



JZ8M4634

数据手册

8BIT
TK+AD 型
MTP MCU
版本号V1.0
2018年5月



修改记录

版本号	修改说明	备注
V1.0	完成初稿	



目录

1 产品简述	7
1.1 特性.....	7
1.2 引脚图.....	8
1.2.1 LQFP32.....	8
1.2.2 SOP28.....	8
1.3 引脚描述.....	9
2 中央处理器（CPU）	12
2.1 程序存储器.....	12
2.1.1 复位向量（0000H）	12
2.1.2 中断向量（0008H）	13
2.1.3 查表.....	13
2.2 数据存储器.....	14
2.2.1 数据存储器结构.....	14
2.2.2 数据存储器寻址模式.....	14
2.2.3 系统寄存器定义.....	15
2.2.4 INDF0 间接寻址寄存器 0.....	15
2.2.5 INDF1 间接寻址寄存器 1.....	15
2.2.6 FSR0 间接寻址指针 0.....	15
2.2.7 FSR1 间接寻址指针 1.....	16
2.2.8 HBUF 查表数据高 8 位.....	16
2.2.9 PCL 程序计数器指针低位.....	16
2.2.10 STATUS 状态寄存器.....	16
3 复位	17
3.1 复位方式.....	17
4 系统时钟	18
4.1 概述.....	18
4.2 OSCM 寄存器.....	18
4.3 IRCCAL 寄存器.....	19
4.4 系统时钟结构框图.....	20
4.5 系统时钟高低频切换.....	20
5 中断	21
5.1 概述.....	21
5.2 OPTION 配置寄存器.....	21
5.3 INTCR0 中断控制寄存器 0.....	22
5.4 INTF0 中断标志寄存器 0.....	22
5.5 INTCR1 中断控制寄存器 1.....	23
5.6 INTF1 中断标志寄存器 1.....	24
5.7 INTCR2 中断控制寄存器 2.....	25
5.8 INTF2 中断标志寄存器 2.....	25



6 端口	26
6.1 IOA.....	26
6.2 IOB.....	27
6.3 IOC.....	28
6.4 IOD.....	29
6.5 IODS 端口驱动设置寄存器.....	30
7 定时器 0/1(TC0/1)	31
7.1 概述.....	31
7.2 TxCR 控制寄存器.....	31
7.3 TCxCL TCx 计数器低位.....	32
7.4 TCxCH TCx 计数器高位.....	32
7.5 TCxPRL TCx 周期寄存器低位.....	33
7.6 TCxPRH TCx 周期寄存器高位.....	33
8 定时器 2 (TC2)	34
8.1 概述.....	34
8.2 T2CR 控制寄存器.....	35
8.3 TC2CL 计数器低位.....	35
8.4 TC2CH 计数器高位.....	36
8.5 TC2CL 周期寄存器低位.....	36
8.6 TC2CH 周期寄存器高位.....	36
8.7 TC2GCR 门控制寄存器.....	36
8.7.1 门控-TC0 溢出周期.....	37
8.7.2 门控-上升沿到下降沿模式.....	37
8.7.3 门控-下降沿到上升沿模式.....	37
8.7.4 门控-上升沿到上升沿模式.....	38
8.7.5 门控-下降沿到下降沿模式.....	38
9 脉宽调制模块 PWM	39
9.1 概述.....	39
9.2 PWMxCR 控制寄存器.....	39
9.3 PWMS 输出端口控制寄存器.....	40
9.4 PWMxDH 数据高位.....	40
9.5 PWMxDL 数据低位.....	40
9.6 DEADT PWM 死区控制寄存器.....	40
9.7 PWM 输出波形示例.....	41
9.7.1 互补 PWM 输出.....	41
9.7.2 带死区的互补 PWM 输出.....	41
9.7.3 PWM 波形图.....	42
9.7.4 有效电平选择.....	42
10 通用串行通讯口 (USART)	43
10.1 概述.....	43
10.2 TX0CR 发送控制寄存器.....	43
10.3 TX0REG 发送数据寄存器.....	43



10.4 RX0CR 接收控制寄存器.....	44
10.5 RX0REG 发送数据寄存器.....	44
10.6 BRGDH 波特率寄存器高位.....	44
10.7 BRGDH 波特率寄存器低位.....	45
10.8 USART 使用说明.....	45
10.8.1 波特率设置.....	45
10.8.2 异步发送.....	45
10.8.3 异步接收.....	47
10.8.4 同步发送.....	48
10.8.5 同步接收.....	49
10.8.6 唤醒及休眠模式下通讯.....	50
11 串行通讯口 (I2C)	51
11.1 概述.....	51
11.2 通讯波形示意.....	51
11.3 I2CCON I2C 控制寄存器.....	51
11.4 I2CBUF 数据寄存器.....	52
11.5 唤醒及休眠模式下通讯.....	52
11.6 通讯波形图.....	52
11.7 应用示例.....	53
11.7.1 从机软件流程图.....	54
11.7.1 例程.....	54
12 触摸按键 (CDC)	57
12.1 概述.....	57
12.2 原理框图.....	57
12.3 TKXCTR0 控制寄存器 0.....	58
12.3 TKXCHSH/L 触摸按键通道选择寄存器.....	58
12.4 TKXCNTNTH/L 触摸按键计数寄存器.....	59
12.5 操作说明.....	59
13 模数转换器(ADC).....	60
13.1 概述.....	60
13.2 ADCON0 寄存器.....	60
13.3 ADCON1 寄存器.....	61
13.4 ADCON2 寄存器.....	62
13.5 ADH ADC 数据高位.....	62
13.6 ADL ADC 数据低位.....	62
13.7 AD 转换时间.....	62
14 比较器(CMP).....	63
14.1 概述.....	63
14.2 比较器框图.....	63
14.3 CMPC0 比较器控制寄存器 0.....	64
14.4 CMPC1 比较器控制寄存器 1.....	65
14.5 COPAC 运放/比较器控制寄存器.....	65
15 运算放大器(OPA).....	66



15.1 概述.....	66
15.2 OPA0C0 运放 OPA0 控制寄存器 0.....	66
15.3 OPA0C1 运放 OPA0 控制寄存器 1.....	67
15.4 COPA0C 运放 OPA0 控制寄存器 2.....	67
15.5 OPA1C0 运放 OPA1 控制寄存器 0.....	67
15.6 OPA1C1 运放 OPA1 控制寄存器 1.....	68
15.7 COPA1C 运放 OPA0 控制寄存器 2.....	69
16 看门狗 (WDT)	70
16.1 概述.....	70
16.2 OPTION 配置寄存器.....	70
16.3 WDTC 看门狗控制寄存器.....	70
17 芯片配置字 (OPTION BIT)	71
18 电性参数.....	73
18.1 极限参数.....	73
18.2 直流特性.....	73



1 产品简述

JZ8M4634 是一颗采用高速低功耗 CMOS 工艺设计开发的 8 位高性能精简指令单片机，内部有 4K×16 位多次擦写编程存储器（MTP，擦写次数 1000），256×8 位的数据存储器（RAM），30 个双向 I/O 口，三个 8/16 位 Timer 定时器/计数器，1 路 UART，1 路 I2C 从机，3 路 8+4 位分辨率的互补 PWM，11+5 路 12 位 AD 转换器，2 组触摸按键，1 路比较器，2 路运算放大器，支持多种系统工作模式和多个中断源。

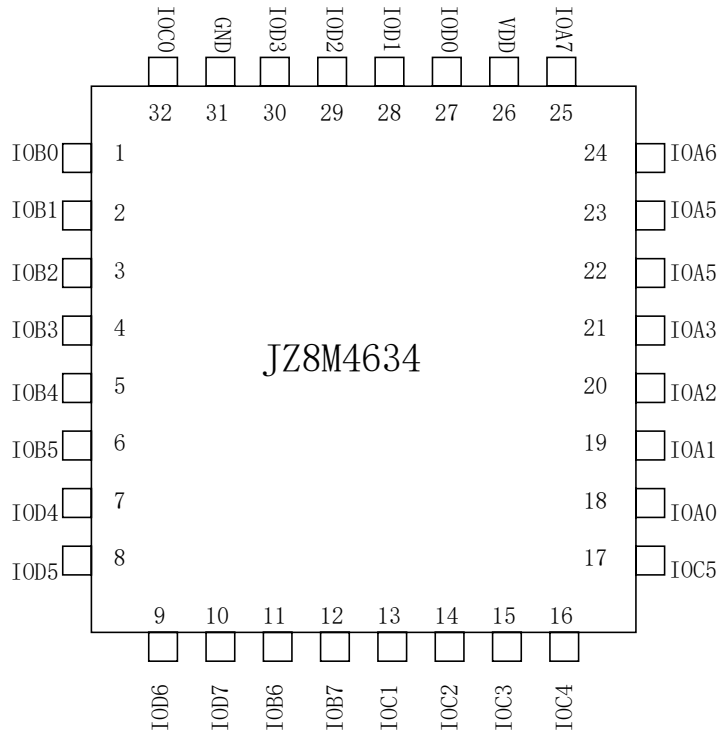
1.1 特性

- CPU 特性
 - 高性能精简指令
 - 4K×16位的MTP程序存储器
 - 256×8位的数据存储器
 - 8级堆栈缓存器
 - 支持查表指令
- I/O 口
 - 最多30个双向I/O口
 - 所有端口可编程弱上拉/弱下拉
 - IOB/IOC/IOD口变化中断
- 三个定时器/计数器
 - TC0: 具有自动装载功能的定时/计数器
 - TC1: 具有自动装载功能的定时/计数器
 - TC2: 带有门控功能的定时/计数器
- 三路 PWM
 - 时基可独立选择TC0、TC1或TC2
 - 互补输出及死区控制
 - 16位高分辨率
- 系统时钟
 - 内部高速RC振荡器： 16MHz
 - 内部低速RC振荡器： 32KHz/500KHz
 - 外部高速晶体振荡器： 4-20MHz
 - 外部低速晶体振荡器： 32768Hz
- 系统工作模式
 - 普通模式： 高低速时钟同时工作
 - 低速模式： 仅低速时钟工作
 - 休眠模式： 高低速时钟都停止工作
- 通用异步串行通讯口 UART
 - 宽范围波特率（150bps-38400bps）
 - 支持半双工同步模式
- 芯片间通讯口 I2C 从机
 - 高速通讯400Kbps
- 两组触摸按键模块（11 路+16 路）
- 1 路比较器
- 2 路运算放大器
- 11+5 路 12 位 ADC
 - 内嵌参考电压2V、3V、4V、VDD
 - 11路外部输入
 - 1路内部电源电压检测VDD/4
 - 1路内部GND电压检测
 - 1路内部参考电压检测
 - 2路内部运放输出检测
- 多路中断源
 - 定时器中断：TC0/TC1/TC2，TC2门控
 - UART发送中断、UART接收中断
 - IOA6/IOA7外部中断
 - IOB/IOC/IOD口中断
 - 触摸按键中断
 - ADC转换中断
 - 比较器中断
 - 运放中断
- 看门狗定时器
- 特殊功能
 - 可编程代码保护
 - ISP功能
- 封装形式
 - LQFP32/SOP28

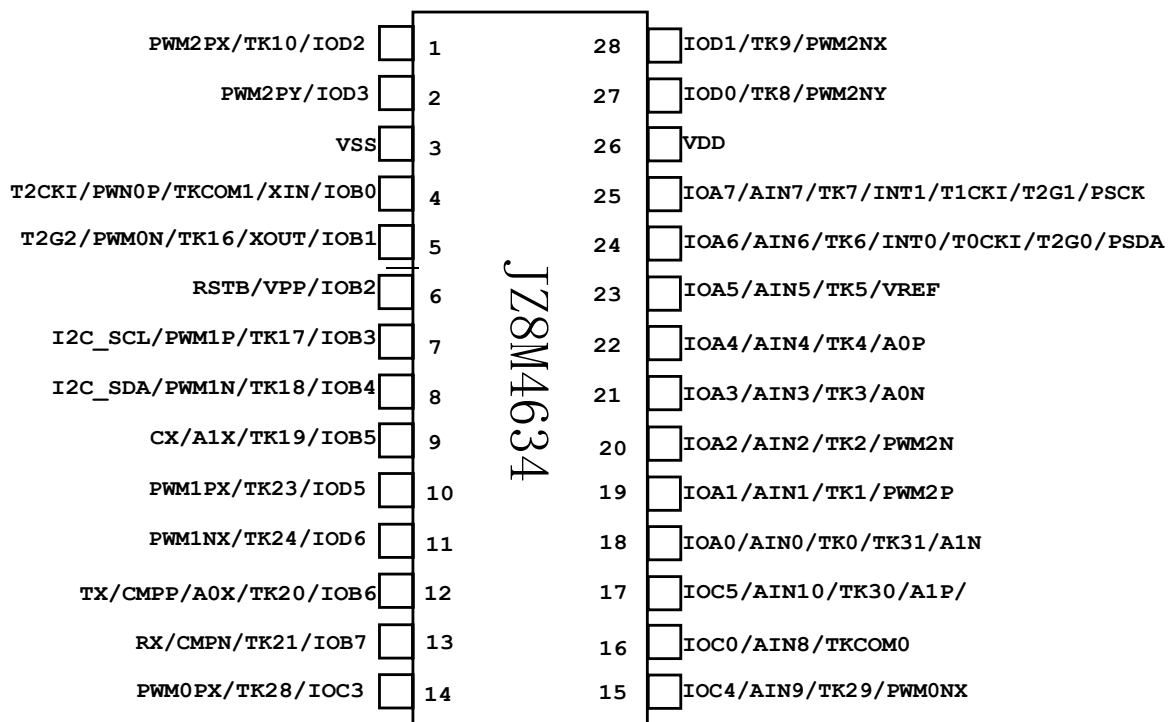


1.2 引脚图

1.2.1 LQFP32



1.2.2 SOP28





1.3 引脚描述

名称	类型	说明
VDD, VSS	P	电源输入端
IOA0	I/O	输入/输出 IO, SMT, 上拉电阻
AIN0	A	AD 通道 0
TK0	A	触摸按键通道 0
TK31	A	触摸按键通道 31
A1N	A	运算放大器 OPA1 负端输入
IOA1	I/O	输入/输出 IO, SMT, 上拉电阻
AIN1	A	AD 通道 1
TK1	A	触摸按键通道 1
PWM2P	O	PWM2 正相输出
IOA2	I/O	输入/输出 IO, SMT, 上拉电阻
AIN2	A	AD 通道 2
TK2	A	触摸按键通道 2
PWM2N	O	PWM2 反相输出
IOA3	I/O	输入/输出 IO, SMT, 上拉电阻
AIN3	A	AD 通道 3
TK3	A	触摸按键通道 3
A0N	A	运算放大器 OPA0 负端输入
IOA4	I/O	输入/输出 IO, SMT, 上拉电阻
AIN4	A	AD 通道 4
TK4	A	触摸按键通道 4
A0P	A	运算放大器 OPA0 正端输入
IOA5	I/O	输入/输出 IO, SMT, 上拉电阻
AIN5	A	AD 通道 5
TK5	A	触摸按键通道 5
VREF	A	AD 外部参考电压输入
IOA6	I/O	输入/输出 IO, SMT, 上拉电阻
AIN6	A	AD 通道 6
TK6	A	触摸按键通道 6
INT0	I	外部中断
T0CKI	I	TC0 外部时钟输入
T2G0	I	TC2 门控信号输入
PSDA	I/O	编程用
IOA7	I/O	输入/输出 IO, SMT, 上拉电阻
AIN7	A	AD 通道
TK7	A	触摸按键通道 7
INT1	I	外部中断
T1CKI	I	TC1 外部时钟输入
T2G1	I	TC2 门控信号输入



PSCK	I/O	编程用
IOB0	I/O	输入/输出 IO, SMT, 上拉电阻
XIN	A	外部晶体振荡器接口
TKCOM1	A	触摸按键灵敏度电容口
PWM0P	O	PWM0 正相输出
T2CKI	O	TC2 外部时钟输入
IOB1	I/O	输入/输出 IO, SMT, 上拉电阻
XOUT	A	外部晶体振荡器接口
TK16	A	触摸按键通道 16
PWM0N	O	PWM0 反相输出
T2G2	I	TC2 门控信号输入
IOB2	I/O	输入/输出 IO, SMT, 上拉电阻
RSTB	I	外部复位输入, 上拉电阻
VPP	P	编程高压电源
IOB3	I/O	输入/输出 IO, SMT, 上拉电阻
PWM1P	O	PWM1 正相输出
TK17	A	触摸按键通道 17
I2C SCL	O	I2C 通讯时钟口
IOB4	I/O	输入/输出 IO, SMT, 上拉电阻
PWM1N	O	PWM1 反相输出
TK18	A	触摸按键通道 18
I2C SDA	I/O	I2C 通讯数据口
IOB5	I/O	输入/输出 IO, SMT, 上拉电阻
A1X	A	运算放大器 OPA1 输出
TK19	A	触摸按键通道 19
CX	O	比较器 CMP 输出
IOB6	I/O	输入/输出 IO, SMT, 上拉电阻
CMPP	A	比较器 CMP 正端输入
A0X	A	运算放大器 OPA0 输出
TK20	A	触摸按键通道 20
TX	O	通用异步串行通讯发送口
IOB7	I/O	输入/输出 IO, SMT, 上拉电阻
CMPN	A	比较器 CMP 负端输入
TK21	A	触摸按键通道 21
RX	I	通用异步串行通讯接收口
IOC0	I/O	输入/输出 IO, SMT, 上拉电阻
AIN8	A	AD 通道 8
TKCOM0	A	触摸按键灵敏度电容口
IOC1	I/O	输入/输出 IO, SMT, 上拉电阻
TK26	A	触摸按键通道 26
IOC2	I/O	输入/输出 IO, SMT, 上拉电阻
TK27	A	触摸按键通道 27
IOC3	I/O	输入/输出 IO, SMT, 上拉电阻



TK28 PWM0NX	A O	触摸按键通道 28 PWM0 正相输出
IOC4 AIN9 TK29 PWM0NX	I/O A A O	输入/输出 IO, SMT, 上拉电阻 AD 通道 9 触摸按键通道 29 PWM0 反相输出
IOC5 AIN10 TK30	I/O A A	输入/输出 IO, SMT, 上拉电阻 AD 通道 10 触摸按键通道 30
IOD0 TK8 PWM2NY	I/O A O	输入/输出 IO, SMT, 上拉电阻 触摸按键通道 8 PWM2 反相输出
IOD1 TK9 PWM2NX	I/O A O	输入/输出 IO, SMT, 上拉电阻 触摸按键通道 9 PWM2 反相输出
IOD2 TK10 PWM2PX	I/O A O	输入/输出 IO, SMT, 上拉电阻 触摸按键通道 10 PWM2 正相输出
IOD3 PWM2PY	I/O O	输入/输出 IO, SMT, 上拉电阻 PWM2 正相输出
IOD4 TK22 PWM1PY	I/O A O	输入/输出 IO, SMT, 上拉电阻 触摸按键通道 22 PWM1 正相输出
IOD5 TK23 PWM1PX	I/O A O	输入/输出 IO, SMT, 上拉电阻 触摸按键通道 13 PWM1 正相输出
IOD6 TK24 PWM1NX	I/O A O	输入/输出 IO, SMT, 上拉电阻 触摸按键通道 14 PWM1 反相输出
IOD7 TK25 PWM1NY	I/O A O	输入/输出 IO, SMT, 上拉电阻 触摸按键通道 15 PWM1 反相输出

注：I= 输入 O= 输出 I/O= 输入/ 输出 P= 电源 A= 模拟端口



2 中央处理器（CPU）

2.1 程序存储器

地址	说明
0x0000	复位向量
0x0001 ~ 0x0007	用户区
0x0008	中断向量
0x0009 ~ 0x0FEF	用户区
0x0FF0 ~ 0x0FFF	厂商保留区

2.1.1 复位向量（0000H）

JZ8M4634有以下四种复位方式

- 上电复位
- 看门狗复位
- 外部复位
- 欠压复位

发生上述任一种复位后，程序将从 0000H处重新开始执行，系统寄存器也将都恢复为初始默认值。

例：定义复位向量

```
ORG    0000H    ;
GOTO  MAIN    ; 跳转至用户程序开始
...
MAIN: ...      ; 用户程序开始
...
GOTO  MAIN    ; 用户主程序循环
```



2.1.2 中断向量（0008H）

JZ8M4634中断向量地址为0008H。一旦有中断响应，程序计数器PC的当前值就会存入堆栈缓存器并跳转到0008H处开始执行中断服务程序。

例：中断服务程序：

```

        ORG      0000H
        GOTO     START      ; 跳转到程序开始
        ...
        ORG      0008H
        GOTO     IRQSUB     ; 发生中断后，跳转到中断子程序

START:
        ...
        ...
        GOTO     START      ; 主程序循环

IRQSUB:
        PUSH                     ; 进入中断子程序后，先保存现场
        ...
        POP                      ; 退出中断子程序前，恢复现场
        RETIE

        END

```

2.1.3 查表

利用RDT指令可以读取程序区数据，其中读到的16位数据高位放在HBUF中，低位放在A寄存器中；FSR1的低3位和FSR0组成11位程序区数据寻址指针。

例：查找 ROM 地址为“DTAB”的值

	MOVIA	HIGH(DTAB)	;获取数据表地址高位
	MOVAR	FSR1	;设置数据表高位指针
	MOVIA	LOW(DTAB)	;获取数据表地址低位
	MOVAR	FSR0	;设置数据表低位指针
			;若需读取表的其它数据，修改指针
	RDT		;读取表的第一个数据0x0102
	MOVAR	TABDL	;将低位数据0x02放在TABDL
	MOVR	HBUF,A	;高位数据读入累加器A
	MOVAR	TABDH	;将高位数据0x01放在TABDH
			”
DTAB:			
	DW	0x0102	
	DW	0x1112	
			”



2.2 数据存储器

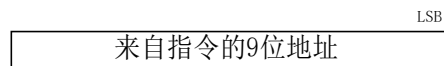
2.2.1 数据存储器结构

数据寄存器分为两个256字节的块区，共512字节，地址0x000~0x1FF。其中0x00~0x0FF为通用寄存器（RAM），0x100~0x1FF分配给特殊功能寄存器。通用寄存器可通过直接寻址或INDF0、INDF2间接寻址；特殊功能寄存器可通过直接寻址或INDF1、INDF2间接寻址。具体地址分配参照下表。

地址	区域	间接寻址 INDF0	间接寻址 INDF1	间接寻址 INDF2	直接寻址
0x100 ~ 0x1FF	SFR	NO	YES	YES	YES
0x000 ~ 0x0FF	GPR	YES	NO		

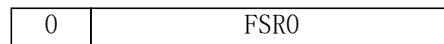
2.2.2 数据存储器寻址模式

9位数据寄存器地址组成



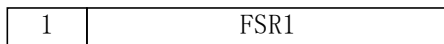
直接寻址模式

例: `MOVAR 0x155` ;把A寄存器内容写入0x55地址



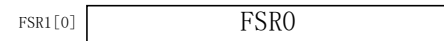
间接寻址模式0

例: `MOVAR INDF0` ;把A寄存器内容写入FSR0指向寄存器



间接寻址模式1

例: `MOVAR INDF1` ;把A寄存器内容写入FSR1指向寄存器



间接寻址模式2

例: `MOVAR INDF2` ;把A寄存器内容写入FSR1/FSR0指向寄存器



2.2.3 系统寄存器定义

数据寄存器映射表								
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0x000 ~ 0x0F0	GPR (256x8)							
0x100 ~ 0x178	RESERVE							
0x180								
0x188								
0x190								
0x198	TC0PRL	TC0PRH	TC1PRL	TC1PRH	TC2PRL	TC2PRH	TK1CNTL	TK1CNTH
0x1A0	TK1CTR0	TK1CHSL	TK1CHSH	TK0CNTL	TK0CNTH	TK0CTR0	TK0CHSL	TK0CHSH
0x1A8	OPA1C0	OPA1C1	COPA1C	CMPC0	CMPC1	OPA0C0	OPA0C1	COPA0C
0x1B0	INDF0	FSR0	TX0CR	TX0REG	RX0CR	RX0REG	BRGDH	BRGDL
0x1B8	INDF1	FSR1	PCL	STATUS	OPTION	OSCM	WDTC	ADCON2
0x1C0	INDF2	HBUF	INTCR2	INTF2	INTCR0	INTF0	INTCR1	INTF1
0x1C8	IOA	OEA	PUA	PDA	IOB	OEB	PUB	PDB
0x1D0	IOC	OEC	PUC	PDC	IOD	OED	PUD	PDD
0x1D8	ANSA	ANSC	IOBICR	IOCICR	IODICR	PWM2CR	PWM2DH	PWM2DL
0x1E0	PWM0CR	PWM0DH	PWM0DL	PWMDEAT	PWM1CR	PWM1DH	PWM1DL	I2CCON
0x1E8	T0CR	TC0CL	TC0CH	IODS	T1CR	TC1CL	TC1CH	I2CBUF
0x1F0	T2CR	TC2CL	TC2CH	T2GCR	ADCON0	ADCON1	ADL	ADH
0x1F8	PWMS				IRCCAL			

(注：GPR 为通用寄存器)

2.2.4 INDF0 间接寻址寄存器 0

访问INDF0寄存器时，实现间接寻址模式0，访问到的是FSR0寄存器所指向的寄存器内容，间接寻址模式0仅可寻址通用寄存器区0x0000~0x00FF空间

2.2.5 INDF1 间接寻址寄存器 1

访问INDF1寄存器时，实现间接寻址模式1，访问到的是FSR1寄存器所指向的寄存器内容，间接寻址模式1仅可寻址通用寄存器区0x0100~0x01FF空间

2.2.6 FSR0 间接寻址指针 0

利用间接寻址模式0访问通用寄存器时，FSR0为地址指针；当以间接寻址模式2访问通用寄存器时，FSR0作为地址指针的低位



2.2.7 FSR1 间接寻址指针 1

利用间接寻址模式1访问通用寄存器时，FSR1为地址指针；当以间接寻址模式2访问通用寄存器时，FSR1作为地址指针的高位

2.2.8 HBUF 查表数据高 8 位

利用RDT指令读取程序区数据时，读到的16位数据高8位放在HBUF中

2.2.9 PCL 程序计数器指针低位

1BAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCL	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	1	0	0	0

Bit[7:0] **PCL[7:0]:** 程序计数器指针低位.

用户将该PCL作为目的操作数做加法运算时（ADDRA PCL、ADCRA PCL），13位PC值参与运算，运算结果写入PC，实现程序的相对跳转；加法运算外的其它运算时，仅PCL参与运算，PCH保持不变。PCH不可寻址。

2.2.10 STATUS 状态寄存器

1BBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	-	-	-	-	-	Z	DC	C
读/写	-	-	-	-	-	R/W	R/W	R/W
复位后	-	-	-	-	-	X	X	X

Bit 2 **Z:** 零标志

1 = 算术/逻辑运算的结果为零;

0 = 算术/逻辑运算的结果非零.

Bit 1 **DC:** 辅助进位标志

1 = 加法运算时低四位有进位，或减法运算后没有向高四位借位;

0 = 加法运算时低四位没有进位，或减法运算后有向高四位借位.

Bit 0 **C:** 进位标志

1 = 加法运算后有进位、减法运算没有借位发生或移位后移出逻辑“1”;

0 = 加法运算后没有进位、减法运算有借位发生或移位后移出逻辑“0”.



3 复位

3.1 复位方式

- 上电复位 (POR)
- 外部复位 (MCLR Reset)
- 欠压复位 (BOR)
- 看门狗定时器复位 (WDT Reset)

JZ8M4634 有以上4种复位方式，任何一种复位都会使PC程序计数器清零，让程序从0000H处开始运行，并且使系统寄存器值复位。



4 系统时钟

4.1 概述

JZ8M4634支持双时钟系统: 高速时钟和低速时钟。高速时钟由外部晶体振荡器或内置的16MHz RC振荡电路 (IHRC 16MHz) 提供, 低速时钟由低速晶体振荡器 (32768Hz) 或内置低速RC振荡电路 (ILRC 64KHz/500KHz) 提供。两种时钟都可作为系统时钟源Fosc, 系统工作在低速模式时, Fosc 2分频后为一个指令周期。低频系统时钟源和高频系统源可根据芯片配置字进行配置。

注: 工作时勿在进行高低频切换同时 STOP CPU 操作, 可能会造成系统紊乱。

4.2 OSCM 寄存器

工作模式控制寄存器 OSCM

1BDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCM	STBH	STBL	-	STOP	CLKM	STPH	LIRC500K	STPL
读/写	RW	RW	-	RW	RW	RW	RW	RW
复位后	x	x	-	0	x	1	0	1

- Bit 7 **STBH:** 高频振荡器稳定标志
- Bit 6 **STBL:** 低频振荡器稳定标志
- Bit 4 **STOP:** CPU工作状态标志位
1 = CPU停止工作
0 = CPU正常工作, 所有复位唤醒
- Bit 3 **CLKM:** 系统时钟状态标志位
1 = CPU运行于低频时钟
0 = CPU运行于高频时钟
- Bit 2 **STPH:** 高频振荡器控制
1 = 休眠状态或低速模式下关闭高频振荡器
0 = 休眠状态或低速模式下高速振荡器仍然工作
- Bit 1 **LIRC500K:** 低频振荡器频率选择
1 = 内部低频振荡器频率500KHz
0 = 内部低频振荡器频率32KHz
- Bit 0 **STPL:** 低频振荡器控制
1 = 休眠状态下低频振荡器停止工作
0 = 休眠状态下低频振荡器仍然工作

注: CLKM 的初始状态由配置字 SPDUP 决定。



4.3 IRCCAL 寄存器

1FCH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCCAL	-	IRCAL6	IRCCAL5	IRCCAL4	IRCCAL3	IRCCAL2	IRCCAL1	IRCCAL0
读/写	-	RW	RW	RW	RW	RW	RW	RW
复位后	-	X	X	X	X	X	X	X

内置的高频 RC 振荡电路在芯片上电后频率为校准过的 16MHz，但程序中可以通过特殊的流程来调整此频率以满足特定应用需求。

例: 调整 IRC 频率

TASK_IRCCAL:

```

MOVIA    0x55
MOVAR    0x1F9           ;//1F9H地址写入055H
MOVIA    0xAA
MOVAR    0x1F9           ;//1F9H地址写入0AAH
MOVIA    VALUE
MOVAR    IRCCAL         ;//写入IRCCAL

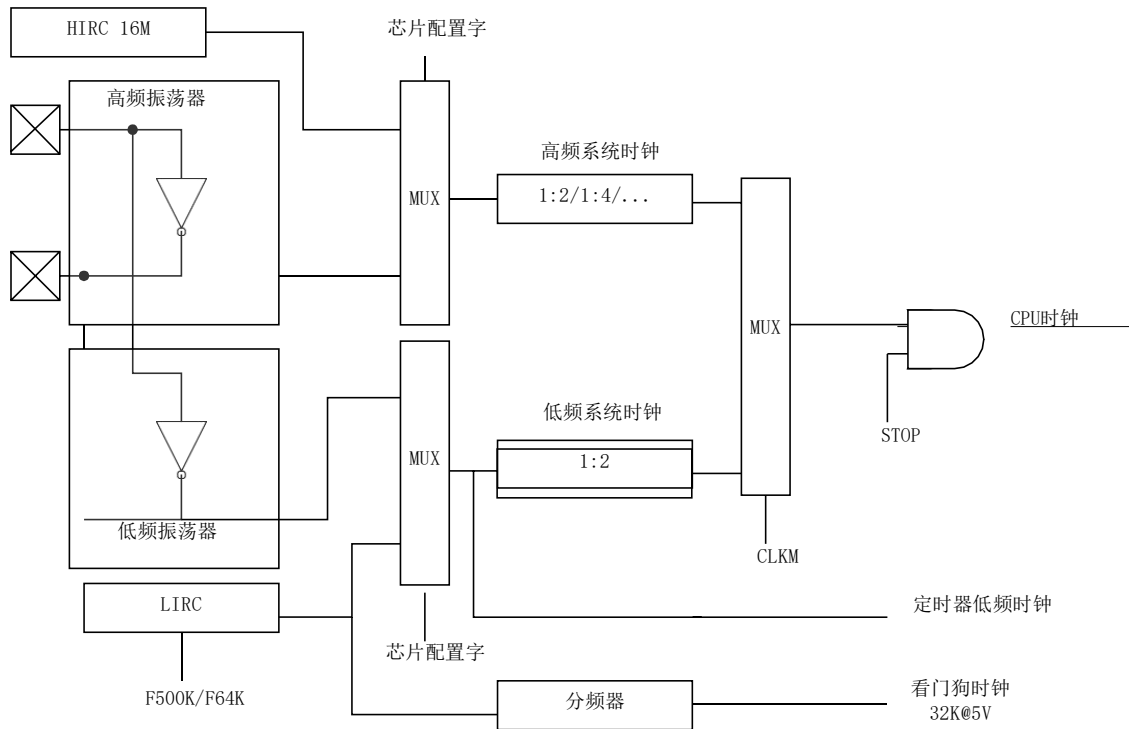
```

”

;//若需继续在IRCCAL寄存器内写入其他值需要重复以上所有步骤



4.4 系统时钟结构框图



	高速运行模式 (CLKM=0)	低速运行模式 (CLKM=1)	休眠模式 (STOP=1)
高频振荡器	运行	由 STPH 决定	由 STPH 决定
低频振荡器	运行	运行	由 STPL 决定
WDT	配置字决定	由配置字决定	由配置字决定
Timer0/Timer1	TXEN	若选择高速时钟，需 STPH=0	高速时钟源&STPH=0 或低 速时钟源&STPL=0

4.5 系统时钟高低频切换

高频振荡器稳定计数器: 64 Clocks (内部 IRC 模式) / 1024 Clocks (外部高频振荡器模式)

低频振荡器稳定计数器: 16 Clocks (内部 RC 模式) / 1024 Clocks (外部低频振荡器模式)

高低频切换时间

高频切低频: 1 个低频时钟周期+1 个高频时钟周期

低频切高频&STBH=0: 1 个低频时钟周期+高频振荡器起振时间+高频振荡器稳定时间

低频切高频&STBH=1: 1 个低频时钟周期+1 个高频时钟周期

唤醒时间:

CLKM=0&STBH=0: 高频振荡器起振时间+高频振荡器稳定时间

CLKM=0&STBH=1: 64 Clocks

CLKM=1&STBL=0: 低频振荡器起振时间+低频振荡器稳定时间

CLKM=1&STBL=1: 16 Clocks



5 中断

5.1 概述

JZ8M4634有多路中断源: TC0/TC1/TC2, TC2门控, IO变化中断, UART发送/接收中断。中断可以将系统从睡眠模式中唤醒, 在唤醒前, 中断请求被锁定。一旦程序进入中断, 寄存器OPTION的位GIE被硬件自动清零以避免响应其它中断。系统退出中断后, 硬件自动将GIE置“1”, 以响应下一个中断。

设置 GIE 和中断控制寄存器 INTCR0/INTCR1/INTCR2 来使能中断, 查询 INTF0/INTF1/INTF2 中断标志寄存器判断中断是否发生。

5.2 OPTION 配置寄存器

1BCH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPTION	GIE	-	TO	PD	MINT11	MINT10	MINT01	MINT00
读/写	R/W	-	R	R	R/W	R/W	R/W	R/W
复位后	0	-	1	1	0	0	0	0

Bit.7 **GIE:** 全局中断控制位
 1 = 总中断使能 (RETIE指令会将该位置1)
 0 = 屏蔽所有中断 (响应中断后自动清零)

Bit[3:2] **MINT1[1:0]:** INT1中断模式选择

MINT1[1:0]	INT1 中断模式选择
00	上升沿中断
01	下降沿中断
1X	变化中断

Bit[1:0] **MINT0[1:0]:** INTO中断模式选择

MINT0[1:0]	INT0 中断模式选择
00	上升沿中断
01	下降沿中断
1X	变化中断



5.3 INTCR0 中断控制寄存器 0

1C4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCR0	I2CIE	ADIE	RXIE	TXIE	TC2GIE	TC2IE	TC1IE	TC0IE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

- Bit 7 **I2CIE:**
 1 = 使能I2C中断
 0 = 屏蔽I2C中断
- Bit 6 **ADIE:**
 1 = 使能ADC转换中断
 0 = 屏蔽ADC转换中断
- Bit.5 **RXIE:**
 1 = 使能串行通讯接收中断
 0 = 屏蔽串行通讯接收中断
- Bit.4 **TXIE:**
 1 = 使能串行通讯发送中断
 0 = 屏蔽串行通讯发送中断
- Bit.3 **TC2GIE:**
 1 = 使能TC2门控中断
 0 = 屏蔽TC2溢出中断
- Bit.2 **TC2IE:**
 1 = 使能TC2溢出中断
 0 = 屏蔽TC2溢出中断
- Bit.1 **TC1IE:**
 1 = 使能TC1溢出中断
 0 = 屏蔽TC1溢出中断
- Bit.0 **TC0IE:**
 1 = 使能TC0溢出中断
 0 = 屏蔽TC0溢出中断

5.4 INTF0 中断标志寄存器 0

1C5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF0	I2CIF	ADIF	RXIF	TXIF	TC2GIF	TC2IF	TC1IF	TC0IF
读/写	R/W	R/W	R	R	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

(注：除UART发送/接收之外的所有中断标志位需软件清零)

- Bit 7 **I2CIF:**
 I2C Master写数据
 1 = 缓冲区内接收到数据（读SSPBUF清零）
 0 = 未收到数据或数据已读



- I2C Master读数据
1 = 缓冲区内无待发数据
0 = 缓冲区已写入待发数据
- Bit 6 **ADIF:**
1 = 产生ADC转换中断
0 = 未产生ADC转换中断
- Bit.5 **RXIF:**
1 = 产生串行通讯接收中断
0 = 未产生串行通讯接收中断
- Bit.4 **TXIF:**
1 = 产生串行通讯发送中断
0 = 未产生串行通讯发送中断
- Bit.3 **TC2GIF:**
1 = 产生TC2门控中断
0 = 未产生TC2溢出中断
- Bit.2 **TC2IF:**
1 = 产生TC2溢出中断
0 = 未产生TC2溢出中断
- Bit.1 **TC1IF:**
1 = 产生TC1溢出中断
0 = 未产生TC1溢出中断
- Bit.0 **TC0IF:**
1 = 产生TC0溢出中断
0 = 未产生TC0溢出中断

5.5 INTCR1 中断控制寄存器 1

1C6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCR1	CMPIE	OPA0IE	OPA1IE	IODCHIE	INT1IE	INT0IE	IOCCHIE	IOBCHIE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

- Bit 7 **CMPIE:**
1 = 使能比较器中断
0 = 屏蔽比较器中断
- Bit 6 **OPA0IE:**
1 = 使能运放OPA0中断
0 = 屏蔽运放OPA0中断
- Bit 5 **OPA1IE:**
1 = 使能运放OPA1中断
0 = 屏蔽运放OPA1中断
- Bit.4 **IODCHIE:**
1 = 使能端口D变化中断
0 = 屏蔽端口D变化中断



- Bit.3 **INT1IE:**
 1 = 使能外部端口中断1
 0 = 使能外部端口中断1
- Bit.2 **INT0IE:**
 1 = 使能外部端口中断0
 0 = 使能外部端口中断0
- Bit.1 **IOCCHIE:**
 1 = 使能端口C变化中断
 0 = 屏蔽端口C变化中断
- Bit.0 **IOBCHIE:**
 1 = 使能端口B变化中断
 0 = 屏蔽端口B变化中断

5.6 INTF1 中断标志寄存器 1

1C7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF1	CMPIF	OPA0IF	OPA1IF	IODCHIF	INT1IF	INT0IF	IOCCHIF	IOBHIF
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

(注：所有中断标志位需软件清零)

- Bit 7 **CMIF:**
 1 = 产生比较器中断
 0 = 未产生比较器中断
- Bit 6 **OPA0IF:**
 1 = 产生运放OPA0中断
 0 = 未产生运放OPA0中断
- Bit 5 **OPA1IF:**
 1 = 产生运放OPA1中断
 0 = 未产生运放OPA1中断
- Bit4 **IODCHIF:**
 1 = IOD对应输入端口状态发生变化
 0 = IOD对应输入端口状态未发生变化
- Bit.3 **INTF1IF:**
 1 = 产生外部中断INT0
 0 = 未产生外部中断INT0
- Bit.2 **INTF0IF:**
 1 = 产生外部中断INT0
 0 = 未产生外部中断INT0
- Bit.1 **IOCCHIF:**
 1 = IOC对应输入端口状态发生变化
 0 = IOC对应输入端口状态未发生变化
- Bit.0 **IOBCHIF:**
 1 = IOB对应输入端口状态发生变化
 0 = IOB对应输入端口状态未发生变化



5.7 INTCR2 中断控制寄存器 2

1C2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCR2	TK1IE	TK0IE	-	-	-	-	-	-
读/写	R/W	R/W	-	-	-	-	-	-
复位后	0	0	-	-	-	-	-	-

Bit 7 **TK1IE:**
 1 = 使能触摸按键中断1
 0 = 屏蔽触摸按键中断1

Bit 6 **TK0IE:**
 1 = 使能触摸按键中断0
 0 = 屏蔽触摸按键中断0

5.8 INTF2 中断标志寄存器 2

1C3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF2	TK1IF	TK0IF	-	-	-	-	-	-
读/写	R/W	R/W	-	-	-	-	-	-
复位后	0	0	-	-	-	-	-	-

Bit 7 **TK1IF:**
 1 = 产生触摸按键中断1
 0 = 未产生触摸按键中断1

Bit 6 **TK0IF:**
 1 = 产生触摸按键中断0
 0 = 未产生触摸按键中断0



6 端口

6.1 IOA

IOA 数据寄存器

1C8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOA	IOA7	IOA6	IOA5	IOA4	IOA3	IOA2	IOA1	IOA0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

IOA 方向寄存器

1C9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OEA	OEA7	OEA6	OEA5	OEA4	OEA3	OEA2	OEA1	OEA0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[7:0] **OEA:** A口输出使能

1 = 输出

0 = 输入

IOA 上拉使能寄存器

1CAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PUA	PUA7	PUA6	PUA5	PUA4	PUA3	PUA2	PUA1	PUA0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[7:0] **PUA:** A口上拉使能

1 = 上拉使能

0 = 上拉关闭

IOA 下拉使能寄存器

1CBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PDA	PDA7	PDA6	PDA5	PDA4	PDA3	PDA2	PDA1	PDA0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[7:0] **PDA:** A口上下使能

1 = 下拉使能

0 = 下拉关闭

IOA 端口模式控制寄存器

1D8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
------	------	------	------	------	------	------	------	------



ANSA	ANSA7	ANSA6	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[7:0] **ANSA:** A口模式控制
 1 = 作为模拟端口 (IO输入功能屏蔽)
 0 = 作为数字IO口

6.2 IOB

IOB 数据寄存器

ICCH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOB	IOB7	IOB6	IOB5	IOB4	IOB3	IOB2	IOB1	IOB0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

IOB 方向寄存器

ICDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OEB	OEB7	OEB6	OEB5	OEB4	OEB3	OEB2	OEB1	OEB0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[7:0] **OEB:** B口输出使能
 1 = 输出
 0 = 输入

注: IOB[2]作为输出口的注意事项

- (1) 需将 PUB2 置 1 才能输出高电平.
- (2) IOB[2]输出的高电平是由上拉电阻提供的, 所以驱动能力弱.
- (3) IOB[2]输出的低电平驱动能力比其他端口略弱一些, 输出低电平时内部电路会关闭上拉电阻.

IOB 上拉使能寄存器

ICEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PUB	PUB7	PUB6	PUB5	PUB4	PUB3	PUB2	PUB1	PUB0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[7:0] **PUB:** B口上拉使能
 1 = 上拉使能
 0 = 上拉关闭

IOB 下拉使能寄存器

ICFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PDB	PDB7	PDB6	PDB5	PDB4	PDB3	PDB2	PDB1	PDB0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0



Bit[7:0] **PDB:** B口下拉使能
 1 = 下拉使能
 0 = 下拉关闭

IOB 变化中断使能寄存器

1DAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOBICR	IOBICR7	IOBICR6	IOBICR5	IOBICR4	IOBICR3	IOBICR2	IOBICR1	IOBICR0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[7:0] **IOBICR:** B口变化中断使能
 1 = 使能B口变化中断
 0 = 关闭B口变化中断

6.3 IOC**IOC 数据寄存器**

1D0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOC	-	-	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0
读/写	-	-	RW	RW	RW	RW	RW	RW
复位后	-	-	0	0	0	0	0	0

IOC 方向寄存器

1D1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OEC	-	-	OEC5	OEC4	OEC3	OEC2	OEC1	OEC0
读/写	-	-	RW	RW	RW	RW	RW	RW
复位后	-	-	0	0	0	0	0	0

Bit[7:0] **OEC:** C口输出使能
 1 = 输出
 0 = 输入

IOC 上拉使能寄存器

1D2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PUC	-	-	PUC5	PUC4	PUC3	PUC2	PUC1	PUC0
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	-	0	0	0	0	0	0

Bit[7:0] **PUC:** C口上拉使能
 1 = 上拉使能
 0 = 上拉关闭

IOC 下拉使能寄存器

1D3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PDC	-	-	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	-	0	0	0	0	0	0



Bit[7:0] **PDC:** C口下拉使能
 1 = 下拉使能
 0 = 下拉关闭

IOC 端口模式控制寄存器

1D9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ANSA	-	-	ANSA5	ANSA4	-	-	-	ANSA0
读/写	-	-	R/W	R/W	-	-	-	R/W
复位后	-	-	0	0	-	-	-	0

Bit[5:4,0] **ANSC:** C口模式控制
 1 = 作为模拟端口 (IO输入功能屏蔽)
 0 = 作为数字IO口

IOC 变化中断使能寄存器

1DBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOCICR	-	-	IOCICR5	IOCICR4	IOCICR3	IOCICR2	IOCICR1	IOCICR0
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	-	0	0	0	0	0	0

Bit[7:0] **IOCICR:** C口变化中断使能
 1 = 使能C口变化中断
 0 = 关闭C口变化中断

6.4 IOD**IOD 数据寄存器**

1D4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOD	IOD7	IOD6	IOD5	IOD4	IOD3	IOD2	IOD1	IOD0
读/写	RW	RW	RW	RW	RW	RW	RW	RW
复位后	0	0	0	0	0	0	0	0

IOD 方向寄存器

1D5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OED	OED7	OED6	OED5	OED4	OED3	OED2	OED1	OED0
读/写	RW	RW	RW	RW	RW	RW	RW	RW
复位后	0	0	0	0	0	0	0	0

Bit[7:0] **OED:** D口输出使能
 1 = 输出
 0 = 输入

IOD 上拉使能寄存器

1D6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PUD	PUD7	PUD6	PUD5	PUD4	PUD3	PUD2	PUD1	PUD0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0



Bit[7:0] **PUD:** D口上拉使能
 1 = 上拉使能
 0 = 上拉关闭

IOD 下拉使能寄存器

1D7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PDD	PDD7	PDD6	PDD5	PDD4	PDD3	PDD2	PDD1	PDD0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[7:0] **PDD:** D口下拉使能
 1 = 下拉使能
 0 = 下拉关闭

IOD 变化中断使能寄存器

IDCH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IODICR	IODICR7	IODICR6	IODICR5	IODICR4	IODICR3	IODICR2	IODICR1	IODICR0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[7:0] **IODICR:** D口变化中断使能
 1 = 使能D口变化中断
 0 = 关闭D口变化中断

6.5 IODS 端口驱动设置寄存器

1EBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IODS	IODIS	IOCIS	IOBIS	IOAIS	IODDS	IOCDS	IOBDS	IOADS
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[7:4] **IOxIS:** IO翻转电平选择

IOxIS	IO 输入翻转电平选择
0	SMT 翻转
1	低翻转点

Bit[3:0] **IOxDS:** IO驱动能力选择

IOxDS	IO 驱动能力选择 (端口压差 0.5V 时)
0	标准驱动 (设计值 35mA)
1	小驱动 (设计值 10mA)



7 定时器0/1(TC0/1)

7.1 概述

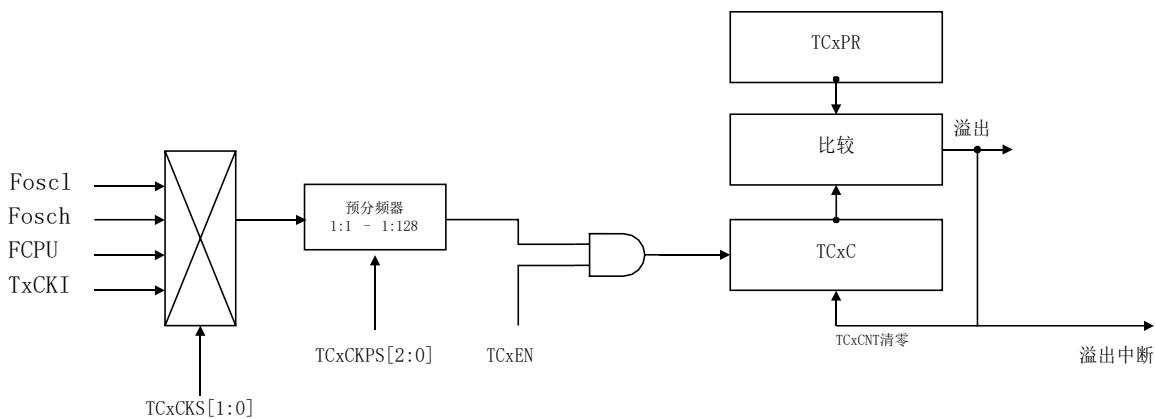
JZ8M4634 TC0/TC1 为带有可设置 1:128 预分频器及周期寄存器的 16 位定时计数器，具有休眠状态下唤醒功能。

向上计数模式，TCx 使能后，TCxC 递增，当 TCxC 计数值与 TCxPR 相等时，TCx 溢出，将 TCxCH 清零重新开始计数，同时将中断标志位 TxIF 置 1。

向下计数模式，TCx 使能后，TCxC 递减，当 TCxC 计数值为 0 时，TCx 溢出，TCxC 将重新载入 TCxPR 的值开始递减计数，同时将中断标志位 TxIF 置 1。

中间对齐方式，TCx 使能后，TCxC 递增计数，当 TCxC 递增到 TCxPR 时，开始递减，递减到 0 时，产生溢出中断，同时开始递增。

- 可选择时钟源，高频系统时钟 Fosch、低频系统时钟 Foscl、指令时钟 Fcpu 和外部时钟 TOCKI
- 16 位周期寄存器
- 预分频比多级可选，最大可选择 1:128
- 溢出中断功能
- 溢出中断唤醒功能（当输入频率选择 Foscl，Fosch 或 TOCKI 时，若所选择的时钟源振荡器一直工作，此时 TC0/TC1 在休眠状态下依然工作，溢出中断可唤醒 CPU）



7.2 TxCR 控制寄存器

1E8H/1ECH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T0CR/T1CR	TCxEN	TCxMOD1	TCxMOD0-	TCxCKS1	TCxCKS0	TCxCKPS2	TCxCKPS1	TCxCKPS0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit 7 **TCxEN:** TCx模块使能位
1 = 使能TCx



0 = 关闭TCx

Bit [6:5] **TCxMOD:** TCx模式选择位

TCxMOD[1:0]	TCx 模式选择
00	递增模式
01	递减模式
1X	中心对齐模式

中心对齐模式下，TCxMOD0为只读模式，表示目前定时器的状态

Bit[4:3] **TCxCKS:** TCx时钟源选择

TCOCKS[1:0]	TC0 时钟源选择
00	Foscl (低频系统时钟)
01	Fosch (高频系统时钟)
10	Fcpu
11	T0cki (TC0) T1cki (TC1)

Bit[2:0] **TCxCKPS[2:0]:** TCx预分频比选择

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

7.3 TCxCL TCx 计数器低位

1E9H/1EDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0CL/TC1CL	TCxCL7	TCxCL6	TCxCL5	TCxCL4	TCxCL3	TCxCL2	TCxCL1	TCxCL0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	X	x	x	x	x	x

7.4 TCxCH TCx 计数器高位

1EAH/1EEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0CH/TC1CH	TCxCH7	TCxCH6	TCxCH5	TCxCH4	TCxCH3	TCxCH2	TCxCH1	TCxCH0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



复位后	x	x	x	x	x	x	x	x
-----	---	---	---	---	---	---	---	---

7.5 TCxPRL TCx 周期寄存器低位

198H/19AH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0/1PRL	TCxPRL7	TCxPRL6	TCxPRL5	TCxPRL4	TCxPRL3	TCxPRL2	TCxPRL1	TCxPRL0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	X	x	x	x	x	x

7.6 TCxPRH TCx 周期寄存器高位

199H/19BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0/1PRH	TCxPRH7	TCxPRH6	TCxPRH5	TCxPRH4	TCxPRH3	TCxPRH2	TCxPRH1	TCxPRH0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x



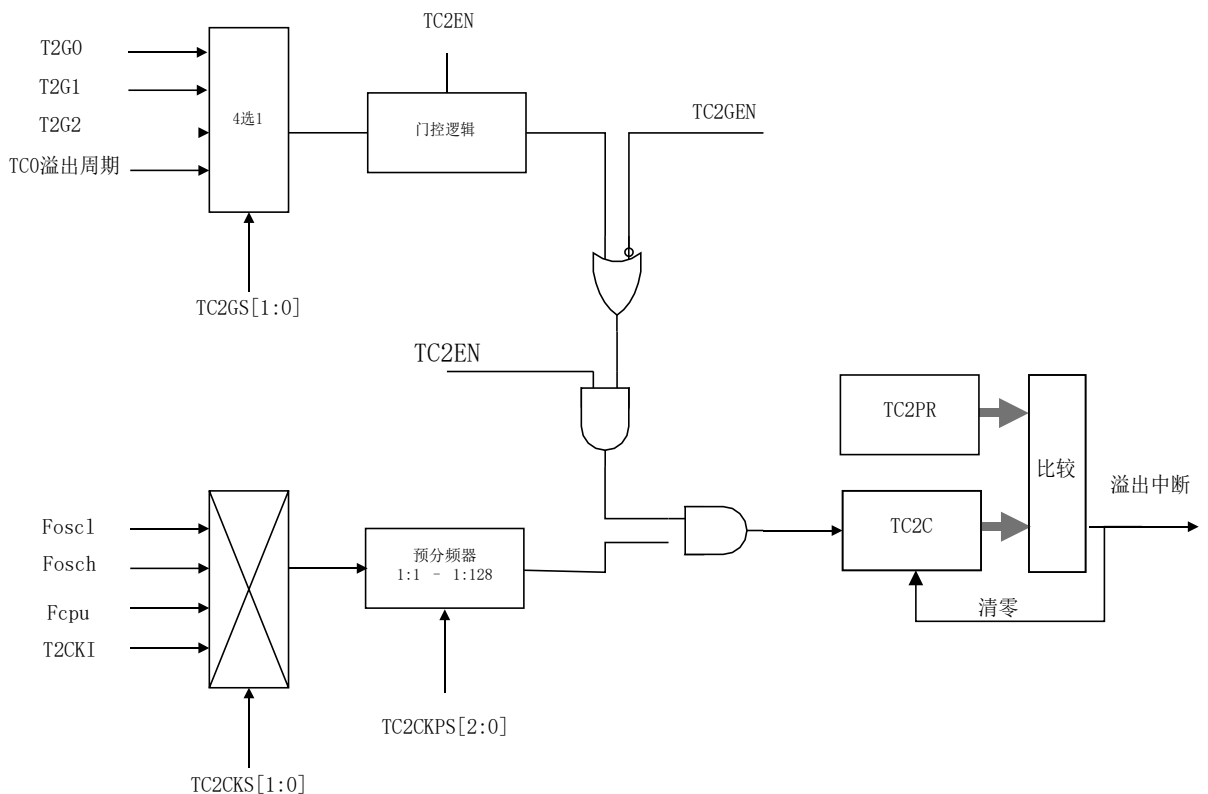
8 定时器2 (TC2)

8.1 概述

JZ8M4634 TC2 为带门控功能的可设置 1:128 预分频器及周期寄存器的 16 位定时计数器，具有休眠状态下唤醒功能。

16 位工作方式与 TC0/TC1 相同。

- 可选择时钟源：高频系统时钟 Fosch，低频系统时钟 Foscl、指令时钟 Fcpu 和 RFCI
- 可选择预分频比，最大 1:128
- 门控模式可选择门控源：TC0 溢出信号、RFC 输出和外部门控信号 T2G0/T2G1
- 溢出中断功能
- 溢出中断唤醒功能（当输入频率选择 Foscl，Fosch 输出时，若所选择的时钟源振荡器一直工作，此时 TC2 在休眠状态下依然工作，溢出中断可唤醒 CPU）





8.2 T2CR 控制寄存器

1F0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2CR	TC2EN	T2MOD1	T2MOD0	TC2CKS1	TC2CKS0	TC2CKPS2	TC2CKPS1	TC2CKPS0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit 7 **TC2EN:** TC2模块使能位

1 = 使能TCx

0 = 关闭TCx

Bit [6:5] **TC2MOD:** TCx模式选择位

TC2MOD[1:0]	TC2 模式选择
00	递增模式
01	递减模式
1X	中心对齐模式

中心对齐模式下，TCxMOD0为只读模式，表示目前定时器的状态

Bit[4:3] **TC2CKS[1:0]:** TC2时钟源选择位

TC2CKS[1:0]	TC2 计数时钟源选择
00	Fosc1(低频系统时钟)
01	Fosch(高频系统时钟)
10	Fcpu
11	T2CKI

Bit[2:0] **TC2CKPS[1:0]:** TC2预分频比选择

TCxCKPS[2:0]	TC2 预分频比
000	1:1
001	1:2
010	1:4
011	1:8
100	1:16
101	1:32
110	1:64
111	1:128

8.3 TC2CL 计数器低位

1F1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2CL	TC2CL7	TC2CL6	TC2CL5	TC2CL4	TC2CL3	TC2CL2	TC2CL1	TC2CL0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	X



8.4 TC2CH 计数器高位

1F2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2CH	TC2CH7	TC2CH6	TC2CH5	TC2CH4	TC2CH3	TC2CH2	TC2CH1	TC2CH0
读/写	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	X

8.5 TC2CL 周期寄存器低位

19CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2PRL	TC2PRL7	TC2PRL6	TC2PRL5	TC2PRL4	TC2PRL3	TC2PRL2	TC2PRL1	TC2PRL0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	X

8.6 TC2CH 周期寄存器高位

19DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2PRH	TC2PRH7	TC2PRH6	TC2PRH5	TC2PRH4	TC2PRH3	TC2PRH2	TC2PRH1	TC2PRH0
读/写	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	X

8.7 TC2GCR 门控控制寄存器

1F3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2GCR	TC2GEN	TC2GO	-	-	TC2GS1	TC2GS0	TC2GM1	TC2GM0
读/写	R/W	R/W	-	-	R/W	R/W	R/W	R/W
复位后	0	0	-	-	0	0	0	0

Bit 7 **TC2GEN:** TC2门控模式使能位

1 = 使能门控功能

0 = 关闭门控功能, TC2是否计数仅受TC2EN控制

Bit 6 **TC2GO:** 启动门控控制位

1 = 启动门控

0 = 完成门控计数, 自动清零

Bit[3:2] **TC2GS[1:0]:** TC2门控源选择位

TC2GS[1:0]	TC2 门控选择
00	TC0 溢出周期 (支持上升沿到上升沿模式)



01	RFC 时钟 RFCI
10	T2G0
11	T2G1

Bit[1:0]

TC2GM[1:0]: TC2门控模式选择位

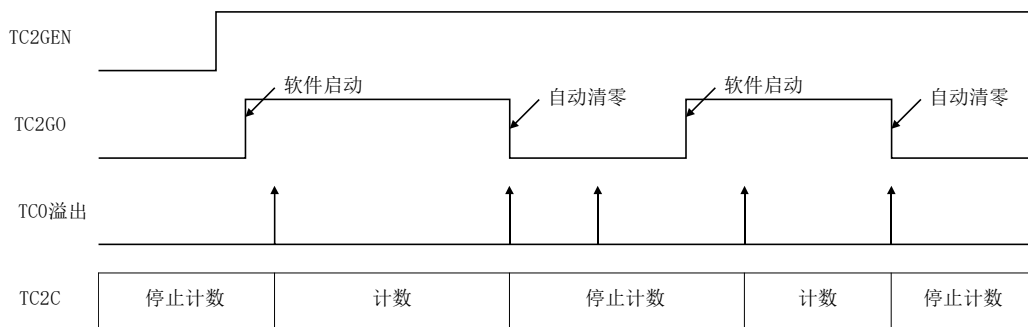
TC2GM[1:0]	TC2 门控模式选择
00	上升沿到下降沿
01	下降沿到上升沿
10	上升沿到上升沿
11	下降沿到下降沿

注:

- (1) 启动门控前需先将门控使能, 不可同时置 1。
- (2) 选择 TC0 溢出信号作为门控源时, 若 TC0 是 8 位模式, 则 TC0CL 不能为 0

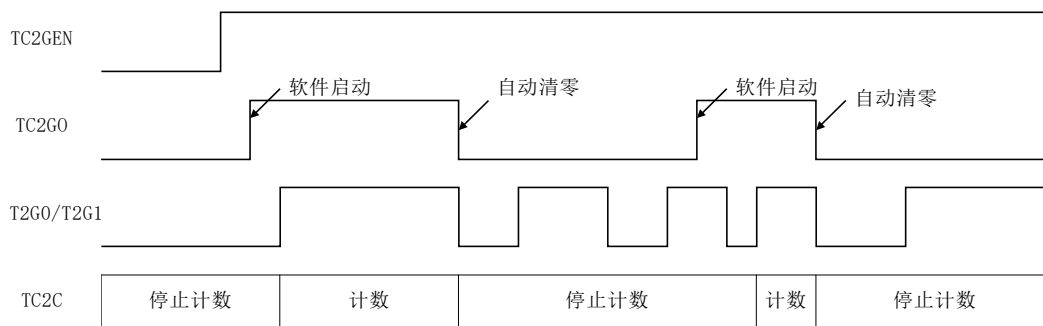
8.7.1 门控-TC0 溢出周期

当门控源选择 TC0 溢出周期时, 需选择上升沿到上升沿模式, 启动门控计数后, 门控逻辑从第一次 TC0 溢出开始计数, 第二次 TC0 溢出停止计数, 如下图。



8.7.2 门控-上升沿到下降沿模式

上升沿到下降沿模式: 启动门控计数后, 门控逻辑从捕获到的第一个上升沿开始计数, 然后捕获到下降沿停止计数, 如下图:

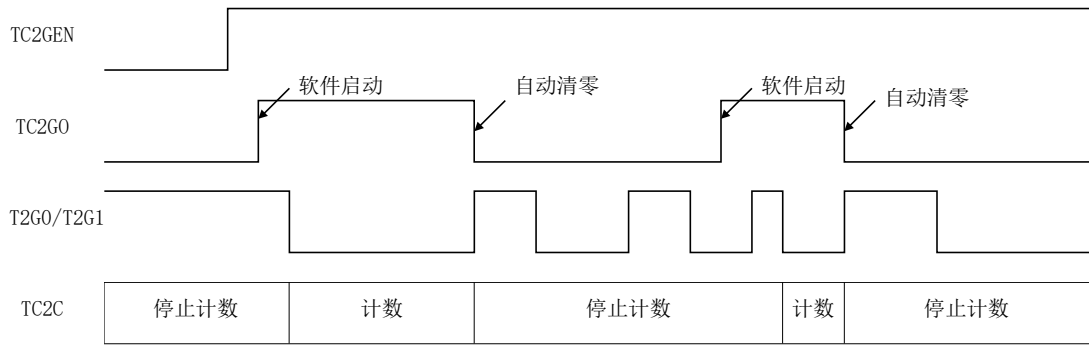


8.7.3 门控-下降沿到上升沿模式

下降沿到上升沿模式: 启动门控计数后, 门控逻辑从捕获到的第一个下降沿开始计数, 然后

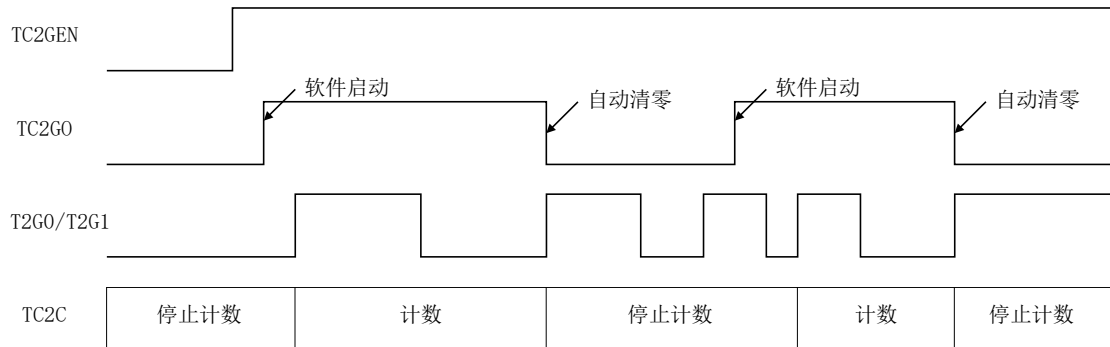


捕获到上升沿停止计数，如下图：



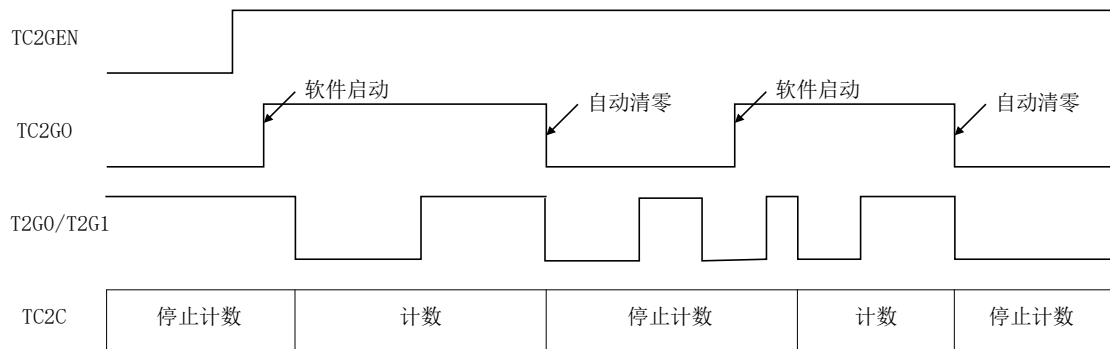
8.7.4 门控-上升沿到上升沿模式

上升沿到上升沿模式：启动门控计数后，门控逻辑从捕获到的第一个上升沿开始计数，然后捕获到第二个上升沿停止计数，如下图：



8.7.5 门控-下降沿到下降沿模式

下降沿到下降沿模式：启动门控计数后，门控逻辑从捕获到的第一个下降沿开始计数，然后捕获到第二个下降沿停止计数，如下图：





9 脉宽调制模块PWM

9.1 概述

JZ8M4634 有 3 路带有死区控制的 PWM，可独立进行设置，16 位分辨率。

- 互补输出
- 死区控制

9.2 PWMxCR 控制寄存器

1E0/1E4/1DDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMxCR	PWMxEN	PWMxPOE	PWMxNOE	PWMxPAS	PWMxNAS	PWMxDEN	PWMxTBS1	PWMxTBS0
读/写	R/W	R/W	R/W	R/W	R/W	DEADEN	RW	RW
复位后	0	0	0	0	0	0	1	1

Bit 7 **PWMxEN:** PWM模块使能位

- 1 = 使能PWM
- 0 = 关闭PWM

Bit 6 **PWMxPOE:** PWM正相波形输出使能位

- 1 = 端口输出PWMxP波形
- 0 = 端口用作IO

Bit 5 **PWMxNOE:** PWM反相波形输出使能位

- 1 = 端口输出PWMxN波形
- 0 = 端口用作IO

Bit 4 **PWMxPAS:** PWMxP波形有效电平选择

- 1 = PWMxP波形有效电平为低电平
- 0 = PWMxP波形有效电平为高电平

Bit 3 **PWMxNAS:** PWMxN波形有效电平选择

- 1 = 端口输出PWM波形高电平有效
- 0 = 端口输出PWM波形低电平有效

Bit 2 **PWMxDEN:** PWM死区控制使能

- 1 = 使能死区控制
- 0 = 关闭死区控制

Bit [1:0] **PWMxTBS:** PWM时基选择

PWMxTBS	PWM 时基选择
00	TC0
01	TC1
10	TC2
11	未定义



9.3 PWMS 输出端口控制寄存器

1F8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMS	-	-	-	PWM0S	PWM2S0	PWM2S0	PWM1S1	PWM1S0
读/写	-	-	-	R/W	R/W	R/W	R/W	R/W
复位后	-	-	-	0	0	0	0	0

Bit 4 **PWM0S**: PWM0 输出管脚选择

1 = PWM0PX/PWM0NX

0 = PWM0P/PWM0N

Bit [3:2] **PWM2S[1:0]**: PWM2 输出管脚选择

PWM2S[1:0]	PWM2 输出管脚
00	PWM2N/PWM2P
01	PWM2NX/PWM2PX
10	PWM2NY/PWM2PY
11	PWM2NX+Y/PWM2PX+Y

Bit [1:0] **PWM1S[1:0]**: PWM1 输出管脚选择

PWM2S[1:0]	PWM1 输出管脚
00	PWM1N/PWM1P
01	PWM1NX/PWM1PX
10	PWM1NY/PWM1PY
11	PWM1NX+Y/PWM1PX+Y

9.4 PWMxDH 数据高位

1E1/1E5/1DEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMxDH	PWMxH7	PWMxH6	PWMxH5	PWMxH4	PWMxH3	PWMxH2	PWMxH1	PWMxH0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

9.5 PWMxDL 数据低位

1E2/1E6/1DFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMxDL	PWMxL7	PWMxL6	PWMxL5	PWMxL4	PWMxL3	PWMxL2	PWMxL1	PWMxL0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

9.6 DEADT PWM 死区控制寄存器

1E3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
------	------	------	------	------	------	------	------	------



DEADT	DEADTF3	DEADTF2	DEADTF1	DEADTF0	DEADTR3	DEADTR2	DEADTR1	DEADTR0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [7:4] **DEADTF[3:0]**: 前死区宽度设置
 $T_{deadf} = DEADTF * \text{时基时钟周期}$

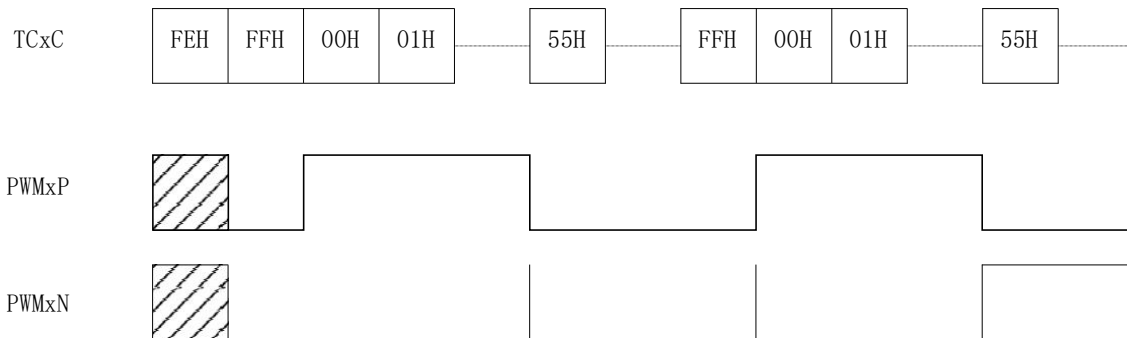
Bit [3:0] **DEADTR[3:0]**: 后死区宽度设置
 $T_{deadr} = DEADTR * \text{时基时钟周期}$

注：
 (1) 时基时钟周期即各 PWM 所选择的时钟源经预分频之后的时钟周期。
 (2) 3 路 PWM 共用同一档前/后死区宽度设置寄存器，但当每路 PWM 选择不同的时基时，死区宽度计算是对应不同的时基时钟周期。

9.7 PWM 输出波形示例

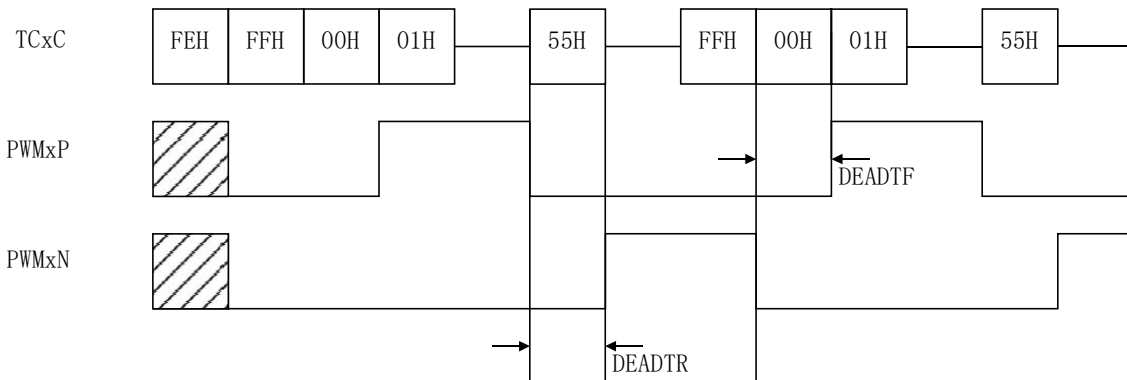
9.7.1 互补 PWM 输出

例: PWMxCR=11100000B PWMxDH=55H, PWMxDL=0H, DEADT=33H, TC0CL=FFH



9.7.2 带死区的互补 PWM 输出

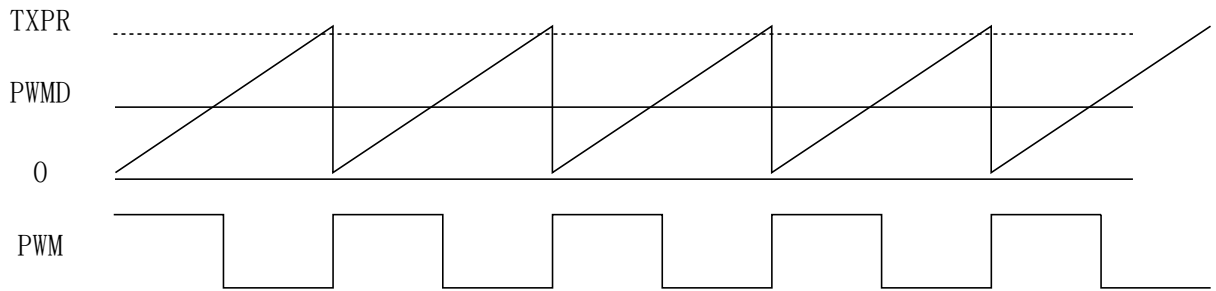
例: PWMxCR=11100100B PWMxDH=55H, PWMxDL=0H, TC0CL=FFH



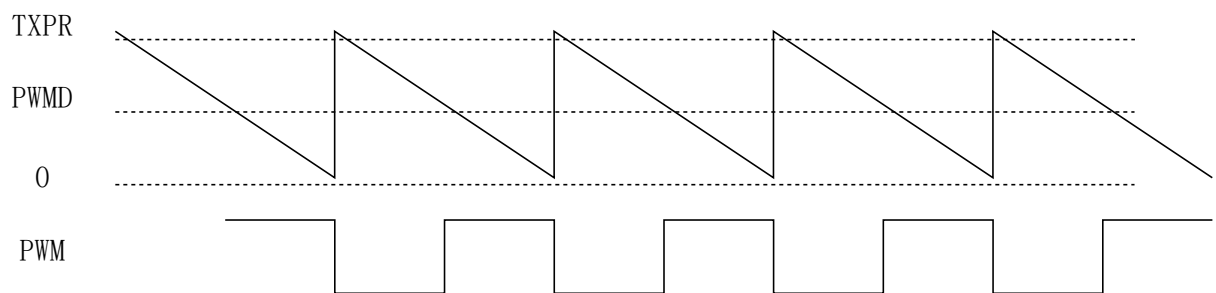


9.7.3 PWM 波形图

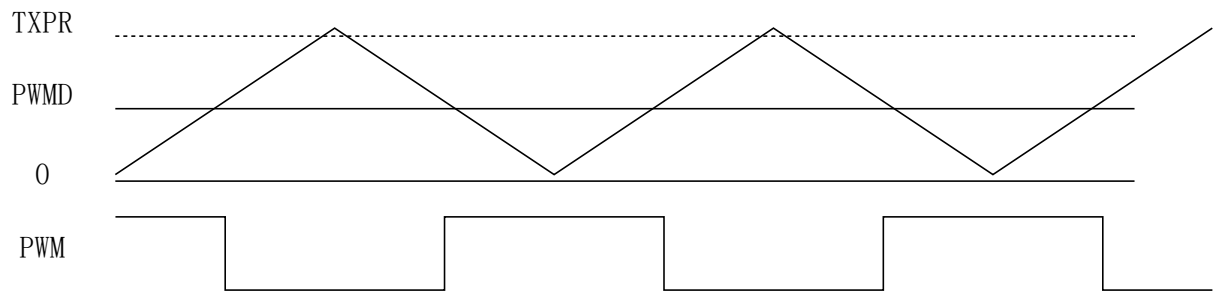
TCx递增模式



TCx递减模式

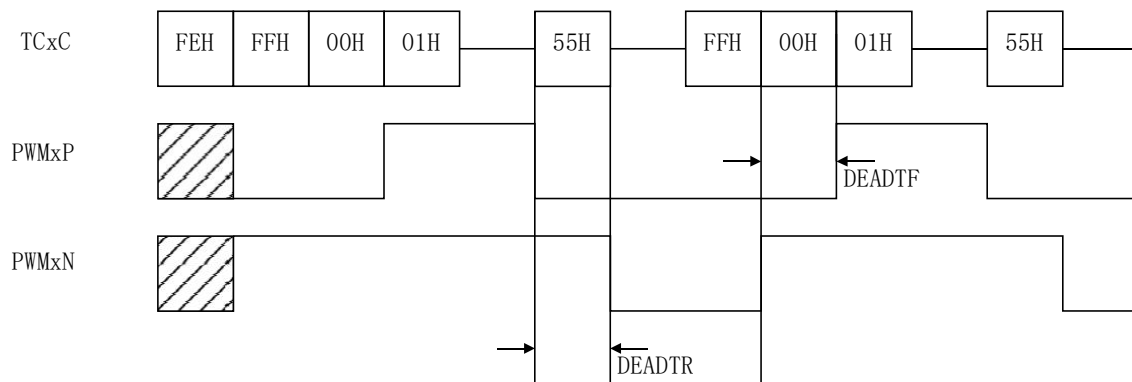


TCx中心对齐模式



9.7.4 有效电平选择

例: PWMxCR=11101100B PWMxDH=55H, PWMxDL=00H, TC0CL=FFH





10 通用串行通讯口 (USART)

10.1 概述

JZ8M4634 支持异步全双工模式和同步半双工模式。

10.2 TX0CR 发送控制寄存器

1B2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TX0CR	TXEN	TMCLR	SYNC	TX9	SLAVE	SPD1	SPD0	TXD9
读/写	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	1	0	0	0	0	0	0

- Bit 7 **TXEN:** 使能发送
 1 = 使能USART发送功能
 0 = 屏蔽USART发送功能
- Bit 6 **TMCLR:** 发送寄存器空标志
 1 = 数据已发送, 移位寄存器空
 0 = 正在发送数据, 移位寄存器不空
- Bit 5 **SYNC:** 同步模式
 1 = 选择同步模式
 0 = 选择异步模式
- Bit 4 **TX9:** 数据长度选择
 1 = 9位数据
 0 = 8位数据
- Bit 3 **SLAVE:** 同步发送/接收模式
 1 = SLAVE
 0 = Master
- Bit[2:1] **SPD[1:0]:** 发送接收速度选择

SPD[1:0]	波特率分频比(n)
11	256
10	64
01	16
00	4

- Bit 0 **TXD9:** 发送数据第9位数据

10.3 TX0REG 发送数据寄存器

1B3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TX0REG	TX0D7	TX0D6	TX0D5	TX0D4	TX0D3	TX0D2	TX0D1	TX0D0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	X	x	x	x	x	x



10.4 RX0CR 接收控制寄存器

1B4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RX0CR	RXEN	CKPS	-	RX9	SREN	RXOVF	FRER	RXD9
读/写	R/W	R/W	-	R/W	R/W	R	R	R
复位后	0	0	-	0	0	0	0	0

- Bit 7 **RXEN:** 使能发送
 1 = 使能USART接收功能
 0 = 屏蔽USART接收功能
- Bit 6 **CKPS:** 同步模式时钟模式选择
 1 = 下降沿发送数据
 0 = 上升沿发送数据
- Bit 4 **RX9:** 数据长度选择
 1 = 9位数据
 0 = 8位数据
- Bit 3 **SREN:** 同步接收开始
 1 = 开始同步接收, 单字节接收模式下接收完一个字节自动清零
 0 = 停止异步接受
- Bit 2 **RXOVF:** 接受缓冲区溢出标志
 1 = 接收缓冲区溢出, 读缓冲区自动清零
 0 = 接收缓冲区未发生溢出
- Bit 1 **FRER:** 接收数据格式错
 1 = 当前接收数据格式错 (未收到停止位)
 0 = 当前接收数据未发生格式错
- Bit 0 **RXD9:** 发送数据第9位数据

10.5 RX0REG 发送数据寄存器

1B5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RX0REG	RX0D7	RX0D6	RX0D5	RX0D4	RX0D3	RX0D2	RX0D1	RX0D0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	X	x	x	x	x	x

10.6 BRGDH 波特率寄存器高位

1B6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRGDH	SBYTE	-	-	-	-	-	BRGD9	BRGD8
读/写	R/W	-	-	-	-	-	R/W	R/W
复位后	0	-	-	-	-	-	0	0

- Bit 1 **SBYTE:** 同步接收模式选择
 1 = 单字节接收, 接收完一个字节后自动清除SREN
 0 = 多字节接收



10.7 BRGDH 波特率寄存器低位

1B7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRGDL	BRGD7	BRGD6	BRGD5	BRGD4	BRGD3	BRGD2	BRGD1	BRGD0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

10.8 USART 使用说明

10.8.1 波特率设置

通过设置 BRGD 和 SPD 来获得所需的波特率。

波特率计算公式: 目标波特率 = $F_{osc} / ((BRGD+1) \times n)$

常用波特率设置 ($F_{osc}=16MHz$)

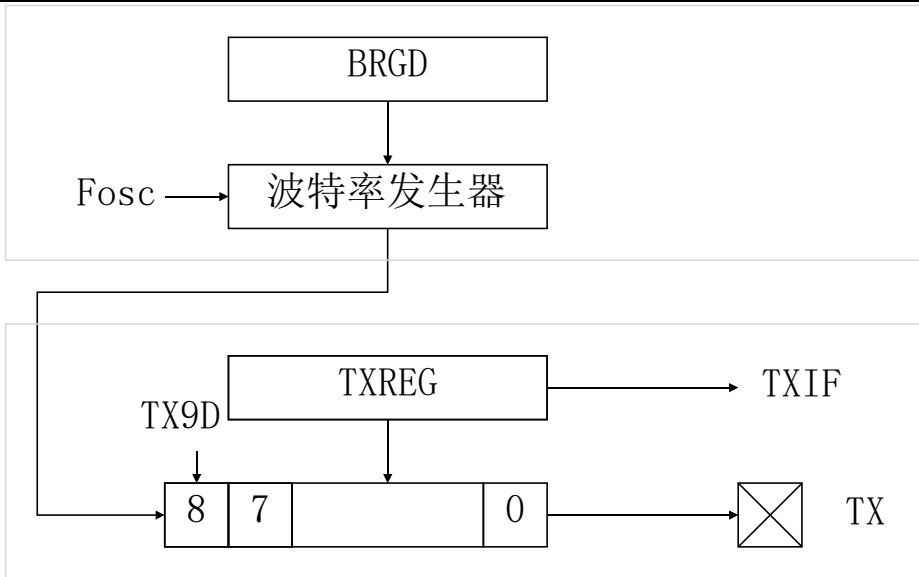
n	目标波特率	150	300	600	1200	2400	4800	9600	19200	38400
256	实际波特率	150.2	300.5	601.0	--	--	--	--	--	--
	BRGD	19FH	CFH	67H	--	--	--	--	--	--
64	实际波特率	--	300.1	601.0	1201.9	2403.8	--	--	--	--
	BRGD	--	340H	19FH	CFH	67H	--	--	--	--
16	实际波特率	--	--	--	1200.5	2403.8	4807.7	9615.4	--	--
	BRGD	--	--	--	340H	19FH	CFH	67H	--	--
4	实际波特率	--	--	--	--	--	4801.9	9615.4	19230.8	38461.5
	BRGD	--	--	--	--	--	340H	19FH	CFH	67H

10.8.2 异步发送

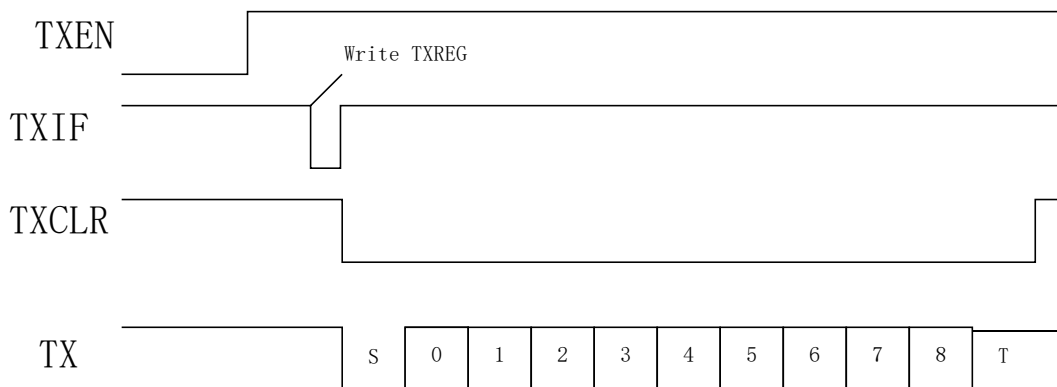
当 TXEN 使能时, TXIF 中断标志为 1 说明 TXREG 发送寄存器为空, TMCLR=1 说明发送移位寄存器为空, 发送器处于空闲状态。

空闲状态时写入 TXREG, 写入数据将立即装载到发送移位寄存器中, 此时, TXIF 为 1, TMCLR=0, 发送器进入发送状态。此时再次写入 TXREG, TXIF 将清零, 说明 TXREG 有未发送数据, 发送移位寄存器发送完毕后, TXREG 数据将自动载入发送移位寄存器继续发送, 且 TXIF 为空。

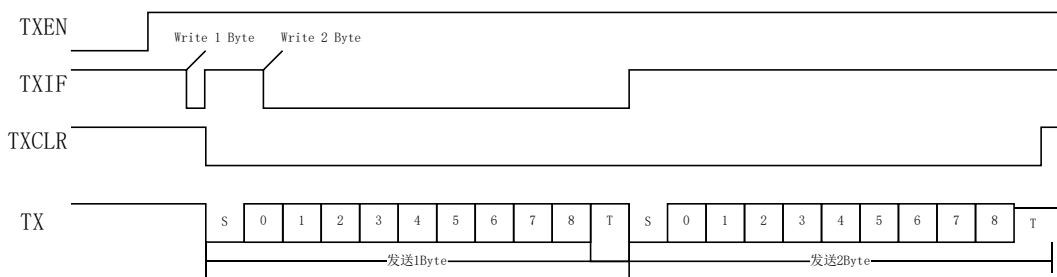
当 TXIF 为 0 时写入 TXREG, 将覆盖上次写入数据。



单字节发送:



多字节发送:

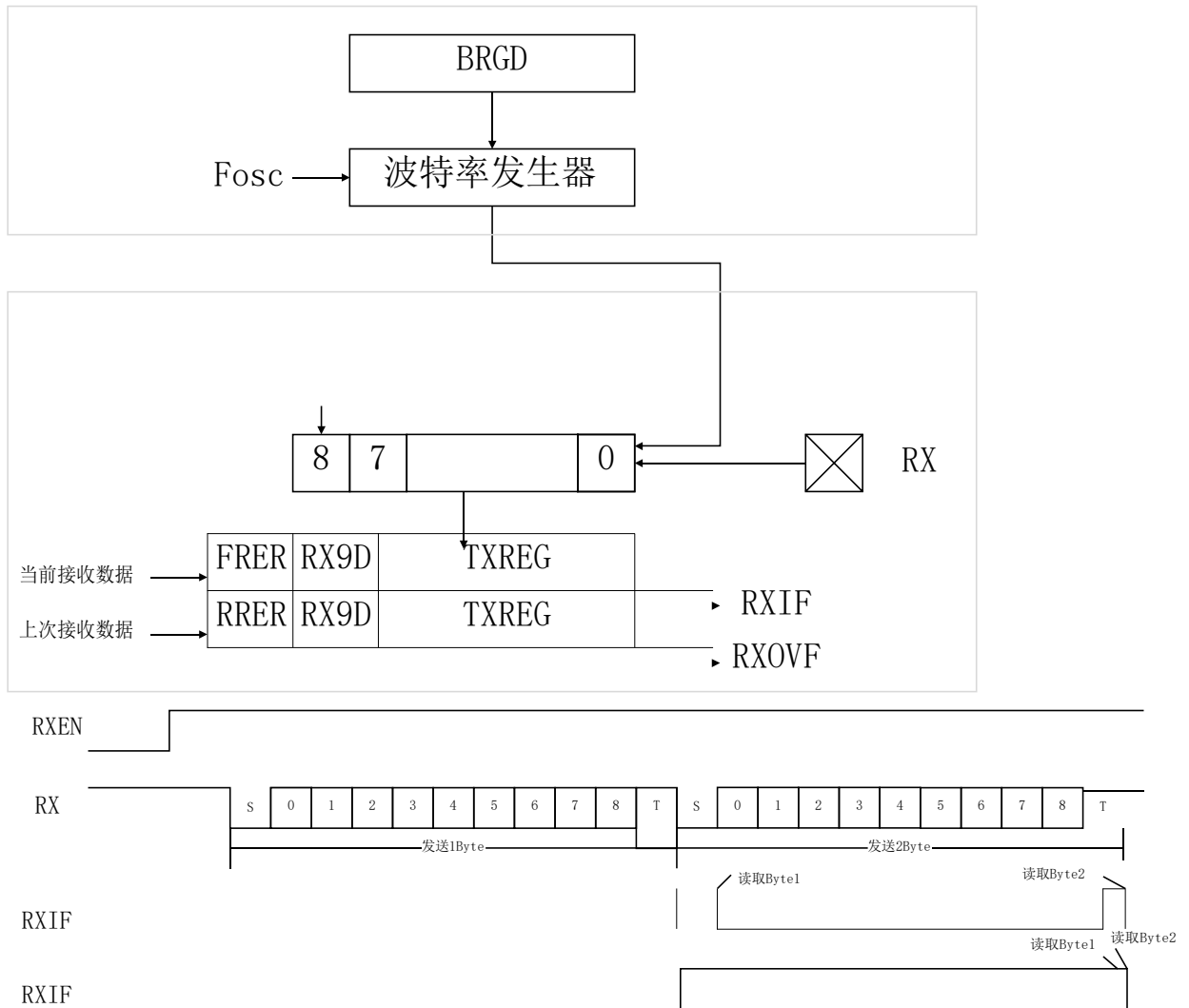


- STEP1: 设置波特率 SSPD=X, BRGD=X, SYNC=0
- STEP2: 设置 TXEN=1, 设置数据模式 TX9=X
- STEP3: 写入数据高位 TXD9
- STEP4: 写入 TXREG, 启动发送
- STEP5: 当 TXIF=1 时, 写入 TXREG 发送下一个字节
- STEP6: 重复 STEP5, 直到该帧数据发送完成



10.8.3 异步接收

设置异步模式，使能 RXEN，开始启动异步接收。RX 管脚处于高电平时，接收器处于空闲状态，当检测到 RX 变为低电平，接收器检测该低电平是否有效起始位，若为有效起始位，则启动数据时钟恢复电路和数据恢复电路进行接收。1 个数据接收完成后，RXIF 置 1，当接收 3 个数据未读取，RXOVF 置 1，同时舍弃第三个接收数据。完全读取 RXREG 后 RXIF 自动清零。



- STEP1: 设置波特率 SSPD=X, BRGD=X, SYNC=0
- STEP2: 设置 RXEN=1
- STEP3: 等待接收完成 RXIF=1
- STEP4: 判断 FRER=0, 若为 1, 帧格式错误, 舍弃数据
- STEP5: 读取 RX9D
- STEP6: 读取 RXREG, 重复 3-6

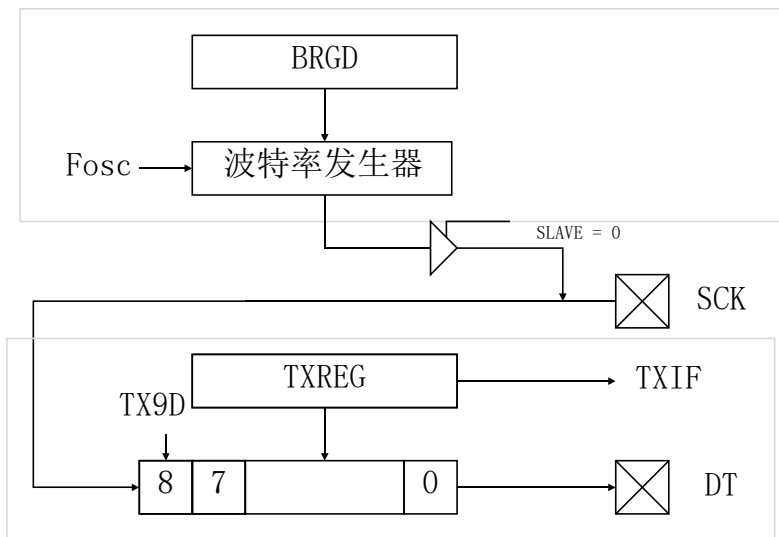


10.8.4 同步发送

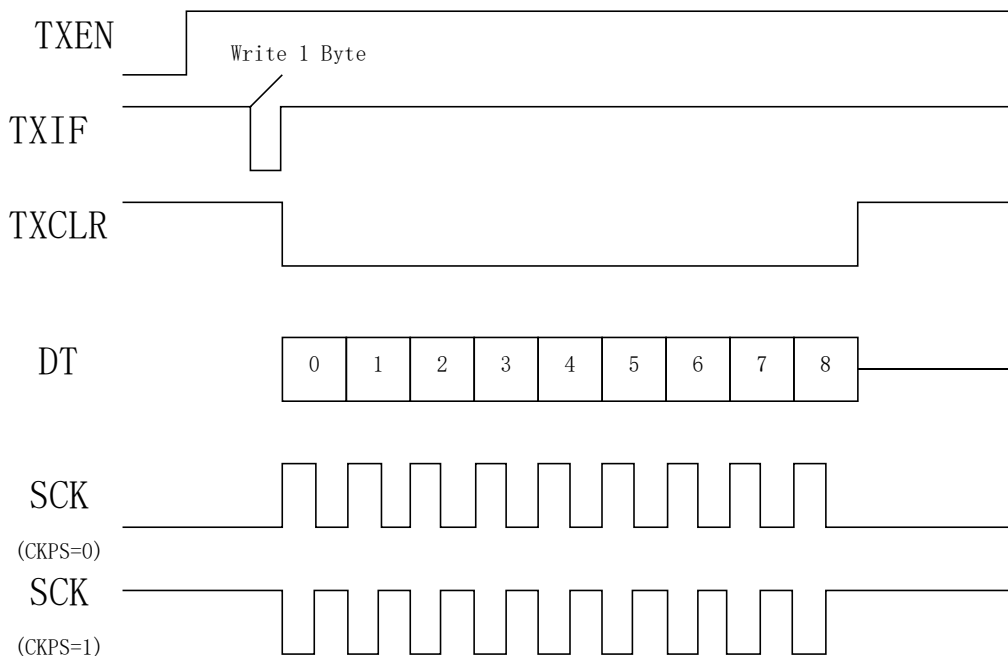
当 TXEN=1, SYNC=1 时, 使能同步发送功能。CKPS 选择发送时钟极性, TXIF 中断标志为 1 说明 TXREG 发送寄存器为空, TMCLR=1 说明发送移位寄存器为空, 发送器处于空闲状态。

空闲状态写入 TXREG, 写入数据将立即装载到发送移位寄存器中, 此时, TXIF 为 1, TMCLR=0, 发送器进入发送状态。此时再次写入 TXREG, TXIF 将清零, 说明 TXREG 有未发送数据, 发送移位寄存器发送完毕后, TXREG 数据将自动载入发送移位寄存器继续发送, 且 TXIF 为空。

当 TXIF 为 0 时写入 TXREG, 将覆盖上次写入数据。

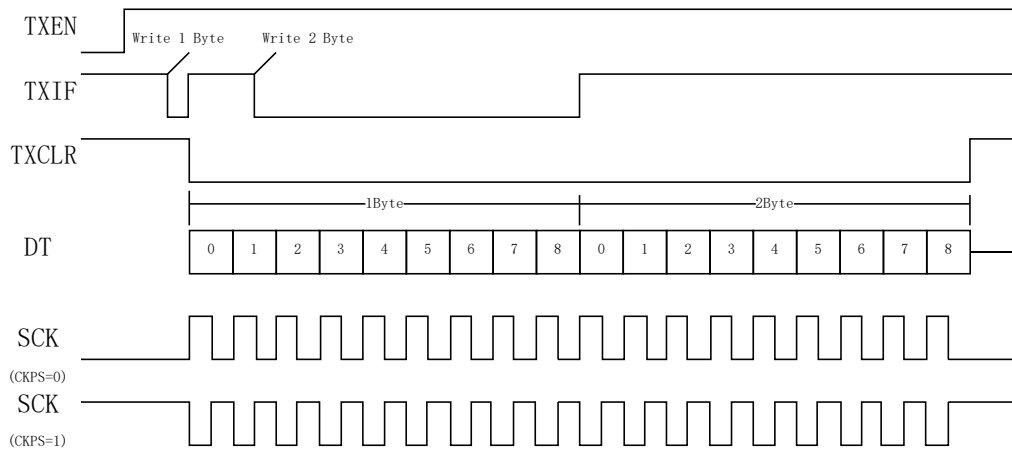


单字节发送:





多字节发送:



参考操作步骤 SLAVE=0:

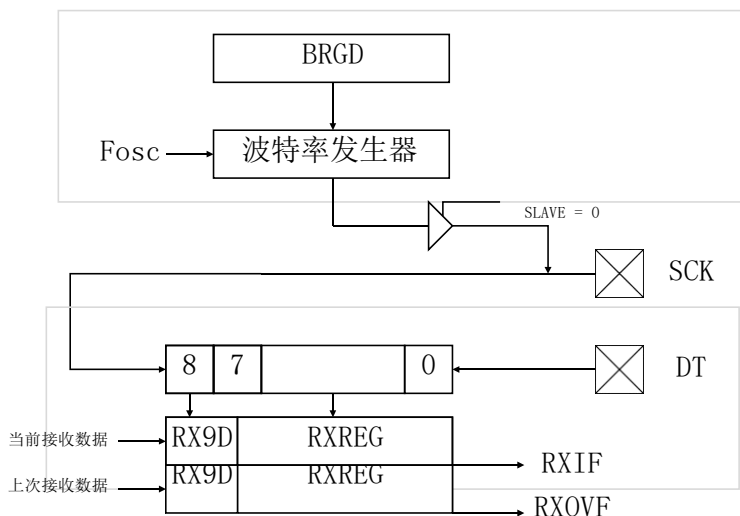
- STEP1: 设置波特率 SSPD=X, BRGD=X, SYNC=1
- STEP2: 设置 TXEN=1, 设置数据模式 TX9=X
- STEP3: 写入数据高位 TXD9
- STEP4: 写入 TXREG, 启动发送
- STEP5: 当 TXIF=1 时, 写入 TXREG 发送下一个字节
- STEP6: 重复 STEP5, 直到该帧数据发送完成

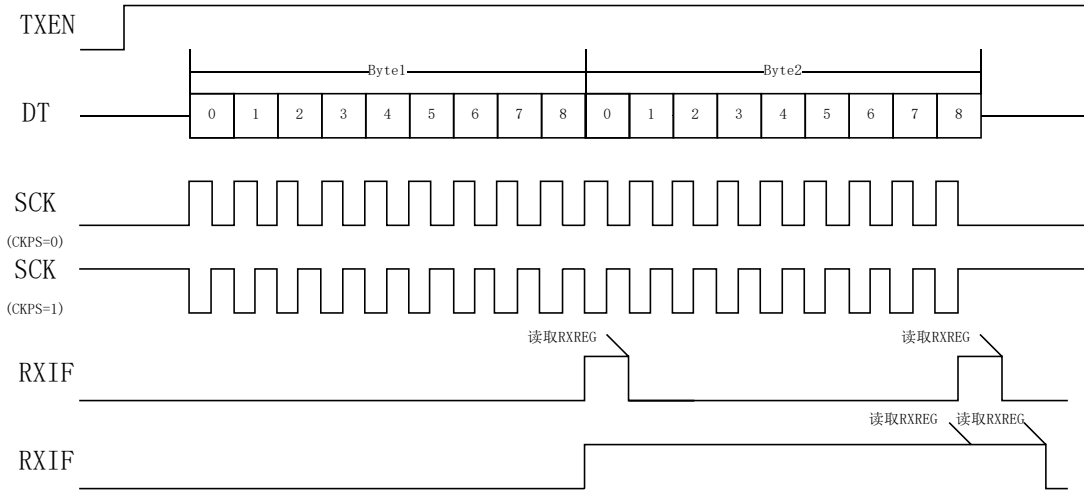
参考操作步骤 SLAVE=1:

- STEP1: 设置波特率 SSPD=X, BRGD=X, SYNC=1
- STEP2: 设置 TXEN=1, 设置数据模式 TX9=X
- STEP3: 当 TXIF=1 时, 写入数据高位 TXD9
- STEP4: 写入 TXREG 等待发送下一个字节
- STEP5: 重复 STEP3-4, 直到该帧数据发送完成

10.8.5 同步接收

设置同步 SYNC=1 模式, 使能 RXEN, 开始启动异步接收。RX 管脚处于高电平时, 接收器处于空闲状态, 当检测到 RX 变为低电平, 接收器检测该低电平是否有效起始位, 若为有效起始位, 则启动数据时钟恢复电路和数据恢复电路进行接收。1 个数据接收完成后, RXIF 置 1, 当接收 3 个数据未读取, RXOVF 置 1, 同时舍弃第三个接收数据。完全读取 RXREG 后 RXIF 自动清零。





参考操作步骤 SLAVE=0:

STEP1: 设置波特率 SSPD=X, BRGD=X, SYNC=0

STEP2: 设置 RXEN=1

STEP3: 写 SREN 启动接收

STEP4: 等待接收完成 RXIF=1

STEP5: 读取 RX9D

STEP6: 读取 RXREG, 单字节接收 (SBYTE=1) 重复 3-6; 多字节接收 (SBYTE=0) 重复 4-6

参考操作步骤 SLAVE=1:

STEP1: 设置波特率 SSPD=X, BRGD=X, SYNC=0

STEP2: 设置 RXEN=1

STEP3: 写 SREN 启动接收

STEP4: 等待接收完成 RXIF=1

STEP5: 读取 RX9D

STEP6: 读取 RXREG, 单字节接收 (SBYTE=1) 重复 3-6; 多字节接收 (SBYTE=0) 重复 4-6

10.8.6 唤醒及休眠模式下通讯

TXIE 置 1 时, TXIF 中断标志唤醒 CPU

RXIE 置 1 时, RXIF 中断标志唤醒 CPU

异步接收时, 检测到 START 位将自动使能高频振荡器, 接收完成后唤醒 CPU

同步接收时, 若作为主机, 则休眠状态下部工作; 作为从机, 则接收 1 个字节完成后唤醒 CPU



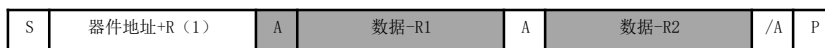
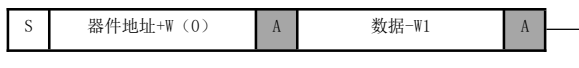
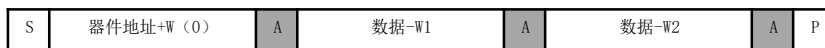
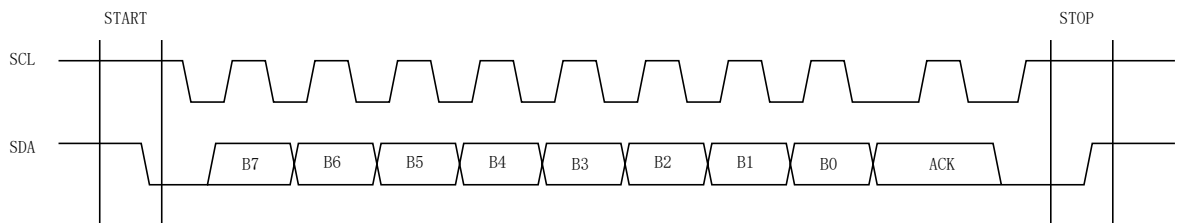
11 串行通讯口 (I2C)

11.1 概述

JZ8M4634 支持高速 I2C (400K) Slave。

注 (1) CPU 时钟选择 2T 时不支持。

11.2 通讯波形示意



11.3 I2CCON I2C 控制寄存器

1E7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CCON	I2CEN	R_W	D_A	BF	NACK	A2	A1	A0
读/写	R/W	R	R	R	R	R/W	R/W	R/W
复位后	0	1	0	0	0	0	0	0

Bit 7

I2CEN: 使能发送

1 = 使能串行通讯功能

0 = 屏蔽串行通讯功能

Bit 6

R_W: 发送寄存器空标志

1 = Master读数据 (主机发接收数据命令)



0 = Master写数据（复位后、START、STOP后）

- Bit 5 **D_A**: 数据地址标志
 1 = 主机发送的是数据(Start后发过的第一个数据)
 0 = 主机发送的是地址
- Bit 4 **BF**: 数据缓冲区满
 R_W = 0
 1 = 从机接收到数据，未读
 0 = 已读或未接受到数据
 R_W = 1
 1 = 有数据待发送
 0 = 数据已发送或正在发送
- Bit 3 **NACK**: Master读数据ACK
 1 = Master停止接收数据
 0 = Master继续接收下一个数据
- Bit[2:0] **A[2:0]**: 从机地址
 地址(写入): $0xA0 + A[2:0]*2$
 地址(读取): $0xA0 + A[2:0]*2 + 1$

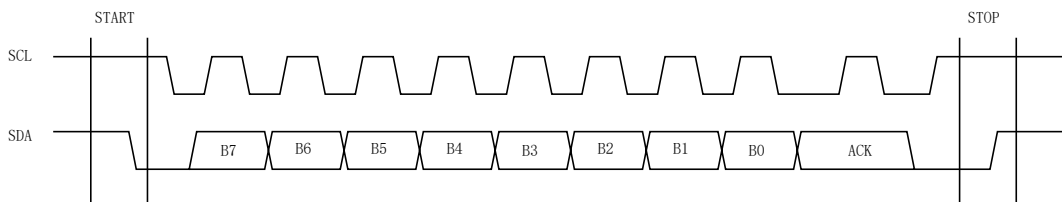
11.4 I2CBUF 数据寄存器

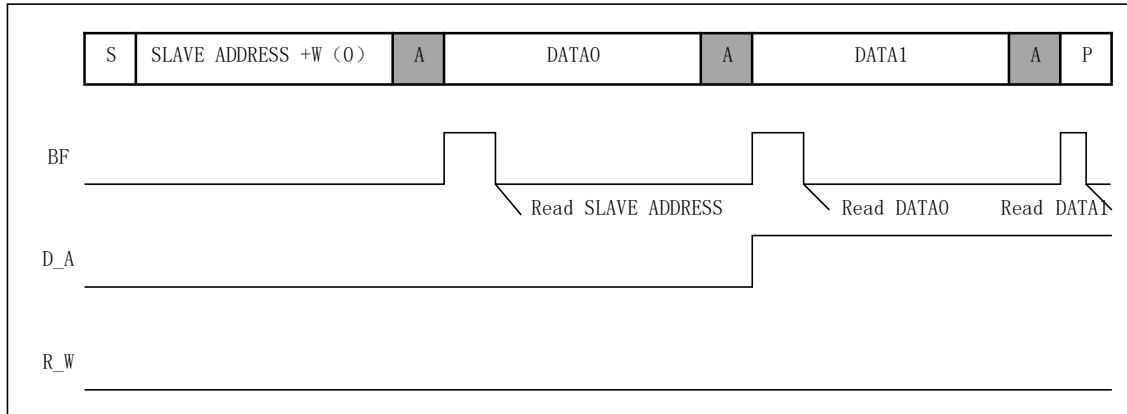
1EFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CBUF	I2CBUF7	I2CBUF6	I2CBUF5	I2CBUF4	I2CBUF3	I2CBUF2	I2CBUF1	I2CBUF0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	X	x	x	x	x	x

11.5 唤醒及休眠模式下通讯

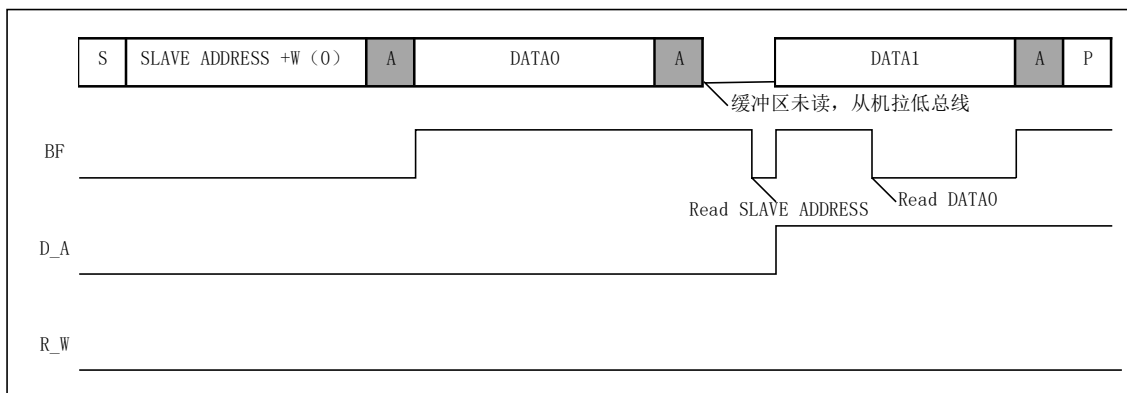
当 I2CCON.7 为 1 时，SCL、SDA 线的低电平会唤醒 CPU，并开始通讯，通讯期间 CPU 无法进入休眠模式

11.6 通讯波形图

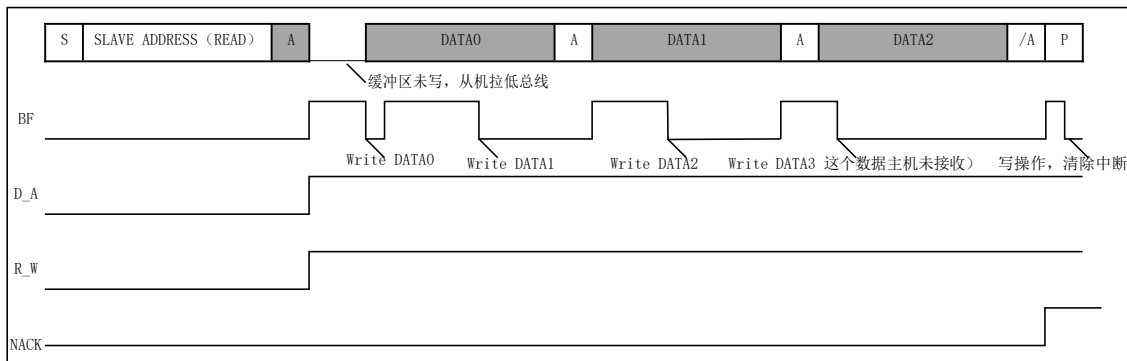




地址匹配后，主机向从机写入两个数据



两个数据没有读取，从机拉低 SCL，等待数据读取



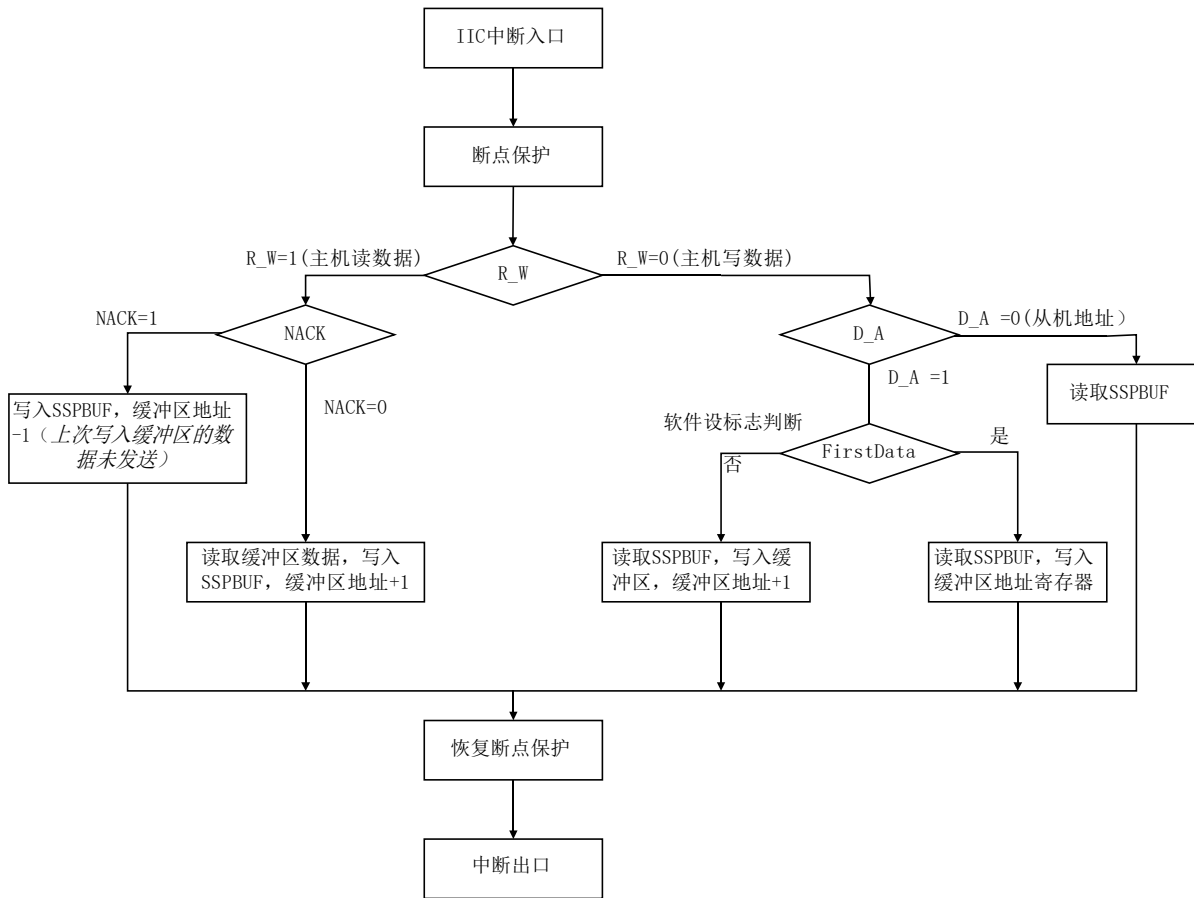
I2C Master 读数据此操作

11.7 应用示例

JZ8M4634 作为从机，允许主机通过 I2C 接口对其 RAM 进行读写操作



11.7.1 从机软件流程图



11.7.1 例程

```

;+++++
LIST P= JZ8M4634
#include JZ8M4634.inc
;+++++
#define      bFirstByte  0x0f,0
#define      SlaveBuffAddr  0x0e
#define      StartAdr 0x10
;+++++
ORG      .0
GOTO      MainProgram
;+++++
//中断入口
      ORG  .8
      GOTO  isr_pro
;+++++

```



```
MainProgram:
    MOVIA    b'10000010' ;//IICSLAVE 为 0xA2(写)/0xA3 (读)
    MOVAR    SSPCON
    BSET     SSPCON,SSPIE ;//使能 IIC 中断
    BSET     OPTION,GIE

MainLoop:
    CLRWDT
    GOTO     MainLoop

;/+-----+
; //中断处理子程序
isr_pro:
    PUSH
    MOVR     FSR0,A
    MOVAR    sFSR0      ;//中断中用到 FSR0, 保护
    JBTS0    intf2,0
    GOTO     sspint

isr_End:
    BCLR     FLAG
    MOVR     sFSR0,A
    MOVAR    FSR0      ;//中断中用到 FSR0, 保护
    POP
    RETIE

sspint:
    BSET     FLAG
    JBTS1    sspcon,R_W ;//R_W
    GOTO     iicMasterWrite

iicMasterRead:
    JBTS0    sspcon,NACK
    GOTO     iicMasterReadNoAck
    MOVR     SlaveBuffAddr,A
    MOVAR    FSR0
    MOVR     INDF0,A
    MOVAR    SSPBUF
    INCR     SlaveBuffAddr,R
    GOTO     isr_End

iicMasterReadNoAck:      ;//Master 读数据完毕
    DECR     SlaveBuffAddr,R ;//上次写入数据未接收, 地址减 1
    MOVAR    SSPBUF      ;//清除中断
    GOTO     isr_End

iicMasterWrite:
    JBTS0    SSPCON,D_A
    GOTO     iicGetBufAddr
```



```
iicAddr:
    MOVR    SSPBUF,A      ;//读数据,清空缓冲区
    BSET    bFirstByte
    GOTO    isr_End

iicGetBufAddr:
    JBTS1   bFirstByte
    GOTO    iicGetData
    MOVR    SSPBUF,A
    MOVAR   SlaveBuffAddr
    BCLR    bFirstByte
    GOTO    isr_End

iicGetData:
    MOVR    SlaveBuffAddr,A
    MOVAR   FSR0
    MOVR    SSPBUF,A
    MOVAR   INDF0
    INCR    SlaveBuffAddr,R
    GOTO    isr_End

;//+++++
```

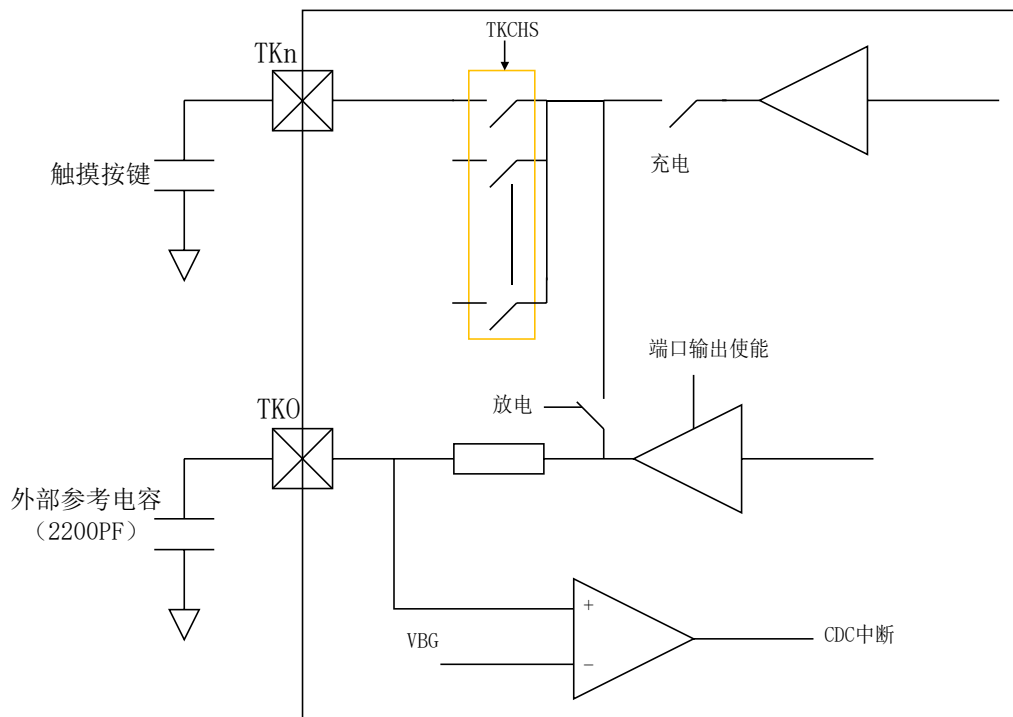



12 触摸按键 (CDC)

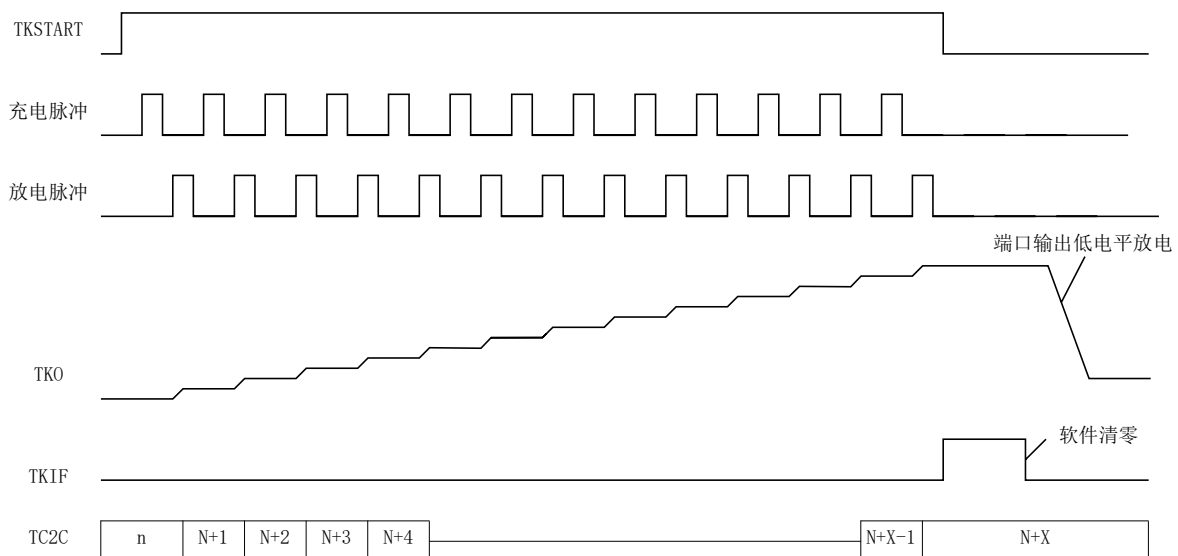
12.1 概述

JZ8M4634 有 2 组触摸按键模块，灵敏度可通过外接电容调节，可替代机械式触摸按键，实现防水防尘，简单易用的操作接口。

12.2 原理框图



信号波形示意图:





12.3 TKxCTR0 控制寄存器 0

1A5/1A0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKxCTR0	TKEN	TKSTART	TKCKS1	TKCKS0	-	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-	-
POR	0	0	0	0	-	-	-	-

Bit 7 **TKEN**: CDC模块使能控制位

1 = 使能CDC模块

0 = 关闭CDC模块

Bit 6 **TKSTART**: 启动通道转换

1 = 启动通道转换

0 = 通道转换完成, 自动清零

Bit [5:4] **TKCKS[1:0]**: CDC时钟分频选择位

TKCKS [1:0]	输入信号选择
00	Fosch/4
01	Fosch/8
10	Fosch/16
11	Fosch/32

12.3 TKxCHSH/L 触摸按键通道选择寄存器

1A6/1A1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKxCHSL	TKxCHS7	TKxCHS6	TKxCHS5	TKxCHS4	TKxCHS3	TKxCHS2	TKxCHS1	TKxCHS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

1A7/1A2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKxCHSH	TKxCHS15	TKxCHS14	TKxCHS13	TKxCHS12	TKxCHS11	TKxCHS10	TKxCHS9	TKxCHS8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit [7:0] **TKxCHSn** 触摸通道连通

1 = 连通该通道

0 = 不连通该通道

注 (1) CDC0 模块有效通道为 CHS10~CHS0, 对应 IO 口 TK10~TK0.

(2) CDC1 模块通道 CHS15~CHS0 对应 IO 口 TK31~TK16.



12.4 TKxCNTH/L 触摸按键计数寄存器

1A3/19EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKxCNTH	TKxCNTH7	TKxCNTH6	TKxCNTH5	TKxCNTH4	TKxCNTH3	TKxCNTH2	TKxCNTH1	TKxCNTH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

1A4/19FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKxCNTL	TKxCNTL7	TKxCNTL6	TKxCNTL5	TKxCNTL4	TKxCNTL3	TKxCNTL2	TKxCNTL1	TKxCNTL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

注：（1）按键扫描过程中，不允许写操作。

12.5 操作说明

- 1- 设置相关通道 IO 方向控制及设置为模拟 PIN
- 2- 定时器 2 时钟源设置为 CDC 输出 TKCLK
- 3- CDC 相关通道/转换时钟设置
- 4- 使能 CDC 模块 TKEN=1
- 5- 设置 TKO 管脚输出 0，对外接电容放电（保证足够时间放电完全）
- 6- 设置 TKO 管脚为输入模式
- 7- 清除定时器 2 TC2CH/TC2CL
- 8- 启动 CDC 转换（TKSTART 置 1）
- 9- 等待转换完成（TKSTART=0）/或使用中断模式（TKIF）
- 10- 读取定时器 2 的计数值，判断是否有按键发生
- 11- 重复 3-10 对不同通道进行扫描



13 模数转换器(ADC)

13.1 概述

JZ8M4634有个11路外部通道（AIN0~AIN10）和5路内部通道（VDD_DIV，VREF，OPAO和GND）12位分辨率的A/D转换器，可以将模拟信号转换成12位数字信号。进行AD转换时，首先要选择输入通道，然后启动AD转换。转换结束后，系统自动将EOC设置为“1”，并将转换结果存入寄存器ADH和寄存器ADL中。

13.2 ADCON0 寄存器

1F4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	ADENB	ADS	EOC	ADFM	CHS3	CHS2	CHS1	CHS0
读/写	R/W	R/W	R	R/W	RW	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

- Bit 7 **ADENB:** ADC使能控制位。
 1 = 使能ADC
 0 = 关闭ADC
- Bit 6 **ADS:** ADC 启动位。
 1 = 开始（每次写入1将重新启动ADC）。
 0 = 停止，转换完成自动清零
- Bit 5 **EOC:** ADC 状态控制位。
 1 = 转换结束，ADS 复位。
 0 = 转换过程中；
- Bit 4 **ADFM:** 数据格式选择位。
 1 = ADRES = {ADH[3:0], ADL[7:0]}; ADH[7:4] = 0。
 0 = ADRES = {ADH[7:0], ADL[7:4]}; ADL[3:0] = 0。
- Bit[3:0] **CHS[3:0]:** ADC 输入通道选择位。
 [0000] ~ [1010] = AIN0 ~ AIN10
 [1011] = 运放 OPA0 输出[1100]
 = VDD/4
 [1101] = 内建 VREF 基准电平
 [1110] = GND
 [1111] = 运放 OPA1 输出

注 1) 若 ADENB = 1, 用户应设置 IOA.n/AINn 为无上拉电阻的输入模式。系统不会自动设置。若已经设置了 ANSEL.n, IOA.n/AINn 的数字 I/O 功能（都是隔离开来的）。



13.3 ADCON1 寄存器

1F5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	VHS2	ADCKS2	ADCKS1	ADCKS0	VREMS1	VREMS0	VHS1	VHS0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[6:4] **ADCKS[2:0]**: ADC 时钟源选择位。

ADCKS[2:0]	ADC 时钟源选择
000	Fcpu
001	Fcpu/2
010	Fcpu/4
011	Fcpu/8
100	Fcpu/16
101	Fcpu/32
110	Fcpu/64
111	

Bit[3:2] **VREMS[1:0]**: ADC 参考电压模式选择位。

VREMS[1:0]	ADC 参考电压模式
00	VDD
01	内部参考电压
10	外部参考电压
11	内部参考与外部参考连接

Bit[7, 1:0] **VHS[2:0]**: ADC 内建基准电平选择位。

内建 VREF 基准电平		
VHS[1:0]	VHS[2]=0	VHS[2]=1
00	关闭内部参考	关闭内部参考
01	2.0V	180mV
10	3.0V	270mV
11	4.0V	360mV

注：1) 若由 VHS[1:0]控制选择的内部 VREF 电平高于 VDD，内部 VREF 为 VDD。

例：VHS[1:0] = 11 (内部 VREF = 4.0V)，VDD = 3.0V，则实际内部 VREF = 3.0V。



13.4 ADCON2 寄存器

1BFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON2	-	-	-	-	ADVOS3	ADVOS2	ADVOS1	ADVOS0
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位后	-	-	-	-	0	0	0	0

Bit[3:0] ADVOS[3:0]: ADC失调补偿寄存器

13.5 ADH ADC 数据高位

1F7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADH	-	-	-	-	-	-	-	-
读/写	R	R	R	R	R	R	R	R
复位后	X	X	X	X	X	X	X	X

13.6 ADL ADC 数据低位

1F6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADL	-	-	-	-	-	-	-	-
读/写	R	R	R	R	R	R	R	R
复位后	X	X	X	X	X	X	X	X

注: ADH/ADL 的数据格式与 ADM 相关, 当 ADFM=1 时, ADH[7:4] =0, ADH[3:0]存放高四位数据, ADL[7:0]存放低 8 位数据; 当 ADFM=0 时, ADH[7:0]存放高 8 位数据, ADL[7:4]存放低 4 位数据, AD:[3:0] = 0

13.7 AD 转换时间

12 位 AD 转换时间 = $1/(\text{ADC clock}) * 16 \text{ sec}$

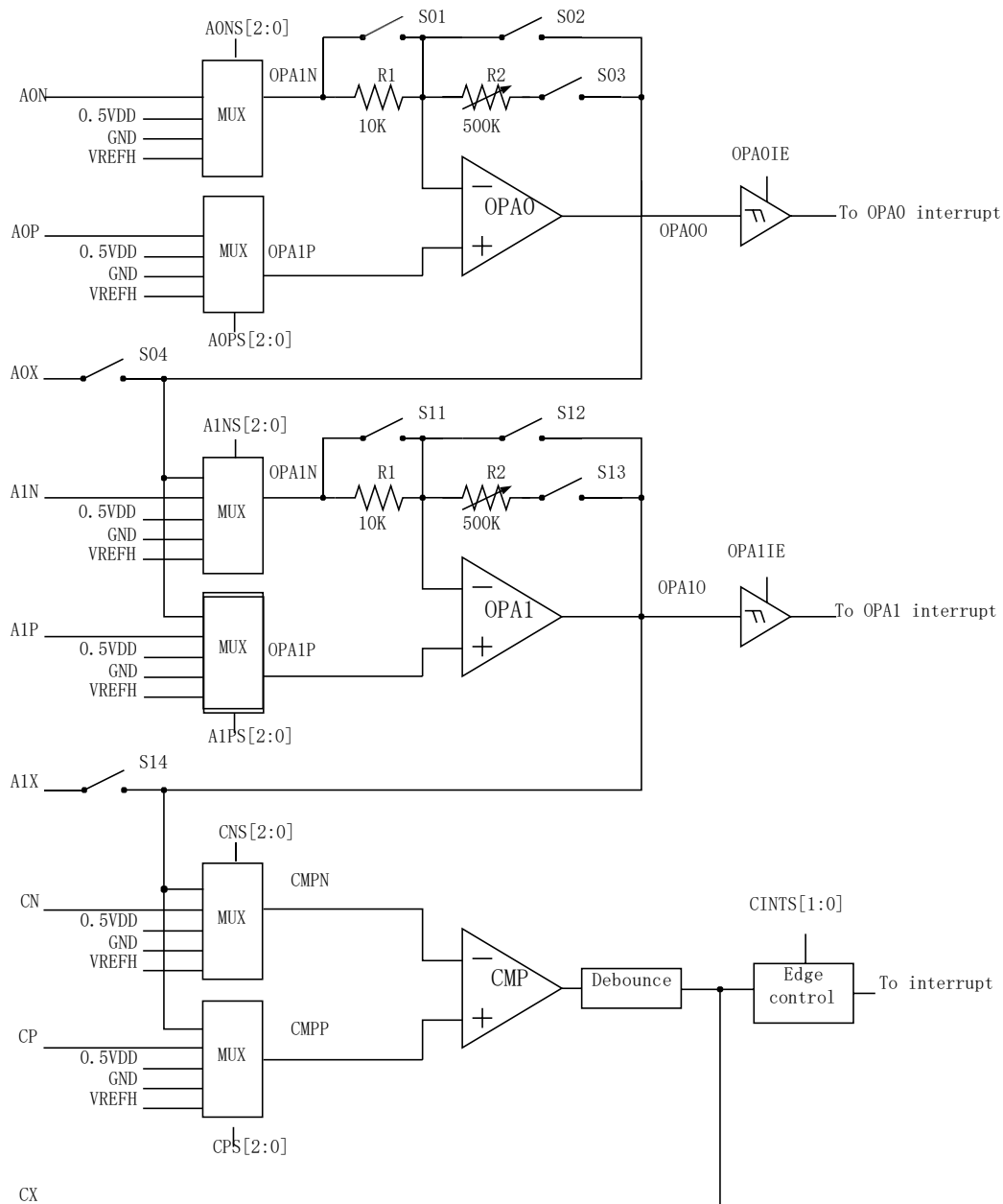


14 比较器(CMP)

14.1 概述

JZ8M4634 包含一个比较器，具有多种输入源，多种参考电压选项，输出极性控制，输出到定时计数器，多种输出中断触发和输出唤醒 MCU 功能，增强了使用的灵活性，适应各种广泛的应用。

14.2 比较器框图





14.3 CMPC0 比较器控制寄存器 0

1ABH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPC0	CMPEN	CMPOUT	CMPNS2	CMPNS1	CMPNS0	CMPPS2	CMPPS1	CMPPS0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **CMPEN**: 比较器使能控制位

1 = 使能比较器

0 = 关闭比较器

Bit 6 **CMPOUT**: 比较器输出位

1 = CP脚输入电压大于CN脚

0 = CP脚输入电压小于CN脚

Bit[5:3] **CMPNS[2:0]**: 比较器反相输入信号选择位

CMPNS [2:0]	输入信号选择
000	CMPNX
001	CMPN
010	运放输出 OPA10
011	VDD/2
100	VREF
101	GND
110	运放输出 OPA00
111	未定义

Bit[1:0] **CMPPS[2:0]**: 比较器正相输入信号选择位

CMPPS [2:0]	输入信号选择
000	CMPPX
001	CMPP
010	运放输出 OPA10
011	VDD/2
100	VREF
101	GND
110	运放输出 OPA00
111	未定义

注：设计者必须在使能比较器中断之前将比较器使能，以避免未知的中断发生。



14.4 CMPC1 比较器控制寄存器 1

1ACH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPC1	CMPOEN	-	CMPHIEN	CMPOFM	CMPOF3	CMPOF2	CMPOF1	CMPOF0
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	-	0	0	0	0	0	0

Bit 7 **CMPOEN**: 比较器输出使能
 1 = 比较器信号从端口输出
 0 = 关闭比较器信号输出

Bit 5 **CMPHIEN**: 比较器功耗选择
 1 = 大功耗
 0 = 小功耗

Bit[4] **CMPOFM**: 校准模式选择
 1 = 失调校准模式
 0 = 正常工作模式

Bit[3:0] **CMPOF[3:0]**: 校准参数位

14.5 COPAC 运放/比较器控制寄存器

1A7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
COPAC	CINTS1	CINTS0	-	A0GS	A0SW3	A0SW2	A0SW1	A0SW0
R/W	RW	RW	-	R/W	R/W	R/W	R/W	R/W
POR	0	0	-	0	0	0	0	0

Bit 7-6 **CINTS[1:0]**: 比较器中断触发类型选择

CINTS[1:0]	触发类型
00	下降沿触发
01	上升沿触发
10	双边沿触发
11	无



15 运算放大器(OPA)

15.1 概述

JZ8M4634 包含两个运算放大器 OPA，可用于用户特定的模拟信号处理。通过设置相应的控制寄存器，可使能或关闭运算放大器或实现如跟随器，同相放大器，反相放大器或各种滤波器等。

15.2 OPA0C0 运放 OPA0 控制寄存器 0

1ADH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPA0C0	OPA0EN	OPA0O	OPA0NS2	OPA0NS1	OPA0NS0	OPA0PS2	OPA0PS1	OPA0PS0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **OPAEN**: 运放使能控制位

1 = 使能运放

0 = 关闭运放

Bit 6 **OPA0O**: 运放输出位

1 = OPAP脚输入电压大于OPAN脚

0 = OPAP脚输入电压小于OPAN脚

Bit[5:3] **OPA0NS[2:0]**: 运放反相输入信号选择位

OPA0NS[2:0]	输入信号选择
000	OPA0NX
001	OPA0N
010	GND
011	VDD/2
100	VREF
101	GND
110	未定义
111	未定义

Bit[1:0] **OPA0PS[2:0]**: 运放正相输入信号选择位

OPA0PS[2:0]	输入信号选择
000	OPA0PX
001	OPA0P
010	GND
011	VDD/2
100	VREF
101	GND
110	未定义
111	未定义

注：设计者必须在使能运放中断之前将比较器使能，以避免未知的中断发生。



15.3 OPA0C1 运放 OPA0 控制寄存器 1

1AEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPA0C1	A0ENLO	OPA0VRC	OPAHIEN	OPAOFM	OPAOF3	OPAOF2	OPAOF1	OPAOF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7 **A0ENLO**: 运放逻辑输出使能
 1 = 使能运放逻辑输出功能
 0 = 屏蔽运放逻辑输出功能
- Bit 6 **OPA0VRC**: 运放OPAP端信号从端口输出
 1 = OPA P端信号从端口输出
 0 = 关闭
- Bit 5 **OPAHIEN**: 运放功耗选择
 1 = 大功耗
 0 = 小功耗
- Bit[4] **OPAOFM**: 校准模式选择
 1 = 失调校准模式
 0 = 正常工作模式
- Bit[3:0] **OPAPOF[3:0]**: 校准参数位

15.4 COPA0C 运放 OPA0 控制寄存器 2

1AFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
COPAC	CINTS1	CINTS0	-	OPAGS	A0SW3	A0SW2	A0SW1	A0SW0
R/W	RW	RW	-	R/W	R/W	R/W	R/W	R/W
POR	0	0	-	0	0	0	0	0

- Bit 4 **OPA0GS**: 运放内建放大倍数选择
 1 = 50倍放大
 0 = 10倍放大
- Bit[3:0] **A0SW[3:0]**: 运放通路开关选择

15.5 OPA1C0 运放 OPA1 控制寄存器 0

1A8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPA1C0	OPA1EN	OPA1O	OPA1NS2	OPA1NS1	OPA1NS0	OPA1PS2	OPA1PS1	OPA1PS0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7 **OPA1EN**: 运放使能控制位
 1 = 使能运放
 0 = 关闭运放



Bit 6 **OPA1O**: 运放输出位
 1 = OPAP脚输入电压大于OPAN脚
 0 = OPAP脚输入电压小于OPAN脚

Bit[5:3] **OPA1NS[2:0]**: 运放反相输入信号选择位

OPA1NS[2:0]	输入信号选择
000	A1NX
001	A1N
010	OPA00
011	VDD/2
100	VREF
101	GND
110	未定义
111	未定义

Bit[1:0] **OPA1PS[2:0]**: 运放正相输入信号选择位

OPA1PS[2:0]	输入信号选择
000	A1PX
001	A1P
010	OPA00
011	VDD/2
100	VREF
101	GND
110	未定义
111	未定义

注：设计者必须在使能运放中断之前将比较器使能，以避免未知的中断发Th。

15.6 OPA1C1 运放 OPA1 控制寄存器 1

1A9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPA1C1	A1ENLO	OPA1VRC	OPA1HIEN	OPA1OFM	OPA1OF3	OPA1OF2	OPA1OF1	OPA1OF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **A1ENLO**: 运放逻辑输出使能
 1 = 运放逻辑输出使能
 0 = 运放逻辑输出屏蔽

Bit 6 **OPA1VRC**: 运放OPAP端信号从端口输出
 1 = OPAP端信号从端口输出
 0 = 关闭

Bit 5 **OPA1HIEN**: 运放功耗选择
 1 = 大功耗
 0 = 小功耗

Bit[4] **OPA1OFM**: 校准模式选择
 1 = 失调校准模式



0 = 正常工作模式

Bit[3:0] OPA1POF[3:0]: 校准参数位

15.7 COPA1C 运放 OPA0 控制寄存器 2

1AAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
COPA1C			-	OPA1GS	A1SW3	A1SW2	A1SW1	A1SW0
R/W			-	R/W	R/W	R/W	R/W	R/W
POR			-	0	0	0	0	0

Bit 4 OPA1GS: 运放内建放大倍数选择

1 = 50倍放大

0 = 10倍放大

Bit[3:0] A1SW[3:0]: 运放通路开关选择



16 看门狗 (WDT)

16.1 概述

看门狗定时器的时钟为内部独立 RC 时钟。

配置字 WDTEN 设置看门狗定时器的三种工作状态：

- (1) 始终使能：在 STOP 模式下仍然工作，溢出可唤醒 STOP
- (2) STOP 下关闭
- (3) 始终关闭

配置字 TWDTEN 设置看门狗的四钟溢出时间：4.5ms、18ms、72ms 或 288ms。

16.2 OPTION 配置寄存器

1BCH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPTION	GIE	-	TO	PD	-	-	-	-
读/写	R/W	-	R	R	-	-	-	-
复位后	0	-	1	1	0	0	0	0

Bit.5 **TO:** 超时位
 1 = 上电复位或清除WDT
 0 = WDT发生溢出

Bit.4 **PD:** 掉电位
 1 = 上电复位或清除WDT
 0 = 进入休眠模式

16.3 WDTC 看门狗控制寄存器

1BEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDTC	WDTC7	WDTC6	WDTC5	WDTC4	WDTC3	WDTC2	WDTC1	WDTC0
读/写	W(*)	W(*)	W(*)	W(*)	W(*)	W(*)	W(*)	W(*)
复位后	-	-	-	-	-	-	-	-

(*) [1] WDTC 写入 0x5A 将清除 WDT 定时器，写入其他值无效。

[2] CLRWDTC 指令也可清除 WDT 定时器。



17 芯片配置字 (OPTION BIT)

烧录选项	内容	说明
OSCM1	HHS+LIRC	双系统时钟
	HXT+LIRC	
	HIRC+LXT	
	HIRC+LIRC	
HCPUS	2T	高频模式下 CPU 速度选择； 低频模式下固定为 2T
	4T	
	8T	
	16T	
	32T	
	64T	
	128T	
	256T	
LVRS	LVR=1.41V	系统高速运行时，请选择相应较高的 LVR 电压，以保证系统的可靠性
	LVR=1.51V	
	LVR=1.59V	
	LVR=1.70V	
	LVR=1.79V	
	LVR=1.89V	
	LVR=1.98V	
	LVR=2.10V	
	LVR=2.22V	
	LVR=2.32V	
	LVR=2.41V	
	LVR=2.51V	
	LVR=3.52V	
	LVR=3.62V	
LVR=3.71V		
WDTC	始终开启看门狗	
	休眠模式下关闭看门狗	
	始终关闭看门狗	
WDTT	WDT 溢出时间=4.5mS	VDD=5V 典型值
	WDT 溢出时间=18mS	
	WDT 溢出时间=72mS	
	WDT 溢出时间=288mS	
MCLRE	使能外部复位，对应管脚作为复位脚	
	屏蔽外部复位，对应管脚作为输入脚	



SPDUP	启动时钟为低速时钟	
	启动时钟为高速时钟	
RDPIN	Read From Pin	
	Read From Register	
CP	屏蔽代码保护功能	
	使能代码保护功能	



18 电性参数

18.1 极限参数

储存温度.....	-50℃~125℃
工作温度.....	-40℃~85℃
电源供应电压.....	VSS-0.3V~VSS+6.0V
端口输入电压.....	VSS-0.3V~VDD+0.3V

18.2 直流特性

符号	参数	测试条件		最小值	典型值	最大值	单位
		VDD	条件 (常温 25℃)				
VDD	工作电压	—	Fosc = 16MHz, 8T	1.8		5.5	V
IDD1	工作电流 1	3V	Fosc = 16MHz, 16T, 无负载,		1.0		mA
		5V			1.5		mA
IDD2	工作电流 2	3V	内部低频 RC, 高频振荡器关闭 低速模式, 无负载		350		uA
		5V			500		uA
IDD3	工作电流 3	3V	高频振荡器关闭, 外部低频晶 振打开 (32768), 进入 STOP, 定时器周期唤醒, 无负载		5		
		5V			11		
ISP1	静态电流	3V	休眠模式, WDT 使能, 无负载		3		uA
		5V			12		uA
ISP2	静态电流	3V	休眠模式, WDT 禁止, 无负载			1	uA
		5V				1	uA
V _{IL1}	输入低电平		有施密特			0.2VDD	
V _{IH1}	输入高电平		有施密特	0.8VDD			
I _{PH}	上拉电阻	5V	输入到 GND		TBD		uA
		3V	输入到 GND		TBD		
I _{OL1}	输出灌电流	5V	输出口, Vout = VSS+0.6V		5		mA
		3V			5		
I _{OH1}	输出拉电流	5V	输出口, Vout=VDD-0.6V	—	5	—	mA
		3V			5		mA
I _{OL2}	输出灌电流	5V	输出口, Vout=VSS+0.6V		10		mA
		3V			10		
I _{OH2}	输出拉电流	5V	输出口, Vout=VDD-0.6V	—	10	—	mA
		3V			10		mA



I _{OL3}	输出灌电流	5V	输出口, V _{out} =V _{SS} +0.6V		35		mA
		3V			35		
I _{OH3}	输出拉电流	5V	输出口, V _{out} =V _{DD} -0.6V	—	35	—	mA
		3V			35		mA
I _{OL4}	输出灌电流	5V	输出口, V _{out} =V _{SS} +0.6V		70		mA
		3V			70		
I _{OH4}	输出拉电流	5V	输出口, V _{out} =V _{DD} -0.6V	—	70	—	mA
		3V			70		mA