芯海科技
CHIPSEA
股票代码:688595

CS8M32X 系列 8 位 MCU

用户手册

V1.0 版本

涉密等级：公开

芯海科技（深圳）股份有限公司
www.chipsea.com
sales@chipsea.com
+86-0755-8616 9257
518000

版本历史

版本	修改内容	日期
V1.0	首次正式发布	2023-12-19

目 录

文档缩写

- 可读可写（r/w）
- 只读（r）.
- 只写(w).
- 可读/写 1 清零（r/w1c）
- 可读/写 0 清零（r/w0c）
- 可读/读清零（r/rc）
- 可读/写 1 置 1（r/w1s）
- 保留(Res)

适用产品

产品系列	产品型号	CVC	Flash	封装
CS8M320	CS8M320L3M6Nx	Y	8KW	SOP16
	CS8M320F3V6Nx	Y	8KW	QFN20(3*3*0.75, e=0.4)
CS8M322	CS8M322B2E6Nx	Y	4KW	MSOP10
	CS8M322L2U6Nx	Y	4KW	QFN16(3*3*0.75, e=0.5)
	CS8M322F2V6Nx	Y	4KW	QFN20(3*3*0.75, e=0.4)

注：若表格中无对应的型号，请查看最新版的数据手册或咨询芯海科技技术支持人员：x=U、R 或 T，代表不同的包装，具体请参考对应的数据手册。

1. 标准功能

1.1. CPU 核

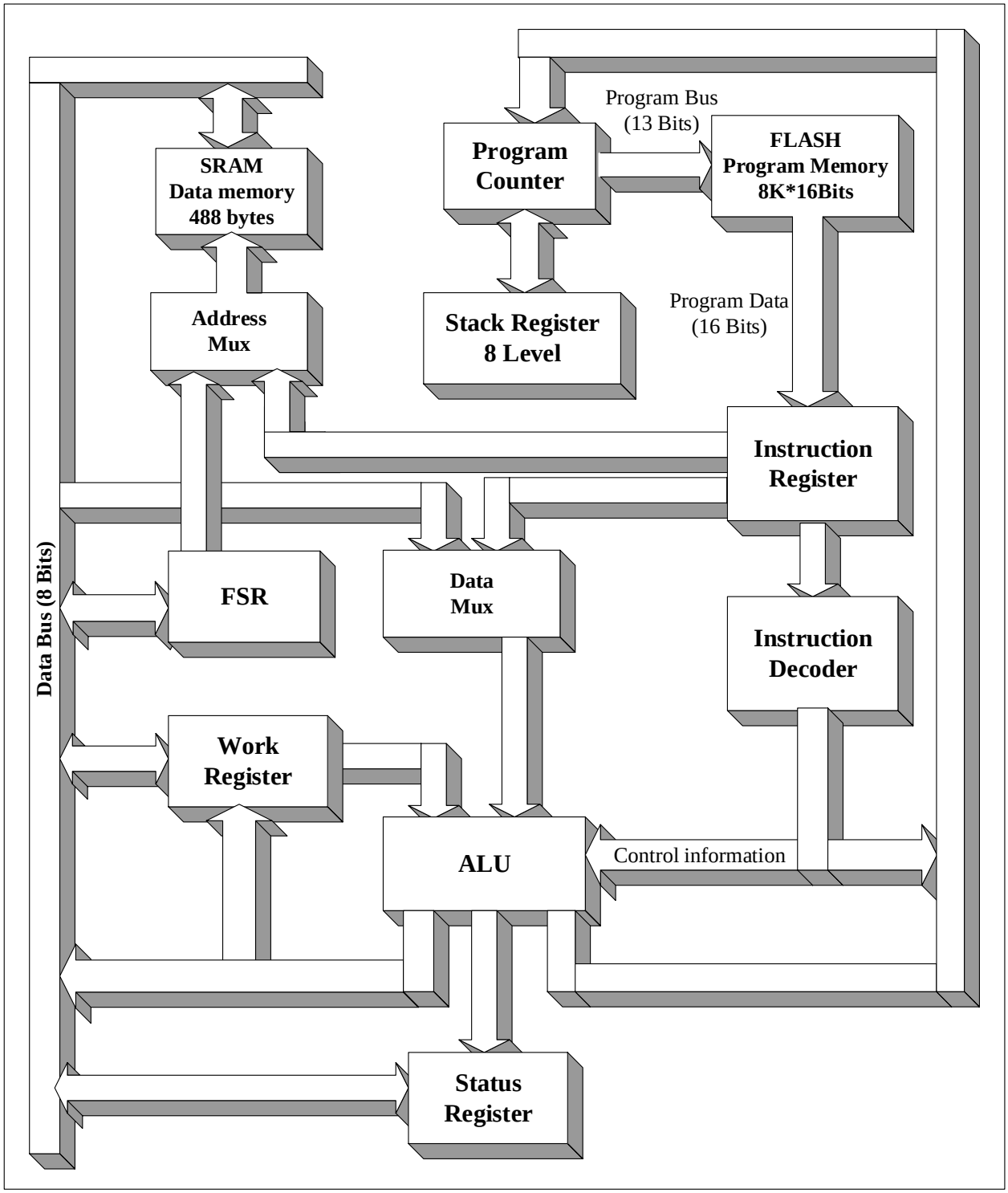


图 1　CS8M32X CPU 核的功能模块图

从 CPU 核的功能模块图中，可以看到它主要包含 7 个主要寄存器及 2 个存储器单元。

表 1　MCU 架构说明

模块名称	描述
程序计数器	此寄存器在 CPU 的工作周期间起到很重要的作用，它记录 CPU 每个周期处理程序存储器中指令的指针。在一个 CPU 周期中，程序计数器将程序存储器地址（13bits），指令指针推送到程序存储器，然后自动加 1 以进行下一次周期。
堆栈寄存器	堆栈寄存器是用来记录程序返回的指令指针。当程序调用函数，程序计数器会将指令指针推送到堆栈寄存器。在函数执行结束之后，堆栈寄存器会将指令指针送回到程序计数器以继续原来的程序处理。
指令寄存器	程序计数器将指令指针（程序存储器地址）推送到程序存储器，程序存储器将程序存储器的数据（16bits）推送到指令寄存器 。 CS8M32X 的指令是 16bits，包括 3 种信息：直接地址，立即数及控制信息。 直接地址（8bits） ：数据存储器的地址。CPU 能利用此地址来对数据存储器进行操作。 立即数（8bits） ：CPU 通过 ALU 利用此数据对工作寄存器进行操作。 控制信息 ：它记录着 ALU 的操作信息。
指令译码器	指令寄存器将控制信息推送到指令译码器以进行译码，然后译码器将译码后的信息发送到相关的寄存器。
算术逻辑单元	算术逻辑单元不仅能完成 8 位二进制的加，减，加 1，减 1 等算术计算，还能对 8 位变量进行逻辑的与，或，异或，循环移位，求补，清零等逻辑运算。
工作寄存器	工作寄存器用来缓存数据存储器的数据和立即数。
状态寄存器	当 CPU 利用 ALU 处理寄存器数据时，如下的状态会随着如下顺序变化：PD，TO，DC，C 及 Z。
文件选择寄存器	在 CS8M32X 的指令集中，FSR 是用于间接数据处理（即实现间接寻址）。用户可以利用 FSR 来存放数据存储器中的某个寄存器地址，然后通过间接地址寄存器（IND）对这个寄存器进行处理。
程序存储器	CS8M32X 内带 8K×16 位的 flash 作为程序存储器。由于指令的操作码（OPCODE）是 16bits，用户最多只能编程 8K 的指令。程序存储器的地址总线是 13bits，数据总线是 16bits。
数据存储器	CS8M32X 内带 488 bytes 的 SRAM 作为数据存储器。此数据存储器的地址总线是 9 bits，数据总线是 8 bits。其中 SRAM 最后 24 个地址为堆栈寄存器，因此不能对 SRAM 的最后 48 个地址进行写操作。

1.1.1. 存储器

1.1.1.1. 程序存储器

程序存储器主要用于指令的存储，在CS8M320/M322中，该程序存储器是8K/4K*16bit的程序FLASH，对于程序员来说，该存储器可读。系统的 reset 地址为 000H，中断入口地址为 004H，需要注意的是所有的中断共用同一个中断入口地址。

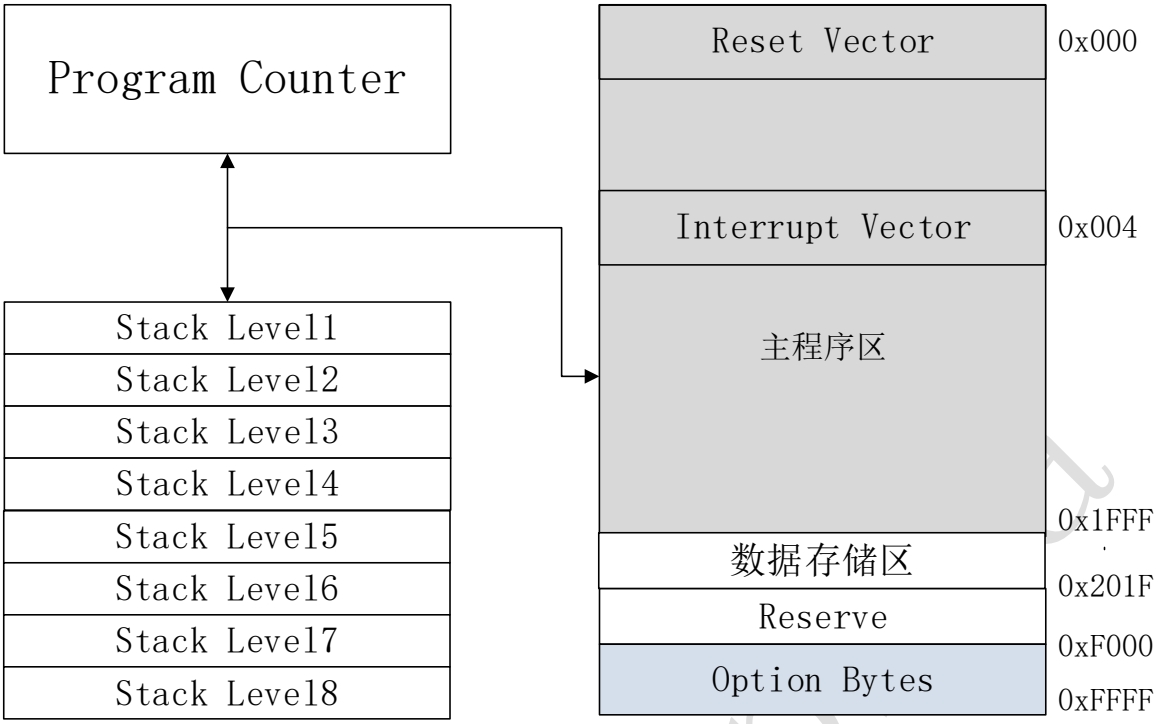


图 2　M320 程序存储器

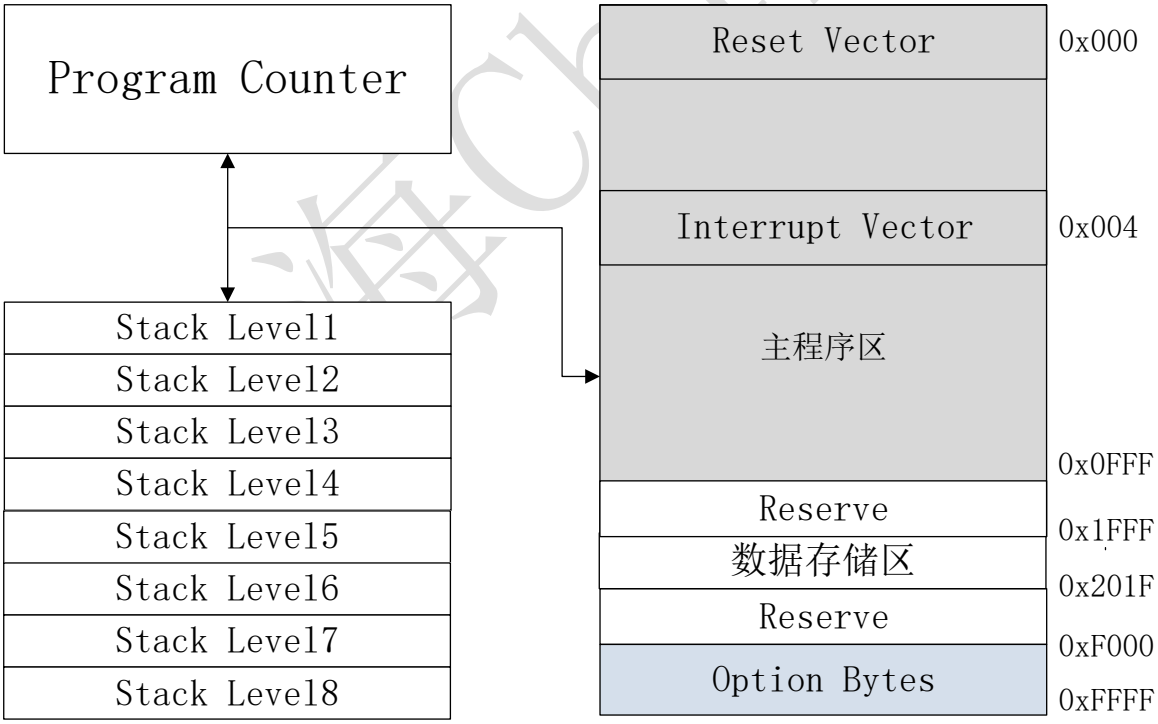


图 3　M322 程序存储器

1.1.1.2.　数据存储器

数据存储器主要用于存储程序运行过程中的全局变量以及中间变量。该存储器分为三个部分。地址的 00H 至 08H 是系统特殊功能寄存器，例如间接地址，间接地址指针，状态寄存器，工作寄存器，中断标志位，中断控制寄存器。地址 09H 至 7FH 对应外设特殊功能寄存器，例如 IO 端口，定时器。系统特殊功能

寄存器和外设特殊功能寄存器是用寄存器实现，而通用数据存储器是 RAM 实现，可以读出也可以写入。

BSR 选择寄存器（地址为 08H）

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06H	BSR	IRP0	IRP1						PAGE[0]	00uuuuu0

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0
BSR	IRP0	IRP1						PAGE[0]

位地址	标识符	功能
7	IRP0	IND0 间接页寻址位 0 = 间接寻址 IND0 时，访问前 256byte 地址 1 = 间接寻址 IND0 时，访问后 256byte 地址
6	IRP0	IND1 间接页寻址位 0 = 间接寻址 IND1 时，访问前 256byte 地址 1 = 间接寻址 IND1 时，访问后 256byte 地址
5:1	RESERVE	保留
0	PAGE[0]	PAGE 页选择 0 = 直接寻址和间接寻址访问 PAGE0 1 = 直接寻址和间接寻址访问 PAGE1

注：488 byte SRAM 的空间占用 PAGE0 全部和 PAGE1 的一部分

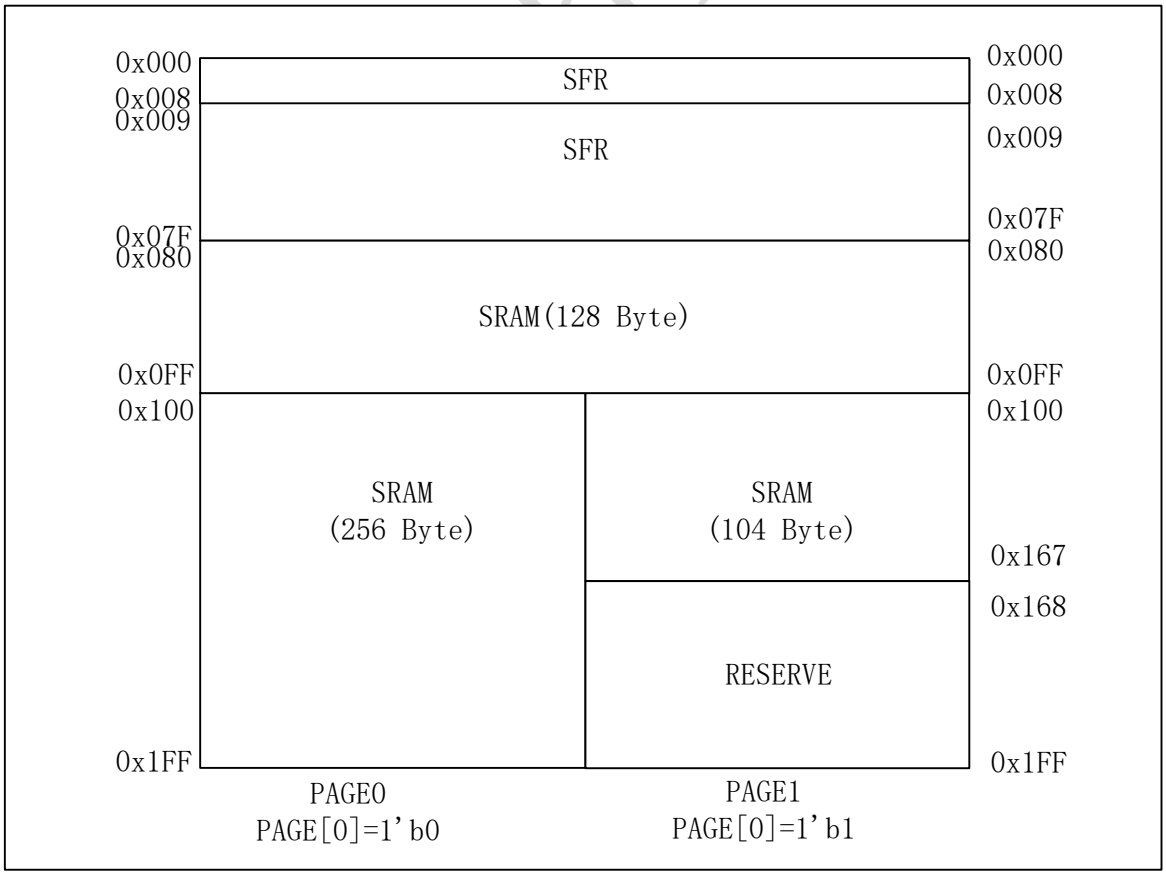


图 4　数据存储器地址分配图

表 2　数据存储器地址分配

数据存储器	起始地址	结束地址
系统特殊功能寄存器	0x00	0x08
外设特殊功能寄存器 (不区分 PAGE)	0x09	0x7F
通用数据存储器 (不区分 PAGE)	0x80	0xFF
通用数据存储器 PAGE0	0x100	0x1FF
通用数据存储器 PAGE1	0x100	0x167

通过 IND0、IRP0 和 FSR0 或 IND1、IRP1 和 FSR1 寄存器可以对数据存储器以及特殊功能寄存器进行间接访问。当从间接地址寄存器(IND0/IND1)读入数据时，MCU 实际上是以 FSR0/FSR1 中的值作为地址去访问数据存储器得到数据。当向间接寄存器(IND0/IND1)写入数据时，MCU 实际上是以 FSR0/FSR1 中的值作为地址去访问数据存储器将值存入该地址。其访问方式见下图

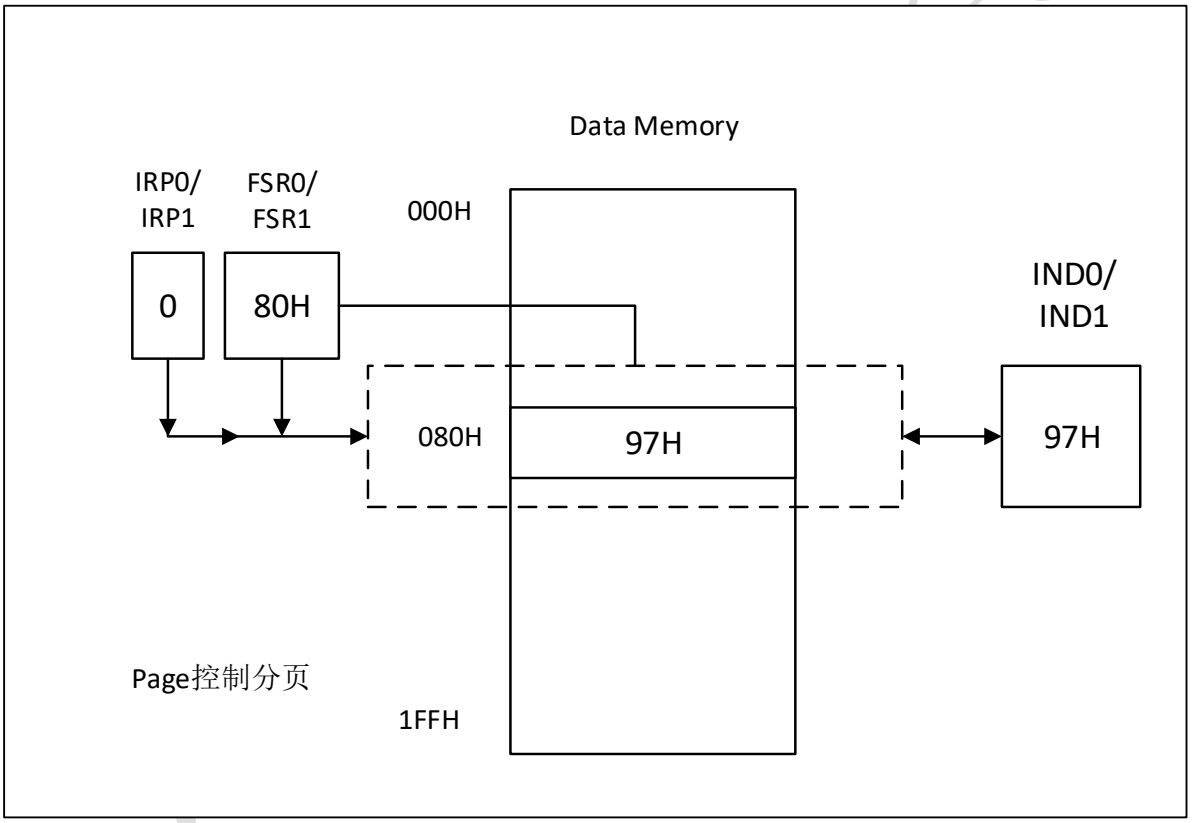


图 5　间接地址访问

1.1.2.　状态寄存器

状态寄存器包含 ALU 的算术状态及复位状态。状态寄存器与其它寄存器一样，可以作为任何指令的目标寄存器。如果状态寄存器是某条指令的目标寄存器，而且影响到 Z，DC 或 C 位，那么对这三个位的写是无效的。这些位是由器件逻辑进行置位或清零。TO 及 PD 位是不可写的。

1.1.2.1.　STATUS 状态寄存器（地址为 04h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	R-0	R-0	R/W-0	R/W-0	R/W-0
STATUS				PD	TO	DC	C	Z

位地址	标识符	功能
7：5	RESERVE	保留
4	PD	掉电标志位，SLEEP 后置此位 1：执行 SLEEP 指令后 0：上电复位后或硬件复位或 CLRWDT 指令之后
3	TO	看门狗定时溢出标志，看门狗定时溢出设置此位 1：看门狗定时溢出发生 0：上电复位后或硬件复位或 CLRWDT 指令后或 SLEEP 指令后
2	DC	半字节进位标志/借位标志 用于进位时 1：结果的第 4 位出现进位溢出 0：结果的第 4 位未出现进位溢出 用于借位时，极性相反 0：结果的第 4 位出现借位溢出 1：结果的第 4 位未出现借位溢出
1	C	进位标志/借位标志 用于进位时 1：结果的最高位（MSB）出现进位溢出 0：结果的最高位（MSB）不出现进位溢出 用于借位时，极性相反 0：结果的最高位（MSB）出现借位溢出 1：结果的最高位（MSB）不出现借位溢出
0	Z	零标志 1：算术运算或逻辑操作结果为 0 0：算术运算或逻辑操作结果不为 0

特性（Property）：
R＝ 可读位　　W＝ 可写位　　U＝ 无效位
-n＝ 上电复位后的值　‘1’ ＝ 位已设置　　‘0’＝ 位已清零 X＝ 不确定位

1.1.3. SFR

特殊功能寄存器（SFR）包含系统专用寄存器和辅助专用寄存器。

系统专用寄存器用于完成 CPU 核的功能，包括间接地址，间接地址指针，状态寄存器，工作寄存器，中断标志及中断控制寄存器。

辅助专用寄存器是为辅助功能而设计，比如 I/O 口，定时器，外设控制寄存器。

表 3　寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
00h	IND0	以 FSR0 中内容作为地址的数据存储器中的数据								xxxxxxxx
01h	IND1	以 FSR1 中内容作为地址的数据存储器中的数据								xxxxxxxx
02h	FSR0	间接数据存储器的地址指针 0								00000000

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
03H	FSR1	间接数据存储器的地址指针 1								00000000
04h	STATUS				PD	TO	DC	C	Z	uuu00000
05h	WORK	工作寄存器								00000000
06h	INTF	BRKIF	TM2IF	CAPIF	TM0IF	SRADIF		E1IF	E0IF	00000u00
07h	INTE	GIE	TM2IE	CAPIE	TM0IE	SRADIE		E1IE	E0IE	00000u00
08h	BSR	IRP0	IRP1						PAGE[0]	00uuuuu0
09h	RSTSR						EMCF	ILOPF		uuuuu00u
0Ah	EADRH	EADR[15:8]								00000000
0Bh	EADRL	EADR [7:0]								00000000
0Ch	EDATH	EDATH[7:0]								00000000
0Dh	WDTCON	WDTEN					WTS[2:0]			0uuuu000
0Eh	WDTIN	WDTIN[7:0]								11111111
0Fh	TM0CON	T0EN	T0RATE[2:0]				T0RSTB	T0SEL[1:0]		0000u100
10h	TM0IN	TM0IN[7:0]								11111111
11h	TM0CNT	TM0CNT[7:0]								00000000
14h	PAGECTL								PAGE2	uuuuuuu0
15h	TM2CAP	CAPEN	CAPCCP[1:0]		CNTCLR	CAPSEL[1:0]		BRKEN	BRKPOL	00000000
16h	MCK			CST_WDT						uu1uuuuu
17h	TM2CON	T2EN	T2RATE[2:0]			T2CKS	T2RSTB	T2OUT	PWM2OUT	00000100
18h	TM2IN	TM2IN[7:0]								11111111
19h	TM2CNT	TM2CNT[7:0]								00000000
1ah	TM2R	TM2R[7:0]								00000000
1bh	TM3CON	T3EN	T3RATE[2:0]			RSV	T3RSTB	T3OUT	PWM3OUT	00000100
1ch	TM3IN	TM3IN[7:0]								11111111
1dh	TM3CNT	TM3CNT[7:0]								00000000
1eh	TM3R1	TM3R1[7:0]								00000000
1fh	TM3INH					TM3INH[11:8]				uuuu0000
20h	PT1			PT1[5:3]					PT1[0]	uuxxxuux
21h	PT1EN			PT1EN[5:3]					PT1EN[0]	uu000uu0
22h	PT1PU			PT1PU[5:3]					PT1PU[0]	uu000uu0
23h	PT1CON			PT1W[2:0]			E1M	E0M[1:0]		uu000000
24h	TM2INH					TM2IN[11:8]				uuuu0000
25h	TM2CNTH	TM3CNT_DT1[2:0]				TM2CNT[11:8]				000u0000
26h	TM2RH	TM3CNT_DT2[2:0]				TM2R[11:8]				000u0000

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
27h	TM3CNTH	TM3CNT_DT3[2:0]				TM3CNT[11:8]				000u0000
28h	PT3	PT3VTH[1]	PT3[6:0]							0xxxxxxx
29h	PT3EN	PT3VTH[2]	PT3EN[6:0]							00000000
2ah	PT3PU	PT3VTH[3]	PT3PU[6:0]							00000000
2bh	PT3CON		PT3CON[6:0]							u0000000
2ch	TM3R3	TM3R3[7:0]								00000000
2dh	TM3CON2	DT3CK[1:0]		DT3CNT[2:0]			DT3_EN	P3H1OEN	P3L1OEN	00000000
2eh	CONFIG1	P3H1INV	P3L1INV		PWM3STALL	PWM2STALL	PWM2PO	PWM3PO1		00u0000u
2fh	CONFIG2	VTHSEL	PWM3MD	PWM3NUM	PWM3WV			PWM3PO3	PWM3PO2	0000uu00
30h	PT5	PT5[7:5]				PT5[3:0]				xxxuxxxx
31h	PT5EN	PT5EN[7:5]				PT5EN[3:0]				000u0000
32h	PT5PU	PT5PU[7:5]			PT1PD[3]	PT5PU[3:0]				00000000
33h	PT5CON	PT5CON[7:5]				PT5CON[3:0]				110u0000
34h	SRADCON0	ADC2V_EN	SARVCMSEL	SRADACKS[1:0]				SRADCKS[1:0]		1100uu00
35h	SRADCON1	SRADEN	SRADS	OFTEN	CALIF	ENOV	OFFEX	VREFS[1:0]		00000010
36h	SRADCON2	CHS[3:0]							SAR_DIFFEN	0000uuu1
37h	SRADL	SRAD[7:0]								00000000
38h	SRADH					SRAD[11:8]				uuuu0000
39h	SROFTL	SROFT[7:0]								00000000
3ah	SROFTH					SROFT[11:8]				uuuu0000
3bh	TM3R2	TM3R2[7:0]								00000000
3ch	INTF2				TM3IF			I2CIF	UR0IF	00u0uu00
3dh	INTE2				TM3IE					u0u0uuuu
40h	INTCFG	INTCFG[7:0]								00000000
41h	PT1CON1								PT1CON[0]	uuuuuuu0
42h	PT5CON1	PT5W[7:5]								000uuuuu
45h	TM3CON3	P3H3INV	P3L3INV	P3H3OEN	P3L3OEN	P3H2INV	P3L2INV	P3H2OEN	P3L2OEN	00000000

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
50h	UR0_CR1	TX9D	RX9D	TX9_EN	RX9_EN	RX_EN	TX_EN	UART0_SEL	UART0_EN	00000000
51h	UR0_BRR0	BRR0[7:0]								00000000
52h	UR0_BRR1		UARTDIV[2:0]			BRR1[3:0]				u0000000
53h	UR0_TX_REG	TX_REG[7:0]								00000000
54h	UR0_RX_REG	RX_REG[7:0]								00000000
55h	UR0_ST	UR_TINV	UR_RINV	UR_SWAP	TXFIFO_EMPTY	RX_BUSY	TX_BUSY	RX_OV_ERR	STOP_ERR	00010000
56h	UR0_INTF				UR0ERRIF	UR0_RHIF	UR0_RNIF	UR0WK_IF	UR0_TEIF	uuu00000
57h	UR0_INTE				UR0ERRIE	UR0_RHIE	UR0_RNIE	UR0WK_IE	UR0_TEIE	uuu00000
59h	I2CCON	I2C_EN	AWK_EN	CST_EN	ACK_EN	I2CSTUS[3:0]				00000000
5ah	I2CCON1	I2C_SEL[1:0]						I2C_DIV[1:0]		0uuuuu00
5bh	I2CDAT	I2CDAT[7:0]								00000000
5ch	I2CADR	I2CADR[6:0]							GC_EN	01001100
5dh	I2C_INTF	I2C_TIF	I2C_RIF	I2C_STIF						000uuuuu
5eh	I2C_INTE	I2C_TIE	I2C_RIE	I2C_STIE						000uuuuu
60h	ISPCON1		ISPWDTRF						ISPOF	u0uuuuu0
63h	WRPRT								WRPRTF	uuuuuuu0
77h	TBLPEN	TBLPEN[7:0]								00h
79h	METCH					METCH3	MCK1[1:0]		METCH0	uuuu0100
7Ch	CONFLASH			CONFLASH[5:0]						uu000000
7Eh	TESTOP					TMOD[3:0]				uuuu0000
7Fh	TEST	TEST[7:0]								00000000

注：进行读操作时，无效位读出为 0

特性（Property）：
R = 可读位　　W = 可写位　　U = 无效位
-n = 上电复位后的值　‘1’ = 位已设置　　‘0’ = 位已清零 X = 不确定位

1.2. 时钟系统

1.2.1. 概述

芯片的时钟系统包括内置 32MHz 的 RC 振荡时钟（HIRC）、内置低速 32KHz 的 WDT 时钟。仅内置 32MHz RC 振荡时钟可以做为系统时钟源 Fosc。Fcpu 是 CPU 时钟频率计算方法如下：

Fcpu=Fosc/N，N=4、8、16

1.2.2. 时钟框图

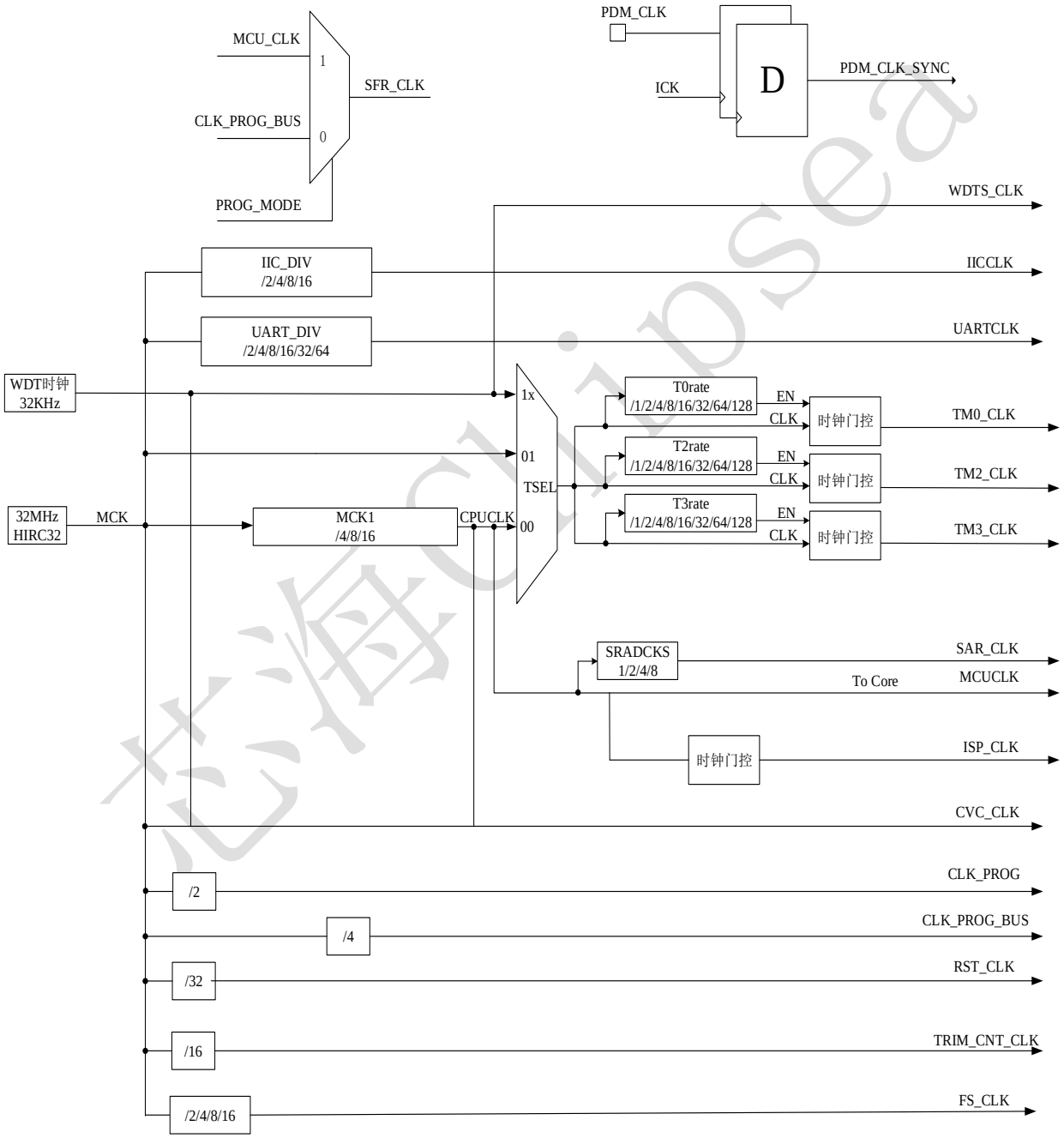


图 6　CS8M32X 时钟树框图

1.2.3.　寄存器

表 4　CS8M32X 时钟系统寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bit2	Bit1	Bit0	上电复位值
16h	MCK			CST_WDT						uu1uu0u0
79h	METCH					METCH 3	MCK1[1:0]		METCH 0	uuuu0100

1.2.3.1.　MCK 寄存器（地址 16h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	R/W-1	U-0	U-0	R/W-0	U-0	R/W-0
MCK			CST_WDT					

位地址	标识符	功能
5	CST_WDT	内部 32K 低速振荡器启动开关，当 WDT_CFG 配置为内部 32K 低速振荡器固定打开时，该位无效。 1：内部 32K 低速振荡器关闭 0：内部 32K 低速振荡器打开

对 MCK 寄存器进行写操作时，建议使用 BCF 或 BSF 指令。

1.2.3.2.　METCH 寄存器（地址 79h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-1	R/W-0	R/W-0
METCH					METCH3	MCK1[1:0]		METCH0

位地址	标识符	功能
3	METCH3	POR bias 和 LVR 测试使能。 1：por_bias 从 PT3.0 输出；同时 lvr 测试输出口从 PT1.0 输出 0：关闭使能。
2:1	MCK1[1:0]	指令周期选择： 00：保留 01：指令周期 4MHz 即 8 个时钟周期 10：指令周期 2MHZ 即 16 个时钟周期（默认） 11：保留 注： 1、在从高频率切换到低频率时，最大有 160us 的延时，输出切换后的时钟 2、a、电源电压低于 2.4V 时，必须选择 2MHZ 指令周期 b、电源电压高于 2.4V 时，才可以选择 4MHZ 指令周期
0	METCH0	PT3.1/PT3.2/PT3.3 IO 口防倒灌功能使能 1：使能防倒灌功能，0：禁止防倒灌功能（1）

注(1)：METCH0 寄存器为 0 时，PT3.1/PT 3.2 用作 GPIO/I2C 则开漏输出，PT3.1/PT 3.2 用作 UART 则推挽输出，PT3.3 用作 GPIO 则推挽输出。

1.2.4.　内部高速 RC 时钟

内部高速 RC 时钟（32MHz），系统只能使用内部高速 RC 时钟做为系统的主时钟。

1.2.5. 内部低速 WDT 时钟

内部低速 WDT 时钟（32KHz），可以通过 WDT_CFG 配置，当 WDT_CFG 为 0 时，内部 32K 低速振荡器固定打开，软件无法关闭。当 WDT_CFG 为 1 时，通过寄存器 CST_WDT 使能开关。内部 WDT 时钟不能做为系统主时钟，只能做为 WDT、定时器时钟、CVC 时钟使用。

1.3. 复位系统

CS8M32X 有以下方式复位：

1） 上电复位
2） NRST 硬件复位（正常操作）
3） NRST 硬件复位（从 SLEEP 模式）
4） WDT 复位（正常操作）
5） WDT 复位（从 SLEEP 模式）
6） 低电压复位（LVR）
7） 非法指令复位
8） EMC 复位

上述任意一种复位发生时，所有系统寄存器恢复默认状态（WDT 复位时 TO、PD 标志位除外），程序停止运行，同时程序计数器 PC 清零。复位结束后，系统地址从 000H 重新开始。各种复位情况下的 TO，PD 标志位如下表所示。

表 5　复位信号和状态寄存器关系

条件	TO	PD	EMCF	ILOPF
上电复位	0	0	0	0
NRST 硬件复位（正常操作）	0	0	0	0
NRST 硬件复位（从 SLEEP 模式）	0	0	0	0
WDT 复位（正常操作）	1	不变	不变	不变
WDT 复位（从 SLEEP 模式）	1	1	不变	不变
低电压复位	0	0	不变	不变
非法指令复位	不变	不变	不变	1
EMC 复位	不变	不变	1	0

复位电路原理图如下所示

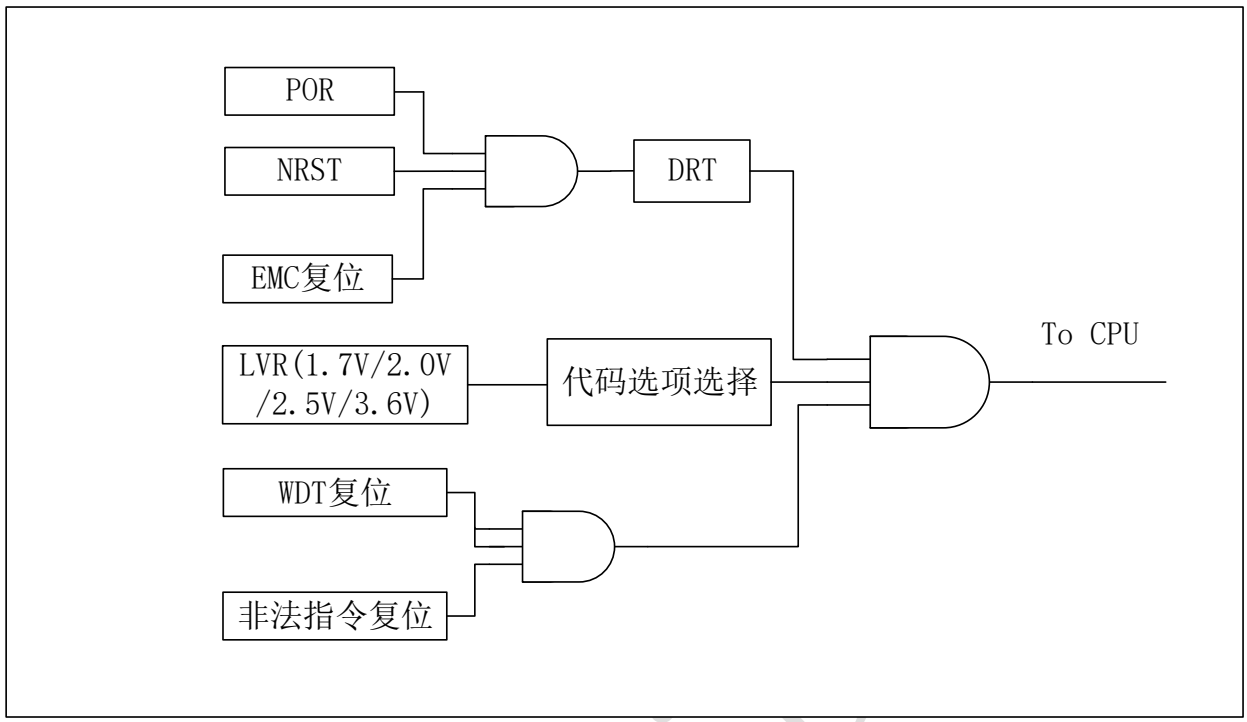


图 7　复位电路框图

任何一种复位情况都需要一定的响应时间，系统提供完善的复位流程以保证复位动作的顺利进行。对于不同类型的振荡器起振的时间不同，所以完成复位的时间也有所不同。用户应在上电复位后，预留一定的时间等待系统稳定。用户在终端使用过程中，应注意考虑主机对上电复位的要求。

1.3.1.　上电复位/掉电复位

系统上电时，电压呈现逐渐上升的曲线形式，需要一定时间才能达到正常的工作电压①。上电复位（POR）电路会监测芯片电源电压（VDD），保持 POR 状态，直至 VDD 超过 V_{POR} 阈值；VDD 超过 V_{POR} 阈值后，POR 电路延时 t_{WVS} 后才会释放 POR 状态，以等待振荡器稳定、VDD 上升至正常工作电压。用户应保证 VDD 在 t_{WVS} 结束前上升到正常的工作电压；否则，需要开启低电压复位（LVR）功能，并选择合适的 LVR 电压，以保证芯片正常工作。

注：对于不同的指令周期所需工作电压是不同的，指令周期越快相应所需的工作电压就越高，见数据手册“直流特性”。如果指令周期是 4MHz 时，建议使用 2.5V 低电压复位。

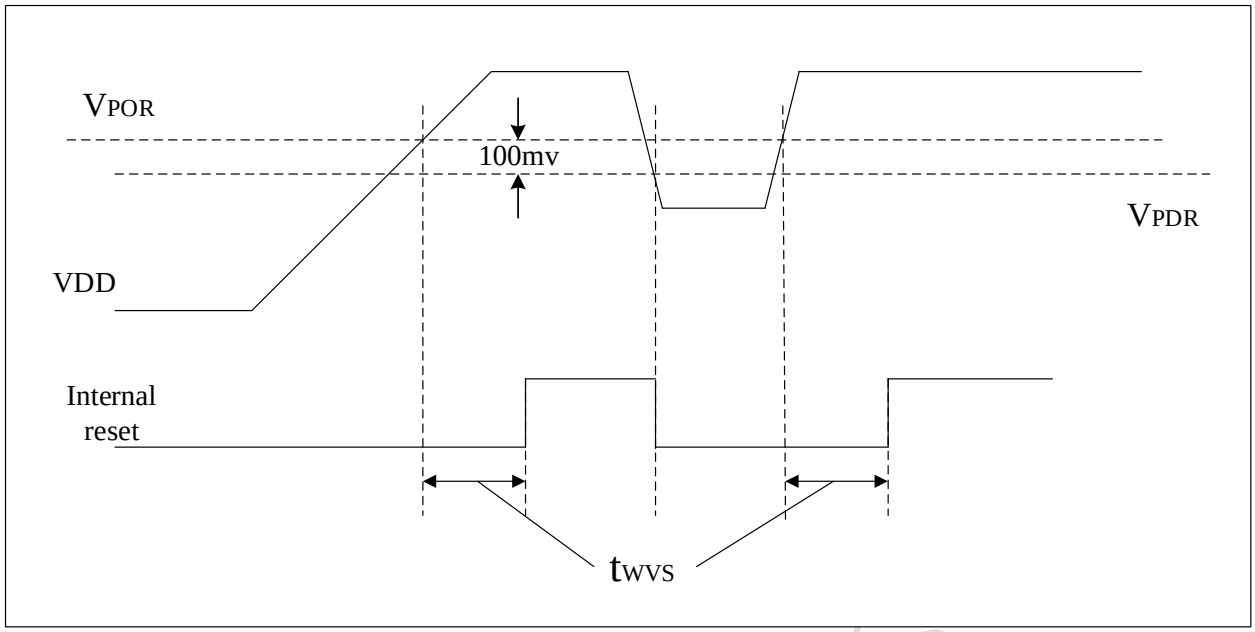


图 8　上电复位电路示例及上电过程

参数	最小值	典型值	最大值
V_{POR}	1.1V	1.4V	1.8V
V_{PDR}	1.0V	1.3V	1.65V
t_{WVS} (测试条件：VDD=5V，T=25℃)	21ms	30ms	41.6ms

V_{POR}：上电复位
V_{PDR}：掉电复位
t_{WVS}：等待电压稳定时间

1.3.2.　掉电复位

掉电复位是针对外部引起的电源电压跌落情况，例如受到干扰或者负载变化。电源掉电可能会引起系统工作状态不正常或者程序执行错误。

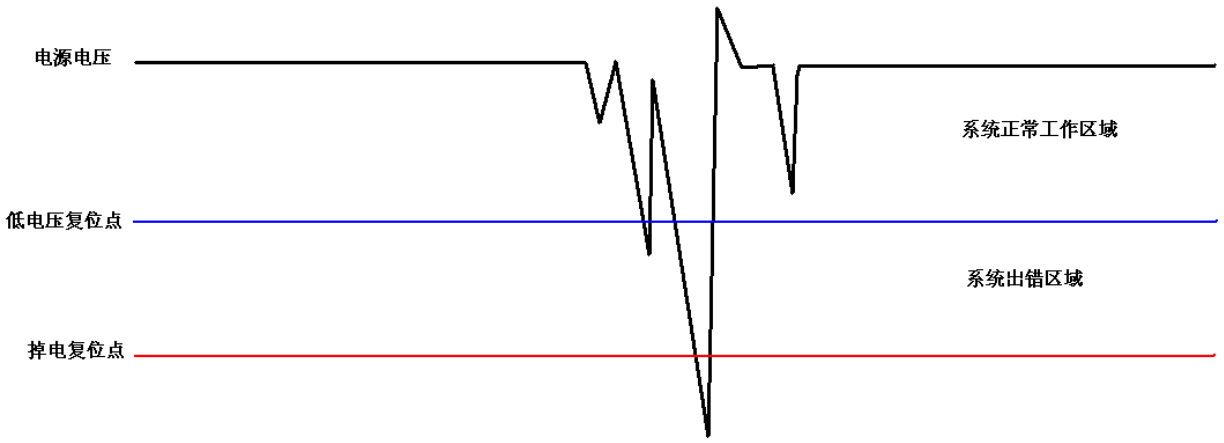


图 9　系统掉电复位示意图

电源电压跌落可能会导致芯片进入系统死区。系统死区，即电源电压不能满足系统的最小工作电压[①]要求。为避免进入系统死区，建议利用低电压复位（LVR）功能，并选择合适的 LVR 电压[①]。

掉电复位性能的改善可以通过如下几点实现：

1) 低电压复位（LVR）
2) 看门狗复位
3) 降低系统指令周期
4) 采用外部复位电路（稳压二极管复位电路；电压偏移复位电路；外部 IC 复位）

注①：对于不同的指令周期所需工作电压是不同的，指令周期越快相应所需的工作电压就越高，见数据手册“直流特性”。如果指令周期是 4MHz 时，建议使用 2.5V 低电压复位。

1.3.3. 低电压复位（LVR）

低电压复位（LVR）功能提供内部低电压复位。低电压复位具有迟滞功能，迟滞电压为 100mV。低电压复位满足阈值条件后产生复位。

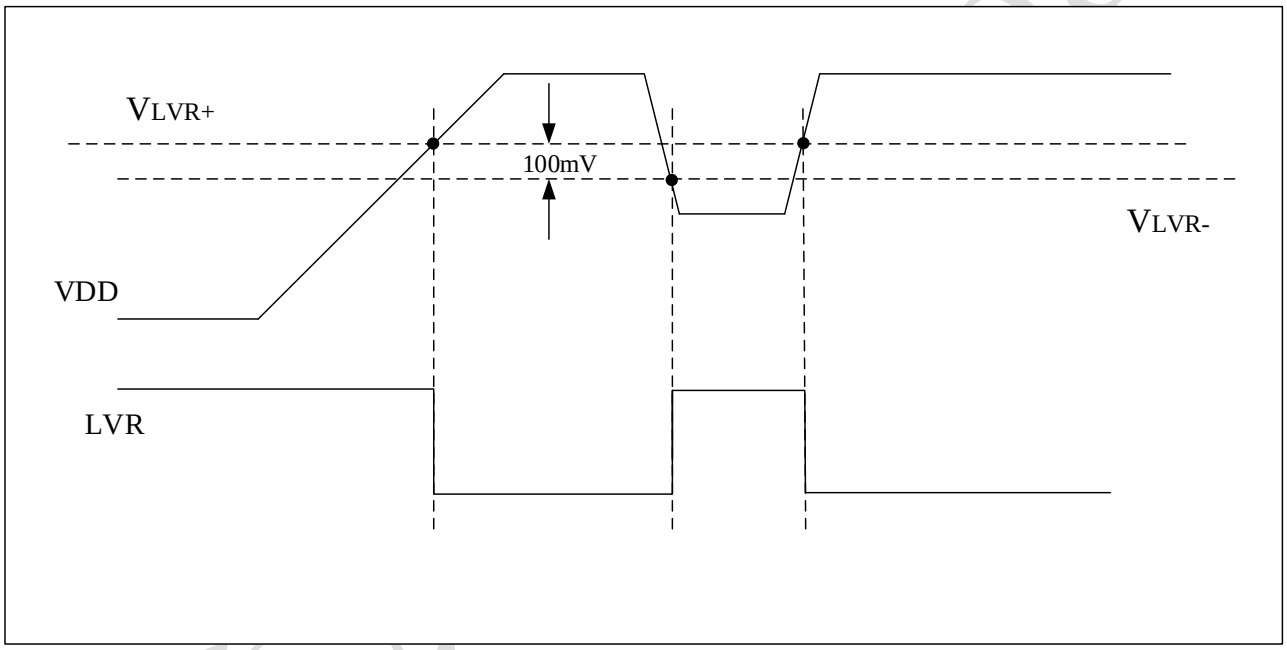


图 10　系统 LVD 示意图

1.3.4. 外部硬件复位(NRST)

外部复位(RST)由代码选项 RESET_PIN 控制。通过设置该代码选项，可使能外部硬件复位功能。外部硬件复位引脚为施密特触发结构，根据代码选项区可配置成低电平有效或者高电平有效。硬件复位引脚为非复位电平时，系统正常工作；硬件复位引脚为复位电平时，系统复位。

在芯片代码选项使能外部硬件复位功能后，需要注意的是：在系统上电完成后，外部复位需要输入非复位电平，否则，系统会一直复位，直到外部硬件复位结束。

外部硬件复位可以在上电过程中使用系统复位。良好的外部复位电路可以保护系统避免进入系统死区。

1.3.5. 看门狗复位

看门狗复位是一种系统保护设置。在正常状态下，程序将看门狗定时器清零。如出错，系统处于未知状态，此时利用看门狗复位。看门狗复位后，系统重新进入正常状态。

看门狗复位可以唤醒 SLEEP 模式和 HALT 模式，芯片复位，系统重新进入正常状态。

1.3.6. 非法指令复位

为了增强芯片抗干扰能力，芯片会自动检测系统非法指令，如果检测到非法指令，自动产生 MCU 复位信号，将芯片复位。可以通过代码选项关闭非法指令使能位。

非法指令包括以下几种情况：

1、 本文档指令集和伪指令以外的其他指令码。

2、 在中断服务程序中将 GIE 位置 1。

1.3.7. EMC 复位

为了防止由于 ESD、EFT 等强干扰导致芯片寄存器被改写，CS8M32X 对关键寄存器增加影子寄存器，它们之间是反码关系，当系统检测到它们之间的反码关系不成立，则产生复位信号将芯片复位可以通过代码选项关闭 EMC 使能位。

CS8M32X 中增加了影子寄存器进行校验的寄存器包括以下几个：

1、 WDT 模块使能位

2、 WDT 时钟使能位

3、 代码选项

4、 SLEEP 寄存器

影子寄存器的值用户无法读取，仅做校验使用。

1.3.8. 寄存器说明

表 6　复位系统寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bit0	上电复位值
09h	RSTSR						EMCF	ILOPF		uuuuu00u

1.3.8.1. RSTSR 寄存器（地址 09h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0
RSTSR						EMCF	ILOPF	

位地址	标识符	功能
2	EMCF	EMC 复位标志位 该位硬件置 1，软件清零。 0：未发生 EMC 复位 1：发生了 EMC 复位
1	ILOPF	非法指令复位标志位 该位硬件置 1，软件清零。 0：未发生非法指令复位 1：发生了非法指令复位
0	RESERVE	保留

1.4. 中断

1.4.1. 中断概述

CS8M32X 有 13 个中断源，只有 1 个中断入口地址 004H。与中断相关的 SFR：中断使能控制寄存器 INTE 和中断标志位寄存器 INTF。除 CVC_SLP_IF、BRKIF、IIC 和 UART 外，其他 9 个中断源都各自有一个中断使能，并且共用一个中断总使能位 GIE，它们的标志位硬件置位，软件清 0。其中 I2CIF 和 UR0IF 中断标志是有对应 I^2C 模块和 URAT 模块产生的，只可读，清除时要清除对应模块内部的中断标志。

当芯片响应中断请求时，会把当前的 PC 值入栈保护，并将 PC 置为 004H，同时把总使能位 GIE 清 0，执行完中断服务程序后,用户用 RETFIE 指令返回到之前的主程序，同时硬件把 GIE 置 1。

CS8M32X 中断系统不支持中断嵌套，没有中断优先级，因此禁止在中断服务程序中将 GIE 位置 1，如果在中断服务程序中将 GIE 置 1，硬件自动产生复位请求，将芯片复位。产生该复位后，硬件自动将 RSTS 寄存器的 ILOPF 位置 1，需要软件进行清零。

所有的中断都可以唤醒 SLEEP 睡眠模式和 HALT 停止模式。

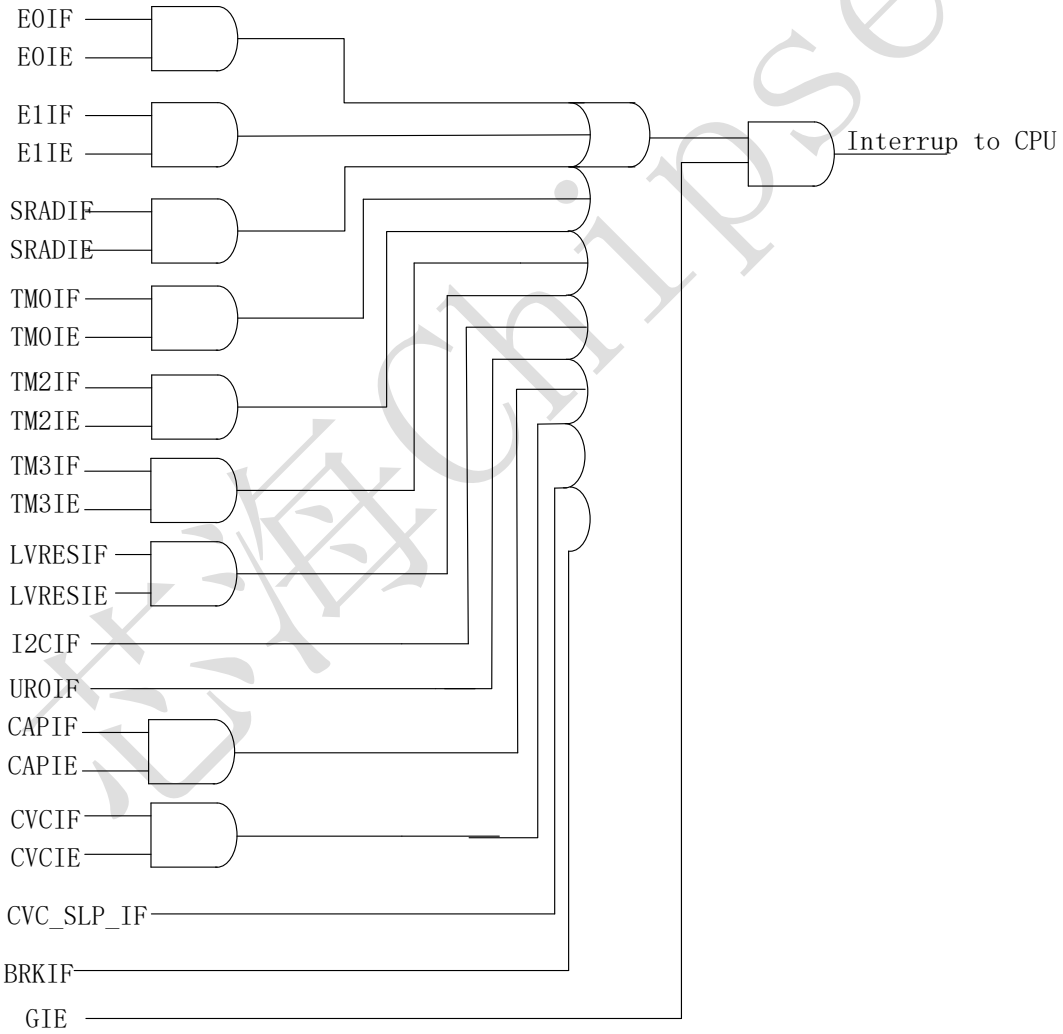


图 11　中断逻辑

1.4.2.　中断使能寄存器描述

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06H	INTF	BRKIF	TM2IF	CAPIF	TM0IF	SRADIF		E1IF	E0IF	00000u00
07H	INTE	GIE	TM2IE	CAPIE	TM0IE	SRADIE		E1IE	E0IE	00000u00
23h	PT1CON			PT1W[2:0]			E1M	E0M[1:0]		uu000000
40h	INTCFG	INTCFG[7:0]								00000000
42h	PT5CON1	PT5W[7：5]								000uuuuu

1.4.2.1.　INTE 寄存器（地址为 07h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INTE	GIE	TM2IE	CAPIE	TM0IE	SRADIE		E1IE	E0IE

位地址	标识符	功能
7	GIE	全局中断使能 1 = 使能所有非屏蔽中断 0 = 不使能所有中断
6	TM2IE	12-Bit 定时/计数器 2 中断使能 1 = 使能定时/计数器 2 中断 0 = 不使能定时/计数器 2 中断
5	CAPIE	捕获事件中断使能位 0：不使能捕获事件中断 1 = 使能捕获事件中断
4	TM0IE	8-Bit 定时 0 器中断使能 1 = 使能定时器 0 中断 0 = 不使能定时器 0 中断
3	SRADIE	SAR_ADC 中断使能 1 = 使能 SAR_ADC 中断 0 = 不使能 SAR_ADC 中断
2	RESERVE	保留
1	E1IE	外部中断 1 使能 1 =使能外部中断 1 0 = 不使能外部中断 1
0	E0IE	外部中断 0 使能 1 = 使能外部中断 0 0 = 不使能外部中断 0

1.4.2.2.　INTE2 寄存器（地址为 3dh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W-0	U-0	R/W-0	U-0	R/W-0	U-0	U-0
INTE2				TM3IE				

位地址	标识符	功能
7-5	RESERVE	保留
4	TM3IE	12-Bit 定时/计数器 3 中断使能 1 = 使能定时/计数器 3 中断 0 = 不使能定时/计数器 3 中断
3	RESERVE	保留
2	RESERVE	保留
1：0	RESERVE	保留

特性（Property）：
R = 可读位　　W = 可写位　　U = 无效位
-n = 上电复位后的值　'1' = 位已设置　　'0' = 位已清零 X = 不确定位

1.4.3. 中断标志寄存器

中断标志位都是硬件置 1，软件清 0。**在对应的中断使能位没有置 1 的情况下，中断标志位也会硬件置 1**。所以建议在中断函数里先判断中断使能位是否打开，然后再判断相应的中断标志位。否则有可能造成中断函数误执行。

1.4.3.1. INTF 寄存器（地址为 06h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INTF	BRKIF	TM2IF	CAPIF	TM0IF	SRADIF		E1IF	E0IF

位地址	标识符	功能
7	BRKIF	刹车事件中断标志位，该位硬件置 1，软件写 0 清零。 0：未发生刹车事件 1：发生刹车事件
6	TM2IF	12-Bit 定时/计数器 2 中断标志，软件清零，硬件置高 1 = 发生定时中断，必须软件清 0 0 = 没发生定时中断
5	CAPIF	捕获事件中断标志位，该位硬件置 1，软件写 0 清零。 0：未发生捕获事件 1：发生捕获事件
4	TM0IF	8-Bit 定时器 0 中断标志，软件清零，硬件置高 1 = 发生定时中断，必须软件清 0 0 = 没发生定时中断
3	SRADIF	SARAD 中断中断标志，软件清零，硬件置高 1 = 发生 AD 中断，必须软件清 0 0 = 没发生 AD 中断
2	RESERVE	保留
1	E1IF	外部中断 1 中断标志，软件清零，硬件置高 1 =外部中断 1 发生中断，必须软件清 0 0 =外部中断 1 没发生中断
0	E0IF	外部中断 0 中断标志，软件清零，硬件置高

位地址	标识符	功能
		1 = 外部中断 0 发生中断，必须软件清 0 0 =外部中断 0 没发生中断

1.4.3.2. INTF2 寄存器（地址为 3ch）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R-0	R-0
INTF2				TM3IF			I2CIF	UR0IF

位地址	标识符	功能
7:5	RESERVE	保留
4	TM3IF	12-Bit 定时/计数器 3 中断标志，软件清零，硬件置高 1 = 发生定时中断，必须软件清 0 0 = 没发生定时中断
3	RESERVE	保留
2	RESERVE	保留
1	I2CIF	I^2C 模块总中断标志位（只读） 0 = I^2C 模块未发生接收或发送中断 1 = I^2C 模块发生接收或发送中断
0	UR0IF	UART0 模块总中断标志位（只读） 0 = UART0 模块未发生接收或发送中断 1 = UART0 模块发生接收或发送中断

特性（Property）：
R = 可读位　　W = 可写位　　U = 无效位
-n = 上电复位后的值　‘1’ = 位已设置　　‘0’ = 位已清零　X = 不确定位

1.4.4. 外部中断 0

PT3.6 和 PT1.0 为外部中断 0 的输入端。触发方式由 PT1CON 寄存器中的 E0M[1:0]寄存器决定。INTE 寄存器中的 E0IE 为外部中断 0 的使能位，INTF 寄存器中的 E0IF 为中断标志位，硬件置 1，软件清 0。可唤醒 SLEEP 或 HALT 模式。只要 PT3.6 或 PT1.0 被触发，中断标志位 E0IF 就会置 1。

1.4.4.1. PT1CON 寄存器（地址为 23h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT1CON			PT1W[2:0]			E1M	E0M[1:0]	

位地址	标识符	功能
2	E1M	外部中断 1 触发模式 1 = 外部中断 1 为下降沿触发 0 = 外部中断 1 在状态改变时触发
1：0	E0M[1:0]	外部中断 0 触发模式 11 = 外部中断 0 在状态改变时触发 10 = 外部中断 0 在状态改变时触发

		01 = 外部中断 0 为上升沿触发 00 = 外部中断 0 为下降沿触发

1.4.4.2. INTCFG 寄存器（地址为 40h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTCFG	INTCFG[7:0]							

位地址	标识符	功能
7	INTCFG [7]	PT1.0 外部中断 0 使能 0 = 禁止 PT1.0 外部中断 0 1 = 使能 PT1.0 外部中断 0
6	INTCFG [6]	PT3.6 外部中断 0 使能 0 = 禁止 PT3.6 外部中断 0 1 = 使能 PT3.6 外部中断 0

1.4.5. 外部中断 1

PT1.3、PT1.4、PT1.5、PT3.0、PT3.1、PT3.2、PT3.3、PT3.4、PT3.5、PT5.5、PT5.6、PT5.7 都可作为外部中断 1 的输入端。触发方式由 PT1CON 寄存器中的 E1M 寄存器决定。INTE 寄存器中的 E1IE 为外部中断 1 的使能位，INTF 寄存器中的 E1IF 为中断标志位，硬件置 1，软件清 0。只要对应 PT 口作为外部中断输入端，且外部中断 1 被触发，中断标志位 E1IF 就会置 1。

1.4.5.1. PT1CON 寄存器（地址为 23h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT1CON			PT1W[2:0]			E1M	E0M[1:0]	

位地址	标识符	功能
5	PT1W[2]	PT1.5 外部中断 1 使能 0 = 禁止 PT1.5 外部中断 1 1 = 使能 PT1.5 外部中断 1
4	PT1W[1]	PT1.4 外部中断 1 使能 0 = 禁止 PT1.4 外部中断 1 1 = 使能 PT1.4 外部中断 1
3	PT1W[0]	PT1.3 外部中断 1 使能 0 = 禁止 PT1.3 外部中断 1 1 = 使能 PT1.3 外部中断 1
2	E1M	外部中断 1 触发模式 1 = 外部中断 1 为下降沿触发 0 = 外部中断 1 在状态改变时触发

1.4.5.2. INTCFG 寄存器（地址为 40h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTCFG	INTCFG[7:0]							

位地址	标识符	功能
5	INTCFG [5]	PT3.5 外部中断 1 使能 0 =　禁止 PT3.5 外部中断 1 1 =　使能 PT3.5 外部中断 1
4	INTCFG [4]	PT3.4 外部中断 1 使能 0 =　禁止 PT3.4 外部中断 1 1 =　使能 PT3.4 外部中断 1
3	INTCFG [3]	PT3.3 外部中断 1 使能 0 =　禁止 PT3.3 外部中断 1 1 =　使能 PT3.3 外部中断 1
2	INTCFG [2]	PT3.2 外部中断 1 使能 0 =　禁止 PT3.2 外部中断 1 1 =　使能 PT3.2 外部中断 1
1	INTCFG [1]	PT3.1 外部中断 1 使能 0 =　禁止 PT3.1 外部中断 1 1 =　使能 PT3.1 外部中断 1
0	INTCFG [0]	PT3.0 外部中断 1 使能 0 =　禁止 PT3.0 外部中断 1 1 =　使能 PT3.0 外部中断 1

1.4.5.3.　PT5CON1 寄存器（地址为 42h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
PT5CON1	PT5W[7：5]							

位地址	标识符	功能
7	PT5W[7]	PT5.7 外部中断 1 使能 0 =　禁止 PT5.7 中断 1 1 =　使能 PT5.7 中断 1
6	PT5W[6]	PT5.6 外部中断 1 使能 0 =　禁止 PT5.6 部中断 1 1 =　使能 PT5.6 部中断 1
5	PT5W[5]	PT5.5 外部中断 1 使能 0 =　禁止 PT5.5 部中断 1 1 =　使能 PT5.5 部中断 1

特性（Property）：
R = 可读位　　W = 可写位　　U = 无效位
-n = 上电复位后的值　‘1’ = 位已设置　　‘0’ = 位已清零 X = 不确定位

1.4.6.　ADC 中断

INTE 寄存器中的 SRADIE 为 ADC 中断的使能位，INTF 寄存器中的 SRADIF 为中断标志位，软件清

0。当 ADC 转换完成时， SRADIF 就会硬件置 1。

1.4.7. 定时器 0 溢出中断

INTE 寄存器中的 TM0IE 为定时器 0 中断的使能位，INTF 寄存器中的 TM0IF 为中断标志位，软件清 0。当定时器 0 溢出时， TM0IF 就会硬件置 1。

1.4.8. 定时/计数器 2 溢出中断

INTE 寄存器中的 TM2IE 为定时/计数器 2 中断的使能位，INTF 寄存器中的 TM2IF 为中断标志位，软件清 0。当定时/计数器 2 溢出时， TM2IF 就会硬件置 1。

1.4.9. 定时/计数器 3 溢出中断

INTE2 寄存器中的 TM3IE 为定时/计数器 3 中断的使能位，INTF2 寄存器中的 TM3IF 为中断标志位，软件清 0。当定时/计数器 3 溢出时， TM3IF 就会硬件置 1。

1.4.10. PUSH 和 POP 处理

CS8M32X 有 8 级的 PUSH 和 POP 堆栈。有中断请求被响应后，程序跳转到 004h 执行子程序。响应中断之前必须保存 WORK 和 STATUS 中的的标志位(只保存 C，DC，Z)。芯片提供 PUSH 和 POP 指令进行入栈保存和出栈恢复功能，从而避免中断结束后程序运行错误。子程序中也可以使用 PUSH 和 POP 指令对 WORK 和 STATUS(C，DC，Z)进行保存和恢复。

1.4.11. I^2C 中断

I2C_INTE 寄存器中的 I2C_TIE、I2C_RIE 以及 I2C_STIE 为 I^2C 的不同中断对应的使能信号。INTF2 寄存器中的 I2CIF 为 I^2C 的总中断标志位，只可读；清除 I^2C 模块内部的中断标志（I2C_TIF、I2C_RIF 以及 I2C_RIF），就清除了 I2CIF。当 I^2C 模块内部中断使能打开并且对应的中断标志置位，I2CIF 就会硬件置 1。

1.4.12. UART 中断

UR0_INTE 寄存器中的 UR0ERRIE、UR0_RHIE、UR0_RNIE、UR0WK_IE 以及 UR0_TEIE 为 UART 的不同中断对应的使能信号。INTF2 寄存器中的 UR0IF 为 UART 的总中断标志位，只可读；清除 UART 模块内部的中断标志（UR0ERRIF、UR0_RHIF、UR0_RNIF、UR0WK_IF 以及 UR0_TEIF），就清除了 UR0IF。当 UART 模块内部中断使能打开并且对应的中断标志置位，UR0IF 就会硬件置 1。

1.4.13. 刹车中断

BRKEN 为刹车的使能，INTF 寄存器中的 BRKIF 为中断标志位，软件清 0。当外部输入刹车功能有效时， BRKIF 就会硬件置 1。

1.4.14. 捕获中断

INTE 寄存器中的 CAPIE 为捕获中断的使能位，INTF 寄存器中的 CAPIF 为中断标志位，软件清 0。当捕获动作完成时， CAPIF 就会硬件置 1。

```
        …
        ORG    004H
        GOTO INT_SERVER
    …
INT_SERVER:
        PUSH
        BTFSS INTE,E0IE     ;判断外部中断 0 中断使能
        GOTO    $+3
        BTFSC INTF,E0IF     ;判断外部中断 0 标志
        GOTO    EX0_INT

        BTFSS INTE,E1IE     ;判断外部中断 1 中断使能
        GOTO    $+3
        BTFSC INTF,E1IF     ;判断外部中断 1 标志
        GOTO    EX1_INT

        BTFSS INTE,TM0IE    ;判断定时/计数器 0 中断使能
        GOTO    $+3
        BTFSC INTF,TM0IF    ;判断定时器 0 中断标志
        GOTO    TM0_INT

        BTFSS INTE,TM2IE    ;判断定时/计数器 2 中断使能
        GOTO    $+3
        BTFSC INTF,TM2IF    ;判断定时/计数器 2 中断标志
        GOTO    TM2_INT

        BTFSS INTE2，TM3IE    ;判断定时/计数器 3 中断使能
        GOTO    $+3
        BTFSC INTF2，TM3IF    ;判断定时/计数器 3 中断标志
        GOTO    TM3_INT


    …
EX0_INT:
        BCF INTF, E1IF       ;清除 E1IF
        …
        POP
        RETFIE
        …
```

1.5. 定时器 0

1.5.1. 定时器 0 概述

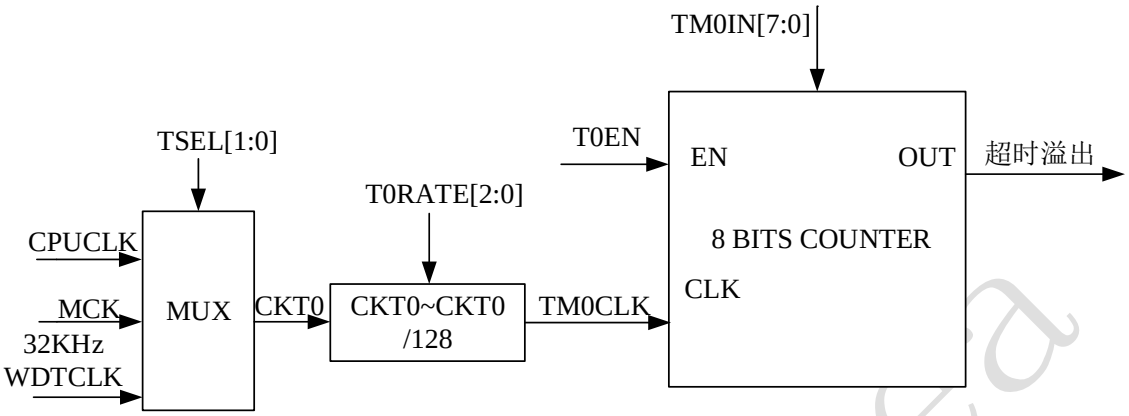


图 12　定时器 0 功能框图

定时器 0 模块的输入时钟通过 T0SEL[1:0]进行选择，可选时钟源有 CPUCLK、MCK 或者内部 32K WDT 时钟。当选择内部 32KWDT 时钟进行计数时，可以唤醒 SLEEP 模式。定时器 0 模块集成了一个分频器，分频后的时钟 TM0CLK 作为 8 Bits 计数器的输入时钟。

当用户设置了定时器 0 模块的使能位，8 Bits 计数器将启动，计数值将会从 000H 递增到 TM0IN。用户需要设置 TM0IN（定时器 0 模块计数溢出值）以选择定时超时中断时间。

当定时超时发生时，定时器 0 中断标志位 TM0IF 硬件置 1，该位只能通过软件清零。如果使能了定时器 0 中断（TM0IE=1）和中断总使能（GIE），程序计数器会跳转到 004H 以执行中断服务程序。

1.5.2. 定时器 0 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06H	INTF				TM0IF					u0u00u00
07H	INTE	GIE			TM0IE					00u00u00
0FH	TM0CON	T0EN	T0RATE[2:0]				T0RSTB	TSEL[1:0]		0000u100
10H	TM0IN	TM0IN[7:0]								11111111
11H	TM0CNT	TM0CNT[7:0]								00000000

1.5.3. TM0CON 寄存器(地址 0FH)

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
TM0CON	T0EN	T0RATE[2:0]				T0RSTB	TSEL[1:0]	

位地址	标识符	功能
7	T0EN	定时器 0 使能位 1：使能定时器 0 0：禁止定时器 0
6:4	T0RATE[2:0]	定时器 0 时钟分频选择，其中 CKT0 为通过 T0SEL[1:0]选择的时钟源。

<table>
<tr><th>位地址</th><th>标识符</th><th>功能</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>T0RATE [2:0]</td><td>TM0CLK</td></tr>
<tr><td>000</td><td>CKT0</td></tr>
<tr><td>001</td><td>CKT0/2</td></tr>
<tr><td>010</td><td>CKT0/4</td></tr>
<tr><td>011</td><td>CKT0/8</td></tr>
<tr><td>100</td><td>CKT0/16</td></tr>
<tr><td>101</td><td>CKT0/32</td></tr>
<tr><td>110</td><td>CKT0/64</td></tr>
<tr><td>111</td><td>CKT0/128</td></tr>
</table>
</td></tr>
<tr><td>2</td><td>T0RSTB</td><td>定时器 0 复位，默认值为 1
1：禁止定时器 0 复位
0：使能定时器 0 复位
当将该位配置为 0，一个指令周期后定时器 0 复位完成，T0RSTB 硬件置 1。</td></tr>
<tr><td>1:0</td><td>TSEL[1:0]</td><td>时钟源选择
<table>
<tr><td>T0SEL[1:0]</td><td>所有定时器时钟源</td></tr>
<tr><td>00</td><td>CPUCLK</td></tr>
<tr><td>01</td><td>MCK</td></tr>
<tr><td>1x</td><td>内部 32K WDT 时钟，
仅当内部 WDT 晶振打开时有效</td></tr>
</table>
</td></tr>
</table>

1.5.3.1. TM0IN 寄存器(地址 10H)

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TM0IN	TM0IN[7:0]							

位地址	标识符	功能
7 ： 0	TM0IN[7:0]	定时器 0 溢出值（溢出值：1~255）

1.5.3.2. TM0CNT 寄存器(地址 11H)

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM0CNT	TM0CNT[7:0]							

位地址	标识符	功能
7 ： 0	TM0CNT[7:0]	定时器 0 计数寄存器，**只读**

1.5.4. 定时功能

定时器 0 的时钟源可以选择 CPUCLK、MCK、或者内部 32KWDT 时钟。当选择内部 32KWDT 时钟进行计数时，可以唤醒 SLEEP 模式。

定时功能操作步骤如下。操作：

1) 设置 TM0CON 低 2 位值，为定时器 0 模块选择时钟输入。
2) 设置 TM0IN，选择定时器 0 溢出值。（溢出值：1~255）
3) 清除定时器 0 中断标志位：将 INTF 寄存器的 TM0IF 位清零。
4) 设置定时器 0 中断使能：将 INTE 寄存器的 TM0IE 位 与 GIE 位置位，使能定时器 0 中断。
5) 清零定时器 0 计数值：将 TM0CON 寄存器的 T0RSTB 位配置为 0，复位定时器 0 模块的计数器。
6) 使能定时器 0 模块：将 TM0CON 寄存器的 T0EN 置位，使能定时器 0 模块的 8 bits 计数器。
7) 当定时超时发生时，程序计数器会跳转到 004H。

定时器 0 溢出时间计算方法：
定时器 0 溢出时间=（TM0IN+1）/TM0CLK.

1.6. I/O PORT

1.6.1. I/O 口概述

微控制器中的通用 I/O 口（GPIO）用于通用的输入与输出功能。用户可以通过 GPIO 接收数据信号或将数据传送给其它的数字设备。CS8M32X 的部分 GPIO 可以被定义为其它的特殊功能。在本节，只说明 GPIO 的通用 I/O 口功能，特殊功能将会在接下来的章节中说明。
补充类似下图的示意图：

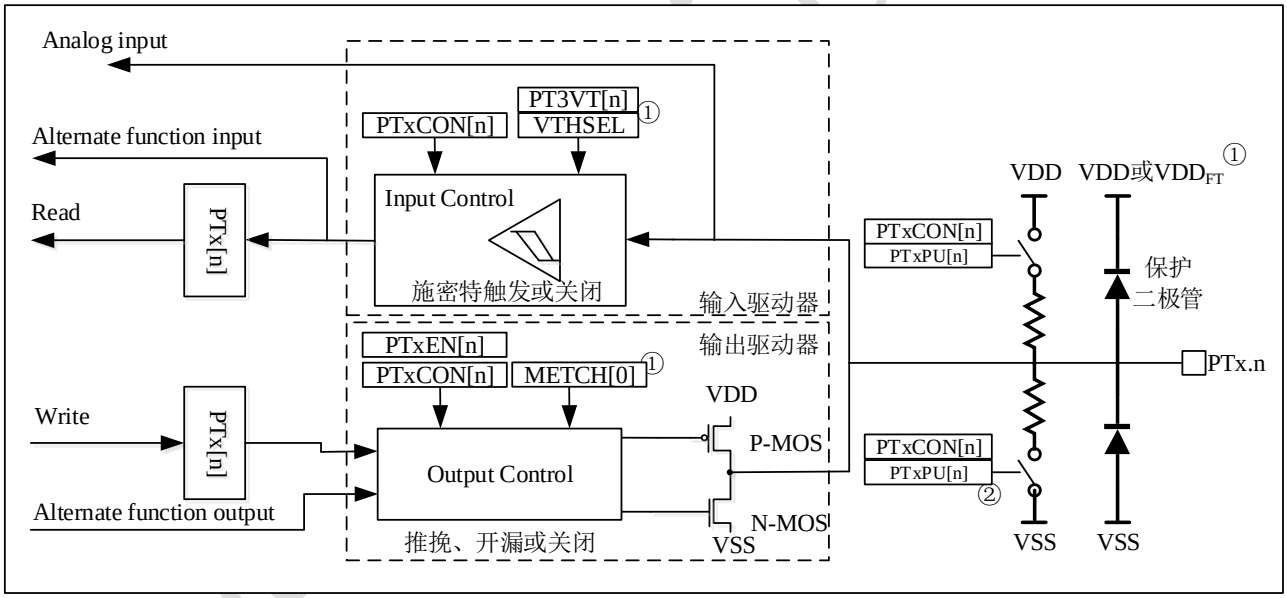


图 13　IO 端口基本结构

注：PT3VT[n]、METCH[0]、VDD_{FT} 仅适用于 PT3.1、PT3.2、PT3.3
　：仅 PT1.3 支持下拉电阻

1.6.2. I/O 口寄存器表

表 7　I/O 口寄存器表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
79h	METCH					METCH3	MCK1[1:0]		METCH 0	uuuu0100
20h	PT1			PT1[5:3]					PT1[0]	uuxxxuux
21h	PT1EN			T1EN[5:3]					T1EN[0]	uu000uu0

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
22h	PT1PU			PT1PU[5:3]					PT1PU[0]	uu000uu0
23h	PT1CON			PT1W[2:0]			E1M	E0M[1:0]		uu000000
28h	PT3	PT3VTH[1]	PT3[6:0]							uxxxxxxx
29h	PT3EN	PT3VTH[2]	PT3EN[6:0]							u0000000
2ah	PT3PU	PT3VTH[3]	PT3PU[6:0]							u0000000
2bh	PT3CON		PT3CON[6:0]							u0000000
2fh	CONFIG2	VTHSEL								0uuu0000
30h	PT5	PT5[7:5]				PT5[3:0]				xxxxxxxx
31h	PT5EN	PT5EN[7:5]				PT5EN[3:0]				00000000
32h	PT5PU	PT5PU[7:5]			PT1PD[4]	PT5PU[3:0]				00000000
33h	PT5CON	PT5CON[7:0]								11000000
41h	PT1CON1								PT1CON[0]	uuuuuuu0
42h	PT5CON1	PT5W[7:5]								000uuuuu

1.6.3. GPIO 上下拉电阻

CS8M32X 的所有 IO 都支持内部上拉电阻，通过寄存器 PT1PU、PT3PU、PT5PU 进行配置。当将 IO 配置为模拟口时，芯片内部上下拉电阻自动断开。而将 IO 配置为数字输出口时，上下拉电阻不会自动断开。

1.6.4. 输入逻辑电平电压配置

1.6.4.1. 施密特输入

CS8M32X 的 IO 均支持施密特触发输入，通过配置 VTHSEL 寄存器，可以使能或关闭施密特触发输入特性。

<table>
<tr><th>位地址</th><th>标识符</th><th>功能</th></tr>
<tr><td>7</td><td>VTHSEL</td><td>输入逻辑电平电压控制信号
<table>
<tr><td>VTHSEL</td><td colspan="5">输入逻辑电平</td></tr>
<tr><td rowspan="3">0</td><td>符号</td><td>参数</td><td>最小值</td><td>典型值</td><td>最大值</td><td>单位</td></tr>
<tr><td>VIH1</td><td>数字输入高电平</td><td>0.7VDD</td><td></td><td></td><td>V</td></tr>
<tr><td>VIL1</td><td>数字输入低电平</td><td></td><td></td><td>0.3VDD</td><td>V</td></tr>
<tr><td rowspan="3">1</td><td>符号</td><td>参数</td><td>最小值</td><td>典型值</td><td>最大值</td><td>单位</td></tr>
<tr><td>VIH2</td><td>数字输入高电平</td><td></td><td>0.5VDD</td><td></td><td>V</td></tr>
<tr><td>VIL2</td><td>数字输入低电平</td><td></td><td>0.5VDD</td><td></td><td>V</td></tr>
</table></td></tr>
</table>

1.6.4.2. 1.8V 逻辑电平

为满足与 1.8V 电压系统的通信要求，PT3.1/PT3.2/PT3.3 用作 GPIO 或复用作 UART/I2C 时均支持 1.8V 逻辑电平。通过设置对应的 PT3VTH 寄存器，可以使能或禁止此特性。

位地址	标识符	功能
	PT3VTH[1]	PT3.1 管脚 1.8V 电压选择位 1 = 管脚 PT3.1 支持 1.8V 电平传输 0 = 管脚 PT3.1 不支持 1.8V 电平传输
	PT3VTH[2]	PT3.2 管脚 1.8V 电压选择位 1 = 管脚 PT3.2 支持 1.8V 电平传输 0 = 管脚 PT3.2 不支持 1.8V 电平传输

	PT3VTH[3]	PT3.3 管脚 1.8V 电压选择位 1 = 管脚 PT3.3 支持 1.8V 电平传输 0 = 管脚 PT3.3 不支持 1.8V 电平传输

1.6.5. 防倒灌与开漏模式

为满足与更高电压系统的通信要求，PT3.1/PT3.2/PT3.3 使能防倒灌功能。通过设置 METCH0 寄存器为 1，断开上拉电阻，会使能防倒灌功能，并把 IO 强制设置为开漏模式。

设置 METCH0 寄存器为 0 时，PT3.1/PT3.2/PT3.3 禁止防倒灌功能。PT3.1/PT 3.2 用作 GPIO/I2C 则开漏输出，PT3.1/PT 3.2 用作 UART 则推挽输出。

位地址	标识符	功能
0	METCH0	PT3.1/PT3.2/PT3.3 IO 口防倒灌功能使能 1：使能防倒灌功能，IO 被强制设置为开漏模式 0：禁止防倒灌功能，IO 为正常模式(1)

注(1)：PT3.1/PT3.2 用作 GPIO/I2C 固定为开漏输出模式；用作 UART 时受 METCH0 寄存器控制，METCH0 为 0 则推挽输出，METCH0 为 1 则开漏输出。PT3.3 用作 GPIO 时受 METCH0 寄存器控制，METCH0 为 0 则推挽输出，METCH0 为 1 则开漏输出。

下表为 PT3.3 说明：

METCH [0]	PT3CON [3]	PT3PU [3]	模拟输入	开漏/推挽	防倒灌
1	1	X	AIN3	-	支持
1	0	1	-	开漏	不支持
1	0	0	-	开漏	支持
0	1	X	AIN3	-	不支持
0	0	X	-	推挽	不支持

下表为 PT3.1/3.2 说明（n=1,2；X 为任意值）：

METCH [0]	PT3CON [n]	PT3PU [n]	I2C_EN	I2C_SEL	UART0_EN	UART0_SEL	模拟输入	开漏/推挽	数字功能	防倒灌
1	1	X	X	X	X	X	AINn	-	-	支持
1	0	1	X	X	X	X	-	开漏	I2C/UART/IO	不支持
1	0	0	X	X	X	X	-	开漏	I2C/UART/IO	支持
0	1	X	X	X	X	X	AINn	-	-	不支持
0	0	X	1	1	X	X	-	开漏	I2C	不支持
0	0	X	X	0	1	0	-	推挽	UART	不支持
0	0	X	0	X	1	0	-	推挽	UART	不支持
0	0	X	X	0	0	X	-	开漏	IO	不支持
0	0	X	0	X	0	X	-	开漏	IO	不支持
0	0	X	X	0	X	1	-	开漏	IO	不支持
0	0	X	0	X	X	1	-	开漏	IO	不支持

注：METCH [0]=1 时，AINn 的输入范围会缩小，输入信号 VDD=2.4V，VIN＞0.5VDD；VDD=3.3V，VIN＞0.6VDD；VDD=5.5V，VIN＞0.7VDD 时，ADC 测量 AINn 信号会失真。

1.6.6.　PT1 口

1.6.6.1.　PT1 寄存器（地址为 20h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	R/W-X	R/W-X	R/W-X	U-0	U-0	R/W-X
PT1			PT1[5:3]					PT1[0]

位地址	标识符	功能
7:6	RESERVE	保留
5:3	PT1[5:3]	PT1 口 Bit5~Bit3 数据位 进行读操作时读取 GPIO 值 进行写操作配置 GPIO 口输出值。
2:1	RESERVE	保留
0	PT1[1:0]	PT1 口 Bit0 数据位 进行读操作时读取 GPIO 值 进行写操作配置 GPIO 口输出值。

1.6.6.2.　PT1EN 寄存器（地址为 21h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
PT1			PT1EN [5:3]					PT1EN[0]

位地址	标识符	功能
7:6	RESERVE	保留
5	PT1EN[5]	PT1.5 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
4	PT1EN[4]	PT1.4 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
3	PT1EN[3]	PT1.3 口输入输出控制位 0 = 定义为输入口 1 = 定义为开漏输出口
2:1	RESERVE	保留
0	PT1EN[0]	PT1.0 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口

1.6.6.3.　PT1PU 寄存器（地址为 22h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
PT1			PT1PU [5:3]					PT1PU[0]

位地址	标识符	功能
7:6	RESERVE	保留
5	PT1PU[5]	PT1.5 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
4	PT1PU[4]	PT1.4 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
3	PT1PU[3]	PT1.3 口上拉电阻使能位(1) 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
2：0	RESERVE	保留
0	PT1PU[0]	PT1.0 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开

注(1): PT1.3 的下拉功能配置在地址 32H 的第 4bit 控制。

1.6.6.4. PT1CON1 寄存器（地址为 41h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
PT1CON1								PT1CON[0]

位地址	标识符	功能
7:1	RESERVE	保留
0	PT1CON[0]	GPIO1Bit 0 的 I/O 控制位 0 = 定义为数字口 1 = 定义为模拟口

特性（Property）：
R = 可读位　　W = 可写位　　U = 无效位
-n = 上电复位后的值　‘1’ = 位已设置　‘0’ = 位已清零　　X = 不确定位

1.6.7. PT3 口

特殊说明：

1、在开代码选项中开启 ICD 调试使能时，PT34 一直是数字输入口，不受 PT34 控制寄存器控制。

2、在 PT3.1 和 PT3.2 做 GPIO 和 I2C 功能时为开漏模式，做 UART 模式时则由 METCH0 设置推挽/开漏模式。

1.6.7.1. PT3 寄存器（地址为 28H）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

特性	R/W-0	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
PT3	PT3VTH[1]	PT3[6:0]						

位地址	标识符	功能
7	PT3VTH[1]	管脚 1.8V 电压选择位 1 = 管脚 PT3.1 支持 1.8V 电平传输 0 = 管脚 PT3.1 不支持 1.8V 电平传输
6：0	PT3[6:0]	PT3 口 Bit6~Bit0 数据位 进行读操作时读取 GPIO 值 进行写操作配置 GPIO 口输出值。

1.6.7.2. PT3EN 寄存器（地址为 29h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	W-0	W-0	R/W-0
PT3EN	PT3VTH[2]	PT3EN[6:0]						

位地址	标识符	功能
7	PT3VTH[2]	管脚 1.8V 电压选择位 1 = 管脚 PT3.2 支持 1.8V 电平传输 0 = 管脚 PT3.2 不支持 1.8V 电平传输
6	PT3EN[6]	PT3.6 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
5	PT3EN[5]	PT3.5 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
4	PT3EN[4]	PT3.4 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
3	PT3EN[3]	PT3.3 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口 注：METCH0=1 设置为开漏输出模式
2	PT3EN[2]	PT3.2 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口 注：该位不可读；普通 IO 输出为开漏模式，不受 METCH0 影响
1	PT3EN[1]	PT3.1 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口 注：该位不可读；普通 IO 输出为开漏模式，不受 METCH0 影响
0	PT3EN[0]	PT3.0 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口

1.6.7.3. PT3PU 寄存器（地址为 2ah）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT3PU	PT3VTH[3]	PT3PU[6:0]						

位地址	标识符	功能
7	PT3VTH[3]	管脚 1.8V 电压选择位 1 = 管脚 PT3.3 支持 1.8V 电平传输 0 = 管脚 PT3.3 不支持 1.8V 电平传输
6	PT3PU[6]	PT3.6 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
5	PT3PU[5]	PT3.5 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
4	PT3PU[4]	PT3.4 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
3	PT3PU[3]	PT3.3 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
2	PT3PU[2]	PT3.2 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
1	PT3PU[1]	PT3.1 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
0	PT3PU[0]	PT3.0 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开

1.6.7.4. PT3CON 寄存器（地址为 2bh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT3CON		PT3CON[6:0]						

位地址	标识符	功能
7	RESERVE	保留
6	PT3CON[6]	PT3.6 口控制位 0 = 定义为数字口 1 = 定义为模拟口
5	PT3CON[5]	PT3.5 口控制位 0 = 定义为数字口 1 = 定义为模拟口
4	PT3CON[4]	PT3.4 口控制位 0 = 定义为数字口 1 = 定义为模拟口
3	PT3CON[3]	PT3.3 口控制位 0 = 定义为数字口 1 = 定义为模拟口
2	PT3CON[2]	PT3.2 口控制位 0 = 定义为数字口 1 = 定义为模拟口
1	PT3CON[1]	PT3.1 口控制位 0 = 定义为数字口 1 = 定义为模拟口
0	PT3CON[0]	PT3.0 口控制位 0 = 定义为数字口 1 = 定义为模拟口

1.6.8. PT5 口

1.6.8.1. PT5 寄存器（地址为 30h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-X	R/W-X	R/W-X	U-0	R/W-X	R/W-X	R/W-X	R/W-X
PT5	PT5[7:5]				PT5[3:0]			

位地址	标识符	功能
7：5	PT5[7:5]	PT5 口 Bit7~Bit5 数据位 进行读操作时读取 GPIO 值 进行写操作配置 GPIO 口输出值。
4	RESERVE	保留
3：0	PT5[3:0]	PT5 口 Bit3~Bit0 数据位 进行读操作时读取 GPIO 值 进行写操作配置 GPIO 口输出值。

1.6.8.2. PT5EN 寄存器（地址为 31h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PT5EN	PT5EN[7:5]				PT5EN[3:0]			

位地址	标识符	功能
7	PT5EN[7]	PT5.7 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
6	PT5EN[6]	PT5.6 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
5	PT5EN[5]	PT5.5 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
4	RESERVE	保留
3	PT5EN[3]	PT5.3 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
2	PT5EN[2]	PT5.2 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
1	PT5EN[1]	PT5.1 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
0	PT5EN[0]	PT5.0 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口

1.6.8.3. PT5PU 寄存器（地址为 32h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT5PU	PT5PU[7:5]			PT1PD[3]	PT5PU[3:0]			

位地址	标识符	功能
7	PT5PU[7]	PT5.7 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
6	PT5PU[6]	PT5.6 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
5	PT5PU[5]	PT5.5 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
4	PT1PD[3]	PT1.3 口下拉电阻使能位

位地址	标识符	功能
		0 = 断开下拉电阻 1 = 使能下拉电阻
3	PT5PU[3]	PT5.3 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
2	PT5PU[2]	PT5.2 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
1	PT5PU[1]	PT5.1 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
0	PT5PU[0]	PT5.0 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开

1.6.8.4. PT5CON 寄存器（地址为 33h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-1	R/W-1	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PT5CON	PT5CON[7:0]							

位地址	标识符	功能
7	PT5CON[7]	PT5.7 口控制位 0 = 定义为数字口 1 = 定义为模拟口
6	PT5CON[6]	PT5.6 控制位 0 = 定义为数字口 1 = 定义为模拟口
5	PT5CON[5]	PT5.5 控制位 0 = 定义为数字口 1 = 定义为模拟口
3	PT5CON[3]	PT5.3 口控制位 0 = 定义为数字口 1 = 定义为模拟口
2	PT5CON[2]	PT5.2 口控制位 0 = 定义为数字口 1 = 定义为模拟口
1	PT5CON[1]	PT5.1 口控制位 0 = 定义为数字口 1 = 定义为模拟口

0	PT5CON[0]	PT5.0 口控制位 0 = 定义为数字口 1 = 定义为模拟口

特性（Property）：
R = 可读位　　W = 可写位　　U = 无效位
-n = 上电复位后的值　‘1’ = 位已设置　‘0’ = 位已清零　　X = 不确定位

2. 增强功能

2.1. HALT 和 SLEEP 模式

CS8M32X 支持低功耗工作模式。为了使 CS8M32X 处于待机状态，可以让 CPU 停止工作使 CS8M32X 进入停止模式或睡眠模式，减低功耗。这两种模式描述如下：

停止模式

CPU 执行停止指令后，程序计数器停止计数直到出现中断事件。停止模式下，芯片内部高速振荡器和内部 32KHz WDT 时钟仍然正常工作，内核时钟停止，定时器可以正常计数，ADC 仍然可以继续进行未完成的转换。

睡眠模式

CPU 执行睡眠指令后，低速振荡器打开。高速振荡器根据 CVC 检测需要定时打开，其他时间高速振荡器停止工作直到出现一个中断事件唤醒 CPU。在睡眠模式下的功耗大约小于 6μA。

为了保证 CPU 在睡眠模式下的功耗最小，在执行睡眠指令之前，需要保证所有的输入口是接到 VDD 或 VSS 电平。ADC 模块在 SLEEP 模式下只有在 CVC 定时检测时才打开，其他时间全程关闭。在 SLEEP 模式下，模拟输入通道不能配置为 1/8VDD（SRADCON2 寄存器的 CHS[3:0]位不能配置为 1100B）。

注：
芯片如果处于 SLEEP 状态，这时候降低电压，配置的低电压复位不会起作用，低于 POR 掉电复位点才会复位。如果 SLEEP 唤醒后，此时还处于低电压复位点以下，则会立即复位。

HALT 示范程序：

```
…
MOVLW 08h
MOVWF pt1up      ;断开 pt1 除 bit3(pt1[3])外的其他接口的上拉电阻
MOVLW 000h
MOVWF pt1en      ;pt1 口做输入口
MOVLW 000h
MOVWF pt1con1    ;pt1.1 口做数字口
CLRF pt3up       ;断开 pt3 上拉电阻
CLRF pt3en       ;pt3 口用作输入口
CLRF pt3con      ;pt3 口用作数字口
CLRF pt3         ;将 pt3 输出为低
CLRF pt5up       ;断开 pt5 上拉电阻
CLRF pt5en       ;pt5 口用作输入口
CLRF pt5con      ;pt5 口用作数字口
CLRF pt5         ;将 pt5 输出为低
CLRF intf        ;清除中断标志位
CLRF intf2       ;清除中断标志位
CLRF ur0_intf    ;清除 UART 中断标志位
CLRF i2c_intf    ;清除 I2C 中断标志位

MOVLW 81h
MOVWF inte       ;使能外部中断 0
halt             ;进入停止模式
…
```

SLEEP 示范程序:

```
…
MOVLW 08h
MOVWF pt1up      ;断开 pt1 除 bit3(pt1[3])外的其他接口的上拉电阻
MOVLW 000h
MOVWF pt1en      ;pt1 口做输入口
MOVLW 000h
MOVWF pt1con1    ;pt1.1 口做数字口

CLRF pt1         ;将 pt1[4:1]输出为低
CLRF pt3up       ;断开 pt3 上拉电阻
CLRF pt3en       ;pt3 口用作输入口
CLRF pt3con      ;pt3 口用作数字口
CLRF pt3         ;将 pt3 输出为低
CLRF pt5up       ;断开 pt5 上拉电阻
CLRF pt5con      ;pt5 口用作数字口
CLRF pt5en       ;pt5 口用作输入口
CLRF pt5         ;将 pt5 输出为低
CLRF intf        ;清除中断标志位
CLRF intf2       ;清除中断标志位
CLRF ur0_intf    ;清除 UART 中断标志位
CLRF i2c_intf    ;清除 I²C 中断标志位
MOVLW 81h
MOVWF inte       ;使能外部中断 0
sleep            ;进入睡眠模式
…
```

2.2. 看门狗(WDT)

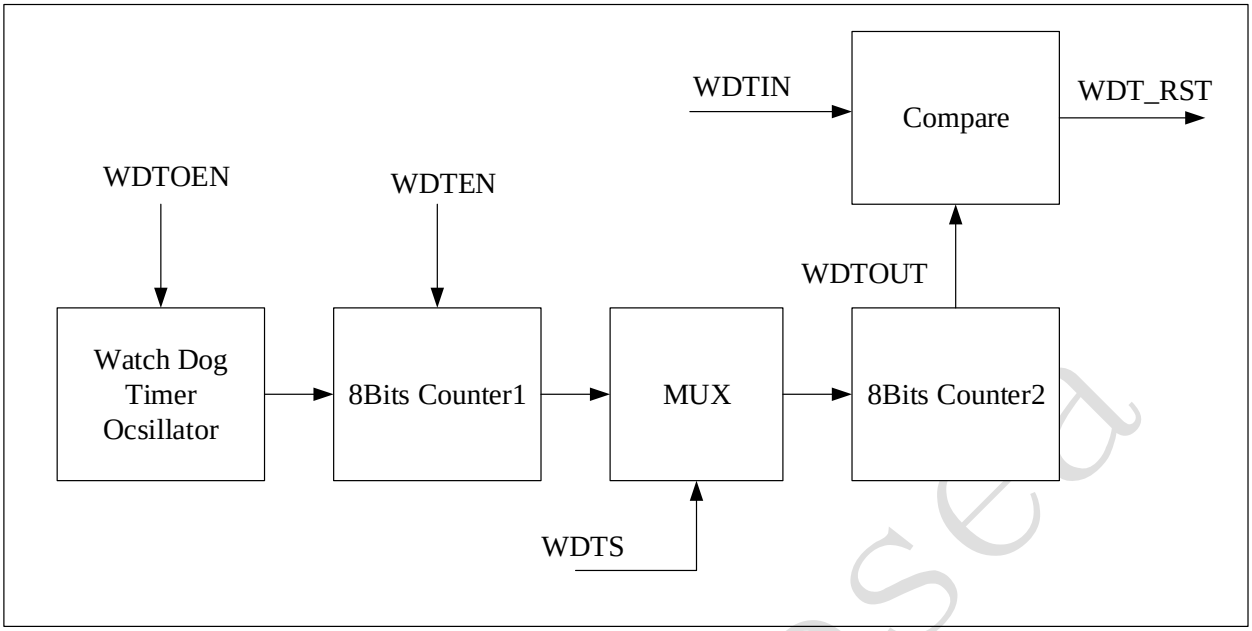


图 14　看门狗定时器功能框图

看门狗定时器（WDT）用于防止程序由于某些不确定因素而失去控制。当 WDT 启动时，WDT 计时超时后将使 CPU 复位。在运行的程序一般在 WDT 复位 CPU 之前先清除 WDT 计数值。当出现某些故障时，程序没有清除 WDT 计数值，而 WDT 计时超时后将 CPU 复位到正常状态下。

WDT 模块使能和 32KHz WDT 时钟使能受代码选项 WDT_CFG 控制，当该位置 0 时，WDT 模块使能和 32KHz WDT 时钟使能固定打开，软件无法关闭；当该位置 1 时，WDT 模块使能有 WDTEN 控制，32KHz WDT 时钟使能由 CST_WDT 控制。

当用户把 CST_WDT 清 0 时，则内部的看门狗定时器振荡器（32KHz）将会启动，产生的时钟被送到 8 位预分频计数器。当用户置位 WDTEN 时，8 位预分频计数器开始计数，预分频时钟通过 WDTS[2:0]控制的多路选择器进行选择，得到 WDT 计数器时钟。当 8 位计数器计数值与 WDTIN 数值相等时溢出，溢出时它会发送 WDTOUT 信号复位 CPU 及置位 TO 标志位。用户可以使用指令 CLRWDT 复位 WDT。

2.2.1. 看门狗定时器寄存器表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
04H	STATUS					TO				xuu00000
0DH	WDTCON	WDTEN					WDTS[2:0]			0uuuu000
0Eh	WDTIN	WDTIN[7:0]								11111111

2.2.1.1. WDTCON 寄存器（地址为 0Dh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
WDTCON	WDTEN					WDTS[2:0]		

位地址	标识符	功能
7	WDTEN	看门狗使能位，高电平有效。当 WDT_CFG 配置为 WDT 模块固定打开时，该位无效。

位地址	标识符	功能
		0：关闭 WDT 模块 1：使能 WDT 模块
6:3	保留	保留
2：0	WDTS[2:0]	WDT 计数时钟分频 WDTS [2:0] / WDT 计数时钟 000 / WDTCLK /256 001 / WDTCLK /128 010 / WDTCLK /64 011 / WDTCLK /32 100 / WDTCLK /16 101 / WDTCLK /8 110 / WDTCLK /4 111 / WDTCLK /2

2.2.1.2. WDTIN 寄存器（地址为 0EH）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
WDTIN	WDTIN[7:0]							

位地址	标识符	功能
7：0	WDTIN[7:0]	WDT 计数输入，默认值为 FFh 该寄存器值不能写入 00h，如果写入 00h，则 WDT_IN[7:0]的值会被硬件修改为 FFh。写入除 00h 以外的其他值则不受影响。

2.2.2. WDT 定时器功能

定时器的时钟源只能是内部 32K WDT 时钟。当要使能 WDT 定时器时，必须先将 MCK 寄存器的 CST_WDT 位清零，使能内部 32K WDT 时钟，即使在 SLEEP 模式下，WDT 定时器仍然可以正常工作。

操作步骤如下：

1. 设置 WDTCON 寄存器的 WDTS[3:0]位，选择 WDT 时钟频率。
2. 设置 WDTIN，选择不同的溢出时间值
3. 使能 32K WDT 时钟：把 MCK 寄存器的 CST_WDT 位清 0，打开 WDT 的晶振。
4. 使能 WDT 定时器：将 WDTCON 寄存器的 WDTEN 位置 1，使能 WDT。
5. 在程序中执行 CLRWDT 指令清除 WDT 计数值。

WDT 溢出时间计算公式：

$$溢出时间=\frac{2^{(8-WDTS[2:0])}}{32k}*(WDTIN[7:0]+1)$$

WDTS[2:0]范围为 0~7，WDTIN[7:0]范围为 0~255。

WDTS[2:0]	计数器时钟	时间（当 WDTIN==FFH）
000	WDTA [0]	2048ms
001	WDTA [1]	1024ms
010	WDTA [2]	512ms

WDTS[2:0]	计数器时钟	时间（当 WDTIN==FFH）
011	WDTA [3]	256ms
100	WDTA [4]	128ms
101	WDTA [5]	64ms
110	WDTA [6]	32ms
111	WDTA [7]	16ms

2.3. 定时/计数器 2

2.3.1. 定时/计数器 2 概述

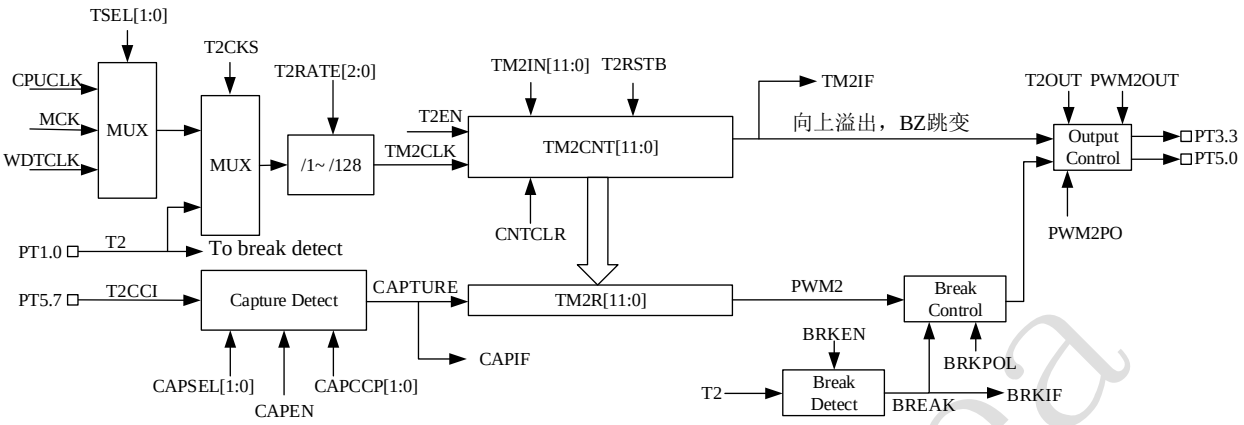


图 15　定时/计数器 2 模块的功能框图

定时/计数器 2 模块的输入时钟是 TM2CLK。当用户置位定时/计数器 2 模块的使能，12 Bits 计数器将启动，从 00h 递增到 TM2IN。用户需要设置 TM2IN（定时器模块计数溢出值）以选择定时超时时间，超时后将产生中断信号。

当定时超时发生时，定时器 2 中断标志位 TM2IF 硬件置 1，该位只能通过软件清零。如果使能了定时器 2 中断（TM2IE=1）和中断总使能（GIE），程序计数器会跳转到 004H 以执行中断服务程序。

主要功能：

1） 12 位可编程定时器；
2） 中断上报功能；
3） PWM2 输出；
4） 刹车输入；
5） 支持捕获功能

2.3.2. 寄存器描述

表 8　定时器寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06h	INTF	BRKIF	TM2IF	CAPIF	TM0IF	SRADIF		E1IF	E0IF	00000u00
07h	INTE	GIE	TM2IE	CAPIE	TM0IE	SRADIE		E1IE	E0IE	00000u00
15h	TM2CNTH	CAPEN	CAPCCP[1:0]		CNTCLR	CAPSEL[1:0]		BRKEN	BRKPOL	00000000
17h	TM2CON	T2EN	T2RATE[2:0]			T2CKS	T2RSTB	T2OUT	PWM2OUT	00000100
18h	TM2IN	TM2IN[7:0]								11111111
19h	TM2CNT	TM2CNT[7:0]								00000000
1ah	TM2R	TM2R[7:0]								00000000
24h	TM2INH					TM2IN[11:8]				uuuu0000
25h	TM2CN					TM2CNT[11:8]				uuuu0000

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
	TH									
26h	TM2RH					TM2R[11:8]				uuuu0000
2eh	CONFIG1					PWM2STALL	PWM2PO			00u0000u
0Fh	TM0CON							TSEL[1:0]		0000u100

2.3.2.1. TM2CON 寄存器（地址 17h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0
TM2CON	T2EN	T2RATE[2:0]			T2CKS	T2RSTB	T2OUT	PWM2OUT

位地址	标识符	功能
7	T2EN	定时/计数器 2 使能位 1：使能定时器 2 0：禁止定时器 2
6:4	T2RATE[2:0]	定时/计数器 2 时钟选择 T2RATE [2:0] / TM2CLK 000 / CKT2 001 / CKT2 /2 010 / CKT2 /4 011 / CKT2 /8 100 / CKT2 /16 101 / CKT2 /32 110 / CKT2 /64 111 / CKT2 /128
3	T2CKS	定时/计数器 2 时钟源选择位 1：PT1.0 作为时钟 0：CPUCLK 或 MCK 或 WDT 的分频时钟
2	T2RSTB	定时/计数器 2 复位 1：禁止定时/计数器 2 复位 0：使能定时/计数器 2 复位 当将该位为 0 时，定时器 2 复位后，T2RSTB 会自动置 1
1	T2OUT	PWM/蜂鸣器输出控制，输出 IO 由 PWM2PO、T2OUT / PWM2OUT / PWM/蜂鸣器输出控制，仅当对应 IO 口配置为输出有效
0	PWM2OUT	0 / 0 / 不做 PWM/蜂鸣器输出 0 / 1 / PWM2 输出 1 / 0 / 蜂鸣器输出 1 / 1 / PWM2 输出

位地址	标识符	功能
		PT5.0 可以输出蜂鸣器输出和 PWM 输出，而 PT3.3 只能输出 PWM。所以当以上寄存器配置为蜂鸣器输出，而 PWM2PO 选择的 IO 口不是 PT5.0 口，则 PT3.3 口都没有输出。

2.3.2.2. TM2IN 寄存器（地址 18h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TM2IN	TM2IN[7:0]							

位地址	标识符	功能
7 ： 0	TM2IN[7:0]	定时/计数器溢出值低 8 位

2.3.2.3. TM2INH 寄存器（地址 24h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TM2INH					TM2IN[11:8]			

位地址	标识符	功能
3 ： 0	TM2INH[11:8]	定时/计数器溢出值高 4 位

2.3.2.4. TM2CNT 寄存器（地址 19h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM2CNT	TM2CNT[7:0]							

位地址	标识符	功能
7 ： 0	TM2CNT[7:0]	定时/计数器 2 计数寄存器低 8 位，**只读**

2.3.2.5. TM2CNTH 寄存器（地址 25h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	U-0	R-0	R-0	R-0	R-0
TM2CNTH					TM2CNT[11:8]			

位地址	标识符	功能
3 ： 0	TM2CNTH[11:8]	定时/计数器 2 计数寄存器高 4 位，**只读**

2.3.2.6. TM2R 寄存器（地址 1ah）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TM2R	TM2R[7:0]							

位地址	标识符	功能
7 ： 0	TM2R[7:0]	定时/计数器 2 的 PWM 高电平占空比控制寄存器低 8 位。

2.3.2.7. TM2RH 寄存器（地址 26h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TM2RH					TM2R[11:8]			

位地址	标识符	功能
3 ： 0	TM2RH[11:8]	定时/计数器 2 的 PWM 高电平占空比控制寄存器 高 4 位。

2.3.2.8. TM2CAP 寄存器（地址 15h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TM2CAP	CAPEN	CAPCCP[1:0]		CNTCLR	CAPSEL[1:0]		BRKEN	BRKPOL

<table>
<tr><th>位地址</th><th>标识符</th><th>功能</th></tr>
<tr><td>7</td><td>CAPEN</td><td>定时器/计数器 2　定时功能和捕获功能选择位
0：定时器/计数器功能
1：输入捕获功能
当配置为输入捕获功能时， TM2R/TM2RH 复用为下降沿捕获值。</td></tr>
<tr><td>6:5</td><td>CAPCCP[1:0]</td><td>捕获极性选择位
00：周期捕获，上升沿~上升沿
01：周期捕获，下降沿~下降沿
10：高电平，上升沿~下降沿
11：低电平，下降沿~上升沿</td></tr>
<tr><td>4</td><td>CNTCLR</td><td>发生捕获事件时，清除计数器控制信号
0：发生捕获事件时，硬件不清除计数器
1：发生捕获事件时，硬件自动清除计数器
当该位配置为 1 时，发生捕获事件时，当前计数值被保存到 TM2R/TM2RH 寄存器，计数器清零，重新开始计数。
当该位配置为 0 时，发生捕获事件时，当前计数值被保存到 TM2R/TM2RH 寄存器，计数器不清零。</td></tr>
<tr><td>3:2</td><td>CAPSEL[1:0]</td><td>捕获输入选择位
<table>
<tr><th>CAPSEL[1:0]</th><th>捕获输入</th></tr>
<tr><td>00</td><td>捕获输入通道 CCI0</td></tr>
<tr><td>01</td><td>捕获输入通道 CCI1</td></tr>
<tr><td>10</td><td>捕获输入通道 CCI0 的值反相</td></tr>
<tr><td>11</td><td>捕获输入通道 CCI1 的值反相</td></tr>
</table></td></tr>
<tr><td>1</td><td>BRKEN</td><td>刹车功能使能位
0：关闭刹车功能
1：打开刹车功能</td></tr>
<tr><td>0</td><td>BRKPOL</td><td>刹车极性选择位
0：当刹车使能并且刹车输入通道为高电平时，PWM 输出低电平</td></tr>
</table>

位地址	标识符	功能
		1：当刹车使能并且刹车输入通道为高电平时，PWM 输出高电平

2.3.2.9. CONFIG1 寄存器（地址 2eh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
CONFIG1					PWM2STALL	PWM2PO		

位地址	标识符	功能
3	PWM2STALL	ICD 调试 STALL 时，定时器 2 单端 PWM 输出电平控制位 1：ICD 调试 STALL 时，定时器 2 单端 PWM 输出高电平 0：ICD 调试 STALL 时，定时器 2 单端 PWM 输出低电平
2	PWM2PO	PWM2 输出脚选择 PWM2PO：0 — PT5.0 做为 PWM2 输出口；1 — PT3.3 做为 PWM2 输出口

2.3.2.10. TM0CON 寄存器(地址 0Fh)

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
TM0CON							TSEL[1:0]	

位地址	标识符	功能
1:0	TSEL[1:0]	时钟源选择 T0SEL[1:0]：00 — CPUCLK；01 — MCK；1x — 内部 32K WDT 时钟，**仅当内部 WDT 晶振打开时有效**

2.3.3. 时钟选择

定时器 2 模块的输入时钟通过 TSEL[1:0]进行选择，可选时钟源有 CPUCLK、MCK、内部 32K WDT 或者 PT1.0 外部输入时钟。值得注意的是，当 T2CKS 为 1 时，定时器 0，定时器 2 和定时器 3 的时钟源是一致的，具体如下图所示：

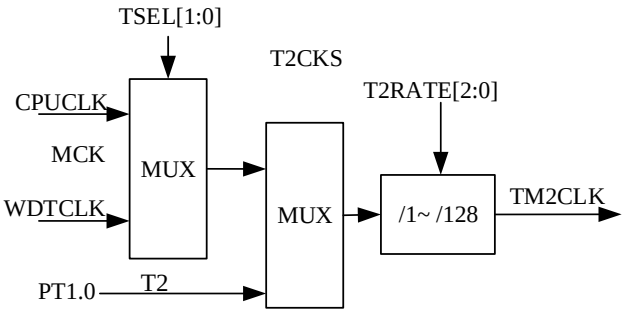


图 16　TM2 定时器时钟选择

2.3.4.　定时/计数器功能

定时/计数器 2 模块的输入时钟是 TM2CLK。当用户置位定时/计数器 2 模块的使能，12 bits 计数器将启动，从 000h 递增到 TM2IN。用户需要设置 TM2IN（定时器模块计数溢出值）以选择定时超时时间，超时后将产生中断信号。

当定时超时发生时，定时器 2 中断标志位 TM2IF 硬件置 1，该位只能通过软件清零。如果使能了定时器 2 中断（TM2IE=1）和中断总使能（GIE 为内核寄存器信号），程序计数器会跳转到 004H 以执行中断服务程序。

定时器 2 的时钟源可以选择 CPUCLK、MCK、WDTCLK，在 HALT 模式下定时器 2 仍然可以继续计数，因此可以唤醒 HALT 模式。SLEEP 模式下内部高速振荡器停止工作，定时器 2 是能在 WDTCLK 时钟工作。

操作：

1) 配置 TM0CNT 的 TSEL[1:0]，为定时器模块选择时钟源。
2) 配置 TM2CON 寄存器的 T2RATE[2:0]，为定时器 2 选择时钟分频。
3) 设置 TM2IN[11:0]，选择定时器溢出值
4) 清除定时器中断标志位：将 TM2IF 位清零
5) 置位寄存器位 TM2IE 与 GIE，使能定时器中断。
6) 清零寄存器位 T2RSTB，复位定时器模块的计数器。
7) 置位寄存器位 T2EN，使能定时器模块的 12 Bits 计数器。
8) 当定时超时发生时，程序计数器会跳转到 004H。

定时器 2 溢出时间计算方法：

定时器 2 溢出时间=（TM2IN[11:0]+1）/TM2CLK.　（TM2IN 不为 0）

2.3.5.　刹车功能

刹车源来自于外部引脚 PT1.0 的 T2。发生刹车时，PWM 输出源由原来的 PWM 电路输出转到刹车极性寄存器输出。直到关闭刹车使能或者外部刹车源改变为止。

操作：

1) 配置 BRKEN=1
2) 配置 BRKPOL，为刹车选择输出极性，该极性在刹车功能有效时，直接通过 PWM 输出脚输出。
3) 当 PT1.0 管脚产生低电平时，刹车功能有效。

4) BRKIF 状态标志位置高。在刹车有效的整个过程中，BRKIF 无法清零。当刹车功能无效时，该状态由软件清零。上电时默认为低电平。

2.3.6. PWM

PWM 输出优先级

定时器 2 的 PWM 输出优先级从上到下递减

寄存器配置					PWM 引脚功能
PT5EN[0]	P3L2OEN	T2OUT	PWM2OUT	PWM2PO	PT5.0
0	X	X	X	X	输入
1	1	X	X	X	互补模式输出 PWM3L2
1	0	X	1	0	PWM2 输出
1	0	1	0	0	蜂鸣器输出
1	0	X	X	1	普通 IO

寄存器配置			PWM 引脚功能
PT3EN[3]	PWM2OUT	PWM2PO	PT3.3
0	X	X	输入
1	1	1	PWM2 输出
1	1	0	普通 IO 输出
1	0	1	普通 IO 输出
1	0	0	普通 IO 输出

配置 PWM 输出步骤：

1) 配置 PT5.0 为输出口。
2) 配置 TM2CON 寄存器的低 2 位，T2OUT 设置为 0，PWM2OUT 设置为 1
3) 配置 TM0CON 的 TSEL[1:0]，为定时器模块选择时钟源。
4) 配置 TM2CON 寄存器的 T2RATE[2:0]，为定时器 2 选择时钟分频。
5) 设置 TM2IN[11:0]来配置 PWM2 的周期。
6) 设置 TM2R[11:0]来配置 PWM2 的高电平的脉宽。
7) 配置 CONFIG1 的 PWM2PO 为 0，即配置 PT5.0 为 PWM 输出端口，之后把 T2EN 置 1 启动定时器。
8) PWM 从 PT5.0 输出。

周期为 TM2IN+1，高电平脉宽为 TM2R。如 TM2IN=0x0F，TM2R=0x03 的 PWM2 波形输出如下：

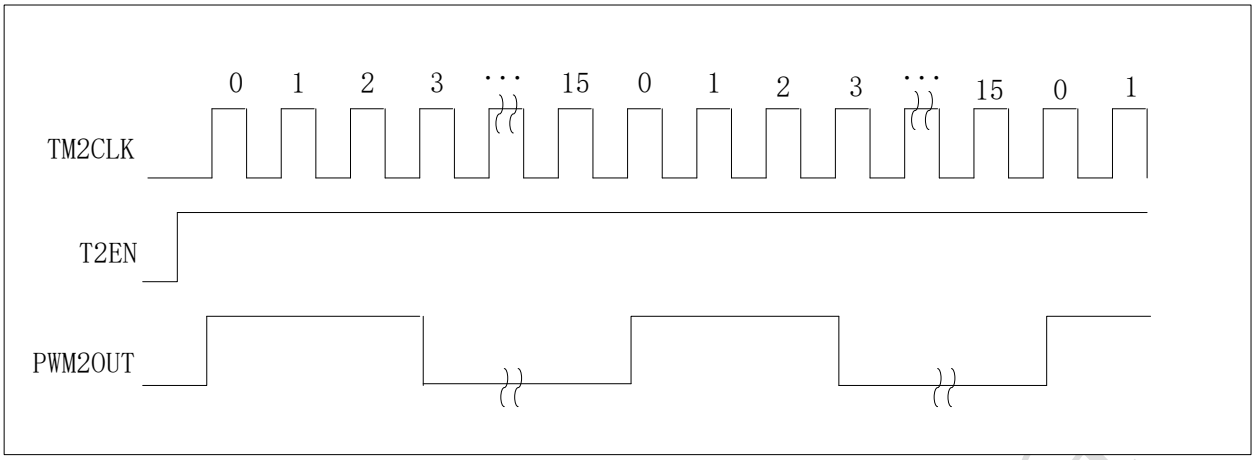


图 17　PWM2 输出波形图

2.3.7.　捕获功能

捕获模式下可以测量输入脉冲信号高低电平或者周期宽度。在使能捕获模式后，计数器在第一个有效边沿将计数器值初始化为 000h，在第二个有效边沿停止计数并将当前计数值存入 TM2RH/TM2R，置位捕获中断标志位 CAPIF。同时计数器值重新初始化为 000h，开始下一次计数。如果在第二次有效边沿发生前，定时器发生溢出，那么定时器溢出中断标志位 TM2IF 会置位。

输入捕获波形

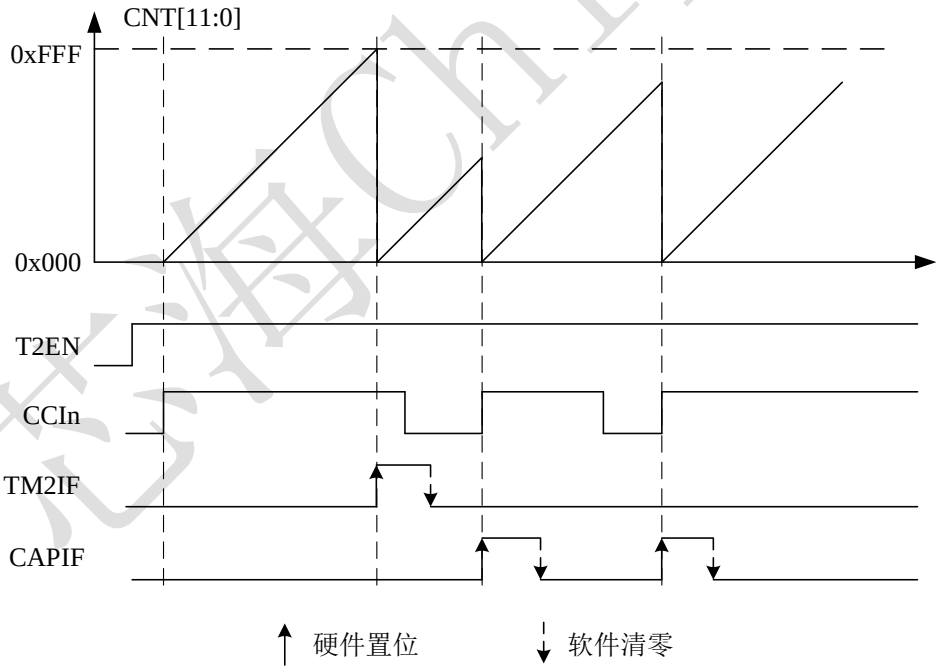


图 18　上升沿到上升沿测量捕获图

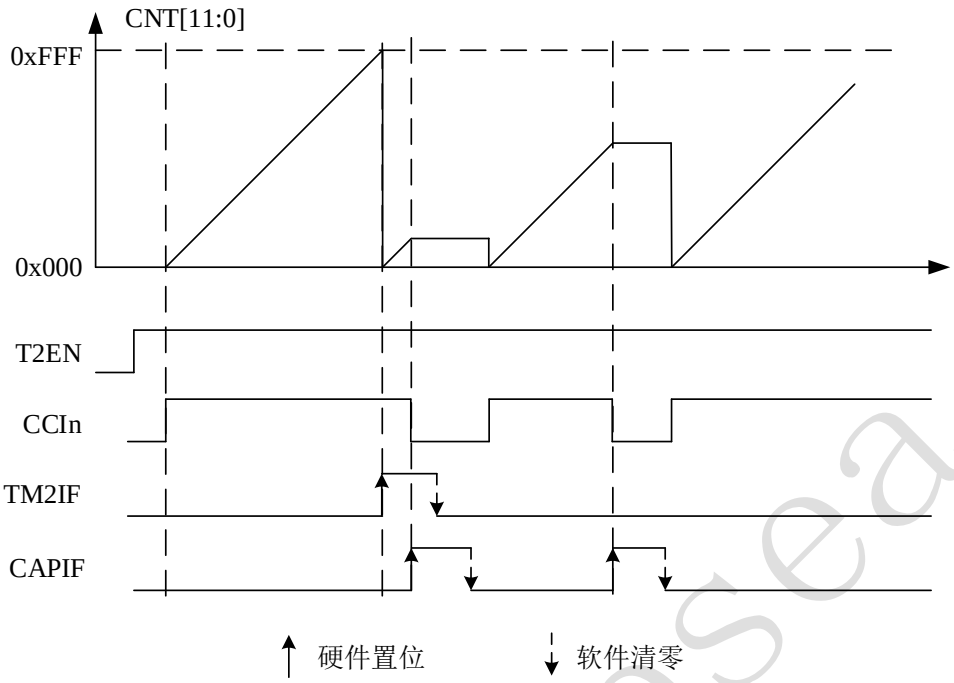


图 19　上升沿到下降沿测量捕获图

2.3.7.1.　输入捕获功能引脚分频

输入捕获引脚通过 CAPSEL[2:0]进行选择，捕获输入引脚分配如下：

CAPSEL[1:0]	捕获输入通道	对应引脚
00	捕获输入通道 CCI0	PT5.7
01	捕获输入通道 CCI1	PT5.7
10	捕获输入通道 CCI2	PT5.7 电平取反
11	捕获输入通道 CCI3	PT5.7 电平取反

2.4. 定时/计数器 3

2.4.1. 定时/计数器 3 概述

定时器 3 包含一个 12 位递增和递增/递减计数器，由 7 位预分频器提供驱动，支持多个 PWM 输出通道，可以产生蜂鸣器(BZ)、PWM、互补 PWM 等波形。

主要功能：

1） 12 位递增和递增/递减计数器，支持 8 位或者 12 位定时器/计数器模式
2） 7 位预分频器，支持 1/2/4/8/16/32/64/128 分频
3） 蜂鸣器(BZ)输出功能
4） 3 通道 8 位或者 2 通道 12 位 PWM 输出，
5） 支持带可编程死区的互补输出
6） 8 位模式下支持中心对齐模式；

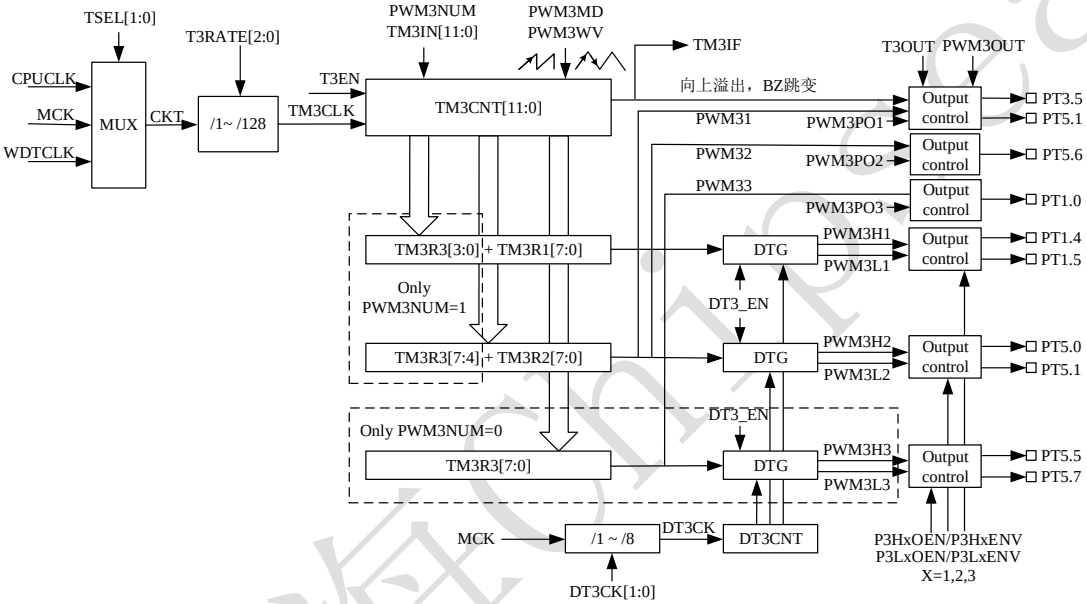


图 20　定时/计数器 3 模块的功能框图

2.4.2. 寄存器描述

表 9　定时器寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
3ch	INTF2				TM3IF					0000u000
3dh	INTE2				TM3IE					u000u0uu
1bh	TM3CON	T3EN	T3RATE[2:0]				T3RSTB	T3OUT	PWM3CON	00000100
1ch	TM3IN	TM3IN[7:0]								11111111
1dh	TM3CNT	TM3CNT[7:0]								00000000
1eh	TM3R1	TM3R1[7:0]								00000000
3bh	TM3R2	TM3R2[7:0]								00000000
1fh	TM3INH					TM3IN[11:8]				uuuu0000
25h	TM2CNTH	TM3CNT_DT1[2:0]								000u0000
26h	TM2RH	TM3CNT_DT2[2:0]								000u0000

27h	TM3CNTH	TM3CNT_DT3[2:0]				TM3CNT[11:8]				uuuu0000
2ch	TM3R3	TM3R3[7:0]								00000000
2dh	TM3CON2	DT3CK[1:0]		DT3CNT[2:0]			DT3_EN	P3H1OEN	P3L1OEN	00000000
2eh	CONFIG1	P3H1INV	P3L1INV		PWM3STALL			PWM3PO1		00u0000u
2f	CONFIG2		PWM3MD	PWM3NUM	PWM3WV			PWM3PO3	PWM3PO2	0000uu00
45H	TM3CON3	P3H3INV	P3L3INV	P3H3OEN	P3L3OEN	P3H2INV	P3L2INV	P3H2OEN	P3L2OEN	00000000

2.4.2.1. TM3CON 寄存器（地址为 1bh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0
TM3CON	T3EN	T3RATE[2:0]			RSV	T3RSTB	T3OUT	PWM3OUT

位地址	标识符	功能
7	T3EN	定时/计数器 3 使能位 1：使能定时器 3 0：禁止定时器 3
6:4	T3RATE[2:0]	定时/计数器 3 时钟分频选择，与 CONFIG2 寄存器的 T3RATE[3]合并后组成 4 位时钟选择寄存器。 T3RATE [2:0] / TM3CLK 000 / CKT3 001 / CKT3/2 010 / CKT3/4 011 / CKT3/8 100 / CKT3/16 101 / CKT3/32 110 / CKT3/64 111 / CKT3/128
3	RSV	保留，固定为低电平
2	T3RSTB	定时/计数器 3 复位 1：禁止定时/计数器 3 复位 0：使能定时/计数器 3 复位 当将该位配置为 0，一个指令周期后定时器 3 复位完成，T3RSTB 硬件置 1
1	T3OUT	蜂鸣器和 PWM 输出控制 00:普通 IO 10：蜂鸣器输出
0	PWM3OUT	

位地址	标识符	功能
		01,11：PWM 输出

2.4.2.2. TM3IN 寄存器（地址为 1ch）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TM3IN	TM3IN[7:0]							

位地址	标识符	功能
7 ： 0	TM3IN[7:0]	定时/计数器溢出值低 8 位

2.4.2.3. TM3INH 寄存器（地址为 1fh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TM3INH					TM3IN[11:8]			

位地址	标识符	功能
3 ： 0	TM3INH[11:8]	定时/计数器溢出值高 4 位

注：该寄存器在 8bit 模式(PWM3_NUM=0)下无效。

2.4.2.4. TM3CNT 寄存器（地址为 1dh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM3CNT	TM3CNT[7:0]							

位地址	标识符	功能
7 ： 0	TM3CNT[7:0]	定时/计数器 3 计数寄存器低 8 位，**只读**

2.4.2.5. TM2CNTH 寄存器（地址为 25h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	U-0	R-0	R-0	R-0	R-0
TM2CNTH	TM3CNT_DT1[2:0]							

位地址	标识符	功能
7 : 5	TM3CNT_DT1[2:0]	定时/计数器 3 死区时间计数寄存器 1，**只读**

2.4.2.6. TM2RH 寄存器（地址为 26h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	U-0	R-0	R-0	R-0	R-0
TM2RH	TM3CNT_DT2[2:0]							

位地址	标识符	功能
7 : 5	TM3CNT_DT2[2:0]	定时/计数器 3 死区时间计数寄存器 2，**只读**

2.4.2.7. TM3CNTH 寄存器（地址为 27h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

特性	R-0	R-0	R-0	U-0	R-0	R-0	R-0	R-0
TM3CNTH	TM3CNT_DT3[2:0]				TM3CNT[11:8]			

位地址	标识符	功能
7 : 5	TM3CNT_DT3[2:0]	定时/计数器 3 死区时间计数寄存器 3，**只读**
3 ： 0	TM3CNTH[11:8]	定时/计数器 3 计数寄存器高 4 位，**只读**

2.4.2.8. TM3R1 寄存器（地址为 1eh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TM3R1	TM3R1[7:0]							

位地址	标识符	功能
7 ： 0	TM3R1[7:0]	定时/计数器 3 的 第 1 个 12 位 PWM 的占空比控制寄存器低 8 位 或者 第 1 个 8 位 PWM 的占空比控制寄存器

2.4.2.9. TM3R2 寄存器（地址为 3bh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TM3R2	TM3R2[7:0]							

位地址	标识符	功能
7 ： 0	TM3R2[7:0]	定时/计数器 3 第 2 个 12 位 PWM 的占空比控制寄存器低 8 位 或者 第 2 个 8 位 PWM 的占空比控制寄存器

2.4.2.10. TM3RH 寄存器（地址为 2ch）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TM3R3	TM3R3[7:0]							

位地址	标识符	功能
7 ： 4	TM3R3[7:4]	定时/计数器 3 的第 2 个 12 位 PWM 的占空比控制寄存器高 4 位 或者 第 3 个 8 位 PWM 的占空比控制寄存器高 4 位
3 ： 0	TM3R3[3:0]	定时/计数器 3 的第 1 个 12 位 PWM 的占空比控制寄存器高 4 位 或者 第 3 个 8 位 PWM 的占空比控制寄存器低 4 位

2.4.2.11. TM3CON2 寄存器（地址为 2dh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TM3CON2	DT3CK[1:0]		DT3CNT[2:0]			DT3_EN	P3H1OEN	P3L1OEN

位地址	标识符	功能
7:6	DT3CK[1:0]	定时器 3 死区时间时钟选择 DT3CK[1:0] | DT3_CLK

位地址	标识符	功能
		00 MCK 01 MCK/2 10 MCK/4 11 MCK/8
5:3	DT3CNT[2:0]	死区时间选择 死区时间=DT3CNT[2:0]*DT3_CLK
2	DT3_EN	死区发生器 3 使能位 0：不使能死区发生器 3 1：使能死区发生器 3
1	P3H1OEN	互补模式输出 PWM3H 从 PT1.4 输出使能 0：PWM3H 不输出 1：PWM3H 从 PT1.4 输出
0	P3L1OEN	互补模式输出 PWM3L 从 PT1.5 输出使能 0：PWM3L 不输出 1：PWM3L 从 PT1.5 输出

2.4.2.12. CONFIG1 寄存器（地址为 2eh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
CONFIG1	P3H1INV	P3L1INV		PWM3STALL			PWM3PO1	

位地址	标识符	功能
7	P3H1INV	取反互补模式输出 PWM3H1 从 PT1.4 输出取反控制位 0：PWM3H 不取反输出 1：PWM3H 取反输出
6	P3L1INV	取反互补模式输出 PWM3L1 从 PT1.5 输出取反控制位 0：PWM3L 不取反输出 1：PWM3L 取反输出
4	PWM3STALL	ICD 调试 STALL 时，定时器 3 单端 PWM 输出电平控制位 1：ICD 调试 STALL 时，定时器 3 单端 PWM 输出高电平 0：ICD 调试 STALL 时，定时器 3 单端 PWM 输出低电平
1	PWM3PO1	PWM3OUT1 正常模式输出脚选择 0：PT3.5 作为 PWM3 第一个正常模式输出口 1：PT5.1 作为 PWM3 第一个正常模式输出口

2.4.2.13. CONFIG2 寄存器（地址为 2fh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
CONFIG2	VTHSEL	PWM3MD	PWM3NUM	PWM3WV			PWM3PO3	PWM3PO2

位地址	标识符	功能
7	VTHSEL	PT 口施密特功能使能 0：打开施密特功能 1 关闭施密特功能
6	PWM3MD	计数器模式选择位 1 = 中心对齐模式 0 = 边沿对齐模式
5	PWM3NUM	定时器 3 计数位宽配置 0 = 8 位计数器模式，支持 3 路 8 位同周期 PWM 输出 1 = 12 位计数器模式，支持 2 路 12 位同周期 PWM 输出
4	PWM3WV	PWM3 波形模式 0：锯齿波。计数周期内从 0 到溢出值 TM3IN 递增加 1 1：三角波。计数周期内先从 0 递增到溢出值 TM3IN，然后再从溢出值 TM3IN 递减到 0
1	PWM3PO3	PWM3 正常模式第三个 PWM 输出脚选择 0：PT1.0 不作为 PWM3 第三个 PWM 正常模式输出口 1：PT1.0 作为 PWM3 第三个 PWM 正常模式输出口 注：这个 PWM 正常模式输出口有效时，需要 PWM3NUM=0
0	PWM3PO2	PWM3 正常模式第二个 PWM 输出脚选择 0：PT5.6 不作为 PWM3 第二个 PWM 正常模式输出口 1：PT5.6 作为 PWM3 第二个 PWM 正常模式输出口

2.4.2.14. TM3CON3 寄存器（地址为 45h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TM3CON3	P3H3INV	P3L3INV	P3H3OEN	P3L3OEN	P3H2INV	P3L2INV	P3H2OEN	P3L2OEN

位地址	标识符	功能
7	P3H3INV	取反互补模式输出 PWM3H3 从 PT5.5 输出取反控制位 0：PWM3H3 不取反输出 1：PWM3H3 取反输出
6	P3L3INV	取反互补模式输出 PWM3L3 从 PT5.7 输出取反控制位 0：PWM3L3 不取反输出 1：PWM3L3 取反输出
5	P3H3OEN	互补模式输出 PWM3H3 从 PT5.5 输出输出使能 0：PWM3H3 不从 PT5.5 输出 1：PWM3H3 从 PT5.5 输出
4	P3L3OEN	互补模式输出 PWM3L3 从 PT5.7 输出输出使能 0：PWM3L3 不从 PT5.7 输出 1：PWM3L3 从 PT5.7 输出
3	P3H2INV	取反互补模式输出 PWM3H2 从 PT5.1 输出取反控制位 0：PWM3H2 不取反输出 1：PWM3H2 取反输出

2	P3L2INV	取反互补模式输出 PWM3L2 从 PT5.0 输出取反控制位 0：PWM3L2 不取反输出 1：PWM3L2 取反输出
1	P3H2OEN	互补模式输出 PWM3H2 从 PT5.1 输出输出使能 0：PWM3H2 不从 PT5.1 输出 1：PWM3H2 从 PT5.1 输出
0	P3L2OEN	互补模式输出 PWM3L2 从 PT5.0 输出输出使能 0：PWM3L2 不从 PT5.0 输出 1：PWM3L2 从 PT5.0 输出

2.4.3. 时基单元

TM3 定时器的主要模块是一个 12 位计数器及其相关的计数器更新寄存器。计数器可以向上(锯齿波)或向上向下计数（三角波）。计数器时钟由预分频器分频获得。

时基单元包括：

- 计数器寄存器（TM3CNT）
- 预分频器寄存器（T3RATE）
- 计数器更新寄存器（TM3IN）

计数器更新寄存器

计数器更新寄存器可以设置计数溢出值，计数器不断与计数溢出值进行比较，当计数值 = 计数溢出值时，这时计数器更新寄存器就会产生一个中断标志。该控制寄存器没有缓冲器，故更新 TM3IN 寄存器的值后，会立刻生效。值得注意的是：实际计数溢出值=计数器更新寄存器写入的值+1，例如：在 TM3IN 中写入 99，则实际计数溢出值为 100。

预分频器

预分频器可以将计数器时钟分频处理，它是 7 位的，支持 1/2/4/8/16/32/64/128 分频。基于通过配置 T3RATE 寄存器来设置分频系数。它可以在运行时进行更改，因为该控制寄存器没有缓冲器，因此修改之后，立刻生效。

2.4.4. 时钟选择

定时器 3 模块的输入时钟通过 TSEL[1:0]进行选择，可选时钟源有 CPUCLK、MCK 或者内部 32K WDT 时钟。值得注意的是，定时器 0，定时器 2 和定时器 3 的时钟源是一致的，具体如下图所示：

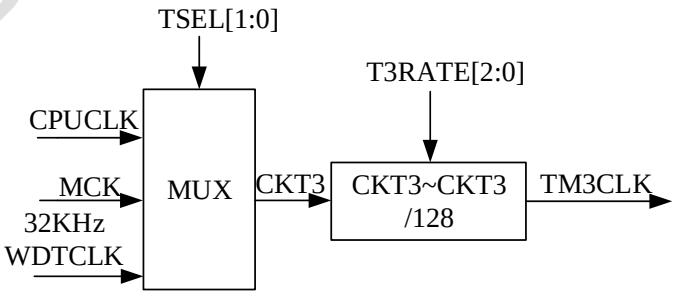


图 21　TM3 定时器时钟选择

2.4.5. 定时/计数器功能

定时器 3 的时钟源可以选择 CPUCLK、MCK，在 HALT 模式下定时器 3 仍然可以继续计数，因此可以唤醒 HALT 模式。SLEEP 模式下内部高速振荡器停止工作，定时器 3 停止计数，因此无法唤醒 SLEEP 模式。

定时/计数器操作：

1) 配置 TM3CON 寄存器的 T3RATE[2:0]，为定时器 3 选择时钟分频
2) 配置计数器模式，PWM3WV=0（锯齿波）。
3) 设置 TM3IN[11:0]，选择定时器溢出值。
4) 置位寄存器位 TM3IE 与 GIE，使能定时器中断。
5) 清零寄存器位 T3RSTB，复位定时器模块的计数器。
6) 置位寄存器位 T3EN，使能定时器模块的 12 Bits 计数器。
7) 当定时超时发生时，BZ 输出信号发生跳变，可作为蜂鸣器输出；程序计数器会跳转到 004H。

定时器 3 溢出时间计算方法：

定时器 3 溢出时间=（TM3IN[11:0]+1）/TM3CLK.　（TM3IN 不为 0）

2.4.6. 蜂鸣器

定时器 3 蜂鸣器输出默认为低电平，当定时器 3 发生计数溢出事件时，蜂鸣器输出翻转，因此蜂鸣器输出是一个占空比固定为 50%，周期为定时器 3 溢出时间 2 倍的方波。

配置蜂鸣器输出操作：

1) 把 PT3.5 配置为输出口。
2) 配置 TM3CON 寄存器的低 2 位，T3OUT 设置为 1，PWM3CON 设置为 0。
3) 配置 TM3CON 寄存器的 T3RATE[2:0]，为定时器 3 选择时钟分频。
4) 配置计数器模式，PWM3WV=0（锯齿波）。
5) 配置 TM3IN，选择定时器溢出值。
6) 清零定时器 3 计数值：将 TM3CON 寄存器的 T3RSTB 配置为 0，复位定时器 3 的计数器。
7) 使能定时器模块：将 TM3CON 的 T3EN 置 1，使能定时器模块的 12 bits 计数器。
8) 当定时超时发生时，BZ 输出信号发生跳变，可作为蜂鸣器输出。

蜂鸣器周期计算方法：

蜂鸣器周期=（TM3IN[11:0]+1）*2/TM3CLK.　（TM3IN 不为 0）

2.4.7. PWM

定时器 3 有 3 个 PWM 通道，可以四个管脚输出

PTXEN 为 PT 口的输入输出控制位，配置为 0 作为输入口，配置为 1 作为输出口。

寄存器配置				引脚功能
PT3EN[5]	PWM3OUT	T3OUT	PWM3PO1	PT3.5
0	X	X	X	输入
1	0	1	X	蜂鸣器输出
1	1	X	0	PWM31 输出
1	0	0	X	普通 IO 输出
X	1	X	1	普通 IO 输出

寄存器配置				引脚功能
PT5EN[1]	**PWM3OUT**	P3H2OEN	PWM3PO1	**PT5.1**
0	X	X	X	输入
1	X	1	X	互补模式输出 PWM3H2
1	1	0	1	PWM31 输出
1	0	0	X	普通 IO 输出
X	X	0	0	普通 IO 输出

寄存器配置			引脚功能
PT3EN[1]	**PWM3PO2**	PWM3NUM	**PT5.6**
0	X	X	输入
1	1	X	PWM32 输出
1	0	X	普通 IO 输出

寄存器配置			引脚功能
PT5EN[1]	**PWM3PO3**	PWM3NUM	**PT1.0**
0	X	X	输入
1	1	0	PWM33 输出
1	0	1	普通 IO 输出
1	1	1	无此配置功能
1	0	0	普通 IO 输出

例：配置 PWM 正常模式输出操作：

1) 配置 PWM3NUM 寄存器，选择计数位宽，当 PWM3NUM=1 时，通道 3 不能输出 PWM。
2) 把 PT3.5/PT5.1、5.6 或者 1.0 配置为输出口。
3) 设置 TM3IN 来配置 PWM3 的周期。
4) 设置 TM3R 来配置 PWM3 的高电平的的脉宽。
5) 使能 PWM3OUT 输出，配置 PT3.5 或者 PT5.1、5.6 或者 1.0 为输出端口，之后把 T3EN 置 1 启动定时器。
6) PWM3 从 PT3.5 或者 PT5.1、5.6 或者 1.0 输出。

周期为 TM3IN+1，高电平脉宽为 TM3R。如 TM3IN=0x0F，TM3R=0x03 的 PWM3 波形输出如下：

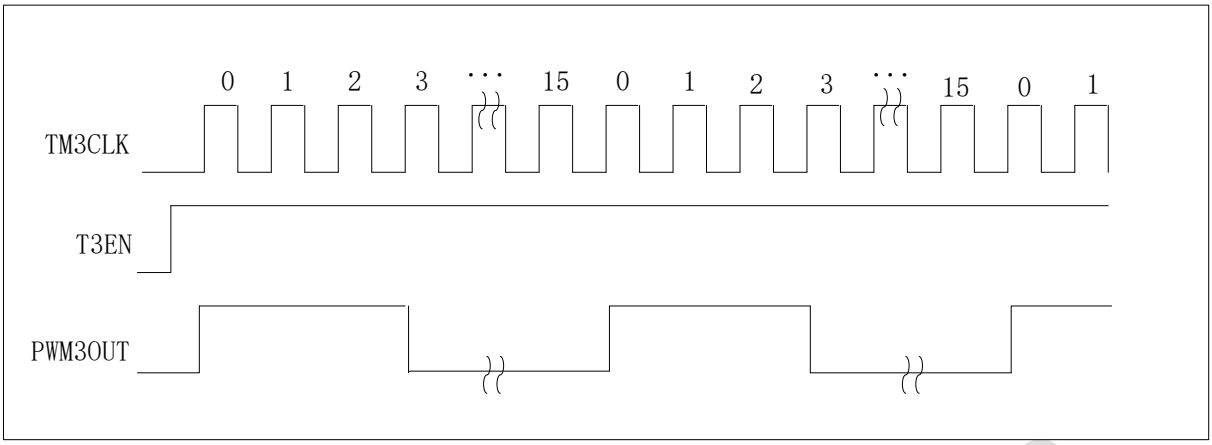


图 22　PWM3 波形输出

2.4.8.　互补式 PWM 输出

定时器 3 支持互补式 PWM 输出， PWM 输出的优先级从上到下递减

PT1EN 和 PT5EN 为 PT1 和 PT5 口的输入输出控制位，配置为 0 作为输入口，1 为输出口。

寄存器配置				引脚功能	
PT1EN[5:4]	**P3H1OEN**	**P3L1OEN**	PWM3NUM	PT1.4	PT1.5
00	X	X	X	输入	输入
11	1	1	X	互补模式输出 PWM3H1	互补模式输出 PWM3L1
11	1	0	X	互补模式输出 PWM3H1	不做互补模式 PWM3L1 输出口
11	0	1	X	不做互补模式 PWM3H1 输出口	互补式输出 PWM3L1
11	0	0	X	普通 IO 输出	普通 IO 输出

寄存器配置				引脚功能		
PT5EN[1:0]	**P3H2OEN**	**P3L2OEN**	**PWM3PO1**	PWM3NUM	PT5.0	PT5.1
00	X	X	X	X	输入	输入
11	1	1	X	X	互补模式输出 PWM3L2	互补模式输出 PWM3H2
11	1	0	X	X	不做互补模式 PWM3L2 输出口	互补模式输出 PWM3H2
11	0	1	X	X	互补模式输出 PWM3L2	不做互补模式输出 PWM3H2

注：PT5.0 具体功能见 2.3.6 章节关于 PT5.0 的表，PT5.1 具体功能见 2.4.7 章节关于 PT5.1 的表

寄存器配置				引脚功能	
PT5EN[7][5]	**P3H3OEN**	**P3L3OEN**	PWM3NUM	PT5.5	PT5.7

00	X	X	0	输入	输入
11	1	1	0	互补模式输出 PWM3H3	互补模式输出 PWM3L3
11	1	0	0	互补模式输出 PWM3H3	普通 IO 输出
11	0	1	0	普通 IO 输出	互补模式输出 PWM3L3
11	0	0	0	普通 IO 输出	普通 IO 输出
11	X	X	1	无此配置功能	无此配置功能

CS8M32X 提供一对源于定时器 3 的互补式输出，可用作 PWM 驱动信号。对于 PMOS 管驱动，PWM 输出为低电平有效，而对于 NMOS 管驱动，PWM 输出为高电平有效。当这对互补式输出同时用于驱动 PMOS 和 NMOS 时，死区时间发生器插入一死区时间以防止直流电流过大，该死区时间可通过 TM3CON2 寄存器的 DT3CK[1:0]和 DT3CNT[2:0]位来定义。在死区时间发生器输入信号的每个上升沿插入一个死区时间。通过死区插入电路，输出信号最终发送至外部功率晶体管。

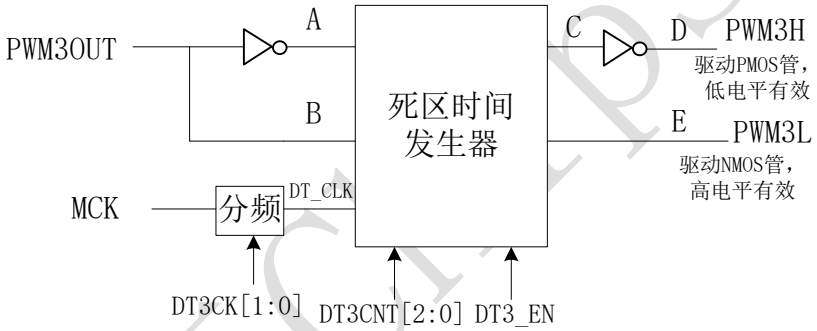


图 23　互补式 PWM 输出方框图

例：配置 PWM 取反互补式输出操作：

1）把 PT1.4/PT1.5 或者 PT5.1/PT5.0 或者 PT5.5/PT5.7 配置为输出口。

2）设置 TM3IN 来配置 PWM3 的周期。

3）设置 TM3R 来配置 PWM3 的高电平的的脉宽。

4）配置死区使能寄存器 DT3_EN=1。

5）设置 DT3CNT 来配置死区时间，设置 P3HXINV 和 P3LXINV 配置是否取反输出。

6）配置输出端口，之后把 T3EN 置 1 启动定时器。

7）PWM3 互补式输出从 PT1.4/PT1.5 或者 PT5.1/PT5.0 或者 PT5.5/PT5.7 输出。

互补式 PWM 输出波形

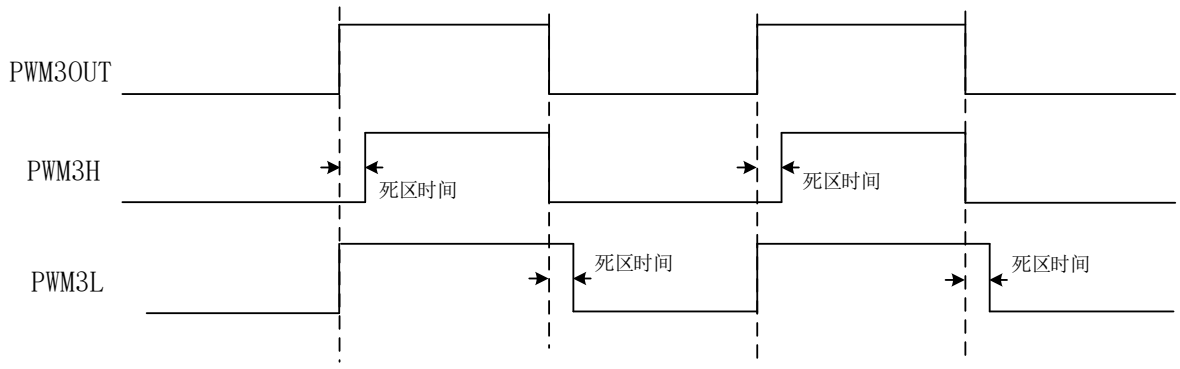


图 24　互补式 PWM 输出波形图

PWM 输出取反后的互补 PWM 输出

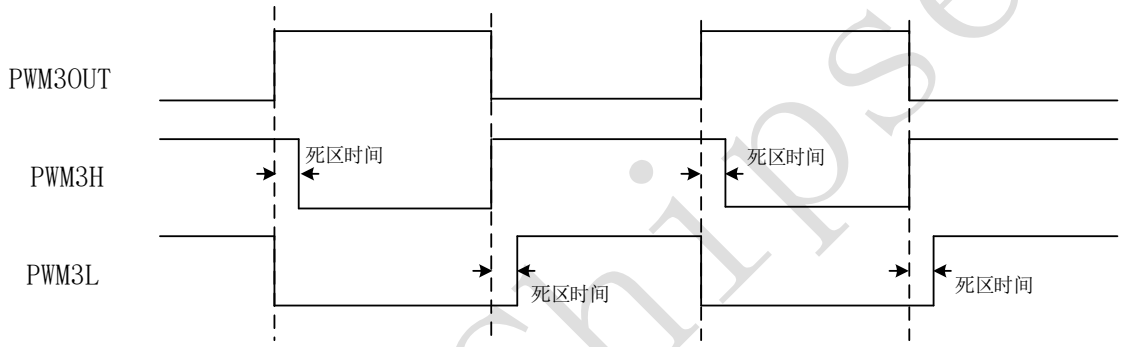


图 25　取反后的互补 PWM 输出波形图

2.4.9.　边沿对齐 PWM 配置

当 PWM 使能后，计数器从 0 开始对时钟信号递增计数，开始一个输出周期。当计数值小于占空比常数寄存器 TM3Rx 时，比较器输出高电平；当计数值大于于占空比常数寄存器 TM3Rx 且小于周期常数寄存器 TM3IN 时，比较器输出低电平，结束一个输出周期。

要使用计数器的比较匹配事件生成边沿对齐 PWM，操作如下：

(1)　配置 PWM3 波形模式,选择边沿对齐 PWM3MD=0;

(2)　配置计数器模式，PWM3WV=0（锯齿波）；

(3)　选择定时器的两个通道，配置 TM3IN 和 TM3Rx 的值设置周期和占空比；

(4)　设置 T3EN=1 来使能计数器。

下图展示了边沿对齐 PWM 生成：

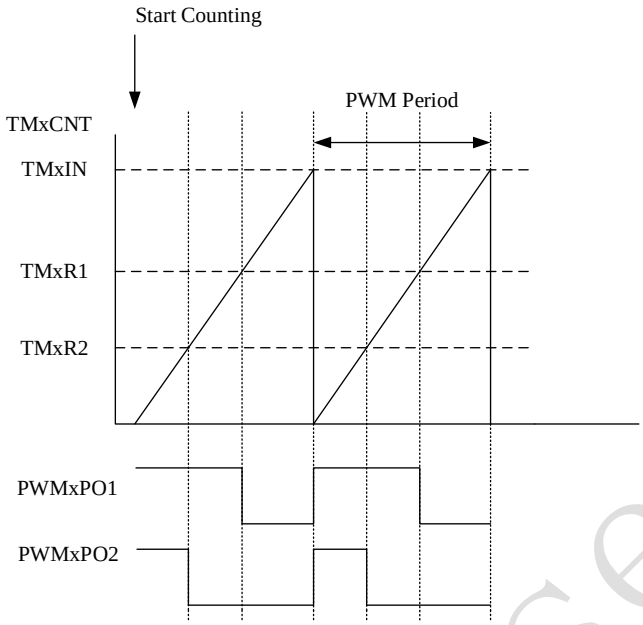


图 26　边沿对齐 PWM 信号

2.4.10. 中心对齐 PWM 配置

在中心对齐模式下，TM3IN 值包含半个周期。波形使用递增比较事件和递减比较事件输出高电平或低电平，具体取决于比较块和计数器的配置设置。PWM 周期为 2*TM3IN，占空比为 TM3Rx/TM3IN。

要使用计数器的比较匹配事件生成中心对齐 PWM，操作如下：

(1)　配置计数器位宽位 8 位，PWM3NUM=0；

(2)　配置 PWM3 波形模式,选择中心对齐 PWM3MD=1；

(3)　配置计数器模式，PWM3WV=1（三角波）；

(4)　选择定时器的两个通道，配置 TM3IN 和 TM3Rx 的值设置周期和占空比；

(5)　设置 T3EN=1 来使能计数器。

下图展示了中心对齐 PWM 生成：

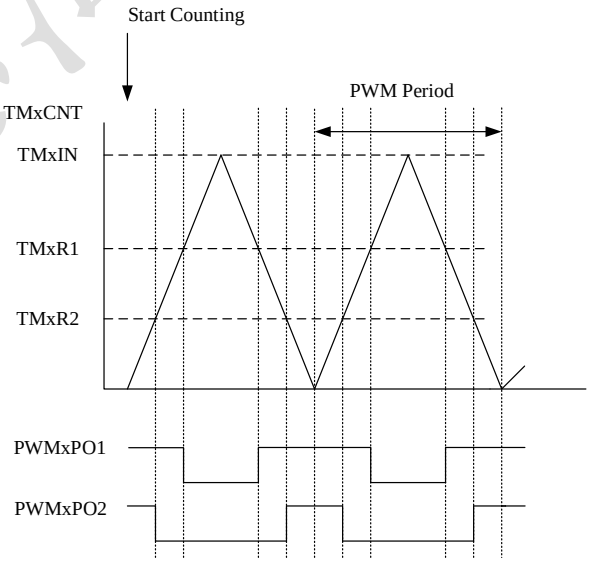


图 27　中心对齐 PWM 信号

2.5. CVC 模块

CVC 模块用于检测外部电容变化，在休眠模式下，CVC 模块支持定时检测功能，并可在变化量超出设定阈值时唤醒 MCU。此外，CVC 模块还具有防碰撞，死机保护功能。

2.6. 模数转换器（ADC）

CS8M32X 模数转换模块共用 8 条外部通道（AIN0~AIN7）和 3 条特殊通道(AIN8：内部 1/8VDD；AIN9：内部参考电压；AIN10：GND；)，可以将模拟信号转换成 12 位数字信号。进行 AD 转换时，首先要选择输入通道(AIN0~AIN10)，然后把 SRADEN 置 1 使能 ADC，之后把 SRADS 置 1，启动 AD 转换。转换结束后，系统自动将 SRADS 清 0，并将转换结果存入寄存器 SRADL 和 SRADH 中。

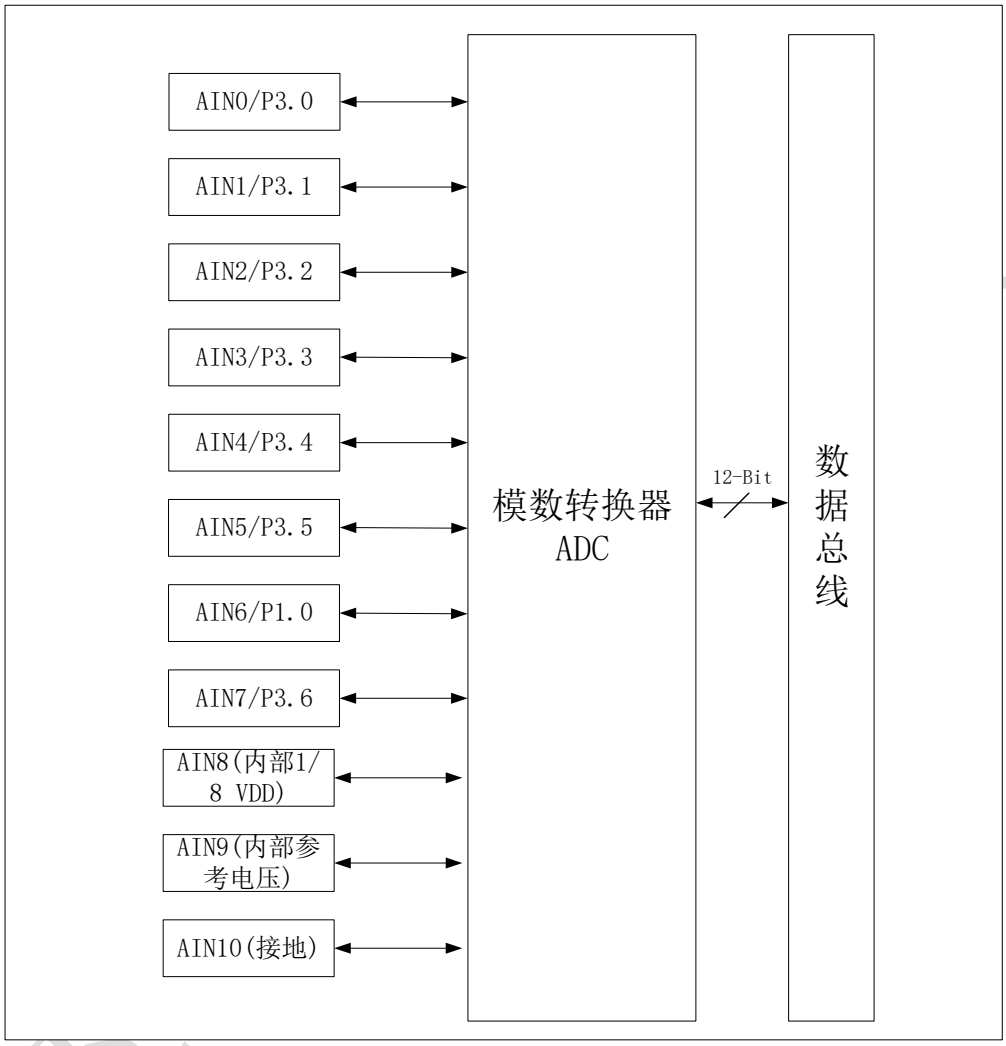


图 28　模数转换器 ADC 功能框图

2.6.1.　寄存器描述

表 10　ADC 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06h	INTF		TM2IF		TM0IF	SRADIF		E1IF	E0IF	00000u00
07h	INTE	GIE	TM2IE		TM0IE	SRADIE		E1IE	E0IE	00000u00
34h	SRADCON0	ADC2V_EN	SARVCMSE	SRADACKS[1:0]				SRADCKS[1:0]		1100uu00
35h	SRADCON1	SRADEN	SRADS	OFTEN	CALIF	ENOV	OFFEX	VREFS[1:0]		00000010
36h	SRADCON2	CHS[3:0]							SAR_DIFFEN	0000uuu1
37h	SRADL	SRAD[7:0]								00000000
38h	SRADH					SRAD[11:8]				uuuu0000
39h	SROFTL	SROFT[7:0]								00000000
3ah	SROFTH					SROFT[11:8]				uuuu0000

2.6.1.1.　SRADCON0 寄存器（地址为 34h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-1	R/W-1	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
SRADCON0	ADC2V_EN	SARVCMSEL	SRADACKS[1:0]				SRADCKS[1:0]	

位地址	标识符	功能
7	ADC2V_EN	ADC 参考电压选择 ADC2V_EN：ADC 参考电压选择 1：VREF=2V，仅在 CVC 功能中使用 0：VREF=1.22V
6	SARVCMSEL	ADC 采样共模电压选择 SARVCMSEL：ADC 采样共模电压 1：SAR_DIFFEN =1 下，输入信号共模为 VDD，选择该配置 0：SAR_DIFFEN =1 下，输入信号共模小于等于 1/2VREF；或者 SAR_DIFFEN =0，选择该配置
5：4	SRADACKS[1:0]	ADC 输入信号采样时钟个数选择信号 SRADACKS[1:0]：ADC 输入信号采样时钟个数 00：16 个 ADC 时钟 01：8 个 ADC 时钟 10：4 个 ADC 时钟 11：2 个 ADC 时钟
1：0	SRADCKS[1:0]	ADC 时钟频率选择信号 SRADCKS[1:0]：ADC 时钟频率选择 00：CPUCLK 01：CPUCLK/2 10：CPUCLK/4 11：CPUCLK/8

2.6.1.2. SRADCON1 寄存器（地址为 35h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0
SRADCON1	SRADEN	SRADS	OFTEN	CALIF	ENOV	OFFEX	VREFS[1:0]	

位地址	标识符	功能
7	SRADEN	ADC 使能位 1：使能 0：禁止
6	SRADS	ADC 启动位/状态控制位 1：开始，转换过程中 0：停止，转换结束 当置位后，启动 ADC 转换，转换完成会自动清 0
5	OFTEN	转换结果选择控制位 1：转换结果放在 SROFT 寄存器中 0：转换结果放在 SRAD 寄存器中
4	CALIF	校正控制位(OFTEN 为 0 时有效) 1：使能校正，即 AD 转换的结果是减去了 SROFT 失调电压值 0：禁止校正，即 AD 转换结果是没有减去 SROFT 失调电压值
3	ENOV	使能比较器溢出模式(CALIF 为 1 时有效) 1：使能，上溢或下溢直接是减去后的结果 0：禁止，下溢为 000h，上溢为 FFFH
2	OFFEX	差分 ADC 正负端内部互换（SAR_DIFFEN=1 时，OFFEX 才有效） 1: 差分 ADC 正负端内部互换 0: 差分 ADC 正负端内部不互换
1:0	VREFS[1:0]	ADC 内部参考电源选择 VREFS[1:0] — AD 参考电压 00 — AD 参考电压为 VDD 01 — AD 参考电压为 PT30 口外部输入电压 10 — AD 参考电压为 1.22V 内部电压 11 — AD 参考电压为 1.22V 内部电压并通过 PT30 口输出

2.6.1.3. SRADCON2 寄存器（地址为 36h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-1
SRADCON2	CHS[3:0]							SAR_DIFFEN

位地址	标识符	功能
7：4	CHS[3:0]	ADC 输入通道选择位 CHS[3:0] — 输入通道 SAR_DIFFEN=1,为差分 ADC 的 0000 — AIN0 / AIN1

<table>
<tr><th>位地址</th><th>标识符</th><th>功能</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>0001</td><td>AIN2/ AIN3</td></tr>
<tr><td>0010</td><td>AIN4/ AIN5</td></tr>
<tr><td>0011</td><td>AIN6/ AIN7</td></tr>
<tr><td>0100</td><td>AIN0/ VSSA</td></tr>
<tr><td>0101</td><td>AIN1/ VSSA</td></tr>
<tr><td>0110</td><td>AIN2/ VSSA</td></tr>
<tr><td>0111</td><td>AIN3/ VSSA</td></tr>
<tr><td>1000</td><td>AIN4/ VSSA</td></tr>
<tr><td>1001</td><td>AIN5/ VSSA</td></tr>
<tr><td>1010</td><td>AIN6/ VSSA</td></tr>
<tr><td>1011</td><td>AIN7/ VSSA</td></tr>
<tr><td>1100</td><td>ADC 正端 1/8VDD　负端 VSSA</td></tr>
<tr><td>1101</td><td>ADC 内部参考电压 /VSSA</td></tr>
<tr><td>1110</td><td>ADC 内部接地/负端 VSSA</td></tr>
<tr><td>1111</td><td>保留</td></tr>
<tr><td colspan="2">SAR_DIFFEN=0 为单端 ADC，以下为单端</td></tr>
<tr><td>0000</td><td>AIN0</td></tr>
<tr><td>0001</td><td>AIN1</td></tr>
<tr><td>0010</td><td>AIN2</td></tr>
<tr><td>0011</td><td>AIN3</td></tr>
<tr><td>0100</td><td>AIN4</td></tr>
<tr><td>0101</td><td>AIN5</td></tr>
<tr><td>0110</td><td>AIN6</td></tr>
<tr><td>0111</td><td>AIN7</td></tr>
<tr><td>1000</td><td>无效</td></tr>
<tr><td>1001</td><td>无效</td></tr>
<tr><td>1010</td><td>无效</td></tr>
<tr><td>1011</td><td>无效</td></tr>
<tr><td>1100</td><td>ADC 内部 1/8VDD</td></tr>
<tr><td>1101</td><td>ADC 内部参考电压</td></tr>
<tr><td>1110</td><td>ADC 内部接地</td></tr>
<tr><td>1111</td><td>CVC 通道</td></tr>
</table>
</td></tr>
<tr><td>3：1</td><td>RESERVE</td><td>保留</td></tr>
<tr><td>0</td><td>SAR_DIFFEN</td><td>
<table>
<tr><td>SAR_DIFFEN</td><td>AD 差分输入使能</td></tr>
<tr><td>1</td><td>1：全差分模式(参考电压不能选择 VDD)；（默认）</td></tr>
<tr><td>0</td><td>ADC 为单端模式</td></tr>
</table>
</td></tr>
</table>

2.6.1.4. SRADL 寄存器（地址为 37h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SRADL	SRAD[7:0]							

位地址	标识符	功能
7：0	SRAD[7:0]	ADC 数据的低 8 位，只可读

2.6.1.5. SRADH 寄存器（地址为 38h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
SRADH					SRAD[11:8]			

位地址	标识符	功能
3：0	SRAD[11:8]	ADC 数据的高 4 位，只可读

2.6.1.6. SROFTL 寄存器（地址为 39h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SROFTL	SROFT[7:0]							

位地址	标识符	功能
7：0	SROFT[7:0]	校正值数据的低 8 位

2.6.1.7. SROFTH 寄存器（地址为 3ah）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
SROFTH					SROFT[11:8]			

位地址	标识符	功能
3：0	SROFT[11:8]	校正值数据的高 4 位

2.6.2. 输入电压和 SRAD 输出数据的关系

当 SAR_DIFFEN=0 时，AD 转换结果没有符号位，支持单端信号输入。

如果确定参考电压值，通过 ADC 转换得到了 ADC 码值，那么可以通过简单计算得出 ADC 输入电压值。计算公式如下

ADC 输入电压 = （SRAD[11:0]/4096）*VREF

例如：当 ADC 参考电压为 1.22V，ADC 转换码值为 0x200，即十进制的 512，那么输入电压值为（512/4096）* 1.22 = 0.1525V。

表 11　输入电压和 SRAD 输出数据的关系

输入电压	SRAD[11:0]											
	11	10	9	8	7	6	5	4	3	2	1	0
0/4096*VREF	0	0	0	0	0	0	0	0	0	0	0	0

输入电压	SRAD[11:0]											
	11	10	9	8	7	6	5	4	3	2	1	0
1/4096*VREF	0	0	0	0	0	0	0	0	0	0	0	1
…												
…												
4094/4096*VREF	1	1	1	1	1	1	1	1	1	1	1	0
4095/4096*VREF	1	1	1	1	1	1	1	1	1	1	1	1

当 SAR_DIFFEN=1 时，AD 转换结果有符号位，其中最高位为 0 表示负值，最高位为 1 表示正值。

如果确定参考电压值，通过 ADC 转换得到了 ADC 码值，那么可以通过简单计算得出 ADC 输入电压值。计算公式如下

ADC 输入电压 = （（SRAD[11:0]-2048）/2048）*VREF

例如：当 ADC 参考电压为 1.22V，ADC 转换码值为 0xA00，即十进制的 2560，那么输入电压值为（（2560-2048）/2048）* 1.22 = 0.305V。

表 12　输入电压和 SRAD 输出数据的关系

输入电压	SRAD[11:0]											
	11	10	9	8	7	6	5	4	3	2	1	0
-2047/2048*VREF	0	0	0	0	0	0	0	0	0	0	0	0
-2046/2048*VREF	0	0	0	0	0	0	0	0	0	0	0	1
…												
…												
0/2048*VREF	1	0	0	0	0	0	0	0	0	0	0	0
1/2048*VREF	1	0	0	0	0	0	0	0	0	0	0	1
…												
…												
2046/2048*VREF	1	1	1	1	1	1	1	1	1	1	1	0
2047/2048*VREF	1	1	1	1	1	1	1	1	1	1	1	1

2.6.3.　输入信号类型

2.6.3.1.　单端模式

当 SAR_DIFFEN=0 时，SARVCMSEL=0，支持单端单极输入信号，ADC 的参考电压 VREF 可以选择内部参考 1.22V 或者 VDD 或者外部灌入参考电压（参考数据手册 3.4 小节 ADC 电气特性对参考电压范围的说明）。

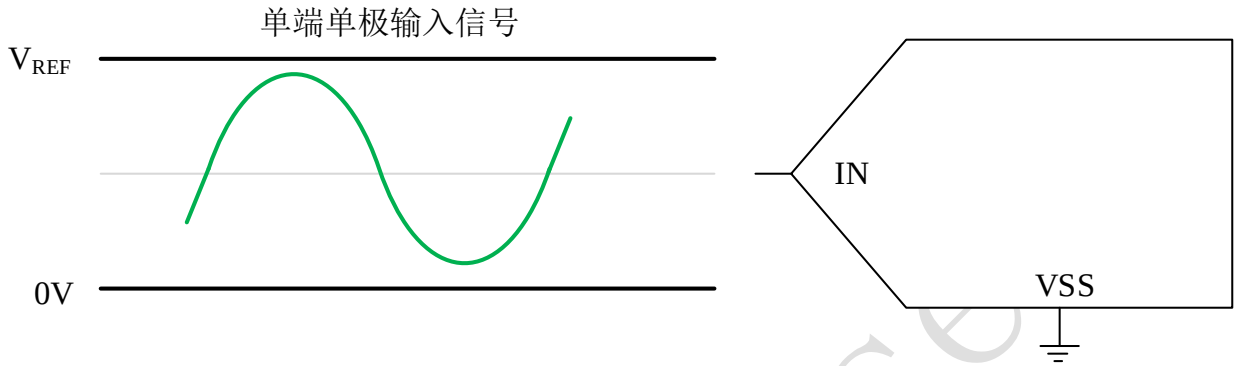


图 29　单端单极输入信号

2.6.3.2.　差分模式

当 SAR_DIFFEN=1 时，SARVCMSEL=0，支持全差分输入信号（全差分输入共模电压 VREF/2±0.05V）或者伪差分输入（伪差分输入 VIN 和 GND 或者 VIN 和 VREF 或者 VIN 和 VREF/2）或者单端单极输入（比如 CHS[3:0]=0100~1110，负端信号内部固定接 GND）。

当 SAR_DIFFEN=1 时，SARVCMSEL=0，支持全差分输入信号，全差分输入共模电压为 VREF/2±0.05V，ADC 的参考电压 VREF 可以选择内部参考 1.22V 或者外部灌入参考电压（参考数据手册 3.4 小节 ADC 电气特性对参考电压范围的说明）。

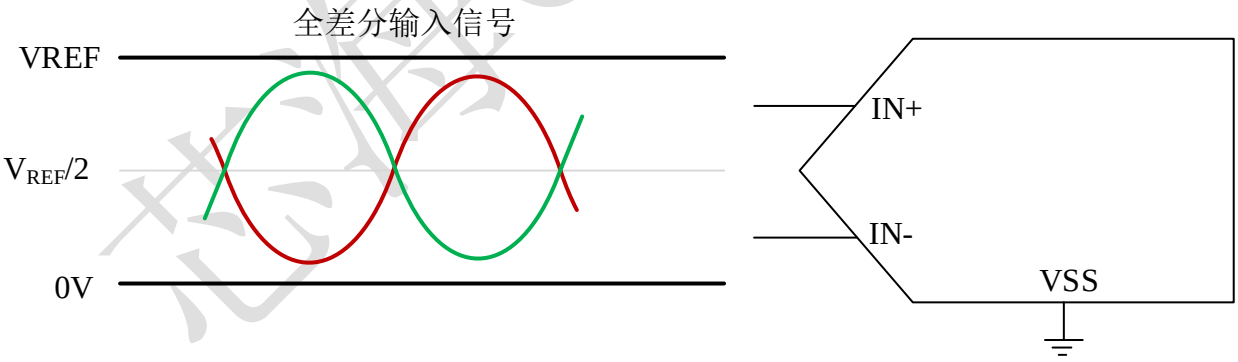


图 30　全差分输入信号

当 SAR_DIFFEN=1 时，SARVCMSEL=0，支持伪差分输入信号，包括伪差分单极输入信号 VIN 和 GND，或者伪差分单极输入信号 VIN 和 VREF，或者伪差分双极输入信号 VIN 和 VREF/2，ADC 的参考电压 VREF 可以选择内部参考 1.22V 或者外部灌入参考电压（参考数据手册 3.4 小节 ADC 电气特性对参考电压范围的说明）。

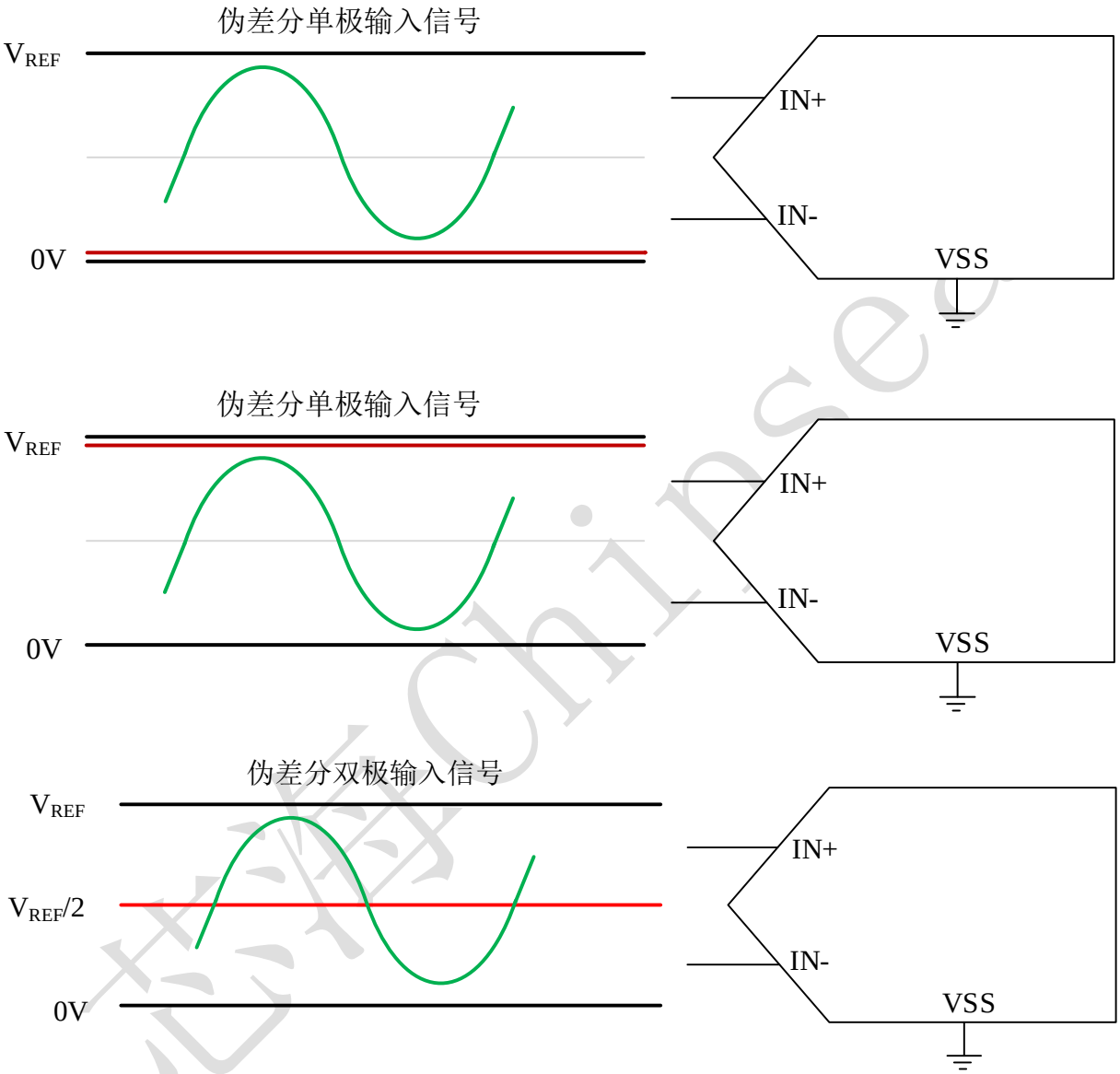


图 31　伪差分输入信号

当 SAR_DIFFEN=1 时，SARVCMSEL=0，支持单端单极输入信号，比如 CHS[3:0]=0100~1110，负端信号内部固定接 GND，ADC 的参考电压 VREF 可以选择内部参考 1.22V 或者外部灌入参考电压（参考数据手册 3.4 小节 ADC 电气特性对参考电压范围的说明）。

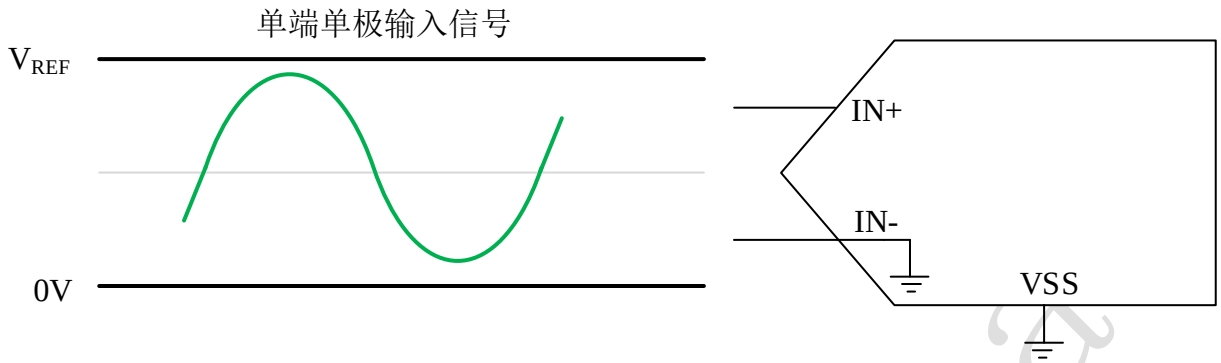


图 32　单端单极输入信号

当 SAR_DIFFEN=1 时，SARVCMSEL=1，支持全差分输入信号，全差分输入共模电压为 VDD±0.05V，ADC 的参考电压 VREF 可以选择内部参考 1.22V 或者外部灌入参考电压（参考数据手册 3.4 小节 ADC 电气特性对参考电压范围的说明）。

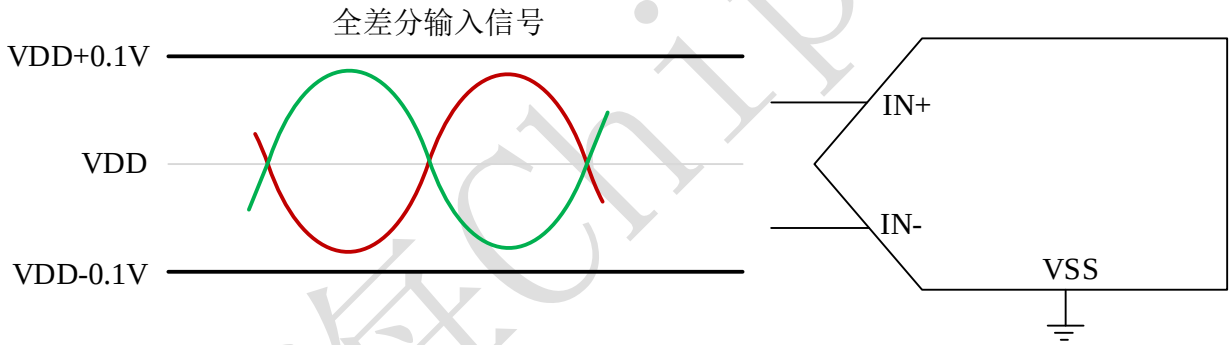


图 33　全差分输入信号

当 SAR_DIFFEN=1 时，SARVCMSEL=1，支持伪差分双极输入信号，伪差分双极输入信号为 VIN 和 VDD，ADC 的参考电压 VREF 可以选择内部参考 1.22V 或者外部灌入参考电压（参考数据手册 3.4 小节 ADC 电气特性对参考电压范围的说明）。

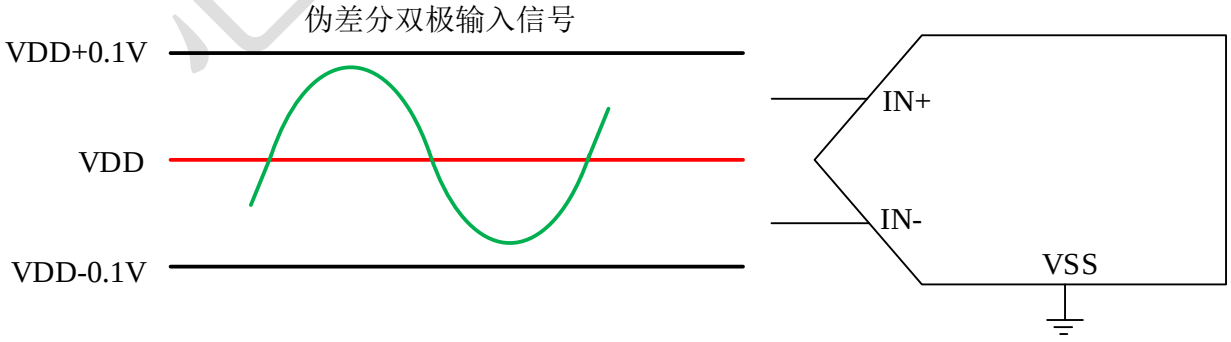


图 34　伪差分双极输入信号

2.6.4. 转换时间

12 位 AD 转换时间=（1/ADC 时钟频率）×　（12+CALIF+ADC 输入信号获取时间）

表 13　转换时间说明表(1)

MCK1[2:1]	CALIF	SRADCKS	SRADACKS	AD 转换时间(3)
4M 指令周期	0	01	00	1/ ((32MHz / 8) / 2)　× (12 + 0 + 16) = 14μs
			01	1/ ((32MHz / 8) / 2)　× (12 + 0 + 8) = 10μs
		10	00	1/ ((32MHz / 8) / 4)　× (12 + 0 + 16) = 28μs
			01	1/ ((32MHz / 8) / 4)　× (12 + 0 + 8) = 20μs
			10	1/ ((32MHz / 8) / 4)　× (12 + 0 + 4) = 16μs
		11	00	1/ ((32MHz / 8) / 8)　× (12 + 0 + 16) = 56μs
			01	1/ ((32MHz / 8) / 8)　× (12 + 0 + 8) = 40μs
			10	1/ ((32MHz / 8) / 8)　× (12 + 0 + 4) = 32μs
			11	1/ ((32MHz / 8) / 8)　× (12 + 0 + 2) = 28μs
	1	01	00	1/ ((32MHz / 8) / 2)　× (12 + 1 + 16) = 14.5μs
			01	1/ ((32MHz / 8) / 2)　× (12 + 1 + 8) = 10.5μs
		10	00	1/ ((32MHz / 8) / 4)　× (12 + 1 + 16) = 29μs
			01	1/ ((32MHz / 8) / 4)　× (12 + 1 + 8) = 21μs
			10	1/ ((32MHz / 8) / 4)　× (12 + 1 + 4) = 17μs
		11	00	1/ ((32MHz / 8) / 8)　× (12 + 1 + 16) = 58μs
			01	1/ ((32MHz / 8) / 8)　× (12 + 1 + 8) = 42μs
			10	1/ ((32MHz / 8) / 8)　× (12 + 1 + 4) = 34μs
			11	1/ ((32MHz / 8) / 8)　× (12 + 1 + 2) = 30μs
2M 指令周期	0	01	00	1/ ((32MHz / 16) / 2)　× (12 + 0 + 16) = 28μs
			01	1/ ((32MHz / 16) / 2)　× (12 + 0 + 8) = 20μs
			10	1/ ((32MHz / 16) / 2)　× (12 + 0 + 4) = 16μs
		10	00	1/ ((32MHz / 16) / 4)　× (12 + 0 + 16) = 56μs
			01	1/ ((32MHz / 16) / 4)　× (12 + 0 + 8) = 40μs
			10	1/ ((32MHz / 16) / 4)　× (12 + 0 + 4) = 32μs
			11	1/ ((32MHz / 16) / 4)　× (12 + 0 + 2) = 24μs
		11	00	1/ ((32MHz / 16) / 8)　× (12 + 0 + 16) = 112μs
			01	1/ ((32MHz / 16) / 8)　× (12 + 0 + 8) = 80μs
			10	1/ ((32MHz / 16) / 8)　× (12 + 0 + 4) = 64μs
			11	1/ ((32MHz / 16) / 8)　× (12 + 0 + 2) = 56μs
	1	01	00	1/ ((32MHz / 16) / 2)　× (12 + 1 + 16) = 29μs
			01	1/ ((32MHz / 16) / 2)　× (12 + 1 + 8) = 21μs
			10	1/ ((32MHz / 16) / 2)　× (12 + 1 + 4) = 17μs
		10	00	1/ ((32MHz / 16) / 4)　× (12 + 1 + 16) = 58μs
			01	1/ ((32MHz / 16) / 4)　× (12 + 1 + 8) = 42μs
			10	1/ ((32MHz / 16) / 4)　× (12 + 1 + 4) = 34μs
			11	1/ ((32MHz / 16) / 4)　× (12 + 1 + 2) = 30μs
		11	00	1/ ((32MHz / 16) / 8)　× (12 + 1 + 16) = 116μs

MCK1[2:1]	CALIF	SRADCKS	SRADACKS	AD 转换时间(3)
			01	1/ ((32MHz / 16) / 8)　× (12 + 1 + 8) = 84μs
			10	1/ ((32MHz / 16) / 8)　× (12 + 1 + 4) = 68μs
			11	1/ ((32MHz / 16) / 8)　× (12 + 1 + 2) = 60μs

(1) Fosc=32MHz

(2) MCK1[2:1]

(3) AD 转换时间随 Fosc 频率的改变而改变。

2.6.5. ADC 采样时间

ADC 采样时间通过 SRADCON0 寄存器进行配置，通过 SRADACKS[1:0]配置采样时钟个数，通过 SRADCKS[1:0]配置 ADC 时钟频率。

ADC 采样时间 = 采样时钟个数 / ADC 时钟频率。

下面以内部高速时钟为 32MHz，指令周期 4MHz 为例，对不同配置下的采样时间进行计算。

表 14　ADC 采样时间表

SRADCKS	SRADACKS	AD 采样时间
00	00	16/(4MHz/1) = 4μs
	01	8/(4MHz/1) = 2μs
	10	4/(4MHz/1) = 1μs
	11	2/(4MHz/1) = 0.5μs
01	00	16/（4MHz/2） = 8μs
	01	8/（4MHz/2） = 4μs
	10	4/（4MHz/2） = 2μs
	11	2/（4MHz/2） = 1μs
10	00	16/（4MHz/4） = 16μs
	01	8/（4MHz/4） = 8μs
	10	4/（4MHz/4） = 4μs
	11	2/（4MHz/4） = 2μs
11	00	16/（4MHz/8） = 32μs
	01	8/（4MHz/8） = 16μs
	10	4/（4MHz/8） = 8μs
	11	2/（4MHz/8） = 4μs

ADC 采样时间与芯片电压、外部负载都有关系，如果想满足 ADC 标称的性能，请满足下面的测试条件要求。

参数	测试条件	最小值	典型值	最大值	单位
采样时间	2.4V≤VDD≤5.5V， R_{AIN}≤0.3 kΩ	1			μs
	2.4V≤VDD<5.5V， R_{AIN}≤1 kΩ	2			μs
	2.4V≤VDD<5.5V， R_{AIN}≤5 kΩ	4			μs
	2V≤VDD<2.4V，R_{AIN}≤5 kΩ	16			μs
	1.8V≤VDD<2V，RAIN≤5 kΩ				μs

R_{AIN} 为输入负载电阻。

2.6.6. 参考电压输出

ADC 内部参考电压输出是将 ADC 参考电压配置为内部参考电压为 1.22V，并通过 PT3.0 口输出，可外接电容提高参考电压精度。

配置 ADC 参考电压输出步骤如下：

1） 配置 PT3CON 寄存器，将 PT3.0 口设置为模拟口

2） 配置 SRADCON1 的 VREFS[1:0]位为 11，将 ADC 参考电压配置为内部参考电压为 1.22V。

3） 打开 ADC 使能

2.6.7. AD 失调电压校正

不同芯片由于离散性的原因，AD 的失调电压可能有正有负。

校正失调电压的方法：

1） 芯片在量产时对 ADC 失调电压进行了测试，并将失调电压对应的码值（1.22V 参考电压下）。存在信息区地址 0xFC0C 中，低 8 位有效。用户需要在用户程序中用 MOVP 指令将该数据读出。

2） 将该数据写入 SROFTL 寄存器中，并向 SROFTH 写入 00h。

3） 将 SRADCON1 寄存器的 OFTEN 位置 0，CALIF 位置 1，ENOV 位置 0。

4） 使能 ADC 转换，那么从 SRADH/SRADL 中读取的 AD 值即为减去失调后的值。

```
…
MOVLW   FCH
MOVWF   EADRH   ;给高字节地址赋值
MOVLW   0CH
MOVWF   EADRL   ;给低字节地址赋值
MOVP            ;执行读操作
NOP
MOVWF   SROFTL
MOVLW   00H
MOVWF   SROFTH
CLRF    SRADCON1        ;VDD 为参考电压,often=0,calif=0;enov=0,offex=0,vrefs=00
BSF   SRADCON1,CALIF ; calif=1;

MOVLW   20h
MOVWF   SRADCON2     ;chs[3:0]=0010，选择通道 2
BSF     SRADCON1,7   ;使能 ADC 模块
CALL    delay_40us
…
BSF     SRADCON1,6   ;srads=1,开始转换
BTFSC   SRADCON1,6   ;检测转换是否完成
GOTO    $-1
MOVLW   sradl
MOVWF   adtmp1_l
MOVLW   sradh
MOVWF   adtmph_l
…
```

2.6.8. 数字比较器

ADC 模块可作为一个数字比较器。被测信号的输入频率应小于转换频率的 1/2。比较器的速率是和 AD 转换频率相关的。两个输入信号的差值必须小于 $V_{REF}/2$，否则比较结果会出错。

操作：

1） 通过 ADC 通道选择控制位 chs[3:0]选择比较器负端的信号输入，之后把 OFTEN 置 1，CALIF 清

0，ENOV 置 0，把 SRADEN 置 1 使能 ADC，SRADS 置 1 启动转换，转换完成可把转换结果写入 SROFT 寄存器。也可以直接把负端信号的 AD 值直接写到 SROFT 寄存器中，即人为指定负端电压值。

2） 通过 ADC 通道选择控制位 CHS[3:0]选择比较器正端的信号输入，之后把 OFTEN 置 0，CALIF 清 1，ENOV 置 1，把 SRADEN 置 1 使能 ADC，SRADS 置 1 启动转换。

3） AD 数据的最高位 SRAD[11]则是比较器的结果，为 0 时表示正端电压大于负端电压，为 1 时表示正端电压小于负端电压。SRAD[11:0]为差值，带符号位的补码。

比较通道 0 和通道 1 的电压值，通道 0 接比较器正端，通道 1 接比较器负端。

```
…
CLRF SRADCON1        ;VDD 为参考电压,often=0,calif=0;enov=0,offex=0,vrefs=00
BSF SRADCON1,5       ;often=1,结果保存在 sroft 寄存器中
MOVLW 00h
MOVWF SRADCON2       ;chs[3:0]=0000，选择通道 0 作为比较器负端
BSF SRADCON1,7       ;使能 ADC 模块
CALL delay_40us
BSF SRADCON1,6       ;srads=1,开始转换
BTFSC SRADCON1,6     ;检测转换是否完成
GOTO $-1
…
MOVLW 10h
MOVWF SRADCON2       ;chs[3:0]=0001，选择通道 1 作为比较器正端
BCF SRADCON1,5       ;often=0
BSF SRADCON1,4       ;calif=1
BSF SRADCON1,3       ;enov=1
BSF SRADCON1,6       ;srads=1,开始转换
BTFSC SRADCON1,6     ;检测转换是否完成
GOTO $-1
BTFSC sradh,3
GOTO le_cmp          ;正端电压小于负端电压
GOTO gt_cmp          ;正端大于等于负端电压
…
```

比较 1V 电压和通道 1 的电压，通道 1 接比较器正端，1V 接比较器负端，假设采用 5V 的 VDD 作为参考电压，那么 1V 的 AD 值为 0x333。

```
…
CLRF SRADCON1        ;VDD 为参考电压,often=0,calif=0;enov=0,offex=0,vrefs=00
MOVLW 10h
MOVWF SRADCON2       ;chs[3:0]=0001，选择通道 1 作为比较器正端
BSF SRADCON1,4       ;calif=1
BSF SRADCON1,3       ;enov=1
MOVLW 03h
MOVWF srofth
MOVLW 33h
MOVWF sroftl         ;sroft 寄存器存入 333h，即 1V 作为比较器负端
BSF SRADCON1,7       ;使能 ADC 模块
CALL delay_40us
BSF SRADCON1,6       ;srads=1,开始转换
BTFSC SRADCON1,6     ;检测转换是否完成
GOTO $-1
BTFSC sradh,3
GOTO le_cmp          ;正端电压小于负端电压
GOTO gt_cmp          ;正端大于等于负端电压
…
```

2.6.9. 内部测量 VDD 的电压

用户可以通过使用内部参考电压或者外部参考电压输入（外部参考电压固定且不随 VDD 电压变化）两种方法来测试芯片内部 VDD 的电压。

使用外部参考电压，需额外提供参考源。

使用内部参考电压不需要额外的硬件条件。但是，使用内部参考电压会由于本身内部参考电压值的不准而影响精度。可以通过内部参考电压校正来提高测试的精度。

示例：外接 3V 作为参考电压，测 VDD 电压。选择通道 5，测出 1/8VDD 的 AD 值，之后乘以 8 得出 VDD 的 AD 值，再乘以参考电压则为 VDD 电压。

```
…
CLRF    SRADCON1        ;often=0,calif=0;enov=0,offex=0,vrefs=00
BSF     SRADCON1,0      ;vrefs=01,选择外部参考电压，接 3V
MOVLW 50h
MOVWF SRADCON2          ;chs[3:0]=0101，选择通道 5 ,1/8VDD
BSF     SRADCON1,7      ;使能 ADC 模块
CALL    delay_40us
BSF     SRADCON1,6      ;srads=1,开始转换
BTFSC SRADCON1,6    ;检测转换是否完成
GOTO    $-1
MOVLW sradl
MOVWF adtmp1
MOVLW sradh
MOVWF adtmph
BCF     status,c
RLF     adtmp1
RLF     adtmph          ;AD 值乘以 2
RLF     adtmp1
RLF     adtmph          ;AD 值乘以 4
RLF     adtmp1
RLF     adtmph          ;AD 值乘以 8,小数点在 adtmph 的 bit3 和 bit4 之间
…
```

2.7. I²C 从机

芯片内部集成了一个 I²C 从机模块，可支持 8 位双向的数据收发，该模块具有以下特性：

1) 符合 I²C 总线规范，可以支持标准模式（100K bit/s）和快速模式（400K bit/s）数据传输
2) 支持 7 位地址寻址，默认从机地址为 0x26，从机地址可配置
3) 支持广播地址呼叫
4) 支持时钟低电平延长
5) 支持 SCL/SDA 输入滤波
6) 支持从机接收模式和从机发送模式
7) 接收响应状态（ACK or NACK）软件可配置
8) 支持 SLEEP 模式异步唤醒，支持 HALT 模式唤醒

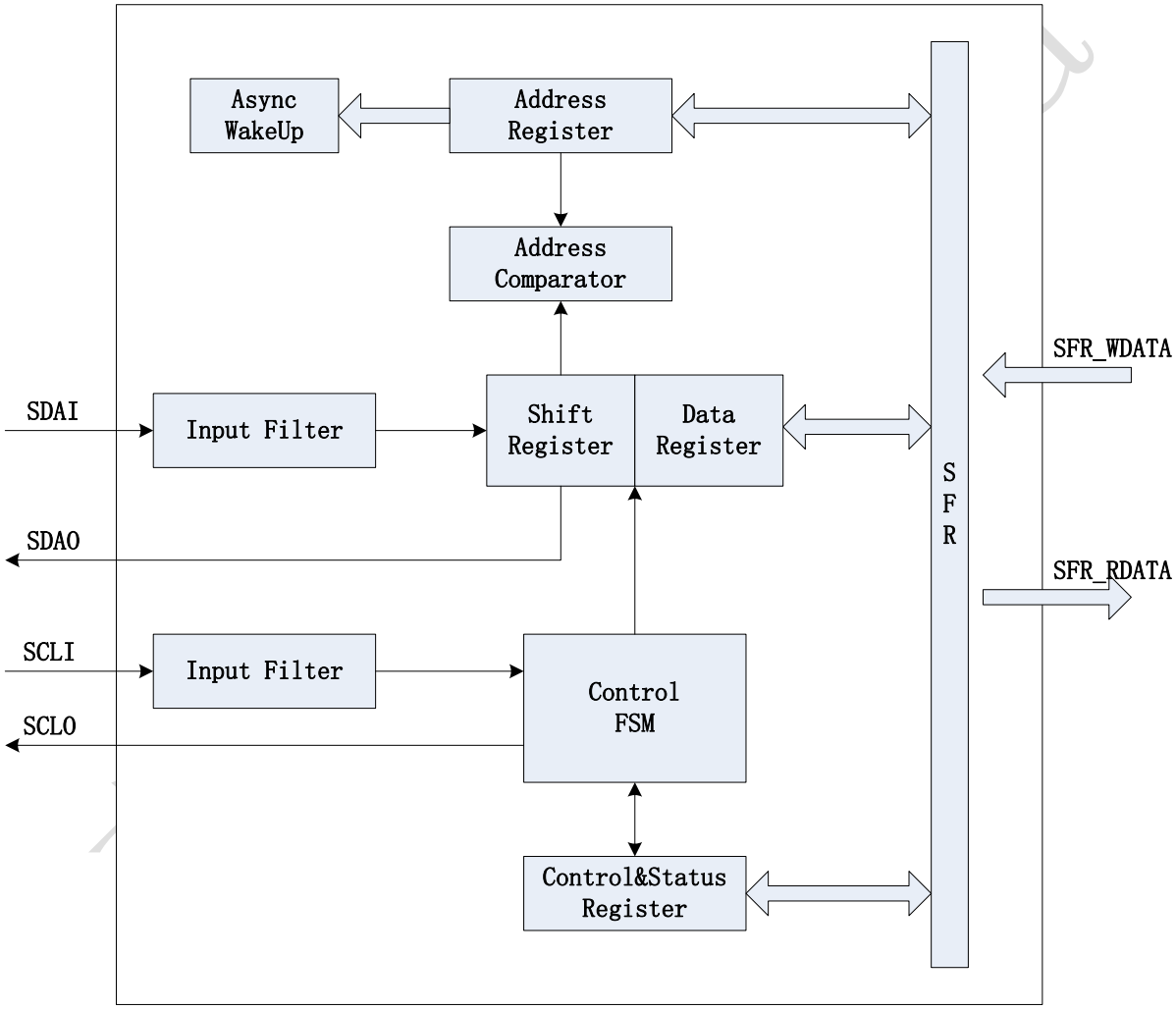


图 35　I²C 功能框图

2.7.1. I²C 数据传输起始和终止条件

START（开始条件）：当 SCL 线为高电平时，SDA 线由高电平向低电平切换，即出现下降沿，则被视为 I²C 传输的起始标志。

STOP（停止条件）：当 SCL 线为高电平时，SDA 线由低电平向高电平切换，即出现上升沿，则被视为 I²C 传输的结束标志。

数据有效性：在数据传输过程中，SDA 线上的电平只允许在 SCL 为低电平时改变，SCL 为高电平时

应保持稳定。

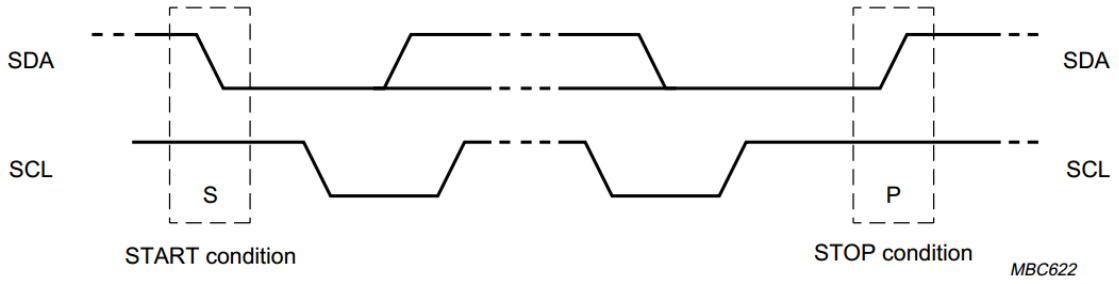


图 36　I²C 起始条件与终止条件时序图

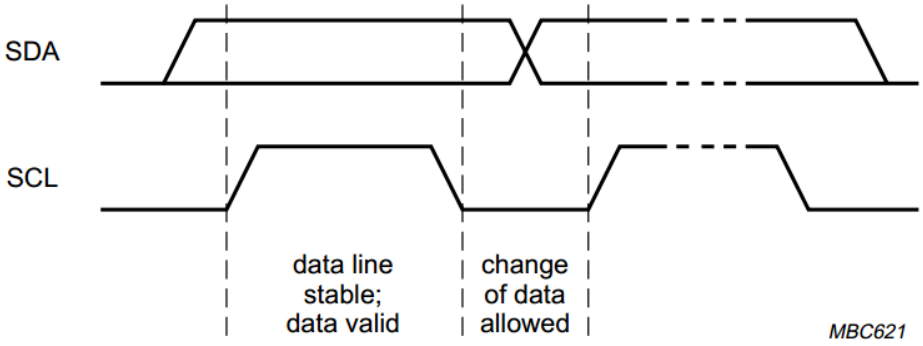


图 37　I²C 读写时序图

2.7.2.　I²C 总线特性

I²C 总线特性如下表所示：

表 15　I²C 总线特性表

参数	符号	标准模式		快速模式		单位
		最小值	最大值	最小值	最大值	
SCL 时钟频率	F_{SCL}		100		400	KHz
SCL 变低至 SDA 数据输出及应答时间	t_{AA}	100		100		nS
新的发送开始前总线空闲时间	t_{BUF}	$2*T_{i2c\_clk}$		$2*T_{i2c\_clk}$		nS
起始信号保持时间，在这个周期后产生第一个时钟脉冲	$t_{HD:STA}$	T_{i2c_clk}		T_{i2c_clk}		nS
时钟低电平周期	t_{LOW}	T_{i2c_clk}		T_{i2c_clk}		nS
时钟高电平周期	t_{HIGH}	T_{i2c_clk}		T_{i2c_clk}		nS
起始信号建立时间	$t_{SU:STA}$	T_{i2c_clk}		T_{i2c_clk}		nS
数据输入保持时间	$t_{HD:DAT}$	50		50		nS
数据输入建立时间	$t_{SUI:DAT}$	50		50		nS
SDA 及 SCL 上升时间	t_R	1000		300		nS
SDA 及 SCL 下降时间	t_F	300		300		nS
停止信号建立时间	$t_{SU:STO}$	100		100		nS
数据输出保持时间	t_{DH}	100		100		nS

I²C_DIV[1:0]	分频数	T_{i2c_clk} (MCK=32M)
00	MCK/2	62.5ns
01	MCK/4	125ns
10	MCK/8	250ns
11	MCK/16	500ns

总线时序：

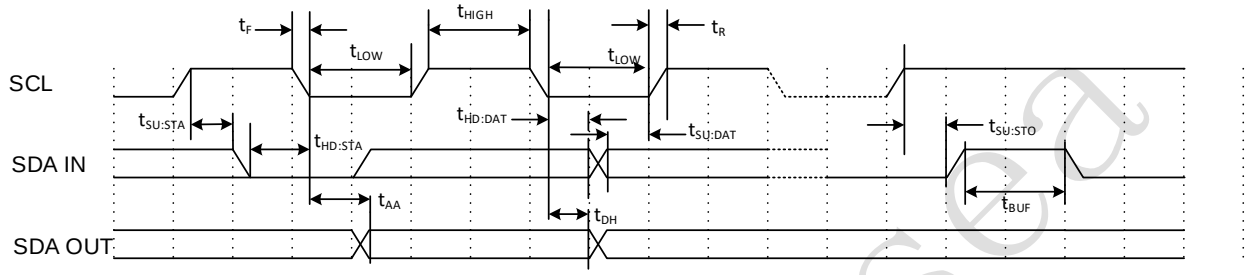


图 38　I²C 总线时序图

2.7.3.　7 位地址寻址

I²C 数据传输遵循下图所示格式。在起始条件（S）后，I²C 主机随后发送从机地址，从机地址一共有 7 位，紧跟从机地址的第 8 位作为读写标志位（R/W），该位为 0 则表示主机向从机发送数据，为 1 则表示主机需向从机读取数据，因此该位决定了数据的传输方向。从机接收到 7 位地址信号后，会对从机地址进行比对，若从机地址匹配就向主机发送应答信号（ACK），如果从机地址不匹配，从机则不响应该寻址，即发送 NACK，随后主机会发出 STOP 条件，终止本次数据传输。

当主机需要进行广播寻址时，只需要将发送的从机地址改为 0 即可。此时主机通过广播呼叫来寻址挂载在 I²C 总线上的每一个器件，支持广播呼叫功能的从机器件如果需要与主机进行数据交换，则必须发送应答信号（ACK）。

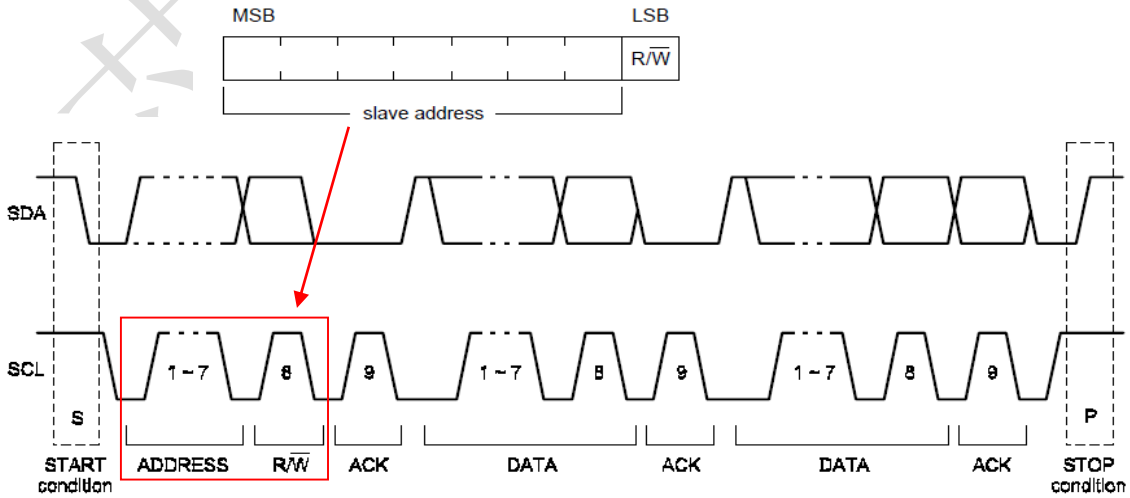


图 39　I²C7 位地址寻址时序图

2.7.4. 数据传输时序

主机和从机之间的数据传输遵从从高位到低位的顺序，即以 MSB 的方式传输。

主机发送数据，从机接收数据时序如下图

主机接收数据，从机接收数据时序如下图

复合传输格式，传输过程中改变方向的时候，起始条件和从机地址都会重复发送，但是 R/W 位取反，时序图如下图。

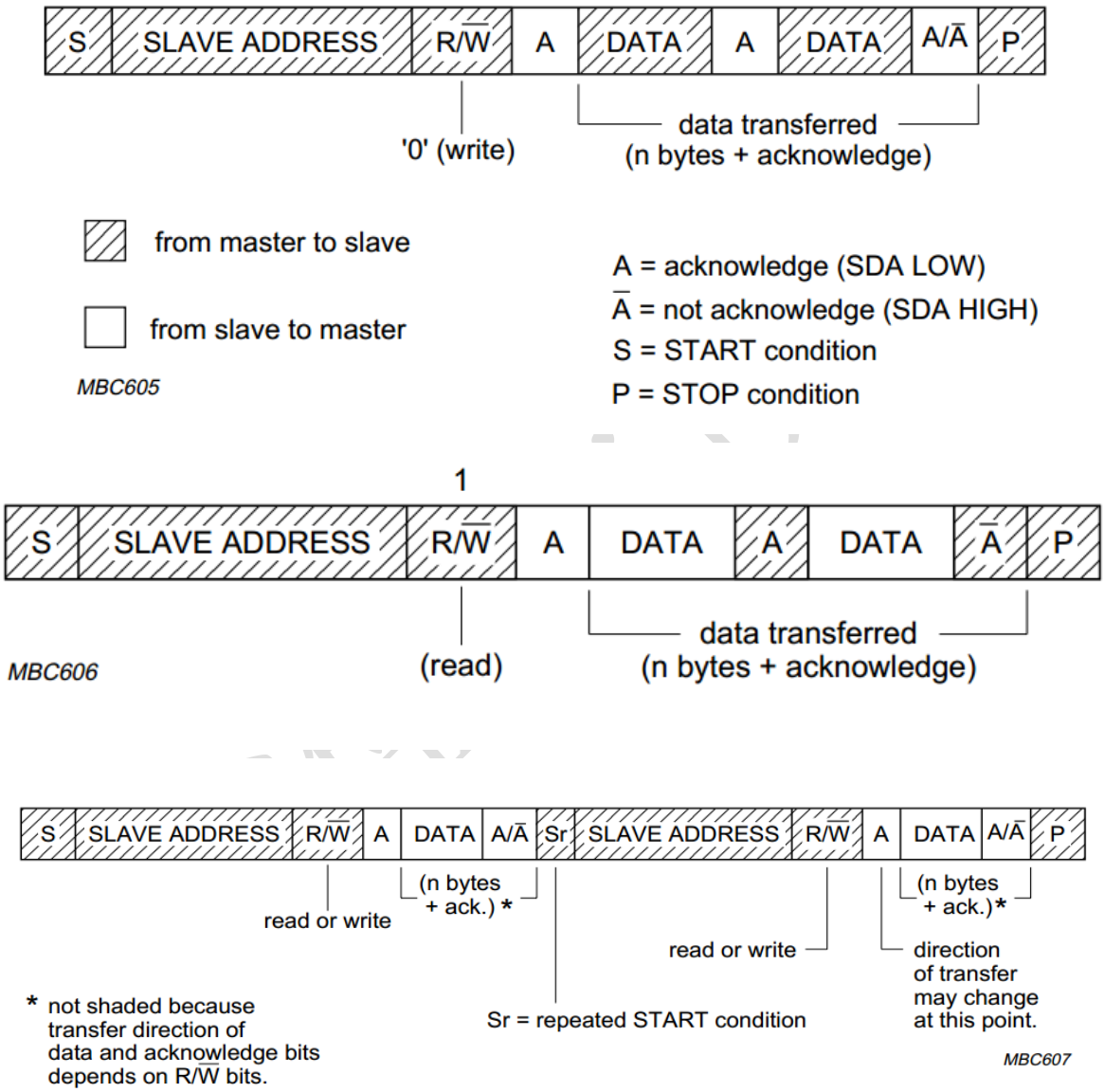


图 40　I²C 数据传输时序图

2.7.5. 寄存器说明

表 16　I²C 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
3ch	INTF2							I2CIF		0000u000
59h	I2CCON	I2C_EN	AWK_EN	CST_EN	ACK_EN	I2CSTUS[3:0]				00000000
5ah	I2CCON1	I2C_SEL						I2C_DIV		0uuuuu00

5bh	I2CDAT	I2CDAT[7:0]								00000000
5ch	I2CADR	I2CADR[6:0]							GC_EN	01001100
5dh	I2C_INTF	I2C_TIF	I2C_RIF	I2C_STIF						000uuuuu
5eh	I2C_INTE	I2C_TIE	I2C_RIE	I2C_STIE						000uuuuu

2.7.5.1. I2CCON 寄存器（地址为 59h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-0	R-0
I2CCON	I2C_EN	AWK_EN	CST_EN	ACK_EN	I2CSTUS[3:0]			

位地址	标识符	功能
7	I2C_EN	I²C 从机使能信号 1：I²C 从机开启 0：I²C 从机关闭
5	AWK_EN	I²C 异步唤醒使能信号 1：开启异步唤醒功能，在系统处于 SLEEP 模式时，如果 I²C 从机接收到主机发送的地址信号与本机的地址匹配，则通过 I2C_RIF 给出中断，唤醒系统。若主机发送的从机地址为 0，且 GC_EN=1、AWK_EN=1 的情况下，也能够利用 I2C_RIF 唤醒系统 0：关闭异步唤醒功能
5	CST_EN	I²C 从机时钟低电平延长使能 1：I²C 从机时钟低电平延长功能开启 0：I²C 从机时钟低电平延长功能关闭
4	ACK_EN	I²C 从机从机响应使能 1：I²C 从机响应开启（回复 ACK） 0：I²C 从机响应关闭（回复 NACK）
3：0	I2CSTUS[3:0]	I²C 从机状态寄存器，**只读** 4’h0：I2C_IDLE　空闲 4’h1：I2C_STDET　等待开始条件 4’h2：I2C_ADDR　接收从机地址 4’h3：I2C_ADDRACK 地址接收完成且匹配时，返回 ACK 4’h4：I2C_RX　接收数据 4’h5：I2C_RXACK　数据接收完，返回 ACK 4’h6：I2C_RXNACK　数据接收完，不回 ACK 4’h7：I2C_TX　发送数据 4’h8：I2C_TXACK　数据发送完成后，等待主机发送 ACK 4’h9：I2C_CLKSTR　从机没准备好，拉低时钟即低电平延长 4’hF：I2C_ERR　接收和发送流程有错误

2.7.5.2. I2CCON1 寄存器（地址为 5ah）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
I2CCON1	I2C_SEL[1:0]						I2C_DIV[1:0]	

位地址	标识符	功能
7	I2C_SEL	I²C 接口选择 1：PT3.2 作为 I²C 的 SDA；PT3.1 作为 I²C 的 SCL 0：PT5.2 作为 I²C 的 SDA；PT5.3 作为 I²C 的 SCL
5：2	RESERVE	保留
1：0	I2C_DIV[1:0]	I²C 从机的时钟分频系数

I2C_DIV[1:0]	分频数（MCK＝32MHz）	频率 f（单位：MHz）	周期 T（单位：ns）
00	MCK/2	16	62.5
01	MCK/4	8	125
10	MCK/8	4	250
11	MCK/16	2	500

2.7.5.3. I2CDAT 寄存器（地址为 5bh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
I2CDAT	I2CDAT[7:0]							

位地址	标识符	功能
7：0	I2CDAT[7:0]	I²C 数据寄存器 当 I²C 工作在从机接收模式时，完成 1Byte 数据接收后，CPU 就可以通过 SFR 总线从该寄存器读出接收的数据。 当 I²C 工作在从机发送模式时，CPU 利用 SFR 总线将需要发送的数据写入该寄存器中。

2.7.5.4. I2CADR 寄存器（地址为 5ch）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-1	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
I2CADR	I2CADR[6:0]							GC_EN

位地址	标识符	功能
7：1	I2CADR[6:0]	I²C 从机地址寄存器，默认为 0x26
0	GC_EN	I²C 从机广播地址响应使能 1：广播地址响应功能开启 0：广播地址响应功能关闭

2.7.5.5. I2C_INTF 寄存器（地址为 5dh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
I2C_INTF	I2C_TIF	I2C_RIF	I2C_STIF					

位地址	标识符	功能
7	I2C_TIF	I²C 从机发送中断标志，硬件清零

位地址	标识符	功能
		1：发生了 I²C 从机发送中断（I²C 使能且发送缓存为空状态） 0：未发生 I²C 从机发送中断（往 I2CDAT 写入数据或硬件清零）
6	I2C_RIF	I²C 从机接收中断标志，硬件清零 当 I2C_EN=1 且 AWK_EN=0 时： 1：I²C 从机完成 1Byte 数据接收，且发送 ACK 后产生接收中断，必须软件清零 0：未发生 I²C 从机接收中断 当 I2C_EN=1 且 AWK_EN=1 时： 1：在 SLEEP 模式下，I²C 从机完成地址匹配产生接收中断，唤醒系统，系统唤醒后需要软件将 AWK_EN 清零 0：未发生 I²C 从机地址匹配中断
5	I2C_STIF	I²C 从机起始中断，软件清零 1：发生了 I2C_START 中断，从机接收到起始位 0：未发生 I2C_START 中断，或软件清零后
4：0	RESERVE	保留

2.7.5.6. I2C_INTE 寄存器（地址为 5eh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
I2C_INTE	I2C_TIE	I2C_RIE	I2C_STIE					

位地址	标识符	功能
7	I2C_TIE	I²C 从机发送中断使能 1： I²C 从机发送中断功能开启 0： I²C 从机发送中断功能关闭
6	I2C_RIE	I²C 从机接收中断使能 1：I²C 从机接收中断功能开启 0：I²C 从机接收中断功能关闭
5	I2C_STIE	I²C 从机起始中断使能 1：I2C_START 中断功能开启 0：I2C_START 中断功能关闭
4：0	RESERVE	保留

注：支持 I²C 从机发送提前写入功能的流程如下：

使用提前写入的功能的前提是，从机后续的传输是发送数据，或主机即将读取从机的数据。此时，使能 I2C_START 中断功能，在系统接收到 I2C_START 中断时，CPU 将需要发送的数据，提前写入至 I2CDAT 寄存器中，当 I²C 从机进入发送数据状态时，I2CDAT 寄存器中的数据早已准备完毕。在不开启 SCL 低电平延长功能或不支持 SCL 低电平延长功能的情况下，通过提前写入数据能够保证 I²C 数据传输的稳定性，避免出现处在发送状态但数据还未准备好所导致的通讯错误。

注：由于从起始标志到数据发送阶段，这中间仅仅只有 8 个 I²C 总线时钟周期，如果 I²C 总线跑在 400KHz，系统中断中需要处理的内容过多，此时即使提前打开写入功能，还是会出现系统来不及写入数据，导致通讯错误的情况。

2.7.6. I²C 使用步骤

2.7.6.1. I²C 从机的初始化

根据 I²C 的工作频率，需要配置 I²C 从机的时钟分频系数 I2C_DIV[1:0]，具体配置参照下面的表格。

I2C_DIV[1:0]	分频数（MCK = 32MHz）	频率 f（单位：MHz）
00	MCK/2	16
01	MCK/4	8
10	MCK/8	4
11	MCK/16	2

I²C 通信速率（单位：Kbit/s）	MCK（单位：MHz）	I2C_DIV[1:0]
400	32	00、01、10
100	32	00、01、10、11

1) 配置 I²C 从机地址 I2CADR[7:1]，默认为 0x26

2) 使能 I²C 从机，关闭异步唤醒功能，I2C_EN=1，AWK_EN=0，根据需求使能低时钟延长，CST_EN=1，使能从机响应，ACK_EN=1。

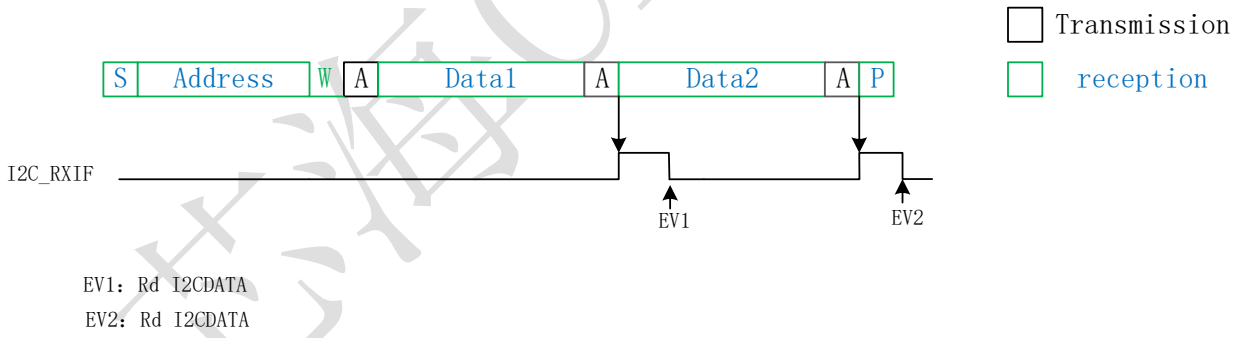
3) I²C 从机自动检测 I²C 主机发送的起始状态 START 或 RESTART，并把接收到的地址跟本机地址进行比较，如果地址匹配则发送响应给主机，准备收发数据。

2.7.6.2. I²C 从机的接收

当 I2CDAT 寄存器接收到数据后，会将 I2C_RIF 标志位拉起，如果中断使能 I2C_RIE 打开，则中断会上传，此时如果总中断 GIE 是开启状态，CPU 就会响应 I²C 的接收中断。

当 CPU 读取 I2CDAT 寄存器 中的数据后，I2C_RIF 标志会被硬件自动清除。

注：当 I²C 数据溢出后，会发送 NACK

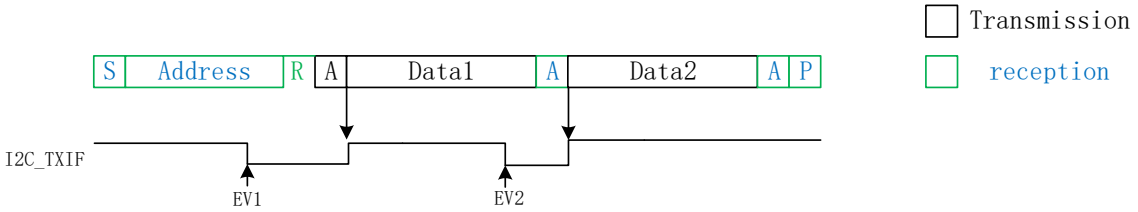


EV1：Rd I2CDATA
EV2：Rd I2CDATA

2.7.6.3. I²C 从机的发送

未开启提前写入功能的流程如下：

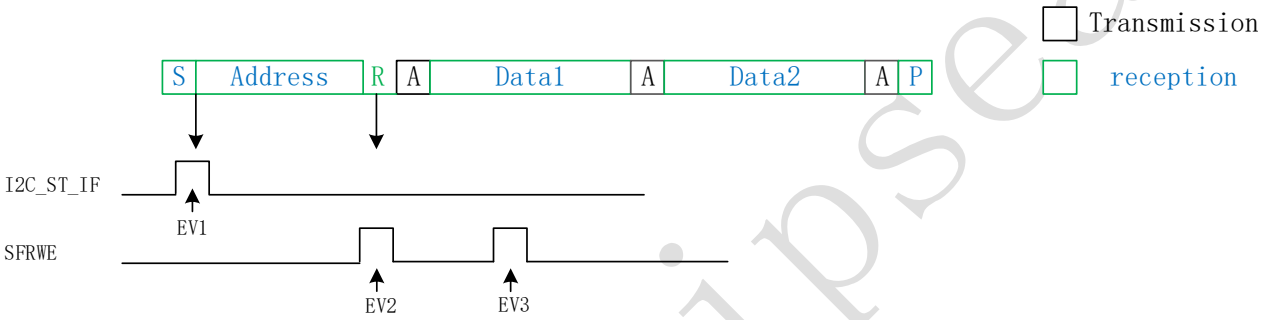
1) 当 I2CDAT 为空时，I2C_TIF 的标志位拉高，上电默认值也是高电平。如果此时中断使能 I2C_TIE 打开，就会产生中断，此时可以写入数据至 I2CDAT 寄存器。当硬件读取 I2CDAT 寄存器或 I²C 从机接收到数据后，I2C_TIF 标志就会被硬件清除。



EV1：Wr I2CDATA
EV2：Wr I2CDATA

开启提前写入功能的流程如下：

1)　当 I2CDAT 为空时，I2C_TIF 的标志位拉高，上电默认值也是高电平。如果此时中断使能 I2C_STIE 打开，就会产生 START 中断，此时可以提前写入数据至 I2CDAT 寄存器。当硬件读取 I2CDAT 寄存器或 I²C 从机接收到数据后，I2C_TIF 标志就会被硬件清除。



EV1：I2C Start Interrupt
EV2：Write First Transmit I2C DATA
EV3：Write Second Transmit I2C DATA

注：当时序上需要 I²C 从机发送数据，此时的 I2CDAT 为空且没有打开时钟延长功能 CST_EN=0，若 I²C 从机地址匹配，则回复 ACK，并跳出发送状态，如果从机地址不匹配，则直接回复 NACK。

2.7.6.4.　I²C 从机低功耗模式下唤醒

在系统进入 SLEEP 模式之前，配置 I2C_EN=0，AWK_EN=1，CST_EN=0，ACK_EN=0。系统进入 SLEEP 模式后，如果 I²C 从机接收到主机发送的地址信号与本机的地址匹配，则通过 I2C_RIF 给出中断，唤醒系统。若主机发送的从机地址为 0，且 GC_EN=1、AWK_EN=1 的情况下，也能够利用 I2C_RIF 唤醒系统。系统唤醒后，在正常工作模式需要把异步唤醒功能关闭 AWK_EN=0。为了简化应用，不建议把 I²C 从机异步唤醒功能和 I²C 从机正常数据收发功能同时使用。

2.8. 串行通信接口

芯片提供 1 个独立可编程全双工异步串行通信接口，具有如下特性：

1) 支持数据同时发送和接收
2) 支持接收滤波消除毛刺
3) 波特率可配，最高速率能够支持 115200bps
4) 支持 8/9 位数据发送/接收
5) 支持 1 级发送 FIFO 和 8 级接收 FIFO
6) 支持 SLEEP 模式下接收唤醒
7) TX/RX 管脚交换可配置

2.8.1. 波特率配置

UART0 的波特率配置公式，$$BR = \frac{UART0\_CLK}{BRR0 + \frac{BRR1}{10}}$$。

其中 $$UART0\_CLK = \frac{ICK}{2^{UART0\_DIV+1}}$$，UART0_DIV 的范围 为 0~5；BRR1 的范围为 0~9。

例如：MCK=32MHz，若要以 115200bps 波特率通讯，采用 UART_CLK0=4MHz，进行第一步的计算，计算出 BRR0= 34, BRR1 = 7；如果 BRR0 的结果小于 100，可以分频系数保持不变。如果 BRR0 计算的结果大于 100，那么分频系数 UART_DIV0 加 1。**分频系数越大，功耗就越低。**

注意，串口支持接收滤波，可消除接收信号毛刺，提高通信可靠性。串口可滤除毛刺的最大宽度为 T_{UART_CLK}*7。

2.8.2. 发送流程

发送的 TXFIFO 深度为 1，可以缓存一个字节的数据。

发送流程为：

1) 设置 BRR0/BRR1，选择合适的波特率
2) 配置 URAT_EN=1，使能 UART0 串口
3) 如果需要发送第 9 位，将 TX9_EN 置 1
4) 将数据写入发送缓存寄存器 UR0_TXREG
5) 置位 TX_EN=1
6) 当发送移位寄存器中的数据传输完成后，若此时 TXFIFO 有数据，硬件自动从 TXFIFO 中读取数据并发送，读取后 TXFIFO 为空，发送中断置位，此时可将下一个数据写入发送缓存寄存器中。

2.8.3. 接收流程

接收的 RXFIFO 深度为 1，可以缓存一个字节的数据。

发送流程为：

1) 设置 BRR0/BRR1，选择合适的波特率
2) 配置 URAT_EN=1，使能 UART0 串口
3) 如果需要接收第 9 位，将 RX9_EN 置 1
4) 置位 RX_EN=1
5) 当接收完成，硬件置位 UR0_RIF，会进入接收中断。若从 RXFIFO 读取数据后 UR0_RIF 仍为

高电平，说明缓存中还有数据，知道 FIFO 中的数据全部读取之后，UR0_RIF 才会被硬件清零

6) 在 SLEEP 模式下，当有数据接收时（RX 端口被拉低），串口唤醒中断置位，若此时串口唤醒中断使能开启的情况下，可唤醒 SLEEP 模式。

2.8.4. 寄存器说明

表 17　串口寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
3ch	INTF2								UR0IF	0000u000
50h	UR0_CR1	TX9D	RX9D	TX9_EN	RX9_EN	RX_EN	TX_EN	UART0_SEL	UART0_EN	00000000
51h	UR0_BRR0	BRR0[7:0]								00000000
52h	UR0_BRR1		UARTDIV[2:0]			BRR1[3:0]				u0000000
53h	UR0_TX_REG	TX_REG[7:0]								00000000
54h	UR0_RX_REG	RX_REG[7:0]								00000000
55h	UR0_ST	UR_TINV	UR_RINV	UR_SWAP	TXFIFO_EMPTY	RX_BUSY	TX_BUSY	RX_OV_ERR	STOP_ERR	00010000
56h	UR0_INTF				UR0ERRIF	UR0_RHIF	UR0_RNIF	UR0WKIF	UR0_TEIF	uuu00000
57h	UR0_INTE				UR0ERRIE	UR0_RHIE	UR0_RNIE	UR0WK_IE	UR0_TEIE	uuu00000

2.8.4.1. UR0_CR1 寄存器（地址 50h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
I2CCON	TX9D	RX9D	TX9_EN	RX9_EN	RX_EN	TX_EN	UART0_SEL	UART0_EN

位地址	标识符	功能
7	TX9D	发送数据第 9 位 1：发送的第 9 位数据为 1 0：发送的第 0 位数据为 0
6	RX9D	接收数据第 9 位，该位只读，不可写 1：接收的第 9 位数据为 1 0：接收的第 9 位数据为 0
5	TX9_EN	发送第 9 位数据使能信号 1：发送第 9 位数据 0：不发送第 9 位数据
4	RX9_EN	接收第 9 位数据使能信号 1：接收第 9 位数据 0：不接收第 9 位数据
3	RX_EN	数据接收控制信号 1：允许接收数据 0：禁止接收数据

位地址	标识符	功能
2	TX_EN	数据发送控制信号 1：允许发送数据 0：允许发送数据
1	UART0_SEL	串口通信 0 接口选择信号 1：PT5.2 作为串口 0 的 RX；PT5.3 作为串口 0 的 TX 0：PT3.2 作为串口 0 的 RX；PT3.1 作为串口 0 的 TX 注：同时使用 I²C 和 UART0 时，二者不能选择相同的引脚
0	UART0_EN	串口 0 模块使能控制信号 1：打开 UART0 0：关闭 UART0

2.8.4.2. UR0_BRR0 寄存器（地址为 51h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UR0_BRR0	BRR0[7:0]							

位地址	标识符	功能
7：0	BRR0[7:0]	波特率设置寄存器 0，整数部分

2.8.4.3. UR0_BRR1 寄存器（地址为 52h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UARTDIV		UARTDIV[2:0]			BRR1[3:0]			

位地址	标识符	功能
7	RESERVE	保留
6：4	UARTDIV[2:0]	UART 时钟分频选择 UARTDIV：UART_CLK 000：MCK/2 001：MCK/4 010：MCK/8 011：MCK/16 100：MCK/32 其他：MCK/64
3：0	BRR1[3:0]	波特率设置寄存器 1，小数部分

UART 的波特率配置公式，$$BR = \frac{UART\_CLK}{BRR0 + \frac{BRR1}{10}}$$。

2.8.4.4. UR0_TX_REG 寄存器（地址为 53h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

UR0_TX_REG	TX_REG[7:0]

位地址	标识符	功能
7：0	TX_REG[7:0]	串口发送数据寄存器

2.8.4.5. UR0_RX_REG 寄存器（地址为 54h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UR0_RX_REG	RX_REG[7:0]							

位地址	标识符	功能
7：0	RX_REG[7:0]	串口接收数据寄存器

2.8.4.6. UR0_ST 寄存器（地址为 55h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-1	R-0	R-0	R/W-0	R/W-0
UR0_ST	UR_TINV	UR_RINV	UR_SWAP	TXFIFO_EMPTY	RX_BUSY	TX_BUSY	RX_OV_ERR	STOP_ERR

位地址	标识符	功能
7	UR_TINV	UART 输入输出信号取反控制信号 1：使能 UART 输出信号取反 0：禁止 UART 输出信号取反（默认）
6	UR_RINV	UART 输入输出信号取反控制信号 1：使能 UART 输入信号取反 0：禁止 UART 输入信号取反（默认）
5	UR_SWAP	交换 TX/RX 引脚 0：TX/RX 引脚功能不交换 1：TX/RX 引脚功能交换使用 软件只能在 UART0_EN=0 时写该位
4	TXFIFO_EMPTY	发送 FIFO 空标志，可在中断中清除 TX_TEIF 后判断发送 FIFO 是否为空状态 1：发送 FIFO 为空 0：发送 FIFO 不为空
3	RX_BUSY	接收 BUSY 指示信号 1：接收端正在接收 0：接收端未进行接收，处于空闲状态 注：当进入 SLEEP 后，系统可以采用 UART 唤醒中断进行唤醒。系统被唤醒后，可以查询该位，若 RX_BUSY=0，表示此次唤醒可能是由于干扰造成。
2	TX_BUSY	发送寄存器 TX_REG 的数据是否全部串行发送完毕 1：还未全部发送完毕 0：已经全部发送完毕

位地址	标识符	功能
1	RX_OV_ERR	接收 FIFO 溢出错误标志 1：发生了接收 FIFO 溢出的错误 0：未发生接收 FIFO 溢出的错误
0	STOP_ERR	接收停止位错误标志 1：发生了停止位错误 0：未发生停止位错误

2.8.4.7. UR0_INTF 寄存器（地址为 56h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	R-0	R-0	R-0	R/W-0	R/W-0
UR0_INTF				UR0ERRIF	UR0_RHIF	UR0_RNIF	UR0WK_IF	UR0_TEIF

位地址	标识符	功能
7:5	RESERVE	保留
4	UR0ERRIF	串口 0 接收错误中断标志 1：发生了串口 0 接口错误中断 0：未发生串口 0 接收错误中断，或软件清除 RX_OV_ERR/STOP_ERR 后 当串口 0 接收停止位错误或 FIFO 溢出错误发生时，标志置位。需查验 UR0_ST 寄存器 RX_OV_ERR、STOP_ERR 确认错误类型，在清除 RX_OV_ERR、STOP_ERR 后，UR0EERIF 自动清零，不能直接清除 UR0ERRIF 标志
3	UR0_RHIF	串口 0 接收 FIFO 半满中断标志 1：发生了串口 0 接收 FIFO 半满中断，接收 FIFO 中数据长度大于等于 4 0：未发生串口 0 接收 FIFO 半满中断，接收 FIFO 中数据长度小于 4
2	UR0_RNIF	串口 0 接收 FIFO 非空中断标志 1：发生了串口 0 接收 FIFO 非空中断，接收 FIFO 不是空状态 0：未发生串口 0 接收 FIFO 非空中断，接收 FIFO 此时为空状态
1	UR0WK_IF	串口接收唤醒中断标志 1：使能串口接收唤醒中断 0：不使能串口接收唤醒中断 在 SLEEP 状态时，串口接收到数据起始位时置 1；在非 SLEEP 状态下串口接收到数据起始位时不触发该中断
0	UR0_TEIF	串口 0 发送 FIFO 空中断发生标志 1：发生了串口 0 发送 FIFO 空中断 0：未发生串口 0 发送 FIFO 空中断 当串口 0 发送 FIFO 空中断产生时，可向发送 FIFO 写入发送数据。需注意，当发送 FIFO 空中断产生时，串口仍处于发送数据状态，必须等待 UR0_ST 寄存器中的 TX_BUSY 标志拉低时，才表示数据发送完成，才能进入 SLEEP 状态。

2.8.4.8. UR0_INTE 寄存器（地址为 57h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UR0_INTE				UR0ERRIE	UR0_RHIE	UR0_RNIE	UR0WK_IE	UR0_TEIE

位地址	标识符	功能
7：5	RESERVE	保留
4	UR0ERRIE	串口 0 接收错误中断标志 1：开启串口 0 接口错误中断使能 0：关闭串口 0 接收错误中断使能
3	UR0_RHIE	串口 0 接收 FIFO 半满中断标志 1：开启串口 0 接收 FIFO 半满中断使能 0：关闭串口 0 接收 FIFO 半满中断使能
2	UR0_RNIE	串口 0 接收 FIFO 非空中断标志 1：开启串口 0 接收 FIFO 非空中断使能 0：关闭串口 0 接收 FIFO 非空中断使能
1	UR0WK_IE	串口接收唤醒中断标志 1：开启串口接收唤醒中断使能 0：关闭串口接收唤醒中断使能
0	UR0_TEIE	串口 0 发送 FIFO 空中断发生标志 1：开启串口 0 发送 FIFO 空中断使能 0：关闭串口 0 发送 FIFO 空中断使能

2.9. 数据查表

通过 MOVP 和 TBLP 指令可以实现对于用户程序存储器的数据读取和写入，用户程序存储器的地址范围为 000H~1FFFH。

2.9.1. 写操作解锁

TBLP 为写保护操作，对 FLASH 进行写操作时，必须解锁写保护。解锁写保护需对 WRPRT 寄存器连续写入 96H、69H、5AH ，对其他地址寄存器进行写操作时，WRPRT 寄存器会被清零，解锁自动失效。**执行 TBLP 指令时建议关闭全局中断使能，当写操作未解锁时写操作时间为 3 个指令周期。**

2.9.2. 寄存器描述

表 18　数据查表寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
05h	WORK	工作寄存器								00000000
0Ah	EADRH	EADR [15:8]								00000000
0Bh	EADRL	EADR [7:0]								00000000
0Ch	EDATH	EDATH[7:0]								00000000
60h	ISPCON1		ISPWDTRF						ISPOF	u0uuuuu0
63h	WRPRT								WRPRTF	uuuuuuu0
77h	TBLPEN	TBLPEN[7:0]								00000000

2.9.2.1. EADRH 寄存器（地址为 0Ah）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EADRH	EDAR[15:8]							

位地址	标识符	功能
7：0	EDAR[15:8]	读操作地址高位，该寄存器的低 3 位与 EADRL 组成 13 位的地址控制寄存器，可以访问 8K*16 空间。超出 8K 空间的地址读出数据为代码选项区或者无效数据。

2.9.2.2. EADRL 寄存器（地址为 0Bh）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EADRL	EDAR[7:0]							

位地址	标识符	功能
7：0	EDAR[7:0]	读操作地址低 8 位，该寄存器与 EADRH 组成 13 位的地址控制寄存器，可以访问 8K*16 空间。

注：8K*16 为 M32X 内核支持的最大空间，实际访问空间以产品实际的 Flash 大小为准。

2.9.2.3. EDATH 寄存器（地址为 0Ch）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EDATH	EDATH [7:0]							

位地址	标识符	功能
7：0	EDATH[7:0]	读取数据高 8 位，低 8 位数据存储在 WORK 寄存器中，一起组成 16 位数据。

2.9.2.4. ISPCON1 寄存器（地址为 60h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0
ISPCON1		ISPWDTRF						ISPOF

位地址	标识符	功能
6	ISPWDTRF	擦写操作时出现 WDT 复位标志位 0：擦写 FLASH 操作时未出现 WDT 复位 1：擦写 FLASH 操作时出现 WDT 复位 该位硬件置位，软件写 0 清零。
5:1	RESERVE	保留
0	ISPOF	写/擦除操作失败标志位。（硬件置位，软件清零，用户模式） 1’b0：写/擦除操作成功。 1’b1：写/擦除操作失败，写 0 清零 注： 1、如果不严格按照，清除 PAGE LATCH，页擦除，清除 PAGE LATCH，PAGE LATCH WRITE，页烧录的操作流程，则 ISPOF 会置位。 2、如果对 USR_W_SEC[x]=1 的区域进行擦写也会置位 ISPOF 3、当 TBLPEN[7:0] 不能与 5AH 时，进行擦写也会置位 ISPOF 4、用户模式下如果对 FLASH 出厂信息区和 ROW2 进行擦写，会置位 ISPOF

2.9.2.5. WRPRT 寄存器（地址为 63h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
WRPRT								WRPRTF

位地址	标识符	功能
0	WRPRTF	对 EEPROM 写操作进行解锁时，需对 WRPRT 寄存器连续写入 96H、69H、5AH。WRPRTF 表示解锁是否成功。 0：表示解锁失败 1：表示解锁成功

2.9.2.6. TBLPEN 寄存器（地址为 77h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TBLPEN	TBLPEN[7:0]							

位地址	标识符	功能
7:0	TBLPEN[7:0]	0101_1010 ：用户模式使能了 TBLP 操作 其它 ：用户模式禁止了 TBLP 操作

EADRH/EADRL 提供读操作的数据地址；
EDATH/WORK 提供读操作所用的数据。
读操作都是基于一个字（16 Bits）的。EDATH 寄存器只可读。

2.9.3. FLASH 读操作

对 FLASH 的读操作步骤如下：

- 配置{EADRH，EADRL}写操作地址。
- 执行指令 MOVP 操作
- 读取完成

执行读操作时，在地址寄存器输入相应的值，之后执行 MOVP 指令，便可在相应的 MTP 地址的数据读入到 EDATH/WORK 寄存器中。执行一次读操作需要 6 个指令周期。

```
MOVLW 03H
MOVWF EADRH    ;给高字节地址赋值
MOVLW 00H
MOVWF EADRL    ;给低字节地址赋值
MOVP           ;执行读操作
NOP
…
```

2.9.4. FLASH 写操作

FLASH 只能进行页擦除和页写入，每个页的大小是 16Bit*32,一共有 257 个页可以进行操作；主程序区和数据存储区的 256 个也可以进行擦写操作。

假如只想改写一个地址的数据，必须先页擦除，才能进行页写入；在页擦除之前要把当前页有用 MOVP 数据读出并进行保存，当页写入时在把该数据写入到 flash 对应的地址中。页擦除之后，FLASH 对应的页会清为零。

区域	FLASH 地址编 EADR[15:0]	
主程序	0000h	第 0 页
	0001h	
	0002h	
	……	
	001Fh	
	……	
	1F7Eh	
	1F7Fh	
	1F80h	
	……	
	1Fe1h	第 255 页
	……	
	1FFEh	
	1FFFh	
数据存储区	2000h	
	……	

区域	FLASH 地址编 EADR[15:0]	
	201Fh	
代码选项区	F000h	
	……	
	……	
	F01Fh	

TBLP 的操作模式列表：

操作模式	操作模式描述
TBLP 5A	清 PAGE LATCH 模式
TBLP 69	PAGE LATCH WRITE 模式
TBLP 96	页烧录模式
TBLP A5	页擦除模式

对 FLASH 的写操作步骤如下：

（1） 配置 TBLPEN 寄存器（地址为 77H）为 5Ah, 用户模式使能了 TBLP 操作
（2） 向 WRPRT 寄存器连续写入 96H、69H、5AH 结解锁 TBLP 操作
（3） 执行指令 TBLP 5AH 操作,清除 PAGE LATCH
（4） 配置擦除操作地址{EADRH，EADRL}
（5） 向 WRPRT 寄存器连续写入 96H、69H、5AH 结解锁 TBLP 操作
（6） 执行 TBLP A5 进行擦除操作。
（7） 向 WRPRT 寄存器连续写入 96H、69H、5AH 结解锁 TBLP 操作
（8） 执行指令 TBLP 5AH 操作,清除 PAGE LATCH
（9） 配置 PAGE LATCH 写地址{EADRH，EADRL}操作地址。
（10） 配置 PAGE LATCH 写数据高 8 位 EDATH
（11） 向 WRPRT 寄存器连续写入 96H、69H、5AH 结解锁 TBLP 操作
（12） 配置 PAGE LATCH 写数据低 8 位 WORK
（13） 执行指令 TBLP 69H 操作，把数据写进 PAGE LATCH。PAGE LATCH 大小为 1 个 PAGE(64byte=32*16bit)。如果想写入整块 PAGE LATCH 空间，跳回（9），重复步骤 9~13 共 32 次（即 PAGE LATCH 写地址从 0 到 1Fh, 也可以不整页写的，仅仅写想写入的地址）。
（14） 配置 PAGE PROGRAM 操作地址{EADRH，EADRL}。
（15） 向 WRPRT 寄存器连续写入 96H、69H、5AH 结解锁 TBLP 操作
（16） 执行指令 TBLP 96H 操作，进行 PAGE PROGRAM。
（17） 配置 TBLPEN 寄存器（地址为 77H）为 00h, 用户模式禁止 TBLP 操作

注：

1、由于 FLASH 只能进行 PAGE ERASE 的擦除，所有在对 FLASH 的地址写一个数据时，需要先把对应 PAGE 中的中的数据读出缓存，然后再通过 FLASH 的 page write 操作，才能改写对应地址中的数据。

2、PAGE LATCH WRITE 模式时,仅仅 EDAR[4:0]有效而 EDAR[15:5]无效，如果为 EDAR[4:0]=1Fh，则写向 PAGE LATCH 的地址为 1Fh 写数据{EDATH，WORK}。

3、页擦除模式，地址 EDAR[15:5]有效，地址 EDAR[4:0]无效，

4、页烧录模式，地址 EDAR[15:5]有效，而 EDAR[4:0]无效

5、一次页擦除，对应的页只能进行一次页烧录

6、清在 PAGE LATCH 模式后，PAGE LATCH 中的数据默认为 1

7、每次 PAGE LATCH 过数据以后，如果想在同一个地址 LATCH 新的数据，必须重新清在 PAGE LATCH 模式后以后再 WRITE LATCH

2.10. 在线调试功能（ICD）

2.10.1. 在线调试功能概述

CS8M32X 支持在线调试功能，通过 PT3.4 口（SWD）与芯片进行通信，配合 IDE 实现在线调试功能，SWD 为开漏输出口。

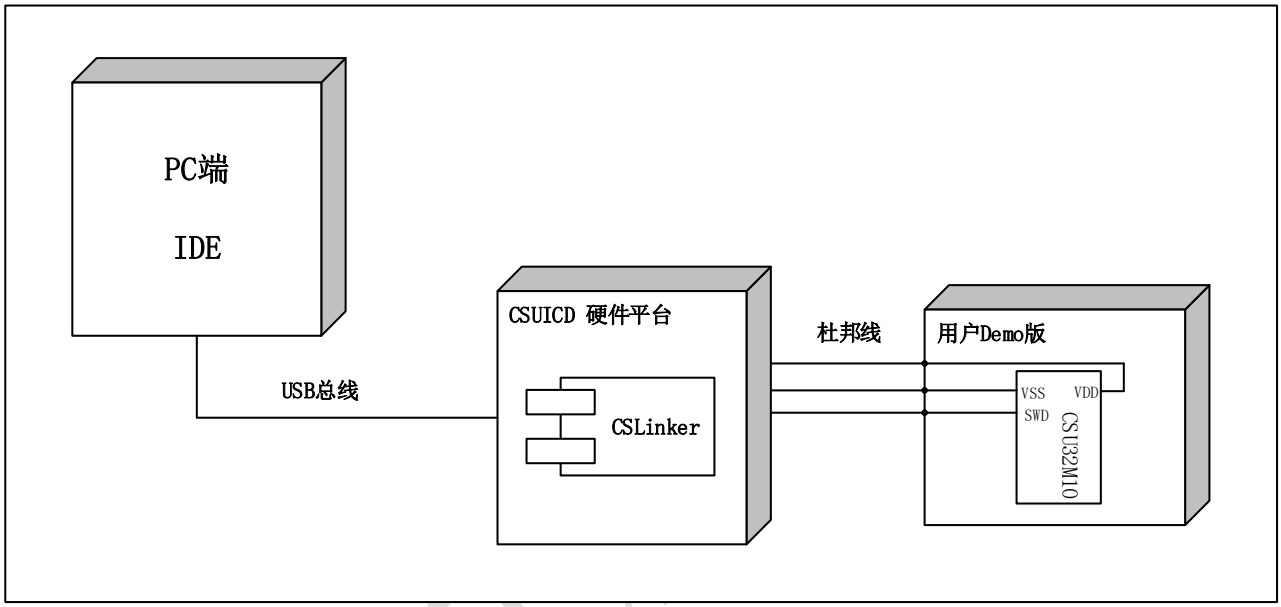


图 41　在线调试系统框图

2.11. 烧录模块

烧录器接口

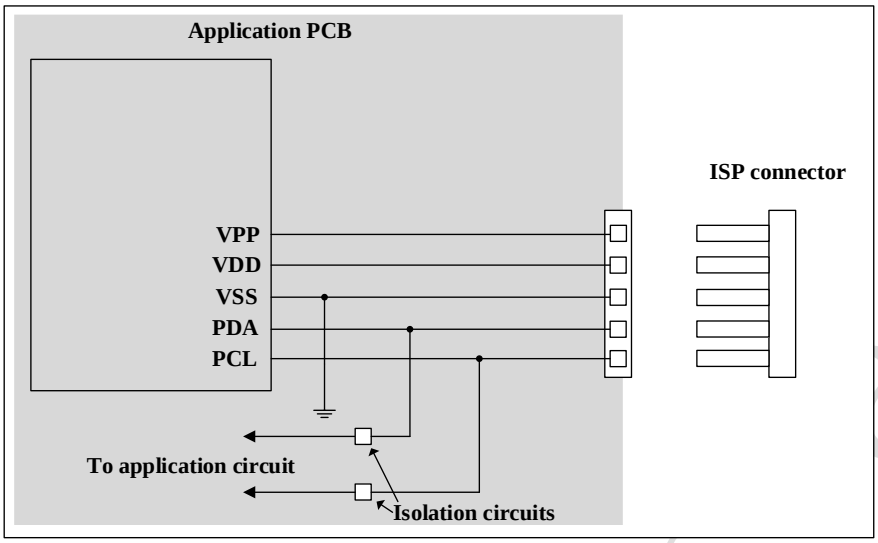


图 42　烧录器接口图

表 19　烧录接口说明

端口名称	型式	说明
VPP	输入	PT1[3]端口，烧录电源
VDD	输入	电源正端
VSS	输入	电源负端
PDA	输入/输出	PT1[4]端口，输入数据信号
PCL	输入	PT1[5]端口，时钟信号

2.11.1. 用户模式下访问烧录模式寄存器

在用户模式下可以访问烧录模式寄存器，但是需要配置一个专用的 PAGE 位，即 PAGECTL 的寄存器位 PAGE2。

2.11.1.1. PAGECTL 寄存器（地址为 14h）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
PAGECTL								PAGE2

位地址	标识符	功能
7:1	RESERVE	保留
0	PAGE2	PAGE 控制位 0：访问正常模式寄存器 1：访问烧录模式寄存器

2.12. 代码选项

2.12.1. OPTIION0

地址为 ID 区 0xF000。

代码选项数据的低 8 位和高 8 位必须为取反关系，否则芯片无法进入正常工作模式。

Bit-15	Bit-14	Bit-13	Bit-12	Bit-11	Bit-10	Bit-9	Bit-8
~RESET_PIN[1:0]		~LVR_SEL[1:0]				~LVR_EN	~SECURITY
Bit-7	**Bit-6**	**Bit-5**	**Bit-4**	**Bit-3**	**Bit-2**	**Bit-1**	**Bit-0**
RESET_PIN[1:0]		LVR_SEL[1:0]				LVR_EN	SECURITY

位地址	标识符	功能
15:14	~RESET_PIN[1:0]	必须为 RESET_PIN[1:0]的取反值
13:12	~LVR_SEL[1:0]	必须为 LVR_SEL[1:0]的取反值
11:10	RSV	保留
9	~LVR_EN	必须为 LVR_EN 的取反值
8	~SECURITY	必须为 SECURITY 的取反值
7:6	RESET_PIN[1:0]	复位引脚选择 00：PT1.3 作为普通 IO 口（默认） 01：PT1.3 作为 NRST 复位引脚 10：PT1.3 作为 PRST 复位引脚 11：PT1.3 作为普通 IO 口
5:4	LVR_SEL[1:0]	LVR 配置 LVR_SEL[1:0] 功能 00 1.8V/1.7V 上电/掉电复位（默认） 01 2.1V/2.0V 上电/掉电复位。 10 2.6V/2.5V 上电/掉电复位。 11 3.7V/3.6V 上电/掉电复位。
3:2	RSV	保留
1	LVR_EN	LVR 复位使能控制位 1：LVR 复位使能 （默认） 0：LVR 复位禁止
0	SECURITY	烧录模式代码保密位 0：禁止代码加密（默认） 1：使能代码加密 注：当加密时，所有区域不能进行 PAGE 擦写，读主程序区和 FLASH SPCIAL0（2000H~2001FH）时，读出的数据固定为 0000H；SPCIAL 1/2/3 可以正常读，在加密时，必须执行 CHIP_ERASE 后， 才能获得所有区域的擦写权限。

2.12.2. OPTIION1

地址为 ID 区 0xF001。

代码选项数据的低 8 位和高 8 位必须为取反关系，否则芯片无法进入正常工作模式。

Bit-15	Bit-14	Bit-13	Bit-12	Bit-11	Bit-10	Bit-9	Bit-8

SAR_LPOWER	WDT_CFG					~ILOPLEN	~ EMCEN
Bit-7	**Bit-6**	**Bit-5**	**Bit-4**	**Bit-3**	**Bit-2**	**Bit-1**	**Bit-0**
SAR_LPOWER	WDT_CFG					ILOPLEN	EMCEN

位地址	标识符	功能
15	~SAR_LPOWER	必须为 SAR_LPOWER 的取反值
14	~WDT_CFG	必须为 WDT_CFG 的取反值
9	~ILOPLEN	
8	~ EMCEN	
7	SAR_LPOWER	SAR_ADC 低功耗模式配置位 1：SAR_ADC 配置为正常功耗模式（默认） 0：SAR_ADC 配置为低功耗模式
6	WDT_CFG	WDT 模块使能和内部 32K 低速振荡器使能配置位 1：WDT 模块使能和内部 32K 低速振荡器使能由软件配置（默认） 0：WDT 模块使能和内部 32K 低速振荡器使能固定打开，软件无法修改。
1	ILOPLEN	非法指令复位使能 0：使能非法指令复位（使能） 1：禁止非法指令复位
0	EMCEN	EMC 复位使能 0：使能 EMC 复位（默认） 1：禁止 EMC 复位

2.12.3. ICD 功能配置选项

地址为 ID 区 0xF002。

Bit-15	**Bit-14**	**Bit-13**	**Bit-12**	**Bit-11**	**Bit-10**	**Bit-9**	**Bit-8**
Bit-7	**Bit-6**	**Bit-5**	**Bit-4**	**Bit-3**	**Bit-2**	**Bit-1**	**Bit-0**
				ICD_CFG[3:0]			

位地址	标识符	功能
15:4	RESERVE	保留
3:0	ICD_CFG[3:0]	ICD 模式使能选项 1111：使能 ICD 功能 其他：禁止 ICD 功能 注：1、在 ICD 功能使能并进入 SLEEP 时 32MHZ 时钟不关闭，这是为了保证在 SLEEP 时能进入 ICD 调试模式。如果想让 SLEEP 时 32HMz 关闭，必须配置禁止 ICD 功能。 2、在开 ICD 功能使能时，PT3[4]一直是数字输入口，不受 PT3[4]控制寄存器控制。

2.12.4. FLASH 进行 TBLP 加密位

地址为 ID 区 0xF003。

位地址	标识符	功能
13	~USR_W_SEC[5]	表示的取反 USR_W_SEC[5]
12	~USR_W_SEC[4]	表示的取反 USR_W_SEC[4]
11	~USR_W_SEC[3]	表示的取反 USR_W_SEC[3]
10	~USR_W_SEC[2]	表示的取反 USR_W_SEC[2]
9	~USR_W_SEC[1]	表示的取反 USR_W_SEC[1]
8	~USR_W_SEC[0]	表示的取反 USR_W_SEC[0]
5	USR_W_SEC[5]	空间 2000H~201FH(SPECIAL ROW0)写加密位: 1 表示加密，用户模式对应空间不能进行 TBLP 操作 0 表示不加密，用户模式对应空间能进行 TBLP 操作（默认）
4	USR_W_SEC[4]	空间 1800H~1FFFH 加密位: 1 表示加密，用户模式对应空间不能进行 TBLP 操作 0 表示不加密，用户模式对应空间能进行 TBLP 操作（默认）
3	USR_W_SEC[3]	空间（1000H~17FFH）加密位: 1 表示加密，用户模式对应空间不能进行 TBLP 操作 0 表示不加密，用户模式对应空间能进行 TBLP 操作（默认）
2	USR_W_SEC[2]	空间（0800H~0FFFH）加密位 1 表示加密，用户模式对应空间不能进行 TBLP 操作 0 表示不加密，用户模式对应空间能进行 TBLP 操作（默认）
1	USR_W_SEC[1]	空间（0400H~07FFH）加密位: 1 表示加密，用户模式对应空间不能进行 TBLP 操作 0 表示不加密，用户模式对应空间能进行 TBLP 操作（默认）
0	USR_W_SEC[0]	空间（0000H~03FFH）加密位: 1 表示加密，用户模式对应空间不能进行 TBLP 操作 0 表示不加密，用户模式对应空间能进行 TBLP 操作（默认）

当写/擦除操作区域所对应 USR_W_SEC[X]值为 1’b0 时，则可以对该地址 PAGE 进行擦除或写操作；当写/擦除操作区域所对应 USR_W_SEC[X]值为 1’b1，则写/擦除操作失败，并且操作标志位 ISPOF 置 1，软件清零；

2.12.5. 校验码

地址为 ID 区 0xF800。CHECKSUM=CRC8(2K CODE)。

CRC8 算法采用 CRC8 ITU 标准:

多项式：　x^8+x^2+x+1

初值：　0x00

结果异或值：0x55

2.12.6. 模式切换及加密策略

每个目标芯片的代码选项区均会预留 4Bit 作为 ICD 调试模式的配置。该代码选项所在的地址根据芯片而定。在**芯片未加密**的情况下，该代码选项区可以通过 IDE 配置，以便于用户在程序发布之前，测试正常模式下的功能，包含：

1） 正常模式下的芯片功耗测试。如果使能 SWIM_CSR 的 OSCOFF 位，调试模式下振荡器不关闭，功耗与实际存在差异。用户可以通过 IDE 配置芯片为正常模式，测试功耗。

2） SWD 引脚复用功能的使用。调试模式下，SWD 引脚默认生效，则该引脚的复用功能无法进行调试。用户如果希望使用该引脚实现复用功能，则需要通过编译程序后，通过 IDE 配置芯片为正常模式，

并下载最新的程序，测试引脚复用功能。

位地址	标识符	功能
15:4	Reserve	保留
3:0	ICD_CFG[3:0]	ICD 模式使能选项 1111：调试模式，使能 ICD 功能，SWD 引脚生效 其他：正常模式，禁止 ICD 功能，SWD 引脚失效 其他：保留，暂时作为正常模式，禁止 ICD 功能

芯片加密后，代码选项区无法改写，因此 ICD_CFG 也无法改写。如果用户设置禁用 ICD 功能，同时打开加密位，则没有任何手段通过 ICD 访问芯片，只能通过烧录器全部擦除芯片。

用户不同开发场景下，芯片工作模式的定义与 ICD 功能说明如下：

模式	SWD 引脚	ICD 访问区域					
		ICD SFR	SFR	RAM	程序区	数据存储区	代码选项区
上电复位	使能	√	×	×	×	×	×
调试模式	使能	√	√	√	√	√	√
正常模式	禁用	×	×	×	×	×	×
加密调试	使能	√	√	√	×	×	×
加密正常	禁用	×	×	×	×	×	×

注：

1、 上电复位期间，SWD 引脚固定打开，主机可以通过从头擦除整个芯片，从而进入改写代码选项配置，进行调试模式和正常模式的切换。

2、 加密调试模式用于当用户对芯片进行加密后，希望在线定位问题。只有用于源码的用户，客户在改模式下继续进行程序调试，但不能设置软件断点，只有 1 个硬件断点可以使用。

各模式的切换流程对应如下：

模式	SWD	跳入条件	跳出条件
上电复位	使能	1） 从机系统重新上电 2） 主机发送 SRST 命令	上电复位完成，如果
调试模式	使能	1） ICD_CFG==1111 2） 上电复位完成	重新上电复位
正常模式	禁用	1） ICD_CFG==1010 2） 上电复位完成	重新上电复位
调试加密	使能	1、 ICD_CFG==1111 2、 加密位生效 3） 上电复位完成	重新上电复位
正常加密	禁用	1、 ICD_CFG==1010 2、 加密位生效 3） 上电复位完成	重新上电复位

用户相关应用场景操作如下：

场景	用户操作	CS-Link 操作
调试模式切换到正常模式，进行正常模式测试	1） 保持芯片连接 2） 配置芯片为正常模式 3） 点击下载程序并直接运行 4） 断开芯片，进行正常模式测试	1、 代码选项区 ICD_CFG 改写为 1010 2、 发送 SRST 命令，芯片重新读取代码选项
正常模式切换到调试模式，继续进行程序调试	1） 连接芯片，且 CS-Link 供电 2） 配置芯片为调试模式 3） 点击进入调试 4） 保持芯片连接，进行芯片程序调试	1） 通过 VDD 引脚复位芯片，从机进入上电复位模式 2） 从头开始擦除程序区，且配置 ICD_CFG=1111，重新写入程序 3） 发送 SRST 命令，重新开始运行
调试加密模式切换到调试模式	同上	同上
正常加密模式切换到正常模式，继续进行程序调试	无法切换，只能通过烧录器擦除重新烧录	无

3. MCU 指令集

表 20　MCU 指令集

指令	操作	指令周期	标志位
ADDLW k	[W]←[W]+k	1	C,DC,Z
ADDPCW	[PC] ←[PC]+1+[W]	2	~
ADDWF f,d	[Destination] ←[f]+[W]	1	C,DC,Z
ADDWFC f,d	[Destination] ←[f]+[W]+C	1	C,DC,Z
ANDLW k	[W]←[W] AND k	1	Z
ANDWF f,d	[Destination] ← [W] AND [f]	1	Z
BCF f,b	[f<b>]←0	1	~
BSF f,b	[f<b>]←1	1	~
BTFSC f,b	Jump if[f<b>]=0	1/2	~
BTFSS f,b	Jump if[f<b>]=1	1/2	~
CALL k	Push PC+1 and Goto K	2	~
CLRF f	[f]←0	1	Z
CLRWDT	Clear watch dog timer	1	~
COMF f,d	[f]←NOT([f])	1	Z
DAW	Decimal Adjust W	1	C,DC
DECF f,d	[Destination] ←[f] -1	1	Z
DECFSZ f,d	[Destination] ←[f] -1,jump if the result is zero	1/2	~
GOTO k	PC←k	2	~
HALT	CPU Stop	1	~
INCF f,d	[Destination] ←[f]+1	1	Z
INCFSZ f,d	[Destination] ←[f]+1,jump if the result is zero	1/2	~
IORLW k	[W]←[W] OR k	1	Z
IORWF f,d	[Destination] ← [W] OR [f]	1	Z
MOVFW f	[W]←[f]	1	~
MOVLW k	[W]←k	1	~
MOVP	Read table list	3	~
MOVWF f	[f]←[W]	1	~
NOP	No operation	1	~
POP	Pop W and Status	2	~
PUSH	Push W　and Status	2	~
RETFIE	Pop PC and GIE =1	2	~
RETLW k	RETURN and W=k	2	~
RETURN	POP PC	2	~
RLF f,d	[Destination<n+1>] ←[f<n>]	1	C,Z
RRF f,d	[Destination<n-1>] ←[f<n>]	1	C,Z
SLEEP	STOP OSC	1	PD
SUBLW k	[W] ← k – [W]	1	C,DC,Z
SUBWF f,d	[Destinnation] ← [f]– [W]	1	C,DC,Z

指令	操作	指令周期	标志位
SUBWFC f,d	[Destinnation] ← [f]– [W]-1+C	1	C,DC,Z
SWAPF f,d	swap f	1	~
TBLP k	Write memory	-	~
XORLW k	[W]←[W] XOR k	1	Z
XORWF f,d	[Destination] ← [W] XOR [f]	1	Z

参数说明：

f:　数据存储器地址(00H ~7FH)

W:　工作寄存器

k:　立即数

d:　目标地址选择:d=0 结果保存在工作寄存器,d=1:结果保存在数据存储器 f 单元

b:　位选择(0~7)

[f]:　f 地址的内容

PC:　程序计数器

C:　进位标志

DC:　半加进位标志

Z:　结果为零标志

PD:　睡眠标志位

TO:　看门狗溢出标志

WDT:　看门狗计数器

表 21　MCU 指令集描述

1

ADDLW	**加立即数到工作寄存器**
指令格式	ADDLW　　K　　(0<=K<=FFh) 6　　8
操作	(W)<—(W)+K
标志位	C，DC，Z
描述	工作寄存器的内容加上立即数 K 结果保存到工作寄存器中
周期	1
例子 ADDLW 08h	在指令执行之前： W=08h 在指令执行之后： W=10h

2

ADDPCW	**将 W 的内容加到 PC 中**
指令格式	ADDPCW 14
操作	(PC)<—(PC)+1+(W)　　当(W)<7Fh (PC)<—(PC)+1+(W)-100h　　其余
标志位	没有
描述	将地址 PC+1+W 加载到 PC 中
周期	2

ADDPCW	将 W 的内容加到 PC 中
例子 1 ADDPCW	在指令执行之前： W=7Eh ， PC=0212h 指令执行之后： PC=0291h
例子 2 ADDPCW	在指令执行之前： W=80h ， PC=0212h 指令执行之后： PC=0193h
例子 3 ADDPCW	在指令执行之前： W=FEh ， PC=0212h 指令执行之后： PC=0211h

3

ADDWF	加工作寄存器到 f
指令格式	ADDWF　f,d　　0<=f<=FFh　d=0,1 7　　7
操作	[目标地址]<—(f)+(W)
标志位	C，CD，Z
描述	将 f 的内容和工作寄存器的内容加到一起。 如果 d 是 0，结果保存到工作寄存器中。 如果 d 是 1，结果保存到 f 中。
周期	1
例子 1 ADDWF f 0	指令执行之前： f=C2h　W=17h 在指令执行之后 f=C2h　W=D9h
例子 2 ADDWF f 1	指令执行之前 f=C2h　W=17h 指令执行之后 f=D9h　W=17h

4

ADDWFC	将 W f 和进位位相加
指令格式	ADDWFC　f，d　0<=f<=FFh　d=0,1 7　　7
操作	(目标地址)<—(f)+(W)+C
标志位	C，DC，Z
描述	将工作寄存器的内容和 f 的内容以及进位位相加 当 d 为 0 时结果保存到工作寄存器 当 d 为 1 时结果保存到 f 中
周期	1
例子 ADDWFC f，1	指令执行之前 C=1　f=02h　W=4Dh

ADDWFC	将 W f 和进位位相加
	指令执行之后 C=0　f=50h　W=4Dh

5

ANDLW	工作寄存器与立即数相与
指令格式	ANDLW　K　　0<=K<=FFh 6　　8
操作	(W)<—(W) AND K
标志位	Z
描述	将工作寄存器的内容与 8bit 的立即数相与，结果保存到工作寄存器中。
周期	1
例子 ANDLW　5Fh	在指令执行之前 W=A3h 在指令执行之后 W=03h

6

ANDWF	将工作寄存器和 f 的内容相与
指令格式	ANDWF　f，d　0<=f<=FFh　d=0,1 7　　7
操作	(目标地址)<—(W) AND (f)
标志位	Z
描述	将工作寄存器的内容和 f 的内容相与 如果 d 为 0 结果保存到工作寄存器中 如果 d 为 1 结果保存到 f 中
周期	1
例子 1 ANDWF f，0	在指令执行之前 W=0Fh　f=88h 在指令执行之后 W=08h　f=88h
例子 2 ANDWF f，1	在指令执行之前 W=0Fh　f=88h 在指令执行之后 W=0Fh　f=08h

7

BCF	清除 f 的某一位
指令格式	BCF　f，b　　0<=f<=FFh　0<=b<=7 BCF b f 4　3　7
操作	(f[b])<—0
标志位	无
描述	F 的第 b 位置为 0
周期	1
例子	指令执行之前：

BCF	**清除 f 的某一位**
BCF FLAG 2	FLAG=8Dh 指令执行之后： FLAG=89h

8

BSF	**F 的 b 位置 1**
指令格式	BSF　f，b　　0<=f<=FFh　0<=b<=7 BSF b f 4　　3 7
操作	(f[b])<—1
标志位	无
描述	将 f 的 b 位置 1
周期	1
例子 BSF FLAG 2	在指令执行之前 FLAG=89h 在指令执行之后 FLAG=8Dh

9

BTFSC	**如果 f 寄存器的 bit[b]为 0，则跳转**
指令格式	BTFSC　f，b　　0<=f<=FFh　0<=b<=7 BTFSC b f 4　　3 7
操作	Skip if (f[b])=0
标志位	无
描述	如果 f 的 bit 位是 0，下一条取到的指令将被丢弃，然后执行一条空指令组成一个两周期的指令。
周期	无跳转则为 1 个指令周期，否则 2 个指令周期

例子
NODE BTFSC FLAG 2
OP1:
OP2:
	在程序执行以前
	PC=address(NODE)
	指令执行之后
	If(FLAG[2])=0
	　PC=address(OP2)
	If(FLAG[2])=1
	　PC=address(OP1)

10

BTFSS	**如果 f 寄存器的 bit[b]为 1，则跳转**
指令格式	BTFSS　f，b　0<=f<=FFh　0<=b<=7 BTFSS　b f 4　　3 7
操作	Skip if (f[b])=1
标志位	无
描述	如果 f 的 bit 位是 1，下一条取到的指令将被丢弃，然后执行一条空指令组成一个两周期的指令。

BTFSS	如果 f 寄存器的 bit[b]为 1，则跳转
周期	无跳转则为 1 个指令周期，否则 2 个指令周期

例子
NODE BTFSS FLAG 2
OP1:
OP2:
	在程序执行以前
	PC=address(NODE)
	指令执行之后
	If(FLAG[2])=0
	PC=address(OP1)
	If(FLAG[2])=1
	PC=address(OP2)

11

CALL	子程序调用
指令格式	CALL　K　0<=K<=7FFh 3　　11
操作	(top stack)<—PC+1 PC<—K
标志位	无
描述	子程序调用，先将 PC+1 压入堆栈，然后把立即数地址下载到 PC 中。
周期	2

12

CLRF	清除 f
指令格式	CLRF　f　0<=f<=FFh 7　　7
操作	(f)<—0
标志位	Z
描述	将 f 的内容清零
周期	1
例子 CLRF　WORK	在指令执行之前 WORK=5Ah 在指令执行之后 WORK=00h

*注。当 CLRF status 寄存器时，标志位 Z 不会置高

13

CLRWDT	清除看门狗定时器
指令格式	CLRWDT 14
操作	看门狗计数器清零
标志位	无
描述	清除看门狗定时器
周期	1
例子 CLRWDT	指令执行之后 WDT=0

14

COMF	**f 取反**
指令格式	COMF　f，d　0<=f<=FFh　d=0,1 7　　7
操作	(目的地址)<—NOT(f)
标志位	Z
描述	将 f 的内容取反， 当 d 为 0 时，结果保存到工作寄存器中， 当 d 为 1 时，结果保存到 f 中。
周期	1
例子 COMF f，0	在指令执行之前 W=88h，f=23h 在指令执行之后 W=DCh，f=23h
例子 2 COMF f，1	在指令执行之前 W=88h，f=23h 在指令执行之后 W=88h，f=DCh

15

DAW	**十进制调整 W 寄存器值**
指令格式	DAW 14
操作	十进制调整 W 寄存器值
标志位	C,DC
描述	一般与加法一起使用。 如果低半字节的值大于 9 或 DC 为 1 时，低半字节加 6； 如果高半字节的值大于 9 或 C 为 1 时，高半字节加 6
周期	1
例子 若 W=25h； ADDLW　39h DAW	在 DAW 指令执行之前 W=25+39 =64=（5EH+6） 在指令执行之后 W=64H

16

DECF	**f 减 1**
指令格式	DECF　f，d　0<=f<=FFh　d=0,1 7　　7
操作	(目的地址)<—(f)-1
标志位	Z
描述	F 的内容减 1 当 d 为 0 时，结果保存到工作寄存器中 当 d 为 1 时，结果保存到 f 中。
周期	1
例子	在指令执行之前

DECF	f 减 1
DECF　f，0	W=88h　f=23h 在指令执行之后 W=22h　f=23h
例子 2 DECF　f，1	在指令执行之前 W=88h　f=23h 在指令执行之后 W=88h　f=22h

17

DECFSZ	f 减 1 如果为 0 则跳转
指令格式	DECFSZ　f，d　0<=f<=FFh　d=0,1 7　　7
操作	(目的地址)<—(f)-1,如果结果为 0 跳转
标志位	无
描述	f 的内容减 1。 如果 d 为 0，结果保存到工作寄存器中。 如果 d 为 1，结果保存到 f 中 如果结果为 0，下一条已经取到的指令将被丢掉，然后插入一条 NOP 指令组成一个两个周期的指令。
周期	无跳转则为 1 个指令周期，否则 2 个指令周期

例子
Node DECFSZ FLAG，1
OP1：
OP2：
	在指令执行之前
	PC=address(Node)
	在指令执行之后
	(FLAFG)=(FLAG)-1
	If(FLAG)=0
	PC=address(OP2)
	If(FLAG)!=0
	PC=address(OP1)

18

GOTO	无条件跳转
指令格式	GOTO　K　0<=K<=7FFh 3　　13
操作	PC<—K
标志位	无
描述	立即地址载入 PC
周期	2

19

HALT	停止 CPU 时钟
指令格式	HALT 14
操作	CPU 停止
标志位	无
描述	CPU 时钟停止，晶振仍然工作，CPU 能够通过内部或者外部中断重启。

HALT	停止 CPU 时钟
周期	1

20

INCF	f 加 1
指令格式	INCF　f，d　0<=f<=FFh　d=0,1 7　　7
操作	(目的地址)<—(f)+1
标志位	Z
描述	f 加 1 如果 d 为 0，结果保存到工作寄存器中 如果 d 为 1，结果保存到 f 中。
周期	1
例子 INCF　f，0	在指令执行之前 W=88h　f=23h 在指令执行之后 W=24h　f=23h
例子 2 INCF　f，1	在指令执行之前 W=88h　f=23h 在指令执行之后 W=88h　f=24h

21

INCFSZ	f 加 1，如果结果为 0 跳转
指令格式	INCFSZ　f，d　0<=f<=FFh　d=0,1 7　　7
操作	(目的地址)<—(f)+1 如果结果为 0 就跳转
标志位	无
描述	f 的内容加 1。 如果 d 为 0，结果保存到工作寄存器中。 如果 d 为 1，结果保存到 f 中 如果结果为 0，下一条已经取到的指令将被丢掉，然后插入一条 NOP 指令组成一个两个周期的指令。
周期	无跳转则为 1 个指令周期，否则 2 个指令周期

例子
Node INCFSZ FLAG，1
OP1：
OP2：
	在指令执行之前
	PC=address(Node)
	在指令执行之后
	(FLAFG)=(FLAG)+1
	If(FLAG)=0
	PC=address(OP2)
	If(FLAG)!=0
	PC=address(OP1)

22

IORLW	工作寄存器与立即数或
指令格式	IORLW　K　0<=K<=FFh 7　　7
操作	(W)<—(W)|K
标志位	Z
描述	立即数与工作寄存器的内容或。结果保存到工作寄存器中。
周期	1
例子 IORLW　85H	在指令执行之前 W=69h 在指令执行之后 W=EDh

23

IORWF	f 与工作寄存器或
指令格式	IORWF　f，d　0<=f<=FFh　d=0,1 7　　7
操作	(目的地址)<—(W)|(f)
标志位	Z
描述	f 和工作寄存器或 当 d 为 0 时，结果保存到工作寄存器中 当 d 为 1 时，结果保存到 f 中
周期	1
例子 IORWF　f,1	在指令执行前 W=88h　f=23h 在指令执行后 W=88h　f=ABh

24

MOVFW	传送到工作寄存器
指令格式	MOVFW　f　0<=f<=FFh 7　　7
操作	(W)<—(f)
标志位	无
描述	将数据从 f 传送到工作寄存器
周期	1
例子 MOVFW　f	在指令执行之前 W=88h　f=23h 在指令执行之后 W=23h　f=23h

25

MOVLW	将立即数传送到工作寄存器中
指令格式	MOVLW　K　0<=K<=FFh 6　　8
操作	(W)<—K

MOVLW	将立即数传送到工作寄存器中
标志位	无
描述	将 8bit 的立即数传送到工作寄存器中
周期	1
例子 MOVLW 23H	在指令执行之前 W=88h 在指令执行之后 W=23h

26

MOVP	读查表区数据
指令格式	MOVP 14
操作	把 MTP 数据读到 EDATH/WORK 中
标志位	无
描述	把地址为 EADRH/EADRL 的查表区数据读到 EDATH/WORK 中
周期	2
例子 MOVP	在指令执行之前 EADRH=04h，EADRL=00h 地址为 0400h 的查表区数据位 1234h 在指令执行之后 EDATH=12h,W=34h

27

MOVWF	将工作寄存器的值传送到 f 中
指令格式	MOVWF　f　0<=f<=FFh 7　7
操作	(f)<—(W)
标志位	无
描述	将工作寄存器的值传送到 f 中
周期	1
例子 MOVWF　f	在指令执行之前 W=88h　f=23h 在指令执行之后 W=88h　f=88h

28

NOP	无操作
指令格式	NOP 14
操作	无操作
标志位	无
描述	无操作
周期	1

29

PUSH	把 work 和 status 寄存器入栈保护
指令格式	PUSH 14
操作	(top stack)<—work/status
标志位	无
描述	把 work 和 status 寄存器的值做入栈处理，支持 8 级堆栈，不同于 PC 堆栈；其中状态寄存器不包括 LVD36，LVD24，PD 和 TO。
周期	2

30

POP	把 work 和 status 寄存器出栈处理
指令格式	POP 14
操作	(Top Stack)=>work/status Pop Stack
标志位	无
描述	把当前栈顶的值做出栈处理，分别更新 work 和 status 寄存器，支持 8 级堆栈，不同于 PC 堆栈；其中状态寄存器不包括 LVD36，LVD24，PD 和 TO。
周期	2

31

RETFIE	从中断返回
指令格式	RETFIE 14
操作	(Top Stack)=>PC Pop Stack 1=>GIE
标志位	无
描述	PC 从堆栈顶部得到，然后出栈，设置全局中断使能位为 1
周期	2

32

RETLW	返回，并将立即数送到工作寄存器中
指令格式	RETLW　K　0<=K<=FFh 6　8
操作	(W)<—K (Top Stack)=>PC Pop Stack
标志位	无
描述	将 8bit 的立即数送到工作寄存器中，PC 值从栈顶得到，然后出栈
周期	2

33

RETURN	从子程序返回
指令格式	RETURN 14

RETURN	从子程序返回
操作	(Top Stack)=>PC Pop Stack
标志位	无
描述	PC 值从栈顶得到，然后出栈
周期	2

34

RLF	带进位左移
指令格式	RLF　f，d　0<=f<=FFh　d=0,1 7　　7
操作	(目标地址[n+1])<—(f[n]) (目标地址[0])<—C C<—(f[7])
标志位	C，Z
描述	F 带进位位左移一位 如果 d 为 0，结果保存到工作寄存器 如果 d 为 1，结果保存到 f 中
周期	1
例子 RLF f，1	在指令执行之前 C=0　W=88h　f=E6h 在指令执行之后 C=1　W=88h　f=CCh

35

RRF	带进位右移
指令格式	RRF　f，d　0<=f<=FFh　d=0,1 7　　7
操作	(目标地址[n-1])<—(f[n]) (目标地址[7])<—C C<—(f[0])
标志位	C
描述	F 带进位位右移一位 如果 d 为 0，结果保存到工作寄存器 如果 d 为 1，结果保存到 f 中
周期	1
例子 RRF f，0	在指令执行之前 C=0　W=88h　f=95h 在指令执行之后 C=1　W=4Ah　f=95h

36

SLEEP	晶振停止
指令格式	SLEEP 14
操作	CPU 晶振停止

SLEEP	晶振停止
标志位	PD
描述	CPU 晶振停止。CPU 通过外部中断源重启
周期	1

37

SUBLW	立即数减工作寄存器的值
指令格式	SUBLW　K　　0<=K<=FFh 6　　8
操作	(W)<—K-(W)
标志位	C，DC，Z
描述	8bit 的立即数减去工作寄存器的值，结果保存到工作寄存器中
周期	1
例子 SUBLW 02H	在指令执行之前 W=01h 在指令执行之后 W=01h　C=1(代表没有借位)　Z=0(代表结果非零)
例子 2 SUBLW 02H	在指令执行之前 W=02h 在指令执行之后 W=00h　C=1(代表没有借位)　Z=1(代表结果为零)
例子 2 SUBLW 02H	在指令执行之前 W=03h 在指令执行之后 W=FFh　C=0(代表有借位)　Z=0(代表结果非零)

38

SUBWF	f 的值减工作寄存器的值
指令格式	SUBWF　f，d　0<=f<=FFh　d=0,1 7　　7
操作	(目标地址)<—(f)-(W)
标志位	C，DC，Z
描述	f 的值减去工作寄存器的值。 如果 d 为 0，结果保存到工作寄存器 如果 d 为 1，结果保存到 f 中
周期	1
例子 SUBWF　f，1	在指令执行之前 f=33h　W=01h 在指令执行之后 f=32h　C=1　Z=0
例子 2 SUBWF　f，1	在指令执行之前 f=01h　W=01h 在指令执行之后 f=00h　C=1　Z=1
例子 3	在指令执行之前

SUBWF	f 的值减工作寄存器的值
SUBWF　f，1	f=04h　W=05h 在指令执行之后 f=FFh　C=0　Z=0

39

SUBWFC	带借位的减法
指令格式	SUBWFC　f，d　　0<=f<=FFh　d=0,1 7　　7
操作	(目标地址)<—(f)-(W)-1+C
标志位	C，DC，Z
描述	f 的值减去工作寄存器的值 如果 d 为 0，结果保存到工作寄存器 如果 d 为 1，结果保存到 f 中
周期	1
例子 SUBWFC f，1	在指令执行之前 W=01h　f=33h　C=1 在指令执行之后 f=32h　　C=1　　Z=0
例子 2 SUBWFC f，1	在指令执行之前 W=01h　f=02h　C=0 在指令执行之后 f=00h　　C=1　　Z=1
例子 3 SUBWFC f，1	在指令执行之前 W=05h　f=04h　C=0 在指令执行之后 f=FEh　　C=0　　Z=0

40

SWAPF	交换寄存器的值
指令格式	SWAPF f，d　0<=f<=FFh d=0,1 7　　7
操作	(des[3:0])<—f[7:4] (des[7:4])<—f[3:0]
标志位	无
描述	把 f 寄存器的高 4 位数据给目标寄存器的低 4 位； 把 f 寄存器的低位数据给目标寄存器的高 4 位 d 为 1 时，f 寄存器为目标寄存器；否则，w 寄存器为目标寄存器
周期	1
例子 SWAPF　f,1	在指令执行之前 f=ACh 在指令执行之后 f=CAh

41

TBLP	写 Memory
指令格式	TBLP k　（k 取 0） 8　　8
操作	写 Memory
标志位	无
描述	把 EDATH/WORK 的数据写到存储器地址 EADRH/EADRL
周期	约为 700us 时间
例子 TBLP 0	在指令执行之前 EDATH=bah，W=Ach，EADRH=04h，EADRL=00h 在指令执行之后 把 baach 写到存储器地址 0400h

42

XORLW	工作寄存器的值与立即数异或
指令格式	XORLW　K　　0<=K<=FFh 6　　8
操作	(W)<—(W)^K
标志位	Z
描述	8bit 的立即数与工作寄存器的值异或，结果保存在工作寄存器中
周期	1
例子 XORLW　5Fh	在指令执行之前 W=Ach 在指令执行之后 W=F3h

43

XORWF	f 的值与工作寄存器的值异或
指令格式	XORWF　f，d　　0<=f<=FFh　d=0,1 7　　7
操作	(目标地址)<—(W)^(f)
标志位	Z
描述	F 的值与工作寄存器的值异或， 当 d 为 0 时，结果保存到工作寄存器中 当 d 为 1 时，结果保存到 f 中
周期	1
例子 XORWF　f，1	在指令执行之前 W=ACh　f=5Fh 在指令执行之后 f=F3h

4. 免责声明和版权公告

免责声明和版权公告

本文档中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

本文档可能引用了第三方的信息，所有引用的信息均为“按现状”提供，芯海科技不对信息的准确性、真实性做任何保证。

芯海科技不对本文档的内容做任何保证，包括内容的适销性、是否适用于特定用途，也不提供任何其他芯海科技提案、规格书或样品在他处提到的任何保证。

芯海科技不对本文档是否侵犯第三方权利做任何保证，也不对使用本文档内信息导致的任何侵犯知识产权的行为负责。本文档在此未以禁止反言或其他方式授予任何知识产权许可，不管是明示许可还是暗示许可。

Wi-Fi 联盟成员标志归 Wi-Fi 联盟所有。蓝牙标志是 Bluetooth SIG 的注册商标。

文档中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

芯海科技
CHIPSEA
股票代码:688595