



8-Bit 通用 AD 型单片机

32P21

格瑞达

版本: V00 日期: 2013/07/20

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	4
2	中央处理器	5
2.1	指令集	5
2.2	程序存储器 (OTP)	5
2.3	数据存储器	5
2.4	堆栈	7
2.5	烧录配置选项 OPBIT	7
2.6	控制寄存器	8
3	系统时钟	13
3.1	外接晶体振荡器	13
3.2	内置高精度 RC 振荡器	13
3.3	内置低速 RC 振荡器	14
3.4	工作模式	14
3.5	低功耗模式	16
4	复位	17
4.1	复位条件	17
4.2	上电复位	17
4.3	外部复位	18
4.4	掉电复位	18
4.5	WDT 看门狗复位	18
5	I/O 口	19
5.1	I/O 工作模式	19
5.2	上拉电阻控制	20
5.3	端口模式控制	20
6	定时器	21
6.1	看门狗 (WDT)	21
6.2	定时器 T0	21
6.3	定时器 T1	23
7	模数转换器 (ADC)	25
7.1	ADC 功能介绍	25
7.2	ADC 转换时序图	25
7.3	ADC 操作步骤	25
7.4	ADC 相关寄存器	26
8	低电压检测 (LVD)	28
9	中断	29
9.1	外中断	29
9.2	键盘中断	29
9.3	定时器中断	29

9.4	ADC 转换中断	29
9.5	中断相关寄存器	30
10	指令集概述	32
10.1	指令汇总表	33
10.2	指令详细说明	36
10.2.1	ADCAI	36
10.2.2	ADCAR	36
10.2.3	ADCRA	36
10.2.4	ADDAI	37
10.2.5	ADDAR	37
10.2.6	ADDRA	37
10.2.7	ANDAI	37
10.2.8	ANDAR	37
10.2.9	ANDRA	38
10.2.10	ASBCAI	38
10.2.11	ASBCAR	38
10.2.12	ASBCRA	39
10.2.13	ASUBAI	39
10.2.14	ASUBAR	39
10.2.15	ASUBRA	39
10.2.16	BCLR	40
10.2.17	BSET	40
10.2.18	CALL	40
10.2.19	CLRA	40
10.2.20	CLRR	40
10.2.21	CLRWDT	40
10.2.22	COMAR	41
10.2.23	COMR	41
10.2.24	DAA	41
10.2.25	DECA	42
10.2.26	DECAR	42
10.2.27	DECR	42
10.2.28	DJZA	42
10.2.29	DJZAR	42
10.2.30	DJZR	43
10.2.31	DSA	43
10.2.32	GOTO	43
10.2.33	INCA	43
10.2.34	INCAR	43
10.2.35	INCR	43
10.2.36	ISBCAI	44
10.2.37	ISUBAI	44
10.2.38	JBCLR	44
10.2.39	JBSET	45

10.2.40 JZA	45
10.2.41 JZAR.....	45
10.2.42 JZR	45
10.2.43 LCALL	45
10.2.44 LGOTO.....	46
10.2.45 MOVAI	46
10.2.46 MOVAR.....	46
10.2.47 MOVR	46
10.2.48 MOVRA	46
10.2.49 MULAR.....	46
10.2.50 MULRA.....	47
10.2.51 NOP	47
10.2.52 ORAI	47
10.2.53 ORAR	47
10.2.54 ORRA	47
10.2.55 RETAI.....	48
10.2.56 RETIE	48
10.2.57 RETURN	48
10.2.58 RLA	48
10.2.59 RLAR	48
10.2.60 RLR	49
10.2.61 RRA	49
10.2.62 RRAR	49
10.2.63 RRR	50
10.2.64 RSBCAR	50
10.2.65 RSBCRA	50
10.2.66 RSUBAR	50
10.2.67 RSUBRA	50
10.2.68 STOP	51
10.2.69 SWAPAR	51
10.2.70 SWAPR	51
10.2.71 XORAI	52
10.2.72 XORAR	52
10.2.73 XORRA	52
11 电气参数	53
11.1 极限参数	53
11.2 直流特性参数	53
11.3 ADC 特性参数	54
11.4 交流电气参数	55
11.5 特性曲线	56
12 封装外形尺寸	61
12.1 DIP14.....	61
12.2 SOP14.....	61
12.3 DIP8.....	62

12.4	SOP8	62
13	订购信息	63

格瑞达

第一章 简述

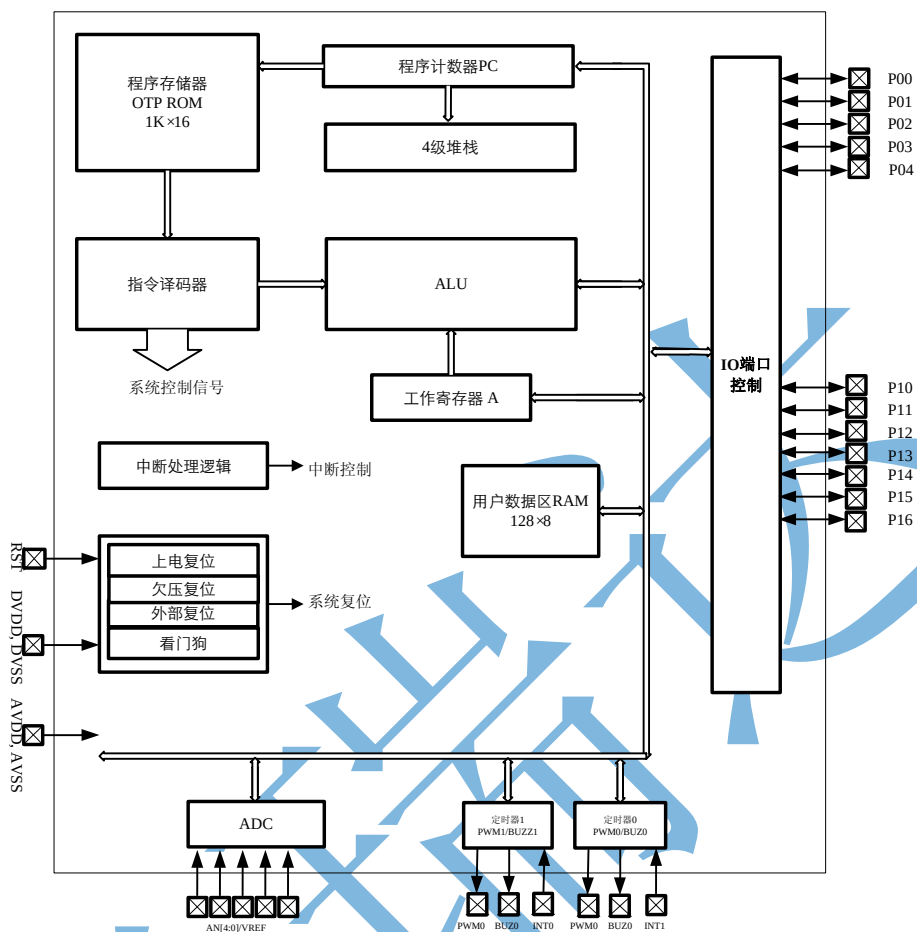
1 产品简介

32P21 是一款高性能 8 位 IO+AD 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。

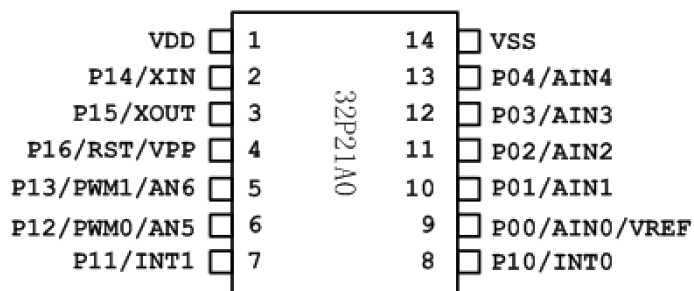
1.1 产品特性

- ◇ 8 位 CPU 内核
 - ✓ 精简指令集
 - ✓ 高频模式下 2T/4T/8T/16T 可设；低频工作模式下为 4T
- ◇ 存储器
 - ✓ 1K*16 程序存储器空间, 4 级深度硬件堆栈（通过 INDE3 可读取 ROM 区内容）
 - ✓ 128 字节通用数据寄存器空间
- ◇ 2 组 I/O 口
 - ✓ 5 位 P0 端口，与 AD 模拟输入复用
 - ✓ 7 位 P1 端口，其中 P16 为开漏 I/O，P12、P13 与模拟输入复用
- ◇ 支持低功耗工作模式（休眠功耗小于 1uA，HOLD 模式小于 10uA）
- ◇ 定时器
 - ✓ 内部自振式看门狗计数器（WDT）
 - ✓ 2 个带有 PWM、BUZ 功能 8 位定时器（PWM 最小可调制脉宽宽度 62.5ns）
- ◇ 12 位模数转换器
 - ✓ 7 路外部模拟信号源输入（AN0-AN6）
 - ✓ 参考电压可选用外接或内置高精度参考电压（VDD、4V、3V、2V 可选）
- ◇ 中断
 - ✓ 两路外部中断源（INT0、INT1）
 - ✓ 7 路键盘中断
 - ✓ 定时器 0 中断
 - ✓ 定时器 1 中断
 - ✓ AD 转换中断
- ◇ 双时钟振荡模式
 - ✓ 外接高频晶体振荡器 + 内嵌低频振荡器模式（28K）
 - ✓ 内嵌高频振荡器（16M）+ 外接低频振荡器模式
 - ✓ 内嵌高频振荡器（16M）+ 内嵌低频振荡器（28K）
- ◇ 低电压复位 LVR（多级复位电压可选）
- ◇ 低电压检测 LVD（多级检测电压可选）
- ◇ 工作电压
 - ✓ 3.6V-5.5V @F_{cpu}=8MHz（16MHz/2T）
 - ✓ 2.4V-5.5V @F_{cpu}=4MHz（16MHz/4T）
 - ✓ 1.8V-5.5V @F_{cpu}=1MHz（16MHz/16T）
- ◇ 封装形式：
 - ✓ DIP14、SOP14、DIP8、SOP8

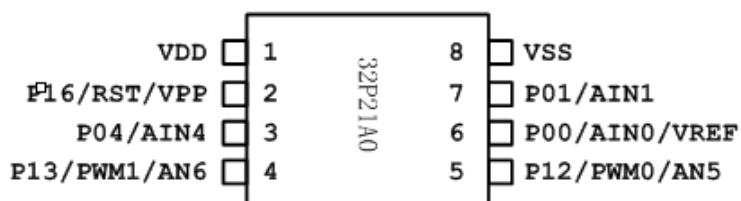
1.2 系统框图



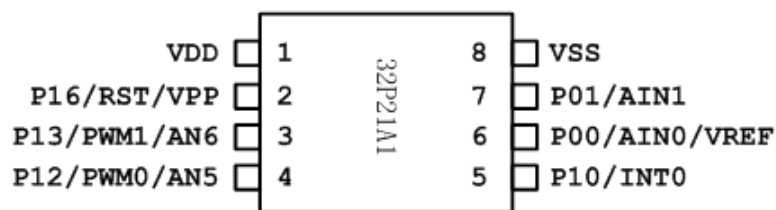
1.3 引脚排列



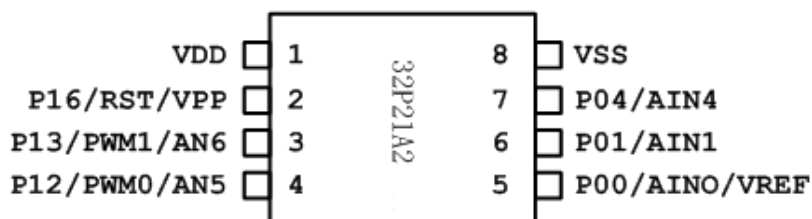
DIP14/SOP14



DIP8/SOP8



DIP8/SOP8



DIP8/SOP8

1.4 引脚说明

引脚名	I/O	描述
VDD	–	电源正端
VSS	–	电源负端
P10/INT0	I/O	GPIO/上拉电阻；外部中断0输入
P11/INT1	I/O	GPIO/上拉电阻；外部中断1输入
P12/PWM0/AN5	I/O	GPIO/上拉电阻；PWM0/BUZ0；模数转换器通道5
P13/PWM1/AN6	I/O	GPIO/上拉电阻；PWM1/BUZ1；模数转换器通道6
P14/RST/VPP	I/O	GPIO/开漏输出/上拉电阻；复位输入/上拉电阻；编程高压输入
P15/XOUT	I/O	GPIO/上拉电阻；外部晶振输出
P16/XOUT	I/O	GPIO/上拉电阻；外部晶振输入
P00/AN0/VREF	I/O	GPIO/上拉电阻；模数转换器通道0；模数转换器参考电压输入
P11-P14/AN1-AN4	I/O	GPIO/上拉电阻；模数转换器通道1-4

第二章 中央处理器

2 中央处理器

2.1 指令集

32P21 的指令是精简指令集。指令集详细资料见本公司手册《MC32 指令集说明书》。

2.2 程序存储器（OTP）

1K*16BIT 的程序存储器空间，程序存储器空间（0000H - 03FFH）可通过 INDF3 间接访问

复位向量（0000H）
通用程序区（0001H - 0007H）
中断向量（0008H）
通用程序区（0009H - 03FFH）
厂商保留区（0400H - 7FFFH）
OPBIT0（8000H）
OPBIT1（8001H）

例：通过 INDF3 访问 $FSR1*256+FSR0$ 指向的程序存储器中内容，高 8 位存放在数据寄存器区 11H，低 8 位存放在数据寄存器区 10H

```

MOVAI    55H
MOVRA    FSR0    ; 将 55H 写入 FSR0
MOVAI    01H
MOVRA    FSR1    ; 将 01H 写入 FSR1
MOVRA    INDF3   ; 读取  $FSR1*256+FSR0$  指向（0155H）程序存储器
                ; 的内容，其中高 8 位放在 HIBYTE 寄存器，低 8
                ; 位放在 A 寄存器
MOVRA    10H     ; 低 8 位放到数据寄存器 10H 地址
MOVRA    HIBYTE  ; 从 HIBYTE 读取高 8 位
MOVRA    11H     ; 高 8 位放到数据寄存器 11H 地址
    
```

2.3 数据存储器

数据寄存器分为三个区，快速通用寄存器区GPR（128Byte空间）和特殊功能寄存器区SFR，扩展寄存器区（本产品未用）必须使用间接寻址模式2进行寻址，具体地址分配参照下表。

数据存储区地址映射表:

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
数据存储区，直接寻址，INDF0，INDF1，INDF2间接寻址								
000H - 07FH	通用数据区							
080H - 17FH	保留							
180H - 187H	INDF0	INDF1	INDF2	HIBYTE	FSR0	FSR1	PCL	PFLAG
188H - 18FH	MCR	INDF3	INTE0	INTF0				
190H - 197H	IOP0	OEP0	PUP0	ANSEL	IOP1	OEP1	PUP1	KBCR
198H - 19FH								
1A0H - 1A7H	TOCR	TOCNT	TOLOAD	TODATA	T1CR	T1CNT	T1LOAD	T1DATA
1A8H - 1AFH						LVDCR	OSCM	
1B0H - 1B7H	ADCR0	ADCR1			ADRH	ADRL		
1B8H - 1F7H								
1F8H - 1FFH				OSCCAL				
0200H-FFFFH	映射到000H - 01FFH地址							

注：上表中灰色部分数据存储区地址未用，读出数据为 0

数据寄存器地址组成

HSB	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LSB
0	0	0	0	0	0	0	0	来自指令的 9 位地址									直接寻址模式
0	0	0	0	0	0	0	0	0	FSR0								间接寻址模式 0
0	0	0	0	0	0	0	0	1	FSR1								间接寻址模式 1
FSR1									FSR0								间接寻址模式 2

直接寻址模式：以指令的低9位作为数据存储器地址

例：通过直接寻址模式把 55H 数据写入 10H 地址，然后对 10H 地址数据加 1

```

MOVAI      55H
MOVRA      10H      ; 把数据 55H 写入 10H 地址数据存储器中
INCR       10H      ; 10H 地址内容加 1
    
```

间接寻址模式0：当访问INDF0时，FSR0作为数据存储器地址

例：通过间接寻址模式 0 把 55H 数据写入 10H 地址

```

MOVAI      10H
MOVRA      FSR0
MOVAI      55H
MOVRA      INDF0    ; 把数据 55H 写入 FSR0 指向数据存储器中
    
```

间接寻址模式1：当访问INDF1时，FSR1作为数据存储器地址

例：通过间接寻址模式 1 把 55H 数据写入 10H 地址

```

MOVAI      10H
    
```

```
MOVRA    FSR1
MOVAI    55H
MOVRA    INDF1    ; 把数据 55H 写入 FSR1 指向数据存储器中
```

间接寻址模式2：当访问INDF2时，FSR1*256+FSR0作为数据存储器地址

例：通过间接寻址模式 2 把 55H 数据写入 0210H 地址数据存储器

```
MOVAI    10H
MOVRA    FSR0
MOVAI    02H
MOVRA    FSR1
MOVAI    55H
MOVRA    INDF2    ; 把数据 55H 写入 FSR1*256+FSR0 指向数据存储器中
```

注：FSR1 高 7 位无效，0210H 地址已被自动映射到 10H

2.4 堆栈

4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 烧录配置选项 OPBIT

用户配置字简称OPBIT是OTP中的2个特殊字，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。32P21的OPBIT定义如下。

OPBIT0:

位	符号	功能说明
BIT[1:0]	WDTC	WDT 工作模式控制位 00: 始终关闭看门狗 01: 休眠模式下关闭看门狗 1X: 始终开启看门狗
BIT13、BIT[3:2]	WDTT	WDT 溢出时间选择位 000: 上电延时=WDT 溢出时间=4mS 001: 上电延时=WDT 溢出时间=16mS 010: 上电延时=WDT 溢出时间=64mS 011: 上电延时=WDT 溢出时间=256mS 100: 上电延时=4mS, WDT 溢出时间=512mS 101: 上电延时=16mS, WDT 溢出时间=1024mS 110: 上电延时=64mS, WDT 溢出时间=2048mS 111: 上电延时=256mS, WDT 溢出时间=4096mS
BIT[6:4]	FCPU	高速模式 CPU 速度选择 000: 指令周期为 2 个高速时钟周期 001: 指令周期为 4 个高速时钟周期 010: 指令周期为 8 个高速时钟周期

		011: 指令周期为 16 个高速时钟周期 100: 指令周期为 32 个高速时钟周期 101: 指令周期为 64 个高速时钟周期 110: 指令周期为 128 个高速时钟周期 111: 指令周期为 256 个高速时钟周期
BIT[7]	MCLRE	外部复位使能位 1: 使能外部复位, P16 作为复位引脚 0: 不使能外部复位, P16 作为 I/O
BIT[9:8]	MOSC	振荡器模式选择位 00: 内部高频 RC 振荡器+内部低频 RC 振荡器 01: 内部高频 RC 振荡器+外部低频晶体振荡器 10: 外部高频晶体振荡器+内部低频 RC 振荡器 11: 保留
BIT[12:10]	VLVRS	系统复位电压选择位 000: LVR 电压=1.60V (参考值) 001: LVR 电压=1.85V (参考值) 010: LVR 电压=2.05V (参考值) 011: LVR 电压=2.18V (参考值) 100: LVR 电压=2.32V (参考值) 101: LVR 电压=2.45V (参考值) 110: LVR 电压=3.05V (参考值) 111: LVR 电压=3.60V (参考值)
BIT[14]	未用	值为 1
BIT[15]	ENCR	代码加密选项 1: 不使能代码加密 0: 使能代码加密

OPBIT1:

位	符号	功能说明
BIT[7:0]	OSCCAL	内部 16M 高频振荡器频率调校位, 产品出厂前厂家已写入
BIT[14:8]	VREFCAL	ADC 内建参考调校位, 产品出厂前厂家已写入
BIT[16:15]	未用	值为 1

2.6 控制寄存器

32P21 全部控制寄存器列在下表中, 具体功能详见各功能模块的说明。

地址	助记符	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	初始值
180H	INDF0	INDF07	INDF06	INDF05	INDF04	INDF03	INDF02	INDF01	INDF00	XXXX XXXX
181H	INDF1	INDF17	INDF16	INDF15	INDF14	INDF13	INDF12	INDF11	INDF10	XXXX XXXX
182H	INDF2	INDF27	INDF26	INDF25	INDF24	INDF23	INDF22	INDF21	INDF20	XXXX XXXX
183H	HIBYTE	HIBYTE7	HIBYTE6	HIBYTE5	HIBYTE4	HIBYTE3	HIBYTE2	HIBYTE1	HIBYTE0	XXXX XXXX
184H	FSR0	FSR07	FSR06	FSR05	FSR04	FSR03	FSR02	FSR01	FSR00	XXXX XXXX
185H	FSR1	FSR17	FSR16	FSR15	FSR14	FSR13	FSR12	FSR11	FSR10	XXXX XXXX
186H	PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0000 0000
187H	PFLAG	-	-	-	-	-	Z	DC	C	---- -XXX

188H	MCR	GIE	-	TO	PD	MINT11	MINT10	MINT01	MINT00	0-00 0000
189H	INDF3	INDF37	INDF36	INDF35	INDF34	INDF33	INDF32	INDF31	INDF30	XXXX XXXX
18AH	INTE	-	ADIE	-	KBIE	INT1IE	INT0IE	T1IE	T0IE	-0-0 0000
18BH	INTF	-	ADIF	-	KBIF	INT1IF	INT0IF	T1IF	T0IF	-0-X XX00
190H	IOP0	-	-	-	P04D	P03D	P02D	P01D	P00D	---X XXXX
191H	OEP0	-	-	-	P04OE	P03OE	P02OE	P01OE	P00OE	---0 0000
192H	PUP0	-	-	-	P04PU	P03PU	P02PU	P01PU	P00PU	---0 0000
193H	ANSEL	-	P13ANS	P12ANS	P04ANS	P03ANS	P02ANS	P01ANS	P00ANS	-000 0000
194H	IOP1	-	P16D	P15D	P14D	P13D	P12D	P11D	P10D	-XXX XXXX
195H	OEP1	-	P16OE	P15OE	P14OE	P13OE	P12OE	P11OE	P10OE	-000 0000
196H	PUP1	-	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU	-000 0000
197H	KBIM	-	KBIM6	KBIM5	KBIM4	KBIM3	KBIM2	KBIM1	KBIM0	-000 0000
1A0H	TOCR	TC0EN	PWM0OE	BUZ0OE	T0PS1	T0PS0	T0PR2	T0PR1	T0PR0	0000 0000
1A1H	TOCNT	TC07	TC06	TC05	TC04	TC03	TC02	TC01	TC00	1111 1111
1A2H	TOLOAD	TOLOAD7	TOLOAD6	TOLOAD5	TOLOAD4	TOLOAD3	TOLOAD2	TOLOAD1	TOLOAD0	1111 1111
1A3H	TODATA	TODATA7	TODATA6	TODATA5	TODATA4	TODATA3	TODATA2	TODATA1	TODATA0	0000 0000
1A4H	T1CR	TC1EN	PWM1OE	BUZ1OE	T1PS1	T1PS0	T1PR2	T1PR1	T1PR0	0000 0000
1A5H	T1CNT	T1C7	T1C6	T1C5	T1C4	T1C3	T1C2	T1C1	T1C0	1111 1111
1A6H	T1LOAD	T1LOAD7	T1LOAD6	T1LOAD5	T1LOAD4	T1LOAD3	T1LOAD2	T1LOAD1	T1LOAD0	1111 1111
1A7H	T1DATA	T1DATA7	T1DATA6	T1DATA5	T1DATA4	T1DATA3	T1DATA2	T1DATA1	T1DATA0	0000 0000
1ADH	LVDCR	LV DEN	LVDS2	LVDS1	LVDS0	-	-	-	LVDF	0000 ---0
1AEH	OSCM	-	-	STBL	STBH	-	CLKS	LFEN	HFEN	--X1 -000
1B0H	ADCR0	ADCHS3	ADCHS2	ADCHS1	ADCHS0	ADCKS1	ADCKS0	ADEOC	ADON	0000 0000
1B1H	ADCR1	-	-	-	-	-	VRS2	VRS1	VRS0	---- -000
1B4H	ADRH	ADR11	ADR10	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	XXXX XXXX
1B5H	ADRL	-	-	-	-	ADR3	ADR2	ADR1	ADRO	---- XXXX
1FBH	OSCCAL	OSCCAL7	OSCCAL6	OSCCAL5	OSCCAL4	OSCCAL3	OSCCAL2	OSCCAL1	OSCCAL0	XXXX XXXX

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] INDF0n - 间接寻址寄存器 0

INDF0: INDF0 不是物理寄存器, 对 INDF0 寻址时间上是对 FSR0 指向的数据存储器地址进行访问, 从而实现间接寻址模式。

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] INDF1n - 间接寻址寄存器 1

INDF1: INDF1 不是物理寄存器, 对 INDF1 的寻址时间上是对 FSR1+256 指向的数据存储器地址进行访问, 从而实现间接寻址模式。

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] **INDF2n** - 间接寻址寄存器 2

INDF2: INDF2 不是物理寄存器, 对 INDF2 的寻址时间上是对 FSR1*256+FSR0 指向的数据存储器地址进行访问, 从而实现间接寻址模式。

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] **INDF3n** - 间接寻址寄存器 3

INDF3: INDF3 不是物理寄存器, 对 INDF3 的寻址时间上是对 FSR1*256+FSR0 指向的程序存储器地址进行访问, 从而实现间接寻址模式。

注: 对 INDF3 仅可进行使用读取指令 (MOVAR INDF3) 进行读取访问, 读取内容高 8 位存放在 HIBYTE, 低 8 位存放在 A 寄存器

字操作高8位缓冲器

	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] **HIBYTEn** - 字操作高字节缓冲器

HIBYTE: 对 INDF3 读取操作, 用于存放 FSR1*256+FSR0 指向的程序存储器内容高 8 位数据。

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] **FSR0n** - 数据指针寄存器 0

FSR0: 间接寻址模式 0 指针或间接寻址模式 2、3 指针低 8 位。

数据指针寄存器1

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

BIT[7:0] **FSR1n** - 数据指针寄存器 1

FSR1: 间接寻址模式 1 指针或间接寻址模式 2、3 指针高位。

程序指针计数器低位

	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] **PCn** - 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: $PC=PC+1$

分支指令 GOTO/CALL: $PC=$ 指令码低 10 位

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

对 PCL 操作指令: $PC = (PC[9:0] + A[7:0])$ (对 PCL 操作的加法指令)

$PC = \{PC[9:8], ALU[7:0] (ALU \text{ 运算结果})\}$ (对 PCL 操作的其它指令)

CPU 状态寄存器

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

BIT[7:3] 未用

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

杂用寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCR	GIE	-	TO	PD	MINT11	MINT10	MINT01	MINT01
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** - 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[6] 未用

BIT[5] **TO** - 看门狗溢出标志

0: 上电复位, 执行 CLRWDI 或 STOP 指令

1: 发生 WDT 溢出

BIT[4] **PD** - 进入低功耗休眠模式标志

0: 上电复位, 执行 CLRWDI

1: 执行STOP指令

BIT[3:2] **MINT1** - 外部中断 1 模式寄存器

00: INT1 上升沿中断

01: INT1 下降沿中断

1x: INT1 电平变化中断

BIT[1:0] **MINT0** - 外部中断 0 模式寄存器

00: INT0 上升沿中断

01: INT0 下降沿中断

1x: INT0 电平变化中断



第三章 系统时钟

3 系统时钟

32P21 为双时钟系统，可根据需要通过软件在高速时钟和低速时钟之间任意切换。高速时钟可选择外接晶体振荡器或内置高精度 RC 振荡器；低速时钟可选用外接低频振荡器或内置低频 RC 振荡器。高低速时钟的模式有 OPBIT 的 MOSC 配置。

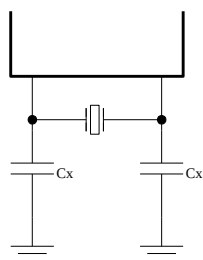
系统选用高频时钟，CPU 的运行速度由 OPBIT 的 FCPU 配置，选用低频时钟，CPU 的指令周期为 4 个低频时钟周期。

低速 RC 振荡器可用于 WDT（看门狗）电路使用。

3.1 外接晶体振荡器

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~16MHz，低频一般是接 32768Hz 时钟晶体，通常 Cx 是必须的。在实际使用中，用户应使晶体离 OSC1、OSC0 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 Cx 的推荐值。



晶体频率	电容 Cx
16MHz	15p
8MHz	15p
4MHz	15p/30p
3.64MHz	15p/30p
432KHz	220p/470p
32768Hz	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 振荡器

32P21 的内置高精度 16MHz RC 振荡器，该振荡器可用于系统高速时钟，其精度可达±2%。

内部高频振荡器频率调校寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCCAL	OSCCAL7	OSCCAL6	OSCCAL5	OSCCAL4	OSCCAL3	OSCCAL2	OSCCAL1	OSCCAL0
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]

OSCCALn - 内部高频 RC 振荡器频率调整寄存器

芯片复位后，OSCCAL 自动加载出厂默认值，该默认值将内部高频 RC 振荡器频率调整到 16MHz，该寄存器允许用户通过程序进行修改，以满足客户的其它频率要求。

3.3 内置低速 RC 振荡器

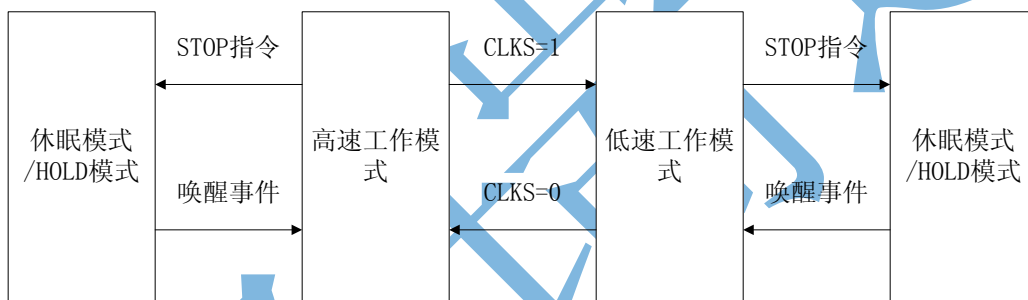
32P21 的内置一个低速 RC 振荡器，该振荡器可用于系统低频时钟，同时用于上电延时定时器、WDT。该振荡器频率典型值 28KHz，误差±50%。

3.4 工作模式

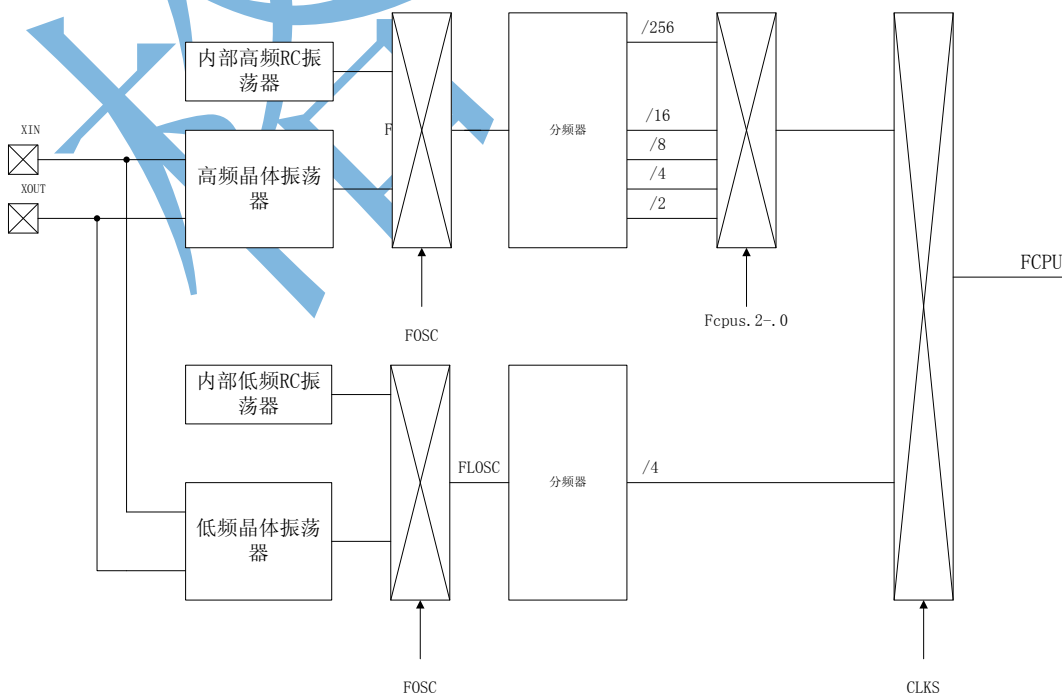
32P21 支持高速工作模式、低速工作模式、休眠模式、HOLD 模式 1 和 HOLD 模式 2 共有 5 种工作模式。

工作模式	进入条件
高速工作模式	系统时钟切换到高频振荡器 (CLKS=0)
低速工作模式	系统时钟切换到低频振荡器 (CLKS=1)
休眠模式	执行 STOP 指令, HFEN=0, LFEN=0
HOLD 模式 1	执行 STOP 指令, HFEN=1, LFEN=X (定时器可在高速时钟模式下继续工作, 溢出可唤醒)
HOLD 模式 2	执行 STOP 指令, HFEN=0, LFEN=1 (定时器可在低速时钟模式下继续工作, 溢出可唤醒)

工作模式间的切换



系统时钟选择



	高速工作模式	低速工作模式	休眠模式/HOLD 模式
高频振荡器	工作	HFEN 决定	HFEN 决定
低频振荡器	工作	工作	LFEN 决定
WDT 振荡器	工作	工作	WDTC 决定

注：当低速时钟选择为内部低频 RC 振荡器，则低频振荡器和 WDT 振荡器共用同一振荡器，低频振荡器工作或 WDT 振荡器工作都会使内部低频 RC 振荡器工作。

工作模式寄存器

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

BIT[7:6] 未用

BIT[5] **STBL** – 低速振荡器稳定标志

1: 低速振荡器已稳定运行

0: 低速振荡器停振或未稳定

BIT[4] **STBH** – 高速振荡器稳定标志

1: 高速振荡器已稳定运行

0: 高速振荡器停振或未稳定

BIT[3] 未用

BIT[2] **CLKS** – 系统工作时钟选择位

1: 低速时钟作为系统时钟

0: 高速时钟作为系统时钟

BIT[1] **LFEN** – 低频振荡器使能

1: 低频振荡器始终工作

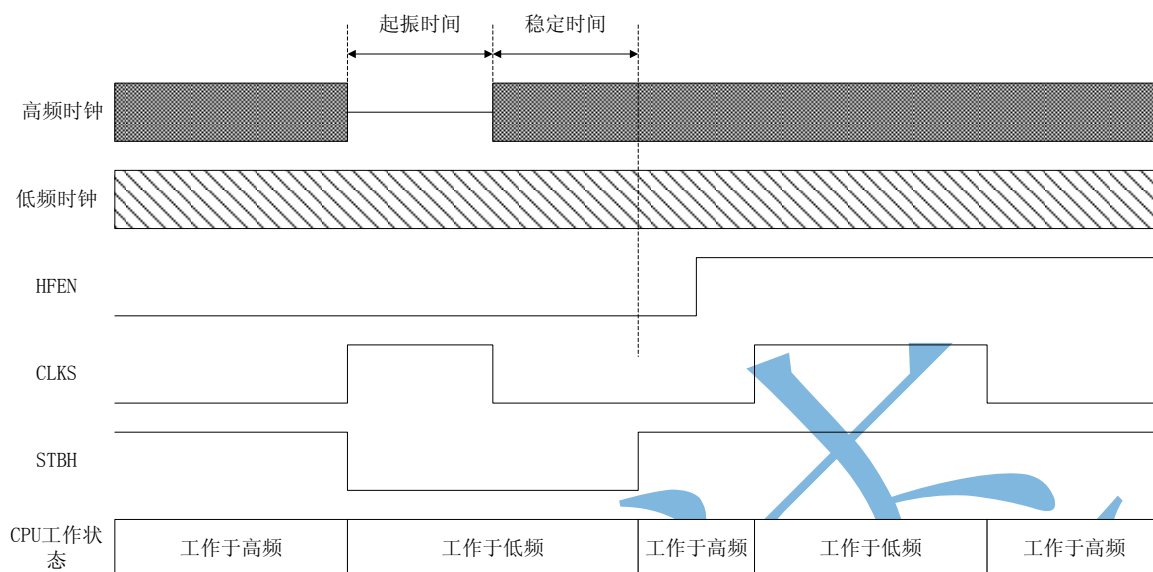
0: 在休眠/HOLD模式下，低频振荡器停止工作

BIT[0] **HFEN** – 高频振荡器使能

1: 高频振荡器始终工作

0: 在低速/休眠/HOLD模式下，高频振荡器停止工作

高低速时钟切换时序图



3.5 低功耗模式

进入低功耗工作方式：STOP 指令。

STOP 指令可使 MCU 进入 STOP 低功耗工作方式，同时对 MCU 会产生以下影响：

- ✧ 停止振荡器振荡。
- ✧ RAM 内容保持不变。
- ✧ 所有的输入输出端口保持原态不变。
- ✧ 定时器 0 和定时器 1 根据其工作模式，可以保持继续工作。

以下情况使 MCU 退出 STOP 方式：

- ✧ 有外部中断请求发生
- ✧ 有键盘中断发生
- ✧ 定时器 0、定时器 1 计数溢出中断请求发生
- ✧ 有 WDT 溢出
- ✧ 外部管脚复位

注 1：STOP 工作模式下，HFEN=0，LFEN=0，则系统停止了所有的操作，所以整体功耗水平非常低，静态电流小于 1uA。

注 2：STOP 模式下，HFEN=1，则高频振荡器仍然工作，系统功耗约 300uA

注 3：STOP 模式下，HFEN=0，LFEN=1，定时器选择 FLOSC，则定时器继续工作，溢出可唤醒中断，功耗小于 5uA

第四章 复位

4 复位

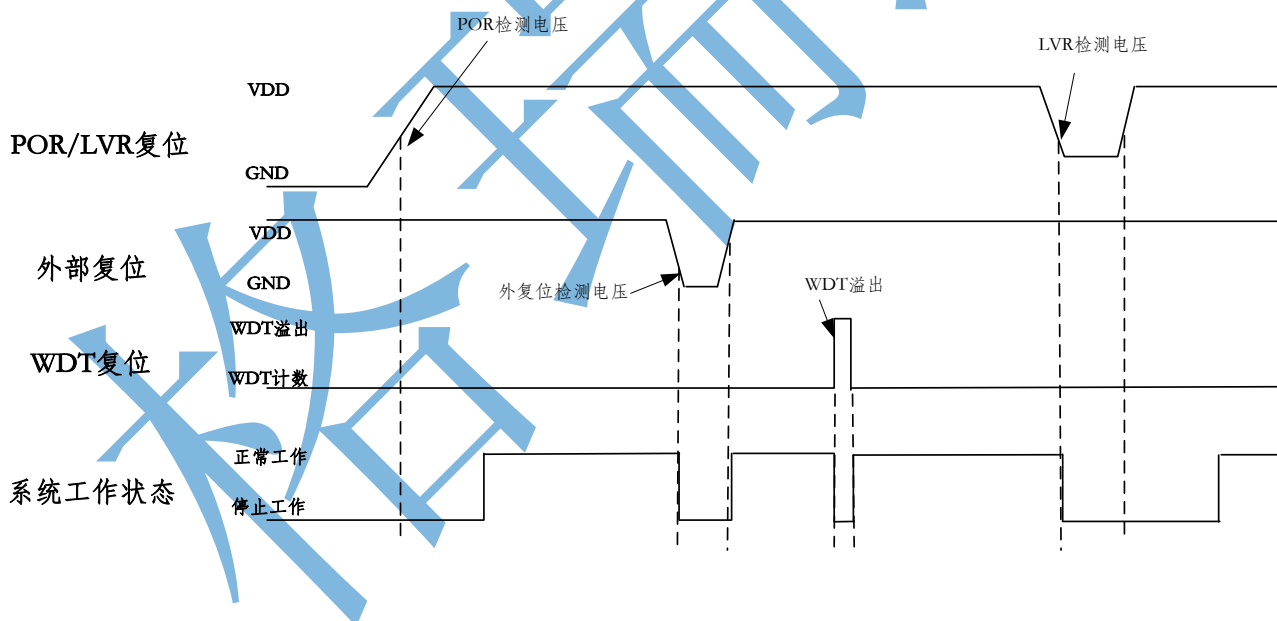
4.1 复位条件

32P21 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 0000H 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

32P21 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 MCLRE 配置，选择 MCLRE 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 掉电复位

32P21 的 LVR 电压有八级（详见烧录配置选项），通过 OPBIT 的 VLVR5 进行配置。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，则当电源电压下降到 LVR 时 LVR 复位有效，而电压需要上升到 LVR+0.5V 时 LVR 复位才会解除。

4.5 WDT 看门狗复位

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，出现程序跑飞的情况，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。

第五章 I/O 口

5 I/O口

5.1 IO 工作模式

一组 5 位端口 P0 和一组 7 位端口 P1。P0[4: 0] 分别和模拟通道 AN[4: 0] 复用，P1[3]、P1[2] 模拟通道 AN[6]、AN[5] 复用。P1[0]、P1[1] 设置为输入端口时，可复用为外部中断输入及定时器 0 和定时器 1 的外部时钟输入；P1 端口可作为键盘中断输入（端口变化中断），相应端口是否作为键盘输入可通过 KBCR 单独设置；当 PWM0OE=1 时，P1[2] 作为 PWM0 输出，PWM1OE=1 时，P1[3] 作为 PWM1 输出；当 BUZZ0OE=1 且 PWM0OE=0 时，P1[3] 作为 BUZZ0 输出，当 BUZZ1OE=1 且 PWM1OE=0 时，P1[3] 作为 BUZZ1 输出。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP0	-	-	-	P04D	P03D	P02D	P01D	P00D
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	-	-	-	X	X	X	X	X

BIT[4:0] P0nD - P0 口数据位 (n=4-0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP1	-	P16D	P15D	P14D	P13D	P12D	P11D	P10D
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

BIT[6:0] P1nD - P1 口数据位 (n=6-0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP0	-	-	-	P04OE	P03OE	P02OE	P01OE	P00OE
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	-	-	-	0	0	0	0	0

BIT[4: 0] P0nOE - P0 口输出使能寄存器 (n=4-0)

- 1: 作为输出口，读 P0 口读取 P0 口数据寄存器值
- 0: 作为输入口，读 P0 口读取端口状态

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP1	-	P16OE	P15OE	P14OE	P13OE	P12OE	P11OE	P10OE
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	0	0	0	0	0	0	0

BIT[6:0] P1nOE - P1 口输出使能寄存器 (n=6-0)

- 1: 作为输出口，读 P1 口读取 P1 口数据寄存器值
- 0: 作为输入口，读 P1 口读取端口状态

注： P16 作为 IO 输出时为开漏输出。配置为复位口时相应控制位无效

5.2 上拉电阻控制

P0 和 P1 口每位都有独立的上拉控制寄存器位，控制其上拉电阻在端口作为输入状态时是否有效，端口处于输出状态时，上拉电阻控制位无效。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUP0	–	–	–	P04PU	P03PU	P02PU	P01PU	P00PU
R/W	–	–	–	R/W	R/W	R/W	R/W	R/W
初始值	–	–	–	0	0	0	0	0

BIT[4:0] **P0nPU** – P0 口上拉电阻选择 (n=4-0)

1: P0n 上拉电阻有效

0: P0n 上拉电阻无效

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

BIT[6:0] **P1nPU** – P1 口上拉电阻选择 (n=6-0)

1: P1n 上拉电阻有效

0: P1n 上拉电阻无效

5.3 端口模式控制

P0 和 P1 部分口可以作为通用 IO 口，也可以复用为模拟信号输入端口，ANSEL 寄存器可以设置这些端口的工作模式。当设置为通用 IO 时，相应端口的模拟输入被屏蔽；设置为模拟输入模式时，相应端口的输入功能被屏蔽。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSEL	–	P13ANS	P12ANS	P04ANS	P03ANS	P02ANS	P01ANS	P00ANS
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	–	0	0	0	0	0	0	0

BIT[6:5] **P1nANS** – P1 口模式选择 (n=3-2)

1: P1n 端口作为模拟输入

0: P1n 端口作为通用 IO 口

BIT[4:0] **P0nANS** – P0 口模式选择 (n=4-0)

1: P0n 端口作为模拟输入

0: P0n 端口作为通用 IO 口

第六章 定时器

6 定时器

6.1 看门狗 (WDT)

看门狗定时器的时钟为独立 RC 时钟，由 OPBIT 的 WDTC 设置看门狗定时器的工作状态。

若选择 WDT 始终使能，在 STOP 下 WDT 依然运行，WDT 溢出时将唤醒休眠，CPU 继续运行；若 CPU 在运行时产生 WDT 溢出，WDT 溢出时复位芯片。

若选择 WDT 运行使能，STOP 关闭，在 STOP 下 WDT 被硬件自动关闭。

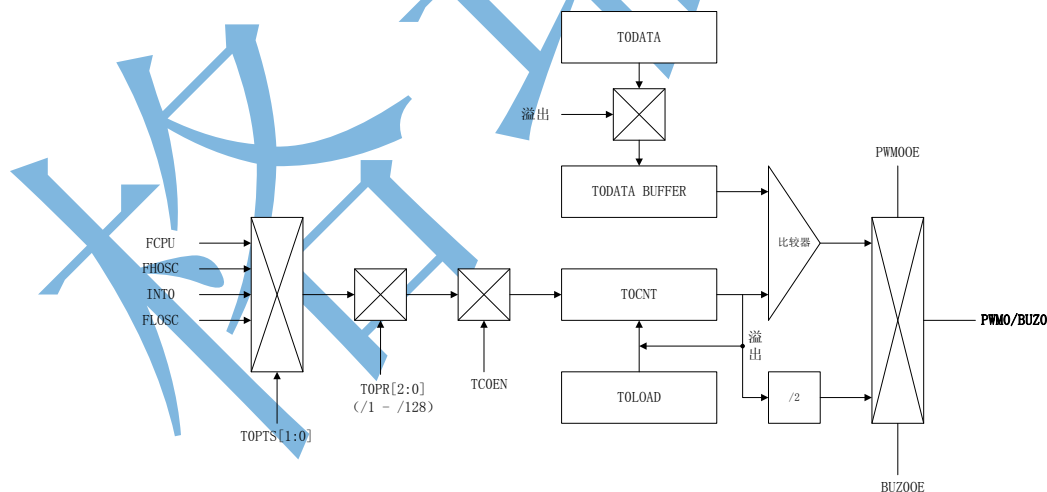
对 WDTC 指令能清 WDT 计数器。

WDT 溢出时间可通过配置设置为 4ms、16ms、64ms、256ms、1024ms、4096ms。

6.2 定时器 T0

2 个包含 PWM 和 BUZ 功能的定时计数器。定时/计数器 0 包含 1 个可编程预分频器，控制寄存器、重载寄存器及比较寄存器。

- 可通过预分频比设置计数频率
- 通过重载寄存器控制计数周期
- 通过比较寄存器设置 PWM 占空比（仅 PWM 模式）
- BUZ 功能
- 溢出中断功能
- 溢出中断唤醒功能



注 1：FHOSC 指系统高频时钟系统时钟；FLOSC 指系统低频时钟源；

注 2：当定时器选择高频时钟源 FHOSC 且时钟控制模块 OSCM 寄存器的 HFEN=1 时，定时器在低频工作模式或休眠模式时可继续工作，溢出中断可唤醒休眠模式；若 HFEN=0 时，定时器在低频或休眠模式下将停止工作

注 3：当定时器选择低频时钟源 FLOSC 且时钟控制模块 OSCM 寄存器的 LFEN=1 时，定时器在休眠模式时可继续工作，溢出中断可唤醒休眠模式；若 LFEN=0 时，定时器在休眠模式下将停止工作

TOPTS 可选择 T0 的时钟源，TOPR 可选择 T0 的预分频比，所选中的时钟源通过预分频器后产生 TOCNT

的计数时钟。

当 TOCNT 递减到 0 时, 此时产生 T0 溢出中断请求标志 T0IF/T1IF 置 1, 重载寄存器值自动置入 TOCNT, TODATA 的值写入缓冲器 TODATABUF 用于新的占空比波形生成, BUZO 信号反相。

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

当 PWM0OE=1 时, 将输出 PWM 波形, 当 TOCNT 计数到与 TODATA 相等时, PWM0 输出置 1; 当 TOCNT 计数溢出时, PWM0 输出清 0, PWM0 占空比的计算如下:

$\text{PWM0 高电平时间} = (\text{TODATA}) * \text{TOCNT 计数时钟周期}$

$\text{PWM0 周期 (T0 的溢出周期)} = (\text{TOLOAD}+1) * \text{TOCNT 的计数周期}$

$\text{PWM0 占空比} = (\text{TODATA} / (\text{TOLOAD}+1))$

当 BUZOOE=1 且 PWM0OE=0 时, 输出 BUZO 信号, BUZO 信号的输出频率为 T0 溢出频率的 2 分频。

与定时器 T0 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
TOCR	TCOEN	PWM0OE	BUZOOE	TOPTS1	TOPTS0	TOPR2	TOPR1	TOPR0
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] **TCOEN** - T0 使能控制

0: 关闭 T0

1: 启动 T0

BIT[6] **PWM0OE** - PWM0 选择

0: 禁止 PWM0 输出, 端口作为 I/O 口

1: 允许 PWM0 输出, 端口输出 PWM 信号

BIT[5] **BUZOOE** - BUZO 选择

0: 禁止 BUZO 输出, 端口作为 I/O 口

1: 允许 BUZO 输出 (PWM0OE), 端口输出 BUZ 信号

BIT[4:3] **TOPTS[1: 0]** - T0 时钟源选择

TOPTS[1: 0]	T0 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	INT0

BIT[2:0] **TOPR[2: 0]** - T0 预分频倍数选择

TOPR2	TOPR1	TOPR0	TOCNT
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
TOCNT	TOC7	TOC6	TOC5	TOC4	TOC3	TOC2	TOC1	TOC0
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] **TOC[7: 0]** - TOCNT 的值，这是一个读写寄存器。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1LOAD	TOLOAD7	TOLOAD6	TOLOAD5	TOLOAD4	TOLOAD3	TOLOAD2	TOLOAD1	TOLOAD0
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] **TOLOAD[7: 0]** - TOLOAD 的值，这是一个读写寄存器，用于设置 T0 重载值。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TODATA	TODATA7	TODATA6	TODATA5	TODATA4	TODATA3	TODATA2	TODATA1	TODATA0
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] **TODATA[7: 0]** - 这是一个读写寄存器，用于设置 PWM0 高电平时间。

注：当 TCOEN=0 时，写 TOLOAD 将自动加载到 TOCNT；当 TCOEN=1 时，写 TOLOAD 时不自动加载到 TOCNT，在计时器溢出时自动加载到 TOCNT

6.3 定时器 T1

定时器 1 与定时器 0 的功能及操作模式完全相同，在此仅做寄存器介绍。

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
T1CR	TC1EN	PWM1OE	BUZ1OE	T1PTS1	T1PTS0	T1PR2	T1PR1	T1PR0
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] **TC1EN** - T1 使能控制

0: 关闭 T1

1: 启动 T1

BIT[6] **PWM1OE** - PWM1 选择

0: 禁止 PWM1 输出，端口作为 I/O 口

1: 允许 PWM1 输出，端口输出 PWM 信号

BIT[5] **BUZ1OE** - BUZ1 选择

0: 禁止 BUZ1 输出，端口作为 I/O 口

1: 允许 BUZ1 输出 (PWM1OE)，端口输出 BUZ 信号

BIT[4:3] **T1PTS[1: 0]** - T1 时钟源选择

T1PTS[1: 0]	T1 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	INT0

BIT[2:0] **T1PR[2: 0]** - T1 预分频倍数选择

T1PR2	T1PR1	T1PR0	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	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] **T1C[7: 0]** - T1CNT 的值，这是一个读写寄存器。

	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]** - T1LOAD 的值，这是一个读写寄存器，用于设置 T1 重载值。

	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]** - 这是一个读写寄存器，用于设置 PWM1 高电平时间。

注：当 TC1EN=0 时，写 T1LOAD 将自动加载到 T1CNT；当 TC1EN=1 时，写 T1LOAD 时不自动加载到 T1CNT，在计时器溢出时自动加载到 T1CNT

第七章 模数转换器

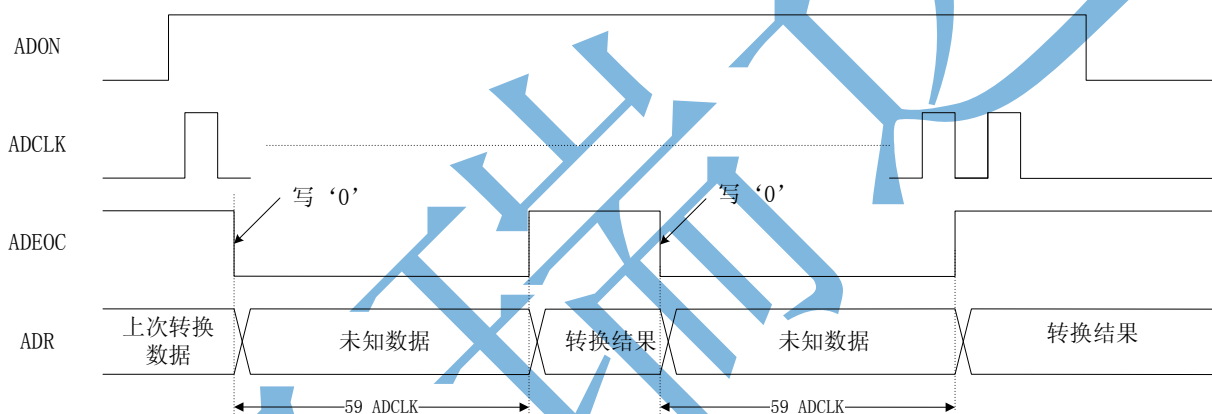
7 模数转换器（ADC）

7.1 ADC 功能介绍

7 通道 12 位模数转换器，可通过 ADON 使能模数转换模块，ADCHS 选择转换的模拟通道，ADCKS 选择 AD 转换速度，ADEOC 为 AD 启动位及转换结束标志位。

当 ADEOC 标志为 ‘1’ 时，对该寄存器写入 ‘0’ 将启动模数转换，转换时间需要 59 个 AD 转换时钟周期，转换结果被放在 ADRH 和 ADRL 中，ADEOC 将自动置 ‘1’，同时中断标志 ADIF 置 ‘1’，若 GIE 和 ADIE 使能，将产生 AD 中断。

7.2 ADC 转换时序图



7.3 ADC 操作步骤

模数转换设置步骤：

- ✧ S1: 设置 OEPX 将相应的端口设置为输入端口
- ✧ S2: 设置 ANSEL 将相应的端口设置为模拟端口
- ✧ S3: 设置 ADCKS 选取适当的 AD 转换时钟
- ✧ S4: 设置参考电压
- ✧ S5: 使能 ADON ^{注 2}
- ✧ S6: 设置 ADCHS 选取 AD 转换通道
- ✧ S7: ADEOC 写入 ‘0’ 启动 AD 转换
- ✧ S8: 等待 ADEOC 置 ‘1’（或利用 AD 中断）
- ✧ S9: 读取 AD 转换结果（ADDRH、ADDRL）
- ✧ 重复 S6~S8 对不同的通道进行转换或对同一通道多次转换

注 1: AD 转换过程中或者 ADON 未使能时，ADDRH/ADDRL 中的数据未知，选在 AD 转换结束且 ADON 使能的情况下读取 AD 转换数据

注 2: 如果选择内部参考电压 2V/3V/4V 时，需要在使能 ADON 后等待参考电压稳定（时间 > 200us）

7.4 ADC 相关寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCRO	ADCHS3	ADCHS2	ADCHS1	ADCHS0	ADCKS1	ADCKS0	ADEOC	ADON
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	1	0

BIT[7:4] **ADCHS[3: 0]** - ADC 模拟通道选择

ADCHS[3: 0]	ADC 模拟通道选择
0000	AIN0
0001	AIN1
0010	AIN2
0011	AIN3
0100	AIN4
0101	AIN5
0110	AIN6
其它	保留

BIT[3:2] **ADCKS[1: 0]** - AD 转换时钟选择

ADCKS[1: 0]	AD 转换时钟频率选择 F_{ADC}
00	$F_{cpu}/16$
01	$F_{cpu}/8$
10	$F_{cpu}/4$
11	$F_{cpu}/2$

BIT[1] **ADEOC** - ADC 启动位及转换结束标志
 1: AD 转结束, 对 ADEOC 写入 0 启动 AD 转换
 0: AD 转换过程中, 转换结束后自动置 1

BIT[0] **ADON** - ADC 功能使能位
 1: 使能 ADC 功能
 0: 不使能 ADC 功能

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCR1	-	-	-	-	-	VRS2	VRS1	VRS0
R/W	-	-	-	-	-	R/W	R/W	R/W
初始值	-	-	-	-	-	0	0	0

BIT[2:0] **VRS[2: 0]** - ADC 参考电压选择

VRS[2: 0]	ADC 参考电压选择
011	内部参考 VDD
010	内部参考 4V
001	内部参考 3V
000	内部参考 2V
1xx	外部参考电压

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

BIT[7:0] **ADRH** - ADC 转换结果寄存器高 8 位 (n=11-4)。

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

BIT[3:0] **ADRL** - ADC 转换结果寄存器低 4 位 (n=3-0)。

第八章 低电压检测

8 低电压检测 (LVD)

32P21 内嵌低电压检测模块，通过设置 LVDEN 可以开启或屏蔽该功能，LVDS 选择检测电压值，可设置 1.8V-3.6V 等不同电压，当 VDD 电压低于设置电压时 LVDF 置 1，否则 LVDF 清 0。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LVDCR	LVDEN	LVDS2	LVDS1	LVDS0	—	—	—	LVDF
R/W	R/W	R/W	R/W	R/W	—	—	—	R
初始值	0	0	0	0	—	—	—	0

BIT[7] **LVDEN** - LVD 功能使能位

1: 使能 LVD 功能

0: 不使能 LVD 功能

BIT[6:4] **LVDS[2:0]** - LVD 检测电压选择

LVDS[2: 0]	LVD 检测电压设置 (参考值)
000	关闭 LVD
001	1.85V
010	2.05V
011	2.18V
100	2.32V
101	2.45V
110	3.05V
111	3.60V

BIT[0] **LVDF** - LVD 检测标志

1: VDD 电压低于设置电压

0: VDD 电压高于设置电压或 LVD 不使能

第九章 中断

9 中断

32P21 的中断有外中断 (INT0, INT1)、键盘中断、定时器中断 (T0, T1)、ADC 转换中断。外部中断、键盘中断、定时器中断可被 CPU 状态寄存器 MCR 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 0008H 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。

32P21 的中断向量地址是 0008H。

9.1 外中断

32P21 有 2 路外部中断源，两路中断源可以设置为上升沿触发、下降沿触发和变化触发三种模式，当外部中断触发时，外部中断标志 (INT0IF、INT1IF) 将被置 1，若中断总使能位 GIE 为 1 且外部中断使能位 (INT0IE、INT1IE) 为 1，则产生外部中断。

9.2 键盘中断

32P21 有 7 路键盘中断源，7 路中断源可以通过 KBCR 寄存器单独屏蔽，任意一路未被屏蔽的中断源电平放生变化时，触发键盘中断，键盘中断标志 (KBIF) 将被置 1，若中断总使能位 GIE 为 1 且键盘中断使能位 (KBIE) 为 1，则产生键盘中断。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
KBCR	—	KBCR6	KBCR5	KBCR4	KBCR3	KBCR2	KBCR1	KBCR0
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	—	0	0	0	0	0	0	0

BIT[6:0] KBCRn - P1 端口键盘中断使能 (n=6-0)

1: 使能 P1n 端口键盘中断功能

0: 屏蔽 P1n 端口键盘中断功能

9.3 定时器中断

定时器 T0、T1、在计数溢出时会置位中断标志 T0IF、T1IF，若中断总使能位 GIE 为 1 且定时器中断使能位 (T0IE、T1IE) 为 1，则产生定时器中断。

9.4 ADC 转换中断

ADC 转换完成后会置位中断标志 ADIF，若中断总使能位 GIE 为 1 且定时器中断使能位 (ADIE) 为 1，

则产生 ADC 中断。

9.5 中断相关寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCR	GIE	–	TO	PD	MINT11	MINT10	MINT01	MINT01
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** – 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[6] 未用
- BIT[5] **TO** – 看门狗溢出标志
 0: 上电复位, 执行CLRWDWT或STOP指令
 1: 发生WDT溢出
- BIT[4] **PD** – 进入低功耗休眠模式标志
 0: 上电复位, 执行CLRWDWT
 1: 执行STOP指令
- BIT[3:2] **MINT1** – 外部中断 1 模式寄存器
 00: INT1 上升沿中断
 01: INT1 下降沿中断
 1x: INT1 电平变化中断
- BIT[1:0] **MINT0** – 外部中断 0 模式寄存器
 00: INT0 上升沿中断
 01: INT0 下降沿中断
 1x: INT0 电平变化中断

中断使能寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTE	–	ADIE	–	KBIE	INT1IE	INT0IE	T1IE	TOIE
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] **ADIE** – ADC 中断使能
 0: 屏蔽 ADC 中断
 1: 使能 ADC 中断
- BIT[4] **KBIE** – 键盘中断使能
 0: 屏蔽键盘中断
 1: 使能键盘中断
- BIT[3] **INT1IE** – 外部中断 1 使能
 0: 屏蔽外部 1 中断
 1: 使能外部 1 中断
- BIT[2] **INT0IE** – 外部中断 0 使能
 0: 屏蔽外部 0 中断
 1: 使能外部 0 中断
- BIT[1] **T1IE** – 定时器 1 使能

- 0: 屏蔽定时器 1 中断
1: 使能定时器 1 中断
- BIT[0] **TOIE** - 定时器 0 使能
0: 屏蔽定时器 0 中断
1: 使能定时器 0 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	–	ADIF	–	KBIF	INT1IF	INT0IF	T1IF	TOIF
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] **ADIF** - ADC 中断标志
0: 未发生 ADC 中断
1: 发生 ADC 中断, 需软件清零
- BIT[4] **KBIF** - 键盘中断标志
0: 未发生键盘中断
1: 发生键盘中断, 需软件清零
- BIT[3] **INT1IF** - 外部中断 1 标志
0: 未发生外部 1 中断
1: 发生外部 1 中断, 需软件清零
- BIT[2] **INT0IF** - 外部中断 0 标志
0: 未发生外部 0 中断
1: 发生外部 0 中断, 需软件清零
- BIT[1] **T1IF** - 定时器 1 标志
0: 未发生屏蔽定时器 1 中断
1: 发生定时器 1 中断, 需软件清零
- BIT[0] **TOIF** - 定时器 0 标志
0: 未发生定时器 0 中断
1: 发生定时器 0 中断, 需软件清零

第十章 指令集

10 指令集概述

MC32 是一种精简指令集 (RISC)，总共有 73 条指令。指令为 16 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 § 1)。

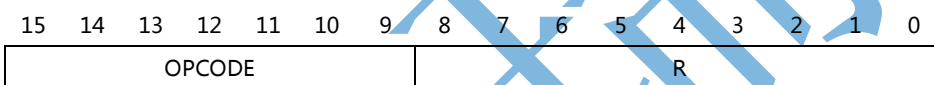
表一是文中所用符号的说明。

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

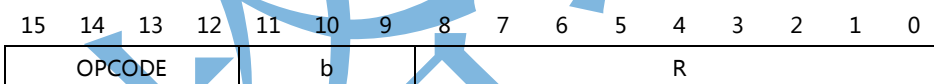
指令按照功能可分为 4 类，即字节操作指令、位操作指令、立即数指令、特殊控制指令、长跳转指令，以下简要说明各类指令的格式。

(1) 字节操作指令



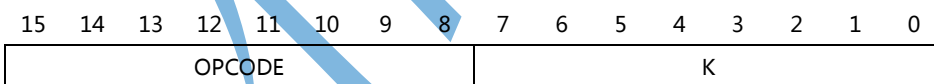
字节操作指令的 OPCODE 长度为 7 位，寄存器地址宽度为 9 位，即地址访问范围 000H~1FFH。

(2) 位操作指令

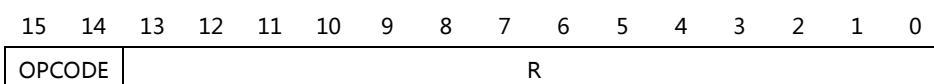


位操作指令的 OPCODE 长度为 4 位，位地址 (0~7) 占 3 位，寄存器地址也是 9 位。

(3) 立即数指令

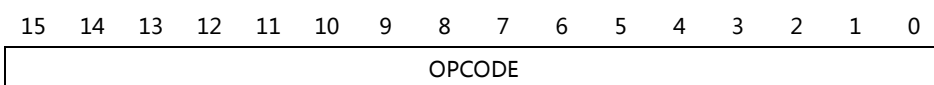


立即数指令的 OPCODE 长度为 8 位，立即数长度为 8 位。



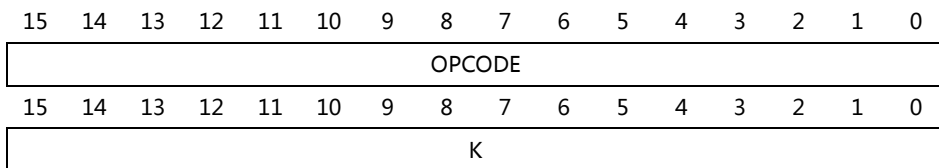
CALL/GOTO 指令的 OPCODE 长度为 3 位，立即数长度为 13 位。

(4) 特殊控制指令



特殊控制指令全部 16 位均为 OPCODE。

(5) 长跳转指令



LCALL/LGOTO 是长跳转指令，为双字节指令，第一个字节 16 位是 OPCODE，第二个字节的 16 位是跳转地址，跳转范围为 0000H~FFFFH。

10.1 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→ACC	1	0111 110r rrrr rrrr	C, DC, Z
ADDRA R	寄存器 R 内容和 ACC 相加，结果存到 R	R+ACC→R	1	0111 111r rrrr rrrr	C, DC, Z
ADCAR R	带 C 标志的加法，结果存到 ACC	R+ACC+C→ACC	1	0001 000r rrrr rrrr	C, DC, Z
ADCRA R	带 C 标志的加法，结果存到 R	R+ACC+C→R	1	0001 001r rrrr rrrr	C, DC, Z
RSUBAR R	寄存器 R 内容和 ACC 相减，结果存到 ACC	R-ACC→ACC	1	0100 100r rrrr rrrr	C, DC, Z
RSUBRA R	寄存器 R 内容和 ACC 相减，结果存到 R	R-ACC→R	1	0100 101r rrrr rrrr	C, DC, Z
RSBCAR R	寄存器 R 内容和 ACC 相减(带 C 标志)，结果存到 ACC	R-ACC-/C→ACC	1	0001 100r rrrr rrrr	C, DC, Z
RSBCRA R	寄存器 R 内容和 ACC 相减(带 C 标志)，结果存到 R	R-ACC-/C→R	1	0001 101r rrrr rrrr	C, DC, Z
ASUBAR R	ACC 和寄存器 R 内容相减，结果存到 ACC	ACC-R→ACC	1	0000 100r rrrr rrrr	C, DC, Z
ASUBRA R	ACC 和寄存器 R 内容相减，结果存到 R	ACC-R→R	1	0000 101r rrrr rrrr	C, DC, Z
ASBCAR R	ACC 和寄存器 R 内容相减(带 C 标志)，结果存到 ACC	ACC-R-/C→ACC	1	0000 010r rrrr rrrr	C, DC, Z
ASBCRA R	ACC 和寄存器 R 内容相减(带 C 标志)，结果存到 R	ACC-R-/C→R	1	0000 011r rrrr rrrr	C, DC, Z
ANDAR R	寄存器 R 内容和 ACC 与操作，结果存到 ACC	R and ACC→ACC	1	0111 100r rrrr rrrr	Z
ANDRA R	寄存器 R 内容和 ACC 与操作，结果存到 R	R and ACC→R	1	0111 101r rrrr rrrr	Z
ORAR R	寄存器 R 内容和 ACC 或操作，结果存到 ACC	R or ACC→ACC	1	0101 110r rrrr rrrr	Z
ORRA R	寄存器 R 内容和 ACC 或操作，结果存到 R	R or ACC→R	1	0101 111r rrrr rrrr	Z
XORAR R	寄存器 R 内容和 ACC 异或操作，结果存到 ACC	R xor ACC→ACC	1	0100 000r rrrr rrrr	Z
XORRA R	寄存器 R 内容和 ACC 异或操作，结果存到 R	R xor ACC→R	1	0100 001r rrrr rrrr	Z
COMAR R	对 R 取反，结果存到 ACC	R 取反→ACC	1	0111 000r rrrr rrrr	Z
COMR R	对 R 取反，结果存到 R	R 取反→R	1	0111 001r rrrr rrrr	Z
CLRA	对 ACC 清零	0→ACC	1	0111 0100 0000 0000	Z
CLRR R	对 R 清零	0→R	1	0111 011r rrrr rrrr	Z
RLA	ACC 循环左移(带 C 标志)	ACC[7]→C ACC[6:0]→ACC[7:1] C→ACC[0]	1	0000 0000 0001 0100	C
RLAR R	寄存器 R 循环左移(带 C 标志)，结果存到 ACC	R[7]→C R[6:0]→ACC[7:1] C→ACC[0]	1	0101 000r rrrr rrrr	C

RLR R	寄存器 R 循环左移(带 C 标志), 结果存到 R	R[7]→C R[6:0]→R[7:1] C→R[0]	1	0101 001r rrrr rrrr	C
RRA	ACC 循环右移(带 C 标志)	C→ACC[7] ACC[7:1]→ACC[6:0] ACC[0]→C	1	0000 0000 0001 0101	C
RRAR R	寄存器 R 循环右移(带 C 标志), 结果存到 ACC	C→ACC[7] R[7:1]→ACC[6:0] R[0]→C	1	0100 110r rrrr rrrr	C
RRR R	寄存器 R 循环右移(带 C 标志), 结果存到 R	C→R[7] R[7:1]→R[6:0] R[0]→C	1	0100 111r rrrr rrrr	C
SWAPAR R	交换 R 的高低字节, 结果存到 ACC	R[7:4]→ACC[3:0] R[3:0]→ACC[7:4]	1	0100 010r rrrr rrrr	-
SWAPR R	交换 R 的高低字节, 结果存到 R	R[7:4]→R[3:0] R[3:0]→R[7:4]	1	0100 011r rrrr rrrr	-
MULAR R	寄存器 R 内容和 ACC 相乘, 结果存到 HIBYTE:ACC	R*ACC→ HIBYTE:ACC ⁽²⁾	1	0001 010r rrrr rrrr	-
MULRA R	寄存器 R 内容和 ACC 相乘 结果存到 HIBYTE:R	R*ACC→HIBYTE:R ⁽²⁾	1	0001 011r rrrr rrrr	-
MOVAR R	将 R 存到 ACC	R→ACC	1	0101 100r rrrr rrrr	Z
MOVR R	将 R 存到 R	R→R	1	0101 101r rrrr rrrr	Z
MOVRA R	将 ACC 存到 R	ACC→R	1	0101 011r rrrr rrrr	-
INCA	ACC 自加 1	ACC+1→ACC	1	0000 0000 0001 0011	-
INCAR R	R 加 1, 结果存到 ACC	R+1→ACC	1	0110 010r rrr rrrr	Z
INCR R	R 加 1, 结果存到 R	R+1→R	1	0110 011r rrr rrrr	Z
DECA	ACC 自减 1	ACC-1→ACC	1	0000 0000 0001 0001	-
DECAR R	R 减 1, 结果存到 ACC	R-1→ACC	1	0110 110r rrrr rrrr	Z
DECR R	R 减 1, 结果存到 R	R-1→R	1	0110 111r rrrr rrrr	Z
JZA	ACC 自加 1; 结果为 0, 则跳过下一条指令	ACC+1→ACC, 结果为 0, 则 PC+2→PC	1 或 2 ⁽¹⁾	0000 0000 0001 0010	-
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⁾	0110 000r rrrr 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⁾	0110 001r rrrr rrrr	-
DJZA	ACC 自减 1; 结果为 0, 则跳过下一条指令	ACC-1→ACC, 结果为 0, 则 PC+2→PC	1 或 2 ⁽¹⁾	0000 0000 0001 0000	-
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⁾	0110 100r rrrr 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⁾	0110 101r rrrr rrrr	-
BCLR R,b	对 R 的第 b 位清零	0→R[b]	1	1101 bbbr rrrr rrrr	-
BSET R,b	对 R 的第 b 位置 1	1→R[b]	1	1100 bbbr rrrr 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→	1 或 2 ⁽¹⁾	1111 bbbr rrrr rrrr	-

		PC			
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⁾	1110 bbb rrrrr rrrrr	-
ADDAI K	立即数 K 和 ACC 相加, 结果存到 ACC	K+ACC→ACC	1	0011 1111 kkkk kkkk	C, DC, Z
ADCAI K	立即数 K 和 ACC 相加(带 C 标志) 结果存到 ACC	K and ACC→ACC	1	0011 1000 kkkk kkkk	C, DC, Z
ISUBAI K	立即数和 ACC 相减, 结果存到 ACC	K-ACC→ACC	1	0011 1010 kkkk kkkk	C, DC, Z
ISBCAI K	立即数和 ACC 相减(带 C 标志), 结果存到 ACC	K-ACC-/C→ACC	1	0011 0111 kkkk kkkk	C, DC, Z
ASUBAI K	ACC 和立即数相减, 结果存到 ACC	ACC-K→ACC	1	0011 0101 kkkk kkkk	C, DC, Z
ASBCAI K	ACC 和立即数相减(带 C 标志), 结果存到 ACC	ACC-K-/C→ACC	1	0011 0110 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	0011 1110 kkkk kkkk	Z
ORAI K	立即数 K 和 ACC 或操作, 结果存到 ACC	K or ACC→ACC	1	0011 1101 kkkk kkkk	Z
XORAI K	立即数和 ACC 异或, 结果存到 ACC	K xor ACC→ACC	1	0011 1001 kkkk kkkk	Z
MOVAI K	将立即数存到 ACC	K→ACC	1	0011 1100 kkkk kkkk	-
RETAI K	从子程序返回, 并将立即数存到 ACC	TOS→PC K→ACC	2	0011 1011 kkkk kkkk	-
RETURN	从子程序返回	TOS→PC	2	0000 0000 0000 1100	-
RETIE	从中断返回	TOS→PC 1→GIE	2	0000 0000 0000 1101	-
CALL K	子程序调用	PC+1→TOS K→PC[12:0]	2	100k kkkk kkkk kkkk	-
GOTO K	无条件跳转	K→PC[12:0]	2	101k kkkk kkkk kkkk	-
NOP	空操作	空操作	1	0000 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	0000 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	0000 0000 0000 1001	-
CLRWDI	清看门狗定时器	0→WDT	1	0000 0000 0000 1110	TO, PD
STOP	进入休眠模式	0→WDT 进入休眠模式	1	0000 0000 0000 1011	TO, PD
LCALL	长子程序调用	PC+1→TOS K→PC[15:0]	2	0010 0001 0000 0000 kkkk kkkk kkkk kkkk	-
LGOTO	长无条件跳转	K→PC[15:0]	2	0010 0001 0001 0000 kkkk kkkk kkkk kkkk	-

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。
- (2) HIBYTE 是一个寄存器, 用于存放乘法结果的高 8 位。

10.2 指令详细说明

10.2.1 ADCAI

语法：ADCAI K

操作数： $0 \leq K \leq 255$

操作： $K + ACC + C \rightarrow ACC$

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：立即数 K 和 ACC 相加（带 C 标志），结果存到 ACC

周期数：1

10.2.2 ADCAR

语法：ADCAR R

操作数： $0 \leq R \leq 511$

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

10.2.3 ADCRA

语法：ADCRA R

操作数： $0 \leq R \leq 511$

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

10.2.4 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

10.2.5 ADDAR

语法： ADDAR R

操作数： $0 \leq R \leq 511$

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

10.2.6 ADDRA

语法： ADDRA R

操作数： $0 \leq R \leq 511$

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

10.2.7 ANDAI

语法： ADCAI K

操作数： $0 \leq K \leq 511$

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

10.2.8 ANDAR

语法： ANDAR R

操作数： $0 \leq R \leq 511$
操作： $R \text{ and } ACC \rightarrow ACC$
影响状态： Z
 Z 与的结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 寄存器 R 内容和 ACC 与操作，结果存到 ACC
周期数： 1

10.2.9 ANDRA

语法： ANDRA R
操作数： $0 \leq R \leq 511$
操作： $R \text{ and } ACC \rightarrow R$
影响状态： Z
 Z 与的结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 寄存器 R 内容和 ACC 与操作，结果存到 R
周期数： 1

10.2.10 ASBCAI

语法： ASBCAI K
操作数： $0 \leq K \leq 255$
操作： $ACC - K / C \rightarrow ACC$
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： ACC 和立即数 K 相减（带 C 标志），结果存到 ACC
周期数： 1

10.2.11 ASBCAR

语法： ASBCAR R
操作数： $0 \leq R \leq 511$
操作： $R - ACC / C \rightarrow ACC$
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： ACC 与寄存器 R 内容减法（带 C 标志），结果存到 ACC
周期数： 1

10.2.12 ASBCRA

语法：ASBCRA R

操作数： $0 \leq R \leq 511$

操作：R-ACC-/C→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 $0 \rightarrow DC$ ；否则 $1 \rightarrow DC$

Z 相减结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$

说明：ACC 与寄存器 R 内容减法（带 C 标志），结果存到 R

周期数：1

10.2.13 ASUBAI

语法：ASUBAI K

操作数： $0 \leq K \leq 255$

操作：ACC-K→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：ACC 和立即数 K 相减，结果存到 ACC

周期数：1

10.2.14 ASUBAR

语法：ASUBAR R

操作数： $0 \leq R \leq 511$

操作：R-ACC→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：ACC 与寄存器 R 内容减法，结果存到 ACC

周期数：1

10.2.15 ASUBRA

语法：ASUBRA R

操作数： $0 \leq R \leq 511$

操作：R-ACC→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 $0 \rightarrow DC$ ；否则 $1 \rightarrow DC$

Z 相减结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$

说明：ACC 与寄存器 R 内容减法，结果存到 R

周期数：1

10.2.16 BCLR

语法： BCLR R, b
操作数： $0 \leq R \leq 511, 0 \leq b \leq 7$
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

10.2.17 BSET

语法： BSET R, b
操作数： $0 \leq R \leq 511, 0 \leq b \leq 7$
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

10.2.18 CALL

语法： CALL K
操作数： $0 \leq K \leq 8191$
操作： $PC+1 \rightarrow TOS(\text{Top of Stack}); K \rightarrow PC[12:0]$
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

10.2.19 CLRA

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

10.2.20 CLRR

语法： CLRR R
操作数： $0 \leq R \leq 511$
操作： $0 \rightarrow R$
影响状态： Z
 Z $1 \rightarrow Z$
说明： 对 R 清零
周期数： 1

10.2.21 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

10.2.22 COMAR

语法： COMAR R
操作数： $0 \leq R \leq 511$
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

10.2.23 COMR

语法： COMR R
操作数： $0 \leq R \leq 511$
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

10.2.24 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

10.2.25 DECA

语法：DECA
操作数：无
操作：ACC-1→ACC
影响状态：Z
Z 减 1 结果为 0，则 1→Z；否则 0→Z
说明：ACC 内容自减 1
周期数：1

10.2.26 DECAR

语法：DECAR R
操作数： $0 \leq R \leq 511$
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

10.2.27 DECR

语法：DECR R
操作数： $0 \leq R \leq 511$
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

10.2.28 DJZA

语法：DJZA
操作数：无
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：ACC 内容自减 1，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)

10.2.29 DJZAR

语法：DJZAR R
操作数： $0 \leq R \leq 511$
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)

10.2.30 DJZR

语法：DJZR R
操作数： $0 \leq R \leq 511$
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)

10.2.31 DSA

语法：DSA
操作数：无
操作： $ACC(\text{十六进制}) \rightarrow ACC(\text{十进制})$
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

10.2.32 GOTO

语法：GOTO K
操作数： $0 \leq K \leq 8191$
操作： $K \rightarrow PC[12:0]$
影响状态：无影响
说明：无条件跳转
周期数：2

10.2.33 INCA

语法：INCA
操作数：无
操作： $R+1 \rightarrow ACC$
影响状态：Z
Z 加 1 结果为 0, 则 $1 \rightarrow Z$; 否则 $0 \rightarrow Z$
说明：ACC 内容加 1, 结果存到 ACC
周期数：1

10.2.34 INCAR

语法：INCAR R
操作数： $0 \leq R \leq 511$
操作： $R+1 \rightarrow ACC$
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

10.2.35 INCR

语法：INCR R

操作数： $0 \leq R \leq 511$
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

10.2.36 ISBCAI

语法： ISBCAI K
操作数： $0 \leq K \leq 255$
操作： $K-ACC \rightarrow ACC$
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（带 C 标志），结果存到 ACC
周期数： 1

10.2.37 ISUBAI

语法： ISUBAI K
操作数： $0 \leq K \leq 255$
操作： $K-ACC \rightarrow ACC$
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

10.2.38 JBCLR

语法： JBCLR R, b
操作数： $0 \leq R \leq 511, 0 \leq b \leq 7$
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）

10.2.39 JBSET

语法： JBSET R, b
操作数： $0 \leq R \leq 511, 0 \leq b \leq 7$
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）

10.2.40 JZA

语法： JZA
操作数： 无
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： ACC 内容自加 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）

10.2.41 JZAR

语法： JZAR R
操作数： $0 \leq R \leq 511$
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）

10.2.42 JZR

语法： JZR R
操作数： $0 \leq R \leq 511$
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）

10.2.43 LCALL

语法： LCALL K
操作数： $0 \leq K \leq 65535$
操作： $PC+1 \rightarrow TOS(Top\ of\ Stack)$ ； $K \rightarrow PC[15:0]$
影响状态： 无影响
说明： 长子程序调用，下一条指令的 PC 地址压栈，当前 PC 赋值为 K
周期数： 2

10.2.44 LGOTO

语法： LGOTO K
操作数： $0 \leq K \leq 65535$
操作： $K \rightarrow PC[15:0]$
影响状态： 无影响
说明： 长无条件跳转
周期数： 2

10.2.45 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

10.2.46 MOVAR

语法： MOVAR R
操作数： $0 \leq R \leq 511$
操作： $R \rightarrow ACC$
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

10.2.47 MOVR

语法： MOVR R
操作数： $0 \leq R \leq 511$
操作： $R \rightarrow ACC$
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

10.2.48 MOVRA

语法： MOVRA R
操作数： $0 \leq R \leq 511$
操作： $ACC \rightarrow R$
影响状态： 不影响
说明： 将 ACC 存到 R
周期数： 1

10.2.49 MULAR

语法： MULAR R

操作数： $0 \leq R \leq 511$
操作： $R * ACC \rightarrow HIBYTE:ACC$
影响状态： 不影响
说明： 寄存器 R 内容和 ACC 相乘，结果高 8 位存到寄存器 HIBYTE，低 8 位存到 ACC
周期数： 1

10.2.50 MULRA

语法： MULRA R
操作数： $0 \leq R \leq 511$
操作： $R * ACC \rightarrow HIBYTE:R$
影响状态： 不影响
说明： 寄存器 R 内容和 ACC 相乘，结果高 8 位存到寄存器 HIBYTE，低 8 位存到 R
周期数： 1

10.2.51 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

10.2.52 ORAI

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

10.2.53 ORAR

语法： ORAR R
操作数： $0 \leq R \leq 511$
操作： $R \text{ or } ACC \rightarrow ACC$
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 或操作，结果存到 ACC
周期数： 1

10.2.54 ORRA

语法： ORRA R
操作数： $0 \leq R \leq 511$
操作： $R \text{ or } ACC \rightarrow R$

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

10.2.55 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

10.2.56 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

10.2.57 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

10.2.58 RLA

语法： RLA
 操作数： 无
 操作： $C \leftarrow ACC[7]; ACC[7:1] \leftarrow ACC[6:0]; ACC[0] \leftarrow C$
 影响状态： C
 $C \quad R[7] \rightarrow C$
 说明： ACC 内容循环左移（带 C 标志）
 周期数： 1

10.2.59 RLAR

语法： RLAR R
 操作数： $0 \leq R \leq 511$
 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 循环左移（带 C 标志），结果存到 ACC
周期数： 1

10.2.60 RLR

语法： RLR R
操作数： $0 \leq R \leq 511$
操作： $C \leftarrow R[7]; R[7:1] \leftarrow R[6:0]; R[0] \leftarrow C$
影响状态： C
C R[7] → C
说明： 寄存器 R 循环左移（带 C 标志），结果存到 R
周期数： 1

10.2.61 RRA

语法： RRA
操作数： 无
操作： $C \rightarrow ACC[7]; R[7:1] \rightarrow ACC[6:0]; R[0] \rightarrow C$
影响状态： C
C R[0] → C
说明： ACC 循环右移（带 C 标志）
周期数： 1

10.2.62 RRAR

语法： RRAR R
操作数： $0 \leq R \leq 511$
操作： $C \rightarrow ACC[7]; R[7:1] \rightarrow ACC[6:0]; R[0] \rightarrow C$
影响状态： C
C R[0] → C
说明： 寄存器 R 循环右移（带 C 标志），结果存到 ACC
周期数： 1

10.2.63 RRR

语法：RRR R
操作数： $0 \leq R \leq 511$
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 循环右移（带 C 标志），结果存到 R
周期数：1

10.2.64 RSBCAR

语法：RSBCAR R
操作数： $0 \leq R \leq 511$
操作： $R - ACC - C \rightarrow ACC$
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：寄存器 R 内容与 ACC 减法（带 C 标志），结果存到 ACC
周期数：1

10.2.65 RSBCRA

语法：RSBCRA R
操作数： $0 \leq R \leq 511$
操作： $R - ACC - C \rightarrow R$
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 $0 \rightarrow DC$ ；否则 $1 \rightarrow DC$
Z 相减结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明：寄存器 R 内容与 ACC 减法（带 C 标志），结果存到 R
周期数：1

10.2.66 RSUBAR

语法：RSUBAR R
操作数： $0 \leq R \leq 511$
操作： $R - ACC \rightarrow ACC$
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：寄存器 R 内容和 ACC 相减，结果存到 ACC
周期数：1

10.2.67 RSUBRA

语法：RSUBRA R

操作数： $0 \leq R \leq 511$
操作： $R - ACC \rightarrow R$
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 $0 \rightarrow DC$ ；否则 $1 \rightarrow DC$
Z 相减结果为 0，则 $1 \rightarrow Z$ ；否则 $0 \rightarrow Z$
说明： 寄存器 R 内容和 ACC 相减，结果存到 R
周期数： 1

10.2.68 STOP

语法： STOP
操作数： 无
操作： $0 \rightarrow WDT$ ；进入休眠模式
影响状态： TO, PD
TO $1 \rightarrow TO$
PD $0 \rightarrow PD$
说明： 进入休眠模式，对看门狗定时器（WDT）清零
周期数： 1

10.2.69 SWAPAR

语法： SWAPAR R
操作数： $0 \leq R \leq 511$
操作： $R[7:4] \rightarrow ACC[3:0]$ ； $R[3:0] \rightarrow ACC[7:4]$
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

10.2.70 SWAPR

语法： SWAPR R
操作数： $0 \leq R \leq 511$
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

10.2.71 XORAI

语法： XORAI K
操作数： $0 \leq K \leq 255$
操作： $K \text{ xor } ACC \rightarrow ACC$
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

10.2.72 XORAR

语法： XORAR R
操作数： $0 \leq R \leq 511$
操作： $R \text{ xor } ACC \rightarrow ACC$
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

10.2.73 XORRA

语法： XORRA R
操作数： $0 \leq R \leq 511$
操作： $R \text{ xor } ACC \rightarrow R$
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 R
周期数： 1

第十一章 电气参数

11 电气参数

11.1 极限参数

参数	符号	值	单位
工作电压	Vdd	-0.3~6.0	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

11.2 直流特性参数

T=25°C

特性	符号	引脚	条件	最小	典型	最大	单位
工作电压	VDD		Fcpu=8MHz	3.6		5.5	V
			Fcpu=4MHz	2.4V		5.5	
			Fcpu=1MHz	1.8V		5.5	
			Fcpu=32768Hz/4	1.6V		5.5	
输入漏电	V _{leak}	所有输入引脚				1	uA
输入高电平	V _{ih}	所有输入引脚		0.7VDD			V
输入低电平	V _{il}	所有输入引脚				0.3VDD	V
上拉电阻 1	R _{pu1}	P00-P04 P10-P15	VDD=5V, Vin=VSS		50		Kohm
			VDD=3V, Vin=VSS		100		Kohm
上拉电阻 2	R _{pu2}	P16	VDD=5V, Vin=VSS		30		Kohm
			VDD=3V, Vin=VSS		30		Kohm
输出高电平 驱动电流	I _{oh}	所有输出引脚	Voh=VDD-0.6V	6			mA
输出低电平 驱动电流	I _{ol}	所有输出引脚	Vol=0.6V	10			mA
VREF 电压	V _{REF1}	P00/VREF	外部输入参考, VDD=5V	2		VDD	V
	V _{REF2}		内部 VDD 参考		VDD		V
	V _{REF3}		内部 4V 参考, VDD=5V	3.92	4	4.08	V
	V _{REF4}		内部 3V 参考, VDD=5V	2.94	3	3.06	V
	V _{REF5}		内部 2V 参考, VDD=5V	1.96	2	2.04	V
动态功耗	I _{ddc}	VDD	VDD=5V, Fcpu=1MHz		1.3		mA
			VDD=3V, Fcpu=1MHz		0.8		mA
			VDD=5V, Fcpu=32768Hz/4		35		uA
			VDD=3V, Fcpu=32768Hz/4		50		uA
休眠模式功耗	I _{sleep}	VDD	ADC 关闭, WDT 关闭, 高频振荡器		1		uA

			关闭, 低频振荡器关闭, 执行 STOP 指令				
HOLD 模式 1 功耗	I_{green1}	VDD	ADC 关闭, VDD=5V, 高频振荡开启, 执行 STOP 指令		300		uA
			ADC 关闭, VDD=3V, 高频振荡开启, 执行 STOP 指令		200		uA
HOLD 模式 2 功耗	I_{green2}	VDD	ADC 关闭, VDD=5V, 高频振荡关闭, 低频振荡开启, 执行 STOP 指令		10		uA
			ADC 关闭, VDD=3V, 高频振荡关闭, 低频振荡开启, 执行 STOP 指令		5		uA

11.3 ADC 特性参数

VDD=5V, T=25℃

特性	符号	条件	最小	典型	最大	单位
积分线性误差	ILE	VDD=VREF=5V FADC=4MHz			±2	LSB
微分线性误差	DLE	VDD=VREF=5V FADC=4MHz			±1	LSB
上限偏置误差	EOT	VDD=VREF=5V FADC=4MHz		±1	±3	LSB
下限偏置误差	EOB	VDD=VREF=5V FADC=4MHz		±1	±3	LSB
转换精度	ACC	VDD=VREF=5V FADC=4MHz			±3	LSB
		VDD=5V, VREF=2V FADC=4MHz			±16	LSB
转换时钟	FADC	VDD=5V			4	MHz
		VDD=3V			2	MHz
转换时间	T_{con}			59		1/FADC
ADC 输入电压	$V_{I_{AN}}$		0		VDD	V
ADC 输入阻抗	$R_{I_{AN}}$		2			Mohm
ADC 输入电流	$I_{I_{AN}}$	VDD=5V			10	uA
ADC 动态电流	I_{add}	VDD=5V AD 转换中		1	3	mA
ADC 静态电流	I_{ads}	VDD=5V ADEN=0		0.1	0.5	uA

11.4 交流电气参数

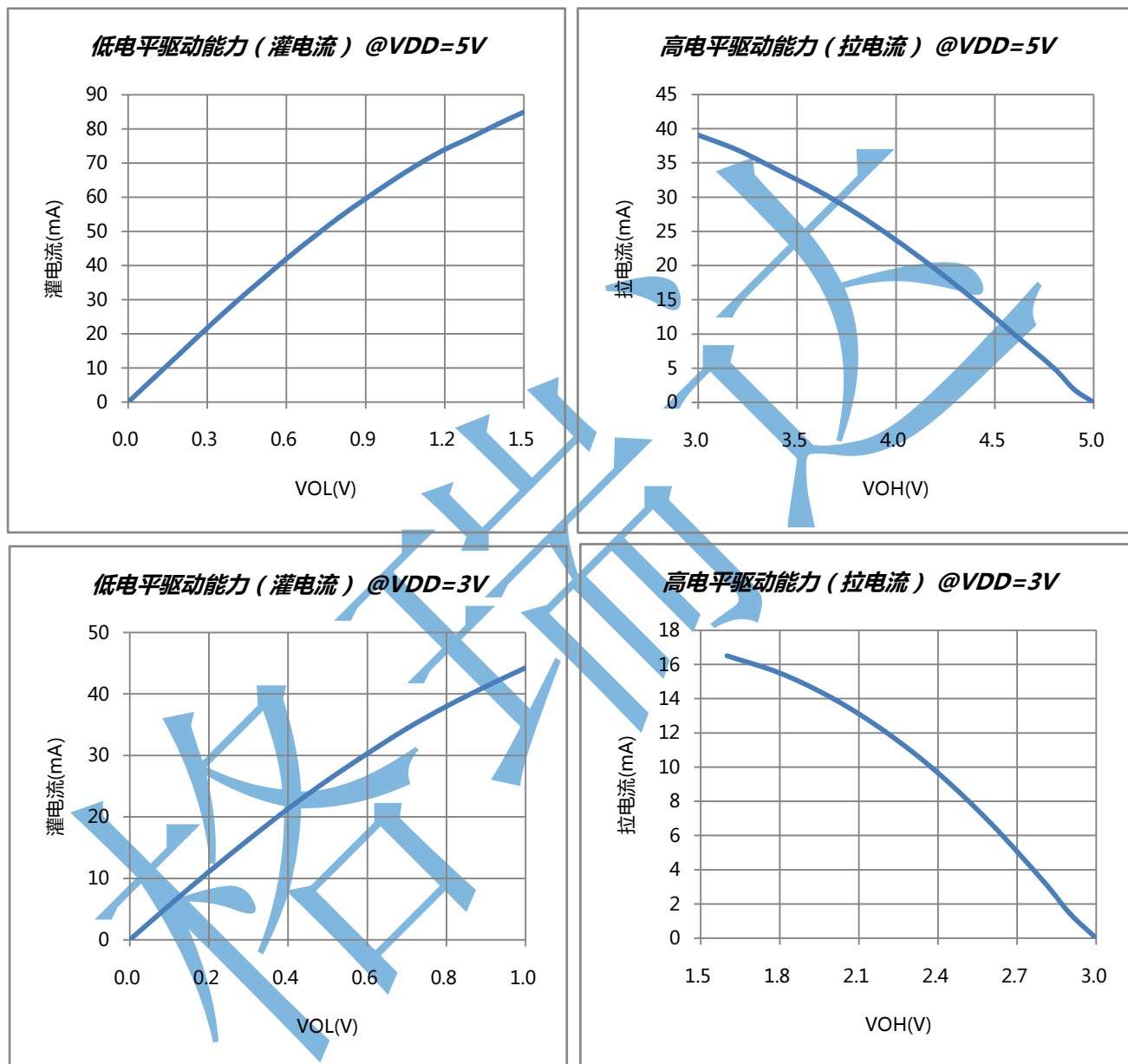
VDD=5V, T=25°C

特性	符号	条件	最小	典型	最大	单位
外部高频晶振频率	F _{osc1}		432K		16M	Hz
外部低频晶振频率	F _{osc2}			32768		Hz
内部高频 RC 振荡频率	F _{hrc1}	T=25°C VDD=5V	-1%		+1%	MHz
	F _{hrc2}	T=25°C VDD=2~5.5V	-1%		+1%	MHz
	F _{hrc3}	T=-40°C~85°C VDD=5V	-2%		+2%	MHz
	F _{hrc4}	T=-40°C~85°C VDD=2~5.5V	-2.5%		+2.5%	MHz
WDT 振荡器频率	F _{wdt}	T=25°C VDD=5V	-20%	28	+20%	KHz
外部振荡器起振时间	T _{oxov}				20	ms

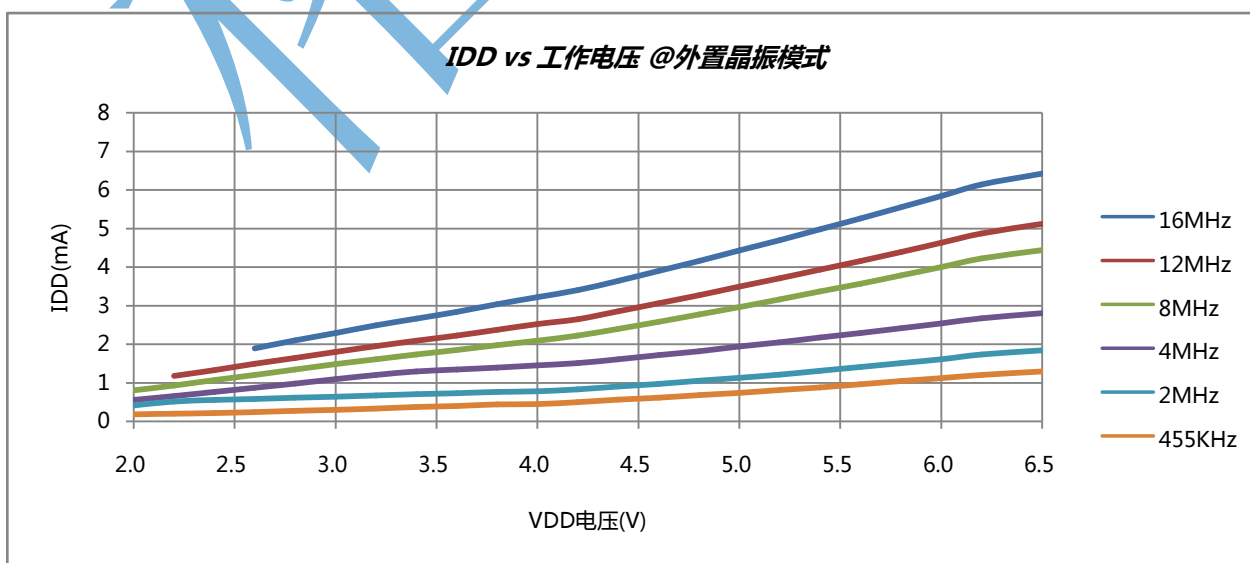
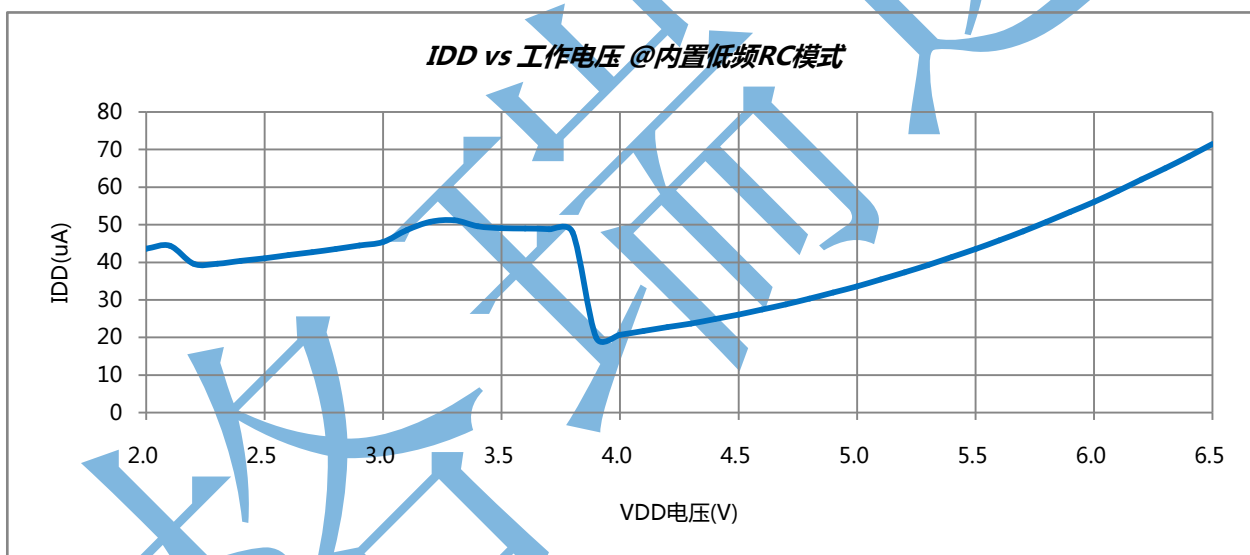
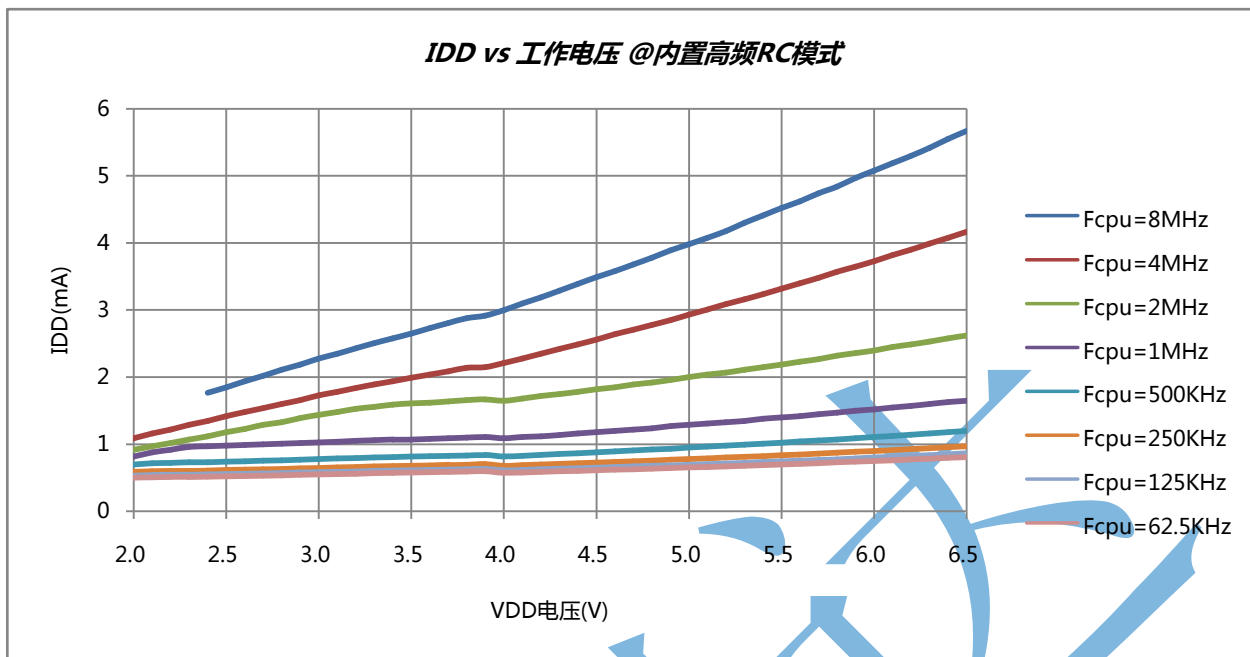
11.5 特性曲线

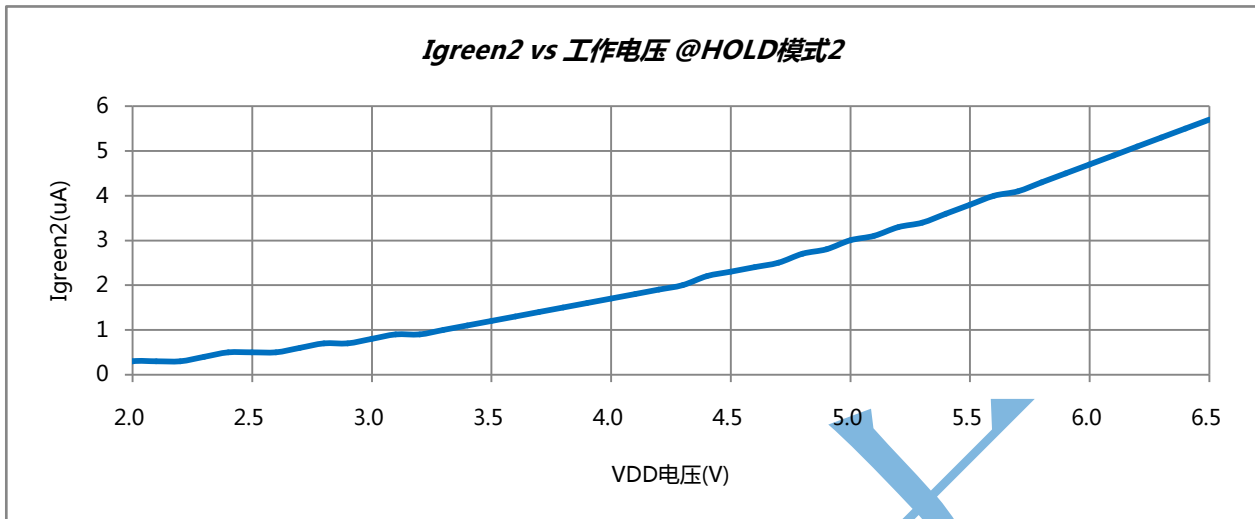
注：本节列出的特性曲线图仅作为设计参考，部分数据可能超出芯片额定的工作条件范围，为保证芯片能正常工作，请严格按照电气特性说明。

I/O口驱动能力特性曲线

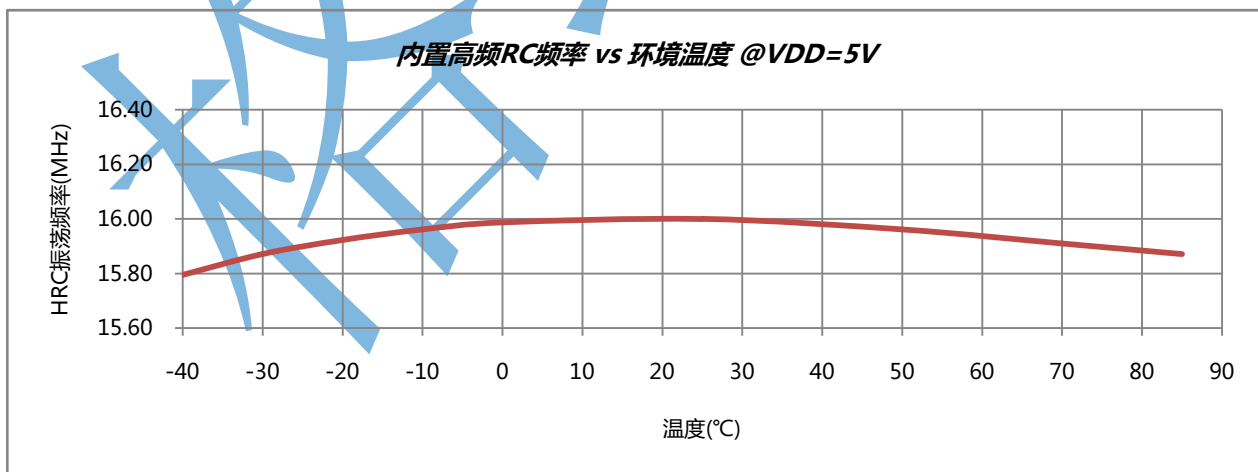
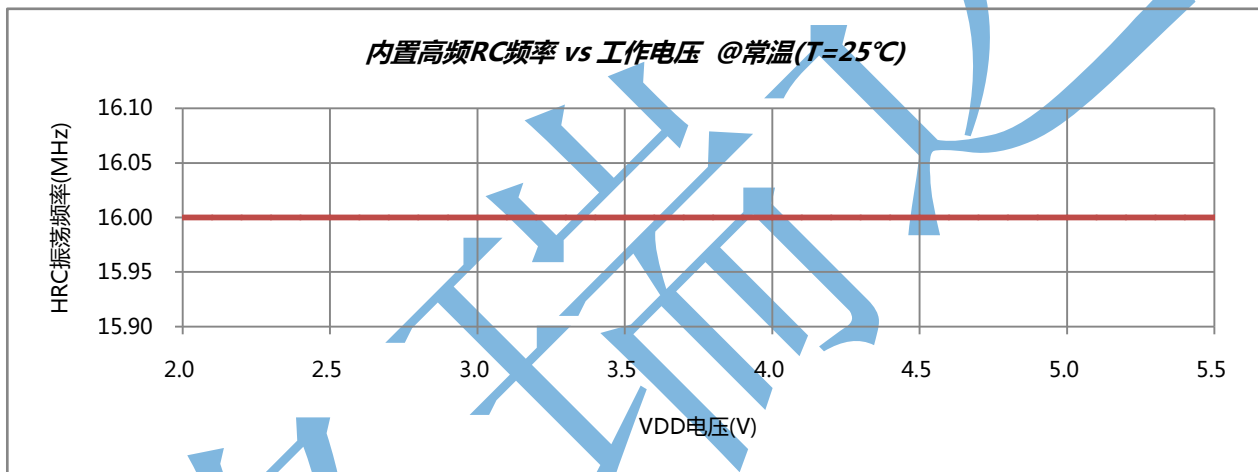


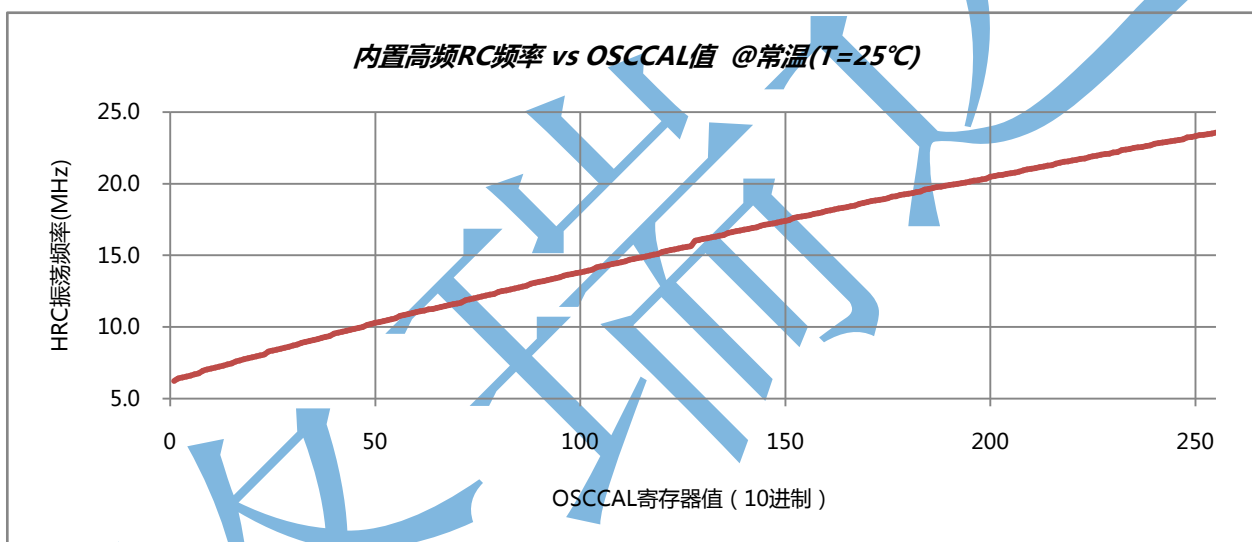
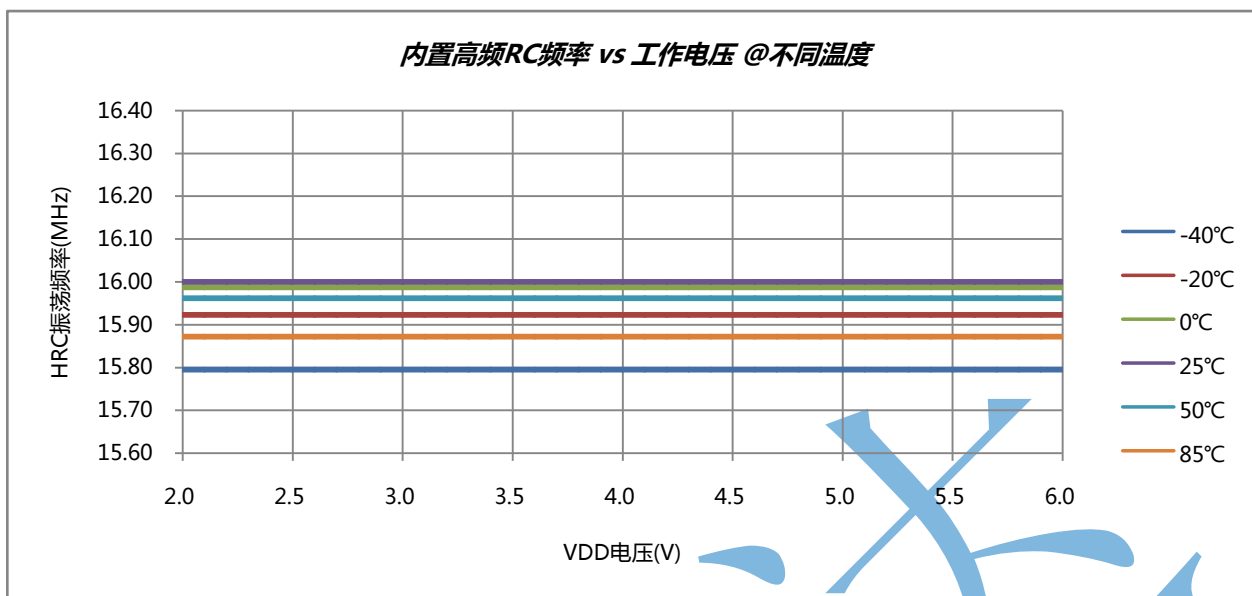
功耗特性曲线



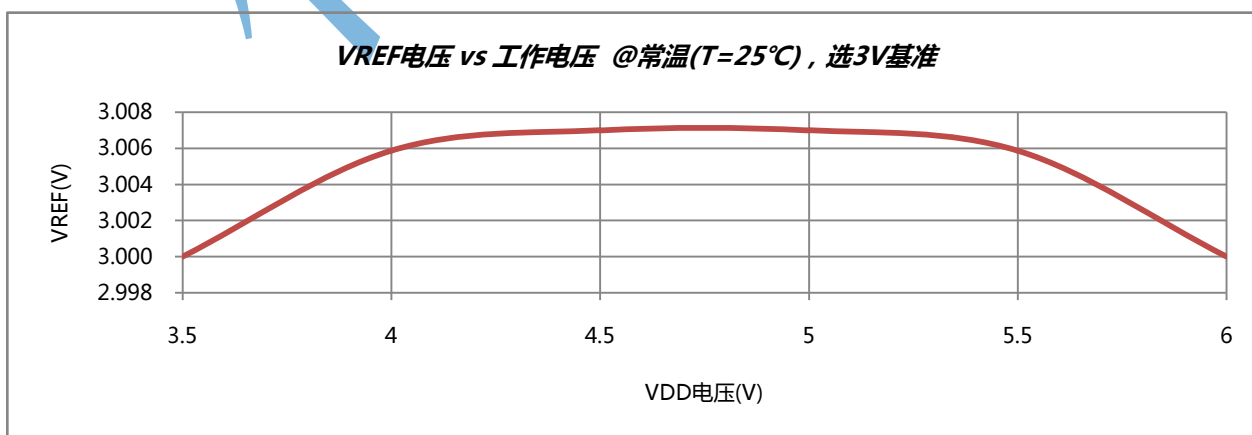


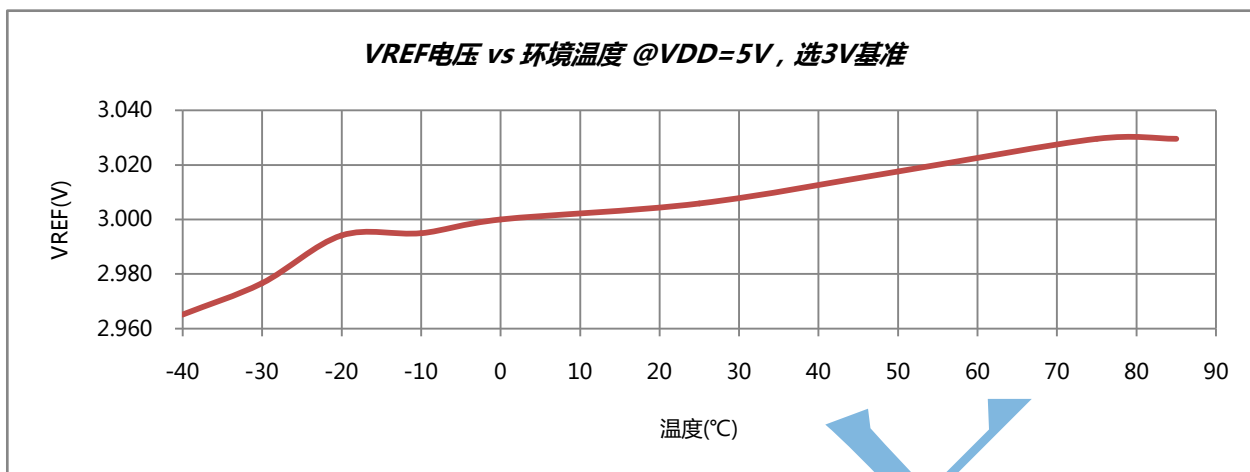
内部RC振荡频率特性曲线





ADC内部基准特性曲线

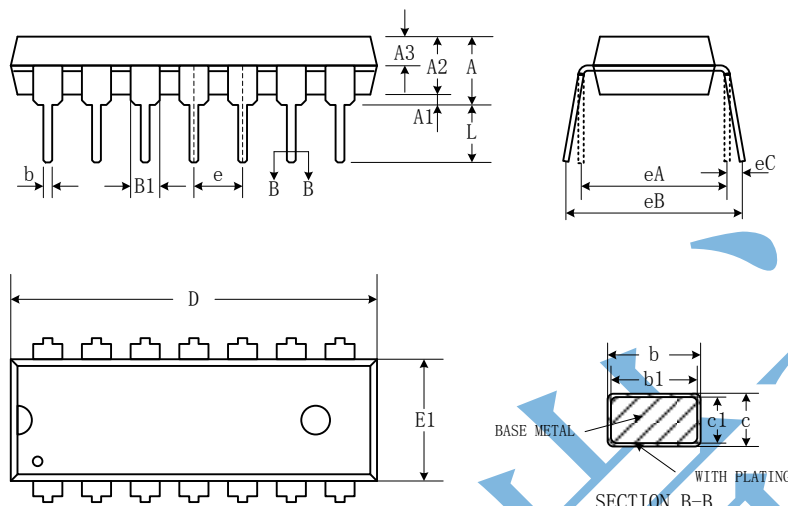




第十一章 封装外形尺寸

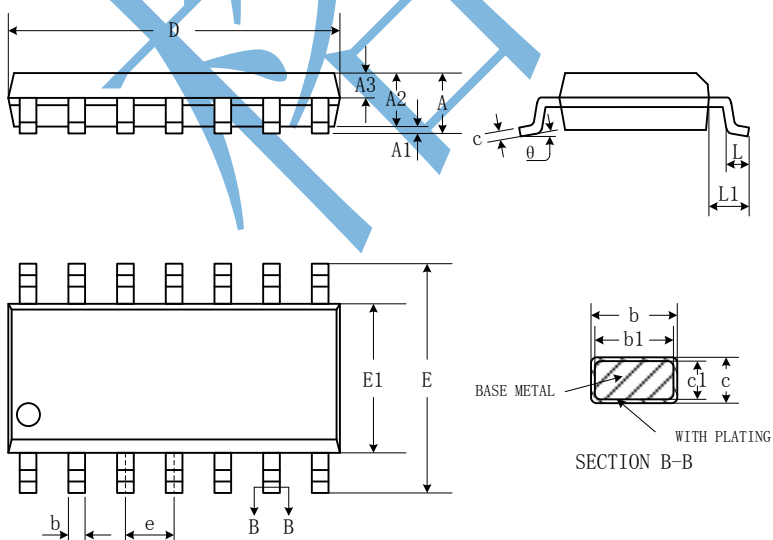
12 封装外形尺寸

12.1 DIP14



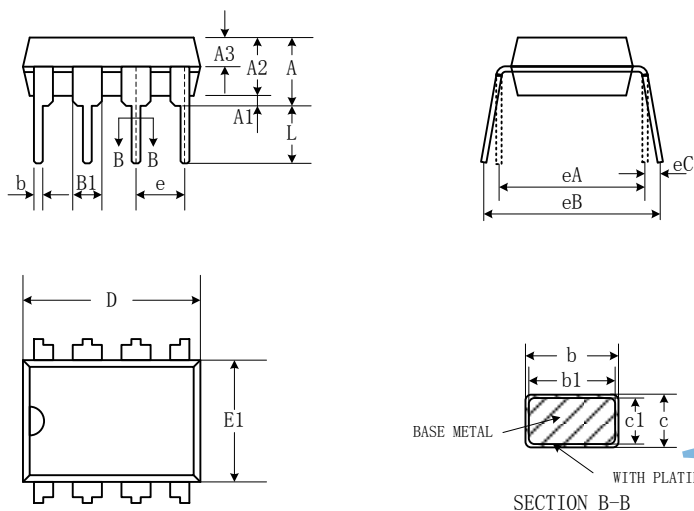
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	3.60	3.80	4.00
A1	0.51	—	—
A2	3.10	3.30	3.50
A3	1.42	1.52	1.62
b	0.44	—	0.53
b1	0.43	0.46	0.48
B1	1.52BSC		
c	0.25	—	0.31
c1	0.24	0.25	0.26
D	18.90	19.10	19.30
E1	6.15	6.35	6.55
e	2.54BSC		
eA	7.62BSC		
eB	7.62	—	9.50
eC	0	—	0.94
L	3.00	—	—

12.2 SOP14



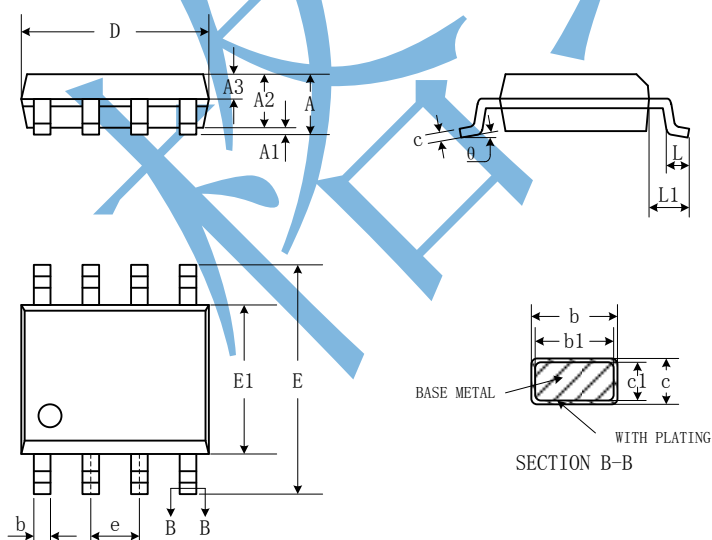
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.77
A1	0.08	0.18	0.28
A2	1.20	1.40	1.60
A3	0.55	0.65	0.75
b	0.39	—	0.48
b1	0.38	0.41	0.43
c	0.21	—	0.26
c1	0.19	0.20	0.21
D	8.45	8.65	8.85
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27BSC		
L	0.50	0.65	0.80
L1	1.05BSC		
θ	0	—	8°

12. 3DIP8



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	—	—

12. 4SOP8



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		
θ	0	—	8°

第十二章 订购

13 订购信息

下单规格	功能简述	芯片型号	封装
32P21A0P14	8-Bit 通用 AD 型单片机	32P21	DIP14
32P21A0S14	8-Bit 通用 AD 型单片机	32P21	SOP14
32P21A0P08	8-Bit 通用 AD 型单片机	32P21	DIP8
32P21A0S08	8-Bit 通用 AD 型单片机	32P21	SOP8
32P21A1P08	8-Bit 通用 AD 型单片机	32P21	DIP8
32P21A1S08	8-Bit 通用 AD 型单片机	32P21	SOP8
32P21A2P08	8-Bit 通用 AD 型单片机	32P21	DIP8
32P21A2S08	8-Bit 通用 AD 型单片机	32P21	SOP8

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