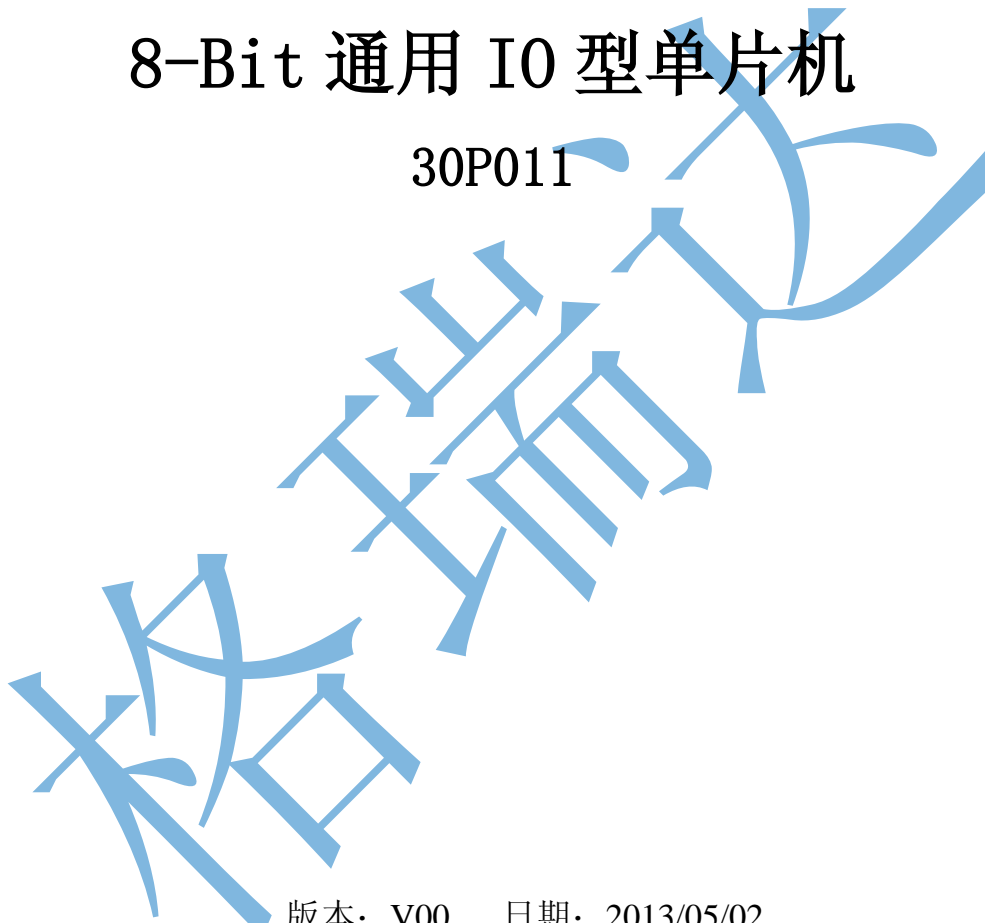




8-Bit 通用 I/O 型单片机

30P011



版本: V00 日期: 2013/05/02

www.greenmcu.com

目 录

1 产品简介	1
1.1 产品特性	1
1.2 系统框图	2
1.3 引脚排列	3
1.4 引脚说明	3
1.5 引脚结构示意图	4
2 中央处理器	7
2.1 指令集	7
2.2 程序存储器 ROM	7
2.3 用户数据存储器 RAM	7
2.4 堆栈	8
2.5 用户配置字 OPBIT	8
2.6 控制寄存器	9
3 系统时钟	11
3.1 外接晶振	11
3.2 内置高精度 RC 振荡	12
3.3 WDT 振荡器	12
4 复位	12
4.1 概述	12
4.2 上电复位	13
4.3 外部复位	13
4.4 LVR 复位电压检测	13
4.5 WDT 复位	13
5 I/O 口	14
5.1 IO 工作模式	14
5.2 上拉，下拉电阻和开漏	15
6 定时器	16
6.1 定时器 T0	16
6.2 定时器 T1	20
6.3 WDT 定时器	24
7 中断	24
7.1 概述	24
7.2 外中断	25
7.3 键盘中断	25
7.4 定时器中断	26

8 系统低功耗 SLEEP 模式.....	26
9 指令集概述.....	26
9.1 指令汇总表	28
9.2 指令详细说明	30
ADCAR	30
ADCRA	30
ADDAI	30
ADDAR	31
ADDRA	31
ANDAI	31
ANDAR	31
ANDRA	32
BCLR.....	32
BSET	32
CALL.....	32
CLRA.....	33
CLRR.....	33
CLRWDT.....	33
COMAR.....	33
COMR	34
DAA.....	34
DECAR.....	34
DECR.....	34
DJZAR.....	35
DJZR.....	35
DSA	35
GOTO	35
INCAR.....	36
INCR.....	36
ISUBAI.....	36
JBCLR.....	36
JBSET	37
JZAR.....	37
JZR.....	37
MOVAI	37
MOVAR.....	38
MOVR	38
MOVRA	38
NOP	38
ORAI	39
ORAR.....	39
ORRA	39
RETAI.....	39
RETIE.....	40

RETURN	40
RLAR.....	40
RLR	40
RRAR	41
RRR	41
RSBCAR	41
RSBCRA	41
RSUBAR	42
RSUBRA	42
STOP	42
SWAPAR	43
SWAPR.....	43
XORAI.....	43
XORAR	43
XORRA	44
10 电气参数.....	44
10.1 极限参数	44
10.2 直流电气参数	44
10.3 交流电气参数	45
11 封装外形尺寸.....	45
12 附录.....	48
12.1 内置 RC 频率曲线	48
12.2 IO 口驱动能力曲线	50
12.3 STOP 模式工作电流曲线	50
12.4 动态工作电流曲线	51
13 订购信息.....	52

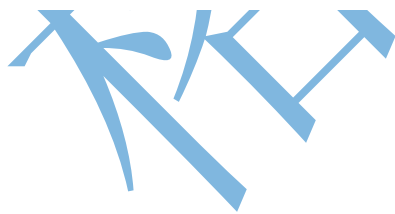
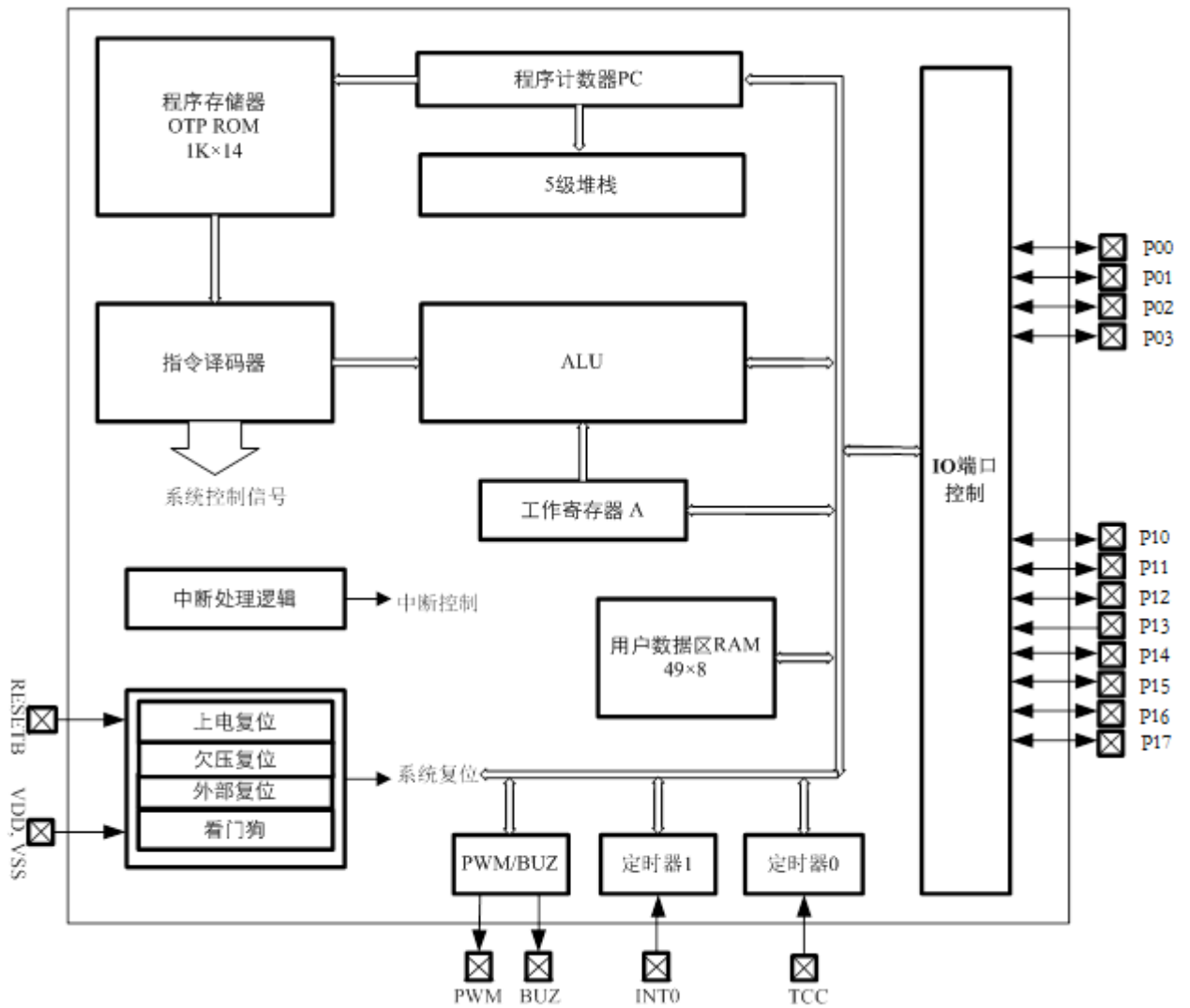
1 产品简介

30P011 是一款高性能 8 位 OTP 型 MCU，内置高精度 RC 振荡器。产品的高抗干扰性能为小家电产品提供良好的解决方案。

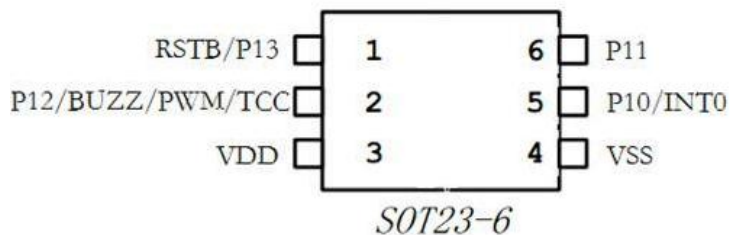
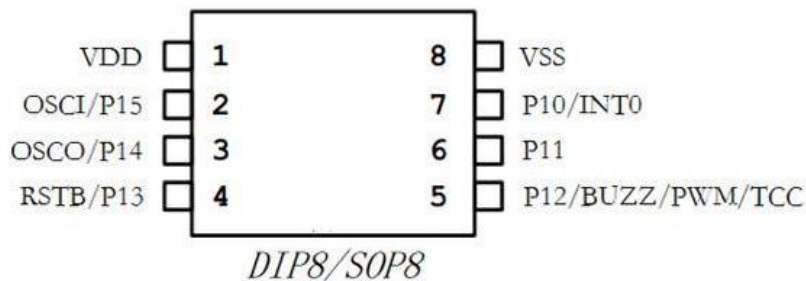
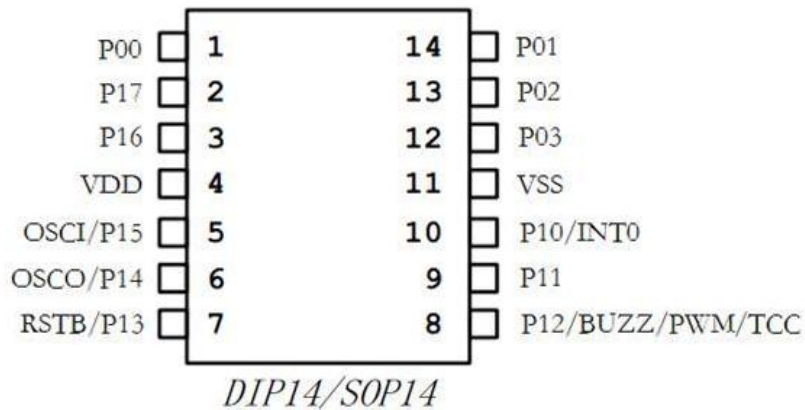
1.1 产品特性

- ◇ 8 位 RISC CPU 内核，2T
- ◇ 存储器
 - ✓ 1K*14 位程序存储器空间,5 级深度硬件堆栈
 - ✓ 49 字节通用数据寄存器空间
- ◇ 2 组 IO 口（最多可支持 11 个通用 IO 口和 1 个输入口）
- ◇ 低功耗模式，STOP 模式功耗小于 1uA，RTC 模式功耗小于 10uA
- ◇ 定时器
 - ✓ 内部自振式看门狗计数器（WDT）
 - ✓ 1 个带有 RTC 功能 8 位定时器，可设置溢出中断
 - ✓ 1 个带有 PWM/BUZ、自动加载功能 8 位定时器，可设置溢出中断
- ◇ 中断
 - ✓ 定时器 0 中断
 - ✓ 定时器 1 中断
 - ✓ 1 个外部中断口（可软件设置为上升沿、下降沿触发）
 - ✓ 8 路键盘中断功能
- ◇ 振荡器
 - ✓ 内部 28K RC 振荡器，用于看门狗计数
 - ✓ 内部 8M RC 振荡器，用于系统时钟
 - ✓ 外部 32768 低频晶体振荡器，用于系统时钟或 RTC
 - ✓ 外部高频晶体振荡器 432KHz-8MHz
- ◇ 低电压复位 LVR（多级复位电压可选）
- ◇ 工作电压
 - ✓ 2.0-5.5V@（振荡频率 432K-4MHz）
 - ✓ 2.7-5.5V@（振荡频率 432K-8MHz）
- ◇ 封装形式：DIP14、SOP14、DIP8、SOP8、SOT23-6

1.2 系统框图



1.3 引脚排列

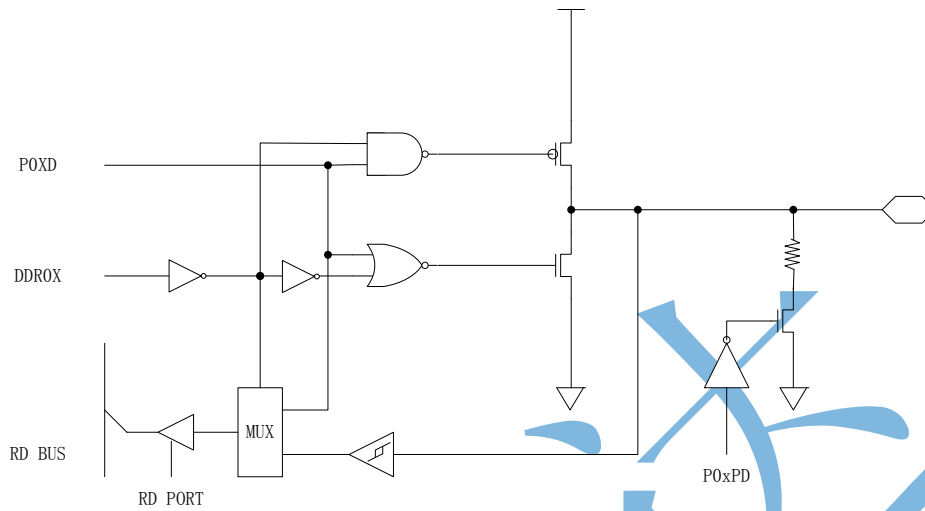


1.4 引脚说明

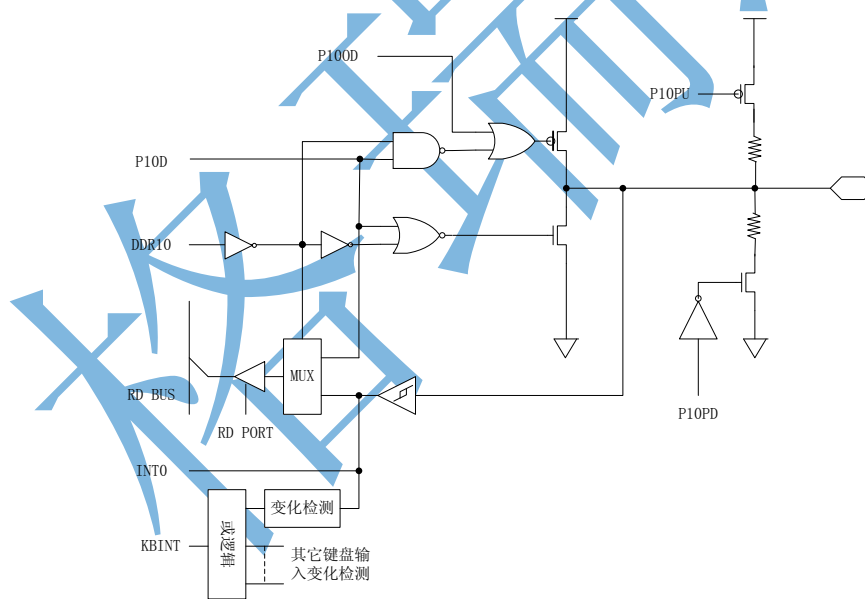
引脚名	I/O	描述
P00-P03	IO	IO, 下拉
P10/INT0	IO/I	IO, 上下拉, 开漏/外部中断输入, TIMER1 时钟输入
P11	IO	IO, 上下拉, 开漏
P12/BUZZ/PWM/TCC	IO/O/O/I	IO, 上下拉, 开漏/蜂鸣器输出/PWM 输出/TIMER0 时钟输入
P13/RESETB	I/I/P	输入端口/外部复位端口
P14/OSCO	IO/O	IO, 上拉, 开漏/振荡器输出
P15/OSCI	IO/I	IO, 上拉, 开漏/振荡器输入
P16-P17	IO	IO, 上拉, 开漏
VDD	P	VDD
GND	P	GND

1.5 引脚结构示意图

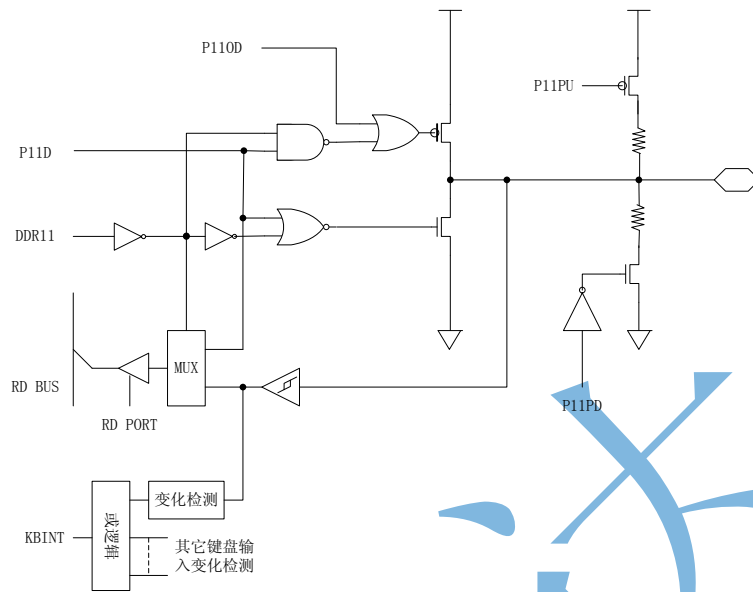
P00-P03:



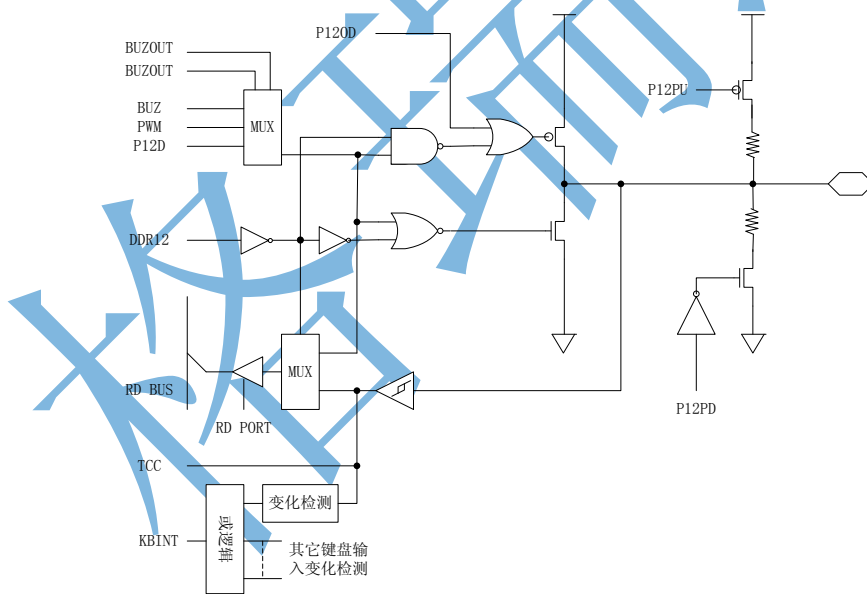
P10:



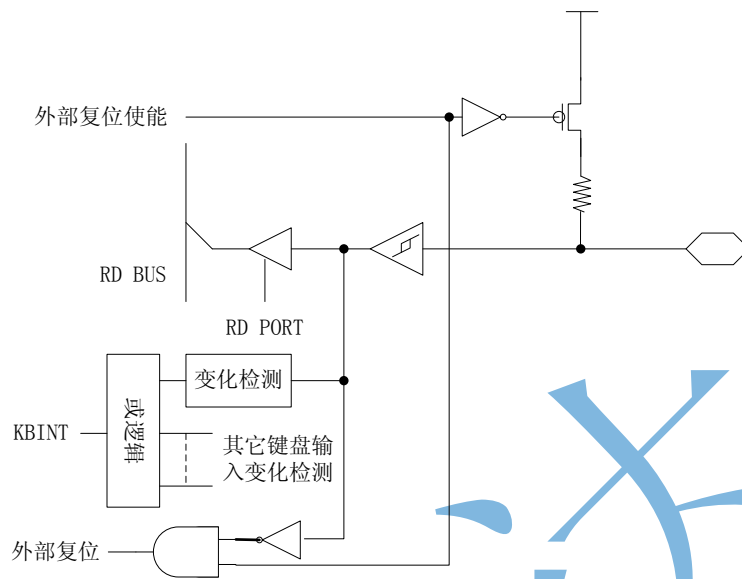
P11:



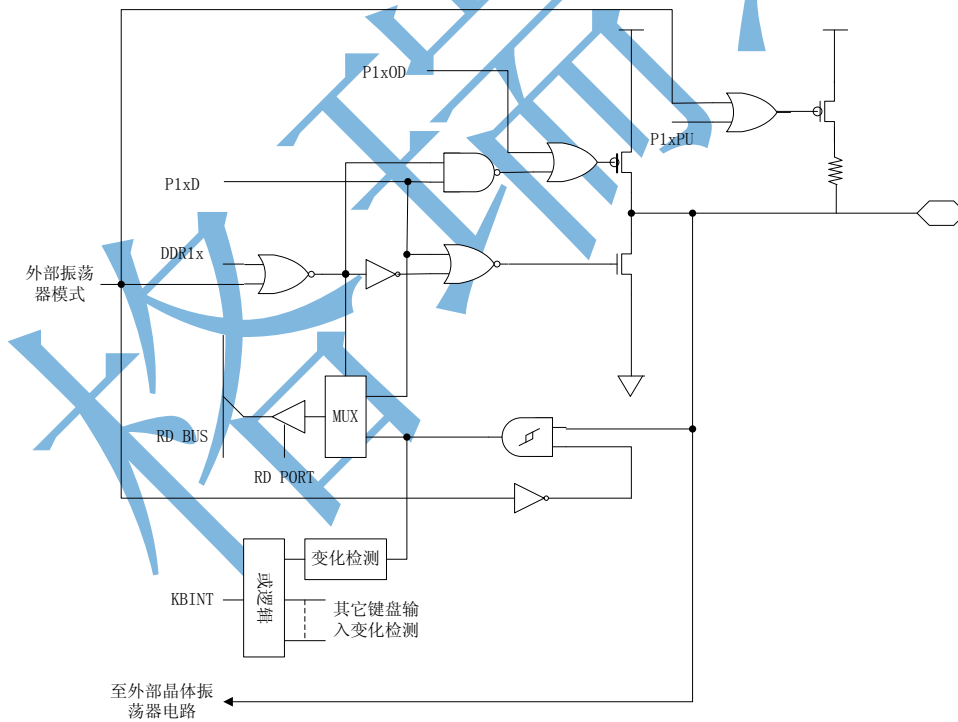
P12:



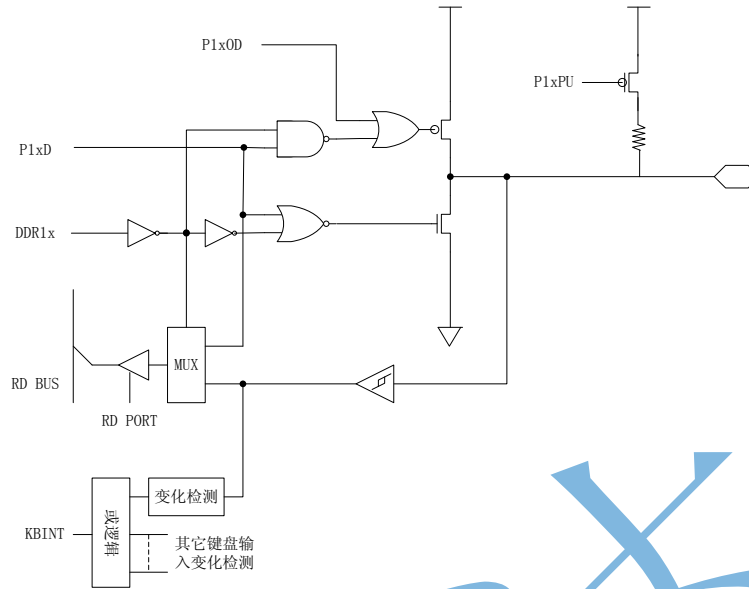
P13:



P14、P15:



P16、P17:



2 中央处理器

2.1 指令集

30P011 的指令是精简指令集。指令集详细资料见本公司手册《精简指令集》。

2.2 程序存储器 ROM

30P011 的程序存储器是 1K*14 bits 的 OTP ROM，可用于存放用户程序。地址范围：0x0000—0x03ff。

2.3 用户数据存储器 RAM

30P011 的用户数据存储器有 49 字节（8 bits）。

地址列表如下：

地址	Bank0 (可直接寻址或间接寻址)	地址	Bank1 (仅直接寻址)
@00	INDF	@40	保留
@01	T0CNT	@41	T0CR
@02	PCL	@42	保留
@03	STATUS	@43	保留
@04	FSR	@44	保留
@05	P0	@45	DDR0
@06	P1	@46	DDR1
@07	GPR (通用寄存器)	@47	保留
@08	MCR	@48	保留
@09	KBIM	@49	保留

@0A	PCLATH	@4A	保留
@0B	PDCON	@4B	TMCR
@0C	ODCON	@4C	TICR
@0D	PUCON	@4D	TICNT
@0E	INTECON	@4E	T1LOAD
@0F	INTFLAG	@4F	TIDATA
@10 ~ @3F	GPR (通用寄存器)		

直接寻址：以指令的低 7 位作为数据存储器地址，例：

MOVAI 0x55

MOVRA 0x20 ; 把数据 0x55 的内容写入 0x20 地址的数据存储器中

器中

间接寻址：当寄存器访问 INDF (0x00 地址) 时，FSR 作为数据存储器地址，例：

MOVAI 0x20

MOVRA FSR

MOVAI 0x55

MOVRA INDF ; 把数据 0x55 的内容写入 0x20 地址的数据存储器中

器中

注：间接寻址不可以寻址 0x40 以上地址空间

2.4 堆栈

30P011 有 5 级堆栈深度，当程序响应中断或执行子程序调用指令时 CPU 会将 PC 自动压栈；当运行子程序返回指令时，栈顶数据赋予 PC

2.5 用户配置字 OPBIT

用户配置字简称 OPBIT 是 OTP 中的一个特殊字节，用于对系统功能进行配置。OPBIT 在烧写用户程序时通过专用烧写器来设置。30P011 的 OPBIT 定义如下。

地址	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
OPBIT0	1	1	WDTEN	TWDT			FCPU	FINTOSC		MCLRE	VBORS			
OPBIT1	CP	OSCM		1	SMTEN	RDPORT	INTOSCCAL							

配置字	说明
VBORS	低电压复位点选择 000 : 1.5V 001: 1.8V 010 : 2.0V 011: 2.2V 100 : 2.4V 101: 2.6V 110 : 3.0V 111 : 3.6V 芯片工作电压范围：2.0V-5.5V
MCLRE	外部复位使能 1 : 使能外部复位功能 0 : 屏蔽外部复位功能，管脚复用为输入脚
FINTOSC	内部 RC 振荡器频率选择 111 : 8M 110 : 4M 101 : 2M 100 : 1M

0FH	INTFLAG	R/W	-	-	-	-	-	INTOIF	KBIF	TOIF	---- -000
41H	TOCR	R/W	-	INT0M	TOPTS	TOSE	TOPTA	T0PR2	T0PR1	T0PR0	-000 0000
45H	DDR0	R/W	-	-	-	-	DDR03	DDR02	DDR01	DDR00	---- 1111
46H	DDR1	R/W	DDR17	DDR16	DDR15	DDR14	-	DDR12	DDR11	DDR10	1111 -111
4BH	TMCR	R/W	TBS	-	-	-	-	-	T1IE	T1IF	0--- --00
4CH	T1CR	R/W	TMRIEN	PWMOUT	BUZOUT	T1PTS1	T1PTS0	T1PR2	T1PR1	T1PR0	0000 0000
4DH	T1CNT	R/W	T1C7	T1C6	T1C5	T1C4	T1C3	T1C2	T1C1	T1C0	0000 0000
4EH	T1LOAD	R/W	T1LOAD7	T1LOAD6	T1LOAD5	T1LOAD4	T1LOAD3	T1LOAD2	T1LOAD1	T1LOAD0	1111 1111
4FH	T1DATA	R/W	T1DATA7	T1DATA6	T1DATA5	T1DATA4	T1DATA3	T1DATA2	T1DATA1	T1DATA0	0000 0000

注：07H 为用户寄存器

	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_n – 间接寻址寄存器

INDF: INDF不是物理寄存器, 对INDF的寻址时间上是对FSR指向的地址进行访问, 从而实现间接寻址模式,

	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	1	1	1	1	1	1	1	1

BIT[7:0] PC_n – 程序指针计数器低 8 位

	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[7:0] PCH_n – 程序指针高 2 位缓冲器 n=1-0

程序指针计数器 (PC) 有以下几种操作模式

顺序运行指令: PC=PC+1

分支指令GOTO/CALL: PC=INST[9:0] (指令码低10位)

子程序返回指令RETIE/RETURN/RETAI:PC=TOS (堆栈栈顶)

对PCL操作指令: PC = {PCLATH[1:0],ALU[7:0]}(ALU运算结果)}

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	RST	-	-	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] RST 唤醒源标志

1: 芯片通过P1变化唤醒

0: 芯片通过其它唤醒源唤醒

- BIT[6:5]** 未用
- BIT[4]** **TO** 看门狗溢出标志
1: 上电复位, 执行CLRWDWT或STOP指令
0: 发生WDT溢出
- BIT[3]** **PD** 进入低功耗休眠模式标志
1: 上电复位, 执行CLRWDWT
0: 执行STOP指令
- BIT[2]** **Z** 零标志
1: 算术或逻辑运算的结果为零
0: 算术或逻辑运算的结果不为零
- BIT[1]** **DC** 半进位标志
1: 加法运算时低四位有进位/减法运算时没有向高四位借位
0: 加法运算时低四位没有进位/减法运算时有向高四位借位
- BIT[0]** **C** 进位标志
1: 加法运算时有进位/减法运算时没有借位发生/移位后移出逻辑1
0: 加法运算时没有进位/减法运算时有借位发生/移位后移出逻辑0

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

- BIT[7:6]** 无效
- BIT[5:0]** **FSRn** – 间接选址数据指针

3 系统时钟

由外接晶体振荡器或内置高精度 RC 电路（由 OPBIT 的 OSCS[2:0]配置）产生的振荡时钟信号 F_{osc} 经 2 分频后产生系统主时钟 F_{sys} 。

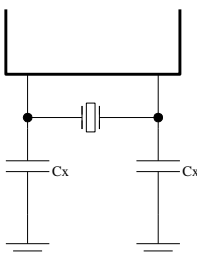
另有一个低速 RC 振荡器专供 WDT（看门狗）电路使用。

当选择内部高频振荡器&RTC 模式时，系统使能内部高频 RC 振荡器作为主时钟，外部 32768 晶体振荡器可用作定时器 0 的计数时钟源

3.1 外接晶振

外部晶体有外接高频和低频两种振荡工作模式，连接方式见下图。高频晶体可选用 432KHz~8MHz，低频一般是接 32768 时钟晶体，通常 C_x 是必须的。在实际使用中，用户应使晶体离 OSC1、OSCO 引脚的距离尽可能短，这样有助于振荡器的起振和振荡的稳定性。

下表列出几种典型频率晶振选用电容 C_x 的推荐值。



晶体频率	电容 C_x
8MHz	15p
4MHz	15p/30p
3.64MHz	15p/30p
432KHz	220p/470p

32768

10p~30p

注意：因为振荡器的品牌很多，电容值仅为推荐值，具体参数请根据实际使用的晶振性能而定。

3.2 内置高精度 RC 振荡

30P011 的内置高精度 RC 振荡器有 500KHZ、1MHZ、2MHZ、4MHZ、8MHZ 六种频率可选。当选用内置 RC 振荡时，P14、P15 引脚则可作为通用 IO 口使用。

选用内置高精度 RC 振荡器

特别提示：为确保振荡的精度和稳定性，在实际应用时需要在芯片的 VDD 和 GND 之间加 47uF 以上的电解电容，且电容和芯片的距离尽可能靠近（建议控制在 1cm 以内）。

3.3 WDT 振荡器

30P011 内置一个低频的 RC 振荡器，WDT 的基本溢出时间由 OPTION 决定，若要更长的时间可对 WDT 时钟进行分频，分频后 WDT 溢出为基本时间的分频倍，例如基本时间选 18ms,然后软件配置为 4 分频，则溢出时间为 $18*4=72ms$ 。该振荡器仅供给 WDT 电路使用，而不能作为系统主时钟用。WDT 振荡器是否开启由 OPBIT 的 WDTE 配置和软件的 WDTE 位双重决定。

4 复位

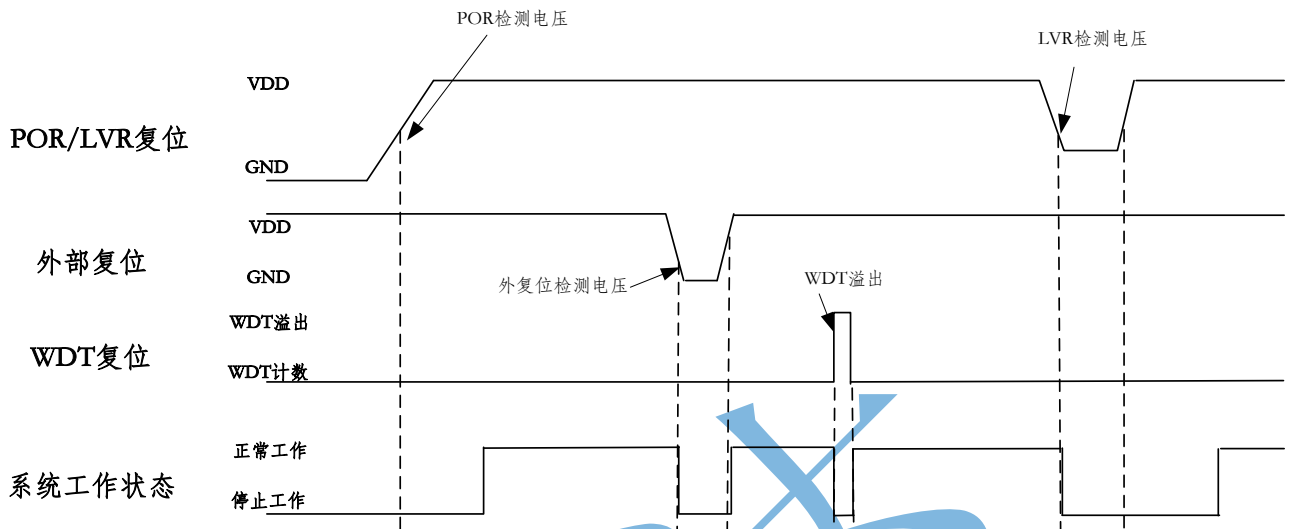
4.1 概述

30P011 有四种可能的复位方式：

- ◇ 上电复位 POR
- ◇ 外部复位
- ◇ 掉电复位 LVR
- ◇ WDT 看门狗复位

任何一种复位发生时，系统将会重新从 0x3ff 地址处开始执行指令；另外系统还会将所有的寄存器重置为默认初始值。

上电复位和 LVR 复位会关闭系统主时钟的振荡器，复位解除后才重新打开振荡器，由于振荡器起振和稳定需要一定的时间，所以系统会在 1024 个时钟周期后开始重新工作。外部复位和 WDT 复位不会关闭系统主时钟振荡器，所以复位解除后 2 个时钟周期后即开始工作。下图是复位产生和系统工作状态之间的关系示意图。



4.2 上电复位

30P011 的上电复位电路可以适应快速、慢速上电的情况，并且当芯片上电过程中出现电源电压抖动时都能保证系统可靠的复位。

上电复位过程可以概括为以下几个步骤：

- (1) 检测系统工作电压，等待电压高于 V_{POR} 并保持稳定；
- (2) 如果外部复位功能开启，则需等待复位引脚电压高于 V_{IH} ；
- (3) 初始化所有寄存器；
- (4) 开启主时钟振荡器，并等待 1024 个时钟周期；
- (5) 上电结束，系统开始执行指令。

4.3 外部复位

外部复位功能是否开启可以通过 OPBIT 的 RSTE 配置，选择 RESE 后复位引脚的内部上拉电阻自动有效。外部复位引脚是施密特结构的，低电平有效。当外复位引脚为高电平时，系统正常运行；为低电平时，系统产生复位。

4.4 LVR 复位电压检测

30P011 的 LVR 电压有八级（1.5V/1.8V/2.0V/2.2V/2.4V/2.6V/3.0V/3.6V），通过 OPBIT 的 VBORS 进行配置（见 § 2.5）。电压检测电路有一定的回滞特性，通常回滞电压为 0.05V 左右，例如，如果选择了 3.6V 的 LVR 电压，则当电源电压下降到约 3.6V 时 LVR 复位有效，而电压需要上升到约 3.65V 时 LVR 复位才会解除。要注意的是不同的工作频率起振电压不同，低于起振电压 LVR 电压不起作用。

4.5 WDT 复位

WDT 看门狗复位是一种对程序正常运行的保护机制。正常情况下，用户软件会按时对 WDT 定时器进行清零操作，定时器不会溢出。若出现异常状况，程序未按预想执行，出现

程序跑飞的状况，那么 WDT 定时器会出现溢出从而触发 WDT 复位，系统重新初始化，返回受控状态。

30P011 的 WDT 看门狗电路有独立的内置 RC 振荡器，不受系统主时钟的影响，即使主时钟振荡器出现异常停振，WDT 复位仍会产生。考虑到分频系数的不同。

有关 WDT 看门狗定时器，可参考 [§3.3](#)。

5 I/O 口

5.1 IO 工作模式

30P011 有 11 个通用双向 IO 口 (P03-P00、P17-P14、P12-P10) 和一个输入口 (P13)。每一个 IO 口都有相应的数据寄存器 (P0、P1) 和方向寄存器 (DDR0、DDR1) 控制，功能如下表所示。

R/W	DDR	功能
W	1	IO 口处于输入状态；数据写到数据寄存器中，端口状态不受影响
W	0	IO 口处于输出状态；数据写到数据寄存器中，端口状态与数据寄存器同时改变

当系统配置成外部晶振模式时，P14、P15 不能作为 IO 口使用，相关的寄存器控制位也无效。

当 IO 处于输出时，读 IO 时可以通过 option 选读寄存器还是端口

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0	-	-	-	-	P03D	P02D	P01D	P00D
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[4:0] **P0n** – P0 口数据位 (n=3-0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1	P17D	P16D	P15D	P14D	-	P12D	P11D	P10D
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

BIT[6:0] **P1n** – P1 口数据位 (n=7-0)

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

BIT[3:0] **DDR0n** – P0 口方向寄存器 (n=3-0)

1: 作为输入口

0: 作为输出口

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DDR1	DDR16	DDR16	DDR15	DDR14	-	DDR13	DDR11	DDR10
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
-----	---	---	---	---	---	---	---	---

BIT[7:0] DDR1n – P1 口方向寄存器 (n=7-0)

1: 作为输入口

0: 作为输出口

注: P13 不能做输出, 但可以做输入 (通过 OPTION 配置)。

5.2 上拉, 下拉电阻和开漏

P0 和 P1 口可通过 PDCON 和 PUCON 选择是否接上拉电阻 (约 25KΩ)。上拉电阻在端口置为输入状态时有效, 置为输出状态时无效。通过 ODCON 控制 P1 端口是否使能开漏输出。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDCON	-	P12PD	P11PD	P10PD	P03PD	P02PD	P01PD	P00PD
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

BIT[6:4] P1nPD – P1 口上拉电阻选择 (n=2-0)

1: P1n 端口下拉电阻无效

0: P1n 端口下拉电阻有效

BIT[3:0] P0nPD – P0 口上拉电阻选择 (n=3-0)

1: P0n 端口下拉电阻无效

0: P0n 端口下拉电阻有效

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUCON	P17PU	P16PU	P15PU	P14PU	-	P12PU	P11PU	P10PU
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

BIT[7:0] P1nPU – P1 口上拉电阻选择 (n=7-4, 2-0)

1: P1n 上拉电阻无效

0: P1n 上拉电阻有效

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ODCON	P17OD	P16OD	P15OD	P14OD	-	P12OD	P11OD	P10OD
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

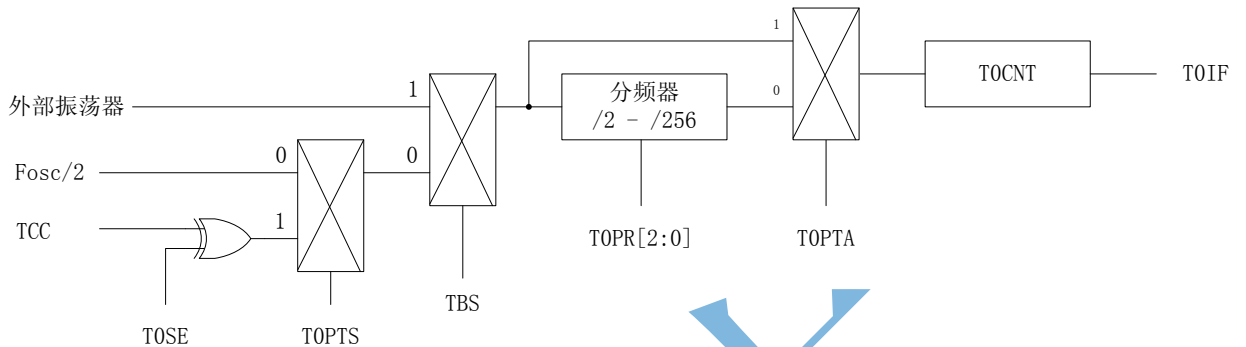
BIT[7:0] P1nOD – P1 口上拉电阻选择 (n=7-4, 2-0)

1: P1n 端口使能开漏输出

0: P1n 端口为普通推挽输出口

6 定时器

6.1 定时器 T0



30P011 的定时器 T0 由 8 位计数器 TCNT0，控制寄存器 TCR0 组成。

TCNT0 的时钟 T0CK 来自系统主时钟 F_{sys} (Fosc/2) 或其分频信号。定时器 T0 的预分频器与 WDT 定时器共用，当 TOPTA=0 时，预分频器分配给 T0 使用；TOPTA=1 时，预分频器分配给 WDT 使用。分频系数由 TOPR[2:0] 决定。

TCNT0 是一个递增计数器，它的值可以读写，当计数到从 FF 溢出到 0 时，产生溢出信号定时中断溢出信号，这个信号有 2 个功能：

- (1) 将 TOIF 标志位置 1，如果此时中断屏蔽位 GIE=1 则产生中断请求；
- (2) 若是 RTC 模式下，并且系统处于 SLEEP 模式下唤醒系统（若 GIE=0 则系统会唤醒，但不会产生中断）

T0 计数时钟频率和计数周期：

$$f = \text{Fosc} / (2 * \text{分频数}) \quad T = 1 / f = 2 * \text{分频数} / \text{Fosc}$$

TCNT0 的计算：假设定时时间为 t

$$\text{TCNT0} = 256 - t / T = 256 - t * f = 256 - (t * \text{Fosc} / (2 * \text{分频数}))$$

与定时器 T0 相关的寄存器说明

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TOCNT	T0C7	T0C6	T0C5	T0C4	T0C3	T0C2	T0C1	T0C0
R/W	R/W	RR/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] **TOC[7:0]** – TCNT0 的值，这是一个读写寄存器，用于设定定时时间。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TOCR		INTOM	TOPTS	TOSE	TOPTA	TOPR2	TOPR1	TOPR0
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[7] 保留

BIT[6] **INTOM** – 外部中断触发方式
0: 外部中断源下降沿触发中断

1: 外部中断源上升沿触发中断
BIT[5] TOPTS – T1 时钟源选择
 0: T0 时钟源为 CPU 运行时钟/RTC

1: T0 时钟源为 TCC
BIT[4] TOSE – TCC 计数方式
 0: TCC 上升沿计数
 1: TCC 下降沿计数

BIT[3] TOPTA – 预分频器分配
 0: 预分频器分配给 T0
 1: 预分频器分配给 WDT

BIT[2:0] TOPR[2:0] – T0 预分频倍数选择
 T0 和 WDT 共用预分频器的分频率的选择位, 系统复位时被置为 111。TOPR[2:0] 的值和分频倍数的对应关系见下表。

TOPR[2]	TOPR [1]	TOPR [0]	T0	WDT
0	0	0	2	1
0	0	1	4	2
0	1	0	8	4
0	1	1	16	8
1	0	0	32	16
1	0	1	64	32
1	1	0	128	64
1	1	1	256	128

注: 当定时器需要 1 分频时, 把预分频器分给 WDT; 当 WDT 需要 1 分频时, 把预分频器分给 T0; 但无法实现两者都是 1 分频。

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

BIT[7] TBS – P1 口上拉电阻选择 (n=7-4, 2-0)
 0: 定时器 0 的时钟源由 TOPTS 决定
 1: 定时器 1 的时钟源为外部 32768 振荡器 (内部高频振荡器&RTC 模式)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTECON	GIE	-	-	-	-	INTOIE	KBIE	TOIE
R/W	R/W	-	-	-	-	R/W	R/W	R/W
初始值	0	-	-	-	-	0	0	0

BIT[7] GIE – 总中断使能
 0: 屏蔽所有中断
 1: 中断源是否产生中断有相应的控制位决定

BIT[0] TOIE – 定时器 0 溢出中断使能
 0: 屏蔽定时器 0 溢出中断
 1: 使能定时器 0 溢出中断

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

BIT[0] TOIF – 定时器 0 溢出中断标志
 0: 定时器 0 未发生溢出
 1: 定时器 0 发生溢出（需软件清零）

T0 使用的例子：使用 4M 晶振，精确定时产生一个 flag_time_10ms 标志

说明，系统的时钟为 $4M/2=2M$ ，2 分频后为 1M，定时器计一个数为 $1/f=1\mu s$ ，定时器溢出一次为 256 μs

定时器初始化代码：

```

MOVAR T0CR ;4M,2 分频
ANDAI 0X40 ;保留 BIT6
ADDAI 0X00 ;预分频给 T0, 时钟源来自 FOSC/2
MOVRA T0CR ;写入 T0CR
CLRR T0CNT
BSET T0IE ;开定时中断
BSET GIE ;开总中断
    
```

中断代码：

```

ORG 0X08 ;中断向量
INTERRUPT_SUB:
MOVRA ACC_TEMP
SWAPAR STATUS
MOVRA STATUS_TEMP ;保护 ACC 和状态寄存器, MOVRA 和 SWAPAR 指令不影响状态标志位
;*****判断是哪个中断*****
START_JUST_INT:
JBCLRT0IE ;判断有没有开定时中断
JBSET T0IF ;判断定时器中断有没有发生
GOTO JUST_INT0
;*****TIMER0*****
BCLR T0IF ;必须手动清中断标志
INCR TIME_250US
MOVAI 6
ADDRA TIME_1US ;每次中断增加 256US

MOVAR 250
RSUBAR TIME_1US ;TIME_1US-250
    
```

```

JBSET C          ;>=则跳过一行执行
GOTO CHECK_1MS

MOVRA TIME_1US
INCR TIME_250US
CHECK_1MS:
MOVAI 4          ;ACC=4
RSUBAR TIME_1MS ;TIME_1MS-4
JBSET C          ;>= 则跳过一行再执行
GOTO END_TIME0_INTERRUPT ;<4 则没到 1MS，退出

MOVRA TIME_125US ;相减后剩下的保存起来
INCR TIME_1MS
MOVAR TIME_1MS
SUBAI 10         ;10-ACC
JBSET Z          ;TIME_1MS==10 则跳过一行
GOTO END_TIME0_INTERRUPT
CLR TIME_1MS
BSET FLAG_TIME_10MS ;10MS 到
END_TIME0_INTERRUPT:
;*****END_TIME0*****
JUST_T1:
JBCLR T1IE      ;判断有没有开定时中断
JBSET T1IF      ;判断定时器中断有没有发生
GOTO JUST_INT0
;*****T1*****
BCLR T1IF
;*****END T1*****
JUST_INT0:
JBCLR INTOIE    ;判断有没有开 INTO 中断
JBSET INTOIF    ;判断 INTO 中断有没有发生
GOTO JUST_KBIM
;*****INT0*****
BCLR INTOIF     ;必须手动清中断标志

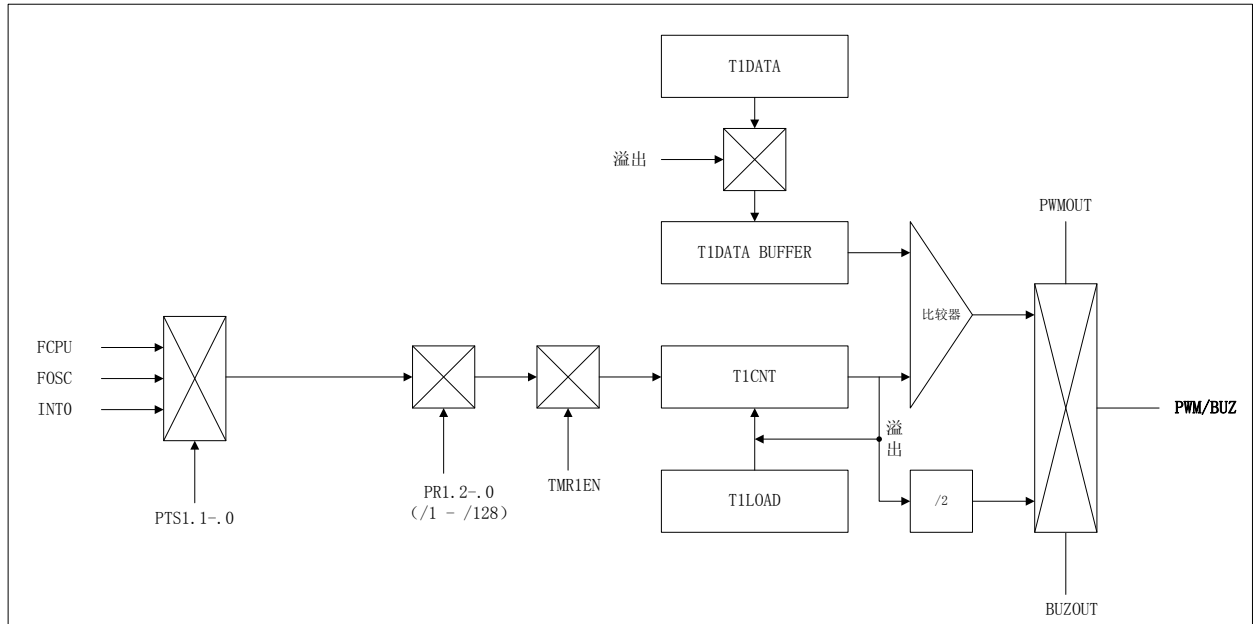
;*****END INTO*****
JUST_KBIM:
JBCLR KBIE      ;判断有没有开键盘中断
JBSET KBIF      ;判断键盘中断有没有发生
GOTO END_JUST_INT
;*****KBIM*****
BCLR KBIF       ;必须手动清中断标志

;*****END_KBIM*****
    
```

```

END_JUST_INT:
EXIT_INTERRUPT:
SWAPRSTATUS_TEMP
MOVRA STATUS
SWAPR ACC_TEMP
SWAPRACC_TEMP ;恢复状态寄存器和 ACC
RETIE          ;中断返回
    
```

6.2 定时器 T1



PTS1 可选择 TIMER1 的时钟源，PR1 可选择 TIMER1 的预分频比，所选中的时钟源通过预分频器后产生 T1CNT 的计数时钟。

当 T1CNT 递减到 0 时，此时产生 TIMER1 溢出中断请求标志 T1IF 置 1，重载寄存器 T1LOAD 值自动置入 T1CNT，T1DATA 的值写入缓冲器 T1DATABUFER 用于新的占空比波形生成，BUZZ 信号反相。

通过 PR1 可选择时钟源的分频比，可选择范围为 1 - 128 分频，对 T1CNT 的写操作将使预分频器清零，分频比保持不变。

（当 PWM1OUT=1 时，将输出 PWM 波形，当 T1CNT 计数到与 T1DATA 相等时，PWM 输出置 1（注：若 T1DATA 为 0，PWM 输出一直为 0）；当 T1CNT 计数溢出时，PWM 输出清 0，PWM 占空比的计算如下：

$$\text{PWM 高电平时间} = (\text{T1DATA}) * \text{T1CNT 计数时钟周期}$$

$$\text{PWM 周期 (TIMER1 的溢出周期)} = (\text{T1LOAD}+1) * \text{T1CNT 的计数周期}$$

$$\text{PWM 占空比} = (\text{T1DATA} / (\text{T1LOAD}+1))$$

当 BUZZIUT=1 且 PWM1OUT=0 时，输出 BUZ 信号，BUZ 信号的输出频率为 TIMER1 溢出频率的 2 分频。

与定时器 T1 相关的寄存器说明如下

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMCR	TBS						T1IE	T1IF

<i>R/W</i>	R/W						R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[1] T1IE - timer1 中断使能位

1: timer1 中断使能

0: timer1 中断不使能

BIT[0] T1IF - timer1 溢出标志位 (写 1 无效)

1: timer1 溢出

0: timer1 无溢出

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CR	TMR1EN	PWMOUT	BUZOUT	T1PTS1	T1PTS0	T1PR2	T1PR1	T1PR0
<i>R/W</i>	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] TMR1EN - T1 使能控制

0: 关闭 timer1

1: 启动 timer1

BIT[6] PWMOUT - PWM1 选择

0: 禁止 PWM1 输出, 相应端口作为 I/O 口

1: 允许 PWM1 输出, 相应端口输出输出 PWM 信号 (相应的 IO 作为输出口才有输出)

BIT[5] BUZOUT - BUZ1 选择

0: 禁止 BUZ1 输出, 相应端口作为 I/O 口

1: 允许 BUZ1 输出 (PWM1OUT 必须为 0), 相应端口输出输出 PWM 信号 (相应的 IO 作为输出口才有输出)

BIT[4:3] T1PTS[1:0] - TIMER1 时钟源

PTS.1-0	TIMER1 时钟源
00	FCPU
01	FOSC
10	INT0 上升沿
11	INT0 下降沿

BIT[2:0] T1PR[2:0] - T1 预分频倍数选择

PR1[2]	PR1[1]	PR1[0]	T1CNT
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

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

T1CNT	T1C7	T1C6	T1C5	T1C4	T1C3	T1C2	T1C1	T1C0
R/W	R/W	RR/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值								

BIT[7:0] T0C[7:0] – T1CNT 的值，这是一个读写寄存器，用于设定定时时间。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1LOAD	<i>T1LO7</i>	<i>T1LO6</i>	<i>T1LO5</i>	<i>T1LO4</i>	<i>T1LO3</i>	<i>T1LO2</i>	<i>T1LO1</i>	<i>T1LO0</i>
R/W	R/W	RR/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值								

BIT[7:0] T0L[7:0] – T1LOAD 的值，这是一个只写寄存器，用于设置 T1LOAD 的值。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1DATA	<i>T1D7</i>	<i>T1D6</i>	<i>T1D5</i>	<i>T1D4</i>	<i>T1D3</i>	<i>T1D2</i>	<i>T1D1</i>	<i>T1D0</i>
R/W	R/W	RR/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值								

T1 使用的例子：使用 4M 晶振，精确定时产生一个 flag_time_10ms 标志

说明，系统的时钟为 4M/2=2M，2 分频后为 1M，定时器计一个数为 1/F=1US，定时器溢出一次为 125US

定时器初始化代码：

```

movai 249
movra T1LOAD

movai 0x00 ;4M,2T,1 分频;中断一次 250us
movra T1CR

bset T1MEN ;开定时器 1
bset T1IE ;使能定时器 1 中断
bset GIE ;开总中断
    
```

中断代码：

```

ORG 0x08 ;其它中断向量
    
```

interrupt_sub:

```

movra acc_temp
swapar STATUS
movra status_temp
    
```

;*****判断是哪个中断*****

start_just_int:

```

jbclr TOIE
jbset TOIF
goto just_t1
    
```

;*****timer0*****

```

bclr TOIF
    
```

end_time0_interrupt:

```

;*****end_time0*****
just_t1:
    jbclr    T1IE
    jbset    T1IF
    goto     just_int0
;*****timer1*****
    bclr     T1IF

    incr     time_125us
    jbset    time_125us,3
    goto     end_time1_interrupt

    clr      time_125us
    incr     time_1ms
    movar    time_1ms
    isubai   10        ;10-acc
    jbset    Z
    goto     end_time1_interrupt
    clr      time_1ms
    bset     flag_time_10ms
end_t1_time:
end_time1_interrupt:
;*****end_timer1*****
just_int0:
    jbclr    INTOIE
    jbset    INTOIF
    goto     just_kbim
;*****int0*****
    bclr     INTOIF
;*****end int0*****
just_kbim:
    jbclr    KBIE
    jbset    KBIF
    goto     end_just_int
;*****kbim*****
    bclr     KBIF
;*****end_kbim*****
end_just_int:
exit_interrupt:
    swapar   status_temp
    movra    STATUS
    swapr    acc_temp
    swapar   acc_temp
    retie
    
```

6.3 WDT 定时器

WDT 定时器的时钟源于一个独立的 RC 振荡器（见 §0），并可以选择是否经过预分频器（见 §0）。WDT 定时器可以用来产生 WDT 复位或唤醒 SLEEP 模式。

当 WDTEN (MCR[7]) 为 0 时，WDT 定时器被软禁止；为 1 时软使能，若要 WDT 使能还需要 OPTION 选项使能。

因为 WDT 定时器的时钟源与系统主时钟无关，所以，即使系统进入 SLEEP 模式，WDT 定时器仍会工作，但在 SLEEP 模式下 WDT 只能产生唤醒信号，并不会产生复位信号。

在正常工作下，当 WDT 计数溢出时，芯片进行复位。

考虑到与分频倍数，WDT 定时器无分频的周期范围是 4.5ms—288ms(由 option 决定)。分频器和 T0 共用，当分频器给 T0 时，WDT 为 1 分频（无分频）；反之当分频器给 WDT 时 T0 分 1 分频（无分频），由 TOPTA (TOCR[3]) 位决定。

与 WDT 定时器相关的寄存器 MCR, TOCR。



	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCR	WDTEN	EIS	-	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-	-
初始值	0	0	-	-	-	-	-	-

BIT[7]

WDTEN – 看门狗定时器软件使能

0: 关闭看门狗定时器断

1: 使能看门狗定时器（必须配置字使能看门狗定时器）

7 中断

7.1 概述

30P011 的中断有外中断 (INT0)、键盘中断 (KBI)、定时器中断 (TM0I, TM1I)。外部中断、键盘中断、定时器中断可被 CPU 状态寄存器 INTECON 的 GIE 位屏蔽。

中断响应过程如下：

- ◇ 当发生中断请求时，CPU 将相关下一条要执行的指令的地址压栈保存（累加器 A 和状态寄存器需要软件保护），对中断屏蔽位 GIE 清 0，禁止中断响应。与复位不同，硬件中断不停止当前指令的执行，而是暂时挂起中断直到当前指令执行完成。
- ◇ CPU 执行中断时，程序跳到中断向量 0x08 地址开始执行中断代码，中断代码应该先保存累加器 A 和状态寄存器，然后判断是哪一个中断响应。
- ◇ 执行中断内容后应该恢复累加器 A 和状态寄存器，然后执行 RETIE 返回主程序。这时，从堆栈取出 PC 的值，然后从中断发生时的那条指令的后一条指令继续执行。30P011 的中断向量地址是 0x08。

7.2 外中断

30P011 的 P10 可以作为外中断输入 INT0，可以响应上升沿、下降沿 2 种方式的中断触发条件。

外部中断 INT0 控制位功能如下：

- (1) EIS (MCR[6]) 为 P10 外中断的使能位。
- (2) INTOE 为中断使能位，INTOE=0 时，不允许外中断；当 INTOE=1 时，允许外中断。
- (3) INTOM (T0CR[6]) 为中断触发位，有下降沿触发、上升沿触发 2 种方式。
- (4) INTOF 为中断标志位，INTOF 不会自动清零，必须通过软件对其清零。

注：要使用外中断 INT0，还必须将 P14 口成输入状态，即令 DDR14=1。

相关寄存器 MCR, T0C, INTECO, INTFLAG

7.3 键盘中断

30P011 的 P17-P10 可以作为键盘中断输入，这些键盘中断请求信号共用一个中断请求端，因而在中断服务程序中通常还要读取 IO 数据寄存器来判断究竟是哪一个键盘输入口有中断请求。

键盘中断请求与以下因素有关。

- (1) KBIE(INTECON[1])位。KBIE 是键盘中断允许位，当 KBIE=1 时，允许键盘中断，KBIE=0 时，不允许键盘中断。
- (2) KBIM[7:0]（对应 P1[7:0]），当 KBIMn=1 时，表示 P1n 的键盘中断功能打开，否则，键盘中断功能关闭。
- (3) P17-P10 的状态，引脚输入电平状态与输出数据寄存器中的值进行比较，如不同，则触发键盘中断请求。
- (4) 产生中断请求时 KBIF (INTFLAG[1]) 位为 1，若此时开总中断，则发生中断，此位须软件清零。

应用时，可先将当前端口状态读入并写到输出数据寄存器中，这样，当引脚电平变化时就会触发键盘中断请求。

相关寄存器说明 INTECON, INTFLAG。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
KBIM	KBIM6-	KBIM6	KBIM5	KBIM4	KBIM3	KBIM2	KBIM1	KBIM0
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[6:0] **KBIM0n** – P1n 口键盘中断允许位 (n=7-0)

- 0: 不允许键盘中断
- 1: 允许键盘中断

7.4 定时器中断

定时器 T0、T1、在计数溢出时会产生中断请求 TMI0、TMI1，它们有各自的屏蔽位 TOIM、T1IM 及标志位 TOIF、T1IF。见 [§0](#)、[§0](#)。

8 系统低功耗 SLEEP 模式

SLEEP 指令可使 MCU 进入 SLEEP 低功耗工作模式，同时对 MCU 会产生以下影响：

- ✧ 系统主时钟的振荡器停止振荡
- ✧ RAM 内容保持不变
- ✧ 所有的输入输出端口保持原态不变
- ✧ 所有的内部操作全部停止 (WDT 不受控)

以下情况使 MCU 退出 SLEEP 方式：

- ✧ 有外中断 INTO 请求发生
- ✧ 有键盘中断 KBI 请求发生
- ✧ 有 timer 中断 (RTC) 请求发生
- ✧ 有 WDT 溢出 WDTI 中断请求发生
- ✧ 任何形式的系统复位发生

SLEEP 工作模式下，系统停止了几乎所有的操作，所以整体功耗水平非常低。

注：进入 sleep 模式并不会自动打开总中断，但只要中断请求发生就唤醒，如果没打开总中断是不会进中断的。有中断请求不表示要进中断

9 指令集概述

30P011 是一种精简指令集 (RISC)，总共有 55 条指令。指令为 14 位，由操作码 (OPCODE) 和 0~2 个操作数 (operand) 组成。除跳转指令外，其他指令执行周期均为 1 个周期 (详见 §0)。

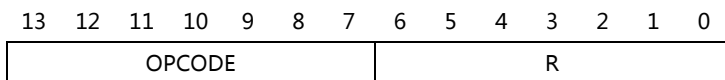
表一是文中所用符号的说明。

符号	说明	符号	说明
R/r	寄存器地址	C	进位标志
ACC	ACC 寄存器	DC	半进位标志
b	位地址	Z	零标志
K/k	立即数	WDT	看门狗定时器
label	标号	TO	看门狗溢出标志
TOS	栈顶	PD	休眠模式标志
PC	PC 指针		

表一

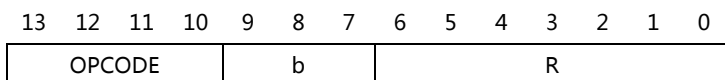
指令按照功能可分为 4 类，即字节操作指令、位操作指令、立即数指令、特殊控制指令，以下简要说明各类指令的格式。

(1) 字节操作指令



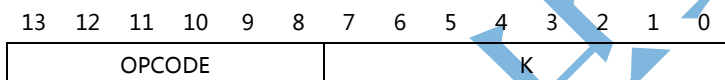
字节操作指令的 OPCODE 长度为 7 位，寄存器地址宽度为 7 位，即地址访问范围 00H~7FH。

(2) 位操作指令



位操作指令的 OPCODE 长度为 4 位，位地址（0~7）占 3 位，寄存器地址也是 7 位。

(3) 立即数指令

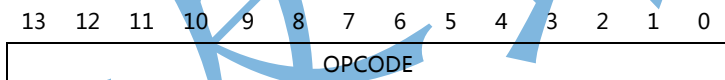


立即数指令的 OPCODE 长度为 6 位，立即数长度为 8 位。



CALL/GOTO 指令的 OPCODE 长度为 3 位，立即数长度为 11 位。

(4) 特殊控制指令



特殊控制指令全部 14 位均为 OPCODE。

9.1 指令汇总表

助记符	说明	操作	周期数	指令码	影响
ADDAR R	寄存器 R 内容和 ACC 相加, 结果存到 ACC	$R+ACC \rightarrow ACC$	1	01 1111 0rrr rrrr	C, DC, Z
ADDRA R	寄存器 R 内容和 ACC 相加, 结果存到 R	$R+ACC \rightarrow R$	1	01 1111 1rrr rrrr	C, DC, Z
ADCAR R	带 C 标志的加法, 结果存到 ACC	$R+ACC+C \rightarrow ACC$	1	00 0111 0rrr rrrr	C, DC, Z
ADCRA R	带 C 标志的加法, 结果存到 R	$R+ACC+C \rightarrow R$	1	00 0111 1rrr rrrr	C, DC, Z
RSUBAR R	寄存器 R 内容和 ACC 相减, 结果存到 ACC	$R-ACC \rightarrow ACC$	1	01 0010 0rrr rrrr	C, DC, Z
RSUBRA R	寄存器 R 内容和 ACC 相减, 结果存到 R	$R-ACC \rightarrow R$	1	01 0010 1rrr rrrr	C, DC, Z
RSBCAR R	带 C 标志的减法, 结果存到 ACC	$R-ACC-/C \rightarrow ACC$	1	00 0110 0rrr rrrr	C, DC, Z
RSBCRA R	带 C 标志的减法, 结果存到 R	$R-ACC-/C \rightarrow R$	1	00 0110 1rrr rrrr	C, DC, Z
ANDAR R	寄存器 R 内容和 ACC 与操作, 结果存到 ACC	$R \text{ and } ACC \rightarrow ACC$	1	01 1110 0rrr rrrr	Z
ANDRA R	寄存器 R 内容和 ACC 与操作, 结果存到 R	$R \text{ and } ACC \rightarrow R$	1	01 1110 1rrr rrrr	Z
ORAR R	寄存器 R 内容和 ACC 或操作, 结果存到 ACC	$R \text{ or } ACC \rightarrow ACC$	1	01 0111 0rrr rrrr	Z
ORRA R	寄存器 R 内容和 ACC 或操作, 结果存到 R	$R \text{ or } ACC \rightarrow R$	1	01 0111 1rrr rrrr	Z
XORAR R	寄存器 R 内容和 ACC 异或操作, 结果存到 ACC	$R \text{ xor } ACC \rightarrow ACC$	1	01 0000 0rrr rrrr	Z
XORRA R	寄存器 R 内容和 ACC 异或操作, 结果存到 R	$R \text{ xor } ACC \rightarrow R$	1	01 0000 1rrr rrrr	Z
COMAR R	对 R 取反, 结果存到 ACC	$R \text{ 取反} \rightarrow ACC$	1	01 1100 0rrr rrrr	Z
COMR R	对 R 取反, 结果存到 R	$R \text{ 取反} \rightarrow R$	1	01 1100 1rrr rrrr	Z
CLRA	对 ACC 清零	$0 \rightarrow ACC$	1	01 1101 0000 0000	Z
CLRR R	对 R 清零	$0 \rightarrow R$	1	01 1101 1rrr rrrr	Z
RLAR R	带 C 标志左移, 结果存到 ACC	$R[7] \rightarrow C$ $R[6:0] \rightarrow ACC[7:1]$ $C \rightarrow ACC[0]$	1	01 0100 0rrr rrrr	C
RLR R	带 C 标志左移, 结果存到 R	$R[7] \rightarrow C$ $R[6:0] \rightarrow R[7:1]$ $C \rightarrow R[0]$	1	01 0100 1rrr rrrr	C
RRAR R	带 C 标志右移, 结果存到 ACC	$C \rightarrow ACC[7]$ $R[7:1] \rightarrow ACC[6:0]$ $R[0] \rightarrow C$	1	01 0011 0rrr rrrr	C
RRR R	带 C 标志右移, 结果存到 R	$C \rightarrow R[7]$ $R[7:1] \rightarrow R[6:0]$ $R[0] \rightarrow C$	1	01 0011 1rrr rrrr	C
SWAPAR R	交换 R 的高低字节, 结果存到 ACC	$R[7:4] \rightarrow ACC[3:0]$ $R[3:0] \rightarrow ACC[7:4]$	1	01 0001 0rrr rrrr	-
SWAPR R	交换 R 的高低字节, 结果存到 R	$R[7:4] \rightarrow R[3:0]$ $R[3:0] \rightarrow R[7:4]$	1	01 0001 1rrr rrrr	-
MOVAR R	将 R 存到 ACC	$R \rightarrow ACC$	1	01 0110 0rrr rrrr	Z
MOVR R	将 R 存到 R	$R \rightarrow R$	1	01 0110 1rrr rrrr	Z

MOVRA R	将 ACC 存到 R	ACC→R	1	01 0101 1rrr rrrr	-
INCAR R	R 加 1, 结果存到 ACC	R+1→ACC	1	01 1001 0rrr rrrr	Z
INCR R	R 加 1, 结果存到 R	R+1→R	1	01 1001 1rrr rrrr	Z
DECAR R	R 减 1, 结果存到 ACC	R-1→ACC	1	01 1011 0rrr rrrr	Z
DECR R	R 减 1, 结果存到 R	R-1→R	1	01 1011 1rrr rrrr	Z
JZAR R	R 加 1, 结果存到 ACC。结果为 0, 则跳过下一条指令	R+1→ACC, 结果为 0, 则 PC+2→PC	1 或 2 ⁽¹⁾	01 1000 0rrr rrrr	-
JZR R	R 加 1, 结果存到 R。结果为 0, 则跳过下一条指令	R+1→R, 结果为 0, 则 PC+2→PC	1 或 2 ⁽¹⁾	01 1000 1rrr rrrr	-
DJZAR R	R 减 1, 结果存到 ACC, 结果为 0, 则跳过下一条指令	R-1→ACC, 结果为 0, 则 PC+2→PC	1 或 2 ⁽¹⁾	01 1010 0rrr rrrr	-
DJZR R	R 减 1, 结果存到 R, 结果为 0, 则跳过下一条指令	R-1→R, 结果为 0, 则 PC+2→PC	1 或 2 ⁽¹⁾	01 1010 1rrr rrrr	-
BCLR R, b	对 R 的第 b 位清零	0→R[b]	1	11 11bb brrr rrrr	-
BSET R, b	对 R 的第 b 位置 1	1→R[b]	1	11 10bb brrr rrrr	-
JBCLR R, b	如 R 的第 b 位为 0, 则跳过下一条指令	如 R[b]=0, 则 PC+2→PC	1 或 2 ⁽¹⁾	11 01bb brrr rrrr	-
JBSET R, b	如 R 的第 b 位为 1, 则跳过下一条指令	如 R[b]=1, 则 PC+2→PC	1 或 2 ⁽¹⁾	11 00bb brrr rrrr	-
ADDAI K	立即数 K 和 ACC 相加, 结果存到 ACC	K+ACC→ACC	1	00 1110 kkkk kkkk	C, DC, Z
ISUBAI K	立即数和 ACC 相减, 结果存到 ACC	K-ACC→ACC	1	00 1001 kkkk kkkk	C, DC, Z
ANDAI K	立即数 K 和 ACC 与操作, 结果存到 ACC	K and ACC→ACC	1	00 1101 kkkk kkkk	Z
ORAI K	立即数 K 和 ACC 或操作, 结果存到 ACC	K or ACC→ACC	1	00 1100 kkkk kkkk	Z
XORAI K	立即数和 ACC 异或, 结果存到 ACC	K xor ACC→ACC	1	00 1000 kkkk kkkk	Z
MOVAI K	将立即数存到 ACC	K→ACC	1	00 1011 kkkk kkkk	-
RETAI K	从子程序返回, 并将立即数存到 ACC	TOS→PC K→ACC	2	00 1010 kkkk kkkk	-
RETURN	从子程序返回	TOS→PC	2	00 0000 0000 1100	-
RETIE	从中断返回	TOS→PC 1→GIE	2	00 0000 0000 1101	-
CALL K	子程序调用	PC+1→TOS K→PC	2	10 0kkk kkkk kkkk	-
GOTO K	无条件跳转	K→PC	2	10 1kkk kkkk kkkk	-
NOP	空操作	空操作	1	00 0000 0000 0000	-
DAA	加法后, 将 ACC 的值调整到十进制	ACC(十六进制)→ACC(十进制)	1	00 0000 0000 1010	C
DSA	减法后, 将 ACC 的值调整到十进制	ACC(十六进制)→ACC(十进制)	1	00 0000 0000 1001	-
CLRWDT	清看门狗定时器	0→WDT	1	00 0000 0000 1110	TO, PD
STOP	进入休眠模式	0→WDT 进入休眠模式	1	00 0000 0000 1011	TO, PD

注 (1) 跳转条件成立, 则指令需 2 个周期, 否则指令只需 1 个周期。

9.2 指令详细说明

ADCAR

语法： ADCAR R
操作数： $0 \leq R \leq 127$
操作： $R + ACC + C \rightarrow ACC$
影响状态： C, DC, Z
 C 相加结果产生进位，则 $1 \rightarrow C$ ；否则 $0 \rightarrow C$
 DC 相加结果第 3 位向第 4 位产生进位，则 $1 \rightarrow DC$ ；否则 $0 \rightarrow DC$
 Z 相加结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 带 C 标志的加法，结果存到 ACC
周期数： 1

ADCRA

语法： ADCRA R
操作数： $0 \leq R \leq 127$
操作： $R + ACC + C \rightarrow R$
影响状态： C, DC, Z
 C 相加结果产生进位，则 $1 \rightarrow C$ ；否则 $0 \rightarrow C$
 DC 相加结果第 3 位向第 4 位产生进位，则 $1 \rightarrow DC$ ；否则 $0 \rightarrow DC$
 Z 相加结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 带 C 标志的加法，结果存到 R
周期数： 1

ADDAI

语法： ADDAI K
操作数： $0 \leq K \leq 255$
操作： $K + ACC \rightarrow ACC$
影响状态： C, DC, Z
 C 相加结果产生进位，则 $1 \rightarrow C$ ；否则 $0 \rightarrow C$
 DC 相加结果第 3 位向第 4 位产生进位，则 $1 \rightarrow DC$ ；否则 $0 \rightarrow DC$
 Z 相加结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 立即数 K 和 ACC 相加，结果存到 ACC
周期数： 1

ADDAR

语法： ADDAR R
操作数： $0 \leq R \leq 127$
操作： $R + ACC \rightarrow ACC$
影响状态： C, DC, Z
C 相加结果产生进位，则 $1 \rightarrow C$ ；否则 $0 \rightarrow C$
DC 相加结果第 3 位向第 4 位产生进位，则 $1 \rightarrow DC$ ；否则 $0 \rightarrow DC$
Z 相加结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 寄存器 R 内容和 ACC 相加，结果存到 ACC
周期数： 1

ADDRA

语法： ADDRA R
操作数： $0 \leq R \leq 127$
操作： $R + ACC \rightarrow R$
影响状态： C, DC, Z
C 相加结果产生进位，则 $1 \rightarrow C$ ；否则 $0 \rightarrow C$
DC 相加结果第 3 位向第 4 位产生进位，则 $1 \rightarrow DC$ ；否则 $0 \rightarrow DC$
Z 相加结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 寄存器 R 内容和 ACC 相加，结果存到 R
周期数： 1

ANDAI

语法： ADCAI K
操作数： $0 \leq K \leq 255$
操作： $K \text{ and } ACC \rightarrow ACC$
影响状态： Z
Z 与的结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 立即数 K 和 ACC 与操作，结果存到 ACC
周期数： 1

ANDAR

语法： ANDAR R
操作数： $0 \leq R \leq 127$
操作： $R \text{ and } ACC \rightarrow ACC$

影响状态： Z
Z 与的结果为 0，则 1→Z；否则 0→Z
说明： 寄存器 R 内容和 ACC 与操作，结果存到 ACC
周期数： 1

ANDRA

语法： ANDRA R
操作数： $0 \leq R \leq 127$
操作： R and ACC→R
影响状态： Z
Z 与的结果为 0，则 1→Z；否则 0→Z
说明： 寄存器 R 内容和 ACC 与操作，结果存到 R
周期数： 1

BCLR

语法： BCLR R, b
操作数： $0 \leq R \leq 127, 0 \leq b \leq 7$
操作： $0 \rightarrow R[b]$
影响状态： 无影响
说明： 对 R 的第 b 位清零
周期数： 1

BSET

语法： BSET R, b
操作数： $0 \leq R \leq 127, 0 \leq b \leq 7$
操作： $1 \rightarrow R[b]$
影响状态： 无影响
说明： 对 R 的第 b 位置 1
周期数： 1

CALL

语法： CALL K
操作数： $0 \leq K \leq 2047$
操作： $PC+1 \rightarrow TOS(\text{Top of Stack}); K \rightarrow PC$
影响状态： 无影响

说明： 子程序调用，下一条指令的 PC 地址压栈，当前 PC 赋值为 K
周期数： 2

CLRA

语法： CLRA
操作数： 无
操作： 0→ACC
影响状态： Z
Z 1→Z
说明： 对 ACC 清零
周期数： 1

CLRR

语法： CLRR R
操作数： $0 \leq R \leq 127$
操作： 0→R
影响状态： Z
Z 1→Z
说明： 对 R 清零
周期数： 1

CLRWDT

语法： CLRWDT
操作数： 无
操作： 0→WDT
影响状态： TO,PD
TO 1→TO
PD 1→PD
说明： 对看门狗定时器 (WDT) 清零
周期数： 1

COMAR

语法： COMAR R
操作数： $0 \leq R \leq 127$
操作： R 取反→ACC

影响状态： Z
 Z 取反结果为 0，则 1→Z；否则 0→Z
说明： 对寄存器 R 内容取反，结果存到 ACC
周期数： 1

COMR

语法： COMR R
操作数： $0 \leq R \leq 127$
操作： R 取反→R
影响状态： Z
 Z 取反结果为 0，则 1→Z；否则 0→Z
说明： 对寄存器 R 内容取反，结果存到 R
周期数： 1

DAA

语法： DAA
操作数： 无
操作： ACC(十六进制)→ACC(十进制)
影响状态： C
 C 根据十进制调整的情况设置 C
说明： 加法后，将 ACC 的值调整为十进制
周期数： 1

DECAR

语法： DECAR R
操作数： $0 \leq R \leq 127$
操作： R-1→ACC
影响状态： Z
 Z 减 1 结果为 0，则 1→Z；否则 0→Z
说明： 对寄存器 R 内容减 1，结果存到 ACC
周期数： 1

DECR

语法： DECR R
操作数： $0 \leq R \leq 127$

操作： R-1→R
影响状态： Z
Z 减 1 结果为 0，则 1→Z；否则 0→Z
说明： 对寄存器 R 内容减 1，结果存到 R
周期数： 1

DJZAR

语法： DJZAR R
操作数： $0 \leq R \leq 127$
操作： R-1→ACC; 结果为 0，则 PC+2→PC; 否则 PC+1→PC
影响状态： 无影响
说明： 对寄存器 R 内容减 1，结果存到 ACC，结果为 0，则跳过下一条指令
周期数： 1 (跳转条件不成立) 或 2 (跳转条件成立)

DJZR

语法： DJZR R
操作数： $0 \leq R \leq 127$
操作： R-1→R; 结果为 0，则 PC+2→PC; 否则 PC+1→PC
影响状态： 无影响
说明： 对寄存器 R 内容减 1，结果存到 R，结果为 0，则跳过下一条指令
周期数： 1 (跳转条件不成立) 或 2 (跳转条件成立)

DSA

语法： DSA
操作数： 无
操作： ACC(十六进制)→ACC(十进制)
影响状态： 不影响
说明： 减法后，将 ACC 的值调整为十进制
周期数： 1

GOTO

语法： GOTO K
操作数： $0 \leq K \leq 2047$
操作： K→PC
影响状态： 无影响

说明： 无条件跳转
周期数： 2

INCAR

语法： INCAR R
操作数： $0 \leq R \leq 127$
操作： $R+1 \rightarrow \text{ACC}$
影响状态： Z
Z 加 1 结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 对寄存器 R 内容加 1，结果存到 ACC
周期数： 1

INCR

语法： INCR R
操作数： $0 \leq R \leq 127$
操作： $R+1 \rightarrow R$
影响状态： Z
Z 加 1 结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 对寄存器 R 内容加 1，结果存到 R
周期数： 1

ISUBAI

语法： ISUBAI K
操作数： $0 \leq K \leq 255$
操作： $K - \text{ACC} \rightarrow \text{ACC}$
影响状态： C, DC, Z
C 相减结果产生借位，则 $0 \rightarrow C$ ；否则 $1 \rightarrow C$
DC 相减结果第 3 位向第 4 位产生借位，则 $0 \rightarrow DC$ ；否则 $1 \rightarrow DC$
Z 相减结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 立即数 K 和 ACC 相减，结果存到 ACC
周期数： 1

JBCLR

语法： JBCLR R, b
操作数： $0 \leq R \leq 127, 0 \leq b \leq 7$

操作： 如 $R[b]=0$ ，则 $PC+2 \rightarrow PC$ ；否则 $PC+1 \rightarrow PC$
影响状态： 无影响
说明： 如 R 的第 b 位为 0，则跳过下一条指令
周期数： 1（跳转条件不成立）或 2（跳转条件成立）

JBSET

语法： JBSET R, b
操作数： $0 \leq R \leq 127, 0 \leq b \leq 7$
操作： 如 $R[b]=1$ ，则 $PC+2 \rightarrow PC$ ；否则 $PC+1 \rightarrow PC$
影响状态： 无影响
说明： 如 R 的第 b 位为 1，则跳过下一条指令
周期数： 1（跳转条件不成立）或 2（跳转条件成立）

JZAR

语法： JZAR R
操作数： $0 \leq R \leq 127$
操作： $R+1 \rightarrow ACC$ ；结果为 0，则 $PC+2 \rightarrow PC$ ；否则 $PC+1 \rightarrow PC$
影响状态： 无影响
说明： 对寄存器 R 内容加 1，结果存到 ACC ，结果为 0，则跳过下一条指令
周期数： 1（跳转条件不成立）或 2（跳转条件成立）

JZR

语法： JZR R
操作数： $0 \leq R \leq 127$
操作： $R+1 \rightarrow R$ ；结果为 0，则 $PC+2 \rightarrow PC$ ；否则 $PC+1 \rightarrow PC$
影响状态： 无影响
说明： 对寄存器 R 内容加 1，结果存到 R ，结果为 0，则跳过下一条指令
周期数： 1（跳转条件不成立）或 2（跳转条件成立）

MOVAI

语法： MOVAI K
操作数： $0 \leq K \leq 255$
操作： $K \rightarrow ACC$
影响状态： 不影响
说明： 将立即数 K 存到 ACC

周期数： 1

MOVAR

语法： MOVAR R
操作数： $0 \leq R \leq 127$
操作： $R \rightarrow \text{ACC}$
影响状态： Z
Z R 为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 将寄存器 R 内容存到 ACC
周期数： 1

MOVR

语法： MOVR R
操作数： $0 \leq R \leq 127$
操作： $R \rightarrow \text{ACC}$
影响状态： Z
Z R 为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 将寄存器 R 内容存到 R，会影响 Z 标志位
周期数： 1

MOVRA

语法： MOVRA R
操作数： $0 \leq R \leq 127$
操作： $\text{ACC} \rightarrow R$
影响状态： 不影响
说明： 将 ACC 存到 R
周期数： 1

NOP

语法： NOP
操作数： 无
操作： 空操作
影响状态： 不影响
说明： 空操作
周期数： 1

ORAI

语法： ORAI K
操作数： $0 \leq K \leq 255$
操作： K or ACC→ACC
影响状态： Z
Z 或的结果为 0，则 1→Z；否则 0→Z
说明： 立即数 K 和 ACC 或操作，结果存到 ACC
周期数： 1

ORAR

语法： ORAR R
操作数： $0 \leq R \leq 127$
操作： R or ACC→ACC
影响状态： Z
Z 或的结果为 0，则 1→Z；否则 0→Z
说明： 寄存器 R 内容和 ACC 或操作，结果存到 ACC
周期数： 1

ORRA

语法： ORRA R
操作数： $0 \leq R \leq 127$
操作： R or ACC→R
影响状态： Z
Z 或的结果为 0，则 1→Z；否则 0→Z
说明： 寄存器 R 内容和 ACC 或操作，结果存到 R
周期数： 1

RETAI

语法： RETAI K
操作数： $0 \leq K \leq 255$
操作： TOS(Top of Stack)→PC; K→ACC
影响状态： 无影响
说明： 从子程序返回（从栈顶取回 PC），并将立即数存到 ACC
周期数： 2

RETIE

语法： RETIE
操作数： 无
操作： TOS(Top of Stack)→PC; 1→GIE
影响状态： 无影响
说明： 从中断返回（从栈顶取回 PC），并 GIE 置 1
周期数： 2

RETURN

语法： RETURN
操作数： 无
操作： TOS(Top of Stack)→PC
影响状态： 无影响
说明： 从子程序返回（从栈顶取回 PC）
周期数： 2

RLAR

语法： RLAR R
操作数： $0 \leq R \leq 127$
操作： $C \leftarrow R[7]; ACC[7:1] \leftarrow R[6:0]; ACC[0] \leftarrow C$
影响状态： C
 $C \quad R[7] \rightarrow C$
说明： 寄存器 R 内容带 C 标志左移，结果存到 ACC
周期数： 1

RLR

语法： RLR R
操作数： $0 \leq R \leq 127$
操作： $C \leftarrow R[7]; R[7:1] \leftarrow R[6:0]; R[0] \leftarrow C$
影响状态： C
 $C \quad R[7] \rightarrow C$
说明： 寄存器 R 内容带 C 标志左移，结果存到 R
周期数： 1

RRAR

语法： RRAR R
操作数： $0 \leq R \leq 127$
操作： $C \rightarrow \text{ACC}[7]; R[7:1] \rightarrow \text{ACC}[6:0]; R[0] \rightarrow C$
影响状态： C
 C $R[0] \rightarrow C$
说明： 寄存器 R 内容带 C 标志右移，结果存到 ACC
周期数： 1

RRR

语法： RRR R
操作数： $0 \leq R \leq 127$
操作： $C \rightarrow R[7]; R[7:1] \rightarrow R[6:0]; R[0] \rightarrow C$
影响状态： C
 C $R[0] \rightarrow C$
说明： 寄存器 R 内容带 C 标志右移，结果存到 R
周期数： 1

RSBCAR

语法： RSBCAR R
操作数： $0 \leq R \leq 127$
操作： $R - \text{ACC} - /C \rightarrow \text{ACC}$
影响状态： C, DC, Z
 C 相减结果产生借位，则 $0 \rightarrow C$ ；否则 $1 \rightarrow C$
 DC 相减结果第 3 位向第 4 位产生借位，则 $0 \rightarrow DC$ ；否则 $1 \rightarrow DC$
 Z 相减结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 带 C 标志的寄存器 R 内容与 ACC 减法，结果存到 ACC
周期数： 1

RSBCRA

语法： RSBCRA R
操作数： $0 \leq R \leq 127$
操作： $R - \text{ACC} - /C \rightarrow R$
影响状态： C, DC, Z
 C 相减结果产生借位，则 $0 \rightarrow C$ ；否则 $1 \rightarrow C$

DC 相减结果第 3 位向第 4 位产生借位, 则 0→DC ; 否则 1→DC

Z 相减结果为 0, 则 1→Z ; 否则 0→Z

说明: 带 C 标志的寄存器 R 内容与 ACC 减法, 结果存到 R

周期数: 1

RSUBAR

语法: RSUBAR R

操作数: $0 \leq R \leq 127$

操作: R-ACC→ACC

影响状态: C,DC,Z

C 相减结果产生借位, 则 0→C ; 否则 1→C

DC 相减结果第 3 位向第 4 位产生借位, 则 0→DC ; 否则 1→DC

Z 相减结果为 0, 则 1→Z ; 否则 0→Z

说明: 寄存器 R 内容和 ACC 相减, 结果存到 ACC

周期数: 1

RSUBRA

语法: RSUBRA R

操作数: $0 \leq R \leq 127$

操作: R-ACC→R

影响状态: C,DC,Z

C 相减结果产生借位, 则 0→C ; 否则 1→C

DC 相减结果第 3 位向第 4 位产生借位, 则 0→DC ; 否则 1→DC

Z 相减结果为 0, 则 1→Z ; 否则 0→Z

说明: 寄存器 R 内容和 ACC 相减, 结果存到 R

周期数: 1

STOP

语法: STOP

操作数: 无

操作: 0→WDT; 进入休眠模式

影响状态: TO,PD

TO 1→TO

PD 0→PD

说明: 进入休眠模式, 对看门狗定时器 (WDT) 清零

周期数: 1

SWAPAR

语法： SWAPAR R
操作数： $0 \leq R \leq 127$
操作： $R[7:4] \rightarrow \text{ACC}[3:0]; R[3:0] \rightarrow \text{ACC}[7:4]$
影响状态： 不影响
说明： 交换寄存器 R 内容的高 4 位和低 4 位，结果存到 ACC
周期数： 1

SWAPR

语法： SWAPR R
操作数： $0 \leq R \leq 127$
操作： $R[7:4] \rightarrow R[3:0]; R[3:0] \rightarrow R[7:4]$
影响状态： 不影响
说明： 交换寄存器 R 内容的高 4 位和低 4 位，结果存到 R
周期数： 1

XORAI

语法： XORAK K
操作数： $0 \leq K \leq 255$
操作： $K \text{ xor } \text{ACC} \rightarrow \text{ACC}$
影响状态： Z
 Z 异或的结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 立即数 K 和 ACC 异或操作，结果存到 ACC
周期数： 1

XORAR

语法： XORAR R
操作数： $0 \leq R \leq 127$
操作： $R \text{ xor } \text{ACC} \rightarrow \text{ACC}$
影响状态： Z
 Z 或的结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 寄存器 R 内容和 ACC 异或操作，结果存到 ACC
周期数： 1

XORRA

语法： XORRA R
操作数： $0 \leq R \leq 127$
操作： R xor ACC → R
影响状态： Z
 Z 或的结果为 0，则 1 → Z；否则 0 → Z
说明： 寄存器 R 内容和 ACC 异或操作，结果存到 R
周期数： 1

10 电气参数

10.1 极限参数

参数	符号	值	单位
工作电压	VDD	-0.3 ~ 6.5	V
输入电压	VIN	VSS-0.3 ~ VDD+0.3	V
工作温度	TA	-40 ~ 85	°C
储存温度	Tstg	-65 ~ 150	°C
焊接温度	WT	260	°C
焊接时间	WL	10	S

10.2 直流电气参数

VDD=5V, T=25°C

特性	符号	引脚	条件	最小	典型	最大	单位
工作电压	VDD		400K-4M	2.0		5.5	V
			400K-8M	2.7		5.5	
输入漏电	V_{leak}	所有输入脚	VIN=VDD,0			±1	uA
输入高电平	V_{ih}	所有输入脚		0.7VDD		VDD	V
输入低电平	V_{il}	所有输入脚		0		0.15VDD	V
上拉电阻	R_{pu}	P10-P12 P14-P17	VIN=0	50	80	110	Kohm
下拉电阻	R_{pd}	P00-P03 P10-P12	VIN=VDD	100	125	150	Kohm
输出高电平 驱动电流	I_{oh}	所有输出脚	$V_{oh}=VDD-0.7V$	6			mA
输出低电平 驱动电流	I_{ol}	所有输出脚	$V_{ol}=0.7V$	10			mA
静态功耗	I_{ds}	VDD	关 LVR,关 WDT		0.5	1	uA
			开 WDT		3	10	
			开 LVR		0.5	1	

			开 RTC		10	20	
动态功耗	I_{dd}	VDD	VDD=5V Fosc=4MHz 无负载			3	mA

10.3 交流电气参数

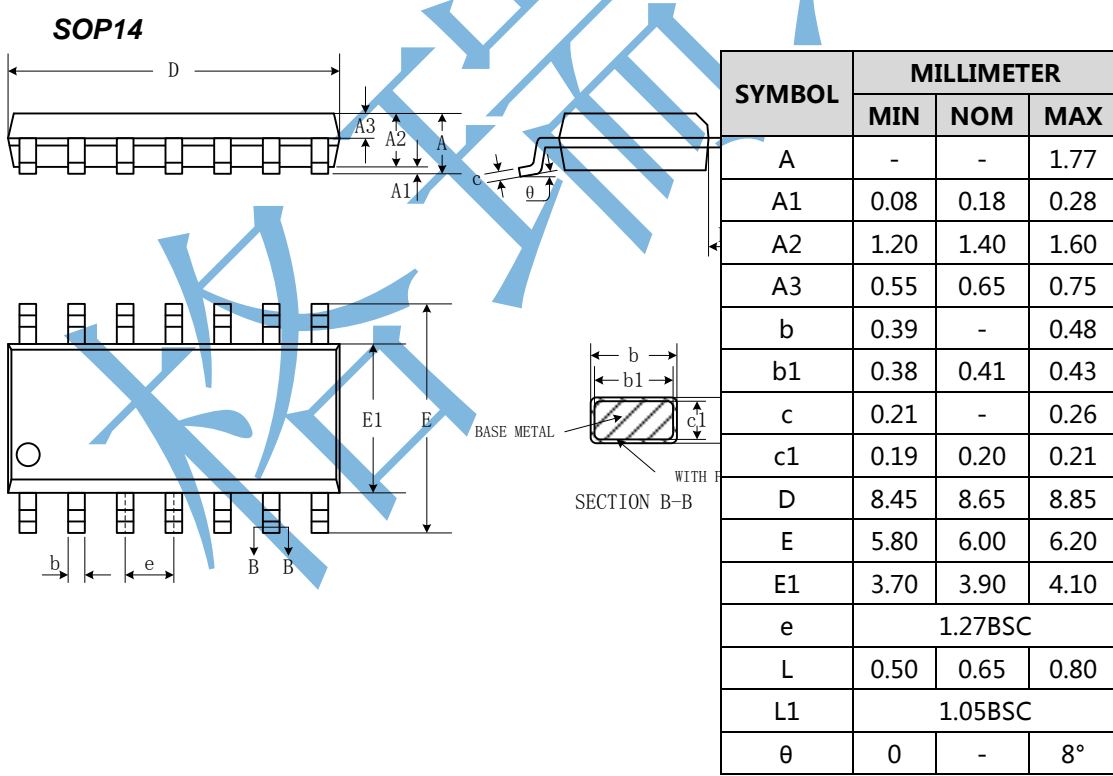
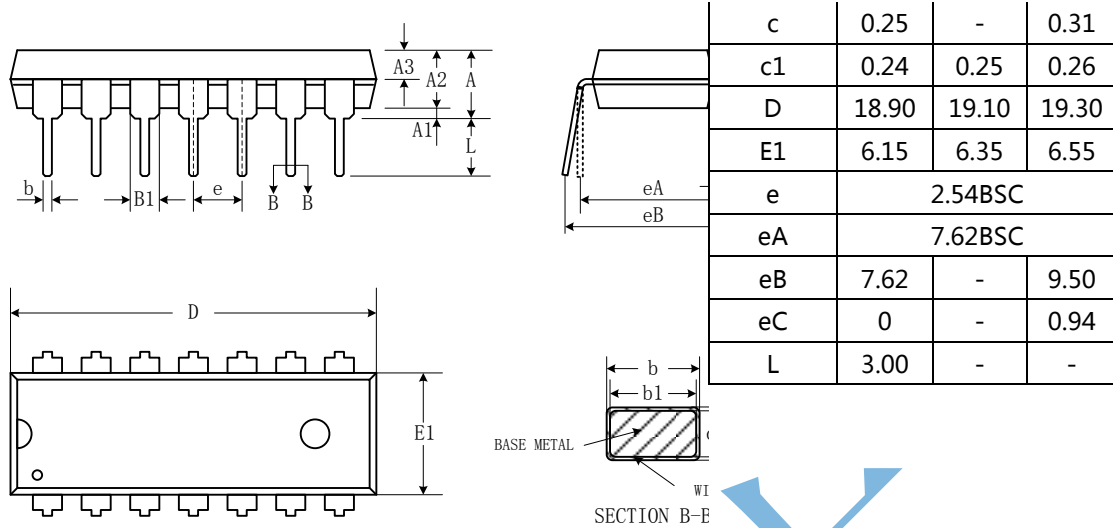
VDD=5V, T=25°C

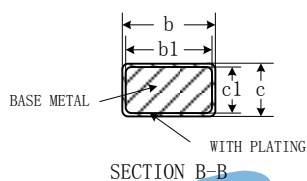
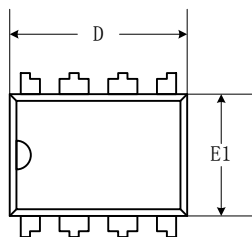
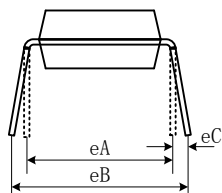
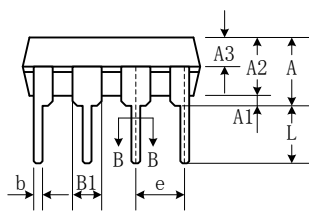
特性	符号	条件	最小	典型	最大	单位
外部晶振频率	F_{osc}		400K		8M	Hz
内部高频 RC 振荡频率	F_{hrc1}	T=25°C VDD=5V	-1%	2	+1%	MHz
			-1%	4	+1%	
			-1%	8	+1%	
	F_{hrc2}	T=-40°C~85°C VDD=5V	-2%	2	+2%	MHz
			-2%	4	+2%	
			-2%	8	+2%	
	F_{hrc3}	T=25°C VDD=2~5.5V	-1.5%	2	+1.5%	MHz
			-1.5%	4	+1.5%	
-1.5%			8	+1.5%		
F_{hrc4}	T=-40°C~85°C VDD=2~5.5V	-2.5%	2	+2.5%	MHz	
		-2.5%	4	+2.5%		
		-2.5%	8	+2.5%		
WDT 振荡器频率	F_{wdt}	T=25°C VDD=5V	-50%	28	+50%	KHz
振荡器起振时间	T_{oxov}				20	ms

11 封装外形尺寸

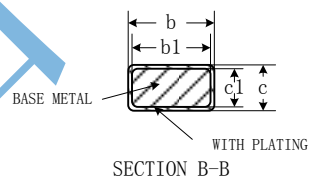
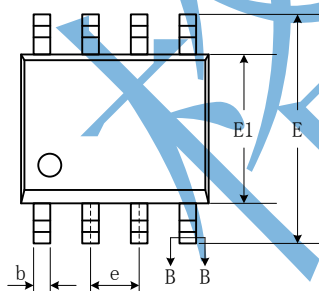
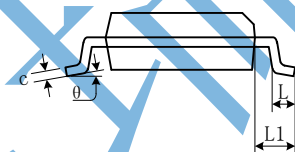
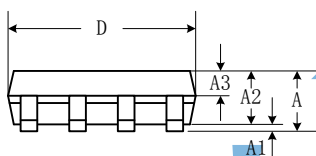
DIP14

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	3.60	3.80	4.00
A1	0.51	-	-
A2	3.10	3.30	3.50
A3	1.42	1.52	1.62
b	0.44	-	0.53
b1	0.43	0.46	0.48
B1	1.52BSC		

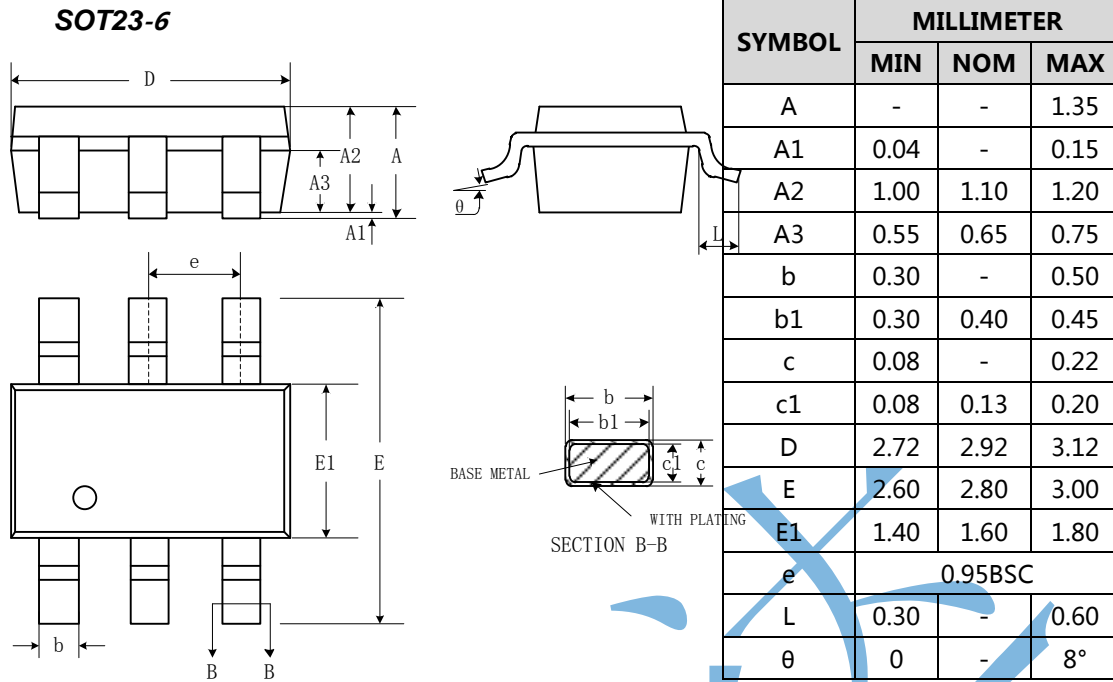


DIP8


SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	3.60	3.80	4.00
A1	0.51	-	-
A2	3.10	3.30	3.50
A3	1.50	1.60	1.70
b	0.44	-	0.53
b1	0.43	0.46	0.48
B1	1.52BSC		
c	0.25	-	0.31
c1	0.24	0.25	0.26
D	9.05	9.25	9.45
E1	6.15	6.35	6.55
e	2.54BSC		
eA	7.62BSC		
eB	7.62	-	9.50
eC	0	-	0.94
L	3.00	-	-

SOP8


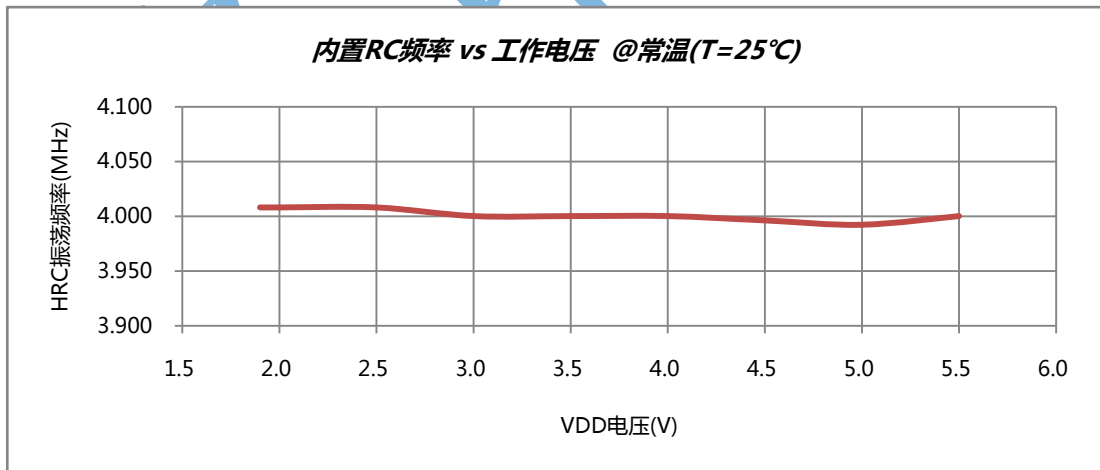
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	-	-	1.77
A1	0.08	0.18	0.28
A2	1.20	1.40	1.60
A3	0.55	0.65	0.75
b	0.39	-	0.48
b1	0.38	0.41	0.43
c	0.21	-	0.26
c1	0.19	0.20	0.21
D	4.70	4.90	5.10
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27BSC		
L	0.50	0.65	0.80
L1	1.05BSC		
theta	0	-	8°

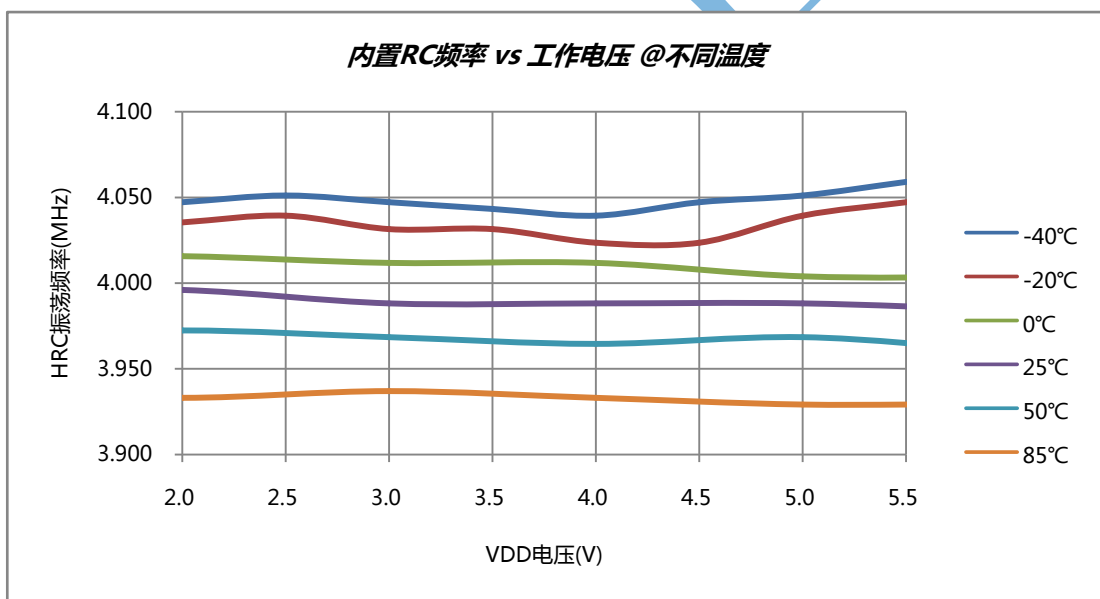
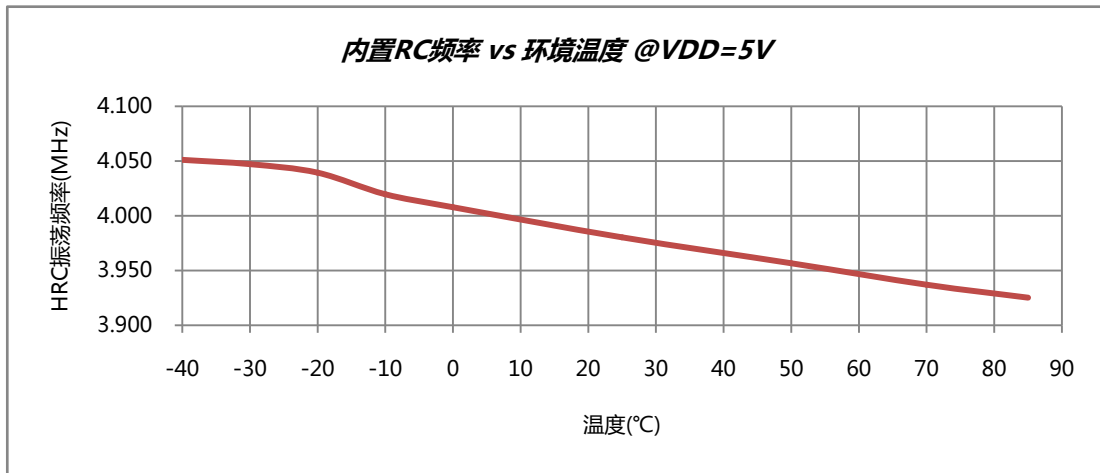


12 附录

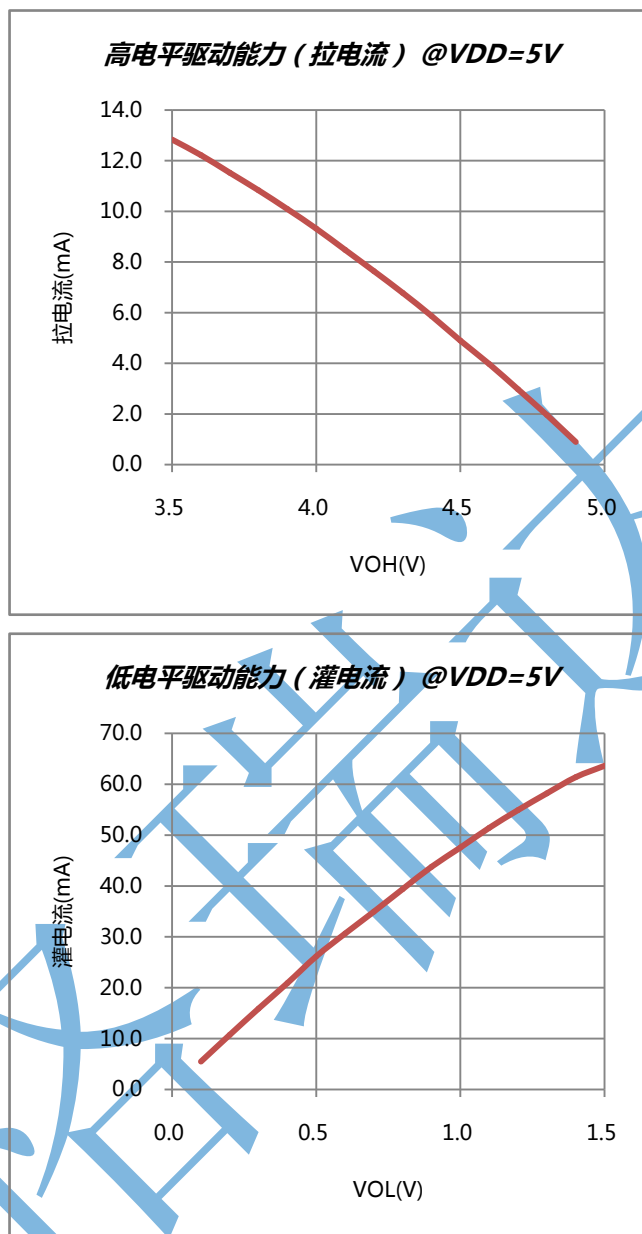
注：附录的内容仅供参考。

12.1 内置 RC 频率曲线



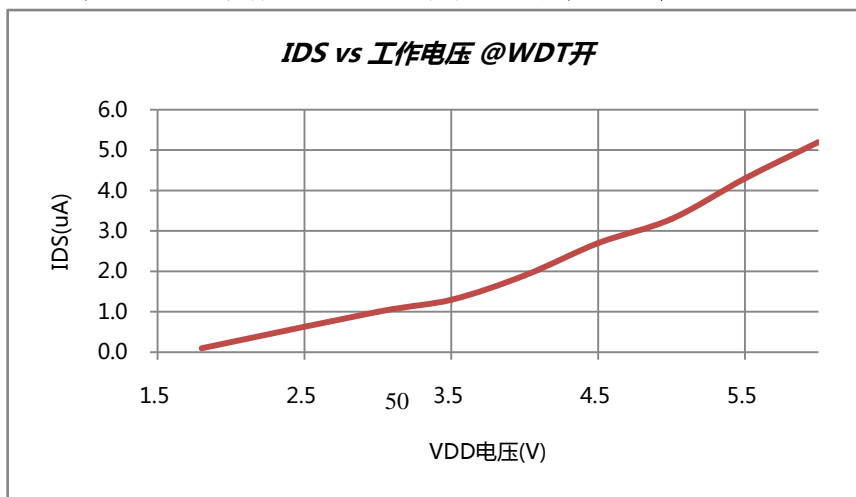


12.2 IO 口驱动能力曲线

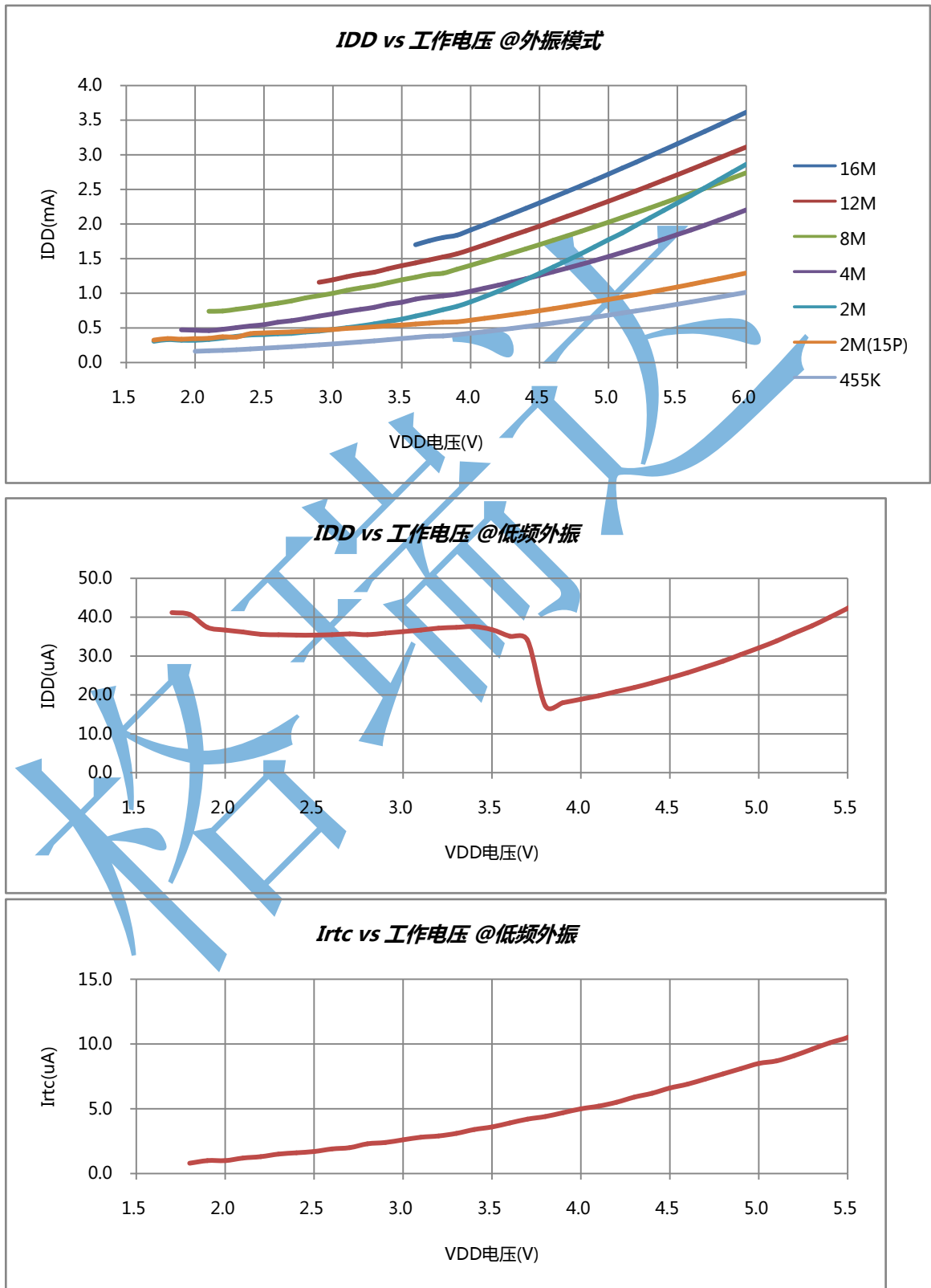


12.3 STOP 模式工作电流曲线

条件：开 WDT，用 WDT 中断约 2.3s 周期性唤醒一次，测试平均电流



12.4 动态工作电流曲线



13 订购信息

下单规格	功能简述	芯片型号	封装
30P011P08	通用型单片机	30P011	DIP8
30P011S08	通用型单片机	30P011	SOP8
30P011P14	通用型单片机	30P011	DIP14
30P011S14	通用型单片机	30P011	SOP14
30P011S06	通用型单片机	30P011	SOT23-6

深圳市格瑞达实业有限公司（总公司）

SHENZHEN GREENMCU TECHNOLOGY CO.,LTD.

地址：深圳市福田区彩田南路海鹰大厦 20B

电话：(86) 755-83051793 82913392

(86) 755-82914749 82913502

传真：(86) 755-82971356

网址：www.greenmcu.com

深圳市格瑞达实业有限公司（顺德办事处）

地址：顺德区容桂镇文海西路保利百合花园 10 栋 B 单元 1901

电话：(86) 757-28302691 22909432

传真：(86) 757-28302691

最新信息请登陆我们的网址：www.greenmcu.com