



芯海科技

CHIPSEA

股票代码:688595

# CSU18M68 用户手册

带 16 位 ADC 的桥式传感器调理芯片

V1.2 版本



芯海科技(深圳)股份有限公司

www.chipsea.com

+86-0755-8616 9257

sales@chipsea.com

518000

## 版本历史

历史版本	修改内容	版本日期
REV 1.0	初始版本	2021-05-18
REV 1.1	更新封面文档格式	2021-11-20
REV 1.2	刷新应用原理图	2022-1-6

## 目 录

版本历史.....	2
1 产品概述.....	4
1.1 功能描述.....	4
1.2 主要特性.....	4
1.3 PIN 配置.....	2
2 应用原理图.....	4
3 标准功能.....	5
3.1 CPU 核.....	5
3.2 SFR.....	4
3.3 时钟系统.....	5
3.4 复位系统.....	9
3.5 中断.....	11
3.6 定时器 0/1.....	14
3.7 I/O PORT.....	18
4 增强功能.....	21
4.1 HALT 和 SLEEP 模式.....	21
4.2 看门狗(WDT).....	21
4.3 电源系统.....	24
4.4 I2C 从机.....	28
4.5 16BIT-ADC 模块.....	37
4.6 数据查表.....	40
4.7 代码选项.....	42
5 MCU 指令集.....	44
6 电气特性.....	61
6.1 极限值.....	61
6.2 直流特性.....	61
6.3 16BIT ADC 的特性.....	61
6.4 MTP 的特性.....	62
7 封装信息.....	64
7.1 WLCSP16 封装尺寸图.....	64
7.2 QFN16 封装尺寸图.....	64
8 产品印字.....	66
9 包装信息.....	68
10 专利保护.....	69

## 1 产品概述

### 1.1 功能描述

CSU18M68 是一个带 16bit ADC 的 SOC，内置 8k×16 位 MTP 程序存储器 和 488 字节数据存储器。

### 1.2 主要特性

#### 高性能的 RISC CPU

- 8 位 MCU
- 内置 8k×16Bits 的 MTP 程序存储器(烧录次数 1000 次)
- 488 字节数据存储器 (SRAM)：需要翻页
- 43 条指令、8 级存储堆栈
- 指令周期为：ICK/2, ICK/4

#### 振荡器

- 内置 8MHz 振荡器，Trim 之后精度为±1%
- 内带 3kHz WDT 振荡器，TRIM 精度为±10%。

#### 外设特性

- 4 个 双向 I/O 口
- 4 个内部中断：16bit-ADC、TIMER0、TIMER1、I2C、
- 2 个外部中断：  
INT0：PT1.0 或 PT1.4  
INT1：PT1.1 或 PT1.5
- 4 个具有唤醒功能的输入口  
分别为 PT1.0/PT1.1 /PT1.4/PT1.5
- 2 路输入全差分 16bit Sigma-Delta 型 ADC，  
1/5/10/20 倍 4 档增益选择，2KHz~32KHz 多种  
DataRate 选择。
- 内置温度传感器
- 1 路 I2C 从机，支持标准模式（100k bit/s）和  
快速模式（400k bit/s），支持 1.8V-3.6V

#### 专用微控制器的特性

- 上电复位 (POR)
- 上电复位延迟定时器 (39ms)
- 内带低电压复位 (LVR)
- 定时器 0/定时器 1  
-可编程预分频的 8 位的定时器
- 看门狗定时器 (3K WDT)

#### CMOS 技术

- 电压工作范围  
— 2.75V~3.6V;

- 工作温度范围  
— -40°C~85°C;

#### 低功耗特性

- MCU 工作电流  
— 正常模式 0.7mA@2MHz, 3.3V  
— 休眠模式 <0.8μA, 关 WDT  
— 休眠模式 <2μA, 开 WDT

#### 封装

- WLCSP16: 1.45mm×1.465mm, e=0.35
- QFN16: 3mm×3mm, e=0.5

### 1.3 PIN 配置

#### WLCSP16

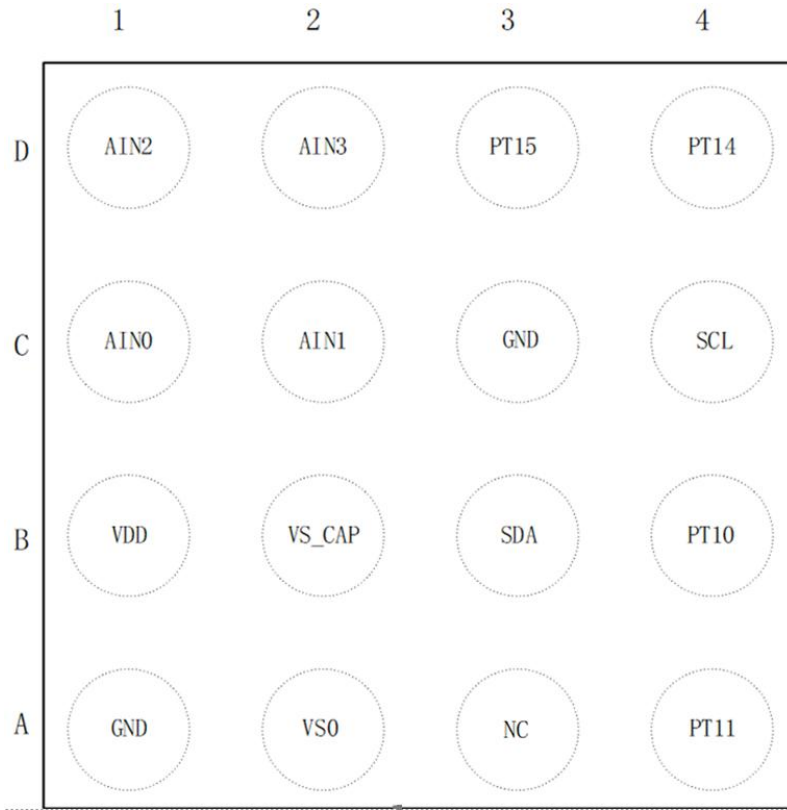


图 1-1 CSU18M68 WLCSP16 封装引脚图 Top View (marking side)

表 1-1 管脚说明

管脚名称	输入/输出	管脚序号	描述
GND	P	A1	地
VS0	AO	A2	参考电压输出
NC	AI	A3	保留管脚, Not Care
PT11	I/O	A4	GPIO 输入, 推挽输出; 时钟烧录管脚
VDD	P	B1	电源
VS_CAP	AI/AO	B2	内置 LDO 电压输出
SDA	I/O	B3	I2C 数据口
PT10	I/O	B4	GPIO 输入, 推挽输出; 数据烧录管脚
AIN0	AI	C1	0 通道正端模拟输入
AIN1	AI	C2	0 通道负端模拟输入
GND	I	C3	地
SCL	I/O	C4	I2C 时钟口
AIN2	AI	D1	1 通道正端模拟输入
AIN3	AI	D2	1 通道负端模拟输入
PT15	I/O	D3	GPIO 输入, 开漏输出
PT14	I/O	D4	GPIO 输入, 开漏输出

QFN16

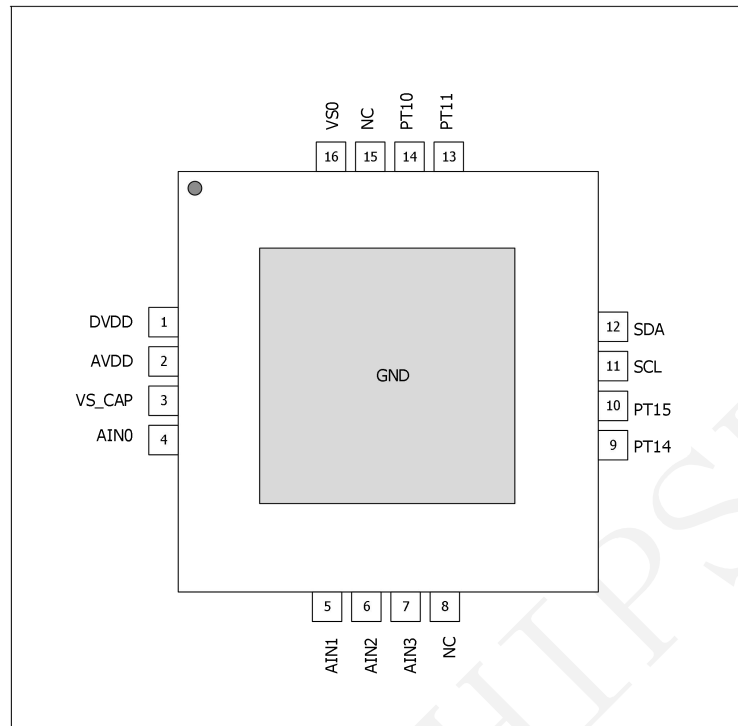
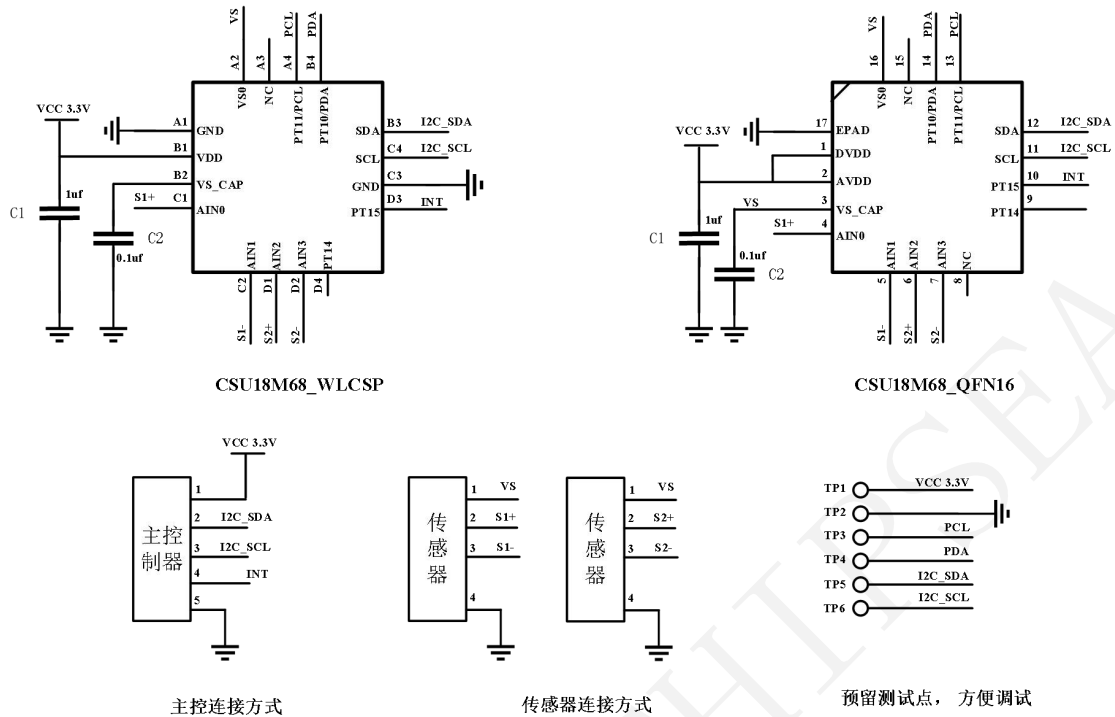


图 1-2 CSU18M68 QFN16 封装引脚图

表 1-2 管脚说明

管脚名称	输入 / 输出	管脚序号	描述
DVDD	P	1	供电电源
AVDD	P	2	供电电源
VS_CAP	AI/AO	3	内置 LDO 电压输出
AIN0	AI	4	0 通道正端模拟输入
AIN1	AI	5	0 通道负端模拟输入
AIN2	AI	6	1 通道正端模拟输入
AIN3	AI	7	1 通道负端模拟输入
NC	-	8	NC
PT14	I/O	9	GPIO 输入，开漏输出
PT15	I/O	10	GPIO 输入，开漏输出
SCL	I/O	11	IIC 时钟管脚
SDA	I/O	12	IIC 数据管脚
PT11	I/O	13	GPIO 输入，推挽输出；时钟烧录管脚
PT10	I/O	14	GPIO 输入，推挽输出；数据烧录管脚
NC	-	15	NC
VS0	AO	16	参考电压输出，外接传感器
GND	P	0	地

## 2 应用原理图



### 注释:

- 1: VS\_CAP : 输出引脚, 给sensor 供电
- 2: AIN0/AIN1与AIN2/AIN3 : Sensor差分信号输入引脚, 如果只用一个传感器, AIN2/AIN3可以用于检测传感器是否接触良好
- 3: SDA/SCL : I2C 通讯接口, 需要外接上拉电阻, 可用于固定升级
- 4: PT15 : 状态输出脚, 开漏输出, 需要外接上拉电阻
- 5: 芯片烧录引脚: VDD,GND,PDA,PCL

图 2-1 应用原理图

- 1) S1+-、S2+-之间根据应用场景需求可选择接电容100pF, 默认情况下不接电容。
  - 2) PT14/PT15 口接上拉电阻 4.7K; I2C 总线接上拉电阻 2K。
  - 3) 如需与CSU18M65兼容, VDD/GND之间电容4.7uF, VS\_CAP/GND之间电容0.47uF, 外接传感器的参考电压VS连接CSU18M68的VS\_CAP。
- 如不需兼容, CSU18M68的VDD/GND之间电容1uF, VS\_CAP/GND之间电容0.1uF。外接传感器的参考电压VS连接CSU18M68的VS0。

### 3 标准功能

#### 3.1 CPU 核

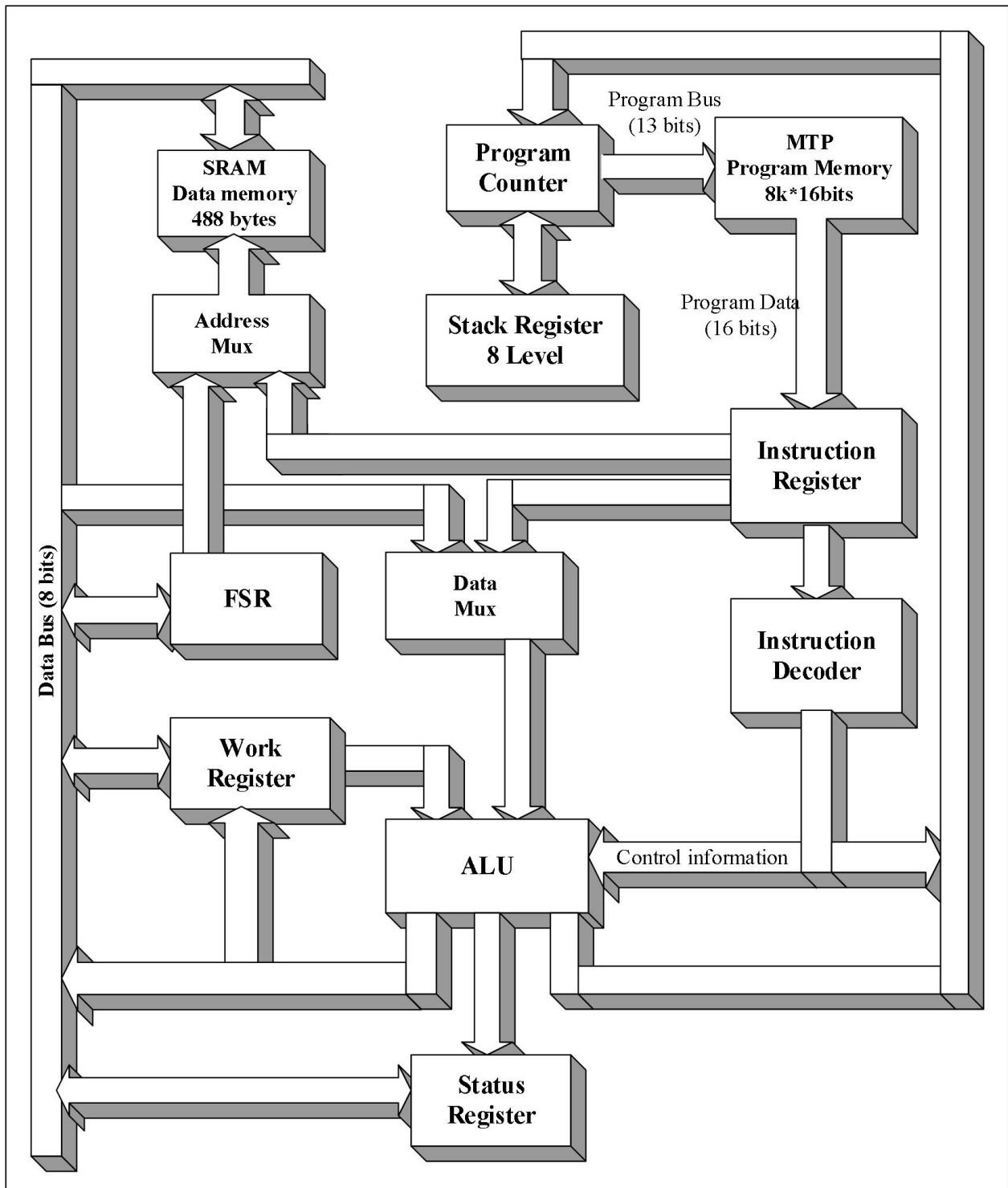


图 3-1 CSU18M68 CPU 核的功能模块图

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 3-1 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。
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	<p>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）及指令推送到指令寄存器。</p> <p>CSU18M68 的指令是 16bits，包括 3 种信息：直接地址，立即数及控制信息。</p> <p>CPU 能将立即数推送到工作寄存器，或者进行某些处理后，根据控制信息，将立即数存储到直接地址所指向的数据存储器寄存器中。</p> <p><b>直接地址（9bits）</b> 数据寄存器的地址。CPU 能利用此地址来对数据存储器进行操作。</p> <p><b>直接数据（8bits）</b> CPU 通过 ALU 利用此数据对工作寄存器进行操作。</p> <p><b>控制信息</b> 它记录着 ALU 的操作信息。</p>
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	工作寄存器是用来缓存数据存储器中某些存储地址的数据。
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 CSU18M68 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 IND 寄存器对这个寄存器进行处理。
程序存储器	CSU18M68 内带 8k×16 位的 MTP 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 8192*16 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	CSU18M68 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 9bits，数据总线是 8bits。

### 3.1.1 存储器

#### （1）程序存储器

程序存储器主要用于指令的存储，在 CSU18M68 中，该程序存储器是 8192\*16bit 的 MTP(地址范围是 0000H-1FFFH)。系统的 reset 地址实际为 0x0000，中断入口地址实际为 0x0004。需要注意的一点就是所有的中断共用同一个中断入口地址。

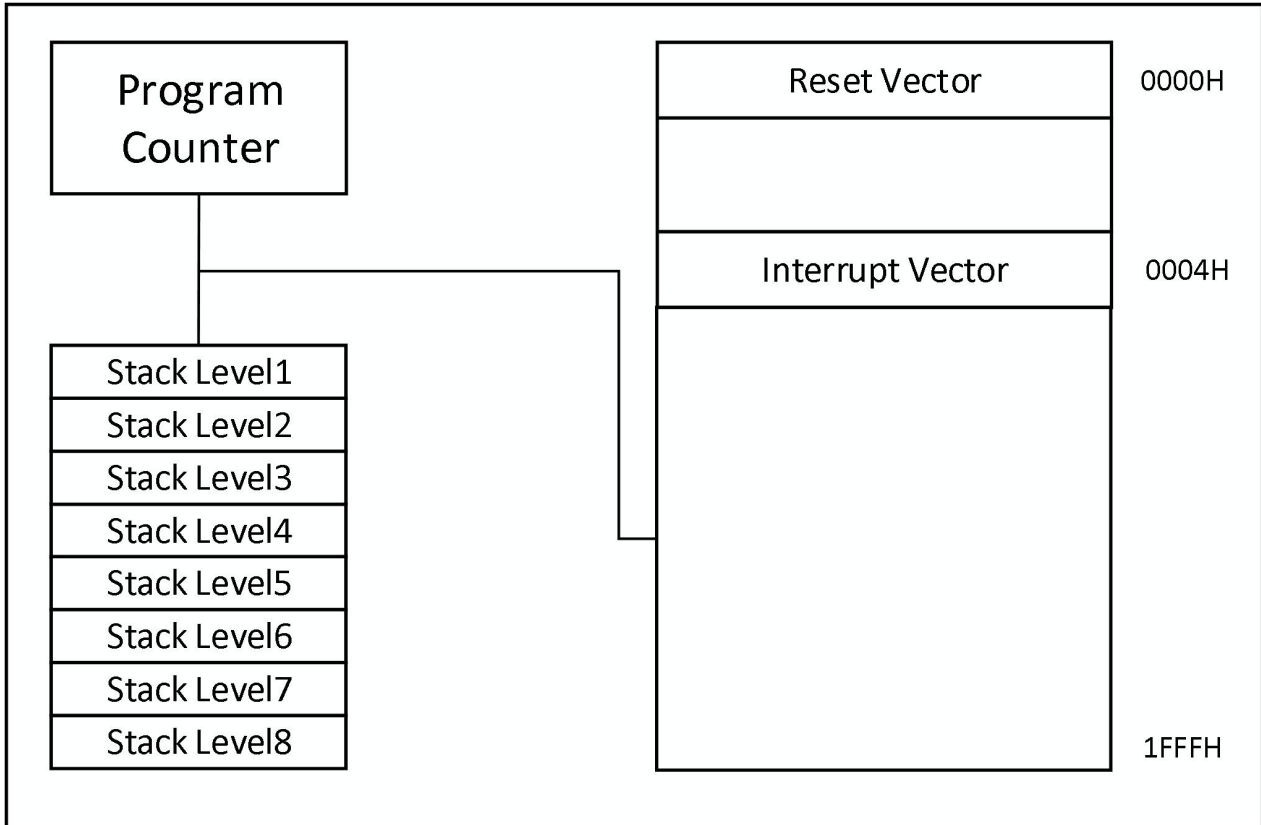


图 3-2 程序存储器

(2) 数据存储器的

数据存储器主要用于程序运行过程中，全局以及中间变量的存储。该存储器分为三个部分。地址的 0x000 至 0x008 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。地址的 0x009 至 0x07F 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。

注意：数据存储器的空间是 488Bytes。

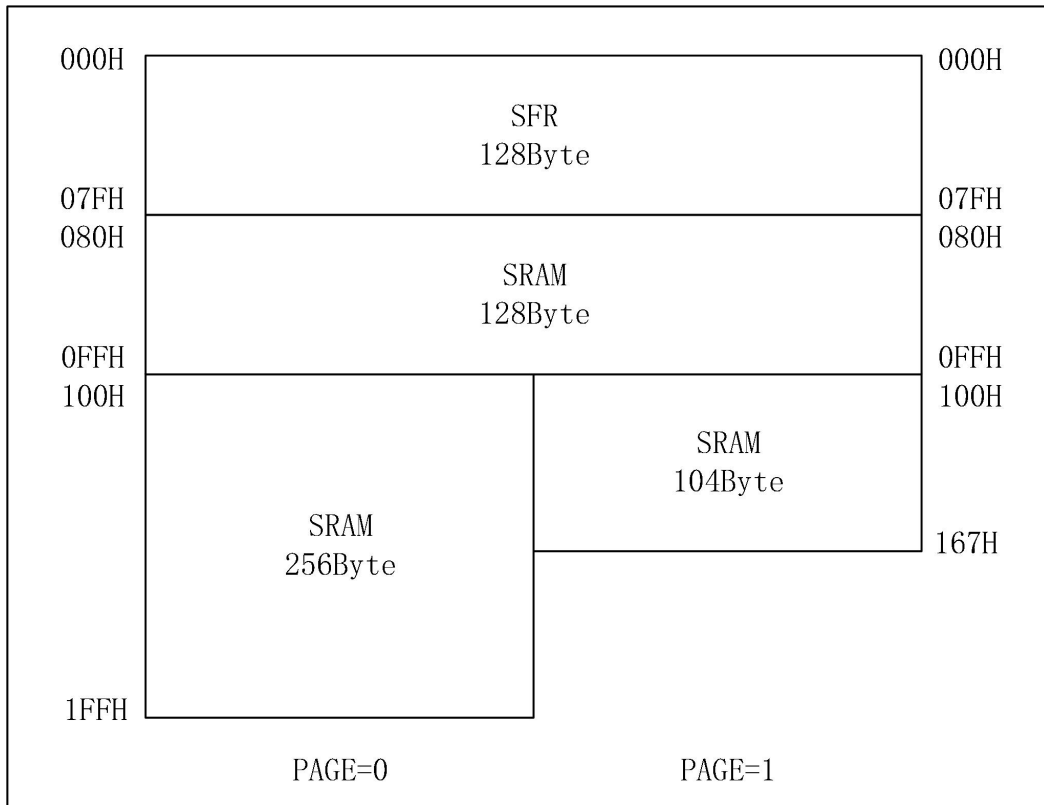


图 3-3 数据存储器

表 3-2 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	000H	008H
外设特殊功能寄存器	009H	07FH
通用数据存储器 (PAGE=1'b0)	080H	1FFH
通用数据存储器 (PAGE=1'b1)	080H	167H

注意，80H-FFH 地址段不分 PAGE，任意 PAGE 均可访问。

Bank 选择寄存器 (地址为 08H)

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	R/W-0	R/W-0
	IRP0	IRP1	-	-	-	-	PAGE1	PAGE0

位地址	标识符	功能
7	IRP0	1 = 间接寻址 IND0 时, 访问后 256byte 地址 0 = 间接寻址 IND0 时, 访问前 256byte 地址
6	IRP1	1 = 间接寻址 IND1 时, 访问后 256byte 地址 0 = 间接寻址 IND1 时, 访问前 256byte 地址
5: 2	RESERVE	保留位
1	PAGE1	1 = 间接寻址 IND1 时, 访问后 512byte 地址 0 = 间接寻址 IND1 时, 访问前 512byte 地址
0	PAGE0	1 = 直接和间接寻址 IND0 时, 访问后 512byte 地址 0 = 直接和间接寻址 IND0 时, 访问前 512byte 地址

通过 IND0 及 PAGE0、FSR0 或 IND1 及 PAGE1、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。其访问方式见下图间接地址访问。

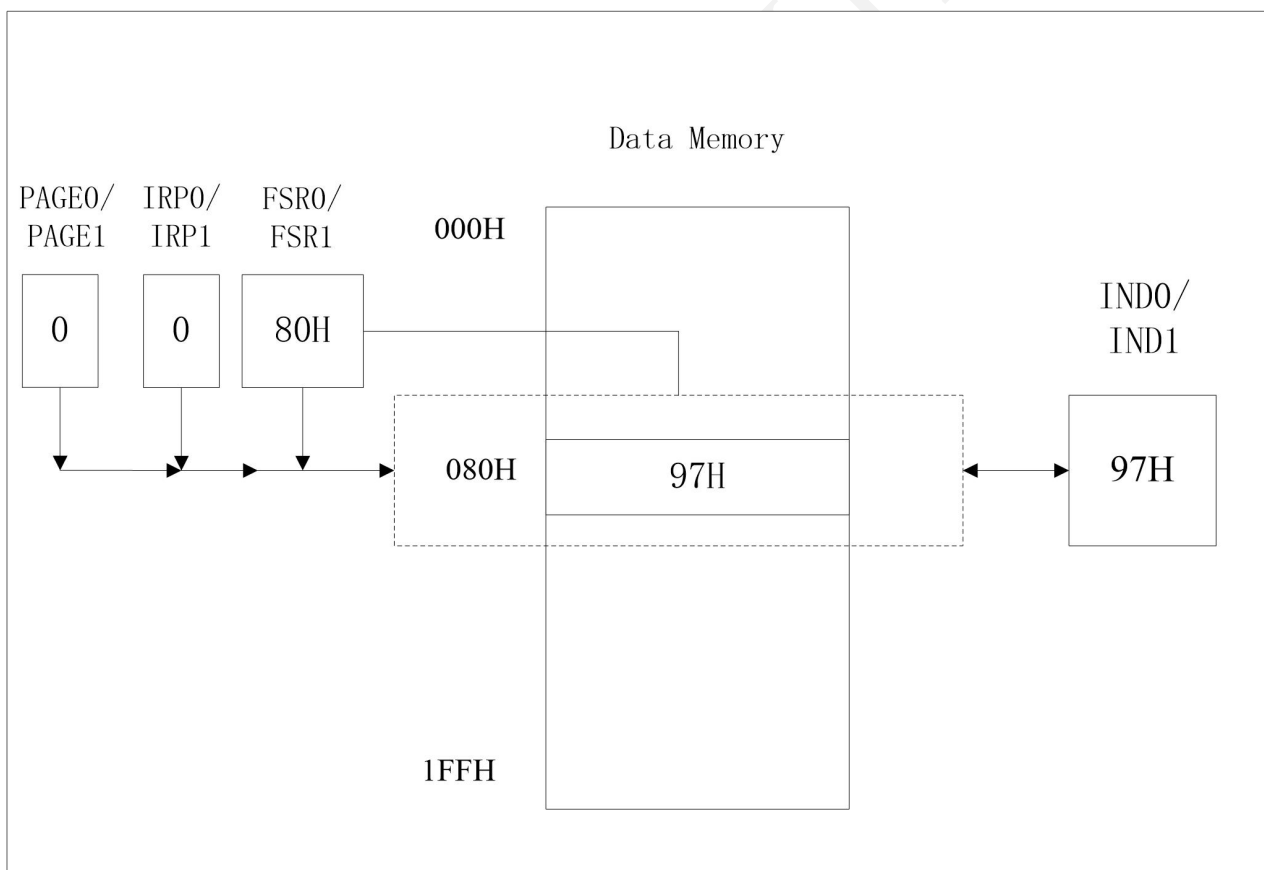


图 3-4 间接地址访问

### 3.1.2 状态寄存器

状态寄存器包含 ALU 的算术状态及复位状态。状态寄存器类似于其它寄存器, 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

写是不使能。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。

### 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	保留位，需保持复位值
4	PD	掉电标志位， <code>sleep</code> 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，看门狗定时溢出置位此位 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

注：DC、C、Z 寄存器位写入的数据可能会被运算结果冲刷掉。

#### 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

### 3.1.3 中断寄存器

中断系统的入口地址为 0x0004，各个中断之间没有优先级，靠程序控制各个中断的优先级。只要有中断标志位，就会有中断响应，响应中断之后需要软件将中断标志位清除，否则会不断响应中断。

INTE 及 INTF 寄存器是可读、可写的，包括使能位及标志位，用于中断器件。

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06H	INTF	-	-	TM1IF	TM0IF	-	ADIF	E1IF	E0IF	uu00u000
07H	INTE	GIE	-	TM1IE	TM0IE	-	ADIE	E1IE	E0IE	0u00u000
5AH	INTF3	I2C_TIF	I2C_RIF	-	-	-	I2C_STIF	-	-	00uuu0uu
5BH	INTE3	I2C_TIE	I2C_RIE	-	-	-	I2C_STIE	-	-	00uuu0uu

#### INTE 寄存器

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
INTE	GIE	-	TM1IE	TM0IE	-	ADIE	E1IE	E0IE

位地址	标识符	功能
7	GIE	全局中断使能标志 1: 使能所有非屏蔽中断 0: 不使能所有中断
6	RESERVE	保留位
5	TM1IE	8-Bit 定时器 1 中断使能标志 1: 使能定时器 1 中断 0: 不使能定时器 1 中断
4	TM0IE	8-Bit 定时器 0 中断使能标志 1: 使能定时器 0 中断 0: 不使能定时器 0 中断
3	RESERVE	保留位，需保持默认值
2	ADIE	16-bit AD 中断使能标志 1: 使能 AD 中断 0: 不使能 AD 中断
1	E1IE	外部中断 1 使能标志 1: 使能外部中断 1 0: 不使能外部中断 1
0	E0IE	外部中断 0 使能标志 1: 使能外部中断 0 0: 不使能外部中断 0

#### INTF 寄存器

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-----	------	------	------	------	------	------	------	------

特性	U-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
INTF	-	-	TM1IF	TM0IF	-	ADIF	E1IF	E0IF

位地址	标识符	功能
7: 6	RESERVE	保留位
5	TM1IF	8-Bit 定时器 1 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
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	RESERVE	保留位
2	ADIF	Sigma Delta AD 中断标志, 软件清零, 硬件置高 1 = 发生 Sigma Delta AD 中断, 必须软件清 0 0 = 没发生 Sigma Delta AD 中断
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 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

### INTE3 寄存器

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	R/W-0	R/W-0	R/W-0
INTE3	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 : 使能 I2C 从机发送中断 0 : 关闭 I2C 从机发送中断
6	I2C_RIE	I2C 从机接收中断使能 1 : 使能 I2C 从机接收中断 0 : 关闭 I2C 从机接收中断
5:3	RESERVE	保留位
2	I2C_STIE	I2C 从机开始中断使能 1: 使能 I2C 从机开始中断 0: 关闭 I2C 从机开始中断
1:0	RESERVE	保留位

## INTF3 寄存器

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	R/W-0	R/W-0	R/W-0
INTF3	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 1 : 发生 I2C 从机发送中断, 必须软件清 0 0 : 未发生 I2C 从机发送中断					
6	I2C_RIF		I2C 从机接收中断标志 1 : 发生 I2C 从机接收中断, 必须软件清 0 0 : 未发生 I2C 从机接收中断					
5:3	RESERVE		保留位					
2	I2C_STIF		I2C 从机开始中断标志 1: 发生 I2C 从机开始中断, 必须软件清 0 0: 未发生 I2C 从机开始中断					
1:0	RESERVE		保留位					

## 3.2 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。

表 3-3 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								xxxxxxx
01H	IND1	以 FSR1 中内容作为地址的数据存储器中的数据								xxxxxxx
02H	FSR0	间接数据存储器的地址指针 0								00000000
03H	FSR1	间接数据存储器的地址指针 1								00000000
04H	STATUS	-	-	-	PD	TO	DC	C	Z	uuu00xxx
05H	WORK	工作寄存器								00000000
06H	INTF	-		TM1IF	TM0IF		ADIF	E1IF	E0IF	uu00u000
07H	INTE	GIE		TM1IE	TM0IE		ADIE	E1IE	E0IE	0u00u000
08H	BSR	IRP0	IRP1	-	-	-	-	PAGE1	PAGE0	00uuuu00
09H	MCK		M3_CK				WDT_CLK_EN			u0uuu0uu
0AH	EADRH	PROG_BUSY	READ_CHECK	PARH[5:0]						00000000
0BH	EADRL	PARL[7:0]								00000000
0CH	EDAT	EDAT[7:0]								11111111
0DH	EOPEN	EOPEN[7:0]								00000000
0EH	WDTCON	WDTEN	ROOT_EN		I2C_DIV[1:0]		WDTS[2:0]			00u00000



0FH	WDTIN	WDTIN[7:0]								00000000
11H	ADOH	ADO[15:8]								00000000
13H	ROOT	ROOT								00000000
14H	RSTSR					LVDF	EMCF	ILOPF	uuuu00uu	
16H	ADCFG	ADSC			S_GAIN[1:0]				0uu00u00	
17H	ANACFG	LDOEN	LDOS[1:0]		BGR_ENB	BGID	SINL[1:0]		ADEN	00000000
1BH	LVDCON	LVDEN		AIENB1	SILB[2:0]			LBOU	LB_RST_CON	0u1000x0
1CH	ADOL	ADO[7:0]								00000000
1DH	PT1			PT1[5:0]					uuuuuuuu	
1EH	PT1EN			PT1EN[5:0]					uu000000	
1FH	PT1PU							PT1UP[1:0]		uuuuuu00
2CH	INPUT	PT15_VDD	PT14_VDD	I2C_VDD	PT11_VDD	PT10_VDD	I2C_FLT	PT11_OD	PT10_OD	11111100
2DH	PTCON				E1M[1:0]			E0M[1:0]		uuuu0000
2EH	PTINT0						PTW0[2:0]			uuuuuu000
2FH	PTINT1						PTW1[2:0]			uuuuuu000
34H	TM0CON	T0EN	T0RATE[2:0]				T0RSTB		T0SEL	0000u1u0
35H	TM0IN	TM0IN[7:0]								00000000
36H	TM0CNT	TM0CNT[7:0]								00000000
37H	TM1CON	T1EN	T1RATE[2:0]				T1RSTB		T1SEL	0000u1u0
38H	TM1IN	TM1IN[7:0]								00000000
39H	TM1CNT	TM1CNT[7:0]								00000000
5AH	INTF3	I2C_TIF	I2C_RIF				I2C_STIF			00uuu0uu
5BH	INTE3	I2C_TIE	I2C_RIE				I2C_STIE			00uuu0uu
68H	I2CCON	I2C_EN	AWK_EN	CTS_EN	ACK_EN	I2CSTUS[3:0]			00000000	
69H	I2CDAT	I2CDAT[7:0]								00000000
6AH	I2CISR								TXE	uuuuuuu0
6BH	ADCON	VS0_OEN	VS_LIMIT	AINOUT[1:0]				ADM[1:0]		0000uu00
6CH	TEST	TEST[7:0]								00000000
6EH	ICK_TRIM	ICK_TRIM[7:0]								10000000
7EH	METCH	METCH[3:0]				EVPP_EN				00000uuu
7FH	I2CADR	I2CADR[6:0]							GC_EN	01001100

注：进行读操作时，无效位读数为 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

### 3.3 时钟系统

#### 3.3.1 内部振荡器

CSU18M68 内置 8MHZ 时钟，上电默认内置 8M 时钟使能打开，默认一直工作；在 sleep 模式下，内置 8M 时钟自动停止工作，减小功耗。

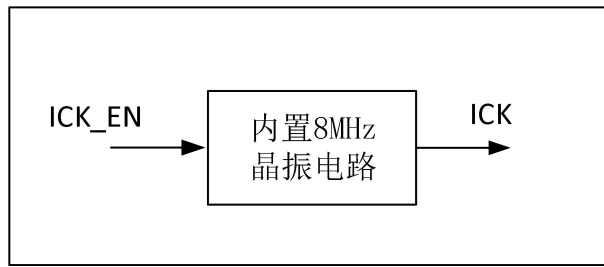


图 3-5 CSU18M68 ICK 振荡器状态框图

CSU18M68 内置 3K WDT 时钟，在烧录模式 WDT\_CLK\_EN 和 WDTEN 均可打开与关闭(默认关闭)。在工作模式下，WDT\_CLK\_EN 和 WDTEN 只能打开（默认关闭），不能关闭，只能通过上电、掉电复位恢复到默认值。

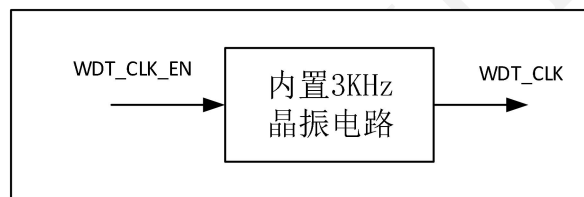


图 3-6 CSU18M68 WDT\_CLK 振荡器状态框图

### 3.3.2 CPU 指令周期

表 3-4 CSU18M68 CPU 指令周期寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bit0	复位值
09H	MCK		M3_CK				WDT_CLK_EN			u0uuu0uu

用户可以通过 MCK 寄存器中的 M3\_CK 寄存器位来选择指令周期（CPUCLK）

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W-0	U-0	U-0	U-0	R/W-0	U-0	U-0
MCK		M3_CK				WDT_CLK_EN		

位地址	标识符	功能
7	RESERVE	保留位
6	M3_CK	指令周期选择配置 1: 指令周期为 MCK/2 (4MHz) 0: 指令周期为 MCK/4 (2MHz)
5: 3	RESERVE	保留位
2	WDT_CLK_EN	内部 WDT 时钟启动开关

		1: WDT_CLK 启动 0: WDT_CLK 关闭 (默认)
1: 0	RESERVE	保留位

### 3.3.3 TMxCLK

TMxCLK 用于定时器 0/1 模块。Timer 的时钟源来自于 CPUCLK 或 WDT，时钟源通过 T0SEL 进行选择。时钟源经过分频之后产生 TM0CLK 用作定时器 0 模块的时钟。

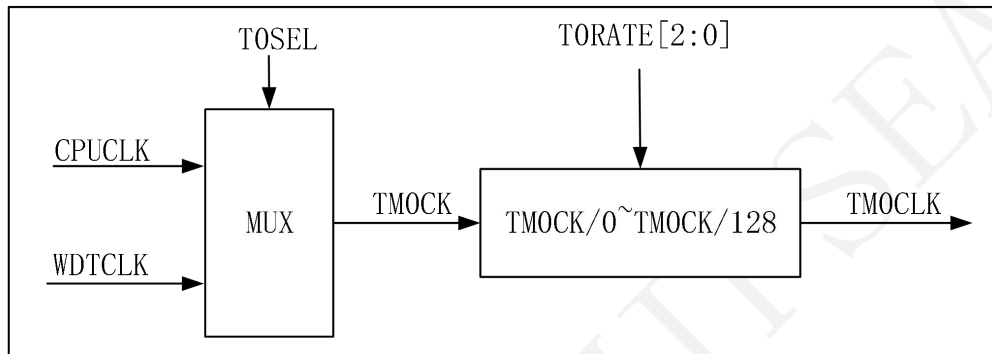


图 3-7 定时器 0/1 时钟分频示意图

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	U-0	R/W-0
TM0CON	T0EN	TORATE[2:0]		-	-	T0RSTB	-	T0SEL

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	TORATE[2:0]	定时器 0 时钟选择 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>TORATE [2:0]</td> <td>TM0CLK</td> </tr> <tr> <td>000</td> <td><b>TM0CK</b></td> </tr> <tr> <td>001</td> <td><b>TM0CK /2</b></td> </tr> <tr> <td>010</td> <td><b>TM0CK /4</b></td> </tr> <tr> <td>011</td> <td><b>TM0CK /8</b></td> </tr> <tr> <td>100</td> <td><b>TM0CK /16</b></td> </tr> <tr> <td>101</td> <td><b>TM0CK /32</b></td> </tr> <tr> <td>110</td> <td><b>TM0CK /64</b></td> </tr> <tr> <td>111</td> <td><b>TM0CK /128</b></td> </tr> </table>	TORATE [2:0]	TM0CLK	000	<b>TM0CK</b>	001	<b>TM0CK /2</b>	010	<b>TM0CK /4</b>	011	<b>TM0CK /8</b>	100	<b>TM0CK /16</b>	101	<b>TM0CK /32</b>	110	<b>TM0CK /64</b>	111	<b>TM0CK /128</b>
TORATE [2:0]	TM0CLK																			
000	<b>TM0CK</b>																			
001	<b>TM0CK /2</b>																			
010	<b>TM0CK /4</b>																			
011	<b>TM0CK /8</b>																			
100	<b>TM0CK /16</b>																			
101	<b>TM0CK /32</b>																			
110	<b>TM0CK /64</b>																			
111	<b>TM0CK /128</b>																			
3	RESERVE	保留位																		
2	T0RSTB	定时器 0 复位 1: 禁止定时器 0 复位																		

		0: 使能定时器 0 复位 当将该位为 0 时, 定时器 0 复位后, T0RSTB 会自动置 1	
1	RESERVE	保留位	
0	T0SEL	时钟源选择	
		T0SEL	定时器 0 时钟源(TM0CK)
		0	CPUCLK
1	内部 3K WDT 时钟, 仅当内部 WDT 晶振打开时有效		

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	U-0	R/W-0
TM1CON	T1EN	T1RATE[2:0]			-	T1RSTB	-	T1SEL

位地址	标识符	功能	
7	T1EN	定时器 1 使能位 1: 使能定时器 1 0: 禁止定时器 1	
6: 4	T1RATE[2:0]	定时器 0 时钟选择	
		T1RATE [2:0]	TM1CLK
		000	<b>TM1CK</b>
		001	<b>TM1CK /2</b>
		010	<b>TM1CK /4</b>
		011	<b>TM1CK /8</b>
		100	<b>TM1CK /16</b>
		101	<b>TM1CK /32</b>
		110	<b>TM1CK /64</b>
111	<b>TM1CK /128</b>		
3	RESERVE	保留位	
2	T1RSTB	定时器 0 复位 1: 禁止定时器 0 复位 0: 使能定时器 0 复位 当将该位为 0 时, 定时器 0 复位后, T0RSTB 会自动置 1	
1	RESERVE	保留位	
0	T1SEL	时钟源选择	
		T0SEL	定时器 0 时钟源(TM0CK)
		0	CPUCLK
1	内部 3K WDT 时钟, 仅当内部 WDT 晶振打开时有效		

### 3.3.4 16bit-ADCCLK

CSU18M68 包含一个 16 位的 sigma delta 型的模数转换器 (ADC), ADC 采样频率 ICK/4 或 ICK/8。

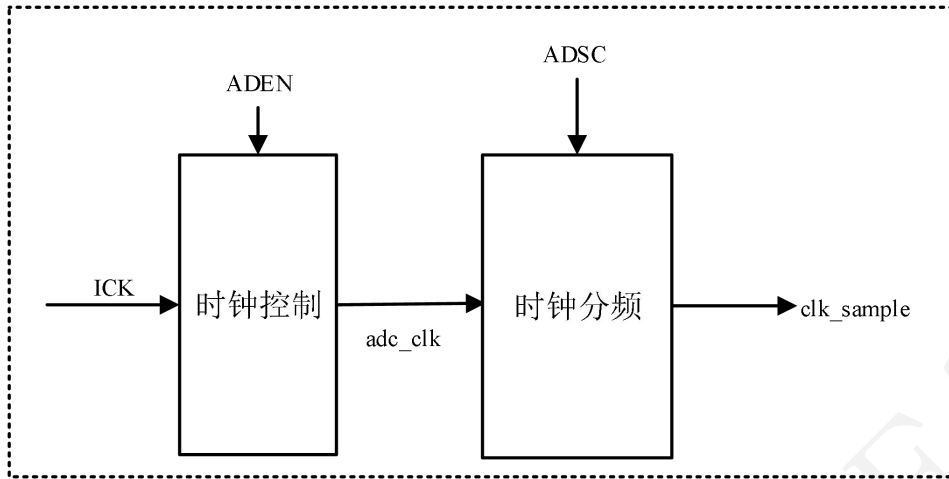


图 3-8 ADC 时钟选择

### 3.4 复位系统

ADSC	clk_sample
0	ICK/8
1	ICK/4

CSU18M68 包括以下几种复位方式：

- 上电复位
- 掉电复位
- 低电压检测复位
- 看门狗复位
- 非法指令复位
- EMC 复位

以上复位发生时，除复位标志外，其他所有的系统寄存器恢复默认状态，程序停止运行，同时程序计数器 PC 清零。复位结束后，系统从向量 0000H 处重新开始运行。

注：低电压复位是对低电压比较器产生的信号 LBOUT 进行判断产生。

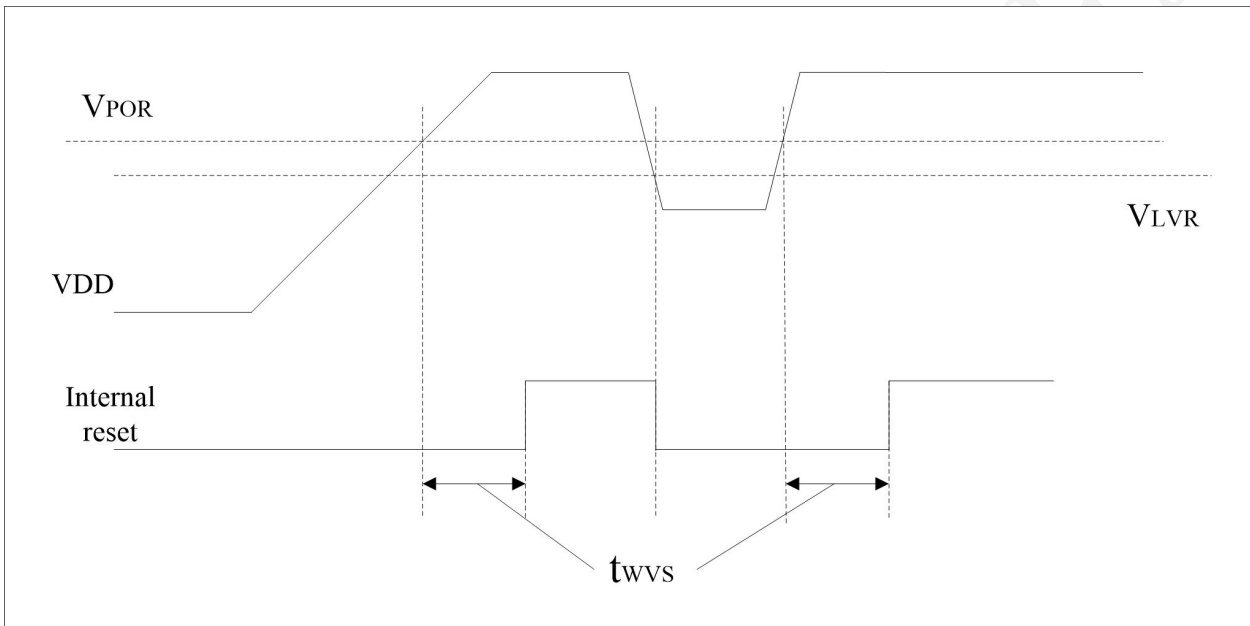


图 3-9 上电复位电路示例及上电过程

参数	最小值	典型值	最大值
VPOR		2.2V	
VLVR		2.0V	
tWVS		39ms	

VPOR：上电复位

VLVR：低电压复位

tWVS：等待电压稳定时间

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	U-0
RSTSR					LVDF	EMCF	ILOPF	

位地址	标识符	功能
7: 4	RESERVE	保留位
3	LVDF	低压复位标志

		1: 发生低电压复位 0: 未发生低电压复位
2	EMCF	EMC 复位标志 1: 发生 EMC 复位 0: 未发生 EMC 复位
1	ILOPF	非法指令复位标志 1: 发生非法指令复位 0: 未发生非法指令复位
0	RESERVE	保留位

复位类型	LVDF	TO	EMCF	ILOPF	重读代码选项	39ms 延时	复位域
上电/掉电复位	0	0	0	0	是	是	数字/模拟
低压复位	1	保持	保持	保持	是	否	数字
WDT 复位	保持	1	保持	保持	是	否	数字
非法指令复位	保持	保持	1	保持	是	否	数字
EMC 复位	保持	保持	保持	1	是	否	数字

## 3.5 中断

### 3.5.1 中断描述

CSU18M68 有 6 个中断源，但只有 1 个中断入口地址 0x0004。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。这 6 个中断源都各自有一个中断使能，和一个总使能位 GIE，并且它们的标志位硬件置位，软件清 0。

当响应中断时，会把当前的 PC 值入栈保护，并把 PC 置为 0x0004，同时把总使能位 GIE 清 0。执行完中断服务程序，并用 RETFIE 返回到之前的主程序，并把 GIE 置 1。

INT0、INT1、TIMER0、TIMER1、I2C 中断可唤醒 sleep 睡眠模式和 halt 停止模式。SDAD 中断只能唤醒 halt 模式。

CSU18M68 有 6 个中断源:

1) 外部中断 0

PT1.0 或 PT1.2 或 PT1.4 为外部中断 0 的输入端。触发方式由 PTCON 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，软件清 0。可唤醒 sleep 或 halt 模式。

2) 外部中断 1

PT1.1 或 PT1.3 或 PT1.5 作为外部中断 1 的输入端。触发方式由 PTCON 寄存器中的 E1M[1:0]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，软件清 0。

3) SDAD 中断

4) 定时器 0 溢出中断

5) 定时器 1 溢出中断

6) I2C 中断

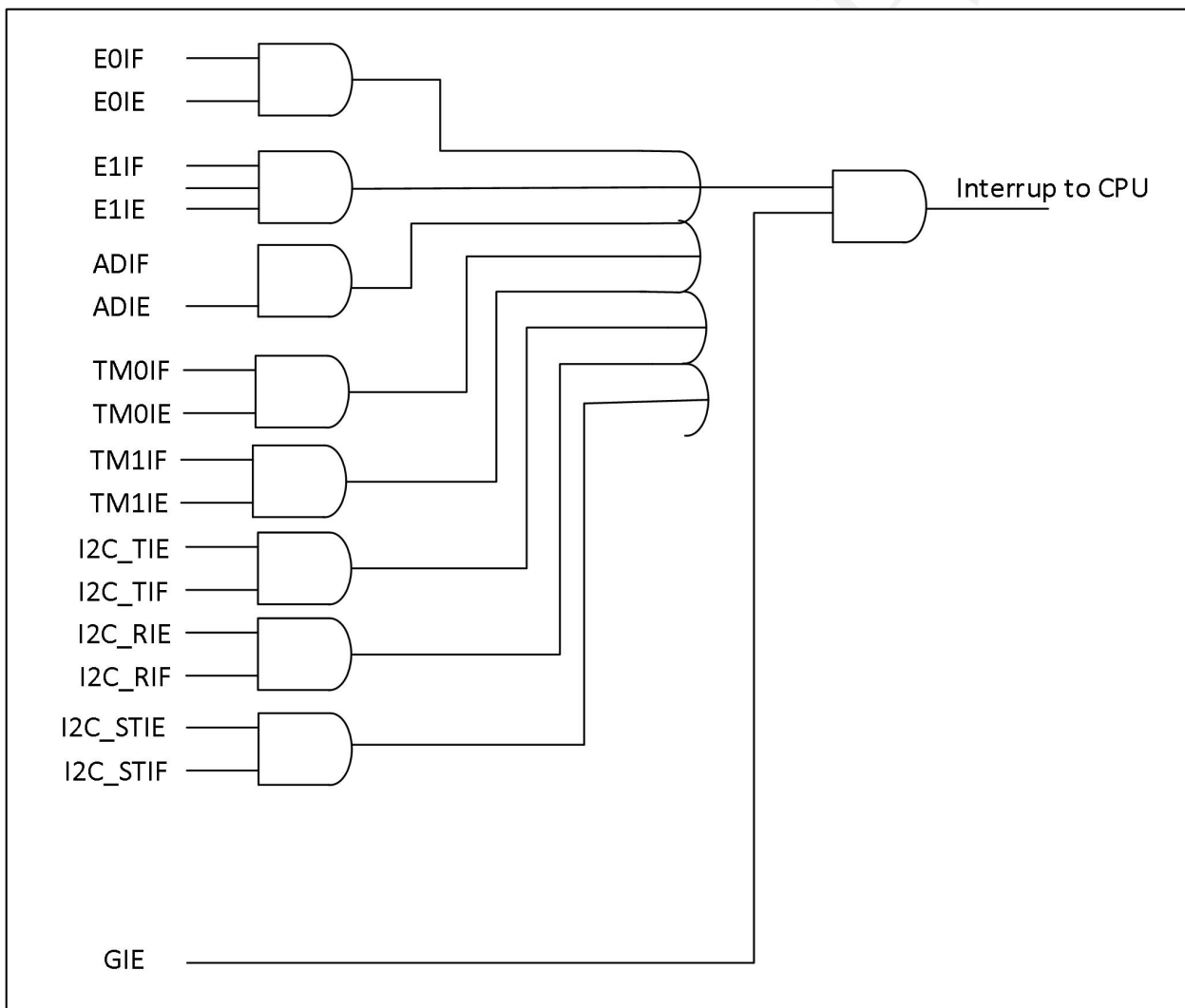


图 3-10 中断逻辑



### 3.5.2 中断寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06H	INTF			TM1IF	TM0IF		ADIF	E1IF	E0IF	uu00u000
07H	INTE	GIE		TM1IE	TM0IE		ADIE	E1IE	E0IE	0u00u000
5AH	INTF3	I2C_TIF	I2C_RIF				I2C_STIF			00uuu0uu
5BH	INTE3	I2C_TIE	I2C_RIE				I2C_STIE			00uuu0uu
2DH	PTCON					E1M[1:0]		E0M[1:0]		uuuu0000
2EH	PTINT0						PTW0[2:0]			uuuuu000
2FH	PTINT1						PTW1[2:0]			uuuuu000

#### PTCON 寄存器

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
PTCON					E1M[1:0]		E0M[1:0]	

位地址	标识符	功能
7: 4	RESERVE	保留位
3: 2	E1M	外部中断 1 触发模式 11 = 外部中断 1 在状态改变时触发 10 = 外部中断 1 在状态改变时触发 01 = 外部中断 1 为上升沿触发 00 = 外部中断 1 为下降沿触发
1: 0	E0M	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

#### PTINT0 寄存器

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	R/W-0
PTINT0						PTW0[2:0]		

位地址	标识符	功能
7: 3	RESERVE	保留位
2: 0	PTW0	PTW0[2]: PT1.4 外部中断 0 使能, 默认为 0 0 = 禁止 PT1.4 外部中断 0 1 = 使能 PT1.4 外部中断 0 PTW0[1]: PT1.2 外部中断 0 使能, 默认为 0

		0 = 禁止 PT1.2 外部中断 0 1 = 使能 PT1.2 外部中断 0 PTW0[0]: PT1.0 外部中断 0 使能, 默认为 0 0 = 禁止 PT1.0 外部中断 0 1 = 使能 PT1.0 外部中断 0 注意: PTW0 只能 1 个 bit 为 1, 及只能选择 1 个 IO 作为外部中断 0, 如果存在多个 bit 位 1, 那么外部中断 0 选择高位为 1 对应的 PAD 位置
--	--	--

### PTINT1 寄存器

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	R/W-0
PTINT1						PTW1[2:0]		

位地址	标识符	功能
7: 3	RESERVE	保留位
2: 0	PTW1	PTW1[2]: PT1.5 外部中断 1 使能, 默认为 0 0 = 禁止 PT1.5 外部中断 1 1 = 使能 PT1.5 外部中断 1 PTW1[1]: PT1.3 外部中断 1 使能, 默认为 0 0 = 禁止 PT1.3 外部中断 1 1 = 使能 PT1.3 外部中断 1 PTW1[0]: PT1.1 外部中断 1 使能, 默认为 0 0 = 禁止 PT1.1 外部中断 1 1 = 使能 PT1.1 外部中断 1 注意: PTW1 只能 1 个 bit 为 1, 及只能选择 1 个 IO 作为外部中断 1, 如果存在多个 bit 位 1, 那么外部中断 1 选择高位为 1 对应的 PAD 位置

## 3.6 定时器 0/1

### 3.6.1 定时器 0/1 框图

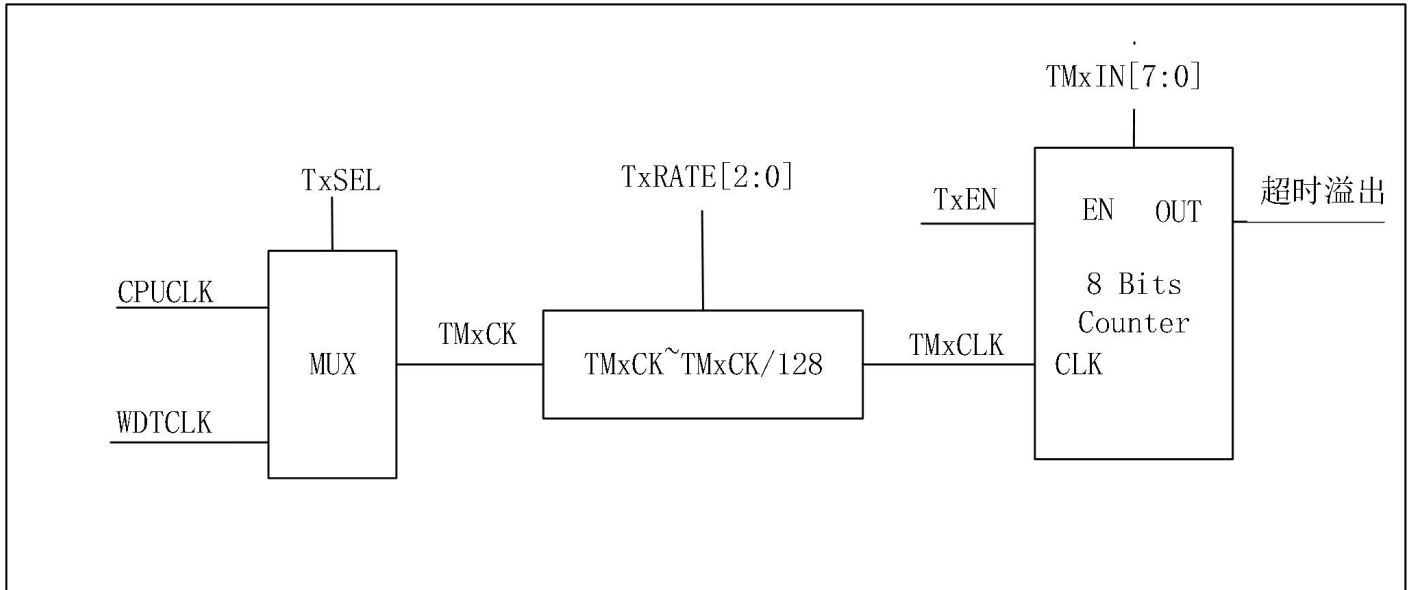


图 3-11 定时器 0/1 功能框图

定时器 0/1 的输入为 CPUCLK、WDTCLK。在定时器 0/1 集成了一个分频器，分频的时钟 TMxCLK 作为 8 bits 计数器的输入时钟。当用户设置了定时器 0/1 模块的使能标志，8 bits 计数器将启动，将会从 00H 递增到 TMxIN。用户需要设置 TMxIN（定时器 0/1 模块中断信号选择器）以选择定时超时中断信号。当定时超时发生时，中断标志位会自设置，程序计数器会跳转到 0004H 以执行中断服务程序。

### 3.6.2 定时器 0/1 寄存器列表

表 3-5 定时器 0/1 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06H	INTF	-		TM11F	TM01F		ADIF	E11F	E01F	uu00u000
07H	INTE	GIE		TM11E	TM01E		ADIE	E11E	E01E	0u00u000
34H	TM0CON	T0EN	T0RATE[2:0]				T0RSTB		T0SEL	0000u1u0
35H	TM0IN	TM0IN[7:0]								00000000
36H	TM0CNT	TM0CNT[7:0]								00000000
37H	TM1CON	T1EN	T1RATE[2:0]				T1RSTB		T1SEL	0000u1u0
38H	TM1IN	TM1IN[7:0]								00000000
39H	TM1CNT	TM1CNT[7:0]								00000000

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	U-0	R/W-0
TM0CON	T0EN	T0RATE[2:0]				T0RSTB		T0SEL

表 3-6 TM0CON 寄存器各位功能表

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	TORATE[2:0]	定时器 0 时钟选择 <table border="1"> <thead> <tr> <th>TORATE [2:0]</th> <th>TM0CLK</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>TM0CK</td> </tr> <tr> <td>001</td> <td>TM0CK /2</td> </tr> <tr> <td>010</td> <td>TM0CK /4</td> </tr> <tr> <td>011</td> <td>TM0CK /8</td> </tr> <tr> <td>100</td> <td>TM0CK /16</td> </tr> <tr> <td>101</td> <td>TM0CK /32</td> </tr> <tr> <td>110</td> <td>TM0CK /64</td> </tr> <tr> <td>111</td> <td>TM0CK /128</td> </tr> </tbody> </table>	TORATE [2:0]	TM0CLK	000	TM0CK	001	TM0CK /2	010	TM0CK /4	011	TM0CK /8	100	TM0CK /16	101	TM0CK /32	110	TM0CK /64	111	TM0CK /128
TORATE [2:0]	TM0CLK																			
000	TM0CK																			
001	TM0CK /2																			
010	TM0CK /4																			
011	TM0CK /8																			
100	TM0CK /16																			
101	TM0CK /32																			
110	TM0CK /64																			
111	TM0CK /128																			
3	NC	保留位																		
2	TORSTB	定时器 0 复位 1: 禁止定时器 0 复位 0: 使能定时器 0 复位 当将该位为 0 时, 定时器 0 复位后, TORSTB 会自动置 1																		
1	RESERVE	保留位																		
0	T0SEL	时钟源选择 <table border="1"> <thead> <tr> <th>T0SEL[1:0]</th> <th>定时器 0 时钟源(TM0CK)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>CPUCLK</td> </tr> <tr> <td>1</td> <td>内部 3K WDT 时钟, 仅当内部 WDT 时钟打开时有效</td> </tr> </tbody> </table>	T0SEL[1:0]	定时器 0 时钟源(TM0CK)	0	CPUCLK	1	内部 3K WDT 时钟, 仅当内部 WDT 时钟打开时有效												
T0SEL[1:0]	定时器 0 时钟源(TM0CK)																			
0	CPUCLK																			
1	内部 3K WDT 时钟, 仅当内部 WDT 时钟打开时有效																			

表 3-7 TM0IN 寄存器各位功能表

位地址	标识符	功能
7 : 0	TM0IN[7:0]	定时器 0 溢出值

表 3-8 TM0CNT 寄存器各位功能表

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

表 3-9 TM1CON 寄存器各位功能表

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	U-0	R/W-0
TM1CON	T1EN	T1RATE[2:0]				T1RSTB		T1SEL

表 3-10 TM1CON 寄存器各位功能表

位地址	标识符	功能
7	T1EN	定时器 1 使能位 1: 使能定时器 1

		0: 禁止定时器 1																		
6:4	T1RATE[2:0]	定时器 1 时钟选择 <table border="1"> <tr> <td>T1RATE [2:0]</td> <td>TM1CLK</td> </tr> <tr> <td>000</td> <td>TM1CK</td> </tr> <tr> <td>001</td> <td>TM1CK /2</td> </tr> <tr> <td>010</td> <td>TM1CK /4</td> </tr> <tr> <td>011</td> <td>TM1CK /8</td> </tr> <tr> <td>100</td> <td>TM1CK /16</td> </tr> <tr> <td>101</td> <td>TM1CK /32</td> </tr> <tr> <td>110</td> <td>TM1CK /64</td> </tr> <tr> <td>111</td> <td>TM1CK /128</td> </tr> </table>	T1RATE [2:0]	TM1CLK	000	TM1CK	001	TM1CK /2	010	TM1CK /4	011	TM1CK /8	100	TM1CK /16	101	TM1CK /32	110	TM1CK /64	111	TM1CK /128
T1RATE [2:0]	TM1CLK																			
000	TM1CK																			
001	TM1CK /2																			
010	TM1CK /4																			
011	TM1CK /8																			
100	TM1CK /16																			
101	TM1CK /32																			
110	TM1CK /64																			
111	TM1CK /128																			
3	NC	保留位																		
2	T1RSTB	定时器 1 复位 1: 禁止定时器 1 复位 0: 使能定时器 1 复位 当将该位为 1 时, 定时器 1 复位后, T1RSTB 会自动置 1																		
1:0	T1SEL	时钟源选择 <table border="1"> <tr> <td>T1SEL[1:0]</td> <td>定时器 1 时钟源(TM1CK)</td> </tr> <tr> <td>0</td> <td>CPUCLK</td> </tr> <tr> <td>1</td> <td>内部 3K WDT 时钟, 仅当内部 WDT 时钟打开时有效</td> </tr> </table>	T1SEL[1:0]	定时器 1 时钟源(TM1CK)	0	CPUCLK	1	内部 3K WDT 时钟, 仅当内部 WDT 时钟打开时有效												
T1SEL[1:0]	定时器 1 时钟源(TM1CK)																			
0	CPUCLK																			
1	内部 3K WDT 时钟, 仅当内部 WDT 时钟打开时有效																			

表 3-11 TM0IN 寄存器各位功能表

位地址	标识符	功能
7 : 0	TM0IN[7:0]	定时器 0 溢出值

表 3-12 TM0CNT 寄存器各位功能表

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

操作:

设置 TMxCLK, 为定时器 0/1 选择输入。

设置 TMxIN, 选择定时器 0/1 溢出值。

设置寄存器标志位: TMxIE 与 GIE, 使能定时器 0/1 中断。

设置寄存器标志位: TMxEN, 使能定时器 0/1 的 8 bits 计数器。

清零寄存器标志位: TxRSTB, 复位定时器 0/1 的计数器。 (复位的持续时间为 1.5\*TMxCK, 注意不是 TMxCLK)

当定时超时发生时, 寄存器标志位 TMxIF 会自复位, 程序计数器会复位为 0004H。

定时器 0/1 溢出时间计算方法:

定时器 0/1 溢出时间= (TMxIN+1) /TMxCLK

### 3.7 I/O PORT

CSU18M68 包括 6 个双向 IO 口，所有的 IO 口均可配置 IO 通信电压。其中 PT1.0、PT1.1 口可以配置上拉。

其他 IO 特性：

IO 其他功能复用见相关功能模块描述

表 3-13 I/O 口寄存器表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
1DH	PT1			PT1[5:0]						uuxxxxxx
1EH	PT1EN			PT1EN[5:0]						uu000000
1FH	PT1PU							PT1UP[1:0]		uuuuuu00
2CH	INPUT	PT15_VDD	PT14_VDD	I2C_VDD	PT11_VDD	PT10_VDD	I2C_FLT	PT11_OD	PT10_OD	11111100
2DH	PTCON					E1M[1:0]		E0M[1:0]		000uuuuu
2EH	PTINT0						PTW0[2:0]			uuuuu000
2FH	PTINT1						PTW1[2:0]			uuuuu000

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。CSU18M68 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。

#### 3.7.1 PT1 寄存器

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	R/W-X	R/W-X	R/W-X
PT1			PT1[5:0]					

位地址	标识符	功能
7: 6	RESERVE	保留位
5	PT1[5]	PT1.5 数据标志位 作为输入时，通过读该位获取输入数据； 作为输出时，通过写该位输出数据
4	PT1[4]	PT1.4 数据标志位 作为输入时，通过读该位获取输入数据； 作为输出时，通过写该位输出数据
3	PT1[3]	PT1.3 数据标志位 作为输入时，通过读该位获取输入数据； 作为输出时，通过写该位输出数据
2	PT1[2]	PT1.2 数据标志位 作为输入时，通过读该位获取输入数据； 作为输出时，通过写该位输出数据

1	PT1[1]	PT1.1 数据标志位 作为输入时，通过读该位获取输入数据； 作为输出时，通过写该位输出数据
0	PT1[0]	PT1.0 数据标志位 作为输入时，通过读该位获取输入数据； 作为输出时，通过写该位输出数据

**PT1EN 寄存器**

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
PT1EN			PT1EN[5:0]					

位地址	标识符	功能
7: 6	RESERVE	保留位
5	PT1EN[5]	PT1.5 输入输出控制标志，默认为输入口 1: PT1.5 做输出口 0: PT1.5 做输入口
4	PT1EN[4]	PT1.4 输入输出控制标志，默认为输入口 1: PT1.4 做输出口 0: PT1.4 做输入口
3	PT1EN[3]	PT1.3 输入输出控制标志，默认为输入口 1: PT1.3 做输出口 0: PT1.3 做输入口
2	PT1EN[2]	PT1.2 输入输出控制标志，默认为输入口 1: PT1.2 做输出口 0: PT1.2 做输入口
1	PT1EN[1]	PT1.1 输入输出控制标志，默认为输入口 1: PT1.1 做输出口 0: PT1.1 做输入口
0	PT1EN[0]	PT1.0 输入输出控制标志，默认为输入口 1: PT1.0 做输出口 0: PT1.0 做输入口

**PT1PU 寄存器**

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	R/W-0	R/W-0
PT1PU							PT1PU[1:0]	

位地址	标识符	功能
7: 2	RESERVE	保留位
1	PT1PU[1]	PT1.1 上拉电阻使能标志 1: 使能 PT1.1 上拉电阻 0: 断开 PT1.1 上拉电阻

0	PT1PU[0]	PT1.0 上拉电阻使能标志 1: 使能 PT1.0 上拉电阻 0: 断开 PT1.0 上拉电阻
---	----------	--

### 3.7.2 其他寄存器

#### INPUT 寄存器

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-0	R/W-0
	PT15_VDD	PT14_VDD	I2C_VDD	PT11_VDD	PT10_VDD	I2C_FLT	PT11_OD	PT10_OD

位地址	标识符	功能
7	PT15_VDD	PT1.5 与 1.8V 芯片通信时，需要设置 PT15_VDD=1
6	PT14_VDD	PT1.4 与 1.8V 芯片通信时，需要设置 PT14_VDD=1
5	I2C_VDD	当外部 I2C 工作电压为 1.8V 通信时，需要设置 I2C_VDD=1
4	PT11_VDD	PT1.1 与 1.8V 芯片通信时，需要设置 PT11_VDD=1
3	PT10_VDD	PT1.0 与 1.8V 芯片通信时，需要设置 PT10_VDD=1
2	I2C_FLT	I2C 口输入滤波使能 1: I2C 口输入滤波使能（默认） 0: I2C 口输入滤波关闭
1	PT11_OD	PT11 开漏输出使能 1: 开漏输出使能 0: 开漏输出关闭（默认）
0	PT10_OD	PT10 开漏输出使能 1: 开漏输出使能 0: 开漏输出关闭（默认）



## 4 增强功能

### 4.1 Halt 和 Sleep 模式

CSU18M68 支持低功耗工作模式。为了使 CSU18M68 处于待机状态，可以让 CPU 停止工作使 CSU18M68 进行停止或进入睡眠模式，减小功耗。这两种模式描述如下：

#### 4.1.1 停止模式

CPU 执行停止指令 (Halt) 后，程序计数器停止计数直到出现中断指令。为了避免由中断返回 (Interrupt Return) 引起的程序错误，建议在停止指令之后加一 NOP 指令以保证程序返回时能正常运行。在进入 Halt 时，必须手动关闭所有功能模块的使能信号，才能保证功耗达到 1.5uA 以下。

停止模式下可以唤醒的方式有：

- 外部中断 0
- 外部中断 1
- SDAD 中断溢出
- 定时器 0 溢出中断
- 定时器 1 溢出中断
- I2C 中断

#### 4.1.2 睡眠模式

CPU 执行睡眠指令 (Sleep) 后，内置 8M 振荡器停止工作(WDT 时钟不受影响)直到出现一个外部中断指令唤醒 CPU。为了避免由中断返回 (Interrupt Return) 引起的程序错误，建议睡眠指令之后加一 NOP 指令以保证程序的正常运行。在睡眠模式下的功耗大约有 1uA。

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，需要把 IO 口的上拉电阻断开，并且保证所有的 I/O 口是接到 DVDD 或 DGND 电平。

睡眠模式下可以唤醒的方式有：

- 外部中断 0
- 外部中断 1
- 定时器中断 0 (使用 WDT 时钟时)
- 定时器中断 1 (使用 WDT 时钟时)
- I2C\_Slave 广播唤醒或者匹配地址才唤醒

## 4.2 看门狗(WDT)

### 4.2.1 看门狗框图

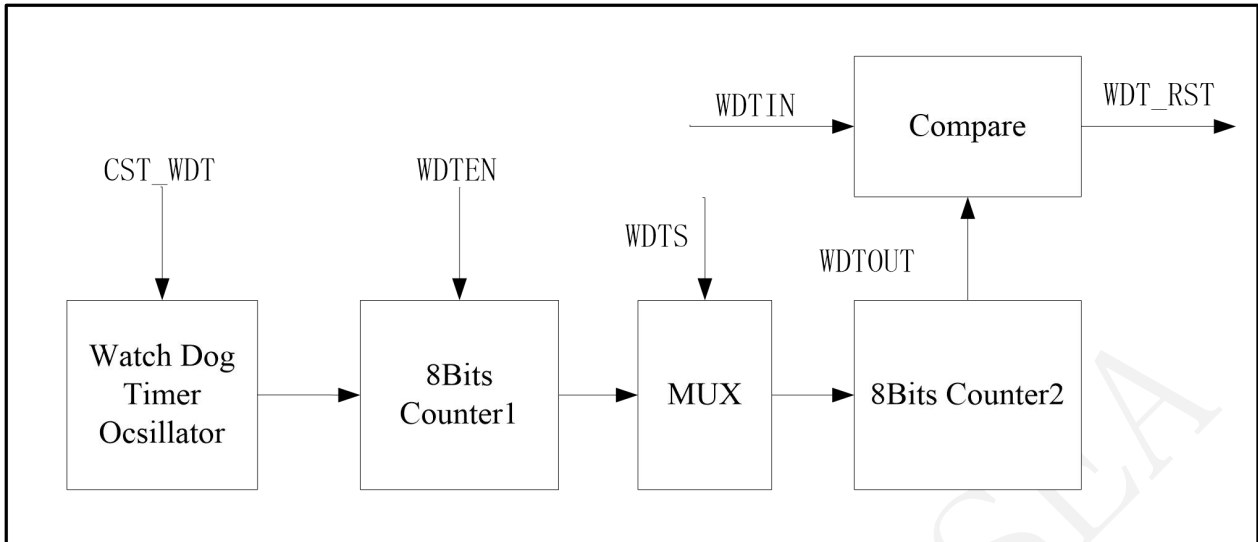


图 4-1 看门狗定时器功能框图

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 计时超时后将复位数字系统（复位标志不会被复位）。

当用户置位 WDTEN 时，“8 bits 计数器 1”开始计数，“8 bits 计数器 1”的输出是内部信号 WDTA[7:0]，被发送到一个受寄存器标志位 WDTS[2:0]控制的多路选择器，选择器的输出作为“8 bits 计数器 2”的时钟输入。当“8 bits 计数器 2”计数值与 WDT\_IN 数值相等时溢出，溢出时它会发送 WDTOUT 信号复位数字系统及置位 TO 标志位。用户可以使用指令 CLRWDT 复位 WDT 计数器。

在正常模式，WDT\_CLK\_EN 和 WDTEN 只能打开不能关闭，只能通过上电、掉电复位到默认值。

#### 4.2.2 看门狗寄存器列表

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				PD	TO	DC	C	Z	uuu00xxx
0EH	WDTCON	WDTEN	ROOT_EN		I2C_DIV[1:0]		WDTS[2:0]			00u00000
0FH	WDTIN	WDT_IN[7:0]								00000000
13H	ROOT	ROOT								00000000

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	R/W-0
WDTCON	WDTEN	ROOT_EN		I2C_DIV[1:0]		WDTS[2:0]		

#### WDTCON 寄存器各位功能表

位地址	标识符	功能
7	WDTEN	看门狗使能位 1: 看门狗使能，看门狗计数器启动，需 ROOT_EN=1 后才能配置该位 0: 看门狗未使能 注意：在正常模式下，WDTEN 使能打开配置为 1，不能关闭，只有在

		上电/掉电复位后才会被复位成 0。																		
6	ROOT_EN	WDT 配置权限使能，需对 ROOT 寄存器按顺序写入 35H、C8H、17H 后解锁写保护。 1: WDT 允许配置 0: WDT 禁止配置 受 ROOT_EN 控制的 WDT 配置有 WDTEN、WDTS、WDTIN																		
5	RESERVE	保留位																		
4: 3	I2C_DIV	I2C 工作时钟配置寄存器 00: I2C 工作时钟为 ICK/2 01: I2C 工作时钟为 ICK/4 10: I2C 工作时钟为 ICK/8 11: I2C 工作时钟为 ICK/16																		
2: 0	WDTS	WDT 计数器时钟配置，需 ROOT_EN=1 后才能配置该位 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>WDTS</th> <th>WDT 计数器时钟</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>WDT_CLK/256</td> </tr> <tr> <td>001</td> <td>WDT_CLK/128</td> </tr> <tr> <td>010</td> <td>WDT_CLK/64</td> </tr> <tr> <td>011</td> <td>WDT_CLK/32</td> </tr> <tr> <td>100</td> <td>WDT_CLK/16</td> </tr> <tr> <td>101</td> <td>WDT_CLK/8</td> </tr> <tr> <td>110</td> <td>WDT_CLK/4</td> </tr> <tr> <td>111</td> <td>WDT_CLK/2</td> </tr> </tbody> </table>	WDTS	WDT 计数器时钟	000	WDT_CLK/256	001	WDT_CLK/128	010	WDT_CLK/64	011	WDT_CLK/32	100	WDT_CLK/16	101	WDT_CLK/8	110	WDT_CLK/4	111	WDT_CLK/2
WDTS	WDT 计数器时钟																			
000	WDT_CLK/256																			
001	WDT_CLK/128																			
010	WDT_CLK/64																			
011	WDT_CLK/32																			
100	WDT_CLK/16																			
101	WDT_CLK/8																			
110	WDT_CLK/4																			
111	WDT_CLK/2																			

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
WDTIN	WDTIN[7:0]							

WDTIN 寄存器各位功能表

位地址	标识符	功能
7: 0	WDTIN	WDT 计数窗值配置寄存器，需 ROOT_EN=1 后才能配置该位 当 WDTEN=1 后，WDT 计数器从 0 开始计数，当计数值等于 WDTIN 时，产生看门狗复位。 注意，WDTIN 不能配置为 0，当 WDTIN 配置为 0 时，

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

WDTIN 寄存器各位功能表

位地址	标识符	功能
7: 0	ROOT	ROOT_EN 保护寄存器，当按顺序对 ROOT 连续写入 35H、C8H、17H

		后，可改写 ROOT_EN 寄存器。
--	--	--------------------

操作：

- 1、对 ROOT 连续写入 35H、C8H、17H 解锁 ROOT\_EN 写保护；
- 2、配置 ROOT\_EN=1; WDT\_CLK\_EN=1
- 3、配置 WDTS[2:0]，选择 WDT 时钟频率；
- 4、配置 WDTIN[7:0]，选择 WDT 溢出窗值；
- 5、置位 WDTEN，使能 WDT；
- 6、对 ROOT 连续写入 35H、C8H、17H 重新解锁 ROOT\_EN 写保护；
- 7、配置 ROOT\_EN=0，禁止改写 WDT 配置；
- 8、定时执行 CLRWDT 指令复位 WDT 计数器；

WDTS[2:0]	计数器时钟	时间（当 WDTIN==BBH）
000	WDTA[7] : WDT_CLK/256	16s
001	WDTA[6] : WDT_CLK/128	8s
010	WDTA[5] : WDT_CLK/64	4s
011	WDTA[4] : WDT_CLK/32	2s
100	WDTA[3] : WDT_CLK/16	1s
101	WDTA[2] : WDT_CLK/8	0.5s
110	WDTA[1] : WDT_CLK/4	0.25s
111	WDTA[0] : WDT_CLK/2	0.125s

## 4.3 电源系统

### 4.3.1 Regulator

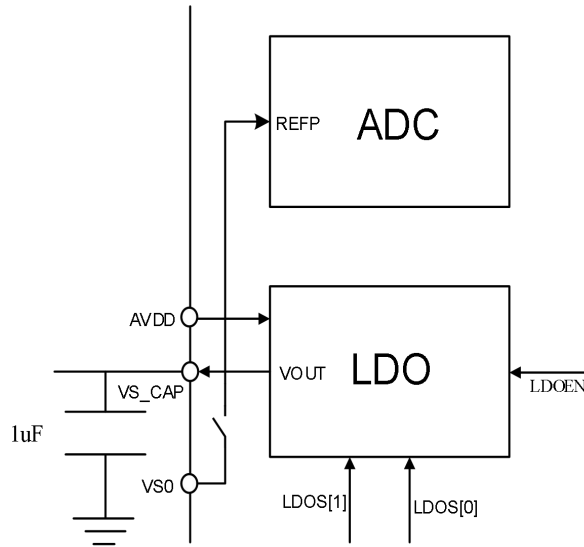


图 4-2 稳压电路

如图 4-2 所示,用于产生 VS 作为传感器和 16bit ADC 的参考电压,通过选择 LDOS 可以使输出 2.45V, 2.6V, 2.8V, 3.0V 可选。LDOEN 作为 LDO 的使能信号。LDO 的控制寄存器标志是 LDOEN 与 LDOS。输出电压是 VS。

稳压电路寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
17H	ANACFG	LDOEN	LDOS[1:0]		BGR_ENB	BGID	SINL[1:0]		ADEN	0000000

ANACFG 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-0	R/W-0	R/W-0	R/W-0	R/W-0
ANACFG	LDOEN	LDOS[1:0]		BGR_ENB	BGID	SINL[1:0]		ADEN

位地址	标识符	功能
7	LDOEN	LDO 使能控制 0: LDO 关闭 1: LDO 打开
6: 5	LDOS	VS 电压值选择 00: VS=2.45V 01: VS=2.6V 10: VS=2.8V 11: VS=3.0V

4	BGR_ENB	Bandgap 使能（只读） 0: Bandgap 打开 1: Bandgap 关闭
3: 1	BGID SINL	ADC 通道选择 000: 16bit ADC 输入端连接到 AIN0 和 AIN1, AIN0 为 $V_{in+}$ , AIN1 为 $V_{in-}$ ; 001: 内短; 010: 16bit ADC 输入端连接到 TEMP; 011: 16bit ADC 输入端连接到 AIN2 和 AIN3, AIN2 为 $V_{in+}$ , AIN3 为 $V_{in-}$ ; 100: 16bit ADC 输入端连接到 AIN0 和 VS/2, AIN0 为 $V_{in+}$ , VS/2 为 $V_{in-}$ ; 101: 16bit ADC 输入端连接到 AIN1 和 VS/2, VS/2 为 $V_{in+}$ , AIN1 为 $V_{in-}$ ; 110: 16bit ADC 输入端连接到 AIN2 和 VS/2, AIN2 为 $V_{in+}$ , VS/2 为 $V_{in-}$ ; 111: 16bit ADC 输入端连接到 AIN3 和 VS/2, VS/2 为 $V_{in+}$ , AIN3 为 $V_{in-}$ ;
0	ADEN	16bit ADC 使能标志 0: 16bit ADC 不使能 1: 16bit ADC 使能

### 4.3.2 低电压比较器

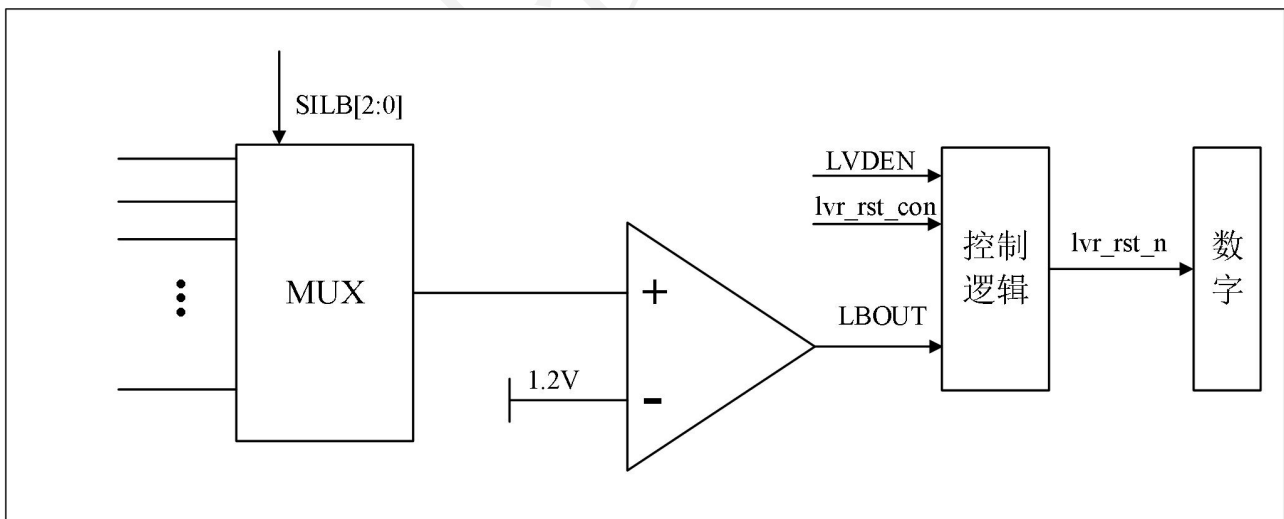


图 4-3 低电压比较功能模块框图

低电压比较器用于 DVDD 的低电压检测。CSU18M68 集成一个可产生  $1/2DVDD$  及  $1/3DVDD$  的分压器。多路选择器用于选择不同的分压连接到低电压比较器的输入端。多路选择器的输出与 1.2V 进行比较，它的控制寄存器标志是 SILB[2:0]及 LVDEN，比较器的输出是 LBOUT，LBOUT 为只读，在 LVDEN=1 打开 LB\_RST\_CON，会对 LBOUT 进行判断，如果 LBOUT=0 的时间超过 130us，芯片产生复位。

## 低电压比较器寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
1BH	LVDCON	LVDEN		AIENB1		SILB[2:0]		LBOUT	LB_RST_CON	0u1000x0

操作：

1. 设置寄存器标志位 LVDEN，使能低电压比较器。
2. 比较器输出是 LBOUT。

**注：在 sleep 模式下 LVD 模块不工作，另外 ICK\_EN=0，。**

## 低电压比较器检测电压的选择列表

SILB[2:0]	检测电压	满足条件	则
000	DVDD	DVDD>2.4V	LBOUT=1
001	DVDD	DVDD>2.5V	LBOUT=1
010	DVDD	DVDD>2.6V	LBOUT=1
011	DVDD	DVDD>2.7V	LBOUT=1
100	DVDD	DVDD>2.8V	LBOUT=1
101	DVDD	DVDD>3.0V	LBOUT=1
110	LPD	LPD>1.2V	LBOUT=1
111	DVDD	DVDD>3.6V	LBOUT=1

**LVDCON 寄存器（地址=1BH）**

特性	R/W-0	U-0	R/W-1	R/W-0	R/W-0	R/W-0	R-x	R/W-0
NETE	LVDEN		AIENB1		SILB[2:0]		LBOUT	LB_RST_CON
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能												
7	LVDEN	低电压检测使能信号 <table border="1"> <tr> <td>LVDEN</td> <td></td> </tr> <tr> <td>0</td> <td>低电压检测不使能</td> </tr> <tr> <td>1</td> <td>低电压检测使能</td> </tr> </table>	LVDEN		0	低电压检测不使能	1	低电压检测使能						
LVDEN														
0	低电压检测不使能													
1	低电压检测使能													
5	AIENB1	LPD 口使能信号，低有效 0：LPD 口使能，PT1.5 数字功能关闭，做模拟 LPD 输入 1：LPD 口关闭，PT1.5 数字功能开启，做数字 IO 口												
4:2	SILB[2:0]	低电压比较器检测电压的选择 <table border="1"> <thead> <tr> <th>SILB[2:0]</th> <th>检测电压</th> <th>满足条件</th> <th>则</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>DVDD</td> <td>DVDD&gt;2.4V</td> <td>LBOUT=1</td> </tr> <tr> <td>001</td> <td>DVDD</td> <td>DVDD&gt;2.5V</td> <td>LBOUT=1</td> </tr> </tbody> </table>	SILB[2:0]	检测电压	满足条件	则	000	DVDD	DVDD>2.4V	LBOUT=1	001	DVDD	DVDD>2.5V	LBOUT=1
SILB[2:0]	检测电压	满足条件	则											
000	DVDD	DVDD>2.4V	LBOUT=1											
001	DVDD	DVDD>2.5V	LBOUT=1											

		010	DVDD	DVDD>2.6V	LBOUT=1
		011	DVDD	DVDD>2.7V	LBOUT=1
		100	DVDD	DVDD>2.8V	LBOUT=1
		101	DVDD	DVDD>3.0V	LBOUT=1
		110	LPD	LPD>1.2V	LBOUT=1
		111	DVDD	DVDD>3.6V	LBOUT=1
1	LBOUT	低电压检测标志，只读。 参见 SILB[2:0]说明			
0	LB_RST_CON	低电压复位使能控制信号，仅在 LVDEN=1 时该控制位才有意义 =0 时低电压复位使能关闭 =1 时低电压复位使能开启；在 LVDEN=1 且 LBOUT=0 的时间超过 100us 时，芯片内部产生低电压复位信号，复位整个数字电路。			

#### 4.4 I2C 从机



CSU18M68 集成一个 I2C 从机模块，可支持 8 位双向的数据收发。

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

支持 7 位地址寻址，默认从机地址为 0x26，从机地址寄存器可配

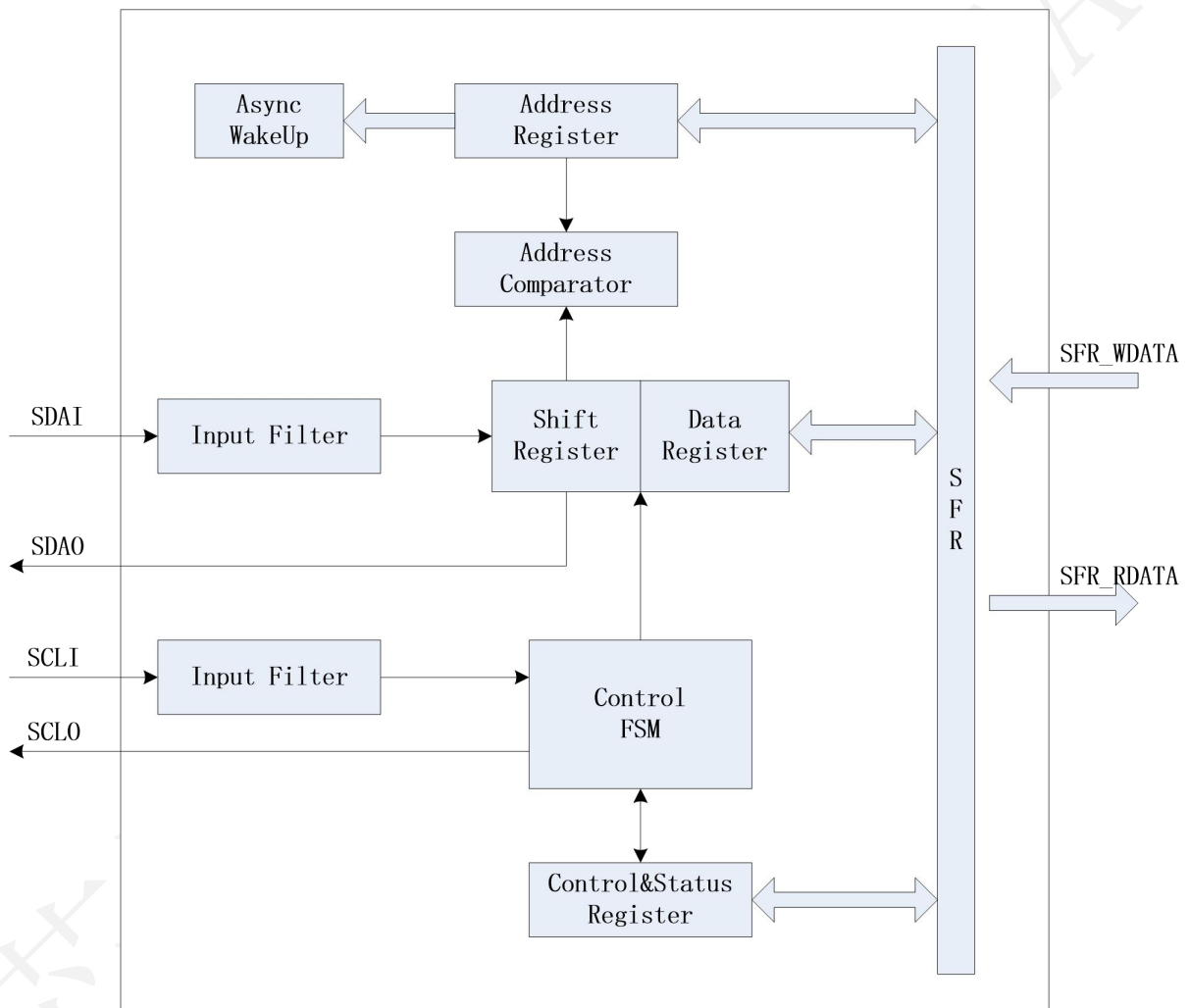
支持广播地址呼叫

支持时钟低电平延长

支持 SCL/SDA 输入滤波

支持从机接收模式和从机发送模式；

- 接收响应状态（ACK or NO ACK）软件可配
- 支持 Sleep 模式异步唤醒，支持 Halt 模式唤醒

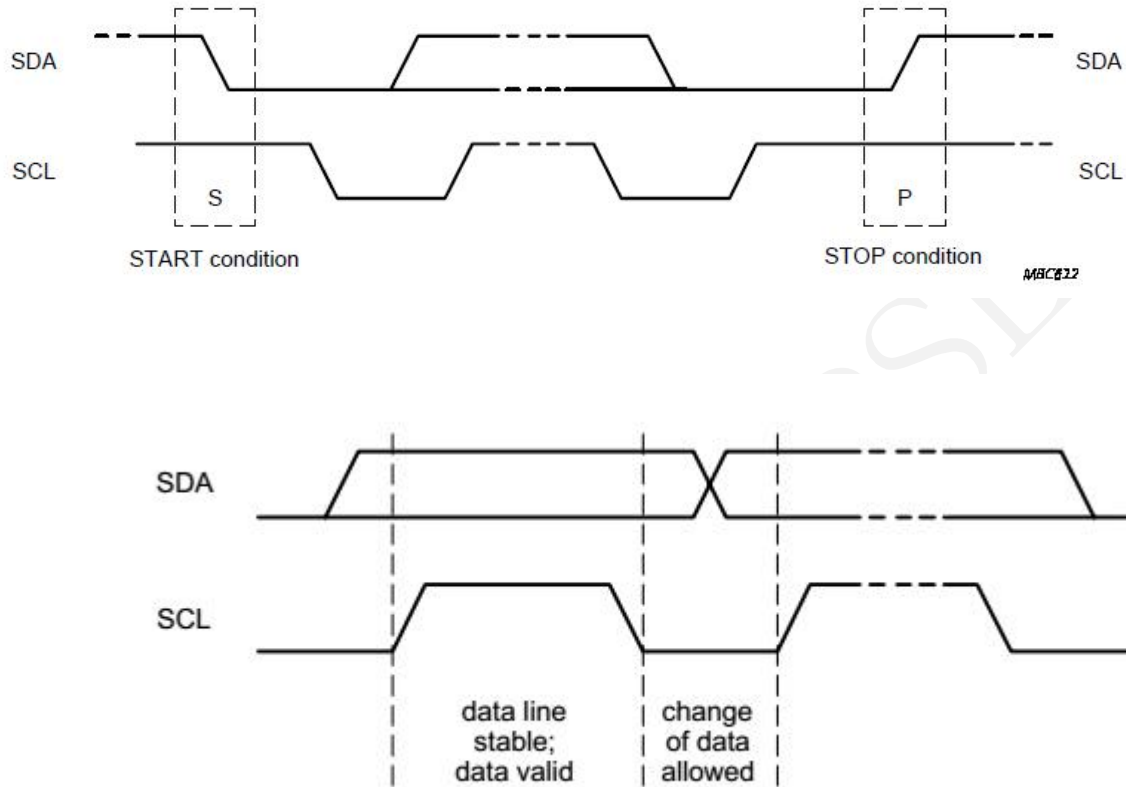


#### 4.4.1 I2C 数据传输起始和终止条件

当 SCL 线为高电平时，SDA 线由高电平向低电平切换，定义为 I2C 传输起始 START。

当 SCL 线为高电平时，SDA 线由低电平向高电平切换，定义为 I2C 传输终止 STOP。

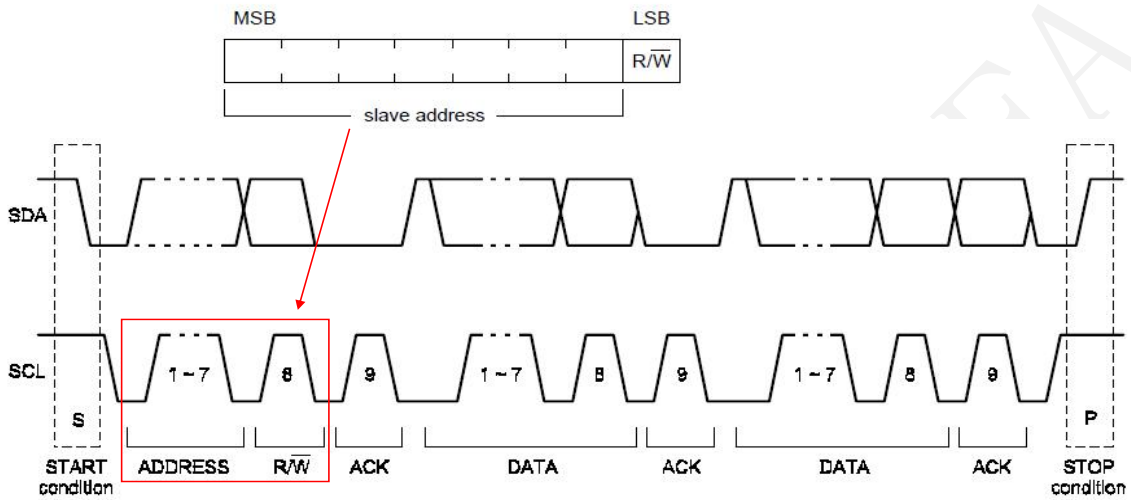
在正常的数据传输过程中，数据只允许在 SCL 为低时候切换，SCL 为高时传输数据保持不变。



#### 4.4.2 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 主机发送从机地址。这个地址共有 7 位, 紧接着的第 8 位为数据方向位(R/W), 该位为 0 表示主机向从机发送数据, 该位为 1 表示主机请求从从机读取数据。从机接收到主机发送的 7 位寻址地址以后, 会跟从机本身的地址进行比较, 如果地址相同, 从机向主机发送相应信号 (ACK)。如果主机发送的地址跟从机本身的地址不同, 从机则不响应该寻址。

当主机发送的从机地址为 0 时, 主机通过广播呼叫来寻址 I2C 总线上的每一个器件, 支持广播呼叫功能的器件如果需要跟主机交互数据, 需要发送响应 (ACK)。



#### 4.4.3 数据传输时序

主机和从机之间的数据传输遵从高位先传低位后传原则。

主机发送数据，从机接收数据时序如下图 4-4-1。

主机接收数据，从机发送数据时序如下图 4-4-2。

复合传输格式，传输改变方向的时候，起始条件和从机地址都会被重复发送，但 R/W 位取反。复合传输时序如下图 4-4-3。

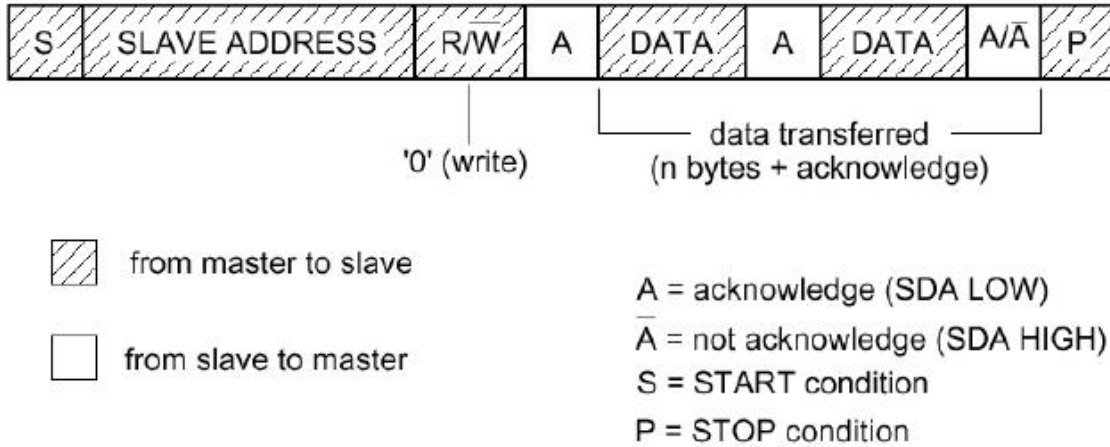


图 4-4-1 主机发送从机接收时序

图

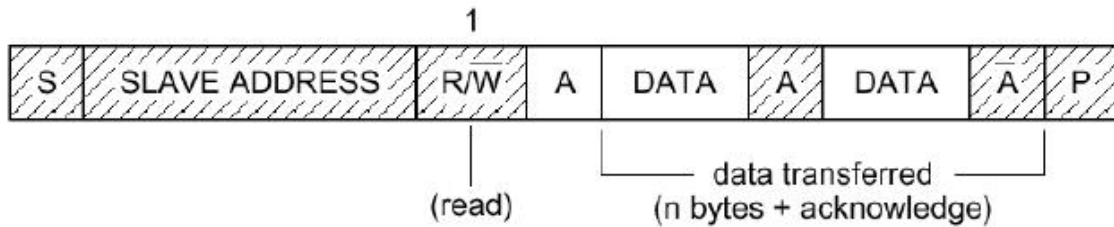


图 4-4-2 主机接收从机发送时序图

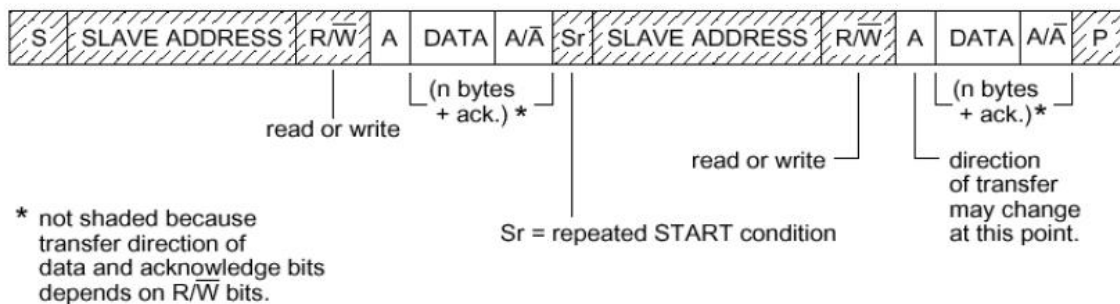


图 4-4-3 复合传输时序图

**4.4.4 寄存器说明**

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bit0	上电复位值
5A	INTF3	I2C_TIF	I2C_RIF				I2C_STIF			00uuu0uu
5B	INTE3	I2C_TIE	I2C_RIE				I2C_STIE			00uuu0uu
0E	WDTCO N	WDTEN	ROOT_EN		I2C_DIV[1:0]		WDTS[2:0]			00u00000
68	I2CCON	I2C_EN	AWK_EN	CST_E N	ACK_EN	I2CSTUS[3:0]				00000000
69	I2CDAT		I2CDAT[7:0]							00000000
6A	I2CISR								TXE	uuuuuuu0
7F	I2CADR	I2CADR[6:0]							GC_EN	01001100

INTF3 寄存器各位功能表

位地址	标识符	功能
7	I2C_TIF	I2C 从机发送中断标志位 1 : I2C 从机请求 MCU 写入发送数据中断, 必须软件清 0 0 : 没发生 I2C 从机请求 MCU 写入发送数据中断
6	I2C_RIF	I2C 从机接收中断标志位 当 I2C_EN=1 且 AWK_EN=0 时: 1 : I2C 从机完成 1-Byte 数据接收中断, 必须软件清 0 0 : 没发生 I2C 从机接收中断 当 I2C_EN=0 且 AWK_EN=1 时: 1 : 在 Sleep 模式下 I2C 从机完成地址匹配产生中断, 唤醒系统, 系统唤醒后需要软件配置 AWK_EN=0 清除该中断(当前地址匹配会被 I2C 从机忽略, 如需数据传输需要 I2C 主机重新寻址从机) 0 : 没发生 I2C 从机地址匹配中断
2	I2C_STIF	I2C 从机开始中断标志位 1 : 发生 I2C 从机开始中断, 必须软件清 0 0 : 没发生 I2C 从机开始中断

INTE3 寄存器各位功能表

位地址	标识符	功能
7	I2C_TIE	I2C 从机发送中断使能 1 : 使能 I2C 从机发送中断 0 : 关闭 I2C 从机发送中断
6	I2C_RIE	I2C 从机接收中断使能 1 : 使能 I2C 从机接收中断 0 : 关闭 I2C 从机接收中断

2	I2C_STIE	I2C 从机开始中断使能 1 : 使能 I2C 从机开始中断 0 : 关闭 I2C 从机开始中断
---	----------	--

**WDTCON 寄存器**

位地址	标识符	功能
4: 3	I2C_DIV	I2C 工作时钟配置 00: I2C 工作时钟为 ICK/2 01: I2C 工作时钟为 ICK/4 10: I2C 工作时钟为 ICK/8 11: I2C 工作时钟为 ICK/16

**I2CCON 寄存器**

位地址	标识符	功能
7	I2C_EN	I2C 从机使能 1: 使能 I2C 从机 (需要关闭异步唤醒使能 AWK_EN=0) 0: 关闭 I2C 从机
6	AWK_EN	I2C 异步唤醒使能 1: 在系统处于 Sleep 模式, 如果 I2C 从机接收到主机发送的地址跟本机的地址相同 (如果 GC_EN=1 则可以响应广播地址), 则通过 I2C_RIF 给出中断, 唤醒系统 (需要关闭 I2C 从机使能 I2C_EN=0) 0: 关闭异步唤醒
5	CST_EN	I2C 从机时钟低电平延长使能 1: 使能 I2C 从机时钟低电平延长 (必须设置为 1) 0: 关闭 I2C 从机时钟低电平延长
4	ACK_EN	I2C 从机响应使能 1: 使能 I2C 从机响应 0: 关闭 I2C 从机响应
3: 0	I2CSTUS	I2C 从机状态寄存器, 只读 4'h0 : I2C_IDLE 4'h1 : I2C_STDET 4'h2 : I2C_ADDR 4'h3 : I2C_ADDRACK 4'h4 : I2C_RX 4'h5 : I2C_RXACK 4'h6 : I2C_RXNACK 4'h7 : I2C_TX 4'h8 : I2C_TXACK 4'h9 : I2C_CLKSTR 4'hF : I2C_ERR

**I2CDAT 寄存器**

位地址	标识符	功能
7: 0	I2CDAT	I2C 数据寄存器 当 I2C 工作在从机接收模式，完成 1-Byte 数据接收后可以从该寄存器读出接收的数据 当 I2C 工作在从机发送模式，写入该寄存器的数据将被发送给主机

**I2CADR 寄存器**

位地址	标识符	功能
7: 1	I2CADR	I2C 从机地址，默认为 0x26
0	GC_EN	I2C 从机广播地址响应使能 1: 使能 I2C 从机广播地址响应（是否响应还要看 ACK_EN 的状态） 0: 关闭 I2C 从机广播地址响应

**I2CISR 寄存器**

位地址	标识符	功能
7: 1	RESERVE	保留位
0	TXE	I2C 发送数据缓存空标志 1: I2C 发送数据缓存为空 0: I2C 发送数据缓存为非空 在 I2C 使能时，或者写入 I2CDAT 数据发送后，此位硬件置 1。在要发送的数据写入 I2CDAT 后，此位清零。 如果需要提前将数据写入 I2CDAT，可以软件将此位置 1，然后再 I2CDAT 写入要发送的数据。

#### 4.4.5 I2C 使用说明

操作:

- 1) 配置 I2C 从机的时钟分频系数 I2C\_DIV[1:0]，确保 I2C 从机内部工作时钟大于 I2C 传输频率的 4 倍以上。例如 I2C 工作在快速模式（400kbit/s）且时钟源为 4MHz，需要配置 I2C\_DIV[1:0]=0。
- 2) 配置 I2C 从机的地址 I2CADR[7:1]，默认为 0x26。
- 3) 使能 I2C 从机接收中断和发送中断 I2C\_RIE=1，I2C\_TIE=1。
- 4) 使能 I2C 从机，关闭异步唤醒功能，使能低时钟延长，使能从机响应，I2C\_EN=1，AWK\_EN=0，CST\_EN=1，ACK\_EN=1。
- 5) 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。
- 6) 从机接收数据：
  - a).I2C 从机判断主机发送的 RW 位，如果 RW=0，从机自动进入数据接收模式。
  - b).当从机接收完 1-Byte 数据，如果此时 I2CDAT 寄存器此时状态已经为满，则回 NACK 给主机，该笔数据将被丢弃（不移入 I2CDAT 寄存器）。如果此时 I2CDAT 寄存器为空，则回 ACK 给主机，并将接收到的数据移入 I2CDAT 寄存器中，同时发起 I2C\_RIF 中断通知 MCU 读取数据。
  - c).软件读取 I2CDAT 寄存器，得到 I2C 主机发送的 1-Byte 数据，在每次数据从 I2CDAT 寄存器读走后 I2C\_RIF 自动清除一次。
  - d).根据 I2C 主机和从机间的通讯协议，当最后一笔数据接收完成，主机发送 STOP 指令，从机退出数据接收。

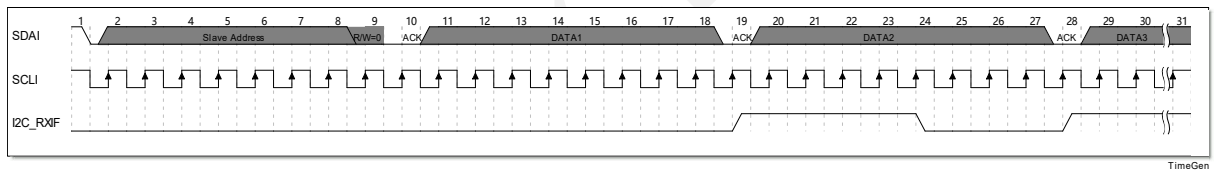


图 4-4-4 从机连续接收时序图

- 7) 从机发送数据：
  - a). I2C 从机判断主机发送的 RW 位，如果 RW=1，从机自动进入数据发送模式。
  - b).如果 I2CDAT 寄存器中当前有发送数据，从机自动把数据移入发送移位寄存器中，开始发送 1-Byte 数据，同时发起 I2C\_TIF 中断。
  - c).如果 I2CDAT 寄存器中当前没有发送数据，从机自动进入低时钟延长状态，同时发起 I2C\_TIF 中断，等待软件写入发送数据。
  - d).在从机发送中断服务程序中，软件向 I2CDAT 中写入 1-Byte 数据。
  - e).根据 I2C 主机和从机间的通讯协议，当主机接收完最后一笔数据后，主机回复 NACK，从机释放总线。主机发送 STOP 指令，从机退出数据发送。

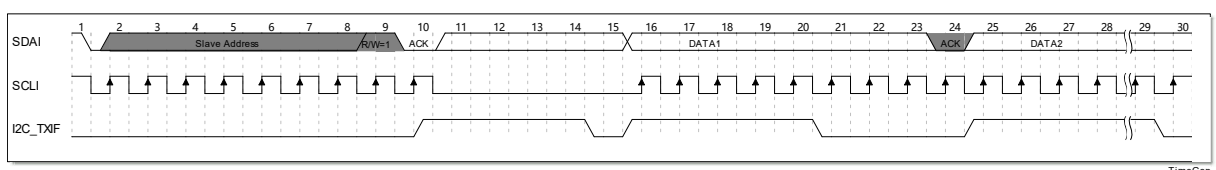


图 4-4-5 从机连续发送时序图



8) 从机接收转从机发送:

- a). 主机先向从机发送数据，从机接收数据，软件在接收中断中读取接收到的数据。
- b). 根据 I2C 主机和从机的通讯协议，软件判断接下来需要做的动作。如果需要继续接收数据，则等待下一个接收中断起来继续读取接收的数据。如果可以提前判断接下来需要从机发送数据给主机，在读取完从机接收的最后一笔数据后，往 I2CDAT 寄存器中写入将要发送的 1-Byte 数据，可以降低从机发送模式进入低时钟延长的概率，是提升从机发送数据效率的可选项之一。

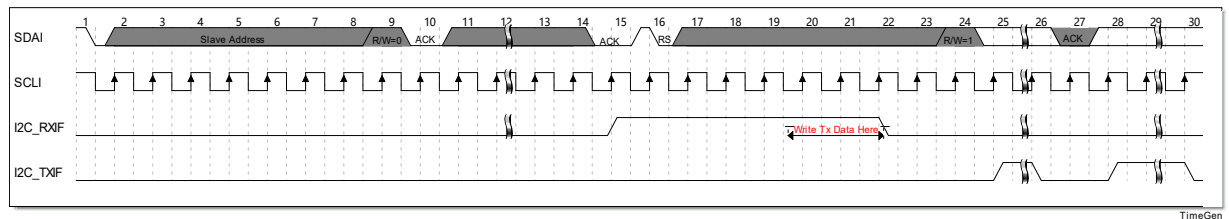


图 4-4-6 从机接收转从机发送时序图

- 9) 当 I2C 从机检查到主机发送的 STOP 信号，完成数据收发。如果不需要继续跟 I2C 主机传输数据，需要配置 I2C\_EN=0, CST\_EN=0, ACK\_EN=0, 关闭 I2C 从机以减小功耗。
- 10) 当要使用 I2C 从机异步唤醒功能时，在系统进入 Sleep 模式之前，配置 I2C\_EN=0, AWK\_EN=1, CST\_EN=0, ACK\_EN=0, 系统进入睡眠模式后，当主机发送的地址与从机相同（如果 GC\_EN=1 则可以相应广播地址），I2C 从机产生 I2C\_RIF 中断，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。
- 11) I2C 从机可以把主机从 Halt 模式下唤醒并立即开始数据接收，在系统进入 Halt 模式之前，配置 I2C\_EN=1, AWK\_EN=0, CST\_EN=1, ACK\_EN=1, 系统进入 Halt 模式后，当主机发送的地址与从机相同（如果 GC\_EN=1 则可以相应广播地址），I2C 从机响应主机并回复 ACK，当 I2C 从机接收到 1Byte 数据产生 I2C\_RIF，唤醒系统，当前数据有效，软件读取 I2CDAT 获取该笔数据。

## 4.5 16bit-ADC 模块

CSU18M68 包含一个 16 位的 sigma delta 型的模数转换器 (ADC)。该 ADC 的基准由内部的基准电压 VS 提供，但是也可以由外部提供基准电压（此时需要关闭内部的基准电压，将外部的基准电压接入 REFP）。同时还提供内部温度检测以及内短测试功能。

首先打开 VS (LDOEN)，然后配置好 16Bit ADC 的各种配置参数，打开全局中断使能以及 16bit ADC 中断使能，然后将 ADEN 打开，在刚开始工作时，数字滤波器需要一段建立时间，对于 4 阶数字滤波器，需要 4 个数据转换周期的建立时间。

### 4.5.1 16bit ADC 寄存器说明

16bit ADC 功能模块相关寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
06H	INTF	-		TM1IF	TM0IF		ADIF	E1IF	E0IF	uu00u000
07H	INTE	GIE		TM1IE	TM0IE		ADIE	E1IE	E0IE	0u00u000

11H	ADOH	ADO[15:8]						00000000
1CH	ADOL	ADO[7:0]						00000000
6BH	ADCON	VS0_OEN	VS_LIMT	AINOUT[1:0]			ADM[1:0]	0000uu00
16H	ADCFG	ADSC		S_GAIN[1:0]				0uu00u00
17H	ANACFG	LDOEN	LDOS[1:0]	BGR_ENB	BGID	SINL[1:0]	ADEN	00000000

### ADOH 寄存器

特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ADOH	ADO[15:8]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

### ADOL 寄存器

特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ADOL	ADO[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 15-0 ADO[15:0]: 16bit ADC 数字输出

ADO[15] = 16bit ADC 数字输出符号位。0 = 输出为正；1 = 输出为负。

ADO[14] = 16bit ADC 数字输出数据 bit14

~

ADO[0] = 16bit ADC 数字输出数据 bit0

### ADCON 寄存器（地址为 6BH）

特性	RW-0	RW-0	RW-0	RW-0	U-0	U-0	R/W-0	R/W-0
ADCON	VS0_OEN	VS_LIMT	AINOUT[1:0]				ADM[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7 VS0\_OEN: VS0 输出使能

1: 使能 VS0 输出

0: 禁止 VS0 输出

VS\_LIMT: VS 限流配置

1: 关闭限流

0: 打开限流（默认）

AINOUT [1: 0] :

00: AIN0/AIN1/AIN2/AIN3 正常使用

01: AIN0/AIN1 输出 1/29\*VS 差分信号

10: AIN2/AIN3 输出 1/29\*VS 差分信号

11: AIN0/AIN1/AIN2/AIN3 正常使用

Bit1-0 ADM[1:0]:16bit ADC 降采样速率选择寄存器

VS\_LIMT/VS0\_OEN 配置要求:

在芯片上电后 VS\_LIMT=0 默认限流; 在配置 LDO\_EN=1 后, 软件延时 1ms (至少 500us) 后, 配置 VS\_LIMT=1 关闭限流, 之后配置 VS0\_OEN 使能 VS0 输出;

在睡眠唤醒后, LDO\_EN、VS\_LIMT、VS0\_OEN 依次打开, 不需要软件延时。

16bit ADC 输出速率选择列表

ADM[1:0]	降采样	DataRate(Hz)
00	64 (4 阶 Comb)	CK_SMP/64
01	128 (4 阶 Comb)	CK_SMP/128
10	256 (4 阶 Comb)	CK_SMP/256
11	512 (4 阶 Comb)	CK_SMP/512

ADCFG 寄存器

特性	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0
NETA	ADSC			S_GAIN[1:0]				
	Bit7	Bit6		Bit4	Bit3	Bit2	Bit1	Bit0

Bit7 ADSC: ADC 采样时钟频率选择

0: ADC 采样时钟频率为 ICK/8, 当负载小于 14K 时建议配置该档位

1: ADC 采样时钟频率为 ICK/4, 当负载小于 6K 且有速度需求时建议配置该档位

Bit4-3 S\_GAIN: ADC 的 PGA 选择:

Gain	S_GAIN[1:0]
1	00
5	01
10	10
20	11

ANACFG 寄存器 (地址为 17H)

特性	R/W-0	R/W -0	R/W -0	R -0	R/W-0	R/W-0	R/W-0	R/W -0
	LDOEN	LDOS[1:0]		BGR_ENB	BGID	SINL[1:0]		ADEN
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7 LDOEN: LDO 电源使能信号

LDOEN=1: LDO 电源使能

LDOEN=0: LDO 电源不使能

Bit6~5 LDOS[1:0]: VS 电压值选择

LDOS[1:0] 00 VS=2.45V

LDOS[1:0] 01 VS=2.6V

LDOS[1:0] 10 VS=2.8V

LDOS[1:0] 11 VS=3.0V

Bit4 BGR\_ENB: bandgap 使能

BGR\_ENB =0 打开 bandgap

BGR\_ENB =1 关闭 bandgap

Bit3-1 BGID&SINL: ADC 通道选择,

{BGID, SINL[1:0]} :

000 = 16bit ADC 输入端连接到 AIN0 和 AIN1, AIN0 为  $V_{in+}$ , AIN1 为  $V_{in-}$ ;

001 = 内短;

010 = 16bit ADC 输入端连接到 TEMP;

011 = 16bit ADC 输入端连接到 AIN2 和 AIN3, AIN2 为  $V_{in+}$ , AIN3 为  $V_{in-}$ ;

100 = 16bit ADC 输入端连接到 AIN0, AIN0 为  $V_{in+}$ ,  $VS/2$  为  $V_{in-}$

101 = 16bit ADC 输入端连接到 AIN1,  $VS/2$  为  $V_{in+}$ , AIN1 为  $V_{in-}$

110 = 16bit ADC 输入端连接到 AIN2, AIN2 为  $V_{in+}$ ,  $VS/2$  为  $V_{in-}$

111 = 16bit ADC 输入端连接到 AIN3,  $VS/2$  为  $V_{in+}$ , AIN3 为  $V_{in-}$

Bit 0ADEN: 16bit ADC 使能标志

1 = 16bit ADC 使能

0 = 16bit ADC 不使能

## 4.5.2 温度传感器

### 温度传感器使用

在应用阶段进行温度测量方式如下:

- 1、配置 SINL<1:0>=10 选择温度传感器输入 16bit ADC
- 2、S\_GAIN[1:0]=00, 选择 PGA=1。
- 3、ADM=11, 选择 512 降采样, VS0\_OEN = 1'b0 禁止 VS0 输出
- 4、ADSC=0
- 5、LDOS=xx
- 6、ADEN=1, 读取 5 笔 AD 值数据, 前 4 笔丢弃。第 5 笔数据为 ADT。
- 7、通过查表方式从代码选项读取对应 LDOS=xx 配置存放的 ADx 值。
- 8、通过查表方式从代码选项 0x2010 读取标定温度值 TEMP\_BD

当前温度公式:

当前温度=ADT\*TEMP\_BD/ADx/100-273.15

注: LDOS=00 对应存放标定温度 AD 值的代码选项地址 0x2011

LDOS=01 对应存放标定温度 AD 值的代码选项地址 0x2012

LDOS=10 对应存放标定温度 AD 值的代码选项地址 0x2013

LDOS=11 对应存放标定温度 AD 值的代码选项地址 0x2014

## 4.6 数据查表

电路要求：在线烧录时，使用芯片自带的升压电路。

地址要求：CSU18M68 芯片通过 PARH[5:0]和 PARL[7:0]寄存器来选择地址，寻址空间是 0000H~201FH，包含程序存储器(地址范围 0000H~1FFFH)和代码选项 (地址范围 2000H~201FH)；

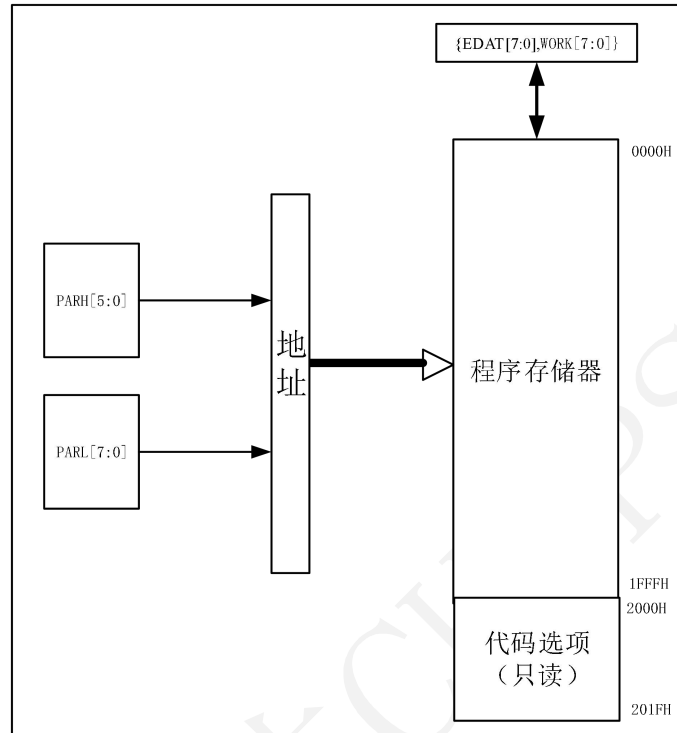


图 4-5 在线烧录地址寻址示意图

在线烧录寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
05H	WORK	工作寄存器								00000000
0AH	EADRH	PROG_B USY	READ_C HECK	PARH[5:0]						00000000
0BH	EADRL	PARL[7:0]								00000000
0CH	EDAT	EDAT[7:0]								11111111
0DH	EOPEN	EOPEN[7:0]								00000000

**EOPEN**：在线烧录保护寄存器，在对 MTP 烧录时必须对该地址写入 AAH、55H、A5H；对其他地址寄存器进行写操作时，EOPEN 寄存器会被清零。(除了 work 寄存器)

**EADRH /EADRL**：提供 MTP 在线烧录或者在线读 MTP/OPTION 的地址。

**EDAT**：提供 MTP 在线烧录时的烧录数据或者在线读 MTP/OPTION 时的读出的数据高 8 位。

**Work**：提供 MTP 在线烧录时的烧录数据或者在线读 MTP/OPTION 时的读出的数据低 8 位。

**READ\_CHECK**：当 MTP 在线烧录完毕后，硬件会自动对该地址的数据读出后比较，如果发生错误，此 Bit 为 1，如果正确此 Bit 为 0。

**PROG\_BUSY**：进入 MTP 在线烧录 可以打开 WDT 计时，如果出现烧录异常一直没有等到 Busy 信号拉低，WDT 溢出会终止烧录，在 WDT 复位 PC 后可以读取此 Bit，如果此 Bit 还是为 1，表示 WDT 复位时 BUSY 还处于高。

操作方式：

在线烧录时：

1. 使用 CLRWDT 指令复位 WDT；
2. 将烧录地址写入 EADRH, EADRL 寄存器；
3. 高八位数据写入 EDAT 寄存器；
4. 对 EOPEN 写入相应的数据；
5. 将烧录数据低八位数据写入 work 寄存器；注意此步骤不能提前。
6. 用在线烧录指令（TBLP）烧录；

注：在线烧录时，要打开 WDT，但在对 EOPEN 操作之前先使用 CLRWDT 指令复位 WDT。

在线读 MTP/OPTION 数据时：

1. 将读 MTP/OPTION 地址写入 EADRH, EADRL 寄存器；
2. 用在线读 MTP 指令（MOVP）读出 MTP/OPTION 数据，执行该指令后，读出存放在 EDAT、work 寄存器；

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 WORK 寄存器中。执行一次读操作需要 6 个指令周期。

注：在使用 TBLP 指令时，不能开中断。

## 4.7 代码选项

### 1) OPTIION0

位地址	标识符	功能
15	RFU	
14:1	SECU_NUM	当 MTP ROM 区被加密后，无法进行读操作，也不能改写保密位，当 MTP 被加密后，只有对 MTP ROM 进行从 0 地址到 SECU_NUM 地址重写写入操作后，才能改写保密位，和 SECU_NUM。
0	SECURITY	代码保密位 0：使能代码加密 1：禁止代码加密

### 1) 温度传感器校正代码选项

地址为 0x2010，标定点的温度。

位地址	标识符	功能
15:0	TEMP_BD	标定点的绝对温度值。

TEMP\_BD[15:0]=30000 表示温度标定时时的绝对温度值的 300.00K，即等于  $300.0-273.15=26.85^{\circ}\text{C}$ 。

TEMP\_BD[15:0]=30315 表示温度标定时时的绝对温度值的 303.15K，即等于  $303.15-273.15=30.00^{\circ}\text{C}$ 。

地址为 0x2011，标定点温度对应测量到的 AD 值，参考电压选择 LDOS=00。

位地址	标识符	功能
-----	-----	----

15:0	TEMP_AD0	标定点温度对应测量到的 AD0 值。LDOS=00
------	----------	---------------------------

地址为 0x2012，标定点温度对应测量到的 AD 值，参考电压选择 LDOS=01。

位地址	标识符	功能
15:0	TEMP_AD1	标定点温度对应测量到的 AD1 值。LDOS=01

地址为 0x2013，标定点温度对应测量到的 AD 值，参考电压选择 LDOS=10。

位地址	标识符	功能
15:0	TEMP_AD2	标定点温度对应测量到的 AD2 值。LDOS=10

地址为 0x2014，标定点温度对应测量到的 AD 值，参考电压选择 LDOS=11。

位地址	标识符	功能
15:0	TEMP_AD3	标定点温度对应测量到的 AD3 值。LDOS=11

## 5 MCU 指令集

MCU 指令集

芯海科技CHIPSEA



指令	操作	指令周期	标志位
ADDLW k	$[W] \leftarrow [W] + k$	1	C,DC,Z
ADDPCW	$[PC] \leftarrow [PC] + 1 + [W]$	2	~
ADDWF f,d	$[Destination] \leftarrow [f] + [W]$	1	C,DC,Z
ADDWFC f,d	$[Destination] \leftarrow [f] + [W] + C$	1	C,DC,Z
ANDLW k	$[W] \leftarrow [W] \text{ AND } k$	1	Z
ANDWF f,d	$[Destination] \leftarrow [W] \text{ AND } [f]$	1	Z
BCF f,b	$[f<b>] \leftarrow 0$	1	~
BSF f,b	$[f<b>] \leftarrow 1$	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	$[f] \leftarrow 0$	1	Z
CLRWDT	Clear watch dog timer	1	~
COMF f,d	$[f] \leftarrow \text{NOT}([f])$	1	Z
DAW	Decimal Adjust W	1	C,DC
DECF f,d	$[Destination] \leftarrow [f] - 1$	1	Z
DECFSZ f,d	$[Destination] \leftarrow [f] - 1$ , jump if the result is zero	1/2	~
GOTO k	$PC \leftarrow k$	2	~
HALT	CPU Stop	1	~
INCF f,d	$[Destination] \leftarrow [f] + 1$	1	Z
INCFSZ f,d	$[Destination] \leftarrow [f] + 1$ , jump if the result is zero	1/2	~
IORLW k	$[W] \leftarrow [W] \text{ OR } k$	1	Z
IORWF f,d	$[Destination] \leftarrow [W] \text{ OR } [f]$	1	Z
MOVFW f	$[W] \leftarrow [f]$	1	~
MOVLW k	$[W] \leftarrow k$	1	~
MOVP	Read table list	3	~
MOVWF f	$[f] \leftarrow [W]$	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	$[Destination<n+1>] \leftarrow [f<n>]$	1	C,Z
RRF f,d	$[Destination<n-1>] \leftarrow [f<n>]$	1	C,Z
SLEEP	STOP OSC	1	PD
SUBLW k	$[W] \leftarrow k - [W]$	1	C,DC,Z
SUBWF f,d	$[Destinnation] \leftarrow [f] - [W]$	1	C,DC,Z
SUBWFC f,d	$[Destinnation] \leftarrow [f] - [W] - 1 + C$	1	C,DC,Z
SWAPF f,d	swap f	1	~
TBLP k	$[EADRH, EADRL] \leftarrow \text{WORK}$	k+1	~
XORLW k	$[W] \leftarrow [W] \text{ XOR } k$	1	Z

XORWF f,d	[Destination] ← [W] XOR [f]	1	Z
-----------	-----------------------------	---	---

参数说明:

f:数据存储器地址(000H ~1FFH)

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

### MCU 指令集描述

1

<b>ADDLW</b>	加立即数到工作寄存器
指令格式	ADDLW K (0<=K<=FFH)
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

<b>ADDPCW</b>	将 W 的内容加到 PC 中
指令格式	ADDPCW
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
例子 1 ADDPCW	在指令执行之前: W=7FH , PC=0212H 指令执行之后: PC=0292H
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

<b>ADDWF</b>	加工作寄存器到 f
指令格式	ADDWF f,d 0<=f<=1FFH d=0,1
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

<b>ADDWFC</b>	将 W f 和进位位相加
指令格式	ADDWFC f, d 0<=f<=1FFH d=0,1
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 指令执行之后 C=0 f=50H W=4DH

## 5

<b>ANDLW</b>	工作寄存器与立即数相与
指令格式	ANDLW K 0<=K<=FFH
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

<b>ANDWF</b>	将工作寄存器和 f 的内容相与
指令格式	ANDWF f, d 0<=f<=1FFH d=0,1
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

<b>BCF</b>	清除 f 的某一位
指令格式	BCF f, b 0<=f<=1FFH 0<=b<=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 BCF FLAG 2	指令执行之前: FLAG=8DH 指令执行之后: FLAG=89H

## 8

<b>BSF</b>	F 的 b 位置 1
指令格式	BSF f, b 0<=f<=1FFH 0<=b<=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

<b>BTFSC</b>	如果 bit 测试为 0 则跳转
指令格式	BTFSC f, b 0<=f<=1FFH 0<=b<=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到, 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

<b>BTFSS</b>	如果 bit 测试为 1, 则跳转
指令格式	BTFSS f, b 0<=f<=1FFH 0<=b<=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到, 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 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

<b>CALL</b>	子程序调用
指令格式	CALL K 0<=K<=1FFFH
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

<b>CLRF</b>	清除 f
指令格式	CLRF f 0<=f<=1FFH
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

<b>CLRWDT</b>	清除看门狗定时器
指令格式	CLRWDT
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

<b>COMF</b>	f 取反
指令格式	COMF f, d 0<=f<=1FFH d=0,1
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

<b>DAW</b>	十进制调整 W 寄存器
指令格式	DAW
操作	十进制调整 W 寄存器
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=25; ADDLW 39 DAW	在 DAW 指令执行之前 W=25+39=64=5EH 在指令执行之后 W=64H

## 16

<b>DECF</b>	f 减 1
指令格式	DECF f, d 0<=f<=1FFH d=0,1
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 DECF f, 0	在指令执行之前 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

<b>DECFSZ</b>	f 减 1 如果为 0 则跳转
指令格式	DECFSZ f, d 0<=f<=1FFH d=0,1
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 DECF SZ FLAG, 1 OP1: OP2:	在指令执行之前 PC=address(Node) 在指令执行之后 (FLAG)=(FLAG)-1 If(FLAG)=0 PC=address(OP2) If(FLAG)!=0 PC=address(OP1)
--	--

18

<b>GOTO</b>	无条件跳转
指令格式	GOTO K 0<=K<=1FFFH
操作	PC<-K
标志位	无
描述	立即地址载入 PC
周期	2

19

<b>HALT</b>	停止 CPU 时钟
指令格式	HALT
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。
周期	1

20

<b>INCF</b>	f 加 1
指令格式	INCF f, d 0<=f<=1FFFH d=0,1
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<=1FFH d=0,1
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 (FLAG)=(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
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<=1FFH d=0,1
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

<b>MOVFW</b>	传送到工作寄存器
指令格式	MOVFW f 0<=f<=FFH
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

<b>MOVLW</b>	将立即数传送到工作寄存器中
指令格式	MOVLW K 0<=K<=FFH
操作	(W)<-K
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

<b>MOVP</b>	读查表区数据
指令格式	MOVP
操作	把 EPROM 数据读到 WORK 中
标志位	无
描述	把地址为 EADRH/EADRL 的查表区数据读到 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 34H 在指令执行之后 W=34H

## 27

<b>MOVWF</b>	将工作寄存器的值传送到 f 中
指令格式	MOVWF f 0<=f<=1FFH
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

<b>NOP</b>	无操作
指令格式	NOP
操作	无操作
标志位	无
描述	无操作
周期	1

29

<b>PUSH</b>	把 work 和 status 寄存器入栈保护
指令格式	PUSH
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 4 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

<b>POP</b>	把 work 和 status 寄存器出栈处理
指令格式	POP
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 4 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

<b>RETFIE</b>	从中断返回
指令格式	RETFIE
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

<b>RETLW</b>	返回，并将立即数送到工作寄存器中
指令格式	RETLW K 0<=K<=FFH
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

<b>RETURN</b>	从子程序返回
指令格式	RETURN
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

<b>RLF</b>	带进位左移
指令格式	RLF f, d 0<=f<=1FFH d=0,1
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

<b>RRF</b>	带进位右移
指令格式	RRF f, d 0<=f<=1FFH d=0,1
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

<b>SLEEP</b>	晶振停止
指令格式	SLEEP
操作	CPU 时钟停止
标志位	PD
描述	CPU 时钟停止。CPU 通过外部中断源重启
周期	1

## 37

<b>SUBLW</b>	立即数减工作寄存器的值
指令格式	SUBLW K 0<=K<=FFH
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

<b>SUBWF</b>	f 的值减工作寄存器的值
指令格式	SUBWF f, d 0<=f<=1FFH d=0,1
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 SUBWF f, 1	在指令执行之前 f=04H W=05H 在指令执行之后 f=FFH C=0 Z=0

## 39

<b>SUBWFC</b>	带借位的减法
指令格式	SUBWFC f, d 0<=f<=1FFH d=0,1
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

<b>SWAPF</b>	交换寄存器的值
指令格式	SWAPF f, d 0<=f<=1FFH d=0,1
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	将 MTP 的{EADRH,EADRL}的地址写入{WORK}中的数据
指令格式	TBLP k
操作	MTP({EADRH,EADRL})←({WORK})
标志位	无
描述	将 MTP 的{EADRH,EADRL}的地址写入{WORK}中的数据
周期	与工作时钟有关
例子 TBLP 100	在指令执行之前 MTP memory: 17H= FFH EADRH =01H EADRL = 17H WORK = 05H 在指令执行之后 MTP memory: 117H = 05H

## 42

XORLW	工作寄存器的值与立即数异或
指令格式	XORLW K 0≤K≤FFH
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≤1FFH d=0,1
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



## 6 电气特性

### 6.1 极限值

表 6-1 CSU18M68 极限值

参数	测试条件	最小值	典型	最大	单位
VDD 极限电压		-0.3	3.3	4.5	V
引脚输入电压		GND-0.3		VDD+0.3	V
环境工作温度		-40		85	°C
存储温度		-55		150	°C
焊接温度, 时间		-	220°C, 10 秒	-	/

### 6.2 直流特性

 表 6-2 CSU18M68 直流特性 (VDD = 3.3V, T<sub>A</sub> = 25°C, 如无其他说明则都是此条件)

参数	测试条件	最小值	典型	最大	单位
VDD 工作电压		2.4	3.3	3.6	V
halt 模式工作电流	25°C ; VDD=3.3V		0.5		mA
sleep 模式工作电流 (开 WDT)	25°C ; VDD=3.3V		2	3	uA
输入漏电流	25°C ; VDD=3.3V			1	uA
数字输入高电平 (VIH)	PT1x_VDD=0; I2C_VDD=0	0.7*VDD			V
数字输入低电平 (VIL)	PT1x_VDD=0; I2C_VDD=0			0.29*VDD	V
数字输入高电平 (VIH)	PT1x_VDD=1; I2C_VDD=1	1.35			V
数字输入低电平 (VIL)	PT1x_VDD=1; I2C_VDD=1			0.4	V
上拉电流 (PT10、PT11)	VDD=3.3V	90	130		uA
高电平输出电流 (IOH)	VOUT=0.9*VDD; VDD=3.3V	2.7			mA
低电平输出电流 (IOL)	VOUT=0.1*VDD; VDD=3.3V	4.5			mA
VS 稳压输出电流		5			mA
I2C 滤波脉宽	I2C_FLT=1	50			nS
VS_CAP 电压输出	VDD ≥ 2.75V	2.35	2.45	2.55	V
VS_CAP 电压输出	VDD ≥ 2.9V	2.5	2.6	2.7	V
VS_CAP 电压输出	VDD ≥ 3.1V	2.7	2.8	2.9	V
VS_CAP 电压输出	VDD ≥ 3.3V	2.9	3	3.1	V

### 6.3 16bit ADC 的特性

 表 6-3 Sigma Delta ADC 性能指标 (VDD= 3.3V, T<sub>A</sub> = 25°C, 如无其他说明则都是此条件)

参数	测试条件	最小值	典型	最大	单位
----	------	-----	----	----	----

共模输入电压范围	VREF=VS0 引脚输入输出电压	0	VREF/2	VREF	V
满幅输入电压		-VREF/PGA		VREF/PGA	V
输入电容 (PGA=1)			0.4		pF
输入电容 (PGA=5)			2		pF
输入电容 (PGA=10)			4		pF
输入电容 (PGA=20)			4		pF
分辨率	带符号位		16		bit
等效输入噪声 (rms, PGA=1)	ADM=00		30		uV
等效输入噪声 (rms, PGA=5)	ADM=00		8		uV
等效输入噪声 (rms, PGA=10)	ADM=00		6.7		uV
等效输入噪声 (rms, PGA=20)	ADM=00		5.1		uV
等效输入噪声 (rms, PGA=1)	ADM=01		21		uV
等效输入噪声 (rms, PGA=5)	ADM=01		5.7		uV
等效输入噪声 (rms, PGA=10)	ADM=01		4		uV
等效输入噪声 (rms, PGA=20)	ADM=01		3.6		uV
等效输入噪声 (rms, PGA=1)	ADM=10		14.7		uV
等效输入噪声 (rms, PGA=5)	ADM=10		4		uV
等效输入噪声 (rms, PGA=10)	ADM=10		2.8		uV
等效输入噪声 (rms, PGA=20)	ADM=10		2.6		uV
等效输入噪声 (rms, PGA=1)	ADM=11		10		uV
等效输入噪声 (rms, PGA=5)	ADM=11		2.9		uV
等效输入噪声 (rms, PGA=10)	ADM=11		2		uV
等效输入噪声 (rms, PGA=20)	ADM=11		1.8		uV
积分线性度		-0.01		0.01	% of 0.9*FS
失调误差		-1.5/PGA		1.5/PGA	mV
失调误差漂移		-2/PGA		2/PGA	uV/°C
增益误差		-1		1	%
VS_CAP 电压输出	VDD ≥ 2.75V LDOS=00	2.35	2.45	2.55	V
VS_CAP 电压输出	VDD ≥ 2.9V LDOS=01	2.5	2.6	2.7	V
VS_CAP 电压输出	VDD ≥ 3.1V LDOS=10	2.7	2.8	2.9	V
VS_CAP 电压输出	VDD ≥ 3.3V LDOS=11	2.9	3	3.1	V
参考电压抑制比		30			dB
温度传感器线性		-2	±1	2	°C
温度传感器每摄氏度的电压变化值			0.828		mV/°C

## 6.4 MTP 的特性

表 6-4 MTP 特性

参数	测试条件	最小值	典型	最大	单位
工作电压范围		2.5		3.6	V
工作温度范围		-40		85	°C
Endurance		1K			Cycle
Retention	@85°C	10			Year
烧写时间	By byte	1.5			mS

## 7 封装信息

### 7.1 WLCSP16 封装尺寸图

WLCSP16-pin(min ptch:350um+thickness:340±25um)

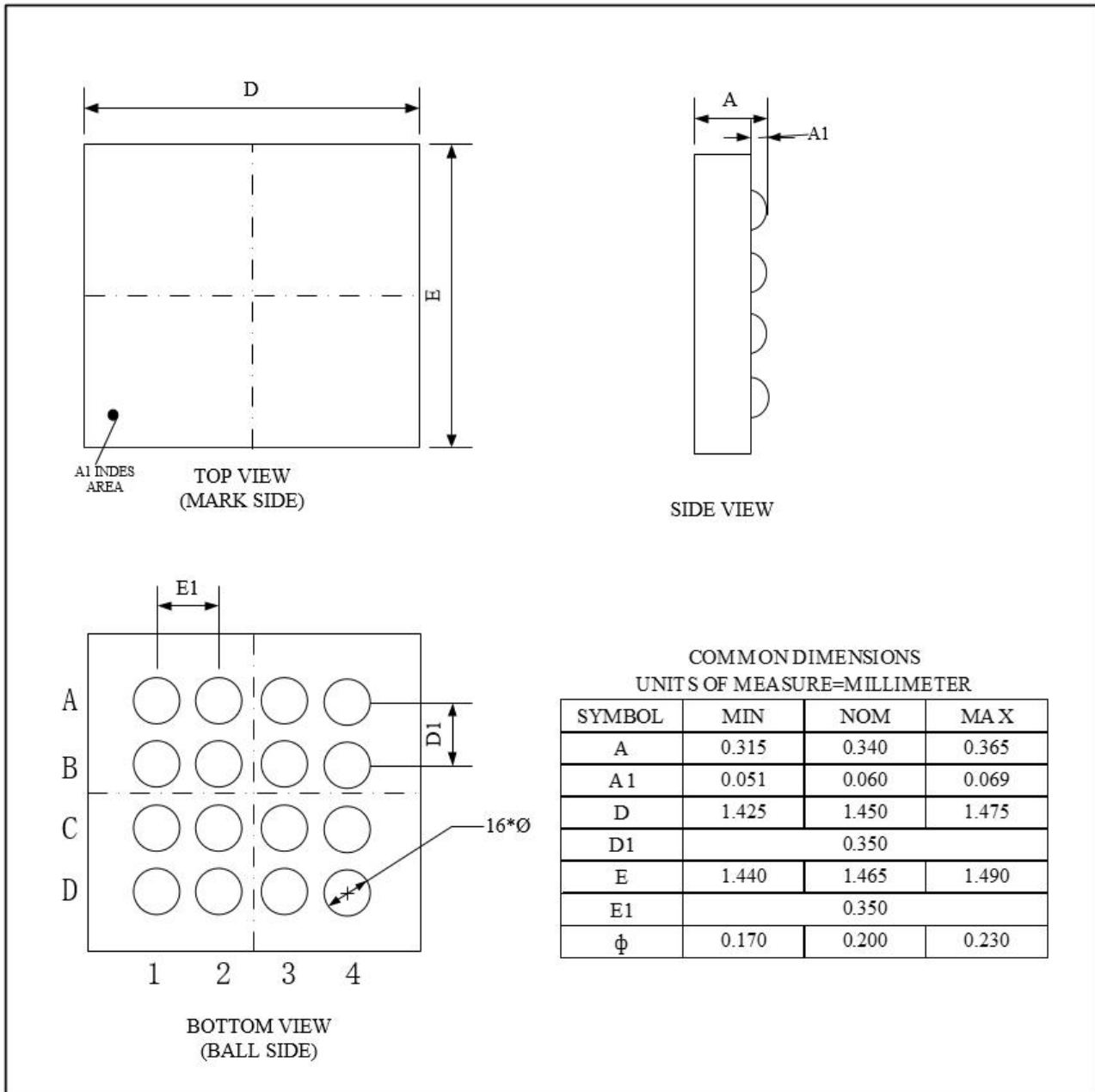


图 7-1 WLCSP16 package outline

### 7.2 QFN16 封装尺寸图

QFN16(03x03x0.75-0.50)

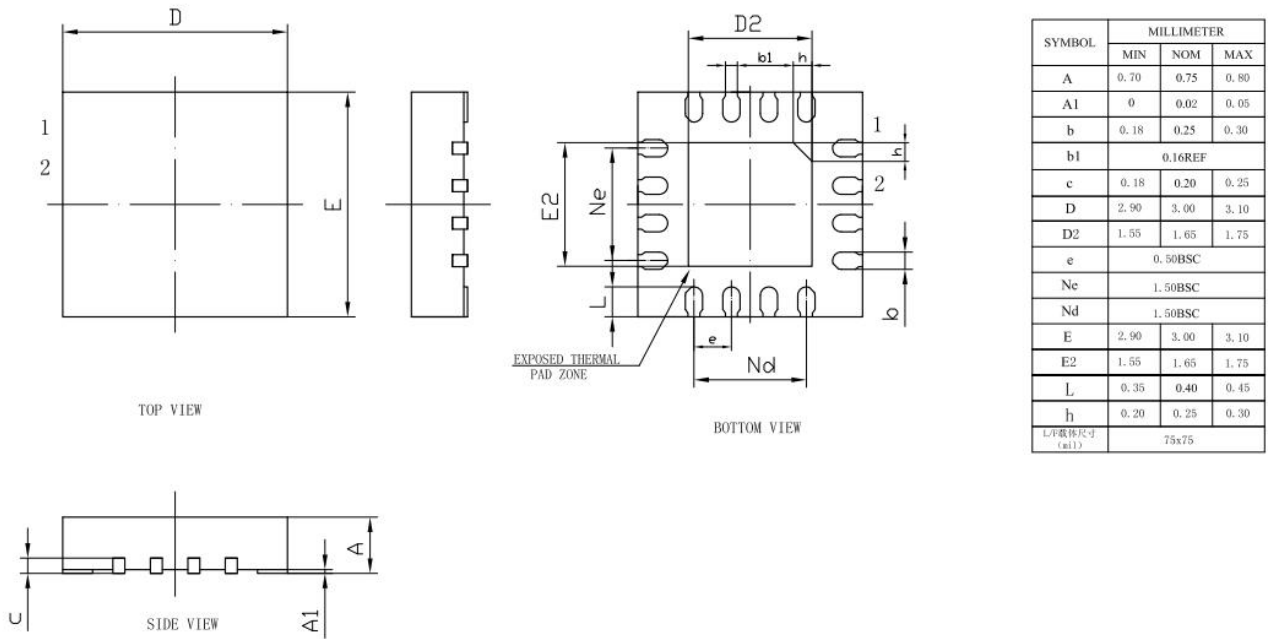


图 7-2 QFN16 package outline

## 8 产品印字

### WLCSP16:



图 8-1 WLCSP16 封装打标

#### 印字描述

- 1、正面 A1 处为 PIN1 脚标识，距边为 120um，直径  $\Phi$  150um。
- 2、正面第一行（CHIPSEA）为公司名称。
- 3、正面第二行（18M68）为产品名称。
- 4、正面第三行（YYWWXXX）为追溯信息，其中 YY 取自公历年号后两位，WW 取自本年度日历周数（不足两位时左端补 0），XXX 为固定码，订单给出
- 5、正面第四行（XXXXXXZ），前六位为坐标（列：横坐标 XXX:010，纵坐标 XXX:060）；Z 为片号（1-25#对应字母 A-Y）
- 6、所有字高 0.18mm，行距 0.12mm，行距 0.06mm，字间距 0.05mm。
- 7、字体为“Arial”，字体笔画加粗。
- 8、打印方式为激光正印。

## QFN16:



图 8-2 QFN16 封装打标

### 丝印描述:

- 1、正面 A 处为 PIN1 脚标识，直径  $\Phi 0.4\text{mm}$
- 2、正面第一行（CHIPSEA）为公司名称
- 3、正面第二行（18M68-QN）为产品名称
- 4、正面第三行（YYWWXXX）为追溯信息，其中 YY 取自公历年号后两位，WW 取自本年度日历周数（不足两位时左端补 0），XXX 订单给出
- 5、字体为“Arial”，字体笔画加粗
- 6、打印方式为激光正印

## 9 包装信息

WLCSP16 编带包装-Tape&Reel

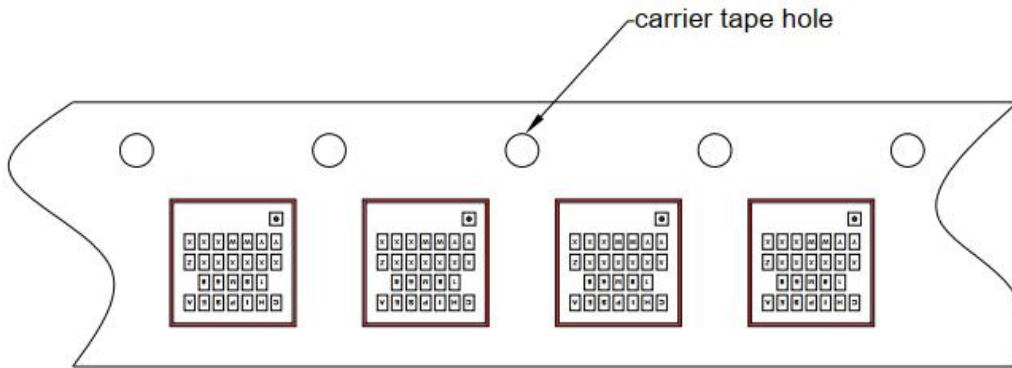


图 9-1 编带包装

QFN16 : TRAY

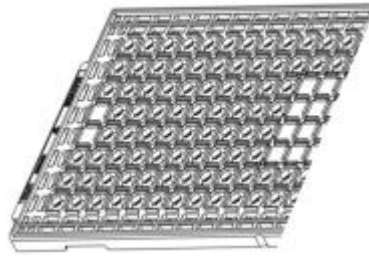


图 9-2 托盘包



## 10 专利保护

芯海科技一直致力于【信号测量领域】的技术创新与知识产权保护。自 2015 年，芯海科技开始在【压感领域】进行技术研发与专利布局。截至 2020.6 月，芯海科技在压感领域累计布局超过 100 件的专利，其中已授权专利超过 50 件，保护对象涉及到芯片、电路结构、传感器模组、传感器结构、终端产品、软件应用等。在【压感领域】持续性的研发投入及技术创新，确保了芯海科技在【压感领域】的技术领先性，同时通过全方位的知识产权布局，着力打造专利护城河，确保客户及合作伙伴的产品不存在知识产权风险。

### 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。

版权归 © 2022 芯海科技（深圳）股份有限公司，保留所有权利。



芯海科技  
CHIPSEA

股票代码:688595