



# JZ51F9003

## 数据手册

版本 V1.0



## 1. 总体概述

本芯片是一款高性能的 8 位单片机。内部包含 16K 字节 Flash, 1K 字节 SRAM, 128 字节 EEPROM, 1 个 8 位定时器、3 个 16-bit 定时器/计数器, 具有独立时钟的看门狗, 1 个 12-bit ADC, 2 个模拟比较器, 2 路 UART 和 1 路 I2C 和 1 路 SPI 通讯接口, 片内 POR, BOR 和 LVD, 内部 16MHz RC 振荡器和 32KHz WDT 振荡器。具有两种低功耗式。

## 2. 主要功能

### 内核:

- 超高速 8051 内核 (1T)
- 指令周期可配:  
2.6MHz, VDD $\geq$ 2.4V  
16MHz, VDD $\geq$ 4.5V

工作电压: 2.4V~5.5V

工作温度: -40°C~85°C

Flash ROM: 16K 字节 Flash ROM, 用于存储用户代码(擦写次数典型值 1000 次)

EEPROM: 128 字节(擦写次数 10000 次)

SRAM: 1K 字节 SRAM

### 时钟:

- 内部 16MHz RC 振荡器(可微调)  
误差不超过 $\pm 2\%$ (-40°C~85°C)
- 内部 32KHz 低速 RC 振荡器(误差不超过 $\pm 10\%$ )

### 复位:

- 上电复位、复位脚复位、看门狗溢出复位
- 欠压复位(2.19V、2.51V、2.88V、3.58V、4.21V)

低电压检测: LVD 共 5 级 (2.32V、2.75V、3.03V、3.81V、4.54V)

### 中断(INT):

- Timer0、Timer1、Timer2、Timer3、SCK3、WDT、ADC、UART0~1、SPI、I2C、LVD、CMP0~1、P0~P2 共 17 个中断源, 全部 GPIO 可设上升沿、下降沿、高电平、低电平中断。

### 数字外设:

- 1 个 8 位基本定时器  
预分频 1、2、4、8、16、32、64、128
- 3 个 16 位高级定时器, 支持 6 路 PWM 输出功能  
支持捕获和刹车功能  
支持周期中断和占空比中断
- 1 个 16 位看门狗定时器
- 2 路 UART  
全双工、支持单线半双工  
支持 9bit 数据位
- 1 路 SPI: 支持 RZ 编码输出
- 1 路 I2C: 支持主机模式和从机模式

速率 100KHz/400KHz

### 12 位 ADC:

- 外部输入: 7 路
- 内部输入: 1 路(1/4 VDD)
- 参考源: 外部参考、内部参考 1.2V/2.4V 和电源参考
- 采样可以通过 PWM 或者管脚的上升沿或者下降沿触发

### 模拟比较器:

- 2 个比较器

5 位的 DAC, 参考电压来自 VBG (1.2V) 或者电

### 源

### 18 个 GPIO:

- P0[7:0]、P1[7:0]、P2[1:0]
- PT02、PT16 默认开漏上拉输出, 其余 I/O 默认为输入高阻态
- 所有 IO 可单独配置上下拉 10K 电阻(匹配精度 5%)  
上下拉可同时打开

### 省电模式:

- 深度休眠可由看门狗复位、睡眠定时器中断、引脚中断唤醒。
- 深度休眠电流: 4.2uA (典型值)

### 支持 ICP、IAP (应用编程) 功能:

- 支持在线调试仿真和 IAP 功能
- ICP 功能通过 I2C 接口实现, 配合烧写器可以在线更新程序代码
- ISP/IAP 模式通过内部程序执行可以完成代码升级或者对程序空间进行写入操作, 支持对 ISP/IAP 模式的锁定操作, 即只有解锁成功才可以进行 ISP/IAP 操作

### 支持在线调试功能:

- 可以通过 4 线 JTAG 或 2 线调试接口实现程序的在线调试

### 封装:

- SOP16/TSSOP20/QFN20(3\*3)



## 目 录

1. 总体概述 .....	2
2. 主要功能 .....	2
3. 系统功能框图及脚位图 .....	5
3.1 脚位图 .....	5
3.2 引脚描述 .....	6
4. 端口结构 .....	8
4.1 结构框图 .....	8
4.2 配置 I/O 口 .....	9
4.3 外设功能管脚 .....	9
4.4 与 GPIO 相关寄存器定义 .....	10
5. CPU .....	21
5.1 CPU 内核概述 .....	21
5.2 CPU 内核 SFR 寄存器 .....	21
6. 存储器 .....	25
6.1 程序存储器 .....	25
6.2 数据存储器 .....	26
6.3 SFR 空间 .....	27
6.4 XDATA 空间 .....	27
6.5 FLASH 控制器 .....	28
7. 中断控制器 .....	32
7.1 概述 .....	32
7.2 GPIO 中断 .....	32
7.3 中断向量表 .....	32
7.4 中断优先级和中断屏蔽 .....	33
7.5 中断触发方式 .....	33
7.6 与中断相关寄存器定义 .....	33
8. 时钟 .....	37
8.1 概述 .....	37
8.2 时钟结构框图 .....	37
8.3 CPU 时钟 .....	38
8.4 SCK1 和 SCK2 时钟 .....	38
8.5 SCK3 时钟 .....	38
8.6 32K 时钟 .....	38
8.7 与时钟相关寄存器定义 .....	38
9. 复位 .....	41
9.1 POR 复位 .....	41
9.2 引脚复位 .....	41
9.3 看门狗复位 .....	41
9.4 欠压复位 .....	41
10. 外设 .....	42
10.1 8-bit 基本计数器 .....	42
10.2 16-bit 高级计数器 .....	44
10.3 UART .....	77
10.4 I2C .....	83
10.5 12-bit ADC .....	88
10.6 SPI .....	93
10.7 模拟比较器 .....	98
11. 省电模式和看门狗 .....	103
11.1 省电模式 .....	103
11.2 睡眠模式 .....	103



---

---

11.3	看门狗 .....	104
11.4	睡眠定时器中断 .....	104
11.5	与省电模式和看门狗相关寄存器定义 .....	104
12.	在线调试 .....	108
13.	配置选项 .....	109
13.1	系统控制 .....	109
13.2	模拟控制 .....	110
14.	电气特性 .....	113
14.1	绝对最大额定值 .....	113
14.2	直流特性 .....	113
14.3	ADC 特性 .....	114
14.4	比较器特性 .....	114
14.5	EMC 特性 .....	114
15.	芯片封装信息 .....	115

### 3. 系统功能框图及脚位图

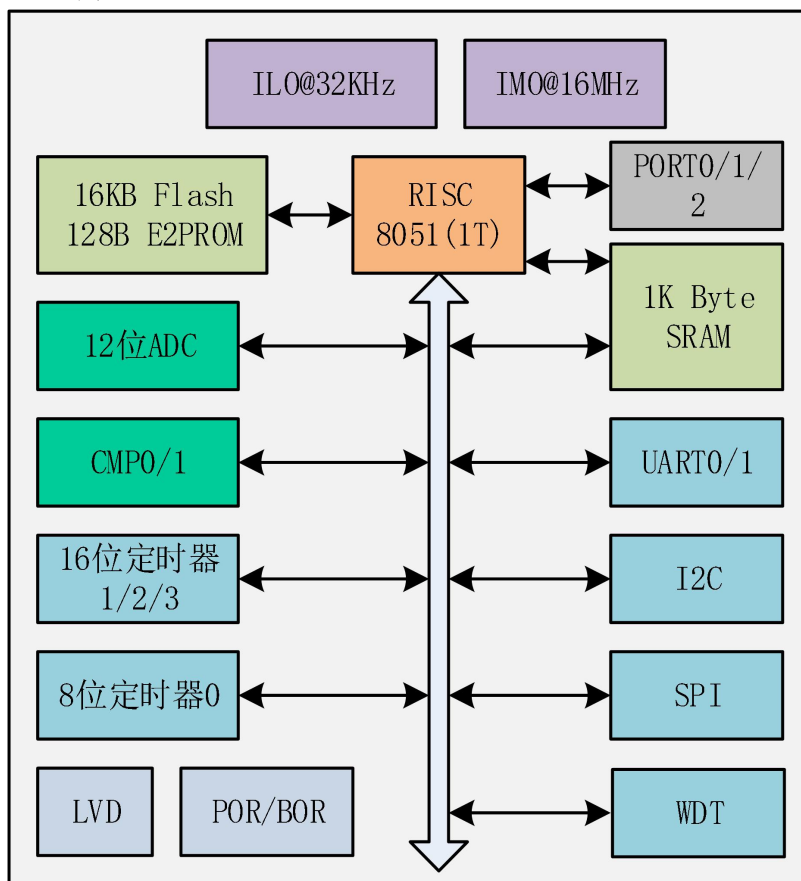


图 1 芯片整体结构框图

#### 3.1 脚位图

VDD	1	16	GND
CON1/[TIM3_CHB]/SPI_NSS/P15	2	15	P16/SDA/TXD1/TDI
TDO/[SDA]/BKIN/[TIM1_CHB]/P14	3	14	P17/[TIM1_CHA]
TMS/[SCL]/[ADC_ETR]/P13	4	13	P21/OSCIN/C1N0
SCK1/TIM2_CHA/SPI_SCK/CO_0/P10	5	12	P20/RST_N
CLK_MTP/TIM2_CHB/SPI_MOSI/C1_0/P00	6	11	P07/AIN6/RXD/C1P0
MTP_TEST/TIM3_CHA/SPI_MISO/P01	7	10	P06/AIN5/TXD/COPO
TCK/[TIM2_CHA]/RXD1/SCL/AIN1/P02	8	9	P04/AIN3/ADC_ETR/[TIM3_CHA]

图 2 SOP16 封装脚位图

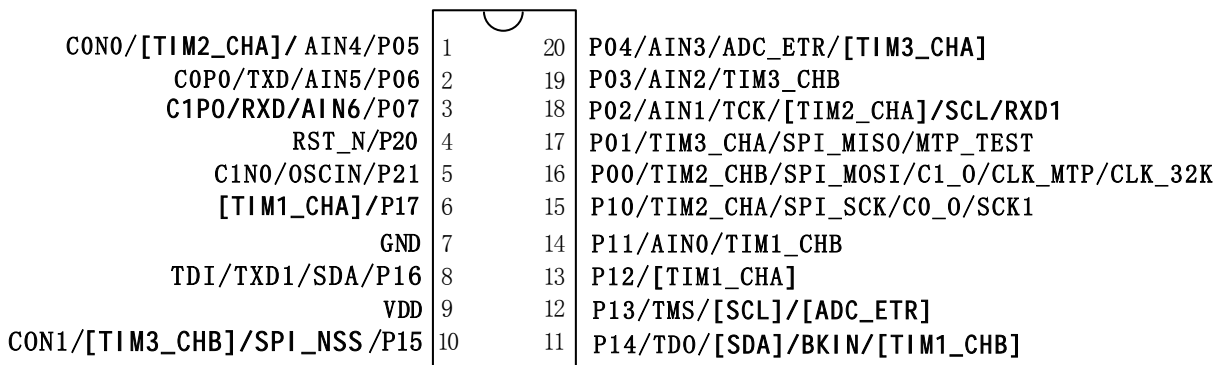


图 3 TSSOP20 封装脚位图

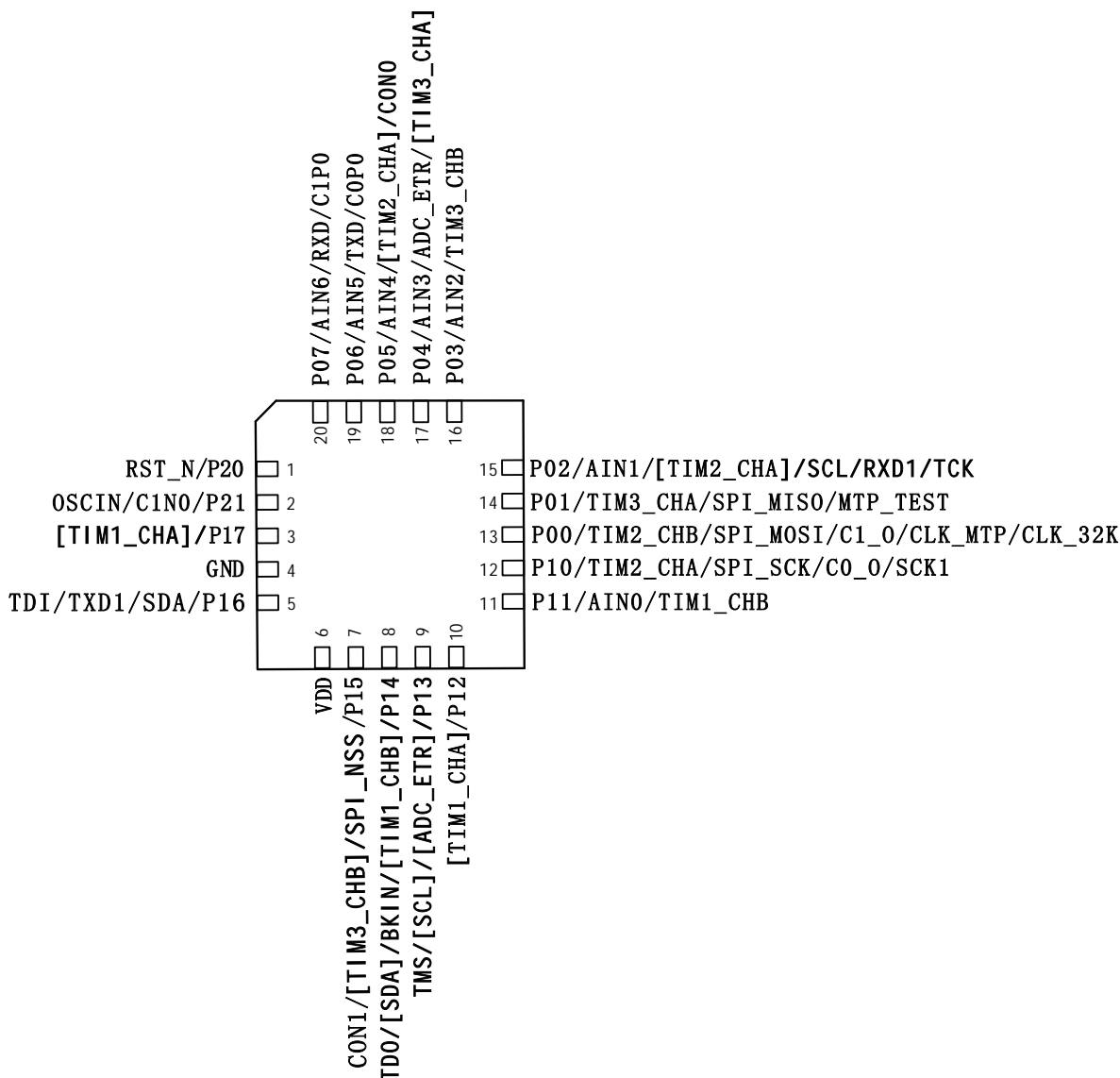


图 4 QFN20 封装脚位图

### 3.2 引脚描述

引脚名称	I/O 类型	说明
GPIO 端口（所有的 GPIO 都可以产生中断）		



P0.0~P0.7	I/O	8 位双向 I/O 端口 P0
P1.0~P1.7	I/O	8 位双向 I/O 端口 P1
P2.0~P2.1	I/O	2 位双向 I/O 端口 P2
<b>PWM 控制器</b>		
TIM1_CHA	I/O	16 位 PWM 定时器 TIMER1 输出引脚 A, 16 位输入捕获引脚 A
TIM1_CHB	I/O	16 位 PWM 定时器 TIMER1 输出引脚 B, 16 位输入捕获引脚 B
TIM2_CHA	I/O	16 位 PWM 定时器 TIMER2 输出引脚 A, 16 位输入捕获引脚 A
TIM2_CHB	I/O	16 位 PWM 定时器 TIMER2 输出引脚 B, 16 位输入捕获引脚 B
TIM3_CHA	I/O	16 位 PWM 定时器 TIMER3 输出引脚 A, 16 位输入捕获引脚 A
TIM3_CHB	I/O	16 位 PWM 定时器 TIMER3 输出引脚 B, 16 位输入捕获引脚 B
BKIN	I	PWM 刹车输入引脚
<b>UART</b>		
TXD	O	UART 数据输出引脚
RXD	I	UART 数据输入引脚
TXD1	O	UART1 数据输出引脚
RXD1	I	UART1 数据输入引脚
<b>SPI</b>		
SPI_NSS	I/O	SPI 片选引脚
SPI_MISO	I/O	SPI MISO 引脚
SPI_MOSI	I/O	SPI MOSI 引脚
SPI_SCK	I/O	SPI 时钟引脚
<b>I2C</b>		
SCL	I/O	I2C 时钟引脚
SDA	I/O	I2C 数据引脚(注意:上电 2.1MS, 该脚会输出 50uS 左右的低电平)
<b>时钟和电源</b>		
OSCIN	I	外部输入时钟引脚
VDD	POWER	电源(2.4V~5.5V)
GND	POWER	接地
<b>JTAG</b>		
TCK	I	JTAG 时钟输入引脚
TDI	I	JTAG 数据输入引脚
TDO	O	JTAG 数据输出引脚
TMS	I	JTAG 模式选择输入引脚
<b>两线调试</b>		
TCK	I/O	两线调试时钟引脚
TDI	I/O	两线调试数据引脚
<b>ADC</b>		
AIN0~AIN6	I	ADC 外部采样输入通道
ADC_ETR	I	ADC 外部触发采样输入引脚
<b>比较器</b>		
CON0, CON1	I	比较器 0 N 端输入引脚
COPO	I	比较器 0 P 端输入引脚
CO_0	O	比较器 0 比较结果输出引脚
C1N0	I	比较器 1 N 端输入引脚
C1P0	I	比较器 1 P 端输入引脚
C1_0	O	比较器 1 比较结果输出引脚
<b>外部复位引脚</b>		
RST_N	I	外部复位引脚

## 4. 端口结构

### 4.1 结构框图

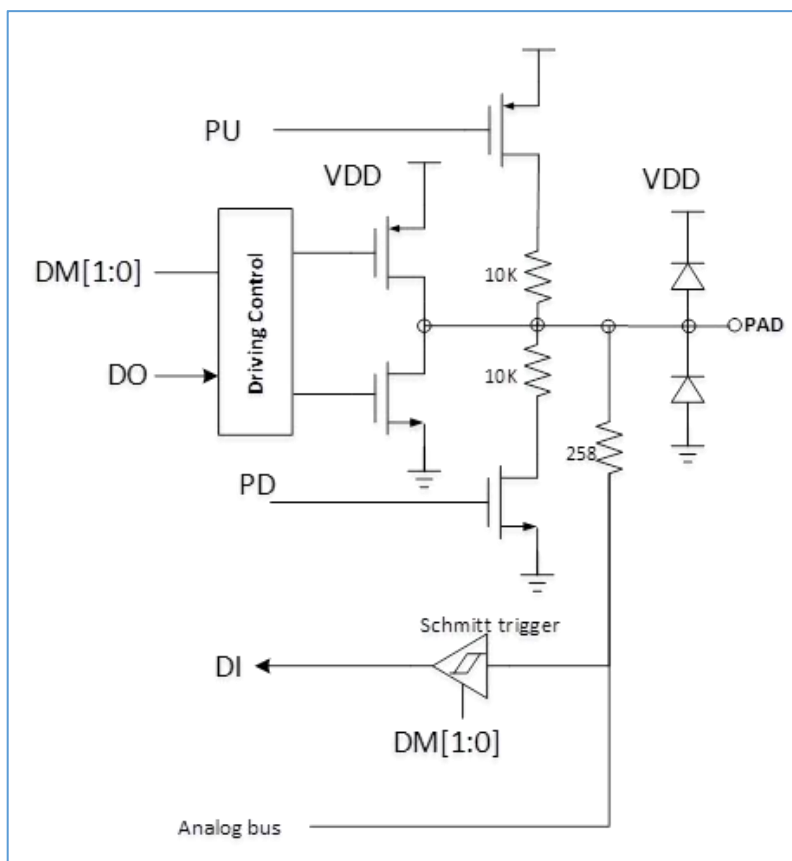


图 5 GPIO 结构图



## 4.2 配置 I/O 口

每个 I/O 使用两个寄存器进行配置输入和输出模式。

以 P0 口为例，配置 P0 口需要使用 P0\_DM0 和 P0\_DM1 两个寄存器进行配置，如下图所示：

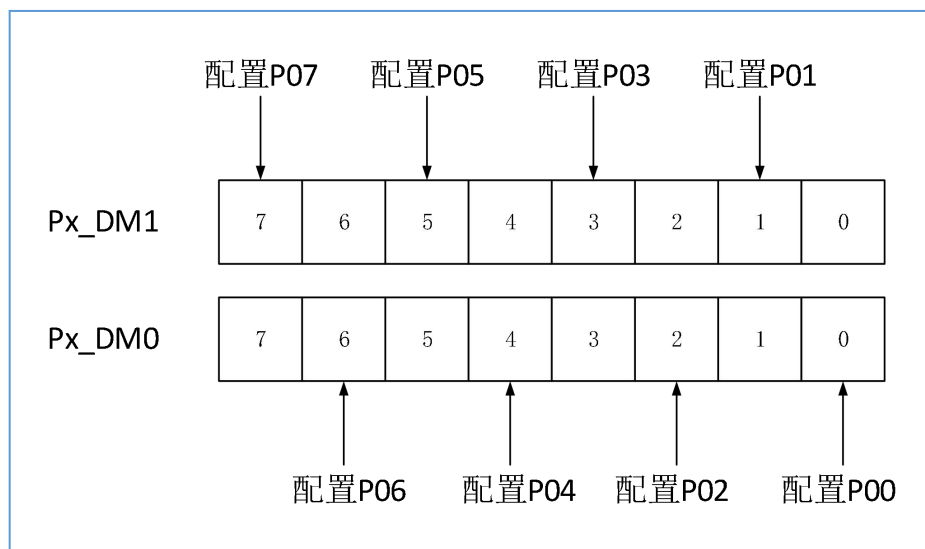


图 6 I/O 配置图

即 P0\_DM0 的第 0 位和 P0\_DM1 的第 0 位合起来配置 P00 的模式；

即 P0\_DM0 的第 1 位和 P0\_DM1 的第 1 位合起来配置 P01 的模式；

GPIO 模式的配置说明如下表和下图所示：

表 1 GPIO 驱动模式

Px_DM1	Px_DM0	驱动模式	施密特开关	应用场景
0	0	配置 PTx 的对应 I/O 为高阻输出，此时数字输入使能打开，此时读 DR 寄存器读到的是 PAD 电平值	ON	数字输入/ UART RX
0	1	配置 PTx 的对应 I/O 为强推挽输出，此时数字输入使能关闭，此时读 DR 寄存器读到的是 DR 寄存器的值	OFF	数字输出/ UART TX/ PWM
1	0	配置 PTx 的对应 I/O 为模拟输入，此时数字输入使能关闭，此时读 DR 寄存器读到的是 DR 寄存器的值	OFF	模拟信号
1	1	配置 PTx 的对应 I/O 为开漏低输出，此时数字输入使能打开，此时读 DR 寄存器读到的是 PAD 电平值 <sup>注1</sup>	ON	I2C

注 1：当 GPIO 配置成开漏低输出时，需要配合使能内部上拉或者接外部上拉，此时如果管脚输出低电平，则在该管脚上会形成上拉电阻到地的通路，会有大概 VDD/10K 的电流（比如 5V 电源供电，则会有 500uA 电流）通过。在系统进入 deepsleep 时需要注意开漏的管脚要避免输出低电平。

## 4.3 外设功能管脚

本芯片支持数字外设功能使用不同的管脚位置，通过 Px\_GE，PT\_SEL，PER0\_PEN 和 PER1\_PEN 寄存器来进行设置。完整的管脚映射见第 3.1 章中脚位图。



- Px\_GE 相应 bit 为 1 时使能对应管脚的数字外设功能，否则对应管脚为普通的 GPIO 功能；
- PT\_SEL 寄存器可以修改部分数字外设的管脚位置，具体见寄存器说明；
- PERPO\_EN 和 PERP1\_EN 寄存器用来使能部分数字外设的管脚功能。

## 4.4 与 GPIO 相关寄存器定义

名字	地址	读写	复位值	描述
PT_SEL0	0xFF10	读写	00000000	端口位置配置寄存器 0
PT_SEL1	0xFF11	读写	00000000	端口位置配置寄存器 1
PERPO_EN	0xFF18	读写	00000000	外设管脚位置使能配置寄存器 0
PERP1_EN	0xFF19	读写	00000100	外设管脚位置使能配置寄存器 1
PERP2_EN	0xFF1A	读写	00000000	外设管脚位置使能配置寄存器 2
P0_DR	0x98	读写	00000000	端口 0 数据寄存器
P0_DM0	0x99	读写	00000100	端口 0 模式 0 位
P0_DM1	0x9A	读写	00000100	端口 0 模式 1 位
P0_FLAG	0xFF20	读写	00000000	端口 0 中断标志位
P0_GE	0xFF21	读写	00000100	端口 0 数字复用使能寄存器
P0_PU	0xFF23	读写	00000100	端口 0 上拉控制寄存器
P0_PD	0xFF24	读写	00000000	端口 0 下拉控制寄存器
P0_IE	0xFF25	读写	00000000	端口 0 中断使能寄存器
P0_IC0	0xFF26	读写	00000000	端口 0 中断控制 0 位
P0_IC1	0xFF27	读写	00000000	端口 0 中断控制 1 位
P1_DR	0xB0	读写	00000000	端口 1 数据寄存器
P1_DM0	0xB1	读写	01000000	端口 1 模式 0 位
P1_DM1	0xB2	读写	01000000	端口 1 模式 1 位
P1_FLAG	0xFF30	读写	00000000	端口 1 中断标志位
P1_GE	0xFF31	读写	01000000	端口 1 数字复用使能寄存器
P1_PU	0xFF33	读写	01000000	端口 1 上拉控制寄存器
P1_PD	0xFF34	读写	00000000	端口 1 下拉控制寄存器
P1_IE	0xFF35	读写	00000000	端口 1 中断使能寄存器
P1_IC0	0xFF36	读写	00000000	端口 1 中断控制 0 位
P1_IC1	0xFF37	读写	00000000	端口 1 中断控制 1 位
P2_DR	0xB8	读写	00000000	端口 2 数据寄存器
P2_DM0	0xB9	读写	00000000	端口 2 模式 0 位
P2_DM1	0xBA	读写	00000000	端口 2 模式 1 位
P2_FLAG	0xFF40	读写	00000000	端口 2 中断标志位
P2_GE	0xFF41	读写	00000000	端口 2 数字复用使能寄存器
P2_PU	0xFF43	读写	00000000	端口 2 上拉控制寄存器
P2_PD	0xFF44	读写	00000000	端口 2 下拉控制寄存器



P2_IE	0xFF45	读写	00000000	端口 2 中断使能寄存器
P2_IC0	0xFF46	读写	00000000	端口 2 中断控制 0 位
P2_IC1	0xFF47	读写	00000000	端口 2 中断控制 1 位

## 4.4.1 PT\_SEL0 (0xFF10)

Bit	7	6	5	4	3	2	1	0
Name	TIM3_CHB_SEL	TIM3_CHA_SEL	ADC_ETR_SEL	I2C_SEL	TIM2_CHA_SEL[1:0]		TIM1_CHB_SEL	TIM1_CHA_SEL
Reset	0	0	0	0	0		0	
Type	R/W	R/W	R/W	R/W	R/W		R/W	

Bit	Name	Function
7	TIM3_CHB_SEL	TIM3_CHB 管脚位置选择寄存器: 0 TIM3_CHB 使用 P0.3 1 TIM3_CHB 使用 P1.5
6	TIM3_CHA_SEL	TIM3_CHA 管脚位置选择寄存器: 0 TIM3_CHA 使用 P0.1 1 TIM3_CHA 使用 P0.4
5	ADC_ETR_SEL	ADC 外部触发采样输入管脚 ADC_ETR 位置选择寄存器 0 ADC_ETR 使用 P0.4 1 ADC_ETR 使用 P1.3
4	I2C_SEL	I2C 管脚位置选择寄存器 0 SCL 使用 P0.2, SDA 使用 P1.6 1 SCL 使用 P1.3, SDA 使用 P1.4
3:2	TIM2_CHA_SEL[1:0]	TIM2_CHA 管脚位置选择寄存器: 00 TIM2_CHA 使用 P1.0 01 TIM2_CHA 使用 P0.2 10 TIM2_CHA 使用 P0.5 11 保留
1	TIM1_CHB_SEL	TIM1_CHB 管脚位置选择寄存器: 0 TIM1_CHB 使用 P1.1 1 TIM1_CHB 使用 P1.4
0	TIM1_CHA_SEL	TIM1_CHA 管脚位置选择寄存器: 0 TIM1_CHA 使用 P1.2 1 TIM1_CHA 使用 P1.7

## 4.4.2 PT\_SEL1 (0xFF11)

Bit	7	6	5	4	3	2	1	0
Name	BKIN_LVL	-						
Reset	0	-						
Type	R/W	-						

Bit	Name	Function
7	BKIN_LVL	BKIN 管脚电平选择寄存器: 0 BKIN 低电平刹车有效 1 BKIN 高电平刹车有效
6:0	N/A	保留位, 读 0



## 4.4.3 PERPO\_EN (0xFF18)

Bit	7	6	5	4	3	2	1	0
Name	PRST_PEN	SPI_PEN	TIM3_CHB_PEN	TIM3_CHA_PEN	TIM2_CHB_PEN	TIM2_CHA_PEN	TIM1_CHB_PEN	TIM1_CHA_PEN
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	PRST_PEN	PRST_PEN 外设管脚位置使能配置寄存器 0 PRST_PEN 管脚位置不使能 1 PRST_PEN 管脚位置使能
6	SPI_PEN	SPI_PEN 外设管脚位置使能配置寄存器 0 SPI_PEN 管脚位置不使能 1 SPI_PEN 管脚位置使能
5	TIM3_CHB_PEN	TIM3_CHB 外设管脚位置使能配置寄存器 0 TIM3_CHB 管脚位置不使能 1 TIM3_CHB 管脚位置使能
4	TIM3_CHA_PEN	TIM3_CHA 外设管脚位置使能配置寄存器 0 TIM3_CHA 管脚位置不使能 1 TIM3_CHA 管脚位置使能
3	TIM2_CHB_PEN	TIM2_CHB 外设管脚位置使能配置寄存器： 0 TIM2_CHB 管脚位置不使能 1 TIM2_CHB 管脚位置使能
2	TIM2_CHA_PEN	TIM2_CHA 外设管脚位置使能配置寄存器： 0 TIM2_CHA 管脚位置不使能 1 TIM2_CHA 管脚位置使能
1	TIM1_CHB_PEN	TIM1_CHB 外设管脚位置使能配置寄存器： 0 TIM1_CHB 管脚位置不使能 1 TIM1_CHB 管脚位置使能
0	TIM1_CHA_PEN	TIM1_CHA 外设管脚位置使能配置寄存器： 0 TIM1_CHA 管脚位置不使能 1 TIM1_CHA 管脚位置使能

## 4.4.4 PERP1\_EN (0xFF19)

Bit	7	6	5	4	3	2	1	0
Name	–	MTP_TEST_PEN	ADC_ETR_PEN	CLK_MTP_PEN	UART1_PEN	I2C_PEN	BRKIN_PEN	UART0_PEN
Reset	–	0	0	0	0	1	0	0
Type	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	N/A	保留位，读 0
6	MTP_TEST_PEN	MTP_TEST_PEN 输出管脚位置配置寄存器： 0 MTP_TEST_PEN 管脚位置输出不使能 1 MTP_TEST_PEN 管脚位置输出使能
5	ADC_ETR_PEN	ADC_ETR_PEN 输出管脚位置配置寄存器：



		0 ADC_ETR_PEN 管脚位置输出不使能 1 ADC_ETR_PEN 管脚位置输出使能
4	CLK_MTP_EN	CLK_MTP_PEN 输出管脚位置配置寄存器： 0 CLK_MTP_PEN 从 P0.0 管脚位置输出不使能 1 CLK_MTP_PEN 从 P0.0 管脚位置输出使能
3	UART1_PEN	UART1_PEN 外设管脚位置配置寄存器： 0 UART1 管脚位置不使能 1 UART1 管脚位置使能
2	I2C_PEN	I2C 外设管脚位置配置寄存器： 0 I2C 管脚位置不使能 1 I2C 管脚位置使能
1	BRKIN_PEN	BRKIN 管脚位置配置寄存器： 0 BRKIN 管脚位置不使能 1 BRKIN 管脚位置使能
0	UART0_PEN	UART0 外设管脚位置配置寄存器： 0 UART0 管脚位置不使能 1 UART0 管脚位置使能

## 4.4.5 PERP2\_EN (0xFF1A)

Bit	7	6	5	4	3	2	1	0
Name				RZ_MODE_PEN	ILO_PEN	OSCIN_PEN	CMP1_PEN	CMPO_PEN
Reset				0	0	0	0	0
Type				R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:5	N/A	保留位，读 0
4	RZ_MODE_PEN	RZ 码输出管脚位置配置寄存器 0 RZ 码管脚位置输出不使能 1 RZ 码管脚位置输出使能
3	ILO_PEN	32K ILO 输出管脚位置配置寄存器 0 ILO 管脚位置输出不使能 1 ILO 管脚位置输出使能
2	OSCIN_PEN	OSCIN 输管脚位置配置寄存器 0 OSCIN 管脚位置输入不使能 1 OSCIN 管脚位置输入使能
1	CMP1_PEN	CMP1 比较结果输出管脚位置配置寄存器 0 CMP1 管脚位置不使能 1 CMP1 管脚位置使能
0	CMPO_PEN	CMPO 比较结果输出管脚位置配置寄存器 0 CMPO 管脚位置不使能 1 CMPO 管脚位置使能

## 4.4.6 PO\_DR (0x98)

Bit	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---



Name	PO_DR
Reset	0x00
Type	R/W

Bit	Name	Function
7:0	PO_DR	端口 0 的数据寄存器，写该寄存器会更新端口输出，读该寄存器得到端口输出值，读该寄存器详细说明见 4.2 中表格 xx。

## 4.4.7 PO\_GE (0xFF21)

Bit	7	6	5	4	3	2	1	0
Name	GEO.7	GEO.6	GEO.5	GEO.4	GEO.3	GEO.2	GEO.1	GEO.0
Reset	0	0	0	0	0	1	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	GEO.7	端口 0 的外设复用功能使能： 0 关闭复用使能，输出由 Px.DR 决定，如果输入则输入电平寄存在 Px.DR 中 1 打开复用使能，输出由 GDO 决定，如果输入使能则输入电平到 GDI
6	GEO.6	
5	GEO.5	
4	GEO.4	
3	GEO.3	
2	GEO.2	
1	GEO.1	
0	GEO.0	

## 4.4.8 PO\_DMO (0x99)

Bit	7	6	5	4	3	2	1	0
Name	PO_DMO							
Reset	0x04							
Type	R/W							

Bit	Name	Function
7:0	PO_DMO	PO 模式控制寄存器。

## 4.4.9 PO\_DM1 (0x9A)

Bit	7	6	5	4	3	2	1	0
Name	PO_DM1							
Reset	0x04							
Type	R/W							

Bit	Name	Function
7:0	PO_DM1	PO 模式控制寄存器。

## 4.4.10 PO\_PU (0xFF23)

Bit	7	6	5	4	3	2	1	0
Name	PO_PU							
Reset	0x04							
Type	R/W							



Bit	Name	Function
7:0	PO_PU	上拉使能： 0 关闭上拉 1 打开上拉

## 4.4.11 PO\_PD (0xFF24)

Bit	7	6	5	4	3	2	1	0
Name	PO_PD							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	PO_PD	下拉使能： 0 关闭下拉 1 打开下拉

## 4.4.12 PO\_IE (0xFF25)

Bit	7	6	5	4	3	2	1	0
Name	PO_IE							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	PO_IE	中断使能： 0 关闭端口中断 1 打开端口中断

## 4.4.13 PO\_IC0/PO\_IC1 (0xFF26/0xFF27)

Bit	7	6	5	4	3	2	1	0
Name	PO_IC0							
Reset	0xFF							
Type	R/W							

Bit	7	6	5	4	3	2	1	0
Name	PO_IC1							
Reset	0xFF							
Type	R/W							

Bit	Name	Function
7:0	PO_IC0	[PO_IC1: PO_IC0] 中断触发模式控制： 00 上升沿中断 01 下降沿中断 10 高电平中断 11 低电平中断
7:0	PO_IC1	



## 4.4.14 P0\_FLAG (0xFF20)

Bit	7	6	5	4	3	2	1	0
Name	P0_FLAG							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P0_FLAG	中断标志： 0 没有中断发生 1 有中断发生 写 1 清除该中断标志

## 4.4.15 P1\_DR (0xB0)

Bit	7	6	5	4	3	2	1	0
Name	P1_DR							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P1_DR	端口 1 的数据寄存器，写该寄存器会更新端口输出，读该寄存器得到端口输出值，读该寄存器详细说明见 4.2 中表格 xx。

## 4.4.16 P1\_GE (0xFF31)

Bit	7	6	5	4	3	2	1	0
Name	GE1.7	GE1.6	GE1.5	GE1.4	GE1.3	GE1.2	GE1.1	GE1.0
Reset	0	1	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	GE1[7:0]	端口 1 的外设复用功能使能： 0 关闭复用使能，输出由 Px.DR 决定，如果输入则输入电平寄存在 Px.DR 中 1 打开复用使能，输出由 G0 决定，如果输入使能则输入电平到 GI

## 4.4.17 P1\_DMO (0xB1)

Bit	7	6	5	4	3	2	1	0
Name	P1_DMO							
Reset	0x40							
Type	R/W							

Bit	Name	Function
7:0	P1_DMO	P1 模式控制寄存器。

## 4.4.18 P1\_DM1 (0xB2)

Bit	7	6	5	4	3	2	1	0
Name	P1_DM1							
Reset	0x40							





Type	R/W							
------	-----	--	--	--	--	--	--	--

Bit	Name	Function
7:0	P1_DM1	P1 模式控制寄存器。

## 4.4.19 P1\_PU (0xFF33)

Bit	7	6	5	4	3	2	1	0
Name	P1_PU							
Reset	0x40							
Type	R/W							

Bit	Name	Function
7:0	P1_PU	上拉使能： 0 关闭上拉 1 打开上拉

## 4.4.20 P1\_PD (0xFF34)

Bit	7	6	5	4	3	2	1	0
Name	P1_PD							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P1_PD	下拉使能： 0 关闭下拉 1 打开下拉

## 4.4.21 P1\_IE (0xFF35)

Bit	7	6	5	4	3	2	1	0
Name	P1_IE							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P1_IE	中断使能： 0 关闭端口中断 1 打开端口中断

## 4.4.22 P1\_IC0/P1\_IC1 (0xFF36/0xFF37)

Bit	7	6	5	4	3	2	1	0
Name	P1_IC0							
Reset	0xFF							
Type	R/W							

Bit	7	6	5	4	3	2	1	0
Name	P1_IC1							
Reset	0x00							
Type	R/W							



Bit	Name	Function
7:0	P1_IC0	[P1_IC1: P1_IC0] 中断触发模式控制: 00 上升沿中断
		01 下降沿中断
7:0	P1_IC1	10 高电平中断
		11 低电平中断

## 4.4.23 P1\_FLAG (0xFF30)

Bit	7	6	5	4	3	2	1	0
Name	P1_FLAG							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P1_FLAG	中断标志: 0 没有中断发生 1 有中断发生 写 1 清除该中断标志

## 4.4.24 P2\_DR (0xB8)

Bit	7	6	5	4	3	2	1	0
Name	P2_DR							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P2_DR	端口 2 的数据寄存器，写该寄存器会更新端口输出，读该寄存器得到端口输出值，读该寄存器详细说明见 4.2 中表格 xx。

## 4.4.25 P2\_GE (0xFF41)

Bit	7	6	5	4	3	2	1	0
Name	-						GE2.1	GE2.0
Reset	-						0	0
Type	-						R/W	R/W

Bit	Name	Function
7:2	N/A	保留位，读 0
1:0	GE2[1:0]	端口 2 的外设复用功能使能: 0 关闭复用使能，输出由 Px.DR 决定，如果输入则输入电平寄存在 Px.DR 中
		1 打开复用使能，输出由 GDO 决定，如果输入使能则输入电平到 GDI

## 4.4.26 P2\_DM0 (0xB9)

Bit	7	6	5	4	3	2	1	0
Name	P2_DM0							
Reset	0x00							
Type	R/W							



Bit	Name	Function
7:0	P2_DM0	P2 模式控制寄存器。

## 4.4.27 P2\_DM1 (0xBA)

Bit	7	6	5	4	3	2	1	0
Name	P2_DM1							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P2_DM1	P2 模式控制寄存器。

## 4.4.28 P2\_PU (0xFF43)

Bit	7	6	5	4	3	2	1	0
Name	P2_PU							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P2_PU	上拉使能： 0 关闭上拉 1 打开上拉

## 4.4.29 P2\_PD (0xFF44)

Bit	7	6	5	4	3	2	1	0
Name	P2_PD							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P2_PD	下拉使能： 0 关闭下拉 1 打开下拉

## 4.4.30 P2\_IE (0xFF45)

Bit	7	6	5	4	3	2	1	0
Name	P2_IE							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P2_IE	中断使能： 0 关闭端口中断 1 打开端口中断



## 4.4.31 P2\_IC0/P2\_IC1 (0xFF46/0xFF47)

Bit	7	6	5	4	3	2	1	0
Name	P2_IC0							
Reset	0x03							
Type	R/W							

Bit	7	6	5	4	3	2	1	0
Name	P2_IC1							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P2_IC0	[P2_IC1: P2_IC0] 中断触发模式控制:
		00 上升沿中断
7:0	P2_IC1	01 下降沿中断
		10 高电平中断
		11 低电平中断

## 4.4.32 P2\_FLAG (0xFF40)

Bit	7	6	5	4	3	2	1	0
Name	P2_FLAG							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P2_FLAG	中断标志: 0 没有中断发生 1 有中断发生 写 1 清除该中断标志。

## 5. CPU

本芯片全兼容传统的 8051 微控制器，所有指令的助记符和二进制码都和 8051 兼容。

### 5.1 CPU 内核概述

处理器采用了一些体系结构上的优化，相比传统的 8051 在性能上面有了很大的提升。内部的 ALU 配合内部的 ACC (0xE0), B (0xF0), PSW (0xD0) 寄存器可以实现各种 8 位运算操作。

ALU 可以进行典型操作如下：

- 基本算术运算：加法、减法、乘法、除法
- 其他算术运算：自加、自减、BCD 调整、比较
- 逻辑运算：与、或、异或、取反、移位
- 布尔比特运算：置位、清零、取反、按位判断跳转、进位操作

还有一些 8051 核内部使用的寄存器可以通过 SFR 地址访问，包括 SP、DPL0/1、DPH0/1、DPS 等。具体地址分配见 5.2 中描述。

### 5.2 CPU 内核 SFR 寄存器

名字	地址	读写	复位值	描述
ACC	0xE0	读写	00000000	累加寄存器
B	0xF0	读写	00000000	B 寄存器
PSW	0xD0	读写	00000000	程序状态字寄存器
P2	0xA0	读写	00000000	P2 读写寄存器
IE	0xA8	读写	00000000	系统中断使能寄存器
SP	0x81	读写	00000111	堆栈指针，指向 IDATA 空间
DPL0	0x82	读写	00000000	DPTR0 寄存器的低 8 位
DPH0	0x83	读写	00000000	DPTR0 寄存器的高 8 位
DPL1	0x84	读写	00000000	DPTR1 寄存器的低 8 位
DPH1	0x85	读写	00000000	DPTR1 寄存器的高 8 位
DPS	0x86	读写	00000000	DPTR0/DPTR1 选择寄存器

#### 5.2.1 ACC 寄存器 (0xE0)

Bit	7	6	5	4	3	2	1	0
Name	ACC. 7	ACC. 6	ACC. 5	ACC. 4	ACC. 3	ACC. 2	ACC. 1	ACC. 0
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	ACC	累加寄存器。



## 5.2.2 B 寄存器 (0xF0)

Bit	7	6	5	4	3	2	1	0
Name	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	B	乘法运算和除法运算的时候使用，其他情况用作普通寄存器。

## 5.2.3 PSW 寄存器 (0xD0)

Bit	7	6	5	4	3	2	1	0
Name	CY	AC	F0	RS[1:0]		OV	F1	P
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	CY	进位标志
6	AC	辅助进位标志
5	F0	通用标志 0
4:3	RS[1:0]	寄存器组选择： 00 寄存器组 0，数据地址 0x00-0x07 01 寄存器组 1，数据地址 0x08-0x0F 10 寄存器组 2，数据地址 0x10-0x17 11 寄存器组 3，数据地址 0x18-0x1F
2	OV	溢出标志
1	F1	通用标志 1
0	P	奇偶校验标志

## 5.2.4 P2 寄存器 (0xA0)

Bit	7	6	5	4	3	2	1	0
Name	P2							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P2	使用 MOVX 指令使用 R0 或者 R1 的时候访问 XRAM 空间的时候标志地址的[15:8]位。

## 5.2.5 IE 寄存器 (0xA8)

Bit	7	6	5	4	3	2	1	0
Name	IE_EA	-						
Reset	0	-						
Type	R/W	-						

Bit	Name	Function
7	IE_EA	CPU 中断允许位总开关，1 表示使能中断，0 表示不使能中断。



6:0	N/A	保留位，读 0
-----	-----	---------

## 5.2.6 SP 寄存器 (0x81)

Bit	7	6	5	4	3	2	1	0
Name	SP							
Reset	0x07							
Type	R/W							

Bit	Name	Function
7:0	SP	堆栈指针，指向 IDATA 区域。

## 5.2.7 DPL0 寄存器 (0x82)

Bit	7	6	5	4	3	2	1	0
Name	DPTR0[7:0]							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DPTR0[7:0]	用于 DPTR0[7:0]。

## 5.2.8 DPH0 寄存器 (0x83)

Bit	7	6	5	4	3	2	1	0
Name	DPTR0[15:8]							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DPTR0[15:8]	用于 DPTR0[15:8]。

## 5.2.9 DPL1 寄存器 (0x84)

Bit	7	6	5	4	3	2	1	0
Name	DPTR1[7:0]							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DPTR1[7:0]	用于 DPTR1[7:0]。

## 5.2.10 DPH1 寄存器 (0x85)

Bit	7	6	5	4	3	2	1	0
Name	DPTR1[15:8]							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DPTR1[15:8]	用于 DPTR1[15:8]。



5.2.11 DPS 寄存器 (0x86)

Bit	7	6	5	4	3	2	1	0
Name	-							SEL
Reset	-							0
Type	-							R/W

Bit	Name	Function
7:1	N/A	保留位，读 0
0	SEL	SEL=0 时系统使用 DPTR0 寄存器； SEL=1 时系统使用 DPTR1 寄存器。



## 6. 存储器

本芯片内部有 3 种存储器：SFR，内部数据存储器，程序存储器。

程序存储器只能读不能写，该存储器大小为 16K 字节。内部数据存储器大小为 1K 字节。SFR 为内部特殊功能寄存器。

### 6.1 程序存储器

本芯片的程序指针为 16 位，最大寻址空间可达 64K 字节，实际只实现了 16K 字节的程序存储空间。

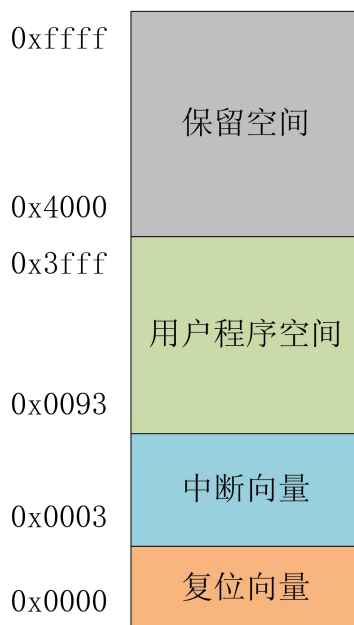


图 7 程序存储空间

复位后，MCU 从 0000H 开始执行。从 0003H 开始是中断向量表，当发生中断且中断使能后，PC 会跳转到对应的中断向量位置去执行。

## 6.2 数据存储

数据存储分内部数据存储器和外部数据存储器，内部数据存储器空间为 256 字节，其中低 128 字节可以直接访问（通过地址 0x00~0x7f），高 128 字节和 SFR 共用一个地址空间（通过地址 0x80~0xff），直接寻址方式可以访问到 SFR 空间，通过间接寻址方式可以访问内部数据存储器的高 128 字节。低 128 字节数据存储空间可以划分为如下图所示的不同空间。



图 8 数据存储器

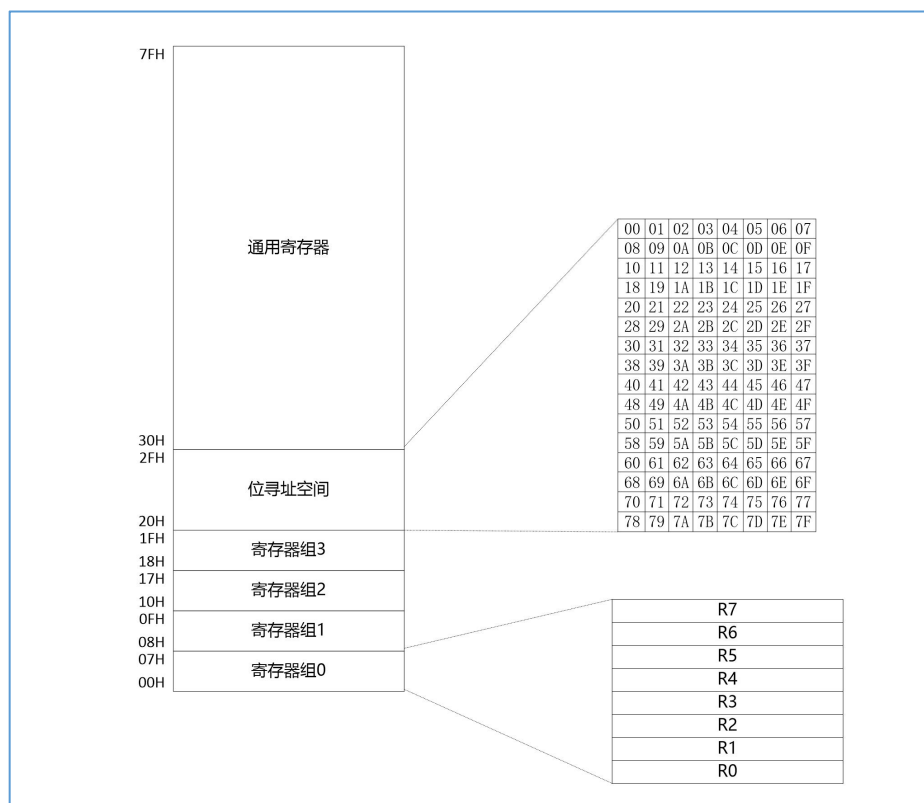


图 9 内部低 128 字节数据空间分配

此外，片上还集成了 768 字节的片上 RAM 映射在外部数据存储空间，地址范围为 0x0000~0x02ff。



## 6.3 SFR 空间

表 5.1 SFR 空间地址表

地址	寄存器	地址	寄存器	地址	寄存器	地址	寄存器
0x80		0x90		0xa0	P2	0xb0	P1_DR
0x81	SP	0x91	SCR_CFG	0xa1	I2C_ADDR	0xb1	P1_DMO
0x82	DPL0	0x92	SCR_SLEEP	0xa2	I2C_CR	0xb2	P1_DM1
0x83	DPH0	0x93	SCR_CALI	0xa3	I2C_STAT	0xb3	
0x84	DPL1	0x94	CLK_CR	0xa4	I2C_DR	0xb4	
0x85	DPH1	0x95	PCLK_CR	0xa5	I2C_MCR	0xb5	
0x86	DPS	0x96	PCLK_DIV12	0xa6		0xb6	
0x87		0x97	PCLK_DIV3	0xa7		0xb7	
0x88	SLPTIM_CR	0x98	PO_DR	0xa8	IE	0xb8	P2_DR
0x89	SLPTIM_SR	0x99	PO_DMO	0xa9		0xb9	P2_DMO
0x8a	SLPTIM_CLR	0x9a	PO_DM1	0xaa	INT_MSK0	0xba	P2_DM1
0x8b	SLPTIM_WDT	0x9b		0xab	INT_MSK1	0xbb	
0x8c	SLPTIM_CNTL	0x9c	UART0_DR	0xac	INT_MSK2	0xbc	UART1_DR
0x8d	SLPTIM_CNTH	0x9d	UART0_CR	0xad	INT_FWSET0	0xbd	UART1_CR
0x8e	SLPTIM_PRDRL	0x9e	UART0_STAT	0xae	INT_FWSET1	0xbe	UART1_SR
0x8f	SLPTIM_PRDRH	0x9f	UART0_CLK	0xaf	INT_FWSET2	0xbf	UART1_CFG
0xc0	TIM1_CR	0xd0	PSW	0xe0	ACC	0xf0	B
0xc1	TIM1_IE	0xd1		0xe1	ACO_CR1	0xf1	SPI_ICR
0xc2	TIM1_SR	0xd2		0xe2	ACO_CR2	0xf2	SPI_DR
0xc3	TIM1_PR	0xd3		0xe3		0xf3	SPI_CR
0xc4		0xd4		0xe4	AC1_CR1	0xf4	SPI_STAT
0xc5		0xd5		0xe5	AC1_CR2	0xf5	SPI_CR2
0xc6		0xd6		0xe6		0xf6	SPI_WRADDR
0xc7		0xd7		0xe7		0xf7	
0xc8	TIM2_CR	0xd8		0xe8	ADC_CR0	0xf8	TIMO_CR
0xc9	TIM2_IE	0xd9		0xe9	ADC_CR1	0xf9	TIMO_CNTR
0xca	TIM2_SR	0xda		0xea	ADC_CR2	0xfa	TIMO_ARR
0xcb	TIM2_PR	0xdb		0xeb	ADC_CHSEL	0xfb	TIMO_IE
0xcc	TIM3_CR	0xdc		0xec	ADC_CON	0xfc	TIMO_SR
0xcd	TIM3_IE	0xdd		0xed	ADC_DLY	0xfd	SSCONR
0xce	TIM3_SR	0xde		0xee	ADC_RESL	0xfe	ADC_COMPL
0xcf	TIM3_PR	0xdf		0xef	ADC_RESB	0xff	ADC_COMPH

## 6.4 XDATA 空间

除 768 字节片上 RAM 外，芯片中一部分寄存器放在外部数据存储器 XDATA 空间，该部分地址空间大小 256 字节，地址范围 0xFF00~0xFFFF。下面表所示：

表 5.2 寄存器 XDATA 空间地址表



	0H/8H	1H/9H	2H/AH	3H/BH	4H/CH	5H/DH	6H/EH	7H/FH
FF98					BUF_OSN	BUF_OSP	DAC_BUFPCR2	
FF90								DAC_BUFPCR1
FF88	IMO_CR	IMO_TRIM	ILO_TRIM	ILO_TEST	IMO_TRIMH		MBIST_KEY	
FF80	BG_CR	BG_VTRIM	BG_ITRIM	BG_TCTRIM	BG_TEST	BORLVD_CR	BORLVD_STA	ANA_TEST
FF78	TIM3_CNTL	TIM3_CNTH	TIM3_ARR	TIM3_ARRH	TIM3_GCMAR	TIM3_GCMAR	TIM3_GCMBR	TIM3_GCMBR
FF70	TIM3_FCON	TIM3_VPER	TIM3_DTU	TIM3_BRAK	TIM3_DTR	TIM3_PCONR	TIM3_PCONR	
FF68	TIM2_CNTL	TIM2_CNTH	TIM2_ARR	TIM2_ARRH	TIM2_GCMAR	TIM2_GCMAR	TIM2_GCMBR	TIM2_GCMBR
FF60	TIM2_FCON	TIM2_VPER	TIM2_DTU	TIM2_BRAK	TIM2_DTR	TIM2_PCONR	TIM2_PCONR	
FF58	TIM1_CNTL	TIM1_CNTH	TIM1_ARR	TIM1_ARRH	TIM1_GCMAR	TIM1_GCMAR	TIM1_GCMBR	TIM1_GCMBR
FF50	TIM1_FCON	TIM1_VPER	TIM1_DTU	TIM1_BRAK	TIM1_DTR	TIM1_PCONR	TIM1_PCONR	
FF48								
FF40	P2_FLAG	P2_GE		P2_PU	P2_PD	P2_IE	P2_IC0	P2_IC1
FF38								
FF30	P1_FLAG	P1_GE		P1_PU	P1_PD	P1_IE	P1_IC0	P1_IC1
FF28								
FF20	PO_FLAG	PO_GE		PO_PU	PO_PD	PO_IE	PO_IC0	PO_IC1
FF18	PERPO_EN	PERP1_EN	PERP2_EN					
FF10	PT_SELO	PT_SEL1						
FF08								
FF00	FLASH_CR	FLASH_CFG	FLASH_KEY	FLASH_ADL	FLASH_ADH	FLASH_PBUF	FLASH_PBUF	FLASH_DR

## 6.5 FLASH 控制器

本芯片内部实现了一个大小为 16KB 的 FLASH 存储器，编程次数可达 1000 次。其中包含了 16K 字节 (8K\*16bit) 的主存储区域，用来存放程序；64 字节 (32\*16bit) 的 Information 区域，用来存放 FT 测试时的修调数据；128 字节 (128\*8bit) 的 EEPROM 区域，用于用户存储数据。FLASH 控制器用来控制 8051 访问的 FLASH 存储器的读时序和编程器通过编程接口编程 FLASH 存储器。

### 6.5.1 与 FLASH 控制器相关寄存器定义

名字	地址	读写	复位值	描述
FLASH_CR	0xFF00	读写	00000000	FLASH 控制寄存器
FLASH_CFG	0xFF01	读写	00000011	FLASH 配置寄存器
FLASH_KEY	0xFF02	读写	00000000	FLASH key 寄存器
FLASH_ADL	0xFF03	读写	00000000	FLASH 编程地址低 8 位
FLASH_ADH	0xFF04	读写	00000000	FLASH 编程地址高 6 位
FLASH_PBUFL	0xFF05	读写	00000000	FLASH 编程缓冲地址低 8 位
FLASH_PBUFH	0xFF06	读写	00000000	FLASH 编程缓冲地址高 2 位
FLASH_DR	0xFF07	只读	xxxxxxxx	FLASH 读数据寄存器



## 6.5.1.1 FLASH\_CR (0xFF00)

Bit	7	6	5	4	3	2	1	0
Name	ISAVB	STATICEN	WRSZ[1:0]		CKEN	FWEEP	IFREN	BUSY
Reset	1	1	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	ISAVB	Option setting for Read operation : 0 ISAVB = L to select the low power or power saving mode. 1 ISAVB = H to select the high speed mode. 注意：用于 MTP 测试，正常工作时该位不能写 0
6	STATICEN	0 STATICEN = L, NON-STATIC mode (INDS <10uA) 1 STATICEN = H, STATIC mode (IDS <500uA). 注意：用于 MTP 测试，正常工作时该位不能写 0
5:4	WRSZ[1:0]	FLASH 存储器编程数据大小，单位为半字（2 个字节） 00: 2 01: 32 10: 64 11: 128
3	CKEN	FLASH 时钟使能： 0 关闭时钟 1 使能时钟
2	FWEEP	0 不选择 EEPROM 区域 1 选择 EEPROM 区域，备注：FWEEP 和 IFREN 不能同时为 1
1	IFREN	0 不选择 FLASH 用户区域 1 选择 FLASH 信息区域，备注：FWEEP 和 IFREN 不能同时为 1
0	BUSY	读模式下 BUSY 的值表示： 0 FLASH 编程完成 1 FLASH 编程没有完成 写 1 开始编程操作。

## 6.5.1.2 FLASH\_CFG (0xFF01)

Bit	7	6	5	4	3	2	1	0
Name	FWSEL	CLEAN	SRL	MRGN	SAVPWR1	SAVPWRO	RDCYC[1:0]	
Reset	0	0	0	0	0	0	1	1
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	FWSEL	FLASH 控制信号选择： 0 使用默认的 FLASH CLEN, ISAVB, STATICEN 信号 1 使用寄存器定义的 FLASH CLEN, ISAVB, STATICEN 信号
6	CLEAN	FLASH 测试模式
5	SRL	FLASH SRL 控制
4	MRGN	FLASH MRGN 控制
3	SAVPWR1	SLEEP 模式门控 CS 信号： 0 SLEEP 模式时 CS 信号门控打开



		1 SLEEP 模式时 CS 信号门控关闭 注意: SLEEP 模式时建议此位置 1, 休眠功耗较未置 1 会更小。
2	SAVPWRO	SLEEP 模式门控 READ 信号: 0 SLEEP 模式时 READ 信号门控打开 1 SLEEP 模式时 READ 信号门控关闭 注意: SLEEP 模式时建议此位置 1, 休眠功耗较未置 1 会更小。
1:0	RDCYC[1:0]	FLASH 访问周期: 00 1 个周期 01 2 个周期 10 5 个周期 11 6 个周期 注意: 当芯片电压低于 4.5V 时候, 要配置 RDCYC 为 11 (6 个周期)。芯片 FLASH 内部实现了 2 个字节的缓冲, VDD 电压高于 4.5V 时使用 01 (2 个周期) 配置即可, 这样可以保证性能和功耗的平衡。 读 EEPROM 时, 所有电压下都需要配置 RDCYC 为 11 (6 个周期)。

## 6.5.1.3 FLASH\_KEY (0xFF02)

Bit	7	6	5	4	3	2	1	0
Name	FLASH_KEY[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	FLASH_KEY[7:0]	FLASH 烧写密钥, 写 0xCA 打开, 打开后才能向 FLASH_CR 的 bit 0 写 1 来启动烧写 FLASH。

## 6.5.1.4 FLASH\_ADL (0xFF03)

Bit	7	6	5	4	3	2	1	0
Name	ADL[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	ADL[7:0]	FLASH 访问地址低 8 位。

## 6.5.1.5 FLASH\_ADH (0xFF04)

Bit	7	6	5	4	3	2	1	0
Name	-		ADH					
Reset	-		0	0	0	0	0	0
Type	-		R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	N/A	保留位, 读 0
5:0	ADH	FLASH 访问地址高 6 位。

## 6.5.1.6 FLASH\_PBUFL (0xFF05)

Bit	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---



Name	PBUFL							
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	PBUFL	FLASH 编程缓冲地址低 8 位，在 MTP MBIST 过程中作为读和写的的数据，高 8bit 和低 8bit 都为 PBUFL

## 6.5.1.7 FLASH\_PBUFH (0xFF06)

Bit	7	6	5	4	3	2	1	0
Name	-						PBUFH	
Reset	-						0	0
Type	-						R/W	R/W

Bit	Name	Function
7:0	PBUFH	FLASH 编程缓冲地址高 2 位

## 6.5.1.8 FLASH\_DR (0xFF07)

Bit	7	6	5	4	3	2	1	0
Name	DR							
Reset	X	X	X	X	X	X	X	X
Type	R	R	R	R	R	R	R	R

Bit	Name	Function
7:0	DR	FLASH 读数据。



## 7. 中断控制器

### 7.1 概述

本芯片支持多达 17 个中断源。每个中断源都有独立的中断使能信号，可以根据需要配置两级优先级。中断控制器有以下特性：

- 从 17 个中断源接收中断
- 每个中断有固定的中断号，中断号越小优先级越高，同时可根据需要配置寄存器提高中断号大的中断源的优先级
- 中断延时：5~8 机器周期

### 7.2 GPIO 中断

GPIO 中断来自引脚，可以根据寄存器配置来选择中断发生的条件。GPIO 中断可以通过 Px\_IC0/1 来选择中断触发条件。寄存器 Px\_FLAG 保存每个中断的中断标志，写 1 清除该位。

### 7.3 中断向量表

中断控制器支持 17 个中断源。当中断发生且中断使能之后，跳转到对应向量地址去执行 LCALL 指令来进入中断服务程序。

表 1 中断向量表

中断源	中断等级	中断号	中断地址	说明
LVD	低	0	0003H	低压检测中断
P0	低	1	000BH	GPIO0 脚中断
P1	低	2	0013H	GPIO1 脚中断
P2	低	3	001BH	GPIO2 脚中断
保留	-	4	-	保留
SCK3	低	5	002BH	SCK3 时钟有效中断
Timer0	低	6	0033H	定时器 0 中断
Timer1	低	7	003BH	定时器 1 中断
Timer2	低	8	0043H	定时器 2 中断
Timer3	低	9	004BH	定时器 3 中断
ADC	低	10	0053H	ADC 转换完成中断
CMP0	低	11	005BH	模拟比较器 0 中断
CMP1	低	12	0063H	模拟比较器 1 中断
I2C	低	13	006BH	I2C 状态中断
UART0	低	14	0073H	UART0 状态中断
UART1	低	15	007BH	UART1 状态中断
SPI	低	16	0083H	SPI 中断
WDT	低	17	008BH	看门狗中断





## 7.4 中断优先级和中断屏蔽

每个中断有一个唯一的中断号。中断号越小，中断的优先级更高。同时，每个中断源都有一个优先级配置位，用户可以根据需要配置该位以提高对应中断的优先级。每个中断有一个中断屏蔽位，用户通过设置中断屏蔽位可以屏蔽对应的中断。

## 7.5 中断触发方式

在每个功能模块中，中断置位和清零是沿敏感；在中断控制器中，中断是电平敏感。

## 7.6 与中断相关寄存器定义

在每个功能模块中，中断置位和清零是沿敏感；在中断控制器中，中断是电平敏感。

名字	地址	读写	复位值	描述
INT_MSK0	0xAA	读写	00000000	中断屏蔽寄存器 0
INT_MSK1	0xAB	读写	00000000	中断屏蔽寄存器 1
INT_MSK2	0xAC	读写	00000000	中断屏蔽寄存器 2
INT_PRI0	0xAD	读写	00000000	软件触发中断寄存器 0
INT_PRI1	0xAE	读写	00000000	软件触发中断寄存器 1
INT_PRI2	0xAF	读写	00000000	软件触发中断寄存器 2

### 7.6.1 INT\_MSK0 (0xAA)

Bit	7	6	5	4	3	2	1	0
Name	T1MSK	T0MSK	SCK3MSK	-	P2MSK	P1MSK	P0MSK	LVDMSK
Reset	0	0	0	-	0	0	0	0
Type	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W

Bit	Name	Function
7	T1MSK	0 = 不屏蔽 Timer1 中断 1 = 屏蔽 Timer1 中断
6	T0MSK	0 = 不屏蔽 Timer0 中断 1 = 屏蔽 Timer0 中断
5	SCK3MSK	0 = 不屏蔽 SCK3 中断 1 = 屏蔽 SCK3 中断
4	N/A	保留位，读 0
3	P2MSK	0 = 不屏蔽 GPIO 2 中断 1 = 屏蔽 GPIO 2 中断
2	P1MSK	0 = 不屏蔽 GPIO 1 中断 1 = 屏蔽 GPIO 1 中断
1	P0MSK	0 = 不屏蔽 GPIO 0 中断 1 = 屏蔽 GPIO 0 中断
0	LVDMSK	0 = 不屏蔽 LVD 中断 1 = 屏蔽 LVD 中断



## 7.6.2 INT\_MSK1 (0xAB)

Bit	7	6	5	4	3	2	1	0
Name	UART1MSK	UART0MSK	I2CMSK	CMP1MSK	CMP0MSK	ADCMSK	T3MSK	T2MSK
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	UART1MSK	0 = 不屏蔽 UART1 中断 1 = 屏蔽 UART1 中断
6	UART0MSK	0 = 不屏蔽 UART0 中断 1 = 屏蔽 UART0 中断
5	I2CMSK	0 = 不屏蔽 I2C 中断 1 = 屏蔽 I2C 中断
4	CMP1MSK	0 = 不屏蔽比较器 1 中断 1 = 屏蔽比较器 1 中断
3	CMP0MSK	0 = 不屏蔽比较器 0 中断 1 = 屏蔽比较器 0 中断
2	ADCMSK	0 = 不屏蔽 ADC 中断 1 = 屏蔽 ADC 中断
1	T3MSK	0 = 不屏蔽 Timer3 中断 1 = 屏蔽 Timer3 中断
0	T2MSK	0 = 不屏蔽 Timer2 中断 1 = 屏蔽 Timer2 中断

## 7.6.3 INT\_MSK2 (0xAC)

Bit	7	6	5	4	3	2	1	0
Name	-						WDTMSK	SPIMSK
Reset	-						0	0
Type	-						R/W	R/W

Bit	Name	Function
7: 2	N/A	保留位, 读 0
1	WDTMSK	0 = 不屏蔽 WDT 中断 1 = 屏蔽 WDT 中断
0	SPIMSK	0 = 不屏蔽 SPI 中断 1 = 屏蔽 SPI 中断

## 7.6.4 INT\_PRI0 (0xAD)

Bit	7	6	5	4	3	2	1	0
Name	T1PRI	TOPRI	SCK3PRI	-	P2PRI	P1PRI	POPRI	LVDPRI
Reset	0	0	0	-	0	0	0	0
Type	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W

Bit	Name	Function
7	T1PRI	0 Timer1 中断为低优先级 1 Timer1 中断为高优先级



6	TOPRI	0	Timer0 中断为低优先级
		1	Timer0 中断为高优先级
5	SCK3PRI	0	SCK3 中断为低优先级
		1	SCK3 中断为高优先级
4	N/A	保留位, 读 0	
3	P2TPRI	0	GPIO 2 中断为低优先级
		1	GPIO 2 中断为高优先级
2	P1PRI	0	GPIO 1 中断为低优先级
		1	GPIO 1 中断为高优先级
1	POPRI	0	GPIO 0 中断为低优先级
		1	GPIO 0 中断为高优先级
0	LVDPRI	0	LVD 中断为低优先级
		1	LVD 中断为高优先级

## 7.6.5 INT\_PRI1 (0xAE)

Bit	7	6	5	4	3	2	1	0
Name	UART1PRI	UARTOPRI	I2CPRI	CMP1PRI	CMPOPRI	ADCPRI	T3PRI	T2PRI
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function	
7	UART1PRI	0	不触发 UART0 中断
		1	触发 UART0 中断
6	UARTOPRI	0	不触发 UART0 中断
		1	触发 UART0 中断
5	I2CPRI	0	I2C 中断为低优先级
		1	I2C 中断为高优先级
4	CMP1PRI	0	比较器 1 中断为低优先级
		1	比较器 1 中断为高优先级
3	CMPOPRI	0	比较器 0 中断为低优先级
		1	比较器 0 中断为高优先级
2	ADCPRI	0	ADC 中断为低优先级
		1	ADC 中断为高优先级
1	T3PRI	0	Timer3 中断为低优先级
		1	Timer3 中断为高优先级
0	T2PRI	0	Timer2 中断为低优先级
		1	Timer2 中断为高优先级

## 7.6.6 INT\_PRI2 (0xAF)

Bit	7	6	5	4	3	2	1	0
Name	-						WDTPRI	SPIPRI
Reset	-						0	0
Type	-						R/W	R/W



Bit	Name	Function
7:2	N/A	保留位，读 0
1	WDTPRI	0 WDT 中断为低优先级
		1 WDT 中断为高优先级
0	SPIPRI	0 SPI 中断为低优先级
		1 SPI 中断为高优先级

## 8. 时钟

### 8.1 概述

系统有三个时钟源，来自内部的 16MHz 高速 RC 振荡器、来自外部的 16MHz 时钟和内部 32KHz 低速 RC 振荡器。

### 8.2 时钟结构框图

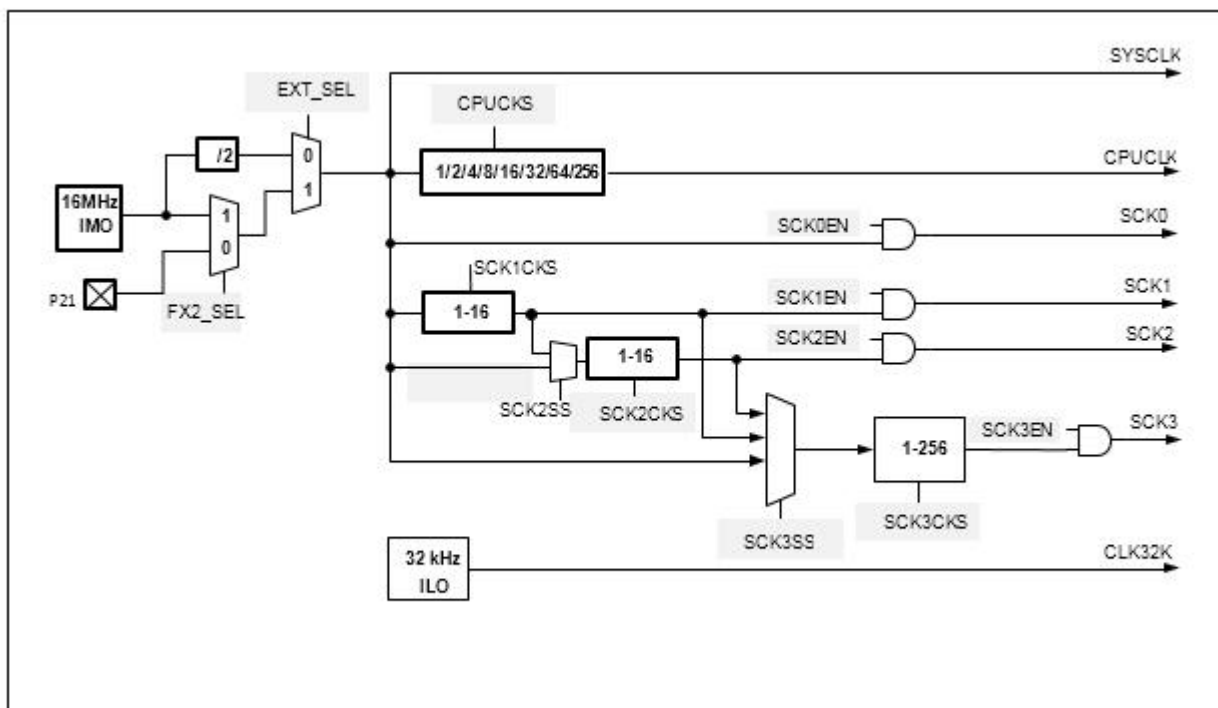


图 10 时钟结构框图

外设	总线时钟	工作时钟
CPU	HCLK_CORE	HCLK_CORE
RAM	HCLK_SRAM	HCLK_SRAM
睡眠定时器	HCLK_MEM	CLK_32K
看门狗	HCLK_MEM	CLK_32K
定时器 0~3	HCLK_MEM	SCK0/CLK_32K
ADC	SCK0	SCK0
SPI	SCK0	SCK0/SCK1/SCK2/SCK3
比较器 0/1	HCLK_CORE	SCK0
UART	HCLK_CORE	SCK1/SCK2/SCK3
I2C	HCLK_CORE	SCK1/SCK2
GPIO	HCLK_MEM	DPx_DI 采样和中断检测 FCLK ADC_ETR 和 BRKIN 异步打拍 FCLK
其他外设		



ANA_CTRL	HCLK_MEM	
----------	----------	--

## 8.3 CPU 时钟

CPU 时钟源来自系统时钟 SYSCLK，分频比可以通过寄存器配置为 1、2、4、8、16、32、64、256。K 时钟提供 8051 内核工作时钟。

## 8.4 SCK1 和 SCK2 时钟

SCK1 可以对 SYSCLK 做 1 到 16 分频，SCK2 可以对 SYSCLK 或 SCK1 做 1 到 16 分频，每个都带使能控制。

## 8.5 SCK3 时钟

SCK3 有 3 个时钟源，分别可以来自 SYSCLK, SCK1, SCK2。SCK3 带一个使能控制，通过使能位可以控制 SCK3 时钟的开关。SCK3 时钟自带一个中断，可以单独使能，每次当 SCK3 的上升沿到来的时候产生一次中断，用户可用该中断来做定时器。

注意：SCK3 还作为 MTP 烧录时 MTP IP 所需要的时钟，MTP IP 要求的时钟周期范围 10~15us，典型值 12.5us，我们电路实现上使用的是 SCK3 时钟的二分频。SCK3 的复位默认值为 160K，二分频后 80KHz，周期 12.5us，对应 MTP IP 要求的典型值。所有的烧录方式包括 ICP/IAP 等，在开始烧录之前都需要确保 SCK3 打开并且周期为 160KHz，否则 MTP 的烧录不可靠。

## 8.6 32K 时钟

如果 SYSCLK 域使用了，32K 时钟（来自 ILO）会同步到 SYSCLK。

## 8.7 与时钟相关寄存器定义

名字	地址	读写	复位值	描述
CLK_CR	0x94	读写	10000011	系统时钟控制寄存器
PCLK_CR	0x95	读写	11110001	外设时钟控制寄存器
PCLK_DIV12	0x96	读写	00001111	SCK1、SCK2 时钟控制寄存器
PCLK_DIV3	0x97	读写	00110001	SCK3 时钟控制寄存器

### 8.7.1 CLK\_CR (0x94)

Bit	7	6	5	4	3	2	1	0
Name	SCK3IF	-				CPUCKS[2:0]		
Reset	1	-				0	1	1
Type	R/W	-				R/W	R/W	R/W

Bit	Name	Function
7	SCK3IF	0 = 没有 SCK3 中断发生 1 = 有 SCK3 中断发生



		对该位写 1 会将其清零 注意：SCK3IF 复位值为 0，而 SCK3 默认情况下是有效的，而且会在软件启动之前就起振，因此软件看到的复位值为 0x83。
6:3	N/A	保留位，读 0
2:0	CPUCKS[2:0]	内核工作频率选择： 000 SYSCLK/8 001 SYSCLK/4 010 SYSCLK/2 011 SYSCLK 100 SYSCLK/16 101 SYSCLK/32 110 SYSCLK/64 111 SYSCLK/256

## 8.7.2 PCLK\_CR (0x95)

Bit	7	6	5	4	3	2	1	0
Name	SCK0EN	SCK1EN	SCK2EN	SCK3EN	SCK3_IE	SCK2SS	SCK3SS[1:0]	
Reset	1	1	1	1	0	0	0	1
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	SCK0EN	0 = 禁止 SCK0 时钟 1 = 使能 SCK0 时钟
6	SCK1EN	0 = 禁止 SCK1 时钟 1 = 使能 SCK1 时钟
5	SCK2EN	0 = 禁止 SCK2 时钟 1 = 使能 SCK2 时钟
4	SCK3EN	0 = 禁止 SCK3 时钟 1 = 使能 SCK3 时钟
3	SCK3_IE	0 = 禁止 SCK3 时钟中断 1 = 使能 SCK3 时钟中断
2	SCK2SS	SCK2 时钟源选择，具体使用见 SCK2CKS 说明： 0 CLK_SYS 作为 SCK2 的时钟源 1 SCK1 作为 SCK2 的时钟源
1:0	SCK3SS[1:0]	SCK3 时钟源选择： 00 关闭 SCK3 时钟 01 来自 SYSCLK 10 来自 SCK1 时钟 11 来自 SCK2 时钟

## 8.7.3 PCLK\_DIV12 (0x96)

Bit	7	6	5	4	3	2	1	0
Name	SCK1CKS				SCK2CKS			



Reset	0	0	0	0	1	1	1	1
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:4	SCK1CKS	控制 SCK1 时钟分频: $f_{SCK1} = f_{SYSCLK} / (SCK1CKS + 1)$
3:0	SCK2CKS	控制 SCK2 时钟分频: SCK2SS=0 时 $f_{SCK2} = f_{SYSCLK} / (SCK2CKS + 1)$ SCK2SS=1 时 $f_{SCK2} = f_{SYSCLK} / (SCK2CKS + 1) / (SCK1CKS + 1)$

## 8.7.4 PCLK\_DIV3 (0x97)

Bit	7	6	5	4	3	2	1	0
Name	SCK3CKS							
Reset	0	0	1	1	0	0	0	1
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	SCK3CKS	控制 SCK3 时钟的分频，频率和 SCK3SS 的值相关，具体计算方法如下： 当 SCK3SS 等于： 00 关闭 SCK3 时钟 10 $f_{SCK3} = f_{SYSCLK} / (SCK3CKS + 1)$ 10 $f_{SCK3} = f_{SYSCLK} / (SCK3CKS + 1) / (SCK1CKS + 1)$ 11 $f_{SCK3} = f_{SYSCLK} / (SCK3CKS + 1) / (SCK2CKS + 1)$





## 9. 复位

复位源有 5 个，其中包括 4 个全局复位：POR 复位、BOR 复位、引脚复位和看门狗复位，还有一个软复位。

### 9.1 POR 复位

芯片上电过程会产生 POR 复位，并且该复位信号将会一直保持到电源电压升高到芯片能够正常工作的电压为止。

### 9.2 引脚复位

P20 脚可以用作引脚复位，复位状态下 P20 默认用作普通 GPIO，通过软件配置相关寄存器可以使 P20 用作引脚复位。引脚复位低有效，复位持续 2-3 个 32K 时钟周期（约 100us）有效。

### 9.3 看门狗复位

参考 11.3。

### 9.4 欠压复位

芯片内建欠压复位（BOR）模块，如果检测到了电源电压低于欠压复位所设定的点会触发欠压复位。欠压复位模块复位后默认使能，只要发生上电复位该模块都会处于使能状态。欠压电压 4 档可调。欠压复位的寄存器描述见 13.2.1.2。

## 10. 外设

### 10.1 8-bit 基本计数器

#### 10.1.1 概述

8 位基本定时器内部包含一个 8 位自动重装向上计数器，带预分频。可以用作基本的间隔定时器中断，计时溢出可以产生中断。主要特性如下：

- 8-bit 自动重装向上计数器
- 3-bit 可编程预分频，分频比 1, 2, 4, 8, 16, 32, 64, 128
- 计数器溢出产生中断同时重装计数器
- 计数时钟可选 SCK0 时钟，32KHz 看门狗时钟

#### 10.1.2 结构框图

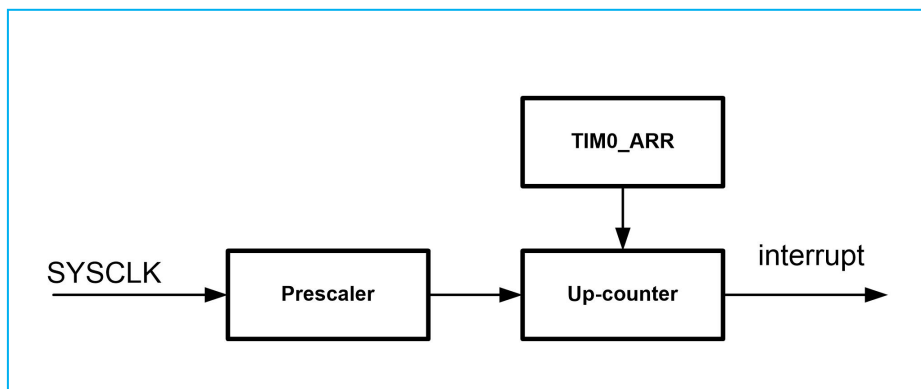


图 11 TIMERO 结构框图

#### 10.1.3 与 TIMO 相关寄存器定义

名字	地址	读写	默认值	描述
TIMO_CR	0xF8	读写	0x00	Timer0 控制寄存器
TIMO_CNTR	0xF9	只读	0x00	Timer0 计数值寄存器
TIMO_ARR	0xFA	读写	0x00	Timer0 自动重装寄存器
TIMO_IE	0xFB	读写	0x00	Timer0 中断控制寄存器
TIMO_SR	0xFC	读写	0x00	Timer0 状态寄存器

##### 10.1.3.1 TIMO\_CR (0xF8)

Bit	7	6	5	4	3	2	1	0
Name	-		TIMO_CLKSEL[1:0]		TIMO_CLKDIV[2:0]			TIMO_EN
Reset	-		0	0	0	0	0	0
Type	-		R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	N/A	保留位，读 0



5:4	TIMO_CLKSEL[1:0]	TIMERO 时钟选择: 00 SCK0 01 内部 32K 时钟 10/11 保留
3:1	TIMO_CLKDIV[2:0]	TIMERO 预分频选择: 000 1 分频 001 2 分频 010 4 分频 011 8 分频 100 16 分频 101 32 分频 110 64 分频 111 128 分频
0	TIMO_EN	0 = TIMERO 关 1 = TIMERO 开 注意: 修改 TIMO_CLKSEL 和 TIMO_CLKDIV 寄存器配置必须在 TIMO_EN 为 0 的时候进行。

### 10.1.3.2 TIMO\_CNTR (0xF9)

Bit	7	6	5	4	3	2	1	0
Name	TIMO_CNTR							
Reset	0	0	0	0	0	0	0	0
Type	RO:0	RO:0	RO:0	RO:0	RO:0	RO:0	RO:0	RO:0

Bit	Name	Function
7:0	TIMO_CNTR	TIMERO 计数值寄存器

### 10.1.3.3 TIMO\_ARR (0xFA)

Bit	7	6	5	4	3	2	1	0
Name	TIMO_ARR							
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	TIMO_ARR	TIMERO 自动重装寄存器。

### 10.1.3.4 TIMO\_IE (0xFB)

Bit	7	6	5	4	3	2	1	0
Name								TIMO_TCIE
Reset								0
Type								R/W

Bit	Name	Function
7:1	N/A	保留位, 读 0
0	TIMO_TCIE	0 = 溢出中断关 1 = 溢出中断开



## 10.1.3.5 TIMO\_SR (0xFC)

Bit	7	6	5	4	3	2	1	0
Name								TIMO_TC
Reset								0
Type								R/W

Bit	Name	Function
7:1	N/A	保留位，读 0
0	TIMO_TC	定时器 0 溢出标志位： 0 TIMER0 未发生溢出 1 TIMER0 发生溢出 写 1 清零该标志位，写 0 无效

## 10.2 16-bit 高级计数器

### 10.2.1 概述

高级定时器是一个包含三个定时器 TIMER1/2/3。TIMER1/2/3 是功能相同的高级计数器，可用于产生不同形式的时钟波形，一个定时器可以产生同频的一组互补 PWM 或者 2 路 PWM 独立输出。可以捕获外界输入进行脉冲宽度或周期测量。

### 10.2.2 主要特性

主要特性如下：

- 内置 16 位计数器，向上或者向下计数，自动重装
- 支持三角波计数模式和锯齿波计数模式
- 支持计数周期自动重装载
- 支持 6 种时钟源
  - 系统时钟:SCK0
  - 32KHz 时钟
  - 定时器输入通道 A 上升沿（需打开捕获使能）
  - 定时器输入通道 B 上升沿（需打开捕获使能）
  - 定时器输入通道 A 下降沿（需打开捕获使能）
  - 定时器输入通道 B 下降沿（需打开捕获使能）
- 时钟源预分频，分频系数 1~16
- 输入捕获（上升沿，下降沿和双沿）和比较输出功能
- 对输入沿计数，可选上升沿，下降沿和双边沿
- 可选输入 CHA/CHB 的沿（上升沿，下降沿）作为时钟，进行计数。
- 刹车输入，可以将 TIMER1/2/3 的输出置为特定的状态（高电平，低电平，高阻态）
- 支持输入捕获功能和比较输出功能的周期间隔相应，响应间隔周期为 1、2、4、8、16、32、64、128
- 支持 timer2 捕获 timer1，支持 timer3 捕获 timer1

- 支持 PWM 输出功能
  - 可输出 2 路独立 PWM 或者 1 路互补 PWM，互补输出可编程死区
  - 支持刹车功能，刹车输入包括：ADC 输出，外部引脚 BKIN(输入影子寄存器，对应寄存器按顺序写入更新（先写高 8 位，再写低 8 位）
  - 支持三角波模式和锯齿波模式的 PWM 输出控制
- 中断，在以下事件产生中断：
  - 计数器上溢或下溢
  - 输入捕获
  - 比较输出
  - 刹车产生

### 10.2.3 结构框图

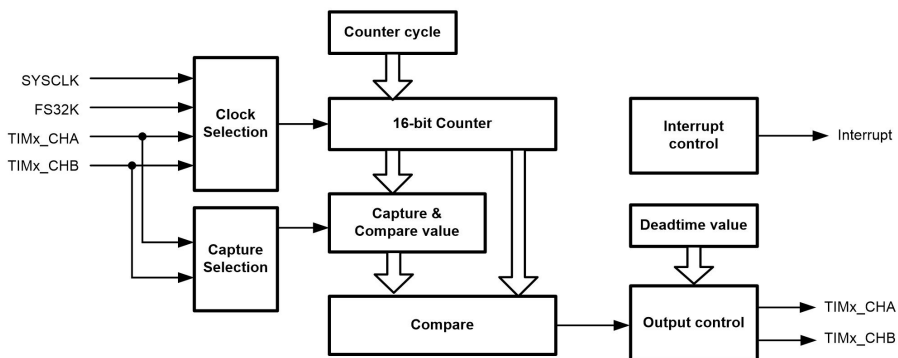


图 12 定时器结构图

### 10.2.4 基本动作

#### 基本波形模式

TIMER1/2/3 有 2 种基本计数波形模式，锯齿波模式和三角波模式。

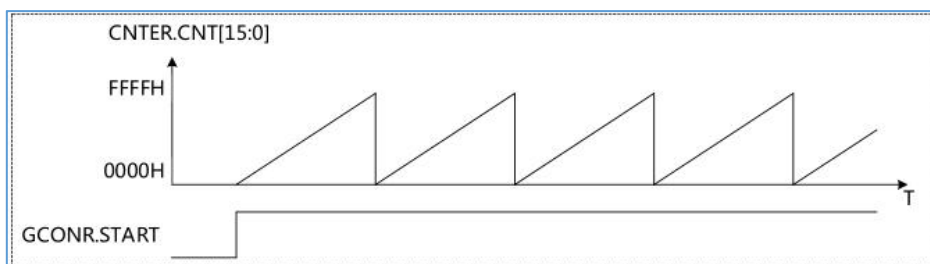
锯齿波模式：

- 向上计数：计数器每节拍增加 1，直至等于计数周期值时重载为 0；
- 向下计数：计数器每节拍减少 1，直至为 0 时自动加载计数周期值；

三角波模式：

- 向上计数：计数器每节拍增加 1，直至等于计数周期值时计数器每节拍减少 1，直至为 0；
- 向下计数：计数器每节拍减少 1，直至等于 0 时计数器每节拍增加 1，直至为计数周期值；

波形模式又由于不同的内部计数动作有所细分，三角波模式分为三角波 A 模式。锯齿波和三角波的基本波形如所示。三角波 A 模式一个周期只发生一次缓存传送（谷点）。



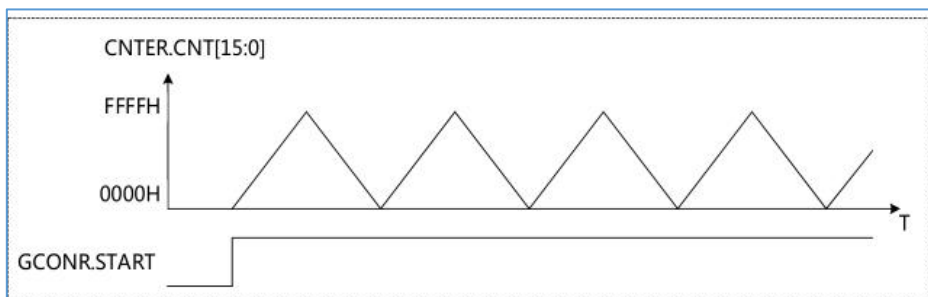


图 13 计数模式示意图

### 比较输出

TIMER1/2/3 一个定时器有 2 个比较输出端口 (TIMx\_CHA、TIMx\_CHB)，可在计数值与计数基准值比较匹配时输出指定的电平。GCMAR、GCMBR 寄存器分别对应了 TIMx\_CHA、TIMx\_CHB 的计数比较基准值。当计数器的计数值和 GCMAR 相等时，TIMx\_CHA 端口输出指定的电平；当计数器的计数值和 GCMBR 相等时，TIMx\_CHB 端口输出指定电平。

TIMx\_CHA、TIMx\_CHB 端口的计数起始电平和计数比较匹配时的电平由 TIM1\_PCONRA.PA\_INITVAL 和 TIM1\_PCONRA.CAPA\_OUT 定义。下图为比较输出的动作示例。

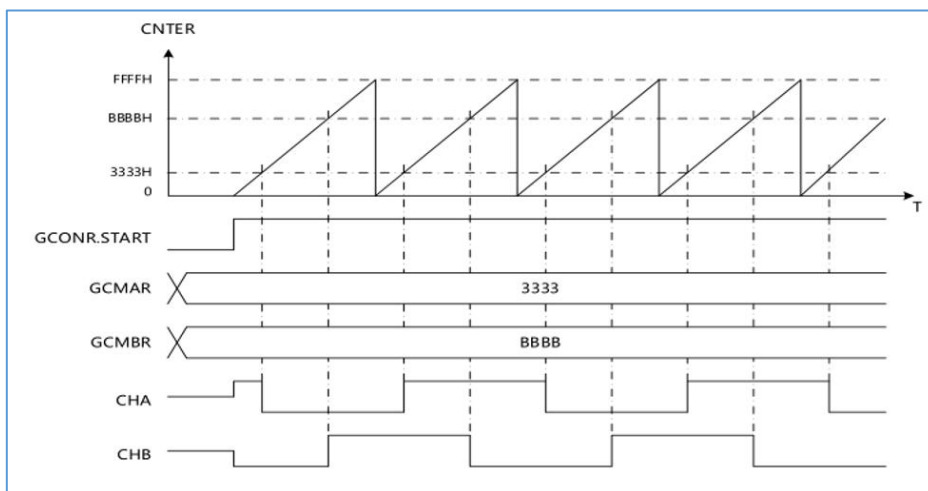


图 14 比较输出示意图

### 捕获输入

TIMER1/2/3 都具有捕获输入功能，具备 2 组捕获输入寄存器 (GCMAR\_S、GCMBR\_S)，用于保存捕获到的计数值。设定端口控制寄存器 (PCONRA/ PCONRB) 的 capa\_en/capb\_en 位为 1，对应端口的捕获输入功能就有效了。当设定了对应的捕获输入条件且该条件有效时，当前的计数值就被保存到相应的寄存器 (GCMAR\_S、GCMBR\_S) 中。每组捕获输入的条件可选 TIMx\_CHA 或 TIMx\_CHB 的上升沿，下降沿或上升下降沿，通过 CAPA\_MODE/CAPB\_MODE 来设定对应端口的捕获条件。图为捕获输入的动作例。

捕获是根据外部信号的沿采样内部计数器的值，TIM1\_ARR\_L 和 TIM1\_ARR\_H 这两个寄存器决定了定时器内部计数器的溢出时间，捕获模式要设置，建议两个寄存器都设置成 0xFF，捕获模式推荐使用三角波模式，三角波模式的捕获图参考下图。

捕获模式读取这两个寄存器的值要把 TIMx\_CR 的 SEL\_SREG 设置成 0 才能读到真的捕获值，否则读取的是配置寄存器时写入的 GCMAR 和 GCMBR 值。SEL\_SREG 只影响这两个寄存器的读，捕获模式下写这两个寄存器没有意义。

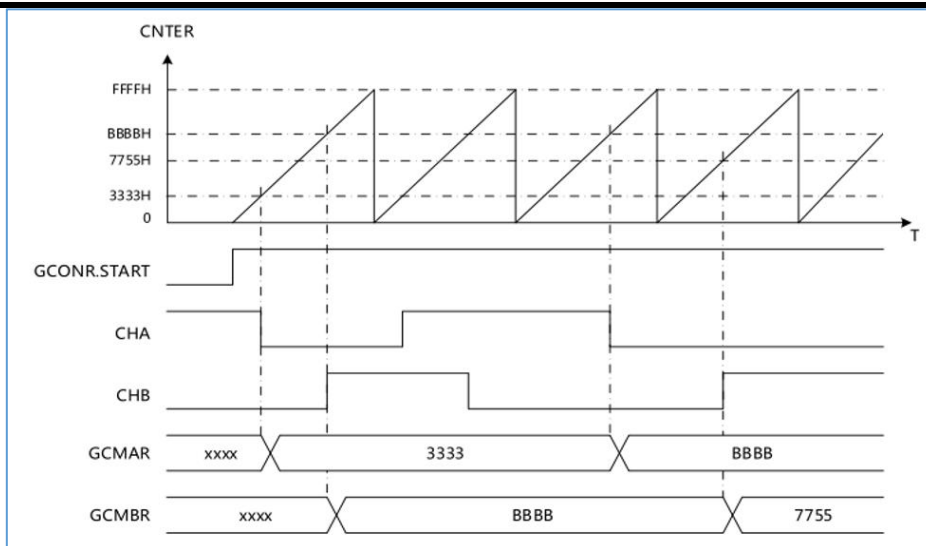


图 15 捕获模式示意图

### 10.2.5 时钟源选择

TIMER1/2/3 的计数时钟可以有以下几种选择：

系统时钟 (SCK0)

内部低速 RC 振荡器 32kHz 时钟

时钟分频 1-16 可选。

可选输入 CHA/CHB 作为时钟，此时可对 CHA/CHB 沿（上升沿，下降沿可选）计数，备注：需要打开捕获功能。

### 10.2.6 计数方向

TIMER1/2/3 的计数器计数方向可通过软件方式改变。不同波形模式时，改变计数方向的方法略有不同。

#### 10.2.6.1 锯齿波计数方向

锯齿波模式时，计数方向可在计数器计数中或者计数停止时设定。

在向上计数中时，设定 GCONR.DIR=0（向下计数），则计数器计数到上溢后变为向下计数模式；在向下计数中时，设定 GCONR.DIR=1（向上计数），则计数器计数到下溢后变为向上计数模式。

在计数停止时，设定 GCONR.DIR 位。则计数开始后直至上溢或下溢时，GCONR.DIR 的设定才会反映到计数中。

#### 10.2.6.2 三角波计数方向

三角波模式时，计数方向只能在计数器停止时设定。在计数中设定计数方向无效。在计数停止时，设定 CR.DIR 位。则计数开始后直至上溢或下溢时，CR.DIR 的设定才会反映到计数中。

### 10.2.7 数字滤波

TIMER1/2/3 的 TIMX\_CHA、TIMX\_CHB 端口输入都有数字滤波功能。可通过设置 PA\_FILTER\_EN 和 PB\_FILTER\_EN 开启对应端口的滤波功能。滤波时钟为计数器当前工作时钟。

在滤波采样基准时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。其动作例如所示。

数字滤波也用于对电压比较器传过来的信号滤波，通过 CHA\_FILTER\_EN/THB\_FILTER\_EN 开启，此时滤波时钟为系统时钟。

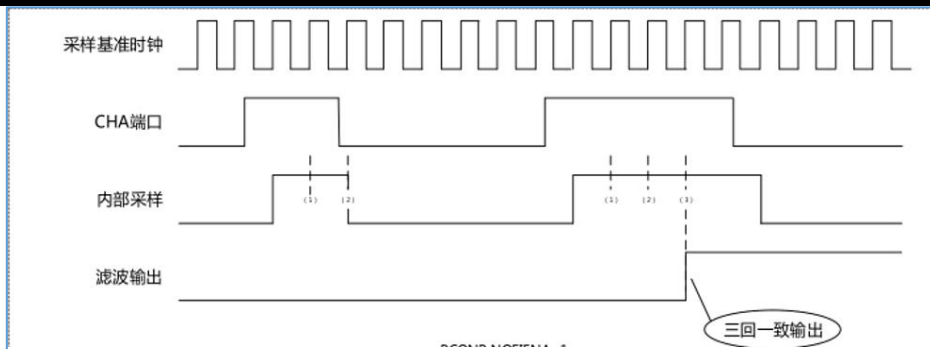


图 16 数字滤波示意图

### 10.2.8 软件同步

TIMER1/2 可通过设定软件同步启动寄存器 (SSCONR)，实现目标 TIMER1/2/3 的同步启动。

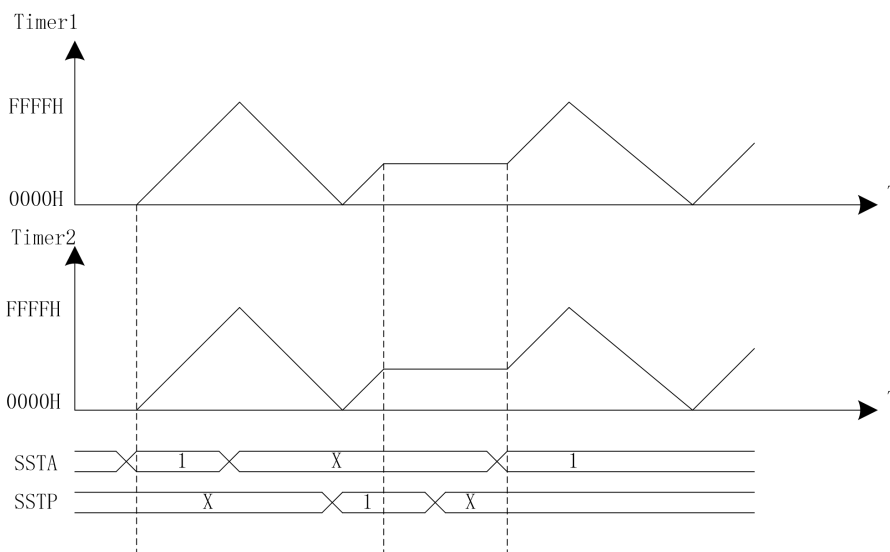


图 17 软件同步示意图

#### 10.2.8.1 软件同步停止

TIMER1/2/3 可通过设定软件同步停止寄存器 (SSCONR)，实现目标 TIMER1/2/3 的同步暂停，此时计数器处于暂停状态，对同步启动寄存器 (SSCONR) 写 1 可以继续计数。

#### 10.2.8.2 软件同步清零

TIMER1/2/3 可通过设定软件同步清零寄存器 (SSCONR)，实现目标 TIMER1/2/3 的同步清零，此时计数器会复位到初始状态。

若设定 SSCONR，即可实现 TIMER1/2/3 的软件同步启动。

软件同步动作相关寄存器 (SSCONR) 是一组独立于 TIMER1/2/3 外的寄存器，这组寄存器的各个位只在写 1 时有效，写 0 无效。在读取 SSCONR 寄存器时，会读出 0。

### 10.2.9 缓存功能

缓存动作是指在缓存传送时间点，发生以下事件：

- 通用周期基准值缓存寄存器 (TIMx\_ARRL、TIMx\_ARRH) 的值自动传送到通用周期基准值寄存器 (TIMx\_ARRL\_S、TIMx\_ARRH\_S) 中；
- 通用比较基准值缓存寄存器 (GCMAR、GCMBR) 的值自动传送到通用比较基准值寄存器 (GCMAR\_S、GCMBR\_S) 中 (比较输出时)；



- c. 通用比较基准值寄存器(GCMAR、GCMBR)的值自动传送到通用比较基准值缓存寄存器(GCMAR\_S、GCMBR\_S)中(捕获输入时);  
 如图所示,是比较输出动作时、通用比较基准值寄存器的单缓存方式的时序图。从中可以看到,在计数期间改变通用比较基准值寄存器(GCMAR)的值可以调整输出占空比,改变通用周期基准值寄存器(TIMx\_ARRL、TIMx\_ARRH)的值可以调整输出周期。

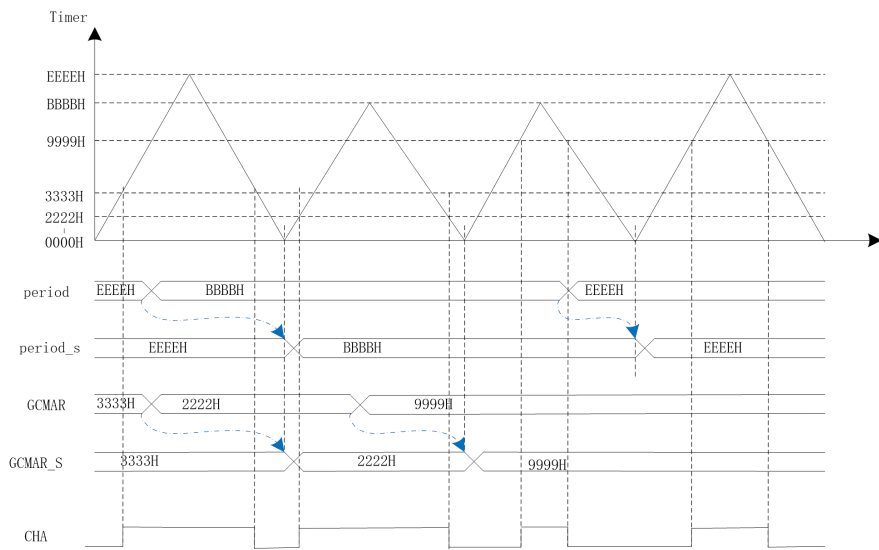


图 18 缓存功能示意图

### 10.2.9.1 缓存传送时间点

周期值缓存传送时间点为锯齿波时递增计数上溢点或递减计数下溢点、三角波时计数谷点。

锯齿波模式时,缓存传送发生在上溢点或下溢点。

三角波 A 模式时,缓存传送发生在计数谷点。

捕获输入动作缓存传送时间点为捕获输入动作时。

在锯齿波计数模式或硬件计数模式时,正常的比较输出动作期间若有清零动作产生,通用周期基准值、通用比较基准值、等会根据相应的缓存动作设定状况发生一次缓存传送。

## 10.2.10 通用 PWM 输出

### 10.2.10.1 独立 PWM 输出

每个定时器的 2 个端口 TIMx\_CHA、TIMx\_CHB 能独立的输出 PWM 波。如图所示,定时器 Timer1 的 CHA 端口输出 PWM 波。(PWM 输出时,需要使能 TIx\_MOE 位)

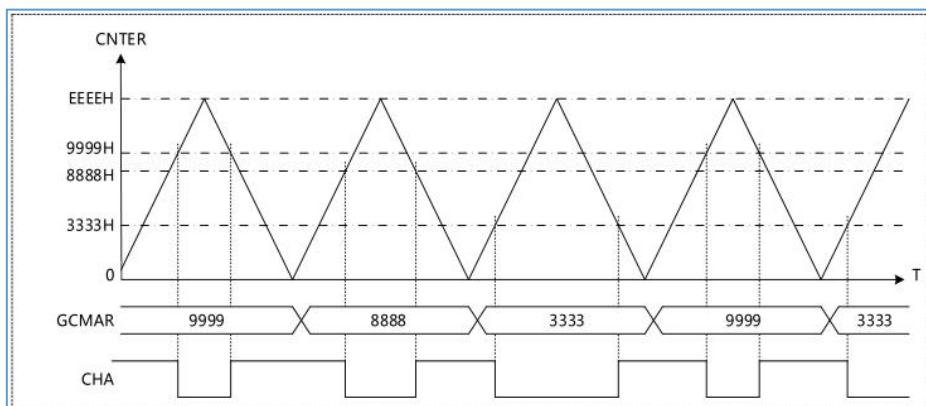


图 19 PWM 输出示意图

### 10.2.10.2 互补 PWM 输出

TIMx\_CHA 端口和 TIMx\_CHB 端口,在不同的模式下可组合输出互补 PWM 波形。(PWM 输出时,需要使能 TIx\_MOE 位)。

#### 软件设定 GCMBR 互补 PWM 输出

软件设定 GCMBR 互补 PWM 输出是指在锯齿波模式和三角波模式下,用于 TIMx\_CHB 端口波形输出的通用比较基准值寄存器 (GCMBR) 的值由寄存器直接设定,与通用比较基准值寄存器 (GCMAR) 的值没有直接关系。下图为软件设定 GCMBR 互补 PWM 波的示例。

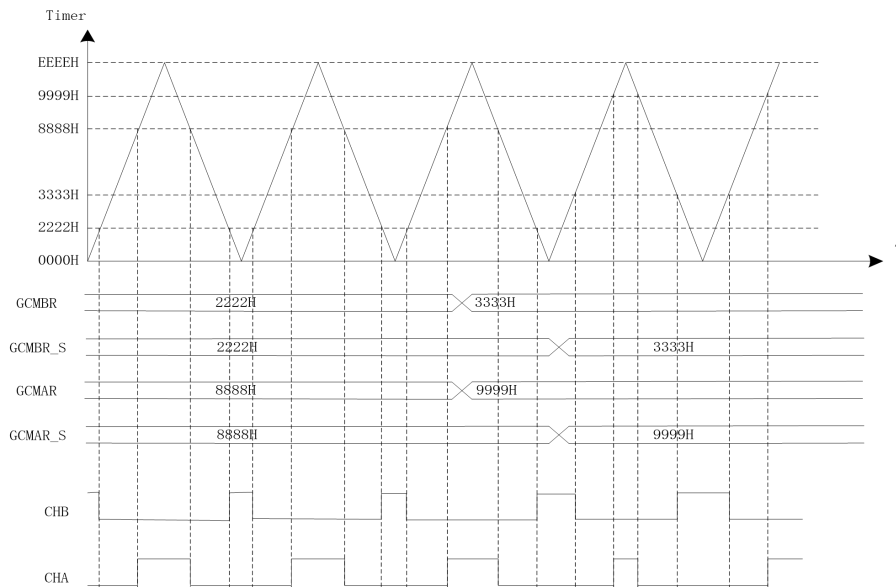


图 20 互补 PWM 输出示意图 1

#### 硬件设定 GCMBR 互补 PWM 输出

硬件设定 GCMBR 互补 PWM 输出是指在三角波 A 模式下,用于 TIMx\_CHB 端口波形输出的通用比较基准值寄存器 (GCMBR) 的值由通用比较基准值寄存器 (GCMAR) 和死区时间基准值寄存器 (DTUA) 的值运算决定。图为硬件设定 GCMBR 互补 PWM 波输出例。死区时间基准值寄存器 (DTUA) 为 8bit,调整范围为 0~255。(PWM 输出时,需要使能 TIx\_MOE 位)。

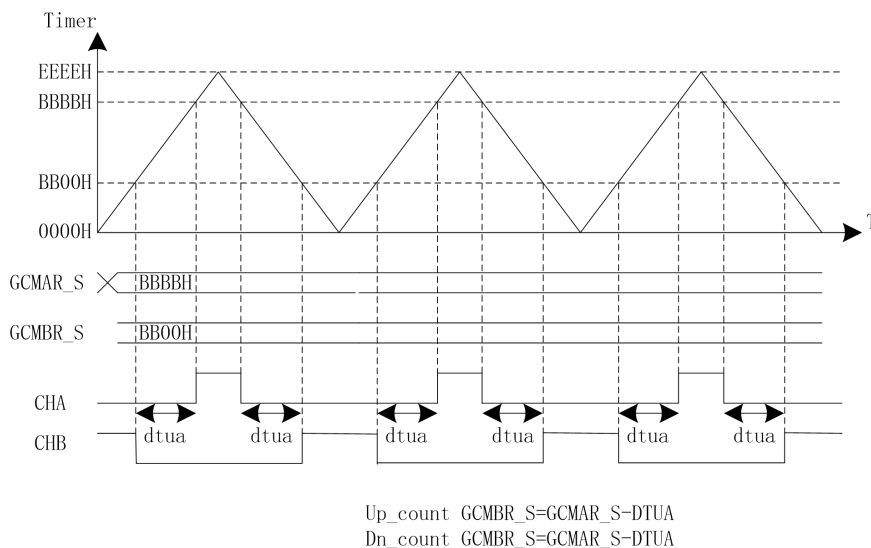


图 21 互补 PWM 输出示意图 2

### 10.2.11 周期间隔响应

TIMER1/2/3 的通用比较基准值寄存器 (GCMAR, GCMBR),在计数比较匹配时可分别产生专用有效请求信号。

该请求信号可以每间隔几个周期后产生一次有效的请求信号。通过设定有效周期寄存器（VPERR）的 VPERR.PCNTS 位来指定每隔多少个周期请求信号有效一次，其它周期内即使计数值和比较基准值寄存器 GCMAR 或 GCMBR 的值相等，也不会输出有效的请求信号。图所示是周期间隔有效请求信号的动作例。

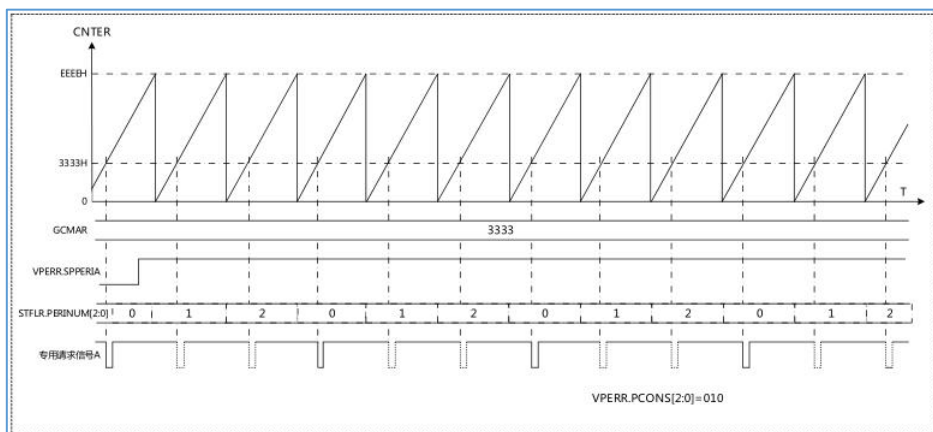


图 22 周期间隔响应示意图

## 10.2.12 保护机制

高级计数器可以对端口的输出状态进行保护控制。

高级计数器有 4 个共用的端口输入无效事件（来自模拟比较器 0 和 1、ADC、外部 BKIN），每个接口上选通的异常状况事件可从刹车控制设定（TIMx\_BRAKE、TIMx\_DTR 寄存器决定刹车使能和刹车事件来源），当这些接口上监测到异常状况时，可以实现对通用 PWM 输出的控制。

端口作为通用 PWM 输出端口在刹车控制异常事件发生时，端口状态可以变为输出高阻态、输出低电平或输出高电平（由对应的 GPIO 的配置决定）。

## 10.2.13 中断说明

TIMER1/2/3 各含有 4 类共计 6 个中断。分别是 2 个通用计数比较匹配中断（含 2 个捕获输入中断）、2 个计数周期匹配中断、2 个刹车保护中断。

## 10.2.14 内部互连

- 电压比较器 0 和 1 可以触发刹车功能。
- ADC 输出可以触发刹车功能。
- 外部 BKIN 引脚可以触发刹车功能。
- TIMER1/2/3 中断可以触发 ADC 采样功能。

## 10.2.15 保护寄存器

保护寄存器用来控制对 Timer 的周期寄存器，比较值寄存器，控制寄存器等的更改，只有当保护寄存器中的数据为 0xCA，才能配置其它 TIMER 的寄存器，否则不能修改。

该寄存器能增强 Timer 的抗干扰能力。

## 10.2.16 timer2 捕获 timer1

timer2 捕获 timer1 的时钟和定时器，触发源为 timer2 的 CHx 端口。

## 10.2.17 timer3 捕获 timer1

timer3 捕获 timer1 的时钟和定时器，触发源为 timer3 的 CHx 端口。



## 10.2.18 TIM1 和 TIM2 和 TIM3 相关寄存器定义

名字	地址	读写	复位值	描述
TIM1_PR	0xC3	读写	00000000	Timer1 密码寄存器
TIM1_CR	0xC0	读写	00000000	Timer1 控制寄存器
TIM1_IE	0xC1	读写	00000000	Timer1 中断控制寄存器
TIM1_SR	0xC2	读写	00000000	Timer1 状态寄存器
TIM1_FCONR	0xFF50	读写	00000000	Timer1 时钟控制寄存器
TIM1_VPERR	0xFF51	读写	00000000	Timer1 周期间隔响应控制寄存器
TIM1_DTUA	0xFF52	读写	00000000	Timer1 死区事件寄存器
TIM1_BRAKE	0xFF53	读写	00000000	Timer1 刹车控制寄存器
TIM1_DTR	0xFF54	读写	00000000	Timer1 死区控制寄存器
TIM1_PCONRA	0xFF55	读写	00000000	Timer1 端口 A 控制寄存器
TIM1_PCONRB	0xFF56	读写	00000000	Timer1 端口 B 控制寄存器
TIM1_CNTL	0xFF58	只读	00000000	Timer1 计数值寄存器低 8 位
TIM1_CNTH	0xFF59	只读	00000000	Timer1 计数值寄存器高 8 位
TIM1_ARRL	0xFF5A	读写	00000000	Timer1 自动重载寄存器低 8 位
TIM1_ARRH	0xFF5B	读写	00000000	Timer1 自动重载寄存器高 8 位
TIM1_GCMARL	0xFF5C	读写	00000000	Timer1 比较捕获寄存器 A 低 8 位
TIM1_GCMARH	0xFF5D	读写	00000000	Timer1 比较捕获寄存器 A 高 8 位
TIM1_GCMBRL	0xFF5E	读写	00000000	Timer1 比较捕获寄存器 B 低 8 位
TIM1_GCMBRH	0xFF5F	读写	00000000	Timer1 比较捕获寄存器 B 高 8 位
TIM2_PR	0xCB	读写	00000000	Timer2 密码寄存器
TIM2_CR	0xC8	读写	00000000	Timer2 控制寄存器
TIM2_IE	0xC9	读写	00000000	Timer2 中断控制寄存器
TIM2_SR	0xCA	读写	00000000	Timer2 状态寄存器
TIM2_FCONR	0xFF60	读写	00000000	Timer2 时钟控制寄存器
TIM2_VPERR	0xFF61	读写	00000000	Timer2 周期间隔响应控制寄存器
TIM2_DTUA	0xFF62	读写	00000000	Timer2 死区事件寄存器
TIM2_BRAKE	0xFF63	读写	00000000	Timer2 刹车控制寄存器
TIM2_DTR	0xFF64	读写	00000000	Timer2 死区控制寄存器
TIM2_PCONRA	0xFF65	读写	00000000	Timer2 端口 A 控制寄存器
TIM2_PCONRB	0xFF66	读写	00000000	Timer2 端口 B 控制寄存器
TIM2_CNTL	0xFF68	只读	00000000	Timer2 计数值寄存器低 8 位
TIM2_CNTH	0xFF69	只读	00000000	Timer2 计数值寄存器高 8 位
TIM2_ARRL	0xFF6A	读写	00000000	Timer2 自动重载寄存器低 8 位
TIM2_ARRH	0xFF6B	读写	00000000	Timer2 自动重载寄存器高 8 位
TIM2_GCMARL	0xFF6C	读写	00000000	Timer2 比较捕获寄存器 A 低 8 位
TIM2_GCMARH	0xFF6D	读写	00000000	Timer2 比较捕获寄存器 A 高 8 位
TIM2_GCMBRL	0xFF6E	读写	00000000	Timer2 比较捕获寄存器 B 低 8 位
TIM2_GCMBRH	0xFF6F	读写	00000000	Timer2 比较捕获寄存器 B 高 8 位
TIM3_PR	0xCF	读写	00000000	Timer3 密码寄存器
TIM3_CR	0xD8	读写	00000000	Timer3 控制寄存器
TIM3_IE	0xD9	读写	00000000	Timer3 中断控制寄存器
TIM3_SR	0xDA	读写	00000000	Timer3 状态寄存器
TIM3_FCONR	0xFF70	读写	00000000	Timer3 时钟控制寄存器



TIM3_VPERR	0xFF71	读写	00000000	Timer3 周期间隔响应控制寄存器
TIM3_DTUA	0xFF72	读写	00000000	Timer3 死区事件寄存器
TIM3_BRAKE	0xFF73	读写	00000000	Timer3 刹车控制寄存器
TIM3_DTR	0xFF74	读写	00000000	Timer3 死区控制寄存器
TIM3_PCONRA	0xFF75	读写	00000000	Timer3 端口 A 控制寄存器
TIM3_PCONRB	0xFF76	读写	00000000	Timer3 端口 B 控制寄存器
TIM3_CNTL	0xFF78	只读	00000000	Timer3 计数值寄存器低 8 位
TIM3_CNTH	0xFF79	只读	00000000	Timer3 计数值寄存器高 8 位
TIM3_ARRL	0xFF7A	读写	00000000	Timer3 自动重载寄存器低 8 位
TIM3_ARRH	0xFF7B	读写	00000000	Timer3 自动重载寄存器高 8 位
TIM3_GCMARL	0xFF7C	读写	00000000	Timer3 比较捕获寄存器 A 低 8 位
TIM3_GCMARH	0xFF7D	读写	00000000	Timer3 比较捕获寄存器 A 高 8 位
TIM3_GCMBRL	0xFF7E	读写	00000000	Timer3 比较捕获寄存器 B 低 8 位
TIM3_GCMBRH	0xFF7F	读写	00000000	Timer3 比较捕获寄存器 B 高 8 位
SSCONR	0xFD	读写	00000000	TIMER1/2/3 软件同步控制寄存器

10.2.18.1 TIM1\_PR (0xC3)

Bit	7	6	5	4	3	2	1	0
Name	TIMKEY							
Reset	0							
Type	R/W							

Bit	Name	Function
7:0	TIMKEY	TIMER1 寄存器写保护密钥，写 0xCA 打开，打开后才能配置其它 TIMER1 的寄存器；密钥打开时读出值为 0xCA，否则读出值为 0

10.2.18.2 TIM1\_CR (0xC0)

Bit	7	6	5	4	3	2	1	0
Name	THB_FILTER_EN	THA_FILTER_EN	ADC_FILTER_EN	SEL_SREG	DIR	MODE[1:0]	TIM1_EN	
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	THB_FILTER_EN	刹车输入 B 滤波控制 0 刹车输入 B 关数字滤波 1 刹车输入 B 开数字滤波 TIMER1/2/3 共用, 只在 TIMER1 设定, TIMER2/3 共用 TIMER1 设定
6	THA_FILTER_EN	刹车输入 A 滤波控制 0 刹车输入 A 关数字滤波 1 刹车输入 A 开数字滤波 TIMER1/2/3 共用, 只在 TIMER1 设定, TIMER2/3 共用 TIMER1 设定
5	ADC_FILTER_EN	刹车输入 ADC 比较输出滤波和 BRKIN 管脚输入滤波控制 0 刹车输入 ADC 比较输出和 BRKIN 管脚关数字滤波 1 刹车输入 ADC 比较输出和 BRKIN 管脚开数字滤波 TIMER1/2 共用, 只在 TIMER1 设定, TIMER2 共用 TIMER1 设定
4	SEL_SREG	影子寄存器控制



		0 ARR GCMAR GCMBR 读到影子寄存器的值或捕获值 1 ARR GCMAR GCMBR 读到当前设定的值
3	DIR	计数器计数方向 0 向上计数 1 向下计数
2:1	MODE[1:0]	计数器计数模式 00 锯齿波计数模式 01 三角波计数模式 10/11 保留
0	TIM1_EN	TIMER1 使能控制 0 关闭 TIMER1 1 使能 TIMER1

### 10.2.18.3 TIM1\_FCONR (0xFF50)

Bit	7	6	5	4	3	2	1	0
Name	-	CLK_SEL[2:0]			PRE_DIV[3:0]			
Reset	-	0	0	0	0	0	0	0
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	N/A	保留位, 读 0
6:4	CLK_SEL[2:0]	TIMER1 时钟源选择: 000 SYSCLK 001 看门狗时钟 32kHz 010 保留 011 保留 100 TIM1_CHA 上升沿 (打开捕获功能) 101 TIM1_CHB 上升沿 (打开捕获功能) 110 TIM1_CHA 下降沿 (打开捕获功能) 111 TIM1_CHB 下降沿 (打开捕获功能)
3:0	PRE_DIV[3:0]	TIMER1 预分频选择: 0~15 对应 1~16 分频 注意: 修改 CLK_SEL 和 PRE_DIV 寄存器配置必须在 TIM1_EN 为 0 的时候进行。

### 10.2.18.4 TIM1\_CNTL (0xFF58)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_CNTL[7:0]							
Reset	0x00							
Type	R0:0							

Bit	Name	Function
7	TIM1_CNTR[7:0]	计数器计数寄存器低 8 位。

### 10.2.18.5 TIM1\_CNTH (0xFF59)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_CNTH							



Reset	0x00
Type	R0:0

Bit	Name	Function
7:0	TIM1_CNTR[15:8]	计数器计数寄存器高 8 位。

## 10.2.18.6 TIM1\_ARRL (0xFF5A)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_ARRL							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM1_ARRL	自动重载值寄存器低 8 位，需先写高 8 位再写低 8 位。

## 10.2.18.7 TIM1\_ARRH (0xFF5B)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_ARRH							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM1_ARRH	自动重载值寄存器高 8 位，需先写高 8 位再写低 8 位。

## 10.2.18.8 TIM1\_GCMARL (0xFF5C)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_GCMARL							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM1_GCMARL	计数模式下比较值，捕获模式下 CHA 捕获值，GCMAR 低 8 位，需先写高 8 位再写低 8 位。

## 10.2.18.9 TIM1\_GCMARH (0xFF5D)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_GCMARH							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM1_GCMARH	计数模式下比较值，捕获模式下 CHA 捕获值，GCMAR 高 8 位，需先写高 8 位再写低 8 位。

## 10.2.18.10 TIM1\_GCMBRL (0xFF5E)

Bit	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---



Name	TIM1_GCMBRL
Reset	0x00
Type	R/W

Bit	Name	Function
7:0	TIM1_GCMBRL	计数模式下比较值，捕获模式下 CHB 捕获值，GCMBR 低 8 位，需先写高 8 位再写低 8 位。

## 10.2.18.11 TIM1\_GCMBRH (0xFF5F)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_GCMBRH							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM1_GCMBRH	计数模式下比较值，捕获模式下 CHB 捕获值，GCMAR 高 8 位，需先写高 8 位再写低 8 位。

## 10.2.18.12 TIM1\_VPERR (0xFF51)

Bit	7	6	5	4	3	2	1	0
Name	-	-	PCNTE[1:0]		-	PCNTS[2:0]		
Reset	-	-	0	0	-	0	0	0
Type	-	-	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Function
7:6	N/A	保留位，读 0
5:4	PCNTE[1:0]	周期间隔响应计数条件： 00 有效周期选择功能无效 01 锯齿波计数上、下溢点或三角波波峰作为计数条件 10 锯齿波计数上、下溢点或三角波波谷作为计数条件 (注意：由于 timer 使能后一段时间后才将周期值加载进 counter 寄存器中，这样会导致误判第一个波谷，可以选择软解规避) 11 锯齿波计数上、下溢点或三角波波谷、波峰作为计数条件 (注意：向下计数条件下不建议使用)
3	N/A	保留位，读 0
2:0	PCNTS[2:0]	周期间隔响应周期： 000 1 个周期响应一次 001 2 个周期响应一次 010 4 个周期响应一次 011 8 个周期响应一次 100 16 个周期响应一次 101 32 个周期响应一次 110 64 个周期响应一次 111 128 个周期响应一次





## 10.2.18.13 TIM1\_DTUA (0xFF52)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_DTUA							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM1_DTUA	TIMER1 死区时间设定值。

## 10.2.18.14 TIM1\_BRAKE (0xFF53)

Bit	7	6	5	4	3	2	1	0
Name	TIB_MOE	TIB_AOE	TIB_SEL	TIB_EN	TIA_MOE	TIA_AOE	TIA_SEL	TIA_EN
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	TIB_MOE	TIM1_CHB 主输出使能： 刹车事件有效时，会立即被同步清零。根据 AOE 的选择，通过软件置 1 或硬件自动置 1。 1 TIM1_CHB 主输出有效 0 TIM1_CHB 主输出关闭
6	TIB_AOE	自动输出使能： 1 有刹车事件产生时，MOE 可被软件和溢出事件置 1 0 有刹车事件产生时，MOE 只被软件置 1
5	TIB_SEL	选择 TIM1_CHB 刹车来源，由 {TIM1_DTR. TIB_ADCS, TIM1_BRAKE. TIB_SEL} 的值共同决定 {TIB_ADCS, TIB_SEL } 00 TIM1_CHB 刹车事件选择模拟比较器 1 输出 01 TIM1_CHB 刹车事件选择模拟比较器 0 输出 10 TIM1_CHB 刹车事件选择 ADC 比较输出 11 TIM1_CHB 刹车事件选择 P1.4 BRKIN 输入
4	TIB_EN	刹车功能控制： 1 TIM1_CHB 刹车有效 0 TIM1_CHB 刹车无效
3	TIA_MOE	TIM1_CHA 主输出使能（PWM 输出时，需要使能该位）： 刹车事件有效时会立即被同步清零。根据 AOE 的选择，通过软件置 1 或硬件自动置 1 1 TIM1_CHA 主输出有效 0 TIM1_CHA 主输出关闭
2	TIA_AOE	TIM1_CHA 自动输出使能： 1 有刹车事件产生时，MOE 可被软件和溢出事件置 1 0 有刹车事件产生时，MOE 只被软件置 1
1	TIA_SEL	选择 TIM1_CHA 刹车来源，由 {TIM1_DTR. TIA_ADCS, TIM1_BRAKE. TIA_SEL} 的值共同决定 {TIA_ADCS, TIA_SEL } 00 TIM1_CHA 刹车事件选择模拟比较器 1 输出 01 TIM1_CHA 刹车事件选择模拟比较器 0 输出 10 TIM1_CHA 刹车事件选择 ADC 比较输出 11 TIM1_CHA 刹车事件选择 P1.4 BRKIN 输入



0	TIA_EN	TIM1_CHA 刹车功能控制： 1 TIM1_CHA 刹车有效 0 TIM1_CHA 刹车无效
---	--------	--

## 10.2.18.15 TIM1\_DTR (0xFF54)

Bit	7	6	5	4	3	2	1	0
Name	TIB_ADCS	TIA_ADCS	–	HW_CPWM	DTB_HO	DTB_EN	DTA_HO	DTA_EN
Reset	0	0	–	0	0	0	0	0
Type	R/W	R/W	–	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	TIB_ADCS	TIM1_CHB 刹车功能控制，见 TIM1_BRAKE 中 TIB_SEL 说明
6	TIA_ADCS	TIM1_CHA 刹车功能控制，见 TIM1_BRAKE 中 TIA_SEL 说明
5	N/A	保留位，读 0
4	HW_CPWM	控制 GCMBR 互补模式 0 硬件设定 GCMBR 互补 PWM 输出模式关 1 硬件设定 GCMBR 互补 PWM 输出模式开
3	DTB_HO	控制死区输出状态 1 保留 0 输出 B 死区置为 0 或 1（由 GPIO 输出决定） 备注：该寄存器不能写 1，只能写 0。功能支持，当刹车事件有效或者死区时，PWM 的输出由对应管脚的 GPIO 设置决定，即此时是作为普通 GPIO 使用。
2	DTB_EN	死区控制使能 1 输出 B 死区控制有效 0 输出 B 死区控制无效
1	DTA_HO	控制死区输出状态 1 保留 0 输出 A 死区置为 0 或 1（由 GPIO 输出决定） 备注：该寄存器不能写 1，只能写 0。功能支持，当刹车事件有效或者死区时，PWM 的输出由对应管脚的 GPIO 设置决定，即此时是作为普通 GPIO 使用。
0	DTA_EN	死区控制使能 1 输出 A 死区控制有效 0 输出 A 死区控制无效

## 10.2.18.16 TIM1\_PCONRA (0xFF55)

Bit	7	6	5	4	3	2	1	0
Name	PA_INITVAL	CMPA_VAL[1:0]		PA_ENO	PA_FILTER_EN	CAPA_MODE[1:0]		CAPA_EN
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	PA_INITVAL	设置 TIM1_CHA 的输出： 1 TIM1_CHA 的初始值为 1 0 TIM1_CHA 的初始值为 0 TIMER1 关时设定有效，TIMER1 开时中间设定无效
6:5	CMPA_VAL[1:0]	配置 TIM1_CHA 比较输出值：



		00 计数值小于比较值为 1, 大于为 0 01 计数值大于比较值为 1, 小于为 0 10 比较值匹配, 输出取反前一状态 11 比较值匹配, 输出保持前一状态
4	PA_ENO	TIM1_CHA 输出控制: 1 TIM1_CHA 输出打开 0 TIM1_CHA 输出关闭
3	PA_FILTER_EN	TIM1_CHA 输入滤波使能 1 TIM1_CHA 输入数字滤波打开 0 TIM1_CHA 输入数字滤波关闭
2:1	CAPA_MODE[1:0]	TIM1_CHA 捕获模式选择: 00 不捕获 01 捕获上升沿 10 捕获下降沿 11 捕获上升沿与下降沿
0	CAPA_EN	TIM1_CHA 捕获模式使能: 1 TIM1_CHA 捕获模式开 0 TIM1_CHA 捕获模式关

## 10.2.18.17 TIM1\_PCONRB (0xFF56)

Bit	7	6	5	4	3	2	1	0
Name	PB_INITVAL	CMPB_VAL[1:0]		PB_ENO	PB_FILTER_EN	CAPB_MODE[1:0]		CAPB_EN
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	PB_INITVAL	设置 TIM1_CHB 的输出: 1 TIM1_CHB 的初始值为 1 0 TIM1_CHB 的初始值为 0 TIMER1 关时设定有效, TIMER1 开时中间设定无效
6:5	CMPB_VAL[1:0]	配置 TIM1_CHB 比较输出值: 00 计数值小于比较值为 1, 大于为 0 01 计数值大于比较值为 1, 小于为 0 10 比较值匹配, 输出取反前一状态 11 比较值匹配, 输出保持前一状态
4	PB_ENO	TIM1_CHB 输出控制: 1 TIM1_CHB 输出打开 0 TIM1_CHB 输出关闭
3	PB_FILTER_EN	TIM1_CHB 输入滤波使能: 1 TIM1_CHB 输入数字滤波打开 0 TIM1_CHB 输入数字滤波关闭
2:1	CAPB_MODE[1:0]	TIM1_CHB 捕获模式选择: 00 不捕获 01 捕获上升沿 10 捕获下降沿 11 捕获上升沿与下降沿
0	CAPB_EN	TIM1_CHB 捕获模式使能:



		1	TIM1_CHB 捕获模式开
		0	TIM1_CHB 捕获模式关

## 10.2.18.18 TIM1\_IE (0xC1)

Bit	7	6	5	4	3	2	1	0
Name	-	-	BRAKEB_IE	BRAKEA_IE	CMPB_IE	CMPA_IE	UD_IE	OV_IE
Reset	-	-	0	0	0	0	0	0
Type	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	N/A	保留位，读 0
5	BRAKEB_IE	TIM1_CHB 刹车中断使能： 1 TIM1_CHB 刹车中断使能开 0 TIM1_CHB 刹车中断使能关
4	BRAKEA_IE	TIM1_CHA 刹车中断使能： 1 TIM1_CHA 刹车中断使能开 0 TIM1_CHA 刹车中断使能关
3	CMPB_IE	TIM1_CHB 比较或者捕获中断使能： 1 TIM1_CHB 比较匹配或者捕获中断开 0 TIM1_CHB 比较匹配或者捕获中断关
2	CMPA_IE	TIM1_CHA 比较或者捕获中断使能： 1 TIM1_CHA 比较匹配或者捕获中断开 0 TIM1_CHA 比较匹配或者捕获中断关
1	UD_IE	下溢中断使能： 1 计数器下溢中断开 0 计数器下溢中断关
0	OV_IE	上溢中断使能： 1 计数器上溢中断开 0 计数器上溢中断关

## 10.2.18.19 TIM1\_SR (0xC2)

Bit	7	6	5	4	3	2	1	0
Name	-	-	BRAKEB_IF	BRAKEA_IF	CMB_IF	CMA_IF	UD_IF	OV_IF
Reset	-	-	0	0	0	0	0	0
Type	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	N/A	保留位，读 0
5	BRAKEB_IF	TIM1_CHB 刹车中断标志： 1 TIM1_CHB 输入发生刹车事件，刹车信号无效时 0 CHB 输入未发生刹车事件 写 1 清零该标志位
4	BRAKEA_IF	TIM1_CHA 刹车中断标志： 1 TIM1_CHA 输入发生刹车事件，刹车信号无效时



		0 TIM1_CHA 输入未发生刹车事件 写 1 清零该标志位
3	CMPB_IF	TIM1_CHB 比较或者捕获中断标志： 1 发生 TIM1_CHB 比较匹配或者捕获，写 1 清零 0 未发生 TIM1_CHB 比较匹配或者捕获 写 1 清零该标志位
2	CMPA_IF	TIM1_CHA 比较或者捕获中断使能： 1 TIM1_CHA 比较匹配或者捕获中断开 0 TIM1_CHA 比较匹配或者捕获中断关 写 1 清零该标志位
1	UD_IF	TIMER1 计数器下溢中断标志： 1 计数器发生下溢，写 1 清零 0 计数器未发生下溢 写 1 清零该标志位
0	OV_IF	TIMER1 计数器上溢中断标志： 1 计数器发生上溢，写 1 清零 0 计数器未发生上溢 写 1 清零该标志位

## 10.2.18.20 TIM2\_PR (0xCB)

Bit	7	6	5	4	3	2	1	0
Name	TIMKEY							
Reset	0							
Type	R/W							

Bit	Name	Function
7:0	TIMKEY	TIMER2 寄存器写保护密钥，写 0xCA 打开，打开后才能配置其它 TIMER2 的寄存器；密钥打开时读出值为 0xCA，否则读出值为 0

## 10.2.18.21 TIM2\_CR (0xC8)

Bit	7	6	5	4	3	2	1	0
Name	-	-	CAP_TIM1	SEL_SREG	DIR	MODE[1:0]		TIM2_EN
Reset	-	-	0	0	0	0	0	0
Type	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	N/A	保留位，读 0
5	CAP_TIM1	TIMER2 捕获 TIMER1 控制： 0 TIMER2 不捕获 TIMER1 1 TIMER2 捕获 TIMER1
4	SEL_SREG	影子寄存器控制： 0 ARR GCMAR GCMBR 读到影子寄存器的值或捕获值 1 ARR GCMAR GCMBR 读到当前设定的值
3	DIR	计数器计数方向： 0 向上计数 1 向下计数



2:1	MODE[1:0]	计数器计数模式： 00 锯齿波计数模式 01 三角波计数模式 10/11 保留
0	TIM2_EN	TIMER1 使能控制： 0 关闭 TIMER2 1 使能 TIMER2

## 10.2.18.22 TIM2\_FCONR (0xFF60)

Bit	7	6	5	4	3	2	1	0
Name	-	CLK_SEL[2:0]			PRE_DIV[3:0]			
Reset	-	0	0	0	0	0	0	0
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	N/A	保留位，读 0
6:4	CLK_SEL[2:0]	TIMER1 时钟源选择： 000 SYSCLK 001 看门狗时钟 32kHz 010 保留 011 保留 100 TIM2_CHA 上升沿 101 TIM2_CHB 上升沿 110 TIM2_CHA 下降沿 111 TIM2_CHB 下降沿
3:0	PRE_DIV[3:0]	TIMER2 预分频选择： 0~15 对应 1~16 分频 注意：修改 CLK_SEL 和 PRE_DIV 寄存器配置必须在 TIM2_EN 为 0 的时候进行。

## 10.2.18.23 TIM2\_CNTL (0xFF68)

Bit	7	6	5	4	3	2	1	0
Name	TIM2_CNTR[7:0]							
Reset	0x00							
Type	RO:0							

Bit	Name	Function
7	TIM2_CNTR[7:0]	计数器计数寄存器低 8 位

## 10.2.18.24 TIM2\_CNTH (0xFF69)

Bit	7	6	5	4	3	2	1	0
Name	TIM2_CNTR[15:8]							
Reset	0x00							
Type	RO:0							

Bit	Name	Function
-----	------	----------



7:0	TIM2_CNTR[15:8]	计数器计数寄存器高 8 位
-----	-----------------	---------------

## 10.2.18.25 TIM2\_ARRL (0xFF6A)

Bit	7	6	5	4	3	2	1	0
Name	TIM2_ARRL							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM2_ARRL	自动重载值寄存器低 8 位, 需先写高 8 位再写低 8 位

## 10.2.18.26 TIM2\_ARRH (0xFF6B)

Bit	7	6	5	4	3	2	1	0
Name	TIM2_ARRH							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM2_ARRH	自动重载值寄存器高 8 位, 需先写高 8 位再写低 8 位

## 10.2.18.27 TIM2\_GCMARL (0xFF6C)

Bit	7	6	5	4	3	2	1	0
Name	TIM2_GCMARL							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM2_GCMARL	计数模式下比较值, 捕获模式下 CHA 捕获值; GCMAR 低 8 位, 需先写高 8 位再写低 8 位

## 10.2.18.28 TIM2\_GCMARH (0xFF6D)

Bit	7	6	5	4	3	2	1	0
Name	TIM2_GCMARH							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM2_GCMARH	计数模式下比较值, 捕获模式下 CHA 捕获值 GCMAR 高 8 位, 需先写高 8 位再写低 8 位

## 10.2.18.29 TIM2\_GCMBRL (0xFF6E)

Bit	7	6	5	4	3	2	1	0
Name	TIM2_GCMBRL							
Reset	0x00							
Type	R/W							



Bit	Name	Function
7:0	TIM2_GCMBRL	计数模式下比较值，捕获模式下 CHB 捕获值 GCMBR 低 8 位，需先写高 8 位再写低 8 位

## 10.2.18.30 TIM2\_GCMBRH (0xFF6F)

Bit	7	6	5	4	3	2	1	0
Name	TIM2_GCMBRH							
Reset	0							
Type	R/W							

Bit	Name	Function
7:0	TIM2_GCMBRH	计数模式下比较值，捕获模式下 CHB 捕获值 GCMAR 高 8 位，需先写高 8 位再写低 8 位

## 10.2.18.31 TIM2\_VPERR (0xFF61)

Bit	7	6	5	4	3	2	1	0
Name	-	-	PCNTE[1:0]		-	PCNTS[2:0]		
Reset	-	-	0	0	-	0	0	0
Type	-	-	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Function
7:6	N/A	保留位，读 0
5:4	PCNTE[1:0]	周期间隔响应计数条件： 00 有效周期选择功能无效 01 锯齿波计数上、下溢点或三角波波峰作为计数条件 10 锯齿波计数上、下溢点或三角波波谷作为计数条件 (注意：由于 timer 使能后一段时间后才将周期值加载进 counter 寄存器中，这样会导致误判第一个波谷，可以选择软解规避) 11 锯齿波计数上、下溢点或三角波波谷、波峰作为计数条件 (注意：向下计数条件下不建议使用)
3	N/A	保留位，读 0
2:0	PCNTS[2:0]	周期间隔响应周期： 000 1 个周期响应一次 001 2 个周期响应一次 010 4 个周期响应一次 011 8 个周期响应一次 100 16 个周期响应一次 101 32 个周期响应一次 110 64 个周期响应一次 111 128 个周期响应一次

## 10.2.18.32 TIM2\_DTUA (0xFF62)

Bit	7	6	5	4	3	2	1	0
Name	TIM2_DTUA							
Reset	0x00							
Type	R/W							





Bit	Name	Function
7:0	TIM2_DTUA	TIMER2 死区时间设定值

## 10. 2. 18. 33 TIM2\_BRAKE (0xFF63)

Bit	7	6	5	4	3	2	1	0
Name	TIB_MOE	TIB_AOE	TIB_SEL	TIB_EN	TIA_MOE	TIA_AOE	TIA_SEL	TIA_EN
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	TIB_MOE	TIM2_CHB 主输出使能： 刹车事件有效时，会立即被同步清零。根据 AOE 的选择，通过软件置 1 或硬件自动置 1。 1 TIM2_CHB 主输出有效 0 TIM2_CHB 主输出关闭
6	TIB_AOE	自动输出使能： 1 有刹车事件产生时，MOE 可被软件和溢出事件置 1 0 有刹车事件产生时，MOE 只被软件置 1
5	TIB_SEL	选择 TIM2_CHB 刹车来源，由 {TIM2_DTR. TIB_ADCS, TIM2_BRAKE. TIB_SEL} 的值共同决定 {TIB_ADCS, TIB_SEL } 00 TIM2_CHB 刹车事件选择模拟比较器 1 输出 01 TIM2_CHB 刹车事件选择模拟比较器 0 输出 10 TIM2_CHB 刹车事件选择 ADC 比较输出 11 TIM2_CHB 刹车事件选择 P1.4 BRKIN 输入
4	TIB_EN	刹车功能控制： 1 TIM2_CHB 刹车有效 0 TIM2_CHB 刹车无效
3	TIA_MOE	TIM2_CHA 主输出使能： 刹车事件有效时会立即被同步清零。根据 AOE 的选择，通过软件置 1 或硬件自动置 1 1 TIM2_CHA 主输出有效 0 TIM2_CHA 主输出关闭
2	TIA_AOE	TIM2_CHA 自动输出使能： 1 有刹车事件产生时，MOE 可被软件和溢出事件置 1 0 有刹车事件产生时，MOE 只被软件置 1
1	TIA_SEL	选择 TIM2_CHA 刹车来源，由 {TIM2_DTR. TIA_ADCS, TIM2_BRAKE. TIA_SEL} 的值共同决定 {TIA_ADCS, TIA_SEL } 00 TIM2_CHA 刹车事件选择模拟比较器 1 输出 01 TIM2_CHA 刹车事件选择模拟比较器 0 输出 10 TIM2_CHA 刹车事件选择 ADC 比较输出 11 TIM2_CHA 刹车事件选择 P1.4 BRKIN 输入
0	TIA_EN	TIM2_CHA 刹车功能控制： 1 TIM2_CHA 刹车有效 0 TIM2_CHA 刹车无效



## 10.2.18.34 TIM2\_DTR (0xFF64)

Bit	7	6	5	4	3	2	1	0
Name	TIB_ADCS	TIA_ADCS	–	HW_CPWM	DTB_HO	DTB_EN	DTA_HO	DTA_EN
Reset	0	0	–	0	0	0	0	0
Type	R/W	R/W	–	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	TIB_ADCS	TIM2_CHB 刹车功能控制，见 TIM2_BRAKE 中 TIB_SEL 说明
6	TIA_ADCS	TIM2_CHA 刹车功能控制，见 TIM2_BREAK 中 TIA_SEL 说明
5	N/A	保留位，读 0
4	HW_CPWM	控制 GCMBR 互补模式： 0 硬件设定 GCMBR 互补 PWM 输出模式关 1 硬件设定 GCMBR 互补 PWM 输出模式开
3	DTB_HO	控制死区输出状态 1 保留 0 输出 B 死区置为 0 或 1（由 GPIO 输出决定） 备注：该寄存器不能写 1，只能写 0。功能支持，当刹车事件有效或者死区时，PWM 的输出由对应管脚的 GPIO 设置决定，即此时是作为普通 GPIO 使用。
2	DTB_EN	死区控制使能： 1 输出 B 死区控制有效 0 输出 B 死区控制无效
1	DTA_HO	控制死区输出状态 1 保留 0 输出 A 死区置为 0 或 1（由 GPIO 输出决定） 备注：该寄存器不能写 1，只能写 0。功能支持，当刹车事件有效或者死区时，PWM 的输出由对应管脚的 GPIO 设置决定，即此时是作为普通 GPIO 使用。
0	DTA_EN	死区控制使能： 1 输出 A 死区控制有效 0 输出 A 死区控制无效

## 10.2.18.35 TIM2\_PCONRA (0xFF65)

Bit	7	6	5	4	3	2	1	0
Name	PA_INITVAL	CMPA_VAL[1:0]		PA_ENO	PA_FILTER_EN	CAPA_MODE[1:0]		CAPA_EN
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	PA_INITVAL	设置 TIM2_CHA 的输出： 1 TIM2_CHA 的初始值为 1 0 TIM2_CHA 的初始值为 0 TIMER1 关时设定有效，TIMER1 开时中间设定无效
6:5	CMPA_VAL[1:0]	配置 TIM2_CHA 比较输出值： 00 计数值小于比较值为 1，大于为 0 01 计数值大于比较值为 1，小于为 0 10 比较值匹配，输出取反前一状态 11 比较值匹配，输出保持前一状态



4	PA_ENO	TIM2_CHA 输出控制： 1 TIM2_CHA 输出打开 0 TIM2_CHA 输出关闭
3	PA_FILTER_EN	TIM2_CHA 输入滤波使能： 1 TIM2_CHA 输入数字滤波打开 0 TIM2_CHA 输入数字滤波关闭
2:1	CAPA_MODE[1:0]	TIM2_CHA 捕获模式选择： 00 不捕获 01 捕获上升沿 10 捕获下降沿 11 捕获上升沿与下降沿
0	CAPA_EN	TIM2_CHA 捕获模式使能： 1 TIM2_CHA 捕获模式开 0 TIM2_CHA 捕获模式关

10. 2. 18. 36 TIM2\_PCONRB (0xFF66)

Bit	7	6	5	4	3	2	1	0
Name	PB_INITVAL	CMPB_VAL[1:0]	PB_ENO	PB_FILTER_EN	CAPB_MODE[1:0]	CAPB_EN		
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	PB_INITVAL	设置 TIM2_CHB 的输出： 1 TIM2_CHB 的初始值为 1 0 TIM2_CHB 的初始值为 0 TIMER2 关时设定有效，TIMER1 开时中间设定无效
6:5	CMPB_VAL[1:0]	配置 TIM2_CHB 比较输出值： 00 计数值小于比较值为 1，大于为 0 01 计数值大于比较值为 1，小于为 0 10 比较值匹配，输出取反前一状态 11 比较值匹配，输出保持前一状态
4	PB_ENO	TIM2_CHB 输出控制： 1 TIM2_CHB 输出打开 0 TIM2_CHB 输出关闭
3	PB_FILTER_EN	TIM2_CHB 输入滤波使能： 1 TIM2_CHB 输入数字滤波打开 0 TIM2_CHB 输入数字滤波关闭
2:1	CAPB_MODE[1:0]	TIM2_CHB 捕获模式选择： 00 不捕获 01 捕获上升沿 10 捕获下降沿 11 捕获上升沿与下降沿
0	CAPB_EN	TIM2_CHB 捕获模式使能： 1 TIM2_CHB 捕获模式开 0 TIM2_CHB 捕获模式关



## 10. 2. 18. 37 TIM2\_IE (0xC9)

Bit	7	6	5	4	3	2	1	0
Name	-	-	BRAKEB_IE	BRAKEA_IE	CMB_IE	CMA_IE	UD_IE	OV_IE
Reset	-	-	0	0	0	0	0	0
Type	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	N/A	保留位, 读 0
5	BRAKEB_IE	TIM2_CHB 刹车中断使能: 1 TIM2_CHB 刹车中断使能开 0 TIM2_CHB 刹车中断使能关
4	BRAKEA_IE	TIM2_CHA 刹车中断使能: 1 TIM2_CHA 刹车中断使能开 0 TIM2_CHA 刹车中断使能关
3	CMB_IE	TIM1_CHB 比较或者捕获中断使能: 1 TIM2_CHB 比较匹配或者捕获中断开 0 TIM2_CHB 比较匹配或者捕获中断关
2	CMA_IE	TIM2_CHA 比较或者捕获中断使能: 1 TIM2_CHA 比较匹配或者捕获中断开 0 TIM2_CHA 比较匹配或者捕获中断关
1	UD_IE	下溢中断使能: 1 计数器下溢中断开 0 计数器下溢中断关
0	OV_IE	上溢中断使能: 1 计数器上溢中断开 0 计数器上溢中断关

## 10. 2. 18. 38 TIM2\_SR (0xCA)

Bit	7	6	5	4	3	2	1	0
Name	-	-	BRAKEB_IF	BRAKEA_IF	CMB_IF	CMA_IF	UD_IF	OV_IF
Reset	-	-	0	0	0	0	0	0
Type	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	N/A	保留位, 读 0
5	BRAKEB_IF	TIM2_CHB 刹车中断标志: 1 TIM2_CHB 输入发生刹车事件, 刹车信号无效时 0 CHB 输入未发生刹车事件 写 1 清零该标志位
4	BRAKEA_IF	TIM1_CHA 刹车中断标志: 1 TIM2_CHA 输入发生刹车事件, 刹车信号无效时 0 TIM2_CHA 输入未发生刹车事件 写 1 清零该标志位
3	CMPB_IF	TIM2_CHB 比较或者捕获中断标志: 1 发生 TIM2_CHB 比较匹配或者捕获, 写 1 清零 0 未发生 TIM2_CHB 比较匹配或者捕获



		写 1 清零该标志位
2	CMPA_IF	TIM2_CHA 比较或者捕获中断使能： 1 TIM2_CHA 比较匹配或者捕获中断开 0 TIM2_CHA 比较匹配或者捕获中断关 写 1 清零该标志位
1	UD_IF	TIMER2 计数器下溢中断标志： 1 计数器发生下溢，写 1 清零 0 计数器未发生下溢 写 1 清零该标志位
0	OV_IF	TIMER2 计数器上溢中断标志： 1 计数器发生上溢，写 1 清零 0 计数器未发生上溢 写 1 清零该标志位

10.2.18.39 TIM3\_PR (0xCF)

Bit	7	6	5	4	3	2	1	0
Name	TIMKEY							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIMKEY	TIMER3 寄存器写保护密钥，写 0xCA 打开，打开后才能配置其它 TIMER3 的寄存器；密钥打开时读出值为 0xCA，否则读出值为 0

10.2.18.40 TIM3\_CR (0xCC)

Bit	7	6	5	4	3	2	1	0
Name	-	-	CAP_TIM1	SEL_SREG	DIR	MODE[1:0]		TIM3_EN
Reset	-	-	0	0	0	0		0
Type	-	-	R/W	R/W	R/W	R/W		R/W

Bit	Name	Function
7:6	N/A	保留位，读 0
5	CAP_TIM1	TIMER3 捕获 TIMER1 控制 0 TIMER3 不捕获 TIMER1 1 TIMER3 捕获 TIMER1
4	SEL_SREG	影子寄存器控制 0 ARR GCMAR GCMBR 读到影子寄存器的值或捕获值 1 ARR GCMAR GCMBR 读到当前设定的值
3	DIR	计数器计数方向 0 向上计数 1 向下计数
2:1	MODE[1:0]	计数器计数模式 00 锯齿波计数模式 01 三角波 A 计数模式 10/11 保留



0	TIM3_EN	TIMER3 使能控制 0 关闭 TIMER3 1 使能 TIMER3
---	---------	---

## 10.2.18.41 TIM3\_FCONR (0xFF70)

Bit	7	6	5	4	3	2	1	0
Name	-	CLK_SEL[2:0]			PRE_DIV[3:0]			
Reset	0x00							
Type	R/W							

Bit	Name	Function
7	N/A	保留位, 读 0
6:4	CLK_SEL[2:0]	TIMER3 时钟源选择 000 SCK0 001 看门狗时钟 32kHz 010 保留 011 保留 100 TIM3_CHA 上升沿 (打开捕获功能) 101 TIM3_CHB 上升沿 (打开捕获功能) 110 TIM3_CHA 下降沿 (打开捕获功能) 111 TIM3_CHB 下降沿 (打开捕获功能)
3:0	PRE_DIV[3:0]	TIMER3 预分频选择 0~15 对应 1~16 分频 备注: 修改 CLK_SEL 和 PRE_DIV 寄存器配置必须在 TIM3_EN 为 0 的时候进行。

## 10.2.18.42 TIM3\_CNTL (0xFF78)

Bit	7	6	5	4	3	2	1	0
Name	TIM3_CNTRL							
Reset	0x00							
Type	RO							

Bit	Name	Function
7:0	TIM3_CNTRL	计数器计数寄存器低 8 位

## 10.2.18.43 TIM3\_CNTH (0xFF79)

Bit	7	6	5	4	3	2	1	0
Name	TIM3_CNTRH							
Reset	0x00							
Type	RO							

Bit	Name	Function
7:0	TIM3_CNTRH	计数器计数寄存器高 8 位

## 10.2.18.44 TIM3\_ARRL (0xFF7A)

Bit	7	6	5	4	3	2	1	0
Name	TIM3_ARRL							



Reset	0x00
Type	R/W

Bit	Name	Function
7:0	TIM3_ARRL	自动重载值寄存器低 8 位, 需先写高 8 位再写低 8 位

## 10.2.18.45 TIM3\_ARRH (0xFF7B)

Bit	7	6	5	4	3	2	1	0
Name	TIM3_ARRH							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM3_ARRH	自动重载值寄存器高 8 位, 需先写高 8 位再写低 8 位

## 10.2.18.46 TIM3\_GCMARL (0xFF7C)

Bit	7	6	5	4	3	2	1	0
Name	TIM3_GCMARL							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM3_GCMARL	计数模式下比较值, 捕获模式下 CHA 捕获值 GCMAR 低 8 位, 需先写高 8 位再写低 8 位

## 10.2.18.47 TIM3\_GCMARH (0xFF7D)

Bit	7	6	5	4	3	2	1	0
Name	TIM3_GCMARH							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM3_GCMARH	计数模式下比较值, 捕获模式下 CHA 捕获值 GCMAR 高 8 位, 需先写高 8 位再写低 8 位

## 10.2.18.48 TIM3\_GCMBRL (0xFF7E)

Bit	7	6	5	4	3	2	1	0
Name	TIM3_GCMBRL							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM3_GCMBRL	计数模式下比较值, 捕获模式下 CHB 捕获值 GCMBR 低 8 位, 需先写高 8 位再写低 8 位



## 10.2.18.49 TIM3\_GCMBRH (0xFF7F)

Bit	7	6	5	4	3	2	1	0
Name	TIM3_GCMBRH							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM3_GCMBRH	计数模式下比较值，捕获模式下 CHB 捕获值 GCMBR 高 8 位，需先写高 8 位再写低 8 位

## 10.2.18.50 TIM3\_VPERR (0xFF71)

Bit	7	6	5	4	3	2	1	0
Name	-	-	PCNTE[1:0]		-	PCNTS[2:0]		
Reset	-	-	0	0	-	0	0	0
Type	-	-	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Function
5:4	PCNTE[1:0]	周期间隔响应计数条件： 00 有效周期选择功能无效 01 锯齿波计数上、下溢点或三角波波峰作为计数条件 10 锯齿波计数上、下溢点或三角波波谷作为计数条件 (注意：由于 timer 使能后一段时间后才将周期值加载进 counter 寄存器中，这样会导致误判第一个波谷，可以选择软解规避) 11 锯齿波计数上、下溢点或三角波波谷、波峰作为计数条件 (注意：向下计数条件下不建议使用)
2:0	PCNTS[2:0]	周期间隔响应周期 000 1 个周期响应一次 001 2 个周期响应一次 010 4 个周期响应一次 011 8 个周期响应一次 100 16 个周期响应一次 101 32 个周期响应一次 110 64 个周期响应一次 111 128 个周期响应一次

## 10.2.18.51 TIM3\_DTUA (0xFF72)

Bit	7	6	5	4	3	2	1	0
Name	TIM3_DTUA							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TIM3_DTUA	TIMER3 死区时间设定值

## 10.2.18.52 TIM3\_DTR (0xFF74)

Bit	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---





Name	TIB_ADCS	TIA_ADCS	-	HW_CPWM	DTB_HO	DTB_EN	DTA_HO	DTA_EN
Reset	0	0	-	0	0	0	0	0
Type	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	TIB_ADCS	TIM3_CHB 刹车功能控制，见 TIM3_BRAKE 中 TIB_SEL 说明
6	TIA_ADCS	TIM3_CHA 刹车功能控制，见 TIM3_BRAKE 中 TIA_SEL 说明
5	N/A	保留位，读 0
4	HW_CPWM	控制 GCMBR 互补模式 0 硬件设定 GCMBR 互补 PWM 输出模式关 1 硬件设定 GCMBR 互补 PWM 输出模式开
3	DTB_HO	控制死区输出状态 1 保留 0 输出 B 死区置为 0 或 1（由 GPIO 输出决定） 备注：该寄存器不能写 1，只能写 0。功能支持，当刹车事件有效或者死区时，PWM 的输出由对应管脚的 GPIO 设置决定，即此时是作为普通 GPIO 使用。
2	DTB_EN	死区控制使能 1 输出 B 死区控制有效 0 输出 B 死区控制无效
1	DTA_HO	控制死区输出状态 1 保留 0 输出 A 死区置为 0 或 1（由 GPIO 输出决定） 备注：该寄存器不能写 1，只能写 0。功能支持，当刹车事件有效或者死区时，PWM 的输出由对应管脚的 GPIO 设置决定，即此时是作为普通 GPIO 使用。
0	DTA_EN	死区控制使能 1 输出 A 死区控制有效 0 输出 A 死区控制无效

## 10.2.18.53 TIM3\_BRAKE (0xFF73)

Bit	7	6	5	4	3	2	1	0
Name	TIB_MOE	TIB_AOE	TIB_SEL	TIB_EN	TIA_MOE	TIA_AOE	TIA_SEL	TIA_EN
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	TIB_MOE	TIM3_CHB 主输出使能（PWM 输出时，需要使能该位） 刹车事件有效时，会立即被同步清零。根据 AOE 的选择，通过软件置 1 或硬件自动置 1。 1 TIM3_CHB 主输出有效 0 TIM3_CHB 主输出关闭
6	TIB_AOE	自动输出使能 1 有刹车事件产生时，MOE 可被软件和溢出事件置 1 0 有刹车事件产生时，MOE 只被软件置 1
5	TIB_SEL	选择 TIM3_CHB 刹车来源，由 {TIM3_DTR, TIB_ADCS, TIM3_BRAKE.TIB_SEL} 的值共同决定。 {TIB_ADCS, TIB_SEL } 00 TIM3_CHB 刹车事件选择模拟比较器 1 输出 01 TIM3_CHB 刹车事件选择模拟比较器 0 输出 10 TIM3_CHB 刹车事件选择 ADC 比较输出



		11	TIM3_CHB 刹车事件选择 P1.4 BRKIN 输入
4	TIB_EN	1	TIM3_CHB 刹车有效
		0	TIM3_CHB 刹车无效
3	TIA_MOE	TIM3_CHA 主输出使能 (PWM 输出时, 需要使能该位) 刹车事件有效时, 会立即被同步清零。根据 AOE 的选择, 通过软件置 1 或硬件自动置 1	
		1	TIM3_CHA 主输出有效
		0	TIM3_CHA 主输出关闭
2	TIA_AOE	TIM3_CHA 自动输出使能	
		1	有刹车事件产生时, MOE 可被软件和溢出事件置 1
		0	有刹车事件产生时, MOE 只被软件置 1
1	TIA_SEL	选择 TIM3_CHA 刹车来源, 由 {TIM3_DTR, TIA_ADCS, TIM3_BRAKE, TIA_SEL} 的值共同决定 {TIA_ADCS, TIA_SEL}	
		00	TIM3_CHA 刹车事件选择模拟比较器 1 输出
		01	TIM3_CHA 刹车事件选择模拟比较器 0 输出
		10	TIM3_CHA 刹车事件选择 ADC 比较输出
		11	TIM3_CHA 刹车事件选择 P1.4 BRKIN 输入
0	TIA_EN	TIM3_CHA 刹车功能控制	
		1	TIM3_CHA 刹车有效
		0	TIM3_CHA 刹车无效

## 10.2.18.54 TIM3\_PCONRA (0xFF75)

Bit	7	6	5	4	3	2	1	0
Name	PA_INITVAL	CMPA_VAL[1:0]		PA_ENO	PA_FILTER_EN	CAPA_MODE[1:0]		CAPA_EN
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	PA_INITVAL	设置 TIM3_CHA 的输出: 1 TIM3_CHA 的初始值为 1 0 TIM3_CHA 的初始值为 0 TIMER3 关时设定有效, TIMER3 开时中间设定无效
6:5	CMPA_VAL[1:0]	配置 TIM3_CHA 比较输出值: 00 计数值小于比较值为 1, 大于为 0 01 计数值大于比较值为 1, 小于为 0 10 比较值匹配, 输出取反前一状态 11 比较值匹配, 输出保持前一状态
4	PA_ENO	TIM3_CHA 输出控制: 1 TIM3_CHA 输出打开 0 TIM3_CHA 输出关闭
3	PA_FILTER_E	TIM3_CHA 输入滤波使能: 1 TIM3_CHA 输入数字滤波打开 0 TIM3_CHA 输入数字滤波关闭
2:1	CAPA_MODE[1:0]	TIM3_CHA 捕获模式选择: 00 不捕获 01 捕获上升沿 10 捕获下降沿



		11 捕获上升沿与下降沿
0	CAPA_EN	TIM3_CHA 捕获模式使能： 1 TIM3_CHA 捕获模式开 0 TIM3_CHA 捕获模式关

## 10.2.18.55 TIM3\_PCONRB (0xFF76)

Bit	7	6	5	4	3	2	1	0
Name	PB_INITVAL	CMPB_VAL[1:0]		PB_ENO	PB_FILTER_EN	CAPB_MODE[1:0]		CAPB_EN
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	PB_INITVAL	设置 TIM3_CHB 的输出： 1 TIM3_CHB 的初始值为 1 0 TIM3_CHB 的初始值为 0 TIMER3 关时设定有效，TIMER2 开时中间设定无效
6:5	CMPB_VAL[1:0]	配置 TIM3_CHB 比较输出值： 00 计数值小于比较值为 1，大于为 0 01 计数值大于比较值为 1，小于为 0 10 比较值匹配，输出取反前一状态 11 比较值匹配，输出保持前一状态
4	PB_ENO	TIM3_CHB 输出控制： 1 TIM3_CHB 输出打开 0 TIM3_CHB 输出关闭
3	PB_FILTER_E	TIM3_CHBB 输入滤波使能： 1 TIM3_CHB 输入数字滤波打开 0 TIM3_CHB 输入数字滤波关闭
2:1	CAPB_MODE[1:0]	TIM3_CHB 捕获模式选择： 00 不捕获 01 捕获上升沿 10 捕获下降沿 11 捕获上升沿与下降沿
0	CAPB_EN	TIM3_CHB 捕获模式使能： 1 TIM3_CHB 捕获模式开 0 TIM3_CHB 捕获模式关

## 10.2.18.56 TIM3\_IE (0xCD)

Bit	7	6	5	4	3	2	1	0
Name	–	–	BRAKEB_IE	BRAKEA_IE	CMPB_IE	CMPA_IE	UD_IE	OV_IE
Reset	–	–	0	0	0	0	0	0
Type	–	–	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	N/A	保留位，读 0
5	BRAKEB_IE	TIM3_CHB 刹车中断使能： 1 TIM3_CHB 刹车中断使能开



		0	TIM3_CHB 刹车中断使能关
4	BRAKEA_IE	TIM3_CHA 刹车中断使能： 1 TIM3_CHA 刹车中断使能开 0 TIM3_CHA 刹车中断使能关	
3	CMPB_IE	TIM3_CHB 比较或者捕获中断使能： 1 TIM3_CHB 比较匹配或者捕获中断开 0 TIM3_CHB 比较匹配或者捕获中断关	
2	CMPA_IE	TIM3_CHA 比较或者捕获中断使能： 1 TIM3_CHA 比较匹配或者捕获中断开 0 TIM3_CHA 比较匹配或者捕获中断关	
1	UD_IE	下溢中断使能： 1 计数器下溢中断开 0 计数器下溢中断关	
0	OV_IE	上溢中断使能： 1 计数器上溢中断开 0 计数器上溢中断关	

## 10.2.18.57 TIM3\_SR (0xCE)

Bit	7	6	5	4	3	2	1	0
Name	-		BRAKEB_IF	BRAKEA_IF	CMPB_IF	CMPA_IF	UD_IF	OV_IF
Reset	-		0	0	0	0	0	0
Type	-		R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	N/A	保留位，读 0
5	BRAKEB_IF	TIM3_CHB 刹车中断标志 1 TIM3_CHB 输入发生刹车事件，刹车信号无效时 0 CHB 输入未发生刹车事件 写 1 清零该标志位
4	BRAKEA_IF	TIM3_CHA 刹车中断标志 1 TIM3_CHA 输入发生刹车事件，刹车信号无效时 0 TIM3_CHA 输入未发生刹车事件 写 1 清零该标志位
3	CMPB_IF	TIM3_CHB 比较或者捕获中断标志 1 发生 TIM3_CHB 比较匹配或者捕获，写 1 清零 0 未发生 TIM3_CHB 比较匹配或者捕获 写 1 清零该标志位
2	CMPA_IF	TIM3_CHA 比较或者捕获中断标志 1 发生 TIM3_CHA 比较匹配或者捕获，写 1 清零 0 未发生 TIM3_CHA 比较匹配或者捕获 写 1 清零该标志位
1	UD_IF	TIMER3 计数器下溢中断标志 1 计数器发生下溢，写 1 清零 0 计数器未发生下溢 写 1 清零该标志位
0	OV_IF	TIMER3 计数器上溢中断标志



		1 计数器发生上溢，写 1 清零 0 计数器未发生上溢 写 1 清零该标志位
--	--	--

## 10.3 UART

### 10.3.1 概述

本芯片集成 2 个 UART 模块，UART 模块可以实现和外部设备异步通讯的功能，支持同时收发的全双工通信方式。UART 模块包含以下主要特性。

- 全双工
- 异步模式
- LSB 在前
- 集成波特率发生器
- 支持 9 位数据
- 支持奇偶校验，支持校验位作为数据位
- 帧错误检测
- 接收数据超限检测
- 支持发送传输完成中断、接收传输完成中断和帧错误中断（目前发生帧错误不会触发中断，只是状态寄存器帧错误标志位会跳起）

### 10.3.2 结构框图

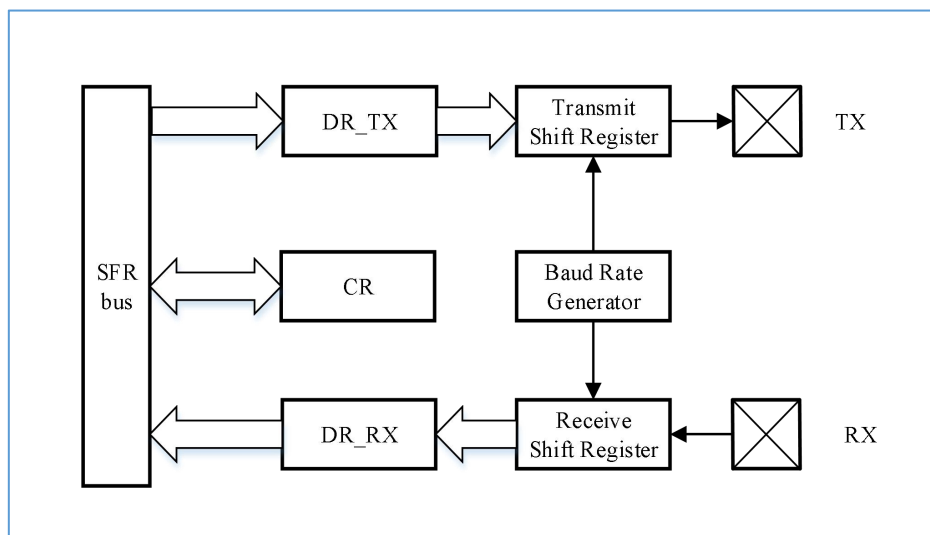


图 23 UART 结构框图

### 10.3.3 时钟发生器

时钟源来自系统时钟 SCK1、SCK2 和 SCK3 中的一个。

### 10.3.4 UART 发送

通过配置 UART 控制寄存器的 EN 位来使能 UART,同时配置控制寄存器的 T\_EN 位来将 UART 配置在发送模式。控制寄存器配置完成后,往 UART 数据寄存器中写值会启动一次 UART 发送操作,如果发送完成,UART 退回到空闲状态。一次发送完成后会置发送完成标志,该标志触发 UART 发送中断。发送完成标志可通过软件清除。

### 10.3.5 UART 接收

通过配置 UART 控制寄存器的 EN 位来使能 UART,同时配置控制寄存器的 R\_EN 位来将 UART 配置在接收模式。之后开始检测 RX 数据输入。如果检测到开始信号,UART 开始接收数据,如果成功检测到停止位,那么认为这一帧数据是有效的,将数据存储到 UART 数据寄存器,同时置位接收成功标志。如果接收到数据准备更新到 UART 数据寄存器时,接收标志也有效,则置接收超限标志。为了确保不触发错误的接收超限标志,用户必须在接收完成一帧数据后,清除接收标志。

### 10.3.6 UART 全双工

通过配置 UART 控制寄存器的 EN 位来使能 UART,同时配置控制寄存器的 R\_EN 位和 T\_EN 位来将 UART 配置成全双工模式。在全双工模式下,UART 既可以发送数据又可以接收数据,写 UART 数据寄存器会启动一次 UART 发送操作,一次发送完成后会置发送完成标志,该标志触发 UART 发送中断。全双工模式下,开始检测 RX 数据输入。如果检测到开始信号,UART 开始接收数据,如果成功检测到停止位,那么认为这一帧数据是有效的,将数据存储到 UART 数据寄存器,同时置位接收成功标志。当接收成功标志置为 1 时,再读取 UART 数据寄存器,就可将接收到的数据读出。如果未接收到数据时就读数据寄存器,则会读出全 0,所以在进行读 UART 数据寄存器时,需要检测 UART 状态寄存器,当检测到有接收完成标志后,再进行数据读取。

### 10.3.7 UART 半双工

UART 半双工的应用方式是使用 UART0/1CR.PSEL 寄存器,通过配置 PSEL 寄存器为 0 或 1 来切换 TXD 和 RXD 管脚的位置,得到 TXD 和 RXD 使用同一个 PIN 的目的。

### 10.3.8 校验位模式

配置 PAR\_EN=1 使能发送和接收时使用 9bit 模式,第 9bit 称为校验位,如何使用校验位支持两种模式:

- 当作奇偶校验位  
配置 PAR\_EN=1 并且 PAR\_DIR=0 使能该模式,这时候发送和接收按照正常
- 当作第 9bit 数据位  
配置 PAR\_EN=1 并且 PAR\_DIR=1 使能该模式,这时候发送的第 9bit 数据使用 PAR\_ODD 的值,接收的第 9bit 数据存储 ERR\_PAR 中。

### 10.3.9 波特率设置

波特率时钟可以来自 SCK1、SCK2、SCK3 三个时钟源,通过寄存器 UART\_CFG 配置。UART 使用时钟源的 4 分频来作为波特率时钟,接收和发送使用同样的波特率。

示例 1,使用 SCK3 配置 9600bps

$16000000 / (9600 * 4) = 416.6 \approx 4 * 104$

```
IMO_CR = 0xc9;    //SYS_CLK = 16M
PCLK_DIV12 = 0x3f; //SCK1 = Fsys/4
PCLK_DIV3 = 103;  //SCK3 = Fsck1/104
PCLK_CR = 0xf2;   //sck3 select clock source is sck1
```



## 10.3.10 与 UART 相关寄存器定义

名字	地址	读写	复位值	描述
UART0_CR	0x9D	读写	00000000	UART0 控制寄存器
UART0_DR	0x9C	读写	00000000	UART0 数据寄存器
UART0_SR	0x9E	读写	00000000	UART0 状态寄存器
UART0_CFG	0x9F	读写	00000000	UART0 配置寄存器
UART1_CR	0xBD	读写	00000000	UART1 控制寄存器
UART1_DR	0xBC	读写	00000000	UART1 数据寄存器
UART1_SR	0xBE	读写	00000000	UART1 状态寄存器
UART1_CFG	0xBF	读写	00000000	UART1 配置寄存器

## 10.3.10.1 UART0\_CR (0x9D)

Bit	7	6	5	4	3	2	1	0
Name	IE	R_EN	PAR_DIR	PSEL	PAR_ODD	PAR_EN	T_EN	EN
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	IE	0 发送完成或者接受满不产生中断 1 发送完成或者接受满产生中断
6	R_EN	0 不使能接收模式 1 使能接收模式
5	PAR_DIR	当 PAR_EN=1 时用来配置校验位模式 0 发送时根据 PAR_ODD 自动生成校验位, 接收时自动做校验检查 1 发送时使用 PAR_ODD 的值作为校验位, 接收到的校验位的值放在 ERR_PAR 寄存器中
4	PSEL	将 UART0 的 TX 和 RX 信号交换 0 P0.6 作为 TXD, P0.7 作为 RXD 1 P0.6 作为 RXD, P0.7 作为 TXD 注意: 使用单个引脚进行数据通信, 在切换引脚时, 建议先将 T_EN 与 R_EN 都关掉, 将 GPIO 都配置好后, 再打开对应的 R_EN/T_EN.
3	PAR_ODD	0 偶校验 1 奇校验 必须使能奇偶校验, 校验才会生效。
2	PAR_EN	0 关闭奇偶校验 1 使能奇偶校验 接收模式下, 收到的第 9 位数据数据位奇偶校验位; 发送模式下, 发送的第 9 位数据位前面 8 位数据的校验值。注: 奇偶校验自动生成, 不能配置。
1	T_EN	0 不使能发送模式 1 使能发送模式
0	EN	0 模块关闭 1 模块使能



## 10.3.10.2 UART0\_DR (0x9C)

Bit	7	6	5	4	3	2	1	0
Name	DATA							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DATA	发送模式该寄存器只能写。该寄存器只能在 UART0 使能之后才能写入。 接收模式下只能读，读取内容表示接收到的数据。

## 10.3.10.3 UART0\_SR (0x9E)

Bit	7	6	5	4	3	2	1	0
Name	RX_FULL	RX_ACTIVE	ERR_FRAME	ERR_PAR	OVERRUN	–	–	TX_COMPLETE
Reset	0	0	0	0	0	–	–	0
Type	R/W	R/W	R/W	R/W	R/W	–	–	R/W

Bit	Name	Function
7	RX_FULL	0 没有接收到数据 1 接收到了数据 读数据寄存器会清该标志位。发送模式下该位常为 0
6	RX_ACTIVE	0 没有接收数据 1 正在接收数据 发送模式下该位常为 0
5	ERR_FRAME	0 没有发生帧错误 1 发生帧错误 该位只有在接收模式下有效，接收数据时如果停止位收到高电平会触发帧错误。发送模式下该位常为 0。写 1 清零。
4	ERR_PAR	0 没有发生奇偶校验错误 1 发生奇偶校验错误 接收模式下，如果数据校验错误会置 1。发送模式下该位常为 0。写 1 清零该位。
3	OVERRUN	0 没有接收超限 1 接收超限 接收模式下，如果接收到了数据后又收到了数据会将该位置 1。 发送模式下常为 0。写 1 清零该标志位。
2:1	N/A	保留位，读 0
0	TX_COMPLETE	0 发送没有完成 1 发送完成 发送模式下，如果发送完成将该位置 1。接收模式下常为 0。写 1 清零该位。

## 10.3.10.4 UART0\_CFG (0x9F)

Bit	7	6	5	4	3	2	1	0
Name	–	–	–	–	–	–	CKSEL[1:0]	
Reset	–	–	–	–	–	–	0	
Type	–	–	–	–	–	–	R/W	

Bit	Name	Function
7:2	N/A	保留位，读 0





1:0	CKSEL[1:0]	选择 UART 时钟源 00 选择 SCK1 01 选择 SCK2 10 选择 SCK3 11 选择 SCK3
-----	------------	---

10.3.10.5 UART1\_DR (0xBC)

Bit	7	6	5	4	3	2	1	0
Name	DATA							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DATA	发送模式该寄存器只能写。该寄存器只能在 UART1 使能之后才能写入。 接收模式下只能读，读取内容表示接收到的数据。

10.3.10.6 UART1\_CR (0xBD)

Bit	7	6	5	4	3	2	1	0
Name	IE	R_EN	PAR_DIR	PSEL	PAR_ODD	PAR_EN	T_EN	EN
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	IE	0 发送完成或者接受满不产生中断 1 发送完成或者接受满产生中断
6	R_EN	0 不使能接收模式 1 使能接收模式
5	PAR_DIR	当 PAR_EN=1 时用来配置校验位模式 0 发送时根据 PAR_ODD 自动生成校验位，接收时自动做校验检查 1 发送时使用 PAR_ODD 的值作为校验位，接收到的校验位的值放在 ERR_PAR 寄存器中
4	PSEL	将 UART1 的 TX 和 RX 信号交换 0 P1.6 作为 TXD, P0.2 作为 RXD 1 P1.6 作为 RXD, P0.2 作为 TXD 注意：使用单个引脚进行数据通信，在切换引脚时，建议先将 T_EN 与 R_EN 都关掉，将 GPIO 都配置好后，再打开对应的 R_EN/T_EN.
3	PAR_ODD	0 偶校验 1 奇校验 必须使能奇偶校验，校验才会生效。
2	PAR_EN	0 关闭奇偶校验 1 使能奇偶校验 接收模式下，收到的第 9 位数据数据位奇偶校验位；发送模式下，发送的第 9 位数据位前面 8 位数据的校验值。注：奇偶校验自动生成，不能配置。
1	T_EN	0 不使能发送模式 1 使能发送模式
0	EN	0 模块关闭 1 模块使能



## 10.3.10.7 UART1\_SR (0xBE)

Bit	7	6	5	4	3	2	1	0
Name	RX_FULL	RX_ACTIVE	ERR_FRAME	ERR_PAR	OVERRUN	–	–	TX_COMPLETE
Reset	0	0	0	0	0	–	–	0
Type	R/W	R/W	R/W	R/W	R/W	–	–	R/W

Bit	Name	Function
7	RX_FULL	0 没有接收到数据 1 接收到了数据 读数据寄存器会清该标志位。发送模式下该位常为 0
6	RX_ACTIVE	0 没有接收数据 1 正在接收数据 发送模式下该位常为 0
5	ERR_FRAME	0 没有发生帧错误 1 发生帧错误 该位只有在接收模式下有效，接收数据时如果停止位收到高电平会触发帧错误。发送模式下该位常为 0。写 1 清零。
4	ERR_PAR	0 没有发生奇偶校验错误 1 发生奇偶校验错误 接收模式下，如果数据校验错误会置 1。发送模式下该位常为 0。写 1 清零该位。
3	OVERRUN	0 没有接收超限 1 接收超限 接收模式下，如果接收到了数据后又收到了数据会将该位置 1。 发送模式下，常为 0，写 1 清零该标志位。
2:1	N/A	保留位，读 0
0	TX_COMPLETE	0 发送没有完成 1 发送完成 发送模式下，如果发送完成将该位置 1。接收模式下常为 0。写 1 清零该位。

## 10.3.10.8 UART1\_CFG (0xBF)

Bit	7	6	5	4	3	2	1	0
Name	–	–	–	–	–	–	CKSEL[1:0]	
Reset	–	–	–	–	–	–	0	
Type	–	–	–	–	–	–	R/W	

Bit	Name	Function
7:2	N/A	保留位，读 0
1:0	CKSEL[1:0]	选择 UART1 时钟源 00 选择 SCK1 01 选择 SCK2 10 选择 SCK3 11 选择 SCK3

## 10.4 I2C

### 10.4.1 概述

I2C 是一种简单、双向的二进制同步串行总线，只需两根线即可在连接于总线上的器件之间传送信息。下图为 I2C 的架构图，MCU 通过总线访问 I2C 内部寄存器控制 I2C 的传输过程，I2C 通过两个双向的 GPIO 口与外部连接，发送或接收数据。

I2C 模块可以配置为主机或者从机模式或者主从模式。包含以下特性。

- 主机或者从机模式
- 多主机仲裁
- 速率 5Kbps、100Kbps、400Kbps
- 7 位从机地址
- 支持中断

### 10.4.2 结构框图

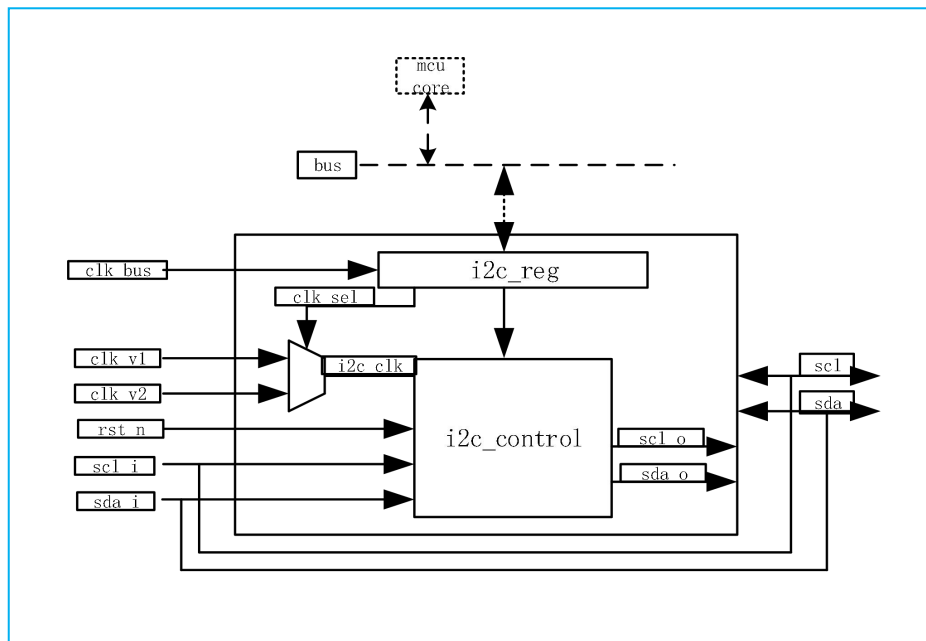


图 24 I2C 结构框图

### 10.4.3 应用描述

I2C 支持主从模式下的数据发送和接收。

#### 1. 基本数据传输方式

主器件产生传输用的时钟（SCL）信号，开始信号（START）和结束信号（STOP）。

数据（SDA）必须在时钟的低电平时改变，并在高电平时保持。

SCL 为高时，检测到 SDA 上有由高到低的跳变，为 START；

SCL 为高时，检测到 SDA 上有由低到高的跳变，为 STOP。

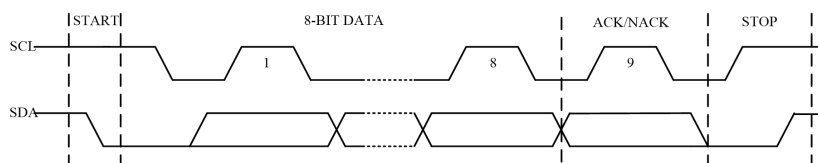


图 25 I2C 协议示意图

## 2. 从模式 (slave)

从模式下，会持续监听总线上是否有 START 信号。当监听到 START，会收到 8bit 的数据，其中包括 7bit 的 address 和 1bit 的 R/W 标志，从器件会根据收到的地址来确认是否响应主器件的读写请求。

如果地址正确，确认响应主器件的请求，从器件会根据 R/W 标志确认是传输数据还是接收数据，过程如图所示

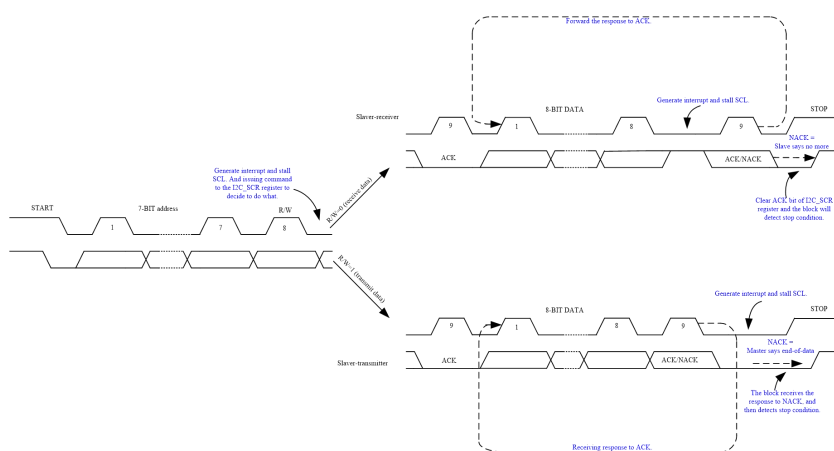


图 26 I2C 协议从机握手示意图

从器件成功发送 1byte 数据过程如下：

- 1) 确认寄存器都在初始状态。
  - 2) 打开从模式 (I2C\_CR)，处于监听状态。
- 收到 8-bit data (slave address) 后产生中断。
- 3) 将要发送的数据写入 I2C\_DR
  - 4) ACK bit 和 transmit bit 置 1 (I2C\_STAT)。
  - 5) Byte Complete bit 置 1 (I2C\_STAT)。
- 收到 8-bit data 和响应后产生中断。
- 6) 检查 LRB bit (I2C\_STAT)。
- 重复步骤 3~6，可以发送多 byte 数据

从器件成功接收 1byte 数据过程如下：

- 1) 确认寄存器都在初始状态。
  - 2) 打开从模式 (I2C\_CR)，处于监听状态。
- 收到 8-bit data (slave address) 后产生中断。
- 3) ACK bit 置 1, transmit bit 清 0 (I2C\_STAT)。
  - 4) Byte Complete bit 置 1 (I2C\_STAT)。
- 收到 8-bit data 后产生中断。。
- 5) ACK bit 清 0 (I2C\_STAT)。
- 重复步骤 3~4，可以接收多 byte 数据

### 3. 主模式

主模式下，发起一个传送请求前，主设备必须先判断总线是否处于空闲状态。当总线上有设备在传输数据时，总线忙状态位（Bus Busy）会一直置为1，直到检测到一个 STOP 信号，此时，当前设备获得总线使用权，启动一个读/写过程。

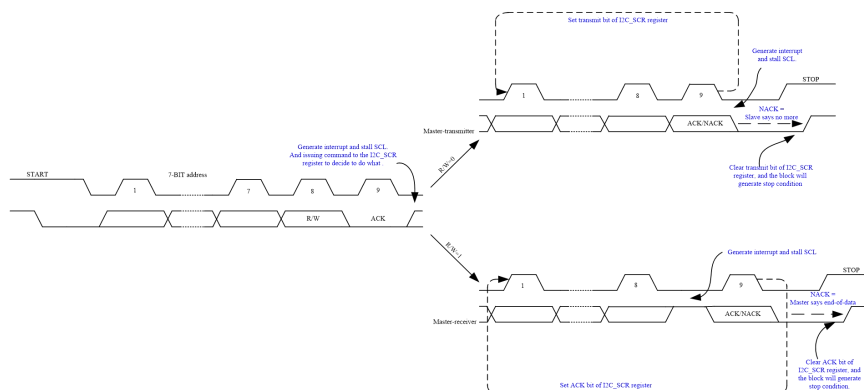


图 27 I2C 协议主机握手示意图

主器件成功发送 1byte 数据过程如下：

- 1) 确认寄存器都在初始状态。
- 2) 打开主模式 (I2C\_CR).
- 3) 将数据(slave address+W)写入 I2C\_DR.
- 4) Start Gen bit 置 1(I2C\_MCR).

主设备发送完 8bit 数据并收到 ACK，产生中断。

- 5) 将要发送数据写入 I2C\_DR.
  - 6) Transmit bit 置 1 (I2C\_STAT) .
- 主设备发送完 8bit 数据并收到 ACK，产生中断。
- 7) 发送完成，Transmit bit 清零 (I2C\_STAT register) .

重复步骤 5~6，可以发送多 byte 数据。

主器件成功接收 1byte 数据过程如下：

- 1) 确认寄存器都在初始状态。
  - 2) 打开主模式 (I2C\_CR).
  - 3) 将数据(slave address+W)写入 I2C\_DR.
  - 4) Start Gen bit 置 1(I2C\_MCR) .
- 主设备发送完 8bit 数据并收到 ACK，产生中断。
- 5) Transmit bit 清 0 (I2C\_STAT) .
- 主设备收到 8bit 数据，产生中断。
- 6) 如果需要接收更多数据，ACK bit 置 1，接收完成 ACK bit 置 0 .
- 重复步骤 5~6，能接收多 byte 数据。

#### 10.4.4 中断

I2C 提供 5 种类型的中断：

- 总线错误中断
- 停止中断
- NACK 中断
- 硬件地址匹配中断
- 传输完成中断



## 10.4.5 波特率设置

主机模式下，发送时钟来自时钟源的 17 分频。

## 10.4.6 与 I2C 相关寄存器定义

名字	地址	读写	复位值	描述
I2C_ADDR	0xA1	读写	01100110	I2C 从机地址寄存器
I2C_CR	0xA2	读写	00000001	I2C 控制寄存器
I2C_STAT	0xA3	读写	00000000	I2C 状态寄存器
I2C_DR	0xA4	读写	00000000	I2C 数据寄存器
I2C_MCR	0xA5	读写	00000000	I2C 主机控制寄存器

## 10.4.6.1 I2C\_ADDR (0xA1)

Bit	7	6	5	4	3	2	1	0
Name	HwAddrEn	Slave Address[6:0]						
Reset	0x66							
Type	R/W	R/W						

Bit	Name	Function
7	HwAddrEn	1 = 打开地址比较功能 0 = 关掉地址比较功能 只用于从模式下。 I2C_ADDR[6:0]为当前 I2C 设备号， HwAddrEn 为 1，收到请求后，会比较收到的地址是否与 Slave Address 一致，如果一致，则响应请求，不一致则不响应； HwAddrEn 为 0，会响应收到的所有请求。
6:0	Slave Address[6:0]	只用于从模式，当前设备的地址。

## 10.4.6.2 I2C\_CR (0xA2)

Bit	7	6	5	4	3	2	1	0
Name	I2C IE	-	Bus Error IE	Stop IE	-	Clk_sel	Enable Master	Enable Slave
Reset	0	-	0	0	-	0	0	1
Type	R/W	-	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Function
7	I2C IE	1 = 打开 I2C 全部中断 0 = 关闭 I2C 全部中断
6	N/A	保留位，读 0
5	Bus Error IE	1 = 打开 Bus Error 中断 0 = 关闭 Bus Error 中断.
4	Stop IE	1 = 打开结束中断 0 = 关闭结束中断



3	N/A	保留位, 读 0
2	Clk_sel	0 = SCK1 1 = SCK2
1:0	Enable Master or Slave	00 = 主模式关&从模式关 01 = 主模式关&从模式开 10 = 主模式开&从模式关 11 = 主模式开&从模式开

## 10.4.6.3 I2C\_STAT (0xA3)

Bit	7	6	5	4	3	2	1	0
Name	Bus Error	Lost Arb	Stop Status	ACK	Address	Transmit	LRB	Trans Complete
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	Bus Error(状态位)	只用于主模式, 数据传送过程中检测到总线上有开始或结束条件时置 1。 只能通过写 0 清除。 注意: 若发生了 Bus Error, 则需要配置成非主机模式或关掉 I2C。
6	Lost Arb(状态位)	只用于主模式, 失去对总线的控制权时置 1; 可以通过写 0 清除; 每次检测到开始信号都会自动清零。 注意: 若主机失去对总线控制, 则需要配置成非主机模式或关掉 I2C。
5	Stop Status(状态位)	检测到结束状态时置 1; 只能通过写 0 清除。
4	ACK(控制位)	1 = 发送 ack 0 = 不发送 ack (nack)
3	Address(状态位)	收到一个地址时置 1; 只能通过写 0 清除。
2	Transmit(状态位)	1 = 发送模式 0 = 接收模式
1	LRB(状态位)	1 = 最后收到的 bit 是 NACK 0 = 最后收到的 bit 是 ACK 写 0 清除或者检测到 START 信号清除。
0	Trans Complete(状态位)	单字节方式: 1: 接收完成 发送模式: 8bits 数据传送完成并收到响应 (ACK 或者 NACK)。 接收模式: 8bits 数据接收完成。 写 0 清除或者检测到 START 信号清除。

## 10.4.6.4 I2C\_DR (0xA4)

Bit	7	6	5	4	3	2	1	0
Name	Data							



Reset	0x00
Type	R/W

Bit	Name	Function
7:0	Data	主从模式接收，保存收到的数据，只读； 主模式产生开始信号前，需写入要发送到总线上的地址； 主从模式开始发送数据前，需写入要发送到客户端的数据。

#### 10.4.6.5 I2C\_MCR (0xA5)

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	Bus Busy	Master Mode	Restart Gen	Start Gen
Reset	-	-	-	-	0	0	0	0
Type	-	-	-	-	R	R	R/W	R/W

Bit	Name	Function
7:4	N/A	保留位，读 0
3	Bus Busy	检测到开始信号，状态置为 1； 检测到结束信号，状态置为 0。
2	Master Mode	产生开始信号，状态置为 1； 产生结束信号，状态置为 0。
1	Restart Gen	1 传送过程中收到响应为 NACK，重启传送过程，重新传送。
0	Start Gen	1 产生开始信号并发送地址到 i2c 总线上 传送完成后清零。

## 10.5 12-bit ADC

### 10.5.1 概述

本芯片内部集成了一个 12 位高精度，高转换速率的逐次逼近型模数转换器(SAR ADC)模块。具有以下特性：

- 12 位转换精度；
- 高达 180K SPS 的转换速度；
- 支持 8 路可选的单端输入通道：7 路外部输入通道，1 路片内 1/4VDD 电压输入通道；
- 支持 4 路可选的参考电压源；
- ADC 的有效电压输入范围：0~Vref；
- 软件可配置 ADC 的采样/转换时钟频率；
- 软件可配置 ADC 的采样时间；
- 可以配置 PWM、比较器输出或输入管脚边沿触发采样；
- 提供 ADC 转换结果比较器，比较结果可用于触发 PWM 故障刹车



### 10.5.2 结构框图

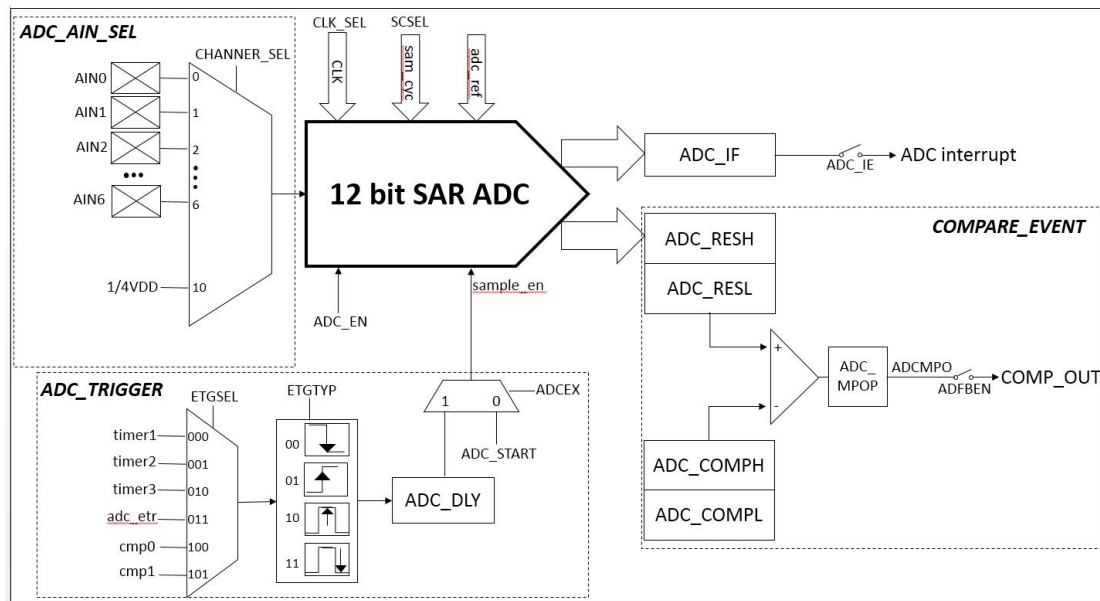


图 28 ADC 结构框图

### 10.5.3 ADC 转换时序

ADC 采样周期可配 4-128T，转换周期为 14T（1T 为一个机器周期）。

### 10.5.4 与 ADC 相关寄存器定义

名字	地址	读写	复位值	描述
ADC_CR0	0xE8	读写	00000000	ADC 转换控制寄存器 0
ADC_CR1	0xE9	读写	00000001	ADC 转换控制寄存器 1
ADC_CR2	0xEA	读写	00000011	ADC 转换控制寄存器 2
ADC_CHSEL	0xEB	读写	00001010	ADC 模拟量输入通道选择寄存器
ADC_CON	0xEC	读写	00000000	ADC 配置寄存器
ADC_DLY	0xED	读写	11110000	ADC 触发延迟配置寄存器
ADC_RES[RESL]	0xEE	读	00000000	ADC 转换结果低位寄存器
ADC_RES[RESH]	0xEF	读	00000000	ADC 转换结果高位寄存器
ADC_COMPL	0xFE	读写	00000000	ADC 比较值低 4 位
ADC_COMPH	0xFF	读写	00000000	ADC 比较值高 8 位

#### 10.5.4.1 ADC\_CR0 (0xE8)

Bit	7	6	5	4	3	2	1	0
Name	ADC_EN	—	ADC_START	ADC_IF	ADC_IE	ADCEX	CLK_SEL[1:0]	
Reset	0	—	0	0	0	0	0	0
Type	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	ADC_EN	ADC 使能位 0 = ADC 转换电路关闭



		1 = ADC 转换电路开启
6	N/A	保留位，读 0
5	ADC_START	ADC 软件触发采样控制位。ADC 使能后，该位写 1 开始 ADC 转换，转换完成后硬件自动将此位清零。 0 = 无影响。即使 ADC 已经开始转换工作，写 0 也不会停止 A/D 转换。 1 = 开始 ADC 转换，转换完成后硬件自动将此位清零。
4	ADC_IF	ADC 转换结束标志。当 ADC 完成一次转换后，硬件会自动将此位置 1，当中断使能有效时会向 CPU 发出中断请求。此标志位必须软件写 1 清零或复位清零。
3	ADC_IE	ADC 中断使能。 0 = 关闭 ADC 中断 1 = 使能 ADC 中断
2	ADCEX	该位决定启动 ADC 的触发条件 0 = 软件触发 1 = 硬件触发
1:0	CLK_SEL[1:0]	ADC 时钟选择： 00 = 系统时钟的 4 分频 01 = 系统时钟的 8 分频 10 = 系统时钟的 16 分频 11 = 系统时钟的 32 分频 注意：修改 CLK_SEL 寄存器配置必须在 ADC_EN 为 0 的时候进行。

## 10.5.4.2 ADC\_CR1 (0xE9)

Bit	7	6	5	4	3	2	1	0
Name	ETGSEL[2:0]			ETGTYP[1:0]		SCSEL[2:0]		
Reset	0			0		0	0	1
Type	R/W			R/W		R/W	R/W	R/W

Bit	Name	Function
7:5	ETGSEL[2:0]	外部触发源选择 当 ADCEX 为 1 时，该位选择外部触发 ADC 的来源 000 = timer1 触发 001 = timer2 触发 010 = timer3 触发 011 = adc_etr 触发 100 = 比较器 0 输出触发 101 = 比较器 1 输出触发 默认值：000 其他值：保留
4:3	ETGTYP[1:0]	外部触发信号类型选择 当 ADCEX 为 1 时该位决定响应外部触发的类型： 00 = 下降沿触发 01 = 上升沿触发 10 = 一个 PWM 周期的中点 11 = 一个 PWM 周期的终点 注：PWM 周期中点或终点仅适用于三角波模式的 PWM 输出
2:0	SCSEL[2:0]	ADC 采样时间周期选择寄存器： 000 = 4 个 ADC 时钟周期



		001 = 8 个 ADC 时钟周期 010 = 16 个 ADC 时钟周期 011 = 32 个 ADC 时钟周期 100 = 64 个 ADC 时钟周期 101 = 128 个 ADC 时钟周期 默认值: 001 其他值: 保留 片外很高的输入阻抗时, 增加采样时间, 提高转换精度。
--	--	--

## 10.5.4.3 ADC\_CR2 (0xEA)

Bit	7	6	5	4	3	2	1	0
Name	-		CTRL					
Reset	-		0	0	0	0	1	1
Type	-		R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	N/A	保留位, 读 0
5:0	CTRL[5:0]	<p>[5] 参考低噪声使能配置 0 = 正常工作模式 1 = 参考噪声减低 默认: 0</p> <p>[4] 参考测试模式 0 = 正常模式 1 = 测试模式 默认值: 0</p> <p>[3] 参考 buffer 增益选择 0 = 参考 buffer 输出是参考的 2 倍, 即 2.4V; 1 = 参考 buffer 输出是参考的 1 倍, 即 1.2V; 默认值: 0</p> <p>[2] 参考 buffer 输入选择 0 = 选择内部参考 Bandgap 电压; 1 = 选择外部参考电压 默认值: 0</p> <p>[1:0] ADC 参考电压选择 00 = 选择片外电压不通过 buffer, 直接做 ADC 参考电压; 10 = 选择 VDD, 做 ADC 参考电压; 11 = 选择 buffer 输出做 ADC 参考电压。 默认值: 11</p>

## 10.5.4.4 ADC\_CHSEL (0xEB)

Bit	7	6	5	4	3	2	1	0
Name	-				CHANNEL_SEL [3:0]			
Reset	-				1	0	1	0
Type	-				R/W	R/W	R/W	R/W



Bit	Name	Function
7:4	N/A	保留位，读 0
3:0	CHANNEL_SEL[3:0]	ADC 模拟量输入通道选择 0000 = 通道 0    0001 = 通道 1    0010 = 通道 2 0011 = 通道 3    0100 = 通道 4    0101 = 通道 5 0110 = 通道 6    0111 = 通道 7    1000 = 通道 8 1001 = 通道 9    1010 = 片内电源输入通道 默认值：1010    其他值：保留

## 10.5.4.5 ADC\_CON (0xEC)

Bit	7	6	5	4	3	2	1	0
Name	ADFBEN	ADCMPOP	ADCM PEN	ADCMPO	CLEAR	–	–	ADCDLY8
Reset	0	0	0	0	0	–	–	0
Type	R/W	R/W	R/W	R	W	–	–	R/W

Bit	Name	Function
7	ADFBEN	ADC 比较结果响应故障刹车使能寄存器 0 = 关闭 1 = ADC 触发故障刹车功能打开 默认值：0
6	ADCMPOP	ADC 比较器输出极性选择位 0 = 若 ADC 输出值大于或等于设定的比较值，则 ADCMPO 为 1 1 = 若 ADC 输出值小于设定的比较值，则 ADCMPO 为 1 默认值：0
5	ADCM PEN	ADC 结果比较使能位 0 = ADC 结果比较功能关闭 1 = ADC 结果比较功能打开
4	ADCMPO	ADC 比较结果输出位，每次 AD 转换结束都会更新输出
3	CLEAR	比较结果清零位，写 1 清
2:1	N/A	保留位，读 0
0	ADCDLY8	ADC 外部触发延时计数器数值的高 1 位

## 10.5.4.6 ADC\_DLY (0xED)

Bit	7	6	5	4	3	2	1	0
Name	ADCDLY[7:0]							
Reset	0xF0							
Type	R/W							

Bit	Name	Function
7:0	ADCDLY[7:0]	ADC 外部触发启动延迟计数器的低 8 位，默认值为 0xF0

## 10.5.4.7 ADC\_RESL (0xEE)

Bit	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---



Name	-	ADC_RESL[3:0]			
Reset	-	0	0	0	0
Type	-	R	R	R	R

Bit	Name	Function
7:4	N/A	保留位, 读 0
3:0	ADC_RESL[3:0]	ADC 转换结果低 4 位

#### 10.5.4.8 ADC\_RESB (0xEF)

Bit	7	6	5	4	3	2	1	0
Name	ADC_RESB[11:4]							
Reset	0x00							
Type	R							

Bit	Name	Function
7:0	ADC_RESB[11:4]	ADC 转换结果高 8 位

#### 10.5.4.9 ADC\_COMPL (0xFE)

Bit	7	6	5	4	3	2	1	0
Name	-				ADC_COMPL			
Reset	-				0			
Type	-				R/W			

Bit	Name	Function
7:4	N/A	保留位, 读 0
3:0	ADC_COMPL	ADC 比较值低 4 位

#### 10.5.4.10 ADC\_COMPB (0xFF)

Bit	7	6	5	4	3	2	1	0
Name	ADC_COMPB							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	ADC_COMPB	ADC 比较值高 8 位

## 10.6 SPI

### 10.6.1 概述

SPI 是一种常用的串行外设总线, 通过 4 线接口完成数据传输。SPI 模块支持以下特征:

- 全双工;
- 支持主机模式和从机模式;
- 支持数据传输高位优先和低位优先;

- 最大传输速率 8Mbps;
- 支持独立中断。

## 10.6.2 操作说明

### 10.6.2.1 SPI 主机

- (1) 配置 GPIO 为 SPI 模式;
- (2) 配置 SCK 时钟分频以及 SPI\_CR2 用于配置 SPI 输出时钟的频率;
- (3) 配置 SPI\_DR 寄存器准备用于发送的数据;
- (4) 配置 SPI\_ICR 寄存器用于选择中断上报事件;
- (5) 配置 SPI\_CR 寄存器用于配置 SPI 输出模式以及使能 SPI 外设。

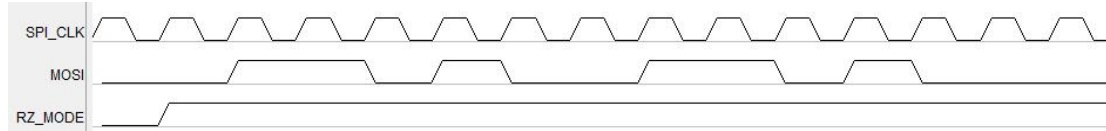
### 10.6.2.2 SPI 从机

- (6) 配置 GPIO 为 SPI 模式;
- (7) 配置 SPI\_DR 寄存器准备用于发送的数据;
- (8) 配置 SPI\_ICR 寄存器用于选择中断上报事件;
- (9) 配置 SPI\_CR 寄存器用于配置 SPI 输出模式以及使能 SPI 外设。
- (10) 注意事项: 在中断触发的 SPI\_DR 写操作事件中, 在系统时钟配置为 16M 条件下需要至少 4 $\mu$ s 才能第一次进行中断服务程序里的写 SPI\_DR 操作。若 FIFO 已发空而外部主机时钟继续发送, 本 SPI 从机将会循环发出最后 2Byte 数据。

### 10.6.2.3 RZ 码调制功能

- (1) 配置 GPIO 为 RZ 码模式 (RZ\_PEN 为 1), 除 MOSI 端口外, 其他 3 个 SPI 端口可作普通 IO 使用;
- (2) 配置 SCK 时钟分频以及 SPI\_CR2 用于配置 RZ 码输出时钟的频率;
- (3) 配置 SPI\_DR 寄存器准备用于发送的数据;
- (4) 配置 SPI\_ICR 寄存器用于选择中断上报事件;
- (5) 配置 SPI\_CR 寄存器将 SPI 配置为 RZ 输出模式以及使能外设。

注意事项: RZ 码只能在主机模式下使用; RZ 码的输出频率为配置 SPI 工作频率的 1/3, 下图为发送数据 A 的 RZ 码波形图。



### 10.6.3 与 SPI 相关寄存器定义

名字	地址	读写	复位值	描述
SPI_ICR	0xF1	读写	00000000	SPI 中断控制寄存器
SPI_DR	0xF2	读写	00000000	SPI 数据寄存器
SPI_CR	0xF3	读写	00010000	SPI 控制寄存器



SPI_STAT	0xF4	读写	00000000	SPI 状态寄存器
SPI_CR2	0xF5	读写	00000000	SPI 突发控制寄存器
SPI_WRADDR	0xF6	读	00000000	SPI 突发地址寄存器

## 10.6.3.1 SPI\_ICR (0xF1)

Bit	7	6	5	4	3	2	1	0
Name	–	TX_OVERFLOW_EN	TX_EMPTY_EN	TX_EMPTY_EN	RX_UNDERFLOW_EN	RX_OVERFLOW_EN	RX_AFULL_EN	RX_FULL_EN
Reset	–	0	0	0	0	0	0	0
Type	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	N/A	保留位，读 0
6	TX_OVERFLOW_EN	发送 FIFO 溢出中断使能 0 不上报中断 1 上报中断
5	TX_AEMPTY_EN	发送 FIFO 将空中断使能 0 不上报中断 1 上报中断
4	TX_EMPTY_EN	发送 FIFO 发空中断使能 0 不上报中断 1 上报中断
3	RX_UNDERFLOW_EN	接收 FIFO 下溢中断使能 0 不上报中断 1 上报中断
2	RX_OVERFLOW_EN	接收 FIFO 溢出中断使能 0 不上报中断 1 上报中断
1	RX_AFULL_EN	接收 FIFO 将满中断使能 0 不上报中断 1 上报中断
0	RX_FULL_EN	接收 FIFO 收满中断使能 0 不上报中断 1 上报中断

## 10.6.3.2 SPI\_DR (0xF2)

Bit	7	6	5	4	3	2	1	0
Name	DATA							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DATA	SPI 移位寄存器，写入数据为发送数据，读出数据为接收数据



## 10.6.3.3 SPI\_CR (0xF3)

Bit	7	6	5	4	3	2	1	0
Name	-	RZ_MODE	SS	MASTER	CPHA	CPOL	LSBF	ENABLE
Reset	-	0	0	1	0	0	0	0
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	N/A	保留位，读 0
6	RZ_MODE	RZ 码调制模式使能 0 禁用 RZ 调制模式 1 使能 RZ 调制模式（本功能仅支持处于 SPI 主机模式下工作）
5	SS	SPI 从机模式片选输入使能（作为从机需将此位配为 0） 0 = 从机模式片选信号默认为 0 1 = 使用 SPI 输入片选信号
4	MASTER	主机和从机模式选择 0 = 从机模式 1 = 主机模式
3	CPHA	时钟相位选择 0 = 时钟第一个跳变沿采样 1 = 时钟第二个跳变沿采样
2	CPOL	时钟极性选择 0 = SPI 总线空闲时时钟为低电平 1 = SPI 总线空闲时时钟为高电平
1	LSBF	低位优先控制位 0 = MSB First 1 = LSB First
0	ENABLE	SPI 使能 0 = SPI 功能关闭 1 = SPI 使能

## 10.6.3.4 SPI\_STAT (0xF4)

Bit	7	6	5	4	3	2	1	0
Name	-	TX_OVERFLOW	TX_AMPTY	TX_EMPTY	RX_UNDERFLOW	RX_OVERFLOW	RX_AFULL	RX_FULL
Reset	-	0	0	0	0	0	0	0
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	N/A	保留值，读 0
6	TX_OVERFLOW	发送 FIFO 写溢出标志 0 发送 FIFO 未写溢出 1 发送 FIFO 写溢出，通过对该位写 1 可清零
5	TX_AEMPTY	发送 FIFO 将空标志 0 发送 FIFO 中数据多于 2B 1 发送 FIFO 中数据小于等于 2B，通过写 SPI_DR 可将该位清零
4	TX_EMPTY	发送 FIFO 发空标志





		0 发送 FIFO 未发空 1 发送 FIFO 发空，通过写 SPI_DR 可将该位清零
3	RX_UNDERFLOW	接收 FIFO 读下溢标志 0 接收 FIFO 未读下溢 1 接收 FIFO 读下溢，通过对该位写 1 可清零
2	RX_OVERFLOW	接收 FIFO 写溢出标志 0 接收 FIFO 未写溢出 1 接收 FIFO 写溢出，通过对该位写 1 可清零
1	RX_AFULL	接收 FIFO 将满标志 0 接收 FIFO 中数据少于 2B 1 接收 FIFO 中数据大于等于 2B，通过读 SPI_DR 可将该位清零
0	RX_FULL	接收 FIFO 满标志 0 接收缓存不满 1 接收缓存满；对 SPI_DR 进行读操作可以将该位清零

## 10.6.3.5 SPI\_CR2 (0xF5)

Bit	7	6	5	4	3	2	1	0
Name	CLKIN[1:0]		-					
Reset	0		-					
Type	R/W		-					

Bit	Name	Function
7:6	CLKIN[1:0]	选择 SPI 工作时钟 00 = 使用 SCK1 作为 SPI 工作时钟 01 = 使用 SCK2 作为 SPI 工作时钟 10 = 使用 SCK3 作为 SPI 工作时钟 11 = 使用 CLK_SYS 作为 SPI 工作时钟
5:0	N/A	保留位，读 0

## 10.6.3.6 SPI\_WRADDR (0xF6)

Bit	7	6	5	4	3	2	1	0
Name	-						WRTADDR[1:0]	
Reset	-						0	
Type	-						RO	

Bit	Name	Function
7:2	N/A	保留位，读 0
1:0	WRTADDR[1:0]	当前缓存的写地址

## 10.7 模拟比较器

### 10.7.1 概述

模拟电压比较器 CMP 用于比较两个输入模拟电压的大小，并根据比较结果输出高/低电平。当“+”输入端电压高于“-”输入端电压时，电压比较器输出为高电平；当“+”输入端电压低于“-”输入端电压时，电压比较器输出为低电平。本芯片内部集成的模拟电压比较器 CMP 具有以下特性：

- 支持电压比较功能；
- 支持内部 20 阶 VBG 分压作为输入；
- 支持外部输入端口；
- 支持三种软件可配置的中断触发方式：上升沿触发、下降沿触发、上下沿触发；
- 电压比较器的输出可以作为 Timer1、Timer2 以及 Timer3 的刹车输入；
- 提供软件可配置的滤波时间以增强芯片的抗干扰能力。

### 10.7.2 结构框图

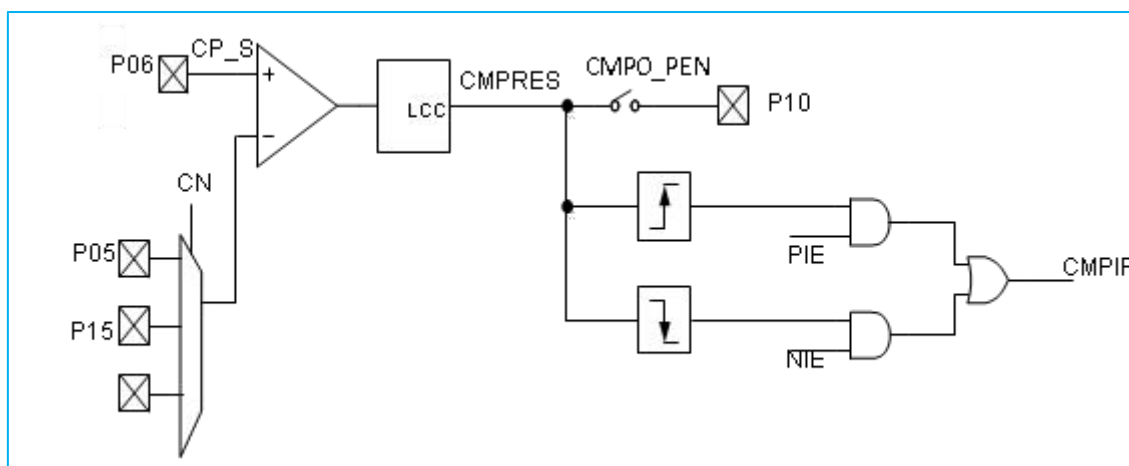


图 29 比较器 0 电路结构框图

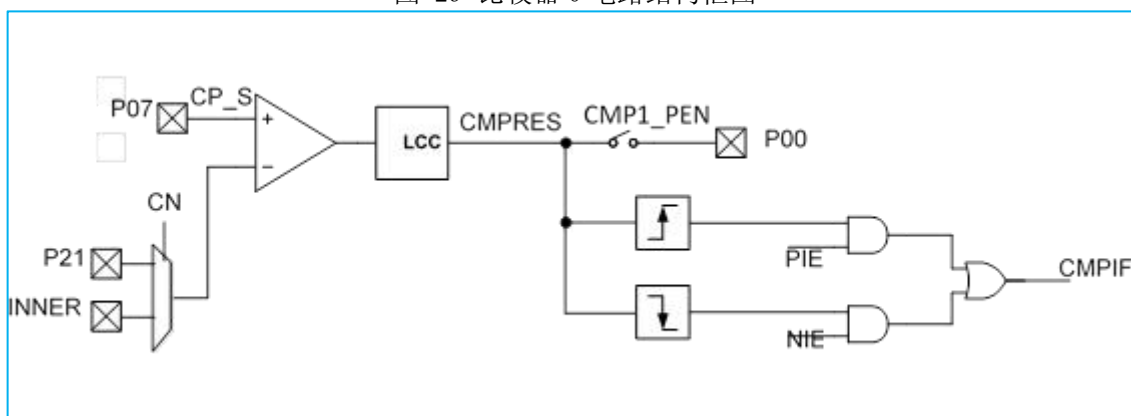


图 30 比较器 1 电路结构框图

### 10.7.3 比较器时钟和滤波功能

模拟比较器输出通过 SCK0 同步到内部数字系统，数字滤波时钟采用 SCK0 作为输入，如果要使用比较器要打开 SCK0。

数字滤波工作即为数字信号去抖动功能。当比较器输出变化 LCDTY 个时钟周期稳定后才认为数据变化是有效的。数字滤波时钟来自 SCK0，所以如果要使用数字滤波功能，要使能 SCK0 时钟。

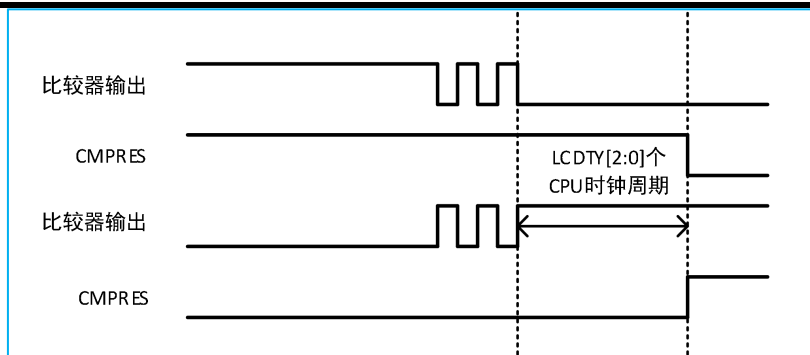


图 31 比较器数字滤波功能

备注：模拟比较器的输出到数字滤波电路之间有两个 SCK0 时钟周期延时。

### 10.7.4 与比较器相关寄存器定义

名字	地址	读写	复位值	描述
ACO_CR1	0xE1	读写	00000000	模拟比较器 0 控制寄存器 1
ACO_CR2	0xE2	读写	00000000	模拟比较器 0 控制寄存器 2
AC1_CR1	0xE4	读写	00000000	模拟比较器 1 控制寄存器 1
AC1_CR2	0xE5	读写	00000000	模拟比较器 1 控制寄存器 2
DAC_BUFCCR1	0xFF97	读写	00000000	OFFSET 修调配置寄存器 1
DAC_BUFCCR2	0xFF9E	读写	00000000	OFFSET 修调配置寄存器 2
BUF_OSN	0xFF9C	读写	00000000	比较器 BUFFER N 端修调配置寄存器
BUF_OSP	0xFF9D	读写	00000000	比较器 BUFFER P 端和 LVD BUFFER 修调配置寄存器

#### 10.7.4.1 ACO\_CR1 (0xE1)

Bit	7	6	5	4	3	2	1	0
Name	CMPEN	CMPIF	PIE	NIE	CN[1:0]		-	CMPRES
Reset	0	0	0	0	0	0	-	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	-	R

Bit	Name	Function
7	CMPEN	比较器开启控制位 0 关闭比较功能 1 使能比较功能 备注：当 CMPEN=1 即使能的时候，如果 CN=0b11，则比较器输出结果不可预测
6	CMPIF	比较器中断标志位。当 PIE 或者 NIE 被使能后，若产生相应的中断信号，硬件自动将 CMPIF 置 1，并向 CPU 提出中断请求。此标志位须由用户软件写 0 清零。（备注：如果没有使能比较器中断时，硬件不会设置此中断标志，即使使用查询方式访问比较器时，不能查询此中断标志）
5	PIE	比较器上升沿中断使能位 0 禁止比较器上升沿中断 1 使能比较器上升沿中断。
4	NIE	比较器下降沿中断使能位 0 禁止比较器下降沿中断 1 使能比较器下降沿中断



3:2	CN[1:0]	比较器的负输入端选择位 00 选择外部端口 P0.5 作为比较器的负极输入 01 选择外部端口 P1.5 作为比较器的负极输入 10 选择内部参考来自 VDD 分压或者 1.2V 的分压 11 比较器的负输入通道都关闭
1	N/A	保留位, 读 0
0	CMPRES	比较器的比较结果, 此位为只读位 0 表示 CMP+的电平低于 CMP-的电平 1 表示 CMP+的电平高于 CMP-的电平 CMPRES 是数字滤波后的输出信号, 而不是比较器的直接输出结果。

## 10.7.4.2 ACO\_CR2 (0xE2)

Bit	7	6	5	4	3	2	1	0
Name		-		CMPSEL	-	LCDTY[2:0]		
Reset		-		0	-	0	0	0
Type		-		R/W	-	R/W	R/W	R/W

Bit	Name	Function
7:5	N/A	保留位, 读 0
4	CMPSEL	比较器输出结果选择 0 表示选择输出滤波后的 CMP 结果 1 表示选择输出滤波前的 CMP 结果 CMPRES 是数字滤波后的输出信号, 而不是比较器的直接输出结果。
3	N/A	保留位, 读 0
2:0	LCDTY[2:0]	数字滤波功能。当比较器结果发生上升沿或者下降沿变化时, 比较器侦测变化后的信号必须维持 LCDTY 所设置的 CPU 时钟数不发生变化, 才认为数据变化是有效的; 否则若 LCDTY 设置为 0 时表示关闭数字滤波功能。

## 10.7.4.3 DAC\_BUFCCR1 (0xFF97)

Bit	7	6	5	4	3	2	1	0
SLPIE	-	EN_BUF	EN_BUF_DIV	OPT0				
Reset	-	DAEN	0	0				
Type	-	R/W	R/W	R/W				

Bit	Name	Function
7	N/A	保留位, 读 0
6	EN_BUF	BUFFER 使能端口, 高电平有效
5	EN_BUF_DIV	使能 BUFFER 的分压电阻, 高有效
4:0	OPT0[4:0]	分压档位选择 0 选择地 1~20 选择 OPT0/20 VDD 或 V1P2

## 10.7.4.4 DAC\_BUFCCR2 (0xFF9E)

Bit	7	6	5	4	3	2	1	0
Name	-	EXC_V1P2_VDD	-					
Reset	-	0	-					



Type	–	R/W	–
Bit	Name	Function	
7	N/A	保留位, 读 0	
6	EXC_V1P2_VDD	电阻分压源头切换控制 0 选择 1.2V 分压 1 选择 VDD 分压 当为 1 时, 必须把 BUFFER 关闭, 也就是 EN_BUF 为 0	
5:0	N/A	保留位, 读 0	

## 10.7.4.5 AC1\_CR1 (0xE4)

Bit	7	6	5	4	3	2	1	0
Name	COMPEN	CMPIF	PIE	NIE	–	CN	–	COMPRES
Reset	0	0	0	0	–	0	–	0
Type	R/W	R/W	R/W	R/W	–	R/W	–	R

Bit	Name	Function
7	COMPEN	0 关闭比较功能 1 使能比较功能
6	CMPIF	比较器中断标志位。当 PIE 或者 NIE 被使能后, 若产生相应的中断信号, 硬件自动将 CMPIF 置 1, 并向 CPU 提出中断请求。次标志位必须用户软件清零。(备注: 如果没有使能比较器中断时, 硬件不会设置此中断标志, 即使使用查询方式访问比较器时, 不能查询此中断标志)
5	PIE	比较器上升沿中断使能位: 0 禁止比较器上升沿中断 1 使能比较器上升沿中断。
4	NIE	比较器下降沿中断使能位: 0 禁止比较器下降沿中断 1 使能比较器下降沿中断
3	N/A	保留位, 读 0
2	CN	比较器的负输入端选择位 0 选择 C1N0, 即 P2.1 1 选择内部 buffer 输入
1	N/A	保留位, 读 0
0	COMPRES	比较器的比较结果, 此位为只读位 0 表示 CMP+的电平低于 CMP-的电平 1 表示 CMP+的电平高于 CMP-的电平 COMPRES 是数字滤波后的输出信号, 而不是比较器的直接输出结果。

## 10.7.4.6 AC1\_CR2 (0xE5)

Bit	7	6	5	4	3	2	1	0
Name	–			COMPSEL	–	LCDTY[2:0]		
Reset	–			0	–	0	0	0
Type	–			R/W	–	R/W	R/W	R/W
Bit	Name	Function						
7:5	N/A	保留位, 读 0						



4	CMPSEL	比较器输出结果选择 0 表示选择输出滤波后的 CMP 结果 1 表示选择输出滤波前的 CMP 结果 CMPRES 是数字滤波后的输出信号，而不是比较器的直接输出结果。
3	N/A	保留位，读 0
2:0	LCDTY[2:0]	数字滤波功能。当比较器结果发生上升沿或者下降沿变化时，比较器侦测变化后的信号必须维持 LCDTY 所设置的 CPU 时钟数不发生变化，才认为数据变化是有效的；否则若 LCDTY 设置为 0 时表示关闭数字滤波功能。

## 10.7.4.7 BUF\_OSN (0XFF9C)

Bit	7	6	5	4	3	2	1	0
Name	EN_CMP_BUFTS	BUF_OSLCS_N	BUF_OSN					
Reset	0	0	0					
Type	R/W	R/W	R/W					

Bit	Name	Function
7	EN_CMP_BUFTS	比较器参考 buffer 测试使能， 1: 测试模式 0: 正常模式
6	BUF_OSLCS_N	比较器参考 buffer 的 N 端最低修调位
5:0	BUF_OSN	比较器参考 buffer 的 N 端 6 至 1 修调位

## 10.7.4.8 BUF\_OSP (0XFF9D)

Bit	7	6	5	4	3	2	1	0
Name	LVD_REF_SEL	BUF_OSLCS_P	BUF_OSP					
Reset	0	0	0					
Type	R/W	R/W	R/W					

Bit	Name	Function
7	LVD_REF_SEL	LVD 比较器参考电压来源配置 0: 选择 BG 电压（没有修调） 1: 选择采用比较器里经过参考 BUFFER 的电压（有修调） 注：LVD 和 BOR 的比较器参考电压同源，都可以选择 BG 电压或经过参考 BUFFER 的电压，由 LVD_REF_SEL 配置决定；当选择采用经过参考 BUFFER 的电压时，需要配置 DAC_BUFCCR1/2 寄存器，将 buffer 及其分压电阻使能，分压来源 1.2V，并选择分压电阻满量程，否则 LVD/BOR 配置无效。
6	BUF_OSLCS_P	比较器参考 buffer 的 P 端最低修调位
5:0	BUF_OSP	比较器参考 buffer 的 P 端 6 至 1 修调位

## 11. 省电模式和看门狗

### 11.1 省电模式

本芯片有两种低功耗模式来优化设备功耗：

- 睡眠模式
- 深度睡眠模式

这两种模式下程序都停止运行。

外设	睡眠模式	深度睡眠模式
CPU	停止	停止
RAM	保持	保持
睡眠定时器	运行	运行
看门狗	运行	运行
定时器 0~3	运行	停止
ADC	运行	停止
UART0/1	运行	停止
I2C	运行	停止
内部 16MHz 振荡器	运行	停止
内部 32KHz 振荡器	运行	运行
I/O 口	保持	保持
其他外设	运行	停止
唤醒条件	引脚复位，看门狗复位，所有中断	看门狗复位，引脚中断，引脚中断，睡眠定时器中断

### 11.2 睡眠模式

写 SCR 寄存器 SLEEP=1 且 SLEEPDEEP=0 进入到睡眠模式。该模式下，内部 16MHz 晶振保持工作。同时继续给外设提供时钟，但是 CPU 时钟停止。该模式可以通过复位和中断唤醒。如果使用复位唤醒，那么整个系统会复位而初始化。

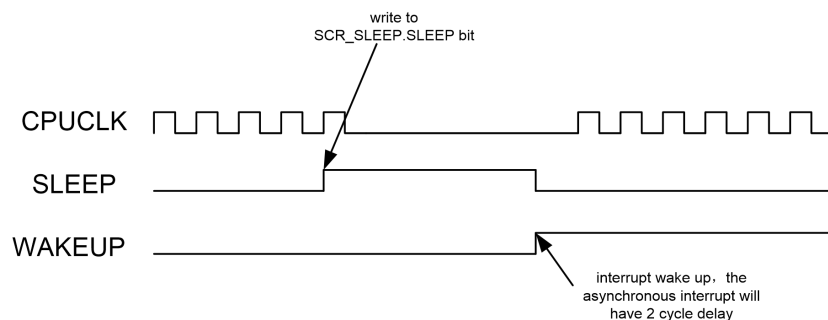


图 32 睡眠模式休眠和唤醒时序图

### 11.2.1 深度睡眠模式

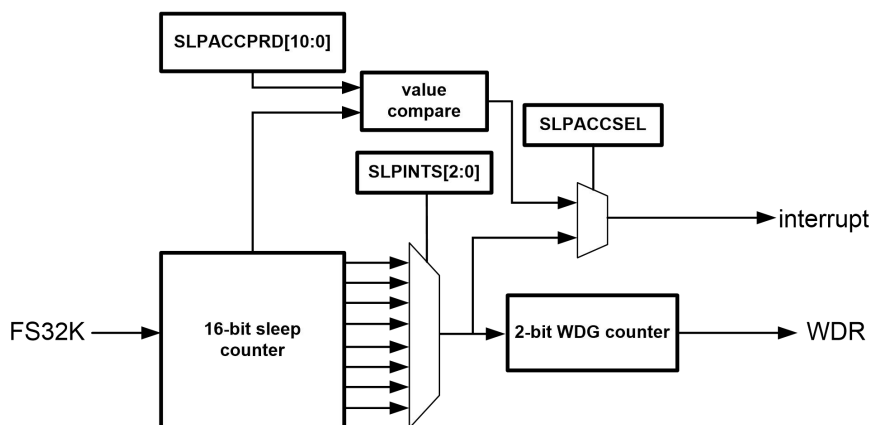
深度睡眠模式通过写 SCR 的 SLEEP=1 且 SLEEPDEEP=1 进入。该模式下，16MHz 主振荡器停止工作，32KHz 低功耗振荡器继续工作。系统时钟和外设时钟停止，但是睡眠定时器和看门狗继续工作。

### 11.2.2 深度休眠模式唤醒

深度睡眠模式可以通过复位和中断唤醒。复位重新初始化所有的控制寄存器，所以重新工作。振荡器的重新工作需要一定时间的延时。下面的图描述了深度休眠唤醒的时序。

## 11.3 看门狗

看门狗定时器由 16 位睡眠定时器和 2 位看门狗定时器组成，如果看门狗使能且计数到 3 并溢出的话那么会触发看门狗复位。看门狗复位如果被触发会保持 1 个 32K 时钟周期。看门狗计数器可以通过写一个特殊寄存器 WDCLR 来清零。睡眠计数器也可以通过写 WDCLR 来清零。



**注意：**上电后看门狗默认开启，如果不需要使用需要配置关闭。

图 33 看门狗定时器

## 11.4 睡眠定时器中断

看门狗内部实现了一个 16 位的睡眠向上计数器，该定时器用作看门狗的预分频同时也可以作用定时功能。可以产生中断，中断使能可以控制。该定时器有两种用法，第一种 SLPACCSSEL 等于 0 时，通过 SLPINTS 选择固定的定时周期，第二种 SLPACCSSEL 等于 1 时，选择对于的溢出值来产生中断，溢出值通过 SLPACCPRD[10: 0] 来配置。（备注：当选用第二种配置方法时，实际溢出值为配置值加 1 个 32K 时钟周期）

## 11.5 与省电模式和看门狗相关寄存器定义

名字	地址	读写	复位值	描述
SLPTIM_CR	0x88	读写	00100100	睡眠计数器控制寄存器
SLPTIM_SR	0x89	读写	00000000	睡眠计数状态
SLPTIM_CLR	0x8A	写	00000000	看门狗清除寄存器





SLPTIM_WDT	0x8B	读写	00000000	看门狗计数器状态
SLPTIM_CNTL	0x8C	只读	00000000	睡眠计数器计数值
SLPTIM_CNTH	0x8D	只读	00000000	睡眠计数器计数值
SLPTIM_PRDRL	0x8E	读写	00000000	睡眠计数器预分频
SLPTIM_PRDRH	0x8F	读写	00000000	睡眠计数器预分频

### 11.5.1 SLPTIM\_CR (0x88)

Bit	7	6	5	4	3	2	1	0
SLPIE	SLPIE	-	WDTEN	RSVO	SLEEPDIS	SLPINTS[2:0]		
Reset	0	-	1	0	0	1	0	0
Type	R/W	-	R/W	WO	R/W	R/W	R/W	R/W

Bit	Name	Function
7	SLPIE	0 = 睡眠定时器中断禁止 1 = 睡眠定时器中断使能
6	N/A	保留位，读 0
5	WDTEN	0 = 看门狗定时器禁止 1 = 看门狗定时器使能 注：看门狗默认开启，在 768ms 时发生看门狗复位
4	RSVO	只能写 0 备注：该寄存器位 X32K_SEL，0 时表示选择内部 32K 作为 WDT 的 32K 工作时钟，1 时表示选择外部 CLK_RTC 时钟；由于已经删除了 CLK_RTC 时钟，因此该寄存器只能写 0，不能写 1。
3	SLEEPDIS	0 = 使能睡眠定时器 1 = 禁止睡眠定时器
2:0	SLPINTS[2:0]	睡眠定时器溢出时间： 000 4ms 001 8ms 010 16ms 011 32ms 100 256ms 101 512ms 110 1024ms 111 2048ms

### 11.5.2 SLPTIM\_SR (0x89)

Bit	7	6	5	4	3	2	1	0
Name	SLPEV	-	-	-	-	-	-	RSV
Reset	0	-	-	-	-	-	-	0
Type	R/W	-	-	-	-	-	-	R/W

Bit	Name	Function
7	SLPEV	0 = 睡眠计数器没有溢出 1 = 睡眠计数器溢出 写 0 清除该位。



6:1	N/A	保留位，读 0
0	RSV	该保留位只能写 0，读为 0。

## 11.5.3 SLPTIM\_CLR (0x8A)

Bit	7	6	5	4	3	2	1	0
Name	SLPTIM_CLR							
Reset	-							
Type	W							

Bit	Name	Function
7:0	SLPTIM_CLR	写任何值到该寄存器清除看门狗。

## 11.5.4 SLPTIM\_CNTRL (0x8C)

Bit	7	6	5	4	3	2	1	0
Name	CNTRL							
Reset	0x00							
Type	RO							

Bit	Name	Function
7:0	CNTRL	看门狗计数器计数值低 8 位。

## 11.5.5 SLPTIM\_CNTRH (0x8D)

Bit	7	6	5	4	3	2	1	0
Name	CNTRH							
Reset	0x00							
Type	RO							

Bit	Name	Function
7:0	CNTRH	看门狗计数器计数值高 8 位。

## 11.5.6 SLPTIM\_WDT (0x8B)

Bit	7	6	5	4	3	2	1	0
Name	-	WDTOV	WDTCNTR[1:0]		-	-	-	-
Reset	-	0	0	0	-	-	-	-
Type	-	RO	R/W	R/W	-	-	-	-

Bit	Name	Function
7	N/A	保留位，读 0
6	WDTOV	看门狗溢出标志： 0 看门狗没有溢出 1 看门狗溢出
5:4	WDTCNTR[1:0]	2 bit 看门狗计数器计数值，只能通过写 0 清除。
3:0	N/A	保留位，读 0



## 11.5.7 SLPTIM\_PRDL (0x8E)

Bit	7	6	5	4	3	2	1	0
Name	ACCPRDL							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	ACCPRDL	睡眠定时器溢出值低 8 位

## 11.5.8 SLPTIM\_PRDRH (0x8F)

Bit	7	6	5	4	3	2	1	0
Name	ACCSEL	-	-	-	-	ACCPRDRH		
Reset	0	-	-	-	-	0		
Type	R/W	-	-	-	-	R/W		

Bit	Name	Function
7	ACCSEL	0 = 选择睡眠定时器溢出值为固定值 1 = 选择睡眠定时器溢出值为 11 位可配置
6:3	N/A	保留位, 读 0
2:0	ACCPRDRH	睡眠定时器高 3 位



## 12. 在线调试

本芯片支持在线调试功能，可以通过 R-LINK 仿真器和 JTAG 协议接口对本芯片进行在线调试。8051 内核自带 JTAG 仿真调试接口，外部逻辑通过 JTAG 接口访问 8051 内部的仿真调试逻辑实现仿真调试功能。

仿真调试功能锁定：

本芯片在退出复位（必须保证 jtag 序列识别逻辑和 GPIO 模块退出复位）后一段时间 CPU 内核还处于复位状态，此段时间内内核还没有开始执行指令，称为内核复位窗口期，持续时间约 2ms。内核退出复位后，开始执行程序对芯片进行初始化，包括对管脚映射的配置。为保证 JTAG 仿真调试功能，必须在程序改变 JTAG 管脚功能前锁定 JTAG 功能。

在内核复位窗口期内，阻止对 JTAG 相关管脚（P0.2、P1.6、P1.4、P1.3）功能配置寄存器的写操作（写无效果），从而保证这几个管脚作的 JTAG 功能，直到芯片被复位。



## 13. 配置选项

### 13.1 系统控制

#### 13.1.1 系统模式控制寄存器定义

名字	地址	读写	复位值	描述
SCR_CFG	0x91	读写	10000011	系统配置寄存器
SCR_SLEEP	0x92	读写	00000000	休眠寄存器

##### 13.1.1.1 SCR\_CFG (0x91)

Bit	7	6	5	4	3	2	1	0
Name	CALI_WDR	-	CALL_SYSRSTRE	-	-	RSTREQ	BOOT_SHADOW	BOOT
Reset	0	-	0	-	-	0	0	0
Type	R/W1C	-	R	-	-	W	R	R

Bit	Name	Function
7	CALI_WDR	看门狗复位标志 0 没有看门狗复位（该寄存器的清零可以通过外部复位引脚、POR、BOR、写 1 来实现） 1 看门狗复位发生 写 1 清清零 CALI_SYSRSTREQ, CALI_WDR
6	N/A	保留位，读 0
5	CALL_SYSRSTRE	软件复位标志 0 没有软件复位发生 1 软件复位发生
4:3	N/A	保留位，读 0
2	RSTREQ	软件复位使能： 0 不复位系统 1 复位系统 注意：RSTREQ 会将 CPU 复位，以及 CPU 相关的中断控制器、SRAM 和 MTP 等，外设不会被复位。
1	BOOT_SHADOW	系统复位标志 1 系统发生复位 0 系统复位完成 该标志可以写 1 清除
0	BOOT	系统复位标志 1 系统发生复位 0 系统复位完成 该标志可以写 1 清，清除该标志时会一同清除掉 BOOT_SHADOW 标志



## 13.1.1.2 SCR\_SLEEP (0x92)

Bit	7	6	5	4	3	2	1	0
Name	FLASH_BUSY	-					SLEEPDEEP	SLEEP
Reset	0	-					0	0
Type	R	-					R/W	R/W

Bit	Name	Function
7	FLASH_BUSY	FLASH 编程读模式下 FLASH_BUSY 的值表示： 0 FLASH 编程完成 1 FLASH 编程没有完成
6:2	N/A	保留位，读 0
1	SLEEPDEEP	深度休眠模式控制： 0 深度休眠模式关闭 1 深度休眠模式打开
0	SLEEP	休眠模式控制： 0 正常工作模式 1 休眠模式

## 13.2 模拟控制

## 13.2.1 模拟控制寄存器定义

名字	地址	读写	复位值	描述
BG_CR	0xFF80	读写	00000000	Bandgap 使能寄存器
BORLVD_CR	0xFF85	读写	00010101	BORLVD 控制寄存器
BORLVD_STAT	0xFF86	读写	00000000	BORLVD 状态寄存器
IMO_CR	0xFF88	读写	00000001	IMO 控制寄存器

## 13.2.1.1 BG\_CR (0xFF80)

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	BG_VON_N	-	-	-	BG_EN_N
Reset	-	-	-	0	-	-	-	0
Type	-	-	-	R/W	-	-	-	R/W

Bit	Name	Function
7:5	N/A	保留位，读 0
4	BG_VON_N	Bandgap 输出控制位： 0 输出 Bandgap 1 Bandgap 工作但不输出
3:1	N/A	保留位，读 0
0	BG_EN_N	Bandgap 使能控制位： 0 使能 Bandgap 1 关闭 Bandgap



## 13.2.1.2 BORLVD\_CR (0xFF85)

Bit	7	6	5	4	3	2	1	0
Name	BOR_VSEL[2:0]			BOR_EN	LVD_VSEL[1:0]			LVD_EN
Reset	0			1	0			1
Type	R/W			R/W	R/W			R/W

Bit	Name	Function
7:5	BOR_VSEL[2:0]	BOR 电压点选择: 000 2.12V(默认值) 001 2.51V 010 2.88V 011 3.58V 100 4.21V 其它 保留
4	BOR_EN	BOR 控制位 0 关闭 BOR 1 使能 BOR
3:1	LVD_VSEL[2:0]	LVD 电压点选择 000 2.32V(默认值) 001 2.75V 010 3.03V 011 3.81V 100 4.54V 其它 保留
0	LVD_EN	LVD 控制位 0 关闭 LVD 1 使能 LVD

## 13.2.1.3 BORLVD\_STAT (0xFF86)

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	IE_LVD	STAT_BOR	-	-	STAT_LVD
Reset	-	-	-	0	0	-	-	0
Type	-	-	-	R/W	R/W	-	-	R/W

Bit	Name	Function
7:5	N/A	保留位, 读 0
4	BOR_VSEL	LVD 中断使能位 0 禁止 LVD 中断 1 使能 LVD 中断
3	STAT_BOR	BOR 输出状态 0 BOR 没有发生 1 BOR 发生
2:1	N/A	保留位, 读 0



0	STAT_LVD	LVD 输出状态 0 没有 LVD 事件发生 1 检测到 LVD 事件 该标志位只能读，不能清除。
---	----------	--

## 13.2.1.4 IMO\_CR (0xFF88)

Bit	7	6	5	4	3	2	1	0
Name	EXT_SEL	FX2_SEL	-	RESET_CTRL	IMO_TSTEN	-	-	IMO_EN
Reset	0	0	-	0	0	-	-	1
Type	R/W	R/W	-	R/W	R/W	-	-	R/W

Bit	Name	Function
7:6	[EXT_SEL:FX2_SEL]	系统时钟源选择 0x 选择内部 8MHz 时钟 10 选择 P2.1 做系统时钟，需要将 P2.1 复用功能打开 11 选择内部 16MHz 时钟
5	N/A	保留位，读 0
4	RESET_CTRL	模拟模块振荡器深睡唤醒复位使能 0: 唤醒不复位 1: 唤醒复位
3	IMO_TSTEN	0 IMO 测试功能关闭 1 IMO 测试功能打开，选择 SCK1 到 P1.0 口。 备注：使用 IMO 测试功能之前，要把 P1.0 的 GPIO 复用功能打开。IMO 测试功能使能时优先级比 TIM2_CHA 优先级高。
2	N/A	保留位，勿操作
1	N/A	保留位，读 0
0	IMO_EN	写模式下 0 使能 IMO 1 关闭 IMO 读模式下 0 IMO 关闭 1 IMO 使能 备注：关闭 IMO 需要分两个步骤，先写一次 IMOCR 将系统时钟源切换成外部时钟并保持 IMO 使能，再写一次 IMOCR 将 IMO 关闭并保持外部时钟作为系统时钟。





## 14. 电气特性

### 14.1 绝对最大额定值

参数	最小值	最大值	单位
存储器温度	-55	125	°C
工作温度	-40	85	°C
工作电压	2.4	5.5	V
VDD 对地电压	-0.3	5.5	V
I/O 对地电压	-0.3	VDD+0.3	V

### 14.2 直流特性

符号	参数	测试条件	最小值	典型值	最大值	单位
		VDD=5V, 常温 25°C				
f <sub>FLASH</sub>	FLASH 工作频率	4.5 ≤ VDD < 5.5			8	MHz
		2.4V ≤ VDD < 5.5V			2.6	MHz
IDD1	工作电流 1	内部 16MHz RC 振荡器工作, CPU 工作在 16MHz, PT02、PT16 默认开漏上拉, 其余 I/O 输入高阻, 开启上拉, 关闭 WDT, BOR 开启。		6.5		mA
IDD2	工作电流 2	内部 16MHz RC 振荡器工作, CPU 关闭, PT02、PT16 默认开漏上拉, 其余 I/O 输入高阻, 开启上拉, 关闭 WDT, 开启 BOR。		2.2		mA
ISP	静态电流	内部 16MHz RC 振荡器关闭, 32KHz 时钟打开, CPU 工作在 DEEPSLEEP 模式, PT02、PT16 默认开漏上拉, 其余 I/O 输入高阻, 开启上拉, 关闭 WDT, 开启 BOR。		4.2	7.8	uA
VIL	输入低电平				0.3VDD	
VIH	输入高电平		0.5VDD			
R <sub>PU</sub>	上拉电阻			10		K Ω
R <sub>PD</sub>	下拉电阻			10		K Ω
I <sub>OH1</sub>	拉电流 1	GPIO 强推挽输出输出 5V, 无负载		13		mA
I <sub>OL1</sub>	灌电流 1	GPIO 配置强推挽输出, 关闭上下拉, 输出低电平 (VDD=5V), IO 与 GND 灌入一个 0.3V 电压		12		mA



### 14.3 ADC 特性

符号	参数	条件	最小值	典型值	最大值	单位
	分辨率	单端转换		12		比特
	积分非线性误差			±2	±3	LSB
	微分非线性误差			±2	±3	LSB
	增益误差			±2	±6	LSB
	偏移误差			±2	±4	LSB
	转换速度			200	300	ksps
$V_{DD}$	模拟供电电压		2.4	5	5.5	V
$V_{REF}$	参考电压		0		$V_{CC}$	V
$V_{IN}$	输入电压		0		$V_{REF}$	V
$V_{INT}$	内部电压参考			1.2		V
				2.4		V

### 14.4 比较器特性

符号	参数	条件	最小值	典型值	最大值	单位
$V_{CMPIN}$	输入电压范围		0		$V_{DD}-1$	V

### 14.5 EMC 特性

#### Electrostatic discharge (ESD)

符号	参数	测试条件	封装	最大值	单位
$V_{ESD(HBM)}$	Electrostatic discharge voltage (Human body model)	ANSI/ESDA/JEDEC JS-001-2017	TSSOP20	±8000	V
$V_{ESD(CDM)}$	Electrostatic discharge voltage (Charge device model)	ANSI/ESDA/JEDEC JS-002-2018		±2000	V
$V_{ESD(MM)}$	Electrostatic discharge voltage (Machine model)	JEDEC EIA/JESD22-A115C 2010		±600	V
LU	Latch-up	JEDEC STANDART NO. 78E APRIL 2016		±200	mA

## 15. 芯片封装信息

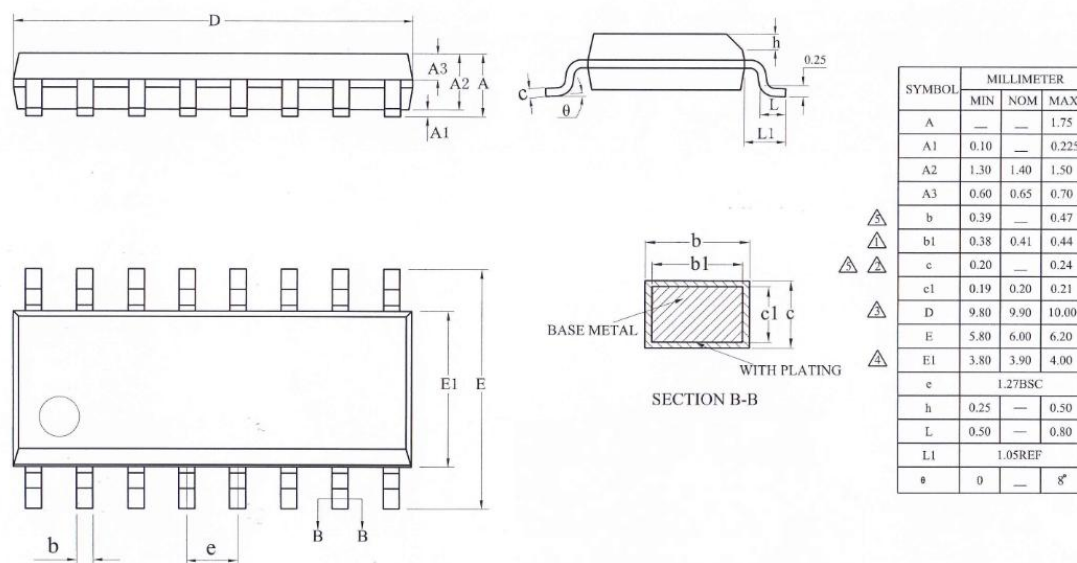


图 34 SOP16 封装外形图

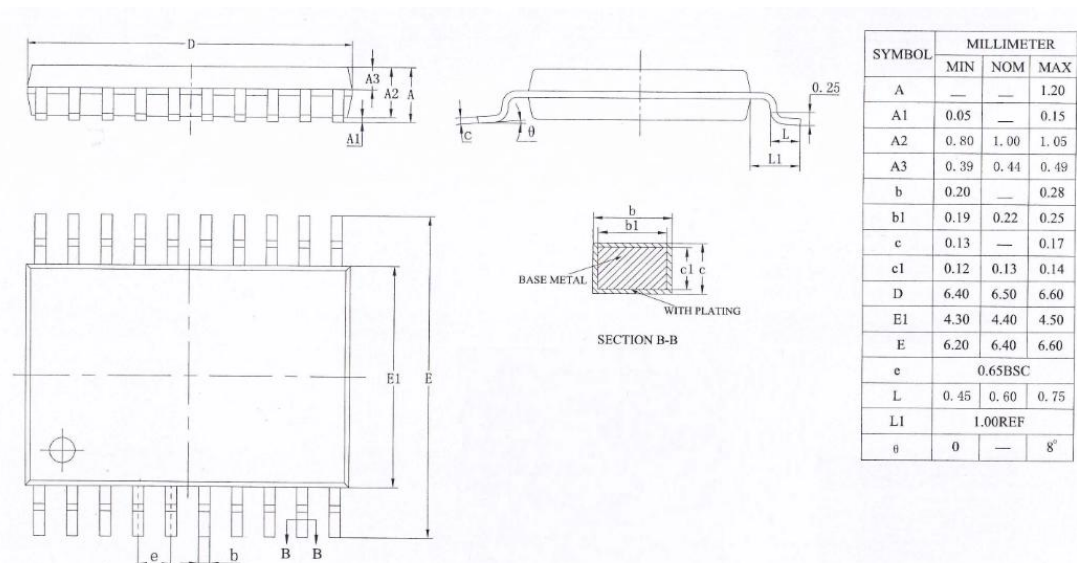


图 35 TSSOP20 封装外形图

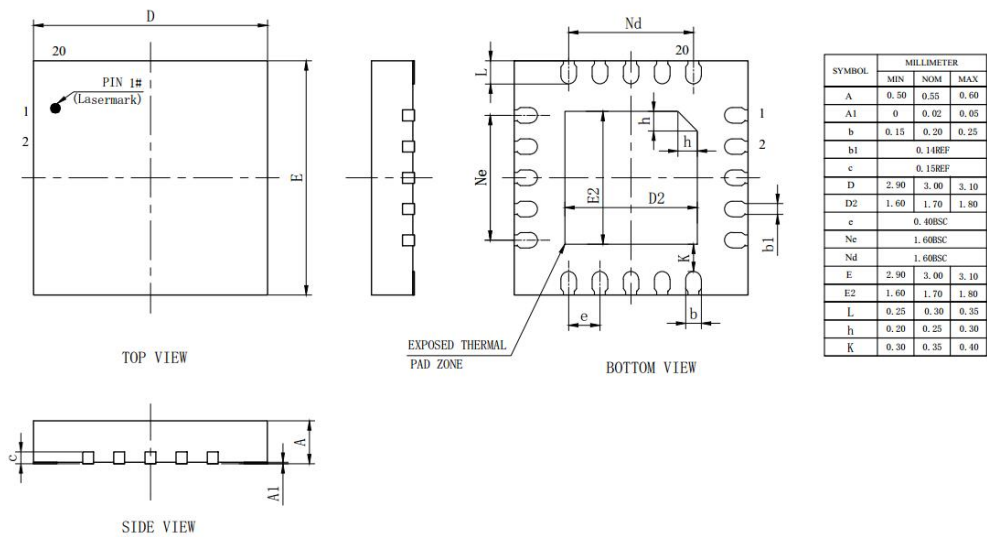


图 36 QFN20 封装外形图