

SinoMCU 8 位单片机

MC9931

用户手册

V1.2



目录

1	产品概要	4
1.1	产品特性	4
1.2	订购信息	5
1.3	引脚排列	5
1.4	端口说明	7
2	电气特性	8
2.1	极限参数	8
2.2	直流电气特性	8
2.3	交流电气特性	9
2.4	PFRC 特性参数	9
2.5	ADC 特性参数	10
3	CPU 及存储器	11
3.1	指令集	11
3.2	程序存储器	13
3.3	数据存储器	14
3.4	堆栈	15
3.5	控制寄存器	15
3.6	用户配置字	19
4	系统时钟	20
4.1	内部高频 RC 振荡器	20
4.2	内部低频 RC 振荡器	20
4.3	系统工作模式	21
4.4	低功耗模式	22
5	复位	24
5.1	复位条件	24
5.2	上电复位	25
5.3	外部复位	25
5.4	低电压复位	25
5.5	看门狗复位	25
6	I/O 端口	26
6.1	通用 I/O 功能	26
6.2	内部上/下拉电阻	27
6.3	端口模式控制	28
6.4	端口驱动控制	28
7	定时器 TIMER	30
7.1	看门狗定时器 WDT	30
7.2	定时器 T0	30
7.3	定时器 T1	32
7.4	定时器 T2	37
8	频率可编程 RC 振荡器 PFRC	41
9	模数转换器 ADC	42
9.1	ADC 概述	42

9.2	ADC 相关寄存器.....	43
9.3	ADC 操作步骤.....	45
9.4	ADC 零点偏移修调流程.....	46
10	低电压检测 LVD	47
11	中断.....	48
11.1	外部中断.....	48
11.2	定时器中断.....	48
11.3	键盘中断.....	48
11.4	ADC 中断.....	49
11.5	LVD 中断.....	49
11.6	中断相关寄存器.....	49
12	特性曲线.....	52
12.1	I/O 特性.....	52
12.2	功耗特性.....	56
12.3	模拟电路特性.....	60
13	封装尺寸.....	66
13.1	QFN20 (3×3)	66
13.2	QFN20 (4×4)	66
13.3	DFN8.....	67
14	修订记录.....	68

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 4/8/16/32/64 分频，系统低频时钟下 F_{CPU} 固定为 2 分频
- 存储器
 - ✧ 2K×16 位 OTP 型程序存储器，可通过间接寻址读取程序存储器内容
 - ✧ 128 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
- 2 组共 16 个 I/O
 - ✧ P0 (P00~P07), P1 (P10~P17)
 - ✧ P04 为输入/开漏输出口，可复用为外部复位 RST 输入，编程时为高压 V_{PP} 输入
 - ✧ 所有端口均内置上拉和下拉电阻，均可单独使能
 - ✧ P17 上拉电阻和 P16 下拉电阻复位初有效，并可配置复位后的初始状态
 - ✧ P0 所有端口均支持键盘中断唤醒功能，并可单独使能
 - ✧ P00-P01、P03-P02 之间内置 NMOS 电路，可通过寄存器位控制其关断/导通
- 系统时钟源
 - ✧ 内置高频 RC 振荡器 (32MHz)，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：CPU 停止运行，高频时钟源工作
 - ✧ HOLD2 模式：CPU 停止运行，高频时钟源停止工作，低频时钟源工作
 - ✧ 休眠模式：CPU 停止运行，所有时钟源停止工作
- 内部自振式看门狗计数器 (WDT)
 - ✧ 溢出时间可配置：16ms/64ms/256ms/1024ms
 - ✧ 工作模式可配置：始终开启、始终关闭、低功耗模式下关闭
- 3 个定时器
 - ✧ 16 位定时器 T0，可实现外部计数功能
 - ✧ 8 位定时器 T1，可实现外部计数、1 对互补且死区可设的 8+3 模式 PWM
 - ✧ 8 位定时器 T2，可实现 2 路共周期独立占空比的 PWM
- 1 个频率可编程 RC 振荡器 PFRC
 - ✧ 振荡频率复位初始值：32MHz
 - ✧ 最大调节范围：(32MHz -10%) ~ (32MHz +4%) (以实际芯片为准)
 - ✧ 输出时钟 F_{PFRC} 可用作定时器时钟源
- 1 个 12 位高精度逐次逼近型 ADC
 - ✧ 12 路外部通道：AN0~AN11；4 路内部通道：GND、V_{DD}/4、EVN0/4、EVN1/4
 - ✧ 参考电压可选：V_{DD}、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 32/64/128/256 分频
 - ✧ 支持零点校准

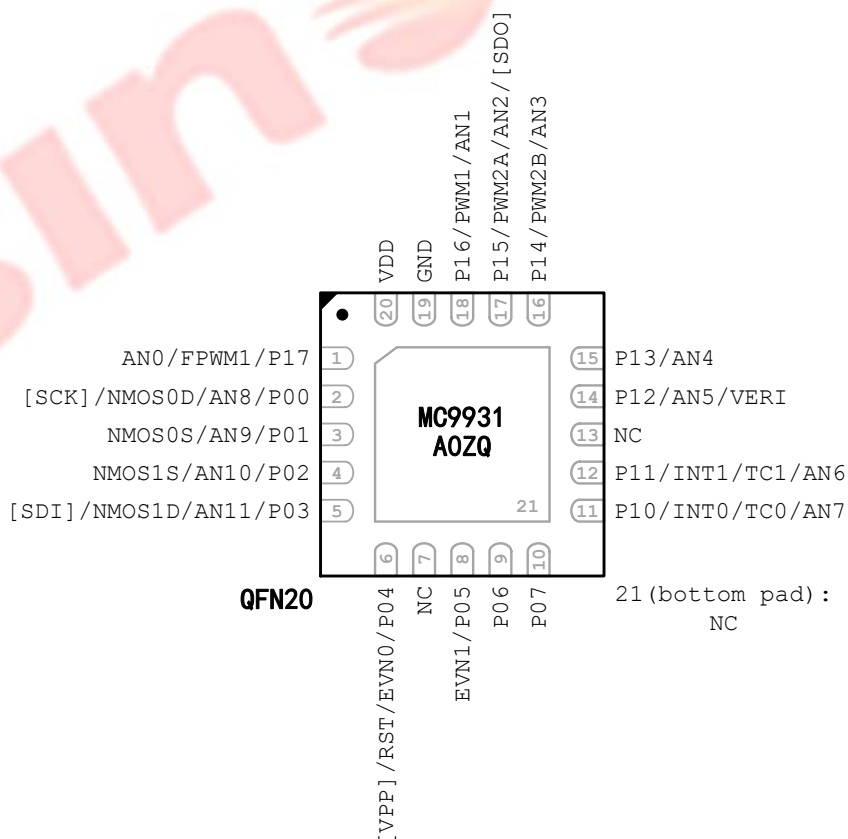
- 中断
 - ◇ 外部中断 (INT0~INT1), 键盘中断 (P00~P07), 定时器中断 (T0~T2)
 - ◇ ADC 中断, LVD 中断
- 低电压复位 LVR: 1.8V/2.0V/2.4V/2.7V/3.2V
- 低电压检测 LVD: 2.0V/2.2V/2.4V/2.6V/2.8V/3.2V/3.6V/4.0V
- 工作电压
 - ◇ $V_{LVR27} \sim 5.5V @ F_{cpu} = 0 \sim 8MHz$
 - ◇ $V_{LVR20} \sim 5.5V @ F_{cpu} = 0 \sim 4MHz$
 - ◇ $V_{LVR18} \sim 5.5V @ F_{cpu} = 0 \sim 500KHz$
- 封装形式: QFN20/DFN8

1.2 订购信息

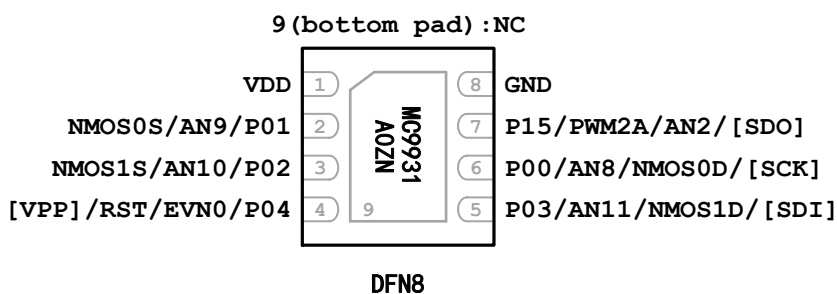
产品名称	封装形式	备注
MC9931A0ZQ	QFN20 (3×3)	
MC9931A0ZN	DFN8	
MC9931A0YE	QFN20 (4×4)	与 NMOS、PMOS 合封

1.3 引脚排列

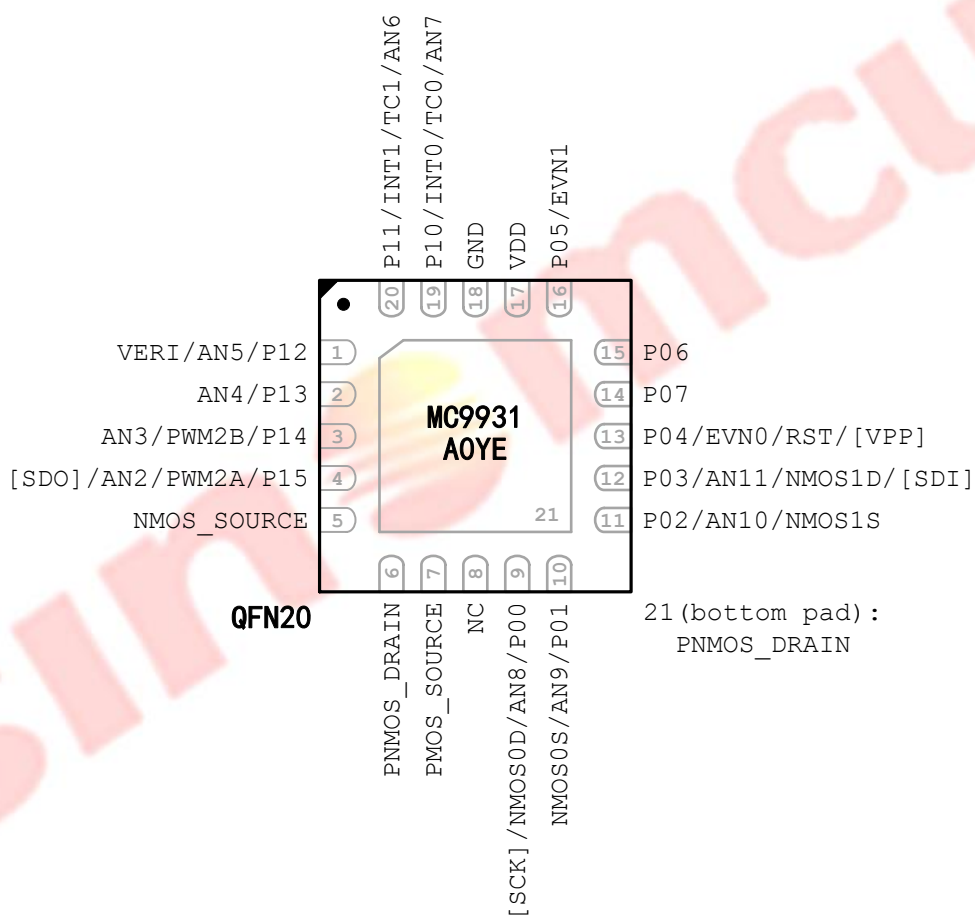
MC9931A0ZQ



MC9931A0ZN



MC9931A0YE

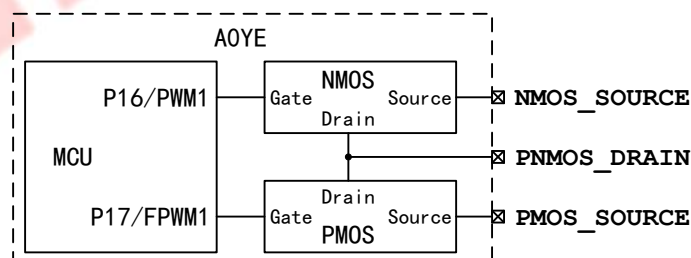


1.4 端口说明

端口名称 (MCU)	类型	功能说明
VDD	P	电源
GND	P	地
P0 (除 P04), P1	D	GPIO (推挽输出), 内部上/下拉
P04	D	GPIO (开漏输出), 内部上/下拉
INT0~INT1	DI	外部中断输入
TC0~TC1	DI	定时器 T0~T1 的外部计数输入
PWM1, FPWM1	DO	定时器 T1 的 PWM 及其互补输出
PWM2A~PWM2B	DO	定时器 T2 的 2 路 PWM 输出
AN0~AN11	AI	ADC 模拟输入通道
EVN0~EVN1	AI	ADC 内部 1/4 分压输入通道
VERI	AI	ADC 外部参考电压输入
NMOS0S, NMOS0D	P	内部 NMOS0 源极/漏极端口
NMOS1S, NMOS1D	P	内部 NMOS1 源极/漏极端口
RST	DI	外部复位输入
SCK, SDI, SDO	D	编程时钟/数据输入/数据输出接口
VPP	P	编程高压输入
端口名称 (PMOS/NMOS)	类型	功能说明
PMOS_SOURCE	P	合封 PMOS 源极端口
NMOS_SOURCE	P	合封 NMOS 源极端口
PNMOS_DRAIN	P	合封 PMOS/NMOS 漏极端口

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

合封产品引脚示意



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
输入电压	Vin	-0.3~VDD+0.3	V
工作温度	Ta	-40~85	°C
储存温度	Tstg	-65~150	°C
流入 VDD 最大电流	IVDDmax	60	mA
流出 GND 最大电流	IGNDmax	60	mA
流过 NMOS 最大电流	INMOSmax	150	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，则会影响其可靠性。

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	Fcpu=8MHz@FHIRC/4	VLVR27		5.5	V
			Fcpu=4MHz@FHIRC/8	VLVR20		5.5	
			Fcpu=2MHz@FHIRC/16	VLVR20		5.5	
			Fcpu=1MHz@FHIRC/32	VLVR20		5.5	
			Fcpu=500KHz@FHIRC/64	VLVR18		5.5	
			Fcpu=16KHz@FLIRC/2	VLVR18		5.5	
输入漏电流	Ileak	所有输入脚	VDD=5V	-1		1	uA
输入高电平	Vih	所有输入脚	SMTVS 配置	0.8VDD			V
			SMTVS 配置	2.0			V
输入低电平	Vil	所有输入脚	SMTVS 配置			0.2VDD	V
			SMTVS 配置			0.8	V
输出拉电流	Ioh	推挽输出脚	Voh=VDD-0.6V		20		mA
输出灌电流	Iol	所有输出脚	Vol=0.6V		30		mA
内置 NMOS 耐流值	Inmos	P00-P01 之间 P03-P02 之间	VDD=5V, 压降 0.2V	100			mA
内置 NMOS 电压降	Vnmos	P00-P01 之间 P03-P02 之间	VDD=5V, 电流 100mA			0.2	V
上拉电阻	Rpu1	P1,P0(除 P04)	VDD=5V, Vin=0	-25%	16	+25%	KΩ
	Rpu2	P04	VDD=5V, Vin=0		75		KΩ
下拉电阻	Rpd1	P1, P0(除 P00,P03,P04)	Vin=VDD=5V	-25%	16	+25%	KΩ

	Rpd2	P04	Vin=VDD=5V		75		KΩ
	Rpd3	P00, P03	Vin=VDD=5V, P00PDR/P03PDR=0	-25%	16	+25%	KΩ
运行模式功耗	Irun	VDD	Vin=VDD=5V, P00PDR/P03PDR=1		850		KΩ
			Fcpu=8MHz@HIRC		3.5		mA
			Fcpu=4MHz@HIRC		2.2		mA
			Fcpu=2MHz@HIRC		1.5		mA
			Fcpu=1MHz@HIRC		1.2		mA
			Fcpu=500KHz@HIRC		1.0		mA
			Fcpu=32KHz/2@LIRC		9		uA
HOLD1 功耗	Ihold1	VDD	CPU 停, HIRC/LIRC 开		800		uA
HOLD2 功耗	Ihold2	VDD	CPU 停, HIRC 关, LIRC 开		2.5		uA
休眠模式功耗	Istop	VDD	休眠模式, WDT/LVR 关		0.2	1	uA
			休眠模式, WDT 开, LVR 关		2.5	5	uA
			休眠模式, WDT 关, LVR 开		10	20	uA
低压检测电压	VLVD	VDD		-10%		+10%	V
低压复位电压	VLVR	VDD	LVRVS 配置	-5%		+5%	V
LVD/LVR 回滞电压		VDD			6%	12%	

注：条件项中，未注明模块默认关闭，无关端口状态为输出无负载、输入或开漏输出高则端口电压为 VDD/GND。

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=5V, T=25°C	-1.5%	32	+1.5%	MHz
		VDD=2.2V~5.5V, T=-20°C~70°C	-3%		+3%	
		VDD=2.2V~5.5V, T=-40°C~85°C	-5%		+5%	
LIRC 振荡频率	FLIRC	VDD=5V, T=25°C	-50%	32	+50%	KHz

2.4 PFRC 特性参数

特性	符号	条件	最小	典型	最大	单位
PFRC 振荡频率	FPFRC	VDD=5V, T=25°C	-1.5%	32	+1.5%	MHz
		VDD=2.2V~5.5V, T=-20°C~70°C	-3%		+3%	
		VDD=2.2V~5.5V, T=-40°C~85°C	-5%		+5%	

2.5 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	2.5		5.5	V
积分非线性误差	INL	V _{REF} =VDD, F _{ADC} =1MHz, T _{con} =27us			±4	LSB
微分非线性误差	DNL	V _{REF} =VDD, F _{ADC} =1MHz, T _{con} =27us			±2	LSB
零点偏移误差	EZ	V _{REF} =VDD, F _{ADC} =1MHz, T _{con} =27us			±4	LSB
增益误差	ET	V _{REF} =VDD, F _{ADC} =1MHz, T _{con} =27us			±4	LSB
转换时钟	F _{ADC}	VDD=5V			1	MHz
转换时间	T _{con}		14		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	uA
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	uA
模拟信号源推荐阻抗	Z _{AIN}				10	KΩ
内部 1/4 分压电阻总值	R _{EV1}	V _{in} =VDD=2.5V~5.5V	32	48	64	KΩ
电阻分压比值			-1%	1/4	+1%	V _{EV1}
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	-1.5%		+1.5%	
		选择内部参考电压 V _{IR} , T=-20°C~70°C	-3%	2/3/4	+3%	
		选择内部参考电压 V _{IR} , T=-40°C~85°C	-5%		+5%	
		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		VDD	V

3 CPU 及存储器

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，其余指令均为单周期指令，即执行时间为 1 个指令周期；所有指令均为单字指令，即指令码只占用 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 ACC 相加，结果存入 ACC	$R + ACC \rightarrow ACC$	1	1	C,DC,Z
ADDRA R	R 和 ACC 相加，结果存入 R	$R + ACC \rightarrow R$	1	1	C,DC,Z
ADCAR R	R 和 ACC 相加（带 C 标志），结果存入 ACC	$R + ACC + C \rightarrow ACC$	1	1	C,DC,Z
ADCRA R	R 和 ACC 相加（带 C 标志），结果存入 R	$R + ACC + C \rightarrow R$	1	1	C,DC,Z
RSUBAR R	R 和 ACC 相减，结果存入 ACC	$R - ACC \rightarrow ACC$	1	1	C,DC,Z
RSUBRA R	R 和 ACC 相减，结果存入 R	$R - ACC \rightarrow R$	1	1	C,DC,Z
RSBCAR R	R 和 ACC 相减（带 C 标志），结果存入 ACC	$R - ACC - C \rightarrow ACC$	1	1	C,DC,Z
RSBCRA R	R 和 ACC 相减（带 C 标志），结果存入 R	$R - ACC - C \rightarrow R$	1	1	C,DC,Z
ASUBAR R	ACC 和 R 相减，结果存入 ACC	$ACC - R \rightarrow ACC$	1	1	C,DC,Z
ASUBRA R	ACC 和 R 相减，结果存入 R	$ACC - R \rightarrow R$	1	1	C,DC,Z
ASBCAR R	ACC 和 R 相减（带 C 标志），结果存入 ACC	$ACC - R - C \rightarrow ACC$	1	1	C,DC,Z
ASBCRA R	ACC 和 R 相减（带 C 标志），结果存入 R	$ACC - R - C \rightarrow R$	1	1	C,DC,Z
ANDAR R	R 和 ACC 与操作，结果存入 ACC	$R \text{ and } ACC \rightarrow ACC$	1	1	Z
ANDRA R	R 和 ACC 与操作，结果存入 R	$R \text{ and } ACC \rightarrow R$	1	1	Z
ORAR R	R 和 ACC 或操作，结果存入 ACC	$R \text{ or } ACC \rightarrow ACC$	1	1	Z
ORRA R	R 和 ACC 或操作，结果存入 R	$R \text{ or } ACC \rightarrow R$	1	1	Z
XORAR R	R 和 ACC 异或操作，结果存入 ACC	$R \text{ xor } ACC \rightarrow ACC$	1	1	Z
XORRA R	R 和 ACC 异或操作，结果存入 R	$R \text{ xor } ACC \rightarrow R$	1	1	Z
COMAR R	对 R 取反，结果存入 ACC	$R \text{ 取反} \rightarrow ACC$	1	1	Z
COMR R	对 R 取反，结果存入 R	$R \text{ 取反} \rightarrow R$	1	1	Z
RLA	ACC 循环左移（带 C 标志）	$ACC[7] \rightarrow C$ $ACC[6:0] \rightarrow ACC[7:1]$ $C \rightarrow ACC[0]$	1	1	C
RLAR R	R 循环左移（带 C 标志），结果存入 ACC	$R[7] \rightarrow C$ $R[6:0] \rightarrow ACC[7:1]$ $C \rightarrow ACC[0]$	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	ACC 循环右移（带 C 标志）	$C \rightarrow ACC[7]$ $ACC[7:1] \rightarrow ACC[6:0]$ $ACC[0] \rightarrow C$	1	1	C

RRAR	R	R 循环右移 (带 C 标志), 结果存入 ACC	R[0]→C R[7:1]→ACC[6:0] C→ACC[7]	1	1	C
RRR	R	R 循环右移 (带 C 标志), 结果存入 R	R[0]→C R[7:1]→R[6:0] C→R[7]	1	1	C
SWAPAR	R	交换 R 的高低半字节, 结果存入 ACC	R[7:4]→ACC[3:0] R[3:0]→ACC[7:4]	1	1	-
SWAPR	R	交换 R 的高低半字节, 结果存入 R	R[7:4]→R[3:0] R[3:0]→R[7:4]	1	1	-
MOVRA	R	将 ACC 存入 R	ACC→R	1	1	-
MOVAR	R	将 R 存入 ACC	R→ACC	1	1	Z
MOVR	R	将 R 存入 R	R→R	1	1	Z
CLRA		清零 ACC	0→ACC	1	1	Z
CLRR	R	清零 R	0→R	1	1	Z
INCA		ACC 自加 1	ACC+1→ACC	1	1	-
INCR	R	R 自加 1	R+1→R	1	1	Z
INCAR	R	R 加 1, 结果存入 ACC	R+1→ACC	1	1	Z
DECA		ACC 自减 1	ACC-1→ACC	1	1	-
DECR	R	R 自减 1	R-1→R	1	1	Z
DECAR	R	R 减 1, 结果存入 ACC	R-1→ACC	1	1	Z
JZA		ACC 自加 1; 结果为 0 则跳过下一条指令	ACC+1→ACC; 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 ACC; 结果为 0 则跳过下一条指令	R+1→ACC; 结果为 0 则 PC+2→PC	1/2	1	-
DJZA		ACC 自减 1; 结果为 0 则跳过下一条指令	ACC-1→ACC; 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 ACC; 结果为 0 则跳过下一条指令	R-1→ACC; 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	K	K 和 ACC 相加, 结果存入 ACC	K+ACC→ACC	1	1	C,DC,Z
ADCAI	K	K 和 ACC 相加 (带 C 标志), 结果存入 ACC	K+ACC+C→ACC	1	1	C,DC,Z
ISUBAI	K	K 和 ACC 相减, 结果存入 ACC	K-ACC→ACC	1	1	C,DC,Z
ISBCAI	K	K 和 ACC 相减 (带 C 标志), 结果存入 ACC	K-ACC-/C→ACC	1	1	C,DC,Z
ASUBAI	K	ACC 和 K 相减, 结果存入 ACC	ACC-K→ACC	1	1	C,DC,Z
ASBCAI	K	ACC 和 K 相减 (带 C 标志), 结果存入 ACC	ACC-K-/C→ACC	1	1	C,DC,Z
ANDAI	K	K 和 ACC 与操作, 结果存入 ACC	K and ACC→ACC	1	1	Z
ORAI	K	K 和 ACC 或操作, 结果存入 ACC	K or ACC→ACC	1	1	Z
XORAI	K	K 和 ACC 异或操作, 结果存入 ACC	K xor ACC→ACC	1	1	Z
MOVAI	K	将 K 存入 ACC	K→ACC	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 K	从子程序返回, 并将 K 存入 ACC	TOS→PC K→ACC	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 码加法后, 将 ACC 的值调整为 BCD 码	ACC(HEX 码)→ACC(BCD 码)	1	1	C
DSA	BCD 码减法后, 将 ACC 的值调整为 BCD 码	ACC(HEX 码)→ACC(BCD 码)	1	1	-
CLRWDT	清零看门狗定时器	0→WDT	1	1	TO,PD
STOP	进入低功耗模式	0→WDT; CPU 暂停	1	1	TO,PD

注:

- 1、ACC-算术逻辑单元累加器, R-数据存储器, K-立即数;
- 2、对于条件跳转类指令, 若跳转条件成立, 则指令需 2 个周期, 否则只需 1 个周期;
- 3、对于访问数据寄存器 R 且影响 C,DC,Z 标志的指令, 不能用该指令访问寄存器 PFLAG;

3.2 程序存储器

芯片的程序存储器为 OTP 型存储器, 2K×16 位的地址空间范围为 0000H~07FFH。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 - 07FFH)

程序存储器支持间接寻址, 可通过寄存器 INDF3 访问 FSR1×256+FSR0 指向的程序存储器地址中的内容。例: 通过间接寻址读取程序存储器 0155H 地址中的内容, 高 8 位存入数据存储器 11H 地址, 低 8 位存入数据存储器 10H 地址

```

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 指向的程序存储器地址 (0155H)
                        ; 中的内容, 高 8 位存入 HIBYTE, 低 8 位存入 A 寄存器
MOVRA    10H           ; 低 8 位存入数据存储器 10H 地址
MOVAR    HIBYTE        ; 从 HIBYTE 读取高 8 位
MOVRA    11H           ; 高 8 位存入数据存储器 11H 地址

```

3.3 数据存储器

数据存储器包括通用数据存储器 GPR 和特殊功能寄存器 SFR，具体地址分配参照下表。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~07FH	GPR	通用数据存储器区							
080H~17FH	保留	保留							
180H~187H	SFR	INDF0	INDF1	INDF2	HIBYTE	FSR0	FSR1	PCL	PFLAG
188H~18FH		MCR	INDF3	INTE	INTF	OSCMR	LVDCR	DRVCR	
190H~197H		IOP0	OEP0	PUP0	PDP0	IOP1	OEP1	PUP1	PDP1
198H~19FH		T0CR	T0CNTH	T0CNTL	T0LOADH	T0LOADL		PFRCCR	PFRCCAL
1A0H~1A7H		T1CR	T1CNT	T1LOAD	T1DATA	PWM1CR0	PWM1CR1	PWM1CR2	
1A8H~1AFH		T2CR	T2CNT	T2LOAD	T2DATA	T2DATB	PWM2CR		
1B0H~1B7H		ADCR0	ADCR1	ADRH	ADRL	OSADJCR	P0ADCR	P1ADCR	P0KBCR
1B8H~1FFH	保留								

注：上表中灰色部分地址为系统保留区，读出数据不确定，写入操作可能会影响芯片正常工作。

数据存储器寻址模式地址组成

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	寻址模式
0	0	0	0	0	0	0	来自指令的 9 位地址									直接寻址模式
0	0	0	0	0	0	0	0	FSR0								间接寻址模式 0
0	0	0	0	0	0	0	1	FSR1								间接寻址模式 1
FSR1							FSR0									间接寻址模式 2

直接寻址模式，是以指令的低 9 位为数据存储器地址，通过指令访问，寻址范围为 000H~1FFH。

例：通过直接寻址模式将数据 55H 写入数据存储器 010H 地址

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

间接寻址模式 0，是以 FSR0 为数据存储器地址指针，通过 INDF0 访问，寻址范围为 000H~0FFH。

例：通过间接寻址模式 0 将数据 55H 写入数据存储器 010H 地址

```
MOVAI    10H
MOVRA    FSR0
MOVAI    55H
MOVRA    INDF0        ; 将数据 55H 写入 FSR0 指向的数据存储器中
```

间接寻址模式 1，是以 FSR1 为数据存储器地址指针，通过 INDF1 访问，寻址范围为 100H~1FFH。

例：通过间接寻址模式 1 将数据 55H 写入数据存储器 110H 地址

```
MOVAI    10H
```


MOVRA FSR1
MOVAI 55H
MOVRA INDF1 ; 将数据 55H 写入 FSR1 指向的数据存储器中

间接寻址模式 2, 是以 [FSR1:FSR0] 为数据存储器地址指针, 通过 INDF2 访问, 寻址范围为 0000H ~ FFFFH。例: 通过间接寻址模式 2 将数据 55H 写入数据存储器 0110H 地址

MOVAI 01H
MOVRA FSR1
MOVAI 10H
MOVRA FSR0
MOVAI 55H
MOVRA INDF2 ; 将数据 55H 写入 FSR1×256+FSR0 指向的数据存储器中

注: 间接寻址模式 2 可寻址 0~FFFFH, 但访问数据存储器中未定义的地址时, 读出数据不确定, 写入数据可能会更改其他地址中的数据。

3.4 堆栈

8 级堆栈深度, 当程序响应中断或执行子程序调用指令时 CPU 会将 PC 自动压栈保存; 当执行中断返回指令或子程序返回指令时, 栈顶数据自动出栈载入 PC。

3.5 控制寄存器

数据指针寄存器

	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: 间接寻址模式 0 的指针, 或间接寻址模式 2、3 的指针低 8 位。

	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: 间接寻址模式 1 的指针, 或间接寻址模式 2、3 的指针高 8 位。

间接寻址寄存器

	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 寻址实际是对 FSR0 指向的数据存储器地址进行访问，从而实现间接寻址功能。

	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 寻址实际是对 FSR1+256 指向的数据存储器地址进行访问，从而实现间接寻址功能。

	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 寻址实际是对 FSR1×256+FSR0 指向的数据存储器地址进行访问，从而实现间接寻址功能。

	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 寻址实际是对 FSR1×256+FSR0 指向的程序存储器地址进行访问，从而实现间接寻址功能。

注：寄存器 INDF3 仅可使用读取指令 (MOVAR INDF3) 进行读取操作，读取内容高 8 位存入 HIBYTE 寄存器，低 8 位存入 A 寄存器。

字操作高字节缓存器

	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]** – 字操作高字节缓存器

HIBYTE: 用于存放通过 INDF3 访问程序存储器时所读取内容的高 8 位数据。

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；
- ✧ 分支指令 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 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/借位标志位

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

BIT[0] **C** – 进/借位标志位

- 0：加法运算时无进位；减法运算时有借位；移位后移出逻辑 0；
- 1：加法运算时有进位；减法运算时无借位；移位后移出逻辑 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	INT0M1	INT0M0
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
10	电平变化触发
11	

BIT[1:0] **INT0M[1:0]** – 外部中断 INT0 触发方式选择位

INT0M[1:0]	INT0 触发方式
00	上升沿触发
01	下降沿触发
10	电平变化触发
11	

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中，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。

芯片的用户配置字，定义如下：

符号	功能说明
FCPUS	高频时钟下 F _{CPU} 分频选择： F _{CPU} =F _{HOSC} /4; F _{CPU} =F _{HOSC} /8; F _{CPU} =F _{HOSC} /16; F _{CPU} =F _{HOSC} /32; F _{CPU} =F _{HOSC} /64;
RSTEN	RST 外部复位设置： P04 为外部复位脚; P04 为输入/输出脚;
LVRVS	LVR 复位电压选择：(LVR 电压应满足由 F _{CPU} 决定的工作电压特性) 1.8V; 2.0V; 2.4V; 2.7V; 3.2V;
LVRMD	LVR 模式设置： LVR 始终开启; LVR 在运行模式下开启，在低功耗模式下关闭;
WDTM	WDT 模式设置： WDT 始终关闭; WDT 在低功耗模式下关闭; WDT 始终开启;
WDTT	WDT 溢出时间选择： 16ms; 64ms; 256ms; 1024ms;
SMTVS	端口施密特阈值选择： 2.0V/0.8V; 0.8VDD/0.2VDD;
P17RUC	P17 上拉电阻复位后初态设置： P17 上拉电阻复位后有效; P17 上拉电阻复位后无效;
P16RDC	P16 下拉电阻复位后初态设置： P16 下拉电阻复位后有效; P16 下拉电阻复位后无效;
ENCR	程序代码加密设置： 程序代码加密; 程序代码不加密;

4 系统时钟

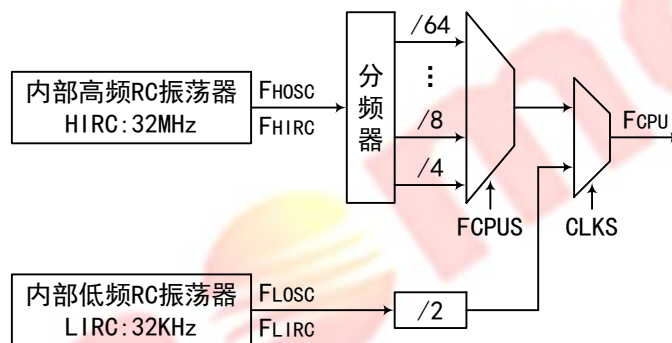
芯片为双时钟系统，内部电路均在系统高频时钟 FHOSC 或系统低频时钟 FLOSC 下工作，部分模块的时钟还可在 FHOSC 和 FLOSC 之间切换。

系统高频时钟 FHOSC 固定为内部高频 RC 振荡器 HIRC（32MHz）时钟 FHIRC；系统低频时钟 FLOSC 固定为内部低频 RC 振荡器 LIRC（32KHz）时钟 FLIRC。

CPU 为双时钟，时钟源可在系统高频时钟 FHOSC 和系统低频时钟 FLOSC 之间切换。选择 FHOSC 时，CPU 的时钟频率 F_{CPU} 通过配置字 FCPUS 选择；选择 FLOSC 时，F_{CPU} 固定为 FLOSC 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。

系统时钟示意图



4.1 内部高频 RC 振荡器

芯片内置 1 个振荡频率为 32MHz 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、WDT 定时器等电路。

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。

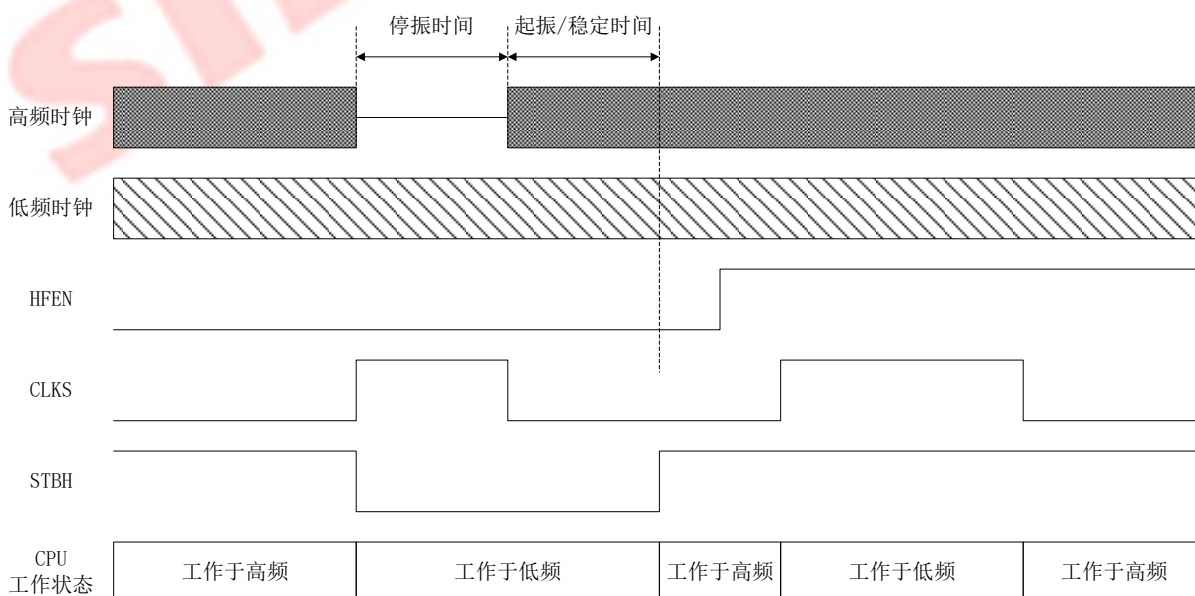
工作模式	切入条件	系统状态
高速	任意模式下，系统复位	CPU 高速运行，高/低频时钟源均工作
	低速模式下，CLKS 清 0	
	HOLD1/HOLD2/休眠模式下，CPU 唤醒	
低速	高速模式下，CLKS 置 1	CPU 低速运行，高频时钟源由 HFEN 决定
	HOLD1/HOLD2/休眠模式下，CPU 唤醒	
HOLD1	高/低速模式下，HFEN 置 1，执行 STOP 指令	CPU 暂停，高频时钟源工作，低频时钟源由 LFEN 决定
HOLD2	高/低速模式下，HFEN 清 0，LFEN 置 1，执行 STOP 指令	CPU 暂停，高频时钟源停止，低频时钟源工作
休眠	高/低速模式下，HFEN 清 0，LFEN 清 0，执行 STOP 指令	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



工作模式寄存器

	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, 低频时钟源停止工作;
- 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, 高频时钟源停止工作;
- 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态不变;
- ✧ 定时器若其时钟源未停止, 则可以保持继续工作;

以下情况可使系统退出低功耗模式:

- ✧ 上电复位;
- ✧ 外部复位 (若有外部复位功能);
- ✧ 有 WDT 溢出 (若低功耗模式下 WDT 保持继续工作);
- ✧ 有外部中断请求发生 (若有外部中断功能);

- ◇ 定时器溢出中断发生（若低功耗模式下定时器保持继续工作）；
- ◇ 有键盘中断请求发生（若有键盘中断功能）；

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

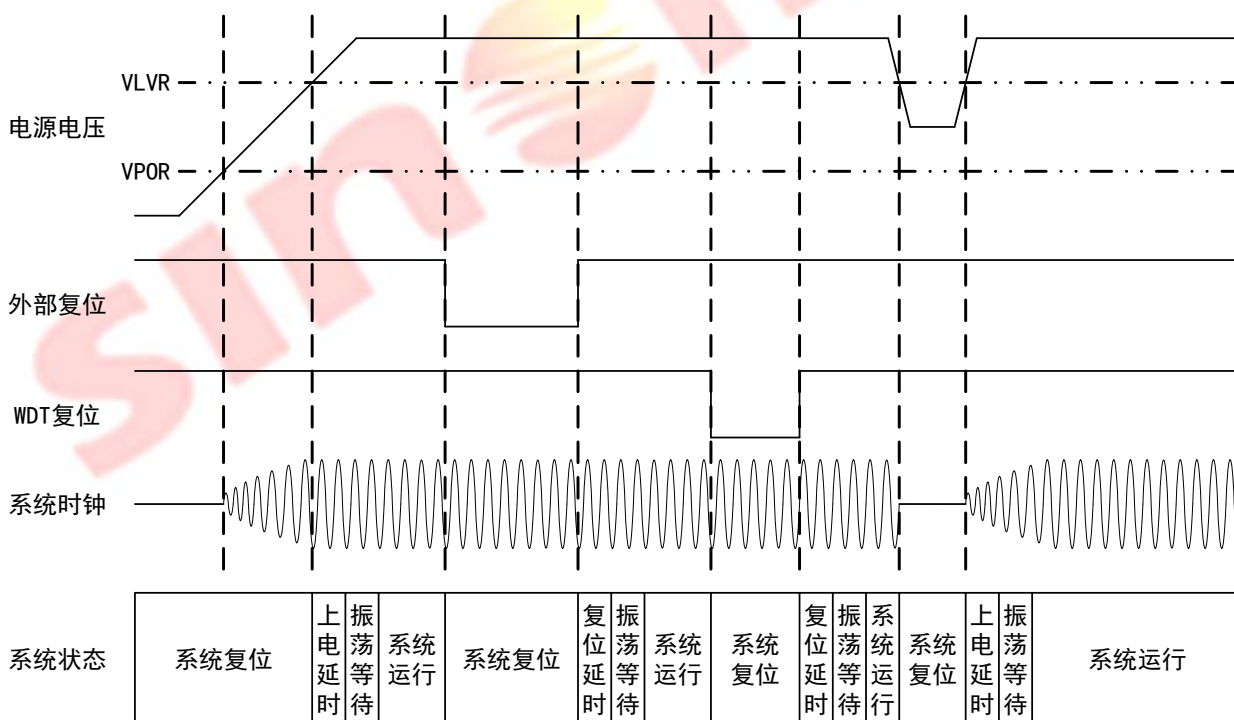
芯片共有如下几种复位方式：

- ✧ 上电复位 POR；
- ✧ 外部复位；
- ✧ 低电压复位 LVR；
- ✧ 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 开始重新从程序存储器 0000H 地址处执行指令。

上电复位 POR 和低电压复位 LVR 会关闭系统主时钟振荡器，复位解除后才重新打开振荡器，因为振荡器起振和稳定需要一定的时间，所以系统会保持一定时间的上电延时和振荡等待后才开始工作；而外部复位和 WDT 复位不会关闭主时钟振荡器，复位解除时系统会在较短的复位延时和振荡等待后即开始工作。

下图是复位产生和系统工作状态之间时序关系的示意图：



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，则应在复位后进行软件延时，以确保芯片开始工作时 VDD 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快速、慢速上电的情况，且当芯片上电过程中出现电源电压抖动时均能保证系统可靠的复位。

上电复位过程可以概括为以下几个步骤：

- (1) 检测系统工作电压，等待电压高于上电复位电压 V_{POR} 并保持稳定；
- (2) 若有 LVR 功能，则需等待电压高于 V_{LVR} 并保持稳定；
- (3) 若有外部复位功能，则需等待复位引脚电压高于 V_{ih} ；
- (4) 初始化所有寄存器；
- (5) 开启主时钟振荡器，并等待一段时间以待振荡器稳定；
- (6) 上电结束，系统开始执行指令。

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开启外部复位功能，端口的内部上拉电阻自动使能。外部复位输入端口 RST 为施密特结构，低电平有效。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可通过配置字 LVRVS 选择。电压检测电路有一定的回滞特性，回滞电压为 6% 左右（典型值），当电源电压下降至 LVR 电压时 LVR 复位有效，反之则电源电压需上升至 LVR 电压 +6% 后 LVR 复位才解除。

5.5 看门狗复位

看门狗（WDT）复位是一种对程序正常运行的保护机制。正常情况下，用户程序需定时对 WDT 定时器进行清零操作，以保证 WDT 不溢出。若出现异常情况，程序未按时对 WDT 定时器清零，则芯片会因 WDT 溢出而产生看门狗复位，系统重新初始化，返回受控状态。

注：低功耗模式下 CPU 暂停工作，若此时有 WDT 溢出，则仅唤醒 CPU 而不产生复位。

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。所有 I/O 端口均支持施密特输入，除 P04 外的 I/O 端口均支持推挽输出。除用作通用数字 I/O 端口外，部分端口还具有外部中断输入、PWM 输出、ADC 模拟输入、或触摸按键输入等复用功能。

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)

注：开漏输出口输出高电平时，若端口电压不等于 VDD 电压则会产生漏电流。

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输入电平状态；
- 1: 端口作为输出口，读端口操作将读取端口的数据寄存器值；

	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输入电平状态；
- 1: 端口作为输出口，读端口操作将读取端口的数据寄存器值；

6.2 内部上/下拉电阻

所有端口均有内部上拉和下拉电阻，且均有单独的寄存器位控制其上/下拉电阻在端口处于输入状态时是否有效。端口处于输出状态时，上/下拉电阻及其控制位无效。

P17 上拉电阻和 P16 下拉电阻复位初有效，并可配置复位后的初始状态。

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	1-U	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;

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	1-U	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;

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
P0ADCR	-	-	P05DC	P04DC	P03DC	P02DC	P01DC	P00DC
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] **P0nDC** – P0n 端口数字功能控制位（n=5-0）

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；

6.4 端口驱动控制

芯片在 P00 与 P01 之间、P03 与 P02 之间内置有 NMOS 电路，可通过寄存器位控制其关断/导通。

端口驱动控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DRVCR	-	-	-	-	P03PDR	P00PDR	NMOS1ON	NMOS0ON
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3] **P03PDR** – P03 端口下拉电阻选择位

0: 端口内部下拉电阻为 16K Ω ；

1: 端口内部下拉电阻为 850K Ω ；

BIT[2] **P00PDR** – P00 端口下拉电阻选择位

0: 端口内部下拉电阻为 16K Ω ；

1: 端口内部下拉电阻为 850K Ω ；

BIT[1]	NMOS1ON – P03-P02 之间 NMOS 管控制位 0: 内部 NMOS 管关断; 1: 内部 NMOS 管导通;
BIT[0]	NMOS0ON – P00-P01 之间 NMOS 管控制位 0: 内部 NMOS 管关断; 1: 内部 NMOS 管导通;

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，WDT 计数器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。若选择始终开启，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；若选择低功耗模式下关闭，则 WDT 在休眠/HOLD 模式下自动停止、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。

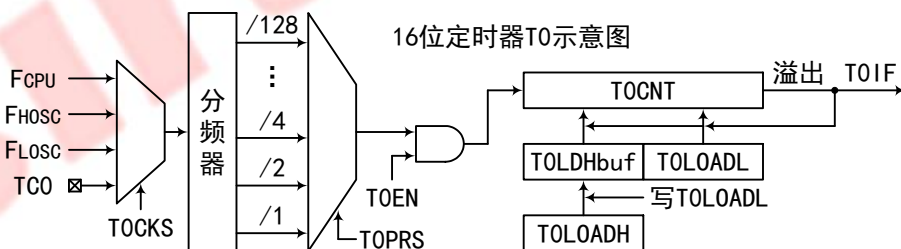
WDT 溢出时间可配置为 16ms/64ms/256ms/1024ms。

注：WDT 溢出时间为典型值，实际值偏差大，必须保证清 WDT 时间小于典型值的 1/4。

7.2 定时器 T0

定时器 T0 为 16 位定时/计数器，包含 1 个 16 位递减计数器、可编程预分频器、控制寄存器、16 位重载寄存器。

- ✧ 可通过预分频比设置计数频率，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 T0，可通过寄存器位 T0CKS 选择时钟源，通过 T0PRS 选择预分频比，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）。预分频比可选择 1~128 分频，对 TOCNT 的写操作将清零预分频计数器，而预分频比保持不变。16 位 TOCNT 的高字节 TOCNTH 配有读/写缓冲器，写 TOCNT 时需先写 TOCNTH 再写 TOCNTL，而读 TOCNT 时则需先读 TOCNTL 再读 TOCNTH。

当 T0EN=0 时，TOCNT 保持不变，写重载寄存器 TOLOAD 将立即载入 TOCNT 中；当 T0EN=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 T0IF 将被置 1，同时 T0 自动将当前 TOLOAD 值载入 TOCNT 中重新开始计数。

16 位 TOLOAD 配有 1 个 8 位的高位缓冲器（TOLDHbuf），写 TOLOADL 时会同时将 TOLOADH 值

载入该缓冲器中。此时若 T0EN=0，则会同时再将[缓冲器:T0LOADL]载入 T0CNT；若 T0EN=1，则需在 T0 溢出后才将[缓冲器:T0LOADL]的值载入 T0CNT。调整 T0LOAD 值时需先写 T0LOADH，再写 T0LOADL。

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
T0CR	T0EN	保留	-	T0CKS1	T0CKS0	T0PRS2	T0PRS1	T0PRS0
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] **T0EN** – 定时器 T0 使能位

0: 关闭定时器 T0;

1: 开启定时器 T0;

BIT[6] 保留位，需固定写“0”

BIT[4:3] **T0CKS[1:0]** – T0 时钟源选择位

T0CKS[1:0]	T0 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	TC0 上升沿

BIT[2:0] **T0PRS[2:0]** – T0 预分频比选择位

T0PRS[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
T0CNTH	T0CNT15	T0CNT14	T0CNT13	T0CNT12	T0CNT11	T0CNT10	T0CNT9	T0CNT8
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] **T0CNT[15:8]** – T0 计数器高 8 位，为可读写的递减计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0CNTL	T0CNT7	T0CNT6	T0CNT5	T0CNT4	T0CNT3	T0CNT2	T0CNT1	T0CNT0

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] T0CNT[7:0] – T0 计数器低 8 位，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
T0LOADH	T0LOAD15	T0LOAD14	T0LOAD13	T0LOAD12	T0LOAD11	T0LOAD10	T0LOAD9	T0LOAD8
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] T0LOAD[15:8] – T0 重载寄存器高 8 位，用于设置 T0 的计数周期

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0LOADL	T0LOAD7	T0LOAD6	T0LOAD5	T0LOAD4	T0LOAD3	T0LOAD2	T0LOAD1	T0LOAD0
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

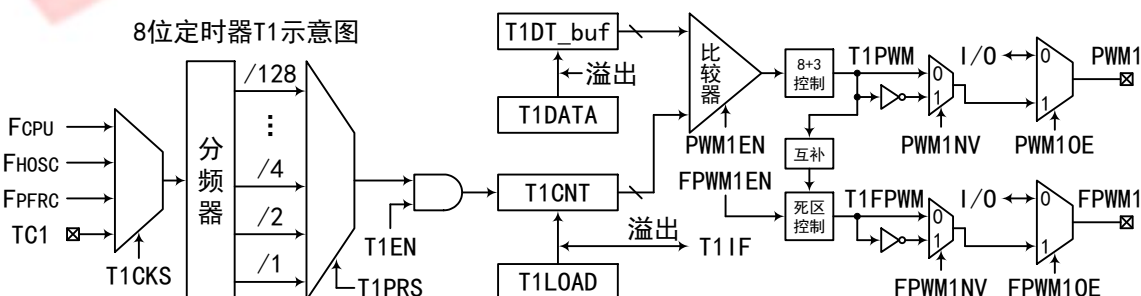
BIT[7:0] T0LOAD[7:0] – T0 重载寄存器低 8 位，用于设置 T0 的计数周期

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。

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 位重载寄存器及比较寄存器。

- ◇ 可通过预分频比设置计数频率，可通过重载寄存器控制计数周期；
- ◇ 支持 8+3 模式 PWM 输出，可通过比较寄存器设置 PWM 占空比；
- ◇ 可扩展为 1 对死区可设的互补 PWM 输出；
- ◇ 时钟源可选 F_{PFRC}；
- ◇ 支持溢出中断和溢出唤醒功能；



定时器 T1，可通过寄存器位 T1CKS 选择时钟源，通过 T1PRS 选择预分频比，所选时钟源通过预分频器后产生 T1 计数器 T1CNT 的计数时钟（上升沿计数）。预分频比可选择 1~128 分频，对 T1CNT 的写操作将清零预分频计数器，而预分频比保持不变。

当 T1EN=0 时, T1CNT 保持不变, 写重载寄存器 T1LOAD 将立即载入 T1CNT 中; 当 T1EN=1 时, T1CNT 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 T1IF 将被置 1, 同时 T1 自动将当前 T1LOAD 值载入 T1CNT 中重新开始计数。

如图所示, 定时器 T1 可实现 8+3 模式 PWM 功能(PWM1), 可通过寄存器位使能/关闭 PWM 功能, 或控制端口是否输出 PWM 波形。PWM1 关闭时 T1PWM 信号为低电平。PWM1 使能后, T1CNT 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 T1DATA 相等时, T1PWM 变为高电平; 当计数溢出时, T1PWM 变为低电平。

定时器 T1 可由 T1PWM 扩展生成 1 路带死区控制的互补 PWM 信号 T1FPWM, 从而扩展为 1 对死区可设的互补 PWM。若 FPWM1EN=0, 则 T1FPWM 信号仅为 T1PWM 信号的互补信号(无死区), 需 FPWM1EN=1 才能实现死区(即 2 路互补信号非交叠时间)控制。

T1DATA 配有 1 个 8 位的比较缓冲器(T1DT_buf)用于与 T1CNT 比较, PWM1 关闭时写 T1DATA 将立即载入比较缓冲器中, 而 PWM1 使能后写 T1DATA 则将在 T1 溢出时才载入比较缓冲器中。需注意在 FPWM1 使能但 PWM1 关闭时, 写 T1DATA 将导致比较缓冲器中数据变化且不确定。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。

T1PWM 信号的占空比计算如下:

- ✧ 高电平时间 = (T1DATA) × T1CNT 计数时钟周期
- ✧ 周期 (T1 的溢出周期) = (T1LOAD+1) × T1CNT 计数时钟周期
- ✧ 占空比 = 高电平时间 / 周期 = (T1DATA) / (T1LOAD+1)

PWM1/FPWM1 的占空比可通过 PWM1DB 位微调半个计数时钟。PWM1DB=1 时, PWM1 在计数时钟的触发沿(上升沿)变为高电平(延展周期则正常延展半个时钟); PWM1DB=0 时, PWM1 滞后半个时钟才翻转为高电平(延展周期则变为不延展)。FPWM1 翻转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。

PWM1/FPWM1 支持 8+3 模式, 可在普通模式 PWM 的时钟下增加占空比的调节精度, 以每 8 个普通 PWM 周期作为一个大周期, 通过占空比延展控制位 PWM1D[2:0]决定在这 8 个 PWM 周期中的部分周期进行延展, 这些延展周期内的 T1PWM 高电平将延展半个计数时钟(即为 T1DATA+0.5)。

8+3 模式相当于在整体上提高了 PWM 占空比的调节精度, 而不是对单个 PWM 周期提高占空比精度。PWM 单个周期中正常的高电平时间由 T1DATA 决定, 而 PWM1D[2:0]决定每 8 个周期中哪几个周期的高电平需要延展半个计数时钟。

PWM1 的 8+3 模式一个大周期的占空比计算如下(包含 PWM1DB 功能, $0 < T1DATA < T1LOAD$):

- ✧ 高电平时间 = $[(T1DATA + (PWM1DB - 1)/2) \times 8 + (PWM1D[2:0])/2] \times T1CNT$ 计数时钟周期
- ✧ 周期 = $(T1LOAD + 1) \times 8 \times T1CNT$ 计数时钟周期
- ✧ 占空比 = 高电平时间/周期 = $[T1DATA + (PWM1DB - 1)/2 + (PWM1D[2:0])/16] / (T1LOAD + 1)$

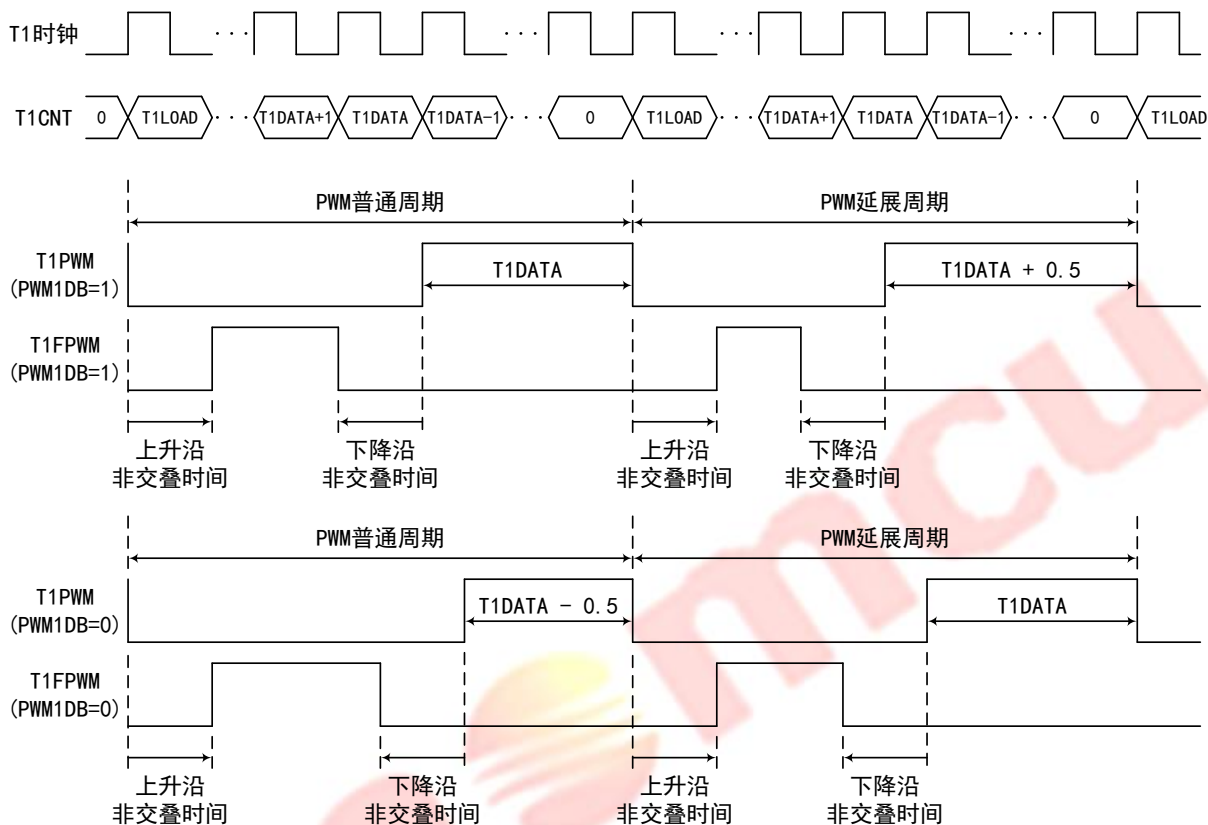
举例说明 (PWM1DB=1):

若 PWM1D[2:0]=000, 则对 8 个普通的 PWM 周期均不延展, 大周期中的占空比与普通 PWM 的占空比一样, 等于 $[T1DATA / (T1LOAD + 1)]$ 。

若 PWM1D[2:0]=001, 则对 8 个普通的 PWM 周期中的第 1 个周期延展, 大周期中的占空比为 $[(T1DATA + 1/16) / (T1LOAD + 1)]$ 。

若 PWM1D[2:0]=101, 则对 8 个普通的 PWM 周期的第 (1、2、3、5、7) 个周期延展, 大周期中的占空比为 $[(T1DATA + 5/16) / (T1LOAD + 1)]$ 。

PWM 的互补死区、PWM1DB 及 8+3 模式波形示意



注:

- 1、PWM 工作中不可修改 PWM 周期和死区;
- 2、应用互补 PWM 时, 非交叠时间的设置应匹配 PWM 周期及占空比的设置, 否则 PWM 无法正常工作;
- 3、PWM1 关闭但 FPWM1 使能时, 禁止写 T1DATA;
- 4、8+3 模式下, 当 PWM1DB=1 时, 普通周期 (不延展的周期) 的占空比 (高电平时间) 为 T1DATA, 而延展周期的占空比则延展半个时钟 (即为 T1DATA+0.5); 当 PWM1DB=0 时, 所有周期 (包括普通周期和延展周期) 的占空比均缩减半个时钟;
- 5、当 T1DATA 值设为 0 或与 T1LOAD 值相等时, 不支持 PWM1DB 的半周期微调功能;
- 6、若定时器的计数时钟为 FCPU 的 1 分频时, PWM1DB 微调功能仅滞后约 1/4 个时钟周期;

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	PWM1EN	FPWM1EN	T1CKS1	T1CKS0	T1PRS2	T1PRS1	T1PRS0
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] **T1EN** – 定时器 T1 使能位

0: 关闭定时器 T1;

1: 开启定时器 T1;

BIT[6] **PWM1EN** – PWM1 使能位

0: 关闭 PWM1 功能;

1: 使能 PWM1 功能;

BIT[5] **FPWM1EN** – FPWM1 使能位

0: 关闭 FPWM1 功能;

1: 使能 FPWM1 功能;

BIT[4:3] **T1CKS[1:0]** – T1 时钟源选择位

T1CKS[1:0]	T1 时钟源
00	FCPU
01	FHOSC
10	FPFRC
11	TC1 上升沿

BIT[2:0] **T1PRS[2:0]** – T1 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。

定时器 T1 比较寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1DATA	T1DATA7	T1DATA6	T1DATA5	T1DATA4	T1DATA3	T1DATA2	T1DATA1	T1DATA0
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] **T1DATA[7:0]** – T1 比较寄存器，用于设置 PWM1 的占空比

PWM1 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CR0	-	-	-	-	FPWM1NV	FPWM1OE	PWM1NV	PWM1OE
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3] **FPWM1NV** – FPWM1 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] **FPWM1OE** – FPWM1 端口输出使能位

- 0: 禁止端口输出 PWM 波形；
- 1: 允许端口输出 PWM 波形；

BIT[1] **PWM1NV** – PWM1 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] **PWM1OE** – PWM1 端口输出使能位

- 0: 禁止端口输出 PWM 波形；
- 1: 允许端口输出 PWM 波形；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CR1	-	-	FPWM1E5	FPWM1E4	FPWM1E3	FPWM1E2	FPWM1E1	FPWM1E0
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] **FPWM1E[5:0]** – T1FPWM 上升/下降沿非交叠时间选择位

FPWM1E[5:0]	上升沿非交叠时间	下降沿非交叠时间
00 0000	1 个 T1 时钟周期	1 个 T1 时钟周期
00 0001	2 个 T1 时钟周期	2 个 T1 时钟周期
...	...	...
11 1111	64 个 T1 时钟周期	64 个 T1 时钟周期

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CR2	-	-	-	-	PWM1DB	PWM1D2	PWM1D1	PWM1D0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	1	0	0	0

BIT[3] **PWM1DB** – T1PWM/T1FPWM 匹配翻转时钟沿选择位

0: T1PWM/T1FPWM 占空比匹配时, 输出电平滞后半时钟周期翻转;

1: T1PWM/T1FPWM 占空比匹配时, 输出电平与时钟触发沿同步翻转;

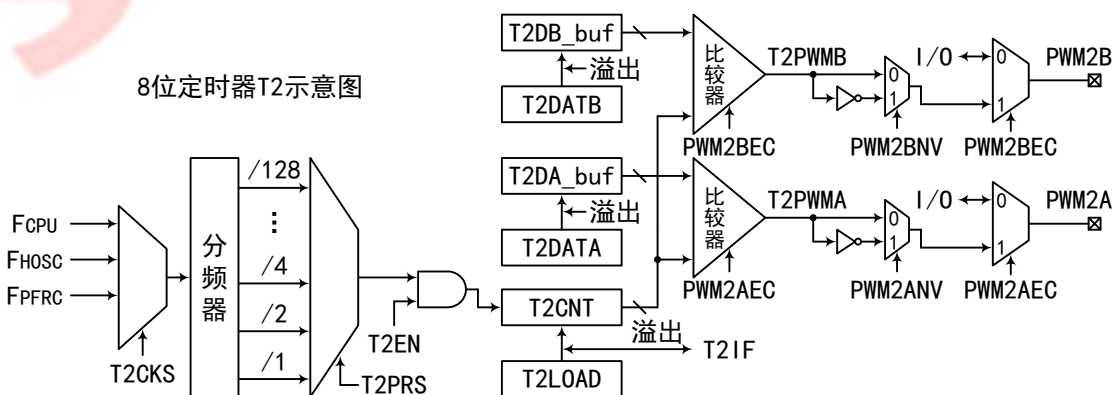
BIT[2:0] **PWM1D[2:0]** – T1PWM/T1FPWM 占空比延展控制位

PWM1D[2:0]	T1PWM/T1FPWM 每 8 个周期中占空比延展半个时钟宽度的周期数
000	每 8 个周期中有 0 个周期的占空比进行延展
001	每 8 个周期中有 1 个周期 (第 1 个周期) 的占空比进行延展
010	每 8 个周期中有 2 个周期 (第 1,5 个周期) 的占空比进行延展
011	每 8 个周期中有 3 个周期 (第 1,3,5 个周期) 的占空比进行延展
100	每 8 个周期中有 4 个周期 (第 1,,3,5,7 个周期) 的占空比进行延展
101	每 8 个周期中有 5 个周期 (第 1,2,3,5,7 个周期) 的占空比进行延展
110	每 8 个周期中有 6 个周期 (第 1,2,3,5,6,7 个周期) 的占空比进行延展
111	每 8 个周期中有 7 个周期 (第 1,2,3,4,5,6,7 个周期) 的占空比进行延展

7.4 定时器 T2

定时器 T2 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 位重载寄存器及 2 个 8 位比较寄存器。

- ◇ 可通过预分频比设置计数频率, 可通过重载寄存器控制计数周期;
- ◇ 支持共周期 2 路 8 位 PWM 输出, 可通过对应的比较寄存器分别设置每路 PWM 占空比;
- ◇ 时钟源可选 F_{PFRC};
- ◇ 支持溢出中断和溢出唤醒功能;



定时器 T2, 可通过寄存器位 T2CKS 选择时钟源, 通过 T2PRS 选择预分频比, 所选时钟源通过预分频器后产生 T2 计数器 T2CNT 的计数时钟 (上升沿计数)。预分频比可选择 1~128 分频, 对 T2CNT 的

写操作将清零预分频计数器，而预分频比保持不变。

当 T2EN=0 时，T2CNT 保持不变，写重载寄存器 T2LOAD 将立即载入 T2CNT 中；当 T2EN=1 时，T2CNT 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 T2IF 将被置 1，同时 T2 自动将当前 T2LOAD 值载入 T2CNT 中重新开始计数。

如图所示，定时器 T2 可实现共周期的 2 路 PWM 功能（PWM2x，x=A,B，下同），且可分别设置每路 PWM 的占空比。可通过寄存器位使能/关闭 PWM 功能，或控制端口是否输出 PWM 波形。PWM2x 关闭时 T2PWMx 信号为低电平。PWM2x 使能后，T2CNT 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 T2DATx 相等时，T2PWMx 变为高电平；当计数溢出时，T2PWMx 变为低电平。

T2DATx 均配有 1 个 8 位的比较缓冲器（T2Dx_buf）用于与 T2CNT 比较，PWM2x 关闭时写 T2DATx 将立即载入比较缓冲器中，而 PWM2x 使能后写 T2DATx 则将在 T2 溢出时才载入比较缓冲器中。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。

T2PWMx 信号（x=A,B，下同）的占空比计算如下：

- ✧ 高电平时间 = (T2DATx) × T2CNT 计数时钟周期
- ✧ 周期（T2 的溢出周期）= (T2LOAD+1) × T2CNT 计数时钟周期
- ✧ 占空比 = 高电平时间 / 周期 = (T2DATx) / (T2LOAD+1)

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	-	-	T2CKS1	T2CKS0	T2PRS2	T2PRS1	T2PRS0
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] T2EN – 定时器 T2 使能位

0: 关闭定时器 T2;

1: 开启定时器 T2;

BIT[4:3] T2CKS[1:0] – T2 时钟源选择位

T2CKS[1:0]	T2 时钟源
00	FCPU
01	FHOSC
10	FPRFC
11	-

BIT[2:0] T2PRS[2:0] – T2 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 比较寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2DATA	T2DATA7	T2DATA6	T2DATA5	T2DATA4	T2DATA3	T2DATA2	T2DATA1	T2DATA0
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] **T2DATA[7:0]** – T2 比较寄存器 A，用于设置 PWM2A 的占空比

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2DATB	T2DATB7	T2DATB6	T2DATB5	T2DATB4	T2DATB3	T2DATB2	T2DATB1	T2DATB0
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] **T2DATB[7:0]** – T2 比较寄存器 B，用于设置 PWM2B 的占空比

PWM2 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2CR	PWM2AEC	PWM2BEC	PWM2ANV	PWM2BNV	-	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-	-
初始值	0	0	0	0	-	-	-	-

BIT[7] **PWM2AEC** – PWM2A 使能位及端口输出控制位

0: 关闭 PWM2A 功能，并禁止端口输出 PWM 波形；

1: 使能 PWM2A 功能，并允许端口输出 PWM 波形；

- BIT[6] **PWM2BEC** – PWM2B 使能位及端口输出控制位
0: 关闭 PWM2B 功能，并禁止端口输出 PWM 波形；
1: 使能 PWM2B 功能，并允许端口输出 PWM 波形；
- BIT[5] **PWM2ANV** – PWM2A 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[4] **PWM2BNV** – PWM2B 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；

8 频率可编程 RC 振荡器 PFRC

芯片内置 1 个频率可编程的高频 RC 振荡器（PFRC），输出时钟 F_{PFRC} 可用作定时器时钟源。

PFRC 振荡频率可微调校准，校准值保存在 8 位微调寄存器 PFRCCAL 中，芯片复位后，寄存器自动加载出厂默认值作为初始值，将 F_{PFRC} 调整至 32MHz。该寄存器允许通过软件进行微调，可生成一定范围内的其他频率。

PFRC 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PFRCCR	PFRCEN	-	-	-	-	-	-	-
R/W	R/W	-	-	-	-	-	-	-
初始值	0	-	-	-	-	-	-	-

BIT[7] PFRCEN – PFRC 使能位

0: 关闭 PFRC;

1: 开启 PFRC;

PFRC 微调寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PFRCCAL	PFRCCAL7	PFRCCAL6	PFRCCAL5	PFRCCAL4	PFRCCAL3	PFRCCAL2	PFRCCAL1	PFRCCAL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	U	U	U	U	U	U	U	U

BIT[7:0] PFRCCAL[7:0] – PFRC 频率微调校准位（复位初始值为出厂设定值）

注:

- 1、PFRC 开启后需延时以待其振荡稳定（时间>200us），才可正常应用；
- 2、PFRC 最大调节范围为 (32MHz - 10%) ~ (32MHz + 4%)（以实际芯片为准）；
- 3、步长增减的频率是非线性的，最大约为 0.5%；
- 4、PFRCCAL 中已存有出厂校准值，软件微调前应备份原值以便恢复；

9 模数转换器 ADC

9.1 ADC 概述

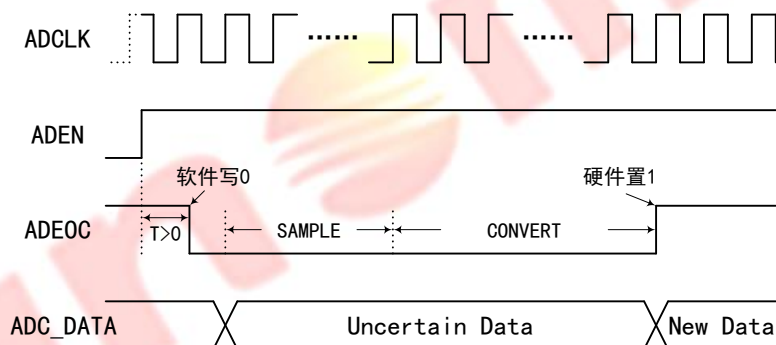
芯片内置 1 个 12 位高精度逐次逼近型的模数转换器 ADC。

- ◇ 12 路外部通道：AN0~AN11；4 路内部通道：GND、VDD/4、EVN0/4、EVN1/4；
- ◇ 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：F_{HIRC} 的 32/64/128/256 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 ADC 的转换状态。当 ADEOC 为 1 时对其写 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 2/4/8/15 个 ADCLK (即 ADC 时钟周期)，转换 (CONVERT) 时间固定为 12 个 ADCLK，一次 ADC 转换为 14/16/20/27 个 ADCLK。

ADC 转换时序如下图所示：



注：

- 1、AD 转换过程中或者 ADEN 未使能时，ADRH/ADRL 中的数据未知，应在 AD 转换结束且 ADEN 使能的情况下读取 AD 转换数据；
- 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} 将随之下降；
- 3、使能 ADC 模块后需延迟 500ms 以上、切换参考电压需延迟 200us 以上，待电路稳定后才能启动 AD 转换；切换输入通道后，受外部输入影响，前两次转换的结果会有误差，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；

9.2 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
ADCR0	ADEN	ADEOC	ADCKS1	ADCKS0	ADCHS3	ADCHS2	ADCHS1	ADCHS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	1	1	1	1	1	1	1

BIT[7] **ADEN** – ADC 使能位

0: 关闭 ADC;

1: 开启 ADC;

BIT[6] **ADEOC** – ADC 启动控制位及转换状态标志位

0: AD 转换中, 转换结束后自动置 1;

1: AD 转换结束, 写 0 启动 AD 转换;

BIT[5:4] **ADCKS[1:0]** – ADC 转换时钟选择位

ADCKS[1:0]	ADC 转换时钟 F_{ADC}
00	F _{HIRC} /32
01	F _{HIRC} /64
10	F _{HIRC} /128
11	F _{HIRC} /256

BIT[3:0] **ADCHS[3:0]** – ADC 输入通道选择位

ADCHS[3:0]	ADC 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	EVN0/4
1101	EVN1/4
1110	VDD/4
1111	GND

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCR1	ADRSEL	ADVRS2	ADVRS1	ADVRS0	-	-	ADSPS1	ADSPS0
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] **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] **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 VER1 的输入电压);

BIT[5:4] **ADVRS[1:0]** – ADC 参考电压选择位

ADVRS[1:0]	ADC 参考电压
00	VDD
01	内部 4.0V
10	内部 3.0V
11	内部 2.0V

BIT[1:0] **ADSPS[1:0]** – ADC 采样时间选择位

ADSPS[1:0]	ADC 采样时间
00	15 个 ADCLK
01	8 个 ADCLK
10	4 个 ADCLK
11	2 个 ADCLK

ADC 转换结果寄存器

ADRSEL=0 时:

	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]** – 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]** – ADC 转换结果低 4 位

ADRSEL=1 时:

	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]** – 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]** – 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	OSADJT3	OSADJT2	OSADJT1	OSADJT0
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] **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[4] **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[3:0] **OSADJT[3:0]** – ADC 零点偏移修调电压选择位

OSADJT[3:0]	修调电压 (典型值)
0000	0mV
0001	1mV
0010	2mV
...	...
1110	14mV
1111	15mV

9.3 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 ADC 转换时钟；
- (4) 若采样时间可选，则设置 ADSPS，选择适当的 ADC 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；

9.4 ADC 零点偏移修调流程

- (1) 设置 ADC 输入通道为内部接地，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 ADC 结果为 0，则执行（4）；
 - ✧ 若 ADC 结果不为 0，则执行（3）；
- (3) OSADJT 加 1 后进行 ADC 转换：
 - ✧ 若 ADC 结果为 0，则跳至（6）；
 - ✧ 若 ADC 结果不为 0，则循环执行（3），直到结果为 0 或 OSADJT=0FH，跳至（6）；
- (4) 设置 OSADJTD=1、OSADJT=0FH，进行 ADC 转换：
 - ✧ 若 ADC 结果为 0，则跳至（6）；
 - ✧ 若 ADC 结果不为 0，则执行（5）；
- (5) OSADJT 减 1 后进行 ADC 转换：
 - ✧ 若 ADC 结果为 0，则跳至（6）；
 - ✧ 若 ADC 结果不为 0，则循环执行（5），直到结果为 0 或 OSADJT=00H，跳至（6）；
- (6) OSADJTD 及 OSADJT[3:0]的值即为零点偏移最佳修调结果，修调流程结束，后续 ADC 工作时直接使用，不需要再次修调。

10 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 选择电压检测量值。当 VDD 电压低于电压检测量值时检测状态标志位 LVDF 将被置 1，同时中断标志 LVDIF 置 1 触发 LVD 中断；因 LVD 电路有回滞特性（回滞电压典型值为 6%），VDD 电压需上升至电压检测量值+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	-	LVDVS2	LVDVS1	LVDVS0	-	-	LVDF
R/W	R/W	-	R/W	R/W	R/W	-	-	R
初始值	0	-	0	0	0	-	-	X

BIT[7] **LVDEN** – 低电压检测 LVD 使能位

0: 关闭 LVD;

1: 开启 LVD;

BIT[5:3] **LVDVS[2:0]** – LVD 电压检测量值选择位

LVDVS[2:0]	LVD 电压检测量值
000	2.0V
001	2.2V
010	2.4V
011	2.6V
100	2.8V
101	3.2V
110	3.6V
111	4.0V

BIT[0] **LVDF** – LVD 检测状态标志位

0: VDD 电压高于电压检测量值，或 LVD 关闭;

1: VDD 电压低于电压检测量值;

注：开启 LVD、切换电压检测量值等操作，需待电路稳定（时间>2ms）后 LVD 输出才有效。

11 中断

芯片的中断源包括外部中断（INT0~INT1）、定时器中断（T0~T2）、ADC 中断和键盘中断等。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，自动将当前指令的下一条要执行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，而是暂时挂起中断直到当前指令执行完成。
- ✧ CPU 响应中断后，程序跳到中断入口地址（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，然后执行 RETIE 返回主程序。此时芯片将自动恢复 GIE 为 1，然后从堆栈取出 PC 值，从中断产生时当前指令的下一条指令继续执行。

注：要使用外部中断功能或键盘中断功能，需将相应端口设为输入状态。

11.1 外部中断

芯片有 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，若中断总使能位 GIE 为 1 且外部中断使能位 INTnIE（n=0-1）为 1，则产生外部中断。

11.2 定时器中断

定时器 Tn（n=0-2）在计数溢出时触发定时器中断，中断标志 TnIF（n=0-2）将被置 1，若中断总使能位 GIE 为 1 且定时器中断使能位 TnIE（n=0-2）为 1，则产生定时器中断。

11.3 键盘中断

芯片有 8 路键盘中断源，均可通过寄存器位单独使能或屏蔽，任意一路使能的中断源的输入电平发生变化时，均会触发键盘中断，中断标志 KBIF 将被置 1，若中断总使能位 GIE 为 1 且键盘中断使能位 KBIE 为 1，则产生键盘中断。

键盘中断控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0KBCR	P07KE	P06KE	P05KE	P04KE	P03KE	P02KE	P01KE	P00KE
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] **P0nKE** – P0n 端口键盘中断使能位 (n=7-0)

0: 屏蔽端口键盘中断功能;

1: 使能端口键盘中断功能;

11.4 ADC 中断

ADC 转换完成后触发 ADC 中断, 中断标志 ADIF 将被置 1, 若中断总使能位 GIE 为 1 且 ADC 中断使能位 ADIE 为 1, 则产生 ADC 中断。

11.5 LVD 中断

当 VDD 电压从高于电压检测量值降为低于量值时触发 LVD 中断, 中断标志 LVDIF 将被置 1, 若中断总使能位 GIE 为 1 且 LVD 中断使能位 LVDIE 为 1, 则产生 LVD 中断。

11.6 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
INTE	T2IE	LVDIE	ADIE	KBIE	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] **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;

BIT[6] **LVDIE** – LVD 中断使能位

0: 屏蔽 LVD 中断;

1: 使能 LVD 中断;

BIT[5] **ADIE** – ADC 中断使能位

- 0: 屏蔽 ADC 中断;
1: 使能 ADC 中断;

BIT[4] **KBIE** – 键盘中断使能位

- 0: 屏蔽键盘中断;
1: 使能键盘中断;

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;

中断标志寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTF	T2IF	LVDIF	ADIF	KBIF	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] **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;

BIT[6] **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[5] **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[4] **KBIF** – 键盘中断标志位

- 0: 未触发键盘中断;
1: 已触发键盘中断, 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;

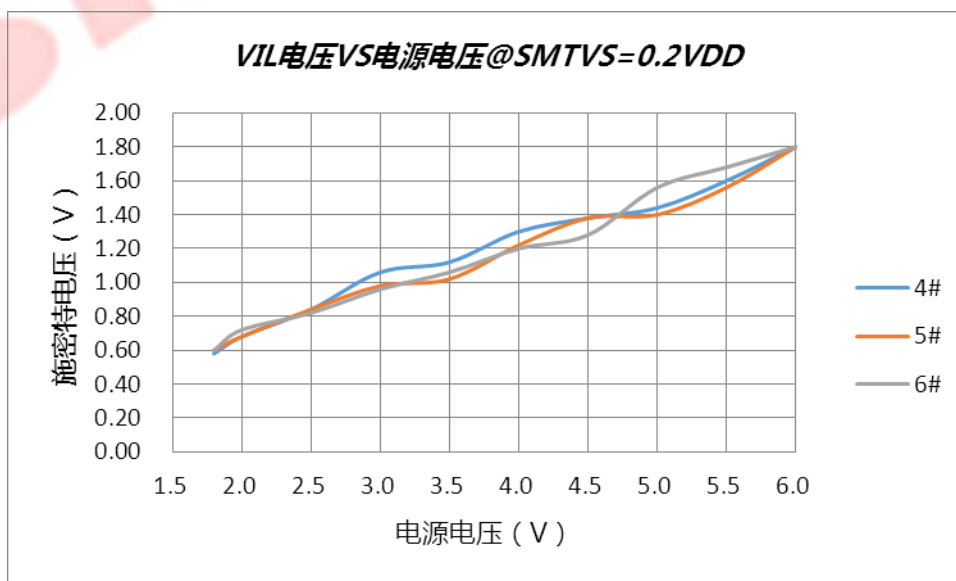
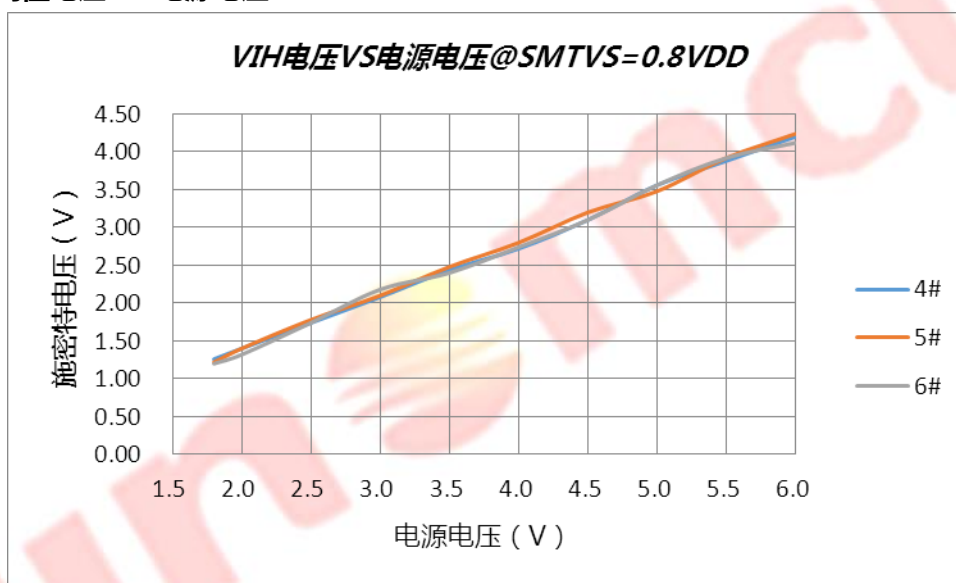
12 特性曲线

注:

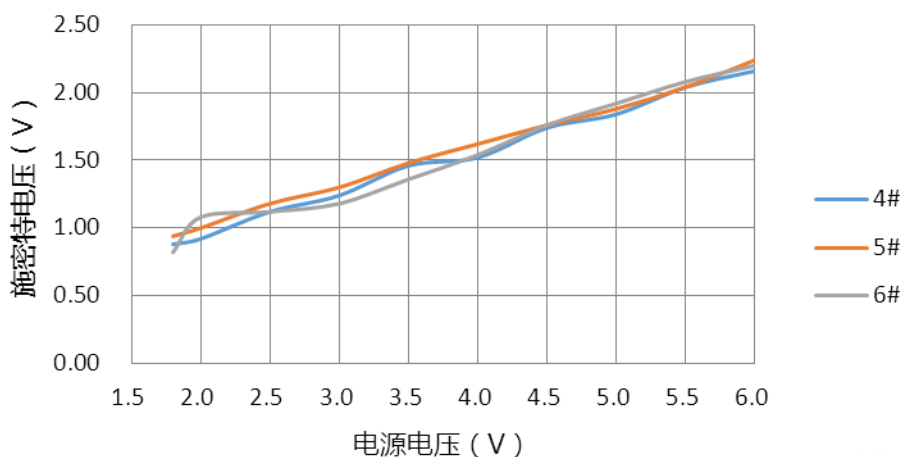
- 1、特性曲线图中数据均来自抽样实测，仅作为应用参考，部分数据因生产工艺偏差，可能与实际芯片不符；为保证芯片能正常工作，请确保其工作条件符合电气特性参数说明；
- 2、若图文中无特别说明，则电压特性曲线的温度条件为 $T=25^{\circ}\text{C}$ ，温度特性曲线的电压条件为 $VDD=5\text{V}$ ；

12.1 I/O 特性

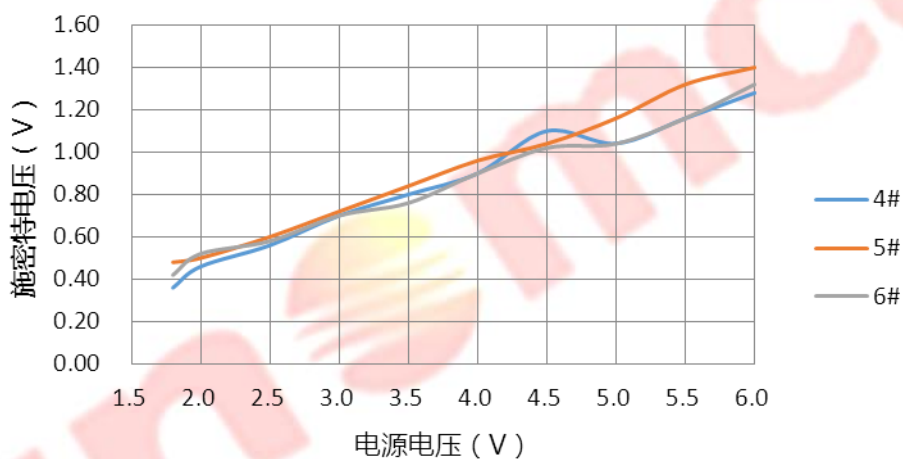
输入 SMT 阈值电压 VS 电源电压



VIH电压VS电源电压@输入高@SMTVS=2.0V

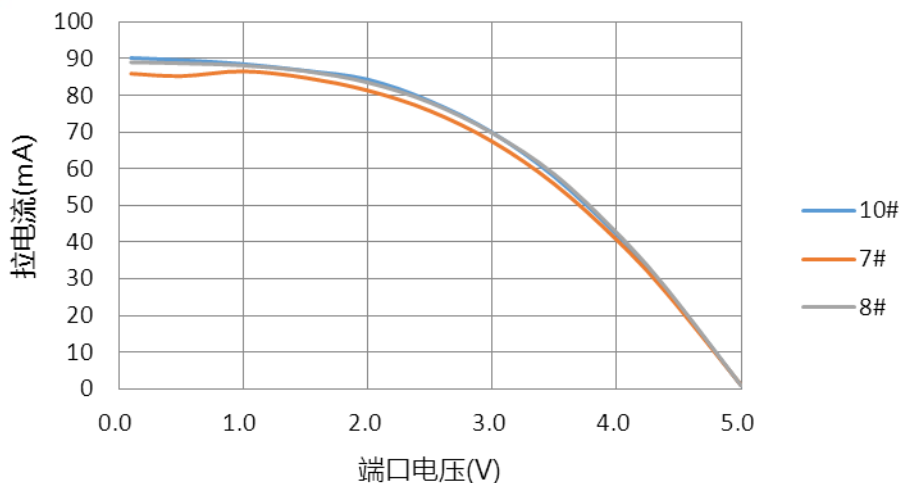


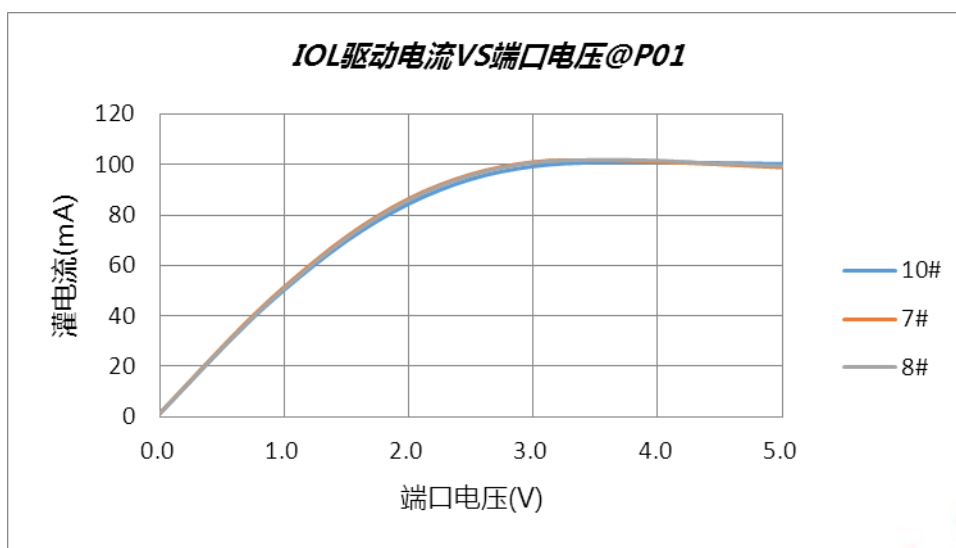
VIL电压VS电源电压@输入低@SMTVS=0.8V



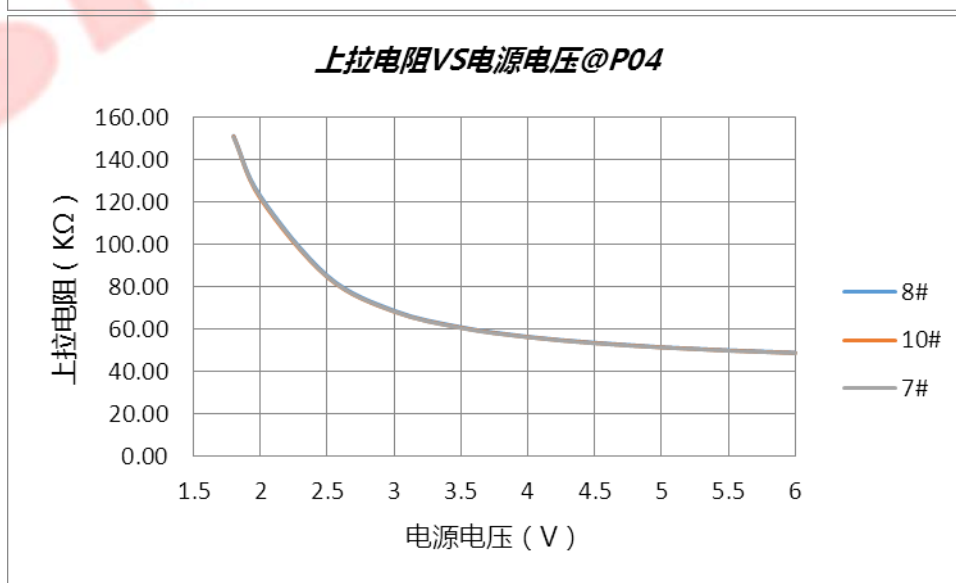
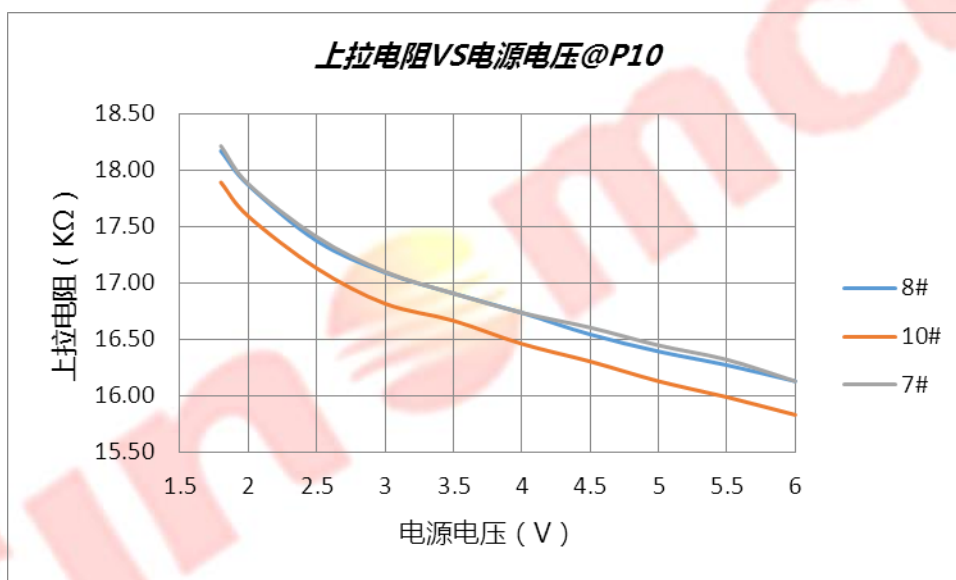
I/O 输出 驱动电流 VS 端口电压 (VDD=5V)

IOH驱动电流VS端口电压@P01

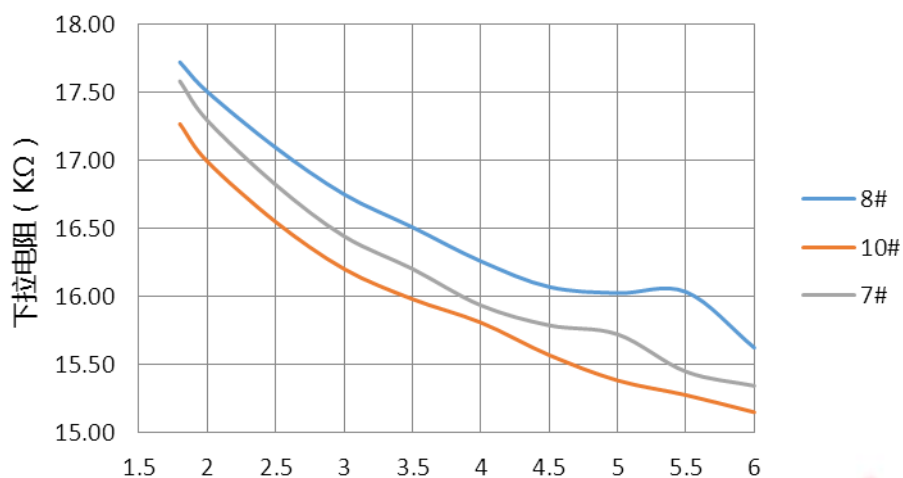




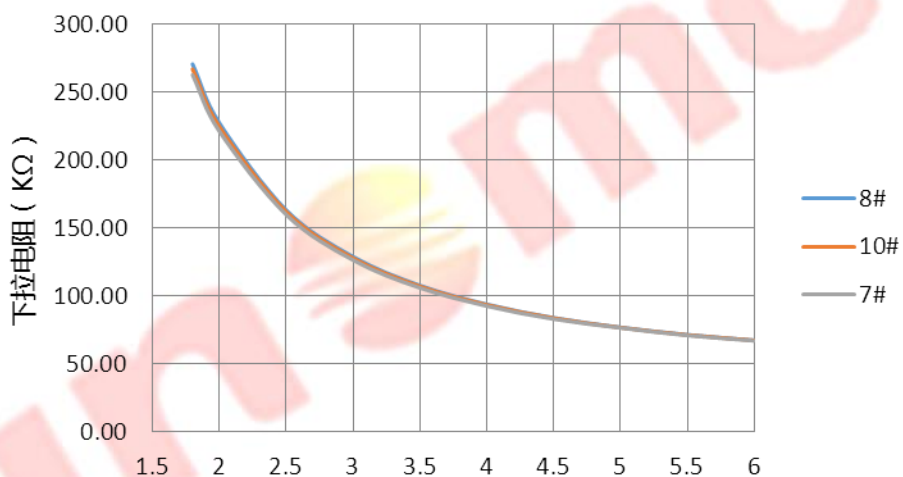
上/下拉电阻值 VS 电源电压



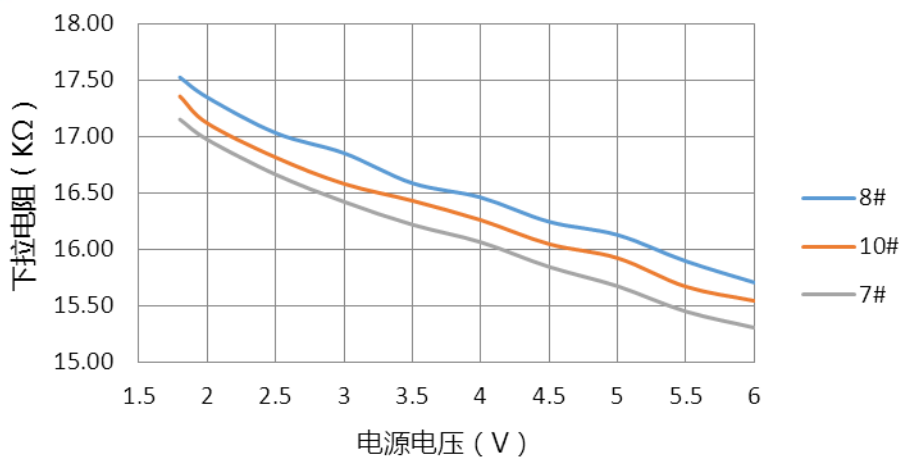
下拉电阻VS电源电压@P10

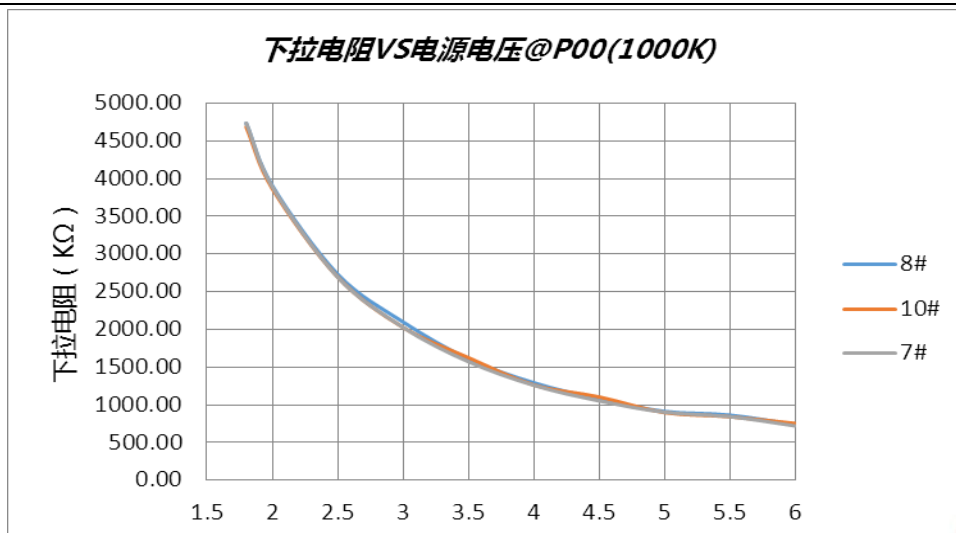


下拉电阻VS电源电压@P04



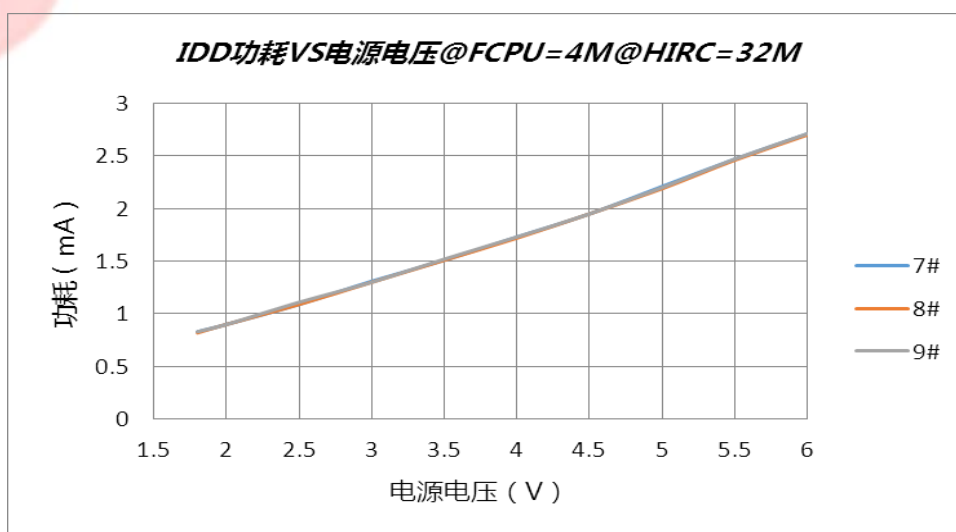
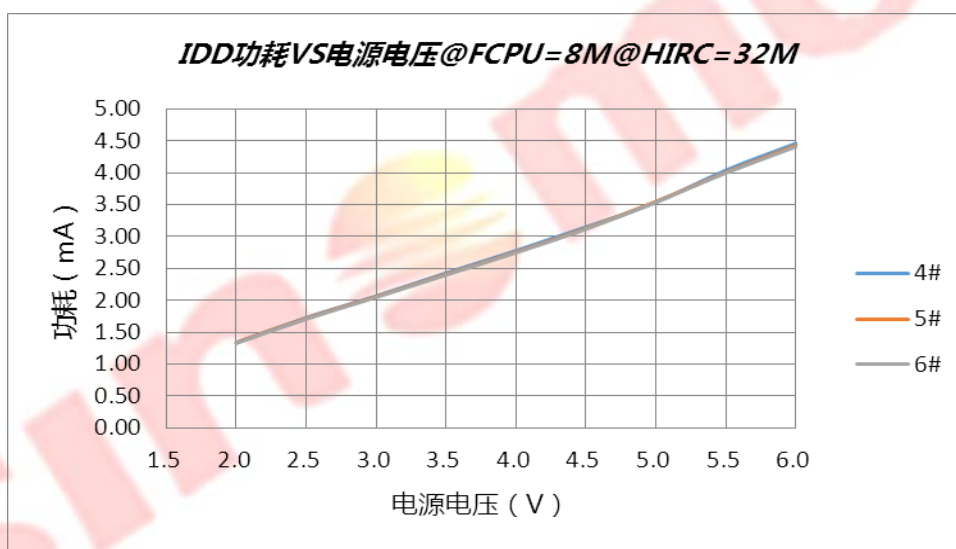
下拉电阻VS电源电压@P00(20K)

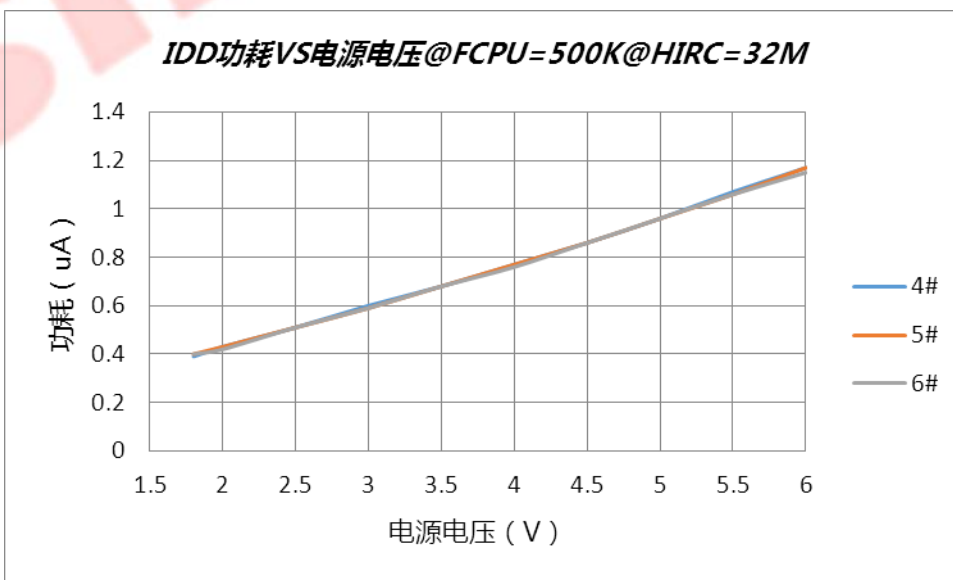
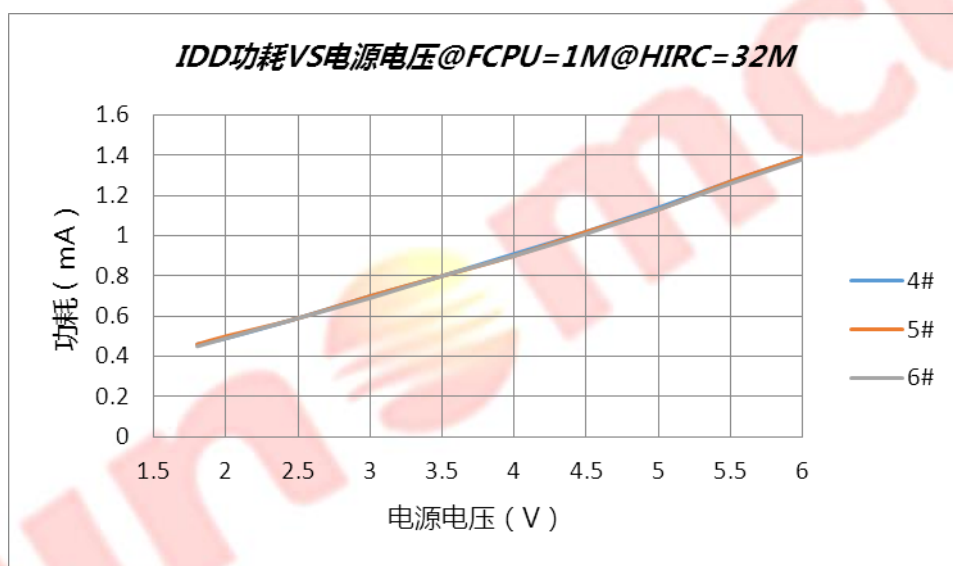
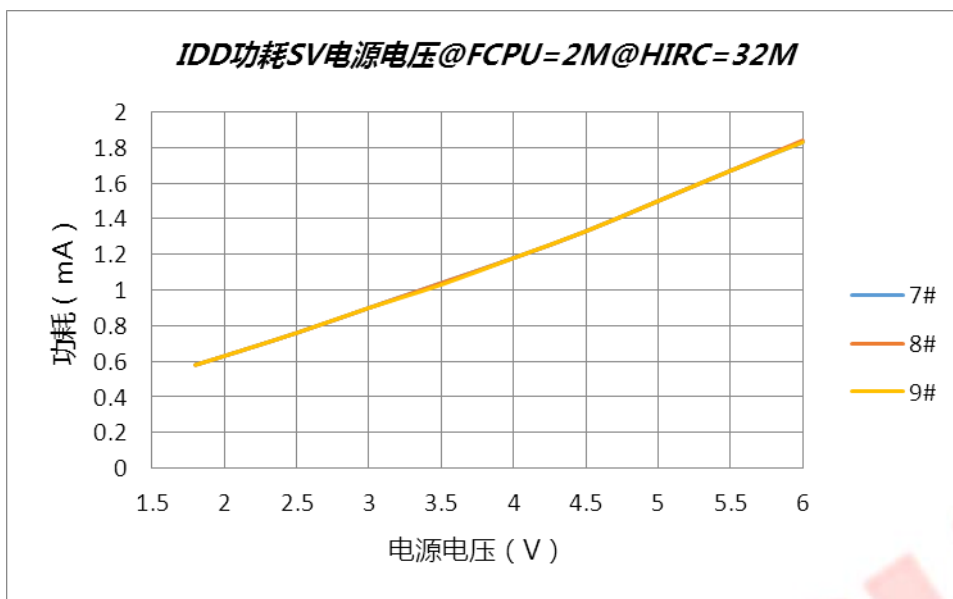




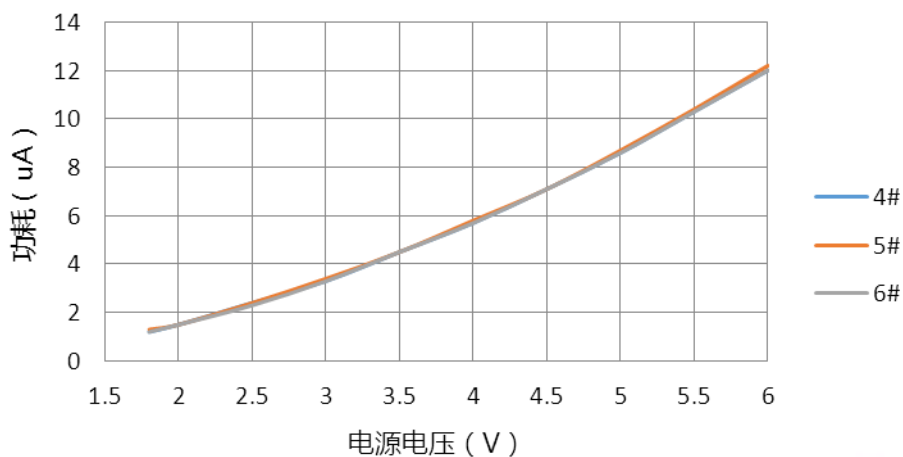
12.2 功耗特性

运行模式 功耗 VS 电源电压



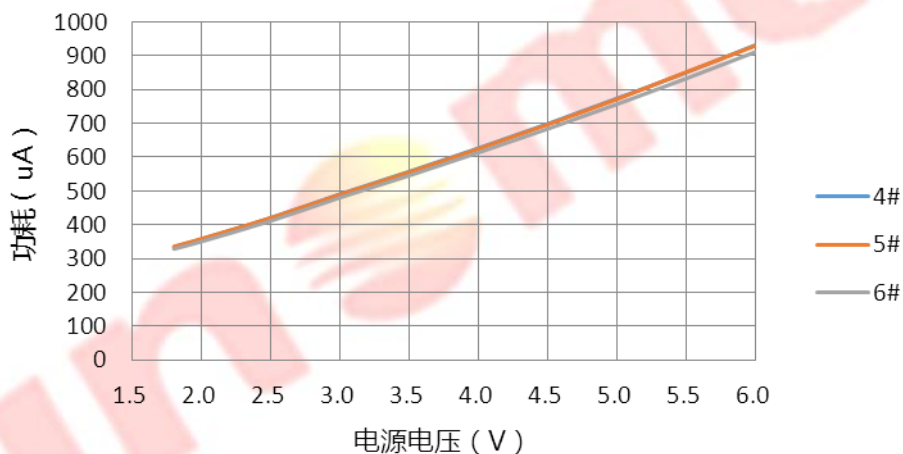


IDD功耗VS电源电压@FCPU=32KHz/2@LIRC=32K

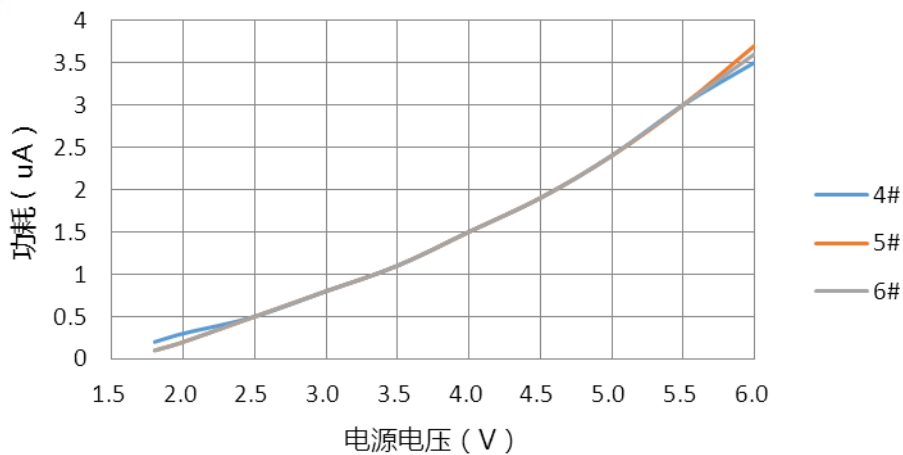


HOLD 模式 功耗 VS 电源电压

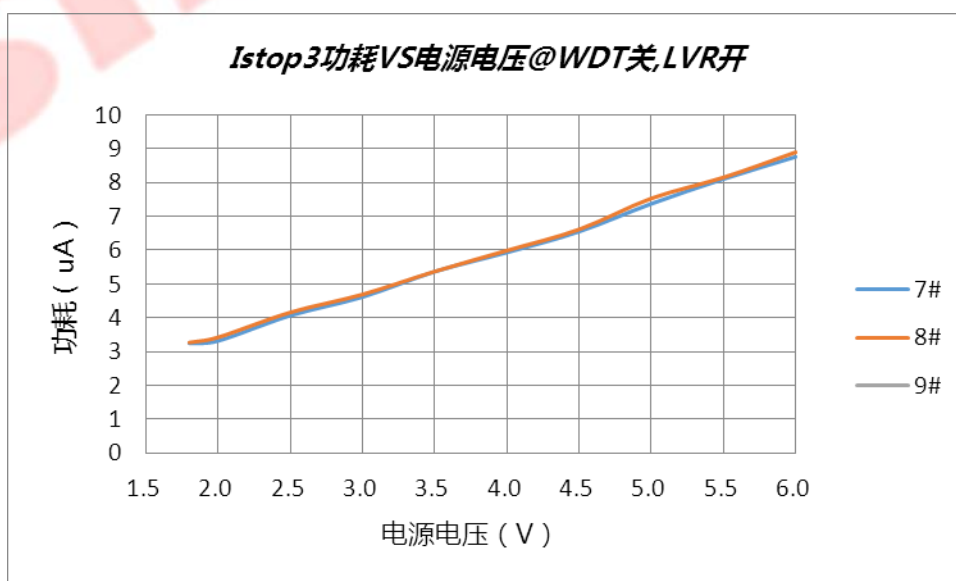
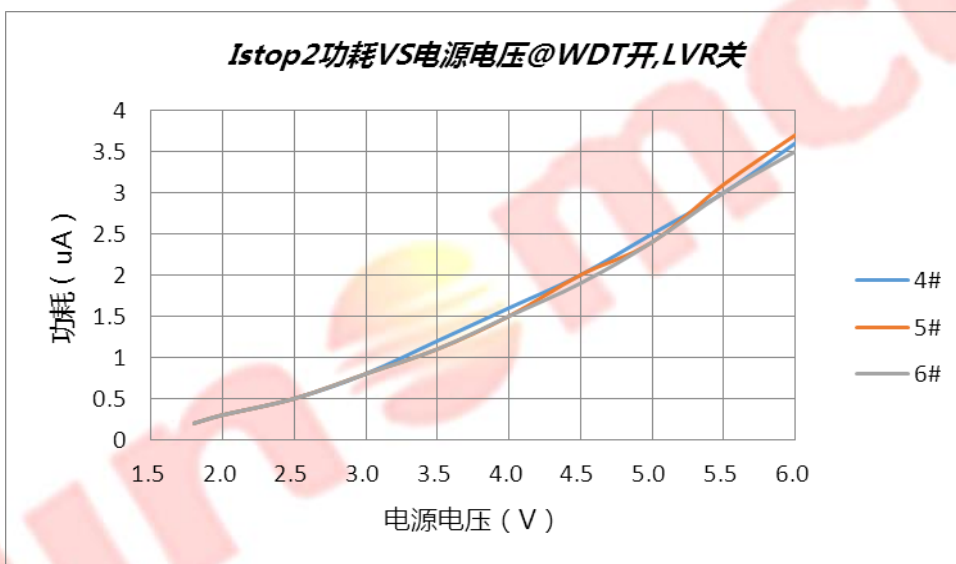
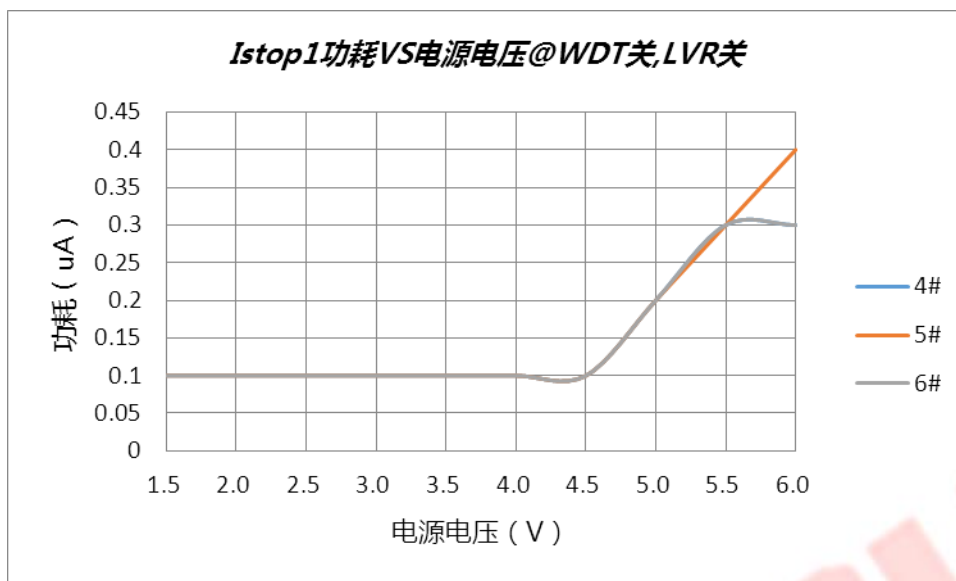
Ihold1功耗VS电源电压



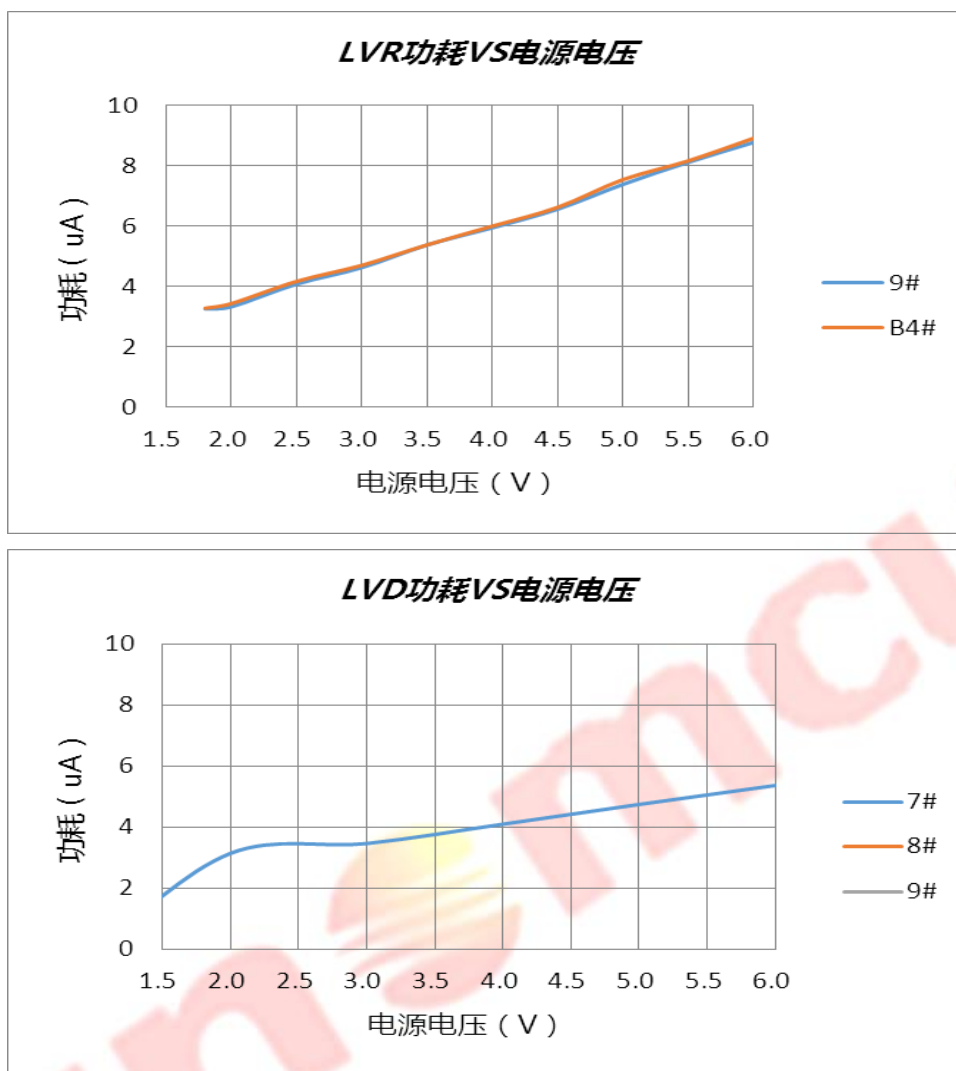
Ihold2功耗VS电源电压



休眠模式 功耗 VS 电源电压

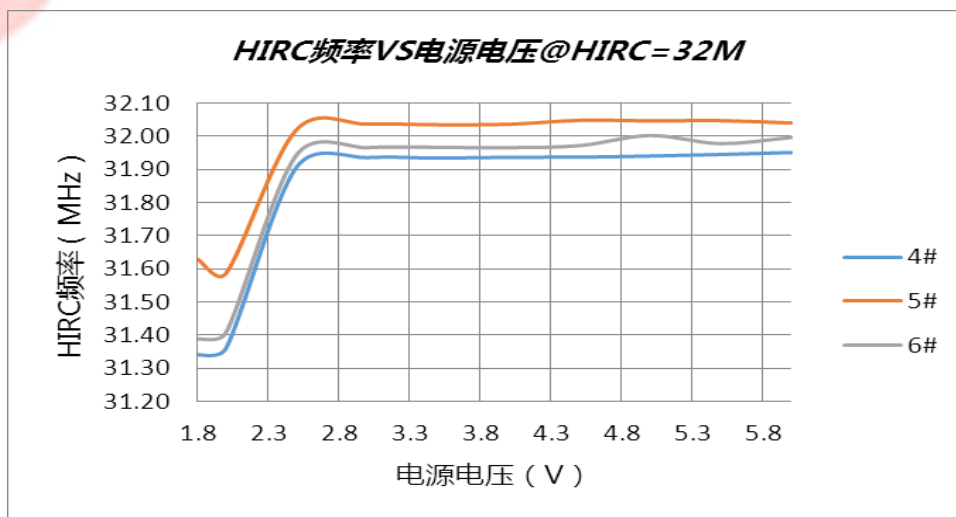


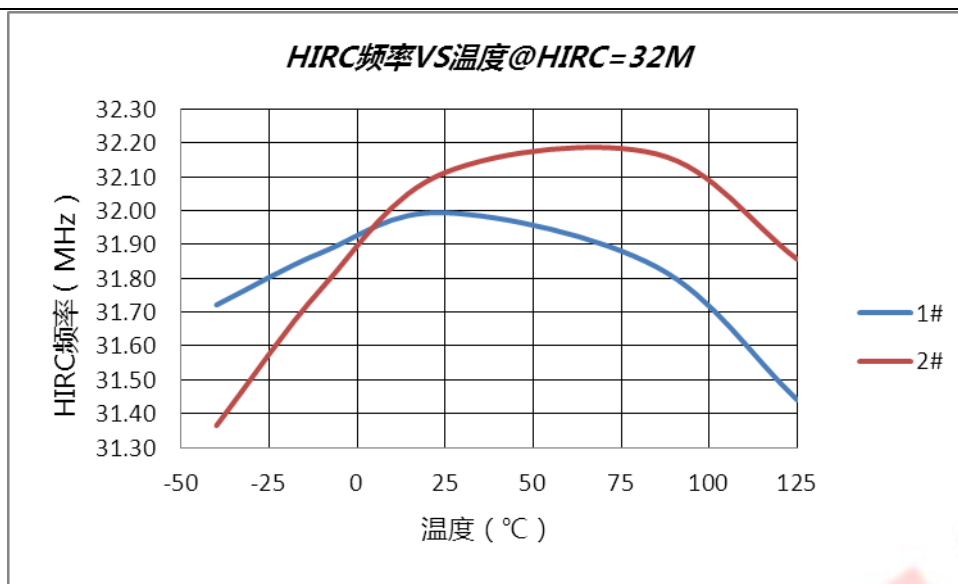
LVR/LVD 功耗 VS 电源电压



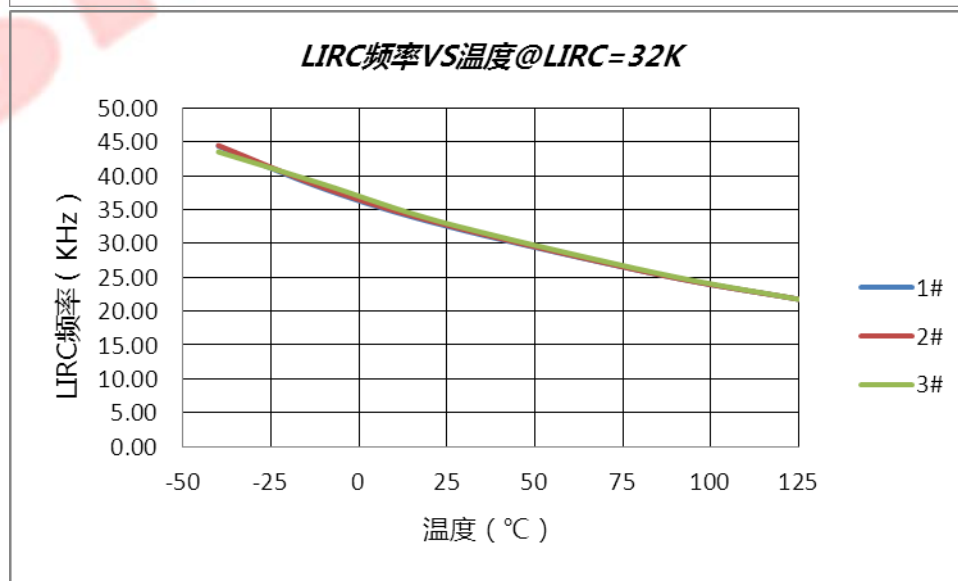
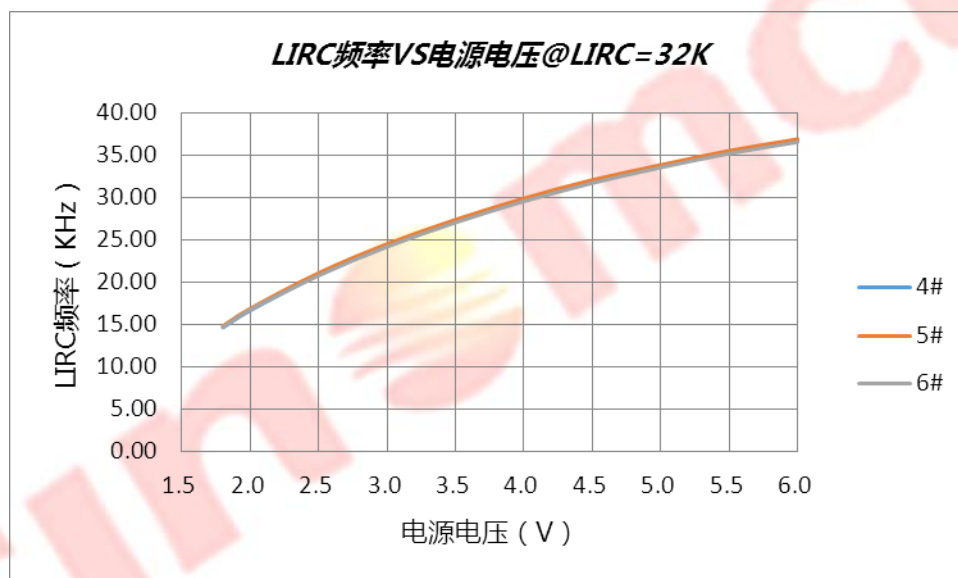
12.3 模拟电路特性

HIRC 频率 VS 电源电压/温度

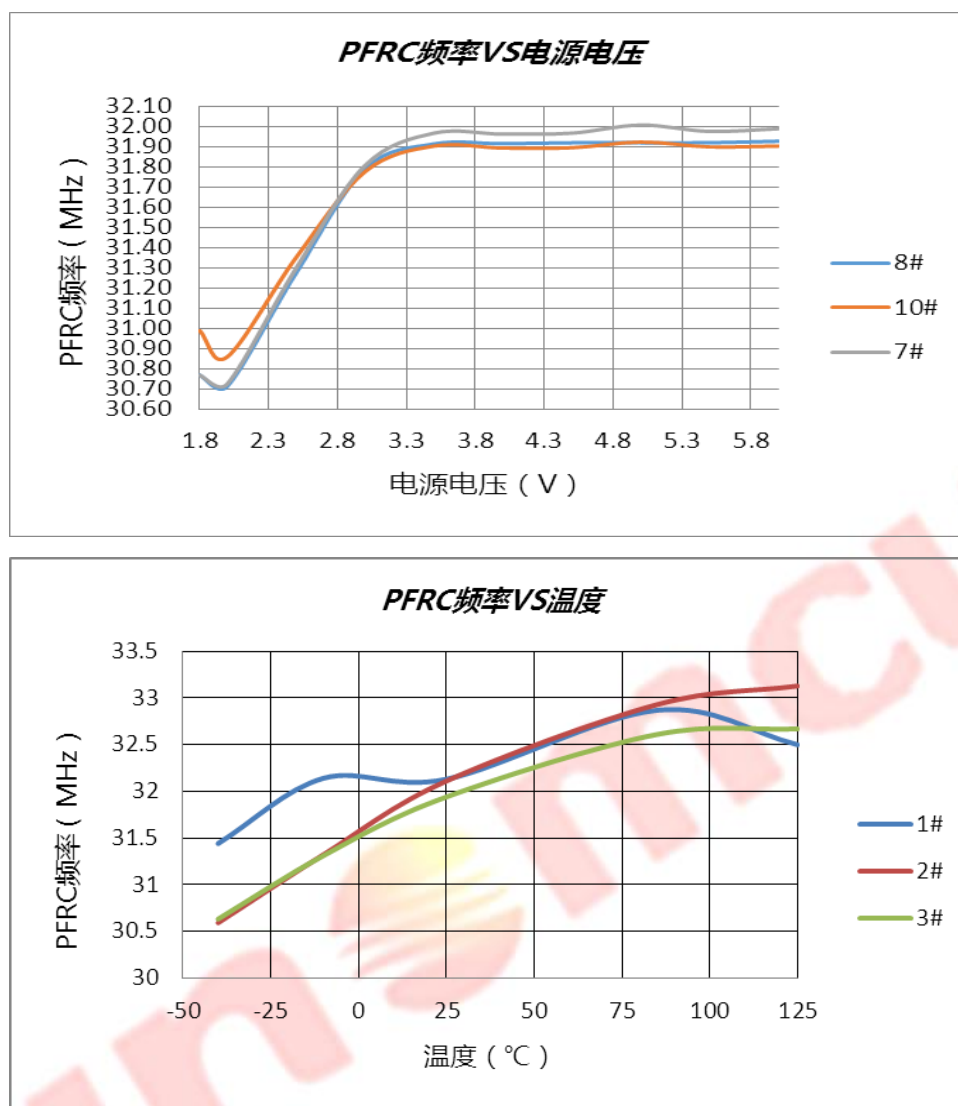




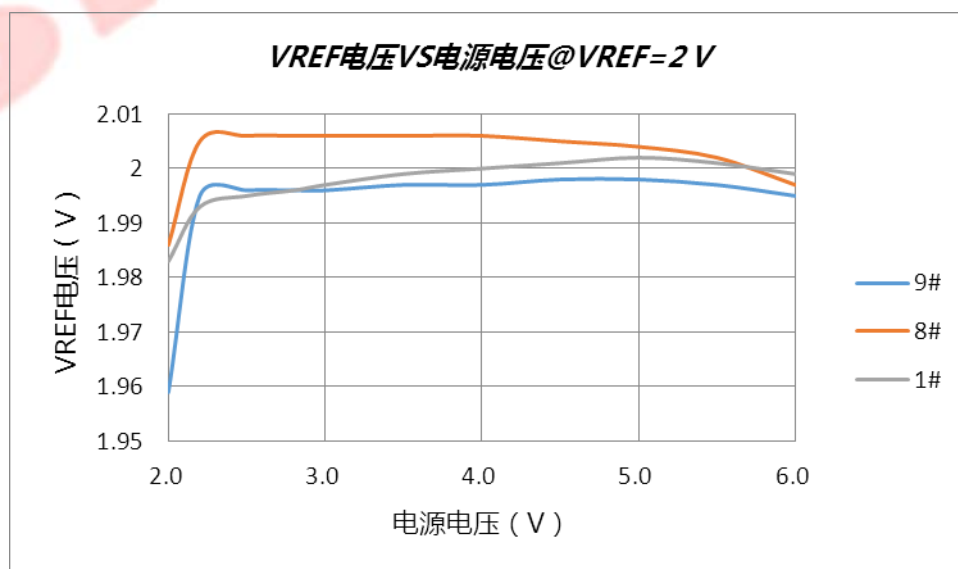
LIRC 频率 VS 电源电压/温度



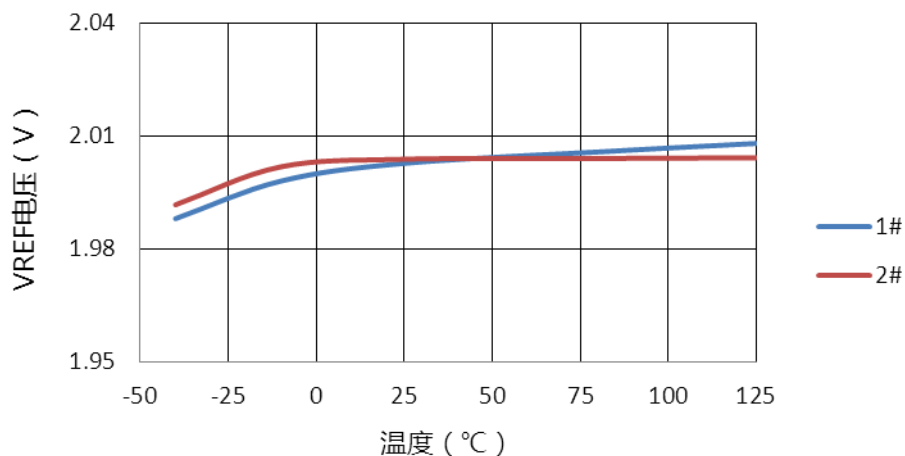
PFRC 频率 VS 电源电压/温度



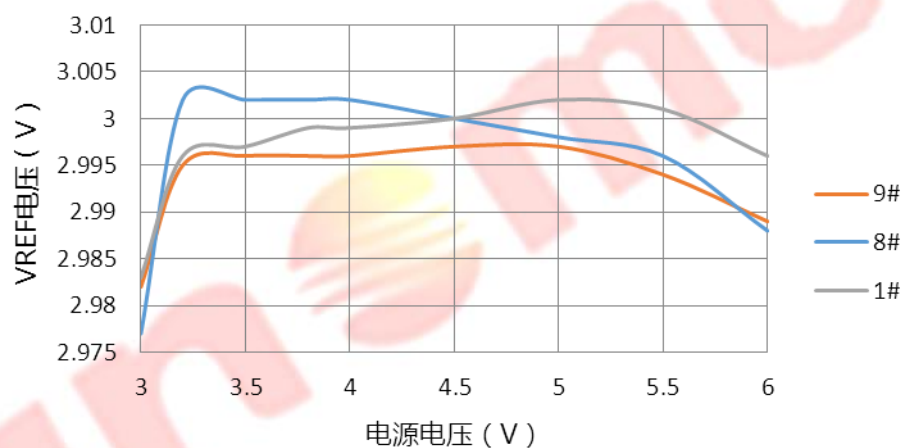
VIR 电压 VS 电源电压/温度



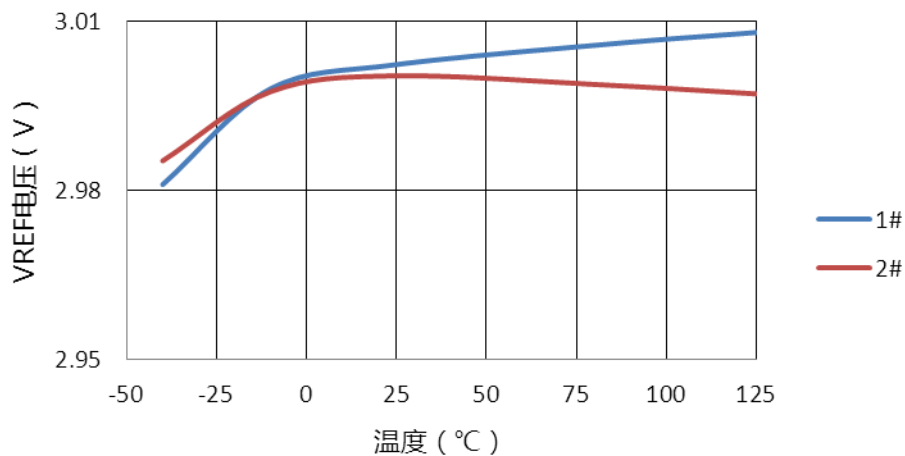
VREF电压VS温度@VREF=2V

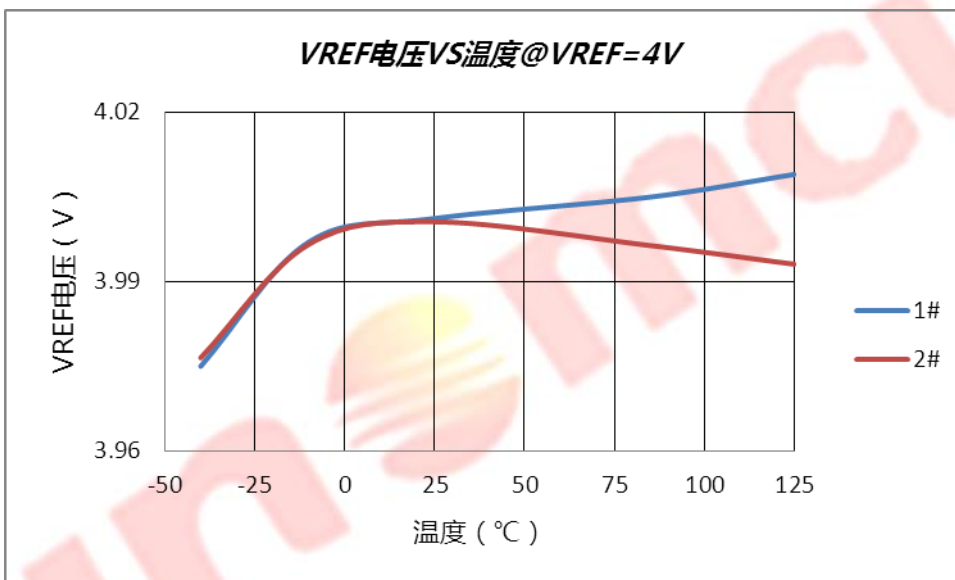
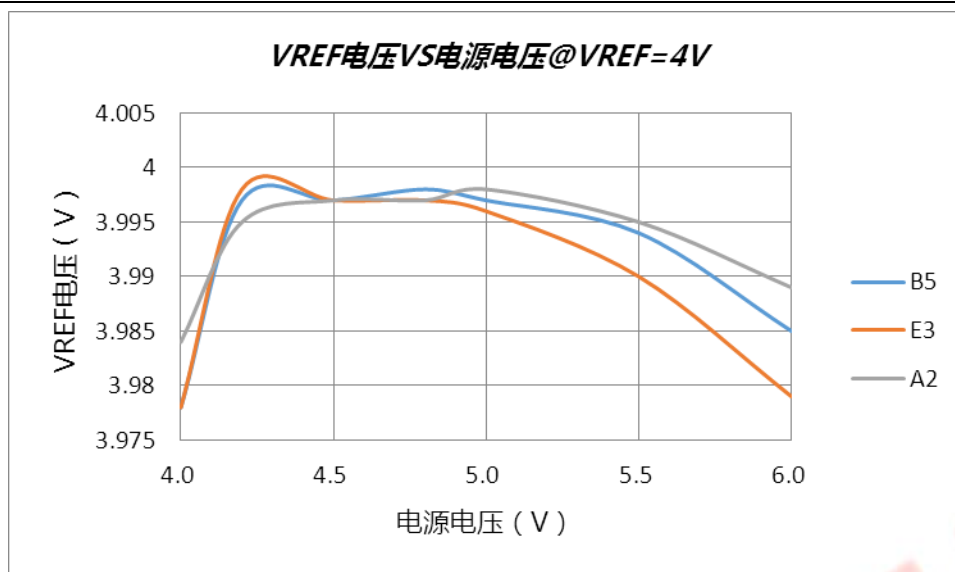


VREF电压VS电源电压@VREF=3V

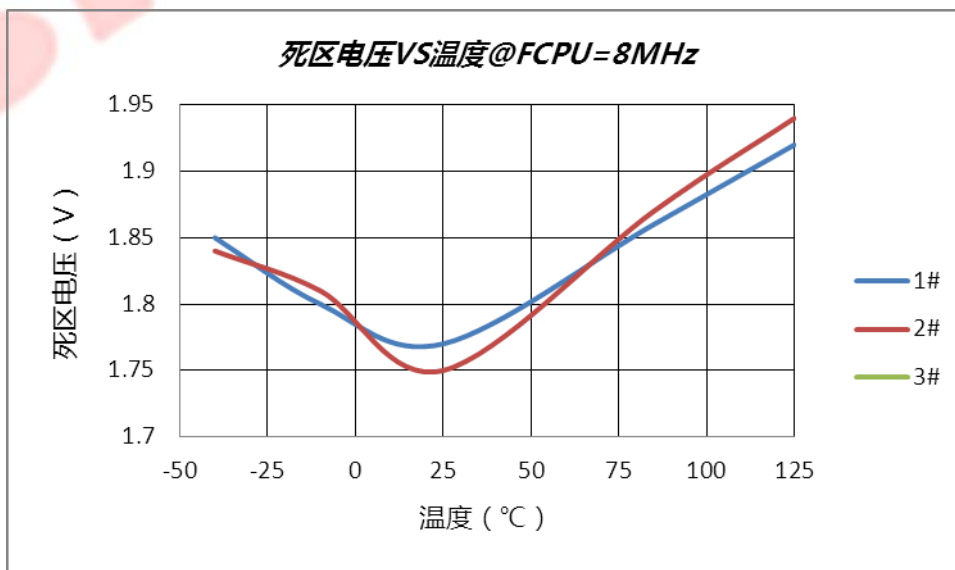


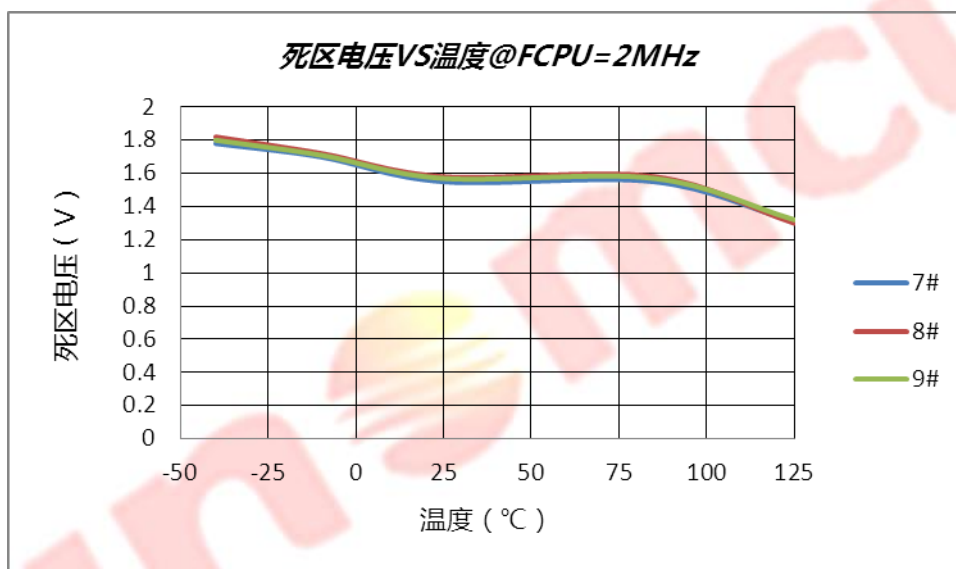
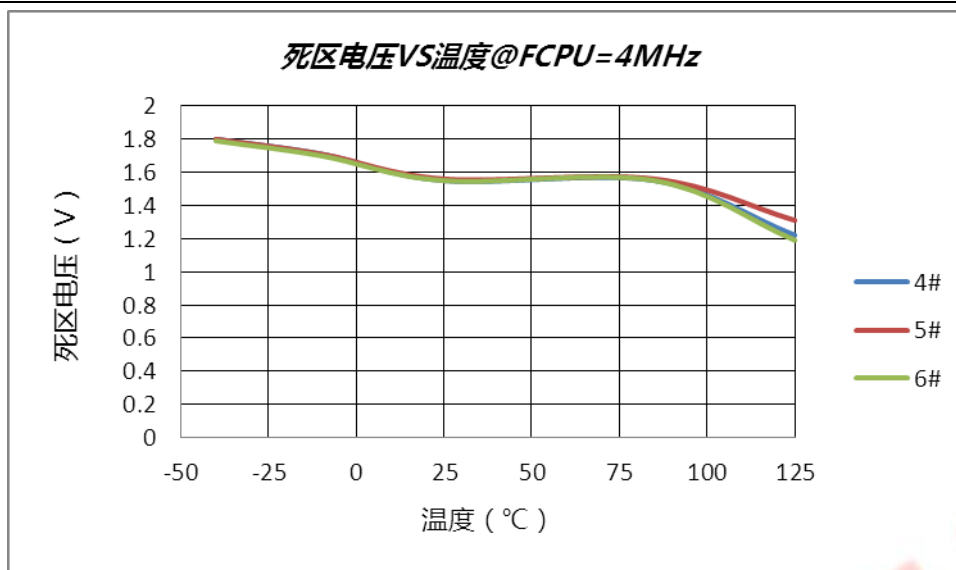
VREF电压VS温度@VREF=3V





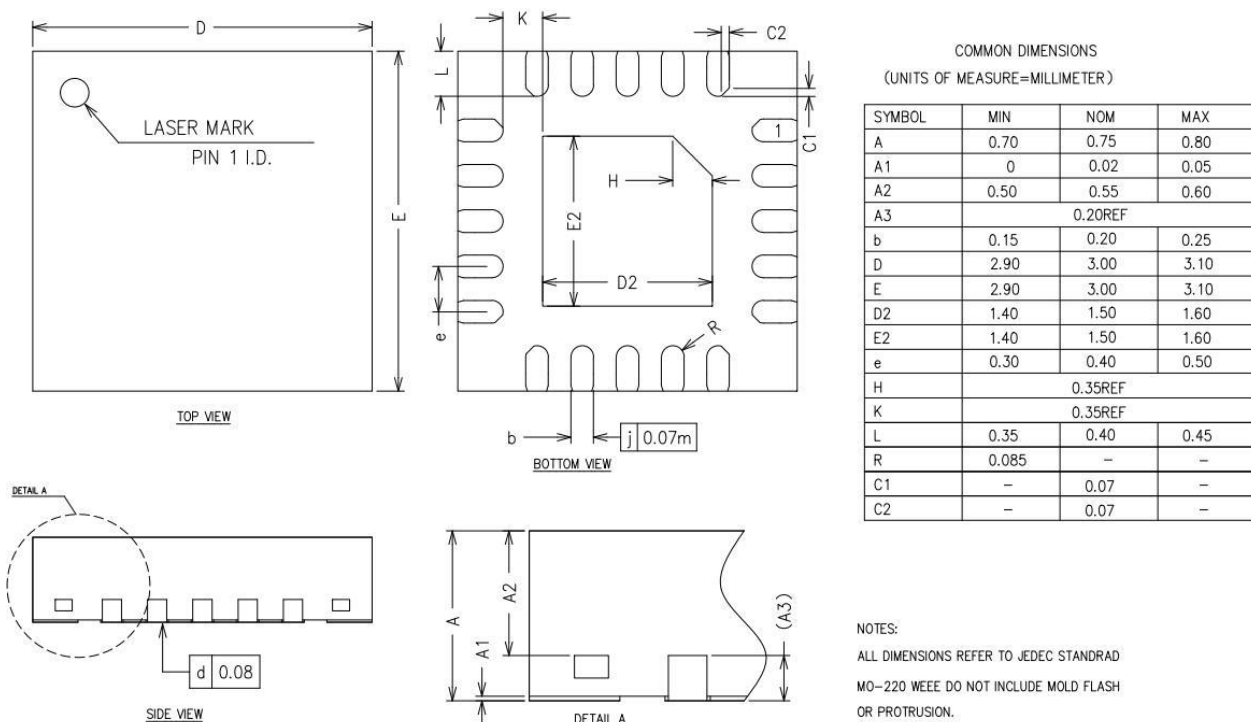
最低工作电压 VS 温度



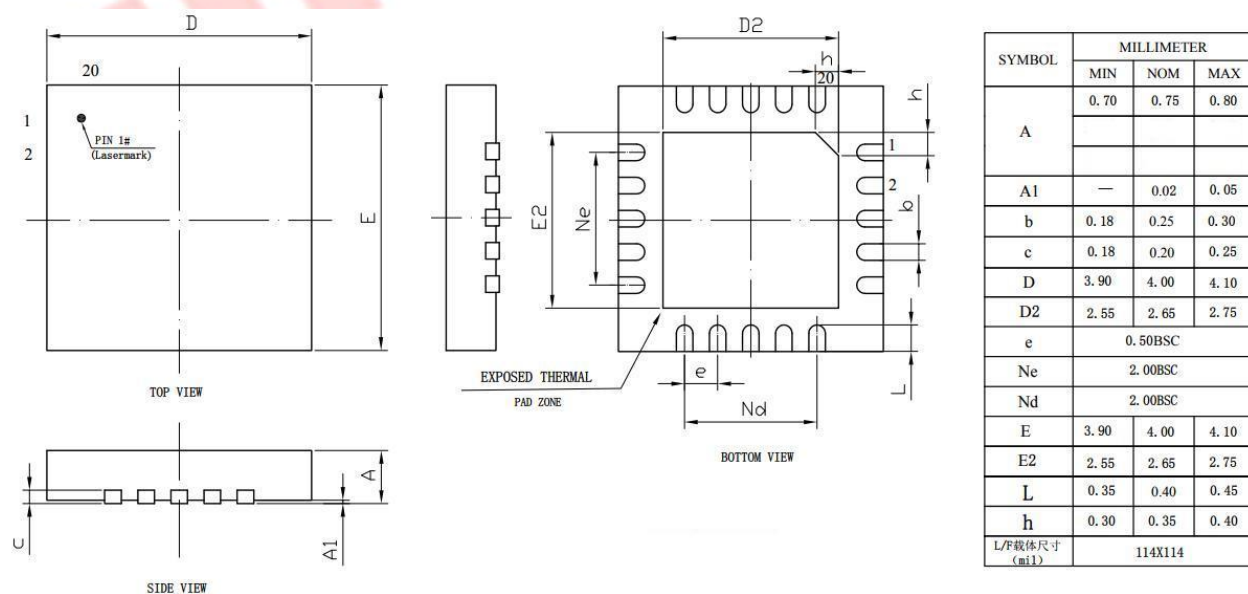


13 封装尺寸

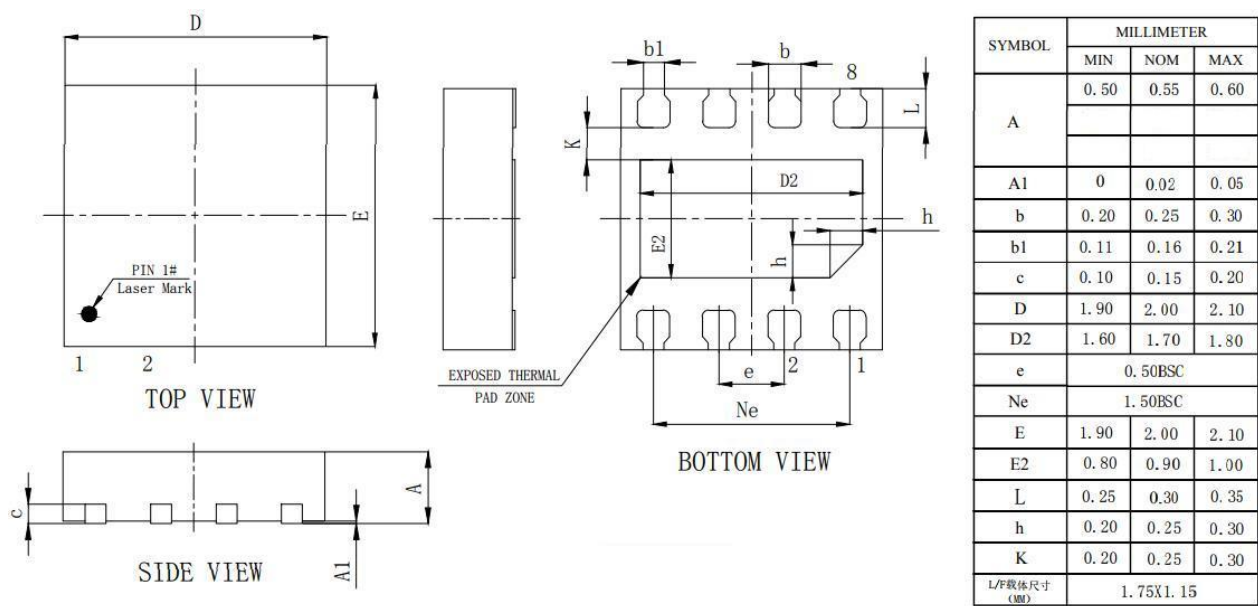
13.1 QFN20 (3×3)



13.2 QFN20 (4×4)



13.3 DFN8



14 修订记录

版本	修订日期	修订内容
V1.0	2020-01-15	初版发布;
V1.1	2020-03-10	新增 DFN8 封装 A0ZN 和 QFN20(4×4)封装 A0YE;
V1.2	2020-06-19	修改引脚排列示意图, 增加 bottom pad 引脚说明; 新增 PWM1 关闭而 FPWM1 使能时禁止写 T1DATA 的注释;