



MC30F6920 用户手册

基于 RISC MC30 内核的 8 位 IO 型微控制器

Rev1.4.0

本资料内容为晟矽微电子（以下简称“我公司”）版权所有。

我公司将力求本资料的内容做到准确无误，但同时保留在不通知用户的情况下，对本资料内容的修改权。如您需要获得最新的资料，请及时联系我公司。

我公司将尽最大努力为您提供高品质、高稳定性的产品。尽管如此，由于一般半导体器件的电气敏感性及其易受到外部物理损伤等固有特性存在，所以难免造成半导体器件出现故障或失效的可能。当您使用我公司产品时，有责任按照本资料以及相关资料中提到的“规则”来设计一个稳定及安全可靠的系统环境。另外，在遇到超规格（本资料中未描述到内容）的使用，请您提前咨询我公司，以免因我公司产品在一些特殊设备中或者特殊环境下的使用，导致财产损失、人员伤亡等严重后果的发生。因超规格的使用、未经咨询授权我公司产品的使用，我公司对此不承担任何法律责任。



目录

1	产品概要	4
1.1	产品特性	4
1.2	订购信息	5
1.3	引脚排列	6
1.4	端口说明	7
2	电气特性	8
2.1	极限参数	8
2.2	直流电气特性	8
2.3	交流电气特性	9
2.4	EEPROM 特性参数	9
3	CPU 与存储器	11
3.1	指令集	11
3.2	程序存储器	13
3.3	数据存储器	13
3.4	堆栈	14
3.5	控制寄存器	14
3.6	用户配置字	16
4	系统时钟	18
4.1	内部高频 RC 振荡器	18
4.2	内部低频 RC 振荡器	18
4.3	系统工作模式	19
4.4	低功耗模式	20
5	复位	22
5.1	复位条件	22
5.2	上电复位	23
5.3	外部复位	23
5.4	低电压复位	23
5.5	看门狗复位	24
6	I/O 端口	25
6.1	通用 I/O 功能	25
6.2	内部上/下拉电阻	26
6.3	端口模式控制	27



7	定时器 TIMER	28
7.1	看门狗定时器 WDT	28
7.2	定时器 T0.....	29
7.3	定时器 T1.....	31
8	低电压检测 LVD.....	39
9	EEPROM 存储器	41
9.1	EEPROM 概述	41
9.2	EEPROM 相关寄存器	41
9.3	EEPROM 操作示例.....	42
10	FLASH 烧录编程	44
10.1	FLASH 在板编程	44
11	中断	45
11.1	外部中断.....	45
11.2	定时器中断	45
11.3	键盘中断.....	46
11.4	中断相关寄存器	46
12	特性曲线	48
12.1	I/O 特性	48
12.2	功耗特性	52
12.3	模拟电路特性.....	55
13	封装尺寸	58
13.1	SOP16	58
13.2	DIP16.....	58
13.3	SOP14	59
13.4	DIP14.....	59
14	修订记录	60



1 产品概要

1.1 产品特性

- 8 位 CPU 内核
 - ◇ 精简指令集，5 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 2/4/8/16/32/64 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
 - ◇ 1K×14 位 FLASH 型程序存储器
 - ◇ 支持在板不带电烧录编程，擦写次数至少 1000 次
- 数据存储器
 - ◇ 48 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
 - ◇ 64 字节 EEPROM 型数据存储器，支持单独烧录和软件读写，擦写次数至少 10000 次
- 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~P17 还可选推挽或开漏输出
 - ◇ P13 可复用为外部复位 RST 输入
 - ◇ 所有端口均内置上拉和下拉电阻，均可单独使能
 - ◇ P10 可复用为外部中断输入，支持外部中断唤醒功能
 - ◇ 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 (16MHz)，可用作系统高频时钟源
 - ◇ 内置低频 RC 振荡器 (32KHz)，可用作系统低频时钟源
- 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
 - ◇ 休眠模式（低功耗模式）：CPU 暂停，高频时钟源停止，低频时钟源由 WDT 等外设决定
- 内部自振式看门狗计数器 (WDT)
 - ◇ 与定时器 T0 共用预分频器
 - ◇ 溢出时间可配置：4.5ms/18ms/72ms/288ms（无预分频）
 - ◇ 工作模式可配置：始终开启、始终关闭，也可软件控制开启或关闭
- 2 个定时器



- ◇ 8 位定时器 T0，支持系统低频时钟，可实现外部计数功能，与 WDT 共用预分频器
- ◇ 8 位定时器 T1，可实现外部计数、4 路共周期独立占空比的 PWM（可组合成 2 对互反的带死区互补 PWM）
- 中断
 - ◇ 外部中断（INT），键盘中断（P00~P05，P10~P17）
 - ◇ 定时器中断（T0~T1）
- 低电压检测 LVD
 - ◇ 1.8V/2.0V/2.1V/2.2V/2.4V/2.6V/2.8V/2.9V/3.0V/3.1V/3.2V/3.3V/3.6V/3.8V/4.0V/4.2V
- 低电压复位 LVR
 - ◇ 1.8V/2.0V/2.7V/3.2V
- 工作电压
 - ◇ VLVR27 ~ 5.5V @ Fcpu = 0~8MHz
 - ◇ VLVR20 ~ 5.5V @ Fcpu = 0~4MHz
 - ◇ VLVR18 ~ 5.5V @ Fcpu = 0~1MHz
- 封装形式
 - ◇ SOP16/DIP16/SOP14/DIP14

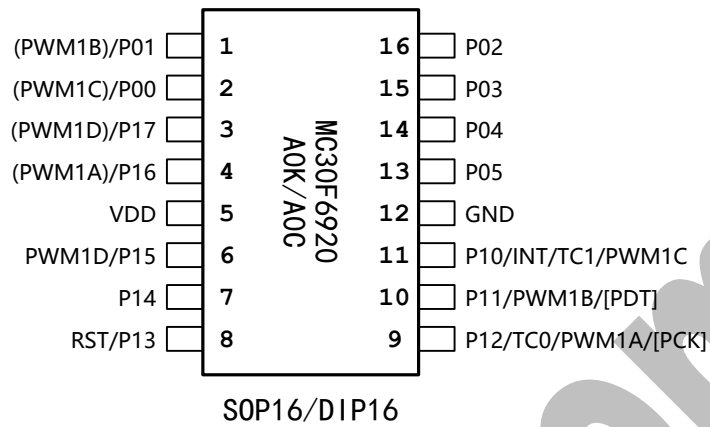
1.2 订购信息

产品名称	封装形式	备注
MC30F6920A0K	SOP16	
MC30F6920A0C	DIP16	
MC30F6920A1K	SOP16	
MC30F6920A1C	DIP16	
MC30F6920A2C	DIP16	
MC30F6920A0J	SOP14	
MC30F6920A0B	DIP14	
MC30F6920A0H	SOP8	

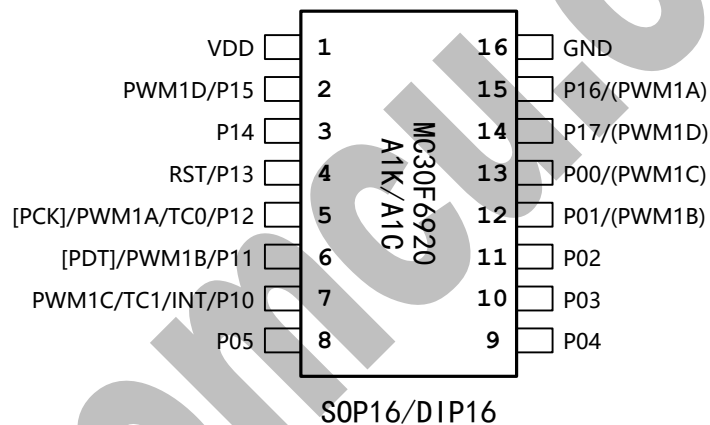


1.3 引脚排列

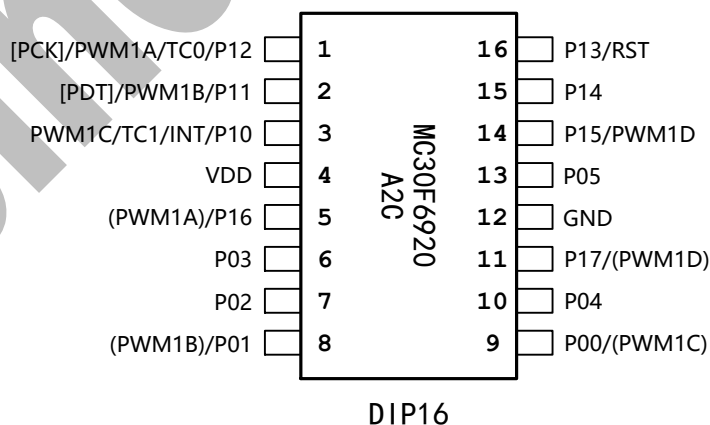
MC30F6920A0K/A0C



MC30F6920A1K/A1C

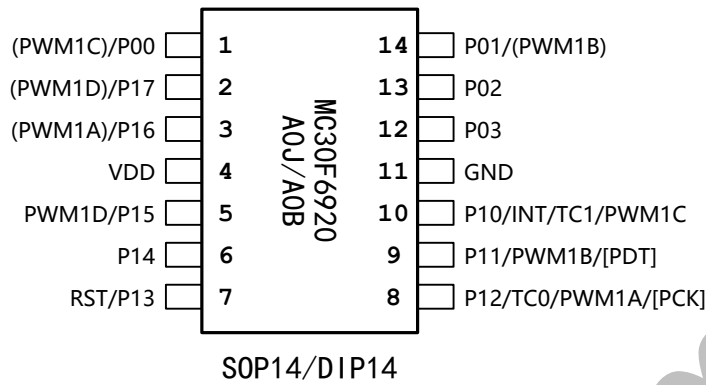


MC30F6920A2C

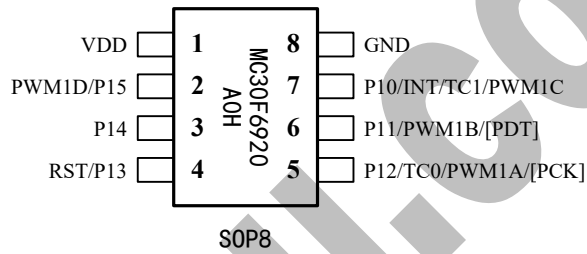




MC30F6920A0J/A0B



MC30F6920A0H



1.4 端口说明

端口名称	类型	功能说明
VDD	P	电源
GND	P	地
P00~P05	D	GPIO（推挽输出），内部上/下拉
P10~P17	D	GPIO（推挽输出或开漏输出），内部上/下拉
INT	DI	外部中断输入
TC0~TC1	DI	定时器 T0~T1 的外部计数输入
PWM1A~PWM1D	DO	定时器 T1 的 4 路 PWM 输出
RST	DI	外部复位输入
PCK, PDT	D	编程时钟/数据接口

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



2 电气特性

2.1 极限参数

参数	符号	值	单位
电源电压	VDD	-0.3~6.0	V
I/O 输入电压	Vin	-0.3~VDD+0.3	V
工作温度	Ta	-40~85	°C
储存温度	Tstg	-65~150	°C
流入 VDD 最大电流	IVDDmax	50	mA
流出 GND 最大电流	IGNDmax	50	mA

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

2.2 直流电气特性

VDD=5V, T=25°C

特性	符号	端口	条件	最小	典型	最大	单位
工作电压	VDD	VDD	Fcpu=8MHz@FHIRC/2	VLVR27		5.5	V
			Fcpu=4MHz@FHIRC/4	VLVR20		5.5	
			Fcpu=2MHz@FHIRC/8	VLVR20		5.5	
			Fcpu=1MHz@FHIRC/16	VLVR18		5.5	
			Fcpu=500KHz@FHIRC/32	VLVR18		5.5	
			Fcpu=250KHz@FHIRC/64	VLVR18		5.5	
			Fcpu=16KHz@FLIRC/2	VLVR18		5.5	
输入漏电流	Ileak	所有输入脚	VDD=5V	-1		1	μA
输入高电平	Vih	所有输入脚	SMT 开启, SMTVS 配置	0.8VDD			V
			SMT 开启, SMTVS 配置	2.0			
			SMT 关闭	2.0			
输入低电平	Vil	所有输入脚	SMT 开启, SMTVS 配置			0.2VDD	V
			SMT 开启, SMTVS 配置			0.8	
			SMT 关闭			1.0	
上拉电阻	Rpu	P1	VDD=5V, Vin=0		20		KΩ
下拉电阻	Rpd	P1	Vin=VDD=5V		20		KΩ



输出源电流	Ioh	推挽输出脚	Voh=VDD-0.6V, IOHS 选择限流输出		3		mA
			Voh=VDD-0.6V, IOHS 选择正常输出		12		mA
输出灌电流	Iol	所有输出脚	Vol=0.6V		20		mA
运行模式功耗	Irun	VDD	Fcpu=8MHz@HIRC		1.6		mA
			Fcpu=4MHz@HIRC		0.9		mA
			Fcpu=2MHz@HIRC		620		μA
			Fcpu=1MHz@HIRC		480		μA
			Fcpu=500KHz@HIRC		410		μA
			Fcpu=250KHz@HIRC		380		μA
			Fcpu=32KHz/2@LIRC(HIRC 关)		10		μA
休眠模式功耗	Istop	VDD	休眠模式, LIRC 关		0.6	3	μA
			休眠模式, LIRC 开		2		μA
低压检测电压	VLVD	VDD	LVDVS 选择	-10%		+10%	V
LVD 响应时间	TLVD			1	50	200	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

特性	符号	条件	最小	典型	最大	单位
HIRC 振荡频率	FHIRC	VDD=2.0V~5.5V, T=25°C	-3%	16	+3%	MHz
		VDD=2.0V~5.5V, T=-40°C~85°C	-5%		+5%	
LIRC 振荡频率	FLIRC	VDD=5V, T=25°C	-50%	32	+50%	KHz

2.4 EEPROM 特性参数

特性	符号	条件	最小	典型	最大	单位
EEPROM 读操作电压	VEERD	T=-40°C~85°C	2.0		5.5	V
EEPROM 写操作电压	VEEWR	T=-40°C~85°C	2.0		5.5	V



EEPROM 写操作电流	I _{EEWR}	T=-40°C~85°C		2		mA
EEPROM 静态电流	I _{EESTP}	T=25°C		1		μA
EEPROM 单字节写入时间	T _{EEWR}	VDD=2.0V~5.5V, T=-40°C~85°C		5		ms
EEPROM 擦写次数		VDD=5V, T=25°C	10000			cycle



3 CPU 与存储器

3.1 指令集

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

指令汇总表

助记符	说明	操作	周期	长度	标志
ADDAR R	R 和 A 相加，结果存入 A	$R+A \rightarrow A$	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
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
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
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
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
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
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	-
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
ISUBAI	I	I 和 A 相减, 结果存入 A	$I-A \rightarrow A$	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	$PC+1 \rightarrow TOS$; $K \rightarrow PC[10:0]$	2	1	-
GOTO	K	无条件跳转	$K \rightarrow PC[10:0]$	2	1	-
RETURN		从子程序返回	$TOS \rightarrow PC$	2	1	-
RETAI	I	从子程序返回, 并将 I 存入 A	$TOS \rightarrow PC$; $I \rightarrow A$	2	1	-
RETIE		从中断返回	$TOS \rightarrow PC$; $1 \rightarrow GIE$	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	$A(\text{HEX 码}) \rightarrow A(\text{BCD 码})$	1	1	C
DSA		BCD 码减法操作后, 将 A 的值调整为 BCD 码	$A(\text{HEX 码}) \rightarrow A(\text{BCD 码})$	1	1	-
CLRWDI		将看门狗计数器清零	$0 \rightarrow WDT CNT$	1	1	TO,PD
STOP		进入低功耗模式	$0 \rightarrow WDT CNT$; 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;



3.2 程序存储器

芯片的程序存储器为 FLASH 型存储器，1K×14 位的地址空间范围为 0000H~03FFH。程序存储器地址分配如下图所示：

复位起始地址（0000H）
通用程序区 （0001H - 0007H）
中断入口地址（0008H）
通用程序区 （0009H - 03FFH）

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（48 字节）和特殊功能寄存器 SFR，地址映射如下表所示。GPR 和 SFR 均可直接寻址或通过 INDF 间接寻址。数据存储器还包括掉电非易失的 EEPROM 型数据存储器（64 字节），需通过 SFR 进行读写操作，详细说明请参见后续章节。

数据存储器区地址映射表

地址	类型	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
00H-07H	SFR	INDF	T0CNT	PCL	STATUS	FSR	P0	P1	
08H-0FH		MCR	T0CR	PCLATH	PDCON1	PUCON1	ODCON1	INTE	INTF
10H-3FH	GPR	通用数据存储器区							
40H-47H	SFR	P0KBCR	P1KBCR	OSCMR	PDCON0	PUCON0	DDR0	DDR1	
48H-4FH					TMRCR	T1CR	T1CNT	T1LOAD	T1DATA
50H-57H		T1DATB	T1DATC	T1DATD			PWM1CR0	PWM1CR1	PWM1CR2
58H-5FH						EECR	EEPR	EEAR	EEDR
60H-7FH	保留								

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



数据存储器寻址方式地址组成

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

直接寻址方式，是以指令的低 7 位为数据存储器地址，通过指令访问，寻址范围 0~7FH。例如，采用直接寻址方式将数据 55H 写入数据存储器 10H 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，是以 FSR 为数据存储器地址指针，通过 INDF 访问，寻址范围 0~7FH。例如，采用间接寻址方式将数据 55H 写入数据存储器 10H 地址中：

MOVAI 10H

MOVRA FSR

MOVAI 55H

MOVRA INDF ； 将 55H 写入 FSR 所指地址的数据存储器中

3.4 堆栈

芯片的堆栈为 5 级深度的硬件堆栈。当 CPU 响应中断或执行子程序调用指令时，会自动将下一条指令的 PC 值压栈保存；当 CPU 执行中断返回或子程序返回指令时，会自动将栈顶内容出栈载入 PC。

3.5 控制寄存器

数据指针寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSR	-	FSR6	FSR5	FSR4	FSR3	FSR2	FSR1	FSR0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	X	X	X	X	X	X	X

BIT[6:0] FSR[6:0] – 数据指针寄存器

FSR：间接寻址方式的指针。



间接寻址寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INDF	INDF7	INDF6	INDF5	INDF4	INDF3	INDF2	INDF1	INDF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] INDF[7:0] – 间接寻址寄存器

INDF: INDF 不是物理寄存器, 对 INDF 操作实际是对 FSR 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。

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

程序指针计数器高位缓存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCLATH	-	-	-	-	-	-	PCH1	PCH0
R/W	-	-	-	-	-	-	R/W	R/W
初始值	-	-	-	-	-	-	0	0

BIT[1:0] PCH[1:0] – 程序指针计数器高 2 位缓存器 (仅 C 编译模式下有效)

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: $PC = \text{指令码低 10 位}$;
- ✧ 返回指令 RETIE/RETURN/RETAI: $PC = \text{堆栈栈顶 (TOS)}$;

对 PCL 操作指令 (汇编模式):

- ✧ 对 PCL 操作的加法指令: $PC = (PC[9:0] + ALU[7:0])$;
- ✧ 对 PCL 操作的其他指令: $PC = (PC[9:8]:ALU[7:0](ALU \text{ 运算结果}))$;

对 PCL 操作指令 (C 编译模式):

- ✧ 对 PCL 操作的所有指令: $PC = (PCLATH[1:0]:ALU[7:0](ALU \text{ 运算结果}))$;



CPU 状态寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	WKUP	-	-	TO	PD	Z	DC	C
R/W	R/W	-	-	R	R	R/W	R/W	R/W
初始值	0	-	-	1	1	X	X	X

BIT[7] WKUP – 唤醒中断源标志位

- 0: 芯片被其他中断唤醒;
- 1: 芯片被外部中断或键盘中断唤醒;

BIT[4] TO – 看门狗溢出标志位

- 0: 发生 WDT 溢出;
- 1: 上电复位, 或执行 CLRWDT/STOP 指令;

BIT[3] PD – 进入低功耗模式标志位

- 0: 执行 STOP 指令;
- 1: 上电复位, 或执行 CLRWDT 指令;

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;

3.6 用户配置字

芯片为保证系统正常工作, 会将关键模块的配置信息预先存储于单独的存储器区域内, 在上电或其他复位发生后将配置信息载入寄存器中, 通过寄存器控制关键模块的工作状态。该部分存储器中用户可



选的内容即为用户配置字，可在烧录用户程序代码时进行配置与烧录。

芯片的用户配置字，定义如下：

符号	功能说明
FCPUS	高频时钟下 F _{CPU} 分频选择： F _{CPU} =F _{HOSC} /2; F _{CPU} =F _{HOSC} /4; F _{CPU} =F _{HOSC} /8; F _{CPU} =F _{HOSC} /16; F _{CPU} =F _{HOSC} /32; F _{CPU} =F _{HOSC} /64;
CLKSET	系统复位初始工作模式选择： 系统复位后初始为高速模式（CLKS 初值为 0，系统高频时钟源开启）； 系统复位后初始为低速模式（CLKS 初值为 1，系统高频时钟源关闭）；
RSTEN	RST 外部复位端口设置： P13 为外部复位脚； P13 为输入/输出脚；
LVRVS	LVR 复位电压选择：（LVR 电压应满足由 F _{CPU} 决定的工作电压特性） 1.8V; 2.0V; 2.7V; 3.2V;
WDTM	WDT 模式设置： WDT 始终关闭； WDT 始终开启；
WDTT	WDT 溢出时间（无预分频）选择： 4.5ms; 18ms; 72ms; 288ms;
SMTEN	端口输入施密特设置： 输入 SMT 功能无效； 输入 SMT 功能有效；
SMTVS	端口施密特阈值选择： 2.0V/0.8V; 0.8VDD/0.2VDD;
IOHS	端口源电流输出选择： 源电流正常输出； 源电流限流输出；
PWM1PINS	PWM1 端口复用选择： PWM1A/PWM1B/PWM1C/PWM1D 复用于 P16/P01/P00/P17; PWM1A/PWM1B/PWM1C/PWM1D 复用于 P12/P11/P10/P15;
MCUSEL	芯片模式选择：（C 编译模式支持 C 程序对常量数组的定义，汇编模式则不支持） 汇编模式； C 编译模式；
ENCR	程序代码加密设置： 程序代码加密； 程序代码不加密；



4 系统时钟

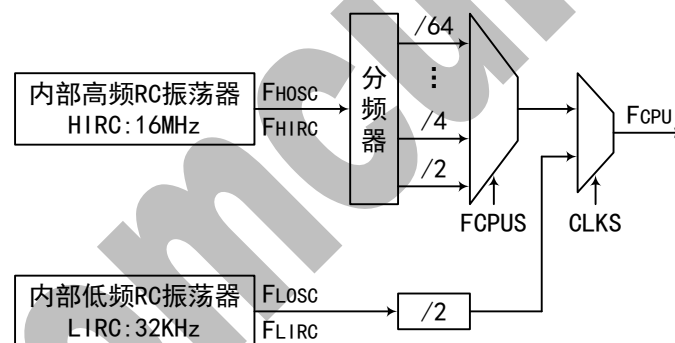
芯片内部电路均在系统高频时钟 F_{HOSC} 或系统低频时钟 F_{LOSC} 下工作，系统及部分外设模块的时钟源还可在 F_{HOSC} 和 F_{LOSC} 之间切换。

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

CPU 的时钟源可在系统高频时钟 F_{HOSC} 和系统低频时钟 F_{LOSC} 之间切换。 F_{HOSC} 下 CPU 的时钟频率 F_{CPU} 通过配置字 $FCPUS$ 选择； F_{LOSC} 下 F_{CPU} 则固定为 F_{LOSC} 的 2 分频。

WDT（看门狗）电路的时钟源固定为内部低频 RC 振荡器 LIRC。

系统时钟示意图



4.1 内部高频 RC 振荡器

芯片内置 1 个振荡频率为 16MHz 的高精度 HIRC 振荡器，可用作系统高频时钟源。

4.2 内部低频 RC 振荡器

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



4.3 系统工作模式

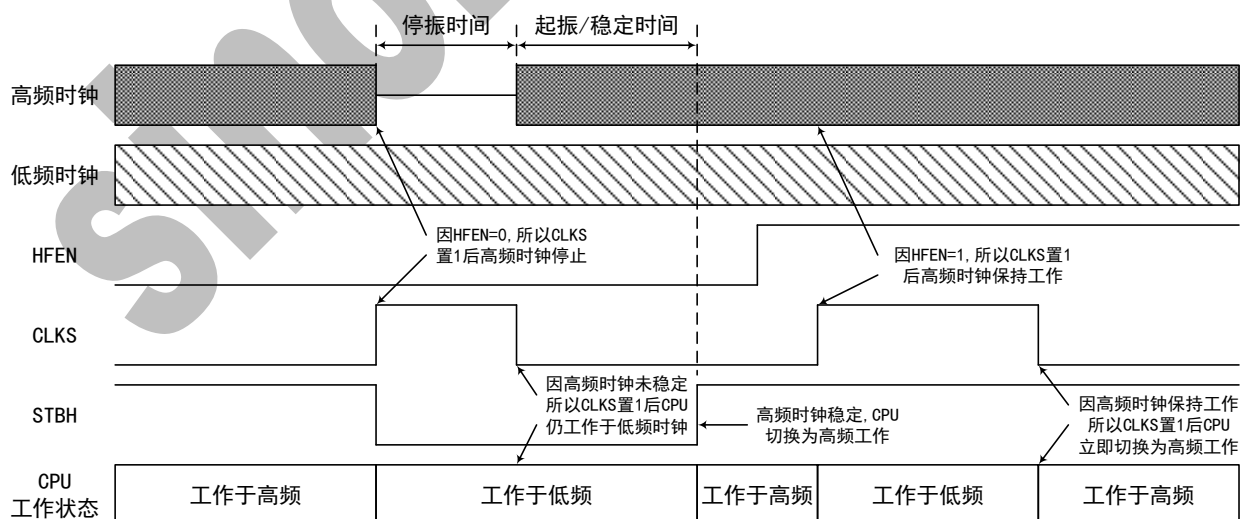
芯片支持高速模式、低速模式、休眠模式等多种系统工作模式。

工作模式	模式切换条件	系统工作状态
高速	任意模式下，系统复位	CPU 高速运行，高/低频时钟源均工作
	低速模式下，CLKS 清 0	
	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
	休眠模式下，CPU 唤醒 (@CLKS=1)	
休眠	高/低速模式下，执行 STOP 指令 (@HFEN=0)	CPU 暂停，高频时钟源停止，低频时钟源由 WDT/T0 决定

工作模式切换示意图



高低频时钟切换时序图



振荡器模式寄存器



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

BIT[5] STBL – 低频时钟源振荡状态标志位

- 0: 低频时钟源停振或未稳定;
- 1: 低频时钟源已稳定振荡;

BIT[4] STBH – 高频时钟源振荡状态标志位

- 0: 高频时钟源停振或未稳定;
- 1: 高频时钟源已稳定振荡;

BIT[2] CLKS – CPU 时钟源选择位 (复位初始值由配置字 CLKSET 决定)

- 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[0] HFEN – 高频时钟源使能位

- 0: 在低速模式下, 高频时钟源暂停工作;
- 1: 在低速模式下, 高频时钟源始终工作;

4.4 低功耗模式

芯片的高速模式、低速模式为运行模式, 而休眠模式则为低功耗模式。

执行 STOP 指令可使系统进入低功耗模式, 同时对系统会产生以下影响:

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

以下情况可使系统退出低功耗模式:



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

注：

- 1、低功耗模式下触发中断请求时，若对应的中断使能位关闭，则不会退出低功耗模式；若对应的中断使能位开启而中断总使能位关闭，则仅唤醒 CPU 执行下一条指令；若对应的中断使能位和中断总使能位均开启，则唤醒 CPU 后将执行中断服务程序；
- 2、未使用或未封出的引脚，应将其对应的 I/O 端口设置为输出、输入上拉或输入下拉等稳定状态，避免因引脚浮空而产生漏电流或非预期的中断唤醒；



5 复位

5.1 复位条件

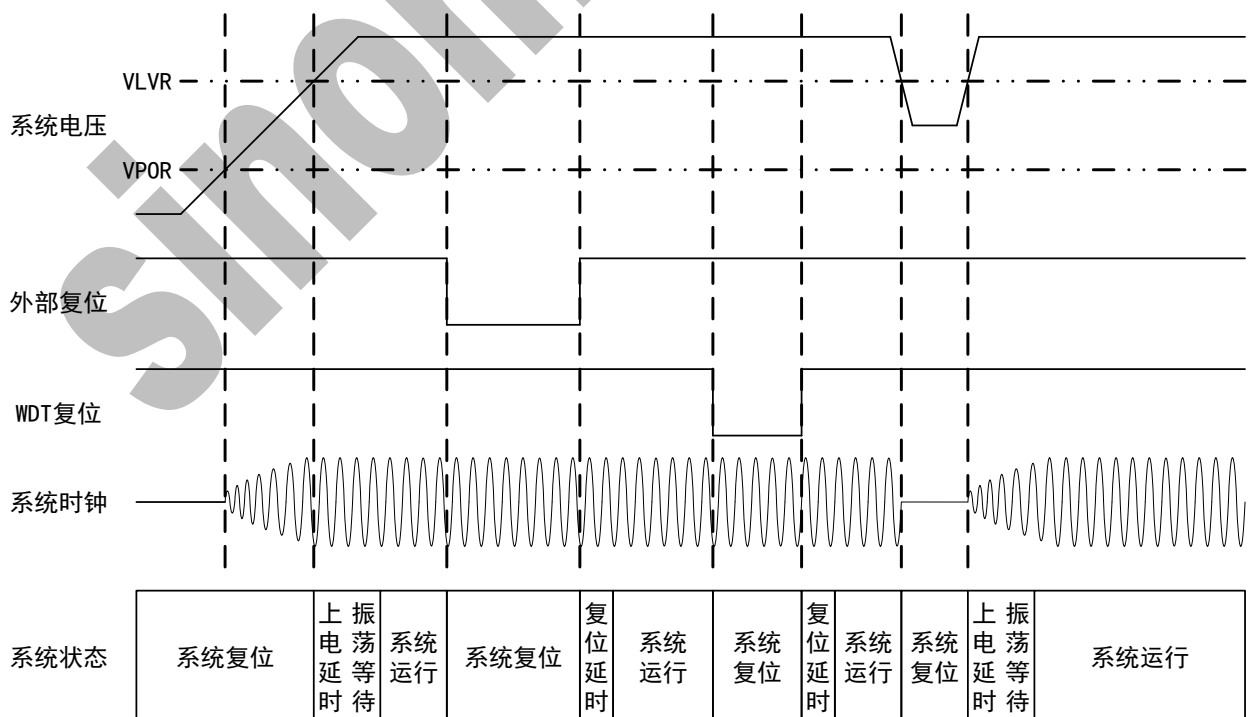
芯片共有如下几种复位方式：

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

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

上电复位 POR 和低电压复位 LVR 会关闭系统主时钟振荡器，复位解除后才重新开启振荡器，因为振荡器起振和稳定需要一定的时间，所以系统将保持一定时间的上电延时（典型值为 16ms）以待振荡器稳定振荡后才开始工作；而外部复位、WDT 复位则不会关闭主时钟振荡器，复位解除后系统将在较短的复位延时（典型值为 2ms）后即开始工作。

下图是复位产生和系统工作状态之间的时序关系示意图：





注：若应用系统在上电或掉电回升时芯片的 VDD 电压上升较慢，则应在复位后 CPU 开始工作时先进行软件延时，以确保芯片开始工作时 VDD 电压已稳定在 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 V_{LVR} 可通过配置字 LVRVS 选择。LVR 检测电路具有一定的回滞特性，回滞电压约为 6%（典型值），当电源电压下降至 V_{LVR} 时发生 LVR 复位，反之电源电压需上升至 $V_{LVR}+6\%$ 后 LVR 复位才解除。

LVR 在运行模式下自动开启，在低功耗模式下自动关闭。



5.5 看门狗复位

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

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



6 I/O 端口

6.1 通用 I/O 功能

芯片的输入/输出端口为一组 6 位端口 P1。所有端口均支持施密特输入，均可选推挽或开漏输出。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 输出等功能。

端口数据寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0	-	-	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
P1	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)

端口方向寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DDR0	-	-	DDR05	DDR04	DDR03	DDR02	DDR01	DDR00
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	1	1	1	1	1	1

BIT[5:0] DDR0n – P0n 端口方向控制位 (n=5-0)

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

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DDR1	DDR17	DDR16	DDR15	DDR14	DDR13	DDR12	DDR11	DDR10
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] DDR1n – P1n 端口方向控制位 (n=7-0)



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

6.2 内部上/下拉电阻

所有端口均具有内部上拉和下拉电阻，且均可单独控制其上/下拉电阻在端口处于输入状态时是否有效。端口处于输出状态时，上/下拉电阻及其控制位无效（但端口处于开漏输出高电平状态时上拉电阻及其控制位依然有效）。

上拉电阻控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUCON0	-	-	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	1	1	1	1	1	1

BIT[5:0] P0nPU – P0n 端口上拉电阻控制位 (n=5-0)

- 0: 端口内部上拉电阻有效；
1: 端口内部上拉电阻无效；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUCON1	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	1	1	1	1	1	1	1	1

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

- 0: 端口内部上拉电阻有效；
1: 端口内部上拉电阻无效；

下拉电阻控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDCON0	-	-	P05PD	P04PD	P03PD	P02PD	P01PD	P00PD
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	1	1	1	1	1	1

BIT[5:0] P0nPD – P0n 端口下拉电阻控制位 (n=5-0)



0: 端口内部下拉电阻有效;

1: 端口内部下拉电阻无效;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDCON1	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	1	1	1	1	1	1	1	1

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

0: 端口内部下拉电阻有效;

1: 端口内部下拉电阻无效;

6.3 端口模式控制

用作数字输出口时，P1 所有端口均可选择推挽输出或开漏输出。当端口为推挽输出、或开漏输出低时，输入通路保持连接；而当端口开漏输出高时，输入通路将自动关断。

端口输出模式寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ODCON1	P17OD	P16OD	P15OD	P14OD	P13OD	P12OD	P11OD	P10OD
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] P1nOD – P1n 端口开漏输出控制位 (n=7-0)

0: 端口输出时为推挽输出;

1: 端口输出时为开漏输出;



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, 可通过预分频器选择不同的 WDT 计数时钟频率。WDT 溢出将复位芯片或唤醒 CPU。

可通过配置字 WDTM 和寄存器位 WDTEN 共同决定是否开启 WDT: WDTM 选择始终关闭、或 WDTEN 为 0, 则 WDT 关闭; 而当 WDTM 选择始终开启且 WDTEN 为 1 时, WDT 才开启。若 WDT 开启, 则在休眠模式下 WDT 依然工作且溢出将唤醒 CPU, 而在 CPU 运行时 WDT 溢出将复位芯片。

WDT 和定时器 T0 共用一个预分频器, 并通过寄存器位决定预分频器的分配。当预分频器分配给 T0 时, WDT 时钟不分频; 而预分频器分配给 WDT 时, T0 时钟不分频。

执行 CLRWDT 指令或 STOP 指令将清零 WDT 计数器, 若预分频器分配给 WDT, 则还将清零预分频计数器 (预分频比不变)。

WDT 的基础溢出时间 (即无预分频的时间) 可配置为 4.5ms/18ms/72ms/288ms。

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。

杂项控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCR	WDTEN	EIS	LVDF	LVDVS3	LVDVS2	LVDVS1	LVDVS0	LVDEN
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
初始值	1	0	0	0	0	0	0	0

BIT[7] WDTEN – 看门狗使能位

0: 关闭 WDT;

1: 开启 WDT;

BIT[6] EIS – INT 端口外部中断功能使能位

0: 关闭端口的外部中断功能;

1: 使能端口的外部中断功能;

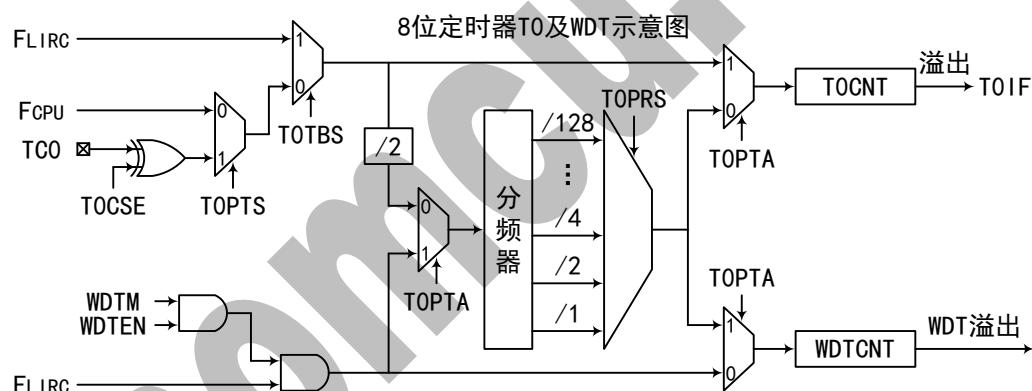


- BIT[5] LVDF – LVD 检测状态标志位
- BIT[4:1] LVDVS[3:0] – LVD 电压检测阈值选择位
- BIT[0] LVDEN – 低电压检测 LVD 使能位

7.2 定时器 T0

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

- ◇ 可通过预分频器设置时钟频率；
- ◇ 时钟源可选 F_{CPU}、F_{LIRC}、或外部时钟（TC0 输入）；
- ◇ 支持溢出中断和溢出唤醒功能；



TOCNT 为 8 位可读写的递增计数器，计数溢出到 0 时产生溢出信号并触发中断，中断标志 T0IF 将被置 1。

预分频器为 T0 与 WDT 共用，通过寄存器位 TOPTA 控制预分频器的分配。

TOPTA=0 时，预分频器分配给 T0 使用，T0 周期（溢出时间）= 预分频比 / T0 计数时钟频率，写 TOCNT 将清零预分频计数器，而执行 CLRWDT 或 STOP 指令则不影响预分频器的计数。

TOPTA=1 时，预分频器分配给 WDT 使用，执行 CLRWDT 或 STOP 指令将清零预分频计数器，而写 TOCNT 则同样不影响预分频器的计数。

通过 TOPTA 改变预分频器的分配时，也将清零预分频计数器。



清零预分频计数器的操作不会改变预分频比，而改变预分频比也不会清零预分频计数器。

当通过 T0TBS 选择 FLIRC 作为 T0 时钟时，在低功耗模式下 T0 将继续工作，溢出可唤醒。

定时器控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMRCR	-	T0TBS	-	-	-	-	T1IE	T1IF
R/W	-	R/W	-	-	-	-	R/W	R/W
初始值	-	0	-	-	-	-	0	0

BIT[6] T0TBS – T0 时钟源选择位

0: T0 时钟源由 T0PTS 决定；

1: T0 时钟源为内部低频时钟 FLIRC；

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] 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；

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	-	INTM	T0PTS	T0CSE	T0PTA	T0PRS2	T0PRS1	T0PRS0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	0	1	1	1	1	1	1

BIT[6] INTM – 外部中断 INT 触发方式选择位

0: 下降沿触发；

1: 上升沿触发；

BIT[5] T0PTS – T0 时钟源选择位（仅 T0TBS=0 时有效）

0: T0 时钟源为 F_{CPU}；



1: T0 时钟源为 TC0 输入的外部时钟（端口需设为输入状态）；

BIT[4] T0CSE – T0 外部时钟计数沿选择位

0: T0 在外部时钟上升沿计数；

1: T0 在外部时钟下降沿计数；

BIT[3] T0PTA – 预分频器分配控制位

0: 预分频器分配给 T0；

1: 预分频器分配给 WDT；

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 (T0PTA=0)	WDT 时钟预分频比 (T0PTA=1)
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

定时器 T0 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0CNT	T0CNT7	T0CNT6	T0CNT5	T0CNT4	T0CNT3	T0CNT2	T0CNT1	T0CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

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

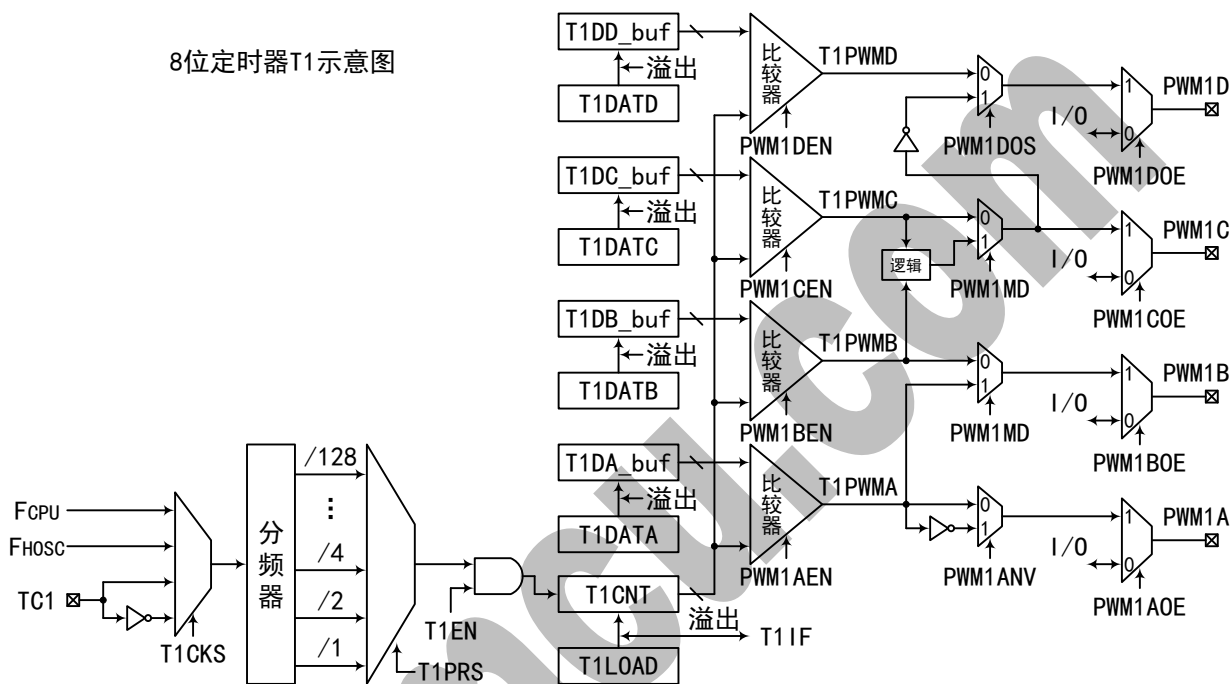
7.3 定时器 T1

定时器 T1 为 8 位定时/计数器，包含 1 个 8 位递减计数器、可编程预分频器、控制寄存器、8 位重

载寄存器和 4 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 4 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；
- ✧ 可组合成 2 对互反的带死区互补 PWM；
- ✧ 支持溢出中断和溢出唤醒功能；

8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。

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。

如图所示，定时器 T1 可实现 4 路共周期的 PWM 功能（PWM1x，x=A,B,C,D，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。PWM1x 关闭时 T1PWMx 信号为低电平。PWM1x 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 T1DATx 相等时，T1PWMx 变为高电平；当计数溢出时，T1PWMx 变为低电平。



T1DATx 均配有 1 个 8 位比较缓冲器 (T1Dx_buf) 用于与 T1CNT 比较, PWM1x 关闭时写 T1DATx 将立即载入缓冲器中, 而 PWM1x 使能后写 T1DATx 则将在 T1 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。

T1PWMx 信号 (x=A,B,C,D, 下同) 的占空比计算如下:

- ✧ 高电平时间 = (T1DATx) × 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) = (T1DATx) / (T1LOAD + 1)

PWM1x 的占空比可通过寄存器位 PWM1xDB 微调半个计数时钟周期。PWM1xDB=1 时, PWM 周期内 T1PWMx 信号在计数时钟上升沿变为高电平; PWM1xDB=0 时, PWM 周期内 T1PWMx 信号滞后半个时钟周期在时钟下降沿变为高电平。

如图所示, 当 PWM1MD=1 时, 端口 PWM1B 输出 T1PWMA 信号, 端口 PWM1C 输出 T1PWMB 和 T1PWC 的组合逻辑(同或/异或)信号, 从而组合成 1 对带死区的互补 PWM; 而端口 PWM1A 和 PWM1D 则可输出互补 PWM 的反向波形, 从而组合成 2 对互反的带死区互补 PWM。

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	PWM1AOE	PWM1AEN	T1CKS1	T1CKS0	T1PRS2	T1PRS1	T1PRS0
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] T1EN – 定时器 T1 使能位

- 0: 关闭定时器 T1;
- 1: 开启定时器 T1;

BIT[6] PWM1AOE – PWM1A 端口输出使能位

- 0: 禁止端口输出脉宽调制波形;
- 1: 允许端口输出脉宽调制波形;

BIT[5] PWM1AEN – PWM1A 使能位

- 0: 关闭 PWM1A 功能;
- 1: 使能 PWM1A 功能;



BIT[4:3] T1CKS[1:0] – T1 时钟源选择位

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

BIT[2:0] T1PRS[2:0] – T1 时钟预分频比选择位

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

定时器 T1 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CNT	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，为可读写的递减计数器

定时器 T1 重载寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1LOAD	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，用于设置 T1 的计数周期

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。



定时器 T1 比较寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1DATA	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 A，用于设置 PWM1A 的占空比

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1DATB	T1DATB7	T1DATB6	T1DATB5	T1DATB4	T1DATB3	T1DATB2	T1DATB1	T1DATB0
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] T1DATB[7:0] – T1 比较寄存器 B，用于设置 PWM1B 的占空比

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1DATC	T1DATC7	T1DATC6	T1DATC5	T1DATC4	T1DATC3	T1DATC2	T1DATC1	T1DATC0
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] T1DATC[7:0] – T1 比较寄存器 C，用于设置 PWM1C 的占空比

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1DATD	T1DATD7	T1DATD6	T1DATD5	T1DATD4	T1DATD3	T1DATD2	T1DATD1	T1DATD0
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] T1DATD[7:0] – T1 比较寄存器 D，用于设置 PWM1D 的占空比

PWM1 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CR0	-	-	PWM1DOE	PWM1DEN	PWM1COE	PWM1CEN	PWM1BOE	PWM1BEN
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] PWM1DOE – PWM1D 端口输出使能位

- 0: 禁止端口输出脉宽调制波形;
- 1: 允许端口输出脉宽调制波形;



- BIT[4] PWM1DEN – PWM1D 使能位
0: 关闭 PWM1D 功能;
1: 使能 PWM1D 功能;
- BIT[3] PWM1COE – PWM1C 端口输出使能位
0: 禁止端口输出脉宽调制波形;
1: 允许端口输出脉宽调制波形;
- BIT[2] PWM1CEN – PWM1C 使能位
0: 关闭 PWM1C 功能;
1: 使能 PWM1C 功能;
- BIT[1] PWM1BOE – PWM1B 端口输出使能位
0: 禁止端口输出脉宽调制波形;
1: 允许端口输出脉宽调制波形;
- BIT[0] PWM1BEN – PWM1B 使能位
0: 关闭 PWM1B 功能;
1: 使能 PWM1B 功能;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CR1	-	-	-	-	PWM1DOS	PWM1ANV	PWM1MD	PWM1LG
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	1	0	0	0

- BIT[3] PWM1DOS – PWM1D 端口输出信号选择位
0: 输出 T1PWMD 信号;
1: 输出 PWM1C 端口信号的反向波形;
- BIT[2] PWM1ANV – PWM1A 端口输出取反控制位
0: 端口输出正向波形;
1: 端口对电平取反后输出;
- BIT[1] PWM1MD – PWM1 输出模式选择位



- 0: P11 (PWM1B) 输出 T1PWMB 信号, P10 (PWM1C) 输出 T1PVMC 信号;
1: P11 输出 T1PWMA 信号, P10 输出 T1PWMB 和 T1PVMC 的组合逻辑信号;

BIT[0] PWM1LG – 组合逻辑控制位

- 0: 组合逻辑为 T1PWMB 与 T1PVMC 同或;
1: 组合逻辑为 T1PWMB 与 T1PVMC 异或;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CR2	-	-	-	-	PWM1DDB	PWM1CDB	PWM1BDB	PWM1ADB
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	1	1	1	1

BIT[3] PWM1DDB – T1PWMD 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] PWM1CDB – T1PVMC 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;

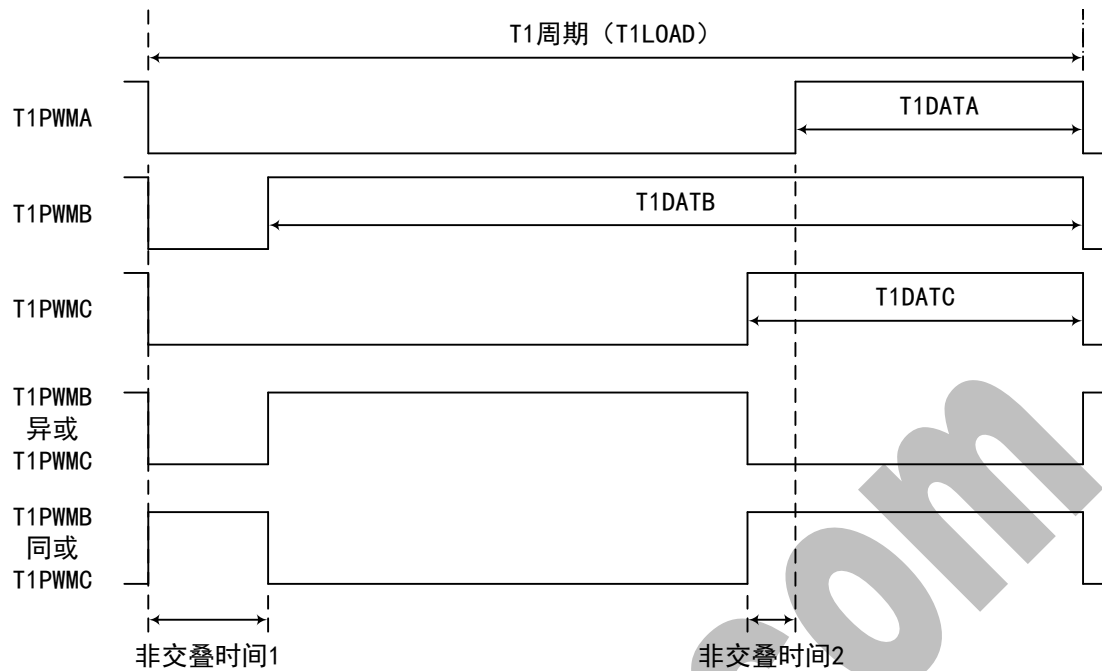
BIT[1] PWM1BDB – T1PWMB 匹配翻转时钟沿选择位

- 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;

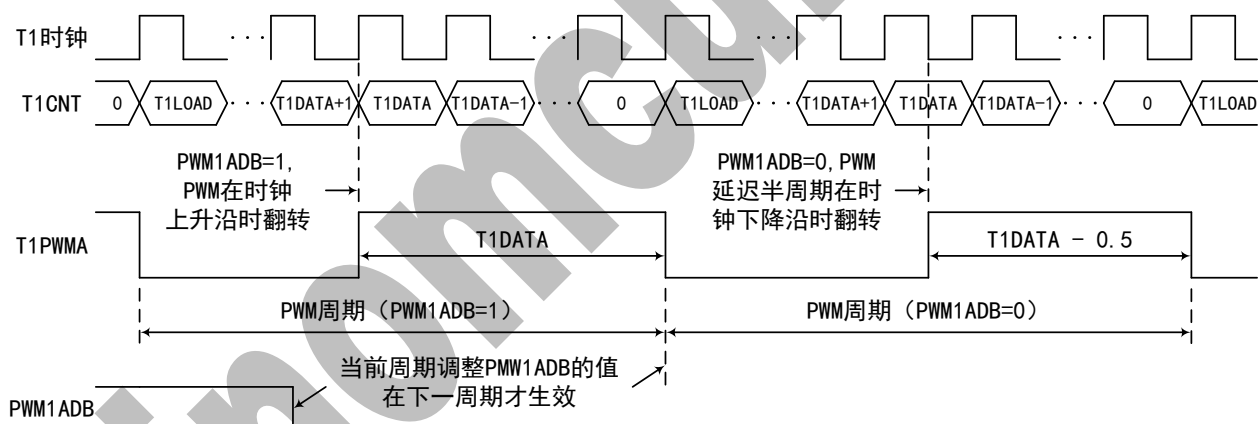
BIT[0] PWM1ADB – T1PWMA 匹配翻转时钟沿选择位

- 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;

PWM1 互补波形示意



PWM1 半个时钟微调波形示意



注:

- 1、PWM1x 的高电平时间需超过 1 个计数时钟周期 (即 $T1DATx$ 需 >1) 时, 半周期微调功能才会有效;
- 2、因 F_{CPU} 的占空比不为 50%, 所以当 PWM 计数时钟为 F_{CPU} 的 1 分频时, 微调功能的翻转延迟并不正好为半个时钟周期, 而是延迟为 F_{CPU} 的高电平时间;



8 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；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。

注：开启 LVD 或切换电压检测阈值等操作，需待电路稳定（时间>200μs）后 LVD 输出才有效。

杂项控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCR	WDTEN	EIS	LVDF	LVDVS3	LVDVS2	LVDVS1	LVDVS0	LVDEN
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
初始值	1	0	0	0	0	0	0	0

BIT[7] WDTEN – 看门狗使能位

BIT[6] EIS – INT 端口外部中断功能使能位

BIT[5] 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；

BIT[4:1] 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	3.0V
0001	2.0V	1001	3.1V
0010	2.1V	1010	3.2V
0011	2.2V	1011	3.3V
0100	2.4V	1100	3.6V
0101	2.6V	1101	3.8V
0110	2.8V	1110	4.0V
0111	2.9V	1111	4.2V

BIT[0] LVDEN – 低电压检测 LVD 使能位

0：关闭 LVD；



1: 开启 LVD;

Sinomcu.com



9 EEPROM 存储器

9.1 EEPROM 概述

芯片内置 64 字节的 EEPROM 型数据存储器，支持用户程序在带电运行中实时地读写数据。对 EEPROM 中数据的读写操作需通过控制寄存器 EECR、保护寄存器 EEPR、地址寄存器 EEAR 和数据寄存器 EEDR 进行。

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

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

注：

- 1、EEPROM 读/写操作时，CPU 将暂停工作；
- 2、仅支持高速模式下进行 EEPROM 操作；
- 3、对 EECR、EEPR、EEAR 和 EEDR 的写操作，仅能通过 MOVRA 指令进行，其他指令的执行结果不确定；

9.2 EEPROM 相关寄存器

EEPROM 控制寄存器

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

BIT[1] EEREAD – EEPROM 读操作控制位

0：操作未开始或已完成，写 1 开始 EEPROM 读操作；

1：EEPROM 读操作中，完成后自动清 0；



BIT[0] EEWRITE – EEPROM 写操作控制位

0: 操作未开始或已完成, 写 1 开始 EEPROM 写操作;

1: EEPROM 写操作中, 完成后自动清 0;

注: 应用程序中, 在写 EECR 指令和下一条真正执行的指令之间, 需增加至少 2 个 NOP 指令, 以防止时序错误。

EEPROM 保护寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EEPR	EEP7	EEP6	EEP5	EEP4	EEP3	EEP2	EEP1	EEP0
R/W	W	W	W	W	W	W	W	W
初始值	0	0	0	0	0	0	0	0

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

EEPROM 地址寄存器

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

EEPROM 数据寄存器

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

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

9.3 EEPROM 操作示例

例如, 先将数据 55H 写入 EEPROM 存储器 10H 中, 再读取 EEPROM 存储器 11H 地址中内容:

```
MOVAI    10H
```



MOVRA EEAR ; 将 10H 写入 EEAR
MOVAI 55H
MOVRA EEDR ; 将 55H 写入 EEDR
BCLR GIE ; 屏蔽中断
MOVAI 5AH
MOVRA EEPR ; 使能 EE 操作, 第 1 步: EEPR 写 5AH
MOVAI A5H
MOVRA EEPR ; 使能 EE 操作, 第 2 步: EEPR 写 A5H
MOVAI 01H
MOVRA EECR ; 启动 EE 写操作, 将数据 55H 写入 EEPROM 地址 10H

中
令
NOP ; 为防止时序错误, CPU 必须先执行 2-4 个 NOP 指

NOP
BSET GIE ; 允许中断
JBCLR EECR,0 ; 检查 EE 写操作是否完成
GOTO \$-1
MOVAI 11H
MOVRA EEAR ; 将 11H 写入 EEAR
MOVAI 02H
MOVRA EECR ; 启动 EE 读操作, 读取 EEPROM 地址 11H 中内容
NOP ; 为防止时序错误, CPU 必须先执行 2-4 个 NOP 指

令
NOP
MOVAR EEDR ; 从 EEDR 中读取数据



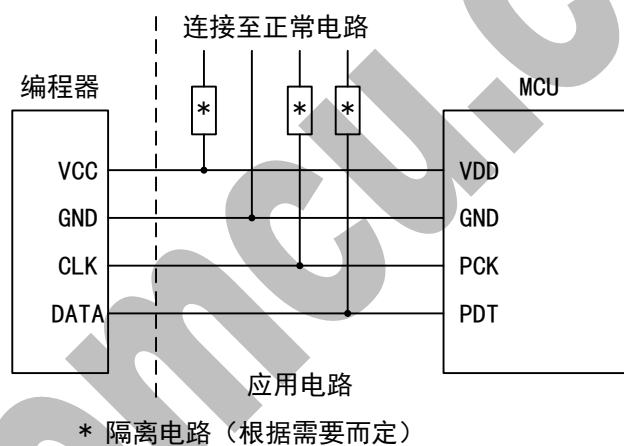
10 FLASH 烧录编程

10.1 FLASH 在板编程

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

芯片也支持对 EEPROM 型数据存储器的在板编程。

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





11 中断

芯片的中断源包括外部中断 (INT)、定时器中断 (T0~T1) 和键盘中断等。可通过中断总使能位 GIE 屏蔽所有中断。

CPU 响应中断的过程如下：

- ✧ CPU 响应中断源触发的中断请求时，自动将当前指令之后将要执行的下一条指令的地址压栈保存，自动清 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 STATUS，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 STATUS，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。

11.1 外部中断

芯片具有 1 路外部中断源 INT，可通过寄存器位 INTM 选择上升沿或下降沿等触发方式。外部中断触发时，中断标志 INTIF 将被置 1，若 GIE 为 1 且相应的外部中断使能位 INTIE 为 1，则产生外部中断。

注：若通过寄存器位 EIS 置 1 使能端口的外部中断功能，则端口的键盘中断唤醒功能无效。

11.2 定时器中断

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



11.3 键盘中断

芯片具有 14 路键盘中断源，均可单独使能或关闭端口的键盘中断功能。任意一路使能键盘中断功能的端口，其输入电平发生变化时均将触发键盘中断，中断标志 KBIF 将被置 1，若 GIE 为 1 且键盘中断使能位 KBIE 为 1，则产生键盘中断。

键盘中断控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0KBCR	-	-	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;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1KBCR	P17KE	P16KE	P15KE	P14KE	P13KE	P12KE	P11KE	P10KE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P1nKE – P1n 端口键盘中断功能使能位 (n=7-0)

0: 关闭端口的键盘中断功能;

1: 使能端口的键盘中断功能;

11.4 中断相关寄存器

中断使能寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTE	GIE	-	-	-	-	INTIE	KBIE	TOIE
R/W	R/W	-	-	-	-	R/W	R/W	R/W
初始值	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[2] INTIE – INT 中断使能位

- 0: 屏蔽 INT 中断;
- 1: 使能 INT 中断;

BIT[1] KBIE – 键盘中断使能位

- 0: 屏蔽键盘中断;
- 1: 使能键盘中断;

BIT[0] TOIE – 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;

中断标志寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTF	-	-	-	-	-	INTIF	KBIF	TOIF
R/W	-	-	-	-	-	R/W	R/W	R/W
初始值	-	-	-	-	-	0	0	0

BIT[2] INTIF – INT 中断标志位

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

BIT[1] 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[0] TOIF – 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;



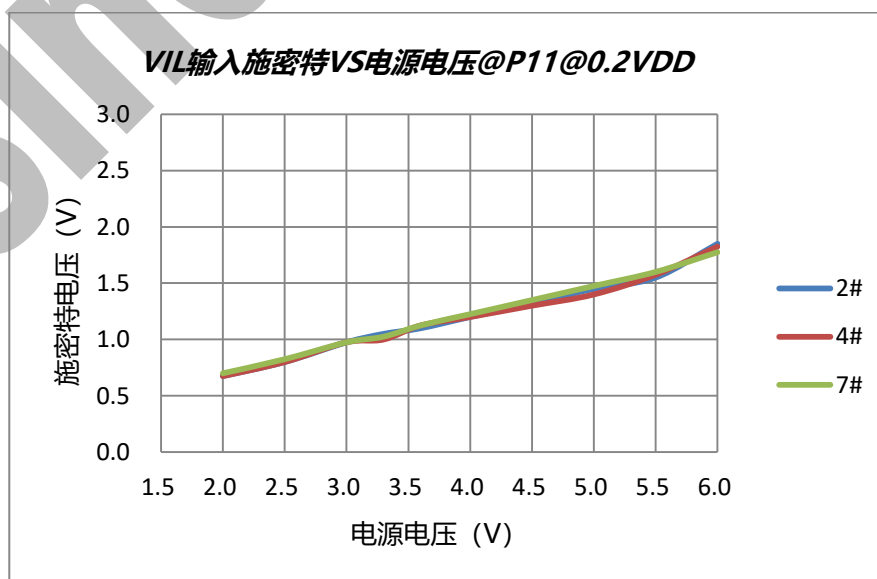
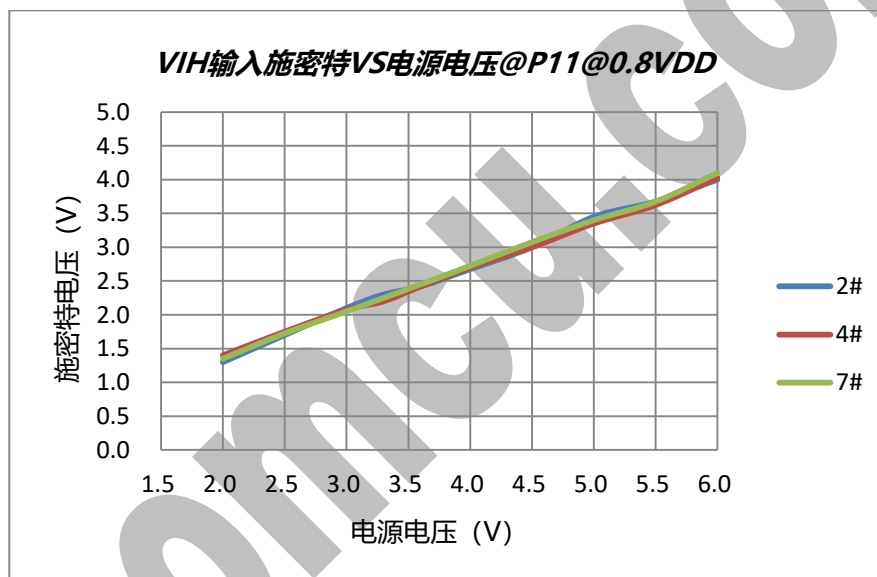
12 特性曲线

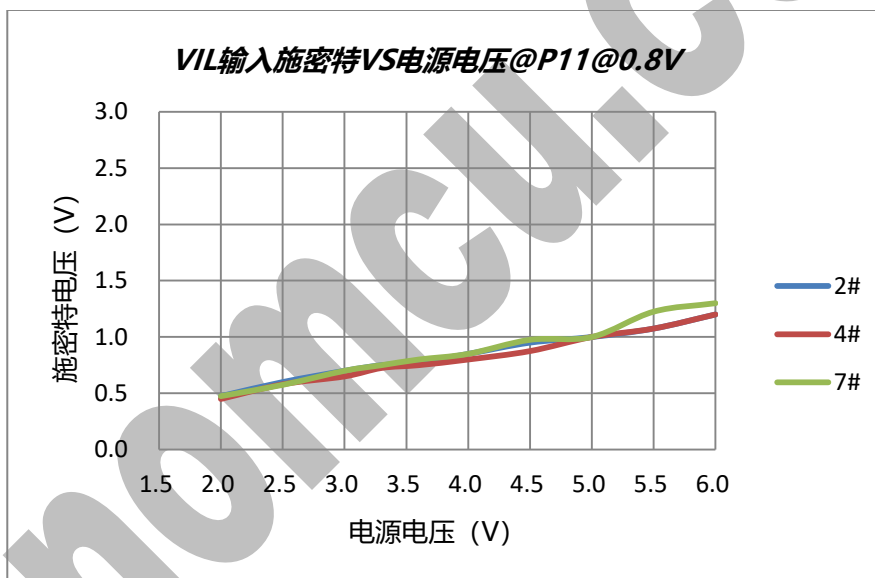
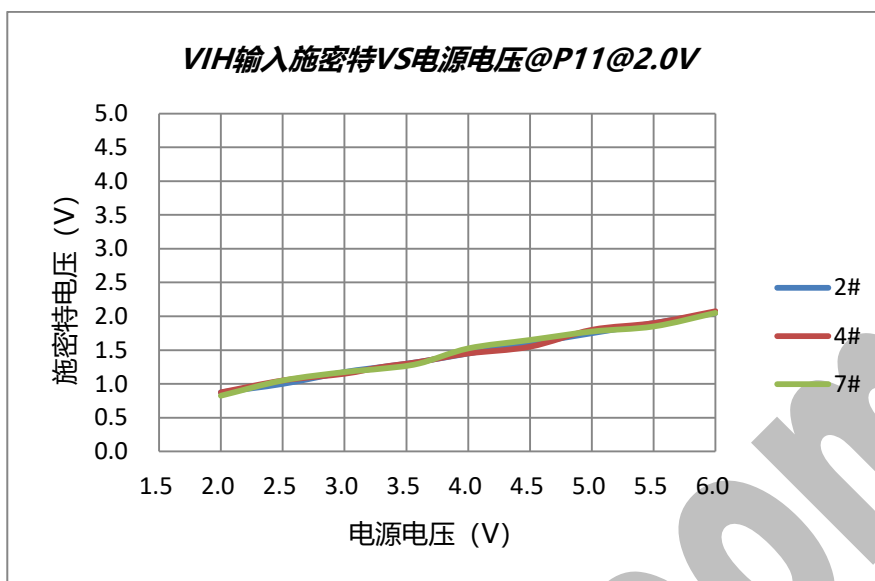
注:

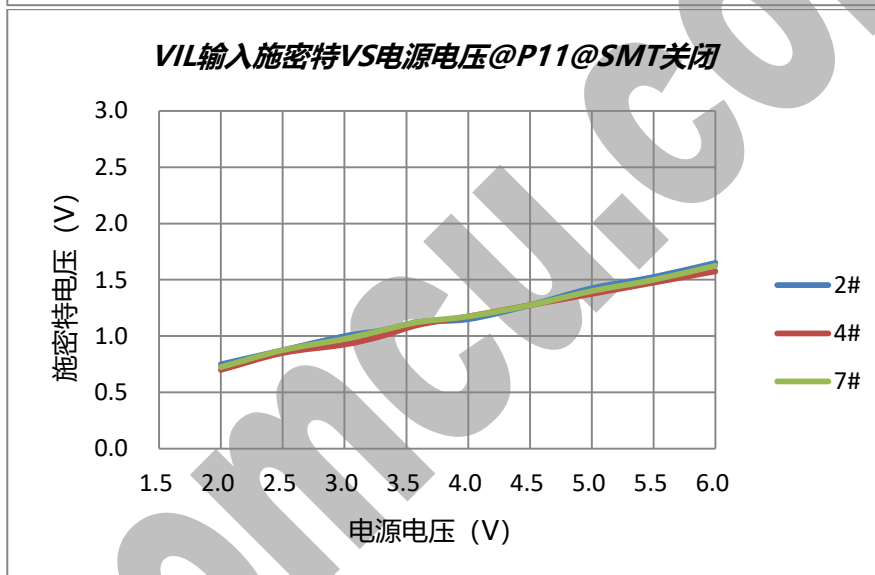
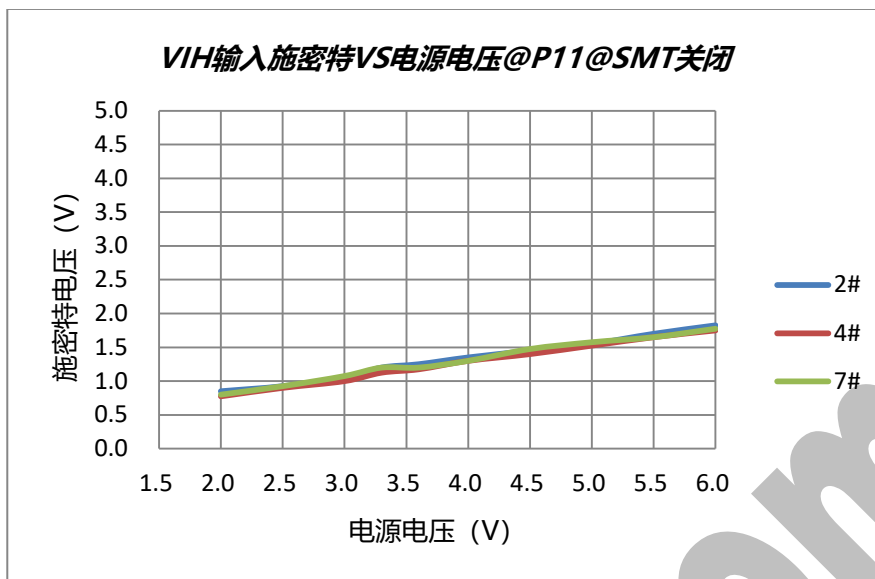
- 1、特性曲线图中数据均源自抽样实测，仅作为应用参考，部分数据因生产工艺偏差，可能与实际芯片不符；为保证芯片能正常工作，请确保其工作条件符合电气特性参数说明；
- 2、图文中若无特别说明，则电压特性曲线的温度条件为 $T=25^{\circ}\text{C}$ ，温度特性曲线的电压条件为 $V_{DD}=5\text{V}$ ；

12.1 I/O 特性

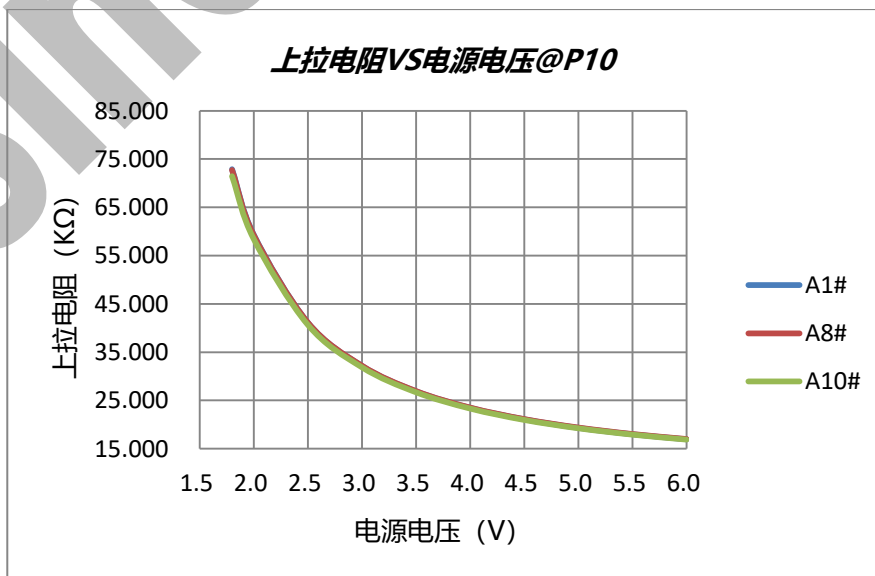
输入 SMT 阈值电压 VS 电源电压

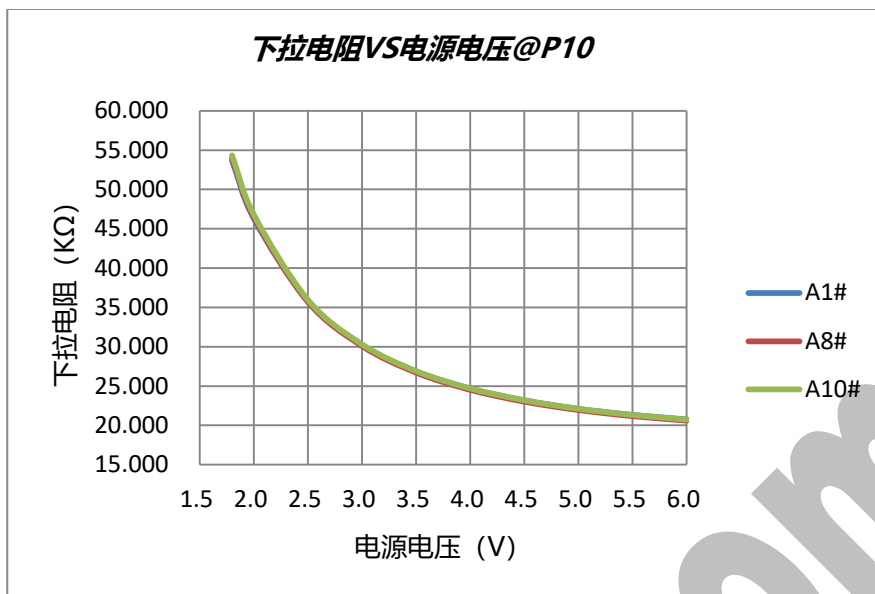




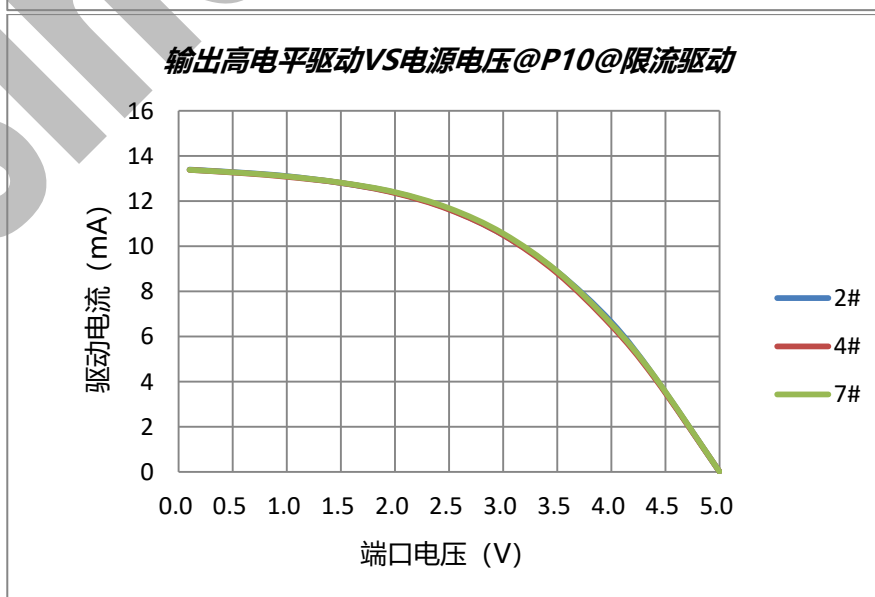
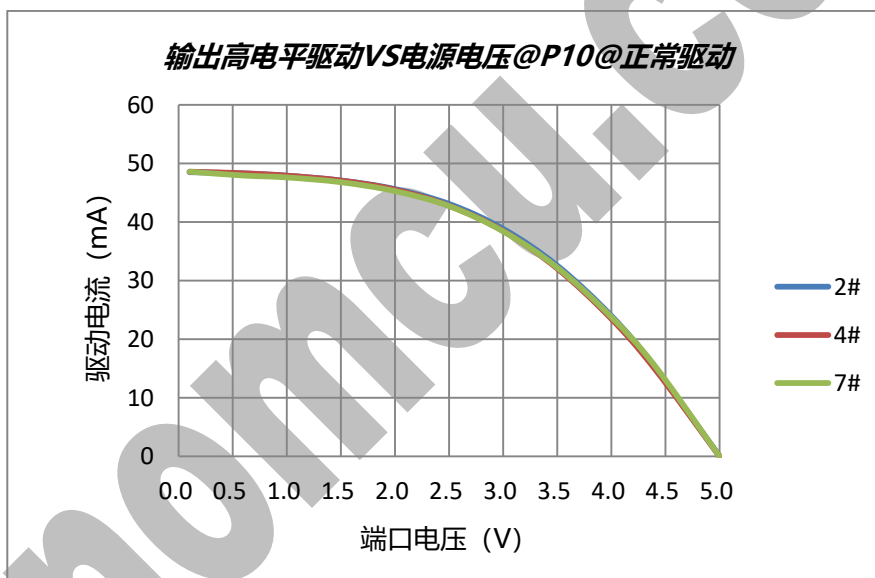


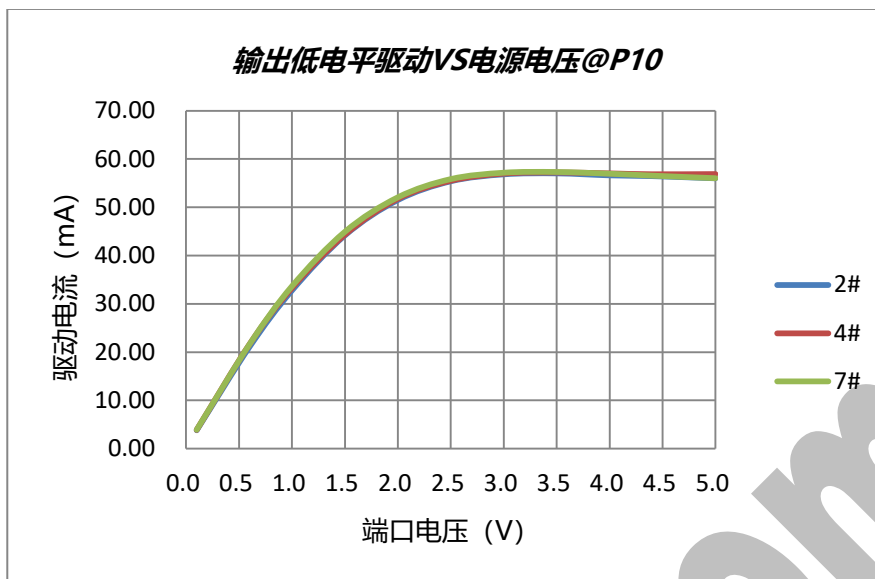
上/下拉电阻值 VS 电源电压





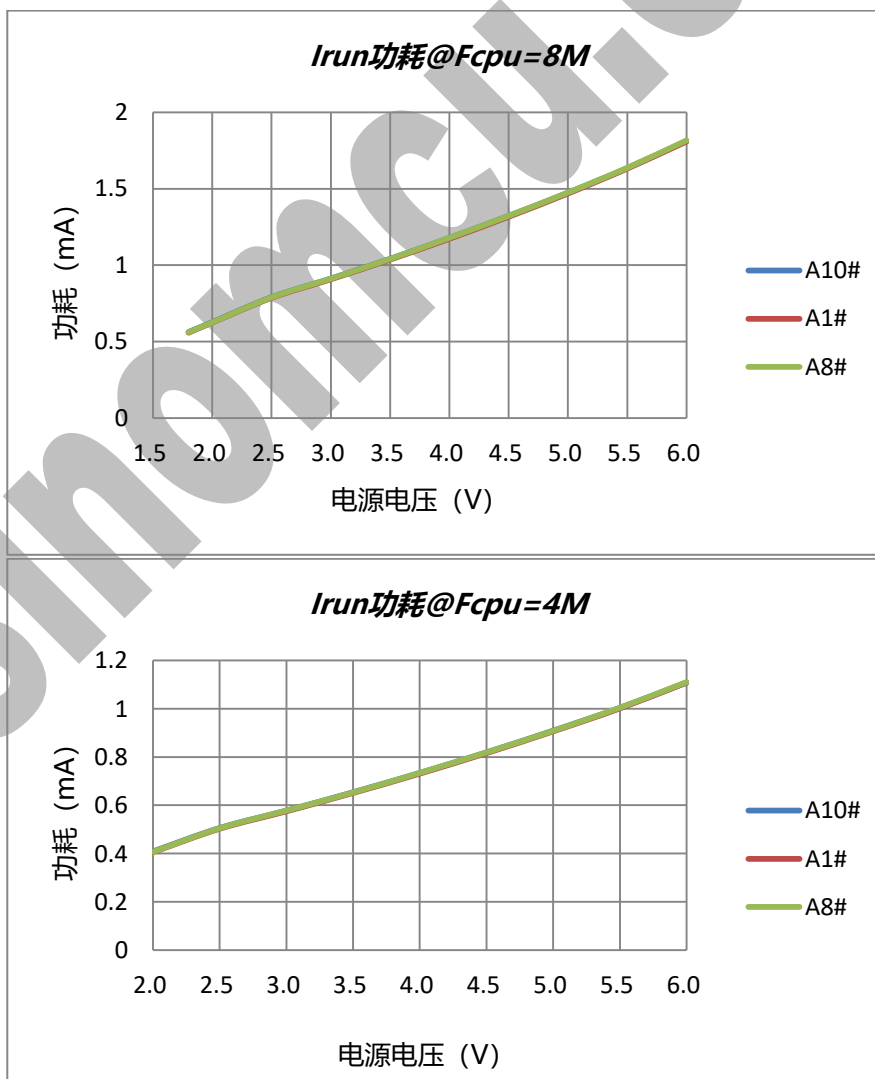
I/O 输出电流 VS 端口电压 (VDD=5V)

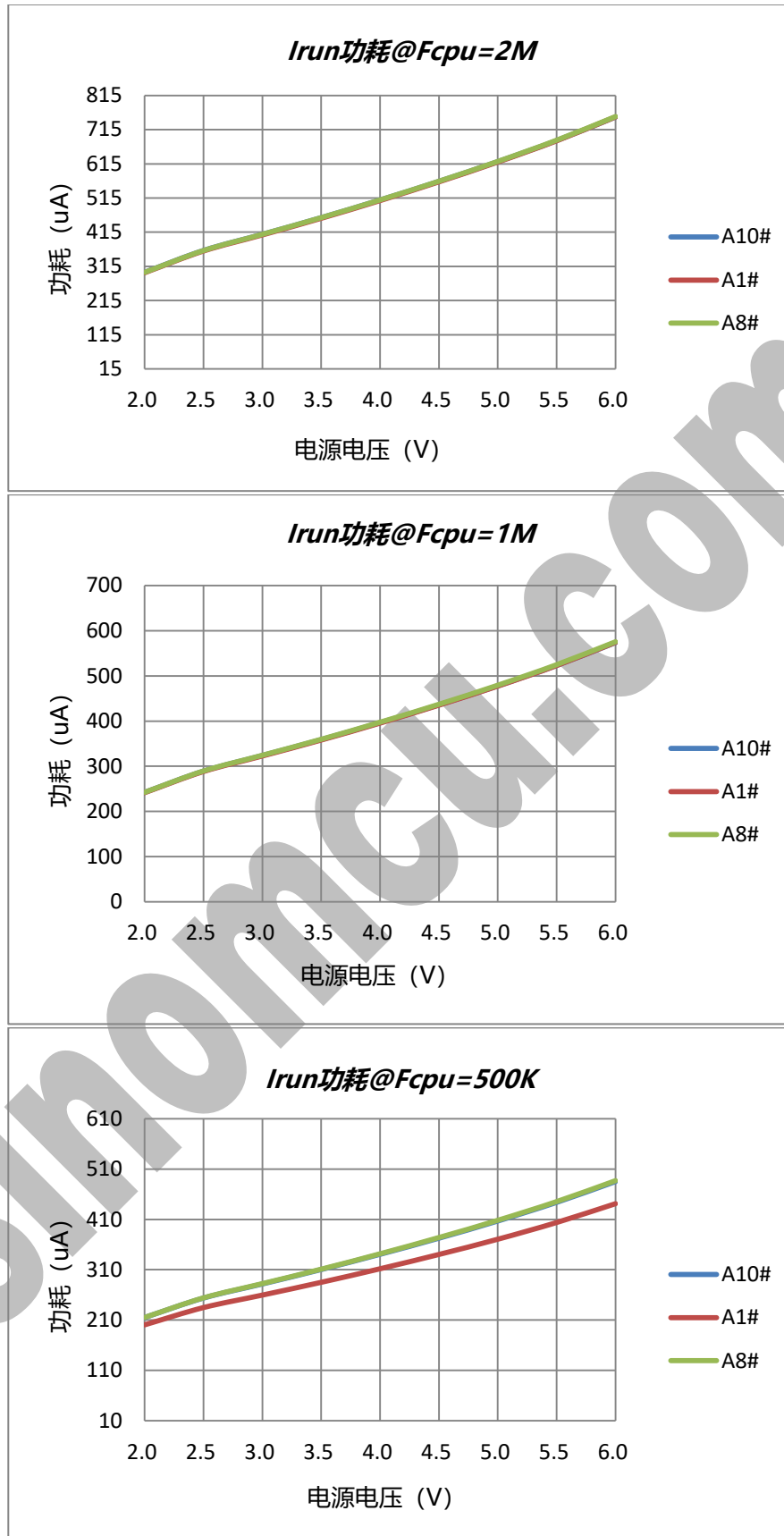


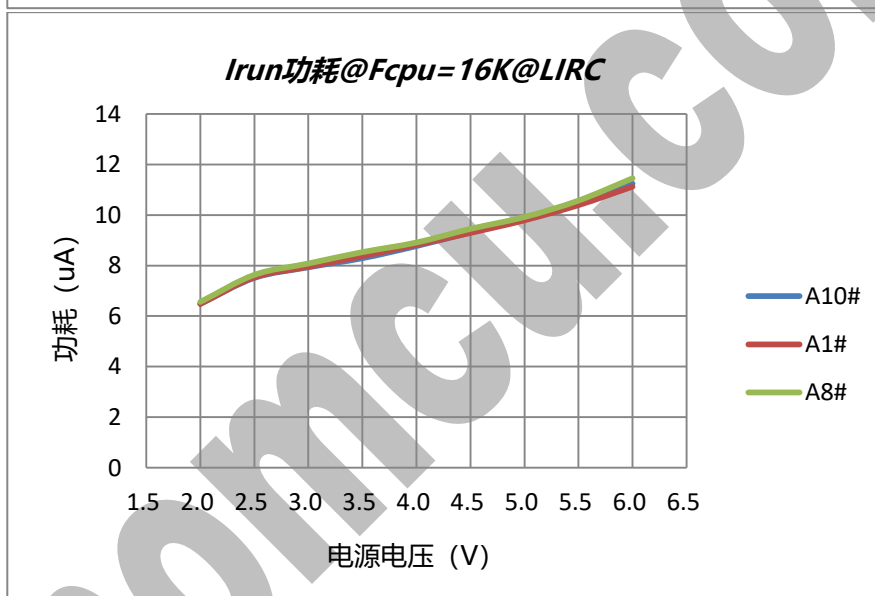
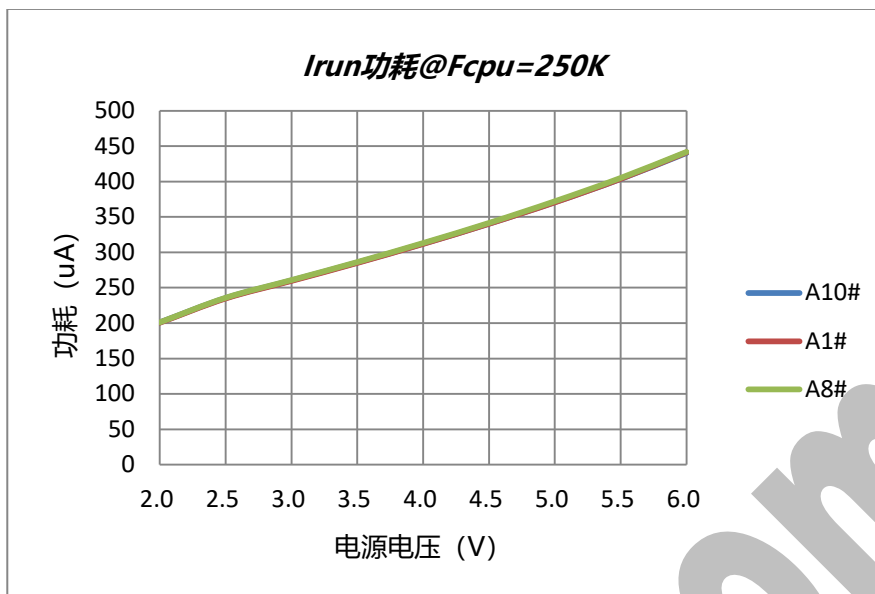


12.2 功耗特性

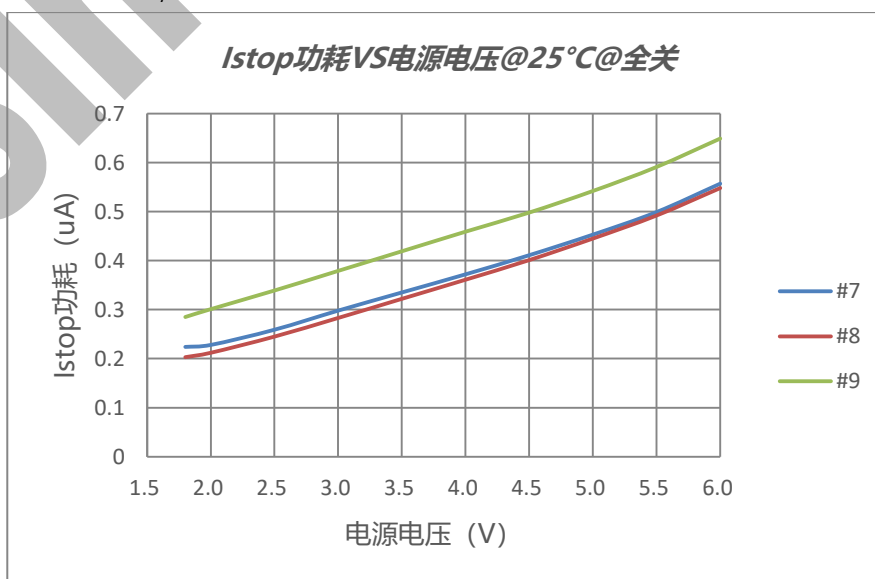
运行模式 功耗 VS 电源电压

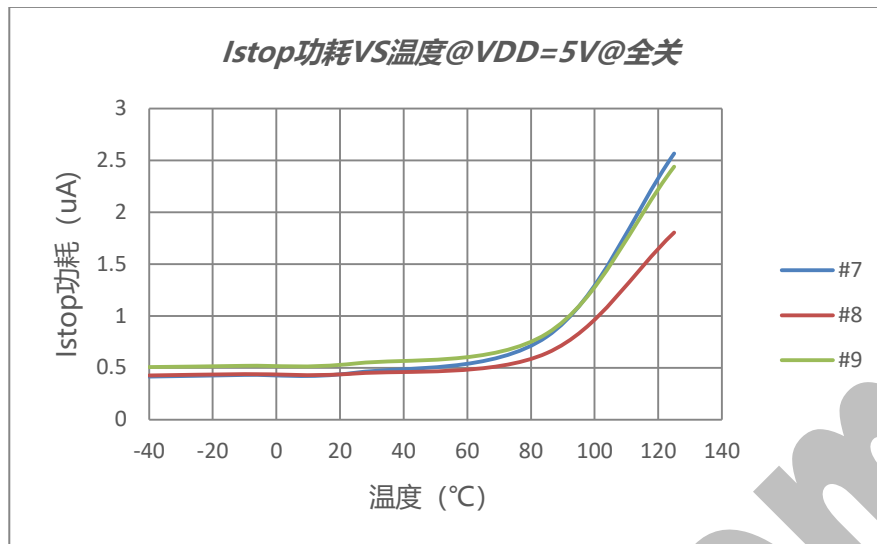






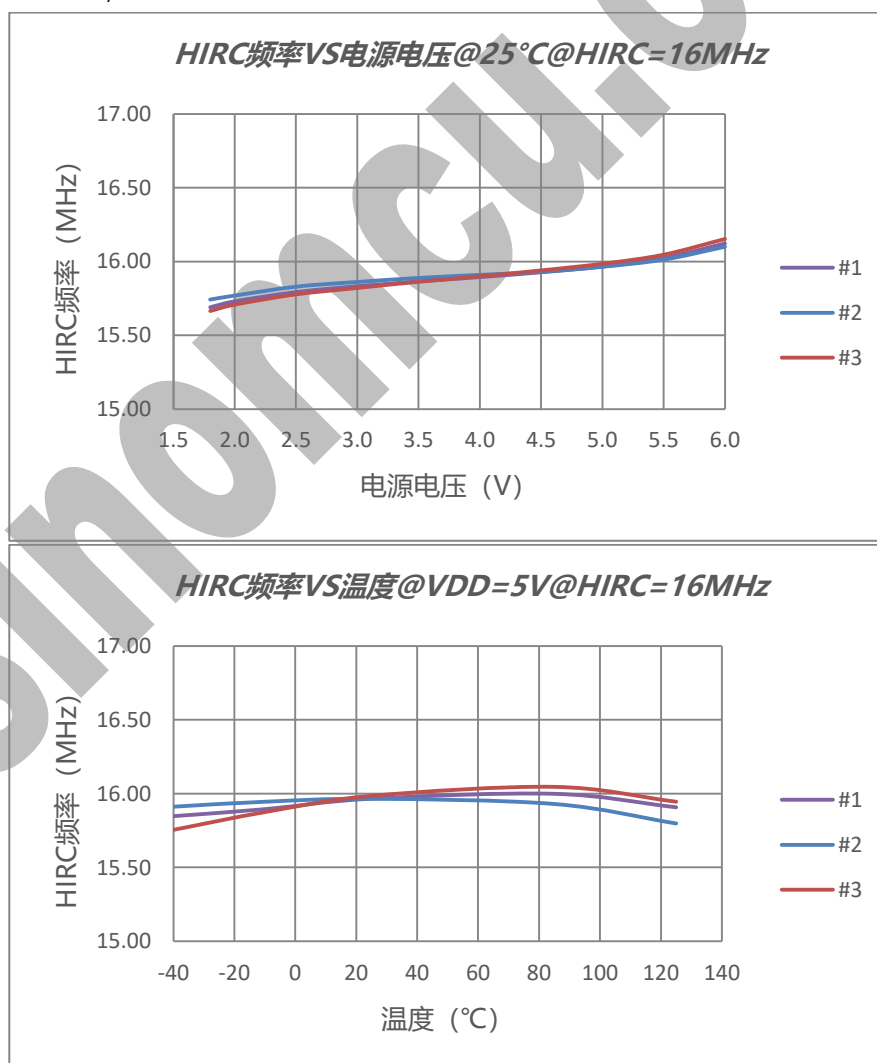
休眠模式 功耗 VS 电源电压/温度





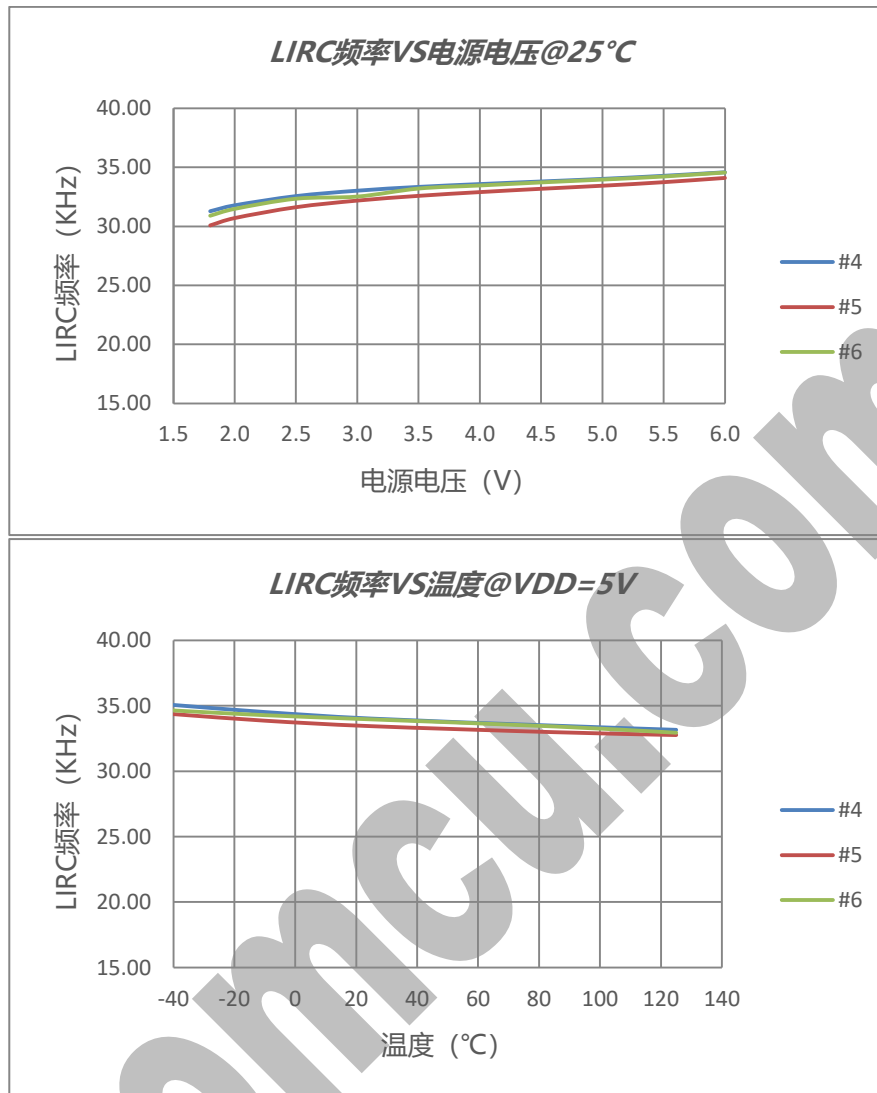
12.3 模拟电路特性

HIRC 频率 VS 电源电压/温度

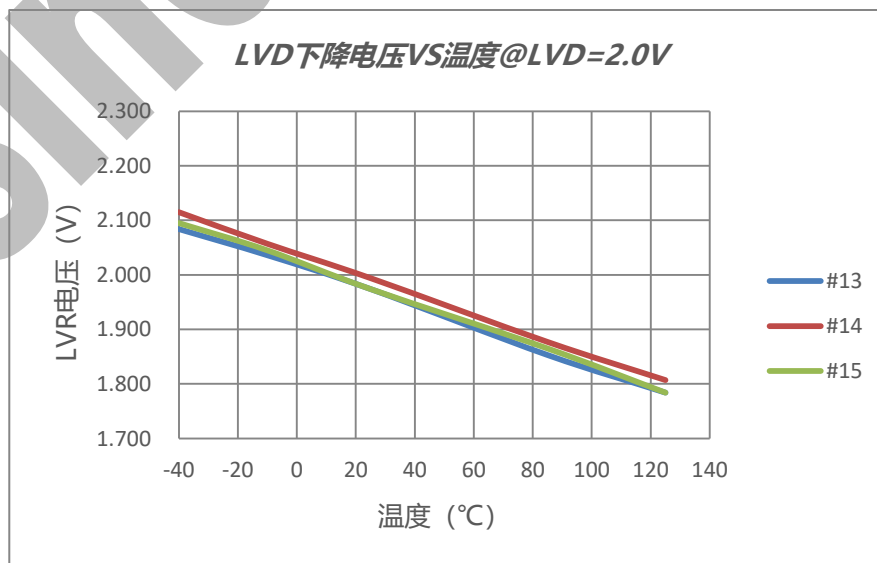


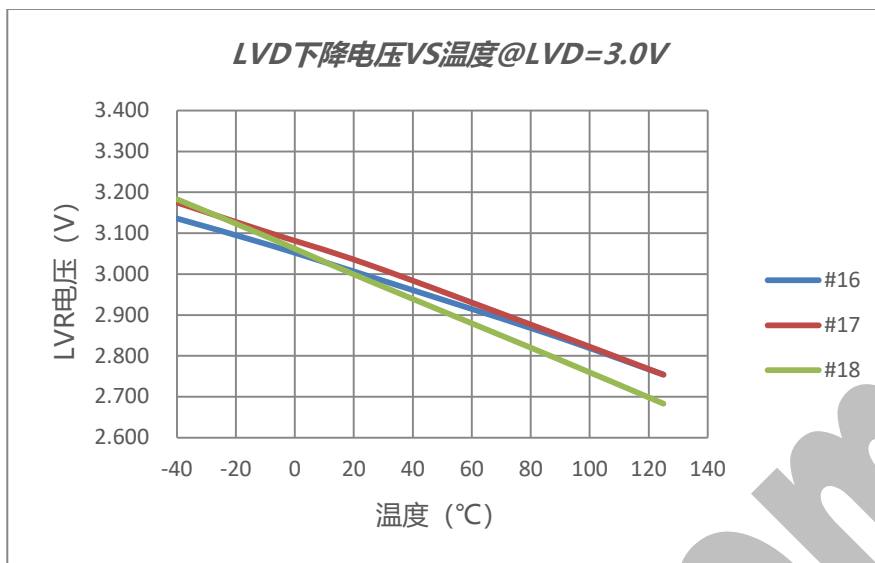


LIRC 频率 VS 电源电压/温度



LVD 阈值电压 VS 温度

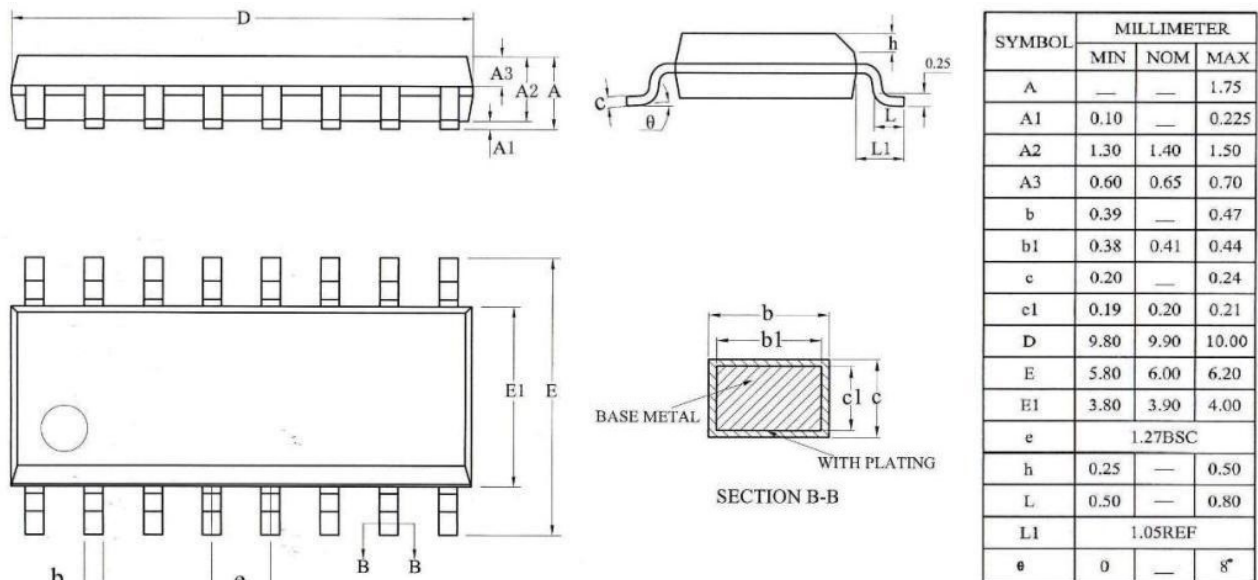




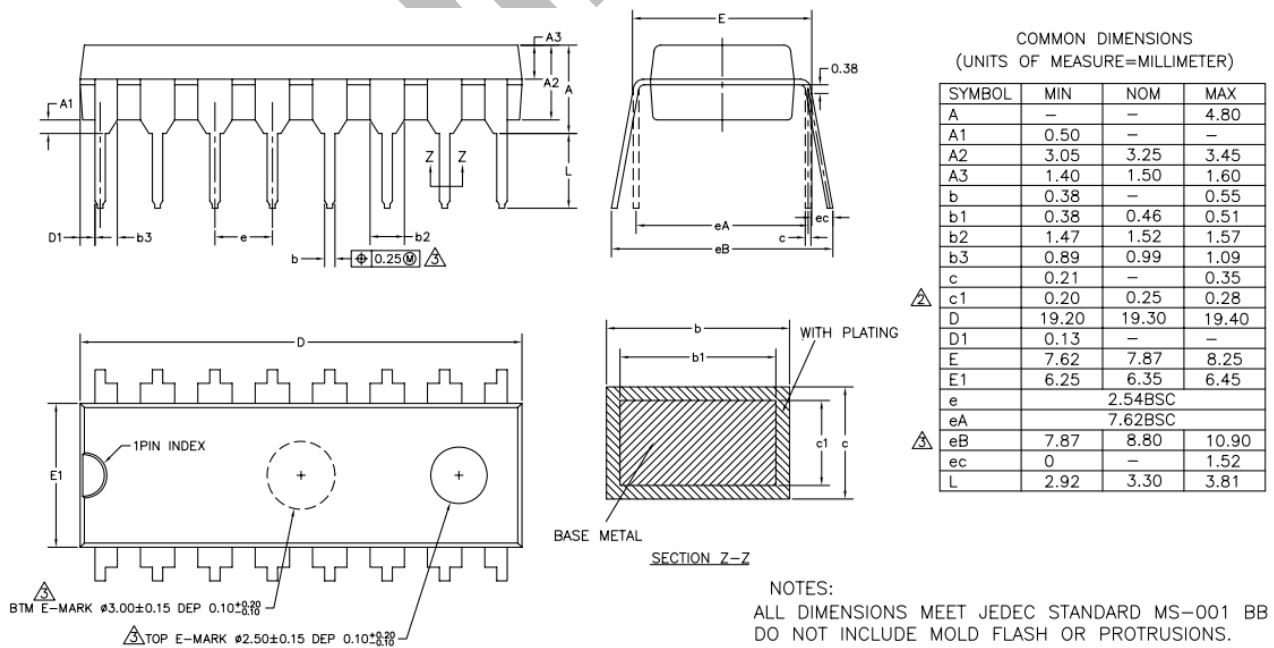


13 封装尺寸

13.1 SOP16

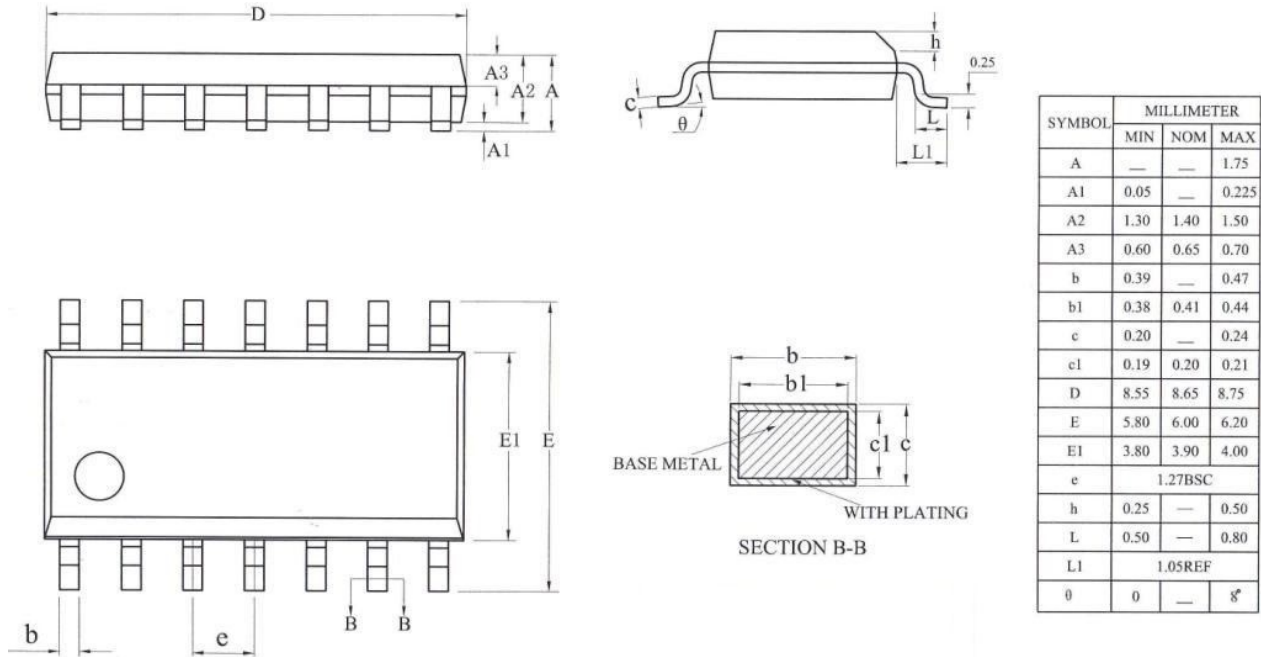


13.2 DIP16

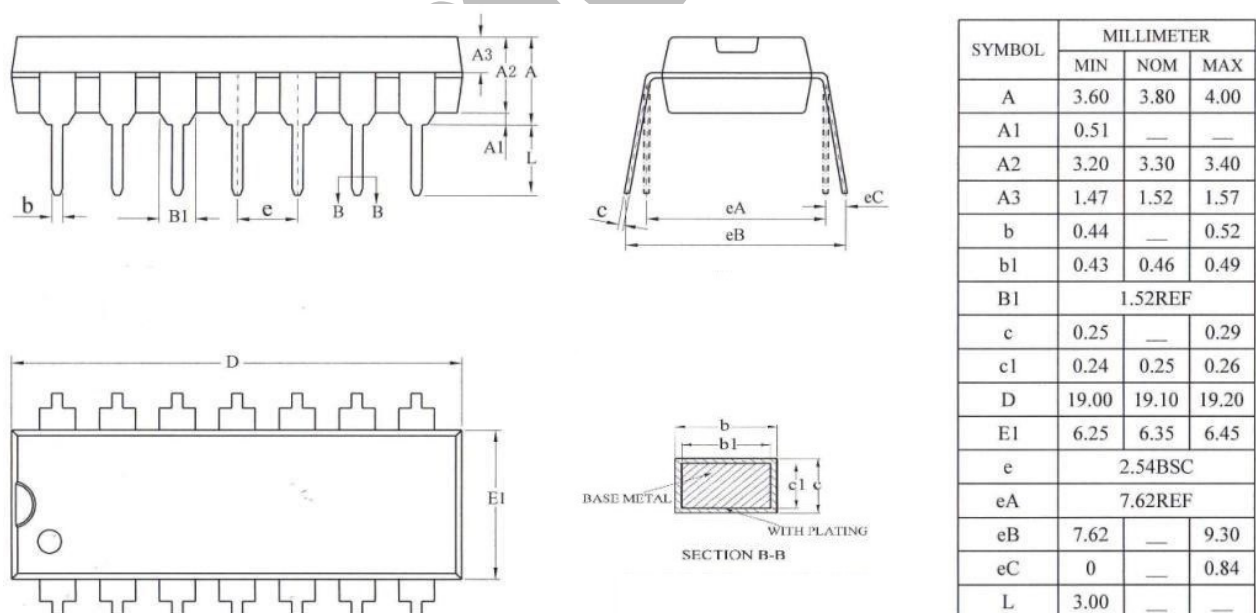




13.3 SOP14



13.4 DIP14





14 修订记录

版本	日期	修订内容
V1.0	2021-10-29	发布初版；
V1.1	2022-05-11	新增写 EECR 指令后需先加 NOP 指令的注释；
V1.2	2024-09-26	增加 DIP16 A2C
V1.3	2024-10-21	更新 A2C 引脚排列图
V1.4	2025-07-07	新增 SOP8 封装 A0H