

8 BIT LCD
MTP MCU
SUM08F041

数 据 手 册

☐ 产品简介

☒ 数据手册

1 产品简述

SUM08F041 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位多次可编程存储器(MTP)，256*8 位的数据存储器(RAM)，32*6 位 LCD 显示数据寄存器，46个双向 I/O 口，2 个 8/16 位 Timer 定时器/计数器，1 个带有门控功能的 16 位 Timer 定时器/计数器，3 路 RFC，1 路 UART，1 路 IIC-SLAVE，1 路基于 TC0 的 8 位分辨率 PWM，4 路基于 TC1 的 8 位分辨率 PWM，1 路比较器 (可用作 LVD)，内建 LCD/LED 输出，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
- 32*6位LCD显示数据寄存器
- 8级堆栈缓存器
- 支持查表指令

■ I/O 口

- 46个双向I/O口
- 可编程弱上拉口 IOA / IOC
- IOC 口变化中断
- 2路外部中断

■ 3 个定时器/计数器 (16Bit)

- TC0: 具有自动装载功能的定时/计数器
 - 作为1路8位分辨率PWM的时基
 - 使用PWM时，自动切换为8位模式
- TC1: 具有自动装载功能的定时/计数器
 - 作为4路8位分辨率PWM的时基
 - 使用PWM时，自动切换为8位模式
- TC2: 带有门控功能的定时/计数器

■ 系统时钟

- 内部高速RC振荡器: 16MHz
- 内部低速RC振荡器: 32KHz
- 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
- 绿色模式: 仅低速时钟工作
- 休眠模式: 高低速时钟都停止工作

■ 3 路 RFC

■ 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
- IOC口变化中断
- INT0 / INT1外部中断
- 比较器中断
- IIC通讯中断

■ 1 路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

■ 芯片间通讯接口IIC (SLAVE)

- 400Kbps

■ 比较器

- 3个正向输入源
- 3个反向输入源
- 1路内部电源电压检测VDD/2
- 1路可设内部参考电压检测

■ LCD/LED

- 最多 6x32 点
- COM 可选择 2-6 个
- COM 口驱动大电流 (目前标准的两倍)
- 电阻模式 (可设 VLCD 电压)
- 1/2Bias、1/3Bias
- LED 模式支持多级亮度设置

■ 看门狗定时器

■ 特殊功能

- 可编程代码保护
- ISP功能

■ 封装形式

- LQFP7×7-48
- QFN5×5-48
- SSOP-24

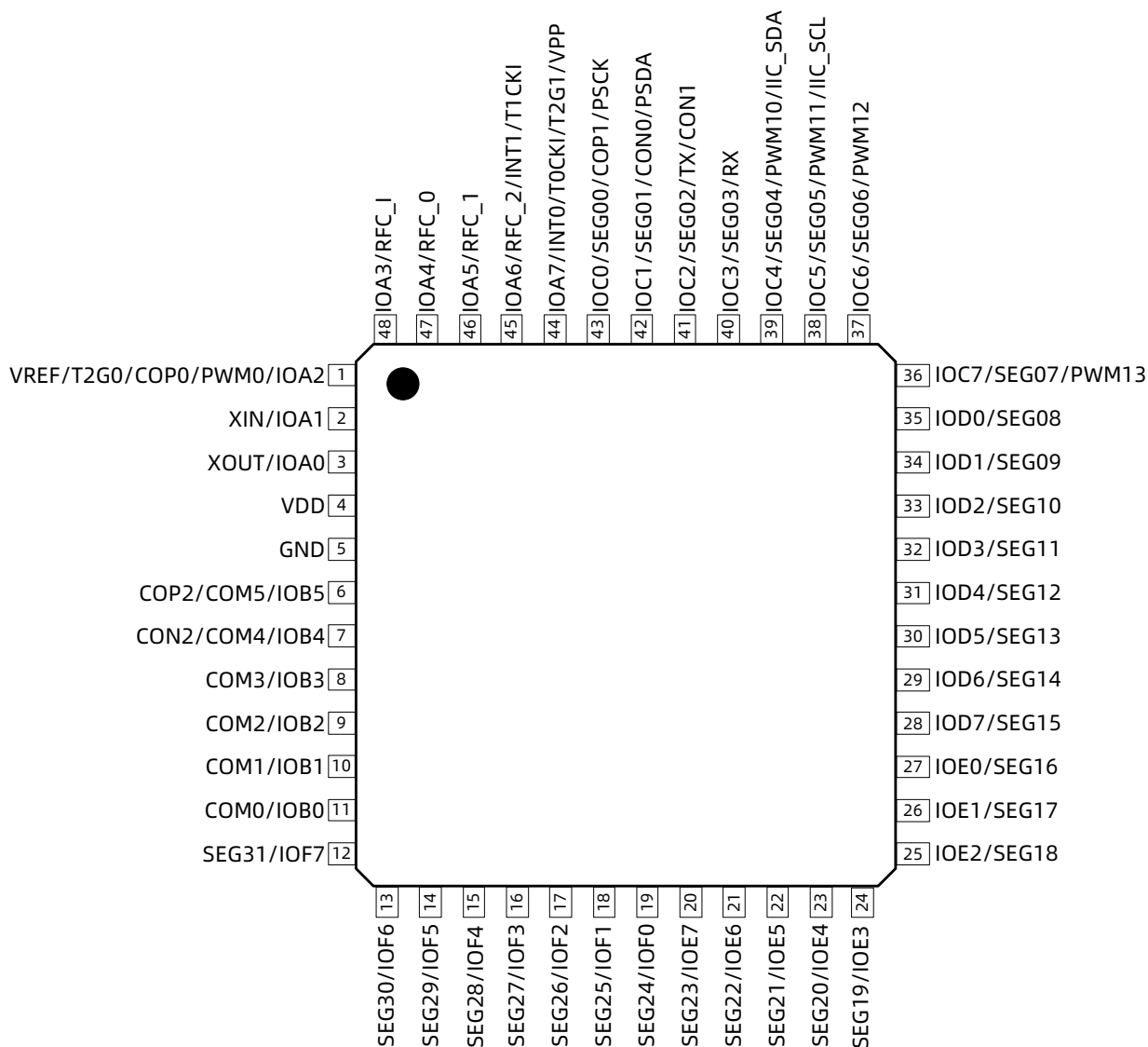
1.2 订购信息

产品名称	封装形式	订购编号	包装
SUM08F041	LQFP7×7-48	SUM08F041LF48	Tape and Reel
	QFN5×5-48	SUM08F041QNC48	Tape and Reel
	SSOP-24	SUM08F041SS24	Tape and Reel

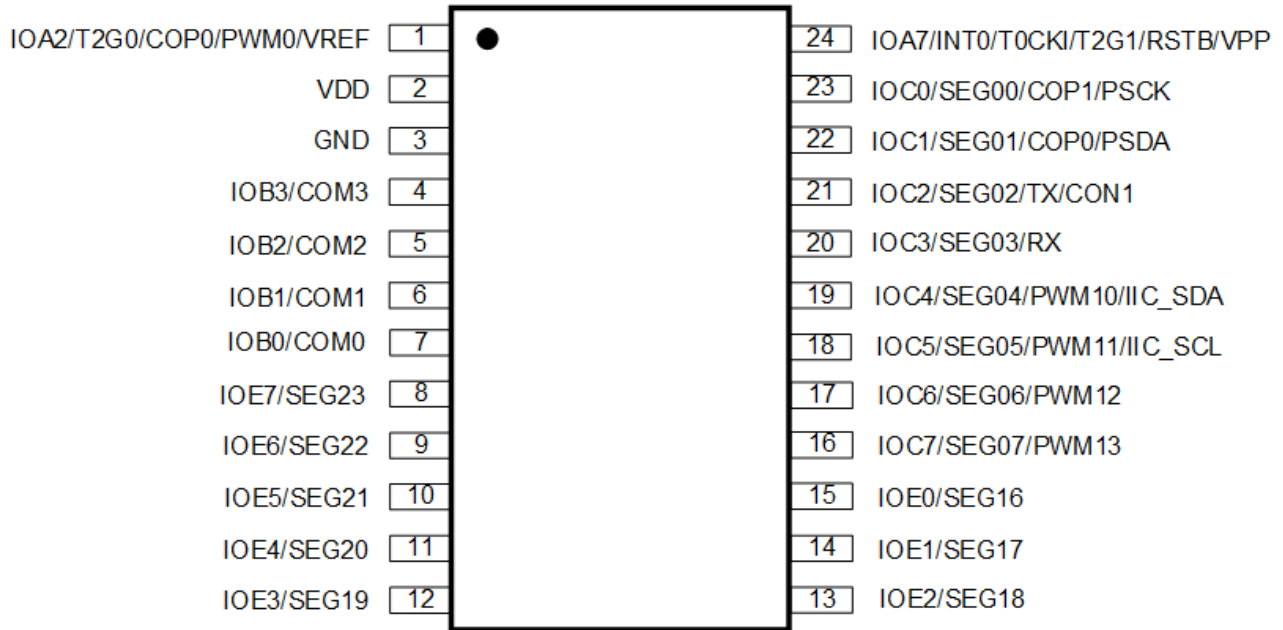
1.3 引脚图

注：芯片仿真烧录口分别是VDD、PSDA、PSCK、VPP和GND。

LQFP7×7-48 / QFN5×5-48



SSOP-24



SSOP-24

1.4 引脚描述-LQFP7×7-48 / QFN5×5-48

Pin No.	描述			Pin No.	描述		
	LCD	复用功能	IO		LCD	复用功能	IO
1		T2G0/COP0/PWM0/VREF	IOA2	48		RFC_I	IOA3
2		XIN	IOA1	47		RFC_0	IOA4
3		XOUT	IOA0	46		RFC_1	IOA5
4		VDD		45		RFC_2/INT1/T1CKI	IOA6
5		GND		44		INT0/T0CKI/VPP/T2G1/RSTB	IOA7
6	COM5	COP2	IOB5	43	SEG00	COP1/PSCK	IOC0
7	COM4	CON2	IOB4	42	SEG01	CON0/PSDA	IOC1
8	COM3		IOB3	41	SEG02	TX/CON1	IOC2
9	COM2		IOB2	40	SEG03	RX	IOC3
10	COM1		IOB1	39	SEG04	PWM10/IIC_SDA	IOC4
11	COM0		IOB0	38	SEG05	PWM11/IIC_SCL	IOC5
12	SEG31		IOF7	37	SEG06	PWM12	IOC6
13	SEG30		IOF6	36	SEG07	PWM13	IOC7
14	SEG29		IOF5	35	SEG08		IOD0
15	SEG28		IOF4	34	SEG09		IOD1
16	SEG27		IOF3	33	SEG10		IOD2
17	SEG26		IOF2	32	SEG11		IOD3
18	SEG25		IOF1	31	SEG12		IOD4
19	SEG24		IOF0	30	SEG13		IOD5
20	SEG23		IOE7	29	SEG14		IOD6
21	SEG22		IOE6	28	SEG15		IOD7
22	SEG21		IOE5	27	SEG16		IOE0
23	SEG20		IOE4	26	SEG17		IOE1
24	SEG19		IOE3	25	SEG18		IOE2

SSOP-24

Pin No.	描述			Pin No.	描述		
	LCD	复用功能	IO		LCD	复用功能	IO
1		T2G0/COP0/PWM0/VREF	IOA2	13	SEG18		IOE2
2		VDD		14	SEG17		IOE1
3		GND		15	SEG16		IOE0
4	COM3		IOB3	16	SEG07	PWM13	IOC7
5	COM2		IOB2	17	SEG06	PWM12	IOC6
6	COM1		IOB1	18	SEG05	PWM11/IIC_SCL	IOC5
7	COM0		IOB0	19	SEG04	PWM10/IIC_SDA	IOC4
8	SEG23		IOE7	20	SEG03	RX	IOC3
9	SEG22		IOE6	21	SEG02	TX/CON1	IOC2
10	SEG21		IOE5	22	SEG01	COP0/PSDA	IOC1
11	SEG20		IOE4	23	SEG00	COP1/PSCK	IOC0
12	SEG19		IOE3	24		INT0/T0CKI/T2G1/RSTB/VPP	IOA7

名称	类型	说明
VDD, GND	P	电源输入端
IOA[0] XOUT	I/O A	输入/输出 IO, SMT, 上拉电阻 外部晶体振荡器接口
IOA[1] XIN	I/O A	输入/输出 IO, SMT, 上拉电阻 外部晶体振荡器接口
IOA[2] T2G0 COP0 PWM0	I/O I A A	输入/输出 IO, SMT, 上拉电阻 TC2 门控信号输入信号 0 模拟比较器 P 端输入信号 0 PWM0 输出
IOA[3] RFC_I	I/O A	输入/输出 IO, SMT, 上拉电阻 RFC INPUT
IOA[4] RFC_0	I/O A	输入/输出 IO, SMT, 上拉电阻 RFC 通道 0
IOA[5] RFC_1	I/O A	输入/输出 IO, SMT, 上拉电阻 RFC 通道 1
IOA[6] RFC_2 INT1 T1CKI	I/O A I I	输入/输出 IO, SMT, 上拉电阻 RFC 通道 2 外部中断 1 TC1 外部时钟输入信号
IOA[7] VPP RSTB INT0 T0CKI T2G1	I/O A I I I I	输入/输出 IO, SMT, 上拉电阻 编程高压输入 外部复位输入 外部中断 0 TC0 外部时钟输入信号 TC2 门控信号输入信号 1
IOB[0] COM[0]	I/O O	输入/输出 IO, SMT LCD COM
IOB[1] COM[1]	I/O O	输入/输出 IO, SMT LCD COM
IOB[2] COM[2]	I/O O	输入/输出 IO, SMT LCD COM
IOB[3] COM[3]	I/O O	输入/输出 IO, SMT LCD COM

名称	类型	说明
IOB[4]	I/O	输入/输出 IO, SMT
COM[4]	O	LCD COM
CON2	A	模拟比较器 N 端输入信号 2
IOB[5]	I/O	输入/输出 IO, SMT
COM[5]	O	LCD COM
COP2	A	模拟比较器 P 端输入信号 2
IOC[0]	I/O	输入/输出 IO, SMT, 上拉电阻, 变化中断唤醒
COP1	A	模拟比较器 P 端输入信号 1
PSCK	A	编程/仿真串行时钟输入
SEG[00]	O	LCD SEGMENT 口输出
IOC[1]	I/O	输入/输出 IO, SMT, 上拉电阻, 变化中断唤醒
CN0	A	模拟比较器 N 端输入信号 0
PSDA	A	编程/仿真串行数据
SEG[01]	O	LCD SEGMENT 口输出
IOC[2]	I/O	输入/输出 IO, SMT, 上拉电阻, 变化中断唤醒
CON1	A	模拟比较器 N 端输入信号 1
TX	O	通用异步串行通讯发送口
SEG[02]	O	LCD SEGMENT 口输出
IOC[3]	I/O	输入/输出 IO, SMT, 上拉电阻, 变化中断唤醒
RX	O	通用异步串行通讯接收口
SEG[03]	O	LCD SEGMENT 口输出
IOC[4]	I/O	输入/输出 IO, SMT, 上拉电阻, 变化中断唤醒
I2C_SDA	I/O	I2C 通讯数据口
PWN10	O	PWM10 输出
SEG[04]	O	LCD SEGMENT 口输出
IOC[5]	I/O	输入/输出 IO, SMT, 上拉电阻, 变化中断唤醒
I2C_SCL	I/O	I2C 通讯时钟口
PWM11	O	PWM11 输出
SEG[05]	O	LCD SEGMENT 口输出
IOC[6]	I/O	输入/输出 IO, SMT, 上拉电阻, 变化中断唤醒
PWM12	O	PWM12 输出
SEG[06]	O	LCD SEGMENT 口输出
IOC[7]	I/O	输入/输出 IO, SMT, 上拉电阻, 变化中断唤醒
PWM13	O	PWM13 输出
SEG[07]	O	LCD SEGMENT 口输出
IOD[0]	I/O	输入/输出 IO, SMT
SEG[08]	O	LCD SEGMENT 口输出
IOD[1]	I/O	输入/输出 IO, SM
SEG[09]	O	LCD SEGMENT 口输出
IOD[2]	I/O	输入/输出 IO, SMT
SEG[10]	O	LCD SEGMENT 口输出
IOD[3]	I/O	输入/输出 IO, SMT
SEG[11]	O	LCD SEGMENT 口输出
IOD[4]	I/O	输入/输出 IO, SMT
SEG[12]	O	LCD SEGMENT 口输出

名称	类型	说明
IOD[5] SEG[13]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOD[6] SEG[14]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOD[7] SEG[15]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOE[0] SEG[16]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOE[1] SEG[17]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOE[2] SEG[18]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOE[3] SEG[19]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOE[4] SEG[20]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOE[5] SEG[21]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOE[6] SEG[22]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOE[7] SEG[23]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOF[0] SEG[24]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOF[1] SEG[25]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOF[2] SEG[26]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOF[3] SEG[27]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOF[4] SEG[28]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOF[5] SEG[29]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOF[6] SEG[30]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出
IOF[7] SEG[31]	I/O O	输入/输出 IO, SMT LCD SEGMENT 口输出

注：（1）IOA7 为外部复位脚时，低电平复位，作普通I/O 使用并输出高时为弱上拉驱动，驱动能力很弱，不建议用于直接驱动外围电路。接烧录器板载烧录时需要把外围电路断开，防止烧录时的高压把外围电路击穿。

（2）仿真模式下需设置VPP口为输入口，防止VPP口的输出和高压相冲突

（3）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 ~ 0x0FFF	用户区
0x8000 ~ 0x801F	厂商保留区

2.1.1 复位向量（0000H）

SUM08F041 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_Program	;//跳转至用户程序开始
...		
Main_Program:		;//用户程序开始
...		
Main:		
...		
GOTO	Main	;//用户主程序循环

2.1.2 中断向量（0008H）

SUM08F041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     Main_Program      ;//跳转到程序开始
    ORG      0008H
    GOTO     Interrupt         ;//发生中断后,跳转到中断子程序
Main_Program:
    ...
Main
    ...
    GOTO     Main              ;//主程序循环

Interrupt:
    PUSH                                ;//压栈、保存A、STATUS
    ...
    POP                                ;//出栈、恢复 A、STATUS
    RETIE

    END
    
```

2.1.3 查表

使用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和FSR0组成11位程序区数据寻址指针。

例 1：查找 ROM 地址为“DTAB” 的值

```

    MOVIA    0                  ;//要查的数据在表中的位置
    ADDIA    LOW(DTAB)          ;//获取数据表地址低位
    MOVAR    FSR0               ;//设置数据表低位指针
    MOVIA    0                  ;//要查的数据在表中的位置
    ADCIA    HIGH(DTAB)         ;//获取数据表地址高位
    MOVAR    FSR1               ;//设置数据表高位指针
                                ;//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
    
```

使用加 PCL 地址来跳转，通过 GOTO 指令可以跳转不同的程序标号。

例 2:+PCL GOTO 表

ADDRESS	EQU	0x00	;//查表地址
MOVR	ADDRESS,A		;//获取表格地址
ADDAR	PCL,R		
GOTO	TAB1		;//PCL +0 处理程序
GOTO	TAB2		;//PCL +1 处理程序
GOTO	TAB3		;//PCL +2 处理程序
TAB1:			
	处理程序		
TAB2:			
	处理程序		
TAB3:			
	处理程序		

使用加PCL地址来跳转，通过RETIA指令可以读取数据表。

例 3:+PCL RETIA 表

ADDRESS	EQU	0x00	;//查表地址
MOVR	ADDRESS,A		;//获取地址
ADDAR	PCL,R		;//地址指针加 1
RETIA	0		;//PCL +0
RETIA	1		;//PCL +1
RETIA	2		;//PCL +2
...			

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
0x000 ~ 0x0FF	YES	NO	YES	YES
0x100 ~ 0x1FF	NO	YES		

2.2.2 数据存储器寻址模式

☆ 直接寻址模式

地址 来自指令低9位

如：MOVAR 0x001； A 中的值传送给地址为 0x001 的 RAM 中

☆ 间接寻址模式 0

地址 0 FSR0

如：MOVAR INDF0； A 中的值传送给 FSR0 指向的 RAM 中

☆ 间接寻址模式 1

地址 1 FSR1

如：MOVAR INDF1； A 中的值传送给 FSR1 指向的 RAM 中

☆ 间接寻址模式 2

地址 FSR1 FSR0

如：MOVAR INDF2； A 中的值传送给{FSR1:FSR0}指向的 RAM 中

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 ~ 0x0FF	GPR (256x8)							
0x100 ~ 0x16F	RESERVE							
0x170	LCDDS00	LCDDS01	LCDDS02	LCDDS03	LCDDS04	LCDDS05	LCDDS06	LCDDS07
0x178	LCDDS08	LCDDS09	LCDDS10	LCDDS11	LCDDS12	LCDDS13	LCDDS14	LCDDS15
0x180	LCDDS16	LCDDS17	LCDDS18	LCDDS19	LCDDS20	LCDDS21	LCDDS22	LCDDS23
0x188	LCDDS24	LCDDS25	LCDDS26	LCDDS27	LCDDS28	LCDDS29	LCDDS30	LCDDS31
0x190	RESERVE							
0x198								
0x1A0								
0x1A8	LCDCR0	LCDCR1	LCDCR2	LCDCR3	SEGIO5F	SEGIO5E	SEGIO5D	SEGIO5C
0x1B0	INDF0	FSR0	TXCR	TXREG	RXCR	RXREG	BRGDH	BRGDL
0x1B8	INDF1	FSR1	PCL	STATUS	OPTION	OSCM	WDTC	RFCCR
0x1C0	INDF2	HBUF	INTCR2	INTF2	INTCR0	INTF0	INTCR1	INTF1
0x1C8	IOA	OEA	IOB	OEB	IOC	OEC	IOD	OED
0x1D0	IOE	OEE	IOF	OEF	-	-	-	-
0x1D8	PWM0CR	PWM0D	PWM1CR	PWM10D	PWM11D	PWM12D	T2CR	T2GCR
0x1E0	PUA	ANSCMP	PUC	PWM13D	IOCICR	IODSCR0	IODSCR1	IOICR
0x1E8	T0CR	TC0CL	TC0CH	T1CR	TC1CL	TC1CH	TC2CL	TC2CH
0x1F0	I2CADDR	I2CCON	I2CBUF	CMPC0	CMPC1	CMPC2	-	-
0x1F8	-	-	-	-	-	-	-	-

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 INDF2 间接寻址寄存器 2

访问INDF2寄存器时，实现间接寻址模式2，访问到的是{FSR1:FSR0}寄存器所指向的寄存器内容，间接寻址模式2可寻址通用寄存器区0x0000~0x01FF空间。

2.2.7 FSR0 间接寻址指针 0

使用间接寻址模式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。

2.2.8 FSR1 间接寻址指针 1

使用间接寻址模式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。

2.2.9 HBUF 查表数据高 8 位

使用RDT指令读取程序区数据时，读到的16位数据的高8位放在HBUF中。

2.2.10 PCL 程序计数器指针低位

	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	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（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。

2.2.11 STATUS 状态寄存器

	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

0 = 算术/逻辑运算的结果非零

1 = 算术/逻辑运算的结果为零

Bit 1 **DC:** 辅助进位标志

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

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

Bit 0 **C:** 进位标志

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”

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”

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)

SUM08F041 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

SUM08F041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（4MHz-20MHz）或内置的16MHz RC振荡电路（HIRC 16MHz）提供，低速时钟由外部低速晶体振荡器（32768Hz）或内置的低速RC振荡电路（LIRC 64KHz）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

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCM	STBH	STBL	-	STOP	CLKM	STPH	-	STPL
读/写	R/W	R/W	-	R/W	R/W	R/W	-	R/W
复位后	X	X	-	0	X	1	-	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

0 = CPU正常工作

1 = CPU停止工作，所有中断或看门狗溢出唤醒

Bit 3 **CLKM**：系统时钟状态标志位

0 = CPU运行于高频时钟

1 = CPU运行于低频时钟

Bit 2 **STPH**：高频振荡器控制

0 = 休眠状态或低速模式下高速振荡器仍然工作

1 = 休眠状态或低速模式下关闭高频振荡器

Bit 0 **STPL**：低频振荡器控制

0 = 休眠状态下低频振荡器仍然工作

1 = 休眠状态下低频振荡器停止工作

注：CLKM 的初始状态由配置字决定。

4.3 系统时钟的工作模式

普通模式：普通模式有两种分别是： 1.高频时钟工作，低频时钟工作，不进 STOP
(电流特性参考电性参数表 I_{DD1})
2.高频时钟停止，低频时钟工作，不进 STOP

绿色模式：绿色模式有两种分别是： 1.高频时钟工作，低频时钟工作，进 STOP
(电流特性参考电性参数表 I_{SP1})
2.高频时钟停止，低频时钟工作，进 STOP
(电流特性参考电性参数表 I_{SP2})

绿色模式可以由所有中断或 WDT 唤醒。

休眠模式：休眠模式只有一种是：高频时钟停止，低频时钟停止，进 STOP
(电流特性参考电性参数表 I_{SP3})

休眠模式可以由外部中断、IO 变化中断或 WDT 唤醒。

注：(1) 省电建议，程序运行时跑高频，快速跑完程序然后进休眠，此时休眠下需设置高频时钟停止工作。
(2) 各工作模式的工作电流参考电性参数表。
(3) 绿色和休眠模式下，如果总中断不开启，所有中断唤醒能唤醒芯片但是不会进中断。

4.4 IRCCAL 寄存器

	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:

```

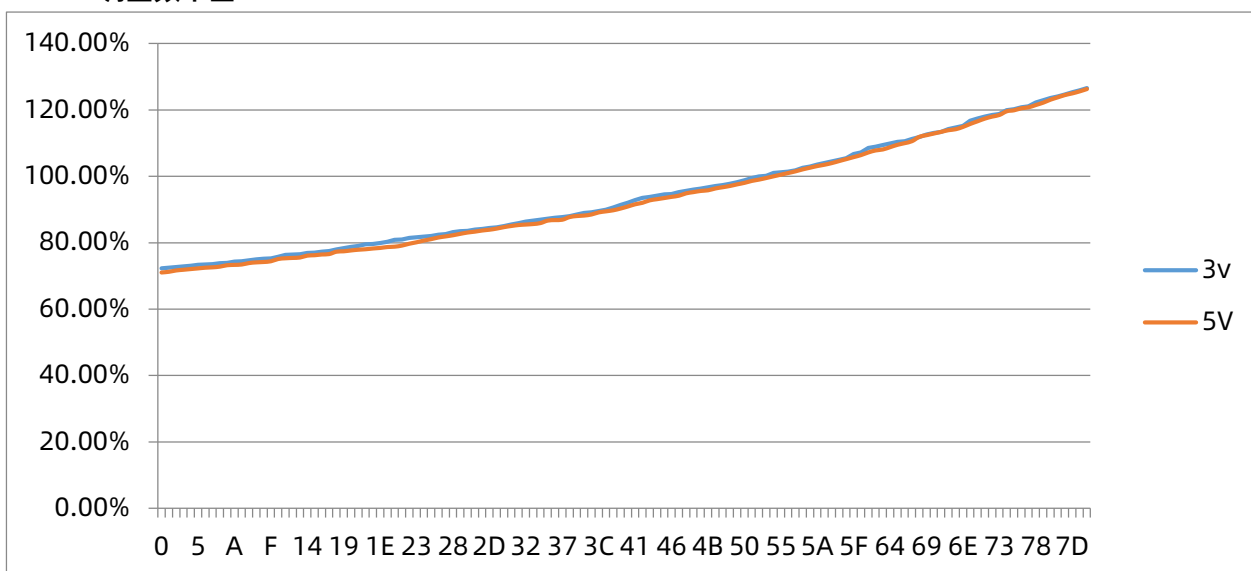
MOVIA    0x55
MOVAR    0x1F9          ;//1F9H地址写入055H
MOVIA    0xAA
MOVAR    0x1F9          ;//1F9H地址写入0AAH
MOVIA    0x00
MOVAR    IRCCAL         ;// IRCCAL写入0x00
...

```

;//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

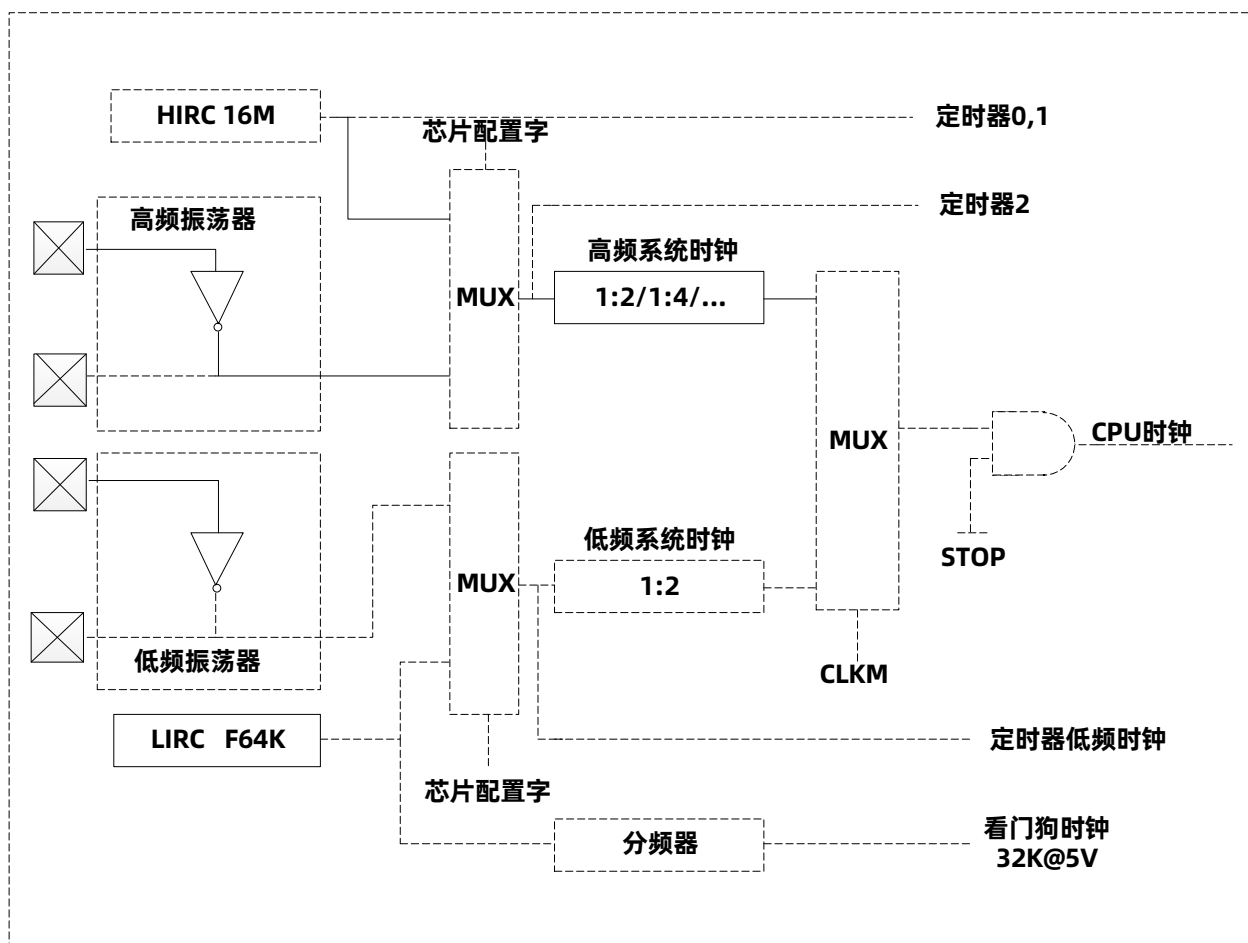
注：IRCCAL 寄存器地址为 1FCH。

IRC 调整频率图



注：具体值不做设计保证。

4.5 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
TC0/TC1	TCxEN	若选择高速时钟, 需 STPH=0	高速时钟源&STPH=0 低速时钟源&STPL=0

4.6 系统时钟高低频切换

高频振荡器稳定计数器：7 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=1：7 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

SUM08F041有多路中断源: TC0/TC1/TC2, TC2门控, IOC变化中断, 外部INT0/INT1中断, UART发送/接收中断, 比较器中断, IIC通讯中断。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

	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

0 = 屏蔽所有中断 (响应中断后自动清零)

1 = 总中断使能 (RETIE指令会将该位置1)

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 IOC 变化中断使能寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOCICR	IOCICR7	IOCICR6	IOCICR5	IOCICR4	IOCICR3	IOCICR2	IOCICR1	IOCICR0
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] **IOCICRn**: IOC端口变化中断使能 (n=0-7)

0 = 屏蔽IOC.n口电平变化中断

1 = 使能IOC.n口电平变化中断

5.4 INTCR0 中断控制寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCR0	-	-	-	-	TC2GIE	TC2IE	TC1IE	TC0IE
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位后	-	-	-	-	0	0	0	0

Bit 3 **TC2GIE:**
0 = 屏蔽TC2门控中断
1 = 使能TC2门控中断

Bit 2 **TC2IE:**
0 = 屏蔽TC2溢出中断
1 = 使能TC2溢出中断

Bit 1 **TC1IE:**
0 = 屏蔽TC1溢出中断
1 = 使能TC1溢出中断

Bit 0 **TC0IE:**
0 = 屏蔽TC0溢出中断
1 = 使能TC0溢出中断

5.5 INTF0 中断标志寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF0	-	-	-	-	TC2GIF	TC2IF	TC1IF	TC0IF
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位后	-	-	-	-	0	0	0	0

Bit 3 **TC2GIF:**
0 = 未产生TC2门控中断
1 = 产生TC2门控中断

Bit 2 **TC2IF:**
0 = 未产生TC2溢出中断
1 = 产生TC2溢出中断

Bit 1 **TC1IF:**
0 = 未产生TC1溢出中断
1 = 产生TC1溢出中断

Bit 0 **TC0IF:**
0 = 未产生TC0溢出中断
1 = 产生TC0溢出中断

5.6 INTCR1 中断控制寄存器 1

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

Bit 7

INT1IE:

0 = 屏蔽外部端口中断1

1 = 使能外部端口中断1

Bit 6

INT0IE:

0 = 屏蔽外部端口中断0

1 = 使能外部端口中断0

Bit 1

IOCCHIE:

0 = 屏蔽外部端口IOC变化中断

1 = 使能外部端口IOC变化中断

5.7 INTF1 中断标志寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF1	INT1IF	INT0IF	-	-	-	-	IOCCHIF	-
读/写	R/W	R/W	-	-	-	-	R/W	-
复位后	X	X	-	-	-	-	X	-

Bit 7

INT1IF:

0 = 无外部中断1产生

1 = 外部中断1产生

Bit 6

INT0IF:

0 = 无外部中断0产生

1 = 外部中断0产生

Bit 1

IOCCHIF:

0 = 对应输入端口状态未发生变化

1 = 对应输入端口状态发生变化

5.8 INTCR2 中断控制寄存器 2

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCR2	CMPIE	-	-	I2CIE	-	-	RXIE	TXIE
读/写	R/W	-	-	R/W	-	-	R/W	R/W
复位后	0	-	-	0	-	-	0	0

Bit 7

CMPIE:

0 = 屏蔽比较器中断

1 = 使能比较器中断

Bit 4

I2CIE:

0 = 屏蔽IIC通讯中断

1 = 使能IIC通讯口中断

Bit 1

RXIE:

0 = 屏蔽UART接收中断

1 = 使能UART接收中断

Bit 0

TXIE:

0 = 屏蔽UART发送中断

1 = 使能UART发送中断

5.9 INTF2 中断标志寄存器 2

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF2	CMPIF	-	-	I2CIF	-	-	RXIF	TXIF
读/写	R/W	-	-	R/W	-	-	R	R
复位后	0	-	-	0	-	-	0	0

Bit 7

CMPIF:

0 = 未发生比较器中断

1 = 发生比较器中断，需软件清零

Bit 4

I2CIF:

I2C Salve接收数据

0 = 未收到数据或数据已读

1 = 缓冲区内接收到数据（读I2CBUF清零）

I2C Slave发送数据

0 = 缓冲区已写入待发数据

1 = 缓冲区内无待发数据

Bit 1

RXIF:

0 = 未发生UART接收中断

1 = 发生UART接收中断

Bit 0

TXIF:

0 = 未发生UART发送中断

1 = 发生UART发送中断

注：所有中断标志位需软件清零。

6 端口

6.1 IOX 数据寄存器 (X = A/B/C/D/E/F)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOA	IOA7	IOA6	IOA5	IOA4	IOA3	IOA2	IOA1	IOA0
IOB	-	-	IOB5	IOB4	IOB3	IOB2	IOB1	IOB0
IOC	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0
IOD	IOD7	IOD6	IOD5	IOD4	IOD3	IOD2	IOD1	IOD0
IOE	IOE7	IOE6	IOE5	IOE4	IOE3	IOE2	IOE1	IOE0
IOF	IOF7	IOF6	IOF5	IOF4	IOF3	IOF2	IOF1	IOF0
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

注：寄存器具体地址参考<2.2.3 系统寄存器定义>

6.2 OEX 方向寄存器 (X = A/B/C/D/E/F)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OEA	OEA7	OEA6	OEA5	OEA4	OEA3	OEA2	OEA1	OEA0
OEB	-	-	OEB5	OEB4	OEB3	OEB2	OEB1	OEB0
OEC	OEC7	OEC6	OEC5	OEC4	OEC3	OEC2	OEC1	OEC0
OED	OED7	OED6	OED5	OED4	OED3	OED2	OED1	OED0
OEE	OEE7	OEE6	OEE5	OEE4	OEE3	OEE2	OEE1	OEE0
OEF	OEF7	OEF6	OEF5	OEF4	OEF3	OEF2	OEF1	OEF0
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] OEXn: X口输出使能 (n=0-7)

0 = 输入

1 = 输出

注：寄存器具体地址参考<2.2.3 系统寄存器定义>

注：IOA[7]作为输出口的注意事项

(1) 需将 PUA7 置 1 才能输出高电平。

(2) IOA[7]输出的高电平是由上拉电阻提供的，所以驱动能力弱。

(3) IOA[7]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。

6.3 PUX 上拉使能寄存器 (X = A/C)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PUA	PUA7	PUA6	PUA5	PUA4	PUA3	PUA2	PUA1	PUA0
PUC	PUC7	PUC6	PUC5	PUC4	PUC3	PUC2	PUC1	PUC0
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] **PUXn**: X口上拉使能 (n=0-7)

0 = 上拉关闭

1 = 上拉使能

注: 寄存器具体地址参考<2.2.3 系统寄存器定义>

6.4 IOCICR 变换中断使能寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOCICR	IOCICR7	IOCICR6	IOCICR5	IOCICR4	IOCICR3	IOCICR2	IOCICR1	IOCICR0
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] **IOCICRn**: IOC端口变化中断使能 (n=0-7)

0 = 关闭IOC.n口电平变化中断

1 = 使能IOC.n口电平变化中断

注: 寄存器具体地址参考<2.2.3 系统寄存器定义>

6.5 SEGIOX 端口设置 (X = C/D/E/F)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SEGIOSC	SEGIOSC7	SEGIOSC6	SEGIOSC5	SEGIOSC4	SEGIOSC3	SEGIOSC2	SEGIOSC1	SEGIOSC0
SEGIOSD	SEGIOSD7	SEGIOSD6	SEGIOSD5	SEGIOSD4	SEGIOSD3	SEGIOSD2	SEGIOSD1	SEGIOSD0
SEGIOSE	SEGIOSE7	SEGIOSE6	SEGIOSE5	SEGIOSE4	SEGIOSE3	SEGIOSE2	SEGIOSE1	SEGIOSE0
SEGIOSF	SEGIOSF7	SEGIOSF6	SEGIOSF5	SEGIOSF4	SEGIOSF3	SEGIOSF2	SEGIOSF1	SEGIOSF0
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] **SEGIOXn**: LCD的SEG口选择 (X=C/D/E/F, n=0-7)

0 = 作为IO口

1 = 作为LCD/LED的SEG口

注: 寄存器具体地址参考<2.2.3 系统寄存器定义>

6.6 端口驱动设置寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IODSCR0	IOADS	-	IOBDS	-	-	IOFDS	IOEDS	IODDS
读/写	R/W	-	R/W	-	-	R/W	R/W	R/W
复位后	0	-	0	-	-	0	0	0

Bit [7,2,1,0] **IOxDS:** IOx口驱动能力选择 (x=A、F、E、D)

IOxDS	IO 驱动能力选择
0	标准驱动 IOL1/ IOH1
1	小驱动 IOL2/ IOH2

Bit 5 **IOBDS:** IOB口驱动能力选择

IOBHDS	IO 驱动能力选择
0	标准驱动 IOL1/ IOH2
1	COM 口 N 驱动大电流 IOL3/ IOH1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IODSCR1	IOC7DS	IOC6DS	IOC5DS	IOC4DS	IOC3DS	IOC2DS	IOC1DS	IOC0DS
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] **IOCxDS:** IOCx口驱动能力选择 (x=0-7)

IOCxDS	IO 驱动能力选择
0	标准驱动 IOL1/ IOH1
1	小驱动 IOL2/ IOH2

注：IO 口的各个驱动能力参考电性参数表。

6.7 端口输入设置寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOICR			IOCHIS	IOCLIS			IOAHIS	IOALIS
读/写	-	-	R/W	R/W	-	-	R/W	R/W
复位后	-	-	0	0	-	-	0	0

Bit [5,1] **IOxHIS**: IOxH口翻转电平选择 (x=A、C)

IOxHIS	IOx [5,1] 输入翻转电平选择
0	SMT 翻转
1	低翻转点

Bit [4,0] **IOxLIS**: IOxL口翻转电平选择 (x=A、C)

IOxLIS	IOx [4,0] 输入翻转电平选择
0	SMT 翻转
1	低翻转点

注: VPP 口没有低翻转, IOB 和 IOD 翻转电压不可选低翻转点

6.8 ANSCMP 比较器模拟端口设置寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ANSCMP	-	-	ANSP2	ANSN2	ANSN1	ANSN0	ANSP0	ANSP0
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	-	0	0	0	0	0	0

Bit 5 **ANSP2**: CMP 的 P2 (B5 口) 端口类型设置

0 = 数字端口

1 = 模拟端口功能 (数字输入功能被屏蔽, 即端口状态 MCU 无法读取)

Bit 4 **ANSN2**: CMP 的 N2 (B4 口) 端口类型设置

0 = 数字端口

1 = 模拟端口功能 (数字输入功能被屏蔽, 即端口状态 MCU 无法读取)

Bit 3 **ANSN1**: CMP 的 N1 (C2 口) 端口类型设置

0 = 数字端口

1 = 模拟端口功能 (数字输入功能被屏蔽, 即端口状态 MCU 无法读取)

Bit 2 **ANSN0**: CMP 的 N0 (C1 口) 端口类型设置

0 = 数字端口

1 = 模拟端口功能 (数字输入功能被屏蔽, 即端口状态 MCU 无法读取)

Bit 1 **ANSP1**: CMP 的 P1 (C0 口) 端口类型设置

0 = 数字端口

1 = 模拟端口功能 (数字输入功能被屏蔽, 即端口状态 MCU 无法读取)

Bit 0 **ANSP0**: CMP 的 P0 (A2 口) 端口类型设置

0 = 数字端口

1 = 模拟端口功能 (数字输入功能被屏蔽, 即端口状态 MCU 无法读取)

7 定时器 0/1(TC0/1)

7.1 概述

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。

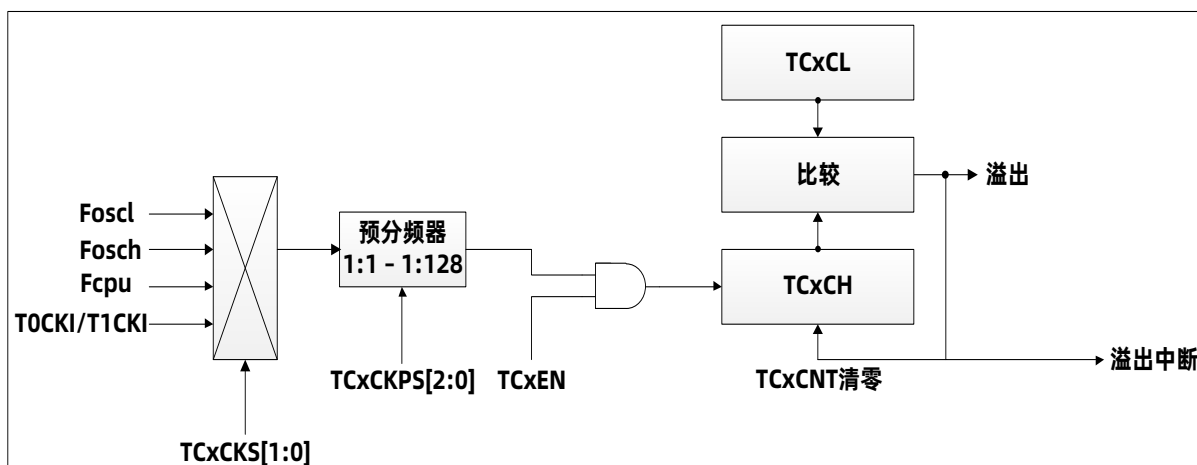
在 8 位模式下，TCxCL 作为 TCx 的周期寄存器，TCx 使能后，TCxCH 递加，当 TCxCH 与 TCxCL 数值相等时，TCx 溢出，将 TCxCH 清零重新开始计数，同时将中断标志位 TCxIF 置 1。

在 16 位模式下，[TCxCH,TCxCL]作为 16 位的计数器，TCx 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。

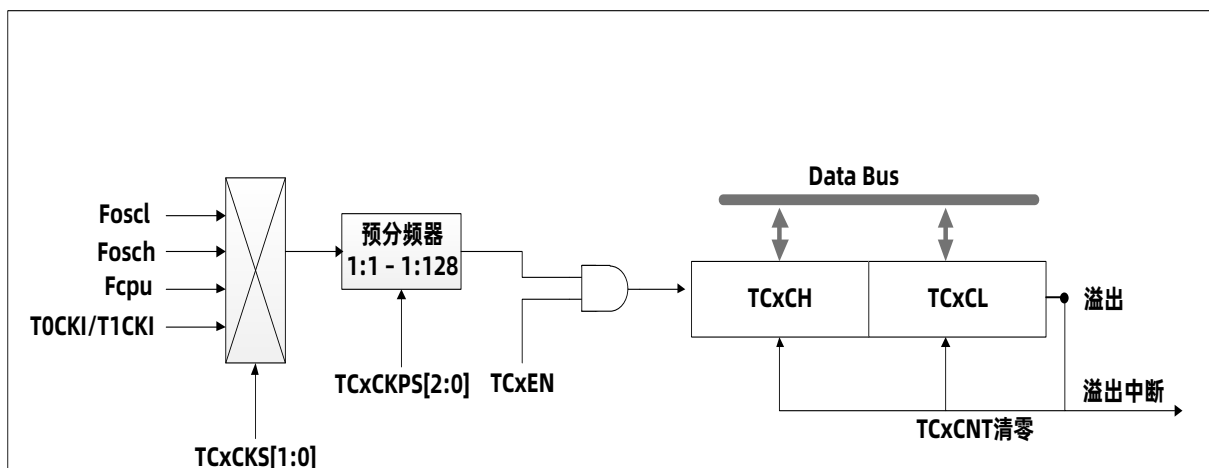
- 可选择时钟源，高频系统时钟 Fosch、低频系统时钟 Foscl、指令时钟 Fcpu 和外部时钟 T0CKI/T1CKI
- 可选择 8 位模式和 16 位模式
 - 8 位模式下，通过设置周期寄存器，可任意设置 TC0/TC1 的周期
- 预分频比多级可选，最大可选择 1:128
- 溢出中断功能
- 溢出中断唤醒功能（当输入频率选择 Foscl 或 Fosch，若所选择的时钟源振荡器一直工作，此时 TC0/TC1 在休眠状态下依然工作，溢出中断可唤醒 CPU）

TC0/TC1 两种模式框图：

8 位模式



16 位模式



7.2 TxCR 控制寄存器(x=0,1)

	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

0 = 关闭TCx

1 = 使能TCx

Bit 6 **TCxMOD:** TCx模式选择位

0 = 8位模式

1 = 16位模式

Bit [4:3] **TCxCKS[1:0]:** TCx时钟源选择

TCxCKS[1:0]	TCx 时钟源选择
00	FoscI(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(x=0,1)

	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(x=0,1)

	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

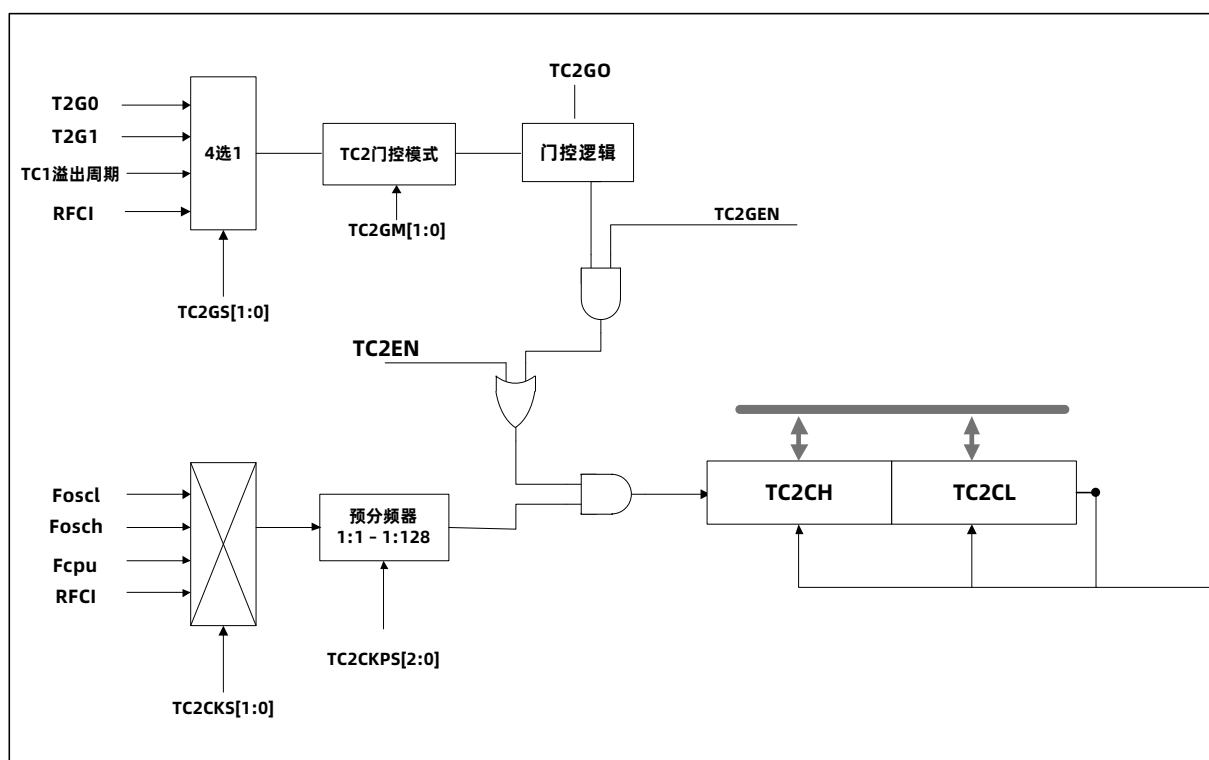
8 定时器2 (TC2)

8.1 概述

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。

在 16 位模式下，[TC2CH,TC2CL]作为 16 位的计数器，TC2 使能后，16 位计数器开始计数，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 TC2IF 置 1。

TC2 16 位模式框图



8.2 T2CR 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CR	TC2EN	TC2M	-	TC2CKS1	TC2CKS0	TC2CKPS2	TC2CKPS1	TC2CKPS0
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 **TC2EN**: TC2模块使能位

0 = 关闭TC2

1 = 使能TC2

Bit 6 **TC2M**: TC2模式选择位

0 = 未实现, 不可用

1 = 16位模式

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	Foscl(低频系统时钟)
01	Fosch(高频系统时钟)
10	Fcpu
11	RFC 时钟 RFCI

Bit [2:0] **TC2CKPS[2:0]**: TC2预分频比选择

TC2CKPS[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

	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

	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	R/W
复位后	X	X	X	X	X	X	X	X

注：（1）使能定时器后，第二个时钟上升沿开始计数。

（2）16 位模式下，写 TC2CL/TC2CH 会暂停计数，写操作完成后，需要有一个时钟下降沿后才开始正常计数。

8.5 T2GCR 门控控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2GCR	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
0 = 关闭门控功能, TC2是否计数仅受TC2EN控制
1 = 使能门控功能

Bit 6 **TC2GO**: 启动门控控制位
0 = 完成门控计数, 自动清零
1 = 启动门控

Bit [3:2] **TC2GS[1:0]**: TC2门控源选择位

TC2GS[1:0]	TC2 门控选择
00	TC1 溢出周期 (支持上升沿到上升沿模式)
01	RFC 时钟 RFCI
10	T2G0
11	T2G1

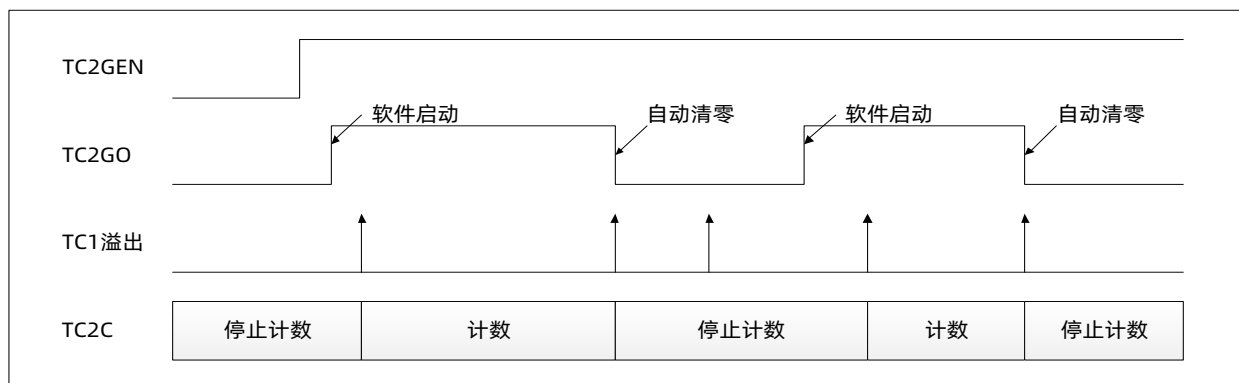
Bit [1:0] **TC2GM[1:0]**: TC2门控模式选择位

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 TC1 溢出信号作为门控源时, 若 TC1 是 8 位模式, 则 TC1CL 不能为 0

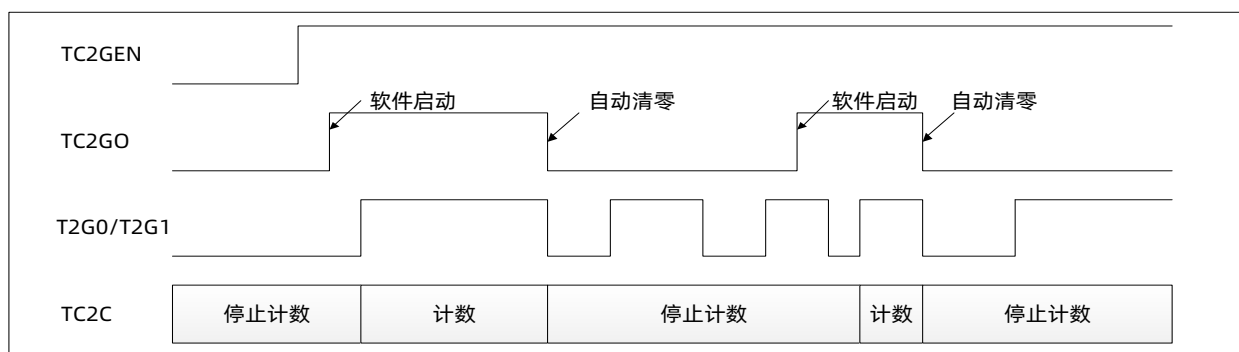
8.5.1 门控-TC1 溢出周期

当门控源选择 TC1 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 TC1 溢出开始计数，第二次 TC1 溢出停止计数，如下图：



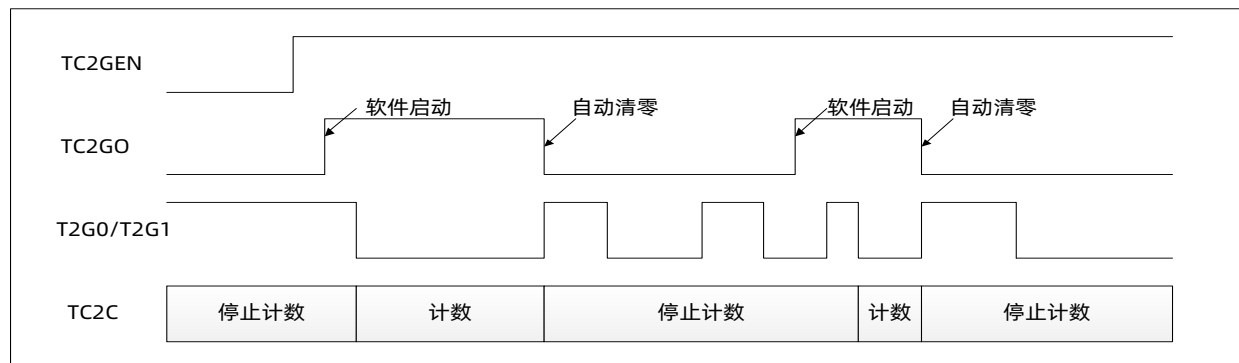
8.5.2 门控-上升沿到下降沿模式

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：



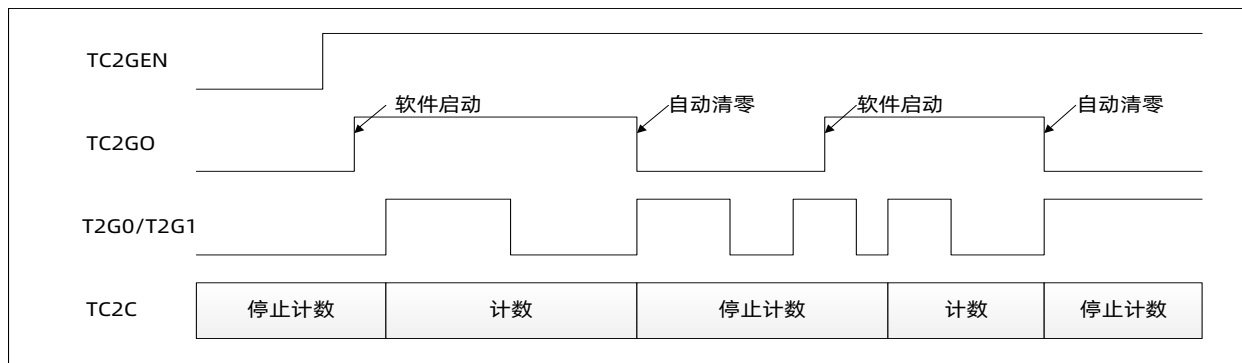
8.5.3 门控-下降沿到上升沿模式

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：



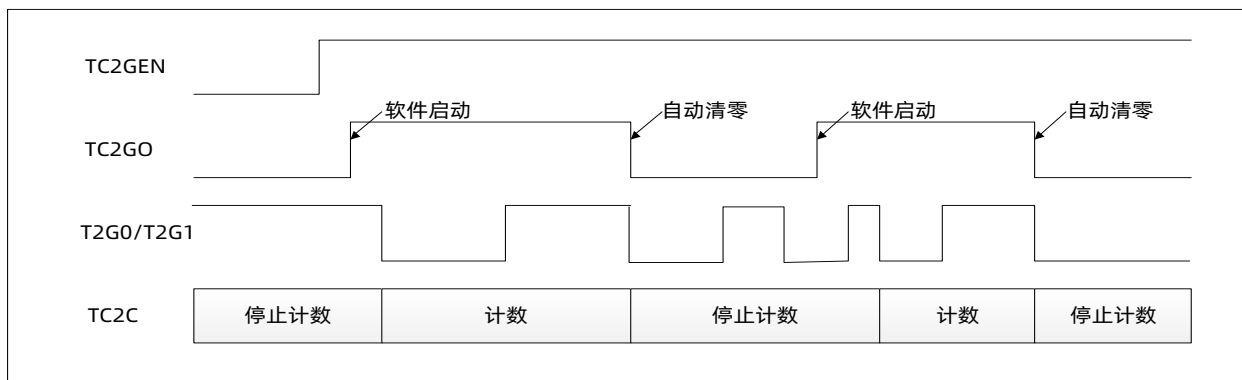
8.5.4 门控-上升沿到上升沿模式

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：



8.5.5 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PWM0

9.1 概述

SUM08F041 有 1 路基于 TC0 为时基的 8 位分辨率 PWM。

9.2 PWM0CR 控制寄存器

	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: PWM模块使能位

0 = 关闭PWM0

1 = 使能PWM0

Bit 6

PWM0OE: PWM正相波形输出使能位

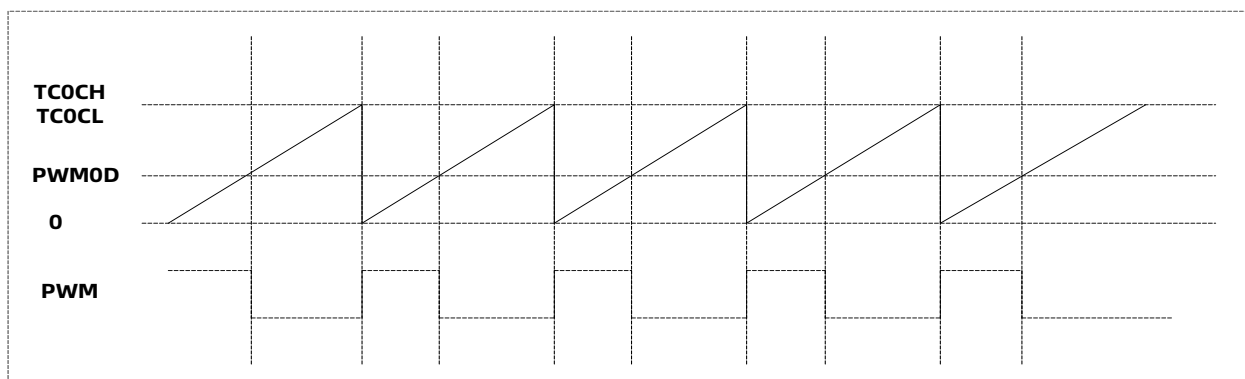
0 = 端口用作IO

1 = 端口输出PWM0波形

9.3 PWM0D 数据寄存器

	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

9.4 PWM0 波形图



10 脉宽调制模块 PWM1

10.1 概述

SUM08F041有 4 路基于 TC1 为时基的 8 位分辨率 PWM。

10.2 PWM1CR 控制寄存器

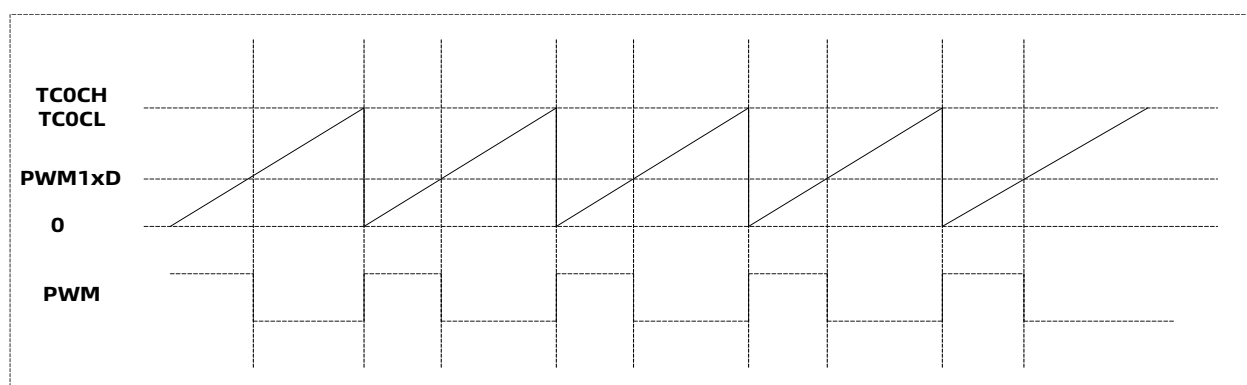
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1CR	PWM13EN	PWM13OE	PWM12EN	PWM12OE	PWM11EN	PWM11OE	PWM10EN	PWM10OE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

- Bit 7 **PWM13EN:** PWM13模块使能位
 0 = 关闭PWM13
 1 = 使能PWM13
- Bit 6 **PWM13OE:** PWM13输出控制
 0 = PWM13信号不从管脚输出，管脚用做IO
 1 = PWM13信号从管脚输出
- Bit 5 **PWM12EN:** PWM12模块使能位
 0 = 关闭PWM12
 1 = 使能PWM12
- Bit 4 **PWM12OE:** PWM12输出控制
 0 = PWM12信号不从管脚输出，管脚用做IO
 1 = PWM12信号从管脚输出
- Bit 3 **PWM11EN:** PWM11模块使能位
 0 = 关闭PWM11
 1 = 使能PWM11
- Bit 2 **PWM11OE:** PWM11输出控制
 0 = PWM11信号不从管脚输出，管脚用做IO
 1 = PWM11信号从管脚输出
- Bit 1 **PWM10EN:** PWM10模块使能位
 0 = 关闭PWM10
 1 = 使能PWM10
- Bit 0 **PWM10OE:** PWM10输出控制
 0 = PWM10信号不从管脚输出，管脚用做IO
 1 = PWM10信号从管脚输出

10.3 PWM1xD 数据寄存器 (x = 0/1/2/3)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM10D	PWM10D[7:0]							
PWM11D	PWM11D[7:0]							
PWM12D	PWM12D[7:0]							
PWM13D	PWM13D[7:0]							
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

10.4 PWM1xD 波形图



注：由于四路 PWM 共用一个 TC1 作为 PWM 的时基，所以四路 PWM 的频率是不能独立调节的

11 通用串行通讯口 (USART)

11.1 概述

SUM08F041 有一路异步串行通讯口，支持异步全双工模式和同步半双工模式。

11.2 TXCR 发送控制寄存器

	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

0 = 屏蔽USART发送功能

1 = 使能USART发送功能

Bit 6 **TMCLR**: 发送寄存器空标志

0 = 正在发送数据，移位寄存器不空

1 = 数据已发送，移位寄存器空

Bit 5 **SYNC**: 同步模式

0 = 选择异步模式

1 = 选择同步模式

Bit 4 **TX9**: 数据长度选择

0 = 8位数据

1 = 9位数据

Bit 3 **SLAVE**: 同步发送/接收模式

0 = Master

1 = SLAVE

Bit [2:1] **SPD[1:0]**: 发送接收速度选择

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

Bit 0 **TXD9**: 发送数据第9位数据

11.3 TXREG 发送数据寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TXREG	TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0
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

11.4 RXCR 接收控制寄存器

	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	1

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

11.5 RXREG 接收数据寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RXREG	RXD7	RXD6	RXD5	RXD4	RXD3	RXD2	RXD1	RXD0
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

11.6 BRGDH 波特率寄存器高位

	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 7 **SBYTE**: 同步接收模式选择

0 = 多字节接收

1 = 单字节接收, 接收完一个字节后自动清除SREN

Bit [1:0] **BRGD [9:8]**: BRGD 第9位和第8位

11.7 BRGDL 波特率寄存器低位

	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

Bit [7:0] **BRGD [7:0]**: BRGD 第7位和第0位

11.8 USART 使用说明

11.8.1 波特率设置

通过设置 BRGD 和 SPD (波特率分频比) 来获得所需的波特率。

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

常用波特率设置 ($F_{osc}=16MHz$)

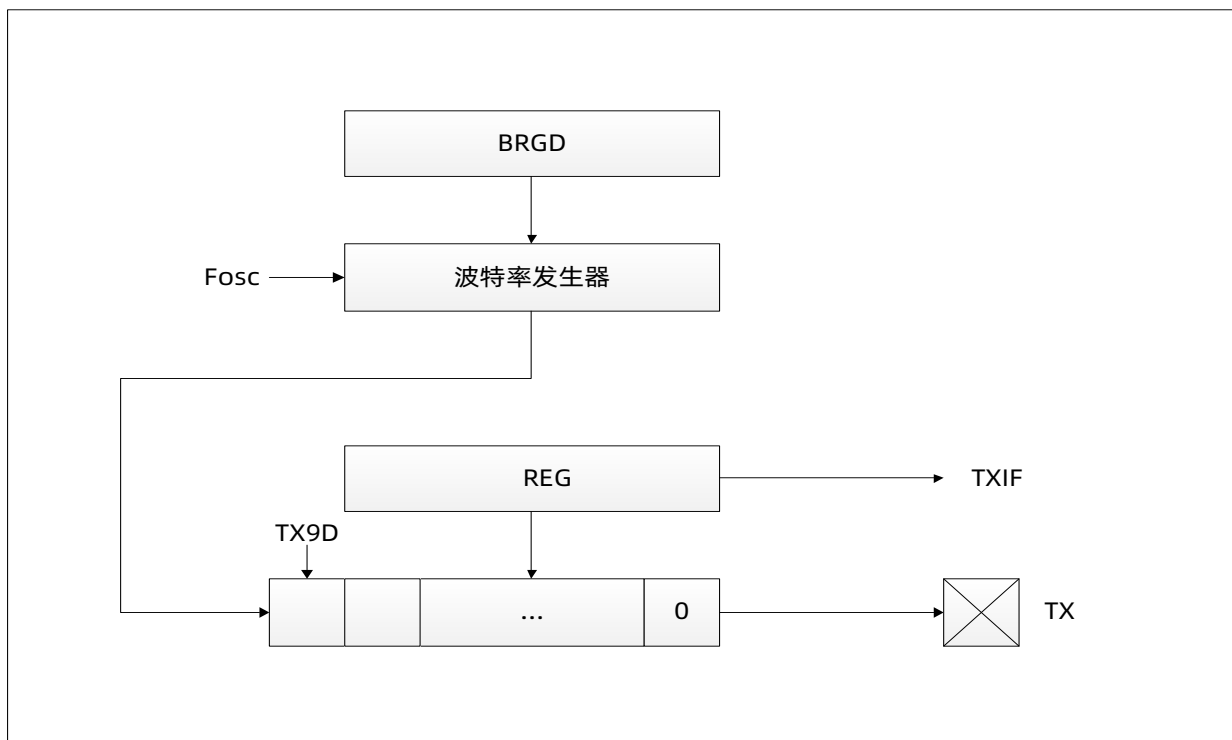
常用波特率	N (波特率分频比)	BRGD	偏差
300	256	0x0CF	0.16%
600	256	0x067	0.16%
1200	64	0x0CF	0.16%
2400	64	0x067	0.16%
4800	16	0x0CF	0.16%
9600	16	0x067	0.16%
19200	4	0x0CF	0.16%
38400	4	0x067	0.16%
57600	4	0x044	0.64%
115200	4	0x022	-0.79%

11.8.2 异步发送

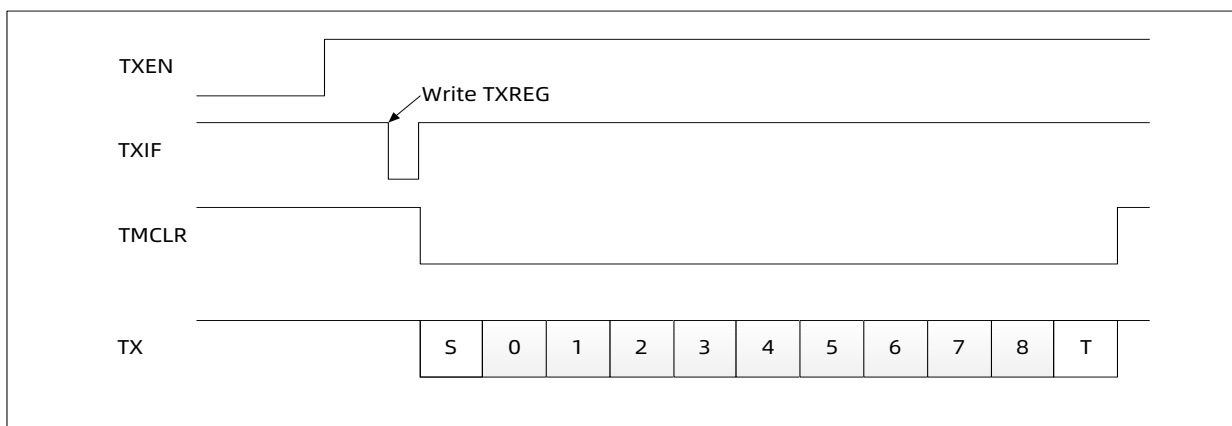
当 TXEN 使能时，TXIF 中断标志为 1 说明 TXREG 发送寄存器为空，TMCLR=1 说明发送移位寄存器为空，发送器处于空闲状态。

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。

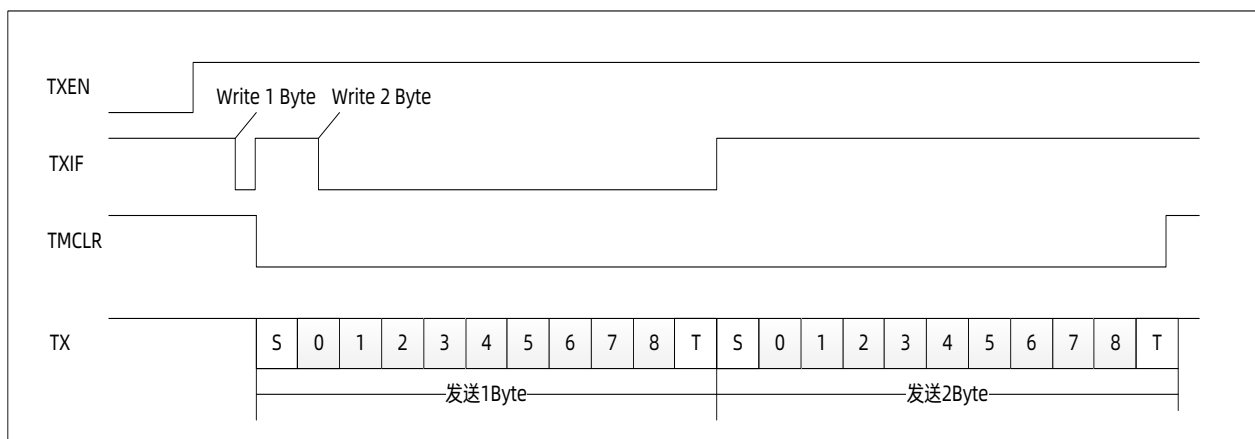
当 TXIF 为 0 时写入 TXREG，将覆盖上次写入数据。



单字节发送：



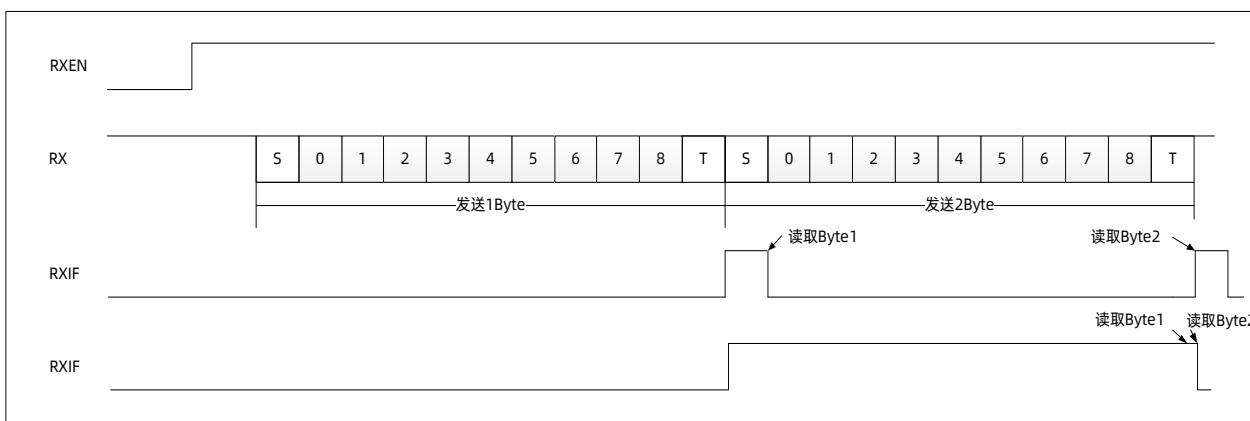
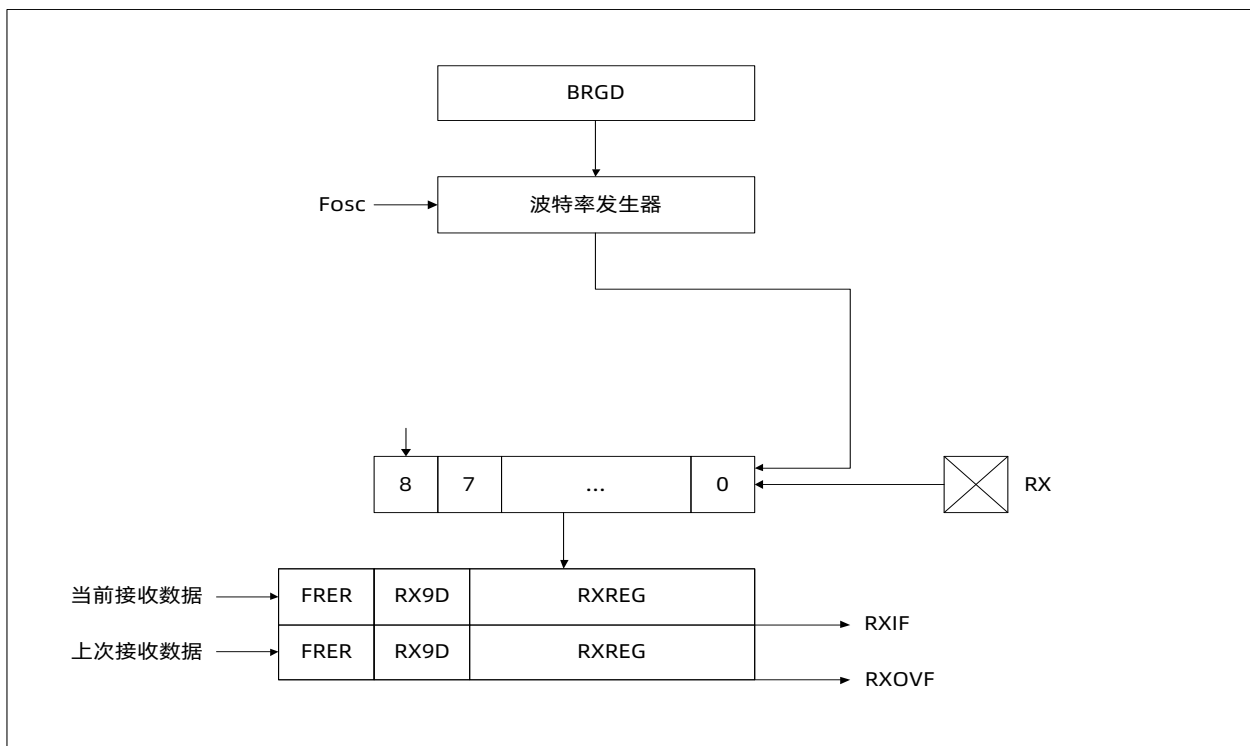
多字节发送：



- 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 TMCLR =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

11.8.3 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。



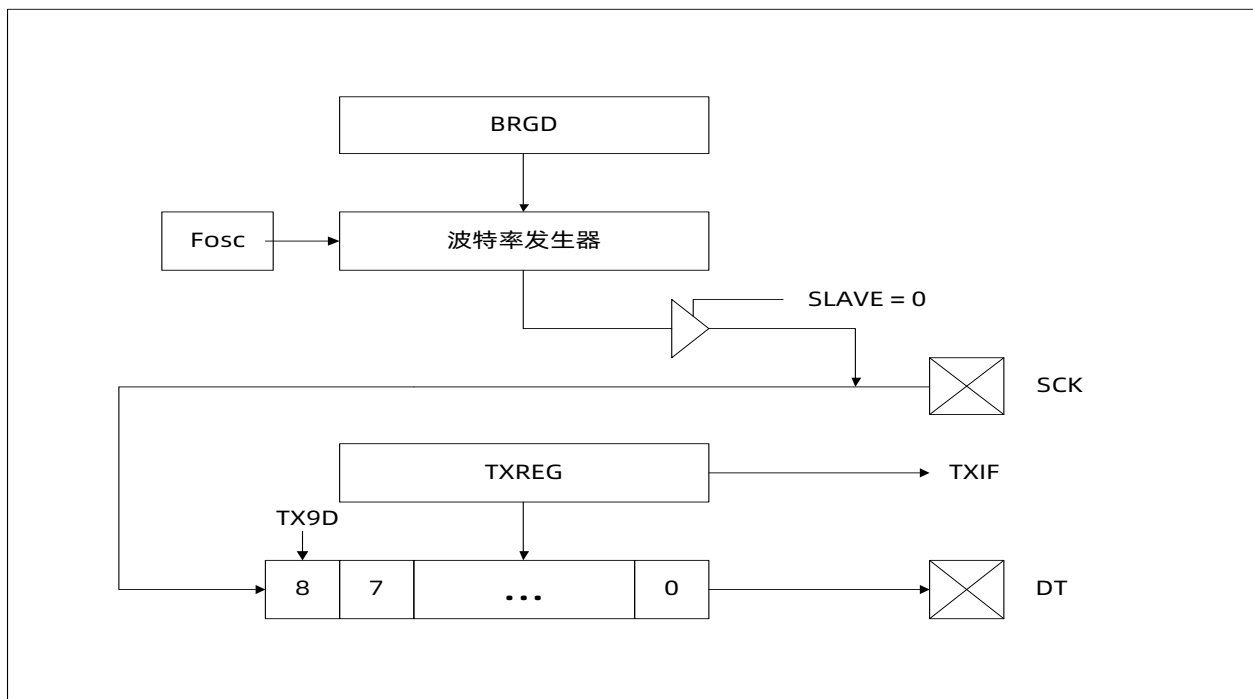
- STEP1: 设置波特率 SSPD=X, BRGD=X, SYNC=0
- STEP2: 设置 RXEN=1
- STEP3: 等待接收完成 RXIF=1
- STEP4: 判断 FRER=0, 若为 1, 帧格式错误, 舍弃数据
- STEP5: 读取 RX9D
- STEP6: 读取 RXREG, 重复 3-6

11.8.4 同步发送

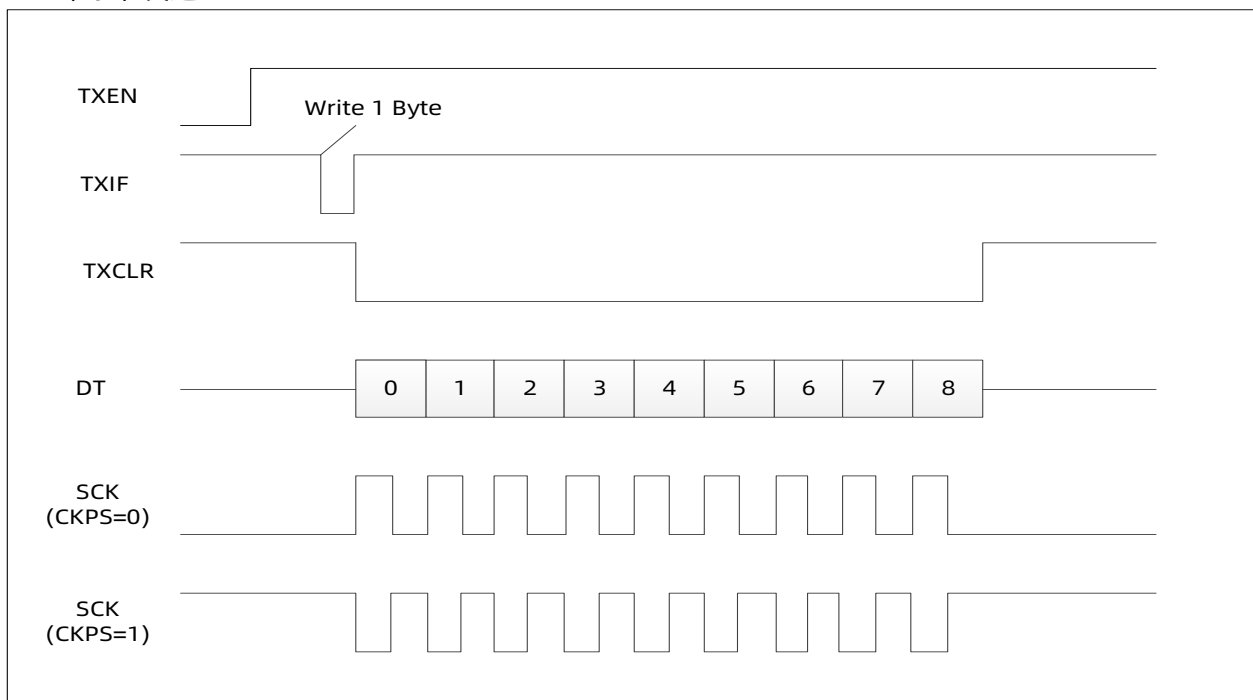
当 TXEN=1, SYNC=1 时, 使能同步发送功能。CKPS 选择发送时钟极性, TXIF 中断标志为 1 说明 TXREG 发送寄存器为空, TMCLR=1 说明发送移位寄存器为空, 发送器处于空闲状态。

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。

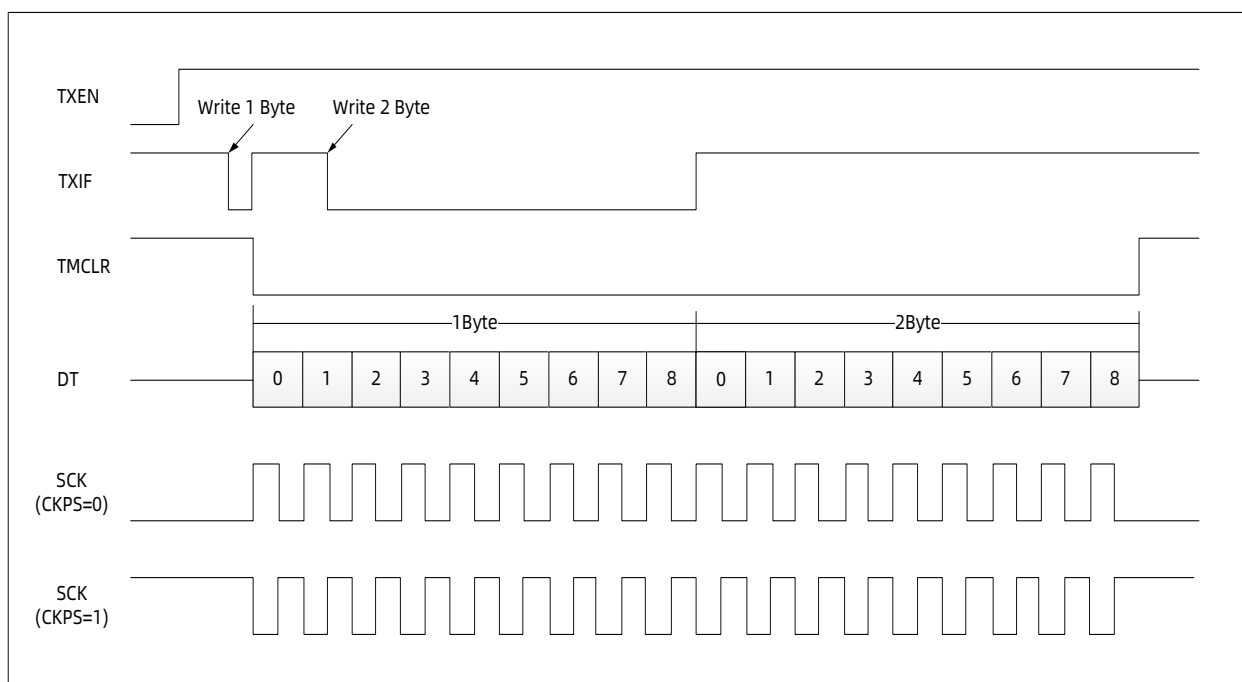
当 TXIF 为 0 时写入 TXREG, 将覆盖上次写入数据。



单字节发送:



多字节发送：



参考操作步骤 SLAVE=0:

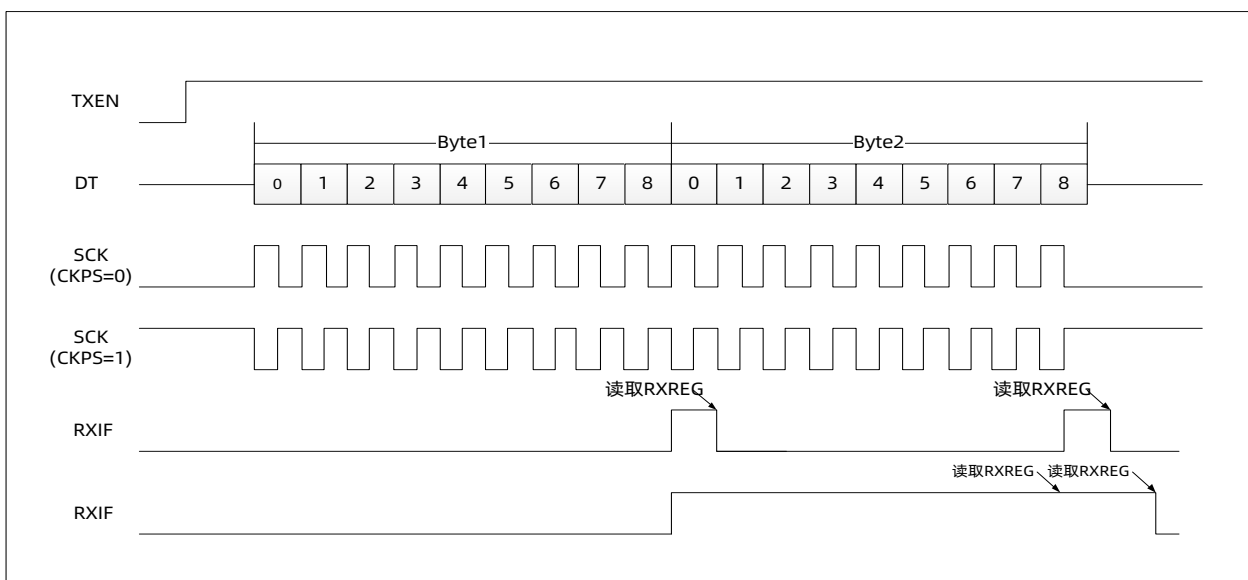
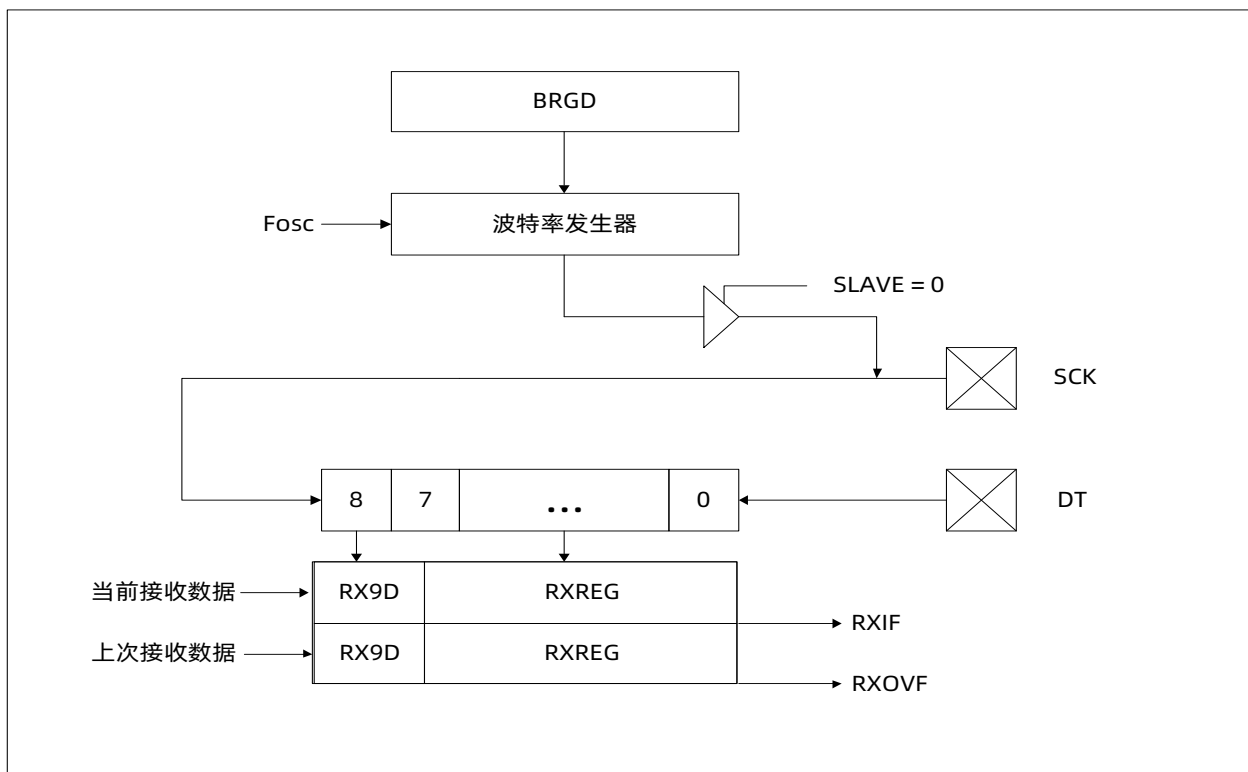
- STEP1: 设置波特率 SSPD=X, BRGD=X, SYNC=1
- STEP2: 设置 TXEN=1, 设置数据模式 TX9=X
- STEP3: 写入数据高位 TXD9
- STEP4: 写入 TXREG, 启动发送
- STEP5: 当 TXIF=1 时, 写入 TXREG 发送下一个字节
- STEP6: 重复 STEP5, 直到该帧数据发送完成

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

11.8.5 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

11.8.6 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

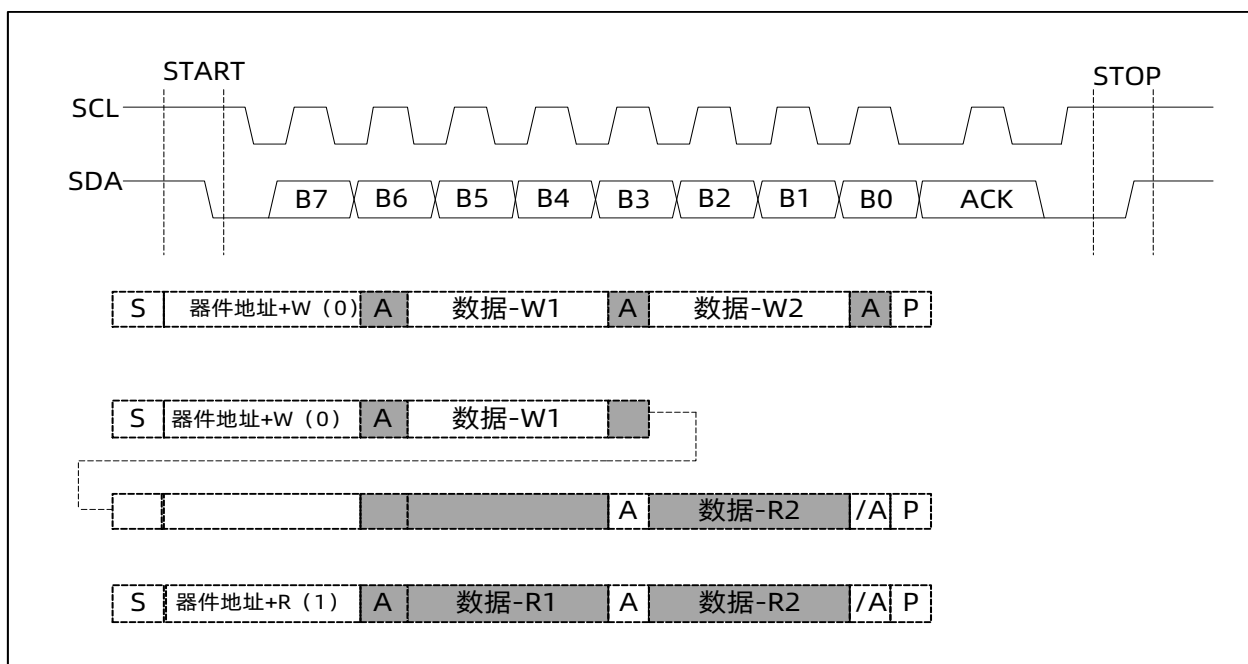
同步接收时, 若作为主机, 则休眠状态下不工作; 作为从机, 则接收 1 个字节完成后唤醒 CPU

12 串行通讯口 (I2C)

12.1 概述

SUM08F041 支持高速 I2C (400K) Slave。

12.2 通讯波形示意



12.3 I2CCON I2C 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CCON	I2CEN	R_W	D_A	BF	NACK	-	-	-
读/写	R/W	R	R	R	R	-	-	-
复位后	0	0	0	0	0	-	-	-

Bit 7 **I2CEN:** 使能I2C通讯

0 = 屏蔽I2C通讯功能

1 = 使能I2C通讯功能

Bit 6 **R_W:** 发送寄存器空标志

0 = Master写数据（复位后、START、STOP后）

1 = Master读数据（主机发接收数据命令）

Bit 5 **D_A:** 数据地址标志

0 = 主机发送的是地址

1 = 主机发送的是数据(START后发过的第一个数据)

Bit 4 **BF:** 数据缓冲区满

R_W = 0

0 = 已读或未接受到数据

1 = 从机接收到数据，未读

R_W = 1

0 = 数据已发送或正在发送

1 = 有数据待发送

Bit 3 **NACK:** Master读数据ACK

0 = Master继续接收下一个数据

1 = Master停止接收数据

12.4 I2CBUF 数据寄存器

	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

12.5 I2CADDR 地址寄存器

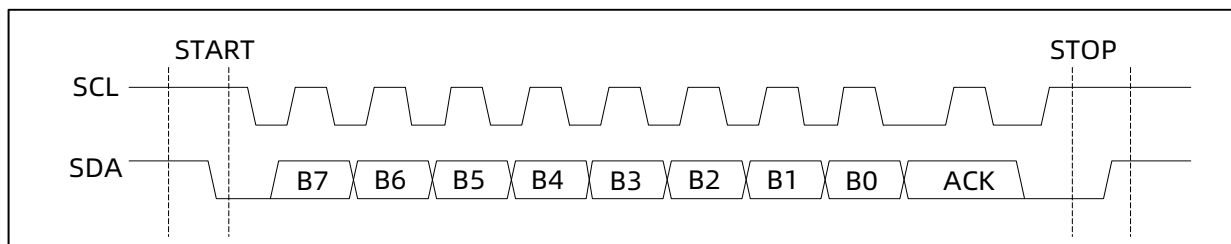
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CADDR	I2CADDR7	I2CADDR6	I2CADDR5	I2CADDR4	I2CADDR3	I2CADDR2	I2CADDR1	I2CADDR0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
复位后	X	X	X	X	X	X	X	X

Bit[7:0] I2CADDR[7:0]: 从机地址
地址 (I2C 写入): I2CADDR
地址 (I2C 读取): I2CADDR+1

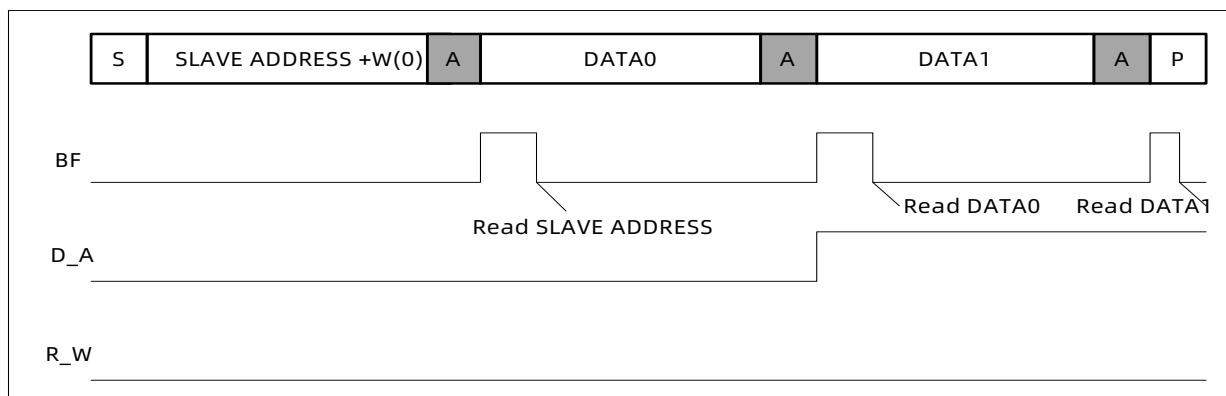
12.6 唤醒及休眠模式下通讯

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。

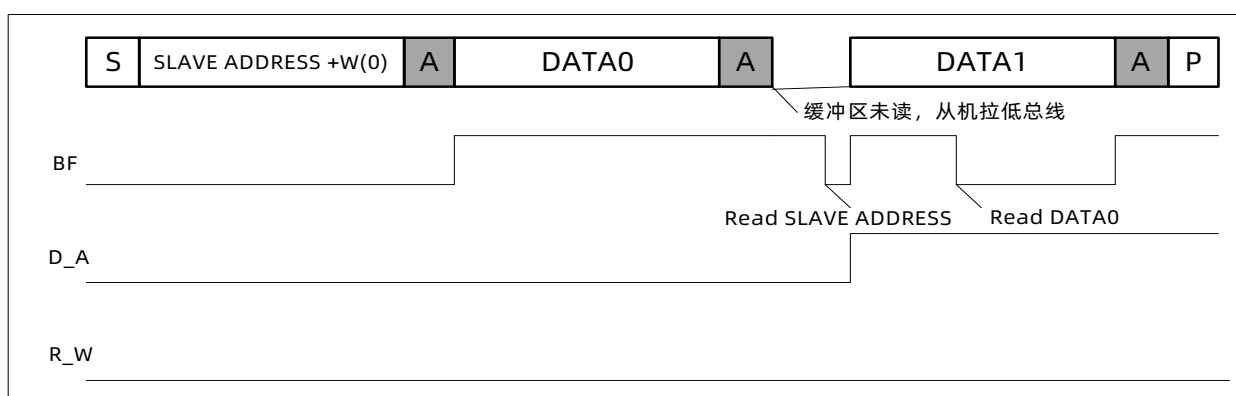
12.7 通讯波形图



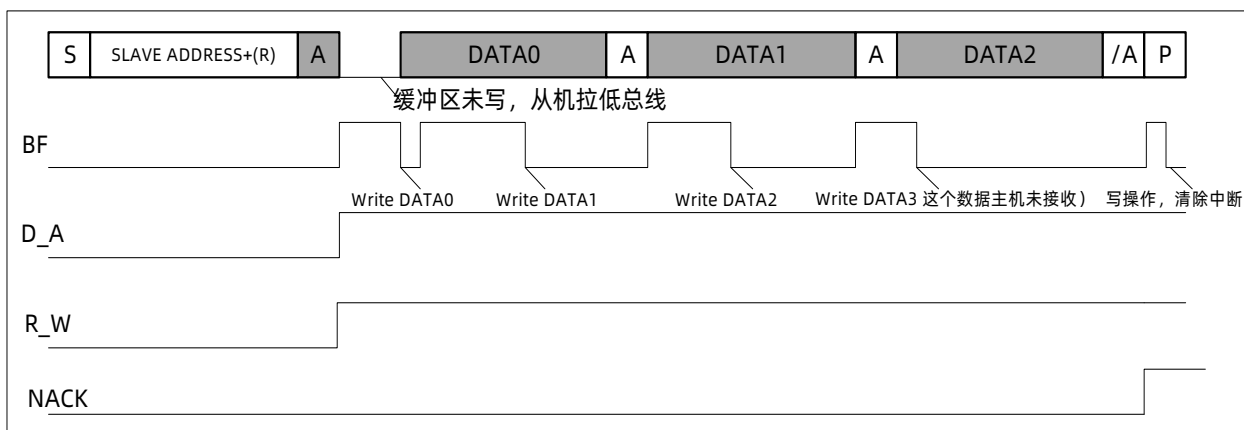
地址匹配后，主机向从机写入两个数据



两个数据没有读取，从机拉低 SCL，等待数据读取



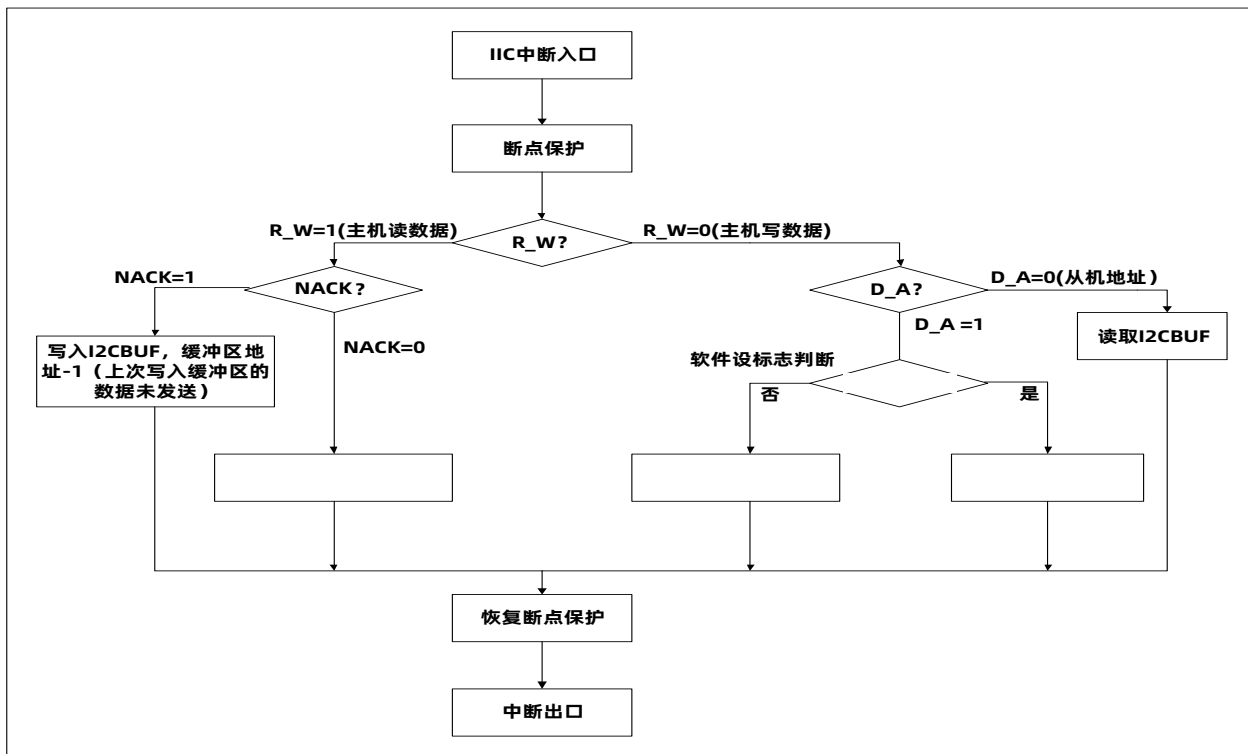
I2C Master 读数据此操作



12.8 应用示例

SUM08F041 作为从机，允许主机通过 I2C 接口对其 RAM 进行读写操作。

12.8.1 从机软件流程图



12.8.2 例程

```

LIST P= DN3083
#include    DN3083 .inc
#define                                FirstByte      2
sFSR0      EQU    H'12'
R_SlaveBuffAddr EQU    H'37'    ;//数据地址
R_I2Cflag  EQU    H'3A'    ;//软件标志
IIC_Data   EQU    H'3B'

        ORG      0000H
        GOTO     MainProgram
        ORG      0008H
        GOTO     Interrupt

MainProgram:
IIC_Init:                                ;//IIC 初始化
; //1、IO 端口设置
        BSET     PUC,5                ;//0:上拉关闭 1:上拉使能
        BSET     PUC,4                ;//0:上拉关闭 1:上拉使能
        BCLR     OEC,5                ;//0:输入 1:输出
        BCLR     OEC,4                ;//0:输入 1:输出
; //2、设置串行通讯功能
        BSET     I2CCON,7              ;//使能串行通讯功能
; //3、设置从机地址
        MOVIA    0xA0
        MOVAR    I2CADDR
; //4、中断使能设置
        BCLR     INTF2,I2CIF          ;//清中断标志
        BSET     INTCR2, I2CIE
        CLRR     INTF2
        BSET     OPTION, GIE

Main:
        CLRWDT
        GOTO     Main

Interrupt:
        PUSH     ;//进栈、保存 A、STATUS
; //中断处理程序
        MOVR     FSR0,A                ;//保存 FSR0
        MOVAR    sFSR0                ;//中断中用到 FSR0,保护
        JBTS0    INTF2, I2CIF
        GOTO     I2C_SLAVE_SSPINT
        NOP
    
```

```

Interrupt_End:
; // 中断结束
    MOVR    sFSR0,A           ; // 恢复 FSR0
    MOVAR   FSR0              ; // 中断中用到 FSR0,保护
    POP     ; // 出栈、恢复 A、STATUS
    RETIE

I2C_SLAVE_SSPINT:
    JBTS1   I2CCON, 6
    GOTO    I2CMasterWrite
    GOTO    I2CMasterRead

; // 判断主机是发送地址还是数据
I2CMasterWrite:
    JBTS0   I2CCON, 5
    GOTO    I2C_D_A_Send_data

; // 主机向从机写地址 读取 I2CBUF
    MOVR    I2CBUF, A
    BSET    R_I2Cflag, FirstByte
    GOTO    Interrupt_End

; // 主机向从机读数据
I2CMasterRead:
    JBTS0   I2CCON, NACK
    GOTO    I2C_NACK_1
    GOTO    I2C_NACK_0

; // 读取缓冲区数据,写入 I2CBUF,缓冲区地址+1
I2C_NACK_0:
    MOVR    R_SlaveBuffAddr, A
    MOVAR   FSR0
    MOVR    INDF0, A
    MOVAR   I2CBUF
    INCR    R_SlaveBuffAddr, R
    GOTO    Interrupt_End

; // 写入 I2CBUF,缓冲区地址 -1 (上次写入缓冲区的数据未发送)
I2C_NACK_1:
    DECR    R_SlaveBuffAddr,R
    MOVAR   I2CBUF
    GOTO    Interrupt_End

    END

```

13 LCD

13.1 概述

SUM08F041 LCD 驱动模块由 LCD 控制寄存器以及数据寄存器组成，可以驱动 LCD/LED 段码。

- 可选择 2~6COM
- 可调节帧频
- 1/2、1/3BIAS 电压可选
- 内部电阻模式（可设 VLCD 电压）
- LED 模式支持多级亮度设置

13.2 LCD 控制寄存器

13.2.1 控制寄存器 LCDCR0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LDCR0	LCDEN	LCDM	LCDR	BIAS	-	-	-	-
读/写	R/W	R/W	R/W	R/W	-	-	-	-
复位后	0	0	0	0	-	-	-	-

- Bit 7 **LCDEN:** LCD 功能使能位
0 = 屏蔽 LCD 功能
1 = 使能 LCD 功能
- Bit 6 **LCDM:** LCD 驱动模式选择
0 = B 类驱动模式
1 = A 类驱动模式
- Bit 5 **LCDR:** LCD 小电阻模式开启选择
0 = 小电阻模式未始终开启
1 = 小电阻模式始终开启
- Bit 4 **BIAS:** LCD BIAS 选择
0 = 1/2BIAS
1 = 1/3BIAS

13.2.2 控制寄存器 LCDCR1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCR1	LCDFR3	LCDFR2	LCDFR1	LCDFR0	LCDCKS	CIO52	CIO51	CIO50
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] **LCDFR:** 帧频选择& LED 扫描频率

FLCD	LCDFR	帧频				
		6Duty	5Duty	4Duty	3Duty	2Duty
32768	0	546	546	512	546	512
32768	1	273	273	256	273	256
32768	2	182	182	171	182	171
32768	3	137	137	128	137	128
32768	4	109	109	102	109	102
32768	5	91	91	85	91	85
32768	6	78	78	73	78	73
32768	7	68	68	64	68	64
32768	8	61	61	57	61	57
32768	9	55	55	51	55	51
32768	10	50	50	47	50	47
32768	11	46	46	43	46	43
32768	12	42	42	39	42	39
32768	13	39	39	37	39	37
32768	14	36	36	34	36	34
32768	15	34	34	32	34	32

Bit 3 **LCDCKS:** LCD 时钟选择

LCDCKS	LCD 时钟源选择
0	高频时钟的 512 分频
1	低频时钟 (32KHz)

Bit [2:0] **CIO5:** LCD Duty&IO 选择位

CIO5[2:0]	LCD Duty & IO 选择
000	Com0-5 作为 IO 口
001	Com0-1 作为 Com 口,Com2-5 作为 IO 口
010	Com0-2 作为 Com 口,Com3-5 作为 IO 口
011	Com0-3 作为 Com 口,Com4-5 作为 IO 口
100	Com0-4 作为 Com 口,Com5 作为 IO 口
101	Com0-5 作为 Com 口

13.2.3 控制寄存器 LCDCR2

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCR2	LEDEN	LEDPOL	-	SBRIGH4	SBRIGH3	SBRIGH2	SBRIGH1	SBRIGH0
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 **LEDEN:** LED 功能使能位

0 = 屏蔽 LED 功能

1 = 使能 LED 功能

Bit 6 **LEDPOL:** LED 驱动极性选择

0 = 0 选通

1 = 1 选通

Bit [4:0] **SBRIGH[4:0]:** LCD 小电阻开启时间/LED 亮度调节

当 LCD 使能时, SBRIGH 为 LCD 小电阻开启时间

$T_{spd} = (SBRIGH * 32) \mu s$ ($SBRIGH > 0$)

$T_{spd} = 0 \mu s$ ($SBRIGH = 0$)

当 LED 使能时, SBRIGH 为 LED 亮度调节

LED 亮度随着 SBRIGH 增加而增加 $SBRIGH < 2 * (LCDFR + 1)$

LED 亮度最大 $SBRIGH > 2 * (LCDFR + 1)$ 或 $SBRIGH = 0$

13.2.4 控制寄存器 LCDCR3

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCR3	-	-	-	VLCD4	VLCD3	VLCD2	VLCD1	VLCD0
读/写	-	-	-	R/W	R/W	R/W	R/W	R/W
复位后	-	-	-	0	0	0	0	0

Bit [4:0] **VLCD:** VLCD 电压选择

VLCD[4:0]	VLCD	VLDOS[4:0]	VLCD
00000	0.50 VDD	01001	0.78 VDD
00001	0.53 VDD	01010	0.81 VDD
00010	0.56 VDD	01011	0.84 VDD
00011	0.59 VDD	01100	0.88 VDD
00100	0.63 VDD	01101	0.91 VDD
00101	0.66 VDD	01110	0.94 VDD
00110	0.69 VDD	01111	0.97 VDD
00111	0.72 VDD	1xxxx	VDD
01000	0.75 VDD		

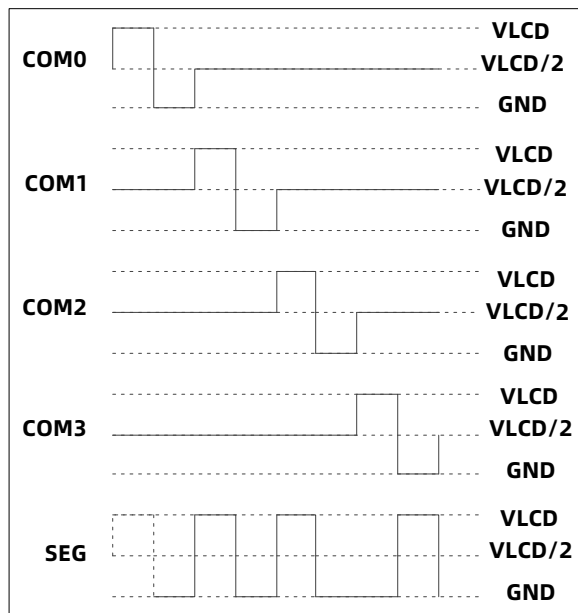
13.3 LCDDSxx : LCD 显示数据寄存器

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
LCDDS00	170H	-	-	C5S0	C4S0	C3S0	C2S0	C1S0	C0S0
LCDDS01	171H	-	-	C5S1	C4S1	C3S1	C2S1	C1S1	C0S1
LCDDS02	172H	-	-	C5S2	C4S2	C3S2	C2S2	C1S2	C0S2
LCDDS03	173H	-	-	C5S3	C4S3	C3S3	C2S3	C1S3	C0S3
LCDDS04	174H	-	-	C5S4	C4S4	C3S4	C2S4	C1S4	C0S4
LCDDS05	175H	-	-	C5S5	C4S5	C3S5	C2S5	C1S5	C0S5
LCDDS06	176H	-	-	C5S6	C4S6	C3S6	C2S6	C1S6	C0S6
LCDDS07	177H	-	-	C5S7	C4S7	C3S7	C2S7	C1S7	C0S7
LCDDS08	178H	-	-	C5S8	C4S8	C3S8	C2S8	C1S8	C0S8
LCDDS09	179H	-	-	C5S9	C4S9	C3S9	C2S9	C1S9	C0S9
LCDDS10	17AH	-	-	C5S10	C4S10	C3S10	C2S10	C1S10	C0S10
LCDDS11	17BH	-	-	C5S11	C4S11	C3S11	C2S11	C1S11	C0S11
LCDDS12	17CH	-	-	C5S12	C4S12	C3S12	C2S12	C1S12	C0S12
LCDDS13	17DH	-	-	C5S13	C4S13	C3S13	C2S13	C1S13	C0S13
LCDDS14	17EH	-	-	C5S14	C4S14	C3S14	C2S14	C1S14	C0S14
LCDDS15	17FH	-	-	C5S15	C4S15	C3S15	C2S15	C1S15	C0S15
LCDDS16	180H	-	-	C5S16	C4S16	C3S16	C2S16	C1S16	C0S16
LCDDS17	181H	-	-	C5S17	C4S17	C3S17	C2S17	C1S17	C0S17
LCDDS18	182H	-	-	C5S18	C4S18	C3S18	C2S18	C1S18	C0S18
LCDDS19	183H	-	-	C5S19	C4S19	C3S19	C2S19	C1S19	C0S19
LCDDS20	184H	-	-	C5S20	C4S20	C3S20	C2S20	C1S20	C0S20
LCDDS21	185H	-	-	C5S21	C4S21	C3S21	C2S21	C1S21	C0S21
LCDDS22	186H	-	-	C5S22	C4S22	C3S22	C2S22	C1S22	C0S22
LCDDS23	187H	-	-	C5S23	C4S23	C3S23	C2S23	C1S23	C0S23
LCDDS24	188H	-	-	C5S24	C4S24	C3S24	C2S24	C1S24	C0S24
LCDDS25	189H	-	-	C5S25	C4S25	C3S25	C2S25	C1S25	C0S25
LCDDS26	18AH	-	-	C5S26	C4S26	C3S26	C2S26	C1S26	C0S26
LCDDS27	18BH	-	-	C5S27	C4S27	C3S27	C2S27	C1S27	C0S27
LCDDS28	18CH	-	-	C5S28	C4S28	C3S28	C2S28	C1S28	C0S28
LCDDS29	18DH	-	-	C5S29	C4S29	C3S29	C2S29	C1S29	C0S29
LCDDS30	18EH	-	-	C5S30	C4S30	C3S30	C2S30	C1S30	C0S30
LCDDS31	18FH	-	-	C5S31	C4S31	C3S31	C2S31	C1S31	C0S31

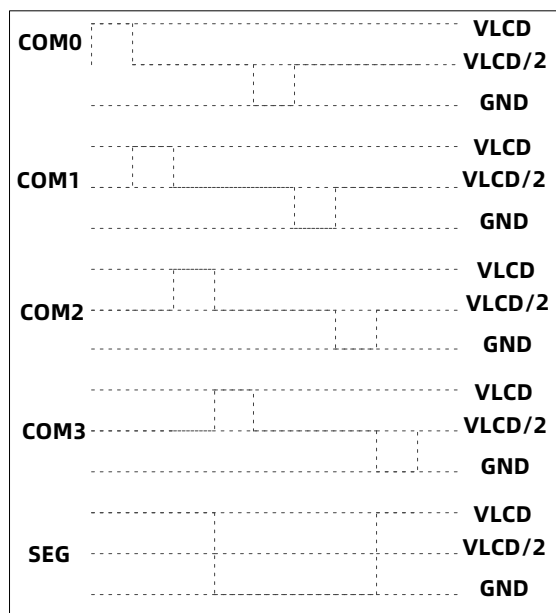
13.4 LCD 驱动波形

1/4Duty、1/2BIAS (SEG 输入 08)

A 类驱动模式

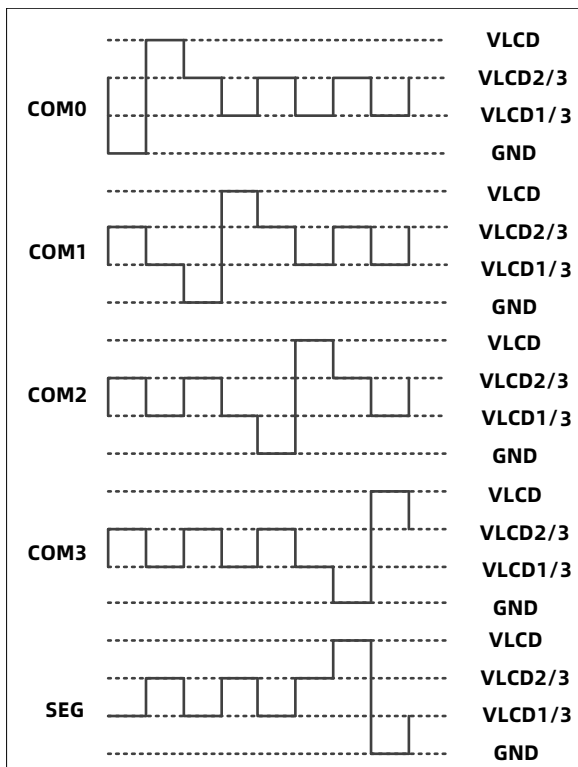


B 类驱动模式

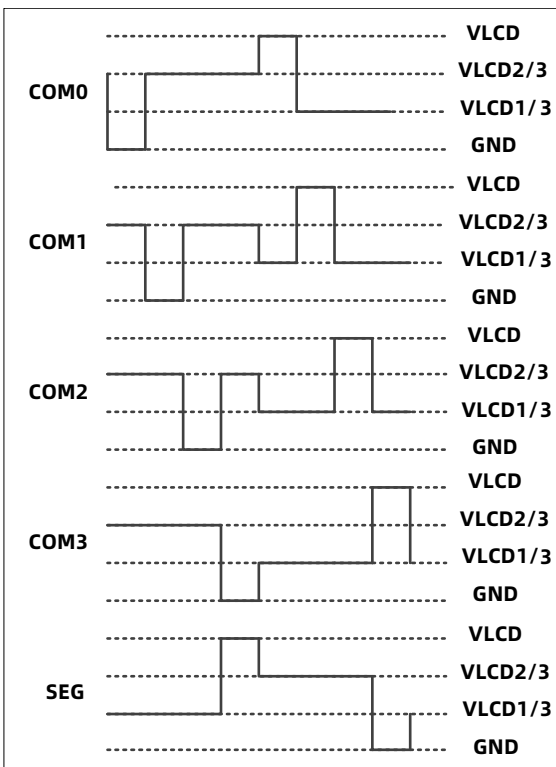


1/4Duty、1/3BIAS (SEG 输入 08)

A 类驱动模式

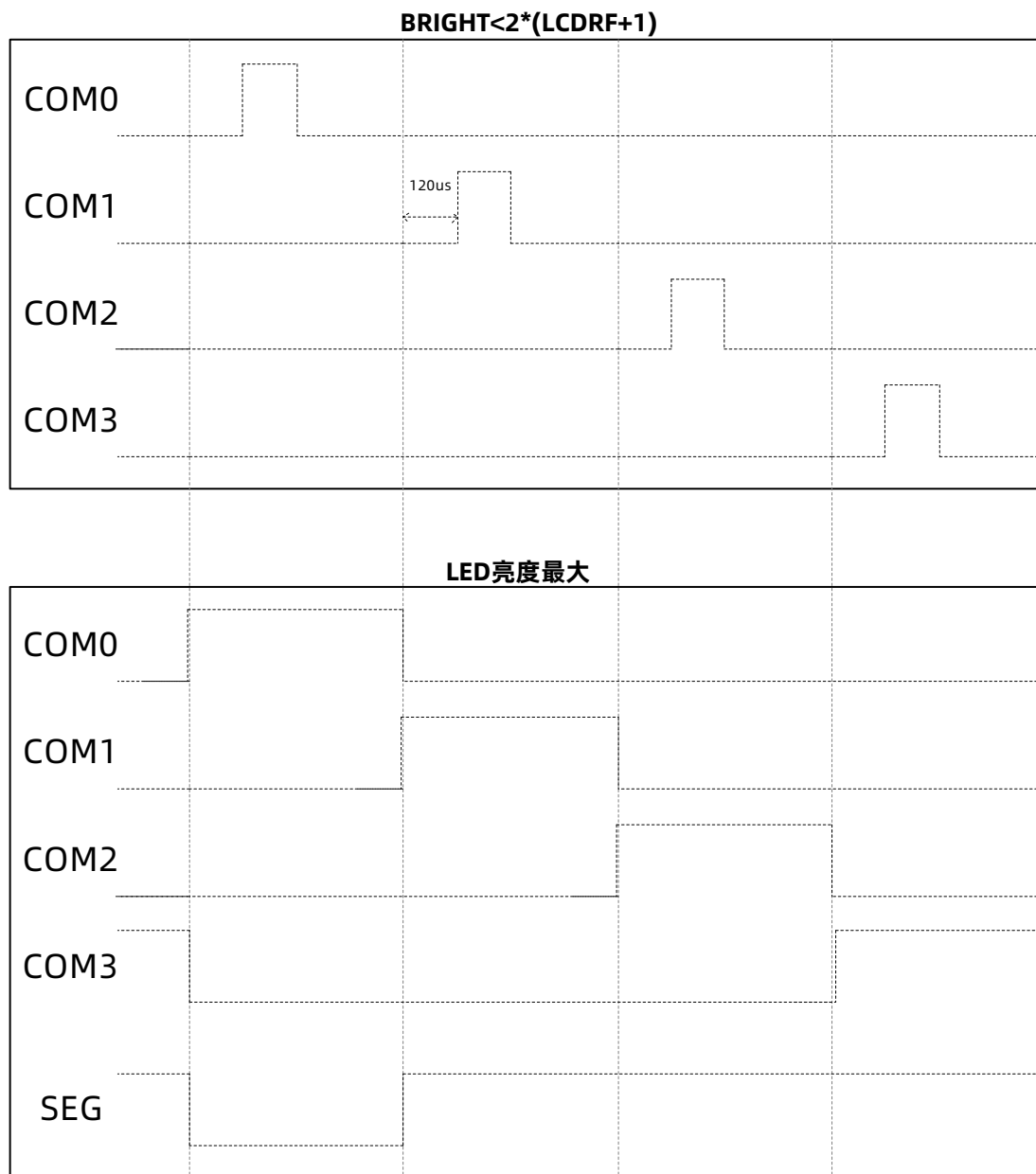


B 类驱动模式



13.5 LED 驱动波形

COM0-3 全部作为 LED 使用

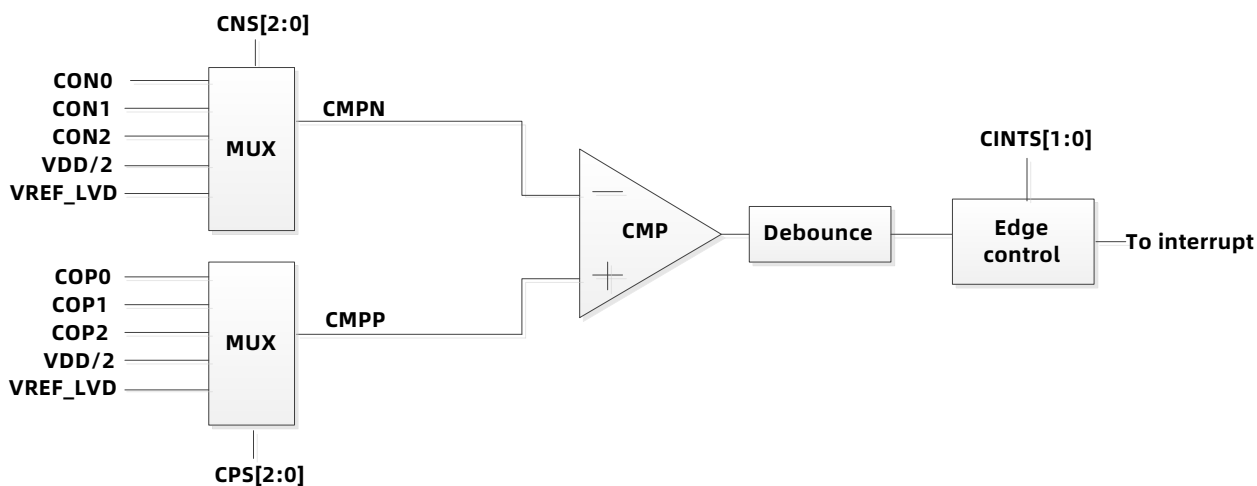


14 比较器 (CMP)

14.1 概述

SUM08F041 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，可做 LVD。

14.2 比较器框图



14.3 CMPC0 比较器控制寄存器 0

	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
复位后	0	0	0	0	0	0	0	0

Bit 7 **CMPEN**: 比较器使能控制位

0 = 关闭比较器

1 = 使能比较器

Bit 6 **CMPOUT**: 比较器输出位

0 = COP脚输入电压小于CON脚

1 = COP脚输入电压大于CON脚

Bit [5:3] **CMPNS[2:0]**: 比较器反相输入信号选择位

CMPNS[1:0]	输入信号选择
000	CON0
001	CON1
010	CON2
011	VDD*1/2
100	VREF_LVD

Bit [2:0] **CMPPS[2:0]**: 比较器正相输入信号选择位

CMPPS[1:0]	输入信号选择
000	COP0
001	COP1
010	COP2
011	VDD*1/2
100	VREF_LVD

注: (1) 用户使用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。

(2) 比较器信号输入端多选用外部端口, 则该端口会自动设置为模拟口。

14.4 CMPC1 比较器控制寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPC1	-	CPVRC	CMPOFM	CMPOF4	CMPOF3	CMPOF2	CMPOF1	CMPOF0
R/W	-	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	-	0	0	0	0	0	0	0

Bit 6 **CPVRC**: 比较器COP端信号从端口输出

0 = 关闭

1 = COP端信号从端口输出 (端口输出CP端信号, 如果外部有干扰会影响到比较器功能)

Bit 5 **CMPOFM**: 校准模式选择

0 = 正常工作模式

1 = 失调校准模式

Bit [4:0] **CMPOF[4:0]**: 校准参数位

14.5 CMPC2 比较器参考电压选择

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPC2	CINTS1	CINTS0	-	VLVDS4	VLVDS3	VLVDS2	VLVDS1	VLVDS0
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[7:6] **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
1X	双边沿触发

Bit[4:0] **VLCD[4:0]**: LVD 电压选择

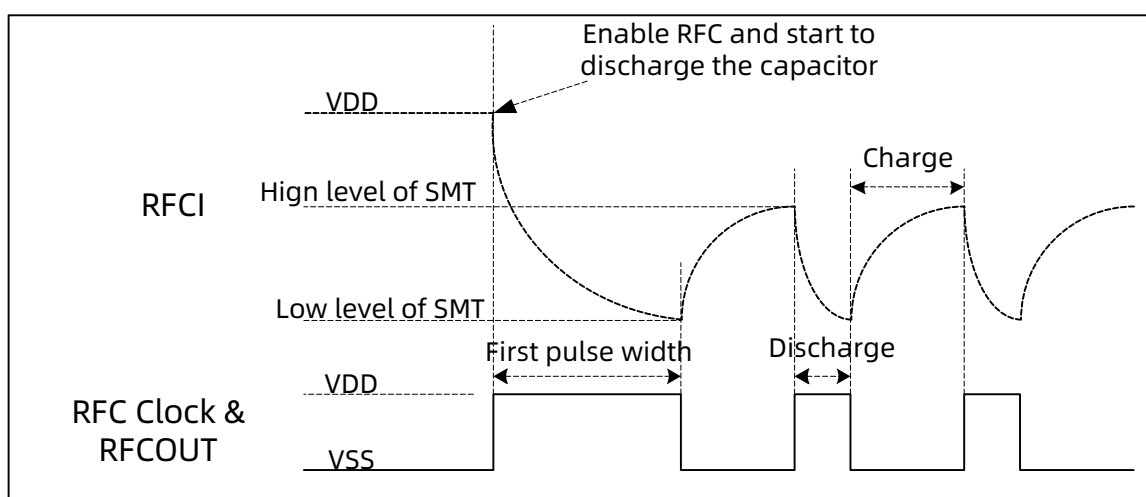
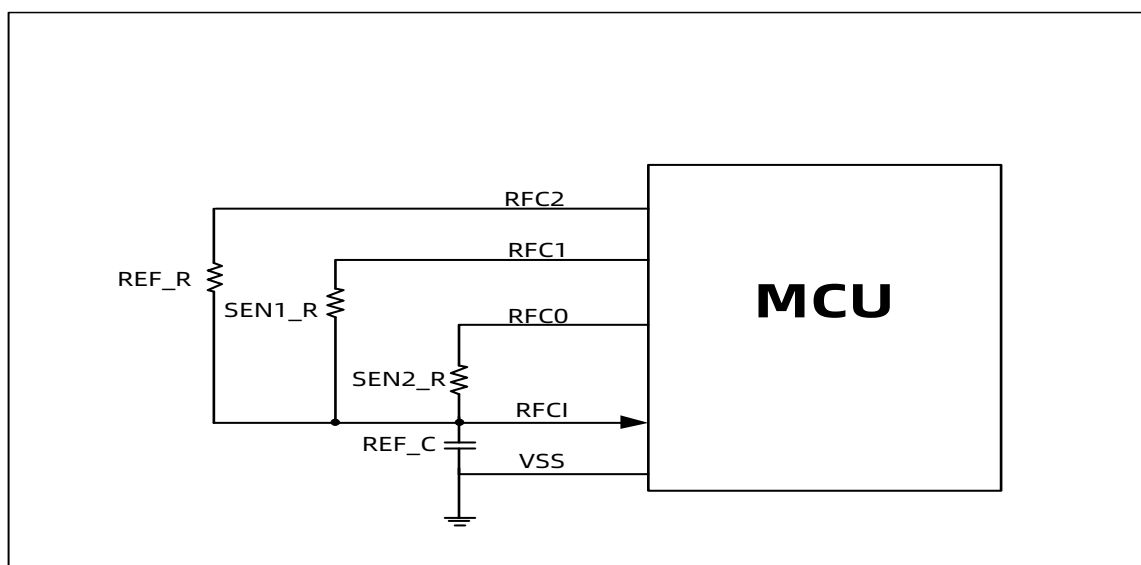
LVDS[4:0]	VREF_LVD	LVDS[4:0]	VREF_LVD	LVDS[4:0]	VREF_LVD	LVDS[4:0]	VREF_LVD
00000	1.15	01000	1.55	10000	1.95	11000	2.35
00001	1.20	01001	1.60	10001	2.00	11001	2.40
00010	1.25	01010	1.65	10010	2.05	11010	2.45
00011	1.30	01011	1.70	10011	2.10	11011	2.50
00100	1.35	01100	1.75	10100	2.15	11100	2.55
00101	1.40	01101	1.80	10101	2.20	11101	2.60
00110	1.45	01110	1.85	10110	2.25	11110	2.65
00111	1.50	01111	1.90	10111	2.30	11111	2.70

15 电阻频率转换 (RFC)

15.1 概述

SUM08F041 RFC 模块可以将各个通道不同电阻值转换成相应的频率值，然后通过 Timer2 定时器/计时器的 RFC 脉宽测试模式或频率测试模式进行转换频率信号的脉宽或频率测试。

15.2 电阻频率转换工作原理图



RFCOUT 输出到 TC2 的门控端进行脉宽计算，或输出到 TC2 的外部时钟输入端进行频率计算；通过测量各个 Sensor 的脉冲宽度及参考电阻值，可以计算各个传感器的电阻。

15.3 RFCCR RFC 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFCCR	-	RFCH2E	RFCH1E	RFCH0E	-	-	RFCH1	RFCH0
读/写	-	R/W	R/W	R/W	-	-	R/W	R/W
复位后	-	0	0	0	-	-	0	0

- Bit 6 **RFCH2E:** RFC通道2输入功能使能位
0 = 屏蔽 RFC 通道 2 输入功能, 用作 I/O 输入/输出口
1 = 使能 RFC 通道 2 输入功能, 用作 RFC 通道 2 输入口
- Bit 5 **RFCH1E:** RFC通道1输入功能使能位
0 = 屏蔽 RFC 通道 1 输入功能, 用作 I/O 输入/输出口
1 = 使能 RFC 通道 1 输入功能, 用作 RFC 通道 1 输入口
- Bit 4 **RFCH0E:** RFC通道0输入功能使能位
0 = 屏蔽 RFC 通道 0 输入功能, 用作 I/O 输入/输出口
1 = 使能 RFC 通道 0 输入功能, 用作 RFC 通道 0 输入口
- Bit [1:0] **RFCH[1:0]:** RFC通道选择位
00 = 选择 RFC 通道 0
01 = 选择 RFC 通道 1
10 = 选择 RFC 通道 2
11 = 无定义

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

注：看门狗正常溢出后，程序复位到 0000H，但是在休眠模式下看门狗溢出程序是继续往下运行。

16.2 OPTION 配置寄存器

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

- Bit 5 **TO：** 超时位
 0 = WDT发生溢出
 1 = 上电复位或清除WDT
- Bit 4 **PD：** 掉电位
 0 = 进入休眠模式
 1 = 上电复位或清除WDT

16.3 WDTC 看门狗控制寄存器

	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	-	-	-	-	-	-	-	-

注：(1) WDTC 写入 0x5A 将清除 WDT 定时器，写入其他值无效。
(2) CLRWDT 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
振荡器模式	HXT+LIRC	双系统时钟
	HIRC+LXT	
	HIRC+LIRC	
CPU 运行速度选择	4T (LVR>2.1V)	高频模式下 CPU 速度选择
	8T (LVR>1.7V)	
	16T	
	32T	
	64T	
	128T	
	256T	
晶振驱动电流	驱动 0 (普驱)	默认驱动 1
	驱动 1 (强驱)	
晶振反馈电阻	电阻 0 (小电阻)	默认电阻 0
	电阻 1 (大电阻)	
外部复位端口	作为外部复位端口	
	作为 IO 端口	
输出端口读取	从端口读取	
	从输出寄存器读取	

烧录选项	内容	说明
芯片代码加密	不使能	
	使能	
复位电压 选择	LVR=4.2V	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=4.0V	
	LVR=3.8V	
	LVR=3.6V	
	LVR=3.4V	
	LVR=3.2V	
	LVR=3.0V	
	LVR=2.8V	
	LVR=2.6V	
	LVR=2.5V	
	LVR=2.4V	
	LVR=2.3V	
	LVR=2.2V	
	LVR=2.0V (FCPU<4T)	
	LVR=1.8V (FCPU<4T)	
	LVR=1.6V (FCPU<8T)	
低频晶振电压 选择	芯片 VDD 供电	
	BIAS 供电	
WDT 使能 选择	屏蔽 WDT 功能	
	使能，绿色或休眠模式下关闭	
	始终开启 WDT 功能	
WDT 溢出 时间	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	
仿真电压 选择	VDD 5.0V (<200mA)	
	VDD 3.3V (<300mA)	
	外供电源	

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, 16T	1.6	-	5.5	V
IDD1	动态电流 1	3V	高频运行（HIRC=16M） 低频运行（LIRC=64KHZ） FCPU=HIRC/16T	-	0.7	-	mA
		5V	全速工作	-	0.9	-	
ISP1	静态电流 1	3V	高频运行（HIRC=16M） 低频运行（LIRC=64KHZ）	-	80	-	uA
		5V	STOP=1 无唤醒源	-	135	-	
ISP2	静态电流 2	3V	高频停止 低频运行（LIRC=64KHZ）	-	2	-	uA
		5V	STOP=1 无唤醒源	-	7.5	-	
ISP3	静态电流 3	3V	高频停止 低频停止 STOP=1	-	-	0.5	uA
		5V	无唤醒源	-	-	0.7	
ISP4	静态电流 4	3V	高频停止 低频运行（LIRC=64KHZ） STOP=1	-	2	-	uA
		5V	WDT 唤醒（72ms）	-	7.2	-	
ISP5	静态电流 5	3V	高频停止 外部低频运行(LXT=32768HZ)	-	2.5	-	
		5V	STOP=1 无唤醒源	-	10.5	-	

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°C)				
V _{IL1}	输入低电平	3V	SMT	0	-	0.3VDD	V
V _{IH1}	输入高电平	3V		0.7VDD	-	VDD	
V _{IL2}	输入低电平	5V		0	-	0.3VDD	
V _{IH2}	输入高电平	5V		0.7VDD	-	VDD	
V _{IL3}	输入低电平	3V	低翻转	0	-	0.2VDD	
V _{IH3}	输入高电平	3V		0.3VDD	-	VDD	
V _{IL4}	输入低电平	5V		0	-	0.2VDD	
V _{IH4}	输入高电平	5V		0.2VDD	-	VDD	
R _{PH}	上拉电阻	5V	输入到GND	-	23	-	kΩ
		3V	输入到GND	-	43	-	
I _{OL1}	输出灌电流	5V	输出口, V _{out} =GND+0.6V		56		mA
		3V			31		
I _{OH1}	输出拉电流	5V	输出口, V _{out} =VDD-0.6V		20		
		3V			16		
I _{OL2}	输出灌电流	5V	输出口, V _{out} =GND+0.6V		6.5		
		3V			3.5		
I _{OH2}	输出拉电流	5V	输出口, V _{out} =VDD-0.6V		3.2		
		3V			2.5		
I _{OL3}	输出灌电流	5V	输出口, V _{out} =GND+0.6V		97		
		3V			55		

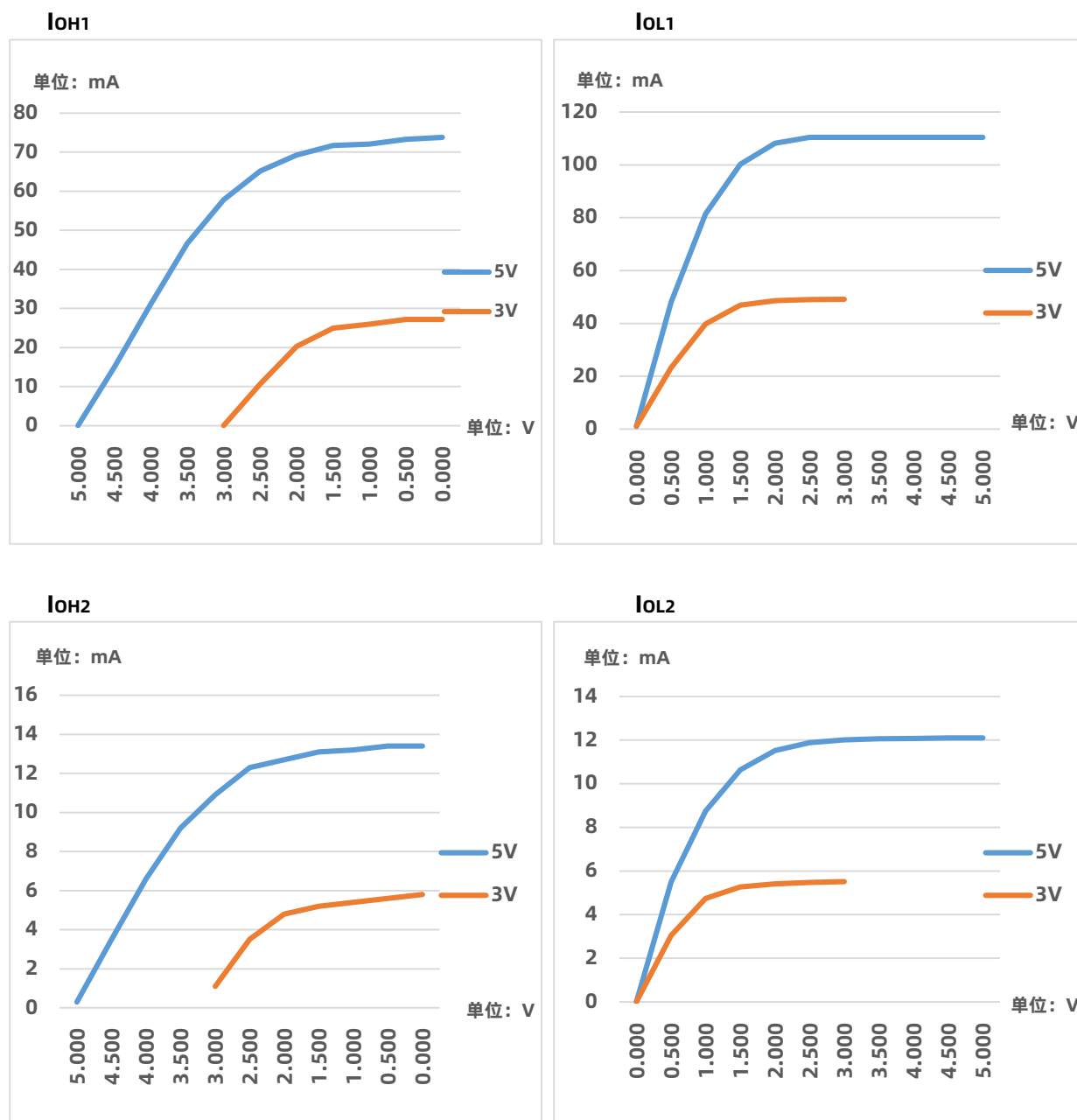
18.3 交流特性

常温(25°C)下，测试了 5 种不同等效串联电阻（ESR）类型的 32.768KHZ 晶振，在以下推荐的两种晶振配置字下所对应的不同负载电容的起振时间、过秒和电流数据：

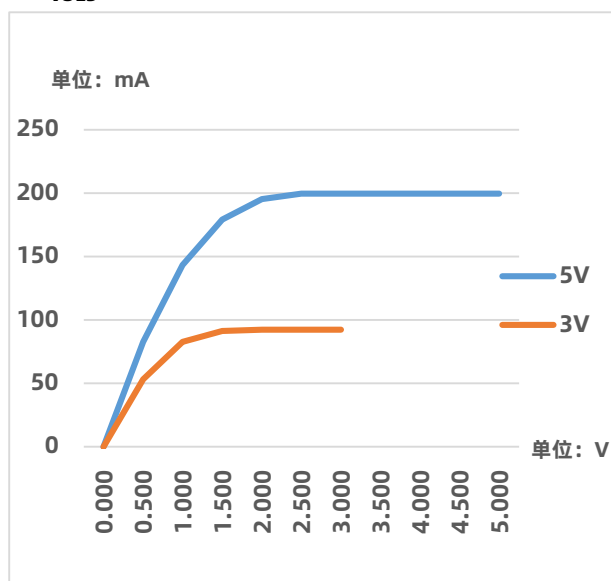
晶振配置字参数	测试条件			起振时间 (mS)	过秒 (M/S)	电流 (uA)
	等效串联电阻 ESR 类型(KΩ)	负载电容 (pF)	VDD			
驱动电流:驱动 1(强驱) 反馈电阻:电阻 0(小电阻) 供电选择:芯片 VDD 供电	30	18	5V	140	239	11.4
			3V	250	34	2.6
	40	18	5V	150	169	11.6
			3V	270	20	2.7
	50	18	5V	180	99	12.4
			3V	380	-3	2.9
	60	24	5V	170	125	11.8
			3V	300	26	3.1
	70	24	5V	160	122	11.4
			3V	320	18	3.1
驱动电流:驱动 1(强驱) 反馈电阻:电阻 0(小电阻) 供电选择:BIAS 供电 (此配置为默认配置字)	30	18	5V	530	26	7.8
			3V	930	15	5.6
	40	18	5V	750	13	7.8
			3V	1300	5	5.6
	50	18	5V	1100	-17	7.8
			3V	2000	-22	5.6
	60	24	5V	1100	-2	7.8
			3V	1600	-5	5.6
	70	24	5V	900	2	7.8
			3V	1400	-3	5.6

注：（1）晶振供电选择为 VDD 供电时电流会偏大，并且过秒会随电压变化而变化。
晶振供电选择为 BIAS 供电时电流会偏小，并且过秒不会随电压变化而变化。
（2）数据仅供参考，具体值不做设计保证。

18.4 IO 口拉灌电流特性曲线

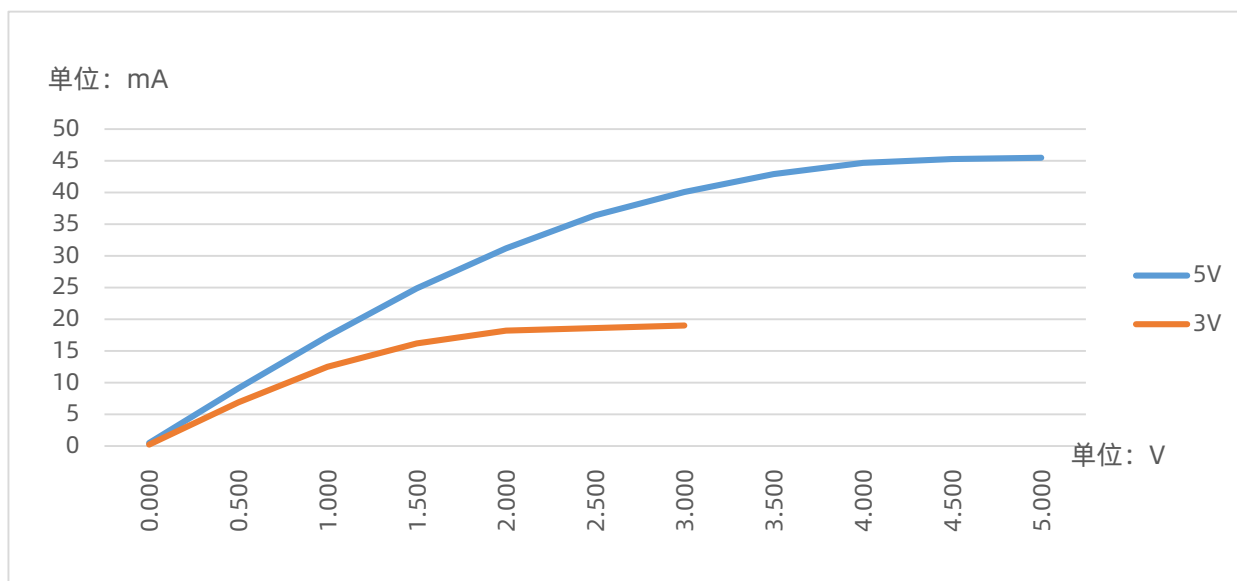


IoL3



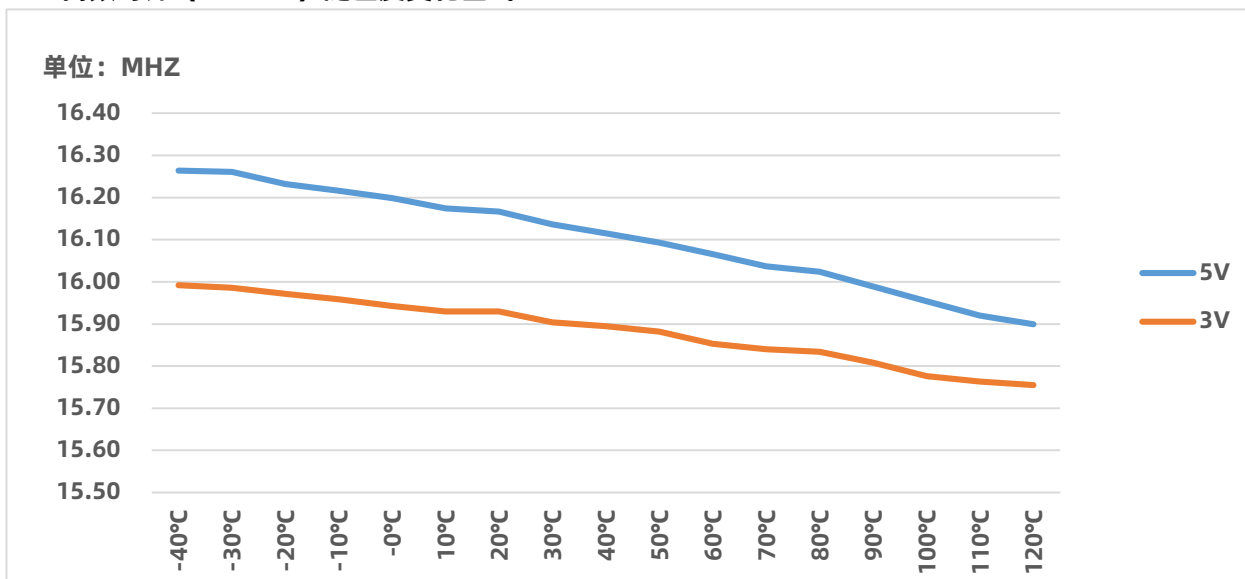
注: 具体值不做设计保证。

IOA7 (VPP □) 灌电流 IoL4

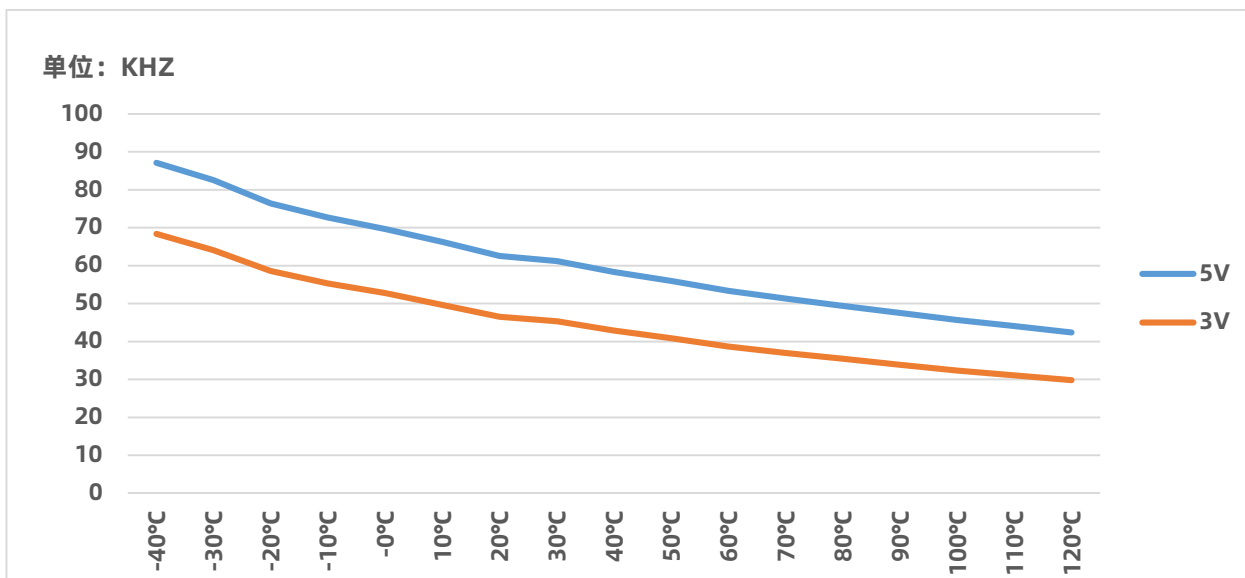


18.5 系统时钟特性

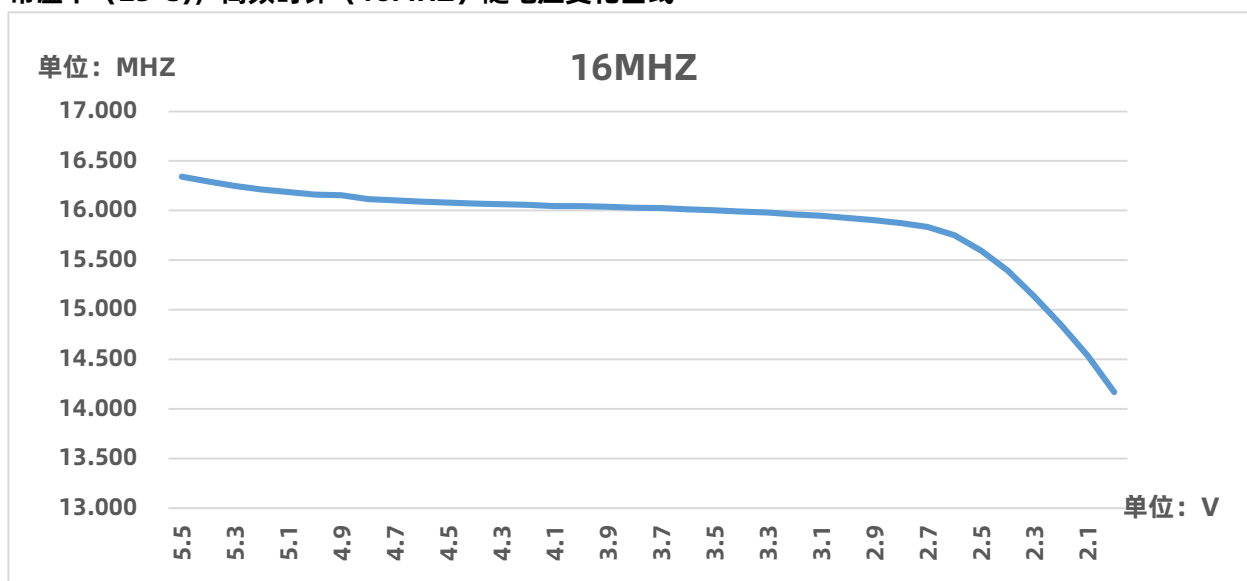
高频时钟（16MHZ）随温度变化曲线



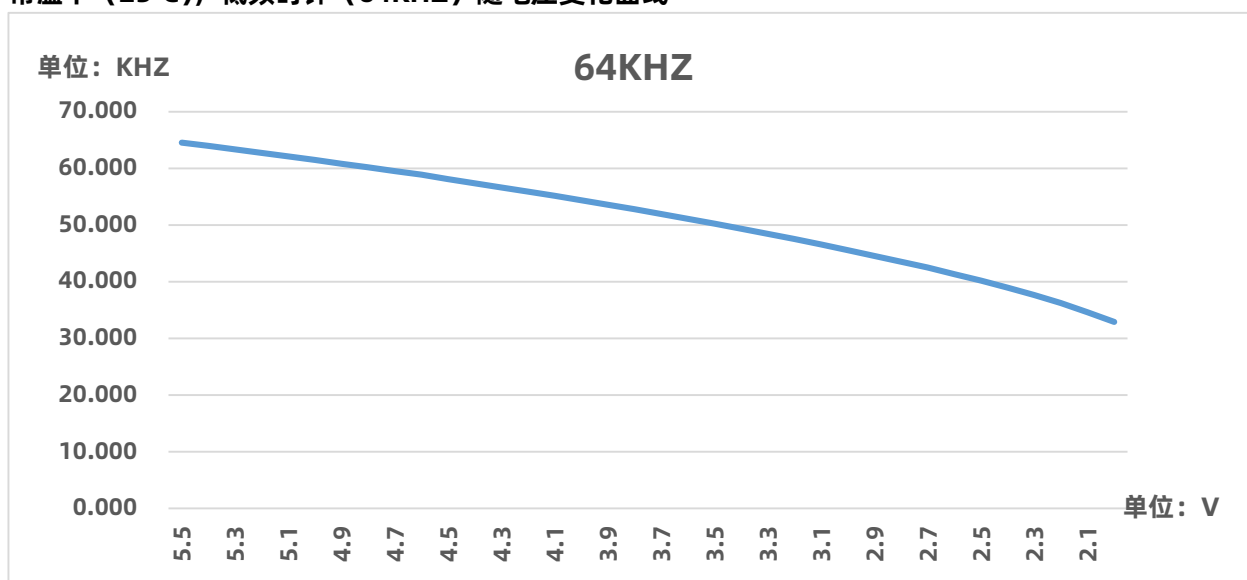
低频时钟（64KHZ）随温度变化曲线



常温下 (25℃), 高频时钟 (16MHZ) 随电压变化曲线



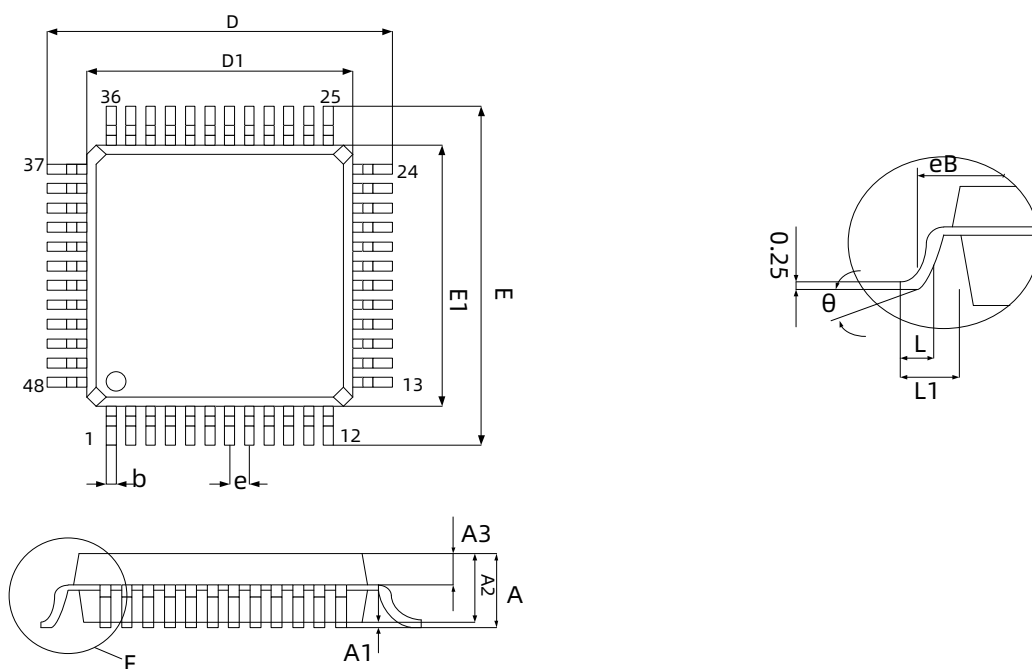
常温下 (25℃), 低频时钟 (64KHZ) 随电压变化曲线



注: 具体值不做设计保证。

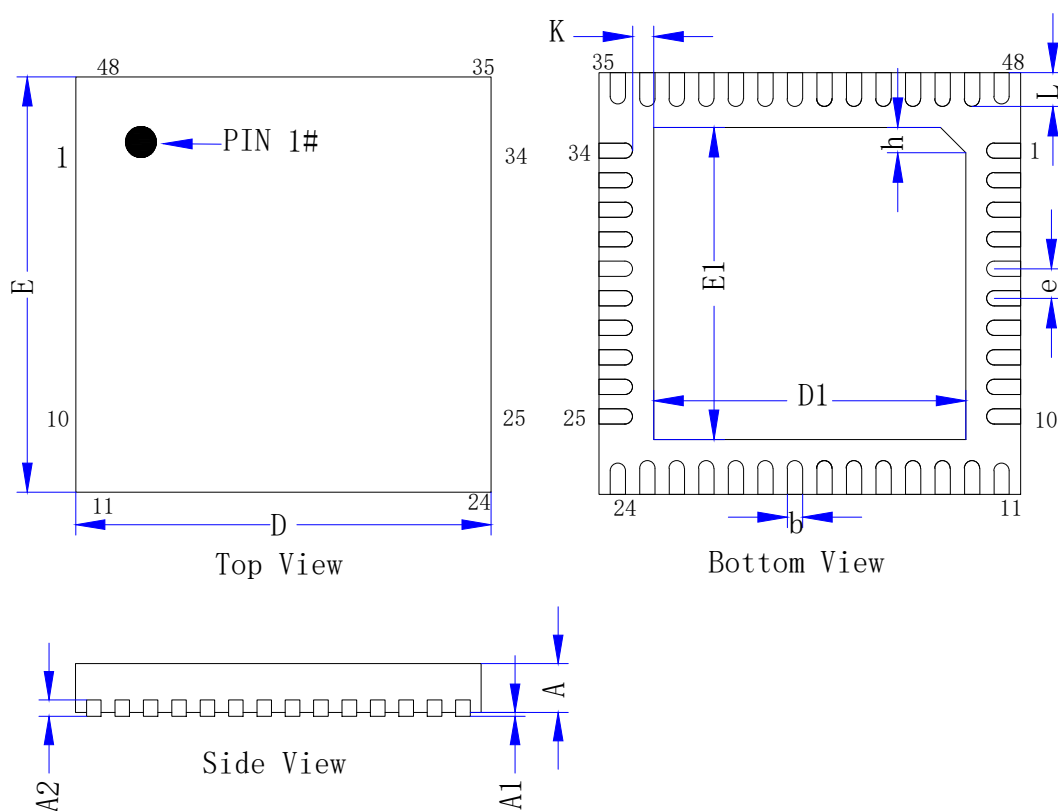
19 封装信息

19.1 LQFP7×7-48



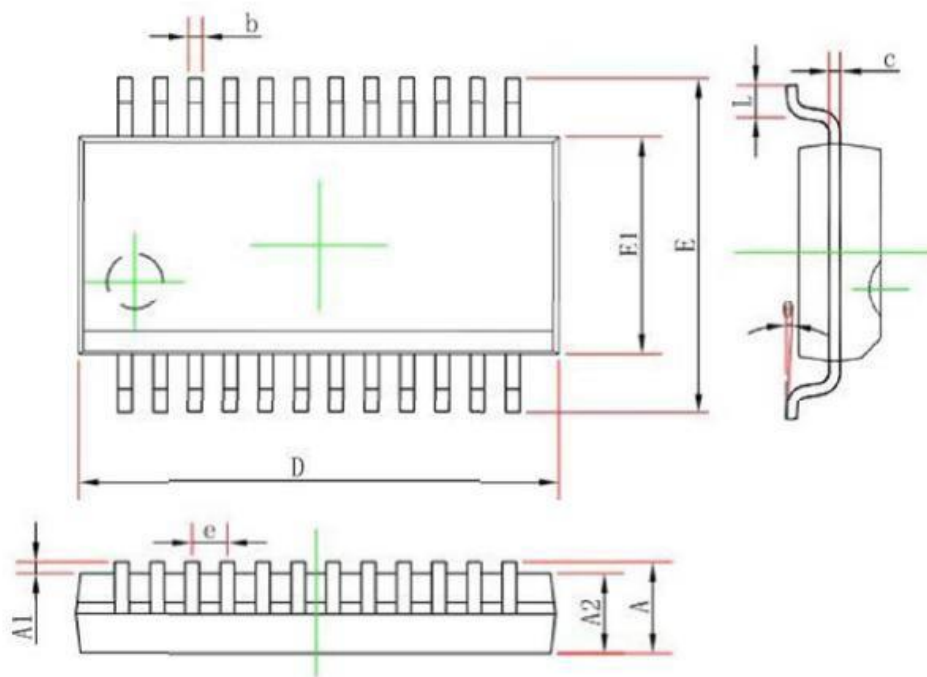
符号	单位 (mm)		
	最小	正常	最大
A	---	---	1.600
A1	0.050	---	0.150
A2	1.350	1.400	1.450
A3	0.600	---	0.640
b	0.180	---	0.260
c	0.130	---	0.170
D	8.800	9.000	9.200
D1	6.900	7.000	7.100
e	0.500BSC		
E	8.800	9.000	9.200
E1	6.900	7.000	7.100
L	0.400	---	0.650
L1	---	1.000REF	---
eB	8.100	---	8.250
θ	0°	---	8°

19.2 QFN5×5-48



SYMBOL	MILLIMETERS		
	MIN	NOM	MAX
A	0.55	0.60	0.65
A1	0.00	0.02	0.05
A2	0.203REF		
D	4.90	5.00	5.10
E	4.90	5.00	5.10
D1	3.65	3.70	3.75
E1	3.65	3.70	3.75
e	0.35BSC		
L	0.30	0.40	0.45
b	0.13	0.18	0.23
K	0.20	0.25	0.30
h	0.25	0.30	0.35

19.3 SSOP-24



Symbol	Dimensions In Millimeters	
	Min	Max
A	—	1.750
A1	0.050	0.080
A2	1.400	1.500
b	0.203	0.305
c	0.102	0.254
D	8.550	8.650
E1	3.800	4.000
E	5.800	6.200
e	0.635 (BSC)	
L	0.400	1.270
θ	0°	8°

20 指令集简述

20.1 概述

SUM08F041系列指令集是一种精简指令集（RISC），指令宽度为16位，由操作码和0~2个操作数组成。指令按照功能可分为5类，即字节操作指令、位操作指令、立即数指令、分支指令、特殊控制指令。

一个指令周期由1个系统时钟周期组成，除非条件测试结果为真或指令执行改变了程序计数器的值，否则执行所有的指令都只需要一个指令周期。对于上述两种特征情况，指令执行需要两个指令周期。任何一条指定文件寄存器作为指令一部分的指令都进行读-修改-写操作。读寄存器、修改数据并根据指令或目标标识符“d”存储结果。即使是写寄存器的指令也将先对改寄存器进行读操作。

20.2 符号说明

符号	范围	说明	符号	范围	说明
R/r	0-0x1ff	寄存器地址	C	-	进位标志
A	-	ACC 寄存器	DC	-	半进位标志
B/b	0-7	位地址	Z	-	零标志
I/i	0-0xff	立即数	d	0-1	目的操作数定义
K/k	0-0x1fff	标号	GIE	-	总中断使能位
TOS	-	栈顶	stkp	-	堆栈指针
PC	-	PC 指针			

20.3 指令集表

指令集表中, d=1, 目的操作数为 R; d=0, 目的操作数为 A

指令类型	助记符	指令说明	周期数	影响标志位	备注
寄存器操作指令	ADDAR R,d	$R+A \rightarrow d$	1	Z,DC,C	
	ADCAR R,d	$R+A+C \rightarrow d$	1	Z,DC,C	
	SUBAR R,d	$A-R \rightarrow d$	1	Z,DC,C	
	SBCAR R,d	$A-R-C \rightarrow d$	1	Z,DC,C	
	SUBRA R,d	$R-A \rightarrow d$	1	Z,DC,C	
	SBCRA R,d	$R-A-C \rightarrow d$	1	Z,DC,C	
	ANDAR R,d	$R \& A \rightarrow d$	1	Z	
	ORAR R,d	$R A \rightarrow d$	1	Z	
	XORAR R,d	$R \wedge A \rightarrow d$	1	Z	
	COMR R,d	$R \rightarrow d$	1	Z	
	MOVR R,d	$R \rightarrow d$	1	Z	
	MOVAR R	$A \rightarrow R$	1	-	
	CLRR R	$0 \rightarrow R$	1	Z	
	SWAPR R,d	R 半字节交换 $\rightarrow d$	1	-	
	RLR R,d	$R[7] \rightarrow C, \{R[6:0], C\} \rightarrow d$	1	C	
	RLRNC R,d	$\{R[6:0], 0\} \rightarrow d$	1	-	
	RRRR,d	$R[0] \rightarrow C, \{C, R[7:1]\} \rightarrow d$	1	C	
	RRRNC R,d	$\{0, R[7:1]\} \rightarrow d$	1	-	
	DECR R,d	$R-1 \rightarrow d$	1	Z	
	DJZR R,d	$R-1 \rightarrow d, \text{SKIP if } 0$	1(2)	-	
	INCR R,d	$R+1 \rightarrow d$	1	Z	
	JZR R,d	$R+1 \rightarrow d, \text{SKIP if } !0$	1(2)	-	
	JNZR R,d	$R+1 \rightarrow d, \text{SKIP if } !0$	1(2)	-	
	DJNZR R,d	$R-1 \rightarrow d, \text{SKIP if } !0$	1(2)	-	
	JCMPPAR R	SKIP if $A=R$	1(2)	Z,C	
	JNCMPAR R	SKIP if $A \neq R$	1(2)	Z,C	
	JGAR R	SKIP if $A \geq R$	1(2)	Z,C	
	JLAR R	SKIP if $A < R$	1(2)	Z,C	
	XCHAR R	$A \leftrightarrow R$	1	-	
位操作指令	JBTS0 R,b	SKIP if $R[b]=0$	1(2)	-	
	JBTS1 R,b	SKIP if $R[b]=1$	1(2)	-	
	BCLR R,b	$0 \rightarrow R[b]$	1	-	
	BSET R,b	$1 \rightarrow R[b]$	1	-	

指令类型	助记符	指令说明	周期数	影响标志位	备注
立即数操作指令	ADDIA I	I+A → A	1	Z,DC,C	
	ADCIA I	I+A+C → A	1	Z,DC,C	
	SUBIA I	I-A → A	1	Z,DC,C	
	SBCIA I	I-A-C → A	1	Z,DC,C	
	SUBAI I	A-I → A	1	Z,DC,C	
	SBCAI I	A-I-C → A	1	Z,DC,C	
	ANDIA I	A&I → A	1	Z	
	ORIA I	A I → A	1	Z	
	XORIA I	A^I → A	1	Z	
	MOVIA I	I → A	1	-	
	RETIA I	Stack → PC, I → A	2	-	
	JCMPAI I	SKIP if A=I	1(2)	Z,C	
	JNCPAII	SKIP if A≠I	1(2)	Z,C	
特殊操作指令	RLA	A[7] → C, {A[6:0],C} → A	1	C	
	RLANC	{A[6:0],0} → A	1	-	
	RRA	A[0] → C, {C,A[7:1]} → A	1	C	
	RRANC	{0,A[7:1]} → A	1	-	
	DECA	A-1 → A	1	Z	
	DJZA	A-1 → A, SKIP if 0	1(2)	-	
	INCA	A+1 → A	1	-	
	JZA	A+1 → A, SKIP if 0	1(2)	-	
	RETIE	Stack → PC, 1 → GIE	2	-	
	RETURN	Stack → PC	2	-	
	NOP	None Operation	1	-	
	RDT	ROM[{fsr1,fsr0}] → {HBUF, A}	3	-	
	DAA	加法后十进制调整	1	DC, C	
	DSA	减法后十进制调整	1	DC, C	
	PUSH	A, STATUS 压栈	1	-	
	POP	A, STATUS 出栈	1	Z, DC, C	
	CLRWDT	清除 WDT 寄存器	1	PD, TO	
分支指令	CALL I	I → PC, PC → Stack	2	-	
	GOTO I	I → PC	2	-	