

晟矽微电 8 位单片机

MC32F7072

用户手册

V1.1





目录

1	产品概要	5
1.1	产品特性	5
1.2	订购信息	6
1.3	引脚排列	7
1.4	端口说明	7
2	电气特性	9
2.1	极限参数	9
2.2	直流电气特性	9
2.3	交流电气特性	10
2.4	ADC 特性参数	10
2.5	CMP 特性参数	11
2.6	OPA 特性参数	11
2.7	EEPROM 特性参数	12
3	CPU 与存储器	13
3.1	指令集	13
3.2	程序存储器	15
3.3	数据存储器	16
3.4	堆栈	17
3.5	控制寄存器	17
3.6	用户配置字	20
4	系统时钟	22
4.1	内部高频 RC 振荡器	22
4.2	内部低频 RC 振荡器	22
4.3	系统工作模式	23
4.4	低功耗模式	24
5	复位	26
5.1	复位条件	26
5.2	上电复位	27
5.3	外部复位	27
5.4	低电压复位	27
5.5	看门狗复位	27
6	I/O 端口	28
6.1	通用 I/O 功能	28
6.2	内部上/下拉电阻	29
6.3	端口模式控制	30
7	定时器 TIMER	32
7.1	看门狗定时器 WDT	32
7.2	定时器 T0	32
7.3	定时器 T1	38
7.4	定时器 T2	39
7.5	定时器 T3	42



8	模数转换器 ADC.....	45
8.1	ADC 概述.....	45
8.2	ADC 相关寄存器.....	46
8.3	ADC 操作步骤.....	49
8.4	ADC 零点偏移修调流程.....	50
9	模拟比较器 CMP.....	51
9.1	CMP 概述.....	51
9.2	CMP 相关寄存器.....	52
9.3	CMP 失调电压调校流程.....	59
10	运算放大器 OPA	60
10.1	OPA 概述	60
10.2	OPA 相关寄存器.....	60
10.3	OPA 失调电压调校流程.....	61
11	低电压检测 LVD	63
12	总线通讯 IIC.....	64
12.1	IIC 概述.....	64
12.2	IIC 数据传输.....	64
12.3	IIC 工作模式.....	65
12.4	IIC 相关寄存器.....	65
12.5	IIC 应用流程.....	66
13	EEPROM 存储器.....	68
13.1	EEPROM 概述.....	68
13.2	EEPROM 相关寄存器.....	68
13.3	EEPROM 操作示例.....	70
14	FLASH 烧录编程	71
14.1	FLASH 在板编程	71
15	中断.....	73
15.1	外部中断.....	73
15.2	定时器中断.....	73
15.3	T2 捕捉中断.....	73
15.4	ADC 中断.....	74
15.5	CPO 触发中断.....	74
15.6	比较器中断.....	74
15.7	LVD 中断.....	74
15.8	IIC 中断.....	74
15.9	中断相关寄存器.....	75
16	特性曲线.....	79
16.1	I/O 特性.....	79
16.2	功耗特性.....	82
16.3	模拟电路特性.....	87
17	封装尺寸.....	91
17.1	SOP20.....	91
17.2	SOP16.....	91
18	修订记录.....	92



Sinomcu.com



1 产品概要

1.1 产品特性

- 8 位 CPU 内核
 - ◇ 精简指令集，8 级深度硬件堆栈
 - ◇ CPU 为双时钟，可在系统高/低频时钟之间切换
 - ◇ 系统高频时钟下 F_{CPU} 可配置为 F_{HOSC} 的 2/4/8/16/32/64 分频
 - ◇ 系统低频时钟下 F_{CPU} 固定为 F_{LOSC} 的 2 分频
- 程序存储器
 - ◇ 4K×16 位 FLASH 型程序存储器，可通过间接寻址读取程序存储器内容
 - ◇ 支持在板带电烧录编程，擦写次数至少 1000 次
- 数据存储器
 - ◇ 256 字节 SRAM 型通用数据存储器，支持直接寻址、间接寻址等多种寻址方式
 - ◇ 128×16 位 EEPROM 型数据存储器，支持单独烧录和软件读写，擦写次数至少 10000 次
- 3 组共 18 个 I/O
 - ◇ P0 (P00~P07), P1 (P10~P17), P2 (P20~P21)
 - ◇ 所有端口均支持施密特输入，均支持推挽输出
 - ◇ P20 可复用为外部复位 RST 输入，P00/P02 复用为 SCL/SDA 时为开漏输出
 - ◇ 所有端口均内置上拉和下拉电阻，均可单独使能
 - ◇ P0 输出电流 2 级可配置，P1 和 P2 所有端口均为大电流端口
 - ◇ P15/P20 可复用为外部中断输入，支持外部中断唤醒功能
- 系统时钟源
 - ◇ 内置高频 RC 振荡器 (32MHz/16MHz)，可用作系统高频时钟源
 - ◇ 内置低频 RC 振荡器 (32KHz)，可用作系统低频时钟源
- 系统工作模式
 - ◇ 高速模式：CPU 在高频时钟下运行，低频时钟源工作
 - ◇ 低速模式：CPU 在低频时钟下运行，高频时钟源可选停止或工作
 - ◇ HOLD1 模式（低功耗模式）：CPU 暂停，高频时钟源工作，低频时钟源可选停止或工作
 - ◇ HOLD2 模式（低功耗模式）：CPU 暂停，高频时钟源停止，低频时钟源工作
 - ◇ 休眠模式（低功耗模式）：CPU 暂停，高/低频时钟源均停止
- 内部自振式看门狗计数器 (WDT)
 - ◇ 溢出时间可配置：64ms/2048ms
 - ◇ 工作模式可配置：始终开启、始终关闭、低功耗模式下关闭
- 4 个定时器
 - ◇ 8 位定时器 T0，可实现外部计数、1 对 8+3 模式的带死区互补 PWM
 - ◇ 8 位定时器 T1，可实现比较器 CP0 输出信号 CP0_OUT 的下降沿计数功能
 - ◇ 8 位定时器 T2，可实现内/外部计数、高/低电平脉宽测量和脉冲周期宽度测量等功能
 - ◇ 8 位定时器 T3
- 1 个 12 位高精度 SAR 型 ADC
 - ◇ 14 路外部通道：AN0~AN13；2 路内部通道：GND、VDD/4
 - ◇ 参考电压可选：VDD、内部参考电压 V_{IR} (2V/3V/4V)、外部参考电压 V_{ER} (VERI 输入)



- ◇ ADC 时钟：F_{HIRC} 的 8/16/32/64/128/256/512/1024 分频
- ◇ 支持零点校准
- 5 个模拟比较器 CP0~CP4
 - ◇ 输入共模 0 ~ (VDD-1.4V)，支持失调电压自消除调校（调校精度±2mV），输出无回滞
 - ◇ CP0，正/负端输入均为外部输入电压，输出信号 CP0_OUT 支持去抖处理
 - ◇ CP1~CP4，负端输入为外部输入电压，正端输入为 VDD/V_{IR} 的内部分压电压（分压精度 1%），输出均支持去抖处理
 - ◇ CP1，正端 16 级分压电压：(0.34~0.64) × VDD、或 (0.0625~0.875) × V_{IR}
 - ◇ CP2，正端 8 级分压电压：(0.05~0.70) × VDD、或 (0.425~0.8) × V_{IR}
 - ◇ CP3，正端 32 级分压电压：(0.06~0.70) × VDD、或 (0.075~0.875) × V_{IR}
 - ◇ CP4，正端 8 级分压电压：(0.02~0.50) × VDD、或 (0.425~0.8) × V_{IR}
- 1 个运算放大器 OPA
 - ◇ 输入共模 0 ~ (VDD-1.4V)，支持失调电压自消除调校（调校精度±2mV）
 - ◇ 开环放大倍数 60dB
 - ◇ 内置组合电路，输出信号可作为 ADC 输入或比较器 CP3 负端输入
- 1 组总线通讯 IIC 从机接口
 - ◇ 支持 7 位地址编码的从机模式
 - ◇ 通讯速率最高支持 400Kbps
 - ◇ 地址匹配、接收完成、发送完成等事件发生时，可触发中断
- 中断
 - ◇ 外部中断（INT0~INT1），CP0 触发中断
 - ◇ 定时器中断（T0~T3），T2 捕捉中断
 - ◇ ADC 中断，比较器中断（CP1~CP4），LVD 中断
 - ◇ IIC 中断
- 低电压检测 LVD
 - ◇ 3.3V/4.2V
- 低电压复位 LVR
 - ◇ 2.0V/2.3V/2.7V/3.3V
- 工作电压
 - ◇ VLVR33 ~ 5.5V @ F_{cpu} = 0~16MHz
 - ◇ VLVR27 ~ 5.5V @ F_{cpu} = 0~8MHz
 - ◇ VLVR20 ~ 5.5V @ F_{cpu} = 0~4MHz
- 封装形式
 - ◇ SOP20/SOP16

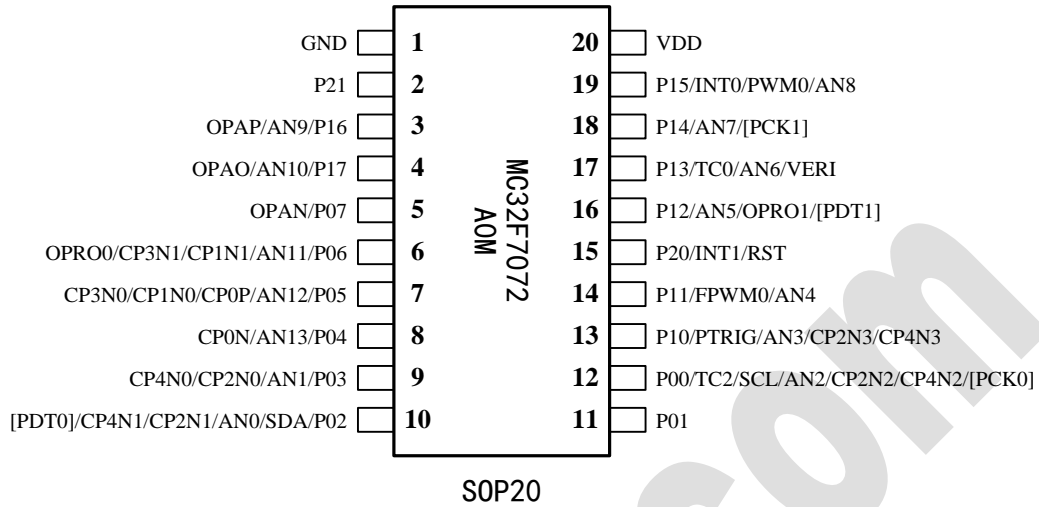
1.2 订购信息

产品名称	封装形式	备注
MC32F7072A0M	SOP20	
MC32F7072A0K	SOP16	

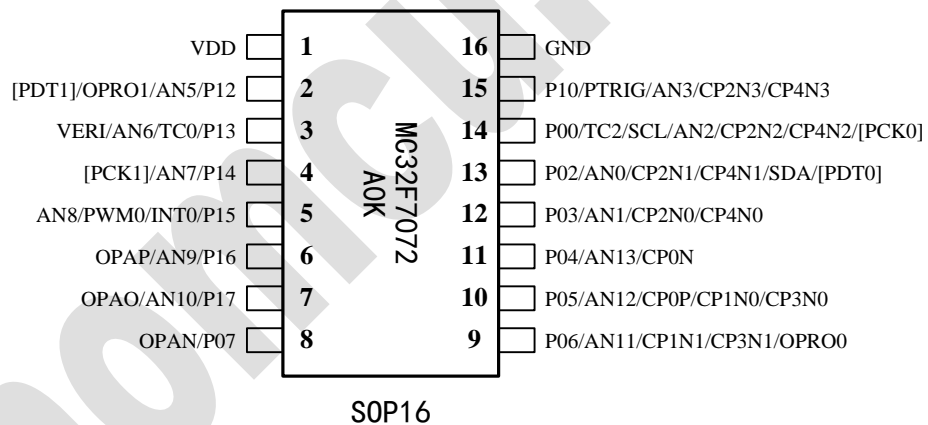


1.3 引脚排列

MC32F7072A0M



MC32F7072A0K



1.4 端口说明

端口名称	类型	功能说明
VDD	P	电源
GND	P	地
P0, P1, P2	D	GPIO (推挽输出), 内部上/下拉
INT0~INT1	DI	外部中断输入
TC0, TC2	DI	定时器 T0/T2 的外部计数输入
PWM0, FPWM0	DO	定时器 T0 的 PWM 及其互补输出
PTRIG	DI	外部触发输入



AN0~AN13	AI	ADC 外部输入通道
VERI	AI	ADC 外部参考电压输入
CP0P, CP0N	AI	CP0 正/负端外部输入
CP1N0~CP1N1	AI	CP1 负端外部输入通道
CP2N0~CP2N3	AI	CP2 负端外部输入通道
CP3N0~CP3N1	AI	CP3 负端外部输入通道
CP4N0~CP4N3	AI	CP4 负端外部输入通道
OPAP, OPAN	AI	OPA 正/负端外部输入
OPAO	AO	OPA 输出
OPRO0~OPRO1	AO	OPA 带电阻输出通道
SCL, SDA	D	IIC 通讯时钟/数据端口, 开漏输出
RST	DI	外部复位输入
PCK0/PDT0, PCK1/PDT1	D	编程时钟/数据接口

注: P-电源端口; D-数字端口, DI-数字输入, DO-数字输出; A-模拟端口, AI-模拟输入, AO-模拟输出。



2 电气特性

2.1 极限参数

参数	符号	值	单位
电源电压	VDD	-0.3~6.0	V
I/O 输入电压	V _{in}	-0.3~VDD+0.3	V
工作温度	T _a	-40~85	°C
储存温度	T _{stg}	-65~150	°C
流入 VDD 最大电流	IVDDmax	50	mA
流出 GND 最大电流	IGNDmax	50	mA

注：若芯片工作条件超过极限值，则会造成永久性损坏；若芯片长时间工作在极限条件下，则将影响其可靠性。

2.2 直流电气特性

VDD=5V, T=25°C

特性	符号	端口	条件	最小	典型	最大	单位
工作电压	VDD	VDD	F _{cpu} =16MHz@F _{HIRC} (32M)/2	V _{LVR33}		5.5	V
			F _{cpu} =8MHz@F _{HIRC} (32M)/4	V _{LVR27}		5.5	
			F _{cpu} =8MHz@F _{HIRC} (16M)/2	V _{LVR27}		5.5	
			F _{cpu} =4MHz@F _{HIRC} (16M)/4	V _{LVR20}		5.5	
			F _{cpu} =16KHz@F _{LIRC} (32K)/2	V _{LVR20}		5.5	
输入漏电流	I _{leak}	所有输入脚	VDD=5V	-1		1	μA
输入高电平	V _{ih}	所有输入脚		0.8VDD			V
输入低电平	V _{il}	所有输入脚				0.2VDD	V
上拉电阻	R _{pu}	P0, P1, P2	VDD=5V, V _{in} =0		30		KΩ
下拉电阻	R _{pd}	P0, P1, P2	V _{in} =VDD=5V		30		KΩ
输出源电流	I _{oh}	P0	V _{oh} =0.9VDD, P0DRVS 配置		10		mA
			V _{oh} =0.9VDD, P0DRVS 配置		20		mA
		P1, P2	V _{oh} =0.9VDD		20		mA
输出灌电流	I _{ol}	P0	V _{ol} =0.1VDD, P0DRVS 配置		15		mA
			V _{ol} =0.1VDD, P0DRVS 配置		30		mA
		P1, P2	V _{ol} =0.1VDD		30		mA
运行模式功耗	I _{run}	VDD	F _{cpu} =16MHz@HIRC(32M)		4.0		mA
			F _{cpu} =8MHz@HIRC(32M)		2.5		mA
			F _{cpu} =8MHz@HIRC(16M)		2.3		mA



			Fcpu=4MHz@HIRC(16M)		1.6		mA
			Fcpu=2MHz@HIRC(16M)		1.3		mA
			Fcpu=1MHz@HIRC(16M)		1.0		mA
			Fcpu=16KHz@LIRC(32K)		60		μA
HOLD1 功耗	Ihold1	VDD	CPU 停, HIRC(32M)/LIRC 开		1200		μA
			CPU 停, HIRC(16M)/LIRC 开		750		μA
HOLD2 功耗	Ihold2	VDD	CPU 停, HIRC 关, LIRC 开		2.5	5	μA
休眠模式功耗	Istop	VDD	休眠模式, WDT/LVR 关		0.1	1	μA
			休眠模式, WDT 开, LVR 关		2.5	5	μA
			休眠模式, WDT 关, LVR 开		8	20	μA
低压检测电压	VLVD	VDD	LVDVS 选择	-5%		+5%	V
LVD 响应时间	TLVD			1	50	200	μs
低压复位电压	VLVR	VDD	LVRVS 配置	-10%		+10%	V
LVD/LVR 回滞电压		VDD			6%	12%	

注：条件项中，无关模块默认关闭，无关端口设为低电平无负载输出或内部上/下拉电阻无效且外接 GND 的输入。

2.3 交流电气特性

特性	符号	条件	最小	典型	最大	单位
HIRC 振荡频率	FHIRC	VDD=2.0V~5.5V, T=25°C	-2%	16/32	+2%	MHz
		VDD=2.0V~5.5V, T=-40°C~85°C	-4%		+4%	
LIRC 振荡频率	FLIRC	VDD=5V, T=25°C	-50%	32	+50%	KHz

2.4 ADC 特性参数

VDD=5V, T=25°C

特性	符号	条件	最小	典型	最大	单位
ADC 有效工作电压	V _{ADC}	T=-40°C~85°C	2.7		5.5	V
积分非线性误差	INL	V _{REF} =VDD, F _{ADC} =1MHz, T _{con} =27μs			±4	LSB
微分非线性误差	DNL	V _{REF} =VDD, F _{ADC} =1MHz, T _{con} =27μs			±2	LSB
零点偏移误差	EZ	V _{REF} =VDD, F _{ADC} =1MHz, T _{con} =27μs			±4	LSB
增益误差	ET	V _{REF} =VDD, F _{ADC} =1MHz, T _{con} =27μs			±4	LSB
转换时钟	F _{ADC}	VDD=5V			2	MHz
转换时间	T _{con}	F _{ADC} =1MHz	16		27	1/F _{ADC}



ADC 输入电压	V _{AIN}		GND		V _{REF}	V
ADC 输入阻抗	R _{AIN}		2			MΩ
ADC 输入电流	I _{AIN}				2	μA
ADC 动态电流	I _{ADD}	VDD=5V, AD 转换中		1	3	mA
ADC 静态电流	I _{ADS}	VDD=5V, ADC 关闭		0.1	1	μA
模拟信号源推荐阻抗	Z _{AIN}				10	KΩ
ADC 参考电压	V _{REF}	选择 VDD		VDD		V
		选择内部参考电压 V _{IR} , T=25°C	-1%	2/3/4	+1%	
		选择内部参考电压 V _{IR} , T=-40°C~85°C	-3%		+3%	
		选择外部参考电压 V _{ER}	2	VDD		
V _{IR} 有效工作电压	V _{VIR}	选择内部参考电压 V _{IR}	V _{IR} +0.5		5.5	V

2.5 CMP 特性参数

VDD=5V, T=25°C

特性	符号	条件	最小	典型	最大	单位
比较器有效工作电压	V _{CMP}	T=-40°C~85°C	2.7		5.5	V
工作电流	I _{CMP}			200		μA
输入失调电压	V _{offset}	未调校	-15		+15	mV
		已调校	-2		+2	
输入共模电压	V _{com}		0		VDD-1.4	V
响应时间	T _{RESP}				2	μs
比较器参考电压	V _{CPR}	选择 VDD		VDD		V
		选择内部参考电压 V _{IR} , T=25°C	-2%	2/3/4	+2%	
		选择内部参考电压 V _{IR} , T=-40°C~85°C	-4%		+4%	
电阻分压比值			-1%		+1%	

2.6 OPA 特性参数

VDD=5V, T=25°C

特性	符号	条件	最小	典型	最大	单位
OPA 有效工作电压	V _{OPA}	T=-40°C~85°C	2.7		5.5	V
输入失调电压	V _{offset}	未调校	-15		+15	mV
		已调校	-2		+2	
输入共模电压	V _{com}		0		VDD-1.4	V



输出电压摆幅	V _{ovs}		0	0.2	0.3	V
			4.7	4.8	5.0	
电压摆率	SR		0.1	0.2		V/ μ s
开环增益	A _{VOL}		60	80		dB
电源抑制比	PSRR		60	80		dB
共模抑制比	CMRR		60	80		dB
正端输入接地电阻	R _{OPAP}			6		K Ω
输出端内部电阻	R _{OPA0}			60		K Ω

2.7 EEPROM 特性参数

特性	符号	条件	最小	典型	最大	单位
EEPROM 读操作电压	V _{EErd}	T=-40°C~85°C	2.0		5.5	V
EEPROM 写操作电压	V _{EEwr}	T=-40°C~85°C	2.0		5.5	V
EEPROM 写操作电流	I _{EEwr}	T=-40°C~85°C		2		mA
EEPROM 单地址写入时间	T _{EEwr}	V _{DD} =2.0V~5.5V, T=-40°C~85°C		5		ms
EEPROM 擦写次数		V _{DD} =5V, T=25°C	10000			cycle
EEPROM 数据保持时间			10			year



3 CPU 与存储器

3.1 指令集

芯片的指令集为精简指令集。

除程序跳转类指令外，其他指令均为单周期指令，即执行时间为 1 个指令周期（CPU 时钟周期）；所有指令均为单字指令，即指令码仅占用 1 个程序存储器地址空间。

指令汇总表

助记符	说明	操作	周期	长度	标志
ADDAR R	R 和 A 相加，结果存入 A	$R+A \rightarrow A$	1	1	C, DC, Z
ADDRA R	R 和 A 相加，结果存入 R	$R+A \rightarrow R$	1	1	C, DC, Z
ADCAR R	R 和 A 相加（带 C 标志），结果存入 A	$R+A+C \rightarrow A$	1	1	C, DC, Z
ADCRA R	R 和 A 相加（带 C 标志），结果存入 R	$R+A+C \rightarrow R$	1	1	C, DC, Z
RSUBAR R	R 和 A 相减，结果存入 A	$R-A \rightarrow A$	1	1	C, DC, Z
RSUBRA R	R 和 A 相减，结果存入 R	$R-A \rightarrow R$	1	1	C, DC, Z
RSBCAR R	R 和 A 相减（带 C 标志），结果存入 A	$R-A-C \rightarrow A$	1	1	C, DC, Z
RSBCRA R	R 和 A 相减（带 C 标志），结果存入 R	$R-A-C \rightarrow R$	1	1	C, DC, Z
ASUBAR R	A 和 R 相减，结果存入 A	$A-R \rightarrow A$	1	1	C, DC, Z
ASUBRA R	A 和 R 相减，结果存入 R	$A-R \rightarrow R$	1	1	C, DC, Z
ASBCAR R	A 和 R 相减（带 C 标志），结果存入 A	$A-R-C \rightarrow A$	1	1	C, DC, Z
ASBCRA R	A 和 R 相减（带 C 标志），结果存入 R	$A-R-C \rightarrow R$	1	1	C, DC, Z
ANDAR R	R 和 A 与操作，结果存入 A	$R \text{ and } A \rightarrow A$	1	1	Z
ANDRA R	R 和 A 与操作，结果存入 R	$R \text{ and } A \rightarrow R$	1	1	Z
ORAR R	R 和 A 或操作，结果存入 A	$R \text{ or } A \rightarrow A$	1	1	Z
ORRA R	R 和 A 或操作，结果存入 R	$R \text{ or } A \rightarrow R$	1	1	Z
XORAR R	R 和 A 异或操作，结果存入 A	$R \text{ xor } A \rightarrow A$	1	1	Z
XORRA R	R 和 A 异或操作，结果存入 R	$R \text{ xor } A \rightarrow R$	1	1	Z
COMAR R	对 R 取反，结果存入 A	$R \text{ 取反} \rightarrow A$	1	1	Z
COMR R	对 R 取反，结果存入 R	$R \text{ 取反} \rightarrow R$	1	1	Z
RLA	A 循环左移（带 C 标志）	$A[7] \rightarrow C; A[6:0] \rightarrow A[7:1]; C \rightarrow A[0]$	1	1	C
RLAR R	R 循环左移（带 C 标志），结果存入 A	$R[7] \rightarrow C; R[6:0] \rightarrow A[7:1]; C \rightarrow A[0]$	1	1	C
RLR R	R 循环左移（带 C 标志），结果存入 R	$R[7] \rightarrow C; R[6:0] \rightarrow R[7:1]; C \rightarrow R[0]$	1	1	C
RRA	A 循环右移（带 C 标志）	$A[0] \rightarrow C; A[7:1] \rightarrow A[6:0]; C \rightarrow A[7]$	1	1	C
RRAR R	R 循环右移（带 C 标志），结果存入 A	$R[0] \rightarrow C; R[7:1] \rightarrow A[6:0]; C \rightarrow A[7]$	1	1	C
RRR R	R 循环右移（带 C 标志），结果存入 R	$R[0] \rightarrow C; R[7:1] \rightarrow R[6:0]; C \rightarrow R[7]$	1	1	C
SWAPAR R	交换 R 的高低半字节，结果存入 A	$R[7:4] \rightarrow A[3:0]; R[3:0] \rightarrow A[7:4]$	1	1	-
SWAPR R	交换 R 的高低半字节，结果存入 R	$R[7:4] \rightarrow R[3:0]; R[3:0] \rightarrow R[7:4]$	1	1	-



MOVRA	R	将 A 存入 R	A→R	1	1	-
MOVAR	R	将 R 存入 A	R→A	1	1	Z
MOV R	R	将 R 存入 R	R→R	1	1	Z
CLRA		将 A 清零	0→A	1	1	Z
CLRR	R	将 R 清零	0→R	1	1	Z
INCA		A 自加 1	A+1→A	1	1	-
INCR	R	R 自加 1	R+1→R	1	1	Z
INCAR	R	R 加 1, 结果存入 A	R+1→A	1	1	Z
DECA		A 自减 1	A-1→A	1	1	-
DECR	R	R 自减 1	R-1→R	1	1	Z
DECAR	R	R 减 1, 结果存入 A	R-1→A	1	1	Z
JZA		A 自加 1: 结果为 0 则跳过下一条指令	A+1→A: 结果为 0 则 PC+2→PC	1/2	1	-
JZR	R	R 自加 1: 结果为 0 则跳过下一条指令	R+1→R: 结果为 0 则 PC+2→PC	1/2	1	-
JZAR	R	R 加 1, 结果存入 A: 结果为 0 则跳过下一条指令	R+1→A: 结果为 0 则 PC+2→PC	1/2	1	-
DJZA		A 自减 1: 结果为 0 则跳过下一条指令	A-1→A: 结果为 0 则 PC+2→PC	1/2	1	-
DJZR	R	R 自减 1: 结果为 0 则跳过下一条指令	R-1→R: 结果为 0 则 PC+2→PC	1/2	1	-
DJZAR	R	R 减 1, 结果存入 A: 结果为 0 则跳过下一条指令	R-1→A: 结果为 0 则 PC+2→PC	1/2	1	-
BCLR	R, b	将 R 的第 b 位清 0	0→R[b]	1	1	-
BSET	R, b	将 R 的第 b 位置 1	1→R[b]	1	1	-
JBCLR	R, b	若 R 的第 b 位为 0, 则跳过下一条指令	若 R[b]=0, 则 PC+2→PC	1/2	1	-
JBSET	R, b	若 R 的第 b 位为 1, 则跳过下一条指令	若 R[b]=1, 则 PC+2→PC	1/2	1	-
ADDAI	I	I 和 A 相加, 结果存入 A	I+A→A	1	1	C, DC, Z
ADCAI	I	I 和 A 相加 (带 C 标志), 结果存入 A	I+A+C→A	1	1	C, DC, Z
ISUBAI	I	I 和 A 相减, 结果存入 A	I-A→A	1	1	C, DC, Z
ISBCAI	I	I 和 A 相减 (带 C 标志), 结果存入 A	I-A-/C→A	1	1	C, DC, Z
ASUBAI	I	A 和 I 相减, 结果存入 A	A-I→A	1	1	C, DC, Z
ASBCAI	I	A 和 I 相减 (带 C 标志), 结果存入 A	A-I-/C→A	1	1	C, DC, Z
ANDAI	I	I 和 A 与操作, 结果存入 A	I and A→A	1	1	Z
ORAI	I	I 和 A 或操作, 结果存入 A	I or A→A	1	1	Z
XORAI	I	I 和 A 异或操作, 结果存入 A	I xor A→A	1	1	Z
MOVAI	I	将 I 存入 A	I→A	1	1	-
CALL	K	子程序调用	PC+1→TOS; K→PC[12:0]	2	1	-
GOTO	K	无条件跳转	K→PC[12:0]	2	1	-
RETURN		从子程序返回	TOS→PC	2	1	-
RETAI	I	从子程序返回, 并将 I 存入 A	TOS→PC; I→A	2	1	-
RETIE		从中断返回	TOS→PC; 1→GIE	2	1	-
NOP		空操作	空操作	1	1	-
DAA		BCD 码加法操作后, 将 A 的值调整为 BCD 码	A (HEX 码)→A (BCD 码)	1	1	C
DSA		BCD 码减法操作后, 将 A 的值调整为 BCD 码	A (HEX 码)→A (BCD 码)	1	1	-



CLRWDT	将看门狗计数器清零	0→WDTCNT	1	1	TO, PD
STOP	进入低功耗模式	0→WDTCNT; CPU 暂停	1	1	TO, PD

注:

- 1、A-算术逻辑单元累加器 ALU, R-数据存储器, I-立即数, K-程序存储器地址, TOS-堆栈栈顶;
- 2、对于条件跳转类指令, 若跳转条件成立, 则执行时间需 2 个指令周期, 否则仅需 1 个指令周期;
- 3、禁止采用对 C, DC, Z 标志有影响的指令访问寄存器 PFLAG;

3.2 程序存储器

芯片的程序存储器为 FLASH 型存储器, 4K×16 位的地址空间范围为 0000H~0FFFH。程序存储器地址分配如下图所示:

复位起始地址 (0000H)
通用程序区 (0001H - 0007H)
中断入口地址 (0008H)
通用程序区 (0009H - 0FFFH)

程序存储器支持间接寻址, 可通过寄存器 INDF3 访问地址为 (FSR1×256+FSR0) 的程序存储器内容, 高 8 位将缓存于寄存器 HIBYTE, 低 8 位将缓存于寄存器 A。

例如, 采用间接寻址读取程序存储器 0155H 地址中内容, 高 8 位存入通用数据存储器 11H 地址中, 低 8 位存入通用数据存储器 10H 地址中:

```

MOVAI    01H
MOVRA    FSR1           ; 将 01H 写入 FSR1
MOVAI    55H
MOVRA    FSR0           ; 将 55H 写入 FSR0
MOVAR    INDF3          ; 读取 (FSR1×256+FSR0) 所指地址的程序存储器中内容
                                ; 高 8 位缓存于 HIBYTE, 低 8 位缓存于 A
MOVRA    10H           ; 将 A 中缓存的低 8 位存入通用数据存储器 10H 地址中
MOVAR    HIBYTE        ; 读取 HIBYTE 中缓存的高 8 位
MOVRA    11H           ; 高 8 位存入通用数据存储器 11H 地址中

```



3.3 数据存储器

芯片的数据存储器包括通用数据存储器 GPR（256 字节）和特殊功能寄存器 SFR，地址映射如下表所示。其中 GPR 可直接寻址或通过 INDF0/INDF2 间接寻址，SFR 可直接寻址或通过 INDF1/INDF2 间接寻址。

数据存储器还包括掉电非易失的 EEPROM 型数据存储器（128×16 位），需通过 SFR 进行读写操作，详细说明请参见后续章节。

数据存储器区地址映射表

地址	类型	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
000H-0FFH	GPR	通用数据存储器区							
100H-17FH	保留	保留							
180H-187H	SFR	INDF0	INDF1	INDF2	HIBYTE	FSR0	FSR1	PCL	PFLAG
188H-18FH		MCR	INDF3	INTE0	INTF0	OSCMR	INTE1	INTF1	LVDCR
190H-197H		IOP0	OEP0	PUP0	PDP0	IOP1	OEP1	PUP1	PDP1
198H-19FH		IOP2	OEP2	PUP2	PDP2	PWMOCR0	PWMOCR1	PWMOCR2	
1A0H-1A7H		TOCR	TOCNT	TOLOAD	TODATA		T1CR	T1CNT	T1LOAD
1A8H-1AFH		T2CR	T2CNT	T2LOAD		T2EFR	T3CR	T3CNT	T3LOAD
1B0H-1B7H		IICCR	IICAR	IICDR	EEDRH	EECR	EEMASK	EEAR	EEDRL
1B8H-1BFH		ADCR0	ADCR1	ADRH	ADRL	POADCR	PIADCR	OSADJCR	MEFCR
1C0H-1C7H						CP0TDB	OPACR	OPAACL	CP4CLR
1C8H-1CFH		CPCR0	CPSR	CPVR0	CPVR1	CP0CLR	CP1CLR	CP2CLR	CP3CLR
1D0H-1D7H		CPCR1	CPCR2	CPDB0	CPDB1			DBGCR	DBGPR
1D8H-1FFH		保留	保留						

注：上表中灰色部分的存储器地址为系统保留区，禁止对其中未定义的地址进行读写操作。

数据存储器寻址方式地址组成

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	寻址方式
/	/	/	/	/	/	/	取自指令的 9 位地址								直接寻址方式	
/	/	/	/	/	/	/	0	FSR0								间接寻址方式 0
/	/	/	/	/	/	/	1	FSR1								间接寻址方式 1
FSR1							FSR0								间接寻址方式 2	

直接寻址方式，是以指令的低 9 位为数据存储器地址，通过指令访问，寻址范围 0~1FFH。例如，采用直接寻址方式将数据 55H 写入数据存储器 010H 地址中：

```
MOVAI    55H
MOVRA    10H          ; 将 55H 写入数据存储器 10H 地址中
```

间接寻址方式 0，是以 FSR0 为数据存储器地址指针，通过 INDF0 访问，寻址范围 0~0FFH。例如，采用间接寻址方式 0 将数据 55H 写入数据存储器 010H 地址中：



```

MOVAI    10H
MOVRA    FSR0
MOVAI    55H
MOVRA    INDF0          ; 将 55H 写入 FSR0 所指地址的数据存储器中

```

间接寻址方式 1，是以 FSR1 为数据存储器地址指针，通过 INDF1 访问，寻址范围 100H~1FFH。例如，采用间接寻址方式 1 将数据 55H 写入数据存储器 110H 地址中：

```

MOVAI    10H
MOVRA    FSR1
MOVAI    55H
MOVRA    INDF1          ; 将 55H 写入 (FSR1+256) 所指地址的数据存储器中

```

间接寻址方式 2，是以[FSR1:FSR0]为数据存储器地址指针，通过 INDF2 访问，寻址范围 0~FFFFH。例如，采用间接寻址方式 2 将数据 55H 写入数据存储器 0010H 地址中：

```

MOVAI    00H
MOVRA    FSR1
MOVAI    10H
MOVRA    FSR0
MOVAI    55H
MOVRA    INDF2          ; 将 55H 写入 (FSR1×256+FSR0) 所指地址的数据存储器中

```

注：间接寻址方式 2 最大可寻址 FFFFH，但访问数据存储器中未定义的地址时，读出数据不确定，写入操作可能会更改其他地址中的内容。

3.4 堆栈

芯片的堆栈为 8 级深度的硬件堆栈。当 CPU 响应中断或执行子程序调用指令时，会自动将下一条指令的 PC 值压栈保存；当 CPU 执行中断返回或子程序返回指令时，会自动将栈顶内容出栈载入 PC。

3.5 控制寄存器

数据指针寄存器 0

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
FSR0	FSR07	FSR06	FSR05	FSR04	FSR03	FSR02	FSR01	FSR00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **FSR0[7:0]** – 数据指针寄存器 0

FSR0: 间接寻址方式 0 的指针，或间接寻址方式 2、3 的指针低 8 位。



数据指针寄存器 1

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
FSR1	FSR17	FSR16	FSR15	FSR14	FSR13	FSR12	FSR11	FSR10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **FSR1[7:0]** – 数据指针寄存器 1

FSR1: 间接寻址方式 1 的指针, 或间接寻址方式 2、3 的指针高 8 位。

间接寻址寄存器 0

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INDF0	INDF07	INDF06	INDF05	INDF04	INDF03	INDF02	INDF01	INDF00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **INDF0[7:0]** – 间接寻址寄存器 0

INDF0: INDF0 不是物理寄存器, 对 INDF0 操作实际是对 FSR0 所指向地址的数据存储器进行操作, 从而实现间接寻址功能。

间接寻址寄存器 1

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INDF1	INDF17	INDF16	INDF15	INDF14	INDF13	INDF12	INDF11	INDF10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **INDF1[7:0]** – 间接寻址寄存器 1

INDF1: INDF1 不是物理寄存器, 对 INDF1 操作实际是对 (FSR1+256) 所指向地址的数据存储器进行操作, 从而实现间接寻址功能。

间接寻址寄存器 2

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INDF2	INDF27	INDF26	INDF25	INDF24	INDF23	INDF22	INDF21	INDF20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **INDF2[7:0]** – 间接寻址寄存器 2

INDF2: INDF2 不是物理寄存器, 对 INDF2 操作实际是对 (FSR1×256+FSR0) 所指向地址的数据存储器进行操作, 从而实现间接寻址功能。

间接寻址寄存器 3

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INDF3	INDF37	INDF36	INDF35	INDF34	INDF33	INDF32	INDF31	INDF30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X



BIT[7:0] **INDF3[7:0]** – 间接寻址寄存器 3

INDF3: INDF3 不是物理寄存器, 对 INDF3 操作实际是对 (FSR1×256+FSR0) 所指向地址的程序存储器进行操作, 从而实现间接寻址功能。

注: 对寄存器 INDF3 仅可执行读取操作, 且仅可使用读取指令 (MOVAR INDF3), 所读程序存储器内容的高 8 位存入寄存器 HIBYTE, 低 8 位存入寄存器 A。

字操作高字节寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
HIBYTE	HIBYTE7	HIBYTE6	HIBYTE5	HIBYTE4	HIBYTE3	HIBYTE2	HIBYTE1	HIBYTE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **HIBYTE[7:0]** – 字操作高字节寄存器

HIBYTE: 用于缓存通过 INDF3 访问程序存储器时所读取内容的高 8 位。

程序指针计数器低字节

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **PC[7:0]** – 程序指针计数器低 8 位

程序指针计数器 (PC) 有以下几种操作模式:

- ◇ 顺序运行指令: $PC = PC + 1$;
- ◇ 程序跳转指令 GOTO/CALL: $PC =$ 指令码低 13 位;
- ◇ 返回指令 RETIE/RETURN/RETAI: $PC =$ 堆栈栈顶 (TOS);

对 PCL 操作指令:

- ◇ 对 PCL 操作的加法指令: $PC = (PC[12:0] + ALU[7:0])$;
- ◇ 对 PCL 操作的其他指令: $PC = (PC[12:8]:ALU[7:0](ALU \text{ 运算结果}))$;

CPU 状态寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PFLAG	-	-	-	-	-	Z	DC	C
R/W	-	-	-	-	-	R/W	R/W	R/W
初始值	-	-	-	-	-	X	X	X

BIT[2] **Z** – 零标志位

- 0: 算术或逻辑运算的结果不为零;
- 1: 算术或逻辑运算的结果为零;

BIT[1] **DC** – 半字节进位/借位标志位

- 0: 加法运算中半字节无进位; 减法运算中半字节有借位;
- 1: 加法运算中半字节有进位; 减法运算中半字节无借位;



- BIT[0] **C** – 进位/借位标志位
 0: 加法运算中无进位；减法运算中有借位；移位操作中移出位为 0；
 1: 加法运算中有进位；减法运算中无借位；移位操作中移出位为 1；

杂项控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
MCR	GIE	–	TO	PD	INT1M1	INT1M0	INTOM1	INTOM0
R/W	R/W	–	R	R	R/W	R/W	R/W	R/W
初始值	0	–	0	0	0	0	0	0

- BIT[7] **GIE** – 中断总使能位
 0: 屏蔽所有中断；
 1: 由相应的中断使能位决定 CPU 是否响应中断源所触发的中断；

- BIT[5] **TO** – 看门狗溢出标志位
 0: 上电复位，或执行 CLRWDT/STOP 指令；
 1: 发生 WDT 溢出；

- BIT[4] **PD** – 进入低功耗模式标志位
 0: 上电复位，或执行 CLRWDT 指令；
 1: 执行 STOP 指令；

- BIT[3:2] **INT1M[1:0]** – 外部中断 INT1 触发方式选择位

INT1M[1:0]	INT1 触发方式
00	上升沿触发
01	下降沿触发
1X	电平变化触发

- BIT[1:0] **INTOM[1:0]** – 外部中断 INTO 触发方式选择位

INTOM[1:0]	INTO 触发方式
00	上升沿触发
01	下降沿触发
1X	电平变化触发

3.6 用户配置字

芯片为保证系统正常工作，会将关键模块的配置信息预先存储于单独的存储器区域内，在上电或其他复位发生后将配置信息载入寄存器中，通过寄存器控制关键模块的工作状态。该部分存储器中用户可选的内容即为用户配置字，可在烧录用户程序代码时进行配置与烧录。



4 系统时钟

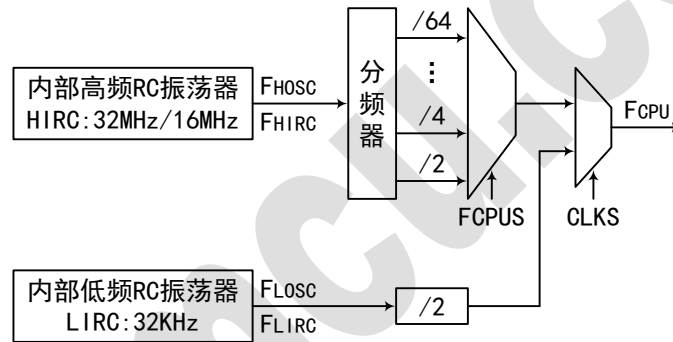
芯片内部电路均在系统高频时钟 F_{HOSC} 或系统低频时钟 F_{LOSC} 下工作，系统及部分外设模块的时钟源还可在 F_{HOSC} 和 F_{LOSC} 之间切换。

系统高频时钟 F_{HOSC} 固定为内部高频 RC 振荡器 $HIRC$ (32MHz/16MHz) 时钟 F_{HIRC} ；系统低频时钟 F_{LOSC} 固定为内部低频 RC 振荡器 $LIRC$ (32KHz) 时钟 F_{LIRC} 。

CPU 的时钟源可在系统高频时钟 F_{HOSC} 和系统低频时钟 F_{LOSC} 之间切换。 F_{HOSC} 下 CPU 的时钟频率 F_{CPU} 通过配置字 $FCPUS$ 选择； F_{LOSC} 下 F_{CPU} 则固定为 F_{LOSC} 的 2 分频。

WDT (看门狗) 电路的时钟源固定为内部低频 RC 振荡器 $LIRC$ 。

系统时钟示意图



4.1 内部高频 RC 振荡器

芯片内置 1 个振荡频率可通过配置字 $HIRCFS$ 选择 (32MHz/16MHz) 的高精度 $HIRC$ 振荡器，可用作系统高频时钟源。

4.2 内部低频 RC 振荡器

芯片内置 1 个振荡频率典型值为 32KHz 的 $LIRC$ 振荡器，可用作系统低频时钟源，也用于系统上电延时控制、看门狗定时器 (WDT) 等电路。



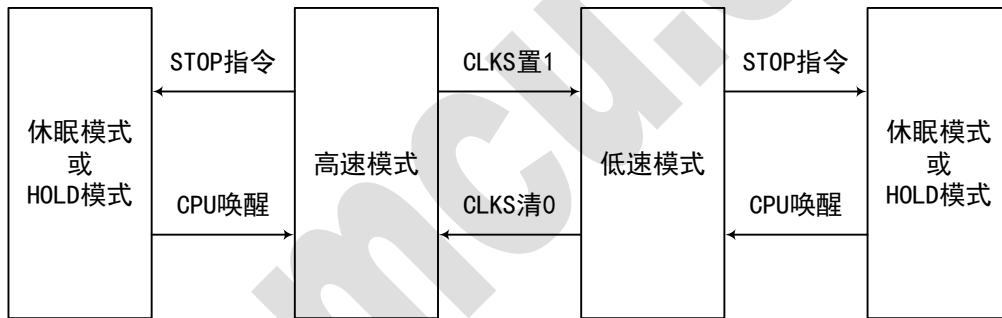
4.3 系统工作模式

芯片支持高速模式、低速模式、HOLD1 模式、HOLD2 模式和休眠模式等多种系统工作模式。

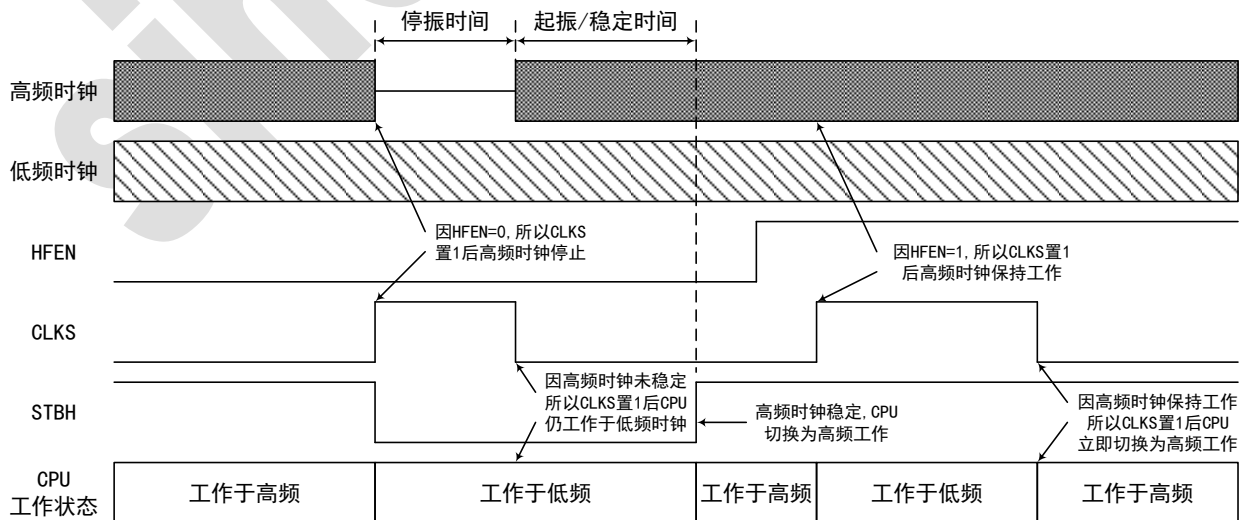
工作模式	模式切换条件	系统工作状态
高速	任意模式下, 系统复位	CPU 高速运行, 高/低频时钟源均工作
	低速模式下, CLKS 清 0	
	HOLD1/HOLD2/休眠模式下, CPU 唤醒 (@CLKS=0)	
低速	高速模式下, CLKS 置 1	CPU 低速运行, 低频时钟源工作, 高频时钟源由使能位 HFEN 决定
	HOLD1/HOLD2/休眠模式下, CPU 唤醒 (@CLKS=1)	
HOLD1	高/低速模式下, 执行 STOP 指令 (@HFEN=1)	CPU 暂停, 高频时钟源工作, 低频时钟源由使能位 LFEN 决定
HOLD2	高/低速模式下, 执行 STOP 指令 (@HFEN=0, LFEN=1)	CPU 暂停, 高频时钟源停止, 低频时钟源工作
休眠	高/低速模式下, 执行 STOP 指令 (@HFEN=0, LFEN=0)	CPU 暂停, 高/低频时钟源均停止

注: WDT 时钟源为 LIRC, WDT 开启时 LIRC 将一直工作而不受系统工作模式影响。

工作模式切换示意图



高低频时钟切换时序图





振荡器模式寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OSCMR	-	-	STBL	STBH	-	CLKS	LFEN	HFEN
R/W	-	-	R	R	-	R/W	R/W	R/W
初始值	-	-	X	1	-	0	0	0

BIT[5] **STBL** – 低频时钟源振荡状态标志位

- 0: 低频时钟源停振或未稳定;
- 1: 低频时钟源已稳定振荡;

BIT[4] **STBH** – 高频时钟源振荡状态标志位

- 0: 高频时钟源停振或未稳定;
- 1: 高频时钟源已稳定振荡;

BIT[2] **CLKS** – CPU 时钟源选择位

- 0: 系统高频时钟作为 CPU 时钟源;
- 1: 系统低频时钟作为 CPU 时钟源;

BIT[1] **LFEN** – 低频时钟源使能位

- 0: 在休眠/HOLD 模式下, 低频时钟源暂停工作;
- 1: 低频时钟源始终工作;

BIT[0] **HFEN** – 高频时钟源使能位

- 0: 在低速/休眠/HOLD 模式下, 高频时钟源暂停工作;
- 1: 高频时钟源始终工作;

4.4 低功耗模式

芯片的高速模式、低速模式为运行模式, 而休眠模式、HOLD1 模式、HOLD2 模式则为低功耗模式。

执行 STOP 指令可使系统进入低功耗模式, 同时对系统会产生以下影响:

- ✧ CPU 停止运行;
- ✧ 根据不同模式停止相应时钟源的振荡;
- ✧ RAM 内容保持不变;
- ✧ 所有的输入/输出端口保持原有状态;
- ✧ 定时器若其时钟源未停止, 则可继续工作;

以下情况可使系统退出低功耗模式:

- ✧ 芯片复位;
- ✧ WDT 溢出 (若低功耗模式下 WDT 及其时钟源保持继续工作);
- ✧ 外部中断请求发生 (若有外部中断功能并有效);



- ◇ 定时器中断请求发生（若低功耗模式下定时器及其时钟源保持继续工作）；
- ◇ 比较器 CMP 中断请求发生（若低功耗模式下比较器保持继续工作）；
- ◇ LVD 中断请求发生（若低功耗模式下 LVD 保持继续工作）；

注：

- 1、低功耗模式下触发中断请求时，若对应的中断使能位关闭，则不会退出低功耗模式；若对应的中断使能位开启而中断总使能位关闭，则仅唤醒 CPU 执行下一条指令；若对应的中断使能位和中断总使能位均开启，则唤醒 CPU 后将执行中断服务程序；
- 2、未使用或未封出的引脚，应将其对应的 I/O 端口设置为输出、输入上拉或输入下拉等稳定状态，以免因引脚浮空而产生漏电流或非预期的中断唤醒；



5 复位

5.1 复位条件

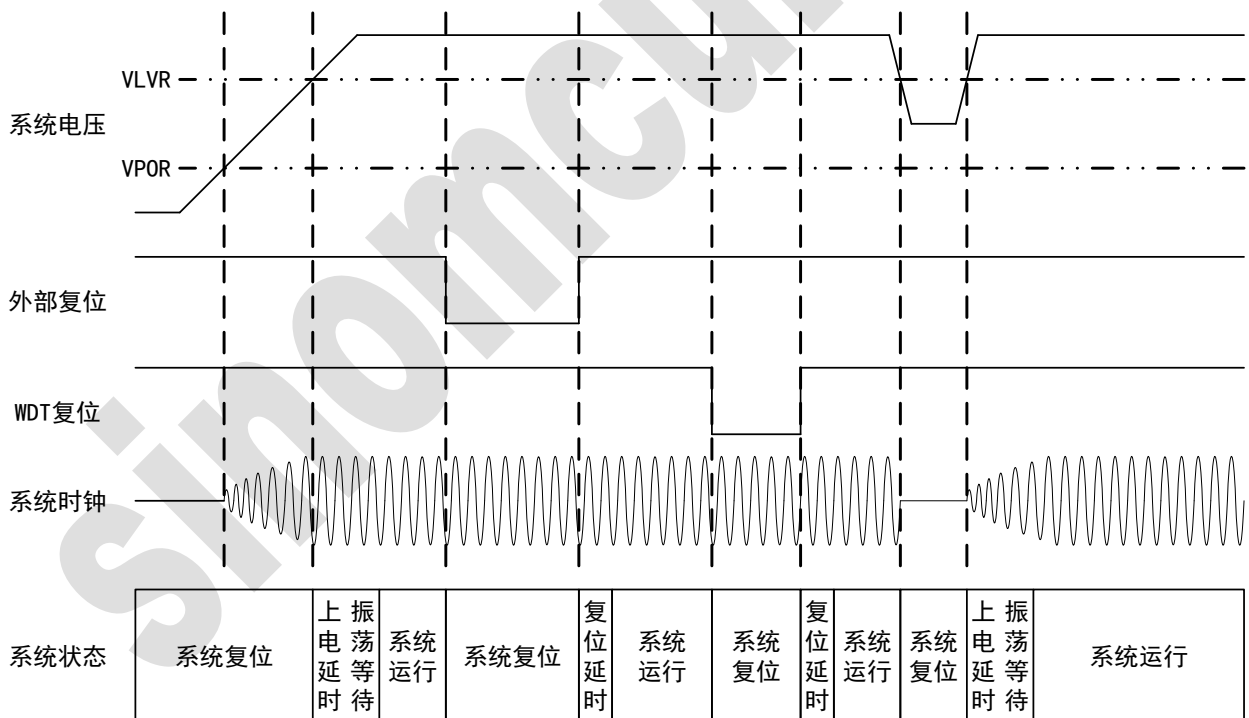
芯片共有如下几种复位方式：

- ◇ 上电复位 POR；
- ◇ 低电压复位 LVR；
- ◇ 外部复位；
- ◇ WDT 看门狗复位；

任何一种复位发生后，系统进入复位状态，执行初始化操作并重置 SFR 为复位初始值；复位条件解除后，系统退出复位状态，CPU 重新从程序存储器 0000H 地址处开始运行。

上电复位 POR 和低电压复位 LVR，复位解除后系统将保持一定时间的上电延时以待稳定后才开始工作；而外部复位、WDT 复位，复位解除后系统将在较短的复位延时后即开始工作。

下图是复位产生和系统工作状态之间的时序关系示意图：



注：若应用系统在上电或掉电回升时芯片的 VDD 电压上升较慢，则应在复位后 CPU 开始工作时先进行软件延时，以确保芯片开始工作时 VDD 电压已稳定在 F_{CPU} 对应的工作电压范围内。



5.2 上电复位

芯片的上电复位电路可以适应系统快速上电或慢速上电等情况，即使上电过程中发生电源电压抖动的情况也能保证系统可靠的复位。

上电复位过程主要包括以下几个步骤：

- (1) 检测系统电源电压，等待电压高于上电复位电压 V_{POR} 并保持稳定；
- (2) 若 LVR 功能开启，则需等待电压高于低电压复位电压 V_{LVR} 并保持稳定；
- (3) 若有外部复位功能并已开启，则需等待外部复位引脚电压高于 V_{ih} ；
- (4) 初始化所有初始值确定的寄存器；
- (5) 开启主时钟振荡器，并等待一段时间以待振荡器稳定；
- (6) 上电复位结束，CPU 开始执行指令；

5.3 外部复位

芯片的外部复位功能可通过配置字 $RSTEN$ 开启，引脚设为外部复位脚即为开启外部复位功能，端口内部上拉电阻将自动使能。外部复位输入端口 RST 为施密特结构，低电平有效，即当端口输入为高电平时系统正常运行，输入为低电平时系统复位。

5.4 低电压复位

芯片的低电压复位电压 V_{LVR} 可通过配置字 $LVRVS$ 选择。LVR 检测电路具有一定的回滞特性，回滞电压约为 6%（典型值），当电源电压下降至 V_{LVR} 时发生 LVR 复位，反之电源电压需上升至 $V_{LVR}+6\%$ 后 LVR 复位才解除。

5.5 看门狗复位

芯片的看门狗定时器（WDT）复位是一种对系统运行程序的保护机制。正常情况下，用户程序需定时对 WDT 执行清零操作，以避免 WDT 溢出。若发生异常情况，程序未及时清零 WDT，则芯片将因 WDT 溢出而产生看门狗复位，系统初始化后重新运行程序，从而返回受控状态。

注：低功耗模式下 CPU 暂停工作，若此时发生 WDT 溢出，则仅唤醒 CPU 而不复位芯片。



6 I/O 端口

6.1 通用 I/O 功能

芯片的输入/输出端口包括两组 8 位端口 P0、P1，和一组 2 位端口 P2。所有端口均支持施密特输入，均支持推挽输出。

除用作通用数字 I/O 端口外，部分端口还可复用为外部中断输入、PWM 输出、或 ADC 模拟输入等功能。

端口数据寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IOP0	P07D	P06D	P05D	P04D	P03D	P02D	P01D	P00D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **P0nD** – P0n 端口数据位 (n=7-0)

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IOP1	P17D	P16D	P15D	P14D	P13D	P12D	P11D	P10D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **P1nD** – P1n 端口数据位 (n=7-0)

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IOP2	-	-	-	-	-	-	P21D	P20D
R/W	-	-	-	-	-	-	R/W	R/W
初始值	-	-	-	-	-	-	X	X

BIT[1:0] **P2nD** – P2n 端口数据位 (n=1-0)

端口方向寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OEP0	P07OE	P06OE	P05OE	P04OE	P03OE	P02OE	P01OE	P00OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P0nOE** – P0n 端口输出使能位 (n=7-0)

- 0: 端口作为输入口，读端口操作将读取端口的电平状态；
- 1: 端口作为输出口，读端口操作将读取端口的数据位值；

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OEP1	P17OE	P16OE	P15OE	P14OE	P13OE	P12OE	P11OE	P10OE



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P1nOE** – P1n 端口输出使能位 (n=7-0)

- 0: 端口作为输入口, 读端口操作将读取端口的电平状态;
1: 端口作为输出口, 读端口操作将读取端口的数据位值;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OEP2	-	-	-	-	-	-	P210E	P200E
R/W	-	-	-	-	-	-	R/W	R/W
初始值	-	-	-	-	-	-	0	0

BIT[1:0] **P2nOE** – P2n 端口输出使能位 (n=1-0)

- 0: 端口作为输入口, 读端口操作将读取端口的电平状态;
1: 端口作为输出口, 读端口操作将读取端口的数据位值;

6.2 内部上/下拉电阻

所有端口均具有内部上拉和下拉电阻, 且均可单独控制其上/下拉电阻在端口处于输入状态(或端口数字 I/O 功能关闭)时是否有效。端口处于输出状态时, 上/下拉电阻及其控制位无效。

上拉电阻控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PUP0	P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P0nPU** – P0n 端口上拉电阻控制位 (n=7-0)

- 0: 端口内部上拉电阻无效;
1: 端口内部上拉电阻有效;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PUP1	P17PU	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P1nPU** – P1n 端口上拉电阻控制位 (n=7-0)

- 0: 端口内部上拉电阻无效;
1: 端口内部上拉电阻有效;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PUP2	-	-	-	-	-	-	P21PU	P20PU



R/W	-	-	-	-	-	-	R/W	R/W
初始值	-	-	-	-	-	-	0	0

BIT[1:0] **P2nPU** – P2n 端口上拉电阻控制位 (n=1-0)

- 0: 端口内部上拉电阻无效;
- 1: 端口内部上拉电阻有效;

下拉电阻控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PDP0	P07PD	P06PD	P05PD	P04PD	P03PD	P02PD	P01PD	P00PD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P0nPD** – P0n 端口下拉电阻控制位 (n=7-0)

- 0: 端口内部下拉电阻无效;
- 1: 端口内部下拉电阻有效;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PDP1	P17PD	P16PD	P15PD	P14PD	P13PD	P12PD	P11PD	P10PD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P1nPD** – P1n 端口下拉电阻控制位 (n=7-0)

- 0: 端口内部下拉电阻无效;
- 1: 端口内部下拉电阻有效;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PDP2	-	-	-	-	-	-	P21PD	P20PD
R/W	-	-	-	-	-	-	R/W	R/W
初始值	-	-	-	-	-	-	0	0

BIT[1:0] **P2nPD** – P2n 端口下拉电阻控制位 (n=1-0)

- 0: 端口内部下拉电阻无效;
- 1: 端口内部下拉电阻有效;

6.3 端口模式控制

部分端口除可作为数字端口外，还可复用为模拟端口。端口输入或输出模拟信号时，若数字 I/O 功能同时开启，则会产生漏电流，可通过端口数模控制寄存器关闭端口的数字 I/O 功能（内部上/下拉电阻及其控制位不受影响）。



端口数模控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
POADCR	P07DC	P06DC	P05DC	P04DC	P03DC	P02DC	P01DC	P00DC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P0nDC** – P0n 端口数字功能控制位 (n=7-0)

- 0: 使能端口的数字 I/O 功能;
- 1: 关闭端口的数字 I/O 功能;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
P1ADCR	P17DC	P16DC	P15DC	P14DC	P13DC	P12DC	P11DC	P10DC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P1nDC** – P1n 端口数字功能控制位 (n=7-0)

- 0: 使能端口的数字 I/O 功能;
- 1: 关闭端口的数字 I/O 功能;



7 定时器 TIMER

7.1 看门狗定时器 WDT

看门狗定时器 WDT 的时钟源为内部低频 RC 振荡器 LIRC，WDT 溢出将复位芯片或唤醒 CPU。

可通过配置字 WDTM 设置 WDT 工作模式：选择始终开启，则 WDT 一直工作，高速/低速模式下 WDT 溢出将复位芯片，休眠/HOLD 模式下 WDT 溢出将唤醒 CPU；选择低功耗模式下关闭，则 WDT 在休眠/HOLD 模式下自动关闭、在其他方式唤醒 CPU 后恢复工作。

执行 CLRWDT 指令或 STOP 指令将清零 WDT 计数器。

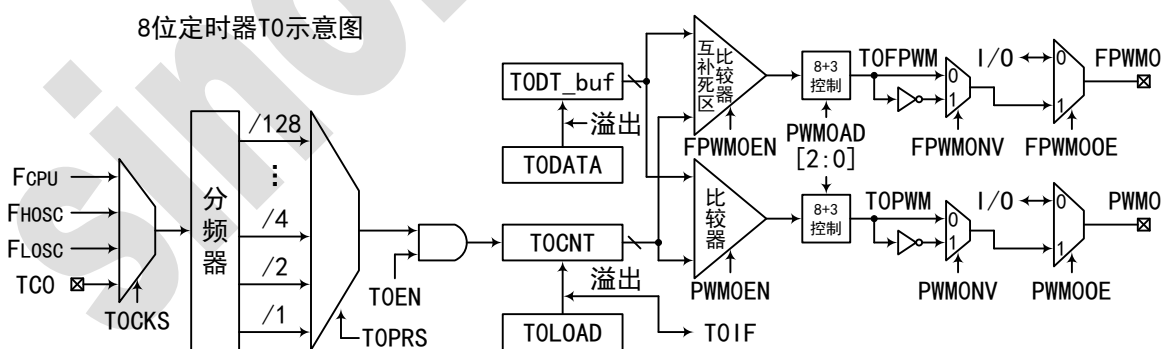
WDT 溢出时间可配置为 64ms/2048ms。

注：WDT 溢出时间为典型值，而实际值偏差较大，必须保证清 WDT 的间隔时间小于 WDT 溢出时间的 1/4。

7.2 定时器 T0

定时器 T0 为 8 位定时/计数器，包含 1 个 8 位递减计数器、可编程预分频器、控制寄存器、8 位重载寄存器和 8 位比较寄存器。

- ◇ 可通过预分频器设置时钟频率，可通过重载寄存器控制计数周期；
- ◇ 支持 1 对 8+3 模式的带死区互补 PWM 输出，可通过比较寄存器设置 PWM 占空比；
- ◇ 支持溢出中断和溢出唤醒功能；



定时器 T0，可通过寄存器位 T0CKS 选择时钟源，通过 T0PRS 选择时钟预分频比，所选时钟源通过预分频器后产生 T0 计数器 TOCNT 的计数时钟（上升沿计数）。写 TOCNT 将清零预分频计数器，而预分频比保持不变。

TOEN=0 时，TOCNT 保持不变，写重载寄存器 TOLOAD 将立即载入 TOCNT；TOEN=1 时，TOCNT 递减计数，计数到 0 的时钟结束后产生溢出信号并触发中断，中断标志 TOIF 将被置 1，同时 T0 自动将当前 TOLOAD 值载入 TOCNT 并重新开始计数。



如图所示，定时器 T0 可实现 1 对互补且带死区（2 路互补信号高电平非交叠时间）控制的 PWM 功能（PWM0/FPWM0），可通过寄存器位使能或关闭 PWM 功能，并控制端口是否输出 PWM 波形。PWM0 关闭时 TOPWM 信号为低电平，FPWM0 关闭时 TOFPWM 信号为高电平。PWM0/FPWM0 使能后 TOCNT 从重载值开始递减计数直到计数溢出为一个 PWM 周期；当计数到与比较寄存器 TODATA 相等时，TOPWM 变为高电平（TOFPWM 提前下降沿非交叠时间变为低电平）；当计数溢出时，TOPWM 变为低电平（TOFPWM 滞后上升沿非交叠时间变为高电平）。

TODATA 配有 1 个 8 位比较缓冲器（T0DT_buf）用于与 TOCNT 比较，PWM0 和 FPWM0 均关闭时写 TODATA 将立即载入缓冲器中，而 PWM0 或 FPWM0 使能后写 TODATA 则将在 T0 溢出时才载入缓冲器中。若要首个 PWM 周期和占空比准确，需先写重载寄存器和比较寄存器，再使能 PWM，最后开启定时器。

TOPWM 信号的占空比计算如下：

- ◇ 高电平时间 = (TODATA) × TOCNT 计数时钟周期
- ◇ 周期 (T0 溢出时间) = (TOLOAD + 1) × TOCNT 计数时钟周期
- ◇ 占空比 (高电平时间/周期) = (TODATA) / (TOLOAD + 1)

PWM0/FPWM0 支持 8+3 模式，即在普通模式 PWM 的时钟下，以每 8 个 PWM 周期作为一个大周期，其中部分周期进行占空比延展（即延展高电平时间），这些延展周期内 TOPWM 信号将比普通周期提前半个计数时钟变为高电平，即 TOPWM 高电平时间延展为 (TODATA+0.5)。

8+3 模式提高的是整体上的 PWM 占空比调节精度，而不是单个 PWM 周期的占空比调节精度。占空比延展控制位 PWM0D[2:0] 决定每 8 个周期内哪几个周期为普通周期，哪几个周期为延展周期。

PWM0/FPWM0 的占空比可通过寄存器位 PWM0DB 微调半个计数时钟周期。PWM0DB=1 时，普通周期内 TOPWM 信号在计数时钟上升沿变为高电平，而延展周期内则提前半个时钟周期在时钟下降沿翻转；PWM0DB=0 时，普通周期内 TOPWM 信号滞后半个时钟周期在时钟下降沿变为高电平，而延展周期内则变为正常的时钟上升沿翻转。TOFPWM 信号变为低电平的时间则在叠加下降沿死区时间后进行相应微调。

PWM0 的 8+3 模式一个大周期的占空比计算如下（需 $0 < TODATA < TOLOAD$ ）：

- ◇ 高电平时间 = $[(TODATA + (PWM0DB - 1)/2) \times 8 + (PWM0D[2:0])/2] \times TOCNT$ 计数时钟周期
- ◇ 周期 = $(TOLOAD + 1) \times 8 \times TOCNT$ 计数时钟周期
- ◇ 占空比 (高电平时间/周期) = $[TODATA + (PWM0DB - 1)/2 + (PWM0D[2:0]/16)] / (TOLOAD + 1)$

举例说明 (PWM0DB=1)：

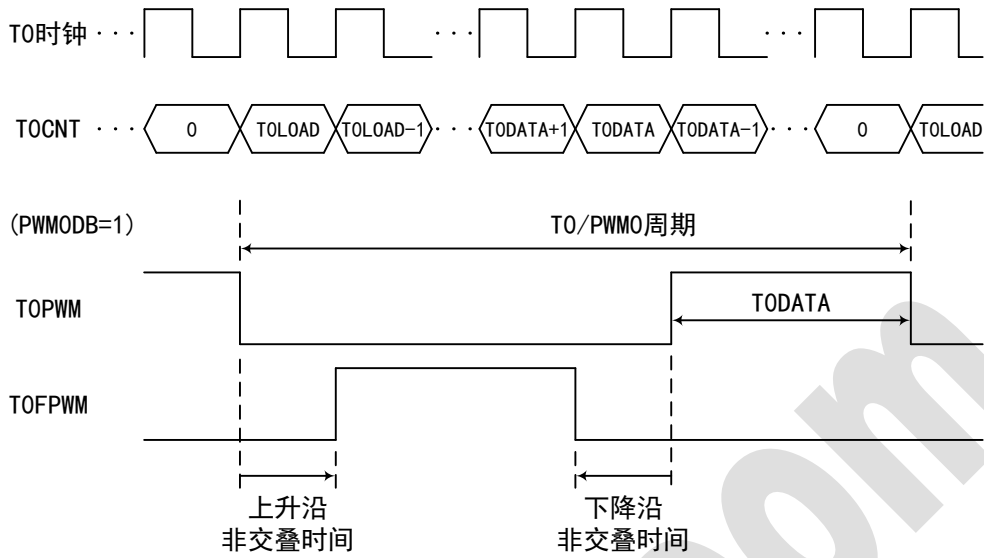
若 PWM0D[2:0]=B'000，则 8 个 PWM 周期均不进行占空比延展，一个大周期 PWM 的占空比与普通模式 PWM 的占空比一样，为 $[(TODATA) / (TOLOAD + 1)]$ ；

若 PWM0D[2:0]=B'001，则每 8 个 PWM 周期中的第 1 个周期进行占空比延展，一个大周期的占空比为 $[(TODATA) + (1/16)] / (TOLOAD + 1)$ ；

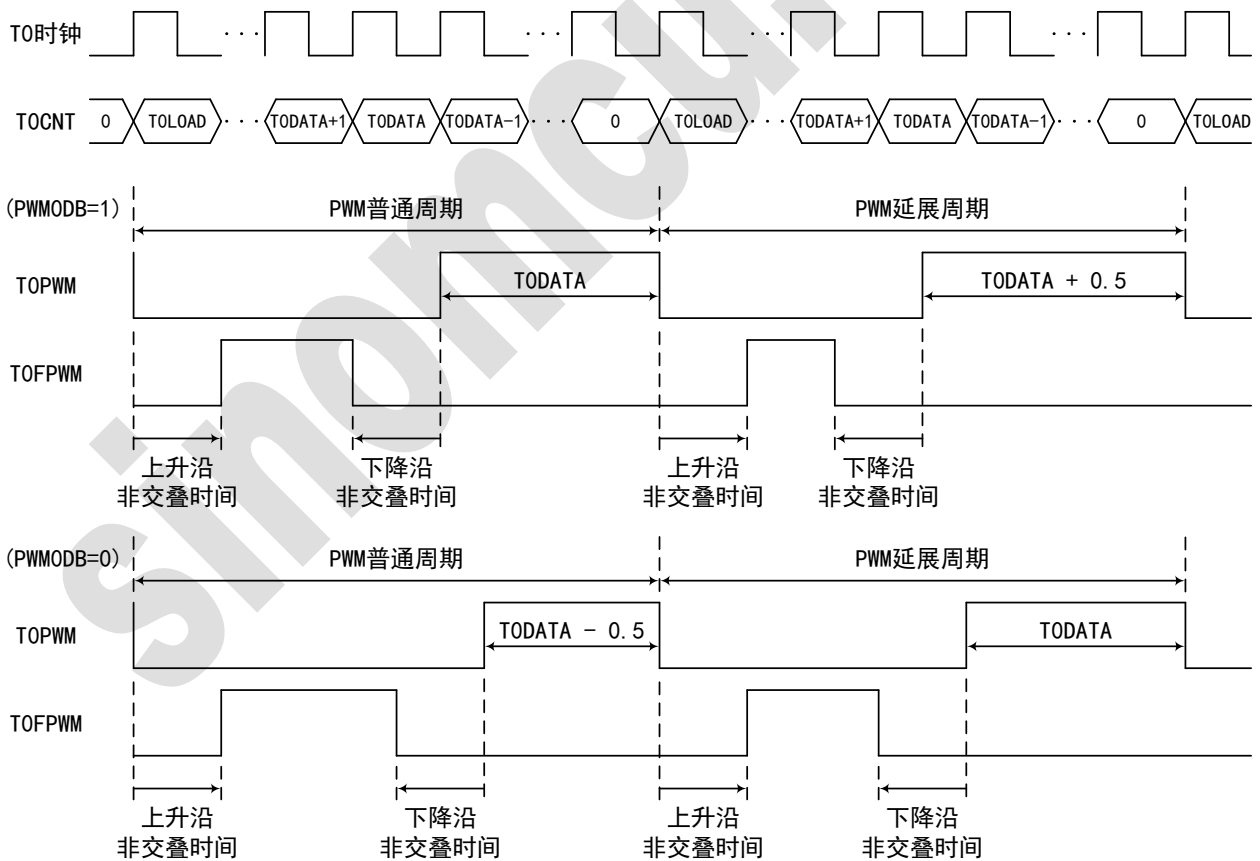
若 PWM0D[2:0]=B'101，则每 8 个 PWM 周期中的第 1、2、3、5、7 个周期（共 5 个周期）进行占空比延展，一个大周期的占空比为 $[(TODATA) + (5/16)] / (TOLOAD + 1)$ ；



PWM 互补及死区波形示意



PWMODB 及 8+3 模式波形示意



注:

- 1、应用互补 PWM 时, 前后死区的总时间应小于 TOPWM 低电平时间, 以确保 TOFPWM 能正常生成高电平;



2. 不可在 PWM 工作时调整 PWM 周期 (即定时器周期) 和死区时间;
3. 8+3 模式下, 当 PWM0DB=1 时, 正常周期 (不延展的周期) 的占空比 (高电平时间) 为 TODATA, 而延展周期的占空比则延展半个时钟周期 (即为 TODATA+0.5); 当 PWM0DB=0 时, 所有周期 (包括正常周期和延展周期) 的占空比均缩减半个时钟周期;
4. 若 TOPWM 或 TOPPWM 在 PWM0DB 为 1 时高电平时间仅为 1 个时钟周期, 则 PWM0DB 清 0 操作并不会将高电平时间缩减为半个时钟周期;
5. 因 Fcpu 的占空比不为 50%, 所以当 PWM 计数时钟为 Fcpu 的 1 分频时, 微调功能的翻转延迟并不正好为半个时钟周期, 而是延迟为 Fcpu 的高电平时间;

定时器 T0 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
TOCR	TOEN	PWM0EN	FPWM0EN	TOCKS1	TOCKS0	TOPRS2	TOPRS1	TOPRS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] **TOEN** – 定时器 T0 使能位

- 0: 关闭定时器 T0;
1: 开启定时器 T0;

BIT[6] **PWM0EN** – PWM0 使能位

- 0: 关闭 PWM0 功能;
1: 使能 PWM0 功能;

BIT[5] **FPWM0EN** – FPWM0 使能位

- 0: 关闭 FPWM0 功能;
1: 使能 FPWM0 功能;

BIT[4:3] **TOCKS[1:0]** – T0 时钟源选择位

TOCKS[1:0]	T0 时钟源
00	Fcpu
01	FHosc
10	FLosc
11	TC0 上升沿

BIT[2:0] **TOPRS[2:0]** – T0 时钟预分频比选择位

TOPRS[2:0]	T0 时钟预分频比
000	1 : 1
001	1 : 2
010	1 : 4
011	1 : 8
100	1 : 16



101	1 : 32
110	1 : 64
111	1 : 128

定时器 T0 计数器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
TOCNT	TOCNT7	TOCNT6	TOCNT5	TOCNT4	TOCNT3	TOCNT2	TOCNT1	TOCNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **TOCNT[7:0]** – T0 计数器，为可读写的递减计数器

定时器 T0 重载寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
TOLOAD	TOLOAD7	TOLOAD6	TOLOAD5	TOLOAD4	TOLOAD3	TOLOAD2	TOLOAD1	TOLOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **TOLOAD[7:0]** – T0 重载寄存器，用于设置 T0 的计数周期

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。

定时器 T0 比较寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
TODATA	TODATA7	TODATA6	TODATA5	TODATA4	TODATA3	TODATA2	TODATA1	TODATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **TODATA[7:0]** – T0 比较寄存器，用于设置 PWM0 的占空比

PWM0 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PWMOCRO	-	-	-	-	FPWM0NV	FPWM0OE	PWM0NV	PWM0OE
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3] **FPWM0NV** – FPWM0 端口输出取反控制位

- 0: 端口输出正向波形；
- 1: 端口对电平取反后输出；

BIT[2] **FPWM0OE** – FPWM0 端口输出使能位

- 0: 禁止端口输出脉宽调制波形；
- 1: 允许端口输出脉宽调制波形；



BIT[1] **PWM0NV** – PWM0 端口输出取反控制位

- 0: 端口输出正向波形;
- 1: 端口对电平取反后输出;

BIT[0] **PWM0OE** – PWM0 端口输出使能位

- 0: 禁止端口输出脉宽调制波形;
- 1: 允许端口输出脉宽调制波形;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PWMOCR1	-	-	FPWMOE5	FPWMOE4	FPWMOE3	FPWMOE2	FPWMOE1	FPWMOE0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] **FPWMOE[5:0]** – TOPPWM 上升/下降沿非交叠时间选择位

FPWMOE[5:0]	上升沿非交叠时间	下降沿非交叠时间
00 0000	1 个计数时钟周期	1 个计数时钟周期
00 0001	2 个计数时钟周期	2 个计数时钟周期
---	---	---
11 1110	63 个计数时钟周期	63 个计数时钟周期
11 1111	64 个计数时钟周期	64 个计数时钟周期

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PWMOCR2	-	-	-	-	PWMODB	PWMOD2	PWMOD1	PWMOD0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	1	0	0	0

BIT[3] **PWMODB** – TOPPWM/TOPPWM 匹配翻转时钟沿选择位

- 0: 占空比匹配时, 脉宽调制信号电平滞后半时钟周期翻转;
- 1: 占空比匹配时, 脉宽调制信号电平与计数时钟的有效沿同步翻转;

BIT[2:0] **PWMOD[2:0]** – TOPPWM/TOPPWM 占空比延展控制位

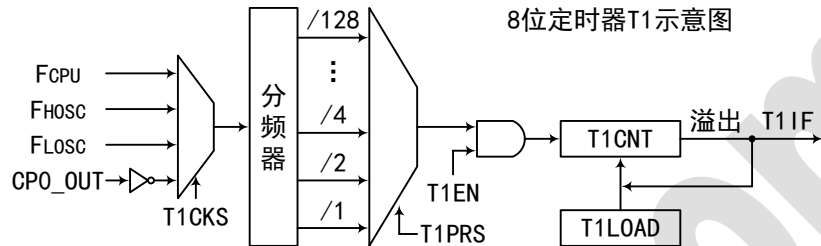
PWMOD[2:0]	每 8 个周期中延展周期选择
000	0 个周期延展
001	1 个周期(第 1 个周期)延展, 7 个周期(第 2, 3, 4, 5, 6, 7, 8 个周期)不延展
010	2 个周期(第 1, 5 个周期)延展, 6 个周期(第 2, 3, 4, 6, 7, 8 个周期)不延展
011	3 个周期(第 1, 3, 5 个周期)延展, 5 个周期(第 2, 4, 6, 7, 8 个周期)不延展
100	4 个周期(第 1, 3, 5, 7 个周期)延展, 4 个周期(第 2, 4, 6, 8 个周期)不延展
101	5 个周期(第 1, 2, 3, 5, 7 个周期)延展, 3 个周期(第 4, 6, 8 个周期)不延展
110	6 个周期(第 1, 2, 3, 5, 6, 7 个周期)延展, 2 个周期(第 4, 8 个周期)不延展
111	7 个周期(第 1, 2, 3, 4, 5, 6, 7 个周期)延展, 1 个周期(第 8 个周期)不延展



7.3 定时器 T1

定时器 T1 为 8 位定时/计数器，包含 1 个 8 位递减计数器、可编程预分频器、控制寄存器、8 位重载寄存器。

- ◇ 可通过预分频器设置时钟频率，可通过重载寄存器控制计数周期；
- ◇ 可实现比较器 CP0 输出信号 CP0_OUT 的下降沿计数功能；
- ◇ 支持溢出中断和溢出唤醒功能；



定时器 T1 的定时/计数功能与定时器 T0 完全相同。当选择比较器 CP0 的输出信号 CP0_OUT 作为时钟源时，T1 可实现对 CP0_OUT 下降沿的计数功能。

定时器 T1 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T1CR	T1EN	-	-	T1CKS1	T1CKS0	T1PRS2	T1PRS1	T1PRS0
R/W	R/W	-	-	R/W	R/W	R/W	R/W	R/W
初始值	0	-	-	0	0	0	0	0

BIT[7] **T1EN** – 定时器 T1 使能位

0: 关闭定时器 T1;

1: 开启定时器 T1;

BIT[4:3] **T1CKS[1:0]** – T1 时钟源选择位

T1CKS[1:0]	T1 时钟源
00	F _{CPU}
01	F _{HOSC}
10	F _{LOSC}
11	CP0_OUT 下降沿

BIT[2:0] **T1PRS[2:0]** – T1 时钟预分频比选择位

T1PRS[2:0]	T1 时钟预分频比
000	1 : 1
001	1 : 2
010	1 : 4
011	1 : 8
100	1 : 16



101	1 : 32
110	1 : 64
111	1 : 128

定时器 T1 计数器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T1CNT	T1CNT7	T1CNT6	T1CNT5	T1CNT4	T1CNT3	T1CNT2	T1CNT1	T1CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T1CNT[7:0] – T1 计数器，为可读写的递减计数器

定时器 T1 重载寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T1LOAD	T1LOAD7	T1LOAD6	T1LOAD5	T1LOAD4	T1LOAD3	T1LOAD2	T1LOAD1	T1LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T1LOAD[7:0] – T1 重载寄存器，用于设置 T1 的计数周期

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。

7.4 定时器 T2

定时器 T2 为 8 位定时/计数器，包含 1 个 8 位递减计数器、可编程预分频器、控制寄存器、8 位重载寄存器。

- ◇ 可通过预分频器设置时钟频率，可通过重载寄存器控制计数周期；
- ◇ 可实现 CP0 输出信号 CP0_OUT（或其去抖后信号 CP0TLV）的上升沿计数功能；
- ◇ 可实现外部 TC2 或内部 CP0_OUT/CP0TLV 等脉冲信号的高/低电平宽度（或周期宽度）的测量功能；
- ◇ 支持溢出中断和溢出唤醒功能；

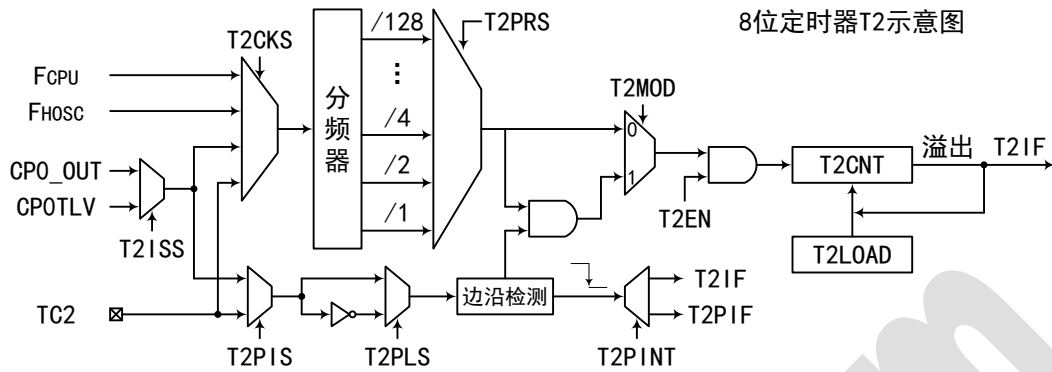
定时器 T2 具有 2 种工作模式：定时/计数模式和脉宽测量模式。

定时/计数模式下，T2 的定时功能与定时器 T0 相同，T2 的计数功能可对外部输入 TC2、内部信号 CP0_OUT 或 CP0TLV 的上升沿进行计数。

脉宽测量模式下，T2 可通过内部时钟对外部输入 TC2、内部信号 CP0_OUT 或 CP0TLV 的高/低电平进行脉宽计数，其中被测电平信号源可通过脉宽测量信号选择位 T2PIS 和 T2 内部信号选择位 T2ISS 选择，高/低电平可通过 T2PLS 选择。T2EN 置 1 使能 T2 后，被测电平产生触发沿时 T2 开始计数（若计数溢出则自动重载计数），直到被测电平产生停止沿时 T2EN 自动清 0，T2 停止计数，同时中断标志 T2PIF 将被置 1 触发 T2 捕捉中断（需 T2PINT=1，若 T2PINT=0 则 T2 捕捉中断无效，被测电平停止沿



产生时将置 1 中断标志 T2IF 触发定时器中断), 通过 T2CNT 的计数差值, 可获得被测电平的脉冲宽度。



注:

- 1、脉宽测量模式时, T2CNT 在测量完成后停止计数到再次触发测量计数期间, 都将保持原值; 若要更改则需重写 T2LOAD 使 T2CNT 重载, 或直接对 T2CNT 赋值;
- 2、内部信号 CPOTLV, 为外部输入 PTRIG 或内部 CPO 输出 CPO_OUT, 经 CPOTDB 去抖后的电平信号, 所以不能将 CPOTLV 的电平变化等同于 PTRIG 或 CPO_OUT 的电平变化。
- 3、被测电平产生停止沿时, 若 T2PINT=1 则将置 1 中断标志 T2PIF 触发 T2 捕捉中断 (适用于长脉宽的测量); 若 T2PINT=0 则将置 1 中断标志 T2IF 触发定时器中断 (适用于短脉宽的测量);

定时器 T2 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T2CR	T2EN	T2MOD	-	T2CKS1	T2CKS0	T2PRS2	T2PRS1	T2PRS0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
初始值	0	0	-	0	0	0	0	0

BIT[7] T2EN – 定时器 T2 使能位

- 0: 关闭定时器 T2;
- 1: 开启定时器 T2;

BIT[6] T2MOD – T2 工作模式选择位

- 0: 定时/计数模式;
- 1: 脉宽测量模式;

BIT[4:3] T2CKS[1:0] – T2 时钟源选择位

T2MOD	T2CKS[1:0]	T2 时钟源
T2MOD=0 定时/计数模式	00	F _{CPU}
	01	F _{HOSC}
	10	CPOTLV/CPO_OUT 上升沿
	11	TC2 上升沿
T2MOD=1 脉宽测量模式	00	F _{CPU}
	01	F _{HOSC}



	10	禁用
	11	禁用

BIT[2:0] **T2PRS[2:0]** – T2 时钟预分频比选择位

T2PRS[2:0]	T2 时钟预分频比
000	1 : 1
001	1 : 2
010	1 : 4
011	1 : 8
100	1 : 16
101	1 : 32
110	1 : 64
111	1 : 128

定时器 T2 计数器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T2CNT	T2CNT7	T2CNT6	T2CNT5	T2CNT4	T2CNT3	T2CNT2	T2CNT1	T2CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T2CNT[7:0]** – T2 计数器，为可读写的递减计数器

定时器 T2 重载寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T2LOAD	T2LOAD7	T2LOAD6	T2LOAD5	T2LOAD4	T2LOAD3	T2LOAD2	T2LOAD1	T2LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T2LOAD[7:0]** – T2 重载寄存器，用于设置 T2 的计数周期

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。

定时器 T2 脉冲控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T2EFR	-	T2PINT	T2PMOD	T2PCT1	T2PCT0	T2ISS	T2PIS	T2PLS
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	0	0	0	0	0	0	0

BIT[6] **T2PINT** – T2 中断模式选择位

0: T2 溢出和脉宽测量完成时，T2IF 均将置 1 而触发定时器中断；

1: T2 溢出，T2IF 置 1 触发定时器中断；脉宽测量完成，T2PIF 置 1 触发 T2 捕捉中断；



BIT[5] **T2PMOD** – T2 脉宽测量模式选择位

- 0: 测量脉冲单个高/低电平宽度（触发沿和停止沿为反向沿）；
- 1: 测量脉冲单个或多个周期宽度（触发沿和停止沿为同向沿）；

BIT[4:3] **T2PCT[1:0]** – T2 脉宽测量周期个数选择位（仅 T2PMOD=1 时有效）

T2PCT[1:0]	T2 脉冲测量周期个数
00	1 个
01	4 个
10	8 个
11	16 个

BIT[2] **T2ISS** – T2 计数或测量时的内部信号选择位

- 0: 内部信号为 CP0 的输出信号 CP0_OUT；
- 1: 内部信号为触发信号 CP0TLV（PTRIG 或 CP0_OUT 信号，经去抖反向生成的信号）；

BIT[1] **T2PIS** – T2 脉宽测量信号源选择位

- 0: 选择测量外部输入 TC2 的电平脉宽；
- 1: 选择测量由 T2ISS 所选内部信号的电平脉宽；

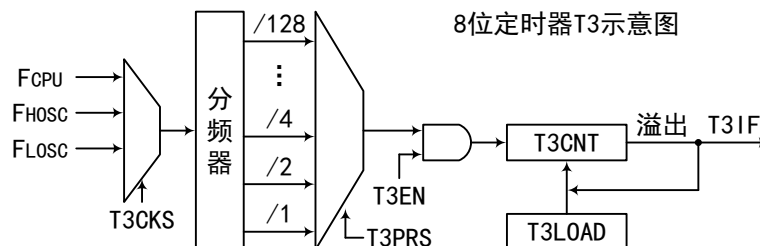
BIT[0] **T2PLS** – T2 测量高/低电平选择位

- 0: 测量低电平脉宽（T2PMOD=0: 下降沿开始计数，上升沿停止计数；T2PMOD=1: 下降沿开始计数，若干周期后的下降沿停止计数）；
- 1: 测量高电平脉宽（T2PMOD=0: 上升沿开始计数，下降沿停止计数；T2PMOD=1: 上升沿开始计数，若干周期后的上升沿停止计数）；

7.5 定时器 T3

定时器 T3 为 8 位定时器，包含 1 个 8 位递减计数器、可编程预分频器、控制寄存器、8 位重载寄存器。

- ◇ 可通过预分频器设置时钟频率，可通过重载寄存器控制计数周期；
- ◇ 支持溢出中断和溢出唤醒功能；



定时器 T3 的定时功能，与定时器 T0 完全相同。



定时器 T3 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T3CR	T3EN	保留	-	T3CKS1	T3CKS0	T3PRS2	T3PRS1	T3PRS0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
初始值	0	0	-	0	0	0	0	0

BIT[7] **T3EN** – 定时器 T3 使能位

0: 关闭定时器 T3;

1: 开启定时器 T3;

BIT[6] 保留位, 需固定写“0”

BIT[4:3] **T3CKS[1:0]** – T3 时钟源选择位

T3CKS[1:0]	T3 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	-

BIT[2:0] **T3PRS[2:0]** – T3 时钟预分频比选择位

T3PRS[2:0]	T3 时钟预分频比
000	1 : 1
001	1 : 2
010	1 : 4
011	1 : 8
100	1 : 16
101	1 : 32
110	1 : 64
111	1 : 128

定时器 T3 计数器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T3CNT	T3CNT7	T3CNT6	T3CNT5	T3CNT4	T3CNT3	T3CNT2	T3CNT1	T3CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T3CNT[7:0]** – T3 计数器, 为可读写的递减计数器

定时器 T3 重载寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T3LOAD	T3LOAD7	T3LOAD6	T3LOAD5	T3LOAD4	T3LOAD3	T3LOAD2	T3LOAD1	T3LOAD0



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T3LOAD[7:0]** – T3 重载寄存器，用于设置 T3 的计数周期

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。

Shomcu.com



8 模数转换器 ADC

8.1 ADC 概述

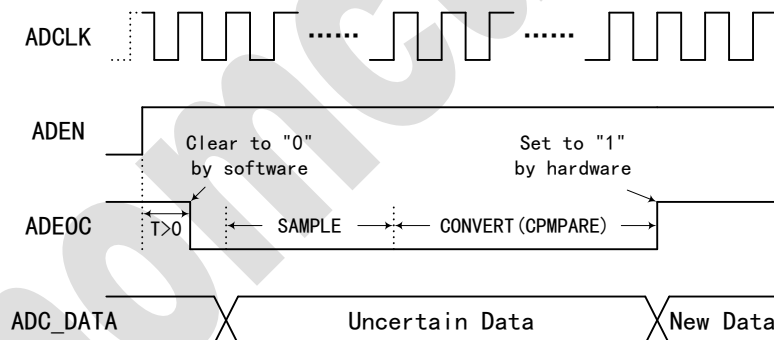
芯片内置 1 个 12 位高精度逐次逼近型的模数转换器 ADC。

- ◇ 14 路外部通道：AN0~AN13；2 路内部通道：GND、VDD/4；
- ◇ 参考电压可选：VDD、内部参考电压 V_{IR} (2V/3V/4V)、外部参考电压 V_{ER} (V_{ER1} 输入)；
- ◇ ADC 时钟：FHIRC 的 8/16/32/64/128/256/512/1024 分频；
- ◇ 支持零点校准；

ADC 模块可通过寄存器位 ADEN 开启，通过 ADCKS 选择转换时钟，通过 ADCHS 选择转换的模拟通道，通过 ADEOC 启动并标识 AD 转换状态。当 ADEOC 为 1 时写 0 将启动模数转换，转换完成后结果存入 ADRH/ADRL 中，ADEOC 自动置 1，同时中断标志 ADIF 置 1 触发 ADC 中断。

ADC 的采样 (SAMPLE) 时间可选择 4/8/15 个 ADCLK (即 ADC 时钟周期)，转换 (CONVERT) 时间固定为 12 个 ADCLK，一次 ADC 转换的时间为 16/20/27 个 ADCLK。

ADC 转换时序如下图所示：



注：

- 1、AD 转换过程中或 ADEN 未使能时，ADRH/ADRL 中的数据未知，应在 AD 转换完成且 ADEN 使能的情况下读取 AD 转换结果数据；
- 2、若选择内部参考电压 V_{IR} ，则需保证 $VDD > (V_{IR} + 0.5V)$ ，否则 V_{IR} 实际电压将降为 $(VDD - 0.5V)$ ；
- 3、使能 ADC 模块、或切换参考电压等操作后，需延时 (时间 $> 200 \mu s$) 以待电路稳定后才可启动 AD 转换；因采样保持电路的电容效应，切换输入通道后的前几次转换结果将会有偏差，建议舍弃；
- 4、AD 转换精度受参考电压精度的影响，且内部参考电压下的转换精度，比外部参考电压下略低 2 个 LSB 左右；
- 5、转换时钟越慢、采样时间越长，则越能过滤外部输入的波动，越能保证 AD 转换的精度；



8.2 ADC 相关寄存器

ADC 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADCRO	ADCHS3	ADCHS2	ADCHS1	ADCHS0	CPVIREN	ADVRS3	ADEOC	ADEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	0	0	0	1	0

BIT[7:4] **ADCHS[3:0]** – ADC 模拟输入通道选择位

ADCHS[3:0]	ADC 模拟输入通道
0000	AN0
0001	AN1
0010	AN2
0011	AN3
0100	AN4
0101	AN5
0110	AN6
0111	AN7
1000	AN8
1001	AN9
1010	AN10
1011	AN11
1100	AN12
1101	AN13
1110	GND
1111	VDD/4

BIT[3] **CPVIREN** – 比较器内部参考电压使能位

- 0: 关闭比较器内部参考电压;
- 1: 使能比较器内部参考电压;

注: CPVIREN=1 时, 比较器的参考电压才可通过 CPVRS 和 ADVRS[1:0] 选择 V_{IR} 。

BIT[2] **ADVRS3** – ADC 参考电压选择位

- 0: ADC 参考电压由 ADVRS[2:0] 决定;
- 1: ADC 参考电压为 VDD;

BIT[1] **ADEOC** – AD 转换控制位

- 0: AD 转换中, 完成后自动置 1;
- 1: 转换未开始或已完成, 写 0 开始 AD 转换;



BIT[0] **ADEN** – ADC 使能位
 0: 关闭 ADC;
 1: 开启 ADC;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADCR1	ADRSEL	ADCKS2	ADCKS1	ADCKS0	–	–	ADSPS1	ADSPS0
R/W	R/W	R/W	R/W	R/W	–	–	R/W	R/W
初始值	0	0	0	0	–	–	1	1

BIT[7] **ADRSEL** – ADC 转换结果数据格式选择位
 0: ADC 转换结果为 12 位数据, 高 8 位存入 ADRH[7:0]、低 4 位存入 ADRL[3:0];
 1: ADC 转换结果为 12 位数据, 高 4 位存入 ADRH[3:0]、低 8 位存入 ADRL[7:0];

BIT[6:4] **ADCKS[2:0]** – ADC 转换时钟选择位

ADCKS[2:0]	ADC 转换时钟 F_{ADC}
000	$F_{HIRC}/8$
001	$F_{HIRC}/16$
010	$F_{HIRC}/32$
011	$F_{HIRC}/64$
100	$F_{HIRC}/128$
101	$F_{HIRC}/256$
110	$F_{HIRC}/512$
111	$F_{HIRC}/1024$

BIT[1:0] **ADSPS[1:0]** – ADC 采样时间选择位

ADSPS[1:0]	ADC 采样时间
00	保留
01	4 个 ADCLK
10	8 个 ADCLK
11	15 个 ADCLK

扩展功能控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
MEFCR	–	–	–	–	–	ADVRS2	ADVRS1	ADVRS0
R/W	–	–	–	–	–	R/W	R/W	R/W
初始值	–	–	–	–	–	0	0	0

BIT[2] **ADVRS2** – ADC 参考电压选择位
 0: ADC 参考电压由 ADVRS[1:0]决定;
 1: ADC 参考电压为外部参考电压 V_{ER} (端口 $VERI$ 输入电压);

BIT[1:0] **ADVRS[1:0]** – ADC/CMP 内部参考电压选择位



ADVRS[1:0]	ADC/CMP 内部参考电压
00	内部 4.0V
01	内部 3.0V
10	内部 2.0V
11	禁用 (内部为高阻)

ADC 的参考电压选择如下表所示:

ADVRS3	ADVRS2	ADVRS[1:0]	ADC 参考电压
0	0	00	内部参考电压 V_{IR} (4.0V)
		01	内部参考电压 V_{IR} (3.0V)
		10	内部参考电压 V_{IR} (2.0V)
		11	禁用
	1	X	外部参考电压 V_{ER} (V_{ERI} 输入)
1	X	X	VDD

ADC 转换结果寄存器

ADRSEL=0:

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADRH	ADR11	ADR10	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4
R/W	R	R	R	R	R	R	R	R
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **ADR[11:4]** – 12 位 ADC 转换结果高 8 位

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADRL	-	-	-	-	ADR3	ADR2	ADR1	ADR0
R/W	-	-	-	-	R	R	R	R
初始值	-	-	-	-	X	X	X	X

BIT[3:0] **ADR[3:0]** – 12 位 ADC 转换结果低 4 位

ADRSEL=1:

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADRH	-	-	-	-	ADR11	ADR10	ADR9	ADR8
R/W	-	-	-	-	R	R	R	R
初始值	-	-	-	-	X	X	X	X

BIT[3:0] **ADR[11:8]** – 12 位 ADC 转换结果高 4 位

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADRL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0



R/W	R	R	R	R	R	R	R	R
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **ADR[7:0]** – 12 位 ADC 转换结果低 8 位

ADC 零点偏移修调控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OSADJCR	OSADJEN	-	OSADJTD	OSADJT4	OSADJT3	OSADJT2	OSADJT1	OSADJT0
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	-	0	0	0	0	0	0

BIT[7] **OSADJEN** – ADC 零点偏移修调使能位

- 0: ADC 零点偏移修调无效;
- 1: ADC 零点偏移修调有效;

BIT[5] **OSADJTD** – ADC 零点偏移修调方向选择位

- 0: 负向修调, 即根据修调电压减小转换值 (转换结果大于理论值时应选择负向修调);
- 1: 正向修调, 即根据修调电压增加转换值 (转换结果小于理论值时应选择正向修调);

BIT[4:0] **OSADJT[4:0]** – ADC 零点偏移修调电压选择位

OSADJT[4:0]	修调电压 (典型值)
0 0000	0
0 0001	$1 \times V_{REF}/4096$
0 0010	$2 \times V_{REF}/4096$
---	---
1 1110	$30 \times V_{REF}/4096$
1 1111	$31 \times V_{REF}/4096$

8.3 ADC 操作步骤

模数转换操作步骤:

- (1) 设置相应端口为输入端口, 关闭端口的内部上/下拉电阻;
- (2) 通过端口数模控制寄存器, 关闭相应端口的数字 I/O 功能;
- (3) 若转换时钟可选, 则设置 ADCKS, 选择适当的转换时钟;
- (4) 若采样时间可选, 则设置 ADSPS, 选择适当的采样时间;
- (5) 若参考电压可选, 则设置 ADVRS, 选择适当的参考电压;
- (6) 若数据格式可选, 则设置 ADRSEL, 选择 ADC 转换结果的数据格式;
- (7) ADEN 置 1, 使能 ADC 模块;
- (8) 设置 ADCHS, 选择 ADC 转换通道;
- (9) 延时等待电路稳定后, ADEOC 写 0, 启动 AD 转换;



- (10) 等待 ADEOC 硬件置 1（或利用 ADC 中断）；
- (11) 读取 ADC 转换结果（ADRH/ADRL）；
- (12) 重复执行（8）~（11），对不同的通道进行转换或对同一通道进行多次转换；

8.4 ADC 零点偏移修调流程

- (1) 设置 ADC 输入通道为 GND，设置 ADC 时钟、采样时间等参数，设置 OSADJEN=1；
- (2) 设置 OSADJTD=0、OSADJT=00H，进行 ADC 转换：
 - ◇ 若转换结果为 0，则执行（4）；
 - ◇ 若转换结果非 0，则执行（3）；
- (3) OSADJT 自加 1 后进行 ADC 转换：
 - ◇ 若转换结果为 0，则跳至（6）；
 - ◇ 若转换结果非 0，则循环执行（3），直到结果为 0 或 OSADJT=1FH 后，跳至（6）；
- (4) 设置 OSADJTD=1、OSADJT=1FH，进行 ADC 转换：
 - ◇ 若转换结果为 0，则跳至（6）；
 - ◇ 若转换结果非 0，则执行（5）；
- (5) OSADJT 自减 1 后进行 ADC 转换：
 - ◇ 若转换结果为 0，则跳至（6）；
 - ◇ 若转换结果非 0，则循环执行（5），直到结果为 0 或 OSADJT=00H 后，跳至（6）；
- (6) OSADJTD 及 OSADJT[4:0]的值即为零点偏移最佳修调结果，修调流程结束，后续 ADC 工作时直接应用，无需再次修调。



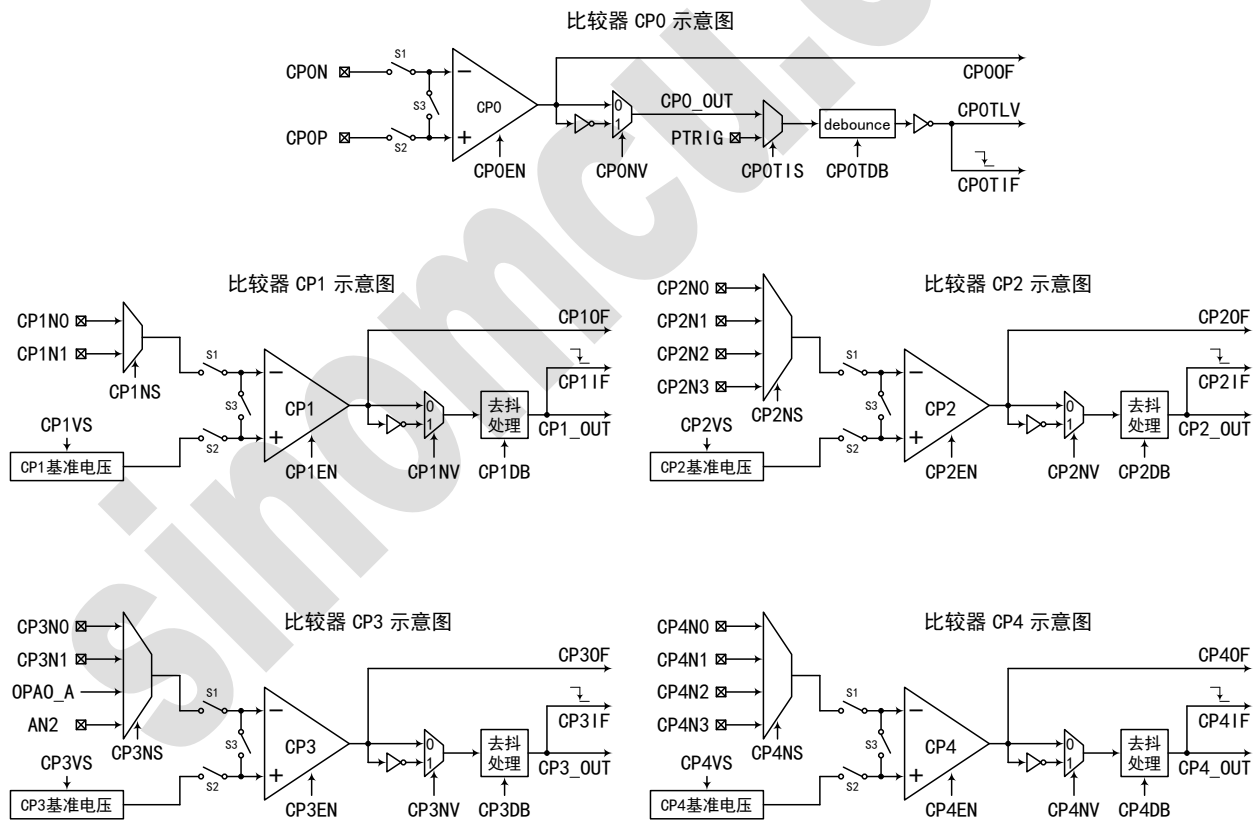
9 模拟比较器 CMP

9.1 CMP 概述

芯片内置 5 个模拟比较器 CP0~CP4，共模输入范围均为 0~(VDD-1.4V)，输入端失调电压均可通过自消除寄存器进行调校，精度为 ±2mV，输出均无回滞。

比较器 CP0 的正/负端输入均为外部输入电压。CP0 输出信号 CP0_OUT 经去抖后的信号 CP0TLV 的下降沿可触发 CP0 触发中断，还可通过定时器 T1 或 T2 对 CP0_OUT 下降沿进行计数。

比较器 CP1~CP4 的负端输入为外部输入电压（均支持多路通道选择），正端输入为参考电压经内部分压后产生的基准电压（参考电压可选 VDD、或内部参考电压 V_{IR}）。CP1~CP4 的输出均支持取反和去抖处理，经过使能/禁止取反和去抖处理后的输出信号 CP1_OUT~CP4_OUT，其下降沿将触发比较器中断。





9.2 CMP 相关寄存器

比较器控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CPCRO	-	CP3NS1	CP3NS0	CP1NS	CP3EN	CP2EN	CP1EN	CPOEN
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	0	0	0	0	0	0	0

BIT[6:5] **CP3NS[1:0]** – CP3 负端输入选择位

CP3NS[1:0]	CP3 负端输入
00	外部 CP3N0 输入
01	外部 CP3N1 输入
10	内部 OPA0_A 信号
11	外部 AN2 输入

BIT[4] **CP1NS** – CP1 负端输入选择位

- 0: CP1 负端输入为端口 CP1N0 输入电压;
- 1: CP1 负端输入为端口 CP1N1 输入电压;

BIT[3] **CP3EN** – 比较器 CP3 使能位

- 0: 关闭 CP3;
- 1: 开启 CP3;

BIT[2] **CP2EN** – 比较器 CP2 使能位

- 0: 关闭 CP2;
- 1: 开启 CP2;

BIT[1] **CP1EN** – 比较器 CP1 使能位

- 0: 关闭 CP1;
- 1: 开启 CP1;

BIT[0] **CPOEN** – 比较器 CP0 使能位

- 0: 关闭 CP0;
- 1: 开启 CP0;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CPCR1	-	-	-	CP4EN	CP4NS1	CP4NS0	CP2NS1	CP2NS0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	-	-	-	0	0	0	0	0

BIT[4] **CP4EN** – 比较器 CP4 使能位

- 0: 关闭 CP4;
- 1: 开启 CP4;



BIT[3:2] CP4NS[1:0] – CP4 负端输入选择位

CP4NS[1:0]	CP4 负端输入
00	外部 CP4N0 输入
01	外部 CP4N1 输入
10	外部 CP4N2 输入
11	外部 CP4N3 输入

BIT[1:0] CP2NS[1:0] – CP2 负端输入选择位

CP2NS[1:0]	CP2 负端输入
00	外部 CP2N0 输入
01	外部 CP2N1 输入
10	外部 CP2N2 输入
11	外部 CP2N3 输入

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CPCR2	-	-	-	CP4NV	CP3NV	CP2NV	CP1NV	CP0NV
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	-	-	-	0	0	0	0	0

BIT[4:0] CPnNV – 比较器 CPn 输出取反控制位 (n=4-0)

- 0: CPn 输出为比较结果的正向电平;
1: CPn 输出为比较结果的反向电平;

比较器状态寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CPSR	-	-	-	CP40F	CP30F	CP20F	CP10F	CP00F
R/W	-	-	-	R	R	R	R	R
初始值	-	-	-	X	X	X	X	X

BIT[4:0] CPnOF – CPn 比较结果状态标志位 (n=4-0)

- 0: CPn 比较结果为低, 即比较器正端输入电压低于负端输入电压; 或 CPn 关闭;
1: CPn 比较结果为高, 即比较器正端输入电压高于负端输入电压;

比较器电压寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CPVR0	CPVRS	CP2VS2	CP2VS1	CP2VS0	CP1VS3	CP1VS2	CP1VS1	CP1VS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] CPVRS – 比较器参考电压选择位

- 0: 比较器参考电压为 VDD 电压;
1: 比较器参考电压由 ADVRS[1:0] 决定 (仅 CPVIREN=1 时有效);



CMP 参考电压选择如下表所示:

CPVRS	ADVRS2	ADVRS[1:0]	ADC 参考电压
0	X	X	VDD
1	X	00	内部参考电压 V_{IR} (4.0V)
		01	内部参考电压 V_{IR} (3.0V)
		10	内部参考电压 V_{IR} (2.0V)
		11	禁用

注: 若选择内部参考电压 V_{IR} , 则需保证 $VDD > (V_{IR} + 0.5V)$, 否则 V_{IR} 实际电压将降为 $(VDD - 0.5V)$ 。

BIT[6:4] CP2VS[2:0] – CP2 正端基准电压选择位

CP2VS[2:0]	CP2 正端基准电压	
	CPVRS=0	CPVRS=1
000	$0.05 \times VDD$	$0.0625 \times V_{IR}$
001	$0.10 \times VDD$	$0.125 \times V_{IR}$
010	$0.20 \times VDD$	$0.25 \times V_{IR}$
011	$0.30 \times VDD$	$0.375 \times V_{IR}$
100	$0.40 \times VDD$	$0.5 \times V_{IR}$
101	$0.50 \times VDD$	$0.625 \times V_{IR}$
110	$0.60 \times VDD$	$0.75 \times V_{IR}$
111	$0.70 \times VDD$	$0.875 \times V_{IR}$

BIT[3:0] CP1VS[3:0] – CP1 正端基准电压选择位

CP1VS[3:0]	CP1 正端基准电压	
	CPVRS=0	CPVRS=1
0000	$0.34 \times VDD$	$0.425 \times V_{IR}$
0001	$0.36 \times VDD$	$0.45 \times V_{IR}$
0010	$0.38 \times VDD$	$0.475 \times V_{IR}$
0011	$0.40 \times VDD$	$0.5 \times V_{IR}$
0100	$0.42 \times VDD$	$0.525 \times V_{IR}$
0101	$0.44 \times VDD$	$0.55 \times V_{IR}$
0110	$0.46 \times VDD$	$0.575 \times V_{IR}$
0111	$0.48 \times VDD$	$0.6 \times V_{IR}$
1000	$0.50 \times VDD$	$0.625 \times V_{IR}$
1001	$0.52 \times VDD$	$0.65 \times V_{IR}$
1010	$0.54 \times VDD$	$0.675 \times V_{IR}$
1011	$0.56 \times VDD$	$0.7 \times V_{IR}$
1100	$0.58 \times VDD$	$0.725 \times V_{IR}$
1101	$0.60 \times VDD$	$0.75 \times V_{IR}$



1110	$0.62 \times VDD$	$0.775 \times V_{IR}$
1111	$0.64 \times VDD$	$0.8 \times V_{IR}$

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CPVR1	CP4VS2	CP4VS1	CP4VS0	CP3VS4	CP3VS3	CP3VS2	CP3VS1	CP3VS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:5] CP4VS[2:0] – CP4 正端基准电压选择位

CP4VS[2:0]	CP4 正端基准电压	
	CPVRS=0	CPVRS=1
000	$0.02 \times VDD$	$0.025 \times V_{IR}$
001	$0.03 \times VDD$	$0.0375 \times V_{IR}$
010	$0.05 \times VDD$	$0.0625 \times V_{IR}$
011	$0.10 \times VDD$	$0.125 \times V_{IR}$
100	$0.20 \times VDD$	$0.25 \times V_{IR}$
101	$0.30 \times VDD$	$0.375 \times V_{IR}$
110	$0.40 \times VDD$	$0.5 \times V_{IR}$
111	$0.50 \times VDD$	$0.625 \times V_{IR}$

BIT[4:0] CP3VS[4:0] – CP31 正端基准电压选择位

CP3VS[4:0]	CP3 正端基准电压	
	CPVRS=0	CPVRS=1
0 0000	$0.06 \times VDD$	$0.075 \times V_{IR}$
0 0001	$0.10 \times VDD$	$0.125 \times V_{IR}$
0 0010	$0.12 \times VDD$	$0.15 \times V_{IR}$
0 0011	$0.14 \times VDD$	$0.175 \times V_{IR}$
0 0100	$0.16 \times VDD$	$0.2 \times V_{IR}$
0 0101	$0.18 \times VDD$	$0.225 \times V_{IR}$
0 0110	$0.20 \times VDD$	$0.25 \times V_{IR}$
0 0111	$0.22 \times VDD$	$0.275 \times V_{IR}$
0 1000	$0.24 \times VDD$	$0.3 \times V_{IR}$
0 1001	$0.26 \times VDD$	$0.325 \times V_{IR}$
0 1010	$0.28 \times VDD$	$0.35 \times V_{IR}$
0 1011	$0.30 \times VDD$	$0.375 \times V_{IR}$
0 1100	$0.32 \times VDD$	$0.4 \times V_{IR}$
0 1101	$0.34 \times VDD$	$0.425 \times V_{IR}$
0 1110	$0.36 \times VDD$	$0.45 \times V_{IR}$
0 1111	$0.38 \times VDD$	$0.475 \times V_{IR}$
1 0000	$0.40 \times VDD$	$0.5 \times V_{IR}$



1 0001	$0.42 \times VDD$	$0.525 \times V_{IR}$
1 0010	$0.44 \times VDD$	$0.55 \times V_{IR}$
1 0011	$0.46 \times VDD$	$0.575 \times V_{IR}$
1 0100	$0.48 \times VDD$	$0.6 \times V_{IR}$
1 0101	$0.50 \times VDD$	$0.625 \times V_{IR}$
1 0110	$0.52 \times VDD$	$0.65 \times V_{IR}$
1 0111	$0.54 \times VDD$	$0.675 \times V_{IR}$
1 1000	$0.56 \times VDD$	$0.7 \times V_{IR}$
1 1001	$0.58 \times VDD$	$0.725 \times V_{IR}$
1 1010	$0.60 \times VDD$	$0.75 \times V_{IR}$
1 1011	$0.62 \times VDD$	$0.775 \times V_{IR}$
1 1100	$0.64 \times VDD$	$0.8 \times V_{IR}$
1 1101	$0.66 \times VDD$	$0.825 \times V_{IR}$
1 1110	$0.68 \times VDD$	$0.85 \times V_{IR}$
1 1111	$0.70 \times VDD$	$0.875 \times V_{IR}$

CP0 触发去抖寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CP0TDB	CP0TIS	CP0TDB6	CP0TDB5	CP0TDB4	CP0TDB3	CP0TDB2	CP0TDB1	CP0TDB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] **CP0TIS** – 触发中断输入信号源选择位
 0: 外部输入 PTRIG 作为 CP0 触发中断输入信号;
 1: CP0 输出 CP0_OUT 作为 CP0 触发中断输入信号;

BIT[6:0] **CP0TDB[6:0]** – CP0 输出去抖时间选择位, 去抖时间为 CP0TDB[6:0] 个 F_{HOSC} 时钟周期

比较器输出去抖寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CPDB0	CP2DB3	CP2DB2	CP2DB1	CP2DB0	CP1DB3	CP1DB2	CP1DB1	CP1DB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:4] **CP2DB[3:0]** – CP2 输出去抖时间选择位

CP2DB[3:0]	CP2 输出去抖时间
0000	0 (不去抖)
0001	1 个 F_{HOSC} 时钟周期
0010	2 个 F_{HOSC} 时钟周期
---	---
1111	15 个 F_{HOSC} 时钟周期



BIT[3:0] **CP1DB[3:0]** – CP1 输出去抖时间选择位

CP1DB[3:0]	CP1 输出去抖时间
0000	0 (不去抖)
0001	1 个 F _{HOSC} 时钟周期
0010	2 个 F _{HOSC} 时钟周期
---	---
1111	15 个 F _{HOSC} 时钟周期

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CPDB1	CP4DB3	CP4DB2	CP4DB1	CP4DB0	CP3DB3	CP3DB2	CP3DB1	CP3DB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:4] **CP4DB[3:0]** – CP4 输出去抖时间选择位

CP4DB[3:0]	CP4 输出去抖时间
0000	0 (不去抖)
0001	1 个 F _{HOSC} 时钟周期
0010	2 个 F _{HOSC} 时钟周期
---	---
1111	15 个 F _{HOSC} 时钟周期

BIT[3:0] **CP3DB[3:0]** – CP3 输出去抖时间选择位

CP3DB[3:0]	CP3 输出去抖时间
0000	0 (不去抖)
0001	1 个 F _{HOSC} 时钟周期
0010	2 个 F _{HOSC} 时钟周期
---	---
1111	15 个 F _{HOSC} 时钟周期

比较器自消除寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CPOCLR	CPOCLRE	CPOCLRS	CPOCLR5	CPOCLR4	CPOCLR3	CPOCLR2	CPOCLR1	CPOCLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	1	0	0	0	0	0

BIT[7] **CPOCLRE** – CP0 自消除模式使能位

- 0: 关闭 CP0 自消除模式;
- 1: 使能 CP0 自消除模式;



BIT[6] **CP0CLRS** – CP0 自消除参考源选择位
 0: CP0 自消除参考源为其负端输入;
 1: CP0 自消除参考源为其正端输入;

BIT[5:0] **CP0CLR[5:0]** – CP0 自消除校准位

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CP1CLR	CP1CLRE	CP1CLRS	CP1CLR5	CP1CLR4	CP1CLR3	CP1CLR2	CP1CLR1	CP1CLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	1	0	0	0	0	0

BIT[7] **CP1CLRE** – CP1 自消除模式使能位
 0: 关闭 CP1 自消除模式;
 1: 使能 CP1 自消除模式;

BIT[6] **CP1CLRS** – CP1 自消除参考源选择位
 0: CP1 自消除参考源为其负端输入;
 1: CP1 自消除参考源为其正端输入;

BIT[5:0] **CP1CLR[5:0]** – CP1 自消除校准位

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CP2CLR	CP2CLRE	CP2CLRS	CP2CLR5	CP2CLR4	CP2CLR3	CP2CLR2	CP2CLR1	CP2CLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	1	0	0	0	0	0

BIT[7] **CP2CLRE** – CP2 自消除模式使能位
 0: 关闭 CP2 自消除模式;
 1: 使能 CP2 自消除模式;

BIT[6] **CP2CLRS** – CP2 自消除参考源选择位
 0: CP2 自消除参考源为其负端输入;
 1: CP2 自消除参考源为其正端输入;

BIT[5:0] **CP2CLR[5:0]** – CP2 自消除校准位

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CP3CLR	CP3CLRE	CP3CLRS	CP3CLR5	CP3CLR4	CP3CLR3	CP3CLR2	CP3CLR1	CP3CLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	1	0	0	0	0	0

BIT[7] **CP3CLRE** – CP3 自消除模式使能位
 0: 关闭 CP3 自消除模式;
 1: 使能 CP3 自消除模式;



BIT[6] **CP3CLRS** – CP3 自消除参考源选择位
 0: CP3 自消除参考源为其负端输入;
 1: CP3 自消除参考源为其正端输入;

BIT[5:0] **CP3CLR[5:0]** – CP3 自消除校准位

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
CP4CLR	CP4CLRE	CP4CLRS	CP4CLR5	CP4CLR4	CP4CLR3	CP4CLR2	CP4CLR1	CP4CLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	1	0	0	0	0	0

BIT[7] **CP4CLRE** – CP4 自消除模式使能位
 0: 关闭 CP4 自消除模式;
 1: 使能 CP4 自消除模式;

BIT[6] **CP4CLRS** – CP4 自消除参考源选择位
 0: CP4 自消除参考源为其负端输入;
 1: CP4 自消除参考源为其正端输入;

BIT[5:0] **CP4CLR[5:0]** – CP4 自消除校准位

9.3 CMP 失调电压调校流程

- (1) 设置 CPnCLRE=1, 使能失调电压自消除模式;
- (2) 设置 CPnCLRS, 选择比较器自消除参考源;
- (3) 设置 CPnCLR[5:0]=00H, 等待至少 100 μ s 后读取标志位 CPnOF;
- (4) CPnCLR[5:0]自加 1, 等待至少 100 μ s 后重新读取 CPnOF:
 - ◇ 若 CPnOF 发生翻转, 则表示已调校到位;
 - ◇ 若 CPnCLR[5:0]=3FH 而 CPnOF 仍未翻转, 则表示共模输入超出范围, 调校失败;
- (5) 循环执行 (4), 直到调校到位或调校失败;
- (6) 判断当前调校输出方向:
 - ◇ 若 CPnCLR[5]=1, 则 CPnCLR[4:0]保持当前值不变;
 - ◇ 若 CPnCLR[5]=0, 则 CPnCLR[4:0]减 1 作为校准值;
- (7) 设置 CPnCLRE=0, 关闭自消除模式;

注: 自消除参考源应选择电压较为稳定的一路输入端, 且参考源输入端的正、负极性不影响调校效果。



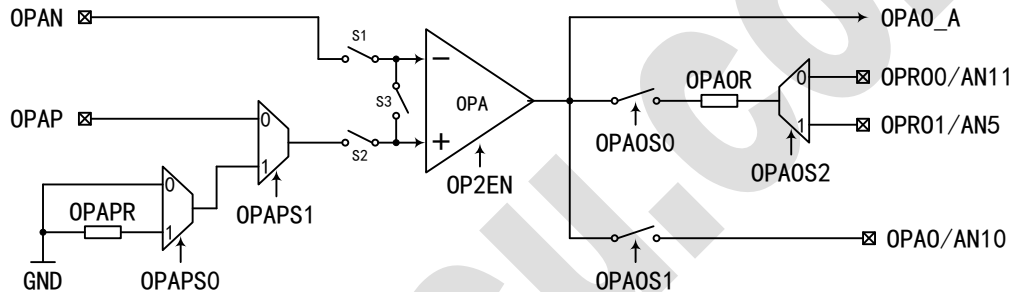
10 运算放大器 OPA

10.1 OPA 概述

芯片内置 1 个运算放大器 OPA，开环放大倍数为 60dB，共模输入范围为 0~ (VDD-1.4V)，输入端失调电压可通过自消除寄存器进行调校，精度为 $\pm 2\text{mV}$ 。

OPA 的正/负端输入和输出信号均可通过控制寄存器 OPACR 选择连接至端口或连接内部电路。OPA 的输出信号可通过 ADC 模块的通道选择进行 AD 转换，且可选择是否连接至比较器 CP3 的负端输入。

OPA 示意图如下，其中电阻 OPAPR 典型值为 $6\text{K}\Omega$ ，OPAOR 典型值为 $60\text{K}\Omega$ 。



10.2 OPA 相关寄存器

OPA 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OPACR	OPAEN	OPAOF	-	OPAOS2	OPAOS1	OPAOS0	OPAPS1	OPAPS0
R/W	R/W	R	-	R/W	R/W	R/W	R/W	R/W
初始值	0	X	-	0	0	0	0	0

BIT[7] **OPAEN** – 运算放大器 OPA 使能位

- 0: 关闭 OPA;
- 1: 开启 OPA;

BIT[6] **OPAOF** – OPA 输出状态标志位 (仅用于 OPA 自消除模式)

- 0: OPA 输出为低, 即 OPA 正端输入电压低于负端输入电压; 或 OPA 关闭;
- 1: OPA 输出为高, 即 OPA 正端输入电压高于负端输入电压;

BIT[4] **OPAOS2** – OPA 带电阻输出端口选择位

- 0: OPA 带电阻输出至端口 OPRO0 (P06);
- 1: OPA 带电阻输出至端口 OPRO1 (P12);



BIT[3] **OPAOS1** – OPA 端口输出控制位
 0: OPA 输出端不可直接连至端口;
 1: OPA 输出端可直接连至端口;

BIT[2] **OPAOS0** – OPA 输出电路串接电阻控制位
 0: OPA 输出端不可通过电阻连至端口;
 1: OPA 输出端可通过电阻连至端口;

BIT[1:0] **OPAPS[1:0]** – OPA 正端输入选择位

OPAPS[1:0]	OPA 正端输入
0X	外部 OPAP 输入
10	GND
11	内部接地电阻 OPAPR

OPA 自消除寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OPACLR	OPACLRE	OPACLRS	OPACLR5	OPACLR4	OPACLR3	OPACLR2	OPACLR1	OPACLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	1	0	0	0	0	0

BIT[7] **OPACLRE** – OPA 自消除模式使能位
 0: 关闭 OPA 自消除模式;
 1: 使能 OPA 自消除模式;

BIT[6] **OPACLRS** – OPA 自消除参考源选择位
 0: OPA 自消除参考源为其负端输入;
 1: OPA 自消除参考源为其正端输入;

注: 自消除参考源应选择电压较为稳定的一路输入端, 且参考源输入端的正、负极性不影响调校效果。

BIT[5:0] **OPACLR[5:0]** – OPA 自消除校准位

10.3 OPA 失调电压调校流程

- (1) 设置 OPACLRE=1, 使能失调电压自消除模式;
- (2) 设置 OPACLRS, 选择 OPA 自消除参考源;
- (3) 设置 OPACLR[5:0]=00H, 等待至少 100 μ s 后读取标志位 OPAOF;
- (4) OPACLR[5:0]自加 1, 等待至少 100 μ s 后重新读取 OPAOF:
 - ◇ 若 OPAOF 发生翻转, 则表示已调校到位;
 - ◇ 若 OPACLR[5:0]=3FH 而 OPAOF 仍未翻转, 则表示共模输入超出范围, 调校失败;



- (5) 循环执行 (4)，直到调校到位或调校失败；
- (6) 判断当前调校输出方向：
 - ✧ 若 $OPACLR[5]=1$ ，则 $OPACLR[4:0]$ 保持当前值不变；
 - ✧ 若 $OPACLR[5]=0$ ，则 $OPACLR[4:0]$ 减 1 作为校准值；
- (7) 设置 $OPACLRE=0$ ，关闭自消除模式；

Sinomcu.com



11 低电压检测 LVD

芯片内置低电压检测模块 LVD，可通过寄存器位 LVDEN 开启，通过 LVDVS 选择电压检测阈值。当 VDD 电压降至电压检测阈值以下时检测状态标志位 LVDF 将被置 1，同时中断标志 LVDIF 置 1 触发 LVD 中断；因 LVD 电路的回滞特性（回滞电压典型值为 6%），VDD 电压需恢复至电压检测阈值+6%后 LVDF 才被清 0（但中断标志不会自动清 0）。

LVD 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
LVDCR	LVDEN	-	-	LVDVS	-	-	-	LVDF
R/W	R/W	-	-	R/W	-	-	-	R
初始值	0	-	-	0	-	-	-	X

BIT[7] **LVDEN** – 低电压检测 LVD 使能位

- 0: 关闭 LVD;
- 1: 开启 LVD;

BIT[4] **LVDVS** – LVD 电压检测阈值选择位

- 0: LVD 电压检测阈值为 3.3V;
- 1: LVD 电压检测阈值为 4.2V;

BIT[0] **LVDF** – LVD 检测状态标志位

- 0: VDD 电压高于电压检测阈值，或 LVD 关闭;
- 1: VDD 电压低于电压检测阈值;



12 总线通讯 IIC

12.1 IIC 概述

芯片内置 1 个 IIC 总线通讯模块，支持 7 位地址编码从机模式的 IIC 总线通讯。IIC 总线通讯接口为时钟线 SCL 和数据线 SDA 的双向两线接口，IIC 使能后复用的 I/O 端口用作 SCL/SDA，此时为输入/开漏输出口，输出时其内部上拉电阻控制位依然有效，可选择内部或外接合适的上拉电阻，以匹配主机传输的通讯速率（最高支持 400Kbps）。

注：IIC 模块的时钟源为系统高频时钟 F_{HOSC} ，仅当系统高频时钟源工作时，IIC 才可正常工作。

12.2 IIC 数据传输

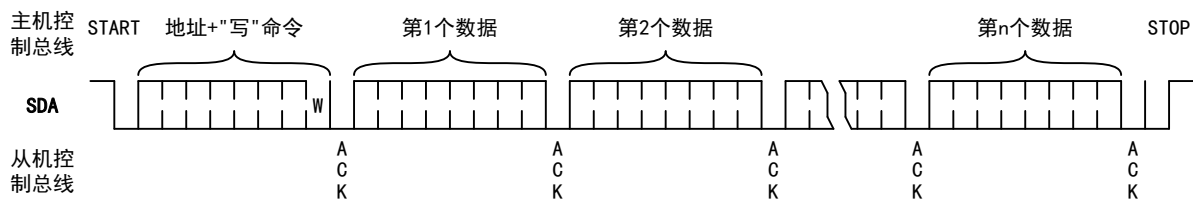
总线空闲时，数据线 SDA 和时钟线 SCL 均为高电平。SDA 电平在 SCL 高电平期间由高变低的下降沿表示起始信号 START，而 SDA 电平在 SCL 高电平期间由低变高的上升沿则表示停止信号 STOP。START/STOP 信号以及 SCL 上的时钟信号均由主机发送，而数据线 SDA 上的数据则由主从双方同步于 SCL 时钟进行单向传输。数据传输时，SDA 电平在 SCL 高电平期间必须保持稳定，只有在 SCL 为低电平时，SDA 电平才允许变化。

一帧数据传输以一个起始信号 START 开始，以一个停止信号 STOP 结束，期间总线不被释放。

每一帧数据传输时需先由主机发送一个以 7 位从机地址和 1 位读/写命令组成的控制字节，再由主机或从机发送一个或多个数据字节。一个完整字节的传输需 9 个时钟，前 8 个时钟传输 8 位字节内容（最高位最先传输），第 9 个时钟则为应答时钟，此时 SDA 上的电平即为接收方返回的应答信号，低电平表示应答（ACK），高电平表示非应答（NACK）。

主机到从机的数据传输

主机先发送起始信号 START，再发送一个包含“写”命令的控制字节，从机返回一个 ACK，然后主机开始发送数据字节，从机接收完每一个数据字节后均返回一个 ACK，主机在接收到最后一个字节从机返回的 ACK 后，发送停止信号 STOP 结束本次数据传输。

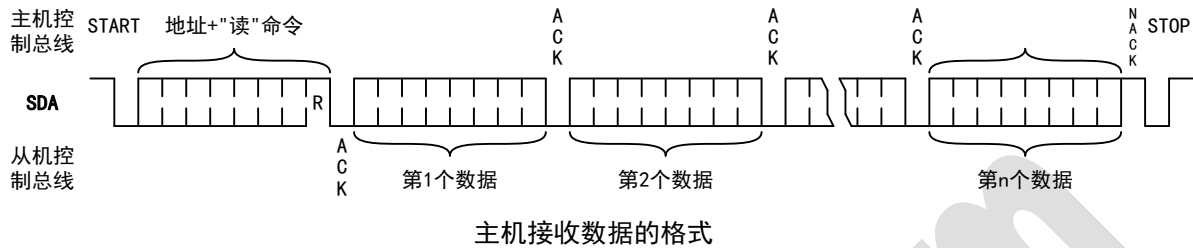


主机发送数据的格式



从机到主机的数据传输

主机先发送起始信号 START，再发送一个包含“读”命令的控制字节，从机返回一个 ACK，然后主机开始接收从机发送的数据字节，并在接收完每一个数据字节后均返回一个 ACK，当主机不再接收数据时则在接收完从机上一个字节后返回一个 NACK，接着发送停止信号 STOP 结束本次数据传输。



12.3 IIC 工作模式

芯片 IIC 总线通讯模块，仅支持 7 位地址编码的从机模式。当在总线上检测到 IIC 起始信号后开始接收主机发送的 7 位地址信息和 1 位读/写命令信息，若地址信息与寄存器 IICAR 中预设的本机地址相符，则发送应答 ACK 以响应主机，并根据主机命令接收或发送数据，完成后检测到总线上的停止信号后结束通讯。

12.4 IIC 相关寄存器

IIC 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IICCR	IICEOF	IICAAS	IICBUSY	IICMOD	SACK	MSTCMD	IICEN	RACK
R/W	R	R	R	R/W	R/W	R	R/W	R
初始值	0	0	0	0	0	0	0	1

BIT[7] **IICEOF** – IIC 通讯一帧数据收发结束标志位

- 0: 收发数据中；
- 1: 一帧数据收发结束；

BIT[6] **IICAAS** – IIC 地址匹配标志位

- 0: 地址不匹配，或接收到数据帧；
- 1: 地址匹配；

BIT[5] **IICBUSY** – IIC 总线状态位

- 0: IIC 总线空闲；
- 1: IIC 总线被占用；



- BIT[4] **IICMOD** – IIC 通讯发送/接收模式控制位
 0: 接收模式, 接收主机发送的数据;
 1: 发送模式, 向主机发送数据;
- BIT[3] **SACK** – 发送应答信号控制位 (IICMOD=0 时有效)
 0: 接收模式下收到数据后应答;
 1: 接收模式下收到数据后不应答;
- BIT[2] **MSTCMD** – 主机发送的命令信号位
 0: 主机发送写命令, 从机将接收数据;
 1: 主机发送读命令, 从机需发送数据;
- BIT[1] **IICEN** – IIC 通讯接口使能位
 0: 关闭 IIC 通讯接口;
 1: 使能 IIC 通讯接口;
- BIT[0] **RACK** – 接收到应答信号状态位 (IICMOD=1 时有效)
 0: 向主机发送数据后收到应答;
 1: 向主机发送数据后未收到应答;

IIC 地址寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IICAR	IICA6	IICA5	IICA4	IICA3	IICA2	IICA1	IICA0	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
初始值	0	0	0	0	0	0	0	-

BIT[7:1] **IICA[6:0]** – 预设的 7 位从机地址信息, 用作本机在总线中的地址编码

IIC 数据寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IICDR	IICD7	IICD6	IICD5	IICD4	IICD3	IICD2	IICD1	IICD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **IICD[7:0]** – IIC 数据寄存器, 缓存 IIC 通讯中待发送或接收到的 8 位数据 (或控制字节)

12.5 IIC 应用流程

IIC 通讯接口, 通过寄存器位 IICEN 控制 IIC 接口的关闭/使能, 通过 IICMOD 设置 IIC 接口当前的工作模式是发送模式还是接收模式。

接收模式下, 通过 SACK 位设置应答信号, 当接收到主机发送的 8 位数据后, 第 9 个时钟 IIC 模块



将 SACK 位内容发送至 IIC 总线，若需继续接收下一个数据，则在接收数据前 SACK 位需清 0。

标志位 IICEOF 在开始传送数据时被清 0，在一帧数据收发结束后被置 1。

标志位 IICAAS 当从器件地址匹配时被置 1，同时 IIC 中断标志（IICIF）被置 1；如果地址不匹配，IICAAS 被清 0。

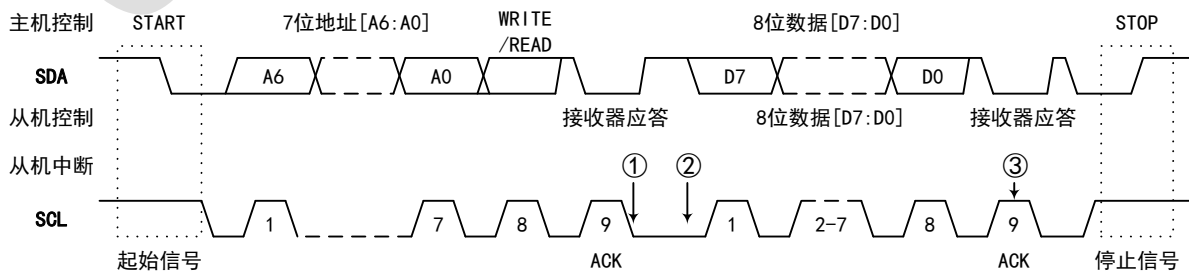
标志位 IICBUSY=1 时表示 IIC 总线忙，即检测到 START 信号；IICBUSY=0 表示 IIC 总线空闲，即检测到“STOP”信号，此时 I2C 总线空闲。

标志位 MSTCMD 表示当地址匹配时，主机同时发送过来的读/写命令。当 IICAAS 被置 1 后，可通过 MSTCMD 位来确定芯片是准备发送数据还是接收数据。当 MSTCMD=1，表示主机将从 IIC 总线读取数据，从机需将数据写到 IIC 总线上，即从机为发送模式；当 MSTCMD=0，表示主机将写数据到 IIC 总线，从机需从总线读取数据，即从机为接收模式。

标志位 RACK=0，表示接收到一个应答信号，在发送模式，从机通过检测 RACK 以确定主机是否要接收下一个数据。从机将一直写数据到 IIC 总线直到 RACK=1，同时将释放 SDA 线，从而主机可以发送 STOP 信号来释放总线。

应用 IIC 通讯接口时，需先将预定的从机地址写入寄存器 IICAR 的第 7~1 位中保存，开始数据传送时，主机通过发送从机地址来指定通讯对象。一旦 IIC 总线上出现起始信号，则所有的从机都会接收到连续的 8 位数，该数据的前 7 位是从机地址，高位在前，低位在后。如果从机地址匹配，系统会将 IICAAS 置 1，同时产生 IIC 中断。进入中断服务程序后，系统要检测 IICAAS 位，以确定 IIC 总线中断是因从机地址匹配而产生的，还是来自 8 位数据传送完毕而产生的。8 位地址帧的最后一位是主机发送的读/写控制位，该位会反映到 MSTCMD。从机通过检测 MSTCMD 位以确定主机是要发送数据还是接收数据，并确定本机是作发送器还是接收器。

- (1) 将本机分配的 IIC 总线地址编码预先写入 IICAR 中；
- (2) IICEN 置 1，使能 IIC 模块；
- (3) IICIE 置 1，允许 IIC 中断；
- (4) 当从机地址匹配且 SACK=0 时，图中①处 IICAAS 将置 1 并产生中断，且会将 SCL 线拉低；
- (5) 通过检测 MSTCMD 来确定本机是准备发送数据还是接收数据：
 - ◇ 若将发送数据，则将 IICMOD 置 1，再将数据写入 IICDR 中；
 - ◇ 若将接收数据，则将 IICMOD 清 0，并设置 SACK 位以决定下一帧是否应答，如图中③处所示；
- (6) IICIF 清 0，并释放 SCL 线，如图中②处所示；
- (7) 当下一帧第 9 个时钟结束时，IICEOF=1 且产生中断。如果芯片处于接收模式且 SACK 为 0，或芯片处于发送模式且 RACK=0，将拉低时钟线 SCL；否则芯片会释放时钟线 SCL 和数据线 SDA 等待主机发 STOP 命令。





13 EEPROM 存储器

13.1 EEPROM 概述

芯片内置 128×16 位的 EEPROM 数据存储单元，支持用户程序在带电工作中实时地读出或写入数据。读写 EEPROM 中的数据时需通过控制寄存器 EECR、数据寄存器 EEDRH/EEDRL、地址寄存器 EEAR 和保护寄存器 EEMASK 进行。

EECR 中 EETRIG 位为操作启动标志位，置 1 启动读写操作，完成后自动清 0；EERW 位为读写命令位，为 0 表示读数据、为 1 表示写数据：读数据操作将从 EEAR 对应的 EEPROM 地址中读出 16 位数据，保存在 EEDRH/EEDRL 中，写数据操作则将 EEDRH/EEDRL 中的 16 位数据写入 EEAR 对应的 EEPROM 地址中。

为防止误触发 EEPROM 读写操作，寄存器 EEMASK 需先写入 5AH 再立即写入 A5H，并在随后 2 个指令周期内将 EETRIG 位置 1，才能执行读写操作。否则 2 个指令周期后 EEMASK 将自动清零，此时 EETRIG 位置 1 不会执行读写操作，也不会自动清 0。

芯片仅支持单地址 16 位数据的读写操作，不支持连续地址读写功能，每次都必须通过 EEAR 设置将要访问的 EEPROM 数据的 7 位地址后，才能进行读写操作。

当启动 EEPROM 读写操作后，CPU 将暂停在当前指令，只有等 EEPROM 读写操作完成后，才能继续执行下一条指令。在读写 EEPROM 时需屏蔽中断并清 WDT 计数器，否则会因系统响应中断而导致读写错误。

注：

- 1、EEPROM 的操作时钟为 CPU 时钟，若 EE 的操作时钟频率过低将导致读写故障，所以 F_{cpu} 低于 EE 操作时钟最低频率时禁止读写 EEPROM (EE 操作时钟最低频率参见特性参数表)；
- 2、与 EEPROM 相关的寄存器 EECR、EEDR、EEAR 和 EEMASK，仅能通过 MOVAR 和 MOVRA 指令进行读写，其他指令的执行结果不确定；

13.2 EEPROM 相关寄存器

EEPROM 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
EECR	EETRIG	EERW	-	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-	-
初始值	0	0	-	-	-	-	-	-



- BIT[7] **EETRIG** – EEPROM 读写启动控制位
 0: 未进行 EEPROM 读写操作, 或 EEPROM 读写操作完成后硬件自动清 0;
 1: 启动 EEPROM 读写操作;
- BIT[6] **EERW** – EEPROM 读写命令位
 0: 从 EEPROM 中读出数据;
 1: 向 EEPROM 中写入数据;

EEPROM 保护寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
EEMASK	EEMASK7	EEMASK6	EEMASK5	EEMASK4	EEMASK3	EEMASK2	EEMASK1	EEMASK0
R/W	W	W	W	W	W	W	W	W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **EEMASK[7:0]** – EEPROM 操作保护位, 需先写 5AH 再立即写 A5H, EETRIG 才能置 1

EEPROM 地址寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
EEAR	-	EEA6	EEA5	EEA4	EEA3	EEA2	EEA1	EEA0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	0	0	0	0	0	0	0

BIT[6:0] **EEA[6:0]** – EEPROM 读写操作的 7 位地址

EEPROM 数据寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
EEDRH	EED15	EED14	EED13	EED12	EED11	EED10	EED9	EED8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **EED[15:8]** – EEPROM 读写操作的 16 位数据高 8 位

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
EEDRL	EED7	EED6	EED5	EED4	EED3	EED2	EED1	EED0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **EED[7:0]** – EEPROM 读写操作的 16 位数据低 8 位



13.3 EEPROM 操作示例

例：向地址为 10H 的 EEPROM 单元写入 16 位数据 55AAH

BSET	HFEN	； 确保系统高频时钟开启
MOVAI	00H	
MOVRA	EECR	； 清 EETRIG 位
MOVAI	10H	
MOVRA	EEAR	； 将地址 10H 写入 EEAR
MOVAI	55H	
MOVRA	EEDRH	； 将数据 55H 写入 EEDRH
MOVAI	AAH	
MOVRA	EEDRL	； 将数据 55H 写入 EEDRL
BCLR	GIE	； 屏蔽中断
CLRWDT		； 清 WDT
MOVAI	5AH	
MOVRA	EEMASK	； 使能 EE 操作，第 1 步
MOVAI	A5H	
MOVRA	EEMASK	； 使能 EE 操作，第 2 步
MOVAI	COH	
MOVRA	EECR	； 启动 EE 写操作，将数据 55AAH 写入 EEPROM 地址 10H 中
NOP		； 为防止时序错误，CPU 必须先执行 1-2 个 NOP 指令
BSET	GIE	； 允许中断

例：从地址为 10H 的 EEPROM 单元读出 16 位数据（数据存放在 EEDRH/EEDRL 中）

BSET	HFEN	； 确保系统高频时钟开启
MOVAI	00H	
MOVRA	EECR	； 清 EETRIG 位
MOVAI	10H	
MOVRA	EEAR	； 将地址 10H 写入 EEAR
BCLR	GIE	； 屏蔽中断
CLRWDT		； 清 WDT
MOVAI	5AH	
MOVRA	EEMASK	； 使能 EE 操作，第 1 步
MOVAI	A5H	
MOVRA	EEMASK	； 使能 EE 操作，第 2 步
MOVAI	80H	
MOVRA	EECR	； 启动 EE 写操作，从 EEPROM 地址 10H 中读出数据
NOP		； 为防止时序错误，CPU 必须先执行 1-2 个 NOP 指令
BSET	GIE	； 允许中断



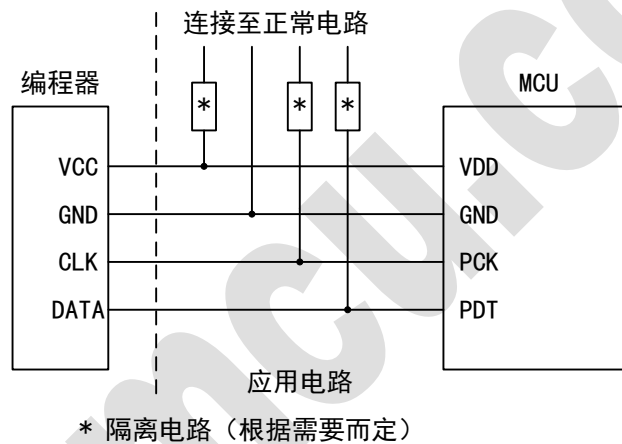
14 FLASH 烧录编程

14.1 FLASH 在板编程

芯片支持编程工具对芯片中程序存储器的在板不带电烧录编程，即在未上电的系统电路板上，借助编程工具，通过芯片的串行编程接口将用户程序代码烧录进芯片的程序存储器中。在板编程功能，可让用户先采用未编程的空芯片制造电路板而仅在产品交付前才将程序代码烧录进芯片，也方便用户直接在电路板上升级 FLASH 存储器中的程序代码。

芯片也支持对 EEPROM 型数据存储器的在板编程。

芯片的在板编程通过引脚 VDD、GND、PCK、PDT 实现，这些编程引脚的外围电路需进行针对性设计，以保证外围电路不会影响在板编程时端口上的电压/电流/时序等特性。下图是典型的在板编程连接示意图：



芯片也支持在板带电烧录编程，即可在系统电路板不掉电（芯片已正常工作）的状态下对存储器编程。当配置字 DBGPINE 和寄存器控制位 DBGGEN 同时使能后，配置字 DBGPINS 选定的端口则可作为编程端口，芯片可通过该组端口进入编程/仿真模式。

注：

- 1、不支持空芯片的在板带电烧录编程；
- 2、端口使能编程功能后，还需设为数字输入状态并关闭复用的模拟功能，才可正常用作编程端口；
- 3、在板带电烧录，编程器 VCC 不接入电路板，芯片由系统电路板通过 VDD 引脚供电；
- 4、在板带电烧录，编程器 GND/CLK/DATA 接入电路板前，需注意编程器与系统电路板的共 GND 是否会发生浮地与市电冲突问题；
- 5、在板带电烧录，编程器在烧录完成后，会将芯片重新复位；

DEBUG 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
DBGCR	-	-	-	-	-	-	-	DBGEN



R/W	-	-	-	-	-	-	-	R/W
初始值	-	-	-	-	-	-	-	0

BIT[0] **DBGEN** – 端口编程功能使能位

0: 关闭端口的编程功能;

1: 使能配置字 **DBGPINS** 所选端口的编程功能;

DEBUG 保护寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
DBGPR	DBGP7	DBGP6	DBGP5	DBGP4	DBGP3	DBGP2	DBGP1	DBGP0
R/W	W	W	W	W	W	W	W	W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **DBGP[7:0]** – **DBGCR** 写操作保护控制位

为防止误触发对 **DBGCR** 的写操作，需先对寄存器 **DBGPR** 写 3CH 再立即写 C3H，**DBGEN** 才能置 1，中间不能插入其他操作（包括 NOP 操作），否则对 **DBGCR** 的写操作将无效。在写 **DBGPR** 前需先屏蔽中断，否则可能会因系统响应中断而导致 **DBGCR** 写操作无效。

例如，在配置字 **DBGPINS** 和 **DBGPINE** 将 PCK1 (P14) /PDT1 (P12) 配置成编程/仿真接口后，芯片带电工作中若 **DBGEN**=0，则 P12/P14 仍用作数字或模拟功能的输入/输出脚，编程器无法通过这两个引脚进行带电烧录编程。

用户程序可按如下例程将 **DBGEN** 置 1，以使编程器可通过这两个引脚进行带电烧录编程：

```

BCLR      P120E
BCLR      P140E      ; P12/P14 作为输入口
BCLR      P12DC
BCLR      P14DC      ; 使能 P12/P14 的数字 I/O 功能
MOVAI     E2H
MOVRA     ADCR0      ; 关闭 ADC，输入通道选为 GND
MOVAI     00H
MOVRA     OPACR      ; 关闭 OPA，断开 OPR01
BCLR      GIE        ; 屏蔽中断
CLRWDT    ; 清 WDT
MOVAI     3CH
MOVRA     DBGPR      ; 使能 DBGEN 写操作，第 1 步
MOVAI     C3H
MOVRA     DBGPR      ; 使能 DBGEN 写操作，第 2 步
MOVAI     01H
MOVRA     DBGCR      ; DBGEN 置 1，P12/P14 用作编程接口
NOP
CLRWDT    ; 清 WDT
GOTO     $-2         ; 死循环，等待编程器连接芯片进行带电烧录编程

```




15 中断

芯片的中断源包括外部中断 (INT0~INT1)、定时器中断 (T0~T3)、T2 捕捉中断、ADC 中断、CP0 触发中断、比较器中断和 IIC 中断等。可通过中断总使能位 GIE 屏蔽所有中断。

CPU 响应中断的过程如下：

- ◇ CPU 响应中断源触发的中断请求时，自动将当前指令之后将要执行的下一条指令的地址压栈保存，自动清 0 中断总使能位 GIE 以暂停响应后续中断。与复位不同，硬件中断不停止当前指令的执行，而是暂时挂起中断继续执行当前指令，完成后再处理中断。
- ◇ CPU 响应中断后，程序跳至中断入口地址 (0008H) 开始执行中断服务程序，中断服务程序应先保存累加器 A 和状态寄存器 PFLAG，然后处理被触发的中断。
- ◇ 中断服务程序处理完中断后，应先恢复累加器 A 和状态寄存器 PFLAG，再执行 RETIE 指令以返回主程序。系统将自动恢复 GIE 为 1，然后从堆栈取出此前保存的 PC 值，CPU 从响应中断时正在执行指令的下一条指令的地址处开始继续运行。

15.1 外部中断

芯片具有 2 路外部中断源 INT0/INT1，可选择上升沿、下降沿或电平变化等触发方式。外部中断触发时，中断标志 INTnIF (n=0-1) 将被置 1，若 GIE 为 1 且相应的外部中断使能位 INTnIE (n=0-1) 为 1，则产生外部中断。

15.2 定时器中断

定时器 Tn (n=0-3) 在计数溢出时将触发定时器中断，中断标志 TnIF (n=0-3) 将被置 1，若 GIE 为 1 且相应的定时器中断使能位 TnIE (n=0-3) 为 1，则产生定时器中断。

T2 的脉宽测量模式，若 T2PINT=0，则当捕捉到被测电平的停止沿时也将触发 T2 的定时器中断。

15.3 T2 捕捉中断

定时器 T2 的脉宽测量模式，若 T2PINT=1，则当捕捉到被测电平的停止沿时将触发 T2 捕捉中断，中断标志 T2PIF 将被置 1，若 GIE 为 1 且 T2 捕捉中断使能位 T2PIE 为 1，则产生 T2 捕捉中断。



15.4 ADC 中断

AD 转换完成时将触发 ADC 中断，中断标志 ADIF 将被置 1，若 GIE 为 1 且 ADC 中断使能位 ADIE 为 1，则产生 ADC 中断。

15.5 CP0 触发中断

CP0 输出信号 CP0_OUT（或 PTRIG 输入的外部信号）经去抖后的信号 CP0TLV 产生下降沿时，将触发 CP0 触发中断，中断标志 CP0TIF 将被置 1，若 GIE 为 1 且 CP0 触发中断使能位 CP0TIE 为 1，则产生 CP0 触发中断。

15.6 比较器中断

比较器 CPn(n=1-4)的输出信号 CPn_OUT 产生下降沿时将触发比较器中断，中断标志 CPnIF(n=1-4) 将被置 1，若 GIE 为 1 且比较器中断使能位 CPnIE (n=1-4) 为 1，则产生比较器中断。

15.7 LVD 中断

当 VDD 电压降至电压检测阈值以下时，将触发 LVD 中断，中断标志 LVDIF 将被置 1，若 GIE 为 1 且 LVD 中断使能位 LVDIE 为 1，则产生 LVD 中断。

15.8 IIC 中断

IIC 从机模式通讯中，当器件地址匹配、8 位数据接收或发送完成等事件发生时，将触发 IIC 中断，中断标志 IICIF 将被置 1，若 GIE 为 1 且 IIC 中断使能位 IICIE 为 1，则产生 IIC 中断。



15.9 中断相关寄存器

中断使能寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTE0	CP0TIE	ADIE	IICIE	LVDIE	INT1IE	INT0IE	T1IE	T0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] **CP0TIE** – CP0 触发中断使能位

- 0: 屏蔽 CP0 触发中断;
- 1: 使能 CP0 触发中断;

BIT[6] **ADIE** – ADC 中断使能位

- 0: 屏蔽 ADC 中断;
- 1: 使能 ADC 中断;

BIT[5] **IICIE** – IIC 中断使能位

- 0: 屏蔽 IIC 中断;
- 1: 使能 IIC 中断;

BIT[4] **LVDIE** – LVD 中断使能位

- 0: 屏蔽 LVD 中断;
- 1: 使能 LVD 中断;

BIT[3] **INT1IE** – INT1 中断使能位

- 0: 屏蔽 INT1 中断;
- 1: 使能 INT1 中断;

BIT[2] **INT0IE** – INT0 中断使能位

- 0: 屏蔽 INT0 中断;
- 1: 使能 INT0 中断;

BIT[1] **T1IE** – 定时器 T1 中断使能位

- 0: 屏蔽定时器 T1 中断;
- 1: 使能定时器 T1 中断;

BIT[0] **T0IE** – 定时器 T0 中断使能位

- 0: 屏蔽定时器 T0 中断;
- 1: 使能定时器 T0 中断;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTE1	CP3IE	CP2IE	CP1IE	CP4IE	–	T2PIE	T3IE	T2IE
R/W	R/W	R/W	R/W	R/W	–	R/W	R/W	R/W



初始值	0	0	0	0	-	0	0	0
-----	---	---	---	---	---	---	---	---

BIT[7] **CP3IE** – 比较器 CP3 中断使能位

- 0: 屏蔽 CP3 中断;
- 1: 使能 CP3 中断;

BIT[6] **CP2IE** – 比较器 CP2 中断使能位

- 0: 屏蔽 CP2 中断;
- 1: 使能 CP2 中断;

BIT[5] **CP1IE** – 比较器 CP1 中断使能位

- 0: 屏蔽 CP1 中断;
- 1: 使能 CP1 中断;

BIT[4] **CP4IE** – 比较器 CP4 中断使能位

- 0: 屏蔽 CP4 中断;
- 1: 使能 CP4 中断;

BIT[2] **T2PIE** – T2 捕捉中断使能位

- 0: 屏蔽 T2 捕捉中断;
- 1: 使能 T2 捕捉中断;

BIT[1] **T3IE** – 定时器 T3 中断使能位

- 0: 屏蔽定时器 T3 中断;
- 1: 使能定时器 T3 中断;

BIT[0] **T2IE** – 定时器 T2 中断使能位

- 0: 屏蔽定时器 T2 中断;
- 1: 使能定时器 T2 中断;

中断标志寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTFO	CP0TIF	ADIF	IICIF	LVDIF	INT1IF	INT0IF	T1IF	T0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] **CP0TIF** – CP0 触发中断标志位

- 0: 未触发 CP0 触发中断;
- 1: 已触发 CP0 触发中断, 需软件清 0;

BIT[6] **ADIF** – ADC 中断标志位

- 0: 未触发 ADC 中断;
- 1: 已触发 ADC 中断, 需软件清 0;



- BIT[5] **IICIF** – IIC 中断标志位
0: 未触发 IIC 中断;
1: 已触发 IIC 中断, 需软件清 0;
- BIT[4] **LVDIF** – LVD 中断标志位
0: 未触发 LVD 中断;
1: 已触发 LVD 中断, 需软件清 0;
- BIT[3] **INT1IF** – INT1 中断标志位
0: 未触发 INT1 中断;
1: 已触发 INT1 中断, 需软件清 0;
- BIT[2] **INT0IF** – INT0 中断标志位
0: 未触发 INT0 中断;
1: 已触发 INT0 中断, 需软件清 0;
- BIT[1] **T1IF** – 定时器 T1 中断标志位
0: 未触发定时器 T1 中断;
1: 已触发定时器 T1 中断, 需软件清 0;
- BIT[0] **T0IF** – 定时器 T0 中断标志位
0: 未触发定时器 T0 中断;
1: 已触发定时器 T0 中断, 需软件清 0;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTF1	CP3IF	CP2IF	CP1IF	CP4IF	–	T2PIF	T3IF	T2IF
R/W	R/W	R/W	R/W	R/W	–	R/W	R/W	R/W
初始值	0	0	0	0	–	0	0	0

- BIT[7] **CP3IF** – 比较器 CP3 中断标志位
0: 未触发 CP3 中断;
1: 已触发 CP3 中断, 需软件清 0;
- BIT[6] **CP2IF** – 比较器 CP2 中断标志位
0: 未触发 CP2 中断;
1: 已触发 CP2 中断, 需软件清 0;
- BIT[5] **CP1IF** – 比较器 CP1 中断标志位
0: 未触发 CP1 中断;
1: 已触发 CP1 中断, 需软件清 0;
- BIT[4] **CP4IF** – 比较器 CP4 中断标志位
0: 未触发 CP4 中断;
1: 已触发 CP4 中断, 需软件清 0;



- BIT[2] **T2PIF** – T2 捕捉中断标志位
0: 未触发 T2 捕捉中断;
1: 已触发 T2 捕捉中断, 需软件清 0;
- BIT[1] **T3IF** – 定时器 T3 中断标志位
0: 未触发定时器 T3 中断;
1: 已触发定时器 T3 中断, 需软件清 0;
- BIT[0] **T2IF** – 定时器 T2 中断标志位
0: 未触发定时器 T2 中断;
1: 已触发定时器 T2 中断, 需软件清 0;

Sinomcu.com



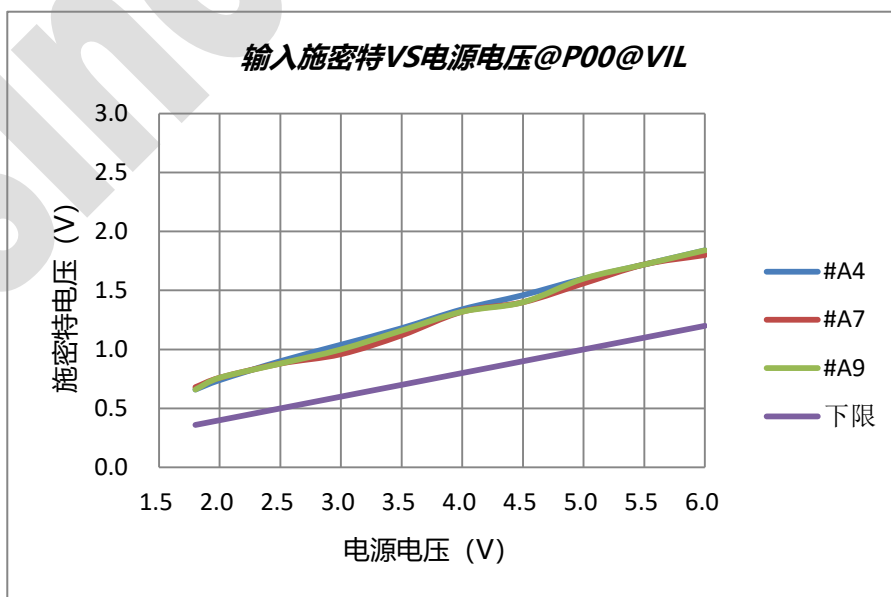
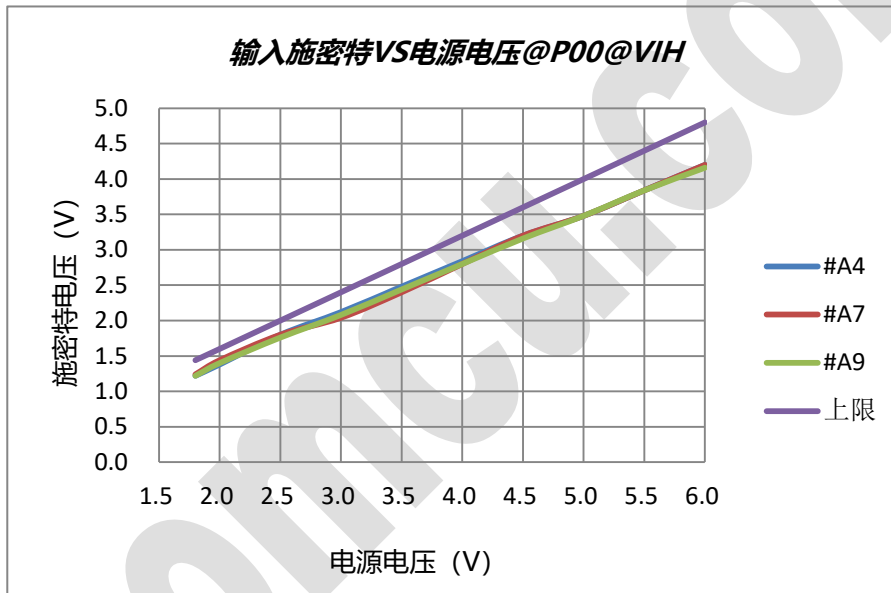
16 特性曲线

注:

- 1、特性曲线图中数据均源自抽样实测, 仅作为应用参考, 部分数据因生产工艺偏差, 可能与实际芯片不符; 为保证芯片能正常工作, 请确保其工作条件符合电气特性参数说明;
- 2、图文中若无特别说明, 则电压特性曲线的温度条件为 $T=25^{\circ}\text{C}$, 温度特性曲线的电压条件为 $V_{DD}=5\text{V}$;

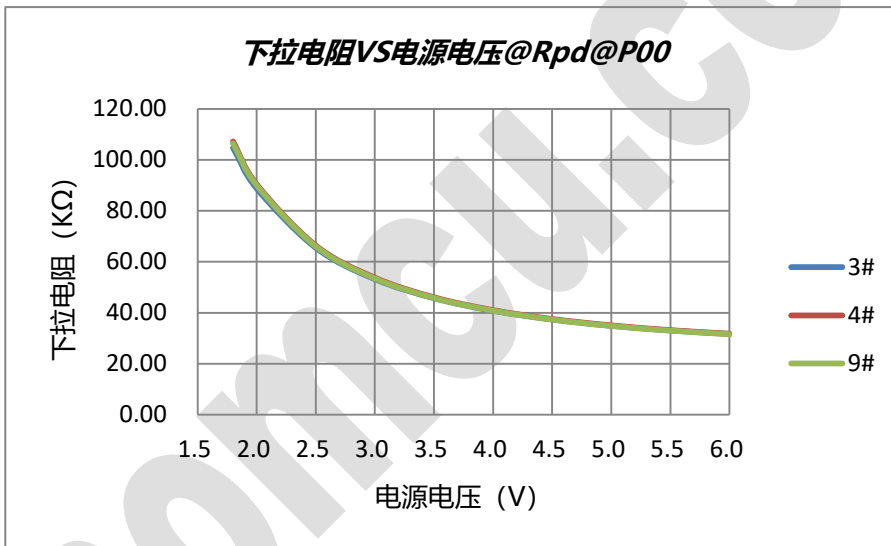
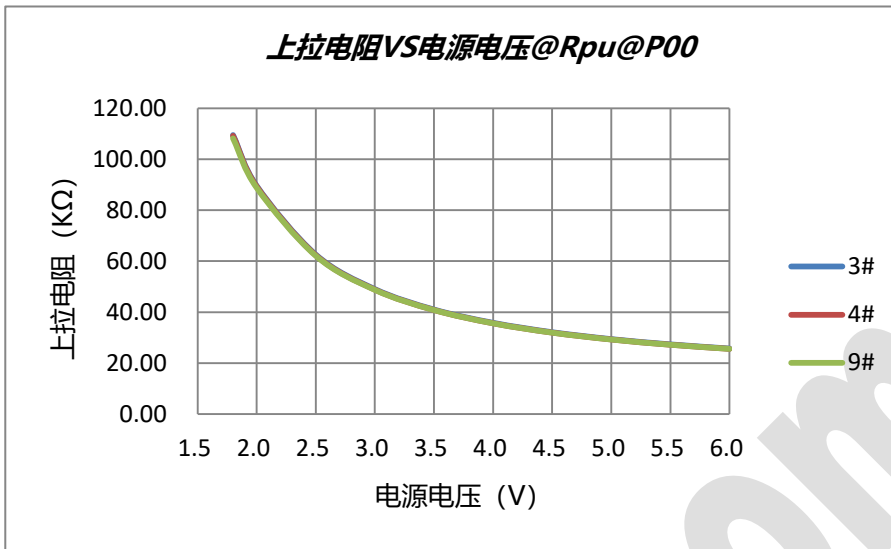
16.1 I/O 特性

输入 SMT 阈值电压 VS 电源电压

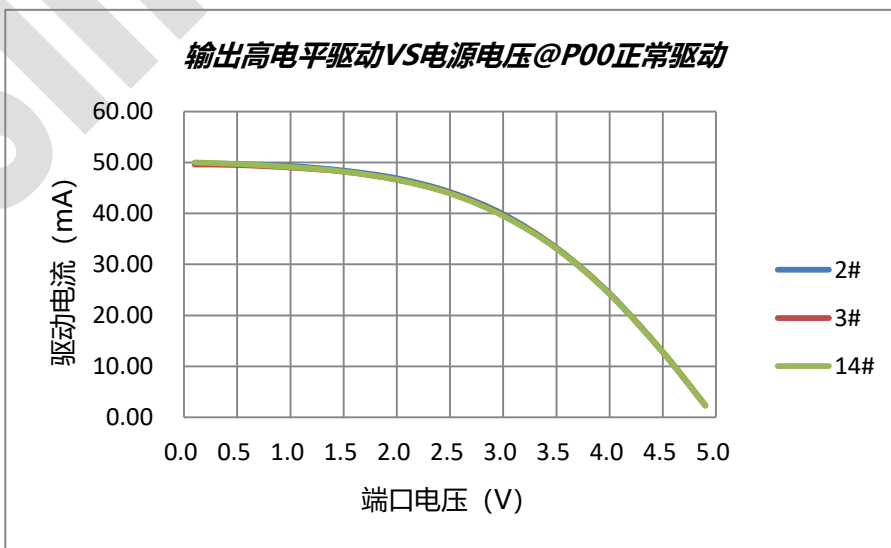


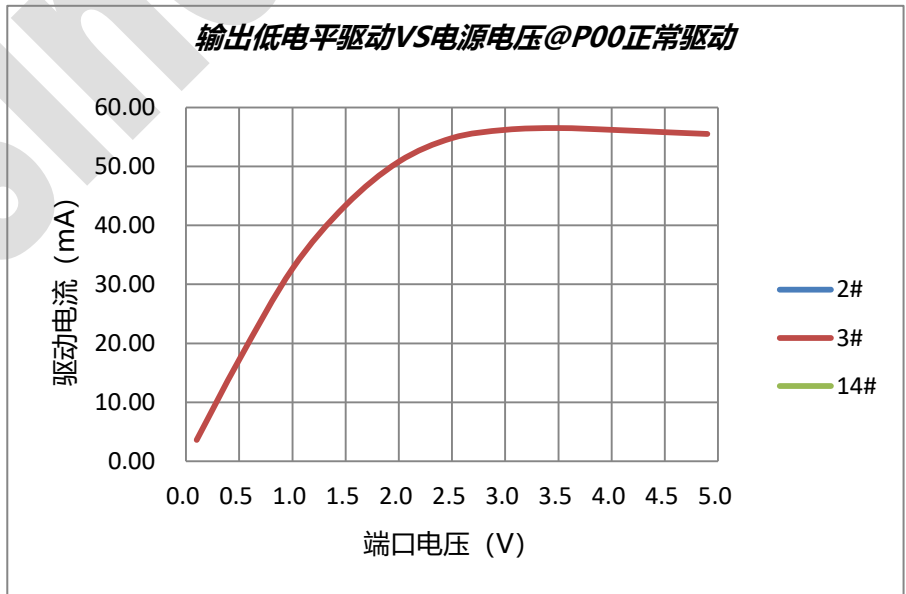
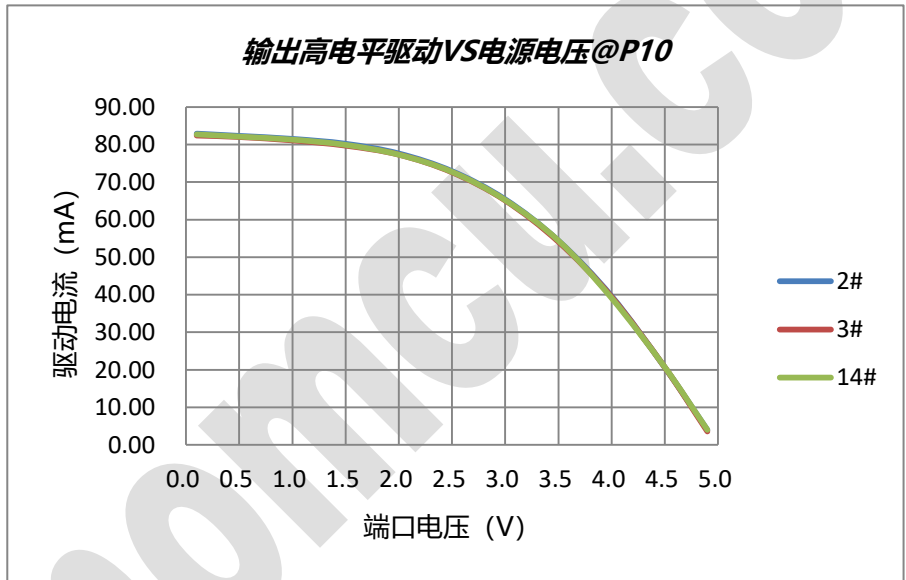
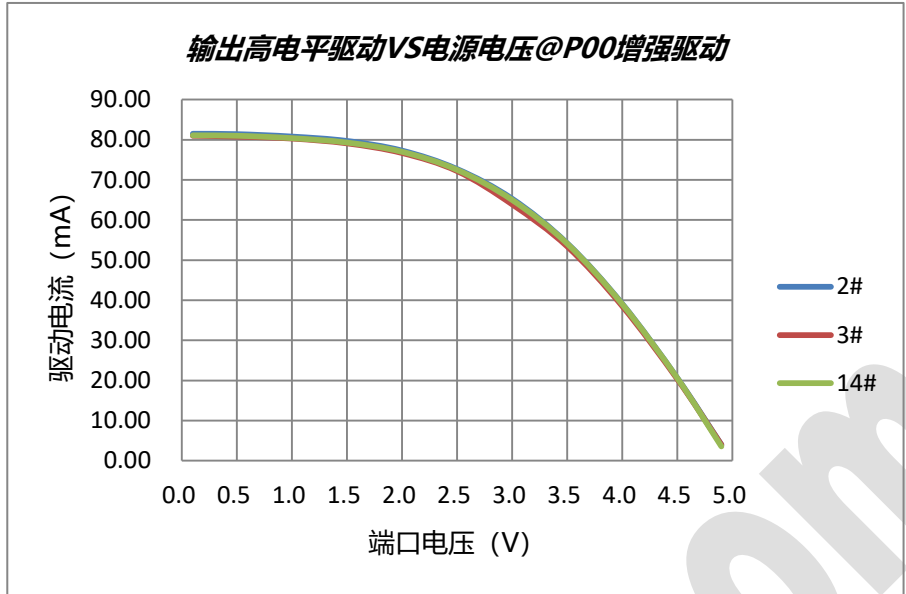


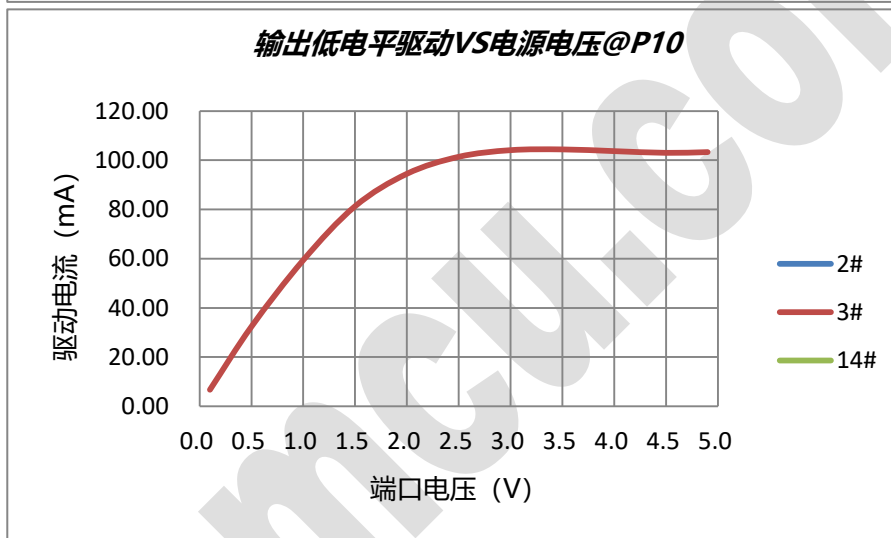
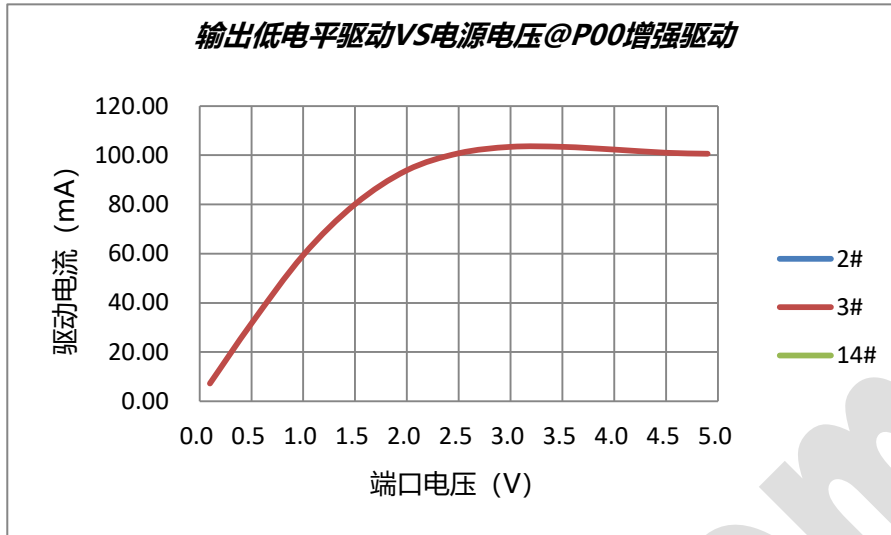
上/下拉电阻值 VS 电源电压



I/O 输出电流 VS 端口电压 (VDD=5V)

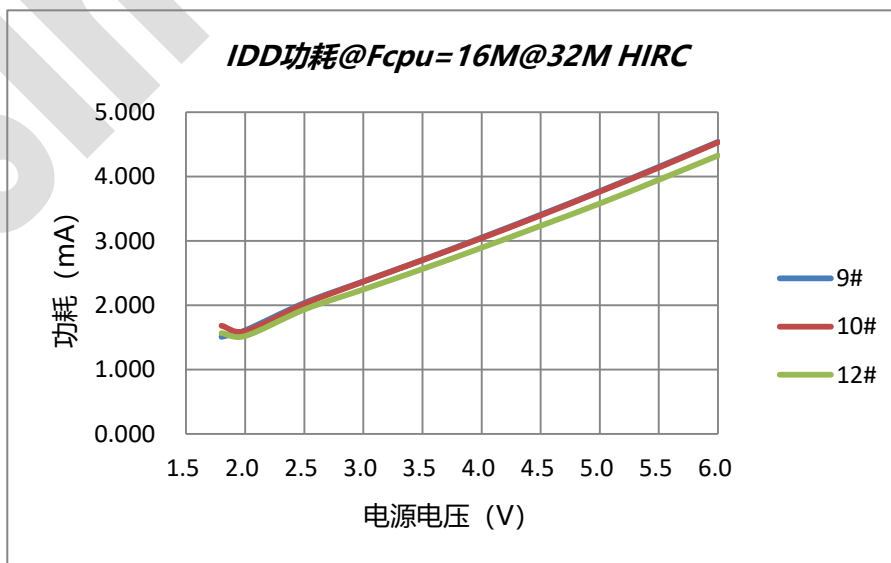


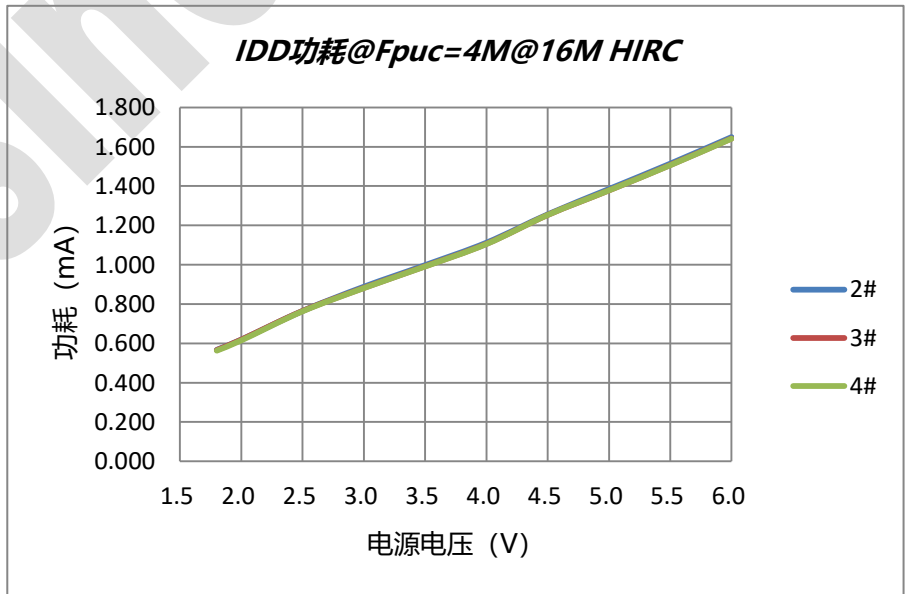
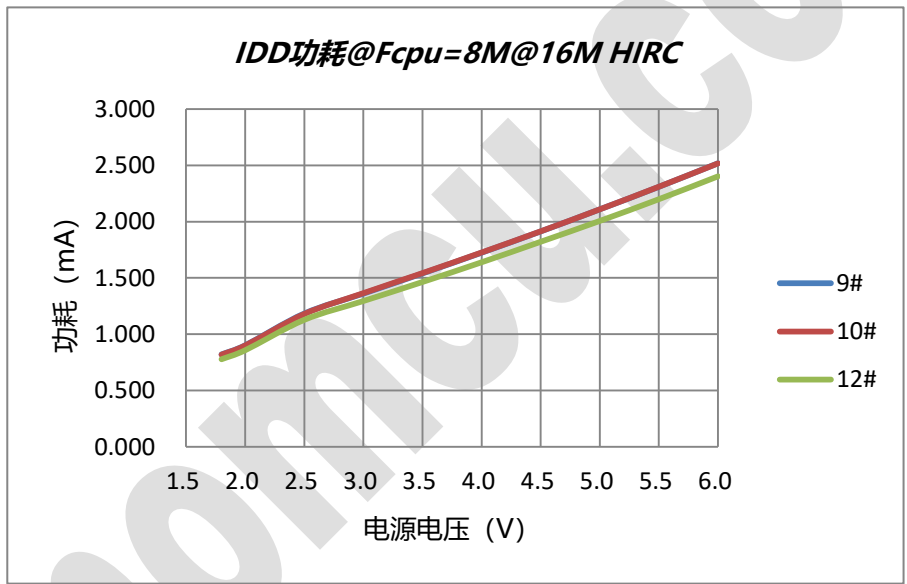
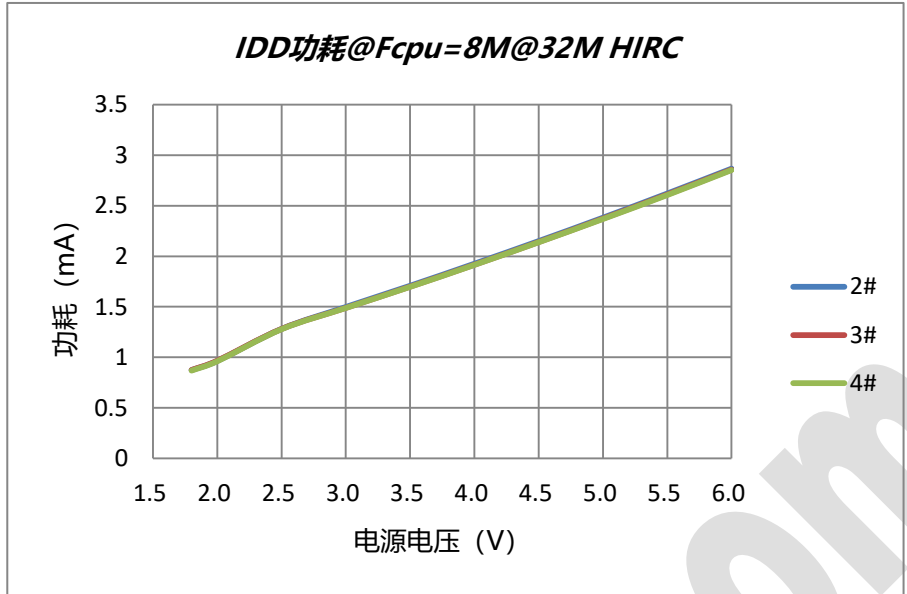


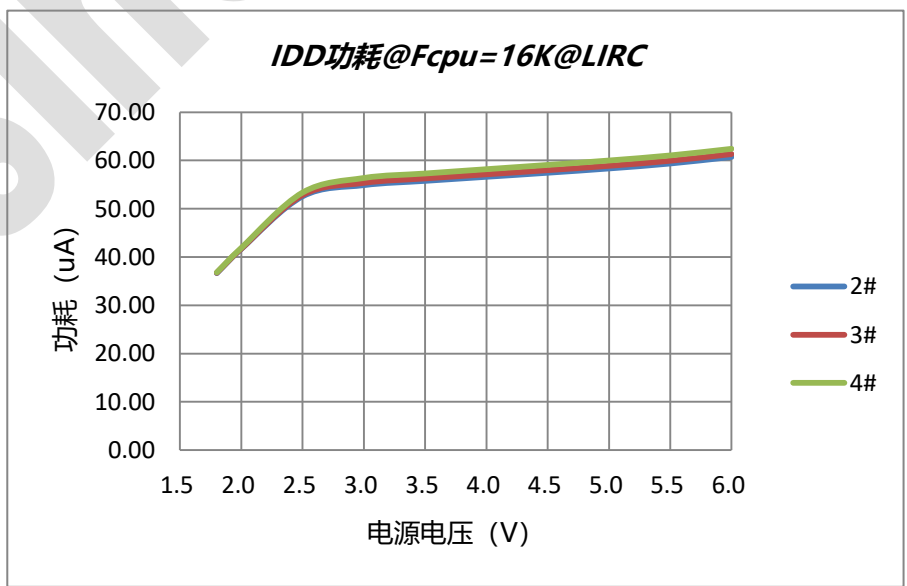
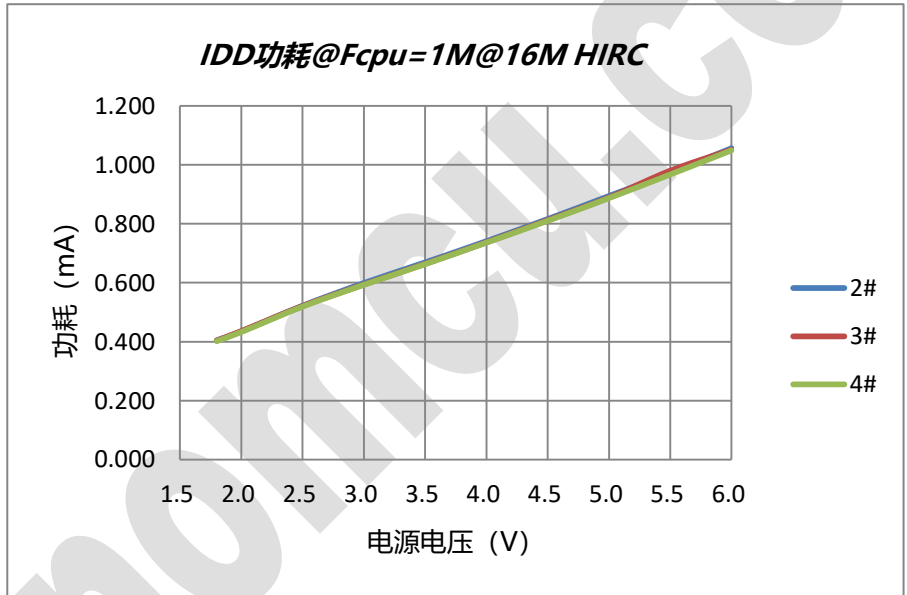
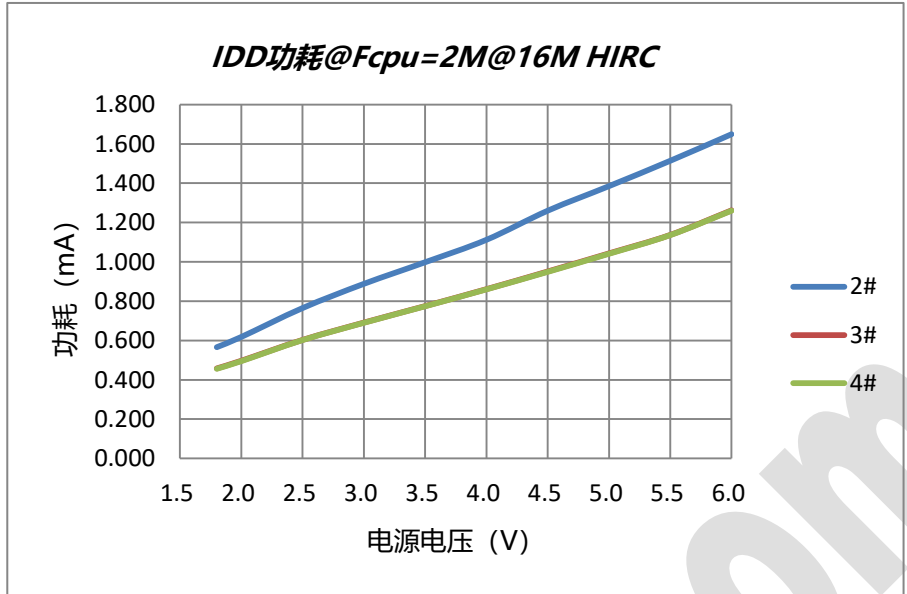


16.2 功耗特性

运行模式 功耗 VS 电源电压

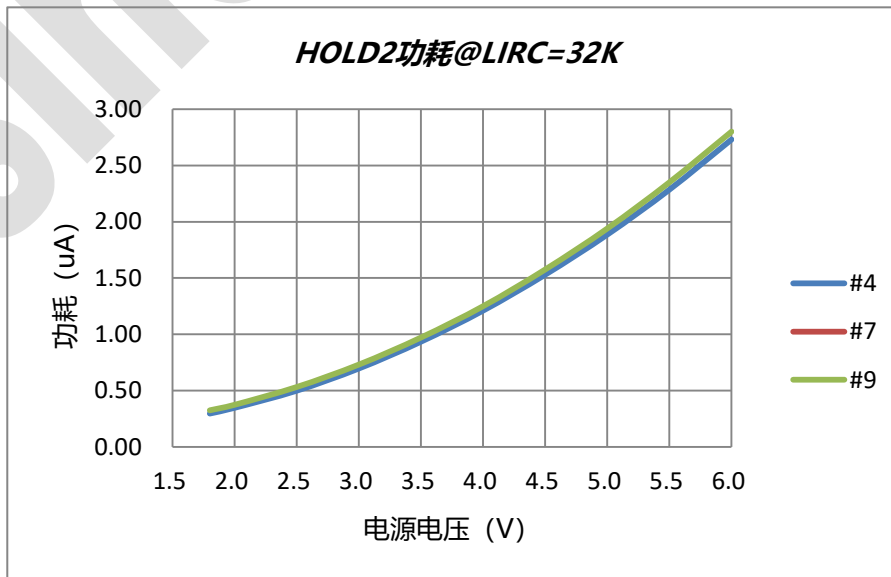
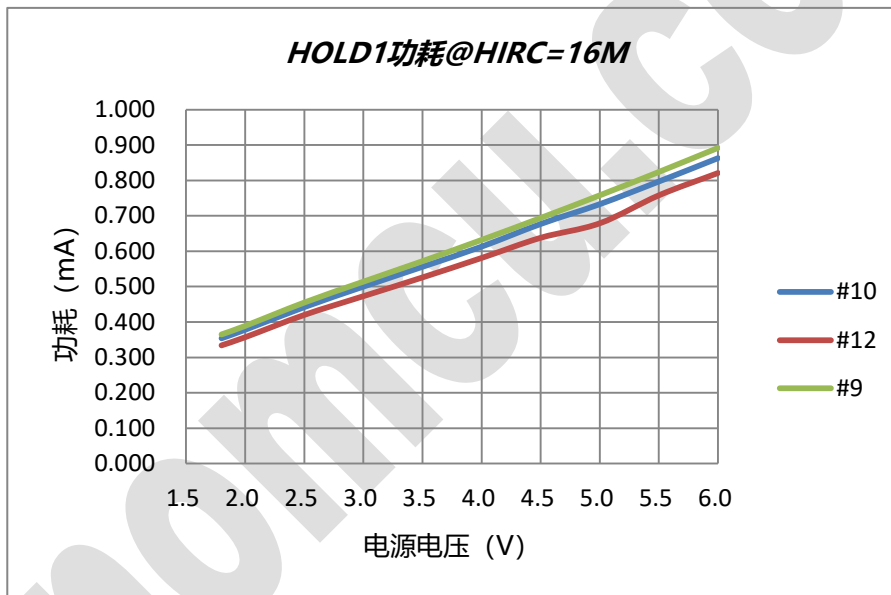
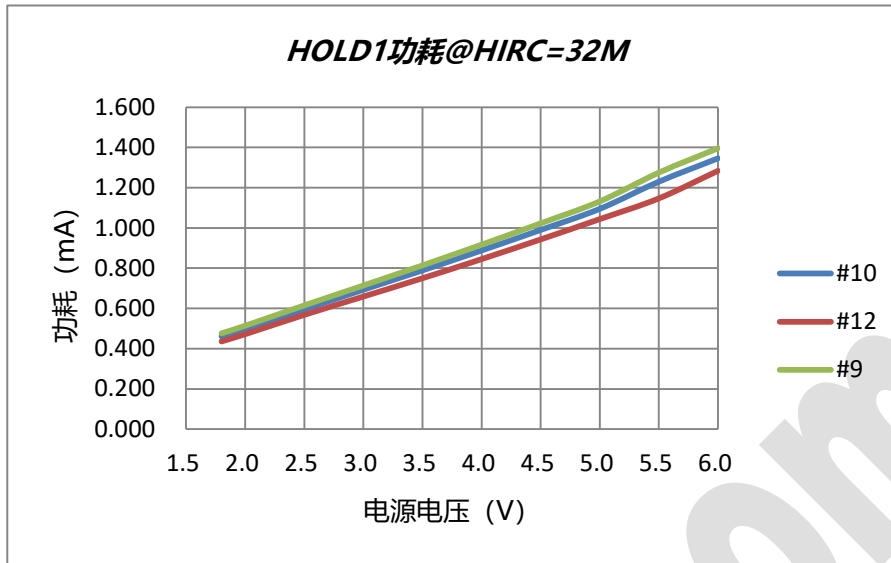






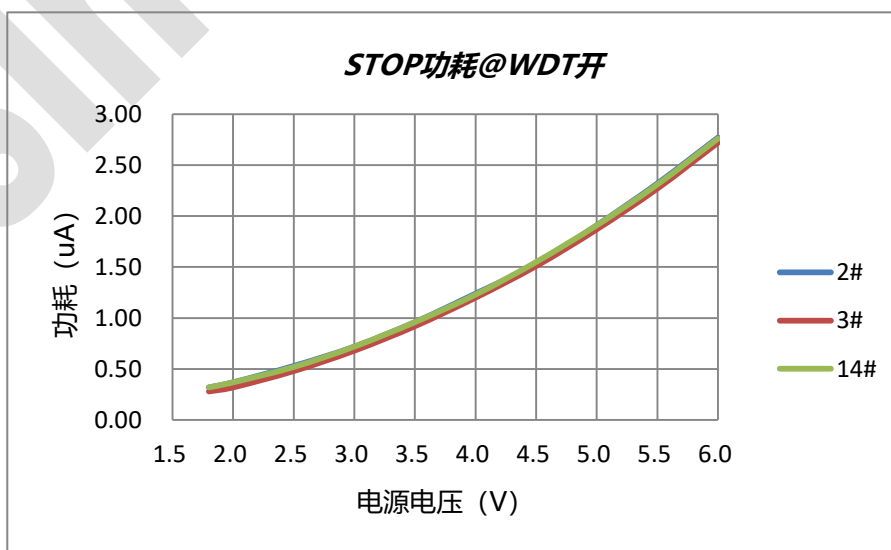
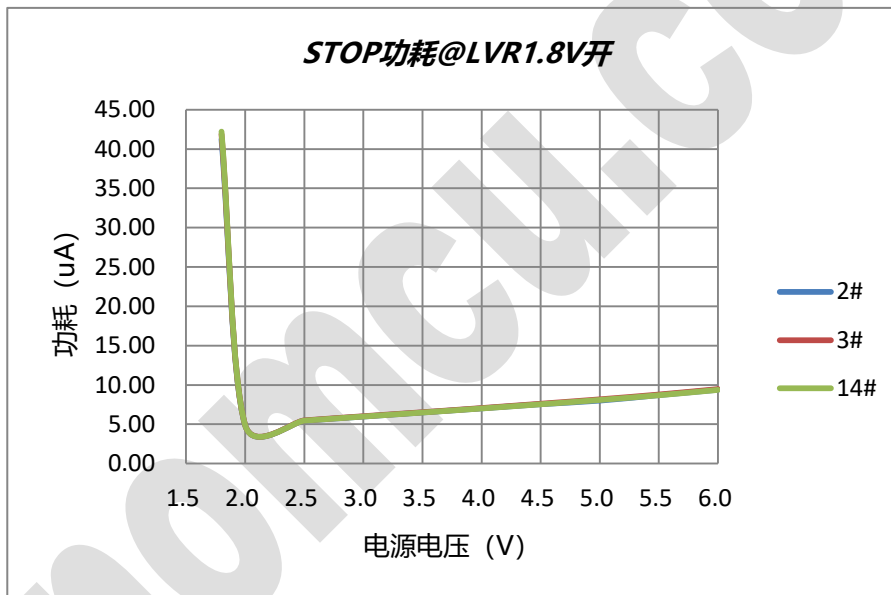
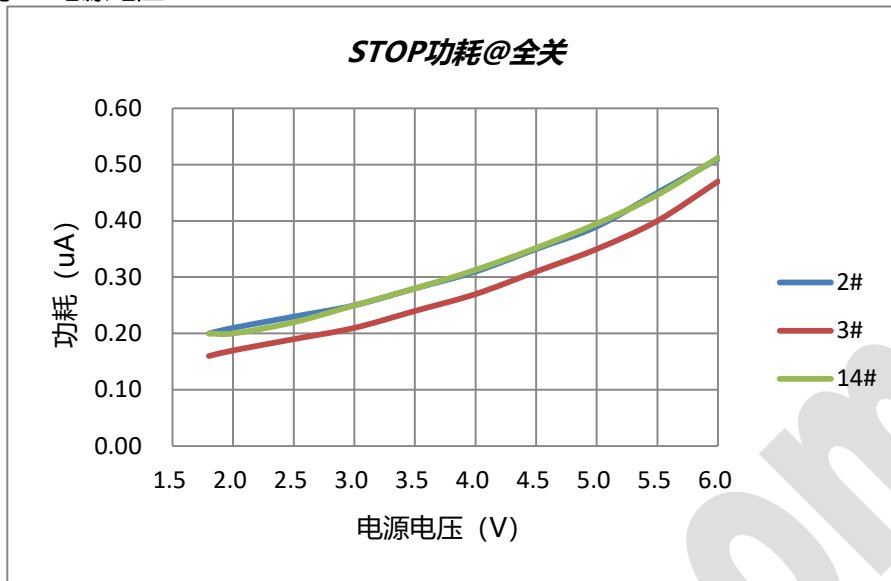


HOLD 模式 功耗 VS 电源电压





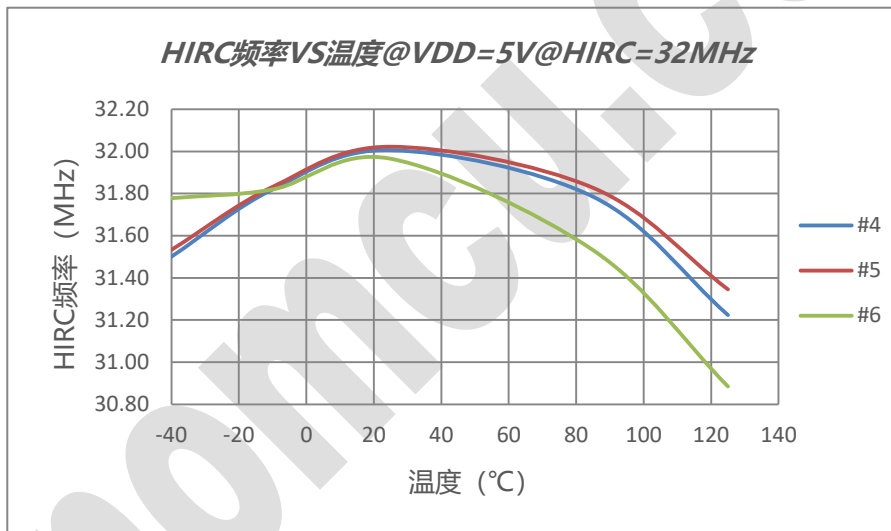
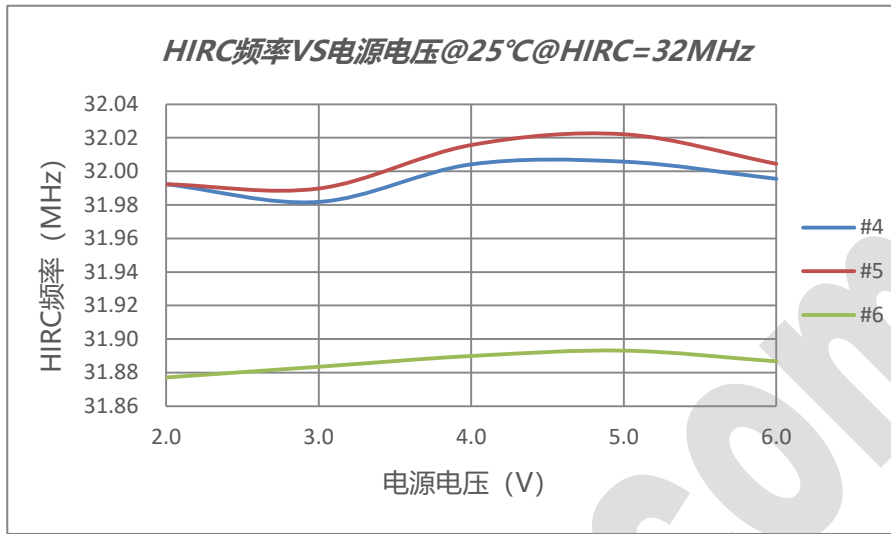
休眠模式 功耗 VS 电源电压



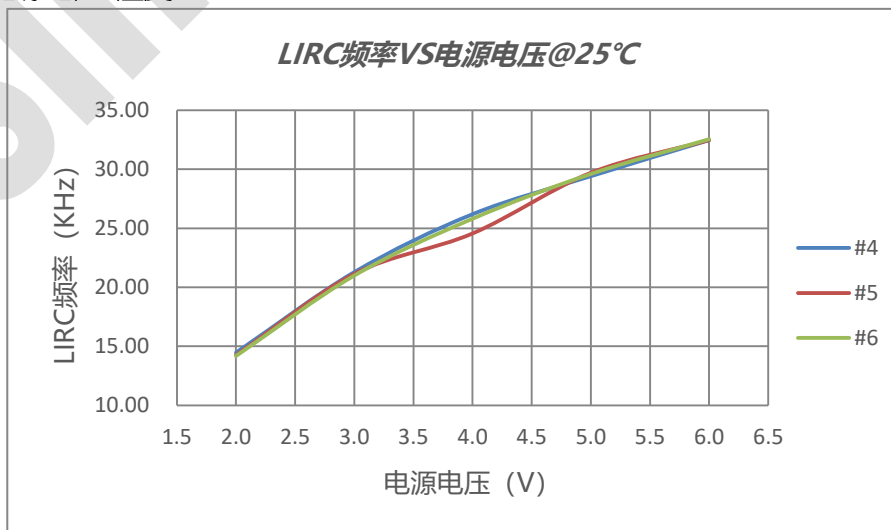


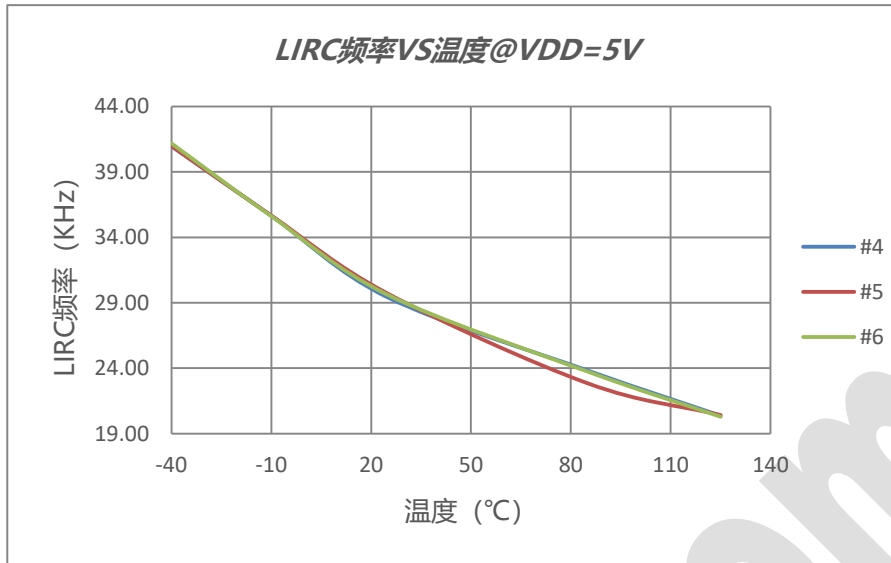
16.3 模拟电路特性

HIRC 频率 VS 电源电压/温度

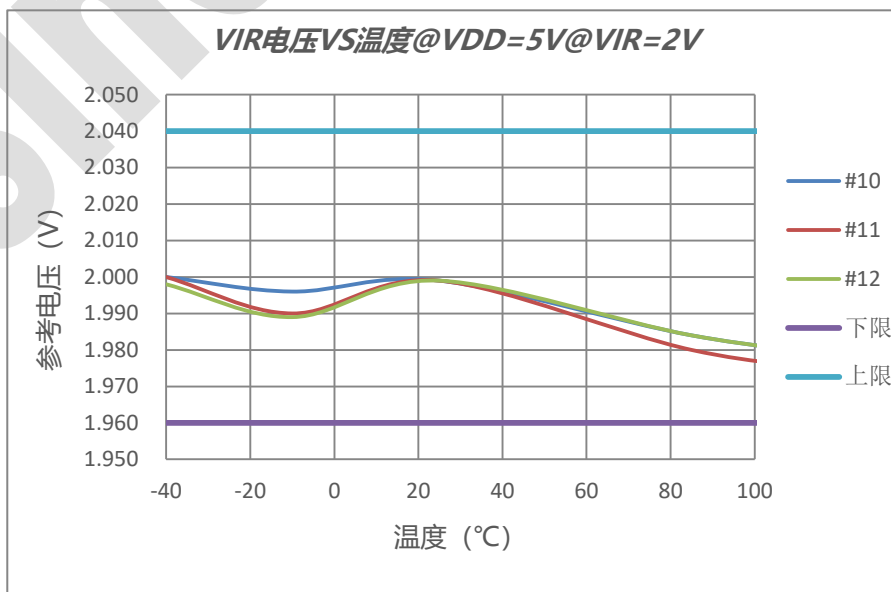
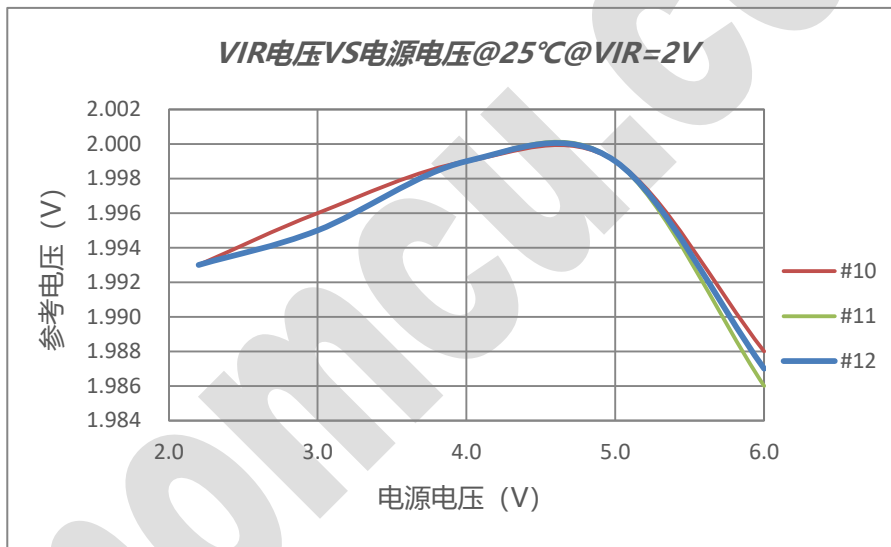


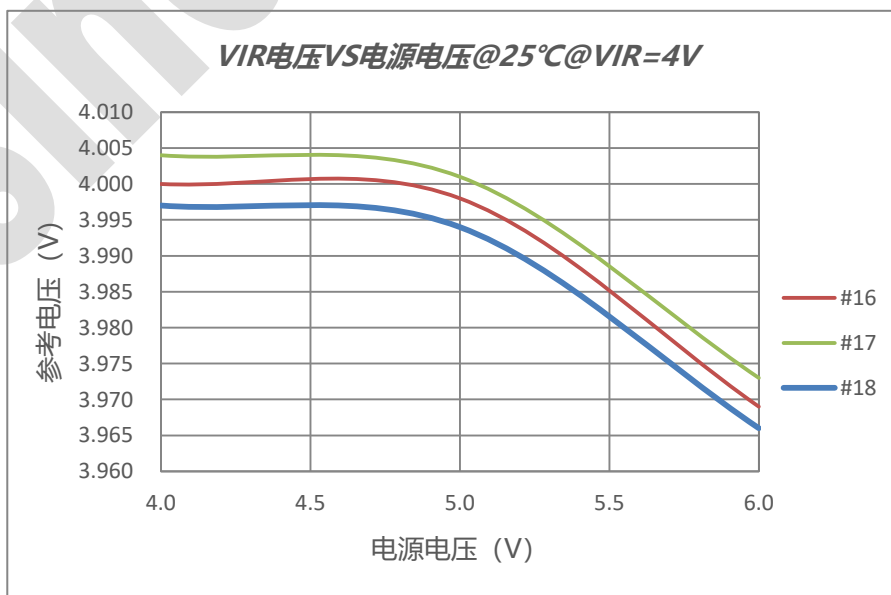
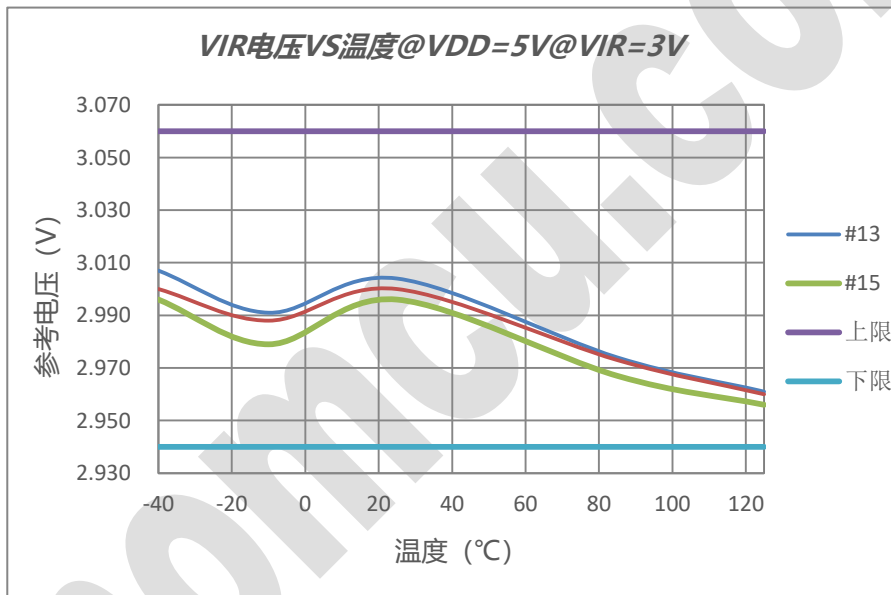
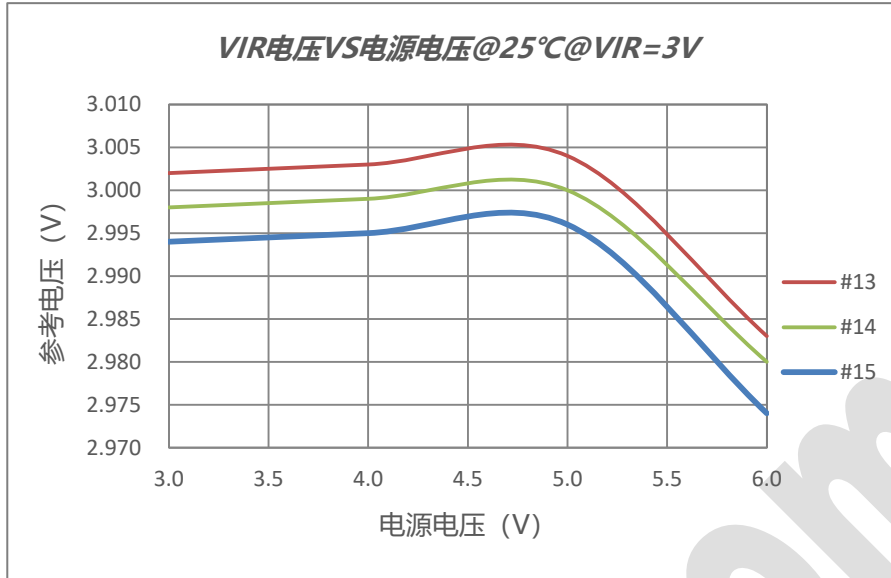
LIRC 频率 VS 电源电压/温度

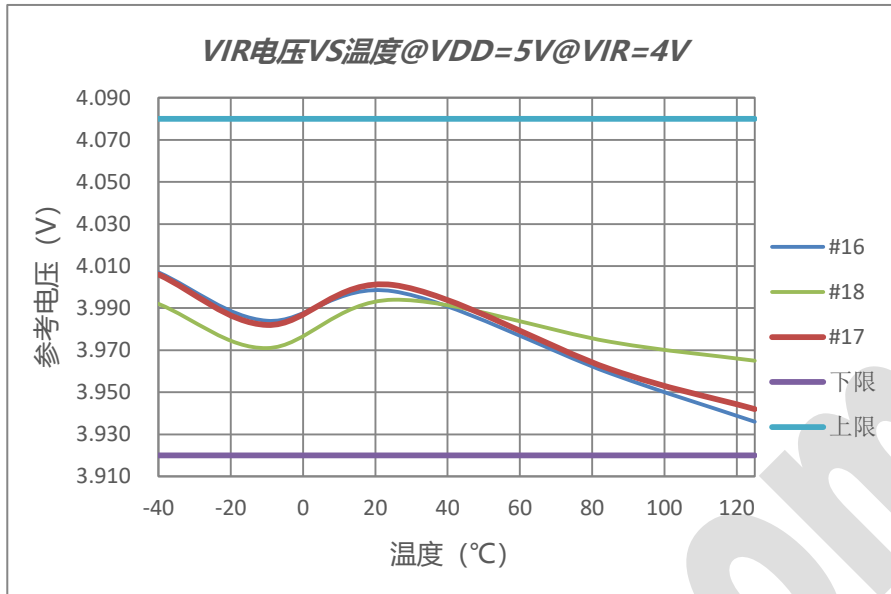




VIR 电压 VS 电源电压/温度





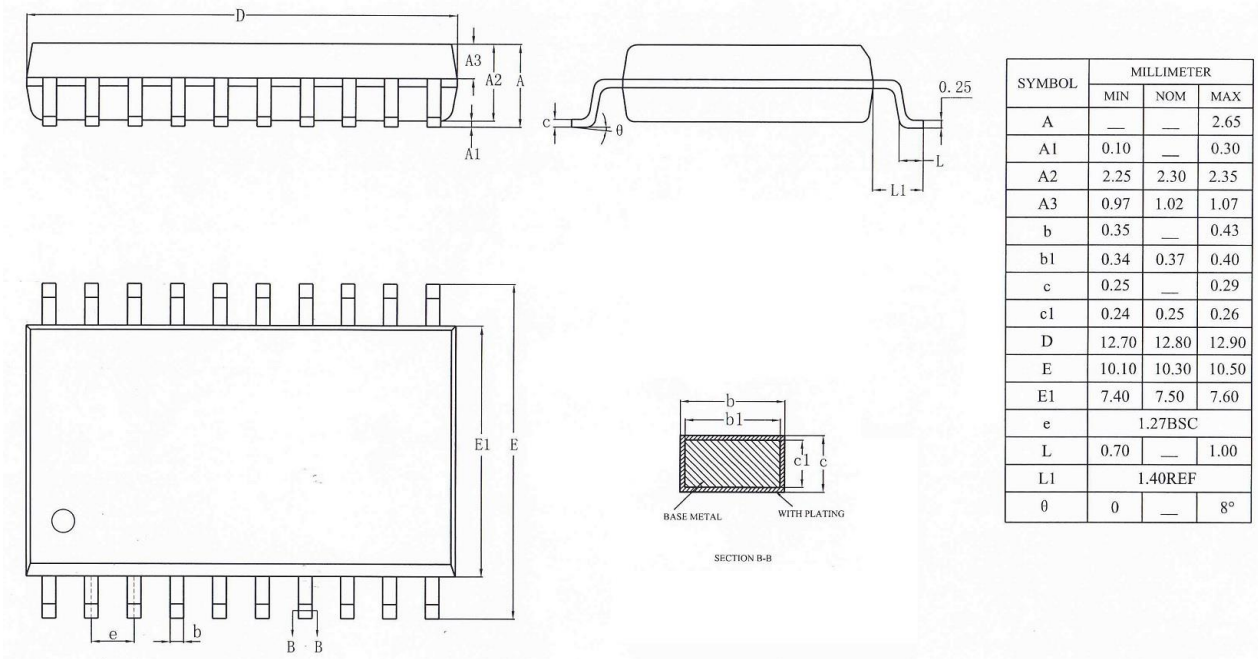


Shomcu.com

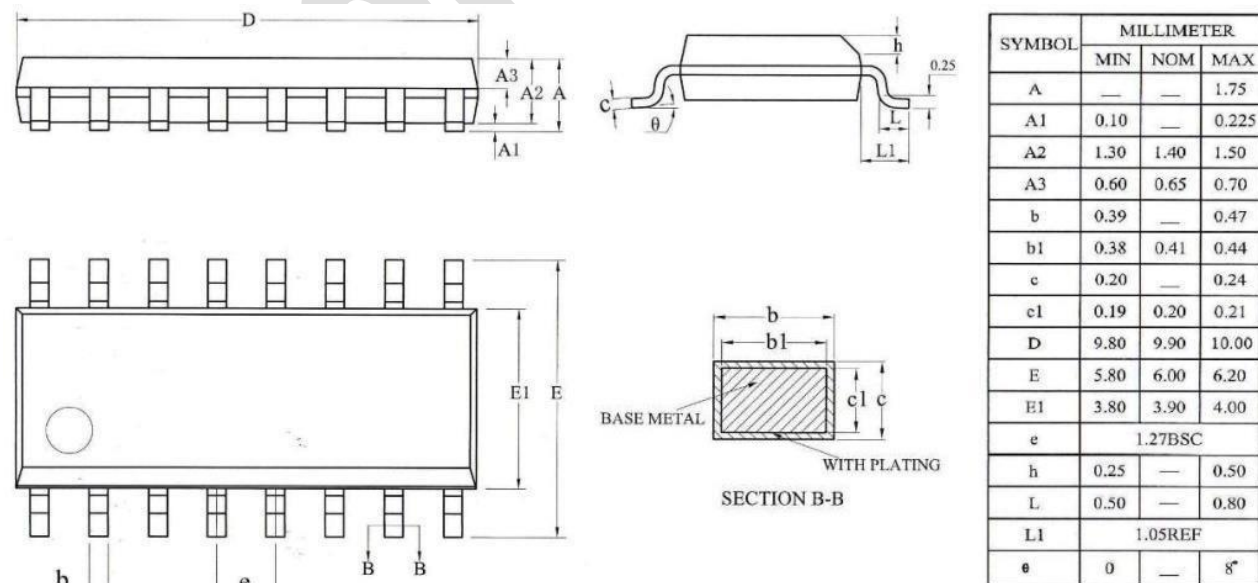


17 封装尺寸

17.1 SOP20



17.2 SOP16





18 修订记录

版本	日期	修订内容
V1.0	2022-10-13	发布初版；
V1.1	2023-04-17	调整引脚图中复用端口排序；完善 CPO 示意图；修正 PWM0 示意图中笔误；