

## 1、概述

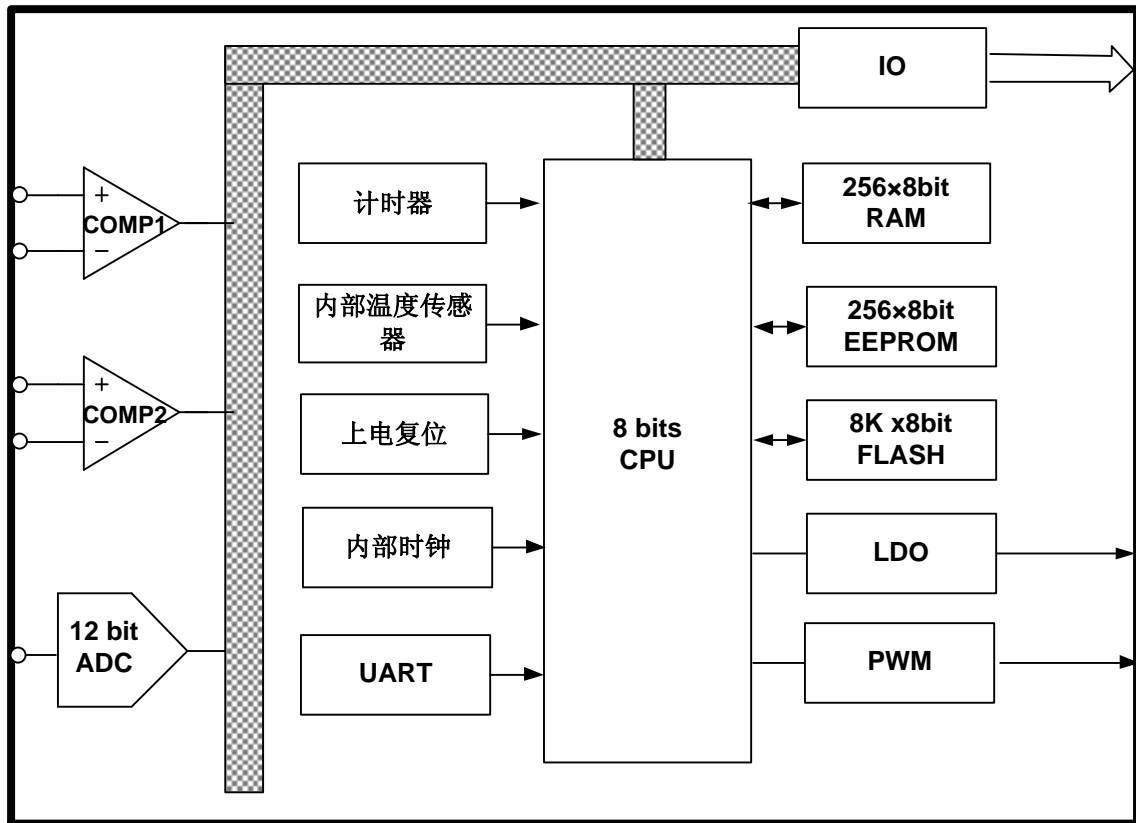
CS88F313 是一款具有 ADC 功能的低功耗可编程传感器专用检测电路，可用于电子测量仪器、环境监控、手持式测量工具、家庭应用、电子控制工具、马达控制、家庭安全系统以及其它方面。

### 特点

- 高性能 RISC CPU:
  - 单指令周期的超快速 8 位 CPU
  - 完全兼容 MCS-51 指令
- 灵活的振荡器结构:
  - 内建 16M/1M 高速系统时钟振荡器
  - 内建 8K 低速时钟振荡器
- 单片机特性:
  - 工作电压范围：2.2V~5.5V
  - 工作温度范围：-40℃~85℃
  - FLASH：8K×8bit 擦写次数大于 1000 次
  - RAM：内部 256×8bit
  - 数据存储（EEPROM）：256×8bit，擦写次数大于 2 万次
  - 看门狗定时器
- 低功耗管理：
  - 多种低功耗工作模式可选
  - 待机功耗：<2μA @3V
- 模拟特性：
  - 模数转换器（ADC）：12 位分辨率
  - 内置 LDO：1.8V/2.048V/2.4V/3.072V
  - 内置温度传感器
  - 内置两个比较器（COMP）
- 外设特性：
  - 中断：2 个中断向量，可配置外部中断（所有 IO 口都可作为外部中断）
  - 多达 18 个双向 I/O 端口，且均单独配置上拉功能
- TIMER：三个 16 位，一个 8 位具有溢出中断功能的可编程定时/计数器
- PWM：四通道 16 位 PWM
- 通讯接口：UART
- 封装形式：SOP8：CS88F313AEO  
SOP16（窄体）：CS88F313BEO  
TSSOP20：CS88F313CEO  
SOP14L：CS88F313DEO

## 2、功能框图与引脚说明

### 2.1、功能框图

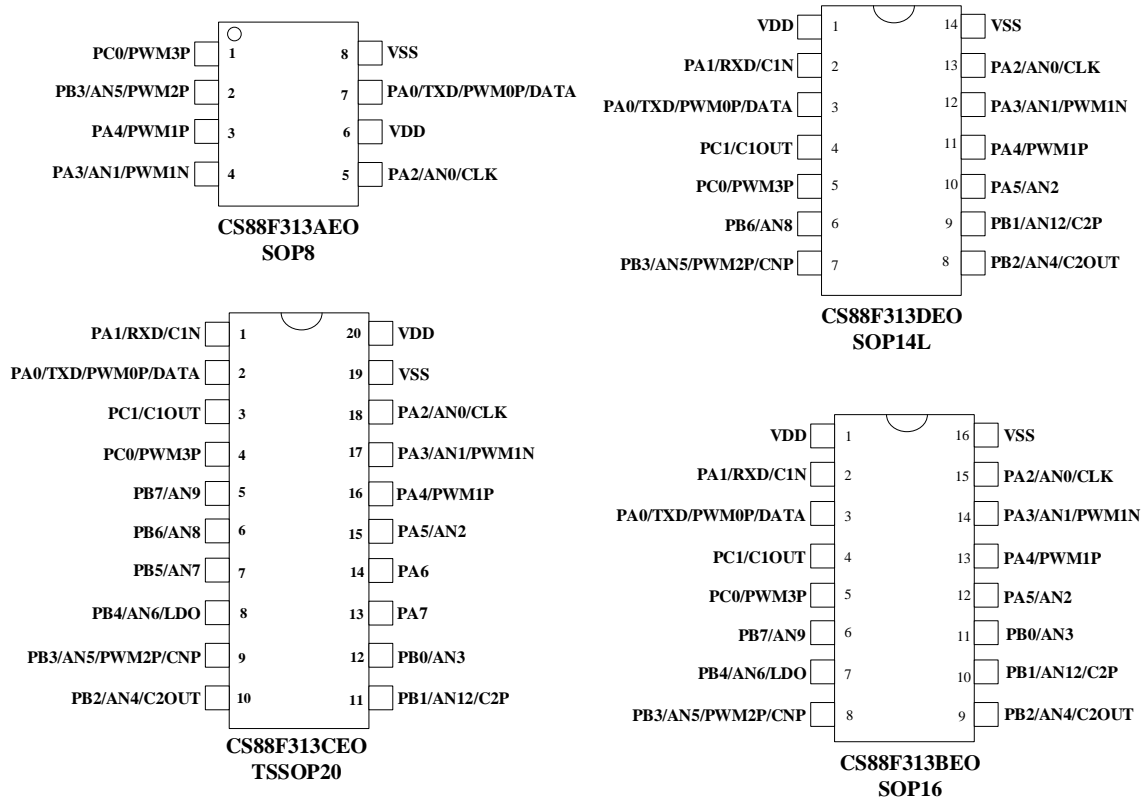


### 2.2、功能描述

CS88F313 是一款具有 ADC 功能的低功耗可编程通用 MCU，专门为需要直接接模拟信号的各式传感器应用产品所设计。内部集成 8 位 CPU 核，8K Bytes FLASH 程序存储器，256 Bytes 数据存储器，256 Bytes RAM 给用户极大的可扩展性。在模拟特性方面，电路包含一个多通道 12 位 A/D 转换器、2 个比较器、四路 PWM 输出、一个可配置的 1.8V、2.048V、2.4V 或 3.072V 的 LDO 电压调节器。内建完整的 UART，为设计者提供了一个易与外部硬件通信的接口。内部看门狗定时器，外加优秀的抗干扰和 ESD 保护性能，确保单片机在恶劣的电磁干扰环境下可靠地运行。

电路还提供一个内置的温度传感器，I/O 使用灵活等其它特性，使电路可以广泛应用于各种采用各式传感器、对外部模拟信号处理控制的产品，例如电子测量仪器、环境监控、手持式测量工具、家庭应用、电子控制工具、马达控制、家庭安全系统以及其它方面。

### 2.3、引脚排列图



### 2.4、引脚说明

引脚	引脚名称	引脚功能描述	属性	说 明
1	PA0/TXD /PWM0P/ DATA	PA0	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		TXD	O	UART 通讯口
		PWM0P	O	PWM0 同向输出端
		DATA	I	串行数据输入/输出
2	PA1/RXD /C1N	PA1	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		RXD	I	UART 通讯口
		C1N	I	比较器 1 反向输入
3	PA2/AN0 /CLK	PA2	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		AN0	I	A/D 通道 0
		CLK	I	串行时钟
4	PA3/AN1 /PWM1N	PA3	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		AN1	I	A/D 通道 1
		PWM1N	O	PWM1 反向输出端

转下页

接上页

引脚	引脚名称	引脚 功能描述	属性	说 明
5	PA4 /PWM1P	PA4	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		PWM1P	O	PWM1 同向输出端
6	PA5/AN2	PA5	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		AN2	I	A/D 通道 2
7	PA6	PA6	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
8	PA7	PA7	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
9	PB0/AN3	PB0	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		AN3	I	A/D 通道 3
10	PB1 /AN12 /C2P	PB1	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		AN12	I	A/D 通道 12
		C2P	I	比较器 2 同向输入
11	PB2/AN4 /C2OUT	PB2	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		AN4	I	A/D 通道 4
		C2OUT	O	比较器 2 输出
12	PB3/AN5 /PWM2P /CNP	PB3	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		AN5	I	A/D 通道 5
		CNP	I	比较器输入
		PWM2P	O	PWM 通道 2 正端
13	PB4/AN6 /LDO	PB4	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		AN6	I	A/D 通道 6
		LDO	O	LDO 输出脚
14	PB5/AN7	PB5	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		AN7	I	A/D 通道 7
15	PB6/AN8	PB6	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		AN8	I	A/D 通道 8
16	PB7/AN9	PB7	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		AN9	I	A/D 通道 9
17	PC0 /PWM3P	PC0	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		PWM3P	O	PWM3 同向输出端
18	PC1 /C1OUT	PC1	I/O	通用 I/O 口，可通过寄存器设置上拉电阻
		C1OUT	O	比较器 1 输出
19	VSS		P	地
20	VDD		P	电源

### 3、电特性

#### 3.1、极限参数（注 1）

除非另有规定， $T_{amb}=25^{\circ}\text{C}$

参 数 名 称	符 号	参 数 范 围	单 位
电源	$V_{DD}$	-0.3~5.5	V
存储温度	$T_{stg}$	-50~125	$^{\circ}\text{C}$
工作环境温度	$T_{amb}$	-40~85	$^{\circ}\text{C}$
数字 IO 电压	$V_{IOIN}$	-0.3~ $V_{DD}+0.3$	V

注 1：最大极限值是指超出该工作范围，芯片有可能损坏。电气参数定义了器件在工作范围内并且在保证特定性能指标的测试条件下的直流和交流电参数规范。

#### 3.2、电特性

##### 3.2.1、直流参数

除非另外规定， $T_{amb}=-40^{\circ}\text{C}\sim 85^{\circ}\text{C}$  及典型应用

参数名称	符号	测 试 条 件		规 范 值			单位
		$V_{DD}$ 电压	条件	最小	典型	最大	
工作电压	$V_{DD}$		$f_{SYS}=16\text{MHz}/1\text{MHz}$	2.2		5.5	V
静态电流 ( $f_{SYS}=0$ )	$I_{STB}$	3V	无负载，空闲 0 模式，系统进入 STOP		1.5	3.0	$\mu\text{A}$
		5V			2.5	5.0	
	$I_{STB1}$	3V	无负载，休眠 0 模式，系统进入 STOP		0.5	1	$\mu\text{A}$
		5V			1	1.5	
输入/输出 口的低电平 输入电压	$V_{IL1}$			0		$0.3 \times V_{DD}$	V
输入/输出 口的高电平 输入电压	$V_{IH1}$			$0.7 \times V_{DD}$		$V_{DD}$	V
输入/输出 口灌电流	$I_{OL}$	3V	$V_{OL}=0.1V_{DD}$	10	20		mA
输入/输出 口源电流	$I_{OH}$	3V	$V_{OH}=0.9V_{DD}$	2.8	10		mA
上拉电阻	$R_{PH}$	3V		40	60	80	k $\Omega$
		5V		10	30	50	k $\Omega$
ADC 工作 电压	$A_{VDD}$			2.2		5.5	V

转下页

接上页

参数名称	符号	测 试 条 件		规 范 值			单位
		V <sub>DD</sub> 电压	条件	最小	典型	最大	
A/D 输入电压	V <sub>AD</sub>			0		V <sub>REF</sub>	V
A/D 输入参考电压范围	V <sub>REF</sub>		A <sub>VDD</sub> =5V	2		V <sub>DD</sub>	V
打开 A/D 增加的功耗	I <sub>ADC</sub>	3V			0.5	1.0	mA
带隙参考缓冲电压	V <sub>BG</sub>			-1%	1.20	+1%	V
内部温度传感器电压	V <sub>TMP</sub>		T <sub>amb</sub> =25℃		1.336		V
内部温度传感器斜率	L <sub>TMP</sub>				-3.4		mV/ ℃

### 3.2.2、交流参数

除非另外规定，T<sub>amb</sub>=-40℃~85℃及典型应用

参数名称	符号	测 试 条 件		规 范 值			单位
		V <sub>DD</sub> 电压	条件	最小	典型	最大	
高频时钟	F <sub>16MHz</sub>	2.2V~5.5V	-40℃~85℃	-2%	16	+2%	MHz
高频时钟	F <sub>1MHz</sub>	2.2V~5.5V	-40℃~85℃	-2%	1	+2%	MHz
低频时钟	F <sub>8KHz</sub>	2.2V~5.5V	-40℃~85℃	-5%	8	+5%	KHz
低压复位时间	t <sub>LVR</sub>			60	120	240	μs
低压中断时间	t <sub>LVD</sub>			60	120	240	μs
LVDO 稳定的时间	t <sub>LVDS</sub>	5V	LVR 关闭 LVD 使能 VBG 就绪	25		100	μs
复位延迟时间	t <sub>RSTD</sub>				100		ms
A/D 时钟周期	t <sub>AD</sub>			0.5			μs
A/D 开启到 A/D 开始工作的时间	t <sub>ON2ST</sub>	2.2V~5.5V		2			μs

## 4、工作原理

### 4.1、中央控制器

CPU 通过内部的系统总线连接到存储器（FLASH、EEPROM、RAM）和外设模块。存储器用来存放程序代码和数据。存储器和寄存器的地址访问都要基于 CPU 的指令集来实现。

所有的中断控制请求都会被收集起来发给 CPU，同时会记录下所有发出中断控制信号。中断控制信号是专门由中断控制模块来产生并保持的。

#### 4.1.1、CPU（中央处理器）

- 1) 1CPU 核心；
- 2) 兼容 8051 的指令集；
- 3) 2 个中断矢量信号；
- 4) 1 个复位矢量信号；
- 5) 系统时钟 0~16MHz。

当 CPU 复位的时候，内部所有的寄存器将会复位到初始状态，堆栈指针强制归 0，中断控制信号置为 1，屏蔽中断信号发射，打开外部中断信号锁定状态。

电路可通过 4 线接口实现在线编程模式：

在 线 烧 录 引 脚 名 称	功 能
DATA（PA0）	串行数据输入/输出
CLK（PA2）	串行时钟
VDD	电源
VSS	地

CPU 的指令集：

算法操作：

Mnemonic	Description	Code	Byte	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 bag	0x38-0x3F	1	1
ADDC A,direct	Add direct byte to A with carry bag	0x35	2	2
ADDC A,@Ri	Add indirect RAM to A with carry bag	0x36-0x37	1	2
ADDC A,#data	Add immediate data to A with carry bag	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



逻辑操作：

Mnemonic	Description	Code	Byte	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

布尔操作：

Mnemonic	Description	Code	Byte	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

数据传输：

Mnemonic	Description	Code	Byte	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 direct1,direct2	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

转下页

接上页

Mnemonic	Description		Code	Byte	Cycles
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,#data 16	Load 16-bit constant into active DPH and DPL inLARGE mode		90	3	3
MOV DPTR,#data 24	Load 24-bit constant into active DPX,DPH and DPL in FLAT mode		90	4	4
MOVC A,@A+DPTR	Move code byte relative to DPTR to accumulator		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 extern RAM(8-bit address)	CODE inside ROM/RAM destination XRAM data	F2-F3	1	4*
		all other cases			5*
MOVX @DPCR,A	Move A to extern RAM(16-bit address)	CODE inside ROM/RAM destination XRAM data	F0	1	3*
		all other cases			4*
PUSH direct	Push direct byte onto stack	LARGE	C0	2	3
		FLAT			3
POP direct	Pop direct byte from stack	LARGE	D0	2	2
		FLAT			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
XCHDA,@Ri	Exchange low-order nibble indirect RAM with A		D6-D7	1	3

程序分支：

Mnemonic	Description		Code	Byte	Cycles
ACALL addr11	Absolute subroutine call	LARGE	0x11-0xF1	2	4
ACALL addr19		FLAT		3	5
LCALL addr16	Long subroutine call	LARGE	3	3	4
LCALL addr24		FLAT		4	6
RET	Return from subroutine	LARGE	22	1	4
		FLAT			5
RETI	Return from interrupt	LARGE	32	1	4
		FLAT			5
AJMP addr11	Absolute jump	LARGE	01-E1	2	3
AJMP addr19		FLAT		3	4
LJMP addr16	Long jump	LARGE	02	3	4
LJMP addr24		FLAT		4	5
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 rel	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 equal		B8-BF	3	4
CJNE @Ri,#data rel	Compare immediate to ind. and jump if not equal		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

指令集简介 1:

Opcode	Mnemonic	Opcode	Mnemonic
00 H	NOP	30 H	JNB bit.rel
01 H	AJMP addr11/addr19	31 H	ACALL aaddr11/addr19
02 H	LJMP addr16/addr24	32 H	RETI
03 H	RR A	33 H	RLC A
04 H	INC A	34 H	ADDC A,#data
05 H	INC direct	35 H	ADDC A,direct
06 H	INC @R0	36 H	ADDC A,@R0
07 H	INC @R1	37 H	ADDC A,@R1
08 H	INC R0	38 H	ADDC A,R0
09 H	INC R1	39 H	ADDC A,R1
0A H	INC R2	3A H	ADDC A,R2
0B H	INC R3	3B H	ADDC A,R3
0C H	INC R4	3C H	ADDC A,R4
0D H	INC R5	3D H	ADDC A,R5
0E H	INC R6	3E H	ADDC A,R6
0F H	INC R7	3F H	ADDC A,R7
10 H	JBC bit,rel	40 H	JC rel
11 H	ACALL addr11/addr19	41 H	AJMP addr11/addr19
12 H	LCALL addr16/addr24	42 H	ORL direct,A
13 H	RRC A	43 H	ORL direct,#data
14 H	DEC A	44 H	ORL A,#data
15 H	DEC direct	45 H	ORL A,direct
16 H	DEC @R0	46 H	ORL A,@R0
17 H	DEC @R1	47 H	ORL A,@R1
18 H	DEC R0	48 H	ORL A,R0
19 H	DEC R1	49 H	ORL A,R1
1A H	DEC R2	4A H	ORL A,R2
1B H	DEC R3	4B H	ORL A,R3
1C H	DEC R4	4C H	ORL A,R4
1D H	DEC R5	4D H	ORL A,R5
1E H	DEC R6	4E H	ORL A,R6
1F H	DEC R7	4F H	ORL A,R7
20 H	JB bit.rel	50 H	JNC rel
21 H	AJMP addr11/addr19	51 H	ACALL aaddr11/addr19
22 H	RET	52 H	ANL direct,A
23 H	RLA	53 H	ANL direct,#data
24 H	ADD A,#data	54 H	ANL A,#data
25 H	ADD A,direct	55 H	ANL A,direct

转下页

接上页

Opcode	Mnemonic	Opcode	Mnemonic
26 H	ADD A,@R0	56 H	ANL A,@R0
27 H	ADD A,@R0	57 H	ANL A,@R1
28 H	ADD A,R0	58 H	ANL A,R0
29 H	ADD A,R1	59 H	ANL A,R1
2A H	ADD A,R2	5A H	ANL A,R2
2B H	ADD A,R3	5B H	ANL A,R3
2C H	ADD A,R4	5C H	ANL A,R4
2D H	ADD A,R5	5D H	ANL A,R5
2E H	ADD A,R6	5E H	ANL A,R6
2F H	ADD A,R7	5F H	ANL A,R7

指令集简介 2:

Opcode	Mnemonic	Opcode	Mnemonic
60 H	JZ rel	90 H	MOV DPTR,#data16 MOV DPTR,#data24
61 H	AJMP addr11	91 H	ACALL aaddr11
62 H	XRL direct,A	92 H	MOV bit,C
63 H	XRL direct,#data	93 H	MOVC A,@A+DPTR
64 H	XRL A.#data	94 H	SUBB A,#data
65 H	XRL A.direct	95 H	SUBB A,direct
66 H	XRL A,@R0	96 H	SUBB A,@R0
67 H	XRL A,@R1	97 H	SUBB A,@R1
68 H	XRL A,R0	98 H	SUBB A,R0
69 H	XRL A,R1	99 H	SUBB A,R1
6A H	XRL A,R2	9A H	SUBB A,R2
6B H	XRL A,R3	9B H	SUBB A,R3
6C H	XRL A,R4	9C H	SUBB A,R4
6D H	XRL A,R5	9D H	SUBB A,R5
6E H	XRL A,R6	9E H	SUBB A,R6
6F H	XRL A,R7	9F H	SUBB A,R7
70 H	JNZ,rel	A0 H	ORL C,bit
71 H	ACALL addr11/addr19	A1 H	AJMP addr11/addr19
72 H	ORL C,direct	A2 H	MOV C,bit
73 H	JMP @A+DPTR	A3 H	INC DPTR
74 H	MOV A,#data	A4 H	MUL AB
75 H	MOV direct,#data	A5 H	-
76 H	MOV @R0,#data	A6 H	MOV @R0,direct
77 H	MOV @R1,#data	A7 H	MOV @R1,direct

转下页

接上页

Opcode	Mnemonic	Opcode	Mnemonic
78 H	MOV R0.#data	A8 H	MOV R0,direct
79 H	MOV R1.#data	A9 H	MOV R1,direct
7A H	MOV R2.#data	AA H	MOV R2,direct
7B H	MOV R3.#data	AB H	MOV R3,direct
7C H	MOV R4.#data	AC H	MOV R4,direct
7D H	MOV R5.#data	AD H	MOV R5,direct
7E H	MOV R6.#data	AE H	MOV R6,direct
7F H	MOV R7.#data	AF H	MOV R7,direct
80 H	SJMP rel	B0 H	ANL C,bit
81 H	AJMP addr11/addr19	B1 H	ACALL aaddr11/addr19
82 H	ANL C,bit	B2 H	CPL bit
83 H	MOVC A,@A+PC	B3 H	CPL C
84 H	DIV AB	B4 H	CJNE A,#data,rel
85 H	MOV direct,direct	B5 H	CJNE A,direct,rel
86 H	MOV direct,@R0	B6 H	CJNE @R0,#data,rel
87 H	MOV direct,@R1	B7 H	CJNE @R1,#data,rel
88 H	MOV direct,R0	B8 H	CJNE R0,#data,rel
89 H	MOV direct,R1	B9 H	CJNE R1,#data,rel
8A H	MOV direct,R2	BA H	CJNE R2,#data,rel
8B H	MOV direct,R3	BB H	CJNE R3,#data,rel
8C H	MOV direct,R4	BC H	CJNE R4,#data,rel
8D H	MOV direct,R5	BD H	CJNE R5,#data,rel
8E H	MOV direct,R6	BE H	CJNE R6,#data,rel
8F H	MOV direct,R7	BF H	CJNE R7,#data,rel

指令集简介 3:

Opcode	Mnemonic	Opcode	Mnemonic
C0 H	PUSH direct	E0 H	MOVX A,@DPTR
C1 H	AJMP addr11/addr19	E1 H	AJMP addr11/addr19
C2 H	CLR bit	E2 H	MOVX A,@R0
C3 H	CLR C	E3 H	MOVX A,@R1
C4 H	SWAP A	E4 H	CLR A
C5 H	XCH A,direct	E5 H	MOV A,direct
C6 H	XCH A,@R0	E6 H	MOV A,@R0
C7 H	XCH A,@R1	E7 H	MOV A,@R1
C8 H	XCH A,R0	E8 H	MOV A,R0
C9 H	XCH A,R1	E9 H	MOV A,R1

转下页

接上页

Opcode	Mnemonic	Opcode	Mnemonic
CA H	XCH A,R2	EA H	MOV A,R2
CB H	XCH A,R3	EB H	MOV A,R3
CC H	XCH A,R4	EC H	MOV A,R4
CD H	XCH A,R5	ED H	MOV A,R5
CE H	XCH A,R6	EE H	MOV A,R6
CF H	XCH A,R7	EF H	MOV A,R7
D0 H	POP direct	F0 H	MOVX @DPTR,A
D1 H	ACALL addr11/addr19	F1 H	ACALL addr11/addr19
D2 H	SETB bit	F2 H	MOVX @R0,A
D3 H	SETB C	F3 H	MOVX @R1,A
D4 H	DA A	F4 H	CPL A
D5 H	DJNZ direct,rel	F5 H	MOV direct,A
D6 H	XCHD A,@R0	F6 H	MOV @R0,A
D7 H	XCHD A,@R1	F7 H	MOV @R1,A
D8 H	DJNZ R0,rel	F8 H	MOV R0,A
D9 H	DJNZ R1,rel	F9 H	MOV R1,A
DA H	DJNZ R2,rel	FA H	MOV R2,A
DB H	DJNZ R3,rel	FB H	MOV R3,A
DC H	DJNZ R4,rel	FC H	MOV R4,A
DD H	DJNZ R5,rel	FD H	MOV R5,A
DE H	DJNZ R6,rel	FE H	MOV R6,A
DF H	DJNZ R7,rel	FF H	MOV R7,A

## 4.2、FLASH

FLASH 的存储量为 8K×8bit。FLASH 放置用户程序及用户配置选项位。FLASH 可支持烧写 1000 次。

## 4.3、EEPROM

内建 256 字节 EEPROM 存储器，通过 MCU 可对 EEPROM 的数据进行读写。对 EEPROM 写数据时不需要用户进行擦除操作，且具有自动写保护功能，防止误操作对 EEPROM 数据破坏。EEPROM 具有非易失的特性，断电后数据仍能安全保存。用户可使用 EEPROM 进行校正值、系统配置以及用户自定义参数等数据的保存。

由于 EEPROM 读写时钟与系统时钟异步，每完成一个字节的读写后，需通过中断方式或查询方式等待异步处理时间。由于 EEPROM 写入特性的限制，每两次 EEPROM 的写操作之间用户需等待至少 5ms 以上。



## 寄存器描述

## 1) E2\_CTRL 寄存器

地址：0xB9

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 6]			保留
[5]	Dram_to_flash_ie	R/W	1: 写入 EEPROM 0: 无操作
[4]	Dram_to_flash_oe	R/W	1: 读取 EEPROM 0: 无操作
[3]	Dram_to_flash_cer	R/W	FLASH 片擦除信号 0 : FLASH 擦除方式由 Dram_to_flash_en_byteer 决定 1: FLASH 擦除方式为片擦除
[2]	Dram_to_flash_pg_mode	R/W	FLASH 写入模式选择, 高速写入 模式速度是低俗写入模式的两 倍 0: 低速写入模式 1: 高速写入模式 (适用于 FLASH $V_{DDH} \geq 5V$ )
[1]	Dram_to_flash_en_byteer	R/W	扇区擦除/字节擦除选择, 只有当 Dram_to_flash_cer 为 0 的时候 此位有效: 0: 扇区擦除操作 1: 字节擦除操作
[0]	Dram_to_flash_lckn	R/W	FLASH/EEPROM 锁存信号, 在锁 存状态下 FLASH 最后一页 (地址 0x1f80-0x1fff) 和 EEPROM 最后一页 (地址 0xfc-0xff) 无法被擦除 0: FLASH 锁存, 无法执行擦除操 作 1: 不锁存

注：EEPROM 每 4 个作为一个扇区，即地址 0x00-0x03、0x04-0x07、0x08-0x0b……  
0xfc-0xff 分别作为一个扇区。当选择扇区擦除时，硬件将自动对当前操作地址所  
在扇区进行扇区擦除。因此，存在一种特殊情况，如设置起始地址为 0x02，操  
作方式设置为 4 字节写入操作时，所有操作地址（0x02、0x03、0x04、0x05）跨越  
两个扇区，因此两个扇区内所有地址（0x00-0x07）都会被擦除。

## 2) E2\_PROTECT 寄存器

地址：0xBA

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	FLASH_ctrl_protect_reg	R/W	当 MCU 将 0xa5 写入该寄存器后，对寄存器 DRAM_CTRL_REG 写入的值才有效

## 3) E2\_START\_ADDR 寄存器

地址：0xBB

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	e2_start_addr	R/W	读写 E2PROM 时的起始地址配置 (addr: 0~255)

## 4) E2\_MODE 寄存器

地址：0xBC

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 1]			保留
[0]	e2_mode	R/W	E2 数据访问模式 0: 单次进行 1 个字节的读或写操作 1: 单次进行连续 4 个字节的读或写操作

## 5) E2\_TERHEE\_H 寄存器

地址：0xBD

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 1]			保留
[0]	terheeh	R/W	EEPROM 擦除时间长度设置高位

## 6) E2\_TERHEE\_M 寄存器

地址：0xBE

复位值：0xcb

访问：R/W

Bit	Name	R/W	Description
[7: 0]	terheem	R/W	EEPROM 擦除时间长度设置中 8 位

## 7) E2\_TERHEE\_L 寄存器

地址：0xBF

复位值：0x20

访问：R/W

Bit	Name	R/W	Description
[7: 0]	terheel	R/W	EEPROM 擦除时间长度设置低 8 位

注：EEPROM 擦除时间计算方式 TERHEE 设定值乘以 FLASH 时钟周期，即  $TERHEE/f_{FLASH}$ ，EEPROM 擦除时间要求大于 5ms。

## 8) E2\_TPGHF\_H 寄存器

地址：0xC0

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 1]			
[0]	tpghfh	R/W	FLASH/EEPROM 一半写入时间长度设置高位

## 9) E2\_TPGHF\_L 寄存器

地址：0xC1

复位值：0xd0

访问：R/W

Bit	Name	R/W	Description
[7: 0]	tpghfl	R/W	FLASH/EEPROM 一半写入时间长度设置低位

注：FLASH/EEPROM 一半写入时间计算方式 TPGHF 设定值乘以 FLASH 时钟周期，即  $TPGHF/f_{FLASH}$ ，FLASH/EEPROM 一半写入时间要求大于 20μs。

## 10) E2\_DATA\_REG0 寄存器

地址：0xEC

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	e2_data_reg0	R/W	操作 E2 时，该地址映射 E2 扇区的第一个地址

## 11) E2\_DATA\_REG1 寄存器

地址：0xED

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	e2_data_reg1	R/W	操作 E2 时，该地址映射 E2 扇区的第二个地址

## 12) E2\_DATA\_REG2 寄存器

地址：0xEE

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	e2_data_reg2	R/W	操作 E2 时，该地址映射 E2 扇区的第三个地址

## 13) E2\_DATA\_REG3 寄存器

地址：0xEF

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	e2_data_reg3	R/W	操作 E2 时，该地址映射 E2 扇区的第四个地址

注：当选择单字节操作时，只有 E2\_DATA\_REG0 寄存器被使用。当选择 4 字节操作时，E2\_DATA\_REG0、E2\_DATA\_REG1、E2\_DATA\_REG2、E2\_DATA\_REG3 寄存器都会被使用。

## EEPROM 操作流程：

### 1) BYTE 写 E2 操作

- ① 配置读写 256 字节 EEPROM 的起始地址 E2\_START\_ADDR 寄存器，在 byte 读写操作模式下，起始地址即为传输地址；
- ② 配置 E2\_MODE 寄存器为 0，选择 E2 byte 读写操作模式；
- ③ 配置 E2\_DATA\_REG：如果操作地址为扇区第一个地址（E2\_START\_ADDR[1:0]=2'b00），配置 E2\_DATA\_REG0；如果操作地址为扇区第二个地址（E2\_START\_ADDR[1:0]=2'b01），配置 E2\_DATA\_REG1；如果操作地址为扇区第三个地址（E2\_START\_ADDR[1:0]=2'b10），配置 E2\_DATA\_REG2；如果操作地址为扇区第四个地址（E2\_START\_ADDR[1:0]=2'b11），配置 E2\_DATA\_REG3；
- ④ 配置 FLASH\_PROTECT\_REG 寄存器为 0xa5；
- ⑤ 配置 E2\_CTRL 寄存器，FLASH\_IE 配置为 1，FLASH\_OE 配置为 0，其他位按需配置；
- ⑥ MCU 配置 PCON bit1 位发 stop 命令，stop 之后需增加 4 个 nop 指令，进入空闲 1 模式；
- ⑦ 硬件电路将让 MCU 休眠（暂停工作），硬件电路根据 MCU 配置的寄存器值自动完成数据搬运；
- ⑧ 数据搬运结束后，硬件电路发 run 信号，自动唤醒 MCU；
- ⑨ MCU 配置 E2\_CTRL 寄存器恢复成默认状态，MCU 配置 FLASH\_PROTECT\_REG 清 0。

### 2) SECTOR 写 E2 操作

- ① 配置读写 256 字节 EEPROM 的起始地址 E2\_START\_ADDR 寄存器，在 sector 读写操作模式下，起始地址设置为需要操作扇区的第一个地址；
- ② 配置 E2\_MODE 寄存器为 1，选择 E2 sector 读写操作模式；
- ③ 配置 E2\_DATA\_REG0、E2\_DATA\_REG1、E2\_DATA\_REG2、E2\_DATA\_REG3，四个寄存器分别对应需要操作的 E2 扇区的四个地址；
- ④ 配置 FLASH\_PROTECT\_REG 寄存器为 0xa5；
- ⑤ 配置 E2\_CTRL 寄存器，FLASH\_OE 配置为 1，FLASH\_IE 配置为 0，其他位按需配置；
- ⑥ MCU 配置 PCON bit1 位发 stop 命令，stop 之后需增加 4 个 nop 指令，进入空闲 1 模式；
- ⑦ 硬件电路将让 MCU 休眠（暂停工作），硬件电路根据 MCU 配置的寄存器值自动完成数据搬运；
- ⑧ 数据搬运结束后，硬件电路发 run 信号，自动唤醒 MCU；
- ⑨ MCU 配置 E2\_CTRL 寄存器恢复成默认状态，MCU 配置 FLASH\_PROTECT\_REG 清 0。

说明：在系统中 flash（eeprom）和 dram 数据交换的优先级最高，数据交换过程中，有外部中断产生将延迟唤醒 mcu（直到数据交换结束）。

#### 4.4、时钟振荡器

电路提供多种时钟振荡器可供客户灵活选择。振荡器除了作为系统时钟源，还作为看门狗定时器和时基功能的时钟源。外部振荡器需要一些外围器件，而集成的两个内部振荡器不需要任何外围器件。它们提供的高速和低速系统振荡器具有较宽的频率范围。所有振荡器选择通过配置选项选择。较高频率的振荡器提供更高的性能，但要求有更高的功耗，反之亦然。动态切换快慢系统时钟的能力使单片机具有灵活、优化的性能和功耗比。此特性对功耗敏感的应用领域尤为重要。

时钟振荡器类型：

类 型	名 称	频 率	引 脚
内部高速 RC	HIRC	1MHZ/16MHz	
内部低速 RC	LIRC	8kHz	

系统时钟配置：

单片机有两个时钟源，两个内部 RC。高速振荡器有内部 1M/16MHz RC 振荡器。低速振荡器有内部 8kHz 振荡器。高速或低速振荡器作为系统时钟的选择是通过配置选项进行选择。

#### 4.5、系统时钟和工作模式

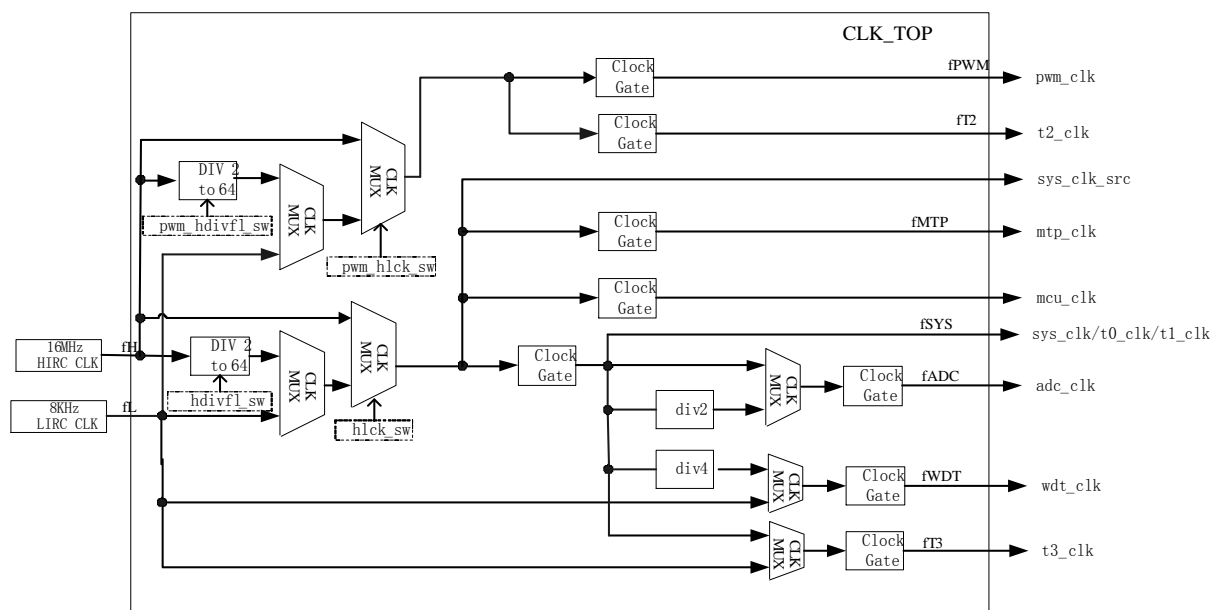
##### 1) 系统时钟

用户使用配置选项和寄存器编程可获取多种时钟，进而使系统时钟获取最大的应用性能。主系统时钟可来自高频时钟源  $f_H$  或低频时钟源  $f_L$ ，通过 FLASH 信息区配置。高频时钟来自内部 HIRC 振荡器，低频系统时钟源来自内部 LIRC 振荡器。快速唤醒发生后，LIRC 为单片机提供时钟，使系统能够在更短的时间内快速唤醒。LIRC 和  $f_{SYS}/4$  用于看门狗定时器的时钟源。

##### 2) 系统工作模式

单片机有 5 种不同的工作模式，每种有它自身的特性，根据应用中不同的性能和功耗要求可选择不同的工作模式。单片机正常工作有两种模式：正常模式和低速模式。剩余的 3 种工作模式：休眠模式 0、空闲模式 0 和空闲模式 1 用于单片机 CPU 关闭时以节省耗电。

时钟网络图：



系统时钟选项：

工作模式	LIRC	HIRC	CPU	$f_{SYS}$	$f_{WDT}$	$f_{ADC}$	$f_{T2}$	$f_{T3}$	$f_{PWM}$	$f_{FLASH}$
正常模式	On	On	On	$f_H \sim f_H/64$	On	On	On	On	On	On
低速模式	On	Off	On	$f_L$	On	On	On	On	On	On
空闲模式 0	On	Off	Off	Off	On/Off	Off	On/Off	On/Off	Off	Off
空闲模式 1	On	On	Off	On	On	On	On	On	On	On
休眠模式 0	Off	Off	Off	Off	Off	Off	Off	Off	Off	Off

### 1) 正常模式

单片机的所有功能均可在此模式中实现且系统时钟由一个高速振荡器提供。该模式下单片机正常工作的时钟源来自 **HIRC** 振荡器。高速振荡器频率可被分为 2~64 的不等比率，实际的比率由 **FLASH** 中的系统时钟控制位选择的。单片机使用高速振荡器分频作为系统时钟可减少工作电流。

### 2) 低速模式

此模式的系统时钟虽为较低速时钟源，但单片机仍能正常工作。该低速时钟源可来自 **LIRC** 振荡器。单片机在此模式中运行所耗工作电流较低。在低速模式下，**fH** 关闭。

### 3) 休眠模式 0

在 **MCU** 发出 **STOP** 指令执行后且 **sys\_reg2** 寄存器中 **IDLEN** 位为低时，系统进入休眠模式。在休眠模式 0 中，**CPU**、**fWDT** 停止运行，看门狗定时器功能关闭，所有振荡器关闭。

### 4) 空闲模式 0

执行 **STOP** 指令后且 **sys\_reg2** 寄存器中 **IDLEN** 位为高，**WDTC** 寄存器中 **FSYSON** 位为低时，系统进入空闲模式 0。在空闲模式 0 中，**CPU** 停止，但一些外围功能如看门狗定时器将继续工作。在空闲模式 0 中，系统振荡器停止，**fWDT**、**fT2**、**fT3** 时钟的开启或关闭由所选时钟源决定。若时钟源为 **fSYS/4**，**fWDT**、**fT2**、**fT3** 关闭；若时钟源为 **LIRC**，**fWDT**、**fT2**、**fT3** 开启。

### 5) 空闲模式 1

执行 **STOP** 指令后且 **sys\_reg2** 寄存器中 **IDLEN** 位为高，**WDTC** 寄存器中 **FSYSON** 位为高时，系统进入空闲模式 1。在空闲模式 1 中，**CPU** 停止，但会提供一个时钟源给一些外围功能如看门狗定时器。在空闲模式 1 中，系统振荡器继续运行，该系统振荡器可以为高速或低速系统振荡器。在该模式中看门狗定时器时钟 **fWDT** 开启。若时钟源为 **fSYS/4** 或 **LIRC**，**fWDT** 开启。



寄存器描述：

1) SYS\_REG1 寄存器

地址：0xE1

复位值：0x08

访问：R/W

Bit	Name	R/W	Description
[7]	test_mode	R	0: 正常工作状况 1: 测试模式（只读）
[6]			保留
[5]	wdt_rst_en	R/W	看门狗复位使能信号，“0”关闭，“1”打开
[4]	wdt_rst_flag	R/W	看门狗复位标志，高有效 写“1”清标志
[3]	hsosc_en_reg	R/W	高速振荡器开关寄存器 0: 关闭高速振荡器时钟 1: 打开高速振荡器时钟
[2: 0]			保留

2) SYS\_REG2 寄存器

地址：0xE2

复位值：0xe0

访问：R/W

Bit	Name	R/W	Description
[7: 5]	hdivfl_sw	R/W	Hlck_sw 为 0 时系统时钟选择位（需先将 Flash_sysck_ctl 置为 1 时设置才有效，为 0 时从 FLASH 读取） 000: fL 001: fL 010: 64 分频 011: 32 分频 100: 16 分频 101: 8 分频 110: 4 分频 111: 2 分频
[4]	div_adc	R/W	ADC 时钟分频设置 0: 1 分频 1: 2 分频

转下页

接上页

Bit	Name	R/W	Description
[3]	lto	R	低速振荡器就绪标志位 0: 未就绪 1: 就绪
[2]	hto	R	高速振荡器就绪标志位 0: 未就绪 1: 就绪
[1]	idlen	R/W	空闲模式控制 0: 除能 1: 使能
[0]	hlck_sw	R/W	系统时钟源选择 (Flash_sysck_ctl 为 1 时 MCU 方可写入, 为 0 时从 FLASH 读取) 0: 时钟源自 fH 分频或 fL 1: 时钟源自 fH

## 3) SYS\_REG3 寄存器

地址: 0xE3

复位值: 0x0f

访问: R/W

Bit	Name	R/W	Description
[7: 6]			保留
[5]	flash_sysck_ctl	R/W	初始为 0 时, 即系统时钟的分频切换由 FLASH 的值控制 为 1 时, 系统时钟的分频切换由 MCU 写入
[4]	stopen_n	R/W	对 MCU 进行 STOP 前, 读取该寄存器为 1 等待, 为 0 可以写入 STOP
[3: 1]	pwm_hdivfl_sw	R/W	Pwm_hlck_sw 为 0 时 PWM 时钟选择位 000: fL 001: fL 010: 64 分频 011: 32 分频 100: 16 分频 101: 8 分频 110: 4 分频 111: 2 分频
[0]	pwm_hlck_sw	R/W	PWM 时钟源选择 0: 时钟源自 fH 分频或 fL 1: 时钟源自 fH

## 4) SYS\_REG\_MASK 寄存器

地址：0xE4

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	sys_reg_mask	R/W	系统寄存器屏蔽寄存器 只有对此寄存器写入“8'hA5”方可操作 sys_reg 寄存器，否则无法改变 sys_reg1、sys_reg2、sys_reg3 寄存器

## 4.6、唤醒

系统进入休眠或空闲模式之后，可以通过 3 种方式唤醒：PA 口、系统中断、WDT 溢出。

PA 口中的每个引脚都可以唤醒功能。按键唤醒后，打开振荡器，计数稳定后打开 MCU 时钟，程序将在“STOP”指令后继续执行。

系统中断唤醒后，打开振荡器，计数稳定后打开 MCU 时钟，程序将在“STOP”指令后继续执行。

由 WDT 溢出唤醒。

## 4.7、特殊功能寄存器

特殊功能寄存器为 128Bytes，CPU 本身需要占用一部分，其余部分为外设 SFR (Timer0/1、UART、Timer2/3、WatchDog、PAPORT、PBPORT、PCPORT)。

特殊功能寄存器分配表：

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
0xF8	IRQSTA TL	IRQS TATH	INTML	INTMH	IRQMS KDL	IRQMS KDH	INTSE LL	INTSE LH	0xF8
0xF0	<b>B</b>	ADR L	ADRH	ADCR	ADCV	ADIFG	LDOC	BAND GAP_T EMP	0xF0
0xE8		<b>STAT</b> US	<b>MXAX</b>	<b>TA</b>	E2_DA TA_RE G0	E2_DA TA_RE G1	E2_DA TA_RE G2	E2_DA TA_RE G3	0xE8
0xE0	<b>ACC</b>	SYS_ REG1	SYS_R EG2	SYS_R EG3	SYS_R EG_M ASK	CMP2	CMPSE L	CMPIF G	0xE0
0xD8	<i>WDCON</i>	WDC NTM	WDCN TH	XDAT A_CTR L	PWM_ CTRL1	PWM_ CTRL2	PWM_ CTRL3	CMP1	0xD8

转下页

接上页

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
0xD0	<b>PSW</b>	PWM_PER IOD2 H	PWM_PULSE 2L	PWM_PULSE 2H	PWM_PERIO D3L	PWM_PERIO D3H	PWM_PULSE 3L	PWM_PULSE 3H	0xD0
0xC8	PWM_PERIOD0 H	PWM_PULSE0 L	PWM_PULSE 0H	PWM_PERIO D1L	PWM_PERIO D1H	PWM_PULSE 1L	PWM_PULSE 1H	PWM_PERIO D2L	0xC8
0xC0	E2_TPG HF_H	E2_TPG HF_L	PWM_PERIO D0L	PAPU	PBPU	PCPU	MISC	ADPCR	0xC0
0xB8	<b>IP</b>	E2_CTRL	E2_PROTECT	E2_START_ADDR	E2_MODE	E2_TERRH	E2_TERRM	E2_TERRL	0xB8
0xB0		PCREG	PCDIR	PCIE	PCIES_L	TMR3C	PCIFG	TMR3	0xB0
0xA8	<b>IE</b>	PBREG	PBDIR	PBIE	PBIES_L	PBIES_H	PBIFG		0xA8
0xA0	<b>P2</b>	PAREG	PADIR	PAIE	PAIES_L	PAIES_H	PAIFG		0xA0
0x98	SCON	SBUF	SMOD	TMR2C	<b>AP</b>	<b>ACON</b>	TMR2L	TMR2H	0x98
0x90	MCU_HI RC_TRIM	MCU_LIRC_TRIM	<b>WTST</b>	<b>DPX0</b>	RC_CTRL1	MCU_RC_PTCT	MCU_BAND GAP_TRIM		0x90
0x88	<b>TCON</b>	TMOD	TL0	TL1	TH0	TH1	<b>CKCON</b>	TTMOD2	0x88
0x80		<b>SP</b>	<b>DPL0</b>	<b>DPL1</b>	WAKEUP_EN	WAKEUP_EDGE	TTCON	<b>PCON</b>	0x80
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

其中黑体为 MCU 已经占用的 SFR 寄存器，其他为用户自定义的特殊功能寄存器。

XDATA\_CTRL 是用来控制 XDATA 总线的读写信号，当 XDATA\_CTRL[0]为 1 时，XDATA 的 memory 和 reg 禁止读写，为 0 时，允许读写。任何时候都能对 WatchDog 相关寄存器写 1，将 XDATA\_CTRL 的 bit7 写 1 后，才能对 WDCON 寄存器的 bit2、bit3、bit5 写 0。

PCON 的 bit1 是 STOP 位，MCU 写 stop 信号为 1 将进入休眠模式，在进入休眠模式之前需将中断控制器及其他接口模块的 mask 信号清 0，以保证 MCU 能被唤醒，在 MCU 唤醒之后，STOP 自动清 0。

注意：在 STOP 指令之后需要至少 4 条 NOP 指令，以保证程序唤醒后正常运行。

#### 4.7.1、CPU 占用的 SFR

##### 1) 寄存器 (ACC)

SFR 地址：0xE0

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	ACC.7~ACC.0	R/W	累加器

##### 2) 寄存器 (B)

SFR 地址：0xF0

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	B.7~B.0	R/W	寄存器，在乘除法指令时用到

##### 3) 寄存器 (PSW)

SFR 地址：0xD0

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7]	CY	R/W	进位标志
[6]	AC	R/W	辅助进位
[5]	F0	R/W	通用标志 0
[4: 3]	RS[1: 0]	R/W	寄存器 R0~R7 bank 选择位 00: Bank0, 地址: 0x00~0x07 01: Bank1, 地址: 0x08~0x0F 10: Bank2, 地址: 0x10~0x17 11: Bank3, 地址: 0x18~0x1F
[2]	OV	R/W	溢出标志
[1]	F1	R/W	通用标志 1
[0]	P	R/W	奇偶标志

#### 4) FLAT/LARGE 模式切换寄存器 (ACON)

SFR 地址: 0x9D

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 2]			保留
[1]	AM	R/W	地址模式选择 0: 16 位程序地址模式--LARGE 1: 23 位扩展程序地址模式--FLAT 本项目存储空间仅适用于 LARGE 模式
[0]			保留

#### 5) 程序写使能寄存器 (PCON)

SFR 地址: 0x87

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 5]			保留
[4]	PWE	R/W	xdata 写操作控制 0: 允许 xdata 写操作 1: 禁止 xdata 写操作
[3]			保留
[2]	SWB		保留, 不能使用
[1]	STOP	R/W	1: MCU 进入休眠状态 MCU 外部电路收到外围设备的指令或内部低频模块的处理完成中断后, 将会唤醒 MCU
[0]	PMM		保留, 不能使用

#### 6) 程序等待周期控制寄存器 (WTST)

SFR 地址: 0x92

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 3]			保留
[2: 0]	WTST	R/W	在本电路中, WTST 保留, 不需要使用

## 7) 寄存器 (DPX0)

SFR 地址: 0x93

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 0]	DPX.7~DPX.0	R/W	外部数据指针低 8 位

## 8) 寄存器 (MXAX)

SFR 地址: 0xEA

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 0]	MXAX.7~MXAX.0	R/W	当在 MOVX 指令中用到 R0 或 R1 寄存器时, 通常把 MXAX 等价于 XRAMADDR[23: 16]

## 9) 寄存器 (P2)

SFR 地址: 0xA0

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 0]	P2.7~P2.0	R/W	被用来产生外部数据存储器地址, 此时 P2 中数据作为外部数据存储器地址的高 8 位 XRAMADDR[15: 8] MOVX A, @Ri MOVX @Ri, A

## 10) 寄存器 (DPL0)

SFR 地址: 0x82

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 0]	DPL0	R/W	数据指针寄存器低 8 位

## 11) 寄存器 (DPL1)

SFR 地址: 0x83

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 0]	DPL1	R/W	数据指针寄存器高 8 位

## 12) 时钟控制寄存器 (CKCON)

SFR 地址: 0x8E

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 3]			保留
[2: 0]	MD[2: 0]	R/W	在本项目中, CKCON 保留, 不需要使用

## 13) 寄存器 (SP)

SFR 地址: 0x81

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 0]	SP.7~SP.0	R/W	堆栈指针寄存器

## 14) 中断使能寄存器 (IE)

SFR 地址: 0xA8

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7]	EA	R/W	全局中断使能位
[6: 3]			保留
[2]	EX1	R/W	Int1 中断使能位
[1]			保留
[0]	EX0	R/W	Int0 中断使能位

本电路中, CPU 共两个中断向量, 如下表所示:

Interrupt Flag	Function	Active Level/Edge	Flag Reset	Vector	Natural Priority
IE0	Device Pin INT0	Low/Falling	Hardware	0x03	1
IE1	Device Pin INT1	Low/Falling	Hardware	0x13	3



#### 15) 中断优先级控制寄存器 (IP)

SFR 地址: 0xB8

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 3]			保留
[2]	PX1	R/W	INT1 优先级控制 1: 高优先级 0: 低优先级
[1]			保留
[0]	PX0	R/W	INT0 优先级控制 1: 高优先级 0: 低优先级

#### 16) 中断配置寄存器 (TCON)

SFR 地址: 0x88

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 4]			保留
[3]	IE1	R/W	Int1 中断标志位: 当 MCU 进入中断处理程序后, 硬件自动清 0
[2]	IT1	R/W	Int1 电平和沿触发选择位 0: 电平触发 1: 沿触发
[1]	IE0	R/W	Int0 中断标志位 当 MCU 进入中断处理程序后, 硬件自动清 0
[0]	IT0	R/W	Int0 电平和沿触发选择位 0: 电平触发 1: 沿触发

说明: 当 INT0 (INT1) 中有多个中断时, 需要设置相应的触发选择位为“电平触发”。

## 17) 中断状态寄存器 (STATUS)

SFR 地址: 0xE9

复位值: 0x00

访问: R

Bit	Name	R/W	Description
[7]			保留
[6]	HIP	R	高优先级中断状态
[5]	LIP	R	低优先级中断状态
[4: 0]			保留

## 4.8、定时器 0/定时器 1

MCU 包含两个 16 位通用定时器 TIMER0 和 TIMER1，这两个定时器既可以作为普通定时器使用，也可以作为 UART 的波特率产生。

作为普通定时器时，定时器模块按照软件配置的定时周期进行计数，和选择的 sys\_clk 频率的不同分频（1 倍，1/2 倍，1/4 倍或 1/8 倍）进行计数，输出定时中断信号；作为波特率产生时，可以选择采用两个 8bits 的定时器分别用两个不同的计数时钟进行计数，共同产生一个计时溢出周期信号。其中低 8 位采用普通定时器的计数 clk（sys\_clk 频率的 1 分频，2 分频，4 分频，8 分频）进行计数，高 8 位可选择采用 sys\_clk 频率的 12 分频或 16 分频的计数 clk 进行计数，波特率周期为高低 8 位的计数时间之和。

## 寄存器描述

## 1) TICON 寄存器

SFR 地址: 0x86

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7]	TF1	R/W	Timer1 中断标志位，高有效
[6]	TF0	R/W	Timer0 中断标志位，高有效
[5]	TR1	R/W	Timer1 计数使能位，高有效
[4]	TR0	R/W	Timer0 计数使能位，高有效
[3]	TE1	R/W	Timer1 使能位，高有效
[2]	TE0	R/W	Timer0 使能位，高有效
[1]	LOAD1	R/W	Timer1 定时周期配置有效位，软件写 1 有效，硬件自动清零
[0]	LOAD0	R/W	Timer0 定时周期配置有效位，软件写 1 有效，硬件自动清零

软件在更新定时周期寄存器的值后，需要配置 TTCON 的 LOAD0 或 LOAD1 为 1 后，配置的定时周期才有效。

TE0 和 TE1 为 Timer0 和 Timer1 的使能位，为低时 Timer0 和 Timer1 处于复位状态。TR0 和 TR1 为 Timer0 和 Timer1 的计数使能位，为低时 Timer0 和 Timer1 处于计数暂停状态（保持当前计数值），在波特率产生方式，不支持暂停控制。

TF0 和 TF1 为 Timer0 和 Timer1 的中断标志位，软件检测到 Timer 中断后应立即清除中断标志，即往 TF0 或 TF1 写 0。即使 Timer0 和 Timer1 的中断使能信号为低，当 Timer0 和 Timer1 定时溢出时 TF0 和 TF1 位也会变成 1。

## 2) TMOD 寄存器

SFR 地址：0x89

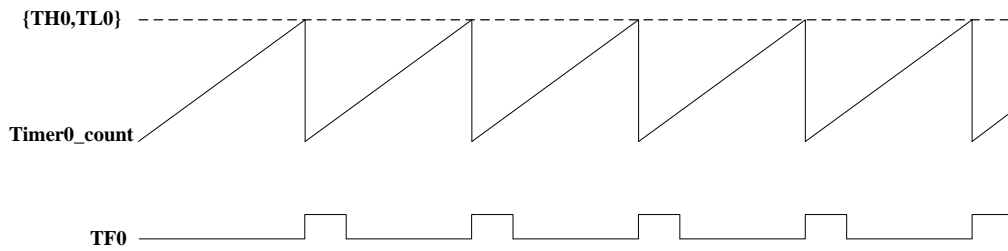
复位值：0x00

访问：R/W

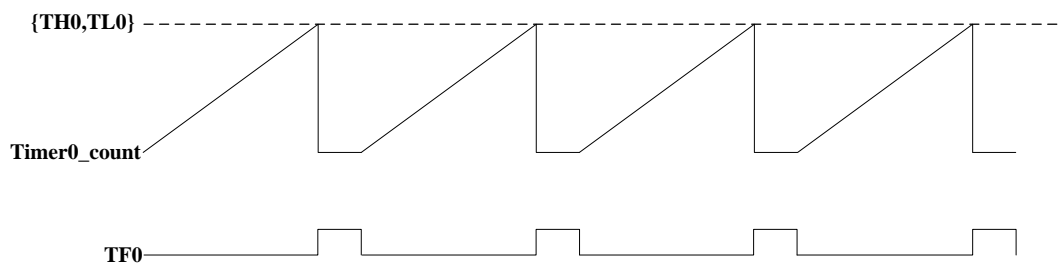
Bit	Name	R/W	Description
[7]	GATE1_EN	R/W	Timer1 门控使能位，高有效
[6]	T1_IE	R/W	Timer1 中断使能位，高有效
[5: 4]	DIV1	R/W	Timer1 预分频系数 00: 1 分频 01: 2 分频 10: 4 分频 11: 8 分频
[3]	GATE0_EN	R/W	Timer0 门控使能位，高有效
[2]	T0_IE	R/W	Timer0 中断使能位，高有效
[1: 0]	DIV0	R/W	Timer0 预分频系数 00: 1 分频 01: 2 分频 10: 4 分频 11: 8 分频

GATE0\_EN/GATE1\_EN 为 Timer0/Timer1 的门控使能信号。当门控使能为高时，如果 Timer0/Timer1 计数溢出，Timer0/ Timer1 将处于复位状态（计数值保持为零并停止计数）直到标志位 TF0/TF1 被清零后继续开始计数（如果 TE0/TE1 和 TR0/TR1 保持有效的的话）；当门控使能为低时，如果 Timer0/Timer1 计数溢出，Timer0/Timer1 将继续进行下一轮计数（如果 TE0/TE1 和 TR0/TR1 保持有效的的话），不管标志位 TF0/TF1 是否被清零，详见门控使能无效时 Timer0/ Timer1 定时器计数示意图和门控使能有效时 Timer0/Timer1 定时器计数示意图。若用作产生串口的波特率时，门控使能信号应配置成 0。

门控使能无效时 Timer0/Timer1 定时器计数示意图：



门控使能有效时 Timer0/Timer1 定时器计数示意图：



### 3) Timer0 计数周期寄存器 (TH0、TL0)

SFR 地址：0x8C

复位值：0x10

访问：R/W

Bit	Name	R/W	Description
[15: 8]	TH0	R/W	Timer0 计数周期高 8 位

SFR 地址：0x8A

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	TL0	R/W	Timer0 计数周期低 8 位

### 4) Timer1 计数周期寄存器 (TH1、TL1)

SFR 地址：0x8D

复位值：0x10

访问：R/W

Bit	Name	R/W	Description
[15: 8]	TH1	R/W	Timer1 计数周期高 8 位

SFR 地址：0x8B

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	TL1	R/W	Timer1 计数周期低 8 位

#### 5) TTMOD2 寄存器

SFR 地址：0x8F

复位值：0x00

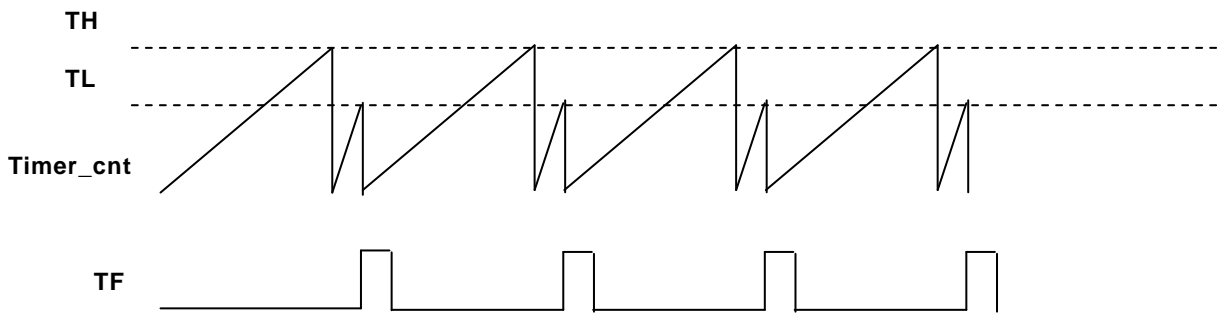
访问：R/W

Bit	Name	R/W	Description
[7]			保留
[6]			保留
[5]	RCLK	R/W	串口接收波特率选择位 0: 选择 Timer1 产生波特率 1: 选择 Timer0 产生波特率
[4]	TCLK	R/W	串口发送波特率选择位 0: 选择 Timer1 产生波特率 1: 选择 Timer0 产生波特率
[3]	H_CLK_SE1	R/W	BAUD_SEL1=1 时, Timer1 高八位计数时钟选择位 0: $f_{SYS}/12$ 1: $f_{SYS}/16$
[2]	BAUD_SEL1	R/W	当 RCLK 或 TCLK 为 0 时, 串口接收或发送选择 Timer1 产生波特率, 高低 8 位分别计数标志位 0: 16 位计数模式 1: 高低 8 位分别计数模式
[1]	H_CLK_SEL0	R/W	BAUD_SEL0=1 时, Timer0 高八位计数时钟选择位 0: $f_{SYS}/12$ 1: $f_{SYS}/16$
[0]	BAUD_SEL0	R/W	当 RCLK 或 TCLK 为 1 时, 串口接收或发送选择 Timer0 产生波特率, 高低 8 位分别计数标志位 0: 16 位计数模式 1: 高低 8 位分别计数模式

如果选择 16 位计数器产生串口的溢出时钟, 即 BAUD\_SEL 为 0, 那么计数器 16 位计数器以系统时钟 div 分频产生溢出时钟, 如“门控使能无效时 Timer0/Timer1 定时器计数示意图”所示, 此时 H\_CLK\_SEL 无作用。

如果选择 8 位计数器产生串口的溢出时钟, 即 BAUD\_SEL 为 1, 那么计数器高低八位分别计数, 以高 8 位为主, 低 8 位为辅, 高 8 位的计数时钟是系统时钟的 1/12 或 1/16(由 H\_CLK\_SEL 选择), 低 8 位的计数时钟是系统时钟的 div 分频。详见下图。

BAUD\_SEL 为 1 时 Timer0/ Timer1 定时器计数示意图：



Timer 工作流程：

- 1、配置 Timer0 和/或 Timer1 计数周期寄存器；
- 2、向 Timer 控制寄存器 TCON 中的 LOAD0 和/或 LOAD1 写 1，将 Timer 计数周期值真正配入硬件；
- 3、配置 Timer 模式寄存器 TMOD；
- 4、配置 Timer 控制寄存器 TCON 并使能 Timer0 和/或 Timer1；
- 5、Timer 正常工作时检测是否有中断或定时器是否溢出，如果检测到中断或定时器溢出，清除中断标志位。

计数时间计算公式：

$$16 \text{ 位计数模式：定时器溢出时间} = \frac{\text{预分频系数}}{f_{\text{sys}}} \times (\{TH0/1, TL0/1\} + 1)$$

$$\text{高低 8 位计数模式：定时器溢出时间} = \frac{\text{预分频系数}}{f_{\text{sys}}} \times (TL0/1 + 1) + (TH0/1 + 1) \times 12/16$$

## 4.9、UART

UART 是一个全双工的串行口，可以用于网络通信，也可以实现串行异步通信，还可以作为同步移位寄存器使用。软件对串口的操作通过 SFR 寄存器来完成。

### 寄存器描述

#### 1) SCON 寄存器

SFR 地址：0x98

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7]	SM0	R/W	串行口方式选择位，两位组合决定了四种工作模式
[6]	SM1	R/W	
[5]	SM2	R/W	多机通讯控制位，主要用于方式 2 和 3 1: 当第 9 (RB9) 位为 1 时，将数据送入 SBUF，置位 RI 并产生中断，否则丢弃 0: 不论第 9 位为 0 或 1，都将数据送入 SBUF，置位 RI 并产生中断
[4]	REN	R/W	允许接收位 1: 允许接收数据 0: 禁止接收数据
[3]	TB8	R/W	发送数据的第 9 位，在方式 2, 3 时，该位由用户通过软件设置，双机通信，作为奇偶校验位使用；多机通信，用作地址 (1) 和数据 (0) 标志。
[2]	RB8	R/W	接收数据的第 9 位，在方式 2、3 时，该位为存放接收到的第 9 位数据，可能是奇偶校验位，也可能是地址或数据的标志位。
[1]	TI	R/W	发送中断标志位，在方式 0 时，发送完 8 位后，硬件置位； 在其他方式下，发送停止位，硬件置位，由软件清零
[0]	RI	R/W	接收中断标志位，在方式 0 时，接收完 8 位后，硬件置位； 在其他方式下，接收停止位，硬件置位，由软件清零

SM0	SM1	工作方式	功 能	波 特 率
0	0	方式 0	扩展移位寄存器方式	$f_{osc}/12$
0	1	方式 1	8 位 UART	定时器溢出率/n (timer0: n=16; timer1: n=32)
1	0	方式 2	9 位 UART	$f_{osc}/32$ 或 $f_{osc}/64$
1	1	方式 3	9 位 UART	定时器溢出率/n (timer0: n=16; timer1: n=32)

## 2) SBUF 寄存器

SFR 地址: 0x99

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 0]	SBUF_REG	R/W	发送和接收 8 位数据缓存: 串行发送时, MCU 向 SBUF 写入数据 串行接收时, MCU 从 SBUF 读出数据

## 3) SMOD 寄存器

SFR 地址: 0x9A

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 1]			保留
[0]	SMOD	R/W	SMOD 是串行口波特率的倍增位, 当 SMOD=1 时, 串行口波特率加倍 (对 Timer1 和方式 2 有用)

串口工作流程:

- 1、将端口设置为 UART 通讯端口;
- 2、选择工作模式, 计算波特率;
- 3、根据工作模式的选择, 配置 SCON;
- 4、根据波特率的要求, 若需要 Timer0/1 产生波特率, 则配置 Timer0/1 的工作模式;
- 5、配置 SMOD;
- 6、若数据接收, 等待接收中断, 从 SBUF 读取输入数据并清除中断后, 执行下一次操作; 若发送数据, 写数据到 SBUF, 等到发送中断达到后, 执行下一次操作。



方式 1 和方式 3 波特率计算公式：

Timer0 定时器产生波特率：Baundy =  $\frac{\text{Timer0 计数溢出率}}{16}$

Timer1 定时器产生波特率：Baundy =  $\frac{\text{Timer1 计数溢出率}}{16/32}$

#### 4. 10、定时器 2

TIMER2 定时/计数器有两种工作方式，可以选择单次计数或者循环计数。另外，TIMER2 还可以对系统时钟进行预分频。

在这里，有两类和定时/计数器相关的寄存器。第一类是存储实际的计数值，赋值给此类寄存器可以设定初始计数值，读取此类寄存器可以获得定时/计数器的内容。而第二类则是控制寄存器，通过 MCU 向 TMR2C 写入特定的值，可以控制定时/计数器的工作模式，并选择具体时钟源。

##### 1) 寄存器 TMR2C

SFR 地址：0x9B

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7]	TIMER2_IFG	R/W	TIMER2溢出标志位 0：无效标志位 1：有效标志位
[6]	TIMER2_CLKSEL	R/W	TIMER2时钟源选择位 0：f <sub>sys</sub> （系统时钟） 1：f <sub>sub</sub> （低速时钟）
[5]	TIMER2_SET	R/W	TIMER2计数方式选择 0：单次计数 1：循环计数
[4]	TIMER2_ON	R/W	TIMER2使能控制位 0：除能 1：使能
[3]	TIMER2_E	R/W	TIMER2计数控制位 0：计数暂停 1：计数使能
[2]	TIMER2_DL	R/W	TIMER2计数值锁存位 0：除能 1：使能

转下页

接上页

Bit	Name	R/W	Description
[1: 0]	TIMER2_PSC	R/W	TIMER2 预分频级数选择 00: $f_{T2}$ 01: $f_{T2}/2$ 10: $f_{T2}/4$ 11: $f_{T2}/8$

## 2) 寄存器 TMR2L

SFR 地址: 0x9E

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 0]	TMR2L	R/W	可通过该寄存器更新 TIMER2 初值低 8 位和显示当前计数值低 8 位 (需注意写入和读出 TMR2H 和 TMR2L 顺序)

## 3) 寄存器 TMR2H

SFR 地址: 0x9F

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 0]	TMR2H	R/W	可通过该寄存器更新 TIMER2 初值高 8 位和显示当前计数值高 8 位 (需注意写入和读出 TMR2H 和 TMR2L 顺序)

推荐工作流程:

- 1) 配置寄存器 TMR2C, 设置定时计数器的工作方式, 工作时钟源、预分频等;
- 2) 配置寄存器 TMR2L、TMR2H, 向 TIMER2 和写入初值,需先写 TMR2L, 后写 TMR2H; 若读取计数结果, 需在计数值锁存位置 1 后先读 TMR2L, 后读 TMR2H, 在读完 TMR2H 后计数值锁存位自动清 0;
- 3) 配置 TMR2C 中使能位启动 TIMER2 工作;
- 4) 单次计数模式下, 计数完成后 timer2\_on 会自动除能, 循环计数模式下软件配置除能;

计数周期计算公式:

$$\text{TIMER2 计数周期} = \frac{\text{预分频系数}}{f_{T2}} \times (65536 - \{\text{TMR2H, TMR2L}\})$$

#### 4.11、定时器 3

TIMER3 为 8bits 定时器。

在这里，有两类和定时/计数器相关的寄存器。第一类是存储实际的计数值，赋值给此类寄存器可以设定初始计数值，读取此类寄存器可以获得定时/计数器的内容。而第二类则是控制寄存器，通过 MCU 向 TMR3C 寄存器写入特定的值，可以控制定时/计数器的工作模式，并选择具体时钟源。

##### 1) TIMER3 控制寄存器

名称：TMR3C

SFR 地址：0xB5

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7]	TIMER3_IFG	R/W	TIMER3溢出标志位 0: 无效标志位 1: 有效标志位
[6]	TIMER3_DL	R/W	TIMER3锁存位（用于读取计数值） 0: 除能 1: 使能
[5]	TIMER3_CLKSEL	R/W	TIMER3时钟选择 0: $f_{SYS}$ （系统时钟） 1: $f_{SUB}$ （低速时钟）
[4]	TIMER3_ON	R/W	TIMER3使能控制位 0: 除能 1: 使能
[3]	TIMER3_E	R/W	TIMER3计数控制位 0: 计数暂停 1: 计数使能
[2: 0]	TIMER3_PSC	R/W	TIMER3 预分频级数选择 000: $f_{tm3}$ 001: $f_{tm3}/2$ 010: $f_{tm3}/4$ 011: $f_{tm3}/8$ 100: $f_{tm3}/16$ 101: $f_{tm3}/32$ 110: $f_{tm3}/64$ 111: $f_{tm3}/128$

timer3\_on 为 TIMER3 定时器使能位，为 0 时 TIMER3 定时器处于复位状态。

timer3\_e 为 TIMER3 定时器计数使能位，为 0 时 TIMER3 定时器处于暂停状态。

timer3\_psc 为 TIMER3 定时的预分配系数，使得定时器能够在更低的时钟频率下工作。

timer3\_ifg 为 TIMER3 的计数溢出中断标志位，软件检测到定时器中断后可以清除中断标志，往 timer3\_ifg 写 0。

## 2) TIMER3 存储实际值寄存器

名称：TMR3

SFR 地址：0xB7

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	TMR3	R/W	可通过该寄存器更新 TIMER3 初值和显示当前计数值

定时器在启动时会自动装载初值，且可在定时/计数器工作过程中更新初值，待溢出后自动更新初值；读取计数值时，需要先将 TIMER3\_DL 信号置 1，当读取完存储寄存器后，TIMER3\_DL 信号会自动清零，或者不读取存储寄存器，通过软件配置清零。

推荐工作流程：

- 1、配置寄存器 TMR3C，设置定时计数器的工作方式，工作时钟源、预分频等；
- 2、配置寄存器 TMR3，向 TIMER3 写入初值；
- 3、配置 TMR3C 中使能位启动 TIMER3 工作。

计数周期计算公式：

$$\text{TIMER3 计数周期} = \frac{\text{预分频系数}}{f_{T3}} \times (255 - \text{TMR3})$$

#### 4.12、看门狗

看门狗定时器的功能在于防止因电磁的干扰等外部不可控制事件所造成的程序不正常动作或跳转到未知的地址。

Watchdog 模块内部有一个定时器，当 Watchdog 使能时，该定时器自动开始计数，如果定时器溢出，Watchdog 模块将复位 MCU 或产生定时溢出中断信号。

Watchdog 模块的工作时钟为系统时钟，系统时钟频率由晶振时钟频率以及分频器参数决定，晶振时钟频率和分频器参数配置请参考时钟复位文档。

Watchdog 模块的寄存器全部分配在特殊功能寄存器中，软件应该通过特殊功能寄存器读写指令在控制 Watchdog 模块。由于 Watchdog 模块特殊功能寄存器为用户自定义特殊功能寄存器，芯片中无法支持 bit 寻址操作。

Watchdog 模块的计数时钟为 8kHz 或  $f_{SYS}/4$ ，在 8kHz 时钟域信号输出复位信号，在 MCU 休眠时仍可以产生 wdt\_rst 信号。

Watchdog 模块的计数器为 24 位计数器，高 16 位可配，模块根据寄存器配置的值，溢出产生溢出中断，或者溢出复位信号。

寄存器描述：

##### 1) WDCON 寄存器

SFR 地址：0xD8

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7]	WDT_IFG	R/W	Watchdog中断标志位，高有效，写0清中断
[6]	WDT_GATE_EN	R/W	Watchdog门控使能位，高有效
[5]	WDT_EN	R/W	Watchdog使能位，高有效
[4]	WDT_CLR	R/W	Watchdog清零位，软件写1，硬件自动清零
[3]	WDT_RE	R/W	Watchdog复位使能位，高有效
[2]			保留
[1]	FSYSON	R/W	FSYSON控制位 1：空闲模式使能 0：空闲模式失能
[0]			保留

##### 2) WDCNTM 寄存器

SFR 地址：0xD9

复位值：0x9a

访问：R/W

Bit	Name	R/W	Description
[7: 0]	WDT_CNTM	R/W	Watchdog计数周期的bit15~bit8

## 3) WDCNTH 寄存器

SFR 地址: 0xDA

复位值: 0x01

访问: R/W

Bit	Name	R/W	Description
[7: 0]	WDT_CNTH	R/W	Watchdog计数周期的bit23~bit16

注: 计数默认值为 1s。

## 4) XDATA\_CTRL 寄存器

SFR 地址: 0xDB

复位值: 0x00

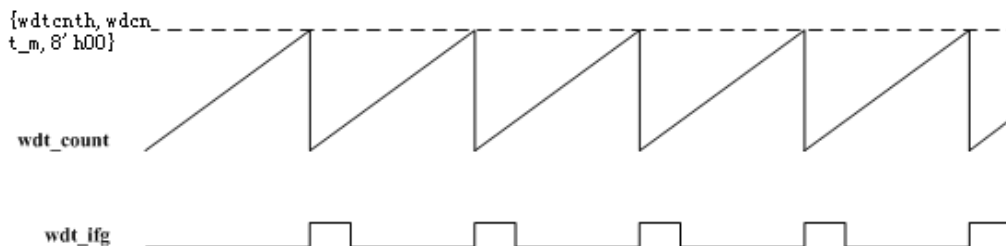
访问: R/W

Bit	Name	R/W	Description
[7]	WDT_PRT_CTRL	R/W	WDCON 寄存器中的 WDT_IE/WDT_RE/WDT_EN 位, 任何时候都能写 1, 如需清零, 需将该位置 1; 清零后, 该位也需置 0
[6: 1]		R/W	保留
[0]	XDATA_CTRL	R/W	XDATA的memory和reg读写使能控制位, 为0时, 允许读写

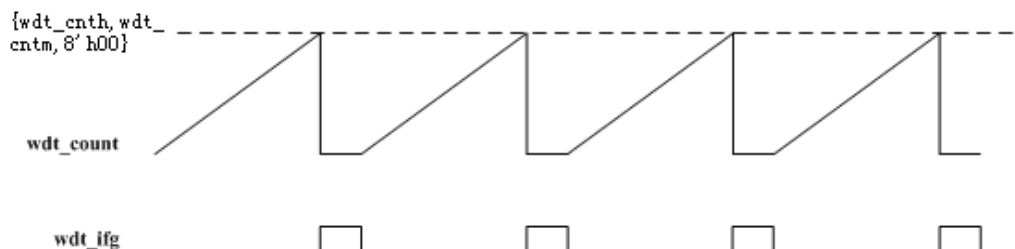
WDT\_GATE\_EN 为 Watchdog 定时器模式 (WDT\_RE 无效) 下的门控使能位。当门控使能位为高时, 如果 Watchdog 计数溢出, Watchdog 计数器将处于复位状态 (计数值保持为零并停止计数) 直到标志位 WDT\_IFG 清零后继续开始计数 (若 WDT\_EN 保持有效); 当门控使能为低时, 如果 Watchdog 计数溢出, Watchdog 继续进行下一轮计数 (若 WDT\_EN 保持有效), 不管标志位 WDT\_IFG 是否被清零, 详见门控使能无效时看门狗定时计数示意图和门控使能有效时看门狗定时计数示意图。如果 Watchdog 的复位使能有效 (WDT\_RE 有效), 当 Watchdog 计数溢出时将不产生中断标志和中断信号, 而是复位整个 MCU。

门控使能信号 WDT\_GATE\_EN 为低时可以实现精确的定时中断, 因为软件在进行中断处理过程时, Watchdog 定时器已经开始新一轮计数。软件应根据需要自行决定是否使用门控使能功能。

门控使能无效时看门狗定时计数示意图:

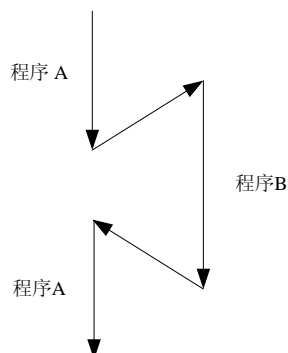


门控使能有效时看门狗定时计数示意图：



#### 4.13、中断

所谓中断，即打断 CPU 目前正在执行的程序而转向执行其他程序的一种 CPU 运行机制。CPU 正在执行程序 A，但是因为某些事件的发生使得 CPU 不得不暂时放弃目前的程序 A 而转向执行程序 B，在程序 B 执行完成之后再重新回来执行程序 A。这就是一个中断过程，其中程序 A 叫做被中断程序，而程序 B 叫做中断服务程序，引起中断的事件叫做中断源。芯片内部许多部件都可以作为中断源，比如 UART，定时器等。



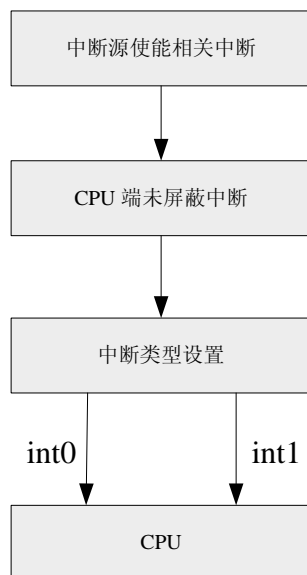
中断源：

序 号	模 块	描 述
1	TIM01_IRQ	来自定时器 0/定时器 1 的组合中断请求
2	TIM2_IRQ	来自定时器 2 中断请求
3	GPIO_IRQ	来自 GPIO 的组合中断请求
4	ADC_IRQ	ADC 中断请求
5	WDT_IRQ	看门狗功能复位中断请求
6	UART_IRQ	来自 UART 串口的组合中断请求
7	PWM0_IRQ	PWM0 中断请求
8	PWM1_IRQ	PWM1 中断请求
9	PWM2_IRQ	PWM2 中断请求
10	PWM3_IRQ	PWM3 中断请求
11	CMP_IRQ	比较器组合中断请求
12	TIM3_IRQ	来自定时器 3 中断请求

现根据上表所列出的中断向量，定义出下表中的寄存器，如超出此范围再扩充位数，如有富余，则在外面接“0”或取其低几位做归约操作。

在控制中断源的相应使能位开启时，中断源的控制就由中断控制模块控制，流程图如下。

中断控制模块流程：



现根据上表所列出的中断向量定义出下表中的寄存器的定义：

地 址	寄 存 器 名 称	位 数	访 问	复 位 值
0XF8	IRQSTATL[7: 0]	8	R	0x00
0XF9	IRQSTATH[3: 0]	4	R	0x00
0xFA	INTML[7: 0]	8	R/W	0xff
0xFB	INTMH[3: 0]	4	R/W	0x0f
0xFC	IRQMSKDL[7: 0]	8	R	0x00
0xFD	IRQMSKDH[3: 0]	4	R	0x00
0xFE	INTSELL[7: 0]	8	R/W	0x00
0xFF	INTSELH[3: 0]	4	R/W	0x00

IRQSTAT 是“中断请求状态”寄存器；INTM 是“中断屏蔽”寄存器；IRQ\_MSKD 是经过“屏蔽处理后的中断请求状态”寄存器；INTSEL0 是“中断选择”寄存器。



### 1) IRQSTATL 寄存器

地址：0xF8

初始值：0x00

访问：R

Bit	Name	R/W	Description
[7]	PWM1_IRQ	R	PWM1 中断请求
[6]	PWM0_IRQ	R	PWM0 中断请求
[5]	TIM2_IRQ	R	来自定时器 2 的中断请求
[4]	TIM01_IRQ	R	来自定时器 0/1 的中断请求
[3]	GPIO_IRQ	R	GPIO 的组合中断请求
[2]	UART_IRQ	R	来自串口的组合中断请求
[1]	ADC_IRQ	R	ADC 的中断请求
[0]	WDT_IRQ	R	看门狗功能复位中断请求

### 2) IRQSTATH 寄存器

地址：0xF9

初始值：0x00

访问：R

Bit	Name	R/W	Description
[7: 4]			保留
[3]	TIM3_IRQ	R	来自定时器 3 的中断请求
[2]	CMP_IRQ	R	比较器组合中断请求
[1]	PWM3_IRQ	R	PWM3 中断请求
[0]	PWM2_IRQ	R	PWM2 中断请求

### 3) INTML 寄存器

地址：0xFA

初始值：0xff

访问：R/W

Bit	Name	R/W	Description
[7]	PWM1_IM	R/W	PWM1_IM: PWM1_IRQ 的中断请求屏蔽 1b: 被屏蔽 0b: 未被屏蔽
[6]	PWM0_IM	R/W	PWM0_IM: PWM0_IRQ 的中断请求屏蔽 1b: 被屏蔽 0b: 未被屏蔽
[5]	TIM2_IM	R/W	TIM2_IM: TIM2_IRQ 的中断请求屏蔽 1b: 被屏蔽 0b: 未被屏蔽

转下页

接上页

Bit	Name	R/W	Description
[4]	TIM01_IM	R/W	TIM01_IM: TIM01_IRQ 的中断请求屏蔽 1b: 被屏蔽 0b: 未被屏蔽
[3]	GPIO_IM	R/W	GPIO_IM: GPIO_IRQ 的中断请求屏蔽 1b: 被屏蔽 0b: 未被屏蔽
[2]	UART_IM	R/W	UART_IM: UART_IRQ 的中断请求屏蔽 1b: 被屏蔽 0b: 未被屏蔽
[1]	ADC_IM,	R/W	ADC_IM: ADC_IRQ 的中断请求屏蔽 1b: 被屏蔽 0b: 未被屏蔽
[0]			保留

#### 4) INTMH 寄存器

地址: 0xFB

初始值: 0x0f

访问: R/W

Bit	Name	R/W	Description
[7: 4]			保留
[3]	TIM3_IM	R/W	TIM3_IM: TIM3_IRQ 的中断请求屏蔽 1b: 被屏蔽 0b: 未被屏蔽
[2]	CMP_IM	R/W	CMP_IM: CMP_IRQ 的中断请求屏蔽 1b: 被屏蔽 0b: 未被屏蔽
[1]	PWM3_IM	R/W	PWM3_IM: PWM3_IRQ 的中断请求屏蔽 1b: 被屏蔽 0b: 未被屏蔽
[0]	PWM2_IM	R/W	PWM2_IM: PWM2_IRQ 的中断请求屏蔽 1b: 被屏蔽 0b: 未被屏蔽

## 5) IRQMSKDL 寄存器

地址：0xFC

初始值：0x00

访问：R

Bit	Name	R/W	Description
[7]	PWM1_INT	R	PWM1 的中断=PWM1_IRQ&~PWM1_IM
[6]	PWM0_INT	R	PWM0 的中断=PWM0_IRQ&~PWM0_IM
[5]	TIM2_INT	R	定时器 2 的中断=TIM2_IRQ&~TIM2_IM
[4]	TIM01_INT	R	定时器 0/ 定时器 1 的组合中断=TIM01_IRQ&~TIM01_IM
[3]	GPIO_INT	R	GPIO 的中断=GPIO_IRQ&~GPIO_IM
[2]	UART_INT	R	UART 中断=UART_IRQ&~ UART_IM
[1]	ADC_INT	R	ADC 的中断=ADC_IRQ&~ADC_IM
[0]	WDT_INT	R	WDT 的中断=WDT_IRQ

## 6) IRQMSKDH 寄存器

地址：0xFD

初始值：0x00

访问：R

Bit	Name	R/W	Description
[7: 4]			保留
[3]	TIM3_INT	R	TIM3 的中断=TIM3_IRQ&~TIM3_IM
[2]	CMP_INT	R	CMP 的中断=CMP_IRQ&~CMP_IM
[1]	PWM3_INT	R	PWM3 的中断=PWM3_IRQ&~PWM3_IM
[0]	PWM2_INT	R	PWM2 的中断=PWM2_IRQ&~PWM2_IM

## 7) INTSELL 寄存器

地址：0xFE

初始值：0x00

访问：R/W

Bit	Name	R/W	Description
[7]	PWM1_IS	R/W	PWM1_INT 的中断输出口选择 1b: 从 INT0 输出 0b: 从 INT1 输出
[6]	PWM0_IS	R/W	PWM0_INT 的中断输出口选择 1b: 从 INT0 输出 0b: 从 INT1 输出
[5]	TIM2_IS	R/W	TIM2_INT 的中断输出口选择 1b: 从 INT0 输出 0b: 从 INT1 输出

转下页

接上页

Bit	Name	R/W	Description
[4]	TIM01_IS	R/W	TIM01_INT 的中断输出口选择 1b: 从 INT0 输出 0b: 从 INT1 输出
[3]	GPIO_IS	R/W	GPIO_INT 的中断输出口选择 1b: 从 INT0 输出 0b: 从 INT1 输出
[2]	UART_IS	R/W	UART_INT 的中断输出口选择 1b: 从 INT0 输出 0b: 从 INT1 输出
[1]	ADC_IS	R/W	ADC_INT 的中断输出口选择 1b: 从 INT0 输出 0b: 从 INT1 输出
[0]	WDT_IS	R/W	WDT_INT 的中断输出口选择 1b: 从 INT0 输出 0b: 从 INT1 输出

## 8) INTSELH 寄存器

地址: 0xFF

初始值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 4]			保留
[3]	TIM3_IS	R/W	TIM3_INT 的中断输出口选择 1b: 从 INT0 输出 0b: 从 INT1 输出
[2]	CMP_IS	R/W	CMP_INT 的中断输出口选择 1b: 从 INT0 输出 0b: 从 INT1 输出
[1]	PWM3_IS	R/W	PWM3_INT 的中断输出口选择 1b: 从 INT0 输出 0b: 从 INT1 输出
[0]	PWM2_IS	R/W	PWM2_INT 的中断输出口选择 1b: 从 INT0 输出 0b: 从 INT1 输出

#### 4.14、GPIO

电路提供 PA~PC 双向输入/ 输出口。这些寄存器的用途是将引脚配置为需要的功能，可以实现独立的引脚配置。所有引脚的上拉电阻设置也由软件控制。

##### (1) 上拉电阻

许多产品在端口处于输入状态时需要外加一个上拉电阻来实现上拉的功能。为了免去外部上拉电阻，当引脚设置为输入时，可由内部连接到一个上拉电阻。这些上拉电阻可通过寄存器 PAPU~PEPU 来设置，它用一个 PMOS 晶体管来实现上拉电阻功能。

##### (2) PA 口唤醒

当使用暂停指令“STOP”迫使单片机进入休眠模式，单片机的系统时钟将会停止以降低功耗，此功能对于电池及低功耗应用很重要。唤醒单片机有很多种方法，其中包括 PA 口上升沿或下降沿触发唤醒。该功能适用于通过外部开关来唤醒的应用。当 MCU 休眠时，可以通过 PA 管脚的任意一个端口来实现外部唤醒。也可以设置外部中断唤醒，通过时钟模块来打开 MCU 时钟。

##### (3) 寄存器控制引脚复用

PA~PC 的大部分引脚都复用了其他功能，复用功能可以通过寄存器来配置。但是一个时间只能选择一种功能，管脚控制模块通过配置寄存器控制多路开关来连接管脚与片内外设。

#### 寄存器描述

##### 1) WAKEUP\_EN 寄存器

地址：0x84

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7]	WAKEUP_EN[7]	R/W	PA7 唤醒功能控制位： 1：使能 0：除能
[6]	WAKEUP_EN[6]	R/W	PA6 唤醒功能控制位： 1：使能 0：除能
[5]	WAKEUP_EN[5]	R/W	PA5 唤醒功能控制位： 1：使能 0：除能
[4]	WAKEUP_EN[4]	R/W	PA4 唤醒功能控制位： 1：使能 0：除能

转下页

接上页

Bit	Name	R/W	Description
[3]	WAKEUP_EN[3]	R/W	PA3 唤醒功能控制位： 1：使能 0：除能
[2]	WAKEUP_EN[2]	R/W	PA2 唤醒功能控制位： 1：使能 0：除能
[1]	WAKEUP_EN[1]	R/W	PA1 唤醒功能控制位： 1：使能 0：除能
[0]	WAKEUP_EN[0]	R/W	PA0 唤醒功能控制位： 1：使能 0：除能

## 2) WAKEUP\_EDGE 寄存器

地址：0x85

复位值：0xff

访问：R/W

Bit	Name	R/W	Description
[7]	WAKEUP_EDGE[7]	R/W	PA7 唤醒触发方式： 1：下降沿触发 0：上升沿触发
[6]	WAKEUP_EDGE[6]	R/W	PA6 唤醒触发方式： 1：下降沿触发 0：上升沿触发
[5]	WAKEUP_EDGE[5]	R/W	PA5 唤醒触发方式： 1：下降沿触发 0：上升沿触发
[4]	WAKEUP_EDGE[4]	R/W	PA4 唤醒触发方式： 1：下降沿触发 0：上升沿触发
[3]	WAKEUP_EDGE[3]	R/W	PA3 唤醒触发方式： 1：下降沿触发 0：上升沿触发
[2]	WAKEUP_EDGE[2]	R/W	PA2 唤醒触发方式： 1：下降沿触发 0：上升沿触发

转下页

接上页

Bit	Name	R/W	Description
[1]	WAKEUP_EDGE[1]	R/W	PA1 唤醒触发方式： 1：下降沿触发 0：上升沿触发
[0]	WAKEUP_EDGE[0]	R/W	PA0 唤醒触发方式： 1：下降沿触发 0：上升沿触发

## 3) PAPU 寄存器

地址：0xC3

复位值：0xff

访问：R/W

Bit	Name	R/W	Description
[7]	PAPU7	R/W	PA7 上拉电阻控制位： 1：使能 0：除能
[6]	PAPU6	R/W	PA6 上拉电阻控制位： 1：使能 0：除能
[5]	PAPU5	R/W	PA5 上拉电阻控制位： 1：使能 0：除能
[4]	PAPU4	R/W	PA4 上拉电阻控制位： 1：使能 0：除能
[3]	PAPU3	R/W	PA3 上拉电阻控制位： 1：使能 0：除能
[2]	PAPU2	R/W	PA2 上拉电阻控制位： 1：使能 0：除能
[1]	PAPU1	R/W	PA1 上拉电阻控制位： 1：使能 0：除能
[0]	PAPU0	R/W	PA0 上拉电阻控制位： 1：使能 0：除能

#### 4) PBPU 寄存器

地址：0xC4

复位值：0xff

访问：R/W

Bit	Name	R/W	Description
[7]	PBPU7	R/W	PB7 上拉电阻控制位： 1：使能 0：除能
[6]	PBPU6	R/W	PB6 上拉电阻控制位： 1：使能 0：除能
[5]	PBPU5	R/W	PB5 上拉电阻控制位： 1：使能 0：除能
[4]	PBPU4	R/W	PB4 上拉电阻控制位： 1：使能 0：除能
[3]	PBPU3	R/W	PB3 上拉电阻控制位： 1：使能 0：除能
[2]	PBPU2	R/W	PB2 上拉电阻控制位： 1：使能 0：除能
[1]	PBPU1	R/W	PB1 上拉电阻控制位： 1：使能 0：除能
[0]	PBPU0	R/W	PB0 上拉电阻控制位： 1：使能 0：除能

#### 5) PCPU 寄存器

地址：0xC5

复位值：0x03

访问：R/W

Bit	Name	R/W	Description
[7: 2]			保留
[1]	PCPU1	R/W	PC1 上拉电阻控制位： 1：使能 0：除能
[0]	PCPU0	R/W	PC0 上拉电阻控制位： 1：使能 0：除能



## 6) MISC 寄存器

地址: 0xC6

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7]	UART_SEL	R/W	PA0, PA1 复用控制: 1: PA0 作 TXD 口, PA1 作 RXD 口 0: PA0 不作 TXD 口, PA1 不作 RXD 口
[6]			保留
[5]	ODE2	R/W	PC0 开漏极使能位 1b: 使能 0b: 除能
[4]	ODE1	R/W	PB5 开漏极使能位 1b: 使能 0b: 除能
[3]	ODE0	R/W	PB4 开漏极使能位 1b: 使能 0b: 除能
[2]	PCR10	R/W	定义 PB1 是否为 AD 输入 0b: 非 AD 输入 1b: AD 输入, AN12
[1]	PCR9	R/W	定义 PB7 是否为 AD 输入 0b: 非 AD 输入 1b: AD 输入, AN9
[0]	PCR8	R/W	定义 PB6 是否为 AD 输入 0b: 非 AD 输入 1b: AD 输入, AN8

Note: 端口中 PB4/PB5/PC0 可通过 MISC 寄存器的 ODE0~ODE2 位设置为开漏极结构。

## 7) ADPCR 寄存器

地址: 0xC7

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7]	PCR7	R/W	定义 PB5 是否为 AD 输入 0b: 非 AD 输入 1b: AD 输入, AN7
[6]	PCR6	R/W	定义 PB4 是否为 AD 输入 0b: 非 AD 输入 1b: AD 输入, AN6
[5]	PCR5	R/W	定义 PB3 是否为 AD 输入 0b: 非 AD 输入 1b: AD 输入, AN5
[4]	PCR4	R/W	定义 PB2 是否为 AD 输入 0b: 非 AD 输入 1b: AD 输入, AN4
[3]	PCR3	R/W	定义 PB0 是否为 AD 输入 0b: 非 AD 输入 1b: AD 输入, AN3
[2]	PCR2	R/W	定义 PA5 是否为 AD 输入 0b: 非 AD 输入 1b: AD 输入, AN2
[1]	PCR1	R/W	定义 PA3 是否为 AD 输入 0b: 非 AD 输入 1b: AD 输入, AN1
[0]	PCR0	R/W	定义 PA1 是否为 AD 输入 0b: 非 AD 输入 1b: AD 输入, AN0

## 8) PAREG 寄存器

SFR 地址: 0xA1

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7]	PA7_DATA	R/W	PA7 输入/输出数据 PA7_DIR=1, PA7_DATA 为 PA7 输出数据 PA7_DIR=0, PA7_DATA 为 PA7 输入数据
[6]	PA6_DATA	R/W	PA6 输入/输出数据 PA6_DIR=1, PA6_DATA 为 PA6 输出数据 PA6_DIR=0, PA6_DATA 为 PA6 输入数据
[5]	PA5_DATA	R/W	PA5 输入/输出数据 PA5_DIR=1, PA5_DATA 为 PA5 输出数据 PA5_DIR=0, PA5_DATA 为 PA5 输入数据
[4]	PA4_DATA	R/W	PA4 输入/输出数据 PA4_DIR=1, PA4_DATA 为 PA4 输出数据 PA4_DIR=0, PA4_DATA 为 PA4 输入数据
[3]	PA3_DATA	R/W	PA3 输入/输出数据 PA3_DIR=1, PA3_DATA 为 PA3 输出数据 PA3_DIR=0, PA3_DATA 为 PA3 输入数据
[2]	PA2_DATA	R/W	PA2 输入/输出数据 PA2_DIR=1, PA2_DATA 为 PA2 输出数据 PA2_DIR=0, PA2_DATA 为 PA2 输入数据
[1]	PA1_DATA	R/W	PA1 输入/输出数据 PA1_DIR=1, PA1_DATA 为 PA1 输出数据 PA1_DIR=0, PA1_DATA 为 PA1 输入数据
[0]	PA0_DATA	R/W	PA0 输入/输出数据 PA0_DIR=1, PA0_DATA 为 PA0 输出数据 PA0_DIR=0, PA0_DATA 为 PA0 输入数据

# 9) PADIR 寄存器

SFR 地址: 0xA2

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7]	PA7_DIR	R/W	PA7 数据输入输出方向 1: 输出 0: 输入
[6]	PA6_DIR	R/W	PA6 数据输入输出方向 1: 输出 0: 输入
[5]	PA5_DIR	R/W	PA5 数据输入输出方向 1: 输出 0: 输入
[4]	PA4_DIR	R/W	PA4 数据输入输出方向 1: 输出 0: 输入
[3]	PA3_DIR	R/W	PA3 数据输入输出方向 1: 输出 0: 输入
[2]	PA2_DIR	R/W	PA2 数据输入输出方向 1: 输出 0: 输入
[1]	PA1_DIR	R/W	PA1 数据输入输出方向 1: 输出 0: 输入
[0]	PA0_DIR	R/W	PA0 数据输入输出方向 1: 输出 0: 输入

### 10) PAIE 寄存器

SFR 地址: 0xA3

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7]	PA7_IE	R/W	PA7 中断使能信号, 高有效
[6]	PA6_IE	R/W	PA6 中断使能信号, 高有效
[5]	PA5_IE	R/W	PA5 中断使能信号, 高有效
[4]	PA4_IE	R/W	PA4 中断使能信号, 高有效
[3]	PA3_IE	R/W	PA3 中断使能信号, 高有效
[2]	PA2_IE	R/W	PA2 中断使能信号, 高有效
[1]	PA1_IE	R/W	PA1 中断使能信号, 高有效
[0]	PA0_IE	R/W	PA0 中断使能信号, 高有效

### 11) PAIES\_L 寄存器

SFR 地址: 0xA4

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 6]	PA3_IES	R/W	PA3 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断
[5: 4]	PA2_IES	R/W	PA2 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断
[3: 2]	PA1_IES	R/W	PA1 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断
[1: 0]	PA0_IES	R/W	PA0 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断

## 12) PAIES\_H 寄存器

SFR 地址: 0xA5

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 6]	PA7_IE	R/W	PA7 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断
[5: 4]	PA6_IE	R/W	PA6 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断
[3: 2]	PA5_IE	R/W	PA5 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断
[1: 0]	PA4_IE	R/W	PA4 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断

## 13) PAIFG 寄存器

SFR 地址: 0xA6

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7]	PA7_IFG	R/W	PA7 中断标志信号, 高有效
[6]	PA6_IFG	R/W	PA6 中断标志信号, 高有效
[5]	PA5_IFG	R/W	PA5 中断标志信号, 高有效
[4]	PA4_IFG	R/W	PA4 中断标志信号, 高有效
[3]	PA3_IFG	R/W	PA3 中断标志信号, 高有效
[2]	PA2_IFG	R/W	PA2 中断标志信号, 高有效
[1]	PA1_IFG	R/W	PA1 中断标志信号, 高有效
[0]	PA0_IFG	R/W	PA0 中断标志信号, 高有效

#### 14) PBREG 寄存器

SFR 地址: 0xA9

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7]	PB7_DATA	R/W	PB7 输入/输出数据 PB7_DIR=1, PB7_DATA 为 PB7 输出数据 PB7_DIR=0, PB7_DATA 为 PB7 输入数据
[6]	PB6_DATA	R/W	PB6 输入/输出数据 PB6_DIR=1, PB6_DATA 为 PB6 输出数据 PB6_DIR=0, PB6_DATA 为 PB6 输入数据
[5]	PB5_DATA	R/W	PB5 输入/输出数据 PB5_DIR=1, PB5_DATA 为 PB5 输出数据 PB5_DIR=0, PB5_DATA 为 PB5 输入数据
[4]	PB4_DATA	R/W	PB4 输入/输出数据 PB4_DIR=1, PB4_DATA 为 PB4 输出数据 PB4_DIR=0, PB4_DATA 为 PB4 输入数据
[3]	PB3_DATA	R/W	PB3 输入/输出数据 PB3_DIR=1, PB3_DATA 为 PB3 输出数据 PB3_DIR=0, PB3_DATA 为 PB3 输入数据
[2]	PB2_DATA	R/W	PB2 输入/输出数据 PB2_DIR=1, PB2_DATA 为 PB2 输出数据 PB2_DIR=0, PB2_DATA 为 PB2 输入数据
[1]	PB1_DATA	R/W	PB1 输入/输出数据 PB1_DIR=1, PB1_DATA 为 PB1 输出数据 PB1_DIR=0, PB1_DATA 为 PB1 输入数据
[0]	PB0_DATA	R/W	PB0 输入/输出数据 PB0_DIR=1, PB0_DATA 为 PB0 输出数据 PB0_DIR=0, PB0_DATA 为 PB0 输入数据

## 15) PBDIR 寄存器

SFR 地址: 0xAA

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7]	PB7_DIR	R/W	PB7 数据输入输出方向 1: 输出 0: 输入
[6]	PB6_DIR	R/W	PB6 数据输入输出方向 1: 输出 0: 输入
[5]	PB5_DIR	R/W	PB5 数据输入输出方向 1: 输出 0: 输入
[4]	PB4_DIR	R/W	PB4 数据输入输出方向 1: 输出 0: 输入
[3]	PB3_DIR	R/W	PB3 数据输入输出方向 1: 输出 0: 输入
[2]	PB2_DIR	R/W	PB2 数据输入输出方向 1: 输出 0: 输入
[1]	PB1_DIR	R/W	PB1 数据输入输出方向 1: 输出 0: 输入
[0]	PB0_DIR	R/W	PB0 数据输入输出方向 1: 输出 0: 输入



## 16) PBIE 寄存器

SFR 地址: 0xAB

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7]	PB7_IE	R/W	PB7 中断使能信号, 高有效
[6]	PB6_IE	R/W	PB6 中断使能信号, 高有效
[5]	PB5_IE	R/W	PB5 中断使能信号, 高有效
[4]	PB4_IE	R/W	PB4 中断使能信号, 高有效
[3]	PB3_IE	R/W	PB3 中断使能信号, 高有效
[2]	PB2_IE	R/W	PB2 中断使能信号, 高有效
[1]	PB1_IE	R/W	PB1 中断使能信号, 高有效
[0]	PB0_IE	R/W	PB0 中断使能信号, 高有效

## 17) PBIES\_L 寄存器

SFR 地址: 0xAC

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 6]	PB3_IES	R/W	PB3 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断
[5: 4]	PB2_IES	R/W	PB2 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断
[3: 2]	PB1_IES	R/W	PB1 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断
[1: 0]	PB0_IES	R/W	PB0 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断

### 18) PBIES\_H 寄存器

SFR 地址: 0xAD

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 6]	PB7_IE	R/W	PB7 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断
[5: 4]	PB6_IE	R/W	PB6 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断
[3: 2]	PB5_IE	R/W	PB5 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断
[1: 0]	PB4_IE	R/W	PB4 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断

### 19) PBIFG 寄存器

SFR 地址: 0xAE

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7]	PB7_IFG	R/W	PB7 中断标志信号, 高有效
[6]	PB6_IFG	R/W	PB6 中断标志信号, 高有效
[5]	PB5_IFG	R/W	PB5 中断标志信号, 高有效
[4]	PB4_IFG	R/W	PB4 中断标志信号, 高有效
[3]	PB3_IFG	R/W	PB3 中断标志信号, 高有效
[2]	PB2_IFG	R/W	PB2 中断标志信号, 高有效
[1]	PB1_IFG	R/W	PB1 中断标志信号, 高有效
[0]	PB0_IFG	R/W	PB0 中断标志信号, 高有效

## 20) PCREG 寄存器

SFR 地址: 0xB1

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 2]			保留
[1]	PC1_DATA	R/W	PC1 输入/输出数据 PC1_DIR=1, PC1_DATA 为 PC1 输出数据 PC1_DIR=0, PC1_DATA 为 PC1 输入数据
[0]	PC0_DATA	R/W	PC0 输入/输出数据 PC0_DIR=1, PC0_DATA 为 PC0 输出数据 PC0_DIR=0, PC0_DATA 为 PC0 输入数据

## 21) PCDIR 寄存器

SFR 地址: 0xB2

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 2]			保留
[1]	PC1_DIR	R/W	PC1 数据输入输出方向 1: 输出 0: 输入
[0]	PC0_DIR	R/W	PC0 数据输入输出方向 1: 输出 0: 输入

## 22) PCIE 寄存器

SFR 地址: 0xB3

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 2]			保留
[1]	PC1_IE	R/W	PC1 中断使能信号, 高有效
[0]	PC0_IE	R/W	PC0 中断使能信号, 高有效

## 23) PCIES\_L 寄存器

SFR 地址: 0xB4

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 4]			保留
[3: 2]	PC1_IES	R/W	PC1 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断
[1: 0]	PC0_IES	R/W	PC0 中断方式选择信号 00: 除能 01: 上升沿中断 10: 下降沿中断 11: 双沿中断

## 24) PCIFG 寄存器

SFR 地址: 0xB6

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7: 2]			保留
[1]	PC1_IFG	R/W	PC1 中断标志信号, 高有效
[0]	PC0_IFG	R/W	PC0 中断标志信号, 高有效

#### 4.15、脉冲宽度调制器（PWM）

PWM 有着广泛的应用，大多数情况下用于控制模拟电路。因为数字信号连续变化的速率相对较快（当然取决于信号周期），因此最终会形成一个用来控制模拟电路的平均电压值。当 PWM 脉冲应用于马达时，马达的转速正比于占空比（从 0%到 100%）。如果占空比增加，马达转速就会提高，反之，如果占空比减小，马达的转速随之也会降低。

PWM 工作特点：

- 1) PWM 的初始电平可以通过 MCU 进行配置；
- 2) PWM 可设置为单次输出和循环输出，单次输出完成后，恢复所设置的初始电平；
- 3) 周期、占空比寄存器值可以更新，PWM 的输出将会在本周期完成后进行调整。

##### 1) PWM\_PERIOD0L 寄存器

地址：0xC2

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PERIOD0L	R/W	PWM0 的周期低 8 位

##### 2) PWM\_PERIOD0H 寄存器

地址：0xC8

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PERIOD0H	R/W	PWM0 的周期高 8 位

##### 3) PWM\_PULSE0L 寄存器

地址：0xC9

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PULSE0L	R/W	PWM0 的占空比低 8 位

##### 4) PWM\_PULSE0H 寄存器

地址：0xCA

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PULSE0H	R/W	PWM0 的占空比高 8 位

## 5) PWM\_PERIOD1L 寄存器

地址：0xCB

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PERIOD1L	R/W	PWM1 的周期低 8 位

## 6) PWM\_PERIOD1H 寄存器

地址：0xCC

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PERIOD1H	R/W	PWM1 的周期高 8 位

## 7) PWM\_PULSE1L 寄存器

地址：0xCD

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PULSE1L	R/W	PWM1 的占空比低 8 位

## 8) PWM\_PULSE1H 寄存器

地址：0xCE

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PULSE1H	R/W	PWM1 的占空比高 8 位

## 9) PWM\_PERIOD2L 寄存器

地址：0xCF

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PERIOD2L	R/W	PWM2 的周期低 8 位

## 10) PWM\_PERIOD2H 寄存器

地址：0xD1

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PERIOD2H	R/W	PWM2 的周期高 8 位

## 11) PWM\_PULSE2L 寄存器

地址：0xD2

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PULSE2L	R/W	PWM2 的占空比低 8 位

## 12) PWM\_PULSE2H 寄存器

地址：0xD3

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PULSE2H	R/W	PWM2 的占空比高 8 位

## 13) PWM\_PERIOD3L 寄存器

地址：0xD4

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PERIOD3L	R/W	PWM3 的周期低 8 位

## 14) PWM\_PERIOD3H 寄存器

地址：0xD5

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PERIOD3H	R/W	PWM3 的周期高 8 位

## 15) PWM\_PULSE3L 寄存器

地址：0xD6

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PULSE3L	R/W	PWM3 的占空比低 8 位

## 16) PWM\_PULSE3H 寄存器

地址：0xD7

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	PWM_PULSE3H	R/W	PWM3 的占空比高 8 位

## 17) PWM\_CTRL1 寄存器

地址：0xDC

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7]	PWM3_LOAD	R/W	PWM3 周期装载控制位,当高低位配置完成置 1 0b: PWM3 装载除能 1b: PWM3 装载使能
[6]	PWM2_LOAD	R/W	PWM2 周期装载控制位,当高低位配置完成置 1 0b: PWM2 装载除能 1b: PWM2 装载使能
[5]	PWM1_LOAD	R/W	PWM1 周期装载控制位,当高低位配置完成置 1 0b: PWM1 装载除能 1b: PWM1 装载使能
[4]	PWM0_LOAD	R/W	PWM0 周期装载控制位,当高低位配置完成置 1 0b: PWM0 装载除能 1b: PWM0 装载使能
[3]	PWM3_OE	R/W	PWM3 输出模式 0b: 循环输出 1b: 单次输出
[2]	PWM2_OE	R/W	PWM2 输出模式 0b: 循环输出 1b: 单次输出
[1]	PWM1_OE	R/W	PWM1 输出模式 0b: 循环输出 1b: 单次输出

转下页



接上页

Bit	Name	R/W	Description
[0]	PWM0_OE	R/W	PWM0 输出模式 0b: 循环输出 1b: 单次输出

## 18) PWM\_CTRL2 寄存器

地址: 0xDD

复位值: 0x00

访问: R/W

Bit	Name	R/W	Description
[7]	PWM3_EN	R/W	PWM3 输出使能 0b: 禁止 PWM3 输出 (单次触发模式, 该位自动清零, 循环模式, 需 MCU 将其清零。) 1b: 开启 PWM3 输出
[6]	PWM2_EN	R/W	PWM2 输出使能 0b: 禁止 PWM2 输出 (单次触发模式, 该位自动清零, 循环模式, 需 MCU 将其清零。) 1b: 开启 PWM2 输出
[5]	PWM1_EN	R/W	PWM1 输出使能 0b: 禁止 PWM1 输出 (单次触发模式, 该位自动清零, 循环模式, 需 MCU 将其清零。) 1b: 开启 PWM1 输出
[4]	PWM0_EN	R/W	PWM0 输出使能 0b: 禁止 PWM0 输出 (单次触发模式, 该位自动清零, 循环模式, 需 MCU 将其清零。) 1b: 开启 PWM0 输出
[3]	INT_SEL3	R/W	PWM3 中断模式选择: 0b: PWM3 的上升沿产生中断 1b: PWM3 的下降沿产生中断
[2]	INT_SEL2	R/W	PWM2 中断模式选择: 0b: PWM2 的上升沿产生中断 1b: PWM2 的下降沿产生中断
[1]	INT_SEL1	R/W	PWM1 中断模式选择: 0b: PWM1 的上升沿产生中断 1b: PWM1 的下降沿产生中断
[0]	INT_SEL0	R/W	PWM0 中断模式选择: 0b: PWM0 的上升沿产生中断 1b: PWM0 的下降沿产生中断

## 19) PWM\_CTRL3 寄存器

地址：0xDE

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 5]			保留
[4]	PWM1N_STOP	R/W	PWM1N 互补输出使能： 0b: PWM1N 输出 1b: PWM1N 不输出
[3]	PWM3_IRQ	R/W	pwm3 中断信号（软件清零） 0b: 中断清零 1b: 产生中断
[2]	PWM2_IRQ	R/W	pwm2 中断信号（软件清零） 0b: 中断清零 1b: 产生中断
[1]	PWM1_IRQ	R/W	pwm1 中断信号（软件清零） 0b: 中断清零 1b: 产生中断
[0]	PWM0_IRQ	R/W	pwm0 中断信号（软件清零） 0b: 中断清零 1b: 产生中断

推荐 PWM 控制器工作流程：

MCU 设置\读取 PWM 控制模块寄存器流程：

- 1) PWM 使能寄存器默认为 8'h00，处于关闭。MCU 为相应位写“1”开启相应 PWM；
- 2) MCU 根据需要配置 PWM 周期和脉宽控制寄存器，将 PWMX\_LOAD 位写“1”，表示参数配置完成。在 PWM 未使能时，周期、占空比直接写入；
- 3) 配置输出模式，中断模式，再打开输出使能，将相应管脚配置为 PWM 输出；
- 4) 内部计数器根据设定周期计数并与脉宽寄存器比较得到 PWM 周期波形；
- 5) MCU 可读当前 PWM 的周期和脉宽以及控制信号信息；
- 6) 设置好某路 PWM 的周期、脉宽及触发模式，然后 MCU 开启对应使能位，即输出想要的 PWM；默认为循环模式，由 MCU 关闭使能结束输出；如在单次触发使能位写“1”，则为单次触发模式，完成一个周期 PWM 输出同时清除对应使能位；为单次触发使能位写“0”，则进入循环模式；
- 7) 在工作期间，需要更改周期、占空比信息，应先重新配置相应的寄存器，并将 PWM\_LOAD 使能，待本周输出完成后，会自动装载，并硬件自动将 load 信号清零。

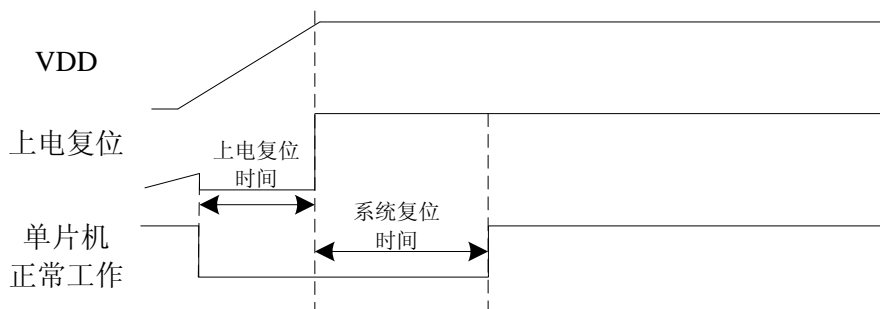
#### 4.16、复位

单片机共有 3 种复位方式：

##### 1) 上电复位

发生在单片机上电后。除了保证程序存储器从开始地址执行，上电复位也使得其它寄存器被设定在预设条件。所有的输入/输出端口控制寄存器在上电复位时会保持高电平，以确保上电后所有引脚被设定为输入状态。下图为上电复位的时序图，其中系统复位时间的典型值为 25ms。

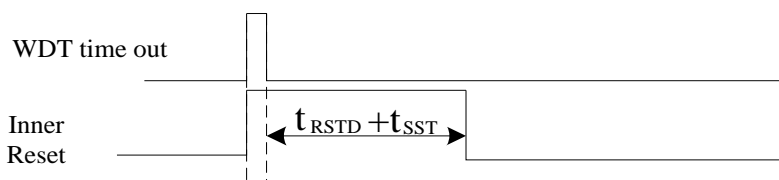
上电复位时序图：



##### 2) 正常运行时看门狗溢出复位

除了看门狗溢出标志位 TO 将被设为“1”之外，正常运行时看门狗溢出复位和 RES 复位相同。

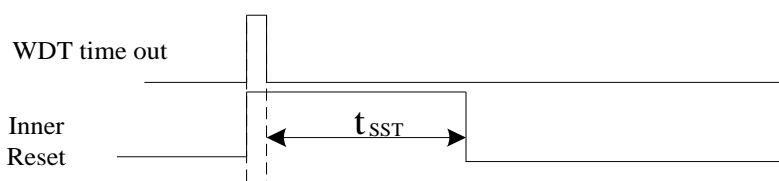
正常运行时看门狗溢出复位时序图：



##### 3) 休眠或空闲时看门狗溢出复位

休眠或空闲时看门狗溢出复位和其它种类的复位有些不同。除了程序计数器与堆栈指针将被清“0”及 TO 位被设为“1”外，绝大部分的条件保持不变。

休眠或空闲时看门狗溢出复位时序图：



#### 4.17、温度传感器

电路提供内部集成的温度传感器，温度传感器电压可通过 ADC 进行转换。温度传感器采集的是芯片工作的环境温度，可通过寄存器 BANDGAP\_TEMP 中的 TEMP\_EN 来使能温度传感器。另外，BANDGAP\_EN 位为带隙基准电压模块的使能控制位，默认为使能。若 BANDGAP\_EN 设置为除能，则 LDO 模块和温度传感器模块无法正常工作。

温度传感器的基准温度（25℃）对应的典型电压值约为 1.336V，温度传感器的斜率约为 -3.4mV/℃。

BANDGAP\_TEMP 控制寄存器：

地址：0xF7

复位值：0x01

访问：R/W

Bit	Name	R/W	Description
[7: 2]			保留，读出为 0
[1]	TEMP_EN	R/W	温度传感器使能控制位 0：除能 1：使能
[0]	BANDGAP_EN	R/W	带隙电压使能控制位 0：除能 1：使能

#### 4. 18、LDO 功能

电路内部集成的 LDO 稳压器，用于产生一个稳定的电压。LDO 电路可提供 1.8V、2.048V、2.4V 和 3.072V 四个输出电压可选，可由寄存器 LDOC 控制选择。LDO 输出的选项可由寄存器控制，为 A/D 转换器参考电压及外部器件的供电电源提供一个稳定的电压参考。

寄存器 LDOC:

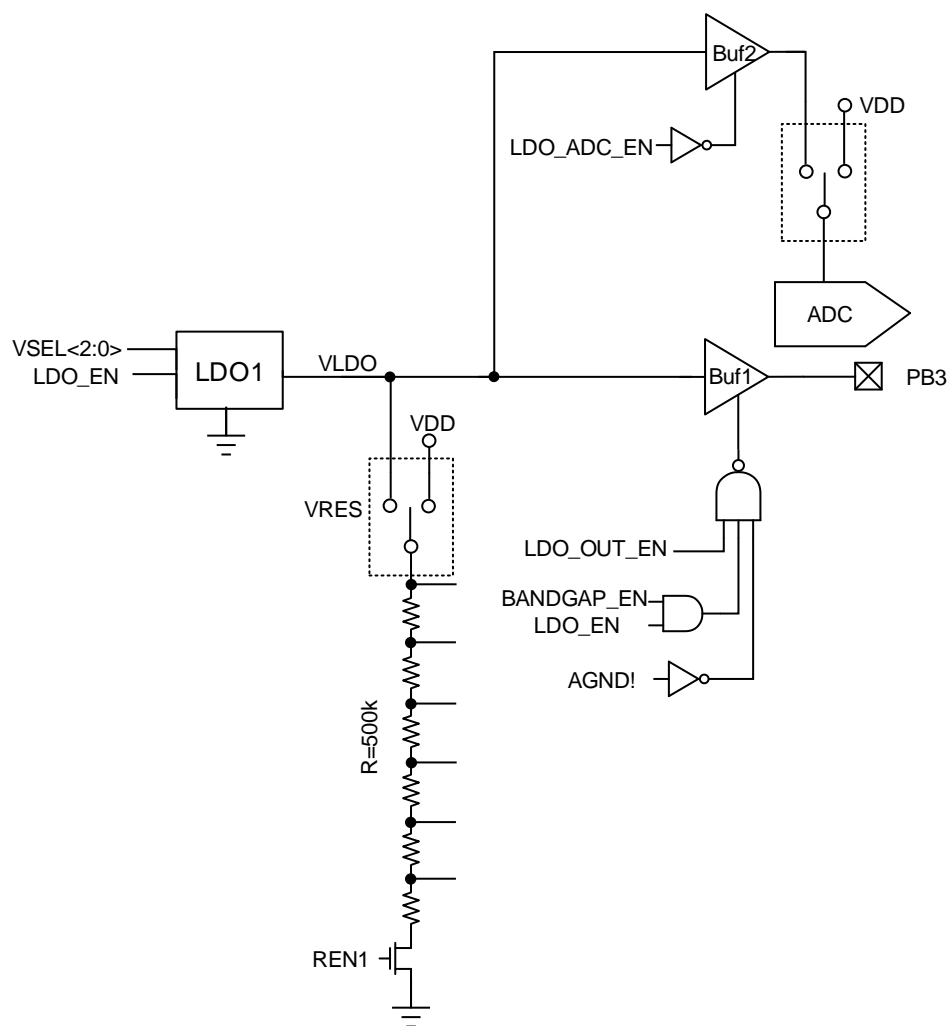
地址: 0xF6

复位值: 0x03

访问: R/W

Bit	Name	R/W	Description
[7: 6]			保留
[5]	LDO_ADC_EN	R/W	ADC 参考电压控制位 0: 除能 1: 使能
[4]	LDO_OUT_EN	R/W	LDO 输出电压控制位 0: 除能 1: 使能
[3]	LDOEN	R/W	LDO 控制位 0: 关闭 1: 使能
[2: 0]	LDO_VSEL	R/W	LDO 输出电压选择位 011: 1.8V 101: 2.048V 110: 2.4V 111: 3.072V

LDO 的控制电路图:

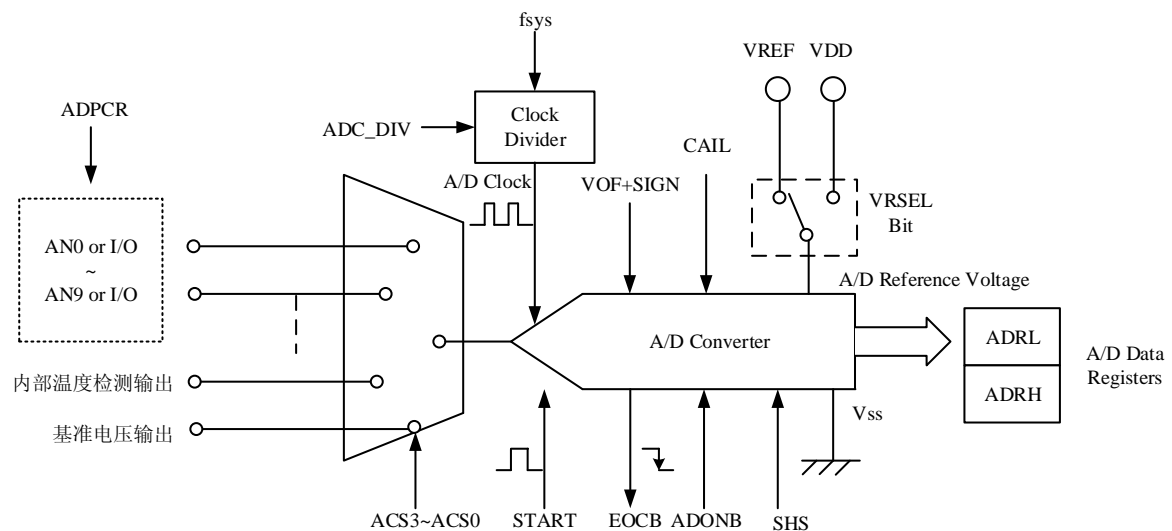


#### 4. 19、A/D 转换器

A/D 转换器的所有工作由五个寄存器控制：两个只读寄存器（ADRH、ADRL）来存放 12 位 ADC 数据的值，另外三个控制寄存器（ADCR、ADCV、ADIFG）设置 A/D 转换器的操作和控制功能。如下表所示：

寄存器名称	Bit							
	7	6	5	4	3	2	1	0
ADCR	START	ADONB	VRSEL [1]	VRSEL [0]	ACS3	ACS2	ACS1	ACS0
ADCV	EOCB	VOF [2]	VOF[1]	VOF[0]	SIGN	CAIL _EN	SHS [1]	SHS [0]
ADIFG								ADIFG

A/D 转换器内部结构和相关的寄存器：



寄存器 ADCR，ADCV，ADIFG 用来控制 A/D 转换器的功能和操作。

ADCR 寄存器中的 START 位，用于打开和复位 A/D 转换器：当软件设定此位从逻辑低到逻辑高，然后再到逻辑低，就会开始一个 A/D 转换周期；当 START 位从逻辑低到逻辑高，但不再回到逻辑低时，复位 A/D 转换器。

ADCR 寄存器中的 EOCB 位用于表明 A/D 转换过程的完成：在转换周期结束后，EOCB 位会被硬件置为“0”，同时也会置位中断控制寄存器内相应的 A/D 中断请求标志位，如果中断使能，就会产生对应的内部中断信号，A/D 内部中断信号将引导程序到相应的 A/D 内部中断入口，如果 A/D 内部中断被禁止，可以让单片机轮询 EOCB 位，检查此位是否被清零，以作为另一种侦测 A/D 转换周期结束的方法。

ADCR 寄存器中的 ADONB 位用于控制 A/D 转换电路电源的开/关：该位必须置 0 以开启 A/D 转换器电源，只要 ADONB 设为“0”，就会产生功耗，因此当未使用 A/D 转换器功能时，在功耗敏感的应用中建议设置 ADONB 为 1 以减少功耗。

ADCR 寄存器中的 VRSEL 位用于选择 A/D 转换器的基准电压：可以选择来自正电源电压 VDD 或外部参考源引脚 VREF 或内部 LDO 输出。ADCR 寄存器中的 ACS3~ACS0 用于选择连接至内部 A/D 转换器的模拟通道：电路中的 12 个模拟输入中的每一个都需要分别被发送到 A/D 转换器。

ADPCR 寄存器用来定义 PB 口、PC 口中的哪些引脚为 A/D 转换器的模拟输入，哪些引脚不作为 A/D 转换输入。相应位设为高将选择 A/D 输入功能，设为低将选择 I/O 或其它引脚共用功能。当引脚作为 A/D 输入时，其原来的 I/O 或其它引脚共用功能消失，此外，其内部上拉电阻也将自动断开。

#### 寄存器描述

##### 1) ADRL 寄存器

地址：0xF1

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 4]	ADRL[7: 4]	R	A/D 转换结果的低四位
[3: 0]			保留，读为 0

##### 2) ADRH 寄存器

地址：0xF2

复位值：0x00

访问：R/W

Bit	Name	R/W	Description
[7: 0]	ADRH[7: 0]	R	A/D 转换结果的高八位



## 3) ADCR 寄存器

地址: 0xF3

复位值: 0x60

访问: R/W

Bit	Name	R/W	Description
[7]	START	R/W	启动 A/D 转换位 0→1: 启动一次 A/D 转换, 高电平持续时间大于 ADC 采样时间 配置 START 为 1 会开始 AD 转换, START 信号由硬件自动清零
[6]	ADONB	R/W	A/D 转换器模块电源开启/ 关闭控制位 1: A/D 转换器模块电源关闭 0: A/D 转换器模块电源开启
[5: 4]	VRSEL	R/W	A/D 转换器参考电压选择位 00: VREF 取自电源电压 AVDD 10: 内部 LDO 输出
[3: 0]	ACS[3: 0]	R/W	选择 A/D 通道 0000: AN0 0001: AN1 0010: AN2 0011: AN3 0100: AN4 0101: AN5 0110: AN6 0111: AN7 1000: AN8 1001: AN9 1010: AN10, 内部温感电压 ADC 检测通道 1011: AN11, 基准电压作为 ADC 输入 1100: AN12 这些位是 A/D 通道选择控制位。由于只包含一个内部 A/D 转换电路, 因此通过这些位将 12 个 A/D 输入连接到转换器。

#### 4) ADCV 寄存器

地址：0xF4

复位值：0x80

访问：R/W

Bit	Name	R/W	Description
[7]	EOCB	R	A/D 转换结束标志 1: A/D 转换结束 0: A/D 转换中 此位用于表明 A/D 转换过程的完成。当转换正在进行，此位为低，当转换完成时为高，
[6: 4]	VOF[2: 0]	R/W	电压校正
[3]	SIGN	R/W	校正电压的极性 0: 负 1: 正
[2]	CAIL_EN	R/W	精度校正使能 0: 禁止校正 1: 使能校正
[1: 0]	SHS[1: 0]	R/W	控制 AD 信号切换时间 00: $2T_{AD}$ 01: $4T_{AD}$ 10: $8T_{AD}$ 11: $12T_{AD}$

#### 5) ADIFG 寄存器

地址：0xF5

复位值：0x00

访问：R/W

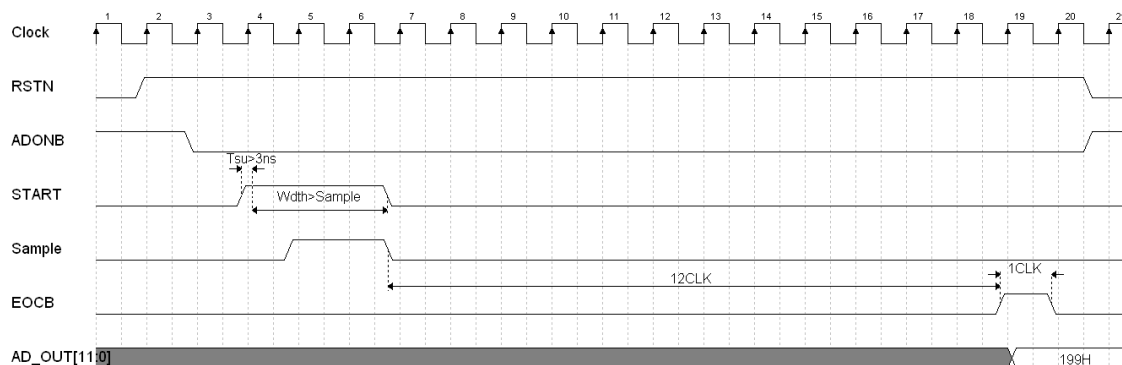
Bit	Name	R/W	Description
[7: 1]			保留，读为 0
[0]	ADIFG	R/W	A/D 中断标志 0: 无效标志位 1: 有效标志位

以下是 A/D 转换步骤：

- (1) 配置 ADCR 寄存器中的 VRSEL 位和 ACS 位选择参考电压和模拟电压的输入通道；
- (2) 配置 ADCV 寄存器，使能精度校正，配置校正电压的极性以及校正电压值，配置 SHS 采样保持时间；
- (3) 配置 ADCR 寄存器，打开 AD 转换模块电源，配置 START 信号为 1 开始 AD 转换。

注：若使用轮询 ADCR 寄存器中 EOCB 位的状态的方法来检查转换过程是否结束时，则中断使能的步骤可以省略。

下图表示模数转换过程的图形与时序，SHS<1:0>决定采样保持时间是 2/4/8/12 个 CLK。由应用程序控制开始 A/D 转换过程后，内部就会开始进行转换，在这个过程中，程序可以继续其它功能。



## 4.20、比较器

电路中含有两个模拟比较器。它们具有暂停、中断等功能，可通过寄存器进行灵活配置。比较器的引脚与普通 I/O 引脚共用，当比较器功能未使用时，此引脚可做普通引脚使用而不浪费 I/O 资源。除此之外，单片机还提供了比较器失调电压校准功能。

电路包含两个比较器功能，用于比较两个模拟电压，基于它们的差值上提供一个输出。比较器的输出可由各自寄存器的一位记录，并且在共用的 I/O 口上输出。此外，比较器功能还包括暂停控制。

### 1) CMP1 控制寄存器

地址：0xDF

复位值：0x10

访问：R/W

Bit	Name	R/W	Description
[7]	CMP1X	R	比较器输出脚；正逻辑电平。此位为只读位
[6]	C1OFM	R/W	输入失调电压校准模式/比较器模式选择位 0：比较器模式 1：输入失调电压校准模式
[5]	C1RS	R/W	比较器输入失调电压校准参考选择位 0：选择 C1N 作为参考输入脚 1：选择 CNP 作为参考输入脚
[4: 0]	C1OF[4: 0]	R/W	比较器输入失调电压校准控制位

### 2) CMP2 控制寄存器

地址：0xE5

复位值：0x10

访问：Read/Write

Bit	Name	R/W	Description
[7]	CMP2X	R	比较器输出脚；正逻辑电平。此为只读位
[6]	C2OFM	R/W	输入失调电压校准模式/比较器模式选择位 0：比较器模式 1：输入失调电压校准模式
[5]	C2RS	R/W	比较器输入失调电压校准参考选择位 0：选择 CNP 作为参考输入端 1：选择 C2P 作为参考输入端
[4: 0]	C2OF[4: 0]	R/W	比较器输入失调电压校准控制位

### 3) CMPSEL 控制寄存器

地址：0xE6

复位值：0x4C

访问：R/W

Bit	Name	R/W	Description
[7]			保留，读为“0”
[6]	CNPSEL	R/W	比较器同相输入控制位 0: CNP 脚作为比较器输入关闭 1: CNP 脚作为比较器输入使能
[5]	C2OUTEN	R/W	比较器 2 输出控制位 0: 关闭 1: 使能
[4]	C1OUTEN	R/W	比较器 1 输出控制位 0: 关闭 1: 使能
[3]	C2PSEL	R/W	比较器 2 同相输入选择位 0: C2P 脚作为输入端关闭 1: C2P 脚作为输入端打开
[2]	C1NSEL	R/W	比较器 1 反相输入端选择位 0: C1N 脚作为输入端关闭 1: C1N 脚作为输入端打开
[1]	CMP2EN	R/W	CMP2EN: 比较器 2 使能/关闭控制位 0: 关闭 1: 使能
[0]	CMP1EN	R/W	CMP1EN: 比较器 1 使能/关闭控制位 0: 关闭 1: 使能

## 4) CMPIFG 控制寄存器

地址：0xE7

复位值：0x00

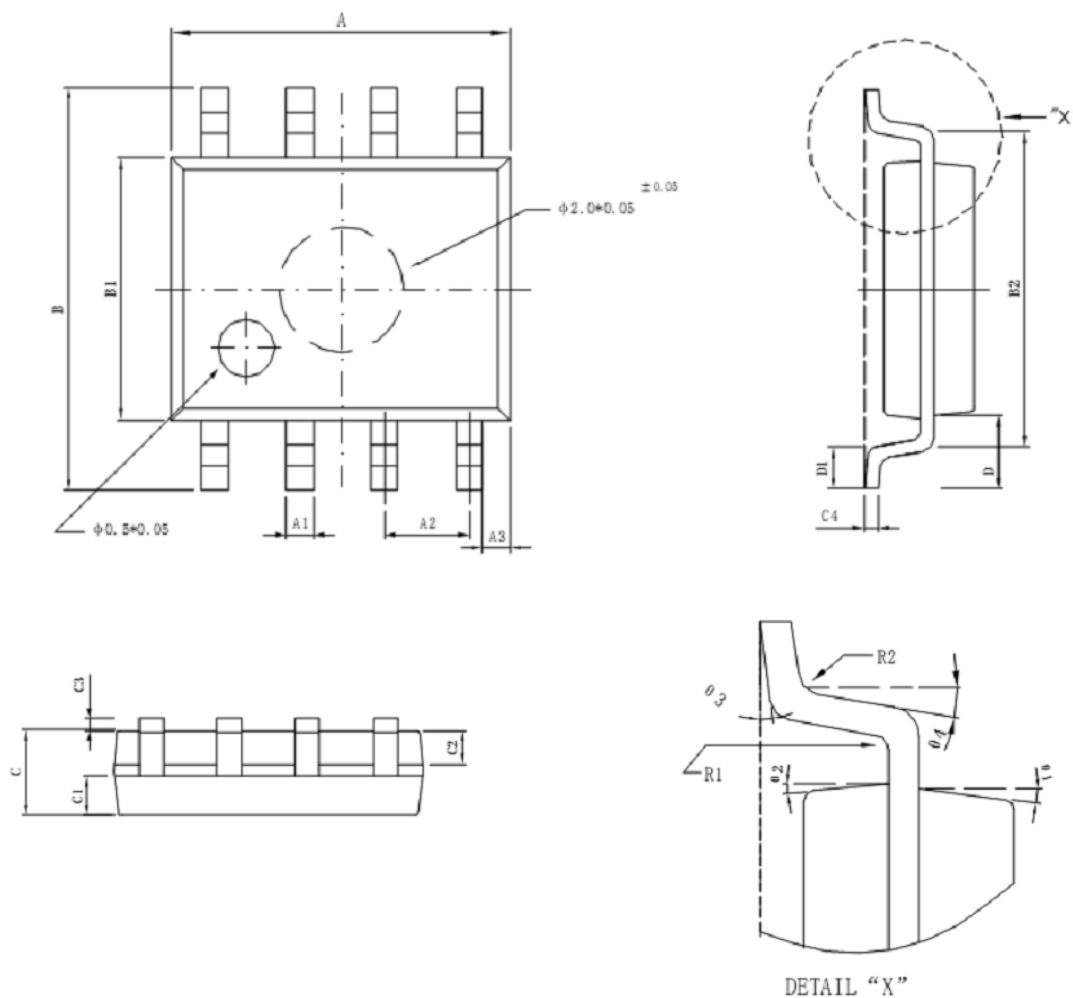
访问：R/W

Bit	Name	R/W	Description
[7]	CMP2_IFG	R/W	比较器 2 中断标志位 1'b0: 无效标志位 1'b1: 有效标志位
[6]	CMP1_IFG	R/W	比较器 1 中断标志位 1'b0: 无效标志位 1'b1: 有效标志位
[5: 4]	CMPEs[1: 0]	R/W	中断边沿控制位 00: 关闭 01: 上升沿触发 10: 下降沿触发 11: 双沿触发
[3]	C2INTEN	R/W	比较器 2 中断控制位 0: 关闭 1: 使能
[2]	C1INTEN	R/W	比较器 1 中断控制位 0: 关闭 1: 使能
[1]	CMP2OF_SEL	R/W	比较器 2 失调电压选择 0: 选择存储在 MTP 的比较器 2 的失调电压值 1: 选择 MCU 写入的比较器 2 的失调电压值
[0]	CMP1OF_SEL	R/W	比较器 1 失调电压选择 0: 选择存储在 MTP 的比较器 1 的失调电压值 1: 选择 MCU 写入的比较器 1 的失调电压值

## 5、封装尺寸与外形图（单位：mm）

### 5.1、SOP8（CS88F313AEO）

#### 5.1.1、外形图

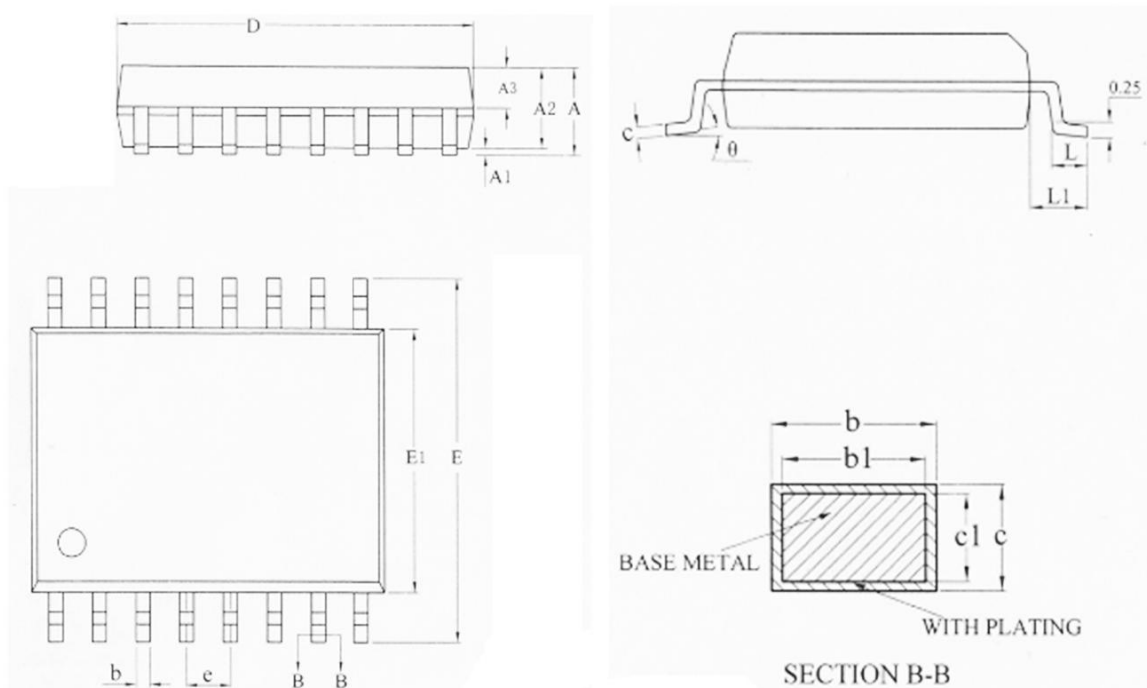


#### 5.1.2、封装尺寸

Symbol	Min.	Max.	Symbol	Min.	Max.
A	4.95	5.15	C3	0.10	0.20
A1	0.37	0.47	C4	0.20 TYP	
A2	1.27 TYP		D	1.05 TYP	
A3	0.41 TYP		D1	0.50 TYP	
B	5.80	6.20	R1	0.07 TYP	
B1	3.80	4.00	R2	0.07 TYP	
B2	5.0 TYP		$\theta 1$	17° TYP	
C	1.30	1.50	$\theta 2$	13° TYP	
C1	0.55	0.65	$\theta 3$	4° TYP	
C2	0.55	0.65	$\theta 4$	12° TYP	

## 5.2、SOP16 (CS88F313BEO)

### 5.2.1、外形图



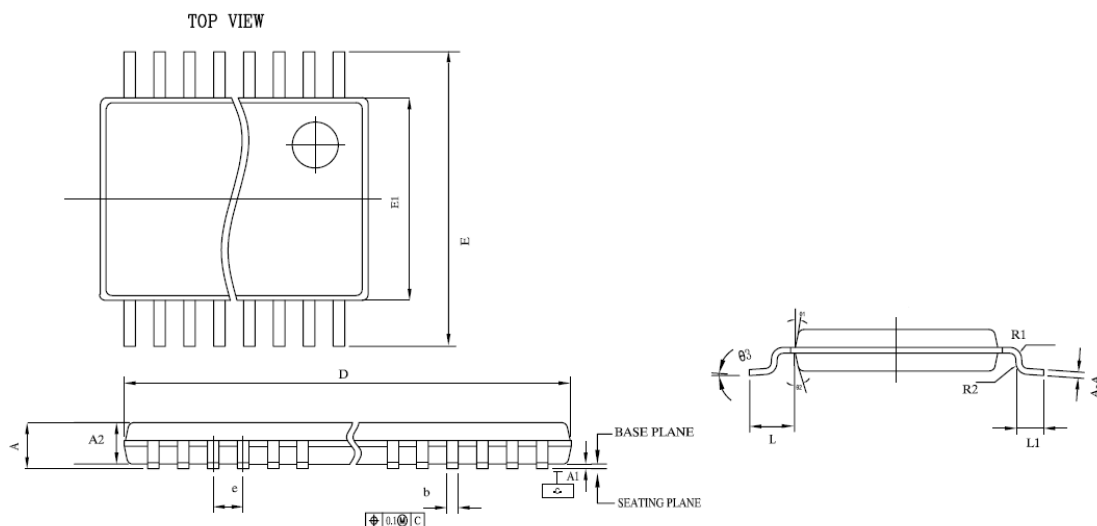
### 5.2.2、封装尺寸

Symbol	Min	Nom	Max	Symbol	Min	Nom	Max
A			1.75	D	9.70	9.90	10.10
A1	0.10		0.225	E	5.80	6.00	6.20
A2	1.30	1.40	1.50	E1	3.70	3.90	4.10
A3	0.60	0.65	0.70	e	1.27BSC		
b	0.39		0.48	h	0.25		0.50
b1	0.38	0.41	0.43	L	0.50		0.80
c	0.21		0.26	L1	1.05BSC		
c1	0.19	0.20	0.21	θ	0°		8°



### 5.3、TSSOP20（CS88F313CEO）

#### 5.3.1、外形图

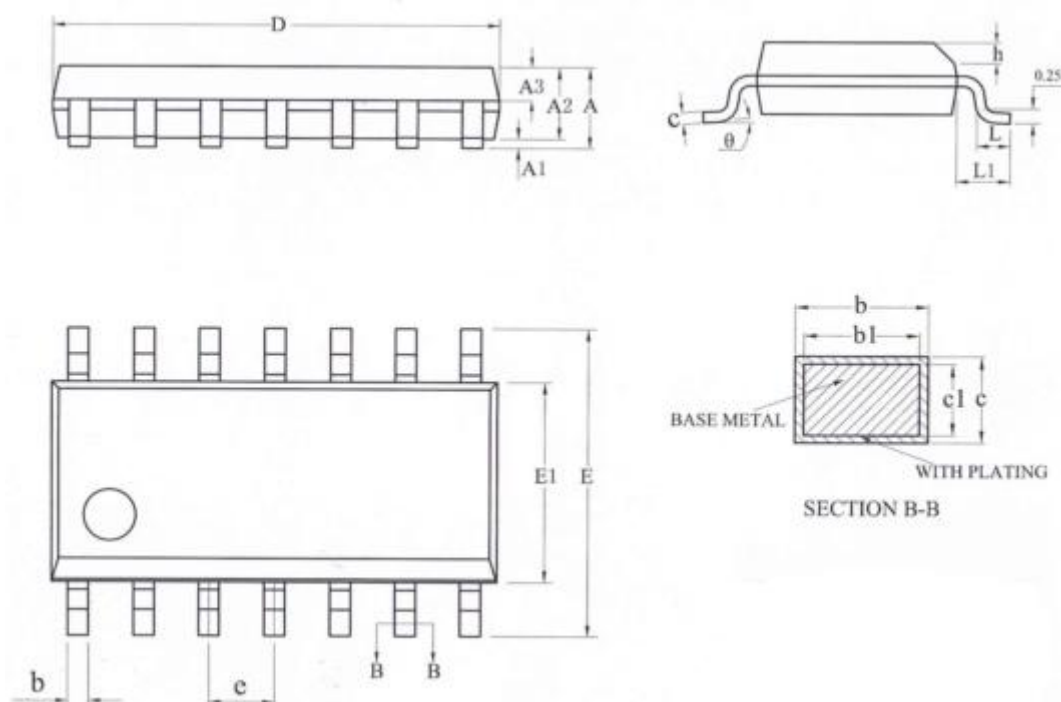


#### 5.3.2、封装尺寸

Symbol	Min	Nom	Max	Symbol	Min	Nom	Max
A			1.2	D	6.4		6.6
A1	0.05		0.15	E	6.25		6.55
A2	0.8		1.05	E1	4.3		4.5
b	0.19		0.3	e	0.65		
R1		0.15		L			1
R2		0.15		L1	0.45		0.75
θ1		12°		θ3	0°		8°
θ2		12°		A-A	0.19		0.3
c1	0.14	0.15	0.16				

## 5.4、SOP14L (CS88F313DEO)

### 5.4.1、封装外形图



### 5.4.2、封装尺寸

Symbol	Min	Nom	Max	Symbol	Min	Nom	Max
A			1.75	D	8.55	8.65	8.75
A1	0.00		0.225	E	5.80	6.00	6.20
A2	1.30	1.40	1.50	E1	3.80	3.90	4.00
A3	0.60	0.65	0.70	e	1.27BSC		
b	0.39		0.47	h	0.25		0.50
b1	0.38	0.41	0.43	L	0.50		0.80
c	0.20		0.24	L1	1.05BSC		
c1	0.19	0.20	0.21	θ	0		8°

## 6、产品中有毒有害物质或元素说明

产品中有毒有害物质或元素的名称及含量

部件名称	有毒有害物质或元素					
	铅 (Pb)	汞 (Hg)	镉 (Cd)	六价铬 (Cr <sup>+6</sup> )	多溴联苯 (PBB)	多溴联苯醚 (PBDE)
引线框	○	○	○	○	○	○
塑封树脂	○	○	○	○	○	○
芯片	○	○	○	○	○	○
内引线	○	○	○	○	○	○
装片胶	○	○	○	○	○	○
说明	○：表示该有毒有害物质的含量在 GBT26572-2011 标准的限量要求以下。×：表示该有毒有害物质的含量超出 GBT26572-2011 标准的限量要求。					

## 华润微集成电路（无锡）有限公司

### CRM ICBG (wuxi) Co., Ltd.

总部地址：江苏省无锡市菱湖大道 180-6 电话：0510-85810118

上海分公司地址：上海市静安区市北智汇园汶水路 299 弄 12 号 电话：021-60738989

深圳分公司地址：广东省深圳市福田区车公庙天安数码城天祥大厦 8 楼 A 座 电话：0755-83572722

### 公司销售联络点：

华东办事处：

江苏省无锡市菱湖大道 180-6 电话：0510-85810118

深圳办事处：

深圳市宝安区西乡街道兴业路老兵衡芳工业城西座 5 楼 5006 室 电话：0755-83572766

广州办事处：

广东省中山市古镇镇同兴路 98 号利和商业中心 3507

台湾办事处：

广东省深圳市福田区车公庙天安数码城天祥大厦 8 楼 A 座 电话：0755-83572722

### 注意：

建议您在使用华润微产品之前仔细阅读本资料。希望您经常和华润微有关部门进行联系，索取最新资料，因为华润微产品在不断更新和提高。本资料中的信息如有变化，恕不另行通知。

本资料仅供参考，华润微不承担任何由此而引起的损失。华润微不承担任何在使用过程中引起的侵犯第三方专利或其它权利的责任。

华润微集成电路(无锡)有限公司有权对所提供的产品和服务进行更正、修改、增强、改进或其它更改，并有权中止提供任何产品和服务。客户在下订单前应获取最新的相关信息，并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的华润微集成电路(无锡)有限公司销售条款与条件。

华润微集成电路(无锡)有限公司保证其所销售的产品的性能符合产品销售时半导体产品销售条件与条款的适用规范。仅在华润微集成电路(无锡)有限公司保证的范围内，且华润微集成电路(无锡)有限公司认为有必要时才会使用测试或其它质量控制技术。除非适用法律做出了硬性规定，否则没有必要对每种产品的所有参数进行测试。

华润微集成电路(无锡)有限公司对应用帮助或客户产品设计不承担任何义务。客户应对其使用华润微集成电路(无锡)有限公司的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险，客户应提供充分的设计与操作安全措施。

华润微集成电路(无锡)有限公司产品未获得用于 FDA Class III（或类似的生命攸关医疗设备）的授权许可，除非各方授权官员已经达成了专门管控此类使用的特别协议。

只有那些华润微集成电路(无锡)有限公司特别注明属于军用等级或“增强型塑料”的华润微集成电路(无锡)有限公司产品才是设计或专门用于军事/航空应用或环境的。购买者认可并同意，对并非指定面向军事或航空航天用途的华润微集成电路(无锡)有限公司产品进行军事或航空航天方面的应用，其风险由客户单独承担，并且由客户独力负责满足与此类使用相关的所有法律和法规要求。

华润微集成电路(无锡)有限公司未明确指定符合 ISO/TS16949 要求的产品不能应用于汽车。在任何情况下，因使用非指定产品而无法达到 ISO/TS16949 要求，华润微集成电路(无锡)有限公司不承担任何责任。