晟矽微电 8 位单片机

MC32F7073
用户手册

V1.2

目录

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 2/4/8/16/32/64 分频
  - 系统低频时钟下 FCPU 固定为 FLOSC 的 2 分频
- 程序存储器
  - 8K×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
  - 384 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
  - 256×16 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 22 个 I/O
  - P0（P00~P07），P1（P10~P17），P2（P20~P25）
  - 所有端口均支持施密特输入，均支持推挽输出
  - P05 可复用为外部复位 RST 输入，P01/P02 复用为 IIC 接口时为开漏输出
  - 所有端口均内置上拉和下拉电阻，均可单独使能
  - P0 端口输出电流 2 级可配置
  - P1 和 P2 所有端口均为大电流端口（Ioh/Iol 典型值=20mA/30mA@VDD=5V）
  - P00/P20/P21/P03 可复用为外部中断输入，支持外部中断唤醒功能
  - P1 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/16MHz），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
- 内部自振式看门狗计数器（WDT）
  - 溢出时间可配置：64ms/2048ms
  - 工作模式可配置：始终开启、始终关闭、低功耗模式下关闭
- 4 个定时器
  - 8 位定时器 T0，可实现外部计数、BUZ、PWM 功能（可扩展为 1 对带死区互补 PWM）
  - 16 位定时器 T1，可实现外部计数和 16 位 PWM 功能
  - 16 位定时器 T2，可用作 UART 波特率发生器
  - 8 位定时器 T3，可实现 3 路共周期独立占空比的 PWM
- 1 个 12 位高精度 SAR 型 ADC

- ✧ 14 路外部通道：AN0~AN13；2 路内部通道：GND、VDD/4
- ✧ 参考电压可选：VDD、内部参考电压 VIR（2V/3V/4V）
- ✧ ADC 时钟：FHIRC 的 8/16/32/64 分频
- ✧ 支持零点校准

■ 1 组总线通讯 IIC 主机接口
- ✧ 支持 7 位地址编码的单主机模式
- ✧ 通讯速率可选（实际速率受芯片及外围电路影响）：100Kbps/400Kbps/800Kbps/1Mbps
- ✧ 发送完起始或停止信号、发送或接收完应答信号等事件发生时，可触发中断

■ 1 组异步通讯 UART 接口
- ✧ 支持 8 位同步半双工、8 位/9 位异步全双工等多种工作方式
- ✧ 波特率可选 CPU 时钟分频、或定时器溢出频率分频
- ✧ 支持 UART 通讯出错检测及地址自动识别等增强功能
- ✧ 2 组端口 RX0/TX0 或 RX1/TX1 可选

■ 中断
- ✧ 外部中断（INT0~INT3），键盘中断（P10~P17）
- ✧ 定时器中断（T0~T3）
- ✧ ADC 中断
- ✧ IIC 中断，UART 中断

■ 低电压复位 LVR
- ✧ 2.0V/2.4V/2.7V/3.6V

■ 工作电压
- ✧ VLVR27 ~ 5.5V @ Fcpu = 0~8MHz
- ✧ VLVR20 ~ 5.5V @ Fcpu = 0~4MHz

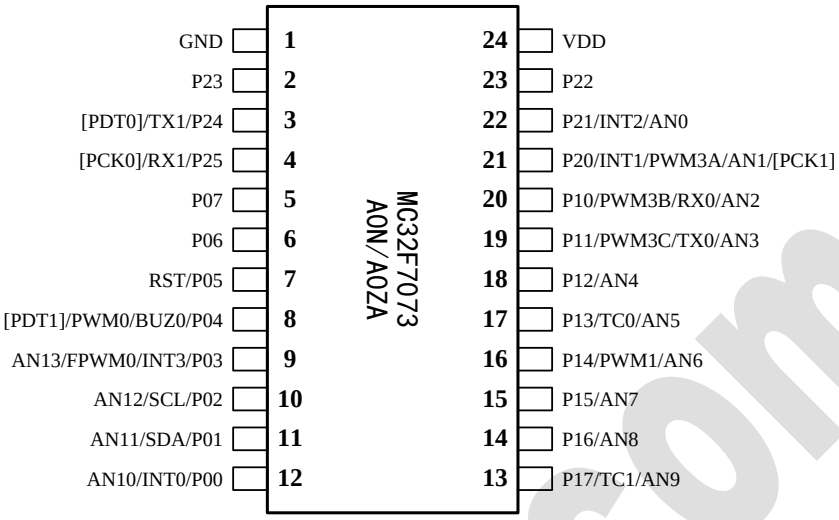
■ 封装形式
- ✧ SOP24/TSSOP24/SOP20/TSSOP20/SSOP20/SOP16

1.2 订购信息

产品名称	封装形式	备注
MC32F7073A0N	SOP24	
MC32F7073A0ZA	TSSOP24	
MC32F7073A0M	SOP20	
MC32F7073A0Y	TSSOP20	
MC32F7073A0YI	SSOP20	e=0.635
MC32F7073A0K	SOP16	
MC32F7073A1K	SOP16	

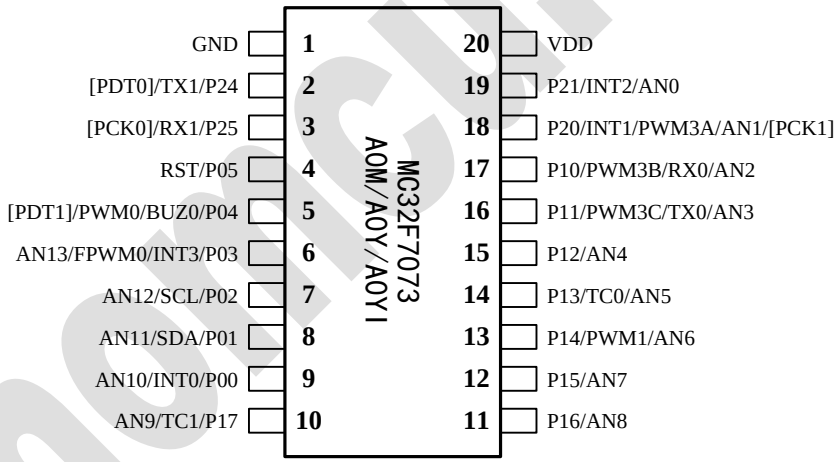
1.3 引脚排列

MC32F7073A0N/A0ZA



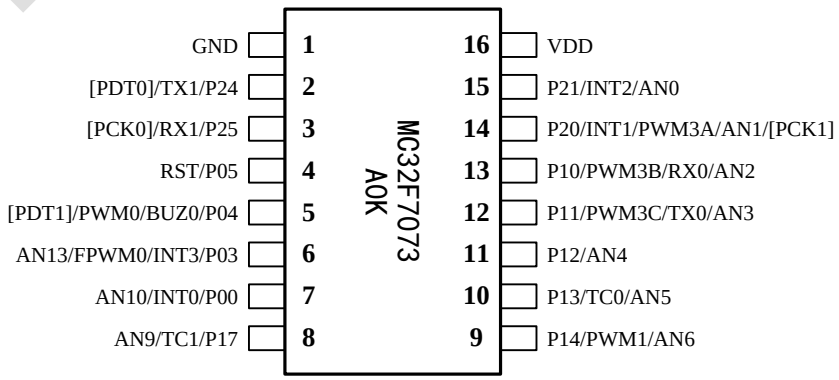
SOP24/TSSOP24

MC32F7073A0M/A0Y/A0YI



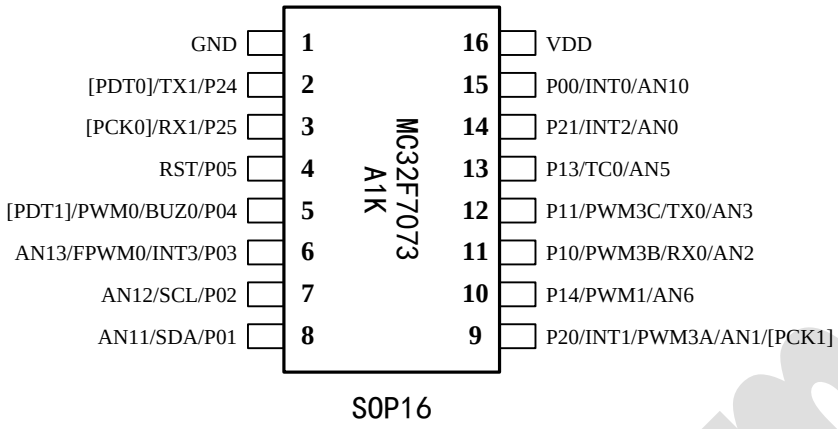
SOP20/TSSOP20/SSOP20

MC32F7073A0K



SOP16

MC32F7073A1K



SOP16

1.4 端口说明

端口名称	类型	功能说明
VDD	P	电源
GND	P	地
P0, P1, P2	D	GPIO（推挽输出），内部上/下拉
INT0~INT3	DI	外部中断输入
TC0~TC1	DI	定时器 T0~T1 的外部计数输入
PWM0, FPWM0	DO	定时器 T0 的 PWM 及其互补输出
BUZ0	DO	定时器 T0 的 BUZ 输出
PWM1	DO	定时器 T1 的 PWM 输出
PWM3A~PWM3C	DO	定时器 T3 的 3 路 PWM 输出
AN0~AN13	AI	ADC 外部输入通道
SCL, SDA	D	IIC 通讯时钟/数据端口，开漏输出
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

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

2 电气特性

2.1 极限参数

参数	符号	值	单位
电源电压	VDD	-0.3~6.0	V
I/O 输入电压	Vin	-0.3~VDD+0.3	V
工作温度	Ta	-40~85	℃
储存温度	Tstg	-65~150	℃
流入 VDD 最大电流	IVDDmax	100	mA
流出 GND 最大电流	IGNDmax	100	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℃

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(32M)/4	VLVR27		5.5	V
			Fcpu=4MHz@FHIRC(32M)/8	VLVR20		5.5	
			Fcpu=2MHz@FHIRC(32M)/16	VLVR20		5.5	
			Fcpu=1MHz@FHIRC(32M)/32	VLVR20		5.5	
			Fcpu=500KHz@FHIRC(32M)/64	VLVR20		5.5	
			Fcpu=250KHz@FHIRC(16M)/64	VLVR20		5.5	
			Fcpu=16KHz@FLIRC(32K)/2	VLVR20		5.5	
输入漏电流	Ileak	所有输入脚	VDD=5V	-1		1	μA
输入高电平	Vih	所有输入脚	SMTVS 配置	0.8VDD			V
			SMTVS 配置	2.0			
输入低电平	Vil	所有输入脚	SMTVS 配置			0.2VDD	V
			SMTVS 配置			0.8	
上拉电阻	Rpu1	P0, P1, P2	VDD=5V, Vin=0		30		KΩ
	Rpu2	SDA, SCL (P01, P02)	VDD=5V, Vin=0, IICRUS=0		4.7		KΩ
			VDD=5V, Vin=0, IICRUS=1		1.3		KΩ
下拉电阻	Rpd	P0, P1, P2	Vin=VDD=5V		30		KΩ
输出源电流	Ioh1	P0	Voh=VDD-0.6V, 正常驱动输出		10		mA
			Voh=VDD-0.6V, 增强驱动输出		20		mA
	Ioh2	P1, P2	Voh=VDD-0.6V		20		mA

输出灌电流	Iol1	P0	Vol=0.6V，正常驱动输出		15		mA
			Vol=0.6V，增强驱动输出		30		mA
	Iol2	P1, P2	Vol=0.6V		30		mA
输出驱动管开关速度		推挽输出脚	高速输出@空载			8	MHz
			低速输出@空载			2	MHz
运行模式功耗	Irun	VDD	VDD=5V Fcpu=8MHz@HIRC(32M)		3.2		mA
			Fcpu=8MHz@HIRC(16M)		3.0		mA
			Fcpu=4MHz@HIRC(16M)		2.0		mA
			Fcpu=2MHz@HIRC(16M)		1.5		mA
			Fcpu=1MHz@HIRC(16M)		1.2		mA
			Fcpu=500KHz@HIRC(16M)		1.0		mA
			Fcpu=250KHz@HIRC(16M)		900		μA
			Fcpu=16KHz@LIRC(32K)		10		μA
			VDD=3V Fcpu=8MHz@HIRC(32M)		2.0		mA
			Fcpu=8MHz@HIRC(16M)		1.9		mA
			Fcpu=4MHz@HIRC(16M)		1.3		mA
			Fcpu=2MHz@HIRC(16M)		1.0		mA
			Fcpu=1MHz@HIRC(16M)		0.9		mA
			Fcpu=500KHz@HIRC(16M)		0.7		mA
			Fcpu=250KHz@HIRC(16M)		650		μA
			Fcpu=16KHz@LIRC(32K)		6		μA
HOLD1 功耗	Ihold1	VDD	VDD=5V CPU 停，HIRC(32M)/LIRC 开		1000		μA
			CPU 停，HIRC(16M)/LIRC 开		800		μA
			VDD=3V CPU 停，HIRC(32M)/LIRC 开		700		μA
			CPU 停，HIRC(16M)/LIRC 开		600		μA
HOLD2 功耗	Ihold2	VDD	VDD=5V, CPU 停，HIRC 关，LIRC 开		3	6	μA
			VDD=3V, CPU 停，HIRC 关，LIRC 开		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		0.4		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		μA
			VDD=5V，休眠模式，WDT 关，LVR 开		10	20	μA
			VDD=3V，休眠模式，WDT 关，LVR 开		6		μA
低压复位电压	VLVR	VDD	LVRVS 配置	-10%		+10%	V
LVR 回滞电压		VDD			6%	12%	

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

2.3 交流电气特性

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

2.4 ADC 特性参数

VDD=5V，T=25℃

特性	符号	条件	最小	典型	最大	单位
ADC 有效工作电压	VADC	T=-40℃~85℃	2.5		5.5	V
积分非线性误差	INL	VREF=VDD, FADC=1MHz, Tcon=27μs			±4	LSB
微分非线性误差	DNL	VREF=VDD, FADC=1MHz, Tcon=27μs			±2	LSB
零点偏移误差	EZ	VREF=VDD, FADC=1MHz, Tcon=27μs			±4	LSB
增益误差	ET	VREF=VDD, FADC=1MHz, Tcon=27μs			±4	LSB
转换时钟	FADC	VDD=5V			2	MHz
转换时间	Tcon		16		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	μA
ADC 动态电流	IADD	VDD=5V, AD 转换中		1	3	mA
ADC 静态电流	IADS	VDD=5V, ADC 关闭		0.1	1	μA
模拟信号源推荐阻抗	ZAIN				10	KΩ
内部 1/4 分压电阻总值	RVDDI	Vin=VDD=2.5V~5.5V		24		KΩ
电阻分压比值			-1%	1/4	+1%	VDD
ADC 参考电压	VREF	选择 VDD		VDD		V
		选择内部参考电压 VIR, T=25℃	-1%	2	+1%	
		选择内部参考电压 VIR, T=-40℃~85℃	-3%		+3%	
		选择内部参考电压 VIR, T=-40℃~85℃	-5%	3/4	+5%	
VIR 有效工作电压	VVIR	选择内部参考电压 VIR	VIR+0.5		5.5	V

2.5 EEPROM 特性参数

VDD=5V

特性	符号	条件	最小	典型	最大	单位
EEPROM 读操作电压	VEERD	T=-40℃~85℃	1.8		5.5	V
EEPROM 写操作电压	VEEWR	T=-40℃~85℃	2.0		5.5	V
EEPROM 写操作电流	IEEWR	T=-40℃~85℃				mA
EEPROM 单地址写入时间	TEEWR	VDD=2.0V~5.5V, T=-40℃~85℃			5	ms
EEPROM 擦写次数		VDD=5V, T=25℃	10000			cycle

3 CPU 与存储器

3.1 指令集

芯片的指令集为精简指令集。

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

指令汇总表

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

MOVRA R	将 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：

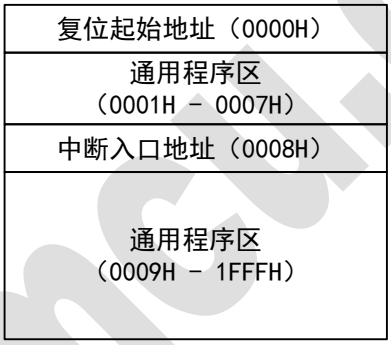
1、 A-算术逻辑单元累加器 ALU，R-数据存储器，I-立即数，K-程序存储器地址，TOS-堆栈栈顶；

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，8K×16 位的地址空间范围为 0000H~1FFFH。程序存储器地址分配如下图所示：



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

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

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

3.3 数据存储器

芯片的数据存储器包括通用数据存储器 GPR（384 字节）和特殊功能寄存器 SFR，地址映射如下表所示。其中 GPR0 可直接寻址或通过 INDF0/INDF2 间接寻址，GPR1 和 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（256×16 位），需通过 SFR 进行读写操作，详细说明请参见后续章节。

数据存储器区地址映射表

地址	类型	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
000H-0FFH	GPR0	通用数据存储器区 0							
100H-17FH	GPR1	通用数据存储器区 1							
180H-187H	SFR	INDF0	INDF1	INDF2	HIBYTE	FSR0	FSR1	PCL	PFLAG
188H-18FH		MCR	INDF3	INTE0	INTF0	OSCMR	INTE1	INTF1	P1KBCR
190H-197H		IOP0	OEP0	PUP0	PDP0	IOP1	OEP1	PUP1	PDP1
198H-19FH		IOP2	OEP2	PUP2	PDP2				
1A0H-1A7H		T0CR	T0CNT	T0LOAD	T0DATA	PWM0CR0	PWM0CR1		
1A8H-1AFH		T1CR	T1CNTH	T1CNTL	T1LOADH	T1LOADL	T1DATAH	T1DATAL	EEDRH
1B0H-1B7H		IICCR	IICSR	IICDR	HIRCCAL	EECR	EEPR	EEAR	EEDRL
1B8H-1BFH		ADCR0	ADCR1	ADRH	ADRL	ADIOS0	ADIOS1	OSADJCR	
1C0H-1C7H		T2CR	T2CNTH	T2CNTL	T2LOADH	T2LOADL	T3CR	T3CNT	T3LOAD
1C8H-1CFH		SCON	SBUF	SADDR	SADEN	T3DATA	T3DATB	T3DATC	PWM3CR
1D0H-1D7H							PMAP	DBGCR	DBGPR
1D8H-1FFH	保留	保留							

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

数据存储器寻址方式地址组成

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

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

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

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

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

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

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

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

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

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

3.4 堆栈

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

3.5 控制寄存器

数据指针寄存器 0

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

BIT[7:0] **FSR0[7:0]** – 数据指针寄存器 0

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

数据指针寄存器 1

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

BIT[7:0] **FSR1[7:0]** – 数据指针寄存器 1

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

间接寻址寄存器 0

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

BIT[7:0] **INDF0[7:0]** – 间接寻址寄存器 0

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

间接寻址寄存器 1

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

BIT[7:0] **INDF1[7:0]** – 间接寻址寄存器 1

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

间接寻址寄存器 2

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

BIT[7:0] **INDF2[7:0]** – 间接寻址寄存器 2

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

间接寻址寄存器 3

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

BIT[7:0] **INDF3[7:0]** – 间接寻址寄存器 3

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

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

字操作高字节缓存器

	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。

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 运算结果))；

CPU 状态寄存器

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

BIT[2] **Z** – 零标志位

0： 算术或逻辑运算的结果不为零；

1： 算术或逻辑运算的结果为零；

BIT[1] **DC** – 半字节进位/借位标志位

0： 加法运算中半字节无进位；减法运算中半字节有借位；

1： 加法运算中半字节有进位；减法运算中半字节无借位；

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

杂项控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
MCR	GIE	-	TO	PD	INT1M1	INT1M0	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，或已执行 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，或已执行 CLRWDT 指令；
1： 已执行 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：

符号	功能说明
HIRCFS	HIRC 振荡频率选择： FHIRC=32MHz；　　FHIRC=16MHz；
FCPUS	高频时钟下 FCPU 分频选择：（FHOSC 为 32MHz 的 FHIRC 时，FCPU 不支持 2 分频） FCPU=FHOSC/2；　　FCPU=FHOSC/4；　　FCPU=FHOSC/8； FCPU=FHOSC/16；　　FCPU=FHOSC/32；　　FCPU=FHOSC/64；
RSTEN	RST 外部复位端口设置： P05 为外部复位脚；　　P05 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 FCPU 决定的工作电压特性） 2.0V；2.4V；2.7V；3.6V；
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： 64ms；　　2048ms；
SMTVS	端口施密特阈值选择： 2.0V/0.8V；　　0.8VDD/0.2VDD；
PODRVS	P0 输出驱动电流选择： 正常驱动输出；　　增强驱动输出；
SPDS	端口输出驱动管开关速度选择： 高速输出；　　低速输出；
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： 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；

4 系统时钟

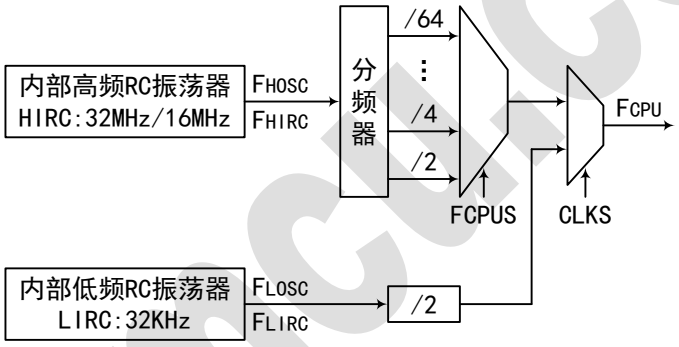
芯片内部电路均在系统高频时钟 FHOSC 或系统低频时钟 FLOSC 下工作，系统及部分外设模块的时钟源还可在 FHOSC 和 FLOSC 之间切换。

系统高频时钟 FHOSC 固定为内部高频 RC 振荡器 HIRC（32MHz/16MHz）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 FCPU 通过配置字 FCPUS 选择；FLOSC 下 FCPU 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。

系统时钟示意图



4.1 内部高频 RC 振荡器

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

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/16MHz。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。

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/16MHz -10%）—（32MHz/16MHz +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/16MHz）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。

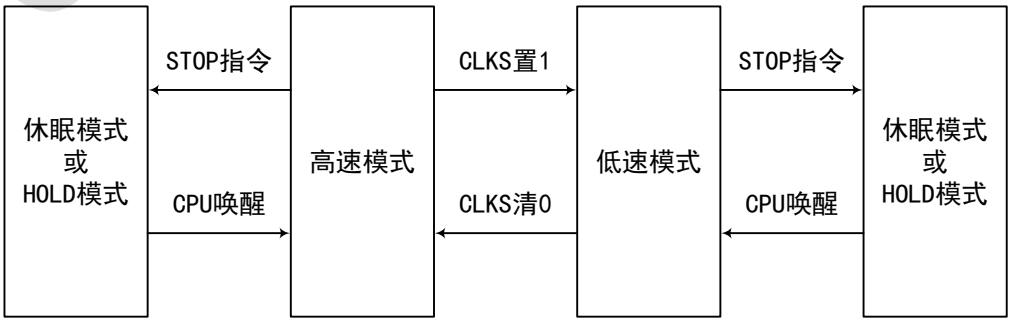
4.3 系统工作模式

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。

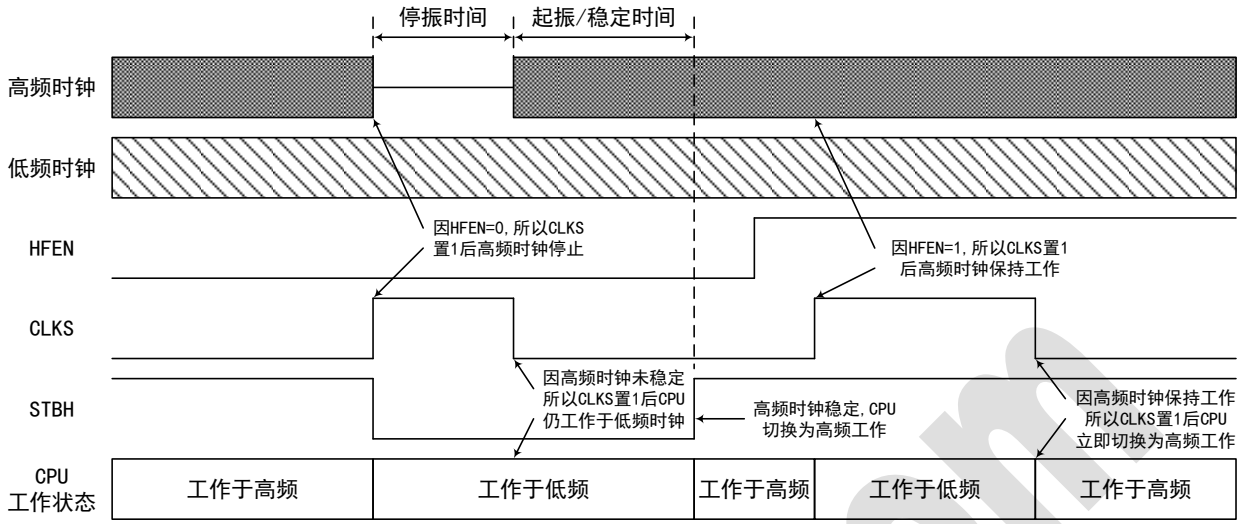
工作模式	模式切换条件	系统工作状态
高速	任意模式下，系统复位	CPU 高速运行，高/低频时钟源均工作
	低速模式下，CLKS 清 0	
	HOLD1/HOLD2/休眠模式下，CPU 唤醒（@CLKS=0）	
低速	高速模式下，CLKS 置 1	CPU 低速运行，低频时钟源工作，高频时钟源由使能位 HFEN 决定
	HOLD1/HOLD2/休眠模式下，CPU 唤醒（@CLKS=1）	
HOLD1	高/低速模式下，执行 STOP 指令（@HFEN=1）	CPU 暂停，高频时钟源工作，低频时钟源由使能位 LFEN 决定
HOLD2	高/低速模式下，执行 STOP 指令（@HFEN=0, LFEN=1）	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 的钟源为 LIRC，WDT 开启时 LIRC 将一直工作而不受系统工作模式影响。

工作模式切换示意图



高低频时钟切换时序图



振荡器模式寄存器

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

BIT[5] **STBL** – 低频时钟源振荡状态标志位

0： 低频时钟源停振或未稳定；

1： 低频时钟源已稳定振荡；

BIT[4] **STBH** – 高频时钟源振荡状态标志位

0： 高频时钟源停振或未稳定；

1： 高频时钟源已稳定振荡；

BIT[2] **CLKS** – CPU 时钟源选择位

0： 系统高频时钟作为 CPU 时钟源；

1： 系统低频时钟作为 CPU 时钟源；

BIT[1] **LFEN** – 低频时钟源使能位

0： 在休眠/HOLD 模式下，低频时钟源暂停工作；

1： 低频时钟源始终工作；

BIT[0] **HFEN** – 高频时钟源使能位

0： 在低速/休眠/HOLD 模式下，高频时钟源暂停工作；

1： 高频时钟源始终工作；

4.4 低功耗模式

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

执行 STOP 指令可使系统进入低功耗模式，同时对系统会产生以下影响：

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

以下情况可使系统退出低功耗模式：

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

注：

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

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

5 复位

5.1 复位条件

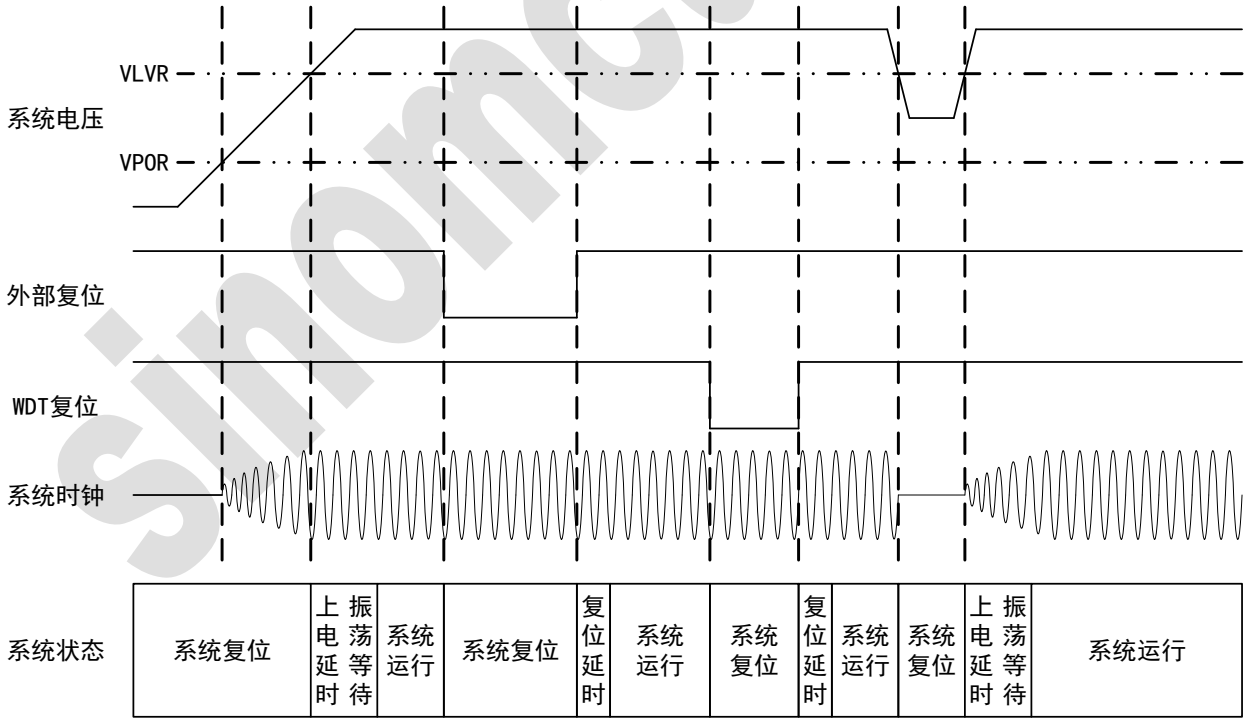
芯片共有如下几种复位方式：

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

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

上电复位 POR 和低电压复位 LVR 会关闭系统主时钟振荡器，复位解除后才重新开启振荡器，因为振荡器起振和稳定需要一定的时间，所以系统将保持一定时间的上电延时（典型值为 9ms）以待振荡器稳定振荡后才开始工作；而外部复位、WDT 复位则不会关闭主时钟振荡器，复位解除后系统将在较短的复位延时后即开始工作。

下图是复位产生和系统工作状态之间的时序关系示意图：



注：若应用系统在上电或掉电回升时芯片的 VDD 电压上升较慢，则应在复位后 CPU 开始工作时先进行软件延时，以确保芯片开始工作时 VDD 电压已稳定在 Fcpu 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 Vih；
(4) 初始化所有初始值确定的寄存器；
(5) 开启主时钟振荡器，并等待一段时间以待振荡器稳定；
(6) 上电复位结束，CPU 开始执行指令；

5.3 外部复位

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

5.4 低电压复位

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

5.5 看门狗复位

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

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

6 I/O 端口

6.1 通用 I/O 功能

芯片的输入/输出端口包括两组 8 位端口 P0、P1，和一组 6 位端口 P2。所有端口均支持施密特输入，均支持推挽输出。

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

端口数据寄存器

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

BIT[7:0] **P0nD** – P0n 端口数据位（n=7-0）

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

BIT[7:0] **P1nD** – P1n 端口数据位（n=7-0）

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IOP2	–	–	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）

端口方向寄存器

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

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

0： 端口作为输入口，读端口操作将读取端口的电平状态；

1： 端口作为输出口，读端口操作将读取端口的数据位值；

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

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

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

0： 端口作为输入口，读端口操作将读取端口的电平状态；

1： 端口作为输出口，读端口操作将读取端口的数据位值；

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OEP2	-	-	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时是否有效。端口处于输出状态时，上/下拉电阻及其控制位无效。

上拉电阻控制寄存器

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

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

0： 端口内部上拉电阻无效；

1： 端口内部上拉电阻有效；

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

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

0： 端口内部上拉电阻无效；

1： 端口内部上拉电阻有效；

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

下拉电阻控制寄存器

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

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

0： 端口内部下拉电阻无效；

1： 端口内部下拉电阻有效；

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

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

0： 端口内部下拉电阻无效；

1： 端口内部下拉电阻有效；

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PDP2	–	–	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）。

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]
ADIOS0	AN7DC	AN6DC	AN5DC	AN4DC	AN3DC	AN2DC	AN1DC	AN0DC
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] **ANnDC** – ANn 端口数字功能控制位（n=7-0）

0： 使能端口的数字 I/O 功能；

1： 关闭端口的数字 I/O 功能；

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADIOS1	–	–	AN13DC	AN12DC	AN11DC	AN10DC	AN9DC	AN8DC
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] **ANnDC** – ANn 端口数字功能控制位（n=13-8）

0： 使能端口的数字 I/O 功能；

1： 关闭端口的数字 I/O 功能；

7 定时器 TIMER

7.1 看门狗定时器 WDT

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

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

执行 CLRWDT 指令或 STOP 指令将清零 WDT 计数器。

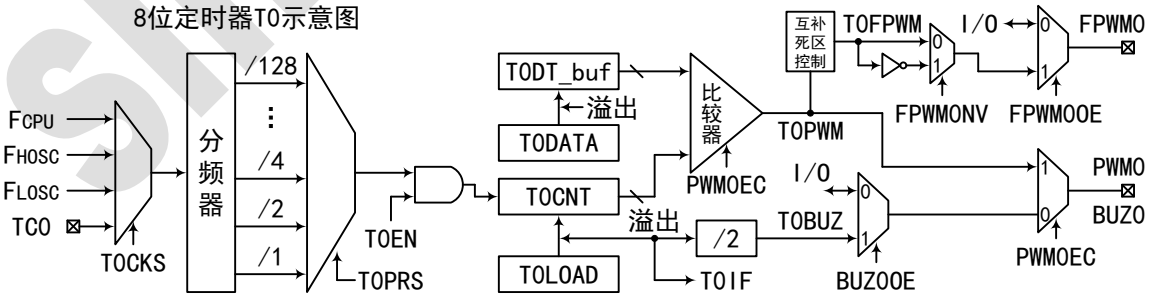
WDT 溢出时间可配置为 64ms/2048ms。

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

7.2 定时器 T0

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

- ✧ 可通过预分频器设置时钟频率，可通过重载寄存器控制计数周期；
- ✧ 支持 8 位 PWM 输出，可通过比较寄存器设置 PWM 占空比；
- ✧ 可扩展为 1 对带死区互补 PWM 输出；
- ✧ 支持 BUZ 输出；
- ✧ 支持溢出中断和溢出唤醒功能；



定时器 T0，可通过寄存器位 T0CKS 选择时钟源，通过 T0PRS 选择时钟预分频比，所选时钟源通过预分频器后产生 T0 计数器 T0CNT 的计数时钟（上升沿计数）。写 T0CNT 将清零预分频计数器，而预分频比保持不变。

T0EN=0 时，T0CNT 保持不变，写重载寄存器 T0LOAD 将立即载入 T0CNT；T0EN=1 时，T0CNT 递减计数，计数到 0 的时钟结束后产生溢出信号并触发中断，中断标志 T0IF 将被置 1，同时 T0 自动将当前 T0LOAD 值载入 T0CNT 并重新开始计数。

如图所示，定时器 T0 可实现 BUZ 功能（BUZ0）。当 BUZ0OE=1 时，端口将输出频率为 T0 溢出频率 2 分频的蜂鸣器驱动信号（需 PWM0EC=0）。

如图所示，定时器 T0 可实现 PWM 功能（PWM0），可通过寄存器位使能或关闭 PWM 功能，并控制端口是否输出 PWM 波形。PWM0 关闭时 T0PWM 信号为低电平。PWM0 使能后 T0CNT 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 T0DATA 相等时，T0PWM 变为高电平；当计数溢出时，T0PWM 变为低电平。

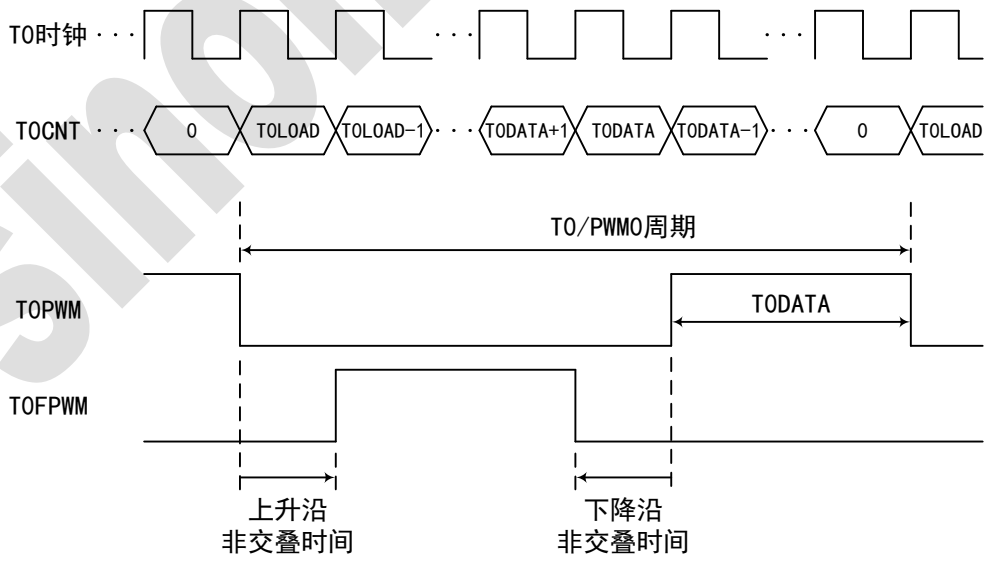
T0DATA 配有 1 个 8 位比较缓冲器（T0DT_buf）用于与 T0CNT 比较，PWM0 关闭时写 T0DATA 将立即载入缓冲器中，而 PWM0 使能后写 T0DATA 则将在 T0 溢出时才载入缓冲器中。若要首个 PWM 周期和占空比准确，需先写重载寄存器和比较寄存器，再使能 PWM，最后开启定时器。

T0PWM 信号的占空比计算如下：

- ✧ 高电平时间 =（T0DATA）× T0CNT 计数时钟周期
- ✧ 周期（T0 溢出时间）=（T0LOAD + 1）× T0CNT 计数时钟周期
- ✧ 占空比（高电平时间/周期）=（T0DATA）/（T0LOAD + 1）

PWM0 还可从 T0PWM 信号衍生 1 路带死区（2 路互补信号高电平非交叠时间）控制的互补信号 T0FPWM，从而扩展为 1 对互补且死区可设的 PWM 输出。

PWM 互补及死区波形示意



注：

1、 应用互补 PWM 时，前后死区的总时间应小于 T0PWM 低电平时间，以确保 T0FPWM 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；

定时器 T0 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T0CR	T0EN	PWM0EC	BUZ0OE	T0CKS1	T0CKS0	T0PRS2	T0PRS1	T0PRS0
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] **T0EN** – 定时器 T0 使能位

0： 关闭定时器 T0；

1： 开启定时器 T0；

BIT[6] **PWM0EC** – PWM0 使能位及端口输出控制位

0： 关闭 PWM0 功能，并禁止端口输出脉宽调制波形；

1： 使能 PWM0 功能，并允许端口输出脉宽调制波形；

BIT[5] **BUZ0OE** – BUZ0 端口输出使能位

0： 禁止端口输出蜂鸣器驱动波形；

1： 允许端口输出蜂鸣器驱动波形（仅 PWM0EC=0 时有效）；

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

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

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。

定时器 T0 比较寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T0DATA	T0DATA7	T0DATA6	T0DATA5	T0DATA4	T0DATA3	T0DATA2	T0DATA1	T0DATA0
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] **T0DATA[7:0]** – T0 比较寄存器，用于设置 PWM0 的占空比

PWM0 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PWM0CR0	–	–	–	–	–	–	FPWM0NV	FPWM0OE
R/W	–	–	–	–	–	–	R/W	R/W
初始值	–	–	–	–	–	–	0	0

BIT[1] **FPWM0NV** – FPWM0 端口输出取反控制位

0： 端口输出正向波形；

1： 端口对电平取反后输出；

BIT[0] **FPWM0OE** – FPWM0 端口输出使能位

0： 禁止端口输出脉宽调制波形；

1： 允许端口输出脉宽调制波形（仅 PWM0EC=1 时有效）；

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PWM0CR1	–	–	FPWM0E5	FPWM0E4	FPWM0E3	FPWM0E2	FPWM0E1	FPWM0E0
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] **FPWM0E[5:0]** – T0FPWM 上升/下降沿非交叠时间选择位

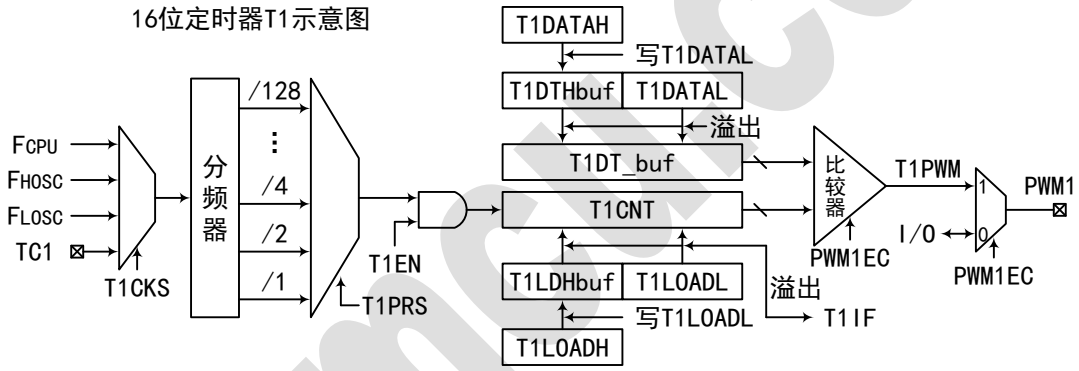
FPWM0E[5:0]	上升沿非交叠时间	下降沿非交叠时间
00 0000	1 个计数时钟周期	1 个计数时钟周期
00 0001	2 个计数时钟周期	2 个计数时钟周期

00 0010	3 个计数时钟周期	3 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

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 位重载寄存器和 16 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 16 位 PWM 输出，可通过比较寄存器设置 PWM 占空比；
- ✧ 支持溢出中断和溢出唤醒功能；



定时器 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 的低字节 T1CNTL 配有读缓冲器，读 T1CNTL 实际为读取缓冲器中内容，而 T1CNTL 实际内容仅在读 T1CNTH 时才同步自动载入缓冲器中，因此读 T1CNT 需先读 T1CNTH（硬件同时自动将 T1CNTL 内容载入缓冲器中）再读 T1CNTL（实际为读取低字节缓冲器中内容），即可保证读取 16 位的 T1CNT 时同步读取 T1CNTH 和 T1CNTL 中的内容。

而写 T1CNTH 或 T1CNTL 均为直接写入，无法实现 16 位 T1CNT 高低字节的同步写入。

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 PWM 功能（PWM1），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。PWM1 关闭时 T1PWM 信号为低电平。PWM1 使能后 T1CNT 从重载值开始递减计数直到计数溢出为一个 PWM 周期：当计数到与比较寄存器 T1DATA 相等时，T1PWM 变为高电平；当计数溢出时，T1PWM 变为低电平。

16 位 T1DATA 配有 1 个高位缓冲器和 1 个 16 位比较缓冲器（T1DT_buf），写 T1DATAL 时会同时将 T1DATAH 内容载入高位缓冲器中，写 T1DATA 需先写 T1DATAH 再写 T1DATAL。此时若 PWM1 关闭，则会同时再将[高位缓冲器:T1DATAL]载入比较缓冲器中；若 PWM1 已使能，则会在 T1 溢出后才将[高位缓冲器:T1DATAL]载入比较缓冲器中。若要首个 PWM 周期和占空比准确，需先写重载寄存器和比较寄存器，再使能 PWM，最后开启定时器。

T1PWM 信号的占空比计算如下：

- ✧ 高电平时间 =（T1DATA）× T1CNT 计数时钟周期
- ✧ 周期（T1 溢出时间）=（T1LOAD + 1）× T1CNT 计数时钟周期
- ✧ 占空比（高电平时间/周期）=（T1DATA）/（T1LOAD + 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]
T1CR	T1EN	PWM1EC	-	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] **PWM1EC** – PWM1 使能位及端口输出控制位

0： 关闭 PWM1 功能，并禁止端口输出脉宽调制波形；

1： 使能 PWM1 功能，并允许端口输出脉宽调制波形；

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

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

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。

定时器 T1 比较寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T1DATAH	T1DATA15	T1DATA14	T1DATA13	T1DATA12	T1DATA11	T1DATA10	T1DATA9	T1DATA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **T1DATA[15:8]** – T1 比较寄存器高 8 位，用于设置 PWM1 的占空比

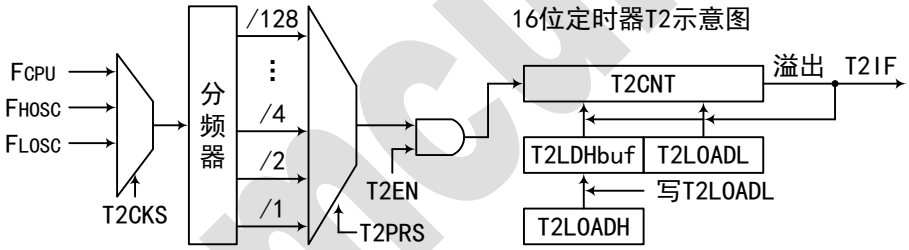
	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T1DATAL	T1DATA7	T1DATA6	T1DATA5	T1DATA4	T1DATA3	T1DATA2	T1DATA1	T1DATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **T1DATA[7:0]** – T1 比较寄存器低 8 位，用于设置 PWM1 的占空比

7.4 定时器 T2

定时器 T2 为 16 位定时器，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；
- ✧ 支持溢出中断和溢出唤醒功能；
- ✧ 可作为 UART 波特率发生器；



定时器 T2 的定时功能与定时器 T1 完全相同。

定时器 T2 还可用作 UART 模块的波特率发生器。当 UART 工作在方式 1 或方式 3 时，其波特率为 T2 溢出频率的 1/16。

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	SMOD	SSTAT	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] **SMOD** – UART 工作方式 2 波特率选择位

0： UART 工作方式 2 的波特率为 FCPU/64；

1： UART 工作方式 2 的波特率为 FCPU/32；

BIT[5] **SSTAT** – UART 寄存器 SCON 功能选择位

0： UART 寄存器位 SCON[7:5]的功能为 SM[0:2]；

1： UART 寄存器位 SCON[7:5]的功能依次为 FERR、RXOV；

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

定时器 T2 计数器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T2CNTH	T2CNT15	T2CNT14	T2CNT13	T2CNT12	T2CNT11	T2CNT10	T2CNT9	T2CNT8
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[15:8]** – T2 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]
T2CNTL	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 计数器低 8 位，为可读写的递减计数器

定时器 T2 重载寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T2LOADH	T2LOAD15	T2LOAD14	T2LOAD13	T2LOAD12	T2LOAD11	T2LOAD10	T2LOAD9	T2LOAD8
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[15:8]** – T2 重载寄存器高 8 位，用于设置 T2 的计数周期

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T2LOADL	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

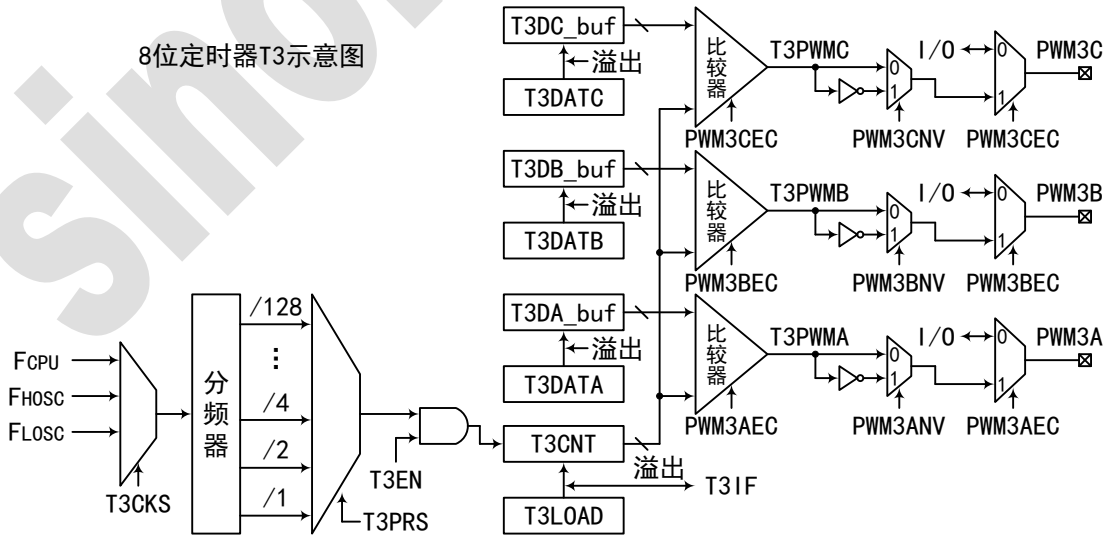
BIT[7:0] **T2LOAD[7:0]** – T2 重载寄存器低 8 位，用于设置 T2 的计数周期

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；若 T2 用于 UART，则重载寄存器值需大于 3。

7.5 定时器 T3

定时器 T3 为 8 位定时器，包含 1 个 8 位递减计数器、可编程预分频器、控制寄存器、8 位重载寄存器和 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；
- ✧ 支持溢出中断和溢出唤醒功能；



定时器 T3，可通过寄存器位 T3CKS 选择时钟源，通过 T3PRS 选择时钟预分频比，所选时钟源通过预分频器后产生 T3 计数器 T3CNT 的计数时钟（上升沿计数）。写 T3CNT 将清零预分频计数器，而预分

频比保持不变。

T3EN=0 时，T3CNT 保持不变，写重载寄存器 T3LOAD 将立即载入 T3CNT；T3EN=1 时，T3CNT 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 T3IF 将被置 1，同时 T3 自动将当前 T3LOAD 值载入 T3CNT 并重新开始计数。

如图所示，定时器 T3 可实现 3 路共周期的 PWM 功能（PWM3x，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。PWM3x 关闭时 T3PWMx 信号为低电平。PWM3x 使能后 T3CNT 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 T3DATx 相等时，T3PWMx 变为高电平；当计数溢出时，T3PWMx 变为低电平。

T3DATx 均配有 1 个 8 位比较缓冲器（T3Dx_buf）用于与 T3CNT 比较，PWM3x 关闭时写 T3DATx 将立即载入缓冲器中，而 PWM3x 使能后写 T3DATx 则将在 T3 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。

T3PWMx 信号（x=A,B,C，下同）的占空比计算如下：

- ✧ 高电平时间 =（T3DATx）× T3CNT 计数时钟周期
- ✧ 周期（T3 溢出时间）=（T3LOAD + 1）× T3CNT 计数时钟周期
- ✧ 占空比（高电平时间/周期）=（T3DATx）/（T3LOAD + 1）

定时器 T3 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T3CR	T3EN	-	-	T3CKS1	T3CKS0	T3PRS2	T3PRS1	T3PRS0
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] **T3EN** – 定时器 T3 使能位

0： 关闭定时器 T3；

1： 开启定时器 T3；

BIT[4:3] **T3CKS[1:0]** – T3 时钟源选择位

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

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

T3PRS[2:0]	T3 时钟预分频比
000	1 ： 1
001	1 ： 2
010	1 ： 4

011	1 ： 8
100	1 ： 16
101	1 ： 32
110	1 ： 64
111	1 ： 128

定时器 T3 计数器

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

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

定时器 T3 重载寄存器

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

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

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。

定时器 T3 比较寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T3DATA	T3DATA7	T3DATA6	T3DATA5	T3DATA4	T3DATA3	T3DATA2	T3DATA1	T3DATA0
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] **T3DATA[7:0]** – T3 比较寄存器 A，用于设置 PWM3A 的占空比

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T3DATB	T3DATB7	T3DATB6	T3DATB5	T3DATB4	T3DATB3	T3DATB2	T3DATB1	T3DATB0
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] **T3DATB[7:0]** – T3 比较寄存器 B，用于设置 PWM3B 的占空比

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T3DATC	T3DATC7	T3DATC6	T3DATC5	T3DATC4	T3DATC3	T3DATC2	T3DATC1	T3DATC0
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] **T3DATC[7:0]** – T3 比较寄存器 C，用于设置 PWM3C 的占空比

PWM3 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PWM3CR	-	-	PWM3CNV	PWM3CEC	PWM3BNV	PWM3BEC	PWM3ANV	PWM3AEC
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] **PWM3CNV** – PWM3C 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] **PWM3CEC** – PWM3C 使能位及端口输出控制位

0： 关闭 PWM3C 功能，并禁止端口输出脉宽调制波形；

1： 使能 PWM3C 功能，并允许端口输出脉宽调制波形；

BIT[3] **PWM3BNV** – PWM3B 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] **PWM3BEC** – PWM3B 使能位及端口输出控制位

0： 关闭 PWM3B 功能，并禁止端口输出脉宽调制波形；

1： 使能 PWM3B 功能，并允许端口输出脉宽调制波形；

BIT[1] **PWM3ANV** – PWM3A 端口输出取反控制位

0： 端口输出正向波形；

1： 端口对电平取反后输出；

BIT[0] **PWM3AEC** – PWM3A 使能位及端口输出控制位

0： 关闭 PWM3A 功能，并禁止端口输出脉宽调制波形；

1： 使能 PWM3A 功能，并允许端口输出脉宽调制波形；

8 模数转换器 ADC

8.1 ADC 概述

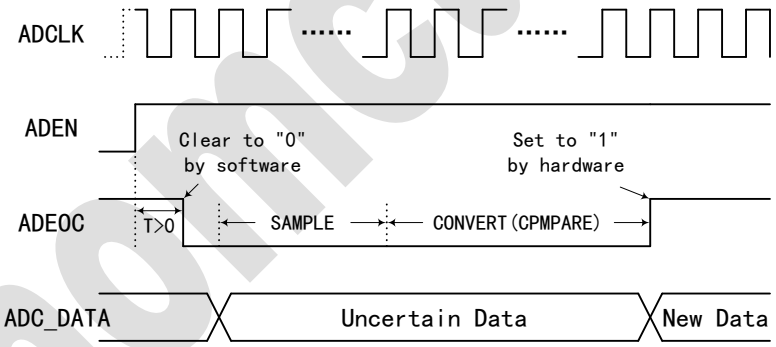
芯片内置 1 个 12 位高精度逐次逼近型的模数转换器 ADC。

- ✧ 14 路外部通道：AN0~AN13；2 路内部通道：GND、VDD/4；
- ✧ 参考电压可选：VDD、内部参考电压 VIR（2V/3V/4V）；
- ✧ ADC 时钟：FHIRC 的 8/16/32/64 分频；
- ✧ 支持零点校准；

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

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

ADC 转换时序如下图所示：



注：

1、 *AD 转换过程中或 ADEN 未使能时，ADRH/ADRL 中的数据未知，应在 AD 转换完成且 ADEN 使能的情况下读取 AD 转换结果数据；*

2、 *若选择内部参考电压 VIR，则需保证 VDD＞（VIR+0.5V），否则 VIR 实际电压将降为（VDD-0.5V）；*

3、 *使能 ADC 模块、或切换参考电压等操作后，需延时（时间＞200μs）以待电路稳定后才可启动 AD 转换；因采样保持电路的电容效应，切换输入通道后的前几次转换结果将会有偏差，建议舍弃；*

4、 *AD 转换精度受参考电压精度的影响，且内部参考电压下的转换精度，比外部参考电压下略低 2 个 LSB 左右；*

5、 *转换时钟越慢、采样时间越长，则越能过滤外部输入的波动，越能保证 AD 转换的精度；*

8.2 ADC 相关寄存器

ADC 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADCR0	ADCHS3	ADCHS2	ADCHS1	ADCHS0	ADVRS1	ADVRS0	ADEOC	ADEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	0	0	1	0

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

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

BIT[3:2] **ADVRS[1:0]** – ADC 参考电压选择位

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

BIT[1] **ADEOC** – AD 转换控制位

0： AD 转换中，完成后自动置 1；

1： 转换未开始或已完成，写 0 开始 AD 转换；

BIT[0] **ADEN** – ADC 使能位

0： 关闭 ADC；

1： 开启 ADC；

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

BIT[7] **ADRSEL** – ADC 转换结果数据格式选择位

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

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

BIT[5:4] **ADCKS[1:0]** – ADC 转换时钟选择位

ADCKS[1:0]	ADC 转换时钟 FADC
00	FHIRC/8
01	FHIRC/16
10	FHIRC/32
11	FHIRC/64

注：ADC 转换时钟不能大于 2MHz。

BIT[1:0] **ADSPS[1:0]** – ADC 采样时间选择位

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

ADC 转换结果寄存器

ADRSEL=0：

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

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

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

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

ADRSEL=1：

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

ADRH	–	–	–	–	ADR11	ADR10	ADR9	ADR8
R/W	–	–	–	–	R	R	R	R
初始值	–	–	–	–	X	X	X	X

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

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

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

ADC 零点偏移修调控制寄存器

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

BIT[7] **OSADJEN** – ADC 零点偏移修调使能位

0： ADC 零点偏移修调无效；

1： ADC 零点偏移修调有效；

BIT[5] **OSADJTD** – ADC 零点偏移修调方向选择位

0： 负向修调，即根据修调电压减小转换值（转换结果大于理论值时应选择负向修调）；

1： 正向修调，即根据修调电压增加转换值（转换结果小于理论值时应选择正向修调）；

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

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

8.3 ADC 操作步骤

模数转换操作步骤：

(1) 设置相应端口为输入端口，关闭端口的内部上/下拉电阻；

(2) 通过端口数模控制寄存器，关闭相应端口的数字 I/O 功能；

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

8.4 ADC 零点偏移修调流程

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

9 总线通讯 IIC

9.1 IIC 概述

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

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

9.2 IIC 数据传输

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

一帧数据传输以一个起始信号 START 开始，以一个停止信号 STOP 或重复起始信号 RE-START 结束，一个重复起始信号 RE-START 也是下一帧数据传输的开始（需从机支持重复起始信号 RE-START），期间总线不被释放。

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

IIC 通讯时可触发中断，中断触发事件发生后，中断标志将被置 1，软件需及时清 0 标志位以便 IIC 正确响应下一次中断，并可通过状态标志位 IICSTAT[2:0]判断上一步已完成的传输过程。

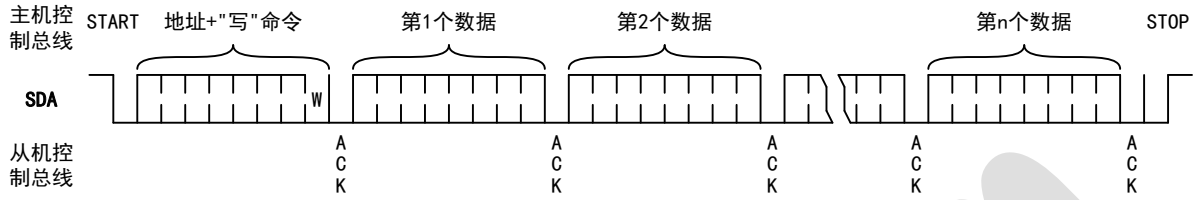
IIC 模块主机模式在通讯时可选择两种不同的中断触发方式：

IICIM=0 时，当主机发送完 8 位数据再接收完从机应答信号、或接收完从机 8 位数据再发送完应答信号、或发送完 STOP 信号后，将触发 IIC 中断；

IICIM=1 时，当主机发送完 START 或 STOP 信号、或发送完 8 位数据再接收完从机应答信号、或接收完从机 8 位数据、或发送完应答信号后，将触发 IIC 中断。

主机到从机的数据传输

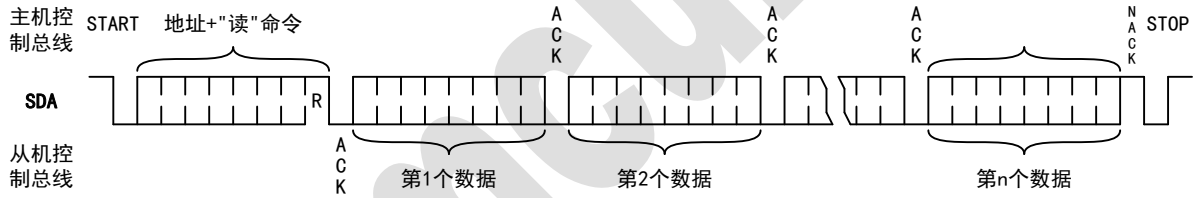
主机先发送起始信号 START，再发送一个包含“写”命令的控制字节，从机返回一个 ACK，然后主机开始发送数据字节，从机接收完每一个数据字节后均返回一个 ACK，主机在接收到最后一个字节从机返回的 ACK 后，发送停止信号 STOP 结束本次数据传输。



主机发送数据的格式

从机到主机的数据传输

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



主机接收数据的格式

9.3 IIC 相关寄存器

IIC 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IICCR	IICEN	IICRUS	IICSPD1	IICSPD0	IICSTAT2	IICSTAT1	IICSTAT0	IICIM
R/W	R/W	R/W	R/W	R/W	R	R	R	R/W
初始值	0	0	0	0	0	1	0	0

BIT[7] **IICEN** – IIC 使能位

0： 关闭 IIC；

1： 使能 IIC；

BIT[6] **IICRUS** – IIC 端口内部上拉电阻选择位（被选上拉电阻仍受端口上拉电阻控制位影响）

0： 选择内部 4.7KΩ 上拉电阻；

1： 选择内部 1.3KΩ 上拉电阻；

BIT[5:4] **IICSPD[1:0]** – IIC 通讯速率选择位（实际速率受芯片及外围电路影响）

IICSPD[1:0]	IIC 时钟频率	IIC 通讯速率
00	FHIRC/160	100Kbps
01	FHIRC/40	400Kbps
10	FHIRC/20	800Kbps
11	FHIRC/16	1Mbps

BIT[3:1] **IICSTAT[2:0]** – IIC 通讯传输状态标志位

IICSTAT[2:0]	IIC 通讯传输状态
00X	主机已发送 START 信号
01X	主机已发送 STOP 信号
10X	主机已发送 8 位数据，并已接收从机应答信号
110	主机已接收从机 8 位数据，尚未发送应答信号
111	主机已接收从机 8 位数据，并已发送应答信号

注：通过状态标志位 IICSTAT[2:0]可判断 IIC 数据传输已完成哪一步，从而决定下一步将要执行的传输操作。

BIT[0] **IICIM** – IIC 中断触发方式选择位

0： 主机发送完 START 信号和接收完从机数据后，不触发中断；

1： 主机发送完 START 信号或接收完从机数据后，触发中断；

IIC 状态寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IICSR	IICSTR	IICSTP	IICRD	IICWR	IICACK	TACKS	RACKF	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	-
初始值	0	0	0	0	0	0	0	-

BIT[7] **IICSTR** – 主机模式发送 START 起始信号控制位

0： 操作未开始或已完成，写 1 开始发送起始信号；

1： 发送起始信号操作中，完成后自动清 0；

BIT[6] **IICSTP** – 主机模式发送 STOP 停止信号控制位

0： 操作未开始或已完成，写 1 开始发送停止信号；

1： 发送停止信号操作中，完成后自动清 0；

BIT[5] **IICRD** – 主机模式接收数据帧控制位

0： 操作未开始或已完成，写 1 开始接收数据帧；

1： 接收数据帧操作中，完成后自动清 0；

BIT[4] **IICWR** – 主机模式发送数据帧再接收应答控制位
0： 操作未开始或已完成，写 1 开始发送数据帧再接收从机应答信号；
1： 发送数据帧再接收从机应答信号操作中，完成后自动清 0；

BIT[3] **IICACK** – 主机模式发送应答信号控制位
0： 操作未开始或已完成，写 1 开始发送应答信号；
1： 发送应答信号操作中，完成后自动清 0；

BIT[2] **TACKS** – 应答信号发送内容选择位
0： 待发送的应答信号为应答（ACK）；
1： 待发送的应答信号为非应答（NACK）；

BIT[1] **RACKF** – 应答信号接收内容标志位
0： 接收到的应答信号为应答（ACK）；
1： 接收到的应答信号为非应答（NACK）；

IIC 数据寄存器

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

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

9.4 IIC 应用流程

主机发送数据

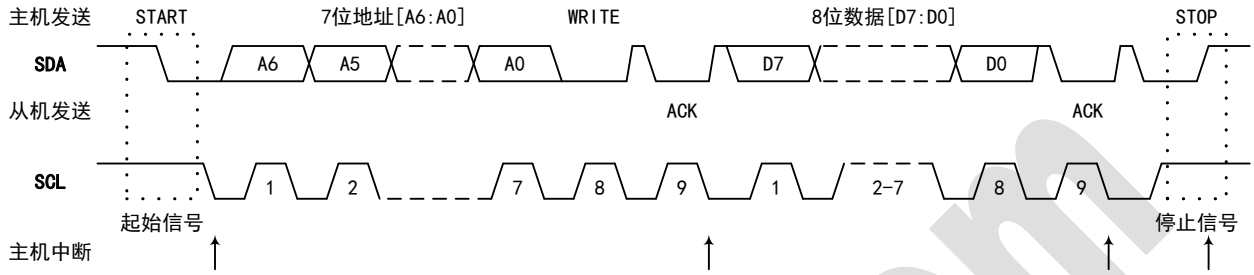
(1) 初始化 IIC 总线通讯模块：
- ✧ 通过 IICIM，选择 IIC 的中断触发方式（本流程中 IICIM=1）；
- ✧ 通过 IICSPD，选择 IIC 的通讯速率；
- ✧ 中断标志位 IICIF 清 0，中断使能位 IICIE 置 1，使能 IIC 中断；
- ✧ IICEN 置 1，使能 IIC 模块；

(2) IICSTR 置 1，主机发送一个 START 信号，启动 IIC 通讯；
(3) 发送完 START 信号后，主机产生第 1 个中断；
(4) IICIF 清 0，并将控制字节（7 位地址+1 位“写”命令）写入 IICDR；
(5) IICWR 置 1，IIC 将自动发送 IICDR 中内容及应答位时钟；
(6) 在应答位时钟接收到从机返回的应答信号（ACK/NACK）后，主机产生第 2 个中断：
- ✧ 若 RACKF=0，则表明通讯成功。将下一个发送数据写入 IICDR，然后将 IICWR 置 1，IIC 将自动发送 IICDR 中数据及应答位时钟；

- ✧ 若 RACKF=1，则表明从机无应答，通讯未成功。可将 IICSTP 置 1 发送 STOP 信号终止通讯、或将 IICSTR 置 1 发送 START 信号重新开始通讯；

(7) 循环执行（6），直到所有数据均发送完成，最后将 IICSTP 置 1 发送 STOP 信号结束通讯；

(8) 主机发送完 STOP 信号后，产生最后 1 个中断，可借此关闭 IIC 模块释放 I/O 端口。



主机发送数据的时序

注：IICIM 为 0 时，主机发送完 START 信号后不触发中断，并将自动接着发送 IICDR 中的内容再接收应答信号，接收完成后再触发中断。

主机接收数据

(1) 初始化 IIC 总线通讯模块：

- ✧ 通过 IICIM，选择 IIC 的中断触发方式（本流程中 IICIM=1）；
- ✧ 通过 IICSPD，选择 IIC 的通讯速率；
- ✧ 中断标志位 IICIF 清 0，中断使能位 IICIE 置 1，使能 IIC 中断；
- ✧ IICEN 置 1，使能 IIC 模块；

(2) IICSTR 置 1，主机发送一个 START 信号，启动 IIC 通讯；

(3) 发送完 START 信号后，主机产生第 1 个中断；

(4) IICIF 清 0，并将控制字节（7 位地址+1 位“读”命令）写入 IICDR；

(5) IICWR 置 1，IIC 将自动发送 IICDR 中内容及应答位时钟；

(6) 在应答位时钟接收到从机返回的应答信号（ACK/NACK）后，主机产生第 2 中断：

- ✧ 若 RACKF=0，则表明通讯成功。将 IICWR 置 1，IIC 将自动发送 8 位时钟，并读取总线上的数据；
- ✧ 若 RACKF=1，则表明从机无应答，通讯未成功。可将 IICSTP 置 1 发送 STOP 信号终止通讯、或将 IICSTR 置 1 发送 START 信号重新开始通讯；

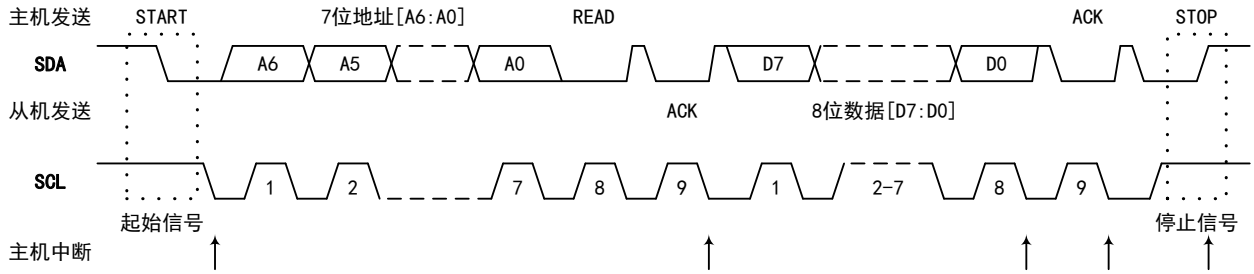
(7) 接收完总线上的 8 位数据后，主机产生第 3 个中断：

- ✧ 若继续接收，则将 TACKS 清 0，再将 IICACK 置 1，发送 ACK 应答；
- ✧ 若终止接收，则将 TACKS 置 1，再将 IICACK 置 1，发送 NACK 非应答；

(8) 发送完应答信号 ACK/NACK 后，主机产生第 4 个中断：

- ✧ 若继续接收，则将 IICRD 置 1 启动下一个数据接收；循环执行（7）（8），直到终止接收；
- ✧ 若终止接收，则将 IICSTP 置 1，发送 STOP 信号终止通讯；

(9) 发送完 STOP 信号后，主机产生最后 1 个中断，可借此关闭 IIC 模块释放 I/O 端口。



主机接收数据的时序

注：IICIM 为 0 时，主机发送完 START 信号后不触发中断，并将自动接着发送 IICDR 中的内容再接收应答信号，接收完成后再触发中断；同时，主机接收完 8 位数据后也不触发中断，而是将接着发送 TACKS 中的应答内容，发送完成后再触发中断，因此在接收最后一个字节前，需预先将 TACKS 设为 1。

10 异步通讯 UART

10.1 UART 概述

芯片内置 1 个增强型异步收发器 UART，波特率可选择 CPU 时钟的分频、或定时器 T2 溢出频率的分频，支持帧出错检测及地址自动识别等增强功能，可实现 8 位同步通讯、或 8 位/9 位异步通讯等多种通讯模式。

注：UART 时钟源为系统高频时钟，因此仅当 HIRC 工作时 UART 方可正常工作。

10.2 UART 工作方式

UART 支持 8 位同步、8 位异步、9 位异步固定波特率、9 位异步可变波特率等 4 种工作方式。在通讯之前需先初始化串口控制寄存器 SCON，选择 UART 的工作方式和波特率。若使用方式 1 或方式 3 则还需先初始化定时器 T2。

在所有 4 种工作方式中，任何将串口缓冲器 SBUF 作为目标寄存器的写操作均将启动 UART 发送。而 REN 置 1 则将使能 UART 接收，除方式 0（同步方式）外在其他方式（异步方式）中当接收端口 RX 检测到起始位时启动 UART 接收；而在方式 0 中则通过 RI 置 1（仍需 REN=1）启动 UART 接收，UART 将在 TX 端口发送时钟信号，同时在 RX 端口上串行移入/移出数据。

UART 方式列表

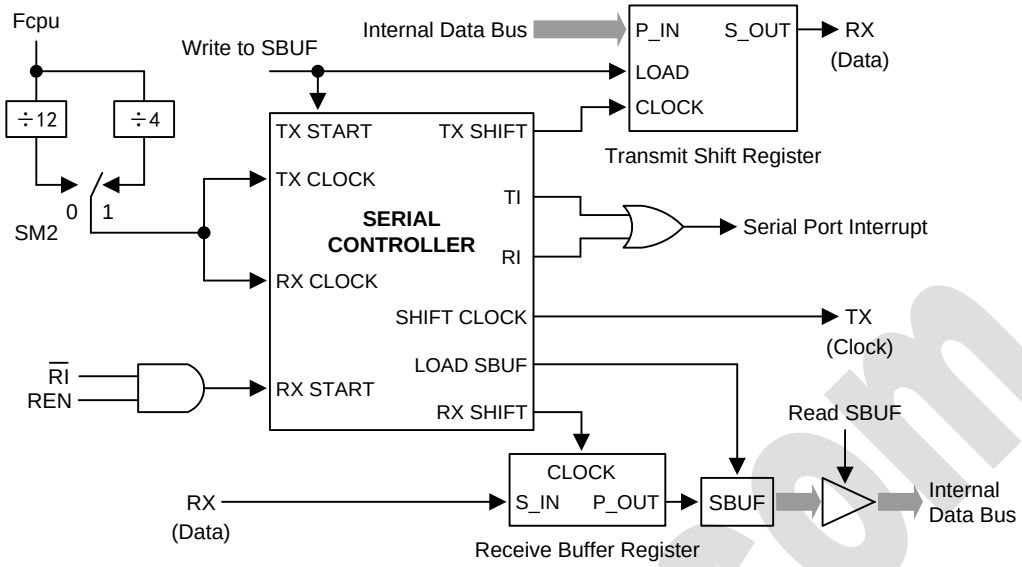
SM[0:1]	方式	类型	波特率	帧长度	帧传输格式			
					起始位	数据位	第 9 位	停止位
00	0	同步	FCPU/4 或 FCPU/12	8 位	无	8 位	无	无
01	1	异步	T2 溢出频率/16	10 位	1 位	8 位	无	1 位
10	2	异步	FCPU/32 或 FCPU/64	11 位	1 位	8 位	有(数据/标志)	1 位
11	3	异步	T2 溢出频率/16	11 位	1 位	8 位	有(数据/标志)	1 位

方式 0：8 位同步半双工

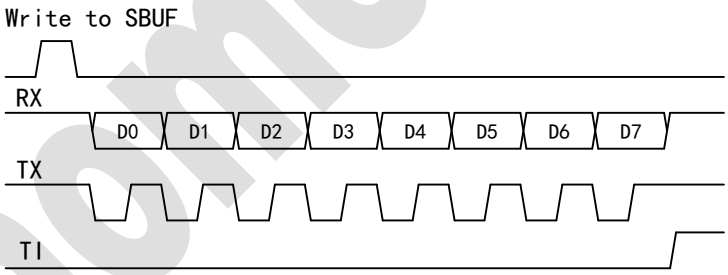
方式 0 支持与外部设备的同步通讯。UART 通过 TX 端口发送移位时钟，在 RX 端口上接收/发送串行数据。方式 0 为半双工的串行通讯，每一帧包含 8 位数据，低位先接收/发送。

方式 0 的波特率可通过寄存器位 SM2 选择为 CPU 时钟 FCPU 的 4 分频或 12 分频。

方式 0 功能模块如下图所示：

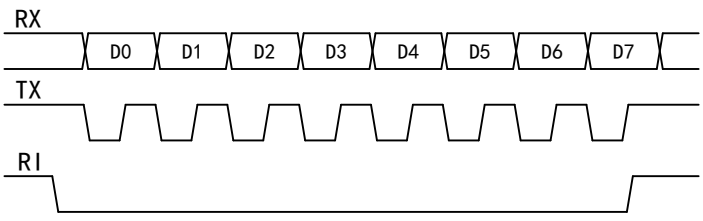


方式 0 中 UART 模块通过 TX 端口输出同步时钟，通过 RX 端口将数据读入或移出串行端口。任何将 SBUF 作为目的寄存器的写操作均将启动 UART 发送。下一个时钟 TX 控制模块开始发送，在移位时钟的下降沿进行数据变化，移位寄存器的内容逐次从左往右移出，空位则置 0。当移位寄存器中所有 8 位数据全部发送完后，TX 控制模块停止发送操作，并在下一个时钟的上升沿将中断标志 TI 置 1。



Send Timing of Mode 0

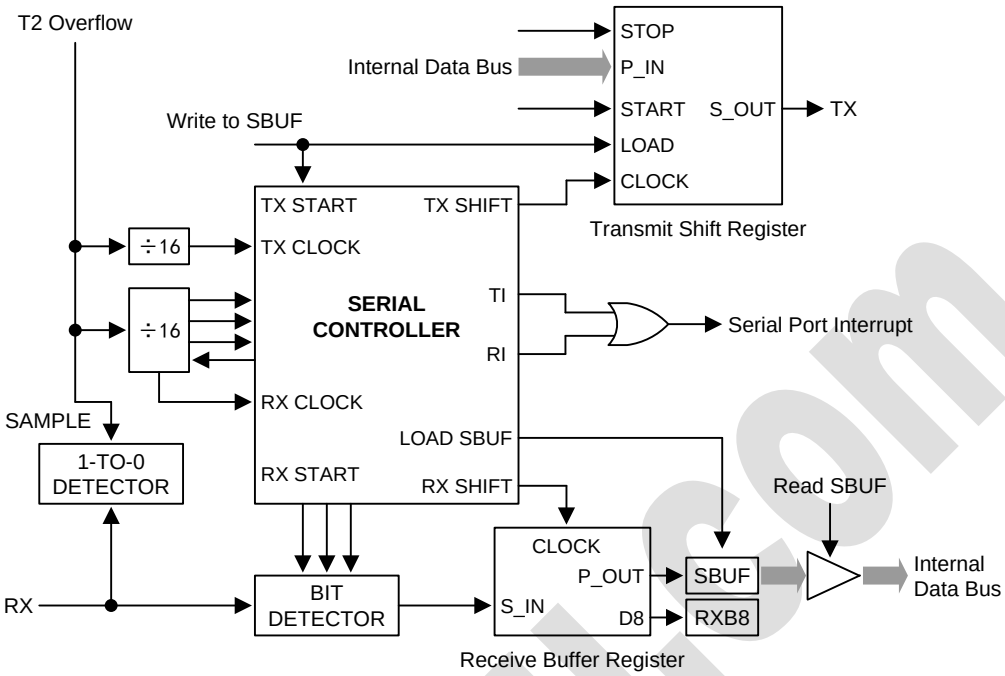
方式 0 中 REN 置 1 后 RI 清 0 将初始化 UART 接收。下一个时钟启动 UART 接收，在移位时钟的上升沿读入并锁存数据，接收缓存器的内容逐次向左移位。当所有 8 位数据全部移入接收缓存器后，RX 控制模块停止接收，并在下一个时钟的上升沿将中断标志 RI 置 1，直到被软件清 0 后才允许再次接收。



Recieve Timing of Mode 0

方式 1：8 位异步全双工，可变波特率

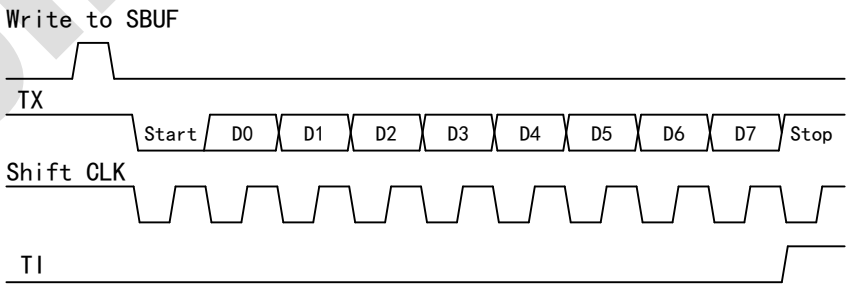
方式 1 功能模块如下图所示：



方式 1 支持 10 位全双工的异步通讯，10 位由 1 位起始位（逻辑 0）、8 位数据位（低位在先）和 1 位停止位（逻辑 1）组成。接收时，8 位数据位缓存于 SBUF 而停止位则移入 RXB8。

方式 1 的波特率是可变的，为定时器 T2 溢出频率的 1/16。

方式 1 中任何将 SBUF 作为目的寄存器的写操作均将启动 UART 发送。发送实际是从系统时钟 16 分频计数器的下一次跳变之后的系统时钟开始的，因此每一位的时间与 16 分频计数器是同步的，而与 SBUF 写操作是不同步的。发送时 TX 控制模块在 TX 端口上首先移出起始位，然后是 8 位数据位，在移位寄存器中所有 8 位数据全部发送完后，再将停止位移出，并在停止位移出的同时将中断标志 TI 置 1。



Send Timing of Mode 1

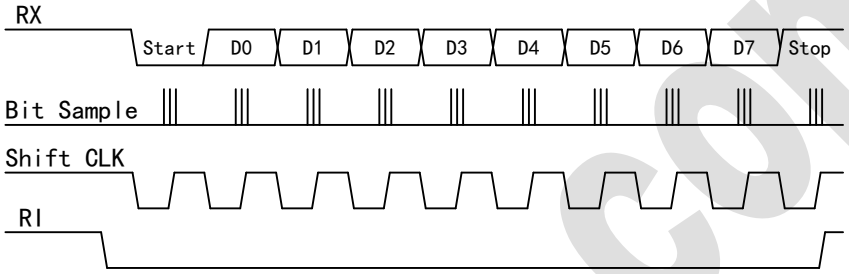
方式 1 中 REN 置 1 后 UART 才允许接收。当 RX 端口检测到下降沿时串行接口开始接收数据，因此 CPU 将持续对 RX 端口进行采样，采样速率为 UART 波特率的 16 倍，当检测到下降沿时，16 分频计

时器立即复位，从而与 RX 端口上的串行数据位保持同步。16 分频计数器将每一位的时间分为 16 个时间段，分别在第 7、8、9 个时间段对 RX 端口电平进行一次采样，为抑制噪声，这 3 次采样中至少有 2 次采样值一致，数据才被接收。若所接收的第 1 位不是逻辑 0，则表明该位不是一帧数据的起始位，该位将被忽略，接收电路复位，等待 RX 端口上的下一个下降沿。若检测到有效起始位，则移入接收缓存器，并继续接收后续数据位移入接收缓存器。当所有 8 位数据位和 1 位停止位移入后，若同时满足下列条件，则接收缓存器的 8 位数据位被载入 SBUF、1 位停止位被载入 RXB8，同时中断标志 RI 将被置 1：

1、RI=0；

2、SM2=0，或接收的停止位为逻辑 1；

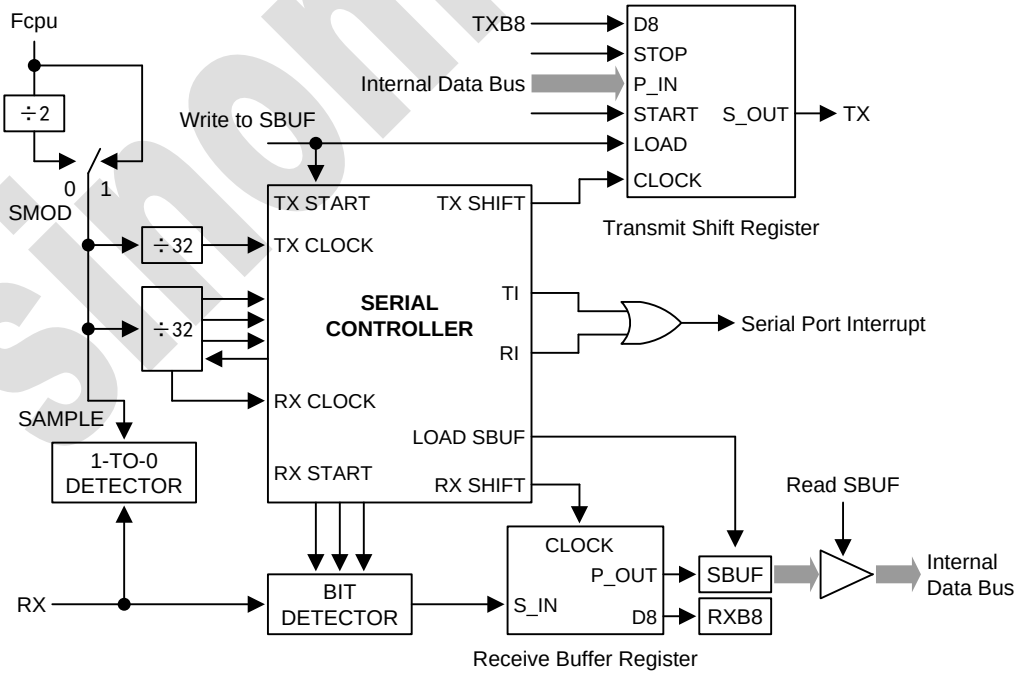
若上述条件不满足，则接收到的帧将被舍弃，RX 控制模块将重新检测 RX 端口的下一个下降沿。需软件清除中断标志 RI 后，才允许再次接收。



Recievd Timing of Mode 1

方式 2：9 位异步全双工，固定波特率

方式 2 功能模块如下图所示：

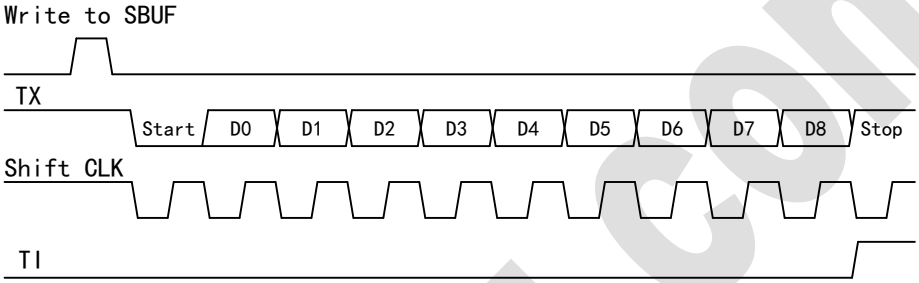


方式 2 支持 11 位全双工的异步通讯，11 位由 1 位起始位（逻辑 0）、8 位数据位（低位在先）和 1 位可编程的第 9 位数据位、以及 1 位停止位（逻辑 1）组成。方式 2 支持多机通讯和硬件地址识别（详

见多机通讯章节）。发送数据时，第 9 位（SCON 中的 TXB8）可编程为 0 或 1，从而可用作奇偶校验位，或用作多机通讯中区分地址帧/数据帧的标志位。接收数据时，第 9 位移入 RXB8 而停止位被舍弃。

方式 2 的波特率可通过寄存器位 SMOD 选择为 CPU 时钟 F_{CPU} 的 32 分频或 64 分频。

方式 2 中任何将 SBUF 作为目的寄存器的写操作均将启动 UART 发送，同时也将 TXB8 载入移位寄存器中的第 9 位。发送实际是从系统时钟 16 分频计数器的下一次跳变之后的系统时钟开始的，因此每一位的时间与 16 分频计数器是同步的，而与 SBUF 写操作是不同步的。发送时 TX 控制模块在 TX 端口上首先移出起始位，然后是 9 位数据位，在移位寄存器中所有 9 位数据全部发送完后，再将停止位移出，并在停止位移出的同时将中断标志 TI 置 1。



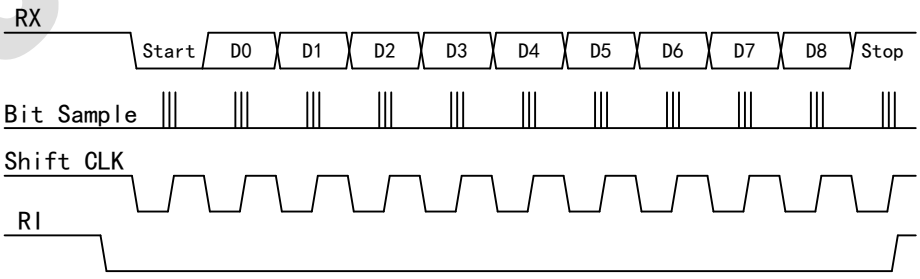
Send Timing of Mode 2

方式 2 中 REN 置 1 后 UART 才允许接收。当 RX 端口检测到下降沿时串行接口开始接收数据，因此 CPU 将持续对 RX 端口进行采样，采样速率为 UART 波特率的 16 倍，当检测到下降沿时，16 分频计时器立即复位，从而与 RX 端口上的串行数据位保持同步。16 分频计数器将每一位的时间分为 16 个时间段，分别在第 7、8、9 个时间段对 RX 端口电平进行一次采样，为抑制噪声，这 3 次采样中至少有 2 次采样值一致，数据才被接收。若所接收的第 1 位不是逻辑 0，则表明该位不是一帧数据的起始位，该位将被忽略，接收电路复位，等待 RX 端口上的下一个下降沿。若检测到有效起始位，则移入接收缓存器，并继续接收后续数据位移入接收缓存器。当所有 9 位数据位和 1 位停止位移入后，若同时满足下列条件，则接收缓存器的 9 位数据位被载入 SBUF 和 RXB8（载入第 9 位），同时中断标志 RI 将被置 1：

1、RI=0；

2、SM2=0，或接收的第 9 位为逻辑 1 且接收的字节匹配本机预设地址；

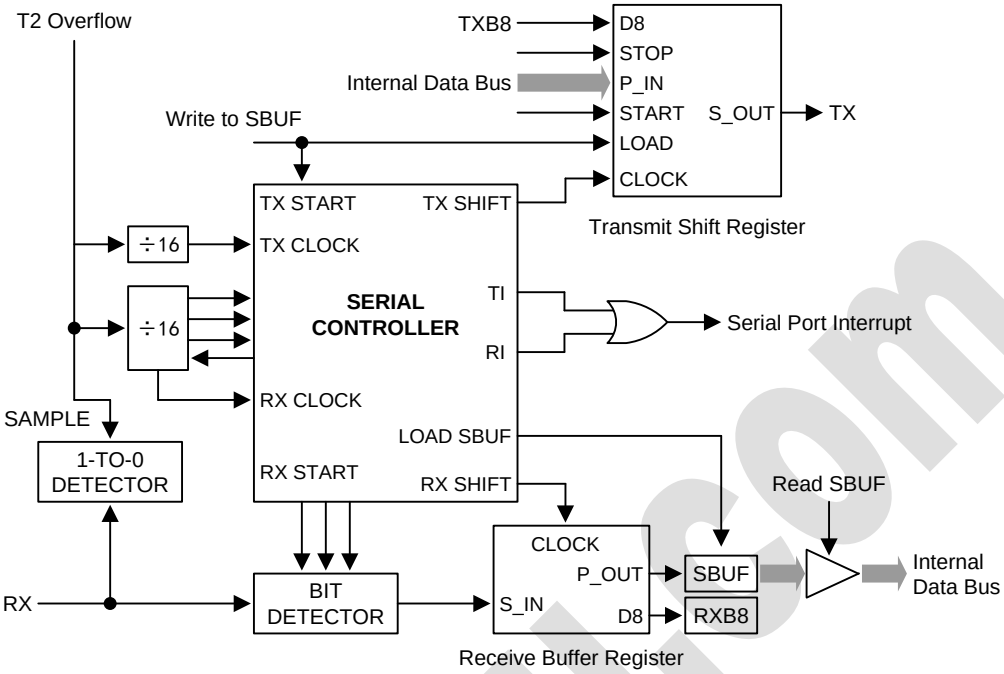
若上述条件不满足，则接收到的数据将被舍弃，RX 控制模块将重新检测 RX 端口的下一个下降沿。需软件清除中断标志 RI 后，才允许再次接收。



Recievd Timing of Mode 2

方式 3：9 位异步全双工，可变波特率

方式 3 功能模块如下图所示：



方式 3 的传输协议与方式 2 相同，波特率产生方法则与方式 1 相同。

10.3 UART 波特率

方式 0 中，波特率可通过寄存器位 SM2 选择为系统时钟的 12 分频或 4 分频。SM2=0 时，串行端口在系统时钟的 12 分频时钟下运行；SM2=1 时，串行端口在系统时钟的 4 分频时钟下运行。

方式 1 和方式 3 中，波特率为定时器 T2 的溢出频率，计算如下：

$$\text{BaudRate} = \frac{1}{16} \times \frac{F_{T2}}{\text{T2LOAD} + 1}$$

其中 F_{T2} 为 T2 的计数时钟频率。

注：T2 用于 UART 时 T2LOAD 不能为 0，否则将无法正确产生波特率。

方式 2 中，波特率可通过寄存器位 SMOD 选择为系统时钟的 32 分频或 64 分频。SMOD=0 时，串行端口在系统时钟的 64 分频时钟下运行；SMOD=1 时，串行端口在系统时钟的 32 分频时钟下运行。

10.4 UART 多机通讯

UART 的方式 2 和方式 3，支持从机地址自动识别，可应用于多机通讯系统。

在多机通讯系统中，每一帧传输内容的第 9 位用作地址/数据帧标志（地址帧第 9 位为 1，数据帧第 9 位为 0），当寄存器位 SM2 为 1 时，UART 在接收到一帧数据后会先识别第 9 位，若为 0 则不响应，若为 1 则再检验接收到的地址是否与本机预设地址匹配。只有接收到地址帧且地址匹配，UART 才将接收到的数据保存并将中断标志 RI 置 1 触发中断；否则 RI 不会被置 1，接收的数据也将被舍弃。

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。

从机需先将 SM2 置 1 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。

从机地址自动识别

方式 2 和方式 3 支持 9 位数据，发送时第 9 位可编程，接收时第 9 位移入 RXB8。UART 接收时可通过 SM2 置 1 使能对 RXB8 以及从机地址的自动识别功能，此时 UART 仅在接收的 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，SM2 需预先置 1。中断触发后，地址匹配的从机需将 SM2 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 SM2 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。

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 SADDR 和 SADEN 预设本机地址。从机地址为 8 位，预设于地址寄存器 SADDR 中，地址掩码寄存器 SADEN 的每一位则决定 SADDR 中对应的寄存器位在检测地址时是否参与检验，若 SADEN 中某一位为 0，则 SADDR 中相应位将被忽略（不参与地址检验，默认为匹配），若 SADEN 中某一位为 1，则 SADDR 中相应位将参与地址检验，这将使从机可在不改变 SADDR 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 SADDR 和 SADEN 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。

	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 (SADEN 中某一位为 0, 则默认 SADDR 中相应位匹配)	1010 0000	1010 0000	1010 0001
	1010 0001	1010 0001	1010 0011
	1010 0010	1010 0100	1010 0101
	1010 0011	1010 0101	1010 0111
从机响应的广播地址列表 (SADDR 和 SADEN 逻辑“或”的值)		1111 1110	
	1111 1111	1111 1111	1111 1111

系统复位后，SADDR 和 SADEN 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。

10.5 UART 出错检测

当寄存器 T2CR 中的 SSTAT 为 1 时，UART 出错检测功能才有效。错误标志位被置 1 后，仅能通过软件清 0，即使后续接收的帧没有任何错误也不会自动清 0。

注：访问寄存器位 SCON[7:5]，SSTAT=0 时是访问 UART 控制位(SM0, SM1, SM2)，SSTAT=1 时是访问 UART 标志位(FERR, RXOV)。

接收溢出

当 RI 被清 0 而接收缓存器中数据尚未被读取时，若又有新的数据存入接收缓存器，则接收溢出标志 RXOV 将被置 1。若发生接收溢出，则接收缓存器中原有数据将丢失。

帧出错

如果检测到一个无效（逻辑 0）停止位，则帧出错标志 FERR 将被置 1。

暂停检测

当连续检测到 11 个传输位均为逻辑 0 时，则认为检测到一个暂停。由于暂停条件同样满足帧出错条件，因此检测到暂停时也会报帧出错。一旦检测到暂停，UART 将进入空闲状态并一直保持，直到接收到有效停止位（RX 端口从逻辑 0 变为逻辑 1）。

10.6 UART 相关寄存器

串行通讯接口控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
SCON	SM0/FERR	SM1/RXOV	SM2	REN	TXB8	RXB8	TI	RI
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[7:6] **SM[0:1]** – UART 工作方式选择位（SSTAT=0）

SM[0:1]	UART 工作方式
00	方式 0：8 位同步方式，固定波特率
01	方式 1：8 位异步方式，可变波特率
10	方式 2：9 位异步方式，固定波特率
11	方式 3：9 位异步方式，可变波特率

BIT[5] **SM2** – UART 功能控制位（SSTAT=0）

SM2	方式 0	方式 1	方式 2/3
0	波特率=$F_{CPU}/12$	不检验停止位，一旦接收到停止位就将 RI 置 1 触发中断	接收到任意帧，不识别第 9 位，均将 RI 置 1 触发中断
1	波特率=$F_{CPU}/4$	检验停止位，接收到有效的停止位才将 RI 置 1 触发中断	仅接收到第 9 位为 1 的地址帧且地址匹配，才将 RI 置 1 触发中断

BIT[7] **FERR** – UART 帧出错标志位（SSTAT=1）

0： 无帧出错；

1： 发生帧出错，硬件自动置 1，需软件清 0；

BIT[6] **RXOV** – UART 接收溢出标志位（SSTAT=1）

0： 接收无溢出；

1： 接收有溢出，硬件自动置 1，需软件清 0；

BIT[4] **REN** – UART 接收使能位

0： 禁止 UART 接收；

1： 允许 UART 接收；

BIT[3] **TXB8** – UART 工作方式 2/3 中发送数据的第 9 位

0： UART 工作方式 2/3 中，发送数据的第 9 位为 0；

1： UART 工作方式 2/3 中，发送数据的第 9 位为 1；

BIT[2] **RXB8** – UART 工作方式 1/2/3 中接收数据的第 9 位（停止位或数据位）

0： UART 工作方式 1/2/3 中，接收数据的第 9 位为 0；

1： UART 工作方式 1/2/3 中，接收数据的第 9 位为 1；

BIT[1] **TI** – UART 发送中断标志位

0： 由软件清 0；

1： 方式 0 中第 8 位发送结束时，或方式 1/2/3 中停止位发送开始时，由硬件置 1；

BIT[0] **RI** – UART 接收中断标志位

0： 由软件清 0；

1： 方式 0 中第 8 位接收结束时，或方式 1/2/3 中停止位接收开始时，由硬件置 1；

串行通讯接口缓冲器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
SBUF	SBUF7	SBUF6	SBUF5	SBUF4	SBUF3	SBUF2	SBUF1	SBUF0
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] **SBUF[7:0]** – UART 数据缓冲器，写 SBUF 为写发送移位寄存器，读 SBUF 为读接收缓存器

注：对 SBUF 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。

从机地址寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
SADDR	SADDR7	SADDR6	SADDR5	SADDR4	SADDR3	SADDR2	SADDR1	SADDR0
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] **SADDR[7:0]** – UART 地址寄存器，用于设置 UART 通讯方式 2/3 中的从机地址

地址使能寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
SADEN	SADEN7	SADEN6	SADEN5	SADEN4	SADEN3	SADEN2	SADEN1	SADEN0
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] **SADENn** – 地址位 SADDRn 检验控制位（n=7-0）

0： 不检验地址位 SADDRn，默认该位地址匹配；

1： 检验地址位 SADDRn 是否匹配；

端口映射寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PMAP	–	–	–	–	–	–	–	URTCHS
R/W	–	–	–	–	–	–	–	R/W
初始值	–	–	–	–	–	–	–	0

BIT[0] **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；

11 EEPROM 存储器

11.1 EEPROM 概述

芯片内置 256×16 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。

读写操作控制位 EETRIG 置 1 将启动 EEPROM 的读写操作，完成后自动清 0；而读写操作选择位 EERW 为 0 表示读数据操作，为 1 表示写数据操作。读操作启动后，EEAR 所指 EEPROM 地址中的 16 位数据将被读出并缓存于 EEDR；写操作启动后，EEDR 中的 16 位数据将被写入 EEAR 指向的 EEPROM 地址中。

芯片仅支持 EEPROM 单地址的数据读写，不支持连续地址的自动读写功能，每次均需通过 EEAR 设置将要访问的 EEPROM 地址后，才能读写该地址指向的 EERPOM 数据。

为防止误触发 EEPROM 读写操作，需先对 EEPR 写 5AH 再立即写 A5H，并在随后 2 个指令周期内将 EETRIG 置 1，才能启动读写操作，中间不能插入其他操作（包括 NOP 操作）。否则 2 个指令周期后 EEPR 将自动清零，此时 EETRIG 置 1 不会启动读写操作，也不会自动清 0。

在启动 EEPROM 读写操作后，CPU 将暂停在当前指令，只有等 EERPOM 读写操作完成后，才继续执行下一条指令。在读/写 EEPROM 前需先屏蔽中断并清 WDT 计数器，否则可能会因系统响应中断或复位而导致无法正常启动或执行 EEPROM 的读写操作。

注：

1、 EEPROM 的操作时钟为 CPU 时钟 FCPU，EEPROM 读/写操作时，CPU 将暂停工作；

2、 若 FCPU 较低（＜500KHz），将导致 EEPROM 读写故障，此时应禁止读写 EEPROM；

3、 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；

11.2 EEPROM 相关寄存器

EEPROM 控制寄存器

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

BIT[7] **EETRIG** – EEPROM 读写操作控制位

0： 操作未开始或已完成，写 1 开始 EEPROM 读写操作；

1： EEPROM 读写操作中，完成后自动清 0；

BIT[6] **EERW** – EEPROM 读/写操作选择位

0： 读操作，从 EEPROM 中读出数据；

1： 写操作，向 EEPROM 中写入数据；

EEPROM 保护寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
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 操作保护控制位，需先写 5AH 再立即写 A5H，EETRIG 才能置 1

EEPROM 地址寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
EEAR	EEA7	EEA6	EEA5	EEA4	EEA3	EEA2	EEA1	EEA0
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] **EEA[7:0]** – EEPROM 读写操作的 8 位地址

EEPROM 数据寄存器

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

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

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

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

11.3 EEPROM 操作示例

例如，将数据 55AAH 写入 EEPROM 存储器 10H 中：

```
BSET    HFEN        ；确保系统高频时钟开启
MOVAI   00H
MOVRA   EECR        ；清 EETRIG
```

```
MOVAI       10H
MOVRA       EEAR            ；将 10H 写入 EEAR
MOVAI       55H
MOVRA       EEDRH           ；将 55H 写入 EEDRH
MOVAI       AAH
MOVRA       EEDRL           ；将 55H 写入 EEDRL
BCLR        GIE             ；屏蔽中断
CLRWDT                      ；清 WDT
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       C0H
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
```

例如，读取 EEPROM 存储器 11H 地址中内容：

```
BSET        HFEN            ；确保系统高频时钟开启
MOVAI       00H
MOVRA       EECR            ；清 EETRIG
MOVAI       11H
MOVRA       EEAR            ；将 11H 写入 EEAR
BCLR        GIE             ；屏蔽中断
CLRWDT                      ；清 WDT
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       80H
MOVRA       EECR            ；启动 EE 写操作，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
BSET        GIE             ；允许中断
MOVAR       EEDRH/EEDRL     ；从 EEDRH/EEDRL 中读取数据
```

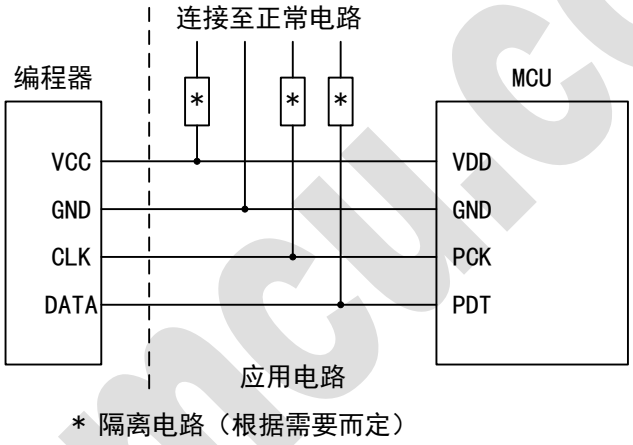
12 FLASH 烧录编程

12.1 FLASH 在板编程

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

芯片也支持对 EEPROM 型数据存储器的在板编程。

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



芯片也支持在板带电烧录编程，即可在系统电路板不掉电（芯片已正常工作）的状态下对存储器编程。当配置字 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；

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。

DEBUG 保护寄存器

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

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

为防止误触发对 DBGCR 的写操作，需先对寄存器 DBGPR 写 3CH 再立即写 C3H，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（P25）/PDT0（P24）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 P25/P24 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              ; 清 WDT
GOTO    $-2         ; 死循环，等待编程器连接芯片进行带电烧录编程
```

13 中断

芯片的中断源包括外部中断（INT0~INT3）、定时器中断（T0~T3）、ADC 中断、键盘中断、IIC 中断和 UART 中断等。可通过中断总使能位 GIE 屏蔽所有中断。

CPU 响应中断的过程如下：

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

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。

13.1 外部中断

芯片具有 4 路外部中断源 INTn（n=0-3），INT0/INT1 可选上升沿、下降沿或电平变化等触发方式，INT2/INT3 固定为下降沿触发。外部中断触发时，中断标志 INTnIF（n=0-3）将被置 1，若 GIE 为 1 且相应的外部中断使能位 INTnIE（n=0-3）为 1，则产生外部中断。

13.2 定时器中断

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

13.3 键盘中断

芯片具有 8 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。

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	P17KE	P16KE	P15KE	P14KE	P13KE	P12KE	P11KE	P10KE
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] **P1nKE** – P1n 端口键盘中断功能使能位（n=7-0）

0： 关闭端口的键盘中断功能；

1： 使能端口的键盘中断功能；

13.4 ADC 中断

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

13.5 IIC 中断

IIC 主机模式通讯中，可通过寄存器位 IICIM 选择不同的中断触发方式：

若 IICIM=0，则当发送完数据再接收完从机应答信号、或接收完从机数据再发送完应答信号、或发送完 STOP 信号等事件发生时，将触发 IIC 中断，中断标志 IICIF 将被置 1，若 GIE 为 1 且 IIC 中断使能位 IICIE 为 1，则产生 IIC 中断。

若 IICIM=1，则当发送完 START 或 STOP 信号、或发送完数据再接收完从机应答信号、或接收完从机数据、或发送完应答信号等事件发生时，将触发 IIC 中断，中断标志 IICIF 将被置 1，若 GIE 为 1 且 IIC 中断使能位 IICIE 为 1，则产生 IIC 中断。

13.6 UART 中断

UART 通讯中，中断标志 TI 或 RI 被硬件置 1 时将触发 UART 中断，中断标志 UARTIF 将被置 1，若 GIE 为 1 且 UART 中断使能位 UARTIE 为 1，则产生 UART 中断。

13.7 中断相关寄存器

中断使能寄存器

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

BIT[7] **UARTIE** – UART 中断使能位
0： 屏蔽 UART 中断；
1： 使能 UART 中断；

BIT[6] **ADIE** – ADC 中断使能位
0： 屏蔽 ADC 中断；
1： 使能 ADC 中断；

BIT[5] **IICIE** – IIC 中断使能位
0： 屏蔽 IIC 中断；
1： 使能 IIC 中断；

BIT[4] **KBIE** – 键盘中断使能位
0： 屏蔽键盘中断；
1： 使能键盘中断；

BIT[3] **INT1IE** – INT1 中断使能位
0： 屏蔽 INT1 中断；
1： 使能 INT1 中断；

BIT[2] **INT0IE** – INT0 中断使能位
0： 屏蔽 INT0 中断；
1： 使能 INT0 中断；

BIT[1] **T1IE** – 定时器 T1 中断使能位
0： 屏蔽定时器 T1 中断；
1： 使能定时器 T1 中断；

BIT[0] **T0IE** – 定时器 T0 中断使能位
0： 屏蔽定时器 T0 中断；
1： 使能定时器 T0 中断；

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTE1	–	–	–	–	INT3IE	INT2IE	T3IE	T2IE
R/W	–	–	–	–	R/W	R/W	R/W	R/W

初始值	–	–	–	–	0	0	0	0

BIT[3] **INT3IE** – INT3 中断使能位
0： 屏蔽 INT3 中断；
1： 使能 INT3 中断；

BIT[2] **INT2IE** – INT2 中断使能位
0： 屏蔽 INT2 中断；
1： 使能 INT2 中断；

BIT[1] **T3IE** – 定时器 T3 中断使能位
0： 屏蔽定时器 T3 中断；
1： 使能定时器 T3 中断；

BIT[0] **T2IE** – 定时器 T2 中断使能位
0： 屏蔽定时器 T2 中断；
1： 使能定时器 T2 中断；

中断标志寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTF0	UARTIF	ADIF	IICIF	KBIF	INT1IF	INT0IF	T1IF	T0IF
R/W	R	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] **UARTIF** – UART 中断标志位
0： 未触发 UART 中断；
1： 已触发 UART 中断（软件清 0 接收/发送中断标志 RI/TI 时，UARTIF 硬件自动清 0）；

BIT[6] **ADIF** – ADC 中断标志位
0： 未触发 ADC 中断；
1： 已触发 ADC 中断，需软件清 0；

BIT[5] **IICIF** – IIC 中断标志位
0： 未触发 IIC 中断；
1： 已触发 IIC 中断，需软件清 0；

BIT[4] **KBIF** – 键盘中断标志位
0： 未触发键盘中断；
1： 已触发键盘中断，需软件清 0；

BIT[3] **INT1IF** – INT1 中断标志位
0： 未触发 INT1 中断；
1： 已触发 INT1 中断，需软件清 0；

BIT[2] **INT0IF** – INT0 中断标志位
0： 未触发 INT0 中断；
1： 已触发 INT0 中断，需软件清 0；

BIT[1] **T1IF** – 定时器 T1 中断标志位
0： 未触发定时器 T1 中断；
1： 已触发定时器 T1 中断，需软件清 0；

BIT[0] **T0IF** – 定时器 T0 中断标志位
0： 未触发定时器 T0 中断；
1： 已触发定时器 T0 中断，需软件清 0；

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTF1	-	-	-	-	INT3IF	INT2IF	T3IF	T2IF
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3] **INT3IF** – INT3 中断标志位
0： 未触发 INT3 中断；
1： 已触发 INT3 中断，需软件清 0；

BIT[2] **INT2IF** – INT2 中断标志位
0： 未触发 INT2 中断；
1： 已触发 INT2 中断，需软件清 0；

BIT[1] **T3IF** – 定时器 T3 中断标志位
0： 未触发定时器 T3 中断；
1： 已触发定时器 T3 中断，需软件清 0；

BIT[0] **T2IF** – 定时器 T2 中断标志位
0： 未触发定时器 T2 中断；
1： 已触发定时器 T2 中断，需软件清 0；

14 特性曲线

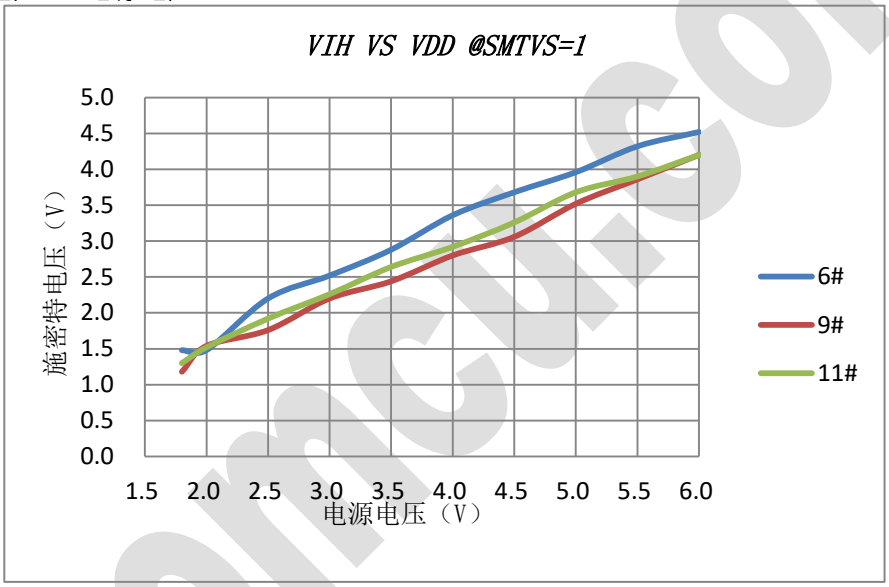
注：

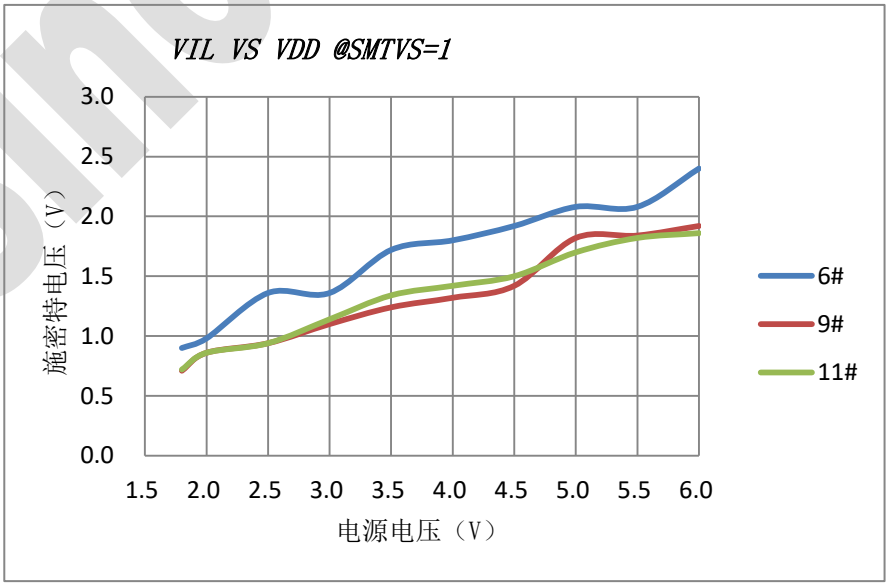
1、 特性曲线图中数据均源自抽样实测，仅作为应用参考，部分数据因生产工艺偏差，可能与实际芯片不符；为保证芯片能正常工作，请确保其工作条件符合电气特性参数说明；

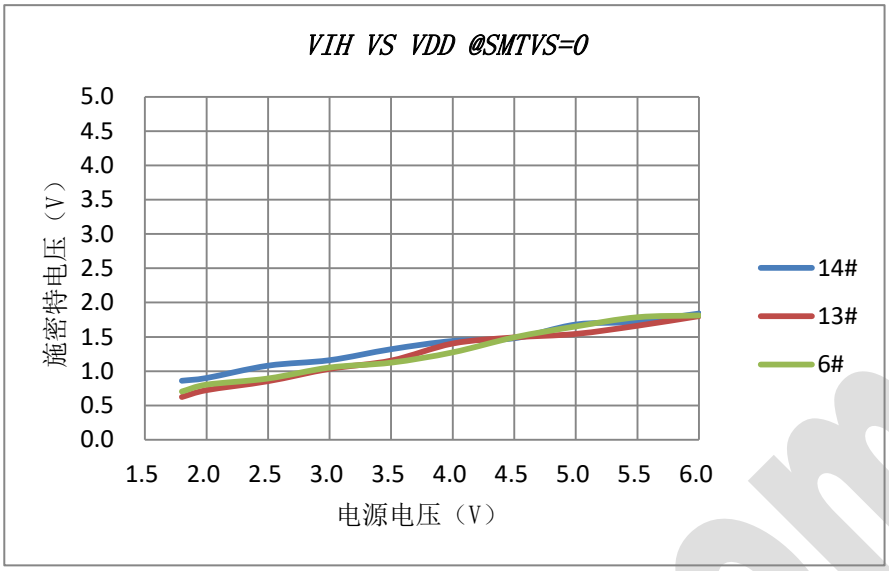
2、 图文中若无特别说明，则电压特性曲线的温度条件为 T=25℃，温度特性曲线的电压条件为 VDD=5V；

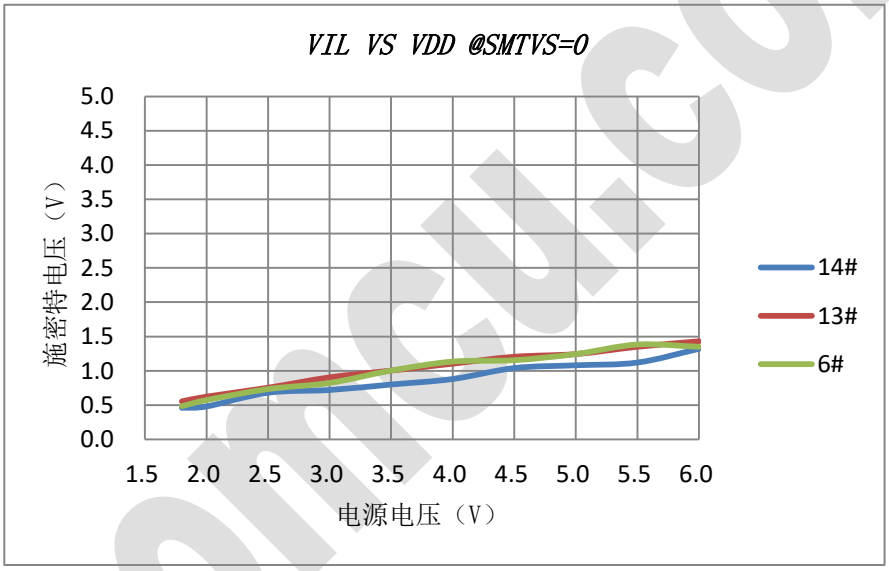
14.1 I/O 特性

输入 SMT 阈值电压 VS 电源电压

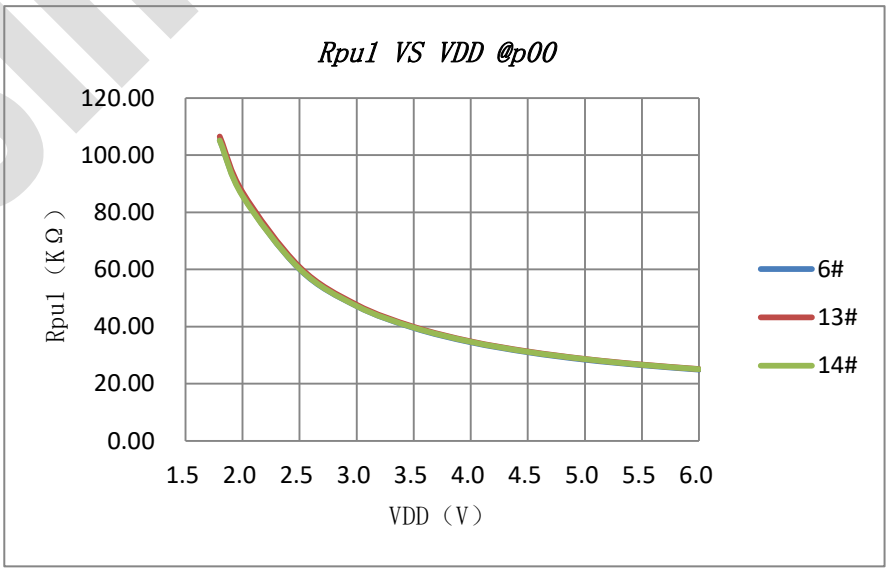


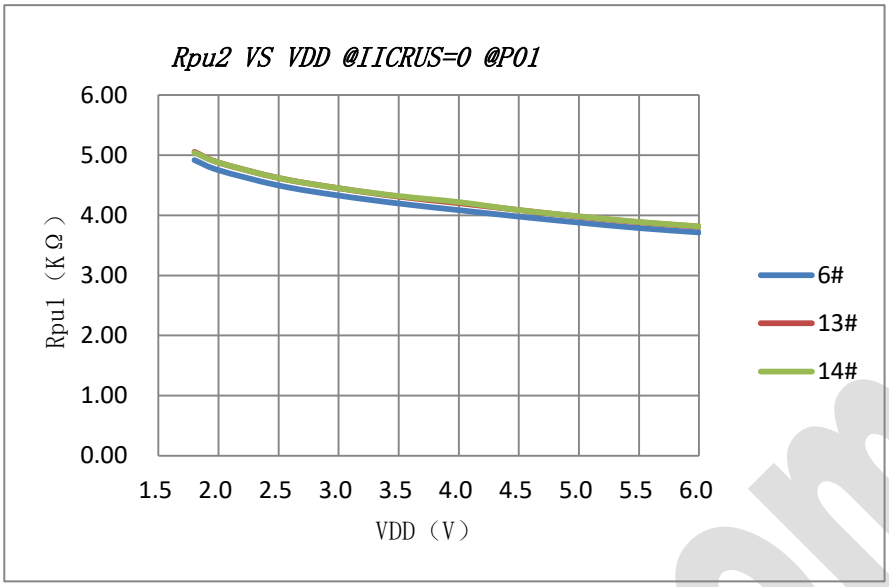


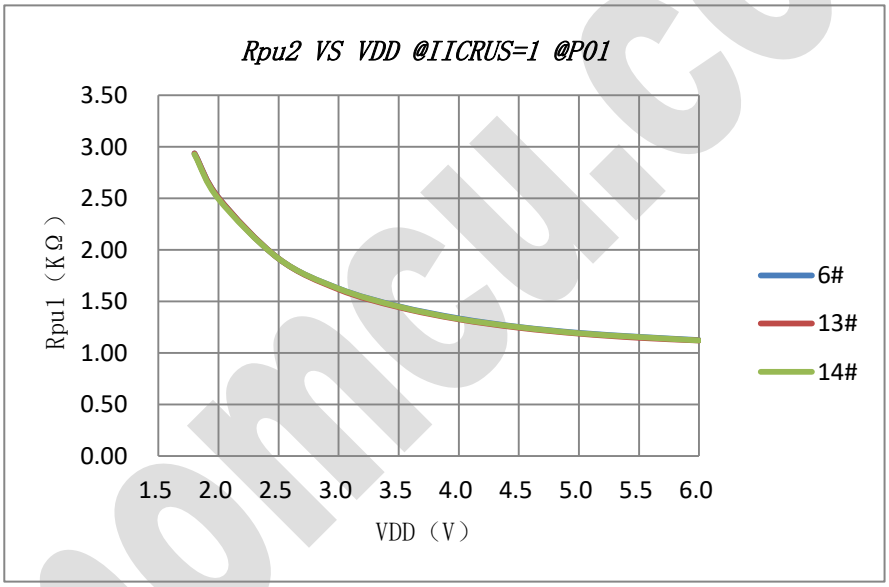


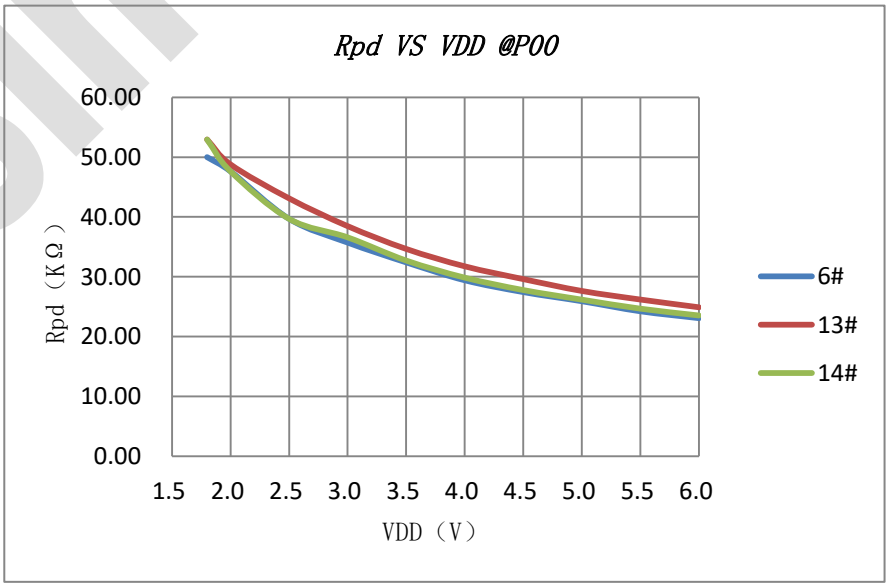


上/下拉电阻值 VS 电源电压

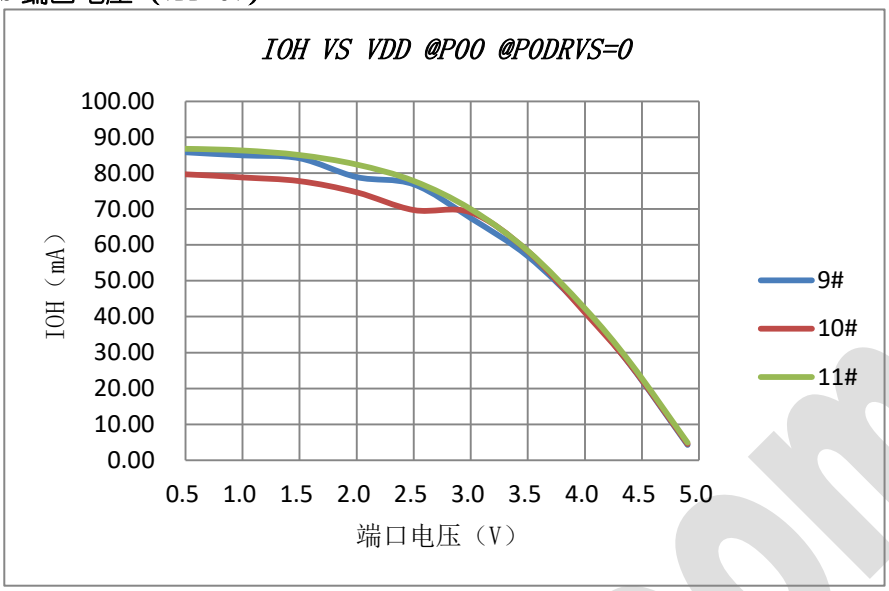


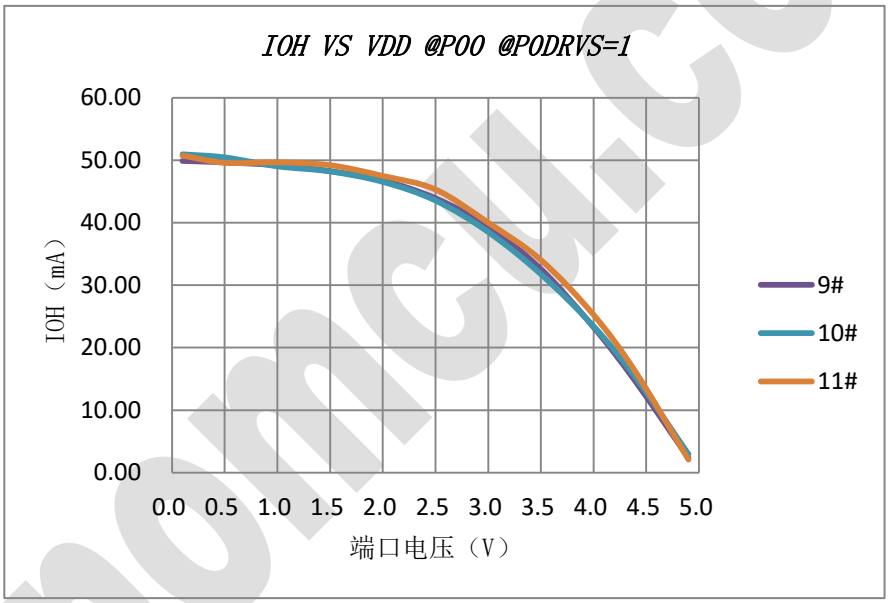

Rpu2 VS VDD @IICRUS=0 @P01
6.00
5.00
4.00
3.00
2.00
1.00
0.00
Rpu1（KΩ）
1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0
VDD（V）
6#
13#
14#

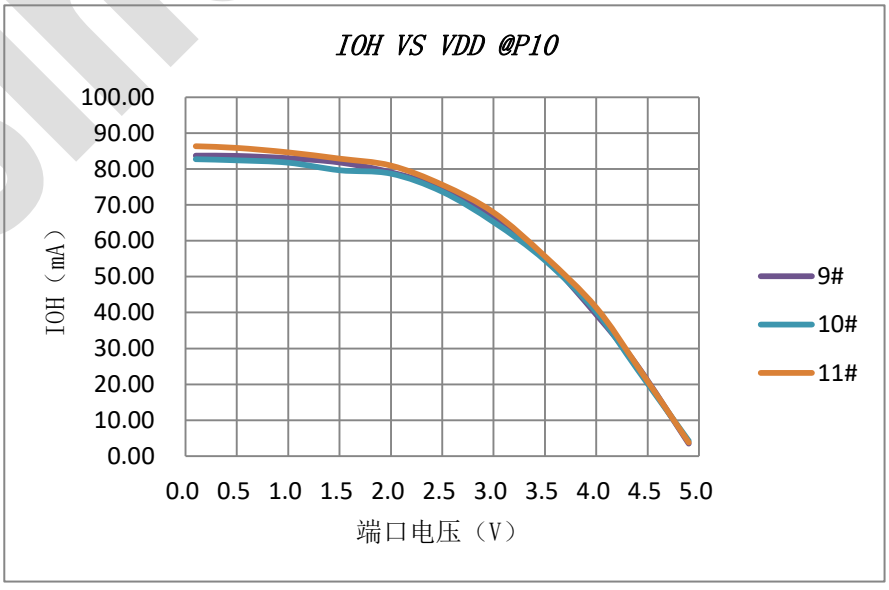

Rpu2 VS VDD @IICRUS=1 @P01
3.50
3.00
2.50
2.00
1.50
1.00
0.50
0.00
Rpu1（KΩ）
1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0
VDD（V）
6#
13#
14#

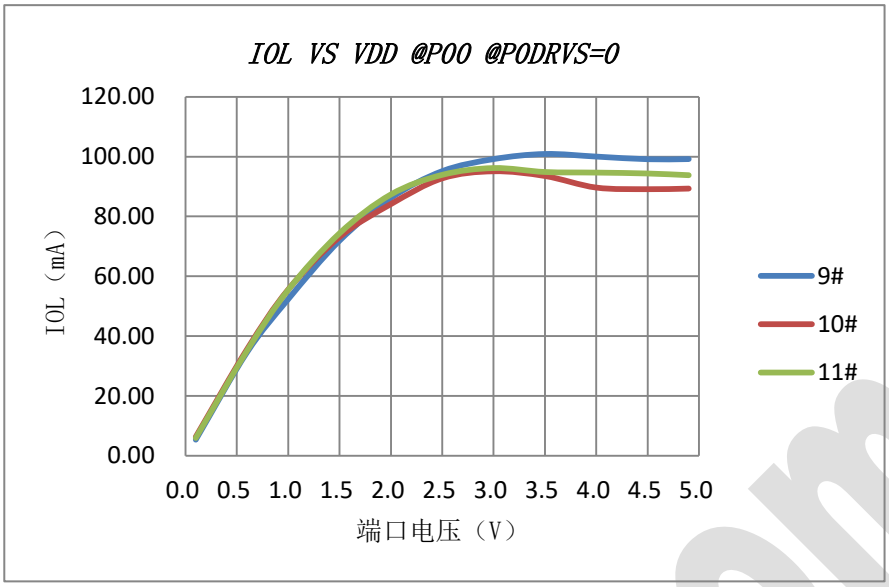

Rpd VS VDD @P00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
Rpd（KΩ）
1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0
VDD（V）
6#
13#
14#

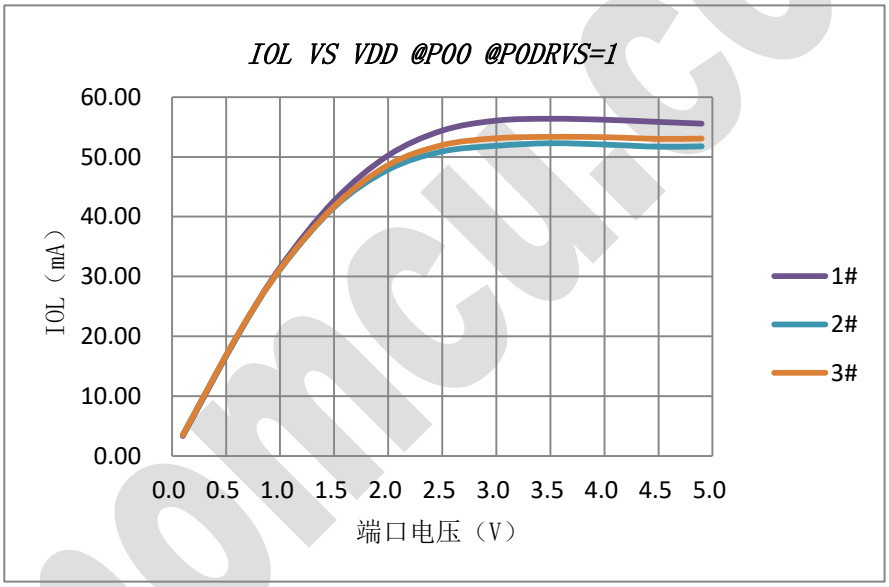
I/O 输出电流 VS 端口电压 (VDD=5V)

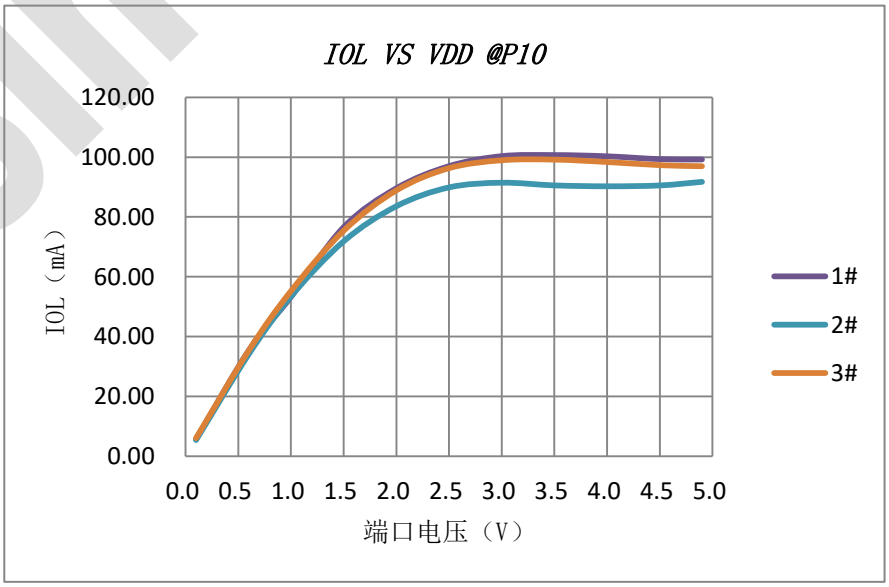






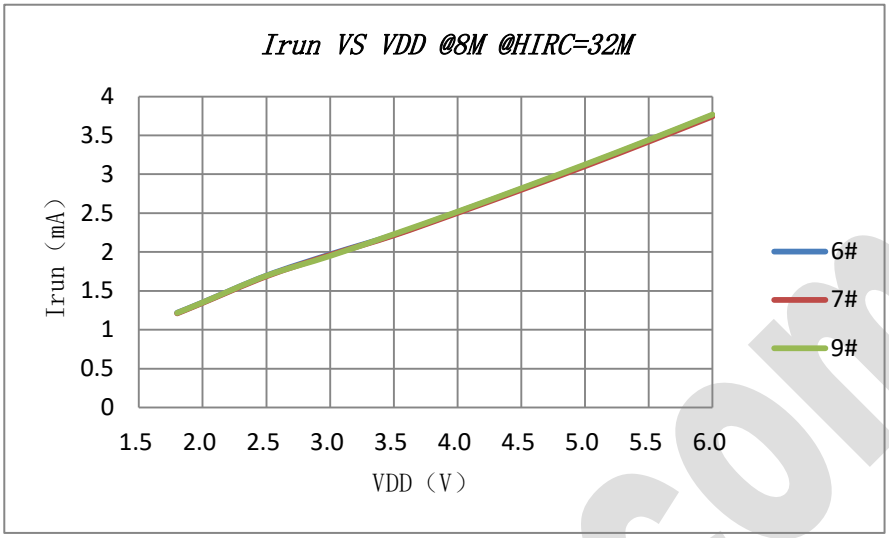

IOL VS VDD @P00 @PODRVS=0
120.00
100.00
80.00
60.00
40.00
20.00
0.00
IOL（mA）
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
端口电压（V）
9#
10#
11#

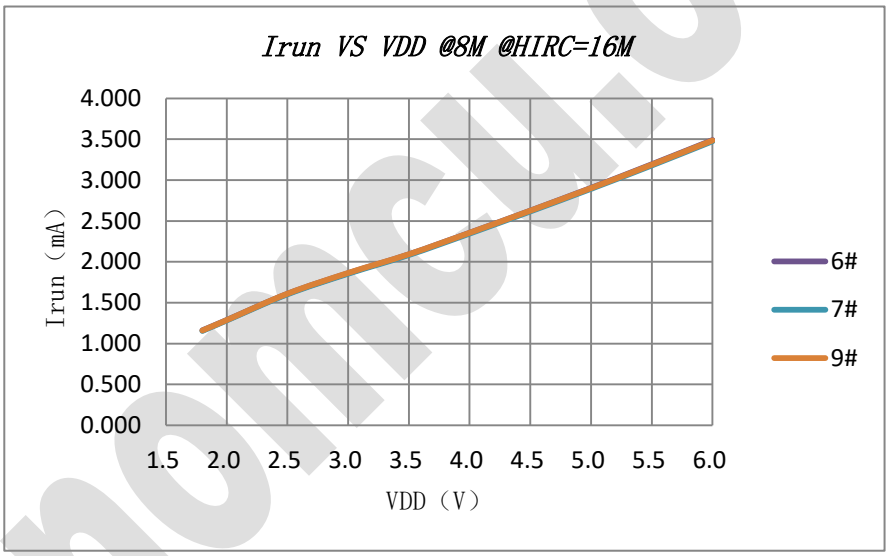

IOL VS VDD @P00 @PODRVS=1
60.00
50.00
40.00
30.00
20.00
10.00
0.00
IOL（mA）
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
端口电压（V）
1#
2#
3#

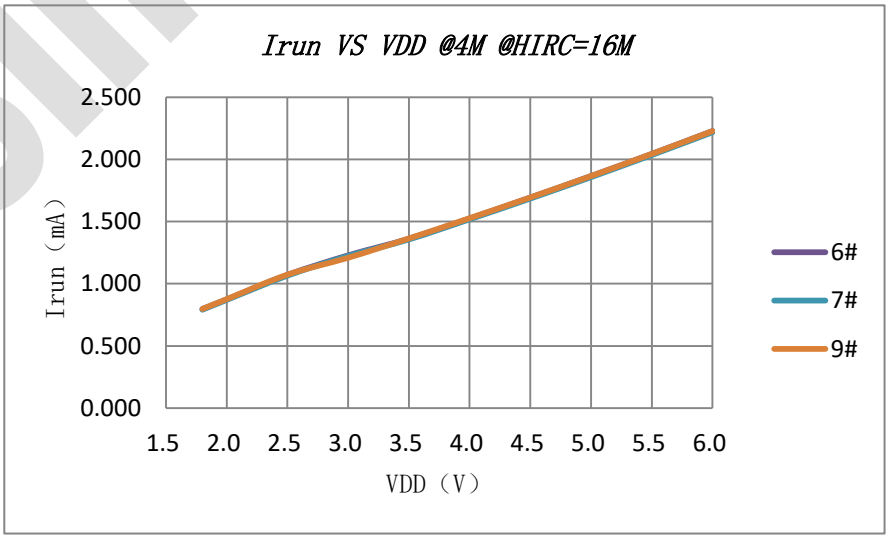

IOL VS VDD @P10
120.00
100.00
80.00
60.00
40.00
20.00
0.00
IOL（mA）
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
端口电压（V）
1#
2#
3#

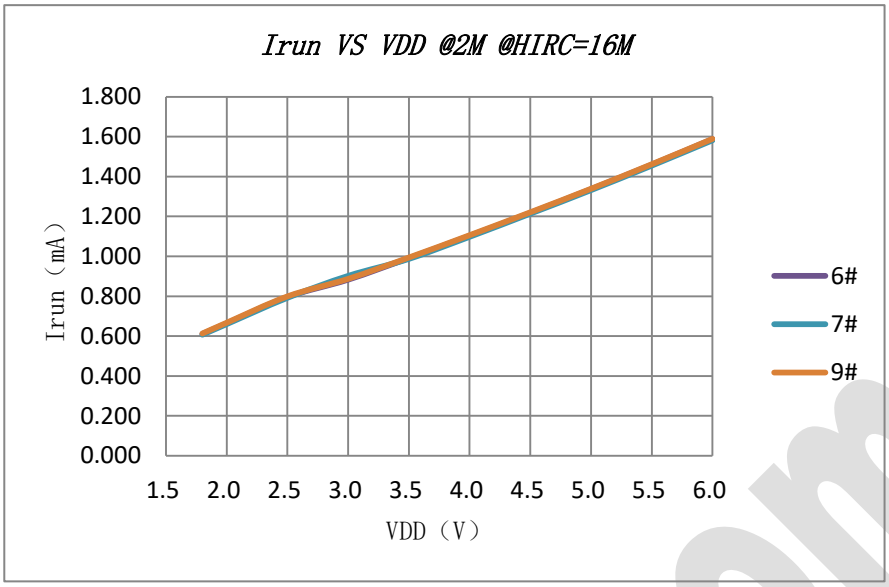
14.2 功耗特性

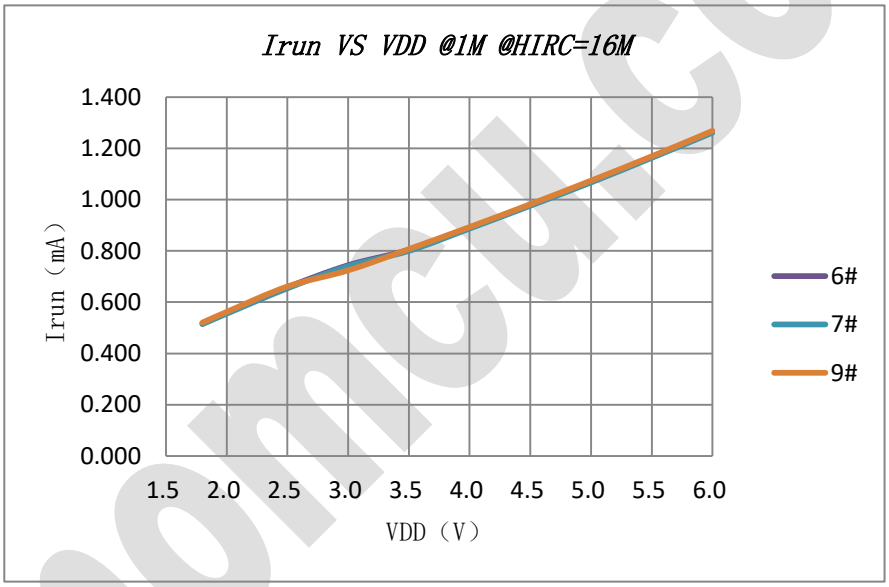
运行模式 功耗 VS 电源电压

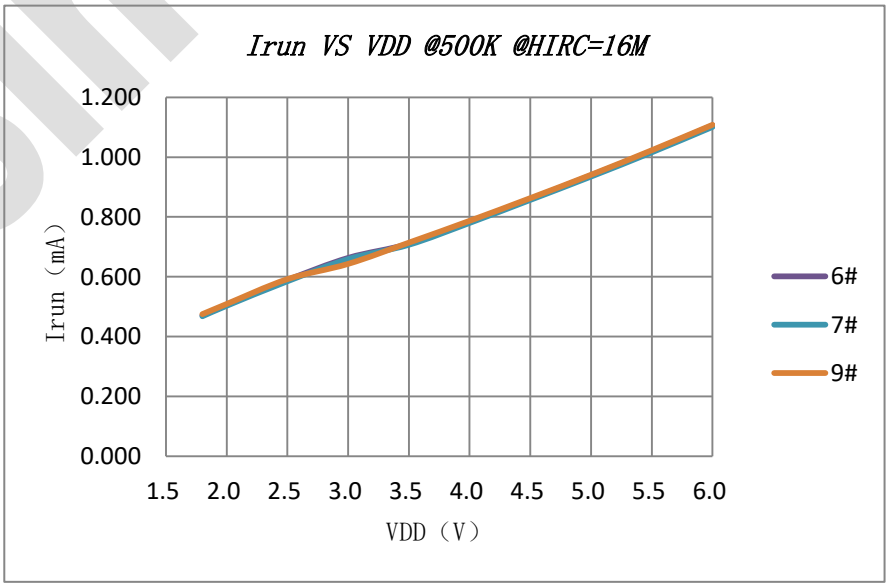


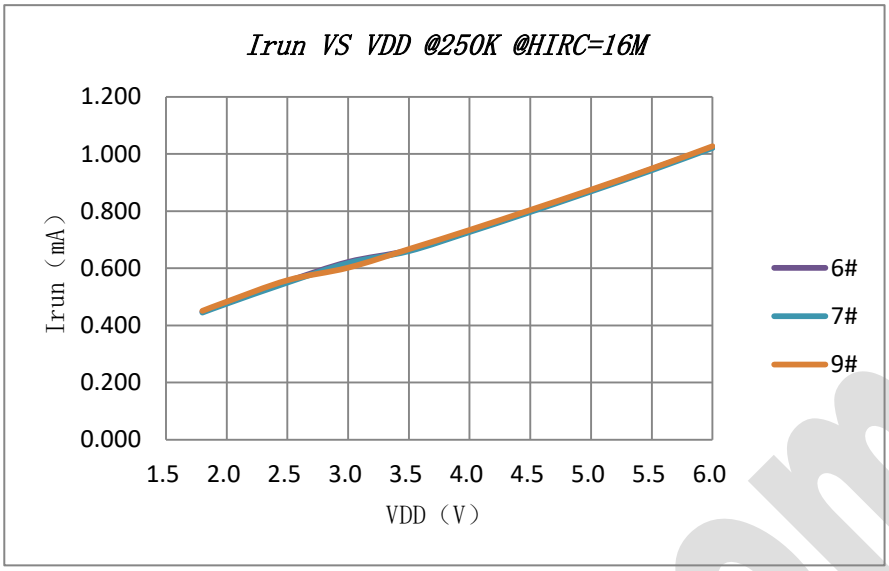


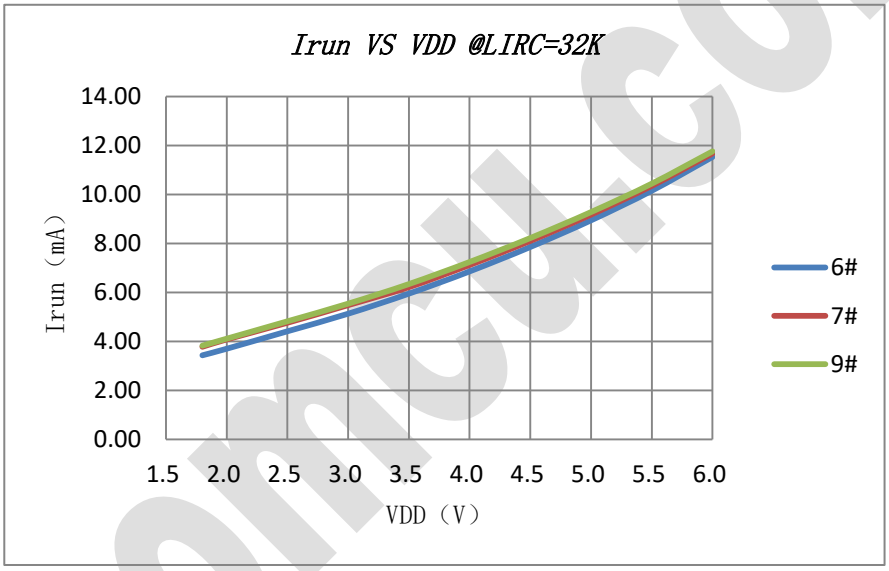



Irun VS VDD @2M @HIRC=16M
1.800
1.600
1.400
1.200
1.000
0.800
0.600
0.400
0.200
0.000
Irun（mA）
1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0
VDD（V）
6#
7#
9#

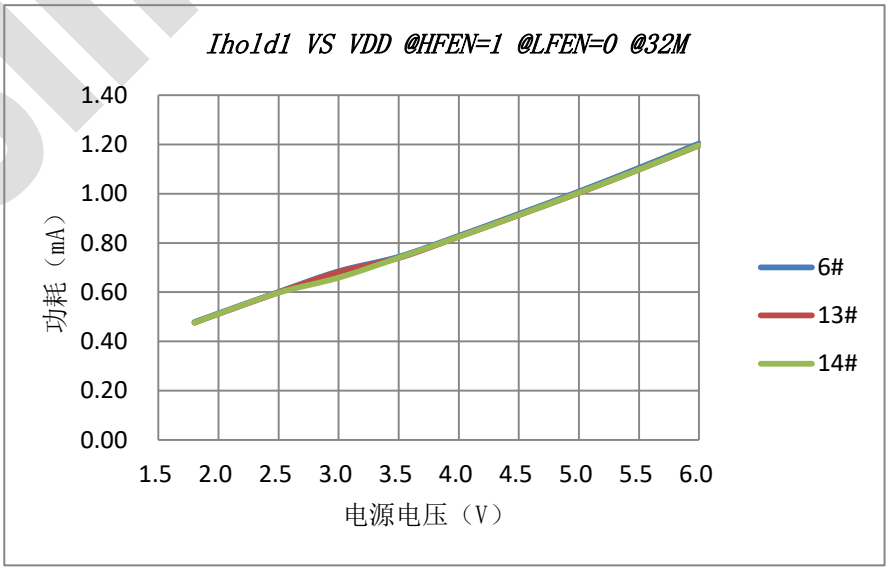

Irun VS VDD @1M @HIRC=16M
1.400
1.200
1.000
0.800
0.600
0.400
0.200
0.000
Irun（mA）
1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0
VDD（V）
6#
7#
9#

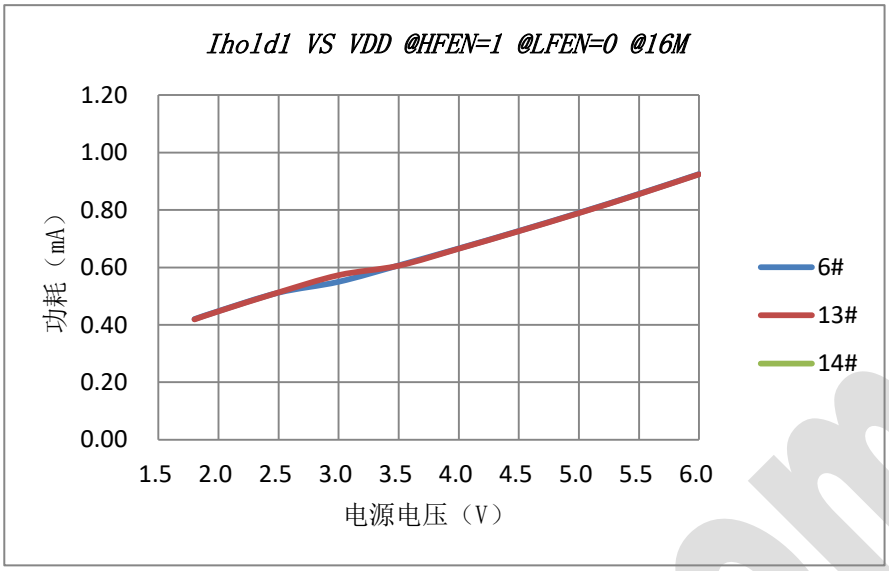

Irun VS VDD @500K @HIRC=16M
1.200
1.000
0.800
0.600
0.400
0.200
0.000
Irun（mA）
1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0
VDD（V）
6#
7#
9#

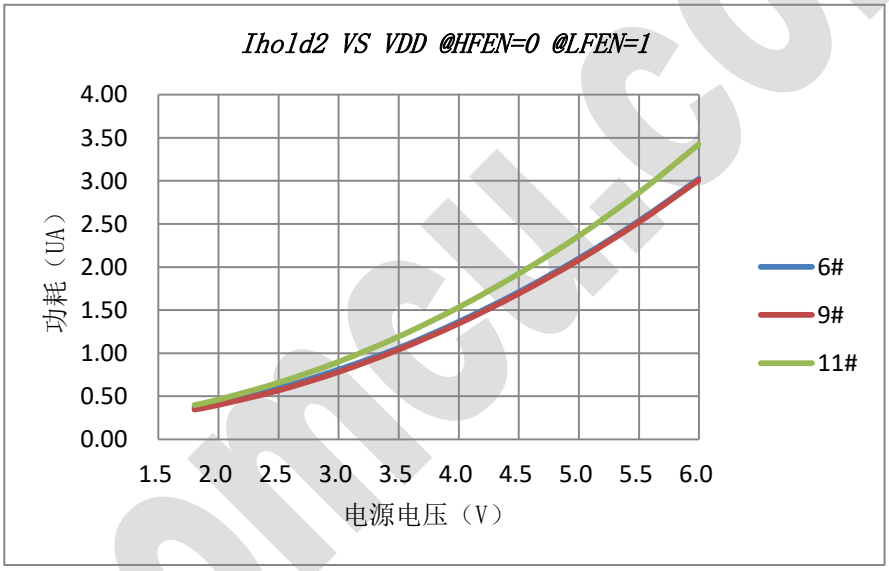




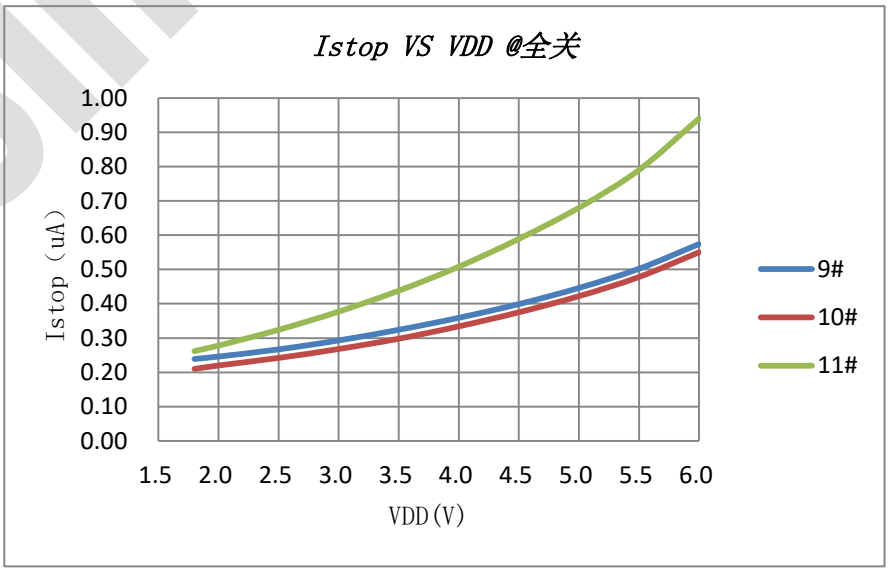
HOLD 模式 功耗 VS 电源电压

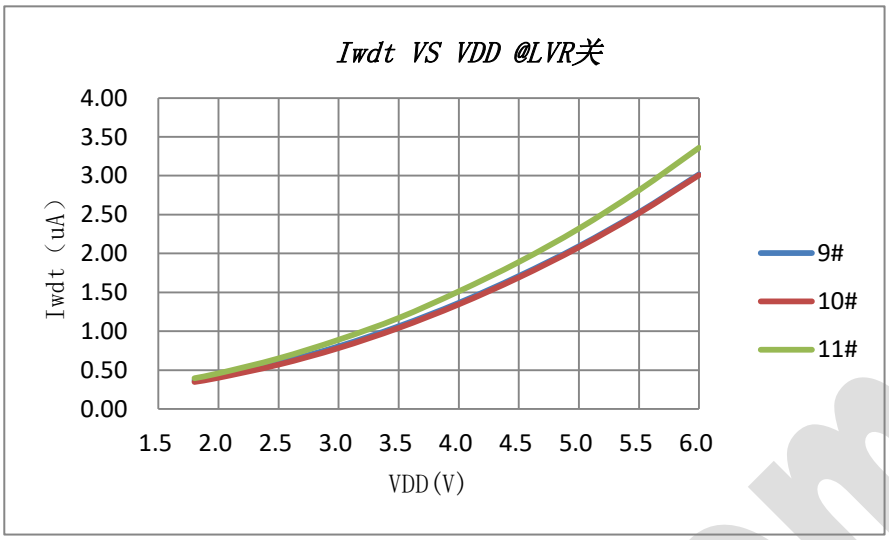


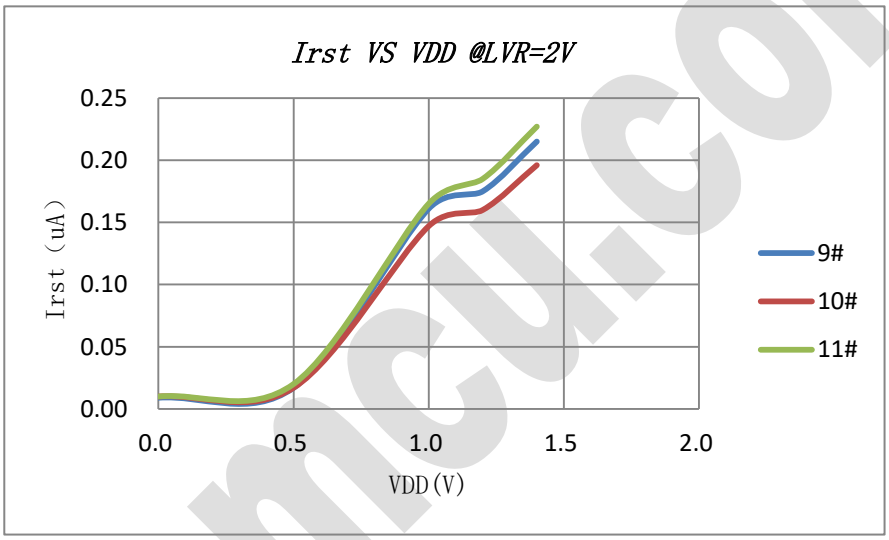




休眠模式 功耗 VS 电源电压

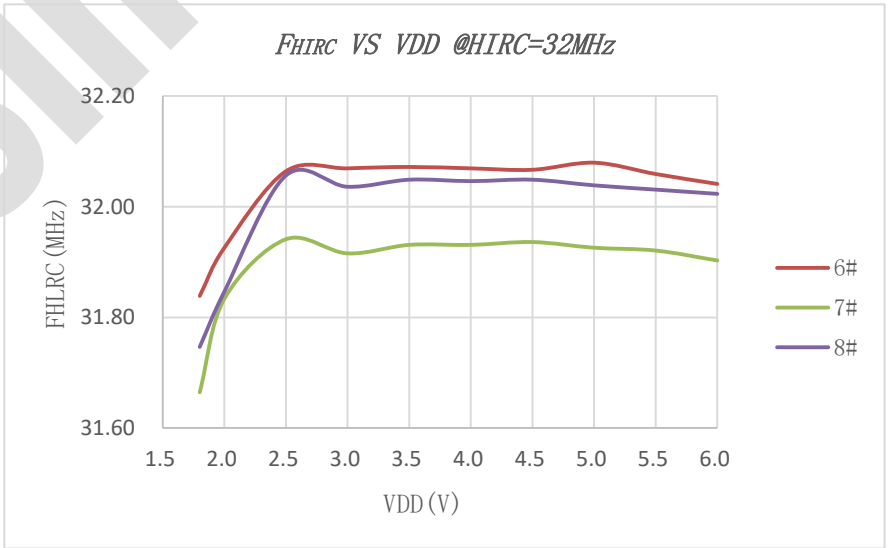


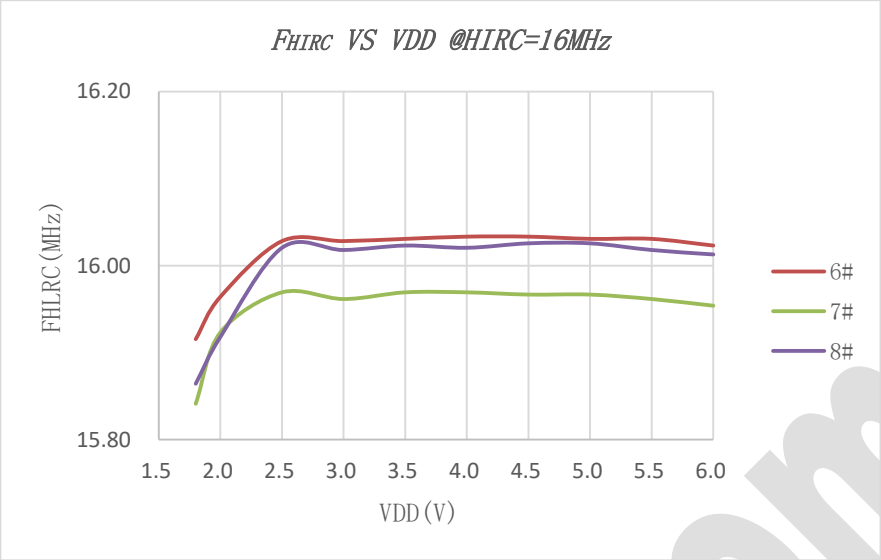


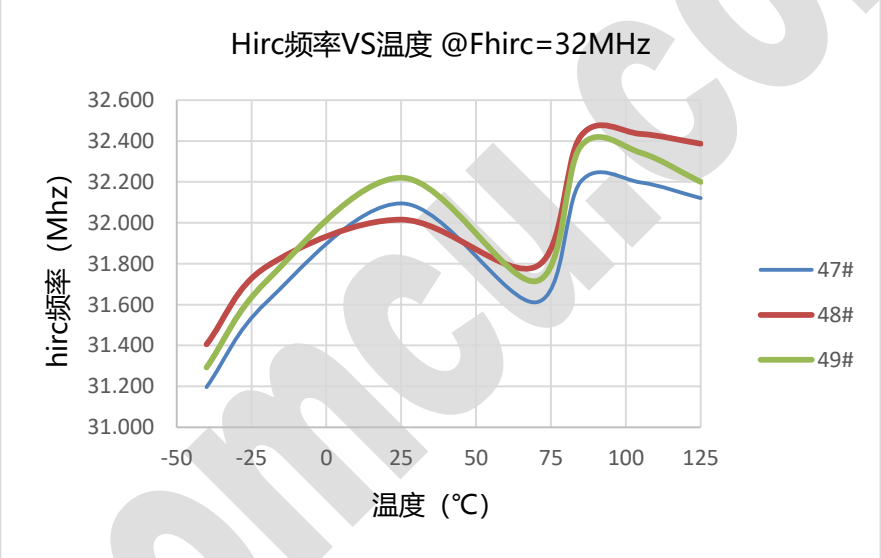


14.3 模拟电路特性

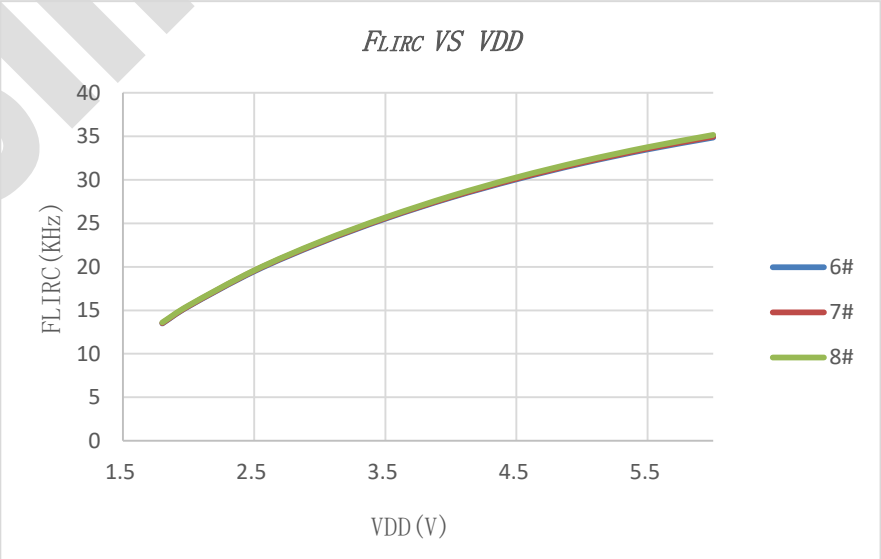
HIRC 频率 VS 电源电压/温度

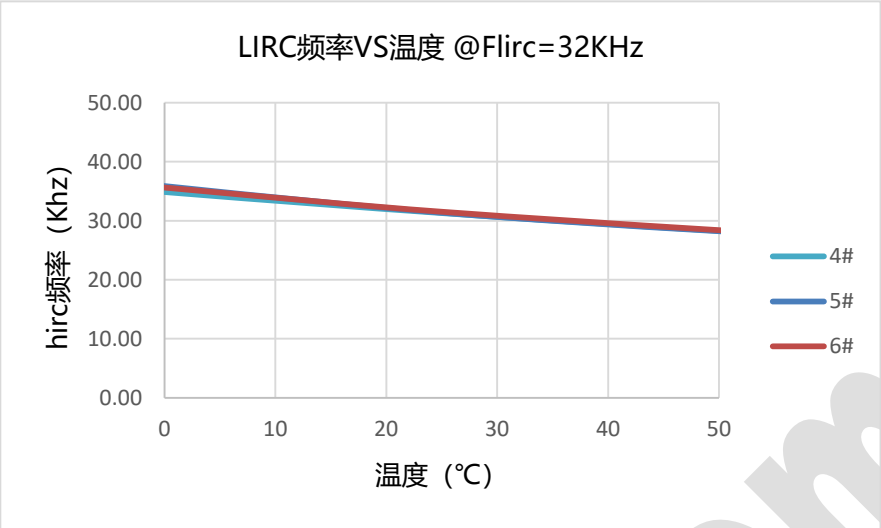




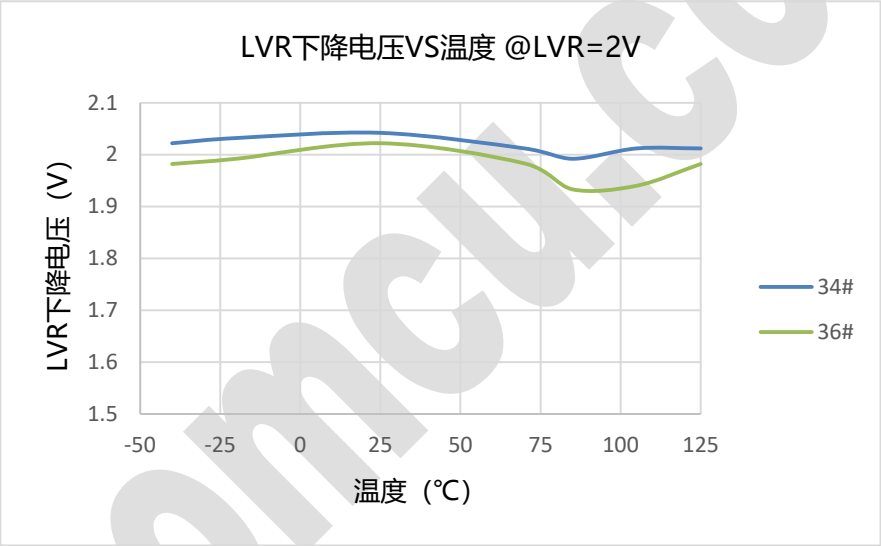


LIRC 频率 VS 电源电压/温度

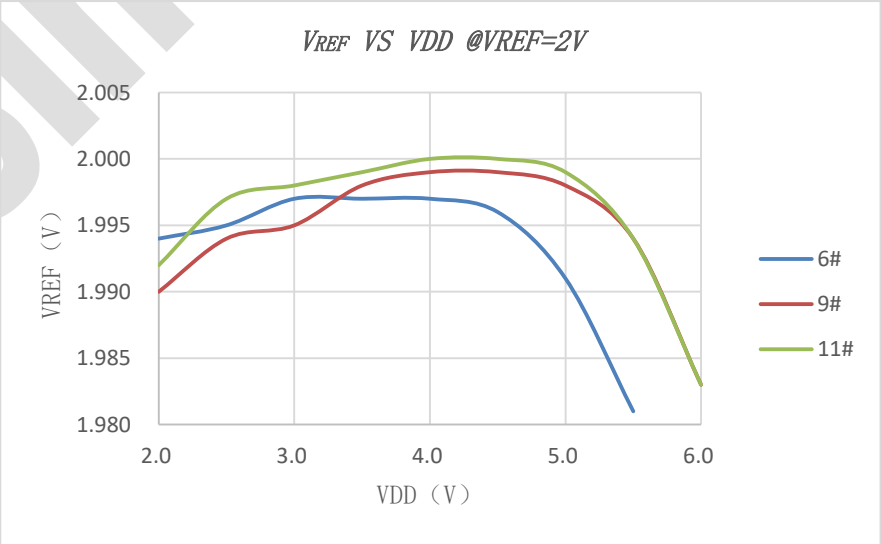


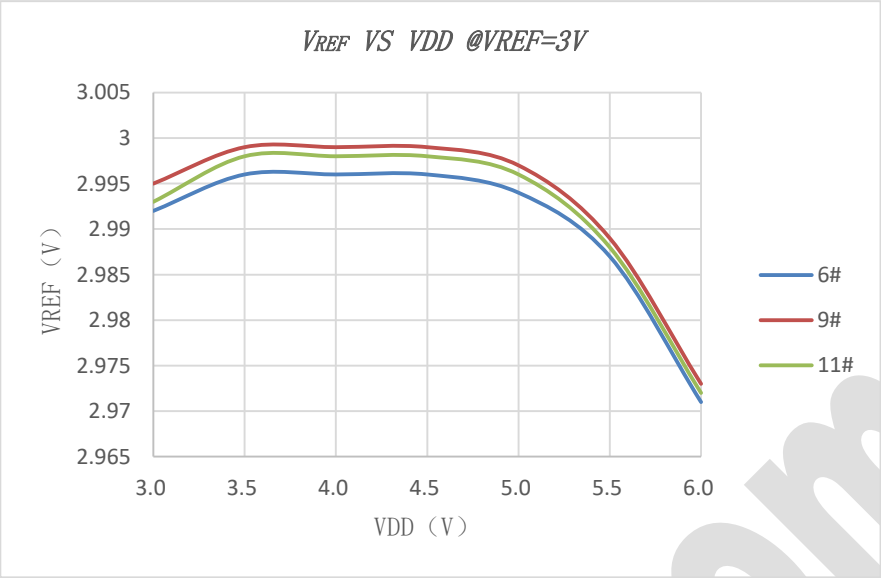


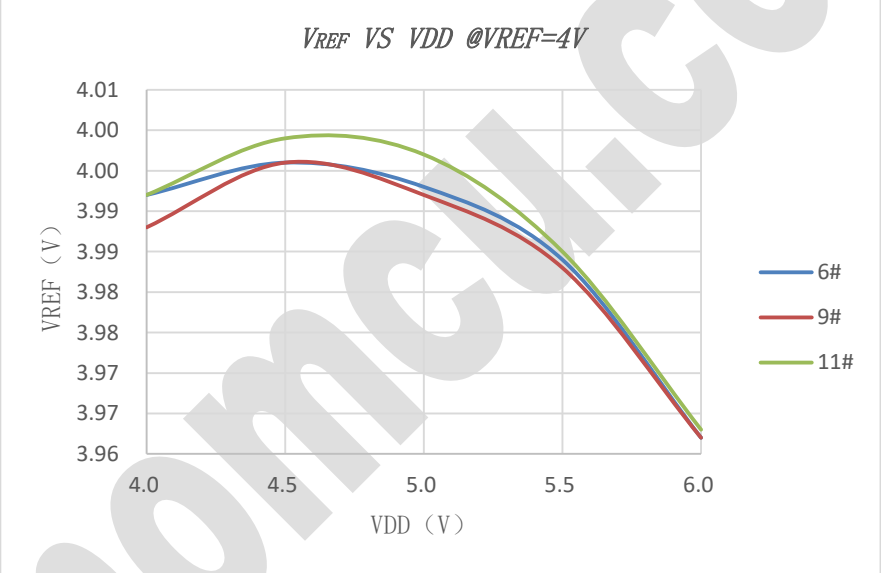
LVR 阈值电压 VS 温度

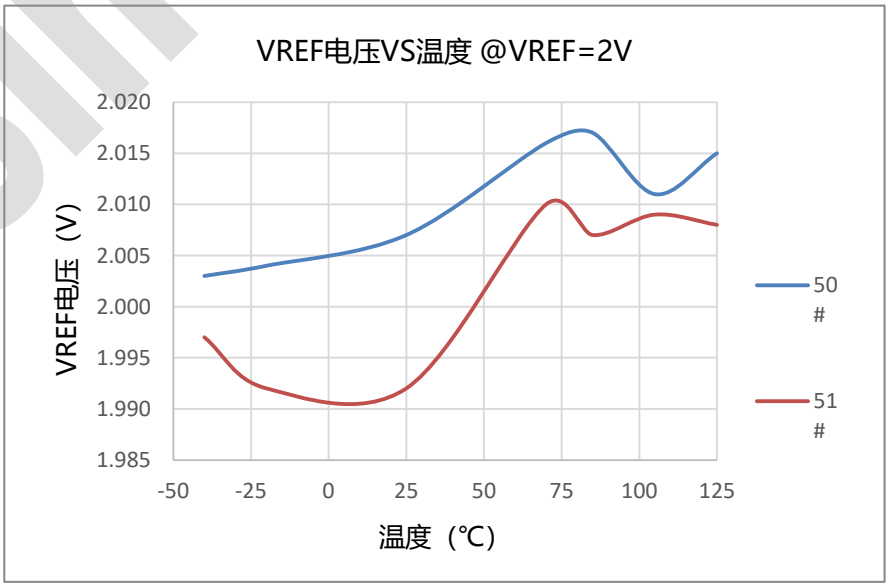


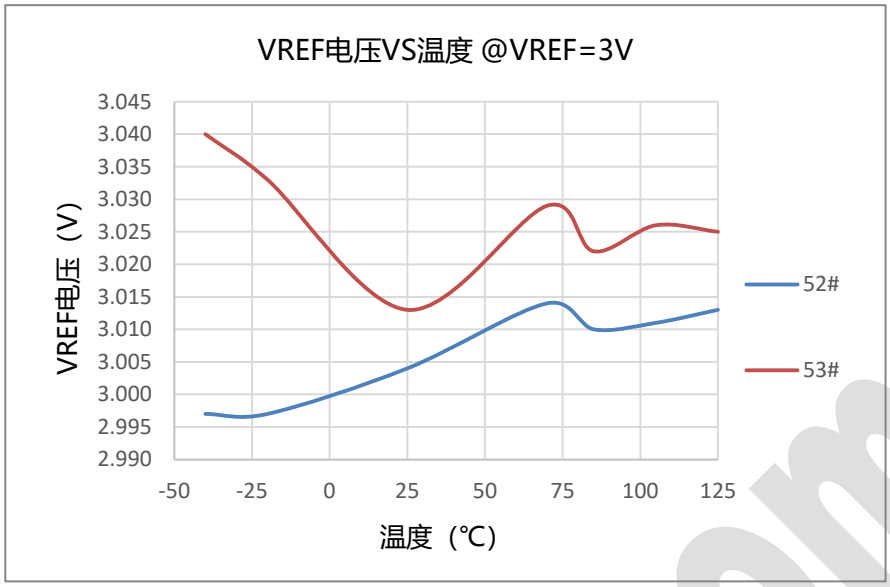
VIR 电压 VS 电源电压/温度

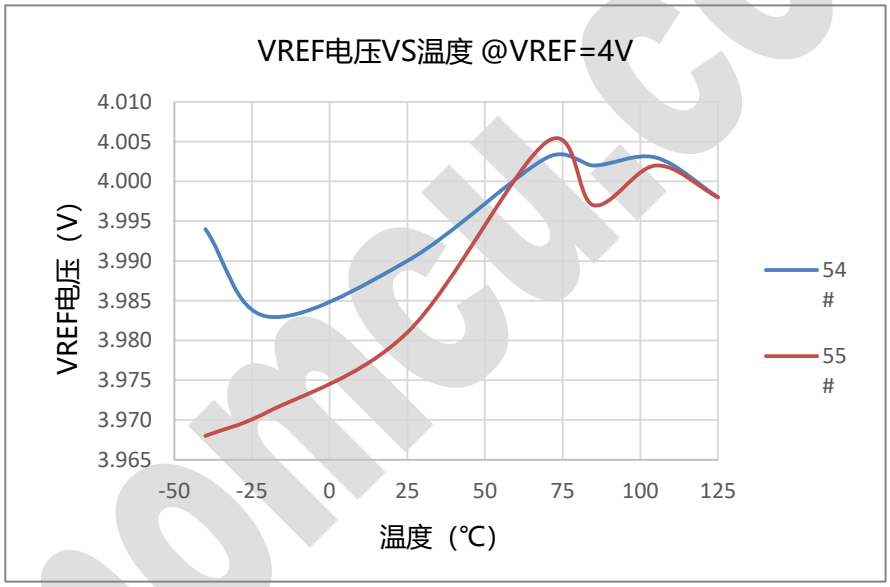



VREF VS VDD @VREF=3V
3.005
3
2.995
2.99
2.985
2.98
2.975
2.97
2.965
VREF（V）
3.0
3.5
4.0
4.5
5.0
5.5
6.0
VDD（V）
6#
9#
11#


VREF VS VDD @VREF=4V
4.01
4.00
4.00
3.99
3.99
3.98
3.98
3.97
3.97
3.96
VREF（V）
4.0
4.5
5.0
5.5
6.0
VDD（V）
6#
9#
11#

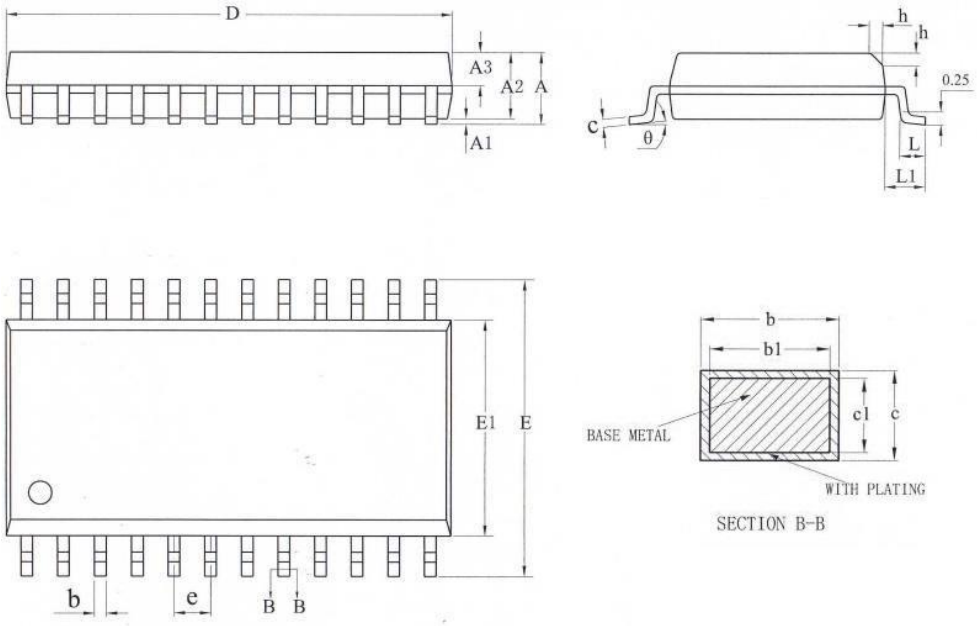

VREF电压VS温度 @VREF=2V
2.020
2.015
2.010
2.005
2.000
1.995
1.990
1.985
VREF电压（V）
-50
-25
0
25
50
75
100
125
温度（℃）
50#
51#


VREF电压VS温度 @VREF=3V
VREF电压（V）
3.045
3.040
3.035
3.030
3.025
3.020
3.015
3.010
3.005
3.000
2.995
2.990
-50
-25
0
25
50
75
100
125
温度（℃）
52#
53#


VREF电压VS温度 @VREF=4V
VREF电压（V）
4.010
4.005
4.000
3.995
3.990
3.985
3.980
3.975
3.970
3.965
-50
-25
0
25
50
75
100
125
温度（℃）
54#
55#

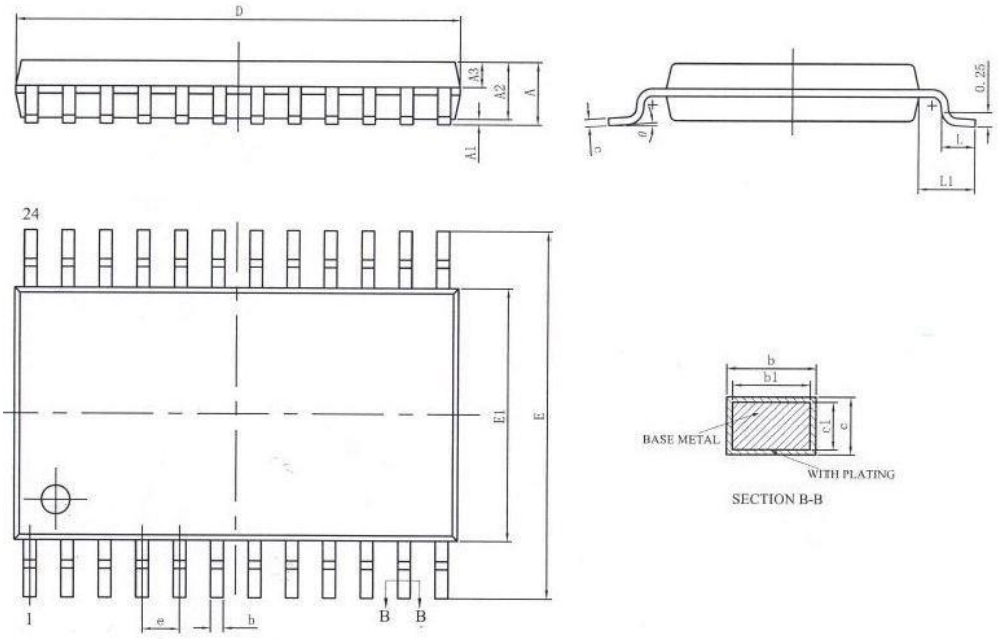
15 封装尺寸

15.1 SOP24



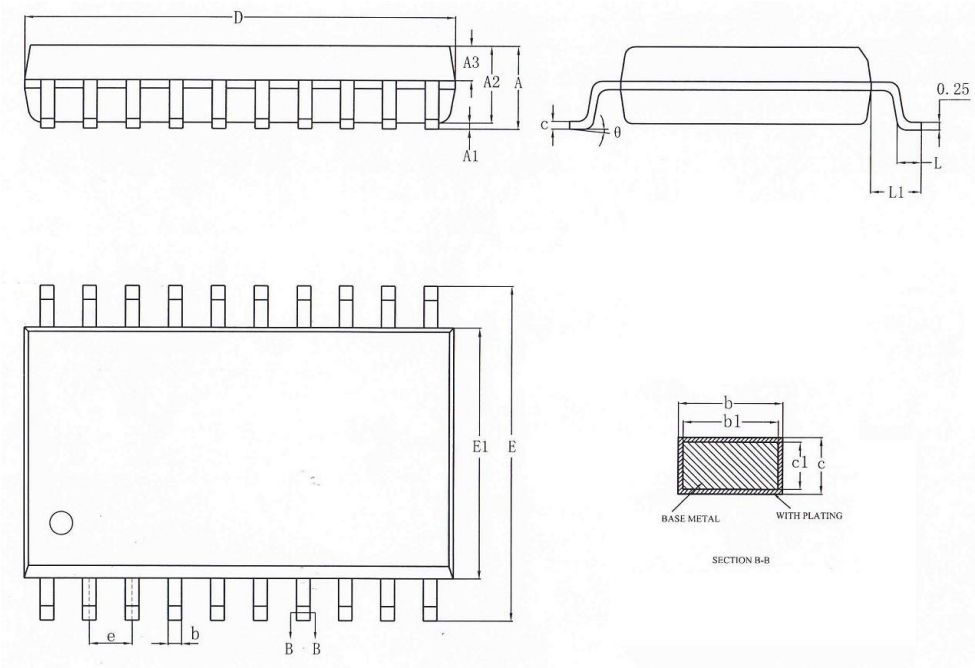
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	2.36	2.54	2.64
A1	0.10	0.20	0.30
A2	2.26	2.30	2.35
A3	0.97	1.02	1.07
b	0.39	—	0.47
b1	0.38	0.41	0.44
c	0.25	—	0.29
c1	0.24	0.25	0.26
D	15.30	15.40	15.50
E	10.10	10.30	10.50
E1	7.40	7.50	7.60
e	1.27BSC		
L	0.70	—	1.00
L1	1.40REF		
h	0.25	—	0.75
θ	0	—	8°

15.2 TSSOP24



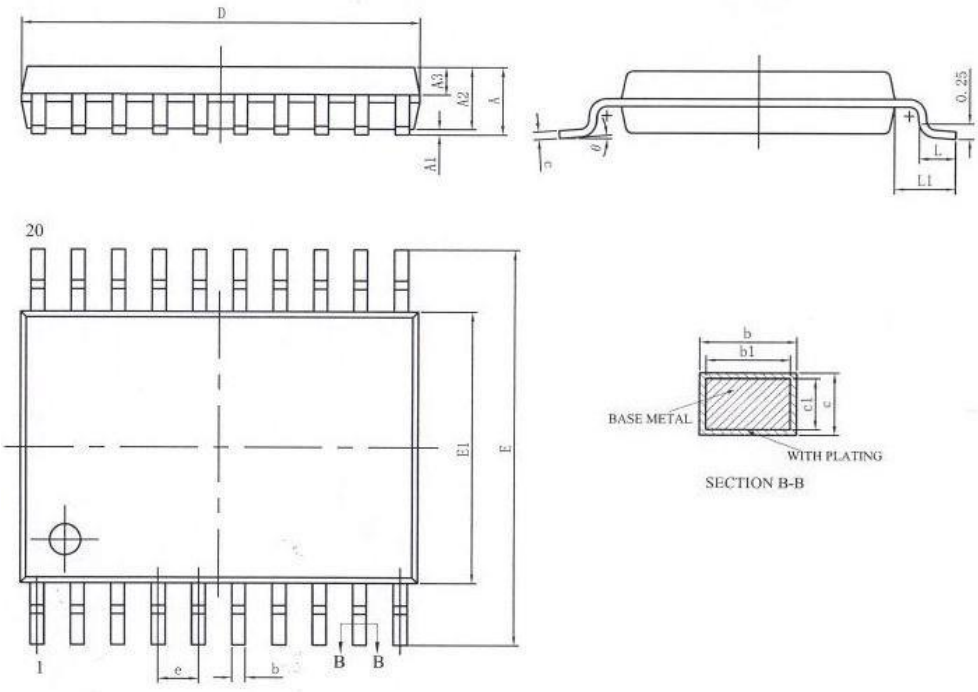
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.20
A1	0.05	—	0.15
A2	0.80	1.00	1.05
A3	0.39	0.44	0.49
b	0.20	—	0.29
b1	0.19	0.22	0.25
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	7.70	7.80	7.90
E	6.20	6.40	6.60
E1	4.30	4.40	4.50
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00BSC		
θ	0	—	8°

15.3 SOP20



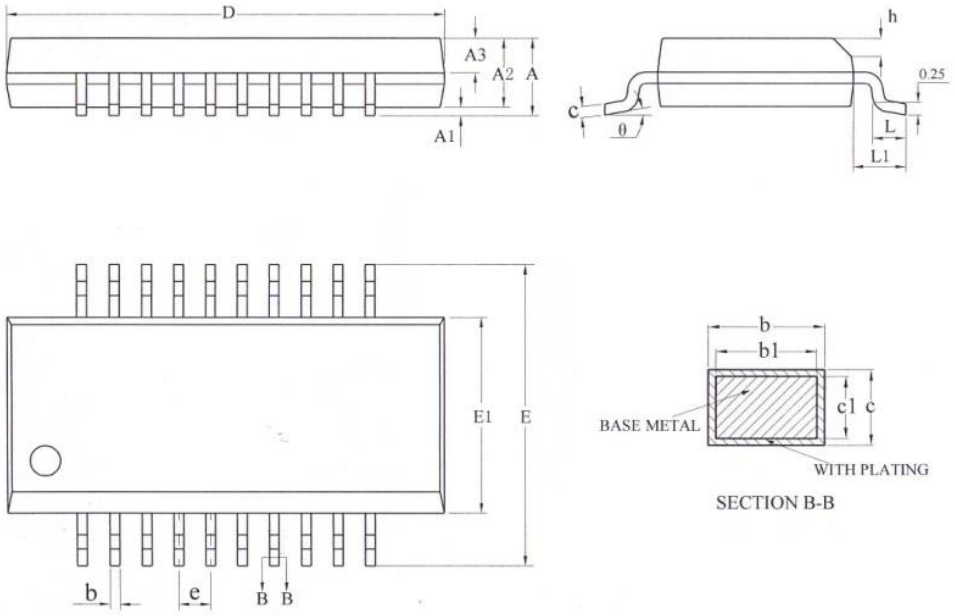
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	2.65
A1	0.10	—	0.30
A2	2.25	2.30	2.35
A3	0.97	1.02	1.07
b	0.35	—	0.43
b1	0.34	0.37	0.40
c	0.25	—	0.29
c1	0.24	0.25	0.26
D	12.70	12.80	12.90
E	10.10	10.30	10.50
E1	7.40	7.50	7.60
e	1.27BSC		
L	0.70	—	1.00
L1	1.40REF		
θ	0	—	8°

15.4 TSSOP20



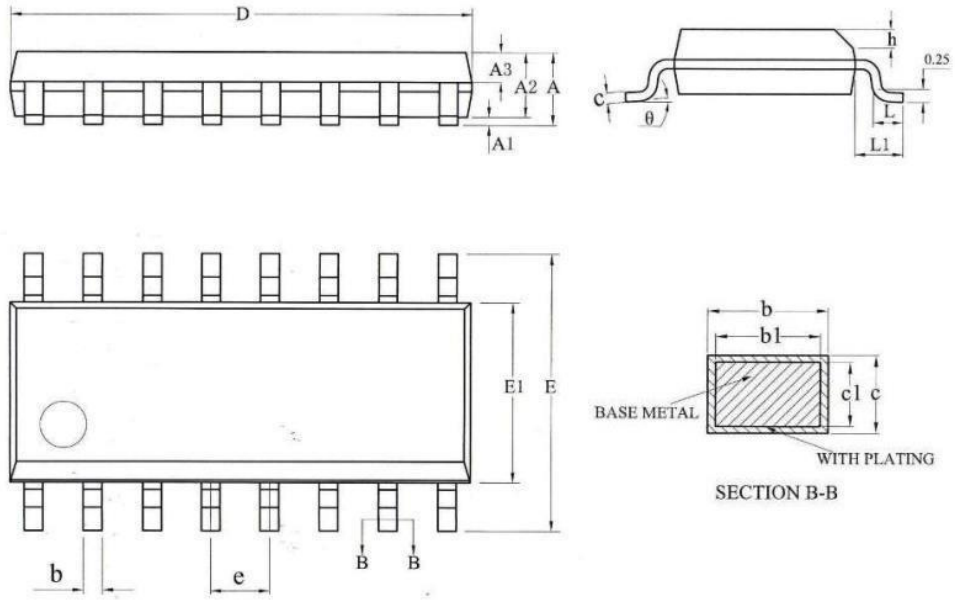
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.20
A1	0.05	—	0.15
A2	0.80	1.00	1.05
A3	0.39	0.44	0.49
b	0.20	—	0.29
b1	0.19	0.22	0.25
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	6.40	6.50	6.60
E1	4.30	4.40	4.50
E	6.20	6.40	6.60
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00BSC		
θ	0	—	8°

15.5 SSOP20（0.635）



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.10	0.15	0.25
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.23	—	0.31
b1	0.22	0.25	0.28
c	0.20	—	0.24
c1	0.19	0.20	0.21
D	8.55	8.65	8.75
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	0.635BSC		
h	0.30	—	0.50
L	0.50	—	0.80
L1	1.05REF		
θ	0	—	8°

15.6 SOP16



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.10	—	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	—	0.47
b1	0.38	0.41	0.44
c	0.20	—	0.24
c1	0.19	0.20	0.21
D	9.80	9.90	10.00
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	1.27BSC		
h	0.25	—	0.50
L	0.50	—	0.80
L1	1.05REF		
θ	0	—	8°

16 修订记录

版本	日期	修订内容
V1.0	2023-02-20	发布初版；
V1.1	2023-05-15	新增 SOP16 封装 A1K；
V1.2	2023-07-10	补充 VDD=3V 的功耗参数；