

*SinoMCU 8 位单片机*

**MC32P7031**

**用户手册**

V1.9



## 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	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	13
3.4	堆栈	14
3.5	控制寄存器	15
3.6	用户配置字	18
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	21
4.3	外部晶体振荡器	21
4.4	外部 RC 振荡器	21
5	系统工作模式	22
5.1	高速模式	23
5.2	低速模式	23
5.3	休眠模式	24
5.4	空闲模式（绿色模式）	24
6	复位	25
6.1	复位条件	25
6.2	上电复位	26
6.3	外部复位	26
6.4	低电压复位	26
6.5	看门狗复位	26
7	I/O 端口	27
7.1	通用 I/O 功能	27
7.2	内部上拉电阻	28
7.3	端口模式控制	29
8	定时器 TIMER	30
8.1	看门狗定时器 WDT	30
8.2	定时器 T0	30
8.3	定时器 T1	33
8.4	定时器 T2	35

9	模数转换器 ADC .....	38
9.1	ADC 概述 .....	38
9.2	ADC 操作步骤 .....	39
9.3	ADC 相关寄存器 .....	39
9.4	ADC 操作说明和注意事项 .....	41
10	中断 .....	43
10.1	外部中断 .....	43
10.2	定时器中断 .....	43
10.3	ADC 中断 .....	44
10.4	中断相关寄存器 .....	44
11	特性曲线 .....	46
11.1	I/O 特性 .....	46
11.2	功耗特性 .....	48
11.3	模拟电路特性 .....	49
12	封装尺寸 .....	58
12.1	SOP16 .....	58
12.2	DIP16 .....	58
12.3	SOP14 .....	59
12.4	DIP14 .....	59
12.5	SOP8 .....	60
12.6	DIP8 .....	60
12.7	MSOP10 .....	61
12.8	MSOP8 .....	61
13	修订记录 .....	62

## 1 产品概要

### 1.1 产品特性

- 8 位 CPU 内核
  - ◇ 精简指令集, 4 级缓存寄存器型堆栈
  - ◇ 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<sub>CPU</sub> 可配置为 1/2/4/8/16/32/64/128 分频
  - ◇ 系统低频时钟下 F<sub>CPU</sub> 固定为 4 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
- 3 组共 14 个 I/O
  - ◇ P0 (P00~P04), P4 (P40~P45), P5 (P52~P54)
  - ◇ 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 VPP 输入
  - ◇ P03/P02 可复用为外部时钟振荡器输入/输出
  - ◇ 所有端口均内置上拉电阻, 均可单独使能
  - ◇ P0 所有端口均支持输入电平变化唤醒功能
- 系统时钟源
  - ◇ 内置高频 RC 振荡器 (455KHz~16MHz), 可用作系统高频时钟源
  - ◇ 支持外接高频晶体振荡器 (455KHz/4MHz~16MHz), 可用作系统高频时钟源
  - ◇ 支持外接 RC 振荡器 (0~4MHz), 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
  - ◇ 支持外接低频晶体振荡器 (32768Hz), 可用作系统高频或低频时钟源
- 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
  - ◇ 空闲模式: CPU 停止运行, 高频时钟源可选停止或工作, 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: 8192 / 内部低频 RC 振荡器频率 (F<sub>LIRC</sub>), 约为 256ms@VDD=5V
  - ◇ 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
  - ◇ 8 位定时器 T0, 可实现外部计数、BUZ 和 PWM 功能, 支持空闲模式下溢出唤醒
  - ◇ 8 位定时器 T1, 可实现外部计数、BUZ 和 PWM 功能
  - ◇ 8 位定时器 T2, 可实现 RTC 计数、BUZ 和 PWM 功能, 支持空闲和休眠模式下溢出唤醒
- 1 个 12 位高精度逐次逼近型 ADC
  - ◇ 6 路外部通道: AIN0~AIN4/AIN6; 1 路内部通道: VDD/4
  - ◇ 参考电压可选: VDD、内部参考电压 V<sub>IR</sub> (2V/3V/4V)、外部参考电压 V<sub>ER</sub> (VREFH 输入)
  - ◇ ADC 时钟: F<sub>CPU</sub> 的 1/2/8/16 分频
- 中断
  - ◇ 外部中断 (INT0~INT1)
  - ◇ 定时器中断 (T0~T2)

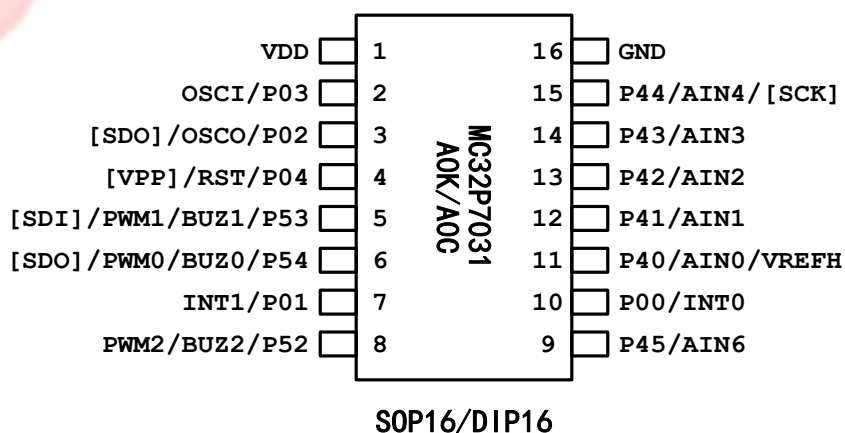
- ◇ ADC 中断
- 低电压复位 LVR: 多级复位电压可配置
- 低电压检测 LVD: 多级检测电压可配置
- 工作电压
  - ◇ VLVR30 ~ 5.5V @ Fcpu = 0~8MHz
  - ◇ VLVR27 ~ 5.5V @ Fcpu = 0~4MHz
  - ◇ VLVR20 ~ 5.5V @ Fcpu = 0~2MHz
- 封装形式
  - ◇ SOP16/DIP16/SOP14/DIP14/SOP8/DIP8/MSOP10/MSOP8

## 1.2 订购信息

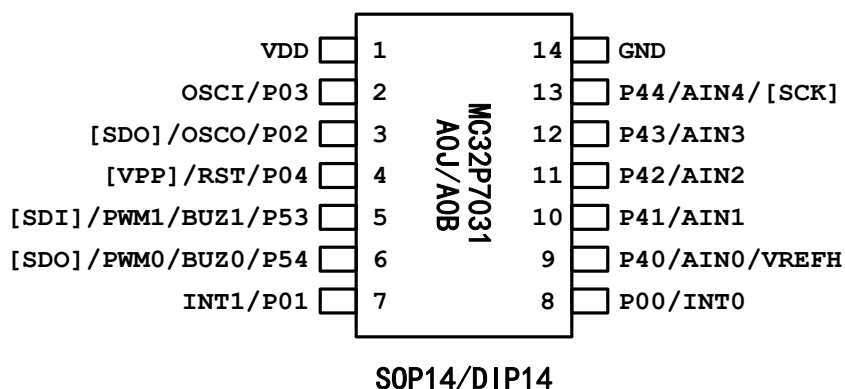
产品名称	封装形式	备注
MC32P7031A0K	SOP16	
MC32P7031A0C	DIP16	
MC32P7031A0J	SOP14	
MC32P7031A0B	DIP14	
MC32P7031A0H	SOP8	
MC32P7031A0A	DIP8	
MC32P7031A0I	MSOP10	
MC32P7031A0F	MSOP8	

## 1.3 引脚排列

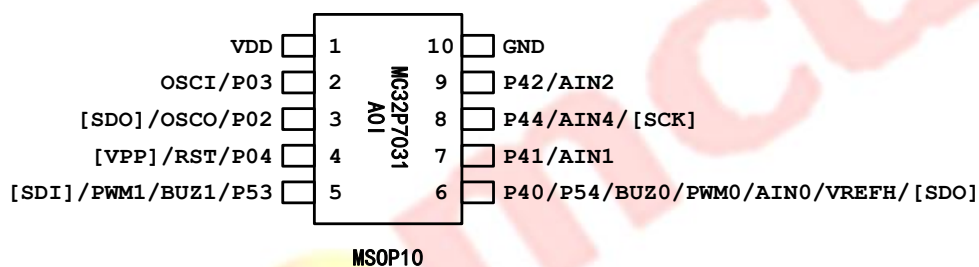
### MC32P7031A0K/A0C



## MC32P7031A0J/A0B

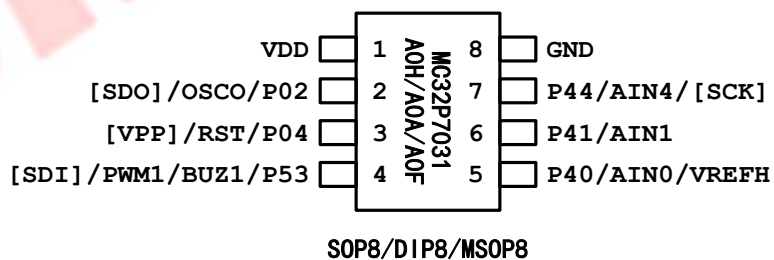


## MC32P7031A0I



**注：**端口 P40 和 P54 均从引脚 6 封出，应用时需确保共用引脚的端口在工作模式、内部上拉或下拉电阻、复用功能、数模端口等设置上相互之间不会产生冲突。

## MC32P7031A0H/A0A/A0F



## 1.4 端口说明

端口名称	类型	功能说明
VDD	P	电源
GND	P	地
P0 (除 P04), P4, P5	D	GPIO, 内部上拉
P04	D	开漏 IO, 内部上拉
INT0~INT1	DI	外部中断输入, 及定时器 T0~T1 的外部计数输入
BUZ0/PWM0	DO	定时器 T0 的 BUZ/PWM 输出
BUZ1/PWM1	DO	定时器 T1 的 BUZ/PWM 输出
BUZ2/PWM2	DO	定时器 T2 的 BUZ/PWM 输出
AIN0~AIN4/AIN6	AI	ADC 模拟输入通道
VREFH	AI	ADC 外部参考电压输入
OSCI, OSCO	A	外部时钟振荡器输入/输出
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

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。

## 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	40	mA
流出 GND 最大电流	IGNDmax	40	mA

注：若芯片工作条件超过极限值，则将造成永久性损坏；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/2	VLVR30		5.5	V
			Fcpu=4MHz@FHIRC/4	VLVR27		5.5	
			Fcpu=2MHz@FHIRC/8	VLVR20		5.5	
			Fcpu=1MHz@FHIRC/16	VLVR20		5.5	
			Fcpu=500KHz@FHIRC/32	VLVR20		5.5	
输入漏电流	Ileak	所有输入脚	VDD=5V	-1		1	uA
输入高电平	Vih	输入脚(除 P04)		0.7VDD			V
		P04		0.8VDD			V
输入低电平	Vil	输入脚(除 P04)				0.3VDD	V
		P04				0.2VDD	V
输出拉电流	Ioh	推挽输出脚	Voh=VDD-0.5V	6	12	24	mA
输出灌电流	Iol	所有输出脚	Vol=0.5V	8	16	32	mA
上拉电阻	Rpu	P0, P4, P5	VDD=5V, Vin=0		70		KΩ
			VDD=3V, Vin=0		140		KΩ
高速模式功耗	Ifast	VDD	VDD=5V, Fcpu=1MHz@HIRC		1.3		mA
			VDD=3V, Fcpu=1MHz@HIRC		0.8		mA
低速模式功耗	Islow	VDD	VDD=5V, Fcpu=8KHz@LIRC, HIRC 开		500		uA
			VDD=5V, Fcpu=8KHz@LIRC, HIRC 关		5	10	uA
			VDD=3V, Fcpu=8KHz@LIRC, HIRC 关		2	4	uA
空闲模式功耗	Ihold	VDD	VDD=5V, 空闲模式, LVR/LVD 关		3	6	uA
			VDD=3V, 空闲模式, LVR/LVD 关		1	3	uA
			VDD=5V, 空闲模式, LVR/LVD 开		4	8	uA



休眠模式功耗	Istop	VDD	VDD=5V, 休眠模式, LVR/LVD 关		0.1	1	uA
			VDD=5V, 休眠模式, LVR/LVD 开		1	3	uA
低压检测电压	VLVD	VDD	VLVDS=0001(LVD 关闭)	-15%	1.2	+15%	V
			VLVDS=其他	-15%		+15%	
低压复位电压	VLVR20	VDD		-15%	2.0	+15%	V
	VLVR27			-15%	2.7	+15%	
	VLVR30			-15%	3.0	+15%	
	其他			-15%		+15%	
LVD/LVR 回滞电压		VDD			6%	12%	

注：功耗特性参数的条件说明中，诸如 HIRC/LIRC/WDT/LVR/LVD/ADC 等未注明模块，默认其为关闭状态。

## 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	FHIRC1	VDD=5V, T=25°C	-2%	16	+2%	MHz
		VDD=2.0V~5.5V, T=-40°C~85°C	-5%		+5%	
	FHIRC2	VDD=5V, T=25°C	-2%	8	+2%	MHz
		VDD=2.0V~5.5V, T=-40°C~85°C	-8%		+8%	
	FHIRC3	VDD=5V, T=25°C	-2%	4	+2%	MHz
		VDD=2.0V~5.5V, T=-40°C~85°C	-10%		+10%	
	FHIRC4	VDD=5V, T=25°C	-2%	2	+2%	MHz
		VDD=2.0V~5.5V, T=-40°C~85°C	-20%		+20%	
	FHIRC5	VDD=5V, T=25°C	-2%	1	+2%	MHz
		VDD=2.0V~5.5V, T=-40°C~85°C	-20%		+20%	
	FHIRC6	VDD=5V, T=25°C	-2%	455	+2%	KHz
		VDD=2.0V~5.5V, T=-40°C~85°C	-20%		+20%	
LIRC 振荡频率	FLIRC	VDD=5V, T=25°C	-50%	32	+50%	KHz
16M 晶振起振电压		T=25°C	3.0			V
8M 晶振起振电压		T=25°C	2.3			V
4M 晶振起振电压		T=25°C	2.0			V
455K 晶振起振电压		T=25°C	2.0			V
32768 晶振起振电压		T=25°C	1.8			V
32768 晶振起振时间		VDD=5V, T=25°C		1		s
外部 RC 振荡频率	FERC1	VDD=5V, T=25°C, R=1KΩ, C=100pF	-50%	3.4	+50%	MHz
	FERC2	VDD=5V, T=25°C, R=3KΩ, C=100pF	-50%	1.3	+50%	MHz
	FERC3	VDD=5V, T=25°C, R=1KΩ, C=1000pF	-50%	590	+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 <sub>ADC</sub>	T=-40°C~85°C	2.0		5.5	V
积分线性误差	INL	V <sub>REF</sub> =VDD, F <sub>ADC</sub> =500KHz			±3	LSB
微分线性误差	DNL	V <sub>REF</sub> =VDD, F <sub>ADC</sub> =500KHz			±3	LSB
零点偏移误差	EZ	V <sub>REF</sub> =VDD, F <sub>ADC</sub> =500KHz			±3	LSB
增益误差	ET	V <sub>REF</sub> =VDD, F <sub>ADC</sub> =500KHz			±3	LSB
转换精度	ACC	V <sub>REF</sub> =VDD, F <sub>ADC</sub> =2MHz			±4	LSB
		V <sub>REF</sub> =V <sub>ER</sub> (2.5V), F <sub>ADC</sub> =1MHz			±4	LSB
		V <sub>REF</sub> =V <sub>IR</sub> (2V/3V/4V), F <sub>ADC</sub> =500KHz			±8	LSB
转换时钟	F <sub>ADC</sub>	VDD=5V			1	MHz
转换时间	T <sub>con</sub>		16		27	1/F <sub>ADC</sub>
ADC 输入电压	V <sub>AIN</sub>		GND		V <sub>REF</sub>	V
ADC 输入阻抗	R <sub>AIN</sub>		2			MΩ
ADC 输入电流	I <sub>AIN</sub>				10	uA
ADC 动态电流	I <sub>ADD</sub>	VDD=5V, AD 转换中		1	3	mA
ADC 静态电流	I <sub>ADS</sub>	VDD=5V, ADC 关闭		0.1	1	uA
失调电压	VOS	V <sub>REF</sub> =VDD, F <sub>ADC</sub> =500KHz	-2		+2	mV
		V <sub>REF</sub> =VDD, F <sub>ADC</sub> =1MHz	-10		+10	mV
模拟信号源推荐阻抗	Z <sub>AIN</sub>				10	KΩ
ADC 内部 1/4 分压电阻	R <sub>EVI</sub>	V <sub>in</sub> =VDD=2.0V~5.5V	16	24	32	KΩ
电阻分压比值			-1%	1/4	+1%	V <sub>EVI</sub>
ADC 参考电压	V <sub>REF</sub>	选择 VDD		VDD		V
		选择内部参考电压 V <sub>IR</sub> , T=25°C	-1.5%	2/3/4	+1.5%	
		选择内部参考电压 V <sub>IR</sub> , T=-40°C~85°C	-3%		+3%	
		选择外部参考电压 V <sub>ER</sub>	2		VDD	
V <sub>IR</sub> 有效工作电压	V <sub>VIR</sub>	选择内部参考电压 V <sub>IR</sub>	V <sub>IR</sub> +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。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	标志
ADDAR R	ACC 和 R 相加，结果存入 ACC	ACC+R→ACC	1	C,DC,Z
ADDRA R	ACC 和 R 相加，结果存入 R	ACC+R→R	1+M	C,DC,Z
ADCAR R	ACC 和 R 相加（带 C 标志），结果存入 ACC	ACC+R+C→ACC	1	C,DC,Z
ADCRA R	ACC 和 R 相加（带 C 标志），结果存入 R	ACC+R+C→R	1+M	C,DC,Z
ASUBAR R	ACC 和 R 相减，结果存入 ACC	ACC-R→ACC	1	C,DC,Z
ASUBRA R	ACC 和 R 相减，结果存入 R	ACC-R→R	1+M	C,DC,Z
ASBCAR R	ACC 和 R 相减（带 C 标志），结果存入 ACC	ACC-R-/C→ACC	1	C,DC,Z
ASBCRA R	ACC 和 R 相减（带 C 标志），结果存入 R	ACC-R-/C→R	1+M	C,DC,Z
ANDAR R	ACC 和 R 与操作，结果存入 ACC	ACC and R→ACC	1	Z
ANDRA R	ACC 和 R 与操作，结果存入 R	ACC and R→R	1+M	Z
ORAR R	ACC 和 R 或操作，结果存入 ACC	ACC or R→ACC	1	Z
ORRA R	ACC 和 R 或操作，结果存入 R	ACC or R→R	1+M	Z
XORAR R	ACC 和 R 异或操作，结果存入 ACC	ACC xor R→ACC	1	Z
XORRA R	ACC 和 R 异或操作，结果存入 R	ACC xor R→R	1+M	Z
RLAR R	R 循环左移（带 C 标志），结果存入 ACC	R[7]→C R[6:0]→ACC[7:1] C→ACC[0]	1	C
RLR R	R 循环左移（带 C 标志），结果存入 R	R[7]→C R[6:0]→R[7:1] C→R[0]	1+M	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	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+M	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	-
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+M	-
MOVRA R	将 ACC 存入 R	ACC→R	1	-
MOVAR R	将 R 存入 ACC	R→ACC	1	Z
XCH R	ACC 和 R 交换	ACC↔R	1+M	-

CLRR	R	清零 R	$0 \rightarrow R$	1	Z
JZR	R	R 自加 1; 结果为 0 则跳过下一条指令	$R+1 \rightarrow R$ ; 结果为 0 则 $PC+2 \rightarrow PC$	$1+J+M$	-
JZAR	R	R 加 1, 结果存入 ACC; 结果为 0 则跳过下一条指令	$R+1 \rightarrow ACC$ ; 结果为 0 则 $PC+2 \rightarrow PC$	$1+J$	-
DJZR	R	R 自减 1; 结果为 0 则跳过下一条指令	$R-1 \rightarrow R$ ; 结果为 0 则 $PC+2 \rightarrow PC$	$1+J+M$	-
DJZAR	R	R 减 1, 结果存入 ACC; 结果为 0 则跳过下一条指令	$R-1 \rightarrow ACC$ ; 结果为 0 则 $PC+2 \rightarrow PC$	$1+J$	-
BCLR	R, b	将 R 的第 b 位清 0	$0 \rightarrow R[b]$	$1+M$	-
BSET	R, b	将 R 的第 b 位置 1	$1 \rightarrow R[b]$	$1+M$	-
JBCLR	R, b	若 R 的第 b 位为 0, 则跳过下一条指令	若 $R[b]=0$ , 则 $PC+2 \rightarrow PC$	$1+J$	-
JBSET	R, b	若 R 的第 b 位为 1, 则跳过下一条指令	若 $R[b]=1$ , 则 $PC+2 \rightarrow PC$	$1+J$	-
CMPR	R	ACC 和 R 比较; 相等则跳过下一条指令	$ACC-R$ ; 结果为 0 则 $PC+2 \rightarrow PC$	$1+J$	C,Z
CMPI	K	ACC 和 K 比较; 相等则跳过下一条指令	$ACC-K$ ; 结果为 0 则 $PC+2 \rightarrow PC$	$1+J$	C,Z
ADDAI	K	ACC 和 K 相加, 结果存入 ACC	$ACC+K \rightarrow ACC$	1	C,DC,Z
ASUBAI	K	ACC 和 K 相减, 结果存入 ACC	$ACC-K \rightarrow ACC$	1	C,DC,Z
ANDAI	K	ACC 和 K 与操作, 结果存入 ACC	$ACC \text{ and } K \rightarrow ACC$	1	Z
ORAI	K	ACC 和 K 或操作, 结果存入 ACC	$ACC \text{ or } K \rightarrow ACC$	1	Z
XORAI	K	ACC 和 K 异或操作, 结果存入 ACC	$ACC \text{ xor } K \rightarrow ACC$	1	Z
MOVAI	K	将 K 存入 ACC	$K \rightarrow ACC$	1	-
CALL	K	子程序调用	$PC+1 \rightarrow TOS$ $K \rightarrow PC[10:0]$	2	-
GOTO	K	无条件跳转	$K \rightarrow PC[10:0]$	2	-
RETURN		从子程序返回	$TOS \rightarrow PC$	2	-
RETIE		从中断返回	$TOS \rightarrow PC$ $1 \rightarrow GIE$	2	-
NOP		空操作	空操作	1	-
PUSH		缓存 ACC 和 C,DC,Z	$ACC \text{ 和 } C,DC,Z \rightarrow BUF$	1	-
POP		恢复 ACC 和 C,DC,Z	$BUF \rightarrow ACC \text{ 和 } C,DC,Z$	1	C,DC,Z
MOVC		读取程序存储器内容	$ROM(@[FSR1:FSR0]) \rightarrow [HIBYTE:ACC]$	2	-

注:

- 1、ACC-算数逻辑单元累加器, 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  $J=1$ , 否则  $J=0$ ;
- 3、对于以 GPR 为目的寄存器的指令, 若对 GPR 仅有写操作, 则  $M=0$ , 若对 GPR 还有读操作, 则  $M=1$ ;
- 4、PUSH/POP 指令涉及的缓存器 BUF 仅有 1 层, 所以 PUSH/POP 必须成对使用, 否则将导致数据错误;
- 5、指令 JBSET/JBCLR/CMPR/JZR/JZAR/DJZR/DJZAR 对 IOP0/IOP4/IOP5 直接操作时, 因指令隐含的回写动作可能会改写寄存器中的其他位而导致发生不可预估的错误, 所以推荐先将 IOP0/IOP4/IOP5 读入缓存 RAM 中, 再用上述指令对缓存 RAM 的数据进行条件判断操作;

### 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，可通过指令 MOV<sub>C</sub> 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
MOVC      ; 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/SFR 可直接寻址或通过 INDF 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
00H-7FH	GPR	通用数据寄存器区							
80H-87H	SFR			HIBYTE	FSR0	FSR1		PFLAG	
88H-8FH									
90H-97H									
98H-9FH									
A0H-A7H									

A8H-AFH								ANSEL	VREF
B0H-B7H		ADCR	ADRH	ADRL					
B8H-BFH		OEP0							PEDGE
C0H-C7H					OEP4	OEP5			
C8H-CFH		INTF	INTE	OSCM		WDTR	T0D	PCL	PCH
D0H-D7H		IOP0				IOP4	IOP5		
D8H-DFH		TXCR		T0CR	T0C	T1CR	T1C	T1D	STKP
E0H-E7H		PUP0				PUP4	PUP5		INDF
E8H-EFH						T2CR	T2C	T2D	
F0H-F7H									
F8H-FFH		STK3L	STK3H	STK2L	STK2H	STK1L	STK1H	STK0L	STK0H

**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。**

直接寻址模式，是以指令的低 8 位为数据存储器地址，通过指令访问，寻址范围为 00H~FFH。例：通过直接寻址模式将数据 55H 写入数据存储器 10H 地址

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

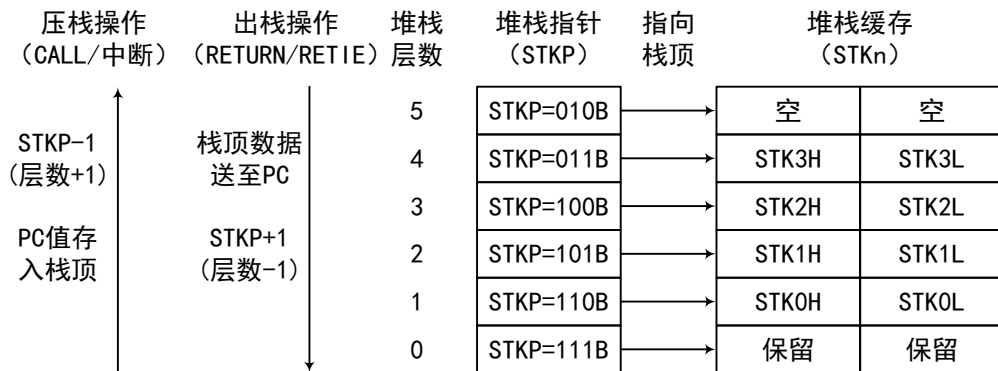
间接寻址模式，是以 [FSR1:FSR0] 为数据存储器地址指针，通过 INDF 访问，寻址范围为 0000H~FFFFH。例：通过间接寻址模式将数据 55H 写入数据存储器 0010H 地址

```
MOVAI    00H
MOVRA    FSR1
MOVAI    10H
MOVRA    FSR0
MOVAI    55H
MOVRA    INDF          ： 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

4 级 11 位堆栈缓存寄存器 STKn (n=3-0)，位于特殊功能寄存器 SFR 区，3 位堆栈指针 STKP[2:0] 位于寄存器 STKP 中，指向堆栈栈顶。CPU 在响应中断或执行子程序调用指令时，先将 STKP 减 1，然后将 PC 值压入 STKP 所指栈顶的堆栈缓存寄存器中；当 CPU 执行中断或子程序返回指令时，先将栈顶数据送入 PC 中，再将 STKP 加 1 指向新的栈顶地址。若在堆栈满时压栈、或堆栈空时出栈，会导致数据错误。



### 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: 间接寻址模式的指针低 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: 间接寻址模式的指针高 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
INDF	INDF7	INDF6	INDF5	INDF4	INDF3	INDF2	INDF1	INDF0
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]      **INDF[7:0]** – 间接寻址寄存器  
INDF: INDF 不是物理寄存器, 对 INDF 寻址实际是对  $FSR1 \times 256 + FSR0$  指向的数据存储器地址进行访问, 从而实现间接寻址功能。



## 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: 用于存放通过指令 MOV<sub>C</sub> 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

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCH	-	-	-	-	-	PC10	PC9	PC8
R/W	-	-	-	-	-	W	W	W
初始值	-	-	-	-	-	0	0	0

BIT[2:0] PC[10:8] – 程序指针计数器高 3 位

注: PCH 只可写不可读, 禁止使用带回写功能的指令直接操作 PCH。

	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:  $PC = \text{指令码低 11 位}$ ;
- ✧ 返回指令 RETIE/RETURN:  $PC = \text{堆栈栈顶 (TOS)}$ ;

对 PCL 操作指令:

- ✧ 对 PCL 操作的加法指令:  $PC = (PC[10:0] + ALU[7:0])$ ;
- ✧ 对 PCL 操作的其它指令:  $PC = \{PC[10:8]: ALU[7:0](ALU \text{ 运算结果})\}$ ;

注: 以 PCL 为目的寄存器的算术/逻辑操作指令中, 仅有加法指令产生进位时会影响 PCH 的值, 而其他指令均不会影响 PCH 的值。

## CPU 状态寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PFLAG	NTO	NPD	LVD36	LVD24	-	C	DC	Z
R/W	R/W	R/W	R	R	-	R/W	R/W	R/W
初始值	X	X	0	0	-	X	X	X



BIT[7,6] NTO,NPD – 复位状态标志位

NTO	NPD	复位状态
0	0	WDT 溢出复位
0	1	保留
1	0	LVR 低电压复位
1	1	RST 外部复位

BIT[5] LVD36 – 3.6V 检测状态标志位（仅配置字 VLVD3[3]=1 时有效）

0: VDD 高于 3.6V（有回滞）；

1: VDD 低于 3.6V；

BIT[4] LVD24 – 2.4V 检测状态标志位（仅配置字 VLVD3[3]=0 时有效）

0: VDD 高于 2.4V（有回滞）；

1: VDD 低于 2.4V；

**注：因 LVD 电路有回滞特性（回滞电压典型值为 6%），所以 VDD 需上升至检测量值+6%后标志位才被清 0。**

BIT[2] 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；比较运算结果<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；比较运算结果≥0；

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] 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；

### 堆栈指针寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STKP	GIE	-	-	-	-	STKP2	STKP1	STKP0
R/W	R/W	-	-	-	-	R/W	R/W	R/W
初始值	0	-	-	-	-	1	1	1

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[2:0] STKP[2:0] – 堆栈指针控制位

**注：虽然堆栈指针的复位初始值已为 111B，但仍推荐在程序初始部分重新设定 STKP=7 并禁止程序再访问堆栈相关寄存器，以确保芯片能正常使用堆栈。**

## 堆栈缓存寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STKnH	-	-	-	-	-	STKn10	STKn9	STKn8
R/W	-	-	-	-	-	R/W	R/W	R/W
初始值	-	-	-	-	-	0	0	0

BIT[2:0] STKn[10:8] – 堆栈缓存寄存器 STKn 的高 3 位 (n=3-0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STKnL	STKn7	STKn6	STKn5	STKn4	STKn3	STKn2	STKn1	STKn0
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] STKn[7:0] – 堆栈缓存寄存器 STKn 的低 8 位 (n=3-0)

## 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
FOSCS	系统高频时钟选择： 000: 内部高频 HIRC; 010: RTC 模式; 100: 外部 4M~16M 晶体; 101: 外部 455K 晶体; 110: 外部 32K 晶体; 111: 外部 RC;
FAS	HIRC 振荡频率选择： 000: 16M; 001: 8M; 010: 4M; 011: 2M; 100: 1M; 101: 455K;
FCPUS	高频时钟下 FCPU 分频选择： 000: FCPU=FHOSC/1; 001: FCPU=FHOSC/2; 010: FCPU=FHOSC/4; 011: FCPU=FHOSC/8; 100: FCPU=FHOSC/16; 101: FCPU=FHOSC/32; 110: FCPU=FHOSC/64; 111: FCPU=FHOSC/128;
WDTC	WDT 模式设置： 00: WDT 始终关闭; 01: WDT 低功耗模式下关闭; 11: WDT 始终开启;
VLVRS	LVR 复位电压选择: (LVR 电压应满足由 FCPU 决定的工作电压特性) 0100: 1.8V; 0101: 2.0V; 0110: 2.2V; 0111: 2.4V; 1000: 2.5V; 1001: 2.6V; 1010: 2.7V; 1011: 2.8V; 1100: 3.0V; 1101: 3.2V; 1110: 3.6V; 1111: 3.8V;

<b>VLVDS</b>	<p>LVD 检测电压选择:</p> <p>0000: 1.8V;    0001: 关闭;    0010: 2.0V;    0011: 2.1V;  0100: 2.2V;    0101: 2.4V;    0110: 2.5V;    0111: 2.6V;  1000: 2.7V;    1001: 2.8V;    1010: 3.0V;    1011: 3.2V;  1100: 3.3V;    1101: 3.6V;    1110: 4.0V;    1111: 4.2V;</p>
<b>SAMPTS</b>	<p>AD 采样时间选择:</p> <p>00: 4 个 ADCLK;                      01: 6 个 ADCLK;  10: 12 个 ADCLK;                      11: 14 个 ADCLK;</p>
<b>MCLRE</b>	<p>RST 外部复位设置:</p> <p>0: P04 为输入/输出脚;                      1: P04 为外部复位脚;</p>
<b>ENCR</b>	<p>程序代码加密设置:</p> <p>0: 程序代码加密;                      1: 程序代码不加密;</p>

## 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  $F_{HOSC}$  或系统低频时钟  $F_{LOSC}$  下工作，部分模块的时钟还可在  $F_{HOSC}$  和  $F_{LOSC}$  之间切换。

系统高频时钟  $F_{HOSC}$  可通过配置字  $FOSCS$  选择以下时钟：

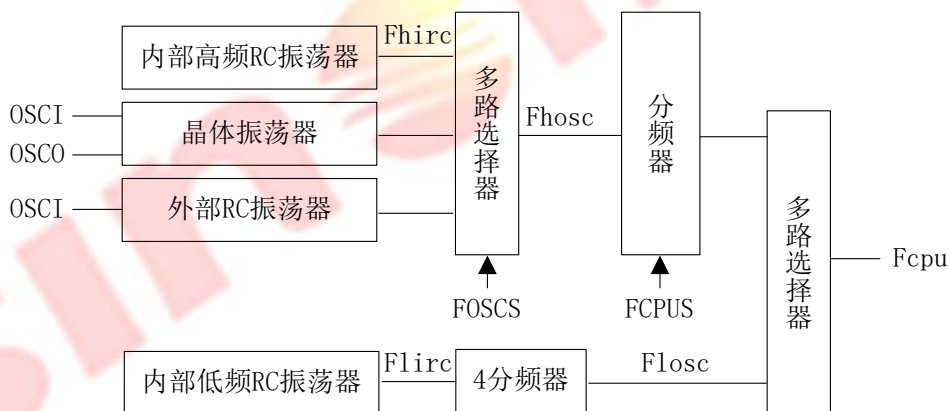
- ✧ 内部高频 RC 振荡器  $HIRC$  (16MHz/8MHz/4MHz/2MHz/1MHz/455KHz) 时钟  $F_{HIRC}$ ;
- ✧ 外部高频晶体振荡器  $HEXT$  (4MHz~16MHz/455KHz) 时钟  $F_{HEXT}$ ;
- ✧ 外部低频晶体振荡器  $LEXT$  (32768Hz) 时钟  $F_{LEXT}$ ;
- ✧ 外部 RC 振荡器  $ERC$  (0~4MHz) 时钟  $F_{ERC}$ ;

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。选择  $F_{HOSC}$  时，CPU 的时钟频率  $F_{CPU}$  通过配置字  $FCPUS$  选择；选择  $F_{LOSC}$  时， $F_{CPU}$  固定为  $F_{LOSC}$  的 4 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 个振荡频率通过配置字  $FAS$  可配置为 16MHz/8MHz/4MHz/2MHz/1MHz/455KHz 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 外部晶体振荡器

芯片支持外接高频 455KHz/4MHz~16MHz、或低频 32768Hz 的晶体振荡器作为系统高频时钟源。还可在系统时钟配置为 RTC 模式时，选择外接 32768Hz 晶体振荡器作为外部 RTC 时钟源。

外接晶振的实际使用中，对地的负载电容  $C_x$  是必须的。用户应使晶体离 OSCI/OSCO 引脚的距离尽可能短，这样有助于振荡器的起振和振荡的稳定性。

下表是典型频率晶振选用电容  $C_x$  的推荐值和相应最低起振电压参考值。

晶振频率 (Hz)	电容 $C_x$ (pF)	最低起振电压 (V)
16M	10	3.0
8M	10/20	2.3
4M	10/20	2.2
455K	100/220	2.0
32768	10/20	1.8

**注：**因晶体品牌繁多且工艺差异较大，故上表中的参数仅供参考，具体应用请以晶体的实测结果为准。

## 4.4 外部 RC 振荡器

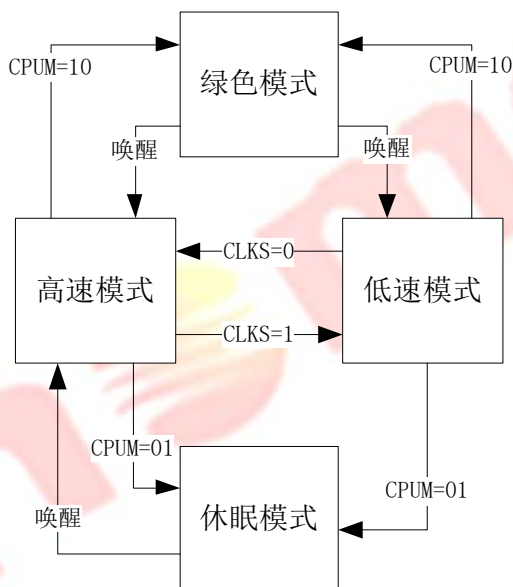
芯片支持外接 RC 振荡器作为系统高频时钟源。外部 RC 振荡电路只需要和 OSCI 引脚连接，电容值不能低于 100pF，电阻值和电容值共同决定 RC 的振荡频率，最高支持 4MHz。

## 5 系统工作模式

本芯片支持高速模式、低速模式、空闲模式和休眠模式等多种系统工作模式，其中空闲模式和休眠模式为低功耗模式，CPU 停止工作。

工作模式	切换条件
高速模式	系统时钟切换到高频振荡器 (CLKS=0)
低速模式	系统时钟切换到低频振荡器 (CLKS=1)
空闲模式	CPUM=10
休眠模式	CPUM=01 或 CLKS=0 并且 HOFF=1

模式切换示意图



工作模式寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCM	-	-	-	CPUM1	CPUM0	CLKS	HOFF	-
R/W	-	-	-	R/W	R/W	R/W	R/W	-
初始值	-	-	-	0	0	0	0	-

BIT[4:3] CPUM[1:0] – 系统工作模式控制位

CPUM[1:0]	系统工作模式
00	高/低速模式（硬件自动清 0）
01	进入休眠模式
10	进入空闲模式
11	保留

- BIT[2]     **CLKS** – 系统时钟选择位  
 0: 系统高频时钟作为系统时钟，系统高频时钟的 1~16 分频作为 CPU 时钟；  
 1: 系统低频时钟的 4 分频作为系统时钟和 CPU 时钟；
- BIT[1]     **HOFF** – 高频时钟源禁止位  
 0: 使能高频时钟源；  
 1: 关闭高频时钟源；

### 系统功能和模块在各工作模式下的状态

功能/模块的工作状态	高速模式	低速模式	空闲模式	休眠模式
内/外部高频时钟源	工作	HOFF 控制	HOFF 控制	停止
内部低频 RC 振荡器	工作	工作	工作	停止
CPU	工作	工作	停止	停止
看门狗定时器 WDT	WDTC 决定	WDTC 决定	WDTC 决定	WDTC 决定
T0	TC0EN 控制	TC0EN 控制	TC0EN 控制，仅 PWM/BUZ 有效	无效
T1	TC1EN 控制	TC1EN 控制	TC1EN 控制，仅 PWM/BUZ 有效	无效
T2	TC2EN 控制	TC2EN 控制	TC2EN 控制，仅 PWM/BUZ 有效	无效
内部中断	全部有效	全部有效	T0IF, T2IF	全部无效
外部中断	全部有效	全部有效	全部有效	全部无效
唤醒方法	-	-	P0, T0IF, T2IF, 复位	P0, T2IF(RTC), 复位

## 5.1 高速模式

高速模式是系统高速时钟正常工作模式，系统时钟源由高速振荡器提供。程序被执行。上电复位或任意一种复位触发后，系统进入高速模式执行程序。当系统从休眠模式被唤醒后进入高速模式。高速模式下，高速振荡器正常工作，功耗最大。

- ✧ 程序被执行，所有的功能都可以控制；
- ✧ 系统速率为高速；
- ✧ 高速振荡器和内部低速 RC 振荡器都正常工作；
- ✧ 通过 OSCM 寄存器，系统可以从高速模式切换到其他任何一种模式；
- ✧ 系统从休眠模式唤醒后进入高速模式；
- ✧ 低速模式可以切换到高速模式；
- ✧ 从高速模式切换到休眠模式，唤醒后返回到高速模式；

## 5.2 低速模式

低速模式为系统低速时钟正常工作模式。系统时钟源由内部低速 RC 振荡器提供。低速模式由 OSCM 寄存器的 CLKS 位控制。当 CLKS=0 时，系统为高速模式；当 CLKS=1 时，系统进入低速模式。切换进入低速模式后，不能自动禁止高速振荡器，必须通过 HOFF 位来禁止以减少功耗。低速模式下，系统速率被固定为 Flirc/4（Flirc 为内部低速 RC 振荡器频率）。



- ◇ 程序被执行，所有的功能都可控制；
- ◇ 系统速率为低速（Flirc/4）；
- ◇ 内部低速 RC 振荡器正常工作，高速振荡器由 HOFF=1 控制。低速模式下，强烈建议停止高速振荡器；
- ◇ 从低速模式切换到休眠模式，唤醒后返回到高速模式；
- ◇ 高速模式可以切换进入低速模式；
- ◇ 从低速模式切换到空闲模式，唤醒后返回到低速模式；

### 5.3 休眠模式

休眠模式是系统的理想状态，不执行程序，振荡器也停止工作。整个芯片的功耗低于 1uA。休眠模式可以由 P0 的电平变换触发唤醒。从任何工作模式进入休眠模式，被唤醒后都返回到高速模式。由 OSCM 寄存器的 CPUM 位控制是否进入休眠模式，当 CPUM=01，系统进入休眠模式。当系统从休眠模式被唤醒后，CPUM 被自动设置为 00。

- ◇ 程序停止执行，所有的功能被禁止；
- ◇ 所有的振荡器，包括外部高速振荡器、内部高速振荡器和内部低速振荡器都停止工作；
- ◇ 功耗低于 1uA；
- ◇ 系统从休眠模式被唤醒后进入高速模式；
- ◇ 休眠模式的唤醒源为 P0 电平变换触发或 T2(RTC 模式下时钟源选择外部晶体)溢出唤醒；

### 5.4 空闲模式（绿色模式）

空闲模式又称为绿色模式，是另外一种理想状态。在休眠模式下，所有的功能和硬件设备都被禁止，但在空闲模式下，系统时钟保持工作，空闲模式下的功耗大于休眠模式下的功耗。空闲模式下，不执行程序，但具有唤醒功能的定时器仍正常工作，定时器的时钟源为仍在工作的系统时钟。空闲模式下有 2 种方式可以将系统唤醒：1、P0 电平变换触发；2、具有唤醒功能的定时器溢出，这样，用户可以给定时器设定固定的周期，系统就在溢出时被唤醒。由 OSCM 寄存器 CPUM 位决定是否进入空闲模式，当 CPUM=10，系统进入空闲模式。当系统从空闲模式下被唤醒后，CPUM 被自动设置为 00。

- ◇ 程序停止执行，所有的功能被禁止；
- ◇ 具有唤醒功能的定时器正常工作；
- ◇ 作为系统时钟源的振荡器正常工作，其他的振荡器工作状态取决于系统工作模式的配置；
- ◇ 由高速模式切换到空闲模式，被唤醒后返回到高速模式；
- ◇ 由低速模式切换到空闲模式，被唤醒后返回到低速模式；
- ◇ 空闲模式下的唤醒方式为 P0 电平变换触发唤醒和指定的定时器（T0、T2）溢出；
- ◇ 空闲模式下 PWM 和 BUZ 功能仍然有效，但是定时器溢出时不能唤醒系统；



## 6 复位

### 6.1 复位条件

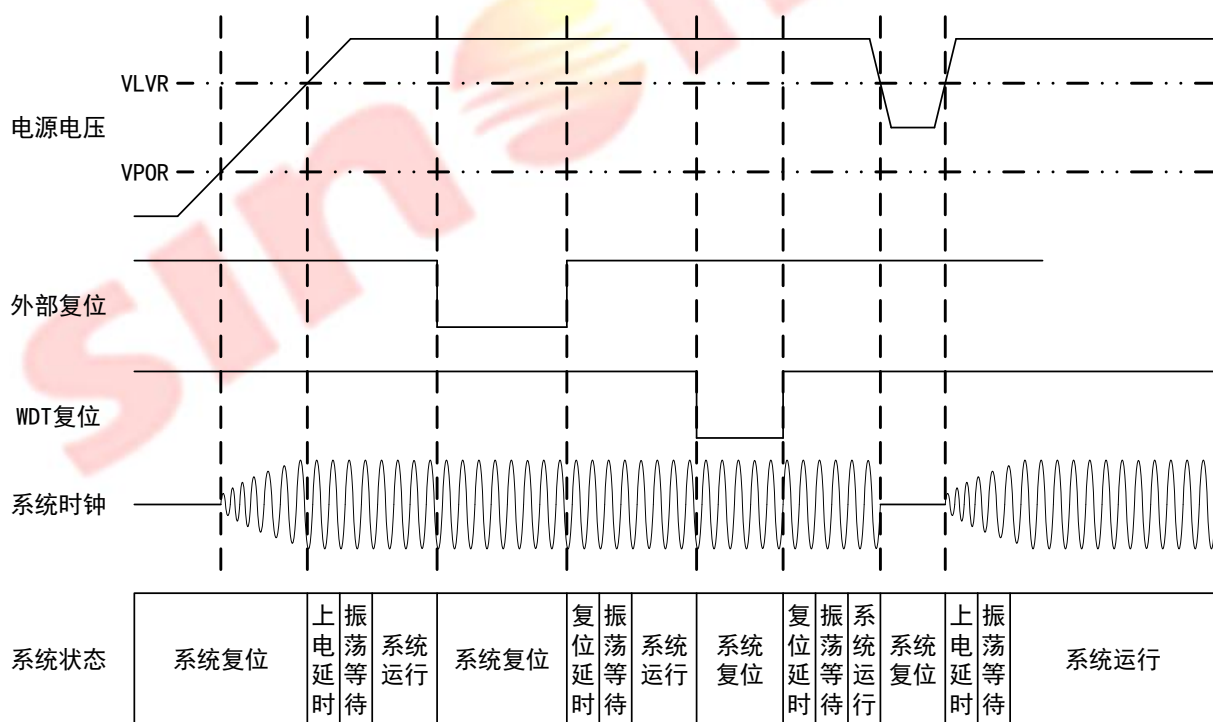
芯片共有如下几种复位方式：

- ✧ 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 复位不会关闭系统主时钟振荡器，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：



## 6.2 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。

## 6.3 外部复位

芯片的外部复位功能可通过配置字 **MCLRE** 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。

## 6.4 低电压复位

芯片的低电压复位电压可通过配置字 **VLVRS** 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。

## 6.5 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。

## 7 I/O 端口

### 7.1 通用 I/O 功能

芯片的输入/输出端口包括一组 5 位端口 P0，一组 6 位端口 P4，和一组 3 位端口 P5。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、或 ADC 模拟输入等复用功能。

#### 端口数据寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP0	-	-	-	P04D	P03D	P02D	P01D	P00D
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	-	-	-	X	X	X	X	X

BIT[4:0] P0nD – P0n 端口数据位 (n=4-0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP4	-	-	P45D	P44D	P43D	P42D	P41D	P40D
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	X	X	X	X	X	X

BIT[5:0] P4nD – P4n 端口数据位 (n=5-0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP5	-	-	-	P54D	P53D	P52D	-	-
R/W	-	-	-	R/W	R/W	R/W	-	-
初始值	-	-	-	X	X	X	-	-

BIT[4:2] P5nD – P5n 端口数据位 (n=4-2)

#### 端口方向寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP0	-	-	-	P04OE	P03OE	P02OE	P01OE	P00OE
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	-	-	-	0	0	0	0	0

BIT[4:0] P0nOE – P0n 端口输出使能位 (n=4-0)

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
OEP4	-	-	P45OE	P44OE	P43OE	P42OE	P41OE	P40OE
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] **P4nOE** – P4n 端口输出使能位 (n=5-0)
- 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
<b>OEP5</b>	-	-	-	P54OE	P53OE	P52OE	-	-
<b>R/W</b>	-	-	-	R/W	R/W	R/W	-	-
<b>初始值</b>	-	-	-	0	0	0	-	-

- BIT[4:2] **P5nOE** – P5n 端口输出使能位 (n=4-2)
- 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;

## 7.2 内部上拉电阻

所有端口均有内部上拉电阻, 且均有单独的寄存器位控制其上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
<b>PUP0</b>	-	-	-	P04PU	P03PU	P02PU	P01PU	P00PU
<b>R/W</b>	-	-	-	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	-	-	-	0	0	0	0	0

- BIT[4:0] **P0nPU** – P0n 端口上拉电阻控制位 (n=4-0)
- 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
<b>PUP4</b>	-	-	P45PU	P44PU	P43PU	P42PU	P41PU	P40PU
<b>R/W</b>	-	-	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	-	-	0	0	0	0	0	0

- BIT[5:0] **P4nPU** – P4n 端口上拉电阻控制位 (n=5-0)
- 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
<b>PUP5</b>	-	-	-	P54PU	P53PU	P52PU	-	-
<b>R/W</b>	-	-	-	R/W	R/W	R/W	-	-
<b>初始值</b>	-	-	-	0	0	0	-	-

- BIT[4:2] **P5nPU** – P5n 端口上拉电阻控制位 (n=4-2)
- 0: 端口内部上拉电阻无效;
- 1: 端口内部上拉电阻有效;

### 7.3 端口模式控制

P4 所有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
<b>ANSEL</b>			P45ANS	P44ANS	P43ANS	P42ANS	P41ANS	P40ANS
<b>R/W</b>			R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>			0	0	0	0	0	0

BIT[5:0] **P4nANS** – P4n 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；

## 8 定时器 TIMER

### 8.1 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 (24KHz @3V, 32KHz @5V), WDT 计数器溢出将复位芯片。可由用户配置字 WDTC 位设置 WDT 工作模式。

若选择始终开启, 则 WDT 在任意模式下都将一直工作, 溢出将复位芯片。若选择低功耗模式下关闭, 则 WDT 在休眠/空闲模式下自动停止、在系统被唤醒后恢复工作。

WDT 控制寄存器 WDTR 写 5AH 将清零 WDT 计数器。

WDT 溢出时间 = 8192 / 内部低频 RC 振荡器频率 (FLIRC)。

**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。**

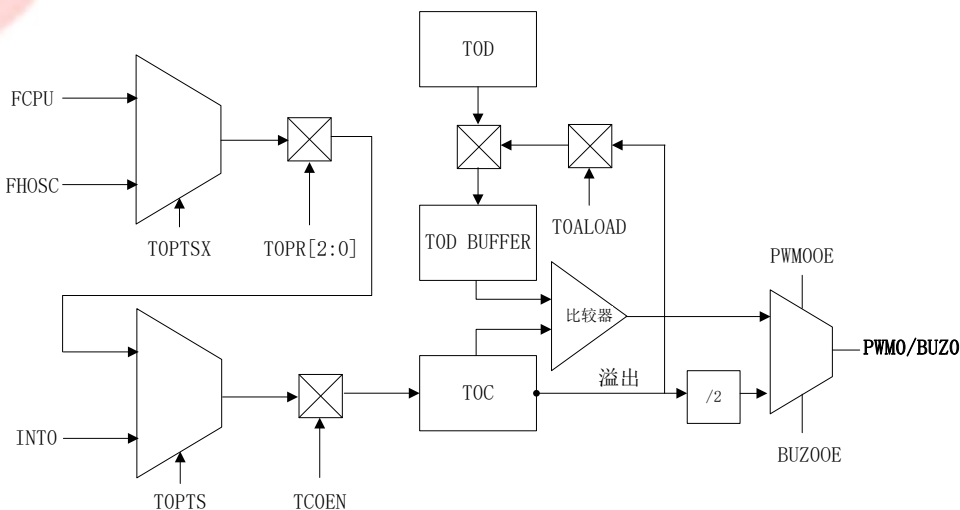
#### WDT 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTR	WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0
R/W	W	W	W	W	W	W	W	W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] WDTR[7:0] – WDT 控制寄存器, 写 5AH 将清零 WDT 计数器

### 8.2 定时器 T0

T0 具有定时器、事件计数器、PWM 和 BUZ 功能。



T0PTS 可选择 T0 的时钟源, T0PR 可选择 T0 的预分频比, 所选中的时钟源通过预分频器后产生 T0C 的时钟, 当 T0C 递增到 FFH 时, 产生 T0 溢出中断请求标志, T0IF 置 1。由 T0CR、T0C、T0D 寄存器控制 T0 的溢出中断间隔时间。T0 通过置位 T0ALOAD 实现自动重装功能, 当 T0 溢出时, T0D 的值自动装入 T0C, T0D 内置双重缓存器。

T0 可实现事件计数器功能, 将 T0 时钟源由系统时钟更改为外部时钟信号, T0 将对外部管脚 INTO 下降沿进行计数。

T0 可实现占空比可编程控制的 PWM 功能, 由 T0PR, T0D 寄存器和 T0CR 寄存器的 T0ALOAD 和 BUZ0OE 位控制占空比/周期。当 PWM0OE=1 时, 将输出 PWM 波形, 当 T0C 计数到与 T0D 相等时, PWM0 输出清 0; 当 T0C 计数溢出时, PWM0 输出置 1。

T0PR、T0ALOAD 和 BUZ0OE 位控制 PWM0 的周期, T0D 控制 PWM0 的占空比 (脉冲高电平的长度)。PWM0 内置 4 种可编程控制的分辨率 (1/256、1/64、1/32、1/16), 在 PWM0OE=1 时由 T0ALOAD 和 BUZ0OE 位控制。

PWM0OE	T0ALOAD	BUZ0OE	PWM0 分辨率	T0D 有效值
1	0	0	256	00H~FFH
1	0	1	64	00H~3FH
1	1	0	32	00H~1FH
1	1	1	16	00H~0FH

T0 内置 BUZ 功能, BUZ0 周期为 T0 溢出间隔时间的 2 倍。当 PWM0OE=0 且 BUZ0OE=1 时, 输出 BUZ0 信号, BUZ0 信号的输出频率为 T0 溢出频率的 2 分频。

T0 内置空闲模式唤醒功能, 由 T0GE 控制, 当芯片进入空闲模式后, 如果 T0GE=1, 当 T0 溢出时将把芯片从空闲模式唤醒。

#### 定时器控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TXCR	-	-	-	T2PTSX	T1PTSX	T0PTSX	T0GE	-
R/W	-	-	-	R/W	R/W	R/W	R/W	-
初始值	-	-	-	0	0	0	0	-

BIT[4] **T2PTSX** – T2 内部时钟选择位  
0: T2 内部时钟来自 F<sub>CPU</sub>;  
1: T2 内部时钟来自 F<sub>HOSC</sub>;

BIT[3] **T1PTSX** – T1 内部时钟选择位  
0: T1 内部时钟来自 F<sub>CPU</sub>;  
1: T1 内部时钟来自 F<sub>HOSC</sub>;

BIT[2] **T0PTSX** – T0 内部时钟选择位  
0: T0 内部时钟来自 F<sub>CPU</sub>;  
1: T0 内部时钟来自 F<sub>HOSC</sub>;



BIT[1] **T0GE** – T0 空闲模式唤醒使能位  
0: 禁止 T0 的唤醒功能;  
1: 使能 T0 的唤醒功能;

### 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
<b>T0CR</b>	TC0EN	T0PR2	T0PR1	T0PR0	T0PTS	T0ALOAD	BUZ0OE	PWM0OE
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7] **TC0EN** – 定时器 T0 使能位  
0: 关闭定时器 T0;  
1: 开启定时器 T0;

BIT[6:4] **T0PR[2:0]** – T0 预分频比选择位

T0PR[2:0]	T0C 时钟	
	T0PTSX=0	T0PTSX=1
000	FCPU/256	FHOSC/128
001	FCPU/128	FHOSC/64
010	FCPU/64	FHOSC/32
011	FCPU/32	FHOSC/16
100	FCPU/16	FHOSC/8
101	FCPU/8	FHOSC/4
110	FCPU/4	FHOSC/2
111	FCPU/2	FHOSC/1

BIT[3] **T0PTS** – T0 时钟源选择位  
0: 内部时钟 (由 TXCR 中 T0PTSX 位选择);  
1: 外部时钟 (由 INT0 输入);

BIT[2] **T0ALOAD** – T0 自动重载使能位 (仅 PWM0OE=0 时有效)  
0: 禁止 T0 自动重载;  
1: 使能 T0 自动重载;

BIT[1] **BUZ0OE** – BUZ0 端口输出使能位 (仅 PWM0OE=0 时有效)  
0: 禁止端口输出 BUZ 波形;  
1: 允许端口输出 BUZ 波形;

BIT[0] **PWM0OE** – PWM0 使能位及端口输出控制位  
0: 关闭 PWM0 功能, 并禁止端口输出 PWM 波形;  
1: 使能 PWM0 功能, 并允许端口输出 PWM 波形;



### 定时器 T0 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T0C</b>	T0C7	T0C6	T0C5	T0C4	T0C3	T0C2	T0C1	T0C0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **T0C[7:0]** – T0 计数器，为可读写的递增计数器

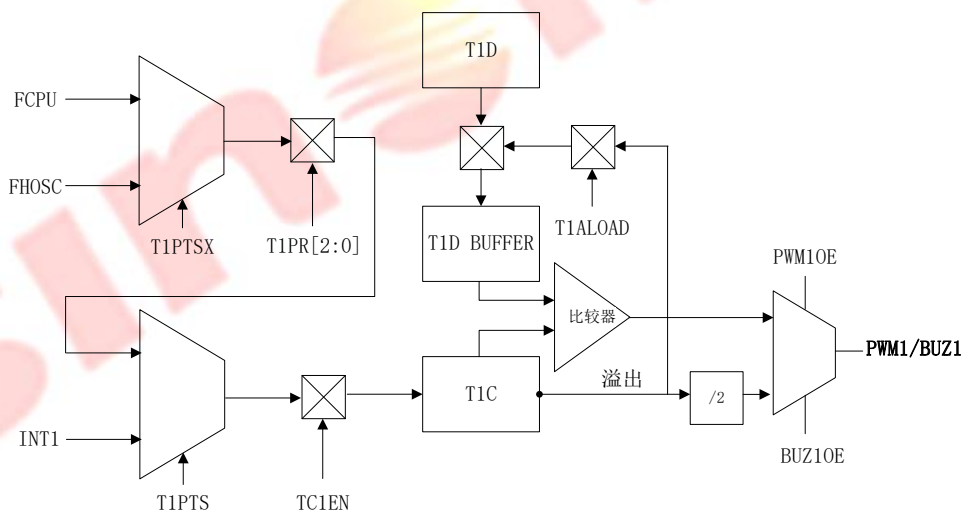
### 定时器 T0 重载寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T0D</b>	T0D7	T0D6	T0D5	T0D4	T0D3	T0D2	T0D1	T0D0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **T0D[7:0]** – T0 重载寄存器，用于设置 T0 重载值或 PWM0 的占空比

## 8.3 定时器 T1

T1 具有定时器、事件计数器、PWM 和 BUZ 功能。



T1PTS 可选择 T1 的时钟源, T1PR 可选择 T1 的预分频比, 所选中的时钟源通过预分频器后产生 T1C 的时钟, 当 T1C 递增到 FFH 时, 产生 T1 溢出中断请求标志, T1IF 置 1。由 T1CR、T1C、T1D 寄存器控制 T1 的溢出中断间隔时间。T1 通过置位 T1ALOAD 实现自动重装功能, 当 T1 溢出时, T1D 的值自动装入 T1C, T1D 内置双重缓存器。

T1 可实现事件计数器功能, 将 T1 时钟源由系统时钟更改为外部时钟信号, T1 将对外部管脚 INT1 下降沿进行计数。

T1 可实现占空比可编程控制的 PWM 功能，由 T1PR，T1D 寄存器和 T1CR 寄存器的 T1ALOAD 和 BUZ1OE 位控制占空比/周期。当 PWM1OE=1 时，将输出 PWM 波形，当 T1C 计数到与 T1D 相等时，PWM1 输出清 0；当 T1C 计数溢出时，PWM1 输出置 1。

T1PR、T1ALOAD 和 BUZ1OE 位控制 PWM1 的周期，T1D 控制 PWM1 的占空比（脉冲高电平的长度）。PWM1 内置 4 种可编程控制的分辨率（1/256、1/64、1/32、1/16），在 PWM1OE=1 时由 T1ALOAD 和 BUZ1OE 位控制。

PWM1OE	T1ALOAD	BUZ1OE	PWM1 分辨率	T1D 有效值
1	0	0	256	00H~FFH
1	0	1	64	00H~3FH
1	1	0	32	00H~1FH
1	1	1	16	00H~0FH

T1 内置 BUZ 功能，BUZ1 周期为 T1 溢出间隔时间的 2 倍。当 PWM1OE=0 且 BUZ1OE=1 时，输出 BUZ1 信号，BUZ1 信号的输出频率为 T1 溢出频率的 2 分频。

当芯片进入空闲模式后，如果 T1 时钟存在则 T1 正常工作，但溢出无法唤醒芯片。

#### 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
<b>T1CR</b>	TC1EN	T1PR2	T1PR1	T1PR0	T1PTS	T1ALOAD	BUZ1OE	PWM1OE
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7] **TC1EN** – 定时器 T1 使能位

- 0: 关闭定时器 T1;
- 1: 开启定时器 T1;

BIT[6:4] **T1PR[2:0]** – T1 预分频比选择位

T1PR[2:0]	T1C 时钟	
	T1PTSX=0	T1PTSX=1
000	FCPU/256	FHOSC/128
001	FCPU/128	FHOSC/64
010	FCPU/64	FHOSC/32
011	FCPU/32	FHOSC/16
100	FCPU/16	FHOSC/8
101	FCPU/8	FHOSC/4
110	FCPU/4	FHOSC/2
111	FCPU/2	FHOSC/1

BIT[3] **T1PTS** – T1 时钟源选择位

- 0: 内部时钟（由 TXCR 中 T1PTSX 位选择）;
- 1: 外部时钟（由 INT1 输入）;

BIT[2] **T1ALOAD** – T1 自动重载使能位（仅 PWM1OE=0 时有效）  
0: 禁止 T1 自动重载;  
1: 使能 T1 自动重载;

BIT[1] **BUZ1OE** – BUZ1 端口输出使能位（仅 PWM1OE=0 时有效）  
0: 禁止端口输出 BUZ 波形;  
1: 允许端口输出 BUZ 波形;

BIT[0] **PWM1OE** – PWM1 使能位及端口输出控制位  
0: 关闭 PWM1 功能，并禁止端口输出 PWM 波形;  
1: 使能 PWM1 功能，并允许端口输出 PWM 波形;

### 定时器 T1 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T1C</b>	T1C7	T1C6	T1C5	T1C4	T1C3	T1C2	T1C1	T1C0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **T1C[7:0]** – T1 计数器，为可读写的递增计数器

### 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
<b>T1D</b>	T1D7	T1D6	T1D5	T1D4	T1D3	T1D2	T1D1	T1D0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **T1D[7:0]** – T1 重载寄存器，用于设置 T1 重载值或 PWM1 的占空比

## 8.4 定时器 T2

T2 具有定时器、RTC 计数器、PWM 和 BUZ 功能。

T2PTS 可选择 T2 的时钟源，T2PR 可选择 T2 的预分频比，所选中的时钟源通过预分频器后产生 T2C 的时钟，当 T2C 递增到 FFH 时，产生 T2 溢出中断请求标志，T2IF 置 1。由 T2CR、T2C、T2D 寄存器控制 T2 的溢出中断间隔时间。T2 通过置位 T2ALOAD 实现自动重装功能，当 T2 溢出时，T2D 的值自动装入 T2C，T2D 内置双重缓存器。

T2 可实现 RTC 计数器功能，将 T2 时钟源由系统时钟更改为外部 RTC 时钟 F<sub>RTC</sub>，T2 将对外部 32K 晶体输入信号的下降沿进行计数。

T2 可实现占空比可编程控制的 PWM 功能，由 T2PR，T2D 寄存器和 T2CR 寄存器的 T2ALOAD 和 BUZ2OE 位控制占空比/周期。当 PWM2OE=1 时，将输出 PWM 波形，当 T2C 计数到与 T2D 相等时，

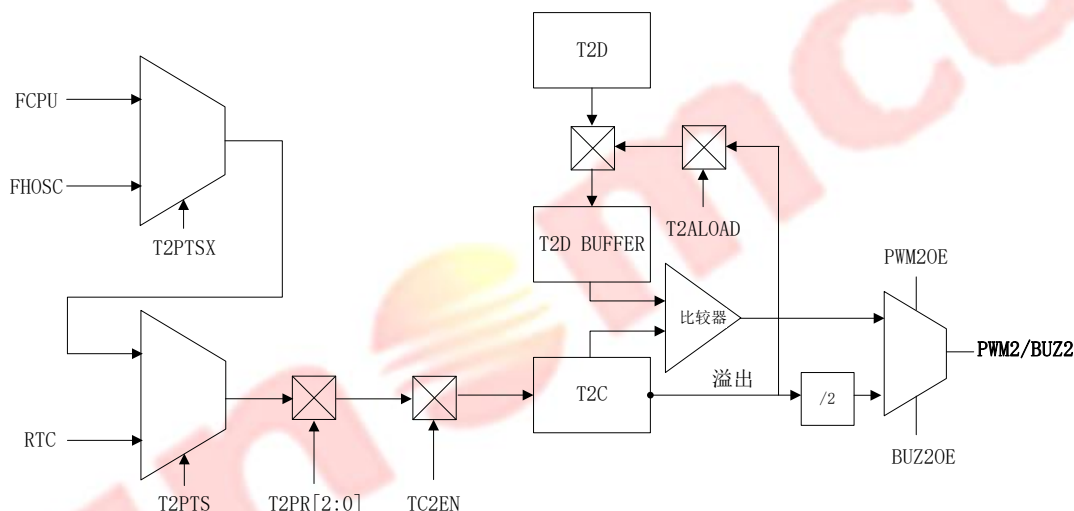
PWM2 输出清 0；当 T2C 计数溢出时，PWM2 输出置 1。

T2PR、T2ALOAD 和 BUZ2OE 位控制 PWM2 的周期，T2D 控制 PWM2 的占空比（脉冲高电平的长度）。PWM2 内置 4 种可编程控制的分辨率（1/256、1/64、1/32、1/16），在 PWM2OE=1 时由 T2ALOAD 和 BUZ2OE 位控制。

PWM2OE	T2ALOAD	BUZ2OE	PWM2 分辨率	T2D 有效值
1	0	0	256	00H~FFH
1	0	1	64	00H~3FH
1	1	0	32	00H~1FH
1	1	1	16	00H~0FH

T2 内置 BUZ 功能，BUZ2 周期为 T2 溢出间隔时间的 2 倍。当 PWM2OE=0 且 BUZ2OE=1 时，输出 BUZ2 信号，BUZ2 信号的输出频率为 T2 溢出频率的 2 分频。

T2 支持空闲模式唤醒功能，无需控制位，当芯片进入空闲模式后，如果 T2 溢出时将把芯片从空闲模式唤醒。



### 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
<b>T2CR</b>	TC2EN	T2PR2	T2PR1	T2PR0	T2PTS	T2ALOAD	BUZ2OE	PWM2OE
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7] **TC2EN** – 定时器 T2 使能位

0: 关闭定时器 T2;

1: 开启定时器 T2;

BIT[6:4] **T2PR[2:0]** – T2 预分频比选择位

T2PR[2:0]	T2C 时钟		
	T2PTS=1	T2PTS=0	
		T2PTSX=0	T1PTSX=1
000	FRTC/128	FCPU/256	FHOSC/128

001	FRTC/64	FCPU/128	FHOSC/64
010	FRTC/32	FCPU/64	FHOSC/32
011	FRTC/16	FCPU/32	FHOSC/16
100	FRTC/8	FCPU/16	FHOSC/8
101	FRTC/4	FCPU/8	FHOSC/4
110	FRTC/2	FCPU/4	FHOSC/2
111	FRTC/1	FCPU/2	FHOSC/1

BIT[3] **T2PTS** – T2 时钟源选择位  
 0: 内部时钟（由 TXCR 中 T2PTSX 位选择）；  
 1: 外部 32K 晶体（需配置字 FOSCS 选择 RTC 模式）；

BIT[2] **T2ALOAD** – T2 自动重载使能位（仅 PWM2OE=0 时有效）  
 0: 禁止 T2 自动重载；  
 1: 使能 T2 自动重载；

BIT[1] **BUZ2OE** – BUZ2 端口输出使能位（仅 PWM2OE=0 时有效）  
 0: 禁止端口输出 BUZ 波形；  
 1: 允许端口输出 BUZ 波形；

BIT[0] **PWM2OE** – PWM2 使能位及端口输出控制位  
 0: 关闭 PWM2 功能，并禁止端口输出 PWM 波形；  
 1: 使能 PWM2 功能，并允许端口输出 PWM 波形；

#### 定时器 T2 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T2C</b>	T2C7	T2C6	T2C5	T2C4	T2C3	T2C2	T2C1	T2C0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **T2C[7:0]** – T2 计数器，为可读写的递增计数器

#### 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
<b>T2D</b>	T2D7	T2D6	T2D5	T2D4	T2D3	T2D2	T2D1	T2D0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **T2D[7:0]** – T2 重载寄存器，用于设置 T2 重载值或 PWM2 的占空比

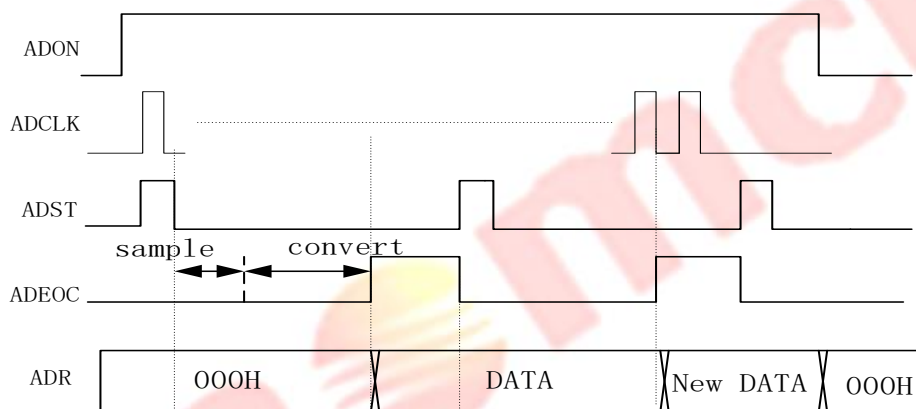
## 9 模数转换器 ADC

### 9.1 ADC 概述

6+1 通道 12 位模数转换器，可通过 ADON 使能模数转换模块，ADCHS 选择转换的模拟通道，可通过 ADCKS 位选择 ADC 的转换速率以决定 ADC 的转换时间，ADCKS 选择 AD 转换速度，ADST 位为 AD 转换启动控制位，ADEOC 位为 AD 转换结束标志位。

当 ADEOC 标志为 1 时，对 ADST 写 1 将启动模数转换，转换结束后，转换结果被放在 ADRH 和 ADRL 中，ADEOC 将自动置 1，同时中断标志 ADIF 置 1，若 GIE 和 ADIE 使能，将产生 AD 中断。

ADC 转换时序如下图所示：



注：

- 1、AD 转换过程中或者 ADON 未使能时，ADRH/ADRL 中的数据未知，应在 AD 转换结束且 ADON 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  $V_{DD} > (V_{IR} + 0.5V)$ ，否则  $V_{IR}$  将随之下降；
- 3、使能 ADC 模块、切换参考电压等操作后，需待电路稳定（时间  $> 200\mu s$ ）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 操作步骤

模数转换操作步骤：

- (1) 设置 OEP4 将相应的端口设置为输入端口，同时关闭上拉电阻；
- (2) 设置 ANSEL 关闭相应端口的数字 I/O 功能；
- (3) 设置 ADCKS 选取适当的 AD 转换时钟；
- (4) 设置 VREF 选择参考电压；
- (5) 使能 ADON；
- (6) 置位 GCHS 和设置 ADCHS 选取 AD 转换通道；
- (7) ADST 写 1 启动 AD 转换；
- (8) 等待 ADEOC 置 1（或利用 AD 中断）；
- (9) 读取 ADC 转换结果（ADRH、ADRL）；
- (10) 重复执行（6）~（8），对不同的通道进行转换或对同一通道进行多次转换；

## 9.3 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
<b>ADCR</b>	ADON	ADST	ADEOC	GCHS	-	ADCHS2	ADCHS1	ADCHS0
<b>R/W</b>	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	-	0	0	0

BIT[7] **ADON** – ADC 功能使能位

- 0: 关闭 ADC 功能；  
1: 使能 ADC 功能；

BIT[6] **ADST** – ADC 启动位

- 0: AD 转换结束后自动清 0；  
1: 写 1 启动 AD 转换；

BIT[5] **ADEOC** – ADC 转换结束标志位

- 0: AD 转换过程中；  
1: AD 转换结束；

BIT[4] **GCHS** – ADC 转换输入通道控制位

- 0: 关闭输入通道；  
1: 开启输入通道；

BIT[2:0] **ADCHS[2:0]** – ADC 模拟通道选择位

ADCHS[2:0]	ADC 模拟通道选择
000	AIN0



001	AIN1
010	AIN2
011	AIN3
100	AIN4
101	VDD/4
110	AIN6

### 参考电压控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>VREF</b>	VREFS	-	-	-	-	-	VRS1	VRS0
<b>R/W</b>	R/W	-	-	-	-	-	R/W	R/W
<b>初始值</b>	0	-	-	-	-	-	0	0

- BIT[7] **VREFS** – ADC 参考电压选择位  
 0: ADC 参考电压选择 VDD、或内部参考电压 VIR;  
 1: ADC 参考电压选择端口 VREFH 的外部输入电压;

- BIT[1:0] **VRS[1:0]** – ADC 参考电压选择位（仅 VREFS=0 时有效）

<b>ADVRS[1:0]</b>	<b>ADC 参考电压选择</b>
00	内部 2V
01	内部 3V
10	内部 4V
11	VDD

### ADC 转换结果寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>ADRH</b>	ADR11	ADR10	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4
<b>R/W</b>	R	R	R	R	R	R	R	R
<b>初始值</b>	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
<b>ADRL</b>	-	ADCKS1	ADLEN	ADCKS0	ADR3	ADR2	ADR1	ADR0
<b>R/W</b>	-	R/W	R/W	R/W	R	R	R	R
<b>初始值</b>	-	0	1	0	X	X	X	X

- BIT[6,4] **ADCKS[1:0]** – ADC 转换时钟选择位

<b>ADCKS[1:0]</b>	<b>ADC 转换时钟 F<sub>ADC</sub></b>
00	F <sub>CPU</sub> /16
01	F <sub>CPU</sub> /8
10	F <sub>CPU</sub> /1
11	F <sub>CPU</sub> /2

**注：**ADC 转换时钟频率需设置为≤500KHz。



- BIT[5]      **ADLEN** –ADC 转换结果选择位
- 0: 转换结果为 8 位, ADC 数据存放于 ADRH 寄存器中;
- 1: 转换结果为 12 位, ADC 数据存放于 ADRH 和 ADRL 寄存器中;

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

## 9.4 ADC 操作说明和注意事项

### ADC 信号格式

ADC 采样电压范围为参考电压高/低电平之间, ADC 参考低电压为 GND, 高电压包括 VDD/4V/3V/2V, 外部参考电压由 VREFH 引脚提供 (由 VREFS 控制)。VREFS=0 时, ADC 参考电压选择内部参考源; VREFS=1 时, ADC 参考电压选择外部参考源 (VREFH)。ADC 参考电压的范围为: (ADC 参考高电压-ADC 参考低电压)  $\geq$  2V, ADC 参考低电压为 GND=0V, 故 ADC 参考高电压范围为 2V~VDD, 外部参考电压需在此范围之内。

- ✧ ADC 内部参考低电压 = 0V;
- ✧ ADC 内部端口低电压 = VDD/4V/3V/2V (VREFS = 0);
- ✧ ADC 外部参考电压 = 2V~VDD (VREFS = 1);

ADC 采样输入信号电压必须在 ADC 参考低电压和 ADC 参考高电压之间, 若 ADC 输入信号的电压不在此范围内, 则 ADC 的转换结果会出错 (满量程或者为 0)。

### AD 转换时间

ADC 转换时间是指从 ADST=1 (开始 ADC) 到 ADEOC=1 (ADC 结束) 所用的时间, 由 ADC 分辨率和 ADC 时钟 Rate 控制, 12 位 ADC 的转换时间为  $1 / (\text{ADC 时钟} / 4) \times (\text{SAMPTS} + 12)$ ; 8 位 ADC 的转换时间为  $1 / (\text{ADC 时钟} / 4) \times (\text{SAMPTS} + 8)$ 。ADC 的时钟源为 Fcpu, 包括 Fcpu/1, Fcpu/2, Fcpu/8, Fcpu/16, 由 ADCKS[1:0] 位控制。

ADC 的转换时间会影响 ADC 的性能, 如果输入高 Rate 的模拟信号, 必须要选择一个高 Rate 的 ADC 转换 Rate。如果 ADC 的转换时间比模拟信号的转换 Rate 慢, 则 ADC 的结果出错。故选择合适的 ADC 时钟 Rat 和 ADC 分辨率才能得到合适的 ADC 转换 Rate。

AD 转换精度与 AD 转换时钟有关系, 时钟越低, 精度越高。为了达到高精度, 建议  $F_{\text{ADC}} \leq 500\text{KHz}$ 。对于  $F_{\text{ADC}}$  大于 500KHz 的情况, 必须对同一个通道连续转换多次, 舍弃前 3 次数据, 取第 4 次之后的数据。

AD 转换过程分为 SAMPLE 时间和 CONVERT 时间, SAMPLE 时间由配置字 SAMPTS 位控制, SAMPTS 选择时间越长精度越高, 建议选择最大的 14 个 ADCLK。

## AD 引脚配置

ADC 输入引脚与 P4 口共用，ADC 输入通道的选择由 ADCHS[2:0]控制，ADCCHS[2:0]=000 时选择 AIN0，ADCCHS[2:0]=001 时选择 AIN1……同一时间设置 P4 口的一个引脚作为 ADC 的输入引脚，该引脚必须设置为输入引脚，禁止内部上拉，并首先由程序使能 ANSEL 寄存器。通过 ADCHS[2:0]选择好 ADC 输入通道后，GCHS 置 1 以使能 ADC 功能。

- ✧ ADC 输入引脚为 GPIO 引脚时必须设为输入模式；
- ✧ 必须禁止 ADC 输入引脚的内部上拉电阻；
- ✧ ADC 输入通道的 ANSEL 位必须置 1；

VREFS = 1 时，P4.0/AIN0 为 ADC 外部参考源的输入引脚，此时，P4.0 必须设为输入模式，并禁止其上拉电阻。

- ✧ ADC 外部参考源输入引脚为 GPIO 引脚时必须设为输入模式；
- ✧ 必须禁止 ADC 外部参考源输入引脚的内部上拉电阻；

ADC 输入引脚与普通 I/O 引脚共用。当输入一个模拟信号到 CMOS 结构端口时，尤其当模拟信号为  $V_{DD}/2$  时，可能产生额外的漏电流。当 P4 输入多个模拟信号时，也会产生额外的漏电流。休眠模式下，上述漏电流会严重影响到系统的整体功耗。ANSEL 为 P4 口的设置寄存器，将 ANSEL[5:0]置 1，其对应的 P4 引脚将被设为纯模拟信号输入引脚，从而避免上述漏电流的产生。

## 10 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，中断服务程序应先通过 PUSH 指令保存累加器 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，应先通过 POP 指令恢复累加器 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。

### 10.1 外部中断

芯片有 2 路外部中断源 INTn（n=0-1），INT0 可选择上升沿、下降沿或电平变化等触发方式，INT1 固定为下降沿触发。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。

#### 外部中断控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>PEDGE</b>	-	-	-	MINT01	MINT00	-	-	-
<b>R/W</b>	-	-	-	R/W	R/W	-	-	-
<b>初始值</b>	-	-	-	1	0	-	-	-

BIT[4:3] MINT0[1:0] – 外部中断 INT0 触发方式选择位

MINT0[1:0]	INT0 触发方式
00	保留
01	上升沿触发
10	下降沿触发
11	电平变化触发

### 10.2 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。

## 10.3 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。

## 10.4 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
<b>INTE</b>	ADIE	T1IE	T0IE	T2IE			INT1IE	INT0IE
<b>R/W</b>	R/W	R/W	R/W	R/W			R/W	R/W
<b>初始值</b>	0	0	0	0			0	0

BIT[7] **ADIE** – ADC 中断使能位

0: 屏蔽 ADC 中断;

1: 使能 ADC 中断;

BIT[6] **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[5] **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[4] **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[1] **INT1IE** – INT1 中断使能位

0: 屏蔽 INT1 中断;

1: 使能 INT1 中断;

BIT[0] **INT0IE** – INT0 中断使能位

0: 屏蔽 INT0 中断;

1: 使能 INT0 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
<b>INTF</b>	ADIF	T1IF	T0IF	T2IF			INT1IF	INT0IF
<b>R/W</b>	R/W	R/W	R/W	R/W			R/W	R/W
<b>初始值</b>	0	0	0	0			0	0

---

BIT[7]	<b>ADIF</b> – 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[6]	<b>T1IF</b> – 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[5]	<b>T0IF</b> – 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[4]	<b>T2IF</b> – 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[1]	<b>INT1IF</b> – 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[0]	<b>INT0IF</b> – 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;

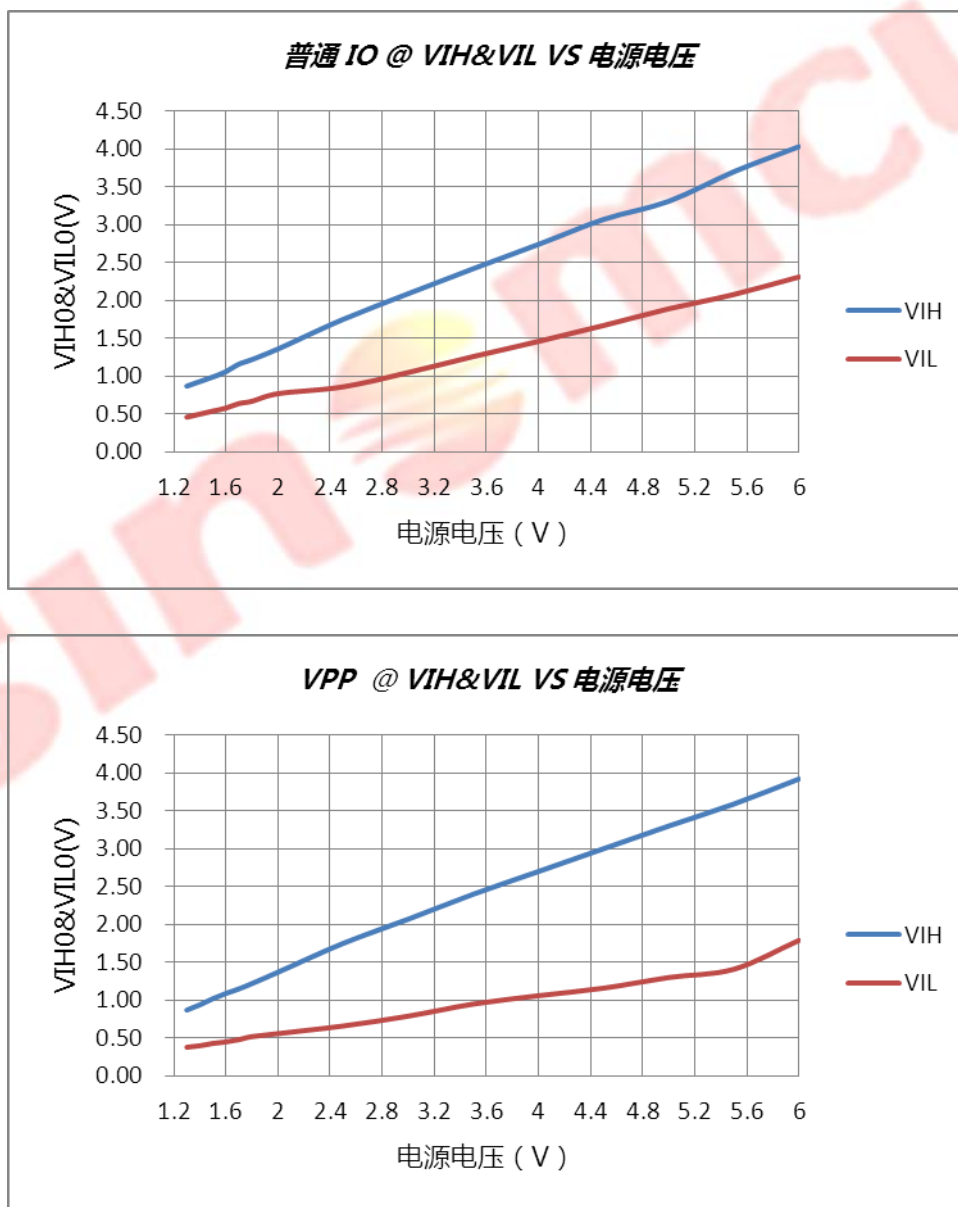
## 11 特性曲线

注:

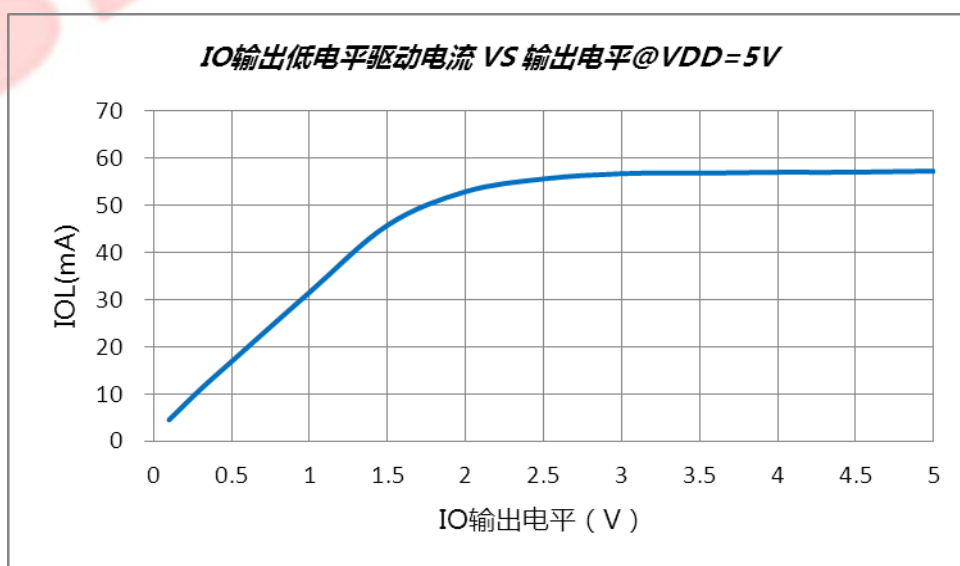
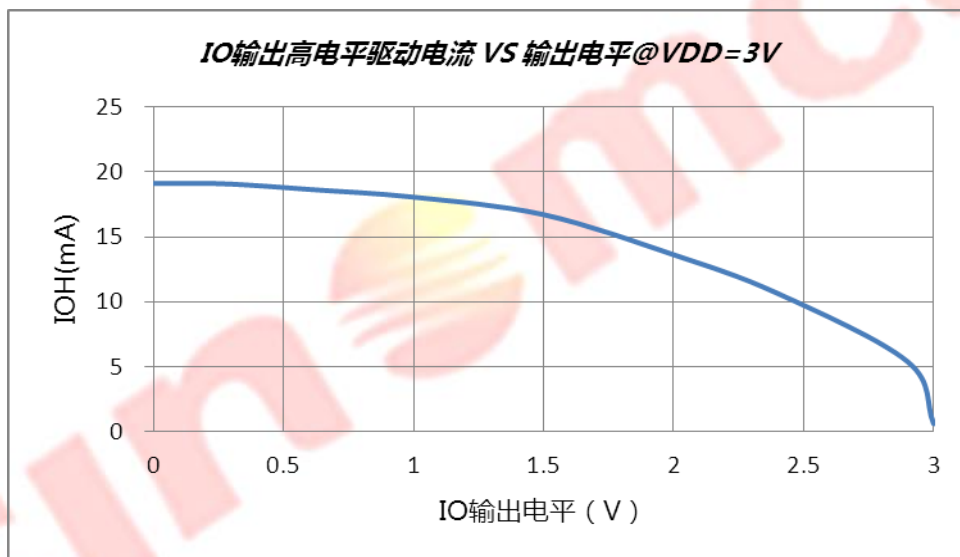
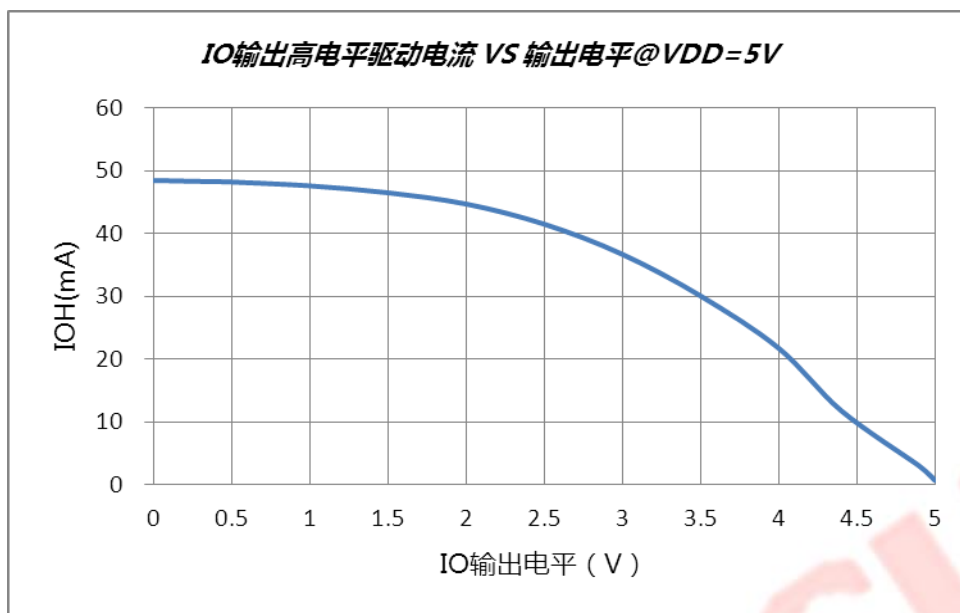
- 1、特性曲线图中数据均来自抽样实测, 仅作为应用参考, 部分数据因生产工艺偏差, 可能与实际芯片不符; 为保证芯片能正常工作, 请确保其工作条件符合电气特性参数说明;
- 2、若图文中无特别说明, 则电压特性曲线的温度条件为  $T=25^{\circ}\text{C}$ , 温度特性曲线的电压条件为  $V_{DD}=5\text{V}$ ;

### 11.1 I/O 特性

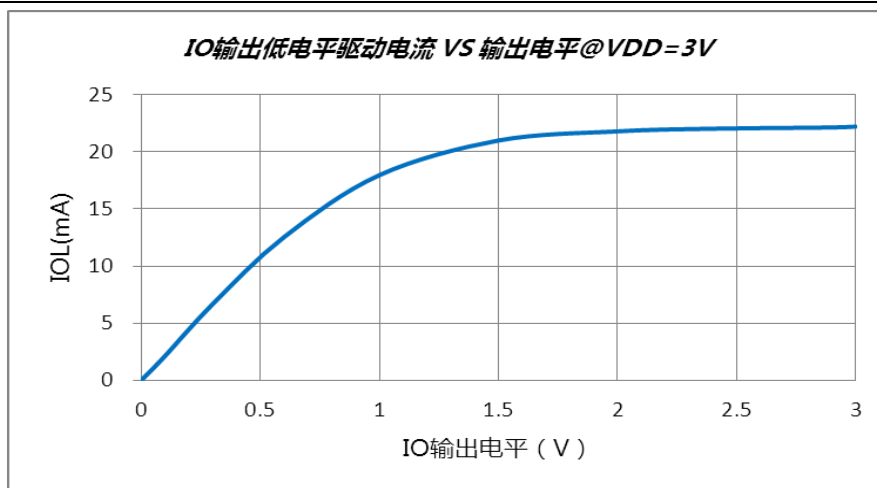
#### 输入 SMT 阈值电压 VS 电源电压



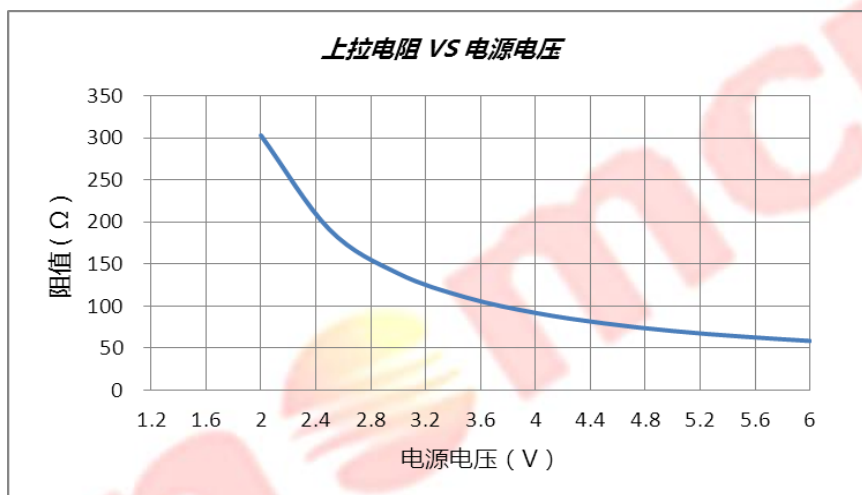
I/O 输出 驱动电流 VS 端口电压





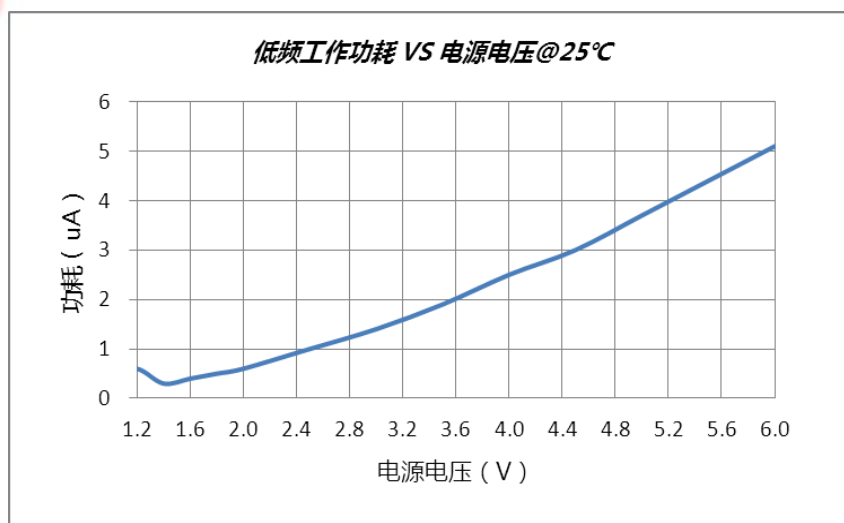


### 上拉电阻值 VS 电源电压

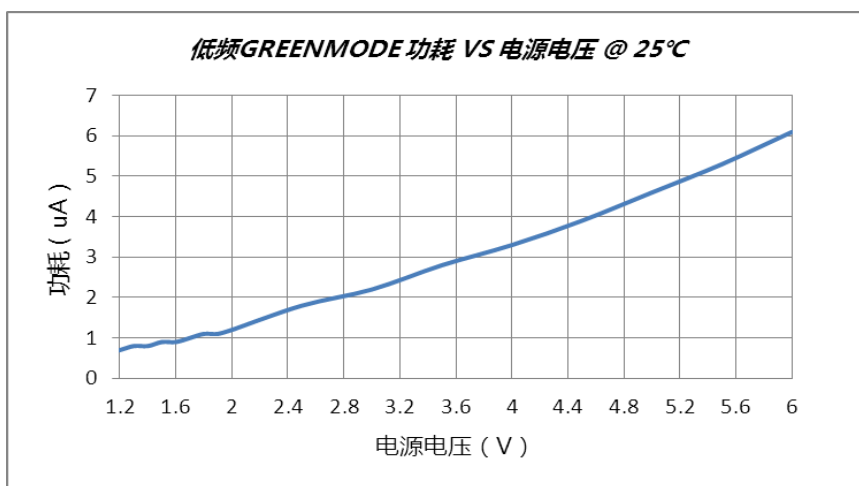


## 11.2 功耗特性

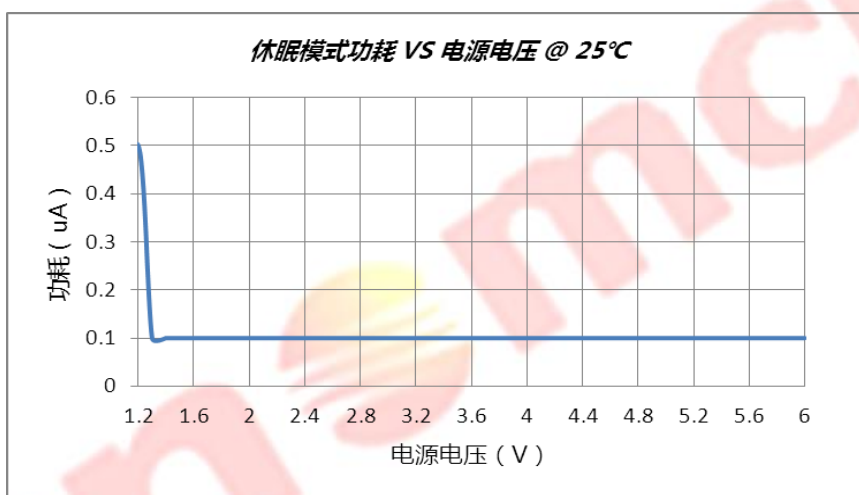
### 低速模式 功耗 VS 电源电压



## 空闲模式 功耗 VS 电源电压

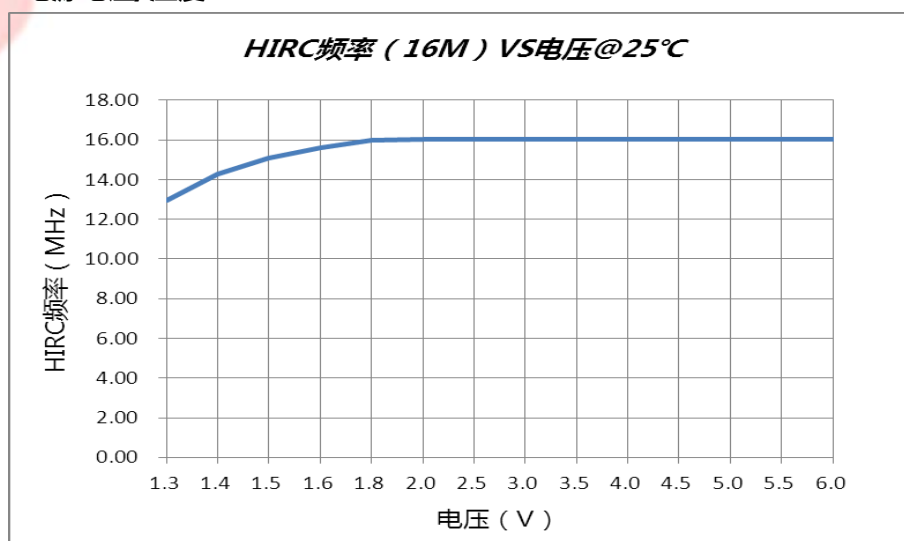


## 休眠模式 功耗 VS 电源电压

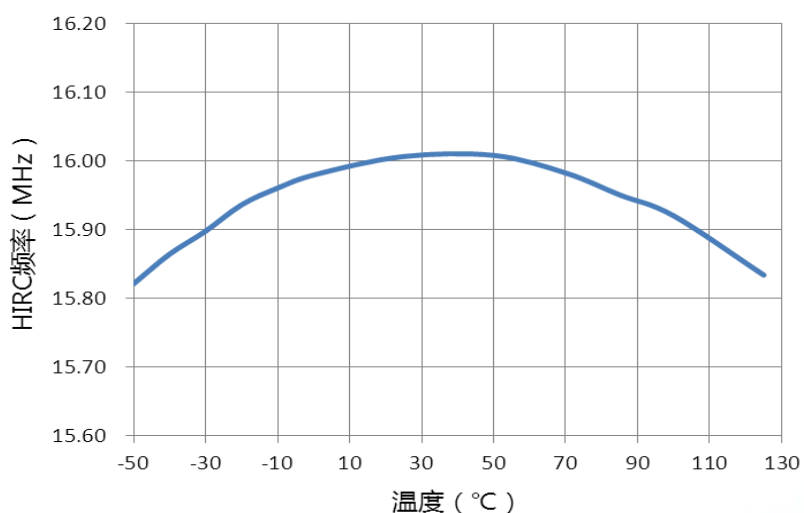


## 11.3 模拟电路特性

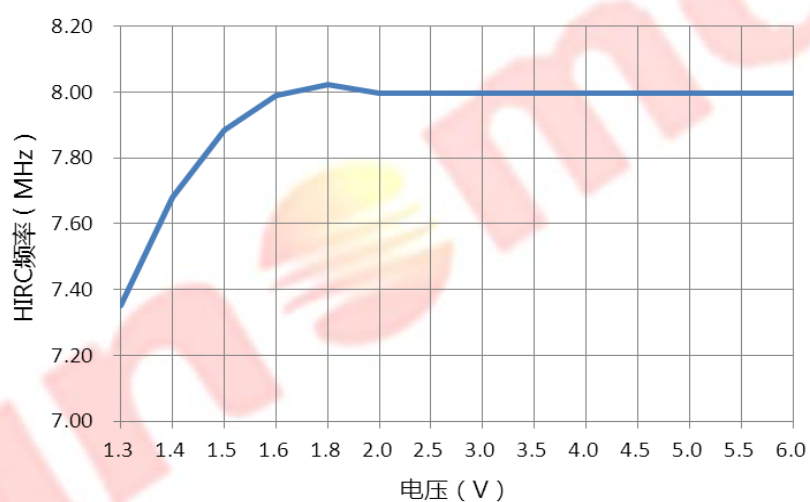
### HIRC 频率 VS 电源电压/温度



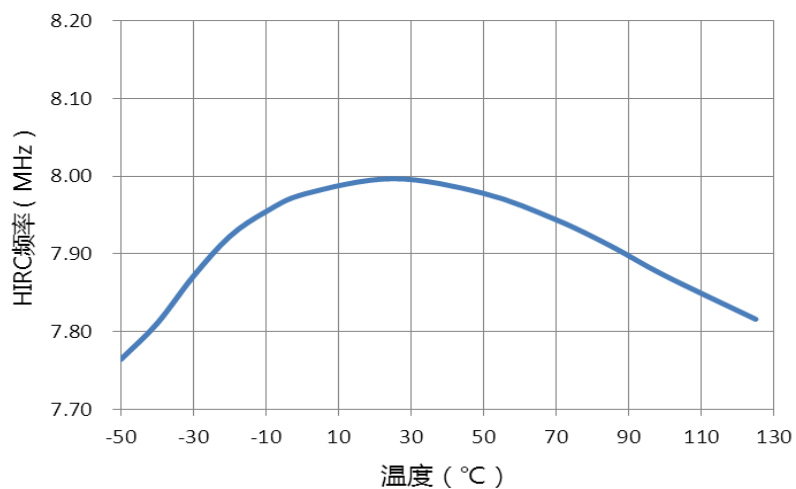
**HIRC频率 (16M) VS温度@VDD=5V**



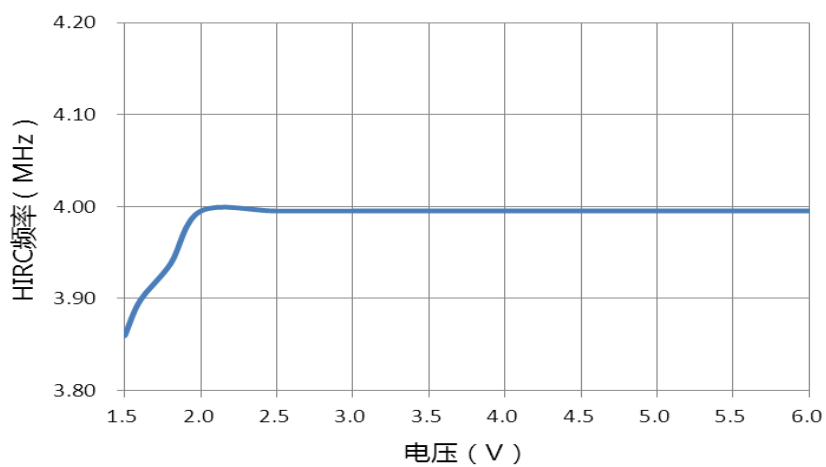
**HIRC频率 (8M) VS电压@25°C**



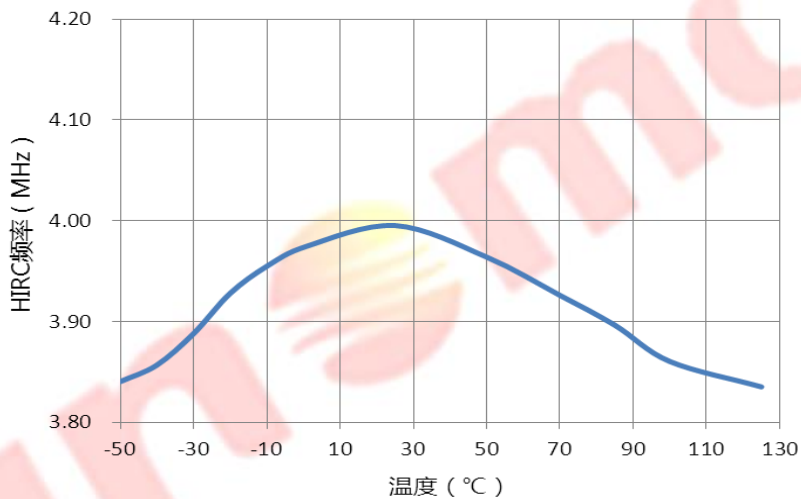
**HIRC频率 (8M) VS温度@VDD=5V**



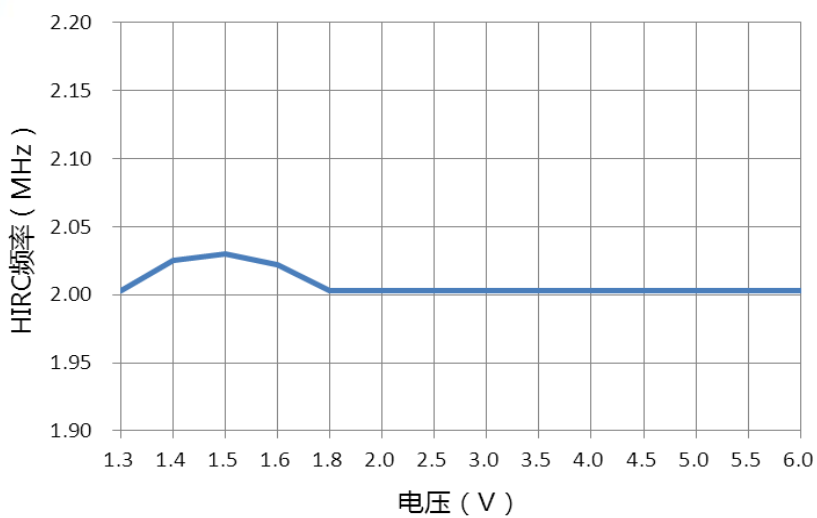
**HIRC频率 (4M) VS电压@25°C**



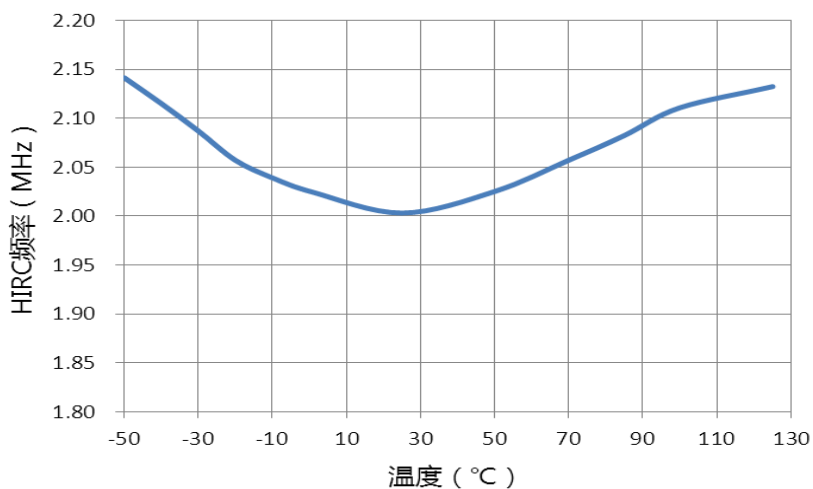
**HIRC频率 (4M) VS温度@VDD=5V**



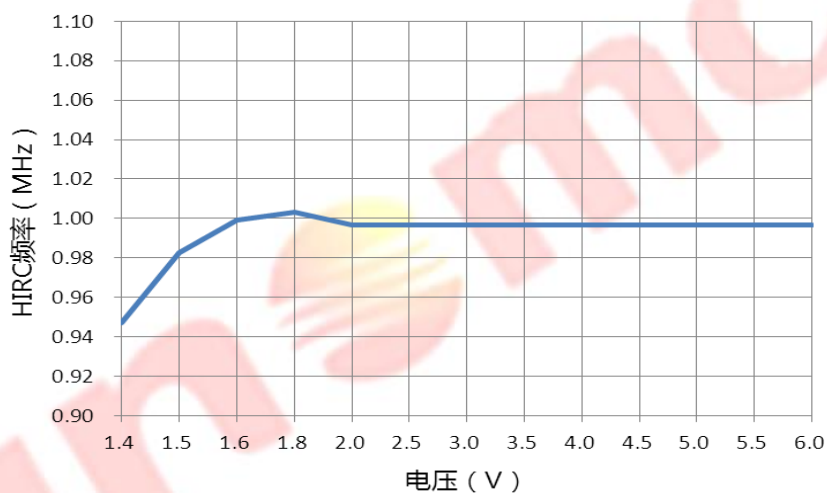
**HIRC频率 (2M) VS电压@25°C**



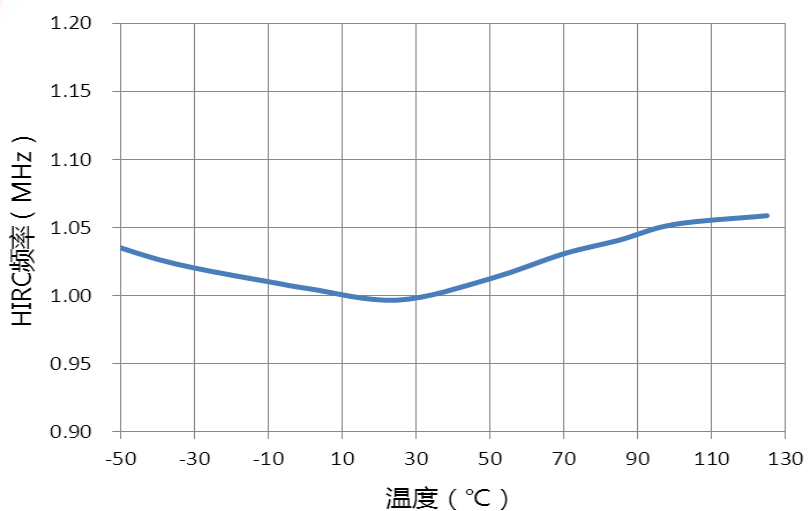
**HIRC频率 ( 2M ) VS温度@VDD=5V**

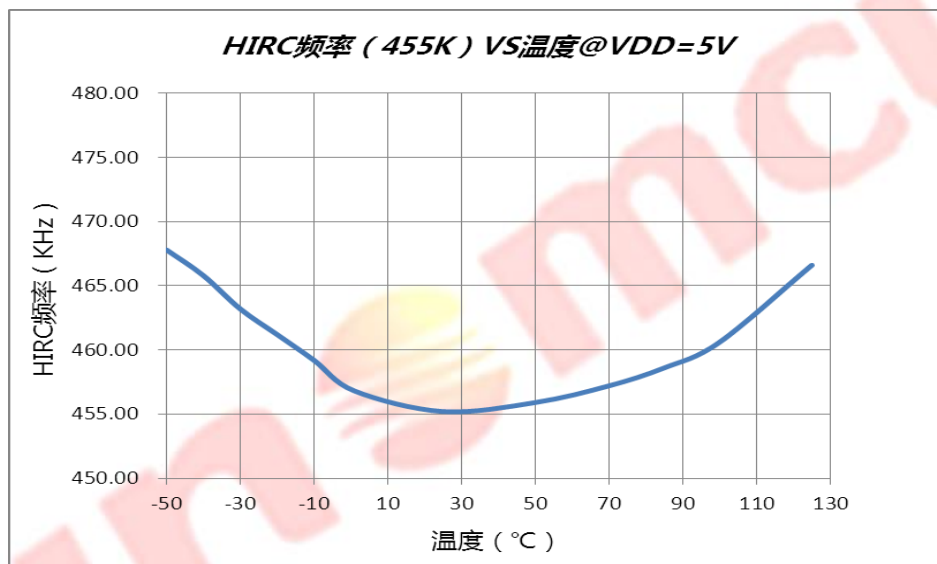
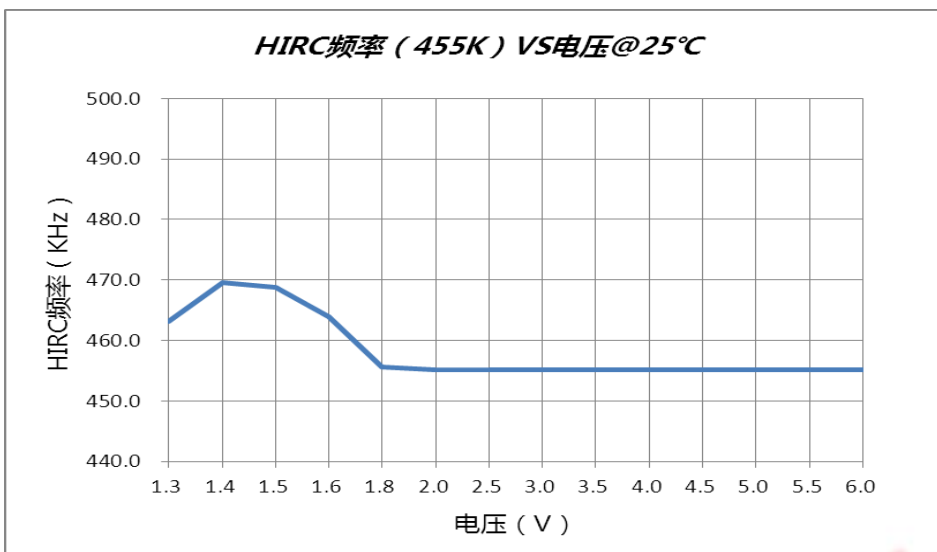


**HIRC频率 ( 1M ) VS电压@25°C**

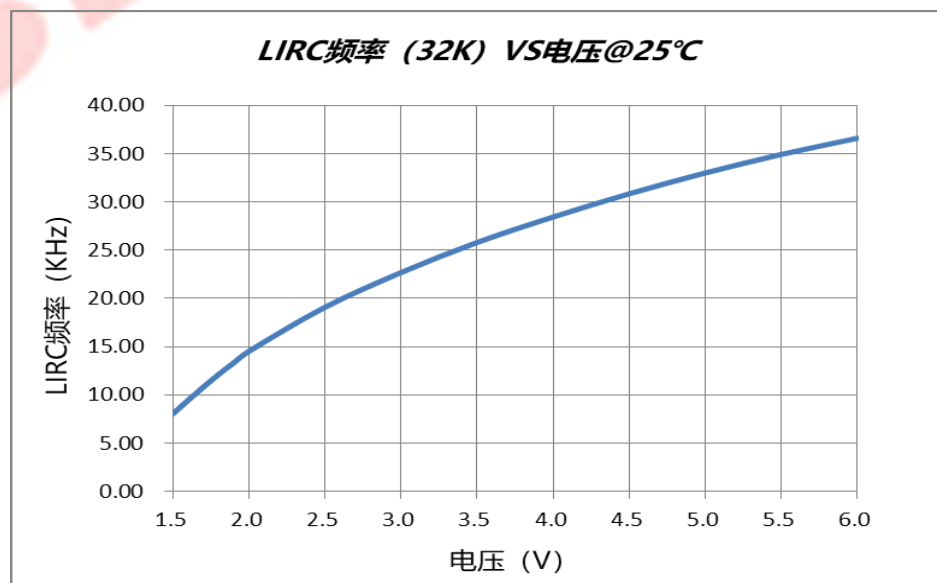


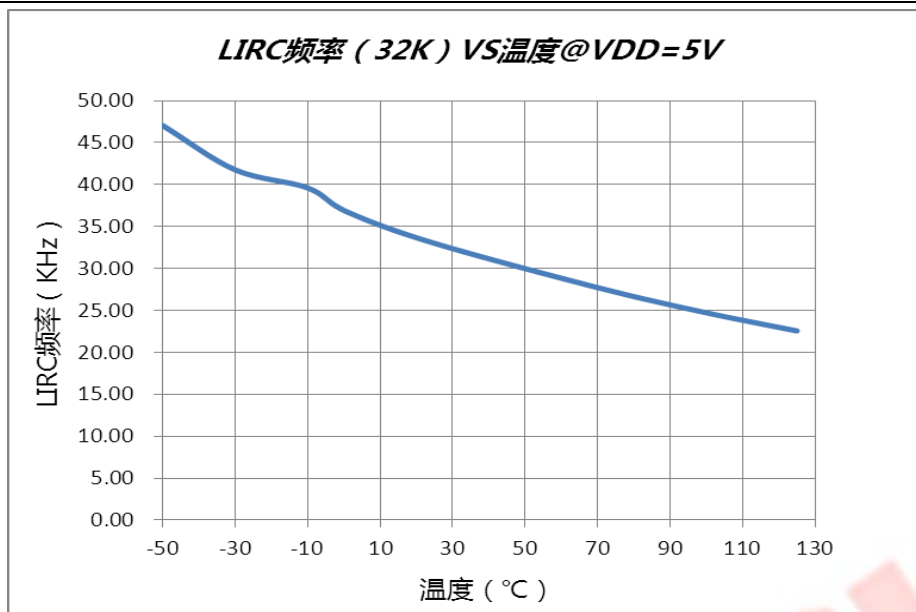
**HIRC频率 ( 1M ) VS温度@VDD=5V**



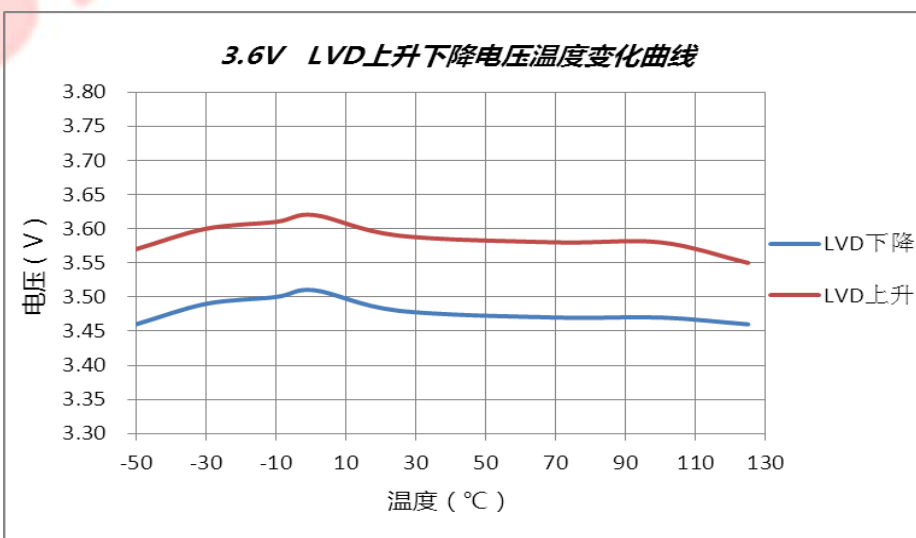
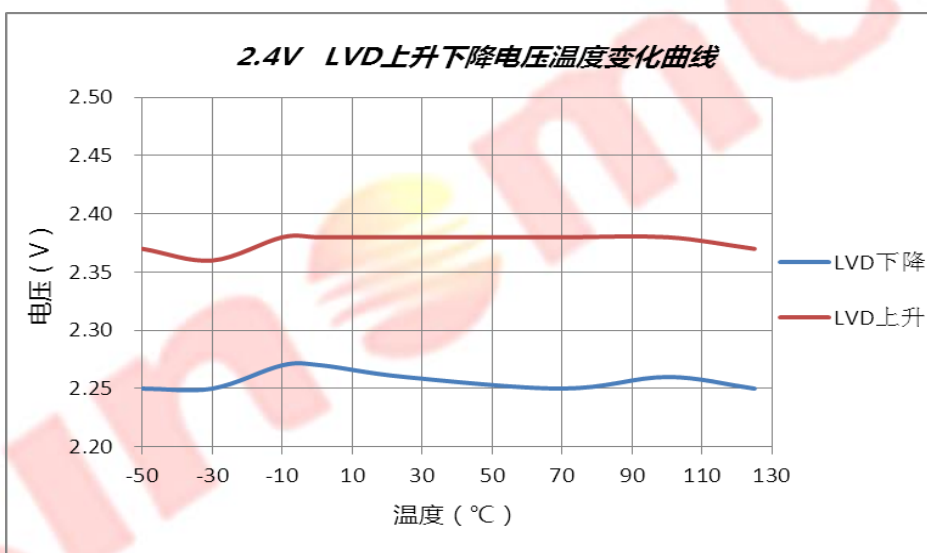


### LIRC 频率 VS 电源电压/温度



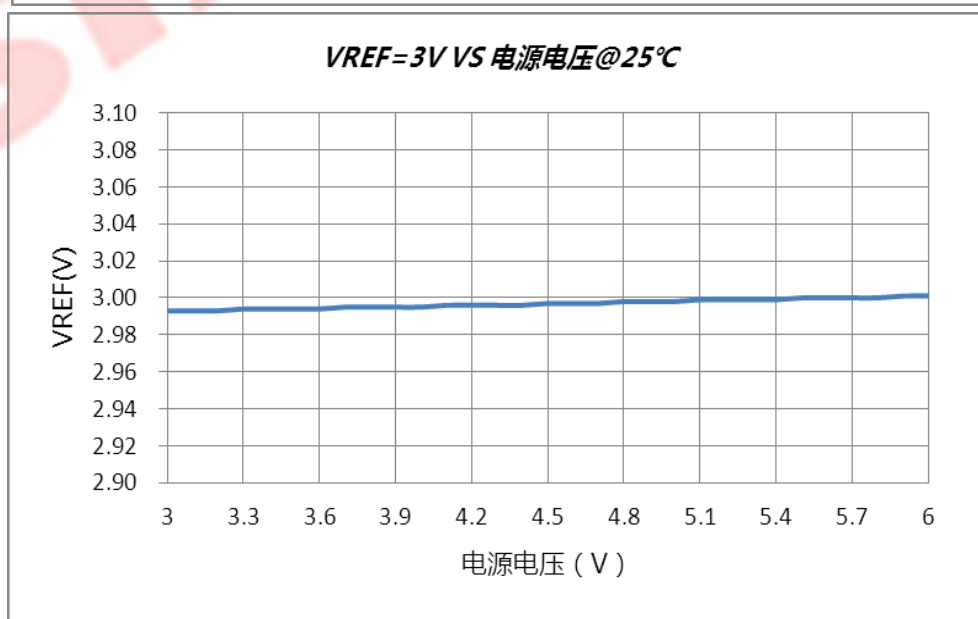
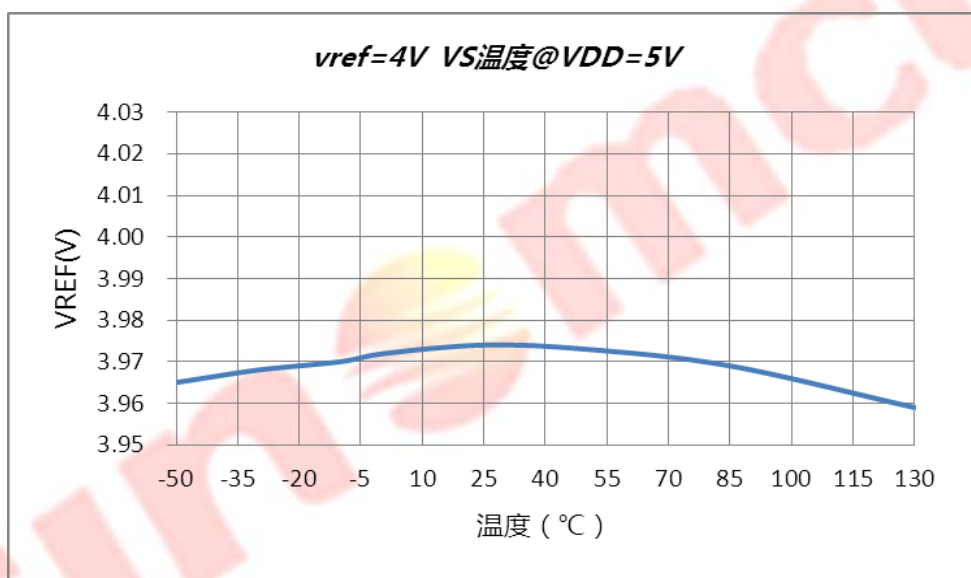
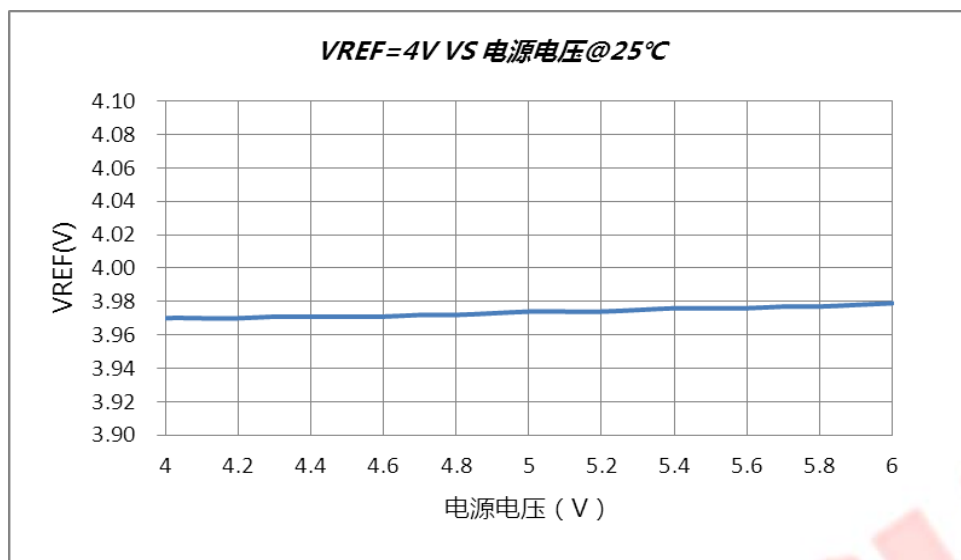


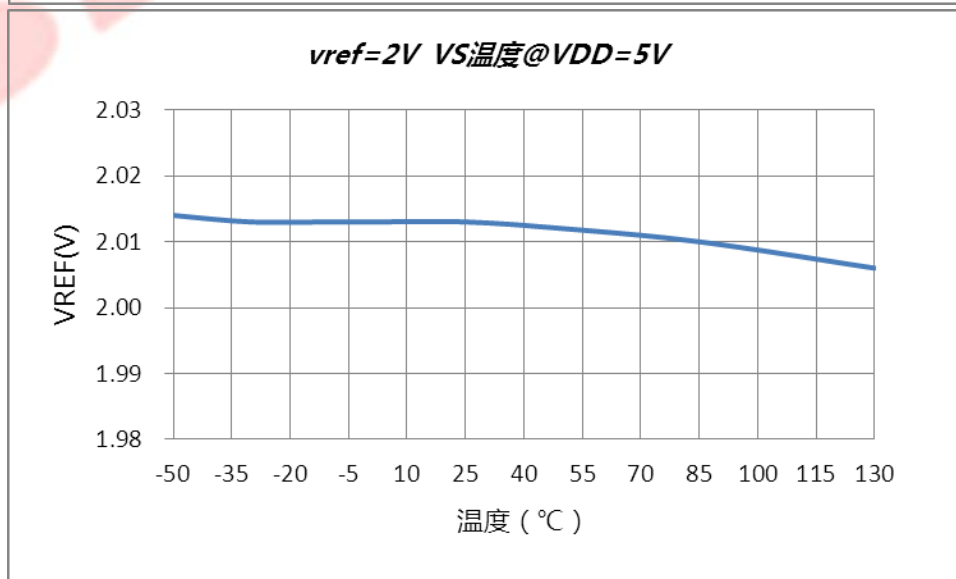
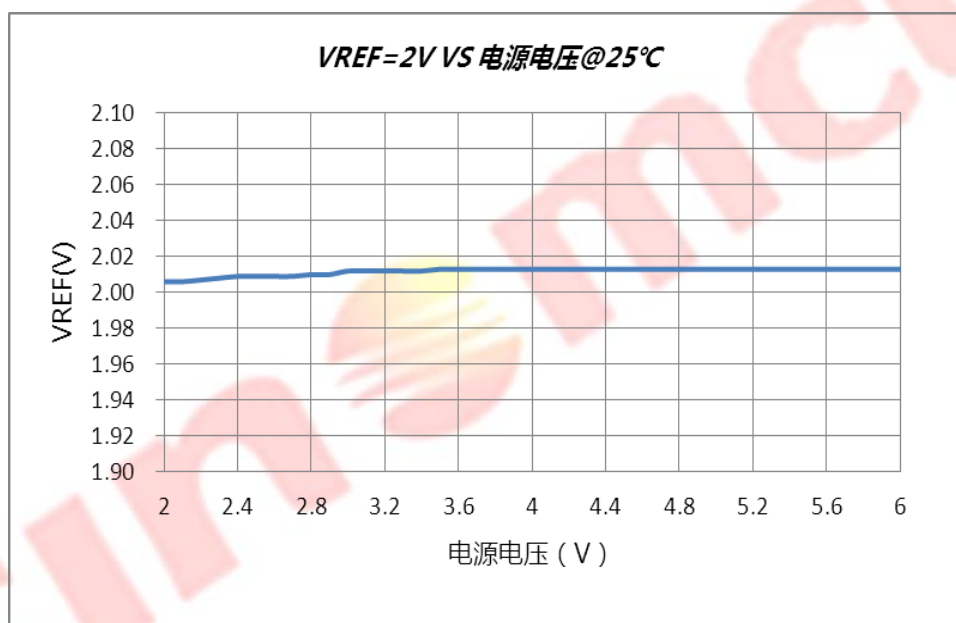
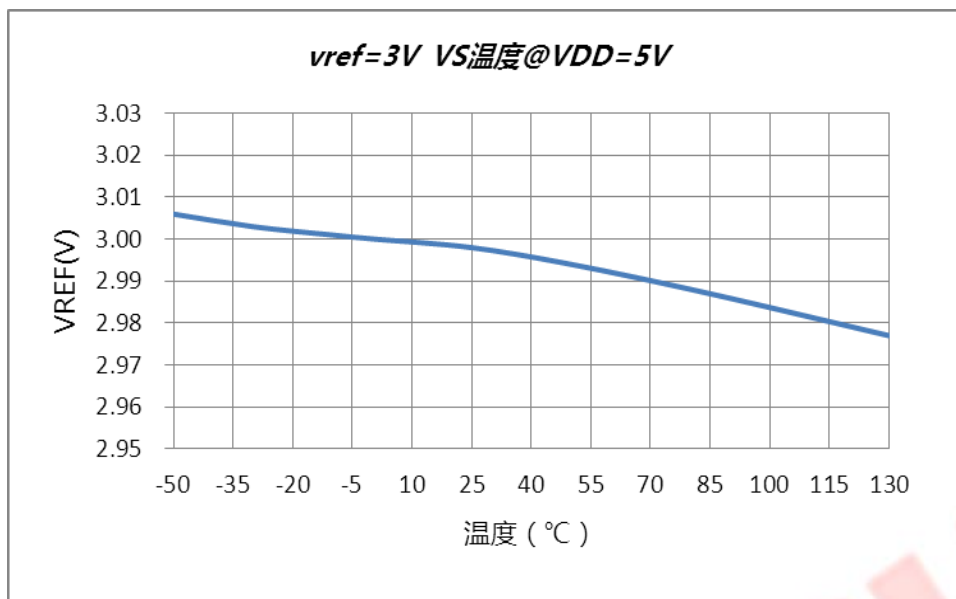
### LVD 电压 VS 温度





内部参考电压 VS 电源电压/温度

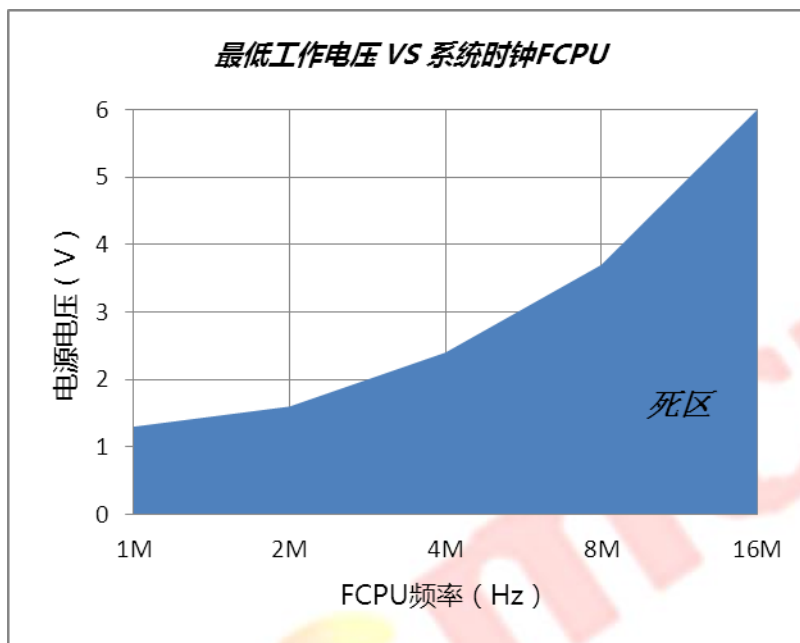




## 最低工作电压 VS CPU 时钟

系统最低工作电压和系统工作频率 FCPU 有关，不同的工作频率 FCPU 最低工作电压不同。

如下图所示，当工作频率提高时系统正常工作电压也随之提高，但由于 POR 电压固定(1.2V@25℃)，在系统最低工作电压和 POR 电压之间就会出现一个不能正常工作的电压区域，此区域系统不能正常工作也不会产生 POR 复位，称之为死区，必须根据不同的工作频率设置大于死区电压的 LVR 电压避免出现死区。

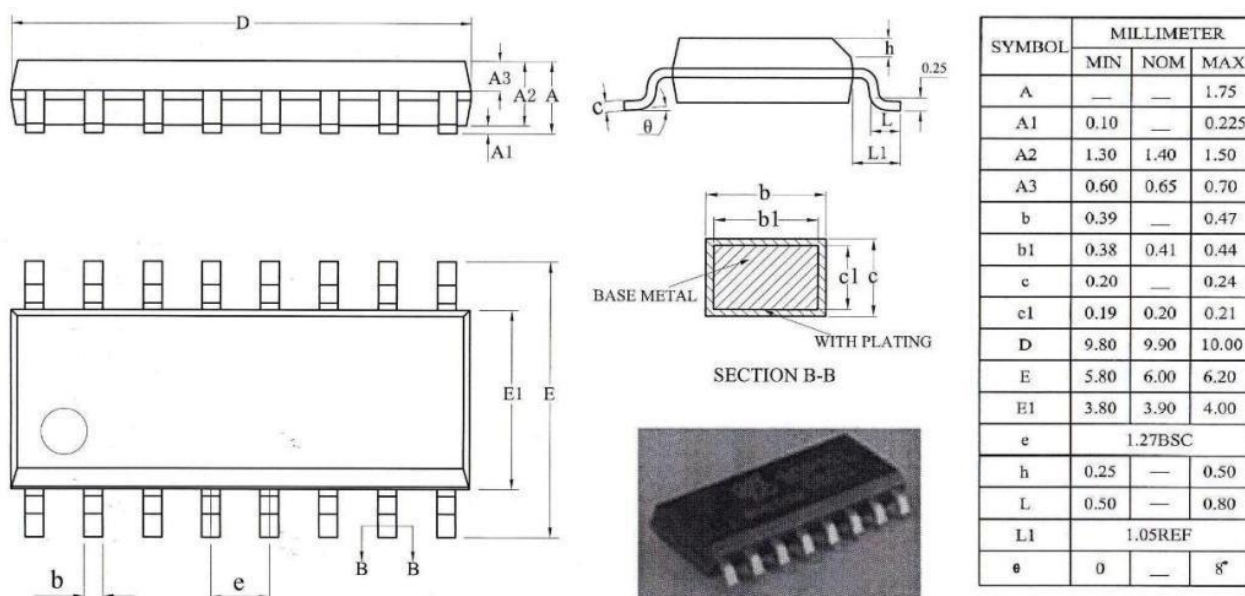


LVR 电压选择如下：

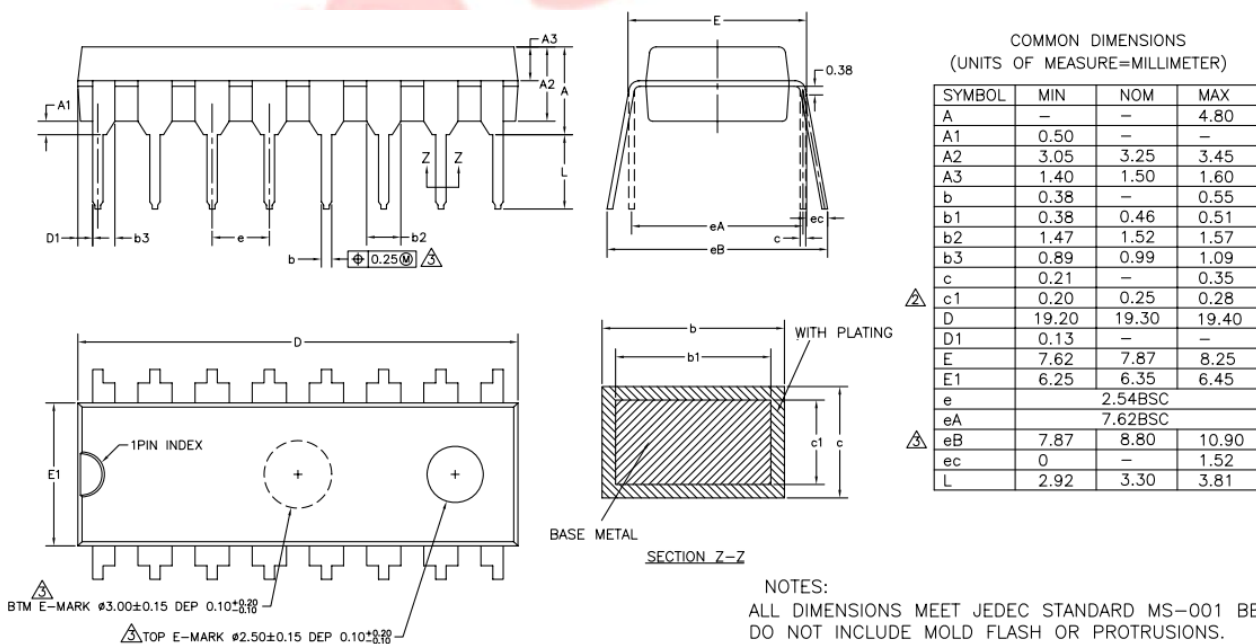
FCPU 频率(Hz)	LVR 电压值 (V)
8M	3.0
4M	2.7
2M	2.0
1M	2.0
500Khz	2.0

## 12 封装尺寸

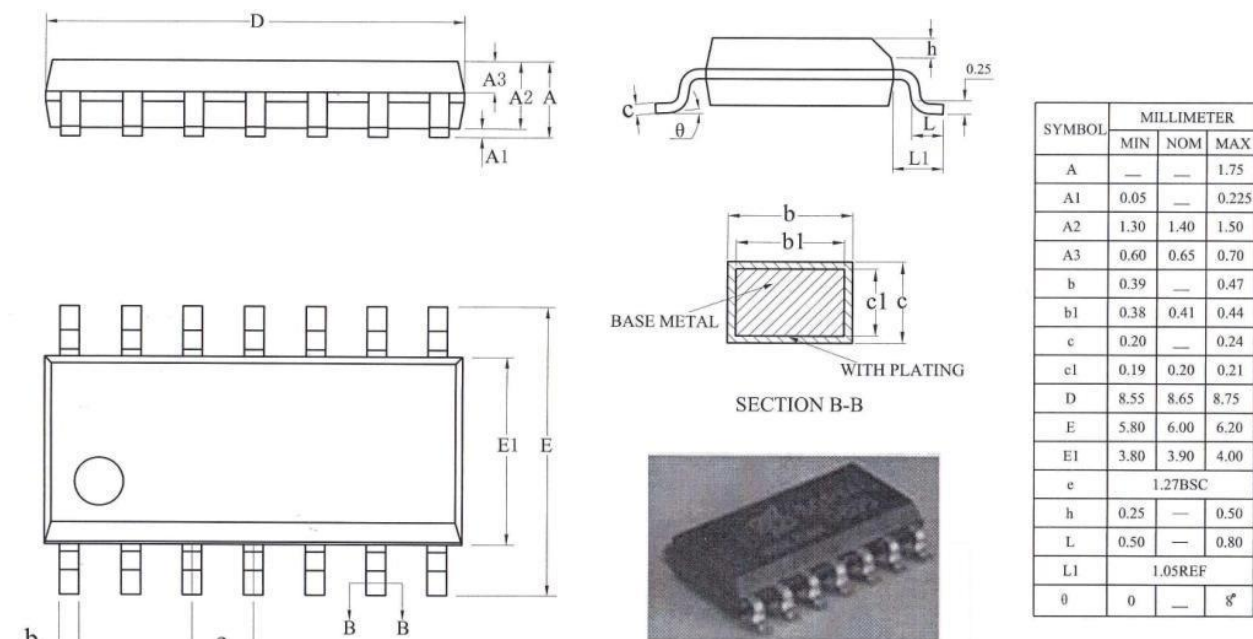
### 12.1 SOP16



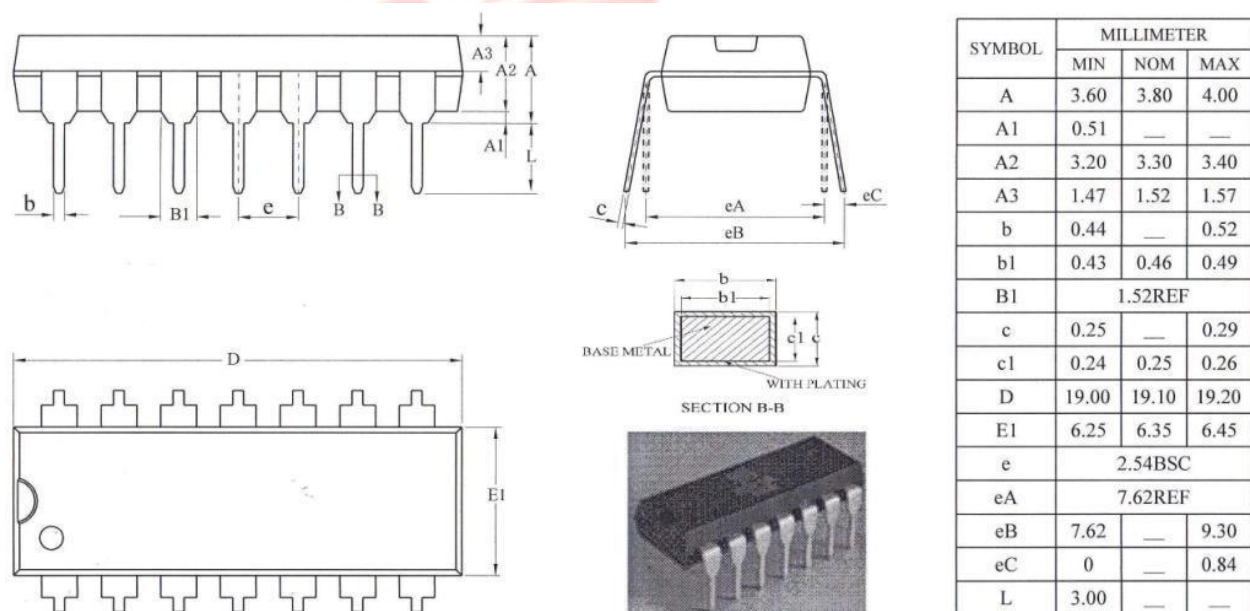
### 12.2 DIP16



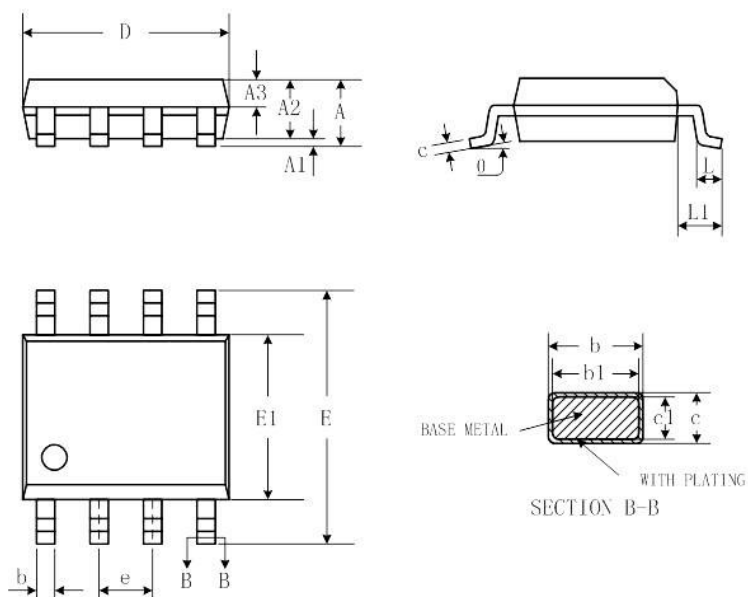
## 12.3 SOP14



## 12.4 DIP14

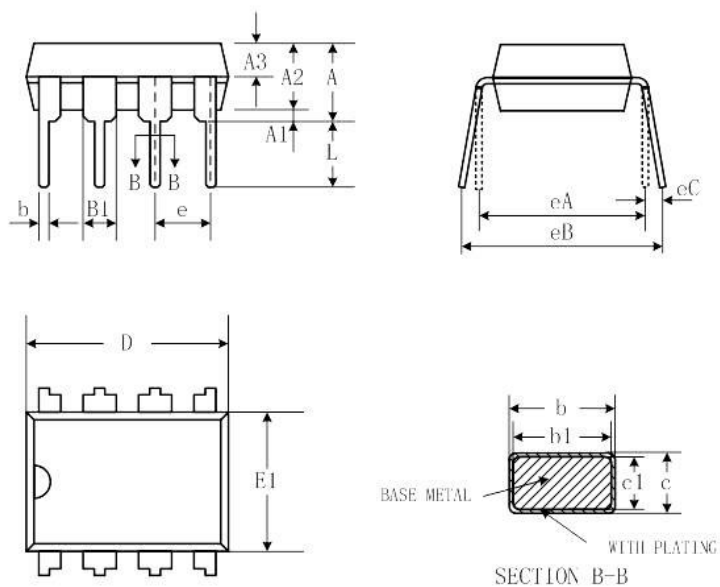


## 12.5 SOP8



SYMBOL	MILLIMETER		
	MIN	TYP	MAX
A	-	-	1.77
A1	0.08	0.18	0.28
A2	1.20	1.40	1.60
A3	0.55	0.65	0.75
b	0.39	-	0.48
b1	0.38	0.41	0.43
c	0.21	-	0.26
c1	0.19	0.20	0.21
D	4.70	4.90	5.10
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27BSC		
L	0.50	0.65	0.80
L1	1.05BSC		
θ	0	-	8°

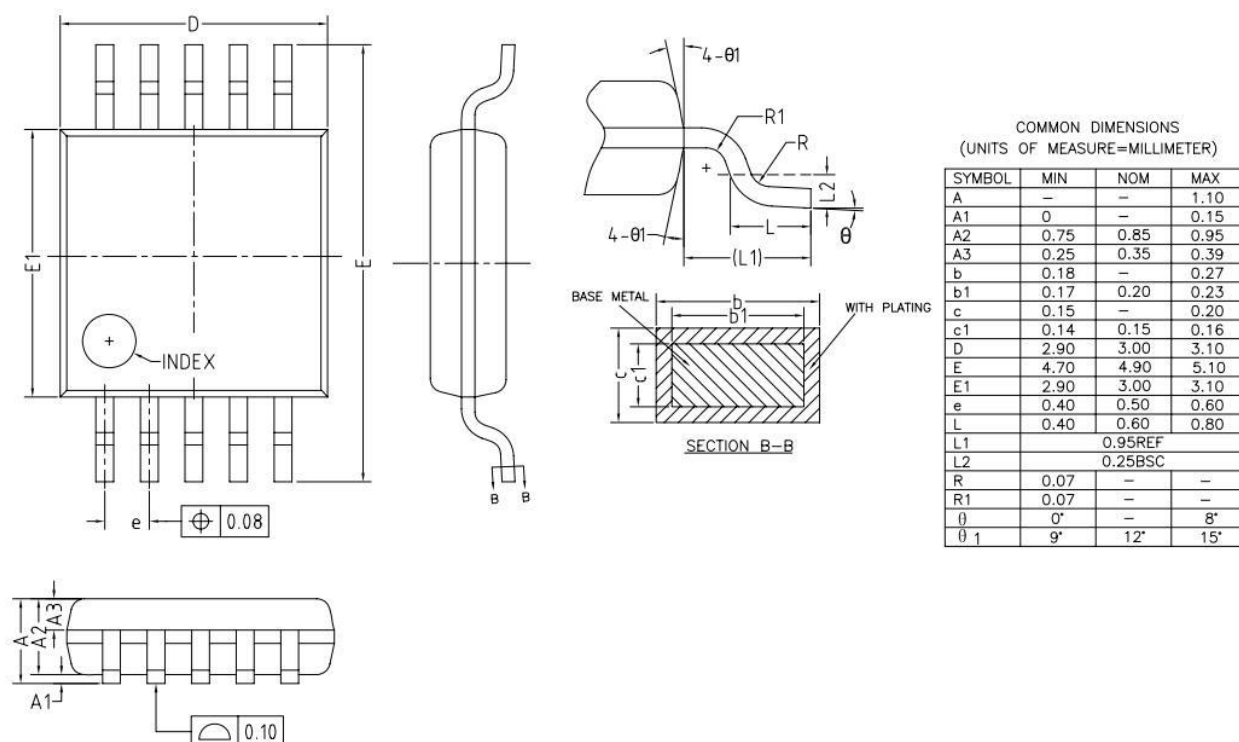
## 12.6 DIP8



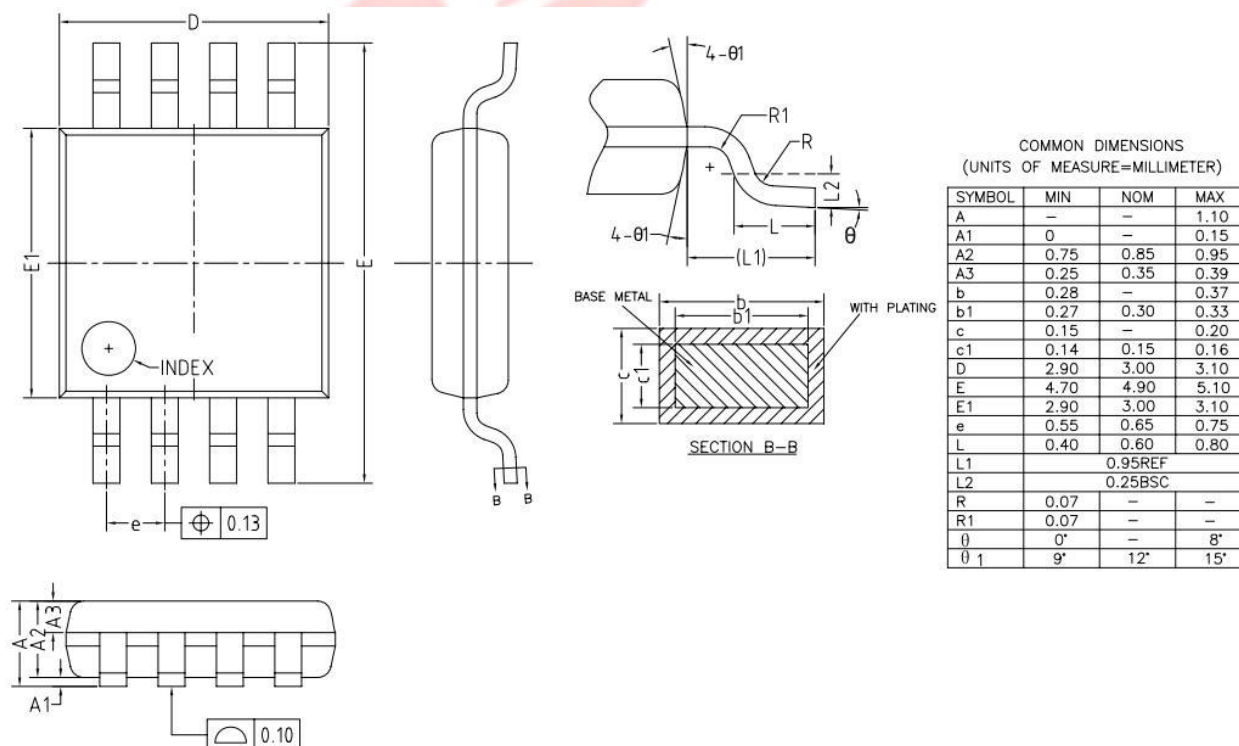
SYMBOL	MILLIMETER		
	MIN	TYP	MAX
A	3.60	3.80	4.00
A1	0.51	-	-
A2	3.10	3.30	3.50
A3	1.50	1.60	1.70
b	0.44	-	0.53
b1	0.43	0.46	0.48
B1	1.52BSC		
c	0.25	-	0.31
c1	0.24	0.25	0.26
D	9.05	9.25	9.45
E1	6.15	6.35	6.55
e	2.54BSC		
eA	7.62BSC		
eB	7.62	-	9.50
eC	0	-	0.94
L	3.00	-	-



## 12.7 MSOP10



## 12.8 MSOP8





## 13 修订记录

版本	修订日期	修订内容
V1.0	2016-01-26	新建;
V1.1	2016-04-05	修订笔误; 新增 8 脚和 14 脚封装; 直流电气特性的 VREF 端口改为 P40; 增加 P02 的 SDO 功能描述, 增加 P04 用作复位脚时带上拉电阻的描述;
V1.2	2017-04-05	修订笔误, 更新页眉; 增加指令的应用笔记说明;
V1.3	2017-10-10	修改 LVR 配置项, 修改工作电压特性及推荐 LVR 值; 修改 HIRC 电气参数;
V1.4	2017-06-20	新增 MSOP8 封装 A0I 和 MSOP10 封装 A0F;
V1.5	2018-10-17	增加 ADC 应用注释, 建议 AD 时钟限制 500KHz 以下, 采样时间配置为 14 个 ADCLK; 去除程序存储器按 1K 容量烧录 2 次的功能;
V1.6	2019-08-23	修订笔误, 更新封装尺寸图, 调整文档格式及章节顺序;
V1.7	2021-04-20	修订空闲模式描述笔误; 调整 HIRC 频率特性参数;
V1.8	2021-10-08	修订 LIRC 特性曲线图中笔误;
V1.9	2021-12-09	修订系统时钟示意图中笔误;