



矽杰微电子
XIJIE MICROELECTRONICS

XC8F9362 用户手册

8 位 Flash 微控制器

Ver1.4

免责声明

无锡矽杰微电子有限公司（简称：无锡矽杰微）保留关于该规格书中产品的可靠性、功能和设计方面的改进作进一步说明的权利。由于使用本用户手册中的信息或内容而导致的直接，间接，特别附带结果的损害，无锡矽杰微没有义务负责。本用户手册中提到的其应用仅仅是用来做说明，本公司不保证这些应用没有更深入的测试就能适用。本规格书中提到的软件（如果有），都是依据授权或保密合约所合法提供的，并且只能在这些合约的许可条件下使用或者复制。无锡矽杰微的产品不是专门设计来应用于生命维持的用具，装置或者系统。无锡矽杰微的产品不支持而且禁止在这些方面的应用。本用户手册内容如有变动恕不另作通知，具体更新信息，请参考公司官方网站 www.xjmcu.com



修改记录说明

版本号	修改说明	备注
V1.0	发布初稿	
V1.1	1. 修改引脚描述 2. 增加看门狗, SLTIM 操作说明 3. 增加串口波特率说明	
V1.2	修改部分寄存器默认值	
V1.3	修改目录错误和部分重复部分, 补充 MSOP10 脚位图和封装图, 修改 FLASH_CFG 部分信息	
V1.4	修改关于 ADC 外部参考电压引脚部分	



目 录

1. 芯片简介	8
1.1 功能特性	8
1.2 引脚分配	9
1.3 引脚描述	12
1.4 系统框图	14
2. 存储器	15
2.1 程序存储区	15
2.2 数据存储区	16
2.3 FLASH 控制概述	17
2.4 SFR	17
2.5 XDATA	17
3. 功能描述	19
3.1 SFR 空间寄存器	19
3.1.1 SFR~0x81/SP (堆栈指针)	19
3.1.2 SFR~0x82/DPL0 (DPTR0 寄存器的低 8 位)	19
3.1.3 SFR~0x83/DPH0 (DPTR0 寄存器的高 8 位)	19
3.1.4 SFR~0x84/DPL1 (DPTR1 寄存器的低 8 位)	19
3.1.5 SFR~0x85/DPH1 (DPTR1 寄存器的高 8 位)	20
3.1.6 SFR~0x86/DPS (DPTR0/DPTR1 选择寄存器)	20
3.1.7 SFR~0x88/SLPTIM_CR (睡眠计数器控制寄存器)	20
3.1.8 SFR~0x89/SLPTIM_SR (睡眠计数状态寄存器)	21
3.1.9 SFR~0x8A/SLPTIM_CLR (看门狗清除寄存器)	21
3.1.10 SFR~0x8B/SLPTIM_WDT (看门狗计数器状态寄存器)	21
3.1.11 SFR~0x8C/SLPTIM_CNTL (看门狗计数器计数值寄存器低 8 位)	22
3.1.12 SFR~0x8D/SLPTIM_CNTH (看门狗计数器计数值寄存器高 8 位)	22
3.1.13 SFR~0x8E/SLPTIM_PRDL (睡眠计数器预分频寄存器低 8 位)	22
3.1.14 SFR~0x8F/SLPTIM_PDRH (睡眠计数器预分频寄存器高 8 位)	22
3.1.15 SFR~0x91/SCR_CFG (系统配置寄存器)	23
3.1.16 SFR~0x92/SCR_SLEEP (睡眠寄存器)	23
3.1.17 SFR~0x93/SCR_CALI (校准寄存器)	23
3.1.18 SFR~0x94/CLK_CR (系统时钟控制寄存器)	24
3.1.19 SFR~0x95/PCLK_CR (外设时钟控制寄存器)	24
3.1.20 SFR~0x96/PCLK_DIV12 (SCK1、SCK2 时钟控制寄存器)	25
3.1.21 SFR~0x97/PCLK_DIV3 (SCK3 时钟控制寄存器)	26
3.1.22 SFR~0x98/P0_DR (端口 0 数字寄存器)	26
3.1.23 SFR~0x99/P0_DMO (端口 0 模式 0 位)	26
3.1.24 SFR~0x9A/P0_DM1 (端口 1 模式 1 位)	26
3.1.25 SFR~0x9C/UARTO_DR (UART 数据寄存器)	27
3.1.26 SFR~0x9D/UARTO_CR (UART 控制寄存器)	27
3.1.27 SFR~0x9E/UARTO_SR (UART 状态寄存器)	28
3.1.28 SFR~0x9F/UARTO_CFG (UART 配置寄存器)	29
3.1.29 SFR~0xA0/P2 (P2 读写寄存器)	29
3.1.30 SFR~0xA1/I2C_ADDR (I2C 从机地址寄存器)	29
3.1.31 SFR~0xA2/I2C_CR (I2C 控制寄存器)	30
3.1.32 SFR~0xA3/I2C_STAT (I2C 状态寄存器)	30
3.1.33 SFR~0xA4/I2C_DR (I2C 数据寄存器)	31



3.1.34	SFR~0xA5/I2C_MCR (I2C 主机控制寄存器)	31
3.1.35	SFR~0xA8/IE (系统中断使能寄存器)	32
3.1.36	SFR~0xAA/INT_MSK0 (中断屏蔽寄存器 0)	32
3.1.37	SFR~0xAB/INT_MSK1 (中断屏蔽寄存器 1)	33
3.1.38	SFR~0xAD/INT_PRI0 (中断优先级配置寄存器 0)	34
3.1.39	SFR~0xAE/INT_PRI1 (中断优先级配置寄存器 1)	34
3.1.40	SFR~0xB0/P1_DR (端口 1 数据寄存器)	35
3.1.41	SFR~0xB1/P1_DMO (端口 1 模式 0 位)	35
3.1.42	SFR~0xB2/P1_DM1 (端口 1 模式 1 位)	36
3.1.43	SFR~0xB8/P2_DR (端口 2 数据寄存器)	36
3.1.44	SFR~0xB9/P2_DMO (端口 2 模式 0 位)	36
3.1.45	SFR~0xBA/P2_DM1 (端口 2 模式 1 位)	36
3.1.46	SFR~0xBC/UART1_DR (UART1 数据寄存器)	37
3.1.47	SFR~0xBD/UART1_CR (UART 控制寄存器)	37
3.1.48	SFR~0xBE/UART1_SR (UART 状态寄存器)	38
3.1.49	SFR~0xBF/UART1_CFG (UART 配置寄存器)	39
3.1.50	SFR~0xC0/TIM1_CR (Timer1 控制寄存器)	39
3.1.51	SFR~0xC1/TIM1_IE (Timer1 中断控制寄存器)	40
3.1.52	SFR~0xC2/TIM1_SR (Timer1 状态寄存器)	40
3.1.53	SFR~0xC8/TIM2_CR (Timer2 控制寄存器)	41
3.1.54	SFR~0xC9/TIM2_IE (Timer2 中断控制寄存器)	42
3.1.55	SFR~0xCA/TIM2_SR (Timer2 状态寄存器)	42
3.1.56	SFR~0xD0/PSW (程序状态字寄存器)	43
3.1.57	SFR~0xE0/ACC (累加寄存器)	43
3.1.58	SFR~0xE8/ADC_CR0 (ADC 转换控制寄存器 0)	44
3.1.59	SFR~0xE9/ADC_CR1 (ADC 转换控制寄存器 1)	45
3.1.60	SFR~0xEA/ADC_CR2 (ADC 转换控制寄存器 2)	45
3.1.61	SFR~0xEB/ADC_CHSEL (ADC 模拟量输入通道选择寄存器)	46
3.1.62	SFR~0xEC/ADC_CON (ADC 配置寄存器)	46
3.1.63	SFR~0xED/ADC_DLY (ADC 触发延迟配置寄存器)	47
3.1.64	SFR~0xEE/ADC_RES_L (ADC 转换结果低位寄存器)	47
3.1.65	SFR~0xEF/ADC_RES_H (ADC 转换结果高位寄存器)	47
3.1.66	SFR~0xF0/B (B 寄存器)	48
3.1.67	SFR~0xF8/TIM0_CR (Timer0 控制寄存器)	48
3.1.68	SFR~0xF9/TIM0_CNTR (Timer0 计数值寄存器)	49
3.1.69	SFR~0xFA/TIM0_ARR (Timer0 自动重装寄存器)	49
3.1.70	SFR~0xFB/TIM0_IE (Timer0 中断控制寄存器)	49
3.1.71	SFR~0xFC/TIM0_SR (Timer0 状态寄存器)	49
3.1.72	SFR~0xFD/SSCONR (Timer1/2 软件同步控制寄存器)	50
3.1.73	SFR~0xFE/ADC_COMPL (ADC 比较值低 4 位)	50
3.1.74	SFR~0xFF/ADC_COMP_H (ADC 比较值高 8 位)	50
3.2	XDATA 空间寄存器	51
3.2.1	XDATA~0xFF00/FLASH_CR (FLASH 控制寄存器)	51
3.2.2	XDATA~0xFF01/FLASH_CFG (FLASH 配置寄存器)	51
3.2.3	XDATA~0xFF10/PT_SEL (端口位置配置寄存器)	52
3.2.4	XDATA~0xFF18/PERP0_EN (外设管脚位置使能配置寄存器 0)	53
3.2.5	XDATA~0xFF19/PERP1_EN (外设管脚位置使能配置寄存器 0)	54
3.2.6	XDATA~0xFF20/PO_FLAG (端口 0 中断标志位)	54



3.2.7 XDATA~0xFF21/P0_GE (端口 0 数字复用使能寄存器)	55
3.2.8 XDATA~0xFF23/P0_PU (端口 0 上拉控制寄存器)	55
3.2.9 XDATA~0xFF24/P0_PD (端口 0 下拉控制寄存器)	55
3.2.10 XDATA~0xFF25/P0_IE (端口 0 中断使能寄存器)	56
3.2.11 XDATA~0xFF26/P0_IC0 (端口 0 中断控制 0 位)	56
3.2.12 XDATA~0xFF27/P0_IC1 (端口 0 中断控制 1 位)	56
3.2.13 XDATA~0xFF30/P1_FLAG (端口 1 中断标志位)	56
3.2.14 XDATA~0xFF31/P1_GE (端口 1 数字复用使能寄存器)	57
3.2.15 XDATA~0xFF33/P1_PU (端口 1 上拉控制寄存器)	57
3.2.16 XDATA~0xFF34/P1_PD (端口 1 下拉控制寄存器)	57
3.2.17 XDATA~0xFF35/P1_IE (端口 1 中断使能寄存器)	58
3.2.18 XDATA~0xFF36/P1_IC0 (端口 1 中断控制 0 位)	58
3.2.19 XDATA~0xFF37/P1_IC1 (端口 1 中断控制 1 位)	58
3.2.20 XDATA~0xFF40/P2_FLAG (端口 2 中断标志位)	58
3.2.21 XDATA~0xFF41/P2_GE (端口 2 数字复用使能寄存器)	59
3.2.22 XDATA~0xFF43/P2_PU (端口 2 上拉控制寄存器)	59
3.2.23 XDATA~0xFF44/P2_PD (端口 2 下拉控制寄存器)	59
3.2.24 XDATA~0xFF45/P2_IE (端口 2 中断使能寄存器)	60
3.2.25 XDATA~0xFF46/P2_IC0 (端口 1 中断控制 0 位)	60
3.2.26 XDATA~0xFF47/P2_IC1 (端口 1 中断控制 1 位)	60
3.2.27 XDATA~0xFF50/TIM1_FCONR (Timer1 时钟控制寄存器)	60
3.2.28 XDATA~0xFF51/TIM1_VPERR (Timer1 周期间隔响应控制寄存器)	61
3.2.29 XDATA~0xFF52/TIM1_DTUA (Timer1 死区事件寄存器)	62
3.2.30 XDATA~0xFF53/TIM1_BRAKE (Timer1 刹车控制寄存器)	62
3.2.31 XDATA~0xFF54/TIM1_DTR (Timer1 死区控制寄存器)	63
3.2.32 XDATA~0xFF55/TIM1_PCONRA (Timer1 端口 A 控制寄存器)	64
3.2.33 XDATA~0xFF56/TIM1_PCONRB (Timer1 端口 B 控制寄存器)	64
3.2.34 XDATA~0xFF58/TIM1_CNTL (Timer1 计数值寄存器低 8 位)	65
3.2.35 XDATA~0xFF59/TIM1_CNTH (Timer1 计数值寄存器高 8 位)	65
3.2.36 XDATA~0xFF5A/TIM1_ARRL (Timer1 自动重载寄存器低 8 位)	66
3.2.37 XDATA~0xFF5B/TIM1_ARRH (Timer1 自动重载寄存器高 8 位)	66
3.2.38 XDATA~0xFF5C/TIM1_GCMARL (Timer1 比较捕获寄存器 A 低 8 位)	66
3.2.39 XDATA~0xFF5D/TIM1_GCMARH (Timer1 比较捕获寄存器 A 高 8 位)	66
3.2.40 XDATA~0xFF5E/TIM1_GCMBRL (Timer1 比较捕获寄存器 B 低 8 位)	67
3.2.41 XDATA~0xFF5F/TIM1_GCMBRH (Timer1 比较捕获寄存器 B 高 8 位)	67
3.2.42 XDATA~0xFF60/TIM2_FCONR (Timer2 时钟控制寄存器)	67
3.2.43 XDATA~0xFF61/TIM2_VPERR (Timer2 周期间隔响应控制寄存器)	68
3.2.44 XDATA~0xFF62/TIM2_DTUA (Timer2 死区事件寄存器)	69
3.2.45 XDATA~0xFF63/TIM2_BRAKE (Timer2 刹车控制寄存器)	69
3.2.46 XDATA~0xFF64/TIM2_DTR (Timer2 死区控制寄存器)	70
3.2.47 XDATA~0xFF65/TIM2_PCONRA (Timer2 端口 A 控制寄存器)	70
3.2.48 XDATA~0xFF66/TIM2_PCONRB (Timer2 端口 B 控制寄存器)	71
3.2.49 XDATA~0xFF68/TIM2_CNTL (Timer2 计数值寄存器低 8 位)	72
3.2.50 XDATA~0xFF69/TIM2_CNTH (Timer2 计数值寄存器高 8 位)	72
3.2.51 XDATA~0xFF6A/TIM2_ARRL (Timer2 自动重载寄存器低 8 位)	72
3.2.52 XDATA~0xFF6B/TIM2_ARRH (Timer2 自动重载寄存器高 8 位)	73
3.2.53 XDATA~0xFF6C/TIM2_GCMARL (Timer2 比较捕获寄存器 A 低 8 位)	73
3.2.54 XDATA~0xFF6D/TIM2_GCMARH (Timer2 比较捕获寄存器 A 高 8 位)	73



3.2.55 XDATA~0xFF6E/TIM1_GCMBRL (Timer2 比较捕获寄存器 B 低 8 位)	73
3.2.56 XDATA~0xFF6F/TIM2_GCMBRH (Timer3 比较捕获寄存器 B 高 8 位)	74
3.2.57 XDATA~0xFF80/BG_CR (Bandgap 使能寄存器)	74
3.2.58 XDATA~0xFF85/BORLVD_CR (BORLVD 控制寄存器)	74
3.2.59 XDATA~0xFF86/BORLVD_STAT (BORLVD 状态寄存器)	75
3.2.60 XDATA~0xFF88/IMO_CR (IMO 控制寄存器)	75
3.3 GPIO	77
3.3.1 配置 I/O 口	77
3.3.2 外设功能管脚	78
3.4 CPU	79
3.5 中断	80
3.5.1 概述	80
3.5.2 GPIO 中断	80
3.5.3 中断优先级和中断屏蔽	80
3.5.4 中断向量表	81
3.6 时钟	82
3.6.1 概述	82
3.6.2 时钟结构框图	82
3.6.3 CPU 时钟	83
3.6.4 SCK1 和 SCK2 时钟	83
3.6.5 SCK3 时钟	83
3.6.6 32K 时钟	83
3.7 复位	84
3.7.1 看门狗复位	84
3.7.2 欠压复位	85
3.7.3 POR 复位	85
3.8 外设	87
3.8.1 8-bit 基本计数器	87
3.8.2 16-bit 高级计数器	88
3.8.3 UART	97
3.8.4 I2C	100
3.8.5 ADC	104
3.9 省电模式	105
3.9.1 睡眠模式	106
3.9.2 深度睡眠模式	106
3.9.3 睡眠定时器中断	107
4. 电气特性	108
4.1 绝对最大额定值	108
4.2 直流特性	108
4.3 ADC 特性	109
4.4 EMC 特性	109
5. 指令集	110
5.1 指令集注释	110
5.2 指令集	111
5.2.1 算术运算	111
5.2.2 逻辑运算	112
5.2.3 布尔操作	113
5.2.4 数据传送	114



5.2.5 分支指令	115
6. 封装尺寸	116



1. 芯片简介

1.1 功能特性

内核

- 超高速 8051 内核 (1T)
- 指令周期可配:
 - 2.6MHz, VDD ≥ 2.4V
 - 8MHz, VDD ≥ 4.5V
- 工作电压: 2.4V~5.5V
- 工作温度: -20°C~85°C
- FlashROM: 8K 字节 Flash ROM (擦写次数典型值 1000 次)

- Flash 包含 64 字节 Information Block
- SRAM: 256 字节 SRAM

时钟

- 内部 16MHz RC 振荡器 (可微调)
 - 误差 ± 2% (2.4~4.5V, -20°C~85°C)
- 内部 32KHz 低速 RC 振荡器 (误差 ± 10%)

复位

- 上电复位
- 欠压复位 (LVR)
 - 2.2V、2.5V、3.6V、4.2V
- 看门狗溢出复位

低电压检测

- LVD 共 4 级
 - 2.3V、2.7V、3.8V、4.5V

中断 (INT)

- Timer0、Timer1、Timer2、SCK3、WDT、ADC、UART0~1、I2C、LVD、P0~P2 共 13 个中断源, 全部 GPIO 可设上升沿、下降沿、高电平、低电平中断
- 可设中断优先级和中断屏蔽

数字外设

- 1 个 8 位基本定时器
 - 预分频 1、2、4、8、16、32、64、128
- 2 个 16 位高级定时器, 支持 4 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
 - 支持全双工和半双工
- 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: 8 路
- 内部输入: 1 路 (1/4 VDD)
- 参考源: 外部参考 (P0.3)、VDD、内部参考: 1.2V/2.4V
- 采样可以通过 PWM 或者管脚的上升沿或者下降沿触发

18 个 GPIO

- PT12、PT13 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%)
- P0_DR、P1_DR、P2_DR 支持位操作读和位操作写

省电模式

- 深度睡眠可由看门狗复位、睡眠定时器中断、引脚中断唤醒
- 深度睡眠电流: 4uA (典型值)

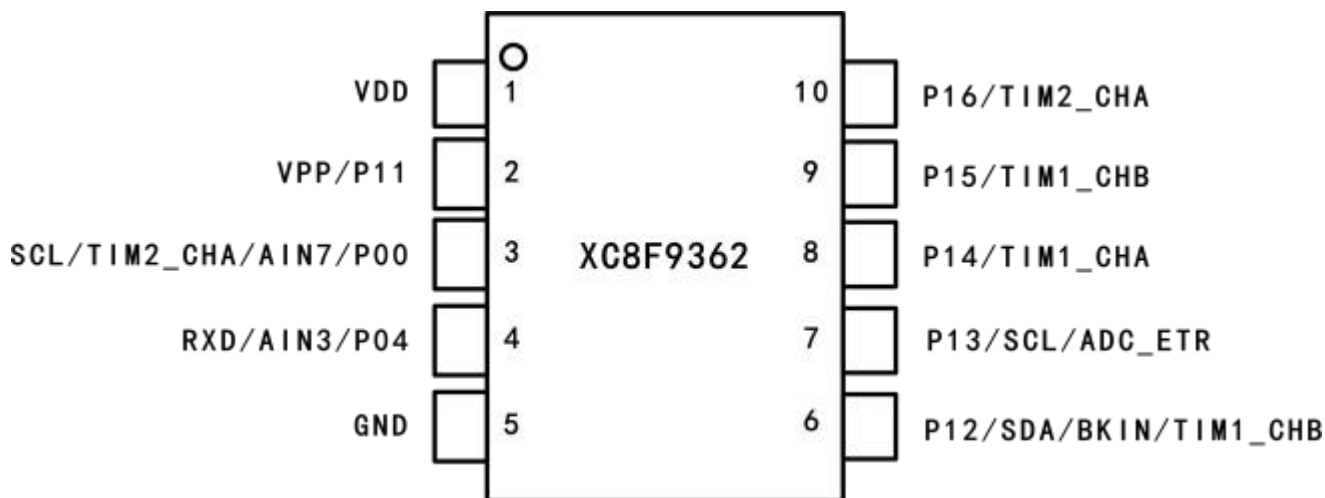
Flash 烧写

- 5 线烧写 (VDD, GND, SDA, SCL, VPP)

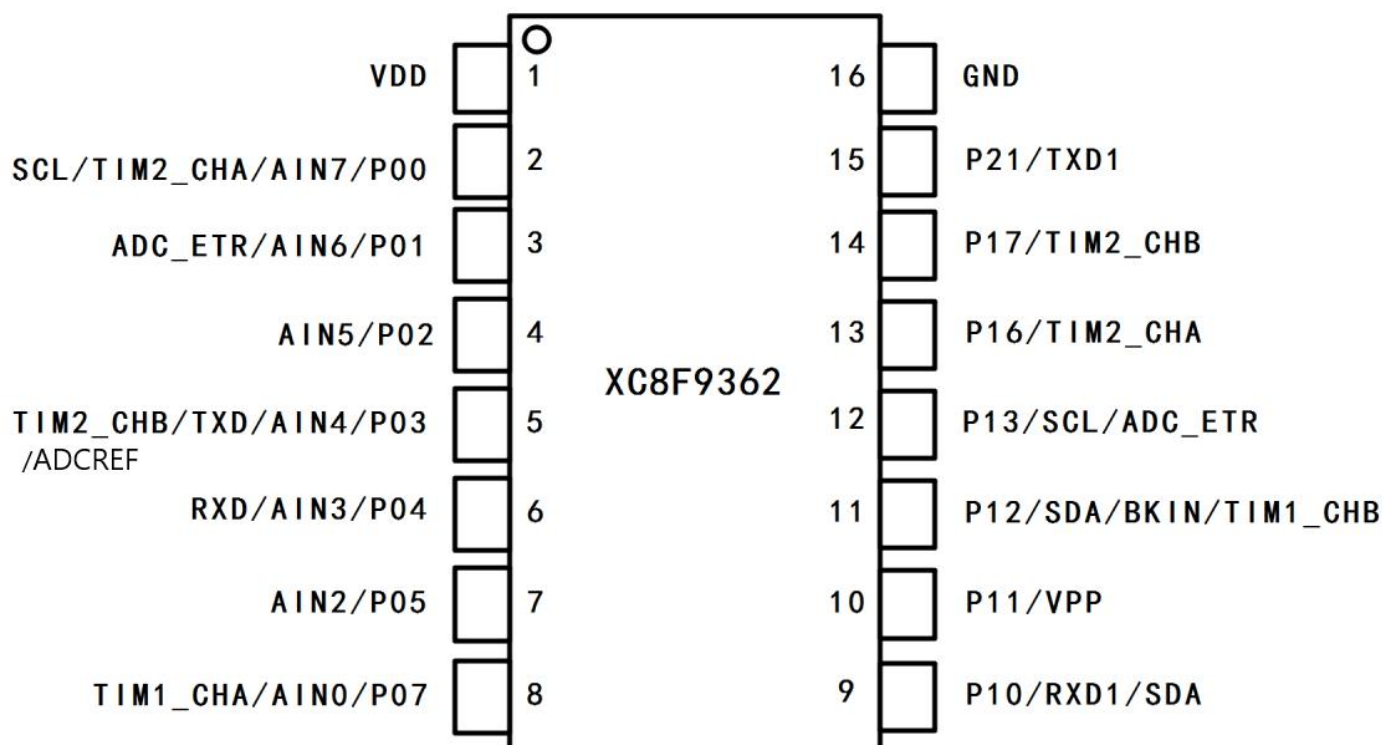
封装类型

- XC8F9362-MSOP10
- XC8F9362-SOP16
- XC8F9362-TSSOP20
- XC8F9362-QFN20 (3*3)

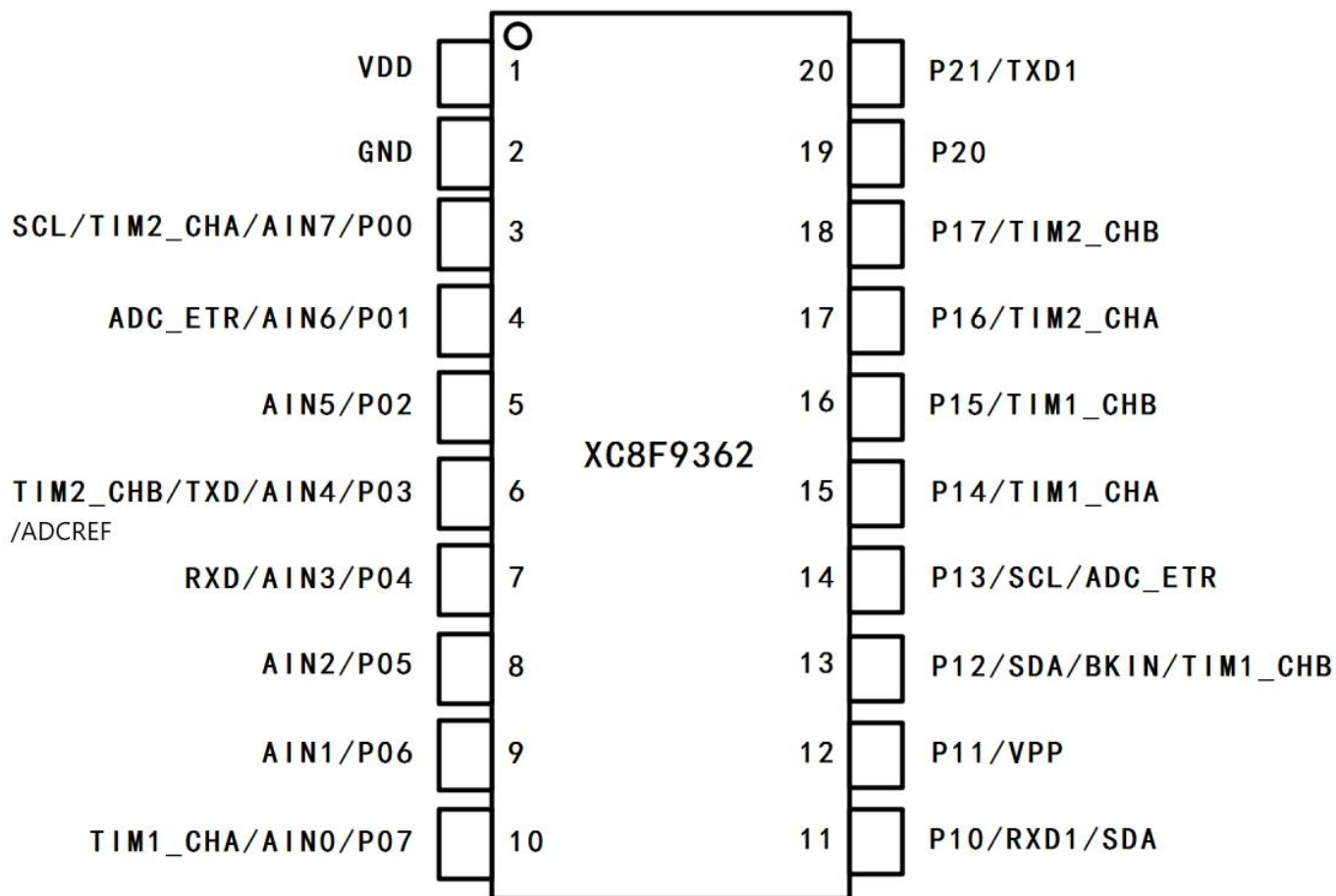
1.2 引脚分配



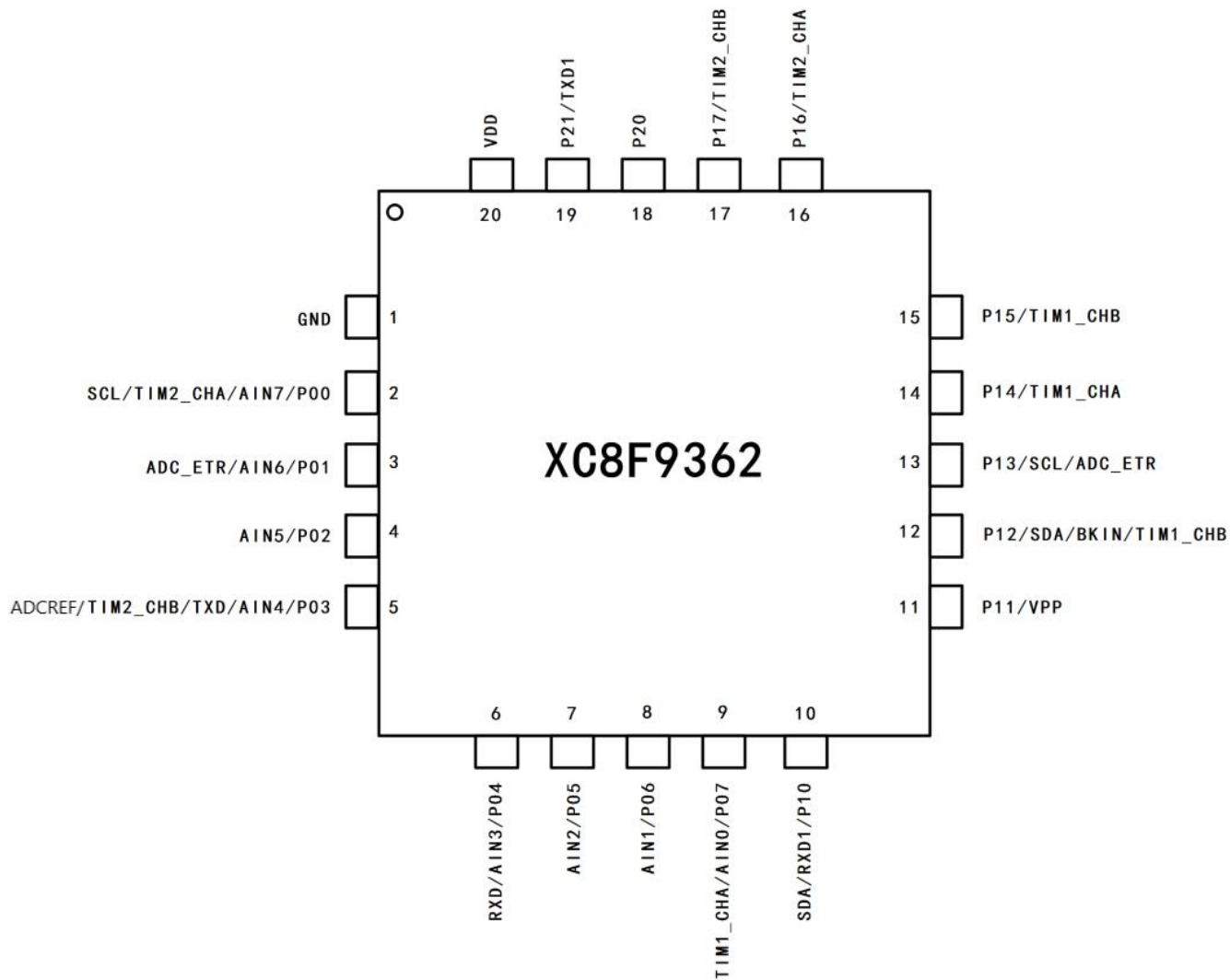
MSOP10 封装脚位图



SOP16 封装脚位图



TSSOP20 封装脚位图



QFN20 封装脚位图



1.3 引脚描述

序号	管脚名	I/O	功能描述
P00	P00	I/O	GPIO, 可单独配置上下拉, 可设中断
	SCL	I/O	I2C 时钟引脚
	TIM2_CHA	I/O	TIMER2 输出引脚 A, 16 位输入捕获引脚 A
	AIN7	I	ADC 外部采样输入通道 7
P01	P01	I/O	GPIO, 可单独配置上下拉, 可设中断
	ADC_ETR	I	ADC 外部触发采样输入引脚
	AIN6	I	ADC 外部采样输入通道 6
P02	P02	I/O	GPIO, 可单独配置上下拉, 可设中断
	AIN5	I	ADC 外部采样输入通道 5
P03	P03	I/O	GPIO, 可单独配置上下拉, 可设中断
	ADCREFP	I	ADC 外部参考电压输入引脚
	TIM2_CHB	I	TIMER2 输出引脚 B, 16 位输入捕获引脚 B
	TXD	O	UART0 数据输出引脚
	AIN4	I	ADC 外部采样输入通道 4
P04	P04	I/O	GPIO, 可单独配置上下拉, 可设中断
	RXD	I	UART0 数据输入引脚
	AIN3	I	ADC 外部采样输入通道 3
P05	P05	I/O	GPIO, 可单独配置上下拉, 可设中断
	AIN2	I/O	ADC 外部采样输入通道 2
P06	P06	I/O	GPIO, 可单独配置上下拉, 可设中断
	AIN1	O	ADC 外部采样输入通道 1
P07	P07	I/O	GPIO, 可单独配置上下拉, 可设中断
	TIM1_CHA	I/O	TIMER1 输出引脚 A, 16 位输入捕获引脚 A
	AIN0	I	ADC 外部采样输入通道 0
P10	P10	I/O	GPIO, 可单独配置上下拉, 可设中断
	RXD1	I	UART1 数据输入引脚
	SDA	I/O	I2C 数据引脚 ⁽¹⁾
P11	P11	I/O	GPIO, 可单独配置上下拉, 可设中断
	VPP	O	烧录高压输入口 (9.6V~10V)



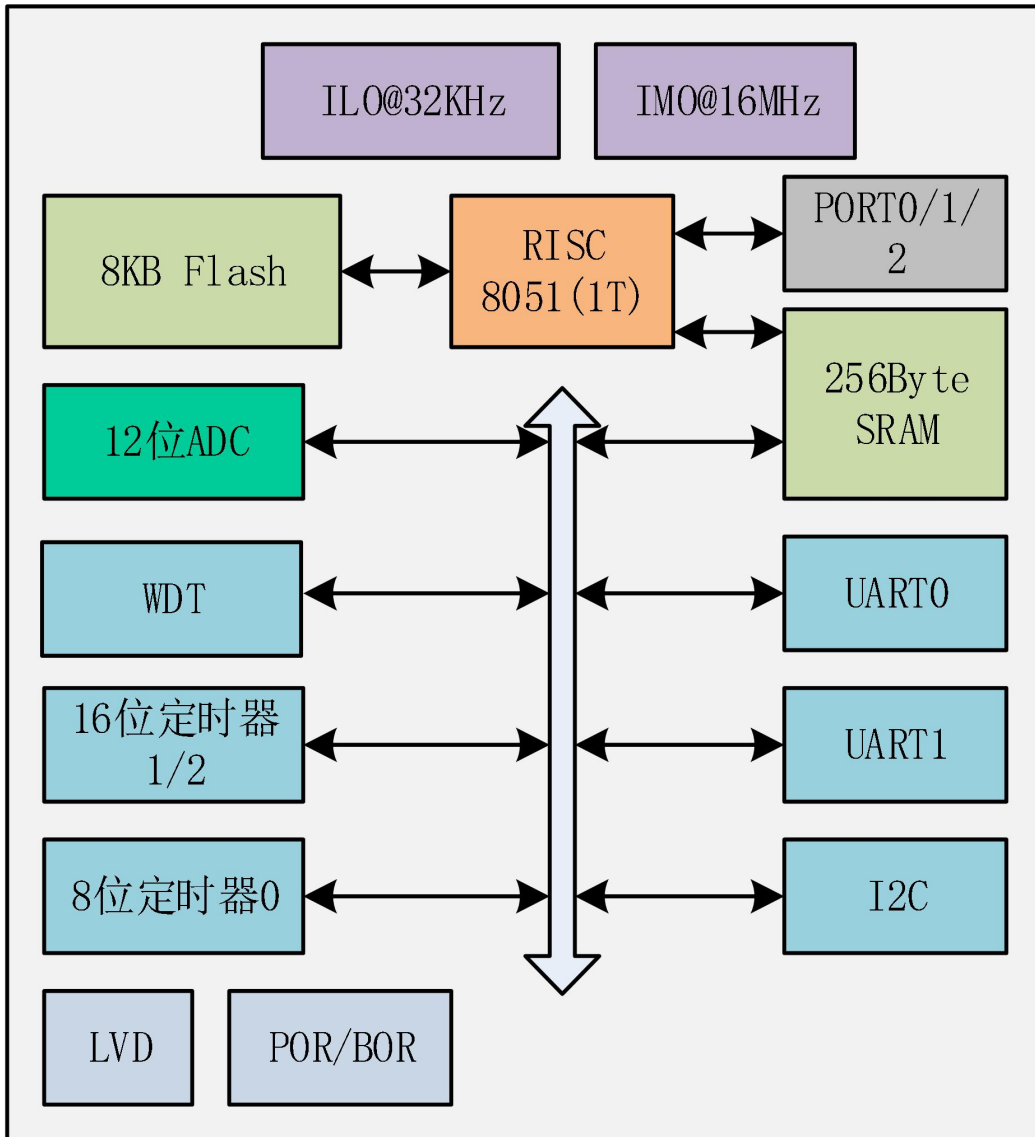
序号	管脚名	I/O	功能描述
P12	P12	I/O	GPIO, 可单独配置上下拉, 可设中断 (默认开漏)
	SDA	I/O	I2C 数据引脚 ⁽¹⁾
	BKIN	I	PWM 刹车输入引脚
	TIM1_CHB	I/O	TIMER1 输出引脚 B, 16 位输入捕获引脚 B
	TDI	I	调试数据输入脚
P13	P13	I/O	GPIO, 可单独配置上下拉, 可设中断 (默认开漏)
	SCL	I/O	I2C 时钟引脚
	ADC_ETR	0	UART0 数据输出引脚
	TCK	I/O	调试时钟脚
P14	P14	I/O	GPIO, 可单独配置上下拉, 可设中断
	TIM1_CHA	I/O	TIMER1 输出引脚 A, 16 位输入捕获引脚 A
P15	P15	I/O	GPIO, 可单独配置上下拉, 可设中断
	TIM1_CHB	I/O	TIMER1 输出引脚 B, 16 位输入捕获引脚 B
P16	P16	I/O	GPIO, 可单独配置上下拉, 可设中断
	TIM2_CHA	I/O	TIMER2 输出引脚 A, 16 位输入捕获引脚 A
P17	P17	I/O	GPIO, 可单独配置上下拉, 可设中断
	TIM2_CHB	I/O	TIMER2 输出引脚 B, 16 位输入捕获引脚 B
P20	P20	I/O	GPIO, 可单独配置上下拉, 可设中断
P21	P21	I/O	GPIO, 可单独配置上下拉, 可设中断
	TXD1	0	UART1 数据输出引脚

注:

1. 上电 2. 1mS, 该脚会输出 50uS 左右的低电平



1.4 系统框图



系统功能结构框图

2. 存储器

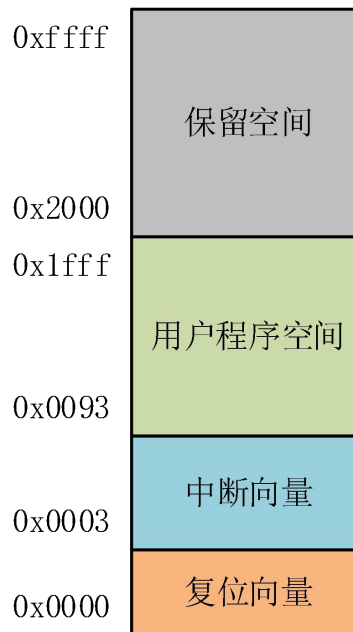
内部 3 种存储器：SFR，内部数据存储器，程序存储器。

程序存储器只能读不能写，该存储器大小为 8k 字节。内部数据存储器大小为 256 字节。SFR 为内部特殊功能寄存器。

2.1 程序存储区

XC8F9362 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 8K 字节的程序存储空间。

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。



程序存储区结构图

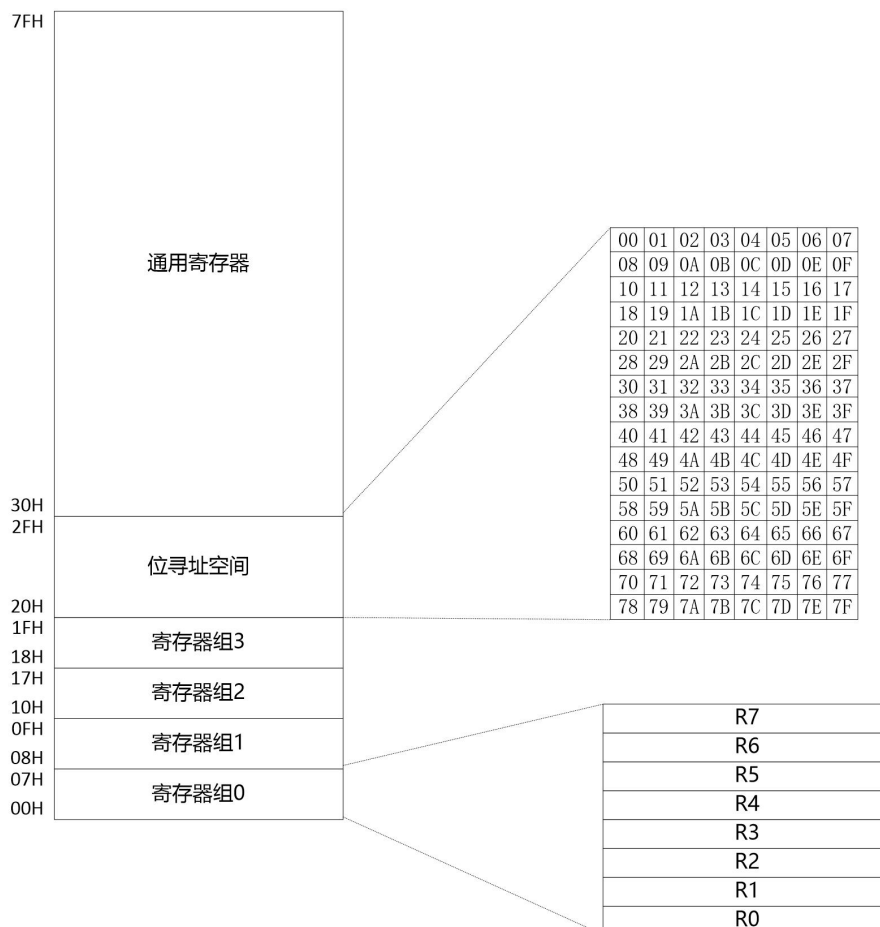


2.2 数据存储区

数据存储器包含 256 字节的内部数据存储器。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。



内部数据存储器



内部低 128 字节数据空间分配



2.3 FLASH 控制概述

本芯片内部实现了一个大小为 8KB 的 FLASH 存储器，包含 64 字节的 Information Block，编程次数可达 1000 次。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。

2.4 SFR

地址	0x8_	0x9_	0xA_	0xB_	0xC_	0xD_	0xE_	0xF_
0	-	-	P2	P1_DR	TIM1_CR	PSW	ACC	B
1	SP	SCR_CFG	I2C_ADDR	P1_DM0	TIM1_IE	-	-	-
2	DPL0	SCR_SLEEP	I2C_CR	P1_DM1	TIM1_SR	-	-	-
3	DPH0	SCR_CALI	I2C_STAT	-	-	-	-	-
4	DPL1	CLK_CR	I2C_DR	-	-	-	-	-
5	DPH1	PCLK_CR	I2C_MCR	-	-	-	-	-
6	DPS	PCLK_DIV12	-	-	-	-	-	-
7	-	PCLK_DIV3	-	-	-	-	-	-
8	SLPTIM_CR	P0_DR	IE	P2_DR	TIM2_CR	-	ADC_CR0	TIMO_CR
9	SLPTIM_SR	P0_DM0	-	P2_DM0	TIM2_IE	-	ADC_CR1	TIMO_CNTR
A	SLPTIM_CLR	P0_DM1	INT_MSK0	P2_DM1	TIM2_SR	-	ADC_CR2	TIMO_ARR
B	SLPTIM_WDT	-	INT_MSK1	-	-	-	ADC_CHSEL	TIMO_IE
C	SLPTIM_CNTRL	UART0_DR		UART1_DR	-	-	ADC_CON	TIMO_SR
D	SLPTIM_CNTH	UART0_CR	INT_PRI0	UART1_CR	-	-	ADC_DLY	SSCONR
E	SLPTIM_PRDRL	UART0_SR	INT_PRI1	UART1_SR	-	-	ADC_RESL	ADC_COMPL
F	SLPTIM_PRDRH	UART0_CFG	-	UART1_CFG	-	-	ADC_RESB	ADC_COMPH

2.5 XDATA

XC8F9362 中一部分寄存器放在外部数据存储 XDATA 空间，该部分地址空间大小 256 字节，地址范围 0xFF00~0xFFFF。

寄存器地址分配如下表（未标示地址为保留位）：

	0H/8H	1H/9H	2H/AH	3H/BH	4H/CH	5H/DH	6H/EH	7H/FH
FF88H	IM0_CR	-	-	-	-	-	-	-
FF80H	BG_CR	-	-	-	-	BORLVD_CR	BORLVD_STAT	-
FF68H	TIM2_CNTRL	TIM2_CNTRH	TIM2_ARRL	TIM2_ARRH	TIM2_GCMARL	TIM2_GCMARH	TIM2_GCMBRL	TIM2_GCMBRH
FF60H	TIM2_FCONR	TIM2_VPERR	TIM2_DTUA	TIM2_BRAKE	TIM2_DTR	TIM2_PCONRA	TIM2_PCONRB	-
FF58H	TIM1_CNTRL	TIM1_CNTRH	TIM1_ARRL	TIM1_ARRH	TIM1_GCMARL	TIM1_GCMARH	TIM1_GCMBRL	TIM1_GCMBRH
FF50H	TIM1_FCONR	TIM1_VPERR	TIM1_DTUA	TIM1_BRAKE	TIM1_DTR	TIM1_PCONRA	TIM1_PCONRB	-
FF40H	P2_FLAG	P2_GE	-	P2_PU	P2_PD	P2_IE	P2_IC0	P2_IC1
FF30H	P1_FLAG	P1_GE	-	P1_PU	P1_PD	P1_IE	P1_IC0	P1_IC1
FF20H	P0_FLAG	P0_GE	-	P0_PU	P0_PD	P0_IE	P0_IC0	P0_IC1
FF18H	PERP0_EN	PERP1_EN	-	-	-	-	-	-
FF10H	PT_SEL	-	-	-	-	-	-	-



FF00H	FLASH_GR	FLASH_CFG	FLASH_KEY	FLASH_ADL	FLASH_ADH	FLASH_PBUF	-	FLASH_DR
-------	----------	-----------	-----------	-----------	-----------	------------	---	----------



3. 功能描述

3.1 SFR 空间寄存器

3.1.1 SFR~0x81/SP (堆栈指针)

0X81	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SP	SP<7:0>							
读/写	R/W							
复位值	0	0	0	0	0	1	1	1

Bit<7:0>: 堆栈指针, 指向 IDATA 区域

3.1.2 SFR~0x82/DPL0 (DPTR0 寄存器的低 8 位)

0X82	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPL0	DPTR0<7:0>							
读/写	R/W							
复位值	0X00							

Bit<7:0>: 用于 DPTR0[7:0]

3.1.3 SFR~0x83/DPH0 (DPTR0 寄存器的高 8 位)

0X83	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPH0	DPTR0<15:8>							
读/写	R/W							
复位值	0X00							

Bit<7:0>: 用于 DPTR0[15:8]

3.1.4 SFR~0x84/DPL1 (DPTR1 寄存器的低 8 位)

0X84	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPL1	DPTR1<7:0>							
读/写	R/W							
复位值	0X00							

Bit<7:0>: 用于 DPTR1[7:0]



3.1.5 SFR~0x85/DPH1 (DPTR1 寄存器的高 8 位)

0X84	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPH1	DPTR1<15:8>							
读/写	R/W							
复位值	0X00							

Bit<7:0>: 用于 DPTR1 [15:8]

3.1.6 SFR~0x86/DPS (DPTR0/DPTR1 选择寄存器)

0X86	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPS	-	-	-	-	-	-	-	SEL
读/写	-	-	-	-	-	-	-	R/W
复位值	X	X	X	X	X	X	X	0

Bit<0>: DPTR0/DPTR1 选择位

0: 系统使用 DPTR0 寄存器

1: 系统使用 DPTR1 寄存器

3.1.7 SFR~0x88/SLPTIM_CR (睡眠计数器控制寄存器)

0X88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SLPTIMCR	SLPIE	-	WDTEN	RSV0	SLEEPDIS	SLPINTS[2:0]		
读/写	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	X	0	0	0	0	0	0

Bit<7>: SLPIE 中断使能位

1: 睡眠定时器中断使能

0: 睡眠定时器中断禁止

Bit<5>: WDTEN 使能位

1: 看门狗定时器使能

0: 看门狗定时器禁止

Bit<4>: RSV0 时钟选择位

1: 无效

0: 选择内部 32K 作为 WDT 的 32K 工作时钟

Bit<3>: SLEEPDIS 睡眠定时器使能位

1: 禁止睡眠定时器

0: 使能睡眠定时器

Bit<2:0>: SLPINTS2~SLPINTS0 睡眠定时器溢出中断时间



SLPINTS<2>	SLPINTS<1>	SLPINTS<0>	中断时间
0	0	0	4ms
0	0	1	8ms
0	1	0	16ms
0	1	1	32ms
1	0	0	256ms
1	0	1	512ms
1	1	0	1024ms
1	1	1	2048ms

注：实际定时时间会比上面描述的时间多一个 32K cycle 即 30us。

3.1.8 SFR~0x89/SLPTIM_SR (睡眠计数状态寄存器)

0X89	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SLPTIMSR	SLPEV	-	-	-	-	-	-	RSV
读/写	R/W	-	-	-	-	-	-	W
复位值	0	X	X	X	X	X	X	0

Bit<7>: SLPEV 睡眠计数器标志位

1: 睡眠计数器溢出

0: 睡眠计数器没有溢出

Bit<0>: RSV 该保留位只能写 0, 读为 0

3.1.9 SFR~0x8A/SLPTIM_CLR (看门狗清除寄存器)

0X8A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDT_CLR	SLPTIM_CLR<7:0>							
读/写	W	W	W	W	W	W	W	W
复位值	X	X	X	X	X	X	X	X

Bit<7:0>: 写任何值到该寄存器清除看门狗

3.1.10 SFR~0x8B/SLPTIM_WDT (看门狗计数器状态寄存器)

0X8B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDT	-	WDTOV	WDTCNTR		-	-	-	-
读/写	-	R	R/W	R/W	-	-	-	-
复位值	X	0	0	0	X	X	X	X



Bit<6>: 看门狗溢出标志位

1: 看门狗溢出

0: 看门狗没有溢出

Bit<5:4>: 看门狗计数器计数值, 只能通过写 0 清除。

3.1.11 SFR~0x8C/SLPTIM_CNTL (看门狗计数器计数值寄存器低 8 位)

0X8C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CNTRL	CNTRL<7:0>							
读/写	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: 看门狗计数器计数值低 8 位

3.1.12 SFR~0x8D/SLPTIM_CNTH (看门狗计数器计数值寄存器高 8 位)

0X8D	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CNTRH	CNTRH							
读/写	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: 看门狗计数器计数值高 8 位

3.1.13 SFR~0x8E/SLPTIM_PRDL (睡眠计数器预分频寄存器低 8 位)

0X8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SLPTIMPRDL	ACCPDRDL<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: 睡眠定时器溢出值低 8 位

3.1.14 SFR~0x8F/SLPTIM_PRDRH (睡眠计数器预分频寄存器高 8 位)

0X8F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SLPTIMPRDRH	ACCSEL	-	-	-	-	ACCPDRH		
读/写	R/W	-	-	-	-	R/W	R/W	R/W
复位值	0	0	X	X	X	0	0	0

Bit<7>: 睡眠定时器溢出选择位

1: 选择睡眠定时器溢出值为 11 位可配置



0: 选择睡眠定时器溢出值为固定值

Bit<2:0>: 睡眠定时器高 3 位

注: 实际定时时间会比上面描述的时间多一个 32K cycle 即 30us。

3.1.15 SFR~0x91/SCR_CFG (系统配置寄存器)

0X91	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CFG	-	-	-	-	-	RSTREQ	-	-
读/写	R	-	-	-	-	W	R	R
复位值	0	X	X	X	X	1	0	0

Bit<2>: 软件复位使能位

1: 复位系统 (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)

0: 不复位系统

3.1.16 SFR~0x92/SCR_SLEEP (睡眠寄存器)

0X92	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SLEEP	-	-	-	-	-	-	SLEEPDEEP	SLEEP
读/写	-	-	-	-	-	-	R/W	R/W
复位值	X	X	X	X	X	X	0	0

Bit<1>: 深度睡眠模式控制位

1: 深度睡眠模式打开

0: 深度睡眠模式关闭

Bit<0>: 睡眠模式控制位

1: 睡眠模式

0: 正常工作模式

3.1.17 SFR~0x93/SCR_CALI (校准寄存器)

0X93	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CALI	CALI_WDR	CALI_XRES	-	-	-	-	-	-
读/写	R/W	R/W	-	-	-	-	-	-
复位值	0	1	X	X	X	0	1	1

Bit<7>: 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引脚复位、POR、BOR、写 1 来实现)

1 看门狗复位发生



写 1 清零 CALI_XRES, CALI_WDR

Bit<6>: 引脚复位标志

- 0 没有外部引脚复位发生 (该寄存器的清零可以通过看门狗复位、POR、BOR、CALI_WDR 写 1 来实现)
- 1 外部引脚复位发生

3.1.18 SFR~0x94/CLK_CR (系统时钟控制寄存器)

0X94	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCR	SCK3IF	-	-	-	-	CPUCKS<2:0>		
读/写	R/W	-	-	-	-	R/W	R/W	R/W
复位值	1	X	X	X	X	0	1	1

Bit<7>: SCK3 中断标志位

- 1: 有 SCK3 中断发生
- 0: 没有 SCK3 中断发生

Bit<2:0>: 内核工作频率选择位

SLPINTS<2>	SLPINTS<1>	SLPINTS<0>	中断时间
0	0	0	SYSCLK/8
0	0	1	SYSCLK/4
0	1	0	SYSCLK/2
0	1	1	SYSCLK
1	0	0	SYSCLK/16
1	0	1	SYSCLK/32
1	1	0	SYSCLK/64
1	1	1	SYSCLK/128

注: 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。

3.1.19 SFR~0x95/PCLK_CR (外设时钟控制寄存器)

0X95	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCLKCR	SCK0EN	SCK1EN	SCK2EN	SCK3EN	SCK3_IE	SCK2SS	SCK3SS<1:0>	
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	0	0	0	1

Bit<7>: SCK0 时钟使能位

- 1: 使能
- 0: 禁止



Bit<6>: SCK1 时钟使能位

1: 使能

0: 禁止

Bit<5>: SCK2 时钟使能位

1: 使能

0: 禁止

Bit<4>: SCK3 时钟使能位

1: 使能

0: 禁止

Bit<3>: SCK3 时钟中断使能位

1: 使能

0: 禁止

Bit<2>: SCK2 时钟源选择位, 具体使用见 SCK2CKS 说明

1: SCK1 作为 SCK2 的时钟源

0: CLK_SYS 作为 SCK2 的时钟源

Bit<1:0>: SCK3 时钟源选择位

SCK3SS<1>	SCK3SS<0>	时钟源
0	0	关闭 SCK3 时钟
0	1	来自 SYSCLK
1	0	来自 SCK1 时钟
1	1	来自 SCK2 时钟

3.1.20 SFR~0x96/PCLK_DIV12 (SCK1、SCK2 时钟控制寄存器)

0x96	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DIV12	SCK1CKS				SCK2CKS			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	1	1	1	1

Bit<7:4>: 控制 SCK1 时钟分频

计算方法: $f_{SCK1} = f_{SYSCLK} / (SCK1CKS + 1)$

Bit<3:0>: 控制 SCK2 时钟分频

计算方法: SCK2SS=0 时 $f_{SCK2} = f_{SYSCLK} / (SCK2CKS + 1)$

计算方法: SCK2SS=1 时 $f_{SCK2} = f_{SYSCLK} / (SCK2CKS + 1) / (SCK1CKS + 1)$



3.1.21 SFR~0x97/PCLK_DIV3 (SCK3 时钟控制寄存器)

0X97	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCK3CKS	SCK3CKS<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	1	1	0	0	0	1

Bit<7:0>: 控制 SCK3 时钟的分频, 频率和 SCK3SS 的值相关

SCK3SS<1:0>		计算方法
0	0	关闭 SCK3 时钟
0	1	$f_{SCK3} = f_{SYSCLK} / (SCK3CKS + 1)$
1	0	$f_{SCK3} = f_{SYSCLK} / (SCK3CKS + 1) / (SCK1CKS + 1)$
1	1	$f_{SCK3} = f_{SYSCLK} / (SCK3CKS + 1) / (SCK2CKS + 1)$

3.1.22 SFR~0x98/P0_DR (端口 0 数字寄存器)

0X98	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PODR	P0<7>	P0<6>	P0<5>	P0<4>	P0<3>	P0<2>	P0<1>	P0<0>
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: 端口 P0 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

3.1.23 SFR~0x99/P0_DM0 (端口 0 模式 0 位)

0X99	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PODM0	P0_DM0<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: P0 模式控制寄存器

3.1.24 SFR~0x9A/P0_DM1 (端口 1 模式 1 位)

0X9A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PODM1	P0_DM1<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: P0 模式控制寄存器



3.1.25 SFR~0x9C/UART0_DR (UART 数据寄存器)

0X9C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UART0DR	DATA<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: 发送模式该寄存器只能写。该寄存器只能在 UART 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。

3.1.26 SFR~0x9D/UART0_CR (UART 控制寄存器)

0X9D	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UART0CR	IE	R_EN	-	PSEL	PAR_ODD	PAR_EN	T_EN	EN
读/写	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
复位值	0	0	X	0	0	0	0	0

Bit<7>: 发送完成或接收完成中断控制位

1: 使能

0: 禁止

Bit<6>: 接收模式使能位

1: 使能

0: 禁止

Bit<4>: UART0 的 TX 和 RX 信号交换位

1: P0.3 作为 RXD, P0.4 作为 TXD

0: P0.3 作为 TXD, P0.4 作为 RXD

Bit<3>: 奇偶校验选择位

1: 奇校验 (必须使能奇偶校验, 校验才会生效)

0: 偶校验 (必须使能奇偶校验, 校验才会生效)

Bit<2>: 奇偶校验控制位

1: 使能

0: 关闭

注: 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

Bit<1>: 发送模式控制位

1: 使能

0: 关闭

Bit<0>: 模块功能控制位

1: 使能



0: 关闭

3.1.27 SFR~0x9E/UART0_SR (UART 状态寄存器)

0X9E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UART0SR	RXFULL	RXACTIVE	ERRFRAME	ERRPAR	OVERRUN	-	-	TXCOMPLETE
读/写	R	R	R/W	R/W	R/W	-	-	R/W
复位值	0	0	0	0	0	X	X	0

Bit<7>: 数据接收标志位

1: 接收到数据 (读数据寄存器会清该标志位。发送模式下该位常为 0)

0: 没有接收到数据

Bit<6>: 接收数据标志位

1: 正在接收数据

0: 没有接收数据 (发送模式下该位常为 0)

Bit<5>: 发生帧错误标志位

1: 有发生帧错误

0: 没有发生帧错误

该位只有在接收模式下有效, 接收数据时如果停止位收到低电平会触发帧错误;

发送模式下该位常为 0。写 1 清零

Bit<4>: 奇偶校验错误标志位

1: 发生奇偶校验错误

0: 没有发生奇偶校验错误

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

Bit<3>: 接收超限标志位

1: 接收超限

0: 没有接收超限

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。

Bit<0>: 发送完成标志位

1: 发送完成

0: 发送没有完成

发送模式下, 如果发送完成会将该位置 1。

接收模式下常为 0。写 1 清零该位。



3.1.28 SFR~0x9F/UART0_CFG (UART 配置寄存器)

0X9F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UART0CFG	-	-	-	-	-	-	CKSEL<1:0>	
读/写	-	-	-	-	-	-	R/W	R/W
复位值	X	X	X	X	X	X	0	0

Bit<0>: UART 时钟选择位

CKSEL<1:0>		UART 时钟源
0	0	选择 SCK1
0	1	选择 SCK2
1	X	选择 SCK3

3.1.29 SFR~0xA0/P2 (P2 读写寄存器)

0XA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2	P2<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	1	1	0	0	1	1	0

Bit<7:0>: 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。

3.1.30 SFR~0xA1/I2C_ADDR (I2C 从机地址寄存器)

0XA1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CADDR	HWADDREN	SLAVE ADDRESS<6:0>						
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	1	1	0	0	1	1	0

Bit<7>: 地址比较功能控制位

1: 使能

0: 禁止

Bit<6: 0>: 只用于从模式, 当前设备的地址

注: 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。



3.1.31 SFR~0xA2/I2C_CR (I2C 控制寄存器)

OXA2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CONT	I2CIE	-	BUSERROR	STOPIE	-	CLKSEL	ENABLEMASTER	ENABLESLAVE
读/写	R/W	-	R/W	R/W	-	R/W	R/W	R/W
复位值	0	X	0	0	X	0	0	1

Bit<7>: I2C 全部中断控制位

- 1: 使能
- 0: 禁止

Bit<5>: Bus Error 中断控制位

- 1: 使能
- 0: 禁止

Bit<4>: 结束中断控制位

- 1: 使能
- 0: 禁止

Bit<2>: I2C 时钟选择位

- 1: SCK2
- 0: SCK1

Bit<1:0>: 主模式 & 从模式控制位

ENABLE MASTER	ENABLE SLAVE	模式状态	
		主模式	从模式
0	0	关	关
0	1	关	开
1	0	开	关
1	1	开	开

3.1.32 SFR~0xA3/I2C_STAT (I2C 状态寄存器)

OXA3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSTAT	BUSERROR	LOSTARB	STOPSTATS	ACK	ADDRESS	TRANSMIT	LRB	TRANSCOMPLETE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: 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: 若发生了 Bus Error, 则需要配置成非主机模式或关掉 I2C。

Bit<6>: 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: 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。



Bit<5>: 检测到结束状态时置 1 (只能通过写 0 清除)

Bit<4>: 发送 ACK 控制位

1: 使能

0: 禁止 (NACK)

Bit<3>: 收到一个地址时置 1 (只能通过写 0 清除)

Bit<2>: 模式状态标志位

1: 发送模式

0: 接收模式

Bit<1>: 是否应答状态标志位

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)

Bit<0>: 单字节方式状态位

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;

0: 未完成 (写 0 清除或者检测到 START 信号清除)

3.1.33 SFR~0xA4/I2C_DR (I2C 数据寄存器)

0xA4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CDR	I2C_DR<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: 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。

3.1.34 SFR~0xA5/I2C_MCR (I2C 主机控制寄存器)

0xA5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMCR	-	-	-	-	BUSBUSY	MASTERMODE	RESTARTGEN	STARTGEN
读/写	-	-	-	-	R	R	R/W	R/W
复位值	X	X	X	X	0	0	0	0

Bit<3>: 检测信号状态标志位

1: 检测到开始信号

0: 检测到结束信号

Bit<2>: 产生信号状态标志位



1: 产生开始信号

0: 产生结束信号

Bit<1>: 1 传送过程中收到响应为 NACK, 重启传送过程, 重新传送

Bit<0>: 1 产生开始信号并发送地址到 i2c 总线上, 传送完成后清零

3.1.35 SFR~0xA8/IE (系统中断使能寄存器)

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	IE_EA	-	-	-	-	-	-	-
读/写	R/W	-	-	-	-	-	-	-
复位值	0	X	X	X	X	X	X	X

Bit<7>: CPU 中断允许位总开关

1: 使能中断

0: 禁止中断

3.1.36 SFR~0xAA/INT_MSK0 (中断屏蔽寄存器 0)

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTMSK0	T2MSK	T1MSK	T0MSK	SCK3MSK	P2MSK	P1MSK	P0MSK	LVDMSK
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: Timer2 中断控制位

1: 禁止

0: 使能

Bit<6>: Timer1 中断控制位

1: 禁止

0: 使能

Bit<5>: Timer0 中断控制位

1: 禁止

0: 使能

Bit<4>: SCK3 中断控制位

1: 禁止

0: 使能

Bit<3>: GPIO 2 中断控制位

1: 禁止

0: 使能

Bit<2>: GPIO 1 中断控制位



1: 禁止

0: 使能

Bit<1>: GPIO 0 中断控制位

1: 禁止

0: 使能

Bit<0>: LVD 中断控制位

1: 禁止

0: 使能

3.1.37 SFR~0xAB/INT_MSK1 (中断屏蔽寄存器 1)

0xAB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTMSK1	-	-	-	WDTMSK	UART1MSK	UART0MSK	I2CMSK	ADCMSK
读/写	-	-	-	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	0	0	0	0	0

Bit<4>: WDT 中断控制位

1: 禁止

0: 使能

Bit<3>: UART1 中断控制位

1: 禁止

0: 使能

Bit<2>: UART0 中断控制位

1: 禁止

0: 使能

Bit<1>: I2CMSK 中断控制位

1: 禁止

0: 使能

Bit<0>: ADC 中断控制位

1: 禁止

0: 使能



3.1.38 SFR~0xAD/INT_PRI0 (中断优先级配置寄存器 0)

0xAD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTPRI0	T2PRI	T1PRI	T0PRI	SCK3PRI	P2PRI	P1PRI	POPRI	LVDPRI
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: Timer2 中断优先级控制位

- 1: 高优先级
- 0: 低优先级

Bit<6>: Timer1 中断优先级控制位

- 1: 高优先级
- 0: 低优先级

Bit<5>: Timer0 中断优先级控制位

- 1: 高优先级
- 0: 低优先级

Bit<4>: SCK3 中断优先级控制位

- 1: 高优先级
- 0: 低优先级

Bit<3>: GPIO 2 中断优先级控制位

- 1: 高优先级
- 0: 低优先级

Bit<2>: GPIO 1 中断优先级控制位

- 1: 高优先级
- 0: 低优先级

Bit<1>: GPIO 0 中断优先级控制位

- 1: 高优先级
- 0: 低优先级

Bit<0>: LVD 中断优先级控制位

- 1: 高优先级
- 0: 低优先级

3.1.39 SFR~0xAE/INT_PRI1 (中断优先级配置寄存器 1)

0xAE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTPRI1	-	-	-	WDTPRI	UART1PRI	UART0PRI	I2CPRI	ADCPRI
读/写	-	-	-	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	0	0	0	0	0



Bit<4>: WDT 中断优先级控制位

1: 高优先级

0: 低优先级

Bit<3>: UART1 中断优先级控制位

1: 高优先级

0: 低优先级

Bit<2>: UART0 中断优先级控制位

1: 高优先级

0: 低优先级

Bit<1>: I2C 中断优先级控制位

1: 高优先级

0: 低优先级

Bit<0>: ADC 中断优先级控制位

1: 高优先级

0: 低优先级

3.1.40 SFR~0xB0/P1_DR (端口 1 数据寄存器)

0XB0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1DR	P1DR<7>	P1DR<6>	P1DR<5>	P1DR<4>	P1DR<3>	P1DR<2>	P1DR<1>	P1DR<0>
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	1	1	0	0

Bit<7:0>: 端口 P1 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

3.1.41 SFR~0xB1/P1_DM0 (端口 1 模式 0 位)

0XB1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1DM0	P1_DM0<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	1	1	0	0

Bit<7:0>: P1 模式控制寄存器



3.1.42 SFR~0xB2/P1_DM1 (端口 1 模式 1 位)

0XB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1DM1	P1_DM1<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	1	1	0	0

Bit<7:0>: P1 模式控制寄存器

3.1.43 SFR~0xB8/P2_DR (端口 2 数据寄存器)

0XB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2DR	P2DR<7>	P2DR<6>	P2DR<5>	P2DR<4>	P2DR<3>	P2DR<2>	P2DR<1>	P2DR<0>
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: 端口 P2 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

3.1.44 SFR~0xB9/P2_DM0 (端口 2 模式 0 位)

0XB9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2DM0	P2_DM0<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: P2 模式控制寄存器

3.1.45 SFR~0xBA/P2_DM1 (端口 2 模式 1 位)

0XBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2DM1	P2_DM1<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: P2 模式控制寄存器



3.1.46 SFR~0xBC/UART1_DR (UART1 数据寄存器)

0xBC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UARTDR	DATA<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: 发送模式该寄存器只能写。该寄存器只能在 UART 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。

3.1.47 SFR~0xBD/UART1_CR (UART 控制寄存器)

0xBD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UART1CR	IE	R_EN	-	PSEL	PAR_ODD	PAR_EN	T_EN	EN
读/写	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
复位值	0	0	X	0	0	0	0	0

Bit<7>: 发送完成或接收完成中断控制位

1: 使能

0: 禁止

Bit<6>: 接收模式使能位

1: 使能

0: 禁止

Bit<4>: UART0 的 TX 和 RX 信号交换位

1: P2.1 作为 RXD, P1.0 作为 TXD

0: P2.1 作为 TXD, P1.0 作为 RXD

Bit<3>: 奇偶校验选择位

1: 奇校验 (必须使能奇偶校验, 校验才会生效)

0: 偶校验 (必须使能奇偶校验, 校验才会生效)

Bit<2>: 奇偶校验控制位

1: 使能

0: 关闭

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

Bit<1>: 发送模式控制位

1: 使能

0: 关闭

Bit<0>: 模块功能控制位

1: 使能



0: 关闭

3.1.48 SFR~0xBE/UART1_SR (UART 状态寄存器)

0xBE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UART1SR	RXFULL	RXACTIVE	ERRFRAME	ERRPAR	OVERRUN	-	-	TXCOMPLETE
读/写	R	R	R/W	R/W	R/W	-	-	R/W
复位值	0	0	0	0	0	X	X	0

Bit<7>: 数据接收标志位

- 1: 接收到数据 (读数据寄存器会清该标志位。发送模式下该位常为 0)
- 0: 没有接收到数据

Bit<6>: 接收数据标志位

- 1: 正在接收数据
- 0: 没有接收数据 (发送模式下该位常为 0)

Bit<5>: 发生帧错误标志位

- 1: 有发生帧错误
 - 0: 没有发生帧错误
- 该位只有在接收模式下有效, 接收数据时如果停止位收到低电平会触发帧错误; 发送模式下该位常为 0。写 1 清零

Bit<4>: 奇偶校验错误标志位

- 1: 发生奇偶校验错误
 - 0: 没有发生奇偶校验错误
- 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

Bit<3>: 接收超限标志位

- 1: 接收超限
 - 0: 没有接收超限
- 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。

Bit<0>: 发送完成标志位

- 1: 发送完成
 - 0: 发送没有完成
- 发送模式下, 如果发送完成会将该位置 1。
接收模式下常为 0。写 1 清零该位。



3.1.49 SFR~0xBF/UART1_CFG (UART 配置寄存器)

0xBF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UART1CFG	-	-	-	-	-	-	CKSEL<1:0>	
读/写	-	-	-	-	-	-	R/W	R/W
复位值	X	X	X	X	X	X	0	0

Bit<0>: UART 时钟选择位

CKSEL<1:0>		UART 时钟源
0	0	选择 SCK1
0	1	选择 SCK2
1	X	选择 SCK3

3.1.50 SFR~0xC0/TIM1_CR (Timer1 控制寄存器)

0xC0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1CR	-	-	ADCFILTEREN	SELSREG	DIR	MODE<1:0>		TIM1_EN
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	0	0	0	0	0	0

Bit<5>: 刹车输入 ADC 比较输出滤波和 BKIN 管脚输入滤波控制位

- 1: 刹车输入 ADC 比较输出和 BKIN 管脚开数字滤波
- 0: 刹车输入 ADC 比较输出和 BKIN 管脚关数字滤波

注: TIMER1/2 共用, 只在 TIMER1 设定, TIMER2 共用 TIMER1 设定

Bit<4>: 影子寄存器控制位

- 1: ARR GCMAR GCMBR 读到当前设定的值
- 0: ARR GCMAR GCMBR 读到影子寄存器的值或捕获值

Bit<3>: 计数器计数方向

- 1: 向下计数
- 0: 向上计数

Bit<2:1>: 计数器计数模式

MODE<1:0>		计数模式
0	0	锯齿波计数模式
0	1	三角波计数模式
1	X	保留

Bit<0>: TIMER1 使能控制位

- 1: 使能 TIMER1
- 0: 关闭 TIMER1



3.1.51 SFR~0xC1/TIM1_IE (Timer1 中断控制寄存器)

0XC1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1IE	-	-	BRAKEB_IE	BRAKEA_IE	CMPB_IE	CMPA_IE	UD_IE	OV_IE
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	0	0	0	0	0	0

Bit<5>: TIM1_CHB 刹车中断使能控制位

1: 使能

0: 禁止

Bit<4>: IM1_CHA 刹车中断使能控制位

1: 使能

0: 禁止

Bit<3>: TIM1_CHB 比较或者捕获中断使能控制位

1: 使能

0: 禁止

Bit<2>: TIM1_CHA 比较或者捕获中断使能控制位

1: 使能

0: 禁止

Bit<1>: 下溢中断使能控制位

1: 使能

0: 禁止

Bit<0>: 上溢中断使能控制位

1: 使能

0: 禁止

3.1.52 SFR~0xC2/TIM1_SR (Timer1 状态寄存器)

0XC2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1SR	-	-	BRAKEB_IF	BRAKEA_IF	CMPB_IF	CMPA_IF	UD_IF	OV_IF
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	0	0	0	0	0	0

Bit<5>: TIM1_CHB 刹车中断标志位

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

Bit<4>: TIM1_CHA 刹车中断标志位

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



Bit<3>: TIM1_CHB 比较或者捕获中断标志位

- 1: 发生 TIM1_CHB 比较匹配或者捕获
- 0: 未发生 TIM1_CHB 比较匹配或者捕获

Bit<2>: TIM1_CHA 比较或者捕获中断标志位

- 1: 发生 TIM1_CHA 比较匹配或者捕获
- 0: 未发生 TIM1_CHA 比较匹配或者捕获

Bit<1>: TIMER1 计数器下溢中断标志位

- 1: 计数器发生下溢
- 0: 计数器未发生下溢

Bit<0>: TIMER1 计数器上溢中断标志位

- 1: 计数器发生上溢
- 0: 计数器未发生上溢

注: 写 1 清零以上标志位

3.1.53 SFR~0xC8/TIM2_CR (Timer2 控制寄存器)

0XC8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2SR	-	-	CAP_TIM1	SEL_SREG	DIR	MODE<1:0>		TIM2_EN
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	0	0	0	0	0	0

Bit<5>: TIMER2 捕获 TIMER1 控制位

- 1: TIMER2 捕获 TIMER1
- 0: TIMER2 不捕获 TIMER1

Bit<4>: 影子寄存器控制位

- 1: ARR GCMAR GCMBR 读到当前设定的值
- 0: ARR GCMAR GCMBR 读到影子寄存器的值或捕获值

Bit<3>: 计数器计数方向控制位

- 1: 向下计数
- 0: 向上计数

Bit<2:1>: 计数器计数模式控制位

MODE<1:0>		计数模式
0	0	锯齿波计数模式
0	1	三角波计数模式
1	X	保留

Bit<0>: TIMER2 使能控制位

- 1: 使能
- 0: 禁止



3.1.54 SFR~0xC9/TIM2_IE (Timer2 中断控制寄存器)

0XC9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2IE	-	-	BRAKEB_IE	BRAKEA_IE	CMPB_IE	CMPA_IE	UD_IE	OV_IE
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	0	0	0	0	0	0

Bit<5>: TIM2_CHB 刹车中断使能控制位

- 1: 使能
- 0: 禁止

Bit<4>: TIM2_CHA 刹车中断使能控制位

- 1: 使能
- 0: 禁止

Bit<3>: TIM2_CHB 比较或者捕获中断使能控制位

- 1: 使能
- 0: 禁止

Bit<2>: TIM2_CHA 比较或者捕获中断使能控制位

- 1: 使能
- 0: 禁止

Bit<1>: 下溢中断使能控制位

- 1: 使能
- 0: 禁止

Bit<0>: 上溢中断使能控制位

- 1: 使能
- 0: 禁止

3.1.55 SFR~0xCA/TIM2_SR (Timer2 状态寄存器)

0XCA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2SR	-	-	BRAKEB_IF	BRAKEA_IF	CMPB_IF	CMPA_IF	UD_IF	OV_IF
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	0	0	0	0	0	0

Bit<5>: TIM2_CHB 刹车中断标志位

- 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

Bit<4>: TIM2_CHA 刹车中断标志位

- 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



Bit<3>: TIM2_CHB 比较或者捕获中断标志位

1: 发生 TIM2_CHB 比较匹配或者捕获

0: 未发生 TIM2_CHB 比较匹配或者捕获

Bit<2>: TIM2_CHA 比较或者捕获中断标志位

1: 发生 TIM2_CHA 比较匹配或者捕获

0: 未发生 TIM2_CHA 比较匹配或者捕获

Bit<1>: TIMER2 计数器下溢中断标志位

1: 计数器发生下溢

0: 计数器未发生下溢

Bit<0>: TIMER2 计数器上溢中断标志位

1: 计数器发生上溢 (写 1 清零以上标志位)

0: 计数器未发生上溢

3.1.56 SFR~0xD0/PSW (程序状态字寄存器)

0XD0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	CY	AC	F0	RS<1:0>		OV	F1	P
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: CY 进位标志位

Bit<6>: AC 辅助进位标志位

Bit<5>: F0 通用标志位 0

Bit<4:3>: 寄存器组选择

RS<1:0>		寄存器组	数据地址
0	0	寄存器组 0	0x00 - 0x07
0	1	寄存器组 1	0x08 - 0x0F
1	0	寄存器组 2	0x10 - 0x17
1	1	寄存器组 3	0x18 - 0x1F

Bit<2>: OV 溢出标志位

Bit<1>: F1 通用标志位 1

Bit<0>: P 奇偶校验标志位

3.1.57 SFR~0xE0/ACC (累加寄存器)

0XE0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ACC	ACC<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0



Bit<7:0>: ACC 累加寄存器

3.1.58 SFR~0xE8/ADC_CR0 (ADC 转换控制寄存器 0)

0XE8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCRO	ADC_EN	-	ADC_START	ADC_IF	ADC_IE	ADC_EX	CLKSEL	
读/写	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	X	0	0	0	0	0	0

Bit<7>: ADC 使能位

- 1: 使能 ADC 转换电路
- 0: 禁止 ADC 转换电路

Bit<5>: ADC 软件触发采样控制位

- 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

注: ADC 使能后，写 1 后开始 ADC 转换，转换完成后硬件自动将此位清零

Bit<4>: ADC 转换结束标志位

- 1: 转换完成
- 0: 未完成

注: 当 ADC 完成一次转换后，硬件会自动将此位置 1，并向 CPU 发出中断请求。此标志位必须由软件写 1 清零或复位清零

Bit<3>: ADC 中断使能控制位

- 1: 使能
- 0: 禁止

Bit<2>: 启动 ADC 的触发条件选择位

- 1: 硬件触发
- 0: 软件触发

Bit<1:0>: ADC 时钟分频选择位

CLKSEL<1:0>		ADC 时钟
0	0	系统时钟的 4 分频
0	1	系统时钟的 8 分频
1	0	系统时钟的 16 分频
1	1	系统时钟的 32 分频

注: 修改 CLK_SEL 寄存器配置必须在 ADC_EN 为 0 的时候进行



3.1.59 SFR~0xE9/ADC_CR1 (ADC 转换控制寄存器 1)

0XE9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCR1	-	ETGSEL<1:0>		ETGTYP<1:0>		SCSEL<2:0>		
读/写	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	0	0	0	0	0	0	1

Bit<6:5>: 外部触发源选择位 (当 ADCEX 为 1 时, 该位选择外部触发 ADC 的来源)

ETGSEL<1:0>		外部触发源
0	0	timer1 的 PWM 事件触发 (默认)
0	1	timer2 的 PWM 事件触发
1	1	外部 PIN 脚 adc_etr 触发

注: timer1/2 用于触发 ADC 采样的 PWM 输出通道为 CHA 通道。

Bit<4:3>: 外部触发信号类型选择位 (当 ADCEX 置 1 时该位决定响应外部触发的类型)

ETGTYP<1:0>		触发类型
0	0	下降沿触发 (默认)
0	1	上升沿触发
1	0	一个 PWM 周期的中点
1	1	一个 PWM 周期的终点

注: adc_etr 触发只能选用边沿触发; PWM 周期中点或终点仅适用于三角波模式的 PWM 输出

Bit<2:0>: ADC 采样时间周期选择位 (片外很高的输入阻抗时, 增加采样时间, 提高转换精度)

SCSEL<2:0>			时间周期
0	0	0	4 个 ADC 时钟周期
0	0	1	8 个 ADC 时钟周期 (默认)
0	1	0	16 个 ADC 时钟周期
0	1	1	32 个 ADC 时钟周期
1	0	0	64 个 ADC 时钟周期
1	0	1	128 个 ADC 时钟周期

3.1.60 SFR~0xEA/ADC_CR2 (ADC 转换控制寄存器 2)

0XEA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCR2	-	-	CTRL5	CTRL4	CTRL3	CTRL2	CTRL<1:0>	
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	0	0	0	0	1	1

Bit<5>: 参考低噪声使能配置位

1: 参考噪声减低

0: 正常工作模式

Bit<4>: 参考测试模式控制位

1: 测试模式



0: 正常模式

Bit<3>: 参考 buffer 增益选择位

1: 参考 buffer 输出是参考的 1 倍

0: 参考 buffer 输出是参考的 2 倍

Bit<2>: 参考 buffer 输入选择位

1: 选择外部参考电压

0: 选择内部参考 Bandgap 电压

Bit<1:0>: ADC 参考电压选择位

CLRTL<1:0>		参考电压
0	0	选择片外电压不通过 buffer, 直接做 ADC 参考电压
0	1	保留
1	0	选择 VDD, 做 ADC 参考电压
1	1	选择 buffer 输出做 ADC 参考电压

3.1.61 SFR~0xEB/ADC_CHSEL (ADC 模拟量输入通道选择寄存器)

0xEB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCHEL	-	-	-	-	CHANNEL_SEL<3:0>			
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位值	X	X	X	X	1	0	0	0

Bit<4:0>: ADC 模拟量输入通道选择位

CHANNEL_SEL<3:0>				通道
0	0	0	0	AIN0
0	0	0	1	AIN1
0	0	1	0	AIN2
0	0	1	1	AIN3
0	1	0	0	AIN4
0	1	0	1	AIN5
0	1	1	0	AIN6
0	1	1	1	AIN7
1	0	0	0	片内电源输入通道(默认)

3.1.62 SFR~0xEC/ADC_CON (ADC 配置寄存器)

0xEC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCON	ADFBEN	ADCMPOP	ADCMPEM	ADCMPO	-	-	-	ADCDLY8
读/写	R/W	R/W	R/W	R	-	-	-	R/W
复位值	0	0	0	0	X	X	X	0

Bit<7>: ADC 比较结果响应故障刹车使能控制位



1: 使能

0: 关闭

Bit<6>: ADC 比较器输出极性选择位

1: 若 ADC 输出值小于设定的比较值, 则 ADCMP0 为 1

0: 若 ADC 输出值大于或等于设定的比较值, 则 ADCMP0 为 1

Bit<5>: ADC 结果比较使能控制位

1: 使能

0: 关闭

Bit<4>: ADC 比较结果输出位

Bit<0>: ADC 外部触发延时计数器数值的高 1 位

3.1.63 SFR~0xED/ADC_DLY (ADC 触发延迟配置寄存器)

0xED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCDLY	ADC_DLP<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	0	0	0	0

Bit<7:0>: ADC 外部触发启动延迟计数器的低 8 位

3.1.64 SFR~0xEE/ADC_RESL (ADC 转换结果低位寄存器)

0xEE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCRESL	-	-	-	-	ADC_RESL<3:0>			
读/写	-	-	-	-	R	R	R	R
复位值	X	X	X	X	0	0	0	0

Bit<3:0>: ADC 转换结果低 4 位

3.1.65 SFR~0xEF/ADC_RESB (ADC 转换结果高位寄存器)

0xEF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCRESB	ADC_RESB<7:0>							
读/写	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: ADC 转换结果高 8 位



3.1.66 SFR~0xF0/B (B 寄存器)

0XF0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
B	B<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: 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。

3.1.67 SFR~0xF8/TIMO_CR (Timer0 控制寄存器)

0XF8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIMOCR	-	-	TIMO_CLKSEL<1:0>		TIMO_CLKDIV<2:0>			TIMO_EN
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	0	0	0	0	0	0

Bit<5:4>: TIMER0 时钟选择位

TIMO_CLKSEL<1:0>		时钟
0	0	SCK0
0	1	内部 32K 时钟
1	0	保留
1	1	保留

Bit<3:1>: TIMER0 预分频选择位

TIMO_CLKDIV<2:0>			预分频
0	0	0	1 分频
0	0	1	2 分频
0	1	0	4 分频
0	1	1	8 分频
1	0	0	16 分频
1	0	1	32 分频
1	1	0	64 分频
1	1	1	128 分频

Bit<0>: TIMO_EN 使能控制位

1: 使能

0: 关闭

注: 修改 TIMO_CLKSEL 和 TIMO_CLKDIV 寄存器配置必须在 TIMO_EN 为 0 的时候进行。



3.1.68 SFR~0xF9/TIMO_CNTR (Timer0 计数值寄存器)

0XF9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIMOCNTR	TIMO_CNTR<7:0>							
读/写	R							
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIMER0 计数值寄存器

3.1.69 SFR~0xFA/TIMO_ARR (Timer0 自动重装寄存器)

0XFA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIMOARR	TIMO_ARR<7:0>							
读/写	R/W							
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIMER0 自动重装寄存器

3.1.70 SFR~0xFB/TIMO_IE (Timer0 中断控制寄存器)

0XFB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIMO_IE	-	-	-	-	-	-	-	TIMOTCIE
读/写	-	-	-	-	-	-	-	R/W
复位值	X	X	X	X	X	X	X	0

Bit<0>: 溢出中断使能控制位

1: 使能

0: 禁止

3.1.71 SFR~0xFC/TIMO_SR (Timer0 状态寄存器)

0XFC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIMO_SR	-	-	-	-	-	-	-	TIMO_TC
读/写	-	-	-	-	-	-	-	R/W
复位值	X	X	X	X	X	X	X	0

Bit<0>: 定时器 TIMER0 溢出标志位

1: 发生溢出

0: 未发生溢出



3.1.72 SFR~0xFD/SSCONR (Timer1/2 软件同步控制寄存器)

0XFD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSCONR	-	-	-	-	SSREQ2<1:0>		SSREQ1<1:0>	
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位值	X	X	X	X	0	0	0	0

Bit<3:2>: TIMER2 操作控制位 (写 0 无效; 读出值为 0)

SSREQ2<1:0>		TIMER2 状态
0	1	TIMER2 开始计数
1	0	TIMER2 停止计数; 此时输出使用 GPIO 配置
1	1	TIMER2 暂停计数, 计数值保持; 此时输出保持前一状态

Bit<1:0>: TIMER1 操作控制位 (写 0 无效; 读出值为 0)

SSREQ1<1:0>		TIMER1 状态
0	1	TIMER1 开始计数
1	0	TIMER1 停止计数; 此时输出使用 GPIO 配置
1	1	TIMER1 暂停计数, 计数值保持; 此时输出保持前一状态

3.1.73 SFR~0xFE/ADC_COMPL (ADC 比较值低 4 位)

0XFE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCOMPL	-	-	-	-	ADC_COMP<3:0>			
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位值	X	X	X	X	0	0	0	0

Bit<3:0>: ADC 比较值低 4 位

3.1.74 SFR~0xFF/ADC_COMPH (ADC 比较值高 8 位)

0XFF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCOMPH	ADC_COMP<11:4>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: ADC 比较值高 8 位



3.2 XDATA 空间寄存器

3.2.1 XDATA~0xFF00/FLASH_CR (FLASH 控制寄存器)

0xFF00	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FLASHCR	-	-	WRSZ<1:0>		CKEN	-	IFREN	BUSY
读/写	-	-	R/W	R/W	R/W	-	R/W	R/W
复位值	X	X	0	0	0	X	0	0

Bit<5:4>: FLASH 存储器编程数据大小, 单位为半字 (2 个字节)

WRSZ<1:0>		数据大小
0	0	2
0	1	32
1	0	64
1	1	128

Bit<3>: FLASH 时钟使能控制位

- 1: 使能
- 0: 禁止

Bit<1>: FLASH 区域选择位

- 1: 选择 FLASH 信息区域
- 0: 选择 FLASH 用户区域

Bit<0>: 读模式下 BUSY 的值表示 (写 1 开始编程操作)

- 1: FLASH 编程没有完成
- 0: FLASH 编程完成

3.2.2 XDATA~0xFF01/FLASH_CFG (FLASH 配置寄存器)

0xFF01	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FLASHCFG	FWSEL	CLEAN	-	-	SAVPWR1	SAVPWRO	RDCYC<1:0>	
读/写	R/W	R/W	-	-	R/W	R/W	R/W	R/W
复位值	0	0	X	X	0	0	1	1

Bit<7>: FLASH 控制信号选择位

- 1: 使用寄存器定义的 FLASH CLEN, ISAVB, STATICEN 信号
- 0: 使用默认的 FLASH CLEN, ISAVB, STATICEN 信号

Bit<6>: FLASH 测试模式

Bit<3>: SLEEP 模式门控 CS 信号

- 1: SLEEP 模式时 CS 信号门控关闭
- 0: SLEEP 模式时 CS 信号门控打开



注：SLEEP 模式时建议此位置 1，睡眠功耗较未置 1 会更小。

Bit<2>: SLEEP 模式门控 READ 信号

- 1: SLEEP 模式时 READ 信号门控关闭
- 0: SLEEP 模式时 READ 信号门控打开

注：SLEEP 模式时建议此位置 1，睡眠功耗较未置 1 会更小。

Bit<1:0>: FLASH 访问周期选择位

DRCYC<1:0>		周期
0	0	1
0	1	2
1	0	5
1	1	6

注:若访问周期要配置 1, 则芯片电压必须高于 4.5V; 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 时优先推荐使用 01 (2 个周期) 配置即可, 这样可以保证性能和功耗的平衡。

具体指令周期时间如下:

READ Cycle Time	Trc	4.5V ≤ VDD ≤ 5.5V	60	-	-	ns
		1.8V ≤ VDD ≤ 4.5V	300	-	-	ns
READ Access Time	Trac	4.5V ≤ VDD ≤ 5.5V	-	-	60	ns
		1.8V ≤ VDD ≤ 4.5V	-	-	300	ns

3.2.3 XDATA~0xFF10/PT_SEL (端口位置配置寄存器)

0xFF1	Bit 7	Bit 6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PTSEL	-	-	ADGETRSE	I2CSE	TIM2CHBSE	TIM2CHASE	TIM1CHBSE	TIM1CHASE
0	-	-	L	L	L	L	L	L
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	0	0	0	0	0	0

Bit<5>: ADC_ETR 管脚位置选择控制位

- 1: ADC_ETR 使用 P1.3
- 0: ADC_ETR 使用 P0.1

Bit<4>: I2C 管脚位置选择控制位

- 1: SCL 使用 P0.0, SDA 使用 P1.0
- 0: SCL 使用 P1.3, SDA 使用 P1.2

Bit<3>: TIM2_CHB 管脚位置选择控制位

- 1: TIM2_CHB 使用 P0.3
- 0: TIM2_CHB 使用 P1.7



Bit<2>: TIM2_CHA 管脚位置选择控制位

1: TIM2_CHA 使用 P0.0

0: TIM2_CHA 使用 P1.6

Bit<1>: TIM1_CHB 管脚位置选择控制位

1: TIM1_CHB 使用 P1.2

0: TIM1_CHB 使用 P1.5

Bit<0>: TIM1_CHA 管脚位置选择控制位

1: TIM1_CHA 使用 P0.7

0: TIM1_CHA 使用 P1.4

3.2.4 XDATA~0xFF18/PERP0_EN (外设管脚位置使能配置寄存器 0)

0xFF18	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PERPOEN	-	-	-	-	TIM2CHBPEN	TIM2CHAPEN	TIM1CHBPEN	TIM1CHAPEN
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位值	X	X	X	X	0	0	0	0

Bit<3>: TIM2_CHB 外设管脚位置使能控制位

1: 使能

0: 禁止

Bit<2>: TIM2_CHA 外设管脚位置使能控制位

1: 使能

0: 禁止

Bit<1>: TIM1_CHB 外设管脚位置使能控制位

1: 使能

0: 禁止

Bit<0>: TIM1_CHA 外设管脚位置使能控制位

1: 使能

0: 禁止



3.2.5 XDATA~0xFF19/PERP1_EN (外设管脚位置使能配置寄存器 0)

0xFF19	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PERPOEN	-	MTPTESTPEN	ADCETRPEN	CLKMTPEN	UART1PEN	I2CPEN	BRKINPEN	UARTOPEN
读/写	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	0	0	0	0	1	0	0

Bit<6>: MTP_TEST 输出管脚位置使能控制位

1: 使能

0: 禁止

Bit<5>: ADC_ETR 输出管脚位置使能控制位

1: 使能

0: 禁止

Bit<4>: CLK_MTP 输出管脚位置配置使能控制位

1: CLK_MTP 从 P0.4 管脚位置输出使能

0: CLK_MTP 从 P0.4 管脚位置输出禁止

Bit<3>: UART1 外设管脚位置配置寄存器:

1: 使能

0: 禁止

Bit<2>: I2C 外设管脚位置配置使能控制位

1: 使能

0: 禁止

Bit<1>: BRKIN 管脚位置配置使能控制位

1: 使能

0: 禁止

Bit<0>: UART0 外设管脚位置配置使能控制位

1: 使能

0: 禁止

3.2.6 XDATA~0xFF20/P0_FLAG (端口 0 中断标志位)

0xFF20	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
POFLAG	POFLAG<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: P0 中断标志位(写 1 清除该中断标志)

1: 有中断发生



0: 没有中断发生

3.2.7 XDATA~0xFF21/P0_GE (端口 0 数字复用使能寄存器)

0XFF21	Bit7	Bit56	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
POGE	GE0<7>	-	-	GE0<4>	GE0<3>	-	GE0<1>	GE0<0>
读/写	R/W	-	-	R/W	R/W	-	R/W	R/W
复位值	0	X	X	0	0	X	0	0

端口 0 的复用输出使能控制位:

- 1: 使能 (复用输出使能, 输出由 G0 决定, 如果输入使能则输入电平到 G1)
- 0: 禁止 (复用输出使能, 输出由 Px. DR 决定, 如果输入则电平寄存在 Px. DR 中)

3.2.8 XDATA~0xFF23/P0_PU (端口 0 上拉控制寄存器)

0XFF23	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
POPU	POPU<7>	POPU<6>	POPU<5>	POPU<4>	POPU<3>	POPU<2>	POPU<1>	POPU<0>
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: P0 端口上拉使能控制位

- 1: 使能
- 0: 禁止

3.2.9 XDATA~0xFF24/P0_PD (端口 0 下拉控制寄存器)

0XFF24	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
POPD	POPD<7>	POPD<6>	POPD<5>	POPD<4>	POPD<3>	POPD<2>	POPD<1>	POPD<0>
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: P0 端口下拉使能控制位

- 1: 使能
- 0: 禁止



3.2.10 XDATA~0xFF25/P0_IE (端口 0 中断使能寄存器)

0xFF25	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0IE	P0IE<7>	P0IE<6>	P0IE<5>	P0IE<4>	P0IE<3>	P0IE<2>	P0IE<1>	P0IE<0>
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: P0 端口中断使能控制位

1: 使能

0: 禁止

3.2.11 XDATA~0xFF26/P0_IC0 (端口 0 中断控制 0 位)

0xFF26	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0IC0	P0_IC0<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

Bit<7:0>: P0 端口中断触发模式控制位

3.2.12 XDATA~0xFF27/P0_IC1 (端口 0 中断控制 1 位)

0xFF27	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0IC1	P0_IC1<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: P0 端口中断触发模式控制位

P0_IC1	P0_IC0	触发模式
0	0	上升沿中断
0	1	下降沿中断
1	0	高电平中断
1	1	低电平中断

3.2.13 XDATA~0xFF30/P1_FLAG (端口 1 中断标志位)

0xFF30	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1FLAG	P1_FLAG<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	0



Bit<7:0>: P1 中断标志位(写 1 清除该中断标志)

- 1: 有中断发生
- 0: 没有中断发生

3.2.14 XDATA~0xFF31/P1_GE (端口 1 数字复用使能寄存器)

0xFF31	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1GE	GE1<7>	GE1<6>	GE1<5>	GE1<4>	GE1<3>	GE1<2>	-	GE1<0>
读/写	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W
复位值	0	0	0	0	1	1	X	0

P1 端口外设复用使能控制位:

- 1: 使能 (输出由 G0 决定, 如果输入使能则输入电平到 GI)
- 0: 禁止 (输出由 Px.DR 决定, 如果输入则输入电平寄存在 Px.DR 中)

3.2.15 XDATA~0xFF33/P1_PU (端口 1 上拉控制寄存器)

0xFF33	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1PU	P1PU<7>	P1PU<6>	P1PU<5>	P1PU<4>	P1PU<3>	P1PU<2>	P1PU<1>	P1PU<0>
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	1	1	0	0

Bit<7:0>: P1 端口上拉使能控制位

- 1: 使能
- 0: 禁止

3.2.16 XDATA~0xFF34/P1_PD (端口 1 下拉控制寄存器)

0xFF34	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1PD	P1PD<7>	P1PD<6>	P1PD<5>	P1PD<4>	P1PD<3>	P1PD<2>	P1PD<1>	P1PD<0>
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: P1 端口下拉使能控制位

- 1: 使能
- 0: 禁止



3.2.17 XDATA~0xFF35/P1_IE (端口 1 中断使能寄存器)

0xFF35	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1IE	P1IE<7>	P1IE<6>	P1IE<5>	P1IE<4>	P1IE<3>	P1IE<2>	P1IE<1>	P1IE<0>
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: P1 端口中断使能控制位

- 1: 使能
- 0: 禁止

3.2.18 XDATA~0xFF36/P1_IC0 (端口 1 中断控制 0 位)

0xFF36	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1IC0	P1_IC0<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

Bit<7:0>: P1 端口中断触发模式控制位

3.2.19 XDATA~0xFF37/P1_IC1 (端口 1 中断控制 1 位)

0xFF37	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1IC1	P1_IC1<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: P1 端口中断触发模式控制位

P1_IC1	P1_IC0	触发模式
0	0	上升沿中断
0	1	下降沿中断
1	0	高电平中断
1	1	低电平中断

3.2.20 XDATA~0xFF40/P2_FLAG (端口 2 中断标志位)

0xFF40	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1FLAG	P2_FLAG<7:0>							
读/写	-	-	-	-	-	-	R/W	R/W
复位值	X	X	X	X	X	X	0	0



Bit<1:0>: P2 中断标志位(写 1 清除该中断标志)

- 1: 有中断发生
- 0: 没有中断发生

3. 2. 21 XDATA~0xFF41/P2_GE (端口 2 数字复用使能寄存器)

0xFF41	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2GE	-	-	-	-	-	-	GE2<1>	-
读/写	-	-	-	-	-	-	R/W	-
复位值	X	X	X	X	X	X	0	X

Bit<1>: P2 端口外设复用功能使能位

- 1: 使能, 输出由 G0 决定, 如果输入使能则输入电平到 GI
- 0: 禁止, 输出由 Px. DR 决定, 如果输入则输入电平寄存在 Px. DR 中

3. 2. 22 XDATA~0xFF43/P2_PU (端口 2 上拉控制寄存器)

0xFF43	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2PU	P2PU<7>	P2PU<6>	P2PU<5>	P2PU<4>	P2PU<3>	P2PU<2>	P2PU<1>	P2PU<0>
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<1:0>: P2 端口上拉使能控制位

- 1: 使能
- 0: 禁止

3. 2. 23 XDATA~0xFF44/P2_PD (端口 2 下拉控制寄存器)

0xFF44	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2PD	P2PD<7>	P2PD<6>	P2PD<5>	P2PD<4>	P2PD<3>	P2PD<2>	P2PD<1>	P2PD<0>
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<1:0>: P2 端口下拉使能控制位

- 1: 使能
- 0: 禁止



3.2.24 XDATA~0xFF45/P2_IE (端口 2 中断使能寄存器)

0xFF35	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2IE	-	-	-	-	-	-	P2IE<1>	P2IE<0>
读/写	-	-	-	-	-	-	R/W	R/W
复位值	X	X	X	X	X	X	0	0

Bit<1:0>: P2 端口中断使能控制位

- 1: 使能
- 0: 禁止

3.2.25 XDATA~0xFF46/P2_IC0 (端口 1 中断控制 0 位)

0xFF46	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2IC0	P2_IC0<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	1	1

Bit<1:0>: P2 端口中断触发模式控制位

3.2.26 XDATA~0xFF47/P2_IC1 (端口 1 中断控制 1 位)

0xFF47	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2IC1	P2_IC1<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<1:0>: P2 端口中断触发模式控制位

P2_IC1	P2_IC0	触发模式
0	0	上升沿中断
0	1	下降沿中断
1	0	高电平中断
1	1	低电平中断

3.2.27 XDATA~0xFF50/TIM1_FCONR (Timer1 时钟控制寄存器)

0xFF50	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1FCONR	-	CLK_SEL<2:0>			PRE_DIV<3:0>			
读/写	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	0	0	0	0	0	0	0



Bit<6:4>: TIMER1 时钟源选择位

CLK_SEL<2:0>			TIMER1 时钟
0	0	0	SCK0
0	0	1	看门狗时钟 32kHz
0	1	0	保留
0	1	1	保留
1	0	0	TIM1_CHA 上升沿
1	0	1	TIM1_CHB 上升沿
1	1	0	TIM1_CHA 下降沿
1	1	1	TIM1_CHB 下降沿

Bit<3:0>: TIMER1 预分频选择位

CLK_SEL<3:0>				TIMER1 预分频
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

3.2.28 XDATA~0xFF51/TIM1_VPERR (Timer1 周期间隔响应控制寄存器)

0XFF51	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1VPERR	-	-	PCNTE[1:0]		-	PCNTS[2:0]		
读/写	-	-	R/W	R/W	-	R/W	R/W	R/W
复位值	X	X	0	0	X	0	0	0

Bit<5:4>: 周期间隔响应计数条件选择位

PCNTE<1:0>		周期间隔响应计数条件
0	0	有效周期选择功能无效
0	1	锯齿波计数上、下溢点或三角波波峰作为计数条件
1	0	锯齿波计数上、下溢点或三角波波谷作为计数条件
1	1	齿波计数上、下溢点或三角波波谷、波峰作为计数条件

Bit<2:0>: 周期间隔响应周期



PCNTS<2:0>			响应周期
0	0	0	1 个周期响应一次
0	0	1	2 个周期响应一次
0	1	0	4 个周期响应一次
0	1	1	8 个周期响应一次
1	0	0	16 个周期响应一次
1	0	1	32 个周期响应一次
1	1	0	64 个周期响应一次
1	1	1	128 个周期响应一次

3.2.29 XDATA~0xFF52/TIM1_DTUA (Timer1 死区事件寄存器)

0xFF52	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1FCNR	TIM1_DTUA<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIMER1 死区时间设定值

3.2.30 XDATA~0xFF53/TIM1_BRAKE (Timer1 刹车控制寄存器)

0xFF53	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1BRAKE	TIB_MOE	TIB_AOE	TIB_SEL	TIB_EN	TIA_MOE	TIA_AOE	TIA_SEL	TIA_EN
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TIM1_CHB 主输出使能位 (PWM 输出时, 需要使能该位)

- 1: TIM1_CHB 主输出有效
- 0: TIM1_CHB 主输出关闭

注: 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

Bit<6>: 自动输出使能位

- 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

Bit<5>: 选择 TIM1_CHB 刹车来源

- 0: TIM1_CHB 刹车事件选择 ADC 比较输出
- 1: TIM1_CHB 刹车事件选择 P1.2BKIN 输入

Bit<4>: 刹车功能控制位

- 1: TIM1_CHB 刹车有效
- 0: TIM1_CHB 刹车无效

Bit<3>: TIM1_CHA 主输出使能位



1: TIM1_CHA 主输出有效

0: TIM1_CHA 主输出关闭

注：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

Bit<2>: TIM1_CHA 自动输出使能位

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

Bit<1>: 选择 TIM1_CHA 刹车来源

0: TIM1_CHA 刹车事件选择 ADC 比较输出

1: TIM1_CHA 刹车事件选择 P1.2 BKIN 输入

Bit<0>: TIM1_CHA 刹车功能控制位

1: TIM1_CHA 刹车有效

0: TIM1_CHA 刹车无效

3.2.31 XDATA~0xFF54/TIM1_DTR (Timer1 死区控制寄存器)

0xFF54	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1DTR	-	-	-	HWCPWM	DTH_B	DTB_EN	DTH_A	DTA_EN
读/写	-	-	-	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	0	0	0	0	0

Bit<4>: 控制 GCMBR 互补模式

1: 硬件设定 GCMBR 互补 PWM 输出模式开

0: 硬件设定 GCMBR 互补 PWM 输出模式关

Bit<3>: 控制死区输出是否使能 HALF 功能

1: 输出 B 使能

0: 输出 B 禁止

Bit<2>: 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

Bit<1>: 控制死区输出是否使能 HALF 功能

1: 输出 A 使能

0: 输出 A 禁止

Bit<0>: 死区控制使能位

1: 输出 A 死区控制有效

0: 输出 A 死区控制无效



3.2.32 XDATA~0xFF55/TIM1_PCONRA (Timer1 端口 A 控制寄存器)

0xFF55	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1PCONRA	PAINITVAL	CMPA_VAL[1:0]		PA_ENO	PAFILTEREN	CAPA_MODE[1:0]		CAPA_EN
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: 设置 TIM1_CHA 的输出控制位 (TIMER1 关时设定有效, TIMER1 开时中间设定无效)

- 1: TIM1_CHA 的初始值为 1
- 0: TIM1_CHA 的初始值为 0

Bit<6:5>: 配置 TIM1_CHA 比较输出值

CMPA_VAL[1:0]		TIM1_CHA 比较输出值
0	0	计数值小于比较值为 1, 大于为 0
0	1	计数值大于比较值为 1, 小于为 0
1	0	比较值匹配, 输出取反前一状态
1	1	比较值匹配, 输出保持前一状态

Bit<4>: TIM1_CHA 输出控制位

- 1: TIM1_CHA 输出打开
- 0: TIM1_CHA 输出关闭

Bit<3>: TIM1_CHA 输入滤波使能位

- 1: TIM1_CHA 输入数字滤波打开
- 0: TIM1_CHA 输入数字滤波关闭

Bit<2:1>: TIM1_CHA 捕获模式选择位

CAPA_MODE[1:0]		捕获模式
0	0	不捕获
0	1	捕获上升沿
1	0	捕获下降沿
1	1	捕获上升沿与下降沿

Bit<0>: TIM1_CHA 捕获模式使能位

- 1: TIM1_CHA 捕获模式开
- 0: TIM1_CHA 捕获模式关

3.2.33 XDATA~0xFF56/TIM1_PCONRB (Timer1 端口 B 控制寄存器)

0xFF56	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1PCONRB	PBINITVAL	CMPB_VAL[1:0]		PB_ENO	PBFILTEREN	CAPB_MODE[1:0]		CAPB_EN
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: 设置 TIM1_CHB 的输出位 (TIMER1 关时设定有效, TIMER1 开时中间设定无效)



1: TIM1_CHB 的初始值为 1

0: TIM1_CHB 的初始值为 0

Bit<6:5>: 配置 TIM1_CHB 比较输出值

CMPB_VAL [1:0]		TIM1_CHB 比较输出值
0	0	计数值小于比较值为 1, 大于为 0
0	1	计数值大于比较值为 1, 小于为 0
1	0	比较值匹配, 输出取反前一状态
1	1	比较值匹配, 输出保持前一状态

Bit<4>: TIM1_CHB 输出控制位

1: TIM1_CHB 输出打开

0: TIM1_CHB 输出关闭

Bit<3>: TIM1_CHB 输入滤波使能位

1: TIM1_CHB 输入数字滤波打开

0: TIM1_CHB 输入数字滤波关闭

Bit<2:1>: TIM1_CHB 捕获模式选择位

CAPA_MODE [1:0]		捕获模式
0	0	不捕获
0	1	捕获上升沿
1	0	捕获下降沿
1	1	捕获上升沿与下降沿

Bit<0>: TIM1_CHB 捕获模式使能位

1: TIM1_CHB 捕获模式开

0: TIM1_CHB 捕获模式关

3.2.34 XDATA~0xFF58/TIM1_CNTRL (Timer1 计数值寄存器低 8 位)

0xFF58	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1CNTRL	TIM1_CNTRL<7:0>							
读/写	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM1 计数器计数寄存器低 8 位

3.2.35 XDATA~0xFF59/TIM1_CNTH (Timer1 计数值寄存器高 8 位)

0xFF59	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1CNTH	TIM1_CNTH<7:0>							
读/写	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0



Bit<7:0>: TIM1 计数器计数寄存器高 8 位

3.2.36 XDATA~0xFF5A/TIM1_ARRL (Timer1 自动重载寄存器低 8 位)

0xFF5A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1ARRL	TIM1_ARRL<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM1 计数器自动重载值寄存器低 8 位, 需先写高 8 位再写低 8 位。

3.2.37 XDATA~0xFF5B/TIM1_ARRH (Timer1 自动重载寄存器高 8 位)

0xFF5B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1ARRH	TIM1_ARRH<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM1 计数器自动重载值寄存器高 8 位, 需先写高 8 位再写低 8 位。

3.2.38 XDATA~0xFF5C/TIM1_GCMARL (Timer1 比较捕获寄存器 A 低 8 位)

0xFF5C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1GCMARL	TIM1_GCMARL<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM1 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。

3.2.39 XDATA~0xFF5D/TIM1_GCMARH (Timer1 比较捕获寄存器 A 高 8 位)

0xFF5D	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1GCMARH	TIM1_GCMARH<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM1 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。



3.2.40 XDATA~0xFF5E/TIM1_GCMBRL (Timer1 比较捕获寄存器 B 低 8 位)

0xFF5E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1GCMBRL	TIM1_GCMBRL<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM1 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。

3.2.41 XDATA~0xFF5F/TIM1_GCMBRH (Timer1 比较捕获寄存器 B 高 8 位)

0xFF5F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1GCMBRH	TIM1_GCMBRH<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM1 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。

3.2.42 XDATA~0xFF60/TIM2_FC0NR (Timer2 时钟控制寄存器)

0xFF60	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2FC0NR	-	CLK_SEL<2:0>			PRE_DIV<3:0>			
读/写	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	0	0	0	0	0	0	0

Bit<6:4>: TIMER2 时钟源选择位:

CLK_SEL<2:0>			TIMER1 时钟
0	0	0	SCK0
0	0	1	看门狗时钟 32kHz
0	1	0	保留
0	1	1	保留
1	0	0	TIM2_CHA 上升沿
1	0	1	TIM2_CHB 上升沿
1	1	0	TIM2_CHA 下降沿
1	1	1	TIM2_CHB 下降沿

Bit<3:0>: TIMER2 预分频选择位



CLK_SEL<3:0>				TIMER1 预分频
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

注：修改 CLK_SEL 和 PRE_DIV 寄存器配置必须在 TIM2_EN 为 0 的时候进行。

3.2.43 XDATA~0xFF61/TIM2_VPERR (Timer2 周期间隔响应控制寄存器)

0XFF61	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2VPERR	-	-	PCNTE[1:0]		-	PCNTS[2:0]		
读/写	-	-	R/W	R/W	-	R/W	R/W	R/W
复位值	X	X	0	0	X	0	0	0

Bit<5:4>: 周期间隔响应计数条件选择位

PCNTE<1:0>		周期间隔响应计数条件
0	0	有效周期选择功能无效
0	1	锯齿波计数上、下溢点或三角波波峰作为计数条件
1	0	锯齿波计数上、下溢点或三角波波谷作为计数条件
1	1	锯齿波计数上、下溢点或三角波波谷、波峰作为计数条件

Bit<2:0>: 周期间隔响应周期

PCNTS<2:0>			响应周期
0	0	0	1 个周期响应一次
0	0	1	2 个周期响应一次
0	1	0	4 个周期响应一次
0	1	1	8 个周期响应一次
1	0	0	16 个周期响应一次
1	0	1	32 个周期响应一次
1	1	0	64 个周期响应一次
1	1	1	128 个周期响应一次



3.2.44 XDATA~0xFF62/TIM2_DTUA (Timer2 死区事件寄存器)

0XFF62	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2FC0NR	TIM2_DTUA<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIMER2 死区时间设定值

3.2.45 XDATA~0xFF63/TIM2_BRAKE (Timer2 刹车控制寄存器)

0XFF63	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2BRAKE	TIB_MOE	TIB_AOE	TIB_SEL	TIB_EN	TIA_MOE	TIA_AOE	TIA_SEL	TIA_EN
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TIM2_CHB 主输出使能位 (PWM 输出时, 需要使能该位)

- 1: TIM2_CHB 主输出有效
- 0: TIM2_CHB 主输出关闭

注: 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

Bit<6>: 自动输出使能位

- 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

Bit<5>: 选择 TIM2_CHB 刹车来源

- 1: TIM2_CHB 刹车事件选择 P1.2 BKIN 输入
- 0: TIM2_CHB 刹车事件选择 ADC 比较输出

Bit<4>: 刹车功能控制位

- 1: TIM2_CHB 刹车有效
- 0: TIM2_CHB 刹车无效

Bit<3>: TIM2_CHA 主输出使能位

- 1: TIM2_CHA 主输出有效
- 0: TIM2_CHA 主输出关闭

注: 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

Bit<2>: TIM2_CHA 自动输出使能位

- 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

Bit<1>: 选择 TIM2_CHA 刹车来源

- 1: TIM2_CHA 刹车事件选择 P1.2 BKIN 输入
- 0: TIM2_CHA 刹车事件选择 ADC 比较输出



Bit<0>: TIM2_CHA 刹车功能控制位

1: TIM2_CHA 刹车有效

0: TIM2_CHA 刹车无效

3.2.46 XDATA~0xFF64/TIM2_DTR (Timer2 死区控制寄存器)

0xFF64	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2DTR	-	-	-	HWCPWM	DTH_B	DTB_EN	DTH_A	DTA_EN
读/写	-	-	-	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	0	0	0	0	0

Bit<4>: 控制 GCMBR 互补模式

1: 硬件设定 GCMBR 互补 PWM 输出模式开

0: 硬件设定 GCMBR 互补 PWM 输出模式关

Bit<3>: 控制死区输出是否使能 HALF 功能

1: 输出 B 使能

0: 输出 B 禁止

Bit<2>: 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

Bit<1>: 控制死区输出是否使能 HALF 功能

1: 输出 A 使能

0: 输出 A 禁止

Bit<0>: 死区控制使能位

1: 输出 A 死区控制有效

0: 输出 A 死区控制无效

3.2.47 XDATA~0xFF65/TIM2_PCONRA (Timer2 端口 A 控制寄存器)

0xFF65	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2PCONRA	PAINITVAL	CMPA_VAL[1:0]		PA_ENO	PAFILTEREN	CAPA_MODE[1:0]		CAPA_EN
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: 设置 TIM2_CHA 的输出控制位 (TIMER2 关时设定有效, TIMER2 开时中间设定无效)

1: TIM2_CHA 的初始值为 1

0: TIM2_CHA 的初始值为 0

Bit<6:5>: 配置 TIM2_CHA 比较输出值



CMPA_VAL [1:0]		TIM2_CHA 比较输出值
0	0	计数值小于比较值为 1, 大于为 0
0	1	计数值大于比较值为 1, 小于为 0
1	0	比较值匹配, 输出取反前一状态
1	1	比较值匹配, 输出保持前一状态

Bit<4>: TIM2_CHA 输出控制位

1: TIM2_CHA 输出打开

0: TIM2_CHA 输出关闭

Bit<3>: TIM2_CHA 输入滤波使能位

1: TIM2_CHA 输入数字滤波打开

0: TIM2_CHA 输入数字滤波关闭

Bit<2:1>: TIM2_CHA 捕获模式选择位

CAPA_MODE [1:0]		捕获模式
0	0	不捕获
0	1	捕获上升沿
1	0	捕获下降沿
1	1	捕获上升沿与下降沿

Bit<0>: TIM2_CHA 捕获模式使能位

1: TIM2_CHA 捕获模式开

0: TIM2_CHA 捕获模式关

3.2.48 XDATA~0xFF66/TIM2_PCONRB (Timer2 端口 B 控制寄存器)

0xFF66	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2PCONRB	PBINITVAL	CMPB_VAL [1:0]		PB_ENO	PBFILTEREN	CAPB_MODE [1:0]		CAPB_EN
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: 设置 TIM2_CHB 的输出位 (TIMER2 关时设定有效, TIMER2 开时中间设定无效)

1: TIM2_CHB 的初始值为 1

0: TIM2_CHB 的初始值为 0

Bit<6:5>: 配置 TIM2_CHB 比较输出值

CMPB_VAL [1:0]		TIM2_CHB 比较输出值
0	0	计数值小于比较值为 1, 大于为 0
0	1	计数值大于比较值为 1, 小于为 0
1	0	比较值匹配, 输出取反前一状态
1	1	比较值匹配, 输出保持前一状态

Bit<4>: TIM2_CHB 输出控制位

1: TIM2_CHB 输出打开

0: TIM2_CHB 输出关闭



Bit<3>: TIM2_CHB 输入滤波使能位

1: TIM2_CHB 输入数字滤波打开

0: TIM2_CHB 输入数字滤波关闭

Bit<2:1>: TIM2_CHB 捕获模式选择位

CAPA_MODE[1:0]		捕获模式
0	0	不捕获
0	1	捕获上升沿
1	0	捕获下降沿
1	1	捕获上升沿与下降沿

Bit<0>: TIM2_CHB 捕获模式使能位

1: TIM2_CHB 捕获模式开

0: TIM2_CHB 捕获模式关

3.2.49 XDATA~0xFF68/TIM2_CNTRL (Timer2 计数值寄存器低 8 位)

0xFF68	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1CNTRL	TIM2_CNTRL<7:0>							
读/写	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM2 计数器计数寄存器低 8 位

3.2.50 XDATA~0xFF69/TIM2_CNTH (Timer2 计数值寄存器高 8 位)

0xFF69	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM1CNTH	TIM2_CNTH<7:0>							
读/写	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM2 计数器计数寄存器高 8 位

3.2.51 XDATA~0xFF6A/TIM2_ARRL (Timer2 自动重载寄存器低 8 位)

0xFF6A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2ARRL	TIM2_ARRL<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM2 计数器自动重载值寄存器低 8 位，需先写高 8 位再写低 8 位。



3.2.52 XDATA~0xFF6B/TIM2_ARRH (Timer2 自动重载寄存器高 8 位)

0XFF6B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2ARRH	TIM2_ARRH<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM2 计数器自动重载值寄存器高 8 位, 需先写高 8 位再写低 8 位。

3.2.53 XDATA~0xFF6C/TIM2_GCMARL (Timer2 比较捕获寄存器 A 低 8 位)

0XFF6C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2GCMARL	TIM2_GCMARL<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM2 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。

3.2.54 XDATA~0xFF6D/TIM2_GCMARH (Timer2 比较捕获寄存器 A 高 8 位)

0XFF6D	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2GCMARH	TIM2_GCMARH<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM2 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。

3.2.55 XDATA~0xFF6E/TIM1_GCMBRL (Timer2 比较捕获寄存器 B 低 8 位)

0XFF6E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2GCMBRL	TIM2_GCMBRL<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM2 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。



3.2.56 XDATA~0xFF6F/TIM2_GCMBRH (Timer3 比较捕获寄存器 B 高 8 位)

0xFF6F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM2GCMBRH	TIM2_GCMBRH<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TIM2 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。

3.2.57 XDATA~0xFF80/BG_CR (Bandgap 使能寄存器)

0xFF85	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BORLVD CR	-	-	-	BG_VON_N	-	-	-	BG_EN_N
读/写	-	-	-	R/W	-	-	-	R/W
复位值	X	X	X	0	X	X	X	0

Bit<4>: Bandgap 输出控制位

- 0: 输出 Bandgap
- 1: Bandgap 工作但不输出

Bit<0>: Bandgap 使能控制位

- 0: 使能
- 1: 禁止

3.2.58 XDATA~0xFF85/BORLVD_CR (BORLVD 控制寄存器)

0xFF85	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BORLVD CR	-	BOR_VSEL<1:0>		BOR_EN	-	LVD_VSEL<1:0>		LVD_EN
读/写	-	R/W	R/W	R/W	-	R/W	R/W	R/W
复位值	X	0	0	1	X	0	0	1

Bit<6:5>: BOR 电压点选择位

BOR_VSEL<2:0>		电压点
0	0	2.2V (默认值)
0	1	2.5V
1	0	3.6V
1	1	4.2V

Bit<4>: BOR 使能控制位

- 1: 使能
- 0: 禁止



Bit<3:1>: LVD 电压点选择位

LVD_VSEL<1:0>		电压点
0	0	2.3V(默认值)
0	1	2.7V
1	0	3.8V
1	1	4.5V

Bit<0>: LVD 使能控制位

- 1: 使能
- 0: 禁止

3.2.59 XDATA~0xFF86/BORLVD_STAT (BORLVD 状态寄存器)

0xFF86	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STAT	-	-	-	IE_LVD	STATBOR	-	-	STATLVD
读/写	-	-	-	R/W	R/W	-	-	R
复位值	X	X	X	0	0	X	X	0

Bit<4>: LVD 中断控制位

- 1: 使能
- 0: 关闭

Bit<3>: BOR 输出状态位

- 1: BOR 发生
- 0: BOR 没有发生

Bit<0>: LVD 输出状态位 (该标志位只能读, 不能清除)

- 1: 检测到 LVD 事件
- 0: 没有 LVD 事件发生

3.2.60 XDATA~0xFF88/IMO_CR (IMO 控制寄存器)

0xFF88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IMOCR	EXT_SEL	FX2_SEL	-	-	IMOTSTEN	-	-	IMO_EN
读/写	R/W	R/W	-	-	R/W	-	-	R/W
复位值	0	0	X	X	0	X	X	1

Bit<7:6>: 系统时钟源选择位

EXT_SEL	FX2_SEL	时钟源
0	X	选择内部 8MHz 时钟
1	0	保留, 不可使用
1	1	选择内部 16MHz 时钟

Bit<3>: IMO 测试功能选择位



1: IM0 测试功能打开, 选择 SCK1 到 P0.4 口

0: IM0 测试功能关闭

注:使用 IM0 测试功能之前, 要把 P0.4 的 GPIO 复用功能打开。并且不允许 CLK_MTP/MTP_TEST/SCK1 输出的使能同时打开, 如果同时打开则 P0.4 输出 0。

Bit<0>: IM0 使能位

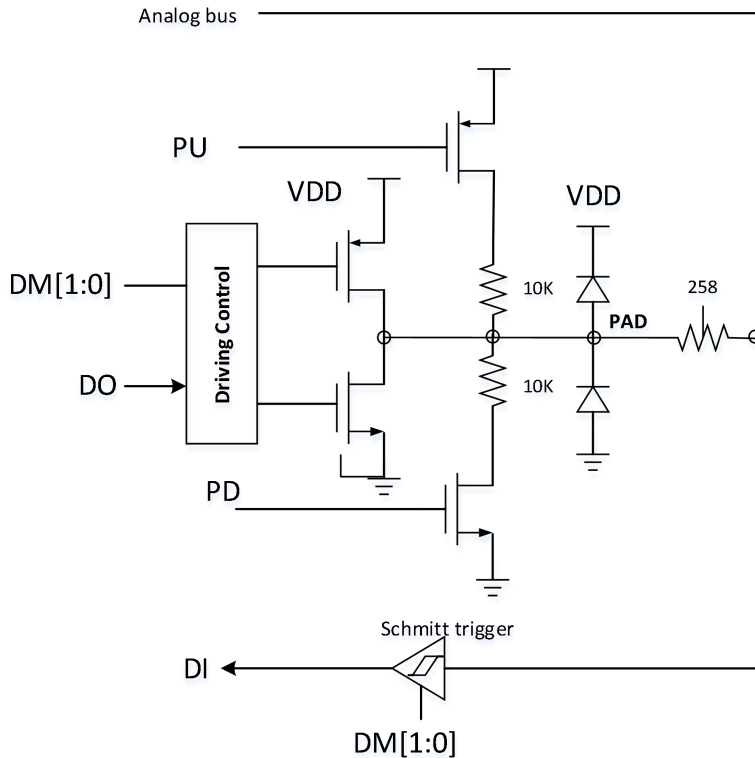
1: 关闭 IM0 (写模式); 使能 IM0 (读模式)

0: 使能 IM0 (写模式); 关闭 IM0 (读模式)

注:不允许关闭 IM0, 否则整个系统时钟会被关掉, 导致系统卡死。

3.3 GPIO

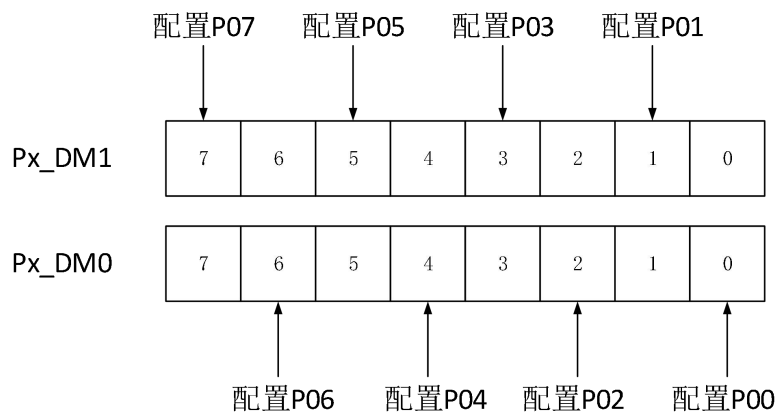
3.3.1 配置 I/O 口



GPIO 结构图

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：



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：

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 为高阻输入，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	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。

3.3.2 外设功能管脚

XC8F9362 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。

- 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；
- PER0_EN 和 PER1_EN 寄存器用来使能部分数字外设的管脚功能。



3.4 CPU

XC8F9362 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。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。



3.5 中断

3.5.1 概述

XC8F9362 支持多达 13 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 13 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

3.5.2 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。

3.5.3 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。



3.5.4 中断向量表

中断控制器支持 13 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。

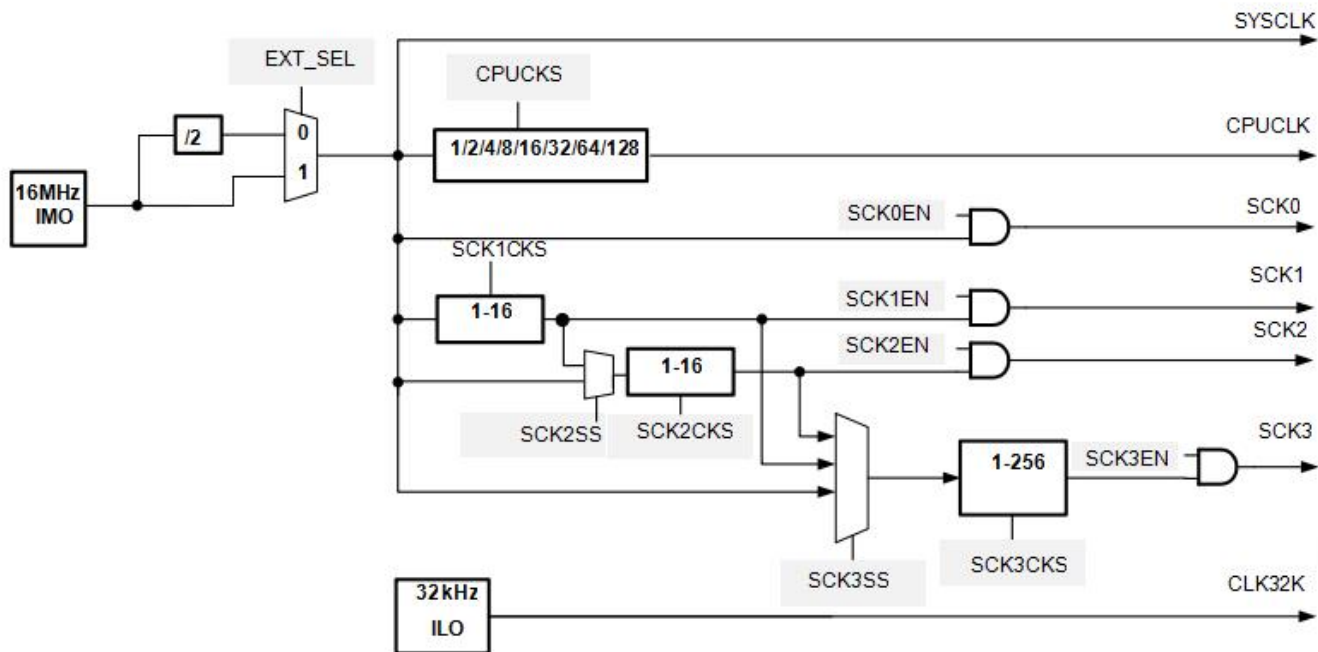
中断源	中断等级	中断号	中断地址	说明
LVD	低	0	0003H	低压检测中断
P0	低	1	000BH	GPIO0 脚中断
P1	低	2	0013H	GPIO1 脚中断
P2	低	3	001BH	GPIO2 脚中断
SCK3	低	4	0023H	SCK3 时钟有效中断
Timer0	低	5	002BH	定时器 0 中断
Timer1	低	6	0033H	定时器 1 中断
Timer2	低	7	003BH	定时器 2 中断
ADC	低	8	0043H	ADC 转换完成中断
I2C	低	9	004BH	I2C 状态中断
UART0	低	10	0053H	UART0 状态中断
UART1	低	11	005BH	UART1 状态中断
WDT	低	12	0063H	看门狗中断

3.6 时钟

3.6.1 概述

系统有两个时钟源，来自内部的 16MHz 高速 RC 振荡器和 32KHz 低速 RC 振荡器。

3.6.2 时钟结构框图



时钟结构框图

外设时钟源表:

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 0~2	HCLK_MEM	SCK0/CLK_32K
ADC	SCK0	SCK0
UART0/1	HCLK_CORE	SCK1/SCK2/SCK3
I2C	HCLK_CORE	SCK1/SCK2
GPIO	HCLK_MEM	DP _x _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	



3.6.3 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、128。CPUCLK 时钟提供 8051 内核工作时钟。

3.6.4 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。

3.6.5 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。

3.6.6 32K 时钟

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

3.7 复位

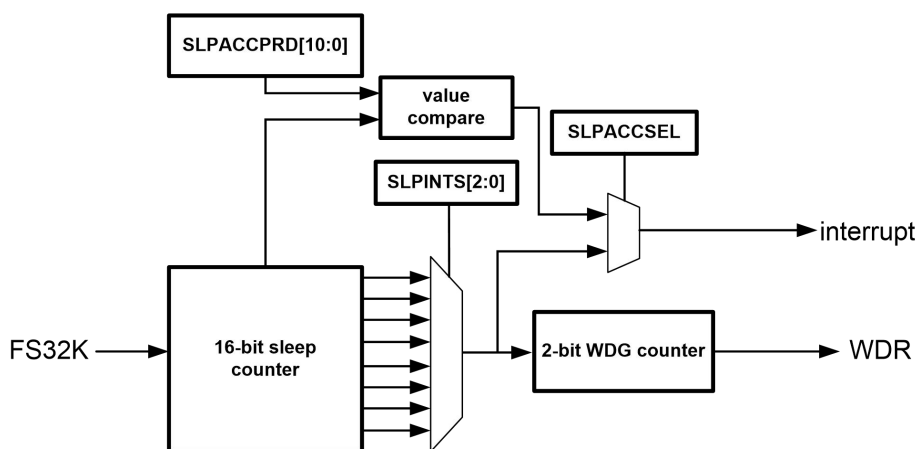
复位源有 4 个，其中包括 3 个全局复位：POR 复位，BOR 复位和看门狗复位，还有一个软复位。

3.7.1 看门狗复位

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。

特性：

- 16 位睡眠定时器，可用作定时中断
- 2 位看门狗定时器，计数到 3 后溢出
- 看门狗定时器计数时间可以配置为 4/8/16/32/256/512/1024/2048ms
- 看门狗定时器计数时间也可自由配置
- 看门狗定时器有单独中断，可用于单独定时

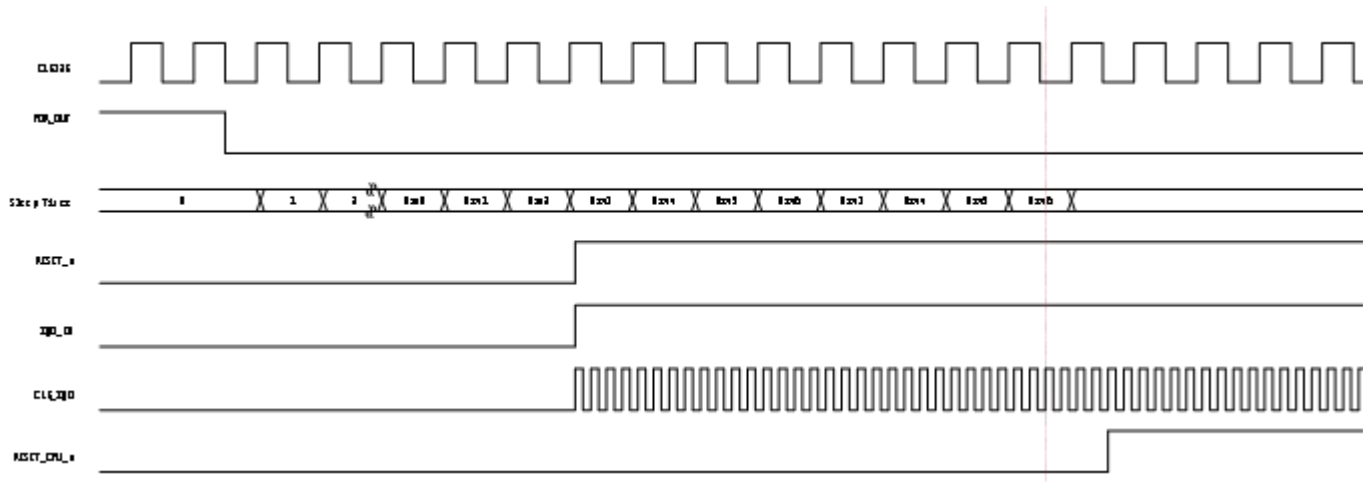


看门狗定时器

操作说明：

对睡眠定时器或者看门狗定时器的操作应遵循以下操作顺序：

1. SLPTIM_CLR 和 SLPTIM_SR 写 0x00 清除睡眠计数器和看门狗，关闭睡眠定时器和看门狗定时器的时钟；
2. 进行睡眠定时器和看门狗定时器的时钟选择或者配置寄存器 SLPTIM_PRDL、SLPTIM_PRDRH 和 SLPTIM_CR；



3.8 外设

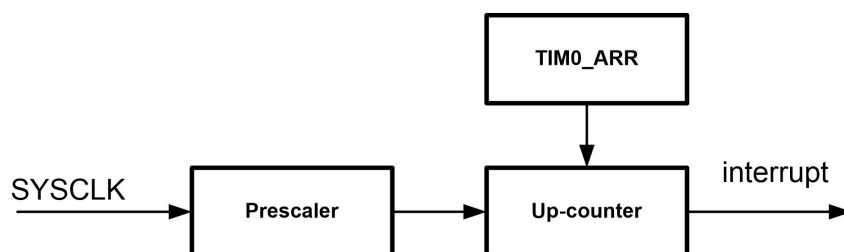
3.8.1 8-bit 基本计数器

3.8.1.1 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

3.8.1.2 结构框图



TIMERO 结构框图



3.8.2 16-bit 高级计数器

3.8.2.1 概述

高级定时器是一个包含两个定时器 TIMER1/2。TIMER1/2 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。

3.8.2.2 主要特性

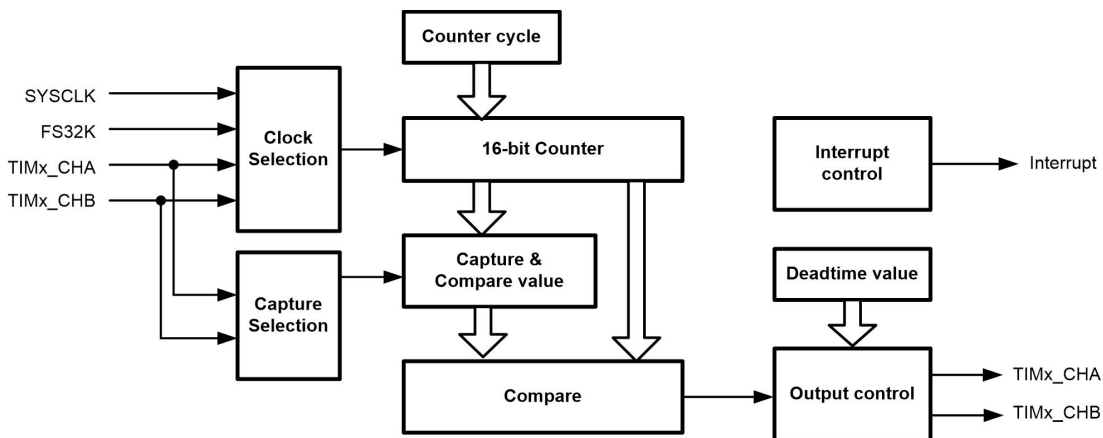
主要特性如下：

- 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
- 支持 6 种时钟源
 - 1) 系统时钟:SYSCLK
 - 2) 32KHz 时钟
 - 3) 定时器输入通道 A 上升沿（需打开捕获使能）
 - 4) 定时器输入通道 B 上升沿（需打开捕获使能）
 - 5) 定时器输入通道 A 下降沿（需打开捕获使能）
 - 6) 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
- 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 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 PWM 输出功能
 - 1) 可输出 2 路独立 PWM 或者 1 路互补 PWM，互补输出可编程死区
 - 2) 支持刹车功能，刹车输入包括：ADC 输出，外部引脚 BKIN (DP1_2) 输入
 - 3) 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）
 - 4) 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：



- 1) 计数器上溢或下溢
- 2) 输入捕获
- 3) 比较输出
- 4) 刹车产生

3.8.2.3 结构框图



timer 时钟结构框图

3.8.2.4 基本动作

1、基本波形模式：

TIMER1/2 有 2 种基本计数波形模式，锯齿波模式和三角波模式。

(1) 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；

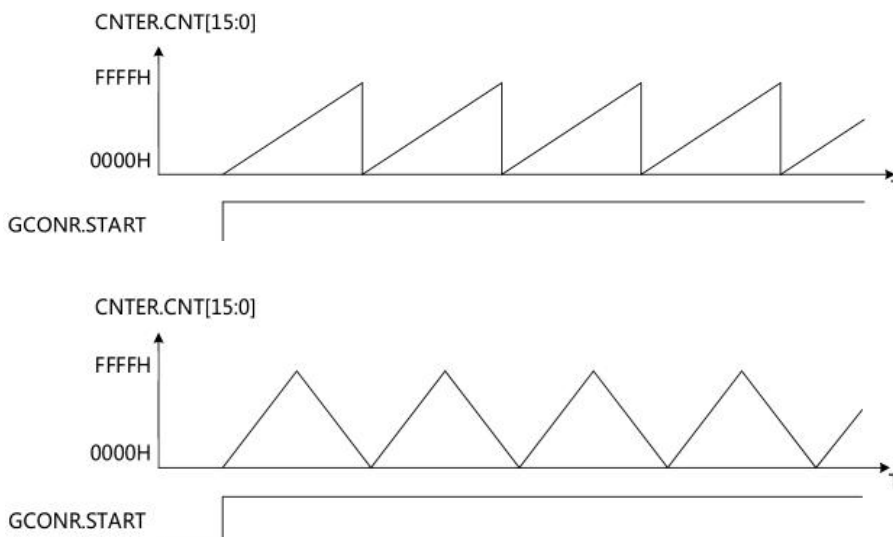
向下计数：计数器每节拍减少 1，直至为 0 时自动加载计数周期值；

(2) 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

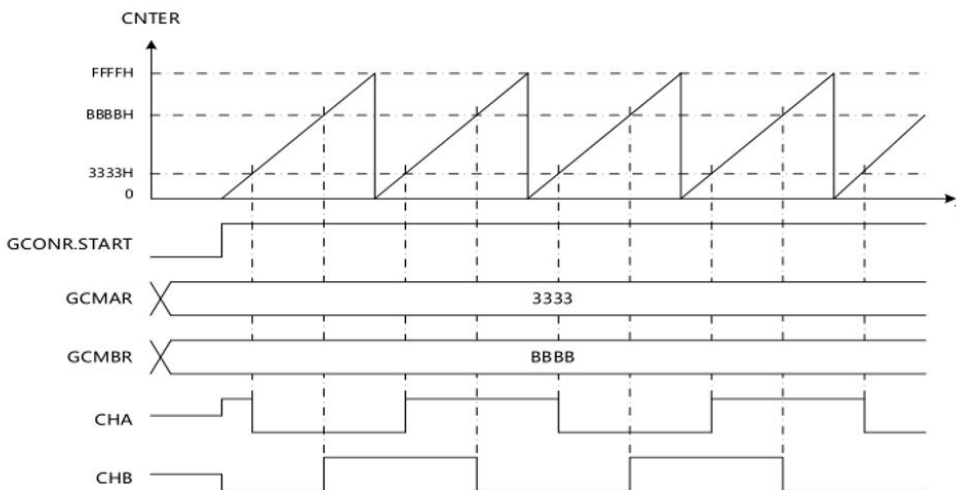
值；



2、比较输出

TIMER1/2 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。

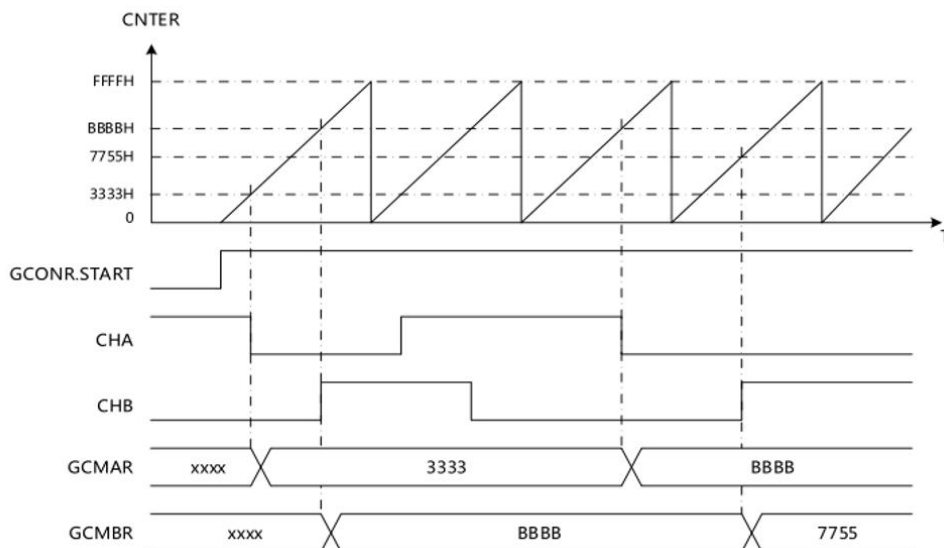


3、捕获输入

TIMER1/2 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。

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。



3.8.2.5 时钟源选择

TIMER1/2 的计数时钟可以有以下几种选择：

- 系统时钟 (SYSCLK)
- 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 的沿（上升沿，下降沿）作为时钟，进行计数。

3.8.2.6 计数方向

TIMER1/2 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。

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。

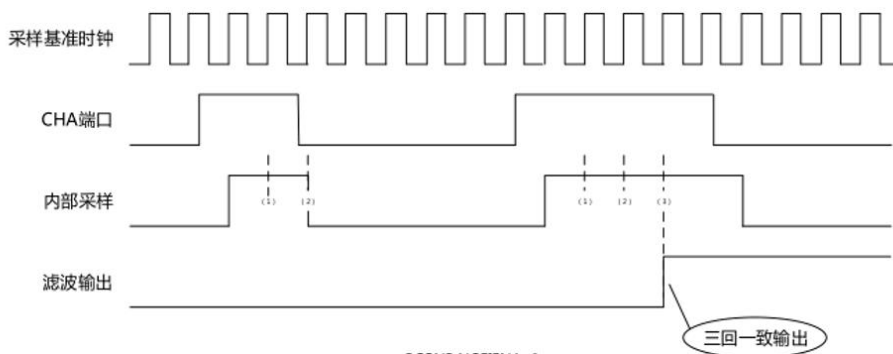
2) 三角波计数方向

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。

3.8.2.7 数字滤波

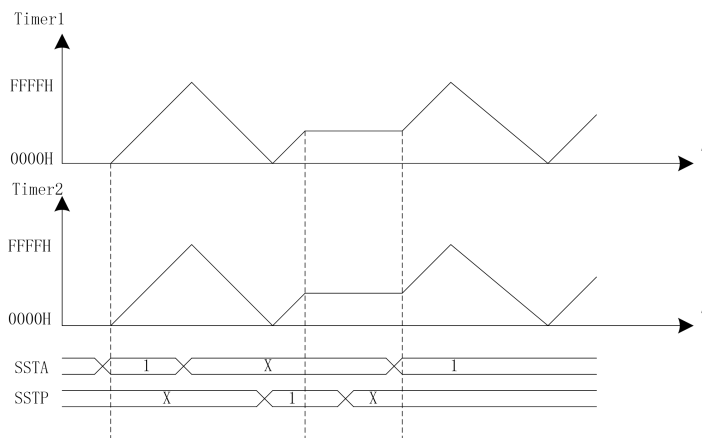
TIMER1/2 的 TIMX_CHA、TIMX_CHB 端口以及刹车输入 ADC 比较输出和 BRKIN 管脚输入都有数字滤波功能。可通过设定 PA_FILTER_EN/PB_FILTER_EN/TIM1_CR 开启对应端口的滤波功能。滤波时钟为计数器当前工作时钟。

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。



3.8.2.8 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 的同步启动。



软件同步暂停

TIMER1/2 可通过设定软件同步停止寄存器 (SSCONR)，实现目标 TIMER1/2 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。

软件同步停止

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

软件同步动作相关寄存器 (SSCONR) 是一组独立于 TIMER1/2 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。

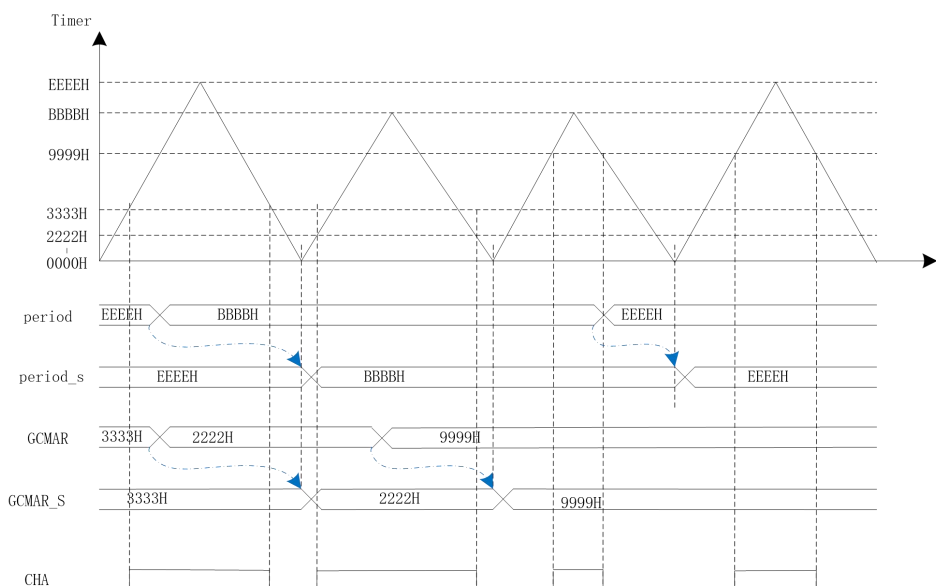


3.8.2.9 缓存功能

缓存动作是指在缓存传送时间点，发生以下事件：

- a. 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；
- b. 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)；

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。



缓存传送时间点：

周期值缓存传送时间点为锯齿波时递加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。

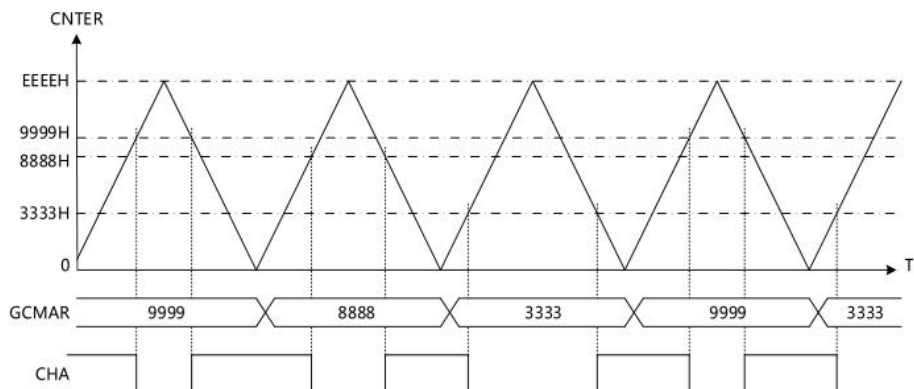
捕获输入动作缓存传送时间点为捕获输入动作时。

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。

3.8.2.10 通用 PWM 输出

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。(输出 PWM 波，则需要配置 TIMx_BRAKE 的 T1x_MOE 为 1)

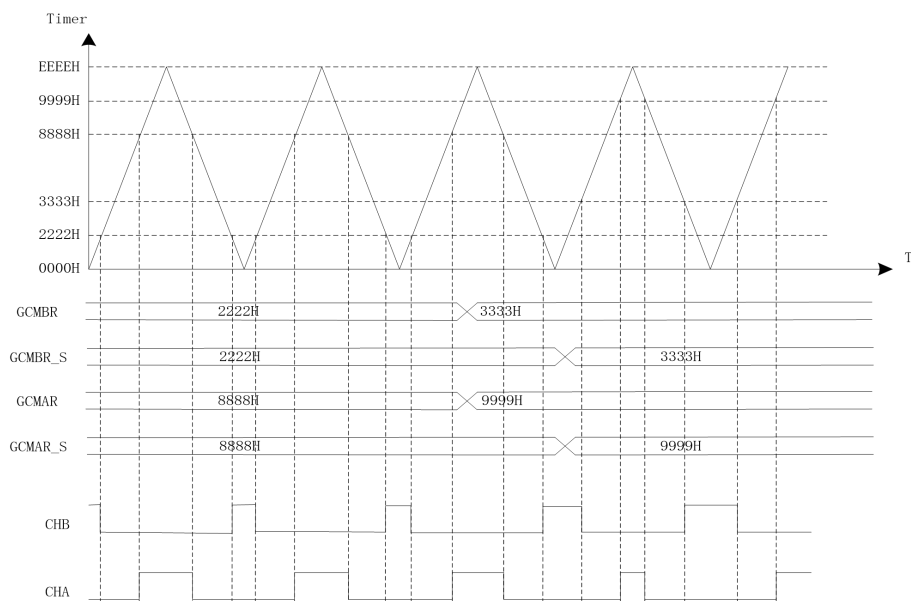


2. 互补 PWM 输出

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。（输出 PWM 波，则需要配置 TIMx_BRAKE 的 TIx_MOE 为 1）。

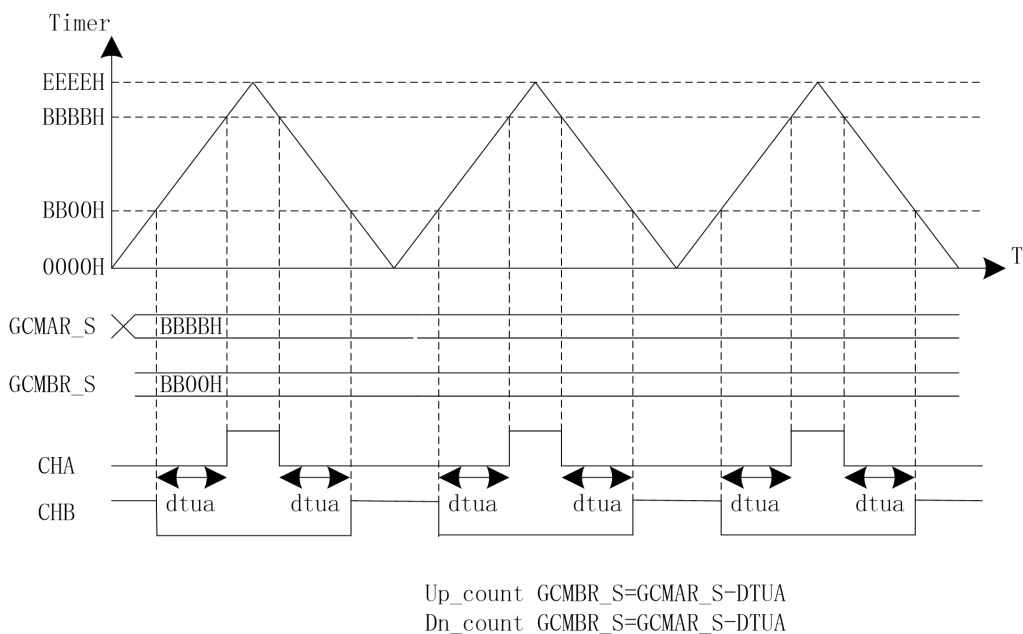
1) 软件设定 GCMBR 互补 PWM 输出

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。



2) 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 1~255 (计数步长时间)。当死区使能时 DTUA 不能为 0。死区使能时，在死区中 PWM 波输出的信号由对应管脚的 GPIO 配置决定，即相应管脚的 DM/DR 寄存器决定。DM/DR 寄存器可以根据需要配置成高阻或者固定高/低电平。（输出 PWM 波，则需要配置 TIMx_BRAKE 的 TIx_MOE 为 1）



PWM 互补输出时，死区可以通过配置寄存器

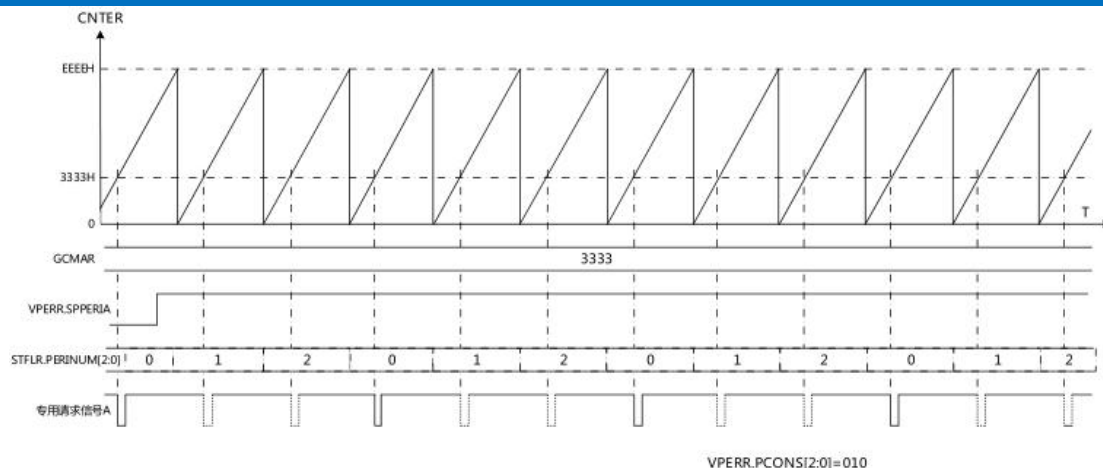
TIM1_DTR. DTH_A/TIM1_DTR. DTH_B/TIM2_DTR. DTH_A/TIM2_DTR. DTH_B 来使能 HALF 功能。HALF 功能使能后，相应 PWM 波输出的死区信号会少半个 16M 时钟周期宽度即 31.25ns。可以用于更细的死区控制，如果用户需要死区宽度为 16M 时钟周期的整数倍，则不需要使能 HALF 功能，如果用户需要半个 16M 时钟周期的奇数倍时则需要使能 HALF 功能。这种场景下，PWM 波实际的输出效果相当于 DTUA 寄存器的值减去 0.5，即： $DTUAreai = DTUA - 0.5$

而当 PWM 互补输出死区使能时，DTUA 最小配置为 1，那么实际产生的死区宽度就可以为 16M 时钟周期的一半即 31.25ns。注意：应用 HALF 功能时，系统时钟只能设置为 16M，不能进行分频。

3.8.2.11 周期间隔响应

TIMER1/2 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。



3.8.2.12 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 ADC、外部引脚 BKIN--DP1_2），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）。

3.8.2.13 中断说明

TIMER1/2 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。

3.8.2.14 内部互连

- ADC 输出可以触发刹车功能。
- 外部 BKIN 引脚 (DP1_2) 可以触发刹车功能。
- TIMER1/2 的 PWM 波输出可以触发 ADC 采样功能。

3.8.2.15 timer2 捕获 timer1

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

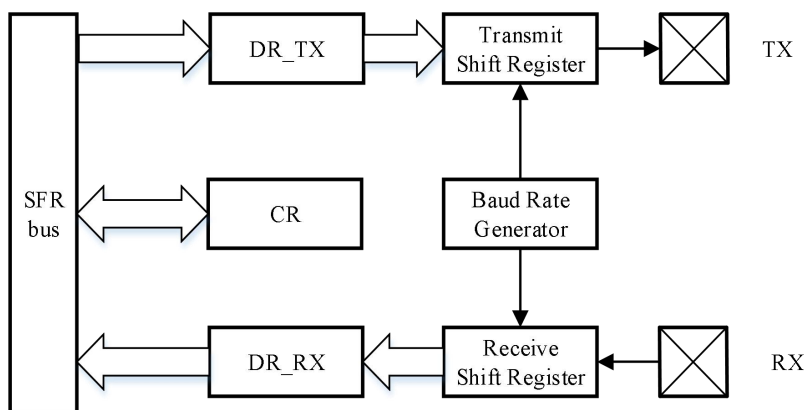
3.8.3 UART

3.8.3.1 概述

850x 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
- 与 GPIO 配合可以兼容半双工即 TX 和 RX 使用同一个 PIN
- 异步模式
- LSB 在前
- 集成波特率发生器
- 8 位数据
- 支持奇偶校验
- 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）

3.8.3.2 结构框图



UART 结构框图

3.8.3.3 时钟发生器

时钟源来自系统时钟 SCK1、SCK2 和 SCK3 中的一个。

3.8.3.4 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。



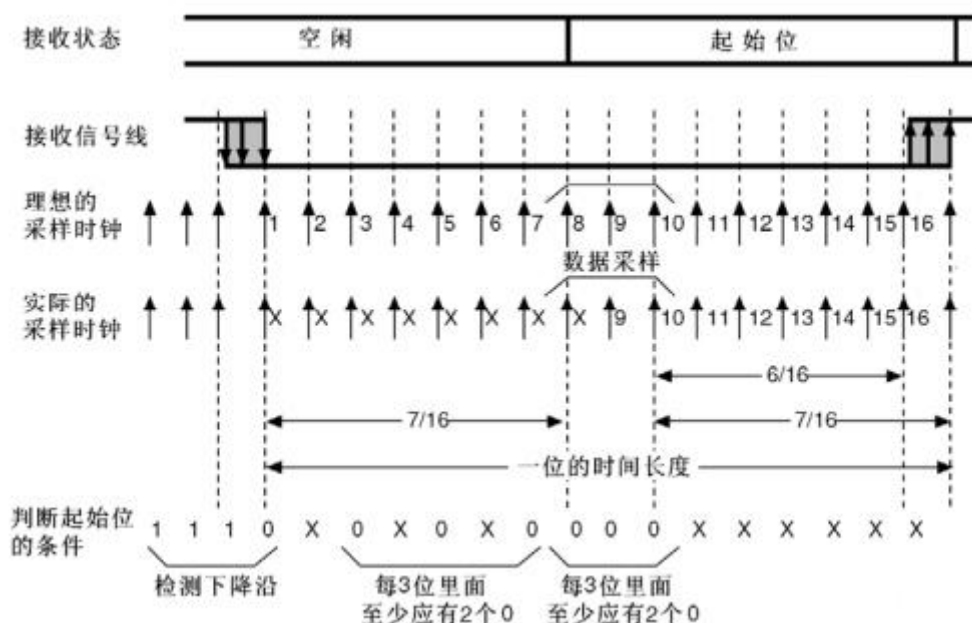
UART 根据以下顺序在 Tx 引脚发送数据：起始位，8 位数据位（低位在前），停止位。

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。

3.8.3.5 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。

接收时序图如下：



3.8.3.6 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。



3.8.3.7 UART 半双工

UART 半双工的应用方式是使用 UART0/1_CR.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。

3.8.3.8 波特率设置

在信息传输通道中，携带数据信息的信号单元叫作码元（因为串口是 1bit 进行传输的，所以其码元就代表一个二进制数），每秒通过信号传输的码元数称为码元的传输速率，简称“波特率”，常用符号“Baud”表示，其单位为“波特每秒”（Bps）。串口常见的波特率有 4800、9600、115 200 等，此处我们选用 9600 的波特率进行讲解。

波特率时钟来自 SCK1、SCK2、SCK3 三个时钟源。

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。

波特率的计算公式为：

$$\text{Baud} = F_{\text{CLK}}/4*\text{时钟源}$$

通过 UART1_CFG 的 CKSEL 配置 UART 时钟源

配置 PCLK_DIV12 和 PCLK_DIV3 来配置所需要的波特率

例如，配置波特率为 9600bps：

$$9600 = 16000000/4*\text{时钟源}$$

$$\text{时钟源} = 4000000/9600 \approx 416.6$$

由于 UART 使用的是时钟源的 4 分频，故时钟源分频系数为：

$$416.6/4 \approx 104$$

故 SCK3 的分频系数为 104

```
IM0_CR    = 0xC0;    //SYS_CLK = 16M
PCLK_CR   = 0xF3;    //CLK_SYS 作 SCK2,SCK3 时钟源为 SCK2
PCLK_DIV12 = 0x83;   //SCK1=System/9, SCK2 = System/4
PCLK_DIV3  = 103;   //SCK3 = SCK2/104
```

3.8.4 I2C

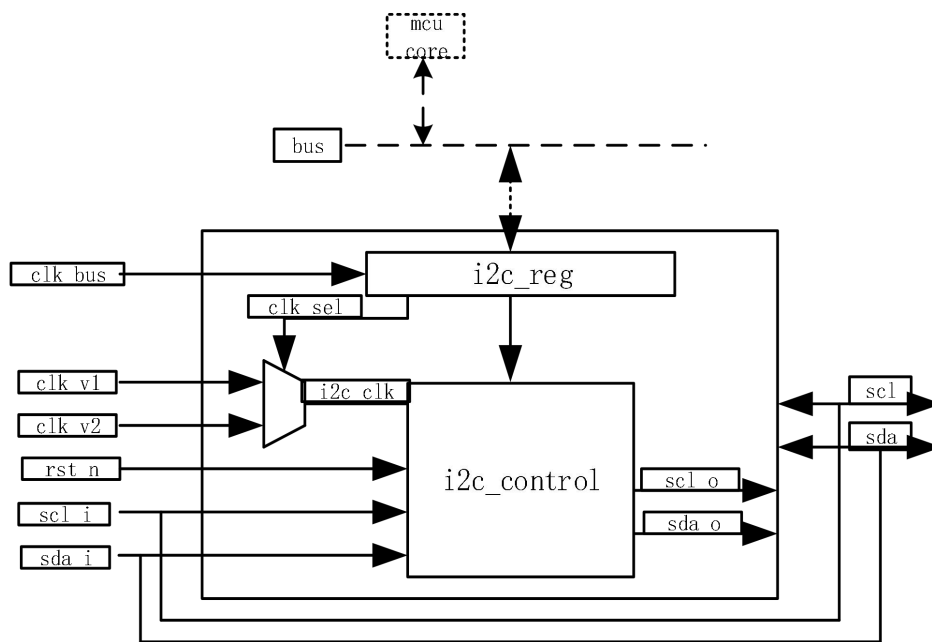
3.8.4.1 概述

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

3.8.4.2 结构框图



I2C 结构框图

3.8.4.3 应用描述

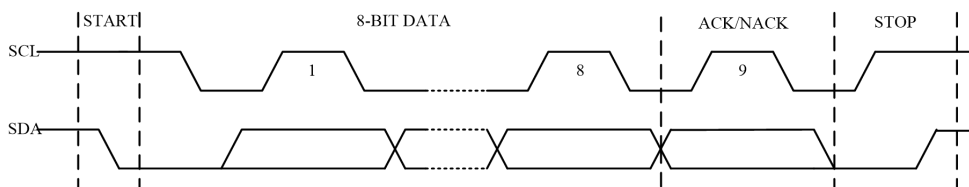
I2C 支持主从模式下的数据发送和接收。

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；

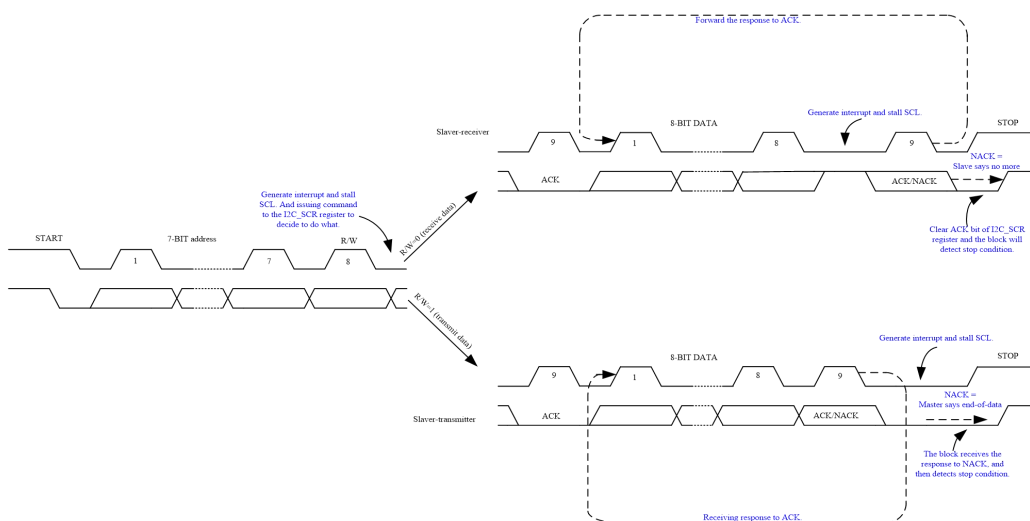
- SCL 为高时，检测到 SDA 上有由低到高的跳变，为 STOP。



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



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

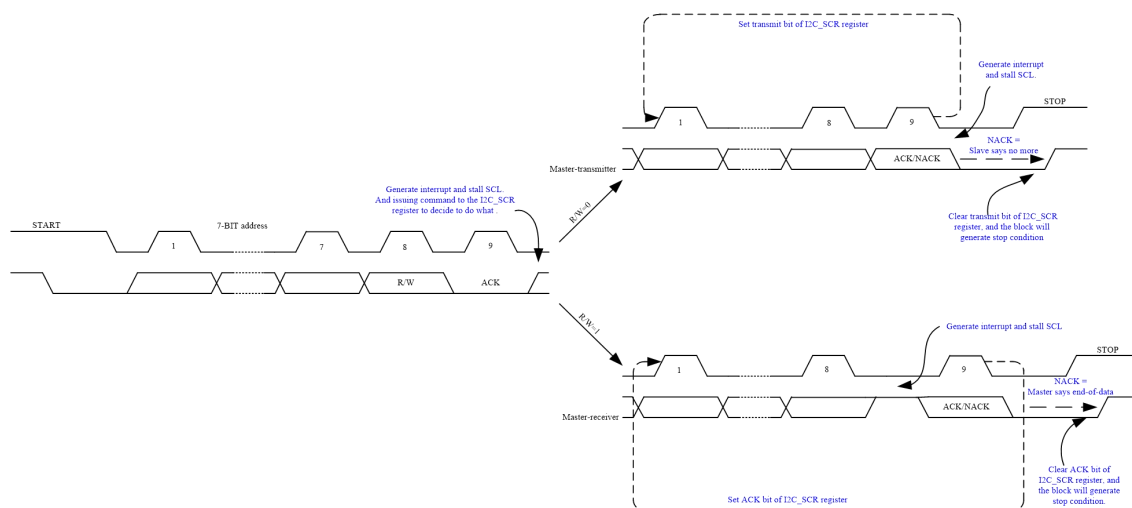


从器件成功接收 1byte 数据过程如下:

- 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。



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。

3.8.4.4 中断

I2C 提供 5 种类型的中断:

- 1) 总线错误中断
- 2) 停止中断
- 3) NACK 中断
- 4) 硬件地址匹配中断
- 5) 传输完成中断

3.8.4.5 波特率设置

主机模式下, 发送时钟来自时钟源的 17 分频。

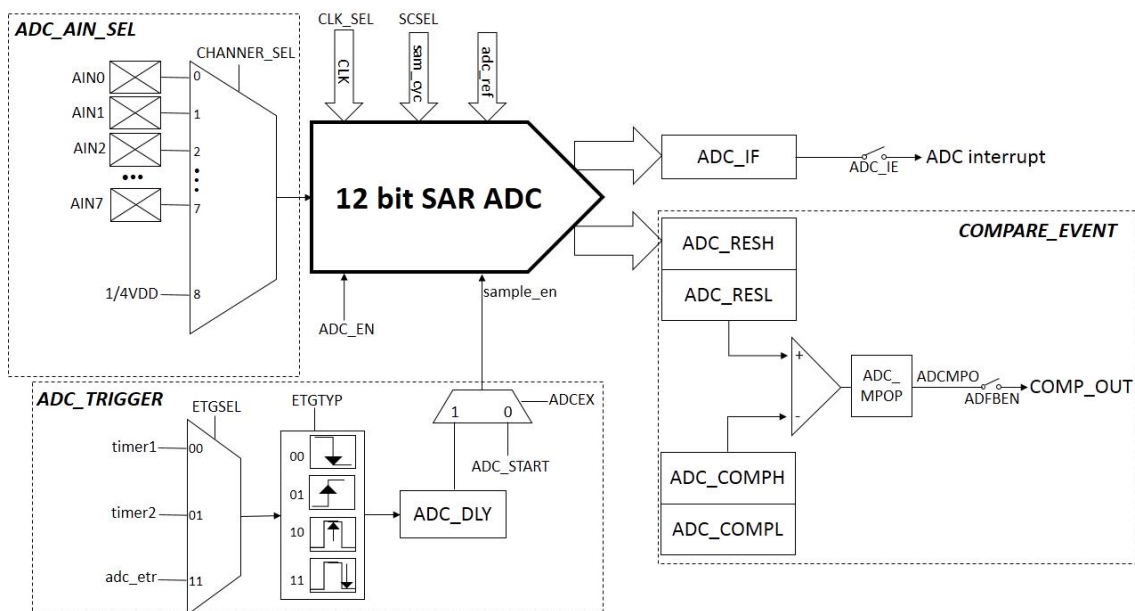
3.8.5 ADC

XC8F9362 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 9 路可选的单端输入通道：8 路外部输入通道，1 路片内 1/4VDD 电压输入通道；
注意：如果当前采样的外部输入通道的电压大于 VDD 电压，则采样结果会不对，有可能为全 0；如果外部输入通道输入的电电压大于 VDD 电压，而此时采样内部通道，则会导致采样不准，结果会比实际理论上的值要大；
- 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、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

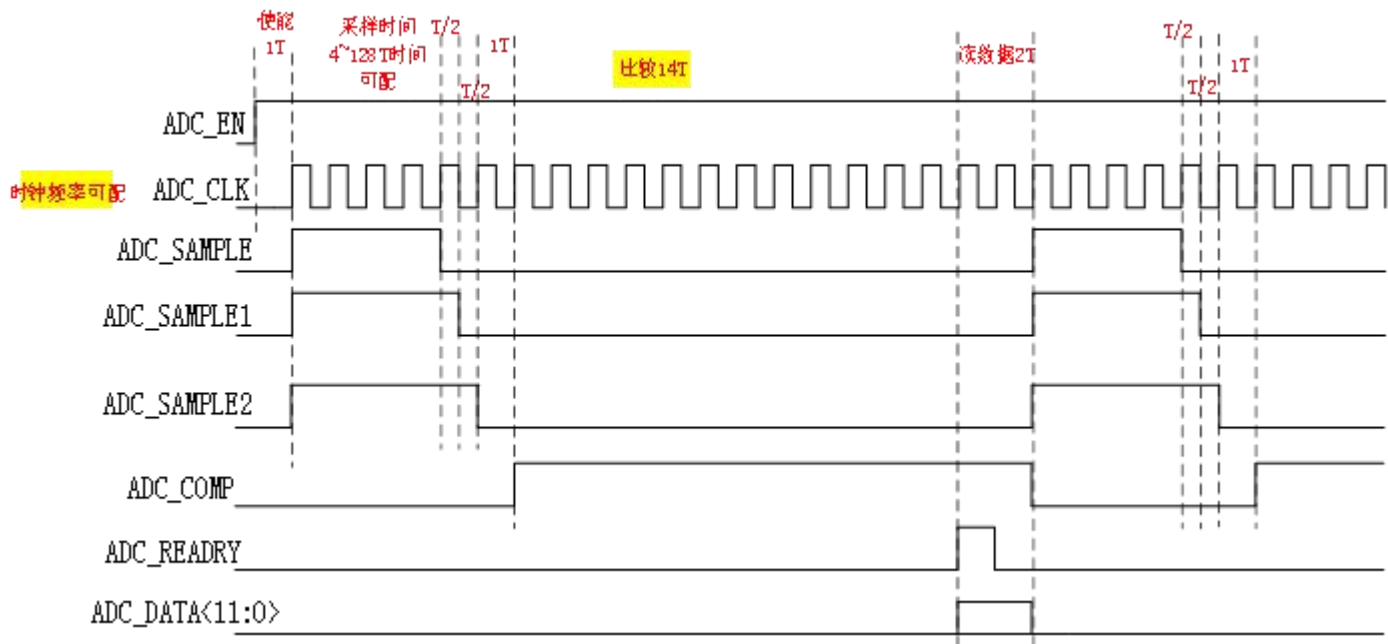
3.8.5.1 结构框图



ADC 结构框图

3.8.5.2 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)。



ADC 时序图

3.9 省电模式

XC8F9362 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。

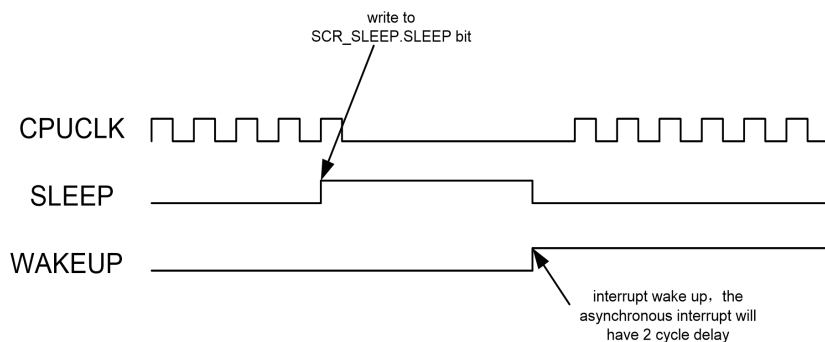
外设	睡眠模式	深度睡眠模式
CPU	停止	停止
RAM	保持	保持
睡眠定时器	运行	运行
看门狗	运行	运行
定时器 0~2	运行	停止
ADC	运行	停止
UART0/1	运行	停止
I2C	运行	停止
内部 16MHz 振荡器	运行	停止
内部 32KHz 振荡器	运行	运行
I/O 口	保持	保持
其他外设	运行	停止



唤醒条件	看门狗复位, 所有中断	看门狗复位, 引脚中断, 睡眠定时器中断
------	-------------	----------------------

3.9.1 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。



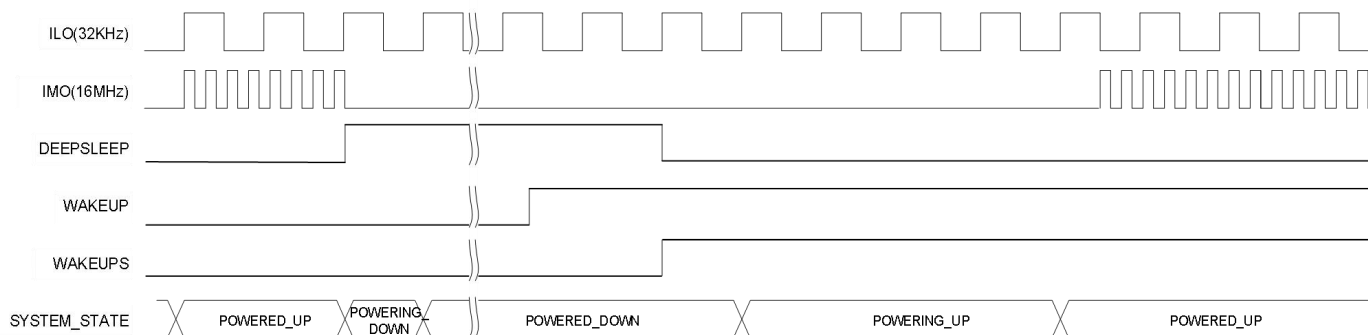
睡眠模式睡眠和唤醒时序图

3.9.2 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。

3.9.2.1 深度睡眠模式唤醒

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睡眠唤醒的时序。



深度睡眠唤醒时序



3.9.3 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 SLPACSEL 等于 0 时，通过 SLPINTS 选择固定的定时周期，第二种 SLPACSEL 等于 1 时，选择对于的溢出值来产生中，溢出值通过 SLPACPRD[10: 0]来配置。

操作说明：

对睡眠定时器或者看门狗定时器的操作应遵循以下操作顺序：

1. SLPTIM_CLR 和 SLPTIM_SR 写 0x00 清除睡眠计数器和看门狗，关闭睡眠定时器和看门狗定时器的时钟；
2. 进行睡眠定时器和看门狗定时器的时钟选择或者配置寄存器 SLPTIM_PRDL、SLPTIM_PRDRH 和 SLPTIM_CR；
3. 配置 SLPTIM_CR 的 SLEEPDIS 位为 0，使能睡眠定时器的时钟；



4. 电气特性

4.1 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	-20	85	°C
工作电压	2.4	5.5	V
VDD 对地电压	-0.3	6.6	V
I/O 对地电压	-0.3	VDD+0.3	V
VPP 对地电压 (仅限烧录模式)	9.6	10	V

4.2 直流特性

符号	参数	测试条件	最小值	典型值	最大值	单位
f_{FLASH}	FLASH 工作频率	$4.5 \leq VDD < 5.5$	-	-	8	MHz
		$2.4V \leq VDD < 5.5V$	-	-	2.6	MHz
IDD1	工作电流 1	内部 16MHz RC 振荡器工作, CPU 工作在 16MHz	-	5.5	-	mA
IDD2	工作电流 2	内部 16MHz RC 振荡器工作, CPU 关闭	-	3.6	-	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	-	4	11	uA
VIL	输入低电平	VDD=5V, 常温 25°C	-	-	0.3VDD	V
VIH	输入高电平		0.5VDD	-	-	V
R _{PU}	上拉电阻		-	10	-	KΩ
R _{PD}	下拉电阻		-	10	-	KΩ
I _{OH1}	拉电流 1	V _{pin} =3.5V (所有 GPIO 输出电流一样, P11 除外 (12.8))	-	18	-	mA
		V _{pin} =4V (所有 GPIO 输出电流一样, P11 除外 (8.5))	-	11.8	-	mA
I _{OL1}	灌电流 1	V _{pin} =0.3V (所有 GPIO 输出电流一样, P11 除外 (8))	-	12	-	mA
		V _{pin} =0.8V (所有 GPIO 输出电流一样, P11 除外 (19))	-	28	-	mA



4.3 ADC 特性

电源工作电压 2.6V—5.5V，典型工作电源电压 3.6V，如无特别说明，下表中参数均代表环境温度为 25°C。

规格	条件	最小值	典型值	最大值	单位
分辨率	单端转换	–	12	–	bit
积分非线性误差	–	–	±2	±6	LSB
微分非线性误差	–	–	±1	±2	LSB
增益误差	–	–	±1	±2	LSB
偏移误差	–	–	±2	±6	LSB
模拟供电电压	–	2.4	5	5.5	V
参考正电压 (可配置)	片内参考	–	1.2	–	V
	片内参考	–	2.4	–	V
	电源	–	VDD	–	V

4.4 EMC 特性

Electrostatic discharge (ESD)

符号	参数	条件	封装	最大值	单位
$V_{ESD(HBM)}$	Electrostatic discharge voltage (Human body model)	Temperature = +24°C Relative humidity 45%	TSSOP20	8000	V
$V_{ESD(CDM)}$	Electrostatic discharge voltage (Charge device model)	Temperature = +24°C Relative humidity 45%		2000	V
$V_{ESD(MM)}$	Electrostatic discharge voltage (Machine model)	Temperature = +24°C Relative humidity 40%		550	V

5. 指令集

5.1 指令集注释

芯片有五种不同的寻址方式:立即、直接、寄存器、间接和相对。在立即寻址模式下,数据包含在操作码中。通过直接寻址,8位地址是操作码的一部分,通过寄存器寻址,在操作码中选择一个寄存器进行操作。间接寻址方式下,在操作码中选择一个寄存器,以指向操作使用的地址。相对寻址模式用于跳转指令。下表概述了微控制器核心的指令集周期。一个周期等于一个时钟周期。表1和表2包含用于指令集表的助记符注释。表3-7显示了每条指令执行所需的十六进制代码、字节数和机器周期。

Rn	Working register R0-R7
direct	128 internal RAM locations, any Special Function Registers
@Ri	Indirect internal or external RAM location addressed by register R0 or R1
#data	8-bit constant included in instruction
#data16	16-bit constant included as bytes 2 and 3 of instruction
bit	256 software flags, any bit-addressable I/O pin, control or status bit
A	Accumulator

Table 1. Notes on data addressing modes

addr16	Destination address for LCALL and LJMP may be anywhere within the 64-Kbyte of program memory address space.
addr11	Destination address for ACALL and AJMP will be within the same 2-Kbyte page of program memory as the first byte of the following instruction.
rel	SJMP and all conditional jumps include an 8-bit offset byte. Range is +127/-128 bytes relative to the first byte of the following instruction

Table 2. Notes on program addressing modes



5.2 指令集

5.2.1 算术运算

Mnemonic	Description	Code	Bytes	Cycles
ADD A, Rn	Add register to accumulator	0x28-0x2F	1	1
ADD A, direct	Add direct byte to accumulator	0x25	2	2
ADD A, @Ri	Add indirect RAM to accumulator	0x26-0x27	1	2
ADD A, #data	Add immediate data to accumulator	0x24	2	2
ADDC A, Rn	Add register to accumulator with carry flag	0x38-0x3F	1	1
ADDC A, direct	Add direct byte to A with carry flag	0x35	2	2
ADDC A, @Ri	Add indirect RAM to A with carry flag	0x36-0x37	1	2
ADDC A, #data	Add immediate data to A with carry flag	0x34	2	2
SUBB A, Rn	Subtract register from A with borrow	0x98-0x9F	1	1
SUBB A, direct	Subtract direct byte from A with borrow	0x95	2	2
SUBB A, @Ri	Subtract indirect RAM from A with borrow	0x96-0x97	1	2
SUBB A, #data	Subtract immediate data from A with borrow	0x94	2	2
INC A	Increment accumulator	0x04	1	1
INC Rn	Increment register	0x08-0x0F	1	2
INC direct	Increment direct byte	0x05	2	3
INC @Ri	Increment indirect RAM	0x06-0x07	1	3
DEC A	Decrement accumulator	0x14	1	1
DEC Rn	Decrement register	0x18-0x1F	1	2
DEC direct	Decrement direct byte	0x15	1	3
DEC @Ri	Decrement indirect RAM	0x16-0x17	2	3
INC DPTR	Increment data pointer	0xA3	1	1
MUL A, B	Multiply A and B	0xA4	1	2
DIV A, B	Divide A by B	0x84	1	6
DA A	Decimal adjust accumulator	0xD4	1	3

Table 3. Arithmetic operations



5.2.2 逻辑运算

Mnemonic	Description	Code	Bytes	Cycles
ANL A, Rn	AND register to accumulator	0x58-0x5F	1	1
ANL A, direct	AND direct byte to accumulator	0x55	2	2
ANL A, @Ri	AND indirect RAM to accumulator	0x56-0x57	1	2
ANL A, #data	AND immediate data to accumulator	0x54	2	2
ANL direct, A	AND accumulator to direct byte	0x52	2	3
ANL direct, #data	AND immediate data to direct byte	0x53	3	3
ORL A, Rn	OR register to accumulator	0x48-0x4F	1	1
ORL A, direct	OR direct byte to accumulator	0x45	2	2
ORL A, @Ri	OR indirect RAM to accumulator	0x46-0x47	1	2
ORL A, #data	OR immediate data to accumulator	0x44	2	2
ORL direct, A	OR accumulator to direct byte	0x42	2	3
ORL direct, #data	OR immediate data to direct byte	0x43	3	3
XRL A, Rn	Exclusive OR register to accumulator	0x68-0x6F	1	1
XRL A, direct	Exclusive OR direct byte to accumulator	0x65	2	2
XRL A, @Ri	Exclusive OR indirect RAM to accumulator	0x66-0x67	1	2
XRL A, #data	Exclusive OR immediate data to accumulator	0x64	2	2
XRL direct, A	Exclusive OR accumulator to direct byte	0x62	2	3
XRL direct, #data	Exclusive OR immediate data to direct byte	0x63	3	3
CLR A	Clear accumulator	0xE4	1	1
CPL A	Complement accumulator	0xF4	1	1
RL A	Rotate accumulator left	0x23	1	1
RLC A	Rotate accumulator left through carry	0x33	1	1
RR A	Rotate accumulator right	0x03	1	1
RRC A	Rotate accumulator right through carry	0x13	1	1
SWAP A	Swap nibbles within the accumulator	0xC4	1	1

Table 4. Logic operations



5.2.3 布尔操作

Mnemonic	Description	Code	Bytes	Cycles
CLR C	Clear carry flag	0xC3	1	1
CLR bit	Clear direct bit	0xC2	2	3
SETB C	Set carry flag	0xD3	1	1
SETB bit	Set direct bit	0xD2	2	3
CPL C	Complement carry flag	0xB3	1	1
CPL bit	Complement direct bit	0xB2	2	3
ANL C, bit	AND direct bit to carry flag	0x82	2	2
ANL C, /bit	AND complement of direct bit to carry	0xB0	2	2
ORL C, bit	OR direct bit to carry flag	0x72	2	2
ORL C, /bit	OR complement of direct bit to carry	0xA0	2	2
MOV C, bit	Move direct bit to carry flag	0xA2	2	2
MOV bit, C	Move carry flag to direct bit	0x92	2	3

Table 5. Boolean manipulation



5.2.4 数据传送

Mnemonic	Description	Code	Bytes	Cycles	
MOV A, Rn	Move register to accumulator	0xE8-0xEF	1	1	
MOV A, direct	Move direct byte to accumulator	0xE5	2	2	
MOV A, @Ri	Move indirect RAM to accumulator	0xE6-0xE7	1	2	
MOV A, #data	Move immediate data to accumulator	0x74	2	2	
MOV Rn, A	Move accumulator to register	0xF8-0xFF	1	1	
MOV Rn, direct	Move direct byte to register	0xA8-0xAF	2	3	
MOV Rn, #data	Move immediate data to register	0x78-0x7F	2	2	
MOV direct, A	Move accumulator to direct byte	0xF5	2	2	
MOV direct, Rn	Move register to direct byte	0x88-8F	2	2	
MOV	Move direct byte to direct byte	85	3	3	
MOV direct, @Ri	Move indirect RAM to direct byte	86-87	2	3	
MOV direct, #data	Move immediate data to direct byte	75	3	3	
MOV @Ri, A	Move accumulator to indirect RAM	F6-F7	1	2	
MOV @Ri, direct	Move direct byte to indirect RAM	A6-A7	2	3	
MOV @Ri, #data	Move immediate data to indirect RAM	76-77	2	2	
MOV DPTR, #data16	Load data pointer with a 16-bit constant	90	3	3	
MOVC A, @A+DPTR	Move code byte relative to DPTR to	93	1	5	
MOVC A, @A+PC	Move code byte relative to PC to accumulator	83	1	4	
MOVX A, @Ri	Move external RAM (8-bit address) to A	E2-E3	1	3*	
MOVX A, @DPTR	Move external RAM (16-bit address) to A	E0	1	2*	
MOVX @Ri, A	Move A to external RAM (8-bit address)	CODE inside ROM/RAM	F2-F3	1	4*
		all other cases			5*
MOVX @DPTR, A	Move A to external RAM (16-bit)	CODE inside ROM/RAM	F0	1	3*
		all other cases			4*
PUSH direct	Push direct byte onto stack	C0	2	3	
POP direct	Pop direct byte from stack	D0	2	2	
XCH A, Rn	Exchange register with accumulator	C8-CF	1	2	
XCH A, direct	Exchange direct byte with accumulator	C5	2	3	
XCH A, @Ri	Exchange indirect RAM with accumulator	C6-C7	1	3	
XCHD A, @Ri	Exchange low-order nibble indirect RAM with	D6-D7	1	3	

Table 6. Data transfer

* MOVX cycles depends on STRETCH register. Table shows values with STRETCH=0.



5.2.5 分支指令

Mnemonic	Description	Code	Bytes	Cycles
ACALL addr11	Absolute subroutine call	0x11-0xF1	2	4
LCALL addr16	Long subroutine call	03	3	4
RET	Return from subroutine	22	1	4
RETI	Return from interrupt	32	1	4
AJMP addr11	Absolute jump	01-E1	2	3
LJMP addr16	Long jump	02	3	4
SJMP rel	Short jump (relative address)	80	2	3
JMP @A+DPTR	Jump indirect relative to the DPTR	73	1	5
JZ rel	Jump if accumulator is zero	60	2	4
JNZ rel	Jump if accumulator is not zero	70	2	4
JC rel	Jump if carry flag is set	40	2	3
JNC	Jump if carry flag is not set	50	2	3
JB bit,rel	Jump if direct bit is set	20	3	5
JNB bit,rel	Jump if direct bit is not set	30	3	5
JBC bit,direct rel	Jump if direct bit is set and clear bit	10	3	5
CJNE A,direct rel	Compare direct byte to A and jump if not equal	B5	3	5
CJNE A,#data rel	Compare immediate to A and jump if not equal	B4	3	4
CJNE Rn,#data rel	Compare immediate to reg. and jump if not	B8-BF	3	4
CJNE @Ri,#data rel	Compare immediate to ind. and jump if not	B6-B7	3	5
DJNZ Rn,rel	Decrement register and jump if not zero	D8-DF	2	4
DJNZ direct,rel	Decrement direct byte and jump if not zero	D5	3	5
NOP	No operation	00	1	1

Table 7. Program branches

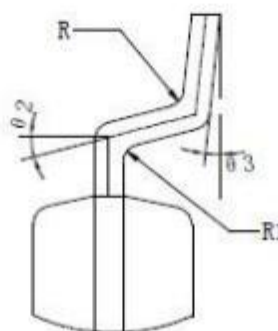
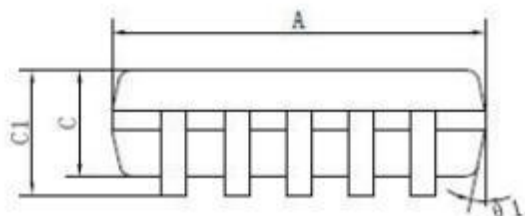
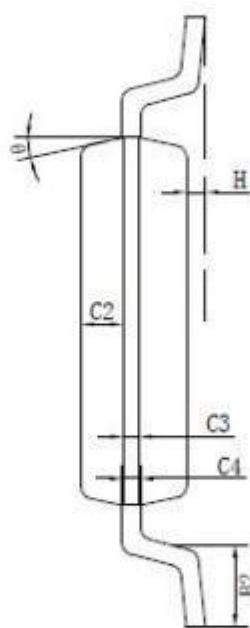
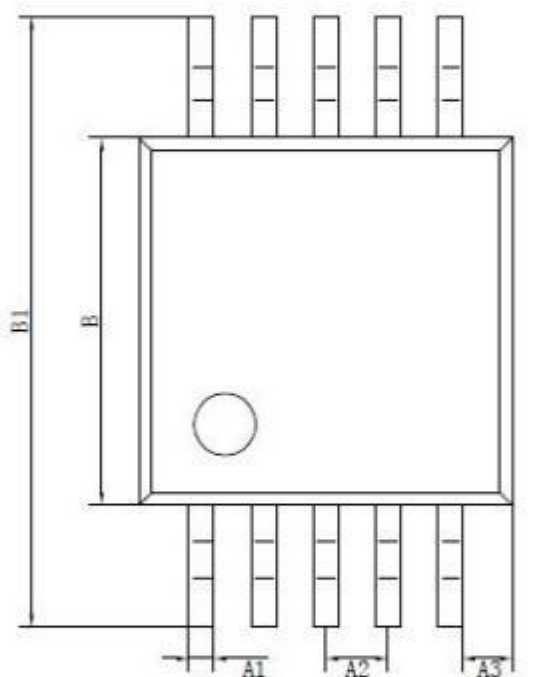


6. 封装尺寸

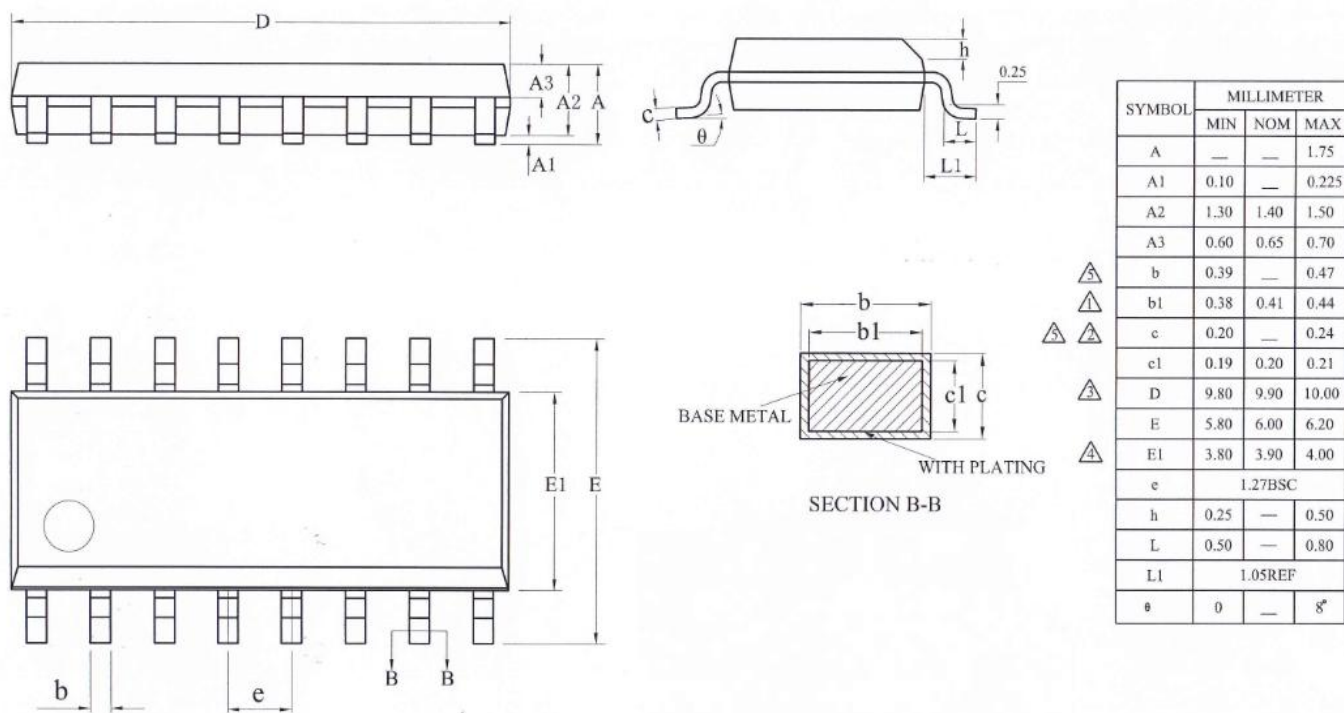
(单位: mm)

标注	尺寸	最小(mm)	最大(mm)	标注	尺寸	最小(mm)	最大(mm)
A		2.90	3.10	C3		0.152	
A1		0.18	0.25	C4		0.15	0.23
A2		0.50TYP		H		0.00	0.09
A3		0.40TYP		θ		15° TYP4	
B		2.90	3.10	$\theta 1$		12° TYP4	
B1		4.70	5.10	$\theta 2$		14° TYP	
B2		0.45	0.75	$\theta 3$		0° ~ 6°	
C		0.75	0.95	R		0.15TYP	
C1		--	1.10	R1		0.15TYP	
C2		0.328TYP					

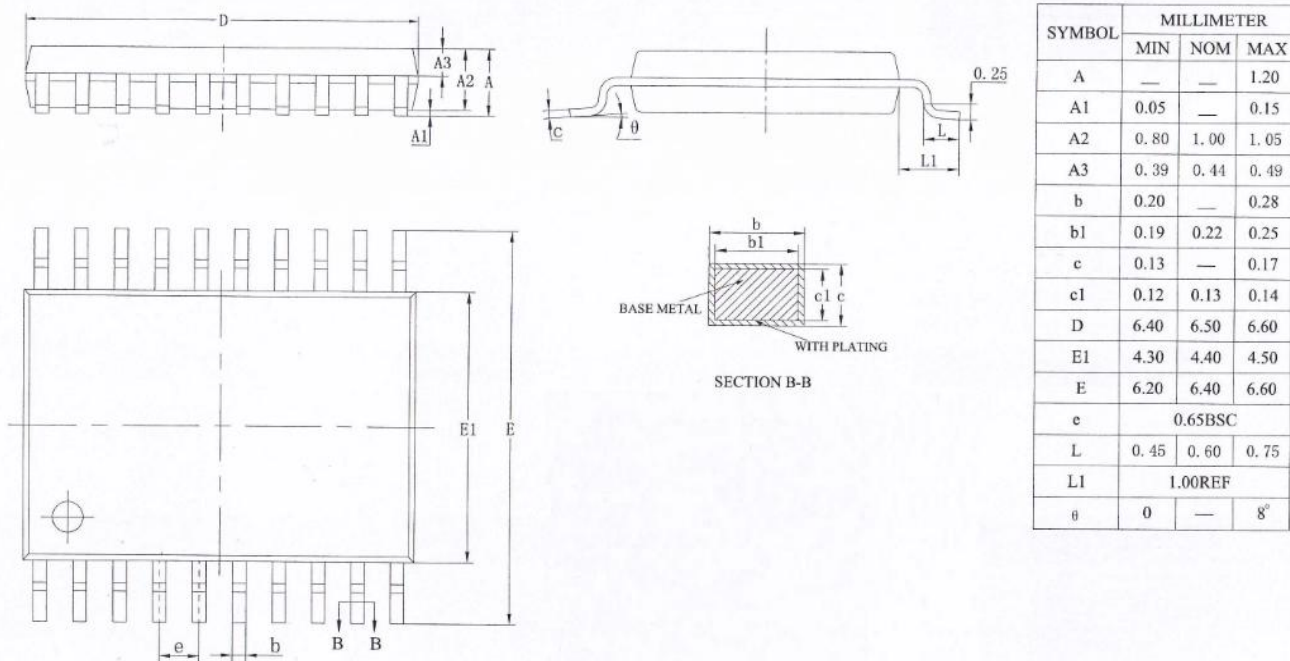
* 注EMSOP10产品共用此图所有数据, Die pad exposure大小是根据引线框架设计。



MSOP10封装外形图



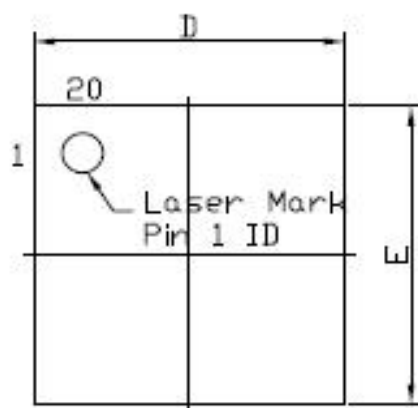
SOP16 封装外形图



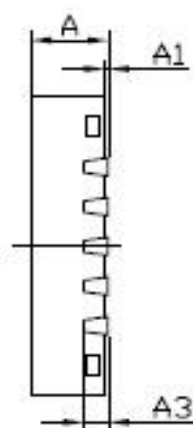
TSSOP20 封装外形图



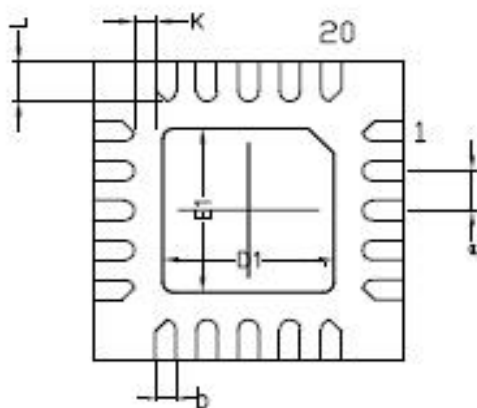
标注	尺寸	最小	标准	最大	标注	尺寸	最小	标准	最大
A		0.70	0.75	0.80	D1		1.55	1.65	1.75
A1		0.00	—	0.05	E1		1.55	1.65	1.75
A3		0.203REF			e		0.40TYP		
b		0.15	—	0.25	k		0.20	—	—
D		2.90	3.00	3.10	L		0.30	0.40	0.50
E		2.90	3.00	3.10					



Top View



Side View



bottom View

QFN20 封装外形图