

---

晟矽微电 8 位单片机

MC32F7362B

用户手册

V1.1





## 目录

1	产品概要.....	6
1.1	产品特性.....	6
1.2	订购信息.....	8
1.3	引脚排列.....	8
1.4	端口说明.....	9
2	电气特性.....	10
2.1	极限参数.....	10
2.2	直流电气特性.....	10
2.3	交流电气特性.....	11
2.4	ADC 特性参数.....	12
2.5	OPA 特性参数.....	12
2.6	EEPROM 特性参数.....	13
3	CPU 与存储器.....	14
3.1	指令集.....	14
3.2	程序存储器.....	16
3.3	数据存储器.....	17
3.4	堆栈.....	18
3.5	控制寄存器.....	18
3.5.1	数据指针寄存器 0.....	18
3.5.2	数据指针寄存器 1.....	19
3.5.3	间接寻址寄存器 0.....	19
3.5.4	间接寻址寄存器 1.....	19
3.5.5	间接寻址寄存器 2.....	19
3.5.6	间接寻址寄存器 3.....	20
3.5.7	字操作高字节缓存器.....	20
3.5.8	程序指针计数器低字节.....	20
3.5.9	CPU 状态寄存器.....	20
3.5.10	杂项控制寄存器.....	21
3.6	用户配置字.....	22
4	系统时钟.....	24
4.1	内部高频 RC 振荡器.....	24
4.1.1	HIRC 微调校准寄存器.....	24
4.2	内部低频 RC 振荡器.....	25
4.3	系统工作模式.....	25
4.3.1	振荡器模式寄存器.....	26
4.4	低功耗模式.....	26
5	复位.....	28
5.1	复位条件.....	28



5.2	上电复位.....	28
5.3	外部复位.....	28
5.4	低电压复位.....	29
5.5	看门狗复位.....	29
6	I/O 端口.....	30
6.1	通用 I/O 功能.....	30
6.1.1	端口数据寄存器.....	30
6.1.2	端口方向寄存器.....	30
6.2	内部上/下拉电阻.....	31
6.2.1	上拉电阻控制寄存器.....	31
6.2.2	下拉电阻控制寄存器.....	32
6.3	端口模式控制.....	32
6.3.1	端口数模控制寄存器.....	33
7	定时器 TIMER.....	34
7.1	看门狗定时器 WDT.....	34
7.2	定时器 T0.....	34
7.2.1	定时器 T0 控制寄存器.....	34
7.2.2	定时器 T0 计数器.....	35
7.2.3	定时器 T0 重载寄存器.....	35
7.3	定时器 T1.....	36
7.3.1	定时器 T1 控制寄存器.....	37
7.3.2	定时器 T1 计数器.....	39
7.3.3	定时器 T1 重载寄存器.....	39
7.4	定时器 T2.....	39
7.4.1	定时器 T2 控制寄存器.....	41
7.4.2	定时器 T2 计数器.....	42
7.4.3	定时器 T2 重载寄存器.....	42
7.4.4	定时器 T2 比较寄存器.....	43
7.4.5	PWM2 控制寄存器.....	43
8	模数转换器 ADC.....	46
8.1	ADC 概述.....	46
8.2	ADC 相关寄存器.....	47
8.2.1	ADC 控制寄存器.....	47
8.2.2	ADC 转换结果寄存器.....	48
8.2.3	ADC 零点偏移修调控制寄存器.....	49
8.3	ADC 操作步骤.....	49
8.4	ADC 零点偏移修调流程.....	50
9	运算放大器 OPA.....	51
9.1	OPA 概述.....	51
9.2	OPA 相关寄存器.....	52



9.2.1	OPA 控制寄存器.....	52
9.2.2	OP0/OP1 分压控制寄存器.....	54
9.2.3	OP2 自消除寄存器.....	55
9.3	OPA 失调电压调校流程.....	56
10	低电压检测 LVD.....	57
10.1.1	LVD 控制寄存器 .....	57
11	异步通讯 UART .....	58
11.1	UART 概述.....	58
11.2	UART 工作模式与传输格式.....	59
11.3	UART 波特率 .....	59
11.4	UART 多机通讯 .....	60
11.4.1	从机地址自动识别.....	60
11.4.2	预设地址和广播地址.....	60
11.4.3	端口输出自动转换.....	61
11.5	UART 相关寄存器.....	62
11.5.1	UART 控制寄存器 .....	62
11.5.2	UART 状态寄存器 .....	63
11.5.3	UART 波特率寄存器.....	64
11.5.4	UART 数据寄存器 .....	64
11.5.5	UART 地址寄存器 .....	64
11.5.6	UART 地址掩码寄存器.....	64
12	EEPROM 存储器.....	66
12.1	EEPROM 概述 .....	66
12.2	EEPROM 相关寄存器 .....	66
12.2.1	EEPROM 控制寄存器 .....	66
12.2.2	EEPROM 保护寄存器 .....	66
12.2.3	EEPROM 地址寄存器 .....	67
12.2.4	EEPROM 数据寄存器 .....	67
12.3	EEPROM 操作示例.....	67
13	FLASH 烧录编程.....	69
13.1	FLASH 在板编程 .....	69
13.1.1	DEBUG 控制寄存器 .....	69
13.1.2	DEBUG 保护寄存器 .....	70
14	中断 .....	71
14.1	外部中断.....	71
14.2	定时器中断.....	71
14.3	T1 捕捉中断 .....	71
14.4	键盘中断.....	72
14.5	ADC 中断.....	72
14.6	OP2 比较中断.....	72



14.7	LVD 中断.....	73
14.8	UART 中断.....	73
14.9	中断相关寄存器 .....	73
14.9.1	中断使能寄存器 .....	73
14.9.2	中断标志寄存器 .....	74
15	特性曲线.....	77
15.1	I/O 特性.....	77
15.1.1	输入 SMT 阈值电压 VS 电源电压.....	77
15.1.2	上/下拉电阻值 VS 电源电压 .....	77
15.1.3	I/O 输出电流 VS 端口电压 (VDD=5V) .....	78
15.2	功耗特性.....	79
15.2.1	运行模式 功耗 VS 电源电压.....	79
15.2.2	HOLD 模式 功耗 VS 电源电压 .....	81
15.2.3	休眠模式 功耗 VS 电源电压.....	82
15.3	模拟电路特性 .....	83
15.3.1	HIRC 频率 VS 电源电压/温度 .....	83
15.3.2	LIRC 频率 VS 电源电压/温度 .....	84
15.3.3	VIR 电压 VS 电源电压/温度 .....	84
15.3.4	LVR 阈值电压 VS 温度.....	86
15.3.5	LVD 阈值电压 VS 温度.....	87
16	封装尺寸.....	89
16.1	SOP20 .....	89
16.2	SSOP20 (e=0.8) .....	89
16.3	TSSOP20.....	90
16.4	SOP14 .....	91
17	修订记录.....	92



# 1 产品概要

## 1.1 产品特性

- ▶ 8 位 CPU 内核
  - 精简指令集, 8 级深度硬件堆栈
  - CPU 为双时钟, 可在系统高/低频时钟之间切换
  - 系统高频时钟下 FCPU 可配置为 FHOSC 的 4/8/16/32/64 分频
  - 系统低频时钟下 FCPU 固定为 FLOSC 的 2 分频
- ▶ 程序存储器
  - 4K×16 位 FLASH 型程序存储器, 可通过间接寻址读取程序存储器内容
  - 支持在板带电烧录编程, 擦写次数至少 1000 次
- ▶ 数据存储器
  - 256 字节 SRAM 型通用数据存储器, 支持直接寻址、间接寻址等多种寻址方式
  - 128×16 位 EEPROM 型数据存储器, 支持单独烧录和软件读写, 擦写次数至少 10000 次
- ▶ 3 组共 15 个 I/O
  - P0 (P00~P02), P1 (P10~P15), P2 (P20~P25)
  - 所有端口均支持施密特输入, 均支持推挽输出
  - P20 可复用为外部复位 RST 输入
  - 所有端口均内置上拉和下拉电阻, 均可单独使能
  - P14/P00 可复用为外部中断输入, 支持外部中断唤醒功能
  - P1 和 P2 所有端口均支持键盘中断唤醒功能, 并可单独使能
- ▶ 系统时钟源
  - 内置高频 RC 振荡器 (32MHz), 可用作系统高频时钟源, 支持软件微调
  - 内置低频 RC 振荡器 (32KHz), 可用作系统低频时钟源
- ▶ 系统工作模式
  - 高速模式: CPU 在高频时钟下运行, 低频时钟源工作
  - 低速模式: CPU 在低频时钟下运行, 高频时钟源可选停止或工作
  - HOLD1 模式 (低功耗模式): CPU 暂停, 高频时钟源工作, 低频时钟源可选停止或工作
  - HOLD2 模式 (低功耗模式): CPU 暂停, 高频时钟源停止, 低频时钟源工作
  - 休眠模式 (低功耗模式): CPU 暂停, 高/低频时钟源均停止
- ▶ 内部看门狗计数器 (WDT)
  - 溢出时间可配置: 16ms/64ms/256ms/1024ms
  - 工作模式可配置: 始终开启、始终关闭、低功耗模式下关闭
- ▶ 3 个定时器
  - 8 位定时器 T0, 可实现外部计数功能
  - 16 位定时器 T1, 可实现外部计数和内/外部信号输入捕捉功能
  - 8 位定时器 T2, 可实现 3 路共周期独立占空比的 PWM (其中 1 路可扩展为 1 对 8+3 模式的带死区互补 PWM)
- ▶ 1 个 12 位高精度 SAR 型 ADC



- 13 路外部通道: AN0~AN12; 3 路内部通道: GND、VDD/4、OP2\_OUT
- 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: FHIRC 的 16/32/64/128 分频
- 支持零点校准
- ▶ 3 个运算放大器 OP0~OP2
  - 输入共模 0 ~ (VDD-1.4V)
  - OP0/OP1 为开漏输出型, 共用一个输出端口, 且负端输入及输出均为纯模拟端口
  - OP0 正端输入可选 14 位的 VDD 内部分压电压、或外部输入电压
  - OP1 正端输入可选 12 位的 VDD 内部分压电压、或外部输入电压
  - OP2 支持失调电压自消除调校 (调校精度±2mV), 内置增益电路, 放大倍数为 10/20 倍
  - OP2 的输出信号 OP2\_OUT 可作为 ADC 输入
  - OP2 可实现模拟比较器功能, 比较模式下输出为数字信号且支持去抖处理, 输出端电平可选择上升沿或下降沿触发中断 (支持中断唤醒), 可从端口输出且支持输出取反
- ▶ 1 组异步通讯 UART 接口
  - 支持双线异步全双工、单线异步半双工等多种工作模式
  - 支持 8 位/9 位数据位 (含校验位)、1 位/2 位停止位等多种传输格式
  - 内置波特率发生器, 时钟源为系统高频时钟 FHOSC 的 2 分频时钟 (16MHz)
  - 内置接收缓冲器和锁存寄存器、发送缓冲器和移位寄存器, 支持收/发数据的 2 级缓存
  - 发送缓冲器空、接收完成、发送完成时触发中断
  - 双线模式 2 组端口 RX0/TX0 或 RX1/TX1 可选, 单线模式 4 路端口可选
  - 支持端口电平反向输入/输出
- ▶ 中断
  - 外部中断 (INT0~INT1), 键盘中断 (P10~P15, P20~P25)
  - 定时器中断 (T0~T2), T1 捕捉中断
  - ADC 中断, OP2 比较中断, LVD 中断
  - UART 中断 (接收完成中断、发送完成中断、发送缓冲器空中断)
- ▶ 低电压检测 LVD
  - 2.0V/2.2V/2.4V/2.6V/2.8V/3.2V/3.6V/4.0V
  - 支持 LVD 中断唤醒
- ▶ 低电压复位 LVR
  - 1.8V/2.0V/2.7V
- ▶ 工作电压
  - VLVR27 ~ 5.5V @ F<sub>cpu</sub> = 0~8MHz
  - VLVR20 ~ 5.5V @ F<sub>cpu</sub> = 0~4MHz
  - VLVR18 ~ 5.5V @ F<sub>cpu</sub> = 0~1MHz
- ▶ 封装形式
  - SOP20/SSOP20/TSSOP20/SOP14



## 1.2 订购信息

表 1-1 订购信息

产品名称	封装形式	备注
MC32F7362B0M	SOP20	
MC32F7362B0YR	SSOP20	e=0.8
MC32F7362B0Y	TSSOP20	
MC32F7362B0K	SOP16	
MC32F7362B0J	SOP14	

## 1.3 引脚排列

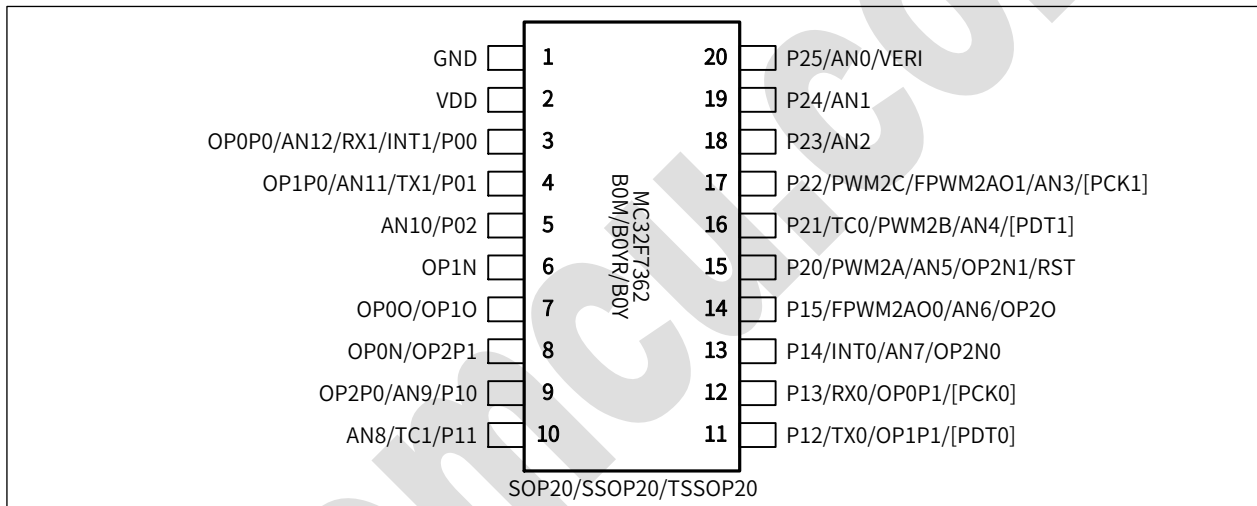


图 1-1 MC32F7362B0M/B0YR/B0Y

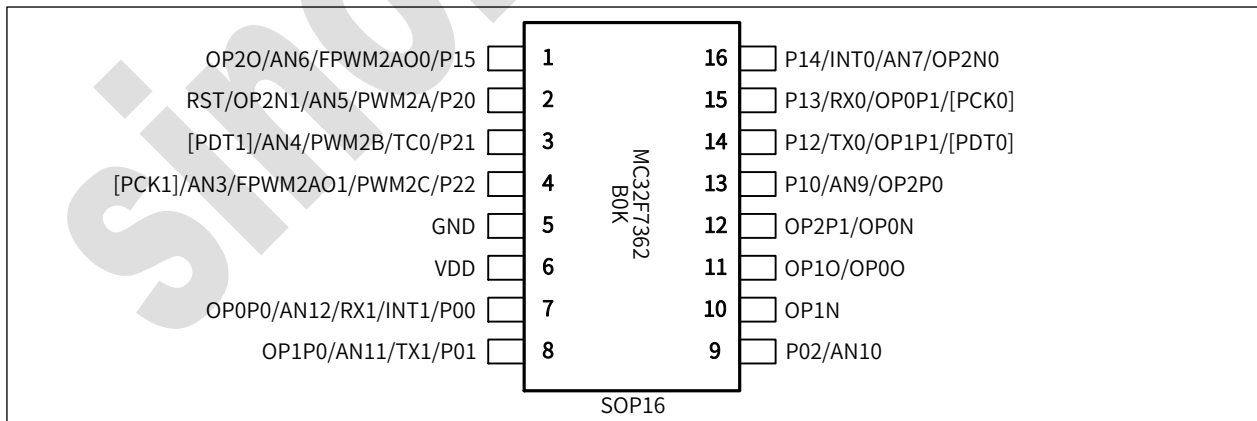


图 1-2 MC32F7362B0K



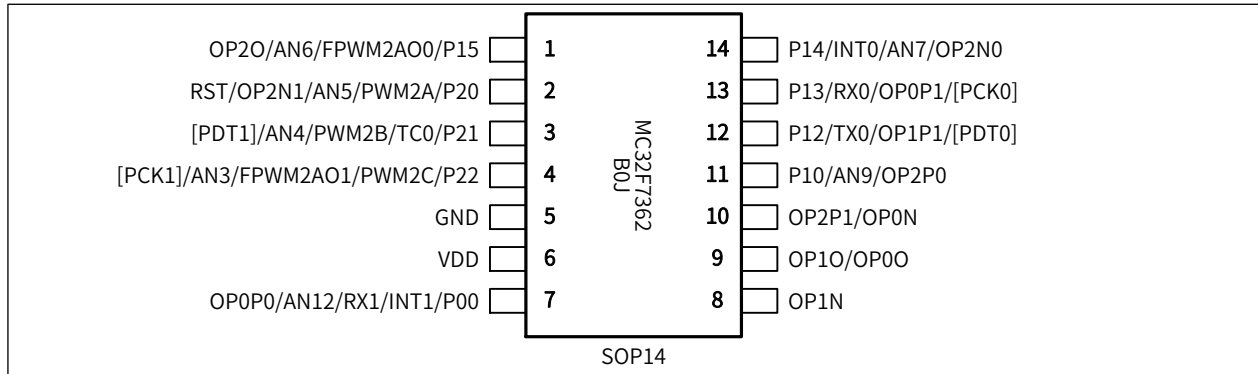


图 1-3 MC32F7362B0J

## 1.4 端口说明

表 1-2 端口说明

端口名称	类型	功能说明
VDD	P	电源
GND	P	地
P0, P1, P2	D	GPIO (推挽输出), 内部上/下拉
INT0~INT1	DI	外部中断输入
TC0~TC1	DI	定时器 T0~T1 的外部计数输入
PWM2A	DO	定时器 T2 中 PWM2A 的 PWM 输出
FPWM2AO0~FPWM2AO1	DO	定时器 T2 中 PWM2A 的互补 PWM 输出通道
PWM2B	DO	定时器 T2 中 PWM2B 的 PWM 输出
PWM2C	DO	定时器 T2 中 PWM2C 的 PWM 输出
AN0~AN12	AI	ADC 外部输入通道
VERI	AI	ADC 外部参考电压输入
OP0P0~OP0P1	AI	OP0 正端外部输入通道
OP1P0~OP1P1	AI	OP1 正端外部输入通道
OP0N, OP1N	AI	OP0/OP1 负端外部输入, 纯模拟端口
OP0O, OP1O	AO	OP0/OP1 输出, 纯模拟端口, 灌电流开漏输出
OP2P0~OP2P1	AI	OP2 正端外部输入通道
OP2N0~OP2N1	AI	OP2 负端外部输入通道
OP2O	AO/DO	OP2 输出
RX0/TX0, RX1/TX1	D	UART 通讯接收/发送端口
RST	DI	外部复位输入
PCK0/PDT0, PCK1/PDT1	D	编程时钟/数据接口; 在线仿真接口

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



## 2 电气特性

### 2.1 极限参数

表 2-1 极限参数

参数	符号	值	单位
电源电压	VDD	-0.3~6.0	V
I/O 输入电压	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	80	mA
流出 GND 最大电流	IGNDmax	80	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，则将影响其可靠性。

### 2.2 直流电气特性

VDD=5V, T=25°C

表 2-2 直流电气特性

特性	符号	端口	条件	最小	典型	最大	单位
工作电压	VDD	VDD	Fcpu=8MHz@FHIRC/4	V <sub>LVR27</sub>		5.5	V
			Fcpu=4MHz@FHIRC/8	V <sub>LVR20</sub>		5.5	
			Fcpu=2MHz@FHIRC/16	V <sub>LVR20</sub>		5.5	
			Fcpu=1MHz@FHIRC/32	V <sub>LVR18</sub>		5.5	
			Fcpu=500KHz@FHIRC/64	V <sub>LVR18</sub>		5.5	
			Fcpu=16KHz@FLIRC/2	V <sub>LVR18</sub>		5.5	
输入漏电流	I <sub>leak</sub>	所有输入脚	VDD=5V	-1		1	μA
输入高电平	V <sub>ih</sub>	所有输入脚	SMTVS 配置	0.8VDD			V
			SMTVS 配置	2.0			
输入低电平	V <sub>il</sub>	所有输入脚	SMTVS 配置			0.2VDD	V
			SMTVS 配置			0.8	
上拉电阻	R <sub>pu</sub>	P0, P1, P2	VDD=5V, Vin=0	-25%	16	+25%	KΩ
下拉电阻	R <sub>pd</sub>	P0, P1, P2	Vin=VDD=5V	-25%	16	+25%	KΩ
输出源电流	I <sub>oh</sub>	推挽输出脚	Voh=VDD-0.6V, DRVS 为“增强输出”		25		mA
			Voh=VDD-0.6V, DRVS 为“正常输出”		15		mA
输出灌电流	I <sub>ol</sub>	所有输出脚	Vol=0.6V, DRVS 为“增强输出”		45		mA
			Vol=0.6V, DRVS 为“正常输出”		25		mA
输出驱动管		推挽输出脚	高速输出@空载			16	MHz



开关速度			低速输出@空载			2	MHz
运行模式功耗	Irun	VDD	VDD =5V	Fcpu=8MHz@HIRC		3.2	mA
				Fcpu=4MHz@HIRC		2.0	mA
				Fcpu=2MHz@HIRC		1.4	mA
				Fcpu=1MHz@HIRC		1.1	mA
				Fcpu=500KHz@HIRC		950	μA
				Fcpu=16KHz@LIRC		9	μA
			VDD =3V	Fcpu=8MHz@HIRC		2.0	mA
				Fcpu=4MHz@HIRC		1.3	mA
				Fcpu=2MHz@HIRC		0.9	mA
				Fcpu=1MHz@HIRC		750	μA
				Fcpu=500KHz@HIRC		650	μA
				Fcpu=16KHz@LIRC		5.5	μA
HOLD1 功耗	Ihold1	VDD	VDD=5V, CPU 停, HIRC/LIRC 开		800		μA
			VDD=3V, CPU 停, HIRC/LIRC 开		550		μA
HOLD2 功耗	Ihold2	VDD	VDD=5V, CPU 停, HIRC 关, LIRC 开		2.5		μA
			VDD=3V, CPU 停, HIRC 关, LIRC 开		1.1		μA
休眠模式功耗	Istop	VDD	VDD=5V, 休眠模式, WDT/LVR 关		0.5	3	μA
			VDD=3V, 休眠模式, WDT/LVR 关		0.3		μA
			VDD=5V, 休眠模式, WDT 开, LVR 关		3	6	μA
			VDD=3V, 休眠模式, WDT 开, LVR 关		1.1		μA
			VDD=5V, 休眠模式, WDT 关, LVR 开		8	20	μA
			VDD=3V, 休眠模式, WDT 关, LVR 开		6.5		μA
低压检测电压	VLVD	VDD	LVDVS 选择	-5%		+5%	V
LVD 响应时间	TLVD			1	50	2000	μs
低压复位电压	VLVR	VDD	LVRVS 配置	-10%		+10%	V
LVD/LVR 回滞电压		VDD			6%	12%	

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

## 2.3 交流电气特性

表 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	F <sub>HIRC</sub>	VDD=2.0V~5.5V, T=25°C	-1.5%	32	+1.5%	MHz
		VDD=2.0V~5.5V, T=-40°C~85°C	-3%		+3%	
LIRC 振荡频率	F <sub>LIRC</sub>	VDD=5V, T=25°C	-50%	32	+50%	KHz



## 2.4 ADC 特性参数

VDD=5V, T=25°C

表 2-4 ADC 特性参数

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	2.5		5.5	V
积分非线性误差	INL	V <sub>REF</sub> =VDD, F <sub>ADC</sub> =1MHz, T <sub>con</sub> =27μs			±4	LSB
微分非线性误差	DNL	V <sub>REF</sub> =VDD, F <sub>ADC</sub> =1MHz, T <sub>con</sub> =27μs			±2	LSB
零点偏移误差	EZ	V <sub>REF</sub> =VDD, F <sub>ADC</sub> =1MHz, T <sub>con</sub> =27μs			±4	LSB
增益误差	ET	V <sub>REF</sub> =VDD, F <sub>ADC</sub> =1MHz, T <sub>con</sub> =27μs			±4	LSB
转换时钟	F <sub>ADC</sub>	VDD=5V			1	MHz
转换时间	T <sub>con</sub>		14		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>				2	μA
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	μA
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		24		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	-0.5%	2	+0.5%	
		选择内部参考电压 V <sub>IR</sub> , T=-40°C~85°C	-3%		+3%	
		选择内部参考电压 V <sub>IR</sub> , T=25°C	-1.5%	3/4	+1.5%	
		选择内部参考电压 V <sub>IR</sub> , T=-40°C~85°C	-4%		+4%	
		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		5.5	V

## 2.5 OPA 特性参数

VDD=5V, T=25°C

表 2-5 OPA 特性参数

特性	符号	条件	最小	典型	最大	单位
----	----	----	----	----	----	----



OPA 有效工作电压	V <sub>OPA</sub>	T=-40°C~85°C	2.4		5.5	V
OPA 启动时间					200	μs
输入失调电压	V <sub>offset</sub>	OP0/OP1 无调校	-7	5	+7	mV
		OP2 未调校	-15		+15	
		OP2 已调校	-2		+2	
输入共模电压	V <sub>com</sub>		0		VDD-1.4	V
OP0/OP1 输出灌电流	I <sub>sink</sub>	VDD=5V, Vol=0.5V	1.6	2.8		mA
		VDD=3V, Vol=0.3V	0.7	1.2		mA
输出电压摆幅	V <sub>ovs</sub>		0	0.2	0.3	V
			4.7	4.8	5.0	
电压摆率	SR		0.1	0.2		V/μs
开环增益	A <sub>vol</sub>		60	80		dB
电源抑制比	PSRR		60	80		dB
共模抑制比	CMRR		60	80		dB
OP0 分压电阻工作电流		VDD=5V		600	800	μA
OP1 分压电阻工作电流		VDD=5V		500	600	μA

## 2.6 EEPROM 特性参数

VDD=5V

表 2-6 EEPROM 特性参数

特性	符号	条件	最小	典型	最大	单位
EEPROM 读操作电压	V <sub>EERD</sub>	T=-40°C~85°C	2.0		5.5	V
EEPROM 写操作电压	V <sub>EEWR</sub>	T=-40°C~85°C	2.0		5.5	V
EEPROM 写操作电流	I <sub>EEWR</sub>	T=-40°C~85°C		2		mA
EEPROM 静态电流	I <sub>EESTP</sub>	T=25°C		1		μA
EEPROM 单地址写入时间	T <sub>EEWR</sub>	VDD=2.0V~5.5V, T=-40°C~85°C		5		ms
EEPROM 擦写次数		VDD=5V, T=25°C	10000			cycle
EEPROM 数据保持时间		T=25°C	10			year



## 3 CPU 与存储器

### 3.1 指令集

芯片的指令集为精简指令集。

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

表 3-1 指令汇总表

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



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



NOP	空操作	空操作	1	1	-
DAA	BCD 码加法操作后, 将 A 的值调整为 BCD 码	A(HEX 码)→A(BCD 码)	1	1	C
DSA	BCD 码减法操作后, 将 A 的值调整为 BCD 码	A(HEX 码)→A(BCD 码)	1	1	-
CLRWDT	将看门狗计数器清零	0→WDTCNT	1	1	TO,PD
STOP	进入低功耗模式	0→WDTCNT; CPU 暂停	1	1	TO,PD

注:

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

## 3.2 程序存储器

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

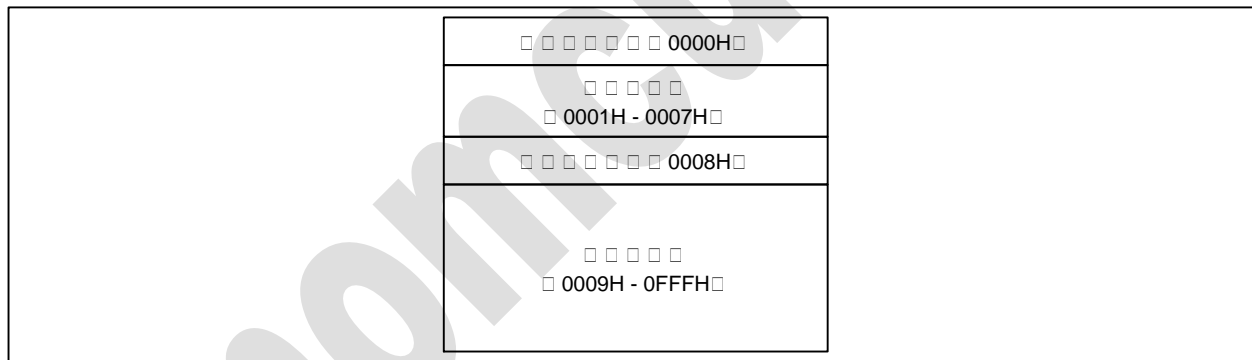


图 3-1 地址映射

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

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

```

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

```





### 3.3 数据存储器

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

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

表 3-2 数据存储器区地址映射

地址	类型	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
000H-0FFH	GPR	通用数据存储器区							
100H-17FH	保留	保留							
180H-187H	SFR	INDF0	INDF1	INDF2	HIBYTE	FSR0	FSR1	PCL	PFLAG
188H-18FH		MCR	INDF3	INTE0	INTF0	OSCMR		INTE1	INTF1
190H-197H		IOP0	OEP0	PUP0	PDP0	IOP1	OEP1	PUP1	PDP1
198H-19FH		IOP2	OEP2	PUP2	PDP2	P1KBCR	P2KBCR		LVDCR
1A0H-1A7H		T0CR	T0CNT	T0LOAD			HIRCCAL	DBGCR	DBGPR
1A8H-1AFH		T1CR0	T1CR1	T1CNTH	T1CNTL	T1LOADH	T1LOADL		
1B0H-1B7H		T2CR	T2CNT	T2LOAD	T2DATA	T2DATB	T2DATC	PWM2CR0	PWM2CR1
1B8H-1BFH		PWM2CR2							
1C0H-1C7H					ECCR	EEPR	EEAR	EEDRH	EEDRL
1C8H-1CFH		ADCR0	ADCR1	ADRH	ADRL	OSADJCR	P0ADCR	P1ADCR	P2ADCR
1D0H-1D7H		OPCR0	OPCR1	OPCR2	OP0VRH	OP0VRL	OP1VRH	OP1VRL	OP2CLR
1D8H-1DFH		URTCR0	URTCR1	URTCR2	URTSR	URTBR	URTDR	URTAR	URTMR
1E0H-1FFH		保留							

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

表 3-3 数据存储器寻址方式地址组成

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

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

```
MOVAI    55H
MOVRA    10H
```

；将 55H 写入数据存储器 10H 地址中



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

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

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

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

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

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

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

### 3.4 堆栈

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

## 3.5 控制寄存器

### 3.5.1 数据指针寄存器 0

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

BIT[7:0] FSR0[7:0] – 数据指针寄存器 0



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

### 3.5.2 数据指针寄存器 1

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

BIT[7:0] FSR1[7:0] – 数据指针寄存器 1

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

### 3.5.3 间接寻址寄存器 0

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

BIT[7:0] INDF0[7:0] – 间接寻址寄存器 0

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

### 3.5.4 间接寻址寄存器 1

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

BIT[7:0] INDF1[7:0] – 间接寻址寄存器 1

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

### 3.5.5 间接寻址寄存器 2

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

BIT[7:0] INDF2[7:0] – 间接寻址寄存器 2

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



### 3.5.6 间接寻址寄存器 3

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

BIT[7:0] INDF3[7:0] – 间接寻址寄存器 3

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

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

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

### 3.5.8 程序指针计数器低字节

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

BIT[7:0] PC[7:0] – 程序指针计数器低 8 位

程序指针计数器 (PC) 有以下几种操作模式:

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

对 PCL 操作指令:

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

### 3.5.9 CPU 状态寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PFLAG	-	-	-	-	-	Z	DC	C



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

BIT[2] Z - 零标志位

0: 算术或逻辑运算的结果不为零;

1: 算术或逻辑运算的结果为零;

BIT[1] DC - 半字节进位/借位标志位

0: 加法运算中半字节无进位; 减法运算中半字节有借位;

1: 加法运算中半字节有进位; 减法运算中半字节无借位;

BIT[0] C - 进位/借位标志位

0: 加法运算中无进位; 减法运算中有借位; 移位操作中移出位为 0;

1: 加法运算中有进位; 减法运算中无借位; 移位操作中移出位为 1;

### 3.5.10 杂项控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
MCR	GIE	-	TO	PD	INT1M1	INT1M0	INT0M1	INT0M0
R/W	R/W	-	R	R	R/W	R/W	R/W	R/W
初始值	0	-	0	0	0	0	0	0

BIT[7] GIE - 中断总使能位

0: 屏蔽所有中断;

1: 由相应的中断使能位决定 CPU 是否响应中断源所触发的中断;

BIT[5] TO - 看门狗溢出标志位

0: 上电复位, 或执行 CLRWDT/STOP 指令;

1: 发生 WDT 溢出;

BIT[4] PD - 进入低功耗模式标志位

0: 上电复位, 或执行 CLRWDT 指令;

1: 执行 STOP 指令;

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

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

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

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



### 3.6 用户配置字

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

芯片的用户配置字，定义如下：

表 3-4 用户配置字

符号	功能说明
FCPUS	高频时钟下 F <sub>CPU</sub> 分频选择： 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;                      F <sub>CPU</sub> =F <sub>HOSC</sub> /32;                      F <sub>CPU</sub> =F <sub>HOSC</sub> /64;
RSTEN	RST 外部复位端口设置： P20 为外部复位脚;                      P20 为输入/输出脚;
LVRMD	LVR 模式设置： LVR 始终开启;                      LVR 在运行模式下开启，在低功耗模式下关闭;
LVRVS	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) 1.8V; 2.0V; 2.7V
WDTM	WDT 模式设置： WDT 始终关闭; WDT 在运行模式下开启，在低功耗模式下关闭; WDT 始终开启;
WDTT	WDT 溢出时间（典型值）选择： 16ms; 64ms; 256ms; 1024ms;
SMTVS	端口施密特阈值选择： 2.0V/0.8V;                      0.8V <sub>D</sub> D/0.2V <sub>D</sub> D;
DRVS	端口输出驱动电流选择： 增强驱动输出;                      正常驱动输出;
SPDS	端口输出驱动管开关速度选择： 高速输出;                      低速输出;
OPAENSET	OP0/OP1 使能设置： OP0/OP1 始终开启，寄存器位 OP0EN/OP1EN 无效; OP0/OP1 是否开启由寄存器位 OP0EN/OP1EN 控制;
ENCR	程序代码加密设置： 程序代码加密;                      程序代码不加密;
DBGPIN0	PCK0/PDT0 编程扩展设置： 端口在复位时固定为编程端口，复位完成后固定为通用端口; 可通过寄存器位将端口切换为编程端口或通用端口;
DBGPIN1	PCK1/PDT1 编程扩展设置：



	<p>端口在复位时固定为编程端口，复位完成后固定为通用端口； 可通过寄存器位将端口切换为编程端口或通用端口；</p>
--	--

Shomcu.com



## 4 系统时钟

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

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

CPU 的时钟源可在系统高频时钟 FHOSC 和系统低频时钟 FLOSC 之间切换。FHOSC 下 CPU 的时钟频率 F<sub>CPU</sub> 通过配置字 FCPUS 选择；FLOSC 下 F<sub>CPU</sub> 则固定为 FLOSC 的 2 分频。

WDT (看门狗) 电路的时钟源固定为内部低频 RC 振荡器 LIRC。

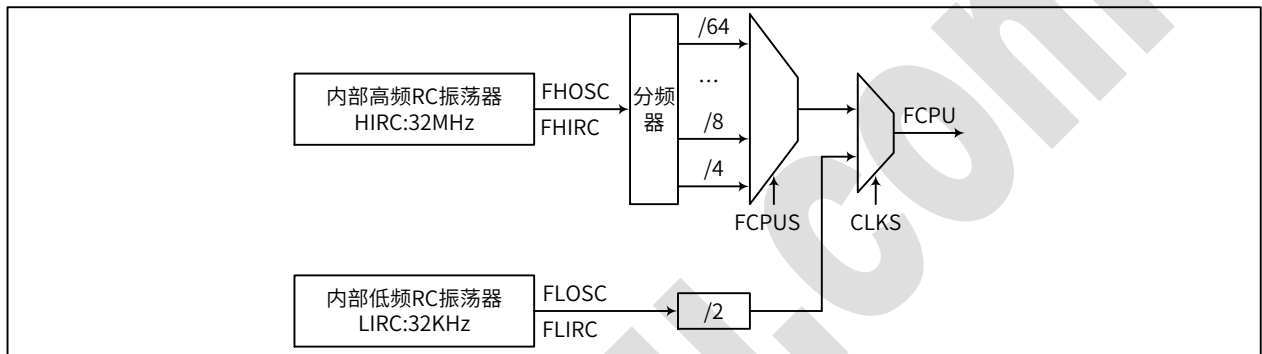


图 4-1 系统时钟框图

### 4.1 内部高频 RC 振荡器

芯片内置 1 个振荡频率为 32MHz 的高精度 HIRC 振荡器，可用作系统高频时钟源。

HIRC 频率可微调校准，校准值保存在 8 位 HIRC 微调校准寄存器 HIRCCAL 中，芯片复位后，寄存器自动加载出厂设置值作为初始值，将 HIRC 频率调整至 32MHz。该寄存器可通过软件进行微调，以获得一定范围内的其他频率。

#### 4.1.1 HIRC 微调校准寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
HIRCCAL	HIRCCAL7	HIRCCAL6	HIRCCAL5	HIRCCAL4	HIRCCAL3	HIRCCAL2	HIRCCAL1	HIRCCAL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	U	U	U	U	U	U	U	U

BIT[7:0] HIRCCAL[7:0] – HIRC 频率微调校准位 (复位初始值为出厂设置值)

注:

- 1、HIRC 最大调节范围为 (32MHz-10%) – (32MHz+4%) (以实际芯片为准)，因芯片及模块的工作电压受其时钟频率限制，所以微调 HIRC 时推荐从初值 (32MHz) 往低频调节，以免影响正常的工作电压范围；
- 2、校准位 1 个 LSB 所调节的频率变化是非线性的，最大约为 0.5%；
- 3、HIRCCAL 中已有出厂校准值，软件微调前应备份原值以便恢复；





## 4.2 内部低频 RC 振荡器

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

## 4.3 系统工作模式

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

表 4-1 工作模式

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

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

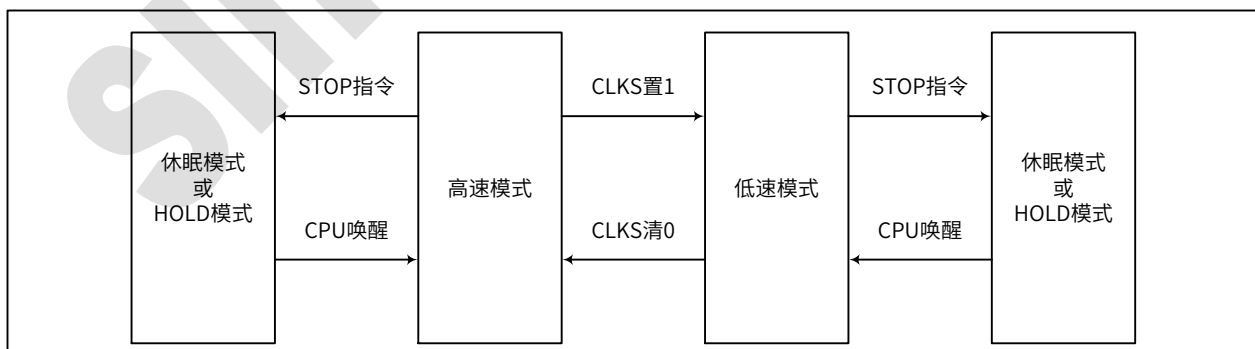


图 4-2 工作模式切换示意图

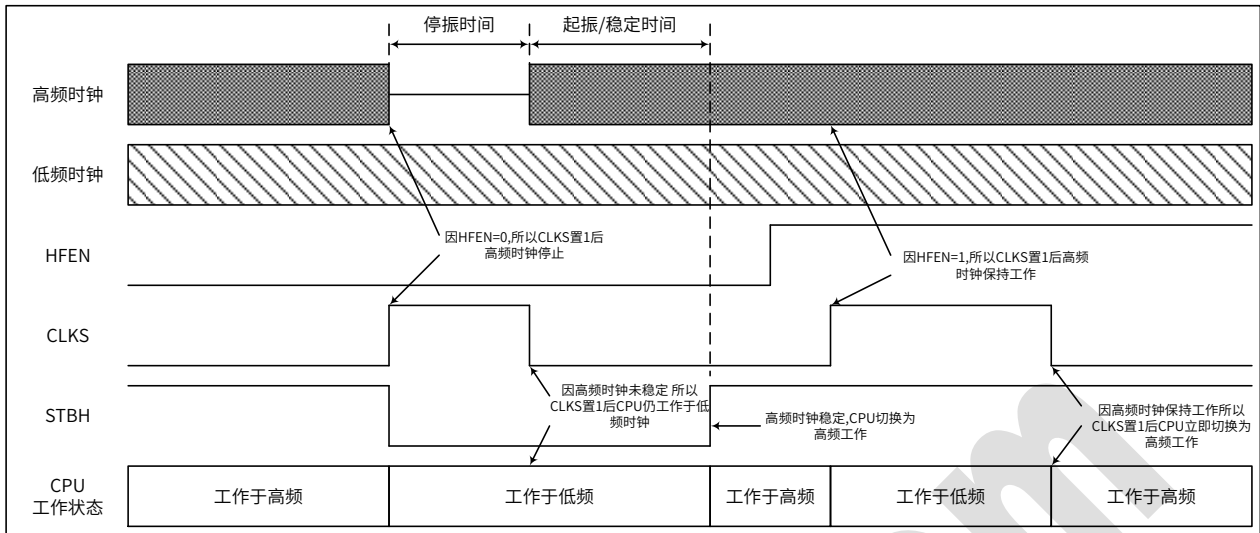


图 4-3 高低频时钟切换时序图

### 4.3.1 振荡器模式寄存器

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

- BIT[5] STBL – 低频时钟源振荡状态标志位
  - 0: 低频时钟源停振或未稳定;
  - 1: 低频时钟源已稳定振荡;
- BIT[4] STBH – 高频时钟源振荡状态标志位
  - 0: 高频时钟源停振或未稳定;
  - 1: 高频时钟源已稳定振荡;
- BIT[2] CLKS – CPU 时钟源选择位
  - 0: 系统高频时钟作为 CPU 时钟源;
  - 1: 系统低频时钟作为 CPU 时钟源;
- BIT[1] LFEN – 低频时钟源使能位
  - 0: 在休眠/HOLD 模式下, 低频时钟源暂停工作;
  - 1: 低频时钟源始终工作;
- BIT[0] HFEN – 高频时钟源使能位
  - 0: 在低速/休眠/HOLD 模式下, 高频时钟源暂停工作;
  - 1: 高频时钟源始终工作;

## 4.4 低功耗模式

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



执行 STOP 指令可使系统进入低功耗模式，同时对系统会产生以下影响：

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

以下情况可使系统退出低功耗模式：

- 芯片复位；
- WDT 溢出（若低功耗模式下 WDT 及其时钟源保持继续工作）；
- 外部中断请求发生（若有外部中断功能并有效）；
- 定时器中断请求发生（若低功耗模式下定时器及其时钟源保持继续工作）；
- 键盘中断请求发生（若有键盘中断功能并有效）；
- LVD 中断请求发生（若低功耗模式下 LVD 保持继续工作）；
- OP2 比较中断请求发生（若 OP2 为比较器模式并在低功耗模式下保持继续工作）；

注：

1、低功耗模式下触发中断请求时，若对应的中断使能位关闭，则不会退出低功耗模式；若对应的中断使能位开启而中断总使能位关闭，则仅唤醒 CPU 执行下一条指令；若对应的中断使能位和中断总使能位均开启，则唤醒 CPU 后将执行中断服务程序；

2、未使用或未封出的引脚，应将其对应的 I/O 端口设置为输出、输入上拉或输入下拉等稳定状态，以免因引脚浮空而产生漏电流或非预期的中断唤醒；



## 5 复位

### 5.1 复位条件

芯片共有如下几种复位方式：

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

任何一种复位发生后，系统进入复位状态，执行初始化操作并重置 SFR 为复位初始值；复位条件解除后，系统退出复位状态，在保持一定时间的复位延时（典型值约为 16ms）以等待电路稳定后，CPU 重新从程序存储器 0000H 地址处开始运行。

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

### 5.2 上电复位

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

上电复位过程主要包括以下几个步骤：

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

### 5.3 外部复位

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



## 5.4 低电压复位

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

## 5.5 看门狗复位

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

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



## 6 I/O 端口

### 6.1 通用 I/O 功能

芯片的输入/输出端口包括一组 3 位端口 P0 和两组 6 位端口 P1、P2。所有端口均支持施密特输入，均支持推挽输出。除用作通用数字 I/O 端口外，部分端口还可复用为外部中断输入、PWM 输出、或 ADC 模拟输入等功能。

#### 6.1.1 端口数据寄存器

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

BIT[2:0] P0nD – P0n 端口数据位 (n=2-0)

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IOP1	-	-	P15D	P14D	P13D	P12D	P11D	P10D
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] P1nD – P1n 端口数据位 (n=5-0)

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IOP2	-	-	P25D	P24D	P23D	P22D	P21D	P20D
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] P2nD – P2n 端口数据位 (n=5-0)

#### 6.1.2 端口方向寄存器

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

BIT[2:0] P0nOE – P0n 端口输出使能位 (n=2-0)

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

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OEP1	-	-	P15OE	P14OE	P13OE	P12OE	P11OE	P10OE
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] P1nOE – P1n 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电平状态;  
1: 端口作为输出口, 读端口操作将读取端口的数据位值;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OEP2	-	-	P25OE	P24OE	P23OE	P22OE	P21OE	P20OE
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] P2nOE – P2n 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电平状态;  
1: 端口作为输出口, 读端口操作将读取端口的数据位值;

## 6.2 内部上/下拉电阻

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

### 6.2.1 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	-	-	-	-	-	P02PU	P01PU	P00PU
R/W	-	-	-	-	-	R/W	R/W	R/W
初始值	-	-	-	-	-	0	0	0

BIT[2:0] P0nPU – P0n 端口上拉电阻控制位 (n=2-0)

- 0: 端口内部上拉电阻无效;  
1: 端口内部上拉电阻有效;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PUP1	-	-	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
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] P1nPU – P1n 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]
--	--------	--------	--------	--------	--------	--------	--------	--------



PUP2	-	-	P25PU	P24PU	P23PU	P22PU	P21PU	P20PU
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] P2nPU – P2n 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;

### 6.2.2 下拉电阻控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PDP0	-	-	-	-	-	P02PD	P01PD	P00PD
R/W	-	-	-	-	-	R/W	R/W	R/W
初始值	-	-	-	-	-	0	0	0

BIT[2:0] P0nPD – P0n 端口下拉电阻控制位 (n=2-0)

- 0: 端口内部下拉电阻无效;
- 1: 端口内部下拉电阻有效;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PDP1	-	-	P15PD	P14PD	P13PD	P12PD	P11PD	P10PD
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] P1nPD – P1n 端口下拉电阻控制位 (n=5-0)

- 0: 端口内部下拉电阻无效;
- 1: 端口内部下拉电阻有效;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PDP2	-	-	P25PD	P24PD	P23PD	P22PD	P21PD	P20PD
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] P2nPD – P2n 端口下拉电阻控制位 (n=5-0)

- 0: 端口内部下拉电阻无效;
- 1: 端口内部下拉电阻有效;

## 6.3 端口模式控制

部分端口除可作为数字端口外，还可复用为模拟端口。端口输入或输出模拟信号时，若数字 I/O 功能同时开启，则会





产生漏电流，可通过端口数模控制寄存器关闭端口的数字 I/O 功能（内部上/下拉电阻及其控制位不受影响）。

### 6.3.1 端口数模控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
P0ADCR	-	-	-	-	-	P02DC	P01DC	P00DC
R/W	-	-	-	-	-	R/W	R/W	R/W
初始值	-	-	-	-	-	0	0	0

BIT[2:0] P0nDC – P0n 端口数字功能控制位 (n=2-0)

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

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
P1ADCR	-	-	P15DC	P14DC	P13DC	P12DC	P11DC	P10DC
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] P1nDC – P1n 端口数字功能控制位 (n=5-0)

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

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
P2ADCR	-	-	P25DC	P24DC	P23DC	P22DC	P21DC	P20DC
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] P2nDC – P2n 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;



## 7 定时器 TIMER

### 7.1 看门狗定时器 WDT

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

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

执行 CLRWDT 指令或 STOP 指令将清零 WDT 计数器。

WDT 溢出时间可配置为 16ms/64ms/256ms/1024ms。

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

### 7.2 定时器 T0

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

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

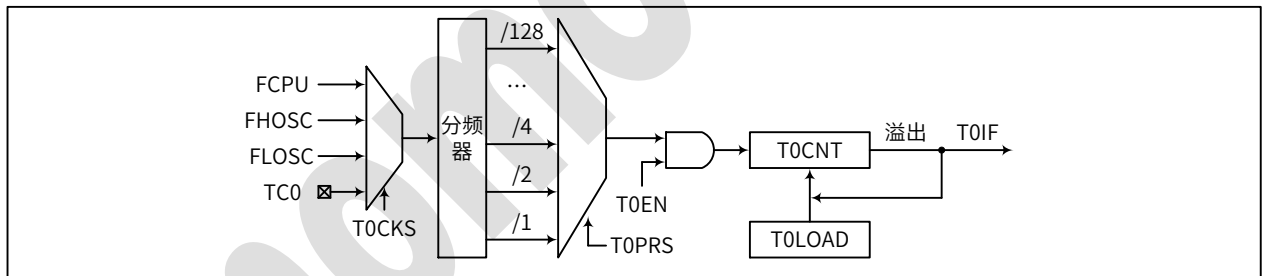


图 7-1 8 位定时器 T0 示意图

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

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

#### 7.2.1 定时器 T0 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
TOCR	T0EN	-	-	T0CKS1	T0CKS0	T0PRS2	T0PRS1	T0PRS0
R/W	R/W	-	-	R/W	R/W	R/W	R/W	R/W
初始值	0	-	-	0	0	0	0	0



BIT[7] T0EN – 定时器 T0 使能位

0: 关闭定时器 T0;

1: 开启定时器 T0;

BIT[4:3] T0CKS[1:0] – T0 时钟源选择位

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

BIT[2:0] T0PRS[2:0] – T0 时钟预分频比选择位

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

### 7.2.2 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	1	1	1	1	1	1	1	1

BIT[7:0] T0CNT[7:0] – T0 计数器，为可读写的递减计数器

### 7.2.3 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]
T0LOAD	T0LOAD7	T0LOAD6	T0LOAD5	T0LOAD4	T0LOAD3	T0LOAD2	T0LOAD1	T0LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

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

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。



### 7.3 定时器 T1

定时器 T1 为 16 位定时/计数器，包含 1 个 16 位递减计数器、可编程预分频器、控制寄存器、16 位重载寄存器。

- 可通过预分频器设置时钟频率，可通过重载寄存器控制计数周期；
- 支持内/外部信号上升/下降沿的输入捕捉功能；
- 支持捕捉中断、溢出中断和溢出唤醒功能；

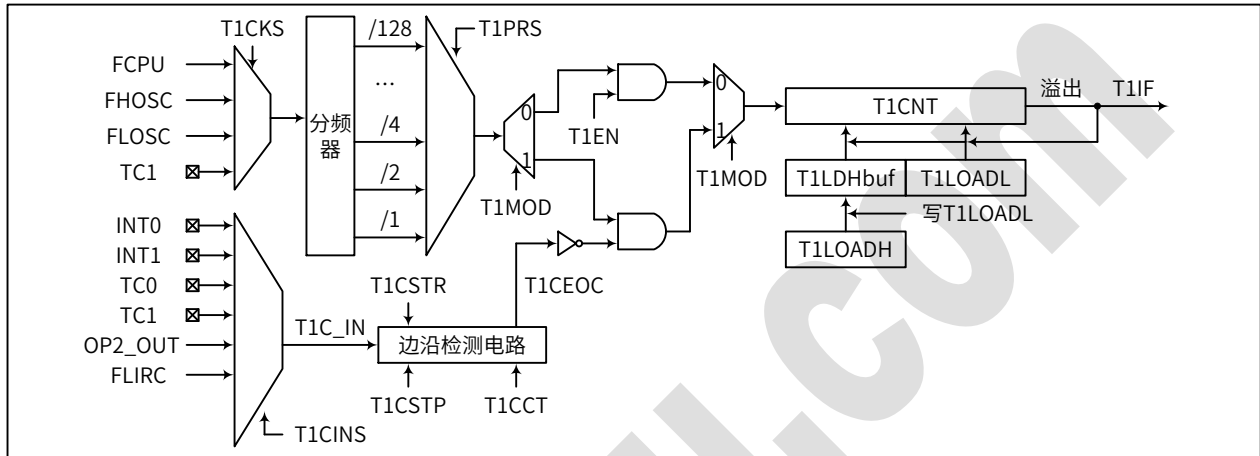


图 7-2 16 位定时器 T1 示意图

定时器 T1，可通过寄存器位 T1CKS 选择时钟源，通过 T1PRS 选择时钟预分频比，所选时钟源通过预分频器后产生 T1 计数器 T1CNT 的计数时钟（上升沿计数）。写 T1CNT 将清零预分频计数器，而预分频比保持不变。

16 位 T1CNT 的高字节 T1CNTH 配有缓冲器，读写 T1CNTH 实际为读写缓冲器中内容，T1CNTH 实际内容仅在读写 T1CNTL 时才自动与缓冲器交互，因此写 T1CNT 需先写 T1CNTH 再写 T1CNTL（硬件同时自动将缓存器内容载入 T1CNTH），而读 T1CNT 则需先读 T1CNTL（硬件同时自动将 T1CNTH 内容载入缓冲器中）再读 T1CNTH。

T1EN=0 时，T1CNT 保持不变，写重载寄存器 T1LOAD 将立即载入 T1CNT；T1EN=1 时，T1CNT 递减计数，计数到 0 的时钟结束后产生溢出信号并触发中断，中断标志 T1IF 将被置 1，同时 T1 自动将当前 T1LOAD 值载入 T1CNT 并重新开始计数。

16 位 T1LOAD 的高字节 T1LOADH 配有缓冲器（T1LDHbuf），写 T1LOADL 时会同时将 T1LOADH 内容载入该缓冲器中，因此调整 T1LOAD 值时需先写 T1LOADH 再写 T1LOADL。此时若 T1EN=0，则会同时再将[缓冲器:T1LOADL]载入 T1CNT；若 T1EN=1，则会在 T1 溢出后才将[缓冲器:T1LOADL]载入 T1CNT。

如图所示，定时器 T1 可实现内/外部信号上升沿或下降沿的输入捕捉功能。通过寄存器位 T1MOD 选择输入捕捉模式，当 T1CEOC 为 1 时写 0 启动 T1 捕捉功能，若 T1CINS 选定的输入信号产生起始沿（T1CSTR 选择），则 T1 开始计数（若计数溢出则自动重载），当输入信号产生终止沿（T1CSTP 选择）时，T1 停止计数且 T1CNT 保持当前值，T1CEOC 自动置 1，同时中断标志 T1CIF 置 1 触发 T1 捕捉中断。通过 T1CNT 前后计数的差值，即可获得输入信号起始沿与终止沿之间的时间宽度。T1 捕捉功能还可通过 T1CCT 选择捕捉沿的个数，以获得 1 个高/低电平脉冲信号的时间宽度（起始沿和终止沿方向相反）或 1/4/8/16 个周期信号的时间宽度（起始沿和终止沿方向相同）。



输入捕捉模式中，在捕捉完成后 T1CEOC 自动置 1，需再次写 0 才能重新启动捕捉功能。在终止沿捕捉完成停止计数到起始沿再次触发恢复计数的期间，T1CNT 将一直保持原值，若要调整则需重新写 T1LOAD 或 T1CNT。

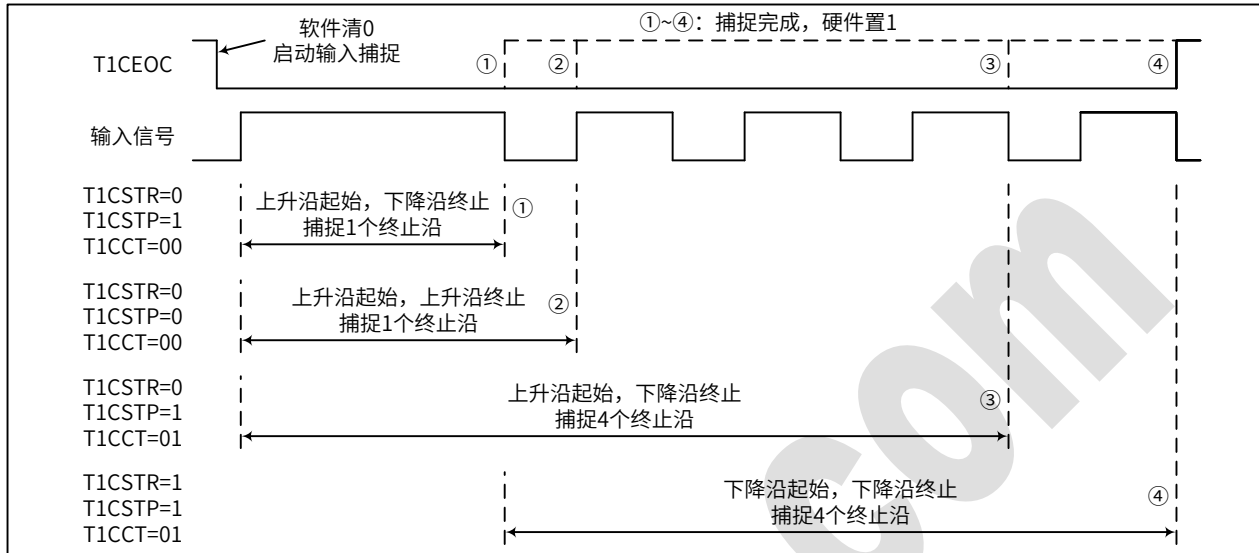


图 7-3 T1 输入捕捉示意图

### 7.3.1 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]
T1CR0	T1EN	T1MOD	-	T1CKS1	T1CKS0	T1PRS2	T1PRS1	T1PRS0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
初始值	0	0	-	0	0	0	0	0

BIT[7] T1EN – 定时器 T1 使能位

- 0: 关闭定时器 T1;
- 1: 开启定时器 T1;

BIT[6] T1MOD – T1 工作模式选择位

- 0: 定时/计数模式;
- 1: 输入捕捉模式;

BIT[4:3] T1CKS[1:0] – T1 时钟源选择位

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

BIT[2:0] T1PRS[2:0] – T1 时钟预分频比选择位

T1PRS[2:0]	T1 时钟预分频比
------------	-----------



000	1 : 1
001	1 : 2
010	1 : 4
011	1 : 8
100	1 : 16
101	1 : 32
110	1 : 64
111	1 : 128

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T1CR1	T1CEOC	T1CINS2	T1CINS1	T1CINS0	T1CSTR	T1CSTP	T1CCT1	T1CCT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	0	0	0	0	0	0	0

BIT[7] T1CEOC – T1 输入捕捉控制位

- 0: 输入捕捉等待起始沿触发或正在计数中，终止沿捕捉完成后自动置 1；
- 1: 输入捕捉未启动或已完成，写 0 启动输入捕捉；

BIT[6:4] T1CINS[2:0] – T1 输入捕捉信号源选择位

T1CINS[2:0]	T1 输入捕捉信号源
000	INT0
001	INT1
010	TC0
011	TC1
100	OP2_OUT
101	FLIRC
110	-
111	-

BIT[3] T1CSTR – T1 输入捕捉起始沿选择位

- 0: 上升沿触发捕捉计数；
- 1: 下降沿触发捕捉计数；

BIT[2] T1CSTP – T1 输入捕捉终止沿选择位

- 0: 上升沿终止捕捉计数；
- 1: 下降沿终止捕捉计数；

BIT[1:0] T1CCT[1:0] – T1 捕捉沿个数选择位

T1CCT[1:0]	T1 捕捉沿个数
00	1 个
01	4 个
10	8 个



11	16 个
----	------

### 7.3.2 定时器 T1 计数器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T1CNTH	T1CNT15	T1CNT14	T1CNT13	T1CNT12	T1CNT11	T1CNT10	T1CNT9	T1CNT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T1CNT[15:8] - T1 计数器高 8 位，为可读写的递减计数器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T1CNTL	T1CNT7	T1CNT6	T1CNT5	T1CNT4	T1CNT3	T1CNT2	T1CNT1	T1CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T1CNT[7:0] - T1 计数器低 8 位，为可读写的递减计数器

### 7.3.3 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]
T1LOADH	T1LOAD15	T1LOAD14	T1LOAD13	T1LOAD12	T1LOAD11	T1LOAD10	T1LOAD9	T1LOAD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T1LOAD[15:8] - T1 重载寄存器高 8 位，用于设置 T1 的计数周期

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T1LOADL	T1LOAD7	T1LOAD6	T1LOAD5	T1LOAD4	T1LOAD3	T1LOAD2	T1LOAD1	T1LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T1LOAD[7:0] - T1 重载寄存器低 8 位，用于设置 T1 的计数周期

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。

## 7.4 定时器 T2

定时器 T2 为 8 位定时器，包含 1 个 8 位递减计数器、可编程预分频器、控制寄存器、8 位重载寄存器和 3 个 8 位比较寄存器。

- 可通过预分频器设置时钟频率，可通过重载寄存器控制计数周期；
- 支持 3 路 8 位共周期 PWM 输出，可通过对应的比较寄存器分别设置每路 PWM 占空比；
- 其中 1 路 PWM 可扩展为 1 对 8+3 模式的带死区互补 PWM 输出；
- 支持溢出中断和溢出唤醒功能；

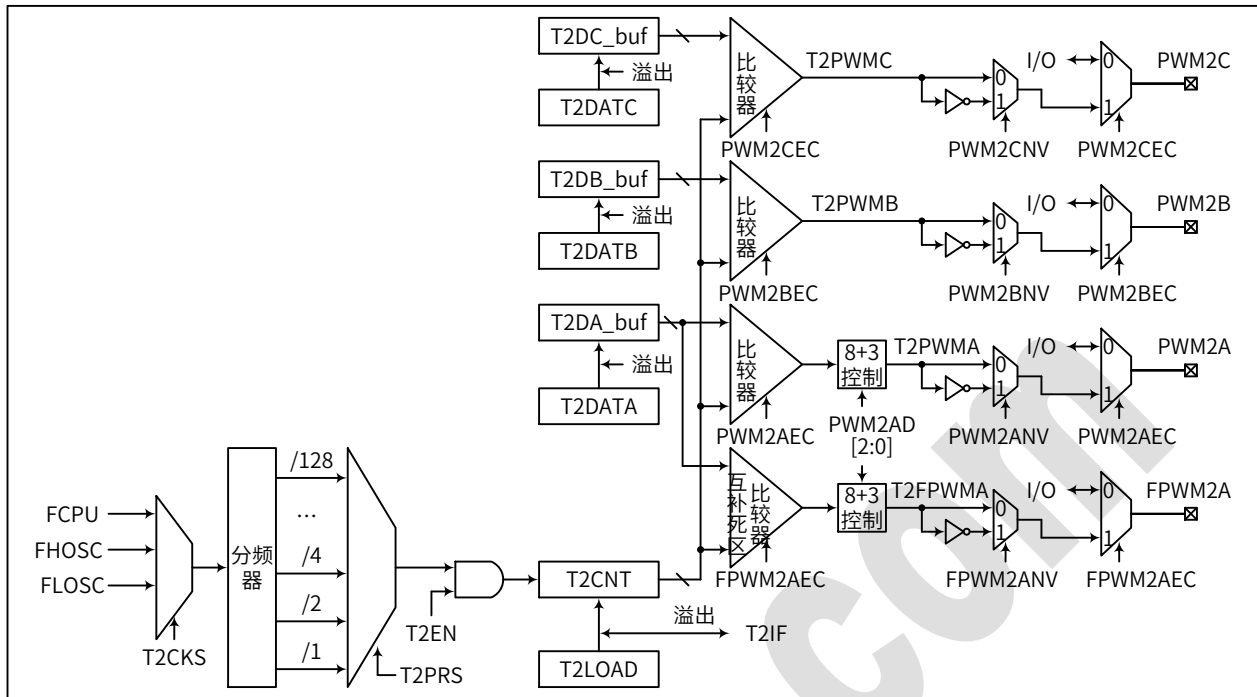


图 7-4 8 位定时器 T2 示意图

定时器 T2，可通过寄存器位 T2CKS 选择时钟源，通过 T2PRS 选择时钟预分频比，所选时钟源通过预分频器后产生 T2 计数器 T2CNT 的计数时钟（上升沿计数）。写 T2CNT 将清零预分频计数器，而预分频比保持不变。

T2EN=0 时，T2CNT 保持不变，写重载寄存器 T2LOAD 将立即载入 T2CNT；T2EN=1 时，T2CNT 递减计数，计数到 0 的时钟结束后产生溢出信号并触发中断，中断标志 T2IF 将被置 1，同时 T2 自动将当前 T2LOAD 值载入 T2CNT 并重新开始计数。

如图所示，定时器 T2 可实现 3 路共周期的 PWM 功能（PWM2x，x=A,B,C，下同），可分别设置每路 PWM 占空比，可通过寄存器位使能或关闭 PWM 功能，并控制端口是否输出 PWM 波形。PWM2x 关闭时 T2PWMx 信号为低电平。PWM2x 使能后 T2CNT 从重载值开始递减计数直到计数溢出为一个 PWM 周期：当计数到与比较寄存器 T2DATx 相等时，T2PWMx 变为高电平；当计数溢出时，T2PWMx 变为低电平。

T2DATx 均配有 1 个 8 位比较缓冲器（T2Dx\_buf）用于与 T2CNT 比较，PWM2x 关闭时写 T2DATx 将立即载入缓冲器中，而 PWM2x 使能后写 T2DATx 则将在 T2 溢出时才载入缓冲器中。若要首个 PWM 周期和占空比准确，需先写重载寄存器和比较寄存器，再使能 PWM，最后开启定时器。

T2PWMx 信号（x=A,B,C，下同）的占空比计算如下：

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

如图所示，定时器 T2 还可实现 1 路与 PWM2A 互补且带死区（2 路互补信号高电平非交叠时间）控制的互补 PWM 功能（FPWM2A），可通过单独的寄存器位使能互补 PWM 功能，并控制端口是否输出互补 PWM 波形。PWM2A 关闭时





T2PWMA 信号为低电平，FPWM2A 关闭时 T2FPWMA 信号为高电平。PWM2A/FPWM2A 使能后 T2CNT 从重载值开始递减计数直到计数溢出为一个 PWM 周期：当计数到与比较寄存器 T2DATA 相等时，T2PWMA 变为高电平（T2FPWMA 提前下降沿非交叠时间变为低电平）；当计数溢出时，T2PWMA 变为低电平（T2FPWMA 滞后上升沿非交叠时间变为高电平）。

T2DATA 的比较缓冲器（T2DA\_buf），PWM2A 和 FPWM2A 均关闭时写 T2DATA 将立即载入缓冲器中，而 PWM2A 或 FPWM2A 使能后写 T2DATA 则将在 T2 溢出时才载入缓冲器中。若要首个 PWM 周期和占空比准确，需先写重载寄存器和比较寄存器，再使能 PWM，最后开启定时器。

PWM2A/FPWM2A 支持 8+3 模式，即在普通模式 PWM 的时钟下，以每 8 个 PWM 周期作为一个大周期，其中部分周期进行占空比延展（即延展高电平时间），这些延展周期内 T2PWMA 信号将比普通周期提前半个计数时钟变为高电平，即 T2PWMA 高电平时间延展为（T2DATA+0.5）。

8+3 模式提高的是整体上的 PWM 占空比调节精度，而不是单个 PWM 周期的占空比调节精度。占空比延展控制位 PWM2AD[2:0] 决定每 8 个周期内哪几个周期为普通周期，哪几个周期为延展周期。

PWM2A/FPWM2A 的占空比可通过寄存器位 PWM2ADB 微调半个计数时钟周期。PWM2ADB=1 时，普通周期内 T2PWMA 信号在计数时钟上升沿变为高电平，而延展周期内则提前半个时钟周期在时钟下降沿翻转；PWM2ADB=0 时，普通周期内 T2PWMA 信号滞后半个时钟周期在时钟下降沿变为高电平，而延展周期内则变为正常的时钟上升沿翻转。T2FPWMA 信号变为低电平的时间则在叠加下降沿死区时间后进行相应微调。

PWM2A 的 8+3 模式一个大周期的占空比计算如下（需  $0 < T2DATA < T2LOAD$ ）：

- 高电平时间 =  $[(T2DATA + (PWM2ADB - 1)/2) \times 8 + (PWM2AD[2:0])/2] \times T2CNT$  计数时钟周期
- 周期 =  $(T2LOAD + 1) \times 8 \times T2CNT$  计数时钟周期
- 占空比（高电平时间/周期） =  $[T2DATA + (PWM2ADB - 1)/2 + (PWM2AD[2:0]/16)] / (T2LOAD + 1)$

举例说明（PWM2ADB=1）：

若 PWM2AD[2:0]=B' 000，则 8 个 PWM 周期均不进行占空比延展，一个大周期 PWM 的占空比与普通模式 PWM 的占空比一样，为  $[(T2DATA) / (T2LOAD + 1)]$ ；

若 PWM2AD[2:0]=B' 101，则每 8 个 PWM 周期中的第 1、2、3、5、7 个周期（共 5 个周期）进行占空比延展，一个大周期的占空比为  $[(T2DATA) + (5/16)] / (T2LOAD + 1)$ ；

#### 7.4.1 定时器 T2 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T2CR	T2EN	PWM2AEC	FPWM2AEC	T2CKS1	T2CKS0	T2PRS2	T2PRS1	T2PRS0
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] T2EN – 定时器 T2 使能位

0：关闭定时器 T2；

1：开启定时器 T2；

BIT[6] PWM2AEC – PWM2A 使能位及端口输出控制位



0: 关闭 PWM2A 功能，并禁止端口输出脉宽调制波形；

1: 使能 PWM2A 功能，并允许端口输出脉宽调制波形；

BIT[5] FPWM2AEC – FPWM2A 使能位及端口输出控制位

0: 关闭 FPWM2A 功能，并禁止端口输出脉宽调制波形；

1: 使能 FPWM2A 功能，并允许端口输出脉宽调制波形；

BIT[4:3] T2CKS[1:0] – T2 时钟源选择位

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

BIT[2:0] T2PRS[2:0] – T2 时钟预分频比选择位

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

#### 7.4.2 定时器 T2 计数器

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

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

#### 7.4.3 定时器 T2 重载寄存器

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

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

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。



## 7.4.4 定时器 T2 比较寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T2DATA	T2DATA7	T2DATA6	T2DATA5	T2DATA4	T2DATA3	T2DATA2	T2DATA1	T2DATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] T2DATA[7:0] – T2 比较寄存器 A，用于设置 PWM2A 的占空比

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T2DATB	T2DATB7	T2DATB6	T2DATB5	T2DATB4	T2DATB3	T2DATB2	T2DATB1	T2DATB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] T2DATB[7:0] – T2 比较寄存器 B，用于设置 PWM2B 的占空比

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T2DATC	T2DATC7	T2DATC6	T2DATC5	T2DATC4	T2DATC3	T2DATC2	T2DATC1	T2DATC0
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] T2DATC[7:0] – T2 比较寄存器 C，用于设置 PWM2C 的占空比

## 7.4.5 PWM2 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PWM2CR0	PWM2CNV	PWM2BNV	PWM2ANV	FPWM2ANV	-	FPWM2ACH S	PWM2CEC	PWM2BEC
R/W	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W
初始值	0	0	0	0	-	0	0	0

BIT[7] PWM2CNV – PWM2C 端口输出取反控制位

- 0: 端口输出正向波形;
- 1: 端口对电平取反后输出;

BIT[6] PWM2BNV – PWM2B 端口输出取反控制位

- 0: 端口输出正向波形;
- 1: 端口对电平取反后输出;

BIT[5] PWM2ANV – PWM2A 端口输出取反控制位

- 0: 端口输出正向波形;
- 1: 端口对电平取反后输出;

BIT[4] FPWM2ANV – FPWM2A 端口输出取反控制位

- 0: 端口输出正向波形;
- 1: 端口对电平取反后输出;

BIT[2] FPWM2ACHS – FPWM2A 输出通道选择位 (FPWM2A 功能仅对被选定通道有效)



0: FPWM2A 输出通道为 FPWM2AO0 (P15);

1: FPWM2A 输出通道为 FPWM2AO1 (P22);

BIT[1] PWM2CEC – PWM2C 使能位及端口输出控制位

0: 关闭 PWM2C 功能, 并禁止端口输出脉宽调制波形;

1: 使能 PWM2C 功能, 并允许端口输出脉宽调制波形;

BIT[0] PWM2BEC – PWM2B 使能位及端口输出控制位

0: 关闭 PWM2B 功能, 并禁止端口输出脉宽调制波形;

1: 使能 PWM2B 功能, 并允许端口输出脉宽调制波形;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PWM2CR1	-	-	FPWM2AE5	FPWM2AE4	FPWM2AE3	FPWM2AE2	FPWM2AE1	FPWM2AE0
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] FPWM2AE[5:0] – T2FPWMA 上升/下降沿非交叠时间选择位

FPWM2AE[5:0]	上升沿非交叠时间	下降沿非交叠时间
00 0000	1 个计数时钟周期	1 个计数时钟周期
00 0001	2 个计数时钟周期	2 个计数时钟周期
---	---	---
11 1111	64 个计数时钟周期	64 个计数时钟周期

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PWM2CR2	-	-	-	-	PWM2ADB	PWM2AD2	PWM2AD1	PWM2AD0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	1	0	0	0

BIT[3] PWM2ADB – T2PWMA/T2FPWMA 匹配翻转时钟沿选择位

0: 占空比匹配时, 脉宽调制信号电平滞后半个时钟周期翻转;

1: 占空比匹配时, 脉宽调制信号电平与计数时钟的有效沿同步翻转;

BIT[2:0] PWM2AD[2:0] – T2PWMA/T2FPWMA 占空比延展控制位

PWM2AD[2:0]	每 8 个周期中延展周期选择
000	0 个周期延展
001	1 个周期(第 1 个周期)延展, 7 个周期(第 2,3,4,5,6,7,8 个周期)不延展
010	2 个周期(第 1,5 个周期)延展, 6 个周期(第 2,3,4,6,7,8 个周期)不延展
011	3 个周期(第 1,3,5 个周期)延展, 5 个周期(第 2,4,6,7,8 个周期)不延展
100	4 个周期(第 1,3,5,7 个周期)延展, 4 个周期(第 2,4,6,8 个周期)不延展
101	5 个周期(第 1,2,3,5,7 个周期)延展, 3 个周期(第 4,6,8 个周期)不延展
110	6 个周期(第 1,2,3,5,6,7 个周期)延展, 2 个周期(第 4,8 个周期)不延展
111	7 个周期(第 1,2,3,4,5,6,7 个周期)延展, 1 个周期(第 8 个周期)不延展

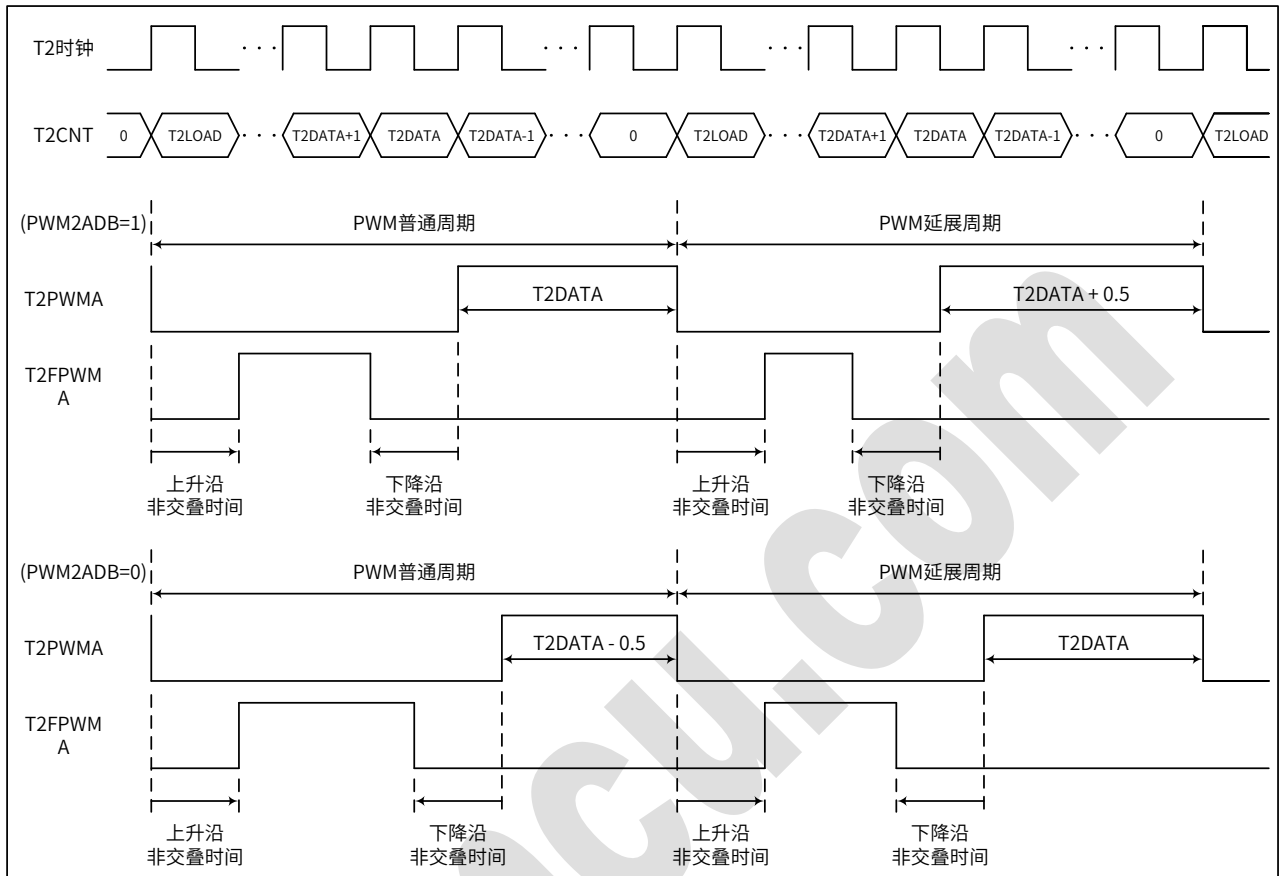


图 7-5 PWM2ADB 及 8+3 模式波形示意

注:

- 1、应用互补 PWM 时，前后死区的总时间应小于  $T2PWMA$  低电平时间，以确保  $T2FPWMA$  能正常生成高电平；
- 2、不可在 PWM 工作时调整 PWM 周期（即定时器周期）和死区时间；
- 3、8+3 模式下，当  $PWM2ADB=1$  时，正常周期（不延展的周期）的占空比（高电平时间）为  $T2DATA$ ，而延展周期的占空比则延展半个时钟周期（即为  $T2DATA+0.5$ ）；当  $PWM2ADB=0$  时，所有周期（包括正常周期和延展周期）的占空比均缩减半个时钟周期；
- 4、若  $T2PWMA$  或  $T2FPWMA$  在  $PWM2ADB$  为 1 时高电平时间仅为 1 个时钟周期，则  $PWM2ADB$  清 0 操作并不会将高电平时间缩减为半个时钟周期；
- 5、因  $F_{CPU}$  的占空比不为 50%，所以当 PWM 计数时钟为  $F_{CPU}$  的 1 分频时，微调功能的翻转延迟并不正好为半个时钟周期，而是延迟为  $F_{CPU}$  的高电平时间；



## 8 模数转换器 ADC

### 8.1 ADC 概述

芯片内置 1 个 12 位高精度逐次逼近型的模数转换器 ADC。

- 13 路外部通道：AN0~AN12；3 路内部通道：GND、VDD/4、OP2\_OUT；
- 参考电压可选：VDD、内部参考电压  $V_{IR}$  (2V/3V/4V)、外部参考电压  $V_{ER}$  ( $V_{ER1}$  输入)；
- ADC 时钟：FHIRC 的 16/32/64/128 分频；
- 支持零点校准；

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

ADC 的采样 (SAMPLE) 时间可选择 2/4/8/15 个 ADCLK (即 ADC 时钟周期)，转换 (CONVERT) 时间固定为 12 个 ADCLK，一次 ADC 转换的时间为 14/16/20/27 个 ADCLK。

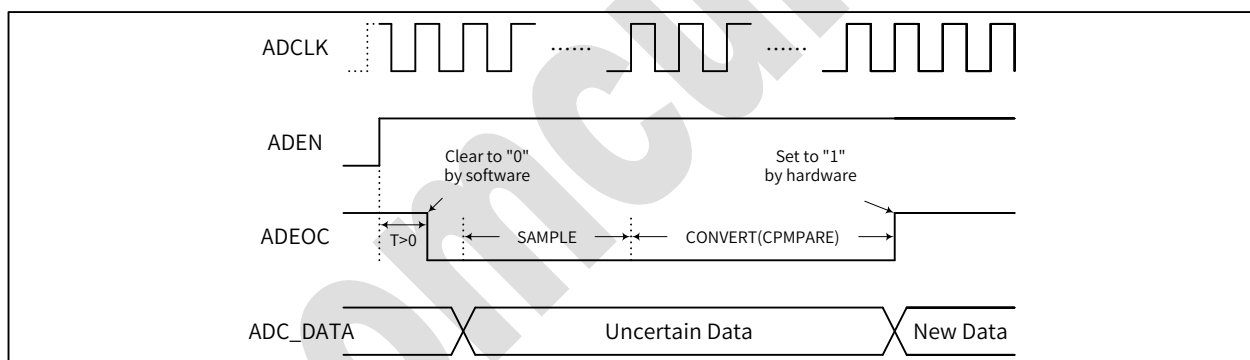


图 8-1 ADC 转换时序

注：

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



## 8.2 ADC 相关寄存器

### 8.2.1 ADC 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADCR0	ADEN	ADEOC	ADCKS1	ADCKS0	ADCCHS3	ADCCHS2	ADCCHS1	ADCCHS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	1	1	1	1	1	1	1

BIT[7] ADEN – ADC 使能位

0: 关闭 ADC;

1: 开启 ADC;

BIT[6] ADEOC – AD 转换控制位

0: AD 转换中, 完成后自动置 1;

1: 转换未开始或已完成, 写 0 开始 AD 转换;

BIT[5:4] ADCKS[1:0] – ADC 转换时钟选择位

ADCKS[1:0]	ADC 转换时钟 $F_{ADC}$
00	$F_{HIRC}/16$
01	$F_{HIRC}/32$
10	$F_{HIRC}/64$
11	$F_{HIRC}/128$

BIT[3:0] ADCCHS[3:0] – ADC 模拟输入通道选择位

ADCCHS[3:0]	ADC 模拟输入通道	ADCCHS[3:0]	ADC 模拟输入通道
0000	AN0	1000	AN8
0001	AN1	1001	AN9
0010	AN2	1010	AN10
0011	AN3	1011	AN11
0100	AN4	1100	AN12
0101	AN5	1101	OP2_OUT
0110	AN6	1110	VDD/4
0111	AN7	1111	GND

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADCR1	ADRSEL	ADVRS2	ADVRS1	ADVRS0	-	-	ADSPS1	ADSPS0
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
初始值	0	0	0	0	-	-	0	0

BIT[7] ADRSEL – ADC 转换结果数据格式选择位

0: ADC 转换结果为 12 位数据, 高 8 位存入 ADRH[7:0]、低 4 位存入 ADRL[3:0];

1: ADC 转换结果为 12 位数据, 高 4 位存入 ADRH[3:0]、低 8 位存入 ADRL[7:0];

BIT[6] ADVRS2 – ADC 参考电压选择位



0: ADC 参考电压由 ADVRS[1:0]决定;

1: ADC 参考电压为外部参考电压 VER (端口 VER1 输入电压);

注: 应用外部参考电压时, 需先关闭相应端口的数字 I/O 功能及其内部上/下拉电阻。

BIT[5:4] ADVRS[1:0] – ADC 参考电压选择位

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

BIT[1:0] ADSPS[1:0] – ADC 采样时间选择位

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

## 8.2.2 ADC 转换结果寄存器

ADRSEL=0:

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

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

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

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

ADRSEL=1:

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

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





	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADRL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
R/W	R	R	R	R	R	R	R	R
初始值	X	X	X	X	X	X	X	X

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

### 8.2.3 ADC 零点偏移修调控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OSADJCR	OSADJEN	OSADJTD	OSADJT5	OSADJT4	OSADJT3	OSADJT2	OSADJT1	OSADJT0
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]      OSADJEN – ADC 零点偏移修调使能位

- 0: ADC 零点偏移修调无效;
- 1: ADC 零点偏移修调有效;

BIT[6]      OSADJTD – ADC 零点偏移修调方向选择位

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

BIT[5:0]      OSADJT[5:0] – ADC 零点偏移修调电压选择位

OSADJT[5:0]	修调电压 (典型值)
00 0000	0
00 0001	$1 \times V_{REF}/4096$
00 0010	$2 \times V_{REF}/4096$
---	---
00 1111	$15 \times V_{REF}/4096$
01 0000	$16 \times V_{REF}/4096$
---	---
11 1110	$62 \times V_{REF}/4096$
11 1111	$63 \times V_{REF}/4096$

## 8.3 ADC 操作步骤

模数转换操作步骤:

- (1) 设置相应端口为输入端口, 关闭端口的内部上/下拉电阻;
- (2) 通过端口数模控制寄存器, 关闭相应端口的数字 I/O 功能;
- (3) 若转换时钟可选, 则设置 ADCKS, 选择适当的转换时钟;



- (4) 若采样时间可选，则设置 ADSPS，选择适当的采样时间；
- (5) 若参考电压可选，则设置 ADVRS，选择适当的参考电压；
- (6) 若数据格式可选，则设置 ADRSEL，选择 ADC 转换结果的数据格式；
- (7) ADEN 置 1，使能 ADC 模块；
- (8) 设置 ADCHS，选择 ADC 转换通道；
- (9) 延时等待电路稳定后，ADEOC 写 0，启动 AD 转换；
- (10) 等待 ADEOC 硬件置 1（或利用 ADC 中断）；
- (11) 读取 ADC 转换结果（ADRH/ADRL）；
- (12) 重复执行（8）~（11），对不同的通道进行转换或对同一通道进行多次转换；

#### 8.4 ADC 零点偏移修调流程

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



## 9 运算放大器 OPA

### 9.1 OPA 概述

芯片内置 3 个运算放大器 OP0~OP2，共模输入范围为 0~ (VDD-1.4V)。

- OP0/OP1 为开漏输出型，共用一个输出端口，且负端输入及输出均为纯模拟端口；
- OP0 正端输入可选 14 位的 VDD 内部分压电压、或外部输入电压；
- OP1 正端输入可选 12 位的 VDD 内部分压电压、或外部输入电压；
- OP2 支持失调电压自消除调校（调校精度 $\pm 2\text{mV}$ ），内置增益电路，放大倍数为 10/20 倍；
- OP2 的输出信号 OP2\_OUT 可作为 ADC 输入；

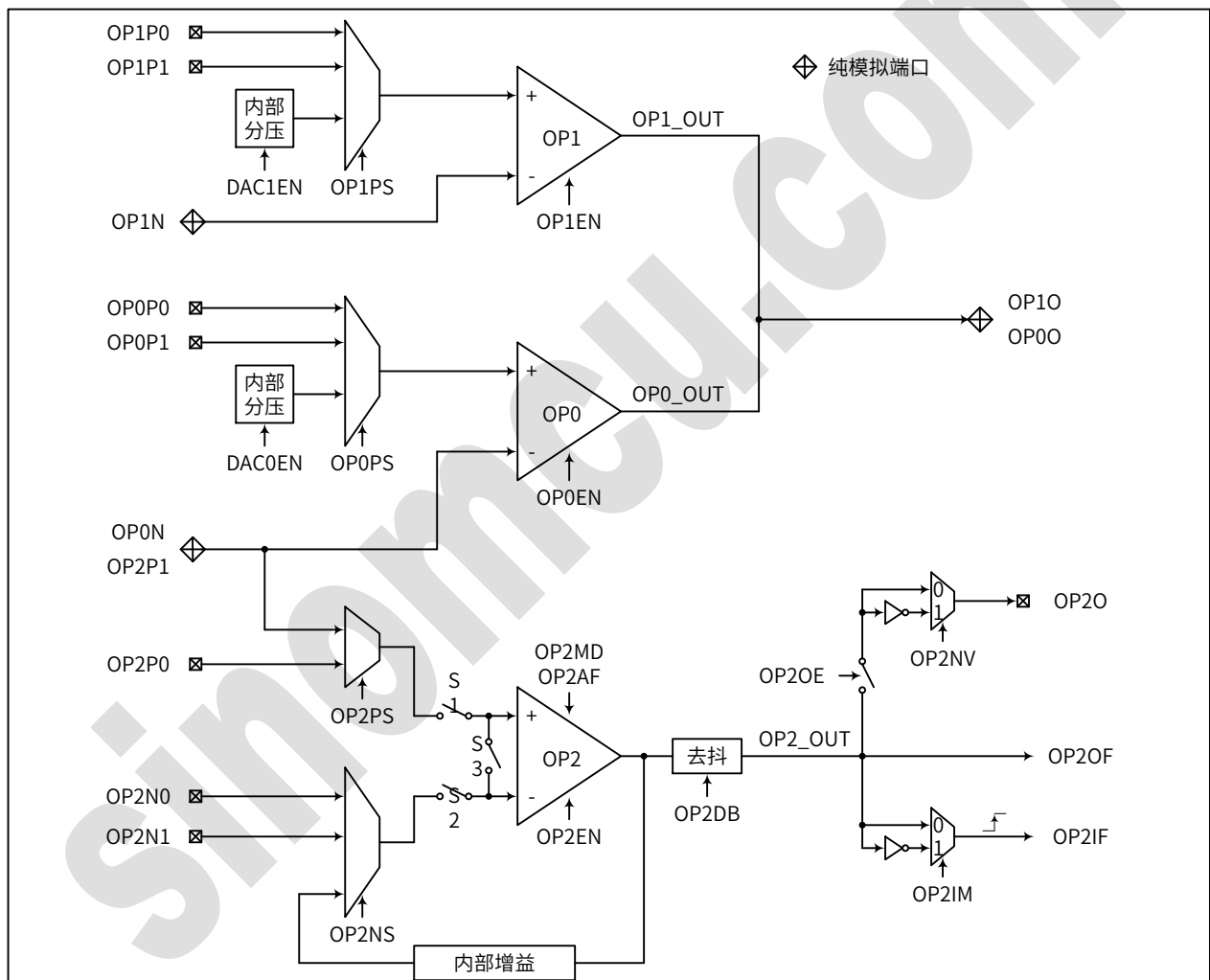


图 9-1 OPA 模块框图

运算放大器 OP2 支持比较器模式，当工作于比较器模式时，输出端信号可通过 OP2AF 选择模拟滤波时间，通过 OP2DB 选择数字去抖处理，去抖后的输出信号 OP2\_OUT 可从端口输出且支持输出取反，并可选择上升沿或下降沿触发 OP2 比较中断。

注：



- 1、OP2 工作于运算放大器模式时，其输出使能控制依然有效，但其输出去抖、输出取反、以及 OP2 比较中断等功能则均无效；
- 2、OP2 工作于比较器模式且选择输出数字去抖时，在低速模式 (HFEN=0)、HOLD2 模式、休眠模式下将因去抖采样时钟 FHOSC 暂停而导致输出电平锁定，从而无法触发中断以及唤醒 CPU；

## 9.2 OPA 相关寄存器

### 9.2.1 OPA 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OPCR0	OP0EN	DAC0EN	OP0PS1	OP0PS0	OP1EN	DAC1EN	OP1PS1	OP1PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	1	1	1	0	1	1	1

BIT[7] OP0EN – 运算放大器 OP0 使能位（仅在配置字 OPAENSET 设为寄存器位控制时有效）

0：关闭 OP0 和 DAC0（此时 OP0 输出处于浮空状态）；

1：开启 OP0；

BIT[6] DAC0EN – DAC0 使能位（仅 OP0 开启时有效）

0：关闭 DAC0；

1：开启 DAC0；

BIT[5:4] OP0PS[1:0] – OP0 正端输入选择位

OP0PS[1:0]	OP0 正端输入
00	外部 OP0P0 输入
01	外部 OP0P1 输入
1X	VDD 内部分压电压

BIT[3] OP1EN – 运算放大器 OP1 使能位（仅在配置字 OPAENSET 设为寄存器位控制时有效）

0：关闭 OP1 和 DAC1（此时 OP1 输出处于浮空状态）；

1：开启 OP1；

BIT[2] DAC1EN – DAC1 使能位（仅 OP1 开启时有效）

0：关闭 DAC1；

1：开启 DAC1；

BIT[1:0] OP1PS[1:0] – OP1 正端输入选择位

OP1PS[1:0]	OP1 正端输入
00	外部 OP1P0 输入
01	外部 OP1P1 输入
1X	VDD 内部分压电压

注：OPn 与 DACn (n=0,1) 的开关控制如下表所示：



OPAENSET	OPnEN	DACnEN	OPn	DACn
OPn 始终开启	X	0	开启	关闭
		1	开启	开启
OPn 是否开启 由寄存器控制	1	0	关闭	关闭
		0	开启	关闭
		1	开启	开启

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OPCR1	OP2EN	OP2MD	OP2PS	OP2NS1	OP2NS0	-	-	OP2OE
R/W	R/W	R/W	R/W	R/W	R/W	-	-	R/W
初始值	0	0	0	0	0	-	-	0

BIT[7] OP2EN – 运算放大器 OP2 使能位

0: 关闭 OP2;

1: 开启 OP2;

BIT[6] OP2MD – OP2 工作模式选择位

0: 运算放大器模式, 输出为模拟信号;

1: 比较器模式, 输出为数字信号;

BIT[5] OP2PS – OP2 正端输入选择位

0: OP2 正端输入为端口 OP2P0 输入电压;

1: OP2 正端输入为端口 OP2P1 输入电压;

BIT[4:3] OP2NS[1:0] – OP2 负端输入选择位

OP2NS[1:0]	OP2 负端输入
00	外部 OP2N0 输入
01	外部 OP2N1 输入
10	内部 10 倍增益电路
11	内部 20 倍增益电路

BIT[0] OP2OE – OP2O 端口输出使能位

0: 禁止端口输出 OP2\_OUT 信号;

1: 允许端口输出 OP2\_OUT 信号;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OPCR2	-	OP2AF	OP2DB2	OP2DB1	OP2DB0	OP2NV	OP2IM	OP2OF
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R
初始值	-	0	0	0	0	0	0	X

BIT[6] OP2AF – OP2 比较模式输出模拟滤波时间选择位 (仅 OP2MD=1 时有效)

0: 50ns (典型值) 模拟滤波;

1: 10ns (典型值) 模拟滤波;



BIT[5:3] OP2DB[2:0] – OP2 输出去抖时间选择位（仅 OP2MD=1 时有效）

OP2DB[2:0]	OP2 输出去抖时间
000	0（不去抖）
001	数字去抖：2 个 F <sub>HOSC</sub> 时钟周期
010	数字去抖：4 个 F <sub>HOSC</sub> 时钟周期
011	数字去抖：8 个 F <sub>HOSC</sub> 时钟周期
100	数字去抖：16 个 F <sub>HOSC</sub> 时钟周期
101	数字去抖：24 个 F <sub>HOSC</sub> 时钟周期
110	数字去抖：32 个 F <sub>HOSC</sub> 时钟周期
111	数字去抖：64 个 F <sub>HOSC</sub> 时钟周期

BIT[2] OP2NV – OP2O 端口输出取反控制位（仅 OP2MD=1 时有效）

0: 端口输出 OP2 比较结果的正向电平；

1: 端口对 OP2 比较结果取反后输出；

BIT[1] OP2IM – OP2 中断触发方式选择位（仅 OP2MD=1 时有效）

0: OP2 输出信号 OP2\_OUT 的上升沿触发中断；

1: OP2 输出信号 OP2\_OUT 的下降沿触发中断；

BIT[0] OP2OF – OP2 输出状态标志位

0: OP2 输出为低，即 OP2 正端输入电压低于负端输入电压；或 OP2 关闭；

1: OP2 输出为高，即 OP2 正端输入电压高于负端输入电压；

### 9.2.2 OP0/OP1 分压控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OP0VRH	-	-	OP0PVS13	OP0PVS12	OP0PVS11	OP0PVS10	OP0PVS9	OP0PVS8
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] OP0PVS[13:8] – OP0 正端内部分压电压选择位高 6 位

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OP0VRL	OP0PVS7	OP0PVS6	OP0PVS5	OP0PVS4	OP0PVS3	OP0PVS2	OP0PVS1	OP0PVS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	1	0	0	0	0	0	0

BIT[7:0] OP0PVS[7:0] – OP0 正端内部分压电压选择位低 8 位

OP0PVS[13:0]	OP0 正端内部分压电压
0000H	0
0001H	1 × V <sub>DD</sub> /16384
0002H	2 × V <sub>DD</sub> /16384



0003H	$3 \times VDD/16384$
---	---
3FFFH	$16383 \times VDD/16384$

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OP1VRH	-	-	-	-	OP1PVS11	OP1PVS10	OP1PVS9	OP1PVS8
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	1	0	0	0

BIT[3:0] OP1PVS[11:8] – OP1 正端内部分压电压选择位高 4 位

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OP1VRL	OP1PVS7	OP1PVS6	OP1PVS5	OP1PVS4	OP1PVS3	OP1PVS2	OP1PVS1	OP1PVS0
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] OP1PVS[7:0] – OP1 正端内部分压电压选择位低 8 位

OP1PVS[11:0]	OP1 正端内部分压电压
0000H	0
0001H	$1 \times VDD/4096$
0002H	$2 \times VDD/4096$
0003H	$3 \times VDD/4096$
---	---
0FFFH	$4095 \times VDD/4096$

注 ( $n=0,1$ ) :

- 1、仅当 OPn 开启且其正端输入选择 VDD 内部分压时，OPn 内部分压电路才开启，否则将自动关闭；
- 2、OPnVRH 配有缓冲器，读写 OPnVRH 实际为读写缓冲器内容，OPnVRH 仅在读写 OPnVRL 时才自动与缓冲器交互，因此写 OPnVR 需先写 OPnVRH 再写 OPnVRL，读 OPnVR 需先读 OPnVRL 再读 OPnVRH；

### 9.2.3 OP2 自消除寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OP2CLR	OP2CLRE	OP2CLRS	OP2CLR5	OP2CLR4	OP2CLR3	OP2CLR2	OP2CLR1	OP2CLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	1	0	0	0	0	0

BIT[7] OP2CLRE – OP2 自消除模式使能位

- 0: 关闭 OP2 自消除模式；  
1: 使能 OP2 自消除模式；



BIT[6] OP2CLRS – OP2 自消除参考源选择位

0: OP2 自消除参考源为其负端输入;

1: OP2 自消除参考源为其正端输入;

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

BIT[5:0] OP2CLR[5:0] – OP2 自消除校准位

### 9.3 OPA 失调电压调校流程

- (1) 设置 OP2CLRE=1, 使能失调电压自消除模式;
- (2) 设置 OP2CLRS, 选择 OP2 自消除参考源;
- (3) 设置 OP2CLR[5:0]=00H, 等待至少 100 $\mu$ s 后读取标志位 OP2OF;
- (4) OP2CLR[5:0]自加 1, 等待至少 100 $\mu$ s 后重新读取 OP2OF:
  - 若 OP2OF 发生翻转, 则表示已调校到位;
  - 若 OP2CLR[5:0]=3FH 而 OP2OF 仍未翻转, 则表示共模输入超出范围, 调校失败;
- (5) 循环执行 (4), 直到调校到位或调校失败;
- (6) 判断当前调校输出方向:
  - 若 OP2CLR[5]=1, 则 OP2CLR[4:0]保持当前值不变;
  - 若 OP2CLR[5]=0, 则 OP2CLR[4:0]减 1 作为校准值;
- (7) 设置 OP2CLRE=0, 关闭自消除模式;





## 10 低电压检测 LVD

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

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

### 10.1.1 LVD 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
LVDCR	LVDEN	-	LVDVS2	LVDVS1	LVDVS0	-	-	LVDF
R/W	R/W	-	R/W	R/W	R/W	-	-	R
初始值	0	-	0	0	0	-	-	X

BIT[7] LVDEN – 低电压检测 LVD 使能位

0：关闭 LVD；

1：开启 LVD；

BIT[5:3] LVDVS[2:0] – LVD 电压检测阈值选择位

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

BIT[0] LVDF – LVD 检测状态标志位

0：VDD 电压高于电压检测阈值，或 LVD 关闭；

1：VDD 电压低于电压检测阈值；



## 11 异步通讯 UART

### 11.1 UART 概述

芯片内置 1 个通用异步收发器 UART，包含接收缓冲器和锁存寄存器、发送缓冲器和移位寄存器、波特率发生器、时钟预分频器和计数器等，可实现 8 位/9 位数据（含校验位）、1 位/2 位停止位的双线全双工或单线半双工等模式的异步通讯。

注：UART 发送完 1 位/2 位的停止位后，将立即发送或接收下一帧的起始位，前后两帧之间无空闲位。

8 位的接收缓冲器（URTRD\_buf）和发送缓冲器（URTTD\_buf）均需通过数据寄存器 URTDR 进行读/写操作，写 URTDR 为写 URTTD\_buf，读 URTDR 则为读 URTRD\_buf。

对 URTDR 进行写操作后将启动 UART 发送。当发送端口 TX 空闲时，新写入 URTTD\_buf 中的数据将移入移位寄存器，中断标志 TXBIF 将被置 1 触发 UART 发送缓冲器空中断，同时 TX 端口开始发送：首先发送起始位，然后发送移位寄存器中内容（低位先发），再发送 TXB8 中内容（若有第 9 位），最后发送停止位。停止位发送完成后，中断标志 TXIF 将被置 1 触发 UART 发送完成中断。

若 URTTD\_buf 非空时对 URTDR 进行写操作（不包括 TXB8 写操作），则发送冲突标志 TXCL 将被置 1，新写入的数据将被忽略。

UREN 置 1 将使能 UART 接收。当接收端口 RX 检测到起始信号有效沿时启动 UART 接收，串行接收到的数据先缓存在锁存寄存器中，当接收完成后再移入 URTRD\_buf 和 RXB8（若有第 9 位）中，第 1 个停止位接收完成后，中断标志 RXIF 将被置 1 触发 UART 接收完成中断。

若 URTRD\_buf 中内容被读取前锁存寄存器又接收完新的数据，则接收溢出标志 RXOV 将被置 1，新接收的数据（包括第 9 位）将被忽略。若检测到起始位或第 1 个停止位无效（即有效电平保持时间过短），则帧出错标志 FERR 将被置 1。

UART 可通过使能位 UPEN 使能或关闭奇偶校验。当使能奇偶校验时，8 位数据格式的第 8 位则为校验位，而 9 位数据格式的第 9 位为校验位，发送时会待发送数据的最后 1 位替换为校验值发送，接收时会接收的数据进行奇偶计算并与最后 1 位比对，若出错则奇偶校验错误标志 PERR 将被置 1。

可通过校验模式选择位 UPMD 选择奇校验或偶校验。奇校验模式下，发送时校验码为使得有效数据加校验码中“1”的个数为奇数的值，而接收时则检测有效数据加校验码中“1”的个数是否为奇数（为奇数则校验通过，为偶数则校验出错）；偶校验模式下，发送时校验码为使得有效数据加校验码中“1”的个数为偶数的值，接收时则检测有效数据加校验码中“1”的个数是否为偶数（为偶数则校验通过，为奇数则校验出错）。

例如，8 位有效数据 B' 00100101，奇校验时校验码应为“0”，偶校验时校验码应为“1”；而 7 位有效数据 B' 0010010，奇校验时校验码应为“1”，偶校验时校验码应为“0”。

当检测到发送冲突、接收溢出、帧出错或奇偶校验错等错误后，相应标志将被置 1，即使后续数据传输中再未发生任何错误，标志也不会自动清除，仅能通过软件清 0。



## 11.2 UART 工作模式与传输格式

UART 支持双线模式和单线模式。双线模式可通过寄存器位 URTCHS 选择 TX0/RX0 或 TX1/RX1 作为一对发送/接收端口，可实现全双工或半双工通讯。单线模式可通过 URTCHS 和 URTPTS 选择 TX0/RX0/TX1/RX1 中一路作为发送/接收端口，仅可实现半双工通讯。

UART 还可通过 URTLNV 选择端口输入/输出信号的电平高/低与逻辑“1” / “0”的对应关系。

UART 可通过寄存器位 UDATF 选择 8 位或 9 位的数据格式、通过 USTPF 选择 1 位或 2 位停止位、通过 UPEN 使能或关闭奇偶校验，从而实现多种数据传输格式。

表 11-1 UART 模式配置

数据格式	奇偶校验	停止位格式	传输格式					
			起始位	有效数据位	校验码	RXB8/TXB8	停止位	帧长度
0	0	0	1 位	8 位	0 位	无意义	1 位	10 位
0	0	1	1 位	8 位	0 位	无意义	2 位	11 位
0	1	0	1 位	7 位	1 位	无意义	1 位	10 位
0	1	1	1 位	7 位	1 位	无意义	2 位	11 位
1	0	0	1 位	9 位	0 位	第 9 位有效数据	1 位	11 位
1	0	1	1 位	9 位	0 位	第 9 位有效数据	2 位	12 位
1	1	0	1 位	8 位	1 位	接收到的校验码 发送时无意义	1 位	11 位
1	1	1	1 位	8 位	1 位	接收到的校验码 发送时无意义	2 位	12 位

## 11.3 UART 波特率

UART 内置波特率发生器，时钟源为内部系统高频时钟 FHOSC（若高于 16MHz 则将选择其不超过 16MHz 的分频时钟作为时钟源），可通过寄存器位 URTPRS 选择时钟预分频比，通过寄存器 URTBR 设置计数器计数初值。UART 使能后，计数器从初值开始递减计数，计数到 0 的时钟结束后产生溢出信号并重置为初值，UART 通讯的波特率即为计数器的溢出频率。

波特率可用以下公式计算（Fuart 为 UART 时钟源频率）：
$$\text{BaudRate} = \frac{F_{\text{uart}}}{(URTBR+1) \times 2^{URTPRS}} \circ$$

表 11-2 波特率配置表

时钟源频率	URTPRS	预分频	URTBR	波特率计算值 (bps)	波特率目标值 (bps)	偏差 (%)
16MHz	110	64 分频	207	1201.92	1200	0.16
	101	32 分频	207	2403.85	2400	0.16



	101	32 分频	146	3401.36	3400	0.04
	100	16 分频	207	4807.69	4800	0.16
	011	8 分频	207	9615.38	9600	0.16
	011	8 分频	138	14388.49	14400	-0.08
	010	4 分频	207	19230.77	19200	0.16
	001	2 分频	207	38461.54	38400	0.16
	001	2 分频	138	57553.96	57600	-0.08
	000	1 分频	138	115107.91	115200	-0.08
	000	1 分频	41	380952.38	384000	-0.79
	000	1 分频	16	941176.47	921600	2.12

注：波特率寄存器 URTBR 的值不能小于 16，否则将无法正确产生波特率。

## 11.4 UART 多机通讯

UART 支持从机地址自动识别，并支持端口在强制输出和被动输出之间的自动转换，可应用于多机通讯系统。在多机通讯系统中，UART 通过每一帧传输内容的第 9 位区分地址帧和数据帧，地址帧的第 9 位为 1，数据帧的第 9 位则为 0。

当主机与多个从机中的一个从机进行通讯时，需先发送一帧地址帧，以激活目标从机，且在再次发送地址帧之前，需确保发送的数据帧均为对目标从机传输的内容。

从机需先开启地址自动识别功能，若接收到数据帧、或接收的地址帧中地址不匹配，从机不响应；当接收到地址帧且地址匹配，从机响应后需关闭地址自动识别功能以接收主机后续传输的数据帧。

### 11.4.1 从机地址自动识别

当数据格式选择 9 位数据并关闭奇偶校验后，可通过地址检测使能位 UAEN 开启从机地址自动识别功能，此时 UART 仅在接收的第 9 位 (RXB8) 为 1 表明接收到地址帧且地址与本机预设地址匹配时，才会触发接收完成中断。若 UART 接收到数据帧、或接收到地址帧的地址不匹配，则不触发中断。

所有从机在等待接收地址帧时，为确保仅在接收地址帧时触发中断，UAEN 需预先置 1。中断触发后，地址匹配的从机需将 UAEN 清 0，以便继续接收数据帧；地址不匹配的从机则不作响应，将继续等待接收与其预设地址匹配的地址帧。一旦全部内容接收完毕，地址匹配的从机需再次将 UAEN 置 1，以屏蔽后续传输的地址不匹配的地址帧和主机发送给其他从机的数据帧，直到接收到下一个地址匹配的地址帧。

### 11.4.2 预设地址和广播地址

使用自动地址识别功能时，主机可通过商定的从机地址选择与一个或多个从机通讯，也可使用商定的广播地址联系所有从机。

从机可通过寄存器 URTAR 和 URTMR 预设本机地址。从机地址为 8 位，预设于地址寄存器 URTAR 中，地址掩码寄



寄存器 URTMR 的每一位则决定 URTAR 中对应的寄存器位在检测地址时是否参与比对,若 URTMR 中某一位为 0,则 URTAR 中相应位将被忽略(即不参与比对,默认为匹配),若 URTMR 中某一位为 1,则 URTAR 中相应位将参与地址比对,这将使从机可在不改变 URTAR 中预设地址的情况下灵活的响应多个地址,也使主机可使用特定地址识别一部分从机而排除其他从机。

从机还可识别广播地址,地址码为 URTAR 和 URTMR 的逻辑或,结果中的 0 表示该位被忽略。主机可通过广播地址与所有从机同时通讯。一般情况下广播地址定义为 FFH,该地址可被所有从机响应。

例如,三路从机按下表所示预设其从机地址和地址掩码。则主机与从机 0 单独通讯时需发送地址(1010-0010),与从机 1 单独通讯时需发送地址(1010-0100),而与从机 2 单独通讯时需发送地址(1010-0011)。若主机希望同时与多路从机通讯,则可发送地址(1010-0000)与从机 0 和从机 1 通讯,或发送地址(1010-0011)与从机 0 和从机 2 通讯,或发送地址(1010-0101)与从机 1 和从机 2 通讯,还可发送地址(1010-0001)与从机 0、从机 1、从机 2 同时通讯。

表 11-3 从机地址

	从机 0	从机 1	从机 2
URTAR 预设值	1010 0011	1010 0100	1010 0111
URTMR 预设值	1111 1100	1111 1010	1111 1001
从机响应的地址列表 (URTMR 中某一位为 0, 则默认 URTAR 中相应位匹配)	<u>1010 0000</u>	<u>1010 0000</u>	<u>1010 0001</u>
	<u>1010 0001</u>	<u>1010 0001</u>	<u>1010 0011</u>
	1010 0010	1010 0100	<u>1010 0101</u>
	<u>1010 0011</u>	<u>1010 0101</u>	1010 0111
从机响应的广播地址列表 (URTAR 和 URTMR 逻辑“或”的值)		1111 1110	
	1111 1111	1111 1111	1111 1111

系统复位后,URTAR 和 URTMR 的初始值均为 0,即初始设定从机地址和广播地址均为 xxxx-xxxx(所有位均被忽略),从而屏蔽从机地址匹配功能,有效地去除了多机通讯的特性,UART 将对任何地址均产生应答,可兼容不支持自动地址识别的芯片,也可按上述方法实现地址自动识别的多机通讯应用。

#### 11.4.3 端口输出自动转换

当 UART 输入/输出电平取反控制位 URTLNV 为 0 时,TX 端口输出逻辑“0”时强制输出低电平,输出逻辑“1”时将自动转为通用 I/O; URTLNV 为 1 时,TX 端口输出逻辑“0”时强制输出高电平,输出逻辑“1”时将自动转为通用 I/O。单机通讯中,可将 TX 端口的通用 I/O 功能设为输出;而多机通讯中,将 TX 端口的通用 I/O 功能设为输入,则可通过外部上拉电阻被动输出高电平(或外接下拉电阻输出低电平),从而在多机同时输出时,不会因总线电平冲突而损坏芯片。



## 11.5 UART 相关寄存器

### 11.5.1 UART 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
URTCR0	URTEN	UREN	UPEN	UPMD	-	URTPRS2	URTPRS1	URTPRS0
R/W	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W
初始值	0	0	0	0	-	0	0	0

BIT[7]            URTEN – UART 使能位

- 0: 关闭 UART;
- 1: 使能 UART;

BIT[6]            UREN – UART 接收使能位

- 0: 禁止 UART 接收;
- 1: 允许 UART 接收;

BIT[5]            UPEN – UART 奇偶校验使能位

- 0: 关闭奇偶校验;
- 1: 使能奇偶校验;

BIT[4]            UPMD – UART 奇偶校验模式选择位

- 0: 偶校验模式;
- 1: 奇校验模式;

BIT[2:0]          URTPRS[2:0] – UART 时钟预分频比选择位

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

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
URTCR1	-	-	-	-	UAEN	UDATF	USTPF	TXB8
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3]            UAEN – UART 地址检测使能位

- 0: 关闭地址检测, 任意帧接收完成后均触发 UART 接收中断;
- 1: 使能地址检测, 仅接收完地址帧 (第 9 位为 1) 并与本机预设地址匹配才触发中断;

BIT[2]            UDATF – UART 数据 (含校验位) 格式选择位



- 0: 数据格式为 8 位数据;  
1: 数据格式为 9 位数据;
- BIT[1] USTPF – UART 停止位格式选择位  
0: 停止位格式为 1 位停止位;  
1: 停止位格式为 2 位停止位;
- BIT[0] TXB8 – UART 发送数据的第 9 位  
0: 发送数据的第 9 位为 0;  
1: 发送数据的第 9 位为 1;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
URTCR2	-	-	-	-	URTCHS	URTPTS	URTMOD	URLNV
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

- BIT[3] URTCHS – UART 通讯接口选择位  
0: UART 通讯接口为 TX0/RX0;  
1: UART 通讯接口为 TX1/RX1;
- BIT[2] URTPTS – 单线模式 UART 通讯接口选择位 (仅 URTMOD=1 时有效)  
0: 单线模式 UART 通讯接口为 RX0 或 RX1 (由 URTCHS 确定);  
1: 单线模式 UART 通讯接口为 TX0 或 TX1 (由 URTCHS 确定);
- BIT[1] URTMOD – UART 工作模式选择位  
0: 双线模式, TX 和 RX 分用不同端口;  
1: 单线模式, TX 和 RX 共用同一端口;
- BIT[0] URLNV – UART 输入/输出电平取反控制位  
0: 输入/输出电平不取反, 低电平为逻辑“0”、高电平为逻辑“1”;  
1: 输入/输出电平取反, 高电平为逻辑“0”、低电平为逻辑“1”;

### 11.5.2 UART 状态寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
URTSR	TXCL	-	-	-	FERR	PERR	RXOV	RXB8
R/W	R/W	-	-	-	R/W	R/W	R/W	R
初始值	0	-	-	-	0	0	0	0

- BIT[7] TXCL – UART 发送冲突标志位  
0: 发送无冲突;  
1: 发送有冲突, 硬件自动置 1, 需软件清 0;
- BIT[3] FERR – UART 帧出错标志位  
0: 无帧出错;  
1: 发生帧出错, 硬件自动置 1, 需软件清 0;
- BIT[2] PERR – UART 奇偶校验错误标志位



- 0: 无奇偶校验错误;  
1: 发生奇偶校验错误, 硬件自动置 1, 需软件清 0;
- BIT[1] RXOV – UART 接收溢出标志位  
0: 接收无溢出;  
1: 接收有溢出, 硬件自动置 1, 需软件清 0;
- BIT[0] RXB8 – UART 接收数据的第 9 位 (数据或校验码)  
0: 接收数据的第 9 位为 0;  
1: 接收数据的第 9 位为 1;

### 11.5.3 UART 波特率寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
URTBR	URTB7	URTB6	URTB5	URTB4	URTB3	URTB2	URTB1	URTB0
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] URTB[7:0] – UART 波特率寄存器, 用于设置 UART 通讯的波特率 (需  $URTBR \geq 16$ )

### 11.5.4 UART 数据寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
URTDR	URTD7	URTD6	URTD5	URTD4	URTD3	URTD2	URTD1	URTD0
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] URTD[7:0] – UART 数据寄存器, 写 URTDR 为写发送缓冲器, 读 URTDR 为读接收缓冲器

注: 对 URTDR 读和写操作的目的寄存器不是同一寄存器, 仅可使用 MOVAR 或 MOVRA 指令进行读或写操作。

### 11.5.5 UART 地址寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
URTAR	URTA7	URTA6	URTA5	URTA4	URTA3	URTA2	URTA1	URTA0
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] URTA[7:0] – UART 地址寄存器, 用于设置 UART 通讯的从机地址

### 11.5.6 UART 地址掩码寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
URTMR	URTM7	URTM6	URTM5	URTM4	URTM3	URTM2	URTM1	URTM0
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] URTMn – 地址位 URTAn 比对控制位 (n=7-0)





- 0: 地址位 URTAn 不参与地址比对, 默认该位地址匹配;
- 1: 地址位 URTAn 参与地址比对;

Shomcu.com



## 12 EEPROM 存储器

### 12.1 EEPROM 概述

芯片内置 128×16 位的 EEPROM 型数据存储器，支持用户程序在带电运行中实时地读写数据。对 EEPROM 中数据的读写操作需通过控制寄存器 EECR、保护寄存器 EEPR、地址寄存器 EEAR 和数据寄存器 EEDRH/EEDRL 进行。

写操作控制位 EEWRITE 置 1 将启动 EEPROM 写操作，EEDRH/EEDRL 中的 16 位数据将被写入 EEAR 指向的 EEPROM 地址中，完成后 EEWRITE 自动清 0。为防止误触发 EEPROM 写操作，需先对 EEPR 写 5AH 再立即写 A5H，EEWRITE 才能置 1，中间不能插入其他操作（包括 NOP 操作），否则 EEWRITE 将无法置 1。在写 EEPR 前需先屏蔽中断，否则可能会因系统响应中断而导致 EEWRITE 无法置 1。

读操作控制位 EEREAD 置 1 将启动 EEPROM 读操作，EEAR 所指 EEPROM 地址中的 16 位数据将被读出并缓存于 EEDRH/EEDRL，完成后 EEREAD 自动清 0。读操作需 2 个指令周期，且在 EEWRITE 为 1 时无效。

注：

EEPROM 读/写操作时，CPU 将暂停工作；

EEPROM 操作的时钟为系统高频时钟  $F_{HOSC}$ ，因此在操作时需确保系统高频时钟源工作；

对 EECR、EEPR、EEAR 和 EEDR 的写操作，仅能通过 MOVRA 指令进行，其他指令的执行结果不确定；

### 12.2 EEPROM 相关寄存器

#### 12.2.1 EEPROM 控制寄存器

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

BIT[1] EEREAD – EEPROM 读操作控制位

0：操作未开始或已完成，写 1 开始 EEPROM 读操作；

1：EEPROM 读操作中，完成后自动清 0；

BIT[0] EEWRITE – EEPROM 写操作控制位

0：操作未开始或已完成，写 1 开始 EEPROM 写操作；

1：EEPROM 写操作中，完成后自动清 0；

#### 12.2.2 EEPROM 保护寄存器

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



EEPR	EEP7	EEP6	EEP5	EEP4	EEP3	EEP2	EEP1	EEP0
R/W	W	W	W	W	W	W	W	W
初始值	0	0	0	0	0	0	0	0

BIT[7:0]      EEP[7:0] – EEPROM 写操作保护控制位，需先写 5AH 再立即写 A5H，EEWRITE 才能置 1

### 12.2.3 EEPROM 地址寄存器

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

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

### 12.2.4 EEPROM 数据寄存器

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

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

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

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

## 12.3 EEPROM 操作示例

例如，先将数据 55H 写入 EEPROM 存储器 10H 低字节中，再读取 EEPROM 存储器 11H 地址低字节中内容：

```

MOVAI    10H
MOVRA    EEAR                ; 将 10H 写入 EEAR
MOVAI    55H
MOVRA    EEDRL              ; 将 55H 写入 EEDRL
BCLR     GIE                ; 屏蔽中断
MOVAI    5AH
MOVRA    EEPR                ; 使能 EE 操作，第 1 步：EEPR 写 5AH
MOVAI    A5H
MOVRA    EEPR                ; 使能 EE 操作，第 2 步：EEPR 写 A5H
MOVAI    01H

```



MOVRA	EECR	: 启动 EE 写操作, 将数据 55H 写入 EEPROM 地址 10H 中
NOP		: 为防止时序错误, CPU 必须先执行 2-4 个 NOP 指令
NOP		
BSET	GIE	: 允许中断
JBCLR	EECR, 0	: 检查 EE 写操作是否完成
GOTO	\$-1	
MOVAI	11H	
MOVRA	EEAR	: 将 11H 写入 EEAR
MOVAI	02H	
MOVRA	EECR	: 启动 EE 读操作, 读取 EEPROM 地址 11H 中内容
NOP		: 为防止时序错误, CPU 必须先执行 2-4 个 NOP 指令
NOP		
MOVAR	EEDRL	: 从 EEDRL 中读取数据



## 13 FLASH 烧录编程

### 13.1 FLASH 在板编程

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

芯片也支持对 EEPROM 型数据存储器的在板编程。

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

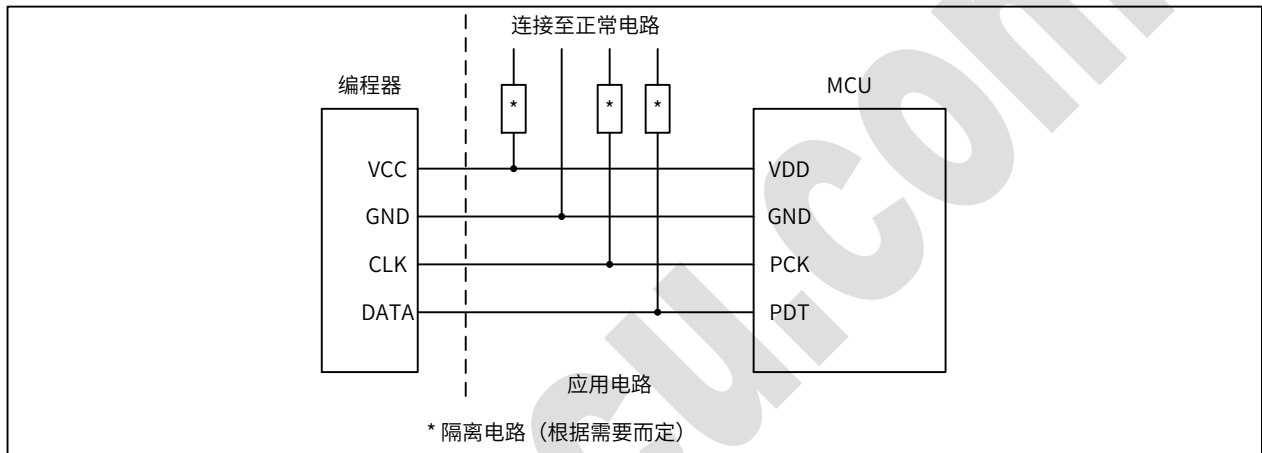


图 13-1 在板编程连接示意图

芯片也支持在板带电烧录编程，即可在系统电路板不掉电（芯片已正常工作）的状态下对存储器编程。当配置字 DBGPIN0/DBGPIN1 允许寄存器位控制后，再通过寄存器位 DBG0EN/DBG1EN 将对应的端口设为编程端口，则端口的通用功能被屏蔽，芯片可通过该组端口进入编程/仿真模式。

注：

- 1、不支持空芯片的在板带电烧录编程；
- 2、在板带电烧录，编程器 VCC 不接入电路板，芯片由系统电路板通过 VDD 引脚供电；
- 3、在板带电烧录，编程器 GND/CLK/DATA 接入电路板前，需注意连接端口之间电压特性是否匹配，尤其需注意编程器与系统电路板的共 GND 是否会发生浮地与市电地短路的问题；
- 4、在板带电烧录，编程器在烧录完成后，会将芯片重新复位；

#### 13.1.1 DEBUG 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
DBGCR	-	-	-	-	-	-	DBG1EN	DBG0EN
R/W	-	-	-	-	-	-	R/W	R/W
初始值	-	-	-	-	-	-	0	0

BIT[1]            DBG1EN – PCK1/PDT1 端口编程功能使能位

0: 端口用作通用端口，端口的编程功能被屏蔽；



1: 使能端口的编程功能, 端口用作 PCK1/PDT1, 端口的通用功能被屏蔽;

BIT[0]      DBG0EN – PCK0/PDT0 端口编程功能使能位

0: 端口用作通用端口, 端口的编程功能被屏蔽;

1: 使能端口的编程功能, 端口用作 PCK0/PDT0, 端口的通用功能被屏蔽;

注: 仅在配置字将端口配置为“可通过寄存器位将端口切换为编程端口或通用端口”时, 才可通过对应的寄存器控制位使能端口的编程功能。

### 13.1.2 DEBUG 保护寄存器

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

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

为防止误触发对 DBGCR 的写操作, 需先对寄存器 DBGPR 写 3CH 再立即写 C3H, DBG1EN 或 DBG0EN 才能置 1, 中间不能插入其他操作 (包括 NOP 操作), 否则对 DBGCR 的写操作将无效。在写 DBGPR 前需先屏蔽中断, 否则可能会因系统响应中断而导致 DBGCR 写操作无效。

例如, 在配置字 DBGPIN0 将 PCK0 (P13) /PDT0 (P12) 配置为“可通过寄存器位将端口切换为编程端口或通用端口”后, 芯片带电工作中若 DBG0EN=0, 则 P13/P12 仍用作数字或模拟等通用功能的输入/输出端口, 编程器无法通过这两个引脚进行带电烧录编程, 用户程序可按如下例程将 DBG0EN 置 1, 以使编程器可通过该组编程接口进行带电烧录编程:

```

BCLR      GIE      ; 屏蔽中断
CLRWDT    ; 清 WDT
MOVAI     3CH
MOVRA     DBGPR    ; 使能 DBGCR 写操作, 第 1 步: DBGPR 写 3CH
MOVAI     C3H
MOVRA     DBGPR    ; 使能 DBGCR 写操作, 第 2 步: DBGPR 写 C3H
MOVAI     01H
MOVRA     DBGCR    ; DBG0EN 置 1, 端口用作编程接口
NOP
CLRWDT    ;
GOTO     $-2      ; 死循环, 等待编程器连接芯片进行带电烧录编程

```



## 14 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、OP2 比较中断和键盘中断等。可通过中断总使能位 GIE 屏蔽所有中断。

CPU 响应中断的过程如下：

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

*注：应用外部中断功能或键盘中断功能，需将相应端口设为输入状态。*

### 14.1 外部中断

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

### 14.2 定时器中断

定时器 Tn (n=0-2) 在计数溢出时将触发定时器中断，中断标志 TnIF (n=0-2) 将被置 1，若 GIE 为 1 且相应的定时器中断使能位 TnIE (n=0-2) 为 1，则产生定时器中断。

### 14.3 T1 捕捉中断

定时器 T1 的输入捕捉模式，捕捉完成时将触发 T1 捕捉中断，中断标志 T1CIF 将被置 1，若 GIE 为 1 且 T1 捕捉中断使能位 T1CIE 为 1，则产生 T1 捕捉中断。



## 14.4 键盘中断

芯片具有 12 路键盘中断源，均可单独使能或关闭端口的键盘中断功能。任意一路使能键盘中断功能的端口，其输入电平发生变化时均将触发键盘中断，中断标志 KBIF 将被置 1，若 GIE 为 1 且键盘中断使能位 KBIE 为 1，则产生键盘中断。

### 14.4.1 键盘中断控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
P1KBCR	-	-	P15KE	P14KE	P13KE	P12KE	P11KE	P10KE
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] P1nKE – P1n 端口键盘中断功能使能位 (n=5-0)

- 0: 关闭端口的键盘中断功能;
- 1: 使能端口的键盘中断功能;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
P2KBCR	-	-	P25KE	P24KE	P23KE	P22KE	P21KE	P20KE
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] P2nKE – P2n 端口键盘中断功能使能位 (n=5-0)

- 0: 关闭端口的键盘中断功能;
- 1: 使能端口的键盘中断功能;

## 14.5 ADC 中断

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

## 14.6 OP2 比较中断

运算放大器 OP2 工作于比较模式时，其输出信号 OP2\_OUT 可选择上升沿或下降沿等方式触发 OP2 比较中断，中断标志 OP2IF 将被置 1，若 GIE 为 1 且 OP2 比较中断使能位 OP2IE 为 1，则产生 OP2 比较中断。





## 14.7 LVD 中断

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

## 14.8 UART 中断

UART 发送过程中，每一帧的停止位发送完成将触发 UART 发送完成中断，中断标志 TXIF 将被置 1；而发送缓冲器变为空时，将触发 UART 发送缓冲器空中断，中断标志 TXBIF 将被置 1。UART 接收过程中，每一帧的第 1 个停止位接收完成时将触发 UART 接收完成中断，中断标志 RXIF 将被置 1。

若 GIE 为 1 且对应的 UART 中断使能位为 1，则产生对应的 UART 中断。

## 14.9 中断相关寄存器

### 14.9.1 中断使能寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTE0	LVDIE	ADIE	T2IE	KBIE	INT1IE	INT0IE	T1IE	TOIE
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] LVDIE – LVD 中断使能位

0: 屏蔽 LVD 中断;

1: 使能 LVD 中断;

BIT[6] ADIE – ADC 中断使能位

0: 屏蔽 ADC 中断;

1: 使能 ADC 中断;

BIT[5] 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[4] KBIE – 键盘中断使能位

0: 屏蔽键盘中断;

1: 使能键盘中断;

BIT[3] INT1IE – INT1 中断使能位



- 0: 屏蔽 INT1 中断;
- 1: 使能 INT1 中断;
- BIT[2]      INTOIE – INTO 中断使能位
  - 0: 屏蔽 INTO 中断;
  - 1: 使能 INTO 中断;
- BIT[1]      T1IE – 定时器 T1 中断使能位
  - 0: 屏蔽定时器 T1 中断;
  - 1: 使能定时器 T1 中断;
- BIT[0]      T0IE – 定时器 T0 中断使能位
  - 0: 屏蔽定时器 T0 中断;
  - 1: 使能定时器 T0 中断;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTE1	-	-	TXBIE	TXIE	RXIE	T1CIE	-	OP2IE
R/W	-	-	R/W	R/W	R/W	R/W	-	R/W
初始值	-	-	0	0	0	0	-	0

- BIT[5]      TXBIE – UART 发送缓冲器空中断使能位
  - 0: 屏蔽 UART 发送缓冲器空中断;
  - 1: 使能 UART 发送缓冲器空中断;
- BIT[4]      TXIE – UART 发送完成中断使能位
  - 0: 屏蔽 UART 发送完成中断;
  - 1: 使能 UART 发送完成中断;
- BIT[3]      RXIE – UART 接收完成中断使能位
  - 0: 屏蔽 UART 接收完成中断;
  - 1: 使能 UART 接收完成中断;
- BIT[2]      T1CIE – T1 捕捉中断使能位
  - 0: 屏蔽 T1 捕捉中断;
  - 1: 使能 T1 捕捉中断;
- BIT[0]      OP2IE – OP2 比较中断使能位
  - 0: 屏蔽 OP2 比较中断;
  - 1: 使能 OP2 比较中断;

### 14.9.2 中断标志寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTF0	LVDIF	ADIF	T2IF	KBIF	INT1IF	INTOIF	T1IF	T0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

- BIT[7]      LVDIF – LVD 中断标志位
  - 0: 未触发 LVD 中断;



- 1: 已触发 LVD 中断, 需软件清 0;
- BIT[6] ADIF – ADC 中断标志位
- 0: 未触发 ADC 中断;
- 1: 已触发 ADC 中断, 需软件清 0;
- BIT[5] T2IF – 定时器 T2 中断标志位
- 0: 未触发定时器 T2 中断;
- 1: 已触发定时器 T2 中断, 需软件清 0;
- BIT[4] KBIF – 键盘中断标志位
- 0: 未触发键盘中断;
- 1: 已触发键盘中断, 需软件清 0;
- BIT[3] INT1IF – INT1 中断标志位
- 0: 未触发 INT1 中断;
- 1: 已触发 INT1 中断, 需软件清 0;
- BIT[2] INTOIF – INTO 中断标志位
- 0: 未触发 INTO 中断;
- 1: 已触发 INTO 中断, 需软件清 0;
- BIT[1] T1IF – 定时器 T1 中断标志位
- 0: 未触发定时器 T1 中断;
- 1: 已触发定时器 T1 中断, 需软件清 0;
- BIT[0] T0IF – 定时器 T0 中断标志位
- 0: 未触发定时器 T0 中断;
- 1: 已触发定时器 T0 中断, 需软件清 0;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTF1	-	-	TXBIF	TXIF	RXIF	T1CIF	-	OP2IF
R/W	-	-	R/W	R/W	R/W	R/W	-	R/W
初始值	-	-	0	0	0	0	-	0

- BIT[5] TXBIF – UART 发送缓冲器空中断标志位
- 0: 未触发 UART 发送缓冲器空中断;
- 1: 已触发 UART 发送缓冲器空中断, 需软件清 0;
- BIT[4] TXIF – UART 发送完成中断标志位
- 0: 未触发 UART 发送完成中断;
- 1: 已触发 UART 发送完成中断, 需软件清 0;
- BIT[3] RXIF – UART 接收完成中断标志位
- 0: 未触发 UART 接收完成中断;
- 1: 已触发 UART 接收完成中断, 需软件清 0;
- BIT[2] T1CIF – T1 捕捉中断标志位
- 0: 未触发 T1 捕捉中断;
- 1: 已触发 T1 捕捉中断, 需软件清 0;



BIT[0]      OP2IF – OP2 比较中断标志位  
0: 未触发 OP2 比较中断;  
1: 已触发 OP2 比较中断, 需软件清 0;

Shomcu.com



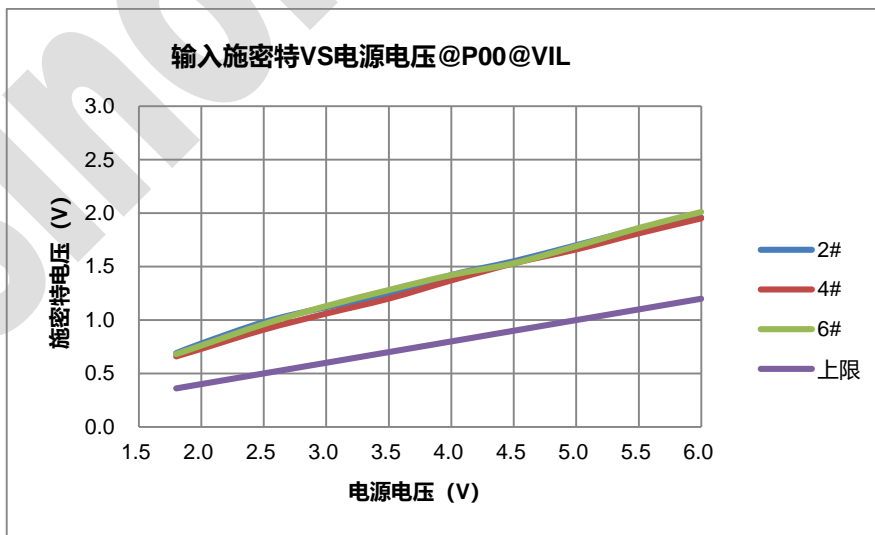
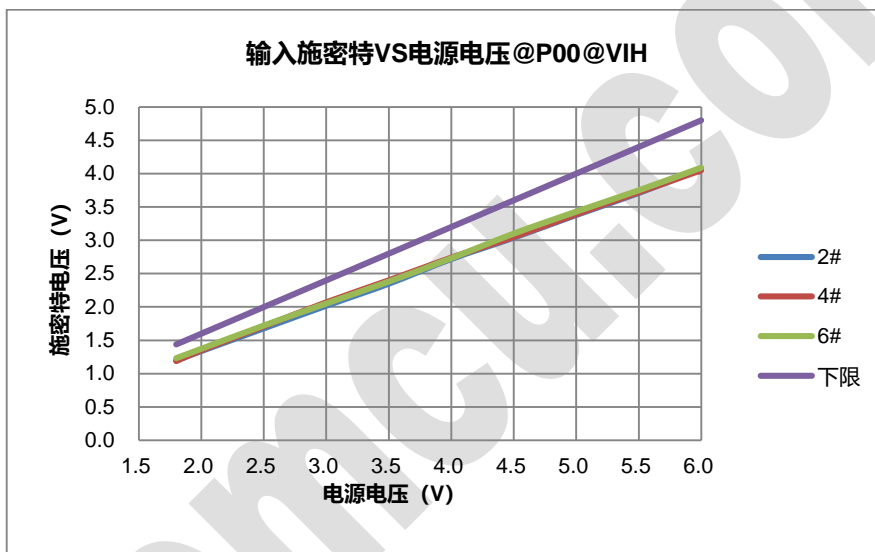
## 15 特性曲线

注:

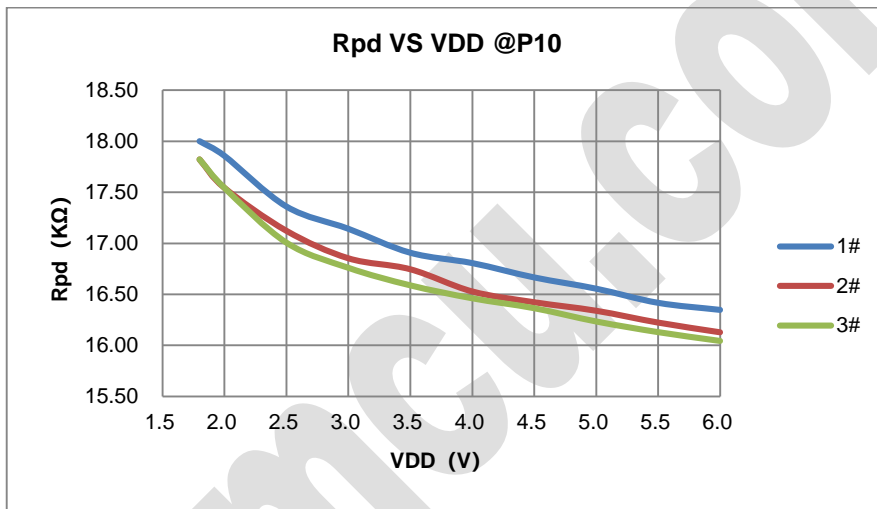
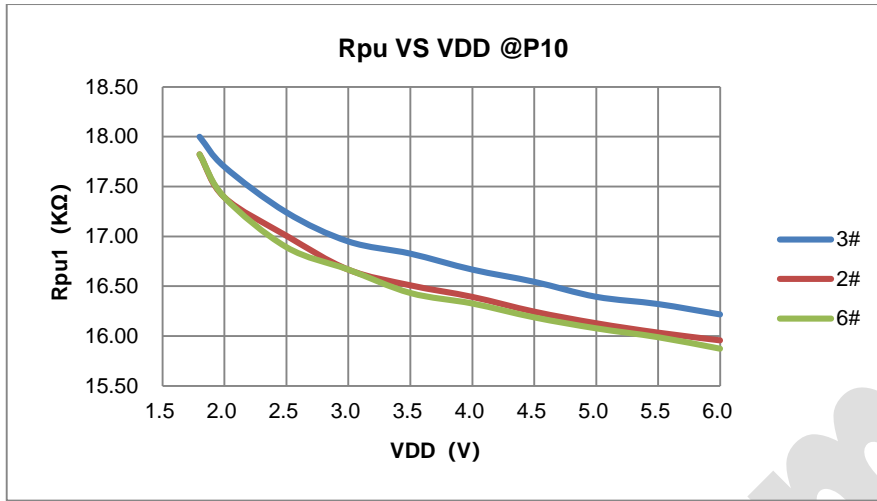
- 1、特性曲线图中数据均源自抽样实测，仅作为应用参考，部分数据因生产工艺偏差，可能与实际芯片不符；为保证芯片能正常工作，请确保其工作条件符合电气特性参数说明；
- 2、图文中若无特别说明，则电压特性曲线的温度条件为  $T=25^{\circ}\text{C}$ ，温度特性曲线的电压条件为  $VDD=5\text{V}$ ；

### 15.1 I/O 特性

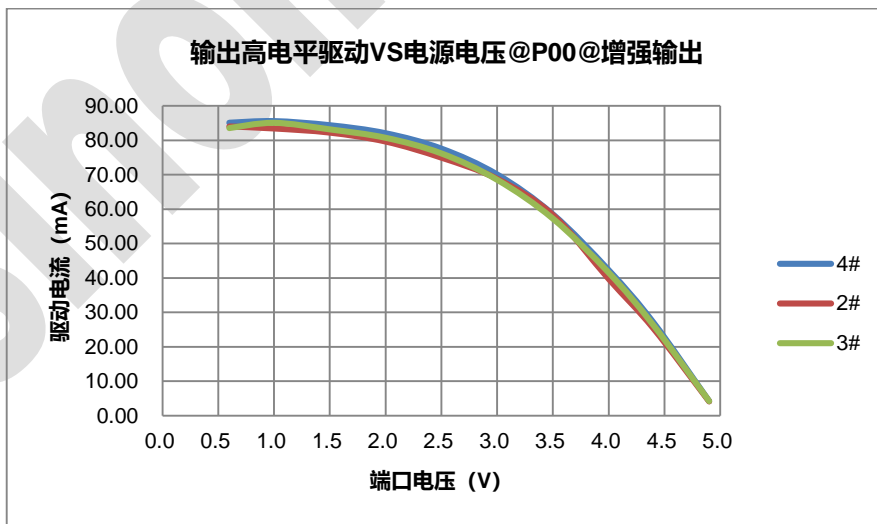
#### 15.1.1 输入 SMT 阈值电压 VS 电源电压

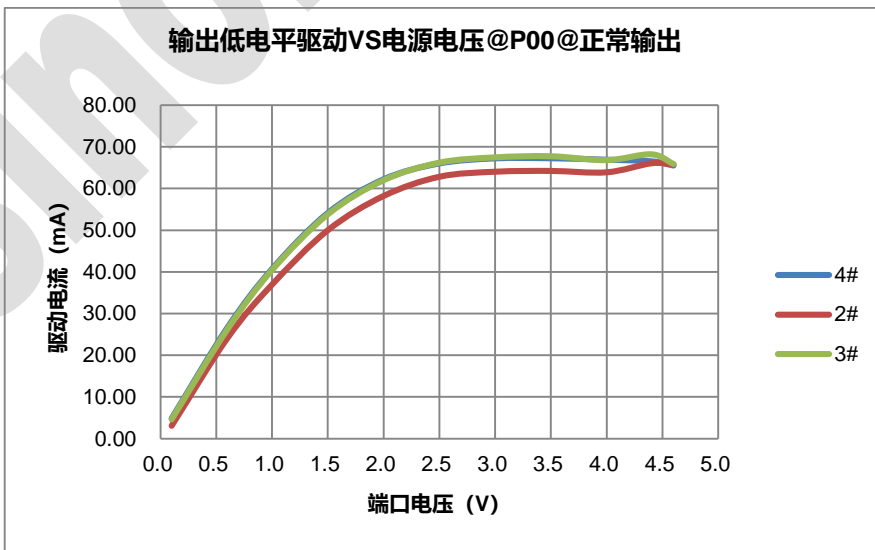
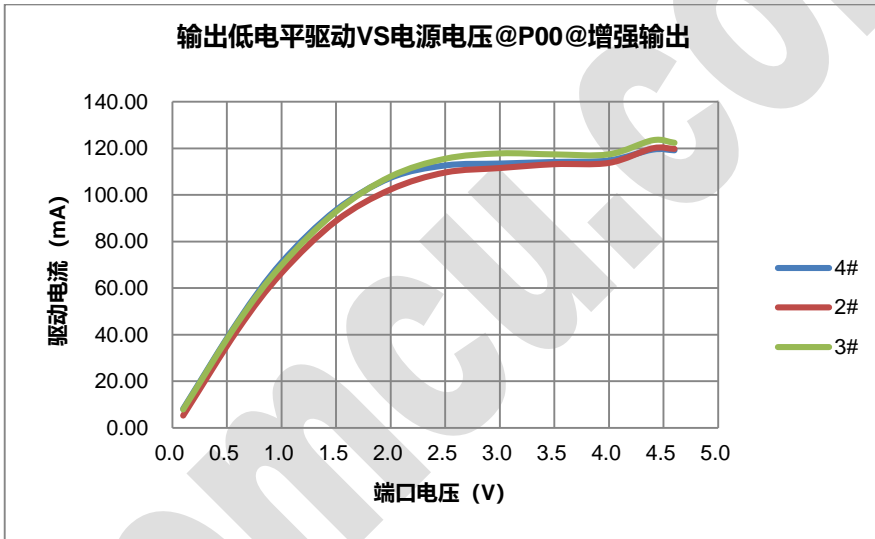
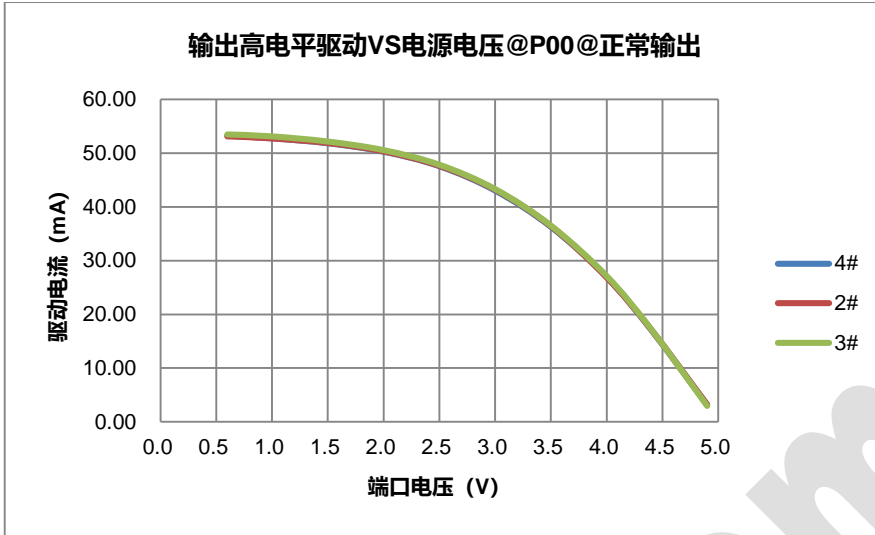


#### 15.1.2 上/下拉电阻值 VS 电源电压



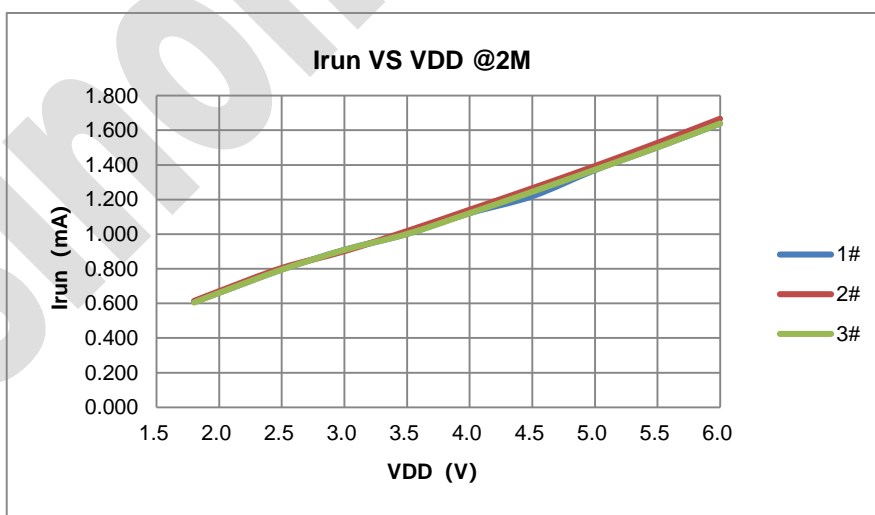
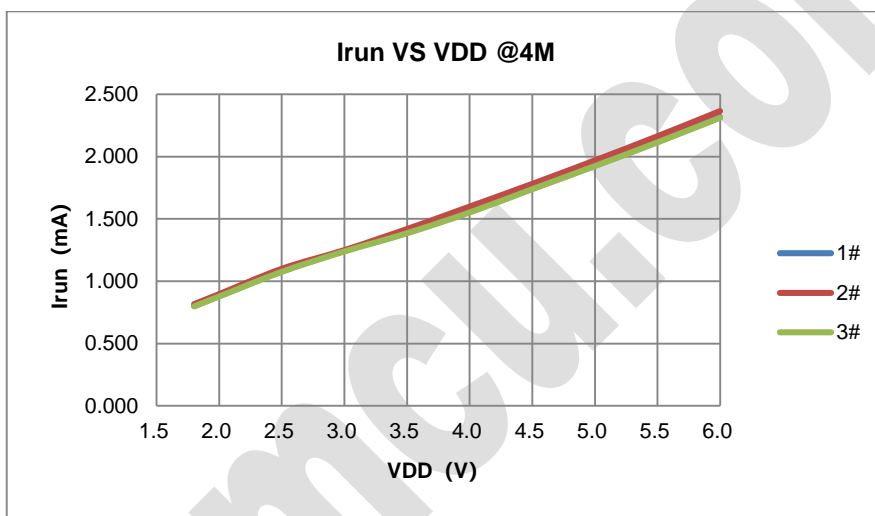
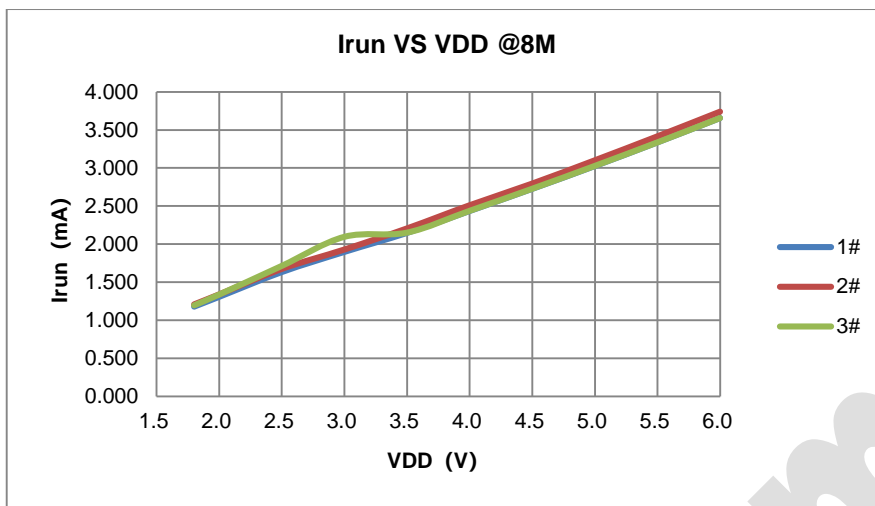
15.1.3 I/O 输出电流 VS 端口电压 (VDD=5V)



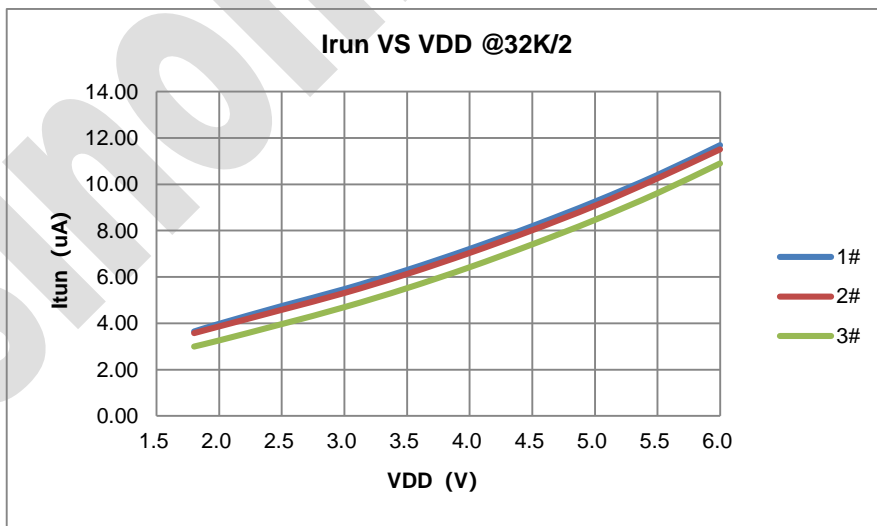
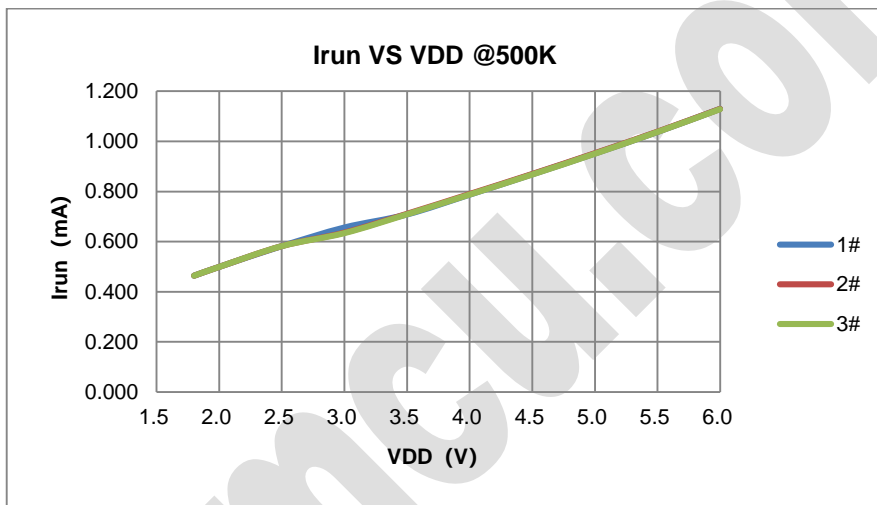


## 15.2 功耗特性

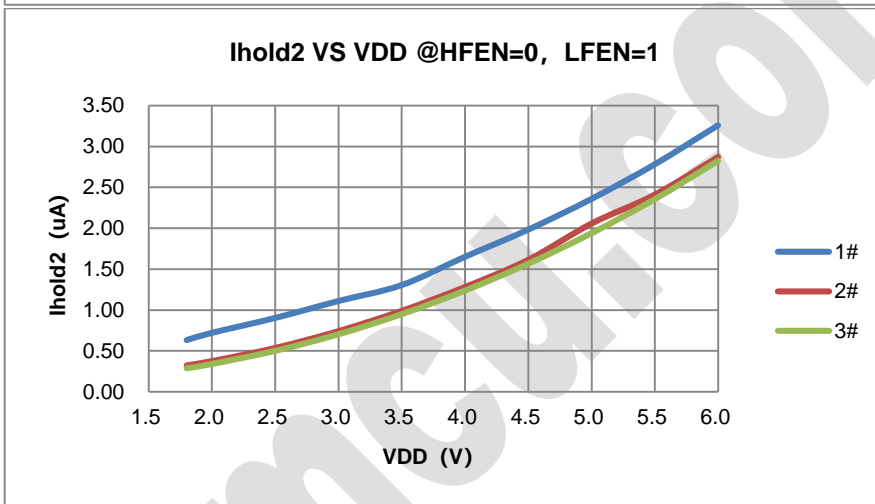
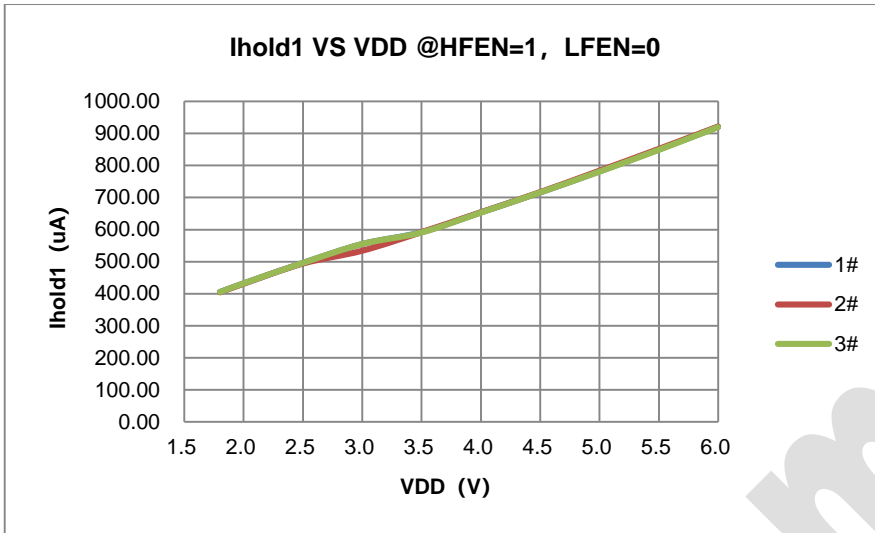
### 15.2.1 运行模式 功耗 VS 电源电压



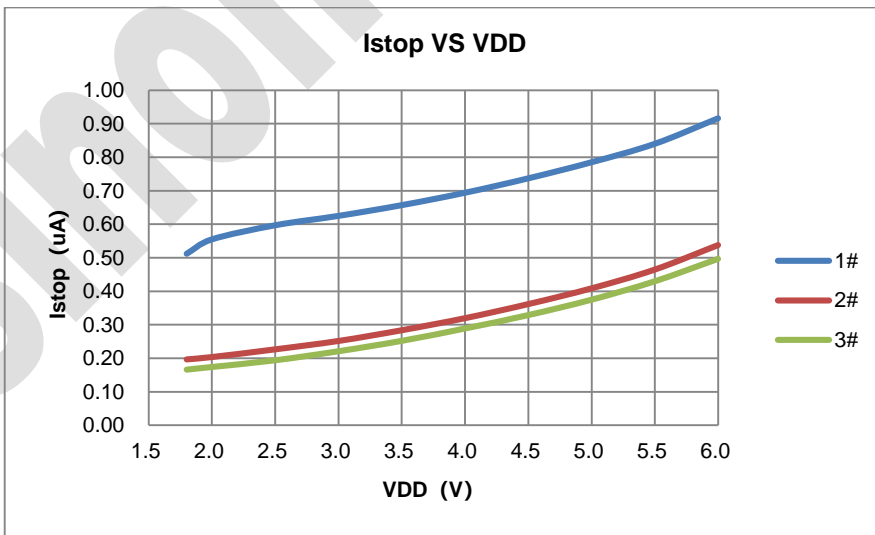


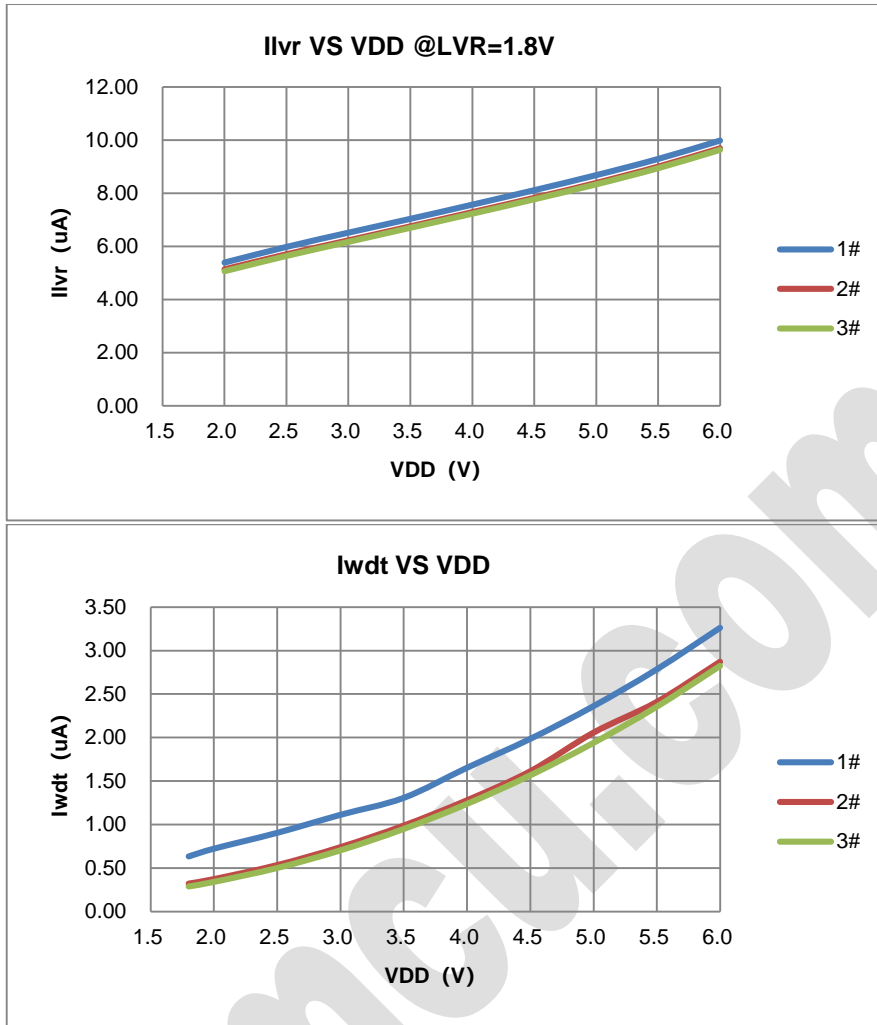


15.2.2 HOLD 模式 功耗 VS 电源电压



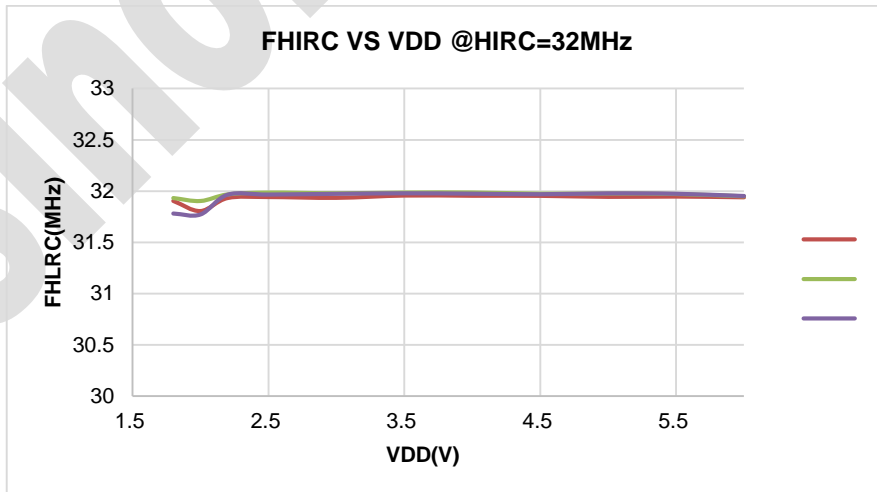
### 15.2.3 休眠模式 功耗 VS 电源电压

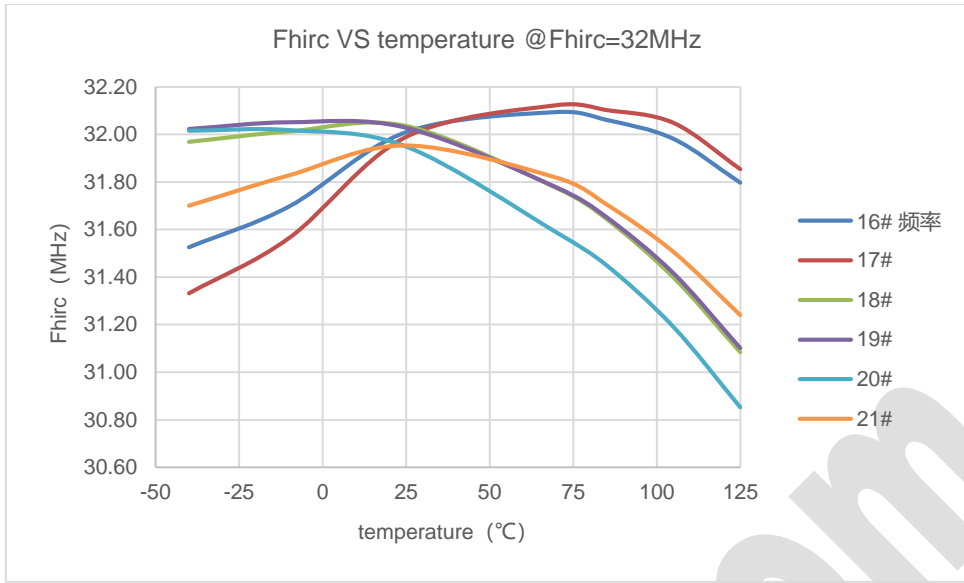




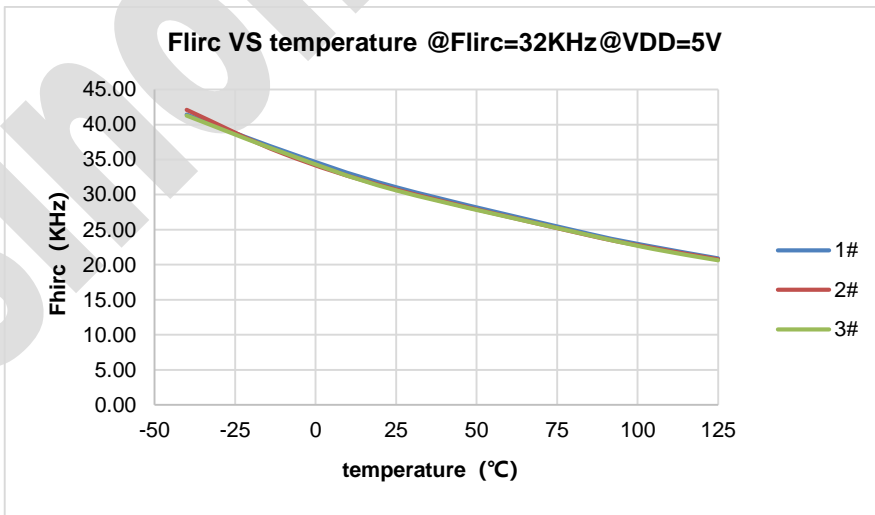
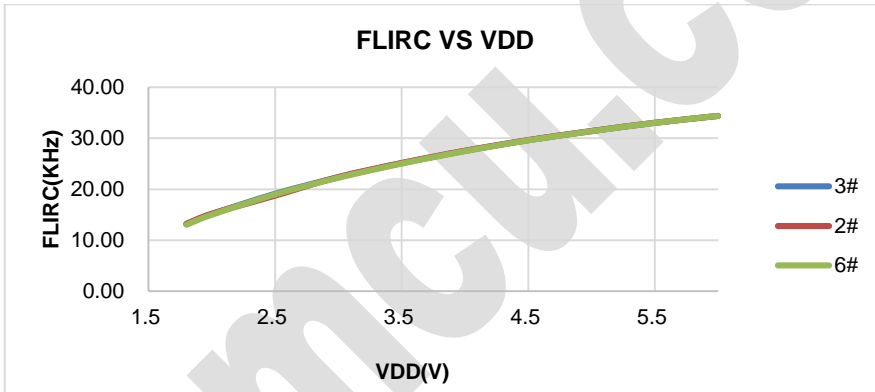
### 15.3 模拟电路特性

#### 15.3.1 HIRC 频率 VS 电源电压/温度

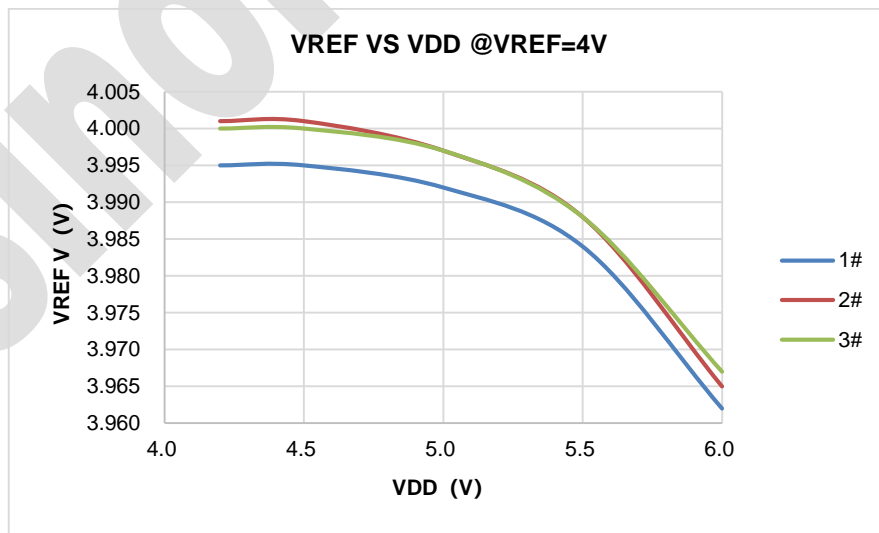
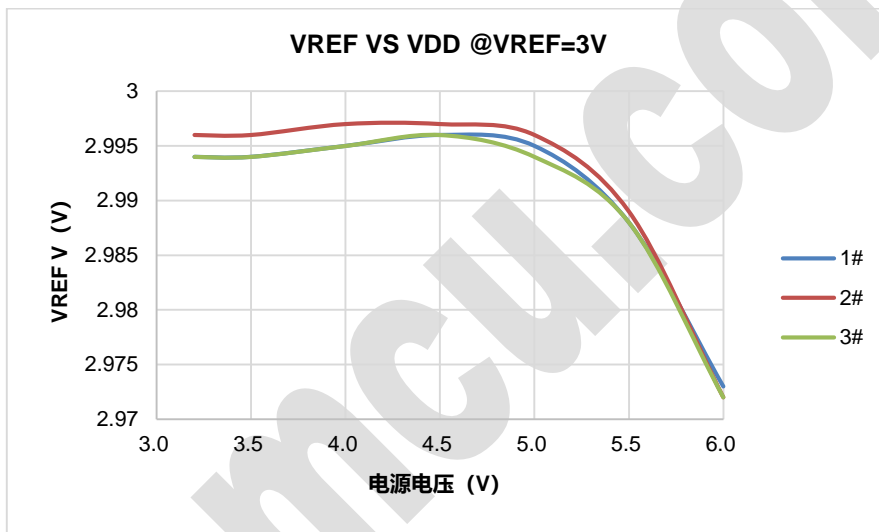
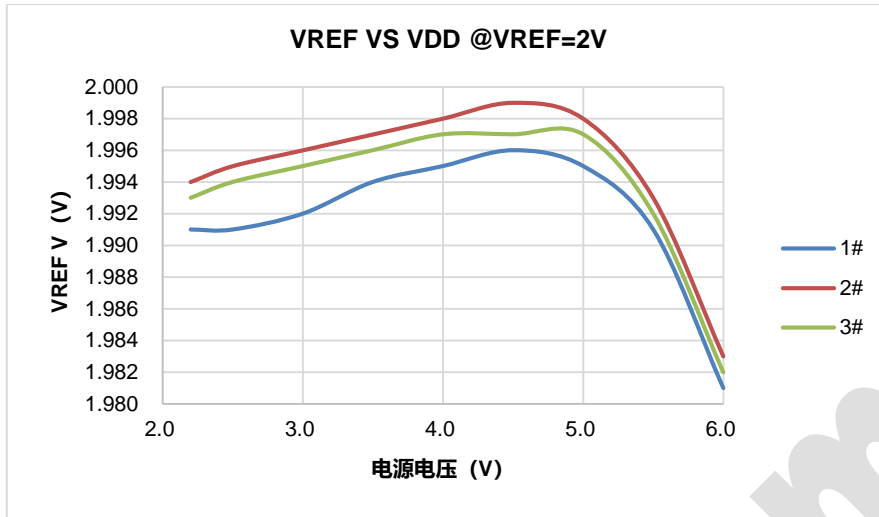


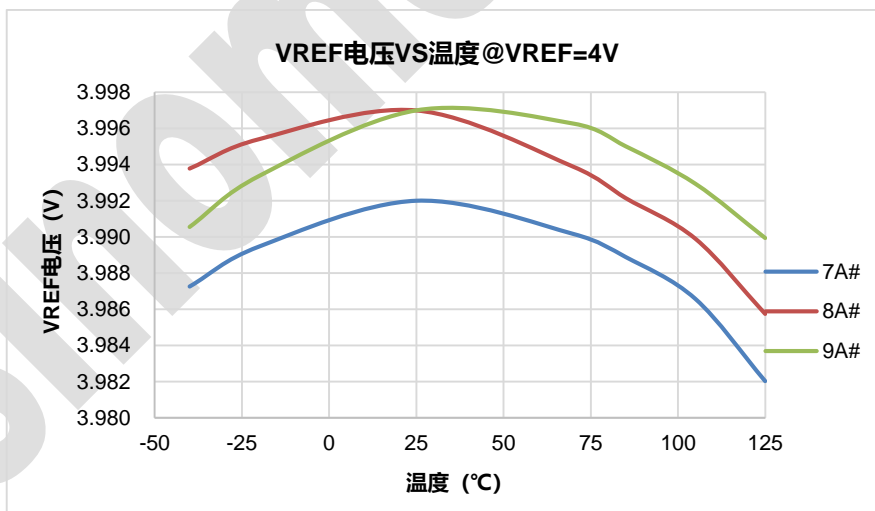
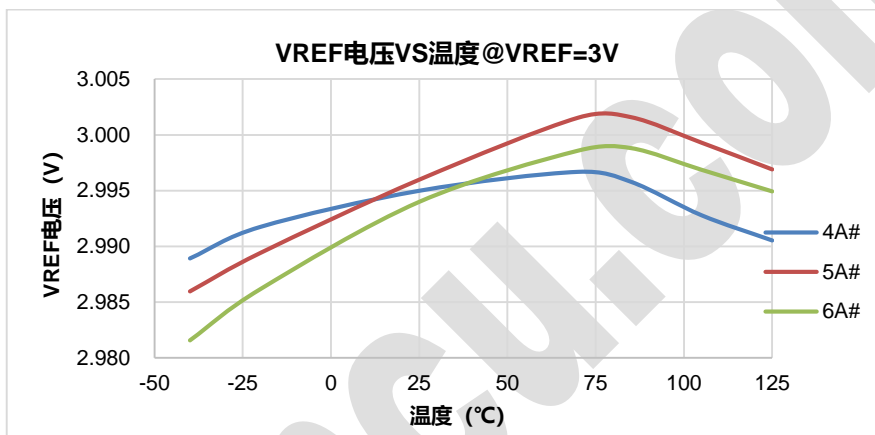
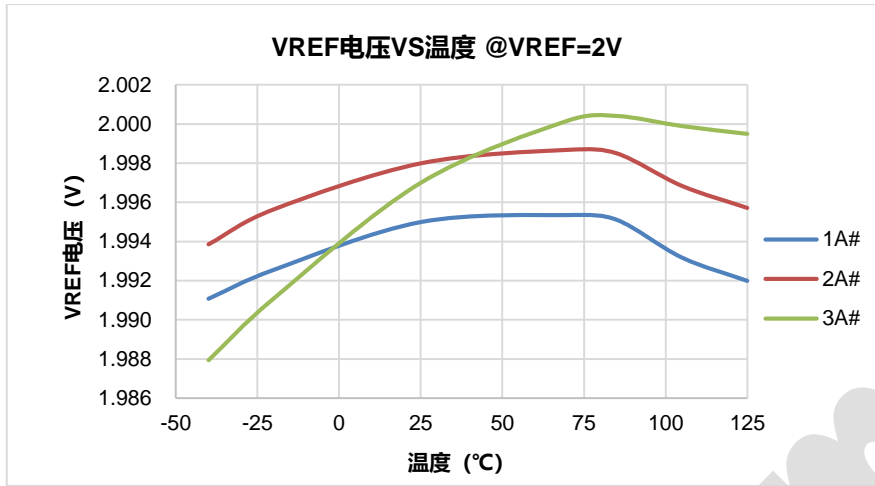


15.3.2 LIRC 频率 VS 电源电压/温度

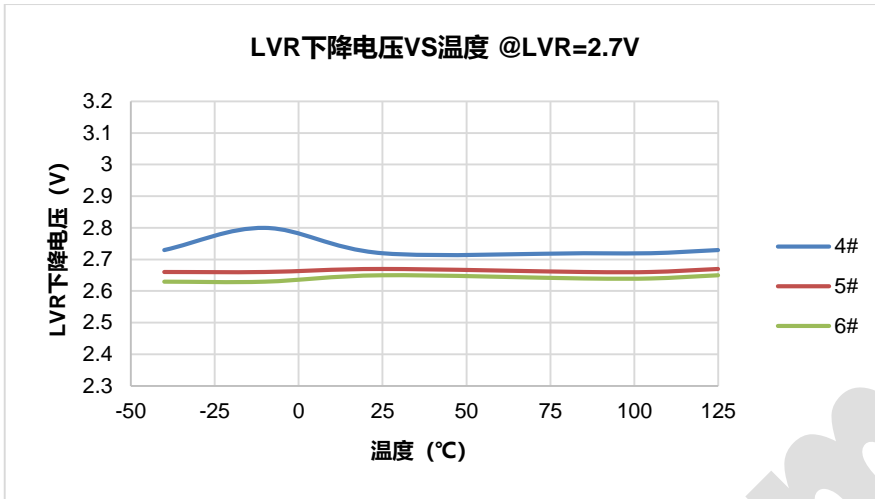


15.3.3 VIR 电压 VS 电源电压/温度

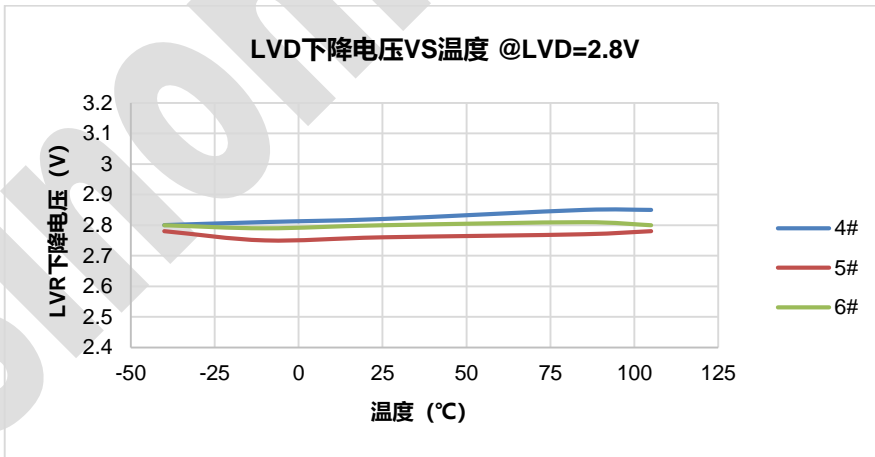
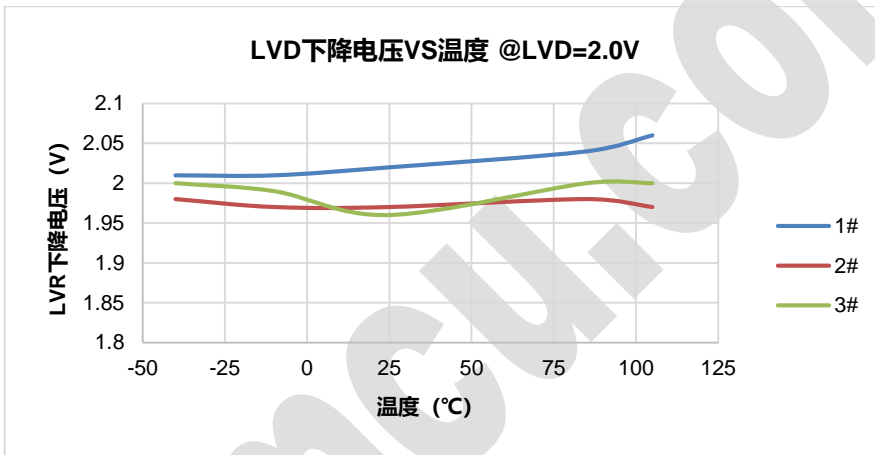


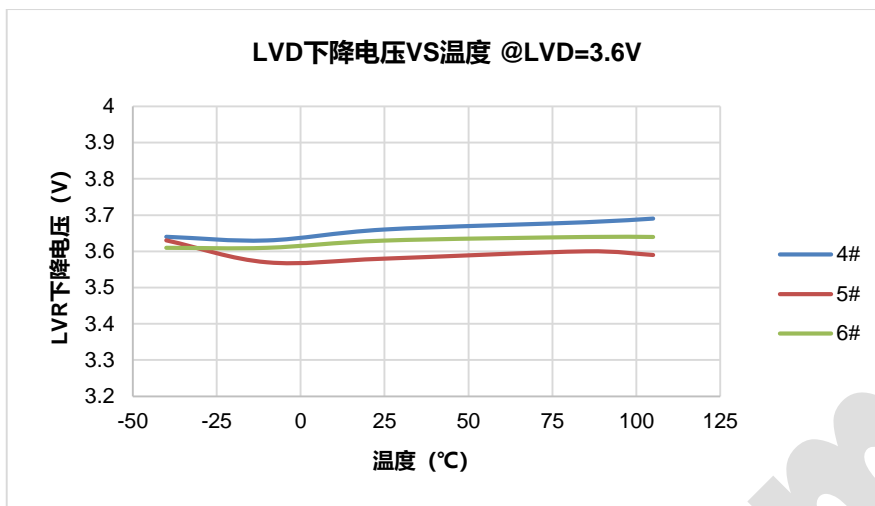


15.3.4 LVR 阈值电压 VS 温度



### 15.3.5 LVD 阈值电压 VS 温度





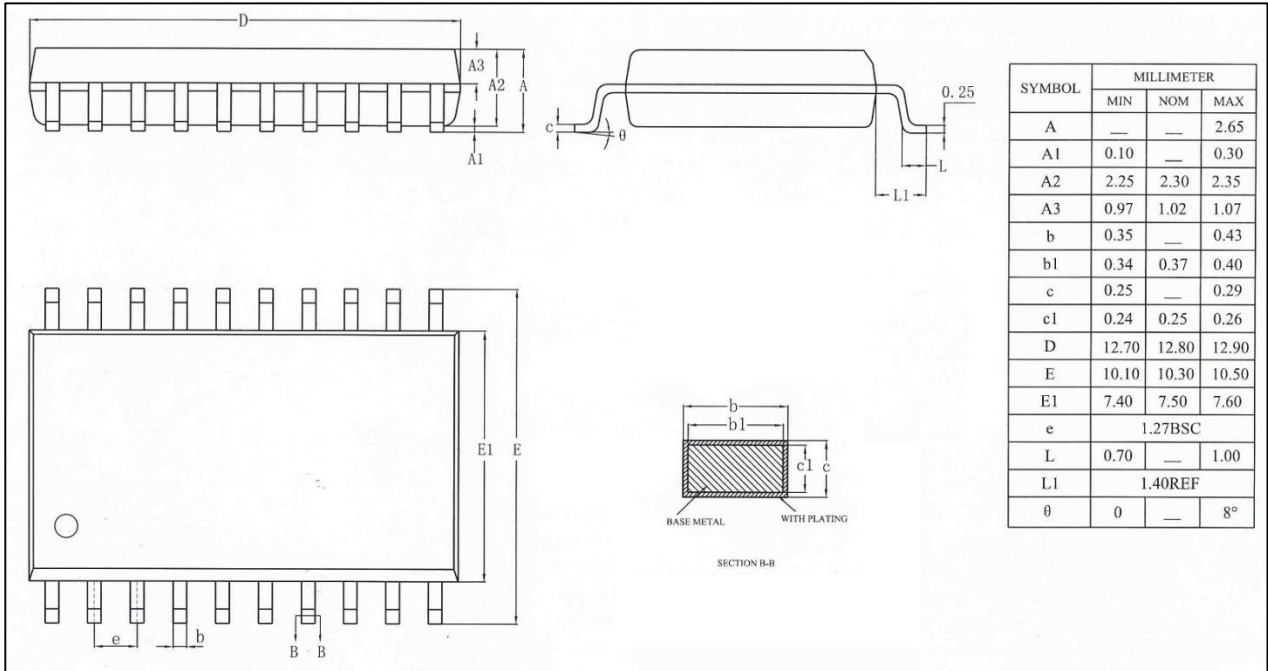
shomcu.com



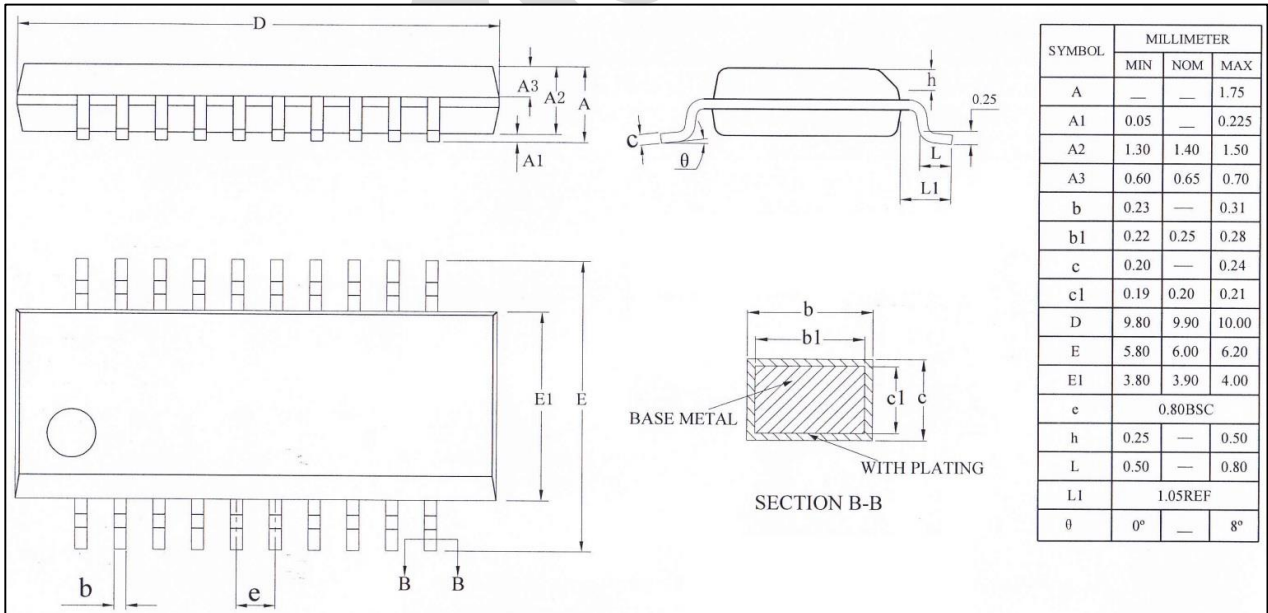


## 16 封装尺寸

### 16.1 SOP20

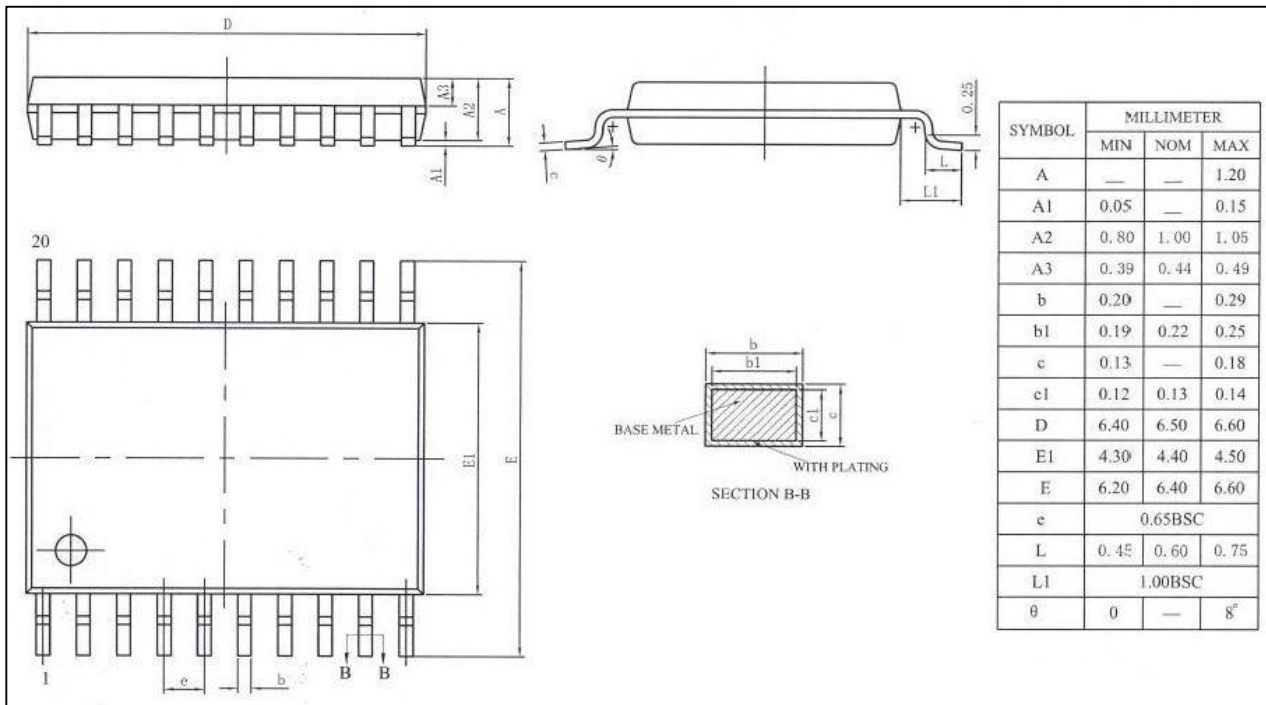


### 16.2 SSOP20 (e=0.8)

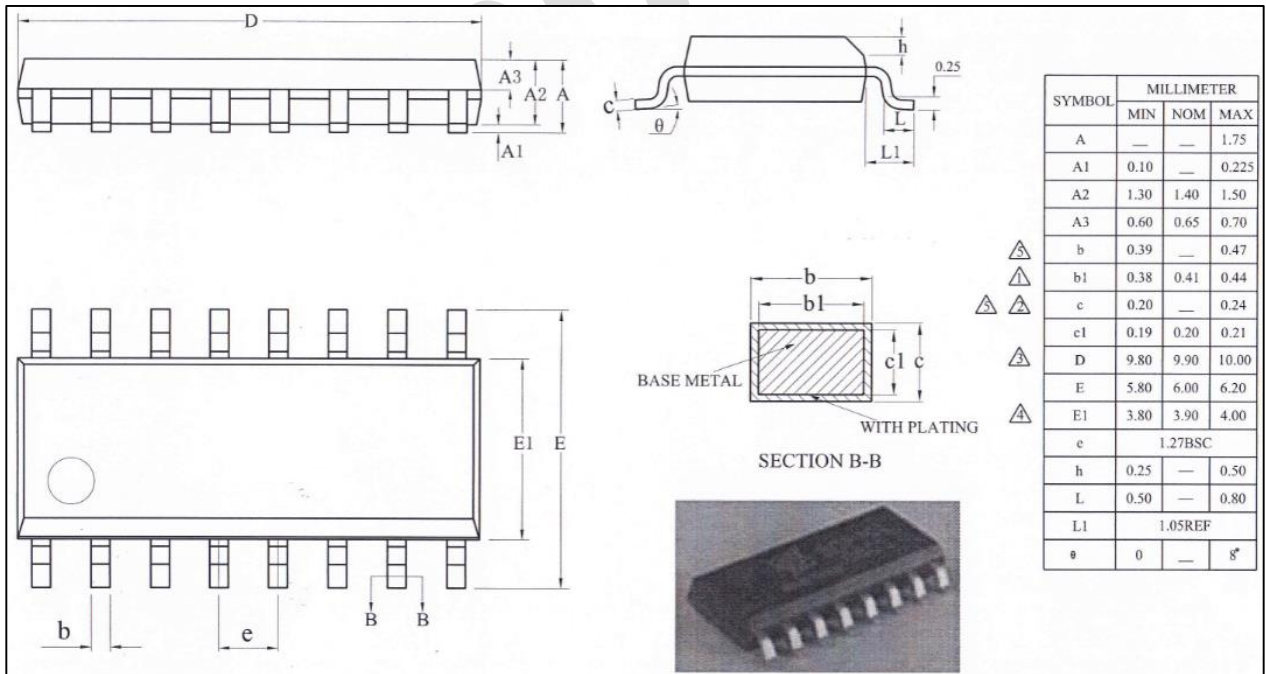




### 16.3 TSSOP20

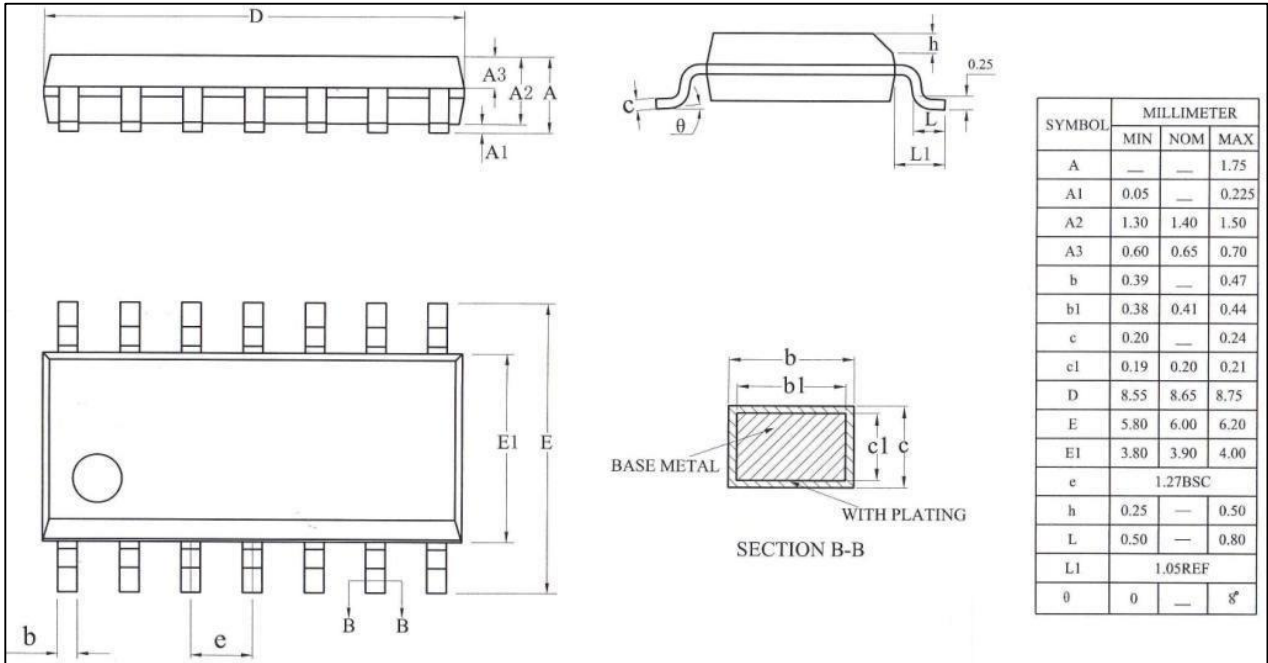


### 16.4 SOP16 (1.27mm pitch)





16.5 SOP14





## 17 修订记录

版本	日期	修订内容
V1.0	2023-10-8	用户手册发布初版;
V1.1	2023-12-12	新增 SOP16 封装, 封装代号 B0K;

Sinomcu.com