

*SinoMCU 8 位单片机*

MC35P7050

用户手册

V1.1



## 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 .....	6
2	电气特性 .....	7
2.1	极限参数 .....	7
2.2	直流电气特性 .....	7
2.3	交流电气特性 .....	8
2.4	ADC 特性参数 .....	8
3	CPU 及存储器 .....	10
3.1	指令集 .....	10
3.2	程序存储器 .....	11
3.3	数据存储器 .....	12
3.4	堆栈 .....	13
3.5	控制寄存器 .....	14
3.6	用户配置字 .....	17
4	系统时钟 .....	18
4.1	内部高频 RC 振荡器 .....	18
4.2	内部低频 RC 振荡器 .....	18
5	系统工作模式 .....	19
5.1	高速模式 .....	21
5.2	低速模式 .....	21
5.3	休眠模式 .....	21
5.4	空闲模式（绿色模式） .....	22
6	复位 .....	23
6.1	复位条件 .....	23
6.2	上电复位 .....	24
6.3	外部复位 .....	24
6.4	低电压复位 .....	24
6.5	看门狗复位 .....	24
7	I/O 端口 .....	25
7.1	通用 I/O 功能 .....	25
7.2	内部上拉电阻 .....	26
7.3	端口模式控制 .....	27
8	定时器 TIMER .....	28
8.1	看门狗定时器 WDT .....	28
8.2	定时器 T0 .....	28
8.3	定时器 T1 .....	31
9	模数转换器 ADC .....	34
9.1	ADC 概述 .....	34
9.2	ADC 操作步骤 .....	35

9.3	ADC 相关寄存器.....	35
9.4	ADC 校准流程.....	37
10	中断.....	39
10.1	外部中断.....	39
10.2	定时器中断.....	39
10.3	ADC 中断.....	40
10.4	中断相关寄存器.....	40
11	特性曲线.....	42
11.1	I/O 特性.....	42
11.2	功耗特性.....	45
11.3	模拟电路特性.....	49
12	封装尺寸.....	52
12.1	SOP8.....	52
12.2	DIP8.....	52
12.3	SOT23-6.....	53
13	修订记录.....	54

## 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 2/4/8/16 分频, 系统低频时钟下 F<sub>CPU</sub> 固定为 4 分频
- 存储器
  - ◇ 1K×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
  - ◇ 64 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 6 个 I/O
  - ◇ P0 (P02/P04), P4 (P40/P41/P44), P5 (P53)
  - ◇ P04 可配置为单输入或输入/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
  - ◇ 除 P04 外其余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 (16MHz), 可用作系统高频时钟源
  - ◇ 内置低频 RC 振荡器 (32KHz@5V, 16KHz@3V), 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
  - ◇ 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
- 2 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
- 1 个 12 位高精度逐次逼近型 ADC
  - ◇ 3 路外部通道: AN0/AN1/AN4; 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> (VERI 输入)
  - ◇ ADC 时钟: F<sub>CPU</sub> 的 4/8/32/64 分频
  - ◇ 支持零点或顶点校准
- 中断
  - ◇ 外部中断 (INT0~INT1)
  - ◇ 定时器中断 (T0~T1)
  - ◇ ADC 中断
- 低电压复位 LVR: 2.0V/2.4V/2.7V/3.6V
- 低电压检测 LVD: 关闭/2.4V/3.6V
- 工作电压

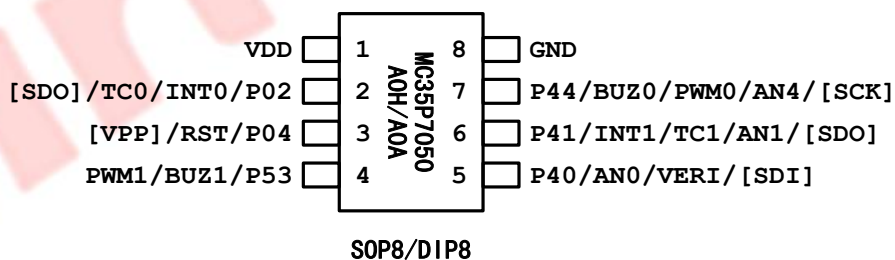
- ◇ VLVR27 ~ 5.5V @ Fcpu = 0~8MHz
- ◇ VLVR20 ~ 5.5V @ Fcpu = 0~4MHz
- 封装形式
  - ◇ SOP8/DIP8/SOT23-6

## 1.2 订购信息

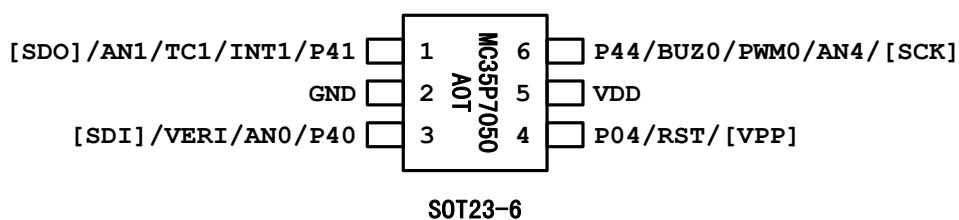
产品名称	封装形式	备注
MC35P7050A0H	SOP8	
MC35P7050A0A	DIP8	
MC35P7050A0T	SOT23-6	
MC35P7050A1T	SOT23-6	

## 1.3 引脚排列

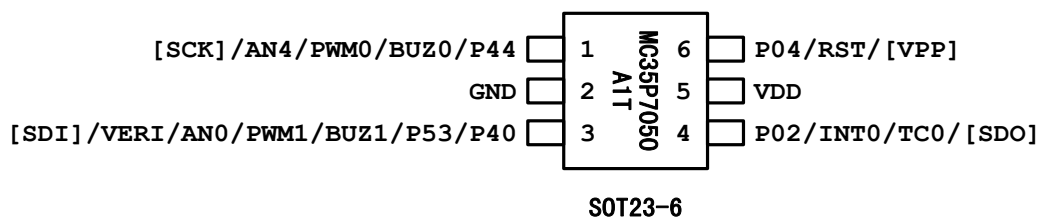
### MC35P7050A0H/A0A



### MC35P7050A0T



## MC35P7050A1T



注：PIN3 是端口 P40 和 P53 同封在 1 个引脚上，使用时应确保 2 个端口的工作模式、上拉电阻、复用功能、数模端口等设置不会产生冲突；若应用 P40 的 AD 功能时，静态功耗会偏大约 100uA。

### 1.4 端口说明

端口名称	类型	功能说明
VDD	P	电源
GND	P	地
P02, P40, P41, P44, P53	D	GPIO ( 推挽输出 ), 内部上拉
P04	D	GPIO ( 可配置为单输入或输入/开漏输出 )
INT0~INT1	DI	外部中断输入
TC0~TC1	DI	定时器 T0~T1 的外部计数输入
PWM0/BUZ0	DO	定时器 T0 的 PWM/BUZ 输出
PWM1/BUZ1	DO	定时器 T1 的 PWM/BUZ 输出
AN0~AN4	AI	ADC 模拟输入通道
VERI	AI	ADC 外部参考电压输入
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	50	mA
流出 GND 最大电流	IGNDmax	50	mA

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

### 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	VLVR27		5.5	V
			Fcpu=4MHz@FHIRC/4	VLVR20		5.5	
			Fcpu=2MHz@FHIRC/8	VLVR20		5.5	
			Fcpu=1MHz@FHIRC/16	VLVR20		5.5	
			Fcpu=8KHz@FLIRC/4	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	所有输入脚		0.8VDD			V
输入低电平	Vil	所有输入脚				0.2VDD	V
输出拉电流	Ioh	推挽输出脚		8	12		mA
输出灌电流	Iol	所有输出脚		8	15		mA
上拉电阻	Rpu	所有输入脚	VDD=5V, Vin=0	50	100	150	KΩ
			VDD=3V, Vin=0	100	200	300	KΩ
高速模式功耗	Ihigh	VDD	Fcpu=8MHz@FHIRC/2		3.0		mA
			Fcpu=4MHz@FHIRC/4		1.8		mA
			Fcpu=2MHz@FHIRC/8		1.0		mA
			Fcpu=1MHz@FHIRC/16		0.7		mA
低速模式功耗	Ilow	VDD	VDD=5V, Fcpu=FLIRC/4, HIRC 开		500		uA
			VDD=5V, Fcpu=FLIRC/4, HIRC 关		5	10	uA
			VDD=3V, Fcpu=FLIRC/4, 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		5	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		2	4	uA
低压检测电压	VLVD	VDD		-15%	2.4	+15%	V
				-15%	3.6	+15%	
低压复位电压	VLVR20	VDD	LVRVS 配置	-15%	2.0	+15%	V
	VLVR24		LVRVS 配置	-15%	2.4	+15%	
	VLVR27		LVRVS 配置	-15%	2.7	+15%	
	VLVR36		LVRVS 配置	-15%	3.6	+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	FHIRC	VDD=5V, T=25°C	-2%	16	+2%	MHz
		VDD=2.0V~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 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	VADC	T=-40°C~85°C	2.7		5.5	V
积分非线性误差	INL	VREF=VDD, FADC=1MHz, Tcon=27us	±2		±4	LSB
微分非线性误差	DNL	VREF=VDD, FADC=1MHz, Tcon=27us	±1		±3	LSB
零点偏移误差	EZ	VREF=VDD, FADC=1MHz, Tcon=27us			±3	LSB
增益误差	ET	VREF=VDD, FADC=1MHz, Tcon=27us			±3	LSB
转换时钟	FADC	VDD=5V			1	MHz
转换时间	Tcon			27		1/FADC
ADC 输入电压	VAIN		GND		VREF	V
ADC 输入阻抗	RAIN		2			MΩ
ADC 输入电流	Iain				2	uA
ADC 动态电流	IADD	VDD=5V, AD 转换中		0.3	2	mA
ADC 静态电流	IADS	VDD=5V, ADC 关闭		0.1	1	uA



模拟信号源推荐阻抗	Z <sub>AIN</sub>				10	KΩ
内部 1/4 分压电阻总值	R <sub>VDDI</sub>	V <sub>in</sub> =VDD=2.5V~5.5V	16	24	32	KΩ
电阻分压比值			-1%	1/4	+1%	VDD
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%	2/3/4	+1%	
		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  $J=1$ ，否则  $J=0$ ；
- 3、部分访问数据存储器的指令，若目的寄存器为 GPR，则  $M=1$ ，若目的寄存器为 SFR，则  $M=0$ ；
- 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；

## 3.2 程序存储器

芯片的程序存储器为 OTP 型存储器， $1K \times 16$  位的地址空间范围为 0000H~03FFH。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 - 03FFH)

程序存储器支持间接寻址, 可通过 MOV<sub>C</sub> 指令访问 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-3FH	GPR	通用数据寄存器区							
40H-7FH	保留	保留							
80H-87H	SFR			HIBYTE	FSR0	FSR1		PFLAG	
88H-8FH									
90H-97H									
98H-9FH									
A0H-A7H									
A8H-AFH								P4CON	VREFCR
B0H-B7H			ADM	ADB	ADR	ADT			
B8H-BFH		OEP0							EINTCR
C0H-C7H						OEP4	OEP5		

C8H-CFH		INTF	INTE	OSCM		WDTCR	T0LDR	PCL	PCH
D0H-D7H		IOP0				IOP4	IOP5		
D8H-DFH		TMRCR		T0CR	T0CNT	T1CR	T1CNT	T1LDR	MCR
E0H-E7H		PUP0				PUP4	PUP5		INDF
E8H-EFH									
F0H-F7H									
F8H-FFH		STKR3L	STKR3H	STKR2L	STKR2H	STKR1L	STKR1H	STKR0L	STKR0H

**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

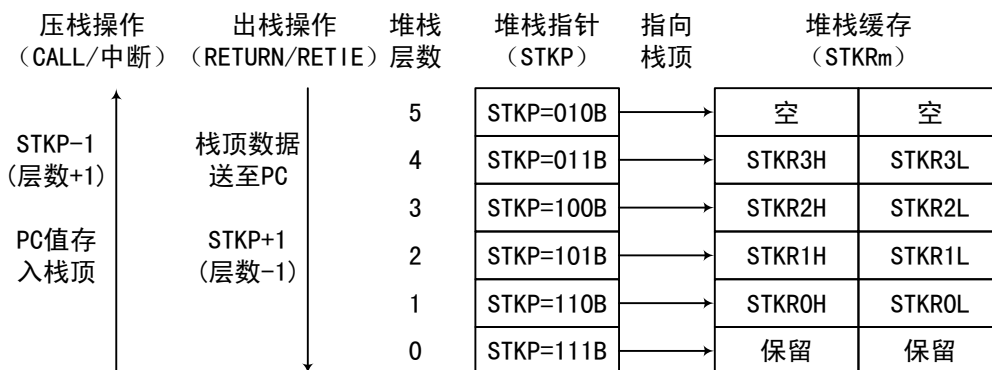
MOVRA FSR0

MOVAI 55H

MOVRA INDF ; 将数据 55H 写入 FSR1×256+FSR0 指向的数据存储器中

### 3.4 堆栈

4 级 10 位堆栈缓存寄存器 STKRm (m=3-0)，位于特殊功能寄存器 SFR 区，3 位堆栈指针 STKP 位于杂项控制寄存器 MCR 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

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>FSR0</b>	FSR07	FSR06	FSR05	FSR04	FSR03	FSR02	FSR01	FSR00
<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>	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。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>FSR1</b>	FSR17	FSR16	FSR15	FSR14	FSR13	FSR12	FSR11	FSR10
<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>	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
<b>INDF</b>	INDF7	INDF6	INDF5	INDF4	INDF3	INDF2	INDF1	INDF0
<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>	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
<b>HIBYTE</b>	HIBYTE7	HIBYTE6	HIBYTE5	HIBYTE4	HIBYTE3	HIBYTE2	HIBYTE1	HIBYTE0
<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>	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 MOVC 指令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
<b>PCH</b>	-	-	-	-	-	-	PC9	PC8
<b>R/W</b>	-	-	-	-	-	-	R/W	R/W
<b>初始值</b>	-	-	-	-	-	-	0	0

BIT[1:0] **PC[9:8]** – 程序指针计数器高 2 位

	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 10 位；
- ✧ 返回指令 RETIE/RETURN：PC = 堆栈栈顶（TOS）；

对 PCL 操作指令：

- ✧ 对 PCL 操作的加法指令：PC = (PC[9:0]+ALU[7:0])；
- ✧ 对 PCL 操作的其它指令：PC = {PC[9:8]:ALU[7:0](ALU 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	TO	PD	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	-	0	0	0

BIT[7,6] TO,PD – 复位状态标志位

TO	PD	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 LVDVS 选择 3.6V 时有效）

0: VDD 电压高于 3.6V（有回滞）；

1: VDD 电压低于 3.6V；

BIT[4] LVD24 – 2.4V 检测状态标志位（仅配置字 LVDVS 选择 2.4V 时有效）

0: VDD 电压高于 2.4V（有回滞）；

1: VDD 电压低于 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；

### 杂项控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>MCR</b>	GIE	-	-	-	-	STKP2	STKP1	STKP0
<b>R/W</b>	R/W	-	-	-	-	R/W	R/W	R/W
<b>初始值</b>	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 (m=0-3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>STKRmH</b>	-	-	-	-	-	-	STKRmB9	STKRmB8
<b>R/W</b>	-	-	-	-	-	-	R/W	R/W
<b>初始值</b>	-	-	-	-	-	-	0	0

BIT[1:0]      **STKRmB[9:8]** – 堆栈缓存寄存器 STKRm 的高 2 位

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>STKRmL</b>	STKRmB7	STKRmB6	STKRmB5	STKRmB4	STKRmB3	STKRmB2	STKRmB1	STKRmB0
<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]      **STKRmB[7:0]** – 堆栈缓存寄存器 STKRm 的低 8 位



### 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
<b>FCPUS</b>	高频时钟下 F <sub>CPU</sub> 分频选择： F <sub>CPU</sub> =F <sub>HOSC</sub> /2； F <sub>CPU</sub> =F <sub>HOSC</sub> /4； F <sub>CPU</sub> =F <sub>HOSC</sub> /8； F <sub>CPU</sub> =F <sub>HOSC</sub> /16；
<b>RSTEN</b>	RST 外部复位设置： P04 为外部复位脚； P04 为输入/输出脚；
<b>LVRVS</b>	LVR 复位电压选择：( LVR 电压应满足由 F <sub>CPU</sub> 决定的工作电压特性 ) 2.0V；2.4V；2.7V；3.6V；
<b>LVDVS</b>	LVD 检测电压选择： LVD 关闭；2.4V；3.6V；
<b>WDTM</b>	WDT 模式设置： WDT 始终关闭； WDT 低功耗模式下关闭； WDT 始终开启；
<b>P04PMS</b>	P04 端口模式选择：( 仅在通过 RSTEN 设置 P04 为输入/输出脚时有效 ) P04 为单输入口； P04 为输入/开漏输出口；
<b>DIGPS</b>	数字功能端口映射选择： 端口不映射：数字功能 PWM0/BUZ0、INT0/TC0、INT1/TC1 无效； 端口映射：PWM0/BUZ0、INT0/TC0、INT1/TC1 端口与 GPIO 复用；
<b>ENCR</b>	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

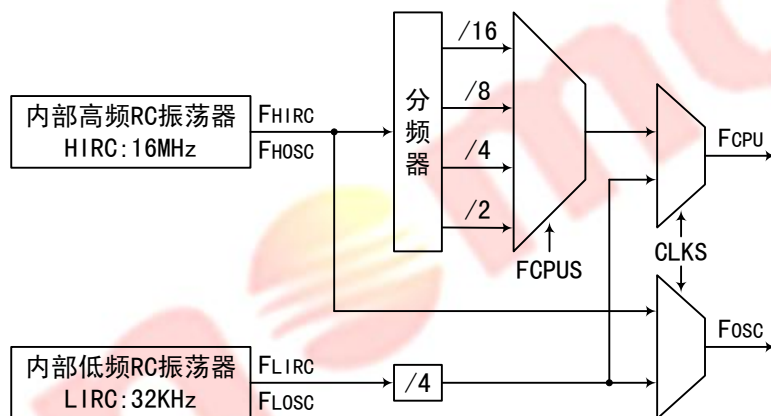
芯片为双时钟系统，内部电路均在系统高频时钟  $F_{HOSC}$  或系统低频时钟  $F_{LOSC}$  下工作，部分模块的时钟还可在  $F_{HOSC}$  和  $F_{LOSC}$  之间切换。

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

CPU 的时钟源可在系统高频时钟  $F_{HOSC}$  和系统低频时钟  $F_{LOSC}$  之间切换。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 个振荡频率为 16MHz 的高精度  $HIRC$  振荡器，可用作系统高频时钟源。

### 4.2 内部低频 RC 振荡器

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

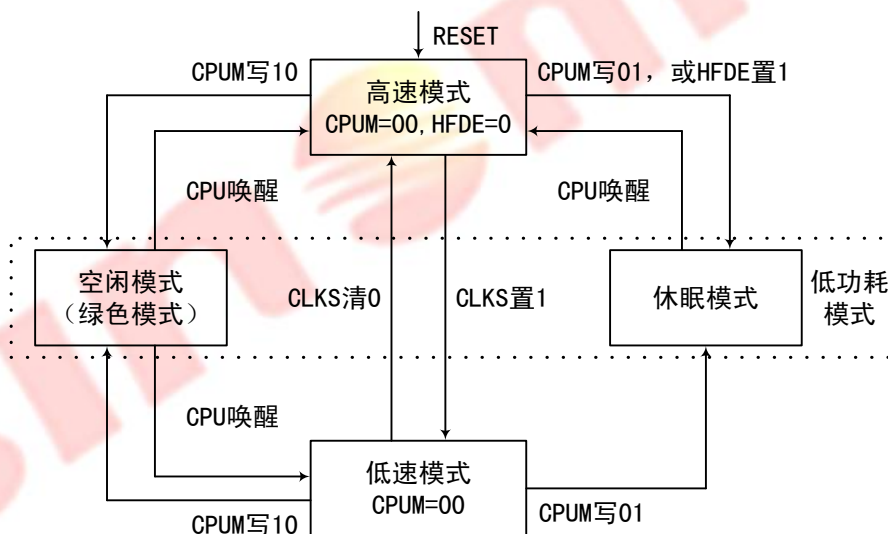
## 5 系统工作模式

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，其中高/低速模式为运行模式，CPU 运行；而空闲模式和休眠模式为低功耗模式，CPU 暂停工作。

工作模式	切入条件	系统状态
高速模式	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	
	空闲/休眠模式下，CPU 唤醒	
低速模式	高速模式下，CLKS 置 1	CPU 低速运行，高频时钟源由 HFDE 决定
	空闲/休眠模式下，CPU 唤醒	
空闲模式	高/低速模式下，CPUM 写 10	CPU 暂停，高频时钟源由 HFDE 决定
休眠模式	高速模式下，CPUM 写 01、或 HFDE 置 1	CPU 暂停，高/低频时钟源均停止
	低速模式下，CPUM 写 01	

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



### 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	HFDE	-
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** – 系统时钟及 CPU 时钟选择位  
 0: FHOSC 作为 FOSC, FHOSC 的分频时钟作为 Fcpu;  
 1: FOSC 的 4 分频时钟作为 Fosc 和 Fcpu;

BIT[1]      **HFDE** – 高频时钟源控制位  
 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	通过 HFDE 控制	继续原状态	停止
内部低频 RC 振荡器	工作	工作	工作	由 WDTM 决定 WDT 开则 LIRC 工作 WDT 关则 LIRC 停止
CPU	工作	工作	停止	停止
看门狗定时器 WDT	由 WDTM 决定	由 WDTM 决定	由 WDTM 决定	由 WDTM 决定
T0 及其中断	通过 T0EN 控制 溢出可产生中断	通过 T0EN 控制 溢出可产生中断	继续原状态, 溢出置标志位 T0WUEN 决定溢出是否唤醒	暂停
T1 及其中断	通过 T1EN 控制 溢出可产生中断	通过 T1EN 控制 溢出可产生中断	继续原状态 溢出仅置标志位, 不可唤醒	暂停
ADC 及其中断	通过 ADEN 控制 结束可产生中断	通过 ADEN 控制 结束可产生中断	暂停	暂停
外部中断	触发可产生中断	触发可产生中断	触发仅置标志位	触发仅置标志位
唤醒方法	-	-	P0 电平变化, T0 溢出, 复位	P0 电平变化, 复位

在高/低速模式中, 可通过位操作指令 BSET 将 CPUM0 置 1 进入休眠模式、或将 CPUM1 置 1 进入空闲模式。但请注意, 因指令的预读寄存器机制及 CPU 的并行流水线架构, 若唤醒后的第一条指令又立即对寄存器 OSCM 进行操作, 则会再次进入休眠或空闲模式。

可按以下例程实现运行-休眠、运行-空闲的模式切换:

```

.....          ; 运行模式
BSET            CPUM0      ; CPUM0 置 1, 进入休眠模式
NOP             ; 唤醒后第一条指令
.....          ; 运行模式
BSET            CPUM1      ; CPUM1 置 1, 进入空闲模式
NOP             ; 唤醒后第一条指令
.....          ; 运行模式
    
```

## 5.1 高速模式

高速模式下，系统（包括 CPU）的工作时钟由高频时钟 FHOSC 提供。上电复位或其他复位触发后，系统以高速模式工作，当系统从休眠模式唤醒后也进入高速模式。

- ✧ CPU 以 FHOSC 为时钟源，在其分频时钟下执行指令代码；
- ✧ 系统在 FHOSC 下工作；所有模块均可在选定的时钟下工作；
- ✧ 高频时钟源和内部低频 RC 振荡器均正常工作；
- ✧ 系统可从高速模式转入其他任意模式；
- ✧ 系统可从低速模式转入高速模式；
- ✧ 系统从高速模式转入空闲模式后，再唤醒将返回到高速模式；
- ✧ 系统从休眠模式唤醒后，转入高速模式；

## 5.2 低速模式

低速模式下，系统（包括 CPU）的工作时钟由 FLOSC 提供。高/低速模式的切换由寄存器位 CLKS 控制，CLKS 清 0 后转入高速模式，CLKS 置 1 后转入低速模式。从高速模式转入低速模式时，高频时钟源并不自动停止，可通过寄存器位 HFDE 关闭高频时钟源以降低功耗；而从低速模式转入高速模式时，需先通过 HFDE 清 0 开启高频时钟源，然后再通过 CLKS 清 0 切换模式。

- ✧ CPU 以 FLOSC 的 4 分频时钟执行指令代码；
- ✧ 系统在 FLOSC 的 4 分频时钟下工作；所有模块均可在选定的时钟下工作；
- ✧ 内部低频 RC 振荡器正常工作，高频时钟源由 HFDE 控制；
- ✧ 系统可从低速模式转入其他任意模式；
- ✧ 系统可从高速模式转入低速模式；
- ✧ 系统从低速模式转入空闲模式后，再唤醒将返回到低速模式；

## 5.3 休眠模式

休眠模式为低功耗模式，系统和 CPU 均停止工作，高频时钟源停止工作，内部低频 RC 振荡器根据 WDT 模式决定（若 WDT 始终工作则 LIRC 工作，否则 LIRC 停止）。休眠模式可由选定端口的电平变化来唤醒，系统从其他任意模式转入到休眠模式后，再唤醒均将以高速模式恢复工作。可通过寄存器位 CPUM 控制是否进入休眠模式，当 CPUM=01 时，系统进入休眠模式，唤醒后 CPUM 自动恢复成 00，CLKS 也自动清为 0，系统进入高速模式。

- ✧ CPU 停止工作，程序代码暂停执行；
- ✧ 包括内部高/低速 RC 振荡器、外部高速振荡器在内的所有时钟源均停止工作。
- ✧ 选定端口的电平变化，将唤醒系统；
- ✧ 系统从休眠模式唤醒后进入高速模式；

**注：高速模式下 HFDE 置 1，系统也将进入休眠模式；在低速模式且 HFDE 置 1 时，直接将 CLKS 清 0 会误进入休眠模式。**

## 5.4 空闲模式（绿色模式）

空闲模式又称为绿色模式，也为低功耗模式，CPU 停止工作，但切入空闲模式前选定的时钟源仍保持工作，系统和内部模块也将在 Fosc 或 Fcpu 下继续工作。空闲模式下，选定端口的电平变化、或使能唤醒功能的定时器的溢出，均将唤醒系统。可通过寄存器位 CPUM 控制是否进入空闲模式，当 CPUM=10 时，系统进入空闲模式，唤醒后 CPUM 自动恢复成 00，系统按之前的模式恢复工作。

- ✧ CPU 停止工作，程序代码暂停执行；
- ✧ 此前作为系统时钟源的振荡器仍正常工作，其他振荡器的状态取决于工作模式的配置；
- ✧ 支持唤醒功能的定时器继续原工作状态；
- ✧ 选定端口的电平变化、或使能唤醒功能的定时器的溢出，将唤醒系统；
- ✧ 由高速模式切换到空闲模式，唤醒后系统返回到高速模式；
- ✧ 由低速模式切换到空闲模式，唤醒后系统返回到低速模式；



## 6 复位

### 6.1 复位条件

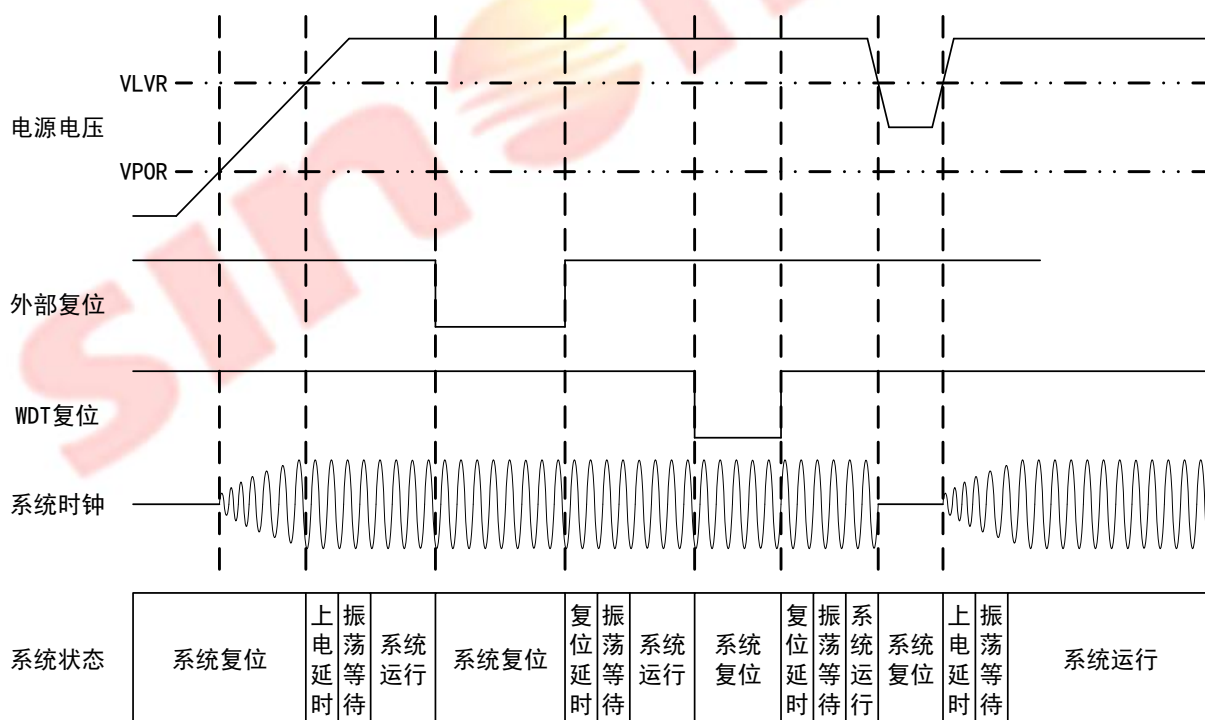
芯片共有如下几种复位方式：

- ✧ 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 **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。

## 6.4 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。

## 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 2 位端口 P0、一组 3 位端口 P4、和一组 1 位端口 P5。所有 I/O 端口均支持施密特输入，除 P14 可配置为单输入或输入/开漏输出口外，其余 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	-	P02D	-	-
R/W	-	-	-	R/W	-	R/W	-	-
初始值	-	-	-	0	-	0	-	-

BIT[4,2] P0nD – P0n 端口数据位 (n=4,2)

注：通过配置字 RSTEN 将 P04 用作复位脚时，P04D 读出恒为“1”。

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

BIT[4,1,0] P4nD – P4n 端口数据位 (n=4,1,0)

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

BIT[3] P5nD – P5n 端口数据位 (n=3)

#### 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	-	P02OE	-	-
R/W	-	-	-	R/W	-	R/W	-	-
初始值	-	-	-	0	-	0	-	-

BIT[4,2] P0nOE – P0n 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；

注：P04 配置为单输入口时，P04OE 位无效。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP4	-	-	-	P44OE	-	-	P41OE	P40OE
R/W	-	-	-	R/W	-	-	R/W	R/W
初始值	-	-	-	0	-	-	0	0

BIT[4,1,0] **P4nOE** – P4n 端口输出使能位 (n=4,1,0)

- 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
OEP5	-	-	-	-	P53OE	-	-	-
R/W	-	-	-	-	R/W	-	-	-
初始值	-	-	-	-	0	-	-	-

BIT[3] **P5nOE** – P5n 端口输出使能位 (n=3)

- 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

除 P04 外其余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
PUP0	-	-	-	P04PU	-	P02PU	-	-
R/W	-	-	-	R	-	W	-	-
初始值	-	-	-	0	-	0	-	-

BIT[4,2] **P0nPU** – P0n 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;

**注: P04 同样具有内部上拉电阻, 在端口配置为复位脚时固定有效, 配置为单输入口时固定无效, 配置为输入/开漏输出口时则在端口处于输入状态时可通过 P04PU 控制上拉电阻是否有效。**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUP4	-	-	-	P44PU	-	-	P41PU	P40PU
R/W	-	-	-	W	-	-	W	W
初始值	-	-	-	0	-	-	0	0

BIT[4,1,0] **P4nPU** – P4n 端口上拉电阻控制位 (n=4,1,0)

- 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
PUP5	-	-	-	-	P53PU	-	-	-
R/W	-	-	-	-	W	-	-	-
初始值	-	-	-	-	0	-	-	-

BIT[3] P5nPU – P5n 端口上拉电阻控制位 (n=3)

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

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）。

#### P4 端口模式设置寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P4CON	-	-	-	P44CON	-	-	P41CON	P40CON
R/W	-	-	-	R/W	-	-	R/W	R/W
初始值	-	-	-	0	-	-	0	0

BIT[4,1,0] P4nCON – P4n 端口数字功能控制位 (n=4,1,0)

0: 使能端口的数字 I/O 功能;

1: 关闭端口的数字 I/O 功能;

用作数字输入口时，P0 所有端口支持端口电平变化唤醒功能，P02/P41 可作为外部中断输入口（需配置字 DIGPS 选择端口映射）。

**注：应用端口电平变化唤醒或外部中断功能时，端口不能浮空且需外部输入电平稳定，否则易产生误触发。**

## 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，WDT 计数器溢出将复位芯片。

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，任意模式下 WDT 溢出均将复位芯片。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、在其他方式唤醒 CPU 后恢复工作。

WDT 控制寄存器 WDTCR 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
WDTCR	WDTCR7	WDTCR6	WDTCR5	WDTCR4	WDTCR3	WDTCR2	WDTCR1	WDTCR0
R/W	W	W	W	W	W	W	W	W
初始值	0	0	0	0	0	0	0	0

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

### 8.2 定时器 T0

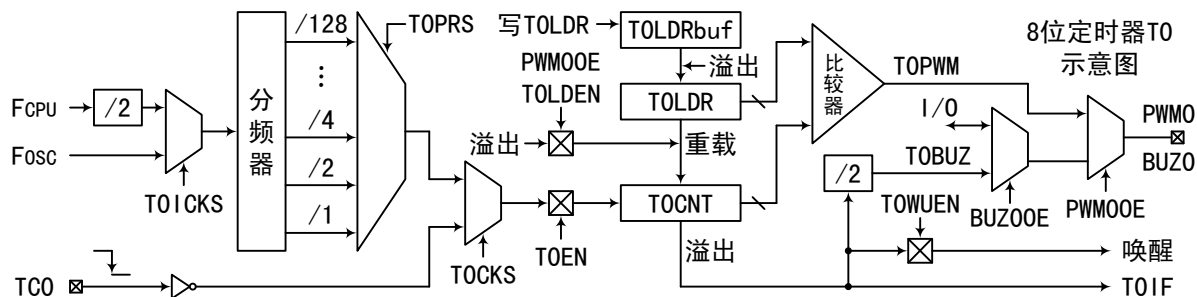
定时器 T0 为 8 位定时/计数器，包含 1 个 8 位递增计数器、可编程预分频器、控制寄存器、8 位重载/比较寄存器。

- ✧ 时钟源可选择内部 F<sub>CPU</sub>、F<sub>OSC</sub>，或外部 TC0 输入；
- ✧ 选择内部时钟时可通过预分频比设置计数频率；
- ✧ 可实现 BUZ、PWM 功能；
- ✧ 用于定时/计数/BUZ 时，可通过重载/比较寄存器控制计数周期；用于 PWM 输出时，可通过重载/比较寄存器设置 PWM 占空比；
- ✧ 支持溢出中断，可使能或关闭溢出唤醒功能；

定时器 T0，可通过寄存器位 T0CKS 和 T0ICKS 选择内部或外部时钟源，通过 T0PRS 选择内部时钟源的预分频比，外部时钟或经过预分频器后的内部时钟作为 T0 计数器 T0CNT 的计数时钟。预分频器对 F<sub>OSC</sub> 或 F<sub>CPU</sub>/2 进行 1~128 分频，对 T0CNT 的写操作将清零预分频计数器，而预分频比保持不变。

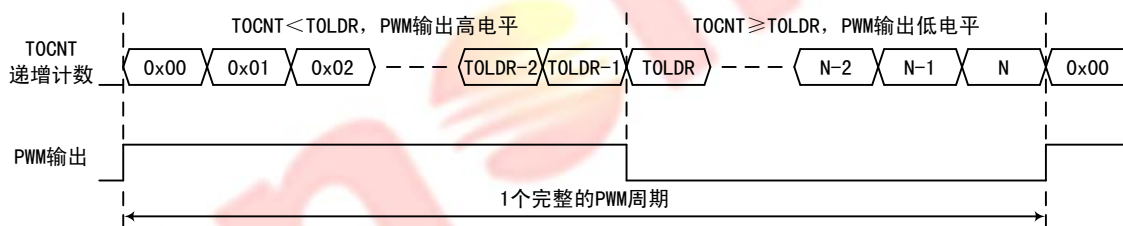
PWM0OE=0 时，T0 可实现定时/计数功能，T0CNT 递增计数至 FFH，溢出后从 0 开始重新计数；可通过 T0LDEN 使能自动重载功能，即 T0CNT 计数溢出时将自动载入 T0LDR 的值后继续递增计数。

T0LDR 配有 1 个 8 位缓冲器，写 T0LDR 操作会将数据先写入该缓冲器中，在 T0 溢出时才将缓冲器中的值自动载入 T0LDR 和 T0CNT 中。若通过 T0LDR 调整 T0 的计数周期，只从下一个周期开始有效，因此在使能 T0 前需对 T0LDR 和 T0CNT 写相同的值，以使首个计数周期正确。



PWM0OE=0 且 BUZ0EN=1 时，T0 可实现 BUZ 功能，端口 BUZ0 将输出蜂鸣器驱动信号，频率为 T0 溢出频率的 2 分频。

PWM0OE=1 时，T0 可实现 PWM 功能，端口 PWM0 将输出 PWM 波形。每个 PWM 周期内，T0CNT 从 0 开始递增计数，并与重载/比较寄存器 T0LDR 的值比较，当计数到与 T0LDR 相等时，PWM0 信号变为低电平；当计数溢出时，PWM0 信号变为高电平。写 T0LDR 操作仅将数据写入缓冲器，T0 溢出时才载入 T0LDR 中，所以通过 T0LDR 调整 PWM0 的占空比，只在下一个周期开始有效。



PWM0OE=1 时，T0LDEN 和 BUZ0EN 位用于设置 T0CNT 的最大计数值 N，即设置 PWM0 的计数周期（详见寄存器位说明）。

T0 选择外部时钟时，端口复用为数字输入端口 TC0，其外部中断及电平变化唤醒功能将被禁止。T0 使能 PWM 或 BUZ 功能后，端口复用为数字输出端口 PWM0/BUZ0，其通用 I/O 功能将被禁止。

系统在高/低速模式之间切换时，T0 的内部时钟源会同时在系统高/低频时钟源之间自动切换。休眠模式下，T0 暂停工作。空闲模式下，T0 继续原有工作状态，溢出时中断标志位将被置 1，T0WUEN=1 时溢出还可唤醒 CPU。

## 定时器控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMRCR	-	-	-	-	T1ICKS	T0ICKS	T0WUEN	-
R/W	-	-	-	-	R/W	R/W	R/W	-
初始值	-	-	-	-	0	0	0	-

BIT[3] **T1ICKS** – T1 内部时钟源选择位

BIT[2] **T0ICKS** – T0 内部时钟源选择位

0: T0 内部时钟源为 F<sub>CPU</sub>/2;

1: T0 内部时钟源为 F<sub>OSC</sub>;

BIT[1] **T0WUEN** – T0 溢出唤醒使能位

0: 关闭 T0 溢出唤醒功能;

1: 使能 T0 溢出唤醒功能;

### 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>	T0EN	T0PRS2	T0PRS1	T0PRS0	T0CKS	TOLDEN	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] **T0EN** – 定时器 T0 使能位

0: 关闭定时器 T0;

1: 开启定时器 T0;

BIT[6:4] **T0PRS[2:0]** – T0 预分频比选择位

<b>T0PRS[2:0]</b>	<b>T0 时钟预分频比</b>
000	1 : 128
001	1 : 64
010	1 : 32
011	1 : 16
100	1 : 8
101	1 : 4
110	1 : 2
111	1 : 1

BIT[3] **T0CKS** – T0 时钟源选择位

0: T0 时钟源为内部时钟源;

1: T0 时钟源为外部 TC0 输入下降沿;

BIT[2] **TOLDEN** – 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[2,1] **T0LDEN, BUZ0OE** – PWM0 周期选择位（仅 PWM0OE=1 时有效）

T0LDEN : BUZ0OE	N	T0CNT 范围	T0CNT/T0LDR 有效值（二进制）	PWM0 周期
00	255	0~255	0000 0000 ~ 1111 1111	256 个时钟周期
01	63	0~63	xx00 0000 ~ xx11 1111	64 个时钟周期
10	31	0~31	xxx0 0000 ~ xxx1 1111	32 个时钟周期
11	15	0~15	xxxx 0000 ~ xxxx 1111	16 个时钟周期

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
T0CNT	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	0	0	0	0	0	0	0	0

BIT[7:0] **T0CNT[7:0]** – T0 计数器，为可读写的递增计数器

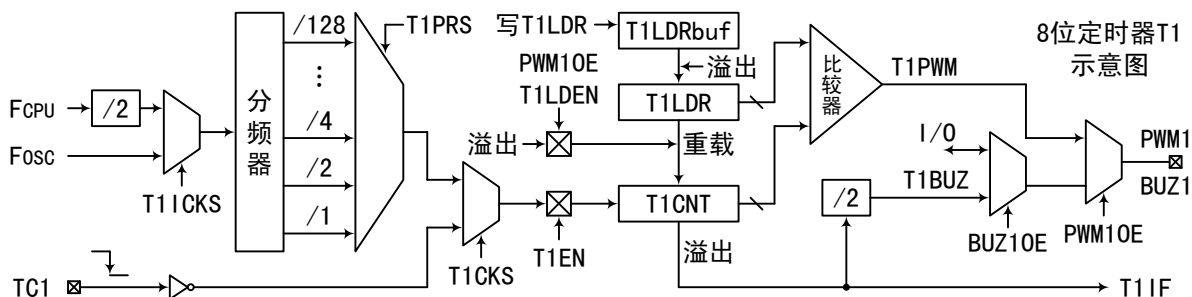
### 定时器 T0 重载/比较寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0LDR	T0LDR7	T0LDR6	T0LDR5	T0LDR4	T0LDR3	T0LDR2	T0LDR1	T0LDR0
R/W	W	W	W	W	W	W	W	W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **T0LDR[7:0]** – T0 重载/比较寄存器，用于设置 T0 的计数周期、或设置 PWM0 的占空比

## 8.3 定时器 T1

定时器 T1 为 8 位定时/计数器，包含 1 个 8 位递增计数器、可编程预分频器、控制寄存器、8 位重载/比较寄存器。除不支持溢出唤醒外，T1 的定时/计数/PWM/BUZ 功能与 T0 完全相同。





### 定时器控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>TMRCR</b>	-	-	-	-	T1ICKS	T0ICKS	T0WUEN	-
<b>R/W</b>	-	-	-	-	R/W	R/W	R/W	-
<b>初始值</b>	-	-	-	-	0	0	0	-

BIT[3] **T1ICKS** – T1 内部时钟源选择位

0: T1 内部时钟源为 F<sub>CPU</sub>/2;

1: T1 内部时钟源为 F<sub>OSC</sub>;

BIT[2] **T0ICKS** – T0 内部时钟源选择位

BIT[1] **T0WUEN** – T0 溢出唤醒使能位

### 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>	T1EN	T1PRS2	T1PRS1	T1PRS0	T1CKS	T1LDEN	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] **T1EN** – 定时器 T1 使能位

0: 关闭定时器 T1;

1: 开启定时器 T1;

BIT[6:4] **T1PRS[2:0]** – T1 预分频比选择位

<b>T1PRS[2:0]</b>	<b>T1 时钟预分频比</b>
000	1 : 128
001	1 : 64
010	1 : 32
011	1 : 16
100	1 : 8
101	1 : 4
110	1 : 2
111	1 : 1

BIT[3] **T1CKS** – T1 时钟源选择位

0: T1 时钟源为内部时钟源;

1: T1 时钟源为外部 TC1 输入下降沿;

BIT[2] **T1LDEN** – 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[2,1] **T1LDEN, BUZ1OE** – PWM1 周期选择位（仅 PWM1OE=1 时有效）

T1LDEN : BUZ1OE	N	T1CNT 范围	T1CNT/T1LDR 有效值（二进制）	PWM1 周期
00	255	0~255	0000 0000 ~ 1111 1111	256 个时钟周期
01	63	0~63	xx00 0000 ~ xx11 1111	64 个时钟周期
10	31	0~31	xxx0 0000 ~ xxx1 1111	32 个时钟周期
11	15	0~15	xxxx 0000 ~ xxxx 1111	16 个时钟周期

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>T1CNT</b>	T1CNT7	T1CNT6	T1CNT5	T1CNT4	T1CNT3	T1CNT2	T1CNT1	T1CNT0
<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] **T1CNT[7:0]** – T1 计数器，为可读写的递增计数器

### 定时器 T1 重载/比较寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T1LDR</b>	T1LDR7	T1LDR6	T1LDR5	T1LDR4	T1LDR3	T1LDR2	T1LDR1	T1LDR0
<b>R/W</b>	W	W	W	W	W	W	W	W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **T1LDR[7:0]** – T1 重载/比较寄存器，用于设置 T1 的计数周期、或设置 PWM1 的占空比

## 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。

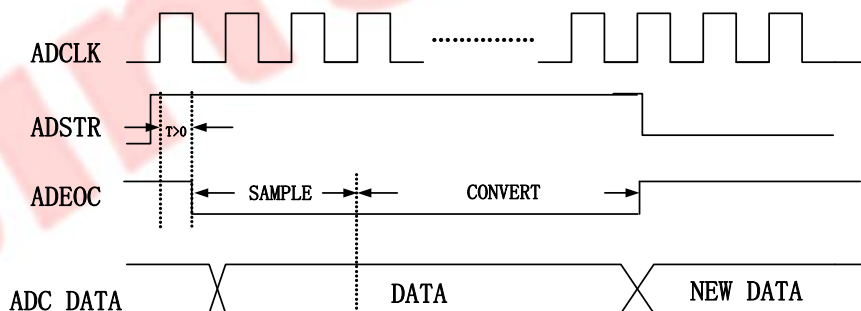
- ✧ 3 路外部通道：AN0/AN1/AN4；1 路内部通道：VDD/4；
- ✧ 参考电压可选：VDD、内部参考电压  $V_{IR}$  (2V/3V/4V)、外部参考电压  $V_{ER}$  ( $V_{ER1}$  输入)；
- ✧ ADC 时钟：F<sub>CPU</sub> 的 4/8/32/64 分频；
- ✧ 支持零点或顶点校准；

ADC 模块可通过寄存器位 ADEN 开启，通过 ADCHS 选择转换的模拟通道，通过 GCHS 控制转换输入的开/关，通过 ADCKS 选择转换速度。ADSTR 为转换启动控制位，ADEOC 为转换状态标志位。ADSTR 位写 1 将启动模数转换；转换完成后 ADSTR 自动清 0、ADEOC 自动置 1，结果存入 ADB/ADR 中，同时中断标志 ADIF 置 1 触发 ADC 中断。

ADEN 使能且 GCHS 开启输入通道后，ADCHS 选定的通道端口将自动转为模拟端口，VEREN 使能外部参考电压输入时，端口将自动转为模拟端口。为保证转换效果及防止漏电流产生，应将端口设为数字输入口且关闭内部上拉电阻，再通过 P4CON 寄存器关闭端口的数字 I/O 功能。

ADC 的采样 (SAMPLE) 时间固定为 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 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，ADB/ADR 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、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) 设置相应端口为输入端口，关闭端口的内部上拉电阻；
- (2) 通过 P4CON 寄存器，关闭相应端口的数字 I/O 功能；
- (3) 设置 ADCKS，选择适当的 ADC 转换时钟；
- (4) ADEN 置 1，使能 ADC 模块；GCHS 置 1，开启转换输入通道；
- (5) 设置 ADCHS，选择 ADC 转换通道；
- (6) 延时等待电路稳定后，ADSTR 写 1，启动 AD 转换；
- (7) 等待 ADEOC 硬件置 1（或利用 ADC 中断）；
- (8) 读取 ADC 转换结果（ADB、ADR）；
- (9) 重复执行（5）~（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>ADM</b>	ADEN	ADSTR	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] **ADEN** – ADC 使能位

- 0: 关闭 ADC;  
1: 开启 ADC;

BIT[6] **ADSTR** – ADC 启动控制位

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

BIT[5] **ADEOC** – ADC 转换状态标志位

- 0: AD 转换中;  
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 输入通道选择位

<b>ADCHS[2:0]</b>	<b>ADC 输入通道</b>
000	AN0
001	AN1

100	AN4
101	VDD/4
其他	禁用

### 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>VREFCR</b>	VEREN	-	-	-	-	-	ADVRS1	ADVRS0
<b>R/W</b>	R/W	-	-	-	-	-	R/W	R/W
<b>初始值</b>	0	-	-	-	-	-	0	0

BIT[7] **VEREN** – ADC 外部参考电压使能位

0: ADC 参考电压由 ADVRS[1:0]决定;

1: ADC 参考电压为外部参考电压  $V_{ER}$  (端口  $VERI$  的输入电压);

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

<b>ADVRS[1:0]</b>	<b>ADC 参考电压</b>
00	内部 2.0V
01	内部 3.0V
10	内部 4.0V
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>ADB</b>	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4
<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] **ADB[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>ADR</b>	-	ADCKS1	-	ADCKS0	ADB3	ADB2	ADB1	ADB0
<b>R/W</b>	-	R/W	-	R/W	R	R	R	R
<b>初始值</b>	-	0	-	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 <math>F_{ADC}</math></b>
00	$F_{CPU}/64$
01	$F_{CPU}/32$
10	$F_{CPU}/4$
11	$F_{CPU}/8$

BIT[3:0] **ADB[3:0]** – ADC 转换结果低 4 位

## ADC 校准寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>ADT</b>	ADTS1	ADTS0	-	ADTR4	ADTR3	ADTR2	ADTR1	ADTR0
<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		U	U	U	U	U

BIT[7:6] **ADTS[1:0]** – ADC 校准模式控制位

<b>ADTS[1:0]</b>	<b>ADC 校准模式控制</b>
00	正常模式
01	
10	零点校准, GND 为 ADC 输入
11	顶点校准, VDD 为 ADC 输入

BIT[4] **ADTR[4]** – ADC 校准方向选择位（复位初始值为出厂设定值）

- 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] **ADTR[3:0]** – ADC 校准修调电压选择位（复位初始值为出厂设定值）

<b>ADTR[3:0]</b>	<b>修调电压 (典型值)</b>
0000	0mV
0001	1mV
0010	2mV
...	...
1110	14mV
1111	15mV

## 9.4 ADC 校准流程

零点偏移校准流程:

- (1) ADTS[1:0]设为零点校准（即选择 GND 为 ADC 输入），设置 ADTR[4:0]=00H;
- (2) 设置 ADC 时钟、参考电压, ADEN 和 GCHS 置 1, ADSTR 置 1 开始 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) ADTR[3:0]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 ADTR[3:0]=0FH 后跳至 (6);
- (4) 设置 ADTR[4:0]=1FH, 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) ADTR[3:0]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 ADTR[3:0]=00H 后跳至 (6);

- (6) ADTR[4:0]中的值即为零点偏移最佳修调电压，校准流程结束，后续 ADC 工作时直接使用，不需要再次修调。
- (7) ADTS[1:0]清 0，恢复为正常模式。

顶点偏移校准流程：

- (1) ADTS[1:0]设为顶点校准（即选择 VDD 为 ADC 输入），设置 ADTR[4:0]=10H；
- (2) 设置 ADC 时钟，设置 VDD 为参考电压，ADEN 和 GCHS 置 1，ADSTR 置 1 开始 ADC 转换：
  - ✧ 若 ADC 结果为 0FFFH，则执行（4）；
  - ✧ 若 ADC 结果不为 0FFFH，则执行（3）；
- (3) ADTR[3:0]加 1 后进行 ADC 转换：
  - ✧ 若 ADC 结果为 0FFFH，则跳至（6）；
  - ✧ 若 ADC 结果不为 0FFFH，则循环执行（3），直到结果为 0FFFH 或 ADTR[3:0]=0FH 后跳至（6）；
- (4) 设置 ADTR[4:0]=0FH，进行 ADC 转换：
  - ✧ 若 ADC 结果为 0FFFH，则跳至（6）；
  - ✧ 若 ADC 结果不为 0FFFH，则执行（5）；
- (5) ADTR[3:0]减 1 后进行 ADC 转换：
  - ✧ 若 ADC 结果为 0FFFH，则跳至（6）；
  - ✧ 若 ADC 结果不为 0FFFH，则循环执行（5），直到结果为 0FFFH 或 ADTR[3:0]=00H 后跳至（6）；
- (6) ADTR[4:0]中的值即为顶点偏移最佳修调电压，校准流程结束，后续 ADC 工作时直接使用，不需要再次修调。
- (7) ADTS[1:0]清 0，恢复为正常模式。

## 10 中断

芯片的中断源包括外部中断（INT0~INT1）、定时器中断（T0~T1）、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
EINTCR	-	-	-	MINT01	MINT00	-	-	-
R/W	-	-	-	R/W	R/W	-	-	-
初始值	-	-	-	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=0-1）在计数溢出时触发定时器中断，中断标志 TnIF（n=0-1）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=0-1）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	-	-	-	INT1IE	INT0IE
<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[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[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	-	-	-	INT1IF	INT0IF
<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[7] **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[6]      **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[5]      **T0IF** – 定时器 T0 中断标志位  
0: 未触发定时器 T0 中断;  
1: 已触发定时器 T0 中断, 需软件清 0;
- BIT[1]      **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[0]      **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;
- 

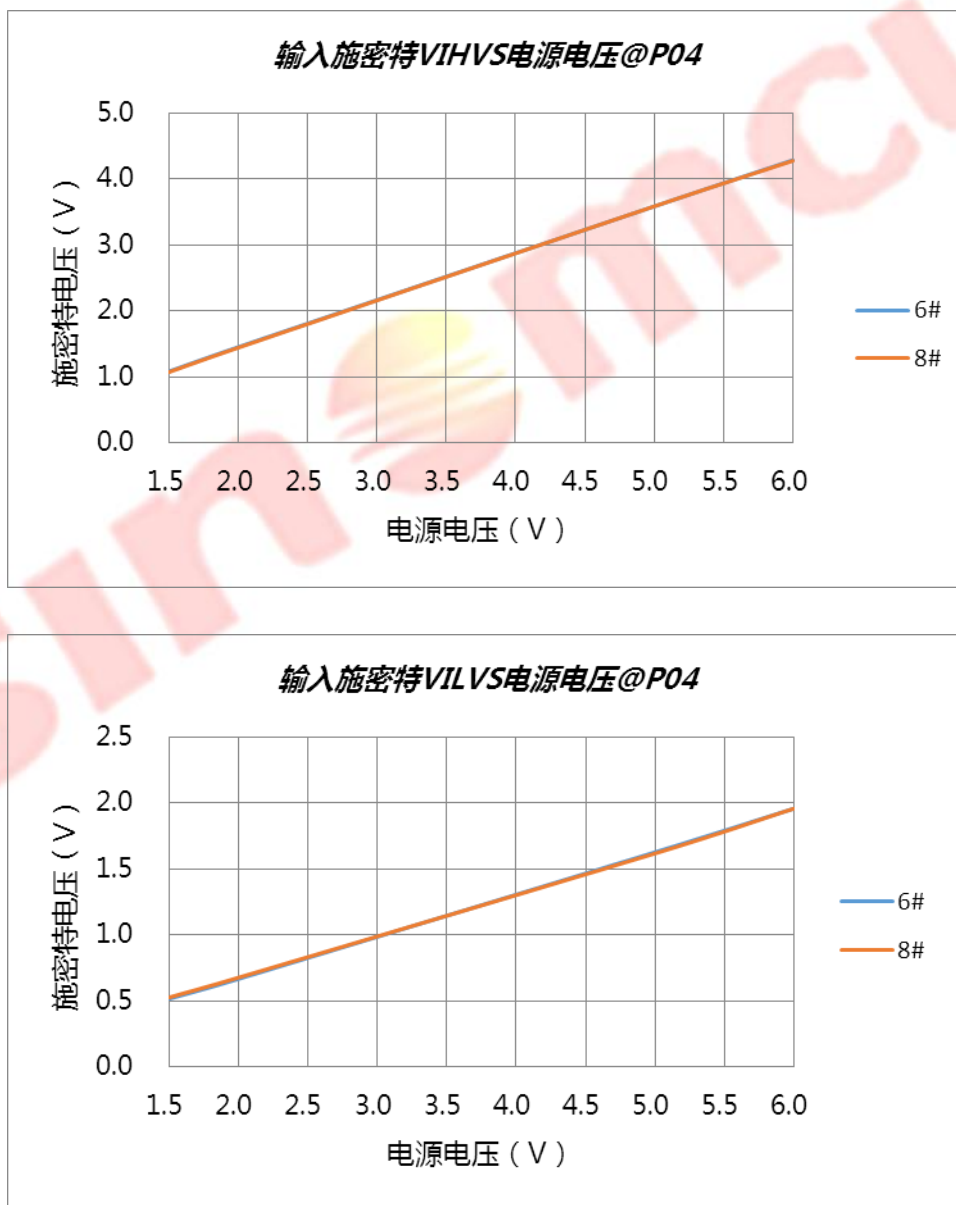
## 11 特性曲线

注：

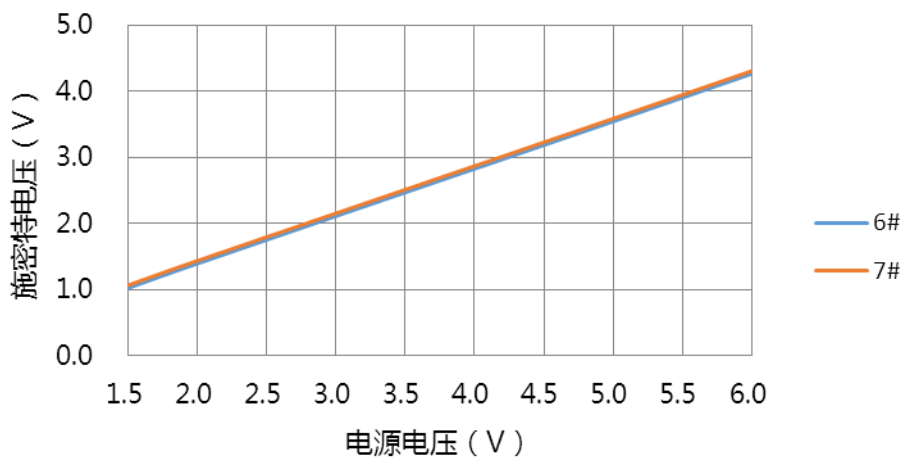
- 1、特性曲线图中数据均来自抽样实测，仅作为应用参考，部分数据因生产工艺偏差，可能与实际芯片不符；为保证芯片能正常工作，请确保其工作条件符合电气特性参数说明；
- 2、若图文中无特别说明，则电压特性曲线的温度条件为  $T=25^{\circ}\text{C}$ ，温度特性曲线的电压条件为  $V_{DD}=5\text{V}$ ；

### 11.1 I/O 特性

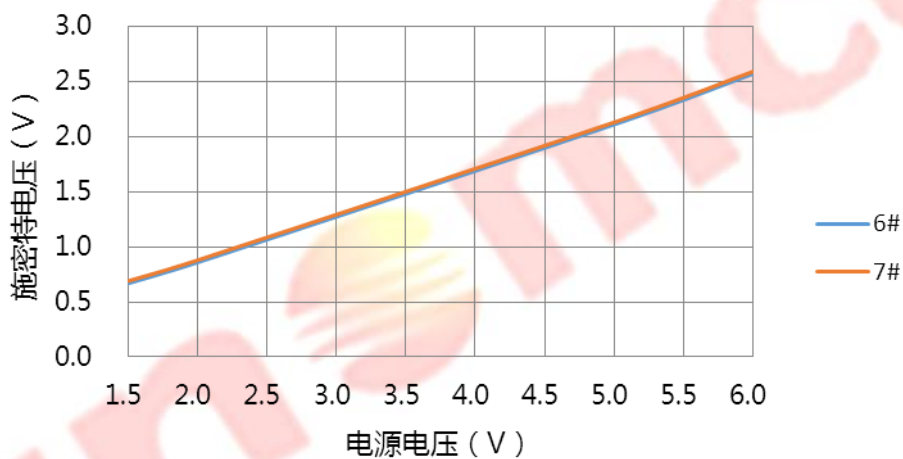
#### 输入 SMT 阈值电压 VS 电源电压



输入施密特VIHVS电源电压@P53

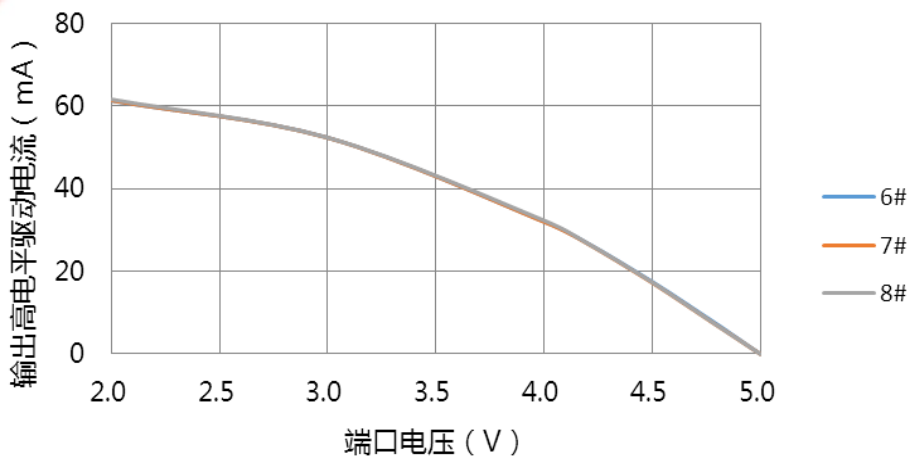


输入施密特VILVS电源电压@P53

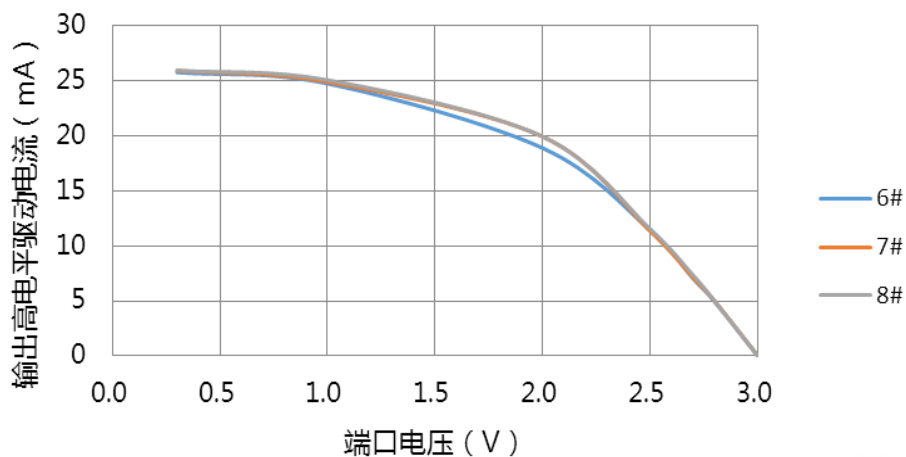


#### I/O 输出 驱动电流 VS 端口电压

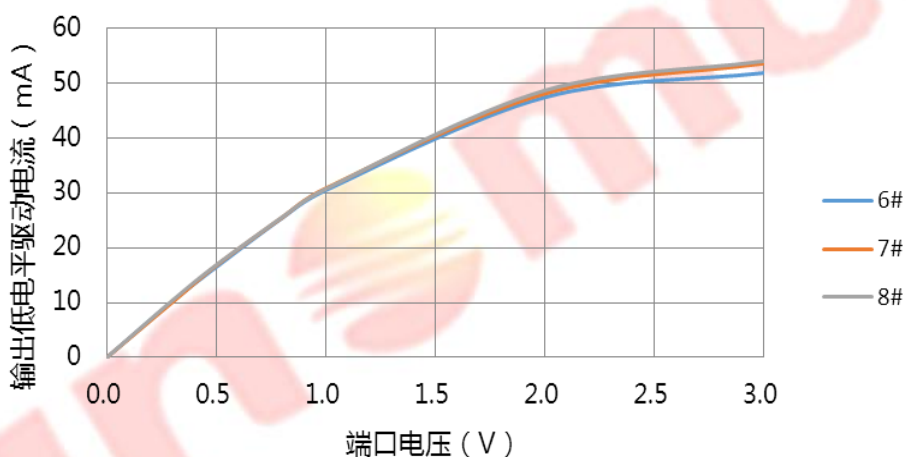
输出高电平驱动电流VS端口电压@P40@VDD=5V



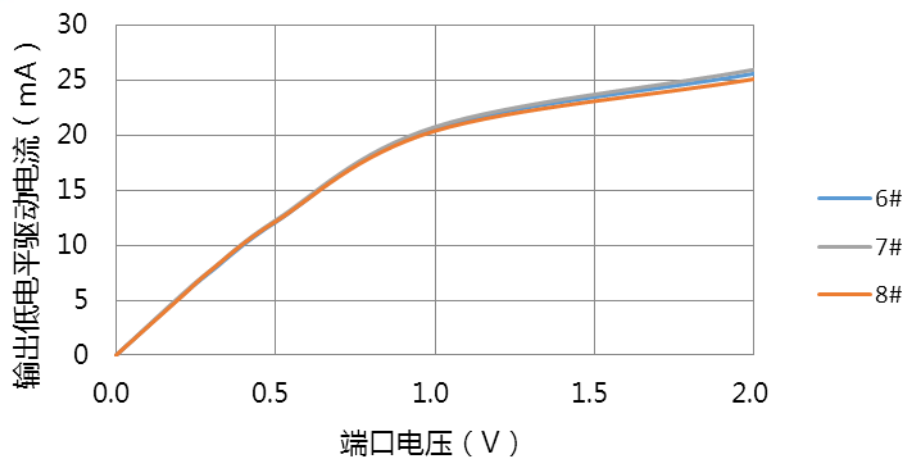
输出高电平驱动电流VS端口电压@P40@VDD=3V



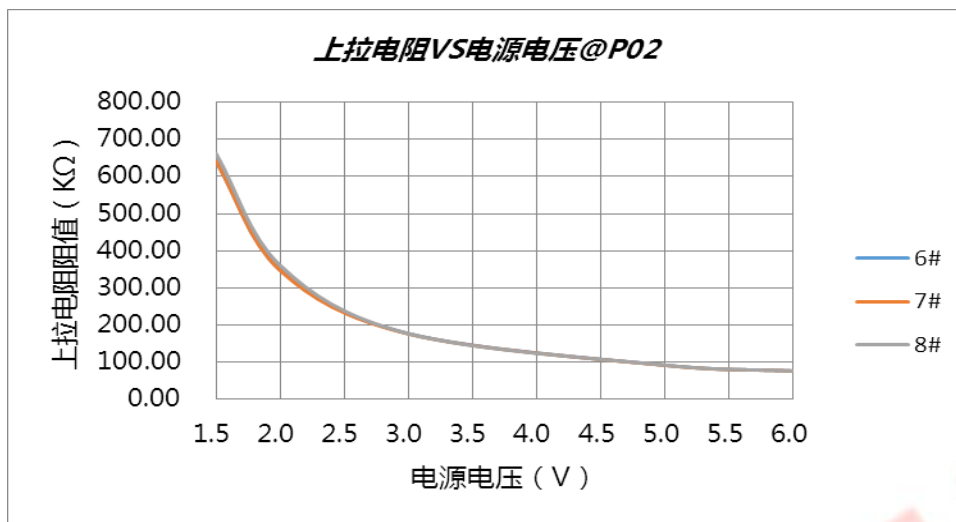
输出低电平驱动电流VS端口电压@P40@VDD=5V



输出低电平驱动电流VS端口电压@P40@VDD=3V

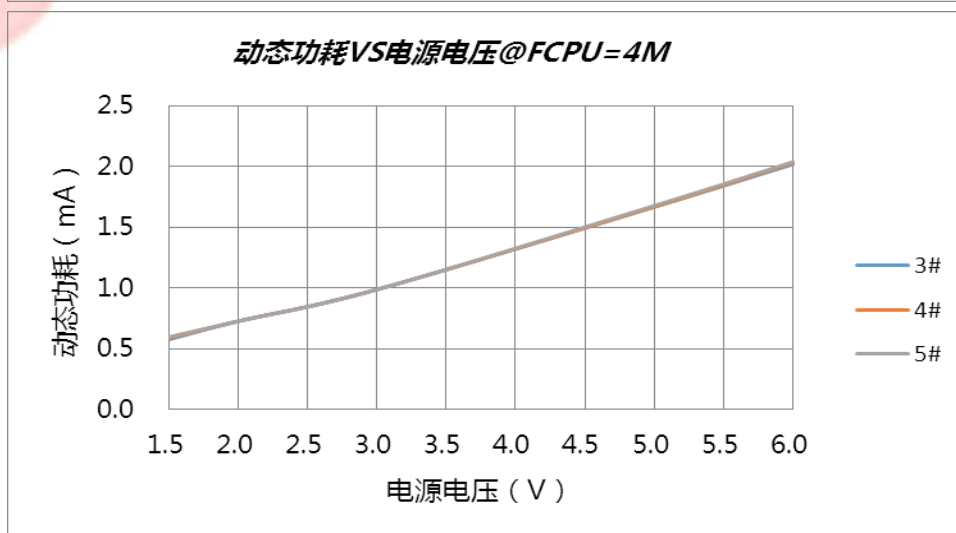
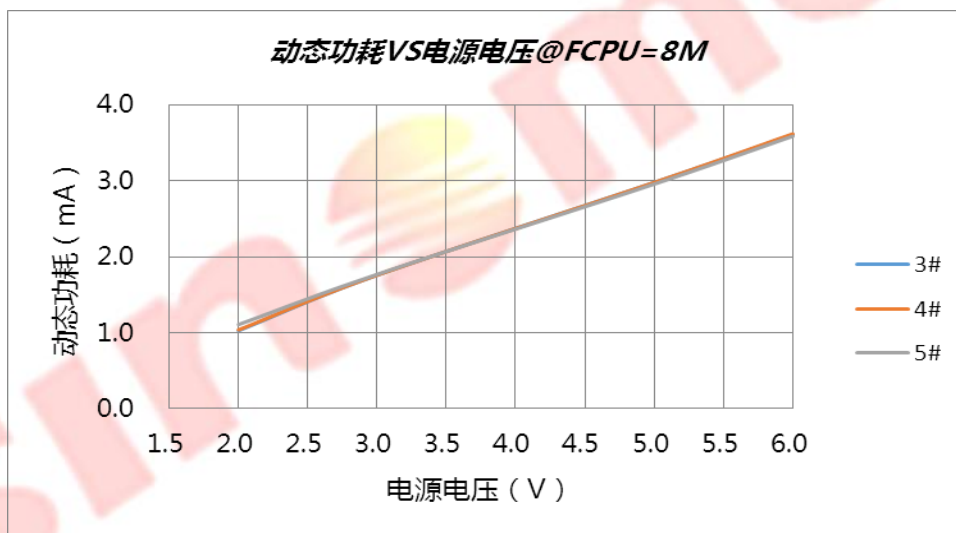


## 上拉电阻值 VS 电源电压

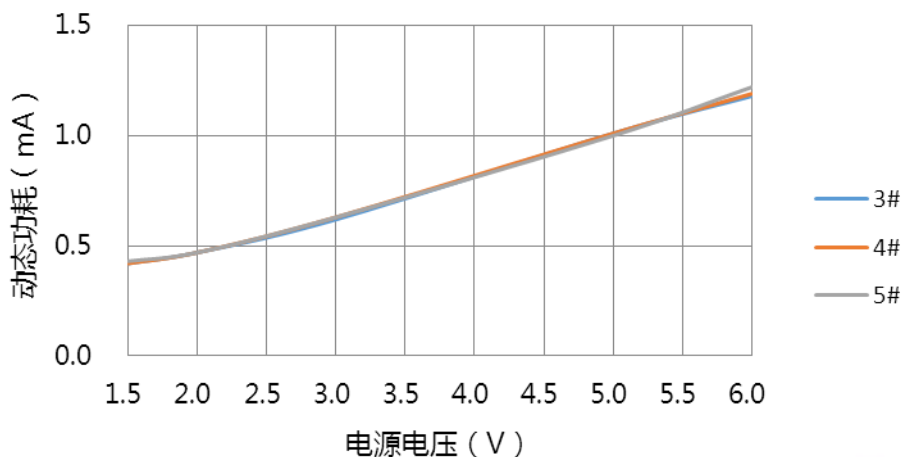


## 11.2 功耗特性

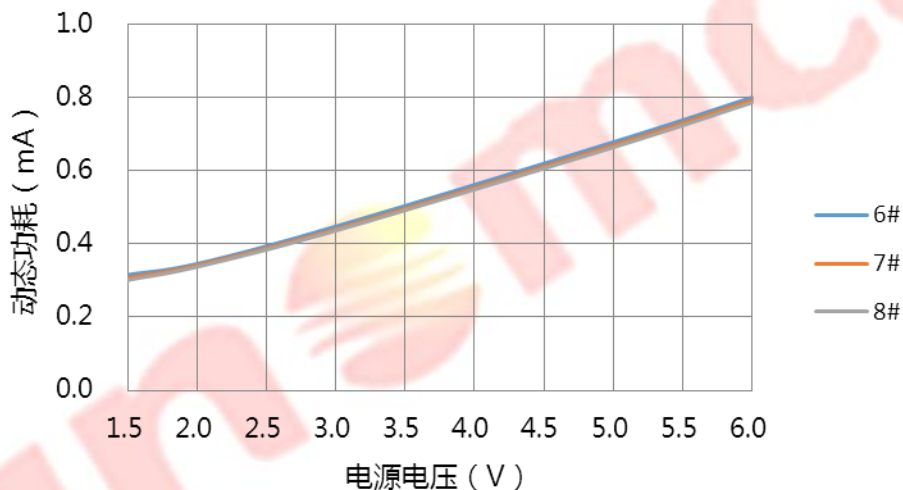
### 运行模式 功耗 VS 电源电压



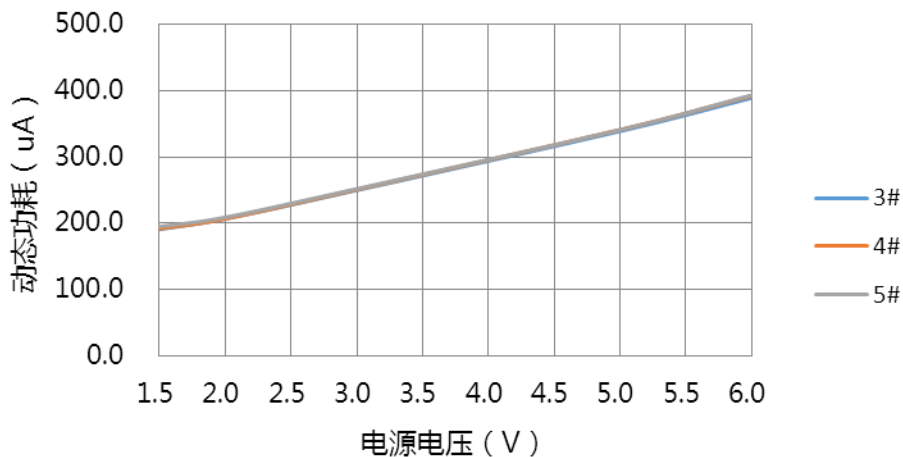
动态功耗VS电源电压@FCPU=2M



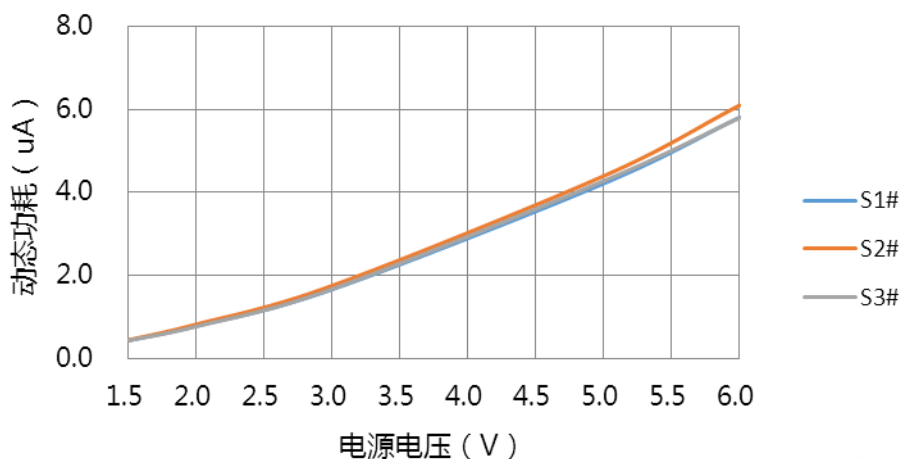
动态功耗VS电源电压@FCPU=1M



动态功耗VS电源电压@FCPU=8K@HIRC开

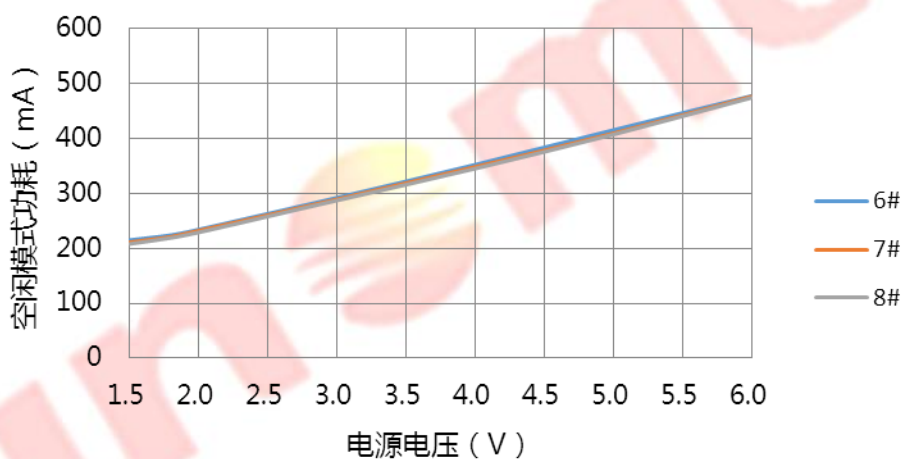


动态功耗VS电源电压@FCPU=8K@HIRC关

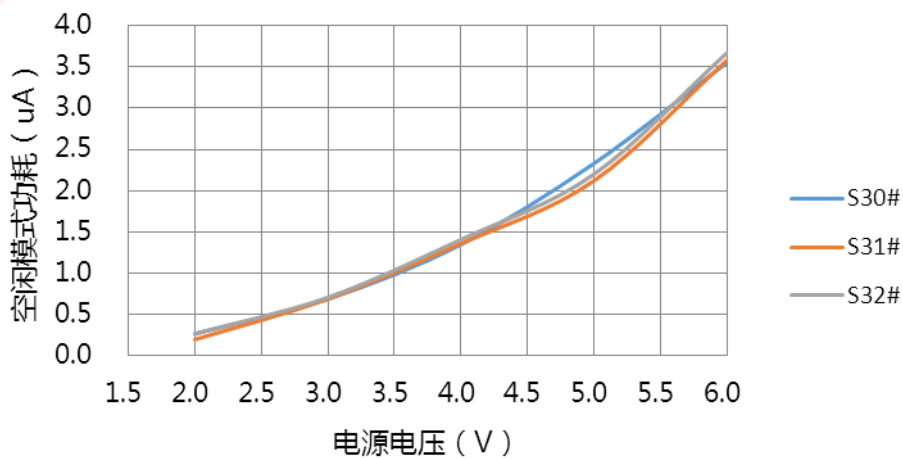


空闲模式 功耗 VS 电源电压/温度

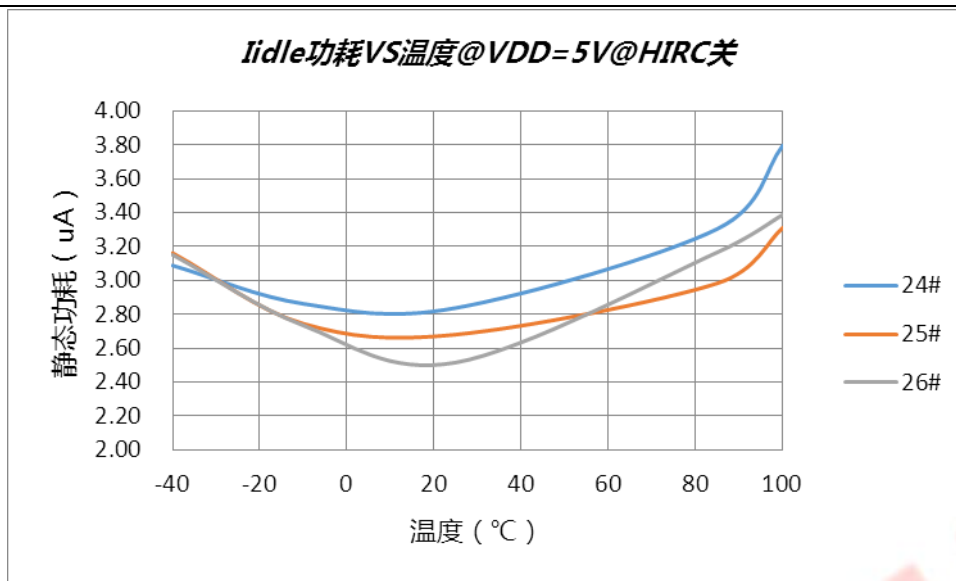
idle功耗VS电源电压@HIRC开



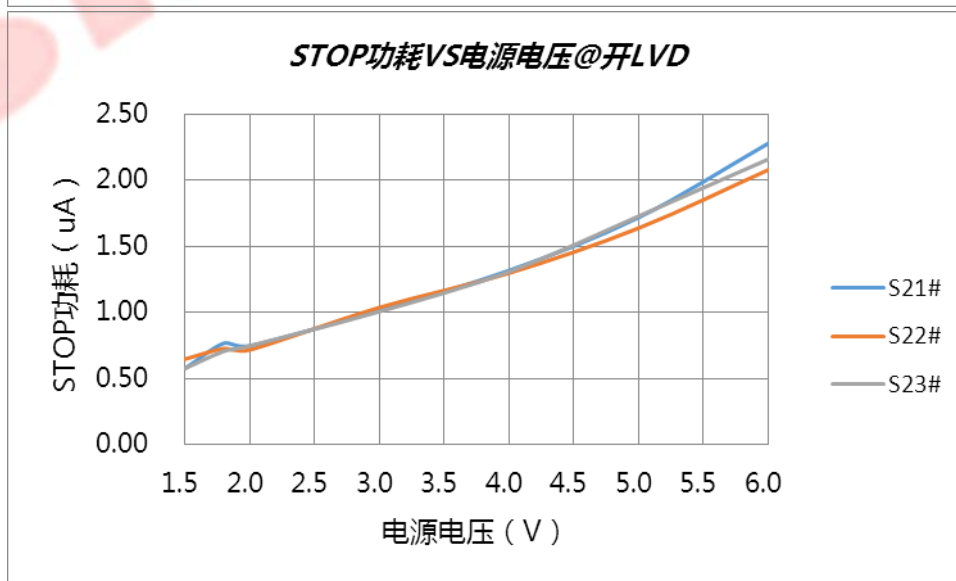
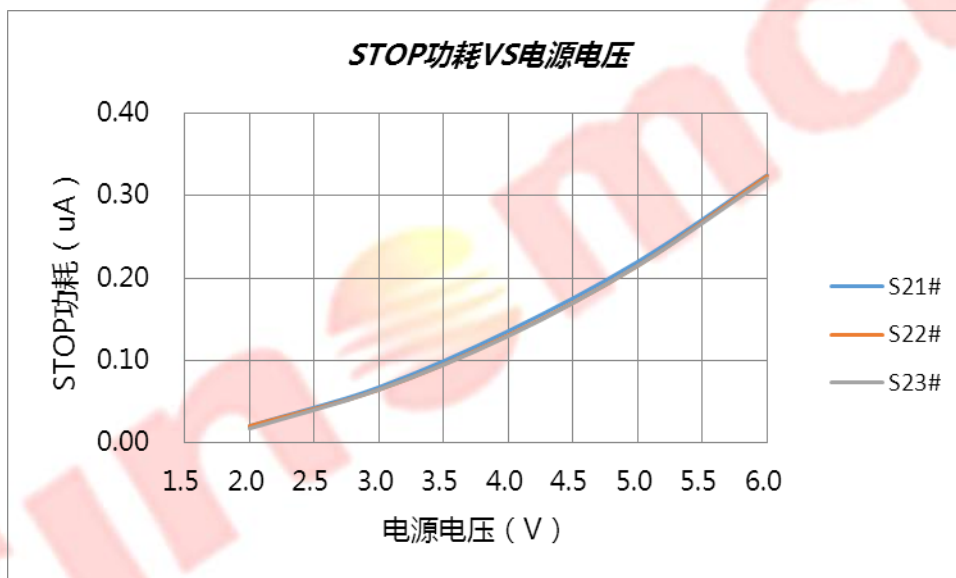
idle功耗VS电源电压@HIRC关

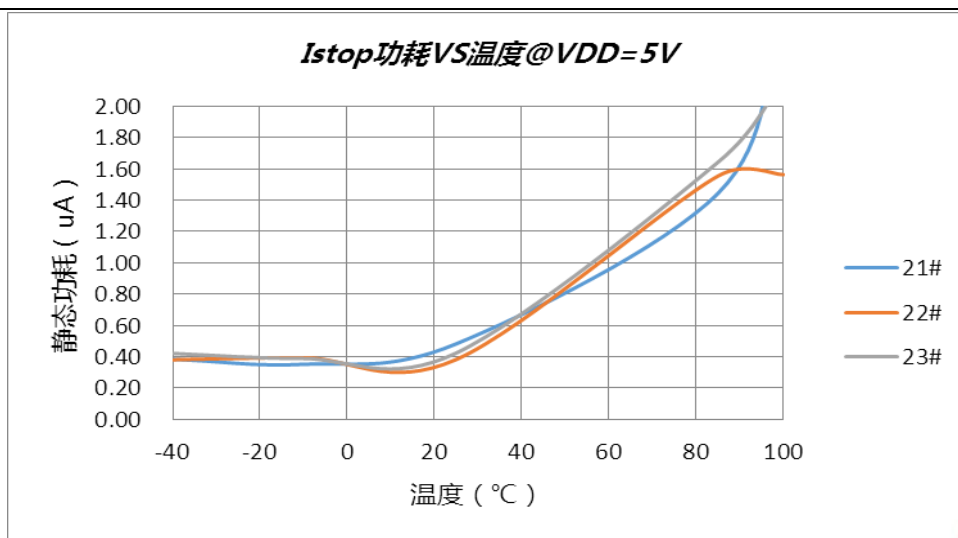






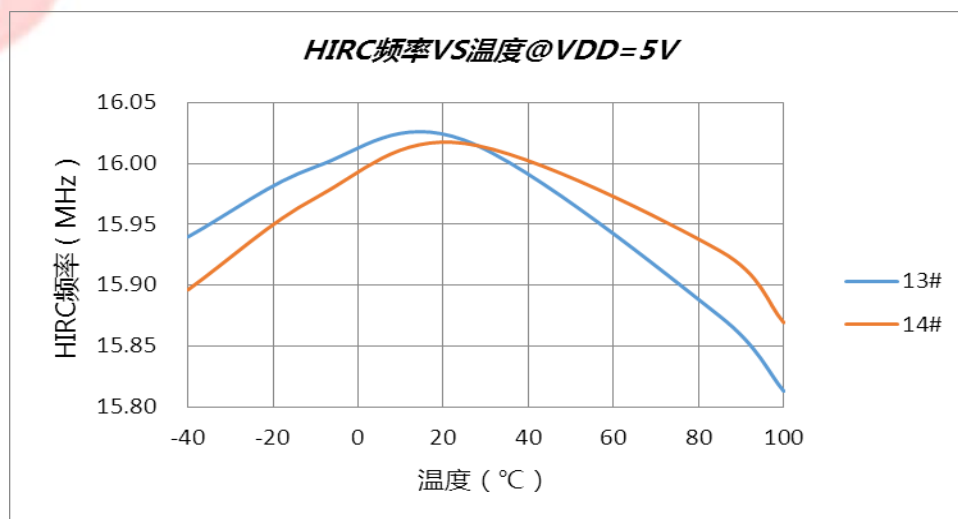
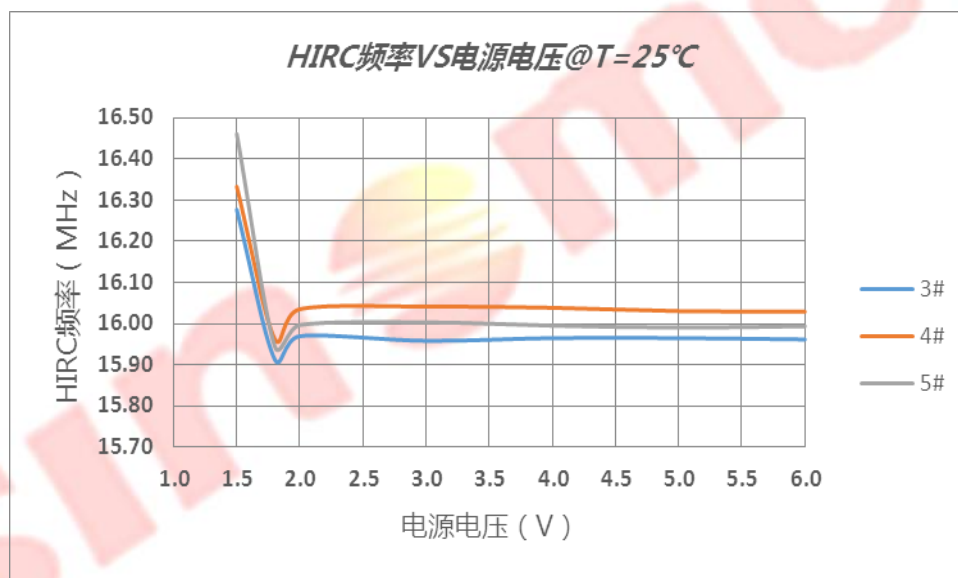
**休眠模式 功耗 VS 电源电压/温度**



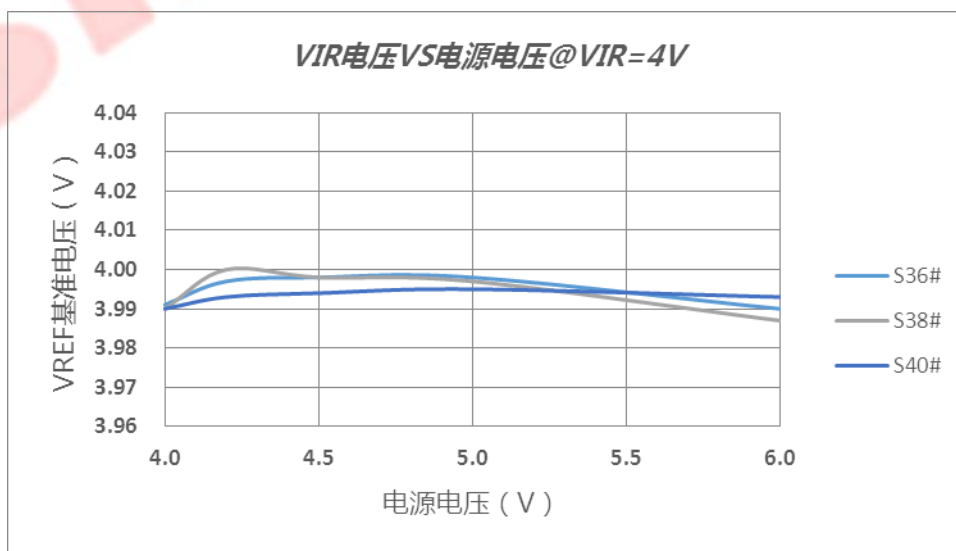
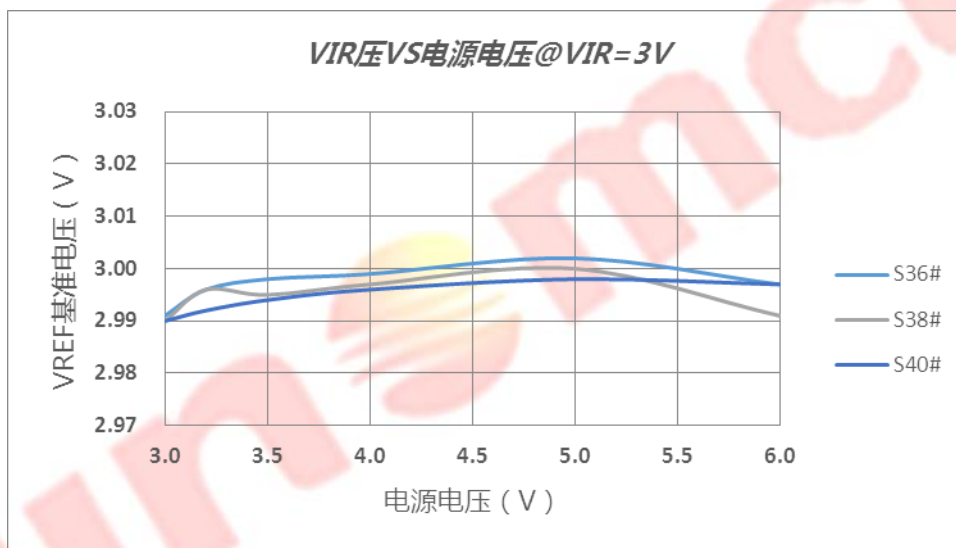
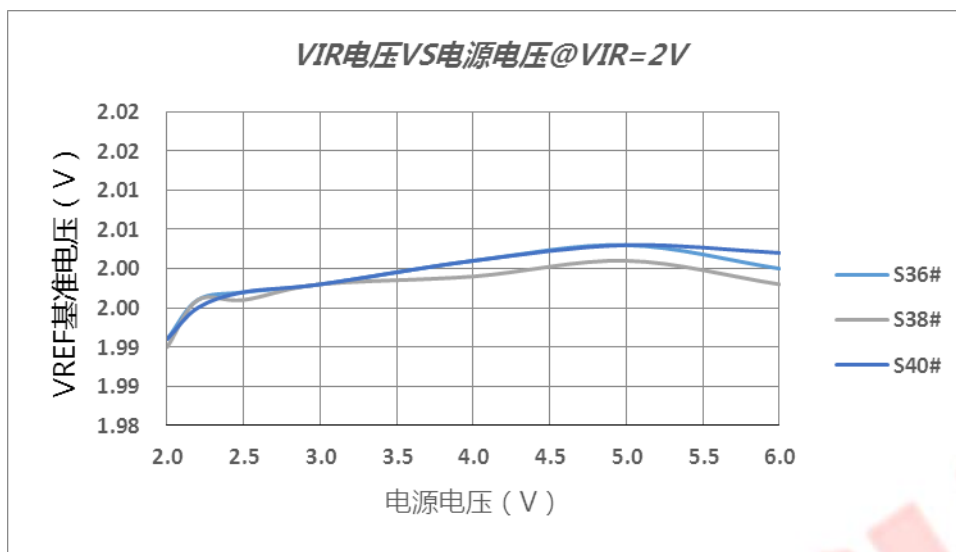


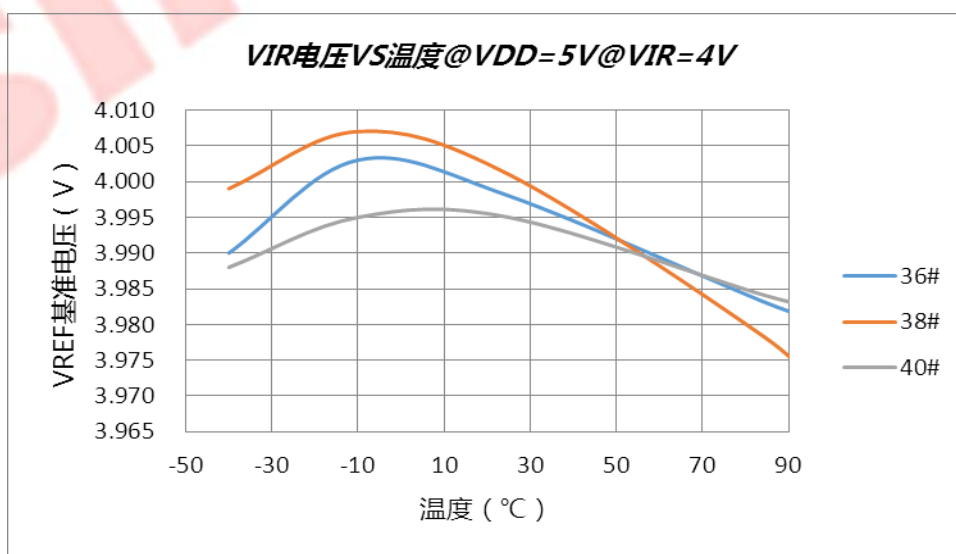
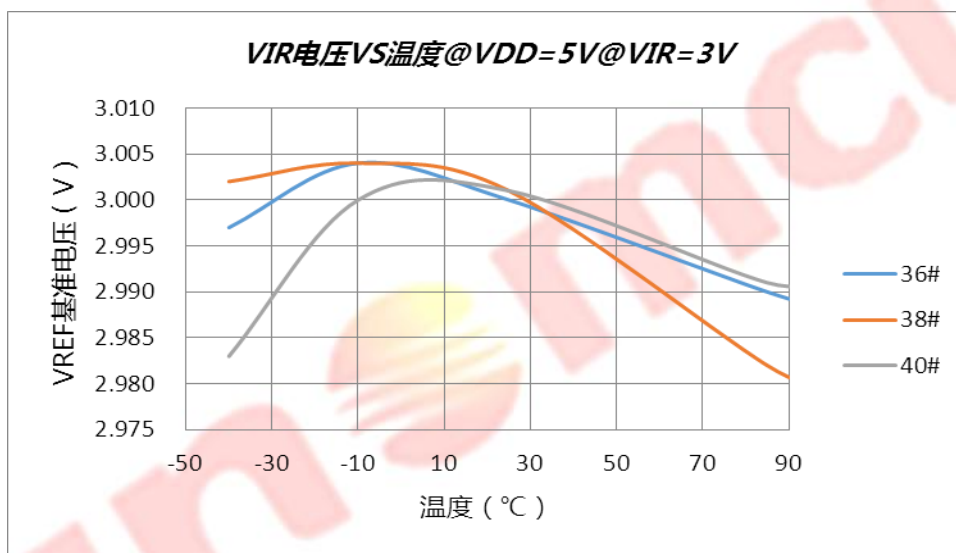
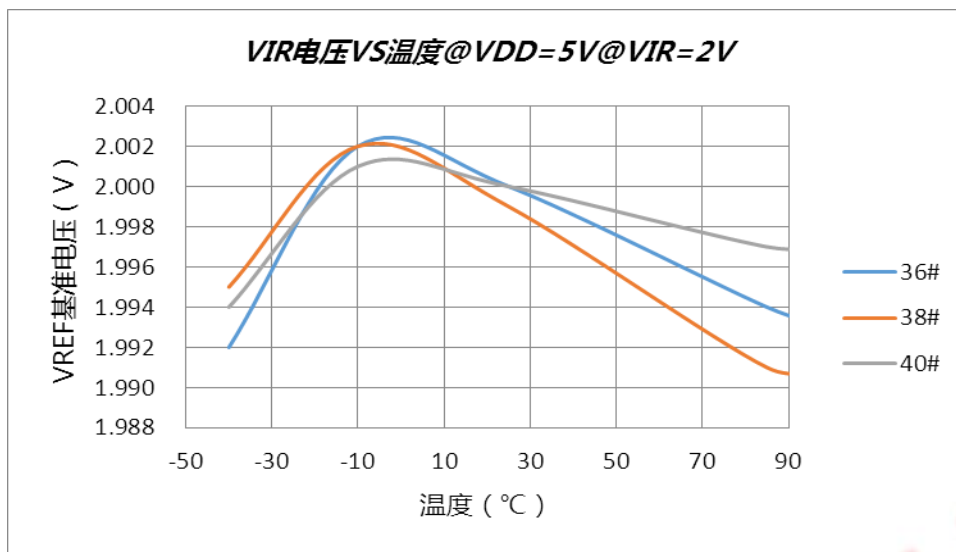
### 11.3 模拟电路特性

#### HIRC 频率 VS 电源电压/温度



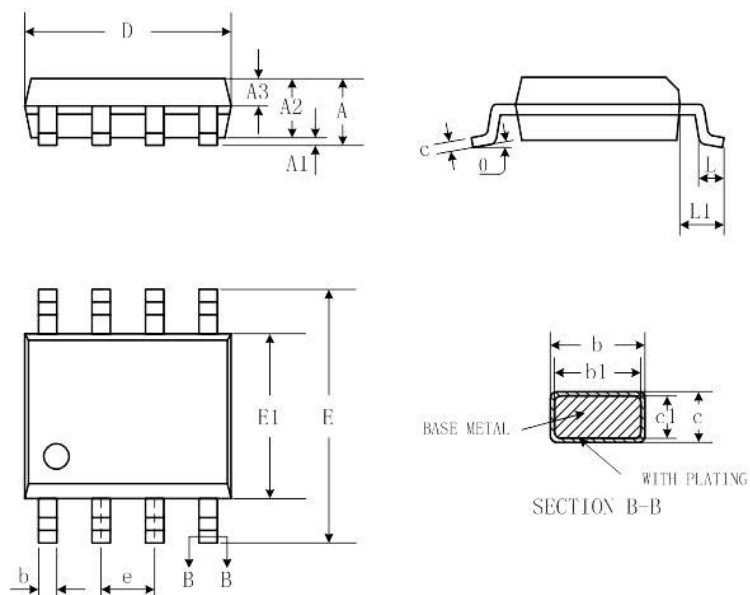
**VIR 电压 VS 电源电压/温度**





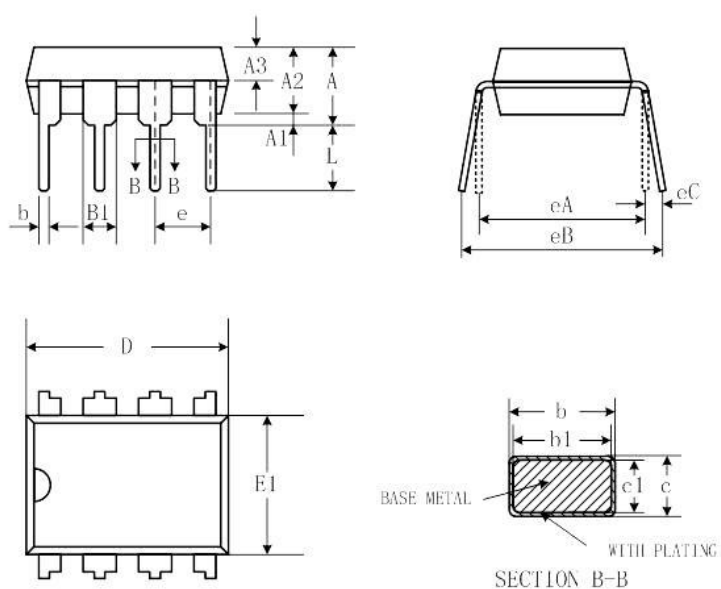
## 12 封装尺寸

### 12.1 SOP8



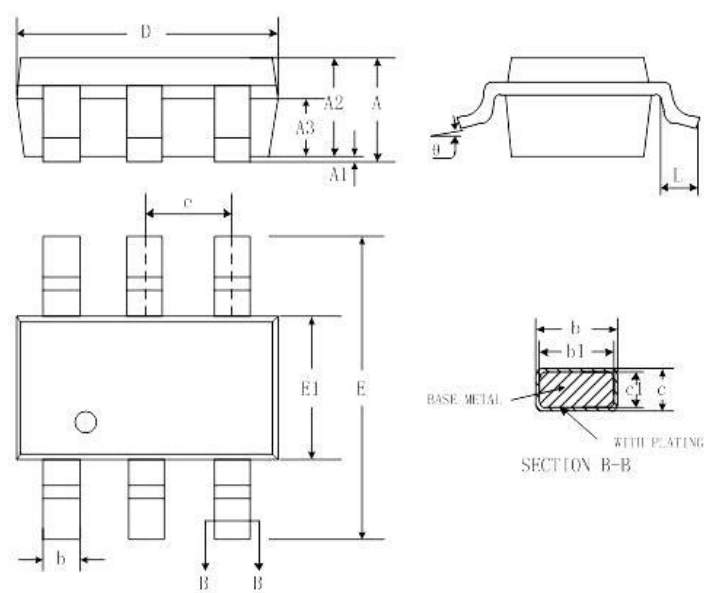
SYMBOL	MILLIMETER		
	MIN	TYP	MAX
A	-	-	1.77
A1	0.08	0.18	0.28
A2	1.20	1.40	1.60
A3	0.55	0.65	0.75
b	0.39	-	0.48
b1	0.38	0.41	0.43
c	0.21	-	0.26
c1	0.19	0.20	0.21
D	4.70	4.90	5.10
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27BSC		
L	0.50	0.65	0.80
L1	1.05BSC		
θ	0	-	8°

### 12.2 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	-	-

12.3 SOT23-6



SYMBOL	MILLIMETER		
	MIN	TYP	MAX
A	-	-	1.35
A1	0.04	-	0.15
A2	1.00	1.10	1.20
A3	0.55	0.65	0.75
b	0.30	-	0.50
b1	0.30	0.40	0.45
c	0.08	-	0.22
c1	0.08	0.13	0.20
D	2.72	2.92	3.12
E	2.60	2.80	3.00
E1	1.40	1.60	1.80
e	0.95BSC		
L	0.30	-	0.60
$\theta$	0	-	8°

## 13 修订记录

版本	修订日期	修订内容
V1.0	2019-10-30	初版发布；
V1.1	2020-03-26	新增 P04 内部上拉电阻的相关注释；新增通过 CPUM 切换系统工作模式的说明及例程；