

DATA SHEET

BJ8M302A

8 位 MTP 型 MCU

产品功能特色

增强型 8051 内核 CPU:

- 基于 1T8051CPU 内核
- 基于 Keil C 优化的 C 编译器架构
- 完全兼容 MCS-51 指令

灵活的振荡器结构:

- 内建 16M/8M/2M/1MHz 系统振荡器
- 频率误差 ±1%
- 实时时钟 RTCC
- 可外挂 32K-16M 晶体振荡器
- 可外挂 RC 振荡电路
- 系统震荡频率可选
- 内建 WCK 振荡器(WDT 专用)
- 内建高速 TCK 振荡器(Timer1/Timer2 模块时钟专用)

单片机特性:

- 工作电压范围: 2.0V~5.5V
- 工作温度范围: -40°C ~ +85°C
- ROM (MTP): 8K*8
- RAM: 内部: 256*8 ; 外部: 256*8
- 高耐用性闪存 (E2PROM): 128*8
- 堆栈: 8 位堆栈指针, 初始值为 0x07
- 可在软件控制下自编程
- 上电复位 (Power-on Reset, POR)
- 可编程低功耗欠压复位: 1.6V (默认)、1.8V、2.3V、2.7V、3.0V、3.3V、3.6V、3.9V
- 可编程低电压检测 (LVD): 2.2V、2.4V、2.6V、2.9V、3.2V、3.5V、3.8V、4.1V
- 扩展型看门狗定时器 (Watch-Dog Timer, WDT)
- 通过 4 线引脚(包含 VDD/VSS)进行在线串行编程
- 可通过 2 个引脚进行在线调试
- 支持程序对 MTP 或者 EEPROM 的自编程功能
- 节能休眠模式 IDLE 和 STOP

低功耗管理:

- STOP 模式: 0.8uA @1.8V
- STOP 模式: 0.9uA @3.3V
- STOP 模式: 1.1uA @5.0V
- RTC 模式: 6uA @5.0V

模拟特性:

- 模数转换器 (ADC):
- 12 位分辨率
- 16 通道 ADC
- 内部集成 VDD/GND/(VDD/4)基准电压
- 可选择外部参考电压作为基准
- 可软件配置转换时钟频率
- 可软件配置输入信号采样时间
- 1 组放大器 (OPA)
- 3 组比较器 (CMP)
- 1 组客户软件配置的 BEEZ

外设特性:

- 中断:
- 20 个中断入口地址, 4 层中断优先级, 可软件配置
- 外部中断(所有 IO 口都可作为外部中断)
- MAX 26 个双向 I/O 端口, 且均单独软件配置上拉功能:
- IO 口灌/拉电流: 15 mA/15 mA
- 4 个高驱动 IO 口输出高达 80mA
- 所有 IO 口都可软件配置唤醒功能
- 可单独编程的电平变化中断
- P1 和 P2 可软件配置 OC 门输入输出
- P1 和 P2 可软件配置下拉输入输出
- Timer0:
- 带有 8 位预分频器的 8 位定时器/计数器
- 可程序配置为 RTCC 使用
- 增强型 Timer1
- TIMERCLK 最快可达到 48MHz:
- 带有预分频器的 16 位定时器/计数器
- 外部门控输入模式可作为定时/计数器
- 可软件配置为 4 通道捕获功能
- T1PWM 位宽 16 位
- 可软件配置 4 路 T1PWM 边沿对齐模式输出
- 可软件配置 4 路 T1PWM 无互补波形输出
- 增强型 Timer2
- TIMERCLK 最快可达到 48MHz:
- 可软件配置为 3 通道捕获功能
- T2PWM 位宽 16 位
- 可软件配置互补输出
- 可软件配置死区时间调节
- 可软件配置中心对齐模式
- 可软件配置边沿对齐模式
- 可软件配置马达专用输出刹车功能

通讯接口:

- 1 通道 SPI
- 1 通道 UART
- 1 通道 I2C

软件 LCD 功能:

- 任意 IO(P0.0 除外)可作为 COM 或者 SEG 口
- 可软件配置 1/2 电压
- 可软件配置 1/3bias 偏电压

特殊硬件单元

- 霍尔传感器
- 快充 (QC3.0) 模块
- 16 位 × 16 位乘法
- 32 位 ÷ 16 位除法
- 16 位 ÷ 16 位除法
- 32 位左/右移位

抗干扰能力:

- ESD: 优于 ±4000V
- EFT: 优于 ±4000V

目 录

产品功能特色.....	2
1.产品概述.....	7
1.1 封装图.....	7
1.2 引脚说明.....	11
2.中央处理器.....	13
2.1 程序存储器 ROM(MTP).....	13
2.2 数据存储器 (SRAM).....	15
2.3 特殊寄存器 (SFR).....	16
2.3.1 SFR 地址映射.....	16
2.2.2 SFR 特殊寄存器汇总.....	17
2.2.3 CPU 内核常用 SFR.....	20
3.复位系统.....	22
3.1 上电复位 (POR).....	22
3.2 外部 RST 复位.....	24
3.3 低电压复位.....	25
3.4 看门狗复位(WDT).....	25
4.时钟系统.....	25
4.1 IRC 模式.....	27
4.2 ERC 模式.....	27
4.3 HXT 模式.....	27
4.4 LXT 模式.....	27
4.4DSYSCK 模式.....	27
4.5 WCK 模式.....	27
4.6 外部时钟输入接口参考电路.....	27
5.省电(低功耗)模式.....	28
5.1 IDLE 模式.....	28
5.2 STOP 模式.....	28
6.看门狗 (WDT).....	30
6.1 相关寄存器.....	30
6.2 WDT 功能选择.....	32
6.3 WDT 操作步骤.....	33
7. I/O 端口.....	33

7.1P0 结构框图.....	34
7.2P1 结构框图.....	35
7.3P2 结构框图.....	36
7.4P3 结构框图.....	36
7.5I/O 翻转换醒 CPU.....	37
7.6 高电流驱动 IO 口.....	37
8.低电压检测.....	39
9.中断.....	40
9.1 中断入口向量表.....	40
9.2 中断源中断向量.....	41
9.3 中断原理框图.....	42
9.4 中断优先级.....	43
9.5 中断相关的寄存器.....	44
9.6 外部中断.....	46
10.TIMER0 模块.....	49
10.1 T0 工作原理.....	49
10.2 T0 定时器.....	50
10.3T0 计数器.....	51
11.TIMER1 模块.....	53
11.1T1 定时计数器工作原理.....	53
11.2 T1 捕获功能(4 路).....	55
11.3 T1PWM 脉宽调制.....	59
12.TIMER2 模块.....	62
12.1T2 定时计数器.....	62
12.2 T2 捕获功能(3 路).....	64
12.3T2 PWM(3 路/含刹车).....	67
12.3.1 边沿对齐模式输出.....	71
12.3.2PWM 互补输出与死区.....	71
12.3.3PWM 中心对齐.....	73
12.3.4PWM 刹车.....	74
13. 数模转换(ADC)模块.....	76
13.1 ADC 转换模块框图.....	76
13.2ADC 转换步骤.....	79
14. 串行外设(SPI)接口.....	81
14.1 信号描述.....	82
14.2 工作模式.....	83
14.3SPSTA 异常情况.....	85
15. 通用串口(UART)接口.....	86

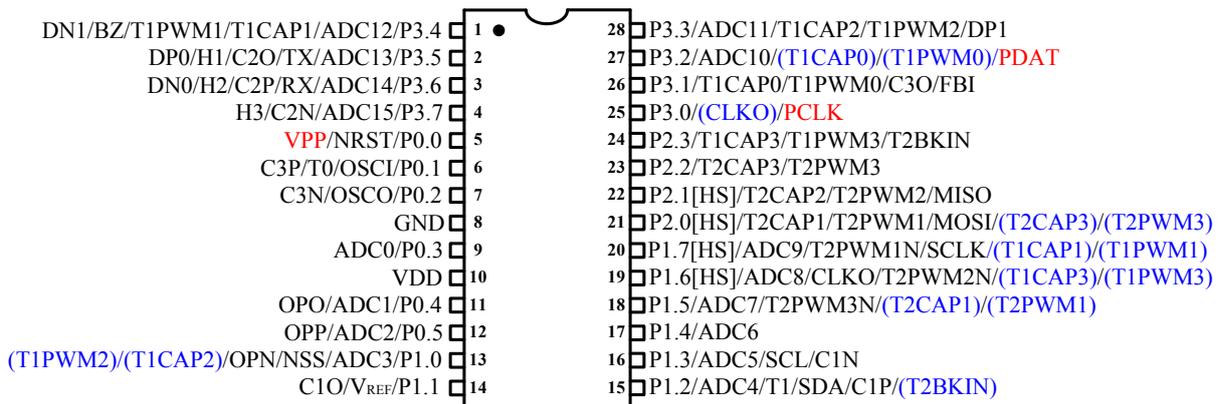
15.1 UART 工作模式	87
15.2 波特率	88
16. 集成电路总线(IIC)接口	90
16.1 IIC 特性	90
16.2 功能描述	91
16.3 主发送器模式	92
16.4 主接收器模式	93
16.5 从发送器模式	97
16.6 从接收器模式	99
17. 运放/比较器	101
17.1 运算放大器	103
17.2 比较器	103
18. 液晶显示(LCD)驱动模块	104
18.1 LCD1/2BIAS 1/4DUTY 应用	105
18.2 LCD1/3BIAS 1/4DUTY 应用	106
19. 实时时钟(RTCC)模块	108
20. 蜂鸣器(BEEZ)	109
21. 霍尔传感器模块	110
21.1 霍尔传感器	110
21.2 马达控制驱动信号	111
21.2 解码后的 PWM 波形	113
22. 快充模块(QC)	116
23. EEPROM&MTP 数据存储	121
23.1 EEPROM 操作描述	122
23.2 MTP 操作描述	123
24. 乘/除法单元	125
24.1 MDSU 运算	126
24.2 移位操作	127
24.3 标准化操作 (NORMALIZING)	127
25. 调试烧录程模块	128
26. 程序配置	129
27. 指令系统	131
27.1 寻址方式	131
27.2 功能命令的说明	133
27.3 读-修改-写指令	136
28. 电气特性	138

29. 开发支持.....	141
30. 封装信息.....	142
31. 产品标识体系.....	145
32. 附录 A: 版本信息&声明	147
33. 读者反馈表.....	148

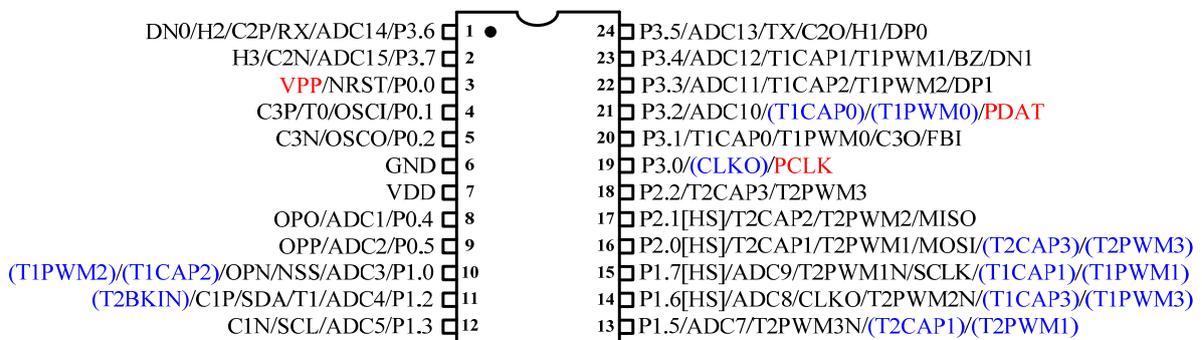
1. 产品概述

BJ8M302A 是一颗增强型的超快速 1T 8051 内核单芯片 8 位工业级 MTP 微控制器，指令系统完全兼容传统 8051 系列产品。

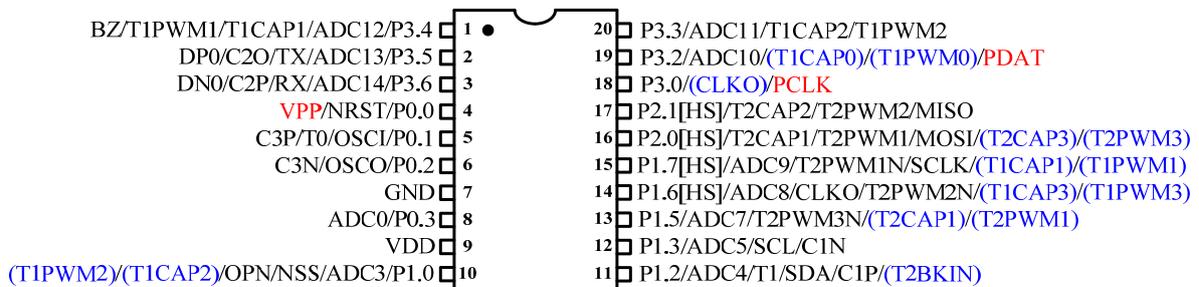
1.1 封装图



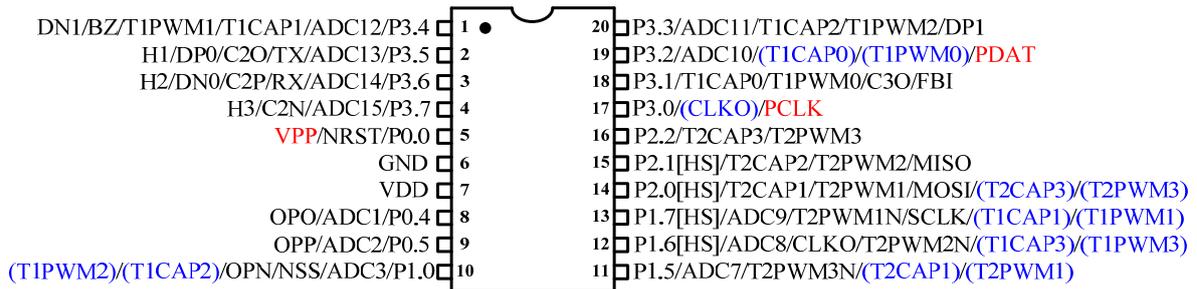
BJ8M302AMJ/BJ8M302ATJ (SOP28/TSSOP28)



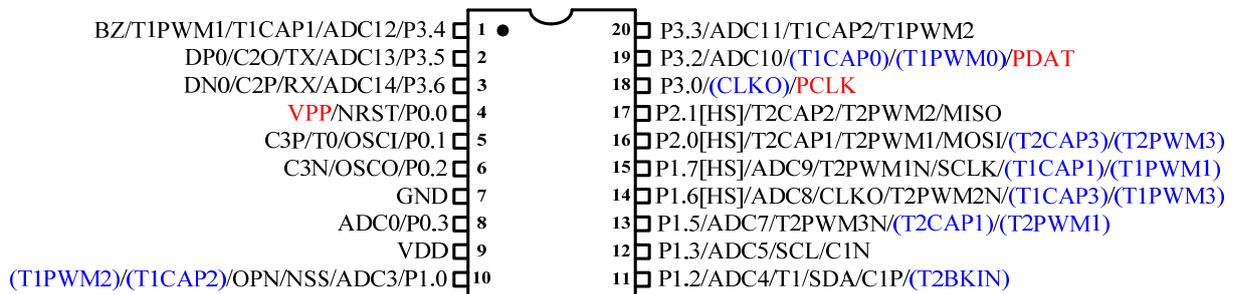
BJ8M302ATH (TSSOP24)



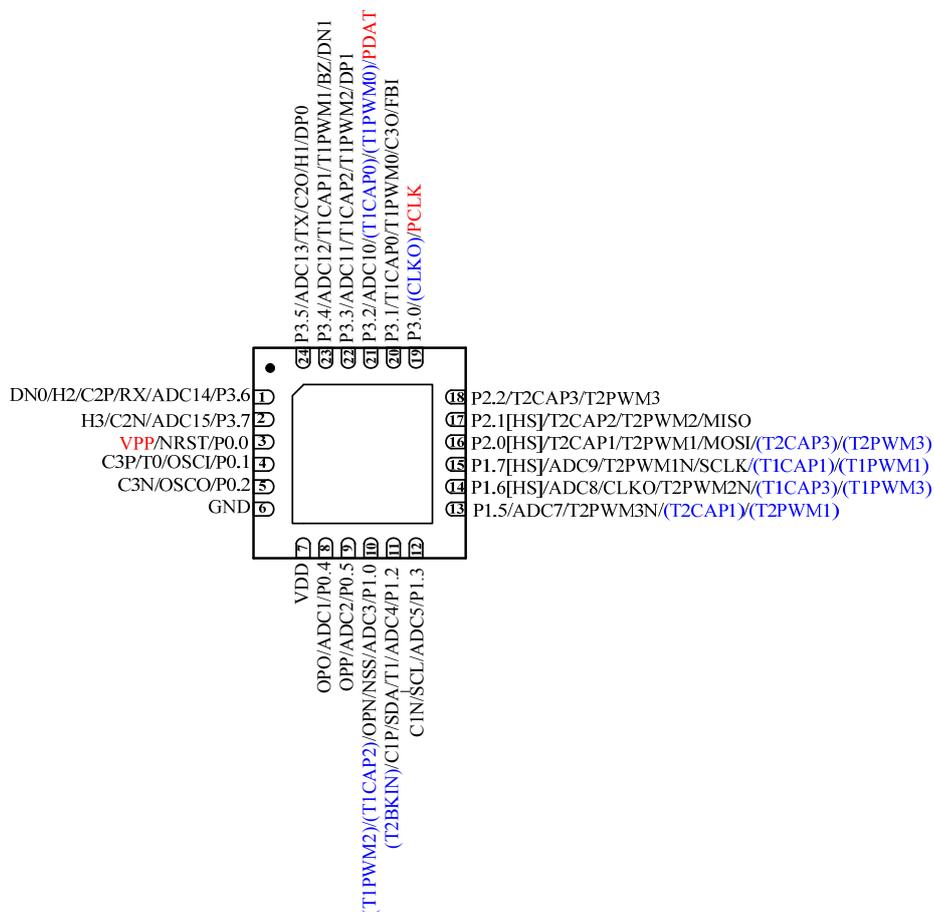
BJ8M302ATG (TSSOP20)



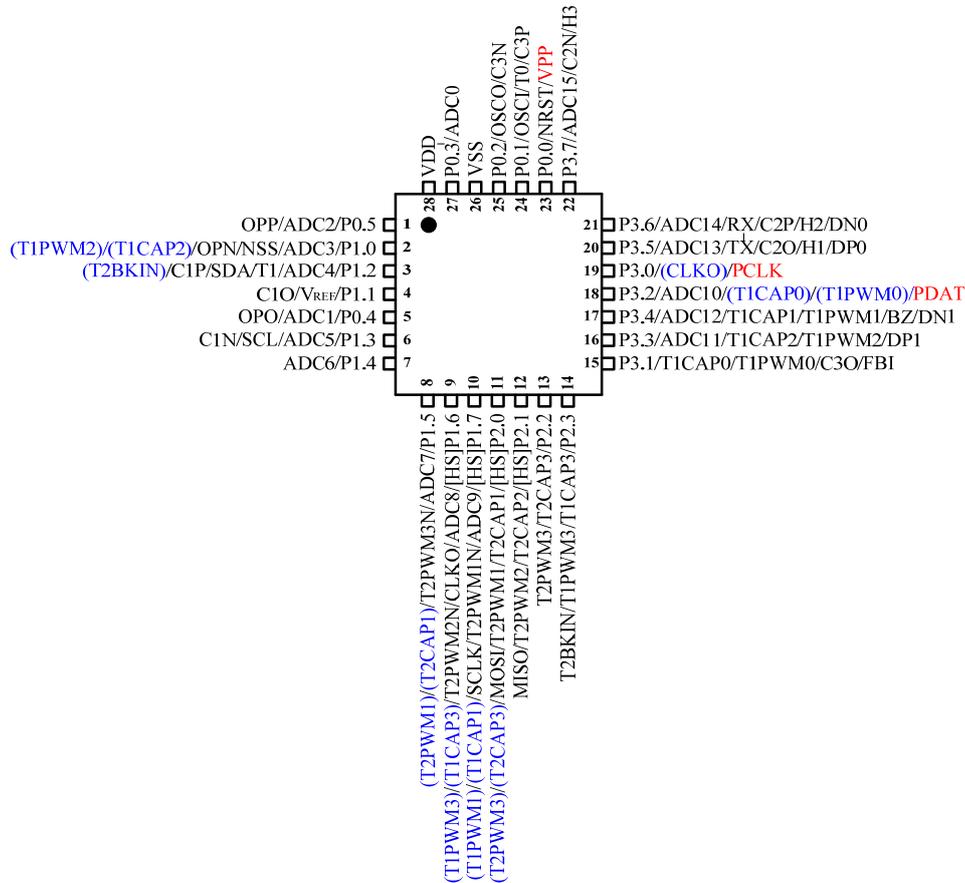
BJ8M302ATG-B(TSSOP20)



BJ8M302AMG (SOP20)



BJ8M302AFH (QFN24)



BJ8M302AFJ-B(QFN28)

引脚名称备注:

- 1.Pn.x[HS] : 表示该 IO 可配置大电流驱动
- 2.T0/T1: Timer n 时钟输入口
- 3.VPP/PDAT/PCLK: 编程管脚
- 4.(XXX): 表示通过配置, XXX 功能可以映射到该管脚输出
- 5.Hn: 霍尔信号输入脚
- 6.OPnP/OPnN/OPnO: 第 n 组放大器正负输入输出(n=0 时, 省略)
- 7.CnP/CnN/CnO: 第 n 组比较器正负输入输出(n=0 时, 省略)
- 8.TnPWMx/TxPWMxN: 为 Tn 控制的一对 PWM 互补波形输出脚
- 9.TnCAPx: Time n 控制的第 x 路捕捉输入引脚
- 10.MISO_n/MOSI_n/SCLK_n/NSS_n: 第 n 组 SPI 通讯口(n=0 时, 省略)
- 11.SCL_n/SDA_n: 第 n 组 I2C 通讯口(n=0 时, 省略)
- 12.TX_n/RX_n : 第 n 组 UART 通讯口(n=0 时, 省略)
- 13.DP1/DN1/FBI: 第 1 组快充接口 (FBI: FeedBack Input)
- 14.DP0/DN0: 第 0 组快充口
- 15.CLKO 为系统时钟输出口, 通过代码配置映射到此引脚输出

1-1 封装管脚位置

序号	28P A	24P B	20P A	20P B	引脚功能
1	5	3	4	5	P0.0/NRST/VPP
2	6	4	5		P0.1/OSCI/T0/C3P
3	7	5	6		P0.2/OSCO/C2N
4	8	6	7	6	GND
5	9		8		P0.3/ADC0
6	10	7	9	7	VDD
7	11	8		8	P0.4/ADC1/OPO
8	12	9		9	P0.5/ADC2/OPP
9	13	10	10	10	P1.0/ADC3/NSS/OPN/(T1CAP2)/(T1PWM2)
10	14				P1.1/V _{REF} /C10
11	15	11	11		P1.2/ADC4/T1/SDA/C1P/(T2BKIN)
12	16	12	12		P1.3/ADC5/SCL/C1N
13	17				P1.4/ADC6
14	18	13	13	11	P1.5/ADC7/T2PWM3N/(T2CAP1)/(T2PWM1)
15	19	14	14	12	P1.6[HS]/ADC8/CLKO/T2PWM2N/(T1CAP3)/(T1PWM3)
16	20	15	15	13	P1.7[HS]/ADC9/T2PWM1N/SCLK/(T1CAP1)/(T1PWM1)
17	21	16	16	14	P2.0[HS]/T2CAP1/T2PWM1/MOSI/(T2CAP3)/(T2PWM3)
18	22	17	17	15	P2.1[HS]/T2CAP2/T2PWM2/MISO
19	23	18		16	P2.2/T2CAP3/T2PWM3
20	24				P2.3/T1CAP3/T1PWM3/T2BKIN
21	25	19	18	17	P3.0/(CLKO)/PCLK
22	26	20		18	P3.1/T1CAP0/T1PWM0/C30/FBI
23	27	21	19	19	P3.2/ADC10/(T1CAP0)/(T1PWM0)/PDAT
24	28	22	20	20	P3.3/ADC11/T1CAP2/T1PWM2/DP1
25	1	23	1	1	P3.4/ADC12/T1CAP1/T1PWM1/BZ/DN1
26	2	24	2	2	P3.5/ADC13/TX/C20/H1/DP0
27	3	1	3	3	P3.6/ADC14/RX/C2P/H2/DN0
28	4	2		4	P3.7/ADC15/C2N/H3

1.2 引脚说明

表 1-2 I/O 引脚说明

引脚名称	功能	类型	功能说明
P0.0/NRST/VPP	P0.0	I/O	通用I/O；内置上拉，具有唤醒中断功能；
	NRST	I	低电平主复位。
	VPP	I	编程电压输入脚
P0.1/OSCI/T0/C3P	P0.1	I/O	通用I/O；内置上拉，具有唤醒中断功能；可配置LCD的SEG口或COM口
	OSCI	I	晶振/谐振器时为振荡信号输入引脚。
	T0	I	Timer0 计数时钟输入。
	C3P	I	第三组比较器正输入引脚 (C:Comarer P: Positeve N:Negative)
P0.2/OSCO/C3N	P0.2	I/O	通用I/O；内置上拉，具有唤醒中断功能；可配置LCD的SEG口或COM口
	OSCO	O	晶振 /谐振器时为振荡信号输出引脚。
	C3N	I	第三组比较器负输入引脚 (C:Comarer P: Positeve N:Negative)
P0.3/ADC0	P0.3	I/O	通用I/O；内置上拉，具有唤醒中断功能；可配置LCD的SEG口或COM口
	ADC0	I	A/D输入通道。
P0.4/ADC1/OPO	P0.4	I/O	通用I/O；内置上拉，具有唤醒中断功能；可配置LCD的SEG口或COM口
	ADC1	I	A/D输入通道。
	OPO	O	放大器输出引脚 (OPO: operational amplifier Output)
P0.5/ADC2/OPP	P0.5	I/O	通用I/O；内置上拉，具有唤醒中断功能；可配置LCD的SEG口或COM口
	ADC2	I	A/D输入通道。
	OPP	I	放大器正端输入引脚 (OP: operational amplifier P: Positeve)
P1.0/ADC3/NSS/OPN/(T1PWM2)/(T1PWM2)	P1.0	I/O	通用I/O；内置上下拉、OC门，具有唤醒中断功能；可配置LCD的SEG口或COM口
	ADC3	I	A/D输入通道。
	NSS	O	SPI通讯：由主器件控制的使能信号
	OPN	I	放大器负端输入引脚 (OP: operational amplifier N:Negative)
	(T1CAP2)	I	Timer 1控制的第三组捕捉功能信号可映射到此引脚输入
(T1PWM2)	O	Timer 1控制的第三组PWM信号可映射到此引脚输出	
P1.1/VREF/C10	P1.1	I/O	通用I/O；内置上下拉、OC门，具有唤醒中断功能；可配置LCD的SEG口或COM口
	VREF	I	A/D参考电压输入。
	C10	O	第一组比较器输出引脚 (C:Comarer 0:Output)
P1.2/ADC4/T1/SDA/C1P/(T2BKIN)	P1.2	I/O	通用I/O；内置上下拉、OC门，具有唤醒中断功能；可配置LCD的SEG口或COM口
	ADC4	I	A/D输入通道。
	T1	I	Timer1 时钟输入。
	SDA	I/O	IIC通讯中的数据输入输出引脚
	C1P	I	第一组比较器正端输入引脚 (C:Comarer P: Positeve N:Negative)
(T2BKIN)	I	Timer 2控制刹车控制信号可映射到此引脚输入	
P1.3/ADC5/SCL/C1N	P1.3	I/O	通用I/O；内置上下拉、OC门，具有唤醒中断功能；可配置LCD的SEG口或COM口
	ADC5	I	A/D输入通道。
	SCL	O	IIC通讯中的时钟信号引脚
C1N	I	第一组比较器负端输入引脚 (C:Comarer P: Positeve N:Negative)	
P1.4/ADC6	P1.4	I/O	通用I/O；内置上下拉、OC门，具有唤醒中断功能；可配置LCD的SEG口或COM口
	ADC6	I	A/D输入通道。
P1.5/ADC7/T2PWM3N/(T2CAP1)/(T2PWM1)	P1.5	I/O	通用I/O；内置上下拉、OC门，具有唤醒中断功能；可配置LCD的SEG口或COM口
	ADC7	I	A/D输入通道。
	T2PWM3N	O	Timer 2控制的第三组PWM互补波形输出信号引脚
	(T2CAP1)	I	Timer 2控制的第一组捕捉功能信号可映射到此引脚输入
(T2PWM1)	O	Timer 2控制的第一组PWM信号可映射到此引脚输出	

表 1-2 I/O 引脚说明 (续)

引脚名称	功能	类型	功能说明
P1.6/ADC8/CLKO /T2PW2N /(T1CAP3) /(T1PWM3)	P1.6[HS]	I/O	通用IO; 内置上下拉、OC门, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	ADC8	I	A/D输入通道。
	CLKO	O	系统时钟输出(需OPTION配置)
	T2PWM2N	O	Timer 2控制的第二组PWM互补波形输出信号引脚
	(T1CAP3)	I	Timer 1控制的第四组捕捉功能信号可映射到此引脚输入
	(T1PWM3)	O	Timer 1控制的第四组PWM信号可映射到此引脚输出
P1.7/ADC9 /T2PWM1N/SCLK /(T1CAP1) /(T1PWM1)	P1.7[HS]	I/O	通用IO; 内置上下拉、OC门, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	ADC9	I	A/D输入通道。
	T2PWM1N	O	Timer 2控制的第一组PWM互补波形输出信号引脚
	SCLK	O	SPI通讯: 主器件产生时钟信号
	(T1CAP1)	I	Timer 1控制的第二组捕捉功能信号可映射到此引脚输入
	(T1PWM1)	O	Timer 1控制的第二组PWM信号可映射到此引脚输出
P2.0/T2CAP1 /T2PWM1/MOSI /(T2CAP3) /(T2PWM3)	P2.0[HS]	I/O	通用IO; 内置上下拉、OC门, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	T2CAP1	I	Timer 2控制的第一组捕捉功能输入信号引脚
	T2PWM1	O	Timer 2控制的第一组PWM波形输出信号引脚
	MOSI	O	SPI通讯: 主器件数据输出, 从器件数据输入引脚
	(T2CAP3)	I	Timer 2控制的第三组捕捉功能信号可映射到此引脚输入
	(T2PWM3)	O	Timer 2控制的第三组PWM信号可映射到此引脚输出
P2.1/T2CAP2 /T2PWM2/MISO	P2.1[HS]	I/O	通用IO; 内置上下拉、OC门, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	T2CAP2	I	Timer 2控制的第二组捕捉功能输入信号引脚
	T2PWM2	O	Timer 2控制的第二组PWM波形输出信号引脚
	MISO	I	SPI通讯: 主器件数据输入, 从器件数据输出
P2.2/T2CAP3 /T2PWM3	P2.2	I/O	通用IO; 内置上下拉、OC门, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	T2CAP3	I	Timer 2控制的第三组捕捉功能输入信号引脚
	T2PWM3	O	Timer 2控制的第三组PWM波形输出信号引脚
P2.3/T1CAP3 /T1PWM3/T2BKIN	P2.3	I/O	通用IO; 内置上下拉、OC门, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	T1CAP3	I	Timer 1控制的第四组捕捉功能输入信号引脚
	T1PWM3	O	Timer 1控制的第四组PWM波形输出信号引脚
	T2BKIN	I	Timer 2控制的刹车信号输入引脚
P3.0/(CLKO) /PCLK	P3.0	I/O	通用IO; 内置上拉, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	(CLKO)	O	系统时钟可映射到此引脚输出
	PCLK	O	串行编程时钟口
P3.1/T1CAP0 /T1PWM0/C30 /FBI	P3.1	I/O	通用IO; 内置上拉, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	T1CAP0	I	Timer 1控制的第一组捕捉功能输入信号引脚
	T1PWM0	O	Timer 1控制的第一组PWM波形输出信号引脚
	C30	O	第三组比较器输出引脚 (C:Comarer 0:Output)
P3.2/ADC10 /(T1CAP0) /(T1PWM0) /PDAT	P3.2	I/O	通用IO; 内置上拉, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	ADC10	I	A/D输入通道。
	(T1CAP0)	I	Timer 1控制的第一组捕捉功能信号可映射到此引脚输入
	(T1PWM0)	O	Timer 1控制的第一组PWM信号可映射到此引脚输出
	PDAT	I/O	串行编程数据口
P3.3/ADC11 /T1CAP2 /T1PWM2/DP1	P3.3	I/O	通用IO; 内置上拉, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	ADC11	I	A/D输入通道。
	T1CAP2	I	Timer 1控制的第三组捕捉功能输入信号引脚
	T1PWM2	O	Timer 1控制的第三组PWM波形输出信号引脚
	DP1	I	QC快充: 第一组USB D+输入端 (P: Positeve)

表 1-2 I/O 引脚说明 (续)

引脚名称	功能	类型	功能说明
P3.4/ADC12 /T1CAP1 /T1PWM1/BZ /DN1	P3.4	I/O	通用I/O; 内置上拉, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	ADC12	I	A/D输入通道。
	T1CAP1	I	Timer 1控制的第二组捕捉功能输入信号引脚
	T1PWM1	O	Timer 1控制的第二组PWM波形输出信号引脚
	BZ	O	1KHz/2KHz/4KHz/8KHz buzzer输出引脚。
	DN1	I	QC快充: 第一组USB D-输入端 (N:Negative)
P3.5/ADC13 /TX/C2O/ H1/DP0	P3.5	I/O	通用I/O; 内置上拉, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	ADC13	I	A/D输入通道。
	TX	O	UART通信中的数据输出引脚
	C2O	I	第二组比较器输出引脚 (C:Comparer O: Output)
	H1	I	霍尔信号输入脚
	DP0	I	QC快充: 第零组USB D+输入端 (P: Positeve)
P3.6/ADC14 /RX/C2P /H2/DN0	P3.6	I/O	通用I/O; 内置上拉, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	ADC14	I	A/D输入通道。
	RX	I	UART通信中的数据输入引脚
	C2P	I	第二组比较器正端输入引脚 (C:Comparer P: Positeve N:Negative)
	H2	I	霍尔信号输入脚
	DN0	I	QC快充: 第零组USB D-输入端 (N:Negative)
P3.7/ADC15 /C2N/H3	P3.7	I/O	通用I/O; 内置上拉, 具有唤醒中断功能; 可配置LCD的SEG口或COM口
	ADC15	I	A/D输入通道。
	C2N	I	第二组比较器负端输入引脚 (C:Comparer P: Positeve N:Negative)
	H3	I	霍尔信号输入脚
VDD	VDD	-	正电源电压
GND	GND	-	接地

2.中央处理器

BJ8M302A 是一款 8 位基于最新 MTP 技术(Multiple-Time Programmable)工艺的 8051 内核 RISC 架构的单片机。它集成了 OTP 低成本、高电压烧录, 同时也集成了 Flash 的多次重复烧录等同优点。

2.1 程序存储器 ROM(MTP)

程序存储器用于存放用户程序、数据等信息, BJ8M302A 单片机内部集成了 8K 字节的 MPT 程序存取空间, 详情如下结构图。其中 0000H-00AFH 是单片机各种中断服务程序入口地址(详情请见: 9. 中断章节的中断向量表); 1FE0H-1FFF 为 程序配置存储空间(详情请见: 23. 程序配置 章节)

图 2-1 程序存储器结构图:



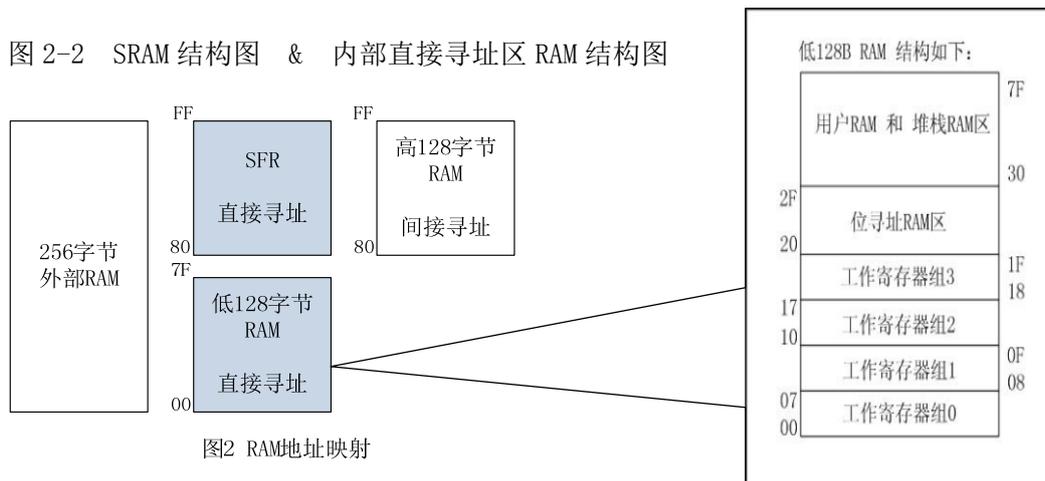
2.2 数据存储器 (SRAM)

BJ8M302A 单片机总共集成了 512 字节的 SRAM，其中由内部 256 字节和外部 256 字节组成。
外部 256 字节 SRAM(地址 00~FF)，只能用 MOVX 指令访问。

内部 256 字节 SRAM 高 128B(地址 80H~FFH)只能用间接寻址的方式访问,低 128B(地址 00H~7FH)可直接寻址的方式访问，也可通过间接寻址的方式来访问。

特殊功能寄存器 SFR 的地址也是 80H~FFH。它的地址与内部 256 字节 SRAM 高 128B 地址相同。
区别是：SFR 寄存器是直接寻址，而高 128B SRAM 只能是间接寻址。

内部 256 字节 SRAM 的低 128B SRAM 区可分为三部分：①工作寄存器组 0~3，地址 00H~1FH, 程序状态字寄存器 PSW 中的 RS0、RS1 组合决定了当前使用的工作寄存器，使用工作寄存器组 0~3 可加快运算的速度；②位寻址区 20H~2FH，此区域用户可以使用普通 RAM 也可用作按位寻址 RAM；按位寻址时，位的地址为 00H~7FH（此地址按位编地址，不同于通用 SRAM 按字节编地址），程序中可由指令区分；③用户 RAM 区 BJ8M302A 复位过后，8 位的堆栈指针 SP 指向堆栈区（默认值为 0x07），用户一般会在初始化程序时设置初值，建议设置在 80H 以后的单元区间。



2.3 特殊寄存器 (SFR)

2.3.1 SFR 地址映射

表 2-1SFR 地址映射

地址	页0	页1	地址	页0	页1	地址	页0	页1	地址	页0	页1
80h	P0	P0	A0h	P2	P2	C0h	T1CON	T1CON	E0h	ACC	ACC
81h	SP	SP	A1h	T2PWMICON	T2PWMICON	C1h	T1OVRL	T1OVRL	E1h	P0OE	P0OE
82h	DP0L	DP0L	A2h	T2BKCON	T2BKCON	C2h	T1OVRH	T1OVRH	E2h	P1OE	P1OE
83h	DP0H	DP0H	A3h	T2CETCON	T2CETCON	C3h	T1CL	T1CL	E3h	P2OE	P2OE
84h	DP1L	DP1L	A4h	T2PWMCON1	T2PWMCON1	C4h	T1CH	T1CH	E4h	P3OE	P3OE
85h	DP1H	DP1H	A5h	T2PWMCON2	T2PWMCON2	C5h	OPACON0	OPACON0	E5h	P1OD	P1OD
86h	OPACON1	OPACON1	A6h	T2PWMDZH	T2PWMDZH	C6h	CMPIMOD	CMPIMOD	E6h	P2OD	P2OD
87h	PCON	PCON	A7h	T2PWMDZL	T2PWMDZL	C7h	LVDCON	LVDCON	E7h	Null	Null
88h	T0CON	T0CON	A8h	IEN0	IEN0	C8h	T2CON	T2CON	E8h	CLKCON	CLKCON
89h	T0OVR	T0OVR	A9h	IOWAKE	IOWAKE	C9h	T2OVRL	T2OVRL	E9h	I2C0SPD	I2C0SPD
8Ah	T0C	T0C	AAh	T2D1H	T2D1H	CAh	T2OVRH	T2OVRH	EAh	I2C0DAT	I2C0DAT
8Bh	P0LCDVO	QCCON0	ABh	T2D1L	T2D1L	CBh	T2CL	T2CL	EBh	I2C0ADR	I2C0ADR
8Ch	P1LCDVO	QCCON1	ACH	T2D2H	T2D2H	CCh	T2CH	T2CH	ECh	I2C0CON	I2C0CON
8Dh	HDCD	HDCD	ADh	T2D2L	T2D2L	CDh	CMP1CON	CMP1CON	EDh	I2C0STA	I2C0STA
8Eh	HDCR	HDCR	AEh	T2D3H	T2D3H	CEh	CMP2CON	CMP2CON	EEh	LCDCON	TRIMICK
8Fh	BZCON	BZCON	AFh	T2D3L	T2D3L	CFh	CMP3CON	CMP3CON	EFh	METCH	METCH
90h	P1	P1	B0h	P3	P3	D0h	PSW	PSW	F0h	B	B
91h	T1PWMCON	T1PWMCON	B1h	T2CAPNUM0	SPI0STA	D1h	T1CAPCON	P0PU	F1h	INFCON	INFCON
92h	T1D0H	T1D0H	B2h	T2CAPNUM1	SPI0CON	D2h	T1CAPNUM0	P1PU	F2h	P0IF	P0IF
93h	T1D0L	T1D0L	B3h	T2CAPNFCNT	SPI0DAT	D3h	T1CAPNUM1	P2PU	F3h	P1IF	P1IF
94h	T1D1H	T1D1H	B4h	P0IE	P0IE	D4h	T1CAPNFCNT	P3PU	F4h	P2IF	P2IF
95h	T1D1L	T1D1L	B5h	P1IE	P1IE	D5h	T2CAPCON	P1PD	F5h	P3IF	P3IF
96h	T1D2H	T1D2H	B6h	P2IE	P2IE	D6h	HDCON	HDCON	F6h	P2LCDVO	IDO
97h	T1D2L	T1D2L	B7h	P3IE	P3IE	D7h	T2CAPINT	T2CAPINT	F7h	P3LCDVO	IUP
98h	S0CON	S0CON	B8h	IEN1	IEN1	D8h	ADCON2	ADCON2	F8h	T1CAPINT	T1CAPINT
99h	S0BUF	S0BUF	B9h	IEN2	IEN2	D9h	ADCON1	ADCON1	F9h	MD0	IAPTIR
9Ah	S0OVR	S0OVR	BAh	IP0	IP0	DAh	ADCON0	ADCON0	FAh	MD1	E2CON
9Bh	S0OVRH	S0OVRH	BBh	IP1	IP1	DBh	ADCHL	ADCHL	FBh	MD2	IAPADRL
9Ch	T1D3H	T1D3H	BCh	PEGCON0	PEGCON0	DCh	ADCHH	ADCHH	FCh	MD3	IAPADRH
9Dh	T1D3L	T1D3L	BDh	CMPNFCNT	CMPNFCNT	DDh	ADDOL	ADDOL	FDh	MD4	IAPDAT
9Eh	T1PWMICON	T1PWMICON	BEh	WDTCON	WDTCON	DEh	ADDOH	ADDOH	FEh	MD5	IAPCON
9Fh	HDFCON	P2PD	BFh	WDTOVR	WDTOVR	DFh	ADDO	ADDO	FFh	ARCON	TRIMTCK

备注:

1. 在黄色背景字体部分寄存器与棕色背景字体部分寄存器物理地址相同,但是在不同的页面地址中。
2. 访问方法:软件配置 PSW[5] (F0) 来选择页。
3. 灰色表示地址尚未实现。

2.2.2 SFR 特殊寄存器汇总

表 2-2SFR 特殊寄存器汇总

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值	
80h	P0	-	-	P0<5:0>						--00 0000	
81h	SP	SP<7:0>									0000 0111
82h	DP0L	DP0L<7:0>									0000 0000
83h	DP0H	DP0H<7:0>									0000 0000
84h	DP1L	DP1L<7:0>									0000 0000
85h	DP1H	DP1H<7:0>									0000 0000
86h	OPACON1	-	-	-	-	OPACRT<3:0>				---- 1000	
87h	PCON	-	-	-	-	-	-	STOP	IDLE	0000 1000	
88h	T0CON	T0CLR	T0M	T0IF	T0EG	T0PSC<2:0>			T0EN	0000 0000	
89h	T0OVR	T0OVR<7:0>									0000 0000
8Ah	T0C	T0C<7:0>									0000 0000
8Bh	P0LCDVO	-	-	PO5VO	PO4VO	PO3VO	PO2VO	PO1VO	-	--00 000-	
8Ch	P1LCDVO	P17VO	P16VO	P15VO	P14VO	P13VO	P12VO	P11VO	P10VO	0000 0000	
8Bh	QCCON0	QCEN	DPOREN	DNOREN	DPODNS	FBEN	VLS0<2:0>			0000 -000	
8Ch	QCCON1	-	DP1REN	DN1REN	-	VLS1P<1:0>		VLS1N<1:0>		0000 0000	
8Dh	HDCD	-	-	-	-	-	SHC	SHB	SHA	---- -000	
8Eh	HDCR	HSEL	-	PWMS	MPMWS	HALS	HDMS	FRS	HDCEN	0-00 0000	
8Fh	BZCON	-	-	-	-	-	BZMOD		BZEN	---- --00	
90h	P1	P1<7:0>									0000 0000
91h	T1PWMCON	T1PWM3S	T1PWM2S	T1PWM1S	T1PWMOS	T1PWM3EN	T1PWM2EN	T1PWM1EN	T1PWMOEN	0000 0000	
92h	T1D0H	T1D0<15:8>									0000 0000
93h	T1D0L	T1D0<7:0>									0000 0000
94h	T1D1H	T1D1<15:8>									0000 0000
95h	T1D1L	T1D1<7:0>									0000 0000
96h	T1D2H	T1D2<15:8>									0000 0000
97h	T1D2L	T1D2<7:0>									0000 0000
98h	S0CON	UARTEN	-	-	UART9	TX9	RX9	TXIF	RXIF	0000 0000	
99h	S0BUF	S0BUF<7:0>									0000 0000
9Ah	S0OVR	S0OVR<7:0>									0000 0000
9Bh	S0OVRH	-	-	-	-	-	S0OVRH<10:8>			---- -000	
9Ch	T1D3H	T1D3<15:8>									0000 0000
9Dh	T1D3L	T1D3<7:0>									0000 0000
9Eh	T1PWM1CON	T1PWM3IF	T1PWM2IF	T1PWM1IF	T1PWM0IF	T1PWM3IE	T1PWM2IE	T1PWM1IE	T1PWMOIE	0000 0000	
9Fh	HDNFCON	HCHKNUM<4:0>					HFRSEL<2:0>				0000 0000
9Fh	P2PD	-	-	-	-	P2PD<3:0>				---- 0000	
A0h	P2	-	-	-	-	P2<3:0>				---- 0000	
A1h	T2PWM1CON	T2PWM3IF	T2PWM2IF	T2PWM1IF	-	T2PWM3IE	T2PWM2IE	T2PWM1IE	-	000- 000-	
A2h	T2BKCON	BKF	BKFLT<1:0>		BKSWT	BKHWSSEL	BKMODE	BKEN	CETEN	0000 0000	
A3h	T2CETCON	CNTIS	-	-	OVRTADS	ZEROTADS	PWMNREN	PWMREN	CNT0IE	0--0 0000	
A4h	T2PWMCON1	T2PWM1NS	T2PWM1S	T2PWMONS	T2PWMOS	T2PWM1NEN	T2PWM1EN	T2PWMONEN	T2PWMOEN	0000 0000	
A5h	T2PWMCON2	T2PWM3NS	T2PWM3S	T2PWM2NS	T2PWM2S	T2PWM3NEN	T2PWM3EN	T2PWM2NEN	T2PWM2EN	0000 0000	
A6h	T2PWMDZH	T2PWMDZ<15:8>									0000 0000
A7h	T2PWMDZL	T2PWMDZ<7:0>									0000 0000
A8h	IEN0	EA	-	-	T2IE	T1IE	P1EA	T0IE	P0EA	0--0 0000	
A9h	IOWAKE	-	-	-	-	-	-	-	IOWU	---- ---0	
AAh	T2D1H	T2D1<15:8>									0000 0000
ABh	T2D1L	T2D1<7:0>									0000 0000
ACh	T2D2H	T2D2<15:8>									0000 0000
ADh	T2D2L	T2D2<7:0>									0000 0000
ACh	T2D3H	T2D3<15:8>									0000 0000
AFh	T2D3L	T2D3<7:0>									0000 0000

备注:

1. 在黄色背景字体部分寄存器与棕色背景字体部分寄存器物理地址相同,但是在不同的页面地址中。
2. 访问方法:软件配置 PSW[5] (F0) 来选择页。
3. 灰色表示地址尚未实现。

表 2-2 SFR 特殊寄存器汇总 (续一)

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值	
B0h	P3	P3<7:0>								0000 0000	
B1h	T2CAPNUM0	T2C1SEL	T2C1NUM<2:0>			-	-	-	-	0000 ----	
B2h	T2CAPNUM1	T2C3SEL	T2C3NUM<2:0>			T2C2SEL	T2C2NUM<2:0>			0000 0000	
B3h	T2CAPNFCNT	T2CAPNFCNT<7:0>								0000 0000	
B1h	SPI0STA	SPIF	WCOL	SSERR	MODF	-	-	-	-	0000 ----	
B2h	SPI0CON	SPR2	SPIEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0	0011 0100	
B3h	SPI0DAT	SPIDAT<7:0>								0000 0000	
B4h	P0IE	-	-	P0IE<5:0>				-	-	--00 0000	
B5h	P1IE	P1IE<7:0>								0000 0000	
B6h	P2IE	-	-	-	-	P2IE<3:0>				---- 0000	
B7h	P3IE	P3IE<7:0>								0000 0000	
B8h	IEN1	-	-	SORXIE	SOTXIE	SPIIE	P3EA	P2EA	I2CIE	--00 0000	
B9h	IEN2	-	LVDIE	CMP3IE	CMP2IE	CMP1IE	ADCIE	-	WDTIE	--00 00-0	
BAh	IP0	-	-	IP0<5:0>					-	-	--00 0000
BBh	IP1	-	-	IP1<5:0>					-	-	--00 0000
BCh	PEGCON0	P3EG<1:0>		P2EG<1:0>		P1EG<1:0>		P0EG<1:0>		0101 0101	
BDh	CMPNFCNT	CMPNUM<4:0>				CMPSEL<2:0>				0000 0000	
BEh	WDTCON	WDTEN	WDTSC<3:0>				WDTFS<1:0>		WDTCLR	0000 0000	
BFh	WDTOVR	WDTOVR<7:0>								1111 1111	
C0h	T1CON	T1CLR	T1M	T1IF	T1EG	T1PSC<2:0>			T1EN	0000 0000	
C1h	T1OVR	T1OVR<7:0>								0000 0000	
C2h	T1OVRH	T1OVR<15:8>								0000 0000	
C3h	T1CL	T1C<7:0>								0000 0000	
C4h	T1CH	T1C<15:8>								0000 0000	
C5h	OPACON0	CMP00	OPAVREF	-	-	OPAFS<2:0>			OPAEN	00-- 0000	
C6h	CMPIMOD	CMP3IMOD<1:0>		CMP2IMOD<1:0>		CMP1IMOD<1:0>		-	-	0000 0000	
C7h	LVDCON	LVDF	LVDIF	-	-	LVDS<2:0>			LVDEN	00-- 0000	
C8h	T2CON	T2CLR	T2M	T2IF	T2EG	T2PSC<2:0>			T2EN	0000 0000	
C9h	T2OVR	T2OVR<7:0>								0000 0000	
CAh	T2OVRH	T2OVR<15:8>								0000 0000	
CBh	T2CL	T2C<7:0>								0000 0000	
CCh	T2CH	T2C<15:8>								0000 0000	
CDh	CMP1CON	CMP10	CMP1HYS	CMP1FS<4:0>					CMP1EN	0000 0000	
CEh	CMP2CON	CMP20	CMP2HYS	CMP2FS<4:0>					CMP2EN	0000 0000	
CFh	CMP3CON	CMP30	CMP3HYS	CMP3FS<4:0>					CMP3EN	0000 0000	
D0h	PSW	CY	AC	F0	RS1	RS0	OV	F1	P	0000 0000	
D1h	T1CAPCON	T1C3MOD<1:0>		T1C2MOD<1:0>		T1C1MOD<1:0>		T1COMOD<1:0>		0000 0000	
D2h	T1CAPNUM0	T1C1SEL	T1C1NUM<2:0>			T1C0SEL	T1CONUM<2:0>			0000 0000	
D3h	T1CAPNUM1	T1C3SEL	T1C3NUM<2:0>			T1C2SEL	T1C2NUM<2:0>			0000 0000	
D4h	T1CAPNFCNT	T1CAPNFCNT<7:0>								0000 0000	
D5h	T2CAPCON	T2C3MOD<1:0>		T2C2MOD<1:0>		T2C1MOD<1:0>		T2COMOD<1:0>		0000 0000	
D1h	POPU	-	-	POPU<5:0>					-	-	--00 0000
D2h	P1PU	P1PU<7:0>								0000 0000	
D3h	P2PU	-	-	-	-	P2PU<3:0>				---- 0000	
D4h	P3PU	P3PU<7:0>								0000 0000	
D5h	P1PD	P1PD<7:0>								0000 0000	
D6h	HDCON	P2HD1		P2HD0		P1HD7		P1HD6		0000 0000	
D7h	T2CAPINT	T2C3IF	T2C2IF	T2C1IF	-	T2C3IE	T2C2IE	T2C1IE	-	000- 000-	
D8h	ADCON2	ADEN	DRDY	ADS	-	-	-	VREFPS<1:0>		0000 -000	
D9h	ADCON1	-	-	-	TSAD<2:0>			ADCKSEL<1:0>		0000 0100	
DAh	ADCON0	ADRDY	AINSL4	VREFSEL<1:0>			AINS<3:0>			0001 0000	
DBh	ADCHL	ADCH<7:0>								0000 0000	
DCh	ADCHH	ADCH<15:8>								0000 0000	
DDh	ADDOL	ADDO<7:0>								0000 0000	
DEh	ADDOH	-	-	-	-	ADDO<11:8>				---- 0000	
DFh	ADDO	ADDO<11:4>								0000 0000	

备注:

1. 在黄色背景字体部分寄存器与棕色背景字体部分寄存器物理地址相同,但是在不同的页面地址中。
2. 访问方法:软件配置 PSW[5] (F0) 来选择页。
3. 灰色表示地址尚未实现。

表 2-2 SFR 特殊寄存器汇总 (续二)

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值	
E0h	ACC	ACC<7:0>								0000 0000	
E1h	P0OE	-	-	P0OE<5:0>						--00 0000	
E2h	P1OE	P1OE<7:0>								0000 0000	
E3h	P2OE	-	-	-	-	P2OE<3:0>				---- 0000	
E4h	P3OE	P3OE<7:0>								0000 0000	
E5h	P1OD	P1OD<7:0>								0000 0000	
E6h	P2OD	-	-	-	-	P2OD<3:0>				---- 0000	
E7h	NULL	Null								-----	
E8h	CLKCON	DEBUGEN	SYSCOEEN	-	TRIMEN	RUNWCK	WCKEN	TCKEN	SYSCKEN	10-0 0111	
E9h	I2C0SPD	I2CSPD<7:0>								0000 0000	
EAh	I2C0DAT	I2CDAT<7:0>								0000 0000	
EBh	I2C0ADR	I2CADR<7:0>								0000 0000	
ECh	I2C0CON	-	I2CEN	STA	STO	I2CIF	AA	-	-	0000 0000	
EDh	I2C0STA	I2C0STAS[4:0]					-	-	-	-	1111 1-00
EEh	LCDCON	-	-	-	-	VOIRS<1:0>		-	VOS	---- 0000	
EEh	TRIMICK	ICK trimming								1000 0000	
EFh	METCH	METCH<7:0>								0000 0000	
F0h	B	B<7:0>								0000 0000	
F1h	INFCON	WDTRF	CMP3IF	CMP2IF	CMP1IF	CMP0IF	ADCIF	-	DPS	0000 00-0	
F2h	P0IF	-	-	P0IF<5:0>						--00 0000	
F3h	P1IF	P1IF<7:0>								0000 0000	
F4h	P2IF	-	-	-	-	P2IF<3:0>				---- 0000	
F5h	P3IF	P3IF<7:0>								0000 0000	
F6h	P2LCDVO	-	-	-	-	P23VO	P22VO	P21VO	P20VO	---- 0000	
F7h	P3LCDVO	P37VO	P36VO	P35VO	P34VO	P33VO	P32VO	P31VO	P30VO	0000 0000	
F6h	IDO	IDO<7:0>								0000 0000	
F7h	IUP	IUP<7:0>								0000 0000	
F8h	T1CAPINT	T1C3IF	T1C2IF	T1C1IF	T1C0IF	T1C3IE	T1C2IE	T1C1IE	T1C0IE	0000 0000	
F9h	MD0	MD0<7:0>								0000 0000	
FAh	MD1	MD1<7:0>								0000 0000	
FBh	MD2	MD2<7:0>								0000 0000	
FCh	MD3	MD3<7:0>								0000 0000	
FDh	MD4	MD4<7:0>								0000 0000	
FEh	MD5	MD5<7:0>								0000 0000	
FFh	ARCON	MDEF	MDOV	SLR	SC<4:0>					0000 0000	
F9h	IAPTIR	IAPTIR<7:0>								0000 0000	
FAh	E2CON	-	-	-	-	-	-	E2RD	E2WR	---- --00	
FBh	IAPADRL	IAPADR<7:0>								1111 1111	
FCh	IAPADRH	IAPADR<15:8>								1111 1111	
FDh	IAPDAT	IAPDAT<7:0>								0000 0000	
FEh	IAPCON	VPPF	IAPLOCK	IAPWSEN	IAPCP	IAPWS<1:0>		IAPRD	IAPWR	0000 1100	
FFh	TRIMTCK	TCK trimming								1000 0000	

备注:

1. 在黄色背景字体部分寄存器与棕色背景字体部分寄存器物理地址相同,但是在不同的页面地址中。
2. 访问方法: 软件配置 PSW[5] (F0) 来选择页。
3. 灰色表示地址尚未实现。

2.2.3 CPU 内核常用 SFR

表 2-3 与 CPU 相关寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
81h	SP	SP<7:0>							
82h	DP0L	DP0L<7:0>							
83h	DP0H	DP0H<7:0>							
84h	DP1L	DP1L<7:0>							
85h	DP1H	DP1H<7:0>							
D0h	PSW	C	AC	F0	RS1	RS0	OV	F1	P
E0h	ACC	ACC<7:0>							
F0h	B	B<7:0>							
F1h	INFCON	WDTIF	CMP3IF	CMP2IF	CMP1IF	CMP0IF	ADCIF	-	DPS

2.3.3.1 程序计数器 PC

程序计数器 PC 不属于 SFR 寄存器。PC 有 16 位，是用来控制指令执行顺序的寄存器。单片机上电或者复位后，PC 值为 0000H，也即是说单片机程序从 0000H 地址开始执行程序。

2.3.3.2 累加器 ACC

累加器 ACC 是 8051 内核单片机的最常用的寄存器之一，指令系统中采用 A 作为助记符。常用来存放参加计算或者逻辑运算的操作数及结果。

2.3.3.3 B 寄存器

B 寄存器在乘除法运算中必须与累加器 A 配合使用。乘法指令 MUL A, B 把累加器 A 和寄存器 B 中的 8 位无符号数相乘，所得的 16 位乘积的低位字节放在 A 中，高位字节放在 B 中。除法指令 DIV A, B 是用 A 除以 B，整数商放在 A 中，余数放在 B 中。寄存器 B 还可以作为通用的暂存寄存器使用。

2.3.3.4 堆栈指针 SP

堆栈指针是一个 8 位的专用寄存器，它指示出堆栈顶部在通用 RAM 中的位置。单片机复位后，SP 初始值为 07H，即堆栈会从 08H 开始向上增加。08H~1FH 为工作寄存器组 R1~R3，最好将 SP 值修改为 80H 后为宜。

2.3.3.5 数据指针 DP0、数据指针 DP1

内核为 DP0 和 DP1 双数据指针结构，能加速数据存储移动。数据指针为 16 位的专用寄存器，它们既可以作为一个 16 位寄存器 DP0 和 DP1 来处理，也可以作为独立的 8 位寄存器 DP0L、DP0H 和 DP1L、DP1H 来处理。

2.3.3.6 程序状态寄存器 PSW

寄存器定义 2-1PSW 程序状态寄存器

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	CY	AC	F0	RS[1:0]		OV	F1	P
D0H	R/W	R/W	R/W	R/W		R/W	R/W	R/W

Bit7CY: 进位标志位

0: 算数或逻辑运算中，没有进位或借位发生

1: 算数或逻辑运算中，有进位或借位发生

Bit6AC: 辅助进位标志位，即半字节进位标志

- 0: 算数或逻辑运算中, 没有辅助进位或借位发生
- 1: 算数或逻辑运算中, 有辅助进位或借位发生

Bit5F0: F0 标志位, SFR 页选择位

- 0: 访问 SFR 页 0
- 1: 访问 SFR 页 1

Bit4-3RS[1:0]: R0~R7 寄存器页选择位

- 00: 页 0 (映射到 00h~07h)
- 01: 页 1 (映射到 08h~0Fh)
- 10: 页 2 (映射到 10h~17h)
- 11: 页 3 (映射到 18h~1Fh)

Bit2OV: ACC 运算溢出标志位

- 0: 没有溢出发生
- 1: 有溢出发生

Bit1F1: F1 标志位

用户自定义标志位

Bit0P: ACC 奇偶校验位

- 0: 累加器中值为 1 的位数为偶数
- 1: 累加器中值为 1 的位数为奇数

2.3.3.7 INFCON 寄存器

通过寄存器 INFCON 的 DPS 进行软件配置, 可选择数据指针 0 或 1, 所有读取或操作数据指针的相关指令将会选择最近一次软件配置选择的数据指针。

寄存器定义 2-INFCON

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INFCON	WDTRF	CMP3IF	CMP2IF	CMP1IF	CMP0IF	ADCIF	-	DPS
F1H	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W

Bit7WDTRF: 看门狗溢出复位标志位 (需要 WDTS[1:0]=00) 看门狗溢出时由硬件置 1, 可由软件、上电/掉电复位、低电压复位和外部复位清 0

- 1: 发生看门狗溢出复位
- 0: 未发生看门狗溢出复位

Bit6-3CMP3IF/CMP2IF/CMP1IF/CMP0IF: CMPxIF 为比较器中断标志申请志位。

- 1: 当比较器 x 中断条件产生时, 此位会被硬件置 1。如果此时中断使能打开的话, cpu 会进入到中断服务程序里面。此位由硬件置 1, 软件清 0。
- 0: 无中断申请事件发生

Bit2ADCIF: ADC 转换完成中断请求标志位

- 1: ADC 中断请求
- 0: 无请求

Bit1: 未使用

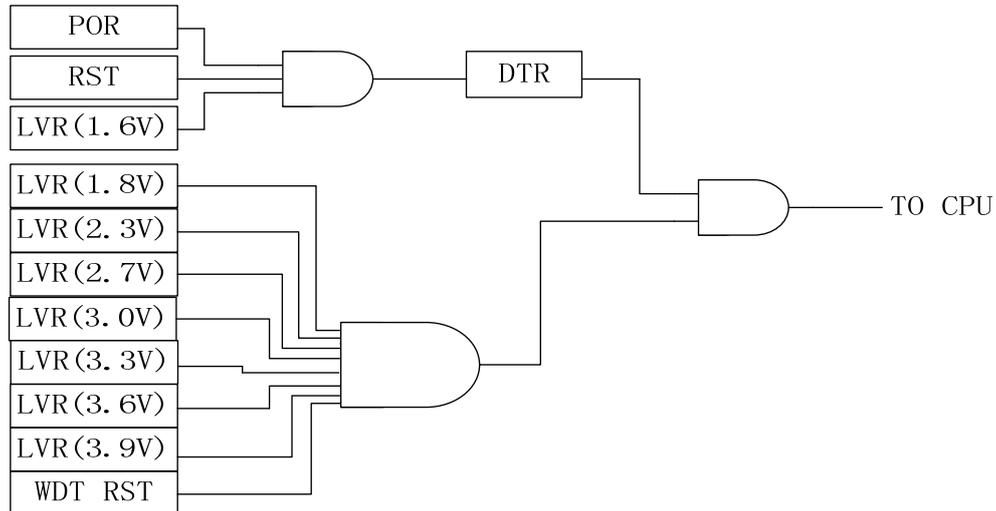
Bit0DPS: 数据指针选择位

- 1: 数据指针 1
- 0: 数据指针 0

3. 复位系统

BJ8M302A 有上电复位 (POR)、外部 RST 复位、低电压复位以及看门狗复位 (WDT) 4 种方式。

图 3-1 复位电路原理框图



3.1 上电复位 (POR)

上电复位 (POR) 电路会将器件一直保持在复位状态，直到 VDD 达到最低工作条件可接受的电平为止。在 VDD 上升缓慢、高速运行或者要求一定模拟性能时，所需要的电压可能高于最低的 VDD，可以使用上电延时电路模块来延长启动周期，直到满足所有器件工作调节为止。其中上电延时电路模块延长复位的时间可以通过程序配置的 RSTSEL 选项项来程序配置，程序配置详情请见 26. 程序配置 章节。

图 3-2 上电复位电路示例及上电过程：

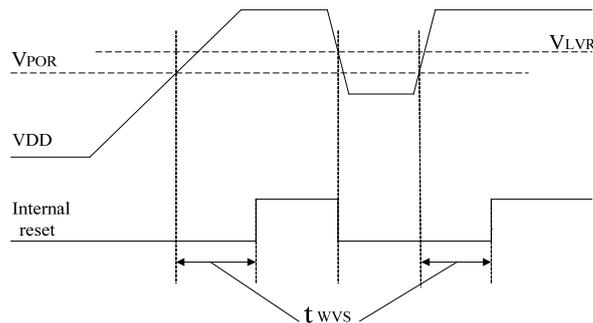


表 3-1 上电复位、低电压复位时间参数表

参数	最小值	典型值	最大值
POR	1.7V	1.75V	1.80V
LVR	1.72	1.80V	1.85V
Twvs	140us	18ms	228ms

备注: V_{POR} : 上电复位 V_{LVR} : 低电压复位 T_{WVS} : 等待电压稳定时间

例 3-1 汇编复位中断向量

```

ORG 0000h
JMP START ; 跳至用户程序。
ORG 0003H
JMP P0_INT ; P0 口中断向量入口
ORG 000BH
JMP T0_INT ; Timer0 口中断向量入口
ORG 0013H
JMP P1_INT ; P1 口中断向量入口
ORG 001BH
JMP T1_INT ; Timer1 口中断向量入口
ORG 0023H
JMP UART_INT ; UART 通信中断向量入口
ORG 002BH
JMP T2PWM_INT ; T2 PWM 中断向量入口
ORG 0033H
JMP T1_INT ; Timer1 口中断向量入口
ORG 0043H
JMP I2C_INT ; IIC 通信中断向量入口
ORG 004BH
JMP P2_INT ; P2 口中断向量入口
ORG 0053H
JMP P3_INT ; P3 口中断向量入口
ORG 005BH
JMP SPI_INT ; SPI 通信中断向量入口
ORG 0063H
JMP T1CAP_INT ; T1 捕获中断向量入口
ORG 006BH
JMP T2_INT ; T2 中断向量入口

ORG 0073H
JMP LVD_INT ; 低电压中断向量入口
ORG 0083H
JMP WDT_INT ; 看门狗中断向量入口
ORG 008BH
JMP T2CAP_INT ; T2 捕获中断向量入口
ORG 0093H
JMP ADC_INT ; ADC 中断向量入口
ORG 009BH
JMP CMP1_INT ; 比较器 1 中断向量入口
ORG 00A3H
JMP CMP2_INT ; 比较器 1 中断向量入口
ORG 00ABH
JMP CMP3_INT ; 比较器 1 中断向量入口
.....
START: ; 用户程序开始。
MOV SP, #81H
..... ; 用户处理程序
JMP START

xxx_INT: ; 各种中断子程序
PUSH PSW
PUSH A
..... ; 中断处理程序
POP A
POP PSW
RETI

ENDP

```

例 3-2C 复位中断向量

```

*****/
#include<Bjx_Type.h>
#include<REG8M302A.h>
#include<intrins.H>
*****

/
unsigned char temp          ;自定义变量
//-----
sbit LED = P1^0;           ;IO 口位定义
//-----

void Init(void);          ;自定义函数
char Function(u8 x);
*****

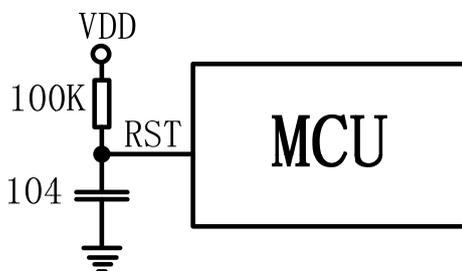
/
void main(void)
{
Init();                  ;初始化代码
    while(1)
    {
        temp+=Function(temp);
        ...
    }
}
*****

```

3.2 外部 RST 复位

外部 RST 复位就是从外部 RST 给 BJ8M302A 一定宽度的复位脉冲信号,来实现 BJ8M302A 的复位。RST/P0.0 管脚在上电时作为复位管脚使用。

图 3-3 外部 RC 基本复位电路:



注意:在一般应用场合可使用 RC 复位电路,在强干扰环境中,建议使用抗干扰复位电路

3.3 低电压复位

BJ8M302A 内建了一个低电压复位电路。而复位的门限电压有 1.8V、2.3V、2.7V、3.0V、3.3V、3.6V、3.9V 等 7 种选择，缺省值是 1.8V。程序配置字请见 26. 程序配置章节。

3.4 看门狗复位(WDT)

详情请见：6. 看门狗 (WDT) 章节。

4. 时钟系统

BJ8M302A 的振荡器模块有六种时钟源选择。他们分别为内部 RC 振荡 (IRC)、外部 RC 振荡 (ERC)、外部高速振荡器 (HXT)、外部低速振荡器 (LXT)、双系统时钟 (DSYSCK) 以及看门狗时钟 (WCK) 作为 CPUCLK 时钟。为了更广泛应用，以及可最大限度地提高性能和降低功耗，振荡器的实际时钟源通过 IDE (Keil) 或者烧录器对 SYSCKM[2:0] 项进行程序配置。当程序配置 IRC 或者 DSYSCK 模式时，系统时钟频率由 ICSEL[2:0] 项确定；当选择其它模式时，系统时钟频率由外部的晶振、或者 RC 或者外部输入时钟频率来确定。

以上 6 种系统模式详情如下：

1. **IRC**：(默认) 系统选择内部 RC 振荡作为系统时钟，而内部时钟的频率需要通过内部时钟频率由 ICKSEL[2:0] 项的程序配置字来确定。
2. **ERC**：系统选择外部 RC 振荡作为系统时钟，而外部时钟的频率与外部连接 R 和 C 的有关。
3. **HXT**：系统选择外部高速晶振或者陶振作为系统时钟，而外部时钟的频率速度与外部振荡器有关。另外，OSCI 也可接入 CLK 作为系统时钟。
4. **LXT**：系统选择外部低速晶振作为系统时钟。
5. **DSYSCK**：程序配置为双系统模式后，外接 32768 晶振时与 Timer 0 模块电路组合成为 RTCC，同时内部 RC 作为系统时钟。RTCC 的软件配置请参考 19. 实时时钟章节，内部 RC 工作模式由 IRC 项的程序配置确定
6. **WCK**：程序配置看门狗时钟 (128KHz) 经过 CLKDIV 预分频后作为 CPU 工作时钟 (CPUCLK)。

表 4-1 振荡器在不同是时钟源工作情况下频率、引脚、STOP 模式下时钟的关系

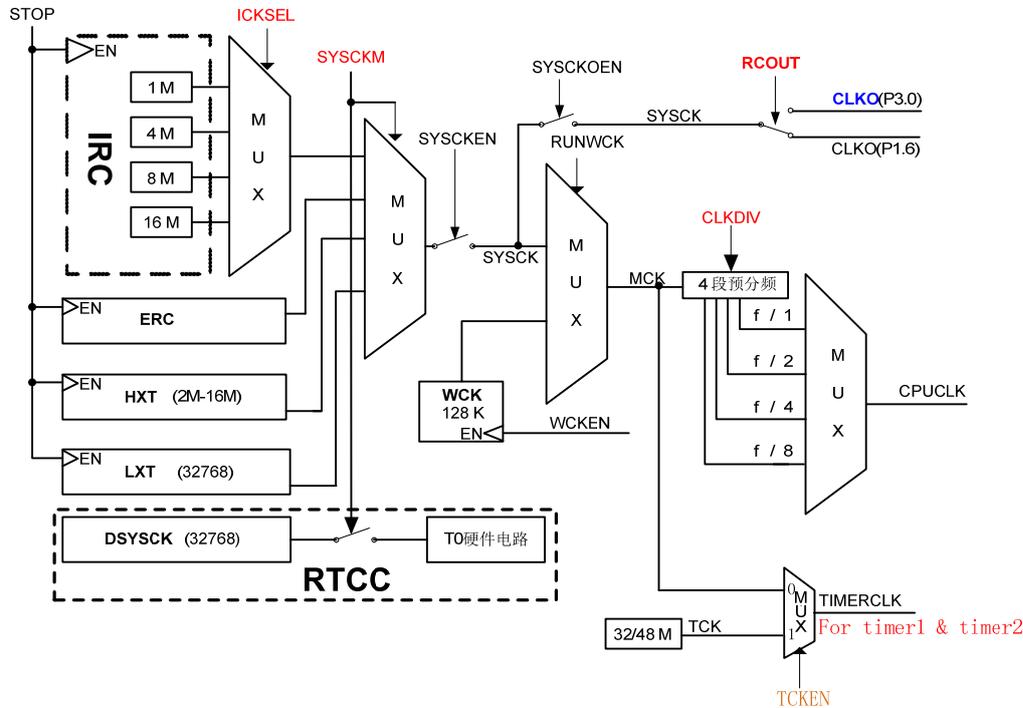
类型	名称	频率	引脚	备注 (STOP 模式下)
内部 RC	IRC	1/4/8/16M	-	关闭内部时钟
外部 RC	ERC		OSCI	关闭内部时钟
外部高速晶振	HXT	2M~16M	OSCI/OSCO	关闭内部时钟
外部低速晶振	LXT	32.768K	OSCI/OSCO	关闭内部时钟
双系统时钟	DSYSCK	RTCC+IRC	OSCI/OSCO	关闭内部时钟，RTCC 时钟正常，CPU 停止执行指令。有信号唤醒后，CPU 才能工作
看门狗时钟	WCK	128K	-	关闭内部时钟，WCK 时钟正常，CPU 停止执行指令。有信号唤醒后，CPU 才能工作

图 4-1 简化的 MCU 时钟源框

注意：红色字体表示通过 IDE 或者烧录器软件进行程序配置，程序配置完成后，不能通过软件自行修改

ICKSEL：内部时钟频率选择 SYSCKM：时钟源&引脚配置选择

CLKDIV：主时钟分频选择 RCOUT：系统时钟（未分频）输出选择



寄存器定义 4-1 CLKCON 时钟系统控制寄存器

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	DEBUGEN	SYSCOEN	-	TRIMEN	RUNWCK	WCKEN	TCKEN	SYSCKEN
E8H	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W

Bit7DEBUGEN: BJX 两线调试/烧录系统使能管脚。默认为 1。

- 1: 使能芯片的调试/烧录功能，此时 P3.0 和 P3.2 不可作为 GPIO 脚使用。
- 0: 禁止芯片的调试/烧录功能，此时 P30 和 P32 可作为 GPIO 脚使用。

Bit6SYSCOEN: 系统时钟输出控制位

- 1: 允许系统时钟输出，输出管脚需要与 OPTION 的 RCOUT 配合使用
- 0: 禁止系统时钟输出

Bit5 未使用

Bit4TRIMEN: ICK 和 TCK 用 SFR 进行 trimming 的使能信号。

- 1: 使能 SFR 写 TRIMICK/TRIMTCK 寄存器，从而对 ICK/TCK 进行 trimming。
- 0: 禁止 SFR 写 TRIMICK/TRIMTCK 寄存器

Bit3RUNWCK: CPU 时钟源选择位 看门狗时钟 WCK 作为 CPUCLK 使能位

- 1: 选择 WCK 作为 MCK 时钟源（若切换之前 WCK 为关闭状态，CPU 会自动打开内置 128K 振荡器，然后将主时钟切换到 WCK）
- 0: 选择 SYSCK 作为 MCK 时钟源

Bit2WCKEN: 看门狗时钟 WCK 使能位

- 1: 使能看门狗时钟 WCK（当需要调试下载功能时，必须使能 WCKEN 位。否则调试下载功能失效）
- 0: 禁止看门狗时钟 WCK

Bit1TCKEN: Timer 时钟 TCK 使能位

- 1: 使能 TCK
- 0: 禁止 TCK

Bit0SYSCKEN: 系统时钟（SYSCK）时钟开关

- 1: 使能 SYSCK 时钟源通道
- 0: 禁止 SYSCK 时钟源通道

4.1 IRC 模式

通过程序配置选项选择步骤，将系统时钟来源选择为 IRC 模式：

1. 通过 IDE 或者烧录软件程序配置 SYSCKM[2:0]选项选择 IRC 模式
2. 通过 IDE 或者烧录软件程序配置 ICSEL[2:0]选项选择内部 RC 振荡频率
3. 通过 IDE 或者烧录软件程序配置 CLKDIV[1:0]选项选择时钟分频后做为系统工作时钟

4.2 ERC 模式

通过程序配置选项选择步骤，将系统时钟来源选择为 ERC 模式：

1. 通过 IDE 或者烧录软件程序配置 SYSCKM[2:0]选项选择 ERC 模式
2. 通过 IDE 或者烧录软件程序配置 CLKDIV[1:0]选项选择时钟分频后做为系统工作时钟

4.3 HXT 模式

通过程序配置选项选择步骤，将系统时钟来源选择为 HXT 模式：

1. 通过 IDE 或者烧录软件程序配置 SYSCKM[2:0]选项选择 HXT 模式
2. 通过 IDE 或者烧录软件程序配置 CLKDIV[1:0]选项选择时钟分频后做为系统工作时钟

4.4 LXT 模式

通过程序配置选项选择步骤，将系统时钟来源选择为 LXT 模式：

1. 通过 IDE 或者烧录软件程序配置 SYSCKM[2:0]选项选择 LXT 模式
2. 通过 IDE 或者烧录软件程序配置 CLKDIV[1:0]选项选择时钟分频后做为系统工作时钟

4.4 DSYSCK 模式

通过程序配置选项选择步骤，将系统时钟来源选择为 DSYSCK 模式，外部 32.768KHz 和 Timer0 整体作为 RTCC：

1. 通过 IDE 或者烧录软件程序配置 SYSCKM[2:0]选项选择双系统工作模式 (DSYSCK)
2. 通过 IDE 或者烧录软件程序配置 ICSEL[2:0]选项选择内部 RC 振荡频率
3. 通过 IDE 或者烧录软件程序配置 CLKDIV[1:0]选项选择时钟分频后做为系统工作时钟

注意：该模式下，CPU 的时钟来源于内部 RC 时钟。

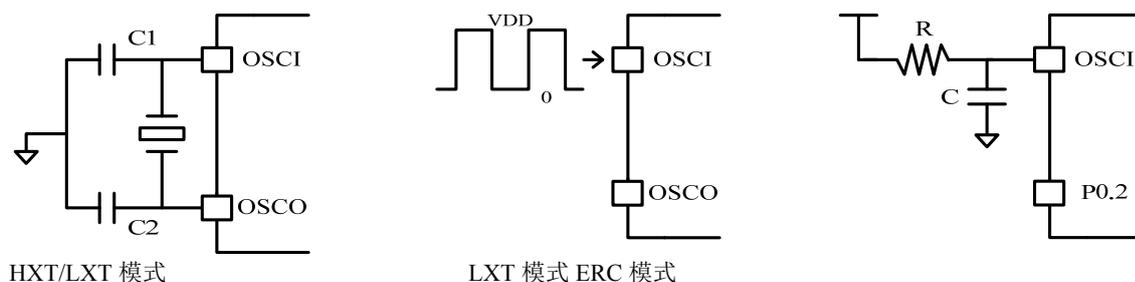
4.5 WCK 模式

通过如下程序配置选项和软件配置操作步骤，可将看门狗时钟源用作作为器件 CPUCLK：

1. 通过 IDE 或者烧录软件程序配置 CLKDIV[1:0]选项选择 MCK 分频后做为 CPU 工作时钟
2. 通过软件配置 WCKEN 为 1。即使能看门狗模块 (WCKEN=1)
3. 通过软件选择看门狗时钟 (WCK) 作为 MCK 时钟 (RUNWCK=1)

4.6 外部时钟输入接口参考电路

图 4-2 外部震荡电路参考应用电路 (OSCI/P0.1 复用 OSCO/P0.2 复用)



HXT/LXT：外接晶振或者陶振时，需要外接与挂晶振配合电容。OSCI/OSCO 口驱动的不同振荡频率，振荡器的驱动电流也不同；

LXT: 从 OSCI 输入 CLK 也可以正常工作。

ERC: OSCO/P0.2 做为 IO, OSCI 为外接匹配的电阻电容即可。OSCO 引脚作为普通的 I/O 口;

5.省电(低功耗)模式

BJ8M302A 支持低功耗模式。为了使 BJ8M302A 处于待机状态, 可以让 CPU 停止工作, 也就是让 CPU 进入停止 (IDLE) 或睡眠模式 (STOP), 减低功耗。

在省电模式下, 包括 IDLE 和 STOP 模式, 芯片的各个 IO 口保持省电设置之前的状态。

寄存器定义 5-1 PCON

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	-	-	-	-	-	-	STOP	IDLE
87H	-	-	-	-	-	-	W	W

Bit7-2 未使用

Bit1 STOP: STOP 模式进入控制位

1: 置 1, 进入 STOP 模式

0: 唤醒时硬件自动清 0, 进入工作模式

Bit0 IDLE: IDLE 模式进入控制位

1: 置 1, 进入 IDLE 模式

0: 唤醒时硬件自动清 0, 进入工作模式

5.1 IDLE 模式

IDLE 模式能够降低系统功耗, 在此模式下, 主基准和 LDO、振荡器等模块不会自动关闭, 程序计数器停止运行, CPU 时钟停止, 但外设时钟继续运行, 如果之前有外设模块已经开启, 则开启的外设模块继续工作。CPU 进入 IDLE 时, 之前所有状态都被保存, 如 PC、PSW、SFR 等等。

所有的中断、IO 翻转变醒、复位信号 (外部硬件复位、WDT 复位、LVR 复位) 等均能唤醒 CPU, 退出 IDLE 模式。

5.2 STOP 模式

STOP 模式 (掉电模式) 能够达到系统最低功耗。CPU 进入 STOP 时, 之前所有状态都被保存, 如 PC、PSW、SFR 等等。STOP 模式典型情况下, 主基准和 LDO、主振荡器模块都将自动关闭 (WCK 需软件关闭), 如果软件也禁止了其他外设功能, 所有输出引脚无负载, 输入引脚不翻转, 系统将达到约 1.2uA 的待机电流。

IO 翻转、外部中断 (只有低电平模式才能唤醒 STOP)、RTCC 定时和看门狗溢出可以唤醒 STOP 模式。

为了避免中断返回可能引起的程序错误, 建议在停止指令之后加一条 NOP 指令以保证程序返回时能正常运行。

外部硬件复位、WDT 复位 (如果被允许)、LVR 复位 (如果被允许) 均能唤醒 CPU, 退出 STOP 模式。

例 5-1 STOP 和 IDLE 模式进入示例

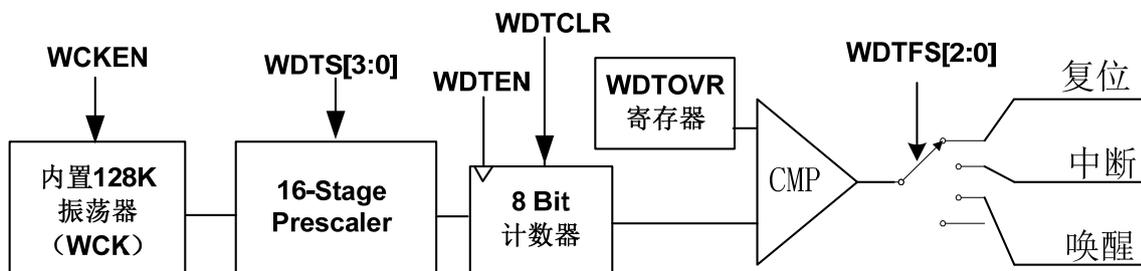
```
//Enter STOP Mode                                //Enter IDLE Mode
PCON  =  0x02//唤醒后需要 4 个 NOP 延时          PCON  =  0x01//唤醒后需要 4 个 NOP 延时
_nop_();                                          _nop_();
_nop_();                                          _nop_();
_nop_();                                          _nop_();
_nop_();                                          _nop_();
```

6.看门狗 (WDT)

看门狗定时器是一个系统定时器,如果软件未在超时周期置位 WDTCLR,看门狗定时器会产生复位。
看门狗定时器通常用于使系统从意外事件中恢复。

BJ8M302A 有单独的内置 128K 振荡器作为 WDT 时钟源 WCK。当用户把 WCKEN 置 1 时,则内部的看门狗定时器振荡器 WCK (128KHz) 将会启动,产生的时钟被送到“16 bits 预分频器”来进行时钟分频。用户根据 WDTS[3:0]来选择不同的时钟分频系数,被分频后的时钟送给后面的“8 bits 计数器”。当用户置位 WDTEN 时,“8 bits 计数器”开始计数,当“8 bits 计数器”的数值与 WDTOVR 数值相等时看门狗便溢出,溢出信号通过 WDTFS 来选择对 MCU 进行复位、中断或者唤醒功能。用户可以通过置位 WDTCLR 清除“8 bits 计数器”的数值而不产生看门狗便溢出信号。

图 6-1 看门狗定时器框图



6.1 相关寄存器

表 6-1 与看门狗相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
A8h	IEN0	EA	-	-	T2IE	T1IE	P1EA	T0IE	POEA
B9h	IEN2	-	LVDIE	CMP3IE	CMP2IE	CMP1IE	ADCIE	-	WDTIE
F1h	INFCON	WDRF	CMP3IF	CMP2IF	CMP1IF	CMP0IF	ADCIF	-	DPS
E8h	CLKCON	DEBUGEN	SYSCOEN	-	TRIMEN	RUNWCK	WCKEN	TCKEN	SYSCKEN
BEh	WDTCON	WDTEN	WDT<3:0>				WDTFS<1:0>		WDTCLR
BFh	WDTOVR	WDTOVR<7:0>							

寄存器定义 6-1 IEN2

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN2	-	LVDIE	CMP3IE	CMP2IE	CMP1IE	ADCIE	-	WDTIE
B9H	-	R/W	R/W	R/W	R/W	R/W	-	R/W

Bit7 未使用

Bit6LVDIE:低电压中断使能位

1:低电压中断使能

0:禁低电压中断

Bit5CMP3IE:比较器 3 中断使能位

1:开启比较器 3 中断

0:禁止比较器 3 中断

Bit4CMP2 IE:比较器 2 中断使能位

1:开启比较器 2 中断

- 0:禁止比较器 2 中断
- Bit3CMP1 IE:**比较器 1 中断使能位
 - 1:开启比较器 1 中断
 - 0:禁止比较器 1 中断
- Bit2ADC IE:**ADC 中断使能位
 - 1:开启 ADC 中断
 - 0:禁止 ADC 中断
- Bit1** 未使用
- Bit0WDTIE:**WDT 溢出中断使能位
 - 1:使能 WDT 中断
 - 0:禁止 WDT 中断

寄存器定义 6-2 WDTCON

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDTCON	WDTEN	WDT5 <3:0>			WDTFS<1:0>		WDTCLR	
BEH	R/W	R/W			R/W		W	

Bit7WDTEN: 看门狗 WDT 使能位

- 1: 开启 WDT
- 0: 禁止 WDT

Bit6-3 WDT5[3:0]: WDT 时钟分频选择位

WDT5[3:0] WDT 时钟分频系数

- 0000 WCK/65536 (默认)
- 0001 WCK/32768
- 0010 WCK/16384
- 0011 WCK/8192
- 0100 WCK/4096
- 0101 WCK/2048
- 0110 WCK/1024
- 0111 WCK/512
- 1000 WCK/256
- 1001 WCK/128
- 1010 WCK/64
- 1011 WCK/32
- 1100 WCK/16
- 1101 WCK/8
- 1110 WCK/4
- 1111 WCK/2

Bit2-1WDTFS[1:0]: WDT 功能选择位

WDTFS[1:0]WDT 功能

- 00 作为复位功能(默认)
- 01 作为中断, 产生中断, 通知 CPU
- 10 作为唤醒功能, 即在 STOP 和 IDLE 模式下, 可以唤醒 CPU
- 11 无效(禁止软件配置)

Bit0WDTCLR: 清 WDT 计数器使能位

- 置 1, WDT 将会重新计数
- WDTCLR 由硬件自动清 0

寄存器 WDTOVR (BFH) 是 WDT 溢出阈值, 数据宽度为 8 位, 数据可以读写。电源上电默认值为 FFH, 当 WDT 溢出后产生的唤醒、中断、复位均不会清 0 该溢出阈值。

注意: WDTOVR 寄存器不能设置为 0 时, 否则看门狗计数器值不可能小于寄存器 WDTOVR 为 0 阈值, 比较器信号永远无法翻转, WDT 也无溢出信号, 即相当于 WDT 功能无效。

WDT 溢出时间计算:

$$T_{OV} = \frac{1}{128KHz} \cdot 2^{(16-WDT5[3:0])} \cdot (WDTOVR[7:0] + 1)$$

表 6-2WCK 分频与溢出时间

WDTS[3:0]	WDT 时钟分频系数	溢出时间 (当 WDTOVR=FFh)
0000	WCK/65536	131072ms
0001	WCK/32768	65536ms
0010	WCK/16384	32768ms
0011	WCK/8192	16384ms
0100	WCK/4096	8192ms
0101	WCK/2048	4096ms
0110	WCK/1024	2048ms
0111	WCK/512	1024ms
1000	WCK/256	512ms
1001	WCK/128	256ms
1010	WCK/64	128ms
1011	WCK/32	64ms
1100	WCK/16	32ms
1101	WCK/8	16ms
1110	WCK/4	8ms
1111	WCK/2	4ms

6.2 WDT 功能选择

当寄存器 WDTCON 的 WDTFS[1:0]=00，作为复位功能。当 WDT 计数溢出后，系统复位到初始值，但 WDTOVR[7:0]计数溢出值不会被复位，此时寄存器 INFCON[7]的 WDTRF 将置 1，以记录 WDT 溢出引起了系统复位。用户可查询此标志位来判断是否是看门狗溢出产生了复位。WDTRF 可以由软件、上电/掉电复位、低电压复位和外部复位清 0。

当寄存器 WDTCON 的 WDTFS[1:0]=01，作为中断功能。此时需要寄存器 IEN0[7]的 EA 为 1，寄存器 IEN2[0]的 EWDT 为 1。WDT 中断的入口地址为 83h。

当寄存器 WDTCON 的 WDTFS[1:0]=10，作为唤醒功能。系统进入 IDLE 和 STOP 模式后，均能通过 WDT 溢出唤醒。唤醒 CPU 后，会继续往下一条指令顺序运行。

6.3 WDT 操作步骤

- (1) 开启 WCK 时钟使能位，即置 CLKCON 的 WCKEN 位为 1。
- (2) 设置 WCK 时钟分频系数，即软件配置寄存器 WDTCON 的 WDTS[3:0] 的值。
- (3) 软件配置寄存器 WDTCON 的 WDTFS[1:0]，选择 WDT 作为复位 00、中断 01、唤醒 10 功能。
- (4) 软件配置 WDT 时钟溢出值 WDTOVR[7:0]。
- (5) 开启 WDT 使能信号，即寄存器 WDTCON 的 WD TEN 为 1。
- (6) 在产生计数溢出前清狗，即将寄存器 WDTCON 的 WDTCLR 置 1。

例6-1 WDT 初始化示例（复位时间间隔时间为 1024mS）

```
void WDT_Init(){
    WDTCON = 0x38; //1.设置 WCK 时钟分频(WCK/512) 2.设置 DTS 为 00 作为复位分频
    CLKCON &= 0xFE; //3.关闭 128K 时钟作为系统时钟通道
    WDTOVR = 0xFF; //4.设置计数阈值为 FF
    INE2   &= 0xFE; //5.禁止 WDT 溢出中断使能
}

void main(){
    WDT_Init();
    .....
    while(1)
    {
        WDTCLR=1; //喂狗
        .....
    }
}
```

7. I/O 端口

BJ8M302A 共有 P0~P3 四组共 26 个通用 I/O 端口，除了 P0.0 仅为输入口外，其它 25 个均为双向 I/O 口，且具备中断功能和翻转唤醒功能。所有通用 I/O 端口均具备上拉功能，P1 和 P2 同时具备开漏输出和下拉功能。内部上下拉典型电阻值为 50K。

用户在软件配置通用 I/O 状态之前，默认为数字输入功能，且上拉、下拉、漏极开路等功能均关闭。若此期间 I/O 引脚的输入电平为中间状态或者输入浮空，将会产生额外的不定态功耗。P0.0/NRST、OSCI/P0.1、OSCO/P0.2 三个复用管脚功能由 IDE 或者烧录软件程序配置选择决定。若程序配置为 I/O 口，与其它 I/O 使用方法一样。

注意：

因调试口时钟输入引脚 PCLK 与 P3.0 复用；调试口数据输入输出引脚 PDAT 与 P3.2 复用。时钟控制寄存器 CLKCON 的第七位 DEBUGEN 位为调试控制寄存器。默认值为 1，即为调试模式。

在调试模式下，P3.0 和 P3.2 作为 I/O 功能的软件配置无效，P3.0 为在线调试/编程接口中的时钟引脚（PCLK）；P3.2 为在线调试/编程接口中的数引脚（PDAT）。

当软件使 DEBUGEN=0 后，取消了调试模式，P3.0 和 P3.2 同其他 I/O 使用方法一样。

注意：黄色背景寄存器位于 SFR 列表第 1 页，需要软件配置 PSW[5]=1 进行访问

表 7-1 与 IO 口相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
80h	P0	-	-	P0<5:0>					
90h	P1	P1<7:0>							
A0h	P2	-	-	-	-	P2<3:0>			
B0h	P3	P3<7:0>							
E1h	P0OE	-	-	P0OE<5:0>					
E2h	P1OE	P1OE<7:0>							
E3h	P2OE	-	-	-	-	P2OE<3:0>			
E4h	P3OE	P3OE<7:0>							
E5h	P1OD	P1OD<7:0>							
E6h	P2OD	-	-	-	-	P2OD<3:0>			
D1h	P0PU	-	-	P0PU<5:0>					
D2h	P1PU	P1PU<7:0>							
D3h	P2PU	-	-	-	-	P2PU<3:0>			
D4h	P3PU	P3PU<7:0>							
D5h	P1PD	P1PD<7:0>							
9Fh	P2PD	-	-	-	-	P2PD<3:0>			
EFh	METCH	-	-	VTHS	-	ADCVOF<3:0>			

Px (P0/P1/P2/P3) 为数据锁存寄存器。相应的数据位可读写。

PxOE (P0OE/P1OE/P2OE/P3OE) 为 Px 口输入/输出控制寄存器。0: 输入 1: 输出。相应的数据位可读写。

PxPU (P0PU/P1PU/P2PU/P3PU) 为 Px 口上拉功能使能寄存器。0: 关闭 1: 开启。相应的数据位可读写。

P1PD/P2PD 为 P1/P2 口下拉功能使能寄存器。0: 关闭 1: 开启。相应的数据位可读写。

P1OD/P2OD 为开漏功能使能寄存器。0: 关闭 1: 开启。相应的数据位可读写。

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
METCH	-	-	VTHS	-	ADCVOF <3:0>			
EFh	-	-	R/W	-	R/W			

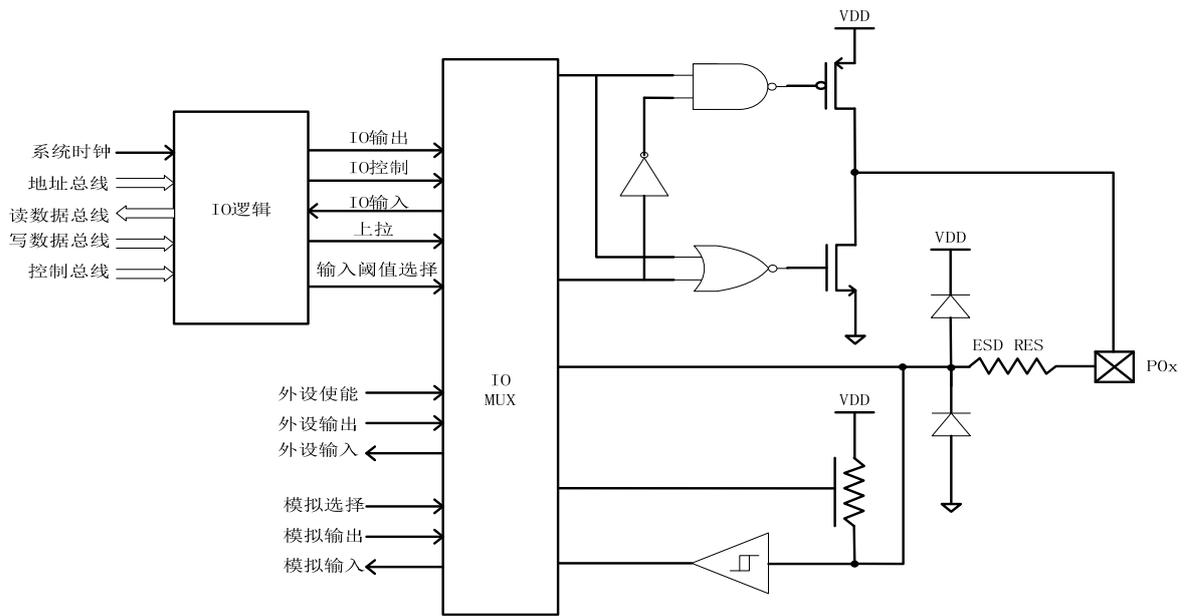
Bit5 VTHS: I/O 输入电压门限选择

0: 输入高电压=0.7VDD, 输入低电压=0.3VDD

1: 输入高电压=0.4VDD, 输入低电压=0.2VDD

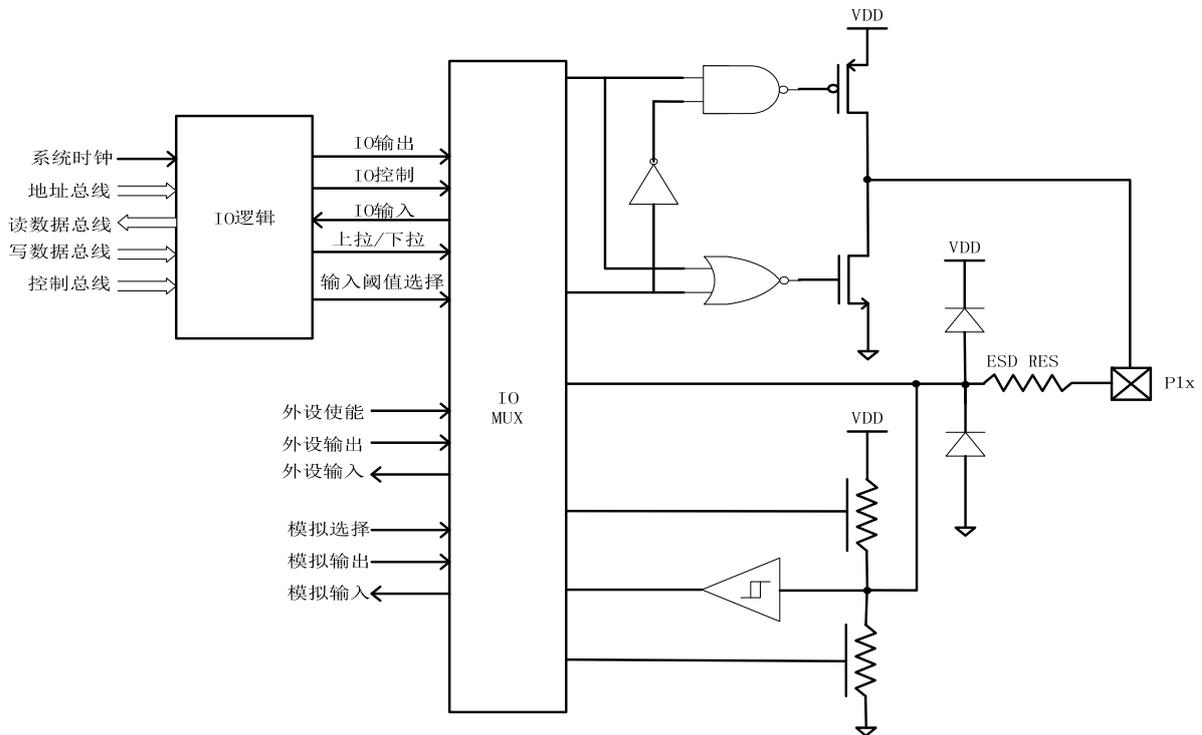
7.1P0 结构框图

图 7-1 P0 口结构框图



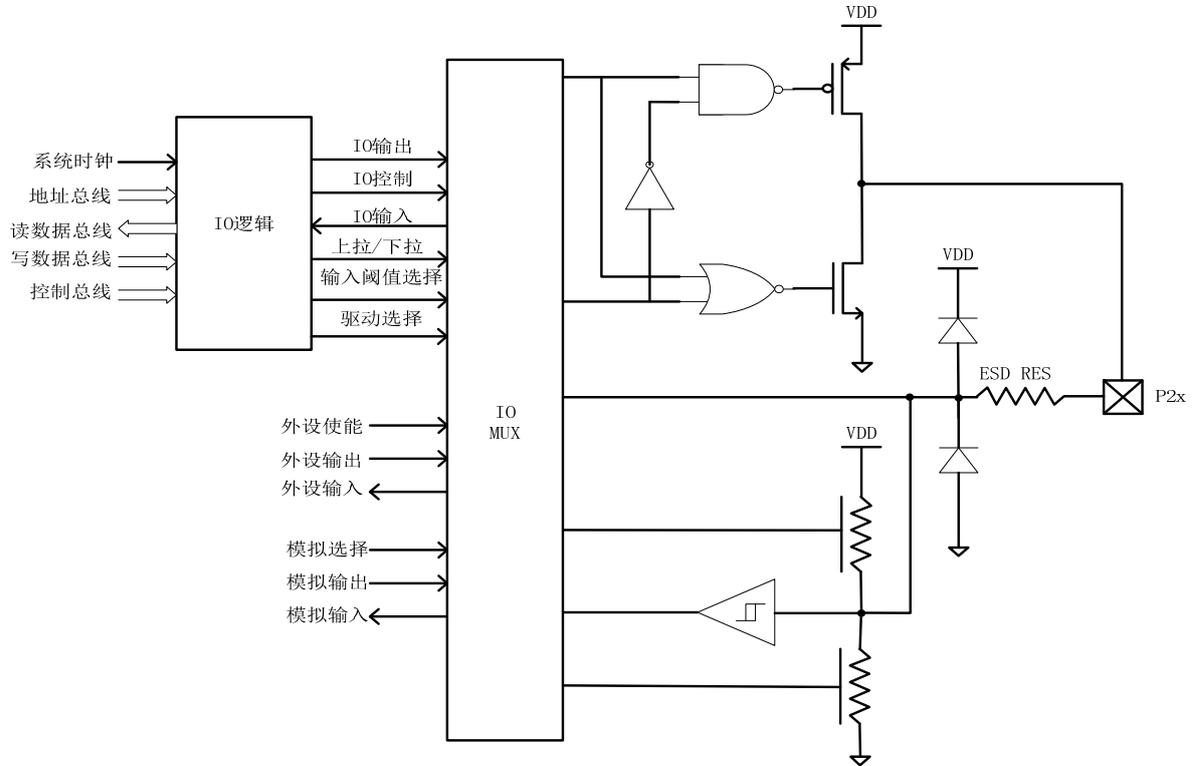
7.2P1 结构框图

图 7-2 P1 口结构框图



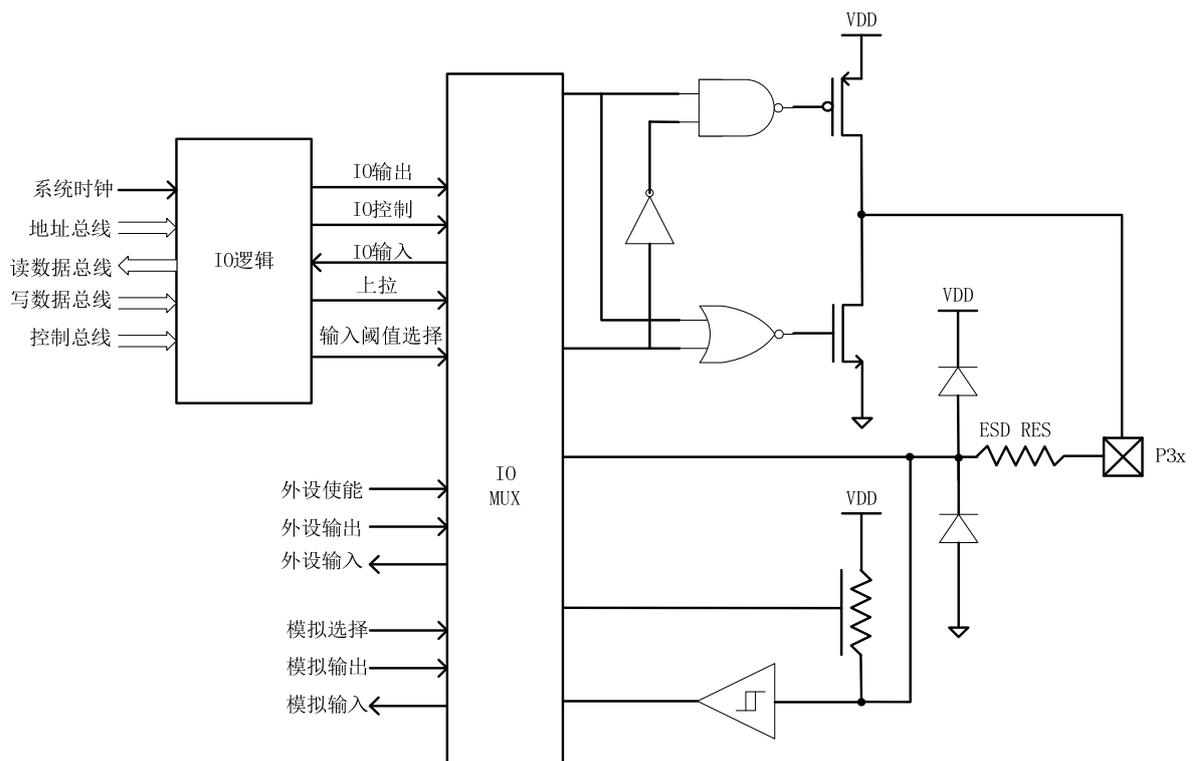
7.3P2 结构框图

图 7-3 P2 口结构框图



7.4P3 结构框图

图 7-4 P3 口结构框图



例 7-1 I/O 口参考示例

```

//P0 口上拉输出 1
//P1 口下拉输出 0
//P2 口开路输入
//p3 口上拉输出 0
#define page0 F0=0
#define page1 F0=1
unsigned char buf;
DEBUGEN= 0; //关闭调试功能
P0OE = 0xFF; //设为输出口
page1;
P0PU = 0xFF; //上拉
page0;
P0 = 0xFF; //输出 1
P1OE = 0xFF; //设为输出口
page1;

P1PU = 0x00; //关闭上拉
P1PD = 0xFF; //开启下拉
page0;
P1 = 0x00; //输出 0
P2OE = 0x00; //设为输入口
page1;
P2PU = 0x00; //关闭上拉
P2PD = 0x00; //关闭下拉
P2OD = 0xFF; //开启漏极开路输入
page0;
buf = P2; //读取 P2 口的值
P3OE = 0xFF; //设为输出口
ppage1;
P3PU = 0xFF; //上拉
page0;
P3 = 0x00; //输出 0

```

7.5 I/O 翻转换醒 CPU

I/O 翻转换醒功能可以唤醒之前处于 IDLE 或者 STOP 的 CPU。唤醒后，CPU 继续往下运行指令。I/O 翻转换醒可以从低电平到高电平变化，也可以是从高电平到低电平的变化。只要是 I/O 的电平状态发生改变，并且使能了 I/O 翻转换醒功能，都可以唤醒之前处于 IDLE 或者 STOP 状态的 CPU，此功能同外部中断唤醒类似。

只需要配置 I/O 唤醒功能使能位 IOWU 为 1 (IOWU 位于 IOWAKE 的第 0bit)，便开启了 I/O 翻转换醒功能。具体需要使用哪一个 I/O 口翻转来唤醒 CPU，可以通过配置 P0IE, P1IE, P2IE, P3IE 来选择。

注意，当开启 I/O 翻转换醒时，如果也同时开启了中断使能，则 I/O 翻转换醒后，CPU 会进入中断服务程序，这个与外部中断相同。如果不开启中断使能，则 I/O 翻转换醒后，CPU 将继续往下运行指令。

7.6 高电流驱动 I/O 口

BJ8M302A 芯片有 4 个高电流输出的 I/O 口，分别为 P16, P17, P20, P21。可以通过软件配置来获得不同档位的电流输出，具体软件配置如下。

寄存器定义 7-1 HDCON I/O 口高电流驱动控制位寄存器

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
HDCON	P2HD1[1:0]		P2HD0[1:0]		P1HD7[1:0]		P1HD6[1:0]	
D6h	R/W		R/W		R/W		R/W	

Bit7-6 P2DH1: P2.1 口高驱动电流控制位

- 00: 输出驱动电流 20mA
- 01: 输出驱动电流 40mA
- 10: 输出驱动电流 60mA
- 11: 输出驱动电流 80mA

Bit5-4 P2DH0: P2.0 口高驱动电流控制位

- 00: 输出驱动电流 20mA
- 01: 输出驱动电流 40mA
- 10: 输出驱动电流 60mA
- 11: 输出驱动电流 80mA

Bit3-2 P1DH7: P1.7 口高驱动电流控制位

- 00: 输出驱动电流 20mA

01: 输出驱动电流 40mA

10: 输出驱动电流 60mA

11: 输出驱动电流 80mA

Bit1-0 P1DH6: P1.6 口高驱动电流控制位

00: 输出驱动电流 20mA

01: 输出驱动电流 40mA

10: 输出驱动电流 60mA

11: 输出驱动电流 80mA

8.低电压检测

BJ8M302A 内部提供 8 档电压检测，LVDS[2:0]为选择电压检测阈值，由 LVDF 标志位看电压检测结果。同时，当芯片工作电压低于设定的电压值时，如果使能了 LVD 的中断使能 IEN2[6]，则此时可以进入到中断，中断入口地址为 73h。

寄存器定义 8-1 LVDCON 低电压检测寄

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDCON	LVDF	LVDFIF	-	-	LVDS<2:0>			LVDEN
C7h	R	R/W	-	-	R/W			R/W

Bit7 LVDF: 供电电压的状态指示位。 **由硬件置 1 或者清 0**

1: 供电电压低于设定电压值。

0: 供电电压高于等于设定电压状态

Bit6LVDFIF: LVD 的中断标识位，只允许软件清零。

1: 产生低电压中断标志，硬件置 1。

0: LVD 中断标志位。 软件清 0。

Bit5-4 未使用

Bit3-1LVDS[2:0]:低电压检测电压阈值选择

000 低电压检测阈值 2.2V (默认)

001 低电压检测阈值 2.4V

010 低电压检测阈值 2.6V

011 低电压检测阈值 2.9V

100 低电压检测阈值 3.2V

101 低电压检测阈值 3.5V

110 低电压检测阈值 3.8V

111 低电压检测阈值 4.1V

Bit0LVDEN: 低电压检测模块使能位

1: 打开低电压检测模块

0: 关闭低电压检测模块

9. 中断

中断系统是为使 CPU 具有外界紧急事件的处理能力而设置的。

当中央处理机 CPU 正在处理某件事的时候外界发生了紧急事件请求，要求 CPU 暂停当前的工作，转而去处理这个紧急事件，处理完以后，再回到原来被中断的地方，继续原来的工作，这样的过程称为中断。实现这种功能的部件称为中断系统，请示 CPU 中断的请求源称为中断源。

BJ8M302A 有 20 个中断源，对应不同的中断入口地址。中断总使能位 EA 可以实现所有中断的打开或者关闭。P0 口 6 个中断源共享一个入口地址；P2 口 4 个中断源共享一个入口地址；P1 和 P3 口均是 8 个中断源共享一个入口地址；有些中断可软件配置，为将 MCU 从休眠模式唤醒。

当 CPU 正在处理一个中断源请求的时候（执行相应的中断服务程序），发生了另外一个比他优先级还高的中断源请求，如果 CPU 能够暂停对原来中断源的服务程序，转而去处理优先级更高的中断请求源，处理完以后，再回到原低级中断服务程序，这样的过程称为中断嵌套；这样的中断系统称为多级中断系统，没有中断嵌套功能的中断系统称为单级中断系统。

BJ8M302A 的 20 个中断源分为 6 组，IP1 和 IP0 是两中断优先级控制寄存器。通过软件配置 IP1[5:0] 和 IP0[5:0] 的值，可以实现 4 层中断优先级，即最多可以中断嵌套 4 次。

9.1 中断入口向量表

表 9-1 中断源、入口地址、分组以及默认优先级

优先级 H ↓ L	组	对应配置	中断源及其入口地址			
	0	IP1[0] IP0[0]	外部P0口 03h	T1 PWM 33h	WDT中断 83h	预留 C3h
1	IP1[1] IP0[1]	T0 中断 0Bh	预留 3Bh	T2捕获 8Bh	预留 CBh	外部P2口 4Bh
2	IP1[2] IP0[2]	外部P1口 13h	LVD 73h	ADC中断 93h	预留 D3h	外部P3口 53h
3	IP1[3] IP0[3]	T1中断 1Bh	预留 7Bh	CMP1中断 9Bh	预留 DBh	SPI中断 5Bh
4	IP1[4] IP0[4]	UART中断 23h	预留 B3h	CMP2中断 A3h	预留 E3	T1 捕获 63h
5	IP1[5] IP0[5]	T2 PWM 2Bh	预留 BBh	CMP3中断 ABh	预留 Ebh	T2中断 6Bh

优先级 Highest —————> Lowest

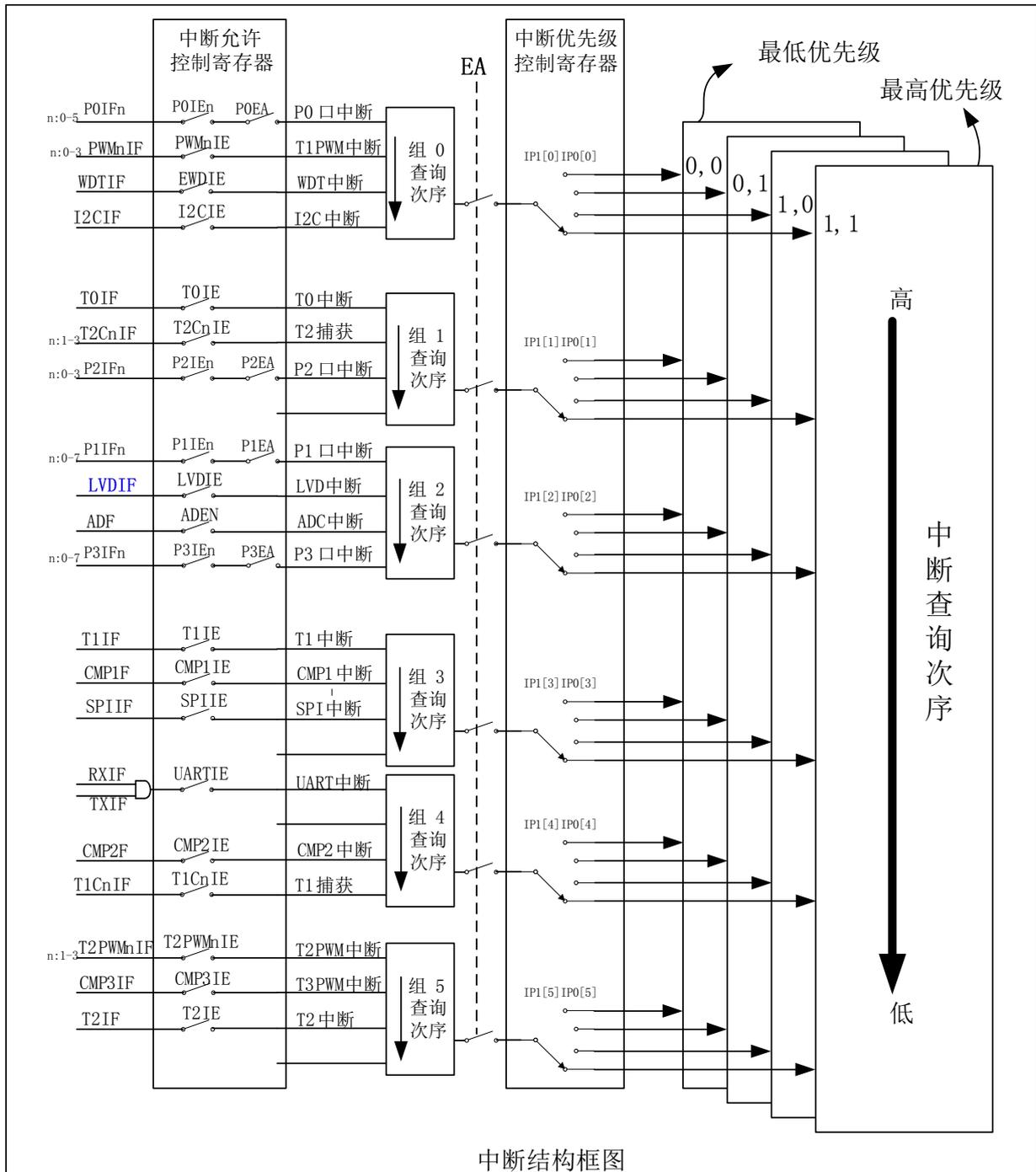
9.2 中断源中断向量

表 9-2 中断入口地址列表

序号	入口地址	中断源类型	相应使能位	相应中断标志	标志处理方式
0	03h	P0口中断	IEN0[0]的P0EA/P0IE[5:0]	EX0IF[5:0]	*软件清零
1	0Bh	T0溢出中断	IEN0[1]的T0IE	TCON[5]的T0IF	中断模式硬件清零
					查询模式软件清零
2	13h	P1口中断	IEN0[2]的P1EA/P1IE[7:0]	P1IF[7:0]	*软件清零
3	1Bh	T1溢出中断	IEN0[3]的T1IE	T1CON[5]的T1IF	中断模式硬件清零
					查询模式软件清零
4	23h	UART中断	IEN1[5][4]的SORXIE/SOTXIE	SOCON的RXIF、TXIF	软件清零
5	2Bh	T2PWM中断	T2PWMINTCON[3:1]	T2PWMINTCON[7:5]	软件清零
6	33h	T1 PWM中断	T1PWMINTCON[3:0]	T1PWMINTCON[7:4]	软件清零
7	3Bh	预留	预留	预留	-
8	43h	I2C中断	IEN1[0]的I2CIE	I2CCON[3]的I2CIF	软件清零
9	4Bh	P2口中断	IEN1[1]的P2EA/P2IE[3:0]	P2IF[3:0]	*软件清零
10	53h	P3口中断	IEN1[2]的P3EA/P3IE[7:0]	P3IF[7:0]	*软件清零
11	5Bh	SPI中断	IEN1[3]的SPIIE	SPSTA[7]的SPIF	软件清零
12	63h	T1捕获中断	T1CAPINT[3:0]	T1CAPINT[7:4]	软件清零
13	6Bh	T2溢出中断	IEN0[4]的T2IE	T2CON[5]的T2IF	中断模式硬件清零
					查询模式软件清零
14	73h	低电压中断	IEN2[6]的LVDIE	LVDCON[6]的LVDIF	中断模式硬件清零
					查询模式软件清零
15	7Bh	预留	预留	预留	-
16	83h	WDT中断	IEN2[0]的WDTIE	INFCON[7]的WDTIF	硬件清零
17	8Bh	T2捕获中断	T2CAPTINT[3:0]	T2CAPINT[7:4]	软件清零
18	93h	ADC中断	IEN2[2]的ADCIE	INFCON[2]的ADCIF	中断模式硬件清零
					查询模式软件清零
19	9Bh	CMP1中断	IEN2[3]的CMP1IE	INFCON[4]的CMP1IF	中断模式硬件清零
					查询模式软件清零
20	A3h	CMP2中断	IEN2[4]的CMP2IE	INFCON[5]的CMP2IF	中断模式硬件清零
					查询模式软件清零
21	ABh	CMP3中断	IEN2[5]的CMP3IE	INFCON[6]的CMP3IF	中断模式硬件清零
					查询模式软件清零

注：*软件清零：外部中断信号未撤除时，软件清零无效。

9.3 中断原理框图



中断结构框图

中断总使能为 SFR 中 IEN0[7] 的 EA，它是各个中断的总开关。当 EA 为 0 时，禁止了所有的中断，即使某个中断源使能为 1 也不会产生相应的中断；开启某些中断时，需要同时置 EA=1，当开启的中断信号发生时，才能产生相应中断。在上电复位后，所有中断使能被清 0。

注意：

1. 中断发生到执行中断子程序所需要的时间不固定。时间的长短与主程序当前执行的指令长度有关。如果主程序在中断发生时执行的单字节，约 4~5 个 CPUCLK 可进入中断；如果是主程序执行的 LJMP，约 12 个 CPUCLK；
2. 当程序进入中断子程序时，硬件自动只保存 PC 指针。如果汇编语言，需要程序员把 PSW、A 等

重要的寄存器内容压栈；如果是基于 Keil C IDE 下编程，Keil C IDE 自动保存了 PSW 和 A 寄存器，其它的寄存器内容需要程序员自己压栈。

9.4 中断优先级

BJ8M302A 把 20 个中断源分为 6 组，每组的优先级可通过优先级控制寄存器 IP1 和 IP0 对应的位赋不同的值决定每组的优先级。每组的 IP1 和 IP0 默认值为 0 时，组与组的优先级相同，当有两组中断请求同时发生时，采查询的方式处理中断服务程序，查询的次序请见中断结构框图中的中断查询次序箭头所示。优先级可通过优先级寄存器软件配置为 4 个等级（【0, 0】：最低；【0, 1】次低；【1, 0】：次高 【1, 1】最高）。在响应一个中断服务程序过程中，可响应更高优先级的中断，不能响应相同或更低优先级的中断。每组内部的优先级相同，当组内有两个中断源同时产生中断请求时，以查询的方式处理中断服务程序。查询的次序请见中断结构框图每组内箭头所示。例如：想设置 UART 接收为最高级的中断源，查看 UART 中断分在第四组。因此可设置为 $IP1|=0x10; IP0|=0x10;$

表 9-4 中断优先级寄存器功能列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BAh	IP0	-	-	IP0[5:0]					
BBh	IP1	-	-	IP1[5:0]					

表 9-3 中断优先级软件配置列表

IP1[x]	IP0[x]	优先级	Level
0	0	低	0
0	1		1
1	0	高	2
1	1		3

上表中的[x]对应中断源组的编号。

注意：在软件配置中断优先级的过程中，建议关闭相应的中断以避免在修改优先级的过程中产生中断。

9.5 中断相关的寄存器

表 9-4 与中断相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
A8h	IEN0	EA	-	-	T2IE	T1IE	P1EA	T0IE	P0EA
B8h	IEN1	-	-	S0RXIE	S0TXIE	SPIIE	P3EA	P2EA	I2CIE
B9h	IEN2	-	LVDIE	CMP3IE	CMP2IE	CMP1IE	ADCIE	-	WDTIE
88h	T0CON	T0CLR	T0M	T0IF	T0EG	T0PSC<2:0>			T0EN
C0h	T1CON	T1CLR	T1M	T1IF	T1EG	T1PSC[2:0]			T1EN
C8h	T2CON	T2CLR	T2M	T2IF	T2EG	T2PSC[2:0]			T2EN
ECh	I2CCON	-	I2CEN	STA	STO	I2CIF	AA	-	-
B1h	SPISTA	SPIF	WCOL	SSERR	MODF	-	-	-	-
98h	S0CON	UARTEN	-	-	UART9	TX9	RX9	TXIF	RXIF
F1h	INFCON	WDTIF	CMP3IF	CMP2IF	CMP1IF	CMP0IF	ADCIF	-	DPS
C7h	LVDCON	LVDF	LVDIF	-	-	LVDS<2:0>			LVDEN
F8h	T1CAPINT	T1C3IF	T1C2IF	T1C1IF	T1C0IF	T1C3IE	T1C2IE	T1C1IE	T1C0IE
D7h	T2CAPINT	T2C3IF	T2C2IF	T2C1IF	-	T2C3IE	T2C2IE	T2C1IE	-
9Eh	T1PWMICON	T1PWM3IF	T1PWM2IF	T1PWM1IF	T1PWM0IF	T1PWM3IE	T1PWM2IE	T1PWM1IE	T1PWM0IE
A1h	T2PWMICON	T2PWM3IF	T2PWM2IF	T2PWM1IF	-	T2PWM3IE	T2PWM2IE	T2PWM1IE	-
B4h	P0IE	-	-	P0IE5	P0IE4	P0IE3	P0IE2	P0IE1	P0IE0
B5h	P1IE	P1IE7	P1IE6	P1IE5	P1IE4	P1IE3	P1IE2	P1IE1	P1IE0
B6h	P2IE	-	-	-	-	P2IE3	P2IE2	P2IE1	P2IE0
B7h	P3IE	P3IE7	P3IE6	P3IE5	P3IE4	P3IE3	P3IE2	P3IE1	P3IE0
F2h	P0IF	-	-	P0IF5	P0IF4	P0IF3	P0IF2	P0IF1	P0IF0
F3h	P1IF	P1IF7	P1IF6	P1IF5	P1IF4	P1IF3	P1IF2	P1IF1	P1IF0
F4h	P2IF	-	-	-	-	P2IF3	P2IF2	P2IF1	P2IF0
F5h	P3IF	P3IF7	P3IF6	P3IF5	P3IF4	P3IF3	P3IF2	P3IF1	P3IF0
BCh	PEGCON0	P3EG[1:0]		P2EG[1:0]		P1EG[1:0]		P0EG[1:0]	
A9h	IOWAKE	-	-	-	-	-	-	-	IOWU

寄存器定义 9-1 IEN0 寄存器定义

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN0	EA	-	-	T2IE	T1IE	P1EA	T0IE	P0EA
A8h	R/W	-	-	R/W	R/W	R/W	R/W	R/W

Bit7 EA:全局中断使能位

- 1:使能中断
- 0:禁止中断

Bit6-5 未使用

Bit4T2IE: Timer2 中断使能位

- 1:使能 Timer2 中断
- 0:禁止 Timer2 中断

Bit3T1IE: Timer1 中断使能位

- 1:使能 Timer1 中断
- 0:禁止 Timer1 中断

Bit2 P1EA: P1 口中断使能位

- 1:使能 P1 口中断
- 0:禁止 P1 口中断

Bit1T0IE: Timer0 中断使能位

- 1:使能 Timer0 中断
- 0:禁止 Timer0 中断

Bit0 POEA: P0 口中断使能位

- 1: 使能 P0 口中断
- 0: 禁止 P0 口中断

寄存器定义 9-2 IEN1 寄存器定义

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN1	-	-	SORXIE	SOTXIE	SPIIE	P3EA	P2EA	I2CIE
B8h	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-6 未使用

Bit5SORXIE: UART 通讯中的收码完毕中断使能位

- 1: 开启 UART 通讯收码完毕后申请中断
- 0: 禁止 UART 通讯收码完毕后申请中断

Bit4SOTXIE: UART 通讯中的发码完毕中断使能位

- 1: 开启 UART 通讯发送数据完毕后申请中断
- 0: 禁止 UART 通讯发送数据完毕后申请中断

Bit3SPIIE: SPI 通讯中断使能位

- 1: 开启 SPI 通讯申请中断
- 0: 禁止 SPI 通讯申请中断

Bit2P3EA: P3 口中断使能位

- 1: 使能 P3 口中断
- 0: 禁止 P3 口中断

Bit1P2EA: P2 口中断使能位

- 1: 使能 P2 口中断
- 0: 禁止 P2 口中断

Bit0I2CIE: IIC 中断使能位

- 1: 使能 IIC 口中断
- 0: 禁止 IIC 口中断

寄存器定义 9-3 IOWAKE IO 口唤醒

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOWAKE	-	-	-	-	-	-	-	IOWU
A9h	-	-	-	-	-	-	-	R/W

Bit7-1 未使用

Bit0IOWU: IO 翻转换醒使能位

- 1: 打开 IO 翻转换醒
- 0: 禁止 IO 翻转换醒

寄存器定义 9-4 PEGCON0 IO 口中断触发方式选择控制寄存器

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PEGCON0	P3EG[1:0]		P2EG[1:0]		P1EG[1:0]		P0EG[1:0]	
BCh	R/W		R/W		R/W		R/W	

Bit7-6P3EG: P3 口中断触发方式选择位

- 00: 低电平触发
- 01: 下降沿触发 (默认)
- 10: 上升沿触发
- 11: 边沿触发 (上升沿和下降沿均触发)

Bit5-4P2EG: P2 口中断触发方式选择位

- 00: 低电平触发
- 01: 下降沿触发 (默认)
- 10: 上升沿触发
- 11: 边沿触发 (上升沿和下降沿均触发)

Bit3-2P1EG: P1 口中断触发方式选择位

- 00: 低电平触发
- 01: 下降沿触发 (默认)
- 10: 上升沿触发
- 11: 边沿触发(上升沿和下降沿均触发)

Bit1-0POEG: P0 口中断触发方式选择位

- 00: 低电平触发
- 01: 下降沿触发 (默认)
- 10: 上升沿触发
- 11: 边沿触发(上升沿和下降沿均触发)

寄存器定义 9-5 P0IE P0 口各个口中断允许控制寄存器

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0IE	-	-	P0IE5	P0IE4	P0IE3	P0IE2	P0IE1	P0IE0
B4h	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-6 未使用

Bit5-0P0IE_x:P0 口各个口中断允许控制寄 x=0-5

- 1: 允许 P0. x 口中断
- 0: 禁止 P0. x 口中断

寄存器定义 9-6 P0IF P0 各个 IO 口中断申请标志位寄存器

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0IF	-	-	P0IF5	P0IF4	P0IF3	P0IF2	P0IF1	P0IF0
F2h	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-6 未使用

Bit5-0P0IF_x:P0 口各个口中断标志位寄存器 x=0-5

- 1: P0. x 口产生中断
- 0: P0. x 口没有差生中断

同理: P1IE/P2IE/P3IE 分别为 P1、P2、P3 口中断允许寄存器
P2IF/P2IF/P3IF 分别为 P1、P2、P3 口中断标志位寄存器。

备注: 当相应 P_nIE_x 有效时, P_nIF_x 才有意义, 否则 P_nIF_x 读为 0。 (n:0-3 x:0-7)

9.6 外部中断

中断模式:

使用 GPIO 的外部中断功能时, 不同组的外部中断 (P0/1/2/3) 对应不同的中断入口地址, P0 的入口地址为 0x0003, P1 的入口地址为 0x0013, P2 的入口地址为 0x004B, P3 的入口地址为 0x0053。P0、P1、P2、P3 四个组分别有 6、8、4、8 个管脚, 当使用同一组管脚时, 由于中断入口地址相同, 所以必须通过判断中断标志位来确定是哪个管脚发生了中断。

当选择低电平触发方式, 发生外部中断时, 中断标志位将保持 1 直至相应 IO 管脚低电平撤销, 软件无需清零也无法清零。在中断子程序执行过程中, 如果不撤销该 IO 管脚低电平条件, 当中断子程序执行完成后, 将再次进入中断。中断子程序可以通过判断相应中断标志位是否清零, 来判断相应 IO 管脚低电平条件是否已经撤销。

当选择沿触发方式, 发生外部中断时, 中断标志位将置 1, 需要软件清零, 中断标志位只可写 0, 写 1 无效。执行中断子程序时, 需要清除中断标志位, 否则当中断子程序执行完成后, 将再次进入中

断。

当选择外部中断为低电平触发时，系统正常运行时，低电平的脉宽必须大于1个当前的系统CPUCLK周期；在IDLE模式下，低电平的脉宽必须大于3个当前的系统CPUCLK周期；在STOP模式下，低电平的脉宽必须大于4个当前的系统CPUCLK周期。

注意，有两种情况可能会产生错误标志位，建议按照步骤使用外部中断：

- ◆ PnIE_x(n:0-3 x:0-7)打开状态时，软件配置触发模式寄存器 PEGCON0，切换不同的触发方式；
- ◆ 如果IO口处于输出状态，设置外部中断使能寄存器 PnIE_x，打开外部中断。

中断模式操作：

- 1) 设置外部中断使能位 PnIE_x，使能相应外部中断管脚（管脚无需软件配置 PnOE 就切换为输入状态）；
- 2) 软件配置触发模式 PEGCON0，选择相应外部中断的触发方式（可省，默认为下降沿触发）；
- 3) 软件配置外部中断优先级 IP1/IP0（可省，默认为 Level 0）；
- 4) 设置相应IO所对应的那组（P0/1/2/3）外部中断使能 PnIE（n:0-3）；
- 5) 清零标志状态位 PnIF_x（在操作步骤1或2后，可能产生错误标志位，建议严格按照步骤操作）；
- 6) 设置全局中断使能 EA；
- 7) 当发生外部中断时，程序会跳转到相应的中断入口地址，执行中断子程序。

查询模式：

使用外部中断查询模式时，必须设置相应中断使能位 PnIE_x，否则中断标志位 PxIF_x 没有意义，读为零。同时，需要关闭中断功能（EA 或 EXnIE 清零 n:0-3）。软件通过查询 PxIF_x，判断哪个口发生外部中断。

当选择低电平触发方式，发生外部中断时，中断标志位将保持1直至相应IO管脚低电平撤销，软件无需清零也无法清零。当选择沿触发方式，发生外部中断时，中断标志位将置1，需要软件清零，中断标志位只可写0，写1无效。无论选择哪种触发方式，都不会进入中断子程序。

查询模式操作：

- 1) 设置外部中断使能位 PnIE_x，使能相应外部中断管脚（管脚无需软件配置 PxOE 就切换为输入状态）；
- 2) 软件配置触发模式 PEGCON0，选择相应外部中断的触发方式（可省，默认为下降沿触发）；
- 3) 清零 EA 或 EXnIE(n:0-3)，关闭中断功能；
- 4) 清零标志状态位 PnIF_x（在操作步骤1或2后，可能产生错误标志位，建议严格按照步骤操作）；
- 5) 软件读取 PnIF_x 寄存器，若置1则表示发生外部中断，此时，不进入中断子程序。

外部中断唤醒：

使用外部中断的中断模式，选择低电平方式，可以唤醒IDLE或者STOP状态；若选择沿触发方式，只能唤醒IDLE状态。

使用外部中断的中断模式，选择低电平方式唤醒 STOP 状态时，需要注意，唤醒源的中断优先级必须高于当前正在处理的中断（中断嵌套时）的优先级，否则无法唤醒。如果当前正在运行的指令不是中断子程序，则任何优先级均可以唤醒。例如，当正在处理优先级位 Level 2 的中断子程序，进入 STOP 状态，那么唤醒当前 CPU 的 STOP 状态的外部中断优先级必须是 Level 3，其他优先级均无法唤醒 STOP。

在 IDLE 模式下，低电平的脉宽必须大于 3 个当前的系统 CPUCLK 周期；在 STOP 模式下，低电平的脉宽必须大于 4 个当前的系统 CPUCLK 周期。

10.Timer0 模块

Timer 0(定时/计数器 0) 模块可以用作 8 位定时器或 8 位计数器。

RTCC 模块与 T0 共用部分硬件电路, 当作为定时器或者计数器时, RTCC 功能不能使用。

定时器模式:

T0 模块的时钟来源于 CUPCLK(详情见: 图 4-1), 由软件配置 TOPSC[2:0] 预分频, 设置 TOM 为 0, 让预分频的时钟进入 TOC 开始向上计数(默认值为 0x00), 在 TOEN 打开的情况下, 计数到 T0OVR 时产生溢出中断。

计数器模式:

根据 TOEG 的配置值, 对端口 P1.7 的输入信号的上升沿或下降沿进行计数(TOEG 配置上升沿或下降沿的触发方式)。计数器从 0 开始向上计数, 计数到 T0OVR 值时产生溢出中断。

10.1 T0 工作原理

图 10-1 Timer0 框图

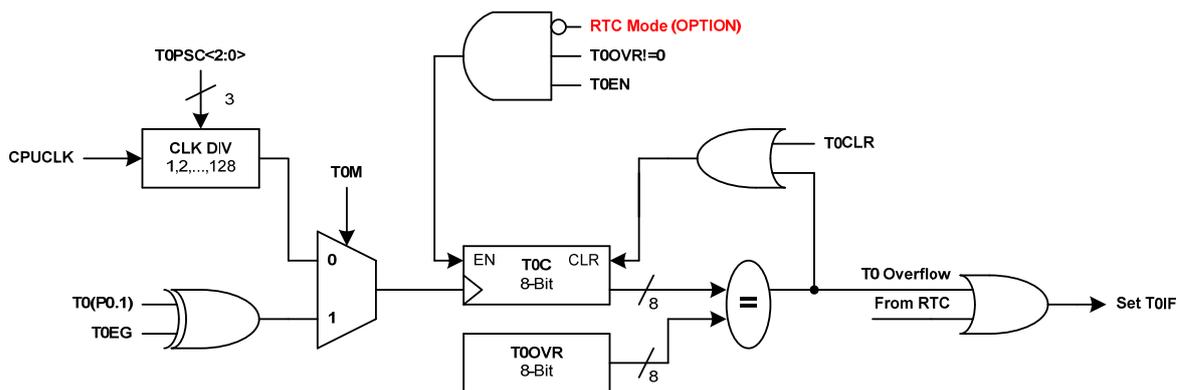


表 10-1 与 Timer0 相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
88h	TOCON	TOCLR	TOM	T0IF	TOEG	TOPSC<2:0>		TOEN	
89h	T0OVR	T0OVR<7:0>							
8Ah	TOC	TOC<7:0>							
A8h	IEN0	EA	-		T2IE	T1IE	P1EA	T0IE	P0EA

寄存器定义 10-1 TOCON Timer0 控制寄存器

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TOCON	TOCLR	TOM	T0IF	TOEG	TOPSC<2:0>		TOEN	
88h	W	R/W	R/W	R/W	R/W		R/W	

Bit7 TOCLR: 计数寄存器 TOC 清 0 使能位

1: 设置为计数寄存器 TOC 到达预设值时自动清 0。该位在写操作完成时由硬件自动清零。读永远为 0

0: 设置为手动对定时器 TOC 寄存器清 0

Bit6 TOM: 定时/计数器 0 模式选择位

1: 计数器模式, P0.1 作为计数时钟口

0: 定时器模式

Bit5 TOIF: T0 溢出标志位

1: 当定时计数溢出时, 此位会被硬件置 1, 需软件清 0。

0: T0 未溢出

Bit4 TOEG: 计数器的有效沿选择位

1: 下降沿计数

0: 上升沿计数

Bit3-1 TOPSC<2:0>: 定时/计数器时钟选择

000: CPUCLK

001: CPUCLK /2

010: CPUCLK /4

011: CPUCLK /8

100: CPUCLK /16

101: CPUCLK /32

110: CPUCLK /64

111: CPUCLK /128

Bit0 TOEN: 定时/计数器 0 使能位

1: 使能定时/计数器 0

0: 禁止定时/计数器 0

寄存器 TOC (8Ah) 可读写, 是 Timer0 计数寄存器

寄存器 T0OVR (89h) 可读写, 是 Timer0 溢出寄存器

10.2 T0 定时器

定时器模块的输入时钟为 CPUCLK。在定时器模块集成了一个分频器, 分频的时钟 TOCLK 作为 8 bits 计数器的输入时钟。当用户设置了定时器模块的使能标志, 8 bits 计数器将启动, 将会从 0 递增到 T0OVR。用户需要设置 T0OVR (定时器模块中断信号选择器) 以选择定时超时中断信号。当定时超时发生时, 中断标志位会自设置, 程序计数器会跳转到 0Bh 以执行中断服务程序。

操作步骤:

- 1) 设置 TOM, 设置为定时器;
- 2) 设置 TOPSC[2:0], 选择时钟预分频;
- 3) 设置 TOCLR 为 1, 选择 T0 中断后 TOC 计时器自动清 0;
- 4) 设置 T0OVR, 选择定时器溢出值;
- 5) 设置寄存器标志位: TOIE 与 EA, 使能定时器中断。
- 6) 设置寄存器标志位: TOEN, 使能定时器。
- 7) 当定时超时发生时, 程序计数器会跳转到 00BH。

定时器 0 溢出时间计算方法:

定时器 0 溢出时间 = (T0OVR[7:0] + 1) / TOCLK

例 10-1 T0 定时器示例

```
/******  
* Function   : TO_TIMER_init  
* Description: Timer0 定时模式  
* Input     :  
* Output    :  
*****/  
void TO_TIMER_init()  
{  
    TOCON &= 0xF1;  
}
```

```

    TOCON |= 0xE0; //128分频
    TOOVR = 0x80; //配置定时器溢出值 (128+1)/TOCLK
    TOM = 0; //设置为定时器模式
TOCLR = 1; //清 TOC
    TOIE = 1; //使能 TO 中断
    EA = 1; //使能总中断
TOEN = 1; //使能 TO
}

/*****
* Function : TO_IRQHandler
* Description: 定时器 0 中断响应函数
* Input :
* Output :
*****/
void TO_IRQHandler(void) interrupt 1
{
    .....
}

```

10.3T0 计数器

计数器时钟模块的输入为 P0.1 口。在定时/计数器模块集成了一个分频器，分频的时钟 TOCLK 作为 8 bits 计数器的输入时钟。当用户设置了定时/计数器模块的使能标志，8 bits 计数器将启动，将会从 0 递增到 TOOVR。用户需要设置 TOOVR（定时器模块中断信号选择器）以选择定时超时中断信号。当定时超时发生时，中断标志位会自设置，程序计数器会跳转到 0bh 以执行中断服务程序。

操作步骤：

- 1) 设置 TOM 为 1，设置为计数器模式；（注意：计数次数=TOOVR+1）
- 2) 设置 TOEG，配置计数器边沿触发方式；
- 3) 设置 TOCLR 为 1，选择 T0 中断后 TOC 计数器自动清 0；
- 4) 设置 TOOVR，选择计数器溢出值；
- 5) 设置寄存器标志位：TOIE 与 EA，使能计数器中断。
- 6) 设置寄存器标志位：TOEN，使能计数器。
- 7) 当计数超时发生时，程序计数器会跳转到 00BH。

定时器 0 溢出时间计算方法：

定时器 0 溢出时间= (TOOVR[7:0]+1) /TOCLK

例 10-2 T0 计数器示例

```

/*****
* Function : TO_TIMER_init
* Description: Timer0 计数模式
* Input :
* Output :
*****/
void TO_TIMER_init()
{
    TOCON &= 0xF1;
    TOCON |= 0xE0; //128分频
    TOOVR = 0x80; //配置定时器溢出值 (128+1)/TOCLK
}

```

```
TOM = 1;           //设置为计数器模式
TOEG = 1;         //设置为下降沿计数, P0.1 为计数时钟口
TOCLR = 1;       //清 TOC
TOIE = 1;        //使能 TO 中断
EA = 1;          //使能总中断
TOEN = 1;        //使能 TO
}

/*****
* Function   : TO_IRQHandler
* Description: 定时器 0 中断响应函数
* Input      :
* Output     :
*****/
void TO_IRQHandler(void) interrupt 1
{
    .....
}
```

11.Timer1 模块

Timer1 的时钟 TIMERCLK 可软件配置来自于 TCK，也可来自于 MCK (TCKEN=0 时)。如图 4-1 简化的 MCU 时钟源框所示。

Timer1 模块的时钟默认来源于独立的 TCK 时钟模块 (Timer Clock Module), 频率通过配置 TCKSEL 项来选择 48M (默认) 还是 32M。当将 TCKEN 设置为 0 后, 则 Timer1 的时钟便来自于 MCK, 同时芯片内部的 TCK 时钟被关掉, 降低功耗。

内部定时计数器为 16 位宽。内部还集成了其它辅助模块, 通过灵活的软件配置, 除了作为普通的定时计数外, 也可以用作捕捉功能 或者 T1PWM 功能。

11.1T1 定时计数器工作原理

图 11-1Timer1 模块框图

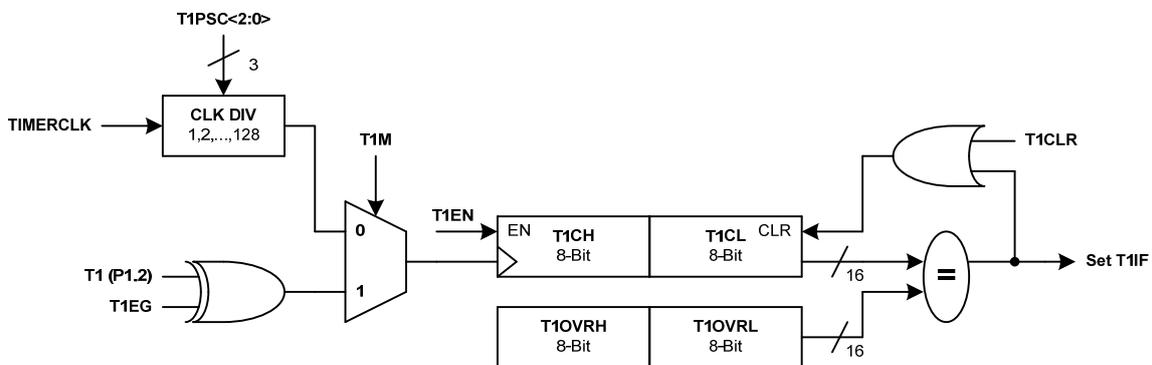


表 11-1 与 Timer1 相关的寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
A8h	IEN0	EA	-	-	T2IE	T1IE	P1EA	T0IE	P0EA
C0h	T1CON	T1CLR	T1M	T1IF	T1EG	T1PSC[2:0]		T1EN	
C4h	T1CH	T1C[15:8]							
C3h	T1CL	T1C[7:0]							
C2h	T1OVRH	T1OVR[15:8]							
C1h	T1OVRL	T1OVR[7:0]							

寄存器定义 11-1 T1CON T1 控制寄存器

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CON	T1CLR	T1M	T1IF	T1EG	T1PSC[2:0]		T1EN	
C0h	W	R/W	R/W	R/W	R/W		R/W	

Bit7 T1CLR: 计数寄存器 T1CH&T1CL 清 0 使能位

1: 设置为计数寄存器 T1C 到达预设值时自动清 0。该位在写操作完成时由硬件自动清零。读永远为 0

0: 设置为手动对定时器 T1C 寄存器清 0

Bit6 T1M: 定时/计数器 1 模式选择位

1: 计数器模式

0: 定时器模式

Bit5 T1IF: T1 溢出标志位

1: 当定时计数溢出时, 此位会被硬件置 1。中断方式时自动清零; 查询方式时软件清 0。

0: T1 未溢出

Bit4 T1EG: 计数器的有效沿选择位

- 1: 下降沿计数
- 0: 上升沿计数

Bit3-1 T1PSC<2:0>: 定时/计数器时钟选择

- 000: TIMERCLK
- 001: TIMERCLK/2
- 010: TIMERCLK/4
- 011: TIMERCLK/8
- 100: TIMERCLK/16
- 101: TIMERCLK/32
- 110: TIMERCLK/64
- 111: TIMERCLK/128

Bit0 T1EN: 定时/计数器 1 使能位

- 1: 使能定时/计数器 1
- 0: 禁止定时/计数器 1

寄存器 T1CH(C4h)&T1CL(C3h) 可读写, 是 Timer1 计数寄存器

寄存器 T1OVRH(C2h)&T1OVR(L1h) 可读写, 是 Timer1 溢出寄存器

定时器操作:

定时器 1 模块的输入为 TIMERCLK, 即 TCK (TCKEN=1) 或 MCK (TCKEN=0)。TCK 最快可达到 48MHz。可以通过代码选项选择到 32MHz。在定时器模块集成了一个分频器, 分频后的时钟作为 16 bits 计数器的输入时钟。当用户设置了定时器模块的使能标志, 16 bits 计数器将启动, 将会从 0 递增到 T1OVR。用户需要设置 T1OVR 以选择定时超时中断信号。当定时超时发生时, 如果有开中断使能的话, 中断标志位会自动置 1, 程序计数器会跳转到 1Bh 以执行中断服务程序。

操作步骤:

- 1) 设置 T1PSC[2:0], 选择 T1CLK 的值作为定时器/计数器时钟。
- 2) 设置 T1M, 软件配置为定时器模式。
- 3) 设置 T1OVR, 选择定时器溢出值。
- 4) 设置寄存器标志位: T1IE 与 EA, 使能定时器中断。
- 5) 设置寄存器标志位: T1EN, 使能定时器。
- 6) 当定时超时发生时, 程序计数器会跳转到 1BH。

定时器 1 溢出时间计算方法:

定时器 1 溢出时间 = (T1OVR[15:0]+1) / T1CLK

计数器操作:

工作于 16 位重加载计数器模式时, 根据 T1EG 的软件配置值, 对端口 P1.2 的输入信号的上升沿或下降沿进行计数。TIMER1 开始工作于 16 位重加载计数器模式, 计数器由 0 开始, 计数到 T1OVR[15:0] 溢出值产生中断。

操作步骤:

- 1) 设置 T1PSC[2:0], 选择 T1CLK 的值作为定时器/计数器时钟。
- 2) 设置 T1M, 软件配置为计数器模式。
- 3) 设置 T1OVR, 选择定时器溢出值。
- 4) 设置 T1EG, 选择上升沿还是下降沿进行计数。
- 5) 设置寄存器标志位: T1IE 与 EA, 使能定时器中断。
- 6) 设置寄存器标志位: T1EN, 使能定时器。

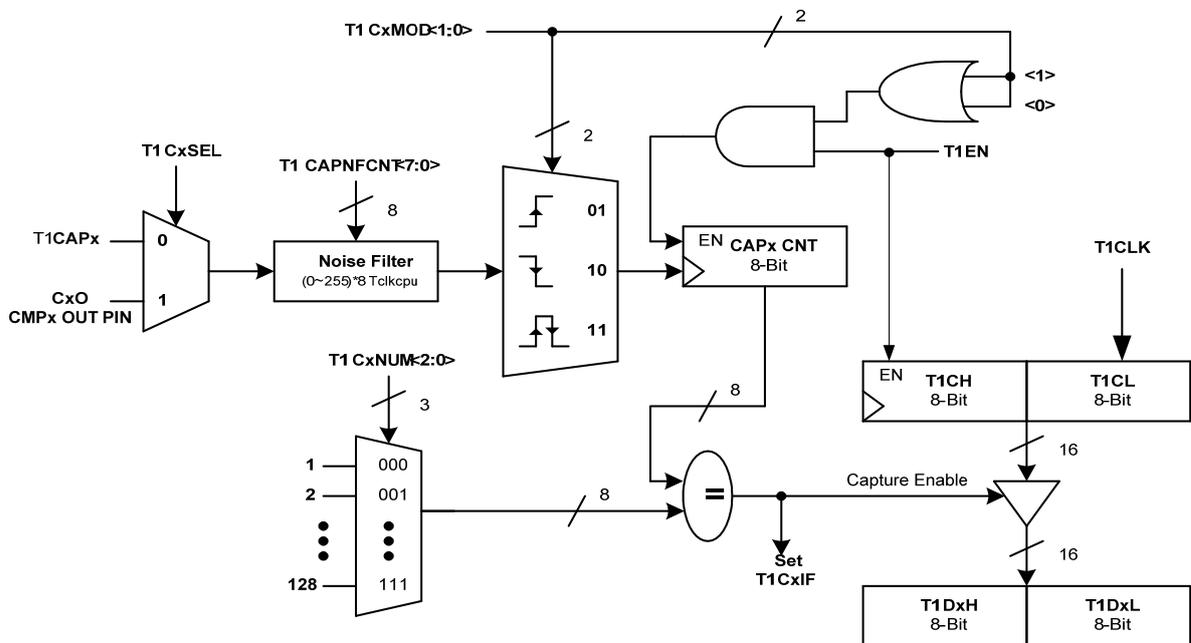
7) 当定时超时发生时，程序计数器会跳转到 1BH。

11.2 T1 捕获功能(4 路)

捕获功能是指该芯片对事件持续的时间计时。定时器 1 共有 4 路捕获，分别从 T1CAP0 (P3.1 [默认]/P3.2)、T1CAP1 (P3.4 [默认]/P1.7)、T1CAP2 (P3.3 [默认]/P1.0)、T1CAP3 (P2.3 [默认]/P1.6) 输入。该模式能够软件配置捕获计数时钟的快慢，软件配置需要捕获的沿的个数，并选择捕获下降沿或者上升沿。捕获完成后，相应的完成标志会置 1，若使能了中断，将产生中断请求。定时器 1 的 4 个捕获中断共用一个中断入口地址 63H。

注意：捕获输入引脚 T2CAPx 与 T1PWMx 输出管脚共用，具体的引脚映射选择需通过程序配置来确定。详情请查阅 26. 程序配置 章节。

图 11-2 捕获结构框图



注意：棕色背景寄存器均位于 SFR 列表第 0 页，需要设置 PSW[5]=0 进行访问。

11.5.1 与捕获相关的寄存器

表 11-2 与捕获相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
A8h	IEN0	EA	-	-	T2IE	T1IE	P1EA	T0IE	P0EA
D1h	T1CAPCON	T1C3MOD<1:0>		T1C2MOD<1:0>		T1C1MOD<1:0>		T1C0MOD<1:0>	
D2h	T1CAPNUM0	T1C1SEL	T1C1NUM<2:0>			T1C0SEL	T1C0NUM<2:0>		
D3h	T1CAPNUM1	T1C3SEL	T1C3NUM<2:0>			T1C2SEL	T1C2NUM<2:0>		
D4h	T1CAPNFCNT	T1CAPNFCNT<7:0>							
F8h	T1CAPINT	T1C3IF	T1C2IF	T1C1IF	T1C0IF	T1C3IE	T1C2IE	T1C1IE	T1C0IE

寄存器定义 11-2 T1CAPCON 捕获模式选择寄存器

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CAPCON	T1C3MOD<1:0>		T1C2MOD<1:0>		T1C1MOD<1:0>		T1C0MOD<1:0>	
D1h	R/W		R/W		R/W		R/W	

Bit7-6T1C3MOD<1:0>:CAP3 模式选择。

- 00: 使能 PWM 模式, 不使能捕获模式。
- 01: 使能捕获模式, 并且是上升沿捕获(关闭 PWM 功能)
- 10: 使能捕获模式, 并且是下降沿捕获(关闭 PWM 功能)
- 11: 使能捕获模式, 并且是双沿捕获(关闭 PWM 功能)

Bit5-4T1C3MOD<1:0>:CAP2 模式选择

- 00: 使能 PWM 模式, 不使能捕获模式。
- 01: 使能捕获模式, 并且是上升沿捕获(关闭 PWM 功能)
- 10: 使能捕获模式, 并且是下降沿捕获(关闭 PWM 功能)
- 11: 使能捕获模式, 并且是双沿捕获(关闭 PWM 功能)

Bit3-2T1C3MOD<1:0>:CAP1 模式选择

- 00: 使能 PWM 模式, 不使能捕获模式。
- 01: 使能捕获模式, 并且是上升沿捕获(关闭 PWM 功能)
- 10: 使能捕获模式, 并且是下降沿捕获(关闭 PWM 功能)
- 11: 使能捕获模式, 并且是双沿捕获(关闭 PWM 功能)

Bit1-0T1C3MOD<1:0>:CAPO 模式选择

- 00: 使能 PWM 模式, 不使能捕获模式。
- 01: 使能捕获模式, 并且是上升沿捕获(关闭 PWM 功能)
- 10: 使能捕获模式, 并且是下降沿捕获(关闭 PWM 功能)
- 11: 使能捕获模式, 并且是双沿捕获(关闭 PWM 功能)

寄存器定义 11-3T1CAPNUM0

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CAPNUM0	T1C1SEL	T1C1NUM<2:0>			T1C0SEL	T1C0NUM<2:0>		
D2h	R/W	R/W			R/W	R/W		

Bit7T1C1SEL:T1 捕获 1 输入选择

- 1: CMP1 输出
- 0: IO 口输入。P3.4 或者 P1.7 输入的选择请见 26. 程序配置章节。(默认 P3.4 输入)

Bit6-4T1C1NUM<2:0>:T1 捕获 1 的沿(下降沿/上升沿)的捕获个数软件配置

- 000: 1
- 001: 2
- 010: 4
- 011: 8
- 100: 16
- 101: 32
- 110: 64
- 111: 128

Bit3T1C0SEL:T1 捕获 0 输入选择

- 1: CMP0 输出
- 0: IO 口输入。P3.1 或者 P3.2 输入的选择请见 26. 程序配置章节。(默认 P3.1 输入)

Bit2-0T1C0NUM<2:0>:T1 捕获 0 的沿(下降沿/上升沿)的捕获个数软件配置

- 000: 1
- 001: 2
- 010: 4
- 011: 8
- 100: 16
- 101: 32
- 110: 64
- 111: 128

寄存器定义 11-4T1CAPNUM1

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CAPNUM1	T1C3SEL	T1C3NUM<2:0>		T1C2SEL	T1C2NUM<2:0>			
D3h	R/W	R/W		R/W	R/W			

Bit7T1C1SEL:T1 捕获 3 输入选择

1: CMP3 输出

0: I/O 口输入。P2.3 或者 P1.6 输入的选择请见 26. 程序配置章节。（默认 P2.3 输入）

Bit6-4T1C1NUM<2:0>:T1 捕获 3 的沿（下降沿/上升沿）的捕获个数软件配置

- 000: 1
- 001: 2
- 010: 4
- 011: 8
- 100: 16
- 101: 32
- 110: 64
- 111: 128

Bit3T1C0SEL:T1 捕获 2 输入选择

1: CMP2 输出

0: I/O 口输入。P3.3 或者 P1.0 输入的选择请见 26. 程序配置章节。（默认 P3.3 输入）

Bit2-0T1C0NUM<2:0>:T1 捕获 2 的沿（下降沿/上升沿）的捕获个数软件配置

- 000: 1
- 001: 2
- 010: 4
- 011: 8
- 100: 16
- 101: 32
- 110: 64
- 111: 128

寄存器定义 11-5T1CAPINT

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CAPINT	T1C3IF	T1C2IF	T1C1IF	T1C0IF	T1C3IE	T1C2IE	T1C1IE	T1C0IE
F8h	R/W							

Bit7 T1C3IF:CAP3 中断标志位

Bit6T1C2IF:CAP2 中断标志位

Bit5T1C1IF:CAP1 中断标志位

Bit4T1C0IF:CAPO 中断标志位

Bit3T1C3IE:CAP3 中断允许标志位

Bit2T1C2IE:CAP2 中断允许标志位

Bit1T1C1IE:CAP1 中断允许标志位

Bit0T1C0IE:CAPO 中断允许标志位

寄存器 T1CAPNFCNT (D4h) 可读写, Noise filter cnt。总的滤波时间为(0~ff) × 8T_{clkcpu}

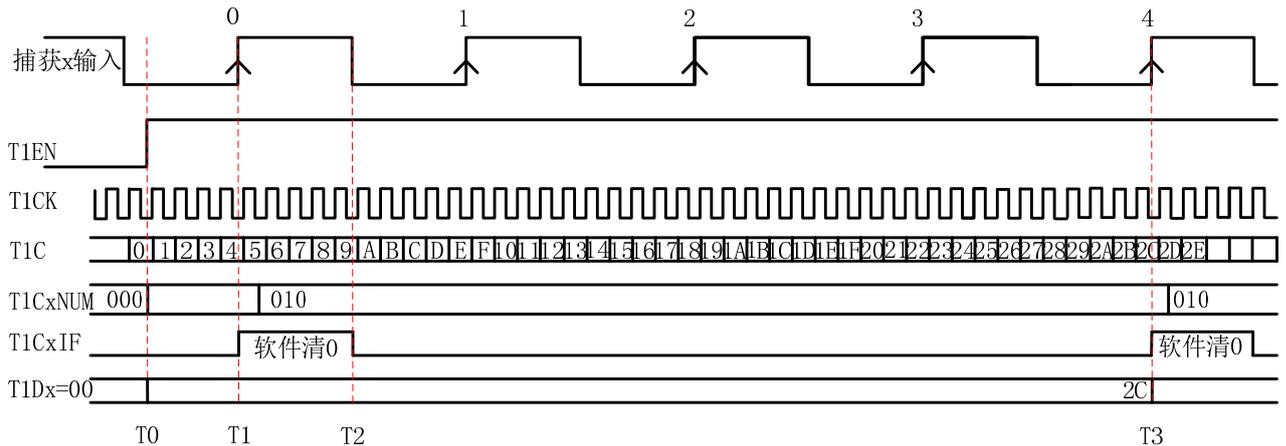
11.5.2 软件配置方法

捕获 x (x=1~4) 的软件配置步骤:

1. 程序配置 T1 的时钟 TIMERCLK, 即 TCK (TCKEN=1) 或 MCK (TCKEN=0), TCK 可以为 48M 或者 32M(程序配置详情请见 26. 程序配置章节列表中的 TCKSEL 选项);
2. 如果捕捉信号是 T1CAPx 输入信号源 (捕捉输入的管脚程序配置请见 26. 程序配置章节列表中的 T1PWMx 选项);
3. 设置 T1PSC<2:0>确定 T1 时钟分频;
4. 设置 T1M=0; 选择 T1 的时钟源

5. 设置 T1CxSEL 软件配置捕捉信号源。
6. 设置 T1CAPNFCNT 噪音滤波时间
7. 设置 T1CxMOD<1:0>触发的方式
8. 设置 T1CxNUM<2:0>一次捕捉触发信号的个数
9. 打开中断允许标志 T1CxIE=1（如果总中断没有打开，请首先打开 EA=1）
10. 打开开始计时 T1EN=1

图 11-3 捕获 x (x=1~4) 的示意图



脉冲周期测量:

示意图中，软件配置捕获的边沿的个数 A=4，即 T1CxNUM<2:0>=010。为了表示方便，只显示了 16 位计数值的低 8 位。

由于被捕获波形输入边沿的时间的不确定性，使能 T1EN 可能大大早于被捕获的边沿，第一次产生计数是完全不准确的。为了解决这个问题，可以采用下述方法：

软件配置捕获的边沿的个数为一个，即 T1CxNUM<2:0>=000。当 t=T0 时，使能 T1EN，定时器即开始计数，当 t=T1 时，第 1 个边沿到来，即第一次捕获完成后，读取 T1Dx[16:0]=04h 的值作为初始值并清除 T1CxIF。此时不关闭捕获，软件配置 T1CxNUM<2:0>=010。其它沿用之前软件配置，当 t=T3 时，第 4 个沿到来，捕获再次完成，读取此时 T1Dx[16:0]=2Ch 的值作为完成值。将捕获完成值减去捕获初始值，即得到被捕获波形的 A 个周期内，T1CK 的个数。如图示脉冲宽度为 2Ch-04h=28h。若依次循环测量，即可跟踪被捕获波形的周期变化。

注意：

1. 为了防止定时器溢出，用户需要注意被捕获的波形周期与计数时钟 T1CK 周期的配合；
2. 在被捕获的波形输入之前，不要过早的使能 T1EN；
3. 为了让捕获计数更精确，推荐用数倍于被捕获波形频率的 T1CK 采样；
4. 增加捕获的边沿的个数（A 的整数倍），再进行平均计算。

电平宽度测量:

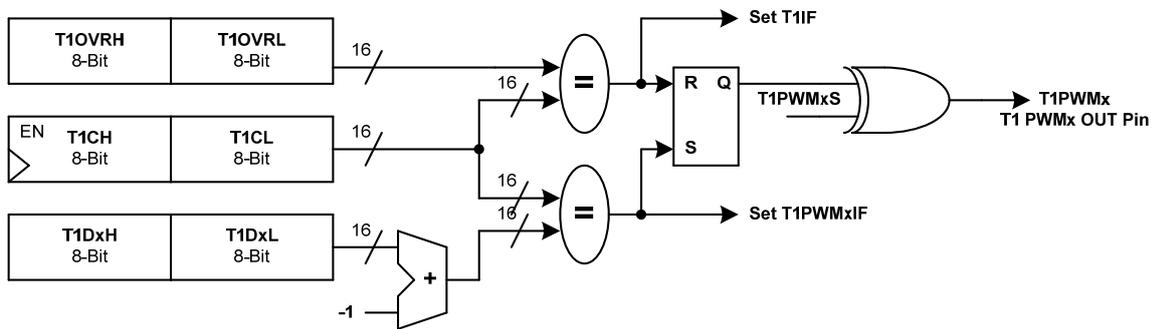
软件配置捕获的边沿的个数为一个，即 T1CxNUM<2:0>=000。当 t=T0 时，使能 T1EN，定时器即开始计数，当 t=T1 时，第 1 个边沿到来，即第一次捕获完成后，读取 T1Dx[16:0]=04h 的值作为初始值并清除 T1CxIF。此时不关闭捕获，改变触发方式，即软件配置 T1CxMOD<1:0>=10，其它沿用之前软件配置，当 t=T2 时，捕获再次完成，读取此时 T1Dx[15:0]=09h 的值作为完成值。将捕获完成值减去捕获初始值，即得到被捕获高电平宽度时间，高电平宽度为 09h-04h=05h。若依次循环测量，即

可跟踪被捕获波形的周期变化。

11.3 T1PWM 脉宽调制

定时器 1 有 4 路 PWM 输出(输出引脚分别为: T1PWM0/T1PWM1/P1PWM2/TIPWM3。输出引脚可通过程序配置确定(详情请见 26. 程序配置章节), 这 4 路 T1PWM 都没有互补输出。4 路 T1PWM 的周期大小通过软件配置 T1PSC[2:0]和 T1OVR[15:0]来实现, T1PWM 输出占空比由 T1D0[15:0]来调整, PWM1/2/3 占空比由 T1D1/2/3[15:0]来调整。这 4 路 T1PWM 都只有边沿对齐模式, 没有中心对齐和没有互补输出。

图 11-3T1PWM 原理框图



注意:

对输出 PWM 控制的 T1PWMCON、T2PWMCON1、T2PWMCON2 三个只能写的寄存器操作, 禁用与、或、非直接操作。如果非得需要与或非操作, 请先自己定义相应的 RAM, 再与自定义 RAM 进行与或非操作后, 然后直接赋值给以上三个寄存器即可。

Timer1 PWM 输出 (4 路)

T1PWM 的周期 = $(T1OVR[15:0]+1) / T1CLK$.

T1PWMx 的有效电平宽度比 = $T1Dx[15:0] / (T1OVR[15:0]+1)$

表 11-3 与 T1 PWM 相关的寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
A8h	IEN0	EA	-	-	T2IE	T1IE	P1EA	T0IE	POEA
C0h	T1CON	T1CLR	T1M	T1F	T1EG	T1PSC[2:0]			T1EN
91h	T1PWMCON	T1PWM3S	T1PWM2S	T1PWM1S	T1PWM0S	T1PWM3EN	T1PWM2EN	T1PWM1EN	T1PWM0EN
9Eh	T1PWMICON	T1PWM3IF	T1PWM2IF	T1PWM1IF	T1PWM0IF	T1PWM3IE	T1PWM2IE	T1PWM1IE	T1PWM0IE
C4h	T1CH	T1C[15:8]							
C3h	T1CL	T1C[7:0]							
C2h	T1OVRH	T1OVR[15:8]							
C1h	T1OURL	T1OVR[7:0]							
92h	T1D0H	T1D0[15:8]							
93h	T1D0L	T1D0[7:0]							
94h	T1D1H	T1D1[15:8]							
95h	T1D1L	T1D1[7:0]							
96h	T1D2H	T1D2[15:8]							
97h	T1D2L	T1D2[7:0]							
9Ch	T1D3H	T1D3[15:8]							
9Dh	T1D3L	T1D3[7:0]							

寄存器定义 11-6T1PWMCON T1 作为 PWM 时控制寄存器

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1PWMCON	T1PWM3S	T1PWM2S	T1PWM1S	T1PWM0S	T1PWM3EN	T1PWM2EN	T1PWM1EN	T1PWM0EN
91h	W	W	W	W	W	W	W	W

Bit7T1PWM3S: PWM3 输出有效电平选择位

- 1: 低电平有效
- 0: 高电平有效

Bit6 T1PWM2S: PWM2 输出有效电平选择位

- 1: 低电平有效
- 0: 高电平有效

Bit5T1PWM1S: PWM1 输出有效电平选择位

- 1: 低电平有效
- 0: 高电平有效

Bit4T1PWM0S: PWM0 输出有效电平选择位

- 1: 低电平有效
- 0: 高电平有效

Bit3T1PWM3EN: PWM3 使能位

- 1: 使能 PWM3 输出
- 0: 关闭 PWM3 输出

Bit2T1PWM2EN: PWM2 使能位

- 1: 使能 PWM2 输出
- 0: 关闭 PWM2 输出

Bit1T1PWM1EN: PWM1 使能位

- 1: 使能 PWM1 输出
- 0: 关闭 PWM1 输出

Bit0T1PWM0EN: PWM0 使能位

- 1: 使能 PWM0 输出
- 0: 关闭 PWM0 输出

寄存器定义 11-7T1PWMICON Timer1 作为 PWM 中断使能和中断申请标志寄存器

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1PWMICON	T1PWM3IF	T1PWM2IF	T1PWM1IF	T1PWM0IF	T1PWM3IE	T1PWM2IE	T1PWM1IE	T1PWM0IE
9Eh	R/W							

Bit7T1PWM3IF: PWM3 中断标志位, 高有效。进入中断子程序, 需要软件清零该位。

Bit6T1PWM2IF: PWM2 中断标志位, 高有效。进入中断子程序, 需要软件清零该位。

Bit5T1PWM1IF: PWM1 中断标志位, 高有效。进入中断子程序, 需要软件清零该位。

Bit4T1PWM0IF: PWM0 中断标志位, 高有效。进入中断子程序, 需要软件清零该位。

Bit3T1PWM3IE: PWM3 中断使能位, 高有效。

1: 使能 PWM3 中断功能

0: 关闭 PWM3 中断功能

Bit2T1PWM2IE: PWM2 中断使能位, 高有效。

1: 使能 PWM2 中断功能

0: 关闭 PWM2 中断功能

Bit1T1PWM1IE: PWM1 中断使能位, 高有效。

1: 使能 PWM1 中断功能

0: 关闭 PWM1 中断功能

Bit0T1PWM0IE: PWM0 中断使能位, 高有效。

1: 使能 PWM0 中断功能

0: 关闭 PWM0 中断功能

寄存器 T1D0H(92h)&T1D0L(93h) 可读写, 是 PWM0 有效电平宽度寄存器

寄存器 T1D1H(94h)&T1D1L(95h) 可读写, 是 PWM1 有效电平宽度寄存器

寄存器 T1D2H(96h)&T1D2L(97h) 可读写, 是 PWM2 有效电平宽度寄存器

寄存器 T1D3H(9Ch)&T1D3L(9Dh) 可读写, 是 PWM3 有效电平宽度寄存器

操作步骤:

1) 设置 T1PSC[2:0], 软件配置 T1PWM 时钟。

2) 设置 T1OVR, 设置 T1PWM 周期。

3) 设置 T1Dx, 设置 T1PWMx 有效电平宽度。

4) 设置 T1PWMxS, 设置 T1PWMx 输出有效电平。

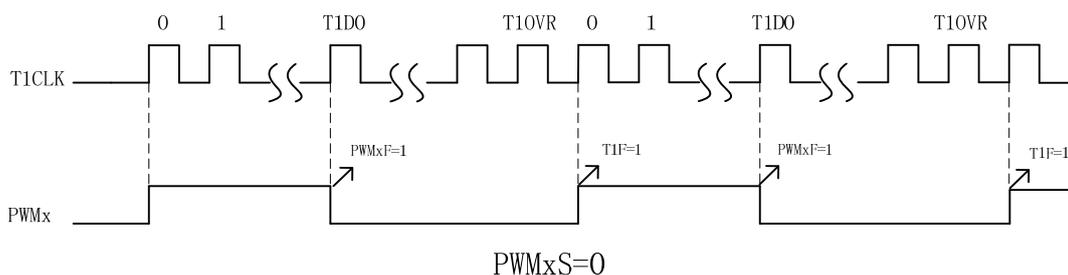
5) 设置寄存器标志位: T1PWMxEN, 使能 T1PWMx 输出。

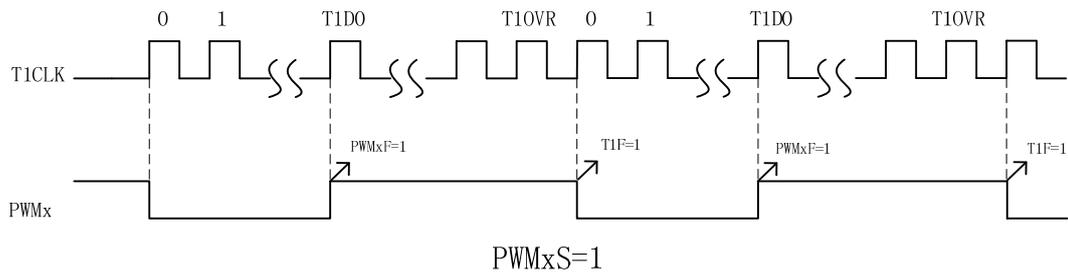
6) 设置寄存器标志位: T1EN, 使能定时器。

7) 当 T1PWMx 内部计数器计数等于 T1Dx 时, 占空比中断标志会置位, 如此时占空比中断使能打开, 程序计数器会跳转到 T1 PWM 中断入口地址 0x33H。进入中断后, 通过查询寄存器 T1PWMICON 的高 4bit 来判断具体是哪个 T1PWM 引起的中断。

8) 当 T1C 计数等于 T1OVR 时, 中断标志会置位。定时器 1 中断使能位如打开的话, 程序计数器会跳转到 1BH

周期为 T1OVR+1, T1PWM 输出高电平脉宽为 T1D0。T1PWM 输出波形输出如下:





12.Timer2 模块

Timer2 的时钟 TIMERCLK 可软件配置来自于 TCK，也可来自于 MCK (TCKEN=0 时)。如图 4-1 简化的 MCU 时钟源框所示。

Timer2 模块的时钟默认来源于独立的 TCK 时钟模块 (Timer Clock Module), 频率通过配置 TCKSEL 项来选择 48M (默认) 还是 32M。当将 TCKEN 设置为 0 后, 则 Timer2 的时钟便来自于 MCK, 同时芯片内部的 TCK 时钟被关掉, 降低功耗。

定时器模块前端集成了一个分频器, 分频的时钟 T2CLK 作为 16 bits 计数器的输入时钟。当用户设置了定时器模块的使能标志, 16 bits 计数器将启动, 将会从 0 递增到 T2OVR。用户需要设置 T2OVR (定时器模块中断信号选择器) 以选择定时超时中断信号。当定时超时发生时, 中断标志位会自设置, 程序计数器会跳转到 6Bh 以执行中断服务程序。内部定时计数器为 16 位宽。同样, T2 还集成了一些辅助模块, 通过灵活的软件配置, 除了作为普通的定时计数外, 可以用作捕捉功能 或者 T2PWM 功能。尤其是针对电机的 T2PWM 软件配置做了大量的优化。

12.1T2 定时计数器

图 12-1Timer2 模块框图

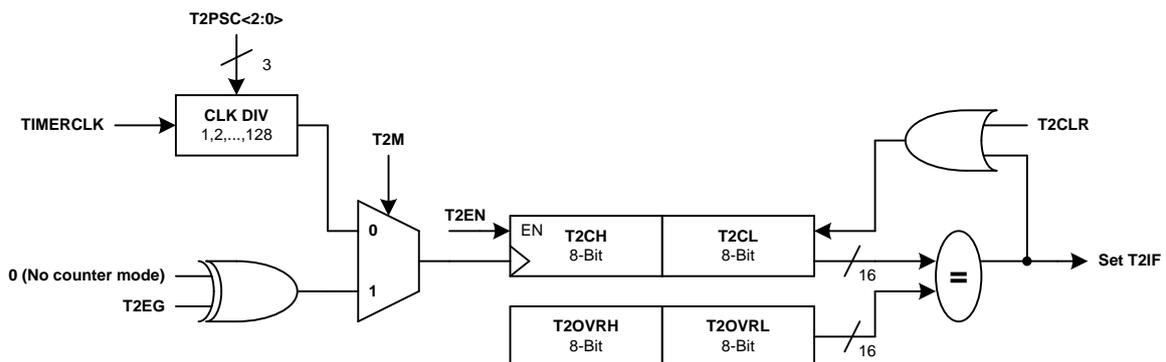


表 12-1 与 Timer1 相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
A8h	IEN0	EA	-	-	T2IE	T1IE	P1EA	T0IE	POEA
C8h	T2CON	T2CL	T2M	T2IF	T2EG	T2PSC[2:0]		T2EN	
CCh	T2CH	T2C[15:8]							
CBh	T2CL	T2C[7:0]							
CAh	T2OVRH	T2OVR[15:8]							
C9h	T2OVR	T2OVR[7:0]							

寄存器定义 12-1T2CON

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CON	T2CLR	T2M	T2IF	T2EG	T2PSC[2:0]		T2EN	
C8h	W	R/W	R/W	R/W	R/W		R/W	

Bit7T2CLR: 计数寄存器 T2CH&T2CL 清 0 使能位

- 1: 数寄存器 T2C 到达预设值时自动清 0。该位在写操作完成时由硬件自动清零。读永远为 0
- 0: 设置为手动对定时器 T2C 寄存器清 0

Bit6 T2M: 定时/计数器 2 模式选择位

- 1: 计数器模式
- 0: 定时器模式

Bit5 T2IF: T2 溢出标志位

- 1: 当定时计数溢出时, 此位会被硬件置 1, 需软件清 0。
- 0: T2 未溢出

Bit4 T2EG: 计数器的有效沿选择位

- 1: 下降沿计数
- 0: 上升沿计数

Bit3-1 T2PSC<2:0>: 定时/计数器时钟选择

- 000: TIMERCLK
- 001: TIMERCLK/2
- 010: TIMERCLK/4
- 011: TIMERCLK/8
- 100: TIMERCLK/16
- 101: TIMERCLK/32
- 110: TIMERCLK/64
- 111: TIMERCLK/128

Bit0T2EN: 定时/计数器 2 使能位

- 1: 使能定时/计数器 2
- 0: 禁止定时/计数器 2

寄存器 T2CH(CCh)&T2CH(CBh) 可读写, Timer 2 计数寄存器

寄存器 T2OVRH(CAh)&T2OVR (C9h) 可读写, Timer2 溢出寄存器

T2 定时计数器操作

定时器:

Timer 2 定时器操作步骤:

- 1) 设置 T2PSC, 选择定时器/计数器时钟。
- 2) 设置 T2M, 软件配置为定时器模式。

- 3) 设置 T2OVR，选择定时器溢出值。
- 4) 设置寄存器标志位：T2IE 与 EA，使能定时器中断。
- 5) 设置寄存器标志位：T2EN，使能定时器。
- 6) 当定时超时发生时，程序计数器会跳转 T2 中断入口地址 0x6BH。

定时器 2 溢出时间计算方法：

$$\text{定时器 2 溢出时间} = (\text{T2OVR}[15:0] + 1) / \text{T2CLK}$$

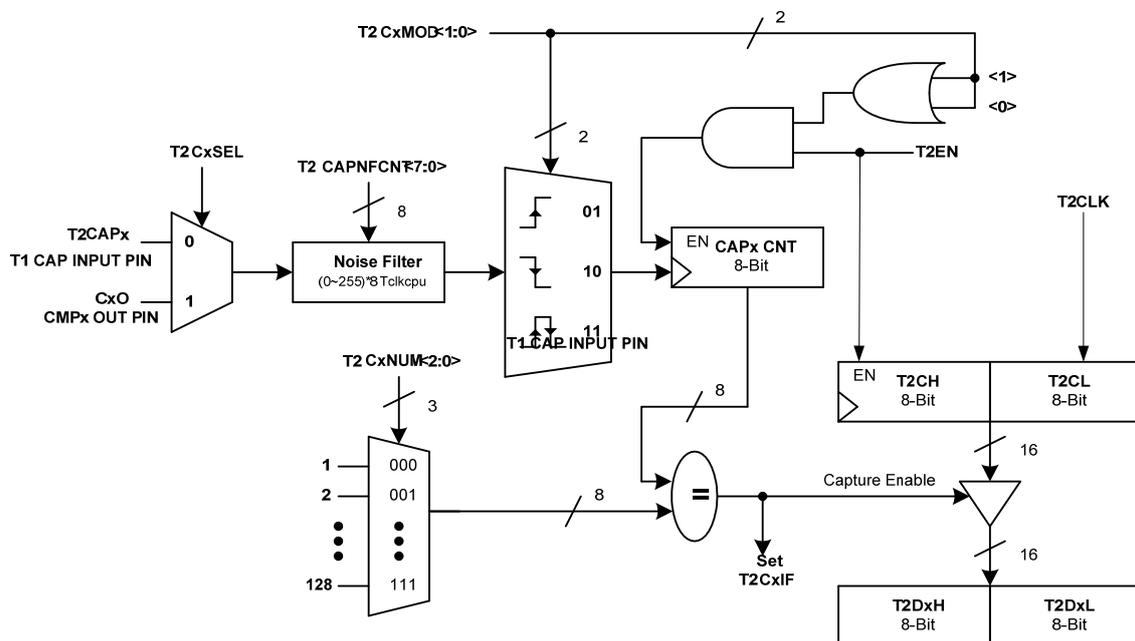
计数器

注意： T2 没有计数器功能。

12.2 T2 捕获功能(3 路)

捕获功能是指该芯片能对事件的持续时间计时。定时器 2 共有 3 路捕获，信号分别来自各捕捉输入引脚 T2CAP1 (P2.0【默认】/P1.5)、T2CAP2 (P2.0)、T2CAP3 (P2.2【默认】/P2.0) 输入) 或选择从比较器 (1/2/3) 内部输出。T2CAP1/2/3 四个通道捕获 Timer2 计数寄存器的值后，分别保存至 T2D1/T2D2/T2D3 寄存器。

图 12-2 捕获框图



注意：T2 捕捉功能与 T1 的捕捉原理完全一样，详情请阅读 T1 捕捉功能的说明方法。

注意： 棕色背景寄存器均位于 SFR 列表第 0 页，需要设置 PSW[5]=0 进行访问。

12.2.1 与捕获功能相关的寄存器

表 12-2 与捕获功能相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
A8h	IEN0	EA	-	-	T2IE	T1IE	P1EA	T0IE	P0EA
D5h	T2CAPCON	T2C3MOD<1:0>		T2C2MOD<1:0>		T2C1MOD<1:0>		-	
B1h	T2CAPNUM0	T2C1SEL	T2C1NUM<2:0>			-	-	-	-
B2h	T2CAPNUM1	T2C3SEL	T2C3NUM<2:0>			T2C2SEL	T2C2NUM<2:0>		
B3h	T2CAPNFCNT	T2CAPNFCNT<7:0>							
D7h	T2CAPINT	T2C3IF	T2C2IF	T2C1IF	-	T2C3IE	T2C2IE	T2C1IE	-

寄存器定义 12-2T2CAPCON

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CAPCON	T2C3MOD<1:0>		T2C2MOD<1:0>		T2C1MOD<1:0>		-	
D5h	R/W		R/W		R/W		-	

Bit7-6T2C3MOD<1:0>:CAP3 模式选择。

Bit5-4T2C2MOD<1:0>:CAP2 模式选择

Bit3-2T2C1MOD<1:0>:CAP1 模式选择

- 00: 使能 PWM 模式, 不使能捕获模式。
- 01: 使能捕获模式, 并且是上升沿捕获(关闭 PWM 功能)
- 10: 使能捕获模式, 并且是下降沿捕获(关闭 PWM 功能)
- 11: 使能捕获模式, 并且是双沿捕获(关闭 PWM 功能)

Bit1-0 未使用

寄存器定义 12-3T2CAPNUM0

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CAPNUM0	T2C1SEL	T2C1NUM<2:0>			-	-	-	-
B1h	R/W	R/W			-	-	-	-

Bit7T1C1SEL:T2 捕获 1 输入选择

- 1: CMP1 输出
- 0: I0 口输入。P2.0 或者 P1.5 输入的选择请见 26. 程序配置章节。(默认 P2.0 输入)

Bit6-4T1C1NUM<2:0>:T1 捕获 1 的沿(下降沿/上升沿)的捕获个数软件配置

- 000: 1
- 001: 2
- 010: 4
- 011: 8
- 100: 16
- 101: 32
- 110: 64
- 111: 128

Bit3-0 未使用

寄存器定义 12-4T2CAPNUM1

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CAPNUM1	T2C3SEL	T2C3NUM<2:0>			T2C2SEL	T2C2NUM<2:0>		
B2h	R/W	R/W			R/W	R/W		

Bit7T2C1SEL:T2 捕获 3 输入选择

- 1: CMP3 输出
- 0: I0 口输入。P2.2 或者 P2.0 输入的选择请见 26. 程序配置章节。(默认 P2.2 输入)

Bit6-4T2C1NUM<2:0>:T2 捕获 3 的沿(下降沿/上升沿)的捕获个数软件配置

000: 1
001: 2
010: 4
011: 8
100: 16
101: 32
110: 64
111: 128

Bit3T2COSEL:T2 捕获 2 输入选择

1: CMP2 输出
0: P21 输入

Bit2-0T2CONUM<2:0>:T2 捕获 2 的沿（下降沿/上升沿）的捕获个数软件配置

000: 1
001: 2
010: 4
011: 8
100: 16
101: 32
110: 64
111: 128

寄存器定义 12-5T2CAPINT

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CAPINT	T2C3IF	T2C2IF	T2C1IF	-	T2C3IE	T2C2IE	T2C1IE	-
D7h	R/W	R/W	R/W	-	R/W	R/W	R/W	-

Bit7T2C3IF:CAP3 中断标志位

Bit6T2C2IF:CAP2 中断标志位

Bit5T2C1IF:CAP1 中断标志位

Bit4 未使用

Bit3T2C3IE:CAP3 中断允许标志位

Bit2T2C2IE:CAP2 中断允许标志位

Bit1T2C1IE:CAP1 中断允许标志位

Bit0 未使用

寄存器 T2CAPNFCNT (B3h) 可读写, Noise filter cnt。总的滤波时间为 $(0 \sim ff) \times 8T_{clkcpu}$

12.2.2 软件配置方法（同 Timer1 捕获原理相同）

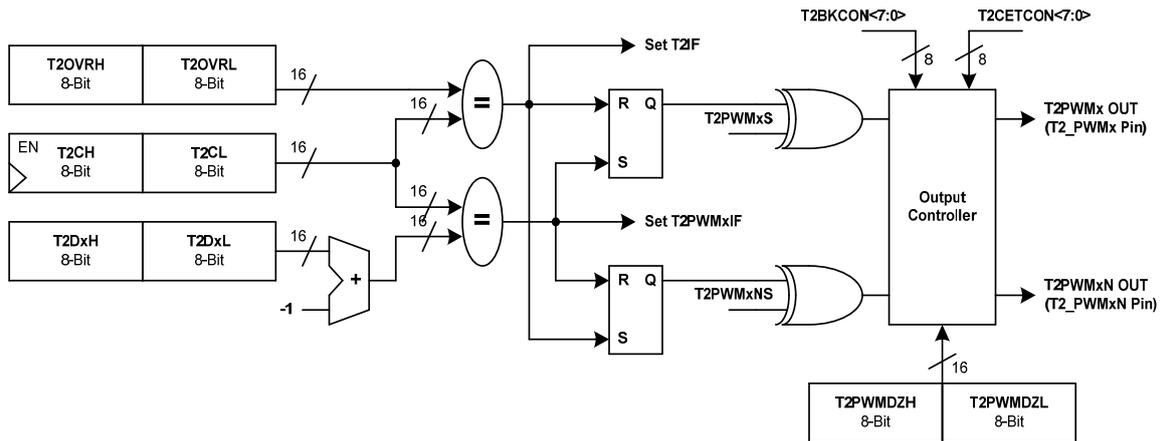
捕获 x (x=1~3) 的软件配置步骤:

1. 程序配置 T2 的时钟 TIMERCLK, 即 TCK (TCKEN=1) 或 MCK (TCKEN=0), TCK 可以为 48M 或者 32M; T2CAPx 输入信号源, 同时需程序配置捕捉输入通道
2. 设置 T2PSC<2:0>确定 T2 时钟分频;
3. 设置 T2M=0; 选择 T2 的时钟源
4. 设置 T2CxSEL 软件配置捕捉信号源。
5. 设置 T2CAPNFCNT 噪音滤波时间
6. 设置 T2CxMOD<1:0>触发的方式
7. 设置 T2CxNUM<2:0>一次捕捉触发信号的个数
8. 打开中断允许标志 T2CxIE=1 (如果总中断没有打开, 请首先打开 EA=1)
9. 打开开始计时 T2EN=1

12.3T2 PWM(3路/含刹车)

定时器 2 有 3 路 T2PWM 输出，即 T2PWM_x 引脚 (x=1/2/3)，对应的互补波形输出引脚为 T2PWM_{xN}，3 路 T2PWM 的周期大小通过软件配置 T2PSC[2:0]和 T2OVR[15:0]来实现，T2PWM_x 占空比由 T2D_x[15:0]来调整，PWM_xNEN 软件配置 T2PWM_x 互补输出，T2PWMDZ[15:0]软件配置 T2PWM_x 互补输出的死区大小。这 3 路 PWM 都有边沿对齐模式，中心对齐模式和互补输出。

图 12-3 Timer2 PWM 原理框图



注意：

对输出 PWM 控制的 T2PWMCON、T2PWMCON1、T2PWMCON2 三个只能写的寄存器操作，禁用与、或、非直接操作。如果非得需要与或非操作，请先自己定义相应的 RAM，再与自定义 RAM 进行与或非操作后，然后直接赋值给以上三个寄存器即可。

表 12-3 与 T2 PWM 功能相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C8h	T2CON	T2CLR	T2M	T2IF	T2EG	T2PSC[2:0]			T2EN
C9h	T2OVR	T2OVR[7:0]							
CAh	T2OVRH	T2OVR[15:8]							
CBh	T2CL	T2C[7:0]							
CCh	T2CH	T2C[15:8]							
A1h	T2PWMICON	T2PWM3IF	T2PWM2IF	T2PWM1IF	-	T2PWM3IE	T2PWM2IE	T2PWM1IE	-
A2h	T2BKCON	BKF	BKFLT<1:0>		BKSWT	BKHSEL	BKMODE	BKEN	CETEN
A3h	T2CETCON	CNTIS	-	-	OVRTADS	ZEROTADS	PWMNREN	PWMREN	CNT0IE
A4h	T2PWMCON1	T2PWM1NS	T2PWM1S	-	-	T2PWM1NEN	T2PWM1EN	-	-
A5h	T2PWMCON2	T2PWM3NS	T2PWM3S	T2PWM2NS	T2PWM2S	T2PWM3NEN	T2PWM3EN	T2PWM2NEN	T2PWM2EN
A6h	T2PWMDZH	T2PWMDZ [15:8]							
A7h	T2PWMDZL	T2PWMDZ [7:0]							
A8h	IEN0	EA	-	-	T2IE	T1IE	P1EA	T0IE	P0EA
AAh	T2D1H	T2D1[15:8]							
ABh	T2D1L	T2D1[7:0]							
ACh	T2D2H	T2D2[15:8]							
ADh	T2D2L	T2D2[7:0]							
A Eh	T2D3H	T2D3[15:8]							
AFh	T2D3L	T2D3[7:0]							

寄存器定义 12-6T2PWMCON1

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2PWMCON1	T2PWM1NS	T2PWM1S	-	-	T2PWM1NEN	T2PWM1EN	-	-
A4h	W	W	-	-	W	W	-	-

Bit7T2PWM1NS: PWM1 互补输出有效电平选择位

- 1: 低电平有效
- 0: 高电平有效

Bit6T2PWM1S: PWM1 输出有效电平选择位

- 1: 低电平有效
- 0: 高电平有效

Bit5-4 未使用

Bit3T2PWM1NEN: PWM1 互补输出使能位

- 1: 使能 PWM1 互补输出
- 0: 关闭 PWM1 互补输出

Bit2T2PWM1EN: PWM1 使能位

- 1: 使能 PWM1 输出
- 0: 关闭 PWM1 输出

Bit1-0 未使用

寄存器定义 12-8 T2PWMCON2

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2PWMCON2	T2PWM3NS	T2PWM3S	T2PWM2NS	T2PWM2S	T2PWM3NEN	T2PWM3EN	T2PWM2NEN	T2PWM2EN
A5h	W	W	W	W	W	W	W	W

Bit7T2PWM3NS: PWM3 互补输出有效电平选择位

- 1: 低电平有效
- 0: 高电平有效

Bit6T2PWM3S: PWM3 输出有效电平选择位

- 1: 低电平有效
- 0: 高电平有效

Bit5T2PWM2NS: PWM2 互补输出有效电平选择位

- 1: 低电平有效
- 0: 高电平有效

Bit4T2PWM2S: PWM2 输出有效电平选择位

- 1: 低电平有效
- 0: 高电平有效

Bit3T2PWM3NEN: PWM3 互补输出使能位

- 1: 使能 PWM3 互补输出
- 0: 关闭 PWM3 互补输出

Bit2T2PWM3EN: PWM3 使能位

- 1: 使能 PWM3 输出
- 0: 关闭 PWM3 输出

Bit1T2PWM2NEN: PWM2 互补输出使能位

- 1: 使能 PWM2 互补输出
- 0: 关闭 PWM2 互补输出

Bit0 T2PWM2EN: PWM2 使能位

- 1: 使能 PWM2 输出
- 0: 关闭 PWM2 输出

寄存器定义 12-7T2PWMICON

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2PWMICON	T2PWM3IF	T2PWM2IF	T2PWM1IF	-	T2PWM3IE	T2PWM2IE	T2PWM1IE	-
A1h	R/W	R/W	R/W	-	R/W	R/W	R/W	-

Bit7T2PWM3IF: PWM3 中断标志位, 高有效。进入中断子程序, 需要软件清零该位。

Bit6T2PWM2IF: PWM2 中断标志位, 高有效。进入中断子程序, 需要软件清零该位。

Bit5T2PWM1IF: PWM1 中断标志位, 高有效。进入中断子程序, 需要软件清零该位。

Bit4 未使用

Bit3T2PWM3IE: PWM3 中断使能位, 1: 开启 PWM3 中断功能 0: 关闭 PWM3 中断功能

Bit2T2PWM2IE: PWM2 中断使能位, 1: 开启 PWM3 中断功能 0: 关闭 PWM3 中断功能

Bit1T2PWM1IE: PWM1 中断使能位, 1: 开启 PWM3 中断功能 0: 关闭 PWM3 中断功能

Bit0 未使用

寄存器定义 12-9T2CETCON (此寄存器是工作在电机驱动模式下, PWM 软件配置为中心对齐, 计数器向上计数到 T2OVR 溢出或者向下计数到 0 时, 硬件开启 ADC 和 PWM 后期输出与否的控制寄存器。用途: 减少在 PWM 输出最大值或者最小值时, 在中断中启动 ADC 和控制后续 PWM 输出与否的软件开销)

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CETCON	CNTIS	-	-	OVRTADS	ZEROTADS	PWMNREN	PWMREN	CNT0IE
A3h	R	-	-	R/W	R/W	R/W	R/W	R/W

Bit7CNTIS: 当使用中心对齐模式时, 此位表示 cnt 是计数到 OVR 或者是倒数计数到 0 时产生的中断判断。

1: 计数到 OVR 时产生的中断

0: 计数到 0 时产生的中断。

Bit6-5 未使用

Bit4 OVRTADS: 置 1 时, 该位有效, 当 CNT 计数到 OVR 时, 硬件启动 AD 模块, 即硬件自动置位 ADS。

1: 当计数到 OVR 时, 硬件置位 ADS, 开启 ADC 转换(ADC 需先前软件配置好 ADC 的通道、频率、AD 参考电压等等参数)。

0: 当计数到 OVR 时, 硬件不可置位 ADS。

Bit3ZEROTADS: 置 1 时, 该位有效, 当 CNT 计数到 0 时, 硬件启动 AD 模块, 即硬件自动置位 ADS。

1: 当计数到 0 时, 硬件置位 ADS, 开启 ADC 转换(ADC 需先前软件配置好 ADC 的通道、频率、AD 参考电压等等参数)。

0: 当计数到 0 时, 硬件不可置位 ADS。

Bit2PWMNREN: 在中心对齐模式下, 当 $T2Dx \leq T2PWMxDZ$ 时, 补波 pwmn(3 路 PWMN 波统一)

是否还要继续产生 pwmn 有效波形。

0: 继续产生 pwmn 有效波形

1: 不产生 pwmn 有效波形, 此时 pwmn 一直输出无效波形

Bit1PWMREN: 在中心对齐模式下, 当 $T2Dx + T2PWMxDZ > T2OVR$ 时, 主波 pwm(3 路 PWM 波统一)

是否还要继续产生 pwm 有效波形。

0: 继续产生 pwm 有效波形

1: 不产生 pwm 有效波形, 此时 pwm 一直输出无效波形

Bit0CNT0IE: 当使用中心对齐模式时, cnt 倒数计数到 0 时, 是否使能中断。

此中断与 timer 溢出中断共用同一个中断使能(ET2)

1: 使能

0: 不使能

寄存器 T2PWMDZH(A6h)&T2PWMDZL(A7h) 可读写, PWM1/2/3 互补输出死区寄存器

寄存器 T2D1H(AAh)&T2D1L(ABh) 可读写, PWM1 有效电平宽度寄存器

寄存器 T2D2H(ACH)&T2D2L(ADh) 可读写, PWM2 有效电平宽度寄存器

寄存器 T2D3H(AEh)&T2D3L(AFh) 可读写, PWM3 有效电平宽度寄存器

12.3.1 边沿对齐模式输出

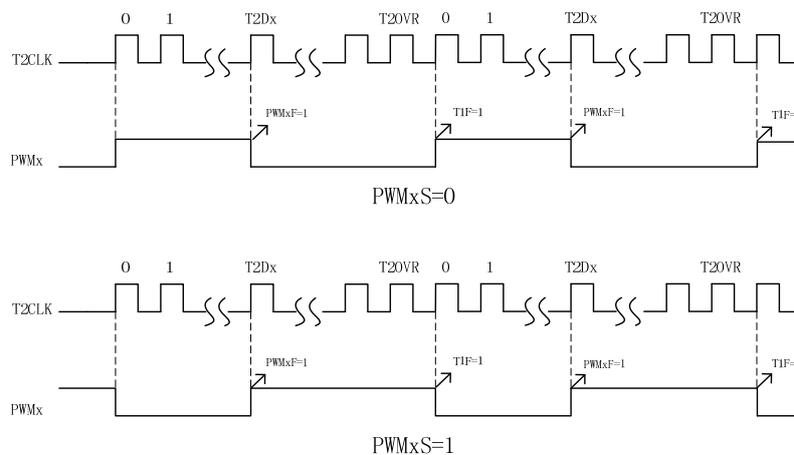
T2PWM 的周期 = (T20VR[15:0]+1) / T2CLK.

T2PWMx 的有效电平宽度比 = T2Dx[15:0] / (T20VR[15:0]+1)

边沿对齐模式操作:

- 1) 设置 T2PSC[2:0], 软件配置 T2PWM 时钟。
- 2) 设置 T20VR, 设置 T2PWM 周期。
- 3) 设置 T2Dx, 设置 T2PWMx 有效电平宽度。
- 4) 设置 T2PWMxS, 设置 T2PWMx 输出有效电平。
- 5) 设置寄存器标志位: T2PWMxEN, 使能 T2PWMx 输出。
- 6) 设置寄存器标志位: T2EN, 使能定时器。
- 7) 当 PWMx 内部计数器计数等于 T2Dx 时, 占空比中断标志会置位, 如此时占空比中断使能打开, 程序计数器会跳转到 002BH。进入中断后, 通过查询寄存器 T2PWMICON 的高 3bit 来判断具体是哪个 T2PWM 引起的中断。
- 8) 当 T2C 计数等于 T20VR 时, 中断标志会置位。定时器 1 中断使能位如打开的话, 程序计数器会跳转到 006BH。

周期为 T20VR+1, T2PWMx 高电平脉宽为 T2Dx。T2PWMx 波形输出如下:



12.3.2 PWM 互补输出与死区

T2PWM 的周期 = (T20VR[15:0]+1) / T2CLK.

T2PWMx 的有效电平宽度比 = T2Dx[15:0] / (T20VR[15:0]+1)

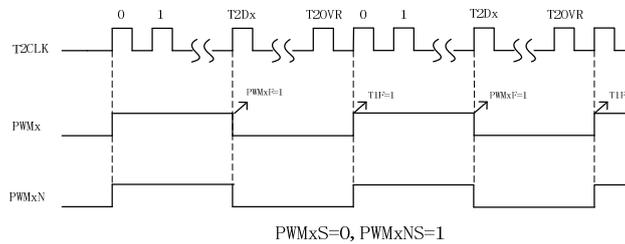
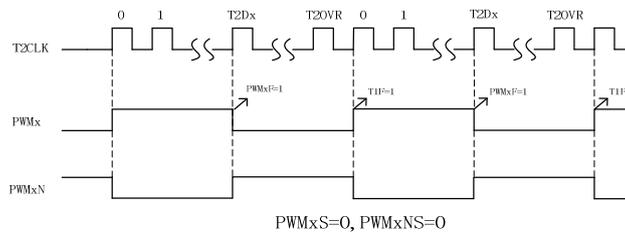
互补输出操作:

- 1) 设置 T2PSC[2:0], 软件配置 T2PWM 时钟。
- 2) 设置 T20VR, 设置 T2PWM 周期。
- 3) 设置 T2Dx, 设置 T2PWMx 有效电平宽度。
- 4) 设置 T2PWMxDZ[15:0], 设置 T2PWMx 死区。
- 5) 设置 T2PWMxS 和 T2PWMxNS, 设置 T2PWMx 和 T2PWMx 互补输出有效电平。
- 6) 设置寄存器标志位: T2PWMxEN 和 T2PWMxNEN, 使能 T2PWMx 和 T2PWMx 互补输出。
- 7) 设置寄存器标志位: T2EN, 使能定时器。

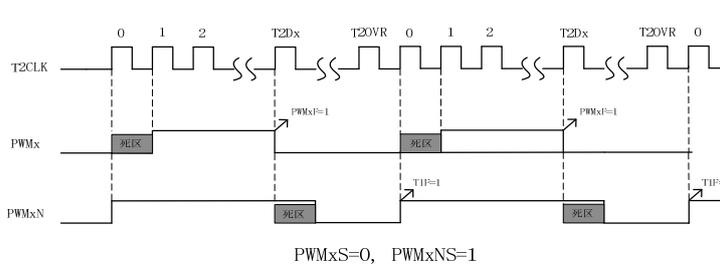
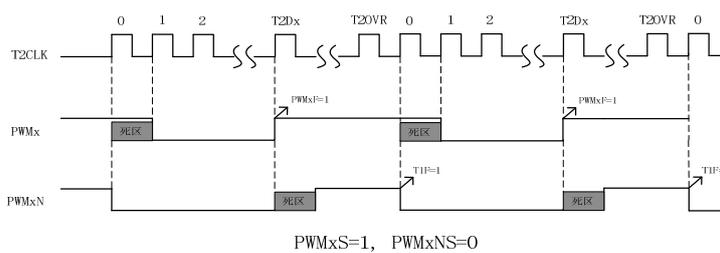
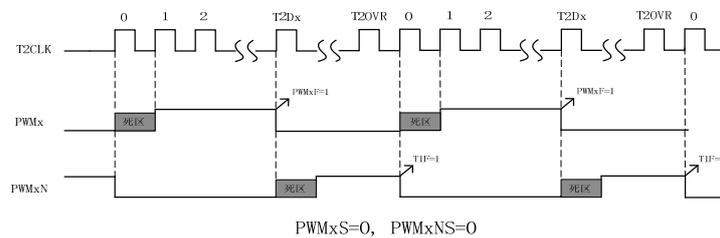
8) 当 PWM 内部计数器计数等于 T2Dx 时, 占空比中断标志会置位, 如此时占空比中断使能打开, 程序计数器会跳转到 2BH。进入中断后, 通过查询寄存器 T2PWICON 的高 3bit 来判断具体是哪个 T2PWM 引起的中断。

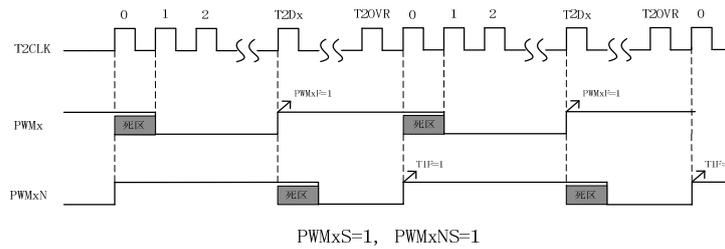
9) 当 T2C 计数等于 T20VR 时, 中断标志会置位。定时器 1 中断使能位如打开的话, 程序计数器会跳转到 6BH。

周期为 T20VR+1, T2PWMx 高电平脉宽为 T2Dx。T2PWMxDZ=0, 即死区为 0。T2PWMx 波形输出如下



周期为 T20VR+1, T2PWMx 高电平脉宽为 T2Dx, 死区为 T2PWMxDZ=1。T2PWMx 波形输出如下:





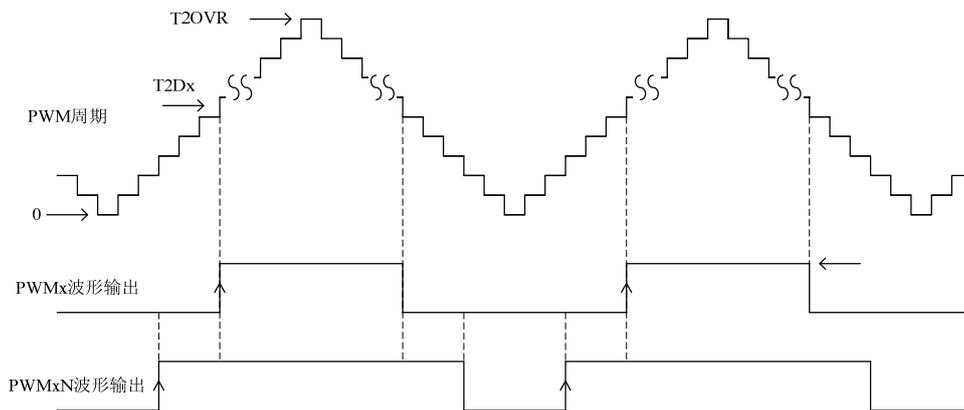
12.3.3 PWM 中心对齐

T2 的 3 路 T2PWM 都支持中心对齐模式，当置位 T2BKCON [0] 的 CETEN 为 1 时，便使能了中心对齐功能。此时，3 路 T2PWM 的输出，包括互补输出都已经是中心对齐方式。

中心对齐模式下，T2PWM 的周期为 $2 * T2OVR$ 。

$$T2PWMx \text{ 的有效电平宽度比} = (2 * (T2OVR - T2Dx) - T2PWMxDZ) / (2 * T2OVR)$$

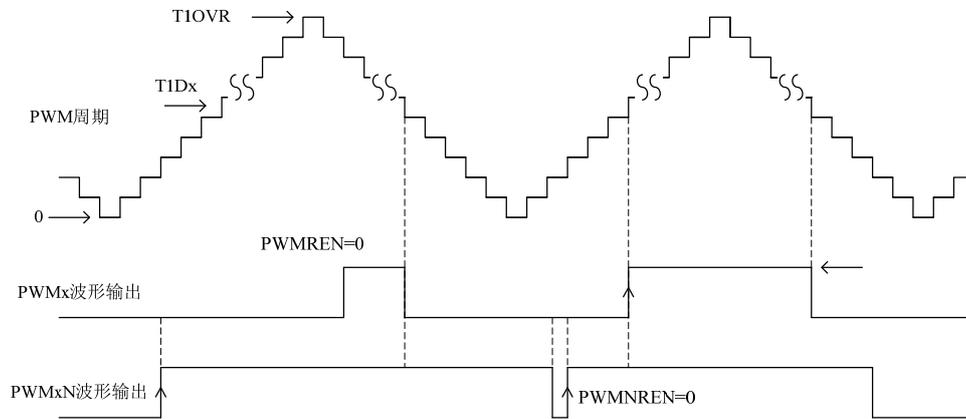
$$T2PWMxN \text{ 的有效电平宽度比} = (2 * T2Dx - T2PWMxDZ) / (2 * T2OVR)$$



PWM 波形特殊情况说明 (仅适用 T2PWM 中心对齐模式):

1. T2PWM 在使用过程中，如果将 T2Dx 设为零，则 T2PWM 脚一直输出有效电平。如有使用互补功能，则 T2PWMxN 脚一直输出无效电平。需要注意，此时 T2PWM 的死区时间无效。此时，如果倒数计数到 0 值的中断使能 (CNT0IE) 开的话，则当计数倒数计数到 0 时，可产生 T2PWM 中断。
2. 当 $T2Dx \geq T2OVR$ 时，T2PWMx 脚一直输出无效电平，而 T2PWMxN 脚一直输出有效电平。需要注意，此时 T2PWM 的死区时间无效。此时，如果倒数计数到 0 值的中断使能 (CNT0IE) 开的话，则当计数倒数计数到 0 时，可产生 T2PWM 中断。
3. 当 $T2PWMxDZ \geq 2 * (T2OVR - T2Dx)$ 时，此时 T2PWMx 一直输出无效电平。
4. 当 $T2Dx + T2PWMxDZ \geq T2OVR$ ，且 $T2PWMxDZ < 2 * (T2OVR - T2Dx)$ 时。此时 T2PWMx 可根据软件配置位 PWMREN 来决定是否输出有效波。如下图波形所示，当 PWMREN=0 时，可输出 T2PWM 的有效波形；当 PWMREN=1 时，则 T2PWM 一直输出无效值。
5. 当 $T2PWMxDZ \geq 2 * T2Dx$ 时，此时 T2PWMxN 一直输出无效电平。
6. 当 T2PWMx 和 T2PWMxN 都输出无效值时，T2PWM 的 duty 中断才不会产生。否则都可以产生 T2PWM 的 duty 中断。

7. 当 $T2PWMxDZ \geq T2Dx$ ，且 $T2PWMxDZ < 2 * T2Dx$ 时，此时 $T2PWMxN$ 可根据软件配置位 $PWMNREN$ 来决定是否输出有效波。如下图波形所示，当 $PWMNREN=0$ 时，可输出 $T2PWMxN$ 的有效波形；当 $PWMNREN=1$ 时，则 $T2PWMxN$ 一直输出无效值。



12.3.4 PWM 刹车

定时器 2 的 3 路 T2PWM 波形只有在中心对齐模式才具备刹车功能。即当刹车触发有效时，定时器 2 的所有 3 路 T2PWM 波形（包括主波和补波）立刻停止输出，并回到初始电平，同时，原本复用为 T2PWM 功能的芯片管脚，恢复到 GPIO 功能。客户在使用 T2PWM 前，可事先考虑软件配置使用到 GPIO 管脚为输出，并且初始电平为无效电平。

通过将寄存器 $T2BKCON[1]$ 的 $BKEN$ 使能刹车模式。刹车触发源由寄存器 $T2BKCON[2]$ 的 $BKMODE$ 软件配置，0 为软件触发 ($BKSWT$) 刹车，1 为硬件触发刹车（P12 或者 P23 口输入触发电平，具体是哪个管脚通过 IDE 或者烧录软件进行程序配置，程序配置详解请查阅 26. 程序配置章节中 $T2BKIN$ 选项）。

寄存器定义 12-10T2BKCON

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2BKCON	BKF	BKFLT<1:0>	BKSWT	BKHWSEL	BKMODE	BKEN	CETEN	
A2h	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7BKF: 刹车完成标志。为 1 时，表示刹车完成。此位由硬件置位，软件清零

Bit6-5BKFLT<1:0>: 硬件 (P12 或者 P23) 刹车滤波时间

- 00: $64 * \text{cpuclk}$
- 01: $128 * \text{cpuclk}$
- 10: $256 * \text{cpuclk}$
- 11: $512 * \text{cpuclk}$

Bit4BKSWT: T2 的 3 路 PWM 软件触发信号。当软件置 1 时，便可刹车。此位又软件置 1，硬件清零。

Bit3BKHWSEL: P2.3 或者 P1.2 平时刹车选择，具体的管脚输入选择请见 26. 程序配置章节。（默认 P2.3 输入）

- 1: P2.3 或者 P1.2 为高电平时刹车。
- 0: P2.3 或者 P1.2 为低电平时刹车。

Bit2BKMODE: T2 的 3 路 PWM 刹车模式选择

- 1: 选择硬件 (P2.3 或者 P1.2) 刹车。
- 0: 使用软件 ($BKSWT$) 刹车。

Bit1BKEN: T2 的 3 路 PWM 是否刹车

- 1: 使能刹车功能
- 0: 不使能刹车功能。

Bit0CETEN: T2 的 3 路 PWM 是否中心对齐

- 1: 为中心对齐
- 0: 非中心对齐模式。

1. 软件刹车

当需要刹车时，软件置位 BKSWT 即可。当软件刹车触发后，硬件会自动清 0。刹车完成后，寄存器 T2BKCON [7] 的 BKF 标志位将置 1，此位需要由软件清 0。

2. 硬件刹车

通过寄存器 T2BKCON[3] 的 BKHSEL 为 1 来选择软件选择刹车信号来源于外部信号。外部刹车型号由 P12 或者 P23 输入，请见 26. 程序配置章节 T1BKIN 选项。当硬件刹车触发的输入电平一直有效时，标志位 BKF 也将一直为高。需要撤销硬件刹车 (P1.2 或者 P2.3 有效电平) 后，才能清除该标志位。

刹车完成后，标志位 (BKF) 置 1，此位由硬件置位，软件清零。

当刹车后，用户需要重新输出 T2PWM 波形时，则需要按如下步骤开启：

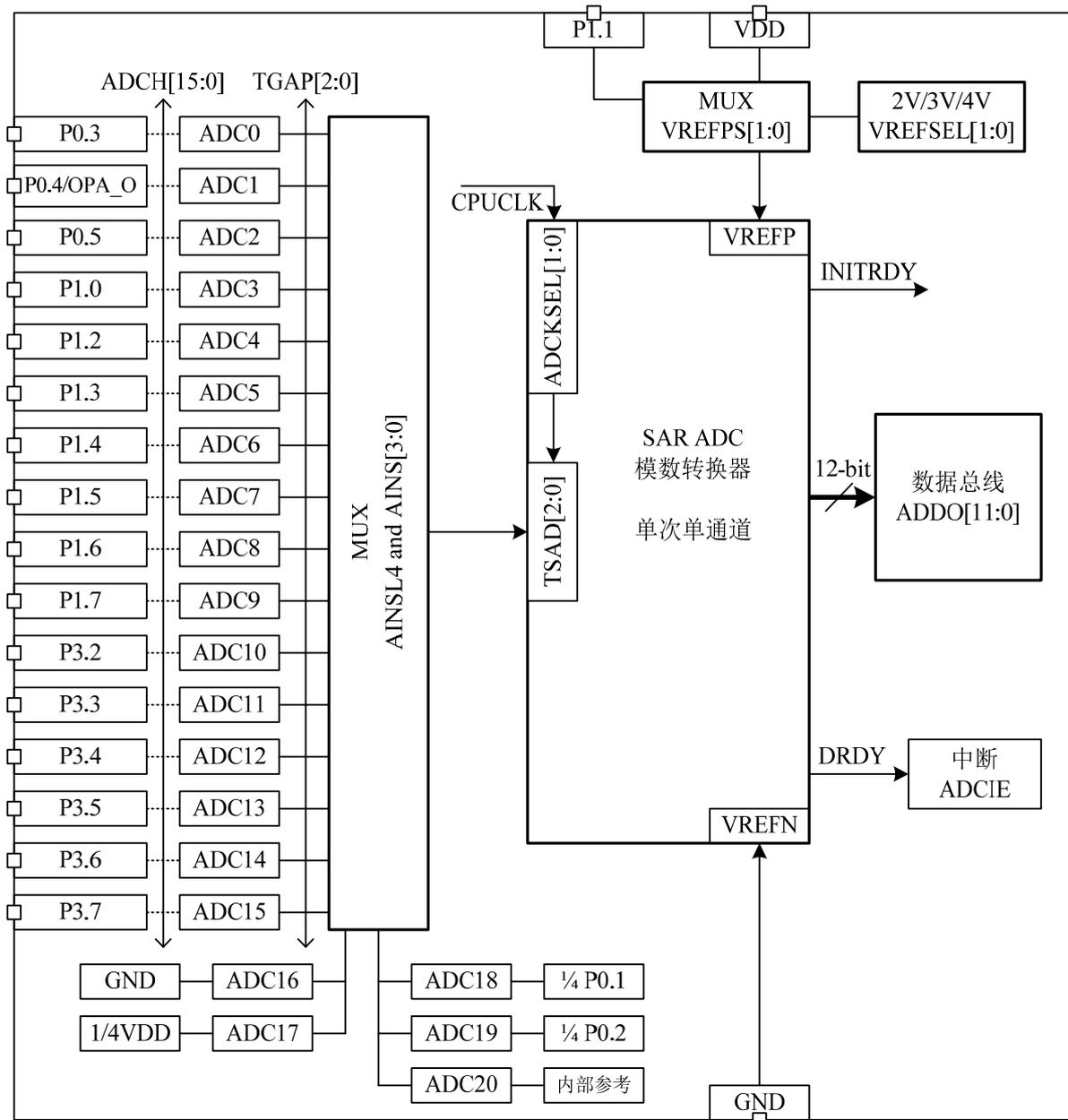
1. 清零刹车标志位 BKF。
2. 重新置位 T2PWM 的使能位，即 T2PWMxEN 和 T2PWWxEN

13. 数模转换(ADC)模块

BJ8M302A 集成了 16 个外部输入通道 (ADC0~ADC15) 和 3 个内部特殊通道 (ADC20:内部参考电压; ADC17:内部 VDD/4; ADC16:内部 GND) 12Bits 分辨率的逐次逼近式 (SAR) 模数转换器 (Analog-to-Digital Converter, ADC)。

13.1 ADC 转换模块框图

图 13-1 转换模块框图



注：当 ADEN=0 时，关闭 ADC 模块功能

ADC 能够灵活的软件配置参考电压：VDD、外部参考电压输入、内部基准电压。能够软件配置转换时钟频率和输入信号采样时间，以适应不同的功耗要求和传感器源阻抗要求。

图 13-2ADC 输入阻抗等效图

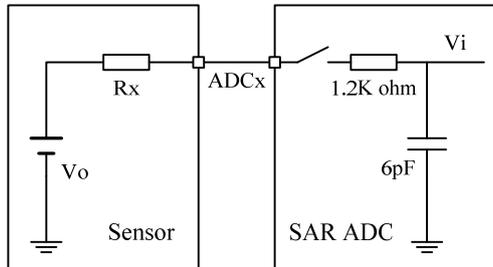


表 13-1 与 ADC 转换相关的寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
F1h	INFCON	WDTRF	CMP3IF	CMP2IF	CMP2IF	CMP0IF	ADCIF	-	DPS
A8h	IEN0	EA	-	-	T2IE	T1IE	P1EA	T0IE	POEA
B9h	IEN2	-	LVDIE	CMP3IE	CMP2IE	CMP1IE	ADCIE	-	WDTIE
DEh	ADDOH	-	-	-	-	ADDO[11:8]			
DDh	ADDOL	ADDO[7:0]							
DCh	ADCHH	ADCH[15:8]							
DBh	ADCHL	ADCH[7:0]							
DAh	ADCON0	ADRDY	AINSL4	VREFSEL[1:0]		AINS[3:0]			
D9h	ADCON1	-	-	-	TSAD[2:0]			ADCKSEL[1:0]	
D8h	ADCON2	ADEN	DRDY	ADS				VREFPS[1:0]	
DFh	ADDO	ADDO[11:4]							
EFh	METCH	-	-	VTHS	-	ADCVOF<3:0>			

寄存器定义 13-1 ADCON0

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	ADRDY	AINSL4	VREFSEL[1:0]		AINS[3:0]			
DAh	R	R/W	R/W		R/W			

Bit7ADRDY:ADEN 使能后，等待模拟稳定标志位，高电平有效

Bit5 VREFSEL[1:0]: ADC 内部参考电压选择

00: 无效

01: 2V

10: 3V

11: 4V

Bit6**Bit3-0**:AINSL4:AINS[3:0] :ADC 采样通道选择 (AINSL4:AINS[3:0]配合构成 5bit)

AINSL4:AINS[3:0] :DC 采样通道

0000: ADC0

0001: ADC1

00010: ADC2

00011: ADC3

00100: ADC4

00101: ADC5

- 00110: ADC6
- 00111: ADC7
- 01000: ADC8
- 01001: ADC9
- 01010: ADC10
- 01011: ADC11
- 01100: ADC12
- 01101: ADC13
- 01110: ADC14
- 01111: ADC15
- 10000: GND
- 10001: VDD/4
- 10010: P01 IO 输入/4
- 10011: P02 IO 输入/4
- 10100~11111:内部参考电压

寄存器定义 13-2 ADCON1

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1				TSAD[2:0]			ADCKSEL[1:0]	
D9h	R/W	R/W	R/W	R/W			R/W	

Bit7-5 保留位

Bit4-2TSAD[2:0] : ADC 输入信号采样建立时间 TS 选择 (Tadck 为工作时钟周期)

- 000: TS=2*TADCK
- 001: TS=4*TADCK (默认)
- 010: TS=6*TADCK
- 011: TS=8*TADCK
- 100: TS=10*TADCK
- 101: TS=12*TADCK
- 110: TS=14*TADCK
- 111: TS=16*TADCK

Bit1-0ADCKSEL[1:0]:ADC 工作时钟频率选择

- 00: FADCK=CPUCLK/1, 如 16M/1 (默认)
- 01: FADCK=CPUCLK/4, 如 16M/4
- 10: FADCK=CPUCLK/16, 如 16M/16
- 11: FADCK=CPUCLK/64, 如 16M/64

寄存器定义 13-3 ADCON2

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON2	ADEN	DRDY	ADS	-	-	-	VREFPS[1:0]	
D8h	R/W	R/W	R/W	-	-	-	R/W	

Bit7 ADCEN:ADC 模块使能

- 1: 打开 ADC 模块
- 0: 关闭 ADC 模块

Bit6 DRDY:ADC 序列转换完成标志位

- 1: 表示完成一个序列的 ADC 转换
- 0: 无序列转换完成标志, 由软件清 0; 每次重新启动 ADS 时硬件会清 DRDY

Bit5 ADS:ADC 序列转换启动位

- 1: 置 1 开始序列转换, 在转换过程中保持为 1, 转换完成后硬件自动清 0 (若转换期间被清 0 会立即终止序列转换)
- 0: 未进行序列转换/转换已完成

Bit4-2 未使用

Bit1-0 VREFPS<1:0>:ADC 参考电压 VREFP 选择

- 00: 选择 VDD 为参考电压 (默认)

- 01: 选择外部 VREF 端口输入作为参考电压
- 10: 选择内部参考电压
- 11: 选择内部参考电压，且把参考电压引到 P1.1 口输出。

寄存器定义 13-4METCH

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
METCH	-	-	VTHS	-	ADCVOF<3:0>			
EFh	-	-	R/W	-	R/W			

Bit3~0 ADCVOF: SAR ADC 的 offset 校准控制位

Bit3控制ADC 的offset 校准的方向。

Bit3=0 时，表示把ADC 输出结果往数值更小的方向调整；

Bit3=1 时，表示把ADC 输出结果往数值更大的方向调整。

如：

0000: 表示不作调整；

0001~0111: 表示把ADC 输出结果往数值更小的方向调整；数值越大则调整的幅度越大；

1000: 表示不作调整；

1001~1111: 表示把ADC 输出结果往数值更的方向调整；数值越大则调整的幅度越大。

寄存器 ADDOH(DEh) & ADDOL(DDh) 可读写，转换的 AD 数据

寄存器 ADCHH(DCh) 可读写，ADC 输入 IO 的 ADC 功能使能，高有效，对应 ADC_x (x=8~15)

寄存器 ADCHL(DBh) 可读写，ADC 输入 IO 的 ADC 功能使能，高有效，对应 ADC_x (x=0~7)

寄存器 ADDO(DFh) 可读写，ADC 转换的高 8 位 AD 数据

13.2 ADC 转换步骤

- (1) 配置 ADC 时钟 (ADCKSEL) 和采样建立时间 (TSAD)。
- (2) 使能 ADC 通道 ADCH[15:0] 和选择 ADC 通道 AINS[4:0]，内部通道默认为使能
- (3) 配置 ADC 内部参考电压 VREFSEL[1:0]，如选择外部参考，可省略。
- (4) 使能 ADC (ADEN=1)，如需中断则 EAD=1。
- (5) 等待 ADRDY=1，即等待 AD 初始化稳定。
- (6) 启动 ADC 转换 (ADS=1)，判断 DRDY 是否置 1，置 1 则可以读取 ADC 数据。如果 AD 中断打开则不用管 DRDY，直接进入中断读取 ADC 数据。中断入口地址为 93H。
- (7) 清 DRDY，完成一次 ADC 序列转换。如果开了 AD 中断，硬件会自动清 DRDY。

例 13-1 A/D 转换 (内部 VDD 参考)

```

/*****
* Function   : ADC_HaveAdValue
* Description: 查询法获取某一个 AD 通道的 AD 值
* Input      :
* Output     : ad_value
*****/
u16 ADC_HaveAdValue(void)
{
    u16 ad_value;
    u8 AD_H, AD_L;
    ad_value = 0;
    AD_H = 0;

```

```
AD_L = 0;
ADCHH = 0x00;
ADCHL = 0x01;           //ADCO
ADCON0 = 0x00;
ADCON1 = 0x04;         //TS=4*TADCK
ADCON2 &= 0xFC;       //设置 VDD 为参考
ADEN = 1;              //使能 ADC 总控制位。必须在设置 AD 通道前进行使能
while(!(ADCON0 | ADRDY)); //ADEN 使能后，等待模拟稳定标志位，高电平有效
    ADS = 1; //ADC 序列转换启动
while(!(DRDY)); //等待转换数据完成
    AD_L = ADDOL; //读取 ADC 数据，
    AD_H = ADDOH & 0x0F;
    ad_value = AD_H << 8;
    ad_value += AD_L;
    ADC_CLR_DRDY;
return ad_value;
}
```

14. 串行外设(SPI)接口

BJ8M302A 的串行外设(Serial Peripheral Interface, SPI)接口的特性:

- 支持 SPI 接口, 包括 master 模式和 slave 模式。
- 全双工, 三线同步传输
- 主模式或从模式
- 7 个可编程主时钟频率
- 极性和相位可编程的串行时钟
- 带 MCU 中断的主模式错误标志
- 写入冲突标志保护
- MSB 优先传输

注意: 黄色背景寄存器位于 SFR 列表第 1 页, 需要软件配置 PSW[5]=1 进行访问。

表 14-1 与 SPI 通讯相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
B1h	SPI0STA	SPIF	WCOL	SSERR	MODF	-	-	-	-
B2h	SPI0CON	SPR2	SPIEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0
B3h	SPI0DAT	SPIDAT[7:0]							
B8h	IEN1	-	-	S0RXIE	S0TXIE	SPIIE	P3EA	P2EA	I2CIE

寄存器定义 14-1 SPI0CON 串口通讯控制字

符号	Bit7	Bit6	Bit6	Bit4	Bit3	Bit2	Bit1	Bit0
SPI0CON	SPR2	SPIEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0
B2h	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit6 SPIEN: SPI 使能位 1: 开启 SPI 功能模块; 0: 禁止 SPI 功能模块

Bit5 SSDIS: NSS 引脚控制位 (从模式下, 不能关闭 NSS 引脚)

0: 在主和从模式下, 打开 NSS 引脚

1: 在主和从模式下, 关闭 NSS 引脚 (如果 SSDIS 置 1, 不产生 MODF 中断请求)

Bit4 MSTR: SPI 模式选择位 1: 主模式; 0: 从模式

Bit3 CPOL: 时钟极性控制位

1: 在空闲状态下 SCK 处于高电平

0: 在空闲状态下 SCK 处于低电平

Bit2 CPHA: SPI 时钟相位控制

1: 在 SCK 偶数边沿采样数据

0: 在 SCK 奇数边沿采样数据

Bit7 Bit1-0: (注意: BIT7、BIT1 和 BIT0 组合成三个比特位) SPI 时钟速率选择位

000: CPUCLK/2

001: CPUCLK/4

010: CPUCLK/8

011: CPUCLK/16

100: CPUCLK/32

101: CPUCLK/64

110: CPUCLK/128

111: 无时钟产生

寄存器定义 14-2 SPI0STA 控制字

符号	Bit7	Bit6	Bit6	Bit4	Bit3	Bit2	Bit1	Bit0
SPI0STA	SPIF	WCOL	SSERR	MODF	-	-	-	-
B1h	R/W	R/W	R/W	R/W	-	-	-	-

Bit7 SPIIF: SPI 数据传输标志位

1: 已完成数据传输, 由硬件置 1

0: 传输过程中由硬件清除。或者当 spif 置位后读取 spsta 后再读 spdat 清除

Bit6 WCOL: 写入冲突标志位

1: 检测到一个冲突, 由硬件置 1

0: 已处理写入冲突, 由软件清 0

Bit5SSERR: 同步串行从属错误标志, 当 NSS 输入低电平在接收序列结束前被释放, 将由硬件置 1。通过将寄存器 SPCON[6] 的 SPEN 位写 0 来清除标志

Bit4MODF: 模式错误标志位

1: 表明 NSS 引脚电平与 SPI 模式不一致, 由硬件置 1

0: 未发生模式错误, 由软件清 0

Bit3-0 未使用

寄存器 **SPI0DAT** (B3h) 可读写, 为 SPI 通讯收发缓存寄存器

14.1 信号描述

主输出从输入 (MOSI)

该信号连接主设备和从设备。数据通过 MOSI 从主设备串行传送到从设备, 主设备输出, 从设备输入。

主输入从输出 (MISO)

该信号连接从设备和主设备。数据通过 MISO 从从设备串行传送到主设备, 从设备输出, 主设备输入。当 SPI 软件配置为从设备并未被选中 (NSS 引脚为高电平), 从设备的 MISO 引脚处于高阻状态。

SPI 串行时钟 (SCK)

SCK 信号用作控制 MOSI 和 MISO 线上输入输出数据的同步移动。每 8 个时钟周期线上传送一个字节。如果从设备未被选中 (NSS 引脚为高电平), SCK 信号被此从设备忽略。

从设备选择引脚 (NSS)

每个从属外围设备由一个从选择引脚 (NSS 引脚) 选择, 当引脚信号为低电平时, 表明该从设备被选中。主设备可以通过软件控制连接于从设备 NSS 引脚的端口电平选择每个从设备, 很明显, 只有一个主设备可以驱动通讯网络。为了防止 MISO 总线冲突, 同一时间只允许一个从设备与主设备通讯。在主设备模式中, NSS 引脚状态关联 SPI 状态寄存器 SPSTA[4] 的 MODF 标志位以防止多个主设备驱动 MOSI 和 SCK。

下列情况, NSS 引脚可以作为普通 IO 或其它功能使用:

设备作为主设备, 寄存器 SPCON[5] 寄存器的 SSDIS 位置 1。这种软件配置仅仅存在于通讯网络中只有一个主设备的情况, 因此, SPI 状态寄存器 SPSTA[4] 的 MODF 标志位不会被置 1。

设备软件配置为从设备, SPI 控制寄存器 SPCON 的 CPHA 位和 SSDIS 位置 1。这种软件配置情况存在于只有一个主设备一个从设备的通讯网络中, 因此, 设备总是被选中的, 主设备也不需要控制从设备的 NSS 引脚选择其作为通讯目标。

注意: 当 CPHA=0 时, NSS 引脚产生下降沿表示启动发送。

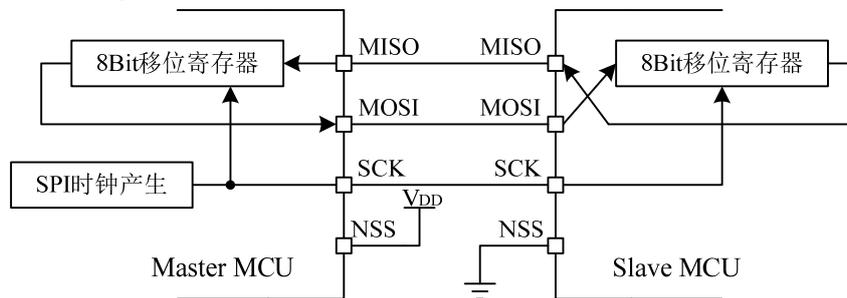
14.2 工作模式

SPI 可软件配置为主模式或从模式中的一种。SPI 模块的软件配置和初始化通过设置寄存器 SPICON 和 SPISTA 来完成。软件配置完成后，通过设置 SPICON、SPISTA、SPIDAT 来完成数据传送。

在 SPI 通讯期间，数据同步地被串行的移进移出。串行时钟线（SCK）使两条串行数据线（MOSI 和 MISO）上数据的移动和采样保持同步。从设备选择 NSS 可以独立地选择 SPI 从属设备；如果从设备没有被选中，则不能参与 SPI 总线上的活动。

当 SPI 主设备通过 MOSI 线传送数据到从设备时，从设备通过 MISO 线发送数据到主设备作为响应，这就实现了在同一时钟下数据发送和接收的同步全双工传输。发送移位寄存器和接收移位寄存器使用相同的特殊功能器地址，对 SPI 数据寄存器 SPDAT 进行写操作将写入发送移位寄存器，对 SPIDAT 寄存器进行读操作将获得接收移位寄存器的数据。

图 14-1 全双工主从工作模式图



主模式->模式启动

SPI 主设备控制 SPI 总线上所有数据传送的启动。当寄存器 SPICON[4] 的 MSTR 位置 1 时，SPI 在主模式下运行，只有一个主设备可以启动传送。

主模式->发送

在 SPI 主模式下，写一个字节数据到 SPI 数据寄存器 SPIDAT，数据将会写入发送移位寄存器。如果发送移位寄存器已经存在一个数据，那么主 SPI 产生一个 WCOL 信号以表明写入太快。但是在发送移位寄存器中的数据不会受到影响，发送也不会中断。另外如果发送移位寄存器为空，那么主设备立即按照 SCK 上的 SPI 时钟频率串行地移出发送移位寄存器中的数据到 MOSI 线上。当传送完毕，寄存器 SPISTA[7] 的 SPIF 位被置 1。如果 SPI 中断被允许，当 SPI F 位置 1 时，也会产生一个中断。

主模式->接收

当主设备通过 MOSI 线传送数据给从设备时，相对应的从设备同时也通过 MISO 线将其发送移位寄存器的内容传送给主设备的接收移位寄存器，实现全双工操作。因此，SPIF 标志位置 1 即表示传送完成也表示接收数据完毕。从设备接收的数据按照 MSB 优先的传送方向存入主设备的接收移位寄存器。当一个字节的数据完全被移入接收寄存器时，处理器可以通过读 SPIDAT 寄存器获得该数据。

从模式->模式启动

当寄存器 SPICON[4] 的 MSTR 位清 0，SPI 在从模式下运行。在数据传送之前，从设备的 NSS 引脚必须被置低，而且必须保持低电平直到一个字节数据传送完毕。

从模式->发送与接收

从属模式下，按照主设备控制的 SCK 信号，数据通过 MOSI 引脚移入，MISO 引脚移出。一个位计数器记录 SCK 的边沿数，当接收移位寄存器移入 8 位数据（一个字节）同时发送移位寄存器移出 8 位数据（一个字节），SPIF 标志位被置 1。数据可以通过读取 SPIDAT 寄存器获得。如果 SPI 中断被允许，当 SPIF 置 1 时，也会产生一个中断。

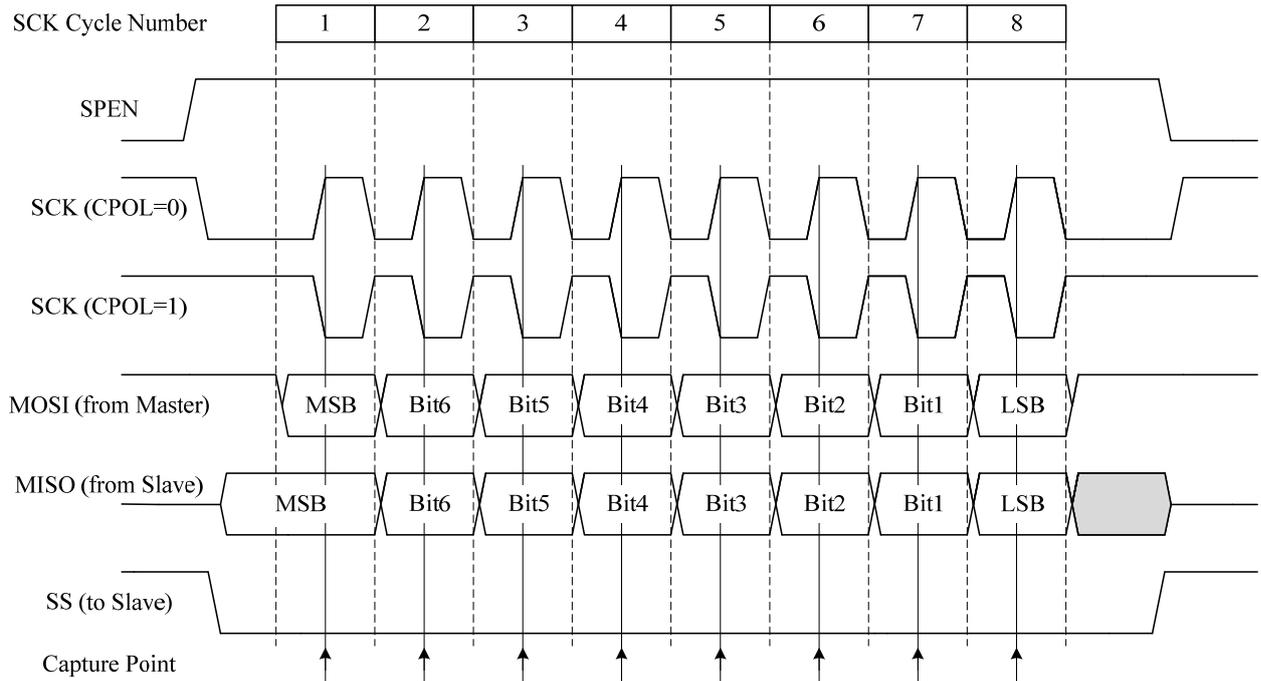
SPI 从设备不能启动数据传送，所以 SPI 从设备必须在主设备开始一次新的数据传送之前将要传送的数据写入发送移位寄存器。如果从设备在第一次开始发送之前未写入数据，从设备将传送“0x00”字节给主设备。如果写 SPIDAT 操作发生在传送过程中，那么 SPI 从设备的 WCOL 标志位置 1，即如果

传送移位寄存器已经含有数据，SPI 从设备的 WCOL 位置 1，表示写 SPIDAT 冲突。但是移位寄存器的数据不受影响，传送也不会被中断。

传送形式

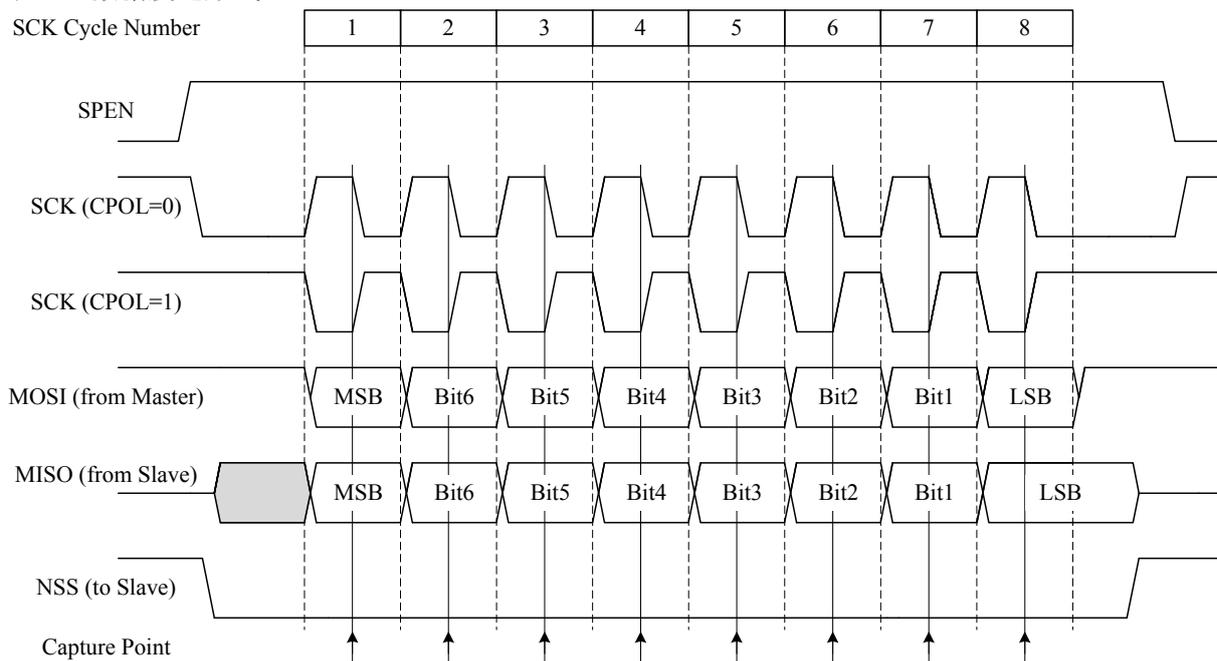
通过软件设置寄存器 SPICON 的 CPOL 位和 CPHA 位，用户可以选择 SPI 时钟极性和相位的四种组合方式。CPOL 位定义时钟的极性，即空闲时的电平状态，它对 SPI 传输格式影响不大。CPHA 位定义时钟的相位，即定义允许数据采样移位的时钟边沿。在主从通讯的两个设备中，时钟极性相位的设置应一致。

图 14-2 数据发送形式 (CPHA=0)



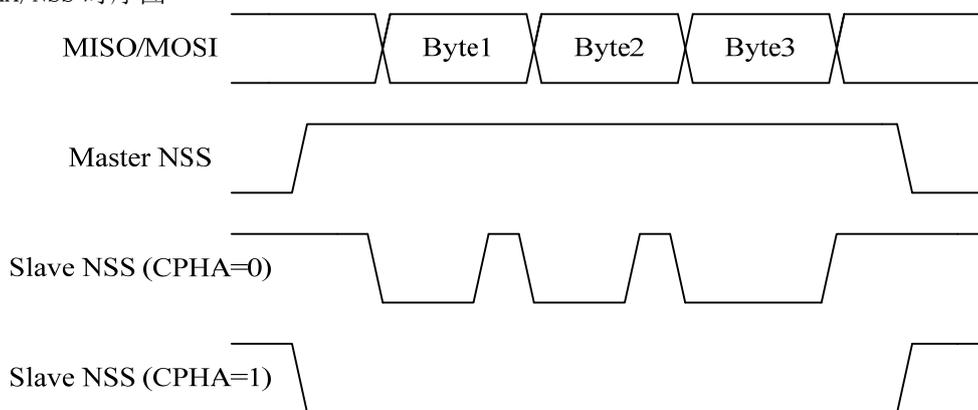
如果 CPHA=0, SCK 的第一个沿捕获数据，从设备必须在 SCK 的第一个沿之前将数据准备好，因此，NSS 引脚的下降沿从设备开始发送数据。NSS 引脚在每次传送完一个字节之后必须被拉高，在发送下一个字节之前重新设置为低电平，因此当 CPHA=0, SSDIS 不起作用。

图 14-3 数据发送形式 (CPHA=1)



如果 CPHA=1, 主设备在 SCK 的第一个沿将数据输出到 MOSI 线上, 从设备把 SCK 的第一个沿作为开始发送信号。用户必须在第一个 SCK 的第二个沿之前完成写 SPIDAT 的操作。NSS 引脚在每个字节数据的传送过程始终保持低电平。这种数据传输形式是一个主设备一个从设备之间通信的首选形式。

图 14-4 CPHA/NSS 时序图



注意: 当 SPI 用作从设备模式, 且寄存器 SPICON[3] 的 CPOL 位清 0, SCK/P2.5 端口必须设置为输入模式, 并在 SPIEN 位置 1 前打开上拉电阻。

14.3 SPSTA 异常情况

SPISTA 寄存器中的标志位表示在 SPI 通讯中的出错情况:
模式错误 (MODF)

SPI 主模式下的模式错误表明 NSS 引脚上的电平状态与实际的设备模式不一致。SPISTA 寄存器中 MODF 位置 1 后, 表明系统控制存在多主设备冲突的问题。这种情况下, SPI 系统受到如下影响:

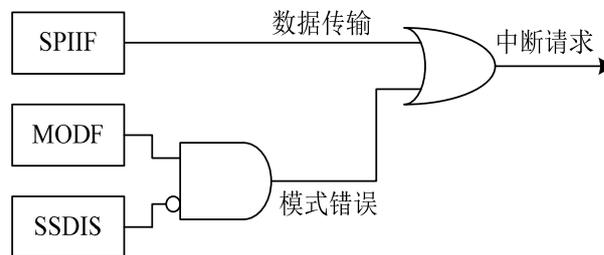
- (1)产生 SPI 接收/错误 CPU 中断请求;
- (2)寄存器 SPISTA[6] 的 SPEN 位清 0, SPI 被禁止;
- (3)寄存器 SPICON[4] 的 MSTR 位清 0。

(4)当寄存器 SPICON[5]的 SSDIS 清 0，NSS 引脚信号为低时，MODF 标志位置 1。然而，对于只有一个主设备的系统来说，主设备的 NSS 引脚被拉低，那决不是另外一个主设备试图驱动网络。这种情况下，为防止 MODF 置 1，可使 SPICON 寄存器中的 SSDIS 位置 1，NSS 引脚作为普通 I/O 口或是其它功能引脚。
(5)重新启动串行通信时，用户必须将 MODF 位软件清 0，将寄存器 SPICON[4]中的 MSTR 位和寄存器 SPISTA[6]的 SPIEN 位置 1，重新启动主模式。

写冲突 (WCOL)

在发送数据序列期间写入 SPIDAT 寄存器会引起的写冲突，寄存器 SPISTA[6]的 WCOL 位置 1。WCOL 位置 1 不会引起中断，发送也不会中止。WCOL 位需由软件清 0。中断两种 SPI 状态标志 SPIF 或 MODF 能产生一个 CPU 中断请求。串行外围设备数据发送标志，SPIF：完成一个字节发送后由硬件置 1。模式错误标志，MODF：该位被置 1 表示 NSS 引脚上的电平与 SPI 模式不一致。SSDIS 位为 0 并且 MODF 置 1 将产生 SPI 接收器/出错 CPU 中断请求。当 SSDIS 置 1 时，无 MODF 中断请求产生。中断需要软件清除。

图 14-6 SPI 中断示意图



注意：中断需要软件清除

15. 通用串口(UART)接口

BJ8M302A 内部集成了 1 组通用串口 (Universal Asynchronous Receiver/Transmitter) 接口。UART 由数据发送脚 UART_TX(P35)，数据接收脚 RX(P36)组成。提供一个可编程全双工串行通信接口，该接口能同时进行数据的发送和接受。TX_PND 和 RX_PND 分别为发送标志位和接收标志位，发送完成或接收完成会置 1，软件清 0。

表 15-1 与 UART 通讯相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
98h	SOCON	UARTEN	-	-	UART9	TX9	RX9	TXIF	RXIF
99h	S0BUF	S0BUF[7:0]							
9Ah	S0OVR	S0OVR[7:0]							
9Bh	S0OVRH	S0OVR[10:8]							
B8h	IEN1	-	-	SORXIE	S0TXIE	SPIIE	P3EA	P2EA	I2CIE

寄存器定义 15-1 SOCON UART 通讯 0 通道控制寄存器

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SOCON	UARTEN	-	-	UART9	TX9	RX9	TXIF	RXIF
98h	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7UARTEN: UART 使能位 1: 使能 UART; 0: 禁止 UART

Bit6-5 未使用

Bit4 UART9: UART 第 9 位使能位 1: 打开第 9 位; 0: 关闭第 9 位

Bit3TX9: UART 发送 BIT9 数据位
 Bit2 RX9: UART 接收 BIT9 数据位
 Bit1TXIF: UART 发送中断申请标志位
 Bit0 RXIF: UART 接收中断申请标志位

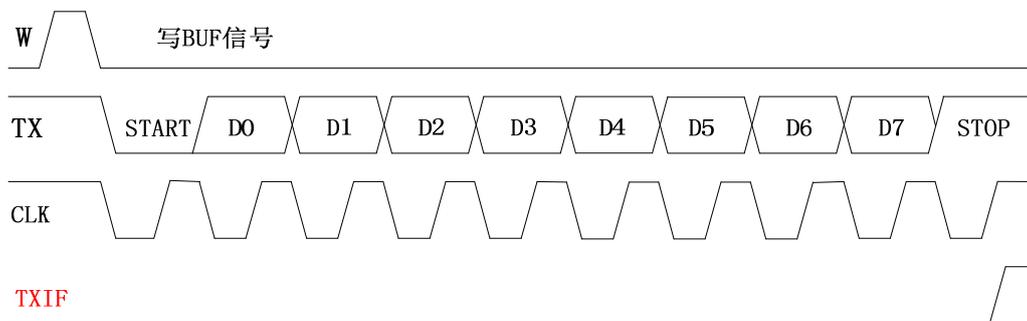
15.1 UART 工作模式

模式 1: 8 位 UART

提供 10 位全双工异步通讯, 10 位由一个起始位 (逻辑 0), 8 个数据位 (低位为第一位) 和一个停止位 (逻辑 1) 组成。在接收时, 这 8 个数据位存储在 SBUF 中。

任何将 S0BUF 作为目标寄存器的写操作都会启动发送, 数据位时间与波特率移位时钟是同步的, 与对 S0BUF 的写操作不同步。起始位首先在 TX 引脚上移出, 然后是 8 位数据位。在发送移位寄存器中的所有 8 位数据都发送完后, 停止位在 TX 引脚上移出, 在停止位发出的同时 TXIF 标志置 1。TXIF 可以软件清掉。

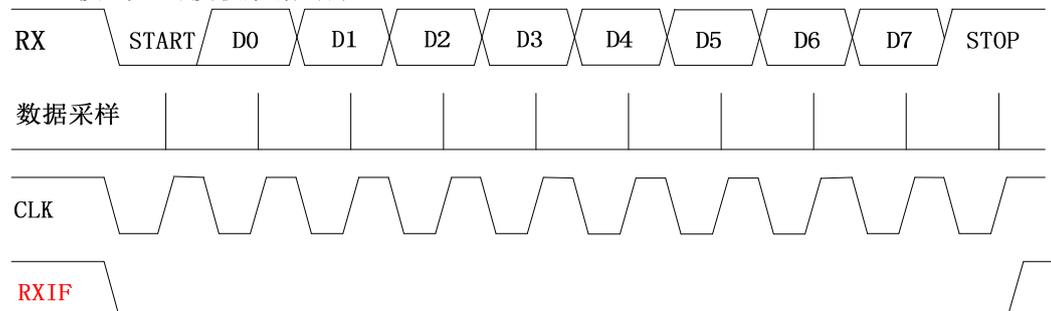
图 15-1 UART 模式 1 的发送数据时序



当 RX 引脚检测到下降沿时串行口开始接收串行数据。CPU 在 RX 数据位的正中进行采样收。如果所接收的第一位不是 0, 说明这位不是一帧数据的起始位, 该位被忽略, 接收电路被复位, 等待 RX 引脚上另一个下降沿的到来。若起始位有效, 则移入移位寄存器, 并接着移入其它位到移位寄存器。8 个数据位和 1 个停止位移入之后, 移位寄存器的内容被装入 S0BUF, RXIF 置 1。

在接收停止位后, 接收器将重新去探测 RX 端的另一个下降沿。用户必须用软件清 RXIF, 然后才能再次接收。

图 15-2 UART 模式 1 的接收数据时序



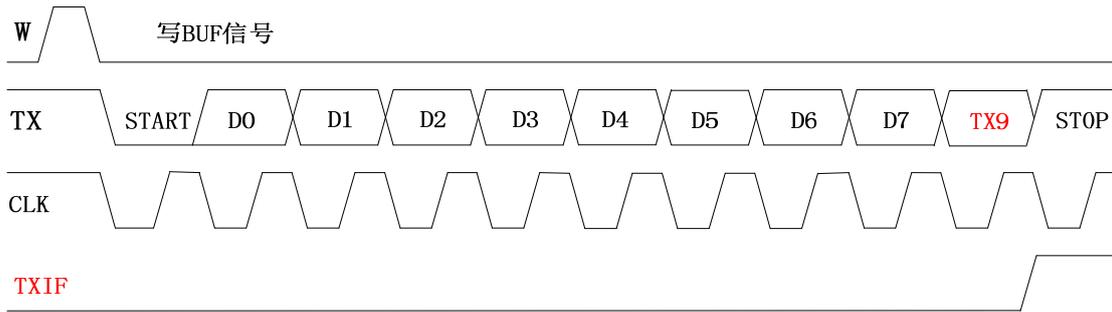
模式 2: 9 位 UART

提供 11 位全双工异步通讯, 一帧由一个起始位 (逻辑 0), 8 个数据位 (低位为第一位), 一个可编程的第 9 数据位和一个停止位 (逻辑 1) 组成。

在数据传送时, 第 9 数据位 (寄存器 S0CON 中的 TX9) 可以写 0 或 1, 例如, 可写入奇偶位 P, 或

用作多机通讯中的数据/地址标志位。当接收到数据时，第 9 数据位移入 RX9 而停止位不保存。任何将 SOBUF 作为目标寄存器的写操作都会启动发送，同时也将 TX9 载入到发送移位寄存器的第 9 位中。数据位时间与波特率移位时钟是同步的，与对 SOBUF 的写操作不同步。起始位首先在 TX 引脚上移出，然后是 9 位数据。在发送转换寄存器中的所有 9 位数据都发送完后，停止位在 TX 引脚上移出，在停止位发出的同时 TX_PND 标志置 1。TXIF 可以软件清掉。

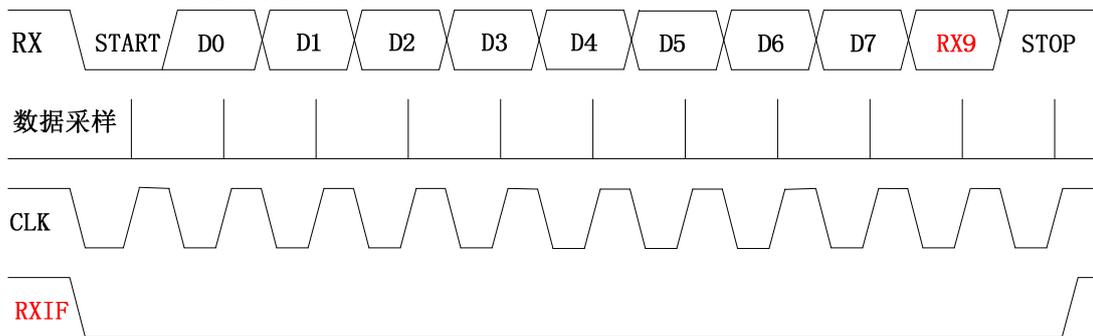
图 15-3 UART 模式 2 的发送数据时序



当 RX 引脚检测到下降沿时串行口开始接收串行数据。CPU 在 RX 数据位的正中进行采样接收。如果所接收的第一位不是 0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待 RX 引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。9 个数据位和 1 个停止位移入之后，移位寄存器的内容被分别装入 SOBUF 和 RB8 中，RXIF 置 1。

在接收停止位后，接收器将重新去探测 RX 端的另一个下降沿。用户必须用软件清 RXIF，然后才能再次接收。

图 15-4 UART 模式 2 的接收数据时序



15.2 波特率

UART 它实质上是一个 16 位递增计数器，其计数时钟为 CPUCLK。用户软件配置寄存器 SRELH 和 SRELL 来软件配置计数器的溢出值，从而实现需要的波特率。

波特率=CPUCLK/SREL[10:0]: CPUCLK=7.937M/2=3.9685M

表 15-2 UART 波特率列表

波特率	Srel	实际波特率	误差%
2400	1653 / 0x675	2400.786449	0.0327
4800	827 / 0x33B	4798.669891	0.0277
9600	413 / 0x19D	9608.958838	0.0933
14400	276 / 0x114	14378.62319	0.1484
19200	207 / 0xCF	19171.49758	0.1484
38400	103 / 0x67	38529.12621	0.3362
56000	71 / 0x47	55894.3662	0.1886
57600	69 / 0x45	57514.49275	0.1484
115200	34 / 0x22	116720.5882	1.3199

16. 集成电路总线(IIC)接口

BJ8M302A 提供一个可编程全双工串行集成电路总线 (Inter-Integrated Circuit, IIC) 接口。该接口能同时进行数据的发送和接受, 也可以作为一个同步移位寄存器使用。

16.1 IIC 特性

- (1) 多主机功能: 如果两个或更多主机同时初始化数据传输, 可以通过冲突检测与仲裁来防止数据被破坏。任何能够进行发送和接收的设备都可以成为主机, 但在任何时间点上只能有一个主机
- (2) I2C 主设备功能: 产生时钟; 产生起始与停止信号
- (3) I2C 从设备功能: 可编程的 I2C 从设备地址; 停止位检测
- (4) 支持通用广播呼叫
- (5) 支持不同的波特率
- (6) 丰富的状态标志

表 16-1 与 I2C 相关的寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
E9h	I2C0SPD	I2CSPD[7:0]							
B8h	IEN1	-	-	S0RXIE	S0TXIE	SPIIE	P3EA	P2EA	I2CIE
ECh	I2C0CON	-	I2CEN	STA	STO	I2CIF	AA	-	-
EDh	I2C0STA	I2CSTA[4:0]					-	-	-
EAh	I2C0DAT	I2CDAT[7:0]							
EBh	I2C0ADR	I2CADR[7:0]							

寄存器定义 16-1I2C0CON 控制字

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2C0CON	-	I2CEN	STA	STO	I2CIF	AA	-	-
ECh	-	R/W	R/W	R/W	R/W	R	-	-

Bit7 未使用

Bit6 I2CEN: I2C 使能位 1: 使能 I2C; 0: 禁止 I2C

Bit5 STA: I2C 起始标志 当 STA = 1, 时, 如果总线是空闲的, 其实信号会发出

Bit4 STO: I2C 停止标志 当 STO = 1, 且处于主模式, stop 信号会被发出

Bit3 I2CIF: I2C 中断标志 改位必须由软件写 0 清除

Bit2 AA: I2C 应答标志

当 aa=1, 满足下面条件下将会返回应答

从模式下, 收到自己的从地址

从模式下, 当 gc 位被设置并且接收到自己的广播地址时

主模式下, 接收到数据

从模式接收到数据

当 aa=0, 任何情况下都不会产生应答信号

Bit1-0 未使用

寄存器定义 16-2I2C0STA I2C 状态寄存器

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2C0STA	I2CSTA[4:0]					-	-	-
EDh	R					-	-	-

Bit7-3I2CSTA[4:0]: I2C 状态列表, 详情请查阅主发送器模式、主接收器模式、从发送器模式、从接收器模式下的 I2C 状态列表

Bit2-0 未使用

寄存器 I2C0SPD(D1h) 可读写, IIC 时钟频率控制寄存器
寄存器 I2C0DAT(3Ch) 可读写, IIC 数据发送接收寄存器
寄存器 I2C0ADR(3Dh) 可读写, IIC 通讯从地址寄存器

16.2 功能描述

I2C 模块接收和发送数据, 并将数据从串行转换成并行, 或并行转换成串行。可以开启和禁止中断。接口通过数据引脚 (SDA) 和时钟引脚 (SCL) 连接到 I2C 总线。

I2C 接口可以下述 4 种模式中的一种运行:

- (1) 主发送器模式
- (2) 主接收器模式
- (3) 从发送器模式
- (4) 从接收器模式

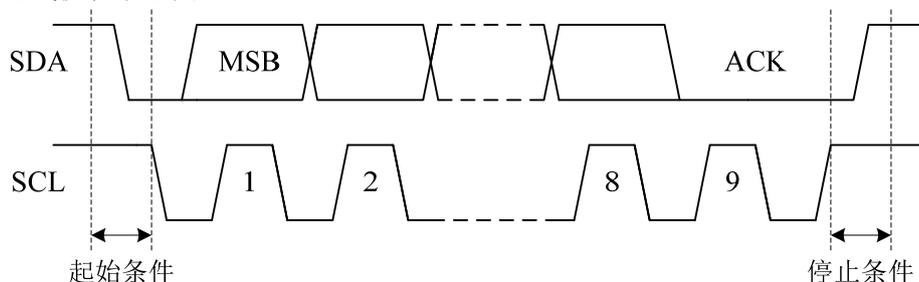
该模块默认工作于从模式。接口在生成起始条件后自动地由从模式切换到主模式; 当仲裁丢失或产生停止信号时, 则由主模式切换到从模式。

允许多主机功能, 即如果两个或更多主机同时初始化数据传输, 可以通过冲突检测与仲裁来防止数据被破坏。任何能够进行发送和接收的设备都可以成为主机, 但在任何时间点上只能有一个主机。

主模式时, I2C 接口启动数据传输并产生时钟信号。串行数据传输总是以起始条件开始并以停止条件结束。起始条件和停止条件都是在主模式下由软件控制产生 (I2CCON 寄存器中的 STA 和 STO 位)。从模式时, I2C 接口能识别它自己的 7 位地址和广播呼叫地址。

数据和地址按 8 位/字节进行传输, 高位在前。跟在起始条件后的 1 个字节是地址。地址只在主模式下发送。在 1 个字节传输的 8 个时钟后的第 9 个时钟期间, 接收器必须回送一个应答位 (ACK) 给发送器。软件可以通过 I2CCON 寄存器中的 AA 位开启或禁止应答 (ACK), 并可以在 I2CADR 寄存器中设置 I2C 接口的地址。

图 16-1 I2C 总线协议示意图



I2C 速度软件配置方法:

根据 CPU 时钟和 I2C 时钟软件配置 I2CSPD 寄存器, 寄存器最小软件配置大于等于 4

$$I2CSPD = (CPUCLK / I2CCLK / 4) - 1$$

例: 需 I2C 的 SCL 需要以 100K 的频率通信 (系统程序跑 4M 的频率), 则:

I2CSPD=(4000 000/100 000/4)-1 =39

16.3 主发送器模式

在主发送器模式下，当 I2C 的状态（I2CSTA 寄存器）发生转变时，中断请求位（I2CCON 寄存器中的 SI 位）将被置 1，若 IEN1 寄存器中的 I2CIE 位和 EA 位被使能，I2C 将发出中断请求。主发送器模式下的 I2C 状态列表如下：

表 16-2 主发送器模式下的 I2C 状态列表

状态码	I2C 状态	应用软件操作					I2C 硬件的下一步动作
		写/读 I2CDAT	写 I2CCON				
			STA	STO	I2CIF	AA	
08H	1 个起始位已发送	写 SLA+W	X	0	0	X	SLA+W 将被发送，并收到 ACK
10H	1 个重新起始位已发送	写 SLA+W	X	0	0	X	SLA+W 将被发送，并收到 ACK
		写 SLA+R	X	0	0	X	SLA+R 将被发送，I2C 被切换到主接收器模式
18H	“SLA+W”已发送，且 ACK 已收到	写数据字节	0	0	0	X	数据字节将被发送，并收到 ACK
		无操作	1	0	0	X	重新起始位将被发送
		无操作	0	1	0	X	结束位将被发送；STO 标志位将被清 0
		无操作	1	1	0	X	起始位与结束位将被发送；STO 标志位将被清 0
20H	“SLA+W”已发送，且 ACK 未收到	写数据字节*	0	0	0	X	数据字节将被发送，并收到 ACK
		无操作	1	0	0	X	重新起始位将被发送
		无操作	0	1	0	X	结束位将被发送；STO 标志位将被清 0
		无操作	1	1	0	X	起始位与结束位将被发送；STO 标志位将被清 0
28H	I2CDAT 中的数据字节已发送，且 ACK 已收到	写数据字节	0	0	0	X	数据字节将被发送，并收到 ACK
		无操作	1	0	0	X	重新起始位将被发送
		无操作	0	1	0	X	结束位将被发送；STO 标志位将被清 0
		无操作	1	1	0	X	起始位与结束位将被发送；STO 标志位将被清 0
30H	I2CDAT 中的数据字节已发送，且 ACK 未收到	写数据字节*	0	0	0	X	数据字节将被发送，并收到 ACK
		无操作	1	0	0	X	重新起始位将被发送
		无操作	0	1	0	X	结束位将被发送；STO 标志位将被清 0
		无操作	1	1	0	X	起始位与结束位将被发送；STO 标志位将被清 0

- (1) “SLA” 表示 7 位的通信目标从机地址，“R” 表示将从目标从机读数据（R/W=1），此位将与“SLA”一起发送，“W” 表示将向目标从机写数据（R/W=0），此位也将与“SLA”一起发送。
- (2) *仅适用于 NACK 并不表示通信结束的那些应用。

主发送器模式的典型软件配置方法(I2C 做主机写):

- (1) 软件配置 I2CCON 的 I2CEN 位使能 I2C, 配置 I2C 的速度(I2CSPD 寄存器)。
- (2) 软件配置 I2CCON 的 STA 位发送起始信号。
- (3) 查询 I2CCON 的 I2CIF 位, 判断中断标志位是否置位。
- (4) 若已置位, 读 I2CSTA, 判断当前状态码是否为 08H。
- (5) 清除 STA 位
- (6) 软件配置 I2CDAT, 发送目标从机地址 SLA+W。
- (7) 对 I2CIF 位写 0 清除标志位。
- (8) 查询 I2CIF 位, 判断中断标志位是否置位。
- (8) 若已置位, 读 I2CSTA, 判断当前状态码是否为 18H。
- (10) 将要发送的数据写入 I2CDAT。
- (11) 对 I2CIF 位写 0 清除标志位。
- (12) 读查询 I2CIF 位, 判断中断标志位是否置位。
- (13) 若已置位, 读 I2CSTA, 判断当前状态码是否为 28H。
- (14) 重复 (10) ~ (13) 。
- (15) 数据全部发送完毕, 则发送停止信号。
- (16) 对 I2CIF 位写 0 清除标志位。(不清会有问题)

16.4 主接收器模式

在主接收器模式下, 当 I2C 的状态 (I2CSTA 寄存器) 发生转变时, 中断请求位 (I2CCON 寄存器中的 I2CIF 位) 将被置 1, 若 IEN1 寄存器中的 I2CIE 位和 EA 位被使能, I2C 将发出中断请求。主接收器模式下的 I2C 状态列表如下:

表 16-3 主接收器模式下的 I2C 状态列表

状态码	I2C 状态	应用软件操作					I2C 硬件的下一步动作
		写/读 I2CDAT	写 I2CCON				
			STA	STO	I2CIF	AA	
08H	1 个起始位已发送	写 SLA+R	X	0	0	X	SLA+R 将被发送, 并收到 ACK
10H	1 个重新起始位已发送	写 SLA+R	X	0	0	X	SLA+R 将被发送, 并收到 ACK
		写 SLA+W	X	0	0	X	SLA+W 将被发送, I2C 被切换到主发送器模式
40H	“SLA+R” 已发送, 且 ACK 已收到	无操作	0	0	0	0	数据字节将被接收, 并未收到 ACK
		无操作	0	0	0	1	数据字节将被接收, 并收到 ACK
48H	“SLA+R” 已发送, 且 ACK 未收到	无操作	1	0	0	X	重新起始位将被发送
		无操作	0	1	0	X	结束位将被发送; STO 标志位将被清 0
		无操作	1	1	0	X	起始位与结束位将被发送; STO 标志位将被清 0
50H	接收到的数据字节已存入 I2CDAT 中, 且 ACK 已收到	读数据字节	0	0	0	0	数据字节将被接收, 并未收到 ACK
		读数据字节	0	0	0	1	数据字节将被接收, 并收到 ACK
58H	接收到的数据字节已存入 I2CDAT 中, 且 ACK 未收到	读数据字节	1	0	0	X	重新起始位将被发送
		读数据字节	0	1	0	X	结束位将被发送; STO 标志位将被清 0
		读数据字节	1	1	0	X	起始位与结束位将被发送; STO 标志位将被清 0

主接收器模式的典型软件配置方法 (I2C 做主机读):

- (1) 软件配置 I2CCON 的 I2CEN 位使能 I2C, 配置 I2C 的速度 (I2CSPD 寄存器)
- (2) 软件配置 I2CCON 的 STA 位发送起始信号。
- (3) 查询 I2CCON 的 I2CIF 位, 判断中断标志位是否置位。
- (4) 若已置位, 读 I2CSTA, 判断当前状态码是否为 08H。
- (5) 清除 STA 位
- (6) 软件配置 I2CDAT, 发送目标从机地址 SLA+W。
- (7) 对 I2CIF 位写 0 清除标志位。
- (8) 查询 I2CCON 的 I2CIF 位, 判断中断标志位是否置位。
- (9) 若已置位, 读 I2CSTA, 判断当前状态码是否为 18H。
- (10) 软件配置 I2CDAT, 发送写地址 (主机写在从机内部的地址)。
- (11) 对 I2CIF 位写 0 清除标志位。
- (12) 若已置位, 读 I2CSTA, 判断当前状态码是否为 28H。
- (13) 发送 Restart, 软件配置 I2CCON 的 STA 位发送起始信号。
- (14) 对 I2CIF 位写 0 清除标志位。
- (15) 查询 I2CCON 的 I2CIF 位, 判断中断标志位是否置位。
- (16) 若已置位, 读 I2CSTA, 判断当前状态码是否为 10H。
- (17) 清除 STA 位
- (18) 软件配置 I2CDAT, 发送目标从机地址 SLA+R。
- (19) 对 I2CIF 位写 0 清除标志位。
- (20) 查询 I2CCON 的 I2CIF 位, 判断中断标志位是否置位。

- (21) 若已置位，读 I2CSTA，判断当前状态码是否为 40H
- (22) 对 I2CIF 位写 0 清除标志位。
- (23) 查询 I2CCON 的 I2CIF 位，判断中断标志位是否置位。
- (24) 若已置位，读 I2CSTA，判断当前状态码是否为 50H。
- (25) 读取 I2CDAT，读到第一笔数据。
- (26) 重复 (22) ~ (25) 直到读取倒数第二笔数据。
- (27) 对 AA 位写 0 清除。
- (28) 对 I2CIF 位写 0 清除标志位。
- (29) 查询 I2CCON 的 I2CIF 位，判断中断标志位是否置位。
- (30) 若已置位，读 I2CSTA，判断当前状态码是否为 58H。
- (31) 读取 I2CDAT，读到最后一笔数据。
- (32) 发送停止信号。
- (33) 对 I2CIF 位写 0 清除标志位。

例 16-1 主模式 IIC 数据读写示例

```
#include "REG8M302A.h"
#include "Bjx_Type.h"

void IICInit();
unsigned char IICRead(u8 addr);
void IICWrite(u8 addr,u8 dat);

void main(void)
{
    unsigned char ReadByte;
    IICInit();
    IICWrite(0x01,0x55);//向 0x01 地址写入 0x55
    ReadByte =IICRead(0x01);//读取 0x01 地址的数据并存入 ReadByte
    while(1){}
}

void IICInit()
{
    I2CSPEED = 0X04;
    I2CCON =(1<<6)|(1<<2);
    IEN1 &= ~(1<<0);
}
```

例 16-1 主模式 IIC 数据读写示例(续: 读子程序)

```
//*****  
// 函数功能: IIC 读取 24c02 一个 Byte  
// 参数: addr:待读取的地址  
// 返回值: 读取的数据  
//*****  
unsigned char IICRead(u8 addr)  
{  
    unsigned char dat;  
    //1、发送起始信号  
    I2CCON |= (1<<5);  
    while(!(I2CCON&0x08));  
    while(I2CSTA!=0x08);  
    I2CCON &=~(1<<5); //清除 Sta  
    //2、发送器件地址+写  
    I2CDAT = 0xa0;  
    I2CCON &=~(1<<3); //清除 SI  
    while(!(I2CCON&0x08));  
    while(I2CSTA!=0x18);  
    //3、发送要写 EEPROM 的地址  
    I2CDAT = addr;  
    I2CCON &=~(1<<3); //清除 SI  
    while(!(I2CCON&0x08));  
    while(I2CSTA!=0x28);  
    //4、发送 Restart  
    I2CCON |= (1<<5);  
    I2CCON &=~(1<<3); //清除 SI  
    while(!(I2CCON&0x08));  
    while(I2CSTA!=0x10);  
    I2CCON &=~(1<<5); //清除 Sta  
    //5、发送器件地址+读  
    I2CDAT = 0xa1;  
    I2CCON &=~(1<<3); //清除 SI  
    while(!(I2CCON&0x08));  
    while(I2CSTA!=0x40);  
    //6、读取数据  
    I2CCON &=~(1<<2); //清除 AA  
    I2CCON &=~(1<<3); //清除 SI  
    while(!(I2CCON&0x08)); //等待 I2C 状态 SI 发生变化  
    while(I2CSTA!=0x58); //判断 I2CDAT 中的数据是否收到  
    dat = I2CDAT;  
    //7、发送停止信号  
    I2CCON |= (1<<4);  
    I2CCON &=~(1<<3); //清除 SI  
    return dat;  
}
```

例 16-1 主模式 IIC 数据读写示例(续: 写子程序)

```

//*****
//函数功能: IIC 写 24c02 一个 Byte
// 参数: addr:待写的地址
//       dat:待写入的数据
//*****
void IICWrite(u8 addr,u8 dat)
{
//1、发送起始信号
    I2CCON |= (1<<5);
    while(!(I2CCON&0x08));
    while(I2CSTA!=0x08);
    I2CCON &=~(1<<5); //清除 Sta
//2、发送器件地址+写
    I2CDAT = 0xa0;
    I2CCON &=~(1<<3); //清除 SI
    while(!(I2CCON&0x08));
    while(I2CSTA!=0x18);
//3、发送要写 EEPROM 的地址
    I2CDAT = addr;
    I2CCON &=~(1<<3); //清除 SI
    while(!(I2CCON&0x08));
    while(I2CSTA!=0x28);
//4、连续写入数据到 EEPROM
    I2CDAT = dat;
    I2CCON &=~(1<<3); //清除 SI
    while(!(I2CCON&0x08));
    while(I2CSTA!=0x28);
//5、发送停止信号
    I2CCON |= (1<<4);
    I2CCON &=~(1<<3); //清除 SI
}

```

16.5 从发送器模式

在从发送器模式下，当 I2C 的状态（I2CSTA 寄存器）发生转变时，中断请求位（I2CCON 寄存器中的 I2CIF 位）将被置 1，若 IEN1 寄存器中的 I2CIE 位和 EA 位被使能，I2C 将发出中断请求。从发送器模式下的 I2C 状态列表如下：

表 16-4 从发送器模式下的 I2C 状态列表

状态码	I2C 状态	应用软件操作					I2C 硬件的下一步动作
		写/读 I2CDAT	写 I2CCON				
			STA	STO	I2CIF	AA	
A8H	自己作为从机的地址“SLA+R”已收到，且 ACK 已发送	写数据字节	X	0	0	0	最后一个数据字节将被发送，并收到 ACK
		写数据字节	X	0	0	1	一个数据字节将被发送，并收到 ACK
B0H	多主机下通过仲裁，主机控制权丢失；自己作为从机的地址“SLA+R”已收到，且 ACK 已发送	写数据字节	X	0	0	0	最后一个数据字节将被发送，并收到 ACK
		写数据字节	X	0	0	1	一个数据字节将被发送，并收到 ACK

B8H	数据字节已发送, 且 ACK 已收到	写数据字节	X	0	0	0	最后一个数据字节将被发送, 并收到 ACK
		写数据字节	X	0	0	1	一个数据字节将被发送, 并收到 ACK
C0H	数据字节已发送, 且 ACK 未收到	无操作	0	0	0	0	切换到不可寻址的从机模式; 对自己的从机地址和通用广播呼叫不响应
		无操作	0	0	0	1	切换到不可寻址的从机模式; 对自己的从机地址和通用广播呼叫响应
		无操作	1	0	0	0	切换到不可寻址的从机模式; 对自己的从机地址和通用广播呼叫不响应; 当总线空闲时发送起始位
		无操作	1	0	0	1	切换到不可寻址的从机模式; 对自己的从机地址和通用广播呼叫响应; 当总线空闲时发送起始位
C8H	最后一个数据字节已发送, 且 ACK 已收到	无操作	0	0	0	0	切换到不可寻址的从机模式; 对自己的从机地址和通用广播呼叫不响应
		无操作	0	0	0	1	切换到不可寻址的从机模式; 对自己的从机地址和通用广播呼叫响应
		无操作	1	0	0	0	切换到不可寻址的从机模式; 对自己的从机地址和通用广播呼叫不响应; 当总线空闲时发送起始位
		无操作	1	0	0	1	切换到不可寻址的从机模式; 对自己的从机地址和通用广播呼叫响应; 当总线空闲时发送起始位

从发送器模式的典型软件配置方法:

- (1) 软件配置从机地址 (I2CADR)。
- (2) 使能 IEN1 寄存器的 I2CIE 位与 EA 打开 I2C 中断。
- (3) 软件配置 I2CCON 的 I2CEN 位使能 I2C。
- (4) 若有主机发送的从机地址与该从机地址吻合, 将会触发 I2C 中断。
- (5) 在 I2C 中断服务程序中读取 I2CDAT, 此次 I2CDAT 中为自己的从机地址, 并确认最低位为 1 (R/W=1 主机从目标从机读数据); 同时读取 I2CSTA 并判断当前状态码是否为 A8H。若是, 则将要发送的第一个数据字节写入 I2CDAT 并清除 I2CIF 标志位。
- (6) 若数据已发送, 且主机有 ACK 应答, 将再次触发 I2C 中断。
- (7) 在 I2C 中断服务程序中读取 I2CSTA 并判断当前状态码是否为 B8H。若是, 则将要发送的下一个数据字节写入 I2CDAT 并清除 I2CIF 标志位。
- (8) 重复 (6) ~ (7)。
- (9) 当从机数据发送最后一个字节数据时, 还需将 I2CCON 中的 AA 清 0。
- (10) 若最后一个字节数据已发送, 且主机有 ACK 应答, 将再次触发 I2C 中断。
- (11) 在 I2C 中断服务程序中读取 I2CSTA 并判断当前状态码是否为 C8H。若是, 则无需做任何操作, 从机将切换到不可寻址的从机模式并结束数据发送。
- (12) 若主机不想继续接收数据, 则主机无 ACK 应答。在此时触发的 I2C 中断服务程序中读取 I2CSTA 并判断当前状态码是否为 C0H。若是, 则无需做任何操作, 从机将切换到不可寻址的从机模式并结束数据发送

16.6 从接收器模式

在从接收器模式下，当 I2C 的状态（I2CSTA 寄存器）发生转变时，中断请求位（I2CCON 寄存器中的 I2CIF 位）将被置 1，若 IEN1 寄存器中的 I2CIE 位和 EA 位被使能，I2C 将发出中断请求。从接收器模式下的 I2C 状态列表如下：

表 16-5 从接收器模式下的 I2C 状态列表

状态码	I2C 状态	应用软件操作				I2C 硬件的下一步动作	
		写/读 I2CDAT	写 I2CCON				
			STA	STO	I2CIF		AA
60H	自己作为从机的地址“SLA+W”已收到，且 ACK 已发送	无操作	X	0	0	0	数据字节将被接收，且无 ACK 返回
		无操作	X	0	0	1	数据字节将被接收，且有 ACK 返回
68H	多主机下通过仲裁，主机控制权丢失；自己作为从机的地址“SLA+W”已收到，且 ACK 已发送	无操作	X	0	0	0	数据字节将被接收，且无 ACK 返回
		无操作	X	0	0	1	数据字节将被接收，且有 ACK 返回
70H	通用广播呼叫地址(00H)已收到，且 ACK 已发送	无操作	X	0	0	0	数据字节将被接收，且无 ACK 返回
		无操作	X	0	0	1	数据字节将被接收，且有 ACK 返回
78H	多主机下通过仲裁，主机控制权丢失；通用广播呼叫地址(00H)已收到，且 ACK 已发送	无操作	X	0	0	0	数据字节将被接收，且无 ACK 返回
		无操作	X	0	0	1	数据字节将被接收，且有 ACK 返回
80H	此前已被寻址为从机；数据字节已接收，且 ACK 已发送	读数据字节	X	0	0	0	数据字节将被接收，且无 ACK 返回
		读数据字节	X	0	0	1	数据字节将被接收，且有 ACK 返回
88H	此前已被寻址为从机；数据字节已接收，且 ACK 未发送	读数据字节	0	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应
		读数据字节	0	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应
		读数据字节	1	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应；当总线空闲时发送起始位
		读数据字节	1	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应；当总线空闲时发送起始位

90H	此前已被通用广播呼叫寻址；数据字节已接收，且 ACK 已发送	读数据字节	X	0	0	0	数据字节将被接收，且无 ACK 返回
		读数据字节	X	0	0	1	数据字节将被接收，且有 ACK 返回
98H	此前已被通用广播呼叫寻址；数据字节已接收，且 ACK 未发送	读数据字节	0	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应
		读数据字节	0	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应
		读数据字节	1	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应；当总线空闲时发送起始位
		读数据字节	1	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应；当总线空闲时发送起始位
A0H	当被寻址为从接收器和从发送器模式时，停止位或重新起始位被接收到	无操作	0	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应
		无操作	0	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应
		无操作	1	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应；当总线空闲时发送起始位
		无操作	1	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应；当总线空闲时发送起始位

从接收器模式的典型软件配置方法：

- (1) 软件配置从机地址（I2CADR）。
- (2) 使能 IEN1 寄存器的 I2CIE 位与 EA 打开 I2C 中断。
- (3) 软件配置 I2CCON 的 I2CEN 位使能 I2C。
- (4) 若有主机发送的从机地址与该从机地址吻合，将会触发 I2C 中断。
- (5) 在 I2C 中断服务程序中读取 I2CDAT，此次 I2CDAT 中为自己的从机地址，并确认最低位为 0（R/W=0 主机向目标从机写数据）；同时读取 I2CSTA 并判断当前状态码是否为 60H。若是，则清除 I2CIF 标志位，等待主机发送数据。
- (6) 若数据已接收，将再次触发 I2C 中断。
- (7) 在 I2C 中断服务程序中读取 I2CSTA 并判断当前状态码是否为 80H。若是，则将接收的一个数据字节从 I2CDAT 中读出并清除 I2CIF 标志位。
- (8) 重复 (6) ~ (7)。
- (9) 若主机发送数据完成，从机接收到停止位或者重新起始位，将再次触发 I2C 中断。
- (10) 在 I2C 中断服务程序中读取 I2CSTA 并判断当前状态码是否为 A0H。若是，则无需做任何操作，从机将切换到不可寻址的从机模式并结束数据接收。
- (11) 若从机不想继续读取数据，需在当前状态码为 80H 的中断服务程序中将 I2CCON 的 AA 位清 0。然后从机将切换到不可寻址的从机模式并结束数据接收。

17. 运放/比较器

运放/比较器模块有四个通道，包括 1 个运算放大器（也可软件配置为比较器 0）和 3 个比较器（1/2/3），比较器 1/2/3 支持中断功能，中断入口地址分别为 0x009B、0x00A3、0x00AB，比较器 0 不支持中断功能，只支持标志位查询功能。四个通道均支持滤波功能，通过软件配置 CMPNFCNT 来选择滤波的宽度和滤波时钟。

表 17-1 与运放/比较器相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
F1h	INFCON	WDTRF	CMP3IF	CMP2IF	CMP1IF	CMP0IF	ADCIF	-	DPS
A8h	IEN0	EA	-	-	T2IE	T1IE	P1EA	T0IE	P0EA
B9h	IEN2	-	LVDIE	CMP3IE	CMP2IE	CMP1IE	ADCIE	-	WDTIE
BDh	CMPNFCNT	CMPNUM<4:0>					CMPSEL<2:0>		
86h	OPACON1	-	-	-	-	OPACRT<3:0>			
C5h	OPACON0	CMP00	OPAVREF	-	-	OPAFS<2:0>		OPAEN	
C6h	CMPIMOD	CMP3IMOD<1:0>		CMP2IMOD<1:0>		CMP1IMOD<1:0>		-	
CDh	CMP1CON	CMP10	CMP1HYS	CMP1FS<4:0>				CMP1EN	
CEh	CMP2CON	CMP20	CMP2HYS	CMP2FS<4:0>				CMP2EN	
CFh	CMP3CON	CMP30	CMP3HYS	CMP3FS<4:0>				CMP3EN	

寄存器定义 17-1 CMPNFCNT 控制字

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPNFCNT	CMPNUM<4:0>					CMPSEL<2:0>		
BDh	R/W					R/W		

Bit7-3 CMPNUM<4:0>:噪声滤波器数值，数值从 0~0x1f 总的滤波时间为(0~1f) × 采样时钟

Bit2-0 CMPSEL<2:0>:采样时钟选择

- 000:CPUCLK/1
- 001:CPUCLK/2
- 010:CPUCLK/4
- 011:CPUCLK/8
- 100:CPUCLK/16
- 101:CPUCLK/32
- 110:CPUCLK/64
- 111:CPUCLK/128

寄存器定义 17-2 CMPIMOD 比较器中断触发模式控制字

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPIMOD	CMP3IMOD<1:0>		CMP2IMOD<1:0>		CMP1IMOD<1:0>		-	
C6h	R/W		R/W		R/W		-	

Bit7-6 CMP3IMOD<1:0>:比较器 CMP3 中断触发模式

Bit5-4 CMP2IMOD<1:0>:比较器 CMP2 中断触发模式

Bit3-2 CMP1IMOD<1:0>:比较器 CMP1 中断触发模式

- 00:上升沿触发
- 01:下降沿触发
- 1x:边沿触发(上升沿和下降沿均触发)

寄存器定义 17-3 OPACON1 比较器 OPA 失调校准寄存器

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPACON1	-	-	-	-	OPACRT<3:0>			
86h	-	-	-	-	R/W			

Bit7-4 未使用

Bit3-0 OPACRT<3:0>:OPA 失调校准, 默认 1000 (软件软件配置)

寄存器定义 17-3 OPACON0

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPACON0	CMP0O	OPAVREF	-	-	OPAFS<2:0>			OPAEN
C5h	R	R/W	-	-	R/W			R/W

Bit7 CMP0O:比较器 0 输出结果

Bit6 OPAVREF:OPA 正端输入参考电压选择

0: 选择内部地

1: 选择内部参考电压 2V, 3V, 4V

Bit5-4 未使用

Bit3-1 OPAFS<2:0>: OPA 选择功能位

000:P0.5, P1.0, P0.4 都作为普通 I/O 引脚

001:作为比较器, P0.5 作为普通 I/O, 正端接内部参考, P1.0 作为比较器负输入引脚, P0.4 作为普通 I/O 引脚

010:作为比较器, P0.5 作为普通 I/O, 正端接内部参考, P1.0 作为比较器负输入引脚, P0.4 作为比较器输出引脚

011:作为比较器, P0.5 作为比较器正输入引脚, P1.0 作为比较器负输入引脚, P0.4 作为普通 I/O 引脚

100:作为比较器, P0.5 作为比较器正输入引脚, P1.0 作为比较器负输入引脚, P0.4 作为比较器输出引脚

101:作为 OP, P0.5, P1.0 作为普通 I/O 引脚, P0.4 作为 OP 的输出引脚, 可将内部参考通过 OP 输出

110:作为 OP, P0.5 作为普通 I/O 引脚, OP 正端接内部地, P1.0 作为 OP 负端输入引脚, P0.4 作为 OP 的输出引脚

111:作为 OP, P0.5 作为 OP 正端输入引脚, P1.0 作为 OP 负端输入引脚, P0.4 作为 OP 的输出引脚

Bit0 OPAEN:OPA 使能位 1: 开启式 OPA 模块; 0: 关闭 OPA 模块

寄存器定义 17-3 CMPnCON (n:1-3)

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPnCON	CMPnO	CMPnHYS	CMPnFS<4:0>				CMPnEN	
CD/CE/CF	R	R/W	R/W				R/W	

Bit7 CMPnO: 比较器 n 输出位 (n:1-3 为比较器组编号)

1: CnP > CnN

0: CnP < CnN

Bit6 CMPnHYS:CMPn 的迟滞效应使能位

1: 带 40mv 的迟滞

0: 无迟滞效果 (默认)

Bit5 CMPnFS<4>: CMPn 输出选择功能位, 默认为 0

为 1 时: n=1:P1.1 作为 CMP1 输出引脚, 同时输出到 CMP1O 寄存器

n=2:P3.5 作为 CMP2 输出引脚, 同时输出到 CMP2O 寄存器

n=3:P3.1 作为 CMP3 输出引脚, 同时输出到 CMP3O 寄存器

为 0 时:n=1:P1.1 为普通 IO 口, 比较输出的信号输出到 CMP1O 寄存器

n=2:P3.5 为普通 IO 口, 比较输出的信号输出到 CMP2O 寄存器

n=3:P3.1 为普通 IO 口, 比较输出的信号输出到 CMP3O 寄存器

Bit4 CMPnFS<3>: CMPn 正端输入选择功能位, 默认为 0

为 1 时: n=1:P1.2 作为 CMP1 正端输入引脚

n=2:P3.6 作为 CMP2 正端输入引脚

n=3:P0.1 作为 CMP3 正端输入引脚

为 0 时: n=1:P1.2 为普通 IO 口, CMP1 的正端输入处于悬空状态

n=2:P3.6 为普通 IO 口, CMP2 的正端输入处于悬空状态

n=3:P0.1 为普通 IO 口, CMP3 的正端输入处于悬空状态

Bit3 CMPnFS<2>:默认为 0, 不能写 1

Bit2 CMPnFS<1>: CMPn 一级负端输入选择功能位, 默认为 0

为 1 时: n=1:P1.3 作为 CMP1 负端输入引脚

n=2:P3.7 作为 CMP2 负端输入引脚

n=3:P0.2 作为 CMP3 负端输入引脚

为 0 时: n=1:P1.3 为普通 IO 口, CMP1 的负端输入由 CMP1FS<0>的软件配置确定

n=2:P3.7 为普通 IO 口, CMP2 的负端输入由 CMP2FS<0>的软件配置确定

n=3:P0.2 为普通 IO 口, CMP3 的负端输入由 CMP3FS<0>的软件配置确定

Bit1 CMPnFS<0>: CMPn 二级负端输入选择功能位, 默认为 0

为 1 时: n=1: CMP1EN 为 1, CMP1FS<2>, CMP1FS<1>为 0, CMP1FS<0>为 1 时,

比较器 1 负端接内部参考电压 (2V, 3V, 4V)

n=2: CMP2EN 为 1, CMP2FS<2>, CMP2FS<1>为 0, CMP2FS<0>为 1 时,

比较器 2 负端接内部参考电压 (2V, 3V, 4V)

n=3: CMP3EN 为 1, CMP3FS<2>, CMP3FS<1>为 0, CMP3FS<0>为 1 时,

比较器 3 负端接内部参考电压 (2V, 3V, 4V)

为 0 时: n=1: CMP1EN 为 1, CMP1FS<2>, CMP1FS<1>为 0, CMP1FS<0>为 0 时, 比较器 1 负端接内部地;

n=2: CMP2EN 为 1, CMP2FS<2>, CMP2FS<1>为 0, CMP2FS<0>为 0 时, 比较器 2 负端接内部地;

n=3: CMP3EN 为 1, CMP3FS<2>, CMP3FS<1>为 0, CMP3FS<0>为 0 时, 比较器 3 负端接内部地;

CMPnFS<2:0> (CMPnCON: Bit3-1)	CMPn 负端输入功能 (n:1-3 为比较器组编号)
1xx	禁止配置
01x	n=1:P1.3 作为 CMP1 负端输入引脚 n=2:P3.7 作为 CMP2 负端输入引脚 n=3:P0.2 作为 CMP3 负端输入引脚
000	n=1: P1.3 为普通 IO 口, CMP1EN 为 1, 比较器 1 负端接内部地 n=2: P3.7 为普通 IO 口, CMP2EN 为 1, 比较器 2 负端接内部地 n=3: P0.2 为普通 IO 口, CMP3EN 为 1, 比较器 3 负端接内部地
001	n=1: P1.3 为普通 IO 口, CMP1EN 为 1, 比较器 1 负端接内部参考电压 (2V, 3V, 4V) n=2: P3.7 为普通 IO 口, CMP2EN 为 1, 比较器 2 负端接内部参考电压 (2V, 3V, 4V) n=3: P0.2 为普通 IO 口, CMP3EN 为 1, 比较器 3 负端接内部参考电压 (2V, 3V, 4V)

Bit0 CMPnEN: CMPn 使能位 1: 开启 CMPn 模块; 0: 关闭 CMPn 模块

17.1 运算放大器

运算放大器通过软件配置寄存器 OPACON1, 达到校准功能。通过软件配置寄存器 OPACON0 的 OPAFS 和 OPAVREF, 选择 OPA 的工作模式和正端输入的参考电压, 也可软件配置 CMPNFCNT 来选择滤波的模式 (默认关闭滤波功能), 然后使能 OPACON0[0], 即 OPAEN 来开启运放。

运放通道也可以通过软件配置 OPAFS<2:0>, 用作比较器 0, 软件配置中断触发模式和滤波后, 使能 OPAEN 来使用比较器 0。比较器 0 通道不支持中断, 可以通过查询中断标志位来判断是否产生中断。

17.2 比较器

比较器 1/2/3 支持数字滤波功能和中断功能, 中断入口地址分别为 0x009B、0x00A3、0x00AB, 三个比较器可以选择 40mv 的迟滞 (默认关闭), 通过软件配置 CMPxCON 选择比较器的正负端以及输出端的连接方式, 同样通过软件配置 CMPIMOD 寄存器选择上升沿触发、下降沿触发、和边沿触发 (上升沿或下降沿) 三种中断触发的方式。当使能中断功能时, 进入中断子程序后, 硬件自动清零中断标志位 CMPxIF。当关闭中断功能时, 中断标志位 CMPxIF 不自动清零。

比较器 1/2/3 具有校准功能, 上电后第一次使用比较器, 硬件将自动完成 100us 左右的硬件自动校准功能, 之后重新启动比较器都以该校正结果工作, 除非断电, 否则不会再次进入硬件自动校准功能。三个通道的校准功能独立分开, 校准期间不允许使用比较器输入。

18. 液晶显示(LCD)驱动模块

J8M302A 所有 IO 口（除了 P00 烧录口）通过软件都可以作为 LCD 的 COM 或 SEG，每个 IO 除了正常 IO 的功能外，还可以输出两组可选电压：1/3VDD，2/3VDD 和 1/2VDD。用户可根据使用情况，随意选择任意 IO（除了 P00 烧录口）作为 LCD 驱动的 COM 或者 SEG。

注意：棕色背景寄存器位于 SFR 列表第 0 页，需要软件配置 PSW[5]=0 进行访问。

表 18-1 与 LCD 相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8Bh	P0LCDVO	-	-	P05VO	P04VO	P03VO	P02VO	P01VO	-
8Ch	P1LCDVO	P17VO	P16VO	P15VO	P14VO	P13VO	P12VO	P11VO	P10VO
F6h	P2LCDVO	-	-	-	-	P23VO	P22VO	P21VO	P20VO
F7h	P3LCDVO	P37VO	P36VO	P35VO	P34VO	P33VO	P32VO	P31VO	P30VO
EEh	LCDCON	-	-	-	-	VOIRS<1:0>		-	VOS

寄存器定义 18-1PnLCDVO(n=0-3 分别对应 P0-P3 口，未使用的位，写无效，读为 0) LCD 电压输出使能寄存器

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PnLCDVO	Pn7VO	Pn6VO	Pn5VO	Pn4VO	Pn3VO	Pn2VO	Pn1VO	Pn0VO
8B/8C/F6/E7	R/W							

Bit7Pn7VO: Pn7 口 LCD 电压输出使能位

1: 输出 Pn7 口的 LCD 输出电压控制

0: 关闭 Pn7 口的 LCD 输出电压控制

...

Bit0 Pn0VO: Pn0 口 LCD 电压输出使能位

1: 输出 Pn0 口的 LCD 输出电压控制

0: 关闭 Pn0 口的 LCD 输出电压控制

寄存器定义 18-2LCDCON

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON	-	-	-	-	VOIRS<1:0>		-	VOS
EEh	-	-	-	-	R/W		-	R/W

Bit7-4 未使用

Bit3-2VOIRS<1:0>: LCD 电压输出口分压电阻选择（根据 LCD 屏大小选择合适的驱动）

00: 关闭内部分压电阻

01: 设定内部最小分压电阻约为 25K

10: 设定内部最小分压电阻约为 50K

11: 设定内部最小分压电阻约为 100K

Bit1 未使用

Bit0 VOS: LCD 电压输出口电压输出选择

0: 当 PxyVO 为 1 时，设定 PxyVO 口输出电压为 1/2VDD

1: 当 PxyVO 为 1，Px.y 为 0 时，设定 PxyVO 口输出电压为 2/3VDD

当 PxyVO 为 1，Px.y 为 1 时，设定 PxyVO 口输出电压为 1/3VDD

注：Px.y 为 Pxy 口的输出寄存器 Px 的 bit y 的值

LCD 的 COM/SEG 的驱动信号为交流模拟信号，LCD 像素点的显示对比度取决于此显示点上的 COM 与 SEG 电压 (Vcom-Vseg/ Vseg-Vcom)，当此电压差大于 LCD 的饱和电压就能打开此点的显示，小于 LCD

的饱和电压就能关闭此 LCD 点的显示。这需要每个 COM/SEG 在一个扫描周期内根据用户需要在选通和非选通电压之间切换。

BJ8M302A 工作在非选通电压 ($1/2VDD@1/2Bias$; $1/3VDD$ 、 $2/3VDD@1/3Bias$) 时需将 IO 的状态设置为高阻输入, 工作在选通电压 (VDD、GND) 时需将 IO 的状态设置为强推挽输出。当 LCD 从选通电压切换为非选通电压时, 需要先将 IO 的状态设置为高阻输入 ($PxLCDV0=1$), $1/3Bias$ 需要再设置 IO 的输出值, 最后软件配置 $PxLCDV0$ 将 IO 设置为 LCD 输出; 当 LCD 从非选通电压切换为选通电压时, 需要先软件配置 $PxLCDV0$ 将 IO 设置为 GPIO 输出, 要再设置 IO 的输出值, 最后将 IO 设置为强推挽输出。

18.1 LCD1/2Bias 1/4Duty 应用

图 18-1 1/2Bias LCD 应用时 COM 的波形图:

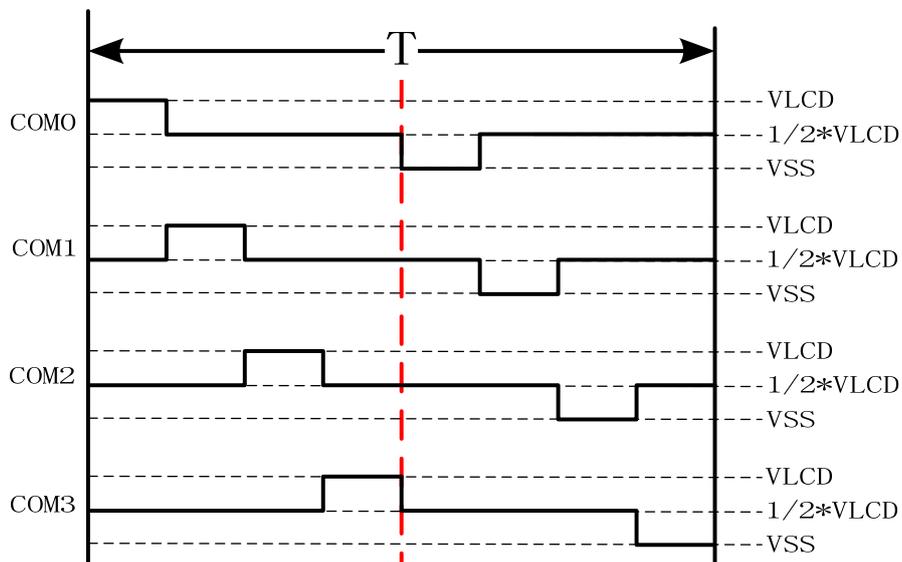


表 18-2 1/4duty1/2Bias LCD COM0~COM3 一轮扫描的 COM&SEG 电平和显示关系:

COM0	COM1	COM2	COM3	SEGN点亮	SEGn+1不亮
VDD	1/2VDD	1/2VDD	1/2VDD	VSS	VDD
VSS	1/2VDD	1/2VDD	1/2VDD	VDD	VSS
1/2VDD	VDD	1/2VDD	1/2VDD	VSS	VDD
1/2VDD	VSS	1/2VDD	1/2VDD	VDD	VSS
1/2VDD	1/2VDD	VDD	1/2VDD	VSS	VDD
1/2VDD	1/2VDD	VSS	1/2VDD	VDD	VSS
1/2VDD	1/2VDD	1/2VDD	VDD	VSS	VDD
1/2VDD	1/2VDD	1/2VDD	VSS	VDD	VSS

以 1/2Bias 为例, 用 P10 作为其中一个 COM 输出 3 级电压的设置步骤如下:

①置 $LCDCON=0x04$; 选择 1/2Bias 输出, 设定内部最小的分压电阻;

- ②输出 VDD: 设置 P1LCDV0=0x00; 再设置 P1=0x01; 最后设置 P10E=0x01;
- ③输出 1/2VDD: 设置 P10E=0x00; 再设置 P1LCDV0=0x01;
- ④输出 GND: 设置 P1LCDV0=0x00; 再设置 P1=0x00; 最后设置 P10E=0x01;

18.2 LCD1/3Bias 1/4Duty 应用

图 18-2 1/3Bias LCD 选通和非选通电压波形:

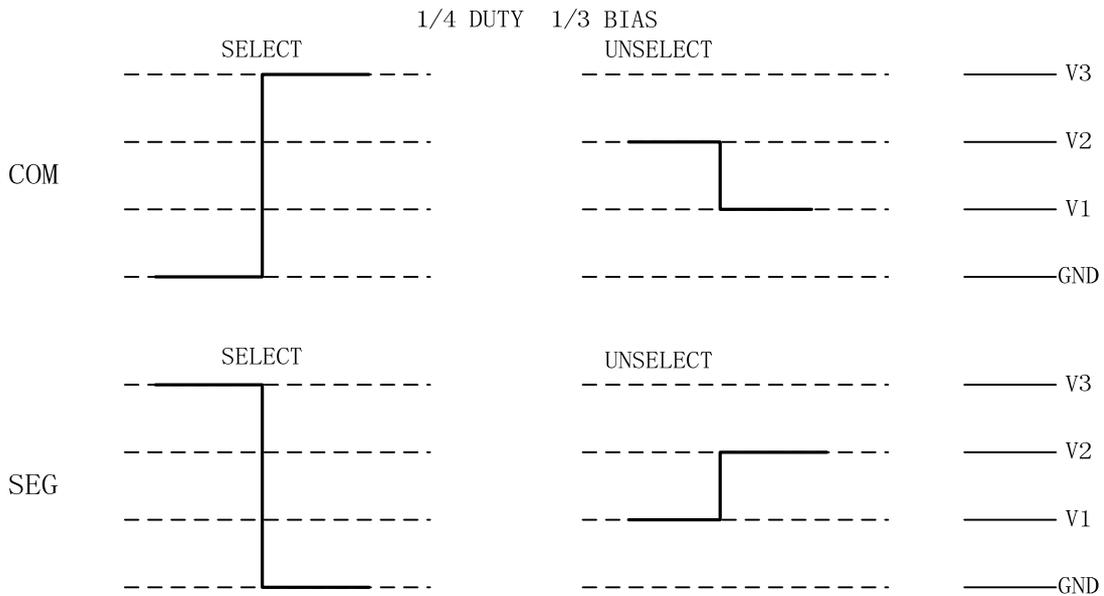


图 18-3 1/3Bias LCD 应用中 COM 和 SEG 的波形图

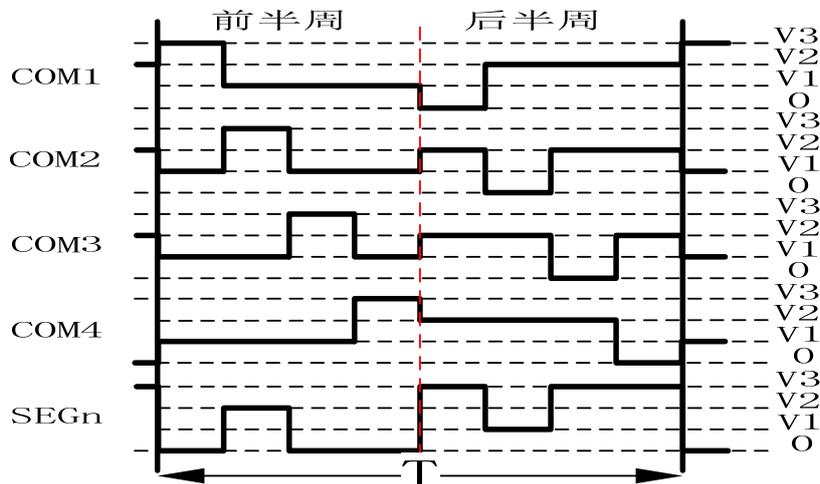


表 18-3 1/3Bias 直流电压电平选择关系表

项 目			1/3Bias LCD
直流分压电平			VDD V2=2/3VDD V1=1/3VDD GND
前半	COM	选通电平	VDD

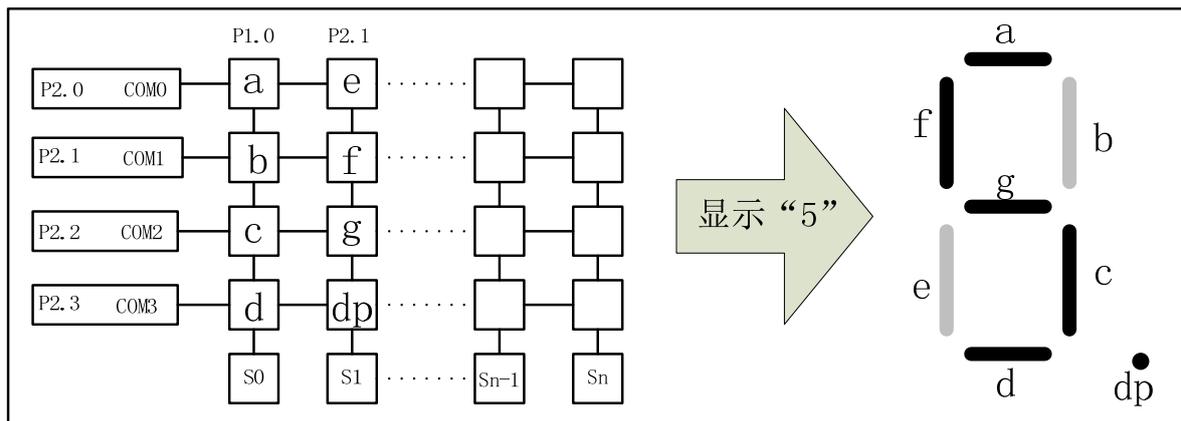
扫描周期	SEG	非选通电平	V1
		选通电平	GND
		非选通电平	V2
后半扫描周期	COM	选通电平	GND
		非选通电平	V2
	SEG	选通电平	VDD
		非选通电平	V1

以 1/3Bias 为例，P10 输出 4 级电压的设置步骤如下：

- ①置 LCDCON=0x05；选择 1/3Bias 输出，设定内部最小的分压电阻；
- ②输出 VDD：设置 P1LCDV0=0x00；再设置 P1=0x01；最后设置 P10E=0x01；
- ③输出 1/3VDD：设置 P10E=0x00；再设置 P1=0x01；最后设置 P1LCDV0=0x01；
- ④输出 GND：设置 P1LCDV0=0x00；再设置 P1=0x00；最后设置 P10E=0x01；
- ⑤输出 2/3VDD：设置 P10E=0x00；再设置 P1=0x00；最后设置 P1LCDV0=0x01；

注意：IO 输出的 1/2VDD 和 1/3VDD、2/3VDD 都是 IC 内部 VDD 经过分压输出（分压电阻值为用户设定 VOIRS），如果 IO 有复用其它功能，请确保 IO 其他功能不会将此电压拉偏。

图 18-4 物理连接显示图：



19. 实时时钟(RTCC)模块

BJ8M302A 的 RTCC(Real-Time Clock and Calendar) 实时时钟模块是由外置 32768 晶振与内部 T0 模块组合而成(内部的 RTCC 模块与 Timer 0 共用硬件电路, 即使用 RTCC 时, T0 不能作为定时计数使用。同理, T0 作为定时计数器时, 就没有 RTCC 功能)。使用时, 程序配置 SYSCKM[2:0]选项, 使其芯片工作为 RTCC 模式后, 软件配置 TOPSC<1:0>来选择中断时间间隔(注意: TOPSC[2]在 RTCC 模式为无效位。因此仅四种模式), 开启中断和使能 T0 即可。在达到设置的定时数值时, 能够唤醒 IDLE/STOP 模式。

图 19-1 TRCC 模块框图

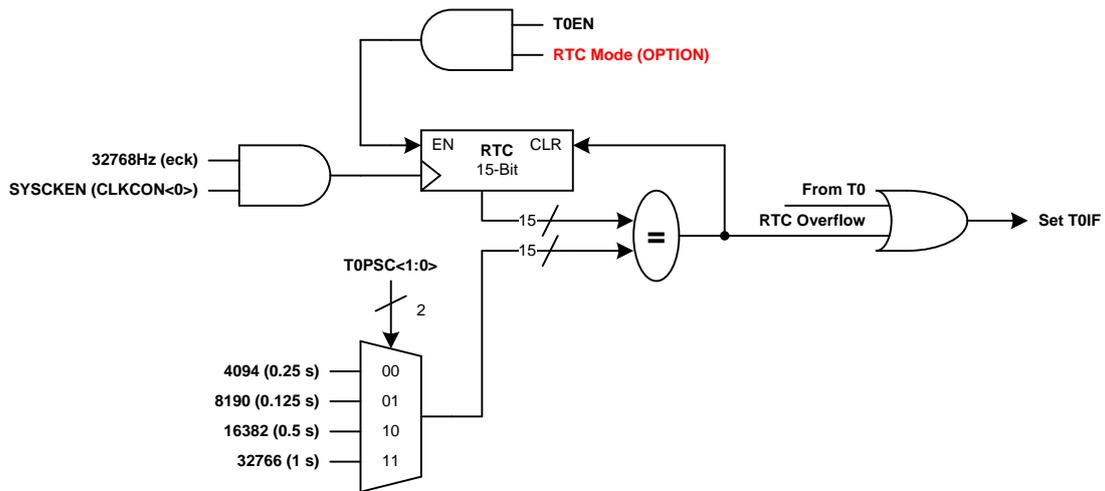


表 19-1 与 RTCC 相关的寄存器汇总

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
88h	T0CON	T0CLR	T0M	T0IF	T0EG	TOPSC2	TOPSC1	TOPSC0	T0EN
E8h	CLKCON	DEBUGEN	SYSCKOEN	-	TRIMEN	RUNWCK	WCKEN	TCKEN	SYSCKEN

T0CON 详细定义请参考 10-1 寄存器定义

CLKCON 详细定义请参考 寄存器定义 4-1 时钟系统控制寄存器

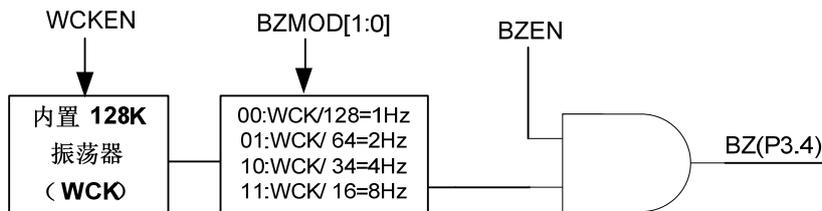
软件配置方法:

- (1) 需要保证 32.768KHz 的 ECK 有效。SYSCKEN=1;
- (2) 先设置 RTCC 需要定时的时间, 即软件配置寄存器 T0CON 的 TOPSC[1:0]。
- (3) 使能 RTCC: 即寄存器 T0CON 的 T0EN=1, 并且代码选项 OPTION 的 SYSCKM[2:0]选项设置 RTCC 模式;
- (4) 硬件会开始计时, 达到设置的时间时, 会唤醒 IDLE 或者 STOP 模式。

20. 蜂鸣器(BEEZ)

BJ8M302A 内部集成了 BEEZER 模块。时钟来源与看门狗 WCK 同源。通过软件配置时钟分频器后，有软件灵活的控制的从 BZ 引脚输出即可。

图 20-1 蜂鸣器工作框图



寄存器定义 20-1 BZCON 蜂鸣器软件配置寄存器

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BZCON	-	-	-	-	-	BZMOD	BZEN	
8Fh	-	-	-	-	-	R/W	R/W	

Bit7-3 未使用

Bit2-1 BZMOD: Beez 输出频率选择

- 00: beez 输出为 1KHz 的方波
- 01: beez 输出为 2KHz 的方波
- 10: beez 输出为 4KHz 的方波
- 11: beez 输出为 8KHz 的方波

Bit0 BZEN: Beez 输出使能位

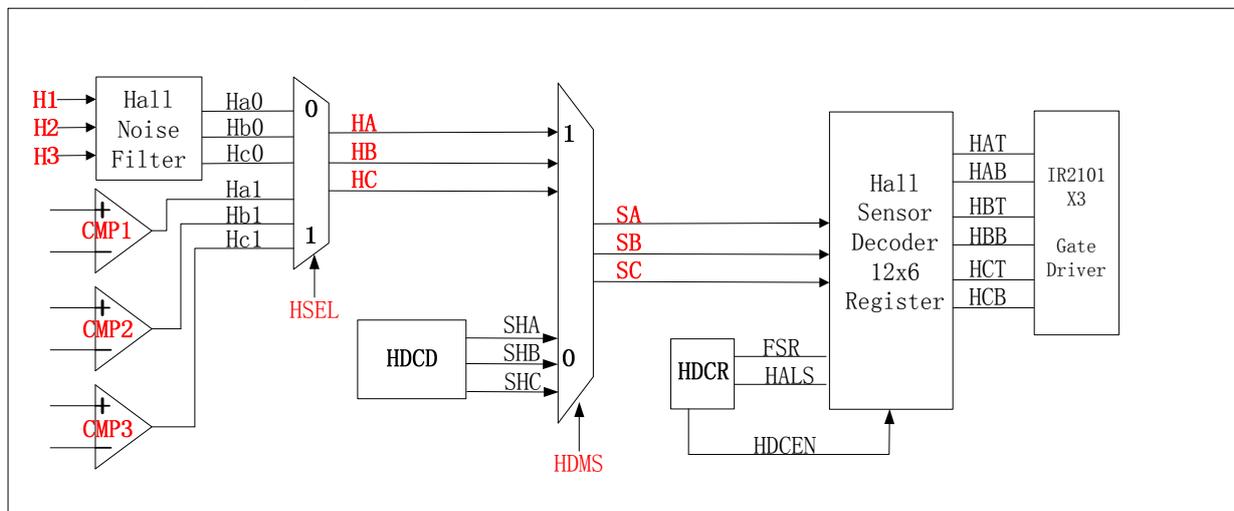
- 1: 使能 BEEZ 的输出。
- 0: 不使能 BEEZ 的输出。

21. 霍尔传感器模块

BJ8M302A 包含一个完全集成的霍尔传感器解码电路连接到霍尔传感器用于 BLDC 马达的方向和转速控制。当打开霍尔模块(HDCEN=1)，并且选择从外部获取霍尔信号(HSEL=0)时，霍尔信号分别依次从 P3.5/P3.6/P3.7 输入，内部对应信号依次为 H1/H2/H3。芯片集成的霍尔传感器解码电路：支持 6 步骤通信的马达方向控制的方法，通过 HDCD/ HDCR 寄存器控制马达的转向前、转向后、刹车、空转。霍尔传感器解码电路的输入可以选择由霍尔传感器解码 (HA/HB/HC) 或 SHA/SHB/SHC。

(注意：PWM 控制、死区时间条件、刹车等模块控制方法请参考 12.3T2PWM 章节描述。)

图 21-1 BLDC 马达控制框图



21.1 霍尔传感器

BJ8M302A 包含一个完全集成的霍尔传感器解码电路连接到霍尔传感器用于 BLDC 马达的方向和转速控制。

注意：棕色背景寄存器位于 SFR 列表第 0 页，需要软件配置 PSW[5]=0 进行访问。

表 21-1 与霍尔传感器相关寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
8Dh	HDCD	-						SHC	SHB	SHA
8Eh	HDCR	HSEL	-	PWMS	MPWMS	HALS	HDMS	FRS	HDCEN	
9Fh	HDFCON	HCHKNUM<4:0>					HFRSEL<2:0>			

寄存器 21-1 HDCD 霍尔传感器选择寄存器

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
HDCD	-					SHC	SHB	SHA
8Dh	-					R/W	R/W	R/W

Bit7-3 未使用

Bit2SHC: S/W 霍尔 C

Bit1 SHB: S/W 霍尔 B
Bit0 SHA: S/W 霍尔 A

寄存器定义 21-2 HDCR 霍尔传感器控制寄存器

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
HDCR	HSEL	-	PWMS	MPWMS	HALS	HDMS	FRS	HDCEN
8Eh	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit7 HSEL: HA/HB/HC 来源选择

1: CMP1/CMP2/CMP3 输出

0: H1/H2/H3

Bit6 未使用

Bit5 PWMS:

1: 底端 PWM 输出选择晶体管对的下臂 (GAB/GBB/GCB)

0: 顶端 PWM 输出选择晶体管对的上臂 (GAT/GBT/GCT)

Bit4 MPWMS: 互补输出使能位

1: 非互补输出

0: 互补输出

Bit3 HALS: 霍尔传感器解码模式选择

1: 霍尔传感器 120 度软件配置

0: 霍尔传感器 60 度软件配置

Bit2 HDMS: 霍尔传感器解码模式选择

1: 霍尔传感器模式

0: S/W 模式

Bit1 FRS: 马达向前转/ 向后转选择

1: 向后转

0: 向前转

Bit0 HDCEN: 霍尔传感器解码使能

1: 使能

0: 除能

寄存器定义 21-3 HDNFCON 霍尔噪声滤波控制寄存器

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
HDNFCON	HCHKNUM<4:0>					HFRSEL<2:0>		
9Fh	R/W					R/W		

Bit7-3 HCHKNUM[4:0]: 噪声滤波器数值, 数值从 0~0x1f 总的滤波时间为 (0~1f) × 采样时钟

Bit2-0 HFRSEL[2:0]: 噪声采样时钟分频选择

000: CPUCLK/1 (默认)

001: CPUCLK/2

010: CPUCLK/4

011: CPUCLK/8

100: CPUCLK/16

101: CPUCLK/32

110: CPUCLK/64

111: CPUCLK/128

21.2 马达控制驱动信号

软件可以通过 HDCR 和 HDCD 寄存器来配置马达的控制驱动方向。302A 芯片内部会根据 SA/SB/SC 信号和 HDCR 的配置值, 产生对应的解码值, 文中用 HDCTx 表示。PWM 就是根据 HDCTx 的码值来产生对应的 PWM 波形, 从而控制马达的转向。其中 HDCT0~HDCT5 为马达向前转表; HDCT6~HDCT11 为马达向后

转表。

霍尔传感器解码真值表如下：

	60 度			120 度			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SA	SB	SC	SA	SB	SC	HAT	HAB	HBT	HBB	HCT	HCB
向前转	1	0	0	1	0	0	HDCT0[5:0]					
HDCEN=1	1	1	0	1	1	0	HDCT1[5:0]					
FRS=0	1	1	1	0	1	0	HDCT2[5:0]					
BRKE=0	0	1	1	0	1	1	HDCT3[5:0]					
	0	0	1	0	0	1	HDCT4[5:0]					
	0	0	0	1	0	1	HDCT5[5:0]					

	60 度			120 度			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SA	SB	SC	SA	SB	SC	HAT	HAB	HBT	HBB	HCT	HCB
向后转	1	0	0	1	0	0	HDCT6[5:0]					
HDCEN=1	1	1	0	1	1	0	HDCT7[5:0]					
FRS=1	1	1	1	0	1	0	HDCT8[5:0]					
BRKE=0	0	1	1	0	1	1	HDCT9[5:0]					
	0	0	1	0	0	1	HDCT10[5:0]					
	0	0	0	1	0	1	HDCT11[5:0]					

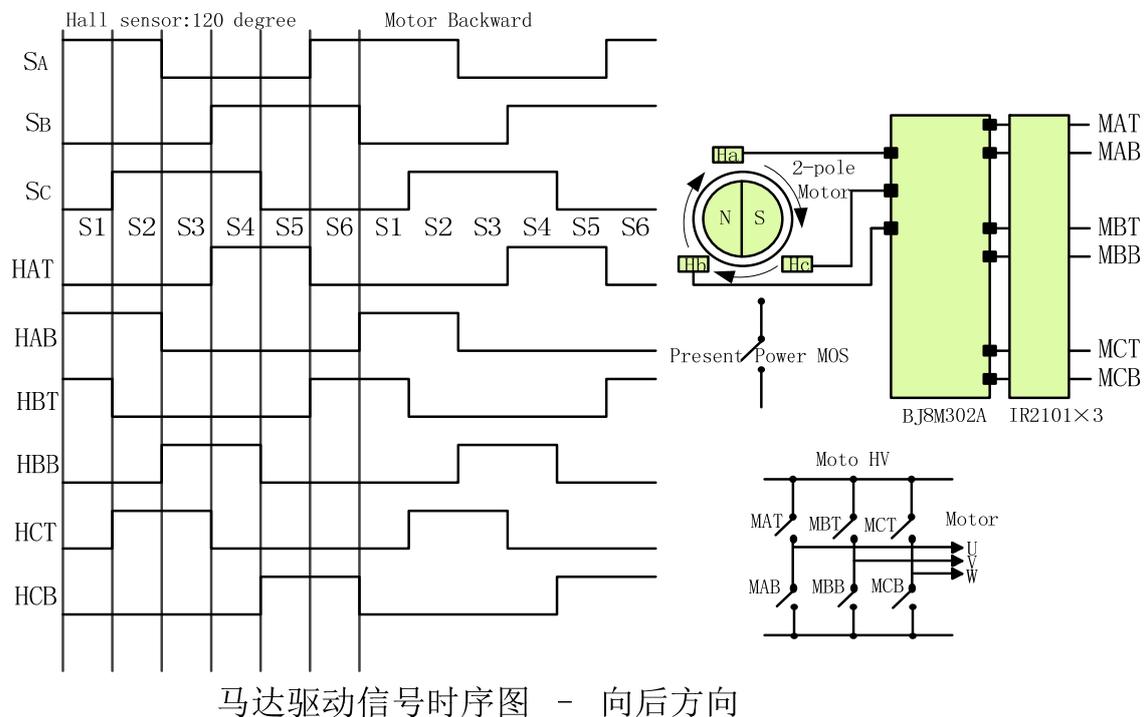
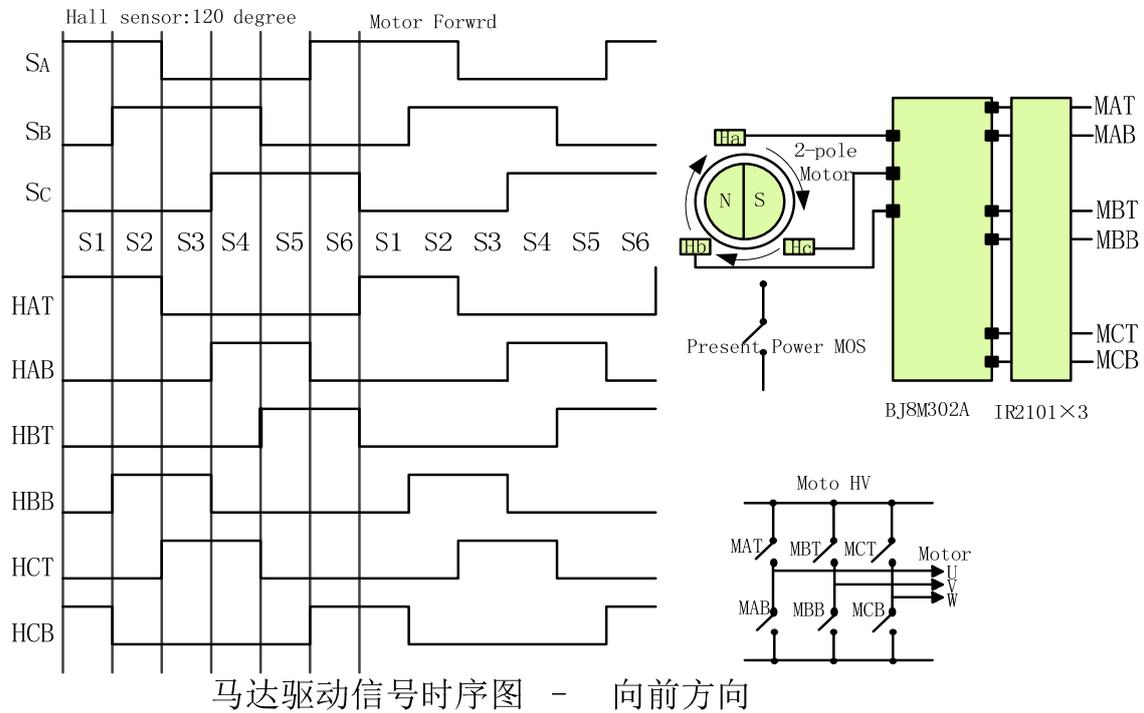
刹车 真值表	BKF=1 HDCEN=X FRS=X	60 度			120 度			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		SA	SB	SC	SA	SB	SC	HAT	HAB	HBT	HBB	HCT	HCB
		V	V	V	V	V	V	V	0	1	0	1	0

霍尔 解码器 除能	HDCEN=0 空转	60 度			120 度			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		SA	SB	SC	SA	SB	SC	HAT	HAB	HBT	HBB	HCT	HCB
		V	V	V	V	V	V	V	0	0	0	0	0

霍尔 传感器 出错	HDCEN=X	60 度			120 度			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		SA	SB	SC	SA	SB	SC	HAT	HAB	HBT	HBB	HCT	HCB
		1	0	1	1	1	1	1	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	

上述真值表中的数据和它们怎么联系实际马达驱动信号如下面时序图所示。以典型 6 步骤通信为例，

下图是马达向前转/ 马达向后转时序图。



21.2 解码后的 PWM 波形

根据上面描述，根据送进来的 SA/SB/SC 和 HDCR 的配置，我们可以得到控制马达转的解码真值表 HDCTx (HAT/HAB/HBT/HBB/HCT/HCB)。然后 Timer2 的 3 路互补输出的 PWM 会根据 HDCTx 的码值，输出对应的 PWM 波形。

软件人员可以通过配置 HDCR 的 PWMS/MPWMS 位决定马达调速的控制方式。

当 PWMS=0 时，底端 PWM 输出选择晶体管对的下臂 (GAB/GBB/GCB)
 当 PWMS=1 时，顶端 PWM 输出选择晶体管对的上臂 (GAT/GBT/GCT)
 当 MPWMS=0 时，PWM 有一个互补式输出
 当 MPWMS=1 时，PWM 有一个非互补式输出

PWM H/W 解码模式

互补式控制：MPWMS=0

PWMS=0	HAT	HAB	AT0	AB0	PWMS=1	HAT	HAB	AT0	AB0	
	0	0	0	0		0	0	0	0	0
	0	1	PWMN	PWM		0	1	0	1	
	1	0	1	0		1	0	PWM	PWMN	
	1	1	0	0		1	1	0	0	

PWMS=0	HBT	HBB	BT0	BB0	PWMS=1	HBT	HBB	BT0	BB0	
	0	0	0	0		0	0	0	0	0
	0	1	PWMN	PWM		0	1	0	1	
	1	0	1	0		1	0	PWM	PWMN	
	1	1	0	0		1	1	0	0	

PWMS=0	HCT	HCB	CT0	CB0	PWMS=1	HCT	HCB	CT0	CB0	
	0	0	0	0		0	0	0	0	0
	0	1	PWMN	PWM		0	1	0	1	
	1	0	1	0		1	0	PWM	PWMN	
	1	1	0	0		1	1	0	0	

非互补式控制：MPWMS=1

PWMS=0	HAT	HAB	AT0	AB0	PWMS=1	HAT	HAB	AT0	AB0	
	0	0	0	0		0	0	0	0	0
	0	1	0	PWM		0	1	0	1	
	1	0	1	0		1	0	PWM	0	
	1	1	0	0		1	1	0	0	

PWMS=0	HBT	HBB	BT0	BB0	PWMS=1	HBT	HBB	BT0	BB0	
	0	0	0	0		0	0	0	0	0
	0	1	0	PWM		0	1	0	1	
	1	0	1	0		1	0	PWM	0	
	1	1	0	0		1	1	0	0	

PWMS=0	HCT	HCB	CT0	CB0	PWMS=1	HCT	HCB	CT0	CB0	
	0	0	0	0		0	0	0	0	0
	0	1	0	PWM		0	1	0	1	
	1	0	1	0		1	0	PWM	0	

	1	1	0	0		1	1	0	0
--	---	---	---	---	--	---	---	---	---

用户使用步骤:

根据需要软件配置三路 PWM:

- 1) 设置 T2PSC[2:0], 软件配置 PWM 时钟。
- 2) 设置 T2OVR, 设置 PWM 周期。
- 3) 设置 T2Dx, 设置 PWMx 有效电平宽度。
- 4) 设置 PWMxS, 设置 PWMx 输出有效电平。
- 5) 设置寄存器标志位: PWMxEN, 使能 PWMx 输出。
- 6) 设置寄存器标志位: T2EN, 使能定时器。

根据需要软件配置霍尔解码寄存器

- 1) 根据 120 度电机或者 60 度电机设置 HDCR 寄存器的 HDLS (0:60 度 1:120 度)
- 2) 设置 HDCR 寄存器 FRS 为正转还是反转 (0:向前转 1:向后转)
- 3) 使能是否使用噪声滤波器
- 4) 使能霍尔传感器 (HDCR 寄存器 HDCEN=1)

例 21-1 霍尔传感器软件配置参考代码

```

void main (void)
{
    int count;
    T2OVRH= 0x00;
    T2OVR= 0x0a;
    //PWM1CONFIG
    T2D1H= 0x00;
    T2D1L= 0x02; //can still have interrupt PWM1
    //PWM2 CONFIG
    T2D2H= 0x00;
    T2D2L= 0x04; //can still have interrupt PWM2
    //PWM3 CONFIG
    T2D3H= 0x00;
    T2D3L= 0x06;
    PWMCON1 |=0x0f; //disable interrupt
    T2CON3 |=0x0f; //enable output
    T2CON4 |=0x0f; //enable output
    EA = 0;
    T2EN= 1;
    //*****
    // 120 度, 正转 FRS=0
    //*****
    HDCR = 0x2d; // pwms=1 mpwms=0 //互补下臂主波
    ...
}

```

22. 快充模块(QC)

BJ8M302A 集成了快充电路中的 VBUS 检测、苹果输出自动识别、三星输出自动识别和安卓系统输出自动识别电路，并且每个都是独立的模块。使用 QC 模块需要配合快充外围电路。QC 模块通过灵活的软件配置，可以灵活管理 USB D-/D- 数据线上的电压、充电放电自动识别等功能。

- 注意：1. 黄色背景寄存器均位于 SFR 列表第 1 页，需要设置 PSW[5]=1 进行访问。
2. 使用 QC 模块时，必须选择内部参考电压 VBG=2V，即配置 VREFSEL[1:0]=01。

表 22-1 与快充相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8Bh	QCCON0	QCEN	DP0REN	DN0REN	DP0DNS	FBEN	VLS0<2:0>		
8Ch	QCCON1	-	PD1REN	DN1REN	-	VLS1P<1:0>		VLS1N<1:0>	
F6h	IDO	IDO<7:0>							
F7h	IUP	IUP<7:0>							

寄存器定义 22-1QCEN0

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
QCCON0	QCEN	DP0REN	DN0REN	DP0DNS	FBEN	VLS0<2:0>		
8Bh	R/W	R/W	R/W	R/W	R/W	R/W		

Bit7QCEN: QC 模块使能位

- 1: 开启 QC 模块功能
- 0: 关闭 QC 模块。关闭后，该模块不会对其它电路产生任何负面影响

Bit6 DP0REN: DP0 (D0+) 下拉电阻使能，高电平有效。

Bit5DN0REN: DN0 (D0-) 下拉电阻使能，高电平有效。

Bit4 DP0DNS: DP0 (D0+)、DN0 (D0-) 互短使能，高电平有效。

Bit3 FBEN: FB 功能使能，高电平有效。

- 1: P31 为 QC 的 FB 口，优先级比较器高
- 0: P31 为 IO 口

Bit2-0VLS0<2:0>: D0 电压输出模式选择控制，详情请见表 20-4 和表 20-5

寄存器定义 22-1QCEN1

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
QCCON1	-	PD1REN	DN1REN	-	VLS1P<1:0>		VLS1N<1:0>	
8Ch	-	R/W	R/W	-	R/W		R/W	

Bit7 未使用

Bit6PD1REN:DP0 (D0+) 下拉电阻使能，高电平有效。

Bit5DN1REN:DN0 (D0-) 下拉电阻使能，高电平有效。

Bit4 使用

Bit3-2VLS1P<1:0>:DP1 (D1+) 电压输出模式选择控制。

Bit1-0VLS1N<1:0>:DP1 (D1-) 电压输出模式选择控制。

寄存器 ID0 (F6h) 可读写，是 PMOS 电流镜组选择控制。

寄存器 IUP (F7h) 可读写，是 NMOS 电流镜组选择控制。

图 22-1 QC3.0 模块框图

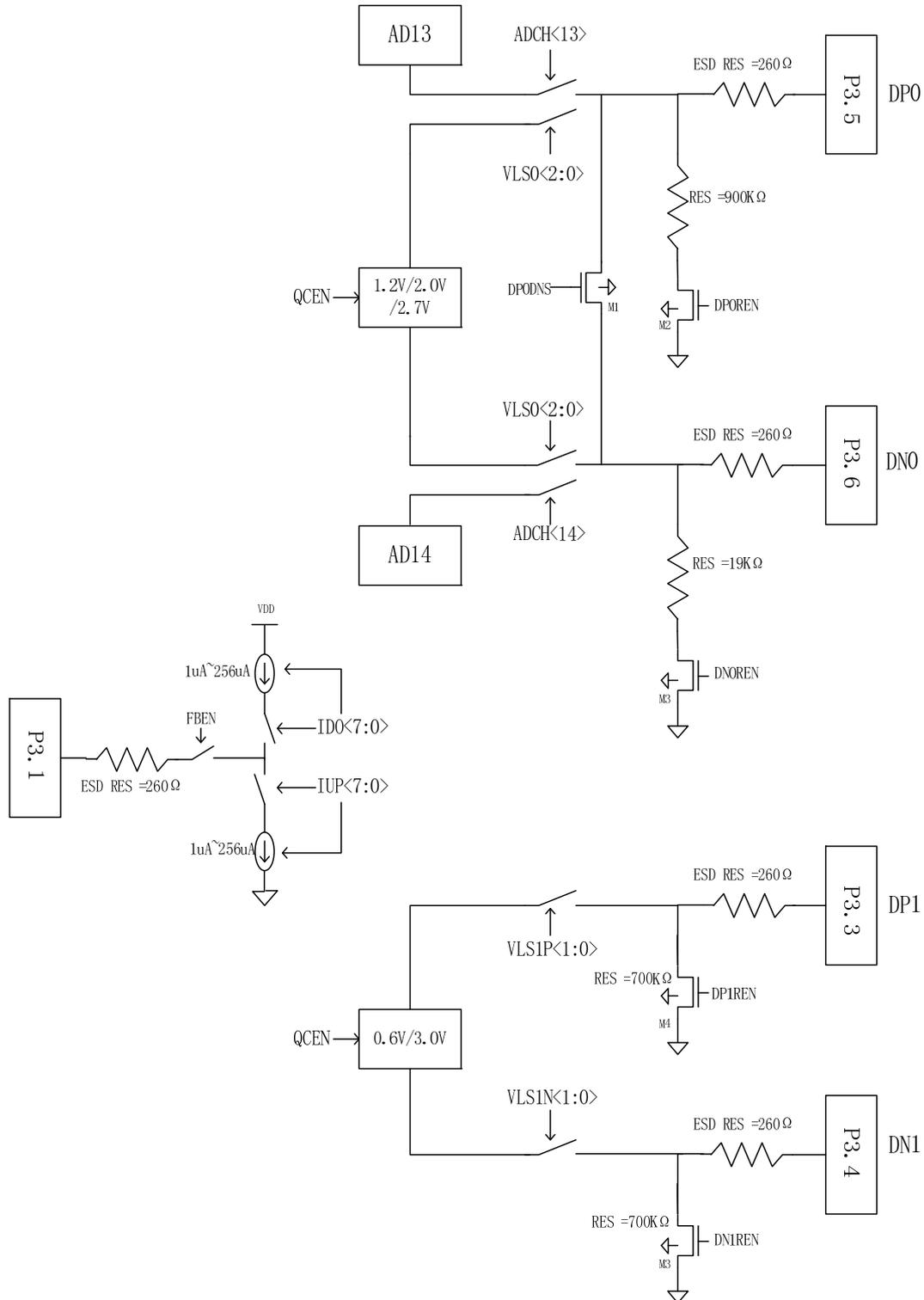
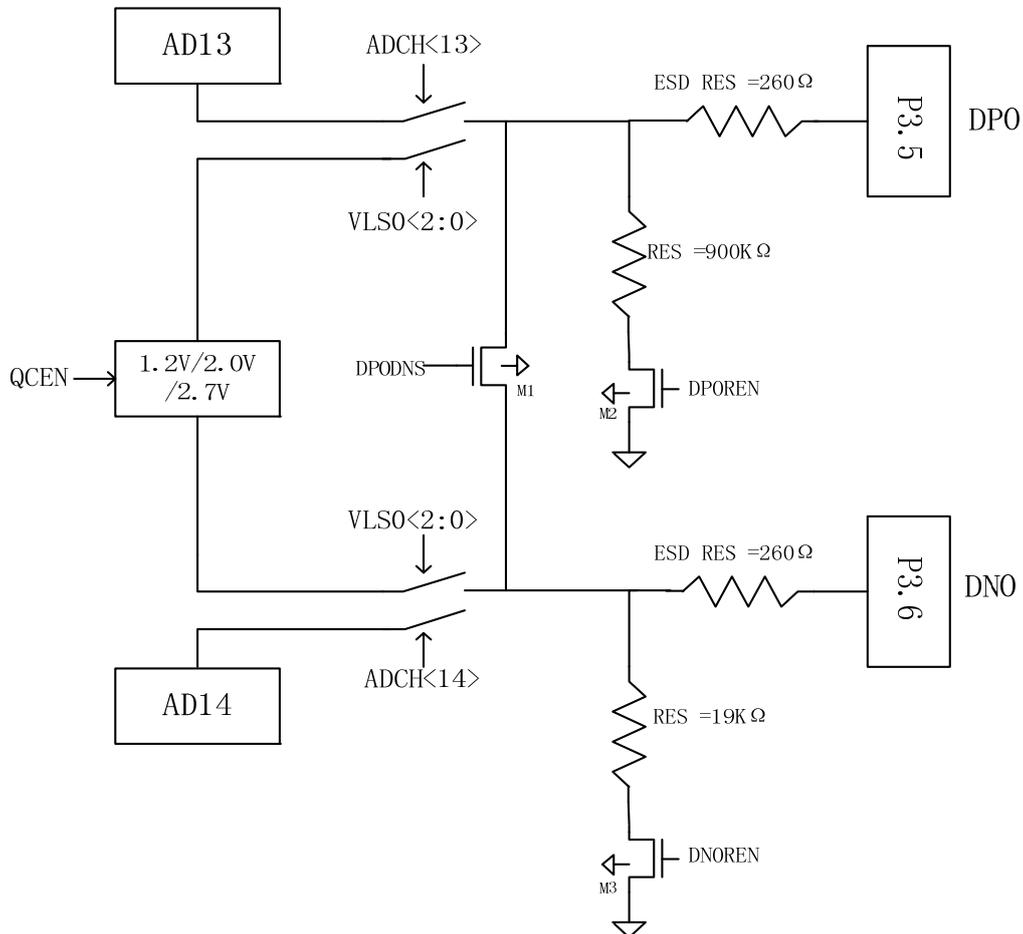


图 22-1 QC3.0 模块框图 (续)



DPO (D0+) 引脚集成 900k 下拉电阻，采用 NMOS 实现，由 DPREN 为高时使能。详情见 表 20-2。
DNO (D0-) 引脚集成 19.53k 下拉电阻，采用 poly 电阻串联控制管 M2 实现，由 DNREN 为高时使能。详情见 表 20-2

DPO (D0+) 与 DNO (D0-) 之间的互短开关 M1 由 trans 实现，由 DPODNS 为高时使能。其 Ron 的详情见 表 20-2

D+ Leakage Resistance	RDAT(LKG)	VBP=3.1-6.3V VD+=0.5-3.6V Switch N1 is OFF	300	900	1500	kΩ
D- Pull-Down Resistance	RDM(DWN)		14.25	19.53	24.5	kΩ
Switch N1 On-Resistance	RDS(ON)N1	VBP=4.3V VD+≤3.6V IDRAIN=200uA		20	40	Ω

为了确保 DP0/DN0 在 AD 输入功能时，相互的串扰较小，M1 由两级 trans 串联实现。在 QCEN=0 时 trans 断开，中间 NMOS 下拉到地。

FB 引脚功能集成 PMOS 电流镜组 (IDO<7:0>控制，高电平使能) 和 NMOS 电流镜组 (IUP<7:0>控制，高电平使能)。以 1uA 的 LSB 步进，全 0 时为 0uA，全 1 时为 255uA。

图 22-2FB 引脚接线等效图

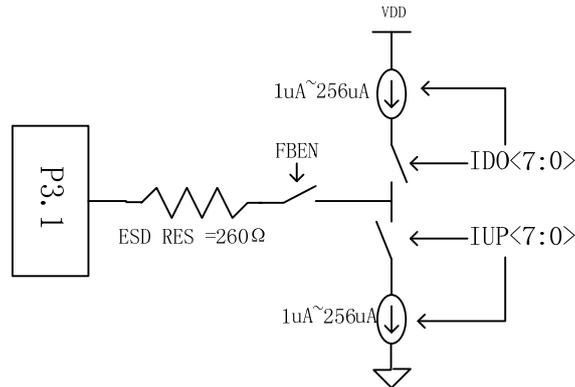


表 22-3 IDO&IUP 软件配置电流表

IDO<7>/ IUP<7>	128uA
IDO<6>/ IUP<6>	64uA
IDO<5>/ IUP<5>	32uA
IDO<4>/ IUP<4>	16uA
IDO<3>/ IUP<3>	8uA
IDO<2>/ IUP<2>	4uA
IDO<1>/ IUP<1>	2uA
IDO<0>/ IUP<0>	1uA

表 22-4 DP0/DN0 的电压输出

控制方式：使能 QCEN 时，使能 LDO 模块

VLS0<2:0>	输出模式
000 (默认)	高阻
001	DP=2.0V, DN=2.0V
010	DP=2.0V, DN=2.7V
011	DP=2.7V, DN=2.0V
100	DP=2.7V, DN=2.7V
101~111	DP=1.2V, DN=1.2V

表 22-5 DP1/DN1 的电压输出

D1+	VLS1P<1:0>	输出模式
	00 (默认)	高阻
	1	0
	10	0.6V

	11	3.0V
--	----	------

D1-	VLS1N<1:0>	输出模式
	00 (默认)	高阻
	1	0
	10	0.6V
	11	3.0V

23. EEPROM&MTP 数据存储

EEPROM 数据存储。 “Electrically Erasable Programmable Read Only Memory” 为电可擦可编程只读存储器，由于其非易失的存储结构，即使在电源掉电的情况下存储器内的数据仍然保存完好。这种存储区扩展了 ROM 空间，对设计者来说增加了许多新的应用机会。EEPROM 可以用来存储产品编号、校准值、用户特定数据、系统软件配置参数或其它产品信息等。EEPROM 的数据读取和写入过程也会变的更简单。BJ8M302A 内部集成了 128X8 数据存储。

BJ8M302A 内部的 MTP 容量大小为 8KX8，当软件代码不需要使用这么多空间时，软件人员可将剩余的 MTP 容量拿来编自编程读/写操作，也就是剩余 MTP 容量可以拿来当做 EEPROM 使用。MTP 和 EEPROM 做自编程读/写使用，操作步骤几乎一样。最大的不同点是，EEPROM 拥有 10 万次的写入操作寿命，而 MTP 却只有 1000 次写入操作的寿命。

注意：黄色背景寄存器均位于 SFR 列表第 1 页，需要设置 PSW[5]=1 进行访问。

表 23-1 与 EEPROM&MTP 操作相关的寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR 值
FEh	IAPCON	VPPF	IAPLOCK	IAPWSEN	IAPCP	IAPWS		IAPRD	IAPWR	0000 1100
FAh	E2CON	-	-	-	-	-	-	E2RD	E2WR	---- --00
F9h	IAPTIR	IAPTIR<7:0>								0000 0000
FBh	IAPADRL	IAPADRL[7:0]								1111 1111
FCh	IAPADRH	IAPADRH[7:0]								1111 1111
FDh	IAPDAT	IAPDAT [7:0]								0000 0000

寄存器定义 23-IAPCON 编程控制寄存器

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IAPCON	VPPF	IAPLOCK	IAPWSEN	IAPCP	IAPWS		IAPRD	IAPWR
FEh	R	R	R/W	R/W	R/W		R/W	R/W

Bit7 VPPF: VPP 管脚高压实时标志位。当为 1 时，表示 VPP 脚有高压。为 0 时，表示 VPP 脚无高压。

Bit6 IAPLOCK: 此位为 1 时，EEPROM/MTP 编程处于锁定状态，需要软件对该位进行清零操作或者重新上电才可以继续进行编程写入 EEPROM/MTP 操作。只有发生不按文档规定的顺序值写入 IAPTIR 时，此位才会变为 1。正常正确的操作都不会使得改位为 1。

Bit5 IAPWSEN: 使能写等待超时功能，与 IAPWS 配合使用。

Bit4 IAPCP: 编程电压选择位

1: 设置编程电压为外部供电电压。VPP 需外接 DC9.0V

0: 设置使用芯片内部 charge pump 后的电压作为编程电压

Bit3-2 IAPWS: EEPROM/MTP 编程写入数据时等待时间选择位

00: 等待 2ms ;

01: 等待 4ms ;

10: 等待 8ms ;

11: 等待 16ms ; (默认)

Bit1 IAPRD: MTP 自编程读出操作使能位，高有效

进行 MTP 读出操作时，先配置好地址，再将该位置 1，则 CPU 会把对应的地址的 MTP 数据读出来，写入到寄存器 IAPDATA 中。在读出过程中，CPU 会将此位自动清 0。

Bit0 IAPWR: MTP 自编程写入操作使能位，高有效

进行 MTP 写入操作时，先配置好地址和数据，再将该位置 1，则 CPU 会把寄存器 IAPDATA 的数据写入到对应的 MTP 地址。在写入过程中，CPU 会将此位自动清 0。

寄存器 23-2 E2CON 数据存储器读写控制寄存器(此寄存器只用于 EEPROM 功能)

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
E2CON	-	-	-	-	-	-	E2RD	E2WR
FEh	-	-	-	-	-	-	R/W	R/W

Bit7-2 未使用

Bit1 E2RD: EEPROM 数据读使能位

1: 先软件配置好需读取数据的地址, 再置 1 即可。读取完成, 硬件自动清 0。

Bit0 E2WR: EEPROM 数据写使能位

1: 先软件配置好需写的数据和地址, 再置 1 即可。写操作完成, 硬件自动清 0

23.1 EEPROM 操作描述

当与数据存储模块接口时, IAPDAT 寄存器存放要读写的 8 位数据, IAPADRL 寄存器存放或被访问的数据单元的地址。这些器件具有 128 字节的数据 EEPROM, 地址范围从 0h 到 7Fh。IAPTIR 寄存器仅在数据 EEPROM 写过程中使用。要使能写操作, 必须将特定模式写入 IAPTIR。读写数据的电压需软件配置 IAPCON 的 IAPCP 位选择编程高压的来源。当软件配置 IAPCP=1 时, 需在 VPP 管脚外接 DC9.0V 电源; 当软件配置 IAPCP=0 时, 启动内部升压模块实现。通过 IAPADRL[6:0] 选择从 00h-7Fh 地址内读写任何数据。

注意: 302A 芯片内部本身有 Charge Pump 模块, 不需要外接高压便可以完成对 MTP 和 EEPROM 的写入操作。当然, 302A 也支持用外部高压来写入 MTP 和 EEPROM, 如果采用外部高压来做写入操作, 则只有在写入 MTP 和 EEPROM 时, VPP 电压 (9.0V) 才可以施加给芯片。在读操作时则需要将此高压撤销掉。

EEPROM 写入数据操作步骤:

- 如果使用外部高压写, 则需软件配置 IAPCP=1, 同时 VPP 外接 9.0V 电压, 检测 VPPF 标志位为 1; 如果使用芯片内部 charge pump, 则无需此步骤, 直接跳到(2)步。
- 使能 IAPWSEN 位, (其它位为默认值)
- 对 IAPTIR 连着写入 0x48, 0x65, 0x88, 进入到 IAP mode。注意: 如果往 IAPTIR 写入其他任何值, 则进入 IAPLOCK 状态。进行重新上电或通过软件写 IAPTIR[6]位为 0 来解除锁定, 软件置 0 后, 需要重新开始进行 IAP 写操作。
- 设置 IAPADRL[6:0], 设置数据要写入 EEPROM 对应 00h-7Fh 地址内的某个地址。
- 设置 IAPDAT, 即把需要写入数据寄存器的数据暂放在 IAPDAT 寄存器中。
- 使能 E2WR, CPU 将 IAPDAT [7:0]数据写入 IAPADRL [6:0]对应的 EEPROM 地址中。
- 插入 4 个 NOP 指令
- 如果要继续写, 则回到步骤(4)开始。如果要退出写入操作, 则到步骤(9)。
- 对 IAPTIR 写入 0, 便可以退出写操作。

例 23-1 写 EEPROM 数据

...

```

SETB    F0                //设置为第 1 页
MOV     IAPTIR, #48H      //进入 IAP MODE
MOV     IAPTIR, #65H
MOV     IAPTIR, #88H
MOV     IAPADRH, #00H    //写入地址 00H
MOV     IAPADRL, #00H
MOV     IAPDAT, #0A5H    //写入数据 A5H
NOP
    
```

```

ORL    E2CON,    #01H//写操作
NOP
NOP
NOP
NOP
...

```

EEPROM 读出数据操作步骤:

- (1) VPP 脚接 VDD 或者悬空状态。
- (2) 使能 IAPWSEN 位, (其它位为默认值)
- (3) 设置 IAPADRL [6:0], 对应 EEPROM 地址。
- (4) 使能 E2RD, CPU 将从 IAPADRL [6:0] 对应的 EEPROM 地址的数据读出到 IAPDAT [7:0]。
- (5) 插入 4 个 NOP 指令
- (6) 如果要继续读, 则回到步骤(3)开始。如果要退出写入操作, 则到步骤(7)。
- (7) 对 IAPTIR 写入 0, 便可以退出读操作。

例 23-2 读 EEPROM 数据

```

...
MOV    IAPADRH,    #00H        //读地址 00H
MOV    IAPADRL,    #00H
ORL    E2CON,    #02H        //读操作
NOP
NOP
NOP
NOP
MOV    A,          IAPDAT      //00H 地址数据赋给 A
MOV    rdata1,    A          //A 数据赋给变量 rdata1
...

```

23.2 MTP 操作描述

MTP 可以自编程的地址空间从 0H 到 1FEFH, IAPADDR[12:0] (由 IAPADRH 低 5 位和 IAPADRL 组成) 对应 MTP 的地址 0H。IAPDATA[7:0] 为 8 位数据。

MTP 写入数据操作步骤:

- (1) 如果使用外部高压写, 则需软件配置 IAPCP=1, 同时 VPP 外接 9.0V 电压, 检测 VPPF 标志位为 1; 如果使用芯片内部 charge pump, 则无需此步骤, 直接跳到(2)步。
- (2) 使能 IAPWSEN 位, (其它位为默认值)
- (3) 对 IAPTIR 连着写入 0x48, 0x65, 0x88, 进入到 IAP mode。注意: 如果往 IAPTIR 写入其他任何值, 则进入 IAPLOCK 状态。进行重新上电或通过软件写 IAPTIR[6]位为 0 来解除锁定, 软件置 0 后, 需要重新开始进行 IAP 写操作。
- (4) 设置 IAPADRH(高位地址)和 IAPADRL(地位地址), 设置数据要写入 MTP 对应 00h-1FEF 地址内的某个地址。
- (5) 设置 IAPDAT, 即把需要写入数据寄存器的数据暂放在 IAPDAT 寄存器中。
- (6) 使能 IAPWR, CPU 将 IAPDAT [7:0] 数据写入 IAPADRH 和 IAPADRL 对应的 MTP 地址中。
- (7) 插入 4 个 NOP 指令
- (8) 如果要继续写, 则回到步骤(4)开始。如果要退出写入操作, 则到步骤(9)。
- (9) 对 IAPTIR 写入 0, 便可以退出写操作。

例 23-1 写 MTP 数据

```

...
        SETB    F0                //设置为第 1 页
MOV     IAPTIR, #48H            //进入 IAP MODE
        MOV     IAPTIR, #65H
        MOV     IAPTIR, #88H
        MOV     IAPADRH, #00H    //写入地址 00H
        MOV     IAPADRL, #00H
        MOV     IAPDAT, #0A5H   //写入数据 A5H
        NOP
        ORL     IAPCON, #01H    //写操作
        NOP
        NOP
        NOP
        NOP
...

```

MTP 读出数据操作步骤:

- (1) VPP 脚接 VDD 或者悬空状态。
- (2) 使能 IAPWSEN 位，（其它位为默认值）
- (3) 设置 IAPADRH(高位地址)和 IAPADRL(地位地址)，对应 MTP 地址。
- (4) 使能 IAPRD，CPU 将从 IAPADRH(高位地址)和 IAPADRL(地位地址)对应的 MTP 地址的数据读出到 IAPDAT[7:0]。
- (5) 插入 4 个 NOP 指令
- (6) 如果要继续读，则回到步骤(3)开始。如果要退出写入操作，则到步骤(7)。
- (7) 对 IAPTIR 写入 0，便可以退出读操作。

例 23-2 读 MTP 数据

```

...
        MOV     IAPADRH, #00H    //读地址 00H
        MOV     IAPADRL, #00H
        ORL     IAPCON, #02H    //读操作
        NOP
        NOP
        NOP
        NOP
        MOV     A, IAPDAT        //00H 地址数据赋给 A
        MOV     rdata1, A       //A 数据赋给变量 rdata1
...

```

24. 乘/除法单元

BJ8M302A 的乘/除法/移位单元可完成 16 位×16 位乘法、32 位÷16 位除法，16 位÷16 位除法，32 位左/右移位，及 32 位标准化 (Normalizing) 等无符号数操作。由于运算操作过程由硬件自动完成，可为程序运行节省大量的运行时间。

主要特性

- 所有操作仅针对无符号数
- 硬件实现在 17 个 CPU 周期内完成 32 位÷16 位除法
- 硬件实现在 9 个 CPU 周期内完成 16 位÷16 位除法
- 硬件实现在 11 个 CPU 周期内完成 16 位×16 位乘法
- 硬件实现在 3~18 个 CPU 周期内完成 32 位左/右移位
- 硬件实现在 4~19 个 CPU 周期内完成 32 位标准化 (Normalizing)
- 具有操作错误标志位和溢出标准位

注意：棕色背景寄存器位于SFR列表第0页，需要软件配置PSW[5]=0进行访问。

表 24-1 与 MDU 相关的寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
F0h	B	B<7:0>								0000 0000
F9h	MD0	MD0<7:0>								0000 0000
FAh	MD1	MD1<7:0>								0000 0000
FBh	MD2	MD2<7:0>								0000 0000
FCh	MD3	MD3<7:0>								0000 0000
FDh	MD4	MD4<7:0>								0000 0000
FEh	MD5	MD5<7:0>								0000 0000
FFh	ARCON	MDEF	MDOV	SLR	SC<4:0>				0000 0000	

MD0、MD1、MD2、MD3、MD4、MD5 是乘除法运算操作中的操作数或者结构寄存器。这些寄存器可以读可写。

寄存器 24-1 ARCON 运算状态寄存器

符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ARCON	MDEF	MDOV	SLR	SC<4:0>				
FFh	R	R	R/W	R/W				

Bit7 MDEF: 运算错误标志位表示一个不当的运算，错误标志机制将自动在运算启动时（第一条写入数据至 MD0 的指令开始运行时）被使能，在运算结束时（最后一条读取运算结果的指令运行后）失效。在以下情况时，该标志位将被置位：(1)在运算的第二阶段（运算执行中）写入 MDx (x=0~5) 寄存器（运算将被重启或被新的运算打断）。(2)在运算的第二阶段（运算执行中）读取 MDx (x=0~5) 寄存器（运算未完成即读取结果），这种情况，错误标志位被设置，但运算操作将不会被影响。

该标志位只读，并且在 ARCON 寄存器被读取后自动清零

Bit6 MDOV: 运算溢出标志。在以下情况时，该标志位将被置位：(1)除数为 0。(2)乘法运算结果大于 0FFFFh。(3) 标准化运算的操作数最高位 (MD3.7) 为 1。**该标志位只读，并且在以上情况被移除时自动清零**

Bit5 SLR: 位移方向选择位，软件设置（仅在移位运算时有效）

- 1: 32 位数据右移
- 0: 32 位数据左移

Bit4-0 SC<4:0>: 移位次数值

当设置为全 0 时，标准化操作将被使能。运算完成后，将记录此次运算中移位的次数

当设置为非全 0 时，移位操作将被使能并表示移位次数值，**移位完成后清零**

24.1 MDSU 运算

乘法运算操作可分为三个阶段：装载 MDx (x=0~5) 寄存器；运算执行；从 MDx (x=0~5) 寄存器读取结果。

1) 装载 MDx (x=0~5) 寄存器

MDSU 将执行的操作完全由 MDx 寄存器的写入顺序决定，如下表所示。例如要执行 16 位 ÷ 16 位除法操作，需最先向 MD0 中写入被除数的 LSB，然后向 MD1 中写入被除数的 MSB，再向 MD4 中写入除数的 LSB，最后向 MD5 中写入除数的 MSB。MDSU 的错误标志机制 (MDEF) 将自动在操作启动时 (第一条写入数据至 MD0 的指令开始运行时) 被使能，而写入数据至 MD5 的指令运行完成后将自动启动操作的第二阶段 (运算执行)。

表 24-2 MDx 寄存器写入顺序

写入顺序	32 位 ÷ 16 位	16 位 ÷ 16 位	16 位 × 16 位	32 位左/右移位 标准化 (Normalizing)
最先写入	MD0 被除数 (LSB)	MD0 被除数 (LSB)	MD0 被乘数 (LSB)	MD0 (LSB)
	MD1 被除数	MD1 被除数 (MSB)	MD4 乘数 (LSB)	MD1
	MD2 被除数		MD1 被乘数 (MSB)	MD2
	MD3 被除数 (MSB)			MD3 (MSB)
	MD4 除数 (LSB)	MD4 除数 (LSB)		
最后写入	MD5 除数 (MSB)	MD5 除数 (MSB)	MD5 乘数 (MSB)	ARCON

2) 运算执行

MDSU 的运算执行过程由硬件自主独立完成，无需 CPU 参与。不同的操作类型需要不同的运算执行时间，如下表所示。软件需在适当的时刻读取运算结果。若提前读取，错误标准机制位 (MDEF) 将被置位，读取值不可预知，但运算执行不会受到干预。

表 24-3 MDSU 运算执行时间

操作类型	所需的 CPU 周期数	
32 位 ÷ 16 位除法	17	
16 位 ÷ 16 位除法	9	
16 位 × 16 位乘法	11	
左/右移位	Min: 3 (当 SC=01h)	Max: 18 (当 SC=1Fh)
标准化 (Normalizing)	Min: 4 (当 SC <-01h)	Max: 19 (当 SC <-01h)

3) 从 MDx (x=0~5) 寄存器读取结果

MDSU 操作运算的结果读取顺序如下表所示。以 16 位 ÷ 16 位除法操作为例，先从 MD0 中读取商的 LSB，然后从 MD1 中读取商的 MSB，再从 MD4 中读取余数的 LSB，最后从 MD5 中读取余数的 MSB。最后一条结果读取指令的执行标志着整个操作的完成。

表 24-4 MDx 寄存器读取顺序

读取顺序	32 位 ÷ 16 位	16 位 ÷ 16 位	16 位 × 16 位	32 位左/右移位 标准化 (Normalizing)
最先读取	MD0 商 (LSB)	MD0 商 (LSB)	MD0 积 (LSB)	MD0 (LSB)
	MD1 商	MD1 商 (MSB)	MD1 积	MD1
	MD2 商		MD2 积	MD2
	MD3 商 (MSB)			
	MD4 余数 (LSB)	MD4 余数 (LSB)		
最后读取	MD5 余数 (MSB)	MD5 余数 (MSB)	MD3 积 (MSB)	MD3 (MSB)

24.2 移位操作

32 位数据的移位结果仍然是 32 位数据。操作数存放在 MD3 (MSB) ~MD0 (LSB) 中，ARCON 寄存器中的 SLR 位决定左/右移，ARCON 寄存器中的 SC[4:0] 决定移位次数，移位结果仍然存放在 MD3 (MSB) ~MD0 (LSB) 中。由于移位操作仅针对无符号数，无论左移还是右移操作，均用 0 来填补移出的空位。

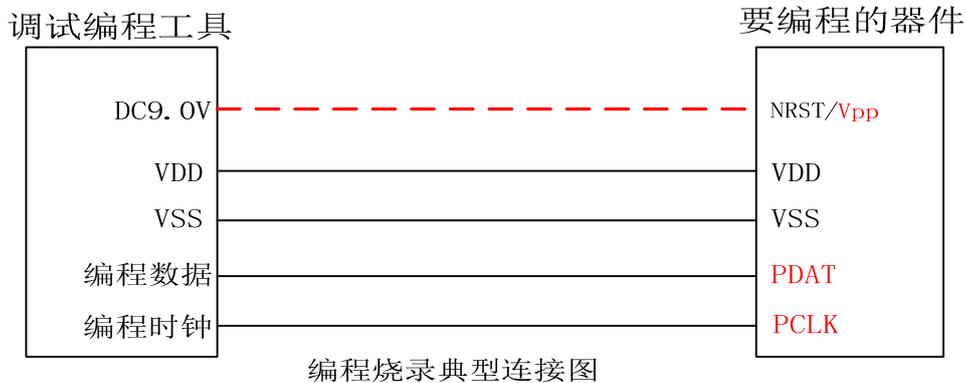
24.3 标准化操作 (Normalizing)

32 位数据标准化操作是将存放于 MD0 (LSB) ~MD3 (MSB) 寄存器中的 32 位数据进行左移操作，直到最高位 (MD3[7]) 为 1 为止。标准化操作结束后，ARCON 寄存器中 SC[4:0] 位将得到在此次标准化操作中移位的次数。若标准化操作的被操作数最高位本为 1，则硬件自动启动溢出机制，即 ARCON 寄存器中的 MDOV 位被置 1。

25. 调试烧录程模块

BJ8M302A 编程可分为自编程、在线调试 和脱机烧录三种。

图 24-1 调试烧录典型接线图



注意：BJ8M302A 内部集成了 Charge Pump 模块。调试和自编程模式下，可通过软件进行开启或者关闭 Charge Pump 模块，从而可以省掉 NRST/Vpp 这条编程线，只需要 VDD/VSS/PDAT/PCLK 这 4 跟线。

在线调试操作步骤：

1. 采用我司调试器按照 图 24-1 进行物理连接。
2. 软件配置 WCKEN=1。
3. 将编辑编译好的代码下载到目标芯片中去。
4. 开始调试。

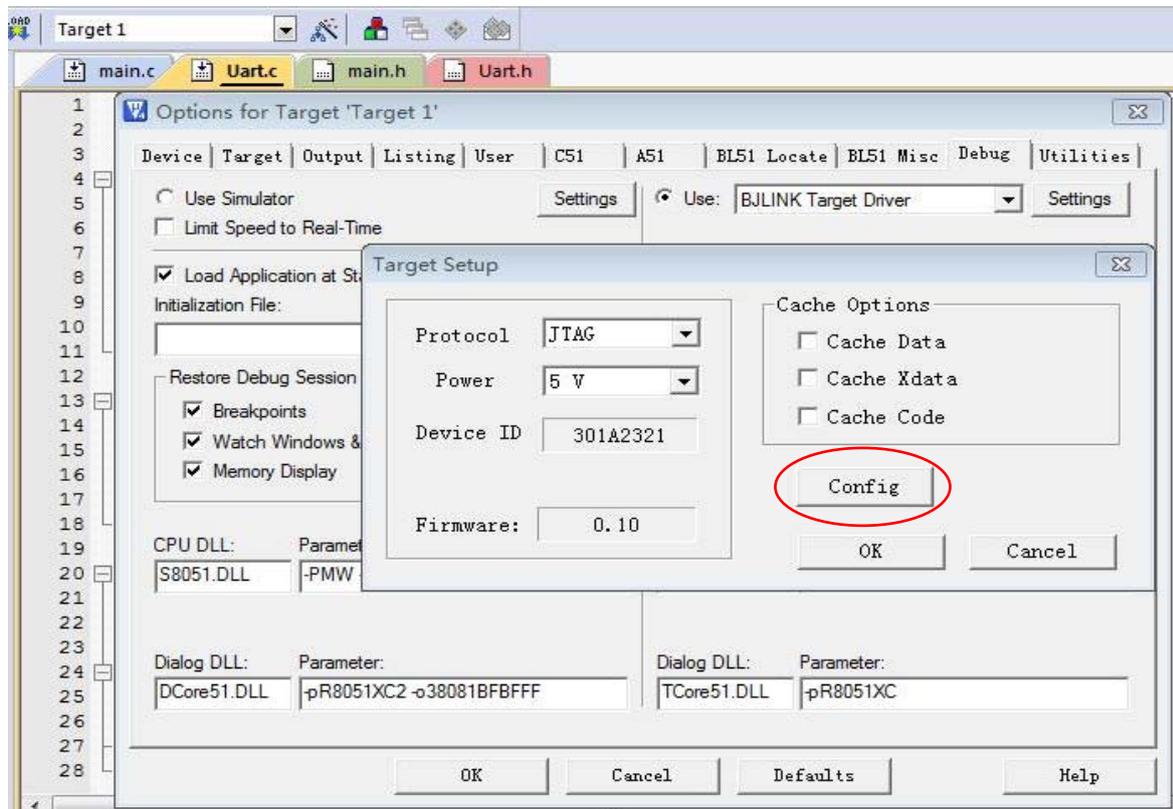
注意：P3.0 和 P3.2 默认为调试功能脚，不能当作 GPIO 脚。如果不需要调试功能，而需要将 P3.0 和 P3.2 设置作为 GPIO 功能，则只需要将 DEBUGEN 设置为 0 便可以。(DEBUGEN 在寄存器 CLKCON 的第 7bit)。

26. 程序配置

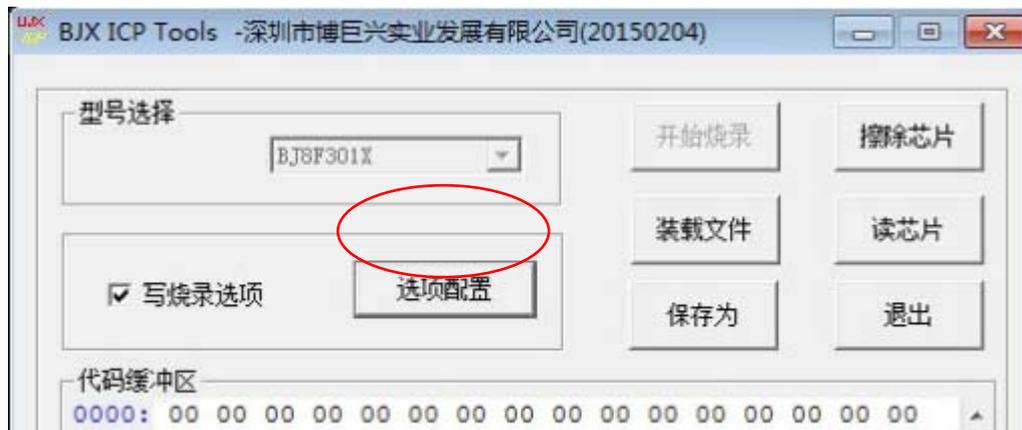
程序配置是通过 Keil IDE 的软件或者 ICP 烧录软件，通过程序配置 IC 内部某个模块，使工作在某种特定模式下的控制命令配置字。它和用户程序一同烧录进入单片机，烧录完后的程序配置字再无法通过应用程序修改。

软件配置是在程序运行的过程中，对特殊寄存器赋值，让其某个模块工作在相应的模式下。程序运行的过程中，可以随时对特殊寄存器赋不同的值来改变模块工作模式。

IDE 软件的程序配置方法：进入如下界面后点击“Config”后的弹出对话框进行程序配置



ICP 烧录软件进行程序配置方法：进入如下界面后点击“选项配置”后的弹出对话框进行程序配置



所有位必须按系统的需要定义，具体内容可参考下表：

序号	功能	选项	序号	功能	选项
1	ICSEL[2:0]内 振 时钟 选择	1. 内部 RC: 1M	4	低电压 复位使能	1. 开启低电压复位
		2. 内部 RC: 2M			2. 关闭低电压复位(默认)
		3. 内部 RC: 8M	5	低电压 复位阈值 选择	1. 1.8V 复位
		4. 内部 RC: 16M			2. 2.3V 复位
2	RSTSEL 复位时间 选择	1. 228mS			3. 2.7V 复位
		2. 72mS			4. 3.0V 复位
		3. 6.5mS			5. 3.3V 复位
		4. 140uS			6. 3.6V 复位
		5. 18mS (默认)			7. 3.9V 复位
3	代码 保护	1. 不进行代码保护	6	CLKDIV [0:0] 主时钟 分频选择	1. MCK 1 分频
		2. 代码保护, 读出全为 0			2. MCK 2 分频
		3. MCK 4 分频			
		4. MCK 8 分频			

序号	功能	选项
7	SYSCKM 【2:0】 管脚 配置	1. IRC: 选择内部 RC 时钟。P0.0 和 P0.1 做为 IO 口 (默认)
		2. ERC: 外部 RC, P0.2 做为外 IO;
		3. HXT: 外接 2M~16M 晶振或者 OSCI (P0.1) 做为外部时钟输入
		4. LXT: 外接低频 32768 晶振
		5. DSYSCK: RTCC 模式。 STOP 模式下, 硬件关闭内部时钟, 外部时钟不关闭。
8	复位引 脚配置	1. P0.0 做为 IO 输入口
		2. P0.0 作为复位口
9	T1PWM3 输出 引脚配置	1. P2.3 作为 T1CAP1/T1PWM3 输入输出口 (默认)
		2. P1.6 作为 T1CAP1/T1PWM3 输入输出口
10	T1PWM2 输出 引脚配置	1. P3.3 作为 T1CAP2/T1PWM2 输入输出口 (默认)
		2. P1.0 作为 T1CAP2/T1PWM2 输入输出口
11	T1PWM1 输出 引脚配置	1. P3.4 作为 T1CAP1/T1PWM1 输入输出口 (默认)
		2. P1.7 作为 T1CAP1/T1PWM1 输入输出口
12	T1PWM0 输出 引脚配置	1. P3.1 作为 T1CAP0/T1PWM0 输入输出口 (默认)
		2. P3.2 作为 T1CAP0/T1PWM0 输入输出口
13	TCKSEL 频率选择	1. 输出 48M 时钟 (默认)
		2. 输出 32M 时钟
14	T2PWM3 输出 引脚配置	1. P2.2 作为 T2CAP3/T2PWM3 输入输出口 (默认)
		2. P2.0 作为 T2CAP3/T2PWM3 输入输出口
15	T2PWM1 输出 引脚配置	1. P2.0 作为 T2CAP1/T2PWM1 输入输出口 (默认)
		2. P1.5 作为 T2CAP1/T2PWM1 输入输出口
16	T2BKIN 输出 引脚配置	1. P2.3 作为 T2 T2BKIN 输入口 (默认)
		2. P1.2 作为 T2 T2BKIN 输入口
17	RCOUTP 输出引 脚配置	1. P30 为 RC 时钟输出口
		2. P16 为 RC 时钟输出口

27. 指令系统

BJ8M302A 集成的快速标准的 8051 内核，下面介绍了寻址方式，助记符说明、指令周期、字节数和机器周期每个指令需要被执行的数字；注意循环的次数，给出了无程序存储器的等待状态等。

27.1 寻址方式

(1)立即寻址

在指令操作数中直接给出参加运算的操作数，通常用“#”作为前缀。指令举例如下：

```
MOV A, #20H
MOV SP, #81H
MOV DPTR, #1234H
```

(2)直接寻址

指令中直接给出操作数的地址。使用于内部数据存储器 RAM 低 128 单元和 80H-FFH 区的 SFR 区。指令举例如下：

```
MOV 65H, A ; 将 A 的内容送入内部 RAM 的 65H 单元地址中
MOV A, direct ; 将直接地址单元的内容送入 A 中
MOV direct, direct ; 将直接地址单元的内容送直接地址单元
MOV P1, #85H ; 将立即数 85H 送入中断允许寄存器 IE
```

(3)间接寻址

间接寻址方式是指寄存器中存放的是操作数的地址，即操作数是通过寄存器间接得到的，因此称为寄存器间接寻址。间接寻址采用 R0 或 R1 前添加“@”符号来表示。

```
MOV A, @R0 ; 把内部 R0 内容指定地址里面的数据放到累加器 A 中
MOVX A, @R0 ; 把外部 R0 内容指定地址里面的数据放到累加器 A 中
MOVX A, @DPTR ; 把外部 DPTR 内容指定地址里面的数据放到累加器 A 中，通常用来访问片外数据存储器
```

(4)寄存器寻址

寄存器寻址时对选定的工作寄存器 R7~R0、累加器 A、通用寄存器 B、地址寄存器和进位 C 中的数进行操作。其中寄存器 R7~R0 由指令码的低 3 位表示，ACC、B、DPTR 及进位位 C 隐含在指令码中。因此，寄存器寻址也包含一种隐含寻址方式。

寄存器工作区的选择由程序状态字寄存器 PSW 中的 RS1、RS0 来决定。指令操作数指定的寄存器均指当前工作区的寄存器

```
MOV A, R2
INC R0
```

(5)相对寻址

把指令中给定的地址偏移量与本指令所在单元地址（PC 内容）相加得到真正有效的操作数所存放的地址。由于目的地址是相对于 PC 中的基地址而言，所以这种寻址方式成为相对寻址。偏移量为带符号的数，所能表示的范围为+127~-128。这种寻址方式主要用于转移指令。

```
JMP @A+DPTR
JC $+100H
```

表示若进位位 C 为 0，则程序计数器 PC 中的内容不改变，即不转移。若进位位 C 为 1，则以 PC 中的当前值及基地址，加上偏移量 100H 后所得到的结果作为该转移指令的目的地址。

(6)变址寻址

在变址寻址方式中，指令操作数制定一个存放变址基址的变址寄存器。变址寻址时，偏移量与变址基值相加，其结果作为操作数的地址。变址寄存器有程序计数器 PC 和地址寄存器 DPTR。

```
MOVC  A,    @A+DPTR
MOV   A,    @A+DPTR
JMP   @A+DPTR

MOVC  A,    @A+PC
```

表示累加器 A 为偏移量寄存器，其内容与地址寄存器 DPTR 中的内容相加，其结果作为操作数的地址，取出该单元中的数送入累加器 A 中。

(7)位寻址

位寻址就是对内部 RAM 或可位寻址的特殊功能寄存器 SFR 内的某个位，直接加以置位为 1 或复位为 0。内部数据存储单元 RAM 的低 128 单元中有一个区域叫位寻址区。它的单元地址是 20H-2FH。共有 16 个单元，一个单元是 8 位，所以位寻址区共有 128 位。这 128 位都单独有一个位地址，其位地址的名字就是 00H-7FH。

```
MOV   C,    20H
```

表 27-1 对数据寻址方式说明

符号	说明
Rn	工作寄存器 R0-R7
direct	128 内部 RAM 的位置之一或任何特殊功能寄存器
@Ri	由寄存器 R0 或 R1 间接寻址的内部或外部 RAM 的位置
#data	包含在指令的 8 位常数（立即数）
#data16	16 位常数列作为字节 2 和 3 指令（立即数）
bit	其中任何 128 软件标志之一位于内部 RAM，或可位寻址的特殊功能寄存器软件标志位，包括 I/O 引脚和状态字。
A	累加器

表 27-2 程序寻址方式说明

符号	说明
addr16	对于 LCALL 或 LJMP 的目的地址，可以是程序存储器地址空间的 64-Kbyte 页内的任何地方
addr11	对于 ACALL 或 AJMP 的目的地址，在同一个 2-Kbyte 页面程序内存指令后的第一个字节
rel	SJMP 和所有的条件跳转包括一个 8 位的偏移字节。它的范围是+127/-128 字节相对于下面的指令的第一个字节

27.2 功能命令的说明

表 27-3 算术运算

助记符	说明	代码	字节	周期
ADD A, Rn	$(A) + (Rn) \rightarrow A$	0x28-0x2F	1	1
ADD A, direct	$(A) + (\text{direct}) \rightarrow A$	0x25	2	2
ADD A, @Ri	$(A) + ((Ri)) \rightarrow A$	0x26-0x27	1	2
ADD A, #data	$(A) + \text{data} \rightarrow A$	0x24	2	2
ADDC A, Rn	$(A) + (Rn) + (Cy) \rightarrow A$	0x38-0x3F	1	1
ADDC A, direct	$(A) + (\text{direct}) + (Cy) \rightarrow A$	0x35	2	2
ADDC A, @Ri	$(A) + ((Ri)) + (CY) \rightarrow A$	0x36-0x37	1	2
ADDC A, #data	$(A) + \text{data} + (CY) \rightarrow A$	0x34	2	2
SUBB A, Rn	$(A) - (Rn) - (CY) \rightarrow A$	0x98-0x9F	1	1
SUBB A, direct	$(A) - (\text{direct}) - (CY) \rightarrow A$	0x95	2	2
SUBB A, @Ri	$(A) - ((Ri)) - (CY) \rightarrow A$	0x96-0x97	1	2
SUBB A, #data	$(A) - \text{data} - (CY) \rightarrow A$	0x94	2	2
INC A	$(A) + 1 \rightarrow A$	0x04	1	1
INC Rn	$(Rn) + 1 \rightarrow Rn$	0x08-0x0F	1	1
INC direct	$(\text{direct}) + 1 \rightarrow \text{direct}$	0x05	2	2
INC @Ri	$((Ri)) + 1 \rightarrow (Ri)$	0x06-0x07	1	2
INC DPTR	$(DPTR) + 1 \rightarrow DPTR$	0xA3	1	1
DEC A	$(A) - 1 \rightarrow A$	0x14	1	1
DEC Rn	$(Rn) - 1 \rightarrow Rn$	0x18-0x1F	1	1
DEC direct	$(\text{direct}) - 1 \rightarrow \text{direct}$	0x15	2	2
DEC @Ri	$((Ri)) - 1 \rightarrow (Ri)$	0x16-0x17	1	2
MUL AB	$(A) \cdot (B) \rightarrow AB$	0xA4	1	4
DIV	$(A) / (B) \rightarrow AB$	0x84	1	4
DA A	对 A 进行十进制调整	0xD4	1	1

表 27-4 逻辑运算

助记符	说明	代码	字节	周期
ANL A, Rn	$(A) \wedge (Rn) \rightarrow A$	0x58-0x5F	1	1
ANL A, direct	$(A) \wedge (\text{direct}) \rightarrow A$	0x55	2	2
ANL A, @Ri	$(A) \wedge ((Ri)) \rightarrow A$	0x56-0x57	1	2
ANL A, #data	$(A) \wedge \text{data} \rightarrow A$	0x54	2	2
ANL direct, A	$(\text{direct}) \wedge (A) \rightarrow \text{direct}$	0x52	2	2
ANL direct, #data	$(\text{direct}) \wedge \text{data} \rightarrow \text{direct}$	0x53	3	3
ORL A, Rn	$(A) \vee (Rn) \rightarrow A$	0x48-0x4F	1	1
ORL A, direct	$(A) \vee (\text{direct}) \rightarrow A$	0x45	2	2
ORL A, @Ri	$(A) \vee ((Ri)) \rightarrow A$	0x46-0x47	1	2
ORL A, #data	$(A) \vee \text{data} \rightarrow A$	0x44	2	2

助记符	说明	代码	字节	周期
ORL direct, A	(direct) \vee (A) \rightarrow direct	0x42	2	2
ORL direct, #data	(direct) \vee data \rightarrow direct	0x43	3	3
XRL A, Rn	(A) 异或 (Rn) \rightarrow A	0x68-0x6F	1	1
XRL A, direct	(A) 异或 (direct) \rightarrow A	0x65	2	2
XRL A, @Ri	(A) 异或 ((Ri)) \rightarrow A	0x66-0x67	1	2
XRL A, #data	(A) 异或 data \rightarrow A	0x64	2	2
XRL direct, A	(direct) 异或 (A) \rightarrow direct	0x62	2	2
XRL direct, #data	(direct) 异或 data \rightarrow direct	0x63	3	3
CLR A	0 \rightarrow A	0xE4	1	1
CPL A	将累加器 A 的值按位取反	0xF4	1	1
RL A	A 循环左移一位	0x23	1	1
RLC A	A 带进位循环左移一位	0x33	1	1
RR A	A 循环右移一位	0x03	1	1
RRC A	A 带进位循环右移一位	0x13	1	1
SWAP A	A 半字节交换	0xC4	1	1

表 27-5 数据传输操作

助记符	说明	代码	字节	周期
MOV A, Rn	(Rn) \rightarrow A	0xE8-0xEF	1	1
MOV A, direct	(direct) \rightarrow A	0xE5	2	2
MOV A, @Ri	((Ri)) \rightarrow A	0xE6-0xE7	1	2
MOV A, #data	data \rightarrow A	0x74	2	2
MOV Rn, A	(A) \rightarrow Rn	0xF8-0xFF	1	1
MOV Rn, direct	(direct) \rightarrow Rn	0xA8-0xAF	2	2
MOV Rn, #data	data \rightarrow Rn	0x78-0x7F	2	2
MOV direct, A	(A) \rightarrow direct	0xF5	2	2
MOV direct, Rn	(Rn) \rightarrow direct	0x88-0x8F	2	2
MOV direct1, direct2	(direct2) \rightarrow direct1	0x85	3	3
MOV direct, @Ri	((Ri)) \rightarrow direct	0x86-0x87	2	2
MOV direct, #data	data \rightarrow direct	0x75	3	3
MOV @Ri, A	(A) \rightarrow (Ri)	0xF6-0xF7	1	1
MOV @Ri, direct	(direct) \rightarrow (Ri)	0xA6-0xA7	2	2
MOV @Ri, #data	data \rightarrow (Ri)	0x76-0x77	2	2
MOV DPTR, #data16	data16 \rightarrow DPTR	0x90	3	3
MOVC A, @A+DPTR	((A) + (DPTR)) \rightarrow A	0x93	1	3
MOVC A, @A+PC	((A) + (PC)) \rightarrow A	0x83	1	3
MOVX A, @Ri	((Ri)) \rightarrow A	0xE2-0xE3	1	3-10
MOVX A, @DPTR	((DPTR)) \rightarrow A	0xE0	1	3-10
MOVX @Ri, A	(A) \rightarrow (Ri) + (P2)	0xF2-0xF3	1	3-12

助记符	说明	代码	字节	周期
MOVX @DPTR, A	(A) → (DPTR)	0xF0	1	3-12
PUSH direct	(SP) +1→SP, (direct) → (SP)	0xC0	2	2
POP direct	((SP)) →direct, (SP) -1→SP	0xD0	2	2
XCH A, Rn	(A) ↔ (Rn)	0xC8-0xCF	1	1
XCH A, direct	(A) ↔ (direct)	0xC5	2	2
XCH A, @Ri	(A) ↔ ((Ri))	0xC6-0xC7	1	2
XCHD A, @Ri	(A) 0-3↔ ((Ri)) 0-3	0xD6-0xD7	1	2

表 27-6 程序分支

助记符	说明	代码	字节	周期
ACALL addr11	(PC) +2→PC, (SP) +1→SP, (PC) L→(SP) (SP) +1→SP, (PC) H→(SP), addr11→PC10~0	xxx10001b	2	2
LCALL addr16	(PC) +2→PC, (SP) +1→SP, (PC) L→(SP) (SP) +1→SP, (PC) H→(SP), addr16 →PC	0x12	3	3
RET	从子程序返回	0x22	1	4
RETI	从中断返回	0x32	1	4
AJMP addr11	绝对跳转: addr11→PC10~0	xxx00001b	2	2
LJMP addr16	长跳转: addr16 →PC	0x02	3	3
SJMP rel	短跳转 (相对地址)	0x80	2	2
JMP @A+DPTR	直接跳转到 DPTR	0x73	1	3
JZ rel	如果累加器为零则跳转	0x60	2	3
JNZ rel	如果累加器不为零则跳转	0x70	2	3
JC rel	如果进位标志被设置则跳转	0x40	2	3
JNC rel	如果进位标志没有被设置则跳转	0x50	2	3
JB bit, rel	若直接寻址位设置则跳转	0x20	3	4
JNB bit, rel	若直接寻址位没有设置则跳转	0x30	3	4
JBC bit, rel	若直接寻址位设置, 则清除位并跳转	0x10	3	4
CJNE A, direct, rel	比较直接寻址的数据和累加器, 不相等则跳转	0xB5	3	4
CJNE A, #data, rel	比较立即数与累加器, 如果不相等则跳转	0xB4	3	4
CJNE Rn, #data, rel	比较立即数与寄存器, 如果不相等则跳转	0xB8-0xBF	3	1-3
CJNE @Ri, #data, rel	比较立即数与间接寻址, 如果不相等则跳转	B6-B7	3	1-4
DJNZ Rn, rel	递减寄存器, 如果不为零则跳转	D8-DF	2	3
DJNZ direct, rel	递减直接寻址, 如果不为零则跳转	D5	3	4
NOP	无操作	0	1	1

表 27-7 布尔操作

助记符	说明	代码	字节	周期
CLR C	清除进位标志	0xC3	1	1
CLR bit	清除直接寻址位	0xC2	2	2
SETB C	设置进位标志	0xD3	1	1
SETB bit	设置直接寻址位	0xD2	2	2
CPL C	进位标志取反	0xB3	1	2
CPL bit	直接寻址位取反	0xB2	2	2
ANL C, bit	直接寻址位和进位标志相与	0x82	2	2
ANL C, /bit	直接寻址位取反和进位标志相与	0xB0	2	2
ORL C, bit	直接寻址位和进位标志相或	0x72	2	2
ORL C, /bit	直接寻址位取反和进位标志相或	0xA0	2	2
MOV C, bit	移动直接寻址位到进位标志	0xA2	2	2
MOV bit, C	移动进位标志直接寻址位	0x92	2	2

每个指令的持续时间可以使用下面的公式来计算：

如果 BYTES > 1 或 CYCLES = 1, 则 DURATION = CYCLES + (BYTES+R) * P + X*D
 否则 DURATION = CYCLES + (2+R) * P + X*D

- 式中：
- BYTES 是指令字节数（见上表）
 - CYCLES 是无等待状态的数目（见上表）
 - R =1 为 MOVC 指令，否则 R =0
 - X =1 为 MOVX 指令，否则 X=0

在程序存储器写模式（PMW）是公式 MOVX 如下：

$$DURATION = CYCLES + (2+X) * P$$

27.3 读-修改-写指令

从 SFR 或内部 RAM 中读取一个字节的指令，修改并重写回来，被称为“读-修改-写”指令。当目标是一个 I/O 端口（P0-P3），或一个端口位，这些指令读取输出锁存器，而不是脚。

表 27-8 RMW 说明

助记符	说明	代码	字节	周期
ANL direct, A	累加器和直接寻址相与	0x52	2	3
ANL direct, #data	立即数和直接寻址相与	0x53	3	4
ORL direct, A	累加器和直接寻址相或	0x42	2	3
ORL direct, #data	立即数和直接寻址相或	0x43	3	4
XRL direct, A	累加器和直接寻址异或	0x62	2	3
XRL direct, #data	立即数和直接寻址异或	0x63	3	4
JBC bit, rel	如果位设置，则清除并跳转	0x10	3	4
CPL bit	按位取反	0xB2	2	3
INC direct	直接递增	0x05	2	3
INC @Ri	间接递增	0x06-0x07	1	3

助记符	说明	代码	字节	周期
DEC direct	直接递减	0x15	2	3
DEC @Ri	间接递减	0x16-0x17	1	3
DJNZ direct,rel	递减, 若不为零则转移	0xD5	3	4
MOV bit,C	移动进位标志位到直接寻址位	0x92	2	3
CLR bit	清除位	0xC2	2	3
SETB bit	设置位	0xD2	2	3

28. 电气特性

28.1 极限值

表 28-1 极限值

参数	标号	条件	范围	单位
供电电压	VDD	-	- 0.3 to + 5.5	V
输入电压	VI	所有 I/O 口	- 0.3 to VDD + 0.3	V
输出电压	VO	所有输出口	- 0.3 to VDD + 0.3	V
I/O 口输出电流	IOH	单个 I/O 口工作时	- 25	mA
		所有 I/O 口工作时	- 80	mA
I/O 口输入电流	IOL	单个 I/O 口工作时	+ 20	mA
		所有 I/O 口工作时	+ 80	mA
工作温度	TA	-	- 40 to + 85	°C
储藏温度	TSTG	-	- 40 to + 125	°C

28.2 直流特性

表 28-2 直流特性 (VDD = 1.8~5.5V, TA = 25°C, 如无其他说明则都是此条件)

参数	标号	条件	最小值	典型值	最大值	单位
工作电压	VDD	振荡器频率 = 0.4 - 2MHz/2CLK	1.8	-	5.5	V
		振荡器频率 = 0.4 - 8MHz/2CLK	3.0	-	5.5	
		振荡器频率 = 0.4 - 16MHz/2CLK	4.0	-	5.5	
输入高电压	VIH1	P0/P1/P2/P3 和 RESET	VDD = 1.8 to 5.5V	0.7VDD	VDD	V
	VIH2	OSCI 和 OSCO		0.8VDD		
输入低电压	VIL1	P0/P1/P2/P3 和 RESET	VDD = 1.8 to 5.5V	-	0.3VDD	V
	VIL2	OSCI 和 OSCO		0.2VDD		
输出高电压	VOH	VIN=0.9VDD, P0/P1/P2/P3	VDD = 5V	20		mA
输出低电压	VOL	VIN=0.1VDD, P0/P1/P2/P3	VDD = 5V	20		mA
	VOL_H1	VIN=0.1VDD, P1.6/P1.7/P2.0/P2.1		40		mA
	VOL_H2	VIN=0.1VDD, P1.6/P1.7/P2.0/P2.1		60		mA
	VOL_H3	VIN=0.1VDD, P1.6/P1.7/P2.0/P2.1		80		mA
输入高漏电流	IIH	所有输入口	VIN = VDD	-1	1	µA
输入低漏电流	IIL	所有输入口	VIN = 0V	-1	1	µA
上拉电阻	RP1	VIN = 0V, P0/P1/P2/P3(不包含 P00)	VDD = 5V	50		kΩ
		VIN = 0V, P00		200		
下拉电阻	RP2	VIN = 0V,	VDD = 5V	50		
供电电流	IDD1 Run 模式	IRC 16MHz 频率, 2CLK	VDD=5V	-	7.7	mA
		IRC 8MHz 频率, 2CLK	VDD=5V		4.9	mA
		IRC 2MHz 频率, 2CLK	VDD=5V		1.5	mA
	IDD2 IDLE 模式	IRC 16MHz 频率, 2CLK	VDD=5V		2.5	mA
		IRC 8MHz 频率, 2CLK	VDD=5V		1.3	mA
		IRC 2MHz 频率, 2CLK	VDD=5V		380	µA
	IDD3	睡眠模式 TA=25°C	VDD = 3.3V		-	0.9
VDD = 5V				-	1.1	

28.3 RC 振荡器特性

表 28-3 RC 振荡器特性 (VDD = 5.0V, TA = 25°C, 如无其他说明则都是此条件)

振荡器	时钟电路	测试条件	最小值	典型值	最大值	单位
外部 RC 振荡器	-	VDD = 5v	0.1	-	10	MHz

振荡器	时钟电路	测试条件	最小值	典型值	最大值	单位
内部 RC 振荡器	-	-	-	16	-	MHz
内部 RC 振荡器精度	-	VDD = 5V TA = 25°C	-	-	1 (TBD)	%
		VDD = 5.0V TA = -40°C to +85°C	-	-	3 (TBD)	%
		VDD = 1.8 to 5.5V TA = -40°C to +85°C	-	-	5 (TBD)	%

28.4 RC 振荡中电阻、电容、电压关系列表

表 28-4 外部 RC 振荡器电阻电容电压搭配关系

R(kΩ)	C (pF)	VDD(V)	频率(MHz)	R(kΩ)	C (pF)	VDD(V)	实际频率(MHz)
8.2	20	5	14.95	8.2	20	3	10.32
30	20	5	7.62	30	20	3	3.90
75	20	5	3.72	75	20	3	1.84
150	20	5	2.08	150	20	3	0.98
300	20	5	1.12	300	20	3	0.51
8.2	30	5	15.09	8.2	30	3	10.46
30	30	5	7.80	30	30	3	4.00
75	30	5	3.81	75	30	3	1.84
150	30	5	2.11	150	30	3	0.98
300	30	5	1.13	300	30	3	0.51
8.2	47	5	15.24	8.2	47	3	10.67
30	47	5	8.00	30	47	3	4.21
75	47	5	4.00	75	47	3	1.88
150	47	5	2.19	150	47	3	0.99
300	47	5	1.15	300	47	3	0.51

28.3 LVR 电路特性

表 28-4 LVR 电路特性 (TA = -40°C to +85°C, VDD = 1.8V to 5.5V)

参数	标号	设置复位值	条件	最小值	典型值	最大值	单位
低电压 复位电压值	VLVR	关闭	-	1.5V	1.6V	1.7V	V
		1.8V		1.7V	1.8V	1.9V	
		2.3V		2.2V	2.3V	2.4V	
		2.7V		2.6V	2.7V	2.8V	
		3.0V		2.9V	3.0V	3.1V	
		3.3V		3.2V	3.3V	3.4V	
		3.6V		3.5V	3.6V	3.7V	
		3.9V		3.8V	3.9V	4.0V	

28.5 LVD 电路特性

表 28-5 LVD 电路特性 (TA = - 40°C to + 85°C, VDD = 1.8V to 5.5V)

参数	标号	条件	最小值	典型值	最大值	单位
低电测试电压值	VLVD		2.2V		2.2V	V
			2.4V		2.4V	
			2.6V		2.6V	
			2.9V		2.9V	
			3.2V		3.2V	
			3.5V		3.5V	
			3.8V		3.8V	
			4.1V		4.1V	

28.6 A/D 转换特性

表 28-6 LVD 电路特性 (VDD = 5V, TA = 25°C, 除非另有说明)

参数	标号	条件	最小值	典型值	最大值	单位
电压范围	V _{AD}	-	2.5	5.0	5.5	V
精度	N _R			12		bit
A/D输入电压	V _{IN}		GND		VREFP	V
A/D输入阻抗	R _{IN}			80		KΩ
外部模拟参考电压	VREF		2.5		VDD	V
A/D转换电流	I _{AD}			0.8		mA
微分非线性误差	DNL			±1		LSB
积分非线性误差	INL			±2		LSB
满刻度误差	E _{Gain}				±4	LSB
偏移量误差	V _{OS}				±2	LSB
总绝对误差	E _{AD}				±8	LSB
ADC时钟频率	F _{AD}		-			MHz
ADC采样时间	T _S		-			us
总转换时间	T _{CON}		5			us
内部参考电压	VBG			2		V
				3		
				4		
内部参考电压误差			-1		1	%
内部参考电压温度特性	TC _{VBG}			60		ppm/°C

29. 开发支持

29.1 集成开发环境

Keil

29.2 在线调试编程器

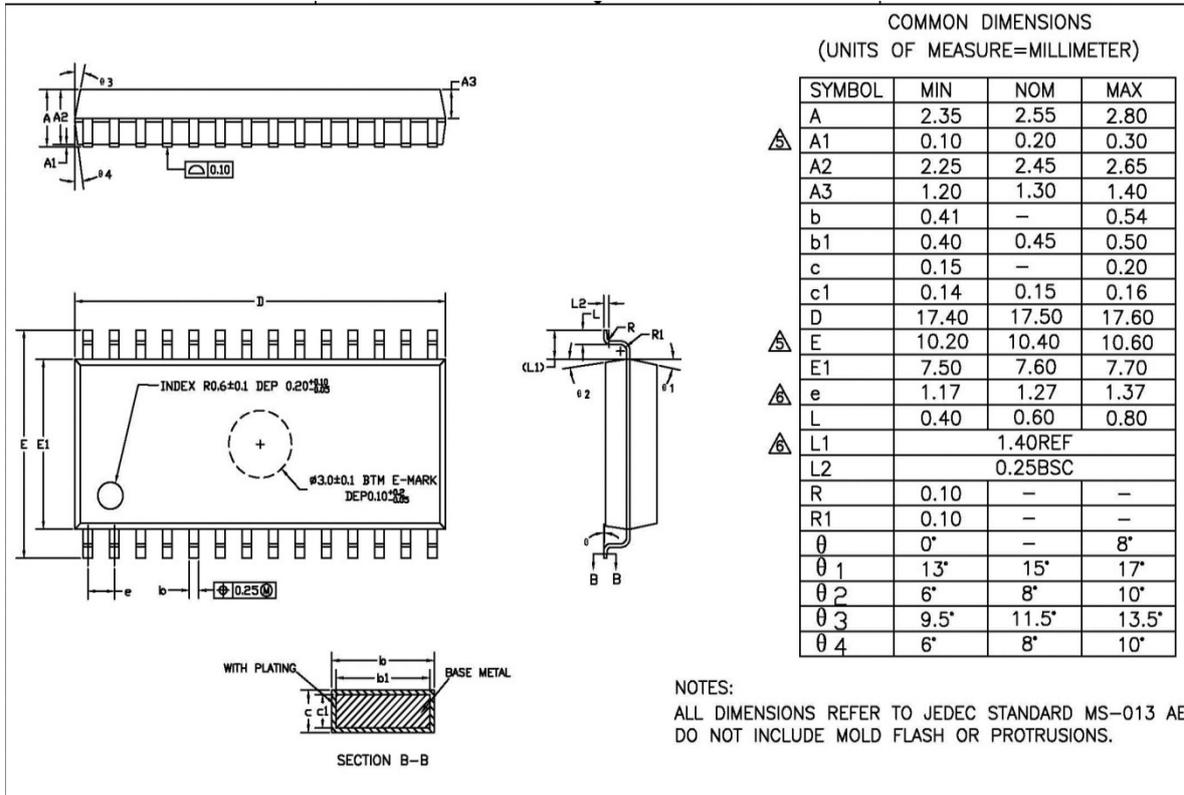
BJX-调试器 结合 Keil 功能强大的图形用户界面，BJX-调试器可对 BJ8M302A 进行调试和编程，且价位较低。BJX-调试器通过高速 USB 接口与设计工程师的 PC 相连，并利用 BJT-调试器调试连接器与目标板相连。连接器使用两个器件 I/O 引脚来实现在线调试和在线串行编程（In-Circuit Serial Programming, ICSP）。

29.3 脱机烧录器

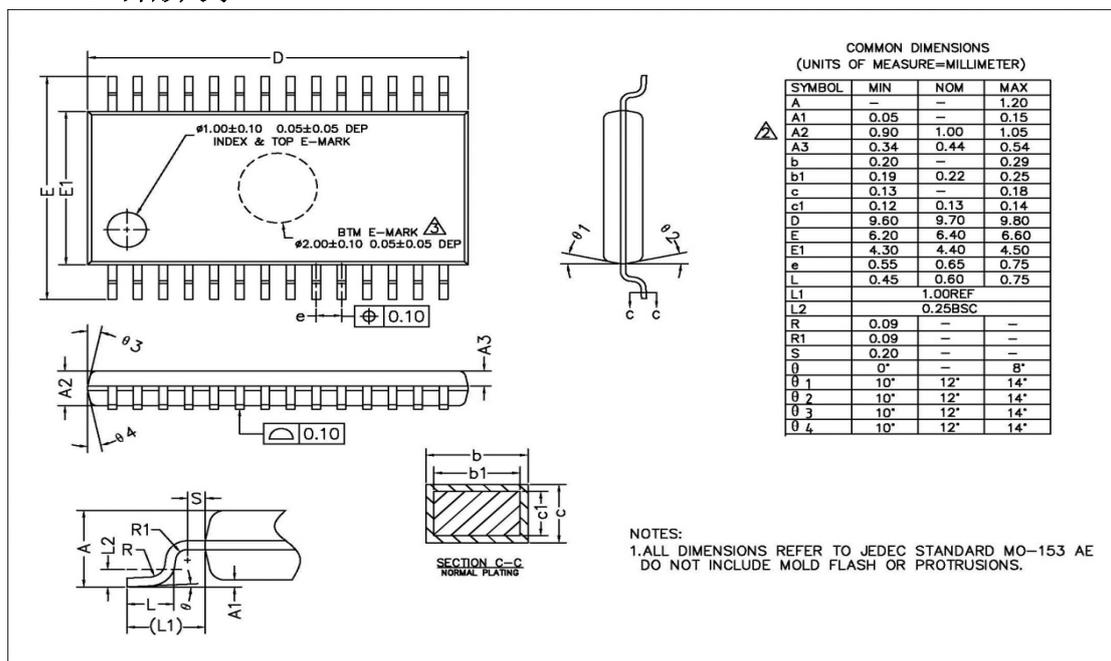
BJX-PE-DWRITERxxx (xxx:版本号) 烧录器是一款博巨兴通用单片机编程器。它有一个用来显示菜单和错误消息的大 LCD 显示器或者 LED 屏，以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSP 电缆。在单机模式下，BJX-PE-DWRITER 烧录器不必与 PC 相连即可对 BJ8M302A 器件进行读取、校验和编程。在该模式下它还可设置代码保护。BJX-PE-DWRITER 通过 USB 电缆连接到 PC 主机上。BJX-PE-DWRITER 具备高速通信能力以及优化算法，可对具有大存储器的器件进行快速编程。

30. 封装信息

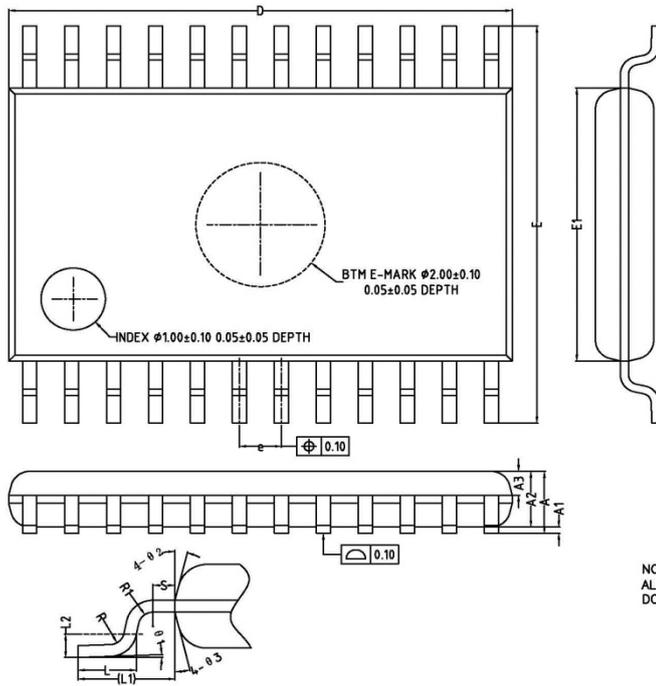
30.1 SOP28 外形尺寸



30.2 TSSOP28 外形尺寸

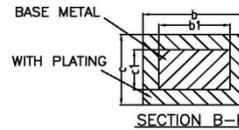


30.3 TSSOP24 外形尺寸



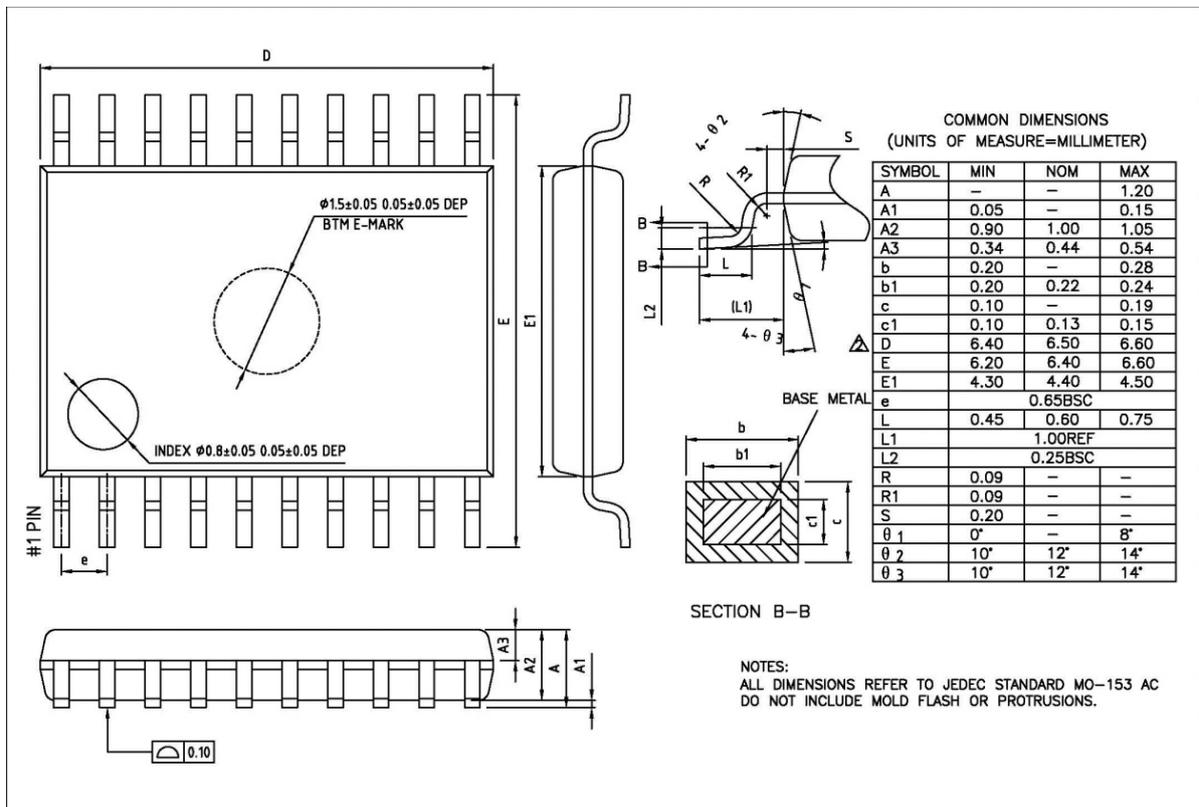
COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	—	—	1.20
A1	0.05	—	0.15
A2	0.80	0.90	1.00
A3	0.34	0.39	0.44
b	0.20	—	0.29
b1	0.19	0.22	0.25
c	0.10	—	0.19
c1	0.10	0.13	0.15
D	7.70	7.80	7.90
E	6.20	6.40	6.60
E1	4.30	4.40	4.50
e	0.55	0.65	0.75
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
R	0.09	—	—
R1	0.09	—	—
S	0.20	—	—
θ 1	0°	—	8°
θ 2	12°	14°	16°
θ 3	12°	14°	16°



NOTES:
ALL DIMENSIONS REFER TO JEDEC STANDARD MO-153 AD
DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

30.3 TSSOP20 外形尺寸



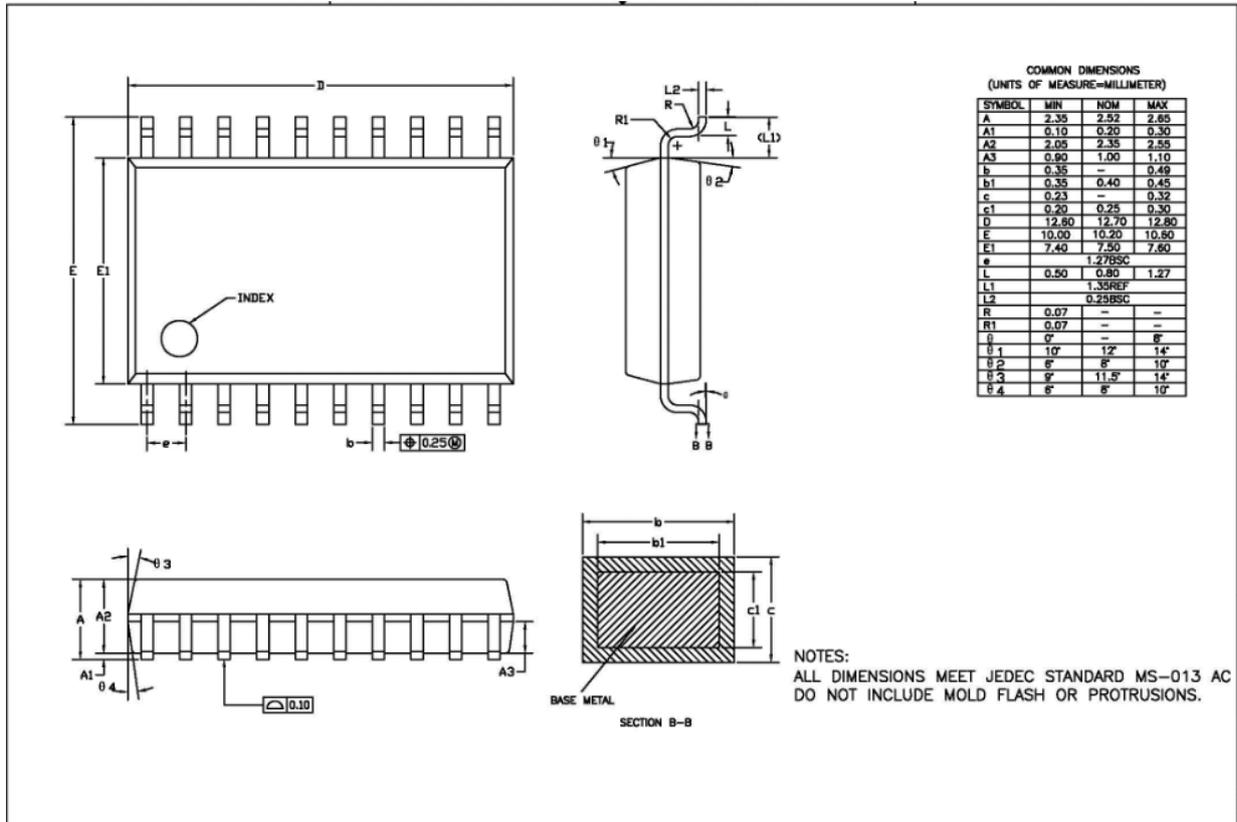
COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	—	—	1.20
A1	0.05	—	0.15
A2	0.90	1.00	1.05
A3	0.34	0.44	0.54
b	0.20	—	0.28
b1	0.20	0.22	0.24
c	0.10	—	0.19
c1	0.10	0.13	0.15
D	6.40	6.50	6.60
E	6.20	6.40	6.60
E1	4.30	4.40	4.50
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
R	0.09	—	—
R1	0.09	—	—
S	0.20	—	—
θ 1	0°	—	8°
θ 2	10°	12°	14°
θ 3	10°	12°	14°

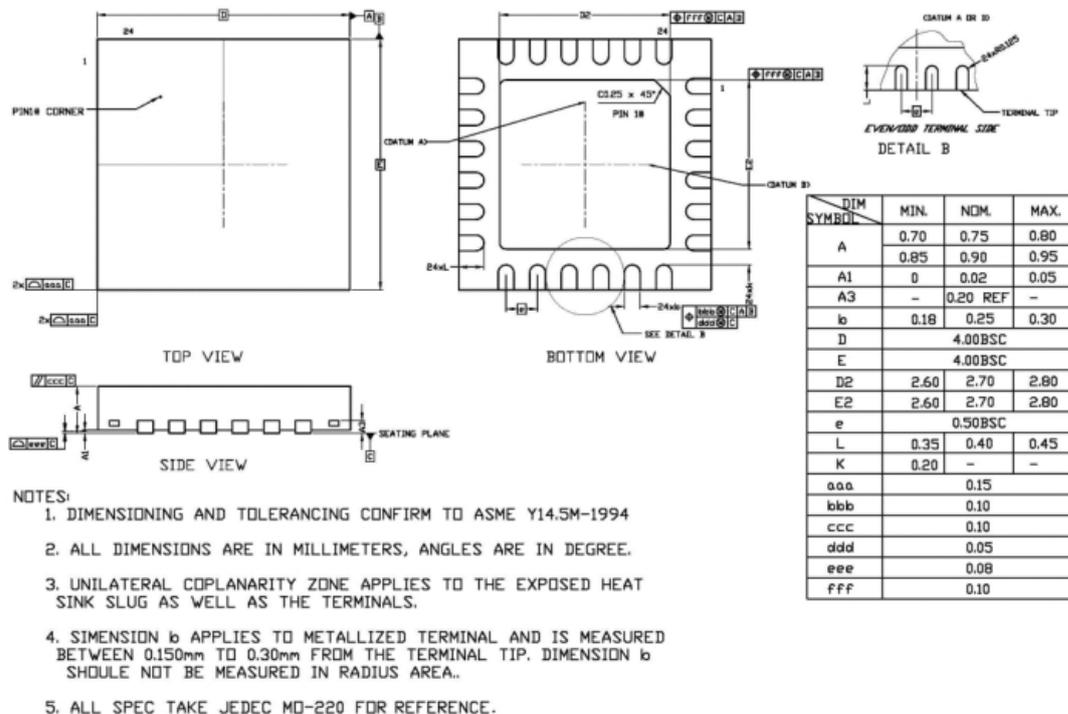
SECTION B-B

NOTES:
ALL DIMENSIONS REFER TO JEDEC STANDARD MO-153 AC
DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

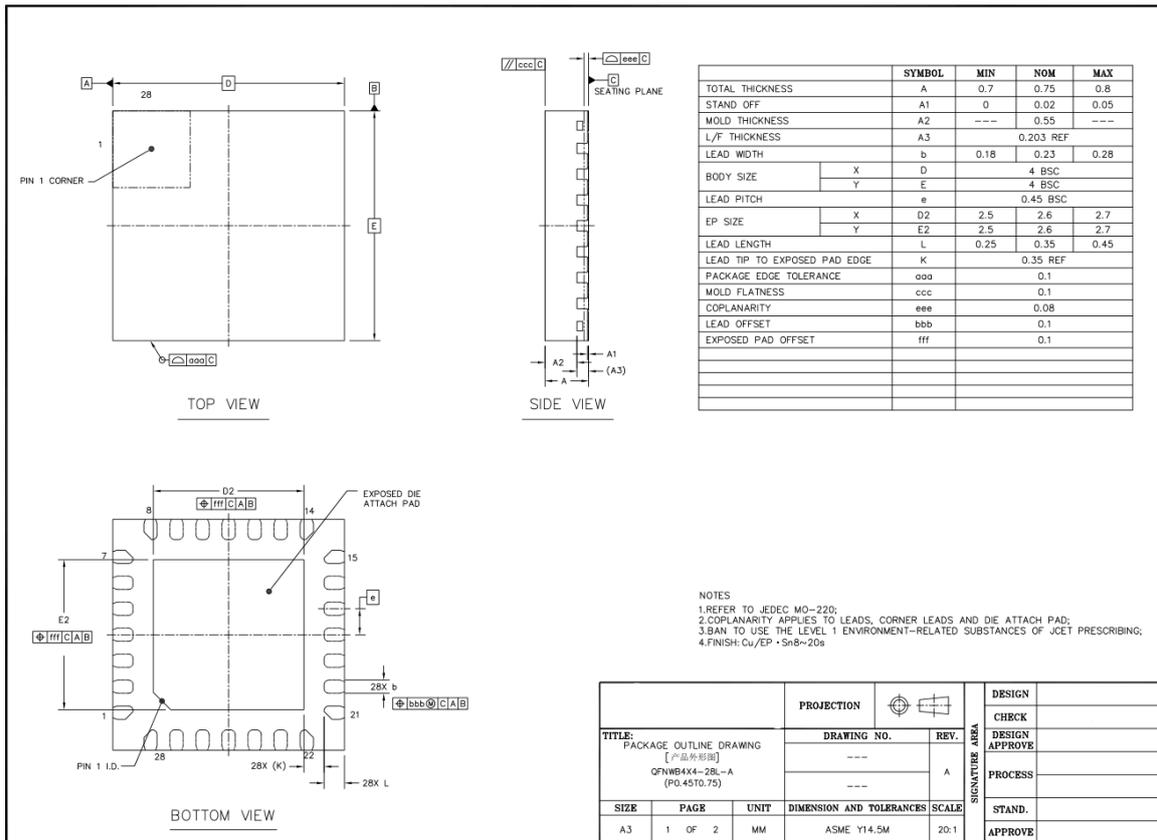
30.4 SOP20 外形尺寸



30.5 QFN24 外形尺寸



30.6 QFN28 外形尺寸



31. 产品标识体系

31.1 产品命名规则

BJ 8M 302 A X X X -X

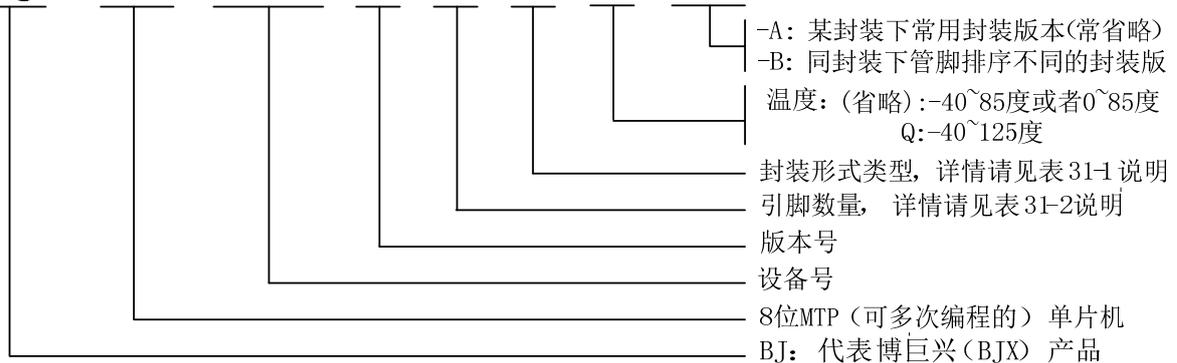


表 31-1 产品外观形态码用一个字符表示

代码	说明	代码	说明	代码	说明	代码	说明	代码	说明
G	SOT23/25	S	SSOP	N	窄距 SOP150MIL	D	DIP	L	LQFP
J	MSOP	T	TSSOP	M	宽距 SOP300MIL	K	SK-DIP	J	MSOP

表 31-2 产品封装脚位码用一个字符表示

代码	说明	代码	说明	代码	说明	代码	说明
A	6PIN	F	18PIN	K	32PIN	Q	80PIN
B	8PIN	G	20PIN	L	36PIN	R	100PIN
C	10PIN	H	24PIN	M	44PIN	S	128PIN
D	14PIN	J	28PIN	N	48PIN	T	5PIN
E	16PIN			P	64PIN	U	3PIN

31.2 完整型号一览表

Part Number	Totle Pins	Package Type	Temp Range
BJ8M302AMJ	28	SOP	-40°C to +85°C
BJ8M302ATJ	28	TSSOP	-40°C to +85°C
BJ8M302ATH	24	TSSOP	-40°C to +85°C
BJ8M302ATG	20	TSSOP	-40°C to +85°C
BJ8M302ATG-B	20	TSSOP	-40°C to +85°C
BJ8M302AMG	20	SOP	-40°C to +85°C
BJ8M302AFH	24	QFN	-40°C to +85°C
BJ8M302AFJ-B	28	QFN	-40°C to +85°C

32. 附录 A: 版本信息&声明

表 32-1 版本历史列表

REV.	修改内容	时间	修改人
1.0	First Version	2016.07.01	PM Carlry
1.1	修改笔误	2016.07.05	TE D.Y
1.2	修改工作电压为 2.0V~5.5V	2016.10.26	Pzye
1.3	增加 SOP20 和 QFN24 脚封装	2017.3.22	Pzye
1.4	修正比较器和 Timer 描述	2017.7.14	Pzye
1.5	修正 BJ8M302ATH 脚位描述	2017.8.28	Pzye
1.6	增加 QFN28 封装	2018.7.4	Pzye
1.7	修改 QFN28 封装尺寸说明	2018.7.10	Pzye
1.8	增加快充模块使用说明	2018.8.6	Pzye
1.9	修改封装管脚位置说明	2019.2.22	QY

表 32-2 声明列表

BJX ©深圳市博巨兴实业发展有限公司

BJX ©SHENZHEN BJX INDUSTRIAL DEVELOPMENT CO., LTD

使用指南中所出现的信息在出版当时相信是正确的,然而博巨兴对于说明书的使用不负任何责任。文中提到的应用目的仅仅是用来做说明,博巨兴不保证或表示这些没有进一步修改的应用将是适当的,也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成伤害甚至死亡的领域。如果将博巨兴的产品应用于上述领域,即使这些是由博巨兴在产品设计和制造上的疏忽引起的,用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接产生的律师费用,并且用户保证博巨兴及其雇员、子公司、分支机构和销售商与上述事宜无关。另外,博巨兴拥有不事先通知而修改产品的权利,对于最新的信息,请参考我们的网址 <http://www.BJXMCU.com>.

33.读者反馈表

我们努力为您提供最佳文档，以确保您能够成功使用 BIX 产品。如果您对文档的组织、条理性、主题及其他有助于提高文档质量的方面有任何意见或建议，请填写本反馈表并传真给我公司。请填写以下信息，并从下面各方面提出您对本文档的意见。

致： PM 经理 总页数 _____

关于： 读者反馈

发自： 姓名：

公司：

地址：

电话：

Email：

应用（选填）：

您希望收到回复吗？ 是 否

器件型号： 文献名称：

问题：

1. 本文档中哪些部分最有特色？
2. 您认为本文档的组织结构便于理解吗？如果不便于理解，那么问题何在？
3. 您认为本文档应该添加哪些内容以改善其结构和主题？
4. 您认为本文档中可以删减哪些内容，而又不会影响整体使用效果？
5. 本文档中是否存在错误或误导信息？如果存在，请指出是什么信息及其具体页数。
6. 您认为本文档还有哪些方面有待改进？