

晟矽微电 8 位单片机

MS32F7223

用户手册

V1.1.2





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.....	11
2.1	极限参数.....	11
2.2	直流电气特性.....	11
2.3	交流电气特性.....	13
2.4	ADC 特性参数.....	13
2.5	BG 特性参数.....	14
2.6	VIR 特性参数.....	14
2.7	LDO 特性参数.....	15
2.8	温度传感器特性参数.....	15
2.9	EEPROM 特性参数.....	15
3	CPU 与存储器.....	17
3.1	指令集.....	17
3.2	程序存储器.....	19
3.3	数据存储器.....	20
3.4	堆栈.....	22
3.5	控制寄存器.....	22
3.5.1	数据指针寄存器 0.....	22
3.5.2	数据指针寄存器 1.....	22
3.5.3	间接寻址寄存器 0.....	22
3.5.4	间接寻址寄存器 1.....	23
3.5.5	间接寻址寄存器 2.....	23
3.5.6	间接寻址寄存器 3.....	23
3.5.7	字操作高字节缓存器.....	23
3.5.8	程序指针计数器低字节.....	24
3.5.9	CPU 状态寄存器.....	24
3.5.10	杂项控制寄存器.....	24
3.6	用户配置字.....	25
4	系统时钟.....	28
4.1	内部高频 RC 振荡器.....	28
4.2	内部低频 RC 振荡器.....	29
4.3	外部晶体振荡器.....	29



4.4	系统工作模式	29
4.4.1	振荡器模式寄存器	30
4.5	低功耗模式	31
5	复位	33
5.1	复位条件	33
5.2	上电复位	33
5.3	外部复位	33
5.4	低电压复位	34
5.5	看门狗复位	34
5.6	软件复位	34
5.6.1	DEBUG 控制寄存器	34
6	I/O 端口	35
6.1	通用 I/O 功能	35
6.1.1	端口数据寄存器	35
6.1.2	端口方向寄存器	35
6.2	内部上/下拉电阻	36
6.2.1	上拉电阻控制寄存器	36
6.2.2	下拉电阻控制寄存器	36
6.3	端口模式控制	37
6.3.1	端口数模控制寄存器	37
7	定时器 TIMER	38
7.1	看门狗定时器 WDT	38
7.2	定时器 T0	38
7.2.1	定时器 T0 控制寄存器	39
7.2.2	定时器 T0 计数器	39
7.2.3	定时器 T0 重载寄存器	40
7.3	定时器 T1	40
7.3.1	定时器 T1 控制寄存器	41
7.3.2	定时器 T1 计数器	42
7.3.3	定时器 T1 重载寄存器	43
7.3.4	定时器 T1 比较寄存器	43
7.4	定时器 T2	44
7.4.1	定时器 T2 控制寄存器	46
7.4.2	定时器 T2 计数器	47
7.4.3	定时器 T2 重载寄存器	47
7.4.4	定时器 T2 比较寄存器	48
7.4.5	PWM2 控制寄存器	48
8	模数转换器 SD-ADC	51
8.1	ADC 概述	51
8.2	ADC 相关寄存器	52



8.2.1	ADC 控制寄存器.....	52
8.2.2	ADC 转换结果寄存器.....	56
8.3	ADC 操作步骤.....	57
8.4	ADC 应用注意.....	57
8.4.1	PGA 差分输入/输出电压超范围控制.....	57
8.5	温度传感器.....	58
9	低电压检测 LVD.....	60
9.1.1	LVD 控制寄存器.....	60
10	总线通讯 IIC.....	61
10.1	IIC 概述.....	61
10.2	IIC 数据传输.....	61
10.3	IIC 工作模式.....	62
10.4	IIC 时序说明.....	63
10.5	IIC 多机通讯.....	64
10.6	IIC 相关寄存器.....	65
10.6.1	IIC 控制寄存器.....	65
10.6.2	IIC 状态寄存器.....	66
10.6.3	IIC 数据寄存器.....	67
10.6.4	IIC 主地址寄存器.....	68
10.6.5	IIC 辅地址寄存器.....	68
10.6.6	IIC 辅地址掩码寄存器.....	68
11	异步通讯 UART.....	69
11.1	UART 概述.....	69
11.2	UART 工作模式与传输格式.....	70
11.3	UART 波特率.....	70
11.4	UART 多机通讯.....	71
11.4.1	从机地址自动识别.....	71
11.4.2	预设地址和广播地址.....	71
11.4.3	端口输出自动转换.....	72
11.5	UART 相关寄存器.....	73
11.5.1	UART 控制寄存器.....	73
11.5.2	UART 状态寄存器.....	75
11.5.3	UART 波特率寄存器.....	76
11.5.4	UART 数据寄存器.....	76
11.5.5	UART 地址寄存器.....	76
11.5.6	UART 地址掩码寄存器.....	76
12	在应用自编程 IAP.....	77
12.1	IAP 概述.....	77
12.2	IAP 操作步骤.....	78
12.3	IAP 相关寄存器.....	79



12.3.1	IAP 控制寄存器	79
12.3.2	IAP 保护寄存器	81
12.3.3	IAP 地址寄存器	81
12.3.4	IAP 数据寄存器	82
12.4	IAP 操作示例	82
13	在板烧录编程 ICP	88
13.1.1	DEBUG 控制寄存器	88
13.1.2	DEBUG 保护寄存器	89
14	中断	90
14.1	外部中断	90
14.2	定时器中断	90
14.3	ADC 中断	90
14.4	键盘中断	91
14.4.1	键盘中断控制寄存器	91
14.5	LVD 中断	91
14.6	IIC 中断	91
14.7	UART 中断	91
14.8	中断相关寄存器	92
14.8.1	中断使能寄存器	92
14.8.2	中断标志寄存器	93
15	特性曲线	96
15.1	I/O 特性	96
15.1.1	输入 SMT 阈值电压 VS 电源电压	96
15.1.2	上/下拉电阻值 VS 电源电压	97
15.1.3	I/O 输出电流 VS 端口电压 (VDD=3V)	98
15.2	功耗特性	101
15.2.1	运行模式 功耗 VS 电源电压	101
15.2.2	HOLD 模式 功耗 VS 电源电压	104
15.2.3	休眠模式 功耗 VS 电源电压	105
15.3	模拟电路特性	106
15.3.1	HIRC 频率 VS 电源电压/温度	106
15.3.2	LIRC 频率 VS 电源电压/温度	107
15.3.3	XTAL 功耗 VS 电源电压	108
15.3.4	LVR 阈值电压 VS 温度	109
16	封装尺寸	111
16.1	SOP16	111
16.2	SOP8	111
17	修订记录	112



1 产品概要

1.1 产品特性

- ▶ 8 位 CPU 内核
 - 精简指令集, 8 级深度硬件堆栈
 - CPU 为双时钟, 可在系统高/低频时钟之间切换
 - 系统高频时钟下 F_{CPU} 可配置为 F_{HOSC} 的 2/4/8/16 分频
 - 系统低频时钟下 F_{CPU} 固定为 F_{LOSC} 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和在应用自编程, 擦写次数至少 10,000 次
- ▶ 数据存储器
 - 512 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 (其中 256 字节分为 2 个 128 字节存储器组, 寻址时需先通过寄存器位选择相应分组)
 - 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 10,000 次
- ▶ 2 组共 14 个 I/O
 - P0 (P00~P05), P1 (P10~P17)
 - 所有端口均支持施密特输入, 均支持推挽输出
 - P10 可复用为外部复位 RST 输入, P00/P01 或 P02/P03 复用为 IIC 接口时为开漏输出
 - P10/P11 可复用为外部时钟振荡器输入/输出
 - 所有端口均内置上拉和下拉电阻, 均可单独使能
 - P16/P17 可复用为外部中断输入, 支持外部中断唤醒功能
 - P0 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 (8MHz±2%@-20°C~70°C/4MHz/2MHz/1MHz), 可用作系统高频时钟源, 且其中 8MHz 时钟可作为外设高频时钟 F_{HCLK} 用作部分外设模块的时钟源
 - 内置低频 RC 振荡器 (32KHz@5V, 22KHz@3V), 可用作系统低频时钟源
 - 支持外接低频晶体振荡器 (32768Hz), 可用作系统低频时钟源
- ▶ 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: 16ms/64ms/256ms/1024ms@5V, 23ms/93ms/372ms/1489ms@3V
 - 工作模式可配置: 始终开启、始终关闭、低功耗模式下关闭
- ▶ 3 个定时器



- 16 位定时器 T0, 可实现外部计数功能
- 12 位定时器 T1, 可实现外部计数和 12 位 PWM 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 24 位全差分 Σ - Δ 型 ADC
 - 工作电压为内部 LDO 输出电压 VLDO (可输出 VDD 电压)
 - 正/负端输入共支持 14 路对 GND 差分的外部单输入通道 (AN0~AN13), 也可组成多对外部差分输入通道
 - 内置前级增益可编程放大电路 PGA, 高输入阻抗
 - 参考电压可选: VLDO、VLDO/2、内部参考电压 VIR (1.024V, 可从端口 VIRO 输出)、外部参考电压 VER (VERI 输入)
 - ADC 时钟: 外设高频时钟 FHCLK (8MHz) 的 8/24 分频
 - 差分输入放大倍数 (含前级 PGA 放大) 为 0.25/0.5/1/2/4/8/16/32/64/128/256 可选
 - 过采样率 64~32768 可选, 采样速率 10Hz@32768~15.626KHz@64 可选
 - 支持 50Hz/60Hz 工频滤波@采样速率=10Hz
 - 内置温度传感器-, 输出电压 Vts 可作为 ADC 的正/负端输入信号
- ▶ 高速线性稳压电路 LDO
 - 输出电压 (典型值) 2.5V/2.8V/3.0V/3.3V 可选
 - 需从端口 LDO 外接 0.1 μ F~10 μ F 对地电容
- ▶ 1 组总线通讯 IIC 主从接口
 - 支持 7 位地址编码的主机/从机模式, 从机模式支持 2 组地址编码及广播地址 (00H) 编码
 - 时钟源为外设高频时钟 FHCLK (8MHz), 通讯速率理论最高支持 400Kbps (受芯片及电路等影响, 实际可能略低)
 - 地址匹配、接收完成、发送完成等事件发生时, 可触发中断
 - 支持多主机时钟同步和总线仲裁
 - 2 组端口 SCL0/SDA0 或 SCL1/SDA1 可选
- ▶ 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, 时钟源为外设高频时钟 FHCLK (8MHz), 波特率最高支持 384Kbps
 - 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 (P00~P05)
 - 定时器中断 (T0~T2)
 - ADC 中断, LVD 中断
 - IIC 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
 - 1.8V/1.9V/2.0V/2.1V/2.2V/2.3V/2.4V/2.5V/2.6V/2.7V/2.8V/2.9V/3.0V/3.1V/3.2V/3.3V
- ▶ 低电压复位 LVR



- 1.8V/2.0V/2.4V
- ▶ 工作电压
 - VLVR20 ~ 5.5V @ Fcpu = 0~4MHz
 - VLVR18 ~ 5.5V @ Fcpu = 0~1MHz
- ▶ 封装形式
 - SOP16/SOP8

1.2 订购信息

表 1-1 订购信息

产品名称	封装形式	备注
MS32F7223A0K	SOP16	
MS32F7223A1K	SOP16	
MS32F7223A0H	SOP8	
MS32F7223A1H	SOP8	

1.3 引脚排列

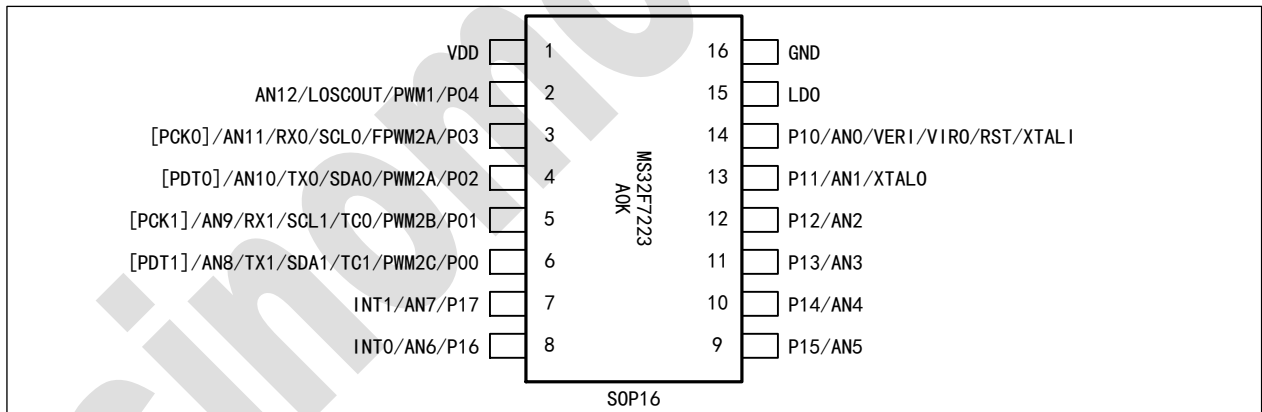


图 1-1 MS32F7223A0K

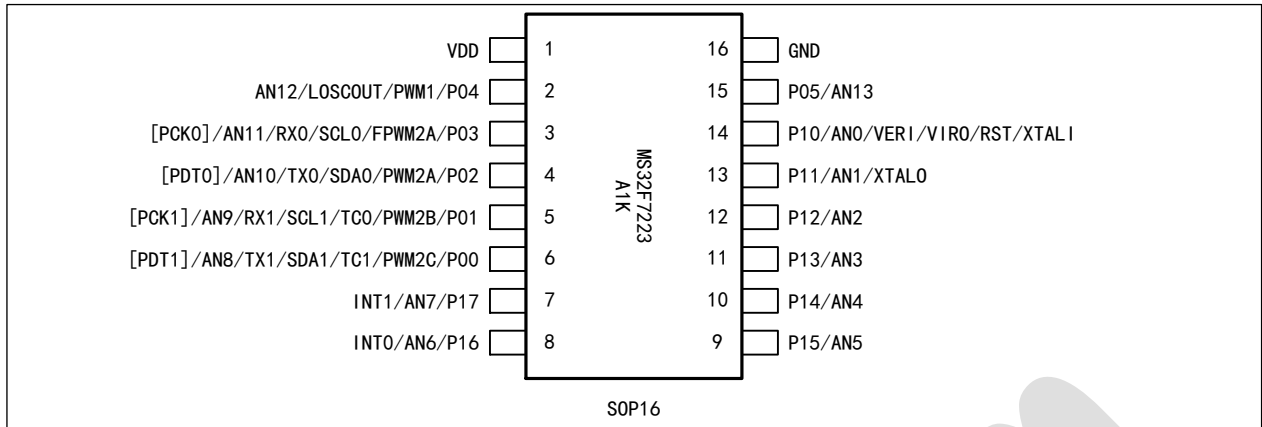


图 1-2 MS32F7223A1K

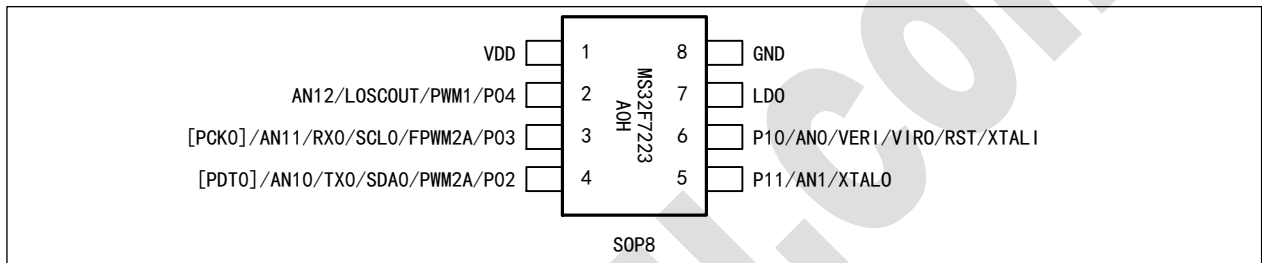


图 1-3 MS32F7223A0H

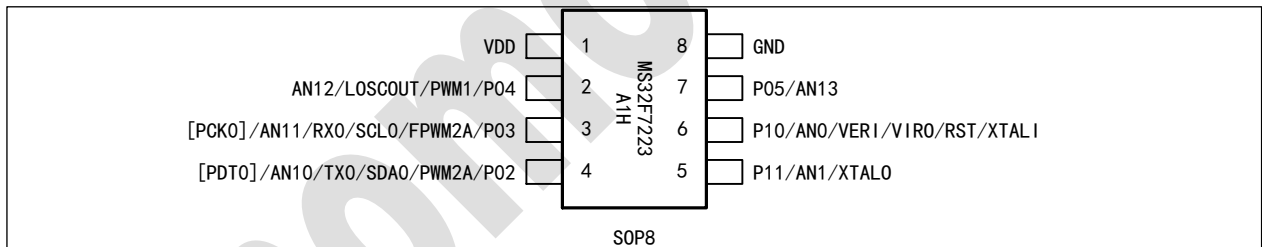


图 1-4 MS32F7223A1H

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
P00~P05, P10~P17	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
PWM1	DO	定时器 T1 的 PWM 输出
PWM2A~PWM2C	DO	定时器 T2 的 3 路 PWM 输出



FPWM2A	DO	定时器 T2 中 PWM2A 的互补 PWM 输出
AN0~AN13	AI	ADC 外部输入通道
VERI	AI	ADC 外部参考电压输入
VIRO	AO	ADC 内部参考电压输出
LDO	AO	LDO 输出, 需外接对地电容
SCL0/SDA0, SCL1/SDA1	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
XTALI, XTALO	A	外部 32768Hz 晶体振荡器输入/输出
LOSCOUT	DO	系统低频时钟信号输出
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

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	50	mA
流出 GND 最大电流	IGNDmax	50	mA

注：若芯片工作条件超过极限值，则将造成永久性损坏；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=3V, 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=4MHz@FHIRC(8M)/2	VLVR20		5.5	V
			Fcpu=2MHz@FHIRC(8M)/4	VLVR20		5.5	
			Fcpu=1MHz@FHIRC(8M)/8	VLVR18		5.5	
			Fcpu=500KHz@FHIRC(8M)/16	VLVR18		5.5	
			Fcpu=16KHz@FLIRC(32K)/2	VLVR18		5.5	
输入漏电流	Ileak	所有输入脚	VDD=3V	-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	VDD=5V, Vin=0		16		KΩ
			VDD=3V, Vin=0		16		KΩ
	Rpu2	SCL/SDA	VDD=2.0V~5.5V, Vin=0, IIC 专用上拉		3		KΩ
下拉电阻	Rpd	P0, P1	Vin=VDD=5V		16		KΩ
			Vin=VDD=3V		16		KΩ
输出源电流	Ioh	推挽输出脚	VDD=5V, Voh=VDD-0.6V, 正常驱动		15		mA



			VDD=5V, Voh=VDD-0.6V, 增强驱动		25		mA
			VDD=3V, Voh=VDD-0.6V, 正常驱动		10		mA
			VDD=3V, Voh=VDD-0.6V, 增强驱动		15		mA
输出灌电流	Iol	所有输出脚	VDD=5V, Vol=0.6V, 正常驱动		20		mA
			VDD=5V, Vol=0.6V, 增强驱动		35		mA
			VDD=3V, Vol=0.6V, 正常驱动		15		mA
			VDD=3V, Vol=0.6V, 增强驱动		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 (VDD=5V)	Irun	VDD	Fcpu=4MHz@HIRC(8M)		1.35		mA
			Fcpu=2MHz@HIRC(8M)		800		μA
			Fcpu=1MHz@HIRC(8M)		480		μA
			Fcpu=500KHz@HIRC(8M)		330		μA
			Fcpu=16KHz@LIRC, HIRC off		9	20	μA
运行模式功耗 (VDD=3V)	Irun	VDD	Fcpu=4MHz@HIRC(8M)		0.85		mA
			Fcpu=2MHz@HIRC(8M)		500		μA
			Fcpu=1MHz@HIRC(8M)		320		μA
			Fcpu=500KHz@HIRC(8M)		220		μA
			Fcpu=2MHz@HIRC(4M)		460		μA
			Fcpu=1MHz@HIRC(4M)		300		μA
			Fcpu=1MHz@HIRC(2M)		290		μA
			Fcpu=11KHz@LIRC, HIRC off		5	10	μA
HOLD1 功耗	Ihold1	VDD	VDD=5V, CPU 停, HIRC/LIRC 开		150		μA
			VDD=3V, CPU 停, HIRC/LIRC 开		120		μA
HOLD2 功耗	Ihold2	VDD	VDD=5V, CPU 停, HIRC 关, LIRC 开		2.5	6	μA
			VDD=3V, CPU 停, HIRC 关, LIRC 开		1	3	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	2	μA
			VDD=5V, 休眠模式, WDT 开, LVR 关		2.5	6	μA
			VDD=3V, 休眠模式, WDT 开, LVR 关		1	3	μA
			VDD=5V, 休眠模式, WDT 关, LVR 开		9	20	μA
			VDD=3V, 休眠模式, WDT 关, LVR 开		6.5	15	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 _{HIRC}	VDD=3.0V, T=25°C	-1%	8/4/2/1	+1%	MHz
		VDD=2.0V~5.5V, T=-20°C~70°C	-2%		+2%	
		VDD=1.8V~5.5V, T=-40°C~85°C	-4%		+4%	
HIRC 起振时间		VDD=1.8V~5.5V, T=-40°C~85°C		10	100	μs
LIRC 振荡频率	F _{LIRC}	VDD=5V, T=25°C	-50%	32	+50%	KHz
		VDD=3V, T=25°C	-50%	22	+50%	KHz
32768 晶振起振电压		T=25°C	1.8			V
32768 晶振起振时间		VDD=3V, T=25°C		1		s

2.4 ADC 特性参数

VDD=3V, 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 _{DADC}	T=-40°C~85°C	2.5	V _{LDO}	5.5	V
ADC 动态电流 (低速采样) (OSR=32768, ODR=10Hz)	I _{DADD1}	V _{LDO} =2.5V, PGA 旁路, gain=1		180		μA
		V _{LDO} =2.5V, PGA 开启, gain=1		530		
		V _{LDO} =2.5V, PGA 开启, gain=128		530		
ADC 动态电流 (高速采样) (OSR=32768, ODR=30Hz)	I _{DADD2}	V _{LDO} =2.5V, PGA 旁路, gain=1		460		μA
		V _{LDO} =2.5V, PGA 开启, gain=1		1100		
		V _{LDO} =2.5V, PGA 开启, gain=128		1100		
ADC 静态电流	I _{DADS}	VDD=3V, ADC 关闭			1	μA
差分输入电压		V _{REF} =V _{DADC}			$\pm 0.5 \cdot V_{REF} / (PGA \cdot Gain)$	V
		V _{REF} =V _{IR} (1.024V)			$\pm 0.9 \cdot V_{REF} / (PGA \cdot Gain)$	V



输入电压			0		V _{DADC}	V
共模输入电压		PGA 开启	0.1		V _{DADC} -0.1	V
		PGA 旁路	0		V _{DADC}	V
共模输出电压		PGA 开启	0.1		V _{DADC} -0.1	V
PGA 输入阻抗			10			MΩ
转换时钟	F _{DADC}		333		1000	KHz
过采样率	OSR		64		32768	
采样速率(数据输出率)	ODR	过采样率 32768@低速采样	10			Hz
分辨率	RES			23		Bits
有效位	ENOB	V _{DADC} =2.5V, F _{DADC} =333KHz, ODR=10Hz,		18		Bits
无噪声码	NFB	gain=1		15.3		Bits

2.5 BG 特性参数

VDD=3V, T=25°C

表 2-5 Bandgap 特性参数

特性	符号	条件	最小	典型	最大	单位
BG 有效工作电压	V _{vb}		1.8	3	5.5	V
工作电流	I _{vb}			22		μA
BG 电压	V _{bg}			1.2		V

2.6 VIR 特性参数

VDD=3V, T=25°C

表 2-6 参考电压特性

特性	符号	条件	最小	典型	最大	单位
ADC 参考电压	V _{REF}	选择 V _{LDO}		V _{LDO}		V
		选择 V _{LDO} /2		V _{LDO} /2		
		选择内部参考电压 V _{IR} , T=25°C	-0.5%	1.024	+0.5%	
		选择内部参考电压 V _{IR} , T=-20°C~70°C	-2%		+2%	
		选择内部参考电压 V _{IR} , T=-40°C~85°C	-3%		+3%	
		选择外部参考电压 V _{ER}	2		VDD	
VIR 有效工作电压	V _{VIR}	选择内部参考电压 V _{IR}	V _{IR} +0.2		3.6	V
工作电流	I _{VIR}			45		μA
温度系数				50		ppm/



						°C
--	--	--	--	--	--	----

2.7 LDO 特性参数

表 2-7 LDO 特性参数

特性	符号	条件	最小	典型	最大	单位
LDO 有效工作电压	V _{LDO}	I _{LDO} =1mA	V _{LDO} +0.2	3	V _{LIN}	V
工作电流	I _{LDO}	V _{LI} =3V		60	120	μA
输入电压范围	V _{LIN}	I _{LDO} =1mA	V _{LDO} +0.2		5.5	V
输出电压	V _{LDO}	LDOVS V _{LIN} =3V, I _{LDO} =1mA, T=25°C	-2%	2.5/2.8	+2%	V
		=0xx V _{LIN} =3V, I _{LDO} =1mA, T=-40°C~85°C	-3%	/3.0/3. 3	+3%	
		LDOVS=0xx		V _{DD}		
LDO 输出温度系数				50	100	ppm/ °C
输出电流	I _{LDO}	V _{LIN} =2.7V~5.5V, T=-40°C~85°C	10			mA
输入-输出电压降	V _{drop}	I _{LO} =10mA		0.2	0.3	V
LDO 启动时间		负载电容 105		5		ms

2.8 温度传感器特性参数

V_{DD}=3V, T=25°C

表 2-8 温度传感器特性参数

特性	符号	条件	最小	典型	最大	单位
TS 有效工作电压	V _{TS}		2.5	3	5.5	V
温度检测精度		T=-40°C~85°C	-	±1	-	°C

2.9 EEPROM 特性参数

V_{DD}=3V, T=25°C

表 2-9 EEPROM 特性参数

特性	符号	条件	最小	典型	最大	单位
读操作电压	V _{EEED}	T=-40°C~85°C	2.0		5.5	V



写操作电压	V _{EEWR}	T=-40°C~85°C	2.0		5.5	V
写操作电流	I _{EEWR}	T=-40°C~85°C		2		mA
静态电流	I _{EESTP}	T=25°C		1		μA
单地址擦除时间	T _{EEER}	VDD=2.0V~5.5V, T=-40°C~85°C		2.5		ms
单地址写入时间	T _{EEWR}	VDD=2.0V~5.5V, T=-40°C~85°C		1.5		ms
擦写次数		VDD=3V, T=25°C	10,000			cycle
数据保持时间			10			year

shomcu.com



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	$R[7] \rightarrow C; R[6:0] \rightarrow A[7:1]; C \rightarrow A[0]$	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	$R[0] \rightarrow C; R[7:1] \rightarrow A[6:0]; C \rightarrow A[7]$	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	$A \rightarrow R$	1	1	-
MOVAR	R	将 R 存入 A	$R \rightarrow A$	1	1	Z
MOVR	R	将 R 存入 R	$R \rightarrow R$	1	1	Z
CLRA		将 A 清零	$0 \rightarrow A$	1	1	Z
CLRR	R	将 R 清零	$0 \rightarrow R$	1	1	Z
INCA		A 自加 1	$A+1 \rightarrow A$	1	1	-
INCR	R	R 自加 1	$R+1 \rightarrow R$	1	1	Z
INCAR	R	R 加 1, 结果存入 A	$R+1 \rightarrow A$	1	1	Z
DECA		A 自减 1	$A-1 \rightarrow A$	1	1	-
DECR	R	R 自减 1	$R-1 \rightarrow R$	1	1	Z
DECAR	R	R 减 1, 结果存入 A	$R-1 \rightarrow A$	1	1	Z
JZA		A 自加 1: 结果为 0 则跳过下一条指令	$A+1 \rightarrow A$: 结果为 0 则 $PC+2 \rightarrow PC$	1/2	1	-
JZR	R	R 自加 1: 结果为 0 则跳过下一条指令	$R+1 \rightarrow R$: 结果为 0 则 $PC+2 \rightarrow PC$	1/2	1	-
JZAR	R	R 加 1, 结果存入 A: 结果为 0 则跳过下一条指令	$R+1 \rightarrow A$: 结果为 0 则 $PC+2 \rightarrow PC$	1/2	1	-
DJZA		A 自减 1: 结果为 0 则跳过下一条指令	$A-1 \rightarrow A$: 结果为 0 则 $PC+2 \rightarrow PC$	1/2	1	-
DJZR	R	R 自减 1: 结果为 0 则跳过下一条指令	$R-1 \rightarrow R$: 结果为 0 则 $PC+2 \rightarrow PC$	1/2	1	-
DJZAR	R	R 减 1, 结果存入 A: 结果为 0 则跳过下一条指令	$R-1 \rightarrow A$: 结果为 0 则 $PC+2 \rightarrow PC$	1/2	1	-
BCLR	R,b	将 R 的第 b 位清 0	$0 \rightarrow R[b]$	1	1	-
BSET	R,b	将 R 的第 b 位置 1	$1 \rightarrow R[b]$	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 $PC+2 \rightarrow PC$	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 $PC+2 \rightarrow PC$	1/2	1	-
ADDAI	I	I 和 A 相加, 结果存入 A	$I+A \rightarrow A$	1	1	C,DC,Z
ADCAI	I	I 和 A 相加 (带 C 标志), 结果存入 A	$I+A+C \rightarrow A$	1	1	C,DC,Z
ISUBAI	I	I 和 A 相减, 结果存入 A	$I-A \rightarrow A$	1	1	C,DC,Z
ISBCAI	I	I 和 A 相减 (带 C 标志), 结果存入 A	$I-A/C \rightarrow A$	1	1	C,DC,Z
ASUBAI	I	A 和 I 相减, 结果存入 A	$A-I \rightarrow A$	1	1	C,DC,Z
ASBCAI	I	A 和 I 相减 (带 C 标志), 结果存入 A	$A-I/C \rightarrow A$	1	1	C,DC,Z
ANDAI	I	I 和 A 与操作, 结果存入 A	$I \text{ and } A \rightarrow A$	1	1	Z
ORAI	I	I 和 A 或操作, 结果存入 A	$I \text{ or } A \rightarrow A$	1	1	Z
XORAI	I	I 和 A 异或操作, 结果存入 A	$I \text{ xor } A \rightarrow A$	1	1	Z
MOVAI	I	将 I 存入 A	$I \rightarrow A$	1	1	-
CALL	K	子程序调用	$PC+1 \rightarrow TOS; K \rightarrow PC[12:0]$	2	1	-
GOTO	K	无条件跳转	$K \rightarrow PC[12:0]$	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:

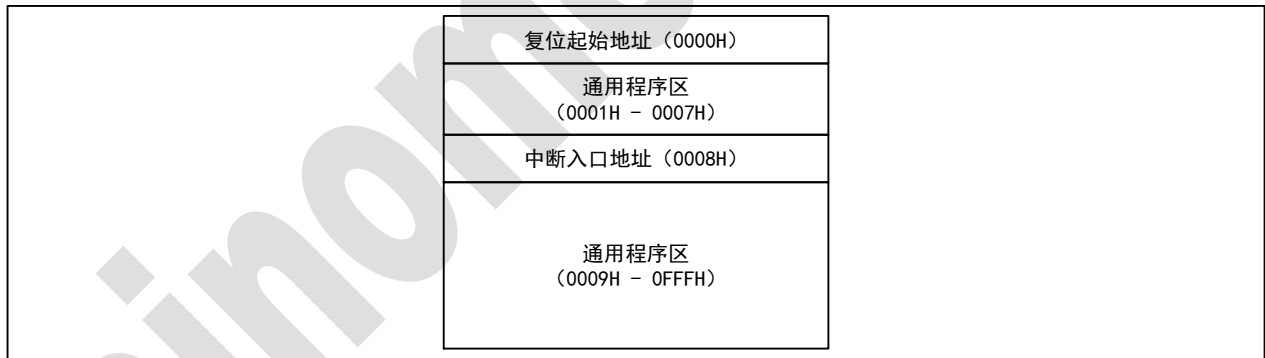


图 3-1 地址映射

程序存储器支持间接寻址, 可通过寄存器 INDF3 访问地址为 (FSR1×256+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 (512 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, 而 GPR1 分为 2 个存储器组, 寻址时需先通过寄存器位选择相应分组。

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。

表 3-2 数据存储区分组示意图

地址	类型	BANK 0 (RBS=0)	BANK 1 (RBS=1)
000H-0FFH	GPR0	通用数据存储器区 0	
100H-17FH	GPR1	通用数据存储器区 1 (存储器组 0)	通用数据存储器区 1 (存储器组 1)
180H-1FFH	SFR	特殊功能寄存器区	

注: 仅 GPR1 寻址受 RBS 影响, 而 GPR0 和 SFR 则无论 RBS 为何值都将寻址到同一地址。

表 3-3 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	HIRCCAL	INTE1	INTF1
190H-197H		IOP0	OEP0	PUP0	PDP0	IOP1	OEP1	PUP1	PDP1
198H-19FH						P0KBCR			LVDCR
1A0H-1A7H		T0CR	T0CNTH	T0CNTL	T0LOADH	T0LOADL			
1A8H-1AFH		T1CR	T1CNTH	T1CNTL	T1LOADH	T1LOADL	T1DATAH	T1DATAL	
1B0H-1B7H		T2CR	T2CNT	T2LOAD	T2DATA	T2DATB	T2DATC	PWM2CR0	PWM2CR1
1B8H-1BFH		PWM2CR2						P0ADCR	P1ADCR
1C0H-1C7H		ADCR0	ADCR1	ADCR2	ADCR3	ADCR4	ADRH	ADRM	ADRL
1C8H-1CFH		IICCR0	IICCR1	IICSR	IICDR	IICAR	IICAAR	IICAMR	
1D0H-1D7H		URTCR0	URTCR1	URTCR2	URTSR	URTBR	URTDR	URTAR	URTMR



1D8H-1DFH		IAPCR0	IAPCR1	IAPPR	IAPSAR	IAPPAR	IAPWAR	IAPDRH	IAPDRL
1E0H-1F7H	保留	保留							
1F8H-1FFH	保留							DBGCR	DBGPR

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 3-4 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	

注：无论采用哪种寻址方式访问 100H-17FH 地址的 GPR1，均需保证 RBS 位选择正确。

直接寻址方式，是指以指令的低 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

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 (FSR1×256+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 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	-	-	RBS	-	-	Z	DC	C
R/W	-	-	R/W	-	-	R/W	R/W	R/W
初始值	-	-	0	-	-	X	X	X

BIT[5] RBS – 数据存储区 GPR1 存储器分组选择位

- 0：选择存储器组 0，即地址 100H~17FH 映射至第 0 组存储器；
- 1：选择存储器组 1，即地址 100H~17FH 映射至第 1 组存储器；

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:



表 3-5 用户配置字

符号	功能说明
HIRCFS	HIRC 振荡频率选择： F _{HIRC} =8MHz; 4MHz; 2MHz; 1MHz;
FCPUS	高频时钟下 F _{CPUS} 分频选择： F _{CPUS} =F _{HOSC} /2; F _{CPUS} =F _{HOSC} /4; F _{CPUS} =F _{HOSC} /8; F _{CPUS} =F _{HOSC} /16;
LOSCS	系统低频时钟选择： F _{LOSC} =内部低频 RC 振荡器时钟 F _{LIRC} ; F _{LOSC} =外部 32768Hz 晶体振荡器时钟 F _{XTAL} ;
XTDRVS	XTAL 驱动能力选择： 弱驱动（功耗最低，起振最慢）； 1 倍增强驱动； 2 倍增强驱动； 4 倍增强驱动（功耗最大，起振最快）；
RSTEN	RST 外部复位端口设置： P10 为外部复位脚； P10 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 _{CPUS} 决定的工作电压特性） 1.8V; 2.0V; 2.4V;
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@5V,23ms@3V; 64ms@5V,93ms@3V; 256ms@5V,372ms@3V; 1024ms@5V,1489ms@3V;
SMTVS	端口施密特阈值选择： 2.0V/0.8V; 0.8V _{DD} /0.2V _{DD} ;
DRVS	端口输出驱动电流选择： 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；
FLASHMOD	FLASH（程序区）模式选择： FLASH 区不划分保护区和非保护区，所有存储空间均允许 IAP 操作； FLASH 区划分保护区和非保护区，保护区存储空间不允许 IAP 操作；
FLASHPGS	FLASH 保护区地址选择： 页地址 N（N=0~511），表示 FLASH 第 0 页至第 N 页存储空间为保护区；
FLASHIAPEN	FLASH（程序区）IAP 设置（仅限无保护区）： FLASH 区屏蔽 IAP 功能； FLASH 区支持 IAP 功能；
INFOIAPEN	INFO（配置区）IAP 设置： INFO 区屏蔽 IAP 功能； INFO 区支持 IAP 功能；



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，而 SDADC、IIC、UART、IAP 模块则固定在外设高频时钟 FHCLK 下工作。

系统高频时钟 FHOSC 固定为内部高频 RC 振荡器 HIRC (8MHz/4MHz/2MHz/1MHz) 时钟 FHIRC；

系统低频时钟 FLOSC 可通过配置字 LOSCS 选择以下时钟：

- 内部低频 RC 振荡器 LIRC (32KHz) 时钟 FLIRC；
- 外部低频晶体振荡器 XTAL (32768Hz) 时钟 FXTAL；

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。

外设高频时钟 FHCLK 固定为内部高频 RC 振荡器 HIRC (8MHz) 时钟 FHIRC (8MHz)。

WDT (看门狗) 电路的时钟源固定为内部低频 RC 振荡器 LIRC。

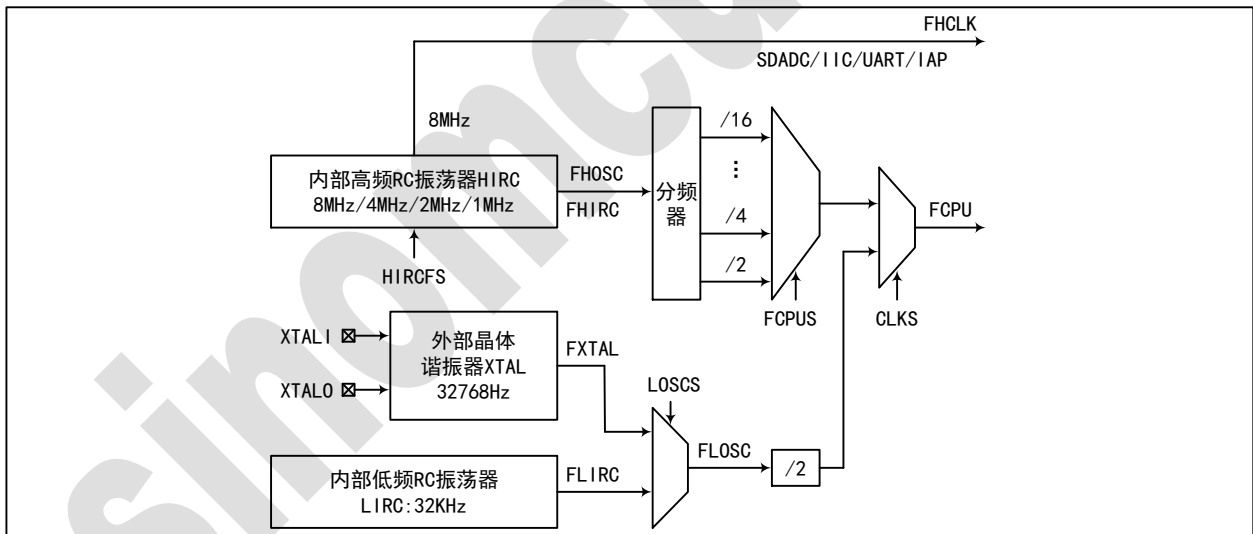


图 4-1 系统时钟框图

4.1 内部高频 RC 振荡器

芯片内置 1 个振荡频率可通过配置字 HIRCFS 选择 (8MHz/4MHz/2MHz/1MHz) 的高精度 HIRC 振荡器，可用作系统高频时钟源，其中 8MHz 时钟可作为外设高频时钟 FHCLK 用作 SDADC/IIC/UART/IAP 模块的时钟源。



4.2 内部低频 RC 振荡器

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

4.3 外部晶体振荡器

芯片支持外接 32768Hz 晶体振荡器作为系统低频时钟源。在通过配置字 LOSCS 选择外部晶体振荡器 XTAL 作为系统低频时钟源后，端口的 I/O 功能及其他复用功能均将无效。

外接晶振的实际应用中，晶振两端的对地电容 CG/CD 是必需的（需外接对地电容）。用户应使晶振离 XTALI/XTALO 引脚的距离尽可能短，这样有助于振荡器的起振和振荡稳定性。下表是典型频率晶振选用电容 CG/CD 的推荐值和相应最低起振电压参考值：

表 4-1 典型晶振负载电容及最低起振电压

晶振频率 (Hz)	电容 CG/CD (pF)	最低起振电压 (V)
32768	10	1.8

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

4.4 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 4-2 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：

1、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；



2、SDADC/IIC/UART/IAP 中有模块开启时，HIRC 将开始工作而不受系统工作模式和寄存器位 HFEN 影响；

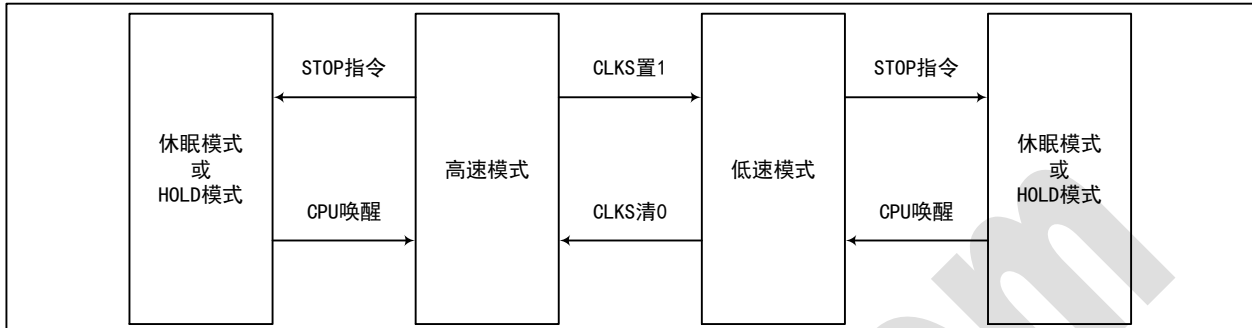


图 4-2 工作模式切换示意图

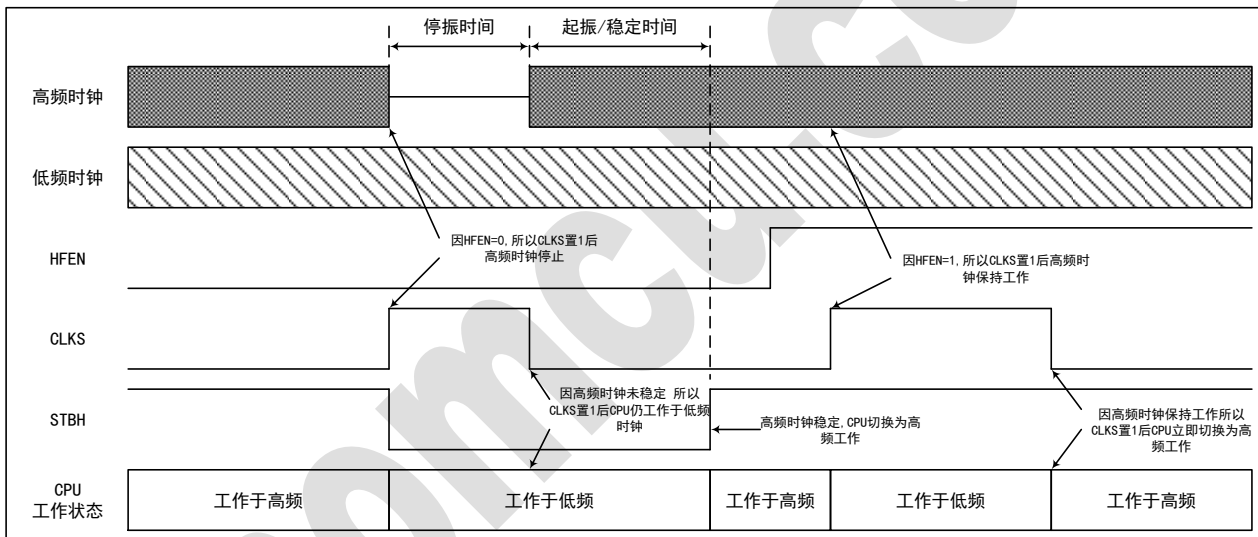


图 4-3 高低频时钟切换时序图

4.4.1 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	LOSCOE	CLKS	LFEN	HFEN
R/W	-	-	R	R	R/W	R/W	R/W	R/W
初始值	-	-	X	1	0	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[3] LOSCOE – LOSCOUT 端口输出使能位
0: 禁止端口输出系统低频时钟信号;
1: 允许端口输出系统低频时钟信号;
- 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;

4.5 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);
- UART 接收完成中断请求发生 (若低功耗模式下 UART 及其时钟源保持继续工作);



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，避免因引脚浮空而产生漏电流或非预期的中断唤醒；

Sinomcu.com



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；
- 软件复位；

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 9ms）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。

5.6 软件复位

可通过软件复位控制位 SWRST 置“1”复位芯片。

5.6.1 DEBUG 控制寄存器

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

BIT[7] SWRST – 软件复位控制位

0: 不复位芯片；

1: 复位芯片；

BIT[1] DBG1EN – PCK1/PDT1 端口编程功能使能位

BIT[0] DBG0EN – PCK0/PDT0 端口编程功能使能位



6 I/O 端口

6.1 通用 I/O 功能

芯片的输入/输出端口包括一组 6 位端口 P0，和一组 8 位端口 P1。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	-	-	P05D	P04D	P03D	P02D	P01D	P00D
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] P0nD – P0n 端口数据位 (n=5-0)

	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)

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	-	-	P05OE	P04OE	P03OE	P02OE	P01OE	P00OE
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] P0nOE – P0n 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]
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；



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	-	-	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
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] P0nPU – P0n 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]
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;

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	-	-	P05PD	P04PD	P03PD	P02PD	P01PD	P00PD
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] P0nPD – P0n 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]
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;

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	-	-	P05DC	P04DC	P03DC	P02DC	P01DC	P00DC
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] P0nDC – P0n 端口数字功能控制位 (n=5-0)

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

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
P1ADCR	P17DC	P16DC	P15DC	P14DC	P13DC	P12DC	P11DC	P10DC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P1nDC – P1n 端口数字功能控制位 (n=7-0)

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



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@5V，23ms/93ms/372ms/1489ms@3V。

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 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；
- 支持溢出中断和溢出唤醒功能；

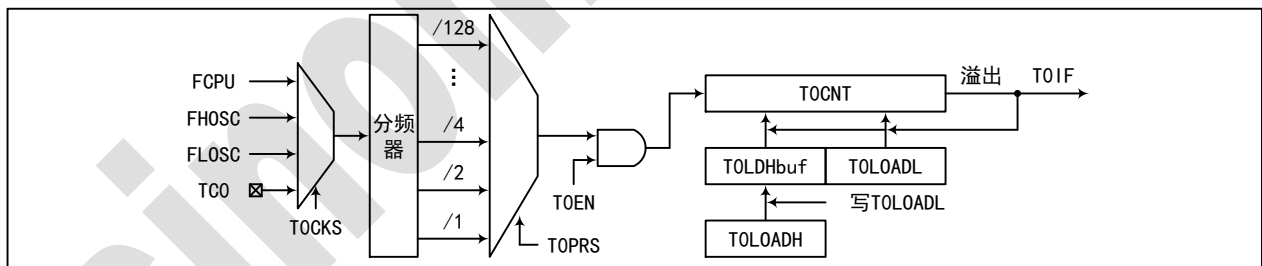


图 7-1 16 位定时器 T0 示意图

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 TOCNT 的计数时钟（上升沿计数）。写 TOCNT 将清零预分频计数器，而预分频比保持不变。

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

T0EN=0 时，TOCNT 保持不变，写重载寄存器 T0LOAD 将立即载入 TOCNT；T0EN=1 时，TOCNT 递减计数，计数到



0 的时钟结束后产生溢出信号并触发中断，中断标志 T0IF 将被置 1，同时 T0 自动将当前 T0LOAD 值载入 T0CNT 并重新开始计数。

16 位 T0LOAD 的高字节 T0LOADH 配有缓冲器 (T0LDHbuf)，写 T0LOADL 时会同时将 T0LOADH 内容载入该缓冲器中，因此调整 T0LOAD 值时需先写 T0LOADH 再写 T0LOADL。此时若 T0EN=0，则会同时再将[缓冲器:T0LOADL]载入 T0CNT；若 T0EN=1，则会在 T0 溢出后才将[缓冲器:T0LOADL]载入 T0CNT。

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]
T0CR	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	F _{CPU}
01	F _{HOSC}
10	F _{LOSC}
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]
T0CNTH	T0CNT15	T0CNT14	T0CNT13	T0CNT12	T0CNT11	T0CNT10	T0CNT9	T0CNT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



初始值	1	1	1	1	1	1	1	1
-----	---	---	---	---	---	---	---	---

BIT[7:0] TOCNT[15:8] – T0 计数器高 8 位，为可读写的递减计数器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
TOCNTL	TOCNT7	TOCNT6	TOCNT5	TOCNT4	TOCNT3	TOCNT2	TOCNT1	TOCNT0
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] TOCNT[7:0] – T0 计数器低 8 位，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]
T0LOADH	T0LOAD15	T0LOAD14	T0LOAD13	T0LOAD12	T0LOAD11	T0LOAD10	T0LOAD9	T0LOAD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T0LOAD[15:8] – T0 重载寄存器高 8 位，用于设置 T0 的计数周期

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
T0LOADL	T0LOAD7	T0LOAD6	T0LOAD5	T0LOAD4	T0LOAD3	T0LOAD2	T0LOAD1	T0LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

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

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。

7.3 定时器 T1

定时器 T1 为 12 位定时/计数器，包含 1 个 12 位递减计数器、可编程预分频器、控制寄存器、12 位重载寄存器和 12 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 12 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；

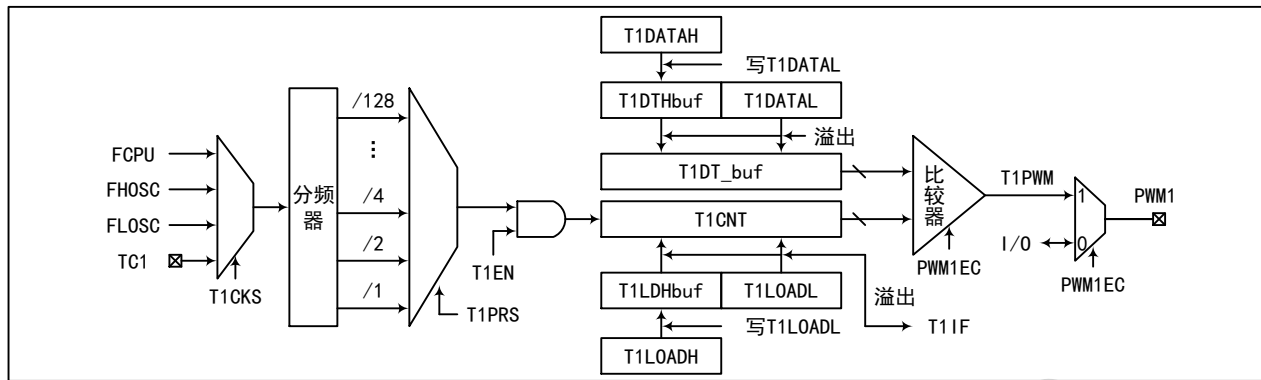


图 7-2 12 位定时器 T1 示意图

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。

12 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。

12 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。

12 位 T1DATA 配有 1 个高位缓冲器和 1 个 12 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)

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]
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	F _{CPU}
01	F _{HOSC}
10	F _{LOSC}
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

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	-	-	-	-	T1CNT11	T1CNT10	T1CNT9	T1CNT8
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	1	1	1	1

BIT[3:0] T1CNT[11:8] – T1 计数器高 4 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	-	-	-	-	T1LOAD11	T1LOAD10	T1LOAD9	T1LOAD8
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	1	1	1	1

BIT[3:0] T1LOAD[11:8] – T1 重载寄存器高 4 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。

7.3.4 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	-	-	-	-	T1DATA11	T1DATA10	T1DATA9	T1DATA8
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3:0] T1DATA[11:8] – T1 比较寄存器高 4 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 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；

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。

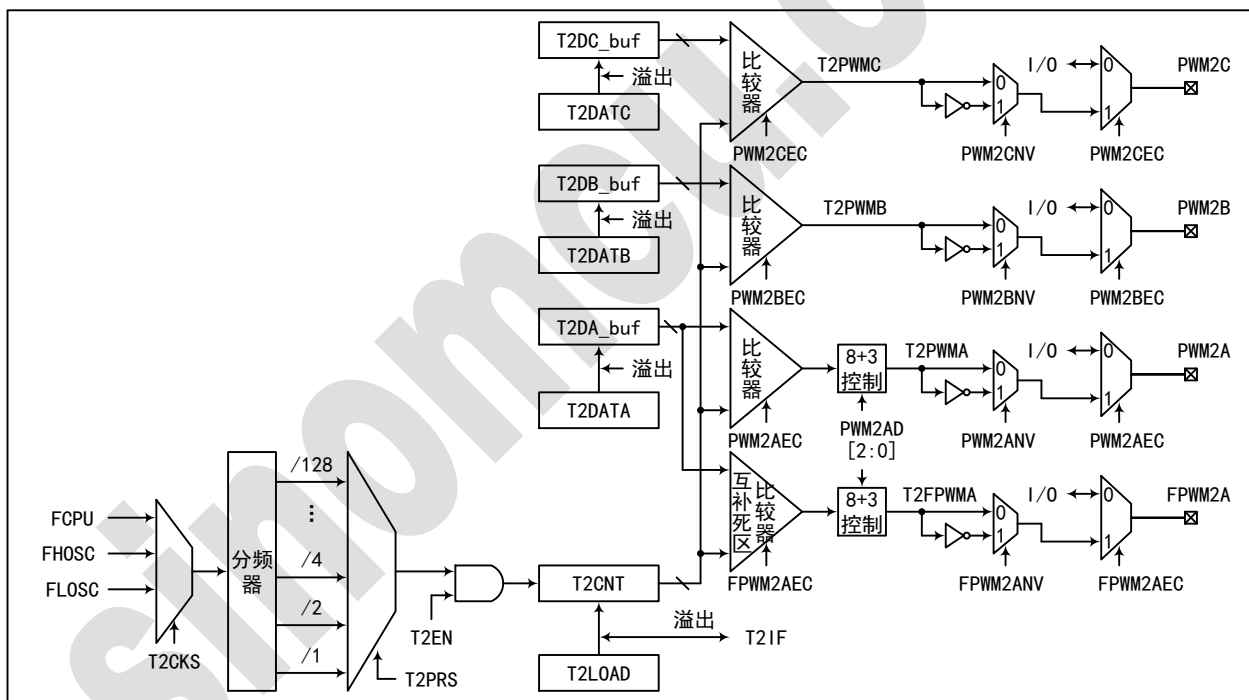


图 7-3 8 位定时器 T2 示意图

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 (T3 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 $[(T2DATA) + (5/16)] / (T2LOAD + 1)$ ；

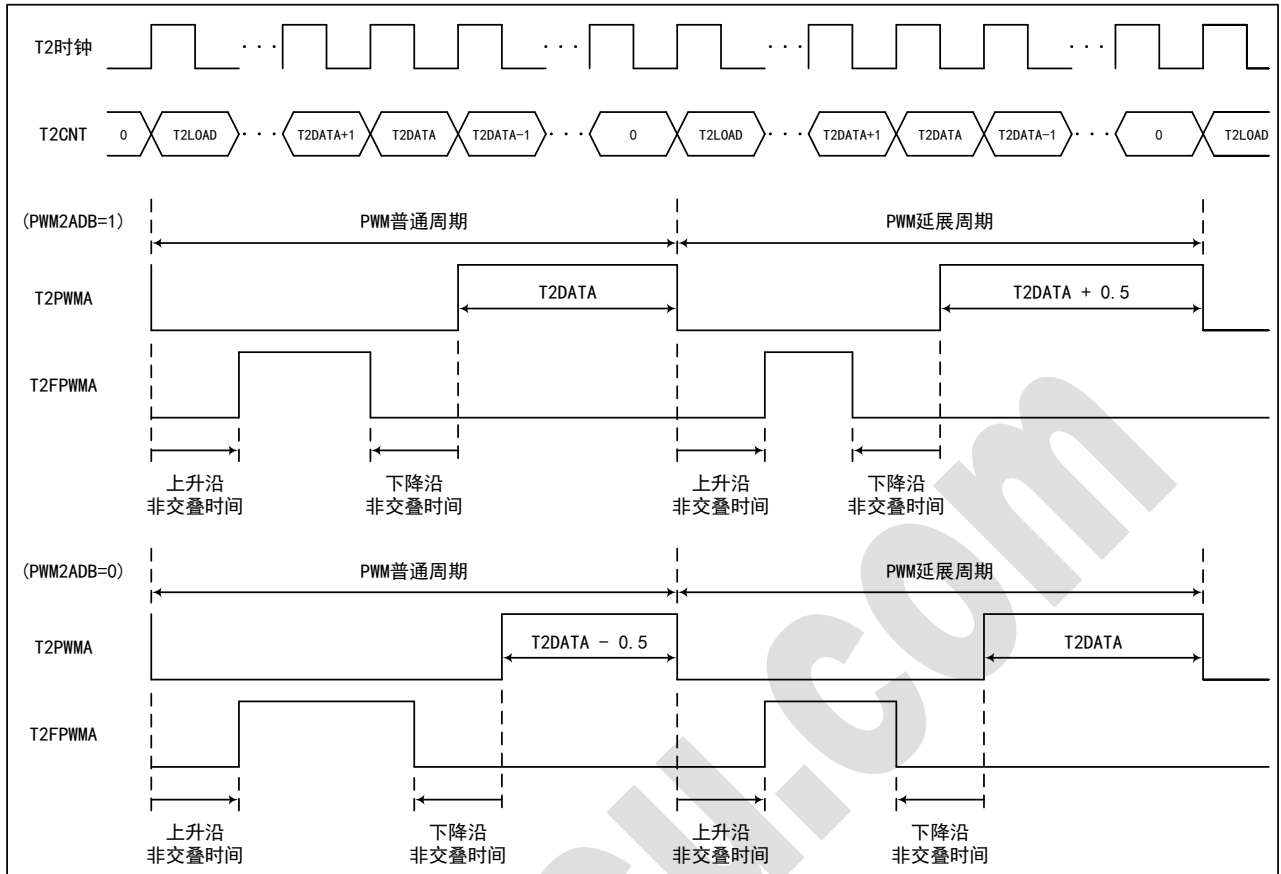


图 7-4 PWM2ADB 及 8+3 模式波形示意

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；

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	-	-	PWM2CEC	PWM2BEC
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] PWM2CNV – PWM2C 端口取反控制位

- 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[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
00 0010	3 个计数时钟周期	3 个计数时钟周期
---	---	---
11 1110	63 个计数时钟周期	63 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



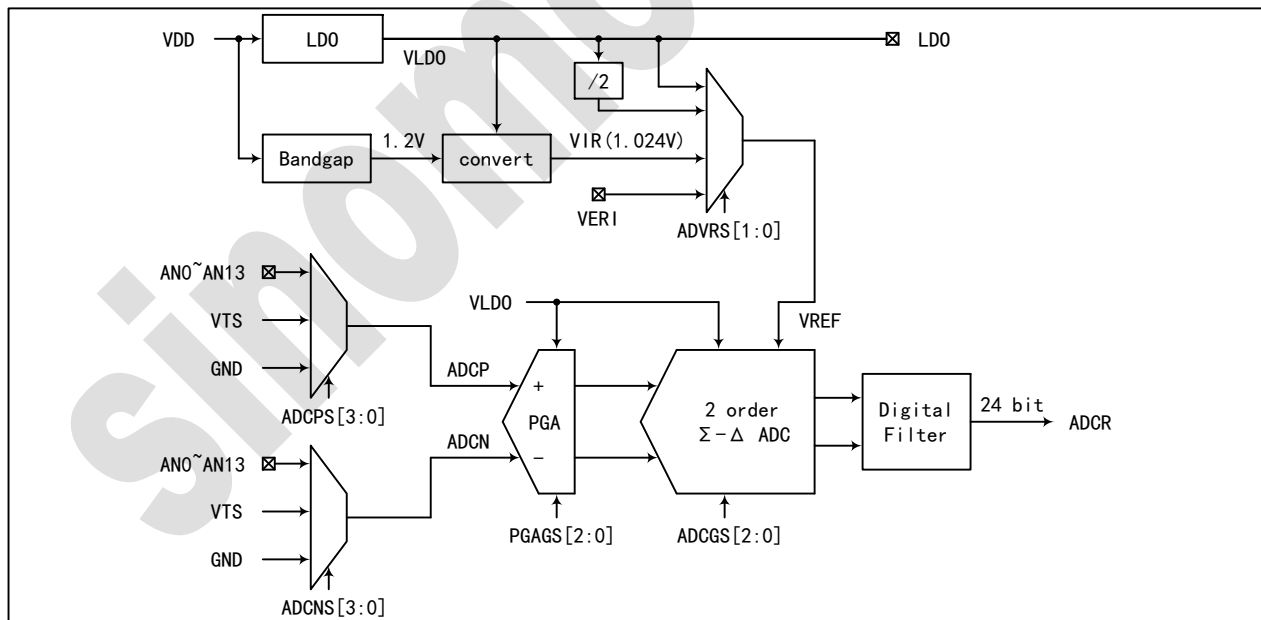
8 模数转换器 SD-ADC

8.1 ADC 概述

芯片内置 1 个 24 位高精度差分输入 Σ - Δ 型的模数转换器 ADC。

- 工作电压为内部 LDO 输出电压 VLDO (可输出 VDD 电压)；
- 正/负端输入共支持 14 路对 GND 差分的外部单输入通道 (AN0~AN13)，也可组成多对外部差分输入通道；
- 内置前级增益可编程放大电路 PGA，高输入阻抗；
- 参考电压可选: VLDO、VLDO/2、内部参考电压 VIR (可从端口 VIRO 输出)、外部参考电压 VER (VERI 输入)；
- ADC 时钟: 外设高频时钟 FHCLK (8MHz) 的 8/24 分频；
- 差分输入放大倍数 (含前级 PGA 放大) 为 0.25/0.5/1/2/4/8/16/32/64/128/256 可选；
- 过采样率 64~32768 可选，采样速率 10Hz@32768~15.626KHz@64 可选；
- 支持 50Hz/60Hz 工频滤波@采样速率=10Hz；
- 内置温度传感器，输出电压 VTS 可作为 ADC 的正端输入信号；

ADC 模块可通过寄存器位 ADEN 开启，通过 ADCPS/ADCNS 选择转换的正负端模拟输入通道，通过 ADCKS 选择转换时钟，通过 ADOVS 选择过采样率，通过 PGAGS 选择前级 PGA 放大倍数，通过 ADCGS 选择 ADC 内部放大倍数。AD 转换使能位 ADCONEN 为 0 时，转换暂停，转换数据重置；ADCONEN 为 1 时，AD 按位串行转换，转换结果存入 ADRH/ADRM/ADRL 中，同时中断标志 ADIF 置 1 触发 ADC 中断。



24 位转换结果，最高位为符号位，低 23 位为数据位。当差分输入正端电压不小于负端电压时，符号位为“0”，数据位为差分输入电压 AD 转换的正向数值；当差分输入正端电压小于负端电压时，符号位为“1”，数据位为差分输入电压 AD 转换的补码数值。



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	ADCONEN	VIREN	VIROE	LDOEN	LDOVS2	LDOVS1	LDOVS0
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] ADEN – ADC 使能位

- 0: 关闭 ADC;
- 1: 开启 ADC;

BIT[6] ADCONEN – AD 转换使能位

- 0: 不使能;
- 1: 使能;

BIT[5] VIREN – 内部参考电压 VIR 使能位

- 0: 关闭 VIR 功能;
- 1: 开启 VIR 功能;

BIT[4] VIROE – VIRO 端口输出使能位

- 0: 禁止端口输出内部 VIR 电压;
- 1: 允许端口输出内部 VIR 电压;

BIT[3] LDOEN – LDO 使能位

- 0: 关闭 LDO 功能 (LDO 输出固定为 GND);
- 1: 开启 LDO 功能 (LDO 输出电压 VLDO 由 LDOVS 决定);

BIT[2:0] LDOVS[2:0] – LDO 输出电压选择位

LDOVS[2:0]	LDO 输出电压
0XX	VDD
100	2.5V
101	2.8V
110	3.0V
111	3.3V

注:

- 1、对于无 LDO 引脚输出封装, 只能选择 0XX(VDD)
- 2、使用 LDO 时必须外挂 0.1u~10uF 电容

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



ADCR1	ADVRS1	ADVRS0	PGAGS2	PGAGS1	PGAGS0	ADCGS2	ADCGS1	ADCGS0
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] ADVRS[1:0] – ADC 参考电压选择位

ADVRS[1:0]	ADC 参考电压
00	V _{LDO}
01	0.5 × V _{LDO}
10	内部 1.024V
11	外部 VER1 输入电压

BIT[5:3] PGAGS[2:0] – 前级 PGA 放大倍数选择位

PGAGS[2:0]	PGA 放大倍数
000	PGA 关闭, 输入旁路
001	1
010	2
011	4
100	8
101	16
110	32
111	64

BIT[2:0] ADCGS[2:0] – ADC 内部放大倍数选择位

ADCGS[2:0]	ADC 放大倍数
000	0.25
001	0.5
010	1
011	2
1XX	4

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADCR2	-	-	TSMOD	ADCKS	ADOVS3	ADOVS2	ADOVS1	ADOVS0
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] TSMOD – 温度传感器模式选择位

0: 模式 0;

1: 模式 1;

注: 模式 0 和模式 1 的电气特性不同, 需根据实际应用环境下的芯片实测结果选择更优的模式。



BIT[4] ADCKS – ADC 转换时钟选择位

- 0: $F_{DADC} = F_{HIRC}/8$ (高速采样, 1MHz);
 1: $F_{DADC} = F_{HIRC}/24$ (低速采样, 333KHz);

BIT[3:0] ADOVS[3:0] – ADC 过采样率选择位

ADOVS[3:0]	ADC 过采样率	ADC 采样速率	
		$F_{HIRC}/8$ (高速采样)	$F_{HIRC}/24$ (低速采样)
0000	64	15626Hz@64	5208Hz@64
0001	128	7813Hz@128	2604Hz@128
0010	256	3906Hz@256	1302Hz@256
0011	512	1953Hz@512	651Hz@512
0100	1024	977Hz@1024	326Hz@1024
0101	2048	488Hz@2048	163Hz@2048
0110	4096	244Hz@4096	81Hz@4096
0111	8192	122Hz@8192	41Hz@8192
1000	16384	61Hz@16384	20Hz@16384
1001	32768	30Hz@32768	10Hz@32768
1010-1111	保留	30Hz@32768	10Hz@32768

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADCR3	CHOPEN	CHOPPGA	CHOPBG	CHOPVIR	CHOPV2V	CHOPFS2	CHOPFS1	CHOPFS0
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] CHOPEN – 斩波总使能位

- 0: 关闭所有斩波;
 1: 允许斩波;

BIT[6] CHOPPGA – PGA 斩波使能位

- 0: 关闭 PGA 斩波;
 1: 开启 PGA 斩波;

BIT[5] CHOPBG – BG 斩波使能位

- 0: 关闭 BG 斩波;
 1: 开启 BG 斩波;

BIT[4] CHOPVIR – VIR 斩波使能位



- 0: 关闭 VIR 斩波;
1: 开启 VIR 斩波;

BIT[3] CHOPV2V – 内部信号 V2V 斩波使能位

- 0: 关闭 V2V 斩波;
1: 开启 V2V 斩波;

BIT[2:0] CHOPFS[2:0] – CHOP 频率选择位

CHOPFS[2:0]	CHOP 频率
000	F _{DADC} /8
001	F _{DADC} /16
010	F _{DADC} /32
011	F _{DADC} /64
100	F _{DADC} /128
101	F _{DADC} /256
110	F _{DADC} /512
111	F _{DADC} /1024

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADCR4	ADCPS3	ADCPS2	ADCPS1	ADCPS0	ADCNS3	ADCNS2	ADCNS1	ADCNS0
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[7:4] ADCPS[3:0] – ADC 正端输入选择位

ADCPS[3:0]	ADC 正端输入
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	温度传感器输出电压 V_{TS}
1111	GND

BIT[3:0] ADCNS[3:0] – ADC 负端输入选择位

ADCNS[3:0]	ADC 负端输入
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	-
1111	GND

注：若使用内部温度传感器功能，需正端输入配置 $ADCPS[3:0]=1110b$ （温度传感器输出电压 V_{TS} ），负端输入配置 $ADCNS[3:0]=1111b$ （GND）。

8.2.2 ADC 转换结果寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
ADRH	ADR23	ADR22	ADR21	ADR20	ADR19	ADR18	ADR17	ADR16
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[23:16] – 24 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]
ADRM	ADR15	ADR14	ADR13	ADR12	ADR11	ADR10	ADR9	ADR8
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[15:8] – 24 位 ADC 转换结果中 8 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] – 24 位 ADC 转换结果低 8 位

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) 设置 ADCKS，选择适当的转换时钟；
- (4) 设置 PGAGS 和 ADCGS，选择适当的放大倍数；
- (5) 设置 ADOVS，选择适当的过采样率和采样速率；
- (6) 设置 ADVRS，选择适当的参考电压；
- (7) ADEN 置 1，使能 ADC 模块；若需 LDO 输出电压为 ADC 工作电压则 LDOEN 置 1 开启 LDO；
- (8) 设置 ADCPS 和 ADCNS，选择 ADC 的正/负端输入通道；
- (9) 延时等待电路稳定后，ADRST 写 1，启动 AD 转换；
- (10) 等待 ADIF 硬件置 1（或利用 ADC 中断），然后清 0 标志位 ADIF。连续执行（10）2 次；
- (11) 第 3 次等待 ADIF 硬件置 1（或利用 ADC 中断）；
- (12) 清 0 标志位 ADIF，读取 ADC 转换结果（ADRH/ADRM/ADRL）；
- (13) 若 ADC 所有设置及输入通道保持不变，则重复执行（11）~（12），对同一通道进行多次转换；若需调整部分设置或变更输入通道，则从（1）开始重新转换；

8.4 ADC 应用注意

8.4.1 PGA 差分输入/输出电压超范围控制

输入共模电压 $V_{incom} = (ADCP + ADCN) / 2$

输入差分电压 $V_{indiff} = (ADCP - ADCN) / 2$

输入电压需满足

当 $V_{incom} \leq 1/2V_{LDO}$ ， $V_{incom} > |V_{indiff}| * PGA_gain$ （下限控制）

当 $V_{incom} > 1/2V_{LDO}$ ， $V_{LDO} - V_{incom} > |V_{indiff}| * PGA_gain$ （上限控制）

否则，PGA 放大后将超出输出量程，造成输出信号失真。

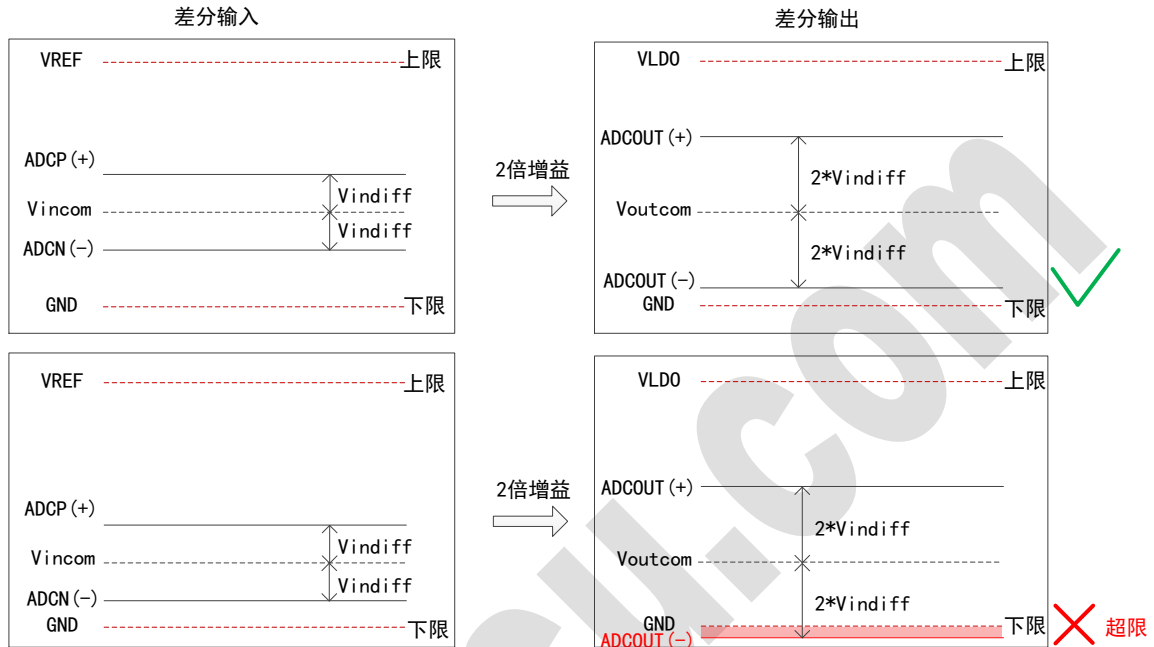
若 PGA 输入一端选择 GND，那 PGA 为单端应用接法，根据上述条件公式，必须配置 PGA 为 1 倍增益或 Bypass；否则输出信号失真，增益计算与规格不符。



举例：

正端差分输入 ADCP=400mv, 负端差分输入 ADCN=200mv, 共模电压为 300mv, 差分电压 100mv, 若配置增益放大 2 倍, 正端差分输出是 500mv, 负端差分输出是 100mv, 此时没有超量程, 如果超过 2 倍放大, 就会超量程。

注：上述为 PGA 输出范围控制, 若接入 ADC 差分输入, 为保证 ADC 转换精度, 还需满足 ADC 差分输入范围要求, 参见 ADC 电气参数章节。



8.5 温度传感器

温度传感器可以用来测量器件的结温(TJ)。

通过配置正端输入配置 ADCPS[3:0]= 1110b (温度传感器输出电压 V_{TS}), 负端输入配置 ADCNS[3:0]=1111b (GND), 内部温度传感器连接至 SDADC 差分输入通道, 可将传感器的输出电压转换为数字值。

温度传感器在使能 ADC 时硬件自动开启 (ADCEN=1)。

温度传感器输出电压随温度呈线性变化, 但由于工艺的不同, 其特性在不同芯片之间可能会有很大的变化。为了提高温度传感器的精度(特别是绝对温度测量), 在出厂测试时, 会测量每颗芯片的校准值, 并存储在 info 区。有关其他信息, 请参阅特定芯片数据手册。

读取温度流程

- 1、 正端输入配置 ADCPS[3:0]= 1110b (温度传感器输出电压 V_{TS}), 负端输入配置 ADCNS[3:0]=1111b (GND), 内部温度传感器连接至 SDADC 差分输入通道。
- 2、 开启 PGA, 且 PGA 配置 1 倍放大; 过采样率推荐至少大于 256; CHOP 打开, 且只 CHOP PGA。
- 3、 参考 8.3 配置 ADC, 并启动 ADC 转换。
- 4、 结果寄存器 (ADRH/ADRM/ADRL) 读取 ADC 转换数据 ADC_Data。

$$\text{计算温度: } T_{\text{ADC_Data}}(^{\circ}\text{C}) = (\text{VREF_Actual} / \text{FULL_SCALE} \times \text{ADC_Data} - b) / k$$

- 5、 从出厂区读出 ADC_Temp1、Sens_Temp1 为出厂校准值; 详细参考下表

$$\text{计算校准点温度(芯片): } T_{\text{ADC_Temp1}}(^{\circ}\text{C}) = (\text{VREF_Factory} / \text{FULL_SCALE} \times \text{ADC_Temp1} - b) / k$$



计算校准点温度(环境): $TSens_Temp1(^{\circ}C) = 0.0625 \times Sens_Temp1$

计算温度 offset: $Toffset = TSens_Temp1 - TADC_Temp1$

6、修正实际温度:

实际温度 T ($^{\circ}C$) = $TADC_Data + Toffset$

其中,

--VREF_Factory 为出厂校准时配置的参考电压源, 为内部参考电压源 (ADVRS[1:0]=10b, 内部 1.024 v);

--VREF_Actual 为实际应用中参考电压电源, 由实际应用确定

--FULL_SCALE = 2^{15}

--ADC_Data, 实际 ADC 转换值

--VADC_Data = $VREF_Actual / FULL_SCALE \times ADC_DATA$

--VADC_Temp1 = $VREF_Factory / FULL_SCALE \times ADC_Temp1$

--k 为温度传感器电压平均斜率 Avg_Slope ($+0.002v/^{\circ}C$), 用户需配置 TSMOD=1);

--b 为温度传感器电压 vs 温度一次曲线常数项 (截距), $b=0.57178v$ (用户需配置 TSMOD=1)

--ADC_Temp1、Sens_Temp1 为出厂校准值, 参见数据手册描述

若实际应用 VREF_Factory 配置同 VREF_Actual, 公式可以简化为:

实际温度 T ($^{\circ}C$) = $((VADC_Data - VADC_Temp1) / Avg_slope) + TSens_Temp1$

温度传感器校准值

名称	说明	存储地址
VREF_Factory	内部参考电压源 1.024 v, 出厂测试实际值, 单位 mV, 16 进制存储	0x803A
Sens_Temp1	获取 ADC_Temp1 的常温温度值, 单位 $0.0625^{\circ}C$, 16 进制存储 举例, 读取的值 0x190(400), 则实际温度 $400 \times 0.0625^{\circ}C = 25^{\circ}C$	0x803B
ADC_Temp1	内部温度传感器通道 ADC 原始数据(常温), VREF 配置为内部参考电压源 (ADVRS[1:0]=10b, 内部 1.024 v), 16 进制存储, 保留 ADC 结果寄存器的高 16 位	0x803F

注: 以上出厂测试值, 都是在常温下, VDD=3.6V, LDO=3.0v 条件下测试获得。每个地址存储 16 位数据。



9 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 $>200\mu\text{s}$ ）后 LVD 输出才有效。

9.1.1 LVD 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
LVDCR	LVDEN	LVDVS3	LVDVS2	LVDVS1	LVDVS0	-	-	LVDF
R/W	R/W	R/W	R/W	R/W	R/W	-	-	R
初始值	0	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[6:3] LVDVS[3:0] – LVD 电压检测阈值选择位

LVDVS[3:0]	LVD 电压检测阈值	LVDVS[3:0]	LVD 电压检测阈值
0000	1.8V	1000	2.6V
0001	1.9V	1001	2.7V
0010	2.0V	1010	2.8V
0011	2.1V	1011	2.9V
0100	2.2V	1100	3.0V
0101	2.3V	1101	3.1V
0110	2.4V	1110	3.2V
0111	2.5V	1111	3.3V

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；



10 总线通讯 IIC

10.1 IIC 概述

芯片内置 1 个 IIC 总线通讯模块，支持 7 位地址编码主从模式的 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（理论最高支持 400Kbps，但受芯片及电路等影响，实际可能略低）。

10.2 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）。

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。

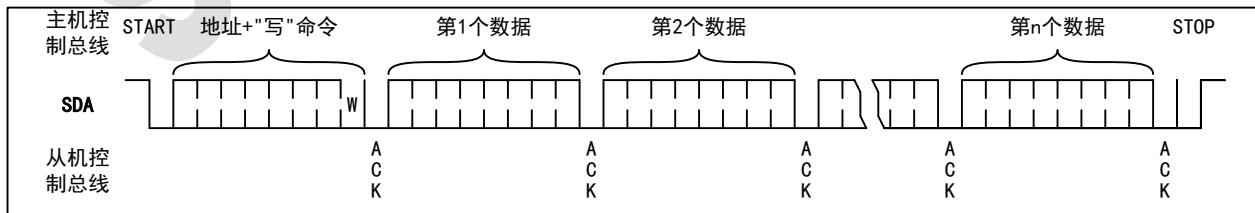


图 10-1 主机发送数据的格式

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。

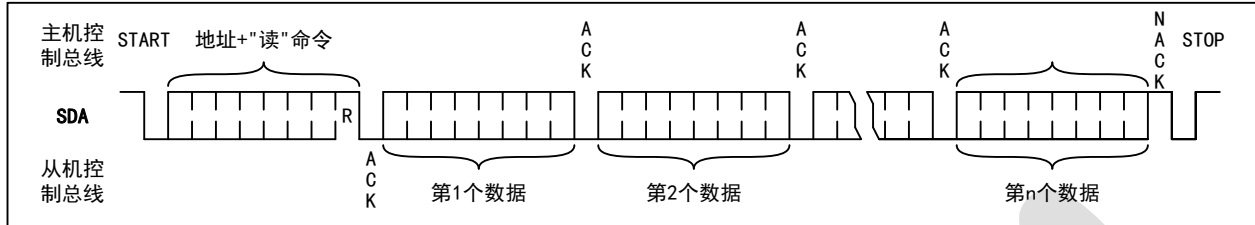


图 10-2 主机接收数据的格式

10.3 IIC 工作模式

IIC 总线接口模块支持 7 位地址编码的主从模式，可通过寄存器位 IICMOD 选择主机工作模式或从机工作模式。主机模式下，总线时钟从 SCL 端口输出，通讯速率可通过寄存器位 IICSPD 选择 100Kbps 或 400Kbps；从机模式下，总线时钟由 SCL 端口输入，通讯速率在同步采样方式 (IICSYN=0) 时需通过 IICSPD 选择 100Kbps 或 400Kbps，而在异步采样方式 (IICSYN=1) 时最高可支持 400Kbps。

主机工作模式

主机模式下，IIC 模块可单独发送地址帧（包含起始信号）、数据帧、或停止信号。

主机发送地址帧时，会将地址寄存器 IICAR 中内容作为地址（高 7 位）和命令（最低位）发送；

主机发送数据帧时，会将发送缓冲器 (IICTD_buf) 中内容作为数据发送，而在接收数据帧时，则将接收的数据存入接收缓冲器 (IICRD_buf) 中，IICTD_buf 和 IICRD_buf 均需通过数据寄存器 IICDR 进行读/写操作，写 IICDR 为写 IICTD_buf，读 IICDR 则为读 IICRD_buf。

主机发送完地址或数据帧（第 9 位接收完从机应答信号）、主机接收完数据帧（第 9 位发送完应答信号）、或主机发送完停止信号后，中断标志 IICIF 将被置 1 触发 IIC 中断。

从机工作模式

从机模式下，IIC 模块在总线上检测到 IIC 起始信号后开始接收主机发送的地址及命令（缓存在接收缓冲器中），若地址与预设的 IICAR 中的从机主地址或 IICAAR 中的从机辅地址（若使能广播地址则还包括地址 00H）匹配，则在第 9 位应答时钟期间将寄存器位 TACKS 内的应答内容发送给主机。若发送的应答内容为 NACK，则发送完成后重置 IIC 模块以等待总线上的下一次起始信号；若发送的应答内容为 ACK，则发送完成后地址匹配标志位 SADMf 将被置 1，同时 IIC 模块会将 SCL 电平拉为低电平（下一次检测到 SCL 电平上升沿时 SADMf 将被清 0），中断标志 IICIF 将被置 1 触发 IIC 中断。

清 0 中断标志 IICIF 可将 SCL 线释放，此前可通过读写命令标志位 SCMDf 准备接收或发送数据。从机接收或发送数据帧时，也会将接收的数据存入接收缓冲器 (IICRD_buf) 中或将发送缓冲器 (IICTD_buf) 中内容发送。接收/发送完数据并发送/接收完应答信号后，数据帧收发完成标志位 SDTFF 将被置 1，同时 IIC 模块会再次将 SCL 电平拉为低电平（下



一次检测到 SCL 电平上升沿时 SDFIF 将被清 0)，中断标志 IICIF 将被置 1 触发 IIC 中断。清 0 中断标志 IICIF 释放 SCL 线，可使主机继续传输数据帧或发送停止信号。

检测到总线上的停止信号后，从机模式通讯结束。

注：从机模式地址匹配时，接收到的地址信息缓存于 IICRD_buf 中，因此若使能广播地址，则可通过 IICDR 读取 IICRD_buf 中的地址信息，判断当前所响应的地址是广播地址 00H 还是预设于 IICAR 中的从机地址。

从机模式还可选择同步或异步采样方式，同步方式可增加采样准确率，异步方式则支持休眠模式下检测总线信号，当有地址匹配或接收完成时将会唤醒 CPU。同步/异步方式的切换需在 SCL 被拉住后进行，且切换后需等待 t_{HIGH} 时间后方可释放 SCL 线。

应答寄存器位

无论主机模式还是从机模式，发送应答信号时将发送寄存器位 TACKS 中内容、接收到的应答内容保存在寄存器位 RACKF 中。

10.4 IIC 时序说明

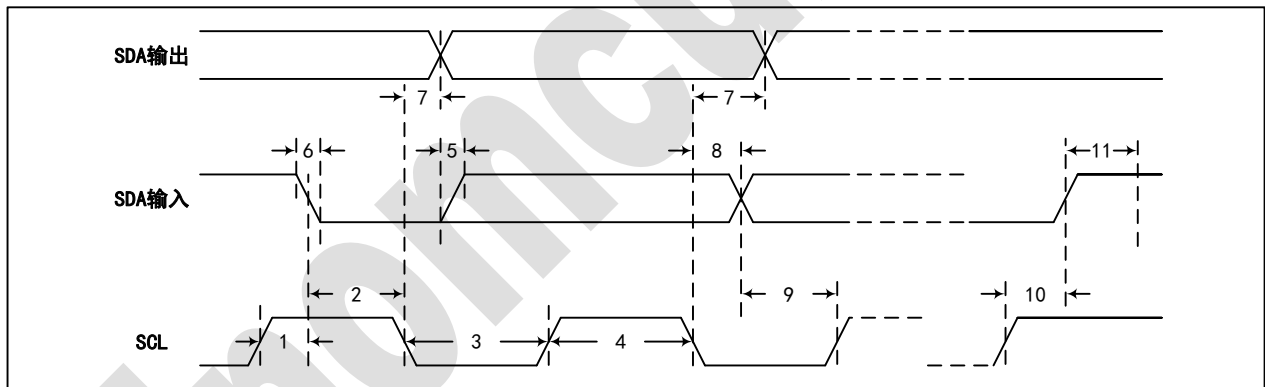


图 10-3 IIC 时序说明

表 10-1 主机模式信号输出时间定义

标号	参数	符号	典型值		单位
			100Kbps	400Kbps	
1	重复起始条件的建立时间	$T_{SU:STR}$	6	1.5	μs
2	起始条件的保持时间	$T_{HD:STR}$	4	1	μs
3	SCL 时钟的低电平周期	T_{LOW}	6	1.5	μs
4	SCL 时钟的高电平周期	T_{HIGH}	4	1	μs



5	SDA 和 SCL 上升沿时间	T_R	1000	300	ns
6	SDA 和 SCL 下降沿时间	T_F	300	300	ns
7	时钟输出有效时间	T_{AA}	5	1	μs
10	停止条件的建立时间	$T_{SU:STP}$	4	1	μs
11	停止条件的保持时间	$T_{HD:STP}$	6	1.5	μs

表 10-2 主机模式信号输入时间要求

标号	参数	符号	最小值		单位
			100Kbps	400Kbps	
8	数据输入的保持时间	$T_{HD:DAT}$	1	0.5	μs
9	数据输入的建立时间	$T_{SU:DAT}$	300	150	ns

表 10-3 从机模式信号输入时间要求

标号	参数	符号	最小值		单位
			100Kbps	400Kbps	
1	重复起始条件的建立时间	$T_{SU:STR}$	4.7	0.6	μs
2	起始条件的保持时间	$T_{HD:STR}$	4	0.6	μs
3	SCL 时钟的低电平周期	T_{LOW}	4.7	1.3	μs
4	SCL 时钟的高电平周期	T_{HIGH}	4	0.6	μs
9	数据输入的建立时间	$T_{SU:DAT}$	300	150	ns
10	停止条件的建立时间	$T_{SU:STP}$	4	0.6	μs
11	停止条件的保持时间	$T_{HD:STP}$	4	0.6	μs

10.5 IIC 多机通讯

IIC 模块支持多主机通讯，可实现时钟同步和总线仲裁功能。

主机模式下，IIC 模块发送时钟的高电平后会检测 SCL 线，直至 SCL 为高后开始高电平宽度计时，在高电平保持时间达到后才发送时钟的低电平，若在高电平计时期间 SCL 线被其他设备拉低，IIC 模块则将直接进入低电平宽度计时，如此 SCL 线上时钟的低电平时间由所有主机中最长的低电平时长决定，而高电平时间由最短的高电平时长决定，从而实现多主机在 SCL 线上的时钟同步。

主机或从机模式下，IIC 模块发送数据的高电平后会进行总线仲裁，即检测 SDA 线上的电平状态，若仍为低电平，则



表示高电平未发送成功，已有其他设备占用总线，此时 IIC 模块将释放总线，总线仲裁状态标志 IICARLO 将被置 1。

主机模式下发送起始信号或重复起始信号时，IIC 模块释放 SCL 总线时检测到 SDA 为低电平和 IIC 模块拉低 SDA 时检测到 SCL 为低电平，总线仲裁状态标志 IICARLO 将被置 1。

主机模式下发送停止信号时，IIC 模块释放 SDA 时检测到 SCL 为低电平和停止信号发送完毕后检测到 SDA 为低电平，总线仲裁状态标志 IICARLO 将被置 1。

主机模式下在发送地址帧/数据帧、发送应答时，从机模式下在发送数据帧或发送应答时，在向 SDA 发送高电平，但在 SCL 的上升沿却从 SDA 上检测到低电平，总线仲裁状态标志 IICARLO 将被置 1。

只有使能位 IICEN 清 0 才能将状态标志 IICARLO 清 0。

10.6 IIC 相关寄存器

10.6.1 IIC 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IICCR0	IICEN	IICMOD	IICSPD	IICSYN	IICGCE	MNASTPE	IICCHS	IICRUS
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] IICEN – IIC 使能位

- 0: 关闭 IIC，端口用于其他功能；
- 1: 使能 IIC，被选端口用作 SCL/SDA；

注：应用 IIC 模块，选定的端口需使其数字 I/O 功能。使能 IIC 后，端口将自动转为输入/开漏输出口，输出时其内部上拉电阻控制位依然有效；

BIT[6] IICMOD – IIC 工作模式选择位

- 0: 主机模式；
- 1: 从机模式；

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

- 0: IIC 通讯速率为 100Kbps；
- 1: IIC 通讯速率为 400Kbps；

BIT[4] IICSYN – IIC 从机模式采样方式选择位

- 0: 同步采样；
- 1: 异步采样；

BIT[3] IICGCE – IIC 从机模式广播呼叫使能位

- 0: 从机模式下不支持广播呼叫功能，即不响应广播地址 00H；



1: 从机模式下支持广播呼叫功能, 即响应广播地址 00H;

BIT[2] MNASTPE – 主机模式发送完 NACK 或接收到 NACK 后自动发送 STOP 信号使能位

0: 发送完 NACK 或接收到 NACK 后, 不自动发送 STOP 信号;

1: 发送完 NACK 或接收到 NACK 后, 自动发送 STOP 信号;

BIT[1] IICCHS – IIC 通讯接口选择位

0: IIC 通讯的时钟/数据接口为 SCL0/SDA0;

1: IIC 通讯的时钟/数据接口为 SCL1/SDA1;

BIT[0] 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: 选择内部通用上拉电阻;

1: 选择内部 IIC 专用上拉电阻;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IICCR1	-	-	-	-	MTSAA	MTSTP	MTDTA	MRDTA
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3] MTSAA – 主机模式发送 START 信号和地址帧再接收应答控制位

0: 操作未开始或已完成, 写 1 开始发送起始信号和地址帧再接收从机应答信号;

1: 发送起始信号和地址帧再接收从机应答信号操作中, 完成后自动清 0;

BIT[2] MTSTP – 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[1] MTDTA – 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[0] MRDTA – 主机模式接收数据帧再发送应答控制位

0: 操作未开始或已完成, 写 1 开始接收从机数据帧再发送应答信号;

1: 接收从机数据帧再发送应答信号操作中, 完成后自动清 0;

10.6.2 IIC 状态寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IICSR	TACKS	IICARLO	IICSTRF	IICSTPF	SADMF	SdTFF	SCMDf	RACKF



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

- BIT[7] 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[6] IICARLO – IIC 总线仲裁状态标志位
 0: IIC 总线上未检测到仲裁失败;
 1: IIC 总线上检测到仲裁失败 (仅在 IIC 关闭后清 0);
- BIT[5] IICSTRF – IIC 总线 START 信号检测状态标志位
 0: 总线上未检测到起始信号;
 1: 总线上检测到起始信号 (检测到停止信号后清 0);
- BIT[4] IICSTPF – IIC 总线 STOP 信号检测状态标志位
 0: 总线上未检测到停止信号;
 1: 总线上检测到停止信号 (检测到起始信号后清 0);
- BIT[3] SADMF – 从机模式地址匹配状态标志位
 0: 从机模式接收到的地址帧中地址不匹配, 或接收到数据帧;
 1: 从机模式接收到地址帧, 且地址与本机预设地址匹配;
- BIT[2] SDTFF – 从机模式数据帧收发完成状态标志位
 0: 从机模式数据帧接收/发送中, 或未接收到数据帧;
 1: 从机模式数据帧接收/发送完成;
- BIT[1] SCMDF – 从机模式读写命令接收内容标志位
 0: 从机模式接收到的主机命令为“主机写”命令 (逻辑“0”);
 1: 从机模式接收到的主机命令为“主机读”命令 (逻辑“1”);
- BIT[0] 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);

10.6.3 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，写操作为写发送缓冲器、读操作为读接收缓冲器

注：对 IICDR 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。

10.6.4 IIC 主地址寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IICAR	IICA6	IICA5	IICA4	IICA3	IICA2	IICA1	IICA0	MCMDS
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[7:1] IICA[6:0] – IIC 主地址寄存器，主机模式下为待发送的地址，从机模式下为预设的从机主地址

BIT[0] MCMDS – 主机模式读写命令发送内容选择位

0：主机模式待发送的命令为“主机写”命令（逻辑“0”）；

1：主机模式待发送的命令为“主机读”命令（逻辑“1”）；

10.6.5 IIC 辅地址寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IICAAAR	IICAA6	IICAA5	IICAA4	IICAA3	IICAA2	IICAA1	IICAA0	-
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[7:1] IICAA[6:0] – IIC 辅地址寄存器，主机模式下无效，从机模式下为预设的从机辅地址

10.6.6 IIC 辅地址掩码寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IICAMR	IICAM6	IICAM5	IICAM4	IICAM3	IICAM2	IICAM1	IICAM0	-
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[7:1] IICAMn – 从机模式 IIC 通讯辅地址掩码控制位（n=6-0）

0：从机模式下，检验预设地址位 IICAA_n 与接收的对应地址位是否匹配；

1：从机模式下，不检验接收的对应地址位；



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，接收时会对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				
UDATF	UPEN	USTPF	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，时钟源为外设高频时钟 FHCLK (8MHz)，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 (Fuart)	URTPRS	预分频	URTBR	波特率计算值 (bps)	波特率目标值 (bps)	偏差 (%)
8MHz	101	32 分频	207	1201.92	1200	0.16



	101	32 分频	103	2403.85	2400	0.16
	100	16 分频	146	3401.36	3400	0.04
	011	8 分频	207	4807.69	4800	0.16
	010	4 分频	207	9615.38	9600	0.16
	010	4 分频	138	14388.49	14400	-0.08
	001	2 分频	207	19230.77	19200	0.16
	000	1 分频	207	38461.54	38400	0.16
	000	1 分频	138	57553.96	57600	-0.08
	000	1 分频	68	115942.03	115200	0.64
	000	1 分频	20	380952.38	384000	-0.79

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	-	-	-	URTCHS1	URTCHS0	URTPTS	URTMOD	URLNV
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	-	-	-	0	0	0	0	0

BIT[4:3] URTCHS[1:0] – UART 通讯接口选择位

URTCHS[1:0]	UART 发送端口	UART 接收端口
00	TX0	RX0
01	TX1	RX1
10	RX0	TX0
11	RX1	TX1

BIT[2] URTPTS – 单线模式 UART 通讯接口选择位 (仅 URTMOD=1 时有效)
 0: UART 双线模式下的接收端口用作 UART 单线模式下的收/发接口;
 1: UART 双线模式下的发送端口用作 UART 单线模式下的收/发接口;

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;

注: 单线/双线模式收发端口如下表所示:

URTMOD	URTCHS[1:0]	URTPTS	UART 发送端口	UART 接收端口
0	00	X	TX0	RX0



(双线模式)	01	X	TX1	RX1
	10	X	RX0	TX0
	11	X	RX1	TX1
1 (单线模式)	00	0	RX0	
		1	TX0	
	01	0	RX1	
		1	TX1	
	10	0	TX0	
		1	RX0	
	11	0	TX1	
		1	RX1	

BIT[0] URTLNv – 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；



12 在应用自编程 IAP

12.1 IAP 概述

芯片包括 3 个掉电非易失的存储器区：8K×16 位的 FLASH 程序代码区（即程序存储器）、256×16 位的 EEPROM 数据区、16×16 位的 INFO 配置区，均支持用户程序在带电运行中实时地读写数据，即支持在应用自编程 IAP 功能（其中 FLASH 区和 INFO 区的 IAP 功能需先通过配置字 FLASHIAPEN 或 INFOIAPEN 使能）。

FLASH 区/EEPROM 区/INFO 区，存储区地址均以 16 个字（WORD）为 1 页（PAGE），FLASH 区还以 64 页为 1 个扇区（SECTOR）。IAP 通过选择位 IAPMS 选择进行 IAP 操作的存储区，通过地址寄存器 IAPSAR（扇区地址）/IAPPAR（扇区内的页地址）/IAPWAR（页内的字地址），和 16 位数据寄存器 IAPDR（IAPDRH/IAPDRL），即可进行 IAP 读/写/擦除等操作。

而 FLASH 区，可通过配置字 FLASHMOD 选择将 FLASH 区划分为保护区和非保护区，保护区的存储空间将不允许 IAP 操作；保护区的地址范围可通过配置字 FLASHPGS 选择从第 0 页至第 N 页（N=0~511）。

表 12-1 IAP 存储区地址范围

存储区	IAPSAR	IAPPAR	IAPWAR
FLASH 区	0-7	0-63	0-15
EEPROM 区	0	0-15	0-15
INFO 区	0	1	0-15

可通过控制位 IAPREAD 置 1 启动 IAP 读操作，读操作完成后，IAPREAD 将自动清 0，IAPMS 所选存储区中地址寄存器 IAPSAR/IAPPAR/IAPWAR 所指地址的 16 位内容将被读出并缓存于 IAPDR 中。

可通过控制位 IAPWRITE 置 1 启动 IAP 写操作，通过控制位 IAPERASE 置 1 启动 IAP 擦除操作。

为防止误触发 IAP 的擦除和写操作，需先对保护寄存器 IAPPR 写 5AH 再立即写 A5H，然后 IAPCR1 中的擦除/写操作启动控制位 IAPERASE/IAPWRITE 才能置 1，中间不能插入其他操作（包括 NOP 操作），否则 IAPERASE/IAPWRITE 将无法置 1。在写 IAPPR 前需先屏蔽中断，否则可能会因系统响应中断而导致 IAPERASE/IAPWRITE 无法置 1。

IAP 的写/擦除操作，都需要通过内置的一个 16×16 位的缓冲区 IAPBUF 间接实现。IAPBUF 中的 16 个地址均具有单独的“空/非空”状态，“空”状态的缓冲区地址在 IAPBUFL 置 1 时加载数据并由“空”状态转换为“非空”状态。在 IAPWRITE/IAPERASE 置 1 启动 IAP 写/擦除操作后，“非空”状态的地址其所对应的 IAP 存储区的页内地址被写入或被擦除，“空”状态的地址其所对应的 IAP 存储区的页内地址保留原值。每次写入或擦除都需要先通过 IAPBUFC 置 1 清空 IAPBUF 使其恢复为全“空”状态，再通过 IAPBUFL 置 1 加载数据到相应 IAPBUF 缓冲区，然后才能通过 IAPWRITE/IAPERASE 置 1 启动 IAP 写/擦除操作。



IAP 可选手动模式或自动模式执行 IAP 的擦除/写操作，每种模式都支持以 PAGE(SECTOR)或 WORD 方式对 IAPBUF 加载数据。WORD 方式下可加载 1 个地址的数据，PAGE 方式下可加载多个地址的数据。

自动模式下缓冲区数据来自 IAPDR 指向的 SRAM, PAGE (SECTOR) 方式以 1 页 16 个地址为单位,对 IAPSAR/IAPPAR 所指向的 1 页中的 16 个地址执行一次性的写入或擦除动作；WORD 方式对 IAPSAR/IAPPAR/IAPWAR 所指向的 1 个地址执行一次性的写入或擦除动作。

自动模式下 IAPBUFC 和 IAPBUFL 操作由硬件自动执行，用户只需提前设置 IAPDR 和相应 SRAM 即可。

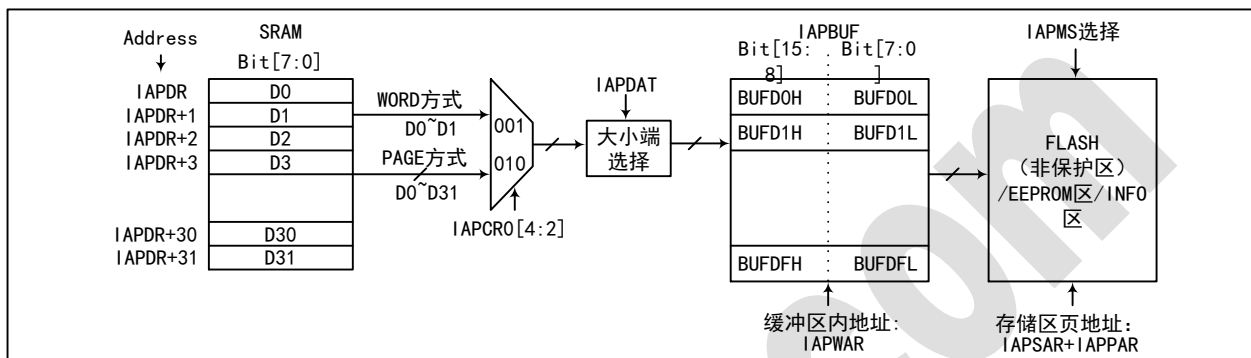


图 12-1 IAP 自动模式写/擦除操作示意图

手动模式下缓冲区数据来自 IAPDR, PAGE (SECTOR) 方式对 IAPSAR/IAPPAR 所指向的 1 页中“非空”状态对应的多个地址执行一次性的写入或擦除动作；WORD 方式对 IAPSAR/IAPPAR/IAPWAR 所指向的 1 个地址执行一次性的写入或擦除动作。

手动模式下 IAPBUFC 和 IAPBUFL 操作需用户程序支持，先 IAPBUFC 置 1，重复修改 IAPWAR 和 IAPDR 并执行 IAPBUFL 置 1，将需操作的缓冲区状态由“空”状态转换为“非空”状态。

注：

IAP 存储区中的内容必须先擦除后才能写入新的数据，不允许多次重复写入；

IAP 操作时，CPU 将暂停工作；

SECTOR 方式只支持擦除存储区操作，擦除操作将 1 个扇区中的 64 页内容全部擦除(手动模式下需 IAPBUF 为全“非空”状态)；

12.2 IAP 操作步骤

手动模式，改写存储区某一个地址（或一页的多个地址）中的内容，步骤如下：

- (1) IAPMOD 清 0，选择手动模式；
- (2) 设置[SECTOR:PAGE:WORD]为 001，选择 WORD 方式；
(若改写页内多个地址中的内容，则设置[SECTOR:PAGE:WORD]为 010，选择 PAGE 方式)；
- (3) 通过 IAPMS 选择存储区，存储区目标的扇区/页地址写入 IAPSAR/IAPPAR；



- (4) IAPBUFC 置 1，清空 IAPBUF；
- (5) 将新数据写入 IAPDR，存储区目标字地址写入 IAPWAR；
- (6) IAPBUFL 置 1，执行 IAPBUF 加载操作；
- (7) 若改写页内多个地址中的内容，则循环执行 (5)、(6) 多次，将多个数据载入 IAPBUF 的多个地址中；
- (8) IAPERASE 置 1，执行 IAP 擦除操作；
- (9) 重复执行 (4) ~ (7) 1 次；
- (10) IAPWRITE 置 1，执行 IAP 写操作；

自动模式，改写存储区中某一个地址（或一页全部 16 个地址）中的内容，步骤如下：

- (1) IAPMOD 置 1，选择自动模式；
- (2) 设置[SECTOR:PAGE:WORD]为 001，选择 WORD 方式；
(若改写一页 16 个地址的内容，则设置[SECTOR:PAGE:WORD]为 010，选择 PAGE 方式)；
- (3) 通过 IAPMS 选择存储区，存储区目标的扇区/页地址写入 IAPSAR/IAPPAR；
- (4) 设置大/小端数据格式；
- (5) 将新数据按大/小端格式写入 SRAM 连续的 2 个暂存字节中，暂存字节起始地址写入 IAPDR，存储区目标字地址写入 IAPWAR；
(若改写一页 16 个地址，则将 16 个新数据写入 SRAM 连续的 32 个暂存字节中，暂存字节起始地址写入 IAPDR)
- (6) 设置[IAPERASE:IAPWRITE]为 11，执行 IAP 先擦除再写入的自动操作；

注：手动模式字操作，自动模式字操作，自动模式页操作支持改写方式（[IAPERASE:IAPWRITE]为 11），手动模式的页操作必须分别执行先擦除后写入的操作。

12.3 IAP 相关寄存器

12.3.1 IAP 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IAPCR0	IAPEN	IAPMOD	IAPDAT	SECTOR	PAGE	WORD	IAPMS1	IAPMS0
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] IAPEN – IAP 使能位

0：禁止 IAP 功能；

1：使能 IAP 功能；

BIT[6] IAPMOD – IAP 工作模式选择位

0：IAP 为手动模式；

1：IAP 为自动模式；



BIT[5] IAPDAT – IAP 自动模式下数据格式选择位
 0: 小端格式, 自动模式下 IAP 先读取的 RAM 内容写入缓冲区 16 位字的低字节;
 1: 大端格式, 自动模式下 IAP 先读取的 RAM 内容写入缓冲区 16 位字的高字节;

BIT[4] SECTOR – IAP 扇区 (SECTOR) 操作方式选择位
 0: 未选择 SECTOR 操作方式;
 1: 选择 SECTOR 操作方式;

注: 扇区操作方式仅支持扇区擦除操作, 而不支持扇区写操作。

BIT[3] PAGE – IAP 页 (PAGE) 操作方式选择位
 0: 未选择 PAGE 操作方式;
 1: 选择 PAGE 操作方式;

BIT[2] WORD – IAP 字 (WORD) 操作方式选择位
 0: 未选择 WORD 操作方式;
 1: 选择 WORD 操作方式;

BIT[1:0] IAPMS[1:0] – IAP 存储区操作选择位

IAPMS[1:0]	IAP 操作存储区
00	FLASH 区 (保护区除外)
01	INFO 区
10	EEPROM 区
11	禁用

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IAPCR1	IAPSTAT	IAPREAD	IAPBUFL	IAPBUFC	-	-	IAPERASE	IAPWRITE
R/W	R	R/W	R/W	R/W	R	R	R/W	R/W
初始值	0	0	0	0	1	1	0	0

BIT[7] IAPSTAT – IAP 擦除/写状态标志位
 0: IAP 擦除/写操作未进行或已完成;
 1: IAP 擦除/写操作进行中;

BIT[6] IAPREAD – IAP 读操作启动控制位
 0: IAP 读操作未进行或已完成;
 1: 启动 IAP 读操作, 完成后硬件自动清 0;



BIT[4] IAPBUFL – IAPBUF 加载操作启动控制位
 0: IAPBUF 加载操作未进行或已完成;
 1: 启动 IAPBUF 加载操作, 完成后硬件自动清 0;

BIT[4] IAPBUFC – IAPBUF 清空操作启动控制位
 0: IAPBUF 清空操作未进行或已完成;
 1: 启动 IAPBUF 清空操作, 完成后硬件自动清 0;

BIT[1] IAPERASE – IAP 擦除操作启动控制位
 0: IAP 擦除操作未进行或已完成;
 1: 启动 IAP 擦除操作, 完成后硬件自动清 0;

BIT[0] IAPWRITE – IAP 写操作启动控制位
 0: IAP 写操作未进行或已完成;
 1: 启动 IAP 写操作, 完成后硬件自动清 0;

注: 对 IAPCR1 中 bit[1:0]的写操作, 仅能通过 MOVRA 指令进行, 且必须紧接 2 个 NOP 指令, 以防止时序错误。

12.3.2 IAP 保护寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IAPPR	IAPP7	IAPP6	IAPP5	IAPP4	IAPP3	IAPP2	IAPP1	IAPP0
R/W	W	W	W	W	W	W	W	W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] IAPP[7:0] – IAP 擦除/写操作保护控制位, 需先写 5AH 再立即写 A5H, 寄存器 IAPCR1 中的控制位 IAPERASE/IAPWRITE 才可改写

12.3.3 IAP 地址寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IAPSAR	-	-	-	-	-	IAPSA2	IAPSA1	IAPSA0
R/W	-	-	-	-	-	R/W	R/W	R/W
初始值	-	-	-	-	-	0	0	0

BIT[2:0] IAPSA[2:0] – IAP 操作的扇区 (SECTOR) 地址

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IAPPAR	-	-	IAPPA5	IAPPA4	IAPPA3	IAPPA2	IAPPA1	IAPPA0
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] IAPPA[5:0] – IAP 操作的扇区内页 (PAGE) 地址

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IAPWAR	-	-	-	-	IAPWA3	IAPWA2	IAPWA1	IAPWA0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3:0] IAPWA[3:0] – IAP 操作的页内字 (WORD) 地址

12.3.4 IAP 数据寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IAPDRH	IAPD15	IAPD14	IAPD13	IAPD12	IAPD11	IAPD10	IAPD9	IAPD8
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] IAPD[15:8] – IAP 手动模式操作的 16 位数据高 8 位，或自动模式下 RAM 暂存区起始字节的 16 位地址高 8 位

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IAPDRL	IAPD7	IAPD6	IAPD5	IAPD4	IAPD3	IAPD2	IAPD1	IAPD0
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] IAPD[7:0] – IAP 手动模式操作的 16 位数据低 8 位，或自动模式下 RAM 暂存区起始字节的 16 位地址低 8 位

12.4 IAP 操作示例

手动模式字操作示例

例如，先将数据 55AAH 写入 EEPROM 存储器 18H 中，再读取 EEPROM 存储器 18H 地址中内容：

```

MOVAI    86H
MOVRA    IAPCR0           ; 使能 IAP，选择 WORD 操作，选择 EE 区

MOVAI    00H
MOVRA    IAPSAR           ; 将 00H 写入 IAPSAR
MOVAI    01H
MOVRA    IAPPAR           ; 将 01H 写入 IAPPAR
    
```



```
BSET      IAPCR1,4      ; 清空缓冲区

MOVAI     08H
MOVRA     IAPWAR        ; 将 08H 写入 IAPWAR

MOVAI     55H
MOVRA     IAPDRH        ; 将 55H 写入 IAPDRH
MOVAI     0AAH
MOVRA     IAPDRL        ; 将 AAH 写入 IAPDRL

BSET      IAPCR1,5      ; 加载数据到缓冲区

BCLR      GIE           ; 屏蔽中断
MOVAI     5AH
MOVRA     IAPPR         ; 使能 IAP 操作, 第 1 步: IAPPR 写 5AH
MOVAI     0A5H
MOVRA     IAPPR         ; 使能 IAP 操作, 第 2 步: IAPPR 写 A5H
MOVAI     03H
MOVRA     IAPCR1        ; 启动擦写操作
NOP       ; 为防止时序错误, CPU 必须先执行 2 个 NOP 指令
NOP
BSET      GIE           ; 允许中断
JBCLR     IAPCR1,7      ; 检查 IAP 操作是否完成
GOTO     $-1

MOVAI     00H
MOVRA     IAPSAR        ; 将 00H 写入 IAPSAR
MOVAI     01H
MOVRA     IAPPAR        ; 将 01H 写入 IAPPAR
MOVAI     08H
MOVRA     IAPWAR        ; 将 08H 写入 IAPWAR

MOVAI     40H
MOVRA     IAPCR1        ; 启动读操作
MOVAR     IAPDRH        ; 从 IAPDRH 中读取数据 55H
MOVAR     IAPDRL        ; 从 IAPDRL 中读取数据 AAH
```

手动模式多个字操作示例

例如, 将数据 55AAH/CC33H 写入 EEPROM 存储器 18H/1AH 中:

```
MOVAI     8AH
MOVRA     IAPCR0        ; 使能 IAP, 选择 PAGE 操作, 选择 EE 区
```



MOVAI	00H	
MOVRA	IAPSAR	; 将 00H 写入 IAPSAR
MOVAI	01H	
MOVRA	IAPPAR	; 将 01H 写入 IAPPAR
BSET	IAPCR1,4	; 清空缓冲区
MOVAI	08H	
MOVRA	IAPWAR	; 将 08H 写入 IAPWAR
MOVAI	55H	
MOVRA	IAPDRH	; 将 55H 写入 IAPDRH
MOVAI	0AAH	
MOVRA	IAPDRL	; 将 AAH 写入 IAPDRL
BSET	IAPCR1,5	; 加载数据到缓冲区
MOVAI	0AH	
MOVRA	IAPWAR	; 将 0AH 写入 IAPWAR
MOVAI	0CCH	
MOVRA	IAPDRH	; 将 CCH 写入 IAPDRH
MOVAI	33H	
MOVRA	IAPDRL	; 将 33H 写入 IAPDRL
BSET	IAPCR1,5	; 加载数据到缓冲区
BCLR	GIE	; 屏蔽中断
MOVAI	5AH	
MOVRA	IAPPR	; 使能 IAP 操作, 第 1 步: IAPPR 写 5AH
MOVAI	0A5H	
MOVRA	IAPPR	; 使能 IAP 操作, 第 2 步: IAPPR 写 A5H
MOVAI	02H	
MOVRA	IAPCR1	; 启动擦除操作
NOP		; 为防止时序错误, CPU 必须先执行 2 个 NOP 指令
NOP		
BSET	GIE	; 允许中断
JBCLR	IAPCR1,7	; 检查 IAP 操作是否完成
GOTO	\$-1	
; 擦除后需重新填充缓冲区再进行写入操作		
BSET	IAPCR1,4	; 清空缓冲区
MOVAI	08H	



```
MOVRA    IAPWAR    ; 将 08H 写入 IAPWAR

MOVAI    55H
MOVRA    IAPDRH    ; 将 55H 写入 IAPDRH
MOVAI    0AAH
MOVRA    IAPDRL    ; 将 AAH 写入 IAPDRL

BSET     IAPCR1,5    ; 加载数据到缓冲区

MOVAI    0AH
MOVRA    IAPWAR    ; 将 0AH 写入 IAPWAR

MOVAI    0CCH
MOVRA    IAPDRH    ; 将 CCH 写入 IAPDRH
MOVAI    33H
MOVRA    IAPDRL    ; 将 33H 写入 IAPDRL

BSET     IAPCR1,5    ; 加载数据到缓冲区

BCLR     GIE        ; 屏蔽中断
MOVAI    5AH
MOVRA    IAPPR      ; 使能 IAP 操作，第 1 步：IAPPR 写 5AH
MOVAI    0A5H
MOVRA    IAPPR      ; 使能 IAP 操作，第 2 步：IAPPR 写 A5H
MOVAI    01H
MOVRA    IAPCR1     ; 启动写入操作
NOP      ; 为防止时序错误，CPU 必须先执行 2 个 NOP 指令
NOP
BSET     GIE        ; 允许中断
JBCLR    IAPCR1,7    ; 检查 IAP 操作是否完成
GOTO     $-1
```

自动模式字操作示例

例如，将暂存区从 123H 开始 2 个字节的数据(AA55H)写入 EEPROM 存储器 20H 中，再读取 EEPROM 存储器 20H 地址中内容：

```
MOVAI    0AAH
MOVRA    123H        ; 将 AAH 写入 123H
MOVAI    55H
MOVRA    124H        ; 将 55H 写入 124H

MOVAI    0C6H
MOVRA    IAPCR0     ; 使能 IAP，选择自动模式，选择 WORD 操作，选择 EE 区
```



```

BSET      IAPCR0,5      ; 选择大端模式

MOVAI     00H
MOVRA     IAPSAR        ; 将 00H 写入 IAPSAR
MOVAI     02H
MOVRA     IAPPAR        ; 将 20H 写入 IAPPAR
MOVAI     00H
MOVRA     IAPWAR        ; 将 00H 写入 IAPWAR

MOVAI     01H
MOVRA     IAPDRH        ; 将 01H 写入 IAPDRH
MOVAI     23H
MOVRA     IAPDRL        ; 将 23H 写入 IAPDRL

BCLR      GIE           ; 屏蔽中断
MOVAI     5AH
MOVRA     IAPPR         ; 使能 IAP 操作，第 1 步：IAPPR 写 5AH
MOVAI     0A5H
MOVRA     IAPPR         ; 使能 IAP 操作，第 2 步：IAPPR 写 A5H
MOVAI     03H
MOVRA     IAPCR1        ; 启动擦写操作
NOP       ; 为防止时序错误，CPU 必须先执行 2 个 NOP 指令
NOP
BSET      GIE           ; 允许中断
JBCLR     IAPCR1,7      ; 检查 IAP 操作是否完成
GOTO     $-1

MOVAI     00H
MOVRA     IAPSAR        ; 将 00H 写入 IAPSAR
MOVAI     02H
MOVRA     IAPPAR        ; 将 01H 写入 IAPPAR
MOVAI     00H
MOVRA     IAPWAR        ; 将 08H 写入 IAPWAR

MOVAI     40H
MOVRA     IAPCR1        ; 启动读操作
MOVAR     IAPDRH        ; 从 IAPDRH 中读取数据 AAH
MOVAR     IAPDRL        ; 从 IAPDRL 中读取数据 55H
    
```

自动模式页操作示例

例如，将暂存区从 123H 开始 32 个字节的数据写入 EEPROM 存储器 20H-2FH 中：

```

MOVAI     0CAH
MOVRA     IAPCR0        ; 使能 IAP，选择自动模式，选择 PAGE 操作，选择 EE 区
    
```



```
MOVAI    00H
MOVRA    IAPSAR        ; 将 00H 写入 IAPSAR
MOVAI    02H
MOVRA    IAPPAR        ; 将 20H 写入 IAPPAR
MOVAI    00H
MOVRA    IAPWAR        ; 将 00H 写入 IAPWAR

MOVAI    01H
MOVRA    IAPDRH        ; 将 01H 写入 IAPDRH
MOVAI    23H
MOVRA    IAPDRL        ; 将 23H 写入 IAPDRL

BCLR     GIE            ; 屏蔽中断
MOVAI    5AH
MOVRA    IAPPR         ; 使能 IAP 操作，第 1 步：IAPPR 写 5AH
MOVAI    0A5H
MOVRA    IAPPR         ; 使能 IAP 操作，第 2 步：IAPPR 写 A5H
MOVAI    03H
MOVRA    IAPCR1        ; 启动擦写操作
NOP      ; 为防止时序错误，CPU 必须先执行 2 个 NOP 指令
NOP
BSET     GIE            ; 允许中断
JBCLR    IAPCR1,7      ; 检查 IAP 操作是否完成
GOTO     $-1
```

注：

- 1、建议通过配置字 FLASHMOD 和 FLASHPGS 选择保护区和非保护区，保护区为 IAP 操作程序，非保护区为用户程序；
- 2、由于擦除后存储器为不定值，在更新用户程序时建议在 EEPROM 区内存储用户程序校验和，以保证程序完整性；
- 3、INFO 区擦写时必须保证不能被打断，否则只能重新上电并用烧录器/仿真器重新下载；
- 4、确认 INFO 区擦写更新成功后，置 DBGCR 中的 SWRST 为 1 完成配置区更新。



13 在板烧录编程 ICP

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，即在上电前的系统电路板上，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：

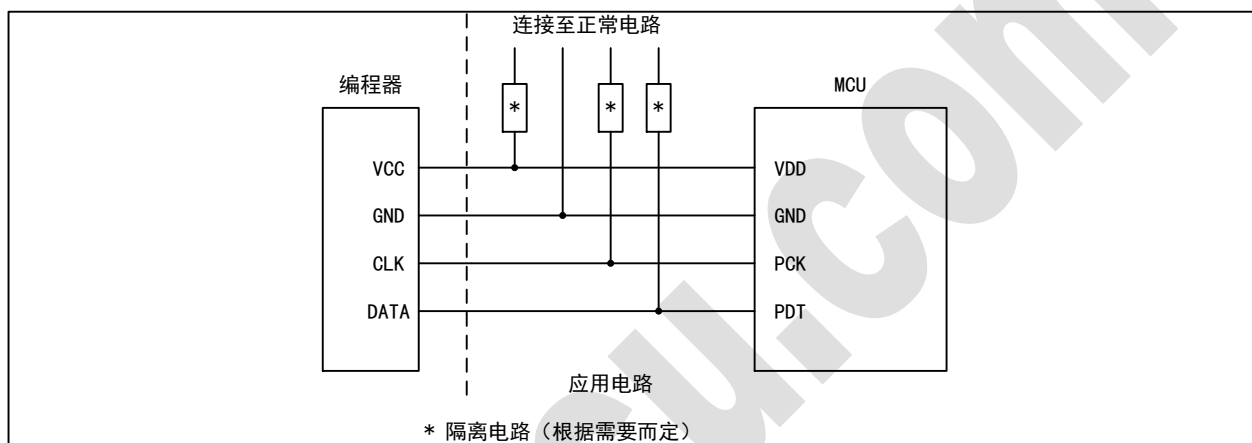


图 13-1 在板编程连接示意图

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	SWRST	-	-	-	-	-	DBG1EN	DBG0EN
R/W	R/W	-	-	-	-	-	R/W	R/W
初始值	0	-	-	-	-	-	0	0



BIT[7] SWRST – 软件复位控制位

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 (P03) /PDT0 (P02) 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 P03/P02 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

```



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、LVD 中断、键盘中断、IIC 中断和 UART 中断 (接收完成、发送完成、发送缓冲器空) 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。

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 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。



14.4 键盘中断

芯片具有 6 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]
P0KBKR	-	-	P05KE	P04KE	P03KE	P02KE	P01KE	P00KE
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] P0nKE – P0n 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 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。

14.6 IIC 中断

IIC 主机模式通讯中，当发送完地址（含命令位）或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 中断使能位为 1，则产生 IIC 中断。

IIC 从机模式通讯中，当接收完地址及命令且匹配预设地址后再发送完应答信号、或接收完主机数据再发送完应答信号、或发送完数据再接收完主机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。

14.7 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。



14.8 中断相关寄存器

14.8.1 中断使能寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTE0	T2IE	LVDIE	ADIE	KBIE	INT1IE	INT0IE	T1IE	T0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] T2IE – 定时器 T2 中断使能位

0: 屏蔽定时器 T2 中断;

1: 使能定时器 T2 中断;

BIT[6] LVDIE – LVD 中断使能位

0: 屏蔽 LVD 中断;

1: 使能 LVD 中断;

BIT[5] ADIE – ADC 中断使能位

0: 屏蔽 ADC 中断;

1: 使能 ADC 中断;

BIT[4] KBIE – 键盘中断使能位

0: 屏蔽键盘中断;

1: 使能键盘中断;

BIT[3] INT1IE – INT1 中断使能位

0: 屏蔽 INT1 中断;

1: 使能 INT1 中断;

BIT[2] INT0IE – INT0 中断使能位

0: 屏蔽 INT0 中断;

1: 使能 INT0 中断;

BIT[1] T1IE – 定时器 T1 中断使能位

0: 屏蔽定时器 T1 中断;

1: 使能定时器 T1 中断;

BIT[0] T0IE – 定时器 T0 中断使能位

0: 屏蔽定时器 T0 中断;

1: 使能定时器 T0 中断;



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

BIT[3] IICIE – IIC 中断使能位

- 0: 屏蔽 IIC 中断;
- 1: 使能 IIC 中断;

BIT[2] 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[1] 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[0] 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;

14.8.2 中断标志寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTF0	T2IF	LVDIF	ADIF	KBIF	INT1IF	INT0IF	T1IF	T0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] T2IF – 定时器 T2 中断标志位

- 0: 未触发定时器 T2 中断;
- 1: 已触发定时器 T2 中断, 需软件清 0;

BIT[6] LVDIF – LVD 中断标志位

- 0: 未触发 LVD 中断;
- 1: 已触发 LVD 中断, 需软件清 0;

BIT[5] ADIF – ADC 中断标志位

- 0: 未触发 ADC 中断;
- 1: 已触发 ADC 中断, 需软件清 0;



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

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

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

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

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

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

BIT[3] 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[2] 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[1] 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[0] 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;

Shomcu.com



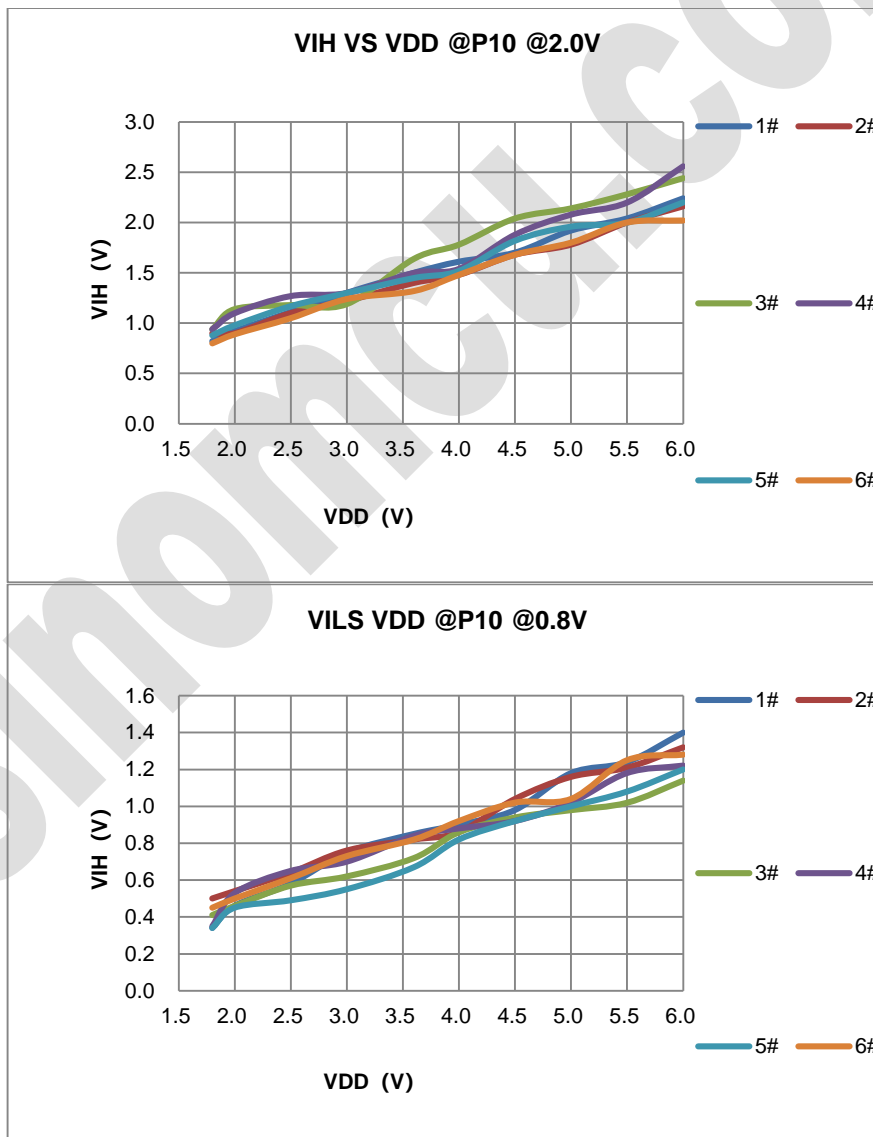
15 特性曲线

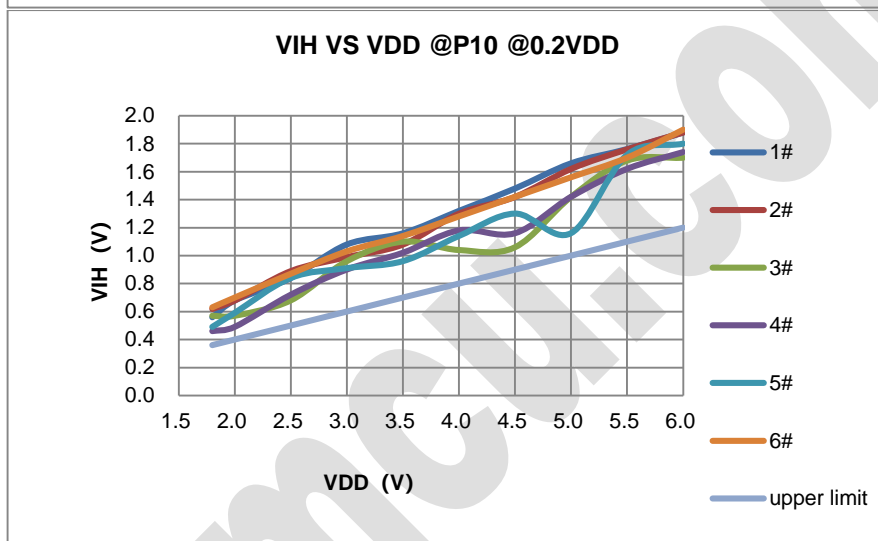
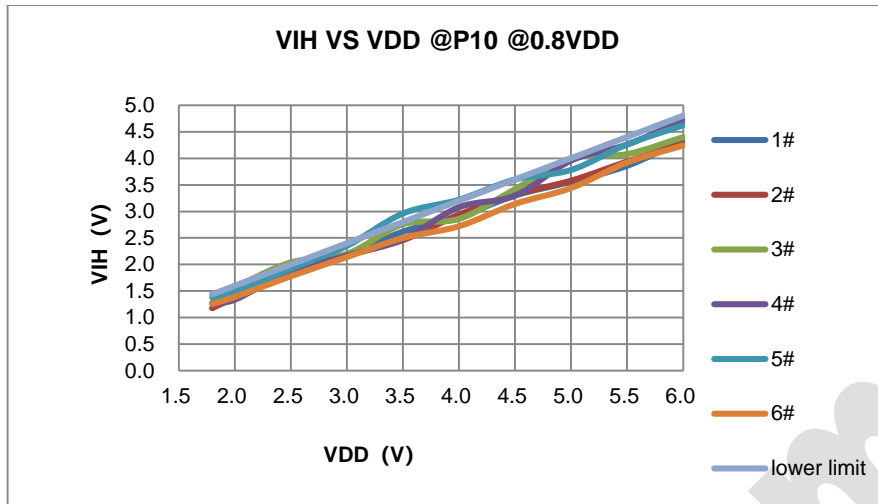
注:

- 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 $V_{DD}=3\text{V}$ ；

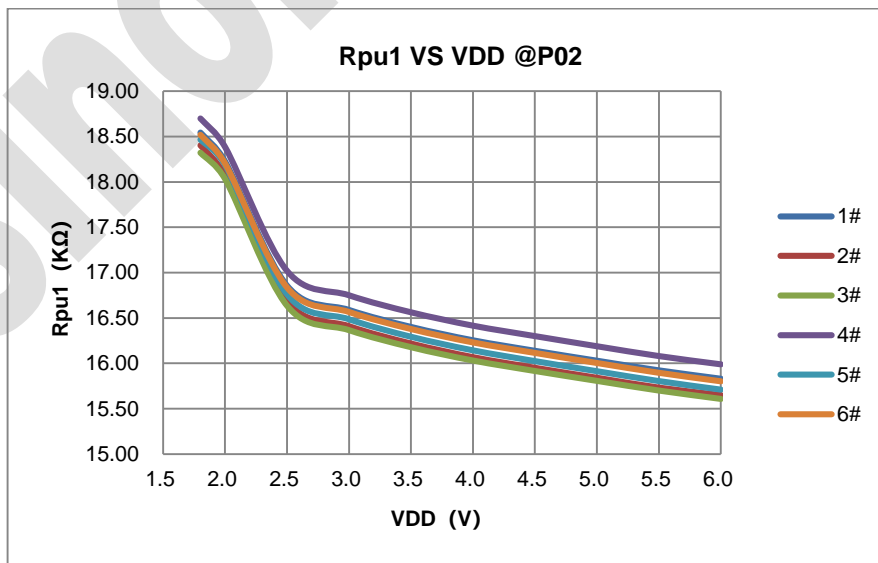
15.1 I/O 特性

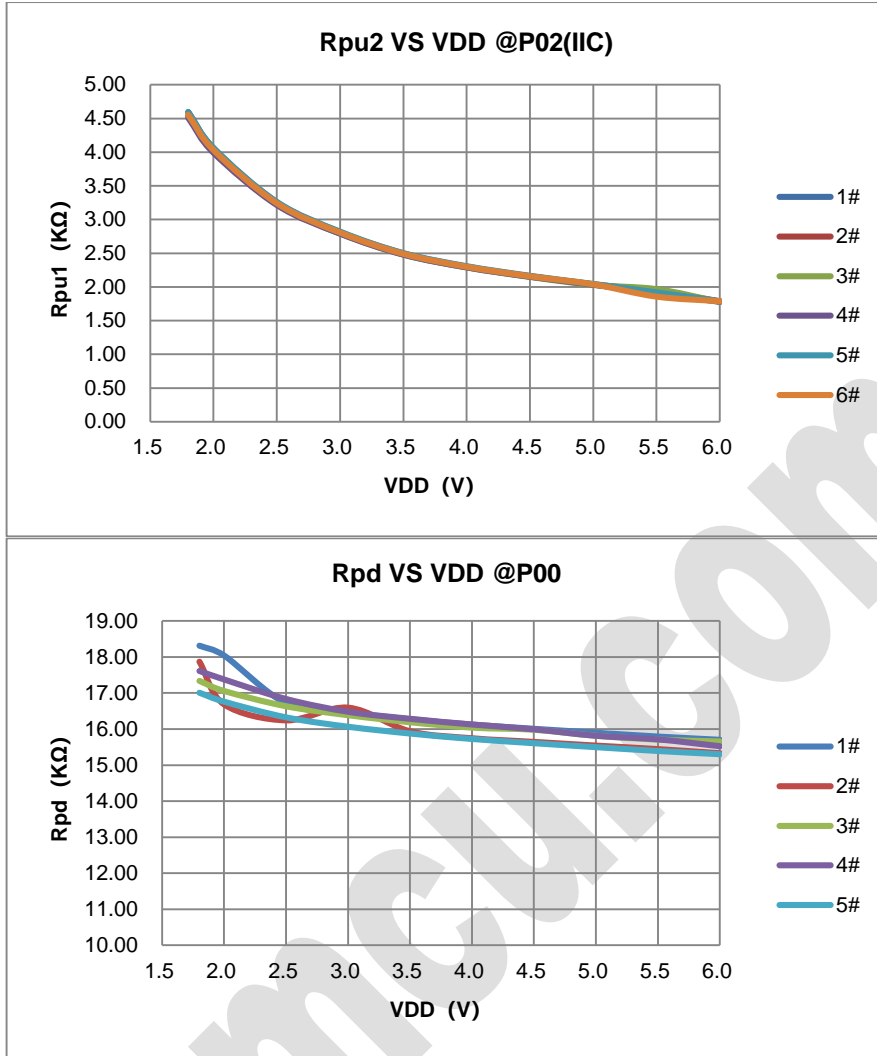
15.1.1 输入 SMT 阈值电压 VS 电源电压



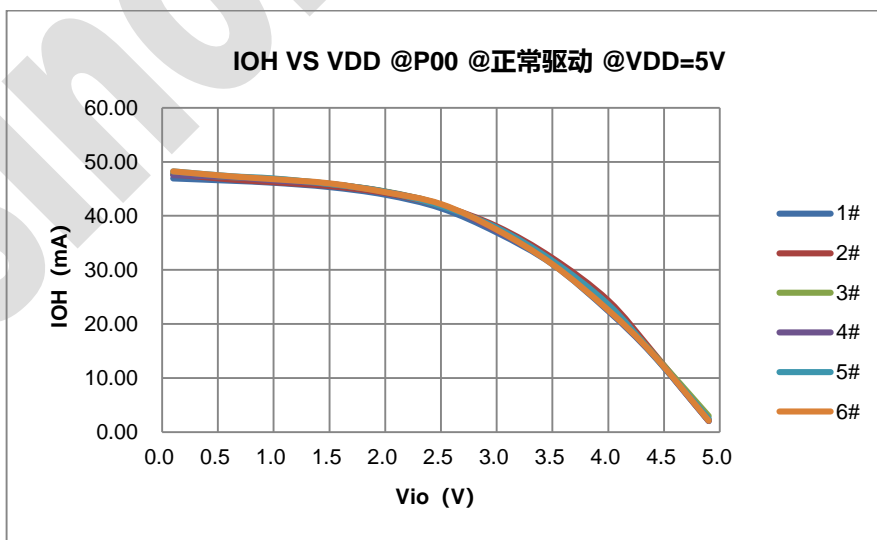


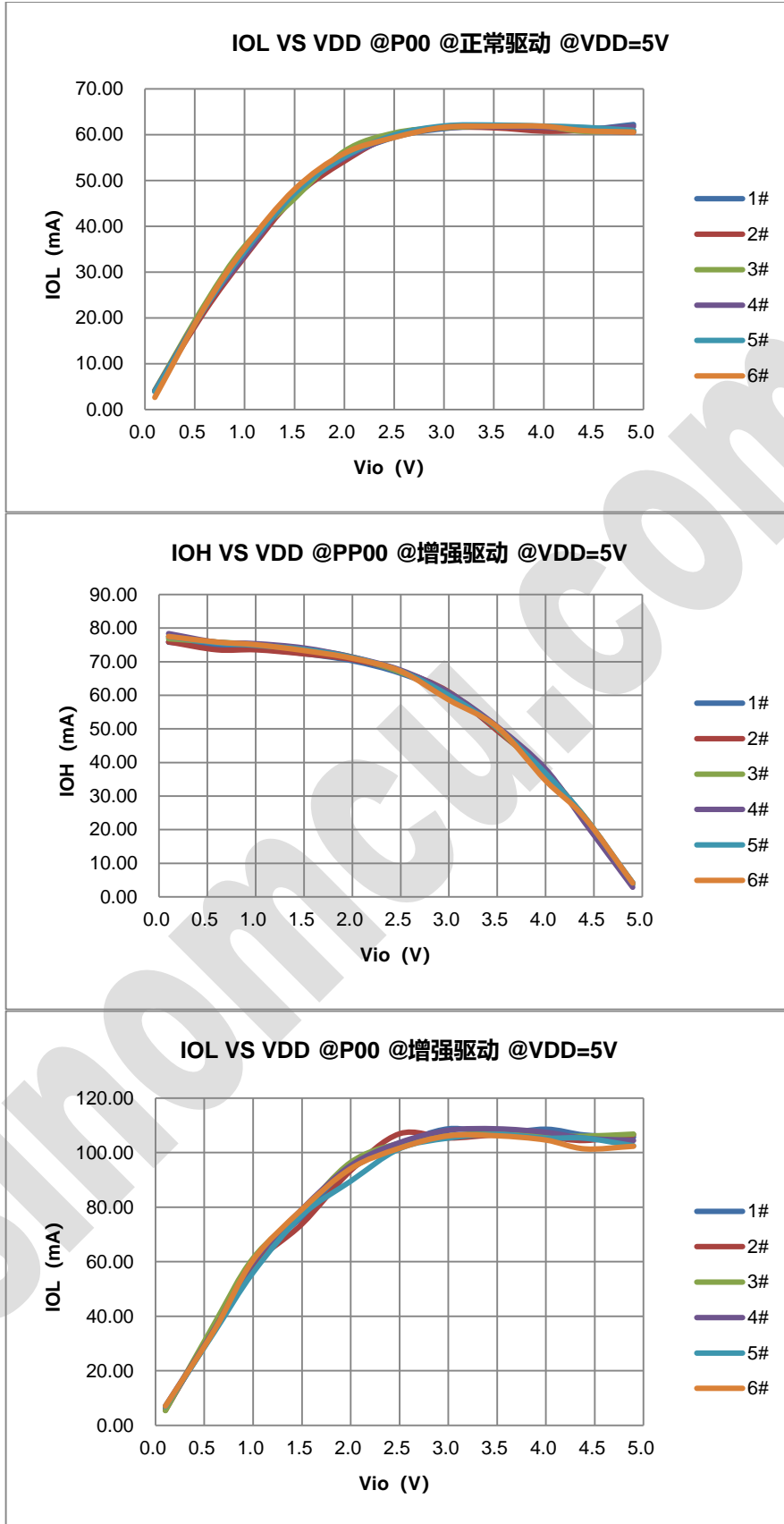
15.1.2 上/下拉电阻值 VS 电源电压

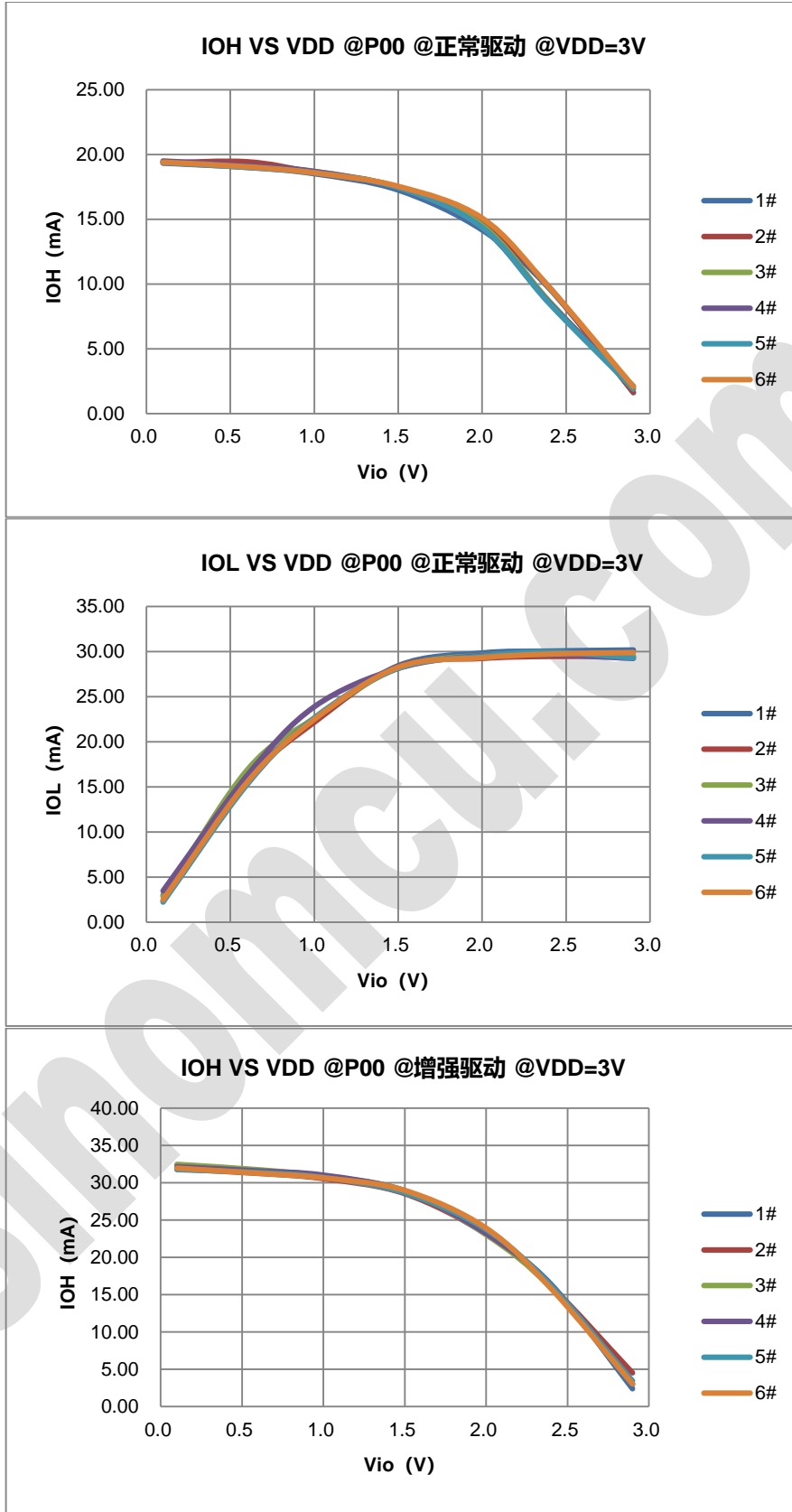


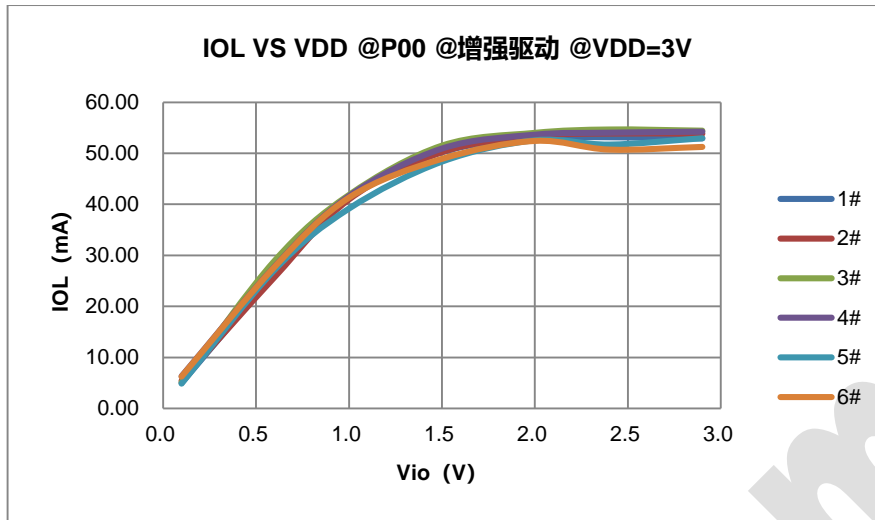


15.1.3 I/O 输出电流 VS 端口电压 (VDD=3V)



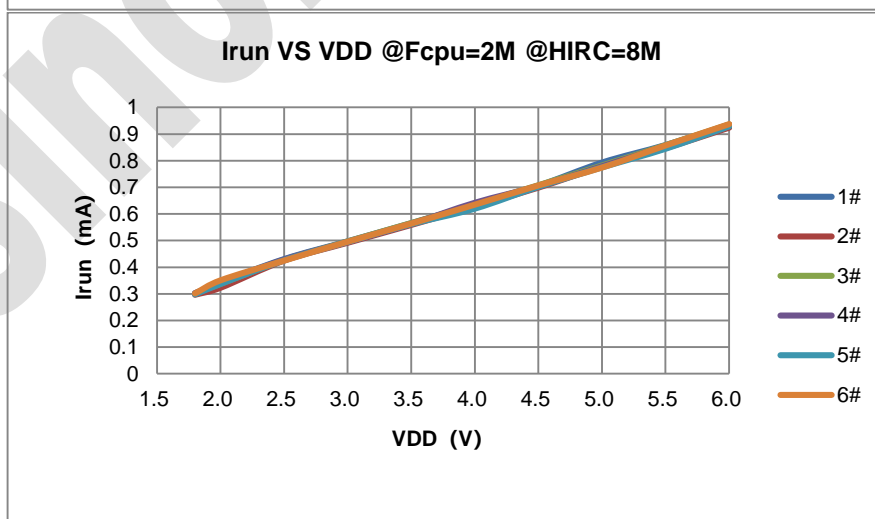
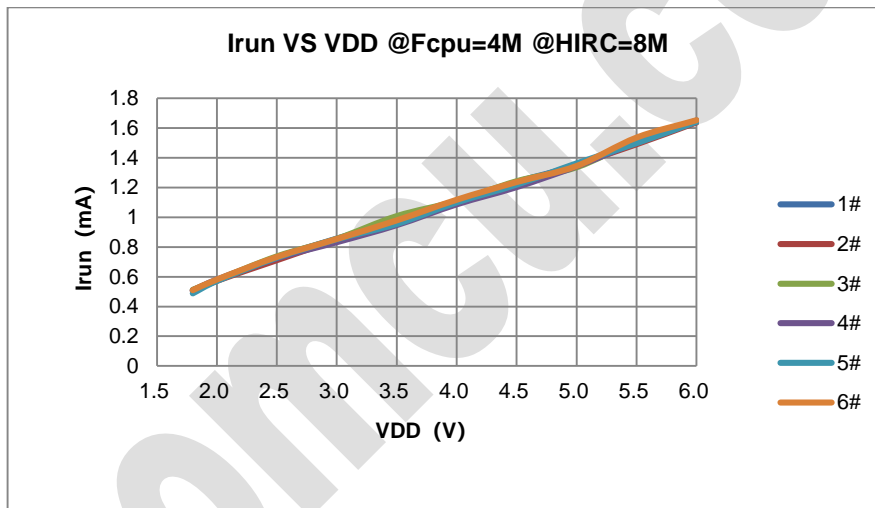


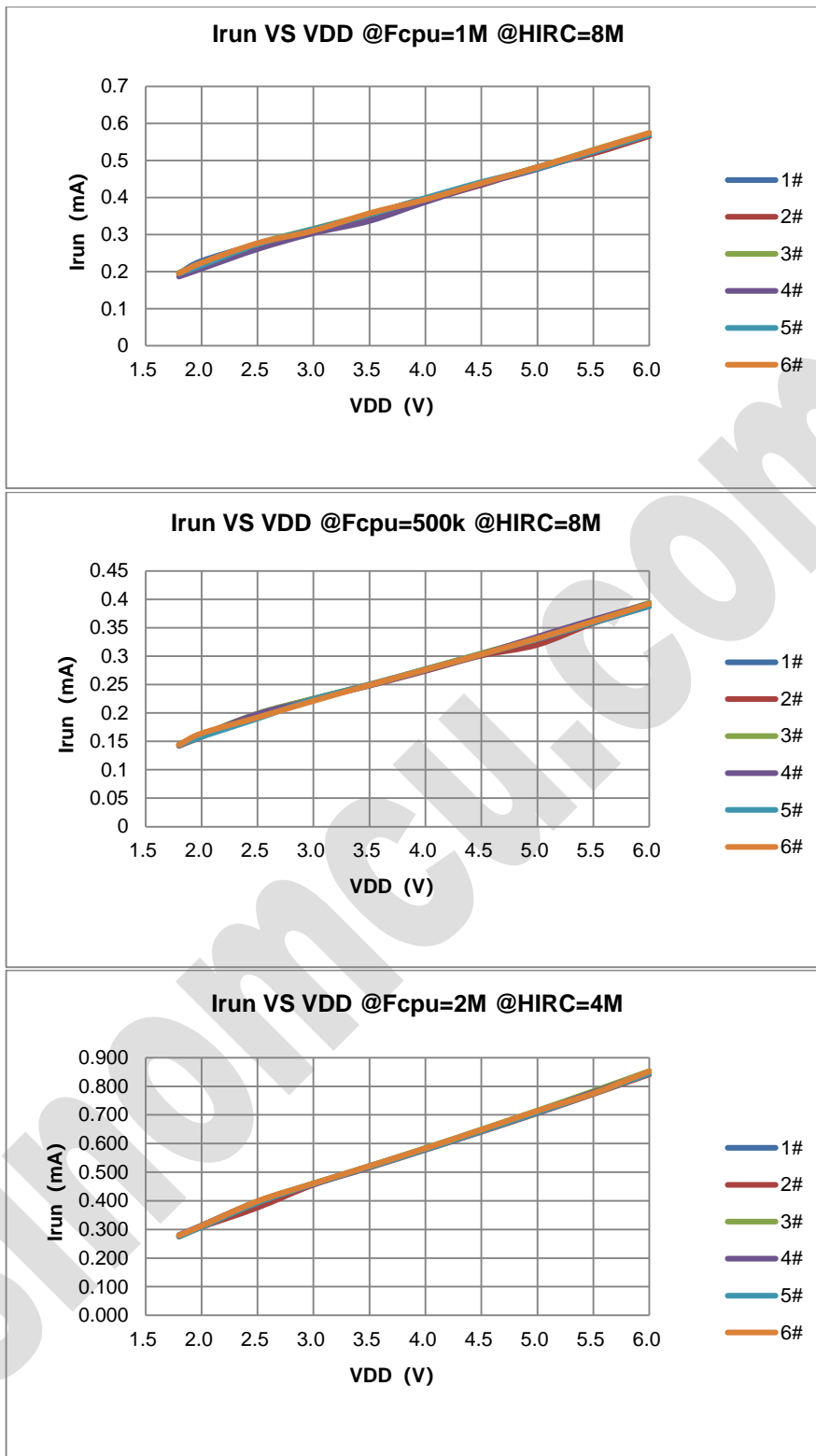


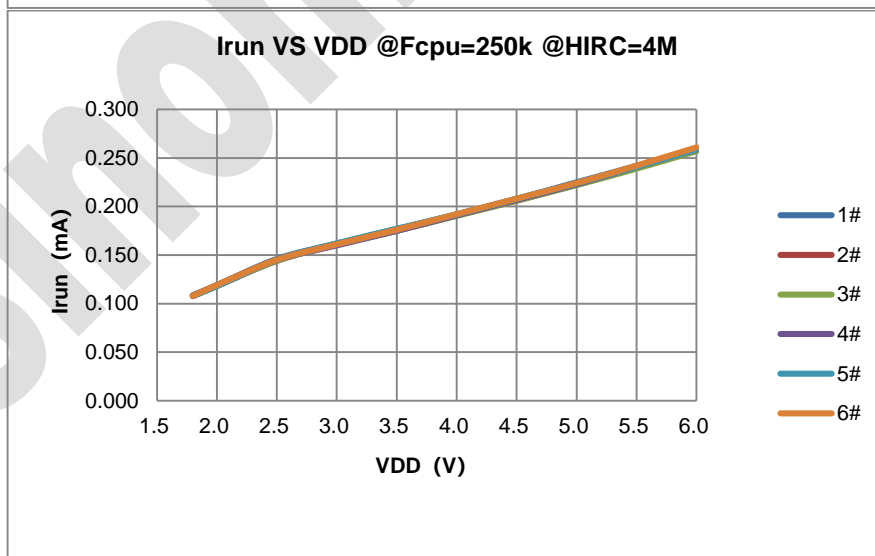
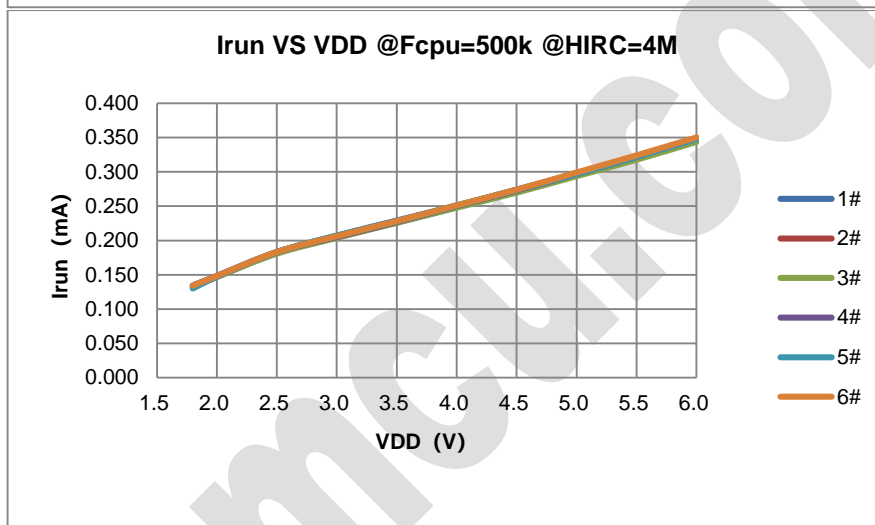
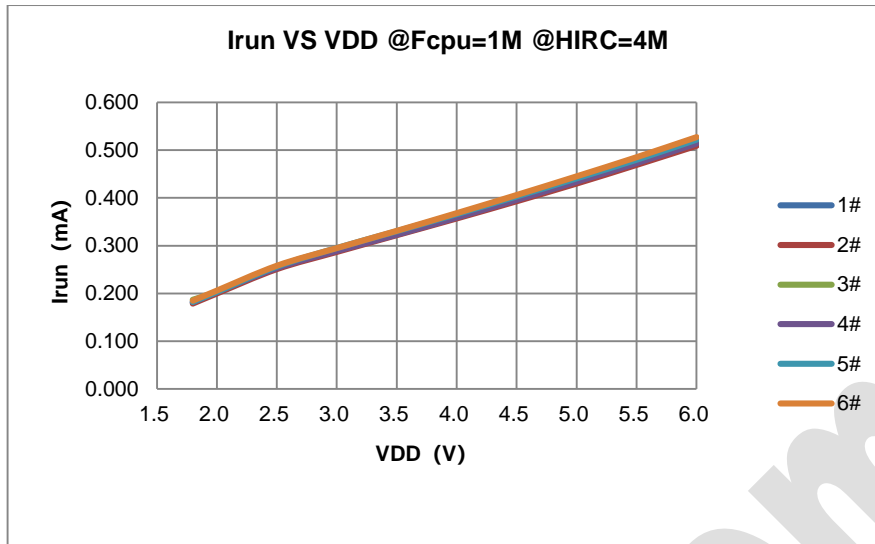


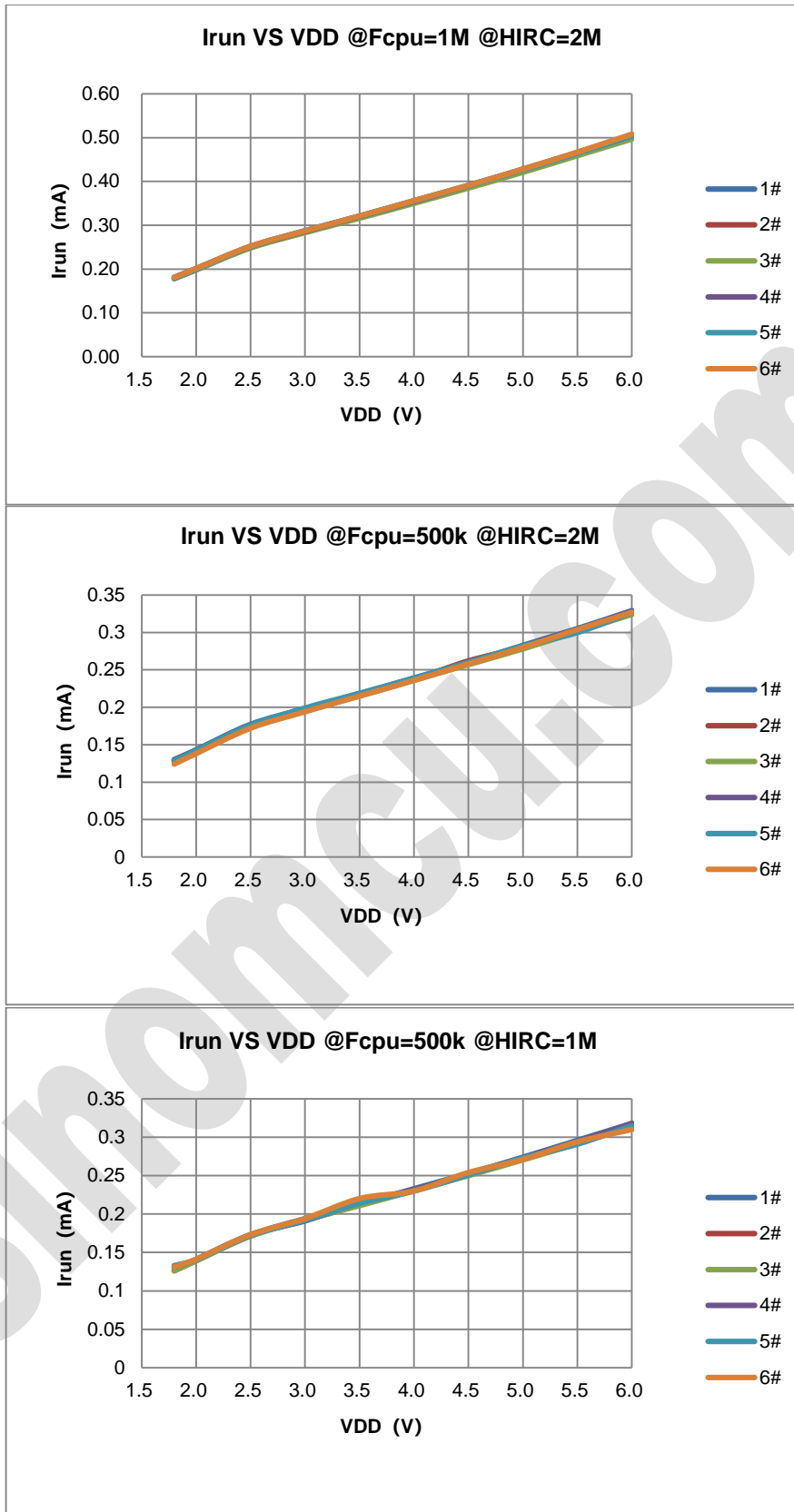
15.2 功耗特性

15.2.1 运行模式 功耗 VS 电源电压

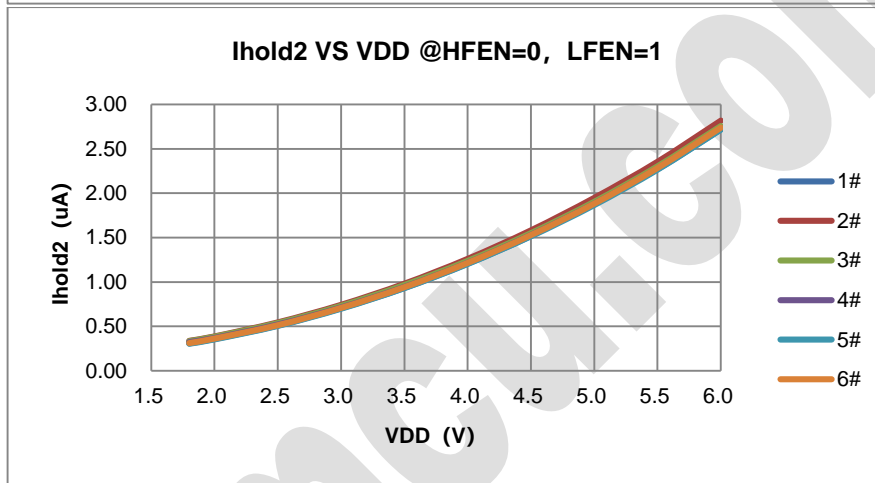
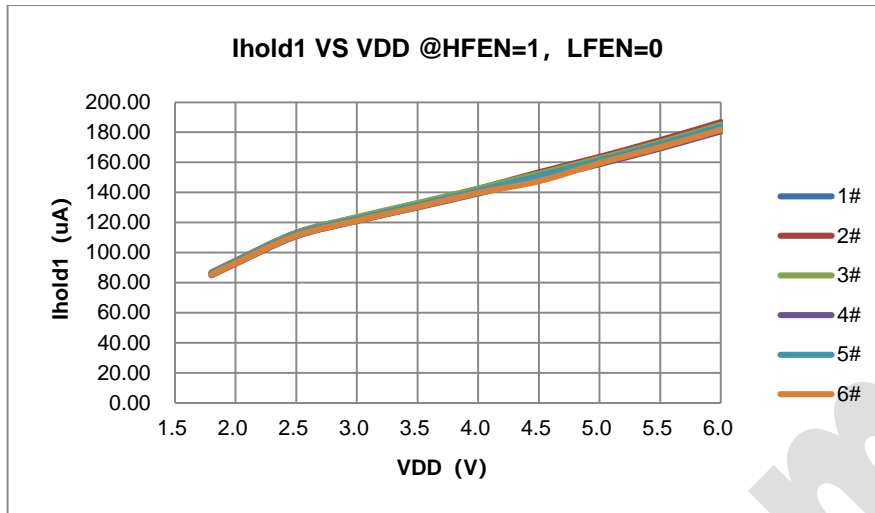




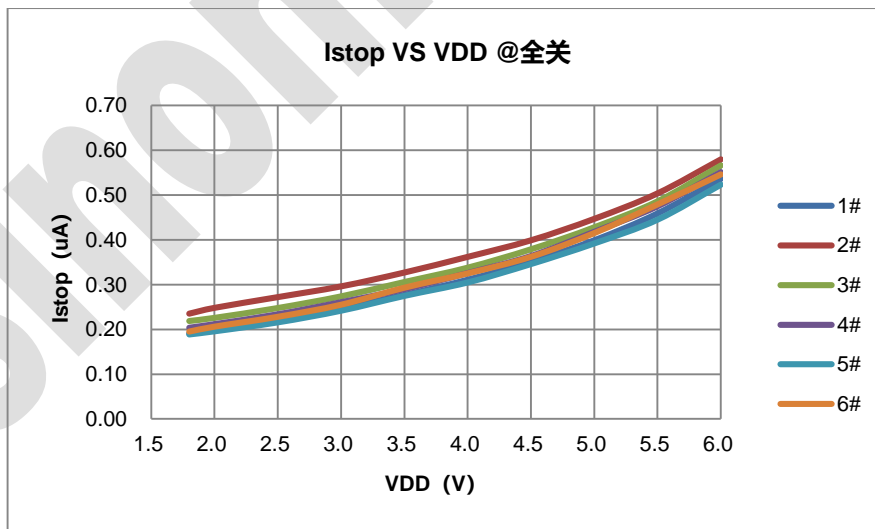


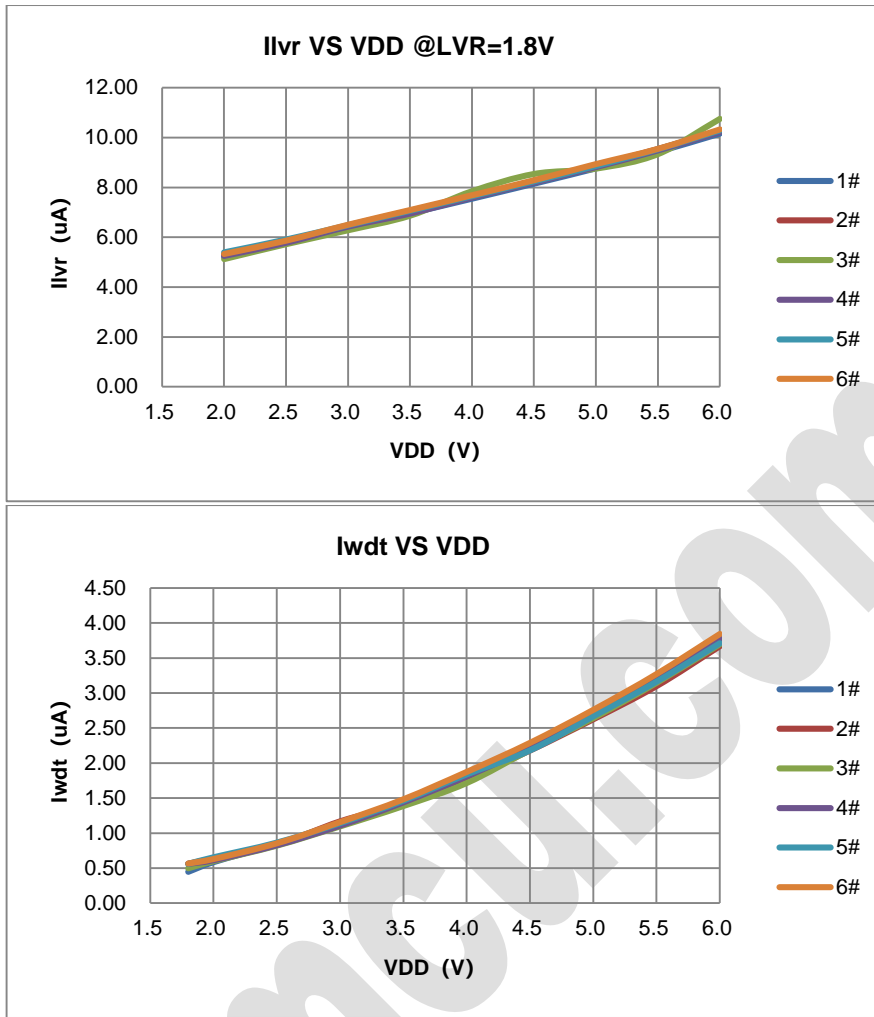


15.2.2 HOLD 模式 功耗 VS 电源电压



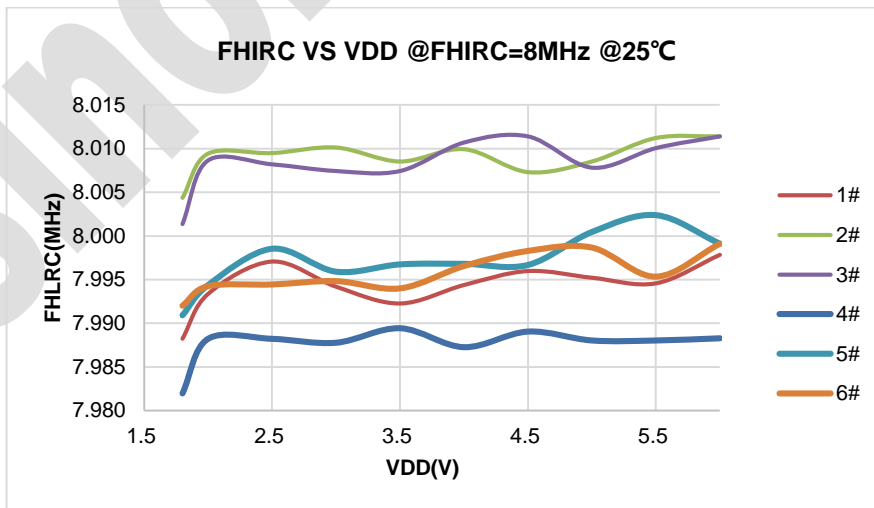
15.2.3 休眠模式 功耗 VS 电源电压

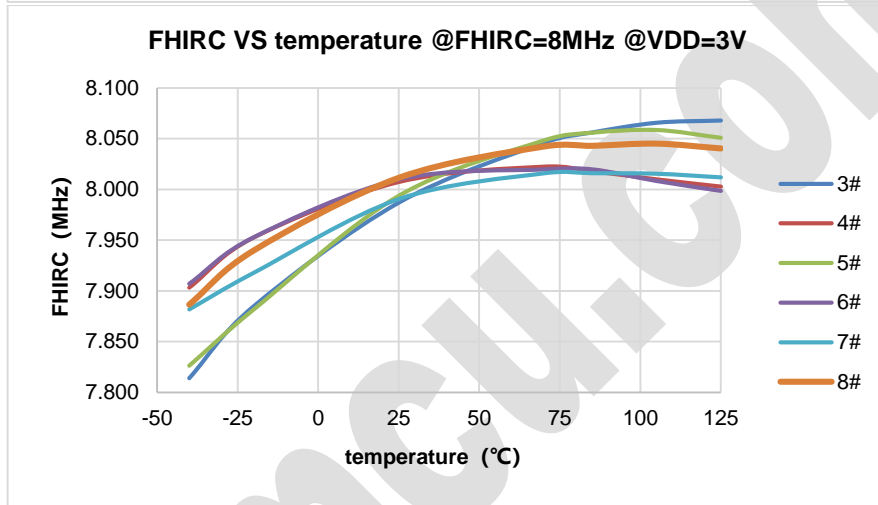
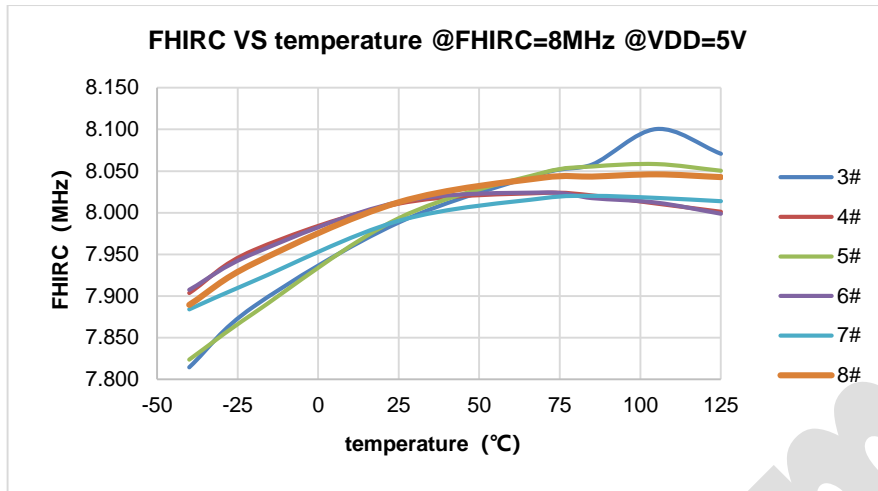




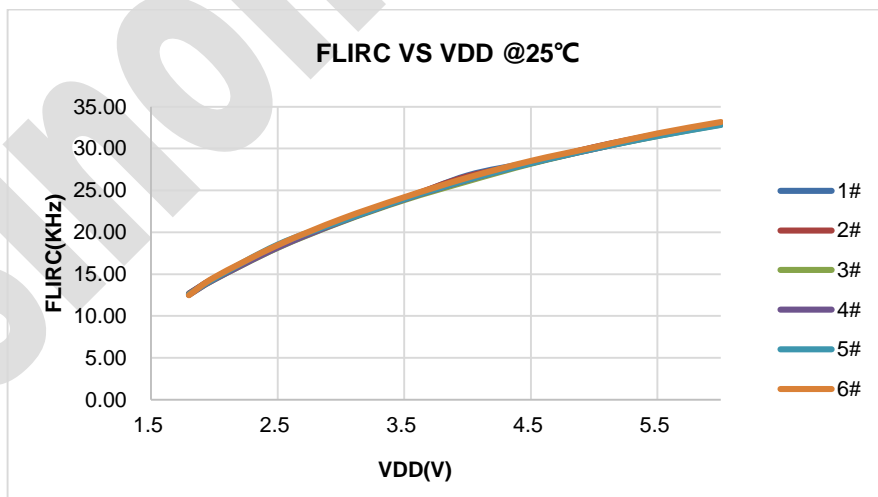
15.3 模拟电路特性

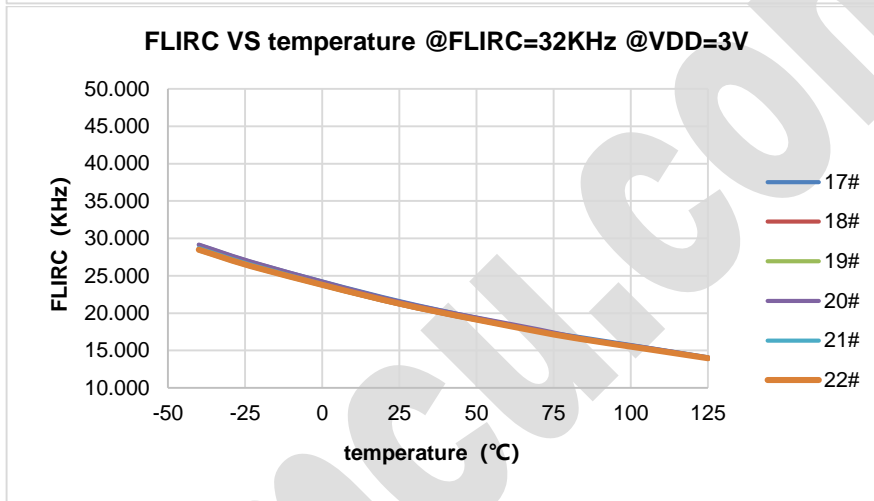
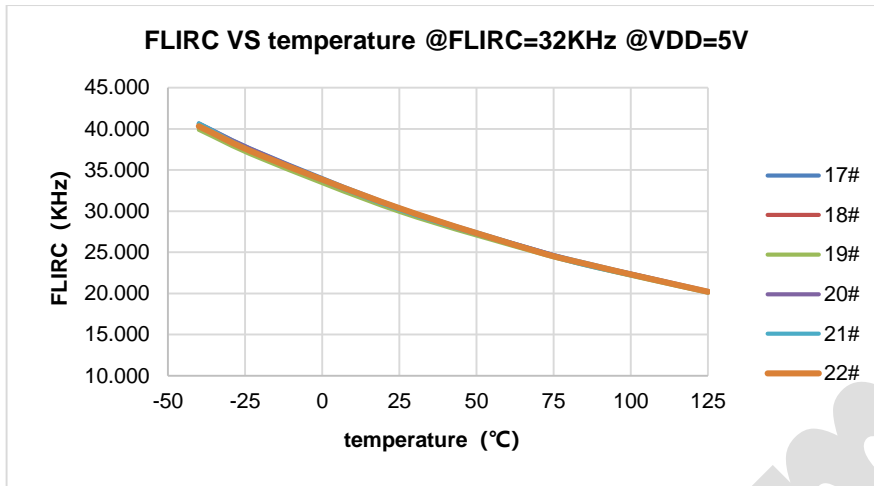
15.3.1 HIRC 频率 VS 电源电压/温度



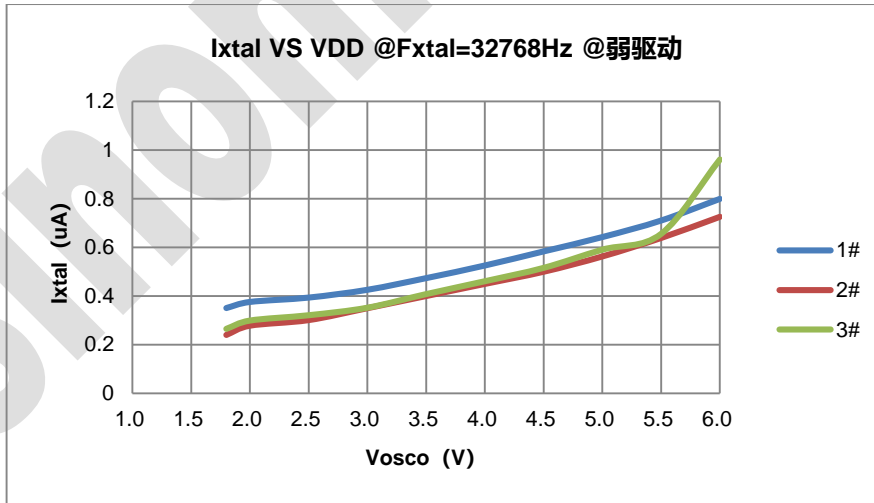


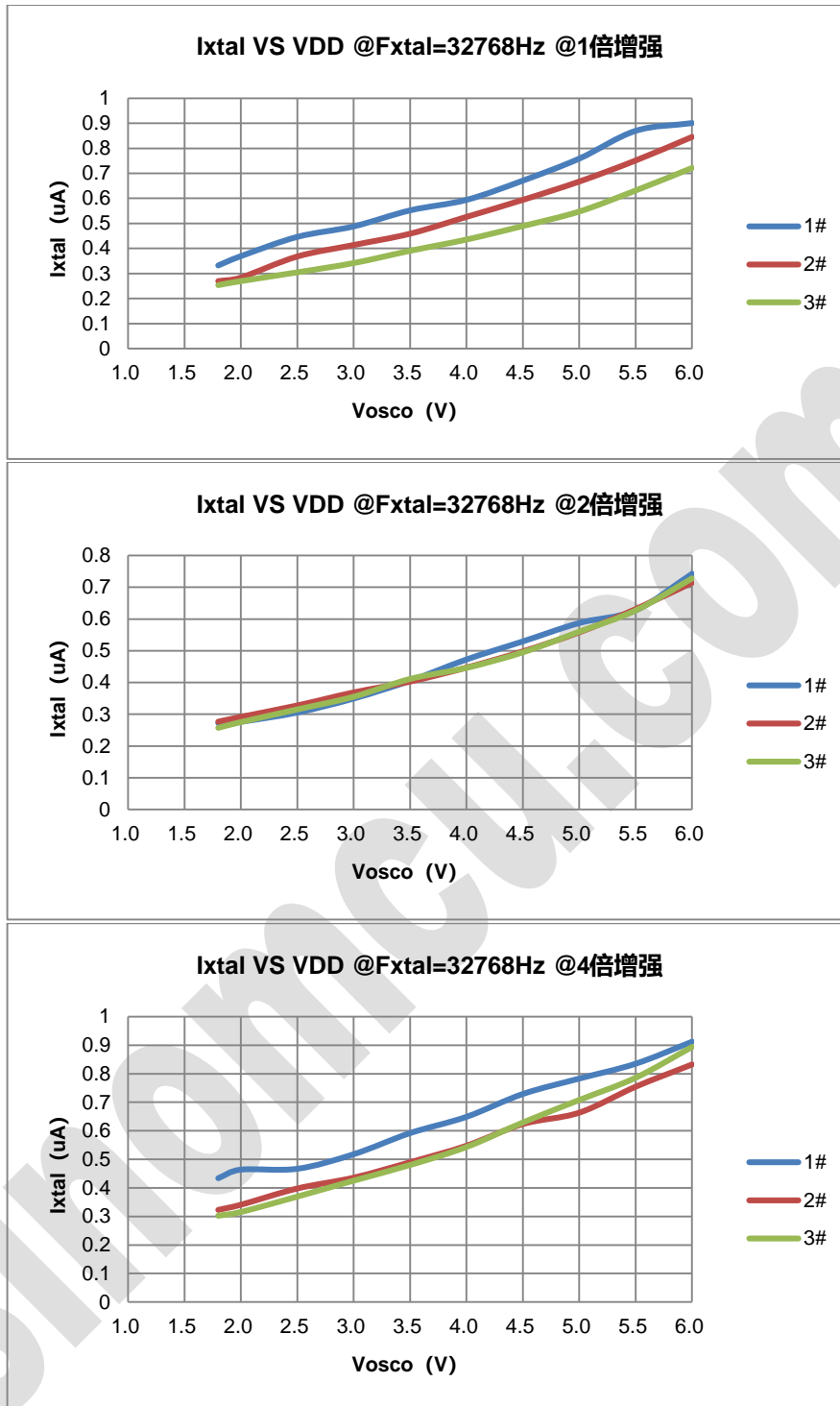
15.3.2 LIRC 频率 VS 电源电压/温度



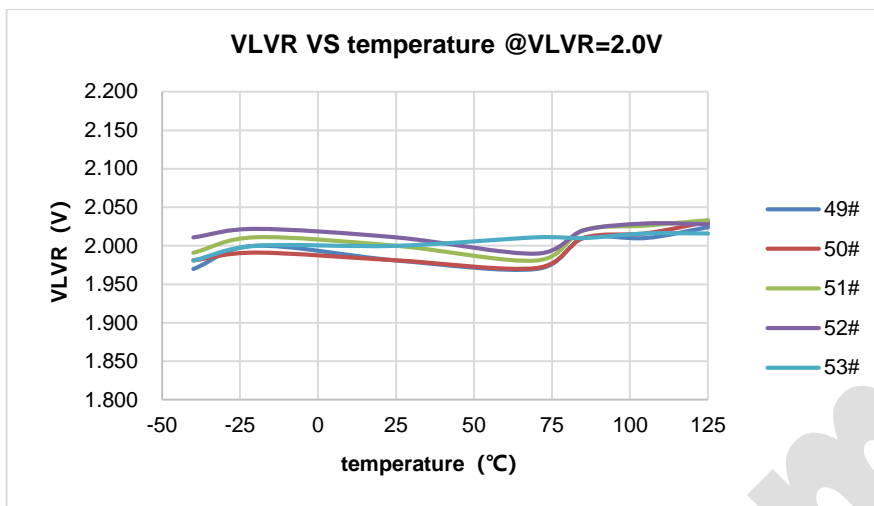


15.3.3 XTAL 功耗 VS 电源电压





15.3.4 LVR 阈值电压 VS 温度

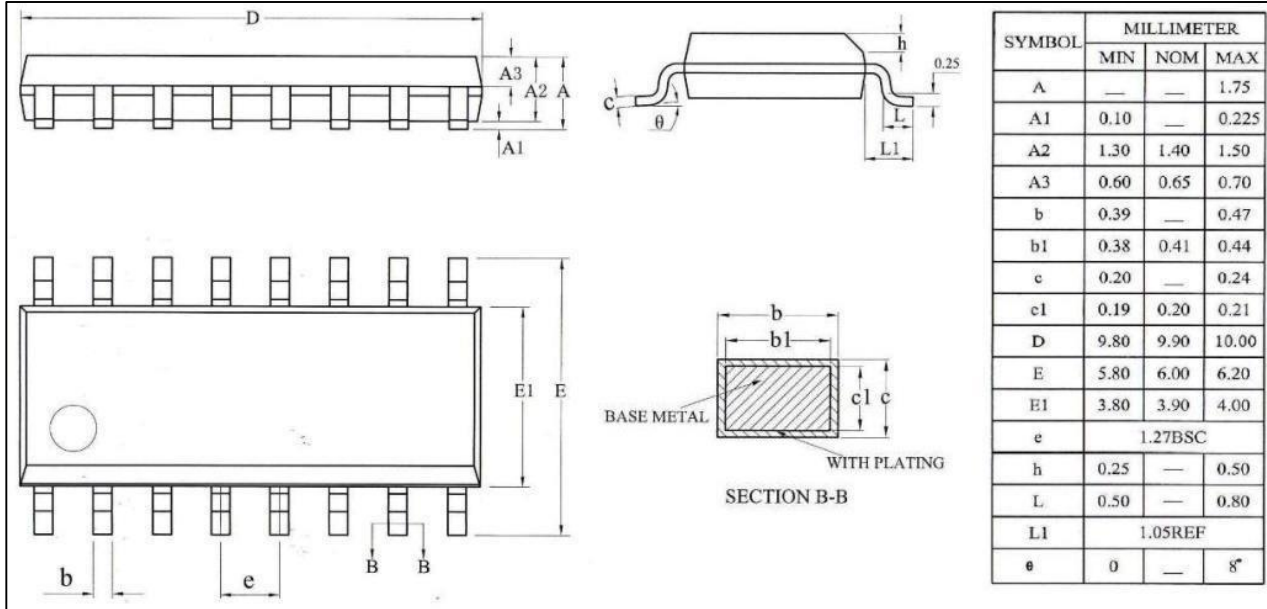


Shomcu.com

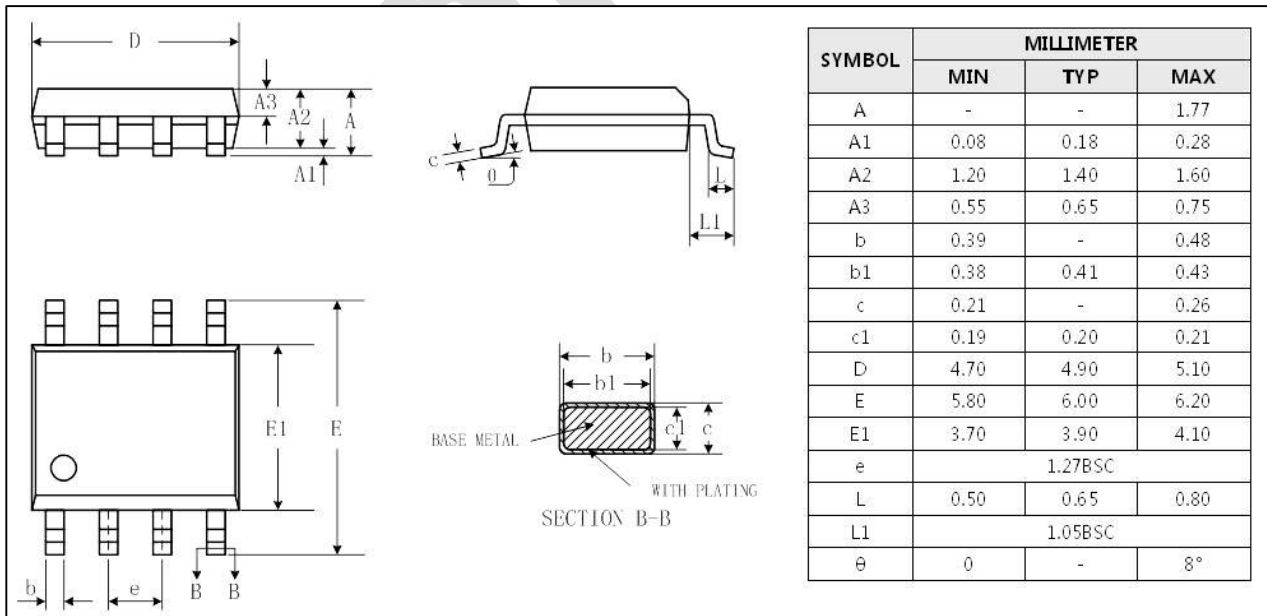


16 封装尺寸

16.1 SOP16



16.2 SOP8





17 修订记录

版本	日期	修订内容
V1.0	2023-07-24	发布初版;
V1.1.0	2023-08-14	修正特性曲线图中笔误;
V1.1.1	2023-10-10	1、更新温度传感器特性参数,精度调整为 $\pm 1^{\circ}\text{C}$ (tpy); 2、更新文档格式
V1.1.2	2024-02-21	1、ADC 章节新增温度传感器章节和 ADC 应用注意章节;共模输入/输出电压参数修调; 2、LDO 输出电流条件修正 3、产品特性章节,去除精度相关描述