带 LED 驱动功能 A/D 型触控键 Flash 单片机

CBM73xx 系列芯片手册

索引

特性

CPU 特性：

- 基于 8051 指令的 8 位单片机
- Flash ROM：8K~48K 字节
- RAM：内部 256 字节，外部 1024 字节
- 类 EEPROM：1024 字节
- 宽工作电压范围：
  -Fsys = 16MHZ，VDD = 2.7V~5.5V
- 振荡器：
  -内部高频振荡器：48MHZ(精度±2%)
  -内部低频振荡器：800KHZ(精度±2%)
- 看门狗定时器(WDT)，可软件使能
- CPU 主频：最大可以达到 16MHZ
- 内建的低电压检测功能(LVDC)
- 支持省电运行模式
  - 休眠模式
  - 空闲模式
- 可编程代码保护防止代码读出
- 支持串口在线调试

低功耗特性：

- 休眠电流：
  - 5.0V 时典型值为 200uA
- 工作电流：
  - 16MHZ、 5.0V 时典型值为 5.0mA

外设特性：

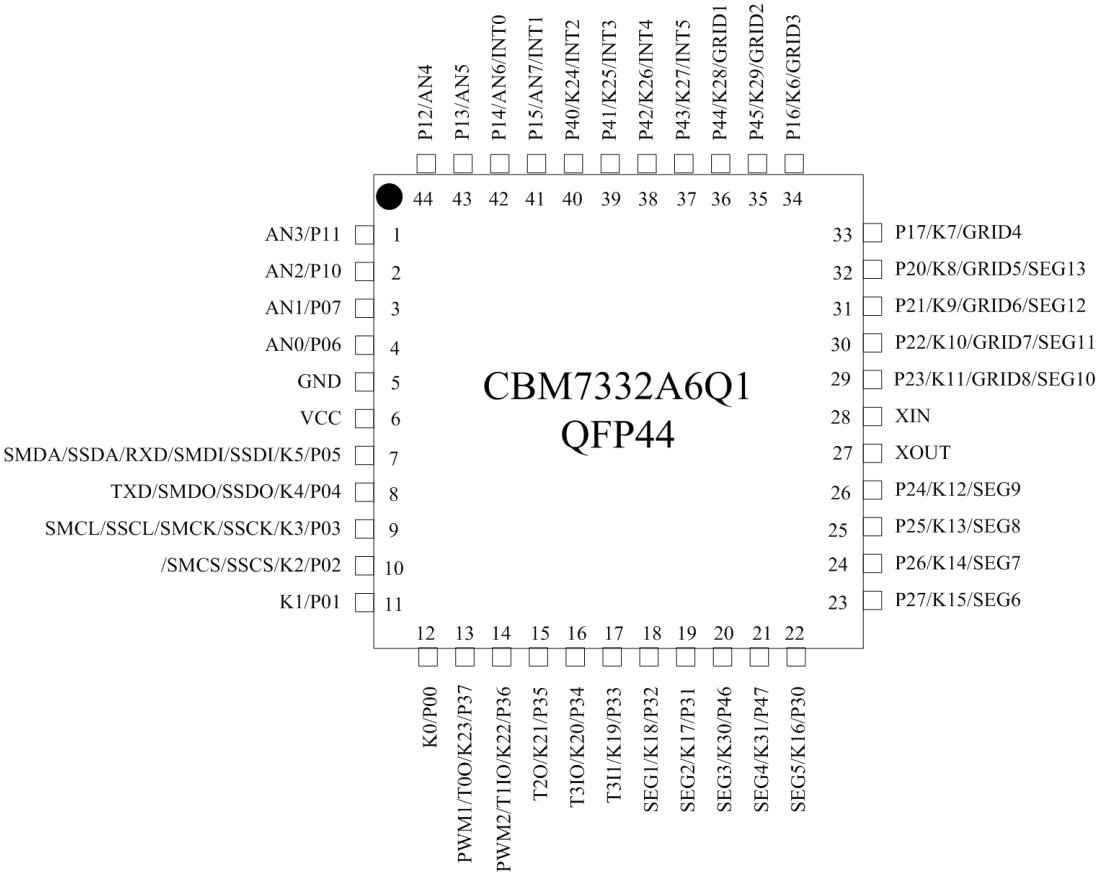
- 多达 40 个双向可单独控制的 I/O 引脚：
  - 高灌/拉电流可直接驱动 LED
  - 电平变化中断引脚(最大 6 路)
  - 可单独编程的弱上拉引脚
  - 可单独编程的弱下拉引脚
  - 可单独编程的开漏引脚
  - 超低功耗唤醒
- A/D 转换器：
  - 10-bit A/D 转换器
  - 最大支持 8 个通道
- 内建 12/16/24/32 路触控按键功能
- 多个定时器模块用于时间测量、捕捉输入、方波输出及单脉冲输出
- 支持多路 PWM 输出
  -最大 2 个 PWM 模块可提供 2 路 PWM 输出
  -最大 4 个定时器模块可提供 4 路 PWM 输出
- 通用 UART：
  - 独立的波特率发生器
  - 自动波特率校准
- 4 线 SPI(总共 4 种模式)
- I²C 主/从模式接口
- LED 驱动器：
  - 4×13 段/5×12 段/6×11 段/7×10 段/9×8 段/10×7 段驱动共阴/共阳 LED
- RTC：具有定时和闹钟功能的实时时钟(需外接 32.768KHZ 晶振)

选型表

芯片型号	程序存储器	数据存储器			I/O	10 位 A/D (通道数)	触摸按键	外部中断
	Flash (字节)	DATA (字节)	XRAM (字节)	类 EEPROM (字节)				
CBM7332A6Q1	48K	256	1024	1024	40	8	32	6
CBM7320A4S1	24K	256	1024	1024	26	6	20	2
CBM7320A4P1	24K	256	1024	1024	26	6	20	2
CBM7312A3S1	16K	256	1024	1024	18	4	14	1
CBM7308A3S1	16K	256	512	1024	14	2	12	1

芯片型号	UART	SPI	I2C	定时器	PWM	LED	RTC	封装
CBM7332A6Q1	Y	Y	Y	4	2	4×13	Y	QFP44
CBM7320A4S1	Y	Y	Y	4	2	4×13	N	SOP28
CBM7320A4P1	Y	Y	Y	4	2	4×13	N	SSOP28
CBM7312A3S1	Y	Y	Y	4	2	N	N	SOP20
CBM7308A3S1	Y	Y	Y	4	2	N	N	SOP16

引脚图—CBM7332A6Q1 (QFP44)：



注:除去VCC、GND、XIN、XOUT外的所有引脚都可以复用为：

SPI Slave：SSDI/SSDO/SSCK/SSCS；

SPI Master：SMDI/SMDO/SMCK/SMCS；

I2C Slave：SSCL/SSDA;

I2C Master：SMCL/SMDA;

uart：TXD/RXD;

timer：T0O/T1IO/T2O/T3IO/T3I1；

PWM：PWM1/PWM2；

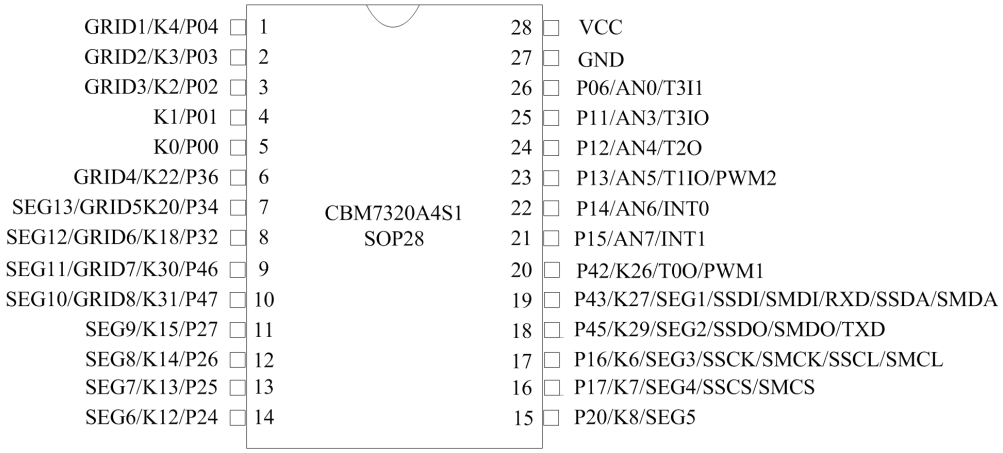
外部中断：**INT0/INT1/INT2/INT3/INT4/INT5；**

LED：SEG1/SEG2/SEG3/SEG4/SEG5/SEG6/SEG7/SEG8/SEG9/
SEG10_GRID8/SEG11_GRID7/SEG12_GRID6/SEG13_GRID5/
GRID4/GRID3/GRID2/GRID1。

但是**P00**、**P01**是调试串口，可用于**program**和**debug**，所以建议**P00**、**P01**只做**program**和**debug**使用。

具体方法请查看**6.2**小节“功能复用”。

引脚图二 CBM7320A4S1 (SOP28)：



注:除去**VCC**、**GND**外的所有引脚都可以复用为：

SPI Slave：SSDI/SSDO/SSCK/SSCS；

SPI Master：SMDI/SMDO/SMCK/SMCS；

I2C Slave：SSCL/SSDA;

I2C Master：SMCL/SMDA;

uart：TXD/RXD;

timer：T0O/T1IO/T2O/T3IO/T3I1；

PWM：PWM1/PWM2；

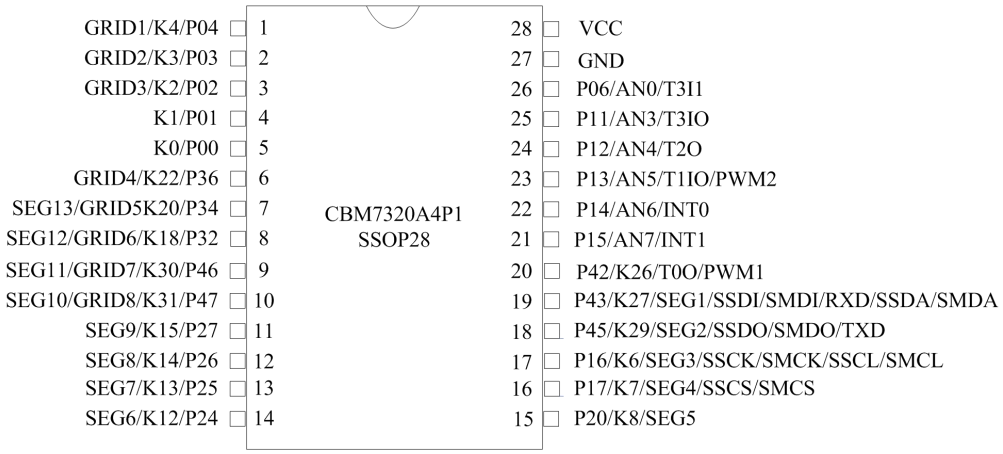
外部中断：**INT0/INT1/INT2/INT3/INT4/INT5；**

LED：SEG1/SEG2/SEG3/SEG4/SEG5/SEG6/SEG7/SEG8/SEG9/
SEG10_GRID8/SEG11_GRID7/SEG12_GRID6/SEG13_GRID5/
GRID4/GRID3/GRID2/GRID1。

但是**P00**、**P01**是调试串口，可用于**program**和**debug**，所以建议**P00**、**P01**只做**program**和**debug**使用。

具体方法请查看**6.2**小节“功能复用”。

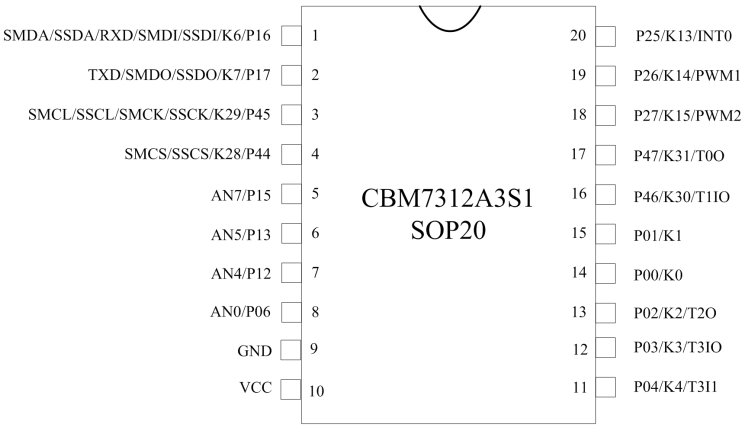
引脚图三 CBM7320A4P1 (SSOP28)：



注:除去VCC、GND外的所有引脚都可以复用为：

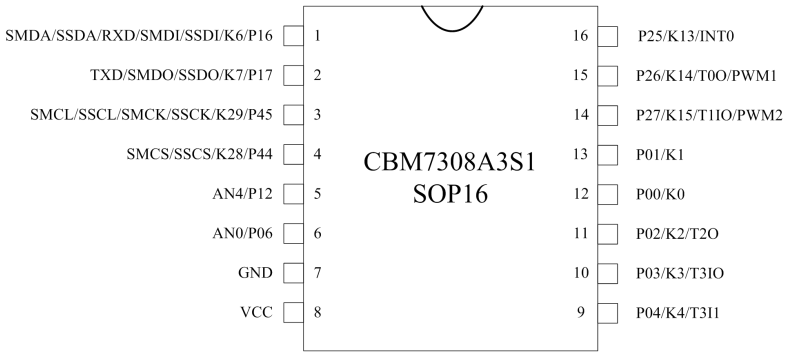
SPI Slave：SSDI/SSDO/SSCK/SSCS；
SPI Master：SMDI/SMDO/SMCK/SMCS；
I2C Slave：SSCL/SSDA;
I2C Master：SMCL/SMDA;
uart：TXD/RXD;
timer：T0O/T1IO/T2O/T3IO/T3I1；
PWM：PWM1/PWM2；
外部中断：**INT0/INT1/INT2/INT3/INT4/INT5；**
LED：SEG1/SEG2/SEG3/SEG4/SEG5/SEG6/SEG7/SEG8/SEG9/
SEG10_GRID8/SEG11_GRID7/SEG12_GRID6/SEG13_GRID5/
GRID4/GRID3/GRID2/GRID1。
但是**P00**、**P01**是调试串口，可用于**program**和**debug**，所以建议**P00**、**P01**只做**program**和**debug**使用。
具体方法请查看**6.2**小节“功能复用”。

引脚图四 CBM7312A3S1 (SOP20)：



注:除去VCC、GND外的所有引脚都可以复用为：
SPI Slave：SSDI/SSDO/SSCK/SSCS；
SPI Master：SMDI/SMDO/SMCK/SMCS；
I2C Slave：SSCL/SSDA;
I2C Master：SMCL/SMDA;
uart：TXD/RXD;
timer：T0O/T1IO/T2O/T3IO/T3I1；
PWM：PWM1/PWM2；
外部中断：**INT0/INT1/INT2/INT3/INT4/INT5；**
但是**P00**、**P01**是调试串口，可用于**program**和**debug**，所以建议**P00**、**P01**只做**program**和**debug**使用。
具体方法请查看**6.2**小节“功能复用”。

引脚图五 CBM7308A3S1 (SOP16)：



注:除去**VCC、GND**外的所有引脚都可以复用为：

SPI Slave：SSDI/SSDO/SSCK/SSCS；

SPI Master：SMDI/SMDO/SMCK/SMCS；

I2C Slave：SSCL/SSDA;

I2C Master：SMCL/SMDA;

uart：TXD/RXD;

timer：T0O/T1IO/T2O/T3IO/T3I1；

PWM：PWM1/PWM2；

外部中断：**INT0/INT1/INT2/INT3/INT4/INT5；**

但是**P00、P01**是调试串口，可用于**program**和**debug**，所以建议**P00、P01**只做**program**和**debug**使用。

具体方法请查看**6.2**小节“功能复用”。

表 1：引脚汇总

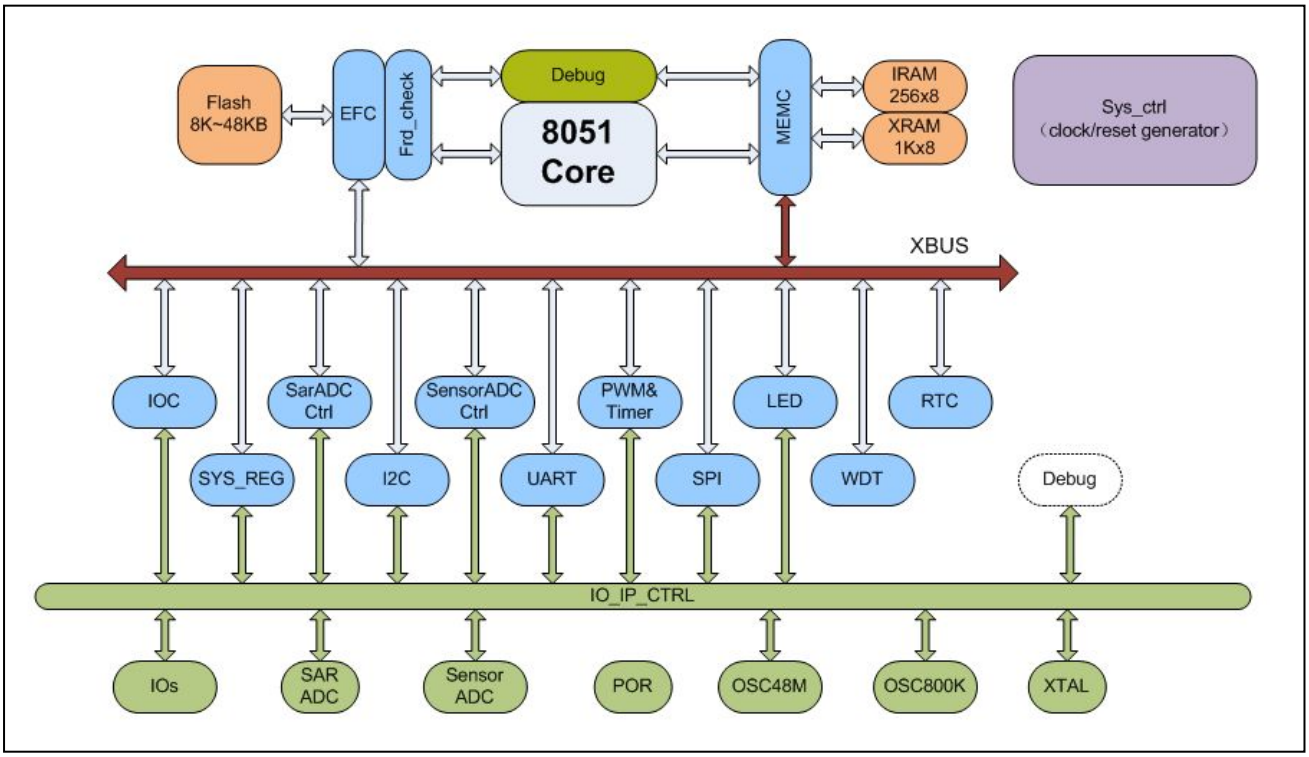
引脚名称	说明
VCC	电源输入管脚
GND	电源地
P00~P47	GPIO
K0~K31	触摸按键输入口
AN0~AN7	ADC 输入
XOUT	32.768K 晶振 XOUT
XIN	32.768K 晶振 XIN
SSDI	SPI Slave 数据输入口
SSDO	SPI Slave 数据输出口
SSCK	SPI Slave 时钟输入口
SSCS	SPI Slave 片选信号
SMDI	SPI Master 数据输入口
SMDO	SPI Master 数据输出口
SMCK	SPI Master 时钟输出口
SMCS	SPI Master 片选信号
SSCL	I^2C Slave 时钟口
SSDA	I^2C Slave 数据口
SMCL	I^2C Master 时钟口
SMDA	I^2C Master 数据口
TXD	通用串口输出口
RXD	通用串口输入口
T0O	定时器 0 输出口
T1IO	定时器 1 输入/输出口
T2O	定时器 2 输出口
T3IO	定时器 3 输入 0/输出口
T3I1	定时器 3 输入 1
PWM1	PWM1 输出口
PWM2	PWM2 输出口
INT0~INT5	外部中断 0~5 输入口
SEG1	LED SEG 口
SEG2	LED SEG 口
SEG3	LED SEG 口
SEG4	LED SEG 口
SEG5	LED SEG 口
SEG6	LED SEG 口
SEG7	LED SEG 口
SEG8	LED SEG 口
SEG9	LED SEG 口
SEG10_GRID8	LED SEG/GRID 口
SEG11_GRID7	LED SEG/GRID 口

SEG12_GRID6	LED SEG/GRID 口
SEG13_GRID5	LED SEG/GRID 口
GRID4	LED GRID 口
GRID3	LED GRID 口
GRID2	LED GRID 口
GRID1	LED GRID 口

1. 概述

本用户手册涵盖了 CBM73xx 系列所有芯片。图 1-1 给出了 CBM73xx 芯片的框图。表 1-1 给出了相应的引脚配置说明。

图 1-1：CBM73XX 系统框图



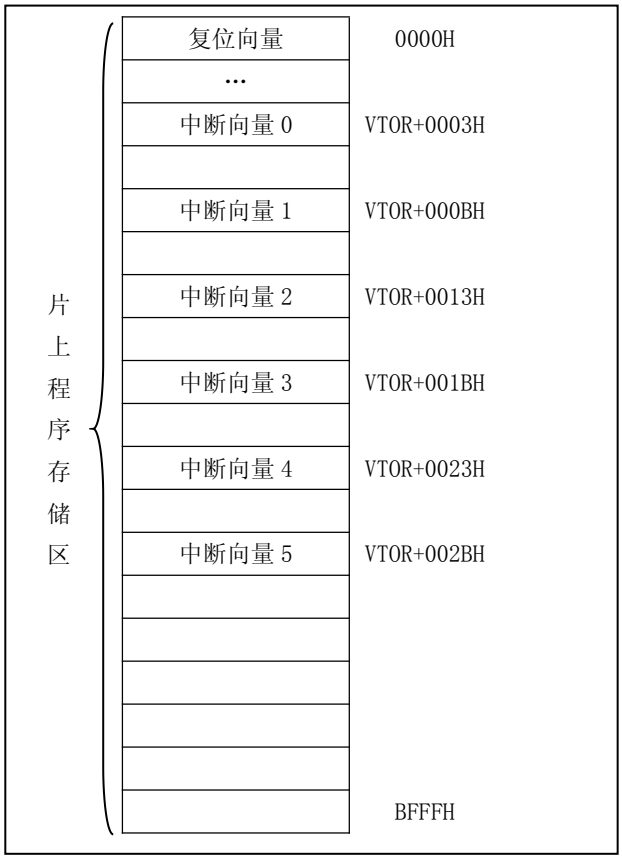
2. 存储器的构成

2.1. 程序存储器的构成

CBM73xx 系列芯片具备 16 位的程序计数器(program counter，PC)，支持寻址 CBM73xx 最大 48K × 8bit(0000H－BFFFH)的程序存储空间。不能访问超出以上边界的单元。复位向量位于 0000H 处，有 6 个中断向量默认分别位于 0003H、000BH、0013H、001BH、0023H、002BH 处(可以配置 VTOR 寄存器，使中断向量产生偏移，详见 3.1.5 小节“中断向量偏移地址寄存器”)。

CBM7332A6Q1 芯片的程序存储器映射如图 2-1 所示。

图 2-1： CBM7332A6Q1 的程序存储器映射



2.2. 数据存储器的构成

数据存储器分为 8051 内核内置片内 128 字节直接寻址区、高 128 字节间接寻址区与高 128 地址的直接寻址特殊功能寄存器、扩展的 1024 字节外部 RAM 区及片外特殊功能寄存器。

2.2.1. 片外 XRAM

CBM73xx 系列芯片除了片内 256 字节 RAM，还在内部集成了 1024 字节片外 XRAM，可供用户用于数据存储。

2.2.2. 片内 RAM

片内 256 字节 RAM 具体分为：

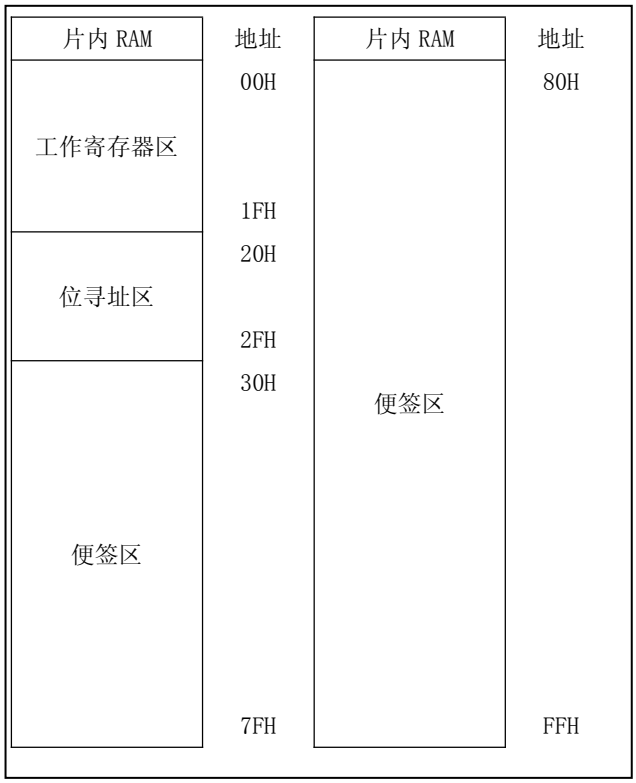
1.工作寄存器区：共 4 个组，每组为 8 个存储单元，即 00H-07H、08H-0FH、10H-17H、18H-1FH，具体选择哪一个由程序状态字(PSW)中的 RS1 位和 RS0 位的组合决定。

2.位寻址区：从 20H-2FH，共 16 个单元，每一位都可以进行位寻址(16*8=128 个位地址)。

3.便笺区：从 30H-FFH，共 208 个单元，作为数据区或堆栈区使用。

CBM73xx 芯片的片内 RAM 映射如图 2-2 所示。

图 2-2： CBM73xx 片内 RAM 映射



3. 特殊功能寄存器(SFR)

特殊功能寄存器是CPU和外设模块用来控制芯片进行所需操作的寄存器。

可分成两类：外设特殊功能寄存器(见图 3-1)和内核特殊功能寄存器(见图 3-2)。内核特殊功能寄存器与片内 RAM 共用 80H-FFH 地址，但是它们拥有不同的物理存储区，使用不同的指令来访问。与外设功能相关的寄存器在相应外设功能的章节介绍，本节将介绍与内核功能相关的寄存器。

图 3-1： CBM7332A6Q1 外设特殊功能寄存器

片外 XRAM	地址	片外 XRAM	地址	片外 XRAM	地址	片外 XRAM	地址
IO 控制		P37IOCFG	101FH	P4PURSELRL	38FFH	LEDDR8	1108H
P00IOCFG	1000H	P40IOCFG	1020H	P4PURSELRH	39FFH	LEDDR9	1109H
P01IOCFG	1001H	P41IOCFG	1021H	P0OD	3AFFH	LEDDR10	110AH
P02IOCFG	1002H	P42IOCFG	1022H	P1OD	3BFFH	LEDDR11	110BH
P03IOCFG	1003H	P43IOCFG	1023H	P2OD	3CFFH	LEDDR12	110CH
P04IOCFG	1004H	P44IOCFG	1024H	P3OD	3DFFH	LEDDR13	110DH
P05IOCFG	1005H	P45IOCFG	1025H	P4OD	3EFFH		
P06IOCFG	1006H	P46IOCFG	1026H	P0LDRVRL	40FFH	**系统相关**	
P07IOCFG	1007H	P47IOCFG	1027H	P0LDRVRH	41FFH	CKDIVCR	50FFH
P10IOCFG	1008H	P0PD	20FFH	P1LDRVRL	42FFH	MDLCKCR	51FFH
P11IOCFG	1009H	P1PD	21FFH	P1LDRVRH	43FFH	SLPWKCR	52FFH
P12IOCFG	100AH	P2PD	22FFH	P2LDRVRL	44FFH	SLPWKDR	53FFH
P13IOCFG	100BH	P3PD	23FFH	P2LDRVRH	45FFH	RSTSR	54FFH
P14IOCFG	100CH	P4PD	24FFH	P3LDRVRL	46FFH	LVDCR	55FFH
P15IOCFG	100DH	P0PU	25FFH	P3LDRVRH	47FFH	MISCR	56FFH
P16IOCFG	100EH	P1PU	26FFH	P4LDRVRL	48FFH	LDOSETR	5FFFH
P17IOCFG	100FH	P2PU	27FFH	P4LDRVRH	49FFH		
P20IOCFG	1010H	P3PU	28FFH			**WDT**	
P21IOCFG	1011H	P4PU	29FFH	**LED**		WDTCR	60FFH
P22IOCFG	1012H	P0OE	2AFFH	LEDCR	C0FFH	WDTSETR	61FFH
P23IOCFG	1013H	P1OE	2BFFH	LEDSETR	C1FFH	WDTSR	62FFH
P24IOCFG	1014H	P2OE	2CFFH	LEDDIVR	C2FFH		
P25IOCFG	1015H	P3OE	2DFFH	GRIDVLDL	C3FFH	**PWM**	
P26IOCFG	1016H	P4OE	2EFFH	GRIDVLDH	C4FFH	PWM1CR	68FFH
P27IOCFG	1017H	P0PURSELRL	30FFH	LEDDR0	1100H	PWM2CR	69FFH
P30IOCFG	1018H	P0PURSELRH	31FFH	LEDDR1	1101H	PWM1LRL	6AFFH
P31IOCFG	1019H	P1PURSELRL	32FFH	LEDDR2	1102H	PWM1LRH	6BFFH
P32IOCFG	101AH	P1PURSELRH	33FFH	LEDDR3	1103H	PWM1HRL	6CFFH
P33IOCFG	101BH	P2PURSELRL	34FFH	LEDDR4	1104H	PWM1HRH	6DFFH
P34IOCFG	101CH	P2PURSELRH	35FFH	LEDDR5	1105H	PWM2LR	6EFFH
P35IOCFG	101DH	P3PURSELRL	36FFH	LEDDR6	1106H	PWM2HR	6FFFH
P36IOCFG	101EH	P3PURSELRH	37FFH	LEDDR7	1107H		

图 3-1： CBM73xx 外设特殊功能寄存器(续)

片外 XRAM	地址	片外 XRAM	地址	片外 XRAM	地址	片外 XRAM	地址
定时器		SCR2	92FFH	I2CMTRDR	CBFFH	**RTC**	
T0CKCR	70FFH	SCR3	93FFH	I2CMCTR	CCFFH	RTCCR	E8FFH
T0CR	71FFH	SSCAPL	94FFH	I2CMSR	CDFFH	RTCSR	E9FFH
T0CMP0	72FFH	SSCAPH	95FFH	I2CSCR	D8FFH	RTCHDR	EAFFH
T0CMP1	73FFH	SSR	96FFH	I2CSSR	D9FFH	RTCMDR	EBFFH
T0SR	74FFH	SDATL	97FFH	I2CSDRSEL	DAFFH	RTCSDR	ECFFH
T1CKCR	78FFH	SDATH	98FFH	I2CSDR	DBFFH	RTCAHDR	EDFFH
T1CR	79FFH	SSELR	9AFFH	I2CSADDR	DCFFH	RTCAMDR	EEFFH
T1CMPR	7AFFH	SAWKCR	9EFFH	I2CSRADR	DDFFH	RTCASDR	EFFFH
T1SR	7BFFH			I2CSHADR	DEFFH		
T2CKCR	80FFH	**EFC(类 EEPROM)**				**中断**	
T2CR	81FFH	FDR	B0FFH	**UART**		IFLTCNTR	F0FFH
T2CMPR0L	82FFH	FADRL	B1FFH	UCR	D0FFH	XINT01TRGR	F1FFH
T2CMPR0H	83FFH	FADRH	B2FFH	USETR	D1FFH	XINT23TRGR	F2FFH
T2CMPR1L	84FFH	FCR	B3FFH	UBPRL	D2FFH	XINT45TRGR	F3FFH
T2CMPR1H	85FFH	FPSR	B4FFH	UBPRH	D3FFH	IRQ0SR	F4FFH
T2SR	86FFH	FDIV	B5FFH	UIER	D4FFH	IRQ1SR	F5FFH
T3CR	88FFH	ADC		UTXR	D5FFH	IRQ2SR	F6FFH
T3CKCR	89FFH	ADCDIVCR	B8FFH	URXR	D6FFH	IRQ3SR	F7FFH
T3OCR	8AFFH	ADCSELR	B9FFH	USR	D7FFH	IRQ4SR	F8FFH
T3CMPR0L	8BFFH	ADCSR	BAFFH			IRQ5SR	F9FFH
T3CMPR0H	8CFFH	ADCCR	BBFFH	**SPI**		IRQ0ER	FAFFH
T3CMPR1L	8DFFH	ADCDRL	BCFFH	SPICR	E0FFH	IRQ1ER	FBFFH
T3CMPR1H	8EFFH	ADCDRH	BDFFH	SPISFMR	E1FFH	IRQ2ER	FCFFH
T3SR	8FFFH			SPIIER	E2FFH	IRQ3ER	FDFFH
		I2C		SPIBPR	E3FFH	IRQ4ER	FEFFH
SensorADC		I2CMPRERL	C8FFH	SPISR	E4FFH	IRQ5ER	FFFFH
SCR0	90FFH	I2CMPRERH	C9FFH	SPIDR	E5FFH		
SCR1	91FFH	I2CMCR	CAFFH				

3.1. 内核特殊功能寄存器

CBM73xx 芯片有 16 字节内核特殊功能寄存器，内核特殊功能寄存器与片内 RAM 共用 80H-FFH 地址，但是它们拥有不同的物理存储区，使用不同的指令来访问。

图 3-2： CBM73xx 内核特殊功能寄存器

内核特殊功能寄存器	地址
P0	80H
SP	81H
DPL	82H
DPH	83H
PSEQ	86H
PCON	87H
VTOR	88H
P1	90H
P2	A0H
IE	A8H
P3	B0H
IP	B8H
P4	C0H
PSW	D0H
ACC	E0H
B	F0H
	FFH

3.1.1. 累加器(ACC)

累加器 ACC 是一个常用的专用寄存器，指令系统中采用 A 作为累加器的助记符。

3.1.2. B 寄存器(B)

在乘除法指令中，会用到 B 寄存器。在其他指令中，B 寄存器可作为暂存器来使用。

3.1.3. 栈指针(SP)

栈指针 SP 是一个 8 位专用寄存器，在执行 PUSH、各种子程序调用、中断响应等指令时，SP 先加一，再将数据压栈；执行 POP、RET、RETI 等指令时，数据退出堆栈后 SP 再减一。堆栈栈顶可以是片上内部 RAM(30H-FFH)的任意地址，系统复位后，SP 初始化为 07H，使得堆栈实际上从 08H 地址开始。

3.1.4. 数据指针(DPTR)

数据指针是一个 16 位专用寄存器。编程时，DPTR 既可以作为一个 16 位寄存器使用，也可以作为 2 个独立的 8 位寄存器 DPH 和 DPL 分开使用，即：

DPH　DPTR 高 8 位字节

DPL　DPTR 低 8 位字节

DPTR 通常在访问外部数据存储器时作地址指针使用。由于外部数据存储器的寻址范围为 64KB，故把 DPTR 设计为 16 位。

3.1.5. 中断向量偏移地址寄存器(VTOR)

VOTR 寄存器保存的是中断向量的偏移地址。设置该寄存器后，当中断发生时，程序跳转到新的地址上去执行。(详细说明见 5.2.2 小节“中断入口”)

当设置编译器，使中断服务程序的入口不在程序空间的开始处(0x0000)时，可以选择设置此寄存器。

VTOR 寄存器上电复位值为 0x00，即复位时中断入口地址和 8051 中断入口地址一致。若用户开发的程序的起始地址为 0x0800，则可以将中断入口地址的高 8 位设置到中断入口偏移地址寄存器 VTOR 中。

3.1.6. 程序状态字(PSW)

程序状态字是一个 8 位寄存器，用于存放程序运行中的各种状态信息。

其中有些位的状态是根据程序执行结果，由硬件自动设置的，而有些位的状态则使用软件方法设定。PSW 的位状态可以用专门指令进行测试，也可以用指令读出。一些条件转移指令根据 PSW 某些位的状态进行程序转移。

PSW 寄存器如寄存器 3-1 所示，包含：

- 算术状态标志位
- 用户自定义标志位
- 工作寄存器组页选择位
- 奇偶校验位

寄存器 2-1： PSW：程序状态字寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CY	AC	F0	RS1	RS0	OV	F1	P
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 CY:进位标志位
0 = 算术或逻辑运算中，没有进位或借位发生
1 = 算术或逻辑运算中，有进位或借位发生

bit 6 AC：辅助进位标志位
0 = 算术或逻辑运算中，没有辅助进位或借位发生
1 = 算术或逻辑运算中，有辅助进位或借位发生

bit 5 F0:用户自定义标志位
提供给用户自定义的标志位

bit 4-3 RS<1:0>：R0-R7 工作寄存器页选择位
00 = 页 0(映射到 00H-07H)
01 = 页 1(映射到 08H-0FH)
10 = 页 2(映射到 10H-17H)
11 = 页 2(映射到 18H-1FH)

bit 2 OV：溢出标志位
0 = 没有溢出发生
1 = 有溢出发生

bit 1 F1:用户自定义标志位
提供给用户自定义的标志位

bit 0 P：奇偶校验位
0 = 累加器 A 中值为 1 的位数为偶数
1 = 累加器 A 中值为 1 的位数为奇数

3.1.7. 中断使能寄存器(IE)

如寄存器 2-2 所示，IE 是中断使能寄存器，包含中断 0、中断 1、…、中断 5 的使能位以及全局中断使能位。

注： 当有中断条件产生时，无论对应的中断使能位或全局使能位(IE 中的 EA)的状态如何，中断标志位都将置 1。用户软件应在使能一个中断之前，确保先将相应的中断标志位清零。

寄存器 2-2： IE：中断使能寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
EA	—	EX5	EX4	EX3	EX2	EX1	EX0
R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 EA：全局中断使能位
0 = 禁止所有中断
1 = 使能所有未被屏蔽的中断

bit 6 未实现：读为 0

bit 5 EX5：中断 5 使能位
0 = 禁止中断 5 中断
1 = 使能中断 5 中断

bit 4 EX4：中断 4 使能位
0 = 禁止中断 4 中断
1 = 使能中断 4 中断

bit 3 EX3：中断 3 使能位
0 = 禁止中断 3 中断
1 = 使能中断 3 中断

bit 2 EX2：中断 2 使能位
0 = 禁止中断 2 中断
1 = 使能中断 2 中断

bit 1 EX1：中断 1 使能位
0 = 禁止中断 1 中断
1 = 使能中断 1 中断

bit 0 EX0：中断 0 使能位
0 = 禁止中断 0 中断
1 = 使能中断 0 中断

3.1.8. 中断优先级控制寄存器(IP)

CBM73xx 中断优先级有两级，IP 寄存器控制 6 个中断的优先级，如寄存器 2-3 所示。

寄存器 2-3： IP：中断优先级控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	PX5	PX4	PX3	PX2	PX1	PX0
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未实现：读为 0

bit 5 PX5：中断 5 优先级控制位
0 = 中断 5 低优先级
1 = 中断 5 高优先级

bit 4 PX4：中断 4 优先级控制位
0 = 中断 4 低优先级
1 = 中断 4 高优先级

bit 3 PX3：中断 3 优先级控制位
0 = 中断 3 低优先级
1 = 中断 3 高优先级

bit 2 PX2：中断 2 优先级控制位
0 = 中断 2 低优先级
1 = 中断 2 高优先级

bit 1 PX1：中断 1 优先级控制位
0 = 中断 1 低优先级
1 = 中断 1 高优先级

bit 0 PX0：中断 0 优先级控制位
0 = 中断 0 低优先级
1 = 中断 0 高优先级

注： 1、高优先级的中断可以打断低优先级中断的处理，反之，不行；
2、同一级的中断，后产生的中断不能打断目前正在处理的中断；
3、同一级的中断也有优先级，PX0 > PX1 > … > PX5；
4、若同时产生两个中断，且是同一级，则高优先级的先处理；

3.1.9. 电源控制寄存器(PCON)

电源控制(PCON)寄存器包含：

- 休眠模式使能位
- 空闲模式使能位

寄存器 2-4： PCON：电源控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	—	SLEEP	IDLE
U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-2 未实现：读为 0

bit 1 SLEEP：SLEEP 模式使能位
0 = 禁止 SLEEP 模式
1 = 使能 SLEEP 模式

bit 0 IDLE：IDLE 模式使能位
0 = 禁止 IDLE 模式
1 = 使能 IDLE 模式

注： 1、芯片处于调试模式下，不能通过调试行为改变该寄存器的值，仅可以通过软件代码改变。
2、写 PCON 寄存器前，必需先依次写 PSEQ = 0x0A ，PSEQ = 0x05。
3、不能同时使能 SLEEP 与 IDLE 模式，即不能写 PCON=0x03。

3.1.10. 写 PCON 前的保护序列寄存器(PSEQ)

PSEQ 是操作 PCON 寄存器前的保护序列寄存器。

寄存器 2-5： PSEQ：写 PCON 前的保护序列寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—				
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未实现：读为 0

bit 3-0 PSEQ<3:0>：写 PCON 前的保护序列寄存器
必需先依次写 PSEQ<3:0> = 0x0A、PSEQ<3:0> = 0x05，才能写 PCON 寄存器。

注： 芯片处于调试模式下，不能通过调试行为改变该寄存器的值，仅可以通过软件代码改变。

4. 振荡器模块

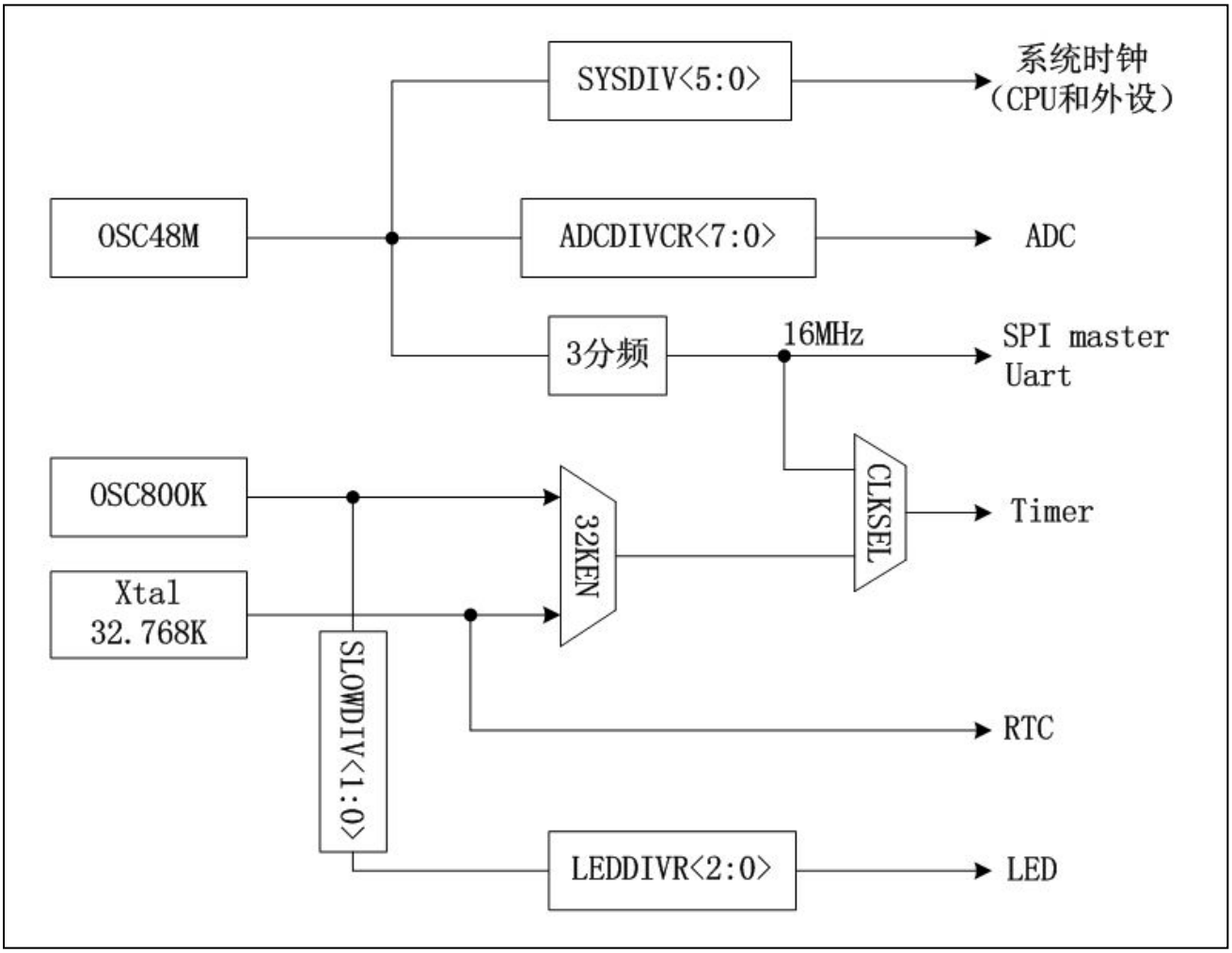
振荡器模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
CKDIVCR	0x50FF	XRAM	时钟分频控制寄存器	1100 0011
MDLCKCR	0x51FF	XRAM	各个功能模块时钟控制寄存器	0000 1000

4.1. 概述

振荡器模块具有多种时钟源,产生不同速度的时钟供给 CPU 和外设。图 4-1 说明了振荡器模块的框图。

图 4-1： MCU 时钟源简化框图



4.2. 振荡器控制

振荡器模块有内部高频 48M、内部低速 800K 或外部 32.768K 3 种时钟。CKDIVCR 寄存器的 SYSDIV<5:0>位可以对内部 48M 高频时钟进行分频，SLOWDIV<1:0>位可以对内部 800K 低速时钟进行分频；再提供给 CPU 及外设使用。

MDLCKCR 寄存器是控制各个功能模块的时钟使能的寄存器。

上电时，系统时钟默认为内部高频 48MHZ 的 4 分频，即 12MHZ。系统最高工作频率为 16MHZ，即内部高频 48MHZ 的 3 分频。所以配置 SYSDIV<5:0>时应注意值不能小于 2。

寄存器 4-1： CKDIVCR：时钟分频控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SLOWDIV1	SLOWDIV0	SYSDIV5	SYSDIV4	SYSDIV3	SYSDIV2	SYSDIV1	SYSDIV0
R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 SLOWDIV<1:0>：内部低速时钟分频控制位

00 = 1 分频

01 = 2 分频

10 = 4 分频

11 = 8 分频

bit 5-0 SYSDIV<5:0>：内部高速时钟分频控制位(供给系统时钟)

000000 = 1 分频

000001 = 2 分频

000010 = 3 分频

000011 = 4 分频

000100 = 5 分频

000101 = 6 分频

000110 = 7 分频

000111 = 8 分频

……

111000 = 57 分频

111001 = 58 分频

111010 = 59 分频

111011 = 60 分频

111100 = 61 分频

111101 = 62 分频

111110 = 63 分频

111111 = 64 分频

注： 1、系统时钟 = OSC48M /(SYSDIV<5:0> + 1)

2、由于 CPU 最快只能运行在 16M 时钟下，所以 SYSDIV<5:0>不能小于 000010(3 分频)。

寄存器 4-2： MDLCKCR：模块时钟控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
32KEN	PWMEN	SPIEN	UARTEN	IOCEN	TIMEREN	LEDEN	WDTEN
R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 32KEN：Timer 模块的低速时钟选择位
0 = Timer 模块的低速时钟切换到内部 800K
1 = Timer 模块的低速时钟切换到外部 32.768K 晶振(必须外接 32.768K 晶振)

bit 6 PWMEN：PWM 时钟使能位
0 = 关闭 PWM 时钟，PWM 不工作
1 = 使能 PWM 时钟，PWM 工作

bit 5 SPIEN：SPI 时钟使能位
0 = 关闭 SPI 时钟，SPI 不工作
1 = 使能 SPI 时钟，SPI 工作

bit 4 UARTEN：UART 时钟使能位
0 = 关闭 UART 时钟，UART 不工作
1 = 使能 UART 时钟，UART 工作

bit 3 IOCEN：GPIO 时钟使能位
0 = 关闭 GPIO 时钟，GPIO 不工作
1 = 使能 GPIO 时钟，GPIO 工作

bit 2 TIMEREN：Timer 时钟使能位
0 = 关闭 Timer 时钟，Timer 不工作
1 = 使能 Timer 时钟，Timer 工作

bit 1 LEDEN：LED 时钟使能位
0 = 关闭 LED 时钟，LED 不工作
1 = 使能 LED 时钟，LED 工作

bit 0 WDTEN：WDT 时钟使能位
0 = 关闭 WDT 时钟，WDT 不工作
1 = 使能 WDT 时钟，WDT 工作

注： 如果要配置某个模块功能时，必需首先配置 MDLCKCR，使能相应模块时钟，才能够进行这个模块的功能配置。

5. MCU 的特性

CBM73xx 系列芯片包含的许多特性旨在最大限度地提高系统的可靠性，通过减少外部元件将成本降至最低，并且还提供了静电防护、低功耗和代码保护功能。

这些功能包括：

- 复位
- 中断
- 休眠模式、空闲模式
- 静电防护
- 代码保护
- 在线串行编程
- 在线调试

MCU 有两种低功耗的工作模式，分别是休眠模式(SLEEP)和空闲模式(IDLE)。

软件配置 PCON=0x01(需先写 PSEQ 寄存器，具体请看 3.1.10 小节“写 PCON 前保护序列寄存器”)，即进入空闲模式，内核停止运行。用户可以通过以下方法从空闲模式唤醒：

- Timer0 中断
- Timer1 中断
- Timer2 中断
- Timer3 中断 0
- Timer3 中断 1
- sensorADC 中断
- RTC 中断
- SPI 中断
- I^2C Slave 中断
- UART 中断
- 外部中断 0、1、2、3、4、5

软件配置 PCON=0x02(需先写 PSEQ 寄存器，具体请看 3.1.10 小节“写 PCON 前保护序列寄存器”)，即进入休眠模式，进入休眠模式后，OSC48M 时钟停止工作，OSC800K、外部 32.768K 晶振(若外接了 32.768K 晶振)继续工作。用户可以通过以下方法从休眠模式唤醒：

- 外部中断 0 唤醒
- 定时自动唤醒
- RTC 定时唤醒
- 特定触摸按键唤醒

休眠唤醒后 MCU 将从下一条指令开始继续运行。

5.1. 复位

复位相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
RSTSR	0x54FF	XRAM	复位源状态寄存器	0000 0001
LVDCR	0x55FF	XRAM	低电压检测电路控制寄存器	x000 0010
MISCR	0x56FF	XRAM	POR 复位系统使能、OSC 输出滤波使能、多次读判决机制使能寄存器	0000 0000

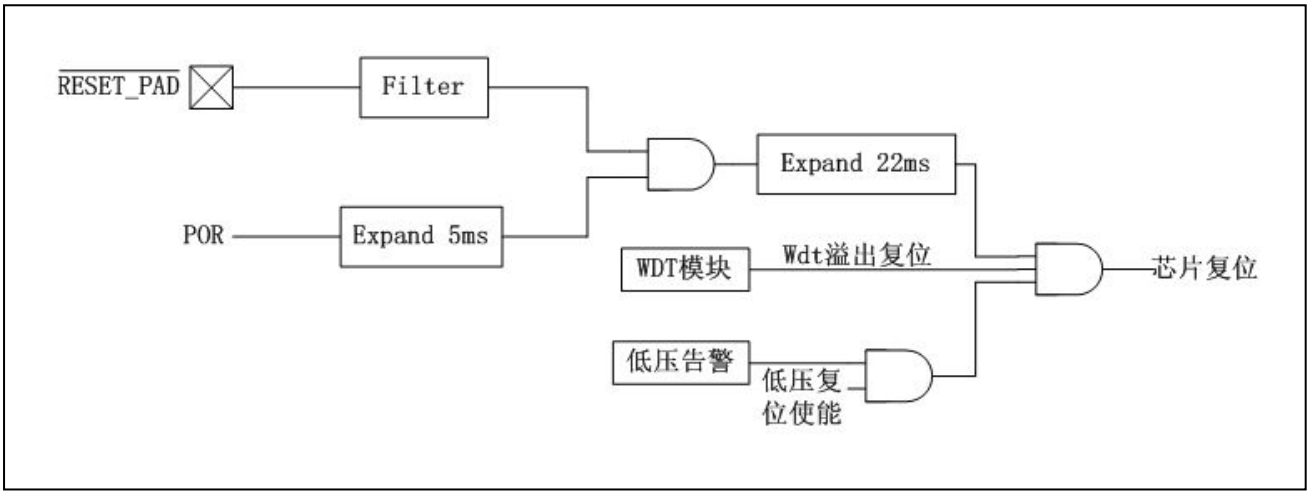
CBM73xx 系列芯片有以下几种不同类型的复位方式：

- 上电复位(POR)
- 正常工作期间的 WDT 溢出复位
- 正常工作期间的$\overline{\text{RESET_PAD}}$复位
- 休眠模式期间的$\overline{\text{RESET_PAD}}$复位
- 空闲模式期间的$\overline{\text{RESET_PAD}}$复位
- 低压复位

图 5-1 给出了片上复位电路的简化框图。

RSTSR 寄存器有 4bit 的复位源状态位，在不同的复位情形下，会分别被置 1 或清零。这些状态位在软件中用于判断产生复位的是哪个复位源。

图 5-1： 片上复位电路的简化框图



寄存器 5-1： RSTSR：复位源状态寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	RESETPAD	LVD	WDT	POR
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未实现：读为 0

bit 3 RESETPAD：外部复位标志位
0 = 复位源不是外部复位
1 = 复位源是外部复位

bit 2 LVD：低压复位标志位
0 = 复位源不是低压复位
1 = 复位源是低压复位

bit 1 WDT：看门狗溢出复位标志位
0 = 复位源不是看门狗溢出复位
1 = 复位源是看门狗溢出复位

bit 0 POR：上电复位标志位
0 = 复位源不是上电复位
1 = 复位源是上电复位

注： 该寄存器各 bit 可由软件清零。

5.1.1. 上电复位(POR)

在 VDD 达到足以使芯片正常工作的电平以前，片上上电复位电路使芯片保持复位状态。如果使能了低压复位，低压复位电路在 VDD 达到 VLVD 以前将保持芯片为复位状态(见第 5.1.2 节“低压复位”)。

当芯片开始正常工作(退出复位状态)时，芯片的工作参数(即电压、频率和温度等)必须得到满足，以确保其正常工作。如果不满足这些条件，那么芯片必须保持在复位状态，直到满足工作条件为止。

MISCR 寄存器(寄存器 5-2)bit0 是上电复位使能禁止位，MISCR_POREN = 0,上电复位使能，MISCR_POREN = 1,上电复位禁止。

寄存器 5-2： MISCR：POR、OSC 输出、多次读判决控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	FMULRDEN	GFLTEN	POREN
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-3 未实现：读为 0

bit 2 FMULRDEN：FLASH 多次读判决机制使能位
0 = 禁止 FLASH 多次读判决机制
1 = 使能 FLASH 多次读判决机制

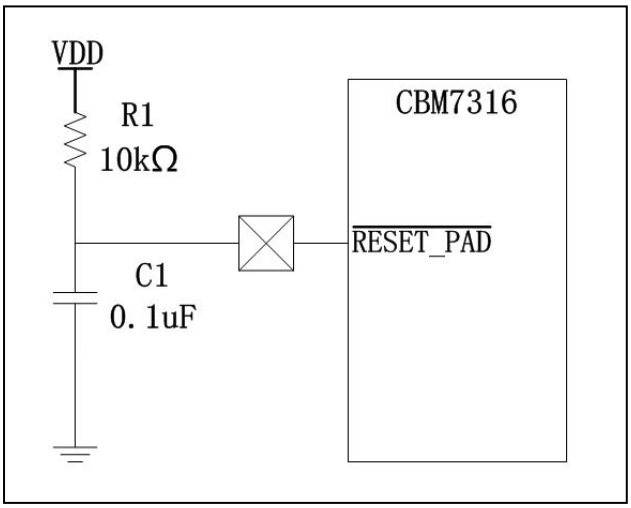
bit 1 GFLTEN：OSC 输出毛刺滤波使能位
0 = OSC 输出不经过毛刺滤波
1 = OSC 输出经过毛刺滤波

bit 0 POREN：上电复位禁止位
0 = 当掉电到 POR 复位电压门限时，复位系统
1 = 当掉电到 POR 复位电压门限时，不复位系统

5.1.2. 外部复位 $\overline{\text{RESET\_PAD}}$

$\overline{\text{RESET_PAD}}$引脚为外部复位引脚，为低电平复位。建议使用如图 5-2 所示的电路。

图 5-2： 推荐的外部复位电路



5.1.3. 低压复位

LVDCR 寄存器是低压检测电路控制寄存器，其中 OPER<1:0>位是检测到低压告警信号后处理措施选择位：

1. OPER<1:0> = 00，暂停工作，电压恢复到阀值之上后从下一条指令继续工作；

2. OPER<1:0> = 01，系统复位，电压恢复到阀值之上后复位开始运行；

3. OPER<1:0> = 110，只记录告警信号；

4. OPER<1:0> = 11，产生中断。

VOL<1:0>是低压复位电压选择位,有 4 种电压可以选择，VOL<1:0> = 00，低压复位电压为 2.7V，VOL<1:0> = 01，低压复位电压为 2.9V，VOL<1:0> = 10，低压复位电压为 3.1V，VOL<1:0> = 11，低压复位电压为 3.3V。

低压复位使能，如果 VDD 下降到 VLVD 以下，且持续时间超过参数值(TLVD)（见第 21.0 小节“电气特性”），低压状况将使芯片复位。不管 VDD 的变化速率如何，上述情况都会发生。如果 VDD 低于 VLVD 的时间少于参数(TLVD)，则不一定发生复位。FLTLEN<2:0>是 LVD 告警信号毛刺过滤宽度设置位，可以改变参数值 TLVD。

任何复位(上电复位、低压复位和看门狗溢出复位等)都会使芯片保持复位状态，直到 VDD 上升到 VLVD 以上(见图 5-3)。

寄存器 5-3： LVDCR：低压检测电路控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
LVDSR	FLTLEN2	FLTLEN1	FLTLEN0	VOL1	VOL0	OPER1	OPER0
R/W-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

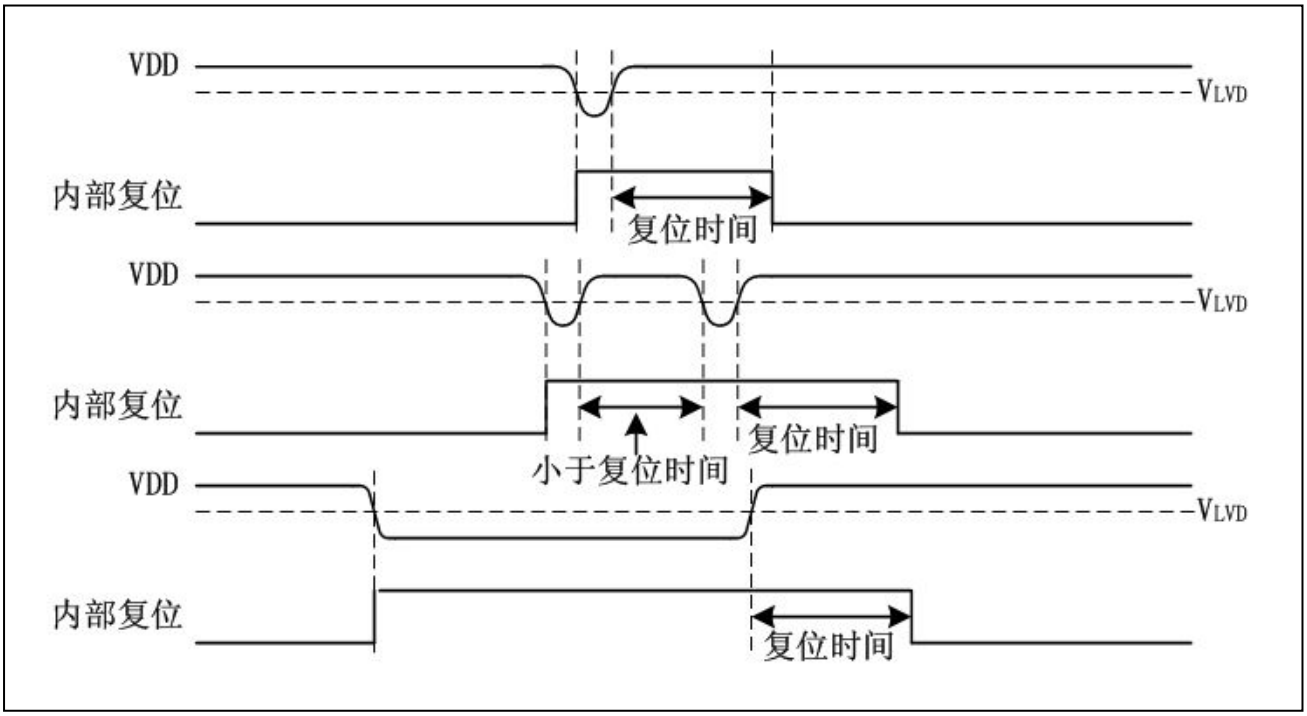
bit 7 LVDSR：低压告警状态位
0 = 无低压告警发生
1 = 有低压告警发生

bit 6-4 FLTLEN<2:0>：LVD 告警信号毛刺过滤宽度设置位
000 = 1us
001 = 2us
010 = 4us
011 = 8us
100 = 16us
101 = 32us
110 = 64us
111 = 128us

bit 3-2 VOL<1:0>：低压复位电压选择位
00 = 2.7V
01 = 2.9V
10 = 3.1V
11 = 3.3V

bit 1-0 OPER<1:0>：检测到低压告警信号处理措施选择位
00 = 系统暂停工作，直到电压恢复到检测阈值之上
01 = 系统复位，直到电压恢复到检测阈值之上
10 = 仅记录状态
11 = 产生中断

图 5-3： 低压复位情形


VDD
VLVD
内部复位
复位时间
VDD
VLVD
内部复位
小于复位时间
复位时间
VDD
VLVD
内部复位
复位时间

5.2. 中断

中断相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
IE	0xA8	SFR	中断使能寄存器	0000 0000
IP	0xB8	SFR	中断优先级控制寄存器	0000 0000
IFLTCNTR	0xF0FF	XRAM	外部中断信号过滤长度配置寄存器	0000 1111
XINT01TRGR	0xF1FF	XRAM	外部中断0、1触发方式配置寄存器	0000 0000
XINT23TRGR	0xF2FF	XRAM	外部中断2、3触发方式配置寄存器	0000 0000
XINT45TRGR	0xF3FF	XRAM	外部中断4、5触发方式配置寄存器	0000 0000
IRQ0SR	0xF4FF	XRAM	中断状态寄存器0	0000 0000
IRQ1SR	0xF5FF	XRAM	中断状态寄存器1	0000 0000
IRQ2SR	0xF6FF	XRAM	中断状态寄存器2	0000 0000
IRQ3SR	0xF7FF	XRAM	中断状态寄存器3	0000 0000
IRQ4SR	0xF8FF	XRAM	中断状态寄存器4	0000 0000
IRQ5SR	0xF9FF	XRAM	中断状态寄存器5	0000 0000
IRQ0ER	0xFAFF	XRAM	中断使能寄存器0	0000 0000
IRQ1ER	0xFBFF	XRAM	中断使能寄存器1	0000 0000
IRQ2ER	0xFCFF	XRAM	中断使能寄存器2	0000 0000
IRQ3ER	0xFDFF	XRAM	中断使能寄存器3	0000 0000
IRQ4ER	0xFEFF	XRAM	中断使能寄存器4	0000 0000
IRQ5ER	0xFFFF	XRAM	中断使能寄存器5	0000 0000

CBM73xx 系列芯片具有以下多种中断源：

- 外部中断 0、1、2、3、4、5
- 看门狗溢出中断
- 低压检测中断
- Timer0 中断
- Timer1 中断
- Timer2 中断
- Timer3 中断 0
- Timer3 中断 1
- SensorADC 中断
- ADC 中断
- RTC 中断
- UART 中断
- I2C Master 中断
- I2C Slave 中断
- SPI 中断

CBM73xx 系列芯片中所有的中断源合并成 6 个中断送给内核。

5.2.1. 中断使能步骤

接口或外设产生一个中断时，要形成真正的中断(打断内核执行，跳转到中断服务程序入口)，需要设置多个使能，包括：

第 1 步：外设模块中的中断使能(例如 TOCR 寄存器的 INTEN 位控制使能 Timer0 中断，UIER 寄存器的 TXIE 控制使能 UART 数据发送完成中断)；

第 2 步：各个模块的中断使能(IRQnER(n = 0、1、2、3、4、5)寄存器中的对应位控制(例如 IRQ2ER 寄存器的 TCM0INT 位控制使能 Timer0 模块的中断，IRQ3ER 寄存器的 UARTINT 位控制使能 UART 模块的中断)；

第 3 步：内核各个中断优先级配置(IP 寄存器的 PX0、PX1、…、PX5 位控制)**(需要注意每个内核中断都包含多个模块的中断，需要协调好这些中断的优先级。若不需要用到中断优先级则可以跳过这一步)**；

第 4 步：内核各个中断使能(IE 寄存器的 EX0、EX1、…、EX5 位控制)；

第 5 步：内核全局中断使能(IE 寄存器的 EA 位控制)。

例如使能 UART 接收数据完成中断的步骤如下：

第 1 步：将 UIER 寄存器的 RXIE 位置 1，使能 UART 接收数据完成中断；

第 2 步：将 IRQ3ER 寄存器的 UARTINT 位置 1，使能 UART 模块中断；(UART 模块包括发送数据完成、接收数据完成两路中断)

第 3 步：配置 IP 寄存器的 PX3 位，设置内核中断 3 的中断优先级；(低优先级清 0，高优先级置 1)

第 4 步：将 IE 寄存器的 EX3 置 1，使能内核中断 3 中断；(内核中断 3 包含外部中断 3、Timer1 中断、UART 中断三个中断)

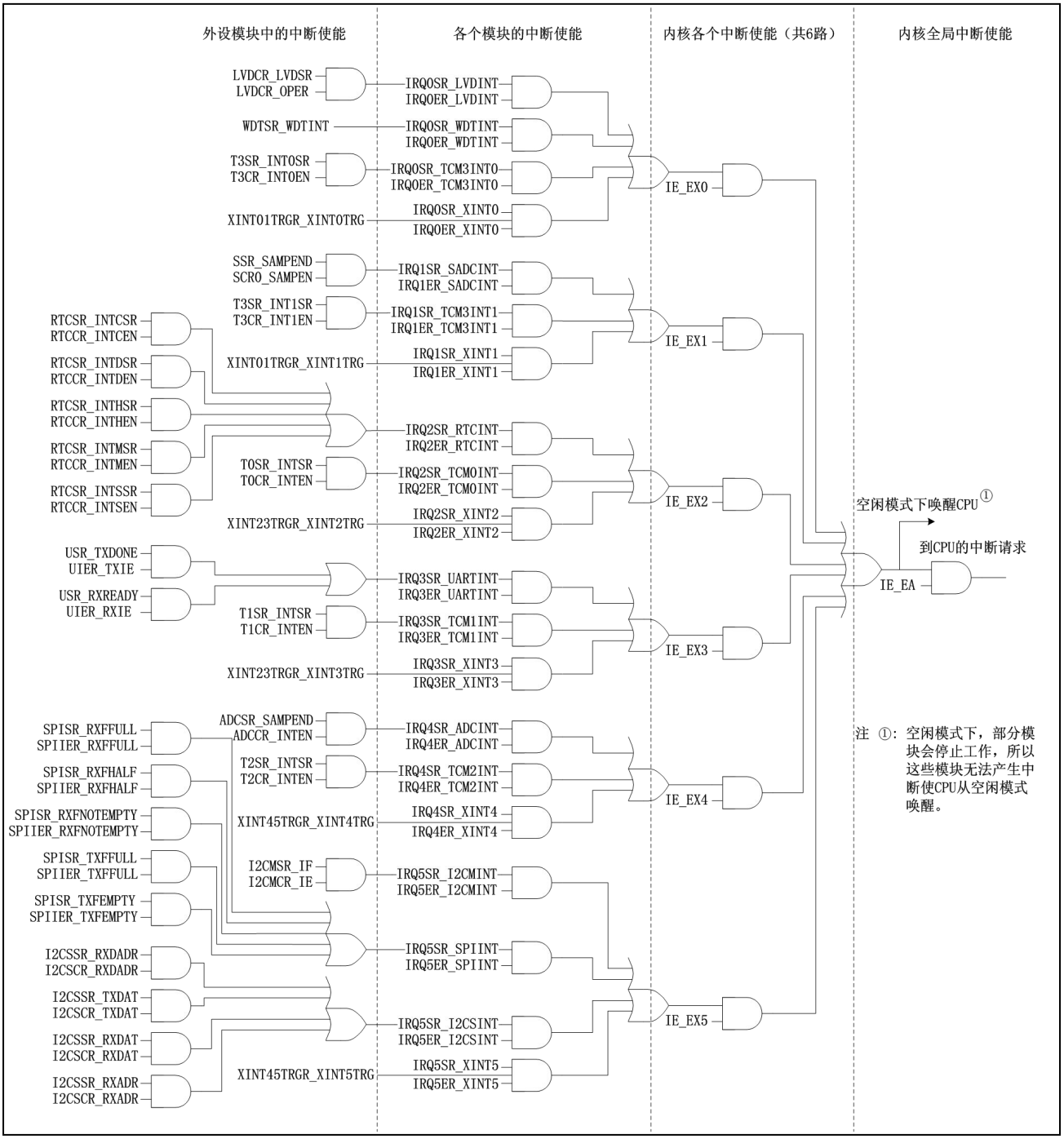
第 5 步：将 IE 寄存器的 EA 位置 1，使能内核全局中断。

CBM7332A6Q1 芯片的中断源如表 5-1 所示；CBM7332A6Q1 芯片的中断控制逻辑如图 5-4 所示：

表 5-1：中断源汇总表

中断总使能位	内核中断使能位	优先级控制位	内核中断	中断源	相关寄存器
EA	EX0	PX0	内核中断0	低电压检测中断	IRQ0SR_LVDINT，IRQ0ER_LVDINT
				看门狗中断	IRQ0SR_WDTINT，IRQ0ER_WDTINT
				定时器3中断0	IRQ0SR_TCM3INT0，IRQ0ER_TCM3INT0
				外部中断0	IRQ0SR_XINT0，IRQ0ER_XINT0
	EX1	PX1	内核中断1	SensorADC 中断	IRQ1SR_SADCINT，IRQ1ER_SADCINT
				定时器3中断1	IRQ1SR_TCM3INT1，IRQ1ER_TCM3INT1
				外部中断1	IRQ1SR_XINT1，IRQ1ER_XINT1
	EX2	PX2	内核中断2	RTC 中断	IRQ2SR_RTCINT，IRQ2ER_RTCINT
				定时器0中断	IRQ2SR_TCM0INT，IRQ2ER_TCM0INT
				外部中断2	IRQ2SR_XINT2，IRQ2ER_XINT2
	EX3	PX3	内核中断3	串口中断	IRQ3SR_UARTINT，IRQ3ER_UARTINT
				定时器1中断	IRQ3SR_TCM1INT，IRQ3ER_TCM1INT
				外部中断3	IRQ3SR_XINT3，IRQ3ER_XINT3
	EX4	PX4	内核中断4	通用 ADC 中断	IRQ4SR_ADCINT，IRQ4ER_ADCINT
				定时器2中断	IRQ4SR_TCM2INT，IRQ4ER_TCM2INT
				外部中断4	IRQ4SR_XINT4，IRQ4ER_XINT4
	EX5	PX5	内核中断5	I2C Master 中断	IRQ5SR_I2CMINT，IRQ5ER_I2CMINT
				SPI 中断	IRQ5SR_SPIINT，IRQ5ER_SPIINT
				I2C Slave 中断	IRQ5SR_I2CSINT，IRQ5ER_I2CSINT
				外部中断5	IRQ5SR_XINT5，IRQ5ER_XINT5

图 5-4：中断控制逻辑图


外设模块中的中断使能
各个模块的中断使能
内核各个中断使能（共6路）
内核全局中断使能
LVDCR_LVDSR
LVDCR_OPER
IRQ0SR_LVDINT
IRQ0ER_LVDINT
WDTSR_WDTINT
IRQ0SR_WDTINT
IRQ0ER_WDTINT
T3SR_INT0SR
T3CR_INT0EN
IRQ0SR_TCM3INT0
IRQ0ER_TCM3INT0
IE_EX0
XINT01TRGR_XINT0TRG
IRQ0SR_XINT0
IRQ0ER_XINT0
SSR_SAMPEND
SCR0_SAMPEN
IRQ1SR_SADCINT
IRQ1ER_SADCINT
T3SR_INT1SR
T3CR_INT1EN
IRQ1SR_TCM3INT1
IRQ1ER_TCM3INT1
IE_EX1
RTCSR_INTCSR
RTCCR_INTCEN
XINT01TRGR_XINT1TRG
IRQ1SR_XINT1
IRQ1ER_XINT1
RTCSR_INTDSR
RTCCR_INTDEN
RTCSR_INTHSR
RTCCR_INTHEN
IRQ2SR_RTCINT
IRQ2ER_RTCINT
RTCSR_INTMSR
RTCCR_INTMEN
T0SR_INTSR
T0CR_INTEN
IRQ2SR_TCM0INT
IRQ2ER_TCM0INT
IE_EX2
RTCSR_INTSSR
RTCCR_INTSEN
XINT23TRGR_XINT2TRG
IRQ2SR_XINT2
IRQ2ER_XINT2
空闲模式下唤醒CPU ①
到CPU的中断请求
USR_TXDONE
UIER_TXIE
IRQ3SR_UARTINT
IRQ3ER_UARTINT
IE_EA
USR_RXREADY
UIER_RXIE
T1SR_INTSR
T1CR_INTEN
IRQ3SR_TCM1INT
IRQ3ER_TCM1INT
IE_EX3
XINT23TRGR_XINT3TRG
IRQ3SR_XINT3
IRQ3ER_XINT3
ADCSR_SAMPEND
ADCCR_INTEN
IRQ4SR_ADCINT
IRQ4ER_ADCINT
SPISR_RXFFULL
SPIIER_RXFFULL
T2SR_INTSR
T2CR_INTEN
IRQ4SR_TCM2INT
IRQ4ER_TCM2INT
SPISR_RXFHALF
SPIIER_RXFHALF
IE_EX4
SPISR_RXFNOTEMPTY
SPIIER_RXFNOTEMPTY
XINT45TRGR_XINT4TRG
IRQ4SR_XINT4
IRQ4ER_XINT4
SPISR_TXFFULL
SPIIER_TXFFULL
I2CMSR_IF
I2CMCR_IE
IRQ5SR_I2CMINT
IRQ5ER_I2CMINT
SPISR_TXFEMPTY
SPIIER_TXFEMPTY
IRQ5SR_SPIINT
IRQ5ER_SPIINT
I2CSSR_RXDADR
I2CSCR_RXDADR
I2CSSR_TXDAT
I2CSCR_TXDAT
IRQ5SR_I2CSINT
IRQ5ER_I2CSINT
IE_EX5
I2CSSR_RXDAT
I2CSCR_RXDAT
XINT45TRGR_XINT5TRG
IRQ5SR_XINT5
IRQ5ER_XINT5
I2CSSR_RXADR
I2CSCR_RXADR
注 ①：空闲模式下，部分模块会停止工作，所以这些模块无法产生中断使CPU从空闲模式唤醒。

IRQnER 寄存器是中断使能寄存器，n = 0、1、2、3、4、5，分别对应六个中断向量，包含所有的中断源。

IRQnSR 寄存器是中断状态寄存器，n = 0、1、2、3、4、5，是对应于 IRQnER 寄存器的中断状态寄存器。

寄存器 5-4： IRQ0ER：中断使能寄存器 0

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	LVDINT	WDTINT	TCM3INT0	XINT0
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未实现：读为 0

bit 3 LVDINT：LVD 中断使能位
0 = 禁止 LVD 中断
1 = 使能 LVD 中断

bit 2 WDTINT：WDT 中断使能位
0 = 禁止 WDT 溢出中断
1 = 使能 WDT 溢出中断

bit 1 TCM3INT0：Timer3 中断 0 使能位
0 = 禁止 Timer3 中断 0 中断
1 = 使能 Timer3 中断 0 中断

bit 0 XINT0：外部中断 0 使能位
0 = 禁止外部中断 0 中断
1 = 使能外部中断 0 中断

寄存器 5-5： IRQ1ER：中断使能寄存器 1

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	SADCINT	TCM3INT1	XINT1
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-3 未实现：读为 0

bit 2 SADCINT：SADC 中断使能位

0 = 禁止 SADC 中断

1 = 使能 SADC 中断

bit 1 TCM3INT1：Timer3 中断 1 使能位

0 = 禁止 Timer3 中断 1 中断

1 = 使能 Timer3 中断 1 中断

bit 0 XINT1：外部中断 1 使能位

0 = 禁止外部中断 1 中断

1 = 使能外部中断 1 中断

寄存器 5-6： IRQ2ER：中断使能寄存器 2

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	RTCINT	TCM0INT	XINT2
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-3 未实现：读为 0

bit 2 RTCINT：RTC 中断使能位

0 = 禁止 RTC 中断

1 = 使能 RTC 中断

bit 1 TCM0INT：Timer0 中断使能位

0 = 禁止 Timer0 中断

1 = 使能 Timer0 中断

bit 0 XINT2：外部中断 2 使能位

0 = 禁止外部中断 2 中断

1 = 使能外部中断 2 中断

寄存器 5-7： IRQ3ER：中断使能寄存器 3

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	UARTINT	TCM1INT	XINT3
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-3 未实现：读为 0

bit 2 UARTINT：UART 中断使能位
0 = 禁止 UART 中断
1 = 使能 UART 中断

bit 1 TCM1INT：Timer1 中断使能位
0 = 禁止 Timer1 中断
1 = 使能 Timer1 中断

bit 0 XINT3：外部中断 3 使能位
0 = 禁止外部中断 3 中断
1 = 使能外部中断 3 中断

寄存器 5-8： IRQ4ER：中断使能寄存器 4

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	ADCINT	TCM2INT	XINT4
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未实现：读为 0

bit 2 ADCINT：ADC 中断使能位
0 = 禁止 ADC 中断
1 = 使能 ADC 中断

bit 1 TCM2INT：Timer2 中断使能位
0 = 禁止 Timer2 中断
1 = 使能 Timer2 中断

bit 0 XINT4：外部中断 4 使能位
0 = 禁止外部中断 4 中断
1 = 使能外部中断 4 中断

寄存器 5-9： IRQ5ER：中断使能寄存器 5

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	I2CMINT	SPIINT	I2CSINT	XINT5
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未实现：读为 0

bit 3 I2CMINT：I2C Master 中断使能位

0 = 禁止 I2C Master 中断

1 = 使能 I2C Master 中断

bit 2 SPIINT：SPI 中断使能位

0 = 禁止 SPI 中断

1 = 使能 SPI 中断

bit 1 I2CSINT：I2C Slave 中断使能位

0 = 禁止 I2C Slave 中断

1 = 使能 I2C Slave 中断

bit 0 XINT5：外部中断 5 使能位

0 = 禁止外部中断 5 中断

1 = 使能外部中断 5 中断

寄存器 5-10： IRQ0SR：中断状态寄存器 0

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	LVDINT	WDTINT	TCM3INT0	XINT0
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未实现：读为 0

bit 3 LVDINT：LVD 中断状态位
0 = 未产生 LVD 告警
1 = 已产生 LVD 告警

bit 2 WDTINT：WDT 中断状态位
0 = 未产生 WDT 溢出
1 = 已产生 WDT 溢出

bit 1 TCM3INT0：Timer3 中断 0 状态位
0 = 未产生 Timer3 中断 0 中断条件
1 = 已产生 Timer3 中断 0 中断条件

bit 0 XINT0：外部中断 0 状态位
0 = 未产生外部中断 0 中断条件
1 = 已产生外部中断 0 中断条件

寄存器 5-11： IRQ1SR：中断状态寄存器 1

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	SADCINT	TCM3INT1	XINT1
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-3 未实现：读为 0

bit 2 SADCINT：SADC 中断状态位

0 = 未完成 SADC 采样

1 = 已完成 SADC 采样

bit 1 TCM3INT1：Timer3 中断 1 状态位

0 = 未产生 Timer3 中断 1 中断条件

1 = 已产生 Timer3 中断 1 中断条件

bit 0 XINT1：外部中断 1 状态位

0 = 未产生外部中断 1 中断条件

1 = 已产生外部中断 1 中断条件

寄存器 5-12： IRQ2SR：中断状态寄存器 2

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	RTCINT	TCM0INT	XINT2
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-3 未实现：读为 0

bit 2 RTCINT：RTC 中断状态位

0 = 未产生 RTC 中断条件

1 = 已产生 RTC 中断条件

bit 1 TCM0INT：Timer0 中断状态位

0 = 未产生 Timer0 计数匹配

1 = 已产生 Timer0 计数匹配

bit 0 XINT2：外部中断 2 状态位

0 = 未产生外部中断 2 中断条件

1 = 已产生外部中断 2 中断条件

寄存器 5-13： IRQ3SR：中断状态寄存器 3

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	UARTINT	TCM1INT	XINT3
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-3 未实现：读为 0

bit 2 UARTINT：UART 中断状态位
0 = 未完成 UART 字节发送或接受
1 = 已完成 UART 字节发送或接受

bit 1 TCM1INT：Timer1 中断状态位
0 = 未产生 Timer1 中断条件
1 = 已产生 Timer1 中断条件

bit 0 XINT3：外部中断 3 状态位
0 = 未产生外部中断 3 中断条件
1 = 已产生外部中断 3 中断条件

寄存器 5-14： IRQ4SR：中断状态寄存器 4

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	ADCINT	TCM2INT	XINT4
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未实现：读为 0

bit 2 ADCINT：ADC 中断状态位
0 = 未完成 ADC 采样
1 = 已完成 ADC 采样

bit 1 TCM2INT：Timer2 中断状态位
0 = 未产生 Timer2 计数匹配
1 = 已产生 Timer2 计数匹配

bit 0 XINT4：外部中断 4 状态位
0 = 未产生外部中断 4 中断条件
1 = 已产生外部中断 4 中断条件

寄存器 5-15： IRQ5SR：中断状态寄存器 5

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	I2CMINT	SPIINT	I2CSINT	XINT5
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未实现：读为 0

bit 3 I2CMINT：I2C Master 中断状态位

0 = 未产生 I2C Master 中断条件

1 = 已产生 I2C Master 中断条件

bit 2 SPIINT：SPI 中断状态位

0 = 未产生 SPI 中断条件

1 = 已产生 SPI 中断条件

bit 1 I2CSINT：I2C Slave 中断状态位

0 = 未产生 I2C Slave 中断条件

1 = 已产生 I2C Slave 中断条件

bit 0 XINT5：外部中断 5 状态位

0 = 未产生外部中断 5 中断条件

1 = 已产生外部中断 5 中断条件

5.2.2. 中断入口

CBM73xx 系列芯片的中断入口地址是在标准的 8051 的中断入口地址上加一个偏移地址，偏移地址由 VTOR 寄存器配置(VTOR 寄存器配置新的中断入口地址的高 8 位，而低 8 位仍取决于中断向量)。

VTOR 寄存器上电复位值为 0x00，即复位时中断入口地址和 8051 中断入口地址一致。若用户开发程序的起始地址为 0x0800，则可以将中断入口地址的高 8 位设置到中断入口偏移地址寄存器 VTOR 中，使中断发生时，程序跳转到新的中断入口处。

若配置 VTOR = 0x08，则新的中断入口地址，如下表所示：

CBM73xx 系列芯片中断入口地址表

中断向量	中断入口地址(原始)	中断入口地址(加偏移)
INT0	0x0003	0x0803
INT1	0x000b	0x080b
INT2	0x0013	0x0813
INT3	0x001b	0x081b
INT4	0x0023	0x0823
INT5	0x002b	0x082b

注：设置了该中断向量偏移寄存器后，产生中断时，程序会跳转到加了偏移的中断入口地址处执行。所以用户的程序需确保中断服务程序被链接到了预期的新地址处。

5.2.3. 中断优先级

CBM73xx 系列芯片中断优先级有两级，IP 寄存器控制 6 个内核中断的优先级，每个内核中断包含多个外设中断，属于同一内核中断的外设中断处于同一优先级。详细请查看第 2.2 节中的 IP 寄存器介绍。

中断优先级规则如下：

1、高优先级的中断可以打断低优先级中断的处理，反之，不行。

2、同一级的中断，后产生的中断不能打断目前正在处理的中断。

3、同一级的中断也存在优先级；若同一级的中断同时产生，高优先级的优先处理。优先级如下：PX0 > PX1 > … > PX5，

注： 1、各中断标志位的置 1 不受对应的中断允许 位 和 EA 位的影响。

2、当执行一条清零 EA 位的指令后，任何一条等待在下一周期执行的中断都将被忽略。当 EA 位再次置 1 后，被忽略的中断仍会继续等待处理。

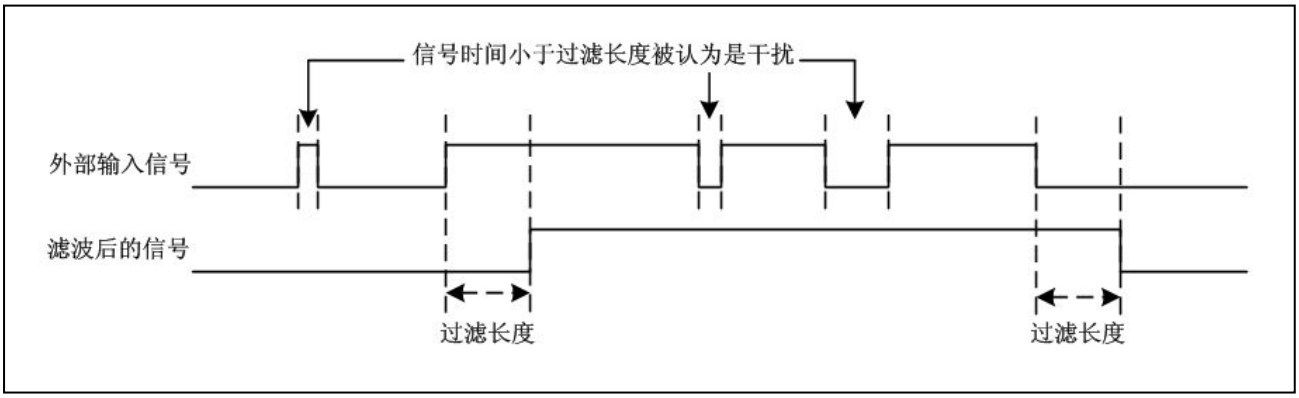
5.2.4. 外部中断

CBM73xx 系列芯片最多支持 6 路外部中断，通过 XINTxxTRGR 寄存器可以配置各个外部中断的触发方式，包括高电平触发、低电平触发、上升沿触发、下降沿触发、双边沿触发、不触发共 6 种方式。

为避免外部干扰，硬件上对外部信号进行了过滤。即，外部 pad 上的电平需持续某个长度的时间，才会认为是正确的电平。所以，即使外部 pad 上发生真实的电平变化，也会滞后一段时间进入中断服务程序。若持续时间不够，则会被认为是干扰信号而被忽略。即使是边沿触发，亦是处理过滤后的边沿情况(见图 5-5)。

信号过滤的时间长度由 IFLTCNTR 寄存器设置，默认是过滤 16 个系统时钟。

图 5-5： 外部中断信号滤波处理



寄存器 5-16： IFLTCNTR：外部中断信号过滤长度配置寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IFLTCNT7	IFLTCNT6	IFLTCNT5	IFLTCNT4	IFLTCNT3	IFLTCNT2	IFLTCNT1	IFLTCNT0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 IFLTCNT<7:0>：外部中断信号过滤长度配置位

0000 0000 = 1 * Tsys
0000 0001 = 2 * Tsys
0000 0010 = 3 * Tsys
0000 0011 = 4 * Tsys
0000 0100 = 5 * Tsys
0000 0101 = 6 * Tsys
0000 0110 = 7 * Tsys
0000 0111 = 8 * Tsys
……
1111 1000 = 249 * Tsys
1111 1001 = 250 * Tsys
1111 1010 = 251 * Tsys
1111 1011 = 252 * Tsys
1111 1100 = 253 * Tsys
1111 1101 = 254 * Tsys
1111 1110 = 255 * Tsys
1111 1111 = 256 * Tsys

寄存器 5-17： XINT01TRGR：外部中断 0、1 触发方式配置寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	XINT1TRG2	XINT1TRG1	XINT1TRG0	—	XINT0TRG2	XINT0TRG1	XINT0TRG0
U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 未实现：读为 0

bit 6-4 XINT1TRG<2:0>：外部中断 1 触发方式配置位

000 = 高电平触发，当外部中断 pad 为高电平时，触发中断。

001 = 低电平触发，当外部中断 pad 为低电平时，触发中断。

010 = 不触发

011 = 不触发

100 = 不触发

101 = 上升沿触发

110 = 下降沿触发

111 = 双沿触发(上升沿和下降沿都可以触发中断)

bit 3 未实现：读为 0

bit 2-0 XINT0TRG<2:0>：外部中断 0 触发方式配置位

000 = 高电平触发，当外部中断 pad 为高电平时，触发中断。

001 = 低电平触发，当外部中断 pad 为低电平时，触发中断。

010 = 不触发

011 = 不触发

100 = 不触发

101 = 上升沿触发

110 = 下降沿触发

111 = 双沿触发(上升沿和下降沿都可以触发中断)

电平触发中断和边沿触发中断的区别：

1、电平触发就是只要电平变为有效电平，在中能使能的前提下，软件会进入中断服务程序。由于内核的 6 个中断引脚都复用了至少 3 个中断源(见 IRQnER 寄存器)，所以在中断服务程序中，软件需查询中断状态寄存器，以判断是由哪个情况引起的中断。从触发中断到软件查询中断寄存器状态，需要一定的时间，在这段时间内，触发中断的有效电平需要持续。若电平失效，软件可能误认为是其他原因产生的中断，导致无法正确处理。

电平触发的中断需要外部主动清除(将外部中断 pad 上的输入电平置为无效电平)。若外部中断 pad 上的中断有效电平一直持续，内核在处理完中断后，会再次进入中断服务程序，导致反复多次进入中断服务程序。如果不希望再次进入中断，则需要外部在首次进入中断服务程序且退出中断服务程序前，主动清除此外部中断。

2、边沿触发中断是靠外部中断 pad 的边沿触发的，即只要出现有效的沿变，硬件会记录下来曾经发生过中断请求。例如，配置成上升沿触发，外部 pad 在从低变高后，硬件即会记录下从低变高的事件，外部 pad 可以恢复到低电平，也不影响此次中断处理。但是，这种情况下的中断服务程序中需要由软件清除中断状态，否则处理完此次中断后，会再次进入中断服务程序。

寄存器 5-18： XINT23TRGR：外部中断 2、3 触发方式配置寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	XINT3TRG2	XINT3TRG1	XINT3TRG0	—	XINT2TRG2	XINT2TRG1	XINT2TRG0
U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 未实现：读为 0

bit 6-4 XINT3TRG<2:0>：外部中断 3 触发方式配置位

000 = 高电平触发，当外部中断 pad 为高电平时，触发中断。

001 = 低电平触发，当外部中断 pad 为低电平时，触发中断。

010 = 不触发

011 = 不触发

100 = 不触发

101 = 上升沿触发

110 = 下降沿触发

111 = 双沿触发(上升沿和下降沿都可以触发中断)

bit 3 未实现：读为 0

bit 2-0 XINT2TRG<2:0>：外部中断 2 触发方式配置位

000 = 高电平触发，当外部中断 pad 为高电平时，触发中断。

001 = 低电平触发，当外部中断 pad 为低电平时，触发中断。

010 = 不触发

011 = 不触发

100 = 不触发

101 = 上升沿触发

110 = 下降沿触发

111 = 双沿触发(上升沿和下降沿都可以触发中断)

寄存器 5-19： XINT45TRGR：外部中断 4、5 触发方式配置寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	XINT5TRG2	XINT5TRG1	XINT5TRG0	—	XINT4TRG2	XINT4TRG1	XINT4TRG0
U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 未实现：读为 0

bit 6-4 XINT5TRG<2:0>：外部中断 5 触发方式配置位

000 = 高电平触发，当外部中断 pad 为高电平时，触发中断。

001 = 低电平触发，当外部中断 pad 为低电平时，触发中断。

010 = 不触发

011 = 不触发

100 = 不触发

101 = 上升沿触发

110 = 下降沿触发

111 = 双沿触发(上升沿和下降沿都可以触发中断)

bit 3 未实现：读为 0

bit 2-0 XINT4TRG<2:0>：外部中断 4 触发方式配置位

000 = 高电平触发，当外部中断 pad 为高电平时，触发中断。

001 = 低电平触发，当外部中断 pad 为低电平时，触发中断。

010 = 不触发

011 = 不触发

100 = 不触发

101 = 上升沿触发

110 = 下降沿触发

111 = 双沿触发(上升沿和下降沿都可以触发中断)

5.3. 休眠与空闲模式

CBM73xx 系列芯片有两种低功耗的工作模式，分别是休眠模式(SLEEP)和空闲模式(IDLE)。

5.3.1. 休眠模式(SLEEP)

软件配置 PCON = 0x02(配置 PCON 前必须先写 PSEQ，具体见 3.1.10 小节“写 PCON 前的保护序列寄存器”)，即进入休眠模式，进入休眠模式后，OSC48M 时钟停止工作，OSC800K、外部 32.768K 晶振(若外接了 32.768K 晶振)会继续工作。

5.3.1.1. 休眠唤醒

休眠唤醒相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
SLPWKCR	0x52FF	XRAM	休眠唤醒控制寄存器	0000 0000
SLPWKDR	0x53FF	XRAM	休眠唤醒参数配置寄存器	1000 0000

通过下面四种方式可以从休眠状态唤醒：

- 外部中断 0 唤醒
- 定时自动唤醒
- RTC 定时唤醒(必须外接 32.768K 晶振)
- 特定触摸按键唤醒

退出休眠模式后 MCU 从下一条指令继续运行。

外部中断 0 唤醒操作步骤：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成外部中断 0 功能时，会自动将引脚方向配置为输入，不需要配置对应的 PnOE 寄存器选择外部中断 0 引脚方向为输入。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. 外部中断 0 唤醒的有效沿选择：

SLPWKCR 寄存器的 EDGESEL 位可以选择外部中断 0 唤醒的有效沿，EDGESEL=0 为下降沿唤醒，EDGESEL=1 为上升沿唤醒。

3. 使能外部中断 0 唤醒：

将 SLPWKCR 寄存器的 EINTWKEN 位置 1，使能外部中断 0 唤醒。

当外部中断 0 输入引脚有有效沿输入时，就会唤醒 CPU。

定时自动唤醒操作步骤：

1. 定时自动唤醒等待周期数配置

定时自动唤醒等待周期数 ATWKCNT 是一个 24 位的数。通过 SLPWKCR 寄存器的 DRSEL<2:0>位和 SLPWKDR 寄存器的配合使用，可以配置定时自动唤醒等待周期数 ATWKCNT。

例如：若要将 ATWKCNT 设置成 0x123456，需执行如下 3 个操作(不分先后)：

1)设置 SLPWKCR_DRSEL=000，写 SLPWKDR=0x56

2)设置 SLPWKCR_DRSEL=001，写 SLPWKDR=0x34

3)设置 SLPWKCR_DRSEL=010，写 SLPWKDR=0x12

若固件要读取 ATWKCNT 的数值，需执行如下 4 个操作：

1)设置 SLPWKCR_DRSEL=000，读 SLPWKDR；

2)设置 SLPWKCR_DRSEL=001，读 SLPWKDR；

3)设置 SLPWKCR_DRSEL=010，读 SLPWKDR；

4)将前 3 次读到的数据拼接为 24 位数

2. 使能定时自动唤醒

将 SLPWKCR 寄存器的 AUTOWKEN 位置 1，使能定时自动唤醒。

定时自动唤醒是以 OSC800K 经过 SLOWDIV<1:0>分频后的时钟为时基，每个时钟计数值加 1，当计数值与 ATWKCNT 匹配时，就会唤醒 CPU。

RTC 定时唤醒操作步骤：

1. 配置 RTC 模块

使能 RTC 功能，具体配置见 13.0 小节“实时时钟模块”。

2. 使能 RTC 定时唤醒

将 SLPWKCR 寄存器的 RTCWKEN 位置 1，使能 RTC 定时唤醒。

当 RTC 模块时间寄存器与闹钟寄存器的值匹配时，就会唤醒 CPU。

特定触摸按键唤醒操作步骤：

1. 配置 sensorADC 模块

配置特定触摸按键唤醒操作步骤基本上与正常工作模式下配置 sensorADC 的操作步骤一致。只有第 13 步禁止休眠唤醒相关配置要改为使能休眠唤醒相关配置。也就是将 SWKCR 寄存器的 CLKSEL 位置 1，选择内部低速 OSC800K 经过 CKDIVCR 寄存器的 SLOWDIV<1:0>位分频后再 4 分频的时钟作为 sensorADC 的时钟。

配置 SWKCR 寄存器的 CMPRSET<2:0>位，设置采样值连续落入阈值内的次数。

2. 按键电容采样值范围设置

按键电容采样值是一个 16 位的数，通过 SLPWKCR 寄存器的 DRSEL<2:0>位和 SLPWKDR 寄存器配合使用来配置按键电容采样值的范围。

例如：若要将电容采样值的范围设置成 0x1234~0x5678，需执行如下 4 个操作(不分先后)：

1)设置 SLPWKCR_DRSEL=100，写 SLPWKDR=0x34

2)设置 SLPWKCR_DRSEL=101，写 SLPWKDR=0x12

3)设置 SLPWKCR_DRSEL=110，写 SLPWKDR=0x78

4)设置 SLPWKCR_DRSEL=111，写 SLPWKDR=0x56

若固件要读取电容采样值的范围，需执行如下 5 个操作：

1)设置 SLPWKCR_DRSEL=100，读 SLPWKDR

2)设置 SLPWKCR_DRSEL=101，读 SLPWKDR

3)设置 SLPWKCR_DRSEL=110，读 SLPWKDR

4)设置 SLPWKCR_DRSEL=111，读 SLPWKDR

5)将前 4 次读到的数据拼接为 2 个 16 位数 SADCMIN、SADCMAX。

3. 使能特定触摸按键唤醒

将 SLPWKCR 寄存器的 SENSORWKEN 位置 1，使能特定触摸按键唤醒。

使能特定触摸按键唤醒后，sensorADC 模块会不断对设定的 sensor 通道进行采样，当 sensorADC 采样值连续 N 次(由 SWKCR 寄存器的 CMPRSET<2:0>位配置)落入到预先设置的阈值范围内，就会唤醒 CPU。

5.3.1.2. 休眠唤醒寄存器的定义

以下寄存器用于休眠唤醒的操作

寄存器 5-20： SLPWKCR：SLEEP 唤醒控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
EDGESEL	DRSEL2	DRSEL1	DRSEL0	EINTWKEN	SENSORWKEN	RTCWKEN	AUTOWKEN
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 EDGESEL：外部中断 0 唤醒有效沿选择位
0 = 下降沿唤醒
1 = 上升沿唤醒

bit 6-4 DRSEL<2:0>：SLEEP 唤醒参数辅助配置位
000 = 使能读写 ATWKCNT 的低 8 位
001 = 使能读写 ATWKCNT 的中间 8 位
010 = 使能读写 ATWKCNT 的高 8 位
011 = 保留
100 = 使能读写 SADCMIN 的低 8 位
101 = 使能读写 SADCMIN 的高 8 位
110 = 使能读写 SADCMAX 的低 8 位
111 = 使能读写 SADCMAX 的高 8 位

bit 3 EINTWKEN：外部中断 0 唤醒使能位
0 = MCU 处于 SLEEP 模式时，禁止外部中断 0 唤醒
1 = MCU 处于 SLEEP 模式时，使能外部中断 0 唤醒

bit 2 SENSORWKEN：特定触摸按键唤醒使能位
0 = MCU 处于 SLEEP 模式时，禁止特定触摸按键唤醒
1 = MCU 处于 SLEEP 模式时，使能特定触摸按键唤醒

bit 1 RTCWKEN：RTC 定时唤醒使能位
0 = MCU 处于 SLEEP 模式时，禁止 RTC 定时唤醒
1 = MCU 处于 SLEEP 模式时，使能 RTC 定时唤醒

bit 0 AUTOWKEN：定时自动唤醒使能位
0 = MCU 处于 SLEEP 模式时，禁止定时自动唤醒
1 = MCU 处于 SLEEP 模式时，使能定时自动唤醒

寄存器 5-21： SLPWKDR：SLEEP 唤醒参数配置寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 SLPWKDR<7:0>：SLEEP 唤醒参数配置寄存器

SLPWKDR 寄存器用于设置两类 SLEEP 唤醒时的参数，需与 SLPWKCR 寄存器的 DRSEL<2:0>位配合使用。

5.3.2. 空闲模式(IDLE)

软件配置 PCON=0x01(配置 PCON 前必须先写 PSEQ，具体见 3.1.10 小节“写 PCON 前的保护序列寄存器”)，即进入空闲模式，进入空闲模式后，51 内核停止工作，但是 OSC48M、OSC800K、外部 32.768K 晶振(若外接了 32.768K 晶振)继续工作。

在空闲模式下，以下模块会停止工作：

- WDT
- ADC
- I^2C Master

用户可以通过以下方法从空闲模式唤醒：

- Timer0 中断
- Timer1 中断
- Timer2 中断
- Timer3 中断 0
- Timer3 中断 1
- sensorADC 中断
- RTC 中断
- SPI 中断
- I2C Slave 中断
- UART 中断
- 外部中断 0、1、2、3、4、5

5.4. 静电防护

CBM73xx 系列芯片内部带有 ESD 检测电路，当发生 ESD 事件时，MCU 会采取特殊处理措施，增强芯片的 ESD 性能。

5.5. 代码保护

当用户编程时写入了 4 字节的编程秘钥，会对用户代码进行加密。加密后，对芯片的读操作将会进行保护。

5.6. 在线串行编程

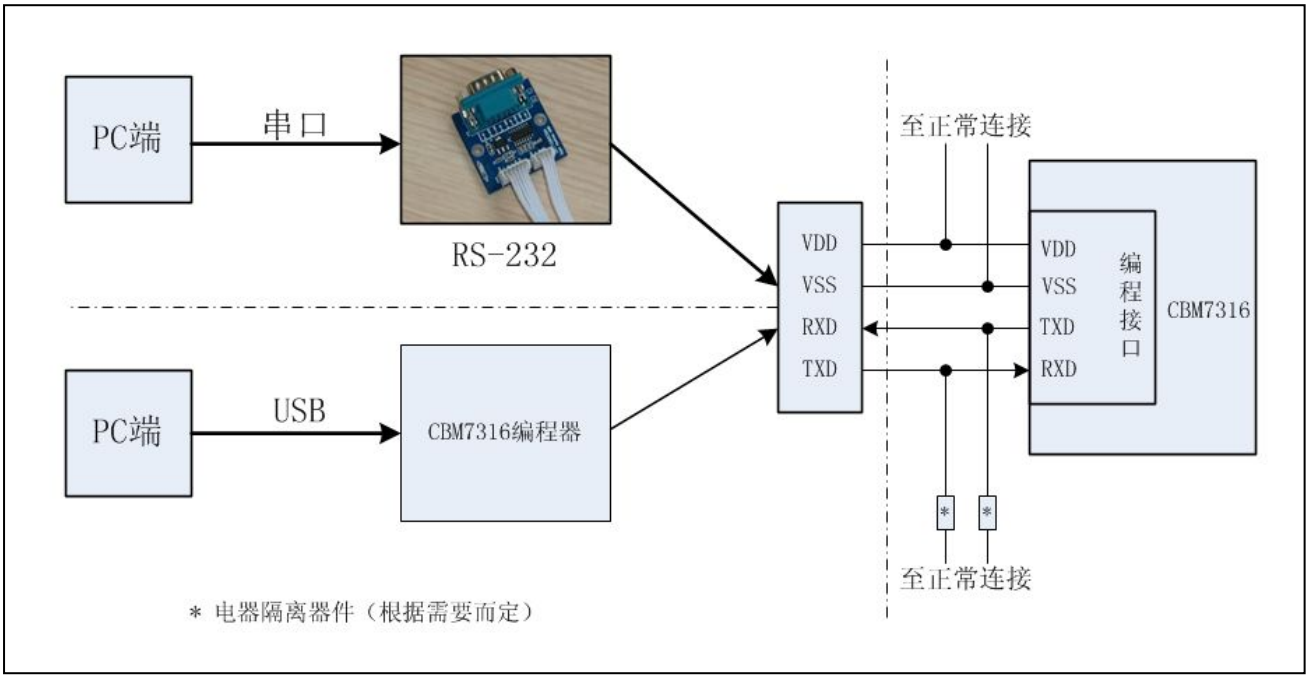
可以在最终应用电路中对 CBM73xx 系列单片机进行串行编程。编程可以简单地通过以下 4 种线完成：

- 电源线(VDD)
- 接地线(VSS)
- 数据发送线(TXD)
- 数据接收线(RXD)

这使用户可使用未编程的芯片制造电路板，而仅在产品交付前才对单片机进行编程。从而可以将最新版本的软件或者定制软件烧写到单片机中。

图 5-6 给出了典型的在线串行编程的连接方式。

图 5-6： 典型的在线串行编程连接方式



5.7. 在线调试

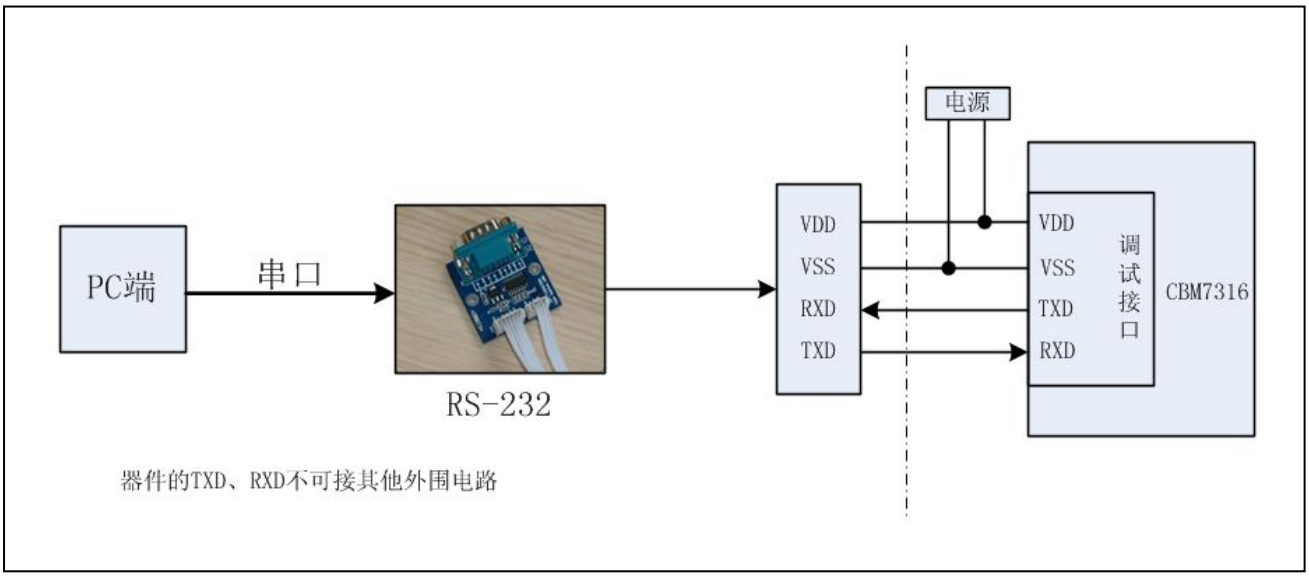
CBM73xx 系列芯片具有片上在线调试器电路，以及用于在线调试器的引脚，芯片被固定在目标应用板上，有 4 根导线 TX、RX、VDD、GND 通过 RS232 与 PC 端连接进行调试。

当单片机使用了在线调试功能后，某些资源将不再具有常规功能。

典型的在线调试连接方式如图 5-7 所示。

注： 用户必须具有支持在线调试的电路，一旦使能在线调试功能，TX、RX 引脚将只能作为调试串口，不能用作其他功能。在线调试电路使用这两个引脚与 PC 进行通信。

图 5-7： 典型的在线调试连接方式



6. I/O 端口

I/O 端口模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
P0	0x80	SFR	P0口数据寄存器	1111 1111
P1	0x90	SFR	P1口数据寄存器	1111 1111
P2	0xA0	SFR	P2口数据寄存器	1111 1111
P3	0xB0	SFR	P3口数据寄存器	1111 1111
P4	0xC0	SFR	P4口数据寄存器	1111 1111
P00IOCFG	0x1000	XRAM	P0.0口功能控制寄存器	0000 0001
P01IOCFG	0x1001	XRAM	P0.1口功能控制寄存器	0011 0000
P02IOCFG	0x1002	XRAM	P0.2口功能控制寄存器	0000 0000
P03IOCFG	0x1003	XRAM	P0.3口功能控制寄存器	0000 0000
P04IOCFG	0x1004	XRAM	P0.4口功能控制寄存器	0000 0000
P05IOCFG	0x1005	XRAM	P0.5口功能控制寄存器	0000 0000
P06IOCFG	0x1006	XRAM	P0.6口功能控制寄存器	0000 0000
P07IOCFG	0x1007	XRAM	P0.7口功能控制寄存器	0000 0000
P10IOCFG	0x1008	XRAM	P1.0口功能控制寄存器	0000 0000
P11IOCFG	0x1009	XRAM	P1.1口功能控制寄存器	0000 0000
P12IOCFG	0x100A	XRAM	P1.2口功能控制寄存器	0000 0000
P13IOCFG	0x100B	XRAM	P1.3口功能控制寄存器	0000 0000
P14IOCFG	0x100C	XRAM	P1.4口功能控制寄存器	0000 0000
P15IOCFG	0x100D	XRAM	P1.5口功能控制寄存器	0000 0000
P16IOCFG	0x100E	XRAM	P1.6口功能控制寄存器	0000 0000
P17IOCFG	0x100F	XRAM	P1.7口功能控制寄存器	0000 0000
P20IOCFG	0x1010	XRAM	P2.0口功能控制寄存器	0000 0000
P21IOCFG	0x1011	XRAM	P2.1口功能控制寄存器	0000 0000
P22IOCFG	0x1012	XRAM	P2.2口功能控制寄存器	0000 0000
P23IOCFG	0x1013	XRAM	P2.3口功能控制寄存器	0000 0000
P24IOCFG	0x1014	XRAM	P2.4口功能控制寄存器	0000 0000
P25IOCFG	0x1015	XRAM	P2.5口功能控制寄存器	0000 0000
P26IOCFG	0x1016	XRAM	P2.6口功能控制寄存器	0000 0000
P27IOCFG	0x1017	XRAM	P2.7口功能控制寄存器	0000 0000
P30IOCFG	0x1018	XRAM	P3.0口功能控制寄存器	0000 0000
P31IOCFG	0x1019	XRAM	P3.1口功能控制寄存器	0000 0000
P32IOCFG	0x101A	XRAM	P3.2口功能控制寄存器	0000 0000
P33IOCFG	0x101B	XRAM	P3.3口功能控制寄存器	0000 0000
P34IOCFG	0x101C	XRAM	P3.4口功能控制寄存器	0000 0000
P35IOCFG	0x101D	XRAM	P3.5口功能控制寄存器	0000 0000
P36IOCFG	0x101E	XRAM	P3.6口功能控制寄存器	0000 0000
P37IOCFG	0x101F	XRAM	P3.7口功能控制寄存器	0000 0000

I/O 端口模块相关寄存器列表(续)

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
P40IOCFG	0x1020	XRAM	P4.0口功能控制寄存器	0000 0000
P41IOCFG	0x1021	XRAM	P4.1口功能控制寄存器	0000 0000
P42IOCFG	0x1022	XRAM	P4.2口功能控制寄存器	0000 0000
P43IOCFG	0x1023	XRAM	P4.3口功能控制寄存器	0000 0000
P44IOCFG	0x1024	XRAM	P4.4口功能控制寄存器	0000 0000
P45IOCFG	0x1025	XRAM	P4.5口功能控制寄存器	0000 0000
P46IOCFG	0x1026	XRAM	P4.6口功能控制寄存器	0000 0000
P47IOCFG	0x1027	XRAM	P4.7口功能控制寄存器	0000 0000
P0PD	0x20FF	XRAM	P0口内部下拉使能寄存器	0000 0000
P1PD	0x21FF	XRAM	P1口内部下拉使能寄存器	0000 0000
P2PD	0x22FF	XRAM	P2口内部下拉使能寄存器	0000 0000
P3PD	0x23FF	XRAM	P3口内部下拉使能寄存器	0000 0000
P4PD	0x24FF	XRAM	P4口内部下拉使能寄存器	0000 0000
P0PU	0x25FF	XRAM	P0口内部上拉使能寄存器	1111 1111
P1PU	0x26FF	XRAM	P1口内部上拉使能寄存器	1111 1111
P2PU	0x27FF	XRAM	P2口内部上拉使能寄存器	1111 1111
P3PU	0x28FF	XRAM	P3口内部上拉使能寄存器	1111 1111
P4PU	0x29FF	XRAM	P4口内部上拉使能寄存器	1111 1111
P0OE	0x2AFF	XRAM	P0口 GPIO 方向选择寄存器	1111 1111
P1OE	0x2BFF	XRAM	P1口 GPIO 方向选择寄存器	1111 1111
P2OE	0x2CFF	XRAM	P2口 GPIO 方向选择寄存器	1111 1111
P3OE	0x2DFF	XRAM	P3口 GPIO 方向选择寄存器	1111 1111
P4OE	0x2EFF	XRAM	P4口 GPIO 方向选择寄存器	1111 1111
P0PURSELRL	0x30FF	XRAM	P0口内部上拉电阻选择寄存器	0000 0000
P0PURSELRH	0x31FF	XRAM	P0口内部上拉电阻选择寄存器	0000 0000
P1PURSELRL	0x32FF	XRAM	P1口内部上拉电阻选择寄存器	0000 0000
P1PURSELRH	0x33FF	XRAM	P1口内部上拉电阻选择寄存器	0000 0000
P2PURSELRL	0x34FF	XRAM	P2口内部上拉电阻选择寄存器	0000 0000
P2PURSELRH	0x35FF	XRAM	P2口内部上拉电阻选择寄存器	0000 0000
P3PURSELRL	0x36FF	XRAM	P3口内部上拉电阻选择寄存器	0000 0000
P3PURSELRH	0x37FF	XRAM	P3口内部上拉电阻选择寄存器	0000 0000
P4PURSELRL	0x38FF	XRAM	P4口内部上拉电阻选择寄存器	0000 0000
P4PURSELRH	0x39FF	XRAM	P4口内部上拉电阻选择寄存器	0000 0000
P0OD	0x3AFF	XRAM	P0口开漏输出使能寄存器	0000 0000
P1OD	0x3BFF	XRAM	P1口开漏输出使能寄存器	0000 0000
P2OD	0x3CFF	XRAM	P2口开漏输出使能寄存器	0000 0000
P3OD	0x3DFF	XRAM	P3口开漏输出使能寄存器	0000 0000
P4OD	0x3EFF	XRAM	P4口开漏输出使能寄存器	0000 0000

I/O 端口模块相关寄存器列表(续)

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
P0LDRVRL	0x40FF	XRAM	P0口驱动电流选择寄存器	0000 0000
P0LDRVRH	0x41FF	XRAM	P0口驱动电流选择寄存器	0000 0000
P1LDRVRL	0x42FF	XRAM	P1口驱动电流选择寄存器	0000 0000
P1LDRVRH	0x43FF	XRAM	P1口驱动电流选择寄存器	0000 0000
P2LDRVRL	0x44FF	XRAM	P2口驱动电流选择寄存器	0000 0000
P2LDRVRH	0x45FF	XRAM	P2口驱动电流选择寄存器	0000 0000
P3LDRVRL	0x46FF	XRAM	P3口驱动电流选择寄存器	0000 0000
P3LDRVRH	0x47FF	XRAM	P3口驱动电流选择寄存器	0000 0000
P4LDRVRL	0x48FF	XRAM	P4口驱动电流选择寄存器	0000 0000
P4LDRVRH	0x49FF	XRAM	P4口驱动电流选择寄存器	0000 0000

CBM73xx 系列芯片最多有 5 组 8 位宽共 40 个通用 I/O 引脚可供使用。

6.1. 引脚配置为 GPIO 的操作步骤

引脚配置为 GPIO 的操作步骤如下：

1. GPIO 模块时钟使能：
   将 MDLCKCR 寄存器的 IOCEN 位置 1，使能 GPIO 模块时钟。

2. 功能复用配置：
   配置对应引脚的 PxyIOCFG=0x00，将引脚配置为 GPIO 功能。

3. 数据寄存器配置：
   配置数据寄存器 Pn，选择输出电平。

4. 驱动电流大小选择：
   配置驱动电流选择寄存器 PnLDRVRH：PnLDRVRL，选择驱动电流大小。默认 00 = 8mA。

5. 内部上拉电阻大小选择：
   配置 PnPURSELRH：PnPURSELRL 寄存器，选择内部上拉电阻大小。

6. 内部上拉使能配置：
   配置 PnPU 寄存器，选择是否使能内部上拉。

7. 内部下拉配置：
   配置 PnPD 寄存器，选择是否使能内部下拉。

8. 开漏输出配置：
   配置 PnOD 寄存器，选择是否使能开漏输出。

9. 方向选择：
   配置 PnOE 寄存器，选择 GPIO 方向。

特别需要注意的是：方向选择、内部上拉、内部下拉、开漏输出这些配置是独立的，也就是说内部上拉、内部下拉、开漏输出是可以同时使能的，而且不管 GPIO 的方向是输出还是输入，所以需要根据实际情况来进行配置。

例 5-1 说明了如何初始化 P0.0、P1.1 口，其他 IO 口初始化也一样。

例 5-1： 初始化 P0.0 - 输出，P1.0 - 上拉输入

```
MDLCKCR_IOCEN = 0x08;   //使能 GPIO 模块时钟
P00IOCFG = 0x00;        //初始化 P0.0 为 GPIO
P10IOCFG = 0x00;        //初始化 P1.0 为 GPIO
P0 = 0x00;              //初始化 P0
P1 = 0x00;              //初始化 P1
P0LDRVRL_0 = 0x00;      //配置 P0.0 驱动电流
P1PURSELRL_0 = 0x00;    //配置 P1.0 上拉电阻
P0PD = 0x00;            //关闭 P0.0 内部下拉
P0PU = 0x00;            //关闭 P0.0 内部上拉
P1PD = 0x00;            //关闭 P1.0 内部下拉
P1PU = 0x01;            //使能 P1.0 内部上拉
P00D = 0x00;            //关闭 P0.0 开漏输出
P10D = 0x00;            //关闭 P1.0 开漏输出
P00E = 0x00;            //P0.0 输出
P10E = 0x01;            //P1.0 输入
```

6.2. 功能复用

PxyIOCFG(x = 0、1、2、3、4，y = 0、1、2、3、4、5、6、7)寄存器是 I/O 功能控制寄存器，配置不同的值，引脚可复用为不同的功能。

寄存器 6-1： PxyIOCFG：IO 功能控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—						
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未实现：读为 0

bit 5-0 PxyIOCFG：IO 功能控制寄存器

00 0000 = GPIO

00 0001 = 调试串口输出。P0.0 默认为调试串口输出，其他引脚不可以配置为调试串口输出

00 0010 = 定时器 1 的输出或输入

00 0011 = 定时器 3 的输出或输入 0

00 0100 = I2C Slave 数据线

00 0101 = I2C Master 时钟线

00 0110 = I2C Master 数据线

00 0111 = 未定义

00 1000 = sensor 通道，除 P0.6、P0.7、P1.0、P1.1、P1.2、P1.3、P1.4、P1.5 外的所有引脚都可配置为 sensor 通道

00 1001 = ADC 通道，仅 P0.6、P0.7、P1.0、P1.1、P1.2、P1.3、P1.4、P1.5 可配置为 ADC 通道

00 1010 = 未定义

……

00 1111 = 未定义

01 0000 = SPI Slave 数据输出

01 0001 = SPI Master 片选输出

01 0010 = SPI Master 时钟输出

01 0011 = SPI Master 数据输出

01 0100 = LED SEG1

01 0101 = LED SEG2

01 0110 = LED SEG3

01 0111 = LED SEG4

01 1000 = LED SEG5

01 1001 = LED SEG6

01 1010 = LED SEG7

01 1011 = LED SEG8

01 1100 = LED SEG9
01 1101 = LED SEG10_GRID8
01 1110 = LED SEG11_GRID7
01 1111 = LED SEG12_GRID6
10 0000 = LED SEG13_GRID5
10 0001 = LED GRID1
10 0010 = LED GRID2
10 0011 = LED GRID3
10 0100 = LED GRID4
10 0101 = 定时器 0 输出
10 0110 = 定时器 2 输出
10 0111 = 通用串口输出
10 1000 = PWM1 输出
10 1001 = PWM2 输出
10 1010 = 未定义
……
10 1111 = 未定义
11 0000 = 调试串口输入。P0.1 默认为调试串口输入，其他引脚不可配置为调试串口输入
11 0001 = I2C Slave 时钟线
11 0010 = 定时器 3 输入 1
11 0011 = 通用串口输入
11 0100 = SPI Slave 片选输入
11 0101 = SPI Slave 时钟输入
11 0110 = SPI Slave 数据输入
11 0111 = SPI Master 数据输入
11 1000 = 外部中断 0
11 1001 = 外部中断 1
11 1010 = 外部中断 2
11 1011 = 外部中断 3
11 1100 = 外部中断 4
11 1101 = 外部中断 5
11 1110 = 未定义
11 1111 = 未定义

注： 这类寄存器不可以位操作。

6.3. 方向选择

PnOE(n = 0、1、2、3、4)寄存器是 GPIO 方向选择寄存器，对应位置 1 为输入，清 0 为输出。复位时默认为输入。

Pn(n = 0、1、2、3、4)寄存器是片内特殊功能寄存器(SFR)，是端口数据寄存器。当 I/O 引脚配置为 GPIO，PnOE 配置为输入时，读 Pn 寄存器即读出端口引脚电平，PnOE 配置为输出时，写 Pn 寄存器即将数据送到端口。

注： 1、复位时，所有 GPIO 都默认为输入口。
2、当 IO 配置为定时器 1 的输出或输入、定时器 3 的输出或输入 0 功能时，需要配置对应的 PnOE 寄存器选择方向，其他功能则不需要配置 PnOE 寄存器。

寄存器 6-2： PnOE：GPIO 方向选择寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 n<7:0>：GPIO 方向选择位
0 = 输出模式
1 = 输入模式

寄存器 6-3： Pn：GPIO 数据寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 n<7:0>：GPIO 数据位
0 = 端口引脚电平为低电平
1 = 端口引脚电平为高电平

6.4. 内部上拉

PnPU(n = 0、1、2、3、4)寄存器是GPIO内部上拉使能寄存器，对应位置1为使能内部上拉，清0为禁止内部上拉。复位默认使能内部上拉。

内部上拉电阻还可以配置不同的上拉电阻值。由PnPURSELRH、PnPURSELRL(n = 0、1、2、3、4)寄存器控制。PnPURSELRH控制高4位，PnPURSELRL控制低4位，有4个档位可以选择，默认档位是00 = 88K。

注： 1、需要注意的是方向选择、内部上拉、内部下拉、开漏输出、驱动电流的配置都是独立的，所以配置其中一项时需要注意其他项是否会产生影响。
2、推荐做法是每次对这些都同时进行配置，防止忽略其中某项配置而产生影响。

寄存器6-4： PnPU：GPIO内部上拉使能寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = 上电复位时的值	1 = 置1	0 = 清零	x = 未知

bit 7-0 n<7:0>：GPIO内部上拉使能位
0 = 禁止内部上拉
1 = 使能内部上拉

寄存器6-5： PnPURSELRH：GPIO内部上拉电阻选择寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7		6		5		4	
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

寄存器6-5： PnPURSELRL：GPIO内部上拉电阻选择寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
3		2		1		0	
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = 上电复位时的值	1 = 置1	0 = 清零	x = 未知

bit (2x+1)-2x n<1:0>：GPIO内部上拉电阻选择位
00 = 88K
01 = 10K
10 = 4.5K
11 = 2.2K

6.5. 内部下拉

PnPD(n = 0、1、2、3、4)寄存器是 GPIO 内部下拉使能寄存器，对应位置 1 为使能内部下拉，清 0 为禁止内部下拉。复位时默认禁止内部下拉。

寄存器 6-6： PnPD：GPIO 内部下拉使能寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 n<7:0>：GPIO 内部下拉使能位
0 = 禁止内部下拉
1 = 使能内部下拉

6.6. 开漏输出

PnOD(n = 0、1、2、3、4)寄存器是 GPIO 开漏输出使能寄存器，对应位置 1 为使能开漏输出，清 0 为禁止开漏输出。复位时默认禁止开漏输出。

寄存器 6-7： PnOD：GPIO 内部下拉使能寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 n<7:0>：GPIO 开漏输出使能位
0 = 禁止开漏输出
1 = 使能开漏输出

6.7. 驱动电流

当 I/O 配置为 GPIO 输出时，可以配置 PnLDRVRH、PnLDRVRL(n = 0、1、2、3、4)寄存器选择不同的驱动能力，满足不同驱动能力的要求。其中 PnLDRVRH 控制高 4 位，PnLDRVRL 控制低 4 位，有 4 个档位可以选择，默认档位是 00 = 8mA。

注： 当 I/O 配置为 LED 功能的 SEG/GRID 口时，也可以配置 PnLDRVRH、PnLDRVRL(n = 0、1、2、3、4)寄存器，实现更多的灰度选择。

寄存器 6-8： PnLDRVRH：GPIO 驱动电流选择寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7		6		5		4	
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

寄存器 6-8： PnLDRVRL：GPIO 驱动电流选择寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
3		2		1		0	
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit (2x+1)-2x　　n<1:0>：GPIO 驱动电流选择位

00 = 8mA

01 = 40mA

10 = 100mA

11 = 8mA

7. 看门狗定时器(WDT)

WDT 相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	复位值
WDTCR	0x60FF	XRAM	清看门狗寄存器	0000 0000
WDTSETR	0x61FF	XRAM	看门狗定时器控制寄存器	0001 0011
WDTSR	0x62FF	XRAM	看门狗溢出中断状态寄存器	0000 0000

WDT 具有以下特性：

- 以 OSC48M 经 CKDIVCR 寄存器的 SYSDIV<5:0>位分频后的时钟作为工作时基
- 若配置 CKDIVCR 寄存器的 SYSDIV<5:0>位等于 2，则 WDT 溢出周期的范围是 64us~131ms，此时溢出复位时间的范围是 128us~262ms。
- 8*16 阶可配置周期

使能 WDT 的时钟(MDLCKCR 寄存器的 WDTEN 位)和 WDT 计数器计数使能(WDTSETR 寄存器的 CNTEN 位)，WDT 计数器就开始计数。WDTSETR 寄存器可以配置 WDT 计数器溢出时间。

计数器递增过程中，软件对 WDTCR 寄存器依次写 5A、A5，可清零计数器，计数器从 0 重新开始计数，注意写 WDTCR 的顺序不能变。

WDT 工作时序如图 7-1 所示。

注：1、WDT 计数器第 1 次溢出时，会产生中断，若第 2 次溢出前不清 WDT 计数器，则第 2 次溢出时，会使芯片复位。
2、WDT 模块中没有中断使能位，只要依次将 IRQOER 寄存器的 WDTINT 位、IE 寄存器的 EX0 位、IE 寄存器的 EA 位置 1，就可以使能 WDT 溢出中断。
3、WDT 在休眠、空闲模式下会停止工作。

7.1. WDT 的操作步骤

配置 WDT 的操作步骤如下：

1. WDT 模块时钟使能：

   将 MDLCKCR 寄存器的 WDTEN 位置 1，使能 WDT 模块时钟。

2. 清 WDT：

   配置 WDT 开始工作前，先对 WDTCR 寄存器依次 5A、A5，清 WDT。

3. 配置 WDT 溢出周期：

   - 计数时钟分频选择：

   WDTSETR 寄存器的 CNTDIV<2:0>位可以对 WDT 计数时钟进行分频。

   - 计数长度选择：

   WDTSETR 寄存器的 CNTLEN<3:0>位可以选择 WDT 计数长度。

   WDT 溢出周期 = 2^{CNTDIV} × (CNTLEN + 1) × 1024 × (SYSDIV + 1) × Tosc48M

   其中 Tosc48M 是 OSC48M 的时钟周期

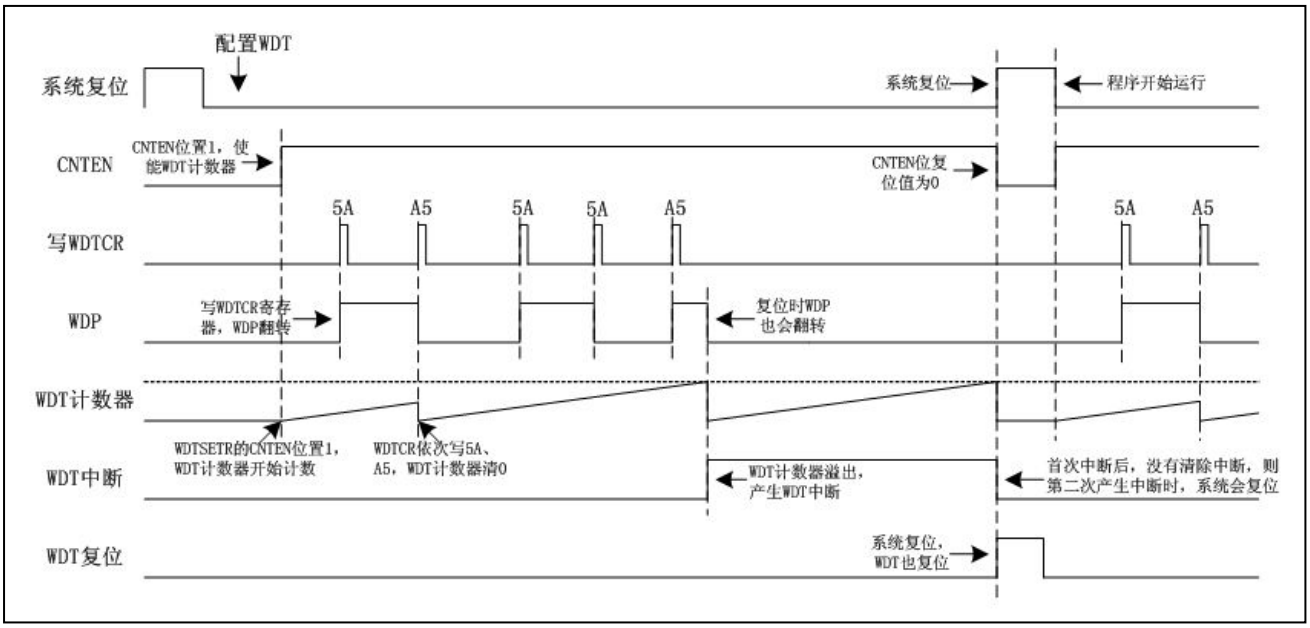
4. 中断控制：

   WDT 模块中没有中断使能位，只要依次将 IRQOER 寄存器的 WDTINT 位、IE 寄存器的 EX0 位、IE 寄存器的 EA 位置 1，就可以使能 WDT 溢出中断。

5. WDT 计数器计数使能：

   将 WDTSETR 寄存器的 CNTEN 位置 1，WDT 计数器开始计数。

图 7-1： 看门狗定时器工作时序图



7.2. WDT 寄存器的定义

以下寄存器用于控制 WDT 的操作。

寄存器 7-1： WDTCR：清看门狗寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
WDPC6	WDPC5	WDPC4	WDPC3	WDPC2	WDPC1	WDPC0	WDP
W-0	W-0	W-0	W-0	W-0	W-0	W-0	R-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-1 WDPC<6:0>：
只写属性，读为 0

bit 0 WDP：
WDP 是只读位，当软件写 WDTCON 时，WDP 电平会翻转。
若 WDT 计数器溢出或改变 WDTMOD，则 WDP 会恢复到 0。

注： 对 WDTCR 寄存器依次 5A、A5(顺序不能变)，可以清 WDT(喂狗)。

寄存器 7-2： WDTSETR：看门狗定时器控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTEN	CNTDIV2	CNTDIV1	CNTDIV0	CNTLEN3	CNTLEN2	CNTLEN1	CNTLEN0
R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 CNTEN：WDT 计数器计数使能位
0 = 禁止 WDT 计数器计数
1 = 使能 WDT 计数器计数

bit 6-4 CNTDIV<2:0>：WDT 计数时钟分频选择位
000 = 1 分频
001 = 2 分频
010 = 4 分频
011 = 8 分频
100 = 16 分频
101 = 32 分频
110 = 64 分频
111 = 128 分频

bit 3-0 CNTLEN<3:0>：WDT 计数长度选择位
0000 = 1024
0001 = 2048
0010 = 3072
0011 = 4096
……
1100 = 13312
1101 = 14336
1110 = 15360
1111 = 16384

注： WDT 溢出复位时间 = $2 \times 2^{CNTDIV} \times (CNTLEN + 1) \times 1024 \times (SYSDIV + 1) \times Tosc48M$

Tosc48M 是 OSC48M 的时钟周期。

寄存器 7-3： WDTSR：看门狗溢出中断状态寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	—	—	WDTINT
U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-1 未实现：读为 0

bit 0 WDTINT：看门狗溢出中断状态位

0 = 未产生看门狗溢出

1 = 已产生看门狗溢出

注： 虽然 WDT 可以产生中断，但清除 WDT 计数器的行为(即“喂狗”)不能放在 WDT 的中断服务程序中，需穿插在用户的程序函数中。(因程序跑飞时，CPU 仍然能正常响应中断，达不到恢复程序的目的)

7.3. WDT 溢出中断

WDT 溢出中断没有单独的使能位，只要将 IRQOER 寄存器的 WDTINT 位置 1，就使能 WDT 溢出中断，WDTSR 寄存器的 WDTINT 位是对应的中断标志位。无论 WDT 溢出中断是否使能，WDT 计数器首次溢出时，WDT 溢出中断标志位都会置 1。

当 WDT 计数器首次溢出时，会产生中断，WDTSR 寄存器的 WDTINT 位也会置 1。如果软件不清除中断，当 WDT 计数器下次溢出时，则会复位系统。清除 WDTINT 的方法是，对 WDTCON 寄存器依次写 5A，A5。不能多写，也不能打乱顺序。

8. 8 位定时器 Timer0

Timer0 模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
T0CKCR	0x70FF	XRAM	Timer0时钟控制寄存器	0000 0000
T0CR	0x71FF	XRAM	Timer0模式控制寄存器	0000 0000
T0CMP0	0x72FF	XRAM	Timer0比较寄存器0	0000 0000
T0CMP1	0x73FF	XRAM	Timer0比较寄存器1	0000 0000
T0SR	0x74FF	XRAM	Timer0状态寄存器	0000 0010

Timer0 模块是一个 8 位定时器，具有如下特点：

- 8 位定时器
- 3 位预分频器
- 8 位周期寄存器(T0CMP0)
- 8 位 PWM 脉宽调制寄存器(T0CMP1)
- 可编程内部高速或低速时钟源
- 可编程内部或外部时钟源

8.1. Timer0 支持的功能

Timer0 支持以下 3 种功能：

- 间隔定时
- 方波输出
- PWM 输出

8.1.1. 间隔定时的操作步骤

配置 Timer0 为间隔定时功能的操作步骤如下：

1. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

2. Timer0 计数时钟参数配置：

- 计数时钟选择：

TOCKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer0 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer0 的时钟。

- 计数时钟分频配置：

TOCKCR 寄存器的 CLKDIV<2:0>位可以对选择的时钟源进行分频。

- 计数时钟使能：

将 TOCKCR 寄存器的 CLKEN 位置 1，使能 Timer0 计数时钟。

3. 工作模式选择：

Timer0 有定时器和 PWM 两种工作模式，通过 TOCR 寄存器的 MODE 位控制，这两种模式都可以实现间隔定时功能。

4. 间隔时间配置：

Timer0 的间隔定时时间是通过 TOCMP0 寄存器配置，具体的间隔时间计算如下：

间隔时间 = (TOCMP0)x Timer0 时钟周期

= (TOCMP0)x Timer0 计数时钟分频 x Timer0 计数时钟周期

Timer0 计数时钟周期由选择的计数时钟决定，共有三种时钟：

- 内部高速 OSC48M 的 3 分频
- 内部低速 800K
- 外部 32.768K 晶振(必须确保外接了 32.768K 晶振)

注： TOCMP0 在 Timer0 工作期间是不能改变的，所以想要改变 TOCMP0 必须先把 TOCR 寄存器的 TEN 位清零，等待 TOSR 寄存器的 WREN 位置 1 后，再修改 TOCMP0。

5. 清 Timer0 中断标志位

使能 Timer0 中断前，先将 TOSR 寄存器的 INTSR 位清 0，清 Timer0 中断标志位。

6. 中断控制：

Timer0 只有一种中断，就是当 Timer0 计数器值等于(TOCMP0-1)时，产生 Timer0 中断。在间隔定时、方波输出、PWM 输出三种模式下都可以产生 Timer0 中断。

TOCR 寄存器的 INTEN 位是 Timer0 中断使能位，TOSR 寄存器的 INTSR 位是 Timer0 中断标志位。

7. Timer0 使能：

将 TOCR 寄存器的 TEN 位置 1,使能 Timer0,Timer0 即开始工作。

8.1.2. 方波输出的操作步骤

配置 Timer0 为方波输出功能的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 0 输出功能时，会自动将引脚方向配置为输出。不需要配置对应的 PnOE 寄存器，选择定时器 0 输出引脚方向为输出。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer0 计数时钟参数配置：

- 计数时钟选择：

TOCKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer0 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer0 的时钟。

- 计数时钟分频配置：

TOCKCR 寄存器的 CLKDIV<2:0>位可以对选择的时钟源进行分频。

- 计数时钟使能：

将 TOCKCR 寄存器的 CLKEN 位置 1，使能 Timer0 计数时钟。

4. Timer0 模式配置：

- 工作模式选择：

TOCR 寄存器的 MODE 位清 0，将工作模式选择为定时器模式。

- 输出反转选择：

TOCR 寄存器的 OUTSEL 位是 Timer0 输出反转使能位，对于方波输出和 PWM 输出都有效，OUTSEL 位置 1 后输出波形会发生反转。

5. 方波周期配置：

Timer0 的方波周期也是通过 TOCMP0 寄存器配置，方波周期其实就是间隔时间的两倍，具体计算如下：

方波周期 = 2 × 间隔时间

= 2 × (TOCMP0) x Timer0 时钟周期

= 2 × (TOCMP0) x Timer0 计数时钟分频 x Timer0 计数时钟周期

Timer0 计数时钟周期由选择的计数时钟决定，共有三种时钟：

- 内部高速 OSC48M 的 3 分频
- 内部低速 800K
- 外部 32.768K 晶振(必须确保外接了 32.768K 晶振)

> 注： TOCMP0 在 Timer0 工作期间是不能改变的，所以想要改变 TOCMP0 必须先把 TOCR 寄存器的 TEN 位清零，等待 TOSR 寄存器的 WREN 位置 1 后，再修改 TOCMP0。

6. 清 Timer0 中断标志位

使能 Timer0 中断前，先将 TOSR 寄存器的 INTSR 位清 0，清 Timer0 中断标志位。

7. 中断控制：

Timer0 只有一种中断，就是 Timer0 计数器值等于(TOCMP0 - 1)时，产生 Timer0 中断。在间隔定时、方波输出、PWM 输出三种模式下都可以产生 Timer0 中断。

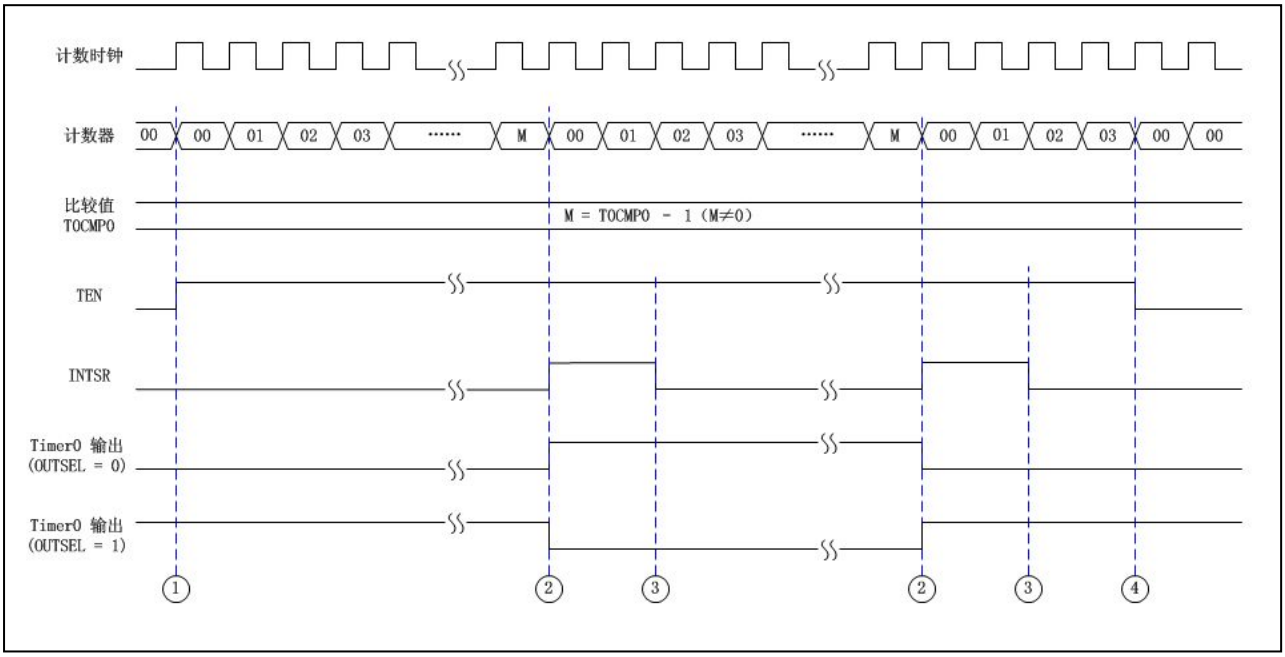
TOCR 寄存器的 INTEN 位是 Timer0 中断使能位，TOSR 寄存器的 INTSR 位是 Timer0 中断标志位。

8. Timer0 使能：

将 TOCR 寄存器的 TEN 位置 1，使能 Timer0，Timer0 即开始工作。

Timer0 间隔定时/方波输出工作时序如图 8-1 所示。

图 8-1： Timer0 间隔定时/方波输出工作时序图(TOCMP0 ≠ 1)



波形说明：

① 使能定时器，计数器开始计数；

② 计数器值等于 M，产生 Timer0 中断(即 TOSR 寄存器的 INTSR 位置 1)，同时 Timer0 输出翻转；

③ 软件处理并清除中断；

④ 不使能定时器，计数器清 0 并停止计数。

间隔时间 = (TOCMP0) x Timer0 时钟周期

= (TOCMP0) x Timer0 计数时钟分频 x Timer0 计数时钟周期

方波周期 = 2 x 间隔时间

注： 1、当 TOCMP0 = 0 时，按 TOCMP0 = 256 计算。

2、当 TOCMP0 = 1 时，没有方波输出，也不产生 Timer0 中断。

8.1.3. PWM 输出的操作步骤

配置 Timer0 为 PWM 输出功能的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 0 输出功能时，会自动将引脚方向配置输出。不需要配置对应的 PnOE 寄存器，选择定时器 0 输出引脚方向为输出。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer0 计数时钟参数配置：

- 计数时钟选择：

TOCKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer0 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer0 的时钟。

- 计数时钟分频配置：

TOCKCR 寄存器的 CLKDIV<2:0>位可以对选择的时钟源进行分频。

- 计数时钟使能：

将 TOCKCR 寄存器的 CLKEN 位置 1，使能 Timer0 计数时钟。

4. Timer0 模式配置：

- 工作模式选择：

将 TOCR 寄存器的 MODE 位置 1，工作模式选择为 PWM 模式。

- 输出反转选择：

TOCR 寄存器的 OUTSEL 位是 Timer0 输出反转使能位，对于方波输出和 PWM 输出都有效，OUTSEL 位置 1 后输出波形会发生反转。

5. PWM 占空比配置：

Timer0 的 PWM 占空比是通过 TOCMP0 与 TOCMP1 寄存器进行配置，具体计算如下：

PWM 周期 = (TOCMP0) x Timer0 时钟周期

= (TOCMP0) x Timer0 计数时钟分频 x Timer0 计数时钟周期

Timer0 计数时钟周期由选择的计数时钟决定，共有三种时钟：

- 内部高速 OSC48M 的 3 分频
- 内部低速 800K
- 外部 32.768K 晶振(必须确保外接了 32.768K 晶振)

PWM 占空比 = TOCMP1 ： TOCMP0

当 TOCMP0 = 0 时，按 TOCMP0 = 256 计算

简单地说，TOCMP0 是配置 PWM 输出的周期，TOCMP1 配置的是输出低电平的时间，所以 TOCMP1 必须小于 TOCMP0。(当输出反转位置 1 时，TOCMP1 配置的是输出高电平的时间)

在 PWM 模式下，可以直接修改 TOCMP1 寄存器来改变 PWM 的占空比，修改 TOCMP1 时，新的占空比配置将在下个 PWM 周期生效。

> 注： TOCMP0 在 Timer0 工作期间是不能改变的，所以想要改变 TOCMP0 必须先把 TOCR 寄存器的 TEN 位清零，等待 TOSR 寄存器的 WREN 位置 1 后，再修改 TOCMP0。

6. 清 Timer0 中断标志位

使能 Timer0 中断前，先将 TOSR 寄存器的 INTSR 位清 0，清 Timer0 中断标志位。

7. 中断控制：

Timer0 只有一种中断，就是 Timer0 计数器值等于(TOCMP0 - 1)时，产生 Timer0 中断。在间隔定时、方波输出、PWM 输出三种模式下都可以产生 Timer0 中断。

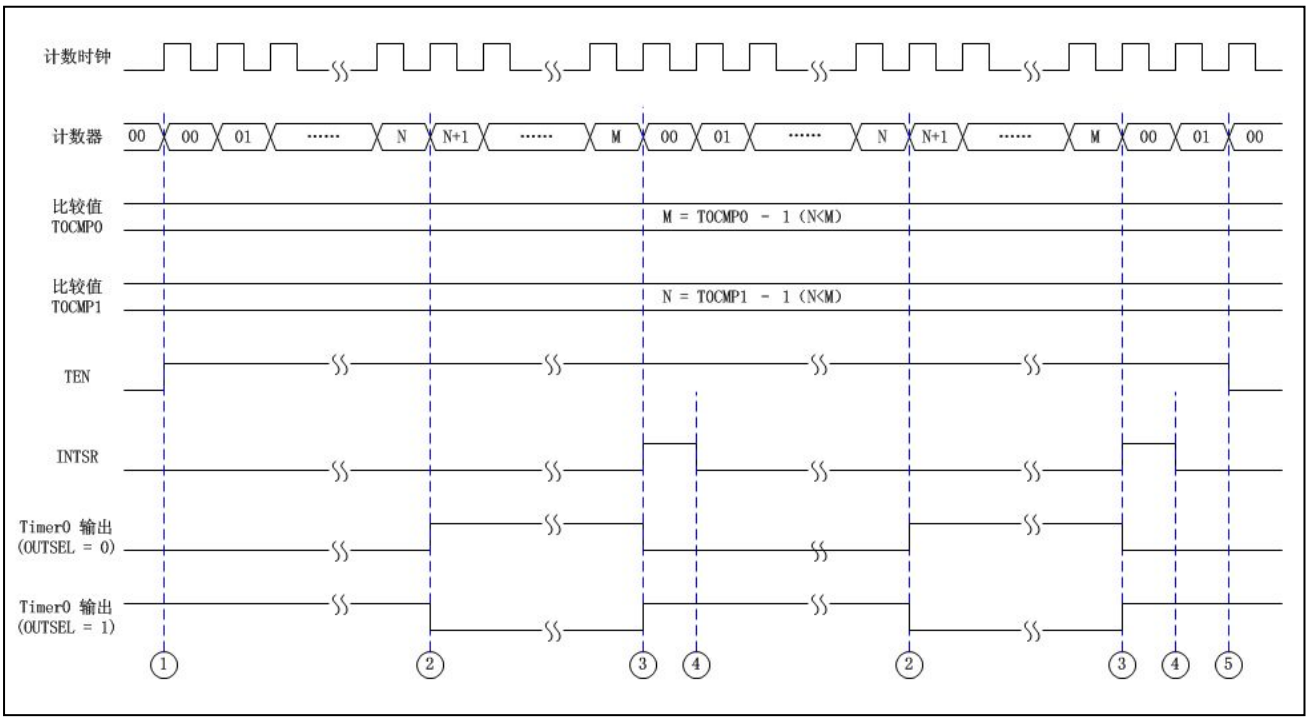
TOCR 寄存器的 INTEN 位是 Timer0 中断使能位，TOSR 寄存器的 INTSR 位是 Timer0 中断标志位。

8. Timer0 使能：

将 TOCR 寄存器的 TEN 位置 1，使能 Timer0，Timer0 即开始工作。

Timer0 PWM 输出工作时序如图 8-2 所示。

图 8-2： Timer0 PWM 输出工作时序图(TOCMP0 ≠ 1,TOCMP1 < TOCMP0)



波形说明：

① 使能定时器，计数器开始计数；

② 计数器值等于 N，Timer0 输出翻转；

③ 计数器值等于 M，产生 Timer0 中断(即 TOSR 寄存器的 INTSR 位置 1)，同时 Timer0 输出翻转；

④ 软件处理并清除中断；

⑤ 不使能定时器，计数器清 0 并停止计数。

PWM 周期 = (TOCMP0)x Timer0 时钟周期

= (TOCMP0) x Timer0 计数时钟分频 x Timer0 计数时钟周期

PWM 占空比 = TOCMP1 ： TOCMP0

注： 1、当 TOCMP0 = 0 时，按 TOCMP0 = 256 计算

2、PWM 模式下，改变 TOCMP1 寄存器后，新的占空比将在下个周期生效，当前 PWM 周期的占空比还是原来的占空比。

8.2. Timer0 寄存器的定义

以下寄存器用于控制 Timer0 的操作。

寄存器 8-1： TOCKCR：Timer0 时钟控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	CLKEN	CLKSEL	—	CLKDIV2	CLKDIV1	CLKDIV0
U-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未实现：读为 0

bit 5 CLKEN：Timer0 计数时钟使能位

0 = 关闭 Timer0 计数时钟

1 = 打开 Timer0 计数时钟

bit 4 CLKSEL：Timer0 计数时钟选择位

0 = 选择内部高速 OSC48M 的 3 分频作为 Timer0 计数时钟

1 = 选择低速时钟作为 Timer0 计数时钟(32KEN 位选择 OSC800K 或外部 32.768K)

bit 3 未实现：读为 0

bit 2-0 CLKDIV：Timer0 计数时钟分频选择位

000 = 2 分频

001 = 4 分频

010 = 8 分频

011 = 16 分频

100 = 32 分频

101 = 64 分频

110 = 128 分频

111 = 1 分频

寄存器 8-2： TOCR：Timer0 模式控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	INTEN	OUTSEL	MODE	TEN
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未实现：读为 0

bit 3 INTEN：Timer0 中断使能位
0 = 禁止计数器与 TOCMP0 匹配中断
1 = 使能计数器与 TOCMP0 匹配中断

bit 2 OUTSEL：Timer0 输出反转使能位
0 = Timer0 输出不反转
1 = Timer0 输出反转

bit 1 MODE：Timer0 工作模式选择位
0 = 定时器模式
1 = PWM 模式

bit 0 TEN：Timer0 使能位
0 = Timer0 停止工作
1 = Timer0 开始工作

寄存器 8-3： TOCMP0：Timer0 比较寄存器 0

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 TOCMP0<7:0>：Timer0 比较寄存器 0
Timer0 比较寄存器 0，在 Timer0 工作期间，不能修改此寄存器

注： TOSR 寄存器的 WREN 位是 TOCMP0 可写状态位，为 1 表示可以修改 TOCMP0 寄存器，为 0 表示不能修改 TOCMP0 寄存器。所以正确的修改 TOCMP0 寄存器如范例 6-1 所示：

范例 6-1：修改 TOCMP0 寄存器：

```
TOCR_TEN = 0x00;                    //Timer0 停止工作
while(0x00 = TOSR_WREN){;}          //等待 TOCMP0 写使能
TOCMP0 = 0x55;                      //修改 TOCMP0 寄存器
……                                  //其他处理
TOCR_TEN = 0x01;                    //Timer0 开始工作
```

寄存器 8-4： TOCMP1：Timer0 比较寄存器 1

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 TOCMP1<7:0>：Timer0 比较寄存器 1

Timer0 比较寄存器 1，在 Timer0 工作期间，可以修改此寄存器。

TOCMP0 决定输出 PWM 的周期，TOCMP1 决定输出 PWM 的占空比，在 Timer0 工作期间，可以修改 TOCMP1 寄存器改变输出 PWM 的占空比。

寄存器 8-5： TOSR：Timer0 状态寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	—	WREN	INTSR
U-0	U-0	U-0	U-0	U-0	U-0	R-1	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-2 未定义：读为 0

bit 1 WREN：TOCMP0 可写标志位

0 = Timer0 正在工作，不可写 TOCMP0

1 = Timer0 停止工作，可写 TOCMP0，关闭 Timer0 后，应等待 WREN 为 1 后，再写 TOCMP0

bit 0 INTSR：Timer0 计数器与 TOCMP0 匹配中断标志位

0 = Timer0 计数器与 TOCMP0 不匹配

1 = 发生了 Timer0 计数器与 TOCMP0 匹配

8.3. Timer0 中断

不论是在定时器模式还是 PWM 模式下，当 Timer0 计数器值等于(TOCMP0 - 1)时，都会产生 Timer0 中断。

只要有 Timer0 中断条件产生，不论是否使能 Timer0 中断，TOSR 寄存器的 INTSR 位(Timer0 中断标志位)都会置 1，INTSR 位必须在软件中清零。Timer0 的中断使能位是 TOCR 寄存器的 INTEN 位。

注： 在休眠模式下，Timer0 中断无法唤醒处理器。
在空闲模式下，Timer0 中断可以唤醒处理器。

9. 8 位定时器 Timer1

Timer1 模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
T1CKCR	0x78FF	XRAM	Timer1时钟控制寄存器	0000 0000
T1CR	0x79FF	XRAM	Timer1模式控制寄存器	0000 0000
T1CMP	0x7AFF	XRAM	Timer1比较寄存器	0000 0000
T1SR	0x7BFF	XRAM	Timer1状态寄存器	0000 0010

Timer1 模块是一个 8 位定时器，具有如下特点：

- 8 位定时器
- 3 位预分频器
- 8 位周期与脉宽调制寄存器(T1CMP)
- 可编程外部时钟边沿选择
- 可编程内部高速或低速时钟源
- 可编程内部或外部时钟源

9.1. Timer1 支持的功能

Timer1 支持以下 4 种功能：

- 间隔定时
- 方波输出
- PWM 输出
- 外部事件计数

9.1.1. 间隔定时的操作步骤

配置 Timer1 为间隔定时功能的操作步骤如下：

1. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

2. Timer1 计数时钟参数配置：

- 计数时钟选择：

T1CKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer1 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer1 的时钟。

- 计数时钟分频配置：

T1CKCR 寄存器的 CLKDIV<3:0>位可以对选择的时钟源进行分频。此时不能配置为外部信号的有效沿选项。

- 计数时钟使能：

将 T1CKCR 寄存器的 CLKEN 位置 1，使能 Timer1 计数时钟。

3. 工作模式选择：

将 T1CR 寄存器的 MODE 位清 0，工作模式选择为定时器模式。

4. 间隔时间配置：

Timer1 的间隔定时时间是通过 T1CMP 寄存器配置，具体的间隔时间计算如下：

间隔时间 = (T1CMP)x Timer1 时钟周期

= (T1CMP)x Timer1 计数时钟分频 x Timer1 计数时钟周期

Timer1 计数时钟周期由选择的计数时钟决定，共有三种时钟：

- 内部高速 OSC48M 的 3 分频
- 内部低速 800K
- 外部 32.768K 晶振(必须确保外接了 32.768K 晶振)

注：1、T1CMP 在定时器模式下，Timer1 工作期间不能修改，所以想要改变 T1CMP 必须先把 T1CR 寄存器的 TEN 位清零，等待 T1SR 寄存器的 WREN 位置 1 后，再修改 T1CMP。

2、T1CMP 在 PWM 模式下，Timer1 工作期间可以直接修改。

5. 清 Timer1 中断标志位

使能 Timer1 中断前，先将 T1SR 寄存器的 INTSR 位清 0，清 Timer1 中断标志位。

6. 中断控制：

Timer1 只有一个中断，但是在间隔定时、方波输出、PWM 输出、外部事件计数模式下，产生 Timer1 中断的条件不同。在间隔定时、方波输出、外部事件计数模式下，当 Timer1 计数器值等于(T1CMP－1)时，会产生 Timer1 中断。在 PWM 输出模式下，当 Timer1 计数器值溢出，即等于 0xFF 时，会产生 Timer1 中断。

T1CR 寄存器的 INTEN 位是 Timer1 中断使能位，T1SR 寄存器的 INTSR 位是 Timer1 中断标志位。

7. Timer1 使能：

将 T1CR 寄存器的 TEN 位置 1，使能 Timer1，Timer1 即开始工作。

9.1.2. 方波输出的操作步骤

配置 Timer1 为方波输出功能的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 1 输出/输入功能，并且配置 PnOE 寄存器，将定时器 1 输出或输入引脚方向置为输出。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer1 计数时钟参数配置：

- 计数时钟选择：

T1CKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer1 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer1 的时钟。

- 计数时钟分频配置：

T1CKCR 寄存器的 CLKDIV<3:0>位可以对选择的时钟源进行分频。此时不能配置为外部信号的有效沿选项。

- 计数时钟使能：

将 T1CKCR 寄存器的 CLKEN 位置 1，使能 Timer1 计数时钟。

4. Timer1 模式配置：

- 输出翻转使能：

T1CR 寄存器的 INVEN 置 1，使能当 Timer1 计数器值等于(T1CMP - 1)时，输出翻转。(不使能没有方波输出)

- 输出初始状态选择：

T1CR 寄存器的 OUTINITV<1:0>位控制 Timer1 输出初始状态，只对方波输出有效，对 PWM 输出无效。

- 工作模式选择：

将 T1CR 寄存器的 MODE 位清 0，工作模式选择为定时器模式。

5. 方波周期配置：

方波周期是通过 T1CMP 寄存器配置，Timer1 的方波周期其实就是间隔时间的两倍，具体计算如下：

方波周期 = 2 × 间隔时间

= 2 × (T1CMP) x Timer1 时钟周期

= 2 × (T1CMP) x Timer1 计数时钟分频 x Timer1 计数时钟周期

Timer1 计数时钟周期由选择的计数时钟决定，共有三种时钟：

- 内部高速 OSC48M 的 3 分频
- 内部低速 800K
- 外部 32.768K 晶振(必须确保外接了 32.768K 晶振)

> 注： 1、T1CMP 在定时器模式下，Timer1 工作期间不能修改，所以想要改变 T1CMP 必须先把 T1CR 寄存器的 TEN 位清零，等待 T1SR 寄存器的 WREN 位置 1 后，再修改 T1CMP。
>
> 2、T1CMP 在 PWM 模式下，Timer1 工作期间可以直接修改。

6. 清 Timer1 中断标志位

使能 Timer1 中断前，先将 T1SR 寄存器的 INTSR 位清 0，清 Timer1 中断标志位。

7. 中断控制

Timer1 只有一个中断，但是在间隔定时、方波输出、PWM 输出、外部事件计数模式下，产生 Timer1 中断的条件不同。在间隔定时、方波输出、外部事件计数模式下，当 Timer1 计数器值等于(T1CMP - 1)时，会产生 Timer1 中断。在 PWM 输出模式下，当 Timer1 计数器值溢出，即等于 0xFF 时，会产生 Timer1 中断。

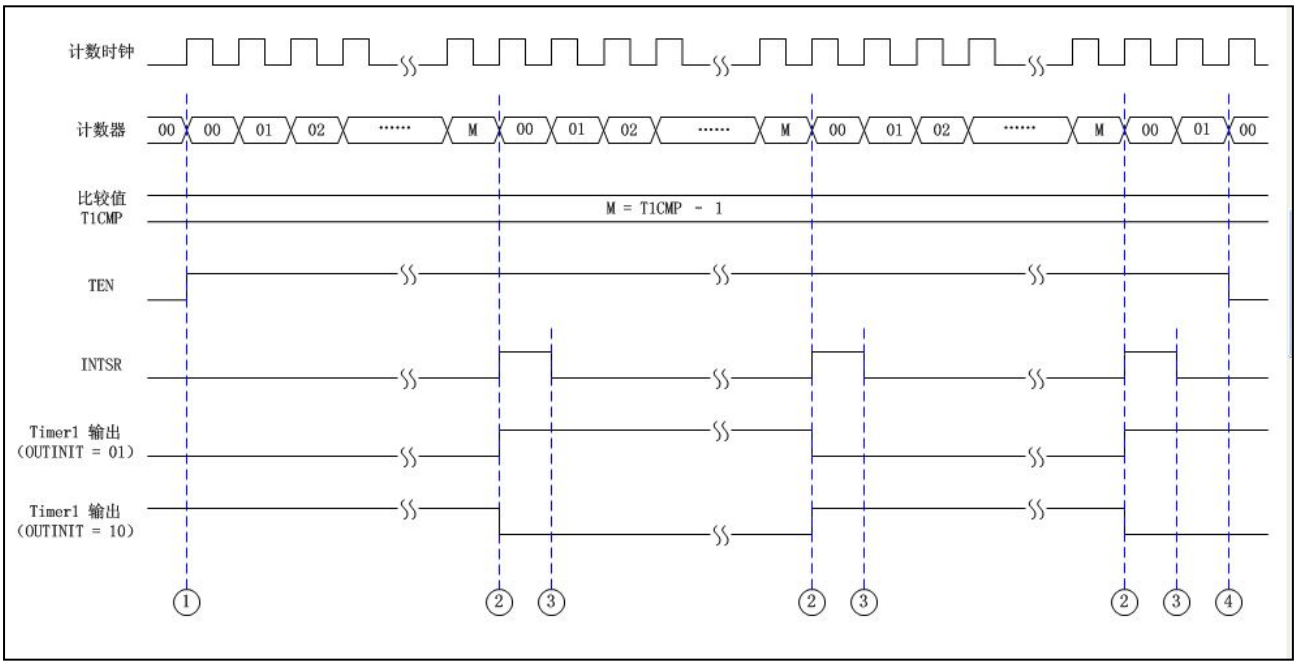
T1CR 寄存器的 INTEN 位是 Timer1 中断使能位，T1SR 寄存器的 INTSR 位是 Timer1 中断标志位。

8. Timer1 使能：

将 T1CR 寄存器的 TEN 位置 1,使能 Timer1,Timer1 即开始工作。

Timer1 间隔定时/方波输出工作时序如图 9-1 所示。

图 9-1： Timer1 间隔定时/方波输出工作时序图(T1CMP ≠ 1)



波形说明：

① 使能定时器，计数器开始计数；

② 计数器值等于 M，产生 Timer1 中断(即 T1SR 寄存器的 INTSR 位置 1)，同时 Timer1 输出翻转；

③ 软件处理并清除中断；

④ 不使能定时器，计数器清 0 并停止计数。

间隔时间 = (T1CMP) x Timer1 时钟周期

= (T1CMP) x Timer1 计数时钟分频 x Timer1 计数时钟周期

方波周期 = 2 x 间隔时间

注： 1、当 T1CMP = 0 时，按 T1CMP = 256 计算

2、当 T1CMP = 1 时，没有方波输出，也不产生 Timer1 中断。

9.1.3. PWM 输出的操作步骤

配置 Timer1 为 PWM 输出功能的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 1 输出/输入功能，并且配置 PnOE 寄存器，将定时器 1 输出或输入引脚方向置为输出。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer1 计数时钟参数配置：

- 计数时钟选择：

T1CKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer1 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer1 的时钟。

- 计数时钟分频配置：

T1CKCR 寄存器的 CLKDIV<3:0>位可以对选择的时钟源进行分频。此时不能配置为外部信号的有效沿选项。

- 计数时钟使能：

将 T1CKCR 寄存器的 CLKEN 位置 1，使能 Timer1 计数时钟。

4. Timer1 模式配置：

- 输出翻转使能：

T1CR 寄存器的 INVEN 置 1，使能当 Timer1 计数器值等于(T1CMP - 1)时，输出翻转。(不使能没有 PWM 波形输出)

- 输出反转选择：

T1CR 寄存器的 OUTSEL 位是 Timer1 输出反转使能位，只对 PWM 输出都有效，OUTSEL 位置 1 后输出波形会发生反转。(详情见图 9-2)

- 工作模式选择：

将 T1CR 寄存器的 MODE 位置 1，工作模式选择为 PWM 模式。

5. PWM 占空比配置：

Timer1 的 PWM 占空比配置与 Timer0 有所区别，只通过 T1CMP 配置，具体计算如下：

PWM 周期 = (0xFF + 1) x Timer1 时钟周期

= (0xFF + 1) x Timer1 计数时钟分频 x Timer1 计数时钟周期

Timer1 计数时钟周期由选择的计数时钟决定，共有三种时钟：

- 内部高速 OSC48M 的 3 分频
- 内部低速 800K
- 外部 32.768K 晶振(必须确保外接了 32.768K 晶振)

PWM 占空比 = (T1CMP) : (0xFF + 1)

简单地说，Timer1 PWM 的周期是固定的(0xFF + 1)，不能配置，T1CMP 配置的是低电平的时间，所以 T1CMP 必须小于(0xFF + 1)，也就是不能等于 00。(当输出反转位置 1 时，T1CMP 配置的是高电平的时间)

注：1、T1CMP 在定时器模式下，Timer1 工作期间不能修改，所以想要改变 T1CMP 必须先把 T1CR 寄存器的 TEN 位清零，等待 T1SR 寄存器的 WREN 位置 1 后，再修改 T1CMP。
2、T1CMP 在 PWM 模式下，Timer1 工作期间可以直接修改。

6. 清 Timer1 中断标志位

使能 Timer1 中断前，先将 T1SR 寄存器的 INTSR 位清 0，清 Timer1 中断标志位。

7. 中断控制：

Timer1 只有一个中断，但是在间隔定时、方波输出、PWM 输出、外部事件计数模式下，产生 Timer1 中断的条件不同。在间隔定时、方波输出、外部事件计数模式下，当 Timer1 计数器值等于(T1CMP - 1)时，会产生 Timer1 中断。在 PWM 输出模式下，当 Timer1 计数器值溢出，即等于 0xFF 时，会产生 Timer1 中断。

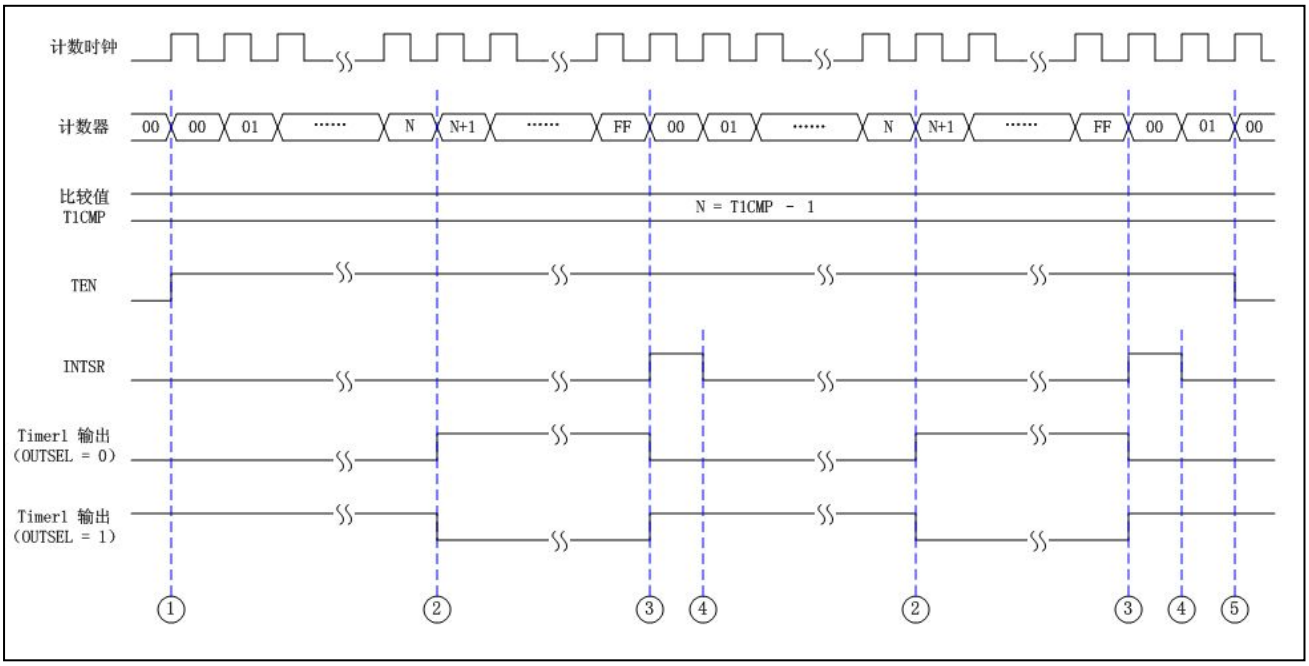
T1CR 寄存器的 INTEN 位是 Timer1 中断使能位，T1SR 寄存器的 INTSR 位是 Timer1 中断标志位。

8. Timer1 使能：

将 T1CR 寄存器的 TEN 位置 1，使能 Timer1，Timer1 即开始工作。

Timer1 PWM 输出工作时序如图 9-2 所示。

图 9-2： Timer1 PWM 输出工作时序图(T1CMP ≠ 0)



波形说明：

① 使能定时器，计数器开始计数；

② 计数器值等于 N，Timer1 输出翻转；

③ 计数器值等于 0xFF，产生 Timer1 中断(即 T1SR 寄存器的 INTSR 位置 1)，同时 Timer1 输出翻转；

④ 软件处理并清除中断；

⑤ 不使能定时器，计数器清 0 并停止计数。

PWM 周期 = (0xFF + 1) x Timer1 时钟周期

= (0xFF + 1) x Timer1 计数时钟分频 x Timer1 计数时钟周期

PWM 占空比 = (T1CMP) : (0xFF + 1)

注： 1、当 T1CMP = 0 时，按 T1CMP = 256 计算，此时产生方波。

2、若需要产生可变占空比的 PWM 波形，可以修改 T1CMP 寄存器。软件修改 T1CMP 寄存器后，在下个 PWM 周期才会生效。

9.1.4. 外部事件计数的操作步骤

配置 Timer1 为外部事件计数功能的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 1 输出/输入功能，并且配置 PnOE 寄存器，将定时器 1 输出或输入引脚方向置为输入。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer1 计数时钟参数配置：

- 外部信号的有效沿选择：

T1CKCR 寄存器的 CLKDIV<3:0>位可以对外部信号的有效沿进行选择，有上升沿、下降沿、双边沿。此时不能配置为计数时钟分频选项。

- 计数时钟使能：

将 T1CKCR 寄存器的 CLKEN 位置 1，使能 Timer1 计数时钟。

4. 工作模式选择：

将 T1CR 寄存器的 MODE 位清 0，工作模式选择为定时器模式。

5. 匹配值设置：

在外部事件计数模式下，Timer1 计数器在定时器 1 输入引脚的每个有效沿都会加 1，当计数器值等于(T1CMP－1)时，会产生中断。另外 T1CMP ≠ 1，T1CMP = 0 时，按 256 计算。

注：1、T1CMP 在定时器模式下，Timer1 工作期间不能修改，所以想要改变 T1CMP 必须先把 T1CR 寄存器的 TEN 位清零，等待 T1SR 寄存器的 WREN 位置 1 后，再修改 T1CMP。

2、T1CMP 在 PWM 模式下，Timer1 工作期间可以直接修改。

6. 清 Timer1 中断标志位

使能 Timer1 中断前，先将 T1SR 寄存器的 INTSR 位清 0，清 Timer1 中断标志位。

7. 中断控制：

Timer1 只有一个中断，但是在间隔定时、方波输出、PWM 输出、外部事件计数模式下，产生 Timer1 中断的条件不同。在间隔定时、方波输出、外部事件计数模式下，当 Timer1 计数器值等于(T1CMP－1)时，会产生 Timer1 中断。在 PWM 输出模式下，当 Timer1 计数器值溢出，即等于 0xFF 时，会产生 Timer1 中断。

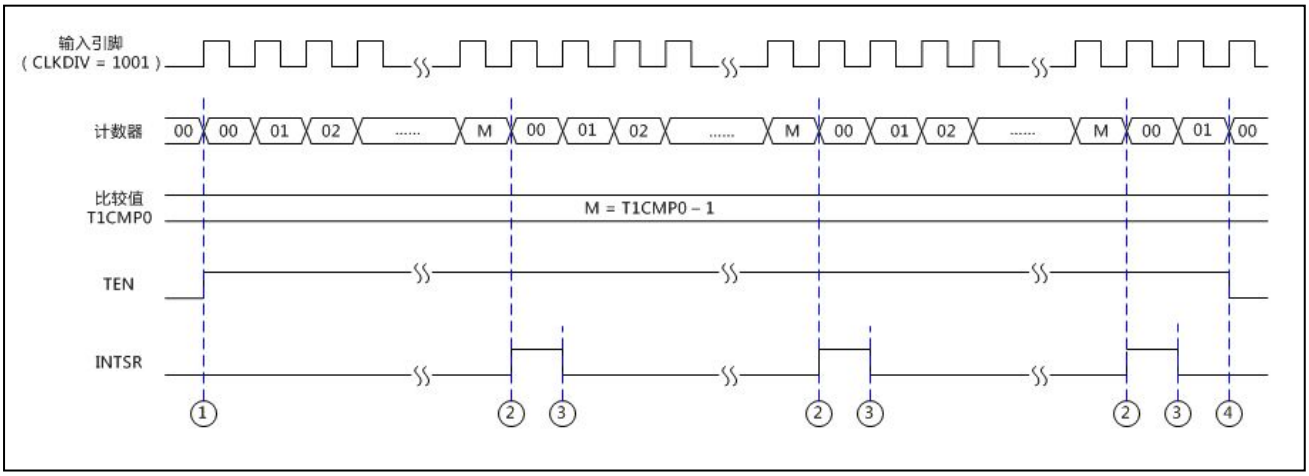
T1CR 寄存器的 INTEN 位是 Timer1 中断使能位，T1SR 寄存器的 INTSR 位是 Timer1 中断标志位。

8. Timer1 使能：

将 T1CR 寄存器的 TEN 位置 1，使能 Timer1，Timer1 即开始工作。

Timer1 外部事件计数工作时序如图 9-3 所示。

图 9-3： Timer1 外部事件计数工作时序图



波形说明：

① 使能定时器，计数器开始计数；

② 当计数器值等于 M，产生 Timer1 中断(即 T1SR 寄存器的 INTSR 位置 1)；

③ 软件处理并清除中断；

④ 不使能定时器，计数器清 0 并停止计数。

注： 1、当 T1CMP = 0 时，按 T1CMP = 256 计算，此时 M = T1CMP - 1 = 255。

2、外部输入信号的周期必须大于 Timer1 计数时钟周期的 2 倍，即外部输入信号的频率必须小于 Timer1 计数时钟频率的 1/2；

3、计数器在每个输入引脚的有效沿都会加 1；

4、T1CKCR 寄存器的 CLKDIV<3:0>位可以选择有效沿。

5、T1CMP = 1 时，不产生 Timer1 中断。

9.2. Timer1 寄存器的定义

以下寄存器用于控制 Timer1 的操作。

寄存器 9-1： T1CKCR：Timer1 时钟控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	CLKEN	CLKSEL	CLKDIV3	CLKDIV2	CLKDIV1	CLKDIV0
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未实现：读为 0

bit 5 CLKEN：Timer1 计数时钟使能位

0 = 关闭 Timer1 计数时钟

1 = 打开 Timer1 计数时钟

bit 4 CLKSEL：Timer1 计数时钟选择位

0 = 选择内部高速 OSC48M 的 3 分频作为 Timer1 计数时钟

1 = 选择低速时钟作为 Timer1 计数时钟(32KEN 位选择 OSC800K 或外部 32.768K)

bit 3-0 CLKDIV<3:0>：Timer1 计数时钟分频或外部信号有效沿选择位

0000 = 2 分频

0001 = 4 分频

0010 = 8 分频

0011 = 16 分频

0100 = 32 分频

0101 = 64 分频

0110 = 128 分频

0111 = 256 分频

1000 = 1 分频

1001 = 外部输入信号的上升沿

1010 = 外部输入信号的下降沿

1011 = 外部输入信号的双边沿

1100 = 1 分频

1101 = 1 分频

1110 = 1 分频

1111 = 1 分频

寄存器 9-2： T1CR：Timer1 模式控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	INVEN	OUTINIT1	OUTINIT0	INTEN	OUTSEL	MODE	TEN
U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 未实现：读为 0

bit 6 INVEN：Timer1 输出翻转使能位

0 = 当 Timer1 计数器与 T1CMP 匹配时，输出不翻转

1 = 当 Timer1 计数器与 T1CMP 匹配时，输出翻转

bit 5-4 OUTINIT<1:0>：Timer1 输出初始状态(定时器模式下才有效)

00 = 保持原状态

01 = 初始输出低电平

10 = 初始输出高电平

11 = 保持原状态

bit 3 INTEN：Timer1 中断使能位

0 = 禁止 Timer1 中断

1 = 使能 Timer1 中断

bit 2 OUTSEL：Timer1 输出反转使能位(PWM 模式下才有效)

0 = Timer1 输出不反转

1 = Timer1 输出反转

bit 1 MODE：Timer1 工作模式选择位

0 = 定时器模式

1 = PWM 模式

bit 0 TEN：Timer1 使能位

0 = Timer1 停止工作

1 = Timer1 开始工作

寄存器 9-3： T1CMP：Timer1 比较寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 T1CMP<7:0>：Timer1 比较寄存器

Timer1 比较寄存器，定时器模式下，在 Timer1 工作期间，不能修改此寄存器；PWM 模式下，可以修改。

注： 1、在定时器模式下，T1SR 寄存器的 WREN 位是 T1CMP 可写状态位，为 1 表示可以修改 T1CMP 寄存器，为 0 表示不能修改 T1CMP 寄存器。如范例 7-1 所示：

2、在 PWM 模式下，可以直接修改 T1CMP 寄存器。

范例 7-1：定时器模式下，修改 T1CMP 寄存器：

```
T1CR_TEN = 0x00;                //Timer1 停止工作
while(0x00 = T1SR_WREN){;}      //等待 T1CMP 写使能
T1CMP = 0x55;                   //修改 T1CMP 寄存器
……                              //其他处理
T1CR_TEN = 0x01;                //Timer1 开始工作
```

寄存器 9-4： T1SR：Timer1 状态寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	—	WREN	INTSR
U-0	U-0	U-0	U-0	U-0	U-0	R-1	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-2 未定义：读为 0

bit 1 WREN：T1CMP 可写标志位

0 = Timer1 正在工作，不可写 T1CMP

1 = Timer1 停止工作，可写 T1CMP，关闭 Timer1 后，应等待 WREN 为 1 后，再写 T1CMP

bit 0 INTSR：Timer1 中断标志位

0 = 没有产生 Timer1 中断条件

1 = 产生了 Timer1 中断条件

9.3. Timer1 中断

产生 Timer1 中断有以下三种方式：

- 间隔定时/方波输出/外部事件计数模式下，当 Timer1 计数器值等于(T1CMP - 1)时；
- PWM 模式下，当 Timer1 计数器值溢出，即等于 0xFF 时。

只要有 Timer1 中断条件产生，不论是否使能 Timer1 中断，T1SR 寄存器的 INTSR 位(Timer1 中断标志位)都会置 1，INTSR 位必须在软件中清零。Timer1 的中断使能位是 T1CR 寄存器的 INTEN 位。

注： 在休眠模式下，Timer1 中断无法唤醒处理器。
在空闲模式下，Timer1 中断可以唤醒处理器。

10. 16 位定时器 Timer2

Timer2 模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
T2CKCR	0x80FF	XRAM	Timer2时钟控制寄存器	0000 0000
T2CR	0x81FF	XRAM	Timer2模式控制寄存器	0000 0000
T2CMP0L	0x82FF	XRAM	Timer2比较寄存器0低8位	0000 0000
T2CMP0H	0x83FF	XRAM	Timer2比较寄存器0高8位	0000 0000
T2CMP1L	0x84FF	XRAM	Timer2比较寄存器1低8位	0000 0000
T2CMP1H	0x85FF	XRAM	Timer2比较寄存器1高8位	0000 0000
T2SR	0x86FF	XRAM	Timer2状态寄存器	0000 0010

Timer2 模块是一个 16 位定时器，具有如下特点：

- 16 位定时器
- 3 位预分频器
- 16 位周期寄存器(T2CMP0H、T2CMP0L)
- 16 位 PWM 脉宽调制寄存器(T2CMP1H、T2CMP1L)
- 可编程内部高速或低速时钟源
- 可编程内部或外部时钟源

10.1. Timer2 支持的功能

Timer2 支持以下 3 种功能：

- 间隔定时
- 方波输出
- PWM 输出

Timer2 与 Timer0 的工作原理及配置相同，只有一点不同就是 Timer0 是 8 位定时器，Timer2 是 16 位定时器。

10.1.1. 间隔定时的操作步骤

配置 Timer2 为间隔定时功能的操作步骤如下：

1. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

2. Timer2 计数时钟参数配置：

- 计数时钟选择：

T2CKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer2 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer2 的时钟。

- 计数时钟分频配置：

T2CKCR 寄存器的 CLKDIV<2:0>位可以对选择的时钟源进行分频。

- 计数时钟使能：

将 T2CKCR 寄存器的 CLKEN 位置 1，使能 Timer2 计数时钟。

3. 工作模式选择：

Timer2 有定时器和 PWM 两种工作模式，通过 T2CR 寄存器的 MODE 位控制，这两种模式都可以实现间隔定时功能。

4. 间隔时间配置：

Timer2 的间隔定时时间可以通过 T2CMP0H：T2CMP0L 寄存器配置，具体的间隔时间计算如下：

间隔时间 = (T2CMP0H：T2CMP0L) x Timer2 时钟周期

= (T2CMP0H：T2CMP0L) x Timer2 计数时钟分频 x Timer2 计数时钟周期

Timer2 计数时钟周期由选择的计数时钟决定，共有三种时钟：

- 内部高速 OSC48M 的 3 分频
- 内部低速 800K
- 外部 32.768K 晶振(必须确保外接了 32.768K 晶振)

> 注：(T2CMP0H：T2CMP0L)在 Timer2 工作期间是不能改变的，所以想要改变(T2CMP0H：T2CMP0L)必须先把 T2CR 寄存器的 TEN 位清零，等待 T2SR 寄存器的 WREN 位置 1 后，再修改(T2CMP0H：T2CMP0L)。

5. 清 Timer2 中断标志位

使能 Timer2 中断前，先将 T2SR 寄存器的 INTSR 位清 0，清 Timer2 中断标志位。

6. 中断控制：

Timer2 只有一种中断，就是当 Timer2 计数器值等于(T2CMP0H：T2CMP0L - 1)时，会产生 Timer2 中断。在间隔定时、方波输出、PWM 输出三种模式下都可以产生 Timer2 中断。

T2CR 寄存器的 INTEN 位是 Timer2 中断使能位，T2SR 寄存器的 INTSR 位是 Timer2 中断标志位。

7. Timer2 使能：

将 T2CR 寄存器的 TEN 位置 1,使能 Timer2,Timer2 即开始工作。

10.1.2. 方波输出的操作步骤

配置 Timer2 为方波输出功能的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 2 输出功能时，会自动将引脚方向配置为输出。不需要配置对应的 PnOE 寄存器，选择定时器 2 输出引脚方向为输出。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer2 计数时钟参数配置：

- 计数时钟选择：

T2CKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer2 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer2 的时钟。

- 计数时钟分频配置：

T2CKCR 寄存器的 CLKDIV<2:0>位可以对选择的时钟源进行分频。

- 计数时钟使能：

将 T2CKCR 寄存器的 CLKEN 位置 1，使能 Timer2 计数时钟。

4. Timer2 模式配置：

- 输出反转选择：

T2CR 寄存器的 OUTSEL 位是 Timer2 输出反转使能位，对于方波输出和 PWM 输出都有效，OUTSEL 位置 1 后输出波形会发生反转。

- 工作模式选择：

将 T2CR 寄存器的 MODE 位清 0，工作模式选择为定时器模式。

5. 方波周期配置：

Timer2 的方波周期是通过 T2CMP0H:T2CMP0L 寄存器配置，方波周期其实就是间隔时间的两倍，具体计算如下：

方波周期 = 2 × 间隔时间

= 2 ×(T2CMP0H：T2CMP0L) x Timer2 时钟周期

= 2 ×(T2CMP0H：T2CMP0L) x Timer2 计数时钟分频 x Timer2 计数时钟周期

Timer2 计数时钟周期由选择的计数时钟决定，共有三种时钟：

- 内部高速 OSC48M 的 3 分频
- 内部低速 800K
- 外部 32.768K 晶振(必须确保外接了 32.768K 晶振)

> 注：(T2CMP0H：T2CMP0L)在 Timer2 工作期间是不能改变的，所以想要改变(T2CMP0H：T2CMP0L)必须先把 T2CR 寄存器的 TEN 位清零，等待 T2SR 寄存器的 WREN 位置 1 后，再修改(T2CMP0H：T2CMP0L)。

6. 清 Timer2 中断标志位

使能 Timer2 中断前，先将 T2SR 寄存器的 INTSR 位清 0，清 Timer2 中断标志位。

7. 中断控制：

Timer2 只有一种中断，就是当 Timer2 计数器值等于(T2CMP0H：T2CMP0L - 1)时，会产生 Timer2 中断。在间隔定时、方波输出、PWM 输出三种模式下都可以产生 Timer2 中断。

T2CR 寄存器的 INTEN 位是 Timer2 中断使能位，T2SR 寄存器的 INTSR 位是 Timer2 中断标志位。

8. Timer2 使能：

将 T2CR 寄存器的 TEN 位置 1，使能 Timer2，Timer2 即开始工作。

10.1.3. PWM 输出的操作步骤

配置 Timer2 为 PWM 输出功能的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 2 输出功能时，会自动将引脚方向配置为输出。不需要配置对应的 PnOE 寄存器，选择定时器 2 输出引脚方向为输出。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer2 计数时钟参数配置：

- 计数时钟选择：

T2CKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer2 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer2 的时钟。

- 计数时钟分频配置：

T2CKCR 寄存器的 CLKDIV<2:0>位可以对选择的时钟源进行分频。

- 计数时钟使能：

将 T2CKCR 寄存器的 CLKEN 位置 1，使能 Timer2 计数时钟。

4. Timer2 模式配置：

- 输出反转选择：

T2CR 寄存器的 OUTSEL 位是 Timer2 输出反转使能位，对于方波输出和 PWM 输出都有效，OUTSEL 位置 1 后输出波形会发生反转。

- 工作模式选择：

将 T2CR 寄存器的 MODE 位置 1，工作模式选择为 PWM 模式。

5. PWM 占空比配置：

Timer2 的 PWM 占空比是通过 T2CMP0H：T2CMP0L 与 T2CMP1H：T2CMP1L 寄存器进行配置，具体计算如下：

PWM 周期 = (T2CMP0H：T2CMP0L) x Timer2 时钟周期

= (T2CMP0H：T2CMP0L)x Timer2 计数时钟分频 x Timer2 计数时钟周期

Timer2 计数时钟周期由选择的计数时钟决定，共有三种时钟：

- 内部高速 OSC48M 的 3 分频
- 内部低速 800K
- 外部 32.768K 晶振(必须确保外接了 32.768K 晶振)

PWM 占空比 = (T2CMP1H：T2CMP1L)：(T2CMP0H：T2CMP0L)

当(T2CMP0H： T2CMP0L)= 0 时， 按(T2CMP0H：T2CMP0L)= 65536 计算。

简单地说，(T2CMP0H：T2CMP0L)是配置 PWM 输出的周期，(T2CMP1H:T2CMP1L)配置的是输出低电平的时间，所以(T2CMP1H:T2CMP1L)必须小于(T2CMP0H:T2CMP0L)。(当输出反转位置 1 时，(T2CMP1H：T2CMP1L)配置的是输出高电平的时间)

在 PWM 模式下，可以直接修改(T2CMP1H：T2CMP1L)寄存器来改变 PWM 的占空比，修改(T2CMP1H：T2CMP1L)时，新的占空比配置将在下个 PWM 周期生效。

> 注： (T2CMP0H：T2CMP0L)在 Timer2 工作期间是不能改变的，所以想要改变(T2CMP0H：T2CMP0L)必须先把 T2CR 寄存器的 TEN 位清零，等待 T2SR 寄存器的 WREN 位置 1 后，再修改(T2CMP0H：T2CMP0L)。

6. 清 Timer2 中断标志位

使能 Timer2 中断前，先将 T2SR 寄存器的 INTSR 位清 0，清 Timer2 中断标志位。

7. 中断控制：

Timer2 只有一种中断，就是当 Timer2 计数器值等于(T2CMP0H：T2CMP0L - 1)时，会产生 Timer2 中断。在间隔定时、方波输出、PWM 输出三种模式下都可以产生 Timer2 中断。

T2CR 寄存器的 INTEN 位是 Timer2 中断使能位，T2SR 寄存器的 INTSR 位是 Timer2 中断标志位。

8. Timer2 使能：

将 T2CR 寄存器的 TEN 位置 1，使能 Timer2，Timer2 即开始工作。

10.2. Timer2 寄存器的定义

以下寄存器用于控制 Timer2 的操作。

寄存器 10-1： T2CKCR：Timer2 时钟控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	CLKEN	CLKSEL	—	CLKDIV2	CLKDIV1	CLKDIV0
U-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未实现：读为 0

bit 5 CLKEN：Timer2 计数时钟使能位

0 = 关闭 Timer2 计数时钟

1 = 打开 Timer2 计数时钟

bit 4 CLKSEL：Timer2 计数时钟选择位

0 = 选择内部高速 OSC48M 的 3 分频作为 Timer2 计数时钟

1 = 选择低速时钟作为 Timer2 计数时钟(32KEN 位选择 OSC800K 或外部 32.768K)

bit 3 未实现：读为 0

bit 2-0 CLKDIV<2:0>：Timer2 计数时钟分频选择位

000 = 2 分频

001 = 4 分频

010 = 8 分频

011 = 16 分频

100 = 32 分频

101 = 64 分频

110 = 128 分频

111 = 1 分频

寄存器 10-2： T2CR：Timer2 模式控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	INTEN	OUTSEL	MODE	TEN
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未实现：读为 0

bit 3 INTEN：Timer2 中断使能位
0 = 禁止计数器与 T2CMP0H、T2CMP0L 匹配中断
1 = 使能计数器与 T2CMP0H、T2CMP0L 匹配中断

bit 2 OUTSEL：Timer2 输出反转使能位
0 = Timer2 输出不反转
1 = Timer2 输出反转

bit 1 MODE：Timer2 工作模式选择位
0 = 定时器模式
1 = PWM 模式

bit 0 TEN：Timer2 使能位
0 = Timer2 停止工作
1 = Timer2 开始工作

寄存器 10-3： T2CMP0L：Timer2 比较寄存器 0 低 8 位

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 T2CMP0L<7:0>：Timer2 比较寄存器 0 低 8 位

Timer2 比较寄存器 0 低 8 位，在 Timer2 工作期间，不能修改此寄存器

注： T2CMP0H 和 T2CMP0L 组成 16 位的比较寄存器 0。

寄存器 10-3： T2CMP0H：Timer2 比较寄存器 0 高 8 位(续)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 T2CMP0H<7:0>：Timer2 比较寄存器 0 高 8 位

Timer2 比较寄存器 0 高 8 位，在 Timer2 工作期间，不能修改此寄存器

注： 1、T2CMP0H 和 T2CMP0L 组成 16 位的比较寄存器 0。

2、T2SR 寄存器的 WREN 位是 T2CMP0H 和 T2CMP0L 可写状态位，为 1 表示可以修改 T2CMP0H 和 T2CMP0L 寄存器，为 0 表示不能修改 T2CMP0H 和 T2CMP0L 寄存器。所以正确的修改 T2CMP0 寄存器如范例 8-1 所示：

范例 8-1：修改 T2CMP0H、T2CMP0L 寄存器：

```
T2CR_TEN = 0x00;                    //Timer2 停止工作
while(0x00 = T2SR_WREN){;}          //等待 T2CMP0H、T2CMP0L 写使能
T2CMP0L = 0x60;                     //修改 T2CMP0L 寄存器
T2CMP0H = 0x02;                     //修改 T2CMP0H 寄存器
……                                  //其他处理
T2CR_TEN = 0x01;                    //Timer2 开始工作
```

寄存器 10-4： T2CMP1L：Timer2 比较寄存器 1 低 8 位

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 T2CMP1L<7:0>：Timer2 比较寄存器 1 低 8 位

Timer2 比较寄存器 1 低 8 位，在 Timer2 工作期间，可以修改此寄存器。

T2CMP0H、T2CMP0L 决定输出 PWM 的周期，T2CMP1H、T2CMP1L 决定输出 PWM 的占空比，在 Timer2 工作期间，可以修改 T2CMP1H、T2CMP1L 寄存器改变输出 PWM 的占空比。

注： 1、T2CMP1H 和 T2CMP1L 组成 16 位的比较寄存器 1。

2、需要注意的是必须先修改 T2CMP1L、在修改 T2CMP1H。

寄存器 10-4： T2CMP1H：Timer2 比较寄存器 1 高 8 位(续)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 T2CMP1H<7:0>：Timer2 比较寄存器 1 高 8 位

Timer2 比较寄存器 1 高 8 位，在 Timer2 工作期间，可以修改此寄存器。

T2CMP0H、T2CMP0L 决定输出 PWM 的周期，T2CMP1H、T2CMP1L 决定输出 PWM 的占空比，在 Timer2 工作期间，可以修改 T2CMP1H、T2CMP1L 寄存器改变输出 PWM 的占空比。

注： 1、T2CMP1H 和 T2CMP1L 组成 16 位的比较寄存器 1。

2、写 T2CMP1H 和 T2CMP1L 在硬件上做了保护，必须写 T2CMP1H 后配置才会有效，如果只写 T2CMP1L，配置不会生效，占空比不变，所以推荐的方法是同时写 T2CMP1H、T2CMP1L，而且必须先修改 T2CMP1L、在修改 T2CMP1H。如范例 8-2 所示：

范例 8-1：修改 T2CMP0H、T2CMP0L 寄存器：

```
……                          //其他处理
T2CMP1L = 0x40;             //先修改 T2CMP1L 寄存器
T2CMP2H = 0x01;             //再修改 T2CMP2H 寄存器
……                          //其他处理
```

寄存器 10-5： T2SR：Timer2 状态寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	—	WREN	INTSR
U-0	U-0	U-0	U-0	U-0	U-0	R-1	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-2 未定义：读为 0

bit 1 WREN：TOCMP0 可写标志位
0 = Timer0 正在工作，不可写 TOCMP0
1 = Timer0 停止工作，可写 TOCMP0，关闭 Timer0 后，应等待 WREN 为 1 后，再写 TOCMP0

bit 0 INTSR：Timer0 计数器与 TOCMP0 匹配中断标志位
0 = Timer0 计数器与 TOCMP0 不匹配
1 = 发生了 Timer0 计数器与 TOCMP0 匹配

10.3. Timer2 中断

不论是在定时器模式还是 PWM 模式下，当 Timer2 计数器值等于(T2CMP0H:T2CMP0L－1)时，都会产生 Timer2 中断。

只要有 Timer2 中断条件产生，不论是否使能 Timer2 中断，T2SR 寄存器的 INTSR 位(Timer2 中断标志位)都会置 1，INTSR 位必须在软件中清零。Timer2 的中断使能位是 T2CR 寄存器的 INTEN 位。

注： 在休眠模式下，Timer2 中断无法唤醒处理器。
在空闲模式下，Timer2 中断可以唤醒处理器。

11. 16 位定时器 Timer3

Timer3 模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
T3CR	0x88FF	XRAM	Timer3模式控制寄存器	0000 0000
T3CKCR	0x89FF	XRAM	Timer3时钟控制寄存器	0000 0000
T3OCR	0x8AFF	XRAM	Timer3输出控制寄存器	0000 0000
T3CMP0L	0x8BFF	XRAM	Timer3比较寄存器0低8位	0000 0000
T3CMP0H	0x8CFF	XRAM	Timer3比较寄存器0高8位	0000 0000
T3CMP1L	0x8DFF	XRAM	Timer3比较寄存器1低8位	0000 0000
T3CMP1H	0x8EFF	XRAM	Timer3比较寄存器1高8位	0000 0000
T3SR	0x8FFF	XRAM	Timer3状态寄存器	0000 1000

Timer3 模块是一个 16 位定时器，具有如下特点：

- 16 位定时器
- 3 位预分频器
- 16 位周期寄存器(T3CMP0H、T3CMP0L)
- 16 位 PWM 脉宽调制寄存器(T3CMP1H、T3CMP1L)
- 16 位捕捉功能寄存器(T3CMP0H、T3CMP0L、T3CMP1H、T3CMP1L)
- 可编程外部时钟边沿选择
- 可编程内部高速或低速时钟源
- 可编程内部或外部时钟源

11.1. Timer3 的支持的功能

Timer3 支持以下 6 种功能：

- 间隔定时
- 方波输出
- PWM 输出
- 外部事件计数
- 可编程脉冲输出
- 外部电平宽度测量

11.1.1. 间隔定时的操作步骤

配置 Timer3 为间隔定时功能的操作步骤如下：

1. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

2. Timer3 计数时钟参数配置：

- 计数时钟选择：

T3CKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer3 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer3 的时钟。

- 计数时钟分频配置：

T3CKCR 寄存器的 CLKDIV<2:0>位可以对选择的时钟源进行分频。此时不能配置为输入引脚的有效沿作为计数时钟选项。

- 输入引脚的有效沿设置：

将 T3CKCR 寄存器的 EDGESEL<1:0>位配置为 00，即输入引脚的有效沿选择为禁止设置。

- 计数时钟使能：

将 T3CKCR 寄存器的 CLKEN 位置 1，使能 Timer3 计数时钟。

3. T3CMP0、T3CMP1 寄存器组的模式选择：

将 T3CR 寄存器的 CMP0M 位清 0，CMP1M 位置 1，配置 T3CMP0 寄存器组作为比较寄存器，T3CMP1 寄存器组作为捕获寄存器。

4. 间隔时间配置：

Timer3 的间隔定时时间可以通过(T3CMP0H：T3CMP0L)寄存器配置，具体的间隔时间计算如下：

间隔时间 = (T3CMP0H：T3CMP0L) x Timer3 时钟周期

= (T3CMP0H：T3CMP0L) x Timer3 计数时钟分频 x Timer3 计数时钟周期

Timer3 计数时钟周期由选择的计数时钟决定，共有三种时钟：

- 内部高速 OSC48M 的 3 分频
- 内部低速 800K
- 外部 32.768K 晶振(必须确保外接了 32.768K 晶振)

> 注：(T3CMP0H：T3CMP0L)在 Timer3 工作期间是不能改变的，所以想要改变(T3CMP0H：T3CMP0L)必须先配置 T3CR 寄存器的 MODE 位 = 000，等待 T3SR 寄存器的 WREN 位置 1 后，再修改(T3CMP0H：T3CMP0L)。

5. 清 Timer3 中断标志位

使能 Timer3 中断前，先将 T3SR 寄存器的 INT0SR、INT1SR 位清 0，清 Timer3 中断标志位。

6. 中断控制：

Timer3 有两个中断，Timer3 中断 0、Timer3 中断 1，Timer3 中断 1 只有在单脉冲输出、外部电平宽度测量模式下才可以产生，Timer3 中断 0 在所有模式下都可以产生。具体请看 11.3 小节“Timer3 中断”。

T3CR 寄存器的 INT0EN、INT1EN 位是 Timer3 中断使能位，T3SR 寄存器的 INT0SR、INT1SR 位是 Timer3 中断标志位。

7. 工作模式选择：

配置 T3CR 寄存器的 MODE<2:0>位 = 110 或者 111，选择 Timer3 的工作模式，配置完成后，Timer3 即开始工作。

11.1.2. 方波输出的操作步骤

配置 Timer3 为方波输出功能的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 3 输出/输入 0 功能，并且配置 PnOE 寄存器，将定时器 3 输出或输入 0 引脚方向置为输出。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer3 计数时钟参数配置：

- 计数时钟选择：

T3CKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer3 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer3 的时钟。

- 计数时钟分频配置：

T3CKCR 寄存器的 CLKDIV<2:0>位可以对选择的时钟源进行分频。此时不能配置为输入引脚的有效沿作为计数时钟选项。

- 输入引脚的有效沿设置：

将 T3CKCR 寄存器的 EDGESEL<1:0>位配置为 00，即输入引脚的有效沿选择为禁止设置。

- 计数时钟使能：

将 T3CKCR 寄存器的 CLKEN 位置 1，使能 Timer3 计数时钟。

4. Timer3 输出配置：

- 禁止单脉冲输出模式(连续脉冲输出模式)：

将 T3OCR 寄存器的 SPOM 位清 0，选择连续脉冲输出模式。

- 输出初始状态选择：

T3OCR 寄存器的 OUTINIT<1:0>位控制 Timer3 输出初始状态，只对方波输出、单脉冲输出有效，对 PWM 输出无效。

- 计数器与 T3CMP0、T3CMP1 寄存器组匹配输出翻转配置：

将 T3OCR 寄存器的 INV0EN 位置 1、INV1EN 位清 0，禁止计数器与(T3CMP1H：T3CMP1L)匹配输出翻转，使能计数器与(T3CMP0H：T3CMP0L)匹配输出翻转。

5. T3CMP0、T3CMP1 寄存器组的模式选择：

将 T3CR 寄存器的 CMP0M 位清 0，CMP1M 位置 1，配置 T3CMP0 寄存器组作为比较寄存器，T3CMP1 寄存器组作为捕获寄存器。

6. 方波周期配置：

Timer3 的方波周期是通过 T3CMP0H:T3CMP0L 寄存器配置，方波周期其实就是间隔时间的两倍，具体计算如下：

方波周期 = 2 × 间隔时间

= 2 × (T2CMP0H：T2CMP0L) x Timer3 时钟周期

= 2 × (T2CMP0H：T2CMP0L) x Timer3 计数时钟分频 x Timer3 计数时钟周期

Timer3 计数时钟周期由选择的计数时钟决定，共有三种时钟：

- 内部高速 OSC48M 的 3 分频
- 内部低速 800K
- 外部 32.768K 晶振(必须确保外接了 32.768K 晶振)

> 注：(T3CMP0H：T3CMP0L)在 Timer3 工作期间是不能改变的，所以想要改变(T3CMP0H：T3CMP0L)必须先配置 T3CR 寄存器的 MODE 位 = 000，等待 T3SR 寄存器的 WREN 位置 1 后，再修改(T3CMP0H：T3CMP0L)。

7. 清 Timer3 中断标志位

使能 Timer3 中断前，先将 T3SR 寄存器的 INT0SR、INT1SR 位清 0，清 Timer3 中断标志位。

8. 中断控制：

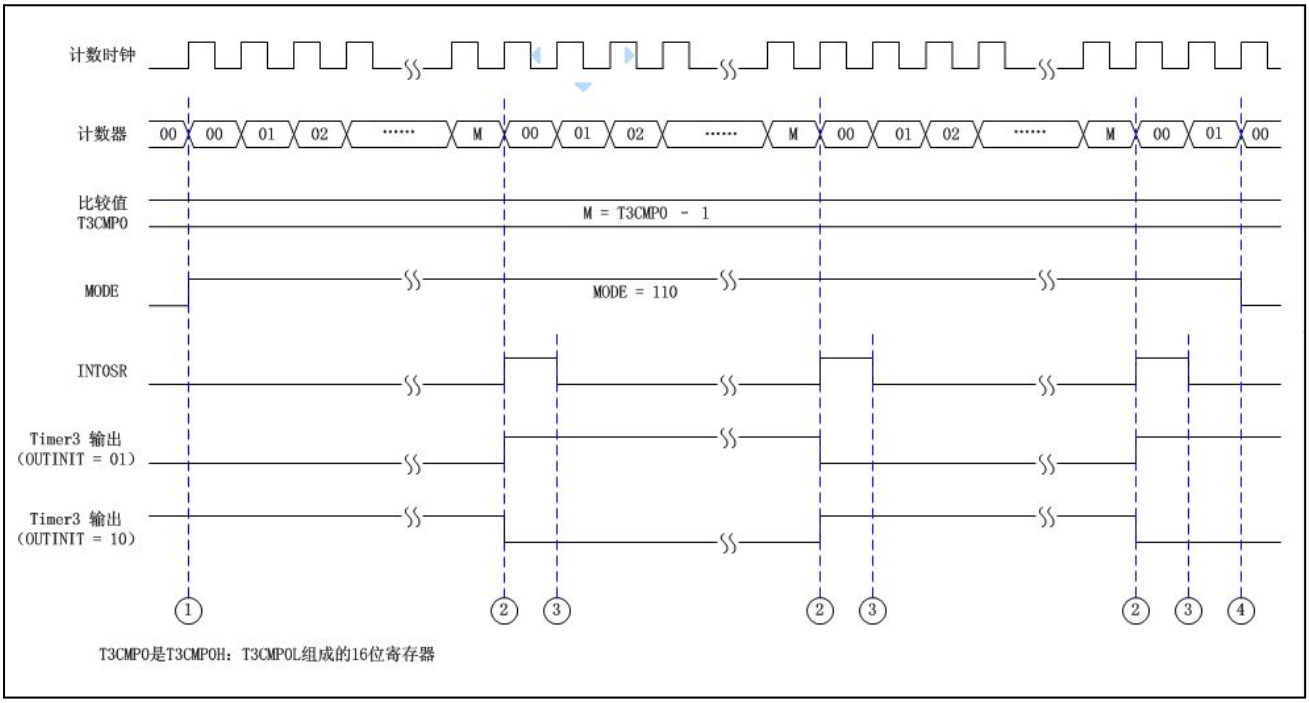
Timer3 有两个中断，Timer3 中断 0、Timer3 中断 1，Timer3 中断 1 只有在单脉冲输出、外部电平宽度测量模式下才可以产生，Timer3 中断 0 在所有模式下都可以产生。具体请看 11.3 小节“Timer3 中断”。

T3CR 寄存器的 INT0EN、INT1EN 位是 Timer3 中断使能位，T3SR 寄存器的 INT0SR、INT1SR 位是 Timer3 中断标志位。

9. 工作模式选择

配置 T3CR 寄存器的 MODE<2:0>位 = 110，选择 Timer3 的工作模式，配置完成后，Timer3 即开始工作。Timer3 的间隔定时/方波输出工作时序如图 11-1 所示。

图 11-1： Timer3 间隔定时/方波输出工作时序图



波形说明：

① 设置计数器模式 MODE<2:0>位 = 110 后，计数器开始计数；

② 当计数器值等于 M，产生 Timer3 中断 0(即 T3SR 寄存器的 INT0SR 位置 1)，同时 Timer3 输出翻转；

③ 软件处理并清除中断；

④ 设置计数器模式 MODE<2:0>位 = 000 后，计数器清零并停止计数。

间隔时间 = (T3CMP0)x Timer3 时钟周期

= (T3CMP0)x Timer3 计数时钟分频 x Timer3 计数时钟周期

方波周期 = 2 x 间隔时间

注： 1、当 T3CMP0 = 0 时，按 T3CMP0 = 65536 计算。

2、当 T3CMP0 = 1 时，没有方波输出，也不产生 Timer3 中断 0。

11.1.3. PWM 输出的操作步骤

配置 Timer3 为 PWM 输出功能的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 3 输出/输入 0 功能，并且配置 PnOE 寄存器，将定时器 3 输出或输入 0 引脚方向置为输出。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer3 计数时钟参数配置：

- 计数时钟选择：

T3CKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer3 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer3 的时钟。

- 计数时钟分频配置：

T3CKCR 寄存器的 CLKDIV<2:0>位可以对选择的时钟源进行分频。此时不能配置为输入引脚的有效沿作为计数时钟选项。

- 输入引脚的有效沿设置：

将 T3CKCR 寄存器的 EDGESEL<1:0>位配置为 00，输入引脚的有效沿选择为禁止设置。

- 计数时钟使能：

将 T3CKCR 寄存器的 CLKEN 位置 1，使能 Timer3 计数时钟。

4. Timer3 输出配置：

- 禁止单脉冲输出模式(连续脉冲输出模式)：

将 T3OCR 寄存器的 SPOM 位清 0，选择连续脉冲输出模式。

- 计数器与 T3CMP0、T3CMP1 寄存器组匹配输出翻转配置：

将 T3OCR 寄存器的 INV0EN、INV1EN 位同时置 1，使能计数器与(T3CMP1H：T3CMP1L)匹配输出翻转，计数器与(T3CMP0H：T3CMP0L)匹配输出翻转。

5. T3CMP0、T3CMP1 寄存器组的模式选择：

将 T3CR 寄存器的 CMP0M、CMP1M 位清 0。配置 T3CMP0 寄存器组、T3CMP1 寄存器组都作为比较寄存器。

6. PWM 占空比配置：

Timer3 的 PWM 占空比是通过 T3CMP0H：T3CMP0L 与 T3CMP1H：T3CMP1L 寄存器进行配置，具体计算如下：

PWM 周期 = (T3CMP0H：T3CMP0L) x Timer3 时钟周期

= (T3CMP0H：T3CMP0L) x Timer3 计数时钟分频 x Timer3 计数时钟周期

Timer3 计数时钟周期由选择的计数时钟决定，共有三种时钟：

- 内部高速 OSC48M 的 3 分频
- 内部低速 800K
- 外部 32.768K 晶振(必须确保外接了 32.768K 晶振)

PWM 占空比 =(T3CMP1H：T3CMP1L)：(T3CMP0H：T3CMP0L)

当(T3CMP0H ： T3CMP0L)= 0 时，按(T3CMP0H ： T3CMP0L)= 65536 计算。

简单地说，(T3CMP0H：T3CMP0L)是配置 PWM 输出的周期，(T3CMP1H:T3CMP1L)配置的是输出低电平的时间，所以(T3CMP1H:T3CMP1L)必须小于(T3CMP0H:T3CMP0L)。

在 PWM 模式下，可以直接修改(T3CMP1H：T3CMP1L)寄存器来改变 PWM 的占空比，修改(T3CMP1H：T3CMP1L)时，新的占空比配置将在下个 PWM 周期生效。

注：(T3CMP0H：T3CMP0L)在 Timer3 工作期间是不能改变的，所以想要改变(T3CMP0H：T3CMP0L)必须先配置 T3CR 寄存器的 MODE 位 = 000，等待 T3SR 寄存器的 WREN 位置 1 后，再修改(T3CMP0H：T3CMP0L)。

7. 清 Timer3 中断标志位

使能 Timer3 中断前，先将 T3SR 寄存器的 INT0SR、INT1SR 位清 0，清 Timer3 中断标志位。

8. 中断控制：

Timer3 有两个中断，Timer3 中断 0、Timer3 中断 1，Timer3 中断 1 只有在单脉冲输出、外部电平宽度测量模式下才可以产生，Timer3 中断 0 在所有模式下都可以产生。具体请看 11.3 小节“Timer3 中断”。

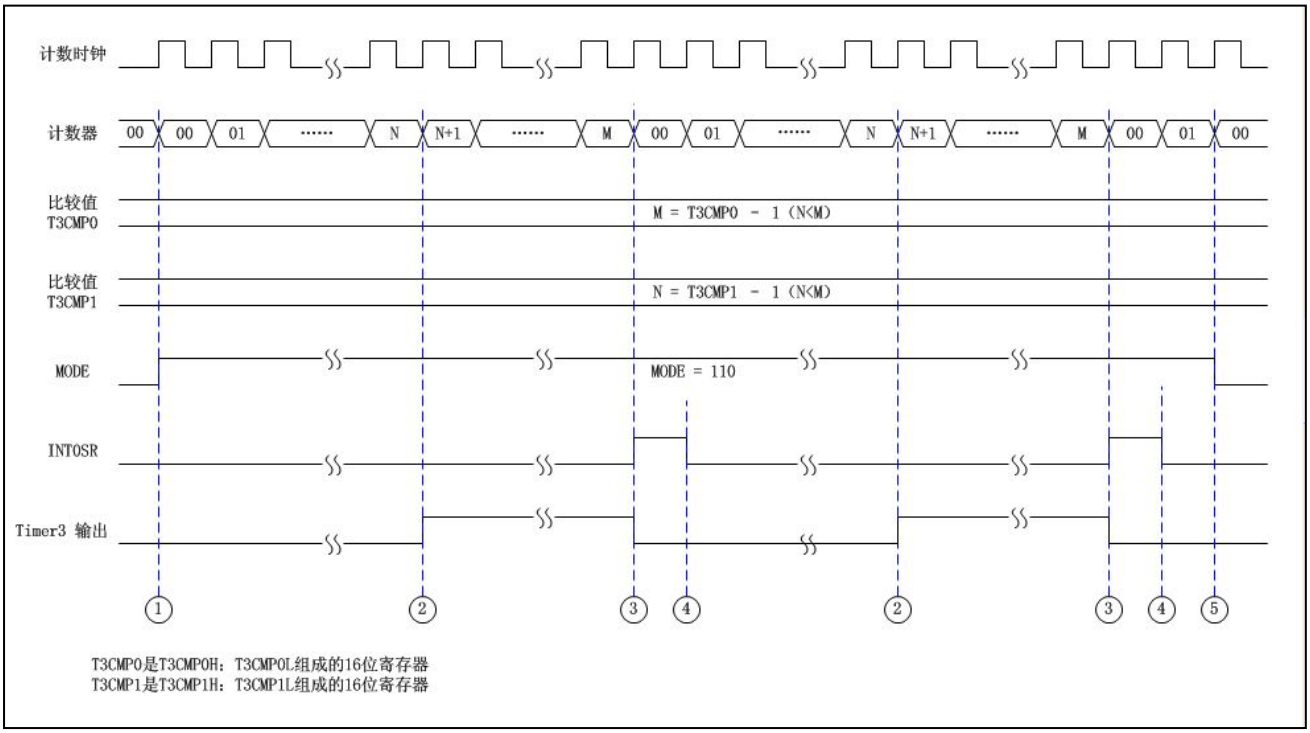
T3CR 寄存器的 INT0EN、INT1EN 位是 Timer3 中断使能位，T3SR 寄存器的 INT0SR、INT1SR 位是 Timer3 中断标志位。

9. 工作模式选择：

配置 T3CR 寄存器的 MODE<2:0>位 = 110，选择 Timer3 的工作模式，配置完成后，Timer3 即开始工作。

Timer3 的 PWM 输出工作时序如图 11-2 所示。

图 11-2： Timer3 PWM 输出工作时序图



波形说明：

① 设置计数器模式 MODE<2:0>位 = 110 后，计数器开始计数；

② 当计数器值等于 N，Timer3 输出翻转；

③ 当计数器值等于 M，产生 Timer3 中断 0(即 T3SR 寄存器的 INT0SR 位置 1)，同时 Timer3 输出翻转；

④ 软件处理并清除中断

⑤ 设置计数器模式 MODE<2:0>位 = 000 后，计数器清零并停止计数。

PWM 周期 = (T3CMP0)x Timer3 时钟周期

= (T3CMP0)x Timer3 计数时钟分频 x Timer3 计数时钟周期

PWM 占空比 = T3CMP1 ： T3CMP0

注： T3CMP0 = 0 时，按 T3CMP0 = 65536 计算

11.1.4. 外部事件计数的操作步骤

配置 Timer3 为外部事件计数功能的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 3 输出/输入 0 功能，并且配置 PnOE 寄存器，将定时器 3 输出或输入 0 引脚方向置为输入。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer3 计数时钟参数配置：

- 计数时钟分频配置：

将 T3CKCR 寄存器的 CLKDIV<2:0>位置为 1XX，计数时钟配置为输入引脚的有效沿。此时不能配置为计数时钟分频选项。

- 输入引脚的有效沿设置：

T3CKCR 寄存器的 EDGESEL<1:0>位可以选择输入引脚的有效沿，有上升沿、下降沿、双边沿三种，此时不能配置为禁止设置选项。

- 计数时钟使能：

将 T3CKCR 寄存器的 CLKEN 位置 1，使能 Timer3 计数时钟。

4. 禁止单脉冲输出模式(连续脉冲输出模式)：

将 T3OCR 寄存器的 SPOM 位清 0，选择连续脉冲输出模式。

5. T3CMP0、T3CMP1 寄存器组的模式选择：

将 T3CR 寄存器的 CMP0M 位清 0，CMP1M 位置 1，配置 T3CMP0 寄存器组作为比较寄存器，T3CMP1 寄存器组作为捕获寄存器。

6. 匹配值设置：

在外部事件计数模式下，Timer3 计数器在定时器 3 输入 0 引脚的每个有效沿都会加 1，当计数器值等于(T3CMP0H:T3CMP0L - 1)时，会产生 Timer3 中断 0。T3CMP0H:T3CMP0L ≠ 1，当 T3CMP0H:T3CMP0L = 0 时，按 T3CMP0H:T3CMP0L = 65536 计算。

> 注：(T3CMP0H：T3CMP0L)在 Timer3 工作期间是不能改变的，所以想要改变(T3CMP0H：T3CMP0L)必须先配置 T3CR 寄存器的 MODE 位 = 000，等待 T3SR 寄存器的 WREN 位置 1 后，再修改(T3CMP0H：T3CMP0L)。

7. 清 Timer3 中断标志位

使能 Timer3 中断前，先将 T3SR 寄存器的 INT0SR、INT1SR 位清 0，清 Timer3 中断标志位。

8. 中断控制：

Timer3 有两个中断，Timer3 中断 0、Timer3 中断 1，Timer3 中断 1 只有在单脉冲输出、外部电平宽度测量模式下才可以产生，Timer3 中断 0 在所有模式下都可以产生。具体请看 11.3 小节“Timer3 中断”。

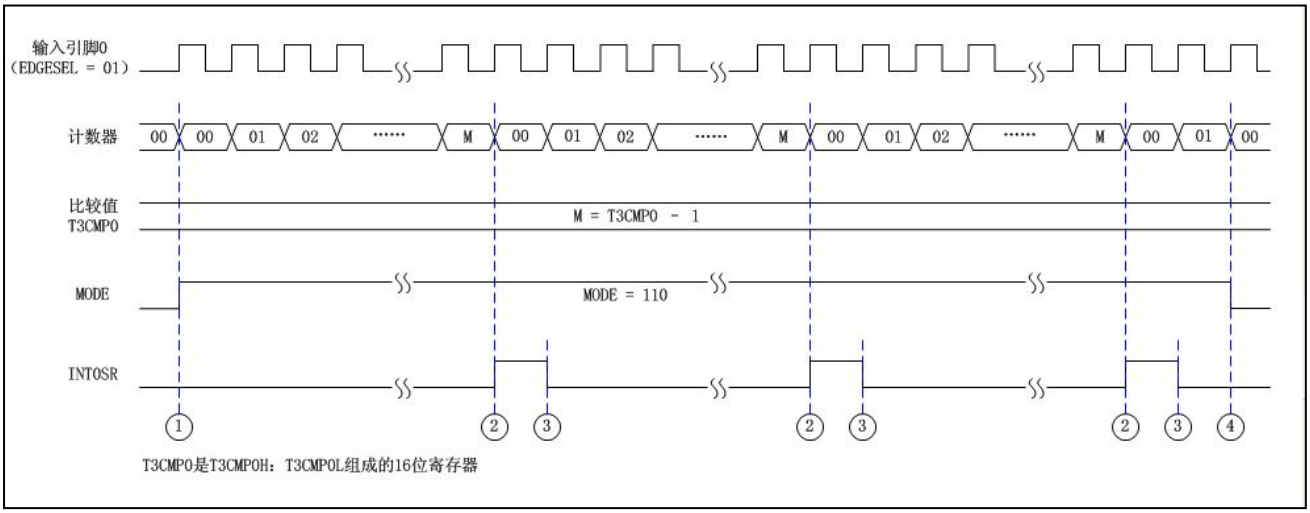
T3CR 寄存器的 INT0EN、INT1EN 位是 Timer3 中断使能位，T3SR 寄存器的 INT0SR、INT1SR 位是 Timer3 中断标志位。

9. 工作模式选择：

配置 T3CR 寄存器的 MODE<2:0>位 = 110 或 111，选择 Timer3 的工作模式，配置完成后，Timer3 即开始工作。

Timer3 外部事件计数工作时序如图 11-3 所示。

图 11-3： Timer3 外部事件计数工作时序图



波形说明：

① 设置计数器模式 MODE<2:0>位 = 110 后，计数器开始计数；

② 当计数器值等于 M 时，产生 Timer3 中断 0(即 T3SR 寄存器的 INT0SR 位置 1)；

③ 软件处理并清除中断；

④ 设置计数器模式 MODE<2:0>位 = 000 后，计数器清零并停止计数。

注： 1、当 T3CMP0 = 0 时，按 T3CMP0 = 65536 计算，此时 M = T3CMP0 - 1 = 65535。

2、计数时钟为输入引脚 0 的有效沿，只有当输入引脚脉冲宽度大于两个时钟宽度才认为是一个有效边沿，以消除干扰；

3、计数器在每个输入引脚的有效沿都会加 1；

4、T3CKCR 寄存器的 EDGESEL 位可以选择有效沿。

5、当 T3CMP0 = 1 时，不产生 Timer3 中断 0。

11.1.5. 单脉冲输出的操作步骤

配置 Timer3 为单脉冲输出功能的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 3 输出/输入 0 功能，并且配置 PnOE 寄存器，将定时器 3 输出或输入 0 引脚方向置为输出。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer3 计数时钟参数配置：

- 计数时钟选择：

T3CKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer3 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer3 的时钟。

- 计数时钟分频配置：

T3CKCR 寄存器的 CLKDIV<2:0>位可以对选择的时钟源进行分频。此时不能配置为输入引脚的有效沿作为计数时钟选项。

- 输入引脚的有效沿设置：

将 T3CKCR 寄存器的 EDGESEL<1:0>位配置为 00，输入引脚的有效沿选择为禁止设置。

- 计数时钟使能：

将 T3CKCR 寄存器的 CLKEN 位置 1，使能 Timer3 计数时钟。

4. Timer3 输出配置：

- 使能单脉冲输出模式：

将 T3OCR 寄存器的 SPOM 位置 1，选择单脉冲输出模式。

- 输出初始状态选择：

T3OCR 寄存器的 OUTINIT<1:0>位控制 Timer3 输出初始状态，只对方波输出、单脉冲输出有效，对 PWM 输出无效。

- 计数器与 T3CMP0、T3CMP1 寄存器组匹配输出翻转配置：

将 T3OCR 寄存器的 INV0EN、INV1EN 位置 1，使能计数器与(T3CMP1H：T3CMP1L)匹配输出翻转，计数器与(T3CMP0H：T3CMP0L)匹配输出翻转。

5. T3CMP0、T3CMP1 寄存器组的模式选择：

将 T3CR 寄存器的 CMP0M、CMP1M 位同时清 0，配置 T3CMP0 寄存器组、T3CMP1 寄存器组都作为比较寄存器。

6. 单脉冲宽度配置：

Timer3 的单脉冲宽度是通过 T3CMP0H：T3CMP0L 与 T3CMP1H：T3CMP1L 寄存器进行配置，具体计算如下：

单脉冲宽度 = (T3CMP0H：T3CMP0L - T3CMP1H：T3CMP1L) x Timer3 时钟周期

= (T3CMP0H：T3CMP0L - T3CMP1H：T3CMP1L) x Timer3 计数时钟分频 x Timer3 计数时钟周期

Timer3 计数时钟周期由选择的计数时钟决定，共有三种时钟：

- 内部高速 OSC48M 的 3 分频
- 内部低速 800K
- 外部 32.768K 晶振(必须确保外接了 32.768K 晶振)

当(T3CMP0H ： T3CMP0L)= 0 时，按(T3CMP0H ： T3CMP0L)= 65536 计算。

其中(T3CMP1H： T3CMP1L)必须小于(T3CMP0H：T3CMP0L)，当输出初始电平为低电平，单脉冲宽度是高电平宽度；当输出初始电平为高电平，单脉冲宽度是低电平宽度。

> 注： (T3CMP0H：T3CMP0L)在 Timer3 工作期间是不能改变的，所以想要改变(T3CMP0H：T3CMP0L)必须先配置 T3CR 寄存器的 MODE 位 = 000，等待 T3SR 寄存器的 WREN 位置 1 后，再修改(T3CMP0H：T3CMP0L)。

7. 清 Timer3 中断标志位

使能 Timer3 中断前，先将 T3SR 寄存器的 INT0SR、INT1SR 位清 0，清 Timer3 中断标志位。

8. 中断控制：

Timer3 有两个中断，Timer3 中断 0、Timer3 中断 1，Timer3 中断 1 只有在单脉冲输出、外部电平宽度测量模式下才可以产生，Timer3 中断 0 在所有模式下都可以产生。具体请看 11.3 小节“Timer3 中断”。

T3CR 寄存器的 INT0EN、INT1EN 位是 Timer3 中断使能位，T3SR 寄存器的 INT0SR、INT1SR 位是 Timer3 中断标志位。

9. 工作模式选择：

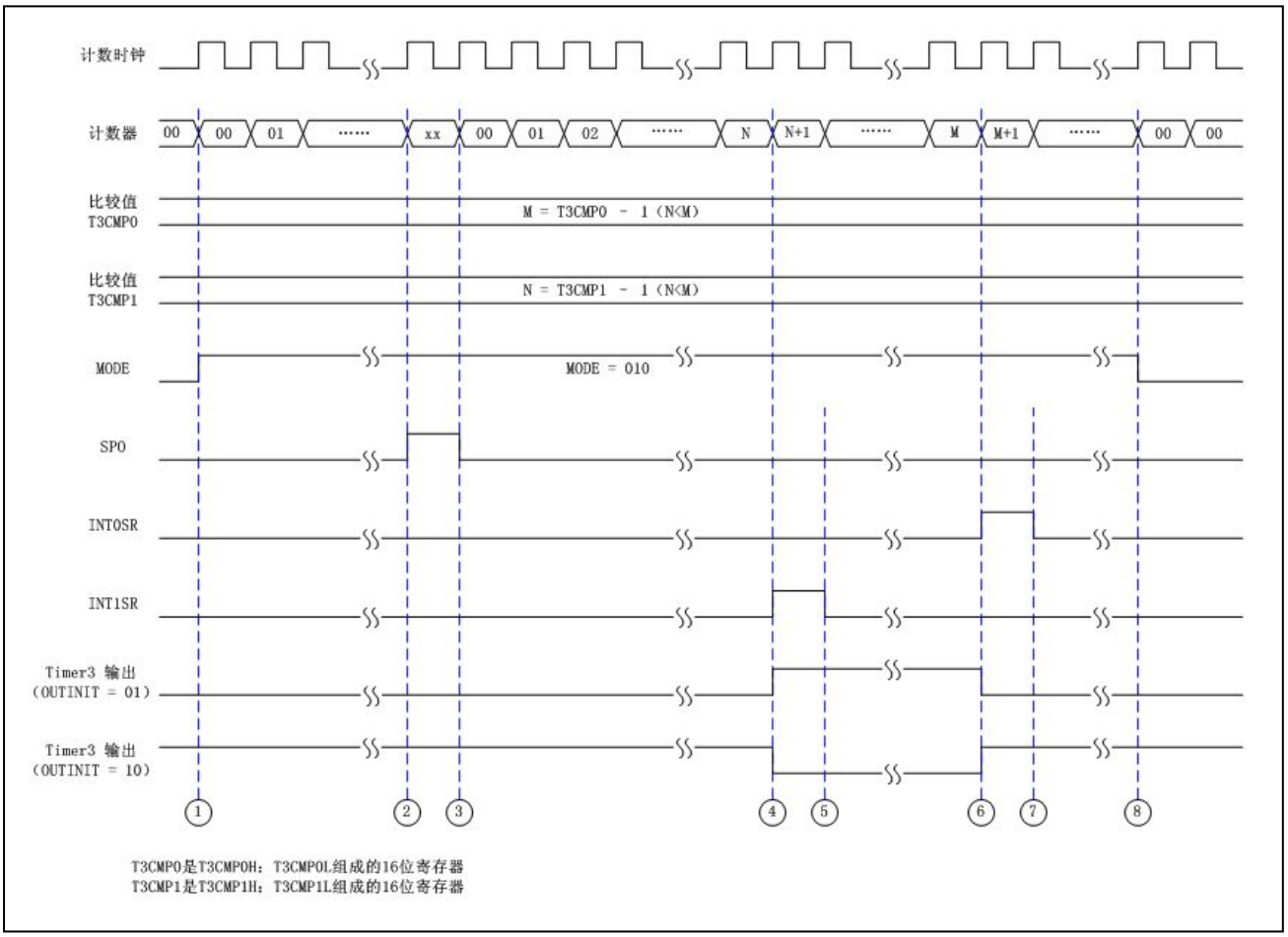
配置 T3CR 寄存器的 MODE<2:0>位 = 010，选择 Timer3 的工作模式，配置完成后，Timer3 即开始工作。

10. 触发单脉冲输出：

每次写 T3OCR 寄存器的 SPO 位为 1，即启动一次单脉冲输出，启动后应查询 T3SR 寄存器的 INT0SR 位是否置 1，判断单脉冲输出是否完成，完成一次单脉冲输出后才可以启动下一次。

Timer3 单脉冲输出工作时序如图 11-4 所示。

图 11-4： Timer3 单脉冲输出工作时序图



波形说明：

① 设置计数器模式 MODE<2:0>位 = 010 后，计数器开始计数；

② 软件写 SP0 为 1；

③ 延时一段时间后，计数器清零，启动单脉冲输出；

④ 当计数器值等于 N，产生 Timer3 中断 1(即 T3SR 寄存器的 INT1SR 位置 1)，同时脉冲信号翻转；

⑤ 软件处理并清除中断 INT1SR；

⑥ 当计数器值等于 M，产生 Timer3 中断 0(即 T3SR 寄存器的 INT0SR 位置 1)，同时脉冲信号翻转；

⑦ 软件处理并清除中断 INT0SR；

⑧ 设置计数器模式 MODE<2:0>位 = 000 后，计数器清零并停止计数。

注： 写一次 SP0 为 1，只能启动 1 个脉冲输出。想发送多个脉冲必须重复写 SP0。

11.1.6. 外部电平宽度测量

外部电平宽度有三种工作方式：

- 使用一组捕获寄存器测量一个输入电平的宽度
- 使用两组捕获寄存器测量两个输入电平的宽度
- 使用两组捕获寄存器测量一个输入电平的宽度

外部电平宽度的工作原理是当外部输入有效沿信号，会产生中断，同时计数器的值捕捉到捕捉寄存器(T3CMP0、T3CMP1 两组寄存器)中；在中断中读取捕捉寄存器的值，连续两次中断读取值的差值可以得出电平宽度。

11.1.6.1. 使用一组捕获寄存器测量一个输入电平的宽度的操作步骤

此模式下，使用的是 T3CMP1H：T3CMP1L 这组寄存器作为捕获寄存器，测量定时器 3 输入 0 的电平宽度。

配置 Timer3 为使用一组捕获寄存器测量一个输入电平的宽度的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 3 输出/输入 0 功能，并且配置 PnOE 寄存器，将定时器 3 输出或输入 0 引脚方向置为输入。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer3 计数时钟参数配置：

- 计数时钟选择：

T3CKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer3 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer3 的时钟。

- 计数时钟分频配置：

T3CKCR 寄存器的 CLKDIV<2:0>位可以对选择的时钟源进行分频。此时不能配置为输入引脚的有效沿作为计数时钟选项。

- 输入引脚的有效沿设置：

配置 T3CKCR 寄存器的 EDGESEL<1:0>位 = 11，选择输入引脚的有效沿为双边沿。

- 计数时钟使能：

将 T3CKCR 寄存器的 CLKEN 位置 1，使能 Timer3 计数时钟。

4. T3CMP0、T3CMP1 寄存器组的模式选择：

将 T3CR 寄存器的 CMP0M 位清 0，CMP1M 位置 1，配置 T3CMP0 寄存器组作为比较寄存器，T3CMP1 寄存器组作为捕获寄存器。

5. 清 Timer3 中断标志位

使能 Timer3 中断前，先将 T3SR 寄存器的 INT0SR、INT1SR 位清 0，清 Timer3 中断标志位。

6. 中断控制：

Timer3 有两个中断，Timer3 中断 0、Timer3 中断 1，Timer3 中断 1 只有在单脉冲输出、外部电平宽度测量模式下才可以产生，Timer3 中断 0 在所有模式下都可以产生。具体请看 11.3 小节“Timer3 中断”。

T3CR 寄存器的 INT0EN、INT1EN 位是 Timer3 中断使能位，T3SR 寄存器的 INT0SR、INT1SR 位是 Timer3 中断标志位。

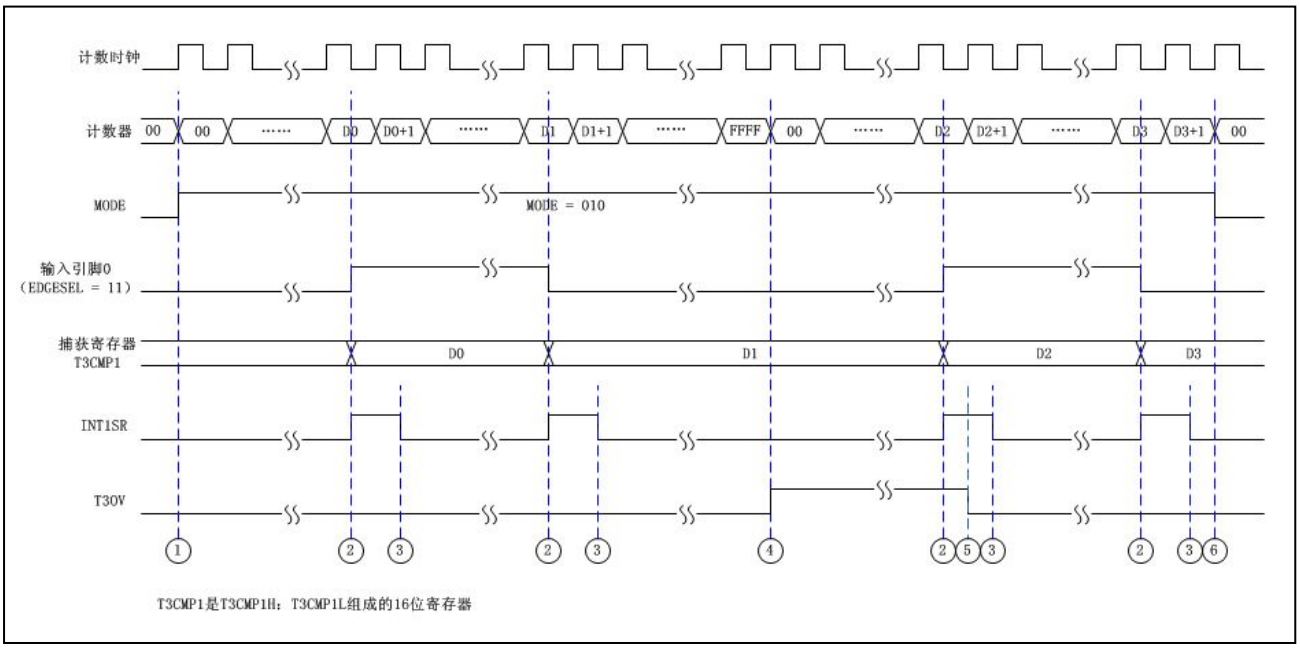
7. 工作模式选择：

配置 T3CR 寄存器的 MODE<2:0>位 = 010，选择 Timer3 的工作模式，配置完成后，Timer3 即开始工作。

配置双边沿为有效沿，如果使能了 Timer3 中断 1，在 Timer3 输入 0 引脚外部信号的上升沿、下降沿都会产生 Timer3 中断 1，连续 3 次中断就可以得出一个低电平宽度时间，一个高电平宽度时间。(具体哪个是高电平时间或低电平时间必须从实际情况分析)

Timer3 使用一组捕获寄存器测量一个输入电平的宽度工作时序如图 11-5 所示。

图 11-5： Timer3 使用一组捕获寄存器测量一个输入电平的宽度工作时序图



波形说明：

① 设置计数器模式 MODE<2:0>位 = 010 后，计数器开始计数；

② 检测到输入引脚 0 的有效沿(这里必须设置有效沿为双边沿)，将计数器的值锁存到捕获寄存器 T3CMP1H:T3CMP1L 中，同时产生 Timer3 中断 1(即 T3SR 寄存器的 INT1SR 位置 1)；

③ 软件处理并清除中断 INT1SR；

④ 计数器溢出；软件应清除其状态。(软件在中断处理程序中读取捕获寄存器的值时，应同时关注溢出状态位，若有溢出，计算脉冲宽度时，应加上 0xFFFF)；

⑤ 软件处理后清除计数器溢出标志位；

⑥ 设置计数器模式 MODE<2:0>位 = 000 后，计数器清零并停止计数。

注： 1、输入引脚的有效沿必须设置为双边沿，这样检测到上升沿、下降沿都会产生 Timer3 中断 1，并锁存计数器的值。

2、若输入引脚是周期稳定的信号，可以根据相邻 3 次中断所捕获到的数据，得到输入引脚的信号的周期/频率。

3、理论上是无法区分出得到的是高电平宽度还是低电平宽度，必须根据实际上输入信号的特征来区分。

11.1.6.2. 使用两组捕获寄存器测量两个输入电平的宽度的操作步骤

此模式下，使用的是 T3CMP0H、T3CMP0L 这组寄存器作为捕获寄存器，测量定时器 3 输入 1 的电平宽度；T3CMP1H、T3CMP1L 这组寄存器作为捕获寄存器，测量定时器 3 输入 0 的电平宽度。

配置 Timer3 为使用两组捕获寄存器测量两个输入电平的宽度的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 3 输出/输入 0 功能，并且配置 PnOE 寄存器，将定时器 3 输出或输入 0 引脚方向置为输入。

将 I/O 配置成定时器 3 输入 1 功能，此时定时器 3 输入 1 的 I/O 方向自动配置为输入。不需要配置对应的 PnOE 寄存器，选择定时器 3 输入 1 引脚方向为输入。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer3 计数时钟参数配置：

- 计数时钟选择：

T3CKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer3 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer3 的时钟。

- 计数时钟分频配置：

T3CKCR 寄存器的 CLKDIV<2:0>位可以对选择的时钟源进行分频。此时不能配置为输入引脚的有效沿作为计数时钟选项。

- 输入引脚的有效沿设置：

配置 T3CKCR 寄存器的 EDGESEL<1:0>位 = 11，选择输入引脚的有效沿为双边沿。

- 计数时钟使能：

将 T3CKCR 寄存器的 CLKEN 位置 1，使能 Timer3 计数时钟。

4. Timer3 模式配置：

- T3CMP0、T3CMP1 寄存器组的模式选择：

将 T3CR 寄存器的 CMP0M、CMP1M 位都置 1，配置 T3CMP0、T3CMP1 寄存器组都作为捕获寄存器。

- T3CMP0 寄存器组捕获时刻选择：

将 T3CR 寄存器的 CMP0CAP 位清 0，选择 T3CMP0 寄存器组捕获时刻为在输入引脚 1 有效沿进行捕获。

5. 清 Timer3 中断标志位

使能 Timer3 中断前，先将 T3SR 寄存器的 INT0SR、INT1SR 位清 0，清 Timer3 中断标志位。

6. 中断控制：

Timer3 有两个中断，Timer3 中断 0、Timer3 中断 1，Timer3 中断 1 只有在单脉冲输出、外部电平宽度测量模式下才可以产生，Timer3 中断 0 在所有模式下都可以产生。具体请看 11.3 小节“Timer3 中断”。

T3CR 寄存器的 INT0EN、INT1EN 位是 Timer3 中断使能位，T3SR 寄存器的 INT0SR、INT1SR 位是 Timer3 中断标志位。

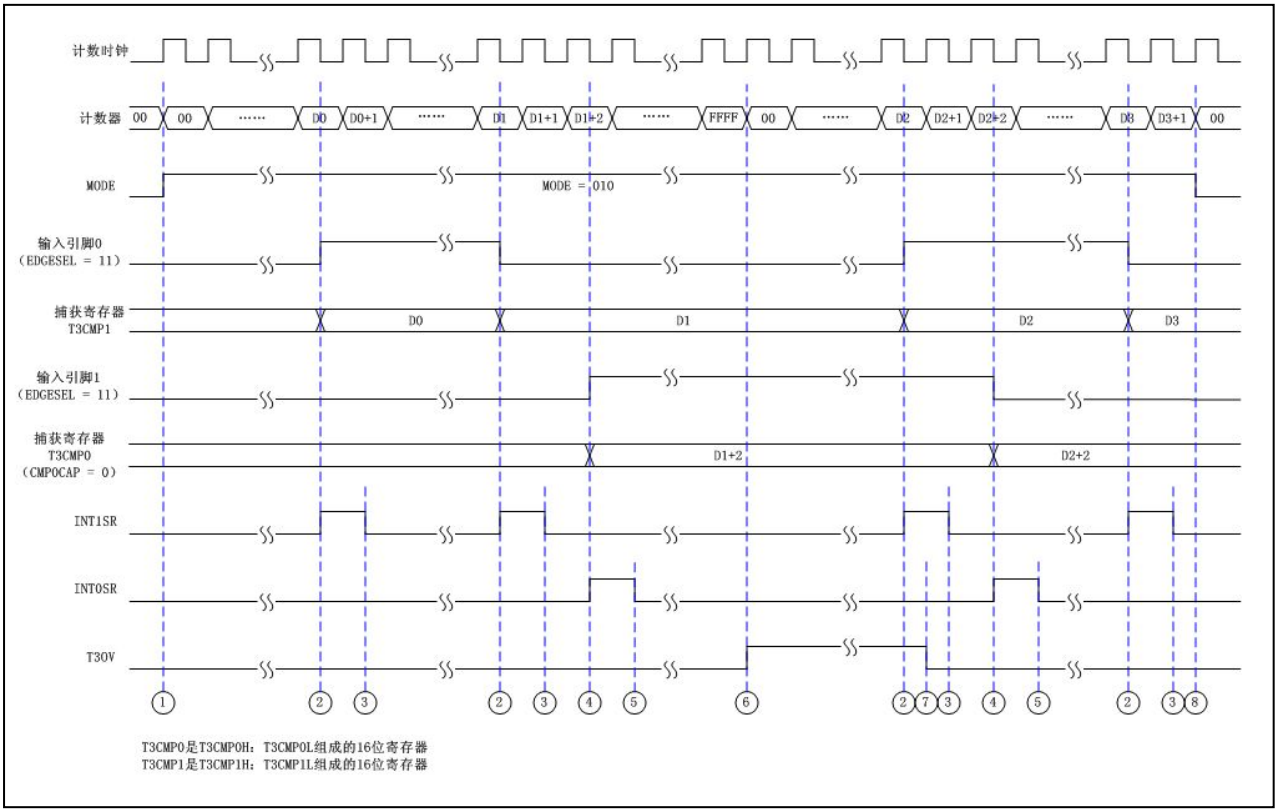
7. 工作模式选择：

配置 T3CR 寄存器的 MODE<2:0>位 = 010，选择 Timer3 的工作模式，配置完成后，Timer3 即开始工作。

配置双边沿为有效沿，如果使能了 Timer3 中断 0、中断 1 中断，在 Timer3 输入 0 引脚外部信号的上升沿、下降沿产生 Timer3 中断 1，Timer3 输入 1 引脚外部信号的上升沿、下降沿产生 Timer3 中断 0，连续 3 次中断就可以得出一个低电平宽度时间，一个高电平宽度时间。(具体哪个是高电平时间或低电平时间必须从实际情况分析)

Timer3 使用两组捕获寄存器测量两个输入电平的宽度工作时序如图 11-6 所示。

图 11-6： Timer3 使用两组捕获寄存器测量两个输入电平的宽度工作时序图



波形说明：

① 设置计数器模式 MODE<2:0>位 = 010 后，计数器开始计数；

② 检测到输入引脚 0 的有效沿(这里必须设置有效沿为双边沿)，将计数器的值锁存到捕获寄存器 T3CMP1H:T3CMP1L 中，同时产生 Timer3 中断 1(即 T3SR 寄存器的 INT1SR 位置 1)；

③ 软件处理并清除中断 INT1SR；

④ 检测到输入引脚 1 的有效沿(这里必须设置有效沿为双边沿)，将计数器的值锁存到捕获寄存器 T3CMP0H:T3CMP0L 中，同时产生 Timer3 中断 0(即 T3SR 寄存器的 INT0SR 位置 1)；

⑤ 软件处理并清除中断 INT0SR；

⑥ 计数器溢出；

⑦ 软件处理后清除计数器溢出标志位；

⑧ 设置计数器模式 MODE<2:0>位 = 000 后，计数器清零并停止计数。

注： 1、输入引脚的有效沿必须设置为双边沿，这样检测到上升沿、下降沿都会产生中断并锁存计数器的值。

2、若输入引脚是周期稳定的信号，可以根据相邻 3 次中断所捕获到的数据，得到输入引脚的信号的周期/频率。

3、理论上是无法区分出得到的是高电平宽度还是低电平宽度，必须根据实际上输入信号的特征来区分。

11.1.6.3. 使用两组捕获寄存器测量一个输入电平的宽度

此模式下，使用的是 T3CMP0H：T3CMP0L、T3CMP1H：T3CMP1L 这两组寄存器作为捕获寄存器，测量定时器 3 输入 0 的电平宽度。

配置 Timer3 为使用两组捕获寄存器测量一个输入电平的宽度的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成定时器 3 输出/输入 0 功能，并且配置 PnOE 寄存器，将定时器 3 输出或输入 0 引脚方向置为输入。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. Timer 模块时钟使能：

将 MDLCKCR 寄存器的 TIMEREN 位置 1，使能 Timer 模块时钟。

3. Timer3 计数时钟参数配置：

- 计数时钟选择：

T3CKCR 寄存器的 CLKSEL 位可以选择内部高速 OSC48M 的 3 分频或者低速时钟作为 Timer3 的时钟。

选择低速时钟时，还可以通过 MDLCKCR 寄存器的 32KEN 位选择内部低速 800K 或外部 32.768K 晶振作为 Timer3 的时钟。

- 计数时钟分频配置：

T3CKCR 寄存器的 CLKDIV<2:0>位可以对选择的时钟源进行分频。此时不能配置为输入引脚的有效沿作为计数时钟选项。

- 输入引脚的有效沿设置：

T3CKCR 寄存器的 EDGESEL<1:0>位可以选择输入引脚的有效沿，只能选择输入引脚的有效沿为上升沿或下降沿，不能配置为其他选项。

- 计数时钟使能：

将 T3CKCR 寄存器的 CLKEN 位置 1，使能 Timer3 计数时钟。

4. Timer3 模式配置：

- T3CMP0、T3CMP1 寄存器组的模式选择：

将 T3CR 寄存器的 CMP0M、CMP1M 位都置 1，配置 T3CMP0、T3CMP1 寄存器组都作为捕获寄存器。

- T3CMP0 寄存器组捕获时刻选择：

将 T3CR 寄存器的 CMP0CAP 位置 1，选择 T3CMP0 寄存器组捕获时刻为在输入引脚 0 有效沿的反沿进行捕获。

5. 清 Timer3 中断标志位

使能 Timer3 中断前，先将 T3SR 寄存器的 INT0SR、INT1SR 位清 0，清 Timer3 中断标志位。

6. 中断控制：

Timer3 有两个中断，Timer3 中断 0、Timer3 中断 1，Timer3 中断 1 只有在单脉冲输出、外部电平宽度测量模式下才可以产生，Timer3 中断 0 在所有模式下都可以产生。具体请看 11.3 小节“Timer3 中断”。

T3CR 寄存器的 INT0EN、INT1EN 位是 Timer3 中断使能位，T3SR 寄存器的 INT0SR、INT1SR 位是 Timer3 中断标志位。

7. 工作模式选择：

配置 T3CR 寄存器的 MODE<2:0>位 = 010，选择 Timer3 的工作模式，配置完成后，Timer3 即开始工作。

如果使能了 Timer3 中断 0、中断 1：

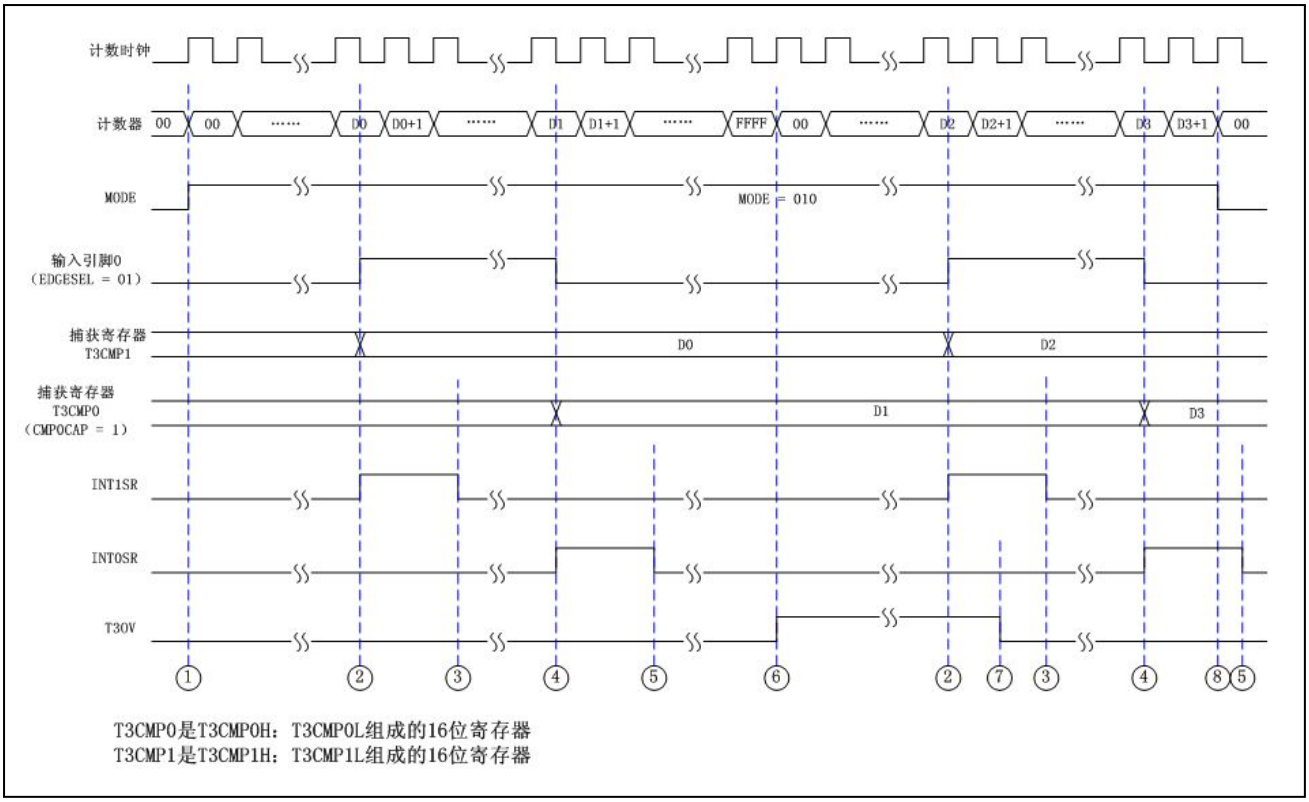
1）若配置输入引脚有效沿为上升沿，则在 Timer3 输入 0 引脚外部信号的上升沿，产生 Timer3 中断 1；在 Timer3 输入 0 引脚外部信号的下降沿，产生 Timer3 中断 0，；

2）若配置输入引脚有效沿为下降沿，则在 Timer3 输入 0 引脚外部信号的下降沿，产生 Timer3 中断 1；在 Timer3 输入 0 引脚外部信号的上升沿，产生 Timer3 中断 0。

3）然后在 Timer3 中断 1 记录 T3CMP1H、T3CMP1L 的值，在 Timer3 中断 0 记录 T3CMP0H、T3CMP0L 的值，通过连续 3 次中断就可以得出高低电平的时间。

Timer3 使用两组捕获寄存器测量一个输入电平的宽度工作时序如图 11-7 所示。

图 11-7： Timer3 使用两组捕获寄存器测量一个输入电平的宽度工作时序图



波形说明：

① 设置计数器模式 MODE<2:0>位 = 010 后，计数器开始计数；

② 检测到输入引脚 0 的有效沿，将计数器的值锁存到捕获寄存器 T3CMP1H:T3CMP1L 中，同时产生 Timer3 中断 1(即 T3SR 寄存器的 INT1SR 位置 1)；

③ 软件处理并清除中断 INT1SR，软件应在中断处理程序中将寄存器 T3CMP1H:T3CMP1L 的值保存到变量 A 中；

④ 检测到输入引脚 0 的有效沿的反沿，将计数器的值锁存到捕获寄存器 T3CMP0H:T3CMP0L 中，同时产生 Timer3 中断 0(即 T3SR 寄存器的 INT0SR 位置 1)；

⑤ 软件处理并清除中断 INT0SR，软件应在中断处理程序中将寄存器 T3CMP0H:T3CMP0L 的值保存到变量 B 中；

⑥ 计数器溢出；

⑦ 软件处理后清除计数器溢出标志位；

⑧ 设置计数器模式 MODE<2:0>位 = 000 后，计数器清零并停止计数。

注： 1、输入引脚的有效沿可以设置为上升沿/下降沿，这样有效沿的反沿就是下降沿/上升沿。在有效沿会产生 Timer3 中断 1，并锁存计数器的值到 T3CMP1H:T3CMP1L 寄存器；在反沿产生 Timer3 中断 0，并锁存计数器的值到 T3CMP0H:T3CMP0L 寄存 器。

2、若输入引脚是周期稳定的信号，可以根据相邻 3 次中断所捕获到的数据，得到输入引脚的信号的周期/频率。

3、本模式可以分辨出得到的是高电平宽度还是低电平宽度。

11.2. Timer3 寄存器的定义

以下寄存器用于控制 Timer3 的操作。

寄存器 11-1： T3CR：Timer3 模式控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CMP1M	CMP0M	INT1EN	INT0EN	CMP0CAP	MODE2	MODE1	MODE0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 CMP1M：T3CMP1 寄存器组的模式选择位
0 = T3CMP1 寄存器组作为比较寄存器
1 = T3CMP1 寄存器组作为捕获寄存器

bit 6 CMP0M：T3CMP0 寄存器组的模式选择位
0 = T3CMP0 寄存器组作为比较寄存器
1 = T3CMP0 寄存器组作为捕获寄存器

bit 5 INT1EN：Timer3 中断 1 使能位
0 = 禁止 Timer3 中断 1 中断
1 = 使能 Timer3 中断 1 中断

bit 4 INT0EN：Timer3 中断 0 使能位
0 = 禁止 Timer3 中断 0 中断
1 = 使能 Timer3 中断 0 中断

bit 3 CMP0CAP：T3CMP0 寄存器组捕获时刻控制位
0 = T3CMP0 寄存器组在输入引脚 1 有效沿进行捕获
1 = T3CMP0 寄存器组在输入引脚 0 有效沿的反沿进行捕获

bit 2-0 MODE<2:0>：Timer3 模式控制位
000 = Timer3 停止操作，输出无变化，无中断产生
001 = Timer3 停止操作，输出无变化，无中断产生
010 = 自由运行模式(计数器溢出清零)
011 = 保留
100 = 由输入引脚的有效沿清除计数器的模式
101 = 由输入引脚的有效沿清除计数器的模式
110 = 由比较匹配清除计数器的模式，当比较匹配时，输出翻转
111 = 由比较匹配清除计数器的模式，无波形输出

寄存器 11-2： T3CKCR：Timer3 时钟控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CLKEN	CLKSEL	EDGESEL1	EDGESEL0	—	CLKDIV2	CLKDIV1	CLKDIV0
R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 CLKEN：Timer3 计数时钟使能位
0 = 关闭 Timer3 计数时钟
1 = 打开 Timer3 计数时钟

bit 6 CLKSEL：Timer3 计数时钟选择位
0 = 选择内部高速 OSC48M 的 3 分频作为 Timer3 计数时钟
1 = 选择低速时钟作为 Timer3 计数时钟(32KEN 位选择 OSC800K 或外部 32.768K)

bit 5-4 EDGESEL<1:0>：Timer3 输入引脚的有效沿选择位
00 = 禁止设置
01 = 上升沿
10 = 下降沿
11 = 双边沿

bit 3 未定义：读为 0

bit 2-0 CLKDIV<2:0>：Timer3 计数时钟分频选择位
000 = 2 分频
001 = 4 分频
010 = 8 分频
011 = 16 分频
1XX = 输入引脚的有效沿作为计数时钟

寄存器 11-3： T3OCR：Timer3 输出控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	SPO	SPOM	OUTINIT1	OUTINIT0	INV1EN	INV0EN
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未定义：读为 0

bit 5 SPO：单脉冲输出触发位

写 1 后触发单脉冲输出，自动清零，该位读一直为 0。

bit 4 SPOM：单脉冲输出模式使能位

0 = 连续脉冲输出模式

1 = 单脉冲输出模式

bit 3-2 OUTINIT<1:0>：Timer3 输出初始状态(方波输出、单脉冲输出才有效)

00 = 保持原状态

01 = 初始输出低电平

10 = 初始输出高电平

11 = 保持原状态

bit 1 INV1EN：Timer3 计数器与 T3CMP1 匹配输出翻转使能位

0 = 当 Timer3 计数器与 T3CMP1 匹配时，输出不翻转

1 = 当 Timer3 计数器与 T3CMP1 匹配时，输出翻转

bit 0 INV0EN：Timer3 计数器与 T3CMP0 匹配输出翻转使能位

0 = 当 Timer3 计数器与 T3CMP0 匹配时，输出不翻转

1 = 当 Timer3 计数器与 T3CMP0 匹配时，输出翻转

寄存器 11-4： T3CMP0L：Timer3 比较寄存器 0 低 8 位

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 T3CMP0L<7:0>：Timer3 比较寄存器 0 低 8 位

Timer3 比较寄存器 0 低 8 位，在 Timer3 工作期间，不能修改此寄存器

注： T3CMP0H 和 T3CMP0L 组成 16 位的比较寄存器 0(T3CMP0)。

寄存器 11-4： T3CMP0H：Timer3 比较寄存器 0 高 8 位(续)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 T3CMP0H<7:0>：Timer3 比较寄存器 0 高 8 位

Timer3 比较寄存器 0 高 8 位，在 Timer3 工作期间，不能修改此寄存器

注： 1、T3CMP0H 和 T3CMP0L 组成 16 位的比较寄存器 0(T3CMP0)；

2、在比较模式下，T3CMP0 存储的是比较值，由软件写入。且在定时器工作期间，不能修改该寄存器的值；

3、在捕获模式下，T3CMP0 存储的是捕获值，由硬件写入；

4、T3SR 寄存器的 WREN 位是 T3CMP0H 和 T3CMP0L 可写状态位，为 1 表示可以修改 T3CMP0H 和 T3CMP0L 寄存器，为 0 表示不能修改 T3CMP0H 和 T3CMP0L 寄存器。所以正确的修改 T3CMP0 寄存器如范例 8-1 所示。

范例 9-1：修改 T3CMP0H、T3CMP0L 寄存器：

```
T3CR_MODE = 0x00;                //Timer3 停止工作
while(0x00 = T3SR_WREN){;}       //等待 T3CMP0H、T3CMP0L 写使能
T3CMP0L = 0x60;                  //修改 T3CMP0L 寄存器
T3CMP0H = 0x02;                  //修改 T3CMP0H 寄存器
……                               //其他处理
T3CR_MODE = 0x06;                //Timer3 开始工作
```

寄存器 11-5： T3CMP1L：Timer3 比较寄存器 1 低 8 位

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 T3CMP1L<7:0>：Timer3 比较寄存器 1 低 8 位

Timer3 比较寄存器 1 低 8 位，在 Timer3 工作期间，可以修改此寄存器。

T3CMP0H、T3CMP0L 决定输出 PWM 的周期，T3CMP1H、T3CMP1L 决定输出 PWM 的占空比，在 Timer3 工作期间，可以修改 T3CMP1H、T3CMP1L 寄存器改变输出 PWM 的占空比。

注： T3CMP1H 和 T3CMP1L 组成 16 位的比较寄存器 1(T3CMP1)。

寄存器 11-5： T3CMP1H：Timer3 比较寄存器 1 高 8 位(续)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 T3CMP1H<7:0>：Timer3 比较寄存器 1 高 8 位

Timer3 比较寄存器 1 高 8 位，在 Timer3 工作期间，可以修改此寄存器。

T3CMP0H、T3CMP0L 决定输出 PWM 的周期，T3CMP1H、T3CMP1L 决定输出 PWM 的占空比，在 Timer3 工作期间，可以修改 T3CMP1H、T3CMP1L 寄存器改变输出 PWM 的占空比。

注： 1、T3CMP1H 和 T3CMP1L 组成 16 位的比较寄存器 1(T3CMP1)；

2、在比较模式下，T3CMP1 存储的是比较值，由软件写入；

3、在捕获模式下，T3CMP1 存储的是捕获值，由硬件写入。

寄存器 11-6： T3SR：Timer3 状态寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	WREN	T3OV	INT1SR	INT0SR
U-0	U-0	U-0	U-0	R-1	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未定义：读为 0

bit 3 WREN：T3CMP0 可写标志位
0 = Timer3 正在工作，不可写 T3CMP0
1 = Timer3 停止工作，可写 T3CMP0，关闭 Timer3 后，应等待 WREN 为 1 后，再写 T3CMP0

bit 2 T3OV：Timer3 计数器溢出状态位
0 = Timer3 计数器没有溢出过
1 = Timer3 计数器已经溢出

bit 1 INT1SR：Timer3 中断 1 标志位
0 = 没有产生 Timer3 中断 1 条件
1 = 产生了 Timer3 中断 1 条件

bit 0 INT0SR：Timer3 中断 0 标志位
0 = 没有产生 Timer3 中断 0 条件
1 = 产生了 Timer3 中断 0 条件

11.3. Timer3 中断

Timer3 有两个中断，Timer3 中断 0 和 Timer3 中断 1，产生 Timer3 中断有以下几种种方式：

- 间隔定时/方波输出/PWM 输出/外部事件计数模式下，当 Timer3 计数器值等于(T3CMP0H：T3CMP0L－1)时，产生 Timer3 中断 0；
- 单脉冲输出模式下，当 Timer3 计数器值等于(T3CMP0H：T3CMP0L－1)时，产生 Timer3 中断 0，当 Timer3 计数器值等于(T3CMP1H：T3CMP1L－1)时，产生 Timer3 中断 1。
- 外部电平宽度测量模式下，产生 Timer3 中断 0、Timer3 中断 1。具体请查看相应模式的介绍。

只要有 Timer3 中断条件产生，不论是否使能 Timer3 中断，T3SR 寄存器的 INT0SR、INT1SR 中断标志位都会置 1，INT0SR、INT1SR 位必须在软件中清零。Timer3 的中断使能位是 T3CR 寄存器的 INT0EN、INT1EN 位。

注： 在休眠模式下，Timer3 中断无法唤醒处理器。
在空闲模式下，Timer3 中断可以唤醒处理器。

12. PWM 模块

PWM 模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
PWM1CR	0x68FF	XRAM	PWM1控制寄存器	0000 0000
PWM1LRL	0x6AFF	XRAM	PWM1脉宽调制寄存器0低8位	0000 0000
PWM1LRH	0x6BFF	XRAM	PWM1脉宽调制寄存器0高8位	0000 0000
PWM1HRL	0x6CFF	XRAM	PWM1脉宽调制寄存器1低8位	0000 0000
PWM1HRH	0x6DFF	XRAM	PWM1脉宽调制寄存器1高8位	0000 0000
PWM2CR	0x69FF	XRAM	PWM2控制寄存器	0000 0000
PWM2LR	0x6EFF	XRAM	PWM2脉宽调制寄存器0	0000 0000
PWM2HR	0x6FFF	XRAM	PWM2脉宽调制寄存器1	0000 0000

PWM 模块包括两个独立的模块：

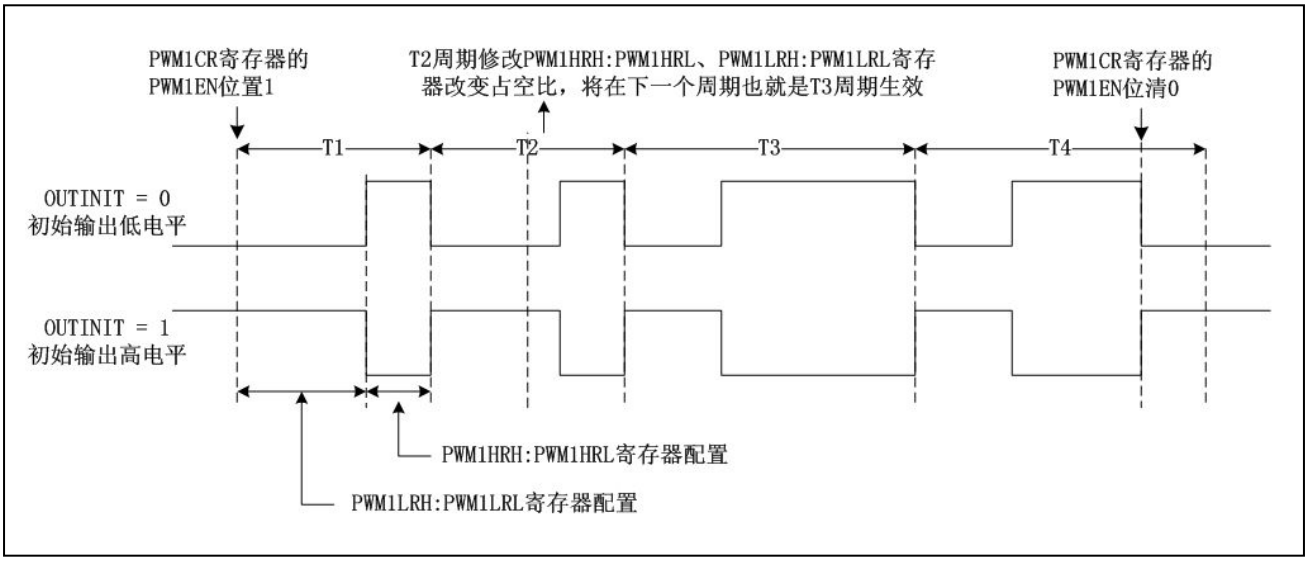
- 16 位 PWM(PWM1)
- 8 位 PWM(PWM2)

PWM 模块的时基就是就是 OSC48M 经过 CKDIVCR 寄存器的 SYSDIV<5:0>位分频后的时钟。

PWM1 与 PWM2 模块的工作原理完全一样，唯一不同的是 PWM1 是 16 位的，而 PWM2 是 8 位的。

PWM1 输出的工作原理如下图所示。

图 12-1： PWM1 输出的工作原理



注： 1、初始输出低电平：低电平宽度 = (PWM1LRH:PWM1LRL + 1) × $2^{(CLKDIV+1)}$ × Tosc48M × (SYSDIV + 1)

高电平宽度 = (PWM1HRH:PWM1HRL + 1) × $2^{(CLKDIV+1)}$ × Tosc48M × (SYSDIV + 1)

2、初始输出高电平：高电平宽度 = (PWM1LRH:PWM1LRL + 1) × $2^{(CLKDIV+1)}$ × Tosc48M × (SYSDIV + 1)

低电平宽度 = (PWM1HRH:PWM1HRL + 1) × $2^{(CLKDIV+1)}$ × Tosc48M × (SYSDIV + 1)

12.1. PWM 的操作步骤

PWM1、PWM2 的操作步骤完全一样，这里以 PWM1 为例。

配置 PWM1 的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成 PWM1 输出功能时，会自动将引脚方向配置为输出。不需要配置对应的 PnOE 寄存器，选择 PWM1 输出引脚方向为输出。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. PWM 模块时钟使能：

将 MDLCKCR 寄存器的 PWMEN 位置 1，使能 PWM 模块时钟。

3. PWM 模块参数配置：

- 输出初始状态选择：

配置 PWM1CR 寄存器的 OUTINIT 位，选择输出初始状态。

- 计数时钟分频选择：

配置 PWM1CR 寄存器的 CLKDIV<2:0>位，选择计数时钟分频。

- 调制占空比：

配置 PWM1LRH：PWM1LRL 与 PWM1HRH：PWM1HRL 寄存器，调整 PWM1 输出的占空比。

当初始输出低电平时，PWM1LRH：PWM1LRL 配置的是输出低电平的时间，PWM1HRH：PWM1HRL 配置的是输出高电平的时间。

当初始输出高电平时，PWM1LRH：PWM1LRL 配置的是输出高电平的时间，PWM1HRH：PWM1HRL 配置的是输出低电平的时间。

4. 使能 PWM1：

将 PWM1CR 寄存器的 PWM1EN 位置 1,PWM1 开始工作产生 PWM 信号输出。

12.2. PWM 寄存器的定义

以下寄存器用于控制 PWM 的操作。

寄存器 12-1： PWM1CR：PWM1 控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	PWM1EN	OUTINIT	—	CLKDIV2	CLKDIV1	CLKDIV0
U-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未定义：读为 0

bit 5 PWM1EN：PWM1 使能位
0 = 禁止 PWM1
1 = 使能 PWM1，产生 PWM 信号输出

bit 4 OUTINIT：PWM1 输出初始状态
0 = 初始输出低电平
1 = 初始输出高电平

bit 3 未定义：读为 0

bit 2-0 CLKDIV<2:0>：PWM1 计数时钟分频选择位
000 = 2 分频
001 = 4 分频
010 = 8 分频
011 = 16 分频
100 = 32 分频
101 = 64 分频
110 = 128 分频
111 = 256 分频

寄存器 12-2： PWM1LRL：PWM1 脉宽调制寄存器 0 低 8 位

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 PWM1LRL<7:0>：PWM1 脉宽调制寄存器 0 低 8 位

PWM1 脉宽调制寄存器 0 低 8 位，与 PWM1LRH 组成 16 位脉宽调制寄存器 0

寄存器 12-2： PWM1LRH：PWM1 脉宽调制寄存器 0 高 8 位(续)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 PWM1LRH<7:0>：PWM1 脉宽调制寄存器 0 高 8 位

PWM1 脉宽调制寄存器 0 高 8 位，与 PWM1LRL 组成 16 位脉宽调制寄存器 0

寄存器 12-3： PWM1HRL：PWM1 脉宽调制寄存器 1 低 8 位

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 PWM1HRL<7:0>：PWM1 脉宽调制寄存器 1 低 8 位

PWM1 脉宽调制寄存器 1 低 8 位，与 PWM1HRH 组成 16 位脉宽调制寄存器 1

寄存器 12-3： PWM1HRH：PWM1 脉宽调制寄存器 1 高 8 位(续)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 PWM1HRH<7:0>：PWM1 脉宽调制寄存器 1 高 8 位

PWM1 脉宽调制寄存器 1 高 8 位，与 PWM1HRL 组成 16 位脉宽调制寄存器 1

寄存器 12-4： PWM2CR：PWM2 控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	PWM2EN	OUTINIT	CLKDIV3	CLKDIV2	CLKDIV1	CLKDIV0
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未定义：读为 0

bit 5 PWM2EN：PWM2 使能位

0 = 禁止 PWM2

1 = 使能 PWM2，产生 PWM 信号输出

bit 4 OUTINIT：PWM2 输出初始状态

0 = 初始输出低电平

1 = 初始输出高电平

bit 3-0 CLKDIV<3:0>：PWM2 计数时钟分频选择位

0000 = 2 分频

0001 = 4 分频

0010 = 8 分频

0011 = 16 分频

0100 = 32 分频

0101 = 64 分频

0110 = 128 分频

0111 = 256 分频

1000 = 512 分频

1001 = 1024 分频

1010 = 2048 分频

1011 = 4096 分频

1100 = 8192 分频

1101 = 16384 分频

1110 = 32768 分频

1111 = 65536 分频

寄存器 12-5： PWM2LR：PWM2 脉宽调制寄存器 0

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 PWM2LR<7:0>：PWM2 脉宽调制寄存器 0

PWM2 脉宽调制寄存器 0

寄存器 12-6： PWM2HR：PWM2 脉宽调制寄存器 1

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 PWM2HR<7:0>：PWM2 脉宽调制寄存器 1

PWM2 脉宽调制寄存器 1

13. 实时时钟模块(RTC)

RTC 模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
RTCCR	0xE8FF	XRAM	RTC 控制寄存器	0000 0000
RTCSR	0xE9FF	XRAM	RTC 状态寄存器	1000 0000
RTCHDR	0xEAFF	XRAM	RTC 时钟时寄存器	0000 0000
RTCMDR	0xEBFF	XRAM	RTC 时钟分寄存器	0000 0000
RTCSDR	0xECFF	XRAM	RTC 时钟秒寄存器	0000 0000
RTCAHDR	0xEDFF	XRAM	RTC 闹钟时寄存器	0001 0111
RTCAMDR	0xEEFF	XRAM	RTC 闹钟分寄存器	0011 1011
RTCASDR	0xEFFF	XRAM	RTC 闹钟秒寄存器	0011 1011

RTC 是实时时钟模块，支持以下功能：

- 时钟功能
- 定时(闹钟)功能(以秒为最小单位)

13.1. RTC 工作原理

RTC 是以外部 32.768K 晶振作为时钟源进行工作，所以必须外接有 32.768K 晶振，RTC 模块才能工作。在外部晶体已产生时钟的情况下，使能 RTC(RTCCR 寄存器的 RTCEN 位置 1)，RTC 就开始工作。

RTCHDR、RTCMDR、RTCSDR 寄存器组成一组对应 24 小时制时间的时、分、秒寄存器，用于 RTC 的时钟功能。设定好当前时间(配置 RTCHDR、RTCMDR、RTCSDR)，启动 RTC(RTCCR 寄存器的 RTCEN 位置 1)，时钟功能就开始运行，之后软件读 RTCHDR、RTCMDR、RTCSDR 寄存器就得到当前的实时时间(连续两次读到的值一样才认为正确)。

RTCAHDR、RTCAMDR、RTCASDR 寄存器组成一组设置闹钟的时、分、秒寄存器，用于 RTC 的闹钟功能。设定好闹钟时间(配置 RTCAHDR、RTCAMDR、RTCASDR)，启动 RTC，当 RTCHDR、RTCMDR、RTCSDR 的值与 RTCAHDR、RTCAMDR、RTCASDR 的值匹配时，就会产生闹钟中断。

注：1、本模块不会影响 RTC 计时精度。RTC 的计时精度取决于所选的 32.768KHz 晶体的精度，以及晶体所处环境的温度等因素。一般在常温(25℃)下，32.768KHz 晶体精度比较高(< 5ppm)，受环境温度影响时，精度可能达到±30ppm。

2、1ppm 误差相当于 1 年偏差 31.5 秒

3、RTC 在正常、休眠、空闲模式下都能正常工作。

13.2. RTC 的操作步骤

配置 RTC 的操作步骤如下：

1. 必须外接 32.768K 晶振：

RTC 是以外部 32.768K 晶振作为时钟源进行工作，所以必须外接有 32.768K 晶振。

2. 配置当前时间：

配置 RTCHDR、RTCMDR、RTCSDR 寄存器可以设置当前时间，必须符合 24 小时制的格式(即 RTCHDR 寄存器的值不能大于 23，RTCMDR、RTCSDR 寄存器得值不能大于 59)，否则会出错。

3. 配置闹钟时间：

配置 RTCAHDR、RTCAMDR、RTCASDR 寄存器可以设置闹钟时间,必须符合 24 小时制的格式(即 RTCAHDR 寄存器的值不能大于 23，RTCAMDR、RTCASDR 寄存器得值不能大于 59)，否则会出错。

4. 清 RTC 中断标志位：

在使能中断前，先写 RTCSR 寄存器 = 0x00，清除所有 RTC 中断标志位。

5. 中断控制：

RTCCR 寄存器的 INTSEN、INTMEN、INTHEN、INTDEN、INTCEN 位是 RTC 中断使能位，RTCSR 寄存器的 INTSSR、INTMSR、INTHSR、INTDSR、INTCSR 位是对应的 RTC 中断标志位。

6. RTC 使能：

将 RTCCR 寄存器的 RTCEN 位置 1，RTC 开始工作。

注：RTCSR 寄存器的 SETEN 位默认值为 1，只有当 SETEN = 1 时,才可以配置 RTCHDR、RTCMDR、RTCSDR、RTCAHDR、RTCAMDR、RTCASDR 寄存器中的一个，当配置其中一个寄存器时,硬件会先将 SETEN 位清 0，等待内部逻辑同步完成，再恢复为 1。软件在配置其中某个寄存器前，应查询到 SETEN 位等于 1 后，才可以配置 RTCHDR、RTCMDR、RTCSDR、RTCAHDR、RTCAMDR、RTCASD 这几个寄存器，否则可能会出错。

13.3. RTC 寄存器的定义

以下寄存器用于控制 RTC 的操作。

寄存器 13-1： RTCCR：RTC 控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RTCEN	—	—	INTCEN	INTDEN	INTHEN	INTMEN	INTSEN
R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 RTCEN：RTC 使能位
0 = RTC 停止工作
1 = RTC 开始工作

bit 6-5 未定义：读为 0

bit 4 INTCEN：RTC 闹钟中断使能位
0 = 禁止 RTC 闹钟中断
1 = 使能 RTC 闹钟中断

bit 3 INTDEN：RTC 天中断使能位
0 = 禁止 RTC 时钟跳到新的一天中断
1 = 使能 RTC 时钟跳到新的一天中断

bit 2 INTHEN：RTC 时中断使能位
0 = 禁止 RTC 时钟跳到新的小时中断
1 = 使能 RTC 时钟跳到新的小时中断

bit 1 INTMEN：RTC 分中断使能位
0 = 禁止 RTC 时钟跳到新的分钟中断
1 = 使能 RTC 时钟跳到新的分钟中断

bit 0 INTSEN：RTC 秒中断使能位
0 = 禁止 RTC 时钟跳到新的秒钟中断
1 = 使能 RTC 时钟跳到新的秒钟中断

寄存器 13-2： RTCSR：RTC 状态寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SETEN	—	—	INTCSR	INTDSR	INTHSR	INTMSR	INTSSR
R-1	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 SETEN：RTC 时钟、闹钟寄存器设置状态位
0 = 不允许配置时钟、闹钟寄存器
1 = 允许配置时钟、闹钟寄存器

bit 6-5 未定义：读为 0

bit 4 INTCSR：RTC 闹钟中断标志位
0 = 产生了 RTC 闹钟中断条件
1 = 产生了 RTC 闹钟中断条件

bit 3 INTDSR：RTC 天中断标志位
0 = RTC 时钟没有跳到新的一天
1 = RTC 时钟跳到了新的一天

bit 2 INTHSR：RTC 时中断标志位
0 = RTC 时钟没有跳到新的一小时
1 = RTC 时钟跳到了新的一小时

bit 1 INTMSR：RTC 分中断标志位
0 = RTC 时钟没有跳到新的一分钟
1 = RTC 时钟跳到了新的一分钟

bit 0 INTSSR：RTC 秒中断标志位
0 = RTC 时钟没有跳到新的一秒
1 = RTC 时钟跳到了新的一秒

注： RTCSR 寄存器的 SETEN 位默认值为 1，只有当 SETEN = 1 时，才可以配置 RTCHDR、RTCMDR、RTCSDR、RTCAHDR、RTCAMDR 、RTCASDR 寄存器中的一个，当配置其中一个寄存器时，硬件会先将 SETEN 位清 0，等待内部逻辑同步完成，再恢复为 1。软件在配置其中某个寄存器前，应查询到 SETEN 位等于 1 后，才可以配置 RTCHDR、RTCMDR、RTCSDR、RTCAHDR、RTCAMDR、RTCASD 这几个寄存器，否则可能会出错。

寄存器 13-3： RTCHDR：RTC 时钟时寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—					
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-5 未实现：读为 0

bit 4-0 RTCHDR<4:0>：RTC 时钟时寄存器

RTC 小时设置寄存器，不管 RTC 是否已启动，都可以配置此寄存器修改当前时间。

寄存器 13-4： RTCMDR：RTC 时钟分寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—						
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未实现：读为 0

bit 5-0 RTCMDR<5:0>：RTC 时钟分寄存器

RTC 分钟设置寄存器，不管 RTC 是否已启动，都可以配置此寄存器修改当前时间。

寄存器 13-5： RTCSDR：RTC 时钟秒寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—						
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未实现：读为 0

bit 5-0 RTCSDR<5:0>：RTC 时钟秒寄存器

RTC 秒钟设置寄存器，不管 RTC 是否已启动，都可以配置此寄存器修改当前时间。

寄存器 13-6： RTCAHDR：RTC 闹钟时寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—					
U-0	U-0	U-0	R/W-1	R/W-0	R/W-1	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-5 未实现：读为 0

bit 4-0 RTCAHDR<4:0>：RTC 闹钟时寄存器

RTC 闹钟小时设置寄存器，不管 RTC 是否已启动，都可以配置此寄存器修改闹钟时间。

寄存器 13-7： RTCAMDR：RTC 闹钟分寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—						
U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-0	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未实现：读为 0

bit 5-0 RTCAMDR<5:0>：RTC 闹钟分寄存器

RTC 闹钟分钟设置寄存器，不管 RTC 是否已启动，都可以配置此寄存器修改闹钟时间。

寄存器 13-8： RTCASDR：RTC 闹钟秒寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—						
U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-0	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未实现：读为 0

bit 5-0 RTCASDR<5:0>：RTC 闹钟秒寄存器

RTC 闹钟秒钟设置寄存器，不管 RTC 是否已启动，都可以配置此寄存器修改闹钟时间。

13.4. RTC 中断

RTC 有 5 个中断，分别是：

- RTC 秒中断(RTCSDR 每增加 1 时)
- RTC 分中断(时间从 xx：xx：59 跳到 xx：xx：00 时)
- RTC 时中断(时间从 xx：59：59 跳到 xx：00：00 时)
- RTC 天中断(时间从 23：59：59 跳到 00：00：00 时)
- RTC 闹钟中断(时钟寄存器与闹钟寄存器匹配时)

只要有 RTC 中断条件产生，不论是否使能 RTC 中断，RTCSR 寄存器的 INTSSR、INTMSR、INTHSR、INTDSR、INTCSR 中断标志位都会置 1，INTSSR、INTMSR、INTHSR、INTDSR、INTCSR 位必须在软件中清零。RTC 的中断使能位是 RTCCR 寄存器的 INTSEN、INTMEN、INTHEN、INTDEN、INTCEN 位。

注： 在休眠模式下，RTC 闹钟中断可以唤醒处理器。
在空闲模式下，RTC 中断可以唤醒处理器。

14. 模数转换器模块(ADC)

ADC 模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
ADCDIVCR	0xB8FF	XRAM	ADC 采样时钟分频寄存器	0010 0000
ADCSELR	0xB9FF	XRAM	ADC 通道选择寄存器	0000 0000
ADCSR	0xBAFF	XRAM	ADC 中断状态寄存器	0000 0000
ADCCR	0xBBFF	XRAM	ADC 控制寄存器	0000 0010
ADCDRL	0xBCFF	XRAM	ADC 结果寄存器0	0000 0000
ADCDRH	0xBDFF	XRAM	ADC 结果寄存器1	0000 0000

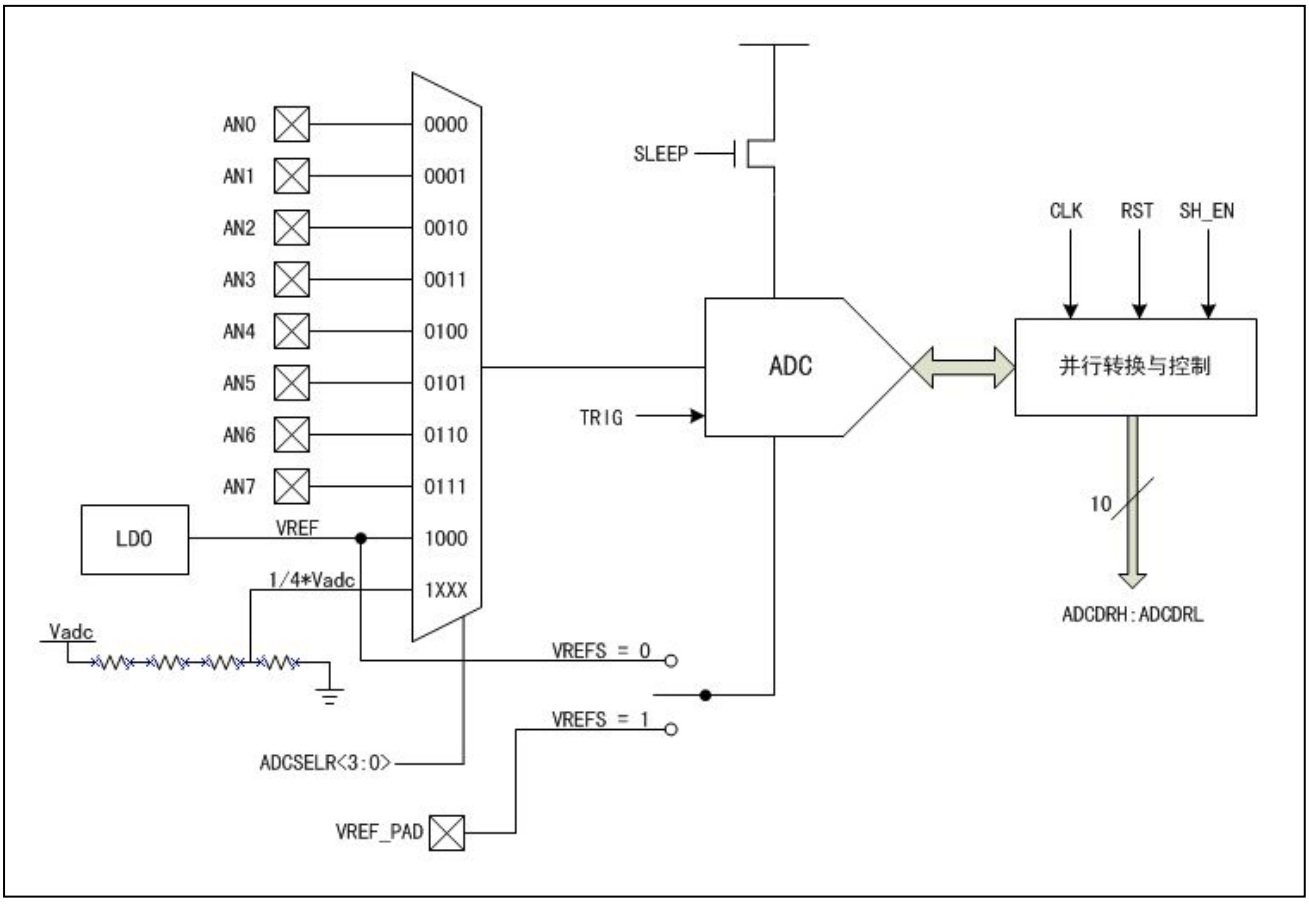
模数转换器(ADC)可以将模拟输入信号转换为表示该信号的一个 10 位二进制数。芯片使用的模拟输入通道共用一个采样保持电路。采样保持电路的输出与模数转换器的输入相连。模数转换器采用逐次逼近法产生一个 10 位二进制结果，选择不同的模式，会对这个结果采用不同的处理方法，并将处理后得到的值保存在 ADC 结果寄存器(ADCDRL 和 ADCDRH)中。

可软件选择转换所使用的参考电压为内部产生的固定参考电压(1V)或者由外部提供。

ADC 在转换完成之后可以产生一个中断。

图 14-1 所示为 ADC 的框图。

图 14-1： ADC 的框图



14.1. ADC 工作原理

14.1.1. 启动转换

要启动 ADC 转换，必须配置成正确的工作方式(只能配置成 14.2 节中“ADC 工作方式选择”所说的三种工作方式，如果配置成外部参考电压硬件方式，采样结果将是错误的)，然后将 ADCCR 寄存器的 TRIG 位置 1，开始模数转换。(TRIG 位为只写位，读为 0)

注： 不能在配置 ADCCR 寄存器的同时将 TRIG 位置 1 启动 ADC 转换。

14.1.2. 完成转换

启动 ADC 转换后，当转换完成时，ADC 模块将：

1) ADCSR 寄存器的 SAMPEND 位置 1

2) 转换得到的结果更新到 ADCDRH：ADCDRL 寄存器中

14.2. ADC 的操作步骤

配置 ADC 的操作步骤如下：

1. 端口配置：

将 ADC 输入引脚 I/O 功能复用配置成 ADC 通道(PxyIOCFG)时，会自动将引脚方向配置为输入。不需要配置对应的 PnOE 寄存器，选择 ADC 通道引脚方向为输入。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

CBM73xx 系列芯片最多支持 8 路 ADC 通道，只有 P0.6、P0.7、P1.0、P1.1、P1.2、P1.3、P1.4、P1.5 这 8 个引脚可以配置成 ADC 通道(对应 ADCSELR 寄存器的 AN0 - AN7 选项)。

2. ADC 模块参数配置：

- ADC 通道选择：

由 ADCSELR 寄存器和 ADCCR 寄存器的 TRIM 位决定将哪个通道连接到采样保持电路。

当选择 AN0~AN7、VREF 通道时，ADCCR 寄存器的 TRIM 位必需清 0；当选择时 1/4*ADC 工作电压通道时，ADCCR 寄存器的 TRIM 位必需置 1。

- ADC 参考电压选择：

ADCCR 寄存器的 VREFS 位控制参考电压选择内部固定参考电压(1V)或者外部参考电压。

需要注意的是选择不同的参考电压，ADC 的工作方式有所区别，具体情形请看 ADC 工作方式选择。

- ADC 工作方式选择：

配置 ADCCR 寄存器的 MODE 位选择 ADC 的工作方式，注意选择硬件方式时，只能选择内部参考电压。

ADC 有三种工作方式：内部参考电压常规方式、内部参考电压硬件方式、外部参考电压常规方式。

1）内部参考电压常规方式工作原理：

第 1 步：首先配置 ADCCR 寄存器的 VREFS 位选择内部固定参考电压、配置 MODE 位选择常规方式；

第 2 步：配置 ADCSELR 寄存器，选择 VREF 作为采样通道，采样内部 1V 参考电压，保存 ADCDRH：ADCDRL 寄存器中的采样值 D1；

第 3 步：配置 ADCSELR 寄存器，选择外部输入电压作为采样通道，采样外部输入电压，保存 ADCDRH：ADCDRL 寄存器中的采样值 D2；

第 4 步：采样值与电压值是等比的，所以可以得出 VIN = D2 / D1。

2）内部参考电压硬件方式工作原理：

第 1 步：首先配置 ADCCR 寄存器的 VREFS 位选择内部固定参考电压、配置 MODE 位选择硬件方式；

第 2 步：配置 ADCSELR 寄存器，选择外部输入电压作为采样通道，采样外部输入电压，保存 ADCDRH、ADCDRL 寄存器中的采样值；

第 3 步：对于采样值，ADCDRH 寄存器表示被采样电压值的整数部，ADCDRL 寄存器表示被采样电压值的小数部分，电压值小数部分 = ADCDATL / 256；

例如，采样完成后得到的采样值 ADCDRH = 0x2，ADCDRL = 0x63；则被采样电压 VIN = ADCDRH + (ADCDRL / 256)= 2.387V。

3）外部参考电压常规方式工作原理：

第 1 步：首先配置 ADCCR 寄存器的 VREFS 位选择外部参考电压、配置 MODE 位选择常规方式；

第 2 步：配置 ADCSELR 寄存器，选择外部输入电压作为采样通道，采样外部输入电压，保存 ADCDRH、ADCDRL 寄存器中的采样值 DAT；

第 3 步：采样值换算成电压的公式为 VIN = (DAT / 1024) × Vref，其中 Vref 为外部参考电压值。

注： 当选择内部参考电压时，采样值的误差大小取决于内部参考电压的准确性，因此芯片在工作中应不断对 LDO 的运放 offset 进行 trim。

- 采样时钟选择：

ADC 的时钟源是 OSC48M，通过 ADCDIVCR 寄存器可以进行分频，完成一位转换的时间定义为 TAD=采样时钟周期。一个完整的 10 位转化需要 11 个 TAD 周期，如图 14-2 所示。

- 采集时间配置：

ADCCR 寄存器的 SAMPCNT<1:0>位控制采集时间。是 ADC 模块采集外部输入电压信号的时间，输入电压信号采集完成后，才会启动转换。如图 14-2 所示。

3. 清 ADC 中断标志位：

使能 ADC 中断前，先将 ADCSR 寄存器的 SAMPEND 位清 0，清除 ADC 采样完成标志位。

4. 中断控制：

ADC 模块允许在完成模数转换后产生一个中断。ADC 中断标志位是 ADCSR 寄存器的 SAMPEND 位，SAMPEND 位必须软件清 0。ADC 中断使能位是 ADCCR 寄存器的 INTEN 位。

5. 启动转换：

将 ADCCR 寄存器的 TRIG 位置 1 启动转换，注意必需先配置 ADCCR 寄存器的其他位，再将 ADCCR 寄存器的 TRIG 位置 1 启动转换，两者不能同时进行。

6. 等待 ADC 转换结束：

等待 ADC 转换结束有两种方法：

1) 查询 ADCSR 寄存器的 SAMPEND 位，等于 1 即转换结束

2) 等待 ADC 中断(使能中断)

7. 读取 ADC 转换结果：

ADC 转换结束后，可以通过读取 ADCDRH：ADCDRL 寄存器得到 ADC 转换结果。

8. 清 ADC 中断标志位：

清 ADC 中断标志位(ADCSR 寄存器的 SAMPEND 位)，以便再次进行 ADC 转换。

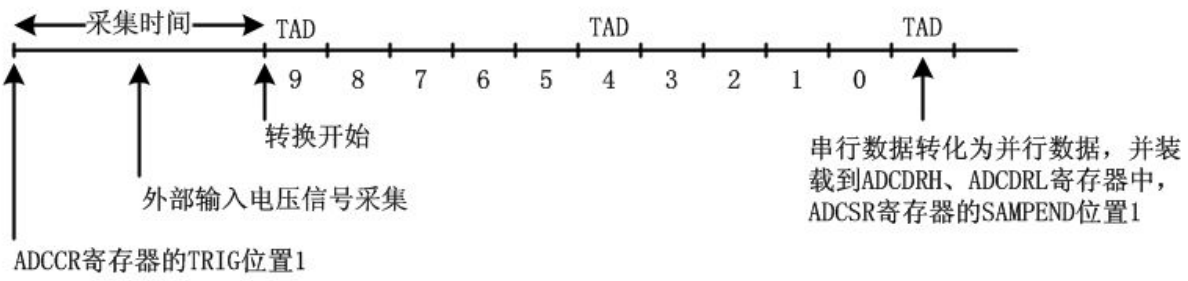
具体 ADC 转换范例如下：

例 14-1： ADC 转换

```
P06IOCFG = 0x09;    //配置 P0.6(AN0)为 ADC 通道
ADCSR = 0x00;       //清 ADC 中断标志位
ADCDIVCR = 0x20;    //配置采样时钟
ADCCR = 0x42;       //配置内部参考电压硬件方式
ADCSELR = 0x00;     //选择 P0.6(AN0)作为 ADC 输入
ADCCR_TRIG = 0x80; //启动 ADC 转换
while(0x00 == ADCSR_SAMPEND){;} //等待转换完成
ADCSR_SAMPEND = 0x00;   //清中断标志位
datl = ADCDRL;          //保存采样结果
dath = ADCDRH;
```

注： 每次转换结束后 SAMPEND 位都会被置 1，与 ADC 中断是否使能无关。所以必须等待 SAMPEND 位置 1 后再去读取采样值。

图 14-2： 模数转换器 TAD 周期



14.3. ADC 寄存器的定义

以下寄存器用于控制 ADC 的操作。

寄存器 14-1： ADCDIVCR：ADC 采样时钟分频寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 ADCDIVCR<7:0>：ADC 采样时钟分频寄存器
ADC 采样时钟周期 = 2 × ADCDIVCR × Tosc48M
(ADCDIVCR 的值必须大于 6)

寄存器 14-2： ADCSELR：ADC 通道选择寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—				
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未定义：读为 0
bit 3-0 ADCSELR<3:0>：ADC 通道选择寄存器
0000 = AN0
0001 = AN1
0010 = AN2
0011 = AN3
0100 = AN4
0101 = AN5
0110 = AN6
0111 = AN7
1000 = VREF
1001 - 1111 = 1/4*ADC 工作电压

寄存器 14-3： ADCSR：ADC 中断状态寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	—	—	SAMPEND
U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-1　　未实现：读为 0

bit 0　　SAMPEND：ADC 采样完成标志位

0 = ADC 采样未完成

1 = 完成一次 ADC 采样

寄存器 14-4： ADCCR：ADC 控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TRIG	MODE	VREFS	SLEEP	INTEN	TRIM	SAMPCNT1	SAMPCNT0
W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 TRIG：ADC 转换启动位
只写位，写 1 即启动 ADC 转换，读始终为 0，应注意必须在 1 次采样完成后才能再次启动。

bit 6 MODE：ADC 工作方式选择位
0 = 常规方式
1 = 硬件方式(只有选择内部参考电压，才可以配置成硬件方式)

bit 5 VREFS：ADC 参考电压选择位
0 = 选择内部固定参考电压
1 = 选择使用外部参考电压

bit 4 SLEEP：ADC 省电模式控制位
0 = ADC 在正常模式
1 = ADC 进入 SLEEP 模式

bit 3 INTEN：ADC 中断使能位
0 = 禁止 ADC 中断
1 = 使能 ADC 中断

bit 2 TRIM：ADC 通道选择辅助位
0 = ADC 可以选择 0-8 通道
1 = ADC 可以选择 9-15 通道

bit 1-0 SAMPCNT<1:0>：ADC 采集时间控制位(T 为采样时钟周期)
00 = 禁止设置
01 = 2T
10 = 4T
11 = 6T

寄存器 14-5： ADCDRL：ADC 结果寄存器 0

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 ADCDRL<7:0>：ADC 结果寄存器 0
与 ADCDRH 组成 ADC 采样完成后结果寄存器

寄存器 14-5： ADCDRH：ADC 结果寄存器 1(续)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—			
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-3 未实现：读为 0
bit 2-0 ADCDRL<2:0>：ADC 结果寄存器 1
与 ADCDRL 组成 ADC 采样完成后结果寄存器

注： 1、在内部参考电压常规方式下，ADCDRH：ADCDRL 寄存器保存的是 10bit 转换结果。
2、在外部参考电压常规方式下，ADCDRH：ADCDRL 寄存器保存的是 10bit 转换结果。
3、在内部参考电压硬件方式下，ADCDRH：ADCDRL 寄存器保存的是 11bit 转换结果。

15. 触摸按键功能(sensorADC)

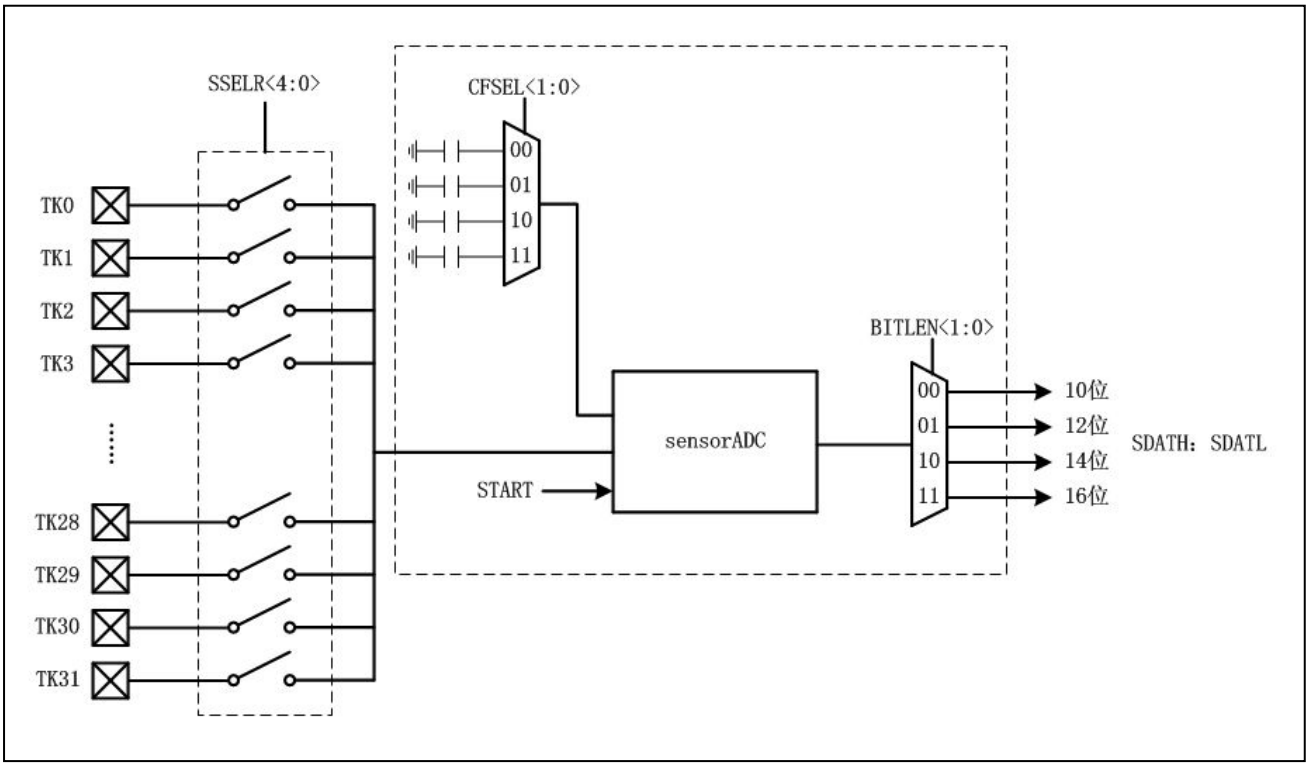
sensorADC 模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
SCR0	90FFH	XRAM	sensorADC 控制寄存器0	0000 0000
SCR1	91FFH	XRAM	sensorADC 控制寄存器1	0011 0100
SCR2	92FFH	XRAM	sensorADC 控制寄存器2	0100 0010
SCR3	93FFH	XRAM	sensorADC 控制寄存器3	1000 0010
SSCAPL	94FFH	XRAM	sensorADC 抵消电容配置寄存器低8位	0000 0000
SSCAPH	95FFH	XRAM	sensorADC 抵消电容配置寄存器高1位	0000 0000
SSR	96FFH	XRAM	sensorADC 中断状态寄存器	0000 0000
SDATL	97FFH	XRAM	sensorADC 结果寄存器低8位	0000 0000
SDATH	98FFH	XRAM	sensorADC 结果寄存器高8位	0000 0000
SSELR	9AFFH	XRAM	sensorADC 通道选择寄存器	0000 0000
SWKCR	9EFFH	XRAM	SensorADC SLEEP 唤醒配置寄存器	0000 0011

CBM73xx 系列芯片内建触摸按键功能模块，最大能连接 32 个按键。内部集成了 4 阶可选灵敏度电容，不需要外接电容，可以根据实际电路板材质以及触摸按键介质调节合适的灵敏度。电容值越小，灵敏度越高；电容值越大，灵敏度越低。

图 15-1 是 sensorADC 的系统框图。

图 15-1： sensorADC 系统框图



15.1. sensorADC 工作原理

15.1.1. 启动转换

必须先配置好参数，选择软件控制方式，使能 sensorADC 模块，然后将 SCR0 寄存器的 START 位置 1，启动 sensorADC 转换。(START 位为只写位，读为 0)

注： 不能在配置 SCR0 寄存器的同时将 START 位置 1 启动 sensorADC 采样。

15.1.2. 完成转换

启动 sensorADC 采样后，当采样完成时，sensorADC 模块将：

1) SSR 寄存器的 SAMPEND 位置 1
2) 转换得到的结果更新到 SDATH：SDATL 寄存器中

15.2. sensorADC 操作步骤

配置休眠模式下特定触摸按键唤醒的操作步骤在 5.3.1.1 小节“休眠唤醒”做了说明，此处介绍正常工作模式下配置 sensorADC 的操作步骤，如下：

1. 端口配置：

将 sensorADC 输入引脚的 I/O 功能复用配置成 sensorADC 通道(PxyIOCFG)时，会自动将引脚方向配置为输入。不需要配置对应的 PnOE 寄存器，选择 sensorADC 通道引脚方向为输入。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

CBM73xx 系列芯片最多支持 32 路 sensorADC 通道，除了 P0.6、P0.7、P1.0、P1.1、P1.2、P1.3、P1.4、P1.5 这 8 个引脚外，都可以配置成 sensorADC 通道。

2. sensorADC 模块参数配置：

- 配置 sensorADC 控制方式：

将 SCR0 寄存器的 CTRLMODE<1:0>位配置为 00，选择软件方式控制 sensorADC 的使能及通道切换。

sensorADC 有三种控制方式：

1) 软件方式：

软件方式控制 sensorADC 使能以及通道的切换；

2) 休眠模式硬件方式：

在休眠期间，由硬件控制对选定的通道进行反复采样，直到满足唤醒条件将芯片从休眠模式下唤醒。

3) 正常工作硬件方式：

应用于触摸库，此处不做介绍。

- 选择硬件滤波器的工作方式：

将 SCR0 寄存器的 FLTMODE 位清 0，选择仅使用当前 sensorADC 的采样值。

另一种硬件滤波器的工作方式是与触摸库相关。

- 选择 sensorADC 工作方式：

SCR1 寄存器的 WORKMODE 位用于配置 sensorADC 工作方式，置 1 为跟随模式，跟随模式可提高 sensorADC 的抗干扰能力。

- 配置灵敏度电容：

sensorADC 模块内部集成了灵敏度电容，可通过 SCR1 寄存器的 CFSEL<1:0>位选择灵敏度电容的值，有 4 种可供选择：

00：CI

01：2/3*CI

10：1/2*CI

11：1/3*CI

- 选择工作电流：

配置 SCR1 寄存器的 CURSEL<3:0>位选择 sensorADC 的电流。

- 配置转化稳定时间：

启动 sensorADC 转换后，硬件上会先将模拟信号转化为数字波形，这段转化时间可以通过 SCR2 寄存器的 STABLEN<5:0>位进行配置。

计算方法如下：

以转化时钟为单位，

实际的拍数 = STABLEN<5:0> × 4。

转化时钟就是 OSC48M 经过 SCR3 寄存器的 TRDIV<3:0> 位分频后的时钟。

- 选择采样精度：

配置 SCR2 寄存器的 BITLEN<1:0>位选择 sensorADC 的采样精度：

00 = 10 位 (采样时间拍数：31)

01 = 12 位 (采样时间拍数：63)

10 = 14 位 (采样时间拍数：127)

11 = 16 位 (采样时间拍数：256)

采样精度越高，采样时间也越长。括号内是采样时间的拍数，以 OSC48M 经过 SCR3 寄存器的 TRDIV<3:0> 位分频后时钟的一个 clock 作为一拍。

- 配置当前通道抵消电容：

由于实际电路板材质以及触摸按键介质存在差异，从而导致外部寄生电容的差异，所以需要根据实际情况配置不同的内部抵消电容。通过 SSCAPH：SSCAPL 寄存器可以配置 9bit 当前通道抵消电容。若采样通道变化，则当前通道抵消电容也应相应的改变。

- 配置采样时钟：

sensorADC 的时钟源是 OSC48M，采样时钟是 OSC48M 经过 SCR3 寄存器的 SHDIV<3:0>位分频后的时钟。

- 配置转化时钟：

sensorADC 的时钟源是 OSC48M，转化时钟是 OSC48M 经过 SCR3 寄存器的 TRDIV<3:0>位分频后的时钟。

- 选择 sensorADC 采样通道：

由 SSELR 寄存器决定将哪个通道连接到 sensorADC 采样电路。

3. 禁止休眠唤醒相关配置

将 SWKCR 寄存器的 CLKSEL 位清 0，选择内部高速 OSC48M 经过 SCR3 寄存器分频后的时钟作为 sensorADC 时钟。

4. 使能 sensorADC：

将 SCR1 寄存器的 ANALOGEN 位置 1，使能 sensorADC 模块，使能后，sensorADC 才能正常工作。

5. 清软件方式采样完成中断标志位：

使能 sensorADC 中断前，先将 SSR 寄存器清 0，清 sensorADC 相关中断标志位。

6. 中断控制：

sensorADC 模块允许在软件方式采样完成后产生一个中断。sensorADC 采样完成中断使能位是 SCR0 寄存器的 SAMPEN 位，sensorADC 中断标志位是 SSR 寄存器的 SAMPEND 位，SAMPEND 位必须软件清 0。

同时还有应用于触摸库的两个中断，中断使能位是 SCR0 寄存器的 FLTEN、CAPEN 位，中断标志位是 SSR 寄存器的 FLTEND、CAPEND 位。这里不做详细介绍，用户使用时直接禁止这两个中断。

SSR 寄存器中 BLEND 位基线更新完成标志位，不产生中断，应用于触摸库。

7. 启动转换：

将 SCR0 寄存器的 START 位置 1 启动转换。注意必需先配置 SCR0 寄存器的其他位，再将 SCR0 寄存器的 START 位置 1 启动转换，两者不能同时进行。

8. 等待 sensorADC 转换结束：

等待 sensorADC 转换结束有两种方法：

1) 查询 SSR 寄存器的 SAMPEND 位，等于 1 即转换结束
2) 等待 sensorADC 软件方式采样完成中断(使能中断)

9. 读取 sensorADC 采样结果：

sensorADC 转换结束后，可以通过读取 SDATH：SDATL 寄存器得到 sensorADC 采样结果。

10. 清软件方式采样完成中断标志位：

将 SSR 寄存器的 SAMPEND 位清 0，清 sensorADC 软件方式采样完成中断标志位。以便再次进行 sensorADC 采样。

例 15-1： sensorADC 转换

```
P00IOCFG = 0x08;    //配置 P0.0(TK0)为 sensor 通道
SSR = 0x00;         //清 sensorADC 相关中断标志位
SCR0= 0x00;         //配置软件方式控制、禁止中断
SCR1 = 0xC0;        //使能 sensorADC、配置部分参数
SCR2 = 0x43;        //配置转化稳定时间，选择精度
SCR3 = 0x82;        //配置采样时钟、转化时钟
SSCAPL = 0x00;      //配置抵消电容
SSCAPH = 0x01;
SWKCR = 0x03;       //选择 OSC48M 分频后的时钟
SSELR = 0x00; //选择 P0.0(TK0)作为 sensorADC 输入
SCR0_START = 0x40; //启动 ADC 转换
while(0x00 == SSR_SAMPEND){;} //等待转换完成
SSR_SAMPEND = 0x00;     //清中断标志位
datl = SDATL;       //保存采样结果
dath = SDATH;
```

15.3. sensorADC 寄存器的定义

以下寄存器用于控制 sensorADC 的操作。

寄存器 15-1： SCR0：sensorADC 控制寄存器 0

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	START	CTRLMODE1	CTRLMODE0	FLTMODE	BLEN	FLTEN	SAMPEN
U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 未实现：读为 0

bit 6 START：sensorADC 转换启动位

写 1 即启动 sensorADC 转换，完成后自动清 0，应注意必须在 1 次采样完成后才能再次启动。

bit 5-4 CTRLMODE<1:0>：sensorADC 控制方式选择位

00 = 软件方式，软件控制 sensorADC 使能和通道切换。

01 = 硬件方式，用作休眠模式下唤醒。在进入休眠前，软件应选择一个通道用作唤醒通道，在休眠期间，由硬件控制对该通道进行反复采样，直到满足唤醒条件将芯片从休眠模式下唤醒。

1X = 硬件方式，由硬件算法控制 sensorADC 使能、通道切换、抵消电容。

bit 3 FLTMODE：硬件滤波器工作方式选择位(触摸库相关)

0 = 仅使用当前 sensorADC 的采样值

1 = 使用当前 sensorADC 的采样值和 XRAM 中存放的值

bit 2 CAPEN：抵消电容自适应完成中断使能位(触摸库相关)

0 = 禁止抵消电容自适应完成中断

1 = 使能抵消电容自适应完成中断

bit 1 FLTEN：硬件滤波采样完成中断使能位(触摸库相关)

0 = 禁止硬件滤波采样完成中断

1 = 使能硬件滤波采样完成中断

bit 0 SAMPEN：软件方式，采样完成中断使能位

0 = 禁止采样完成中断

1 = 使能采样完成中断

寄存器 15-2： SCR1：sensorADC 控制寄存器 1

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
WORKMODE	ANALOGEN	CFSEL1	CFSEL0	CURSEL3	CURSEL2	CURSEL1	CURSEL0
R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-1	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 WORKMODE：sensorADC 工作方式选择位
0 = 固定模式
1 = 跟随模式

bit 6 ANALOGEN：sensorADC 模块使能位
0 = 关闭 sensorADC 模块
1 = 使能 sensorADC 模块

bit 5-4 CFSEL<1:0>：内部灵敏度电容选择(电容越小，灵敏度越高，可测试电容范围变小)
00 = CI
01 = 2/3*CI
10 = 1/2*CI
11 = 1/3*CI

bit 3-0 CURSEL<3:0>：sensorADC 电流控制位
0000 = 禁止设置
0001~1111 共 15 档配置

寄存器 15-3： SCR2：sensorADC 控制寄存器 2

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
STABLEN5	STABLEN4	STABLEN3	STABLEN2	STABLEN1	STABLEN0	BITLEN1	BITLEN0
R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-2 STABLEN<5:0>：转化稳定时间选择位
用于设定 ADC 开始转换稳定时间(振荡时钟经 TRDIV 分频后的时钟为一个拍数)
实际的拍数 = STABLEN × 4。

bit 1-0 BITLEN<1:0>：sensorADC 精度选择位(振荡时钟经 TRDIV 分频后的时钟为一个拍数)
00 = 10 位 (采样时间拍数：31)
01 = 12 位 (采样时间拍数：63)
10 = 14 位 (采样时间拍数：127)
11 = 16 位 (采样时间拍数：256)

寄存器 15-4： SCR3：sensorADC 控制寄存器 3

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SHDIV3	SHDIV2	SHDIV1	SHDIV0	TRDIV3	TRDIV2	TRDIV1	TRDIV0
R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 SHDIV<3:0>：采样时钟分频选择位
当 SHDIV = 0，ADC 的采样时钟频率 = OSC48M；
当 SHDIV = 1~15，ADC 的采样时钟频率 = OSC48M / (2 × SHDIV)；

bit 3-0 TRDIV<3:0>：转化时钟分频选择位
当 TRDIV = 0，ADC 的转化时钟频率 = OSC48M；
当 TRDIV = 1~15，ADC 的转化时钟频率 = OSC48M / (2 × SHDIV)；

寄存器 15-5： SSCAPL：sensorADC 抵消电容配置寄存器低 8 位

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 SSCAPL<7:0>：sensorADC 抵消电容配置寄存器低 8 位
和 SSCAPH 组成 9bit 的当前通道抵消电容配置寄存器

寄存器 15-5： SSCAPH：sensorADC 抵消电容配置寄存器高 1 位(续)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	—	—	
U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-1 未实现：读为 0
bit 0 SSCAPH：sensorADC 抵消电容配置寄存器高 1 位
和 SSCAPL 组成 9bit 的当前通道抵消电容配置寄存器

寄存器 15-6： SSR：sensorADC 中断状态寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	CAPEND	BLEND	FLTEND	SAMPEND
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未实现：读为 0

bit 3 CAPEND：抵消电容自适应完成中断标志位(触摸库相关)

0 = 抵消电容自适应进行中或未启动

1 = 完成抵消电容自适应

bit 2 BLEND：基线更新完成标志位(触摸库相关)

0 = 基线更新进行中或未启动

1 = 完成基线更新

bit 1 FLTEND：硬件滤波采样完成中断标志位(触摸库相关)

0 = 硬件滤波采样进行中或未启动

1 = 完成硬件滤波采样

bit 0 SAMPEND：软件方式，采样完成中断标志位

0 = 软件方式采样进行中或未启动

1 = 完成软件方式采样

寄存器 15-7： SDATL：sensorADC 结果寄存器低 8 位

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 SDATL<7:0>：sensorADC 抵消电容配置寄存器低 8 位

和 SDATH 组成 16bit 的采样结果寄存器

寄存器 15-7： SDATH：sensorADC 结果寄存器高 8 位(续)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 SDATH<7:0>：sensorADC 抵消电容配置寄存器高 8 位
和 SDATL 组成 16bit 的采样结果寄存器

寄存器 15-8： SSELR：sensorADC 通道选择寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—					
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-5 未实现：读为 0

bit 4-0 SSELR<4:0>：sensorADC 通道选择寄存器

00000 = TK0
00001 = TK1
00010 = TK2
00011 = TK3
00100 = TK4
00101 = TK5
00110 = TK6
00111 = TK7
……
11000 = TK24
11001 = TK25
11010 = TK26
11011 = TK27
11100 = TK28
11101 = TK29
11110 = TK30
11111 = TK31

注： TK0~TK31 与 P0.0~P4.7 中除了 P0.6、P0.7、P1.0、P1.1、P1.2、P1.3、P1.4、P1.5 以外的引脚一一对应。

寄存器 15-9： SWKCR：sensorADC 休眠唤醒配置寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	CLKSEL	CMPRSET2	CMPRSET1	CMPRSET0
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未实现：读为 0

bit 3 CLKSEL：sensorADC 时钟源选择位

0 = 时钟选择内部高速 OSC48M 经过 SCR3 寄存器分频后的时钟

1 = 时钟选择内部低速 OSC800K 经过 CKDIVCR 寄存器的 SLOWDIV 位分频后，再 4 分频的时钟。

bit 2-0 CMPRSET<2:0>：休眠唤醒参数配置位

在休眠模式下，要能被触摸按键唤醒，只有 sensorADC 采样值连续 N 次落入到预先设置的阈值范围内(请参考 SLPWKCR，SLPWKDR 寄存器说明)，若还不足 N 次时，出现一次不在阈值范围内的采样值，则重新计算采样值落入阈值的次数。CMPRSET 位用于设置这里的次数 N。

注： 若要使用触摸按键休眠唤醒功能，在进入休眠前，必需将 CLKSEL 位置 1；

16. 通用异步收发器(UART)

UART 模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
UCR	0xD0FF	XRAM	UART 控制寄存器	0000 0100
USETR	0xD1FF	XRAM	UART 数据格式控制寄存器	0000 1000
UBPRL	0xD2FF	XRAM	UART 波特率配置寄存器低8位	1111 1111
UBPRH	0xD3FF	XRAM	UART 波特率配置寄存器高4位	0000 1111
UIER	0xD4FF	XRAM	UART 中断使能寄存器	0000 0000
UTXR	0xD5FF	XRAM	UART 发送数据寄存器	0000 0000
URXR	0xD6FF	XRAM	UART 接收数据寄存器	0000 0000
USR	0xD7FF	XRAM	UART 状态寄存器	0001 0000

通用异步收发器(UART)模块是一个串行 I/O 通信外设，它可被配置为能与 CRT 终端和个人计算机等外设通信的全双工异步系统。

UART 模块包含如下功能：

- 全双工异步发送和接收
- 可将字符长度编程为 5/6/7/8 位
- 输入缓冲器溢出错误检测
- 停止位错误检测
- 奇偶校验位错误检测
- 自动波特率检测和校准

每个发送的字符包括一个起始位，后面跟有 5/6/7/8 个数据位，若有奇偶校验位就再加上 1 位校验位，以及 1 个或多个终止字符发送的停止位。最常用的数据为 8 位数据位不做奇偶校验。每个发送位的持续时间为 1/(波特率)。片上专用 12 位波特率发生器可用于通过振荡器产生标准波特率频率(见寄存器 UBPRH：UBPRL)。

UART 首先发送和接受 LSB。UART 的发送器和接受器在功能上是相互独立的，但采用相同的数据格式和波特率。

16.1. UART 的操作步骤

配置 UART 的操作步骤如下：

1. 端口配置

通过 PxyIOCFG 寄存器将 I/O 配置成 UART 输出/输入功能时，会自动将引脚方向配置为相应的方向。不需要配置对应的 PnOE 寄存器，选择 UART 输出/输入引脚方向为输出/输入。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. UART 模块时钟使能

在操作 UART 模块相关寄存器前，必须首先使能 UART 模块时钟，即将 MDLCKCR 寄存器的 UARTEN 位置 1。

3. UART 模块参数配置：

- 数据格式配置

USETR 寄存器可以配置通讯数据的格式，包括校验位、数据位、停止位。

- 波特率配置

波特率有两种配置方式(通过 UCR 寄存器的 BPRREC 位选择)：

1) 自动波特率校准

采用自校准方式。上位机必须先发送 0x11，芯片分析上位机的波特率，在以后的通讯过程中，都采用和上位机一致的波特率通讯。此种情况对芯片内部时钟精准度无特别要求，但上位机必须先发送特定数据。

2) 12 位波特率配置寄存器

通讯的波特率依据 12 位波特率配置寄存器 UBPRH：UBPRL 的设置确定。在时钟比较准确的情况下，可以采用这种方式。

UART 的时钟源是 OSC48M 3 分频后的时钟(16M)，UBPRL 为 UBPR 值的低 8 位，UBPRH 为 UBPR 的高 4 位。UBPR 是根据波特率推算的配置数：

例如：波特率 bpr 是 115200，计算 UBPR 如下：

UBPR = F / bpr - 1；(F 的单位是 Hz)

= 16 * 1000000 / 115200 - 1

= 138(0x8A)

所以 UBPRH = 0x00, UBPRL = 0x8A。

4. 中断控制

UIER 寄存器的 TXIE、RXIE 位分别是 UART 发送完 1 个 BYTE 数据中断使能位、接收完 1 个 BYTE 数据中断使能位。对应的中断标志位是 USR 寄存器的 TXDONE、RXREADY 位。

不论是否使能发送完成、接收完成中断，当接收到 1 个 BYTE 数据时,RXREADY 都会置 1;当发送完 1 个 BYTE 数据时，TXDONE 都会置 1。

USR 寄存器还包括 UART 的奇偶校验位错误标志位、停止位错误标志位、接收溢出错误标志位，这几种情况不会产生中断，当出现相应错误时标志位就会置 1。

5. 接收、发送使能

当端口、数据格式、波特率都配置好后，可以将 UCR 寄存器的 RXEN 位置 1 使能数据接收，TXEN 位置 1 使能数据发送。数据的发送和接收功能是相互独立的，不会受另一方的影响。

16.2. 发送数据

UTXR 为发送数据寄存器，向 UTXR 寄存器写入一个字符，即启动发送。只有当 USR 寄存器的 TXDONE 位 = 1 时，才能将待发送的数据写入到 UTXR 寄存器中，否则写入的数据会丢失。TXDONE 是只读位，不能由软件清 0。当 TXDONE = 1 时，向 UTXR 寄存器写入一个字符后，TXDONE 会由硬件自动清 0。

如果要在发送数据时使用中断，只在有待发送数据时，才将 TXIE 位置 1。当将待发送的最后一个字符写入 UTXR 后，将 TXIE 位清 0。

16.3. 接收数据

接收器在第一个位的下降沿开始接收字符。第一个位，通常称为起始位，始终为 0。如果起始位校验出错，则放弃接收该字符，只有在接收到正确的起始位，才会继续采样，最后一个位为停止位，总是为 1。如果采样停止位时，采样到 0，则会将停止位错误检测位置 1。

当完成所有数据的采样并将其全部移入接收缓冲寄存器时，USR 寄存器的 RXREADY 位会被置 1，通过读 URXR 寄存器将缓冲寄存器的数据读出后，RXREADY 位会自动清 0。如果在接收到 1 Byte 数据后，软件未及时读 URXR，当接收到新的数据时，就会发生接收溢出错误。(新的数据会丢失)

16.4. UART 寄存器的定义

以下寄存器用于 UART 操作。

寄存器 16-1： UCR：UART 控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	BPRREC	RXEN	TXEN
U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-3 未实现：读为 0

bit 2 BPRREC：UART 波特率自校准使能位
0 = 波特率依据 UBPRH：UBPRL 寄存器的设置
1 = 波特率自校准使能

bit 1 RXEN：UART 接收数据使能位
0 = 禁止接收
1 = 使能接收

bit 0 TXEN：UART 发送数据使能位
0 = 禁止发送
1 = 使能发送

寄存器 16-2： USETR：UART 数据格式控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	PARSEL1	PARSEL0	STOPBIT	BITNUM3	BITNUM2	BITNUM1	BITNUM0
U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 未实现：读为 0

bit 6-5 PARSEL<1:0>：UART 奇偶校验位配置位
00 = 不含奇偶校验位
01 = 奇校验
10 = 偶校验
11 = 含奇偶校验位，但是发送接收都不检测此位

bit 4 STOPBIT：停止位配置位
0 = 停止位 1bit
1 = 停止位 2bit

bit 3-0 BITNUM<3:0>：数据位配置位
1000 = 8 位
0111 = 7 位
0110 = 6 位
0101 = 5 位
其他 = 禁止设置

寄存器 16-3： UIER：UART 中断使能寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TXIE	RXIE	—	—	—	—	—	—
R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 TXIE：发送数据完成中断使能位
0 = 禁止发送数据完成中断
1 = 使能发送数据完成中断

bit 6 RXIE：接收数据完成中断使能位
0 = 禁止接收数据完成中断
1 = 使能接收数据完成中断

bit 5-0 未实现：读为 0

寄存器 16-4： UBPRL：UART 波特率配置寄存器低 8 位

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 UBPRL<7:0>：UART 波特率配置寄存器低 8 位
与 UBPRH 组成 12 位波特率配置寄存器

寄存器 16-4： UBPRH：UART 波特率配置寄存器高 4 位(续)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—				
U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 3-0 UBPRH<3:0>：UART 波特率配置寄存器高 4 位
与 UBPRL 组成 12 位波特率配置寄存器

注： UART 时钟频率 = OSC48M / 3 = 16M
UBPRL 为 UBPR 值的低 8 位，UBPRH 为 UBPR 的高 4 位。
UBPR 是根据波特率推算的配置数。
例如：波特率是 bpr115200，计算 UBPR 如下：
UBPR = f / bpr - 1 (F 的单位是 Hz) = 1000,000 * 16 / 115200 - 1 = 138 (0x8A)

寄存器 16-5： UTXR：UART 发送数据寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 UTXR<7:0>：UART 发送数据寄存器
UTXR 为发送数据寄存器，只有当 TXDONE = 1 时，才能将待发送的数据写入到该寄存器中。否则，写入的数据会丢失。

寄存器 16-6： URXR：UART 发送数据寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 URXR<7:0>：UART 接收数据寄存器

URXR 为接收数据寄存器，只有当 RXREADY = 1 时，才能读取该寄存器，获得接收的数据字节。否则，读出的数据无效。

注： 根据当前 UART 的配置，只有当停止位，奇偶校验位都正确时，才会将接收到的数据存入 URXR 寄存器中。若配置 UART 无奇偶校验位，或不关注奇偶校验位是否正确，则判断接收到的数据是否有效时，无须考虑奇偶校验结果。如果 RXREADY = 1 时不及时读取 URXR，则下一个接收的 Byte 数据将丢失。

寄存器 16-7： USR：UART 状态寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	TXDONE	RXREADY	OVER	STOPB	PAR
U-0	U-0	U-0	R-1	R-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-5 未实现：读为 0

bit 4 TXDONE：数据发送完成标志位

0 = 还未发送出去，此时不能往 UTXR 中写入数据

1 = 发送完成，可以往 UTXR 中写入数据

bit 3 RXREADY：接收 BYTE 数据完成标志位

0 = BYTE 数据还未接收完，不能读 URXR

1 = BYTE 数据已接收完，可以读 URXR 获取接收数据

bit 2 OVER：接收溢出错误标志位

0 = 未发生溢出错误

1 = 已发生溢出错误。溢出错误发送在接收到 1 Byte 数据后，软件未及时读 URXR，当接收到新的数据时，才会发生溢出错误

bit 1 STOPB：接收停止位位错误标志位

0 = 接收时未发现停止位有错误

1 = 接收时发现停止位有错误

bit 0 PAR：接收奇偶校验位错误标志位

0 = 接收时未发现奇偶校验位错误。当 PARSEL=00/11时，不检查奇偶校验位，认为奇偶校验位正确。

1 = 接收时发现奇偶校验位错误。若接收奇偶校验位出错，则当前字节丢弃，软件无法获得该字节。

17. SPI 模块

SPI 模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
SPICR	0xE0FF	XRAM	SPI 控制寄存器	0000 0000
SPISFMR	0xE1FF	XRAM	SPI 软件控制片选信号寄存器	0000 0001
SPIIER	0xE2FF	XRAM	SPI 中断使能寄存器	0000 0000
SPIBPR	0xE3FF	XRAM	SPI 波特率配置寄存器	0010 0000
SPISR	0xE4FF	XRAM	SPI 状态寄存器	0000 0001
SPIDR	0xE5FF	XRAM	SPI 数据接收发送寄存器	0000 0000

SPI 模块允许同时同步发送和接收 8 位数据。支持 SPI 的所有 4 种时序，使用以下 4 个引脚来完成通信：

- SPI 数据输出
- SPI 数据输入
- SPI 时钟信号
- SPI 片选信号

SPI 模块发送和接收都有 4 Byte 深度的数据缓冲器，在启动一组数据的传送前，必须软件写 SPICR 寄存器的 FIFOCLR 位为 1 清空数据缓冲器，防止 FIFO 中存在以前未传送完的数据。FIFOCLR 位为只写位，读为 0。

17.1. SPI 的操作步骤

配置 SPI 的操作步骤如下：

1. 端口配置

SPI 有主控和从动两种工作模式。所以存在两组端口配置方式。

SPI 配置为主控模式时，可通过 PxyIOCFG 寄存器将 I/O 配置成 SPIM 数据输入、SPIM 数据输出、SPIM 片选输出、SPIM 时钟输出功能。

SPI 配置为从动模式时，可通过 PxyIOCFG 寄存器将 I/O 配置 SPIS 数据输入、SPIS 数据输出、SPIS 片选输入、SPIS 时钟输入功能。

只要配置好后，会自动将引脚方向配置为相应的方向。不需要配置对应的 PnOE 寄存器，选择 SPI 引脚方向为输出/输入。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

2. SPI 模块时钟使能

在操作 SPI 模块相关寄存器前，必须首先使能 SPI 模块时钟，即将 MDLCKCR 寄存器的 SPIEN 位置 1。

3. SPI 模块参数配置：

- 主从模式选择

SPI 有主控和从动两种工作模式。通过 SPICR 寄存器的 MSSEL 位控制。必需注意的是 SPICR 的 MSSEL 位与其他位不能同时配置，必需先写 SPICR 的 MSSEL 位配置好模式后，才能配置其他位。

- 时钟速率(仅限主控模式)

时钟线是由主器件控制的，所以在主控模式下，时钟速率的配置才有效。当作为主控模式时，SPIBPR 寄存器可以控制 SPI 通讯的波特率。具体计算方法如下：

SPI 每个 bit 的时间 = 2 * (SPIBPR + 1) * Tspi;

其中 Tspi 为 SPI 的时钟周期，SPI 时钟是 OSC48M 的三分频(也就是 16M)。

虽然理论上 SPI 时钟速度可以达到 8M。但是由于 SPI 工作时钟是 16M，SPI 对输入输出会做滤波处理，所以 SPIBPR 最小只能配置为 0x06，此时时钟速度为 1.14Mhz。

- 片选信号产生方式选择

当 SPI 为主控模式时，可以通过 SPICR 寄存器的 CSMODE 位选择片选信号产生方式，有两种方式：

1) 硬件产生片选信号，此时通讯过程中，硬件自动产生拉高、拉低片选信号。
2) 软件产生片选信号，此时通过 SPISFMR 寄存器控制片选信号拉高、拉低。必须在时钟信号发送前将片选信号拉低，数据传送完成后再将片选信号拉高。若传送中片选拉高则会出错。

- 发送接收位顺序配置

SPICR 寄存器的 DORD 位清 0，选择 LSB 方式，发送时，先发送字节低位，接收时，先接收到的为字节低位。

SPICR 寄存器的 DORD 位置 1，选择 MSB 方式，发送时，先发送字节高位，接收时，先接收到的为字节高位。

- 时钟极性、时钟相位配置

SPI 有 4 中不同的时序，通过 SPICR 寄存器的 CPHA、CPOL 位，可配置成不同的通讯时序，CPHA 位配置时钟相位(采样的时间，上升沿或下降沿采样)，CPOL 位配置时钟极性(时钟线的空闲状态)，如图 17-1、图 17-2 所示：

4. 中断控制

SPIIER 寄存器是 SPI 中断使能寄存器，有 SPIIER<4:0>5 个位与 SPISR<4:0>一一对应，分别是发送 FIFO 空中断、发送 FIFO 满中断、接收 FIFO 非空中断、接收 FIFO 半满中断、接收 FIFO 满中断。SPISR<4:0>就是相应的中断标志位。不论中断是否使能，只要产生中断条件，相应的标志位就会置 1。

SPISR 寄存器还有 RXFOV(接收 FIFO 溢出标志位)、FWRSR(发送 FIFO 写状态位)、TRANS(SPI Master 数据传送启动位)。这些都只是标志位，不会产生中断。详细请查看寄存器介绍。

5. 接收发送使能

当参数都配置好以后，可以将 SPICR 寄存器的 RXEN 位置 1 使能数据接收，TXEN 置 1 使能数据发送。数据的发送和接收功能是相互独立的，不会受另一方的影响。

图 17-1： SPI 通讯时序(CPHA = 0)

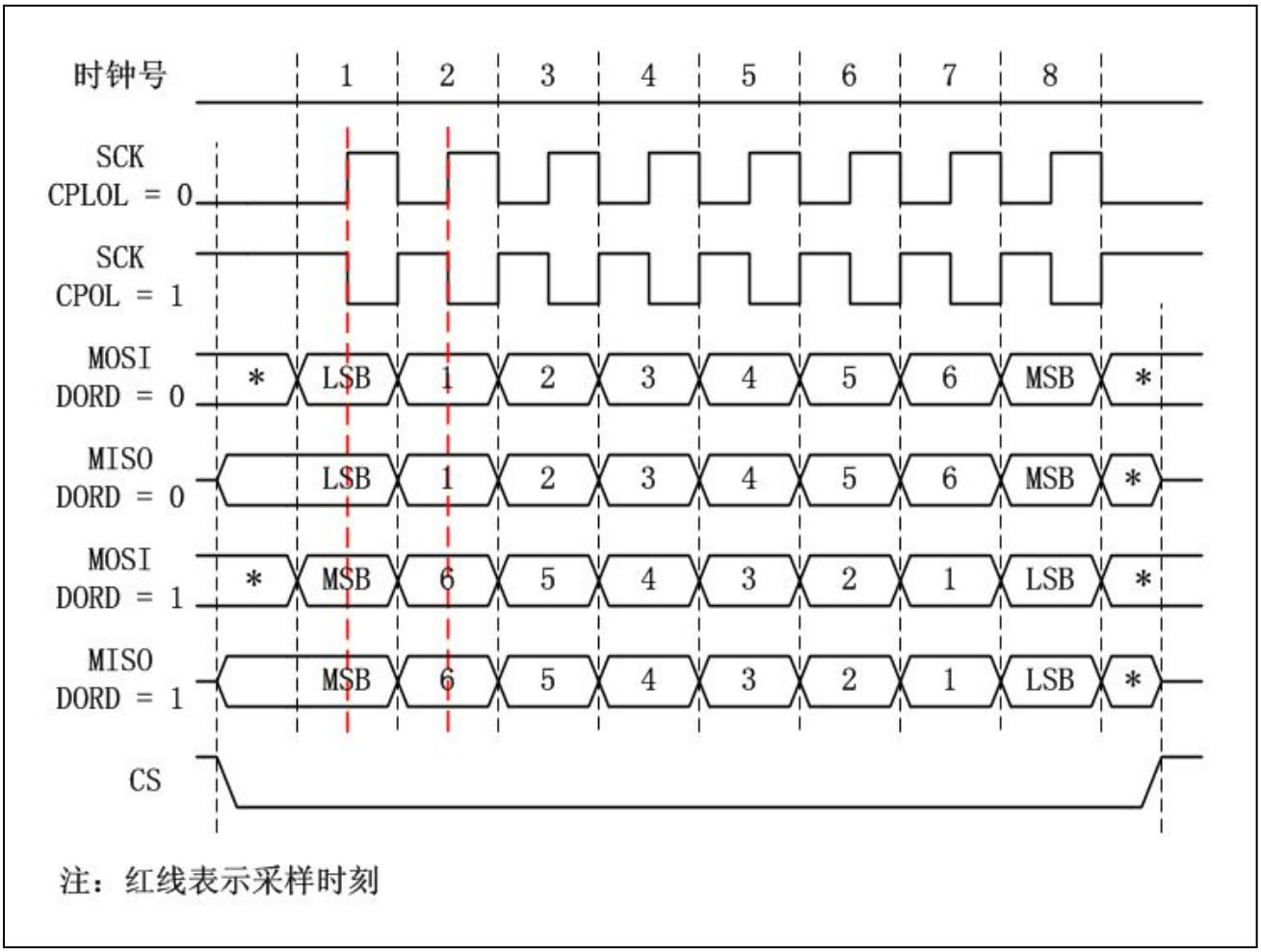
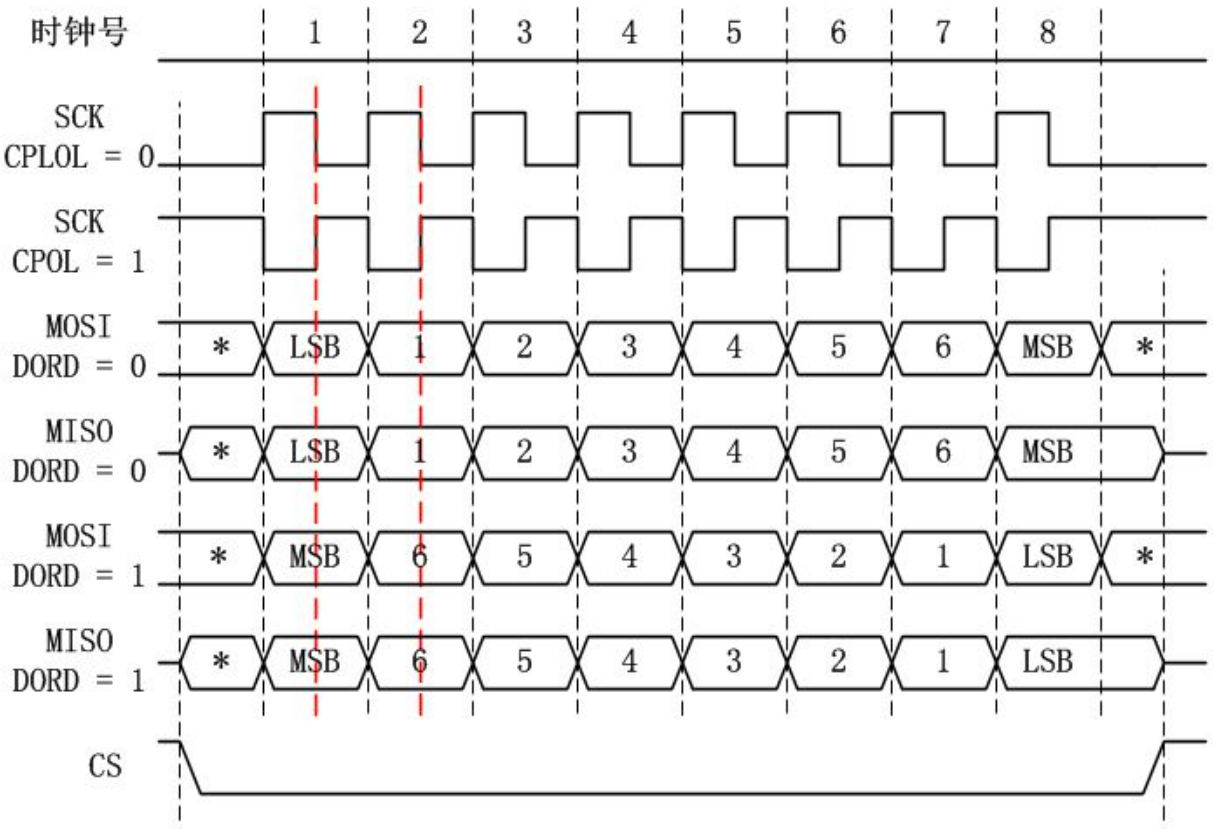


图 17-2： SPI 通讯时序(CPHA = 1)



注：红线表示采样时刻

17.2. 主控模式

主器件控制时钟线，因此可以随时启动数据传输。主器件根据软件协议确定从器件何时传输数据。

1）数据发送：

在主控模式下，软件通过写 SPIDR 寄存器，将数据写入到发送 FIFO 中。最多可写 4 字节数据，FIFO 已满将不能再写入数据。软件应先查询发送 FIFO 满标志位为 0，再写 SPIDR。

数据写入发送 FIFO 后，将 SPISR 寄存器的 TRANS 位置 1 即启动 1Byte 的数据传送，软件查询 TRANS 位为 0 时，即表示 Byte 传输结束。必须等待 TRANS 位为 0 后，才能再次将 TRANS 位置 1 启动下次 Byte 传输。每完成 1Byte 数据传送，发送 FIFO 的数据都会减少 1Byte，SPISR 寄存器的相应标志位也会做出改变。

2）数据接收：

在主控模式下，接收的数据先存入到接收 FIFO 中，深度也为 4Byte，软件读 SPIDR 寄存器，可将接收 FIFO 中的数据取出。

每读一次 SPIDR 寄存器，接收 FIFO 中缓存的数据都会减少 1Byte，SPISR 寄存器的相应标志位也会做出改变。软件上先查询接收 FIFO 非空标志位为 1，再读取 SPIDR。如果接收 FIFO 已满，SPI 仍继续接收数据，则会溢出。

3）时钟速度：

虽然理论上 SPI 时钟速度可以达到 8M。但是由于 SPI 工作时钟是 16M，SPI 对输入输出会做滤波处理，所以 SPIBPR 最小只能配置为 0x06，此时时钟速度为 1.14Mhz。

17.3. 从动模式

在从动模式下，当时钟引脚上出现外部时钟脉冲时，即启动数据发送和接收。当最后一位数据被锁存后，SPISR 寄存器的 RXFNOTEMPTY、RXFHALF、RXFFULL 位会产生相应变化。

1）数据发送和接收：

在从动模式下，发送接收数据与主控模式相同，区别是，从动模式不能控制何时通讯，只能被动的响应主器件的数据传送。

2）片选信号：

从动模式下，必须接收到片选信号，才会响应数据传送。就是说只有片选引脚为低电平时，才能发送和接收数据。

> 注：软件仅在 SPISR 寄存器的 FWRSR 位为 0 时，才可以写 SPIDR 寄存器将待发送数据写入发送 FIFO 中。

17.4. SPI 寄存器的定义

以下寄存器用于 SPI 操作。

寄存器 17-1： SPICR：SPI 控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MSSEL	CSMODE	DORD	FIFOCLR	CPHA	CPOL	TXEN	RXEN
R/W-0	R/W-0	R/W-0	W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 MSSEL：主从模式控制位
0 = 从模式
1 = 主模式

bit 6 CSMODE：片选信号产生方式选择位
0 = 硬件方式产生，通讯过程中，每个字节的通讯，CS 都会自动变低再变高。
1 = 软件方式产生
此后，软件写 SPISFMR 为1即拉高 CS，写0即拉低 CS。
注：软件方式可以实现 CS 保持为低时，完成多个字节通讯

bit 5 DORD：发送接收位顺序控制位
0 = LSB 方式
1 = MSB 方式

bit 4 FIFOCLR：清 FIFO 位
只写位，软件写 1 清空 FIFO

bit 3 CPHA：时钟相位选择位
0 = 在时钟的第一个边沿采样数据
1 = 在时钟的第二个边沿采样数据

bit 2 CPOL：时钟极性选择位
0 = 时钟的空闲电平为低电平
1 = 时钟的空闲电平为高电平

bit 1 TXEN：SPI 发送数据使能位
0 = 禁止发送
1 = 使能发送

bit 0 RXEN：SPI 接收数据使能位
0 = 禁止接收
1 = 使能接收

寄存器 17-2： SPISFMR：SPI 软件控制片选信号寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	—	—	
U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-1 未实现：读为 0

bit 0 SPISFMR：SPI 软件控制片选信号寄存器

0 = 拉低片选信号

1 = 拉高片选信号

寄存器 17-3： SPIIER：SPI 中断使能寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	RXFFULL	RXFHALF	RXFNOTEMPTY	TXFFULL	TXFEMPTY
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-5 未实现：读为 0

bit 4 RXFFULL：接收 FIFO 满中断使能位

0 = 禁止接收 FIFO 满中断

1 = 使能接收 FIFO 满中断

bit 3 RXFHALF：接收 FIFO 半满中断使能位

0 = 禁止接收 FIFO 半满中断

1 = 使能接收 FIFO 半满中断

bit 2 RXFNOTEMPTY：接收 FIFO 非空中断使能位

0 = 禁止接收 FIFO 非空中断

1 = 使能接收 FIFO 非空中断

bit 1 TXFFULL：发送 FIFO 满中断使能位

0 = 禁止发送 FIFO 满中断

1 = 使能发送 FIFO 满中断

bit 0 TXFEMPTY：发送 FIFO 空中断使能位

0 = 禁止发送 FIFO 空中断

1 = 使能发送 FIFO 空中断

寄存器 17-4： SPIBPR：SPI 波特率配置寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—							
U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7　　未实现：读为 0

bit 6-0　　SPIBPR<6:0>：SPI 波特率配置寄存器

当 SPI 作为主控模式时，SPIBPR 控制 SPI 通讯的波特率。

SPI 每个 bit 的时间 = 2 * (SPIBPR + 1) * Tspi

(Tspi 为 SPI 的时钟周期是振荡时钟的三分频)

寄存器 17-5： SPIDR：SPI 数据接收发送寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0　　SPIDR<7:0>：SPI 数据接收发送寄存器

接口发送数据时，软件通过写此寄存器，将数据写入到发送 FIFO 中。

接口接收数据时，数据首先存入到接收 FIFO 中，软件读此寄存器，将接收 FIFO 中的数据取出。

寄存器 17-6： SPISR：SPI 状态寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TRANS	FWRSR	RXFOV	RXFFULL	RXFHALF	RXFNOTEMPTY	TXFFULL	TXFEMPTY
R/W-0	R-0	R/W-0	R-0	R-0	R-0	R-0	R-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 TRANS：主控模式下，Byte 数据传送启动位
0 = 数据传送已完成或不在进行中
1 = 启动数据传送，完成后自动清 0

bit 6 FWRSR：SPI 发送 FIFO 写状态位
0 = 可写 SPIDR 寄存器将数据写入发送 FIFO。
1 = 不可写 SPIDR 寄存器将数据写入发送 FIFO。

bit 5 RXFOV：接收 FIFO 溢出标志位
0 = 接收 FIFO 未溢出
1 = 接收 FIFO 已经溢出

bit 4 RXFFULL：接收 FIFO 满标志位
0 = 接收 FIFO 不满
1 = 接收 FIFO 已满

bit 3 RXFHALF：接收 FIFO 半满标志位
0 = 接收 FIFO 没有半满
1 = 接收 FIFO 半满

bit 2 RXFNOTEMPTY：接收 FIFO 非空标志位
0 = 接收 FIFO 空
1 = 接收 FIFO 非空

bit 1 TXFFULL：发送 FIFO 满标志位
0 = 发送 FIFO 不满
1 = 发送 FIFO 已满

bit 0 TXFEMPTY：发送 FIFO 空标志位
0 = 发送 FIFO 非空
1 = 发送 FIFO 已空

注： 1、发送或接收每个 Byte 都需要写 TRANS 位。软件查询 TRANS 位为 0 时，即表示 Byte 传输结束。
2、软件仅在 FWRSR 位为 0 时，才可以写 SPIDR 寄存器将待发送数据写入发送 FIFO 中。
3、当 RXFFULL 位为 1，SPI 继续接收到新的 Byte 数据，则会溢出，RXFOV 位会置 1。所以软件应及时将接收 FIFO 中的数据读出。

18. I2C 模块

I²C 模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
I2CSCR	0xD8FF	XRAM	I2C Slave 控制寄存器	0000 0000
I2CSSR	0xD9FF	XRAM	I2C Slave 状态寄存器	0000 0000
I2CSDRSEL	0xDAFF	XRAM	I2C Slave 辅助数据寄存器	0000 0000
I2CSDR	0xDBFF	XRAM	I2C Slave 数据寄存器	0000 0000
I2CSADDR	0xDCFF	XRAM	I2C Slave 设备地址寄存器	0010 0010
I2CSRADR	0xDDFF	XRAM	I2C Slave 接收地址寄存器	0000 0000
I2CSHADR	0xDEFF	XRAM	I2C Slave 硬件地址寄存器	0000 0000
I2CMPRERL	0xC8FF	XRAM	I2C Master 波特率配置寄存器低 8 位	0000 0000
I2CMPRERH	0xC9FF	XRAM	I2C Master 波特率配置寄存器高 8 位	0000 0000
I2CMCR	0xCAFF	XRAM	I2C Master 控制寄存器	0000 0000
I2CMDR	0xCBFF	XRAM	I2C Master 数据寄存器	0000 0000
I2CMCTR	0xCCFF	XRAM	I2C Master 传输控制寄存器	0000 0000
I2CMSR	0xCDFF	XRAM	I2C Master 状态寄存器	0000 0000

I²C 模块包含主控和从动两个模块，这两个模块是相互独立的，可以同时工作。

I²C 模块支持 7 位寻址。有两个引脚用于数据传输。分别是时钟引脚和数据引脚。当 IO 复用功能寄存器配置为 I²C 功能(I²C Slave 数据线、I²C Slave 时钟线、I²C Master 数据线、I²C Master 时钟线)后，会自动将引脚方向配置为相应的方向。不需要配置对应的 PnOE 寄存器，选择 I²C 引脚方向为输出/输入。

18.1. I2C Slave(从动模块)

I²C Slave 有 4 种工作方式：

- I²C Slave 硬件快速模式(只读)
- I²C Slave 硬件标准模式(只读)
- I²C Slave 软件快速模式(只读)
- I²C Slave 软件标准模式(读写)

在这 4 种模式下的通讯时序有所不同。硬件快速、硬件标准、软件快速这三种模式下，主机只能从从机读数据而不能对从机写数据。只有软件标准模式下，主机从可以对从机进行读写。具体的通讯时序见对应小节。

18.1.1. I²C Slave 的操作步骤

配置 I²C Slave 的操作步骤如下：

1. 端口配置

当 IO 复用功能寄存器配置为 I²C Slave 数据线、I²C Slave 时钟线后，会自动将引脚方向配置为相应的方向。不需要配置对应的 PnOE 寄存器，选择 I²C Slave 引脚方向为输出/输入。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

PnPU 寄存器可以使能对应时钟、数据线的内部上拉电阻，PnPURSELRH：PnPURSELRL 寄存器可以配置内部上拉电阻的大小。

2. I²C Slave 模块参数配置：

- 设备地址配置

主机发送开始信号后，紧接 7 位地址+1 位读/写位，这 7 位地址数据必须与 I2CSADDR 寄存器的<6:0>位匹配，从机才会做出应答。通过配置 I2CSADDR 寄存器为不同的值，可以实现多机通讯(一主多从)。I2CSADDR 的第 7 位是记录主机读写状态的位，0 表示写，1 表示读。

- 硬件地址配置(在硬件标准模式下)

硬件地址配置只在硬件标准模式下有作用。只有在硬件标准模式下，当发送的设备地址匹配后，需要接着发送 8 位硬件地址。当这 8 位硬件地址数据必须与 I2CSHADR 寄存器匹配时，从机才会做出应答。

- 待发送数据配置

主机读从机数据时，从机必须先把待发送数据写入内部数据寄存器中。通过 I2CSDRSEL 和 I2CSDR 寄存器，将待发送数据写入到内部数据寄存器中。

1) 硬件模式下，内部数据寄存器为 4 个字节：

配置 I2CSDRSEL=00，将数据字节 0 写到 I2CSDR 寄存器；

配置 I2CSDRSEL=01，将数据字节 1 写到 I2CSDR 寄存器；

配置 I2CSDRSEL=10，将数据字节 2 写到 I2CSDR 寄存器；

配置 I2CSDRSEL=11，将数据字节 3 写到 I2CSDR 寄存器。

2) 软件模式下，内部数据寄存器为 2 个字节：

配置 I2CSDRSEL=00，将数据字节 0 写到 I2CSDR 寄存器；

配置 I2CSDRSEL=01，将数据字节 1 写到 I2CSDR 寄存器。

- 工作模式配置

I^2C Slave 有 4 种工作模式，通过 I2CSCR 寄存器的 MODE<1:0>位可以配置 I^2C Slave 的工作模式：

- MODE<1:0> = 00；软件标准模式
- MODE<1:0> = 01；软件快速模式
- MODE<1:0> = 10；硬件标准模式
- MODE<1:0> = 11；硬件快速模式

3. 清 I^2C Slave 中断标志位

使能 I^2C Slave 中断前，先将 I2CSSR 寄存器清 0，清 I^2C Slave 相关中断标志位。

4. 中断控制

I2CSCR 寄存器的 INTEN 位是 I^2C Slave 中断使能位，有 INTEN<3:0>4 个位与 I2CSSR 寄存器的<3:0>位一一对应，分别是 I^2C Slave 接收到地址中断、I^2C Slave 接收到字节数据中断、I^2C Slave 发送完字节数据中断、I^2C Slave 接收到设备地址中断(只在软件快速模式下产生)。I2CSSR<3:0>就是相应的中断标志位。不论中断是否使能，只要产生中断条件，相应的标志位就会置 1。需要注意的是，在不同模式下，产生不同的中断：

1）硬件快速模式下，不产生中断；

2）硬件标准模式下，只产生 I^2C Slave 接收到地址中断，此时的地址是硬件地址，保存在 I2CSRADR 寄存器中；

3）软件快速模式下，只产生 I^2C Slave 发送完字节数据中断、I^2C Slave 接收到设备地址中断；

4）软件标准模式下，只产生 I^2C Slave 接收到地址中断、I^2C Slave 接收到字节数据中断、I^2C Slave 发送完字节数据中断。

5. 使能响应主机读写数据

I2CSCR 寄存器的 ACKEN 位是使能响应主机读写数据位，只有将 ACKEN 置为 1，从机才会响应主机的命令；否则就算时序正确，从机也不会回应主机。

18.1.2. I2C Slave 硬件快速模式(只读)

软件操作步骤：

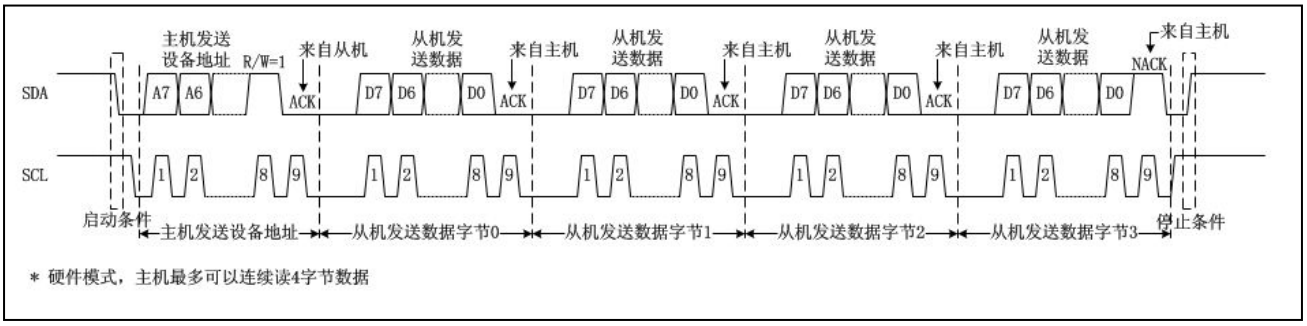
1、I²C Slave 配置成硬件快速模式，使能响应主机读写；

2、通过 I2CSDRSEL 寄存器和 I2CSDR 寄存器，将预发送数据写入到内部数据寄存器中，最多可写入 4 字节数据；

3、I²C 主机发送匹配的设备地址和读标志，即可从 Slave 读取最多 4 个字节数据。

硬件快速模式下的通讯时序如图 18-1 所示。

图 18-1： 硬件快速模式主机读时序图(只读)



18.1.3. I2C Slave 硬件标准模式(只读)

软件操作步骤：

1、I²C Slave 配置成硬件标准模式，使能响应主机读写；

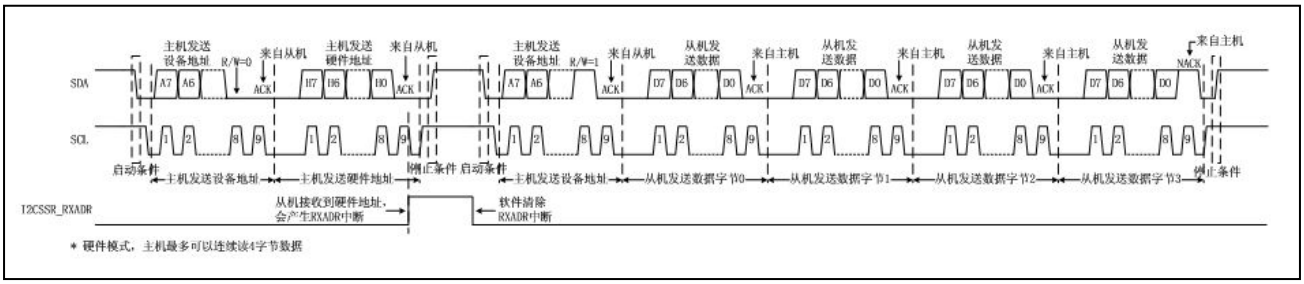
2、配置 I2CSHADR 寄存器；

3、通过 I2CSDRSEL 寄存器和 I2CSDR 寄存器，将预发送数据写入到内部数据寄存器中，最多可写入 4 字节数据。

4、I²C 主机发送匹配的设备地址和写标志，再送入硬件地址，硬件地址要与从机软件配置的 I2CSHADR 寄存器一致，然后主机再次发送匹配的设备地址和读标志，即可从 Slave 读取最多 4 个字节数据。

硬件标准模式下的通讯时序如图 18-2 所示。

图 18-2： 硬件标准模式主机读时序图(只读)



18.1.4. I2C Slave 软件快速模式(只读)

软件操作步骤：

1、I²C Slave 配置成软件快速模式，使能响应主机读写；

2、使能接收到设备地址中断；

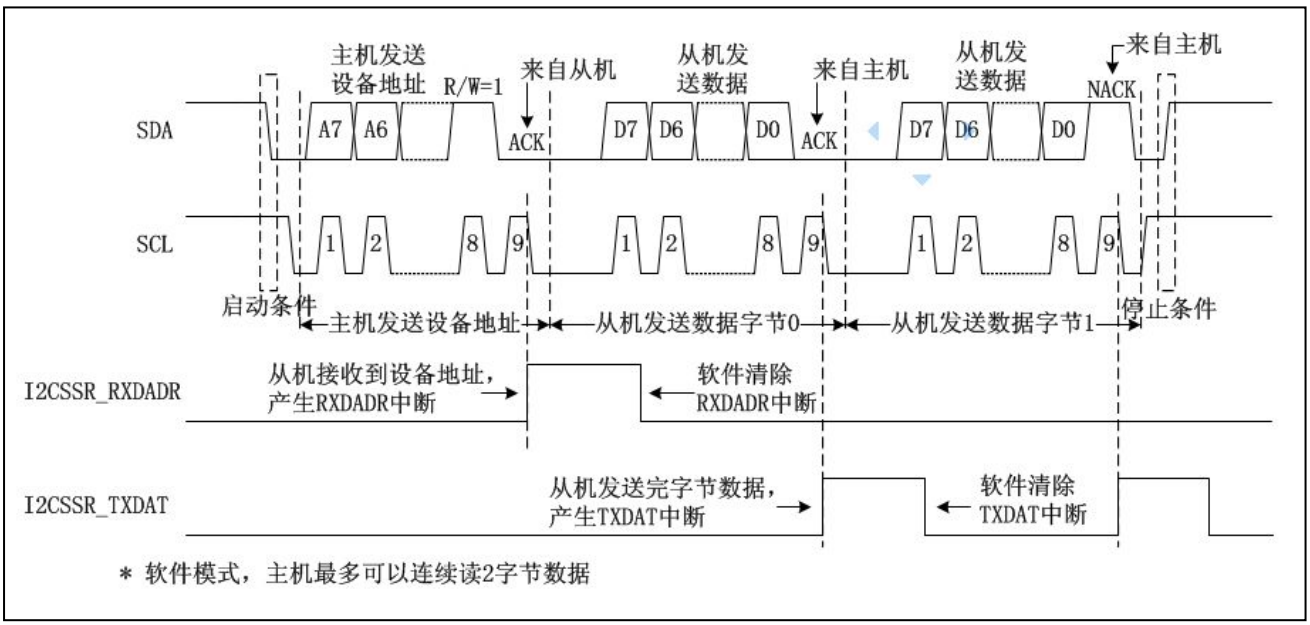
3、通过 I2CSDRSEL 寄存器和 I2CSDR 寄存器，将预发送数据写入到内部数据寄存器中。最多可以写入 2 字节数据；

4、I²C 主机发送匹配的设备地址和读标志，即可从 Slave 读取最多 2 个字节数据；

5、从机中断触发后，通过查询发送完字节数据中断标志位，通过 I2CSDRSEL 寄存器和 I2CSDR 寄存器，将后续待发送数据写入到内部数据寄存器中。

软件快速模式通讯时序如图 18-3 所示。

图 18-3：软件快速模式主机读时序图(只读)



18.1.5. I2C Slave 软件标准模式(读写)

软件操作步骤：

1、I²C Slave 配置成软件标准模式，使能响应主机读写；

2、使能接收到地址中断；

3、通过 I2CSDRSEL 寄存器和 I2CSDR 寄存器，将预发送数据写入到内部数据寄存器中。最多可以写入 2 字节数据；

4、若主机执行写操作，I²C 主机发送匹配的设备地址和写标志，再送入内存地址，再紧跟待写入数据。从机中断触发后，根据接收到的内存地址的值以及接收数据，将数据存入到对应的地址中；

6、若主机执行读操作，I²C 主机发送匹配的设备地址和写标志，再送入内存地址。然后主机再次发送匹配的设备地址和读标志。从机中断触发后，通过查询发送完字节数据中断标志位，根据接收到的内存地址的值，调配相应的数据，通过 I2CSDRSEL 寄存器和 I2CSDR 寄存器，将后续待发送数据写入到内部数据寄存器中。

软件标准模式的通讯写时序如图 18-4 所示，软件标准模式的通讯读时序如图 18-5 所示。

图 18-4： 软件标准模式主机写时序图

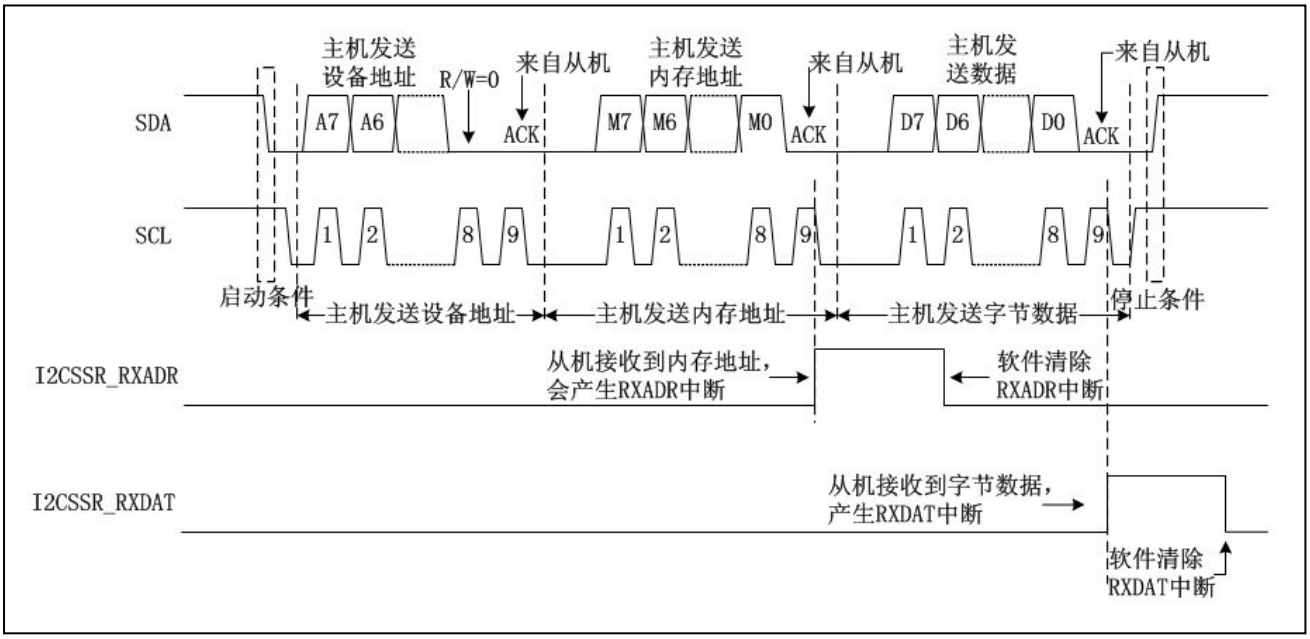
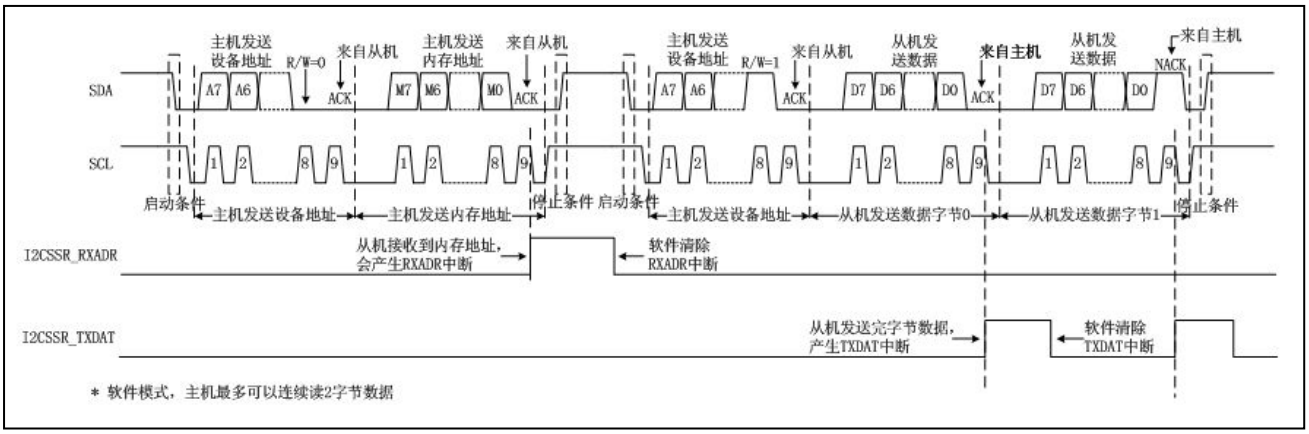


图 18-5： 软件标准模式主机读时序图



18.1.6. I2C Slave 寄存器的定义

以下寄存器用于 I²C Slave 操作。

寄存器 18-1： I2CSCR：I2C Slave 控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	MODE1	MODE0	ACKEN	RXDADR	TXDAT	RXDAT	RXADR
U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 未定义：读为 0

bit 6-5 MODE<1:0>：I²C Slave 工作模式模式选择位
00 = 软件标准模式
01 = 软件快速模式
10 = 硬件标准模式
11 = 硬件快速模式

bit 4 ACKEN：响应主机读写使能位
0 = 禁止响应主机读写
1 = 使能响应主机读写

bit 3 RXDADR：I²C Slave 接收到设备地址中断使能位(只在软件快速模式下产生)
0 = 禁止 I²C Slave 接收到设备地址中断
1 = 使能 I²C Slave 接收到设备地址中断

bit 2 TXDAT：I²C Slave 发送完字节数据中断使能位(只在软件快速、软件标准模式下产生)
0 = 禁止 I²C Slave 发送完字节数据中断
1 = 使能 I²C Slave 发送完字节数据中断

bit 1 RXDAT：I²C Slave 接收到字节数据中断使能位
0 = 禁止 I²C Slave 接收到字节数据中断
1 = 使能 I²C Slave 接收到字节数据中断

bit 0 RXADR：I²C Slave 接收到地址中断使能位
0 = 禁止 I²C Slave 接收到地址中断
1 = 使能 I²C Slave 接收到地址中断

寄存器 18-2： I2CSSR：I2C Slave 状态寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	RXDADR	TXDAT	RXDAT	RXADR
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 未定义：读为 0

bit 3 RXDADR：I²C Slave 接收到设备地址中断标志位(只在软件快速模式下产生)
0 = I²C Slave 未接收到设备地址
1 = I²C Slave 接收到了设备地址

bit 2 TXDAT：I²C Slave 发送完字节数据中断标志位(只在软件快速、软件标准模式下产生)
0 = I²C Slave 字节数据还未发送完
1 = I²C Slave 字节数据发送完成

bit 1 RXDAT：I²C Slave 接收到字节数据中断标志位
0 = I²C Slave 未接收到字节数据
1 = I²C Slave 接收到了字节数据

bit 0 RXADR：I²C Slave 接收到地址中断标志位(这个地址是内存地址或硬件地址，不是设备地址)
0 = I²C Slave 未接收到地址
1 = I²C Slave 接收到了地址

寄存器 18-3： I2CSDRSEL：I2C Slave 辅助数据寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	—		
U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-2 未定义：读为 0

bit 1-0 I2CSDRSEL<1:0>：I²C Slave 辅助数据寄存器
将待发送数据写入到内部数据寄存器，是通过 I2CSDRSEL 和 I2CSDR 寄存器的配合来实现的。详细请查看 18.1.1.4 小节“待发送数据配置”

寄存器 18-4： I2CSDR：I2C Slave 数据寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 I2CSDR<7:0>：I²C Slave 数据寄存器

将待发送数据写入到内部数据寄存器，是通过 I2CSDRSEL 和 I2CSDR 寄存器的配合来实现的。详细请查看 18.1.1.4 小节“待发送数据配置”

寄存器 18-5： I2CSADDR：I2C Slave 设备地址寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
WRSR	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 WRSR：I²C 主机读写状态位

0 = I²C 主机执行写操作

1 = I²C 主机执行读操作

bit 6-0 ADDR<6:0>：I²C Slave 设备地址位

配置 I²C 设备地址，仅当 I²C 主设备发送的设备地址与此寄存器数值匹配时，才能正常通讯。

寄存器 18-6： I2CSRADR：I2C Slave 接收地址寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 I2CSRADR<7:0>：I²C Slave 接收地址寄存器

I²C 主设备发送的地址会保存在 I2CSRADR 中，软件可以读取 I2CSRADR 获得地址值。

寄存器 18-7： I2CSHADR：I2C Slave 硬件地址寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 I2CSHADR<7:0>：I²C Slave 硬件地址寄存器

在硬件标准模式下，如果 I²C 主设备发送硬件地址，需与此寄存器匹配时，才能读取数据。

18.2. I2C Master(主控模块)

所有串行时钟脉冲和启动/停止条件均由主器件产生。停止条件或重复启动条件能结束传输。因为重复启动条件也就是下一次串行传输的开始，因此不会释放 I²C 总线。

在主控发送模式下，串行数据通过数据线输出，而串行时钟由时钟线输出。发送的第一个字节包括接收器件的设备地址(7 位)和 R/W 位。在发送模式下，R/W 位将是逻辑 0。因此，发送的第一个字节是一个 7 位从器件设备地址，后面跟 0 表示发送。串行数据每次发送 8 位。每发送一个字节，会收到一个应答位。启动和停止条件的输出表明串行传输的开始和结束。

在主控接收模式下，串行数据通过数据线接收，而串行时钟由时钟线输出。发送的第一个字节包括发送器件的设备地址(7 位)和 R/W 位。在接收模式下，R/W 位将是逻辑 1。因此，发送的第一个字节是一个 7 位从器件设备地址，后面跟 1 表示接收。每次接收 8 位串行数据。每接收到一个字节，都会发送一个应答位。启动和停止条件的输出表明串行传输的开始和结束。

在 I²C 主控模式下，可以通过 I2CMPRERH:I2CMPRERL 寄存器配置 I²C 通讯的波特率。

18.2.1. I²C Master 的操作步骤

配置 I²C Master 的操作步骤如下：

1. 端口配置

当 IO 复用功能寄存器配置为 I²C Master 数据线、I²C Master 时钟线后，会自动将引脚方向配置为相应的方向。不需要配置对应的 PnOE 寄存器，选择 I²C Master 引脚方向为输出/输入。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

PnPU 寄存器可以使能对应的时钟数据线的内部上拉电阻，PnPURSELRH：PnPURSELRL 可以配置内部上拉电阻的大小。

2. 波特率配置

I²C Master 模块的时钟是由 OSC48M 经过 MDLCKCR 寄存器的 SYSDIV<5:0>位分频后的时钟。通过配置 I2CMPRERH: I2CMPRERL 寄存器的值，可以调整 I²C Master 的通讯波特率。

计算方法如下：

I²C Master 波特率 = OSC48M / (SYSDIV<5:0>+1) / (I2CMPRERH:I2CMPRERL + 1) × 5;

例如：SYSDIV<5:0> = 2，I²C Master 波特率=100KHz;

I2CMPRERH:I2CMPRERL = 48MHz/3/(5*100KHz) - 1

= 31;

= 0x1F;

3. 清 I²C Master 中断标志位

使能 I²C Master 中断前，先将 I2CMCTR 寄存器的 CLRIF 位置 1，清 I²C Master 中断标志位。

4. 中断控制

I²C Master 只有一个中断，若中断使能，每完成 8 位数据传输或者当总线仲裁失败就会产生中断。中断使能位是 I2CMCR 寄存器的 IE 位，I2CMSR 寄存器的 IF 位是对应的中断标志位。不论中断是否使能，只要产生中断条件，相应的标志位就会置 1。

I2CMSR 寄存器除了 IF 中断标志位，还有许多标志位，这些只是标志位，不会产生中断。详细请查看寄存器定义。

5. 主控模式使能

将 I2CMCR 寄存器的 I2CMEN 位置 1，使能 I²C Master 模块功能。使能 I²C Master 后，才能进行数据传输。

6. 传输控制

一旦使能主控模式，用户可以进行以下操作：

- 发出一个启动条件
- 产生一个停止条件
- 开始数据/地址的发送
- 开始接收数据
- 在接收到数据字节后产生应答条件

这些操作都是通过 I2CMCTR 寄存器来控制的，但是需要注意的是启动条件、停止条件和应答条件都不是单独控制的，必须和 8 位数据/地址的传输组合在一起。

例如发出一个启动条件：

1) 因为启动条件后面必须接一个 7 位设备地址加一位 R/W 位，所以先把传送的 7 位设备地址+1 位 R/W 位的数据写入 I2CMDR 寄存器。

2) 把 I2CMCTR 寄存器的 START 位、WR 位同时置 1。

3) 当 START、WR 位同时置 1 后，即启动传输。I²C Master 模块就会发出一个启动条件，后面接着发送 7 位设备地址+1 位 R/W 位组成的 8 位数据。传送完成后，START 和 WR 位会自动清 0。

18.2.2. 范例程序

下面是两个 I²C Master 的范例程序。

例 1：向从机写 1 个字节数据

从机地址：0x4E

字节数据：0x65

```
I2CMTRDR = 0x9C;                    //设备地址+写
I2CMCTR = 0x90;                     //START+WR
while(0x02 == I2CMSR_TIP) {;}       //等待传送完成
if(0x00 == I2CMSR_RXACK) {          //查询是否收到应答信号
    I2CMTRDR = 0x65;                //待发送数据
    I2CMCTR = 0x50;                 //STOP+WR
    while(0x02 == I2CMSR_TIP) {;}   //等待传送完成
}
```

例 2：读取从机 1 个字节数据

从机地址：0x52

读数据地址：0x20

```
I2CMTRDR = 0xA4;                    //设备地址+写
I2CMCTR = 0x90;                     //START+WR
while(0x02 == I2CMSR_TIP) {;}       //等待传送完成
if(0x00 == I2CMSR_RXACK) {          //查询是否收到应答信号
    I2CMTRDR = 0x20;                //读数据地址
    I2CMCTR = 0x10;                 //WR
    while(0x02 == I2CMSR_TIP) {;}   //等待传送完成
    if(0x00 == I2CMSR_RXACK) {      //查询是否收到应答信号
        I2CMTRDR = 0xA5;            //设备地址+读
        I2CMCTR = 0x90;             //START+WR
        while(0x02 == I2CMSR_TIP) {;}   //等待传送完成
        if(0x00 == I2CMSR_RXACK) {      //查询是否收到应答信号
            I2CMCTR = 0x62;             //STOP+RD+不发送应答信号
            while(0x02 == I2CMSR_TIP) {;}//等待传送完成
            dat = I2CMTRDR;             //保存读到的数据
        }
    }
}
```

18.2.3. I2C Master 的时序

以下是启动、停止、应答、发送、接收的时序的时序图。

图 18-6： 主控模式启动条件时序

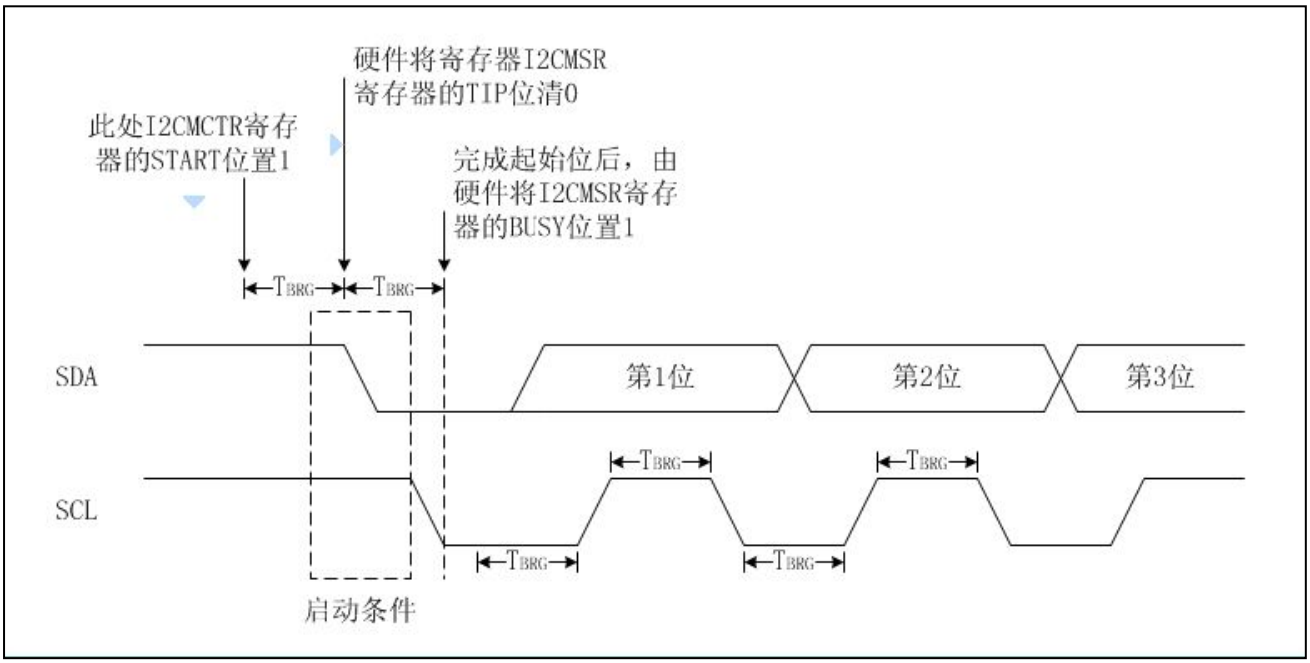


图 18-7： 主控模式停止条件时序

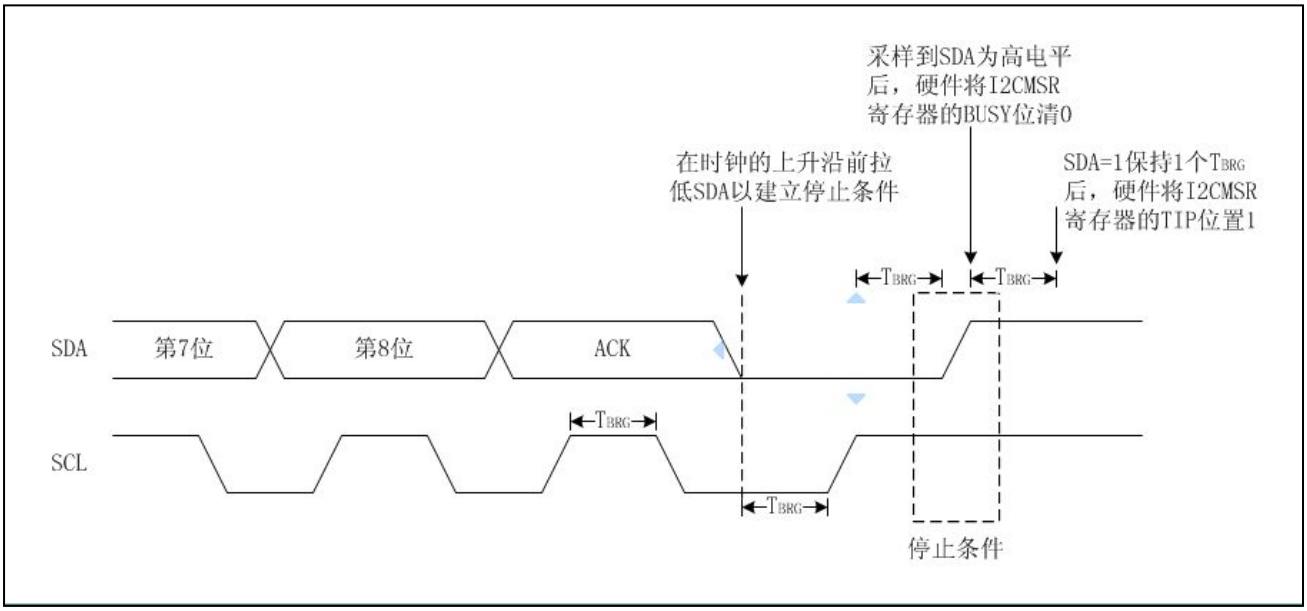


图 18-8： 主控模式应答时序图

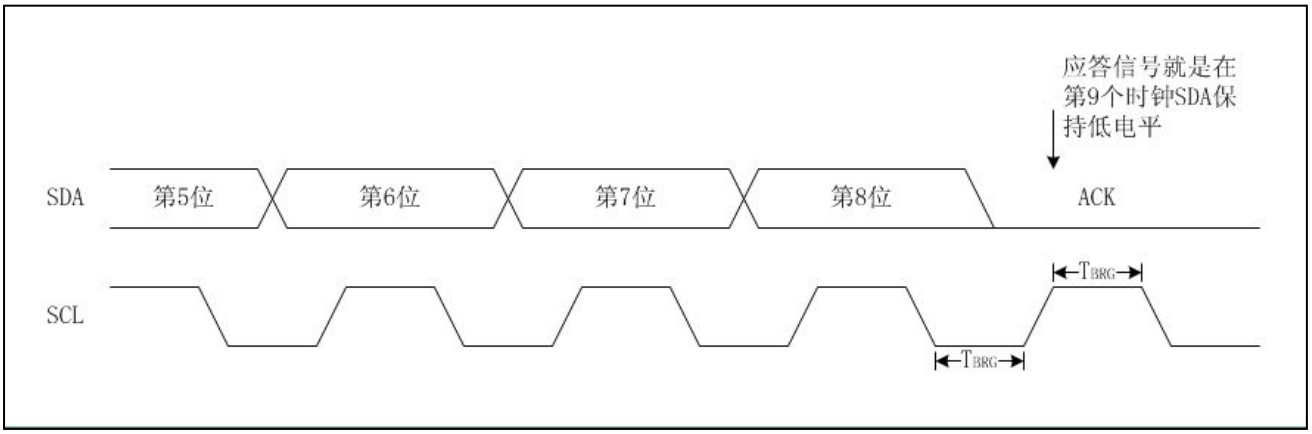


图 18-9： 主控模式发送时序图

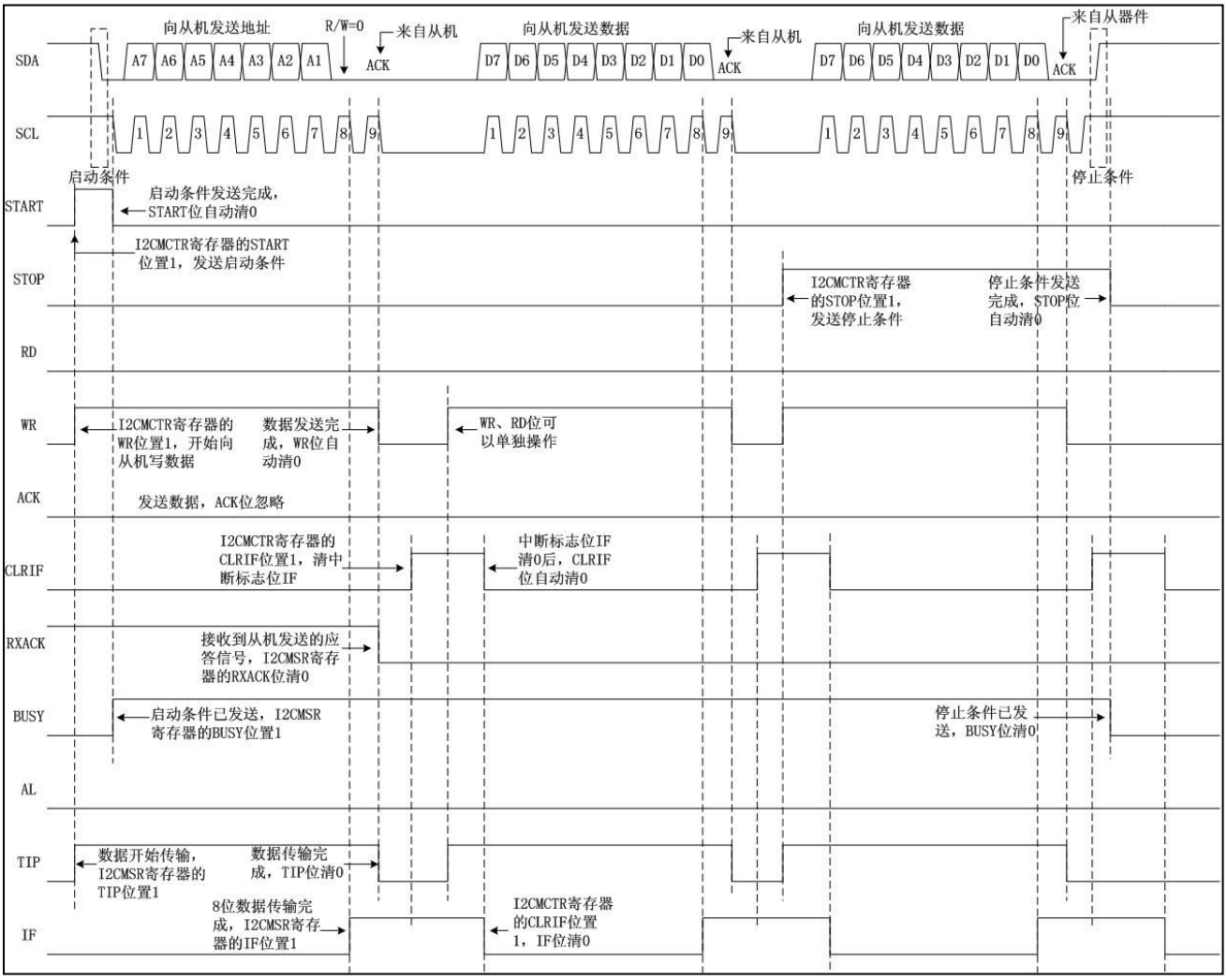
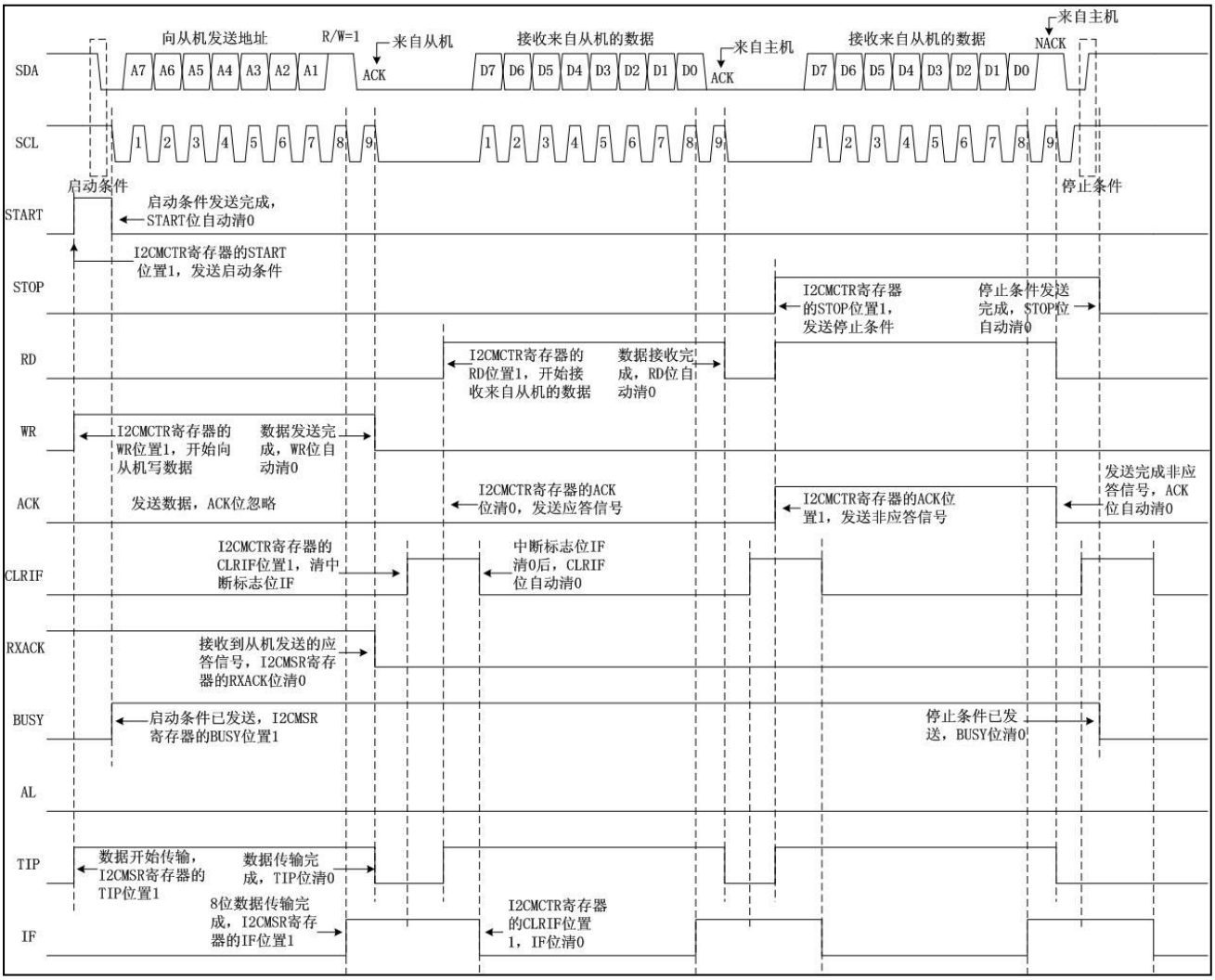


图 18-10： 主控模式接收时序图



18.2.4. I2C Master 寄存器的定义

以下寄存器用于 I²C Master 操作。

寄存器 18-8： I2CMPRERL：I2C Master 波特率配置寄存器低 8 位

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 I2CMPRERL<7:0>：I²C Master 波特率配置寄存器低 8 位
与 I2CMPRERH 寄存器组成 16 位 I²C Master 波特率配置寄存器

寄存器 18-8： I2CMPRERH：I²C Master 波特率配置寄存器高 8 位(续)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 I2CMPRERH<7:0>：I²C Master 波特率配置寄存器低 8 位
与 I2CMPRERL 寄存器组成 16 位 I²C Master 波特率配置寄存器

注： I²C Master 的波特率 = OSC48M / (SYSDIV<5:0> + 1) / (I2CMPRERH:I2CMPRERL + 1) × 5

寄存器 18-9： I2CMCR：I2C Master 控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
I2CMEN	IE	—	—	—	—	—	—
R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 I2CMEN：I^2C 主控模式使能位
0 = 禁止 I^2C 主控模式
1 = 使能 I^2C 主控模式

bit 6 IE：I^2C Master 中断使能位
0 = 禁止 I^2C Master 中断
1 = 使能 I^2C Master 中断

bit 5-0 未实现:读为 0

寄存器 18-10： I2CMDR：I2C Master 数据寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 I2CMDR<7:0>：I^2C Master 数据寄存器
I^2C Master 数据寄存器，保存待发送数据以及接收到的数据

注： 1、发送数据时，保存待发送的 8 位数据；
2、发送从机设备地址时，保存 7 位设备地址+1 位 R/W 位，其中 bit7-1 是设备地址，bit0 是 R/W 位；
3、接收数据时，保存接收到的 8 位数据。

寄存器 18-11： I2CMCTR：I2C Master 传输控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
START	STOP	RD	WR	—	—	ACK	CLRIF
W-0	W-0	W-0	W-0	U-0	U-0	W-0	W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 START：启动条件控制位
START 位置 1，发送启动条件

bit 6 STOP：停止条件控制位
STOP 位置 1，发送停止条件

bit 5 RD：从从机读数据控制位
RD 位置 1，开始从从机读数据

bit 4 WR：向从机写数据控制位
WR 位置 1，开始向从机写数据

bit 3-2 未实现：读为 0

bit 1 ACK：主机接收到数据时，应答条件控制位
0 = 发送应答条件
1 = 不发送应答条件

bit 0 CLRIF：清除中断标志
CLRIF 位置 1，清除 I^2C Master 中断标志位

注： 1、这些位写一后，会自动清零。

2、START、STOP、ACK 位不能单独使用，必须配合 WR、RD 位使用。但是 RD、WR 位可以单独使用。

寄存器 18-12： I2CMSR：I2C Master 状态寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RXACK	BUSY	AL	—	—	—	TIP	IF
R-0	R-0	R-0	U-0	U-0	U-0	R-0	R-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 RXACK：I^2C Master 接收到应答条件标志位
0 = 接收到应答条件
1 = 未接收到应答条件

bit 6 BUSY：I^2C Master 总线忙标志位
0 = 发送了 STOP 信号
1 = 发送了 START 信号

bit 5 AL：I^2C Master 总线仲裁失败标志位
0 = 总线仲裁成功
1 = 总线仲裁失败

bit 4-2 未实现：读为 0

bit 1 TIP：I^2C Master 传输状态标志位
0 = 数据传输完成
1 = 数据传输中

bit 0 IF：I^2C Master 中断标志位
0 = 数据正在传输或没有传输
1 = 8 位数据传输完成或仲裁失败

19. EFC 模块(类 EEPROM)

EFC 模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
FDR	0xB0FF	XRAM	EFC 编程数据寄存器	0000 0000
FADRL	0xB1FF	XRAM	EFC 编程地址寄存器低8位	0000 0000
FADRH	0xB2FF	XRAM	EFC 编程地址寄存器高8位	0000 0000
FCR	0xB3FF	XRAM	EFC 控制寄存器	0000 0000
FPSR	0xB4FF	XRAM	EFC 写保护序列寄存器	0000 0000

CBM73xx 系列芯片的 Flash 存储器包括 Main 区和 NVR 区两部分。Main 区就是芯片的程序存储器(ROM)，大小为 8K ~ 48K；NVR 区的大小为 1K。

Main 区和 NVR 区又分为若干 Sector 区，每个 Sector 区为 512Byte，也即，Main 区包含 16~96 个 Sector(48K)，NVR 区包含 2 个 Sector(1K)。

对于 sector 擦除和字节编程操作，Flash 存储器地址如表 1 所示。

表 1： CBM7332A6Q1 sector 擦除和字节编程 sector 与地址对应表

Main 区	NVR 区
Sector 0(0x0000~0x01FF)	Sector 0(0x0000~0x01FF)
Sector 1(0x0200~0x03FF)	Sector 1(0x0200~0x03FF)
……	
Sector 94(0xBC00~0xBDFF)	
Sector 95(0xBE00~0xBFFF)	

Flash 存储器在正常工作状态下(整个 VDD 范围内)是可读写的。这些存储器并不直接映射到寄存器文件空间，而是通过特殊功能寄存器对其进行间接寻址。共有 5 个特殊功能寄存器用于访问这些存储器：

- FDR
- FADRL
- FADRH
- FCR
- FPSR

19.1. sector 擦除

要写 Flash 存储器之前，必须要先进行擦除操作。EFC 模块只支持 sector 擦除，也就是每次擦除都会将 1 个 sector 区域整个的擦除。

对于 sector 擦除操作，FADRL 和 FADRH 寄存器组成一个 16 位 Flash 存储器地址。FCR 寄存器的 NVRSEL 位控制擦除 Main 区还是 NVR 区，ERASE 位用于控制启动 sector 擦除。只要设置 Flash 存储器地址落在目标 sector 内，启动 sector 擦除后，就会擦除目标 sector。将 ERASE 位置 1，即启动 sector 擦除，擦除完成后会自动清 0，所以需要查询 ERASE 位是否为 0，来判断擦除是否完成。

应注意的是，在写 FCR 寄存器时，必须先按顺序写 FPSR 等于 0x55、0xAA，

具体步骤可参考例 19-1。

例 19-1： sector 擦除

```
void eraseSector(unsigned char sel,unsigned short adr)          //sel=1->NVR  sel=0->Main
{
    FPSR = 0x55;
    FPSR = 0xAA;
    FADRL = adr & 0x00ff;  //adr 是编程地址
    FADRH = (adr & 0xff00) >> 8;
    FCR = 0x02 | (sel << 2);//sel 选择 Main 或 NVR
    while(0x02 == FCR_ERASE){;}//判断是否完成
}
```

19.2. 字节编程

EFC 功能支持对 Flash 存储器字节编程，用户应该先将编程字节的地址写入 FADRL 和 FADRH 寄存器，编程数据写入 FDR 寄存器，FCR 寄存器的 NVRSEL 位控制擦除 Main 区还是 NVR 区，PROG 位用于控制启动字节编程。将 PROG 位置 1，即启动字节编程，编程完成后会自动清 0，所以需要查询 ERASE 位是否为 0，来判断擦除是否完成。

具体步骤可参考例 19-2。

例 19-2： 字节编程

```
void writeByte(unsigned cahr sel,unsigned short adr,unsigned cahr dat)  //sel=1->NVR  sel=0->Main
{
    FPSR = 0x55;
    FPSR = 0xAA;
    FDR = dat;              //dat 是编程数据
    FADRL = adr & 0x00ff;   //adr 是编程地址
    FADRH = (adr & 0xff00) >> 8;
    FCR = 0x01 | (sel << 2);//sel 选择 Main 或 NVR
    while(0x01 == FCR_PORG){;}//判断是否完成
}
```

19.3. 字节读

读 Flash 存储器是通过 movc 指令实现的，对于字节读操作，Flash 存储器地址如表 2 所示。

具体步骤可参考例 19-3。

表 2： CBM7332A6Q1 字节读 sector 与地址对应表

Main 区	NVR 区
Sector 0(0x0000~0x01FF)	Sector 0(0xC000~0xC1FF)
Sector 1(0x0200~0x03FF)	Sector 1(0xC200~0xC3FF)
……	
Sector 94(0xBC00~0xBDFF)	
Sector 95(0xBE00~0xBFFF)	

例 19-3： 字节读

```
unsigned char readByte(unsigned short adr)
{
    unsigned char dat;
    dat = CBYTE[adr];
    return dat;
}
```

19.4. EFC 寄存器的定义

以下寄存器用于 EFC 操作。

寄存器 19-1： FDR：EFC 编程数据寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 FDR<7:0>：EFC 编程数据寄存器

EFC 编程数据寄存器，用于配置编程数据

寄存器 19-2： FADRL：EFC 编程地址寄存器低 8 位

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 FADRL<7:0>：EFC 编程地址寄存器低 8 位

与 FADRH 组成 16 位编程地址

寄存器 19-2： FADRH：EFC 编程地址寄存器高 8 位(续)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 FADRH<7:0>：EFC 编程地址寄存器高 8 位

与 FADRL 组成 16 位编程地址

注： 对于 Sector 擦除操作，只需要设置编程地址落在目标 sector 内即可。

寄存器 19-3： FCR：EFC 控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	NVRSEL	ERASE	PORG
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-3 未定义：读为 0

bit 2 NVRSEL：编程区域选择位

0 = 选择 Main 区

1 = 选择 NVR 区

bit 1 ERASE：编程区域选择位

当要启动 sector 擦除时，将此位置 1，擦除结束后会自动清 0

bit 0 PORG：编程区域选择位

当要启动字节编程时，将此位置 1，编程结束后会自动清 0

注： 写 FCR 寄存器之前，必须按顺序先写 FPSR 寄存器等于 0x55、0xAA。

寄存器 19-4： FPSR：EFC 写保护序列寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 FPSR<7:0>：写保护序列寄存器。

在操作 FCR 寄存器之前，需将 FPSR 写成 0xAA；

将 FPSR 写成 0xAA 的步骤是：

1、当 FPSR 等于 0x00 时，可以将 FPSR 写成 0x55；

2、当 FPSR 等于 0x55 时，可以将 FPSR 写成 0xAA；

3、当编程结束或擦除结束时，FPSR 被强制置为 0x00；

20. LED 驱动器

LED 模块相关寄存器列表

寄存器名称	寄存器地址	地址类型	说明	POR 复位值
LEDCR	0xC0FF	XRAM	LED 控制寄存器	0000 0010
LEDSETR	0xC1FF	XRAM	LED 参数配置寄存器	0000 0111
LEDDIVR	0xC2FF	XRAM	LED 时钟分频寄存器	0000 0010
GRIDVLDL	0xC3FF	XRAM	LED 有效位配置寄存器低8位	0000 0000
GRIDVLDH	0xC4FF	XRAM	LED 有效位配置寄存器高2位	0000 0000
LEDDR0	0x1100	XRAM	LED 数据寄存器0	0000 0000
LEDDR1	0x1101	XRAM	LED 数据寄存器1	0000 0000
LEDDR2	0x1102	XRAM	LED 数据寄存器2	0000 0000
LEDDR3	0x1103	XRAM	LED 数据寄存器3	0000 0000
LEDDR4	0x1104	XRAM	LED 数据寄存器4	0000 0000
LEDDR5	0x1105	XRAM	LED 数据寄存器5	0000 0000
LEDDR6	0x1106	XRAM	LED 数据寄存器6	0000 0000
LEDDR7	0x1107	XRAM	LED 数据寄存器7	0000 0000
LEDDR8	0x1108	XRAM	LED 数据寄存器8	0000 0000
LEDDR9	0x1109	XRAM	LED 数据寄存器9	0000 0000
LEDDR10	0x110A	XRAM	LED 数据寄存器10	0000 0000
LEDDR11	0x110B	XRAM	LED 数据寄存器11	0000 0000
LEDDR12	0x110C	XRAM	LED 数据寄存器12	0000 0000
LEDDR13	0x110D	XRAM	LED 数据寄存器13	0000 0000

LED 驱动器是 LED 显示控制模块，仅需要很少的软件操作，即可实现 LED 显示。支持多种 LED 数码管规格。配合 IO 电流能力寄存器(PnLDRVRH：PnLDRVRL)及本模块中的灰度等级选择位(LEDSETR 寄存器的 GRAY<2:0>位)，可实现多级灰度。

20.1. LED 的操作步骤

配置 LED 的操作步骤如下：

1. 端口配置：

通过 PxyIOCFG 寄存器将 I/O 配置成 LED GRID/SEG 功能时，会自动将引脚方向配置为输出。不需要配置对应的 PnOE 寄存器，选择 LED GRID/SEG 引脚方向为输出。(端口配置前应使能 IOC 模块时钟，即将 MDLCKCR 寄存器的 IOCEN 位置 1)

PnLDRVRH：PnLDRVRL 寄存器用于配置 IO 的驱动电流大小，与 LEDSETR 寄存器的 GRAY<2:0>位配合，可以满足不同的显示需求。

2. LED 模块时钟使能：

在操作 LED 模块相关寄存器前，必须先使能 LED 模块时钟，即将 MDLCKCR 寄存器的 LEDEN 位置 1。

3. LED 模块参数配置：

- 数码管的类型选择：

LED 驱动器可以驱动不同类型的 LED，LEDSETR 寄存器的 TYPE 位可以选择所驱动的数码管是共阴极还是共阳极，MODE<2:0>位可以配置段位数，有 6 种类型可以选择：

- 4 位 13 段
- 5 位 12 段
- 6 位 11 段
- 7 位 10 段
- 9 位 8 段
- 10 位 7 段

- 显示的灰度等级设置：

LEDSETR 寄存器的 GRAY<2:0>位可以配置 LED 显示的灰度等级，最多支持 8 级；值越大，显示越亮。(与 IO 的驱动电流配置寄存器配合后可以满足不同的显示需求)

- 刷新率配置：

LED 模块的时钟是内部 OSC800K 时钟经过 CKDIVCR 寄存器的 SLOWDIV<1:0>位分频后的时钟。

LED 显示的刷新率由 LEDDIVR 寄存器控制,LEDDIVR 寄存器可以配置 LED 时钟的分频数，分频数越大，LED 时钟越慢，LED 刷新率越低(闪烁明显)，反之，刷新率越高(闪烁不明显)。

计算刷新频率：

刷新频率是指 1s 时间内，某位某段点亮的次数。

设显示模式为 M 位 N 段，但实际接的数码管为 Y(1<=Y<=M)位。LED 时钟分频数为 DIV1(LEDDIVR 寄存器配置)；LED 时钟来自于内部 OSC800K 经过 CKDIVCR 分频后的时钟，设分频数为 DIV2(SLOWDIV<1:0>配置)。

则 Y 位数码管依次被点亮的总时间为：

$T = Tosc800k \times (DIV1 + 1) \times 2^{DIV2} \times 8 \times N \times Y$;

因为控制方式为逐位扫描，所以，在 T 时间内，某位某段只被点亮了一次。那么，刷新率计算公式为：

$F = 1/T$

$= 1/((Tosc800k \times (DIV1 + 1) \times 2^{DIV2} \times 8 \times N \times Y)$

以一组实际数据计算为例,M=7,N=10,Y=5,DIV1=3,DIV2=3

则刷新率：

$F = 1s / (1.25us \times 4 \times 2^3 \times 8 \times 10 \times 5)$

$= 62.5Hz$。

实验表明:刷新频率为 60Hz 时闪烁明显,达到 70Hz 时可基本消除闪烁。

- 数码管的有效位选择：

GRIDVLDL 和 GRIDVLDH 寄存器组成 10bit 的 GRIDVLD，用于配置 LED 数码管的有效位。

LED 模块支持最多 10 位数码管，实际使用中，若不足 10 位，可配置 GRIDVLD，将使用到的位置 1，未使用的位清 0。可提高显示刷新率。

举例：若实际使用 6 位数码管，使用的是第 1，2，3，5，6，10 位，则可将 GRIDVLD 设置成 0x0237 (0010_0011_0111)。

4. 显示数据配置：

根据 LEDSETR 寄存器 MODE<2:0>位的配置，LEDDRn 寄存器的用法有所差异。

1)当配置成 4 位 13 段、5 位 12 段、6 位 11 段、7 位 10 段时：

LEDR1 和 LEDR0 组成 16 位数据，取低 n 位(n=10~13)作为数码管第 1 位的数据；LEDR3 和 LEDR2 组成 16 位数据，取低 n 位(n=10~13)作为数码管第 2 位的数据；以此类推。

2）当配置成 9 位 8 段时：

LEDRn(n=1~9)作为 9 位数码管的数据。

3）当配置成 10 位 7 段时：

LEDRn(n=1~10)作为 10 位数码管的数据，LEDRn 取低 7 位。

根据实际的硬件接法，给数码管配置合适的数据。数据的 1 个 bit 对应于 1 位数码管的 1 个段；不管共阴还是共阳的数码管，bit 位写 1 亮，写 0 灭。

> 注： 当 LED 显示使能时，是不能修该显示数据的。要修改显示数据必须先关闭显示使能,查询 LEDCR 寄存器的 WREN 位为 1 后，才能修改显示数据。

5. 显示使能：

LEDCR 寄存器的 ONOFF 位置 1，即使能 LED 显示，清 0，即关闭 LED 显示。需注意的是 ONOFF 位置 1 后，等到 WREN 位被清 0 时，才表示显示真正使能；ONOFF 位清 0，等到 WREN 位被置 1 时，才表示显示已经关闭。

注：当需要更改显示数据时，必须先关闭 LED 显示，查询 LEDCR 寄存器的 WREN 位为 1 后，才能配置新的显示数据，然后再使能 LED 显示。

20.2. 数码管的接法

根据数码管不同的段位配置，数码管有两种接法：

1）当配置成 4 位 13 段、5 位 12 段、6 位 11 段、7 位 10 段时，IO 复用为 GRIDn 的引脚接数码管的位引脚，复用为 SEGn 的引脚接数码管段引脚。

2）当配置成 9 位 8 段、10 位 7 段时，IO 复用为 GRIDn 的引脚接数码管的段引脚，复用为 SEGn 的引脚接数码管位引脚。

20.3. LED 显示控制原理

以 3 位 8 段共阳极数码管为例，LED 显示控制时序如图 20-1 所示。

LED 模块使用的是扫描的方式驱动数码管：

1）共阳极的数码管，所以 3 个 GRID 位逐个变高；

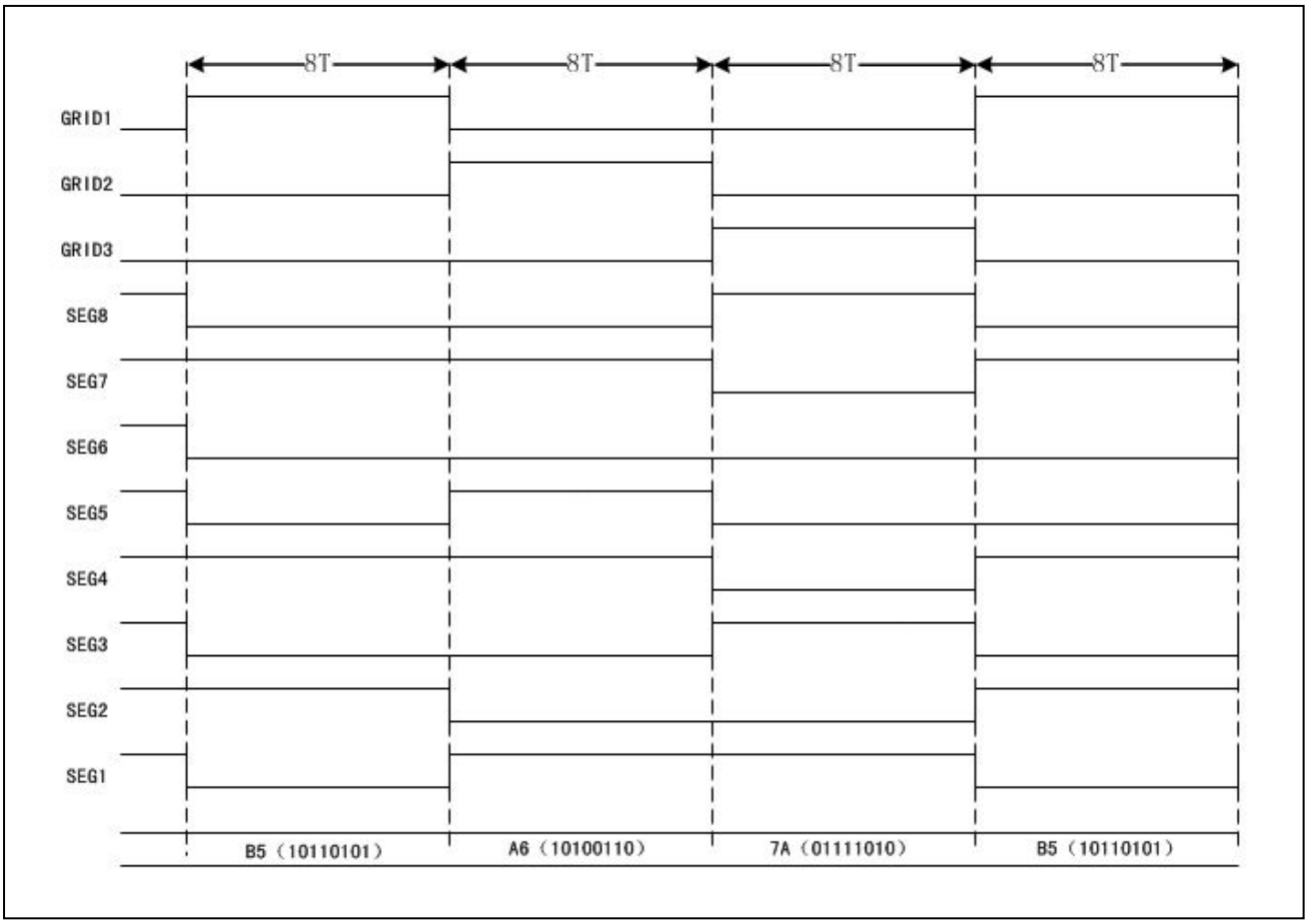
2）8 段的数码管，所以每个 GRID 位为高的时间持续 8 个时间单位(若为 7 段则持续 7 个时间单位)。

3）8 个 SEG 位分别对应于所配置数据位。在这 8 个时间单位，8 个 SEG 位变化为数据位所对应的电平，每个 SEG 位的电平持续时间为 8 个时间单位。即，在同一时刻，最多有 8 个段被点亮。

4）一个时间单位等于 8 个 LED 时钟周期，LED 时钟周期由 CKDIVCR、LEDDIVR 寄存器的配置决定。所以 LED 时钟周期改变时，刷新率会变化；当有效位数增多时，刷新率也会随之改变。

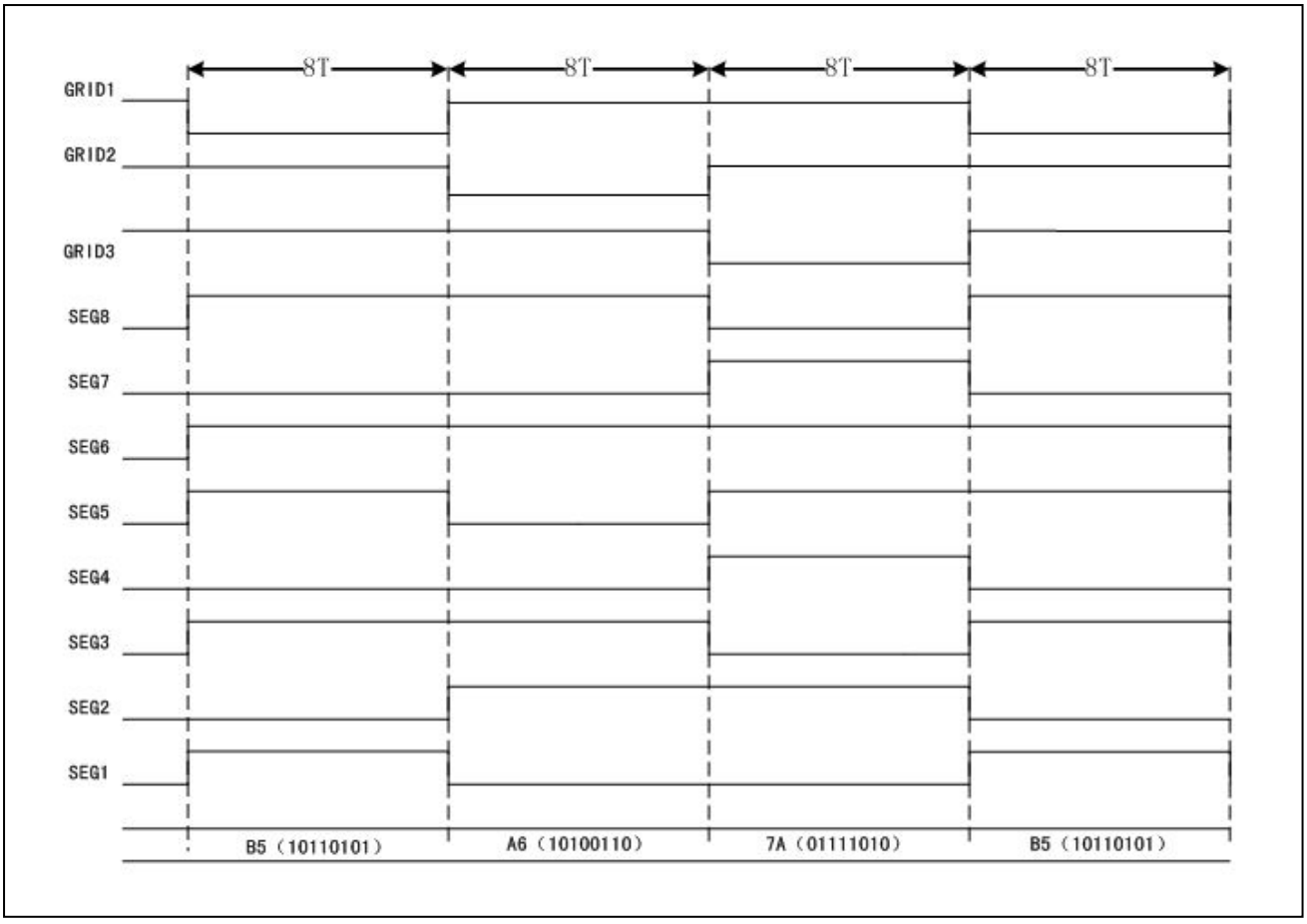
共阴极数码管的时序则完全相反，如图 20-2 所示。

图 20-1： LED 显示控制时序图(共阳极)



注： 图中因为是共阳极，GRID 为高，SEG=0 才点亮，所以当数据位为 1 时，对应 SEG 为 0。

图 20-2： LED 显示控制时序图(共阴极)



注： 图中因为是共阴极，GRID 为低，SEG=1 才点亮，所以当数据位为 1 时，对应 SEG 为 1。

20.4. LED 寄存器的定义

以下寄存器用于 LED 操作。

寄存器 20-1： LEDCR：LED 控制寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	—	WREN	ONOFF
U-0	U-0	U-0	U-0	U-0	U-0	R-1	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-2 未定义：读为 0

bit 1 WREN：可写 LED 数据寄存器标志位
只读寄存器。改变 LED 显示信息前，需先写 ONOFF 使 LED 显示灭，然后查询等待 WREN 为 1 时(当使能 LED 显示时，WREN 会变低)，才可以更改 LED 显示信息；

bit 0 ONOFF：LED 显示使能位
0 = 关闭 LED 显示
1 = 使能 LED 显示

寄存器 20-2： LEDDIVR：LED 时钟分频寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—			
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-1	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-3 未定义：读为 0

bit 2-0 LEDDIVR<2:0>：LED 时钟分频寄存器
000 = 1 分频
001 = 2 分频
010 = 3 分频
011 = 4 分频
100 = 5 分频
101 = 6 分频
110 = 7 分频
111 = 8 分频

寄存器 20-3： LEDSETR：LED 参数配置寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TYPE	MODE2	MODE1	MODE0	—	GRAY2	GRAY1	GRAY0
R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 TYPE：LED 类型选择位
0 = 共阴极
1 = 共阳极

bit 6-4 MODE<2:0>：LED 模式选择位
000 = 4 位 13 段；SEG 脚作为段，GRID 脚作为位
001 = 5 位 12 段；SEG 脚作为段，GRID 脚作为位
010 = 6 位 11 段；SEG 脚作为段，GRID 脚作为位
011 = 7 位 10 段；SEG 脚作为段，GRID 脚作为位
100 = 9 位 8 段；GRID 脚作为段，SEG 脚作为位
101 = 9 位 8 段；GRID 脚作为段，SEG 脚作为位
110 = 10 位 7 段；GRID 脚作为段，SEG 脚作为位
111 = 10 位 7 段；GRID 脚作为段，SEG 脚作为位

bit 3 未定义：读为 0

bit 2-0 GRAY<2:0>：LED 灰度等级选择位
000 = 时间单位的 12.5%
001 = 时间单位的 25%
010 = 时间单位的 37.5%
011 = 时间单位的 50%
100 = 时间单位的 62.5%
101 = 时间单位的 75%
110 = 时间单位的 87.5%
111 = 时间单位的 100%

寄存器 20-4： GRIDVLDL：LED 有效位配置寄存器低 8 位

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 GRIDVLDL<7:0>：LED 有效位配置寄存器低 8 位
GRIDVLDH 组成 10bit 的 LED 有效位配置寄存器 GRIDVLD

寄存器 20-4： GRIDVLDH：LED 有效位配置寄存器高 2 位(续)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	—	—		
U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-2 未定义：读为 0
bit 1-0 GRIDVLDH<1:0>：LED 有效位配置寄存器高 2 位
GRIDVLDL 组成 10bit 的 LED 有效位配置寄存器 GRIDVLD

寄存器 20-5： LEDDRn：LED 数据寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 LEDDRn<7:0>：LED 数据寄存器
LEDDRn(n = 1、2、3、4、5、……、10、11、12、13)是 LED 数据寄存器，根据 LEDSETR 寄存器 MODE 位的配置，LEDDRn 的用法有所区别，详细介绍请查看 20.1.6 小节“显示数据”。

21. 电气特性

21.1. 极限参数

电源供应电压.....VSS-0.3V~VSS+6.0V
端口输入电压.....VSS-0.3V~VDD+0.3V
存储温度.....-50℃~125℃
工作温度.....-40℃~85℃
VSS 引脚的最大输出电流.....90mA
VDD 引脚的最大输入电流.....90mA
任一 I/O 引脚的最大输出灌电流.....24mA
任一 I/O 引脚的最大输出拉电流.....24mA
所有端口(组合)的最大灌电流.....85mA
所有端口(组合)的最大拉电流.....85mA
总功耗.....800mW

注：如果芯片的工作条件超过极限参数所规定的范围，就可能对芯片造成永久性的损坏。上述值仅为运行条件极大值，我们建议不要使芯片在规定的范围以外运行。芯片长时间工作在最大值条件下，其稳定性会受到影响。

21.2. 直流电气特性

温度（25℃）

符号	参数	测试条件		最小	典型	最大	单位
		VDD	条件				
VDD	工作电压	—	低压复位不使能，POR 使能	1.6	—	5.5	V
			低压复位使能	2.7	—	5.5	V
IDD	工作电流（正常模式）	3V	无负载，Fsys = 8MHz，WDT 使能，Touch Key off，LED off，LVD 使能	—			mA
		5V		—			mA
		3V	无负载，Fsys = 16MHz，WDT 使能，Touch Key off，LED off，LVD 使能	—			mA
		5V		—			mA
	工作电流（空闲模式）	3V	无负载，Fsys = 8MHz，WDT 关闭，Touch Key off，LED off，LVD 使能	—			mA
		5V		—			mA
		3V	无负载，Fsys = 16MHz，WDT 关闭，Touch Key off，LED off，LVD 使能	—			mA
		5V		—			mA
	工作电流（休眠模式）	3V	无负载，Fsys 关闭，WDT 关闭，Touch Key off，LED off，LVD 使能	—			mA
		5V		—			mA
		3V	无负载，Fsys 关闭，WDT 关闭，Touch Key off，LED off，LVD 使能	—			mA
		5V		—			mA
VIL	I/O 口或输入引脚的低电平输入电压	5V	—	0	—	1.5	V
		—		0	—	0.3VDD	V
VIH	I/O 口或输入引脚的高电平输入电压	5V	—	3.5	—	5.0V	V
		—		0.7VDD	—	VDD	V
VPOR	上电复位电压	—	POR 使能	-5%		-5%	V
VLVD	低压复位电压	—	低压复位使能，VLVD = 2.7V	-5%		-5%	V
			低压复位使能，VLVD = 2.9V	-5%		-5%	V
			低压复位使能，VLVD = 3.1V	-5%		-5%	V
			低压复位使能，VLVD = 3.3V	-5%		-5%	V
IoL	I/O 口灌电流	3V	VoL = 0.1VDD			—	mA
		5V	VoL = 0.1VDD			—	mA
IoH	I/O 口拉电流	3V	VoH = 0.9VDD，PnLDRVR_x＝00			—	mA
		5V				—	mA
		3V	VoH = 0.9VDD，PnLDRVR_x＝01			—	mA
		5V				—	mA
		3V	VoH = 0.9VDD，PnLDRVR_x＝10			—	mA
		5V				—	mA
RPH	I/O 口上拉电阻	3V	—				kΩ
		5V	—				kΩ

21.3. 交流电气特性

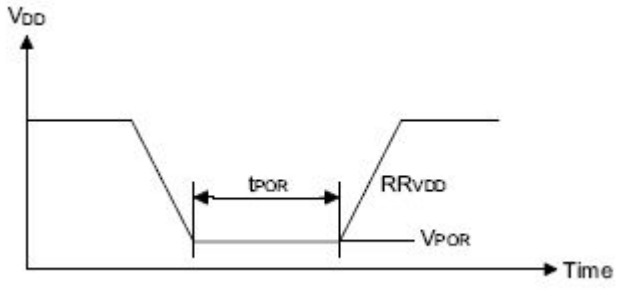
温度（25℃）

符号	参数	测试条件		最小	典型	最大	单位
		VDD	条件				
FOSC	振荡时钟	3V/5V	25℃	-2%	48M	2%	MHz
FLSC	800K 时钟	3V/5V	25℃	-2%	800	2%	KHz
TTIMER	定时器输入脉宽	—	—		—	—	us
TEINT	外部中断脉宽	—	—		—	—	us
TPOR	上电复位脉宽	—	—				us
TLVD	低压复位脉宽	—	—				us
TEFCES	EFC sector 擦除周期	5V	—				ms
TEFCRD	EFC 读周期	5V	—				Tsys
TEFCWR	EFC 写周期	5V	—				ms
TRSTD	系统复位延迟时间（POR 复位,LVD 复位）	—	—				ms
	系统复位延迟时间（WDT 复位）	—	—				ms

21.4. 上电复位特性

温度（25℃）

符号	参数	测试条件		最小	典型	最大	单位
		VDD	条件				
VPOR	上电复位电压	—	—	—	—		mV
RRVDD	上电复位电压速率	—	—		—	—	V/ms
TPOR	VDD 保持为 VPOR 的最小时间	—	—		—	—	ms



21.5. 触控按键电气特性

温度（25℃）

符号	参数	测试条件		最小	典型	最大	单位
		VDD	条件				

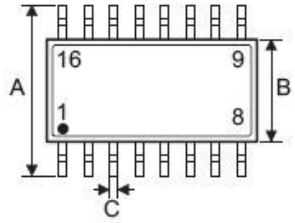
21.6. A/D 转换器电气特性

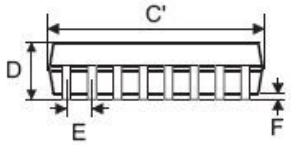
温度（25℃）

符号	参数	测试条件		最小	典型	最大	单位
		VDD	条件				
AVDD	A/D 转换器工作电压	—	—	2.7	—	5.5	V
VADI	A/D 转换器输入电压	—	—	0	—	VREF	V
VREF	A/D 转换器外部参考电压	—	—		—	AVDD	V
VBG	BG 参考电压	—	—	-3%	1.00	3%	V
DNL	非线性微分误差	3V			—		LSB
		5V					
		3V			—		LSB
		5V					
INL	非线性积分误差	3V			—		LSB
		5V					
		3V			—		LSB
		5V					
IADC	打开 A/D 增加的功耗	3V	无负载（TADCK=）	—			mA
		5V	无负载（TADCK=）	—			mA
IBG	使用 VBG 参考电压增加的功耗	—	—	—			uA
TADCK	A/D 转化器时钟周期	—	—		—		us
TADC	A/D 转换时间(包括采样和保持时间)	—	10 bit ADC	—		—	TADCK
TADS	A/D 转换器采样时间	—	—	—		—	TADCK
TON2ST	A/D 转换器 On-to-Start 时间	—	—		—	—	us
TBG	开启 VBG 稳定时间	—	—	—	—		us

22. 封装信息

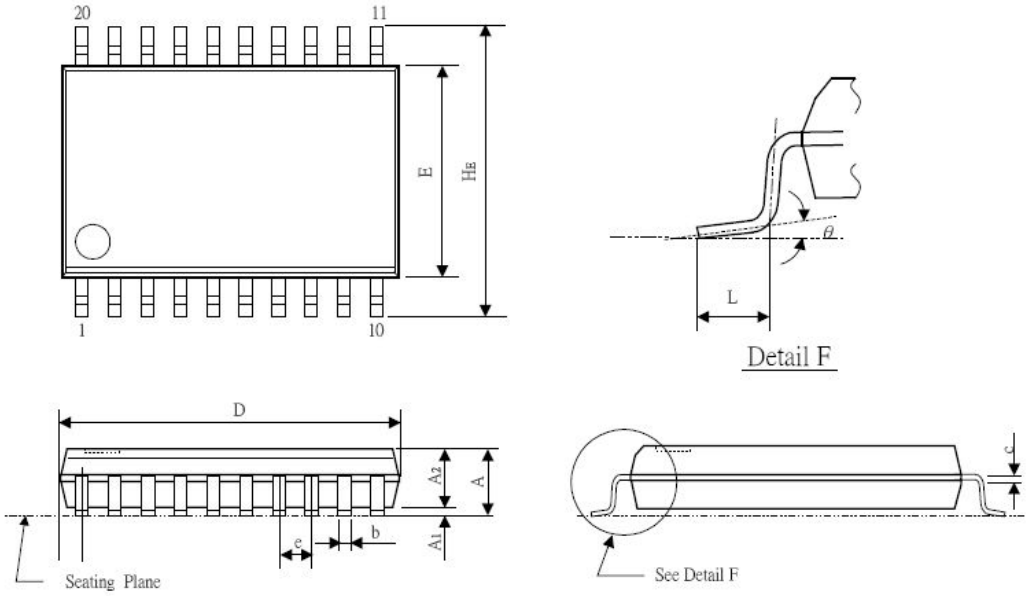
(1)SOP16 外形尺寸





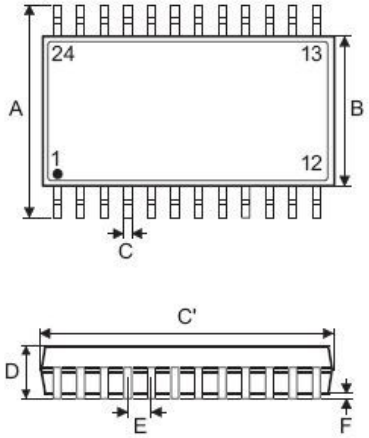
符号	说明	尺寸（mm）		
		最小	正常	最大
A	跨度	5.81	---	6.24
B	塑胶跨度	3.81	---	4.04
C	脚宽度	0.36	---	0.46
C’	塑胶体长	9.91	---	10.10
D	总高	1.40	---	1.70
E	脚间距	1.27		
F	站高	0.05	---	0.18
G	脚长	0.40	---	0.70
H	脚厚度	0.2		
a	脚角度	---	0	8

(2)SOP20 外形尺寸



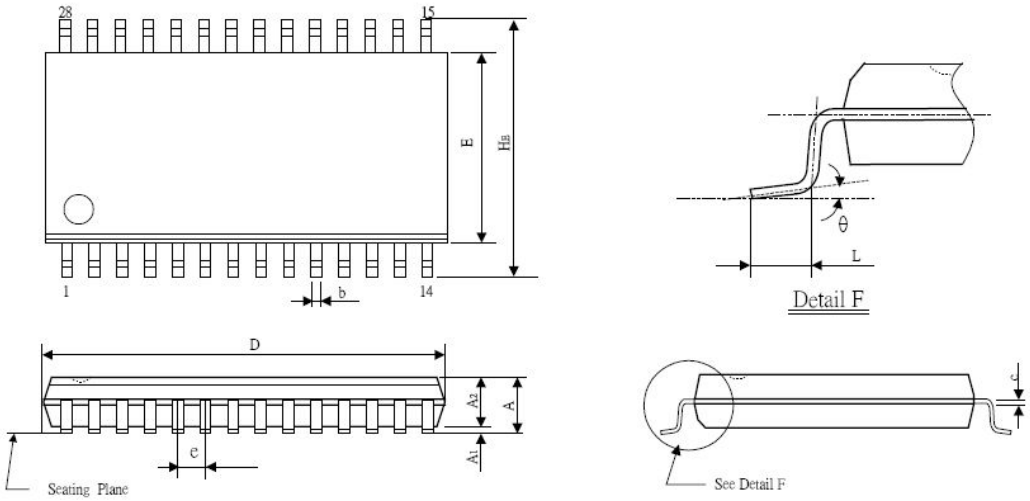
Symbol	Dimensions in inches		Dimensions in mm	
	Min	Max	Min	Max
A	0.093	0.104	2.35	2.65
A1	0.004	0.012	0.10	0.30
A2	0.083	0.098	2.10	2.50
b	0.013	0.020	0.33	0.51
c	0.008	0.013	0.20	0.33
D	0.493	0.512	12.52	13.00
E	0.291	0.299	7.40	7.60
e	0.050(BSC)		1.27(BSC)	
H_E	0.398	0.418	10.11	10.61
L	0.016	0.050	0.40	1.27
θ	0°	8°	0°	8°

(3)SOP24 外形尺寸



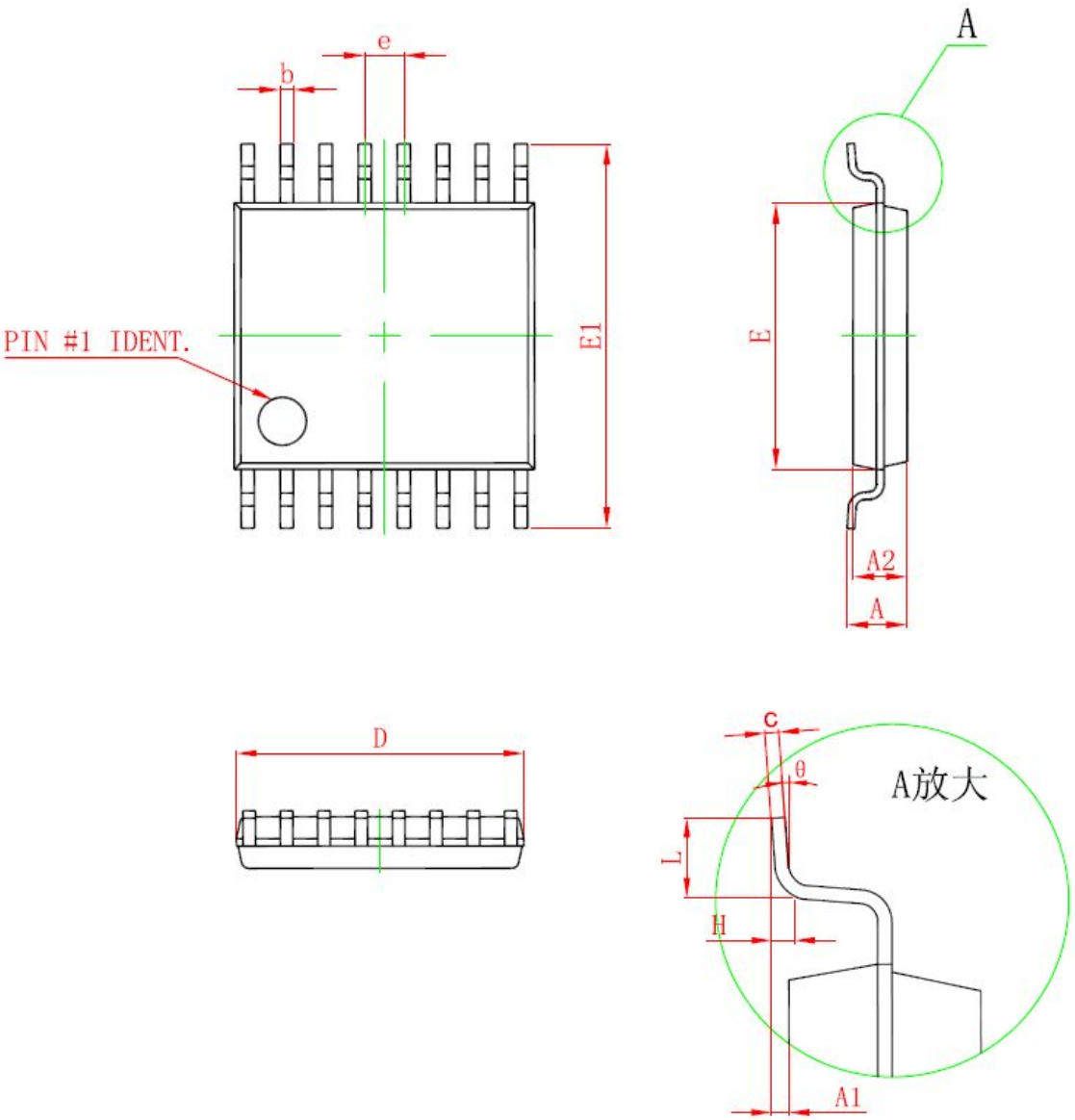
符号	说明	尺寸（mm）		
		最小	正常	最大
A	跨度	9.90	---	10.50
B	塑胶跨度	7.42	---	7.62
C	脚宽度	---	0.406	---
C’	塑胶体长	15.28	---	15.48
D	总高	2.23	---	2.58
E	脚间距	1.27		
F	站高	0.10	---	0.25
G	脚长	0.70	---	0.90
H	脚厚度	0.254		
a	脚角度	---	0	8

(4)SOP 28 外形尺寸



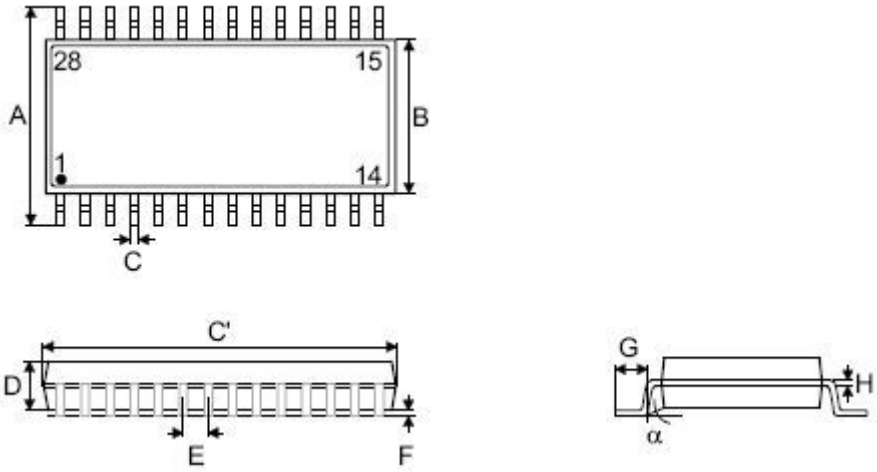
Symbol	Dimensions in inches		Dimensions in mm	
	Min	Max	Min	Max
A	0.085	0.104	2.15	2.65
A1	0.004	0.012	0.10	0.30
A2	0.081	0.098	2.05	2.50
b	0.013	0.02	0.33	0.51
c	0.008	0.014	0.20	0.36
D	0.697	0.713	17.70	18.10
E	0.291	0.3	7.40	7.62
e	0.050(BSC)		1.27(BSC)	
H_E	0.402	0.418	10.21	10.61
L	0.016	0.05	0.40	1.27
θ	0°	8°	0°	8°

（5）TSSOP16 外形尺寸



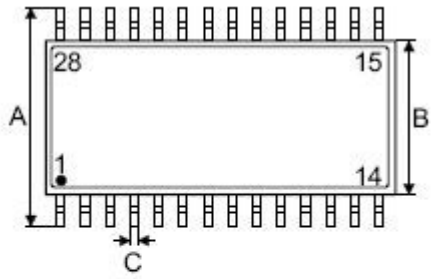
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
D	4.900	5.100	0.193	0.201
E	4.300	4.500	0.169	0.177
b	0.190	0.300	0.007	0.012
c	0.090	0.200	0.004	0.008
E1	6.250	6.550	0.246	0.258
A		1.100		0.043
A2	0.800	1.000	0.031	0.039
A1	0.020	0.150	0.001	0.006
e	0.65 (BSC)		0.026 (BSC)	
L	0.500	0.700	0.020	0.028
H	0.25 (TYP)		0.01 (TYP)	
θ	1°	7°	1°	7°

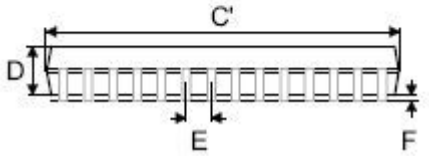
(6)TSSOP28 外形尺寸

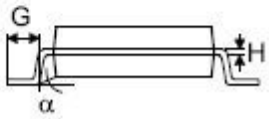


符号	说明	尺寸（mm）		
		最小	正常	最大
A	跨度	6.25	---	6.55
B	塑胶跨度	4.3	---	4.5
C	脚宽度	0.19	---	0.3
C’	塑胶体长	9.6	---	9.8
D	总高	---	1.2	---
E	脚间距	0.65		
F	站高	0.05	---	0.15
G	脚长	---	1	---
H	脚厚度	0.09	---	0.2
a	脚角度	---	0	8

(6)SSOP28 外形尺寸

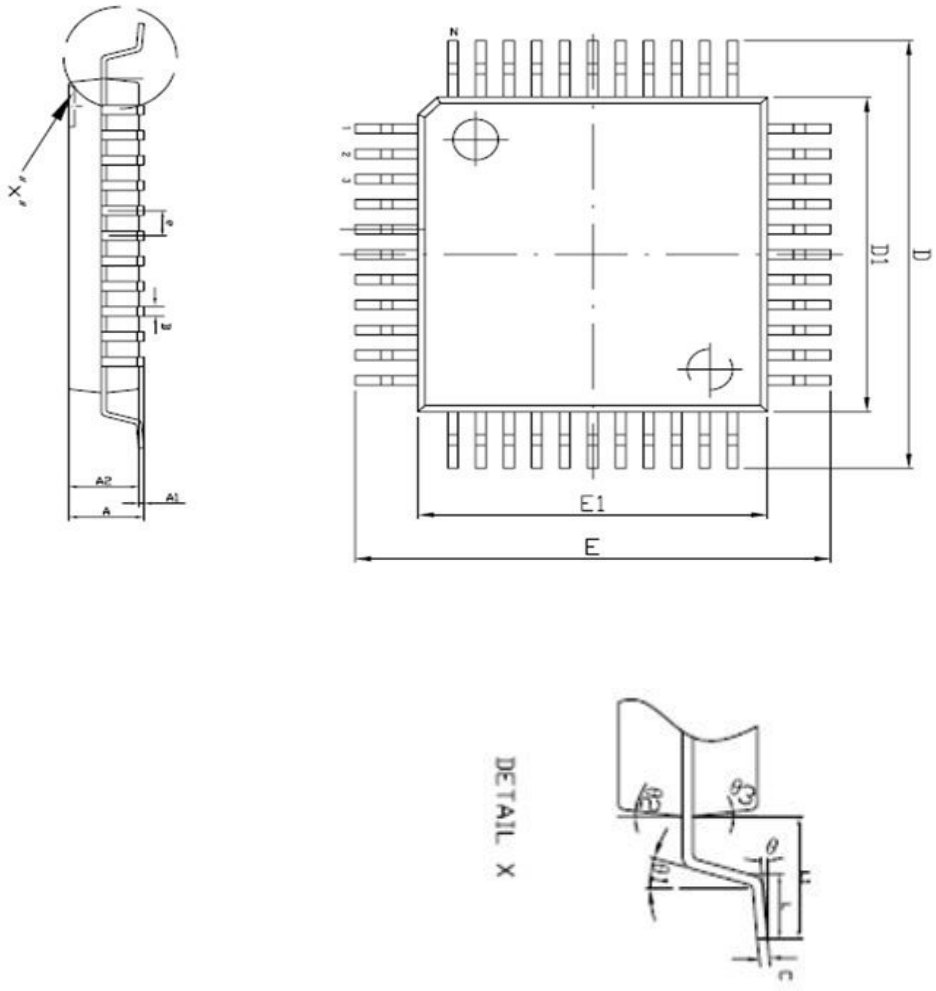






符号	说明	尺寸（mm）		
		最小	正常	最大
A	跨度	7.6	7.8	8.0
B	塑胶跨度	5.1	5.3	5.5
C	脚宽度	0.29	---	0.33
C’	塑胶体长	10	10.2	10.4
D	总高	---	---	2.0
E	脚间距	0.65		
F	站高	0.15	---	0.2
G	脚长	0.55	0.75	0.95
H	脚厚度	0.15	---	0.2
a	脚角度	---	0	8

(7)QFP44 外形尺寸



符号	说明	尺寸		
		最小	正常	最大
A	总高	1.95	---	2.30
A1	站高	0.05	---	0.20
B	脚宽度	0.30	---	0.38
C	脚厚度	0.11	---	0.23
D	跨度	13.20	---	14.00
D1	塑胶体长	9.90	---	10.10
E	跨度	13.20	---	14.00
E1	塑胶跨度	9.90	---	10.10
e	脚间距	0.80		
L	脚长	0.60	---	1.00
a	脚角度	---	0	7

23. 最小起订量

芯片型号	封装	最小包装量	订单要求	备注
CBM7332A6Q1	QPF44	96pcs/盘	1248pcs 起订，按 96 的倍数下订单。	
CBM7320A4S1	SOP28(SOIC28)	27pcs/管	486pcs 起订，按照 27 的倍数下订单。	
CBM7320A4P1	SSOP28	46pcs/管	506pcs 起订，按照 46 的倍数下订单。	
CBM7312A3S1	SOP20(SOIC20)	38pcs/管	494pcs 起订，按照 38 的倍数下订单。	
CBM7308A3S1	SOP16(SOIC16)	50pcs/管	500pcs 起订，按照 50 的倍数下订单。	