

JZ8M2626

8BIT
TK+AD 型
MTP MCU



本公司不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，本公司的产品不是专门设计来应用于外科植入、生命维持和任何本公司产品的故障会对个体造成伤害甚至死亡的领域。如果将本公司的产品应用于上述领域，即使这些是由本公司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，并且用户保证本公司及其雇员、子公司、分支机构和销售商与上述事宜无关。



目录

1 产品简述	6
1.1 特性	6
1.2 引脚图	7
1.2.1 SOP20	7
1.2.2 SOP16	7
1.3 引脚描述	8
2 中央处理器（CPU）	10
2.1 程序存储器	10
2.1.1 复位向量（0000H）	10
2.1.2 中断向量（0008H）	11
2.1.3 查表	11
2.1.4 EECON1 状态寄存器	12
2.1.5 EECON2 状态寄存器	12
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	15
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 IO 口变化中断使能寄存器	21
5.4 INTCR0 中断控制寄存器 0	21
5.5 INTF0 中断标志寄存器 0	22



5.6 INTCR1 中断控制寄存器 1.....	23
5.7 INTF1 中断标志寄存器 1.....	23
6 端口.....	24
6.1 IOA.....	24
6.2 IOB.....	25
6.3 IOC.....	26
6.7 IOIDS 端口驱动设置寄存器.....	26
7 定时器 0/1(TC0/1).....	28
7.1 概述.....	28
7.2 TxCR 控制寄存器.....	29
7.3 TCxCL TCx 计数器低 8 位/周期寄存器.....	29
7.4 TCxCH TCx 计数器高位.....	30
8 脉宽调制模块 PWM0.....	31
8.1 概述.....	31
8.2 PWM0CR 控制寄存器.....	31
8.3 PWM0D 数据高位.....	31
9 脉宽调制模块 PWM1.....	32
9.1 概述.....	32
9.2 PWM1CR 控制寄存器.....	32
9.3 PWM1S 控制寄存器.....	33
9.3 PWM1DH 数据高位.....	33
9.4 PWM1DL 数据低位.....	34
9.5 PWMDEADT PWM 死区控制寄存器.....	34
9.6 PWM 输出波形示例.....	34
9.6.1 互补 PWM 输出.....	34
9.6.2 带死区的互补 PWM 输出.....	35
9.6.3 有效电平选择.....	35
10 通用串行通讯口 (USART)	36
10.1 概述.....	36
10.2 TXCR 发送控制寄存器.....	36
10.3 TXREG 发送数据寄存器.....	36
10.4 RXCR 接收控制寄存器.....	37
10.5 RXREG 接收数据寄存器.....	37
10.6 BRGDH 波特率寄存器高位.....	37
10.7 BRGDL 波特率寄存器低位.....	38
10.8 USART 使用说明.....	38
10.8.1 波特率设置.....	38
10.8.2 异步发送.....	38
10.8.3 异步接收.....	40
10.8.4 同步发送.....	41



10.8.5 同步接收.....	42
10.8.6 唤醒及休眠模式下通讯.....	43
11 串行通讯口 (I2C)	44
11.1 概述.....	44
11.2 通讯波形示意.....	44
11.3 I2CCON I2C 控制寄存器.....	44
11.4 I2ADDR I2C 控制寄存器.....	45
11.4 I2CBUF 数据寄存器.....	45
11.5 唤醒及休眠模式下通讯.....	45
11.6 通讯波形图.....	46
11.7 应用示例.....	47
11.7.1 从机软件流程图.....	47
11.7.1 例程.....	47
12 触摸按键 (CDC)	50
12.1 概述.....	50
12.2 原理框图.....	50
12.3 TKCTR0 控制寄存器 0.....	51
12.4 TKCTR1 控制寄存器 1.....	51
12.5 TKCHSH/L 触摸按键通道选择寄存器.....	51
12.6 TKCNTH/L 触摸按键计数寄存器.....	52
12.7 操作说明.....	52
13 模数转换器(ADC).....	53
13.1 概述.....	53
13.2 ADCON0 寄存器.....	53
13.3 ADCON1 寄存器.....	54
13.4 ADCON2 寄存器.....	54
13.5 ADH ADC 数据高位.....	55
13.6 ADL ADC 数据低位.....	55
13.7 AD 转换时间.....	55
14 看门狗 (WDT)	56
14.1 概述.....	56
14.2 OPTION 配置寄存器.....	56
14.3 WDTC 看门狗控制寄存器.....	56
15 电性参数.....	57
15.1 极限参数.....	57
15.2 直流特性.....	57



1 产品简述

- JZ8M2626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2K×16位多次擦写编程存储器（MTP，擦写次数1000次），128×8位EEPROM（擦写次数10000次），128×8位的数据存储器（RAM），18个双向I/O口，2个8/16位Timer定时器/计数器，1路UART，1路IIC SLAVE，1路8位PWM，1路8+4位分辨率的互补PWM，12+3路12位AD转换器，11路触摸按键，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
- 2K×16位的MTP程序存储器
- 128×8位的EEPROM
- 128×8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18个双向I/O口
- 所有端口可设置弱上拉
- IOB端口可设置弱下拉
- IOB口变化中断
- 两路外部中断

■ 2 个定时器/计数器

- TC0/TC1：具有自动装载功能的定时/计数器

■ 2 路 PWM

- PWM0：8位分辨率PWM
- PWM1：8+4位分辨率PWM，带死区控制及互补输出

■ 系统时钟

- 内部高速RC震荡器：16MHz
- 内部低速RC震荡器：32KHz/500KHz
- 外部高速晶体震荡器：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

■ 12+3 路 12 位 ADC

- 内嵌参考电压2V、3V、4V、VDD
- 12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

■ 触摸按键模块

- 休眠电流小于10uA

■ 多路中断/唤醒源

- 定时器中断：TC0/TC1
- IOB口变化中断
- 触摸按键中断
- ADC转换中断
- 外部中断

■ 看门狗定时器

■ 特殊功能

- 可编程代码保护
- ISP功能

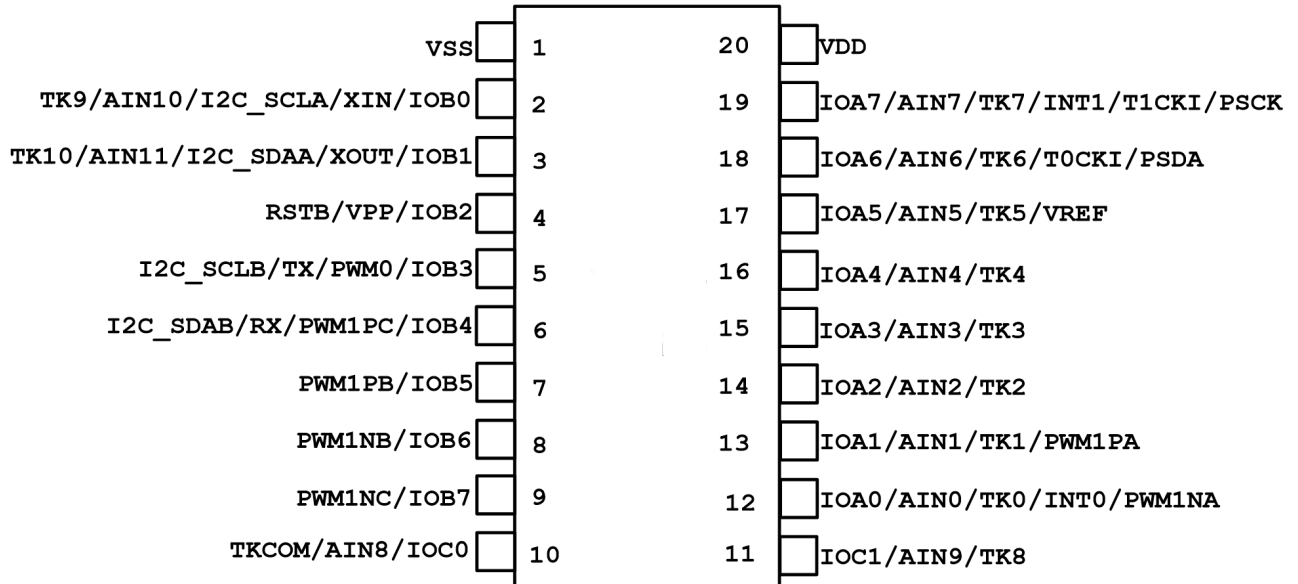
■ 封装形式

- SOP20
- SOP16

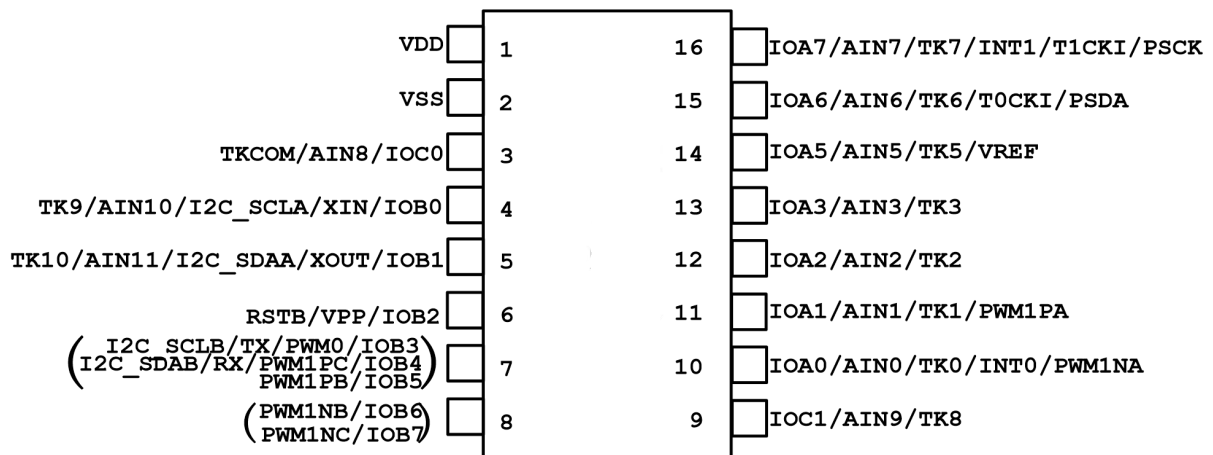


1.2 引脚图

1.2.1 SOP20



1.2.2 SOP16



注：如 B6、B7 为硬件相连的



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
INT0	I	外部中断
PWM1NC	O	PWM1 反相输出
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
PWM1PC	O	PWM1 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
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
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
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
T0CKI	I	TC0 外部时钟输入
PSDA	I/O	编程用
IOA7	I/O	输入/输出 IO, SMT, 上拉电阻
AIN7	A	AD 通道 7
TK7	A	触摸按键通道 7
INT1	I	外部中断
T1CKI	I	TC1 外部时钟输入
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
AIN10	A	AD 通道 10
TK9	A	触摸按键通道 9
I2C_SCL	I/O	I2C 通讯时钟口



IOB1	I/O	输入/输出 IO, SMT, 上拉电阻
XOUT	A	外部晶体振荡器接口
AIN11	A	AD 通道 11
TK10	A	触摸按键通道 10
I2C_SDAA	I/O	I2C 通讯数据口
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
PWM0	O	PWM0 输出
TX	O	通用异步串行通讯发送口
I2C_SCLB	I/O	I2C 通讯时钟口
IOB4	I/O	输入/输出 IO, SMT, 上拉电阻
PWM1PB	O	PWM1 正相输出
RX	I	通用异步串行通讯接收口
I2C_SDAB	I/O	I2C 通讯数据口
IOB5	I/O	输入/输出 IO, SMT, 上拉电阻
PWM1PA	O	PWM1 正相输出
IOB6	I/O	输入/输出 IO, SMT, 上拉电阻
PWM1NA	O	PWM1 反相输出
IOB7	I/O	输入/输出 IO, SMT, 上拉电阻
PWM1NB	O	PWM1 反相输出
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
AIN9	A	AD 通道 9
TK8	A	触摸按键通道 8

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 ~ 0x07EF	用户区
0x07F0 ~ 0x07FF	厂商保留区 (16x16 Bits)
0x0800 ~ 0x087F	EEPROM

2.1.1 复位向量（0000H）

JZ8M2626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）

JZ8M2626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
```



2.1.4 EECON1 状态寄存器

0F0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EECON1	EEMOD	-	-	-	-	-	WERR	EEWEN
读/写	R/W	-	-	-	-	-	R	R/W
复位后	0	-	-	-	-	-	0	0

Bit 7 **EEMOD**: EEPROM写入模式使能

1 = 使能EEPROM写入模式

0 = 退出EEPROM写入模式

Bit 1 **WERR**: EEPROM写错误标志

1 = EEPROM写入失败

0 = 未发生EEPROM写入失败

Bit 0 **EEWEN**: EEPROM写入使能.

1 = 使能EEPROM写入

0 = 退出EEPROM写入

注: 所有非EECON1和EECON2的寄存器写入操作, 将使EEWEN清零

2.1.5 EECON2 状态寄存器

0F1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EECON2	-	-	-	-	EELOCK3	EELOCK2	EELOCK1	EEWE
读/写	-	-	-	-	W	W	W	W
复位后	-	-	-	-	0	0	0	0

Bit 3 **EELOCK3**: 解锁流程1

在EECON1[0]=1, EECON2 =00H时, 使用置位指令置位EELOCK3, 完成EE或MTP写入解锁流程1, 完成该操作后, EECON2的值为08H

Bit 2 **EELOCK2**: 解锁流程2

在EECON1[0]=1, EECON2 =08H时, 使用置位指令置位EELOCK2, 完成EE或MTP写入解锁流程2, 完成该操作后, EECON2的值为04H

Bit 1 **EELOCK1**: 解锁流程3

在EECON1[0]=1, EECON2 =04H时, 使用置位指令置位EELOCK2, 完成EE或MTP写入解锁流程2, 完成该操作后, EECON2的值为02H

Bit 0 **EEWE**: 启动写入操作

在EECON1[0]=1, EECON2 =02H时, 使用置位指令置位EEWE, 启动EE写入操作, 写入操作完成后, EECON2=00H

**例：写 EE**

```
MOVIA    HIGH(DTAB)    ;获取EE地址高位
MOVAR    FSR1           ;设置EE地址高位指针
MOVIA    LOW(DTAB)     ;获取EE地址低位
MOVAR    FSR0           ;设置EE地址低位
MOVIA    DATAH         ;获取EE数据高位
MOVAR    HBUF           ;设置EE数据低位
MOVIA    DATAL

BSET     bEEMOD          ;读写EE/MTP时需置位
BSET     bEEWEN          ;使能EE/MTP写入
BSET     bEELCK3         ;EE/MTP写入解锁流程1
BSET     bEELCK2         ;EE/MTP写入解锁流程2
BSET     bEELCK1         ;EE/MTP写入解锁流程3
BSET     bEEWR           ;写入
                        ;CPU暂停，等待EE写入结束

NOP
BCLR     bEEMOD          ;写完EE，清除该位
JBTS0    bWERR
GOTO     WeeErrProcess   ;//写入失败，跳转到错误处理程序
...
...
```

注：EEPROM或MTP写入时，请关闭所有唤醒源，不然唤醒源可能中断写入操作，造成写入失败

例：读 EE

```
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

BSET     bEEMODE        ;//读EE，需置位
RDT      ;读取表的第一个数据0x0102
BCLR     bEEMODE        ;//清除该位
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
```



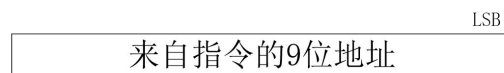
2.2 数据存储器

2.2.1 数据存储器结构

地址	间接寻址 INDF0	间接寻址 INDF1	间接寻址 INDF2	直接寻址
0X100 ~ 0x1FF	NA	YES	YES	YES
0x000 ~ 0x0FF	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 ~ 0x070	GPR							
0x080 ~ 0x0A0	RESERVE							
0x0A8								
0x0B0	INDF0	FSR0	TKCTR0	TKCTR1	TKCNTL	TKCNTH	TKCHSL	TKCHSH
0x0B8	INDF1	FSR1	PCL	STATUS	OPTION	OSCM	WDTC	IOBICR
0x0C0	INDF2	HBUF	PDB		INTCR0	INTF0	INTCR1	INTF1
0x0C8	IOA	OEA	PUA	ANSA	IOB	OEB	PUB	ANSB
0x0D0	IOC	OEC	PUC	ANSC	IOIDS	I2CADR	I2CCON	I2CBUF
0x0D8	PWM0CR	PWM0D	TXCR	TXREG	RXCR	RXREG	BRGDH	BRGDL
0x0E0	ADCON0	ADCON1	ADL	ADH	ADCON2	T1CR	TC1CL	TC1CH
0x0E8	T0CR	TC0CL	TC0CH	PWMDEADT	PWM1S	PWM1CR	PWM1DL	PWM1DH
0x0F0	EECON1	EECON2						
0x0F8			VREFCAL		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

0BAH	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	0	0	0	1

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（ADDAR PCL、ADCAR 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

0BBH	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)

JZ8M2626 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

JZ8M2626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）或内置的低速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

0BDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCM	STBH	STBL	HSPDX2	STOP	CLKM	STPH	LIRC500K	STPL
读/写	RW	RW	R/W	RW	RW	RW	RW	RW
复位后	x	x	0	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 **HSPDX2:** 高频振荡器倍频（仅用于定时器选用高频作为时钟）
 1 = 倍频使能
 0 = 不使能倍频
- 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

0FCH	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 震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

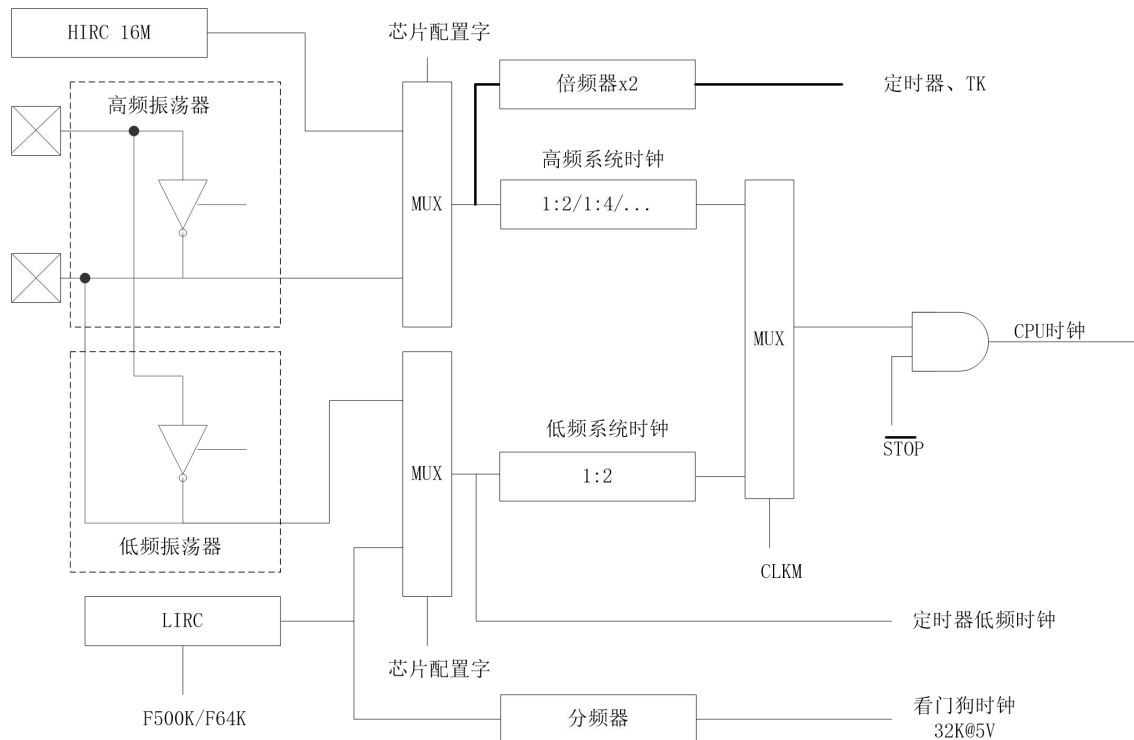
TASK_IRCCAL:

```
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

JZ8M2626有多路中断源: TC0/TC1, IOB口变化中断, UART发送/接收中断, ADC中断, INT0、INT1 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 来使能中断, 查询 INTF0/INTF1 中断标志寄存器判断中断是否发生。

5.2 OPTION 配置寄存器

0BCH	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)

5.3 IO 口变化中断使能寄存器

0BFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOBICR	IOBICR7	IOBICR7	IOBICR7	IOBICR7	IOBICR7	IOBICR7	IOBICR7	IOBICR7
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-Bit.0 **IOBICRn** IO端口变化中断使能
 1 = 使能IOB.n口电平变化中断
 0 = 屏蔽IOB.n口电平变化中断

5.4 INTCR0 中断控制寄存器 0

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



Bit 7	TKIE: 1 = 使能触摸按键中断1 0 = 屏蔽触摸按键中断1
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	I2CIE: 1 = 使能I2C中断 0 = 屏蔽I2C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

5.5 INTF0 中断标志寄存器 0

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

(注：除UART之外的所有中断标志位需软件清零)

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



- 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

5.6 INTCR1 中断控制寄存器 1

0C6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCR1	-	-	-	-	INT1IE	INT0IE	-	IOBCHIE
读/写	-	-	-	-	R/W	R/W	-	R/W
复位后	-	-	-	-	0	0	-	0

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

5.7 INTF1 中断标志寄存器 1

0C7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF1	-	-	-	-	INT1IF	INT0IF	-	IOBCHIF
读/写	-	-	-	-	R/W	R/W	-	R/W
复位后	-	-	-	-	0	0	-	0

（注：所有中断标志位需软件清零）

- Bit.3 **INTF1IF:**
 1 = 产生外部中断INT1
 0 = 未产生外部中断INT1
- Bit.2 **INTF0IF:**
 1 = 产生外部中断INT0
 0 = 未产生外部中断INT0
- Bit.0 **IOBCHIF:**
 1 = 对应输入端口状态发生变化
 0 = 对应输入端口状态未发生变化



6 端口

6.1 IOA

IOA 数据寄存器

0C8H	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	X	X	X	X	X	X	X	X

IOA 方向寄存器

0C9H	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

0CAH	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 端口模式控制寄存器

0CBH	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

0CCH	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	X	X	X	X	X	X	X	X

IOB 方向寄存器

0CDH	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

IOB 上拉使能寄存器

0CEH	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

0C2H	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 端口模式控制寄存器

0CFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ANSB	-	-	-	-	-	-	ANSB1	ANSB0
读/写	-	-	-	-	-	-	R/W	R/W
复位后	-	-	-	-	-	-	0	0

Bit[1:0] **ANSB:** B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

注: 同一 IO 口上下拉电阻同时打开时, IO 口将自动屏蔽输入功能 (读该端口状态为 0), 此时端口电平接近于 $V_{DD}/2$.



6.3 IOC

IOC 数据寄存器

0D0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOC	-	-	-	-	-	-	IOC1	IOC0
读/写	-	-	-	-	-	-	RW	RW
复位后	-	-	-	-	-	-	X	X

IOC 方向寄存器

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

0D2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PUC	-	-	-	-	-	-	PUB1	PUB0
读/写	-	-	-	-	-	-	R/W	R/W
复位后	-	-	-	-	-	-	0	0

Bit[7:0] **PUC:** C口上拉使能

1 = 上拉使能

0 = 上拉关闭

IOC 端口模式控制寄存器

0D3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ANSC	-	-	-	-	-	-	ANSC1	ANSC0
读/写	-	-	-	-	-	-	R/W	R/W
复位后	-	-	-	-	-	-	0	0

Bit[1:0] **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

6.7 IOIDS 端口驱动设置寄存器

0D4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOIDS	IOBHIS	IOBLIS	IOAHIS	IOALIS	IOBHDS	IOBLDS	IOAHDS	IOALDS
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] **IOBHIS:** IOB7-4翻转电平选择

IOBHIS	IOB[7:4] 电平选择
0	SMT
1	低反转点

Bit[6] **IOBLIS:** IOB3-0翻转电平选择

IOBLIS	IOB[3:0] 电平选择
0	SMT
1	低反转点

Bit[5] **IOAHIS:** IOA7-4翻转电平选择

IOAHIS	IOA[7:4] 电平选择
0	SMT
1	低反转点

Bit[4] **IOALIS:** IOA3-0翻转电平选择

IOAHIS	IOA[3:0] 电平选择
0	SMT
1	低反转点

Bit[3] **IOBHDS:** IOB7-4输出驱动电流选择

IOBHDS	IOB[7:4] 输出驱动电流选择
0	大驱动
1	小驱动

Bit[2] **IOBLDS:** IOB3-0输出驱动电流选择

IOBLDS	IOB[3:0] 输出驱动电流选择
0	大驱动
1	小驱动

Bit[1] **IOAHDS:** IOA7-4输出驱动电流选择

IOAHDS	IOA[7:4] 输出驱动电流选择
0	大驱动
1	小驱动

Bit[0] **IOALDS:** IOA3-0输出驱动电流选择

IOALDS	IOA[3:0] 输出驱动电流选择
0	大驱动
1	小驱动



7 定时器0/1(TC0/1)

7.1 概述

JZ8M2626 TC0/TC1 为带有可设置 1:128 预分频器及周期寄存器的 8 位/16 位定时计数器，具有休眠状态下唤醒功能。

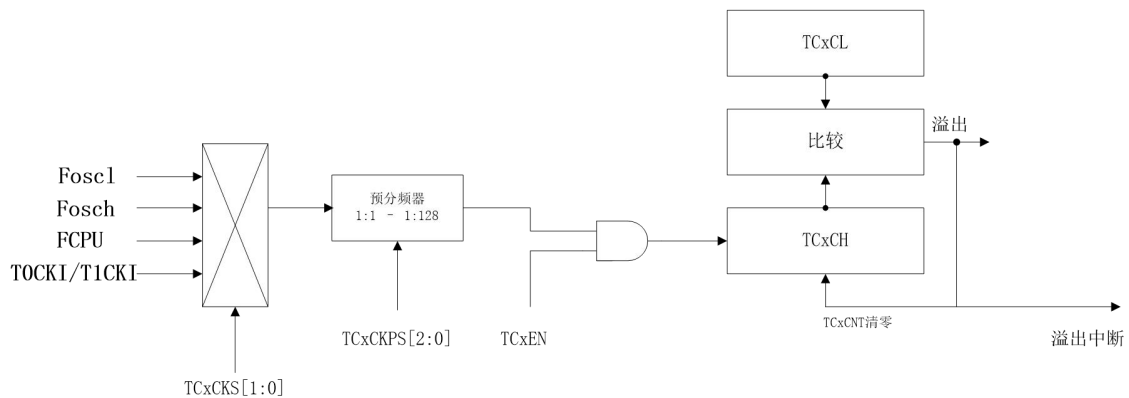
在 8 位模式下，TCxCL 作为 TCx 的周期寄存器，TCx 使能后，TCxCH 递加，当 TCxCH 与 TCxCL 数值相等时，TCx 溢出，将 TCxCH 清零重新开始计数，同时将中断标志位 TxIF 置 1。

在 16 位模式下，[TC0CH, TC0CL]作为 16 位的计数器，TC0 使能后，16 位计数器递加，当计数值等于 0xFFFF 时，16 位计数器将清零重新开始计数，同时将中断标志位 TCxIF 置 1。

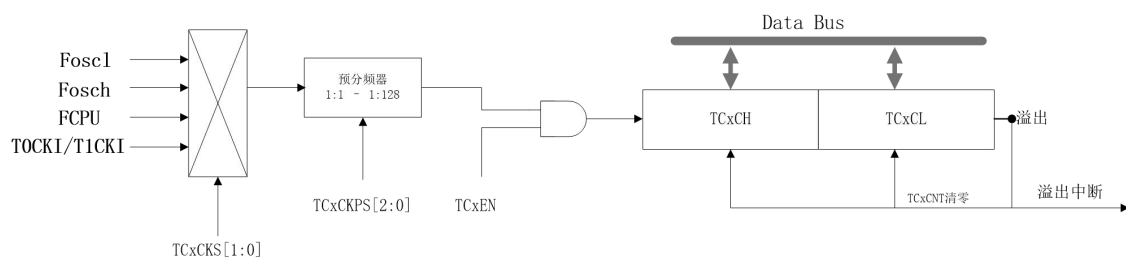
- 可选择时钟源，高频系统时钟 Fosch、低频系统时钟 Fosc1 和指令时钟 Fcpu 、外部时钟 TxCKI
- 可选择 8 位模式和 16 位模式
 - ✓ 8 位模式下，通过设置周期寄存器，可任意设置 TC0 的周期
- 预分频比多级可选，最大可选择 1:128
- 溢出中断功能
- 溢出中断唤醒功能（当输入频率选择 Fosc1 或 Fosch 时，若所选择的时钟源振荡器一直工作，此时 TC0 在休眠状态下依然工作，溢出中断可唤醒 CPU）

TC0/TC1 框图

8 位模式



16 位模式





7.2 TxCR 控制寄存器

0E8/0E5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TxCR	TCxEN	TCxMOD	-	TCxCKS1	TCxCKS0	TCxCKPS2	TCxCKPS1	TCxCKPS0
读/写	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
复位后	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 **TCxMOD:** TCx模式选择位

1 = 16位模式

0 = 8位模式

Bit 5 未定义

Bit[4:3] **TCxCKS:** TCx时钟源选择

TCxCKS[1:0]	TCx 时钟源选择
00	Fosc1 (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 8 位/周期寄存器

0E9/0E6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCxCL	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

0EA/0E7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCxCH	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



8 脉宽调制模块PWM0

8.1 概述

JZ8M2626 有一个 8 位分辨率 PWM0，时基使用 TC0。PWM 功能如下：

8.2 PWM0CR 控制寄存器

0D8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0CR	PWM0EN	PWM0OE	-	-	-	-	-	-
读/写	R/W	R/W	-	-	-	-	-	-
复位后	0	0	-	-	-	-	-	-

Bit 7 **PWM0EN:** PWM0模块使能位

1 = 使能PWM0

0 = 关闭PWM0

Bit 6 **PWM0OE:** PWM0输出控制

1 = PWM0信号从管脚输出

0 = PWM0信号不从管脚输出，管脚用做IO

8.3 PWM0D 数据高位

0D9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0D	PWM0D7	PWM0D6	PWM0D5	PWM0D4	PWM0D3	PWM0D2	PWM0D1	PWM0D0
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 脉宽调制模块PWM1

9.1 概述

JZ8M2626 有 1 路带有死区控制的 PWM，时基使用 TC1, 可独立进行设置，8+4 位分辨率。

- 正常 PWM 模式（定时器 1 需设置为 8 位模式）
- 单脉冲模式（定时器 1 需设置为 16 位模式）
- 互补输出
- 死区控制
- 12 位分辨率模式
- 8+4 分辨率模式
- 多个管脚输出

9.2 PWM1CR 控制寄存器

0EDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1CR	PWM1EN	PWM1POE	PWM1NOE	PWM1PAS	PWM1NAS	-	PWM1M	PWM1T
读/写	R/W	R/W	R/W	R/W	R/W	-	RW	R/W
复位后	0	0	0	0	0	-	0	0

- Bit 7 **PWM1EN:** PWM1模块使能位
 1 = 使能PWM1
 0 = 关闭PWM1
- Bit 6 **PWM1POE:** PWM1正相波形输出使能位
 1 = 端口输出PWM1P波形
 0 = 端口用作IO
- Bit 5 **PWM1NOE:** PWM1反相波形输出使能位
 1 = 端口输出PWM1N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 **PWM1NAS:** PWM1N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 1 **PWM1M:** PWM1模式选择
 1 = 单脉冲模式（需将定时器1设置成16位模式）
 0 = 正常PWM模式（需将定时器1设置成8为模式）
- Bit 0 **PWM1T:** PWM1触发事件设置
 1 = T1CKI上升沿/下降沿
 0 = T1CKI上升沿



注：单脉冲模式下，当 T1CK1 输入 n 个事件（上升沿或下降沿），PWM1DH、PWM1DL 作为 PWM 脉冲宽度。

- $PulsWidth = 62.5ns \times \text{定时器 1 的分频比} \times PWM1D$
若定时器 1 的分频比为 16，则可设脉宽的最大宽度为 4ms
- PWMDEADT[2:0] 作为触发信号的滤波时间设置，触发信号的采样时钟为定时器的计数时钟
若定时器 1 的分频比为 16，则可设的最大的滤波时间为 8us
- PWMDEADT[7:3] 作为触发事件设置，可选 16 个触发事件中产生的脉冲个数

PWMDEADT[7:3]	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
PWMDEADT[7]	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
PWMDEADT[6]		●		●		●		●		●		●		●		●
PWMDEADT[5]			●				●				●				●	
PWMDEADT[4]					●							●				
PWMDEADT[3]								●								

9.3 PWM1S 控制寄存器

0ECH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1S	-	-	-	-	-	PWM1C	PWM1B	PWM1A
读/写	-	-	-	-	-	R/W	RW	RW
复位后	-	-	-	-	-	0	1	0

- Bit 2 **PWM1C:** PWM1C 输出管脚设置
1 = PWM1PC/PWM1NC 用作PWM输出
0 = PWM1PC/PWM1NC 用作IO口
- Bit 1 **PWM1B:** PWM1B 输出管脚设置
1 = PWM1PB/PWM1NB 用作PWM输出
0 = PWM1PB/PWM1NB 用作IO口
- Bit 0 **PWM1A:** PWM1 输出管脚设置
1 = PWM1PA/PWM1NA 用作PWM输出
0 = PWM1PA/PWM1NA 用作IO口

9.3 PWM1DH 数据高位

0EFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DH	PWM1DH7	PWM1DH6	PWM1DH5	PWM1DH4	PWM1DH3	PWM1DH2	PWM1DH1	PWM1DH0
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



9.4 PWM1DL 数据低位

0EEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DL	PWM1DL7	PWMxDL6	PWMxDL5	PWMxDL4	-	-	-	-
读/写	R/W	R/W	R/W	R/W	-	-	-	-
复位后	0	0	0	0	-	-	-	-

9.5 PWMDEADT PWM 死区控制寄存器

0EBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDEADT	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

死区时间设置：

$$T_{deadtr} = DEADTR * \text{时基时钟周期}$$

$$T_{deadtf} = DEADTF * \text{时基时钟周期}$$

注：

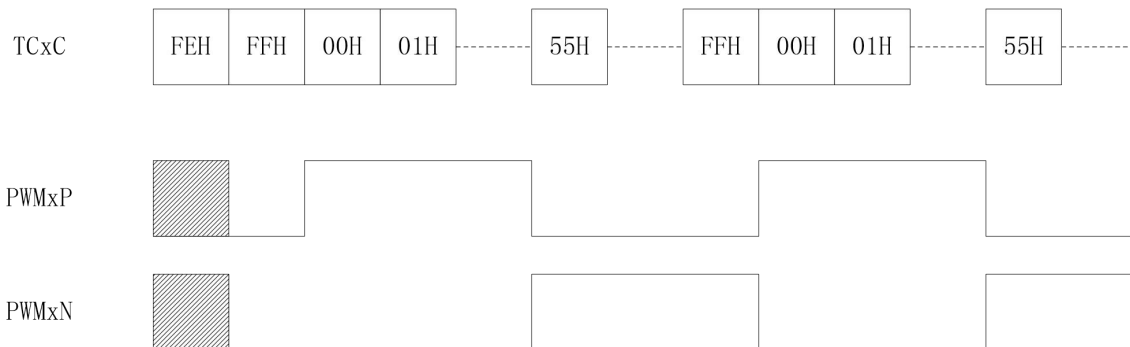
(1) 时基时钟周期即各 PWM 所选择的时钟源经预分频之后的时钟周期。

(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。

9.6 PWM 输出波形示例

9.6.1 互补 PWM 输出

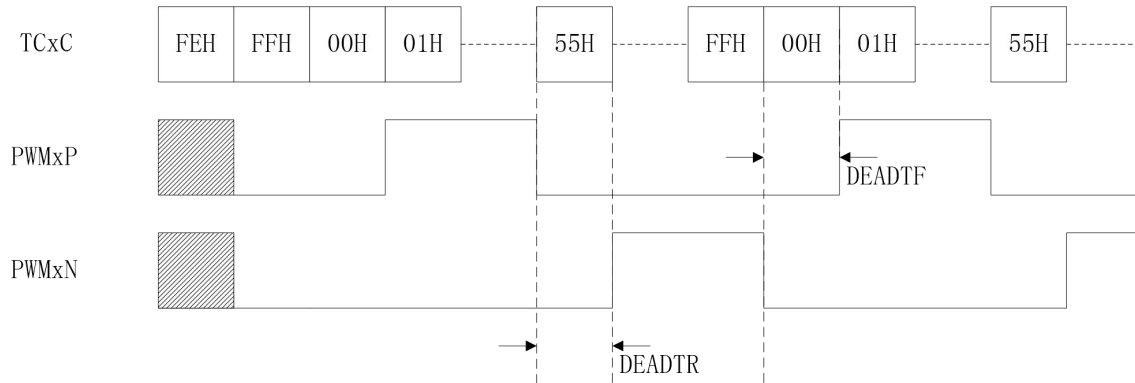
例: PWMxCR=11100000B PWMxDH=55H, PWMxDL=0H, TC0CL=FFH





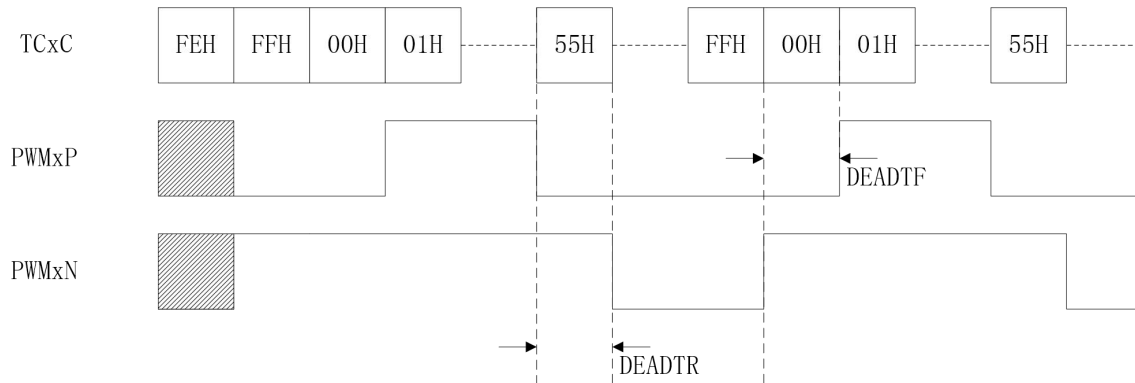
9.6.2 带死区的互补 PWM 输出

例: PWMxCR=11100100B PWMxDH=55H, PWMxDL=0H, DEADT=33H, TC0CL=FFH



9.6.3 有效电平选择

例: PWMxCR=11101100B PWMxDH=55H, PWMxDL=0H, TC0CL=FFH





10 通用串行通讯口 (USART)

10.1 概述

JZ8M2626 支持异步全双工模式和同步半双工模式。

10.2 TXCR 发送控制寄存器

0DAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TXCR	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 TXREG 发送数据寄存器

0DBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TXREG	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 RXCR 接收控制寄存器

0DCH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RXCR	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:** 使能接收
 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 = 停止同步接受
- 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 RXREG 接收数据寄存器

0DDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RXREG	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

0DEH	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 BRGDL 波特率寄存器低位

0DFH	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 ($F_{osc}=16\text{MHz}$)

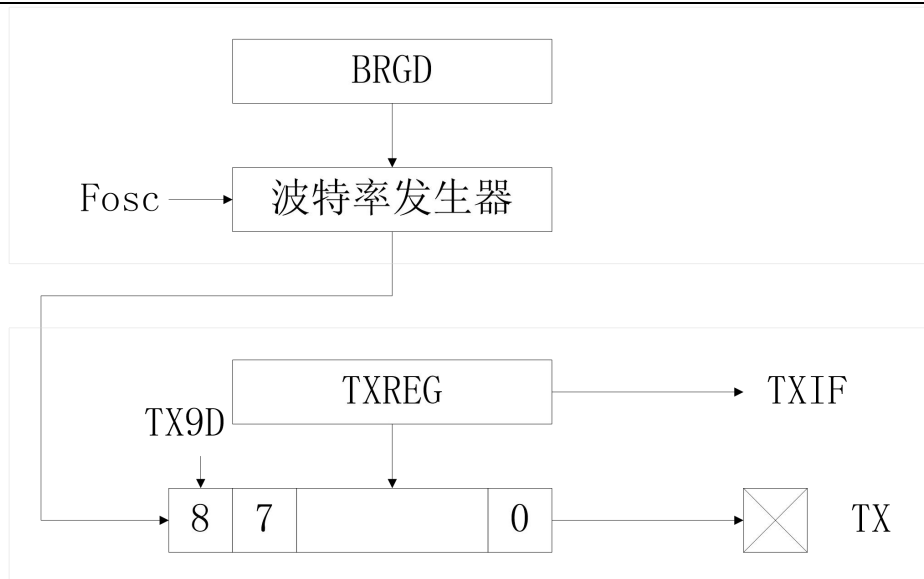
常用波特率	N (波特率分频比)	BRGD	偏差
300	256	0xCF	0.17%
600	256	0x67	0.17%
1200	64	0xCF	0.16%
2400	64	0x67	0.16%
4800	16	0xCF	0.16%
9600	16	0x67	0.16%
19200	4	0xCF	0.16%
38400	4	0x67	0.16%
57600	4	0x44	0.64%
115200	4	0x21	2.12%

10.8.2 异步发送

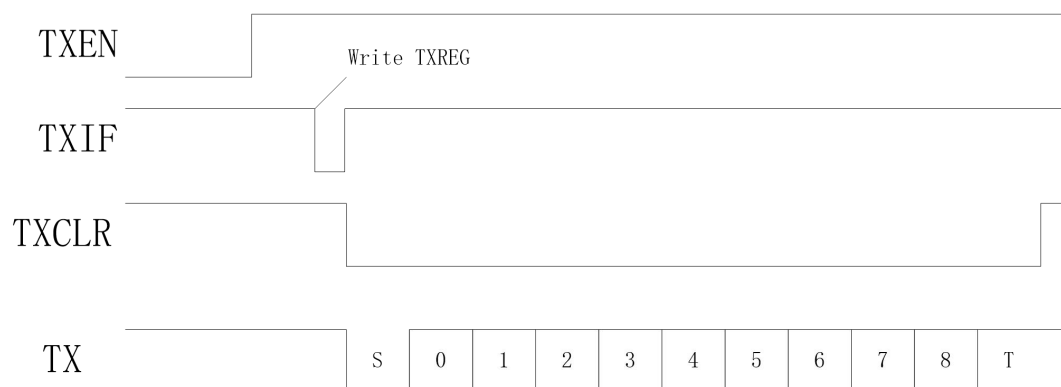
当 TXEN 使能时, TXIF 中断标志为 1 说明 TXREG 发送寄存器为空, TMCLR=1 说明发送移位寄存器为空, 发送器处于空闲状态。

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。

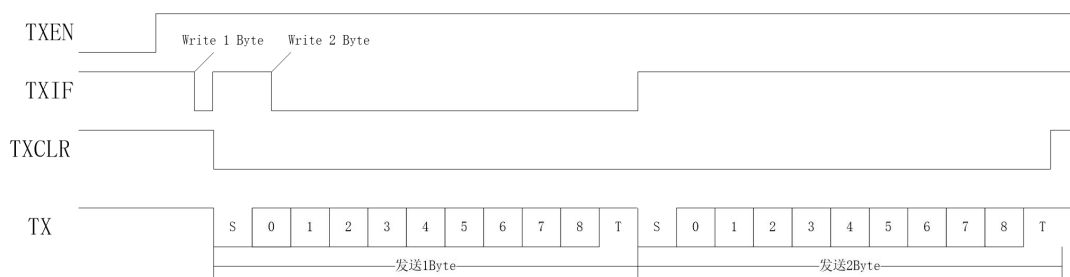
当 TXIF 为 0 时写入 TXREG, 将覆盖上次写入数据。



单字节发送:



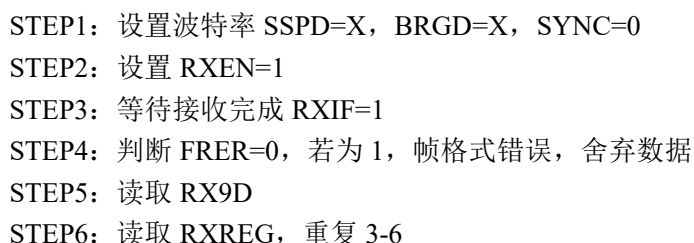
多字节发送:



- 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



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。



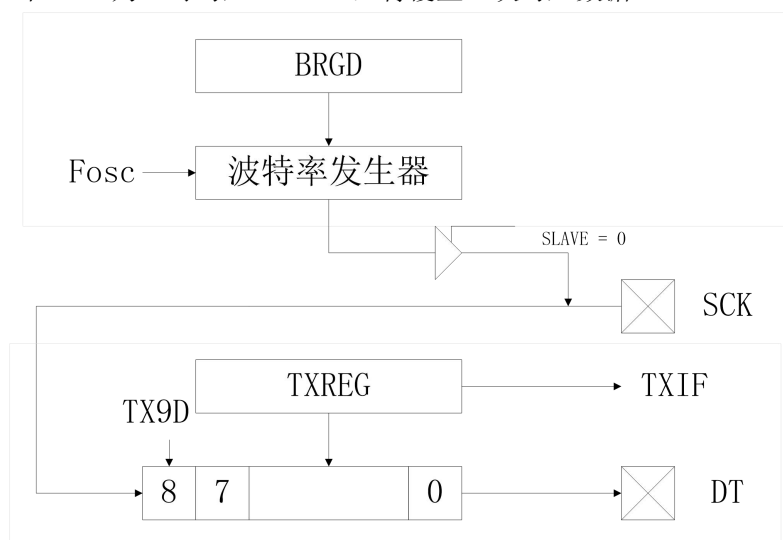


10.8.4 同步发送

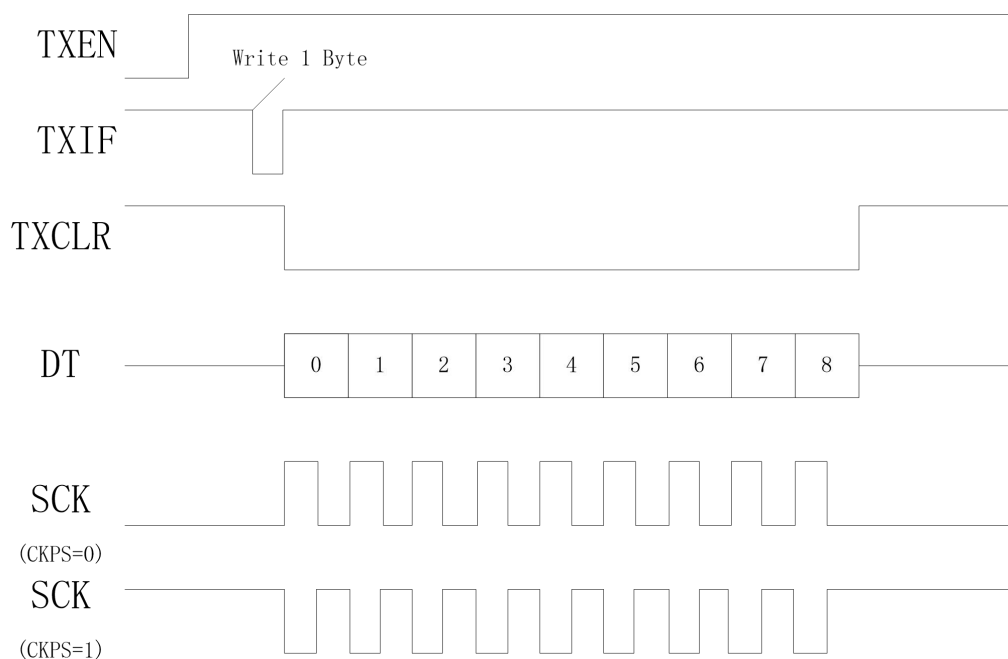
当 $TXEN=1$, $SYNC=1$ 时, 使能同步发送功能。 $CKPS$ 选择发送时钟极性, $TXIF$ 中断标志为 1 说明 $TXREG$ 发送寄存器为空, $TMCLR=1$ 说明发送移位寄存器为空, 发送器处于空闲状态。

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。

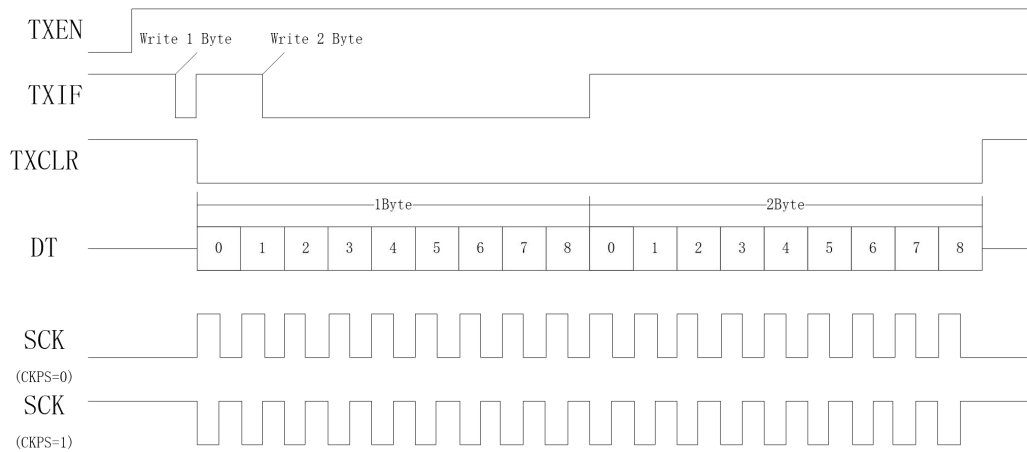


单字节发送:





多字节发送:



参考操作步骤 SLAVE=0:

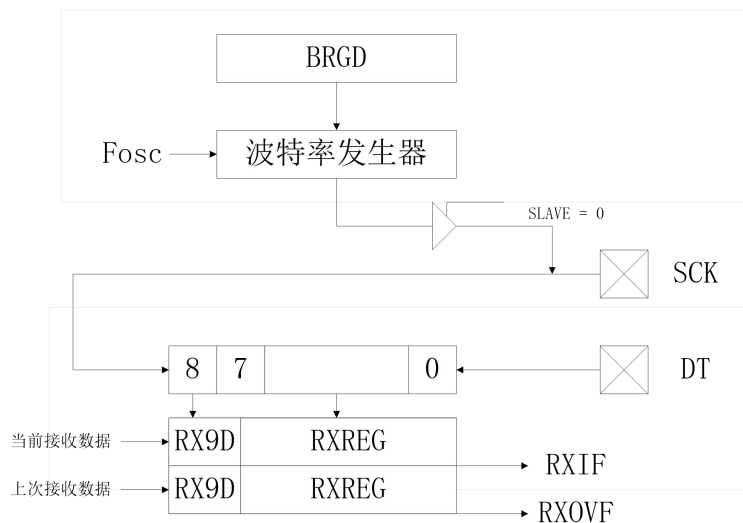
- STEP1: 设置波特率 SSPD=X, BRGD=X, SYNC=1
- STEP2: 设置 TXEN=1, 设置数据模式 TX9=X
- STEP3: 写入数据高位 TXD9
- STEP4: 写入 TXREG, 启动发送
- STEP5: 当 TXIF=1 时, 写入 TXREG 发送下一个字节
- STEP6: 重复 STEP5, 直到该帧数据发送完成

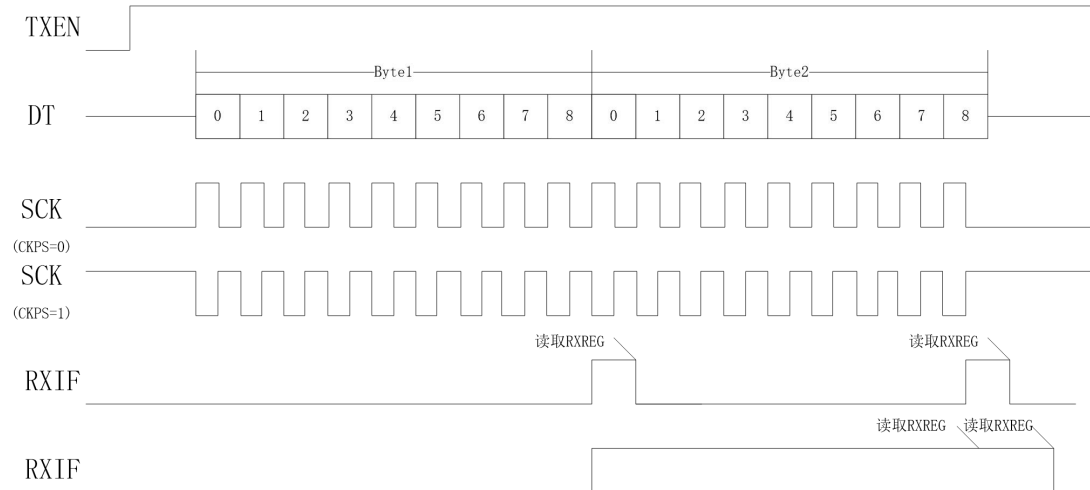
参考操作步骤 SLAVE=1:

- 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, 开始启动同步接收。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



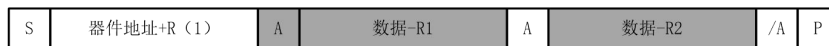
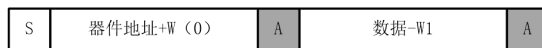
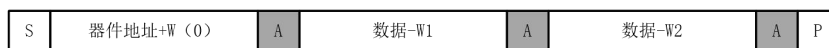
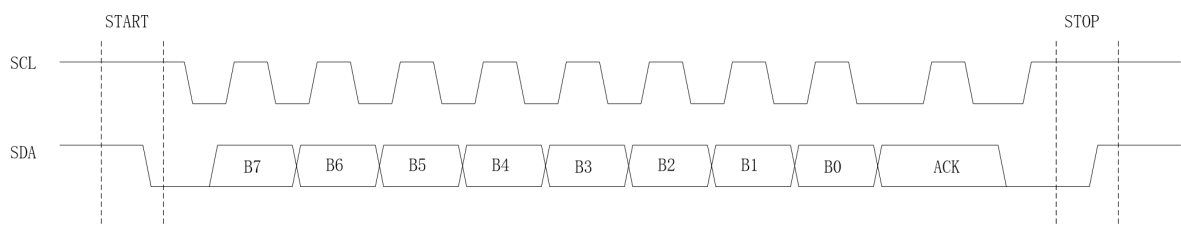
11 串行通讯口（I2C）

11.1 概述

JZ8M2626 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

0D6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CCON	I2CEN	R_W	D_A	BF	NACK	-	IICM	IICPS
读/写	R/W	R	R	R	R	-	R/W	R/W
复位后	0	1	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 1 **IICM**: IIC 总线模式

1 = 从机在未完成接收(未读 I2CBUF)时 HOLD 总线(拉低 I2C_SCL)

0 = 从机在未完成接收不 HOLD 总线(高速 Master 可能会造成通讯紊乱)

Bit 0 **IICPINS**: I2C总线脚位选择

1 = 选择B组总线(I2C_SCLB,I2C_SDAB)

0 = 选择A组总选(I2C_SCL_A,I2C_SDAA)

11.4 I2ADDR I2C 控制寄存器

0D5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CADR	I2CA7	I2CA6	I2CA5	I2CA4	I2CA3	I2CA2	I2CA1	-
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
复位后	0	0	0	0	0	0	0	0

Bit[7:0] **I2CA[2:0]**: 从机地址

地址(I2C 写入): I2CADR

地址(I2C 读取): I2CADR+1

11.4 I2CBUF 数据寄存器

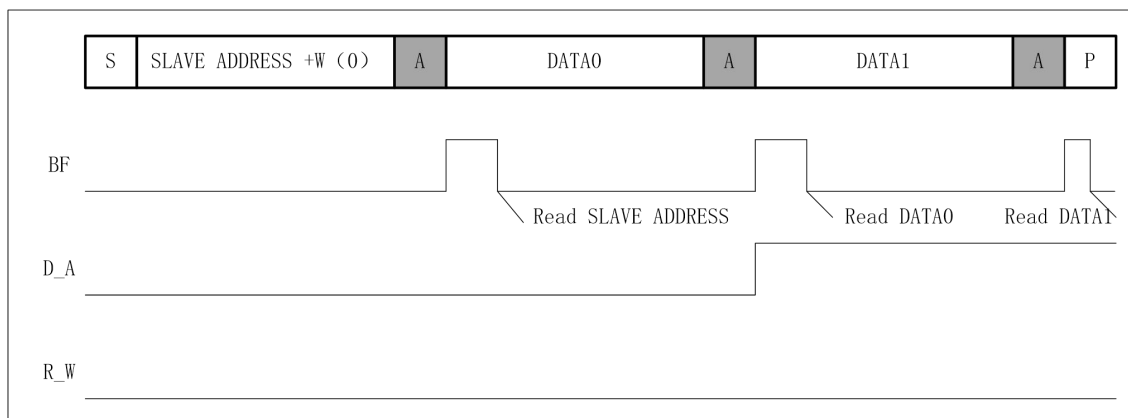
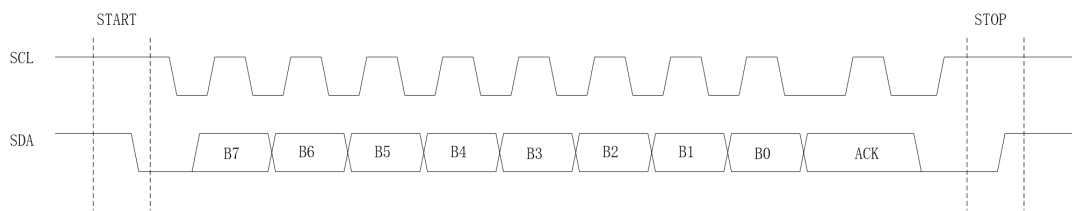
0D7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CBUF	I2CBUF7	I2CBUF6	I2CBUF5	I2CBUF4	I2CBUF3	I2CBUF2	I2CBUF1	I2CBUF0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	X	x	x	x	x	x

11.5 唤醒及休眠模式下通讯

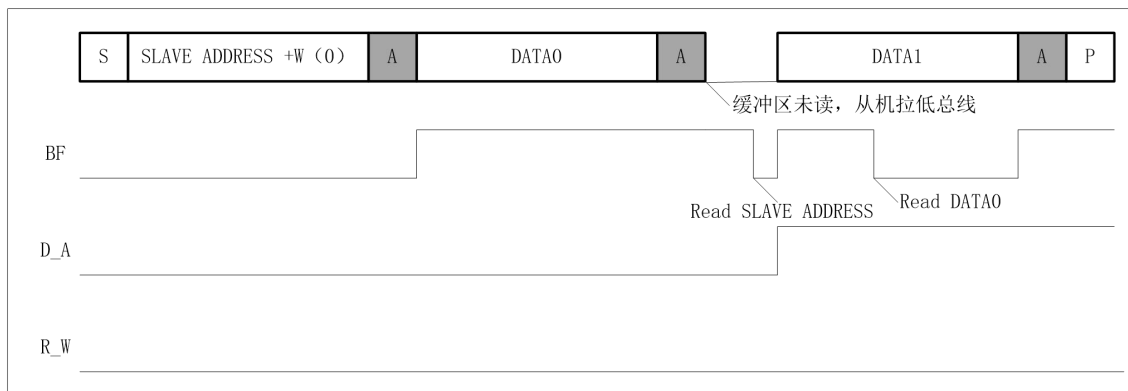
当 I2CCON.7 为 1 时, SCL、SDA 线的低电平会唤醒 CPU, 并开始通讯, 通讯期间 CPU 无法进入休眠模式



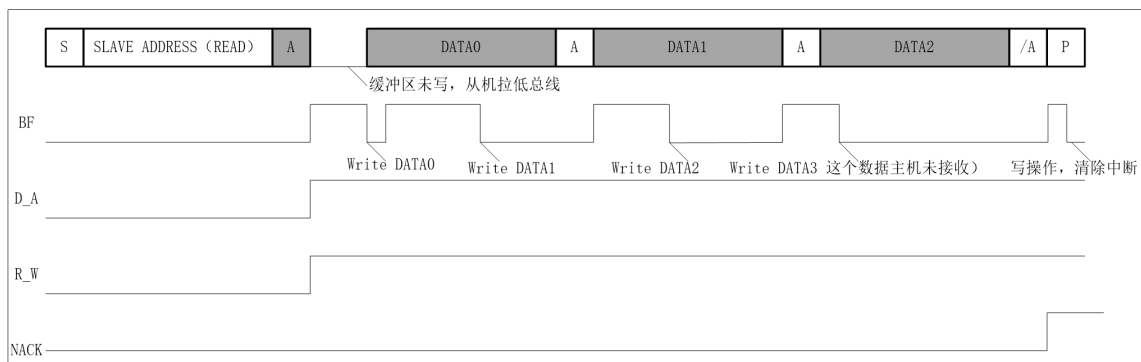
11.6 通讯波形图



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



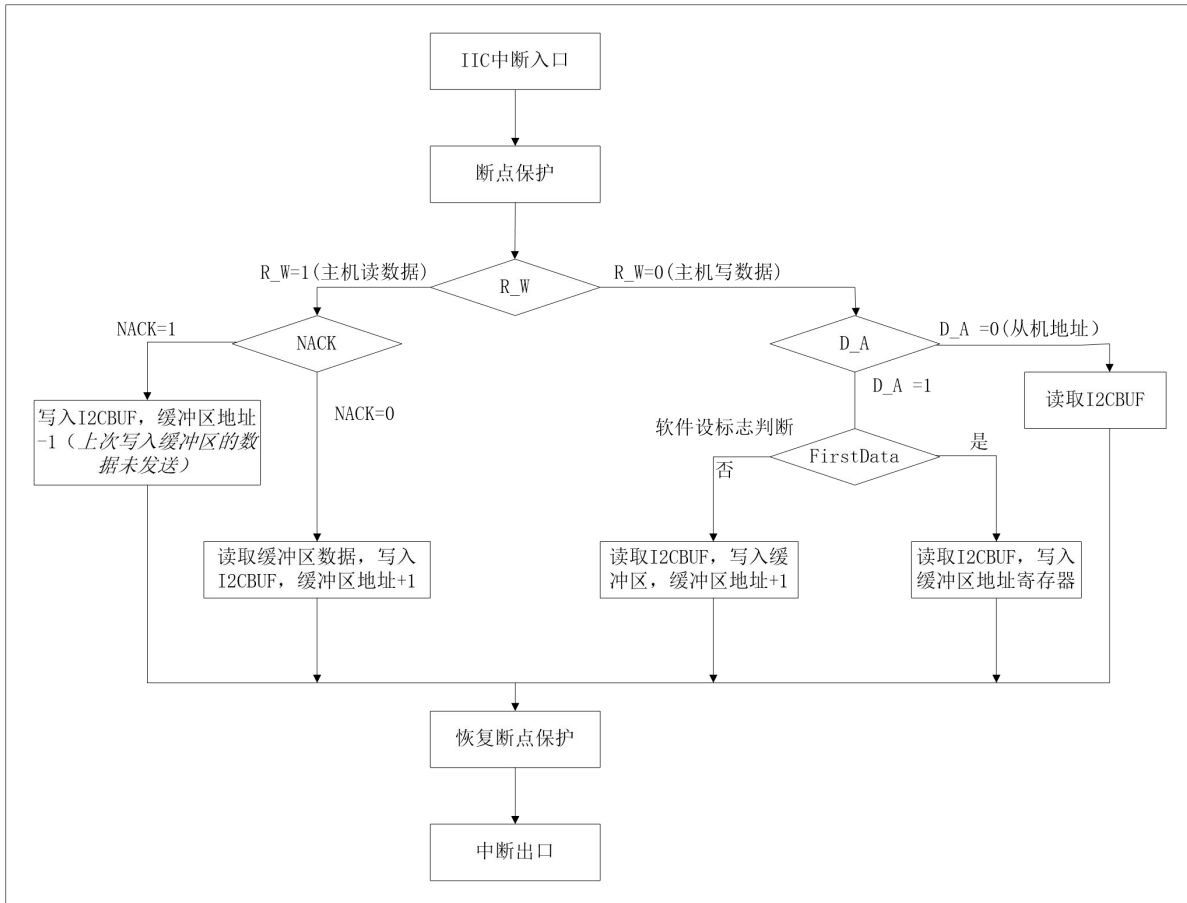
I2C Master 读数据操作



11.7 应用示例

M8E625 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= JZ8M2626
#include JZ8M2626.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'
    MOVAR    I2CCON
    MOVIA    0xA2      ;//IICSLAVE 为 0xA2(写)/0xA3 (读)
    MOVAR    I2CADR
    BSET     bI2CIE     ;//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    I2CCON,R_W ;//R_W
    GOTO     iicMasterWrite
iicMasterRead:
    JBTS0    I2CCON,NACK
    GOTO     iicMasterReadNoAck
    MOVR     SlaveBuffAddr,A
    MOVAR    FSR0
    MOVR     INDF0,A
    MOVAR    I2CBUF
    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    I2CBUF      ;//清除中断
    GOTO     isr_End
iicMasterWrite:
    JBTS0    I2CCON,D_A
    GOTO     iicGetBufAddr
```




```
iicAddr:
    MOVR    I2CBUF,A      ;//读数据,清空缓冲区
    BSET    bFirstByte
    GOTO    isr_End
iicGetBufAddr:
    JBTS1   bFirstByte
    GOTO    iicGetData
    MOVR    I2CBUF,A
    MOVAR   SlaveBuffAddr
    BCLR    bFirstByte
    GOTO    isr_End
iicGetData:
    MOVR    SlaveBuffAddr,A
    MOVAR   FSR0
    MOVR    I2CBUF,A
    MOVAR   INDF0
    INCR    SlaveBuffAddr,R
    GOTO    isr_End
;///+++++
```

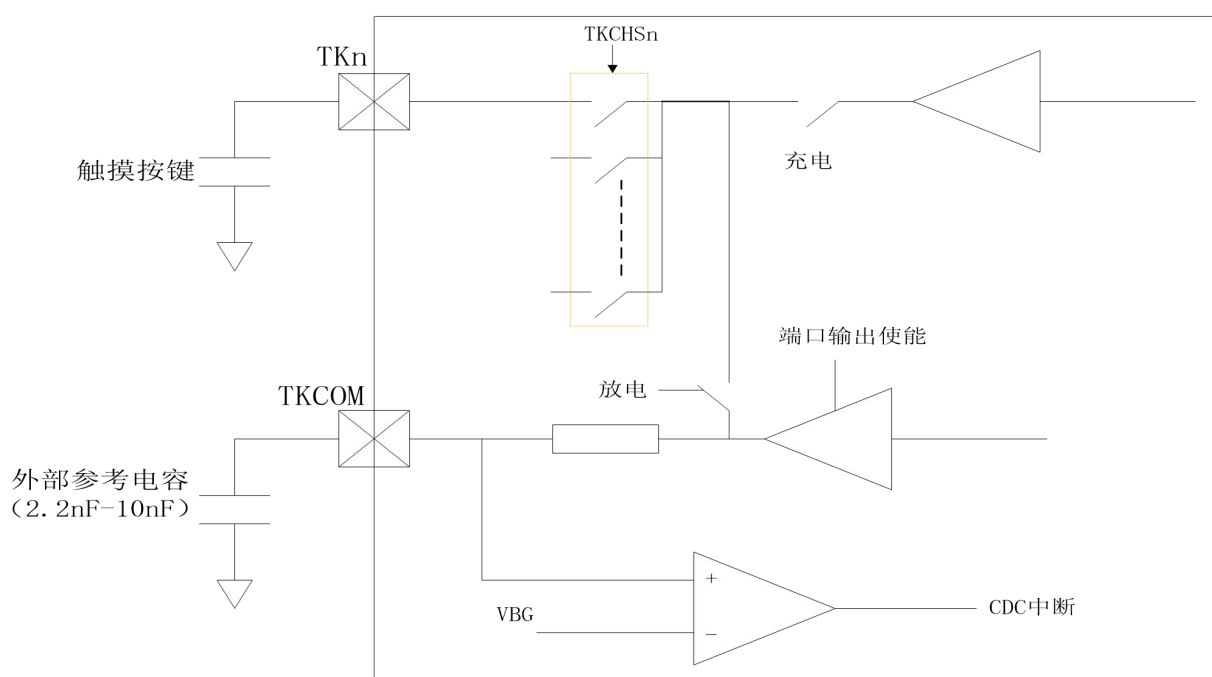


12 触摸按键（CDC）

12.1 概述

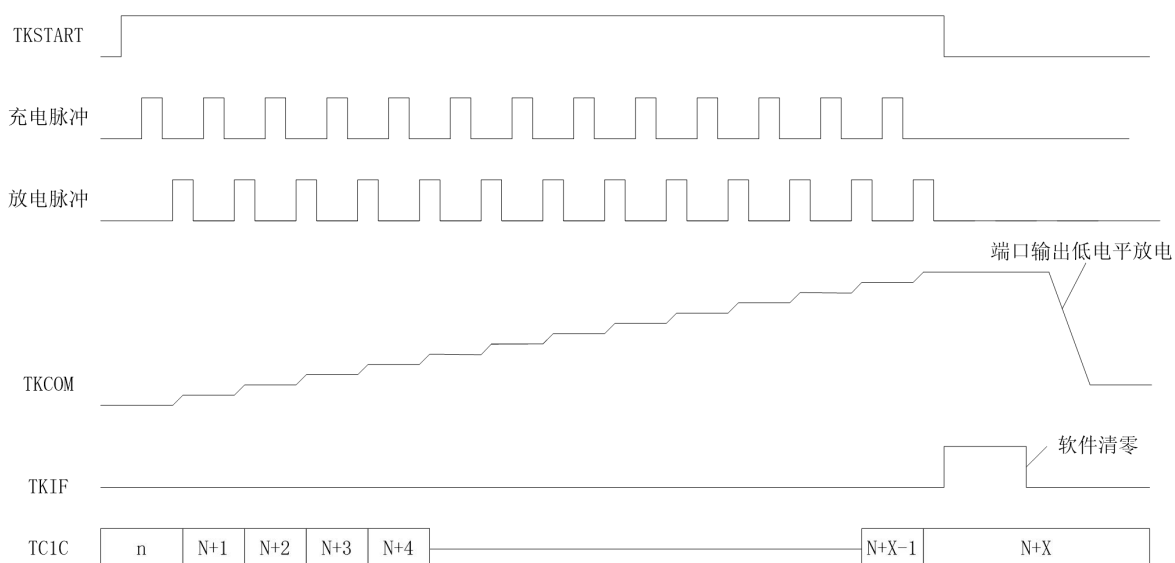
JZ8M2626 有 8 路触摸按键通道，灵敏度可通过外接电容调节，可替代机械式触摸按键，实现防水防尘，简单易用的操作接口。

12.2 原理框图



注：外部参考电容为 NPO 或 X7R 等温度系数较良好的电容。

信号波形示意图：





12.3 TKCTR0 控制寄存器 0

0B2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCTR0	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

12.4 TKCTR1 控制寄存器 1

0B3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCTR1	-	-	-	-	CAPS3	CAPS2	CAPS1	CAPS0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	-	-	-	-	0	0	0	0

Bit [3:0] **CAPS**: 内嵌电容选择

12.5 TKCHSH/L 触摸按键通道选择寄存器

0B6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCHSL	TKCHS7	TKCHS6	TKCHS5	TKCHS4	TKCHS3	TKCHS2	TKCHS1	TKCHS0
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

0B7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCHSH	-	-	-	-	-	TKCHS10	TKCHS9	TKCHS8
R/W	-	-	-	-	-	R/W	R/W	R/W
POR	-	-	-	-	-	0	0	0



Bit [7:0] TKCHSn 触摸通道连通

1 = 连通该通道

0 = 不连通该通道

注：1- 可同时连通 0-11 个通道；用于检测组合键，按键唤醒等
2- 连通的端口在 TKEN 有效时自动设置成模拟输入口

12.6 TKCNTH/L 触摸按键计数寄存器

0B4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCNTL	TKCNTL[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	X	X	X	X	X	X	X	X

0B5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCNTH	TKCNTH[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	X	X	X	X	X	X	X	X

注：1- 按键扫描过程中，不允许写操作

12.7 操作说明

- 1- 开启触摸模块电源
- 2- 等待触摸模块电源稳定 (>100uS)
- 3- 使能 CDC 模块 TKEN=1
- 4- CDC 相关通道/转换时钟设置
- 5- 设置 TKCOM 管脚输出 0，对外接电容放电 (>10uS)
- 6- 设置 TKCOM 管脚为输入模式
- 7- 清除触摸按键计数寄存器 TKxCNTH/TKxCNTL
- 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 TKCNTH/TKCNTL 的计数值，判断是否有按键发生
- 11- 重复 4-10 对不同通道进行扫描



13 模数转换器(ADC)

13.1 概述

JZ8M2626有一个12路外部通道（AIN0~AIN11）和3路内部通道（VDD_DIV，VREF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

0E0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	ADEN	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	1	0	1	0	1	0

- Bit 7 **ADEN:** 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] ~ [1011] = AIN0 ~ AIN11
 [1100] = VDD/4
 [1101] = 内建 VREF 基准电平
 [1111] = GND

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

0E1H	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

Bit6:4] **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[1:0] **VHS[1:0]: ADC 内建基准电平选择位。**

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

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

0E4H	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

0E3H	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

0E2H	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 看门狗 (WDT)

14.1 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 设置看门狗的四种溢出时间：4.5ms、18ms、72ms 或 288ms。

14.2 OPTION 配置寄存器

0BCH	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

14.3 WDTC 看门狗控制寄存器

0BEH	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。



15 电性参数

15.1 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

15.2 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
VIL1	输入低电平		有施密特			0.2VDD	
VIH1	输入高电平		有施密特	0.8VDD			
IPH	上拉电阻	5V	输入到 GND		TBD		uA
		3V	输入到 GND		TBD		
IOL1	输出灌电流	5V	输出口, Vout =VSS+0.6V		5		mA
		3V			5		
IOH1	输出拉电流	5V	输出口, Vout=VDD-0.6V	—	5	—	mA
		3V			5		mA
IOL2	输出灌电流	5V	输出口, Vout=VSS+0.6V		10		mA
		3V			10		
IOH2	输出拉电流	5V	输出口, Vout=VDD-0.6V	—	10	—	mA
		3V			10		mA
IOL3	输出灌电流	5V	输出口, Vout=VSS+0.6V		35		mA



JZ8M2626

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