

AT8BM84A

22 I/O 21+2-通道 ADC 8 位 MTP-Based 单片机

Version 1.1

Sept. 3, 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	1. 修改封装脚位名称。 2. 修改图 34 PB2 到 PA4。 3. 修改工作电压描述。 4. 增加数据存储器访问的直接寻址方式的描述。 5. 添加关于如何使用直接寻址模式的示例代码。 6. 修改 6.2 直流电气特性 7. 修改 6.7 推荐工作电压最小电压。 8. 预分频器 0 的频率需小于系统频率(F _{INST})的二分之一。	12~14 76 9 20 20 122,123 128 45

目 录

MOVARDATA SHEET	1
Revision History	1
1. 概述.....	8
1.1 功能	8
1.2 系统框图	11
1.3 引脚图.....	12
1.3.1 QFN24.....	12
1.3.2 SSOP24.....	13
1.3.3 TSSOP20.....	13
1.3.4 SOP16.....	14
1.4 引脚说明	15
2. 内存结构.....	18
2.1 程序存储器.....	18
2.2 数据存储器.....	18
2.3 数据存储器列表.....	20
3. 功能概述.....	22
3.1 Bank 0 特殊功能寄存器.....	22
3.1.1 INDF (间接寻址寄存器)	22
3.1.2 PCL (程序计数器低字节)	22
3.1.3 STATUS (状态寄存器)	22
3.1.4 FSR (数据指针寄存器)	23
3.1.5 OSCCR (振荡器控制寄存器)	23
3.1.6 PCHBUF (程序计数器高字节)	24
3.1.7 INTCON (中断控制寄存器)	24
3.1.8 PIR1 (中断标志寄存器)	24
3.1.9 PIR2 (中断标志寄存器 2)	25
3.1.10 SELCON (选择控制寄存器)	26
3.2 Bank 1 特殊功能寄存器.....	27
3.2.1 PIE1 (中断使能寄存器 1)	27
3.2.2 PIE2 (中断使能寄存器 2)	28
3.2.3 PCON (Power 寄存器)	28
3.2.4 INTEDG (外部中断控制寄存器)	29
3.2.5 TBHP (表格指针高字节寄存器)	29
3.2.6 TBHD (表格数据高字节寄存器)	29

3.3	Bank 2 特殊功能寄存器.....	30
3.3.1	DLL (波特率除法锁存LSB寄存器)	30
3.3.2	DLH (波特率除法锁存MSB寄存器)	30
3.3.3	LCR (行控制寄存器)	30
3.3.4	LSR (行状态寄存器)	31
3.4	I/O Port.....	32
3.4.1	IOSTA (PortA I/O 控制寄存器)	33
3.4.2	IOSTB (PortB I/O 控制寄存器)	34
3.4.3	IOSTC (PortC I/O 控制寄存器)	34
3.4.4	AWUCON (PortA唤醒控制寄存器)	34
3.4.5	BCDWUCON (PortB/C/D 唤醒控制寄存器)	34
3.4.6	PORTACON30 / PORTACON74 / PORTBCON30 / PORTBCON74 / PORTCCON30 / PORTCCON74 (端口属性控制寄存器)	35
3.4.7	PWMxCON (PWM控制寄存器)	36
3.4.8	PortA (PortA数据寄存器)	37
3.4.9	PortB (PortB数据寄存器)	37
3.4.10	PortC (PortC数据寄存器)	37
3.4.11	PxCON (端口模拟引脚控制寄存器)	38
3.4.12	IO引脚结构框图.....	39
3.5	定时器 0.....	44
3.5.1	T0MD (定时器 0 控制寄存器)	45
3.5.2	TMR0 (定时器 0 寄存器)	46
3.6	定时器 1 / 定时器 4 / 定时器 5.....	46
3.6.1	TMRxL (定时器低字节寄存器)	47
3.6.2	TMRxH (定时器高字节寄存器)	47
3.6.2	TxCR1 (定时器控制寄存器)	47
3.6.3	TxCR2 (定时器控制寄存器 2)	48
3.7	PWM 和 Buzzer.....	50
3.7.1	PWM1 / Buzzer1	50
3.7.2	BZ1CR (Buzzer1 控制寄存器)	51
3.7.3	PWM1DUTY (PWM 1 占空比寄存器)	52
3.7.4	PWM2.....	52
3.7.5	PWM2DUTY (PWM 2 占空比寄存器)	53
3.7.6	PWM3.....	53
3.7.7	PWM3DUTY (PWM 3 占空比寄存器)	54
3.7.8	PWM4.....	54
3.7.9	PWM4DUTY (PWM 4 占空比寄存器)	55
3.7.10	PWM5.....	55
3.7.11	PWM5DUTY (PWM5 占空比寄存器)	56
3.7.12	PWM周期.....	56

3.7.13	PWM占空比.....	56
3.7.14	PWMDBx.....	57
3.8	CCP模式.....	57
3.8.1	模块配置.....	58
3.8.2	CCP I/O 配置.....	58
3.8.3	CCPxCON	58
3.8.4	捕捉模式 (仅在CCP1 中可用)	60
3.8.5	比较模式 (仅在CCP1 中可用)	60
3.8.6	CMPCON (比较器控制寄存器)	61
3.8.7	CMPCR (比较器电压选择控制寄存器)	61
3.8.8	增强型PWM模式.....	62
3.8.9	PWM周期.....	63
3.8.10	PWM占空比.....	63
3.8.11	PWM操作设置.....	64
3.8.12	增强型PWM特性.....	64
3.8.13	半桥模式 (可用于CCP1 和CCP2)	65
3.8.14	全桥模式 (可用于CCP1)	66
3.8.15	全桥模式下的方向改变.....	67
3.8.16	启动注意事项.....	69
3.8.17	电源控制模式下的操作.....	69
3.8.18	复位影响.....	69
3.9	RFC模式.....	69
3.9.1	RFC (RFC 寄存器)	69
3.10	IR (红外) 载波.....	70
3.10.1	IRCR (红外控制寄存器)	71
3.11	低电压检测 (LVD)	71
3.11.1	LVDCON (LVD控制寄存器)	72
3.12	电压比较.....	75
3.13	比较器参考电压 (Vref)	75
3.14	模拟-数字转换器 (ADC)	77
3.14.1	ADC参考电压.....	78
3.14.2	ADC模拟输入通道.....	79
3.14.3	ADC时钟 (ADCLK) 、采样时钟 (SHCLK) 和位数选择.....	80
3.14.4	ADC操作过程.....	81
3.14.5	ADMD (ADC模式寄存器)	81
3.14.6	ADDL (ADC LSB输出寄存器)	82
3.14.7	ADDH (ADC输出数据寄存器)	82
3.14.8	ADCON1 (ADC控制寄存器 1)	82
3.14.9	ADJMD (ADC补偿寄存器)	84

3.14.10	ADCR (ADC采样时间与ADC位数寄存器)	84
3.15	看门狗定时器 (WDT)	84
3.16	中断	85
3.16.1	Timer0 上溢中断	85
3.16.2	Timer1 下溢中断	85
3.16.3	Timer4 下溢中断	85
3.16.4	Timer5 下溢 / CCP 中断	85
3.16.5	WDT 超时中断	86
3.16.6	PA/PB/PC 输入状态改变中断	86
3.16.7	外部中断 0	86
3.16.8	外部中断 1	86
3.16.9	外部中断 2	86
3.16.10	LVD 中断	86
3.16.11	比较器输出状态改变中断	86
3.16.12	ADC 转换结束中断	86
3.16.13	串行接口模式中断	86
3.17	振荡器配置	87
3.18	工作模式	89
3.18.1	正常模式	90
3.18.2	慢速模式	90
3.18.3	待机模式	90
3.18.4	睡眠模式	91
3.18.5	唤醒稳定时间	91
3.18.6	工作模式概述	91
3.18.7	复位	92
3.19	SPI 模式	93
3.19.1	SIMDR (串行接口模式数据寄存器)	94
3.19.2	SIMCR (串行接口模式控制寄存器)	94
3.19.3	SPCR (SPI 控制和状态寄存器)	95
3.19.4	串行时钟极性和相位	96
3.19.5	SPI 错误条件	97
3.20	IIC 模式	98
3.20.1	IIC 模式协议	99
3.20.2	IIC 模式操作	99
3.20.3	MADR (IIC 模式地址寄存器)	100
3.20.4	MFDR (IIC 模式频率寄存器)	100
3.20.5	MCR (IIC 模式控制寄存器)	101
3.20.6	MSR (IIC 模式状态寄存器)	101
3.20.7	仲裁机制	102

3.21 通用异步收发器 (UART)	103
3.21.1 THR/RBR (发送保持寄存器/接收缓冲寄存器)	103
3.22 LCD	103
3.22.1 LCDCON0 (LCD 控制寄存器 0)	104
3.22.2 LCDCON1 (LCD 控制寄存器 1)	105
3.23 片上仿真 (OCD)	105
3.23.1 功能描述	105
3.23.2 OCD 限制	105
3.24 系统内程序 (ISP)	106
3.24.1 功能描述	106
3.24.2 ISP 限制	106
3.24.3 ISP 流程	106
4. 指令设置	107
5. 配置字节表	120
6. 电气特性	122
6.1 最大绝对值	122
6.2 直流电气特性	122
6.3 比较器/ LVD 电气特性	124
6.4 OSC 电气特性	124
6.5 ADC 电气特性	124
6.6 特性曲线图	125
6.6.1 I _{HRC} 的频率与 VDD 曲线图	125
6.6.2 I _{HRC} 的频率与温度曲线图	125
6.6.3 I _{LRC} 的频率与 VDD 曲线图	126
6.6.4 I _{LRC} 的频率与温度曲线图	126
6.6.5 上拉电阻与 VDD 曲线图	127
6.6.6 下拉电阻与 VDD 曲线图	127
6.7 建议工作电压	128
7. 封装尺寸	129
7.1 24 引脚 QFN	129
7.2 24 引脚 SSOP	129
7.3 20 引脚 TSSOP	130
7.4 16 引脚 SOP (150 毫寸)	130
8. 订购信息	131

1. 概述

AT8BM84A 是以MTP作为内存的 8 位微控制器，专为家电或量测等等的I/O应用设计。采用CMOS制程并同时提供客户低成本、高性能、及高抗电磁干扰等显著优势。AT8BM84A 核心建立在RISC精简指令集架构可以很容易地做编辑和控制，共有 40 (强化版) 条指令。除了少数指令需要 2 个时序，大多数指令都是 1 个时序即能完成，可以让使用者轻松地以程控完成不同的应用。因此非常适合各种中低记忆容量但又复杂的应用。

AT8BM84A 内建高精度二十一加二信道十二位模拟数字转换器，与高精度电压比较器，足以应付各种模拟界面的侦测与量测。

在I/O的资源方面，AT8BM84A 有 22 根弹性的双向I/O脚，每个I/O脚都有单独的缓存器控制为输入或输出脚。而且每一个I/O脚位都有附加的程控功能如上拉或下拉电阻或开漏极(Open-Drain) 输出。除此之外，AT8BM84A有 4 根I/O脚可提供较大输出电流以应用于LED显示的驱动，并且有 8 根I/O脚可使用程序输出VDD、1/2 VDD及GND的电压，以做为LCD COM端口。AT8BM84A也内建了三种串接界面(UART、IIC及SPI)，与周边传输数据更加方便。

AT8BM84A 有三组定时器，可用系统频率当作一般的计时的应用或者从外部讯号触发来计数。另外AT8BM84A 提供 5 组 10 位分辨率的PWM输出，1 组蜂鸣器输出可用来驱动马达、LED、或蜂鸣器等等。AT8BM84A也设置有CCP模块，可实现捕捉、比较与进阶PWM，进阶PWM可实现全桥与具备死区互补的半桥PWM输出。

AT8BM84A 采用双时钟机制，高速振荡或者低速振荡都可以分别选择内部RC振荡或外部Crystal输入。在双时钟机制下，AT8BM84A 可选择多种工作模式如正常模式(Normal)、慢速模式(Slow mode)、待机模式(Standby mode) 与睡眠模式(Halt mode)可节省电力消耗延长电池寿命。并且微控制器在使用内部RC高速振荡时，低速振荡可以同时使用外部精准的Crystal计时。可以维持高速处理同时又能精准计算真实时间。

在省电的模式下如待机模式(Standby mode) 与睡眠模式(Halt mode)中，有多种事件可以触发中断唤醒AT8BM84A 进入正常操作模式(Normal) 或 慢速模式(Slow mode) 来处理突发事件。

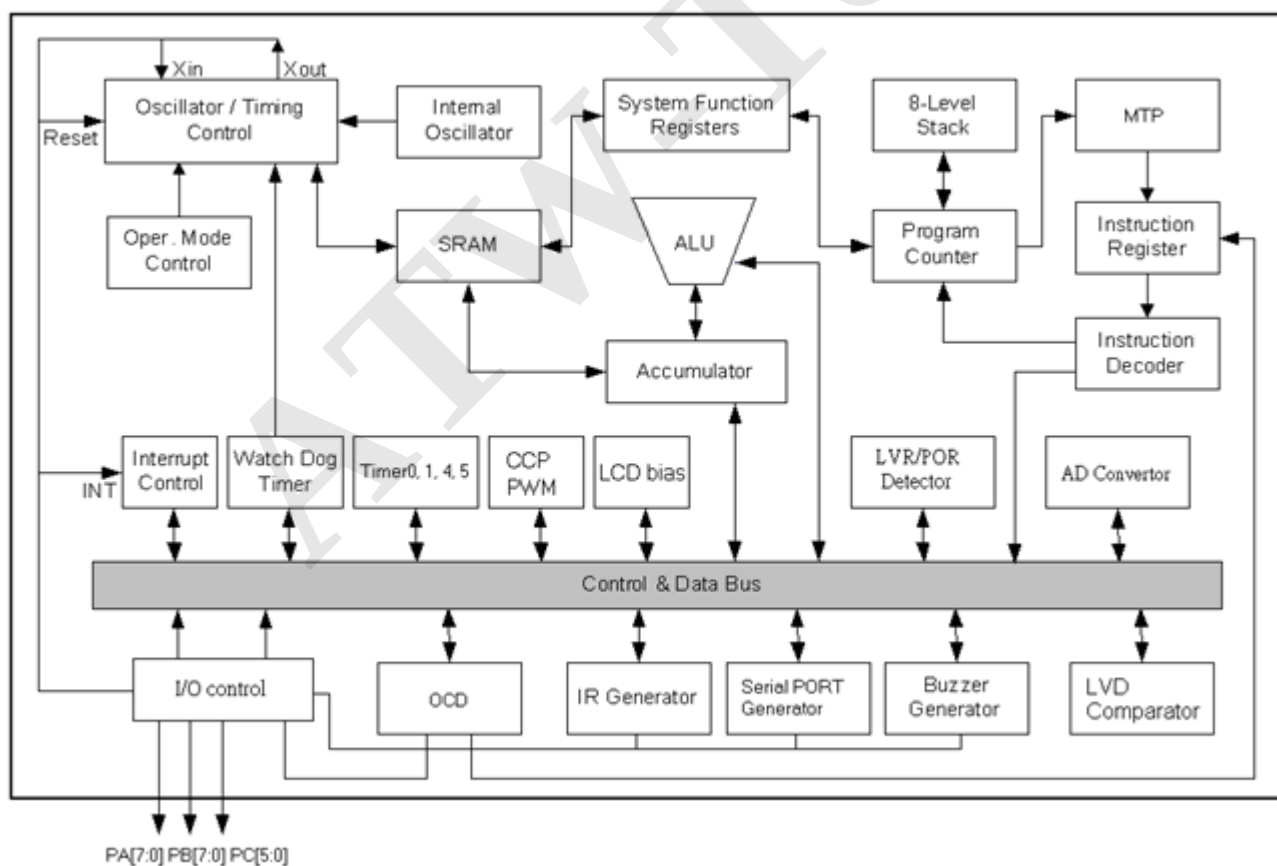
1.1 功能

- 工作电压：
 - 3.0V ~ 5.5V @24MHz/2T。
 - 1.6V ~ 5.5V @4MHz/4T。
- 宽广的工作温度：-40° C ~ 85° C。
- 噪声过滤功能(Noise Filter)
- 4Kx16 bits MTP。
- 336 bytes SRAM。
- 22 根可分别单独控制输入输出方向的I/O脚(GPIO) - PA[7:0]、PB[7:0] 、PC[5:0]。
- PA[7:0]、PB[7:0] 及 PC[5:0]可选择输入时使用内建下拉电阻。
- PA[7:0]、PB[7:0] 及 PC[5:0]可选择输入时使用内建上拉电阻。
- PA[7:0]、PB[7:0] 及 PC[5:0]可选择开漏极输出(Open-Drain)。
- 所有I/O脚输出可选择一般灌电流(Normal Sink Current)、大灌电流(Large Sink Current)或更大灌电流(Ultra Large Sink Current)。所有I/O脚输出可选择一般推电流(Normal Drive Current)或大推电流(Large Drive Current)。
- 四根I/O脚 (PA[3:0])可选择输出更大推电流(Ultra large Drive Current)及超大灌电流(Super Large Sink Current)。

- 8 层程序堆栈(Stack)。
- 存取数据有直接或间接寻址模式。
- 一组 8 位上数定时器(Timer0)包含可程序化的频率预除线路。
- 三组 10 位下数定时器(Timer1, 4, 5)可选重复载入或连续下数计时。
- 五个 10 位脉冲宽度调变(PWM1, 2, 3, 4, 5)。
- 一组全功能CCP(Compare/Capture/PWM(HB,FB))。另一组CCP仅含HB PWM。HB PWM皆有死区互补功能。
- 一个蜂鸣器输出(BZ1)。
- 38/57KHz红外线载波频率可供选择, 同时载波之极性也可以根据数据作选择。
- 内建三合一串接界面(UART/IIC/SPI)。
- 内建准确的低电压侦测电路(LVD)。
- 内建二十一加二信道 12 位模拟数字转换器(Analog to Digital Converter)。
- 内建准确的电压比较器(Voltage Comparator)。
- 内建上电复位电路(POR)。
- 内建低压复位功能(LVR)。
- 内建看门狗计时(WDT), 可由程序软体控制开关。
- 内建LCD功能, 8 个IO脚可输出 1/2 VDD电压。
- 内建电阻频率转换器(RFC)功能。
- 双时钟机制, 系统可以随时切换高速振荡或者低速振荡。
 - 高速振荡: E_HXT (超过 6MHz外部高速石英振荡)
 - E_XT (455K~6MHz外部石英振荡)
 - I_HRC (1~24MHz内部高速RC振荡)
 - 低速振荡: E_LXT (32KHz外部低速石英振荡)
 - I_LRC (内部 32KHz低速RC振荡)
- 四种工作模式可随系统需求调整电流消耗: 正常模式(Normal)、慢速模式(Slow mode)、待机模式(Standby mode) 与 睡眠模式(Halt mode)。
- 十四种硬件中断:
 - Timer0 溢位中断。
 - Timer1 借位中断。
 - Timer4 借位中断。
 - Timer5 借位中断/CCP 中断。
 - WDT 中断。
 - PA/PB/PC 输入状态改变中断。
 - 三组外部中断输入(INT0/INT1/INT2)。
 - 低电压侦测中断。
 - 比较器输出转态中断。
 - 模拟数字转换完成中断。
 - 串接接口模块SIM(IIC&SPI)中断。
 - UART 接口模块读或写中断。
- AT8BM84A在待机模式(Standby mode)下的十三种唤醒中断:
 - Timer0 溢位中断。
 - Timer1 借位中断。

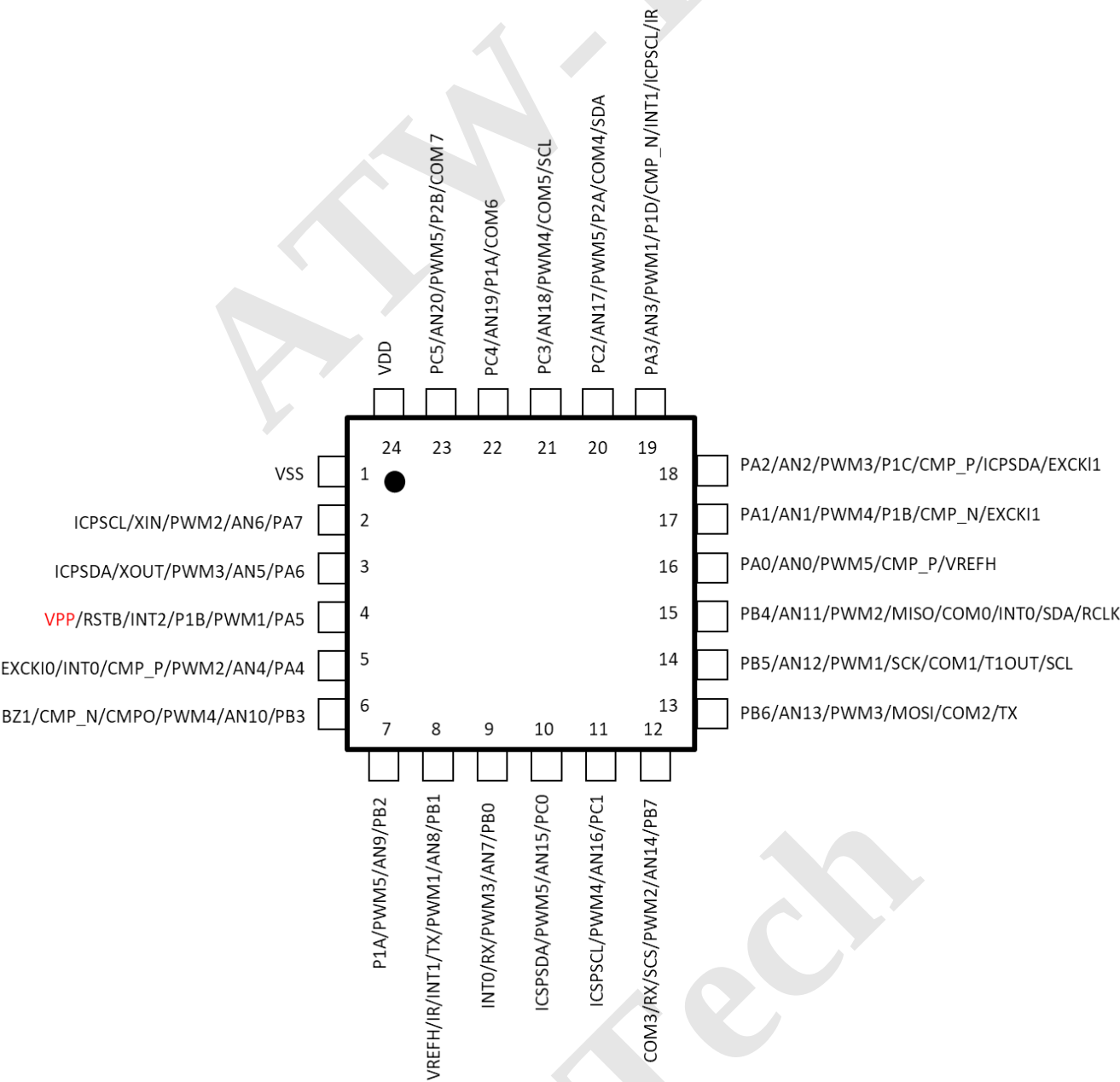
- Timer4 借位中断。
- Timer5 借位中断。
- WDT 中断。
- PA/PB/PC 输入状态改变中断。
- 三组外部中断输入(INT0/INT1/INT2)。
- 低电压侦测中断。
- 比较器输出转态中断。
- 模拟数字转换完成中断。
- 串接口模块SIM(IIC&SPI)中断。
- AT8BM84A在睡眠模式(Halt mode)下的五种唤醒中断：
 - WDT 中断。
 - PA/PB/PC 输入状态改变中断。
 - 三组外部中断输入(INT0/INT1/INT2)。

1.2 系统框图

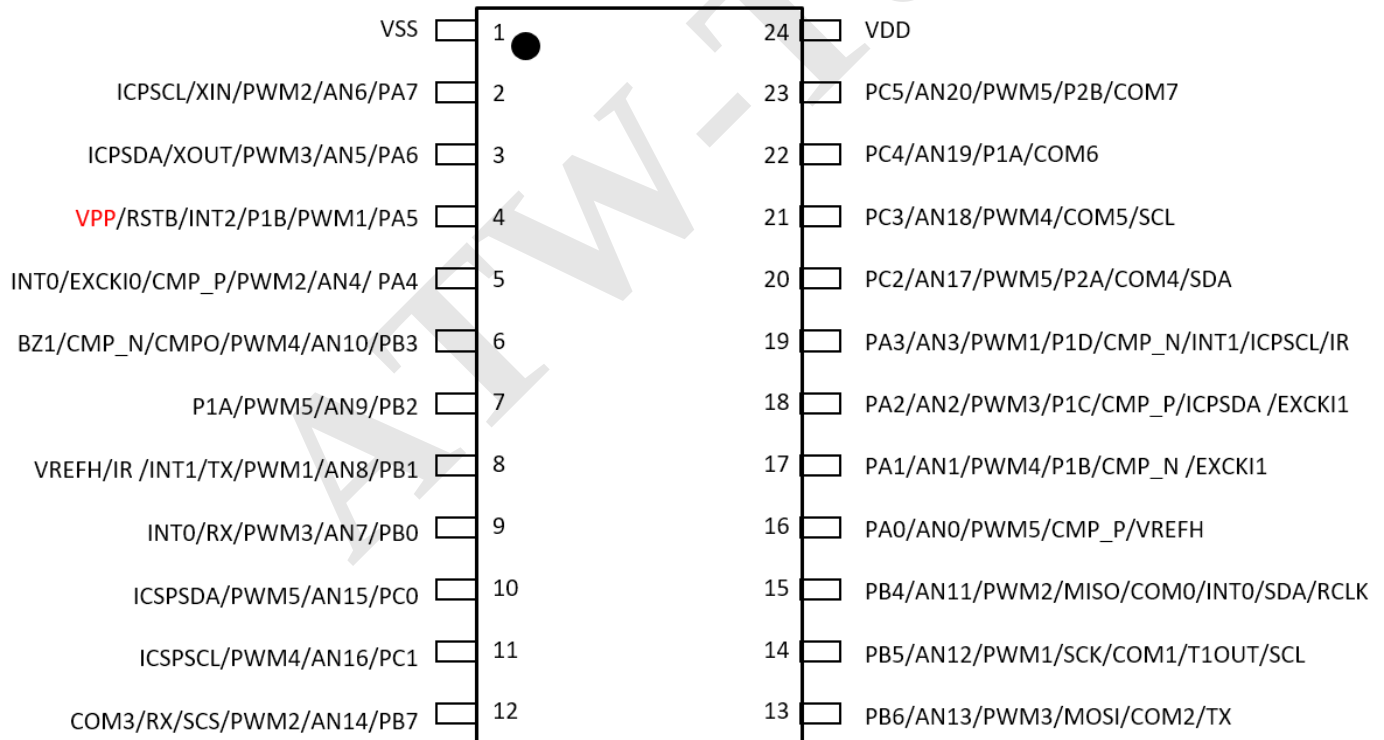


1.3 引脚图

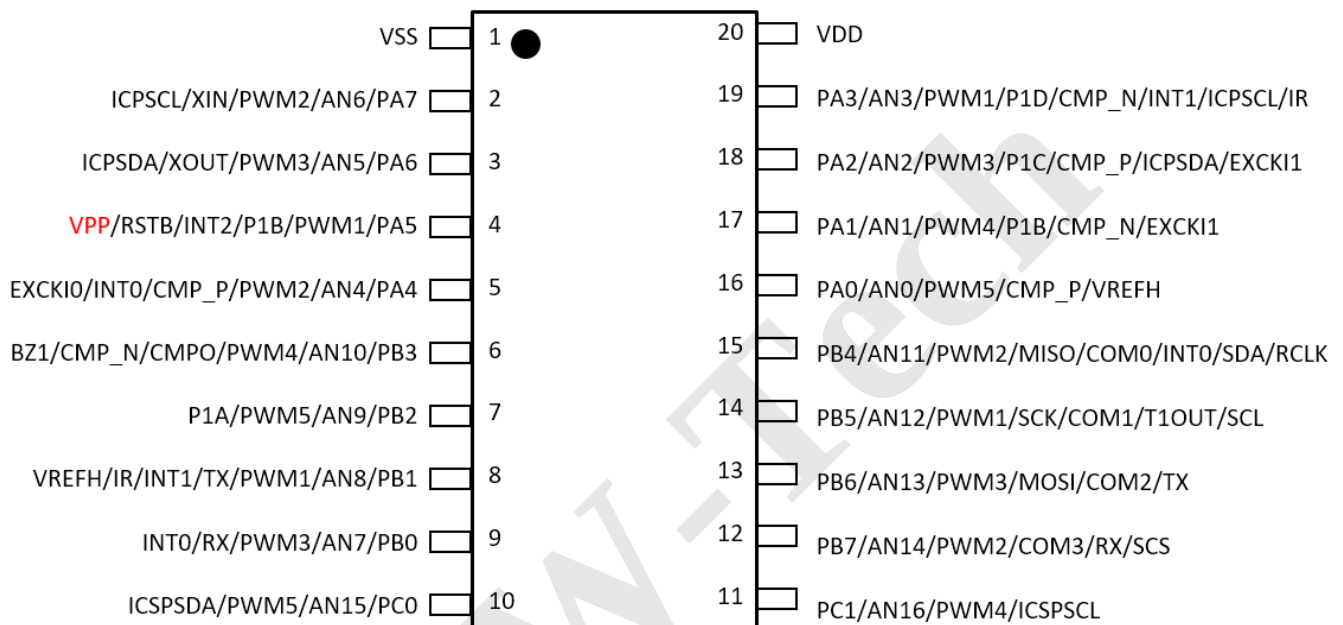
1.3.1 QFN24



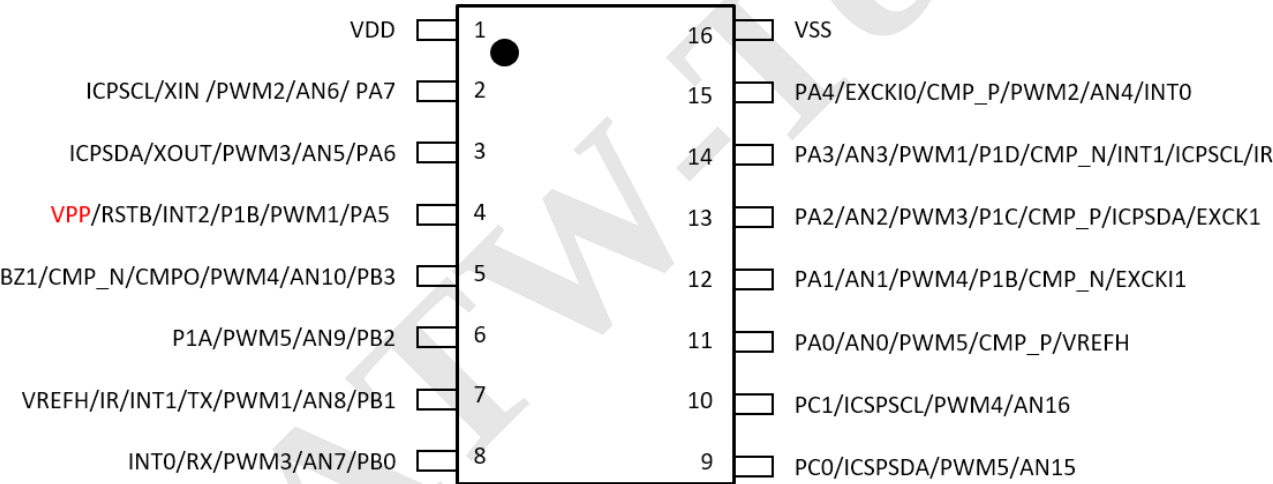
1.3.2 SSOP24



1.3.3 TSSOP20



1.3.4 SOP16



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。 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 SDA 引脚。 EXCKI1 是Timer4/5 的时钟源。
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 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（选项）。（注意 1） PA4 可作为Timer0/1 的时钟源EXCKI0。
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。 PA5 可作为外部中断输入引脚INT2。 PA5 可作为复位引脚RSTB。 PA5 可作为VPP引脚，VPP是编程（烧录）引脚，它也使集成电路进入MTP编程模式。
PA6 / AN5 / PWM3 / Xout / ICPSDA	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。 ICPSDA 是ICP SDA引脚。
PA7 / AN6 / PWM2 / Xin / ICPSCL	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。 ICPSCL是ICP SCL引脚。

引脚名	I/O	描述
PB0 / AN7 / PWM3 / RX / INT0	I/O	PB0 是一个双向I/O引脚。 AN7 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。 AN8 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。
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。(注意 1) 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。 MISO是SPI输出引脚。 INT0 是外部中断引脚。 SDA是IIC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转换。 SCL是IIC时钟输入引脚。 PB5 可输出 LCD 1/2 VDD COM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。 MOSI是SPI输入引脚。 TX是UART TX引脚。 PB6 可输出 LCD 1/2 VDD COM电压。
PB7 / AN14 / PWM2 / SSB / 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。 SSB是SPI使能引脚。 RX是UART RX引脚。 PB7 可输出 LCD 1/2 VDD COM电压。

引脚名	I/O	描述
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/ISP数据引脚。
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/ISP时钟引脚。
PC2 / AN17 / PWM5/ P2A / SDA / COM4		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。（注意 2） P2A是CCP输出引脚。 SDA是IIC SDA引脚。 PC2 可输出 LCD 1/2 VDD COM电压。
PC3 / AN18 / PWM4 / SCL / COM5		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IIC SCL引脚。 PC3 可输出 LCD 1/2 VDD COM电压。
PC4 / AN19 / P1A / COM6		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 COM电压。
PC5 / AN20 / PWM5 / P2B / COM7		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 COM电压。
VDD	-	电源正端。
VSS	-	电源负端。

2. 内存结构

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，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。当执行LGOTO或LCALL时，PCHBUF[3]被加载到PC[11]中进行分页操作。（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。

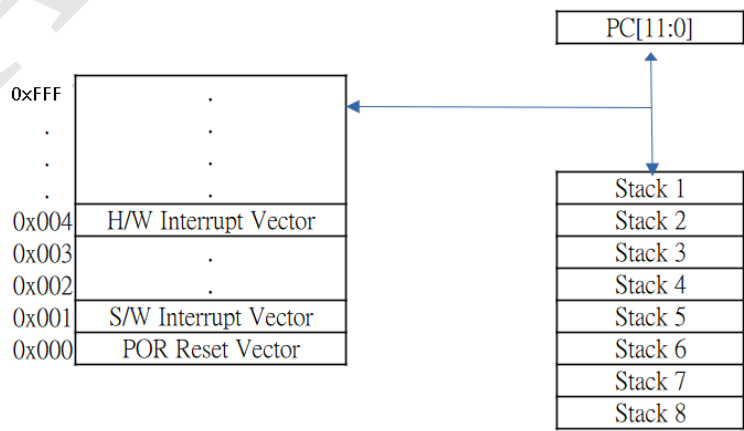


图 2 程序存储器对应地址

2.2 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 个位置未实现，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]。

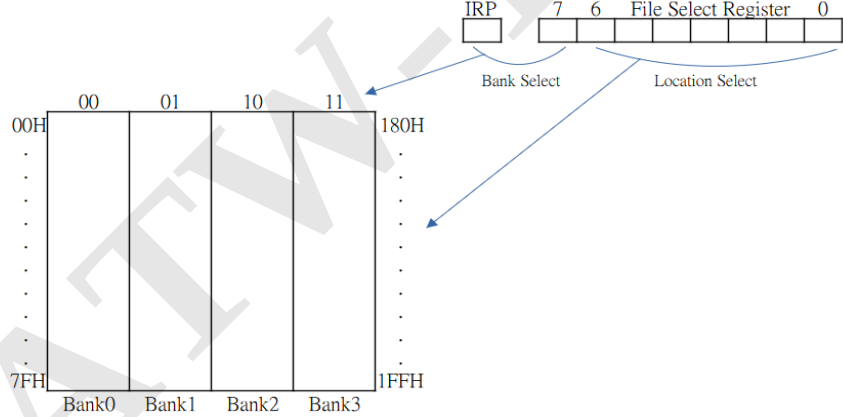


图 3 间接寻址方式存取数据存储器

数据存储器访问的直接寻址模式描述如下。**Bank**选择是由指令操作码[15:14]决定。而位置选择则直接从指令操作码[6:0]开始。

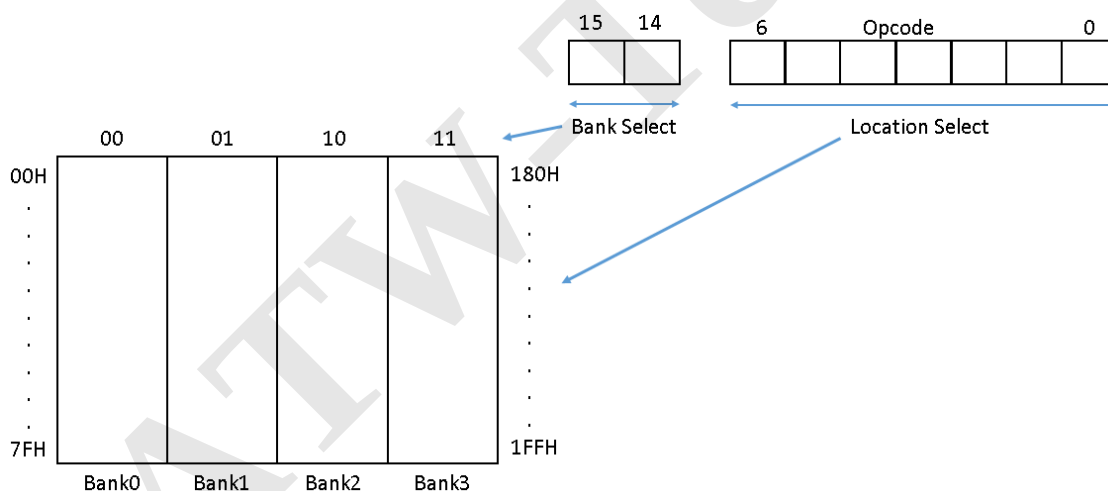


图 4 数据存储器访问的直接寻址方式

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。R页面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实际上会访问Bank0 的地址 0x70~0x7F。

下表描述了AT8BM84A的R-page SFR的寄存器名称和地址映射。

例：如何使用“直接寻址模式”。

1. ASM代码

```
; Access RAM with direct address (Bank3)
movia 0xCC                                ; Write data 0xCC to variable R_RAM_0x1A0
movar  R_RAM_0x1A0                        ; Clear ACC
clra                                       ; Read variable R_RAM_0x1A0 data to ACC
movr   R_RAM_0x1A0,C_SaveToAcc           ; Clear Zero bit of STATUS register
bcr    STATUS,Z_Bit                      ; If ACC=0xCC, then Zero bit=1
xoria  0xCC                               ; If Zero bit of STATUS register = 1, then skip next instruction
btrss  STATUS,Z_Bit                      ; Read date != Write data, set PB0 output low
bcr    PORTB,0                           ; If Zero bit of STATUS register = 0, then skip next instruction
btrsc  STATUS,Z_Bit                      ; Read date = Write data, set PB0 output high
bsr    PORTB,0
```

2. C 代码

```
// Access RAM with direct address (Bank3)
R_RAM_0x1A0 = 0xCC; // Write data 0xCC to variable R_RAM_0x1A0
if(R_RAM_0x1A0 != 0xCC) PORTBbits.PB0 = 0; // Read variable R_RAM_0x1A0 data and check it.
else PORTBbits.PB0 = 1; // Read date = Write data, set PB0 output high
```

2.3 数据存储寄存器列表

Address	Name	Address	Name	Address	Name	Address	Name
00h	INDF	80h	INDF	100h	INDF	180h	INDF
01h	TMR0	81h	T0MD	101h	TMR0	181h	T0MD
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	MADR	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	CMPCR	11Dh	THR/RBR	19Dh	
1Eh		9Eh	TBHP	11Eh	PWMDBx	19Eh	
1Fh	PWMxCON	9Fh	TBHD	11Fh	CCPxCON	19Fh	
20h~6Fh	General Purpose register 80 bytes	A0h~EFh	General Purpose register 80 bytes	120h~16Fh	General Purpose register 80 bytes	1A0h~1EFh	General Purpose register 80 bytes
70h~7Fh	General Purpose register 16 byte (Common)	F0h~FFh	Mapped in bank 0	170h~17Fh	Mapped in bank 0	1F0h~1FFh	Mapped in bank 0

表 1 R-page SFR地址映射

ADDR.	NAME	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value on POR
0h	INDF	INDF[7:0]								xxxx xxxx
1h	TMR0	TMR0[7:0]								xxxx xxxx
2h	PCL	PCL[7:0]								0000 0000
3h	STATUS	IRP	BK1	BK0	/TO	/PD	Z	DC	C	0001 1xxx
4h	FSR	FSR[7:0]								xxxx xxxx
5h	PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	xxxx xxxx
6h	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx xxxx
7h	PORTC			PC5	PC4	PC3	PC2	PC1	PC0	xxxx xxxx
8h										
9h	OSCCR				XTL_BKUP	OPMD[1:0]		STPHOSC	SELHOSC	xxxx 0001
ah	PCBUFH					PCHBUF[4:0]				0000 0000
bh	INTCON	GIE	PEIE	TOIE	INT0IE	PABCIIE	TOIF	INT0IF	PABCIF	0000 0000
ch	PIR1	INT2IF	INT1IF	WDTIF			TSIF/CCPIF	T4IF	T1IF	000x x000
dh	PIR2	ADIF	LVDF	CMPIF		SIMIF	LSRIF	TXIF	RXIF	000x 0010
eh										
fh	TMRxL	TMRx[7:0]								xxxx xxxx
10h	TMRxH							TMRx[9:8]		xxxx xxxx
11h	TxCR1			TmxOE		Tmx_HCK	TxOS	TxRL	TxEN	xxxx xxxx
12h	TxCR2			TxCS	TxCE	/PSxEN	PSxSEL[2:0]			xxxx xxxx
13h	BZ1CR	BZ1EN					BZ1FSEL[3:0]			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)	PC7C1	PC7C0	PC6C1	PC6C0	PC5C1	PC5C0	PC4C1	PC4C0	0000 0000
1ch										
1dh	SELCON	CCPSEL			BCDWUSEL1	BCDWUSEL0	TPSEL2	TPSEL1	TPSEL0	0xx0 0000
1eh										
1fh	PWMxCON							PWMxOEN	PWMxOAL	xxxx xxxx

表 2 Bank 0 SFR bit 映射

ADDR.	NAME	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value on POR
80h	INDF	INDF[7:0]								xxxx xxxx
81h	T0MD	LCKTM0	T0EN	T0CS	T0CE	PS0WDT	PS0SEL[2:0]			0111 1111
82h	PCL	PCL[7:0]								0000 0000
83h	STATUS	IRP	BK1	BK0	/TO	/PD	Z	DC	C	0001 1xxx
84h	FSR	FSR[7:0]								xxxx xxxx
85h	I0STA	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0	1111 1111
86h	I0STB	IOPB7	IOPB6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0	1111 1111
87h	I0STC			IOPC5	IOPC4	IOPC3	IOPC2	IOPC1	IOPC0	xx11 1111
88h										
89h	LVDCON (PCON1)		LVDOUT			LVDS3	LVDS2	LVDS1	LVDS0	xxxx 1111
8ah	PCBUFH	PCHBUF[4:0]								0000 0000
8bh	INTCON	GIE	PEIE	T0IE	INT0IE	PAB0IE	T0IF	INT0IF	PAB0IF	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	IROSC358M					IRCSEL	IRF57K	IREN	0xxx x000
91h	PWM1DUTYL	PWM1DUTY[7:0]								xxxx xxxx
92h	PWM1DUTYH							PWM1DUTY[9:8]		xxxx xxxx
93h	PWM2DUTYL	PWM2DUTY[7:0]								xxxx xxxx
94h	PWM2DUTYH							PWM2DUTY[9:8]		xxxx xxxx
95h	PWM3DUTYL	PWM3DUTY[7:0]								xxxx xxxx
96h	PWM3DUTYH							PWM3DUTY[9:8]		xxxx xxxx
97h	PWM4DUTYL	PWM4DUTY[7:0]								xxxx xxxx
98h	PWM4DUTYH							PWM4DUTY[9:8]		xxxx xxxx
99h	PWM5DUTYL	PWM5DUTY[7:0]								xxxx xxxx
9ah	PWM5DUTYH							PWM5DUTY[9:8]		xxxx xxxx
9bh	INTEDG	INT2EDGE	EIS2	EIS1	EIS0	INT1G1	INT1G0	INT0G1	INT0G0	0000 0101
9ch	CMPCON	CM PEN	BIASEN					CMPOUT	CMPOE	00xx xxx0
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 3 Bank 1 SFR bit 映射

ADDR.	NAME	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value on POR
100h	INDF	INDF[7:0]								xxxx xxxx
101h	TMR0	TMR0[7:0]								xxxx xxxx
102h	PCL	PCL[7:0]								0000 0000
103h	STATUS	IRP	BK1	BK0	/TO	/PD	Z	DC	C	0001 1xxx
104h	FSR	FSR[7:0]								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	PCBUFH						PCHBUF[4:0]			0000 0000
10bh	INTCON	GIE	PEIE	TOIE	INT0IE	PAB0IE	TOIF	INT0IF	PAB0IF	0000 0000
10ch	ADDL	ADDL[7:0]								xxxx xxxx
10dh	ADDH	ADDH[7:0]								xxxx xxxx
10eh	ADCON1	EVHENB		EOC	ADFM	ADCK1	ADCK0	VHS1	VHS0	0xx0 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	RX_PADEN	TX_PADEN	RCLK_PADEN	UREN	0000 0000
114h	SPCR	SPIF	WCOL		MODF	CPOL	CKEG	SPR[1:0]		00x0 0000
115h	MADR	MAD7	MAD6	MAD5	MAD4	MAD3	MAD2	MAD1		0000 000x
116h	MADR				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 4 Bank2 SFR bit 映射

ADDR.	NAME	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value on POR
180h	INDF	INDF[7:0]								xxxx xxxx
181h	T0MD	LCKTM0	T0EN	T0CS	T0CE	PS0WDT	PS0SEL[2:0]			0011 1111
182h	PCL	PCL[7:0]								0000 0000
183h	STATUS	IRP			/TO	/PD	Z	DC	C	0001 1xxx
184h	FSR	FSR[7:0]								xxxx xxxx
185h	I0STA	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0	1111 1111
186h	I0STB	IOPB7	IOPB6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0	1111 1111
187h	I0STC			IOPC5	IOPC4	IOPC3	IOPC2	IOPC1	IOPC0	xxxx xx11
188h										
189h	RFC	RFCEN					PSEL[3:0]			0xxx 0000
18ah	PCBUFH						PCHBUF[4:0]			0000 0000
18bh	INTCON	GIE	PEIE	TOIE	INT0IE	PAB0IE	TOIF	INT0IF	PAB0IF	0000 0000
18ch	LCDCON0	LCDEN						LCDR1	LCDR0	0xxx xx00
18dh	LCDCON1	P7COM	P6COM	P5COM	P4COM	P3COM	P2COM	P1COM	P0COM	0000 0000

表 5 Bank3 SFR bit 映射

3. 功能概述

本章节将详细描述AT8BM84A的操作方式。

3.1 Bank 0 特殊功能寄存器

3.1.1 INDF（间接寻址寄存器）

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							

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。

3.1.2 PCL（程序计数器低字节）

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							

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]。

3.1.3 STATUS（状态寄存器）

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	0x3	IRP	-	-	/TO	/PD	Z	DC	C
读/写属性		读/写	-	-	读/写	读/写	读/写	读/写	读/写

寄存器STATUS包含算术指令的结果和引起复位的原因。

C: 进位/借位标志位。

C=1 时，加法运算有进位或减法运算无借位。

C=0 时，加法运算无进位或减法运算有借位。

DC: 半进位/半借位标志位。

DC=1 时，加法运算低四位有进位或减法运算时没有向高四位借位。

DC=0 时，加法运算低四位无进位或减法运算时有向高四位借位。

Z: 零位。

Z=1 时，逻辑运算的结果是零。

Z=0 时，逻辑运算的结果不为零。

/PD: 睡眠模式标志位。

/PD=1 时，上电或执行CLRWDT指令后。

/PD=0 时，执行SLEEP指令后。

/TO: 看门狗超时标志位。

/TO=1 时，上电或执行CLRWDT或SLEEP指令后。

/TO=0 时，发生WDT上溢。

/IRP: 在间接寻址模式下，采用IRP和FSR[7]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。

3.1.4 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

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

FSR[7]: 在间接寻址模式下，bank选择bit 0。

3.1.5 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		-	-	-	写	读/写	读/写	读/写	
初始值		X	X	X	0	00	0	1	

SELHOSC: 系统振荡器选择（F_{osc}）。

SELHOSC=1 时，F_{osc}是高频振荡器（F_{HOSC}）。

SELHOSC=0 时，F_{osc}是低频振荡器（F_{LOSC}）。

STPHOSC: 关闭/开启高频率振荡器（F_{HOSC}）。

STPHOSC=1 时，F_{HOSC}会停止振荡并被关闭。

STPHOSC=0 时，F_{HOSC}保持振荡。

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

表 6 选择OPMD[1:0]的操作模式

XTL_BKUP: 写 1 来停止晶体 32.768K加速功能，只写。

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。

3.1.6 PCHBUF（程序计数器高字节）

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	0			

PCHBUF[3:0]: PC的第 12~8 位缓存器。

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]。

3.1.7 INTCON（中断控制寄存器）

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCON	0xB	GIE	PEIE	T0IE	INT0IE	PABCIE	T0IF	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

GIE: 全局中断使能位。

GIE=1，打开所有中断。

GIE=0，关闭所有中断。

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。

PEIE=1，打开所有外围中断。

PEIE=0，关闭所有外围中断。

T0IE: Timer0 溢出中断使能位。

T0IE=1，打开TMR0 中断。

T0IE=0，关闭TMR0 中断。

INT0IE: 外部中断 0 使能位。

INT0IE=1，打开EXT0 中断。

INT0IE=0，关闭EXT0 中断。

PABCIE: PortA/PortB/PortC 变化中断使能位。

PABCIE=1，打开 PortA/PortB/PortC 变化中断。

PABCIE=0，关闭 PortA/PortB/PortC 变化中断。

T0IF: Timer0 中断标志位。

T0IF=1，TMR0 溢出。

T0IF=0，TMR0 未溢出。

INT0IF: 外部中断 0 标志位。

INT0IF=1，外部中断 0 发生。

INT0IF=0，外部中断 0 未发生。

PABCIF: PortA/PortB/PortC 变化中断标志位。

PABCIF=1，PortA/PortB/PortC 状态发生改变。

PABCIF=0，PortA/PortB/PortC 状态未发生改变。

3.1.8 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/ CCPIF	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

T1IF: Timer1 溢出中断标志位。

T1IF=1，Timer1 溢出中断发生。

T1IF 必须通过软件清零。

T4IF: Timer4 溢出中断标志位。

T4IF=1, Timer1 溢出中断发生。

T4IF 必须通过软件清零。

T5IF/CCPIF: 当CCP比较或捕捉模式使能时, 这是CCP中断标志, 或者是Timer5 中断标志。

T5IF/CCPIF=1, Timer5 或CCP中断发生。

T5IF/CCPIF必须通过软件清零。

WDTIF: WDT超时中断标志位。

WDTIF=1, WDT超时中断发生。

WDTIF必须通过软件清零。

INT1IF: 外部中断 1 标志位。

INT1IF=1, 外部中断 1 发生。

INT1IF必须通过软件清零。

INT2IF: 外部中断 2 标志位。

INT2IF=1, 外部中断 2 发生。

INT2IF必须通过软件清零。

注意: 当对应的 **INTE** 位未使能时, 读中断标志为 0。

3.1.9 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/写	读/写	R	R
初始值(note*)		0	0	0	X	0	0	1	0

ADIF: ADC转换结束中断标志位。

ADIF=1, ADC转换结束发生。

ADIF必须通过软件清零。

LVDIF: LVD中断标志位。

LVDIF=1, LVD中断发生。

LVDIF必须通过软件清零。

CMPIF: 比较器中断标志位。

CMPIF=1, 比较器中断发生。

CMPIF必须通过软件清零。

SIMIF: 串口中断标志位。

SIMIF=1, 串口中断发生。

SIMIF必须通过软件清零。

LSRIF: LSR中断标志位。

LSRIF=1, 线路状态中断发生。

LSRIF必须通过软件清零。

TXIF: UART发送保持寄存器空标志位。

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) 转移到发送移位寄存器时, 它被设置为逻辑 1。

向THR写入数据将清除此标志。

RXIF: UART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。

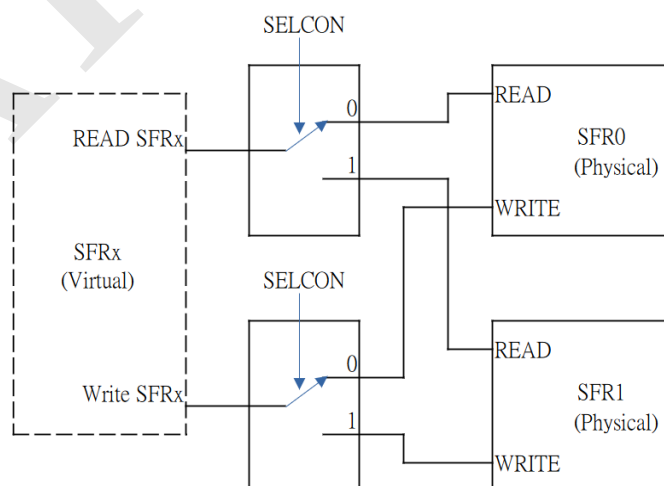
读取RBR数据将清除此标志。

注意: 当对应的 **INTE** 位未使能时, 读中断标志为 0。

3.1.10 SELCON（选择控制寄存器）

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	

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：SFR_x为虚拟寄存器，SFR0 / SFR1 为物理寄存器。当SELCON (CCPSEL, BCDWUSEL或TPSEL) = 0 时，读取或写入SFR_x将实际访问SFR0。当SELCON=1 时，读取或写入SFR_x将实际访问SFR1。也就是说，访问哪个物理寄存器取决于SELCON信号的当前值。



CCPSEL: 虚拟SFR CCP_xCON 和PWMD_{Bx}的映射。

CCPSEL=0, CCP_xCON和PWMD_{Bx}映射到CCP1CON和PWMD_{B1}。(初始值)

CCPSEL=1, CCP_xCON和PWMD_{Bx}映射到CCP2CON和PWMD_{B2}。

BCDWUSEL: 虚拟SFR BCDWU_{CON} 映射。

BCDWUSEL[1:0]=00, BCDWU_{CON}映射到BWU_{CON}。(初始值)

BCDWUSEL[1:0]=01, BCDWU_{CON}映射到CWU_{CON}。

TPSEL: 虚拟 SFR TMR_xL/TMR_xH/TxCR1/TxCR2/PWM_xCON 映射。

TPSEL[2:0]=001, TMR_xL / TMR_xH / TxCR1 / TxCR2 / PWM_xCON 映射到 TMR1L / TMR1H / T1CR1 / T1CR2 / PWM1CON。

TPSEL[2:0]=010, PWM_xCON 映射到 PWM2CON。

TPSEL[2:0]=011, PWM_xCON 映射到 PWM3CON。

TPSEL[2:0]=100, TMR_xL / TMR_xH / TxCR1 / TxCR2 / PWM_xCON 映射到 TMR4L / TMR4H / T4CR1 / T4CR2 / PWM4CON。

TPSEL[2:0]=101, TMR_xL / TMR_xH / TxCR1 / TxCR2 / PWM_xCON 映射到 TMR5L / TMR5H / T5CR1 / T5CR2 / PWM5CON。

虚拟 SFR 与物理 SFR 的映射关系如下表所示：

SELCON	虚拟SFR	SELCON值	物理SFR	注意
CCPSEL	CCP _x CON	0	CCP1CON	
		1	CCP2CON	
	PWMD _{Bx}	0	PWMD _{B1}	
		1	PWMD _{B2}	

SELCON	虚拟SFR	SELCON值	物理SFR	注意
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	

注意：当SELCON被主代码和中断服务程序同时使用时，它可能会导致程序失败或错误的读数。在进入中断服务程序时将SELCON值保存到内存中，并在从中断返回之前恢复它，这是一种很好的做法。

3.2 Bank 1 特殊功能寄存器

3.2.1 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
初始值(note*)		0	0	0	X	X	0	0	0

T1IE: Timer1 溢出中断使能位。

T1IE=1, Timer1 溢出中断开启。

T1IE=0, Timer1 溢出中断关闭。

T4IE: Timer4 溢出中断使能位。

T4IE=1, Timer4 溢出中断开启。

T4IE=0, Timer4 溢出中断关闭。

T5IE/CCPIE: 当CCP比较或捕捉模式使能时，这是CCP中断使能位，或者是Timer5 中断使能位。

T5IE/CCPIE=1, Timer5 或CCP中断开启。

T5IE/CCPIE=0, Timer5 或CCP中断关闭。

WDTIE: WDT超时中断使能位。

WDTIE=1, WDT超时中断开启。

WDTIE=0, WDT超时中断关闭。

INT1IE: 外部中断 1 使能位。

INT1IE=1, 外部中断 1 开启。

INT1IE=0, 外部中断 1 关闭。

INT2IE: 外部中断 2 使能位。

INT2IE=1, 外部中断 2 开启。

INT2IE=0, 外部中断 2 关闭。

注意: 当对应的 **INTE** 位未使能时, 读中断标志为 0。

3.2.2 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(note*)		0	0	0	X	0	0	0	0

ADIE: ADC转换结束中断使能位。

ADIE=1, ADC转换结束中断开启。

ADIE=0, ADC转换结束中断关闭。

LVDIE: LVD中断使能位。

LVDIE=1, LVD中断开启。

LVDIE=0, LVD中断关闭。

CMPIE: 比较器中断使能位。

CMPIE=1, 比较器中断开启。

CMPIE=0, 比较器中断关闭。

SIMIE: 串口中断使能位。

SIMIE=1, 串口中断开启。

SIMIE=0, 串口中断关闭。

LSRIE: LSR (接收线路状态) 中断使能位。

LSRIE=1, LSR中断开启。

LSRIE=0, LSR中断关闭。

TXIE: THR (发送保持寄存器) 为空时, 中断使能位。

TXIE=1, THR为空时, 中断开启。

TXIE=0, THR为空时, 中断关闭。

RXIE: 完全接收 1byte 中断使能位。

RXIE=1, 完全接收 1byte 中断开启。

RXIE=0, 完全接收 1byte 中断关闭。

注意: 当对应的 **INTE** 位未使能时, 读中断标志为 0。

3.2.3 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

LVREN: 开启/关闭 LVR。

LVREN=1时, 开启LVR。

LVREN=0时, 关闭LVR。

LVDEN: 开启/关闭 LVD。

LVDEN=1时, 开启LVD。

LVDEN=0时, 关闭LVD。

WDTEN: 开启/关闭 WDT。

WDTEN=1时, 开启WDT。

WDTEN=0时, 关闭WDT。

3.2.4 INTEDG（外部中断控制寄存器）

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTEDG	0x9B	INT2DEG	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

INT1G1~0: INT1边沿触发选择位。

00: 保留, 01: 上升沿触发, 10: 下降沿触发, 11: 上升/下降沿触发。

INT0G1~0: INT0边沿触发选择位。

00: 保留, 01: 上升沿触发, 10: 下降沿触发, 11: 上升/下降沿触发。

EIS0: 外部中断0引脚选择位。

EIS0=1时, PB0/PB4/PA4选择为外部中断0引脚。

EIS0=0时, PB0/PB4/PA4选择为GPIO。

EIS1: 外部中断1引脚选择位。

EIS1=1时, PB1/PA3选择为外部中断1引脚。

EIS1=0时, PB1/PA3选择为GPIO。

EIS2: 外部中断2引脚选择位。

EIS2=1时, PA5选择为外部中断2引脚。

EIS2=0时, PA5选择为GPIO。

INT2DEG: INT2边沿触发选择位。

0: 下降沿触发, 1: 上升沿触发

3.2.5 TBHP（表格指针高字节寄存器）

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

当指令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。

3.2.6 TBHD（表格数据高字节寄存器）

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	读	读	读	读	读	读	读
初始值		X	X	X	X	X	X	X	X

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。

3.3 Bank 2 特殊功能寄存器

3.3.1 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	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

DLL7~DLL0: 计算波特率除法的低 8 位。

3.3.2 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	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

DLH7~DLH0: 计算波特率除法的高 8 位。

Baud rate = HIRC_Freq / [16x(N)], N={DLH[7:0], DLL[7:0]}

3.3.3 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

WL1~WL0: 字长选择位。

WL[1:0]	字长位数
00	5 位每帧。
01	6 位每帧。
10	7 位每帧。
11	8 位每帧。

STPS: 结束位长度。

STPS	数据位	结束位
0	X	1
1	5	1.5
1	6,7,8	2

PREN: 校验位。

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。

PEVEN: 奇偶位选择。

1: 发送或检查偶数位。

0: 发送或检查奇数位。

PSTUCK: 固定奇偶校验。

1: 当 PEVEN=1 时, 奇偶校验位被发送, 然后被接收器检测为逻辑 00。

0: 当 PEVEN=0 时, 奇偶校验位为逻辑 1。

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。

LOOP: 循环回测试使能。

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: 关闭循环回测试使能。

3.3.4 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	读	读	读	读
初始值		X	1	1	0	0	0	0	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。

OVERR: 超时运行标志。

该位表示在下一个字符被传输到寄存器之前, MCU 没有读取接收器缓冲寄存器中的数据, 从而破坏了前一个字符。

1: 发生超时运行标志。

0: 没有发生超时运行标志。

PERR: 奇偶校验错误。

1: 检测奇偶校验错误。

0: 无奇偶校验错误。

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。

BKINT: 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。

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 写入数据将清除此标志。

TSRE: 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。

3.4 I/O Port

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	-	-

表 10 I/O 端口特性概述

PA、PB 和 PC 的每个 I/O 引脚的电平变化都可能产生中断请求。寄存器 AWUCON[7:0] 和 BCDWUCON[7:0] 将会选择可能产生这种中断的 PA、PB 和 PC 引脚。只要 PA、PB 和 PC 的任意引脚被对应的 AWUCON 和 BCDWUCON 选择，如果被选择的引脚有电平变化，寄存器 PABCIF (INTCON[0]) 将会设置为 1。如果寄存器 PABCIE (INTCON[3]) 和 GIE (INTCON[7]) 同时设定为 1，将发生中断请求并执行中断服务程序。

AT8BM84A 提供 3 个外部中断，当寄存器 EIS0 (INTEDG[4]) 设定为 1 时，PA4、PB0 或 PB5 被当作外部中断 0 的输入引脚。当寄存器 EIS1 (INTEDG[5]) 设定为 1 时，PA3 或 PB1 被当作外部中断 1 的输入引脚。当寄存器 EIS2 (INTEDG[6]) 设定为 1 时，PA5 被当作外部中断 2 的输入引脚。

注意：当 PAD 同时设置为电平变化操作和外部中断时，外部中断优先级更高，PAD 电平变化操作将被禁用。

AT8BM84A 提供红外线 IR 载波生成器。IREN (IRCR[0]) = 1, IR 载波生成。IR_PAD option 用于选择 IR 输出口。如下表 11 所示，

IREN 寄存器	IR_PAD 选项	IR 功能	IR Pad
0	X	OFF	-
1	1	ON	PA3
1	0	ON	PB1

表 11 IR 载波选择

由配置字节决定 PA5 可否当作外部复位输入脚 RSTb。当 PA5 为低电平时将导致 AT8BM84A 发生复位。

当配置字节设置外部晶振 (E_HXT, E_XT 或 E_LXT) 用于高速振荡时钟或低速振荡时钟时，PA7 作为晶振输入引脚 (Xin)，PA6 作为晶振输出引脚 (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，如果使能配置字节，用户可以在 PA7 输出指令时钟 F_{INST}。

如果寄存器 T0MD T0CS 为 1 且 LCK_TM0 为 0，PA4 可当作定时器 0 外部时钟源 EX_CKIO。如果寄存器 T1CS 为 1，

PA4可当作定时器1外部时钟源EX_CK10。如果寄存器T4CS / T5CS为1，PA1或PA2可当作定时器4/5外部时钟源EX_CK11。

如果寄存器 CMPOE=1，PB3可当作比较器输出。如果BZ1EN=1，PB3可当作蜂鸣器输出。如果T1CR1[5] TM1OE为1，PB4可当作T1OUT输出。

AT8BM84A能够以10位分辨率输出5个PWM波形。PWM1、PWM2和PWM3共享相同的时间框架（定时器1），而PWM4和PWM5各有自己的时间框架（定时器4和定时器5）。

注意：PB4 的输出优先级T1OUT > PWM Output。

当I/O配置为输出口，每个引脚的输入电流（sink current）根据配置字节设置为一般灌电流（Normal）（25mA@VDD=5V），大灌电流（large）（40mA@VDD=5V）或极大灌电流（ultra-large）（80mA @VDD =5V），PA3进一步提供超大灌电流（super-large）（120mA@VDD=5V）。

当I/O配置为输出口，每个引脚的输出电流（drive current）根据配置字节设置为一般输出电流（normal）（15mA @VDD =5V）或大电流（large）（30mA @VDD =5V）。PA3~0进一步提供极大输出电流（ultra-large）（45mA @VDD =5V）。

查看下表的Drive/sink电流模式设置：

PXcurrent	PXcsc	Drive mode	Sink mode	Note
0	0	Normal	Normal	
1	0	Large	Large	
1	1	Large	Ultra-Large	
0	1	Ultra-Large	Super-Large	PA3~0 only

表 12 Sink电流模式设置（X=A, B, C）

AT8BM84A提供 3 种串行接口，分别为UART、IIC和SPI，可灵活地与外围设备通信。

对于UART，根据配置字，RX / TX 脚可以是PB0 / PB1 或PB7 / PB6。

对于IIC，根据配置字的不同，SCL / SDA脚可以是PB5 / PB4 或PC3 / PC2。

对于SPI，SSB / MOSI / SCK / MISO脚分别为PB7 / PB6 / PB5 / PB4。

AT8BM84A内置 ½ VDD发生器驱动LCD COM 脚 ½ 偏置LCD面板显示。这些LCD COM脚是PB4，PB5，PB6，PB7，PC2，PC3，PC4 和PC5。

3.4.1 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	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值		1	1	1	1	1	1	1	1

IOPAx: PAx I/O模式选择，0 ≤ x ≤ 7。

IOPAx=1时，PAx设为输入口。

IOPAx=0时，PAx设为输出口。

3.4.2 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	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值		1	1	1	1	1	1	1	1

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

IOPBx=1时, PBx设为输入口。

IOPBx=0时, PBx设为输出口。

3.4.3 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
初始值		X	X	1	1	1	1	1	1

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

IOPCx=1时, PCx设为输入口。

IOPCx=0时, PCx设为输出口。

3.4.4 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	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

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

WUPAx=1时, 开启 PAx 唤醒功能。

WUPAx=0时, 关闭 PAx 唤醒功能。

3.4.5 BCDWUCON (PortB/C/D 唤醒控制寄存器)

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	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

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

WUPBCDx: 开启/关闭 PBx 或 PCy唤醒功能,

BCDWUSEL[1:0]=00: ($0 \leq x \leq 7$)

WUPBCDx=1, 开启 PBx 唤醒功能。

WUPBCDx=0, 关闭 PBx 唤醒功能。

BCDWUSEL[1:0]=01: ($0 \leq x \leq 5$)

WUPBCDx=1, 开启 PCx 唤醒功能。

WUPBCDx=0, 关闭 PCx 唤醒功能。

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

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

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。具体参见下表。

PAXC[1:0]: PORT A 属性设置， $0 \leq x \leq 7$ 。

PAXC	(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]: PORT B 属性设置, $0 \leq x \leq 7$ 。

PBxC	(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]: PORT C 属性设置, $0 \leq x \leq 5$ 。

PCxC	(IOPCx=1) input	(IOPCx=0) output
00	Floating	Push-Pull
01	Pull-down	Push-Pull
10	Pull-High	Open-drain
11	-	Open-drain

3.4.7 PWMxCON (PWM 控制寄存器)

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	X

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

TPSEL[2:0]=001,

PWMxOAL: 定义PWM1 输出有效状态。

PWMxOAL=1, PWM1 输出低有效。

PWMxOAL=0, PWM1 输出高有效。

PWMxOEN: 开启/关闭PWM1 输出。

PWMxOEN=1, PWM1 由PAD输出。

PWMxOEN=0, PAD是一般IO脚。

TPSEL[2:0]=010,

PWMxOAL: 定义PWM2 输出有效状态。

PWMxOAL=1, PWM2 输出低有效。

PWMxOAL=0, PWM2 输出高有效。

PWMxOEN: 开启/关闭PWM2 输出。

PWMxOEN=1, PWM2 由PAD输出。

PWMxOEN=0, PAD是一般IO脚。

TPSEL[2:0]=011,

PWMxOAL: 定义PWM3 输出有效状态。

PWMxOAL=1, PWM3 输出低有效。

PWMxOAL=0, PWM3 输出高有效。

PWMxOEN: 开启/关闭PWM3 输出。

PWMxOEN=1, PWM3 由PAD输出。

PWMxOEN=0, PAD是一般IO脚。

TPSEL[2:0]=100,

PWMxOAL: 定义PWM4 输出有效状态。

PWMxOAL=1, PWM4 输出低有效。

PWMxOAL=0, PWM4 输出高有效。

PWMxOEN: 开启/关闭PWM4 输出。

PWMxOEN=1, PWM4 由PAD输出。

PWMxOEN=0, PAD是一般IO脚。

TPSEL[2:0]=101,

PWMxOAL: 定义PWM5 输出有效状态。

PWMxOAL=1, PWM5 输出低有效。

PWMxOAL=0, PWM5 输出高有效。

PWMxOEN: 开启/关闭PWM5 输出。

PWMxOEN=1, PWM5 由PAD输出。

PWMxOEN=0, PAD是一般IO脚。

3.4.8 PortA (PortA 数据寄存器)

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	数据锁存器值是 xxxxxxxx, 读取值是 xxxxxxxx 端口值(PA7~PA0)								

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, 数据是被写入PortA的输出数据锁存器中。

3.4.9 PortB (PortB 数据寄存器)

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	数据锁存器值是 xxxxxxxx, 读取值是 xxxxxxxx 端口值(PB7~PB0)								

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, 数据是被写入PortB的输出数据锁存器中。

3.4.10 PortC (PortC 数据寄存器)

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		数据锁存器值是 xxxxxx, 读取值是 xxxxxx 端口值(PC5~PC0)							

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, 数据是被写入PortC的输出数据锁存器中。

3.4.11 PxCON (端口模拟引脚控制寄存器)

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
初始值		0	0	0	0	0	0	0	0

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

当 PxSEL=00, PxCON=PACON,

PACON[7:0]: Port A 模拟引脚选择。

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

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

当 PxSEL=01, PxCON=PBCON,

PBCON[7:0]: Port B 模拟引脚选择。

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

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

当 PxSEL=10, PxCON=PCCON,

PCCON[7:0]: Port C 模拟引脚选择。

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

PCCON[i]=0, PCi 可做模拟或数字引脚， $0 \leq i \leq 7$ 。

3.4.12 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 100KΩ 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。

PABCIF: 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。

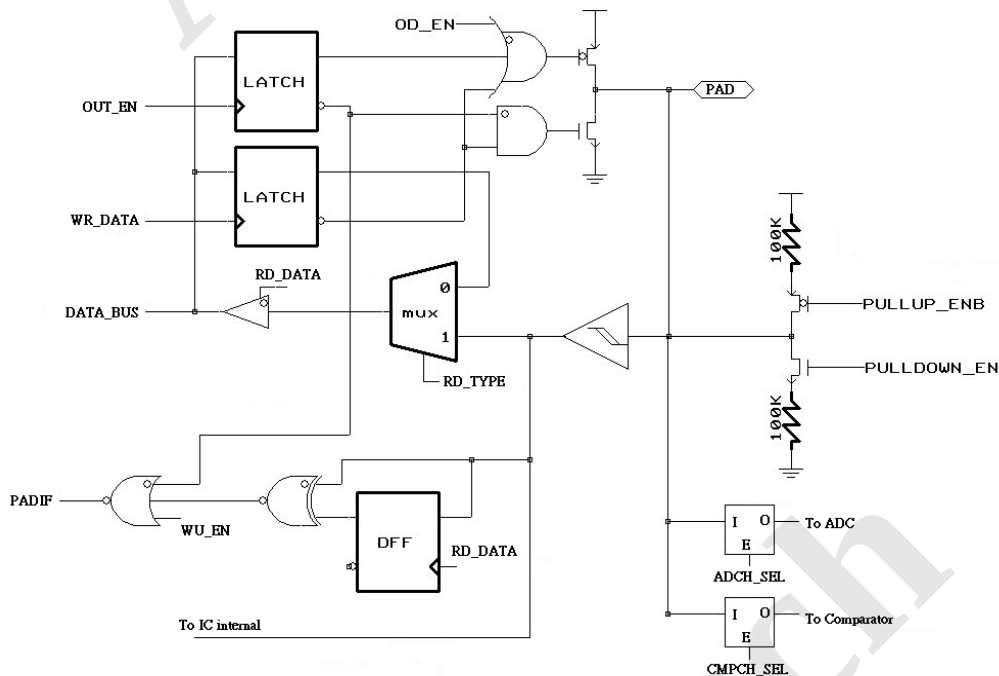


图 5 PA0~PA4, PB3 引脚结构框图

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 100KΩ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。

PABCIF: 引脚变化中断标志。

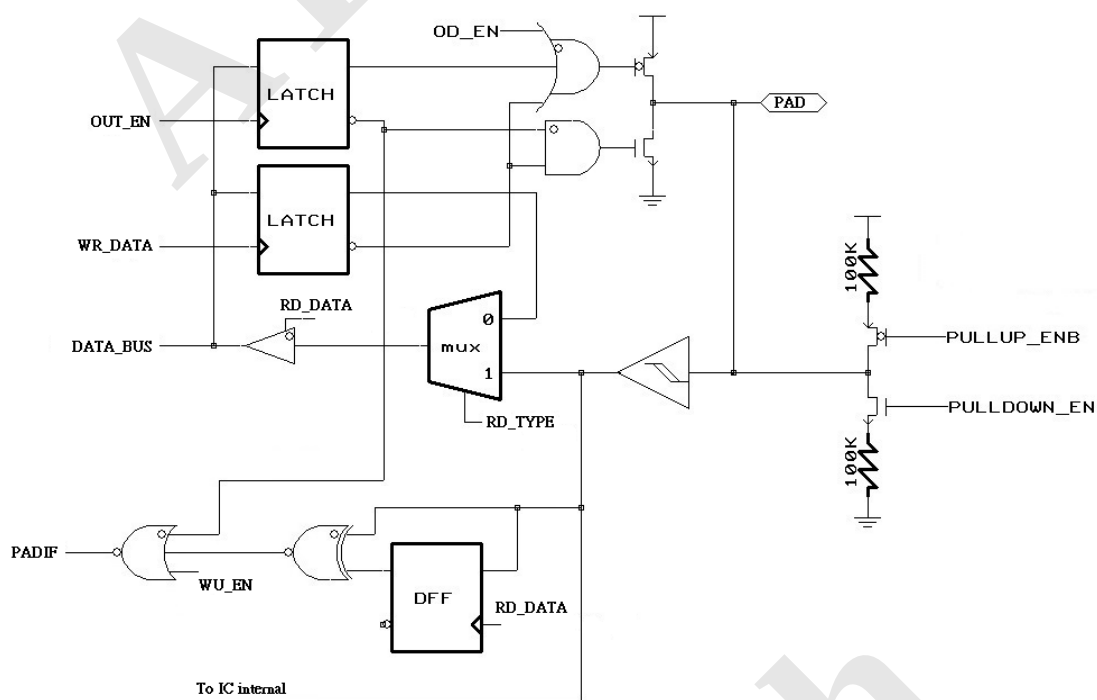


图 6 PA5 引脚结构框图

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 100KΩ 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。

PABCIF: 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。

COM_EN: 开启PAD到 1/2 VDD电压输出。

VDDX0.5: 1/2 VDD电压。

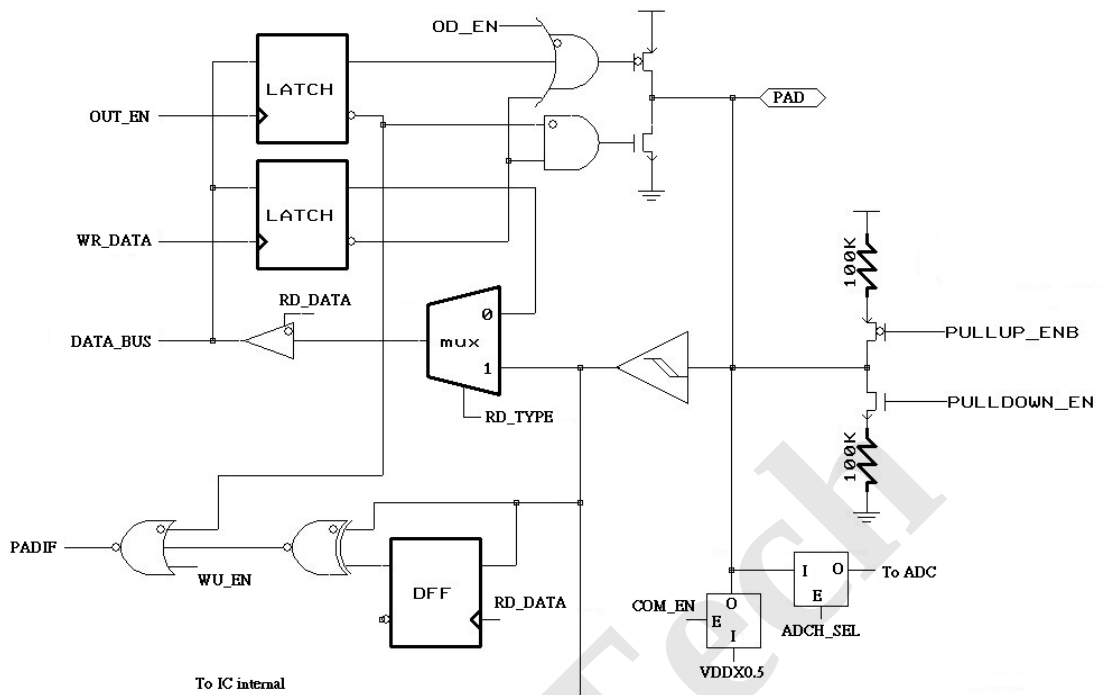


图 8 PB4~PB7, PC2~PC5 引脚结构框图

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 100KΩ 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。

PABCIF: 引脚变化中断标志。

ADCH_SEL: 开启引脚到ADC通道输入。

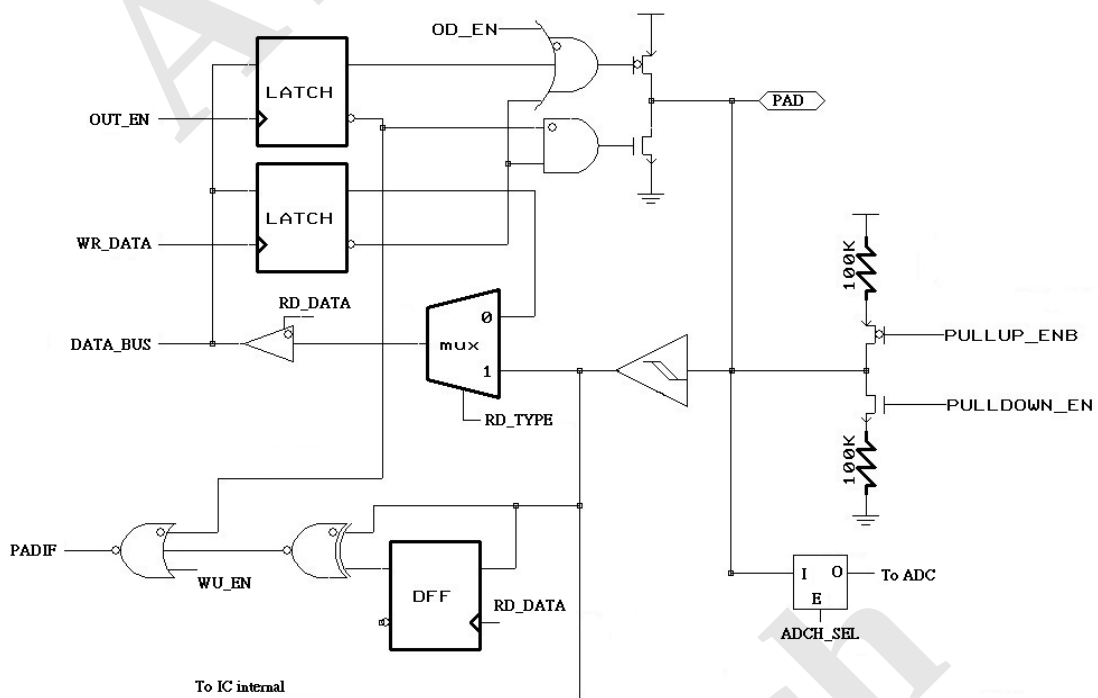


图9 PB0~PB2, PC0~1 引脚结构框图

3.5 定时器 0

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，可以从指令时钟F_{INST}、外部时钟输入引脚EX_CKIO或低频振荡I_LRC / E_LXT中择一。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，取决于配置字节）。汇总成表格如下。（也请参考表13）

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
EXCKIO	1	0	X	X
		X	0	
E_LXT	1	1	1	1
I_LRC	1	1	1	0

表 13 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 / E_LXT的时钟触发沿选择。当T0CE为1，EX_CKIO引脚或I_LRC / E_LXT的上升沿将让定时器0计数加一。当T0CE为0，EX_CKIO引脚或I_LRC / E_LXT的下降沿将让定时器0计数加一。当使用I_LRC / E_LXT作为计时器0的时钟源时，建议使用预分频器0并且比例设置为4分频以上，否则可能会使计数丢失，**预分频器0 的频率需小于系统频率(F_{INST})的二分之一**。

如果寄存器PS0WDT（T0MD[3]）为0，定时器0时钟源可以由预分频器0所分频，预分频器0会被指定到定时器0，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，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TOIE（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：

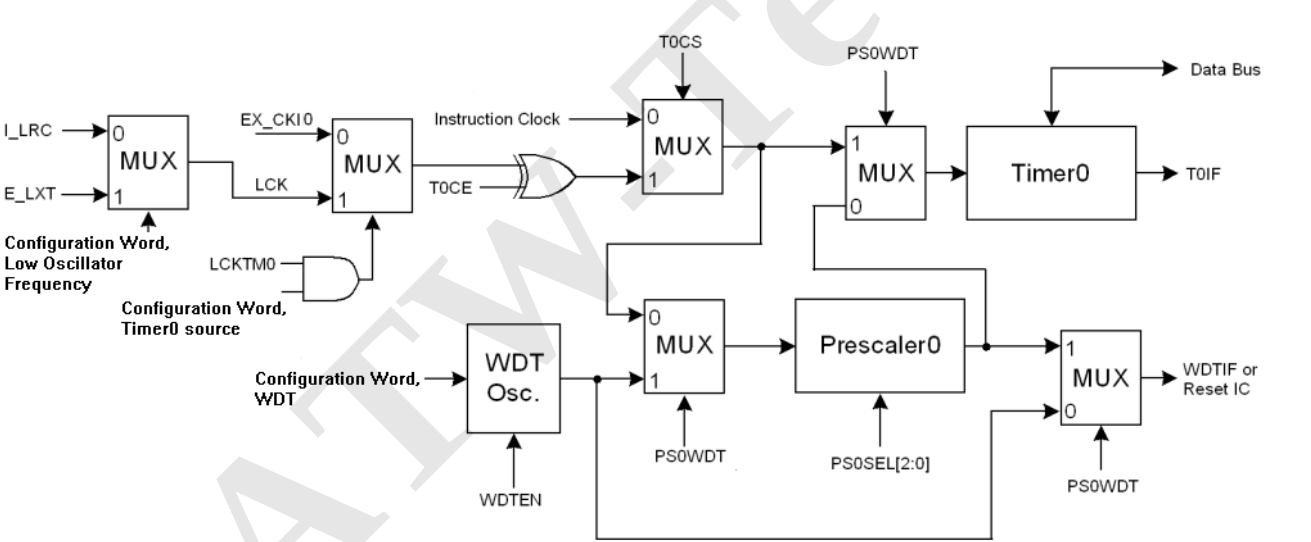


图 10 Timer0 和 WDT结构框图

3.5.1 T0MD（定时器 0 控制寄存器）

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(注意*)		0	1	1	1	1	111		

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 复位)	PS0WDT=1 (WDT 中断)
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

表 8 预分频器 0 的分频比选项

PS0WDT: 预分频器0分配选择。

PS0WDT=1时，预分频器0被分配到WDT。

PS0WDT=0时，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。

T0CE: 定时器0外部时钟源触发沿选择。

T0CE=1时，EX_CKIO发生上升沿信号时定时器0加一。

T0CE=0时，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。

T0CS: 定时器0时钟源选择。

T0CS=1时，选择EX_CKIO引脚或低频振荡I_LRC / E_LXT。

T0CS=0时，选择指令时钟F_{INST}。

T0EN: 开启/关闭定时器0。

T0EN=1时，开启定时器0。

T0EN=0时，关闭定时器0。

LCKTM0: 当T0CS=1，定时器0时钟源可随意选择为低频振荡。

T0CS=0时，指令时钟F_{INST}被选作定时器0时钟源。

T0CS=1时，LCKTM0=0时，外部EX_CKIO引脚被选择当作定时器0时钟源。

T0CS=1时，LCKTM0=1时，低频振荡I_LRC / E_LXT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。

3.5.2 TMR0（定时器 0 寄存器）

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							

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寄存器与配置字节（Configuration Word），定时器0时钟源可以从指令时钟F_{INST}、外部时钟EX_CKIO或低频振荡器I_LRC/E_LXT中选择一个。

3.6 定时器 1 / 定时器 4 / 定时器 5

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。

功能	定时器 1	定时器 4	定时器 5	
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	EXCKI0	EXCKI1	EXCKI1	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	死区控制
CCP2 半桥	否	是	否	死区控制
RFC	是	否	否	

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 将映射到SFR TMR4L / TMR4H / T4CR1 / T4CR2。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M84，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，计时器之间的差异将会指出。

3.6.1 TMRxL（定时器低字节寄存器）

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							

根据TPSEL[2:0]的当前值。访问虚拟SFR TMRxL相当于访问物理SFR TMR1L、TMR4L或TMR5L，根据TPSEL[2:0]的电流值。

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。

3.6.2 TMRxH（定时器高字节寄存器）

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	

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TMRxL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。

3.6.2 TxCR1（定时器控制寄存器）

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

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

该寄存器用于配置定时器 x 功能。（x= 1, 4 或 5）

TxEN: 开启/关闭定时器 x 。

TxEN=1时，开启定时器 x 。

TxEN=0时，关闭定时器 x 。

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。

TxRL=1 时，定时器 x 初始值从重载寄存器TMRx[9:0]重新加载。

TxRL=0 时，当下溢发生时，定时器 x 从 0x3FFF开始连续计数。

TxOS: 当下溢发生时，设置定时器 x 的工作模式。

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

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

TxOS	TxRL	Timerx 下数功能
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。

表 14 定时器 x 功能

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。

TMxOE=1时，TxOUT 输出至PB5。

TMxOE=0时，PB5为GPIO。

TMx_HRC: 定时器 x 时钟源选择。(x= 1, 4或5)

TMx_HRC =1时，PWMx & Timer x 时钟源是内部高频振荡。

TMx_HRC =0时，PWMx & Timer x 时钟源依T1CS寄存器来决定。

注意: 对于 M84，只有定时器 1 具有定时器匹配输出 T1OUT。

注意: 如果 T1OUT和 PWM1 都配置为输出到 PB4，则 T1OUT的输出优先级高于 PWM1。

3.6.3 TxCR2（定时器控制寄存器 2）

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	读/写
初始值		X	X	X	X	X	X	X	X

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

该寄存器用于配置定时器 x 功能。

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

表 15 定时器 x 预分频器的预分频比

注意: 在 /PS1EN=1前须先设定 PS1SEL[2:0]，否则中断可能会发生误触发。

/PSxEN: 关闭/开启定时器 x 预分频器。

/PSxEN=1时，关闭定时器 x 预分频器。

/PSxEN=0时，开启定时器 x 预分频器。

- T1CE:** 定时器 x 外部时钟触发沿选项。
- TxCE=1时, EX_CK1x引脚下降沿时定时器 x 减一。
- TxCE=0时, EX_CK1x引脚上升沿时定时器 x 减一。
- TxCs:** 定时器 x 时钟源选项。
- TxCs=1时, 选择EX_CK1x引脚作为外部时钟输入。
- TxCs=0时, 选择指令时钟F_{INST}。

注意: 对于 M84, 计时器 x 的外部时钟引脚 EXCK1x如下表所示:

Timer	EXCK1x pin	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

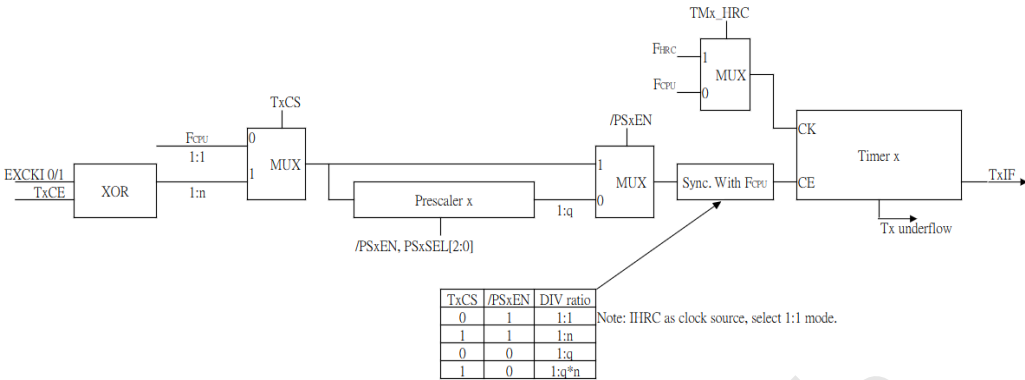


图 11 定时器 1 结构框图

定时器 x 的时钟源可以通过TMx_HRC (TxCR1[3]) 选择。当TMx_HRC=1 时, 定时器 x 时钟源直接I_HRC 输出, 这是最高的定时器时钟源。当TMx_HRC=0 时, 定时器 x 时钟源可以由TxCS (TxCR1[5]) 选择的指令时钟或外部时钟 (EXCKI0 或EXCKI1) 。

定时器 x 时钟源分频比由TxCS和预分频器 x 选择, 如上图所示。当TxCS=0 时, 基准分频比为 1。当TxCS=1 时, 基准分频比为外部时钟 (EXCKI0 或EXCKI1) 周期除以CPU周期 (I_HRC)。当/PSxEN=0 时, 预分频器 x 开启, 由寄存器PS1SEL[2:0] (T1CR2[2:0]) 决定的预分频器 x 分频比为 1:2 ~ 1:256。当/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。

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, 减少定时器 x 的有效沿由寄存器TxCE (TxCR2[4]) 决定。当TxCE为 1 时, EXCK1x上的高到低转换将使定时器 x 递减。当TxCE为 0 时, EXCK1x上的低到高转换将使定时器 x 递减。

定时器 x 的操作可以通过寄存器TxEN (TxCR1[0]) 来开启或关闭。

10 位定时器 x 的MSB 2 位来自TMRxH[1:0], 而LSB 8 位来自TMRxL[7:0]。

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

重新加载寄存器，即TMRxLD[9:8]。当写入寄存器TMRxL时，它将数据写入TMRx[7:0]重新加载寄存器，即TMRxLD[7:0]。

定时器 x 提供两种工作模式：一种是单次模式（One-Shot mode），另一种是连续模式（Non-Stop mode）。当寄存器TxOS（TxCR1[2]）为 1 时，选择单次模式。从寄存器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TxOS（TxCR1[2]）为 0 时，选择连续模式。当发生下溢时，开始下一个下数有两个选择，这是由寄存器TxRL（TxCR1[1]）决定的。当TxRL为 1 时，定时器 x 处于自动加载模式，存储在寄存器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TxRL为 0（连续模式）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（TxOS=0，TxRL=1）采用双缓冲机制：当TxEN=1 时，定时器 x 加载寄存器TMRxLD[9:0]不会上传到定时器 x 计数器（TMRx[9:0]），直到定时器 x 溢出。然而，当TxEN=0 时，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]），无论定时器 x 是在单次模式，自动加载模式还是连续模式。

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。

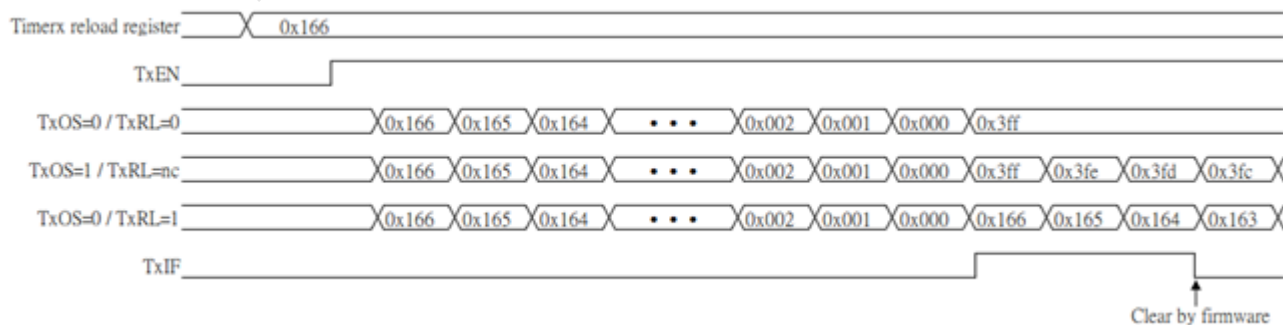


图 12 定时器 x 时序图

3.7 PWM 和 Buzzer

除了CCP PWM，每个定时器都可以通过PWMxCON SFR产生自己的PWM。

3.7.1 PWM1 / Buzzer1

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。

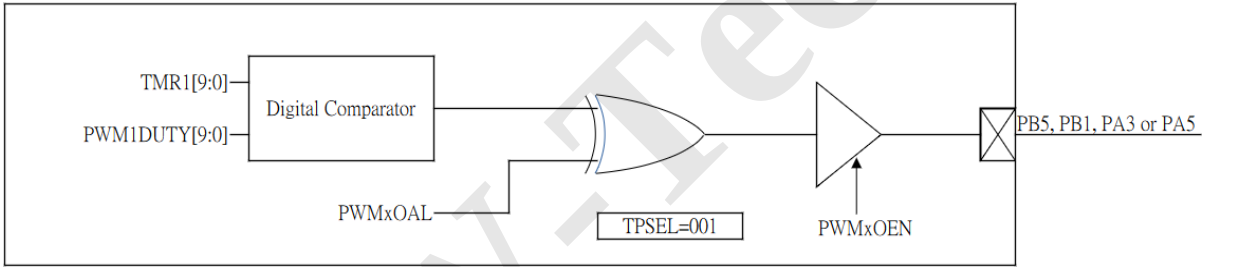


图 13 PWM1 结构框图

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，生成BZ1 输出。当BZ1FSEL[3]为 1 时，选择定时器 1 输出，生成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。

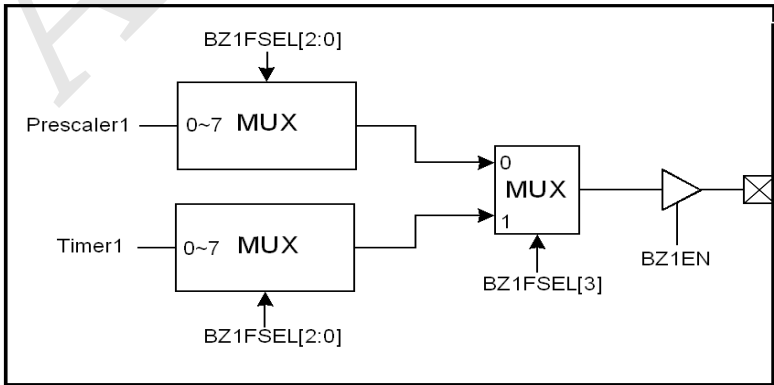


图 14 Buzzer1 结构框图

3.7.2 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

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

BZ1FSEL[3:0]	BZ1 频率选择	
	时钟源	预分频比
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

表 7 蜂鸣器BZ1 输出频率选择

BZ1EN: 开启/关闭蜂鸣器1输出。

BZ1EN=1时，开启蜂鸣器1。

BZ1EN=0时，关闭蜂鸣器1。

3.7.3 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							

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	

PWM1DUTY[9:8]: PWM1 占空比数据 MSB 2 位。

3.7.4 PWM2

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。

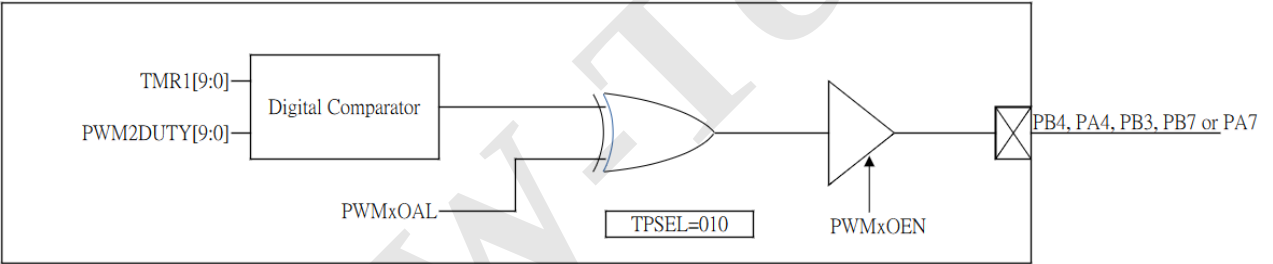


图 15 PWM2 结构框图

3.7.5 PWM2DUTY（PWM 2 占空比寄存器）

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2DUTYL	0x93	PWM2DUTY[7:0]							
读/写属性		写							
初始值		XXXXXXXX							

PWM2DUTY[7:0]: PWM2 占空比数据LSB 8 位。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2DUTYH	0x94	-	-	-	-	-	-	PWM2DUTY[9:8]	
读/写属性		-	-	-	-	-	-	写	
初始值		X	X	X	X	X	X	XX	

PWM2DUTY[9:8]: PWM2 占空比数据MSB 2 位。

3.7.6 PWM3

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。

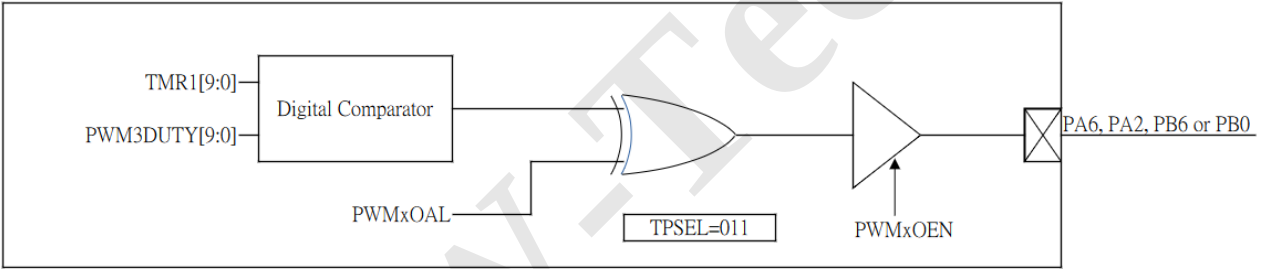


图 16 PWM3 结构框图

3.7.7 PWM3DUTY（PWM 3 占空比寄存器）

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM3DUTYL	0x95	PWM3DUTY[7:0]							
读/写属性		写							
初始值		XXXXXXXX							

PWM3DUTY[7:0]: PWM3 占空比数据LSB 8 位。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM3DUTYH	0x96	-	-	-	-	-	-	PWM3DUTY[9:8]	
读/写属性		-	-	-	-	-	-	写	
初始值		X	X	X	X	X	X	XX	

PWM3DUTY[9:8]: PWM3 占空比数据MSB 2 位。

3.7.8 PWM4

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 周期决定。（对于自动加载模式，定时器 4 周期为 TMR3LD[9:0]）PWM4DUTY 值必须小于或等于定时器 4 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。

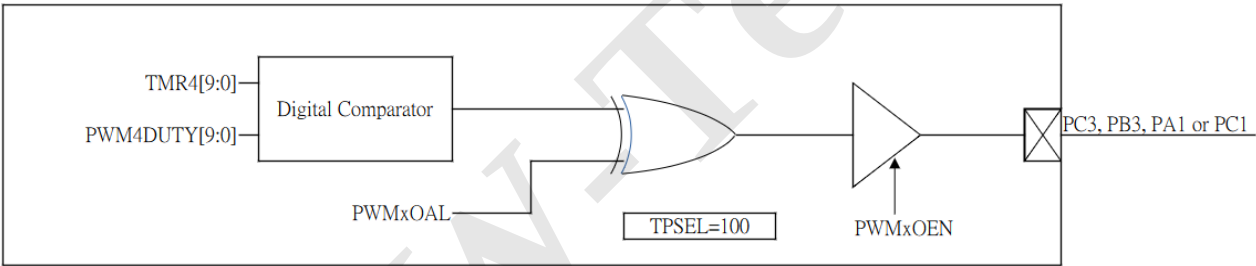


图 17 PWM4 结构框图

3.7.9 PWM4DUTY（PWM 4 占空比寄存器）

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM4DUTYL	0x97	PWM4DUTY[7:0]							
读/写属性		读/写							
初始值		XXXXXXXX							

PWM4DUTY[7:0]: PWM4 占空比数据LSB 8 位。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM4DUTYH	0x98	-	-	-	-	-	-	PWM4DUTY[9:8]	
读/写属性		-	-	-	-	-	-	读/写	
初始值		X	X	X	X	X	X	XX	

PWM4DUTY[9:8]: PWM4 占空比数据MSB 2 位。

3.7.10 PWM5

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PWM4 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。

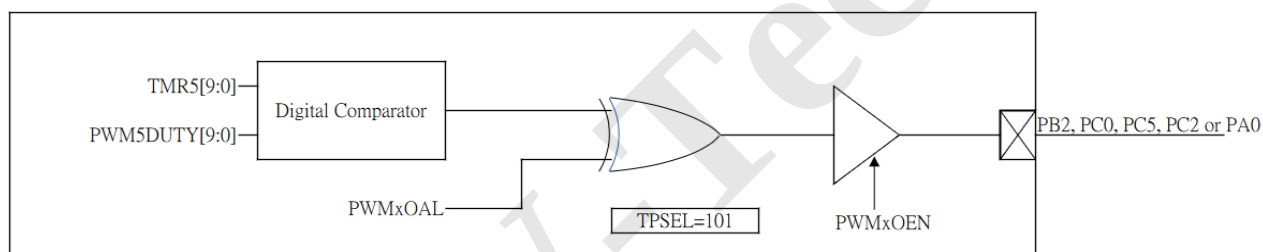


图 18 PWM5 结构框图

3.7.11 PWM5DUTY (PWM5 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		读/写							
初始值		XXXXXXXX							

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		-	-	-	-	-	-	读/写	
初始值		X	X	X	X	X	X	XX	

PWM5DUTY[9:8]: PWM5 占空比数据MSB 2 位。

3.7.12 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：

$$\text{PWM 周期} = [(PRx) + 1] \cdot T_{\text{CPU 周期}} \cdot (TMRx \text{ 时钟周期})$$

-- 用于同步模式

PWM 频率定义为 1/[PWM 周期]。

3.7.13 PWM 占空比

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

公式 2:

$$\text{PWM 占空比 (\%)} = PDx / (PRx+1)$$

$$\text{PWM 占空比 (秒)} = PDx \cdot T_{\text{CPU 周期}} \cdot (TMRx \text{ 时钟周期})$$

-- 同步模式

$$\text{PWM 占空比 (秒)} = PDx \cdot T_{\text{HRC 周期}}$$

-- 异步模式

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）如下公式所示：

公式 3：

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

例如：PWM Resolution (max) = $\log(1023+1) / \log(2)$

$$= 10/1$$

$$= 10 \text{ bits}$$

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 16 在 16 MHz 2T 同步模式下 PWM 频率和分辨率

3.7.14 PWMDBx

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	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

根据CCPSEL的当前值，访问虚拟SFR PWMDBx相当于访问物理SFR PWMDB1 或PWMDB2。

CCPSEL = 0。

DB<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。

CCPSEL = 1。

DB<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。

3.8 CCP模式

AT8BM84A CCP 模块的主要功能包括：

- 一个 16 位输入捕获模块，用于输入一系列边缘事件：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。

3.8.1 模块配置

当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。当这些寄存器被CCP操作占用时，原始定时器或PWM功能可能无法工作。要注意不要开启这个定时器相关的功能。表 17 总结了各种CCP模式及其征用寄存器。

CCP 数量	CCP 模式	被征用的 PWM 寄存器	被征用的定时器寄存器
CCP1	捕捉	捕捉寄存器 (16-bit) = {PWM5DUTYL[7:0], PWM4DUTYL[7:0]}	{TMR5L[7:0], TMR4L[7:0]}
CCP1	比较	比较寄存器 (16-bit) = {PWM5DUTYL[7:0], PWM4DUTYL[7:0]}	{TMR5L[7:0], TMR4L[7:0]}
CCP1	PWM(HB/FB)	PWM 占空比寄存器 (10-bit) = {PWM5DUTYH[1:0], PWM5DUTYL[7:0]}	{TMR5H[1:0], TMR5L[7:0]}
CCP2	PWM(HB)	PWM 占空比寄存器 (10-bit) = {PWM4DUTYH[1:0], PWM4DUTYL[7:0]}	{TMR4H[1:0], TMR4L[7:0]}

表 17 AT8BM84A CCP定时器和PWM寄存器资源

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

3.8.2 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（PxA到PxD）。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 18 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。

模式	比较		捕捉		PxA		PxB		PxC		PxD	
IO 属性	输出		输入		输出		输出		输出		输出	
	CCP1	CCP2	CCP1	CCP2	CCP1	CCP2	CCP1	CCP2	CCP1	CCP2	CCP1	CCP2
比较	PB2	-	-	-	-	-	-	-	-	-	-	-
捕捉	-	-	PB2	-	-	-	-	-	-	-	-	-
PWM 单个	-	-	-	-	PB2/PC4(*1)	-	-	-	-	-	-	-
半桥	-	-	-	-	PB2/PC4(*1)	PC2	PA5/PA1(*1)	PC5	-	-	-	-
全桥	-	-	-	-	PB2/PC4(*1)	-	PA5/PA1(*1)	-	PA2	-	PA3	-

表 18 CCP I/O引脚列表

注意 (*1)：引脚按选项来选择。

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寄存器初始引脚方向，通过IO端口寄存器初始引脚状态。
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，很顺利的配置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。

3.8.3 CCPxCON

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCPxCON	0x11f	PWMxM1	PWMxM0	FBCH1	FBCH0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
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

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

对于CCPSEL=0，

CCPxM[3:0]（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 = 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。

PWMxM[1:0]（可用 CCPxM[3:2] = 11）：

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。

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。

CCP 用户注意比较/捕捉模式步骤：

a. 设置定时器 5 工作模式：连续/单次（T5OS），溢出重载模式（T5RL）（保持 T5EN=PWM5OEN=0）。

b. 设置定时器 4 时钟源，定时器 4 预分频器。

c. 设置定时器 4/定时器 5 重载/初始值。

d. 设置 PWM4 占空比/PWM5 占空比。

e. 进入“比较/捕捉模式”（设置 CCT0MD 寄存器），启动定时器。

f. 在更新 PWM/定时器的数据之前，请关闭比较/捕捉模式。

g. PB2 为“捕捉输入”或“比较输出”。

对于CCPSEL=1，

CCP2M[3:0]（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。

其它 = N.C.

PWM1M[1:0]（可用 CCP2M[3:2] = 11）：

10 = PWM 半桥输出。

其它 = N.C.

3.8.4 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，捕获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。

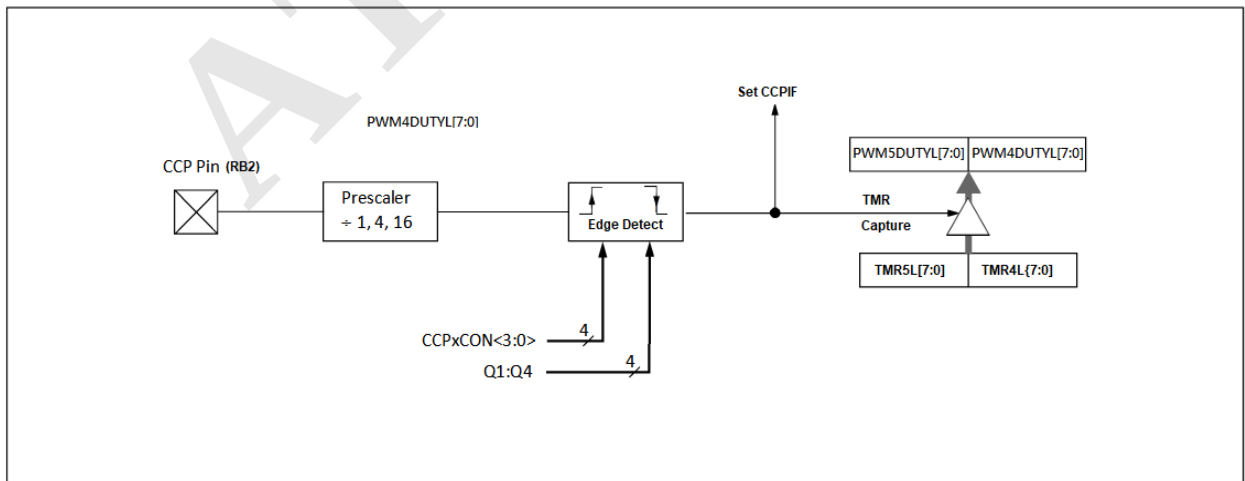


图 19 捕捉模式结构框图

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所示。

3.8.5 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。

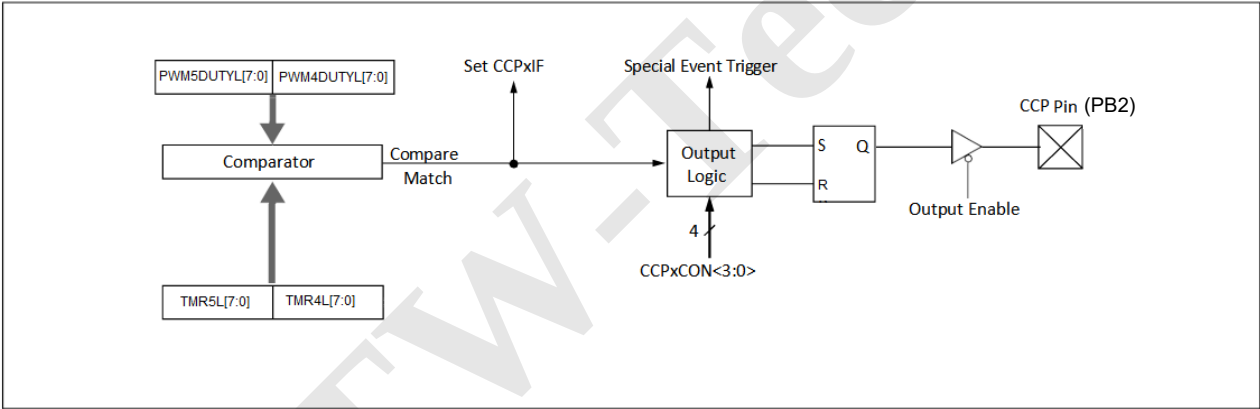


图 20 比较模式结构框图

3.8.6 CMPCON（比较器控制寄存器）

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCON	0x9C	CMPEM	BIASEN	-	-	-	-	CMPOUT	CMPOE
读/写属性		读/写	读/写	-	-	-	读	读/写	读/写
初始值		0	0	X	X	X	X	1	1

CMPEM: 开启/关闭比较器。

CMPEM=1，开启比较器。

CMPEM=0，关闭比较器。

BIASEN: 开启/关闭比较器/ LVD模块偏置。

BIASEN=1，开启比较器/ LVD模块偏置。

BIASEN=0，关闭比较器/ LVD模块偏置。

CMPOE: 开启/关闭比较器输出到PB3 引脚。

CMPOE=1，开启比较器输出到PB3 引脚。

CMPOE=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。

CMPOUT: 比较器输出状态，只读。

3.8.7 CMPCR（比较器电压选择控制寄存器）

名称	地址	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

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

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	-

CMPF_INV: 比较器反向输出控制位。

CMPF_INV = 1, 比较器反向输出。

CMPF_INV = 0, 比较器正向输出。

RBIAS_L, RBIAS_H: 设置相应的电压基准电平。

3.8.8 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 3.8 节中描述的PWM的另一种选择。

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 21 显示了该模块在PWM模式下的简化结构框图。

PWM输出（图 22）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）。

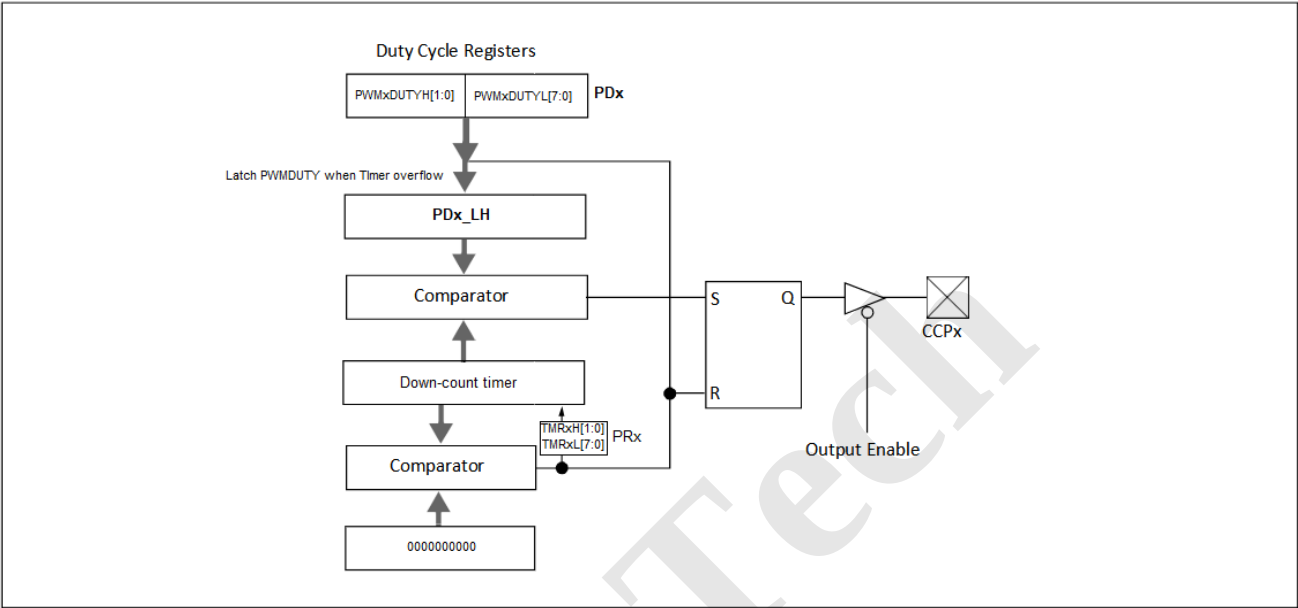


图 21 PWM结构框图

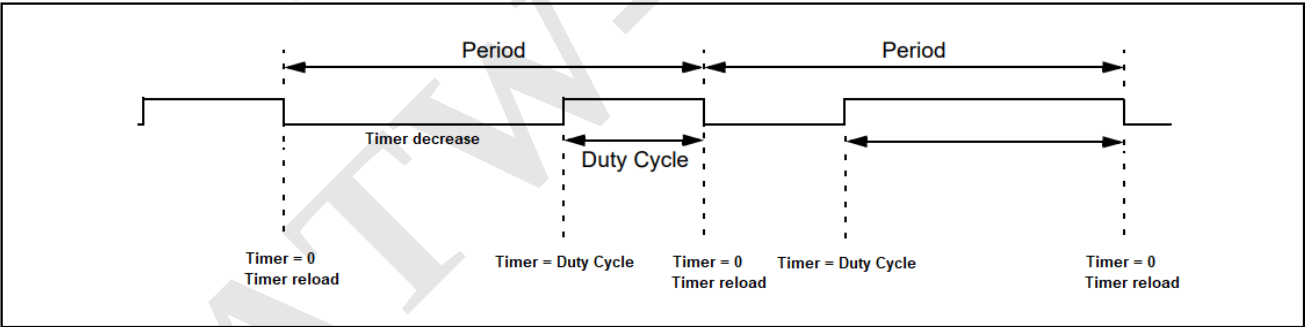


图 22 PWM 输出

3.8.9 PWM 周期

PWM周期通过写入PRx = [TMRxH[1:0], TMRxL[7:0]]寄存器（x= 4,5）来指定。PWM周期的计算公式如下：

$$\text{PWM 周期} = [(PRx) + 1] \cdot T_{\text{CPU 周期}} \cdot (\text{TMRx 预分频值})$$

-- 用于同步模式

$$\text{PWM 周期} = [(PRx) + 1] \cdot T_{\text{HRC 周期}}$$

-- 用于异步模式

PWM频率定义为 $1/[\text{PWM周期}]$ 。CCP计时器从重新加载值PRx中减少计数器和计数。当TMRx[9:0]等于PWM占空比时，设置CCPx引脚（例外：如果PWM占空比= 0，则不设置CCPx引脚）。当TMRx减小到 0 时，下一个增量周期会发生以下三个事件：

- PRx被重新加载到TMRx。
- CCPx引脚清除。
- PWM占空比（PDx）从PWMxDUTYH[1:0], PWMxDUTYL[7:0]（x=4 或 5）中锁存。

3.8.10 PWM 占空比

PWM占空比（PDx）通过写入PWMxDUTYH[1:0], PWMxDUTYL[7:0]（PDx, x=4 或 5）寄存器位来指定。最高可达 10 位分辨率。以下公式用于计算PWM占空比（百分比或时间）：

$$\text{PWM 占空比 (\%)} = PDx / (PRx+1)$$

$$\text{PWM 占空比 (秒)} = PDx \cdot T_{\text{CPU 周期}} \cdot (\text{TMRx 预分频值})$$

-- 用于同步模式

$$\text{PWM 占空比 (秒)} = PDx \cdot T_{\text{HRC 周期}}$$

-- 用于异步模式

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：

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

注意：如果PWM占空比大于PWM周期，CCP引脚将不被设置。

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	2	4	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 19 在 16MHz 2T 同步模式下的典型 PWM 频率和分辨率

3.8.11 PWM 操作设置

为PWM操作配置CCP模块：

- 设置正确的引脚方向和引脚初始值。
- 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 模式，预分频器值，然后通过写入TxCR1 开启定时器 x。
- 为PWM操作配置CCPxCON寄存器。

3.8.12 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 3.8 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）。

表 20 提供了每个增强型PWM模式的引脚分配。

图 23 提供了增强型PWM模块的简化框图示例。

图 24 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。

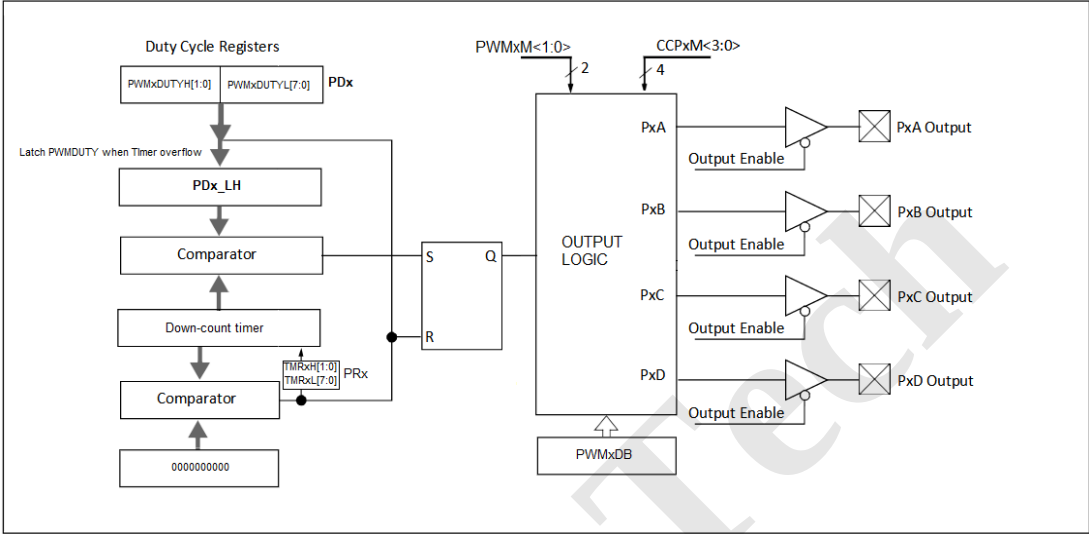


图 23 增强型PWM模式的结构框图

ECCP 模式	PWMxM<1:0>	PxA		PxB		PxC		PxD	
CCPx		CCP1	CCP2	CCP1	CCP2	CCP1	CCP2	CCP1	CCP2
单个	00	PB2/PC4(*1)	-	-	-	-	-	-	-
半桥 (死区)	10	PB2/PC4(*1)	PC2	PA5/PA1(*1)	PC5	-	-	-	-
全桥, 正向	01	PB2/PC4(*1)	-	PA5/PA1(*1)	-	PA2	-	PA3	-
全桥, 反向	11	PB2/PC4(*1)	-	PA5/PA1(*1)	-	PA2	-	PA3	-

表 20 各种PWM增强模式的引脚分配

注（*1）：Pad按选项选择。注 1：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。

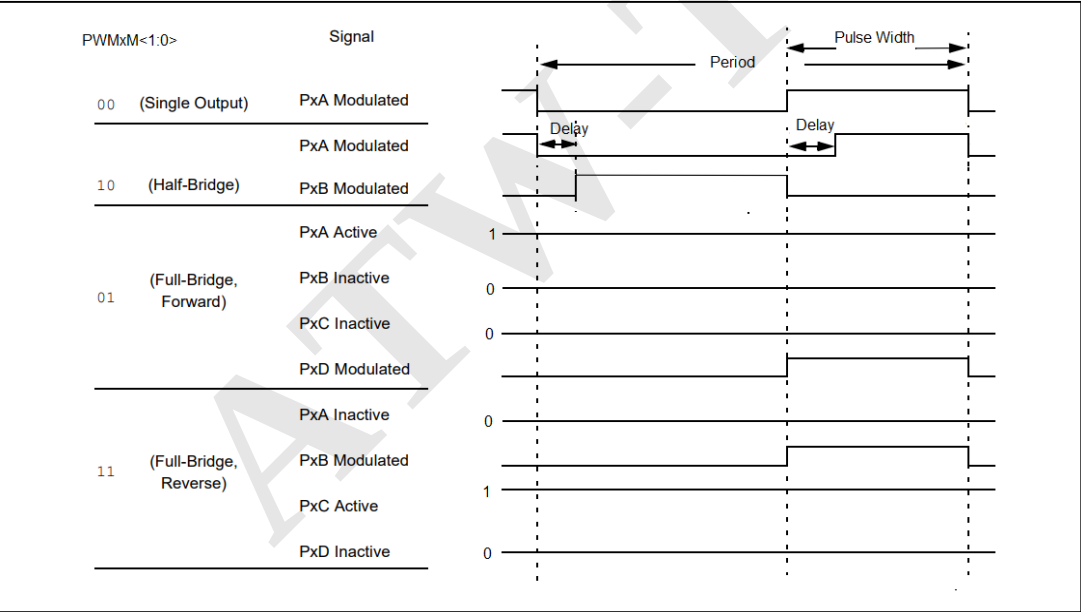


图 24 增强型PWM输出关系（高有效状态）示例

3.8.13 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挽负载。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 8）。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。

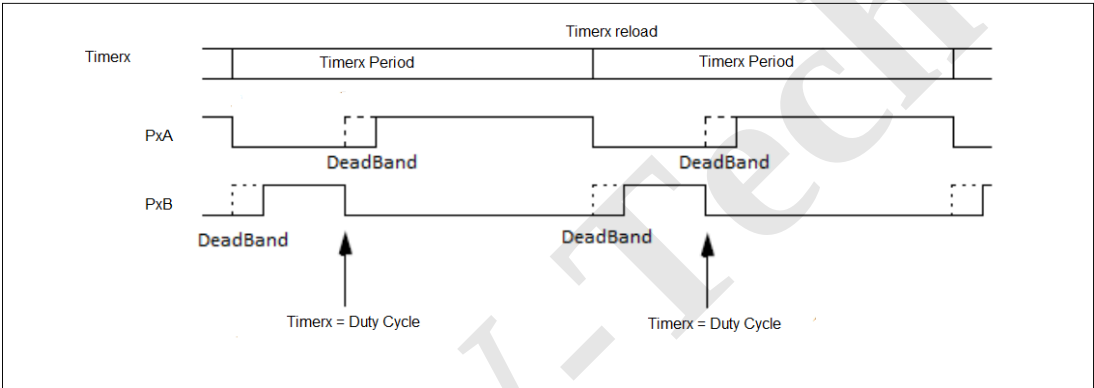


图 25 半桥PWM输出示例

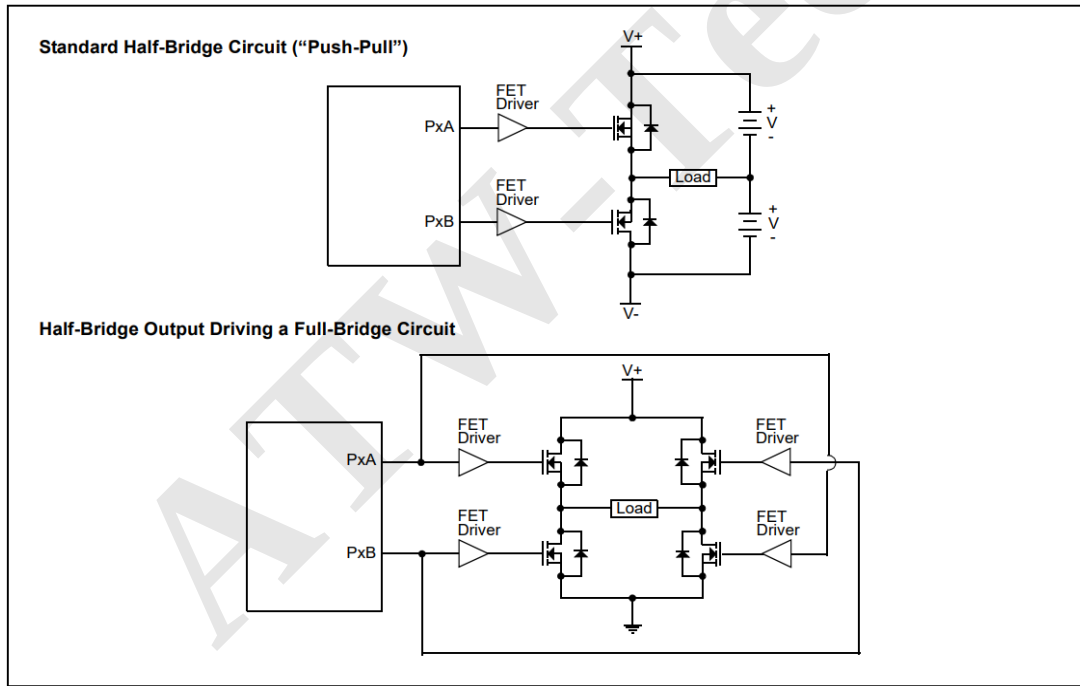


图 26 半桥应用示例

在半桥应用（见图 26）中，所有电源开关都以 PWM 频率调制，电源开关的关闭通常比打开需要更多的时间。如果上下电源开关同时开关（一个打开，另一个关断），两个开关都可能短时间打开，直到一个开关完全关闭。在这个短暂的间隔期间，一个非常大的电流（击穿电流）将流过两个电源开关，使桥接电源短路。为了避免这种潜在的破坏性击穿电流在开关过程中流动，通常会延迟打开任何一个电源开关，以允许另一个开关完全关闭。

在半桥模式下，可以使用数字可编程的死区延迟来避免击穿电流破坏桥接电源开关。延迟发生在信号从非有效状态到有效状态的转换处。相关的 PWMxDB 寄存器的低八位设置了微控制器指令周期（同步模式）或 IHRC 周期（异步模式）方面的延迟周期。

注意：如果死区值大于 PWM 占空比，相应的输出在整个周期内保持非有效状态。

3.8.14 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 27 提供了一个全桥应用的示例。

在正向模式下，PxA 引脚被驱动到其有效状态，PxD 引脚被调制，而 PxB 和 PxC 引脚被驱动到其非有效状态，如图 28 所示。

在反向模式下，PxC 引脚被驱动到其有效状态，PxB 引脚被调制，而 PxA 和 PxD 引脚被驱动到其非有效状态，如图 28 所示。

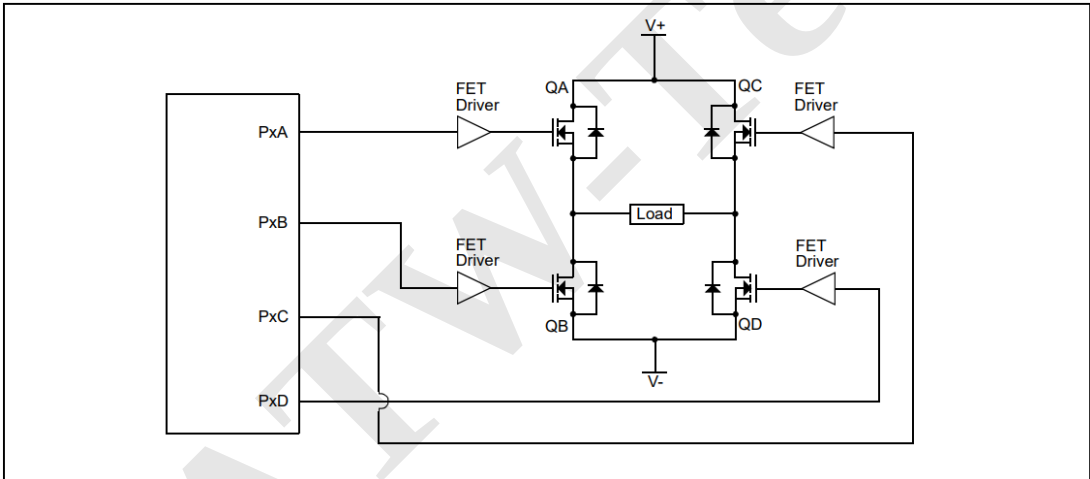


图 27 全桥应用示例

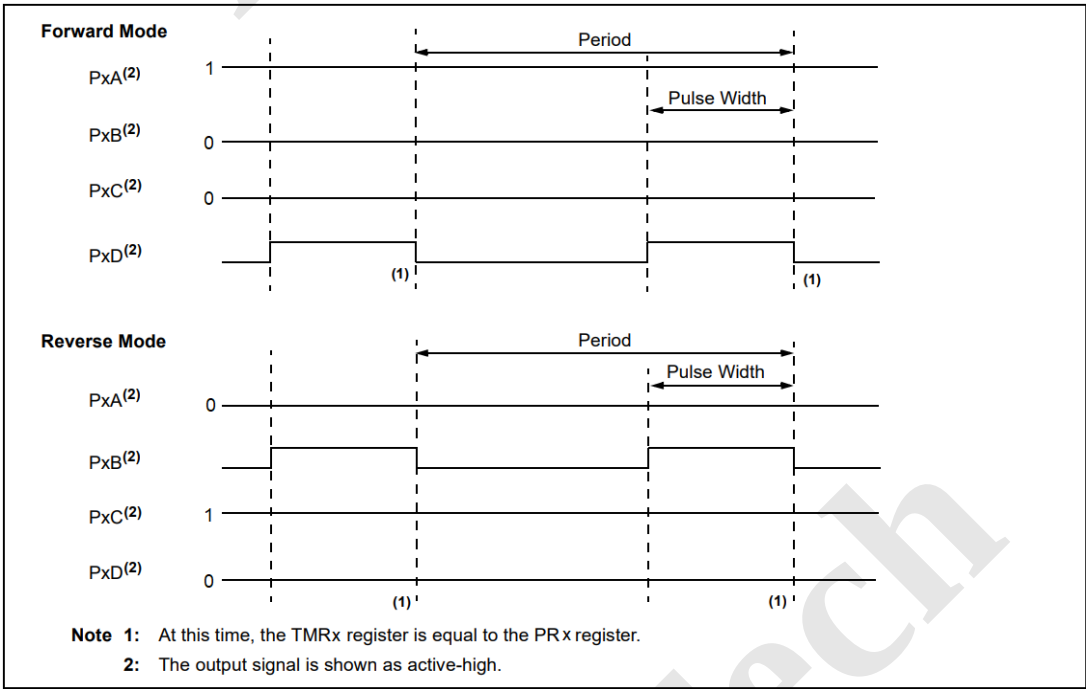


图 28 全桥PWM输出

3.8.15 全桥模式下的方向改变

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。以下顺序发生在当前 PWM 周期结束之前：

- 1. 调制输出（PxB和PxD）被置于非有效状态。
- 2. 相关的未调制输出（PxA和PxC）被切换到相反方向驱动。

PWM调制在下一个周期开始时恢复。为了说明这个序列，参见图 29。

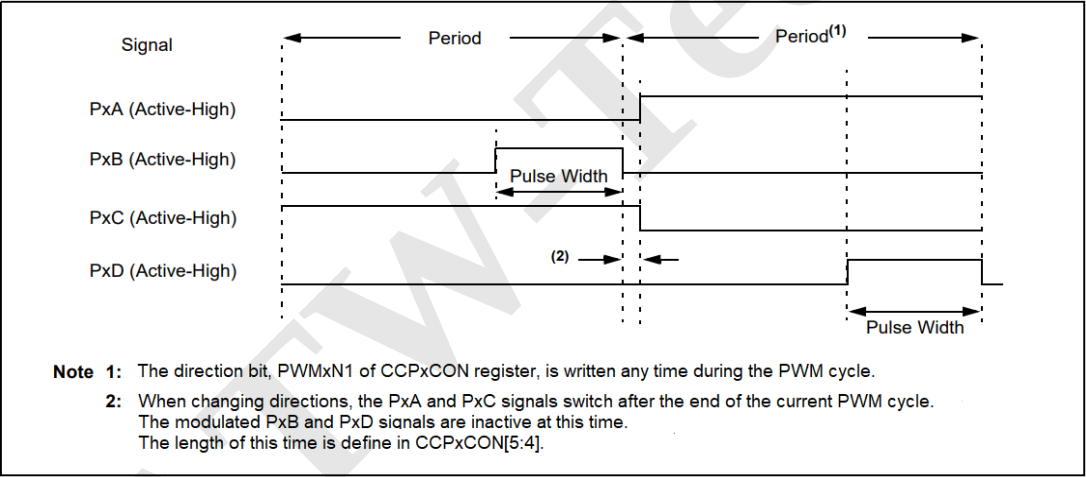


图 29 PWM方向改变示例

全桥模式不提供死区延时。由于一次只调制一个输出，因此通常不需要死区延时。有一种情况需要死区延时。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）的关断时间大于导通时间。

图 30 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输出变为非有效，而PxC输出变为有效。由于电源器件的关断时间比导通时间长，因此在 t 段时间内会有击穿电流流过电源器件QC和QD（见图 10）。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。其他防止击穿电流的选项可能存在。

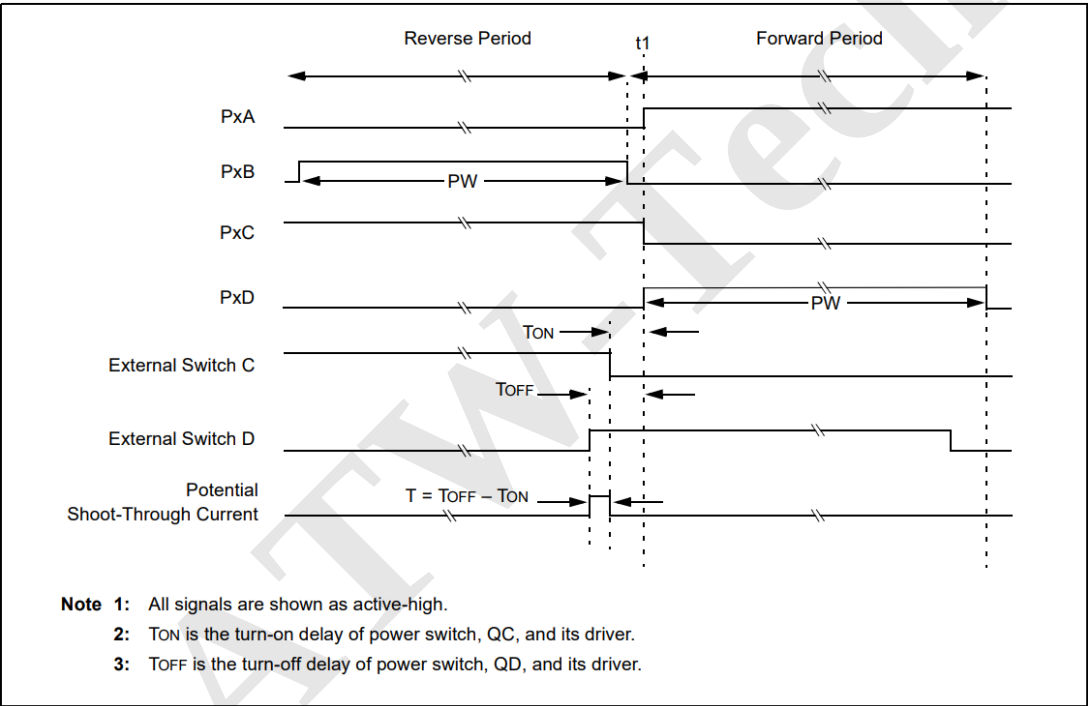


图 30 PWM方向变化在接近 100%占空比的示例

3.8.16 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。

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输出信号是高效还是低有效。在PWM引脚输出驱动器启用之前，必须选择PWM输出极性。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。

3.8.17 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。

3.8.18 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。

3.9 RFC模式

AT8BM84A内置RFC模式。一旦启用RFC模式，所选的输入脚状态将控制Timer1 计数。当所选输入脚被识别为 0 状态（输入脚电压小于 V_{IL} ）时，Timer1 继续计数。当所选引脚被识别为 1 时（输入脚电压大于 V_{IH} ），Timer1 停止计数。RFC模式的工作原理如下图所示：PSEL3~0 用于从 16 个AT8BM84A输入脚中选择一个RFC输入脚。RFCEN用于在正常使能信号T1EN和RFC选择输入状态之间切换Timer1 使能信号。

RFC模式的一个应用是测量电容-电阻充电时间，如图所示，当PSEL3~0=0x01 时，选择PA1 作为RFC输入脚。首先将PA1 设置为输出低电平（PA1 的电压放电到 0）。下一步，清除Timer1 内容，设置PA1 为输入，开启RFC模式。然后Timer1 开始计数,RC电路开始对PA1 充电。当PA1 被充电到 V_{IH} 电压时，由于PA1 输入是高电平，Timer1 计数停止。Timer1 内容将显示RC电路充电时间。

注意：Timer1 是向下计数。

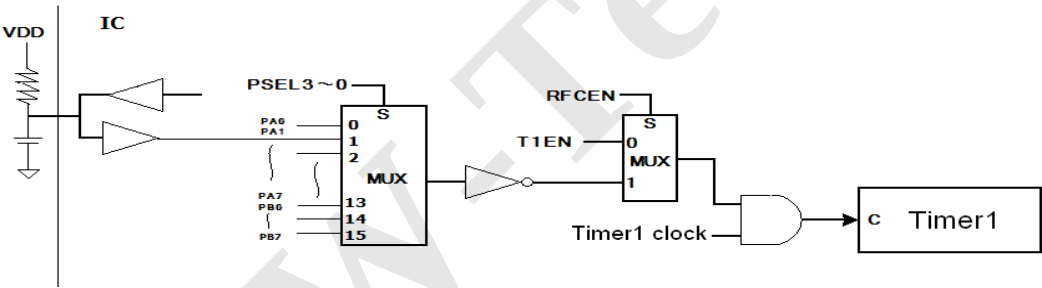


图 31 RFC结构框图

3.9.1 RFC（RFC 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			

RFCEN: 开启/关闭RFC功能。
RFCEN=1，开启RFC功能。
RFCEN=0，关闭RFC功能。

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

表 21 RFC引脚选择

3.10 IR（红外）载波

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F_{HOSC}，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。

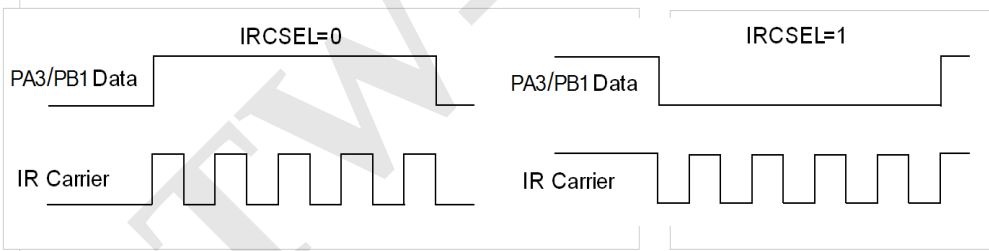


图 32 红外载波极性与输出数据

3.10.1 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	0	0	0

IREN: 开启/关闭红外载波输出。
IREN=1, 开启红外载波输出。
IREN=0, 关闭红外载波输出。

IRF57K: 红外载波频率的选择。
IRF57K=1, 红外载波频率为 57KHz。
IRF57K=0, 红外载波频率为 38KHz。

IRCSEL: 红外载波极性选择。
IRCSEL=0: 当I/O引脚数据为 1 时产生红外载波。
IRCSEL=1: 当I/O引脚数据为 0 时产生红外载波。

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。
IROSC358M=1, 晶振频率为 3.58MHz。
IROSC358M=0, 晶振频率为 455KHz。

注意:

1. 只有高频振荡（F_{HOSC}）（见 3.17 节）可以用作红外时钟源。
2. 不同振荡类型的分频比。

OSC. Type	57KHz	38KHz	条件
High IRC(4MHz)	64	96	HIRC 模式（无论系统时钟是什么，IR 模块的输入都设置为 4MHz）
Xtal 3.58MHz	64	96	Xtal 模式 & IROSC358M=1
Xtal 455KHz	8	12	Xtal 模式 & IROSC358M=0

表 22 不同振荡类型的分频比

3.11 低电压检测（LVD）

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，且 VDD 电压值低于 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。LVD 框图如下：

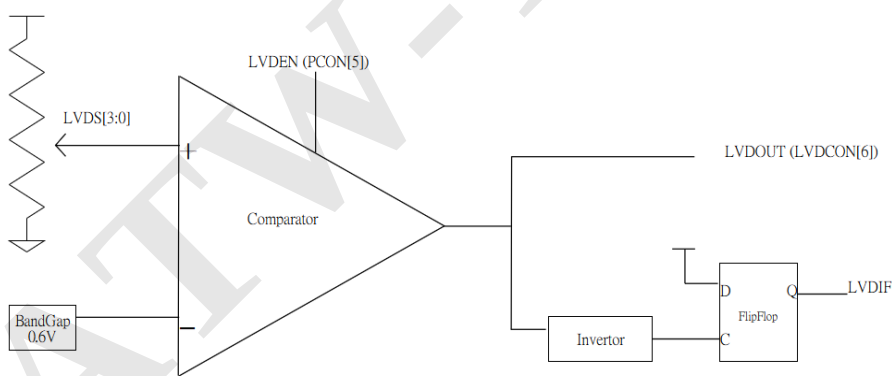


图 33 LVD结构框图

3.11.1 LVDCON（LVD 控制寄存器）

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

LVDOOUT：低压检测输出，只读。

下表为LVD电压选择表。

LVDS[3:0]	电压
0000	1.9V
0001	2.0V
0010	2.2V
0011	2.4V
0100	2.6V
0101	2.8V
0110	2.9V
0111	3.0V
1000	3.15V
1001	3.30V
1010	3.45V
1011	3.60V
1100	3.75V
1101	3.90V
1110	4.05V
1111	4.15V

表 23 LVD电压选择

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:

LVDS[3:0]	电压
0000	--
0001	--
0010	(2.2+0.1) V
0011	(2.4+0.1) V
0100	(2.6+0.1) V
0101	(2.8+0.1) V
0110	(2.9+0.1) V
0111	(3.0+0.1) V
1000	(3.15+0.1) V
1001	(3.30+0.1) V
1010	(3.45+0.1) V
1011	(3.60+0.1) V
1100	(3.75+0.1) V

LVDS[3:0]	电压
1101	(3.90+0.1) V
1110	(4.05+0.1) V
1111	(4.15+0.1) V

表 24 充电应用时的LVD电压选择

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LVDCON[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。

电压阈值可以通过配置RBIAS_H、RBIAS_L和LVDS[3:0]进行调节。

可用的监测级别详见下表：

RBIASH	RBIASL	LVDS[3:0]	Voltage	Unit
0	1	0110	4.837	V
1	1	1100	4.835	
0	1	0101	4.539	
1	1	1011	4.387	
0	0	1111	4.15	
0	0	1110	4.05	
0	1	0100	3.989	
1	1	1010	3.987	
0	0	1101	3.9	
0	0	1100	3.75	
1	1	1001	3.625	
0	0	1011	3.6	
0	1	0011	3.496	
0	0	1010	3.45	
0	0	1001	3.3	
1	1	1000	3.298	
0	0	1000	3.15	
0	1	0010	3.05	
0	0	0111	3	
1	1	0111	3	
0	0	0110	2.9	
1	1	0110	2.816	
0	0	0101	2.8	
0	1	0001	2.645	
1	1	0101	2.642	
0	0	0100	2.6	
1	0	1111	2.58	
1	0	1110	2.518	
0	1	0000	2.456	
1	0	1101	2.424	
0	0	0011	2.4	
1	0	1100	2.331	
1	1	0100	2.322	
1	0	1011	2.238	
0	0	0010	2.2	
1	0	1010	2.145	
1	0	1001	2.051	
1	1	0011	2.035	
0	0	0001	2	

表 25 参考电压选择表

3.12 电压比较

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ANAEN[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，启用比较器。在睡眠模式下，比较器被自动禁用。

比较器结构如下图所示：

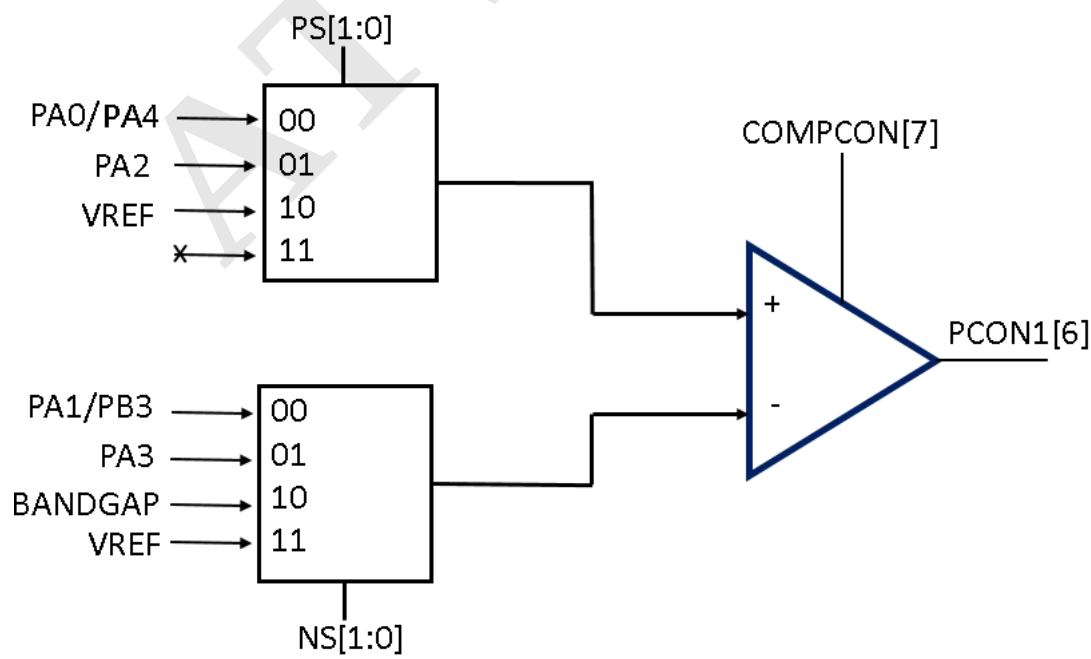


图 34 比较器硬件连接

3.13 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。

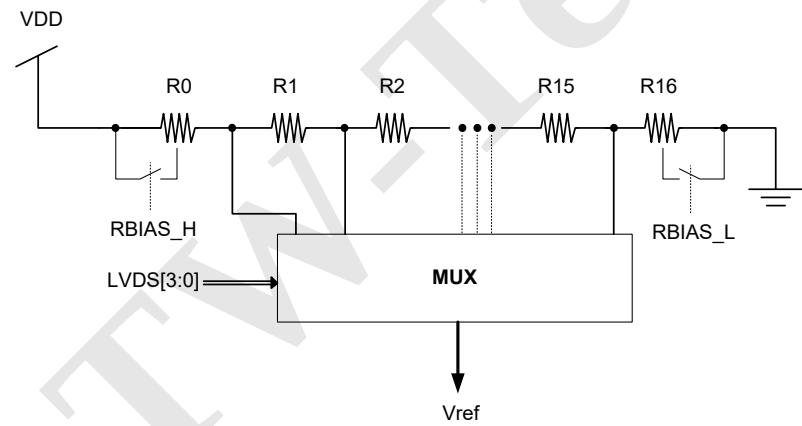


图 35 Vref硬件连接

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。

Condition		
LVD		trim
RBIAS[H:L]	LVD S[3:0]	VDD=5V $V_{IN} = VDD_REF$ (V)
10 0000b		0.508*VDD =
10 0001b		0.483*VDD =
10 0010b		0.439*VDD =
10 0011b		0.402*VDD =
10 0100b		0.371*VDD =
10 0101b		0.345*VDD =
10 0110b		0.333*VDD =
10 0111b		0.322*VDD =
10 1000b		0.306*VDD =
10 1001b		0.292*VDD =
10 1010b		0.280*VDD =
10 1011b		0.268*VDD =
10 1100b		0.257*VDD =
10 1101b		0.247*VDD =
10 1110b		0.238*VDD =
10 1111b		0.233*VDD =
1 0000b		0.244*VDD =
1 0001b		0.227*VDD =
1 0010b		0.197*VDD =
1 0011b		0.172*VDD =
1 0100b		0.150*VDD =
1 0101b		0.132*VDD =
1 0110b		0.124*VDD =
1 0111b		0.116*VDD =
1 1000b		0.106*VDD =
1 1001b		0.096*VDD =
1 1010b		0.088*VDD =
1 1011b		0.080*VDD =
1 1100b		0.072*VDD =
1 1101b		0.065*VDD =
1 1110b		0.059*VDD =
1 1111b		0.055*VDD =

表 26 参考电压Vref选择表

比较器的正向输入由PS[1:0]（寄存器CMPCR[3:2]）决定。

表格如下所示：

PS[1:0]	正向输入
00	PA0 / PA4 (option)
01	PA2
10	Vref
11	---

表 27 正向输入选择

比较器的反相输入由 NS[1:0]（寄存器 CMPCR[1:0]）决定。

表格如下所示：

NS[1:0]	反向输入
00	PA1 / PB3 (option)
01	PA3
10	Bandgap (0.6V)
11	Vref

表 28 反向输入选择

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 功能被启用，则该功能将被禁用。

3.14 模拟-数字转换器（ADC）

AT8BM84A提供 21+2 通道 12 位SAR ADC，将模拟信号转换为 12 位数字数据。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。模拟输入可选择从模拟信号输入引脚PA0~PA4、PA6~PA7、PB0~PB7、PC0~PC5，从内部生成 $1/4 * VDD$ 或VSS。ADC时钟ADCLK可选择I_HRC/1、I_HRC/2、I_HRC/8 或I_HRC/16。采样脉冲宽度可以选择为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低转高后发出。结构框图如下图所示。

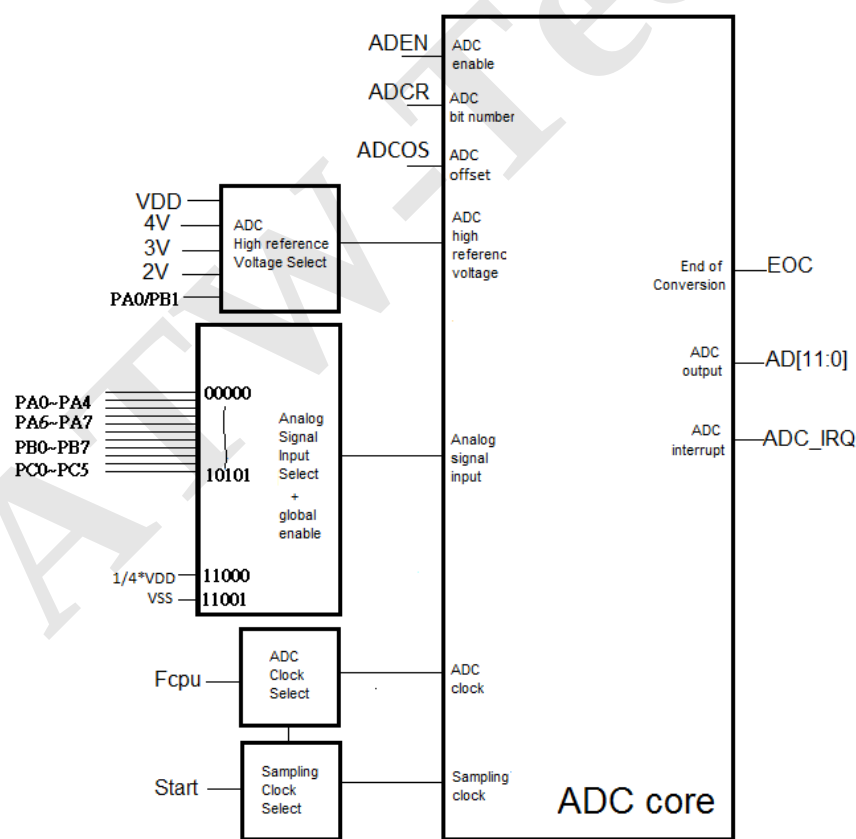


图 36 ADC结构框图

3.14.1 ADC 参考电压

ADC内置 5 个高参考电压源，由ADVREFH寄存器控制（见表 27）。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参考电压为来自外部电压源。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水平 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	x x	PA0 or PB1 (by option)
0	1 1	VDD
0	1 0	4V
0	0 1	3V
0	0 0	2V

表 29 ADC参考电压选择

3.14.2 ADC 模拟输入通道

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	x x x x x	x
X	0	x x x x x	x
1	1	0 0 0 0 0	PA0
1	1	0 0 0 0 1	PA1
1	1	0 0 0 1 0	PA2
1	1	0 0 0 1 1	PA3
1	1	0 0 1 0 0	PA4
1	1	0 0 1 0 1	x
1	1	0 0 1 1 0	PA6
1	1	0 0 1 1 1	PA7
1	1	0 1 0 0 0	PB0
1	1	0 1 0 0 1	PB1
1	1	0 1 0 1 0	PB2
1	1	0 1 0 1 1	PB3
1	1	0 1 1 0 0	PB4
1	1	0 1 1 0 1	PB5
1	1	0 1 1 1 0	PB6
1	1	0 1 1 1 1	PB7
1	1	1 0 0 0 0	PC0
1	1	1 0 0 0 1	PC1
1	1	1 0 0 1 0	PC2
1	1	1 0 0 1 1	PC3
1	1	1 0 1 0 0	PC4
1	1	1 0 1 0 1	PC5
1	1	1 1 0 0 0	1 / 4 * VDD
1	1	1 1 0 0 1	VSS
1	1	1 1 0 1 0 ~	N.C.

表 30 ADC模拟输入源选择

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。

3.14.3 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I_HRC，可从ADCK[1:0]中选择。

ADCK[1:0]	ADC 时钟
0 0	I_HRC/16
0 1	I_HRC/8
1 0	I_HRC/1
1 1	I_HRC/2

表 31 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
0 0	1 ADCLK
0 1	2 ADCLK
1 0	4 ADCLK
1 1	8 ADCLK

表 32 ADC采样时钟选择

ADC bit 数选择来自ADCR[1:0]。

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

表 33 转换位数选择

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

表 34 ADC转换时间

3.14.4 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。在EOC低转高后，ADC结果存储在寄存器ADDH和ADDL中。根据转换位数（ADCR）和右对齐或左对齐（ADFM）设置，ADC结果数据格式不同。具体请参见下图（图 37 ADFM 设置）。

3.14.5 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	W	读/写	读/写	读/写	读/写	读/写	读/写
初始值		0	0	1	0	0	0	0	0

ADEN: ADC开启位。

ADEN=1，ADC开启。

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。

GCHS: ADC全局通道选择位。

GCHS=0，关闭所有ADC通道。

GCHS=1，开启所有ADC通道。

CHS4~0: ADC输入通道选择位：

- 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。

3.14.6 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							

ADDL[7:0]: 低字节ADC数据缓冲区。

3.14.7 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							

ADDH[7:0]: 高字节ADC数据缓冲区。用户检查ADCCON1 (ADC控制寄存器) 图 37。

3.14.8 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	-	读	读/写	读/写	读/写	读/写	读/写
初始值		0	X	X	X	X	X	1	1

EVHENB: ADC参考高压 (VREFH) 选择控制位。

EVHENB=0, ADC参考高压是内部产生的, 所选电压取决于VHS1~0。

EVHENB=1, ADC参考高压由外部引脚PA0 提供。

EOC: ADC状态位, 只读。

EOC=1, ADC为转换结束, 在ADDL和ADDH中存在的ADC数据可用。

EOC=0, ADC正在进行中。

VHS1~0: ADC内部基准高压选择位。

11: VREFH=VDD;

10: VREFH=4V;

01: VREFH=3V;

00: VREFH=2V。

ADCK1~0: ADC时钟选择。

00: ADC clock=I_HRC/16;

01: ADC clock=I_HRC/8;

10: ADC clock=I_HRC/1;

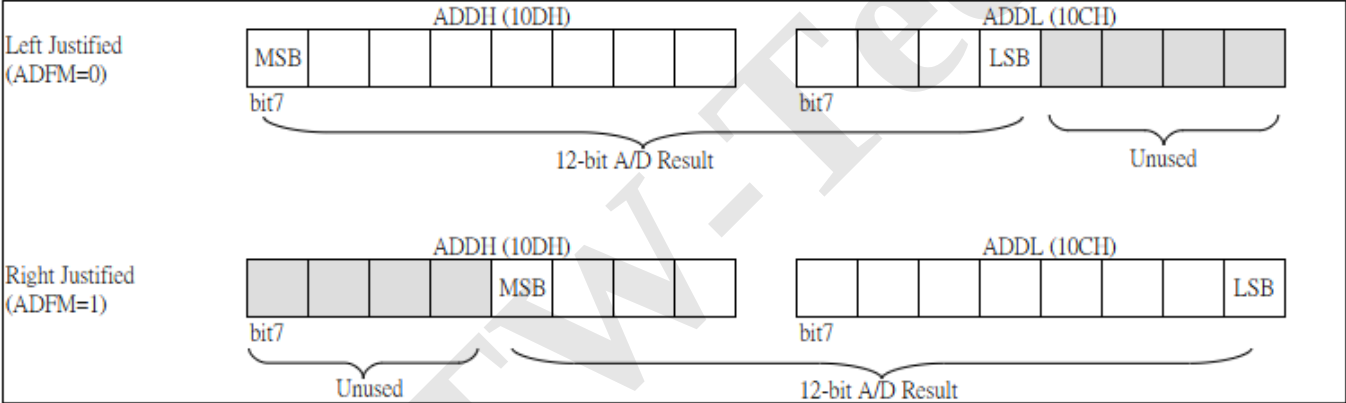
11: ADC clock=I_HRC/2。

ADFM: ADC结果格式选择位。

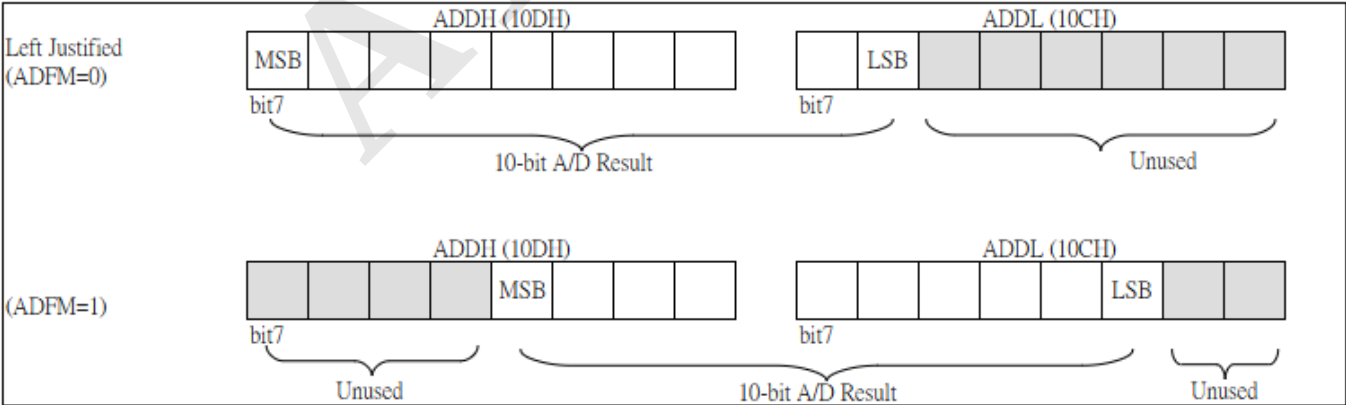
ADFM=1: 右对齐。

ADFM=0: 左对齐。

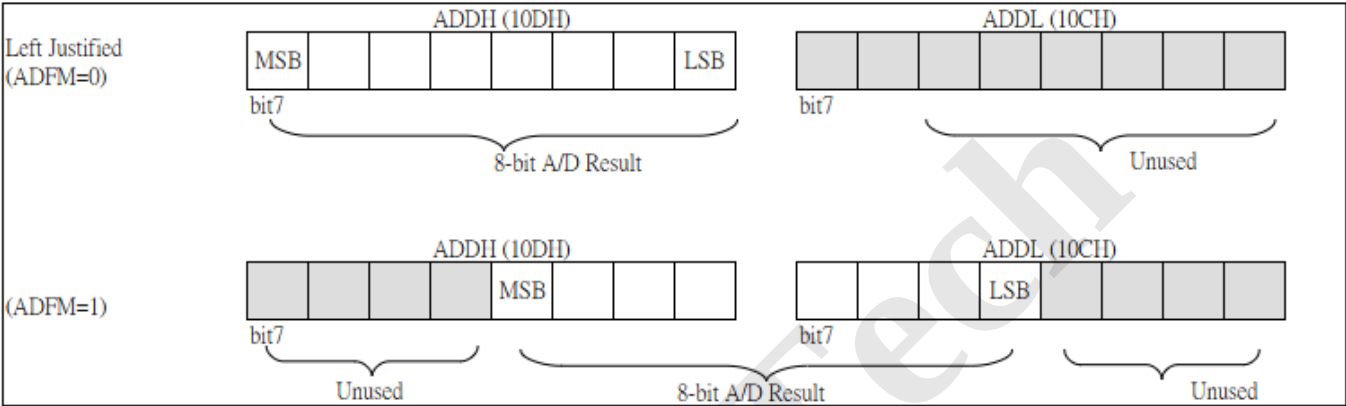
下图为M84 在不同转换位数下的A/D结果数据格式。



12-BIT A/D RESULT FORMATS



10-BIT A/D RESULT FORMATS



8-BIT A/D RESULT FORMATS

图 37 ADFM 设置

3.14.9 ADJMD (ADC 补偿寄存器)

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
初始值		X	X	0	0	0	0	0	0

ADJ[4:0]: ADC调校补偿。

00000 = 补偿 0mv。

11111 = 补偿 11mv。

ADJ_SIGN: ADC调校记号位。

0 = ADC数据减小。

1 = ADC数据增加。

3.14.10 ADCR (ADC 采样时间与 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
初始值		0	0	X	X	1	0	1	0

SHCK1~0: ADC采样时间选择位。

00: 1个ADC时钟; 01: 2个ADC时钟; 10: 4个ADC时钟; 11: 8个ADC时钟。

ADCR1~0: ADC转换位数选择。

00: 8-bit ADC; 01: 10-bit ADC; 1x: 12-bit ADC

PxSEL1~0: 选择虚拟SFR PxCON寄存器到PACON、PBCON或PCCON的映射位。

00=端口A模拟引脚设置 (PACON) 经过虚拟SFR PxCON。

01=端口B模拟引脚设置 (PBCON) 经过虚拟SFR PxCON。

10= C端口模拟引脚设置 (PCCON) 经过虚拟SFR PxCON。

3.15 看门狗定时器 (WDT)

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能被配置字节开启或关闭。当WDT被配置字节开启时, 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节决定复位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, 由两个配置字节决定。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。如果WDT超时将复位AT8BM84A, 分频速率从1:1到1:128。如果选为WDT中断时, 则分频速率从1:2到1:256。

当预分频器0分配给WDT时，执行CLRWDWT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（INTF[6]）位将设置为1。如果寄存器WDTIE（INTE[6]）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。

3.16 中断

AT8BM84A提供14种硬件中断：

- 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）。
- UART接口模块读中断或写中断。

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。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 0x008 取出下一条指令。同时，AT8BM84A将自动清除寄存器位GIE。

3.16.1 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。

3.16.2 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。

3.16.3 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。

3.16.4 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。

3.16.5 WDT 超时中断

当WDT超时，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。

3.16.6 PA/PB/PC 输入状态改变中断

当将 PA_x , $0 \leq x \leq 7$, PB_y , $0 \leq y \leq 7$, PC_z , $0 \leq z \leq 5$ 配置为输入引脚，并且对应的寄存器位WUPAx, WUPBx 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。

3.16.7 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。

3.16.8 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。

3.16.9 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。

3.16.10 LVD 中断

当VDD电平低于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。

3.16.11 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CMP1E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。

3.16.12 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。

3.16.13 串行接口模式中断

当SPI模式的SPIF或IIC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。

3.17 振荡器配置

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）。

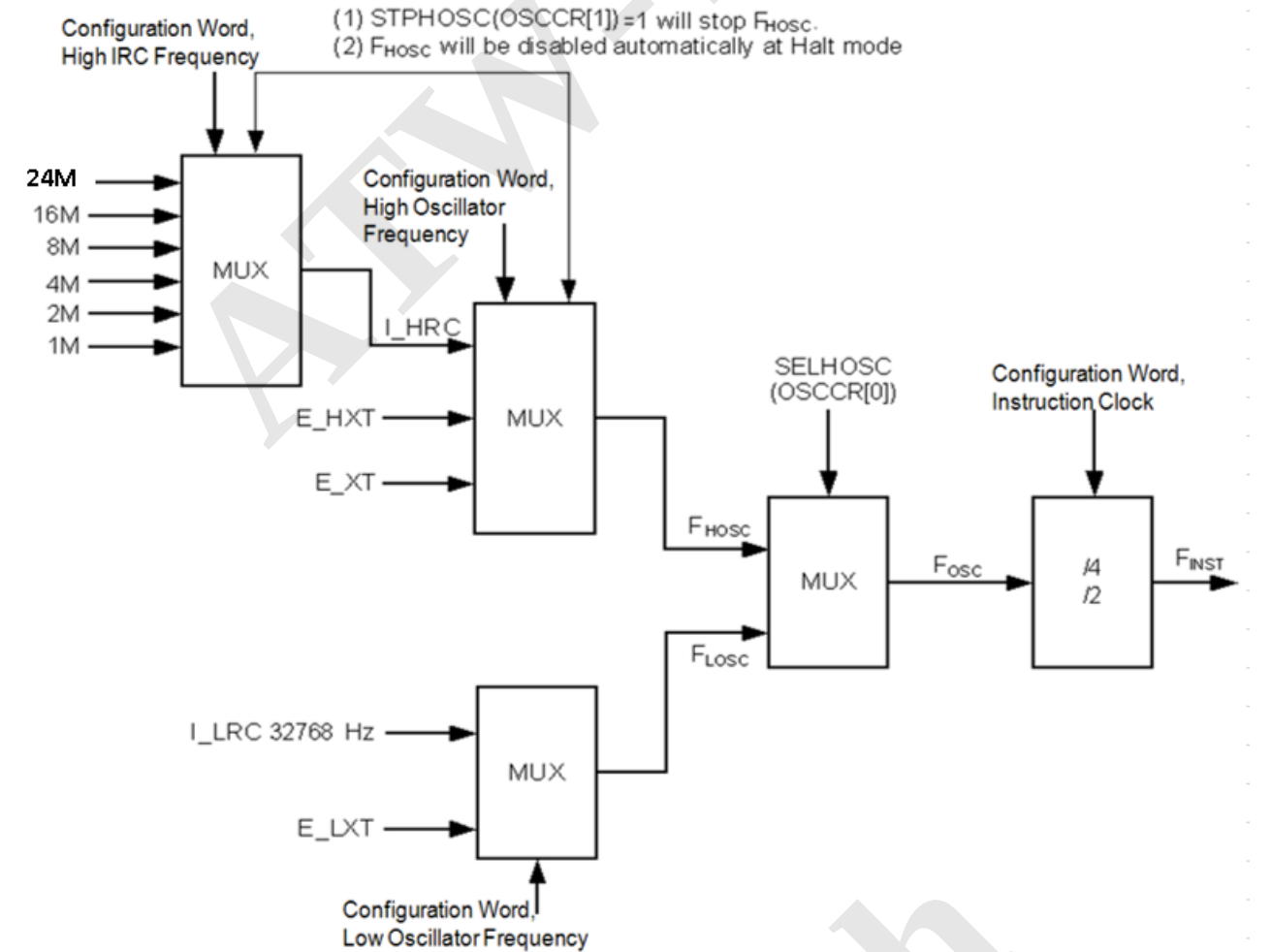


图 38 AT8BM84A的振荡器配置

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 24MHz。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 ~ 24MHz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。PA7 为晶体输出引脚（Xout）， PA6 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。PA7 为晶体输出引脚（Xout）， PA6 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 35 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 36 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 24MHZ谐振器，必须使用 18pF C2 电容。

为了获得精确稳定的 32.768k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：

$$C1 = C2 = 2 * CL - C_{bt}$$

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

在 25℃商业条件下，I_HRC的精度为±1%。

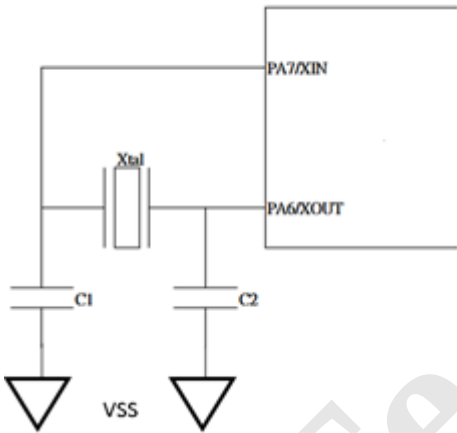


图 39 外部晶体振荡连接

根据寄存器位SELHOSC（OSCCR[0]）的值，可以选择F_HOSC或 F_LOSC作为系统振荡Fosc。当SELHOSC为 1 时，选择F_HOSC作为Fosc。当SELHOSC为 0 时，选择F_LOSC作为Fosc。一旦确定了Fosc，根据配置字的值，指令时钟F_INST可以是Fosc/2 或Fosc/4。

3.18 工作模式

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Halt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。

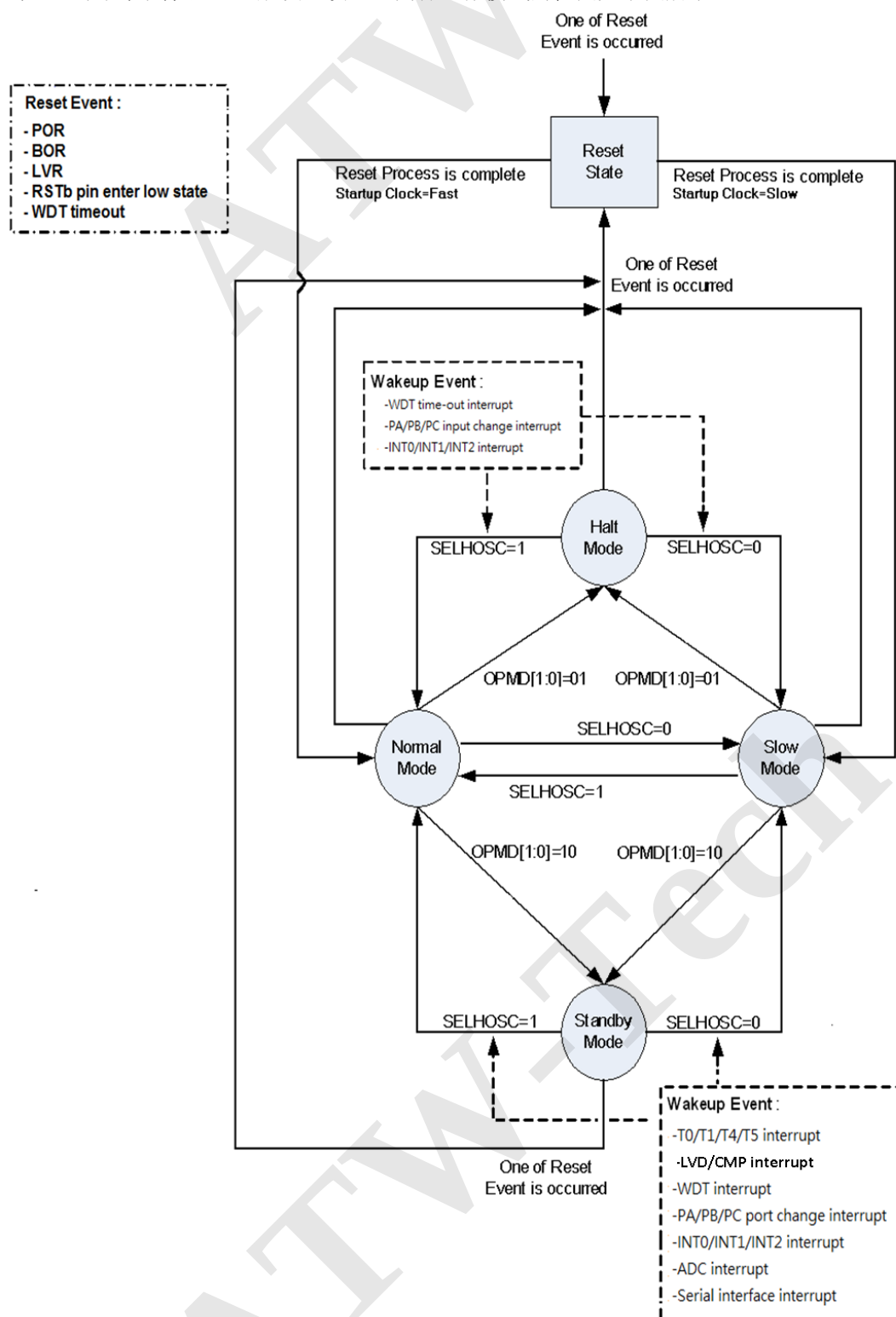


图 40 四种工作模式

3.18.1 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_{HOSC}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_{HOSC}为基础，且所有外围模块可根据相应的模块使能位激活。

F_{LOSC}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LCKTM0 设置为 1 并将相应的配置字Timer0 源设置为 1 来实现。

3.18.2 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_{LOSC}作为系统振荡，以节省功耗，但仍保持IC运行。但是，F_{HOSC}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_{HOSC}，必须先进入慢速模式，然后关闭F_{HOSC}，否则程序可能会被中止。

指令执行以F_{LOSC}为基础，所有外围模块可以根据相应的模块使能位激活。

F_{LOSC}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。

3.18.3 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_{HOSC}，用户必须进入慢速模式并将 1 写入寄存器位STPHOSC以停止F_{HOSC}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]、F_{INST}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_{LOSC} 可以通过向寄存器位STPHOSC写入 1 来关闭。

F_{LOSC} 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。

3.18.4 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。

3.18.5 唤醒稳定时间

睡眠模式的唤醒稳定时间由配置字节决定：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ENI。在唤醒后，AT8BM84A将跳转到地址 0x008，以便执行中断服务程序。如果在进入待机模式或睡眠模式之前执行DISI指令，则在唤醒后执行下一条指令。

3.18.6 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
FHOSC	使能	STPHOSC	STPHOSC	关闭
FLOSC	使能	使能	使能	关闭
指令执行	执行	执行	Stop	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
唤醒源	-	-	-定时器 0/1/4/5 上溢 - 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 37 工作模式概述

3.18.7 复位

当以下任一复位事件发生时，AT8BM84A将会进入复位状态并开始复位动作：

- 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
非睡眠模式时发生WDT复位	0	0
执行SLEEP 指令	1	0
执行CLRWDT 指令	1	1

表 38 /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，它由三位配置字节决定，这个时间可能是140us，4.5ms，18ms，72ms或288ms。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140us时，则Fosc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Fosc的时钟周期OST=16。

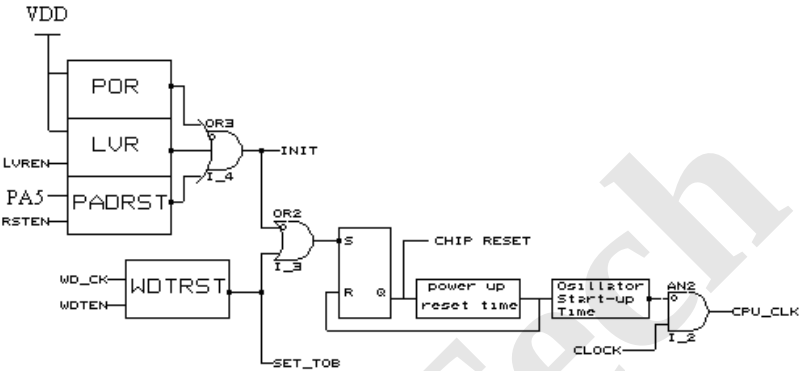


图 41 芯片复位电路框图

如果VDD缓慢上升，建议使用RSTb复位功能，如下图。

- 建议R阻值不大于40KΩ。
- R1 值= 100Ω ~ 1KΩ时，将阻止过大电流，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。

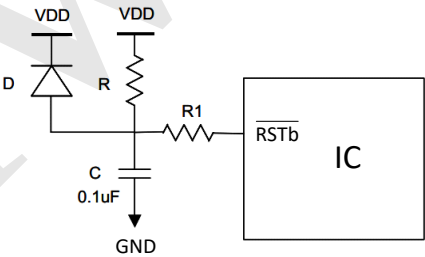


图 42 外部上电复位硬件连接图

3.19 SPI模式

SPI包含以下特性：

- 全双工操作。
- 4个可编程的精准模式频率选择。
- 带可编程极性和相位的串行时钟。
- 传输完成标志。
- 写入冲突标志。
- 总线冲突标志。

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链接。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。

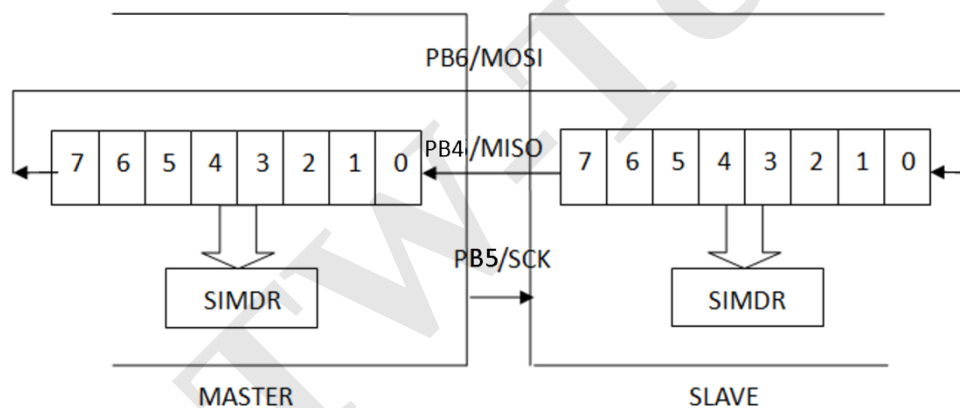


图 43 单个主机/从机

3.19.1 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	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

SIMD7-SIMD0: 如果IIC功能打开，它们是IIC模式数据寄存器；如果SPI功能打开，它们是SPI模式数据寄存器。

3.19.2 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 (SPE)	SIMC0(M EN)	MSTA	SSB_P AD	RX_P ADEN	TX_PA DEN	RCLK_P ADEN	UREN
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

SIMC1: SPI模式开启。

SIMC1 =1, SPI模式开启。

SIMC1 =0, SPI模式关闭。

SIMC0: IIC模式开启。

SIMC0 =1, IIC模式开启。

SIMC0 =0, IIC模式关闭。

MSTA: 主机/从机模式选择（包括IIC模式和SPI模式）。

MSTA =1, 选择主机模式。

MSTA =0, 选择从机模式。

SSB_PAD: 决定PB7 的引脚是否共享为SSBEN，此功能在SPI模式下使用。

SSB_PAD =1, PB7 用作SSBEN。

SSB_PAD =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。

RX_PADEN: UART端口RXPAD。

RX_PADEN =1, PB7 或PB0 用作UART RX信号输入。

RX_PADEN =0, PB7 或PB0 用作一般IO口。

TX_PADEN: UART端口TXPAD。

TX_PADEN =1, PB6 或PB1 用作UART TX信号输出。

TX_PADEN =0, PB6 或PB1 用作一般IO口。

RCLK_PADEN: UART端口 RCLK_PADEN。

RCLK_PADEN =1, PB4 用作UART RX CKT时钟输入，频率=波特率*/16。

RCLK_PADEN =0, PB4 用作一般IO口。

UREN: UART端口开启。

UREN =1, UART端口开启。

UREN =0, UART端口关闭。

3.19.3 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

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。

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。

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。

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。

CKEG: SPI SCK 时钟边沿选择。
CPOL=1
0: SCK空闲时为高电平，上升沿获取数据。
1: SCK空闲时为高电平，下降沿获取数据。
CPOL=0
0: SCK空闲时为低电平，上升沿获取数据。
1: SCK 空闲时为低电平，下降沿获取数据。

SPR[1:0]: SPI 时钟速率。

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

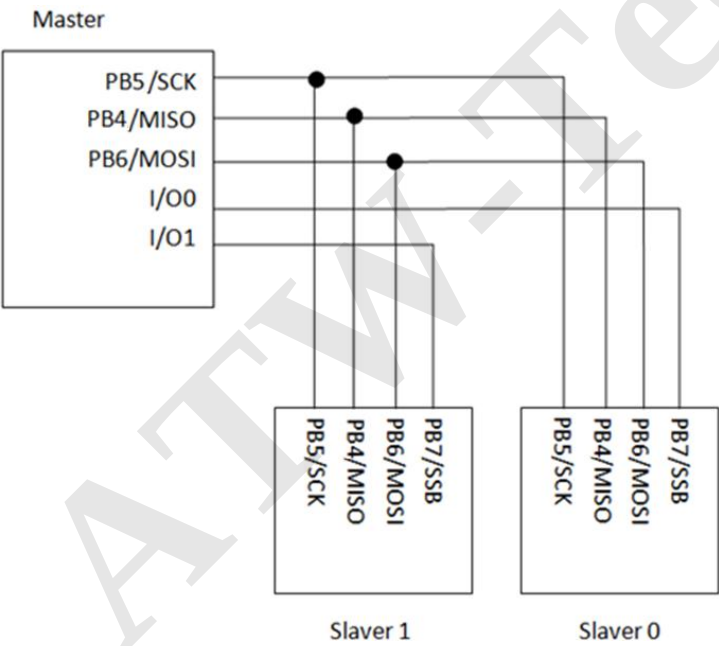


图 44 单个主机和多个从机

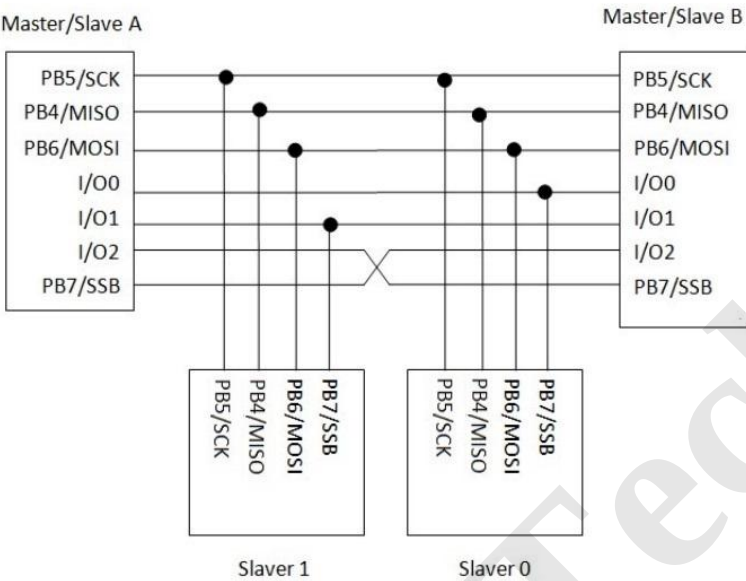


图 45 多个主机和多个从机

3.19.4 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。

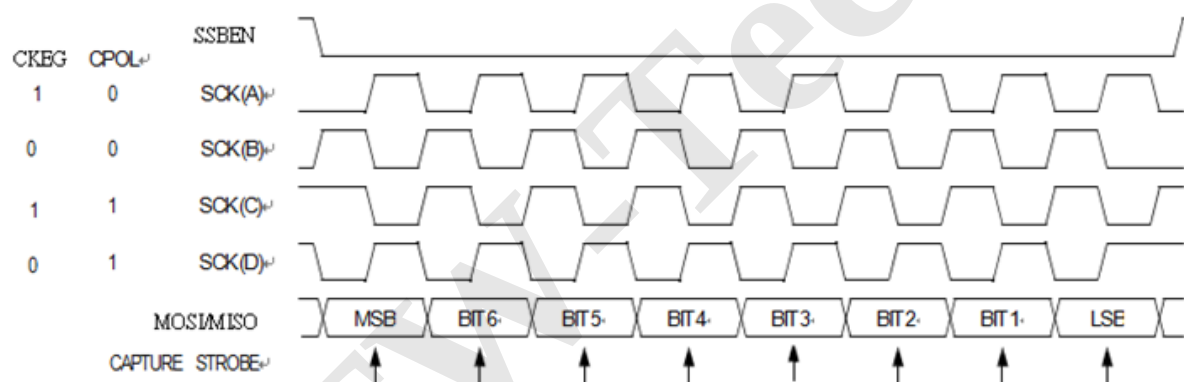
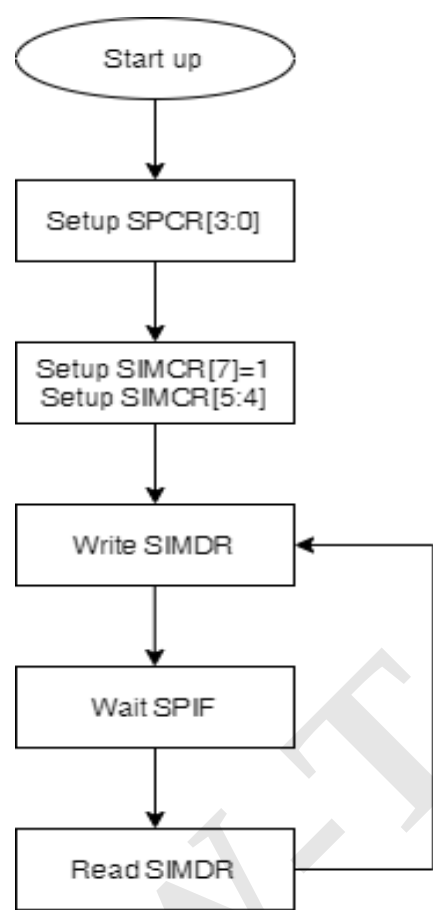


图 46 显示了CPOL和CKEG位如何影响时钟/数据定时

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。



3.19.5 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SPDR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SPDR失败。

3.20 IIC模式

AT8BM84A通过写SIMCR[7:6]=11进入IIC模式，IIC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。

IIC模式适用于需要在多个设备之间进行短距离通信的应用。

IIC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

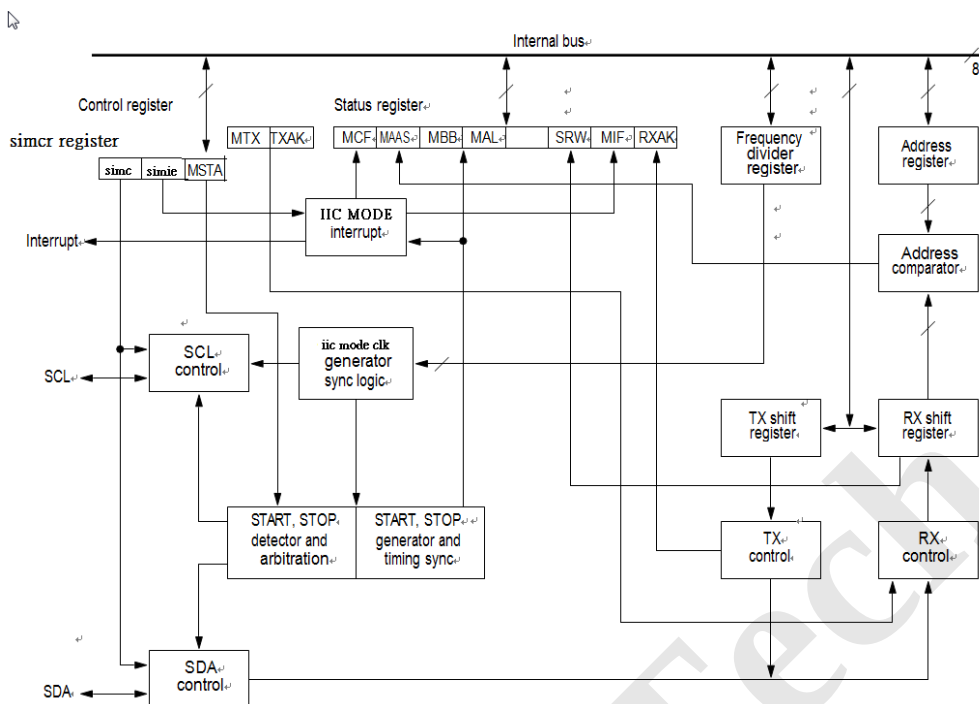
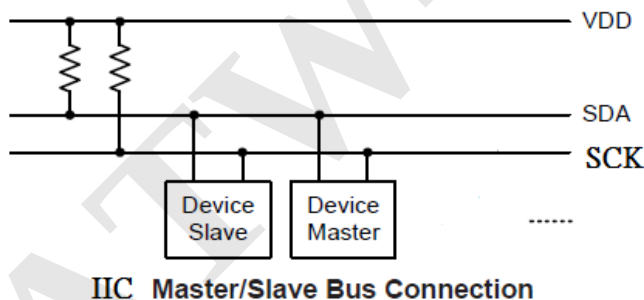


图 47 IIC模式功能框图

下图为IIC操作示例。一台主设备和一台从设备。SDA和SCK需要拉高电阻。



3.20.1 IIC 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:

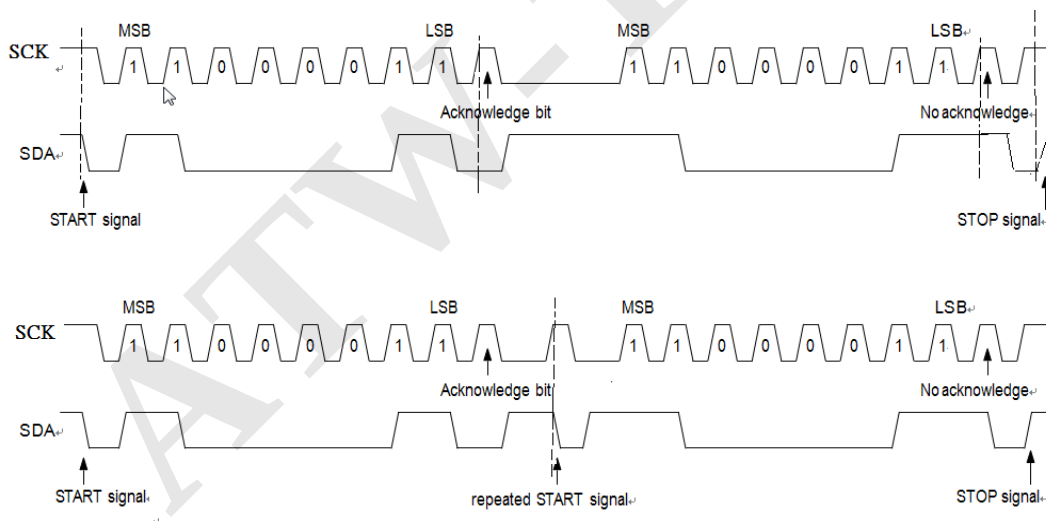


图 48 SDA/SCK示意图

3.20.2 IIC 模式操作

AT8BM84A将通过写入SIMCR[7:6]=11进入IIC模式。当SIMCR[5](MSTA)=1时，设备进入主机模式；当SIMCR[5](MSTA)=0时，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，IIC 模式设置为发送模式，否则MCR[4]=0，IIC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个字节，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，否则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，读入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：

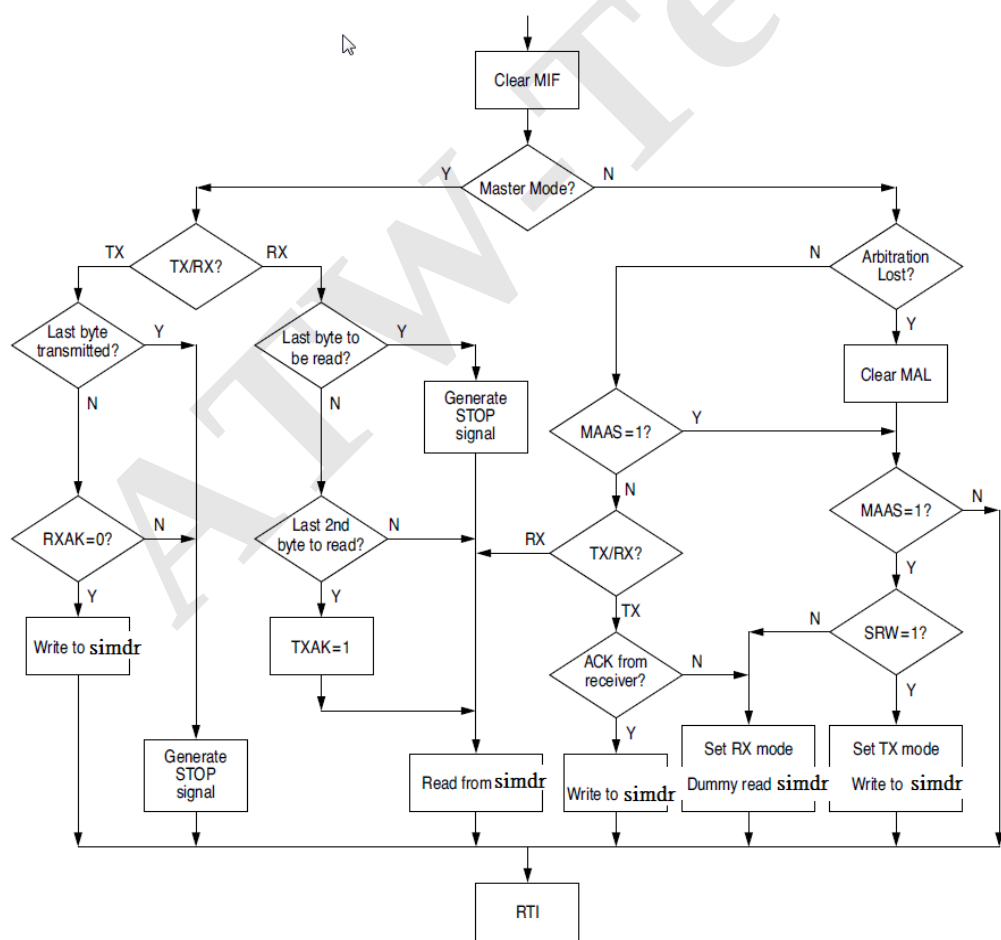


图 49 IIC模式设置流程

3.20.3 MADR（IIC 模式地址寄存器）

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

MAD1-MAD7：IIC的从机地址。

3.20.4 MFDR（IIC 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

FD0-FD4：IIC时钟分频选择，IIC时钟频率等于CPU时钟除以选择的分频。

例如：CPU 时钟=1MHZ，FD[4:0]=2，IIC时钟等于1MHZ/28=35.7KHZ。

FD4	FD3	FD2	FD1	FD0	DIVIDER
0	0	0	0	0	22
0	0	0	0	1	24
0	0	0	1	0	28
0	0	0	1	1	34
0	0	1	0	0	44
0	0	1	0	1	48
0	0	1	1	0	56
0	0	1	1	1	68
0	1	0	0	0	88
0	1	0	0	1	96
0	1	0	1	0	112
0	1	0	1	1	136
0	1	1	0	0	176
0	1	1	0	1	192
0	1	1	1	0	224
0	1	1	1	1	272

FD4	FD3	FD2	FD1	FD0	DIVIDER
1	0	0	0	0	352
1	0	0	0	1	384
1	0	0	1	0	448
1	0	0	1	1	544
1	0	1	0	0	704
1	0	1	0	1	768
1	0	1	1	0	896
1	0	1	1	1	1088
1	1	0	0	0	1408
1	1	0	0	1	1536
1	1	0	1	0	1792
1	1	0	1	1	2176
1	1	1	0	0	2816
1	1	1	0	1	3072
1	1	1	1	0	3584
1	1	1	1	1	4352

3.20.5 MCR (IIC 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

MTX: IIC传输模式与接收模式选择。

1 = 传输模式。

0 = 接收模式。

TXAK: IIC TX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。

3.20.6 MSR (IIC 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

MIF和MAL 需要软件清除，其它位为只读。

MCF: 资料传输完成。

1 = 数据传输完成。

0 = 正在传输数据。

MAAS: 从机地址。

1 = 当前为从机地址。

0 = 没有从机地址。

设置MAAS时，也会设置MIF（IIC模式中断）位。

如果MAAS=1，则CPU需要检查SRW位并相应地设置其MTX位。

MBB: Bus Busy。

1 = 忙。

0 = 空闲。

当检测到开始信号，MBB被置1，当接收到结束信号，MBB被清0。

MAL: 仲裁。

1 = 主机模式总线仲裁失败。

0 = 主机模式总线仲裁成功。

当IIC模式主机仲裁失败时，MAL被置1。

在主传输模式下，置MAL1时，MIF也会置1，必须由软件清除。

SRW: 读/写选择。

1 = 呼叫主机，向从机写数据。

0 = 呼叫主机，向从机读数据。

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IIC模式控制寄存器的MTX位来选择从发送/接收模式。

MIF: IIC 中断。

1 = IIC 发生中断。

0 = IIC 没有发生中断。

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。

2) 从机模式下，主机呼叫地址与从机地址匹配。

3) 总线仲裁失败。

该位必须由中断程序中的软件清除。

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。

3.20.7 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。

3.21 通用异步收发器（UART）

- 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 16X时钟。
 - 错误检测。
 - 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字符。

控制寄存器用于指定数据通信格式。可以通过写入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。

3.21.1 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
初始值		X	X	X	X	X	X	X	X

URD7~URD0: 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 接收到的数据。

3.22 LCD

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 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

COM PORT	SFR PxCOM
PC3	P5COM
PC4	P6COM
PC5	P7COM

表 40 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。

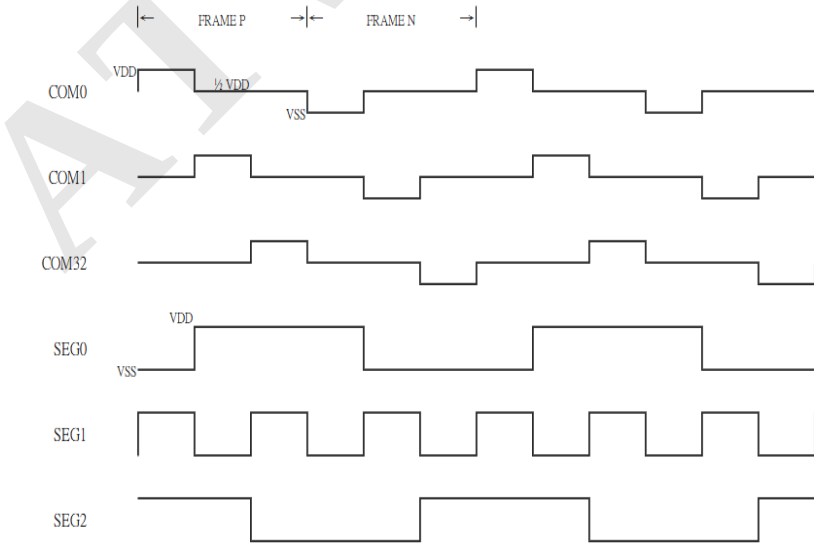


图 50 3 个COM的 LCD波形。

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：

3.22.1 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	0	0

LCDEN: LCD功能开启。
LCDEN=1, LCD功能开启。
LCDEN=0, LCD功能关闭。

LCDRS[1:0]: ½ VDD电流选择。
LCDRS[1:0]=00, 电流为 8uA。
LCDRS[1:0]=01, 电流为 16uA。
LCDRS[1:0]=10, 电流为 50uA。
LCDRS[1:0]=11, 电流为 100uA。

3.22.2 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	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

PxCOM: PxCOM PAD ½ LCD偏置选择， $0 \leq x \leq 7$ 。

P0COM=1，PB4 输出½ LCD。

P1COM=1，PB5 输出½ LCD。

P2COM=1，PB6 输出½ LCD。

P3COM=1，PB7 输出½ LCD。

P4COM=1，PC2 输出½ LCD。

P5COM=1，PC3 输出½ LCD。

P6COM=1，PC4 输出½ LCD。

P7COM=1，PC5 输出½ LCD。

PxCOM=0，PxPAD是一般I/O脚， $0 \leq x \leq 7$ 。

注意：当PxCOM=1 时，对应的PAD输出缓冲区将自动关闭。

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=1 mode=nc output data=1/2 VDD	PxCOM=1 mode=nc output data=1/2 VDD
SEG	mode=output output=VSS	mode=output output=VDD	mode=output output=VDD	mode=output output data=VSS

3.23 片上仿真（OCD）

3.23.1 功能描述

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系统使用两线串行接口SCL和SDA，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/输出管脚，SCL是用于与SDA同步的输入管脚。AT8BM84A还使用SCL和SDA作为控制引脚来写入和读取MTP。

3.23.2 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：当系统处于暂停模式时，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。读访问可能返回垃圾，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停止状态。

3.24 系统内程序（ISP）

3.24.1 功能描述

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，从而缩短了上市时间。

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，确保在编程过程中数据同步。
- **ISPVPP：**VPP 输入引脚，高电平。
- **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。

3.24.2 ISP限制

引脚配置：为了避免编程过程中的电流泄漏，ISPSDA 和 ISPSCK 引脚必须配置为输入 I/O 口上拉。

预编程：ISP 不支持未编程芯片，避免 ISP 引脚状态未知损坏系统。芯片在放入电路板之前必须使用 ICP（In-Circuit Programming）进行预编程，这意味着空白芯片应该使用 ICP 进行编程。

3.24.3 ISP 流程

1. PC0/PC1：不强制输入。当进入 ISP 模式时，通知程序将 PC1/PC0 改为输入引脚。
2. A 版本 ISP：重置后输入 IIC 正常密码。（CPU 停止）

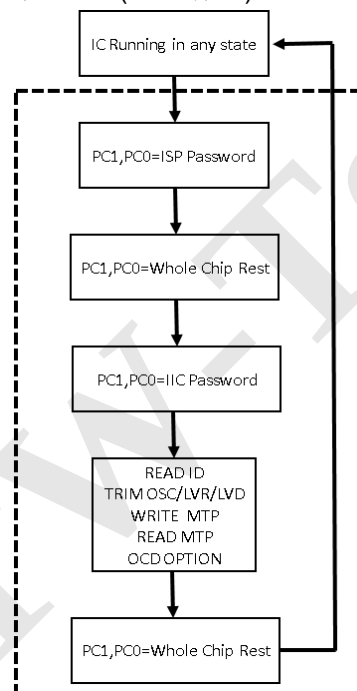


图 51 ISP 模式设置流程

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

表 38 指令设置

ACC: 累加器。

adr: 地址。

bit: R-page中 8 位寄存器的位地址。

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: 半字节进位/借位标记。

dest: 目标。

i: 8 位立即数。

PC: 程序计数器。

PCHBUF: 程序计数器的高字节。

/PD: 睡眠标志位。

/PD=1, 上电或CLRWDWT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 0x0~0x1FF。

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 CLRWDWT 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
语法	ADCAR R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$R + ACC + C \rightarrow dest$
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"。
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.

ADCIA	Add ACC and Immediate with Carry
语法	ADCIA i
操作数	$0 < i < 255$
操作	$ACC + i + C \rightarrow ACC$
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。
周期	1
举例	ADCIA i 执行指令前： ACC=0x12, i=0x34, C=1, 执行指令后： ACC=0x47, C=0.

ADDAR	Add ACC and R
语法	ADDAR R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$ACC + R \rightarrow dest$
状态影响	Z, DC, C
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
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.

ADDIA	Add ACC and Immediate
语法	ADDIA i
操作数	$0 < i < 255$
操作	$ACC + i \rightarrow ACC$
状态影响	Z, DC, C
说明	ACC和8位立即数加法，结果存入ACC。
周期	1
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
语法	ANDAR R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	ACC & R → dest
状态影响	Z
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”。
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.

ANDIA	AND Immediate with ACC
语法	ANDIA i
操作数	$0 \leq i < 255$
操作	ACC & I → ACC
状态影响	Z
说明	ACC和8位立即数做“AND”运算。 结果存入ACC。
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.

BCR	Clear Bit in R
语法	BCR R, bit
操作数	$0 \leq R \leq 511$ $0 \leq \text{bit} \leq 7$
操作	$0 \rightarrow R[\text{bit}]$
状态影响	--
说明	将R寄存器的bit位清0。
周期	1
举例	BCR R, B2 执行指令前: R=0x5A, B2=0x3, 执行指令后: R=0x52.

BSR	Set Bit in R
语法	BSR R, bit
操作数	$0 \leq R \leq 511$ $0 \leq \text{bit} \leq 7$
操作	$1 \rightarrow R[\text{bit}]$
状态影响	--
说明	设置R寄存器的bit位为1。
周期	1
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		BTRSS		Test Bit in R and Skip if Set	
语法		BTRSC	R, bit	语法		BTRSS	R, bit
操作数		$0 \leq R \leq 511$ $0 \leq \text{bit} \leq 7$		操作数		$0 \leq R \leq 511$ $0 \leq \text{bit} \leq 7$	
操作		Skip next instruction, if R[bit] = 0.		操作		Skip next instruction, if R[bit] = 1.	
状态影响		--		状态影响		--	
说明		位判断指令，为“0”则跳过下一条指令。		说明		位判断指令，为“1”则跳过下一条指令。	
周期		1 or 2(skip)		周期		1 or 2(skip)	
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		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。	

CLRA		Clear ACC	
语法		CLRA	
操作数		--	
操作		00h→ACC 1 →Z	
状态影响		Z	
说明		ACC清零，Z标志位置“1”。	
周期		1	
举例		CLRA 执行指令前： ACC=0x55, Z=0. 执行指令后： ACC=0x00, Z=1.	

CLRR	Clear R
语法	CLRR R
操作数	$0 \leq R \leq 511$
操作	00h $\rightarrow$ R 1 $\rightarrow$ Z
状态影响	Z
说明	寄存器R清零, Z标志位置“1”。
周期	1
举例	CLRR R 执行指令前: R=0x55, Z=0. 执行指令后: R=0x00, Z=1.

COMR	Complement R
语法	COMR R, d
操作数	$0 \leq R \leq 511$ d = 0, 1.
操作	$\sim R \rightarrow \text{dest}$
状态影响	Z
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。
周期	1
举例	COMR, d 执行指令前: R=0xA6, d=1, Z=0. 执行指令后: R=0x59, Z=0.

CLRWDT	Clear Watch-Dog Timer
语法	CLRWDT
操作数	--
操作	00h $\rightarrow$ WDT, 00h $\rightarrow$ WDT prescaler 1 $\rightarrow$ /TO 1 $\rightarrow$ /PD
状态影响	/TO, /PD
说明	清WDT计数器和预分频器; /TO和 /PD标志位置“1”。
周期	1
举例	CLRWDT 执行指令前: /TO=0 执行指令后: /TO=1

DECR	Decrease R
语法	DECR R, d
操作数	$0 \leq R \leq 511$ d = 0, 1.
操作	R - 1 $\rightarrow$ dest
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。
周期	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.

DECRSZ	Decrease R, Skip if 0
语法	DECRSZ R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$R - 1 \rightarrow \text{dest},$ Skip if result = 0
状态影响	--
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 or 2(skip)
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
语法	INCR R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$R + 1 \rightarrow \text{dest}.$
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。
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.

INCRSZ	Increase R, Skip if 0
语法	INCRSZ R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$R + 1 \rightarrow \text{dest},$ Skip if result = 0
状态影响	--
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）。
周期	1 or 2(skip)
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。

IORAR	OR ACC with R
语法	IORAR R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$\text{ACC} R \rightarrow \text{dest}$
状态影响	Z
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
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.

IORIA		OR Immediate with ACC		LGOTO		Unconditional Branch	
语法		IORIA i		语法		LGOTO adr	
操作数		$0 \leq i \leq 255$		操作数		$0 \leq \text{adr} \leq 4095$	
操作		$\text{ACC} \mid i \rightarrow \text{ACC}$		操作		$\text{adr} \rightarrow \text{PC}[11:0]$	
状态影响		Z		状态影响		--	
说明		ACC和 8 位立即数做“OR”运算，结果存入ACC。		说明		无条件长跳转，12 位立即数写入 PC[11:0]。	
周期		1		周期		2	
举例		IORIA i 执行指令前： $i=0x50, \text{ACC}=0xAA, Z=0$. 执行指令后： $\text{ACC}=0xFA, Z=0$.		举例		LGOTO Level 执行指令前： $\text{PC}=A0$. 执行指令后： $\text{PC}=\text{address of Level}$.	

LCALL		Call Subroutine		MOVAR		Move ACC to R	
语法		LCALL adr		语法		MOVAR R	
操作数		$0 \leq \text{adr} \leq 4095$		操作数		$0 \leq R \leq 511$	
操作		$\text{PC} + 1$ Top of Stack, $\text{adr} \rightarrow \text{PC}[11:0]$		操作		$\text{ACC} \rightarrow R$	
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-page寄存器	
周期		2		周期		1	
举例		LCALL SUB 执行指令前： $\text{PC}=A0, \text{Stack level}=1$ 执行指令后： $\text{PC}=\text{address of SUB}, \text{Stack}[1]=A0+1, \text{Stack pointer}=2$.		举例		MOVAR R 执行指令前： $R=0x55, \text{ACC}=0xAA$. 执行指令后： $R=0xAA, \text{ACC}=0xAA$.	

MOVIA Move Immediate to ACC

语法	MOVIA i
操作数	$0 < i < 255$
操作	$i \rightarrow \text{ACC}$
状态影响	--
说明	8 位立即数赋值给ACC。
周期	1
举例	MOVIA i 执行指令前: $i=0x55$, $\text{ACC}=0xAA$. 执行指令后: $\text{ACC}=0x55$.

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: $\text{PC}=A0$ 执行指令后: $\text{PC}=A0+1$

MOVR Move R to ACC or R

语法	MOVR R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1$.
操作	$R \rightarrow \text{dest}$
状态影响	Z
说明	R-page寄存器赋值给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。
周期	1
举例	MOVR R, d 执行指令前: $R=0x0$, $\text{ACC}=0xAA$, $Z=0$, $d=0$. 执行指令后: $R=0x0$, $\text{ACC}=0x00$, $Z=1$.

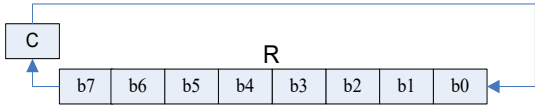
RETIE Return from Interrupt and Enable Interrupt Globally

语法	RETIE
操作数	--
操作	Top of Stack $\rightarrow$ PC $1 \rightarrow \text{GIE}$
状态影响	--
说明	中断返回, 栈顶地址载入PC同时使能中断。
周期	2
举例	RETIE 执行指令前: $\text{GIE}=0$, Stack level=2. 执行指令后: $\text{GIE}=1$, $\text{PC}=\text{Stack}[2]$, Stack pointer=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 $\rightarrow \text{PC}$
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.

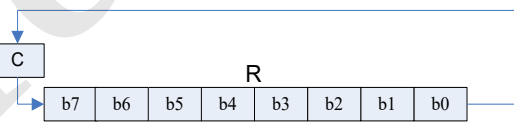
RET	Return from Subroutine
语法	RET
操作数	--
操作	Top of Stack $\rightarrow \text{PC}$
状态影响	--
说明	子程序返回，栈顶载入PC。
周期	2
举例	RET 执行指令前： Stack level=2. 执行指令后： PC=Stack[2], Stack level=1.

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	C
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.

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	C
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.

SBCAR	Subtract ACC and Carry from R
语法	SBCAR R, d
操作数	$0 \leq R \leq 511$ $d = 0, 1.$
操作	$R + (\sim\text{ACC}) + C \rightarrow \text{dest}$
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。
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)

SBCIA	Subtract ACC and Carry from Immediate
语法	SBCIA i
操作数	$0 \leq i < 255$
操作	$i + (\sim\text{ACC}) + C \rightarrow \text{dest}$
状态影响	Z, DC, C
说明	常数和ACC带借位减法, 结果存入ACC。

周期

1

举例

SBCIA i

(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**

语法

SLEEP

操作数

--

操作

00h→WDT,
00h→WDT prescaler
1→/TO
0→/PD

状态影响

/TO, /PD

说明

WDT和分频器 0 清零。/TO标志为0, /PD清零, IC进入睡眠。

周期

1

举例

SLEEP

执行指令前:

/PD=1, /TO=0.

执行指令后:

/PD=0, /TO=1.

SUBAR		SWAPR	
Subtract ACC from R		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 → dest	操作	R[3:0] → dest[7:4]. R[7:4] → 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.

SUBIA		TABLEA	
Subtract ACC from Immediate		Read ROM data	
语法	SUBIA i	语法	TABLEA
操作数	$0 \leq i < 255$	操作数	--
操作	i - ACC → ACC	操作	ROM data{TBHP, ACC} [7:0] → ACC ROM data{TBHP, ACC} [15:8] → TBHD.
状态影响	Z, DC, C	状态影响	--
说明	8 位立即数减 ACC，结果存入 ACC。	说明	ROM 查表指令，高字节存入 TBHD，低字节存入 ACC。
周期	1	周期	2
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)	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.

XORAR	Exclusive-OR ACC with R	XORIA	Exclusive-OR Immediate with ACC
语法	XORAR R, d	语法	XORIA i
操作数	$0 \leq R \leq 511$ $d = 0, 1.$	操作数	$0 \leq i < 255$
操作	$ACC \oplus R \rightarrow dest$	操作	$ACC \oplus i \rightarrow ACC$
状态影响	Z	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。	说明	ACC和 8 位立即数做“XOR”运算。
周期	1	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.	举例	XORIA i 执行指令前: i=0xA5, ACC=0xF0. 执行指令后: ACC=0x55.

5. 配置字节表

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. $6\text{MHz} < F_{\text{HOSC}} \leq 8\text{MHz}$ 3. $10\text{MHz} < F_{\text{HOSC}} \leq 12\text{MHz}$ 5. $16\text{MHz} < F_{\text{HOSC}} \leq 24\text{MHz}$	2. $8\text{MHz} < F_{\text{HOSC}} \leq 10\text{MHz}$ 4. $12\text{MHz} < F_{\text{HOSC}} \leq 16\text{MHz}$			
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. EXCKI0	2. Low Oscillator (I_LRC/E_LXT) 低频振荡器（_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 输出脚	1. PB1	2. PA3			
12	Startup Time 上电复位时间	1. 500us	2. 4.5ms	3. 18ms	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. 1.8V 6. 3.0V	2. 2.0V 7. 3.3V	3. 2.2V 8. 3.6V	4. 2.4V	5. 2.7V
17	VDD Voltage VDD 电压	1. 3.0V	2. 4.5V	3. 5.0V		
18	EX VHREF	1. PB1	2. PA0			
19	Timer1 toggle output 定时器 1 切换输出	1. Enable	2. Disable			
20	Sink current type PA3-0 Sink 电流型态 PA3~0	1. Normal	2. Large	3. Ultra	4. Super	
21	Sink current type Sink 电流型态	1. Normal	2. Large	3. Ultra		
22	Comparator Input pin select 比较器输入引脚选择	1. Enable	2. Disable			

项目	名称	选项			
23	Read Output Data 读取输出口数据	1. I/O Port 一般 I/O	2. Register 寄存器		
24	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		
25	EXCKI0 to Inst. Clock EXCKI0 到指令时钟	1. Sync 同步	2. Async 异步		
26	Startup Clock 上电时钟源	1. Fast (I_HRC/E_HXT/E_XT) 高速	2. Slow (I_LRC/E_LXT) 慢速		
27	Input High Voltage (V _{IH}) 输入高电压 (V _{IH})	1. (0.7VDD)	2. (0.5VDD)		
28	Input Low Voltage (V _{IL}) 输入低电压 (V _{IL})	1. (0.3VDD)	2. (0.2VDD)		
29	External interrupt 0 外部中断 0	1. PB4	2. PB0 or PA4		
30	External interrupt 1 外部中断 1	1. PB1	2. PA3		
31	External interrupt 2 外部中断 2	1. None	2. PA5		
32	UART port UART 端口	1. PB7 / PB6	2. PB0 / PB1		
33	IIC port IIC 端口	1. PB4 / PB5	2. PC3 / PC2		
34	Timer 4/5 external clock 定时器 4/5 外部时钟	1. PA1	2. PA2		
35	PWM1 pad PWM1 输出脚	1. PA3	2. PA5	3. PB1	4. PB5
36	PWM2 pad PWM2 输出脚	1. PA7	2. PA4	3. PB4	4. PB7
37	PWM3 pad PWM3 输出脚	1. PA2	2. PA6	3. PB0	4. PB6
38	PWM4 pad PWM4 输出脚	1. PA1	2. PB3	3. PC1	4. PC3
39	PWM5 pad PWM5 输出脚	1. PA0	2. PB2	3. PC0	4. PC2 or PC5
40	CCP1 dead band pair CCP1 死区对	1. PB2 / PA5	2. PC4 / PA1		
41	Comparator P2P pair 0 (P/N) 比较器 P2P 对 0	1. PA4 / PB3	2. PA0 / PA1		
42	PWM2_PWM4_PAD	1. PA4 / PB3	2. PB3 / PA4		
43	PWM5_PAD	1. PC2	2. PC5		

6. 电气特性

6.1 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

6.2 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	--	--	V	
		5V	3.5	--	--	V	其它所有I/O引脚, EXCKI0/1, INT0/1/2 (0.7 V_{DD})
		3V	2.1	--	--	V	
		5V	2.5	--	--	V	其它所有I/O引脚, EXCKI0/1 (0.5 V_{DD})
		3V	1.5	--	--	V	
V_{IL}	输入低电平	5V	--	--	1.0	V	RSTb (0.2 V_{DD})
		3V	--	--	0.6	V	
		5V	--	--	1.5	V	其它所有I/O引脚, EXCKI0/1, INT0/1/2 (0.3 V_{DD})
		3V	--	--	0.9	V	
		5V	--	--	1.0	V	其它所有I/O引脚, EXCKI0/1 (0.2 V_{DD})
		3V	--	--	0.6	V	
I_{OH}	输出高推电流 (一般电流)	5V	--	19	--	mA	$V_{OH}=4.0V$
		3V	--	11	--	mA	$V_{OH}=2.0V$
I_{OH}	输出高推电流 (大电流)	5V	--	32	--	mA	$V_{OH}=4.0V$
		3V	--	20	--	mA	$V_{OH}=2.0V$
I_{OH}	输出高推电流 (极大电流)	5V	--	45	--	mA	$V_{OH}=4.0V, @PA0\sim PA3$
		3V	--	30	--	mA	$V_{OH}=2.0V, @PA0\sim PA3$
I_{OL}	输出低灌电流 (一般电流)	5V	--	28	--	mA	$V_{OL}=1.0V$
		3V	--	17	--	mA	
I_{OL}	输出低灌电流 (大电流)	5V	--	40	--	mA	$V_{OL}=1.0V$
		3V	--	25	--	mA	
I_{OL}	输出低灌电流 (极大电流)	5V	--	70	--	mA	$V_{OL}=1.0V$
		3V	--	45	--	mA	

符号	参数	V _{DD}	最小值	典型值	最大值	单位	条件
I _{OL}	输出低灌电流 (超大电流)	5V	--	90	--	mA	V _{OL} =1.0V, @PA0~PA3
		3V	--	62	--		
I _{OP}	工作电流	正常模式					
		5V	--	3.1	--	mA	F _{INST} =12MHz @ I _{HRC} : 24MHz/2T @E _{HXT} : 24MHz/2T
		3V	--	1.9	--		
		5V	--	2.7	--	mA	F _{INST} =6MHz @ I _{HRC} : 24MHz/4T @ E _{HXT} : 24MHz/4T
		3V	--	1.7	--		
		5V	--	2.9	--	mA	F _{INST} =8MHz @ I _{HRC} : 16MHz/2T @ E _{HXT} : 16MHz/2T
		3V	--	1.7	--		
		5V	--	2.0	--	mA	F _{INST} =4MHz @ I _{HRC} : 16MHz/4T @ E _{HXT} : 16MHz/4T
		3V	--	1.3	--		
		5V	--	1.4	--	mA	F _{INST} =2MHz @ I _{HRC} : 8MHz/4T @ E _{HXT} : 8MHz/4T
		3V	--	1.0	--		
		5V	--	1.0	--	mA	F _{INST} =1MHz @ I _{HRC} : 4MHz/4T @ E _{XT} : 4MHz/4T
		3V	--	0.8	--		
		慢速模式					
		5V	--	9	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /2
		3V	--	6	--		
		5V	--	6	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /4
		3V	--	4	--		
I _{STB}	待机电流	5V	--	3.5	--	uA	待机模式, F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /4
		3V	--	1.7	--		
I _{HALT}	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关启, LVR 关闭
		3V	--	--	1.5		
R _{PH}	上拉电阻	5V	--	58	--	KΩ	上拉电阻 (不含PA5)
		3V	--	107	--		
		5V	--	90	--	KΩ	上拉电阻 (含PA5)
		3V	--	90	--		
R _{PL}	下拉电阻	5V	--	60	--	KΩ	下拉电阻
		3V	--	120	--		

6.3 比较器/ LVD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_{IVR}	比较器输入电压范围	0	--	5	V	$F_{HOSC}=1MHz$
T_{ENO}	比较器启动等待时间	--	20	--	μS	$F_{HOSC}=1MHz$
I_{CO}	比较器电流消耗	--	135	--	μA	$F_{HOSC}=1MHz$, P2V mode
I_{LVD}	LVD电流消耗	--	150	--	μA	$F_{HOSC}=1MHz$, LVD=4.15V

6.4 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)			± 1	%	用转接座直接从烧录器烧录。
I_HRC 偏移量 (通过夹具烧录)			± 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)			± 5	%	

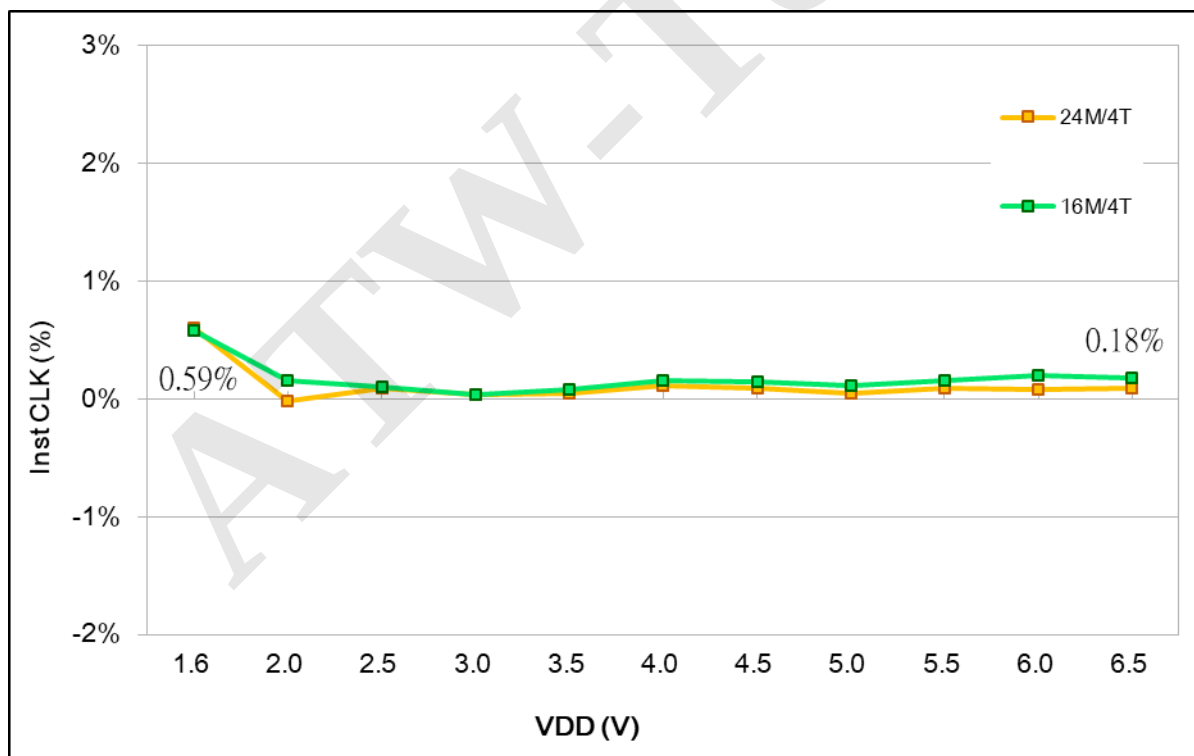
6.5 ADC电气特性

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

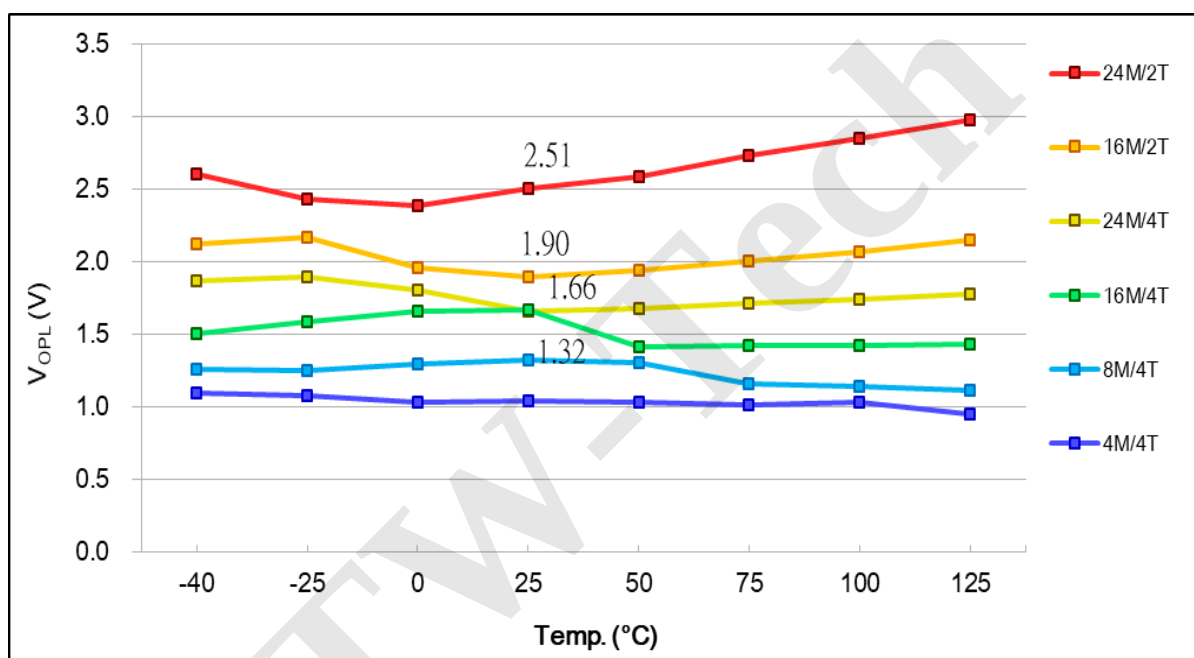
符号	参数	最小值	典型值	最大值	单位	条件
V_{REFH}	模拟参考电压范围	2V	--	V_{DD}	V	Ext. reference voltage
V_{REF4}	内部参考电压 4V, $V_{DD}=5V$	3.95	4	4.05	V	
V_{REF3}	内部参考电压 3V, $V_{DD}=5V$	2.95	3	3.05	V	
V_{REF2}	内部参考电压 2V, $V_{DD}=5V$	1.95	2	2.05	V	
V_{REF}	内部参考电压 V_{DD} , $V_{DD}=5V$	--	V_{DD}	--	V	
	内部参考电压	$V_{REF}+0.5$	--	--	V	Minimum supply voltage
	ADC模拟输入电压	0	--	V_{REFH}	V	
	ADC开启时间	256	--	--	μs	Ready to start convert after set ADENB="1".
$I_{OP(ADC)}$	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_{sample}	ADC转换率	--	--	125	K/sec	$V_{DD}=5V$
DNL	非线性微分误差	± 1	--	--	LSB	$V_{DD}=5.0V$, $AV_{REFH}=5V$, $FADSMP=62.5K$
INL	非线性积分误差	± 2	--	--	LSB	
NMC	无缺码分辨率	10	11	12	Bits	

6.6 特性曲线图

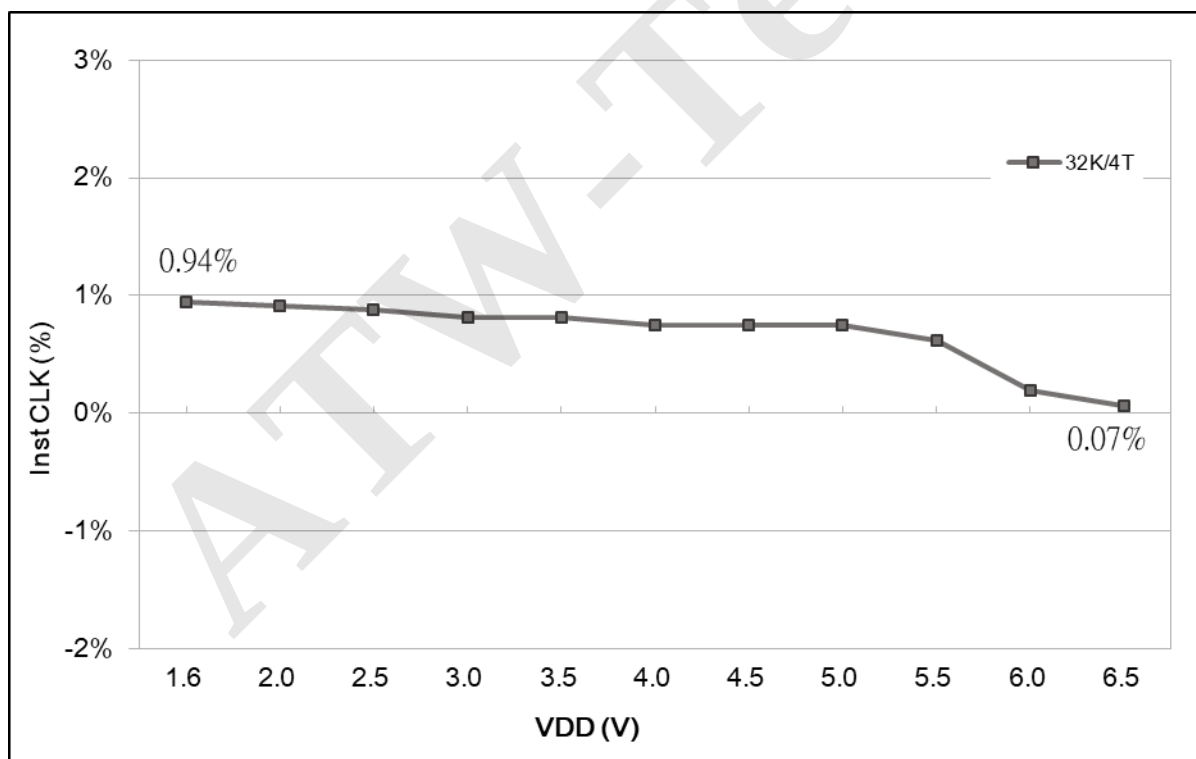
6.6.1 I_HRC 的频率与 VDD 曲线图



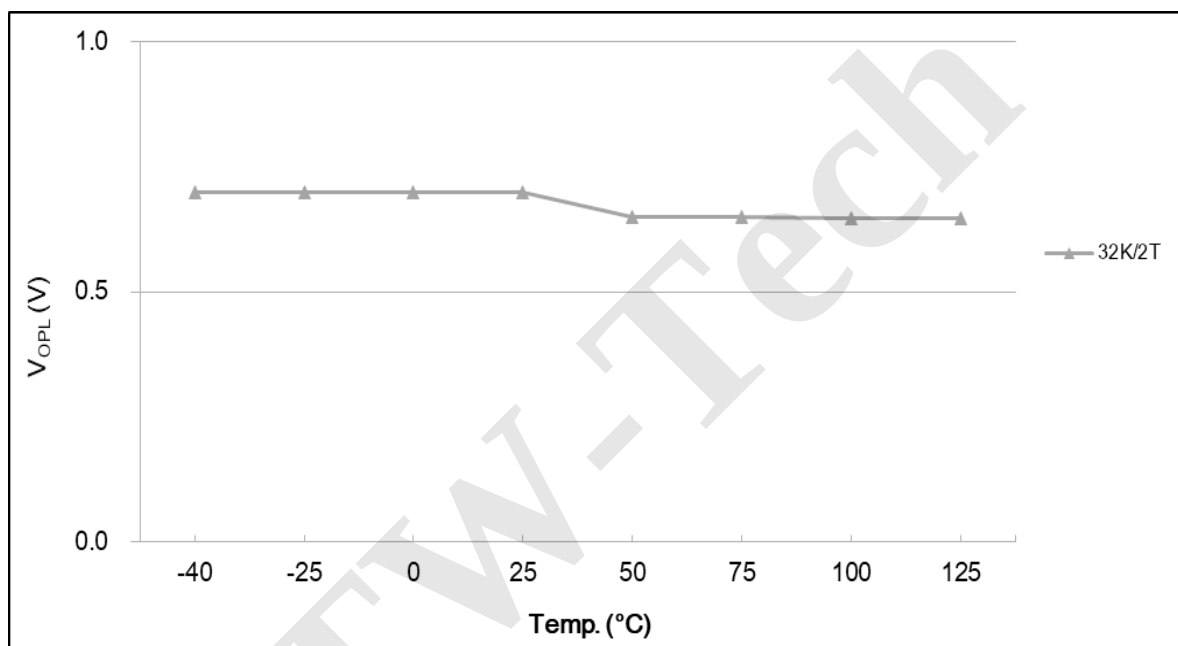
6.6.2 I_HRC 的频率与温度曲线图



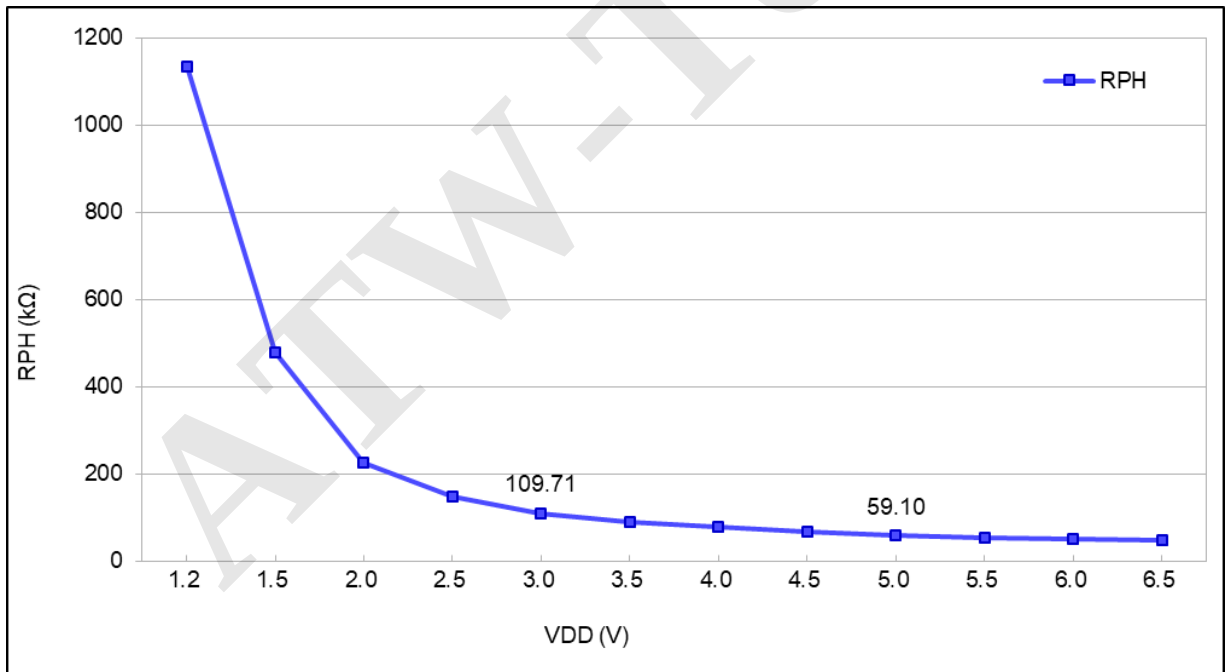
6.6.3 I_LRC 的频率与 VDD 曲线图



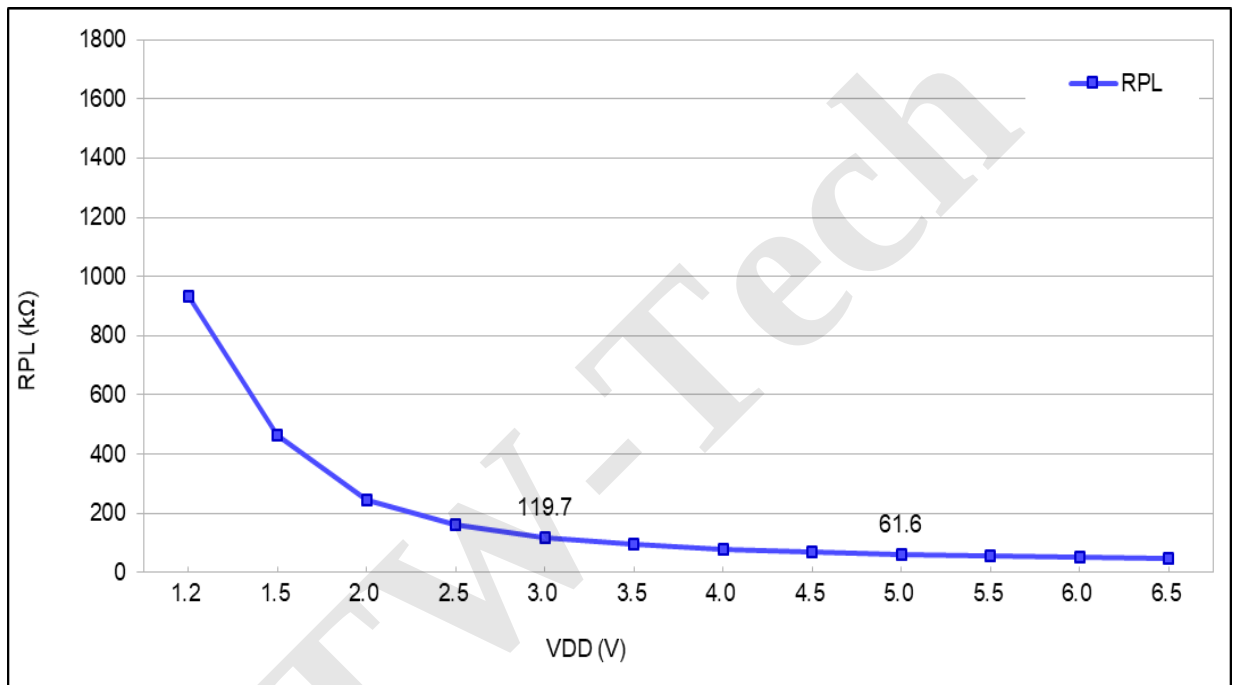
6.6.4 I_LRC 的频率与温度曲线图



6.6.5 上拉电阻与 VDD 曲线图



6.6.6 下拉电阻与 VDD 曲线图



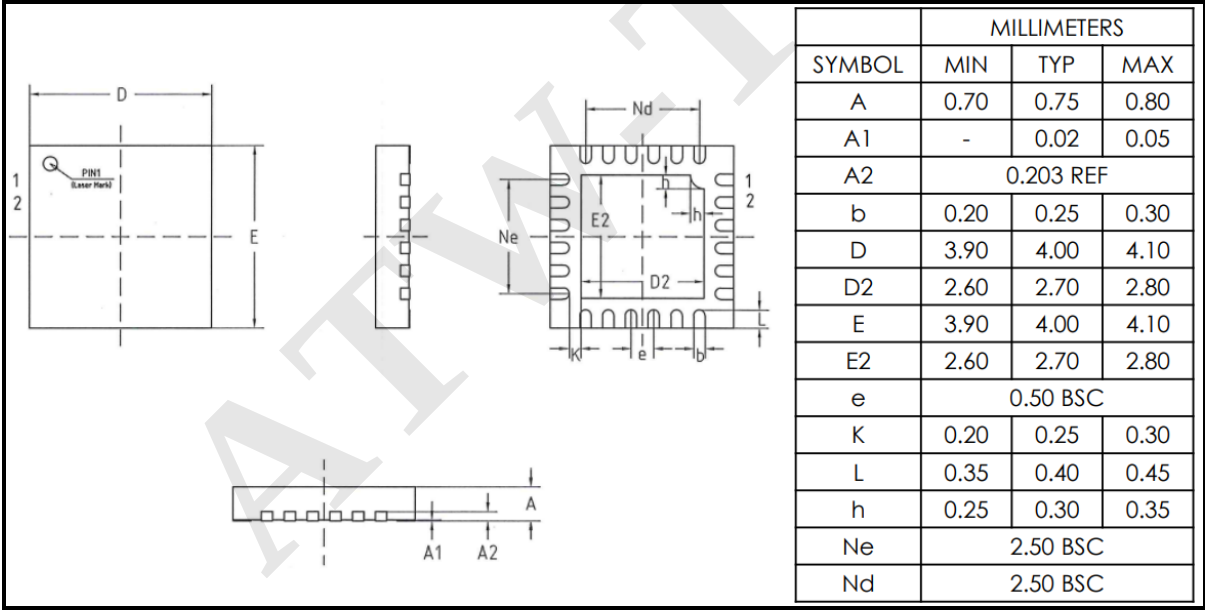
6.7 建议工作电压

建议工作电压（温度范围：-40°C ~ +85°C）

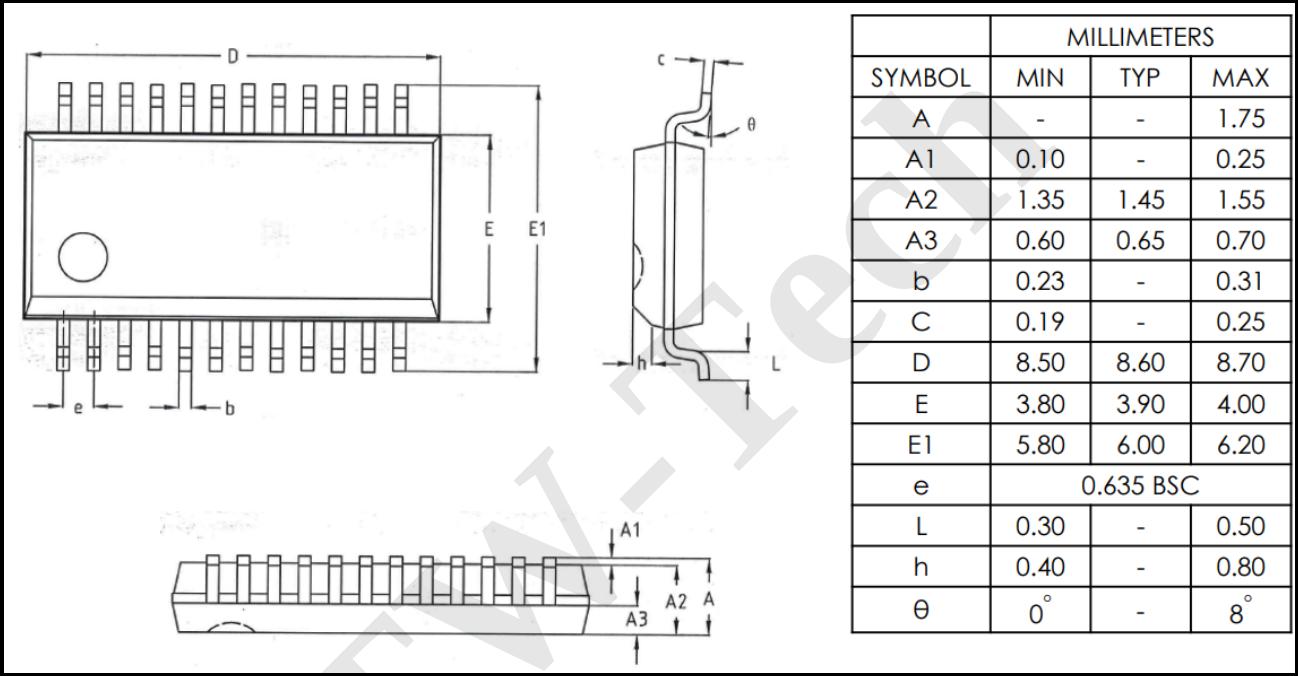
频率	最小工作电压	最大工作电压	LVR 設定	LVR 設定: 默认值 (25°C)
24M/2T	3.0V	5.5V	3.6V	3.3V
16M/2T	2.4V	5.5V	3.3V	3.0V
24M/4T	2.0V	5.5V	2.7V	2.4V
16M/4T	2.0V	5.5V	2.4V	2.2V
8M (2T or 4T)	1.6V	5.5V	1.8V	1.8V
4M (2T or 4T)	1.6V	5.5V	1.8V	1.8V

7. 封装尺寸

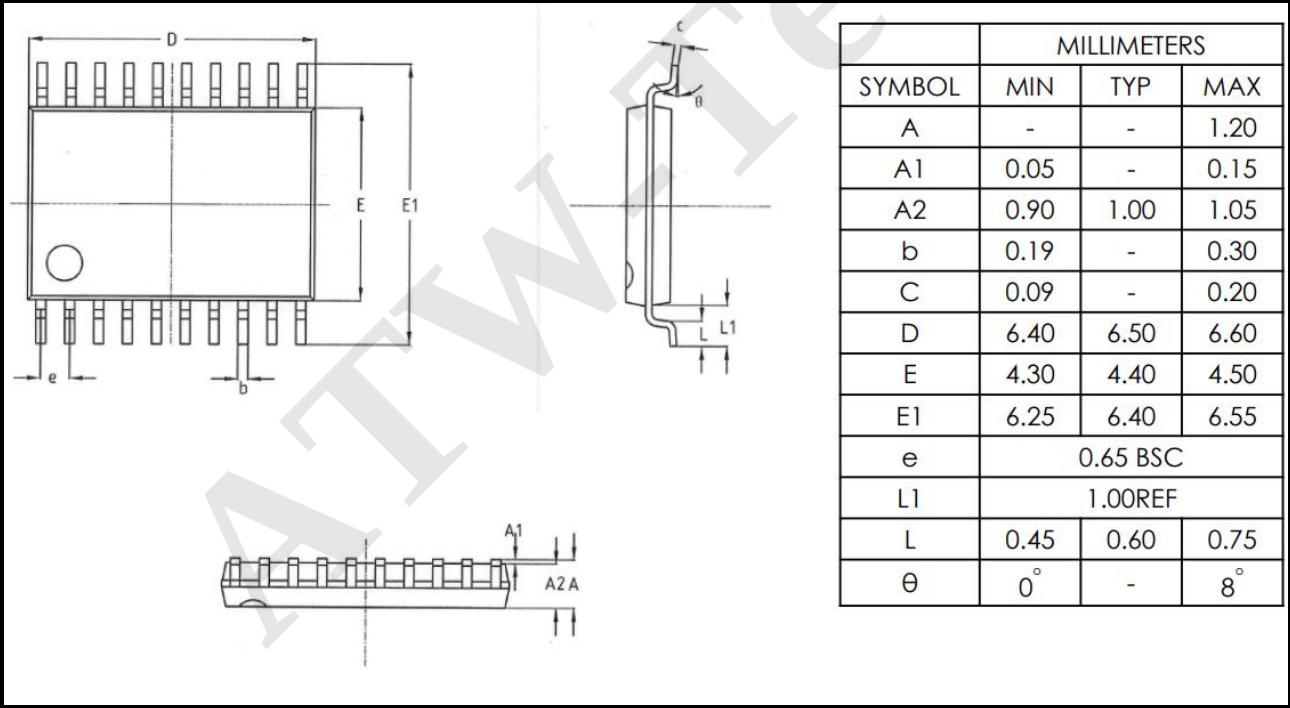
7.1 24 引脚 QFN



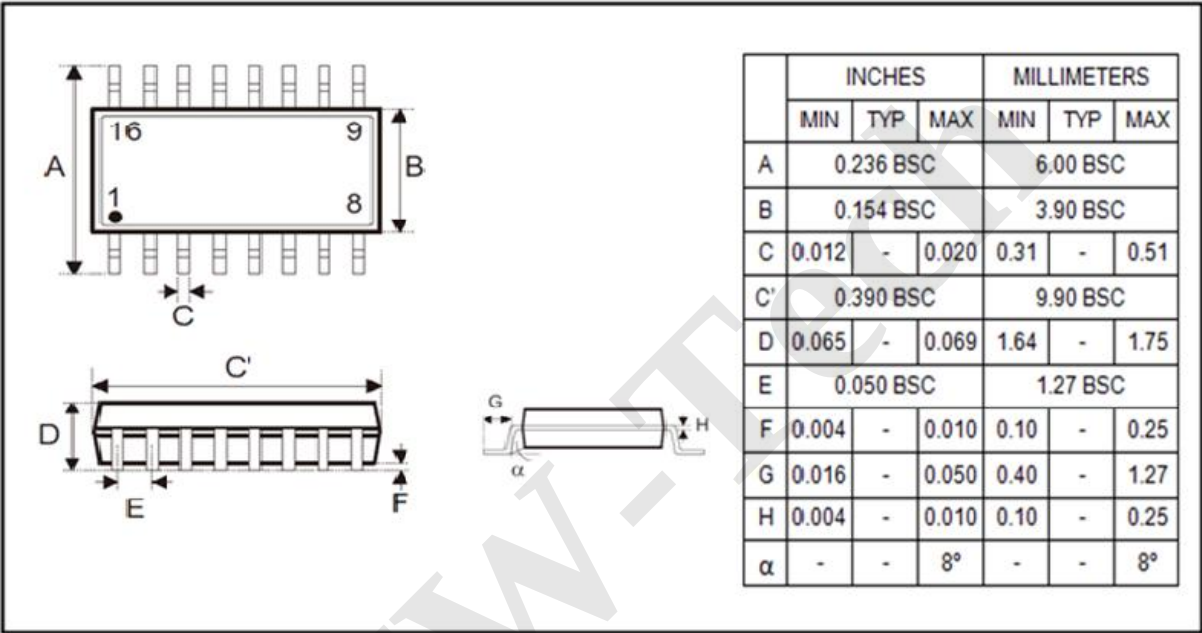
7.2 24 引脚 SSOP



7.3 20 引脚 TSSOP



7.4 16 引脚 SOP (150 毫寸)



8. 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。