	说明	常数和ACC带借位减法, 结果存入ACC。
周期	1	周期	1
举例	SBCAR R, d (a) 执行指令前: $R=0x05, ACC=0x06, d=1,$ $C=0,$ 执行指令后: $R=0xFE, C=0. (-2)$ (b) 执行指令前: $R=0x05, ACC=0x06, d=1,$ $C=1,$ 执行指令后: $R=0xFF, C=0. (-1)$ (c) 执行指令前: $R=0x06, ACC=0x05, d=1,$ $C=0,$ 执行指令后: $R=0x00, C=1. (-0), Z=1.$ (d) 执行指令前: $R=0x06, ACC=0x05, d=1,$ $C=1,$ 执行指令后: $R=0x1, C=1. (+1)$	举例 (a) 执行指令前: $i=0x05, ACC=0x06, C=0,$ 执行指令后: $ACC=0xFE, C=0. (-2)$ (b) 执行指令前: $i=0x05, ACC=0x06, C=1,$ 执行指令后: $ACC=0xFF, C=0. (-1)$ (c) 执行指令前: $i=0x06, ACC=0x05, C=0,$ 执行指令后: $ACC=0x00, C=1. (-0), Z=1.$ (d) 执行指令前: $i=0x06, ACC=0x05, C=1,$ 执行指令后: $ACC=0x1, C=1. (+1)$	

SLEEP	Enter Halt Mode	SUBIA	Subtract ACC from Immediate
语法	SLEEP	语法	SUBIA i
操作数	--	操作数	$0 \leq i < 255$
操作	00h→WDT, 00h→WDT prescaler 1→TO 0→PD	操作	$i - ACC \rightarrow ACC$
状态影响	/TO, /PD	状态影响	Z, DC, C
说明	WDT和分频器 0 清零。/TO标志为 0, /PD清零, IC进入睡眠。	说明	8 位立即数减ACC, 结果存入ACC。
周期	1	周期	1
举例	SLEEP 执行指令前: /PD=1, /TO=0. 执行指令后: /PD=0, /TO=1.	举例	SUBIA i (a) 执行指令前: i=0x05, ACC=0x06. 执行指令后: ACC=0xFF, C=0. (-1) (b) 执行指令前: i=0x06, ACC=0x05, d=1, 执行指令后: ACC=0x01, C=1. (+1)

SUBAR	Subtract ACC from R	SWAPR	Swap High/Low Nibble in R
语法	SUBAR R, d	语法	SWAPR R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$	操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$R - ACC \rightarrow dest$	操作	$R[3:0] \rightarrow dest[7:4].$ $R[7:4] \rightarrow dest[3:0]$
状态影响	Z, DC, C	状态影响	--
说明	R 减去ACC, 若d="0", 结果存入ACC。若d="1", 结果存入R。	说明	寄存器半字节交换, 若d="0", 结果存入ACC; 若d="1", 结果存入R。
周期	1	周期	1
举例	SBCAR R, d (a) 执行指令前: R=0x05, ACC=0x06, d=1, 执行指令后: R=0xFF, C=0. (-1) (b) 执行指令前: R=0x06, ACC=0x05, d=1, 执行指令后: R=0x01, C=1. (+1)	举例	SWAPR R, d 执行指令前: R=0xA5, d=1. 执行指令后: R=0x5A.

TABLEA	Read ROM data	XORIA	Exclusive-OR Immediate with ACC
语法	TABLEA	语法	XORIA i
操作数	--	操作数	$0 \leq i < 255$
操作	ROM data{ TBHP, ACC } [7:0] ACC ROM data{TBHP, ACC} [15:8] TBHD.	操作	$ACC \oplus i \rightarrow ACC$
状态影响	--	状态影响	Z
说明	{TBHP[2:0], ACC}所指向的ROM的8个最低有效位被放置到ACC。 {TBHP[2:0], ACC}所指向的6个最高位被放置到TBHD[7:0].	说明	ACC和8位立即数做“XOR”运算。
周期	2	周期	1
举例	TABLEA 执行指令前: TBHP=0x02, CC=0x34. TBHD=0x01. ROM data[0x234]= 0x35AA 执行指令后: TBHD=0x35, ACC=0xAA.	举例	XORIA i 执行指令前: i=0xA5, ACC=0xF0. 执行指令后: ACC=0x55.
、			

XORAR	Exclusive-OR ACC with R
语法	XORAR R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$ACC \oplus R \rightarrow dest$
状态影响	Z
说明	ACC和R做“XOR”运算，若d=“0”，结果存入ACC；若d=“1”，结果存入R。
周期	1
举例	XORAR R, d 执行指令前: R=0xA5, ACC=0xF0, d=1. 执行指令后: R=0x55.

## 25.配置表

项目	名称	选项		
1	High Oscillator Frequency 高频振荡模式	1. I_HRC	2. E_HXT	3. E_XT
2	Low Oscillator Frequency 低频振荡模式	1. I_LRC	2. E_LXT	
3	High IRC Frequency 内部高速 RC 频率	1. 1MHz 4. 8MHz	2. 2MHz 5. 16MHz	3. 4MHz 6. 24MHz
4	High Crystal Oscillator 外部高频振荡器	1. $6\text{MHz} < F_{HOSC} \leq 8\text{MHz}$ 3. $10\text{MHz} < F_{HOSC} \leq 12\text{MHz}$ 5. $16\text{MHz} < F_{HOSC} \leq 24\text{MHz}$	2. $8\text{MHz} < F_{HOSC} \leq 10\text{MHz}$ 4. $12\text{MHz} < F_{HOSC} \leq 16\text{MHz}$	
5	Instruction Clock 指令时钟	1. 2 oscillator period 2 个振荡周期	2. 4 oscillator period 4 个振荡周期	
6	WDT 看门狗定时器	1. Watchdog Enable (Software control) 2. Watchdog Disable (Always disable)	看门狗开启 (软件控制) 看门狗关闭 (永远关闭)	
7	WDT Event 看门狗定时器事件	1. Watchdog Reset 看门狗复位	2. Watchdog Interrupt 看门狗中断	
8	Timer0 Source 定时器 0 时钟源	1. EXCKIO	2. Low Oscillator (I_LRC/E_LXT) 低频振荡器 (I_LRC/E_LXT)	
9	PA.5	1. PA.5 is I/O PA5 是 I/O 脚。	2. PA.5 is reset PA5 是复位脚。	
10	PA.7	1. PA.7 is I/O PA7 是 I/O 脚。	2. PA.7 is instruction clock output PA7 是指令时钟输出。	
11	IR Pad	1. PB1	2. PA3	
12	Startup Time 启动时间	1. 500us 2. 4.5ms 3. 18ms 4. 4.72ms 5. 288ms		
13	WDT Time Base 看门狗定时器时基	1. 3.5ms 2. 15ms 3. 60ms 4. 250ms		
14	Noise Filter (High_EFT) 噪声过滤功能 (High_EFT)	1. Enable (开启) 2. Disable (关闭)		
15	LVR Setting LVR 开关设定	1. Register Control 寄存器控制 3. Always On LVR 一直开启	2. Register Control + Halt mode Off 寄存器控制+睡眠模式自动关闭 4. Operation mode On + Halt mode Off 一般模式开启+睡眠模式自动关闭	
16	LVR Voltage LVR 电压	1. 1.6V 6. 2.7V 7. 3.0V 8. 3.3V 9. 3.6V	2. 1.8V 3. 2.0V 4. 2.2V 5. 2.4V	
17	VDD Voltage VDD 电压	1. 3.0V 2. 4.5V 3. 5.0V		
18	VHREF	1. PB1 2. PA0		
19	Sink current type PA3-0 Sink 电流型态	1. Normal 2. Large 3. Ultra	4. Super	
20	Sink current type Sink 电流型态	1. Normal 2. Large 3. Ultra		
21	Comparator Input pin select 比较器输入引脚选择	1. Enable 2. Disable		
22	Read Output Data 读取输出口数据	1. I/O Port 2. Register		
23	E_LXT Backup Control E_LXT 强化起振开关设定	1. Auto Off 自动关闭	2. Register Off 寄存器关闭	

项目	名称	选项	
24	EX_CKIO to Inst. Clock EXCKIO 到指令时钟	1. Sync 同步	2. Async 异步
25	Startup Clock 启动时钟	1. Fast (I_HRC/E_HXT/E_XT) 高速	2. Slow (I_LRC/E_LXT) 慢速
26	Input High Voltage (V <sub>IH</sub> ) 输入高电压 (V <sub>IH</sub> )	1. (0.7VDD)	2. (0.5VDD)
27	Input Low Voltage (V <sub>IL</sub> ) 输入低电压 (V <sub>IL</sub> )	1. (0.3VDD)	2. (0.2VDD)
28	External interrupt 0(INT0) 外部中断 0 (INT0)	1. PB4	2. PB0 or PA4
29	External interrupt 1(INT1) 外部中断 1 (INT1)	1. PB1	2. PA3
30	External interrupt 2(INT2) 外部中断 2 (INT2)	1. Disable	2. PA5
31	UART Pin UART 引脚	1. PB7 / PB6	2. PB0 / PB1
32	I <sup>2</sup> C Pin I <sup>2</sup> C 引脚	1. PB4 / PB5	2. PC3 / PC2
33	EX_CK1	1. PA1	2. PA2
34	PWM1 Output Pin PWM1 引脚	1. PA3	2. PA5
35	PWM2 Output Pin PWM2 引脚	1. PA7	2. PA4
36	PWM3 Output Pin PWM3 引脚	1. PA2	2. PA6
37	PWM4 Output Pin PWM4 引脚	1. PA1	2. PB3
38	PWM5 Output Pin PWM5 引脚	1. PA0	2. PB2
39	CCP1 dead band pair CCP1 死区对	1. PB2 / PA5	2. PC4 / PA1
40	Comparator P2P pair 0 (P/N) 比较器 P2P 对 0	1. PA4 / PB3	2. PA0 / PA1
41	PWM2_PWM4_PAD	1. PA4 / PB3	2. PB3 / PA4

## 26. 电气特性

### 26.1 最大绝对值

符号	参数	额定值		单位
$V_{DD} - V_{SS}$	工作电压	-0.5 ~ +6.0		V
$V_{IN}$	输入电压	$V_{SS}-0.3V \sim V_{DD}+0.3$		V
$T_{OP}$	工作温度	-40 ~ +85		°C
$T_{ST}$	储存温度	-40 ~ +125		°C

### 26.2 直流电气特性

(所有参考  $F_{INST}=F_{HOSC}/4$ ,  $F_{HOSC}=16MHz@I_HRC$ , WDT开启, 环境温度 $T_A=25^\circ C$  除其他指定说明外)

符号	参数	$V_{DD}$	最小值	典型值	最大值	单位	条件
$V_{DD}$	工作电压	--	3.0	--	5.5	V	$F_{INST}=12MHz @ I_HRC:24MHz/2T$
			2.4				$F_{INST}=8MHz @ I_HRC:16MHz/2T$
			2.0				$F_{INST}=6MHz @ I_HRC:24MHz/4T$
			2.0				$F_{INST}=4MHz @ I_HRC:16MHz/4T$
			1.6				$F_{INST}=2MHz @ I_HRC:8MHz/4T$
			1.6				$F_{INST}=2MHz @ E_HXT:8MHz/4T$
			1.6				$F_{INST}=1MHz @ I_LRC:4MHz/4T$
			1.6				$F_{INST}=1MHz @ E_LXT:4MHz/4T$
$V_{IH}$	输入高电平	5V	4.0	--	--	V	RSTb (0.8 $V_{DD}$ )
		3V	2.4	--	--		
		5V	3.5	--	--	V	All other I/O pins, EXCKI0/1, INT0/1/2 (0.7 $V_{DD}$ )
		3V	2.1	--	--		
		5V	2.5	--	--	V	All other I/O pins, EXCKI0/1 (0.5 $V_{DD}$ )
		3V	1.5	--	--		
$V_{IL}$	输入低电平	5V	--	--	1.0	V	RSTb (0.2 $V_{DD}$ )
		3V	--	--	0.6		
		5V	--	--	1.5	V	All other I/O pins, EXCKI0/1, INT0/1/2 (0.3 $V_{DD}$ )
		3V	--	--	0.9		
		5V	--	--	1.0	V	All other I/O pins, EXCKI0/1 (0.2 $V_{DD}$ )
		3V	--	--	0.6		
$I_{OH}$	输出高推电流 (一般电流)	5V	--	19	--	mA	$V_{OH}=4.0V$
		3V	--	11	--		$V_{OH}=2.0V$
$I_{OH}$	输出高推电流 (大电流)	5V		32	--	mA	$V_{OH}=4.0V$
		3V		20	--		$V_{OH}=2.0V$
$I_{OH}$	输出高推电流 (超大电流)	5V		45	--	mA	$V_{OH}=4.0V, @PA0~PA3$
		3V		30	--		$V_{OH}=2.0V, @PA0~PA3$
$I_{OL}$	输出低灌电流 (一般电流)	5V	--	28	--	mA	$V_{OL}=1.0V$
		3V	--	17	--		
$I_{OL}$	输出低灌电流 (大电流)	5V	--	40	--	mA	$V_{OL}=1.0V$
		3V	--	25	--		
$I_{OL}$	输出低灌电流 (极大电流)	5V	--	70	--	mA	$V_{OL}=1.0V$
		3V	--	45	--		

符号	参数	V <sub>DD</sub>	最小值	典型值	最大值	单位	条件
I <sub>OL</sub>	输出低灌电流 (超大电流)	5V	--	90	--	mA	V <sub>OL</sub> =1.0V, @PA0~PA3
		3V	--	62	--		
I <sub>OP</sub>	工作电流	正常模式					
		5V	--	3.1	--	mA	F <sub>INST</sub> =12MHz @ I_HRC:24MHz/2T @ E_HXT:24MHz/2T
		3V	--	1.9	--		
		5V	--	2.7	--	mA	F <sub>INST</sub> =6MHz @ I_HRC:24MHz/4T @ E_HXT:24MHz/4T
		3V	--	1.7	--		
		5V	--	2.9	--	mA	F <sub>INST</sub> =8MHz @ I_HRC:16MHz/2T @ E_HXT:16MHz/2T
		3V	--	1.7	--		
		5V	--	2.0	--	mA	F <sub>INST</sub> =4MHz @ I_HRC:16MHz/4T @ E_HXT:16MHz/4T
		3V	--	1.3	--		
		5V	--	1.4	--	mA	F <sub>INST</sub> =2MHz @ I_HRC:8MHz/4T @ E_HXT:8MHz/4T
		3V	--	1.0	--		
		5V	--	1.0	--	mA	F <sub>INST</sub> =1MHz @ I_HRC:4MHz/4T @ E_XT:4MHz/4T
		3V	--	0.8	--		
		慢速模式					
		5V	--	9	--	uA	F <sub>HOSC</sub> 关闭, F <sub>LOSC</sub> =32KHz @ I_LRC/2
		3V	--	6	--		
		5V	--	6	--	uA	F <sub>HOSC</sub> 关闭, F <sub>LOSC</sub> =32KHz @ I_LRC/4
		3V	--	4	--		
I <sub>STB</sub>	待机电流	5V	--	3.5	--	uA	待机模式, F <sub>HOSC</sub> 关闭, F <sub>LOSC</sub> =32KHz @ I_LRC/4
		3V	--	1.7	--		
I <sub>HALT</sub>	睡眠电流	5V	--	--	0.2	uA	睡眠模式, WDT关闭, LVR关闭
		3V	--	--	0.1		
		5V	--	--	3.0	uA	睡眠模式, WDT开启, LVR关闭
		3V	--	--	1.5		
R <sub>PH</sub>	上拉电阻	5V	--	58	--	KΩ	上拉电阻 (不含PA5)
		3V	--	107	--		
		5V	--	90	--	KΩ	上拉电阻 (PA5)
		3V	--	90	--		
R <sub>PL</sub>	下拉电阻	5V	--	60	--	KΩ	下拉电阻
		3V	--	120	--		

### 26.3 比较器/ LVD电气特性

(V<sub>DD</sub>=5V, V<sub>SS</sub>=0V, T<sub>A</sub>=25°C 除其他指定说明外)

符号	参数	最小值	典型值	最大值	单位	条件
V <sub>IVR</sub>	比较器输入电压范围	0	--	5	V	F <sub>HOSC</sub> =1MHz
T <sub>ENO</sub>	比较器启用输出有效时间	--	20	--	uS	F <sub>HOSC</sub> =1MHz
I <sub>CO</sub>	比较器电流消耗	--	135	--	uA	F <sub>HOSC</sub> =1MHz, P2V mode

符号	参数	最小值	典型值	最大值	单位	条件
$I_{LVD}$	LVD电流消耗	--	150	--	uA	$F_{HOSC}=1MHz$ , $LVD=4.15V$
$E_{LVD}$	LVD电压错误	--	5	--	%	$F_{HOSC}=1MHz$ , $LVD=4.15V$

## 26.4 OSC电气特性

(量测条件: 电压  $V_{DD}$ 、环境温度  $T_A$  与烧录时条件相同。)

参数	最小值	典型值	最大值	单位	条件
$I_{HRC}$ 偏移量 (通过转接座烧录)			$\pm 1$	%	用转接座直接从烧录器烧录。
$I_{HRC}$ 偏移量 (通过夹具烧录)			$\pm 3$	%	正确设置自动机台夹具时烧录。
$II_{LRC}$ 偏移量 (通过夹具烧录)			$\pm 5$	%	

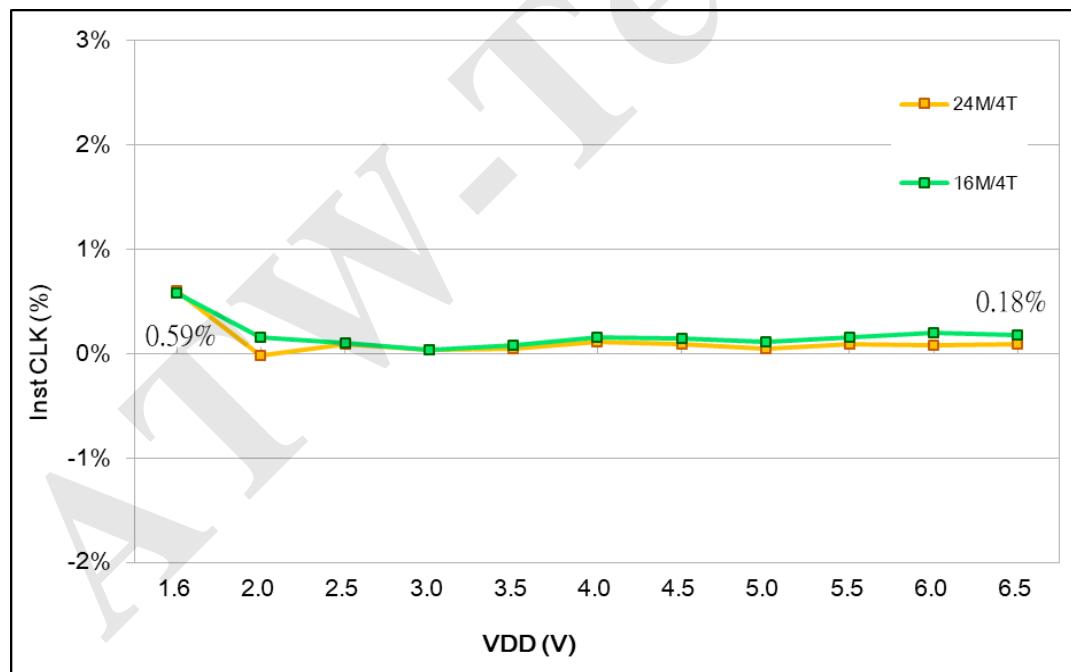
## 26.5 ADC电气特性

( $V_{DD}=5V$ ,  $V_{SS}=0V$ ,  $T_A=25^{\circ}C$  除其他指定说明外。)

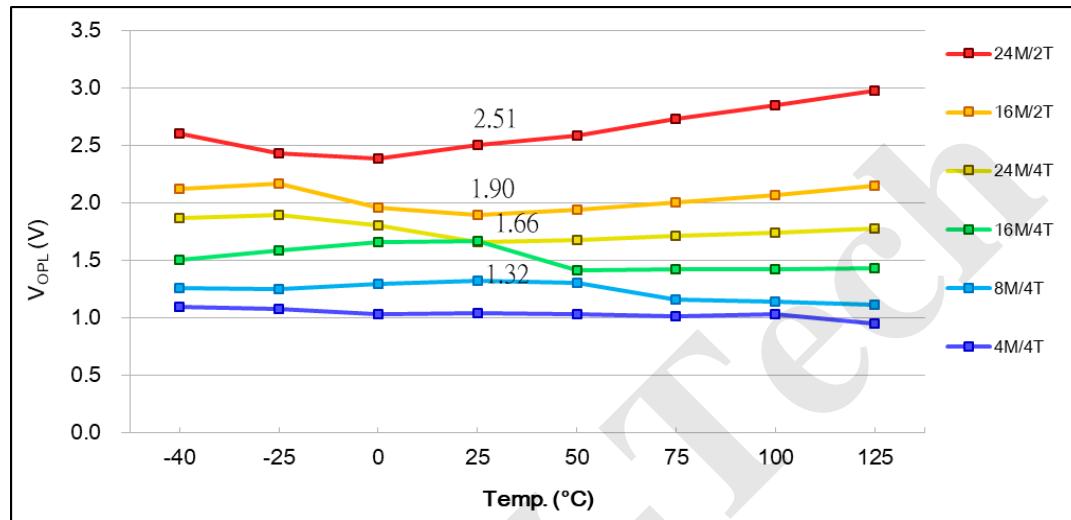
符号	参数	最小值	典型值	最大值	单位	条件
$V_{REFH}$	$V_{REFH}$ 输入电压	2V	--	$V_{DD}$	V	外部参考电压
$V_{REF4}$	内部参考电压 4V, $V_{DD}=5V$	3.95	4	4.05	V	
$V_{REF3}$	内部参考电压 3V, $V_{DD}=5V$	2.95	3	3.05	V	
$V_{REF2}$	内部参考电压 2V, $V_{DD}=5V$	1.95	2	2.05	V	
$V_{REF}$	内部参考电压 $V_{DD}$ , $V_{DD}=5V$	--	$V_{DD}$	--	V	
	内部参考电压	$V_{REF}+0.5$	--	--	V	最小供电电压
	ADC 模拟输入电压	0	--	$V_{REFH}$	V	
	ADC 开启时间	256	--	--	us	在设置 $ADENB= "1"$ 后准备开始转换。
$I_{OP(ADC)}$	ADC 电流消耗	--	0.3	--	mA	
ADCLK	ADC 时钟频率	--	--	2M	Hz	
ADCYCLE	ADC 转换时间周期	16	--		1/ADCLK	$SHCLK=2$ ADC clock
ADC <sub>sample</sub>	ADC 转换率	--	--	125	K/sec	$V_{DD}=5V$
DNL	非线性微分误差	$\pm 1$	--	--	LSB	$V_{DD}=5.0V$ , $AVREFH=5V$ , $FADSM=62.5K$
INL	非线性积分误差	$\pm 2$	--	--	LSB	
NMC	无缺码分辨率	10	11	12	Bits	

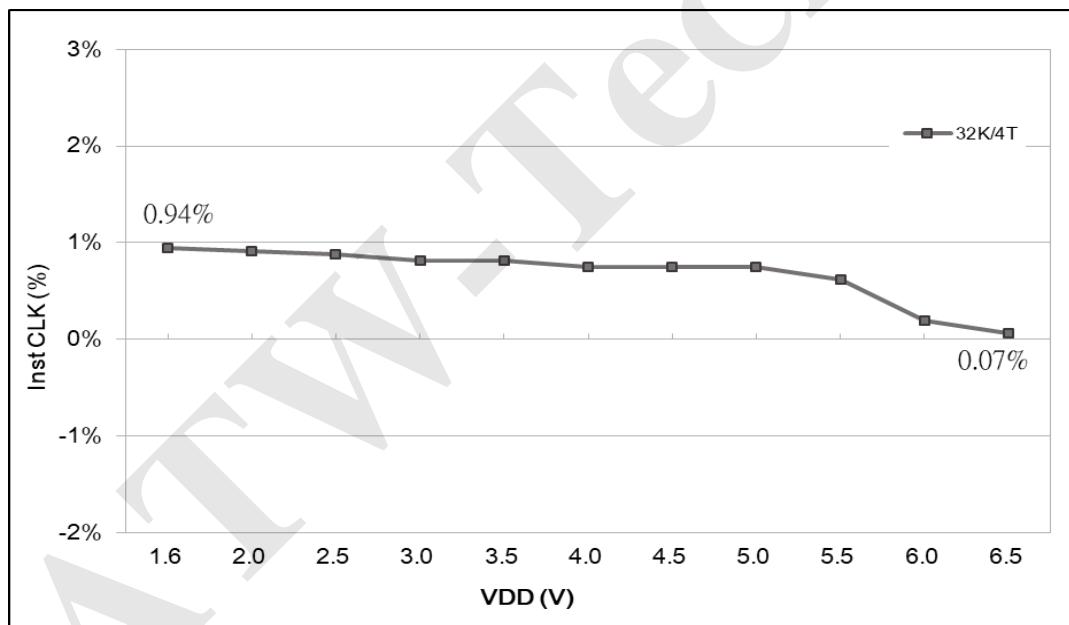
## 26.6 特性曲线图

### 26.6.1 I\_HRC 的频率与 V<sub>DD</sub> 曲线图

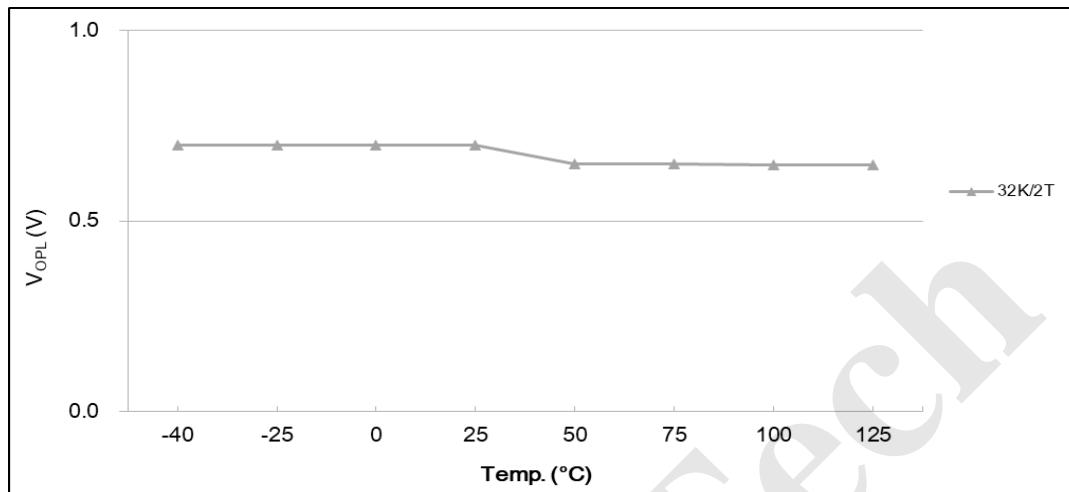


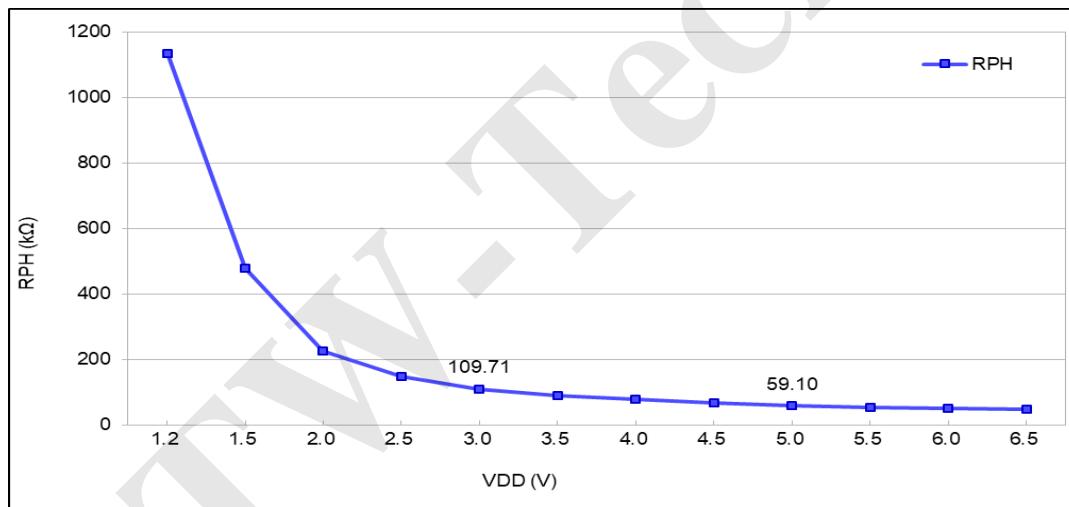
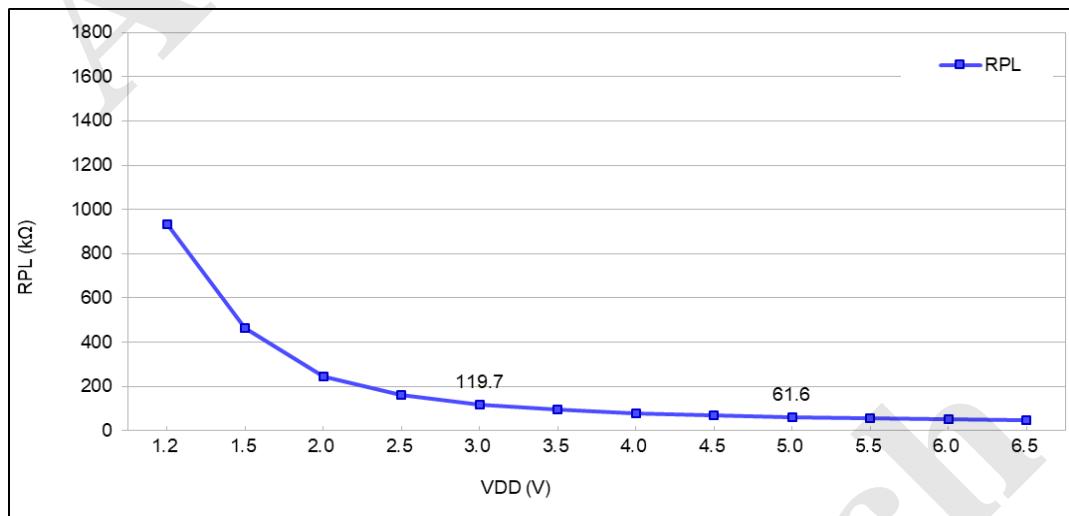
### 26.6.2 I\_HRC 的频率与温度曲线图



26.6.3 I\_LRC 的频率与  $V_{DD}$  曲线图

## 26.6.4 I\_LRC 的频率与温度曲线图



26.6.5 上拉电阻与  $V_{DD}$  曲线图26.6.6 下拉电阻与  $V_{DD}$  曲线图

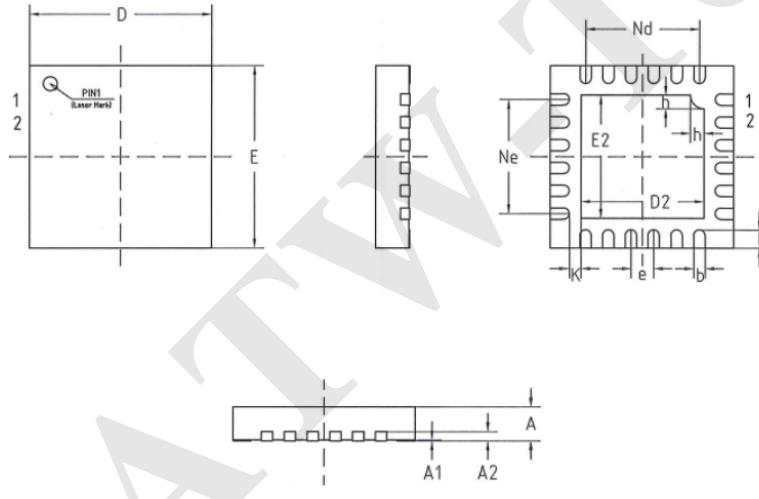
## 26.7 推荐工作电压

推荐工作电压 (温度范围: -40°C ~ +85°C)

频率	最小电压	LVR: 建议值 (-40°C ~ +85°C)
24M/2T	3.6V	3.3V
16M/2T	3.3V	3.0V
24M/4T	2.7V	2.4V
16M/4T	2.2V	2.0V
8M/4T	1.8V	1.8V
4M/4T	1.8V	1.8V

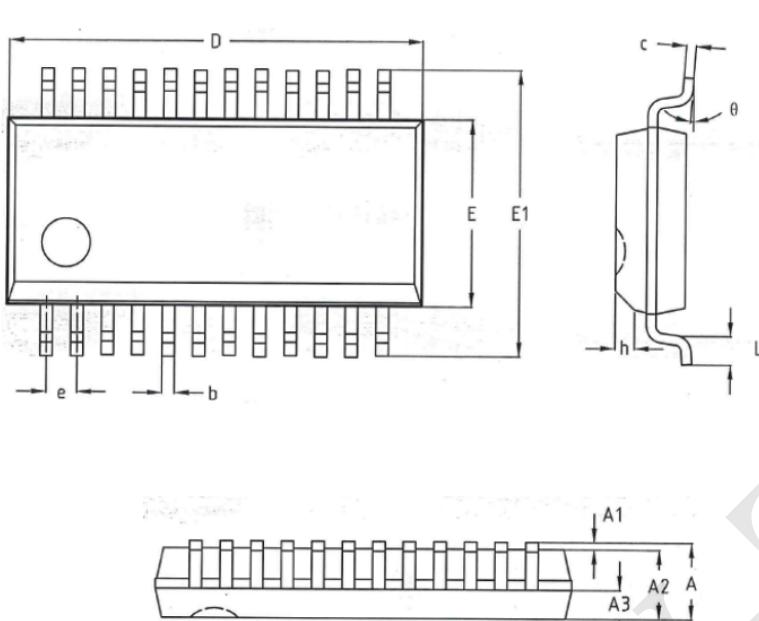
## 27.封装尺寸

## 27.1 24 引脚QFN



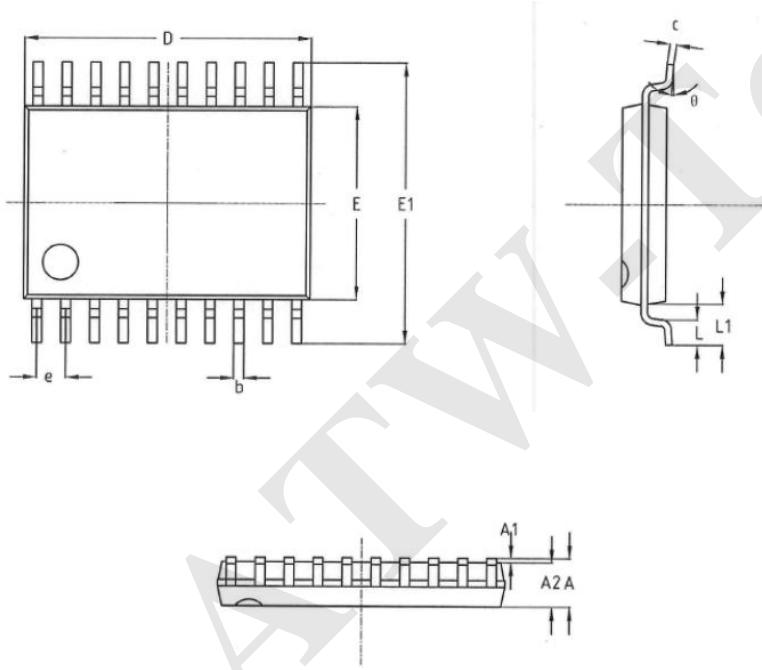
SYMBOL	MILLIMETERS		
	MIN	TYP	MAX
A	0.70	0.75	0.80
A1	-	0.02	0.05
A2		0.203 REF	
b	0.20	0.25	0.30
D	3.90	4.00	4.10
D2	2.60	2.70	2.80
E	3.90	4.00	4.10
E2	2.60	2.70	2.80
e		0.50 BSC	
K	0.20	0.25	0.30
L	0.35	0.40	0.45
h	0.25	0.30	0.35
Ne		2.50 BSC	
Nd		2.50 BSC	

## 27.2 24 引脚SSOP



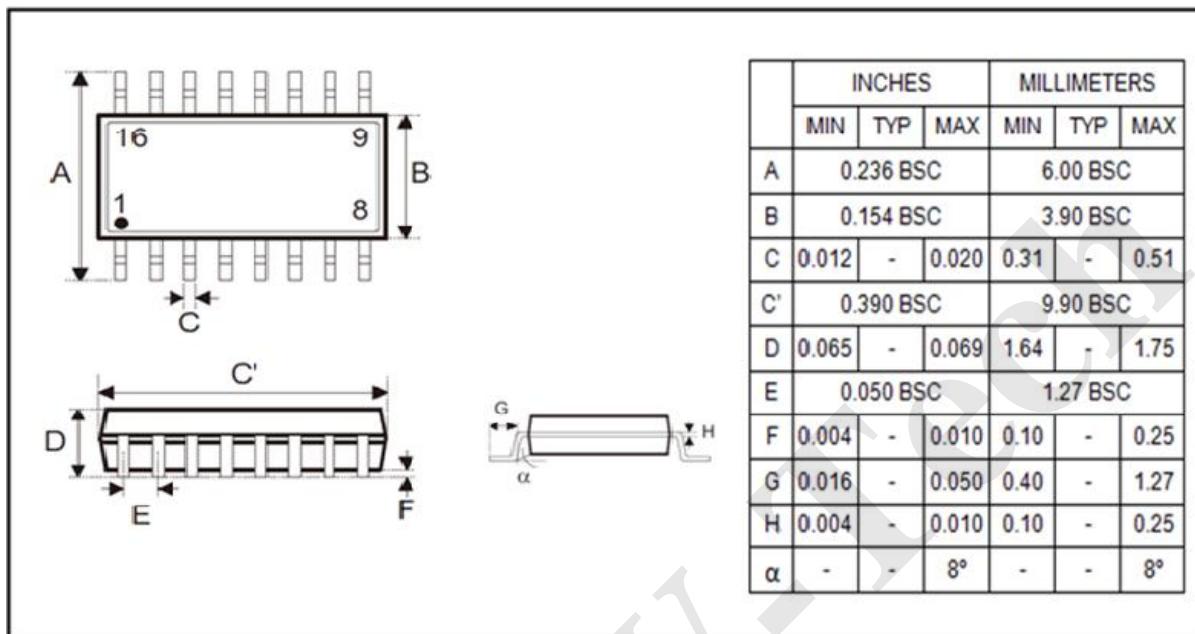
SYMBOL	MILLIMETERS		
	MIN	TYP	MAX
A	-	-	1.75
A1	0.10	-	0.25
A2	1.35	1.45	1.55
A3	0.60	0.65	0.70
b	0.23	-	0.31
C	0.19	-	0.25
D	8.50	8.60	8.70
E	3.80	3.90	4.00
E1	5.80	6.00	6.20
e		0.635 BSC	
L	0.30	-	0.50
h	0.40	-	0.80
θ	0°	-	8°

## 27.3 20 引脚TSSOP



SYMBOL	MILLIMETERS		
	MIN	TYP	MAX
A	-	-	1.20
A1	0.05	-	0.15
A2	0.90	1.00	1.05
b	0.19	-	0.30
C	0.09	-	0.20
D	6.40	6.50	6.60
E	4.30	4.40	4.50
E1	6.25	6.40	6.55
e	0.65 BSC		
L1	1.00REF		
L	0.45	0.60	0.75
θ	0°	-	8°

## 27.4 16 引脚SOP (150 mil)



	INCHES			MILLIMETERS		
	MIN	TYP	MAX	MIN	TYP	MAX
A	0.236 BSC			6.00 BSC		
B	0.154 BSC			3.90 BSC		
C	0.012	-	0.020	0.31	-	0.51
C'	0.390 BSC			9.90 BSC		
D	0.065	-	0.069	1.64	-	1.75
E	0.050 BSC			1.27 BSC		
F	0.004	-	0.010	0.10	-	0.25
G	0.016	-	0.050	0.40	-	1.27
H	0.004	-	0.010	0.10	-	0.25
α	-	-	8°	-	-	8°

## 28. 订购信息

产品名称	封装类型	引脚数	封装尺寸	配送方式
AT8BM84AS16	SOP	16	150 mil	管装: 每管 50 颗。
AT8BM84AT20	TSSOP	20	-	卷装: 每卷 4K 颗。 管装: 每管 70 颗。
AT8BM84AU24	SSOP	24	-	卷装: 每卷 2.5K 颗。 管装: 每管 50 颗。
AT8BM84ANA24	QFN	24	QFN24L (4x4x0.75-P0.5)	卷装: 每卷 4K 颗。