第 1 章 GDMC191XXXX MCU 整体介绍

1.1. 特性简介

- 工作电压：2.7～5.5V
- 工作温度：-40℃~+85℃
- 储存温度：-40℃~+125℃
- 15K FLASH
- 1K 类 EEPROM
- 256Bytes(data)+512(xdata)SRAM
- 内置 RC 振荡电路(1MHz)
- 高速 8051，基于标准 8051 指令流水线结构
- 工作频率（软件配置）：12M、6M、4M 系统时钟，一个时钟/机器周期，速度比普通 8051 快 9~12 倍
- 电容按键，均可复用为 GPIO
- GPIO 支持内置上拉电阻
- 大灌电流口(PB0~PB7)
- 3 个 16 位定时器，具有溢出中断；Timer2 时钟源为内部 32KHz
- INT0~2 外部中断(上升沿、下降沿、双沿)
- 支持空闲模式唤醒，唤醒时间 0.1s ~ 2.304s
- 12bit 10 位精度 ADC 检测
- 支持多键低功耗模式下任意键快速唤醒(<20uA @100ms)
- IIC 硬件从机通信，支持标准模式 100/400KHz
- UART 通信，可配置波特率(2400、4800、9600、19200、57600、115200)，支持 IO 映射
- 两级中断优先级可选
- 中断源
  - ——电容按键中断
  - ——ADC 中断
  - ——外部中断
  - ——Timer0，Timer1，Timer2
  - ——串口中断
  - ——IIC 通信中断
  - ——看门狗（WDT）中断
  - ——LVDT 中断
- 支持掉电复位，掉电电压 2.1V
- 低电压检测 2.4V/3.0V/3.6V/4.2V 可选
- 看门狗定时器，溢出时间 18ms 到 2.304s
- 深度休眠，功耗 6uA@5V
- 各按键的灵敏度可独立设置，配置灵活
- 带 JTAG 调试仿真接口
- 封装型号： SOP20/SOP28

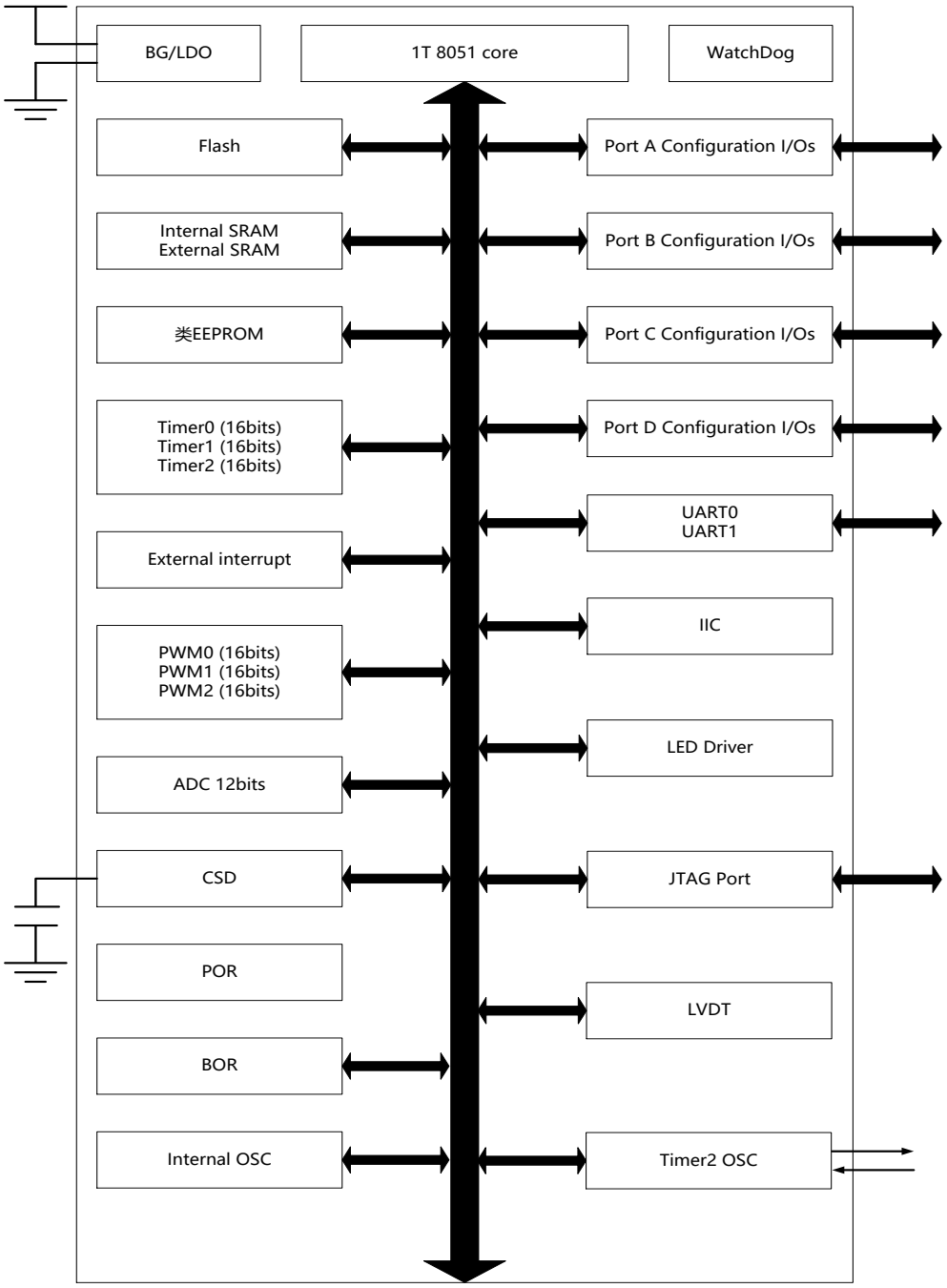
1.2.整体概述

GDMC191XXXX 采用高速 8051 内核，1T 指令周期，相比于标准的 8051(12T)指令周期，具有更快的运行速度，同时兼容标准 8051 指令。

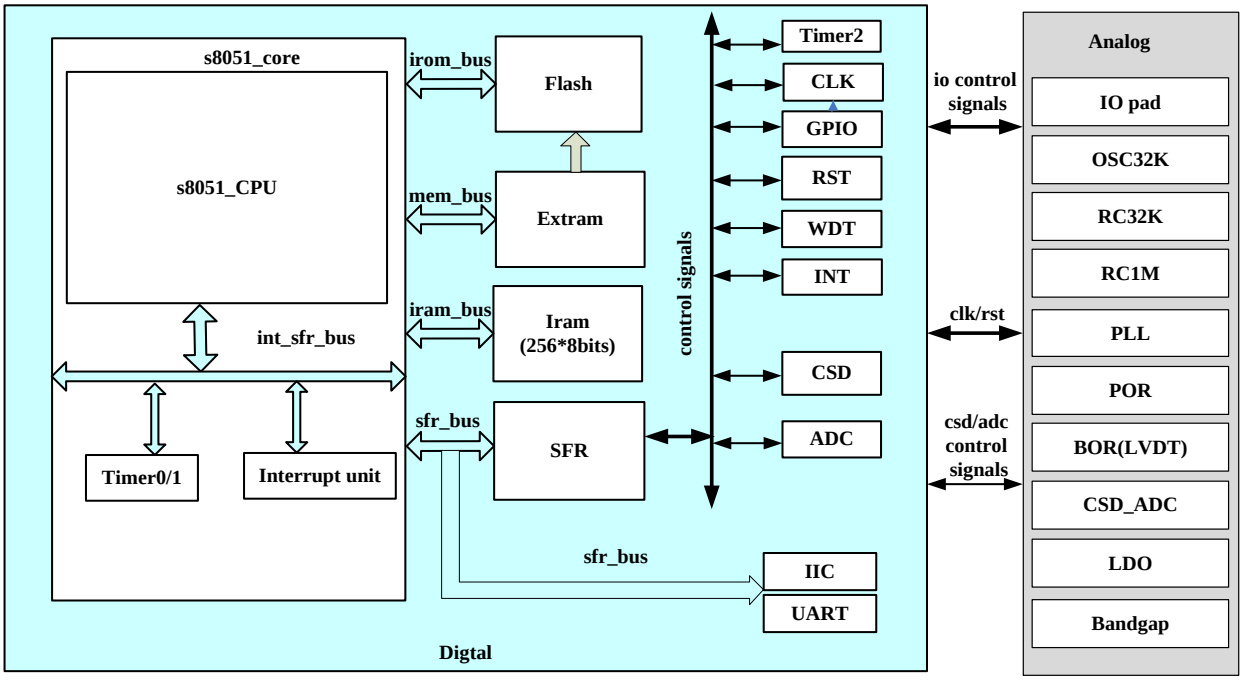
GDMC191XXXX 包含外设有看门狗、按键检测、IIC、UART、低电压检测、掉电复位、、Timer0、Timer1、Timer2、12bit 逐次逼近 ADC、低功耗模式等。

GDMC191XXXX 集成的电容检测通道，它可以用来检测近距离感应或者触摸。其内置 MCU，可灵活配置；通过配置可实现按键、滚轮、滑条等多种应用。按键都能独立的运行，并且每个按键都能通过对相应的功能寄存器来调节灵敏度。

1.3. 系统框图

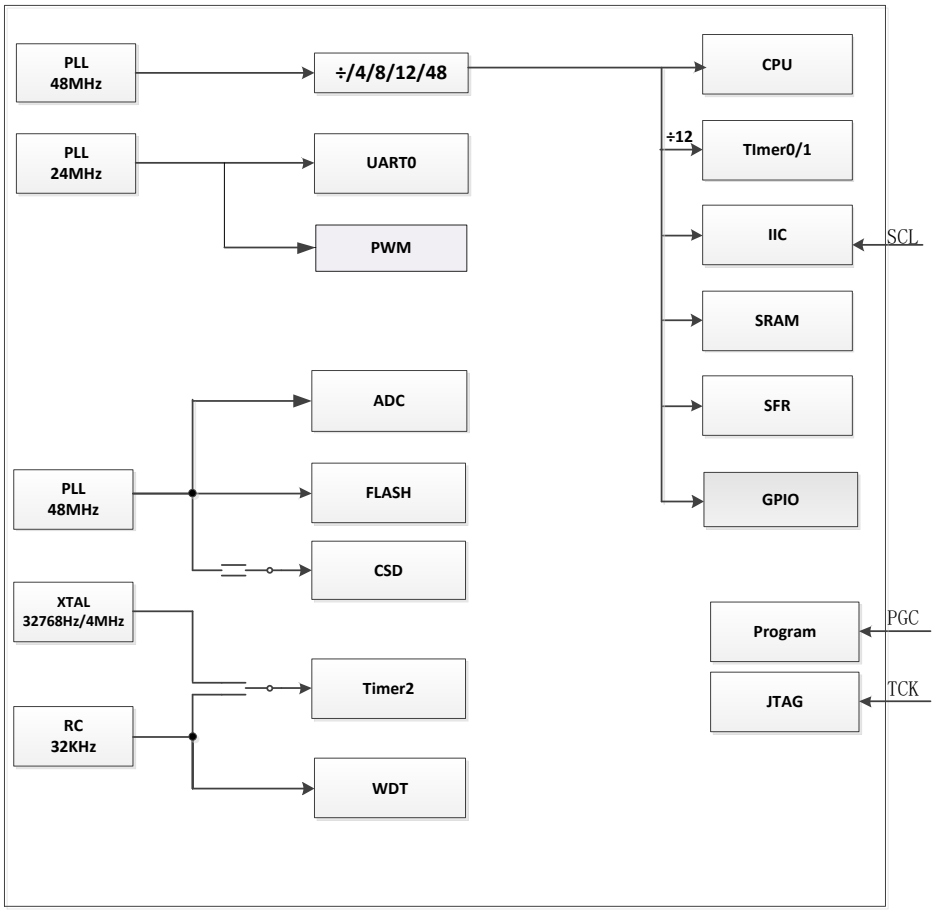

BG/LDO
1T 8051 core
WatchDog
Flash
Internal SRAM
External SRAM
类EEPROM
Timer0 (16bits)
Timer1 (16bits)
Timer2 (16bits)
External interrupt
PWM0 (16bits)
PWM1 (16bits)
PWM2 (16bits)
ADC 12bits
CSD
POR
BOR
Internal OSC
Port A Configuration I/Os
Port B Configuration I/Os
Port C Configuration I/Os
Port D Configuration I/Os
UART0
UART1
IIC
LED Driver
JTAG Port
LVDT
Timer2 OSC

系统框图


s8051_core
s8051_CPU
int_sfr_bus
Timer0/1
Interrupt unit
irom_bus
Flash
mem_bus
Extram
iram_bus
Iram
(256*8bits)
sfr_bus
SFR
sfr_bus
IIC
UART
Digtal
control signals
Timer2
CLK
GPIO
RST
WDT
INT
CSD
ADC
io control
signals
clk/rst
csd/adc
control
signals
Analog
IO pad
OSC32K
RC32K
RC1M
PLL
POR
BOR(LVDT)
CSD_ADC
LDO
Bandgap

系统总线架构图

1.4. 时钟框图



时钟方框图

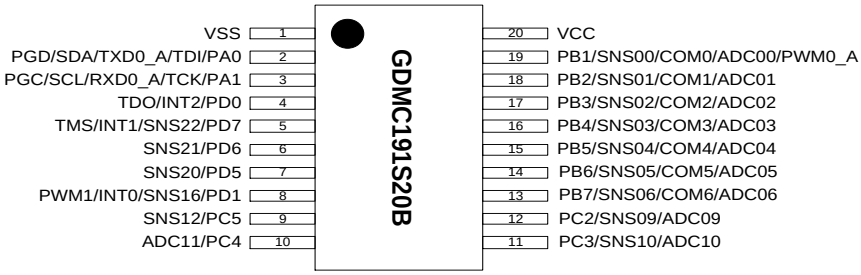
1.5. 选型列表

型号	GDMC191S20B	GDMC191S20C	GDMC191S28C	GDMC191M28C
工作电压(V)	2.7~5.5	2.7~5.5	2.7~5.5	2.7~5.5
内核	1T 8051	1T 8051	1T 8051	1T 8051
工作频率	12M	12M	12M	12M
FLASH	15K	15K	15K	15K
SRAM	256+512	256+512	256+512	256+512
类 EEPROM	1K	1K	1K	1K
GPIO	18	14	18	26
ADC	10	10	12	14
Timer	3	3	3	3
PWM	2	2	2	2
INT	3	2	3	2
IIC	1	1	1	1
UART	1	1	1	1
KEY	14	14	16	15
封装	SOP20	SOP20	SOP28	SOP28

选型列表

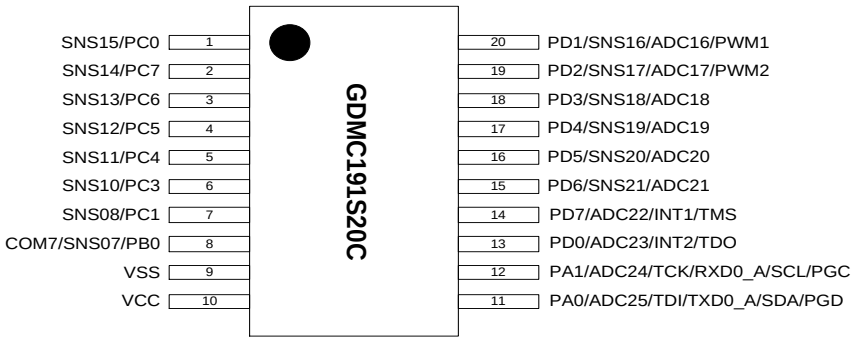
1.6.引脚配置

1.6.1. GDMC191S20B



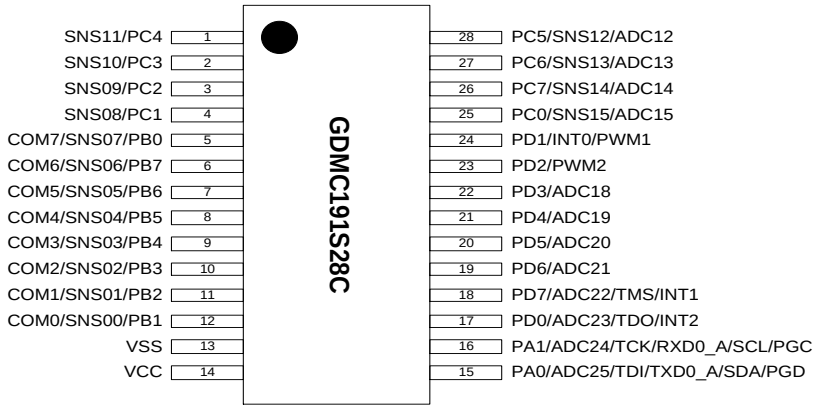
GDMC191S20B SOP20 封装引脚图

1.6.2. GDMC191S20C



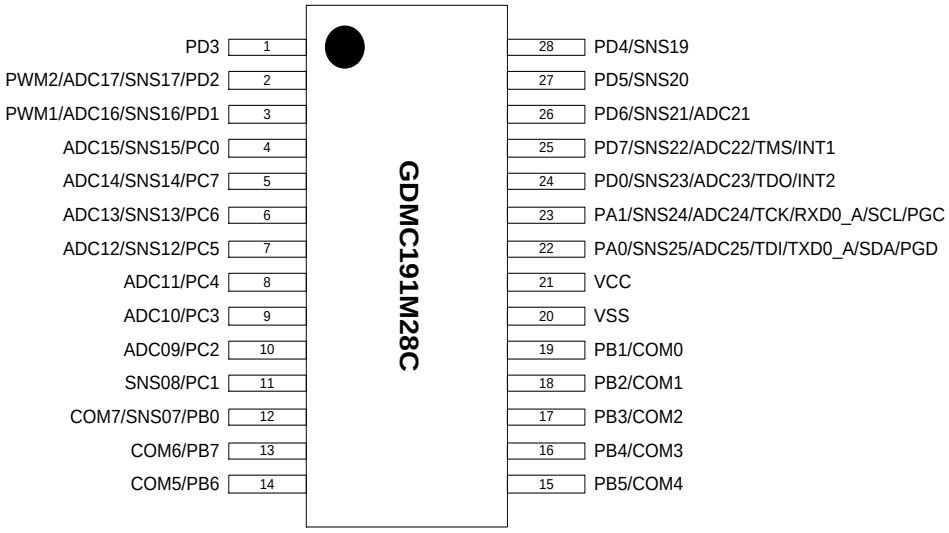
GDMC191S20C SOP20 封装引脚图

1.6.3. GDMC191S28C



GDMC191S28C SOP28 封装引脚图

1.6.4. GDMC191M28C



GDMC191M28C SOP28 封装引脚图

1.7. 引脚说明

1.7.1. GDMC191S20B 引脚说明

GDMC191S20B	功能描述
1	默认功能：电源地 <VSS>
2	默认功能：GPIO <PA0> 其它功能：TAG_TDI、串口发送、IIC_SDA、烧录口_PGD
3	默认功能：GPIO <PA1> 其它功能：JTAG_TCK、串口接收、IIC_SCL、烧录口_PGC
4	默认功能：GPIO <PD0> 其它功能：INT、JTAG_TDO
5	默认功能：GPIO <PD7> 其它功能：INT、JTAG_TMS
6	默认功能：GPIO <PD6> 其它功能：触摸按键<SNS>
7	默认功能：GPIO <PD5> 其它功能：触摸按键<SNS>
8	默认功能：GPIO <PD1> 其它功能：触摸按键<SNS>、外部中断、PWM 功能
9	默认功能：GPIO <PC5> 其它功能：触摸按键<SNS>
10	默认功能：GPIO <PC4> 其它功能：ADC 通道
11	默认功能：GPIO <PC3> 其它功能：触摸按键<SNS>、ADC 通道
12	默认功能：GPIO <PC2> 其它功能：触摸按键<SNS>、ADC 通道
13	默认功能：GPIO <PB7> 其它功能：触摸按键<SNS>、大灌电流口、ADC 通道
14	默认功能：GPIO <PB6> 其它功能：触摸按键<SNS>、大灌电流口、ADC 通道
15	默认功能：GPIO <PB5> 其它功能：触摸按键<SNS>、大灌电流口、ADC 通道
16	默认功能：GPIO <PB4> 其它功能：触摸按键<SNS>、大灌电流口、ADC 通道
17	默认功能：GPIO <PB3>

	其它功能：触摸按键<SNS>、大灌电流口、ADC 通道
18	默认功能：GPIO <PB2> 其它功能：触摸按键<SNS>、大灌电流口、ADC 通道
19	默认功能：GPIO <PB1> 其它功能：触摸按键<SNS>、大灌电流口、ADC 通道、PWM 功能
20	默认功能：电源 <VCC>

1.7.2. GDMC191S20C 引脚说明

GDMC191S20C	功能描述
1	默认功能：GPIO <PC0> 其它功能：触摸按键<SNS>
2	默认功能：GPIO <PC7> 其它功能：触摸按键<SNS>
3	默认功能：GPIO <PC6> 其它功能：触摸按键<SNS>
4	默认功能：GPIO <PC5> 其它功能：触摸按键<SNS>
5	默认功能：GPIO <PC4> 其它功能：触摸按键<SNS>
6	默认功能：GPIO <PC3> 其它功能：触摸按键<SNS>
7	默认功能：GPIO <PC1> 其它功能：触摸按键<SNS>
8	默认功能：GPIO <PB0> 其它功能：触摸按键<SNS>、大灌电流口
9	默认功能：电源地 <VSS>
10	默认功能：电源 <VCC>
11	默认功能：GPIO <PA0> 其它功能：ADC 通道、JTAG_TDI、串口发送、IIC_SDA、烧录口_PGD
12	默认功能：GPIO <PA1> 其它功能：ADC 通道、JTAG_TCK、串口接收、IIC_SCL、烧录口_PGC
13	默认功能：GPIO <PD0> 其它功能：ADC 通道、JTAG_TDO
14	默认功能：GPIO <PD7> 其它功能：ADC 通道、JTAG_TMS

15	默认功能：GPIO <PD6> 其它功能：触摸按键<SNS>、ADC 通道
16	默认功能：GPIO <PD5> 其它功能：触摸按键<SNS>、ADC 通道
17	默认功能：GPIO <PD4> 其它功能：触摸按键<SNS>、ADC 通道
18	默认功能：GPIO <PD3> 其它功能：触摸按键<SNS>、ADC 通道
19	默认功能：GPIO <PD2>
20	默认功能：GPIO <PD1> 其它功能：触摸按键<SNS>、ADC 通道

1.7.3. GDMC191S28C 引脚说明

GDMC191S28C	功能描述
1	默认功能：GPIO <PC4> 其它功能：触摸按键<SNS>
2	默认功能：GPIO <PC3> 其它功能：触摸按键<SNS>
3	默认功能：GPIO <PC2> 其它功能：触摸按键<SNS>
4	默认功能：GPIO <PC1> 其它功能：触摸按键<SNS>
5	默认功能：GPIO <PB0> 其它功能：触摸按键<SNS>、大灌电流口
6	默认功能：GPIO <PB7> 其它功能：触摸按键<SNS>、大灌电流口
7	默认功能：GPIO <PB6> 其它功能：触摸按键<SNS>、大灌电流口
8	默认功能：GPIO <PB5> 其它功能：触摸按键<SNS>、大灌电流口
9	默认功能：GPIO <PB4> 其它功能：触摸按键<SNS>、大灌电流口
10	默认功能：GPIO <PB3> 其它功能：触摸按键<SNS>、大灌电流口
11	默认功能：GPIO <PB2>

	其它功能：触摸按键<SNS>、大灌电流口
12	默认功能：GPIO <PB1> 其它功能：触摸按键<SNS>、大灌电流口
13	默认功能：电源地 <VSS>
14	默认功能：电源 <VCC>
15	默认功能：GPIO <PA0> 其它功能：ADC 通道、JTAG_TDI、串口发送、IIC_SDA、烧录口_PGD
16	默认功能：GPIO <PA1> 其它功能：ADC 通道、JTAG_TCK、串口接收、IIC_SCL、烧录口_PGC
17	默认功能：GPIO <PD0> 其它功能：ADC 通道、JTAG_TDO
18	默认功能：GPIO <PD7> 其它功能：ADC 通道、JTAG_TMS
19	默认功能：GPIO <PD6> 其它功能：ADC 通道
20	默认功能：GPIO <PD5> 其它功能：ADC 通道
21	默认功能：GPIO <PD4> 其它功能：ADC 通道
22	默认功能：GPIO <PD3> 其它功能：ADC 通道
23	默认功能：GPIO <PD2>
24	默认功能：GPIO <PD1> 其它功能：INT
25	默认功能：GPIO <PC0> 其它功能：触摸按键<SNS>、ADC 通道
26	默认功能：GPIO <PC7> 其它功能：触摸按键<SNS>、ADC 通道
27	默认功能：GPIO <PC6> 其它功能：触摸按键<SNS>、ADC 通道
28	默认功能：GPIO <PC5> 其它功能：触摸按键<SNS>、ADC 通道

1.7.4. GDMC191M28C 引脚说明

GDMC191M28C	功能描述
1	默认功能：GPIO <PD3>
2	默认功能：GPIO <PD2> 其它功能：触摸按键<SNS> 、ADC 通道
3	默认功能：GPIO <PD1> 其它功能：触摸按键<SNS> 、ADC 通道
4	默认功能：GPIO <PC0> 其它功能：触摸按键<SNS> 、ADC 通道
5	默认功能：GPIO <PC7> 其它功能：触摸按键<SNS> 、ADC 通道
6	默认功能：GPIO <PC6> 其它功能：触摸按键<SNS> 、ADC 通道
7	默认功能：GPIO <PC5> 其它功能：触摸按键<SNS> 、ADC 通道
8	默认功能：GPIO <PC4> 其它功能：ADC 通道
9	默认功能：GPIO <PC3> 其它功能：ADC 通道
10	默认功能：GPIO <PC2> 其它功能：ADC 通道
11	默认功能：GPIO <PC1> 其它功能：触摸按键<SNS>
12	默认功能：GPIO <PC0> 其它功能：触摸按键<SNS>
13	默认功能：GPIO <PB7> 其它功能：触摸按键<SNS>、大灌电流口
14	默认功能：GPIO <PB6> 其它功能：触摸按键<SNS>、大灌电流口
15	默认功能：GPIO <PB5> 其它功能：大灌电流口
16	默认功能：GPIO <PB4> 其它功能：大灌电流口
17	默认功能：GPIO <PB3>

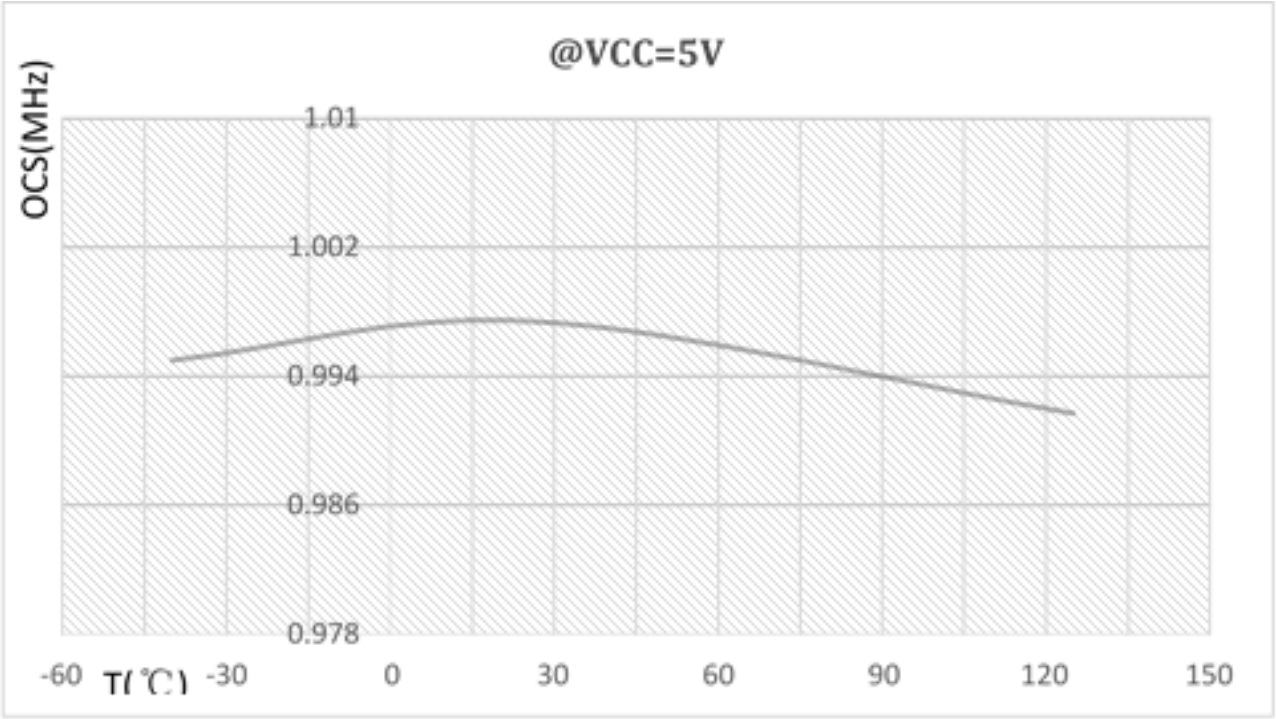
	其它功能：大灌电流口
18	默认功能：GPIO <PB2> 其它功能：大灌电流口
19	默认功能：GPIO <PB1> 其它功能：大灌电流口
20	默认功能：电源地 <VSS>
21	默认功能：电源 <VCC>
22	默认功能：GPIO <PA0> 其它功能：触摸按键<SNS>、ADC 通道、JTAG_TDI、串口发送、IIC_SDA、烧录口_PGD
23	默认功能：GPIO <PA1> 其它功能：触摸按键<SNS>、ADC 通道、JTAG_TCK、串口接收、IIC_SCL、烧录口_PGC
24	默认功能：GPIO <PD0> 其它功能：触摸按键<SNS>、ADC 通道、JTAG_TDO
25	默认功能：GPIO <PD7> 其它功能：触摸按键<SNS>、ADC 通道、JTAG_TMS
26	默认功能：GPIO <PD6> 其它功能：触摸按键<SNS>、ADC 通道
27	默认功能：GPIO <PD5> 其它功能：触摸按键<SNS>
28	默认功能：GPIO <PD4> 其它功能：触摸按键<SNS>

第 2 章 电气特性

2.1.AC 特性

参数	符号	条件	OSC 时钟	单位
基频	RC1M	正常工作 27℃	1M±1%	Hz
		环境温度-40℃~85℃	1M±3%	
WDT 时钟	RC32K	环境温度-40℃~125℃	32K±30%	Hz

AC 特性参数表



OSC 温度特性趋势曲线

2.2.DC 特性

除特殊说明外，典型值为在 27℃条件下的测量值。

参数	符号	条件	最小值	典型值	最大值	单位
工作电压	VCC		2.7	-	5.5	V
工作模式	Active	@5V,系统时钟12M	-	3.6	4.5	mA
		@5V,系统时钟6M	-	2.6	3.5	mA
		@5V,系统时钟4M	-	2.2	3	mA
		@3.3V,系统时钟12M	-	3.5	4.4	mA
		@3.3V,系统时钟6M	-	2.5	3.4	mA
		@3.3V,系统时钟4M	-	2.1	2.8	mA
	Idle	@5V,WDT_CTRL=7，唤醒，1%工作时间，IO输出低，关闭其它所有功能	-	12	20	μA
		@5V,Timer2 外部晶振2S唤醒，1%工作时间，IO输出低，关闭其它所有功能	-	12	20	μA
		@3.3V,WDT_CTRL=7，唤醒，1%工作时间，IO输出低，关闭其它所有功能	-	11	19	μA
		@3.3V,Timer2 外部晶振2S唤醒，1%工作时间，IO输出低，关闭其它所有功能	-	11	19	μA
		@5V,CSD并联中断唤醒，1%工作时间，IO输出低，关闭其它所有功能	-	15	25	μA
		@3.3V,CSD并联中断唤醒，1%工作时间，IO输出低，关闭其它所有功能	-	13	23	μA
	Sleep	@5V PCON = 0x01,BOR关闭，IO输出低，关闭其它所有功能	-	5.8	8	μA
		@3.3V PCON = 0x01,BOR关闭，IO输出低，关闭其它所有功能	-	5.1	7	μA
输入低电压	V_{IL}	VCC=3.3~5.5V	-	-	0.3*VCC	V
输入高电压	V_{IH}	VCC=3.3~5.5V	0.7*VCC	-	-	V
INT0/1/2输入低电压	V_{INTL}	VCC=3.3~5.5V	-	-	0.3*VCC	V
INT0/1/2输入高电压	V_{INTH}	VCC=3.3~5.5V	0.7*VCC	-	-	V
输出低电压	V_{OL}	I_{OL}=4mA@VCC=3.3V，	-	-	0.1*VCC	V

		I_{OL}=10mA@VCC=5V				
输出高电压	V_{OH}	I_{OH}=4mA@VCC=3.3V， I_{OH}=10mA@VCC=5V	0.9VCC	-	-	V
IO灌电流	IOL	V_{OL} =0.1VCC，@VCC=5V	-	40	-	mA
IO源电流	IOH	V_{OH} =0.9VCC，@VCC=5V	-	20	-	mA
PB0~PB7大灌电流	I_{com}	V_{OL} =0.1VCC，@VCC=5V	-	80	-	mA
输入漏电流	I_{IH}&&I_{IL}	VCC=5V	-	1	5	μA
IO内部上拉	Pull_up Res	VCC=5V	-	4.7	-	K

DC 特性参数表

2.3. ADC 特性

除特殊说明外，典型值为在 27℃条件下的测量值。

参 数 名 称	符 号	测 试 条 件	最小	典型	最大	单 位
工作电压	VDD	-	VLVR	-	5.5	V
ADC 输入电压范围	VADCIN	-	0	-	VDD	V
分辨率	ADCRESO	-	12			bit
ADC 采样时间	TAD	(ADC_SPT+1)*4*Tadc_clk	0.5	-	-	us
输入通道	-	-			14	通道
精度	-	-	10			Bit
无丢码	-	-	10			Bits
ADC 检测时间	TCON	TAD=TADC_SPT+TW1+TW2	2.25	-	-	us
积分线性误差	EINL	-	-	±2	±3	LSB
微分线性误差	EDNL	-	-	±1	±2	LSB

ADC 特性参数表

2.4.极限参数

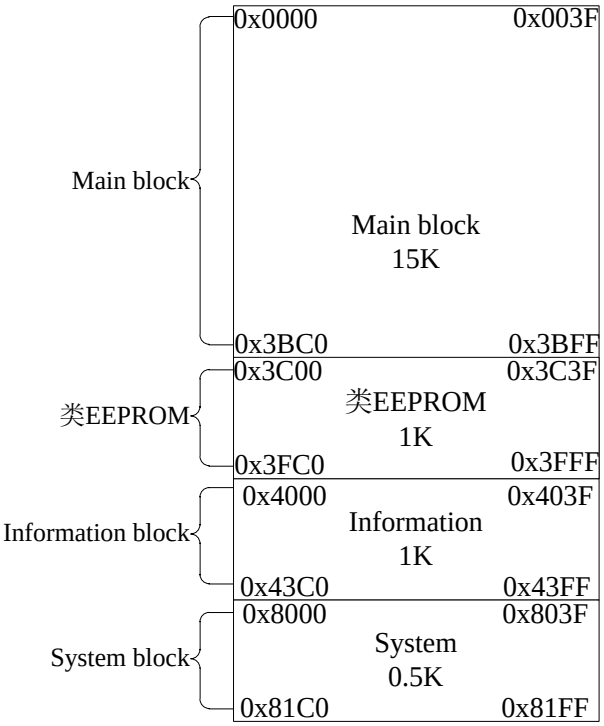
参数	符号	最小值	典型值	最大值	单位
非工作状态储存温度	Tstg	-40	-	125	℃
工作温度	Totg	-40	-	85	℃
I/O 输入电压	Vin	VSS-0.5	-	VCC+0.5	V
I/O 锁存电流	LU	200	-	-	mA
端口静电放电电压	ESD(HBM)	4000	-	-	V
工作时供电电压	VCC	VSS+2.7	-	VSS+5.5	V

极限特性参数表

注：超过极限参数所规定的范围将对芯片造成损害，无法预期芯片在上述标示范围外的工作状态，而且若长期在标示范围外的条件下工作，可能影响芯片的可靠性。

第 3 章 存储器和 SFR

3.1. Flash 存储器



Flash 程序存储器地址分配结构图

说明：

1. 主程序空间的大小最大为 15KBytes，擦写次数至少 10000 次，可分为 15 页、15*16 行；
2. 类 EEPROM 空间大小为 1024Bytes，擦写次数至少 10000 次,1024 Bytes 为一页、页擦除，字节写；
3. Information 的大小为 1024Bytes，可分为 16 行，其中包含出厂信息，不可更改；
4. System 的大小为 512Bytes，可分为 8 行。

3.2. RAM 存储器

内部共有 256 Bytes，地址为 00H~FFH，其中包括工作寄存器组、位寻址区、缓冲以及 SFR，其中缓冲区包含了堆栈区。

内部低 128 字节：00H~7FH 共有 128 Bytes，可通过立即寻址方式或间接寻址方式来读取与写数据。

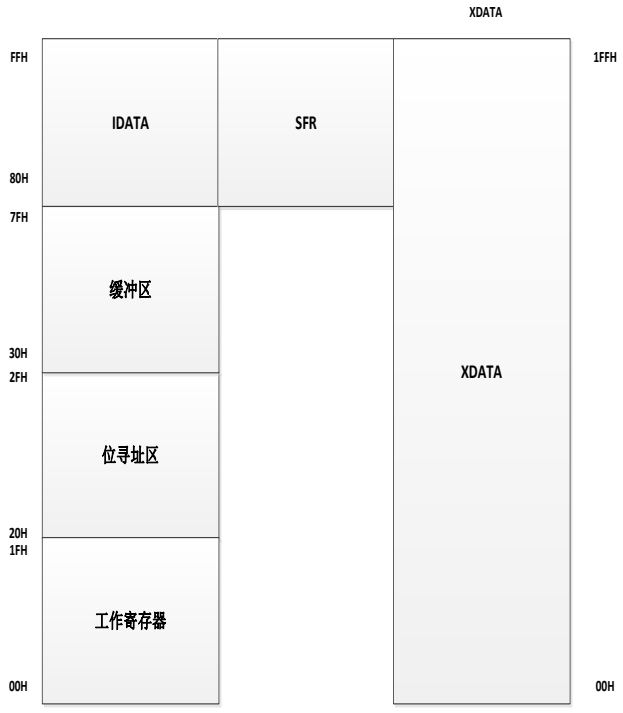
内部高 128 字节：80H~FFH 共有 128 Bytes，只能通过工作寄存器间接寻址方式来读取与写数据。

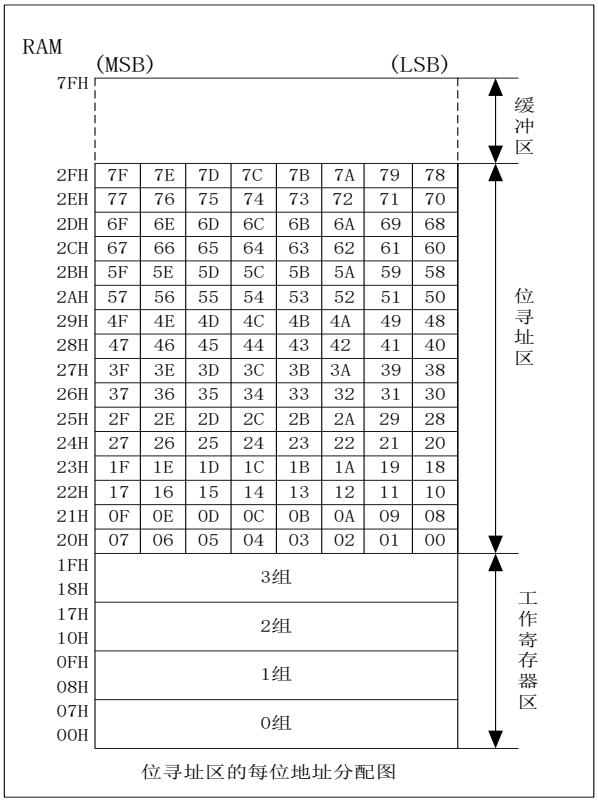
特殊功能寄存器 SFR：地址为 80H~FFH，只能通过直接寻址方式来读取与写数据。

xdata 共有 512 Bytes，地址为 0000H~01FFH，该区域用户可以完全使用。通过数据指针或者工作寄存器寻址方式来读取与写数据。

在编写程序时注意预留堆栈空间，避免堆栈溢出导致程序跑飞。在使用 C 语言编程时，堆栈首地址由程序自动分配，但是一定存放在 data 或者 idata 里。Keil 中可在 STARTUP.A51 中设置堆栈的首地址。

RAM 地址空间分配图：





位寻址区的每位地址分配图

下表中列出了 RAM 中三个模块的取值方式：

DATA	MOV A,direct
	MOV direct,A
	MOV direct,#data
	MOV direct1,direct2
	MOV Rn,direct
	MOV direct,Rn

IDATA	MOV A,@Ri
	MOV @Ri ,A
	MOV direct,@Ri
	MOV @Ri,direct
	MOV @Ri,#data

XDATA	MOVX @DPTR,A MOVX A,@DPTR

RAM 取值指令表

上表中，n 取值 0~7，i 取值 0~1。

第 4 章 寄存器汇总

4.1. SFR 寄存器总表

地址	名称	读写	复位值	说明
0x80	DATAB	RW	0xFF	PB 数据寄存器
0x81	SP	RW	0x07	堆栈指针寄存器
0x82	DPL	RW	0x00	数据指针寄存器 0 低 8 位
0x83	DPH	RW	0x00	数据指针寄存器 0 高 8 位
0x84	SYS_CLK_CFG	RO/RW	0x01	时钟控制寄存器
0x85	INT_PE_STAT	RW	0x00	WDT/Timer2 中断状态寄存器
0x86	INT_POBO_STAT	RW	0x00	LVDT 升压/LVDT 降压中断状态寄存器
0x87	PCON	RW	0x00	低功耗模式选择寄存器
0x88	TCON	RW	0x05	定时器控制寄存器
0x89	TMOD	RW	0x00	定时器模式寄存器
0x8A	TL0	RW	0x00	定时器 0 计时器低 8 位
0x8B	TL1	RW	0x00	定时器 1 计时器低 8 位
0x8C	TH0	RW	0x00	定时器 0 计时器高 8 位
0x8D	TH1	RW	0x00	定时器 1 计时器高 8 位
0x8E	SOFT_RST	RW	0x00	软件复位寄存器
0x90	DATAC	RW	0xFF	PC 数据寄存器
0x91	WDT_CTRL	RW	0x00	看门狗溢出定时配置寄存器
0x92	WDT_EN	RW	0x00	看门狗定时使能配置寄存器
0x93	TIMER2_CFG	RW	0x00	TIMER2 配置寄存器
0x94	TIMER2_SET_H	RW	0x00	TIMER2 计数值配置寄存器，高 8 位
0x95	TIMER2_SET_L	RW	0x00	TIMER2 计数值配置寄存器，低 8 位
0x96	REG_ADDR	RW	0x00	二级总线地址配置寄存器
0x97	REG_DATA	RW	0x00	二级总线数据读写寄存器
0x98	DATAD	RW	0xFF	PD 数据寄存器
0x99	PWM1_L_L	RW	0x00	PWM1 低电平控制寄存器(低 8 位)
0x9A	PWM1_L_H	RW	0x00	PWM1 低电平控制寄存器(高 8 位)
0x9B	PWM1_H_L	RW	0x00	PWM1 高电平控制寄存器(低 8 位)
0x9C	PWM1_H_H	RW	0x00	PWM1 高电平控制寄存器(高 8 位)
0x9D	PWM2_L_L	RW	0x00	PWM2 低电平控制寄存器(低 8 位)
0x9E	PWM2_L_H	RW	0x00	PWM2 低电平控制寄存器(高 8 位)
0x9F	PWM2_H_L	RW	0x00	PWM2 高电平控制寄存器(低 8 位)
0xA1	PWM2_H_H	RW	0x00	PWM2 高电平控制寄存器(高 8 位)
0xA2	PWM_EN	RW	0x00	PWM 控制寄存器
0xB4	ADC_SPT	RW	0x00	ADC 采样时间配置寄存器

0xB5	ADC_SCAN_CFG	RW	6’0x00	ADC 扫描控制寄存器
0xB6	ADCCKC	RW	4’0x00	ADC 时钟控制寄存器
0xB8	IPL0	RW	0x00	中断优先级寄存器 0
0xB9	ADC_RDATAH	R	4’0x00	ADC 扫描结果寄存器高 4 位
0xBA	ADC_RDATAL	R	0x00	ADC 扫描结果寄存器低 8 位
0xBB	ADC_CFG1	RW	0x00	ADC 采样时序控制寄存器 1
0xBC	ADC_CFG2	RW	6’0x02	ADC 采样时序控制寄存器 2
0xBD	UART0_BDL	RW	0x00	UART0 波特率控制寄存器
0xBE	UART0_CON1	RW	0x00	UART0 控制寄存器 1
0xBF	UART0_CON2	RW	4’0x0C	UART0 控制寄存器 2
0xC0	UART0_STATE	RW	0x00	UART0 状态标记寄存器
0xC1	UART0_BUF	RW	0x00	UART0 数据寄存器
0xCA	CSD_START	RW	1’0x00	CSD 扫描开启寄存器
0xCB	SNS_SCAN_CFG1	RW	0x00	触摸按键扫描配置寄存器 1
0xCC	SNS_SCAN_CFG2	RW	0x40	触摸按键扫描配置寄存器 2
0xCD	SNS_SCAN_CFG3	RW	0x70	触摸按键扫描配置寄存器 3
0xCE	CSD_RAWDATAL	R	0x00	CSD 计数值低 8 位
0xCF	CSD_RAWDATAH	R	0x00	CSD 计数值高 8 位
0xD0	PSW	RO/RW	0x00	程序状态字寄存器
0xD1	PULL_I_SELA_L	RW	0x00	CSD 上拉电流源选择寄存器
0xD2	SNS_ANA_CFG	RW	6’0x2F	CSD 扫描参数配置寄存器
0xD3	SNS_IO_SEL1	RW	0x00	SNS 通道选择寄存器 1
0xD4	SNS_IO_SEL2	RW	0x00	SNS 通道选择寄存器 2
0xD5	SNS_IO_SEL3	RW	0x00	SNS 通道选择寄存器 3
0xD6	SNS_IO_SEL4	RW	0x00	SNS 通道选择寄存器 4
0xD7	RST_STAT	RW	rst_stat	复位标记寄存器
0xD9	ADC_IO_SEL1	RW	0x00	ADC 功能选择寄存器 1
0xDA	ADC_IO_SEL2	RW	0x00	ADC 功能选择寄存器 2
0xDB	ADC_IO_SEL3	RW	0x00	ADC 功能选择寄存器 3
0xDC	ADC_IO_SEL4	RW	0x00	ADC 功能选择寄存器 4
0xDD	PU_PA	RW	2’0x00	PA 口上拉电阻选择寄存器
0xDE	PU_PB	RW	0x00	PB 口上拉电阻选择寄存器
0xDF	PU_PC	RW	0x00	PC 口上拉电阻选择寄存器
0xE0	ACC	RW	0x00	累加器
0xE1	IRCON2	RW	0x00	中断标志寄存器 2
0xE2	PU_PD	RW	0x00	PD 口上拉电阻选择寄存器
0xE3	IICADD	RW	0x00	IIC 地址寄存器
0xE4	IICBUF	RW	0x00	IIC 发送接收数据寄存器
0xE5	IICCON	RW	0x10	IIC 配置寄存器

0xE6	IEN1	RW	0x00	中断使能寄存器 1
0xE7	IEN2	RW	0x00	中断使能寄存器 2
0xE8	IICSTAT	RO/RW	0x44	IIC 状态寄存器
0xE9	IICBUFFER	RW	0x00	IIC 发送数接收据缓存寄存器
0xEA	TRISA	RW	2’0x03	PA 方向寄存器
0xEB	TRISB	RW	0xFF	PB 方向寄存器
0xEC	TRISC	RW	0xFF	PC 方向寄存器
0xED	TRISD	RW	0xFF	PD 方向寄存器
0xEE	COM_IO_SEL	RW	0x00	COM 大灌电流选择寄存器
0xEF	ODRAIN_EN	RW	2’0x00	PA 口开漏使能寄存器
0xF0	B	RW	0x00	B 寄存器
0xF1	IRCON1	RW	0x00	中断标志寄存器 1
0xF2	PERIPH_IO_SEL	RW	7’0x40	IIC/UART0/INT 功能控制寄存器
0xF4	IPL2	RW	0x00	中断优先级寄存器 2
0xF6	IPL1	RW	0x00	中断优先级寄存器 1
0xF7	EXT_INT_CON	RW	0x15	外部中断极性控制寄存器
0xF8	DATAA	RW	2’0x03	PA 数据寄存器
0xF9	SPROG_ADDR_H	RW	4’0x00	EEPROM 地址控制寄存器
0xFA	SPROG_ADDR_L	RW	0x00	EEPROM 地址控制寄存器
0xFB	SPROG_DATA	RW	0x00	EEPROM 定操作数据寄存器
0xFC	SPROG_CMD	RW	0x00	EEPROM 定操作命令寄存器
0xFD	SPROG_TIM	RW	0x1A	EEPROM 擦写时间控制寄存器
0xFE	PD_ANA	RW	5’0x1F	模块开关控制寄存器
0xFF	SEL_LVDT_VTH	RW	2’0x00	LVDT 阈值选择寄存器

SFR 寄存器总表

注：地址以 8 或 0 结尾的寄存器可以位操作，例 0x80,0x88 的寄存器地址。

4.2. SFR 寄存器详细说明

地址	名称	位	位名称	RW	说明	复位值
0x80	DATAB	<7:0>	--	RW	PB 数据寄存器，可配置 PB 组 IO 口作为 GPIO 口时的输出电平，读取值为当前 IO 口(输入)的电平状态或配置输出值(输出)	8'hff
0x81	SP	<7:0>	SP	RW	堆栈指针寄存器	8'h07
0x82	DPL	<7:0>	DPL	RW	数据指针寄存器 0 低 8 位	8'h00
0x83	DPH	<7:0>	DPH	RW	数据指针寄存器 0 高 8 位	8'h00
0x84	SYS_CLK_CFG	<4>	XTAL_ERROR	R	保留	1'h0
		<3>	SYS_CLK_SEL	R	保留	1'h0
		<2>	WAIT_MODE	RW	WAIT 模式使能 1：芯片进入 WAIT 模式 0：芯片退出 WAIT 模式	1'h0
		<1:0>	PLL_CLK_SEL	RW	PLL 时钟分频选择寄存器: 00：12Mhz 01：6Mhz 10：4Mhz 11：保留	2'h01
0x85	INT_PE_STAT	<1>	INT_WDT_STAT	RW	WDT 中断状态标记,该位写 0 清零,写 WDT_CTRL 操作也可清零 1：中断有效 0：中断无效	1'h0
		<0>	INT_TIMER2_STAT	RW	TIMER2 中断状态标记，该位写 0 清零，写 TIMER2_CFG 操作也可清零 1：中断有效 0：中断无效	1'h0
0x86	INT_POBO_STAT	<1>	INT_PO_STAT	RW	lvdt 升压中断状态 1：升压中断有效 0：升压中断无效	1'h0
		<0>	INT_BO_STAT	RW	lvdt 降压中断状态 1：降压中断有效 0：降压中断无效	1'h0

0x87	PCON	<7:1>	--	R	保留	7'h00
		<0>	--	RW	1：为 Sleep 低功耗模式,唤醒后硬件清 0	1'h0
0x88	TCON	<7>	TF1	RW	定时器 1 溢出标志位，当 Timer1 溢出时硬件置 1，或者 Timer0 的 TH0 在模式三下溢出。	8'h05
		<6>	TR1	RW	Timer1 启动使能，设置为 1 时启动 Timer1 或启动 Time0 模式三时 TH0 计数。	
		<5>	TF0	RW	定时器 0 溢出标志位，当 Timer0 溢出时硬件置 1。	
		<4>	TR0	RW	Timer0 启动使能，设置为 1 时启动 Timer0 计数。	
		<3>	IE1	RW	外部中断 1 标志位，硬件置 1，可软件清 0。	
		<2>	IT1	R	保留	
		<1>	IE0	RW	外部中断 0 标志位，硬件置 1，可软件清 0。	
		<0>	IT0	R	保留	
0x89	TMOD	<7>	GATE1	R	保留	8'h00
		<6>	C/T1	R	保留	
		<5:4>	M1[1:0]	RW	M1-定时器1 模式选择Bit 1，M0-定时器1 模式选择Bit 0， M1M0： 00=模式 0 －13 位定时器/计数器 01=模式 1 －16 位定时器/计数器 10=模式 2-自动重载初值的8 位计数器 11=模式 3－两个8 位计数器	
		<3>	GATE0	R	保留	
		<2>	C/T0	R	保留	
		<1:0>	M0[1:0]	R/W	M1-定时器0 模式选择Bit 1，M0-定时器1 模式选择Bit 0， M1M0： 00=模式 0－8 位定时器/计数器 01=模式 1－16 位定时器/计数器 10=模式 2-自动重载初值的8 位计数器	

					11=模式 3 - 两个 8 位计数器	
0x8A	TL0	<7:0>	TL0	RW	定时器 0 计时器低 8 位	8'h00
0x8B	TL1	<7:0>	TL1	RW	定时器 1 计时器低 8 位	8'h00
0x8C	TH0	<7:0>	TH0	RW	定时器 0 计时器高 8 位	8'h00
0x8D	TH1	<7:0>	TH1	RW	定时器 1 计时器高 8 位	8'h00
0x8E	SOFT_RST	<7:0>	--	RW	软件复位寄存器，只有在寄存器值为0x55 时，才产生软件复位	8'h00
0x90	DATAC	<7:0>	--	RW	PC 数据寄存器，可配置 PC 组 IO 口作为 GPIO 口时的输出电平，读取值为当前 IO 口(输入)的电平状态或配置输出值(输出)	8'hff
0x91	WDT_CTRL	<2:0>	WDT_TIME_SEL	R/W	看门狗溢出定时配置寄存器，定时长度如下： 0x0：18ms 0x1：36ms 0x2：72ms 0x3：144ms 0x4：288ms 0x5：576ms 0x6：1152ms 0x7：2304ms	3'h0
0x92	WDT_EN	<7:0>	WDT_EN	R/W	看门狗定时使能配置寄存器，当配置值为 8'h55 时，看门狗被关闭	8'h00
0x93	TIMER2_CFG	<3>	TIMER2_CNT_MOD	RW	Timer2 计数步进模式选择寄存器 1：计数步进为 65536 个时钟 0：计数步进为一个时钟	1'h0
		<2>	TIMER2_CLK_SEL	RW	Timer2 时钟选择寄存器 1:保留 0:选择 clk_rc	1'h0
		<1>	TIMER2_RLD	RW	TIMER2 自动重载使能寄存器 1：自动重载模式 0：手动重载模式	1'h0
		<0>	TIMER2_EN	RW	TIMER2 计数使能寄存器 配置 1 开启定时，配置 0 停止定时；在手动重载模式下会在计数完成后硬件自动清零该寄存器，停止计数，在自动重载模式下会在计数完成后维持该使能寄存器，自动重新从零计数，无论哪种模式，计数过程中配置该寄存器为 1 均会开始	1'h0

					从零计数。	
0x94	TIMER2_SET_H	<7:0>	--	RW	TIMER2 计数值配置寄存器，高 8 位，扫描过程中配置该寄存器会重新计数。	8'h00
0x95	TIMER2_SET_L	<7:0>	--	RW	TIMER2 计数值配置寄存器，低 8 位，扫描过程中配置该寄存器会重新计数	8'h00
0x96	REG_ADDR	<5:0>	REG_ADDR	R/W	二级总线地址配置寄存器	6'h00
0x97	REG_DATA	<7:0>	REG_DATA	R/W	二级总线数据读写寄存器	8'h00
0x98	DATAD	<7:0>	--	RW	PD 数据寄存器，可配置 PD 组 IO 口作为 GPIO 口时的输出电平，读取值为当前 IO 口(输入)的电平状态或配置输出值(输出)。	8'hff
0x99	PWM1_L_L	<7:0>	--	RW	PWM1 低电平控制寄存器(低 8 位)	8'h00
0x9A	PWM1_L_H	<7:0>	--	RW	PWM1 低电平控制寄存器(高 8 位)	8'h00
0x9B	PWM1_H_L	<7:0>	--	RW	PWM1 高电平控制寄存器(低 8 位)	8'h00
0x9C	PWM1_H_H	<7:0>	--	RW	PWM1 高电平控制寄存器(高 8 位)	8'h00
0x9D	PWM2_L_L	<7:0>	--	RW	PWM2 低电平控制寄存器(低 8 位)	8'h00
0x9E	PWM2_L_H	<7:0>	--	RW	PWM2 低电平控制寄存器(高 8 位)	8'h00
0x9F	PWM2_H_L	<7:0>	--	RW	PWM2 高电平控制寄存器(低 8 位)	8'h00
0xA1	PWM2_H_H	<7:0>	--	RW	PWM2 高电平控制寄存器(高 8 位)	8'h00
0xA2	PWM_EN	<5>	PWM0_CH3_CMOD	RW	PWM0 通道 3 占空比模式选择寄存器 1：选择通道 0 占空比 0：选择自身通道占空比	1'h0
0xA2	PWM_EN	<2>	PWM2_EN	RW	PWM2 模块使能寄存器 1：使能 0：不使能	1'h0
		<1>	PWM1_EN	RW	PWM1 模块使能寄存器 1：使能 0：不使能	1'h0

		<0>	-	R	保留	1'h0
0xA8	IEN0	<7>	EA	RW	EA-中断允许位。EA=0 屏蔽所有的中断(EA 优先于中断源各自的中断使能位)。EA=1，中断打开， 每个中断源的中断请求是允许还是被禁止，还需由各自的允许位确定。	8'h00
		<6:4>	--	R	保留	
		<3>	ET1	RW	ET1-定时器1 溢出中断允许位。ET1=0，禁止定时器1(TF1)申请中断。ET1=1，允许TF1 标志位申请中断。	
		<2>	EX1	RW	EX1-INT_EXT1 允许位。EX1=0，禁止INT_EXT1 申请中断。EX1=1，允许INT_EXT1 申请中断。	
		<1>	ET0	RW	ET0-定时器0 溢出中断 允许位。ET0=0，禁止定时器0(TF0)申请中断。ET0=1，允许TF0 标志位申请中断。	
		<0>	EX0	RW	EX0-INT_EXT0 允许位。EX0=0，禁止INT_EXT0 申请中断。EX0=1，允许INT_EXT0 申请中断。	
0xB0	DP_CON	<7:1>	--	R	保留	7'h0
		0	COM_MOD	RW	大电流 IO 口驱动使能 1：COM 口功能锁定，作为大电流 IO 口工作 0：COM 口功能不锁定，可通过配置为其他功能 COM 口锁定大电流 IO 口时，通过配置 GPIO 寄存器输出驱动时序。	1'h0
0xB4	ADC_SPT	<7:0>	ADC_SPT	RW	ADC 采样时间配置寄存器 采样时间：sample_Timer = (ADC_SPT+1)*4*Tadc_clk	8'h00
0xB5	ADC_SCAN_CFG	<5:1>	ADC_ADDR	RW	ADC 通道地址选择寄存器	5'h0
		<0>	ADC_START	RW	ADC 扫描开启寄存器	1'h0

0xB6	ADCCKC	<3:2>	ADCCKV	RW	模拟输入时钟信号分频选择 0：12MHZ 1：8MHZ 2：4MHZ 3：2MHZ	4'h0
		<1:0>	ADCK	RW	ADC_CLK 分频选择 0：8MHZ 1：6MHZ 2：4MHZ 3：3MHZ	
0xB8	IPL0	<7:4>	-	R	保留	8'h00
		<3>	PT1	RW	PT1-TF1(Timer1 中断)优先级选择位。 PT1=0 时TF1(Timer1 中断)为低优先级， PT1=1 时TF1(Timer1 中断)为高优先级。	
		<2>	PX2	RW	PX2- INT_EXT1 中断优先级选择位。PX2=0 时INT_EXT1 为低优先级，PX2=1 时INT_EXT1 为高优先级。	
		<1>	PT0	RW	PT0-TF0(Timer0 中断)优先级选择位。 PT0=0 时TF0(Timer0 中断)为低优先级， PT0=1 时TF0(Timer0 中断)为高优先级。	
		<0>	PX0	RW	PX0- INT_EXT0 中断优先级选择位。 PX0=0 时INT_EXT0 为低优先级， PX0=1 时INT_EXT0 为高优先级。	
0xB9	ADC_RDATAH	<3:0>	ADC_RAWDATA<11:8>	R	ADC 扫描结果寄存器	4'h0
0xBA	ADC_RDATAL	<7:0>	ADC_RAWDATA<7:0>			8'h00
0xBB	ADC_CFG1	<7:3>	ADCWNUM	RW	采样完毕后距离转换间隔时间选择 3+ADCWNUM(ADC_CLK)	5'h0
		<2>	SAMBG	RW	采样时序与比较时序间隔选择：0：间隔 0； 1： 间隔 1(ADC_CLK)	1'h0
		<1:0>	SAMDEL	RW	采样延迟时间选择	2'h0

					0：0；1：2；2：4；3：8(ADC_CLK)	
0xBC	ADC_CFG2	<6>	FILTER_R_SEL	RW	输入信号滤波选择，0 为不加 RC 滤波，1 为加 RC 滤波	1'h0
		<5:4>	VREF_IN_ADC_SEL	RW	输入给 ADC26 基准电压选择 01：2.253V；其它：保留 使用时需要从芯片 Flash 读取校准电压值， VREF_IN_ADC_SEL 档电压= {CBYTE[0x43C6],CBYTE[0x43C7]}mV。	2'h0
		<3:2>	ADC_I_SEL[1:0]	RW	ADC 偏置电流大小选择寄存器 ADC_I_SEL[0]: 0 为比较器偏置电流为 4uA; 1 为比较器偏置电流为 5uA; ADC_I_SEL[1]: 0 为运放偏置电流为 4uA; 1 为运放偏置电流为 5uA;	2'h0
		<1:0>	CTRL_SEL[1:0]	RW	ADC 比较器失调消除选择信号，默认值为 10 CTRL_SEL[1:0]: 00/01:为先采样再失调消除; 10：所有开关一起断开; 11：开关依次断开;	2'h2
0xBD	UART0_BDL	<7:0>	--	RW	波特率控制寄存器 波特率模数除数寄存器低 8 位,bandrate={UART0_BDH[1:0],UART0_BDL}, bandrate=0 时不生成波特率时钟，当 bandrate=1~1023 时，波特率=BUSCLK/(16xbandrate)	8'h00
0xBE	UART0_CON1	<6:0>	--	RW	模式控制寄存器 bit[6]:uart0_enable，模块使能，1：模块使能，0：模块关闭 bit[5]:receive_enable，接收器使能，1：接收器打开，0：接收器关闭 bit[4]:multi_mode，多处理器通信模式，1：模式使能，0：模式禁能 bit[3]:stop_mode，stop 位宽选	7'h00

					择，1：2 位，0：1 位 bit[2]:data_mode，数据模式选择，1:9 位模式，0:8 位模式 bit[1]:parity_en，奇偶校验使能，1：奇偶效验使能，0：奇偶效验不使能 bit[0]:parity_sel，奇偶校验选择，1：奇校验，0：偶校验	
0xBF	UART0_CON2	<3:2>	--	RW	控制寄存器： bit[3]:tx_empty_ie,发送中断使能，1：中断使能，0：中断禁止(用于轮询模式) bit[2]:rx_full_ie，接收中断使能，1：中断使能，0：中断禁止(用于轮询模式)	2’0x03
		<1:0>	UART0_BDH	RW	bit[1:0]:UART0_BDH,波特率模数除数寄存器高 2 位	2’0x00
0xC0	UART0_STATE	<6:0>	--	RO/RW	状态标记寄存器： bit[6]:r8,接收器的第 9 个数据，只读 bit[5]:t8，发射器的第 9 个数据 bit[4]:tx_empty_if,发送中断标记，1：发送缓存为空，0：发送缓存为满，软件写 0 清零 bit[3]:rx_full_if，接收中断标记，1：接收缓存为满，0：接收缓存为空，软件写 0 清零 bit[2]:rx_overflow_if，接收溢出标记，1：接收溢出(新数据丢失)，0：没有溢出，软件写 0 清零 bit[1]:frame_err_if，帧错误标记，1：检测到帧错误，0：未检测到帧错误，软件写 0 清零 bit[0]:parity_err_if,奇偶校验错误标记，1：接收器奇偶校验错误，0：奇偶校验正确，软件写 0 清零	7’h00
0xC1	UART0_BUF	<7:0>	--	RW	数据寄存器 读返回只读接收数据缓冲器的内容，写进入只写发送数据缓冲器	8’hff

0xC2	SCI_BDH	<7:0>	--	RW	波特率控制寄存器 bit[7]:break_check_ie，间隔段检测中断使能，1：中断使能，0：中断禁止 bit[6]:rx_edge_ie，RxD 管脚活动边沿中断使能，1：中断使能，0：中断禁止 bit[5]:reserved bit[4:0]:波特率模数除数寄存器高 5 位	8'h00
0xC3	SCI_BDL	<7:0>	--	RW	波特率控制寄存器 波特率模数除数寄存器低 8 位,bandrate={SCI_BDH[4:0],SCI_BDL}， bandrate=0 时不生成波特率时钟，当 bandrate=1~8191 时，SCI 波特率= BUSCLK/(16xbandrate)	8'h00
0xC4	SCI_C1	<7:0>	--	RW	控制寄存器 1 bit[7]:cycle_mode，循环模式使能，1：循环模式或单线模式，txd 连接 rxd，0：正常双线模式 bit[6]:stop_mode，stop 位数选择，1：2bits，0：1bit bit[5]:single_txd，单线模式使能，1：在 cycle_mode=1 时选择单线模式，txd 管脚有效，0：内部循环模式，txd 管脚无效 bit[4]:data_mode,传输数据模式选择，1:9 位模式(第 9 位为奇偶校验位)，0:8 位模式 bit[3]:parity_en，奇偶校验使能，1：奇偶效验使能，0：奇偶效验不使能 bit[2]:parity_sel，奇偶校验选择，1：奇校验，0：偶校验 bit[1]:rate_match_en，同步段(0x55)波特率自动匹配使能，1：自适应波特率更新，0：固定配置波特率 bit[0]:sci_enable，模块工作时	8'h00

					钟门控使能，写 1 表示使能有效，打开模块工作时钟，写 0 会关闭模块工作时钟，并且复位功能模块	
0xC5	SCI_C2	<7:0>	--	RW	控制寄存器 2： bit[7]:tx_empty_ie，发送缓存空中断使能，1：中断使能，0：中断禁止 bit[6]:tx_finish_ie，发送完成中断使能，1：中断使能，0：中断禁止 bit[5]:rx_full_ie，接收满中断使能，1：中断使能，0：中断禁止 bit[4]:idle_ie，闲置线路中断使能，1：中断使能，0：中断禁止 bit[3]:trans_enable，发射器使能，1：发射器打开，0：发射器关闭 bit[2]:receive_enable，接收器使能，1：接收器打开，0：接收器关闭 bit[1]:rwu，接收器唤醒控制，1：接收器处于待机状态，等待唤醒条件，0：接收器正常运行 bit[0]:break_trans_start，发送间隔段，先后将 1 和 0 写入该位，即在发送数据流中排入了一个间隔段	8'h00
0xC6	SCI_C3	<7:0>	--	RO/RW	控制寄存器 3 bit[7]:r8，接收器的第 9 个数据，只读 bit[6]:t8，发射器的第 9 个数据 bit[5]:txd_direct，单线模式下 txd 管脚方向选择，1：TxD 管脚是单线模式中的输出，0：TxD 管脚是单线模式中的输入 bit[4]:txd_inv，txd 端数据反相，1：发送数据被反转，0：发送数据未反转 bit[3]:rxd_inv，rxd 端数据反相，1：接收数据被反转，0：接收数据	8'h00

					未反转 bit[2]:rwu_idlesel，接收唤醒闲置检测，1：在接收待机状态(RWU=1)期间，检测到闲置字符时设置 IDLE 位，0：在接收待机状态(RWU=1)期间，检测到闲置字符时不设置 IDLE 位 bit[1]:idle_sel，闲置线路类型选择，1：停止位后闲置字符位计数开始，0：开始位后闲置字符位计数开始，计数 10 位时间(如果 data_mode=1 或 stop_mode=1，则分别增加 1 位时间) bit[0]:wake_sel，接收器唤醒方式选择，1：地址标记唤醒，0:闲置线路唤醒	
0xC7	SCI_S2	<7:0>	--	RO/RW	同步间隔段控制寄存器 bit[7]:break_check_if，间隔段检测中断标记，1：检测到间隔段，0：未检测到间隔段，该位写 1 清除，写 0 无效 bit[6]:rx_edge_if，RxD 管脚活动边沿中断标记，1：接收管脚上出现活动边沿，0：接收管脚上未出现活动边沿，该位写 1 清除，写 0 无效 bit[5]:rx_active_flag，接收器活动标记，只读，1：接收器活动，0：接收器闲置 bit[4]:reserved bit[3]:reserved bit[2]:reserved bit[1]:break_trans_size，间隔段生成位长度，1:用 13 位时间(如果 data_mode=1 或 stop_mode=1，则分别增加 1 位时间)长度发送，0:用 10 位时间(如果 data_mode=1 或 stop_mode=1，则分别增加 1 位时间)长度发送 bit[0]:break_check_en，间隔段	8'h00

					检测使能，1：在 11 位时间(如果 data_mode=1 或 stop_mode=1，则分别增加 1 位时间)长度上检测，0：不检测	
0xC8	SCI_S1	<7:0>	--	RO	中断状态标记寄存器： bit[7]:tx_empty_if，发送缓存空中断标记，1：发送缓存为空，0：发送缓存为满，只读 bit[6]:tx_finish_if，发送完成中断标记，1：发送完成，发射器闲置，0：发射器正在工作，只读 bit[5]:rx_full_if，接收满中断标记，1：接收缓存为满，0：接收缓存为空，只读 bit[4]:idle_if，闲置线路中断标记，1：检测到闲置线路，0：未检测到闲置线路，只读 bit[3]:rx_overflow_if，接收溢出标记，1：接收溢出(新数据丢失)，0：没有溢出，只读 bit[2]:noise_err_if，噪声标记，1：检测到噪声，0：未检测到噪声，只读 bit[1]:frame_err_if，帧错误标记，1：检测到帧错误，0：未检测到帧错误，只读 bit[0]:parity_err_if，奇偶校验错误标记，1：接收器奇偶校验错误，0：奇偶校验正确，只读	8'h00
0xC9	SCI_D	<7:0>	--	RW	SCI 数据寄存器 读返回只读接收数据缓冲器的内容，写进入只写发送数据缓冲器	8'hff
0xCA	CSD_START	<0>	--	RW	CSD 扫描开启寄存器	1'h0
0xCB	SNS_SCAN_CFG1	<6>	SW_PRE_OFF	RW	前端充放电时钟开关控制 1：关闭 sw_clk；0：打开 sw_clk	1'h0
		<5:0>	PRS_DIV	RW	前端充放电时钟频率选择寄存器: 0~61：为固定频率： F=F48m/2/(PRS_DIV+4) （6M~369K）；	6'h0

					62：最高频率 3M，最低频率 1M，中心频率 1.5M，正态分布； 63：最高频率 3M，最低频率 1M，中心频率 1.5M，均匀分布；	
0xCC	SNS_SCAN_CFG2	<6>	PULL_I_SELA_H	RW	CSD 上拉电流源配置最高位	1'h1
		<5>	PARALLEL_EN	RW	SNS 通道并联使能寄存器 1：多通道并联 0：单通道	1'h0
		<4:0>	CSD_ADDR	RW	检测通道的地址,对应通道号 0~25	5'h0
0xCD	SNS_SCAN_CFG3	<6:4>	RESO	RW	计数器位数选择寄存器 000 ：9 位；001 ：10 位；010 ：11 位；011 ：12 位；100 ：13 位；101 ：14 位；110： 15 位；111 ：16 位。	3'h7
		<3:2>	CSD_DS	RW	计数时钟频率选择寄存器 00：24M； 01：12M； 10：6M； 11：4M； 默认 0	2'h0
		<1>	PRE_CHRG_SEL	RW	预充电时间选择 0：20us 1：40us	1'h0
		<0>	INIT_DISCHRG_SEL	RW	预放电时间选择 0：2us 1：10us	1'h0
0xCE	CSD_RAWDATAL	<7:0>	RAWDATA<7:0>	R	CSD 的计数值	8'h00
0xCF	CSD_RAWDATAH	<7:0>	RAWDATA<15:8>			8'h00
0xD0	PSW	<7>	CY	R/W	进位标志位。 当加法生成进位或减法产生借位时设置，否则清除。当 CJNE 第一个操作数小于第二个操作数时设置，由 MUL 和 DIV 指令清除。也通过鼠标指令(RLC, RRC)和逐位指令影响。	8'h00
		<6>	AC	R/W	辅助进位标志位 当加法从累加器第三位到第四位产生进位时设置，或者当减法从第三位到第四位产生借位时，否则清零。	

		<5>	F0	R/W	0 标志位。 可供用户使用的通用标签。	
		<4:3>	RS[1:0]	R/W	工作寄存器组选择： 选择有效的工作寄存器组: RS[1:0] Bank IRAM Area 00 0 0x00-0x07 01 1 0x08-0x0F 10 2 0x10-0x17 11 3 0x18-0x1F	
		<2>	OV	R/W	溢出标志位。 当加法产生累加器位 6 和 7 的不同进位时，或者减法产生累加器位 6 和 7 的借位。否则清除。OV 标志位表示签名的 8 位数字的结果超出了限制（大于 127 或小于 -128）. 当乘法结果大于 255 或试图除以 0 时，也会设置溢出标志。	
		<1>	F1	R/W	1 标志位。 可供用户使用的通用标签。	
		<0>	P	R	奇偶标志位。始终包含累加器中所有位的形式 2 的总和。	
0xD1	PULL_I_SELA_L	<7:0>	PULL_I_SEL<7:0>	RW	CSD 上拉电流源大小选择开关,默认值 0	8'h00
0xD2	SNS_ANA_CFG	<5:3>	RB_SEL	RW	Rb 电阻大小选择 0：10k；1：20k；2：30k；3：40k；4：60k；5：80k；6：150k；7：300k； 使用时需要从芯片 Flash 读取 Rb80K 校准值： CBYTE[0x43CD]K/80K,进行比例计算归一化灵敏度。	3'h5
		<2:0>	VTH_SEL	RW	VTH 电压选择信号 VTH 电压选择信号，000 为 1.809977V，001 选择 2.144228V，010 选择 2.491199V，011 选择 2.825942V，100 选择 3.17938V，101 选择 3.514271V，110 选择 3.861597V，111 选择 4.101614V	3'h7
0xD3	SNS_IO_SEL1	<7:0>	SEL_SENSOR[7:0]	RW	SENSOR 口选择使能 1：选择 SENSOR，0：不选择 SENSOR	8'h00

0xD4	SNS_IO_SEL2	<7:0>	SEL_SENSOR [15:8]	RW	SENSOR 口选择使能 1：选择 SENSOR，0：不选择 SENSOR	8'h00
0xD5	SNS_IO_SEL3	<7:0>	SEL_SENSOR [23:16]	RW	SENSOR 口选择使能 1：选择 SENSOR，0：不选择 SENSOR	8'h00
0xD6	SNS_IO_SEL4	<1:0>	SEL_SENSOR [25:24]	RW	SENSOR 口选择使能 1：选择 SENSOR，0：不选择 SENSOR	2'h0
0xD7	RST_STAT	<6:0>	--	RW	复位标志位寄存器： {DEBUG_F，SOFT_F，PROG_F，ADDROF_F，BO_F，PO_F，WDTRST_F}	rst stat
0xD8	SCI_INT_CLR	<7:0>	--	W	SCI 模块中断清除寄存器 bit[7]:clr_tx_empty_if，发送缓存空中断清除位，该位写 1 清除相应中断，写 0 无效 bit[6]:clr_tx_finish_if，发送完成中断清除位，该位写 1 清除相应中断，写 0 无效 bit[5]:clr_rx_full_if，接收满中断清除位，该位写 1 清除相应中断，写 0 无效 bit[4]:clr_idle_if，闲置线路中断清除位，该位写 1 清除相应中断，写 0 无效 bit[3]:clr_rx_overflow_if，接收溢出标记清除位，该位写 1 清除相应标记，写 0 无效 bit[2]:clr_noise_err_if，噪声标记清除位，该位写 1 清除相应标记，写 0 无效 bit[1]:clr_frame_err_if，帧错误标记清除位，该位写 1 清除相应标记，写 0 无效 bit[0]:clr_parity_err_if，奇偶校验错误标记清除位，该位写 1 清除相应标记，写 0 无效	8'h00
0xD9	ADC_IO_SEL1	<7:0>	SEL_ADC [7:0]	RW	ADC 功能选择 1：选择 ADC 功能，0：不选择 ADC 功能	8'h00
0xDA	ADC_IO_SEL2	<7:0>	SEL_ADC [15:8]	RW	ADC 功能选择 1：选择 ADC 功能，0：不选择 ADC	8'h00

					功能	
0xDB	ADC_IO_SEL3	<7:0>	SEL_ADC [23:16]	RW	ADC 功能选择 1：选择 ADC 功能，0：不选择 ADC 功能	8'h00
0xDC	ADC_IO_SEL4	<1:0>	SEL_ADC [25:24]	RW	ADC 功能选择 1：选择 ADC 功能，0：不选择 ADC 功能	2'h0
0xDD	PU_PA	<1:0>	--	RW	PA 口上拉电阻控制寄存器 1：上拉电阻使能 0：上拉电阻不使能	2'h0
0xDE	PU_PB	<7:0>	--	RW	PB 口上拉电阻控制寄存器 1：上拉电阻使能 0：上拉电阻不使能	8'h0
0xDF	PU_PC	<7:0>	--	RW	PC 口上拉电阻控制寄存器 1：上拉电阻使能 0：上拉电阻不使能	8'h0
0xE0	ACC	<7:0>	ACC	RW	累加器 目标寄存器适用于所有算术和逻辑运算。	8'h00
0xE1	IRCON2	<7:3>	--	R	保留	8'h00
		<2>	IE10	R	保留	
		<1>	IE9	R	保留	
		<0>	IE8	RW	LVDT 中断标志	
0xE2	PU_PD	<7:0>	--	RW	PD 口上拉电阻控制寄存器 1：上拉电阻使能 0：上拉电阻不使能	8'h00
0xE3	IICADD	<7:1>	IICADD	RW	地址寄存器	8'h00
0xE4	IICBUF	<7:0>	IICBUF	RW	IIC 发送接收数据缓冲器	8'h00
0xE5	IICCON	<7:6>	--			0
		<5>	IIC_RST	RW	IIC 模块复位信号 1：IIC 模块发生复位操作 0：IIC 模块正常工作	0
		<4>	RD_SCL_EN	RW	主机读拉低时钟线控制位 1:使能主机读拉低时钟线功能； 0:不使能主机读拉低时钟线功能	1

		<3>	WR_SCL_EN	RW	主机写拉低时钟线控制位, 1:使能写拉低时钟线的功能; 0:不使能写拉低时钟线的功能	0
		<2>	SCLEN	RW	IIC 时钟使能位 1=时钟正常工作; 0=拉低时钟线	0
		<1>	SR	RW	IIC 转换率控制位 1=转换率控制被关闭以适应标准速度模式(100K); 0=转换率控制被使能以适应快速速度模式(400K)	0
		<0>	IICEN	RW	IIC 工作使能位 1=IIC 正常工作; 0=IIC 不工作	0
0xE6	IEN1	<7>	EX7	RW	WDT/Timer2 中断使能	8'h00
		<6>	EX6	R	保留	
		<5>	EX5	RW	CSD 中断使能	
		<4>	EX4	RW	ADC 中断使能	
		<3>	EX3	RW	IIC 中断使能	
		<2>	EX2	RW	外部中断2 中断使能	
		<1:0>	-	R	保留	
0xE7	IEN2	<7:3>	--	R	保留	8'h00
		<2>	EX10	R	保留	
		<1>	EX9	RW	UART0 中断使能	
		<0>	EX8	RW	LVDT 中断使能	
0xE8	IICSTAT	<7>	IIC_START	R	开始信号标志位 1=表示检测到了启动位; 0=表示未检测到启动位	0
		<6>	IIC_STOP	R	停止信号标志位 1=表示处于停止状态; 0=表示未检测到停止位	1
		<5>	IIC_RW	R	读写标志位 记录最近一次地址匹配后,从地址字节中获得的读/写信息, 1=表示读操作; 0=表示写操作	0
		<4>	IIC_AD	R	地址数据标志位 1=表示最近接收或者发送的字节是数据; 0=表示最近接收或者发送的字节	0

					是地址	
		<3>	IIC_BF	R	IICBUF 满标志位 在 IIC 总线方式下接收时： 1=表示接收成功，缓冲器已经满； 0=表示接收未完成，缓冲器还为空 在 IIC 总线方式下发送时： 1=表示数据发送正在进行(不包括应答位和停止位)，缓冲器还是满的； 0=表示数据发送已经完成(不包括应答位和停止位)，缓冲器已空	0
		<2>	IIC_ACK	R	应答标志位 1：表示无效的应答信号； 0：表示有效的应答信号	1
		<1>	IIC_WCOL	RW	写冲突标志位 1=表示 IIC 正在发送当前的数据的时候，新的数据试图写入发送缓冲器；新的数据是不能被写入缓冲器的； 0=未发生写冲突	0
		<0>	IIC_RECOV	RW	接收溢出标志位 1=表示 IIC 接收的前一个数据还没有取走时，又接收到了新的数据，新的数据是不能被缓冲器接收的 ； 0=表示未发生接收溢出	0
0xE9	IICBUFFER	<7:0>	IICBUFFER	RW	IIC 发送数接收据缓存寄存器	8'h00
0xEA	TRISA	<1:0>	--	RW	PA 方向寄存器，0：输出，1：输入	2'h3
0xEB	TRISB	<7:0>	--	RW	PB 方向寄存器，0：输出，1：输入	8'hff
0xEC	TRISC	<7:0>	--	RW	PC 方向寄存器，0：输出，1：输入	8'hff
0xED	TRISD	<7:0>	--	RW	PD 方向寄存器，0：输出，1：输入	8'hff
0xEE	COM_IO_SEL	<7:0>		RW	COM 口选择配置寄存器，对应 PB 口 1：选择 COM 口模式	8'h0

					0：选择 IO 口模式	
0xEF	ODRAIN_EN	<1:0>	--	RW	PA0/1 开漏输出使能寄存器 1：开漏输出 0：CMOS 输出	2'h0
0xF0	B	<7:0>	B	RW	B 寄存器。 乘法和除法运算的源和目标寄存器。	8'h00
0xF1	IRCON1	<7>	IE7	RW	WDT/Timer2 中断标志	8'h00
		<6>	IE6	R	保留	
		<5>	IE5	RW	CSD 中断标志	
		<4>	IE4	RW	ADC 中断标志	
		<3>	IE3	RW	IIC 中断标志	
		<2>	IE2	RW	外部中断1中断标志	
		<1:0>	-	R	保留	
0xF2	PERIPH_IO_SEL	<6>	IIC_AFIL_SEL	RW	IIC 口模拟滤波选择使能 1：选择模拟滤波功能 0：不选择模拟滤波功能	1'h1
		<5>	IIC_DFIL_SEL	RW	IIC 口数字滤波选择使能 1：选择数字滤波功能 0：不选择数字滤波功能	1'h0
		<4:3>	UART0_IO_SEL	RW	UART0 口选择使能 00：PA0/1 口选择 UART0 功能 01：PB3/4 口选择 UART0 功能 1x：PD4/5 口选择 UART0 功能	2'h0
		<2>	INT2_IO_SEL	RW	INT2 口选择使能，对应 PD7 1：选择 INT2 功能 0：不选择 INT2 功能	1'h0
		<1>	INT1_IO_SEL	RW	INT1 口选择使能，对应 PD6 1：选择 INT1 功能 0：不选择 INT1 功能	1'h0
		<0>	INT0_IO_SEL	RW	INT0 口选择使能，对应 PD0 1：选择 INT0 功能 0：不选择 INT0 功能	1'h0
0xF4	IPL2	<7:3>	--	R	保留	8'h00
		<2>	{IPH2.2, IPL2.2}	R	保留	
		<1>	{IPH2.1, IPL2.1}	RW	UART0 中断优先级 1:为高	

					0:为低	
		<0>	{IPH2.0, IPL2.0}	RW	LVDT 中断优先级 1:为高 0:为低	
0xF6	IPL1	<7>	{IPH1.7, IPL1.7}	RW	WDT/Timer 2 中断优先级 1:为高 0:为低	8'h00
		<6>	{IPH1.6, IPL1.6}	R	保留	
		<5>	{IPH1.5, IPL1.5}	RW	CSD 中断优先级 1:为高 0:为低	
		<4>	{IPH1.4, IPL1.4}	RW	ADC 中断优先级 1:为高 0:为低	
		<3>	{IPH1.3, IPL1.3}	RW	IIC 中断优先级 1:为高 0:为低	
		<2>	{IPH1.2, IPL1.2}	RW	外部中断优先级 1:为高 0:为低	
		<1:0>	--	R	保留	
0xF7	EXT_INT_CON	<5:4>	INT2_POLARITY	RW	外部中断 2 触发极性选择: INT2_POLARITY=01: 下降沿(低功耗模式下低电平唤醒) INT2_POLARITY=10: 上升沿(低功耗模式下高电平唤醒) INT2_POLARITY=00/11:双沿(低功耗模式下低电平唤醒)	2'h1
		<3:2>	INT1_POLARITY	RW	外部中断 1 触发极性选择: INT1_POLARITY=01: 下降沿(低功耗模式下低电平唤醒) INT1_POLARITY=10: 上升沿(低功耗模式下高电平唤醒) INT1_POLARITY=00/11:双沿(低功耗模式下低电平唤醒)	2'h1
		<1:0>	INT0_POLARITY	RW	外部中断 0 触发极性选择: INT0_POLARITY=01: 下降沿(低功耗模式下低电平唤醒)	2'h1

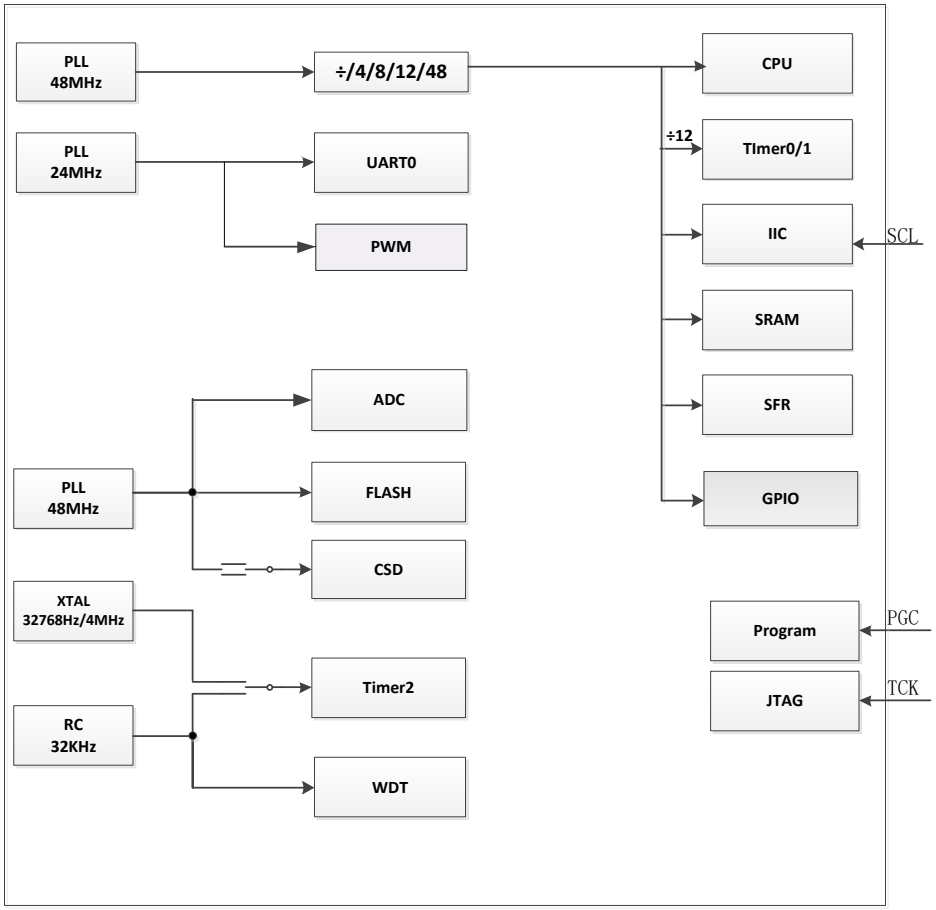
					INT0_POLARITY=10：上升沿(低功耗模式下高电平唤醒) INT0_POLARITY=00/11：双沿(低功耗模式下低电平唤醒)	
0xF8	DATAA	<1:0>	--	RW	PA 数据寄存器，可配置 PA 组 IO 口作为 GPIO 口时的输出电平，读取值为当前 IO 口(输入)的电平状态或配置输出值(输出)	2'h3
0xF9	SPROG_ADDR_H	<3:0>	----	RW	Bit[2]：选择 EEPROM 块(可进行页擦除和字节烧写)， 0：表示选择 block0；1：表示选择 block1。每一 block 大小 1024Bytes。 Bit[1:0]：表示 EEPROM 块地址的高 2 位,SPROG_ADDR[9:8]。	3'h00
0xFA	SPROG_ADDR_L	<7:0>	----	RW	Bit[7:0]：表示 EEPROM 块地址的低 8 位,SPROG_ADDR[7:0]。	8'h00
0xFB	SPROG_DATA	<7:0>	----	RW	EEPROM 烧写：待写入的数据	8'h00
0xFC	SPROG_CMD	<7:0>	----	RW	写入 0x96：EEPROM 页擦除 写入 0x69：EEPROM 字节烧写	8'h00
0xFD	SPROG_TIM	<7:0>	----	RW	bit[7:5]：0~7 对应烧写时间 25~60us,5us 步进。默认 35us bit[4:0]：0~22 对应擦除时间 1~23ms(步进 1ms)。 23~31 对应擦除时间 24~40ms(步进 2ms)。默认 30ms	8'h1a
0xFE	PD_ANA	<4>	PD_LVDT	RW	LVDT 控制寄存器，1：关闭，0：打开，默认关闭	1'h1
		<3>	PD_BOR	RW	BOR 控制寄存器，1：关闭，0：打开,VBOR = 2.1V，默认关闭	1'h1
		<2>	PD_XTAL_32K	RW	1：关闭，0：保留	1'h1
		<1>	PD_CSD	RW	CSD 工作控制寄存器：PD_CSD=0 CSD 模块正常工作；PD_CSD=1 CSD 模块不工作 CSD_EN=0 时，CSD 功能关闭， CSD_EN=1 时，CSD 功能开启，模拟 CSD 由 PD_CSD 控制开关	1'h1
		<0>	PD_ADC	RW	模拟 ADC 关断控制寄存器： PD_ADC=0 ADC 模块正常工作； PD_ADC=1 ADC 模块不工作	1'h1

0xFF	SEL_LVDT_VTH	<1:0>	--	RW	LVDT 阈值选择； 00=2.4V； 01=3.0V； 10=3.6V； 11=4.2V	2'h0

SFR 寄存器详细说明表

第 5 章 时钟、复位、工作模式及看门狗

5.1. 时钟定义



时钟方框图

5.2. 系统时钟选择寄存器详细说明

地址	名称	位	位名称	RW	说明	复位值
0x84	SYS_CLK_CFG	<4>	XTAL_ERROR	R	保留	1'h0
		<3>	SYS_CLK_SEL	R	保留	1'h0
		<2>	WAIT_MODE	RW	WAIT 模式使能 1：芯片进入 WAIT 模式 0：芯片退出 WAIT 模式	1'h0
		<1:0>	PLL_CLK_SEL	RW	PLL 时钟分频选择寄存器: 00：12Mhz 01：6Mhz 10：4Mhz 11：保留	2'h01

GDMC191XXXX 系列时钟定义如下：

RC1MHz：内置 RC 振荡器，频率为 1MHz，及 PLL 时钟。

RC32KHz：内置独立 RC 振荡 32KHz 时钟，该时钟作为看门狗时钟、Timer2 时钟。

PLL_48MHz：锁相环产生的 48MHZ 时钟，直接用于 CSD、ADC、Flash 控制，及分频后得到系统时钟。

SCL：IIC 主机时钟，由 IIC Master 主机发出，作为 IIC 通信时钟。

PGC：编程时钟，编程烧录程序时的下载时钟。

TCK：调试时钟。

时钟设置相关寄存器

SFR 寄存器				
地址	名称	读写	复位值	说明
0x84	SYS_CLK_CFG	RW	0x01	系统时钟配置寄存器

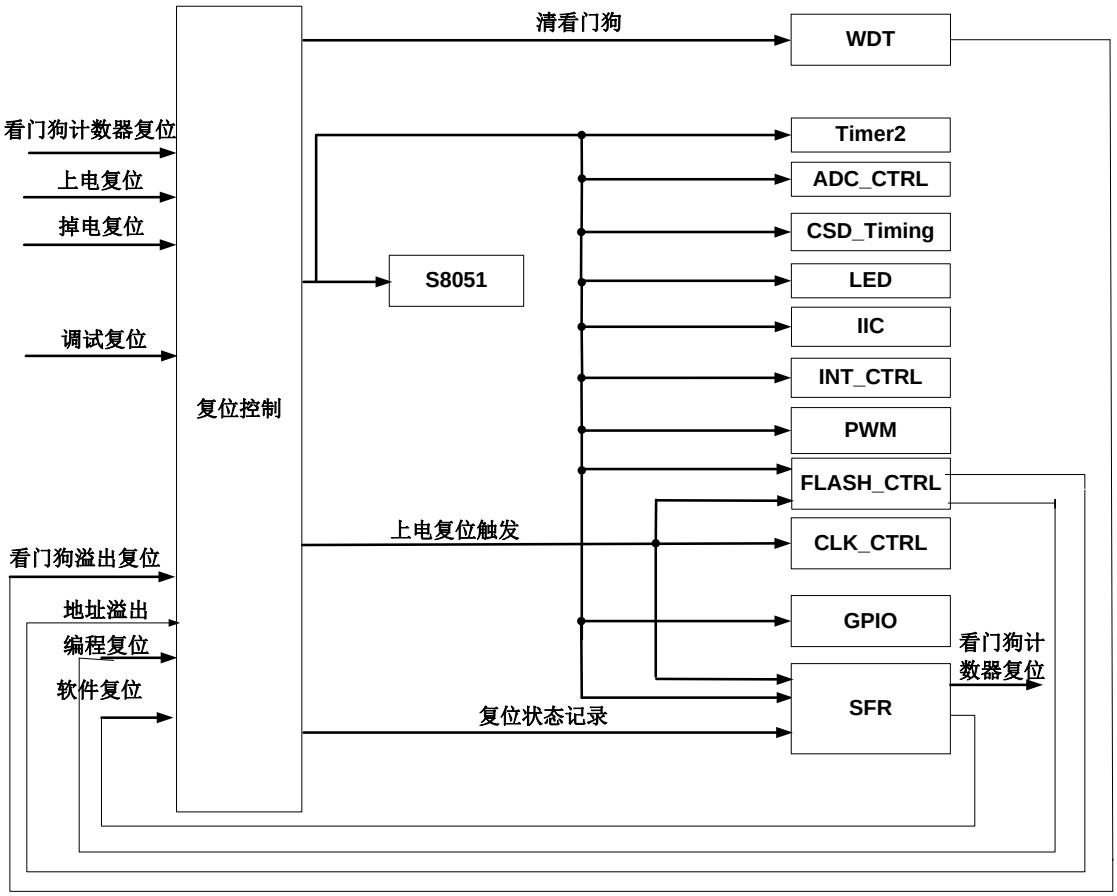
时钟设置 SFR 寄存器列表

5.3. 复位系统

GDMC191XXXX 中有 8 种复位模式：看门狗定时器溢出复位复位(WDTRST_F)、上电复位(PO_F)、掉电复位复位(BO_F)、编程复位复位(PROG_F)、修调配置复位(Debug_rst)、PC 指针溢出复位(ADDROF_F)、软件复位(SOFT_F)、看门狗计数器复位(WDT_CLR)。只要其中任意一种复位发生，系统的全局复位信号就会让整个芯片复位。可由复位标志寄存器来确定芯片进行了何种复位，复位标志位需要软件清零。

复位标记寄存器:

地址	名称	位	位名称	RW	说明	复位值
0xD7	RST_STAT	<6:0>	--	RW	复位标志位寄存器： {DEBUG_F, SOFT_F, PROG_F, ADDROF_F, BO_F, PO_F, WDTRST_F}	rst stat

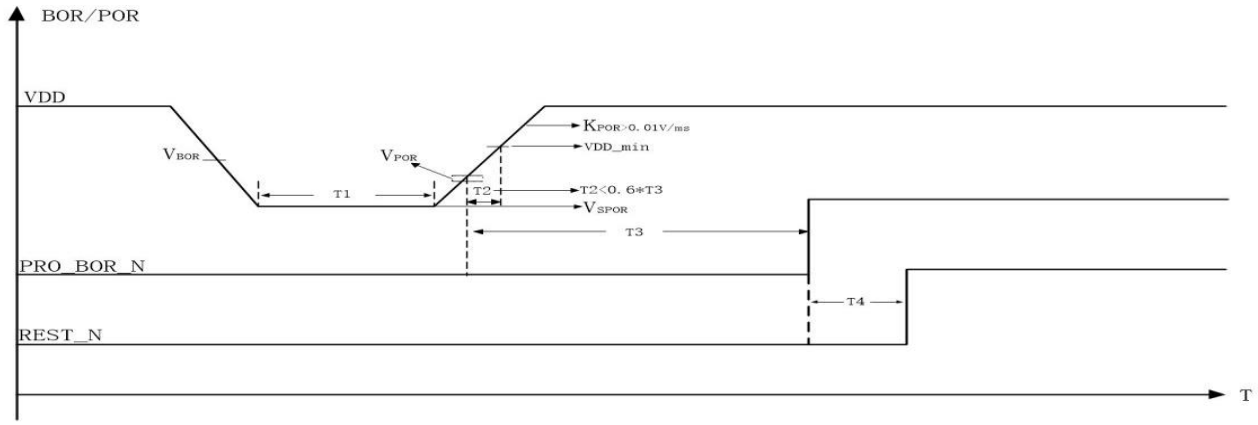


复位方框图

上电复位，系统发生上电后模拟模块产生低电平的信号并持续 93ms/4ms。上电复位为低时整个芯片处于复位状态，变高后全局复位信号继续有效 20ms 后，系统退出复位模

式。

上/掉电时序：



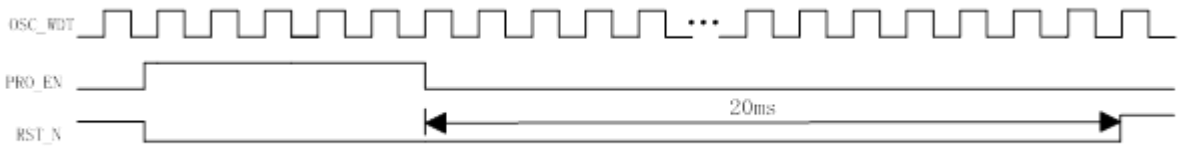
上电复位示意图

上/掉电复位参数：

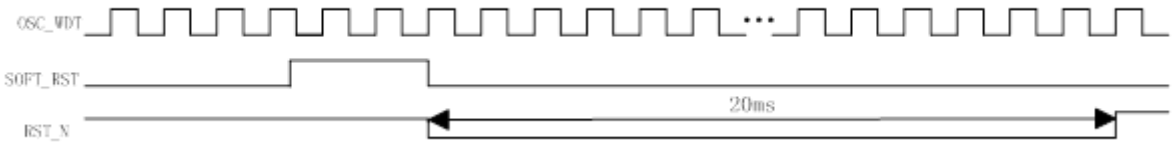
符号	参数	测试条件(VDD)	最小	典型	最大	单位
VSPOR	上电复位起始电压	—	—	—	300	mV
KPRO	上电复位电压速率	—	0.01	—	—	V/ms
VPOR	上电复位电压	—	1.1	1.5	2.2	V
VBOR	掉电复位电压(±10%)，迟滞 0.2V	—	—	VBOR	-	V
VDD_min	最小工作电压	—	2.7	-	-	V
T1	VDD 保持 VSPOR 时间	—	0.1	—	—	ms
T2	VPOR 到 VDD_min 时间	—	-	-	0.6*T3	ms
T3	复位 POR_BOR_N 持续时间	—	55	93	131	ms
T4	全局复位有效时间	—	-	20	-	ms

上电复位特性参数表

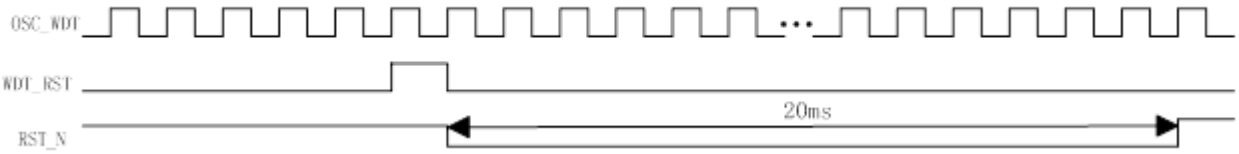
掉电复位，系统发生掉电复位后模拟模块产生低电平的信号。掉电复位信号为低时整个芯片处于复位状态，变高后全局复位信号继续有效 20ms 后，系统退出复位模式。



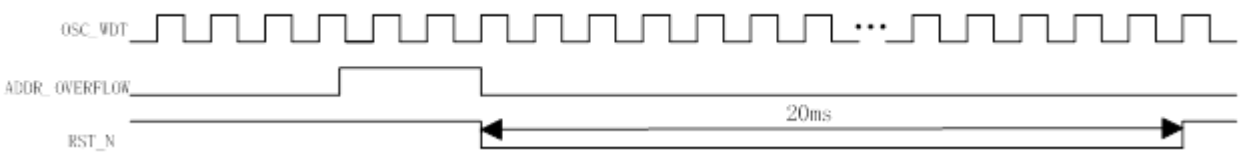
软件复位，通过写 SFR 使软复位信号有效，使全局复位信号有效 20ms，20ms 后，系统退出复位模式。



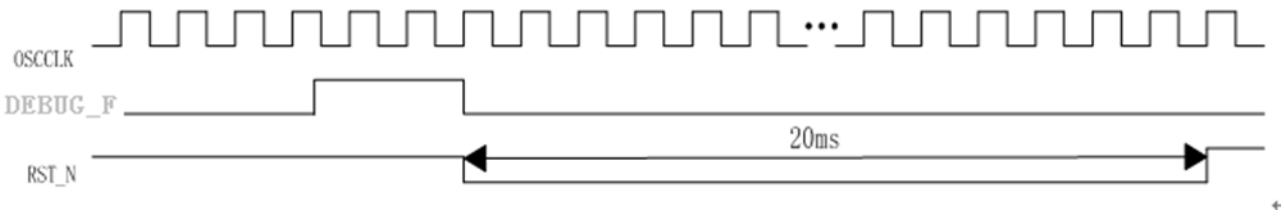
看门狗定时器溢出复位，看门狗定时器溢出后使全局复位 20ms，20ms 后，系统退出复位模式。



PC 指针溢出复位，若 MCU 寻址程序存储器时 PC 指针超出了 flash 有效的地址范围，addr_overflow 信号变高，sys_clk 时钟上升沿检测到 addr_overflow 高电平(需要 1 个时钟周期)后使全局复位 20ms，复位信号会将 addr_overflow 信号清零，20ms 后，系统退出复位模式。



修调配置复位，为核修调模块输出复位信号，低表示复位有效，芯片全局复位，但不会有 20ms 的初始化过程，仅延迟 1 个系统时钟的复位低电平。



复位相关寄存器

SFR 寄存器				
地址	名称	读写	复位值	说明
0x8E	SOFT_RST	RW	0x00	软件复位寄存器

复位 SFR 寄存器列表

复位相关寄存器详细说明

地址	名称	位	对应位名称	RW	说明	复位值
0x8E	SOFT_RST	<7:0>	--	RW	软件复位寄存器，只有在寄存器值为 0x55 时，才产生软件复位	0x00

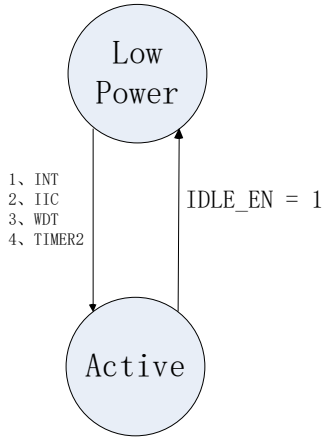
5.4. 工作模式

GDMC191XXXX 系列有 3 种工作模式，可以根据不同的情况进行选择。

正常模式(Active)：即正常工作模式，模块保持正常工作，各模块功能由软件配置控制。

低功耗 Wait 模式(Wait_mode)：核相关模块及 UART0/1 模块不工作，其余模块均可工作并通过中断来退出此模式。

低功耗 Low_power 模式(Low_power)：配置 PCON=1,此时 RC1M 和 PLL 关闭,RC32K 和 OSC32K 可配置工作/关闭(Idle/Sleep)，WDT/TIMER2 可配置工作，CPU 和其余数字模块不工作。



工作模式转换图

此外，所有模块均可以单独配置关闭门控，以此降低功耗。如在正常模式下，可配置关闭 CPU 和系统模块时钟，只使能 CSD/ADC 等模块工作；在低功耗模式下，还可配置关闭 RC32K 和 OSC32K 时钟，这样停止了所有时钟源，实现最低功耗，这时仅能通过外部中断低电平唤醒系统。

退出 Wait 模式的方式：

1. 使能 IIC、External Interrupt0、External Interrupt1、External Interrupt2、WDT、Timer2、CSD、ADC、LVDT，其中任意一种中断产生都可唤醒芯片，退出 Wait 模式，CPU 执行中断服务程序。

退出 Low_power 模式的方式：

2. 使能 IIC、External Interrupt0、External Interrupt1、External Interrupt2、WDT、Timer2，其中任意一种中断产生都可唤醒芯片，退出 low_power 模式，中断响应产生后，CPU 执行中断向量相关的中断服务程序，并在 RETI 返回指令执行后回到使 CPU 进入 low_power 模式的指令的下一条指令继续运行程序。

模式	进入该模式的条件	对时钟的影响结果	
Active、Wait	PCON=0;	RC32K	取决于软件配置
		OSC32K	取决于软件配置
		RC1M	工作
		PLL	工作
Low Power	PCON=1;	RC32K	取决于软件配置
		OSC32K	取决于软件配置
		RC1M	关闭
		PLL	关闭

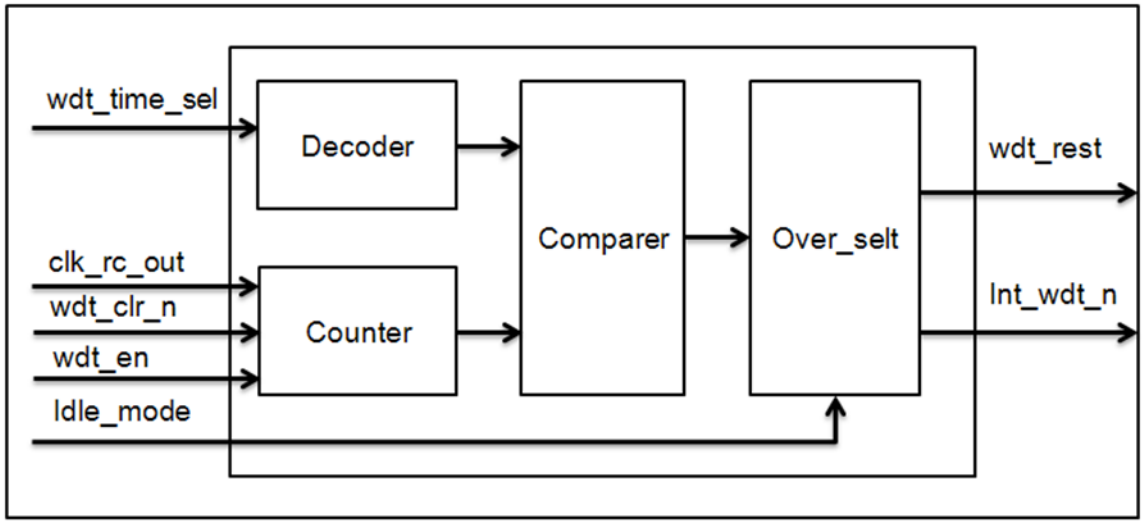
时钟源在各模式下的工作状态表

NO	Module Name	时钟源	工作状态		
			Active	Wait	Low Power
1	s8051	clk_sys12M(包含 RC1M，PLL)	√	×	×
2	UART0	clk_sys12m(包含 RC1M，PLL)	根据程序配置	×	×
3	内部 Timer0	clk_sys12M(包含 RC1M，PLL)	根据程序配置	×	×
4	内部 Timer1	clk_sys12M(包含 RC1M，PLL)	根据程序配置	×	×
5	外部 Timer2	RC32K/OSC32K	根据程序配置	根据程序配置	根据程序配置
6	外部中断	clk_sys12M(包含 RC1M，PLL)	根据程序配置	根据程序配置	根据程序配置
7	WDT	RC32K	根据程序配置	根据程序配置	根据程序配置
8	Adc_ctrl	clk_sys48m(包含 R1CM，PLL)	根据程序配置	根据程序配置	×
9	CSD_Timing	clk_sys96m(包含 R1CM，PLL)	根据程序配置	根据程序配置	×
10	IIC(S)	clk_sys16M(包含 R1CM，PLL)	根据程序配置	根据程序配置	根据程序配置

不同模式下各数字模块的状态表

5.5. WDT 看门狗

看门狗定时计数电路使用内部 RC32KHz 时钟进行定时，可 R 配置定时时间为 2^n*18ms(n=0，1，2，3，4，5，6，7)，-----此处 n 为定时配置寄存器的配置值。



由于系统应用的特殊性，对看门狗定时溢出信号分类：

在正常工作模式下，若发生看门狗定时溢出，则此时溢出信号为看门狗溢出复位信号，看门狗溢出复位影响全局复位，此时系统实现全局复位动作，并重新加载配置信息；

在 IDLE 模式下，若发生看门狗定时溢出，则此时溢出信号为看门狗中断信号，中断唤醒芯片退出 IDLE 模式并执行看门狗中断服务函数。

看门狗模块为定时计数模块，其计数时钟为 32KHz 频率的内部 RC 时钟，其定时清零信号由全局复位及配置清零构成，该信号在复位模块中由看门狗定时时钟进行同步释放处理；对于清零动作，每次 CPU 配置看门狗定时配置寄存器(WDT_CTRL)时产生，看门狗重新开始定时；同时，看门狗计数器存在看门狗计数使能控制，在计数使能有效的情况下，看门狗产生定时溢出(复位或中断)后，只要没有关闭看门狗计数使能，看门狗计数器将重新开始计数。

WDT 相关寄存器

SFR 寄存器				
地址	名称	读写	复位值	说明
0x91	WDT_CTRL	RW	0x00	看门狗溢出定时配置寄存器
0x92	WDT_EN	RW	0x00	看门狗定时使能配置寄存器

WDT SFR 寄存器列表

看门狗时钟寄存器:

看门狗用 32KHz 时钟完成定时功能可以实现从 18ms 到 2.3s 的定时。定时长度由 SFR(WDT_CTRL)控制，如下表所示：

看门狗时钟寄存器 WDT_CTRL - 0x91h

WDT_CTRL<3:0>	间隔
000	18ms
001	36ms
010	72ms
011	144ms
100	288ms
101	576ms
110	1152ms
111	2304ms

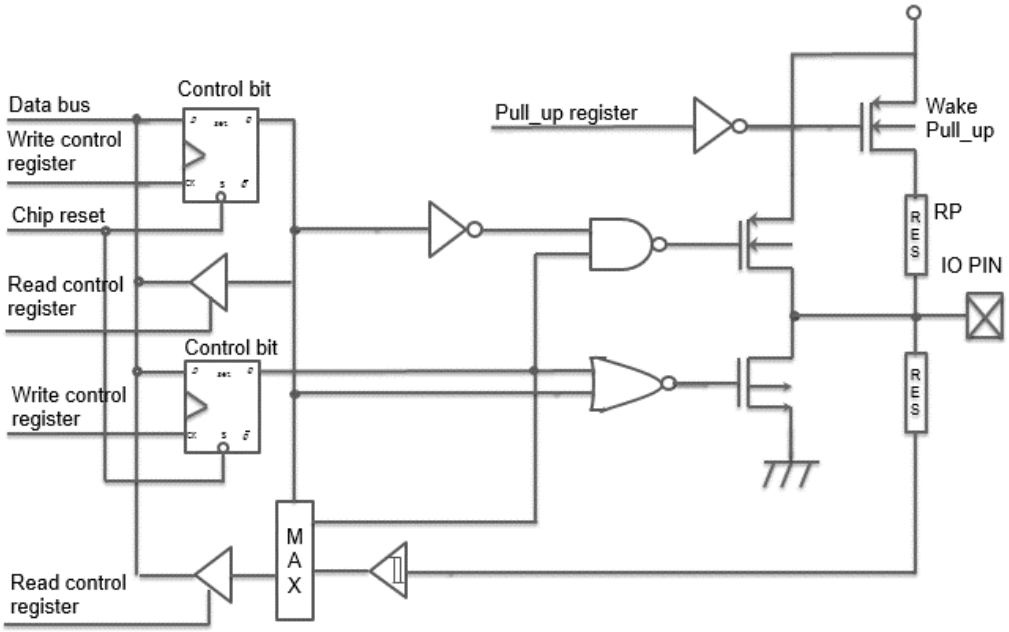
看门狗使能寄存器 WDT_EN - 0x92h

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDT_EN	WDT_EN<7:0>							
SFR	0x00							

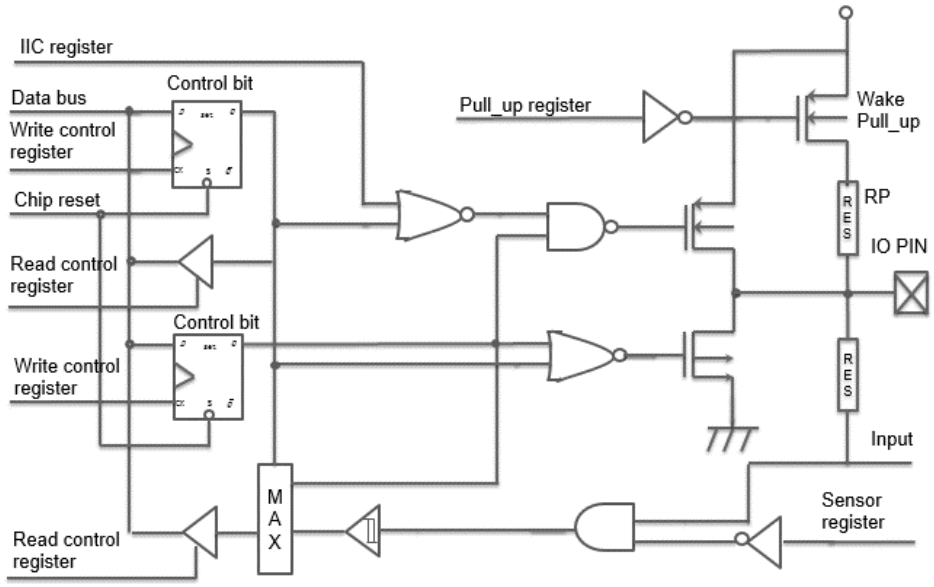
写 0x55 时关闭看门狗，写其他值开启看门狗,看门狗定时器在复位结束后一直工作。看门狗定时器清零是通过写 WDT_CTRL 寄存器完成的，无论向此寄存器中写入何值都会使看门狗定时器清零。

第 6 章 GPIO 端口

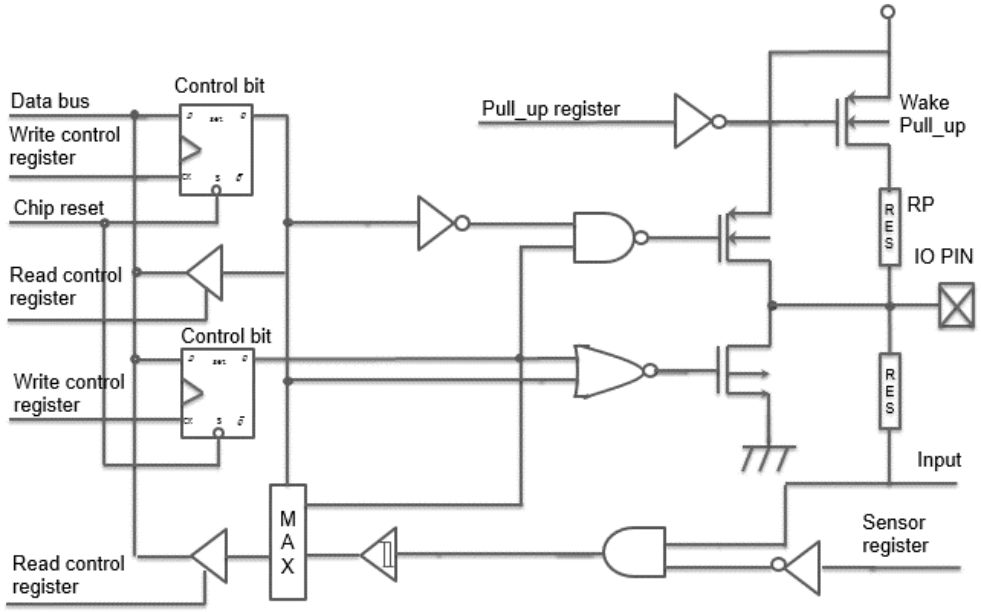
GPIO 端口的一些引脚和器件外设功能复用，同一时间不能同时配置成多种功能，否则会引起功能错乱。IIC 通信口，开漏输出，需要接上拉电阻。



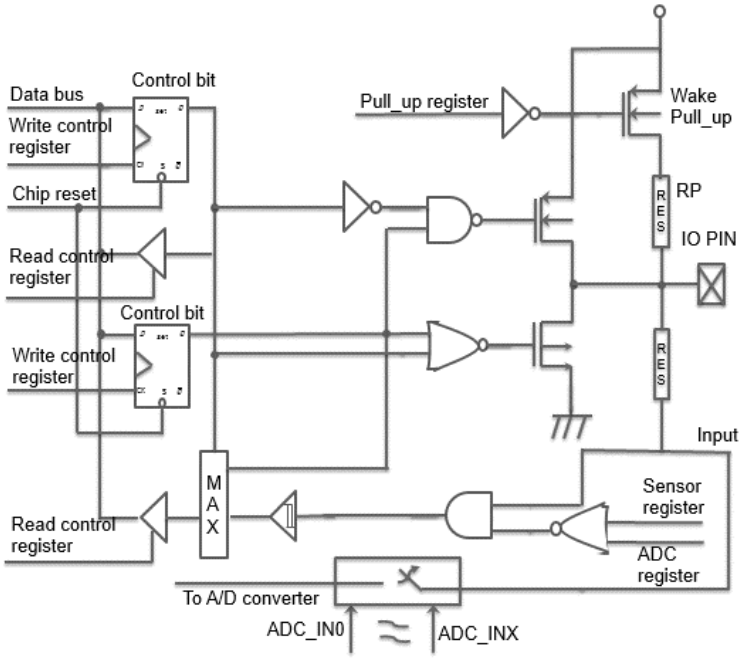
普通 IO 结构图



开漏输出 IO 结构图



SNS IO 结构图



ADC IO 结构图

TRISX 寄存器(方向寄存器)：TRISX 置 1 将对应的引脚配置为输入，清零将对应的引脚配置为输出。

DATAX 寄存器(数据寄存器)：DATAX 置 1 将对应的引脚配置输出高，清零将对应的引脚配置输出低。

PU_PX 寄存器(上拉电阻使能寄存器)：PU_PX 置 1 对应的引脚上拉电阻使能，清零对应的引脚不使能上拉电阻。

ODRAIN_EN 寄存器：ODRAIN_EN 置 1 对应的引脚使能开漏输出，清零则不使能开漏输出功能，使能 IIC 功能后自动开启开漏输出。

6.1. GPIO 相关寄存器

SFR 寄存器				
地址	名称	读写	复位值	说明
0xF8	DATAA	RW	2'h3	PA 数据寄存器
0x80	DATAB	RW	8'hff	PB 数据寄存器
0x90	DATAC	RW	8'hff	PC 数据寄存器
0x98	DATAD	RW	8'hff	PD 数据寄存器
0xEA	TRISA	RW	2'h3	PA 方向寄存器
0xEB	TRISB	RW	8'hff	PB 方向寄存器
0xEC	TRISC	RW	8'hff	PC 方向寄存器
0xED	TRISD	RW	8'hff	PD 方向寄存器

端口配置 SFR 寄存器列表

6.2. GPIO 寄存器详细说明

6.2.1. 数据寄存器

地址	名称	位	对应位名称	RW	说明	复位值
0xF8	DATAA	<1:0>	--	RW	PA 数据寄存器，可配置 PA 组 IO 口作为 GPIO 口时的输出电平，读取值为当前 IO 口（输入）的电平状态或配置输出值（输出）	2'h3
0x80	DATAB	<7:0>	--	RW	PB 数据寄存器，可配置 PB 组 IO 口作为 GPIO 口时的输出电平，读取值为当前 IO 口（输入）的电平状态或配置输出值（输出）	8'hff
0x90	DATAC	<7:0>	--	RW	PC 数据寄存器，可配置 PC 组 IO 口作为 GPIO 口时的输出电平，读取值为当前 IO 口（输入）的电平状态或配置输出值（输出）	8'hff
0x98	DATAD	<7:0>	--	RW	PD 数据寄存器，可配置 PD 组 IO 口作为 GPIO 口时的输出电平，读取值为当前 IO 口（输入）的电平状态或配置输出值（输出）	8'hff

6.2.2.方向寄存器

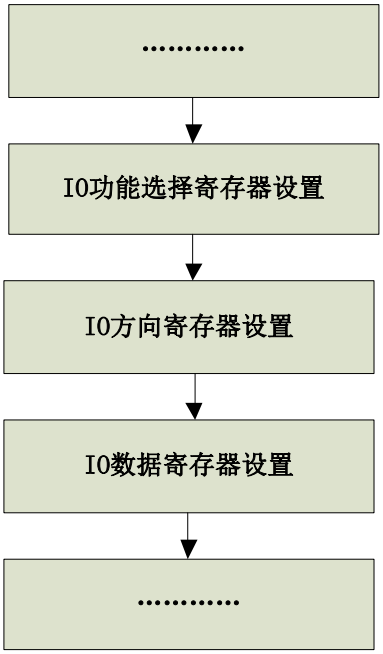
地址	名称	位	对应位名称	RW	说明	复位值
0xEA	TRISA	<1:0>	--	RW	PA 方向寄存器，0：输出，1：输入	2'h3
0xEB	TRISB	<7:0>	--	RW	PB 方向寄存器，0：输出，1：输入	8'hff
0xEC	TRISC	<7:0>	--	RW	PC 方向寄存器，0：输出，1：输入	8'hff
0xED	TRISD	<7:0>	--	RW	PD 方向寄存器，0：输出，1：输入	8'hff

6.2.3.大灌电流口

地址	名称	位	对应位名称	RW	说明	复位值
0xB0	DP_CON	<7:1>	--	R	保留	7'h0
		0	COM_MOD	RW	大电流 IO 口驱动使能 1：COM 口功能锁定，作为大电流 IO 口工作 0：COM 口功能不锁定，可通过配置为其他功能 COM 口锁定大电流 IO 口时，通过配置 GPIO 寄存器输出驱动时序。	1'h0
0xEE	COM_IO_SEL	<7:0>	--	RW	DP_CON [0] = 1 时，COM 口功能锁定，作为大电流 IO 口工作。 COM 口选择配置寄存器，对应 PB 口 1：选择 COM 口模式 0：选择 IO 口模式	8'h0

6.3. GPIO 配置流程

将端口设置为 GPIO 时，需要对以下 3 组寄存器都进行相应的设置。



IO 配置流程图

注：

1、 IO 口默认源电流驱动能力典型为 20mA,灌电流驱动能力典型为 40mA @5V 0.9VCC，当用 IO 来驱动 LED/数码管时，需要注意 LED 灯的 Ifp 电流，建议加限流电阻使 IO 驱动峰值电流限制在 LED/数码管 Ifp 电流以内。

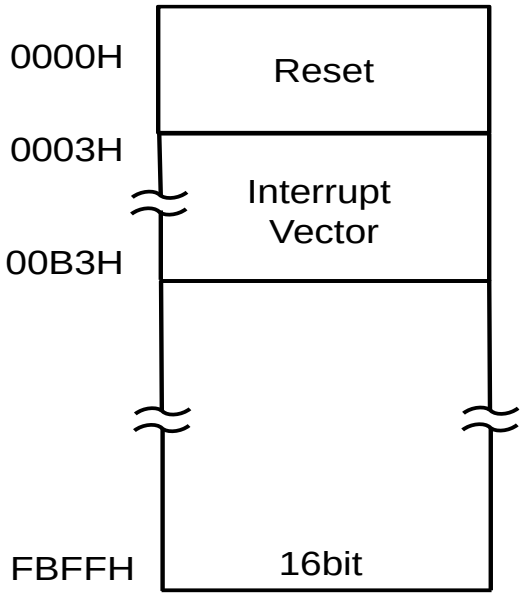
2、 PD3/PD5 IO 口在芯片上电 0V~1.3V 期间处于不定态，使用时注意 IO 上电状态变化，建议加下拉或上拉电阻。

第 7 章 中断

7.1. 中断源及入口地址

中断号	中断	触发类型	地址	中断使能	优先级	中断标志	
0	外部中断 0	下降沿	0x0003	IEN0[0]	IPL0[0]	IE0	TCON[1]
1	Timer0	高电平	0x000B	IEN0[1]	IPL0[1]	TF0	TCON[5]
2	外部中断 1	下降沿	0x0013	IEN0[2]	IPL0[2]	IE1	TCON[3]
3	Timer1	高电平	0x001B	IEN0[3]	IPL0[3]	TF1	TCON[7]
9	外部中断 2	下降沿	0x004B	IEN1[2]	IPL1[2]	IE2	IRCON1[2]
10	IIC 中断	下降沿	0x0053	IEN1[3]	IPL1[3]	IE3	IRCON1[3]
11	ADC 中断	下降沿	0x005B	IEN1[4]	IPL1[4]	IE4	IRCON1[4]
12	CSD 中断	下降沿	0x0063	IEN1[5]	IPL1[5]	IE5	IRCON1[5]
14	Wdt/Timer2 中断	下降沿	0x0073	IEN1[7]	IPL1[7]	IE7	IRCON1[7]
15	LVDT 中断	下降沿	0x007B	IEN2[0]	IPL2[0]	IE8	IRCON2[0]
16	UART0 中断	下降沿	0x0083	IEN2[1]	IPL2[1]	IE9	IRCON2[1]

中断信息列表



当芯片发生复位信号时，程序从 0x0000 地址开始执行。当发生一个中断信号时，程序将跳转到中断向量程序地址执行中断服务程序。

7.2. 中断功能

7.2.1. 中断响应

当发生中断申请时,CPU 根据中断服务程序(ISR)来确定中断的种类。CPU 完整的执行 ISR，除非有优先级高的中断源申请中断。每个 ISR 后有 RETI(中断返回)指令。执行 RETI 指令后，CPU 继续执行在中断没有发生之前的程序。

ISR 只能被优先级更高的中断申请中断。也就是，低优先级的 ISR 能被高优先级的中断申请中断。高优先级的中断 ISR 能被掉电中断中断。

GDMC191XXXX 执行完当前指令后才响应中断请求。如果正在执行的指令是 RETI 指令，或者访问 IP、IE、EIP、EIE 寄存器时，需要执行一条其他指令后才会响应中断请求。

7.2.2. 中断优先级

GDMC191XXXX 有两个中断优先级：中断级和默认优先级。中断级(最高级、高级和低级)优先于默认优先级。如果允许，掉电中断是唯一一个最高级的中断源。其他的中断源可以设置为高优先级或者低优先级。

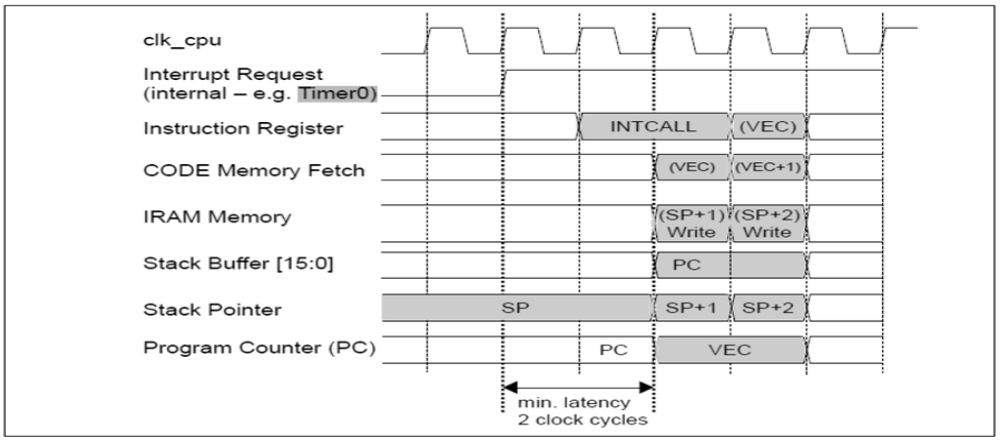
每个中断源既可以分配优先级(高或者低)，还有默认的优先级。同一级别中的中断源(例如都为高优先级)的优先级由默认的优先级决定。正在进行的中断服务程序只能被优先级高的中断请求中断。

7.2.3. 中断采样

内部定时器和串口等内部模块是通过各自的 SFR 中的中断标志位来发生中断请求。当每 1 个指令周期的第 1 个时钟周期(C1)结束时，在时钟的上升沿对外部中断进行采样。

端口外部中断是低电平有效，并且可以通过 TCON SFR 中的 IT0 位来设置选择边缘触发或者电平触发。例如，当 IT0=0 时，INT_EXT 为边缘触发，在采集到 INT_EXT 脚出现由高到低的电平变化时， 外部中断标志位置 1。

为了确保边缘触发型的中断被检测到，相应的端口要首先保持 2 个时钟的高电平，然后保持 2 个时钟的低电平。



中断采样时序图

7.2.4. 中断等待

中断的响应时间由系统当前状态决定。最快的响应时间是 5 个指令周期：1 个周期用来检测中断请求，其他 4 个用来执行长调用(LCALL)至 ISR。
当系统在执行 RETI 指令，并且后面为 MUL 或者 DIV 指令时，中断等待的时间最长(13 个指令周期)。这 13 个指令周期分别为：1 个周期用来检测中断请求，3 个用来完成 RETI 指令，5 个用来执行 DIV 或者 MUL 指令，4 个用来执行长调用(LCALL)至 ISR。在这种情况下，响应时间为 13 个时钟周期。

7.3. 中断相关寄存器

SFR 寄存器				
地址	名称	读写	复位值	说明
0x85	INT_PE_STAT	RW	0x00	WDT/Timer2 中断状态标记
0x86	INT_POBO_STAT	RW	0x00	LVDT 中断状态标记
0xA8	IEN0	RW	0x00	中断使能寄存器
0xB8	IPL0	RW	0x00	中断优先级寄存器 0
0xC0	UART_STATE	RW	0x00	串口 0 中断状态标记
0xC8	SCI_S1	RW	0x00	串口 1 中断状态标记
0xD8	SCI_INT_CLR	RW	0x00	串口 1 中断状态清除标记
0xE1	IRCON2	RW	0x00	中断标志寄存器 2
0xE6	IEN1	RW	0x00	中断使能寄存器 1
0xE7	IEN2	RW	0x00	中断使能寄存器 2
0xE8	IICSTAT	RO/RW	0x44	IIC 状态寄存器
0xF1	IRCON1	RW	0x00	中断标志寄存器 1
0xF2	PERIPH_IO_SEL	RW	0x00	外总中断使能寄存器
0xF4	IPL2	RW	0x00	中断优先级寄存器 2
0xF6	IPL1	RW	0x00	中断优先级寄存器 1
0xF7	EXT_INT_CON	RW	0x15	外部中断触发极性选择寄存器

中断 SFR 寄存器列表

7.4 中断 SFR 寄存器详细说明

地址	名称	位	对应位名称	RW	说明	复位值
0x85	INT_PE_STAT	<1>	INT_WDT_STAT	RW	WDT 中断状态标记，该位写 0 清零，写 WDT_CTRL 操作也可清零 1：中断有效 0：中断无效	1'h0
		<0>	INT_TIMER2_STAT	RW	TIMER2 中断状态标记，该位写 0 清零，写 TIMER2_CFG 操作也可清零 1：中断有效 0：中断无效	1'h0
0x86	INT_POBO_STAT	<1>	INT_PO_STAT	RW	LVDT 升压中断状态 1：升压中断有效 0：升压中断无效	1'h0
		<0>	INT_BO_STAT	RW	LVDTT 降压中断状态 1：降压中断有效 0：降压中断无效	1'h0
0xA8	IEN0	<7>	EA	RW	EA-中断允许位。EA=0 屏蔽所有的中断(EA 优先于中断源各自的中断使能位)。EA=1，中断打开， 每个中断源的中断请求是允许还是被禁止，还需由各自的允许位确定。	8'h00
		<6:4>	--	R	保留	
		<3>	ET1	RW	ET1-定时器1 溢出中断允许位。ET1=0，禁止定时器1(TF1)申请中断。ET1=1，允许TF1 标志位申请中断。	
		<2>	EX1	RW	EX1-INT_EXT1 允许位。EX1=0，禁止INT_EXT1 申请中断。 EX1=1，允许INT_EXT1 申请中断。	
		<1>	ET0	RW	ET0-定时器0 溢出中断 允许位。ET0=0，禁止定时器0(TF0)申请中断。ET0=1，允许TF0 标志位申请中断。	
		<0>	EX0	RW	EX0-INT_EXT0 允许位。EX0=0，禁止INT_EXT0 申请中断。 EX0=1，允许INT_EXT0 申请中	

					断。	
0xB8	IPL0	<7:4>	-	R	保留	8'h00
		<3>	{IPH0.3,IPL0.3}	R/W	PT1-TF1(Timer1 中断)优先级选择位。 PT1=0 时TF1(Timer1 中断)为低优先级， PT1=1 时TF1(Timer1 中断)为高优先级。	
		<2>	{IPH0.2,IPL0.2}	R/W	PX2- INT_EXT1 中断优先级选择位。PX2=0 时INT_EXT1 为低优先级， PX2=1 时INT_EXT1 为高优先级。	
		<1>	{IPH0.1,IPL0.1}	R/W	PT0-TF0(Timer0 中断)优先级选择位。 PT0=0 时TF0(Timer0 中断)为低优先级， PT0=1 时TF0(Timer0 中断)为高优先级。	
		<0>	{IPH0.0,IPL0.0}	R/W	PX0- INT_EXT0 中断优先级选择位。 PX0=0 时INT_EXT0 为低优先级， PX0=1 时INT_EXT0 为高优先级。	
0xC0	UART0_STATE	<6:0>	--	RO/RW	状态标记寄存器： bit[6]:r8，接收器的第 9 个数据，只读 bit[5]:t8，发射器的第 9 个数据 bit[4]:tx_empty_if，发送中断标记，1：发送缓存为空，0：发送缓存为满，软件写 0 清零 bit[3]:rx_full_if，接收中断标记，1：接收缓存为满，0：接收缓存为空，软件写 0 清零 bit[2]:rx_overflow_if，接收溢出标记，1：接收溢出(新数据丢失)，0：没有溢出，软件写 0 清零 bit[1]:frame_err_if，帧错误标记，1：检测到帧错误，0：未	7'h00

					检测到帧错误，软件写 0 清零 bit[0]:parity_err_if，奇偶校验错误标记，1：接收器奇偶校验错误，0：奇偶校验正确，软件写 0 清零	
0xC8	SCI_S1	<7:0>	--	RO	中断状态标记寄存器： bit[7]:tx_empty_if，发送缓存空中断标记，1：发送缓存为空，0：发送缓存为满，只读 bit[6]:tx_finish_if，发送完成中断标记，1：发送完成，发射器闲置，0：发射器正在工作，只读 bit[5]:rx_full_if，接收满中断标记，1：接收缓存为满，0：接收缓存为空，只读 bit[4]:idle_if，闲置线路中断标记，1：检测到闲置线路，0：未检测到闲置线路，只读 bit[3]:rx_overflow_if，接收溢出标记，1：接收溢出(新数据丢失)，0：没有溢出，只读 bit[2]:noise_err_if，噪声标记，1：检测到噪声，0：未检测到噪声，只读 bit[1]:frame_err_if，帧错误标记，1：检测到帧错误，0：未检测到帧错误，只读 bit[0]:parity_err_if，奇偶校验错误标记，1：接收器奇偶校验错误，0：奇偶校验正确，只读	8'h00
0xD8	SCI_INT_CLR	<7:0>	--	W	SCI 模块中断清除寄存器 bit[7]:clr_tx_empty_if，发送缓存空中断清除位，该位写 1 清除相应中断，写 0 无效 bit[6]:clr_tx_finish_if，发送完成中断清除位，该位写 1 清除相应中断，写 0 无效	8'h00

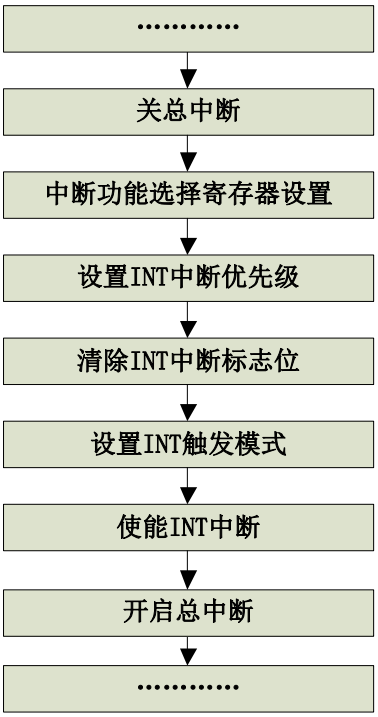
					bit[5]:clr_rx_full_if，接收满中断清除位，该位写 1 清除相应中断，写 0 无效 bit[4]:clr_idle_if，闲置线路中断清除位，该位写 1 清除相应中断，写 0 无效 bit[3]:clr_rx_overflow_if，接收溢出标记清除位，该位写 1 清除相应标记，写 0 无效 bit[2]:clr_noise_err_if，噪声标记清除位，该位写 1 清除相应标记，写 0 无效 bit[1]:clr_frame_err_if，帧错误标记清除位，该位写 1 清除相应标记，写 0 无效 bit[0]:clr_parity_err_if，奇偶校验错误标记清除位，该位写 1 清除相应标记，写 0 无效	
0xE1	IRCON2	<7:3>	--	R	保留	8'h00
		<2>	IE10	R	保留	
		<1>	IE9	RW	UART0 中断标志	
		<0>	IE8	RW	LVDT 中断标志	
0xE6	IEN1	<7>	EX7	RW	WDT/Timer2 中断使能	8'h00
		<6>	EX6	R	保留	
		<5>	EX5	RW	CSD 中断使能	
		<4>	EX4	RW	ADC 中断使能	
		<3>	EX3	RW	IIC 中断使能	
		<2>	EX2	RW	外部中断2 中断使能	
		<1:0>	-	R	保留	
0xE7	IEN2	<7:3>	--	R	保留	8'h00
		<2>	EX10	R	保留	
		<1>	EX9	RW	UART0 中断使能	
		<0>	EX8	RW	LVDT 中断使能	
0xE8	IICSTAT	<7>	IIC_START	R	开始信号标志位 1=表示检测到了启动位； 0=表示未检测到启动位	0

		<6>	IIC_STOP	R	停止信号标志位 1=表示处于停止状态； 0=表示未检测到停止位	1
		<5>	IIC_RW	R	读写标志位 记录最近一次地址匹配后，从地址字节中获得的读/写信息， 1=表示读操作； 0=表示写操作	0
		<4>	IIC_AD	R	地址数据标志位 1=表示最近接收或者发送的字节是数据； 0=表示最近接收或者发送的字节是地址	0
		<3>	IIC_BF	R	IICBUF 满标志位 在 IIC 总线方式下接收时： 1=表示接收成功，缓冲器已经满； 0=表示接收未完成，缓冲器还为空 在 IIC 总线方式下发送时： 1=表示数据发送正在进行(不包括应答位和停止位)，缓冲器还是满的； 0=表示数据发送已经完成(不包括应答位和停止位)，缓冲器已空	0
		<2>	IIC_ACK	R	应答标志位 1：表示无效的应答信号； 0：表示有效的应答信号	1
		<1>	IIC_WCOL	RW	写冲突标志位 1=表示 IIC 正在发送当前的数据的时候，新的数据试图写入发送缓冲器；新的数据是不能被写入缓冲器的； 0=未发生写冲突	0
		<0>	IIC_RECOV	RW	接收溢出标志位 1=表示 IIC 接收的前一个数据还没有取走时，又接收到了新的数据，新的数据是不能被缓冲器接收的 ；	0

					0=表示未发生接收溢出	
0xF1	IRCON1	<7>	IE7	RW	WDT/Timer2 中断标志	8'h00
		<6>	IE6	R	保留	
		<5>	IE5	RW	CSD 中断标志	
		<4>	IE4	RW	ADC 中断标志	
		<3>	IE3	RW	IIC 中断标志	
		<2>	IE2	RW	外部中断1中断标志	
		<1:0>	-	R	保留	
0xF2	PERIPH_IO_SEL	<6>	IIC_AFIL_SEL	RW	IIC 口模拟滤波选择使能 1：选择模拟滤波功能 0：不选择模拟滤波功能	1'h1
		<5>	IIC_DFIL_SEL	RW	IIC 口数字滤波选择使能 1：选择数字滤波功能 0：不选择数字滤波功能	1'h0
		<4:3>	UART0_IO_SEL	RW	UART0 口选择使能 00：PA0/1 口选择 UART0 功能 01：PB3/4 口选择 UART0 功能 1x：PD4/5 口选择 UART0 功能	2'h0
		<2>	INT2_IO_SEL	RW	INT2 口选择使能，对应 PD7 1：选择 INT2 功能 0：不选择 INT2 功能	1'h0
		<1>	INT1_IO_SEL	RW	INT1 口选择使能，对应 PD6 1：选择 INT1 功能 0：不选择 INT1 功能	1'h0
		<0>	INT0_IO_SEL	RW	INT0 口选择使能，对应 PD0 1：选择 INT0 功能 0：不选择 INT0 功能	1'h0
0xF4	IPL2	<7:3>	--	R	保留	8'h00
		<2>	{IPH2.2, IPL2.2}	R	保留	
		<1>	{IPH2.1, IPL2.1}	RW	UART0 中断优先级 1:为高 0:为低	
		<0>	{IPH2.0, IPL2.0}	RW	LVDT 中断优先级 1:为高 0:为低	

0xF6	IPL1	<7>	{IPH1.7, IPL1.7}	RW	WDT/Timer 2 中断优先级 1:为高 0:为低	8'h00
		<6>	{IPH1.6, IPL1.6}	R	保留	
		<5>	{IPH1.5, IPL1.5}	RW	CSD 中断优先级 1:为高 0:为低	
		<4>	{IPH1.4, IPL1.4}	RW	ADC 中断优先级 1:为高 0:为低	
		<3>	{IPH1.3, IPL1.3}	RW	IIC 中断优先级 1:为高 0:为低	
		<2>	{IPH1.2, IPL1.2}	RW	外部中断优先级 1:为高 0:为低	
		<1:0>	--	R	保留	
0xF7	EXT_INT_CON	<5:4>	INT2_POLARITY	RW	外部中断 2 触发极性选择： INT2_POLARITY=01：下降沿(低功耗模式下低电平唤醒) INT2_POLARITY=10：上升沿(低功耗模式下高电平唤醒) INT2_POLARITY=00/11：双沿(低功耗模式下低电平唤醒)	2'h1
		<3:2>	INT1_POLARITY	RW	外部中断 1 触发极性选择： INT1_POLARITY=01：下降沿(低功耗模式下低电平唤醒) INT1_POLARITY=10：上升沿(低功耗模式下高电平唤醒) INT1_POLARITY=00/11：双沿(低功耗模式下低电平唤醒)	2'h1
		<1:0>	INT0_POLARITY	RW	外部中断 0 触发极性选择： INT0_POLARITY=01：下降沿(低功耗模式下低电平唤醒) INT0_POLARITY=10：上升沿(低功耗模式下高电平唤醒) INT0_POLARITY=00/11：双沿(低功耗模式下低电平唤醒)	2'h1

7.5.外部中断配置流程



INT0/1/2 配置流程图

第 8 章 定时器 Timer

GDMC191XXXX 系列包含核内部的 3 个定时器（Timer0、Timer1、Timer2）。每个 Timer 包含一个 16 位的寄存器。在被访问时以两个字节的形式出现：一个低字节（TL0 或 TL1）和一个高字节（TH0 或 TH1）。Timer2 的寄存器为低字节 TIMER2_SET_L，高字节 TIMER2_SET_H。

Timer 的功能特点如下：

- Timer0 连接系统时钟，系统时钟 sys_clk/12；
- Timer1 连接系统时钟，系统时钟 sys_clk/12
- Timer2 可选内部 RC 或外部晶振时钟，频率 32768Hz/4MHz。
- Timer0 支持 8bits 自动重载定时/计数，16bits 手动重载定时/计数功能。
- Timer1 支持 8bits 自动重载定时/计数，16bits 手动重载定时/计数功能。

  Timer2 支持 32bits 自动重载定时和手动重载定时，支持中断唤醒功能。

8.1. Timer0 和 Timer1

定时器 0/1 有四种运行模式，由 TMOD SFR 和 TCON SFR 控制。

定时器 0/1 四种模式如下：

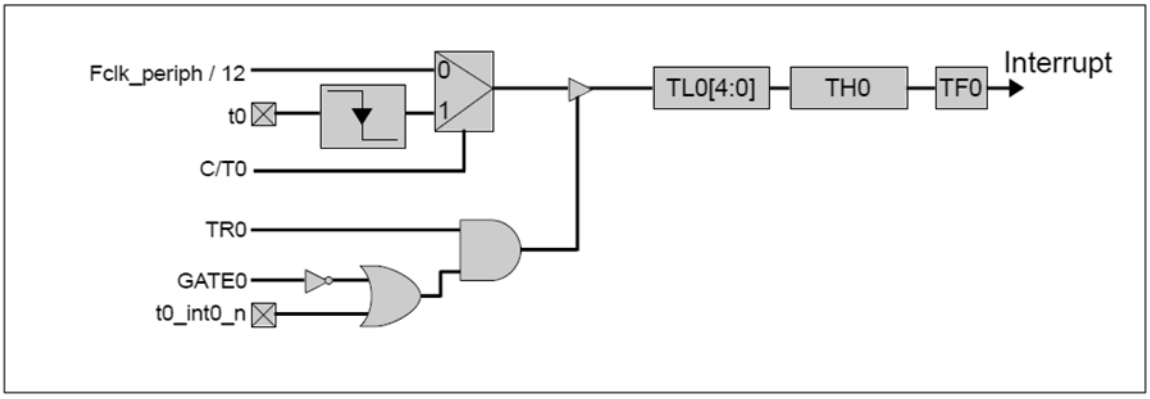
●13 位定时器/计数器 (模式 0)

●16 位定时器/计数器 (模式 1)

●自动重载初值的 8 位计数器(模式 2)

●两个 8 位计数器 (模式 3，只用于定时器/计数器 0)

模式 0 :13 位定时器/计数器



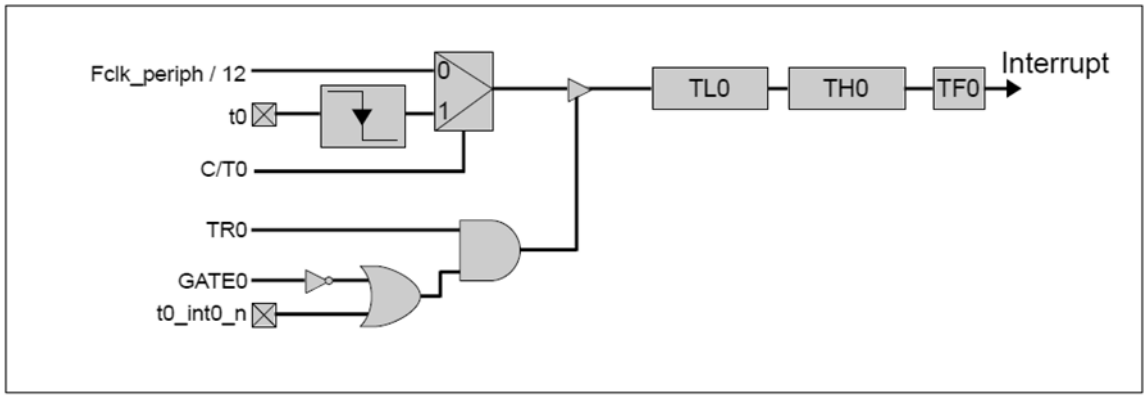
模式 0 逻辑逻辑结构图

在模式 0 下定时器 0 和定时器 1 的工作过程相同，如图所示。在模式 0 中，定时器为 13 位的计数器，其 0-4 位为 TL0(或者 TL1)，另外 8 位为 TH0(或者 TH1)。TCON 寄存器中的使能位(TR0/TR1)来控制定时器的开启和关闭。

定时器对选定的系统时钟源(sys_clk/12)进行计数，当 13 位计数器计数累积到全 1 时，计数器清 0(全 0)，并且 TF0(或者 TF1)置位。在模式 0 中，TL0(或者 TL1)的高 3 位是不确定

的，在读计数值时应屏蔽掉或忽略这 3 位。t0/t1、C/T0/CT1 均为 0，t0_int0_n/t1_int1_n 均为 1，计数使能仅由 TR0/1 决定。

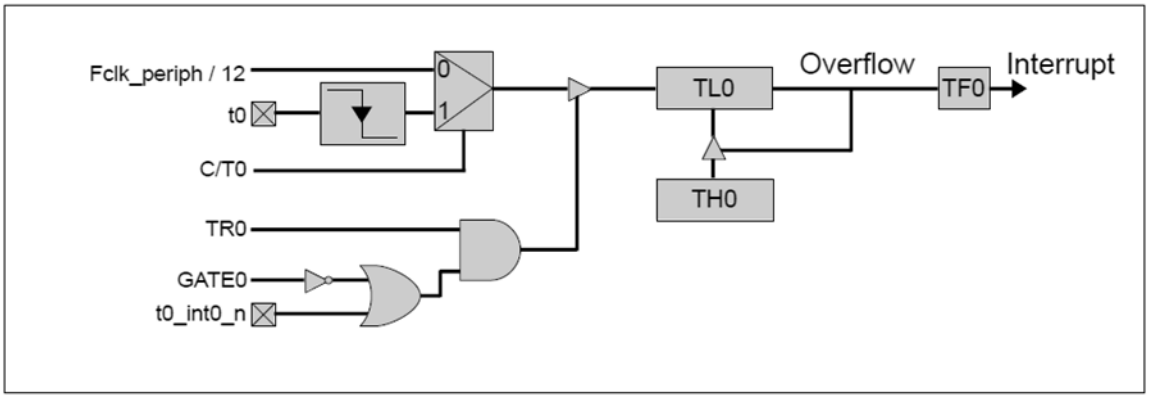
模式 1 :16 位定时器/计数器



模式 1 逻辑逻辑结构图

定时器 0 和定时器 1 的模式 1 是相同的，如图所示。在模式 1 中，定时器为 16 位的计数器。LSB 寄存器(TL0 或者 TL1)的所有 8 位都被使用。当计数器计数累计至 0xFFFF 时，计数器清为全 0。除此之外，模式 1 和模式 0 是相同的。t0/t1、C/T0/CT1 均为 0，t0_int0_n/t1_int1_n 均为 1，计数使能仅由 TR0/1 决定。

模式 2:自动重载初值的 8 位计数器

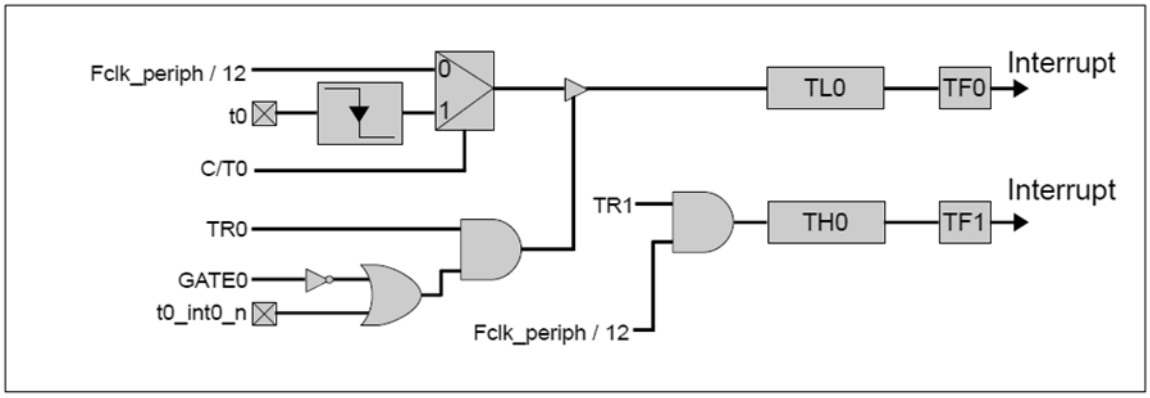


模式 2 逻辑逻辑结构图

定时器 0 和定时器 1 的模式 2 是相同的。在模式 2 中，定时器为一个带有自动重载初值的 8 位计数器。这个计数器就是 LSB 寄存器(TL0 或者 TL1)，需要重载的初值保存在 MSB 寄存器(TH0 或者 TH1)中。

如图所示，模式 2 的计数器控制和模式 0、模式 1 是一样的。但是，在模式 2 中，当 TLn 累计至 FFh，保存在 THn 中的值重载至 TLn。t0/t1、C/T0/CT1 均为 0，t0_int0_n/t1_int1_n 均为 1，计数使能仅由 TR0/1 决定。

模式 3 :两个 8 位计数器



模式 3 逻辑逻辑结构图

在模式 3 中，定时器 0 为两个 8 位的计数器，此时定时器 1 停止计数并且保存它的值。如图 5 所示，TL0 是由定时器 0 的控制位来控制的 8 位寄存器。计数器用 GATE 作为使能端来控制 INT_EXT 信号接收。

TH0 的是一个单独的 8 位计数器。TH0 只能用来计算时钟周期(12 分频)。定时器 1 的控制位和标志位(TR1 和 TF1)用来作为 TH0 的控制位和标志位。

当定时器 0 工作在模式 3 时，定时器 1 的使用受到限制，因为定时器 0 用到了定时器 1 的控制位(TR1)和中断标志位(TF1)。定时器 1 仍然能用来产生波特率，并且定时器 1 在 TL1 和 TH1 寄存器中的值依然有效。

当定时器 0 工作在模式 3 时，通过定时器 1 的模式位来控制定时器 1。要开启定时器 1，需要将定时器 1 设置为模式 0、1 或者 2。要关闭定时器 1，将定时器 1 的模式设置为 3。定时器 1 可以作为定时器(时钟为 clk/12)，但是由于 TR1 和 TF1 被借用，不能产生溢出中断。当定时器 0 工作在模式 3 时，定时器 1 的 GATE 有效。t0/t1、C/T0/CT1 均为 0，t0_int0_n/t1_int1_n 均为 1，计数使能仅由 TR0/1 决定。

8.1.1. Timer0/1 相关寄存器

SFR 寄存器				
地址	名称	读写	复位值	说明
0x88	TCON	RW	0x05	定时器控制寄存器
0x89	TMOD	RW	0x00	定时器模式寄存器
0x8A	TL0	RW	0x00	定时器 0 计时器低 8 位
0x8B	TL1	RW	0x00	定时器 1 计时器低 8 位
0x8C	TH0	RW	0x00	定时器 0 计时器高 8 位
0x8D	TH1	RW	0x00	定时器 1 计时器高 8 位

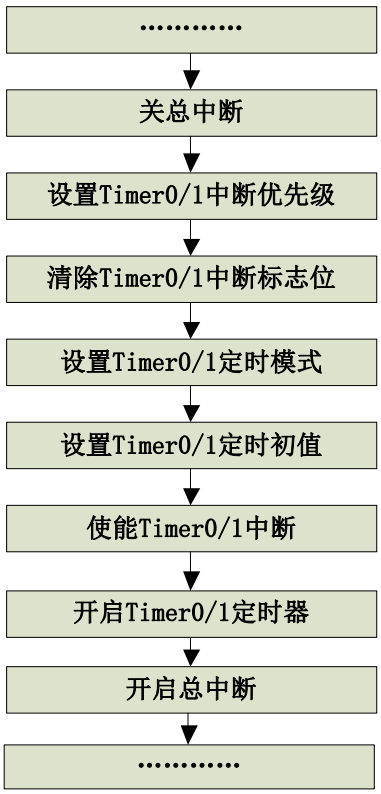
Timer0/1 SFR 寄存器列表

8.1.2. Timer0/1 寄存器详细说明

地址	名称	位	对应位名称	RW	说明	复位值
0x88	TCON	<7>	TF1	RW	定时器 1 溢出标志位，当 Timer1 溢出时硬件置 1，或者 Timer0 的 TH0 在模式三下溢出。	0x05
		<6>	TR1	RW	Timer1 启动使能，设置为 1 时启动 Timer1 或启动 Time0 模式三时 TH0 计数。	
		<5>	TF0	RW	定时器 0 溢出标志位，当 Timer0 溢出时硬件置 1。	
		<4>	TR0	RW	Timer0 启动使能，设置为 1 时启动 Timer0 计数。	
		<3>	IE1	RW	外部中断 1 标志位，硬件置 1，可软件清 0。	
		<2>	IT1	R	保留	
		<1>	IE0	RW	外部中断 0 标志位，硬件置 1，可软件清 0。	
		<0>	IT0	R	保留	
0x89	TMOD	<7>	GATE1	R	保留	0x00
		<6>	C/T1	R	保留	
		<5:4>	M1[1:0]	RW	M1-定时器 1 模式选择 Bit 1，M0-定时器 1 模式选择 Bit 0， M1M0： 00=模式 0 - 13 位定时器/计数器 01=模式 1 - 16 位定时器/计数器 10=模式 2 - 自动重载初值的 8 位计数器 11=模式 3 - 两个 8 位计数器	
		<3>	GATE0	R	保留	
		<2>	C/T0	R	保留	
		<1:0>	M0[1:0]	RW	M1-定时器 0 模式选择 Bit 1，M0-定时器 1 模式选择 Bit 0，	

					M1M0： 00=模式 0 - 8 位定时器/计数器 01=模式 1 - 16 位定时器/计数器 10=模式 2 - 自动重载初值的 8 位计数器 11=模式 3 - 两个 8 位计数器	
0x8A	TL0	<7:0>	TL0	RW	定时器 0 计时器低 8 位	0x00
0x8B	TL1	<7:0>	TL1	RW	定时器 1 计时器低 8 位	0x00
0x8C	TH0	<7:0>	TH0	RW	定时器 0 计时器高 8 位	0x00
0x8D	TH1	<7:0>	TH1	RW	定时器 1 计时器高 8 位	0x00

8.1.3. Timer0/1 配置流程



Timer0/1 配置流程图

8.2. Timer2

TIMER2 模块起定时作用，其内部主要结构为一个 32 位的计数器，通过对输入时钟的计数达到定时的功能，TIMER2 的计数原则为累加计数，当计数器计数到设定值时产生中断；TIMER2 的计数时钟选择内部 RC 时钟 32KHz；TIMER2 有两种工作模式：单次定时模式和自动重装载模式，无论哪种模式，计时完成均会产生中断。

通过寄存器 TIMER2_EN 配置 Timer2 的功能使能，TIMER2_RLD 配置自动重载模式或手动重载模式，定时时间由寄存器 TIMER2_SET_L 和 TIMER2_SET_H 决定。Timer2 支持中断唤醒 wait 模式和低功耗模式功能，在中断处理函数中需要软件清除中断标记。

Timer2 定时时长公式：

TIMER2_CNT_MOD=0：

$T_{TIMER2}=T_{TIMER2\_CLK}*(\{TIMER2\_SET\_H,\ TIMER2\_SET\_L\}+1)$

TIMER2_CNT_MOD=1：

$T_{TIMER2}=65536*T_{TIMER2\_CLK}*(\{TIMER2\_SET\_H,\ TIMER2\_SET\_L\}+1)$

注：T_{TIMER2_CLK} = 1/32KHz (s)

注意： 1、任意配置 TIMER2_SET_H、TIMER2_SET_L、TIMER2_CFG 均可清零计数器；

8.2.1. Timer2 相关寄存器

SFR 寄存器				
地址	名称	读写	复位值	说明
0x93	TIMER2_CFG	RW	0x00	TIMER2 配置寄存器
0x94	TIMER2_SET_H	RW	0x00	TIMER2 计数值配置寄存器，高 8 位
0x95	TIMER2_SET_L	RW	0x00	TIMER2 计数值配置寄存器，低 8 位

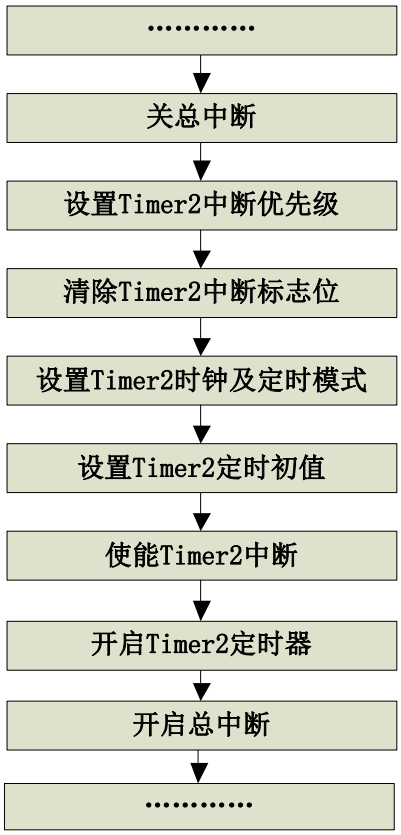
Timer2 SFR 寄存器列表

8.2.3. Timer2 寄存器详细说明

地址	名称	位	对应位名称	RW	说明	复位值
0x93	TIMER2_CFG	<3>	TIMER2_CNT_MOD	RW	Timer2 计数步进模式选择寄存器 1：计数步进为 65536 个时钟 0：计数步进为一个时钟	1'h0
		<2>	TIMER2_CLK_SEL	RW	Timer2 时钟选择寄存器 1:保留 0:选择 clk_rc	1'h0
		<1>	TIMER2_RLD	RW	TIMER2 自动重载使能寄存器 1：自动重载模式 0：手动重载模式	1'h0
		<0>	TIMER2_EN	RW	TIMER2 计数使能寄存器 配置 1 开启定时，配置 0 停止定时 在手动重载模式下会在计数完成后硬件自动清零该寄存器，停止计数，在自动重载模式下会在计数完成后维持该使能寄存器，自动重新从零计数，无论哪	1'h0

					种模式，计数过程中配置该寄存器为 1 均会开始从零计数	
0x94	TIMER2_SET_H	<7:0>	--	RW	TIMER2 计数值配置寄存器，高 8 位，扫描过程中配置该寄存器会重新计数	8'h00
0x95	TIMER2_SET_L	<7:0>	--	RW	TIMER2 计数值配置寄存器，低 8 位，扫描过程中配置该寄存器会重新计数	8'h00
0x96	REG_ADDR	<5:0>	--	RW	Timer2 二级总线地址配置寄存器	6' h00
0x97	REG_DATA	<7:0>	--	RW	Timer2 二级总线地址数据读写寄存器， 1、REG_ADDR = 0x1f; REG_DATA = 0x01; REG_ADDR = 0x00;为选择外部晶振 4MHz 起振； 2、REG_ADDR = 0x1f; REG_DATA = 0x00; REG_ADDR = 0x00;为选择外部晶振 32768Hz 起振；	8' h00

8.2.2. Timer2 配置流程



Timer2 配置流程图

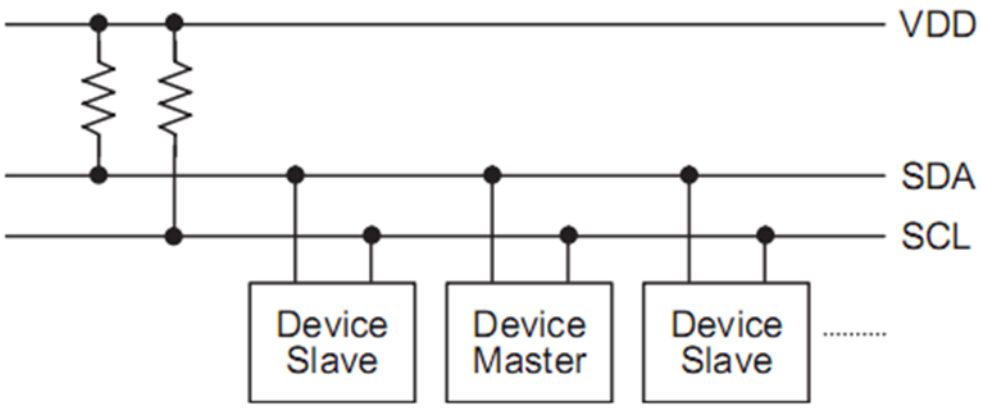
1. 首先配置定时设定值寄存器 TIMER2_SET_H/TIMER2_SET_L 和步进配置 TIMER2_CNT_MOD;
2. 然后根据需要配置自动重装载使能寄存器 TIMER2_RLD，若需自动循环计数则设为 1，否则配置为 0;
3. 最后在配置定时使能寄存器 TIMER2_EN，开启定时配置 TIMER2_EN=1;
4. 停止计时：TIMER2_EN=0。

注： 1.TIMER2_EN=0x1 操作要放在所有配置的最后。
2.在 TIMER2 计时期间，严禁改变相关配置，若要修改，则需先停止计时。
3.如需精确计时，在自动重载模式下，中断处理中不允许配置 TIMER2 的三个寄存器。

第 9 章 IIC

GDMC191XXXX 支持标准 IIC 通信和快速 IIC 通信，具有以下特点：

- 两条串行接口：串行数据线 SDA 和串行时钟线 SCL；
- 符合 philips 的标准通信协议；
- 传输速率：100Kbps、400Kbps；
- 支持 7 位地址寻址；
- 具有延长时钟低电平的功能；
- 在 low_power 模式下可以通过 IIC 中断唤醒核；
- 检测写冲突和缓存 BUF 溢出异常的情况。



IIC 主从机连接图

主机和从机之间由 SCL(串行时钟)线、SDA(串行数据)线连接，IIC 通信模式时，PA0/1 为开漏，SCL、SDA 必须接上拉电阻(建议 4.7K~10K)。当 TS 设备有触摸相关的动作，比如触摸、滑动、手指离开等姿势发生时，主机可通过 IIC 通信来读取从机的触摸状态。

9.1. IIC 相关寄存器

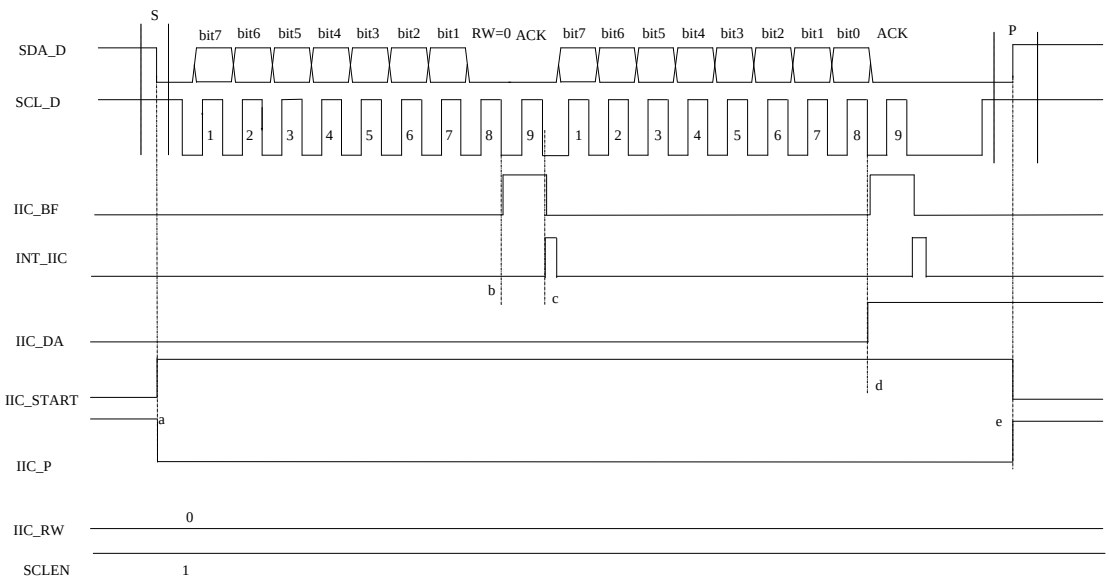
SFR 寄存器				
地址	名称	读写	复位值	说明
0xE3	IICADD	RW	0x00	IIC 地址寄存器
0xE4	IICBUF	RW	0x00	IIC 发送接收数据寄存器
0xE5	IICCON	RW	0x10	IIC 配置寄存器
0xE8	IICSTAT	RO/RW	0x44	IIC 状态寄存器
0xE9	IICBUFFER	RW	0x00	IIC 发送数接收据缓存寄存器
0xF2	PERIPH_IO_SEL <6:5>	RW	7’0x40	IIC 滤波选择使能

IIC SFR 寄存器列表

9.2. 通信时序

GDMC191XXXX 采用硬件从机。当主机读/写数据时，从机接收到地址后，如果地址匹配，则产生中断，发送有效应答信号。并在主机写数据的第八个时钟后产生中断，主机发送停止信号时将不会产生中断信号。下面是 IIC 通信的简单时序图：

IIC 主机写时序描述



主机写不拉低时钟线图

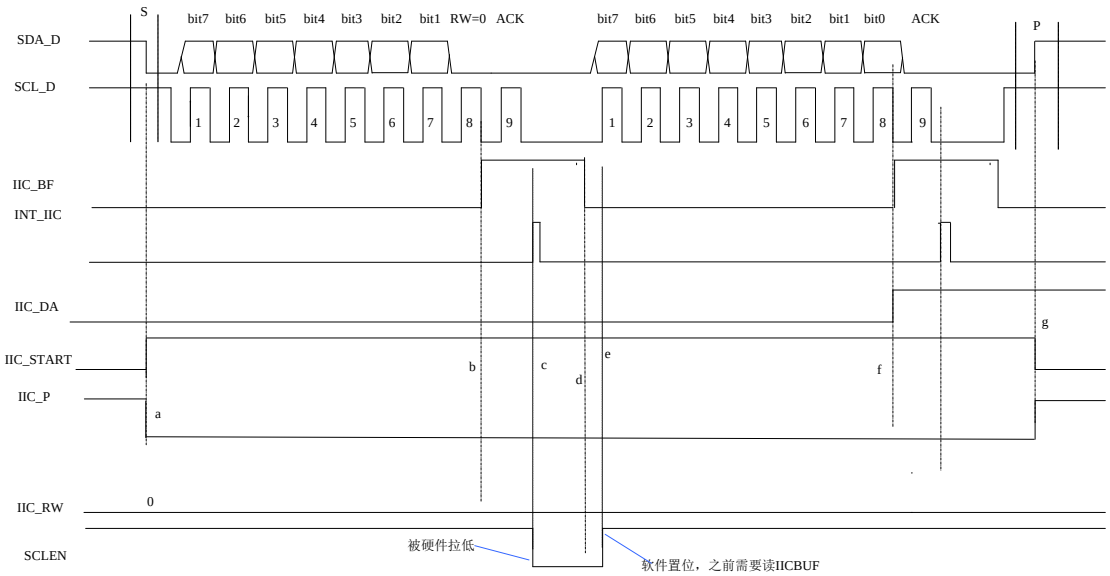
如上图所示是主机写操作时不拉低时钟线的示意图，从中可以看到 IIC 总线的变化情况以及一些内部信号的变化。

首先，主机发送启动信号 IIC_START，从机检测到 IIC_START 信号之后置位 IIC_START 状态位，如图中虚线 a 所示。

然后,主机发送地址字节和读写标志位，从机在接收到地址字节后硬件自动与自身地址比较，如果匹配则在第 8 个时钟的下降沿之后置位 IIC_BF，如虚线 b 所示。在第 9 个时钟的下降沿之后会产生中断信号 INT_IIC，如虚线 c 所示，MCU 执行中断子程序期间需要读取 IICBUF，即使此数据没有用，读取 IICBUF 的操作会使 STAT_BF 间接的清零。主机继续发送数据，在第 2 个字节的第 8 个时钟的下降沿之后 IIC_BF 同样被置位，同时 IIC_AD 标志位也会置位，标志当前接收到的字节是数据，如虚线 d 所示，停止信号对 IIC_AD 标志位没有影响，即检测到停止信号 IIC_STOP，IIC_AD 标志位不会被清零；第 9 个时钟的下降沿之后同样会产生中断，中断子程序需要作同样的操作。如果主机要发送多个字节，则可以继续发送，上图中仅仅示意主机发送一个数据的情况。

最后，主机在发送完毕所有的数据之后发送一个停止信号 IIC_STOP，标志着通信的结束，释放 IIC 总线，总线进入空闲状态。

IIC 主机写拉低时序描述



主机写拉低时钟线图

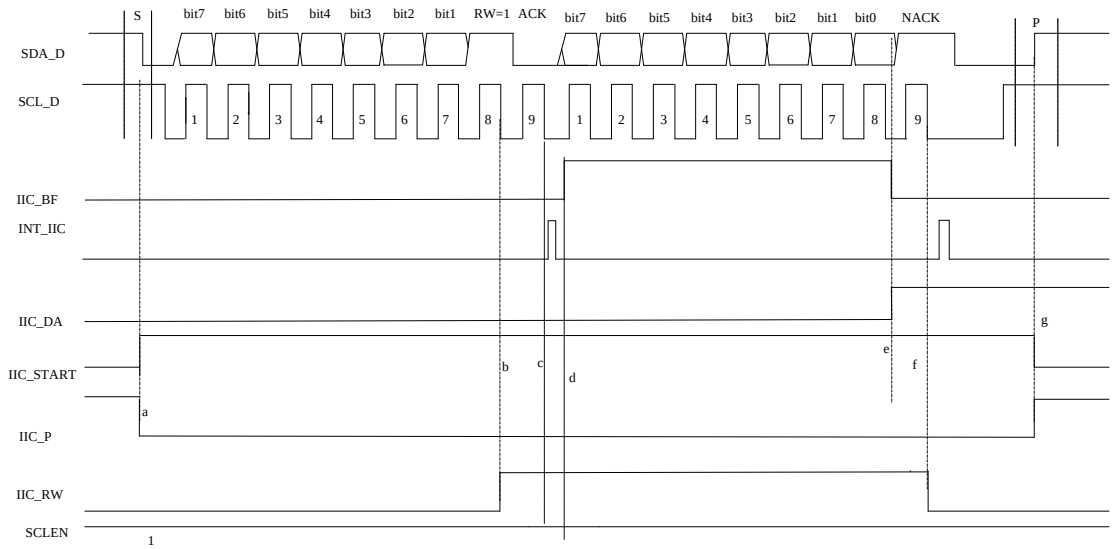
如上图所示是主机写操作时拉低时钟线的示意图，从中可以看到 IIC 总线的变化情况以及一些内部信号的变化。

首先，主机发送启动信号 IIC_START，从机检测到 IIC_START 信号之后置位 IIC_START 状态位，如图中虚线 a 所示。

然后，主机发送地址字节和读写标志位，从机在接收到地址字节后硬件自动与自身地址比较，如果匹配则在第 8 个时钟的下降沿之后置位 IIC_BF，如虚线 b 所示。在第 9 个时钟的下降沿之后会产生中断信号 INT_IIC，如虚线 c 所示。第 9 个时钟的下降沿之后 SCLEN 会被硬件自动清零，此期间用于从机处理或者读取数据，即使此数据没有用，读取 IICBUF 的操作会使 IIC_BF 间接的清零，如虚线 d 所示。再软件置位 SCLEN，释放时钟线，如虚线 e 所示。主机在检测到从机释放 SCL 之后，会继续发送同步时钟，在第 2 个字节的第 8 个时钟的下降沿之后 IIC_BF 同样被置位，同时 IIC_AD 标志位也会置位，标志当前接收到的字节是数据，如虚线 f 所示，停止信号对 IIC_AD 标志位没有影响即检测到停止信号 IIC_STOP，IIC_AD 标志位不会被清零；第 9 个时钟的下降沿之后同样会产生中断。如果主机要发送多个字节，则可以继续发送，如上图中仅仅示意主机发送一个数据的情况。需要注意的情况是在主机发送最后一笔数据时，不使能拉低时钟线的功能。

最后，主机在发送完毕所有的数据之后发送一个停止信号 IIC_STOP，标志着通信的结束，释放 IIC 总线，总线进入空闲状态。

IIC 主机读时序示描述



IIC 主机读不拉低时钟线图

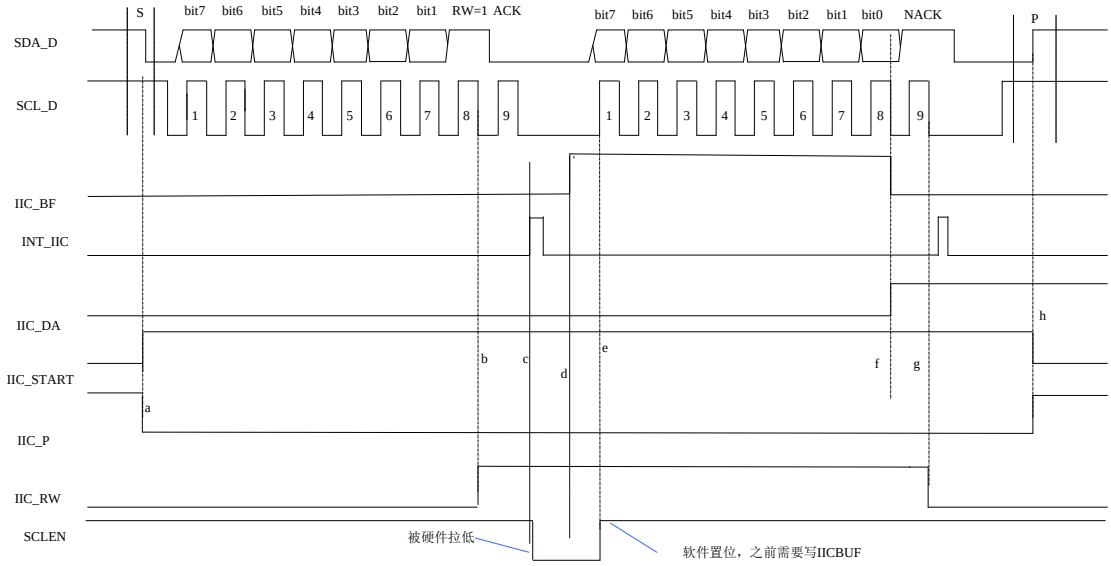
如上图所示，是主机读取从机时钟线拉低的时序示意图。由图可知道总线的变化情况，以及一些电路内部信号的变化情况。

首先，主机发送 IIC_START 信号，标志通信的开始，如虚线 a 所示，内部电路检测到 IIC_START 信号时序后置位状态标志位 IIC_START。

然后，IIC_START 信号之后主机发送地址字节，并且 IIC_RW=1，表示主机读取从机。地址匹配的情况下在第八个时钟的下降沿之后，状态位 IIC_RW 置位，如虚线 b 所示，如果地址不匹配则 IIC_RW 不会置位。在第九个时钟的下降沿之后会产生中断信号，如虚线 c 所示。并将 IICBUFFER 中的数据压载到 IICBUF 中，IIC_BF 被置位，如虚线 d 所示，并将最高位送到总线上。在 8 个时钟的之后，一个字节数据发送完毕，IIC_BF 标志位被清零；同时地址数据标志位也会置位，表示当前发送的字节数据。如虚线 e 所示。第 9 个时钟的下降沿之后会产生中断，如果主机需要继续读取从机，则主机回复有效应答位 ACK，继续通信；如果主机需要的数据已经读取完毕，则主机回复无效应答 NACK，然后发送停止信号 IIC_STOP，终止通信。示意图中主机仅仅读取了一个数据后，回复 NACK，然后发送 IIC_STOP 信号，终止了通信。在检测到 NACK 的时候读写标志位 IIC_RW 被硬件清零了，如虚线 f 所示。如果主机发送 NACK 的时候，从机 SCLEN 是不会被自动拉低的，这点在应用中要注意。

最后，主机在读完所有的数据之后发送一个停止信号 IIC_STOP，标志着通信的结束，检测到 IIC_STOP 信号的时候，状态位 IIC_STOP 置位，IIC_START 清零，释放 IIC 总线，如虚线 g 所示，总线进入空闲状态。

IIC 主机读拉低时序描述



IIC 主机读拉低时钟线图

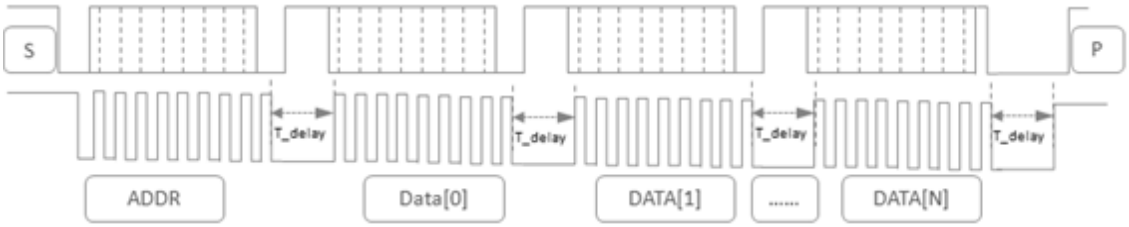
如上图所示，是主机读取从机时钟线拉低的时序示意图。由图可知道总线的变化情况，以及一些电路内部信号的变化情况。

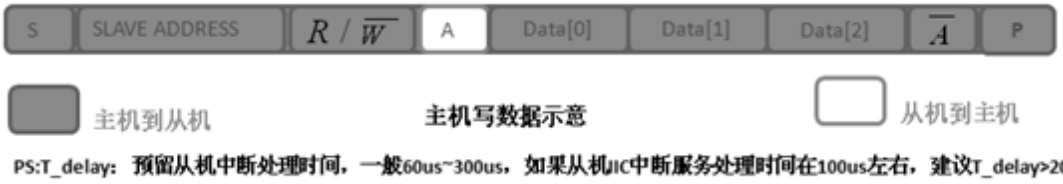
首先，主机发送 IIC_START 信号，标志通信的开始，如虚线 a 所示，内部电路检测到 IIC_START 信号时序后置位状态标志位 IIC_START。

然后，IIC_START 信号之后主机发送地址字节，并且 IIC_RW=1，表示主机读取从机。地址匹配的情况下在第八个时钟的下降沿之后，状态位 IIC_RW 置位，如虚线 b 所示，如果地址不匹配则 IIC_RW 不会置位。在第九个时钟的下降沿之后会产生中断信号，如虚线 c 所示。第 9 个时钟的下降沿之后 SCLEN 也会被硬件自动拉低，此期间用于从机处理或者准备数据，然后把准备好的数据写入 IICBUF，再软件置位 SCLEN，释放时钟线。如虚线 d 所示，在把数据写入 IICBUF 中后，IIC_BF 会置位，标志这 IICBUF 是满的。如虚线 e 所示，软件置位 SCLEN，释放时钟线。主机在检测到从机释放 SCL 之后，会继续发送同步时钟，读取从机的数据，第 8 个时钟的下降沿之后，一个字节数据发送完毕，IIC_BF 标志位被清零；同时地址数据标志位也会置位，表示当前发送的字节数据。如虚线 f 所示。第 9 个时钟的下降沿之后会产生中断，如果主机需要继续读取从机，则回复有效应答位 ACK，继续通信；如果主机需要的数据已经读取完毕，则回复无效应答 NACK，然后发送停止信号 IIC_STOP，终止通信。示意图中主机仅仅读取了一个数据后，回复 NACK，然后发送 IIC_STOP 信号，终止了通信。在检测到 NACK 的时候读写标志位 IIC_RW 被硬件清零了，如虚线 g 所示。

最后，主机在读完所有的数据之后发送一个停止信号 IIC_STOP，标志着通信的结束，检测到 IIC_STOP 信号的时候，状态位 IIC_STOP 置位，IIC_START 清零，释放 IIC 总线，如虚线 h 所示，总线进入空闲状态。

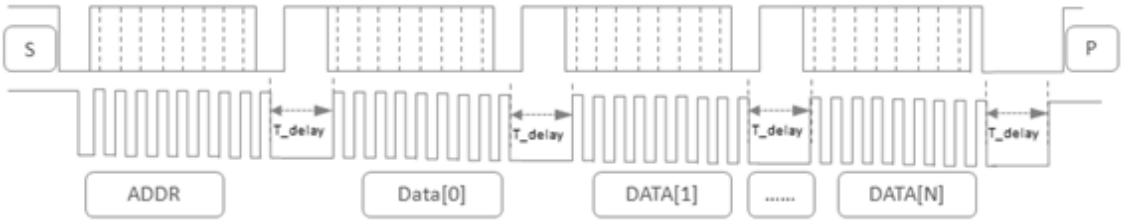
IIC 主机写数据示意图

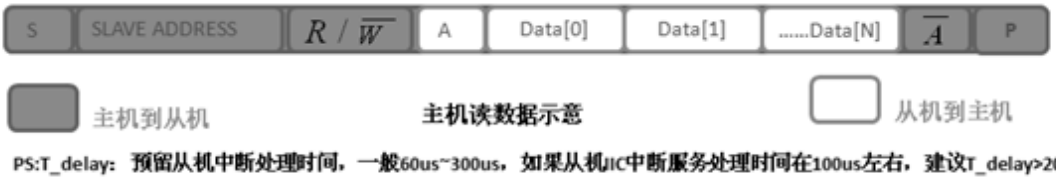




PS:T_delay：预留从机中断处理时间，一般60us~300us，如果从机IIC中断服务处理时间在100us左右，建议T_delay>200us。

IIC 主机读数据示意图





PS:T_delay：预留从机中断处理时间，一般60us~300us，如果从机IIC中断服务处理时间在100us左右，建议T_delay>200us。

从机在第八个时钟的下降沿给出 ACK 信号，第九个时钟的下降沿产生 IIC 中断，建议主机在发完第九个时钟下降沿的时候延时 60us~300us 预留从机 IIC 中断服务数据准备时间，然后再发时钟信号。

9.3. IIC 寄存器详细说明

地址	名称	位	对应位名称	RW	说明	复位值
0xE3	IICADD	<7:1>	IICADD	RW	地址寄存器	0x00
0xE4	IICBUF	<7:0>	IICBUF	RW	IIC 发送接收数据缓冲器	0x00
0xE5	IICCON	<7:6>	--	R	保留	B00
		<5>	IIC_RST	RW	IIC 模块复位信号	B0

					1：IIC 模块发生复位操作 0：IIC 模块正常工作	
		<4>	RD_SCL_EN	RW	主机读拉低时钟线控制位 1:使能主机读拉低时钟线功能； 0:不使能主机读拉低时钟线功能	B1
		<3>	WR_SCL_EN	RW	主机写拉低时钟线控制位, 1:使能写拉低时钟线的功能； 0:不使能写拉低时钟线的功能	B0
		<2>	SCLEN	RW	IIC 时钟使能位 1=时钟正常工作； 0=拉低时钟线	B0
		<1>	SR	RW	IIC 转换率控制位 1=转换率控制被关闭以适应标准速度模式(100K)；不写 0，只写 1。	B0
		<0>	IICEN	RW	IIC 工作使能位 1=IIC 正常工作； 0=IIC 不工作	B0
0xE8	IICSTAT	<7>	IIC_START	R	开始信号标志位 1=表示检测到了启动位； 0=表示未检测到启动位	B0
		<6>	IIC_STOP	R	停止信号标志位 1=表示处于停止状态； 0=表示未检测到停止位	B1
		<5>	IIC_RW	R	读写标志位 记录最近一次地址匹配后，从地址字节中获得的读/写信息， 1=表示读操作；	B0

					0=表示写操作	
		<4>	IIC_AD	R	地址数据标志位 1=表示最近接收或者发送的字节是数据； 0=表示最近接收或者发送的字节是地址	B0
		<3>	IIC_BF	R	IICBUF 满标志位 在 IIC 总线方式下接收时： 1=表示接收成功，缓冲器已经满； 0=表示接收未完成，缓冲器还为空 在 IIC 总线方式下发送时： 1=表示数据发送正在进行(不包括应答位和停止位)，缓冲器还是满的； 0=表示数据发送已经完成(不包括应答位和停止位)，缓冲器已空	B0
		<2>	IIC_ACK	R	应答标志位 1：表示无效的应答信号； 0：表示有效的应答信号	B1
		<1>	IIC_WCOL	RW	写冲突标志位 1=表示 IIC 正在发送当前的数据的时候，新的数据试图写入发送缓冲器；新的数据是不能被写入缓冲器的； 0=未发生写冲突	B0
		<0>	IIC_RECOV	RW	接收溢出标志位 1=表示 IIC 接收的前一个数据还没有取走时，又接收到了新的数据，新的数据是不能被缓冲器接收的 ；	B0

					0=表示未发生接收溢出	
0xE9	IICBUFFER	<7:0>	IICBUFFER	RW	IIC 发送数接收据缓存寄存器；在 RD_SCL_EN 为 0 的情况下，主机读取数据时，在产生中断后 2 个时钟之后将 IICBUFFER 中为的数据发送到从机发送缓存寄存器中，作为从机发送的数据。故在中断产生之前准备 IICBUFFER 中断数据。	0x00
0xF2	PERIPH_IO_SEL<6:5>	<6>	IIC_AFIL_SEL	RW	IIC 口模拟滤波选择使能 1：选择模拟滤波功能 0：不选择模拟滤波功能	1'h1
		<5>	IIC_DFIL_SEL	RW	IIC 口数字滤波选择使能 1：选择数字滤波功能(IIC 无唤醒核功能) 0：不选择数字滤波功能	1'h0

IIC_START：起始信号状态位。当检测到起始信号之后 IIC_START 就会置位，表示总线处于忙的状态。

IIC_STOP：停止信号状态位。当芯片处于停止状态时 IIC_STOP 就会置位，表示总线处于空闲状态，当检测到开始信号时被硬件清零，表示通信开始。

IIC_AD：地址数据标志位。它标志当前接收或者发送的字节是地址或者是数据。IIC_AD=0 标志接收或者发送的字节是地址；IIC_AD=1 标志接收或者发送的字节是数据。开始信号、停止信号、非应答信号对此状态位没有任何影响。此状态位的改变发生在第 8 个时钟的下降沿。

IIC_RW：读写标志位。此标志位记录地址匹配后，从地址字节中获取的读写信息位。IIC_RW=1 表示主机读取从机的操作，IIC_RW=0 表示主机写从机的操作。起始信号、停止信号、非应答信号(NACK)都会清零 IIC_RW。此状态位的改变发生在第九个时钟的下降沿。

IIC_BF：缓冲器满标志位。它标志收发缓冲器当前是满的或者是空的。IIC_BF=0 表示缓冲器没有接收数据，缓冲器为空；IIC_BF=1 表示缓冲器接收到数据，缓冲器已满。此状态位只能间接的置位和清零，不能直接的操作。

地址匹配并且 IIC_RW=0 的情况下，在第 8 个时钟的下降沿之后 IIC_BF 就会置位，标志着 IICBUF 接收到数据。在执行中断程序期间应该读取 IICBUF，读取 IICBUF 的操作会间接的清零 BF 标志位。如果不读取 IICBUF，主机继续发送数据，则会发生接收溢出，虽然从

机仍然接收主机发送数据并压载到 IICBUF，但仍会发送 NACK 信号，给出无效应答。

地址匹配并且 IIC_RW=1 的情况下，从机接收到地址字节后 IIC_BF 标志位不会置位；IIC_RW=1 表示主机读取从机的操作，从机需要把数据写入 IICBUF，从机写 IICBUF 的操作会置位 IIC_BF，然后软件置位 SCLEN，释放时钟线；主机就会发送同步时钟，第 8 个时钟过后，IICBUF 里的数据发送出去之后，IIC_BF 被硬件清零。

IIC_ACK：应答状态位。不管主机是读操作还是写操作，从机都会在第 9 个时钟的上升沿采样数据线，记录应答信息。应答位分为有效应答位 ACK 和非有效应答位 NACK。也就是说第 9 个时钟的上升沿采样到数据为“0”，表示有效应答 ACK，同时 IIC_ACK 被清零，如果采样到数据“1”则 IIC_ACK 被置位，表示非应答。非应答信号之后，主机就会发送停止信号宣告通信结束。启动信号会清零此状态位。

IIC_WCOL：写冲突标志位。IICBUF 只有在 IIC_RW=1，并且 RD_SCL_EN=1,SCLEN=0 的情况下才可以被 CPU 写入。其它任何情况下试图写 IICBUF 是被禁止的，如果不满足上面的条件下发生了写 IICBUF 的操作，则数据不会被写入 IICBUF，同时写冲突标志位 IIC_WCOL 被置位，表示发生了写冲突，此标志位需要软件清零。

IIC_RECOV：接收溢出标志位。在 IICBUF 满的情况下，也就是 IICBUF 里存在数据的情况下，IIC 有接收到新的数据，则会发生接收溢出，IIC_RECOV 会置位，同时 IICBUF 里的数据不会被更新，新接收的数据会丢失。此状态位也需要软件清零，否则的话会应影响后面的通信过程。此种情况只会出现在 IIC_RW=0，BF=1，且 CPU 不读取 IICBUF 时会出现。

9.4.IICCON 寄存器

IIC 控制寄存器，用于控制通信工作情况。

位	名称	R/W	复位值	描述
7：6	--	R	0	保留
5	IIC_RST	R/W	0	IIC 模块复位控制使能位 1:使能 IIC 模块复位的功能； 0:不使能 IIC 模块复位的功能
4	RD_SCL_EN	R/W	1	主机读拉低时钟线控制位 1:使能主机读拉低时钟线功能； 0:不使能主机读拉低时钟线功能
3	WR_SCL_EN	R/W	0	主机写拉低时钟线控制位, 1:使能写拉低时钟线的功能； 0:不使能写拉低时钟线的功能
2	SCLEN	R/W	0	IIC 时钟使能位 1=时钟正常工作； 0=拉低时钟线(IICEN=1 有效)
1	SR	R/W	0	IIC 转换率控制位 1=转换率控制被关闭以适应标准速度模式(100K)；

				0=转换率控制被关闭以适应快速速度模式(400K)
0	IICEN	R	0	IIC 模块使能信号 1=IIC 模块工作；

下面详细介绍各位的作用：

IICEN 是 IIC 模块的使能信号，只有 IICEN=1 时，电路才工作。

SR 是转换速率控制位，SR=1 转换速率控制关闭，端口适应于 100Kbps 的通信。

SCLEN 是时钟使能控制位，虽然从机不能产生通信时钟，但是根据协议从机可以延长时钟的低电平时间。SCLEN=0 时钟线被锁定在低电平，SCLEN=1 释放时钟线。延长时钟低电平的前提是 IICEN=1，否则内部电路不会对 IIC 总线产生任何影响。SCLEN 常用来延长低电平的时间，使主机进入等待状态，这样从机就有足够的时间来处理数据。

WR_SCL_EN 是写拉低线控制位，为 1 时使能中断拉低时钟线的功能，为 0 时不使能中断拉低时钟线的功能。

在 IIC_RW=0 的情况下，可根据主机的通信速率和处理中断的时间来决定是否拉低时钟线，即配置 WR_SCL_EN 位。

当 CPU 能在 8 个 IIC 时钟内能处理完中断并退出中断时，WR_SCL_EN=0 不使能拉低时钟线的功能，此时在中断到来时不会硬件自动拉低时钟线。当 CPU 不能在 8 个 IIC 时钟内处理完中断并退出时，WR_SCL_EN=1 使能拉低时钟线的功能，此时在中断到来时硬件自动拉低时钟线，迫使主机进入等待状态，当写入 IICBUF 中的数据被 CPU 读出后，软件置位 SCLEN。

RD_SCL_EN 是读拉低线控制位，为 1 时使能中断拉低时钟线的功能，为 0 时不使能中断拉低时钟线的功能。

当 RD_SCL_EN=1 时从机在接收到地址字节或者发送完一个字节并且主机发送 ACK 的时候，SCLEN 会被硬件自动拉低，迫使主机进入等待状态。从机要释放 IIC 时钟，需要下面两个操作：先将把要发送的数据写入 IICBUF 中，再软件置位 SCLEN。这样设计的目的是确保拉高 SCL 之前，IICBUF 中已经写入将要发送的数据。

当 RD_SCL_EN=0 时从机在接收到地址字节或者发送完一个字节并且主机发送 ACK 的时候，从机会将 IICBUFFER 寄存器中准备好的数据立即压载到发送缓存寄存器中，然后送到数据线上。故为保证每次传送的数据正确，在中断服务程序中 IICBUFFER 准备好下一个要发送的数据，主机接收的数据都为上一次中断处理好的数据，第一次接收数据为初始化中准备。

注意：当需要拉低时钟线，即 WR_SCL_EN/RD_SCL_EN=1，在发送和接受最后一个 Byte 数据之前，软件应该关闭拉低时钟线的功能,即 WR_SCL_EN/RD_SCL_EN=0，在完成发送和接受最后一个 Byte 数据之后，软件应该打开写拉低时钟线的功能。此种操作可根据主机是软件硬件，中断处理处理时间自行调控。

IIC_RST 是 IIC 模块控制使能位，为 1 时使能 IIC 模块复位的功能；为 0 时不使能 IIC 模块复位的功能。应用时注意配置 1 复位 IIC 模块所有 DFF 触发器，IIC_RST 的复位端为全局复位，其他的复位端为 iic_rst_n，所有先将 iic_rst 位写 0 后，再操作其他寄存器配置。

9.5. IICBUF 寄存器

IIC 读写缓存寄存器，用于控制通信工作情况。

位	名称	R/W	复位值	描述
7：0	IICBUF	R/W	0	IIC 数据接收发送的缓冲器

具体应用过程如下：

在发送状态下，把数据压载到 IICBUF 中后，在主机的同步时钟作用下，数据依次移位发送出去，高位在前。8 个时钟过后，一个字节发送完毕。

在接收状态下，在主机的 8 个时钟过后，数据被写入 BUF 中去，第 9 个时钟过后会产生中断，告诉 CPU 读取 IICBUF 中的数据。

把数据写入 IICBUF 此操作是有条件的，在 RD_SCL_EN=1 时只有 IIC_RW=1，并且 SCLEN=0 的情况下才可以把数据写入 IICBUF 中；否则写 IICBUF 的操作是被禁止的。也就是说条件不满足的情况下，写 IICBUF 的操作不能成功的，数据写不进去，IICBUF 的数据不会改变，同时也会造成写冲突。

例如：IICBUF 已经有数据 55h，在写 IICBUF 的条件不满足的情况下，欲把数据 00h 写入 IICBUF。结果是 IICBUF 里的数据仍然是 55h，同时写冲突标志位 IIC_WCOL 置位，用于告诉用户，操作异常。

在 RD_SCL_EN=0 时，从机需发送的数据为中断信号产生时压载 IICBUFFER 寄存器值得到。

9.6. IICBUFFER 寄存器

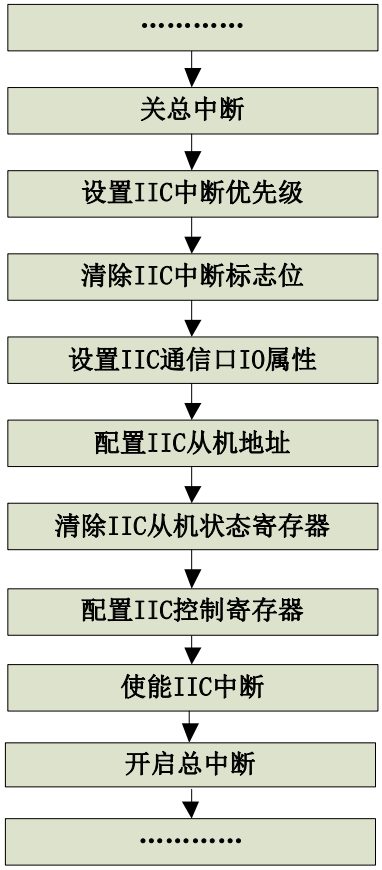
IIC 数据发送缓存器

位	名称	R/W	复位值	描述
7:0	IICBUFFER	R/W	0	IIC 数据发送的缓冲器

具体应用过程如下：

在 RD_SCL_EN 为 0 的情况下，主机读取数据时，在产生中断后 2 个 clk 之后将 IICBUFFER 中的数据送到从机发送缓存寄存器中，作为从机发送的数据。故在中断产生之前，IICBUFFER 中的数据要准备好，一般情况下是在上一个中断服务程序中准备好，设备地址产生中断发送数据为初始化中准备。

9.7. IIC 配置流程



IIC 配置流程图

注：1、IIC 总线上拉电阻 4.7K~10K,对地滤波电容建议 10pf~100pF 靠近引脚芯片。

第 10 章 UART

GDMC191XXXX 系列 UART 模块接口特点：

- 支持全双工，半双工串口通信
- 具有独立的双缓冲接收器，单缓冲发射器
- 可编程波特率（10 位模数分频器）
- 中断驱动型或轮询操作：
- — 发送完成
- — 接收满
- — 接收溢出、奇偶效验错误、帧错误
- 支持硬件奇偶效验生成和检查
- 可编程 8 位或 9 位字符长度
- 可选择 STOP 位 1 位或 2 位
- 支持多处理器模式

10.1.UART0 功能说明

10.1.1.波特率生成

波特率生成模数 bandrate={UART0_BDH[1:0], UART0_BDL}。

波特率计算公式：bandrate=0 时，不生成波特率时钟，当 bandrate=1~1023 时，UART0 波特率= BUSCLK/（16xbandrate）。

BUSCLK 使用系统时钟源的分频时钟，固定为 24M,每次配置波特率寄存器均会清零内部 counter 重新生成波特率信号。通信要求发射器和接收器使用相同的波特率，通信允许的波特率偏差范围：8/11*16=4.5%

10.1.2.发射器功能

发送数据流程：由写入 UART0_BUF 数值开启发送，发送停止位后置位发送中断，软件清除中断标记，等待下一次写入。发射器输出管脚（TXD）闲置状态,默认为逻辑高态。整个发送过程必须在模块使能时进行。通过把数据写入数据寄存器（UART0_BUF），会将数据直接保存到发送数据缓冲器并开启发送过程，在后续完整的发送过程中，不允许写入数据寄存器 UART0_BUF 和 T8，直到发送完毕停止位后，发送中断标志置位，才可以再次写入 UART0_BUF，重新开启新的发送。

串口发射器的中心元件是长度为 10/11/12 位（取决于 data_mode 控制位中的设置）的发送移位寄存器。假设 data_mode=0，选择正常的 8 位数据模式,在 8 位数据模式中，移位寄存器中有 1 个起始位、8 个数据位和 1 个/2 个停止位,发送接收均是小端模式（LSB 先发）。

10.1.3.接收器功能

通过设置 UART0_CON1 中的接收使能位，接收器被使能。当然，整个接收过程必须在模块使能时进行。

接收数据流程：在接收使能有效的情况下，随时接收数据，接收停止位后置位接收中断，软件清除中断标记。

当前接收的数据会有检测机制，可检测接收溢出、帧出错、奇偶校验出错 3 种错误，均需要软件清除标记。建议检测到接收中断后，读出状态标记，读数据 buf，最后将接收数据状态标记均清除（UART0_STATE[3:0]）。

数据字符由逻辑 0 的起始位、8 个（或 9 个）数据位（LSB 先发）和逻辑 1 的停止位（1bit）组成。在把停止位接收到接收移位器后，如果接收数据寄存器还未满(rx_full_if=0)，数据字符就被传输到接收数据寄存器，设置接收数据寄存器已满（rx_full_if=1）状态标记。如果此时已经设置了接收数据寄存器已满的 rx_full_if，就设置溢出（rx_overflow_if）状态标记，新数据会丢失。因为接收器是双缓冲的，程序在设置 rx_full_if 后、读取接收数据缓冲器的数据前，有一个全字符时间来供读取，以避免接收器溢出。

当程序检测到接收数据寄存器已满（rx_full_if=1）时，它通过读 UART0_BUF 从接收数

据寄存器中获取数据。

10.1.4. 接收器采样方法

接收器使用 16 倍波特率时钟进行采样。接收器通过以 16 倍波特率提取逻辑电平样本，以搜索 RXD 串行数据输入管脚上的下降边沿。下降边沿的定义是 3 个连续逻辑 1 采样后的逻辑 0 样本。16 倍波特率时钟用来把位时间划分为 16 个段，分别标记为 RT1 到 RT16。

接收器然后在 RT8，RT9 和 RT10 的每个位时间上进行采样，包括起始位和停止位，以决定该位的逻辑电平。当定位了下降边沿时，通过逻辑电平为 0 确保这是真正的起始位，而不是噪音，如果这三个样本至少有两个样本为 0，接收器假设它与接收器字符同步，开始移位接收下面数据，如果不满足以上条件则退出状态机回到等待下降边沿状态。下降边沿检测逻辑不断寻找下降边沿，如果检测到边沿，样本时钟重新同步位时间。

10.1.5. 多处理器模式

多处理器模式，仅工作在 9 位模式下，接收到的 R8 位=1 时，接收中断置位，否则不置位，此机制的作用是利用硬件检测消除了处理不重要信息字符的软件开销。允许接收器忽略用于不同接收器的信息中的字符。

在这种应用系统中，所有接收器都估计每条信息的地址字符（第 9 位=1），一旦确定该信息旨在用于不同接收器，后续数据字符（第 9 位=0）不接收。

配置流程：配置接收使能，配置多处理器模式，接收到地址数据（第 9 位=1），接收并产生中断，应用确认地址是否匹配，匹配则配置关闭多处理器模式，后续所有数据（第 9 位=0）均能被接收并产生中断，直到下一次接收到地址数据，地址不匹配，则打开多处理器模式，则后续所有数据均不被接收，直到下一个地址数据，依次循环应用。

10.2. UART0 相关寄存器

SFR 寄存器				
地址	名称	读写	复位值	说明
0xBD	UART0_BDL	RW	0x00	UART0 波特率控制寄存器
0xBE	UART0_CON1	RW	0x00	UART0 模式控制寄存器 1
0xBF	UART0_CON2	RW	0x0C	UART0 模式控制寄存器 1
0xC0	UART0_STATE	RW	0x00	UART0 状态标记寄存器
0xC1	UART0_BUF	RW	0xFF	UART0 数据寄存器

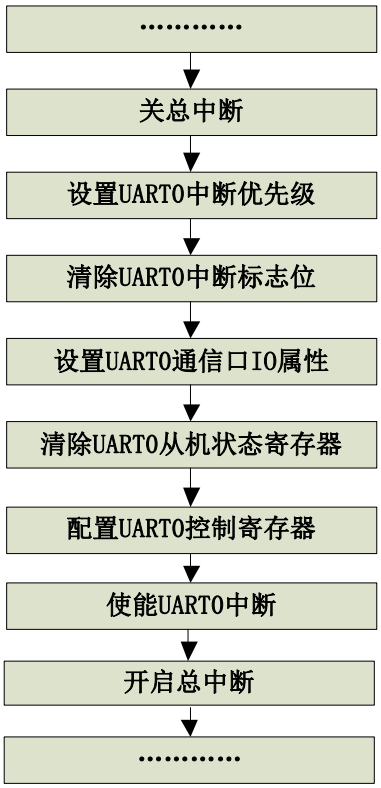
UART0 SFR 寄存器列表

10.3. UART0 寄存器详细说明

地址	名称	位	对应位名称	RW	说明	复位值
0xBD	UART0_BDL	<7:0>	--	RW	波特率控制寄存器 波特率模数除数寄存器低 8 位,bandrate={UART0_BDH[1:0],UART0_BDL}, bandrate=0 时不生成波特率时钟，当 bandrate=1~1023 时，波特率= BUSCLK/(16xbandrate)	0x00
0xBE	UART0_CON1	<6:0>	--	RW	模式控制寄存器 bit[6]:uart0_enable，模块使能，1：模块使能，0：模块关闭 bit[5]:receive_enable，接收器使能，1：接收器打开，0：接收器关闭 bit[4]:multi_mode，多处理器通信模式，1：模式使能，0：模式禁能 bit[3]:stop_mode，stop 位宽选择，1：2 位，0：1 位 bit[2]:data_mode,数据模式选择，1:9 位模式，0:8 位模式 bit[1]:parity_en,奇偶校验使能，1：奇偶效验使能，0：奇偶效验不使能 bit[0]:parity_sel，奇偶校验选择，1：奇校验，0：偶校验	0x00
0xBF	UART0_CON2	<3:2>	--	RW	控制寄存器： bit[3]:tx_empty_ie,发送中断使能，1：中断使能，0：中断禁止(用于轮询模式) bit[2]:rx_full_ie，接收中断使能，1：中断使能，0：	2’0x03

					中断禁止(用于轮询模式)	
		<1:0>	UART0_BDH	RW	bit[1:0]:UART0_BDH,波特率模数除数寄存器高 2 位	2’0x00
0xC0	UART0_STATE	<6:0>	--	RW	状态标记寄存器： bit[6]:r8，接收器的第 9 个数据，只读 bit[5]:t8，发射器的第 9 个数据 bit[4]:tx_empty_if,发送中断标记，1：发送缓存为空，0：发送缓存为满，软件写 0 清零 bit[3]:rx_full_if，接收中断标记，1：接收缓存为满，0：接收缓存为空，软件写 0 清零 bit[2]:rx_overflow_if，接收溢出标记，1：接收溢出(新数据丢失)，0：没有溢出，软件写 0 清零 bit[1]:frame_err_if，帧错误标记，1：检测到帧错误，0：未检测到帧错误，软件写 0 清零 bit[0]:parity_err_if,奇偶校验错误标记，1：接收器奇偶校验错误，0：奇偶校验正确，软件写 0 清零	7’0x00
0xC1	UART0_BUF	<7:0>	--	RW	数据寄存器 读返回只读接收数据缓冲器的内容,写进入只写发送数据缓冲器	0xFF

10.4. UART0 配置流程



UART0 初始化配置流程图

1.配置模块使能、接收使能、模式选择：UART0_CON1;

2.配置波特率，打开中断使能：UART0_BDL、UART0_CON2;

3.写入 UART0_BF 开始发送数据，检测到发送中断后，清除中断标记 tx_empty_if;

4.检测到接收中断，首先读取接收状态 UART0_STATE，然后读取 R8 和 UART0_BUF，最后清除接收状态标记(UART0_STAT[3:0] = B0000)，一次接收流程完毕，等待下一个接收中断。

5.如果配置中断不使能，程序执行 UART 功能，同样要首先读取状态标记，然后读取 R8 和 UART0_BUF，最后清除状态标记。

6.中断标志位清除操作，在全双工工作时，清除标志位操作需要将有效中断位写 0，其它中断位写 1(写 1 为无效操作)，否则容易误操作。例：发送中断有效时，需要写 UART0_STATE = 0x0F;(即配置 UART0_STATE[0:3] = 0x0F，R8 写无效，t8 在 9 位模式且不奇偶校验时需要配置有效发送数据)。

7. 8 位模式：奇偶校验使能无效。

9 位模式：奇偶校验位使能时，第第九位计算得到的奇偶校验位不使能时，第九位为写进去的 T8。仅有发送中断和接收中断，错误标记仅标记当前数据的错误检测，且仅对应位写 0 清除，不出错误中断，发送中断在发送完毕停止位后置 1，软件清 0, 接收中断在接收完毕停止位后置 1，软件清 0。

多处理器模式：仅工作在 9 位模式下，接收到的 R8 位 = 1 时，接收中断置位，否则不置位。使用多处理器模式时，配置接收使能，配置多处理器模式，接收地址数据(第 9 位=1)接收并产生中断，应用确认地址是否匹配，匹配则配置关闭多处理器模式，后续所有数据(第 9 位 = 0)均能被接收中断产生中断，直到下一次接收到地址数据，地址不匹配，则打开多处理器模式，则后续所有数据不被接收，直到下一个地址数据依次循环应用。

硬件响应：发送数据，由写入 UART0_BUF 数值开启，发送停止位后置位发送中断标记，软件清除中断标记，等待下一次写入。接收数据在接收使能有效的情况下，随时接收数据，接收停止位后置位接收中断，软件清除中断标记。当前接收的数据会有检测机制，可检测接收溢出、帧出错、奇偶检验出错 3 种错误，均需要软件清除标记。建议检测到接收中断后，读出状态标记，将接收状态标记均清除 UART0_STATE[0:3]。

第 11 章 PWM

11.1. PWM0 功能说明

PWM0 脉宽调制模块为周期和脉宽都可以通过寄存器进行配置，但是寄存器的配置必须在 PWM0 使能有效（高有效）的情况下，而且每组寄存器（包括 PWM0_MOD_L/H, PWM0_CHX_CNT_L/H）必须按照从低到高的顺序配置，为了保证 PWM0 模块内部计数器正确计数，避免产生错误波形。这些配置值在等到计数器从（PWM0_MOD）变为（PWM0_MOD+1）时更新寄存器值，就是在一个完整的周期之后再更新周期和占空比。

PWM0 模块有一个 16 位的计数器，计数时钟是 24MHz 和系统时钟是同步的。PWM0 信号的周期由周期配置寄存器（PWM0_MOD）的值确定，占空比由通道寄存器（PWM0_CHn_CNT）中的设置确定， PWM0 信号的极性由 PWM0_CH_CTRL 控制位中的设置确定。0%和 100%的占空比都是可能的。

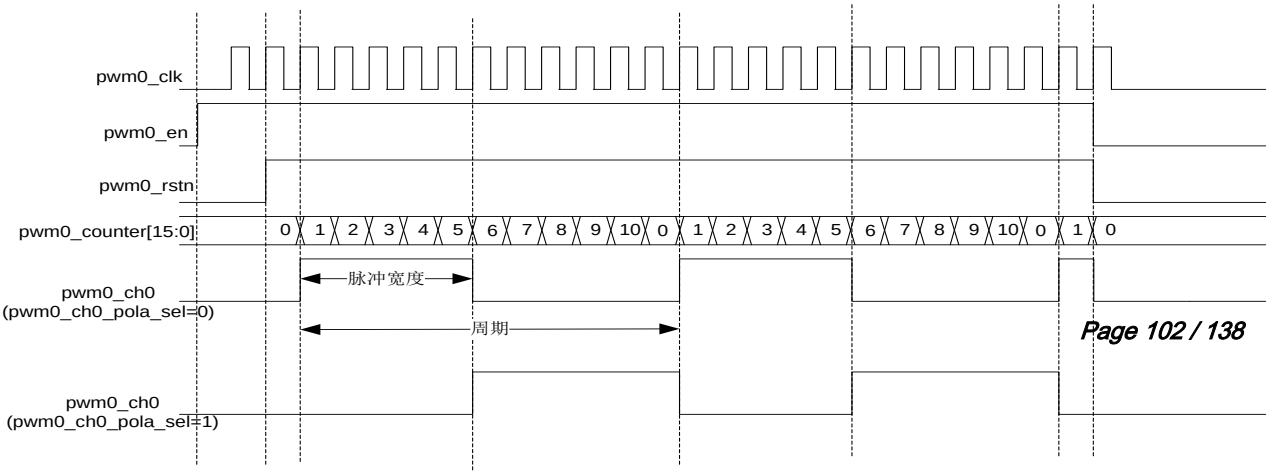
脉冲宽度 =（PWM0_CHn_CNT）

周期 =（PWM0_MOD+1）

占空比=脉冲宽度/周期

PWM0 计数器从 0x0000 开始向上计数，当计到 PWM0_CHn_CNT 时输出翻转，这段时间为脉冲宽度，继续计数直到计到 PWM0_MOD+1 时计数溢出。如果 PWM0_CH0_POLA_SEL=0，输出翻转时 PWM0 信号进入低态，计数溢出时 PWM0 信号进入高态。如果 PWM0_CH0_POLA_SEL=1，输出翻转时 PWM0 信号进入高态，计数溢出时 PWM0 信号进入低态。

当通道计数寄存器（PWM0_CHn_CNT）被设为 0x0000 时，占空比为 0%；当通道计数



寄存器（PWM0_CHn_CNT）设置为大于周期配置寄存器（PWM0_MOD）设置的值可实现100%的占空比。计数器是自动重载的，不会自行停止，直到寄存器 PWM0 使能关闭才会停止，计数器清零。

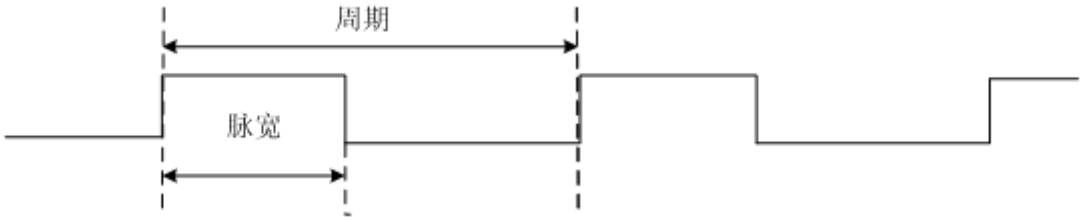
（PWM0_CH0_CNT=5，PWM0_MOD=10，duty_cycle=5/11）

11.2. PWM1/2 功能说明

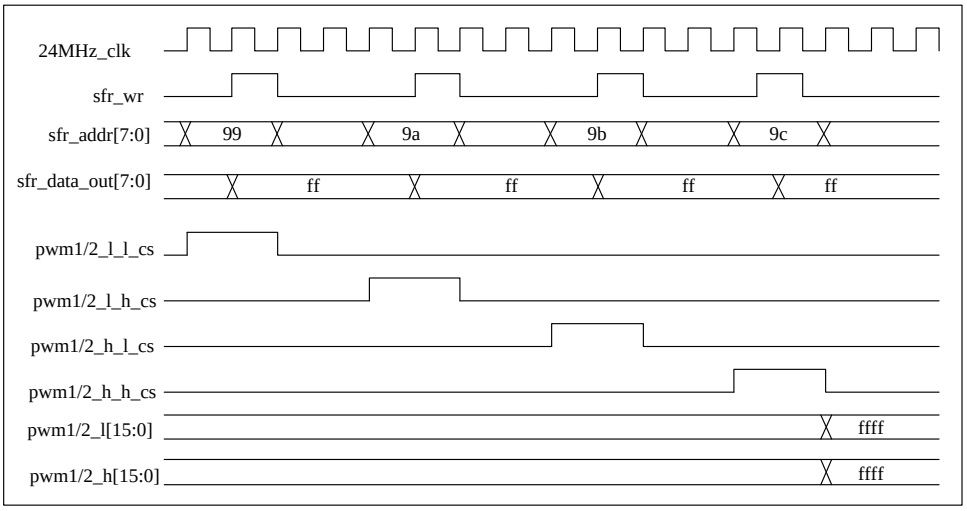
PWM1/2 的功能特点如下：

- 时钟来源时钟 CLK_24MHz；
- PWM1/2 的高电平控制寄存器与低电平控制寄存器均为 16 位寄存器；
- 输出周期：TPWM1/2_data = (PWM1/2_H + PWM1/2_L)* T_{CLK_24MHz}(us)；
- 输出占空比：DPWM1/2_data = PWM1/2_H/(PWM1/2_L + PWM1/2_H)；

PWM1/2 波形意图



PWM1/2 脉宽调制模块为高低电平时间都可以通过寄存器进行配置，但是寄存器的配置必须在 PWM1/2 使能有效（高有效）的情况下，而且高电平控制寄存器和低电平控制寄存器必须按照从低到高的顺序配置，为了保证 PWM1/2 模块内部计数器正确计数，避免产生错误波形。这些配置值在一个完整的周期之后再更新周期和占空比。



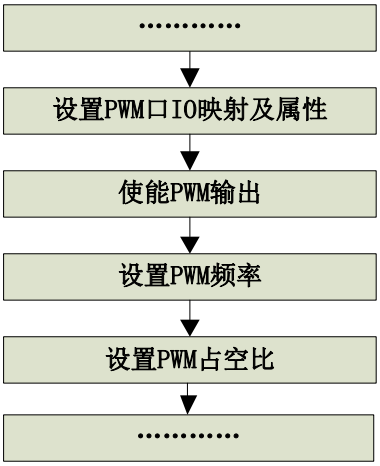
PWM1/2 时序示意图

11.3. PWM0/1/2 寄存器详细说明

11.2.1. PWM0/1/2 寄存器

地址	名称	位	对应位名称	RW	说明	复位值
0x99	PWM1_L_L	<7:0>	--	RW	PWM1 低电平控制寄存器(低 8 位)	8'h00
0x9A	PWM1_L_H	<7:0>	--	RW	PWM1 低电平控制寄存器(高 8 位)	8'h00
0x9B	PWM1_H_L	<7:0>	--	RW	PWM1 高电平控制寄存器(低 8 位)	8'h00
0x9C	PWM1_H_H	<7:0>	--	RW	PWM1 高电平控制寄存器(高 8 位)	8'h00
0x9D	PWM2_L_L	<7:0>	--	RW	PWM2 低电平控制寄存器(低 8 位)	8'h00
0x9E	PWM2_L_H	<7:0>	--	RW	PWM2 低电平控制寄存器(高 8 位)	8'h00
0x9F	PWM2_H_L	<7:0>	--	RW	PWM2 高电平控制寄存器(低 8 位)	8'h00
0xA1	PWM2_H_H	<7:0>	--	RW	PWM2 高电平控制寄存器(高 8 位)	8'h00
0xA2	PWM_EN	<2>	PWM2_EN	RW	PWM2 模块使能寄存器 1：使能 0：不使能	1'b0
		<1>	PWM1_EN	RW	PWM1 模块使能寄存器 1：使能 0：不使能	1'b0
		<0>	-	R	保留	1'b0
0xA3	PWM0_CH_CTRL	<7:5>	--	R	保留	1'b0
		<4>	PWM0_CH0_POLA_SEL	RW	通道 0 极性选择 ch0_pola_sel 1：计数值溢出使输出低 0：计数值溢出使输出高	1'b0
		<3:1>	--	R	保留	1'b0
		<0>	PWM0_CH0_EN	RW	通道 0 使能 ch0_en 1：使能 0：不使能	1'b0
0xA4	PWM0_CH0_CNT_L	<7:0>	PWM0_CH0_CNT_L	RW	通道 0 计数值配置寄存器低 8 位 配置 PWM 输出占空比	8' h00
0xA5	PWM0_CH0_CNT_H	<7:0>	PWM0_CH0_CNT_H	RW	通道 0 计数值配置寄存器高 8 位 配置 PWM 输出占空比	8' h00
0xAD	PWM0_MOD_L	<7:0>	PWM0_MOD_L	RW	PWM0 计数周期配置寄存器低 8 位 配置 PWM 输出占空比	8' h00
0xAE	PWM0_MOD_H	<7:0>	PWM0_MOD_H	RW	PWM0 计数周期配置寄存器高 8 位 配置 PWM 输出占空比	8' h00

11.4 PWM 配置流程



PWM 配置流程示意图

第 12 章 触摸按键

CSD 的功能特点：

- 最多支持 14 个 Sensor 通道；
- CSD 充放电时钟三种模式可选；
  - 系统时钟的固定分频 6M~369K
  - PRS 1.5M 正态分布
  - PRS 1.5M 均匀分布
- CSD 计数时钟 24M、12M、6M、4M 可选；
- 计数位宽 9~16 位可选；
- 仅支持非同步扫描模式。

GDMC191XXXX 通过一系列的寄存器来实现多种功能的应用。电容检测相关量与 SFR 值的关系如下：

计数值大小与 RESO、Rb 电阻、PULL_I_SELA_H 成正比，与 VTH_SEL 成反比。在保证充放电完全的情况下，与通过 PRS_DIV 设定的充放电频率成正比。

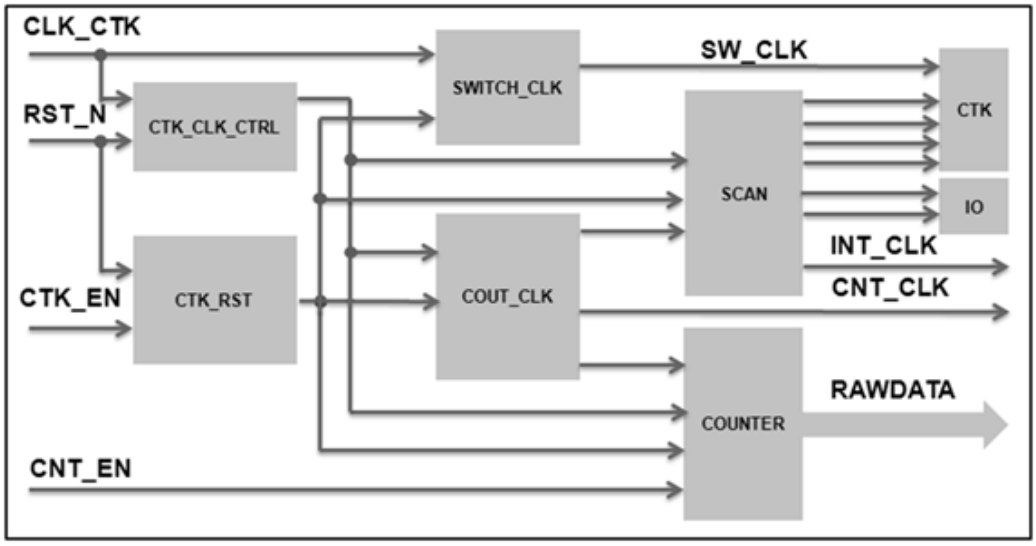
通道触摸变化量与 RESO、Rb 成正比，与 VTH_SEL 成反比。在保证充放电完全的情况下，与通过 PRS_DIV 设定的充放电频率与触摸变化量成正比。

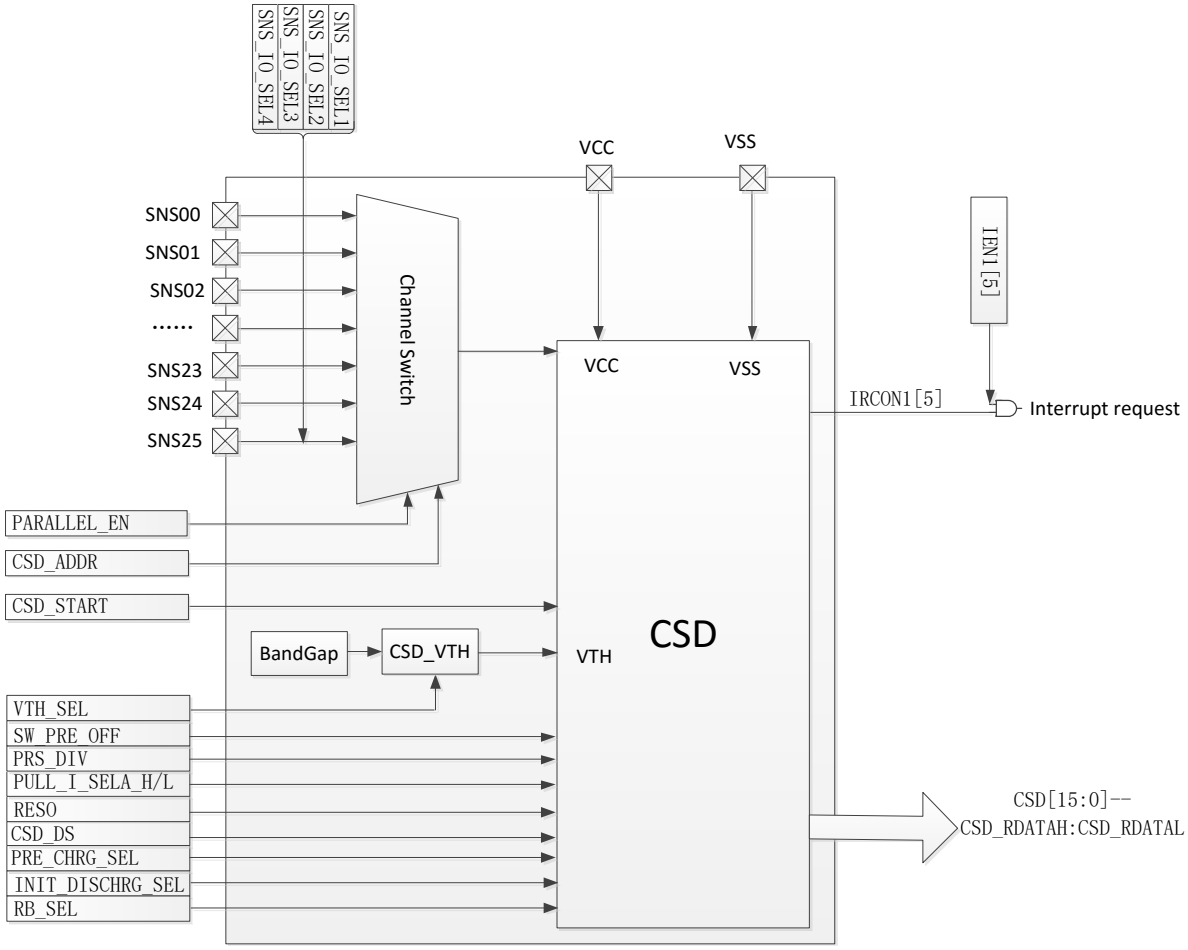
触摸检测的信噪比与 VTH_SEL 成正比，PULL_I_SELA_L,与 CSD_DS 成反比。在充放电不完全时，与通过 PRS_DIV 设定的充放电频率与信噪比成反比。

单个按键检测的时间与 RESO、CSD_DS 有关。

注：配置参数时应保证按键充放电完全。

CSD 模块结构示意图





CSD 结构框图

12.1. 触摸按键相关寄存器

SFR 寄存器				
地址	名称	读写	复位值	说明
0xCA	CSD_START	RW	1`0x00	CSD 扫描开启寄存器
0xCB	SNS_SCAN_CFG1	RW	0x00	触摸按键扫描配置寄存器 1
0xCC	SNS_SCAN_CFG2	RW	0x40	触摸按键扫描配置寄存器 2
0xCD	SNS_SCAN_CFG3	RW	0x70	触摸按键扫描配置寄存器 3
0xCE	CSD_RAWDATAL	R	0x00	CSD 计数值低 8 位
0xCF	CSD_RAWDATAH	R	0x00	CSD 计数值高 8 位
0xD0	PSW	RO/RW	0x00	程序状态字寄存器
0xD1	PULL_I_SELA_L	RW	0x00	CSD 上拉电流源选择寄存器
0xD2	SNS_ANA_CFG	RW	6`0x2F	CSD 扫描参数配置寄存器
0xD3	SNS_IO_SEL1	RW	0x00	SNS 通道选择寄存器 1

0xD4	SNS_IO_SEL2	RW	0x00	SNS 通道选择寄存器 2
0xD5	SNS_IO_SEL3	RW	0x00	SNS 通道选择寄存器 3
0xD6	SNS_IO_SEL4	RW	0x00	SNS 通道选择寄存器 4

CSD SFR 寄存器列表

12.2.触摸按键寄存器详细说明

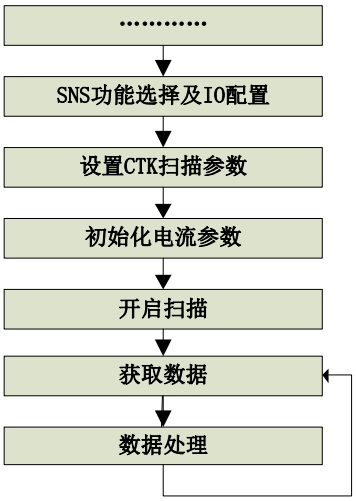
地址	名称	位	对应位名称	RW	说明	复位值
0xCA	CSD_START	<0>	--	RW	CSD 扫描开启寄存器	1'h0
0xCB	SNS_SCAN_CFG1	<6>	SW_PRE_OFF	RW	前端充放电时钟开关控制 1：关闭 sw_clk；0：打开 sw_clk	1'h0
		<5:0>	PRS_DIV	RW	前端充放电时钟频率选择寄存器: 0~61：为固定频率： F=F48m/2/(PRS_DIV+4) （6M~369K）； 62：最高频率 3M，最低频率 1M,中心频率 1.5M,正态分布； 63：最高频率 3M，最低频率 1M,中心频率 1.5M,均匀分布；	6'h0
0xCC	SNS_SCAN_CFG2	<6>	PULL_I_SELA_H	RW	CSD 上拉电流源配置最高位	1'h1
		<5>	PARALLEL_EN	RW	SNS 通道并联使能寄存器 1：多通道并联 0：单通道	1'h0
		<4:0>	CSD_ADDR	RW	检测通道的地址，对应通道号 0~25	5'h0
0xCD	SNS_SCAN_CFG3	<6:4>	RESO	RW	计数器位数选择寄存器 000 ： 9 位； 001 ： 10 位； 010 ： 11 位； 011 ： 12 位； 100 ： 13 位； 101 ： 14 位； 110： 15 位；	3'h7

					111 ：16 位。	
		<3:2>	CSD_DS	RW	计数时钟频率选择寄存器 00：24M； 01：12M； 10：6M； 11：4M； 默认 0	2’h0
		<1>	PRE_CHRG_SEL	RW	预充电时间选择 0：20us 1：40us	1’h0
		<0>	INIT_DISCHRG_SEL	RW	预放电时间选择 0：2us 1：10us	1’h0
0xCE	CSD_RAWDATAL	<7:0>	RAWDATA <7:0>	R	CSD 的计数值	8’h00
0xCF	CSD_RAWDATAH	<7:0>	RAWDATA <15:8>			8’h00
0xD1	PULL_I_SELA_L	<7:0>	PULL_I_SEL <7:0>	RW	CSD 上拉电流源大小选择开关,默认值 0	8’h00
0xD2	SNS_ANA_CFG	<5:3>	RB_SEL	RW	Rb 电阻大小选择 0：10k；1：20k；2：30k；3：40k；4：60k；5：80k；6：150k；7：300k；使用时需要从芯片 Flash 读取 Rb80K 校准值：CBYTE[0x43CD]K/80K,进行比例计算归一化灵敏度。	3’h5
		<2:0>	VTH_SEL	RW	VTH 电压选择信号 VTH 电压选择信号，000 为 1.8V,001 选择 2.1V，010 选择 2.5V,011 选择 2.8V，100 选择 3.2V，101 选择 3.5V,110 选择 3.9V，111 选择 4.1V	3’h7
0xD3	SNS_IO_SEL1	<7:0>	SEL_SENSOR [7:0]	RW	SENSOR 口选择使能 1：选择 SENSOR，0：不选择 SENSOR	8’h00
0xD4	SNS_IO_SEL2	<7:0>	SEL_SENSOR	RW	SENSOR 口选择使能	8’h00

			[15:8]		1：选择 SENSOR，0：不选择 SENSOR	
0xD5	SNS_IO_SEL3	<7:0>	SEL_SENSOR [23:16]	RW	SENSOR 口选择使能 1：选择 SENSOR，0：不选择 SENSOR	8'h00
0xD6	SNS_IO_SEL4	<1:0>	SEL_SENSOR [25:24]	RW	SENSOR 口选择使能 1：选择 SENSOR，0：不选择 SENSOR	2'h0

12.3. 触摸按键配置流程

触摸按键扫描为查询或中断方式，首先，配置触摸按键参数；然后，开启触摸按键扫描；最后，在触摸按键中断获取并保存触摸按键数据，软件算法进行数据的处理及按键的输出判断。



触摸按键扫描配置流程图

通过灵敏度参数配置得到较好信噪比的一组参数，从而提高按键判断的准确性。

1、 **RESO**：0~7 触摸按键电容扫描分辨率，计数器位数：**(RESO + 9)位**，触摸按键电容扫描分辨率越大， Rawdata 向下变化量越大，同时引入的噪声也会随之增大，反之相反。

2、 **VTH_SEL**：0~7，参考电压越小，Rawdata的变化量越大，同时引入的噪声也会随之增大，反之相反。

3、 **CSD_DS**：检测速度 **0:24M,1:12M,2:6M,3:4M**，检测速度越小，采样 Rawdata 的时间越慢，反之相反。建议默认 24M 最快，检测速度至少为 2 倍的 PRS 时钟。

4、 **RB_SEL**：Rb 电阻选择：**0:10K,1:20K,2:30K,3:40K,4:60K,5:80K,6:150K,7:300K**;电阻越大，Rawdata 的变化量越大，同时引入的噪声也会随之增大，反之相反。

5、 **PRS_DIV**：前端充放电时钟频率选择寄存器：

0~61：为固定频率：F=F48m/2/(PRS_DIV+4)（6M~369K）；

62：最高频率 3M，最低频率 1M，中心频率 1.5M，正态分布；

63：最高频率 3M，最低频率 1M，中心频率 1.5M，均匀分布；

PRS 时钟越大，Rawdata 的变化量越大，同时引入的噪声也会增大，反之相反。

6、 **PULL_I_SELA_L**：上拉电流源低8位。

7、 电流电流源大小=255.5-0.5*{PULL_I_SELA_H ,PULL_I_SELA}，电流源越小，计数值越小。默认值：0x00。

8、 **PULL_I_SELA_H**：上拉电流源高位。默认值：0x01。

注：

1.Rawdata 为触摸按键电容计数器的实时原始计数值。

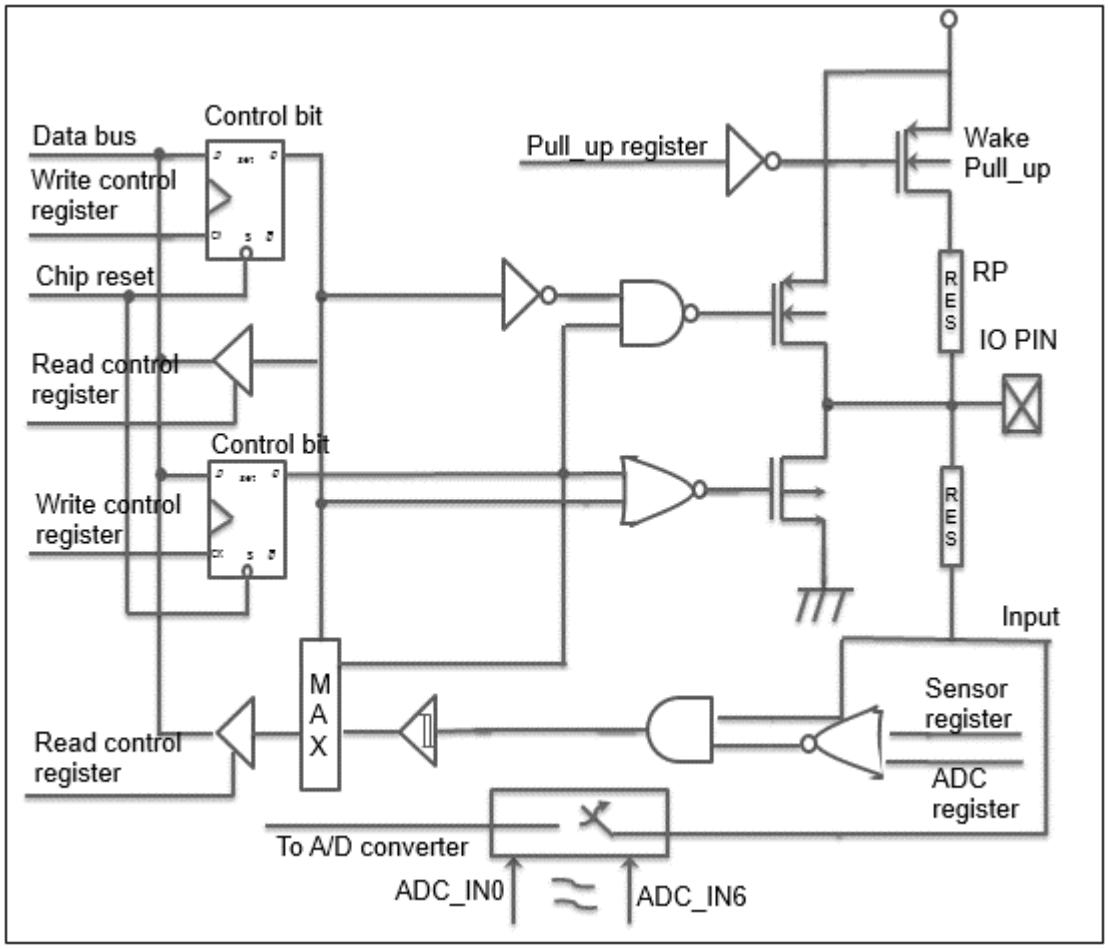
2.实际应用中需要通过烧录调试软件查看数据并进行参数对比得到信噪比较好的一组参数。

3.芯片供电电压与参考电压关系：VCC-VTH>0.5V。

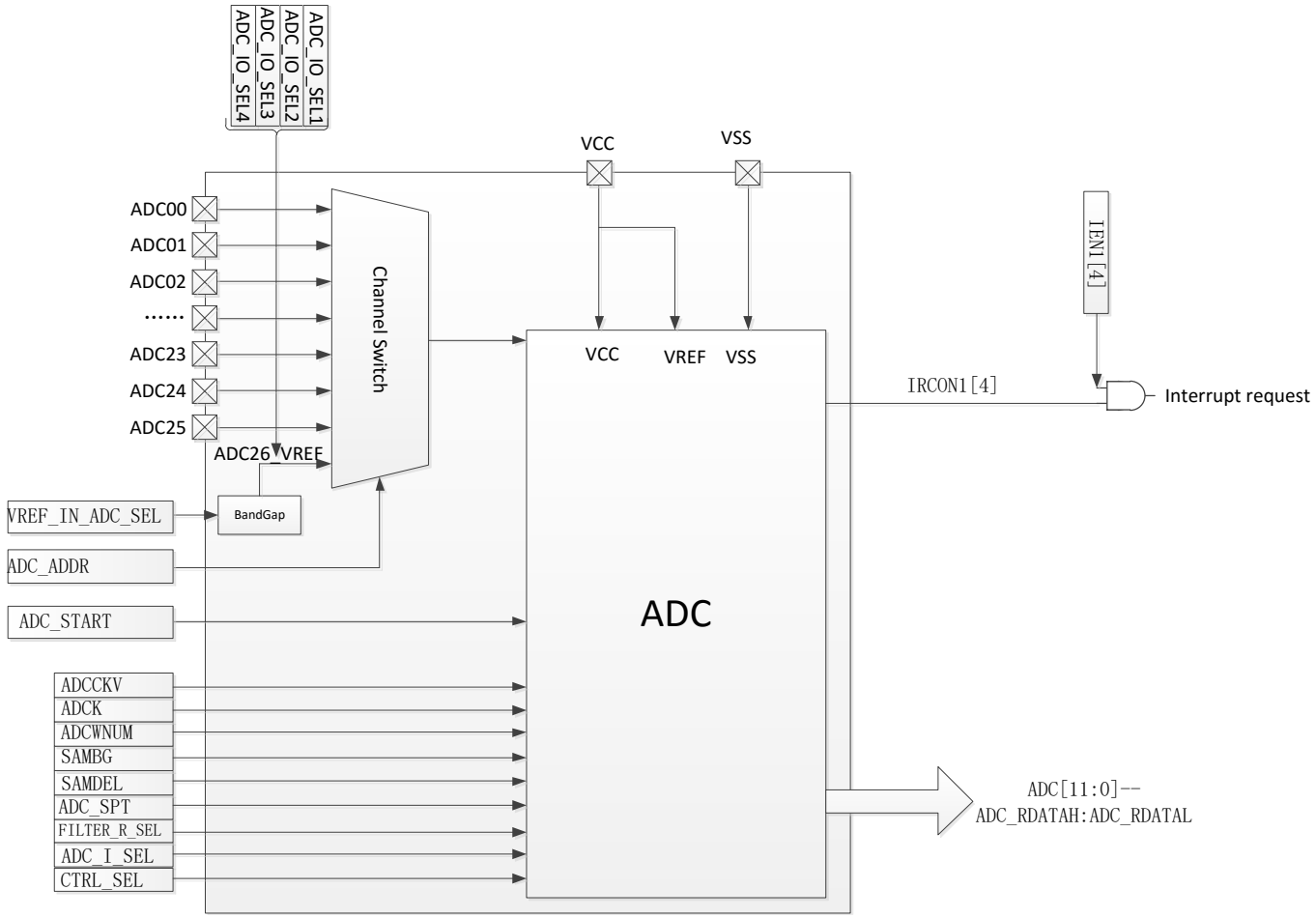
第 13 章 ADC

GDMC191XXXX 芯片包含一个单端、12 位线性逐次逼近的模数转换器(ADC)，ADC 的基准电压与芯片的 VCC 相连。最多支持 14 个 ADC 通道都可以输入独立的模拟信号，ADC 模块每次转换 1 个通道，ADC_START=0→1( )开启转换，转换完成后更新 ADC 结果寄存器并产生一个中断。GDMC191XXXX 芯片的 ADC 模块具有以下特性：

- 12 位分辨率 10 位精度的线性逐次逼近 ADC；
- 最多 14 个模拟输入通道；
- 单次转换模式；
- 采样时间和转换速度可配置
- 支持 wait 模式下唤醒



ADC 模块结构示意图



ADC 结构框图

13.1. ADC 相关寄存器

SFR 寄存器				
地址	名称	读写	复位值	说明
0xB4	ADC_SPT	RW	8'h00	ADC 采样时间配置寄存器
0xB5	ADC_SCAN_CFG	RW	6'h0	ADC 扫描控制寄存器
0xB6	ADCCKC	RW	4'h0	ADC 时钟控制寄存器
0xB9	ADC_RDATAH	R	4'h0	ADC 扫描结果寄存器，高 4 位
0xBA	ADC_RDATAL	R	8'h00	ADC 扫描结果寄存器，低 8 位
0xBB	ADC_CFG1	RW	8'h00	ADC 采样时序控制寄存器 1
0xBC	ADC_CFG2	RW	8'h02	ADC 采样时序控制寄存器 2
0xD9	ADC_IO_SEL1	RW	8'h00	ADC 功能选择寄存器 1
0xDA	ADC_IO_SEL2	RW	8'h00	ADC 功能选择寄存器 2
0xDB	ADC_IO_SEL3	RW	8'h00	ADC 功能选择寄存器 3

0xDC	ADC_IO_SEL4	RW	2'h0	ADC 功能选择寄存器 4

ADC SFR 寄存器列表

13.2. ADC 寄存器详细说明

地址	名称	位	对应位名称	RW	说明	复位值
0xB4	ADC_SPT	<7:0>	ADC_SPT	RW	ADC 采样时间配置寄存器 采样时间：sample_timer = (ADC_SPT+1)*4*Tadc_clk	8'h00
0xB5	ADC_SCAN_CFG	<5:1>	ADC_ADDR	RW	ADC 通道地址选择寄存器	5'h0
		<0>	ADC_START	RW	ADC 扫描开启寄存器	1'h0
0xB6	ADCCKC	<3:2>	ADCCKV	RW	模拟输入时钟信号分频选择 0：12MHZ 1：8MHZ 2：4MHZ 3：2MHZ	4'h0
		<1:0>	ADCK	RW	adc_clk 分频选择 0：8MHZ 1：6MHZ 2：4MHZ 3：3MHZ	
0xB9	ADC_RDATAH	<3:0>	ADC_RAWDATA <11:8>	R	ADC 扫描结果寄存器	4'h0
0xBA	ADC_RDATAL	<7:0>	ADC_RAWDATA <7:0>			8'h00
0xBB	ADC_CFG1	<7:3>	ADCWNUM	RW	采样完毕后距离转换间隔时间选择 3+ADCWNUM(ADC_CLK)	5'h0
		<2>	SAMBG	RW	采样时序与比较时序间隔选择：0：间隔 0；1：间隔 1(ADC_CLK)	1'h0
		<1:0>	SAMDEL	RW	采样延迟时间选择 0：0；1：2；2：4；3：8(ADC_CLK)	2'h0
0xBC	ADC_CFG2	<6>	FILTER_R_SEL	RW	输入信号滤波选择，0 为不加 RC 滤波，1 为加 RC 滤波	1'h0
		<5:4>	VREF_IN_ADC_SEL	RW	输入给 ADC26 基准电压选择	2'h0

					01：2.253V；其它：保留 使用时需要从芯片 Flash 读取校准电压值， VREF_IN_ADC_SEL 档电压={CBYTE[0x43C6],CBYTE[0x43C7]}mV。	
		<3:2>	ADC_I_SEL [1:0]	RW	ADC 偏置电流大小选择寄存器 ADC_I_SEL[0]: 0 为比较器偏置电流为 4uA; 1 为比较器偏置电流为 5uA; ADC_I_SEL[1]: 0 为运放偏置电流为 4uA; 1 为运放偏置电流为 5uA;	2'h0
		<1:0>	CTRL_SEL [1:0]	RW	ADC 比较器失调消除选择信号，默认值为 10 CTRL_SEL[1:0]: 00/01:为先采样再失调消除; 10：所有开关一起断开; 11：开关依次断开;	2'h2
0xD9	ADC_IO_SEL1	<7:0>	SEL_ADC[7:0]	RW	ADC 功能选择 1：选择 ADC 功能，0：不选择 ADC 功能	8'h00
0xDA	ADC_IO_SEL2	<7:0>	SEL_ADC[15:8]	RW	ADC 功能选择 1：选择 ADC 功能，0：不选择 ADC 功能	8'h00
0xDB	ADC_IO_SEL3	<7:0>	SEL_ADC[23:16]	RW	ADC 功能选择 1：选择 ADC 功能，0：不选择 ADC 功能	8'h00
0xDC	ADC_IO_SEL4	<1:0>	SEL_ADC[25:24]	RW	ADC 功能选择 1：选择 ADC 功能，0：不选择 ADC 功能	2'h0

注：

1. 时序要求:（ADCWNUM+3）* Tadc_clk > 4*Taclk;
2. ADC转换时间：T= sample_time + time2 + time3;
   sample_time= (ADC_SPT+1)*4*Tadc_clk;
   time2=（ADCWNUM+3+SAMDEL）* Tadc_clk(SAMDEL=0：0；1：2；2：4；3：8);

time3=（2*1+12 ）*Tadc_clk;

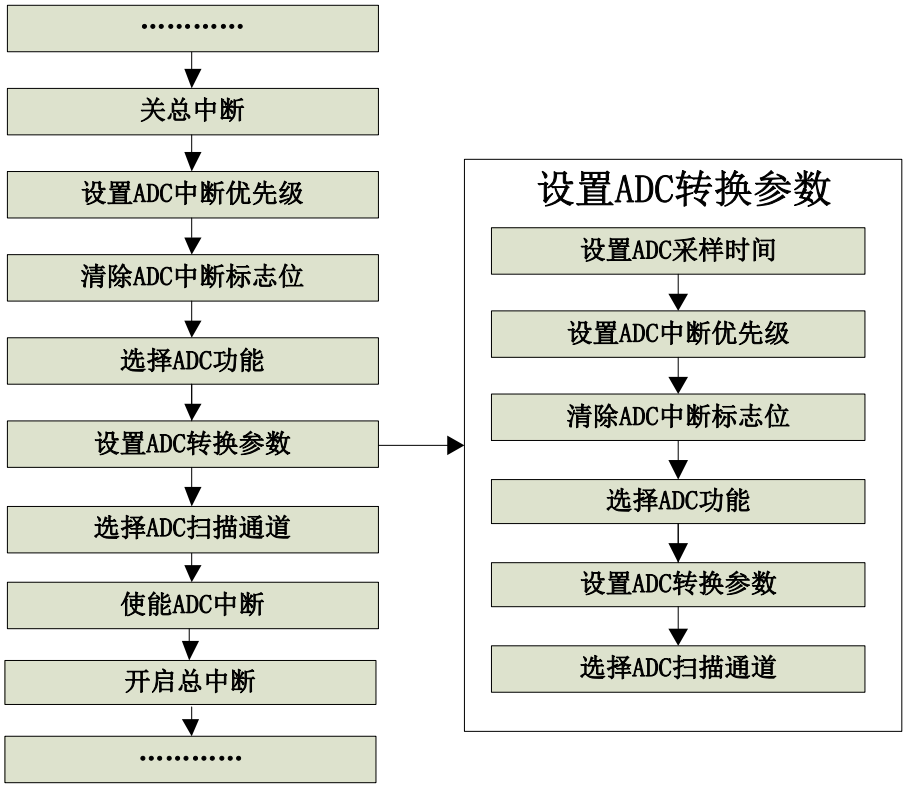
ADC外入信号加RC滤波后的电压建立时间>= 2*(ADC采样转换出数时间);

3. 输入电压Vin = (ADC_Data/ADC26_Data)*VREF_IN_ADC_SEL，VREF_IN_ADC_SEL需要读取芯片校准值，先获取ADC26数据，再获取输入电压ADC_Data数据，两次获取数数间隔时间尽量短。

ADC 检测时间：

公式	说明
$T_{AD}=T_{ADC\_SPT}+T_{W1}+T_{W2}$	ADC 检测时间
$T_{W1}=(ADCWNUM+3)*T_{adc\_clk}$	采样完毕转后距离转换间隔时间
$T_{W2}=(SAMDEL+2*1+12)*T_{adc\_clk}$	采样延迟时间
$T_{ADC\_SPT(us)} = 4*(ADC\_SPT+1)/F_{adc\_clk(MHz)}$	ADC 采样时间
$F_{adc\_clk(MHz)}$	ADC 分频时钟

13.3. ADC 配置流程



ADC 配置流程图

第 14 章 LVDT

GDMC191XXXX 系列支持低压报警功能,有效监控电压动态变化情况。支持 4 个档位电压，分别为：2.4V/3.0V/3.6V/4.2V（预设点降压中断，迟滞 0.1V 产生对应升压中断）。
当电压监控配置上述阈值时，电压下降至此阈值会触发低压中断，系统可根据应用需要，在低压中断中做适当的处理。

14.1. LVDT 相关寄存器

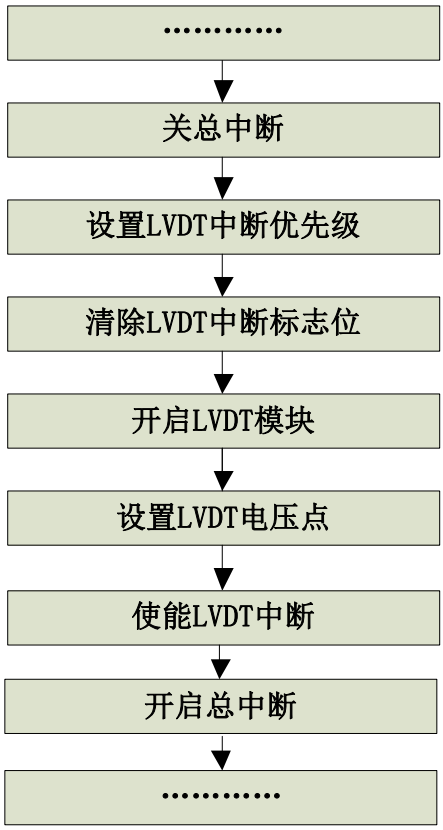
SFR 寄存器				
地址	名称	读写	复位值	说明
0x86	INT_POBO_STAT	RW	2'h0	升压/降压中断状态寄存器
0xFF	SEL_LVDT_VTH	RW	2' h0	LVDT 阈值选择寄存器

LVDT SFR 寄存器列表

14.2. LVDT 寄存器详细说明

地址	名称	位	对应位名称	RW	说明	复位值
0xD5	INT_POBO_STAT	<1:0>	INT_POBO_STAT[1:0]	RW	升压/降压中断状态寄存器	2'h0
0xFF	SEL_LVDT_VTH	<1:0>	--		LVDT 阈值选择： 00：2.4V; 01：3.0V; 10：3.6V; 11：4.2V	2' h0

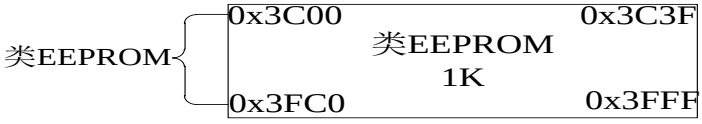
14.3. LVDT 配置流程



LVDT 配置流程图

第 15 章 类 EEPROM

类 EEPROM 大小为 1024 Bytes 为一页，地址为(0x3C00~0x3FFF)，使用时需要进行页擦除，然后进行字节写操作，擦除后仅能被写入一次。



类 EEPROM 地址

15.1. 类 EEPROM 相关寄存器

SFR 寄存器				
地址	名称	读写	复位值	说明
0xF9	SPROG_ADDR_H	RW	4`0x00	EEPROM 地址控制寄存器
0xFA	SPROG_ADDR_L	RW	0x00	EEPROM 地址控制寄存器
0xFB	SPROG_DATA	RW	0x00	EEPROM 定操作数据寄存器
0xFC	SPROG_CMD	RW	0x00	EEPROM 定操作命令寄存器
0xFD	SPROG_TIM	RW	0x1A	EEPROM 擦写时间控制寄存器

15.2. 类 EEPROM 寄存器详细说明

地址	名称	位	位名称	RW	说明	复位值
0xF9	SPROG_ADDR_H	<3:0>	----	RW	Bit[2] ：选择 EEPROM 块(可进行页擦除和字节烧写)， 0：表示选择 block0；1：表示选择 block1。每一 block 大小 1024Bytes。 Bit[1:0]：表示 EEPROM 块地址的高 2 位,SPROG_ADDR[9:8]。	4'h00
0xFA	SPROG_ADDR_L	<7:0>	----	RW	Bit[7:0]：表示 EEPROM 块地址的低 8 位,SPROG_ADDR[7:0]。	8'h00
0xFB	SPROG_DATA	<7:0>	----	RW	EEPROM 烧写 ：待写入的数据	8'h00
0xFC	SPROG_CMD	<7:0>	----	RW	写入 0x96：EEPROM 页擦除 写入 0x69：EEPROM 字节烧写	8'h00
0xFD	SPROG_TIM	<7:0>	----	RW	bit[7:5]：0~7 对应烧写时间 25~60us,5us 步进。默认 35us	8'h1a

					bit[4:0]：0~22 对应擦除时间 1~23ms(步进 1ms)。 23~31 对应擦除时间 24~40ms(步进 2ms)。默认 30ms	

15.3. 页擦除步骤

1. EEPROM 擦写时间控制寄存器(SPROG_TIM)SPROG_TIM[7:5] = 3(建议 35us)，SPROG_TIM[4:0] = 1~31(建议 30ms),在主程序 main()函数只配置一次;
2. 关闭中断；
3. 配置 SPROG_ADDR_L = 0x00;
4. 配置 SPROG_ADDR_H = 0x00;选择擦除该页；
5. 配置 SPROG_CMD = 0x96;
6. 写入 4 个 NOP 指令；
7. 开始擦除，CPU 进入 IDLE 模式，擦除完成后自动退出 IDLE 模式；
8. 需要继续擦除数据，跳转至第 2 步；
9. 配置 SPROG_ADDR_L=0x00，SPROG_ADDR_H=0x00,恢复中断设置；

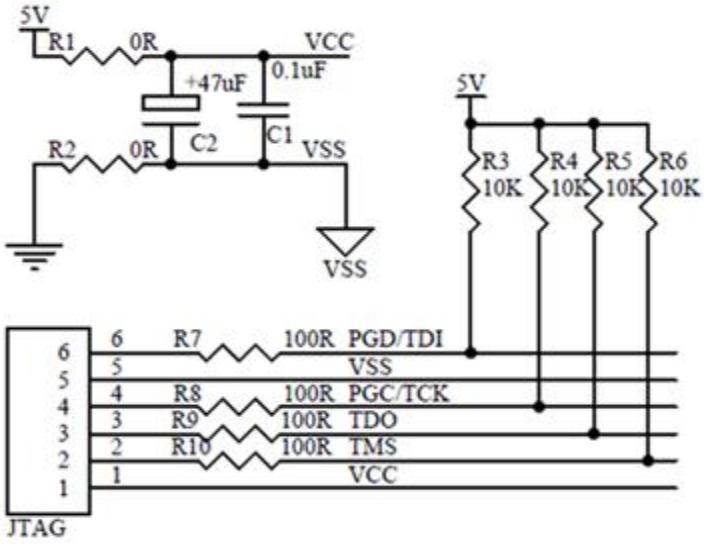
15.4. 字节写步骤

1. EEPROM 擦写时间控制寄存器(SPROG_TIM)SPROG_TIM[7:5] = 3(建议 35us)，SPROG_TIM[4:0] = 1~31(建议 30ms),在主程序 main()函数只配置一次；
2. 关闭中断；
3. 配置 SPROG_ADDR_L 、SPROG_ADDR_H，字节写地址；
4. 配置 SPROG_DATA ；
5. 配置 SPROG_CMD = 0x69;
6. 写入 4 个 NOP 指令；
7. 开始写入，CPU 进入 IDLE 模式，擦除完成后自动退出 IDLE 模式；
8. 需要继续写数据，跳转至第 2 步；
9. 配置 SPROG_ADDR_L=0x00，SPROG_ADDR_H=0x00,恢复中断设置；

第 16 章 烧录调试

16.1. JTAG 电路连接

在进行仿真调试时，需要接 TDI、TCK、TMS、TDO、VCC、VSS 六根线，JTAG 调试模式下，JTAG 口的 IO 功能被屏蔽，建议不要操作配置 JTAG 调试 I/O 口的其它功能，以免影响 JTAG 调试功能。烧录时仅接 TDI、TCK、VCC、VSS 四根线。



JTAG 电路连接参考图

16.2. TouchKey 数据辅助烧录调试

连接芯片 PGC、PGD、VCC、VSS 四根线，进入烧录界面时，选择对应型号的芯片，打开编译好的 HEX 文件，点击一键写 flash 等待烧录完成。

进入调试界面时，先烧录带调试数据发送模式的 HEX 文件，点击开始调试可以查看按键数据。

注：具体操作说明参照 TK 烧录调试指南。

第 17 章 CPU 指令系统

17.1. 指令编码

GDMC191XXXX 的指令分为单字节指令、双字节指令和三字节指令。

单字节指令：单字节指令由 8 位二进制编码构成。指令中只有指令操作码，没有指令操作数或者至指令操作数隐含于指令操作码。该类指令共有 49 条。

双字节指令：双字节指令由两个字节构成，一个为操作码，另一个为操作数(或操作数的地址)，在程序存储器中按顺序存放。该类指令共有 46 条。

三字节指令：三字节指令由一个字节的指令操作码和两个字节的操作数(或操作数的地址)构成。该类指令共有 16 条。

17.2. 指令集

为了描述指令方便，在指令中使用了一些符号，这些符号的含义说明如下：

addr 11	低 11 位地址
addr 16	16 位地址
direct	直接寻址，8 位内部数据及地址(包括特殊功能寄存器)
bit	位地址
#data	8 位立即数
#data16	16 位立即数
rel	带符号的 8 位相对位移量
n	数字 0~7
Rn	当前寄存器组的 R0~R7 工作寄存器
i	数字 0、1
Ri	工作寄存器 R0、R1
@	寄存器间接寻址
←	数据传送方向
∧	逻辑“与”
∨	逻辑“或”
⊕	逻辑“异或”
√	对标志位有影响
×	对标志位无影响

CPU 指令符号含义表

提供使用的汇编指令、各指令的功能、占用的字节数、指令执行周期以及对相应标志位的影响如下表所示：

8 位数据传送类指令								
助记符		功能	对标志位影响				字节数	周期数
			P	OV	AC	CY		
MOV A	Rn	A←(Rn)	√	×	×	×	1	1
	direct	A←(direct)	√	×	×	×	2	1
	@Ri	A←((Ri))	√	×	×	×	1	1
	#data	A←data	√	×	×	×	2	1
MOV Rn	A	Rn←(A)	×	×	×	×	1	1
	direct	Rn←(direct)	×	×	×	×	2	2
	#data	Rn←data	×	×	×	×	2	1
MOV direct1,	A	direct1←(A)	×	×	×	×	2	1
	Rn	direct1←(Rn)	×	×	×	×	2	2
	direct2	direct1←(direct2)	×	×	×	×	3	2
MOV direct,	@Ri	direct←((Ri))	×	×	×	×	2	2

	#data	direct←data	×	×	×	×	3	2
MOV @Ri	A	(Ri)←(A)	×	×	×	×	1	1
	direct	(Ri)←(direct)	×	×	×	×	2	2
	#data	(Ri)←data	×	×	×	×	2	1
16 位数据传送类指令								
助记符		功能	对标志位影响				字节数	周期数
			P	OV	AC	CY		
MOV DPTR,#data16		DPTR←data16	×	×	×	×	3	2
外部数据传送与查表类指令								
助记符		功能	对标志位影响				字节数	周期数
			P	OV	AC	CY		
MOVX @DPTR,A		(DPTR)←(A)	×	×	×	×	1	2
MOVC A,	@A+DPTR	A←((A)+(DPTR))	√	×	×	×	1	2
	@A+PC	A←((A)+(PC))	√	×	×	×	1	2
MOVX A,	@DPTR	A←(DPTR)	√	×	×	×	1	2
注：MOVX 指令的周期数以及字节数可通过寄存器 CKCON<2:0>配置								
交换类指令								
助记符		功能	对标志位影响				字节数	周期数
			P	OV	AC	CY		
XCH A,	Rn	(Rn)←(A)	√	×	×	×	1	2
	direct	(A)←(direct)	√	×	×	×	2	1
	@Ri	(A)←((Ri))	×	×	×	×	1	1
XCHD A,@Ri		(A)3~0~((Ri))3~0	√	×	×	×	1	1
SWAP A		(A)7-4~(A)3-0	√	×	×	×	1	1
算术运算类指令								
助记符		功能	对标志位影响				字节数	周期数
			P	OV	AC	CY		
ADD A,	Rn	A←(A)+(Rn)	√	√	√	√	1	1
	direct	A←(A)+(direct)	√	√	√	√	2	1
	@Ri	A←(A)+((Ri))	√	√	√	√	1	1
	#data	A←(A)+data	√	√	√	√	2	1
ADDC A,	Rn	A←(A)+(Rn)+(C)	√	√	√	√	1	1
	direct	A←(A)+(direct)+(C)	√	√	√	√	2	1
	@Ri	A←(A)+((Ri)) +(C)	√	√	√	√	1	1
	#data	A←(A)+data+(C)	√	√	√	√	2	1
INC	A	A←(A)+1	√	×	×	×	1	1
	Rn	Rn←(Rn)+1	×	×	×	×	1	1
	direct	direct←(direct)+1	×	×	×	×	2	1

	@Ri	(Ri)←((Ri))+1	×	×	×	×	1	1
	DPTR	DPTR←((DPTR))+1	×	×	×	×	1	2
DA A		BCD 码调整	√	×	√	√	1	1
SUBB A	Rn	A←(A)-(Rn)-(C)	√	×	×	×	1	1
	direct	A←(A)-(direct)-(C)	√	√	√	√	2	1
	@Ri	(A)←(A)-((Ri))-(C)	√	√	√	√	1	1
	#data	A←(A)-data-(C)	√	√	√	√	2	1
DEC	A	A←(A)-1	√	×	×	×	1	1
	Rn	Rn←(Rn)-1	×	×	×	×	1	1
	direct	direct←(direct)-1	×	×	×	×	2	1
	@Ri	(Ri)←((Ri))-1	×	×	×	×	1	1
MUL AB		BA←(A)*(B),执行乘法运算后，结果低字节存于 A，高字节存于 B	√	√	×	0	1	4
DIV AB		A←(A)/(B) B←余数	√	√	×	0	1	4
注：DA 指令使用时，，调整规则如下：若累加器 A 低 4 位大于 9 或者 AC=1，则 A←A+06H;若累加器 A 高 4 位大于 9 或者 CY=1，则 A←A+60H								
逻辑运算类指令								
助记符		功能	对标志位影响				字节数	周期数
			P	OV	AC	CY		
CLR A		A←00H	√	×	×	×	1	1
CPL A		A←($\overline{A}$)	√	×	×	×	1	1
ANL A,	Rn	A←(A)∧(Rn)	√	×	×	×	1	1
	direct	A←(A)∧(direct)	√	×	×	×	2	1
	@Ri	A←(A)∧((Ri))	√	×	×	×	1	1
	#data	A←(A)∧data	√	×	×	×	2	1
ANL direct,	A	direct←(A)∧(direct)	×	×	×	×	2	1
	#data	direct←(direct)∧data	×	×	×	×	3	2
ORL A,	Rn	A←(A)∨(Rn)	√	×	×	×	1	1
	direct	A←(A)∨(direct)	√	×	×	×	2	1
	@Ri	A←(A)∨((Ri))	√	×	×	×	1	1
	#data	A←(A)∨data	√	×	×	×	2	1
ORL direct,	A	direct←(direct)∨(A)	×	×	×	×	2	1
	#data	direct←(direct)∨	×	×	×	×	3	2

		data						
XRL A,	Rn	A←(A)⊕(Rn)	√	×	×	×	1	1
	direct	A←(A)⊕(direct)	√	×	×	×	2	1
	@Ri	A←(A)⊕((Ri))	√	×	×	×	1	1
	#data	A←(A)⊕data	√	×	×	×	2	1
XRL direct,	A	direct←(direct)⊕(A)	×	×	×	×	2	1
	#data	direct←(direct)⊕data	×	×	×	×	3	2

循环、移位类指令

助记符	功能	对标志位影响				字节数	周期数
		P	OV	AC	CY		
RL A	A 中内容循环左移一位	×	×	×	×	1	1
RLC A	A 中内容带进位循环左移一位	√	×	×	√	1	1
RR A	A 中内容循环右移一位	×	×	×	×	1	1
RRC A	A 中内容带进位循环右移一位	√	×	×	√	1	1

调用、返回类指令

助记符	功能	对标志位影响				字节数	周期数
		P	OV	AC	CY		
LCALL addr16	(PC)←(PC)+3,(SP)←(PC),(PC)←addr16	×	×	×	×	3	2
ACALL addr11	(PC)←(PC)+2,(SP)←(PC),(PC10~0)←addr11	×	×	×	×	2	2
RET	(PC)←((SP))	×	×	×	×	1	2
RETI	(PC)←((SP))从中断返回	×	×	×	×	1	2

转移类指令

助记符	功能	对标志位影响				字节数	周期数
		P	OV	AC	CY		
LJMP addr16	PC←addr15~0	×	×	×	×	3	2
AJMP addr11	PC10~0←addr10~0	×	×	×	×	2	2
SJMP rel	PC←(PC)+rel	×	×	×	×	2	2
JMP @A+DPTR	PC←(A)+(DPTR)	×	×	×	×	1	2
JZ rel	PC←(PC)+2,	×	×	×	×	2	2

		若 (A)=0,PC←(PC)+rel						
JNZ	rel	PC←(PC)+2, 若 (A)≠0,PC←(PC)+rel	×	×	×	×	2	2
JC	rel	PC←(PC)+2, 若 (CY)=1,PC←(PC)+rel	×	×	×	×	2	2
JNC rel		PC←(PC)+2, 若 (CY)=0,PC←(PC)+rel	×	×	×	×	2	2
JB	bit,rel	PC←(PC)+3, 若 (bit)=1,PC←(PC)+rel	×	×	×	×	3	2
JNB bit,rel		PC←(PC)+3, 若 (bit)=0,PC←(PC)+rel	×	×	×	×	3	2
JBC bit,rel		PC←(PC)+3, 若(bit)=1,则 bit←0, PC←(PC)+rel	×	×	×	×	3	2
CJNE	A, direct,rel	PC←(PC)+3, 若(A) ≠direct 则 PC(PC)+rel 若(A)<(direct),则 CY←1	×	×	×	×	3	2
	A,#data,rel	PC←(PC)+3, 若(A) ≠data 则 PC(PC)+rel 若(A)<(data),则 CY←1	×	×	×	×	3	2
	Rn,#data,rel	PC←(PC)+3, 若(Rn) ≠data 则 PC←(PC)+rel 若(Rn)<(data),则	×	×	×	×	3	2

		CY←1						
	@Ri,#data,rel	PC←(PC)+3,若((Ri)) ≠data 则PC←(PC)+rel 若((Ri))<(data),则CY←1	×	×	×	×	3	2
DJNZ	Rn,rel	PC←(PC)+2,Rn←(Rn)-1, 若(Rn) ≠0,则PC←(PC)+rel	×	×	×	×	2	2
	direct,rel	PC←(PC)+3,(direct)←(direct)-1, 若(direct) ≠0, 则 PC←(PC)+rel	×	×	×	×	3	2

堆栈、空操作类指令

助记符	功能	对标志位影响				字节数	周期数
		P	OV	AC	CY		
PUSH direct	SP←(SP)+1,(SP)←(direct)	×	×	×	×	2	2
POP direct	direct←(SP),SP←(SP)-1	×	×	×	×	2	2
NOP	空操作	×	×	×	×	1	1

位操作类指令

助记符		功能	对标志位影响				字节数	周期数
			P	OV	AC	CY		
MOV	C,bit	CY←bit	×	×	×	√	2	1
	bit,C	bit←CY	×	×	×	×	2	1
CLR	C	CY←0	×	×	×	√	1	1
	bit	bit←0	×	×	×	×	2	1
SETB	C	CY←1	×	×	×	√	1	1
	bit	bit←1	×	×	×	×	2	1
CPL	C	CY←($\overline{CY}$)	×	×	×	√	1	1
	bit	bit←($\overline{bit}$)	×	×	×	×	1	1
ANL	C,bit	C←(C)∧(bit)	×	×	×	√	2	2
	C ,/bit	C←(C)∧($\overline{bit}$)	×	×	×	√	2	2
ORL	C,bit	C←(C)∨(bit)	×	×	×	√	2	2
	C,/bit	C←(C)∨($\overline{bit}$)	×	×	×	√	2	2

伪指令

助记符	指令格式	功能说明

ORG	【标号：】ORG addr16	规定标号的起始地址
EQU	标号 EQU 数值或标号	为标号赋值
DB	【标号：】 DB 项或项表	用于定义内存一个单元或一批单元的字节内容
DW	【标号：】 DW 项或项表	用于定义内存某两单元或多个两个单元构成的 16 位字内容
DS	【标号：】DS 表达式	规定从标号开始留下若干个存储单元
BIT	标号 BIT 位地址	把位地址赋给标号
END	END 放在汇编语言程序的最后，用以告诉汇编程序，源程序到此为止。没有 END 结束的源程序将进入死循环	

CPU 指令集表

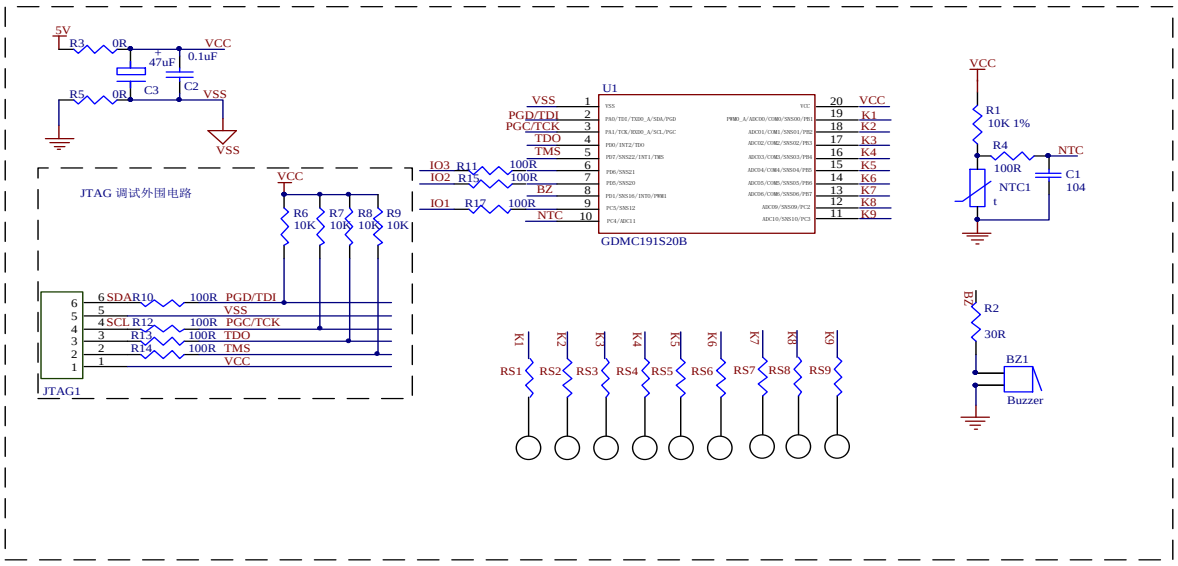
CPU 相关寄存器

SFR 寄存器				
地址	名称	读写	复位值	说明
0x81	SP	RW	0x07	堆栈指针寄存器
0x82	DPL	RW	0x00	数据指针寄存器 0 低 8 位
0x83	DPH	RW	0x00	数据指针寄存器 0 高 8 位
0x87	PCON	RW	0x30	低功耗模式选择寄存器
0xE0	ACC	RW	0x00	累加器
0xF0	B	RW	0x00	B 寄存器

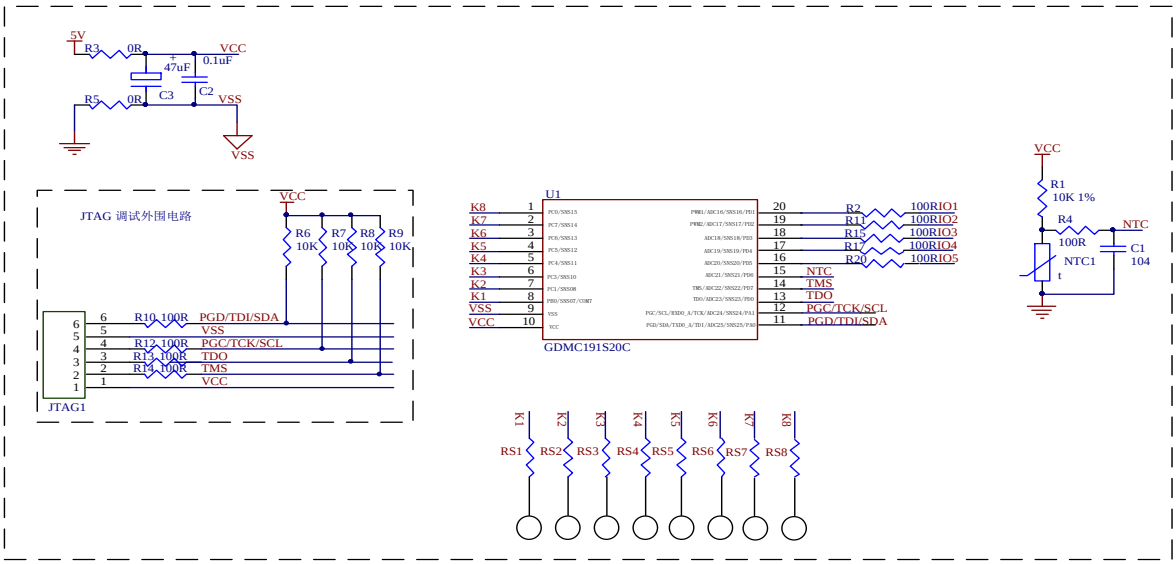
CPU SFR 寄存器列表

第 18 章 参考应用电路

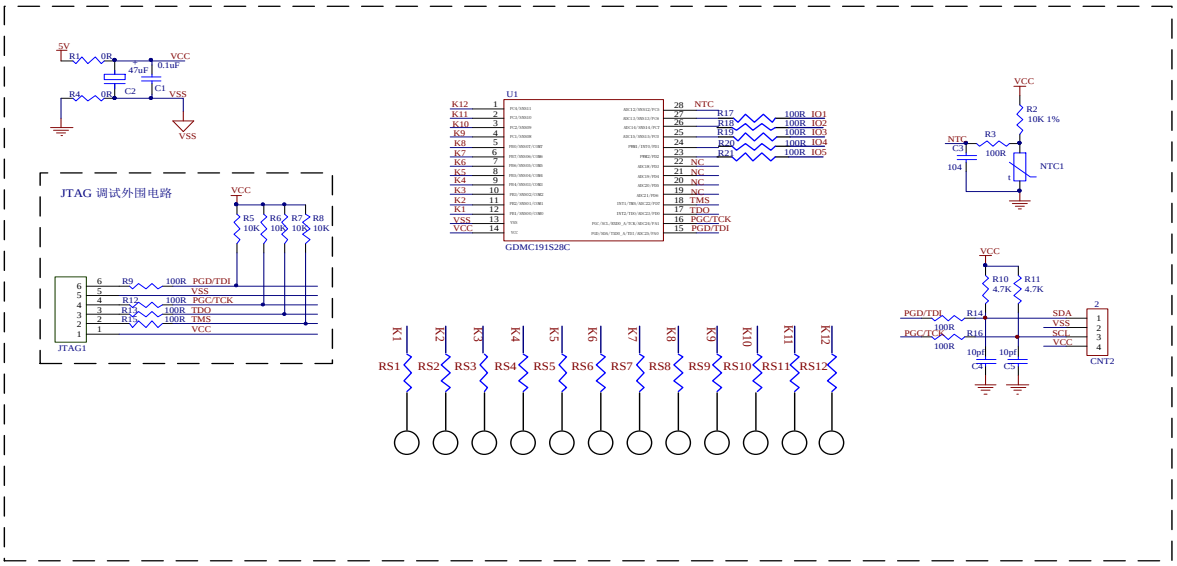
18.1.GDMC191S20B 参考电路



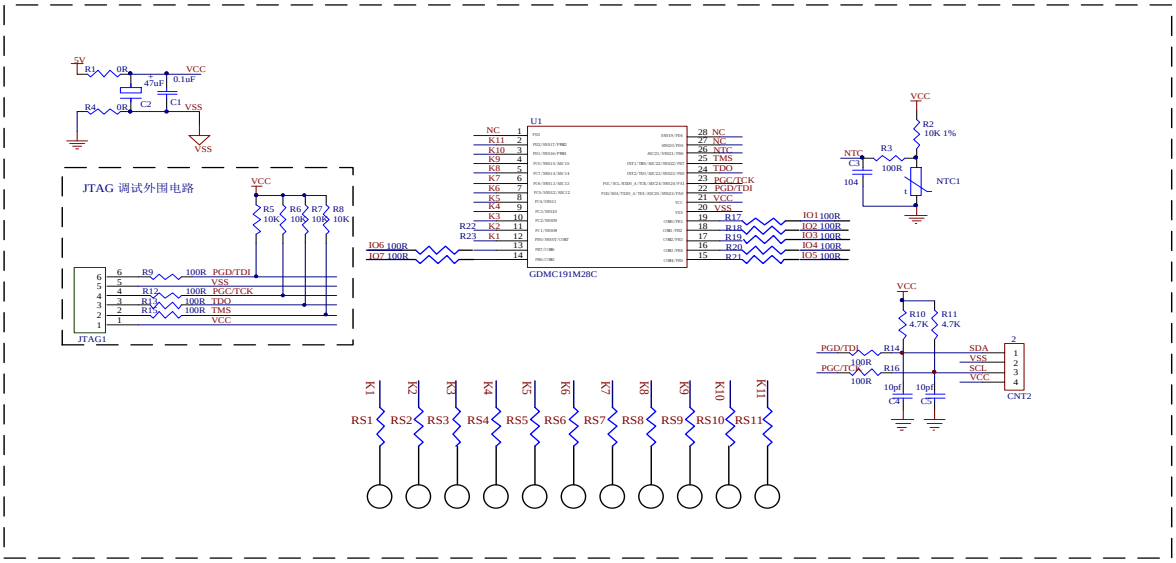
18.2. GDMC191S20C 参考电路



18.3. GDMC191S28C 参考电路



18.4. GDMC191M28C 参考电路



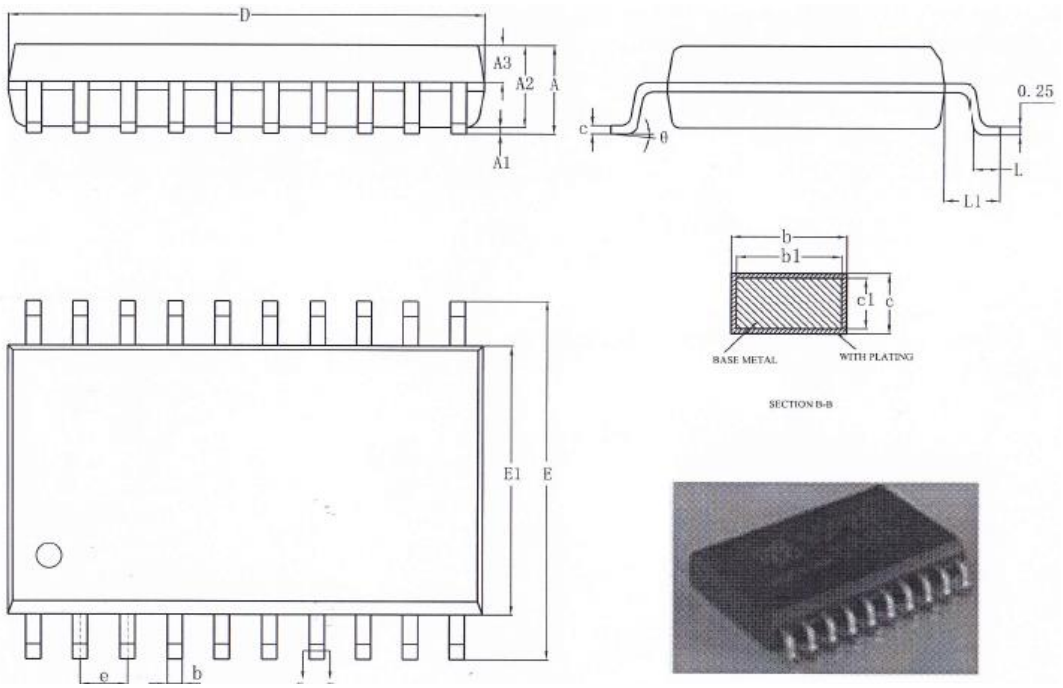
注：

1、 以上原理图仅供参考，RSX 通道电阻建议 100 欧~3K。

2、 PS：JTAG调试外围电路仅JTAG调试用，若仿真器或转接板上已有上拉电阻，则无需接JTAG上接电阻。

第 19 章 封装信息

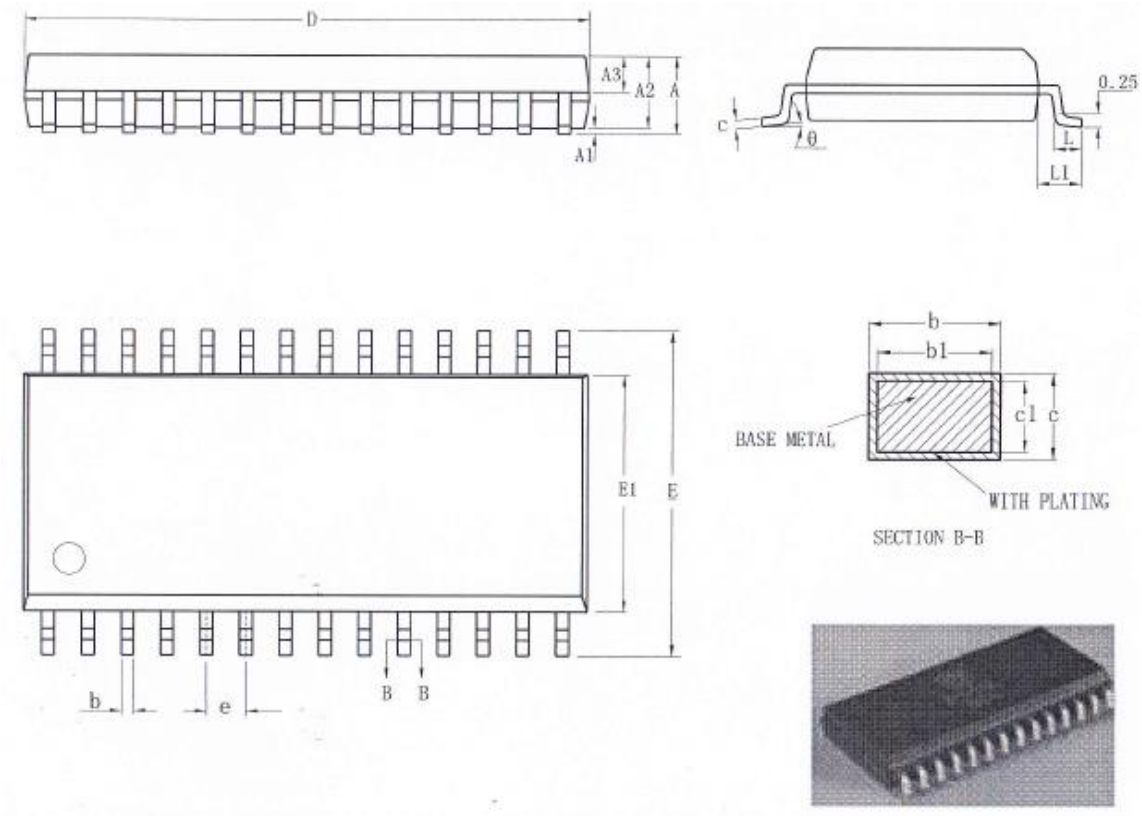
19.1. SOP20



SOP20 封装信息图

DIM	SOP020 MILLIMETERS		
	MIN	NOM	MAX
A	-	-	2.650
A1	0.100	0.200	0.300
A2	2.250	2.300	2.350
b	0.350	-	0.440
c	0.250	-	0.310
D	12.600	12.800	13.000
E1	7.300	7.500	7.700
E	10.100	10.300	10.500
e	1.270(BSC)		
L	0.7	-	1
θ	0°	-	8°
端面废胶	-	-	0.175
塑封体总长度	12.800	13.000	13.300

19.2. SOP28



SOP28 封装信息图

DIM	SOP028 MILLIMETERS		
	MIN	NOM	MAX
A	2.250	2.400	2.650
A1	0.100	0.200	0.300
A2	2.250	2.300	2.350
b	0.300	0.425	0.480
c	0.250	0.285	0.310
D	17.800	18.000	18.200
E1	7.300	7.500	7.700
E	10.100	10.300	10.500
e	1.270(BSC)		
L	0.7	-	1
θ	0°	-	8°
端面废胶	-	-	0.175
塑封体总长度	18.000	18.300	18.500

改版记录

改版日期	改版内容	改版人	备注
2019-07-19	初版	XS	V1.0
2019-07-26	1. 增加 IO COM 大电流口描述。	XS	V1.1

免责声明

1、此文档中的信息可以在不通知用户时进行修改及更新
2、公司将竭尽最大的努力保证本公司产品的高质量与高稳定性。尽管如此，由于一般半导体器件的电气敏感性及易受到外部物理伤害等固有特点，本公司产品有可能在这些情况下出现故障或失效。当使用本公司产品时，使用者有责任遵从安全规则来设计一个安全及稳定的系统环境。使用者可通过去除多余器件、故障预防及火灾预防等措施来避免可能发生的意外、火灾及公共伤害。在用户使用该产品时，请遵从本公司最新说明书上规定的操作步骤来使用该产品。
3、在此文档中的公司的产品是为一般电气应用(电脑、个人工具、办公司工具、测量工具、工业机械器件、家用电器等)所设计的。本公司该产品不能及禁止应用在一些需要极高稳定性及质量的特殊设备上，以免导致人员伤亡等意外发生。产品不能应用范围包括原子能控制设备、飞机及航空器件、运输设备、交通信号设备、燃烧控制设备、医药设备以及所有安全性设备等等。使用者在以上列举的非产品应用范围内使用时造成的损失与伤害，本公司概不负责。

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for EEPROM ***category:***

Click to view products by BYD ***manufacturer:***

Other Similar products are found below :

M29F040-70K6 718278CB 718620G AT28C256-15PU-ND 444358RB 444362FB BR93C46-WMN7TP 442652G 701986CB TC58NVG0S3HBAI4 5962-8751413XA TC58BVG0S3HBAI4 TH58NYG3S0HBAI6 CAT25320YIGT-KK CAT25320DWF LE24C162-R-E 5962-8751417YA 5962-8751409YA CAT25M01LI-G DS28E11P+ BR9016AF-WE2 LE2464DXATBG CAS93C66VP2I-GT3 DS28E25+T DS28EL15Q+T M95320-DFDW6TP DS28E05GB+T AT25320B-SSPDGV-T HE24C64WLCSPD BL24SA128B-CSRC 24FC16T-I/OT 24FC08T-I/OT M24128-BFMN6TP S-24CS04AFM-TFH-U M24C04-FMC5TG M24C16-DRMN3TPK M24C64-DFMN6TP 34AA02-EMS M95080-RMC6TG M95128-DFCS6TP/K M95128-DFDW6TP M95256-DFMN6TP M95320-RDW6TP M95640-RDW6TP AT17LV010-10CU AT24C01C-SSHM-B AT24C01D-MAHM-T AT24C04D-MAHM-T AT24C04D-SSHM-T AT24C08C-SSHM-B