

# BL2220 MCU 用户手册



上海贝岭股份有限公司

## 第一章 概述

1.1 概述.....	(5)
1.2 管脚图.....	(5)
1.3 结构图.....	(6)
1.4 引脚描述.....	(6)
1.4.1 电源类引脚.....	(6)
1.4.2 输入输出类引脚.....	(6)
1.4.3 其它.....	(7)
1.5 掩膜选择.....	(7)
1.6 振荡器及复位电路.....	(7)
1.6.1 晶体振荡器.....	(7)
1.6.2 RC 振荡器.....	(8)
1.6.3 外部时钟.....	(8)
1.6.4 复位引脚.....	(9)

## 第二章 存贮器

2.1 存贮器.....	(10)
2.2 用户存储器.....	(10)
2.3 RAM.....	(11)
2.4 ROM.....	(11)

## 第三章 中央处理单元(CPU)

3.1 CPU 寄存器.....	(11)
3.1.1 累加器(A).....	(12)
3.1.2 变址寄存器(X).....	(12)
3.1.3 堆栈指针(SP).....	(12)
3.1.4 程序计数器(PC).....	(13)
3.1.5 状态寄存器(CCR).....	(13)
3.1.5.1 进位标志位(H).....	(13)
3.1.5.2 中断屏蔽位(I).....	(13)
3.1.5.3 负标志位(N).....	(14)
3.1.5.4 零标志位(Z).....	(14)
3.1.5.5 进位/借位标志位(C).....	(14)
3.2 算术/逻辑运算单元(ALU).....	(14)

## 第四章 中 断

4.1 中断源.....	(14)
4.1.1 中断系统屏蔽有关的寄存器.....	(15)

4.1.2 对各种中断分别进行控制的寄存器.....	(15)
4.1.3 中断请求引脚( $\overline{\text{IRQ}}$ ).....	(15)
4.1.4 键盘中断 PA7-PA0 引脚.....	(15)
4.1.5 定时器中断.....	(16)
4.2 中断处理.....	(16)

## 第五章 复位

5.1 外部复位.....	(17)
5.2 内部复位.....	(17)
5.2 COP 复位.....	(17)
5.3.1 简介.....	(17)
5.3.2 COP Watchdog 框图.....	(17)
5.3.3 使用.....	(17)
5.3.4 Low_Power Modes .....	(18)

## 第六章 低功耗工作方式

6.1 Stop 方式.....	(19)
6.2 Wait 方式.....	(19)

## 第七章 并行 I/O 口

7.1 I/O 口的功能 .....	(19)
7.2 PA 口 .....	(19)
7.2 PB 口 .....	(20)
7.2 PC 口 .....	(20)
7.2 PD 口 .....	(20)

## 第八章 定时器

8.1 定时器控制寄存器(TCR) .....	(20)
8.2 计数器.....	(21)
8.3 预定标器选择逻辑.....	(21)
8.4 定时器数据寄存器(TDR).....	(21)
8.5 溢出检测电路和中断电路.....	(21)

## 第九章 A/D 转换器

9.1 简介.....	(21)
9.2 A/D 转换器结构.....	(21)
9.3 A/D 转换器的寄存器功能.....	(21)

第十章 指令系统

10.1 寻址方式.....	(22)
10.1.1 隐含寻址.....	(22)
10.1.2 立即寻址.....	(22)
10.1.3 直接寻址.....	(22)
10.1.4 扩展寻址.....	(23)
10.1.5 无偏移量变址.....	(23)
10.1.6 8 位偏移量变址.....	(23)
10.1.7 16 位偏移量变址.....	(23)
10.1.8 相对寻址.....	(23)
10.2 指令类型.....	(23)
10.2.1 寄存器/存贮器指令.....	(23)
10.2.2 读---修改---写指令.....	(24)
10.2.3 跳/转指令.....	(24)
10.2.4 位操作指令.....	(25)
10.2.5 控制指令.....	(25)
10.3 指令查找表.....	(25)

第十一章 电气特性

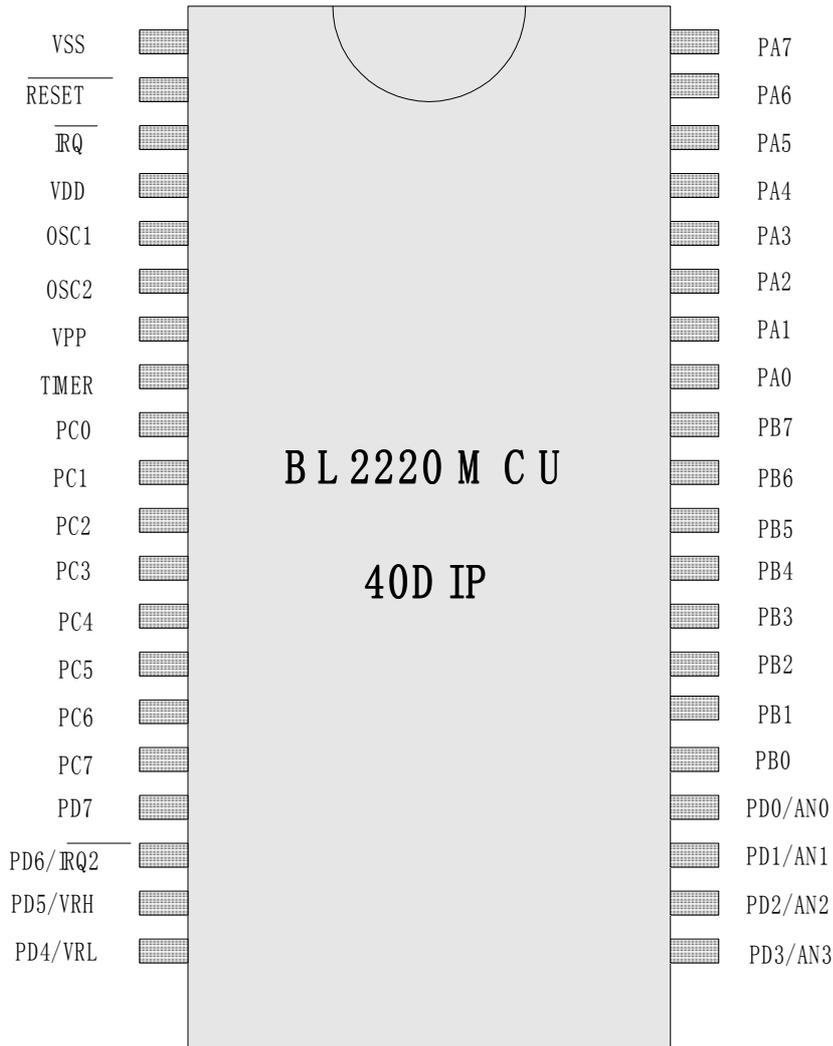
11.1 极限参数值.....	(31)
11.2 交流电特性及定义.....	(31)
11.3 直流电特性及定义.....	(32)

## 第一章 概述

### 1.1 概述

BL2220SR3 是上海贝岭股份有限公司开发的高速低功耗 8 位微控制器单元(MCU)。含片内存贮的 BL2220SR3 包括 3840 个字节用户 ROM 和 128 个字节用户 RAM<sup>®</sup>采用 HCMOS 工艺技术制造。BL2220SR3 可完全替代美国 MOTOROLA 公司的 8 位微控制器 MC68HC05SR3，并在此基础上增加了正常运行看门狗功能

### 1.2 管脚图



### 1.3 结构图

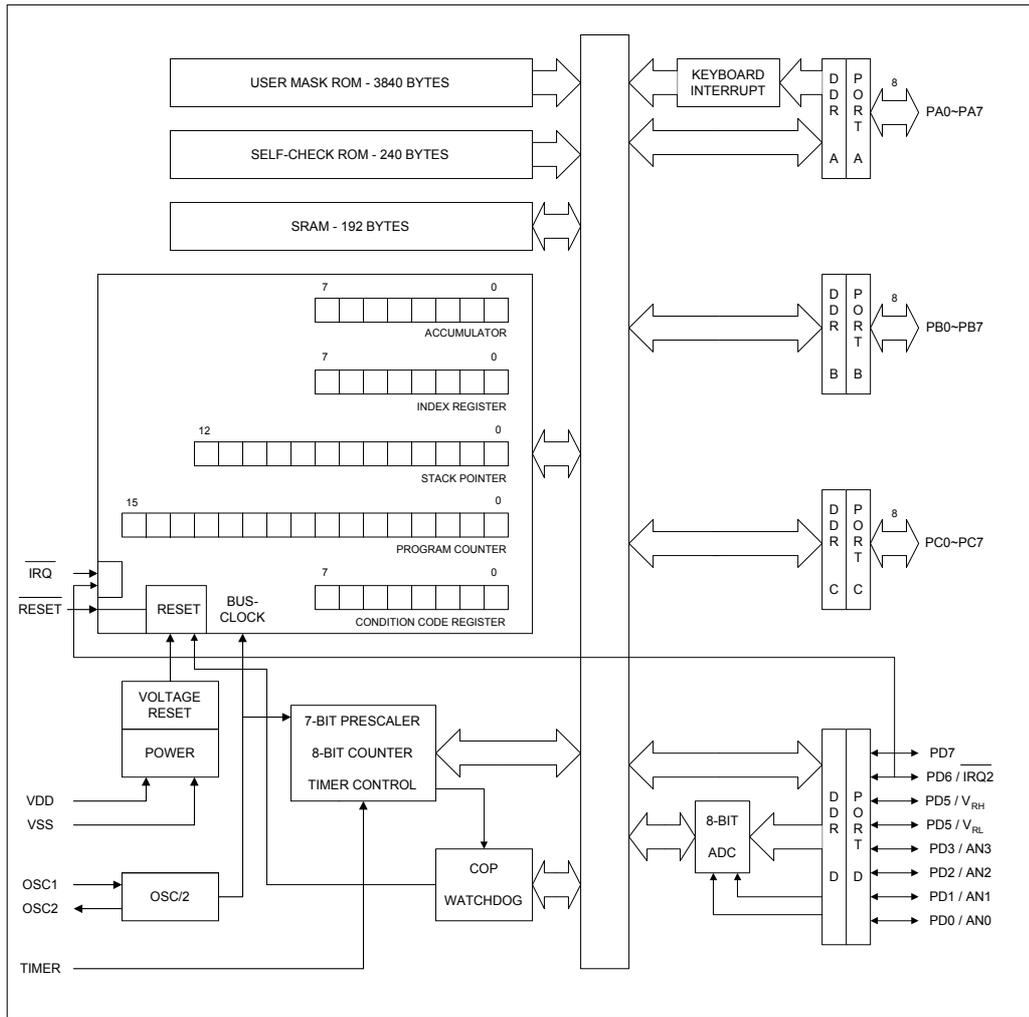


图 1-1 BL2220 结构图

## 1.4 引脚描述

### 1.4.1 电源类引脚

VSS, 引脚 1: 地

VDD, 引脚 4: 3 ~ 5.5 V

VPP, 引脚 7: 用户模式——VDD ~ VSS, 自检模式——2\*VDD

### 1.4.2 输入输出类引脚

PA0 ~ PA7, 引脚 33 ~ 40: PA 输入输出端口

PB0 ~ PB7, 引脚 25 ~ 32: PB 输入输出端口

PC0 ~ PC7, 引脚 9 ~ 16: PC 输入输出端口

PD0 ~ PD7, 引脚 24 ~ 17: PD 输入输出端口

其中:

PD0 ~ PD3: 兼作 A/D 输入口

PD4: A/D 参考地, VRL

PD5: A/D 参考电源, VRH

PD6: 外部中断 IRQ2 非的输入端

### 1.4.3 其它类

RESET 非,            引脚 2: 复位端  
 IRQ 非,            引脚 3: 外部中断 IRQ 非的输入端  
 OSC1, OSC2,      引脚 5, 6: 振荡器引脚  
 TIMER,            引脚 8: 定时器外部信号输入端

### 1.5 掩膜选择

BL2220 提供以下掩膜选择:

- 可以选择晶体/陶瓷振荡器或阻容网络(RC)振荡器, 缺省为晶振。
- 接通电源时的复位延时, 可以有 8 种选择, 如下表所示。缺省为 4096 个总线时钟周期。

可选延时	延时(时钟周期)
1	256
2	512
3	1024
4	2048
5	4096
6	8192
7	16384
8	32768

- 接通电源复位 慢方式 (总线频率位正常的 1/16)。缺省为屏蔽 慢方式。
- watchdog 复位。是否加电就存在的选择。

### 1.6 振荡器及复位电路

OSC1 和 OSC2 引脚用来控制连接在片振荡器。

根据掩膜时的选择, 振荡器可以选择以下几种:

- 晶体
- 阻容网络
- 外部时钟信号

内部振荡器频率是  $f_{osc}$ 。MCU 将内部振荡器的输出频率进行 2 分频产生内部时钟频率  $f_{op}$ 。

#### 1.6.1 晶体振荡器

掩膜时可进行晶体和 RC 振荡器选择, 在 OSC1 与 OSC2 引脚之间连接一个晶体振荡器可以驱动片内振荡器。图 1-2 给出了典型的 AT 切割、电阻并联的晶体振荡器电路连接方式。参照晶振供应商的建议选用合适的晶振参数, 因为它决定了外接部件提供的起振的可靠性和振荡的稳定性。振荡电路中选择电容容量时应当考虑分布电容的影响, 为了减小输出失真, 晶振和电容应当安装在尽可能靠近 MCU 引脚的地方。

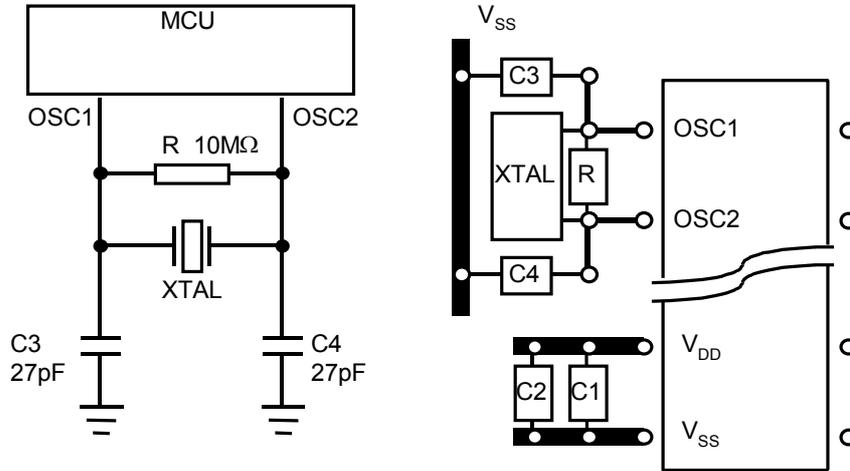


图 1-2 有反馈电阻的晶振连接

### 1.6.2 RC 振荡器

为了最大限度地降低成本，掩膜时可以选择 RC 振荡器，其应用有两种方式：一是将 OSC1 和 OSC2 连接。频率约为 2MHZ。二是 OSC2 和 VDD 之间串入电阻。电阻值为 30K 到 110K 变化时，频率约从 4MHZ 到 1.5MHZ 变化。供用户参考。

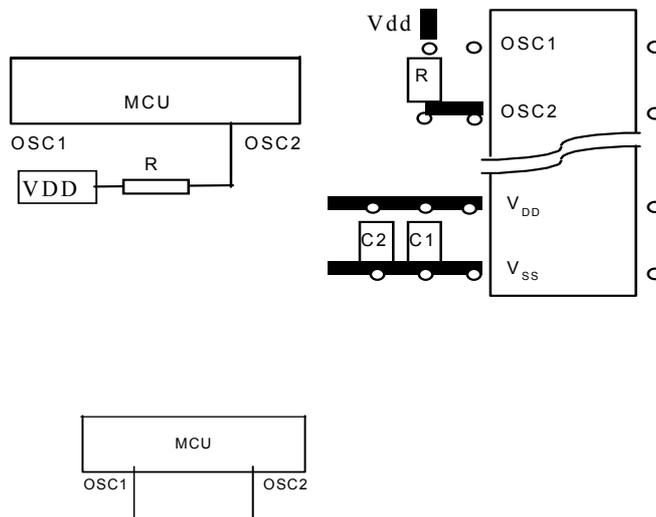


图 1-3 RC 振荡器的连接

### 1.6.3 外部时钟

在掩膜选用 RC 振荡器时，可以在 OSC1 脚输入其他兼容的 CMOS 器件产生的时钟信号，如图 1-4 所示，可以让 OSC2 脚悬空。

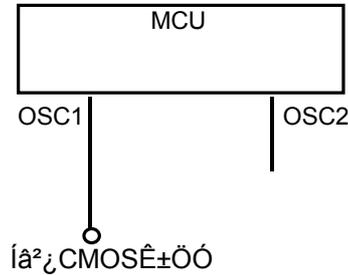


图 1-4 外部时钟的连接

#### 1.6.4 复位脚(RESET)

RESET 脚上的逻辑零可以强制 MCU 进入复位状态。详细信息请参阅 5.1.2 节外部复位。

## 第二章 存储器

### 2.1 存储器

#### BL2220 片内存储器的结构

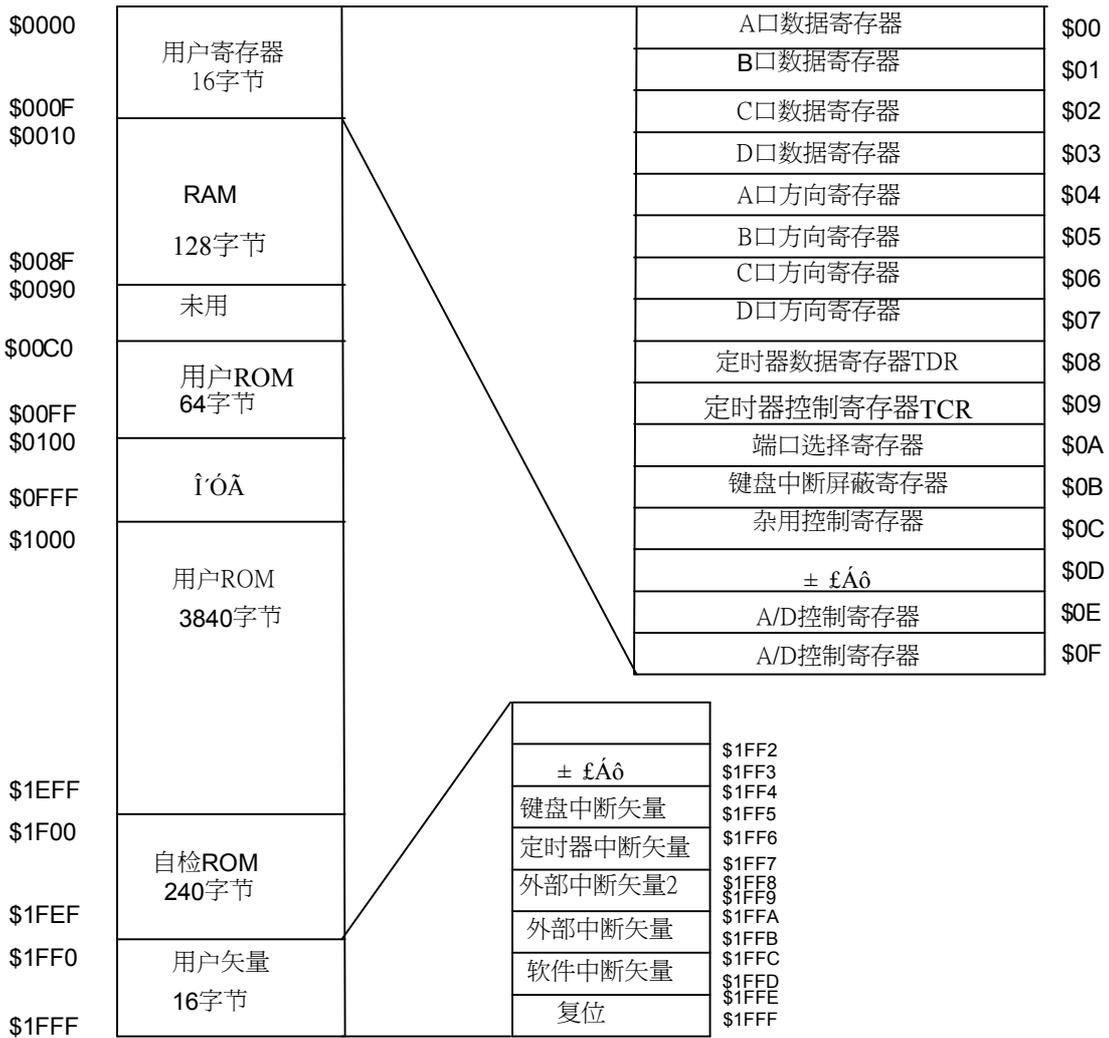


图 2-1 存储器结构图

如图 2-1 所示 CPU 可以寻址 8K 字节的存储空间。存储器的 ROM 区用来存程序指令、固定数据、用户定义矢量和中断服务程序，存储器的 RAM 区用来放置可变数据。用户寄存器映射在存储器中，这样 CPU 可以用与访问存储器其他区域相同的寻址方式来寻址用户寄存器空间。

## 2.2 用户存储器

存储器的前 16 字节地址即\$0001—\$000F，是用户寄存器区。它们分别为数据寄存器，方向寄存器，定时数据寄存器，定时控制寄存器，I/O 端口上拉电阻选择寄存器，键盘中断屏蔽寄存器，杂用控制寄存器，A/D 控制寄存器和 A/D 结果寄存器。

请参阅图 2-2。

	Bit 7	6	5	4	3	2	1	Bit 0	
\$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA

\$0001	Pb7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
\$0002	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
\$0003	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
\$0004	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0	DDRA
\$0005	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	DDRB
\$0006	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0	DDRC
\$0007	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0	DDRD
\$0008	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0	TDR
\$0009	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0	TCR
\$000A	PFS	COPE	PIL	PDP	PCP	PBP	PB1	PB0	POPR
\$000B	KBE7	KBE6	KBE5	KBE4	KBE3	KBE2	KBE1	KBE0	KBIM
\$000C	KBIE	0/KBIC	INTO	INTE	LVRE	SM	0/IRQ2F	IRQ2E	MCR
\$000D									不用
\$000E	COCO	ADRC	ADON			CH2	CH1	CH0	ADSCR
\$000F	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	ADRD

图 2-2 I/O 寄存器 \$0000-\$000F

### 2.3 RAM

从\$0010到\$008F的128个字节作为用户RAM。

从\$00C0到\$00FF的64个字节作为堆栈区使用。在响应一次中断时,CPU使用5个字节的堆栈RAM存贮CPU所有的寄存器内容,即PC进栈,A,X,CCR进栈。处理一个子程序时,CPU使用2个字节来存贮返回地址。当数据压入堆栈时堆栈指针的地址减1,当数据弹出堆栈时堆栈指针加1。在使用嵌套子程序和多级中断时要当心,在处理子程序和响应中断时对堆栈的操作不要超过64单元,否则会将堆栈内的数据覆盖掉。

### 2.4 ROM

存贮器的ROM区由两部分空间组成:

- \$1000-\$1EFF的3840个字节为用户ROM区
- \$1F00-\$1EEF的240个字节为自检ROM区
- \$1FF0到\$1FFF的16个字节为用户中断矢量ROM区

## 第三章 中央处理单元(CPU)

### 3.1 CPU 寄存器

图 3-1 给出了 5 个 CPU 寄存器。CPU 寄存器不是存贮器的一部分。

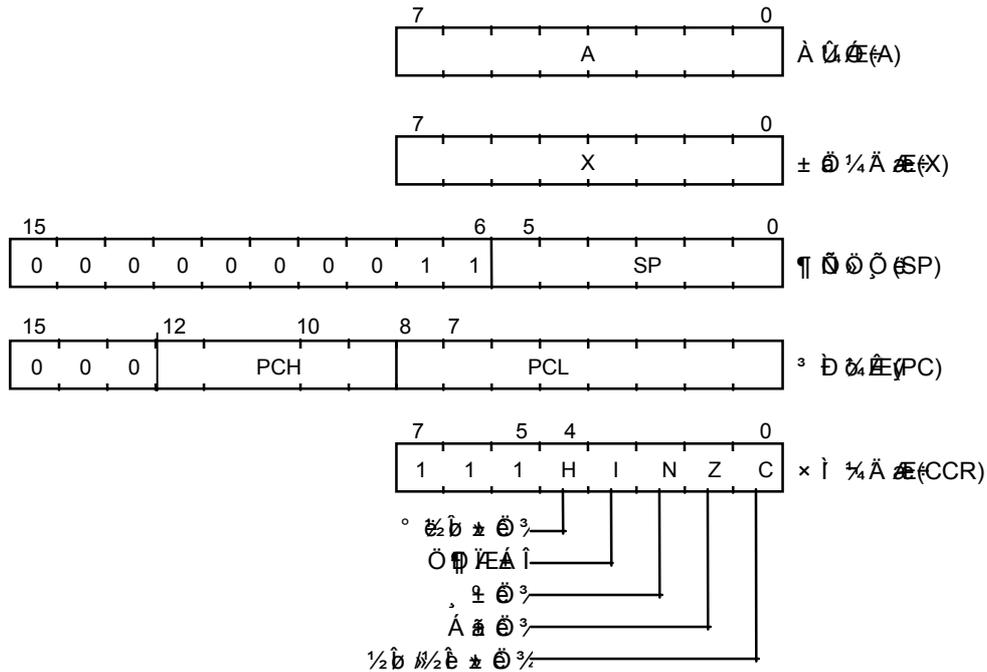
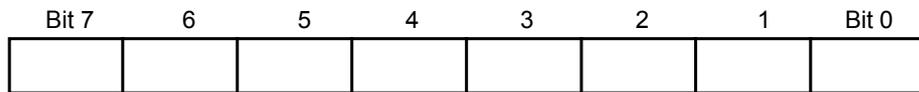


图 3-1 CPU 寄存器

### 3.1.1 累加器(A)

累加器是一个通用的 8 位寄存器。CPU 用累加器存放算术或数据处理时的第一操作数和结果。

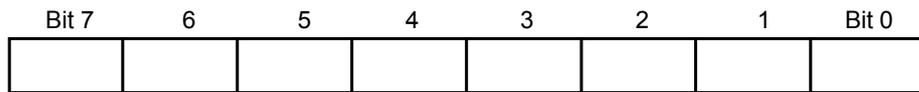


复位: 复位时不影响

图 3-2.累加器(A)

### 3.1.2 变址寄存器(X)

变址寄存器用于变址寻址方式，详细请参阅 9.1 节寻址方式。

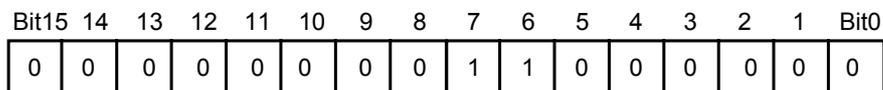


复位: 复位时不影响

图 3-3.变址寄存器(X)

### 3.1.3 堆栈指针(SP)

堆栈指针是一个 16 位的寄存器，用来放置堆栈中下一个空闲单元的地址。在执行 MCU 复位或堆栈复位操作命令(RSP)时，堆栈指针便自动设置栈底为\$00FF 处。当数据压入堆栈时堆栈指针的地址减 1，当数据弹出堆栈时堆栈指针加 1。



复位: 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

图 3-4.堆栈指针 (SP)

堆栈指针的高 10 位被固定设置为 0000000011, 因此堆栈指针的地址范围是 \$00C0 到 \$00FF。

如果子程序或中断处理程序使用的堆栈空间超过 64 个字节, 堆栈指针循环置为 \$00FF, 从而用新的数据将前面存贮的数据覆盖掉, 这可能引起系统工作的崩溃调用一次子程序使用 2 个堆栈单元, 而每一级中断要使用 5 个堆栈单元。

### 3.1.4 程序计数器(PC)

程序计数器是一个 16 位的寄存器, 用来放置即将执行的下一条指令的地址。程序计数器的高 3 位固定设置为 000。

正常情况下, 程序计数器的地址会随着每次取指操作的进行自动增大到下一个单元。在进行跳转、分支和中断操作时程序计数器便装入相应下一个要执行的单元的地址。

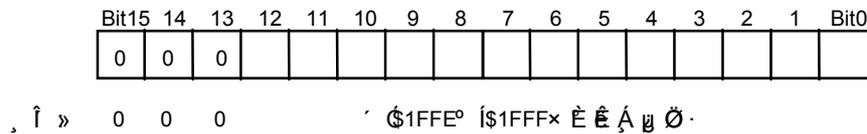


图 3-5.程序计数器(PC)

### 3.1.5 状态寄存器(CCR)

状态寄存器是一个 8 位的寄存器, 其中高三位被固定设置为 111。状态寄存器由一个中断屏蔽位和四个用来指示刚刚执行完的指令的结果的标志位。下面分别介绍状态寄存器功能。

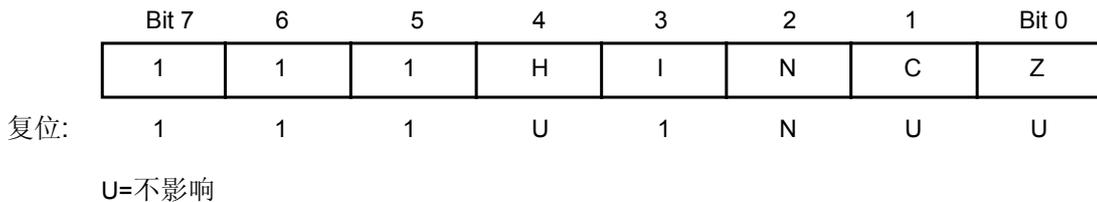


图 3-6.状态寄存器(CCR)

#### 3.1.5.1 半进位标志(H)

当执行加法 ADD 或带进位加法 ADC 指令时, 如果在位三和位四之间发生了进位 CPU 设置半进位标志。执行二进制编码的十进制数(BCD)运算时需要这个标志位

#### 3.1.5.2 中断屏蔽位(I)

中断屏蔽位用来锁存中断。当中断屏蔽位是逻辑零时, 允许中断请求, CPU 将寄存器的内容存贮到堆栈, 设置中断屏蔽位, 然后读取中断矢量。当中断屏蔽位被置位时, 所有的中断请求被锁存。正常情况下, 当中断屏蔽位被清零后, CPU 会尽快处理被锁存的中断。

当执行中断返回指令(RTI)时, CPU 寄存器的内容从堆栈中弹出, 中断屏蔽位恢复到清零状态。以后每次复位时, 中断屏蔽位被置位且只能通过软件指令来清零。

### 3.1.5.3 负标志位(N)

当一个算术操作、逻辑操作或数据传送操作的结果为负数时, CPU 设置负标志位。

### 3.1.5.4 零标志位(Z)

当一个算术操作、逻辑操作或数据传送操作的结果为\$00 时, CPU 设置零标志位。

### 3.1.5.5 进位/借位标志位(C)

当执行加法操作使得累加器内的第 7 位产生进位或执行减法操作需要借位时, CPU 设置进位/借位标志。有些逻辑操作和数据传送操作也会影响到进位/借位标志。

## 3.2 算术/逻辑运算单元(ALU)

算术/逻辑运算单元用于执行算术运算和逻辑运算。乘法定义成一个单独的操作,是在 ALU 内通过一系列加法和移位操作来完成的, 乘法指令(MUL)需要 11 个内部时钟周期来完成这一操作。

## 第四章 中断

### 4.1 中断源

可以产生中断请求的中断源有:

- 复位中断
- 软件中断(SWI)
- 外部中断请求引脚(IRQ, PD6/IRQ2)
- 定时器中断
- 键盘中断(PA7~PA0)

注: 前两种中断不可屏蔽。

中断请求使系统临时停止执行正常的程序流程而去处理特殊事件。中断并不停止正在执行的指令, 而是登记发生的中断, 直到执行完现行的指令。中断处理时系统自动将 CPU 寄存器的数据压入堆栈并将用户定义的矢量地址装入程序计数器。

#### 4.1.1 中断系统屏蔽有关的寄存器

将状态寄存器 CCR 的中断屏蔽位(I)可以禁止外部中断。

	Bit 7	6	5	4	3	2	1	Bit 0
	1	1	1	H	I	N	C	Z
U:	1	1	1	U	1	N	U	U

U=不影响

I=0 允许中断

I=1 禁止中断

### .状态寄存器(CCR)

**4.1.2 对各种中断分别进行控制的寄存器**
**MCR(\$000C)**

Bit 7	6	5	4	3	2	1	Bit 0
KBIE	KBIC	INTO	INTE	LVRE	SM	IRQ2F	IRQ2E
0	0	0	1	0	0	0	0

**4.1.3 中断请求引脚(IRQB)**

中断请求引脚(IRQB)上的中断信号可以锁存外部中断请求。每次当 CPU 执行完当前的指令时便会检测 IRQ 锁存器内的信息。如果 IRQ 锁存器被置位, CPU 继续检测状态寄存器(CCR)中的中断屏蔽位(I)以及中断请求状态和控制寄存器中的 IRQE 位。如果 I 位被清零而 IRQE 位被置位, CPU 中断程序的执行顺序。CPU 在存取中断矢量时, 将 IRQ 锁存器清零, 以便在处理中断服务子程序时, 能够锁存其他的外部中断请求。只要在中断返回前中断屏蔽位被清零, CPU 能够识别新的中断请求。

外部中断输入的触发方式可在掩膜时选择。IRQB 引脚可以设定为仅负跳变触发或低电平和负跳变触发。

当掩膜时 IRQ 引脚设定为电平和边沿触发方式时, IRQ 引脚上的一个低电平和负跳变信号便可以锁存一个中断请求。IRQB 引脚的这种触发方式允许用户把几个中断源用多根线"或"的方式接到 IRQ 脚上, 这样, 只要任一中断源让 IRQ 脚为低电平, 中断请求便可以锁存起来。

当软件对 IRQB 引脚设定为边沿触发方式时, IRQB 引脚上的一个负跳变信号便可以锁存一个中断请求。只有当 IRQ 引脚的电平恢复到逻辑 1 以后再次变为逻辑零时, 下一次的中断请求才可能被锁存起来。

**4.1.4 键盘中断 PA7-PA0 引脚**
**KBIM(\$000B)**

Bit 7	6	5	4	3	2	1	Bit 0
KBE7	KBE6	KBE5	KBE4	KBE3	KBE2	KBE1	KBE0

键盘中断请求和三个主要因素有关:

- 1.KBIE 位
- 2.KBE 位
- 3.PA 端口引脚的状态, 当引脚的状态从高电平变成低电平时, 表示有中断要求。

键盘中断应用注意: 建议首先使 PORTA 端口为高, 然后再设置 \$0B 和 \$0C 寄存器, 避免误判。

**4.1.5 定时器中断**

定时器中断请求和三个条件有关：

- 1 TIM 位：中断屏蔽标志，TIM=1 屏蔽中断，TIM=0 允许中断。
- 2 TIF 位：中断请求标志，TIF=1 有中断请求，TIF=0 无中断请求。
- 3 定时器数据寄存器复位时为 FF，实为减计数器，当结果为“0”时，置 TIF=1

#### 4.2 中断处理

进行中断服务时，CPU 会作出下述响应：

- CPU 寄存器顺序存贮在堆栈中，如图 4-3 所示
- 状态寄存器中的 I 位置位，以防其他中断
- 将下述中断矢量处的内容装入程序计数器内：

中断返回指令 RTI 使 CPU 通过退栈恢复寄存器的内容，如图 4-1 所示。

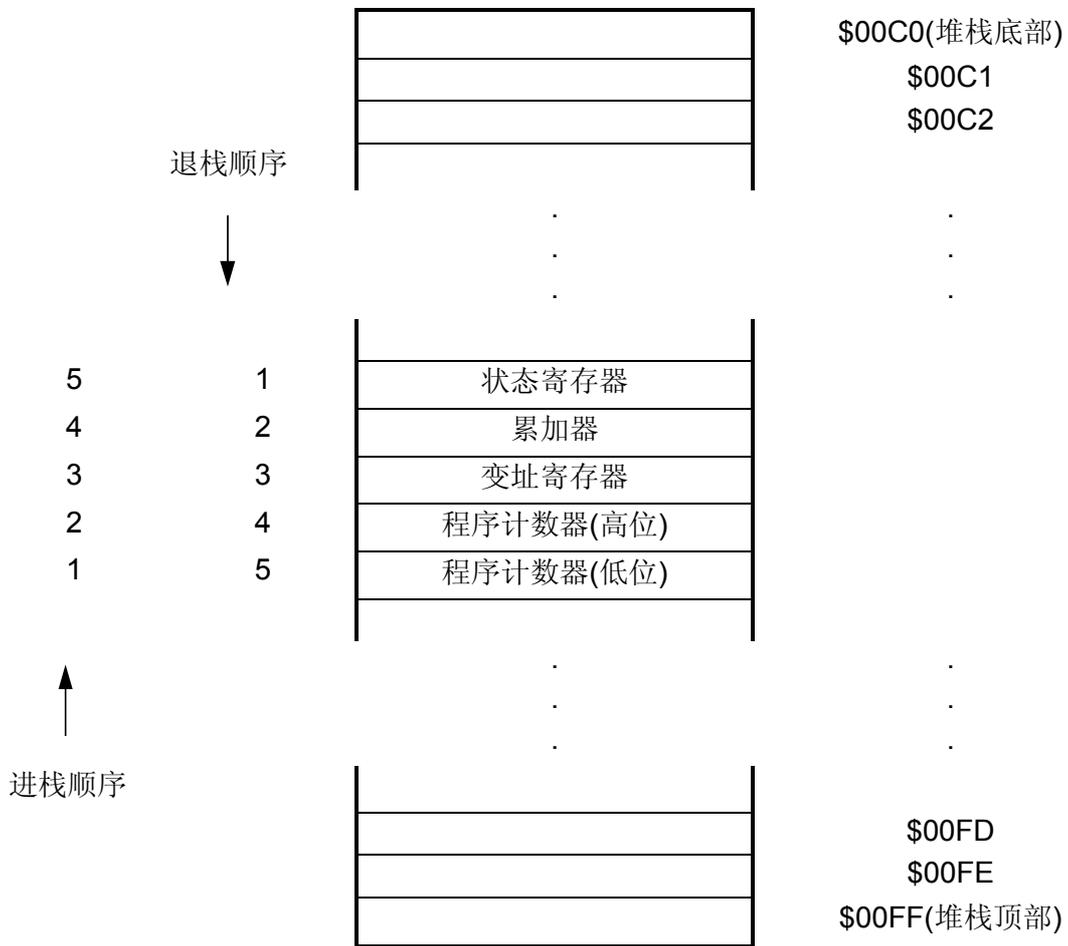


图 4-1 堆栈次序

### 第五章 复位

#### 5.1 外部复位—由外部引脚 RESET 输入低电平实现的。

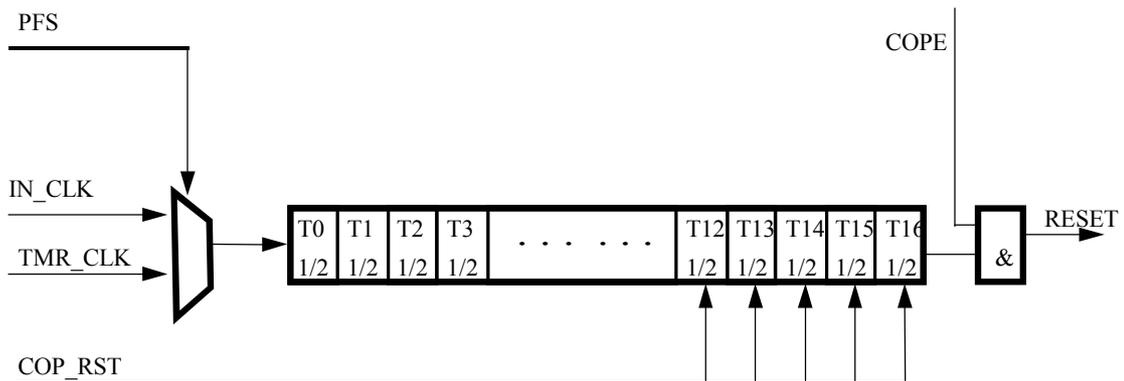
5.2 内部复位—有两种：一种是接通电源复位，另一种是低电压复位。

### 5.3 COP 复位 - WATCHDOG

#### 5.3.1 简介

当软件发生错误时，COP WATCHDOG 就会使 MCU 复位。正确运行的软件会周期复位 WATCHDOG 而防止其计数溢出而复位 MCU。COP WATCHDOG 的功能是可编程选择的。

#### 5.3.2 COP Watchdog 框图



#### 5.3.3 使用

##### 5.3.3.1 掩膜选择

通过掩膜选择可以设置 WATCHDOG 自 POWER-ON 开始就始终起作用或是可编程的。

(A) 当掩膜选择为“1”时，WATCHDOG 自 POWER-ON 起就始终起作用；

(B) 当掩膜选择为“0”时，WATCHDOG 是否起作用是可编程选择的。

##### 5.3.3.2 状态的设置

地址为 \$000A 的寄存器 POPR 的最高两位可以选择 WATCHDOG 的输入时钟信号和 WATCHDOG 是否起作用。

#### POPR

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	reset
\$000A	PFS	COPE	PIL	PDP	PCP	PBP	PB1	PB0	\$00

PFS ----- Pre\_Frequency\_Select

1 (set) ----- Watchdog 的输入时钟信号为 TIMER 中 PRESCLR 分频输出信号

TMR\_CLK;

0 (clear) ----- Watchdog 的输入时钟信号为内部总线时钟 IN\_CLK，频率为 2MHZ，此时 Watchdog 的溢出时间为 64mS。

#### PFS = “1”时 Watchdog 的溢出时间

PRE2	PRE1	PRE0	分频率	溢出时间 (mS)
------	------	------	-----	-----------

0	0	0	1	64
0	0	1	2	128
0	1	0	4	256
0	1	1	8	512
1	0	0	16	1024
1	0	1	32	2048
1	1	0	64	4096
1	1	1	128	8192

COPE ----- COP Enable  
 1 ( set ) ----- Watchdog 开始工作;  
 0 (clear) ----- Watchdog 不起作用。

### 5.3.3.3 复位 Watchdog

复位 Watchdog 的工作可以通过向地址\$1FF0 写“1”来产生 COP\_RST 信号完成, 例如使用指令

```
LDA #$01
STA $1FF0
```

### 5.3.4 Low\_Power Modes

STOP 和 WAIT 指令对 Watchdog 的影响如下:

**StopMode :** 由于内部时钟停止, 因此 Watchdog 保持原状态。当以中断退出 stop 状态时, 为防止在震荡器起振过程中 Watchdog 发出复位信号, 可在 STOP 指令前设置 Watchdog 不起作用, **Wait Mode :** WAIT 指令对 Watchdog 没有影响。

若要 WAIT 状态不被 Watchdog 唤醒, 应在 WAIT 指令前设置 Watchdog 不起作用。

## 第六章 低功耗工作方式

### 6.1 Stop 方式

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。

- 停止内部振荡器振荡。
- 清状态寄存器 I 位, 允许中断。
- RAM 内容保持不变。
- 所有的输入输出端口保留原态不变。
- 所有的内部操作全部停止。

以下情况使 MCU 退出 Stop 方式:

- 有外部复位信号从 RESET 输入
- 有外部中断信号从 IRQ 脚上输入(STOP 前已设置好状态) INTE=1
- 有键盘中断信号从 PA 端口输入(STOP 前已设置好状态) KBIE=1, E(i)=1

### 6.2 Wait 方式

执行 wait 指令 MCU 使进入等待低功耗方式, 同时对 MCU 产生以下影响:

- 停止 CPU 时钟
- 停止所有的处理器和内部总线的活动。
- 其他内部时钟保持作用。

- 允许定时器中断。
- 清状态寄存器 I 位，允许中断
- RAM 内容保持不变。
- 所有的输入输出端口保留原态不变。
- wait 指令不影响其它任何寄存器

以下条件将重新启动 CPU 时钟，使 MCU 退出 wait 方式，并进入正常工作方式

- 外部中断
- 键盘中断
- 定时器中断
- 复位唤醒

## 第七章 并行 I/O 口

### 7.1 I/O 口的功能

32 个并行双向 I/O 引脚组成四组 I/O 口(A,B,C,D\_Ú)，每个 I/O 引脚可由软件设置为输入或输出。数据由方向寄存器的内容，决定每一个 I/O 脚的数据方向。A, B, C, D 口 I/O 脚可软件编程有否上拉器件。

### 7.2 PA 口

PA 口是一个并行八位通用双向 I/O 口，可由软件设置有上拉电阻。有下列特性：

- 10mA 驱动能力(PA7---PA0)。
- 可兼作键盘中断输入口。

### 7.3 PB 口

PB 口是一个并行八位通用 I/O 口，可由软件设置有上拉器件。

- 10mA 驱动能力(PA7---PA0)。
- PB5~PB7 这三条引脚可以进行驱动能力选择。

### 7.4 PC 口

PC 口是一个并行八位通用 I/O 口，可由软件设置有上拉器件。

- 10mA 驱动能力(PA7---PA0)。

### 7.5 PD 口

PD 口是一个并行八位通用 I/O 口，可兼作 A/D 转换器输入端口。

- 10mA 驱动能力(PA7---PA0)。
- ADON=1, PD0~PD3 对应 CH0~CH3 模拟输入通道。

建议：PORTD 由数字口转为模拟口前，使 PD 口为"0"，然后再使 ADON=1，避免首次 A/D 转换时有误差。

## 第八章 定时器

### 8.1 定时器控制寄存器(TCR)

可读/写定时器的各种状态和控制位

\$0009

Bit 7	6	5	4	3	2	1	Bit 0
TIF	TIM	TCEX	TINE	PRER	PRE2	PRE1	PRE0
0	0	0	0	0	1	0	0

图 8-2 定时器状态和控制寄存器(TCR)

TIF—定时器中断请求标志

TIM—定时器中断屏蔽位

TINE 位—定时器输入选择和允许位。

PRER—予定标器的复位。

PRE2~PRE0—予定标器分频率选择位。

TCEX—定时器外部时钟控制位。它和定时器输入选择和允许位 TINE 相结合，以确定定时器输入信号的种类和方式。其作用如下表所示：

PRE2	PRE1	PRE0	分频率
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

### 8.2 计数器

计数器是一个 7 位减法计数器，它用于对定时器的输入计数信号执行计数。

计数器从 127~0 进行计数。它的计数信号由予定标器选择逻辑产生。

### 8.3 予定标器选择逻辑

予定标器选择逻辑是一个对计数器的计数信号执行分频的电路。

予定标器选择逻辑的输出送到定时器数据计数器 TDR 中去进行减法计数。

### 8.4 定时器数据寄存器 TDR

定时器数据寄存器 TDR 是一个可以读写的 8 位寄存器，它的内容是 8 位的定时器计数器的现行值。

### 8.5 溢出检测电路和中断电路

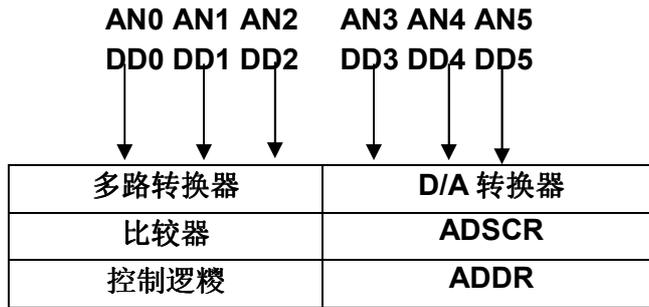
溢出检测电路和中断电路是产生定时器中断请求的电路。

## 第九章 A/D 转换器

### 9.1 简介

该转换器有一个 8 通道，8 位多路输入逐次逼近的 A/D 转换器，其中有四路接外部引脚的输入通道，另外四路内部通道作校正用。

### 9.2 A/D 转换器结构



### 9.3 A/D 转换器的寄存器功能

#### A) A/D 状态及控制寄存器 ADSCR 的功能

ADSCR

\$ 000E

Bit 7	6	5	4	3	2	1	Bit 0
COCO	ADRC	ADON			CH2	CH1	CH0
0	0	0	0	0	0	0	0

COCO—转换结束标志

ADRC—A/D 的 RC 振荡器控制位

ADON— A/D 转换器接通位，ADON=1。

作为数字 I/O 端口引脚在读状态时必须使 ADON=0，即禁止 A/D 转换。

CH2~CH0—输入模拟通道选择位

下面的任何一个过程都会对 COCO 位清零:

1. 对 A/D 状态及控制寄存器 ADSCR 执行写操作.
2. 一个新的 A/D 转化过程自动开始执行.
3. 对 A/D 数据寄存器 ADDR 执行都操作.

#### B) A/D 数据寄存器 ADDR(\$000F)的功能

A/D 数据寄存器 ADDR 是一个 8 位寄存器，它用于存放 A/D 转换的结果。

### 9.4 A/D 转化器的转化过程及控制逻辑

在一般的工作过程,A/D 转换器的过程如下:

首先,A/D 状态及控制寄存器 ADSCR 中的 CH0~CH2 位选择一个模拟输入通道.而 ADSCR 中的 ADRC 位选择 A/D 转换器的时钟源,即选 RC 振荡器还是 CPU 时钟.同时 ADSCR 中的 ADON 位用于开启 A/D 转换器的工作.

当 ADON=1 时,接通 A/D 转换器,故 A/D 转换器开始工作.

从 PD 端口被选中的模拟信号输入到 A/D 转换器中,和内部的 D/A 转换器的输出信号进行比较.内部 D/A 转换器以逐次逼近法形成数据,并将该数据转换成模拟信号.输出的模拟信号和从 PD 端口引脚输入的模拟信号送入到比较器的两个输入端,当两者相等时,则产生输出信号,停止 D/A 转换器的数据逼近操作,这时的现行数据就是转换结果.同时,把转换标志 COCO 置"1",把转换结果送入 A/D 数据寄存器 ADDR 中.到此,一次 A/D 转换过程结束.一次转换过程需要 32 个时钟周期.

使用 AD (ADON = 1) 的同时, 严禁将 PD 口任何一路 (包括 PD6 和 PD7) 作为数字口使用。如果要使 PD 口的任何一路作为数字口使用, 一定要使 ADON = 0。AD 采样时为了排除干扰, 提高可靠性, 建议多次采样取平均值。如果由于程序容量等问题, 只能采样一次, 这样会对系统的抗干扰能力有一定的影响。在不同的采样口之间切换时, 建议在完成一个通道的采样之后先关闭 AD (使 ADON = 0), 然后设置其他的采样端口, 同时打开 AD (使 ADON = 1)。若在通道切换之间不关闭 AD, 应至少在 32us 后采样 AD 转换结果。

## 第十章 指令系统

### 10.1 寻址方式

BL2220SR3CPU 有八种寻址方式, 可适应不同种数据访问方式, 寻址方式决定了 CPU 执行指令的方法。

- 隐含寻址
- 立即寻址
- 直接寻址
- 扩展寻址
- 无偏移量变址
- 8 位偏移量变址
- 16 位偏移量变址
- 相对寻址

#### 10.1.1 隐含寻址

隐含寻址指令中没有操作数, 如中断返回指令(RTI)和停止指令(STOP)一些隐含寻址指定是对 CPU 寄存器操作如设置进位标志(SEC)和累加器增加指令(INCA)隐含寻址不需要存贮器地址, 指令长度为一个字节

#### 10.1.2 立即寻址

立即寻址指令包含一个数值,用来对累加器和变址寄存器的值操作,立即寻址指令不需要地址,指令长度为二字节,第一字节为操作码,立即数为第二字节。

#### 10.1.3 直接寻址

直接寻址指令可以访问存储器中最低的 256 个字节，指令长度为二字节第一字节为操作码，第二字节是操作地址的低 8 位。在直接寻址指令中，CPU 直接把\$00 加到操作地址的高 8 位。

#### 10.1.4 扩展寻址

扩展寻址指令是一个三字节指令，可访问任何存储器单元。第一字节为操作码，第二和第三字节为操作地址的高位、低位。

#### 10.1.5 无偏移量变址

无偏移量变址指令长度为一字节。变址寄存器包含操作地址的低位，可访问存储器的前 256 字节 CPU 直接把\$00 加入高位，指令地址分布为\$0000~\$00FF。无偏移量变址指令经常用来移动表格指针或保存被频繁使用的 RAM 数据及 I/O 地址。

#### 10.1.6 8 位偏移量变址

8 位偏移量变址指令长两字节。可访问存储器前 511 个字节的存储单元 CPU 将变址寄存器中的无符号数加上操作码后续字节一个无符号数，它们之和作为操作地址，这种指令可访问\$0000---\$01FF。8 位偏移指令用于从一个有 N 个元素表中选择第 K 个元素。此表可从前 256 字节中任何一个地方开始，可扩展到\$01FF，K 数值存放于变址寄存器。指令中放表格的首地址。

#### 10.1.7 16 位偏移量变址

16 位偏移量变址指令长三字节，可访问所有存储器地址，CPU 变址寄存器中的无符号整数加上操作码后面的二字节无符号数，它们之和作为操作地址。16 位偏移量变址指令也同样用于 N 个元素表中选择第 K 个元素。

#### 10.1.8 相对寻址

只用于转移指令，发生转移时，操作码后的 8 位有符号整段加到 PC 上，否则顺序执行。偏移量为有符号整数，相对寻址范围为操作码地址开始的-128 到 +127 个字节。

### 10.2 指令类型

此 MCU 共有 64 条基本指令，分为五大类

- 寄存器/存储器指令
- 读-修改-写指令
- 跳转指令
- 位操作指令
- 控制指令

#### 10.2.1 寄存器/存储器指令

大多数为双字节指令，其中一字节操作数在累加器或变址寄存器中，CPU 通过一种寻址方式从存储器中找到另一操作数。表 10-1 列出了寄存器/存储器指令。

指 令	助 记 符
A 加存贮器和进位到 A	ADC
A 加存贮器到 A	ADD
A 和存贮器相“与”后送 A	AND
A 同存贮器位测试(逻辑“与”)	BIT
A 与存贮器算术比较	CMP
X 与存贮器算术比较	CPX
A 与存贮器异或后送 A	EOR
从存贮器装载 A	LDA
从存贮器装载 X	LDX
乘 $X:A \leftarrow X * A$	MUL
A 和存贮器相“或”后送 A	ORA
A 减存贮器及借位后送 A	SBC
A 存入存贮器	STA
X 存入存贮器	STX
A 减存贮器后送 A	SUB

### 10.2.2 读---修改---写指令

这类指令读取存贮器和寄存器中的值，修改它的内容，然后把修改的内容写回到存贮器和寄存器中。零或负测试 TST 指令，指定不属于读—修改—写指令。因为它并不把修改的内容写到原来的存贮器中，表 9-2 列出了读—修改—写指令。

表 10-2 读---修改---写指令

指 令	助 记 符
算术右移	ASR
清除	CLR
求反	COM
减少	DEC
增加	INC
逻辑左移	LSL
逻辑右移	LSR
(二进制)补码	NEG
带进位位循环左移	ROL
带进位位循环右移	ROR
负或零测试	TST

### 10.2.3 跳转指令（转移指令）

该类指令用于在某种条件具备时,使程序发生转移。该类指令均为相对转移指令,

均为双字节指令,除 BSR 为 6 时钟周期外,其他均为 3 个时钟周期。

**表 10-3 跳转指令 (转移指令)**

指令	功能	备注
20 BRA Rel	$(PC)+\$0002+Rel \rightarrow PC$	总是发生转移
21 BRN	$(PC)+\$0002 \rightarrow PC$	相当于 2 字节的 NOP
22 BHI Rel	若 $(C)+(Z)=0$ , $(PC)+\$0002+Rel \rightarrow PC$ 若 $(C)+(Z) \neq 0$ , 则继续	用于比较二进制无符号数的大小,大于则转移。
23 BLS Rel	若 $(C)+(Z)=1$ , $(PC)+\$0002+Rel \rightarrow PC$ 若 $(C)+(Z) \neq 1$ , 则继续	用于比较二进制无符号数的大小小于等于则转移
24 BCC Rel	若 $(C)=0$ , 则 $(PC)+\$0002+Rel \rightarrow PC$ , 否则继续	测试 C 值, 为零则转移。用于比较二进制无符号数,
25 BCS Rel	若 $(C)=1$ , 则 $(PC)+\$0002+Rel \rightarrow PC$ , 否则继续	测试 C 值, 为 1 则转移。
26 BNE Rel	若 $(Z)=0$ , 则 $(PC)+\$0002+Rel \rightarrow PC$ , 否则继续	测试 Z 值, 为零则转移。
27 BEQ Rel	若 $(Z)=1$ , 则 $(PC)+\$0002+Rel \rightarrow PC$ , 否则继续	测试 Z 值, 为 1 则转移。
28 BHCC Rel	若 $(H)=0$ , 则 $(PC)+\$0002+Rel \rightarrow PC$ , 否则继续	测试 H 值, 为零则转移。
29 BHCS Rel	若 $(H)=1$ , 则 $(PC)+\$0002+Rel \rightarrow PC$ , 否则继续	测试 H 值, 为 1 则转移。
2A BPL Rel	若 $(N)=0$ , 则 $(PC)+\$0002+Rel \rightarrow PC$ , 否则继续	测试 N 值, 为零则转移。
2B BMI Rel	若 $(N)=1$ , 则 $(PC)+\$0002+Rel \rightarrow PC$ , 否则继续	测试 N 值, 为 1 则转移。
2C BMC Rel	若 $(I)=0$ , 则 $(PC)+\$0002+Rel \rightarrow PC$ , 否则继续	测试 I 值, 为零则转移。
2D BMS Rel	若 $(I)=1$ , 则 $(PC)+\$0002+Rel \rightarrow PC$ , 否则继续	测试 I 值, 为 1 则转移。
2E BIL Rel	若 $IRQ_=0$ , 则 $(PC)+\$0002+Rel \rightarrow PC$ , 否则继续	测试外部中断引脚, 为零则转移。
2F BIH Rel	若 $IRQ_=1$ , 则 $(PC)+\$0002+Rel \rightarrow PC$ , 否则继续	测试外部中断引脚, 为 1 则转移。

#### 10.2.4 位操作指令

CPU 可对存储器前 256 字节任一可读/写位执行置位或清零。包括端口寄存器, 数据方向寄存器, 定时器寄存器, 以及地址在前 256 字节的 RAM。此 256 个单元的任一位的条件特征状态都可以由软件测试和分支。位操作指令用直接寻址方式。

**表 10-4 位操作指令**

指令	功能	字节	周期
BSET n,addr8	将 (addr8) 的第 n 位置 1	2	5
BCLR n,addr8	将 (addr8) 的第 n 位置 0	2	5
BRSET n,addr8, Rel	若 (addr8) 的第 n 位为 1, 则 $(pc)+\$0003+Rel \rightarrow PC$ 否则继续	3	5
BRCL n,addr8, Rel	若 (addr8) 的第 n 位为 0, 则 $(pc)+\$0003+Rel \rightarrow PC$ 否则继续	3	5

#### 10.2.5 控制指令

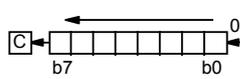
这些指令是寄存器参考指令, 在程序执行时, 用于控制处理器操作, 请参考下面控制指令表:

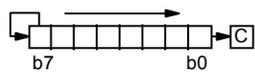
表 10-5 控制指令

指令	助记符
清进位	CLC
清中断屏蔽位	CLI
空操作	NOP
复位堆栈指针	RSP
从中断返回	RTI
从子程序返回	RTS
设进位位为 1	SEC
设中断屏蔽位为 1	SEI
停止	STOP
软件中断	SWI
传送 A 到 X	TAX
传送 X 到 A	TXA
等待	WAIT

## 10.3 指令查找表

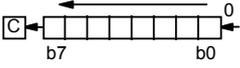
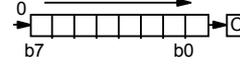
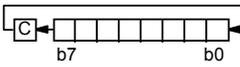
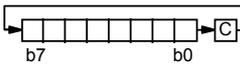
表 10-6 〈字母顺序列出了所有指令，及每个指令对状态寄存器的影响〉

指令	操作	功能	状态					寻址方式	操作码	操作数	周期
			H	I	N	Z	C				
ADC opr ADC opr ADC opr AD opr, X ADC opr,X ADC, X	带进位加	$A \leftarrow (A)+(M)+(C)$	*	-	*	*	*	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #opr ADD opr ADD opr ADD opr, X ADD opr,X ADD, X	无进位加	A (A) (M)	*	-	*	*	*	IMM DIR EXT IX2 IX1 IX	AB BB CB DB EB FB	ii dd hh ll ee ff ff	2 3 4 5 4 3
AND opr AND opr AND opr AND opr, X AND opr,X ADD, X	逻辑与	A (A) (M)	-	-	*	*	-	IMM DIR EXT IX2 IX1 IX	A4 B4 C4 D4 E4 F4	ii dd hh ll ee ff ff	2 3 4 5 4 3
LSL opr LSLA LSLX LSL opr,X LSL,X	逻辑左移		-	-	*	*	*	DIR INH INH IX1 IX	38 48 58 68 78	dd ff	5 2 2 6 5
ASR opr								DIR	37	dd	5

ASRA ASRX ASR opr,X ASR,X	算术右移		-	-	*	*	*	INH INH IX1 IX	47 57 67 77		2 2 6 5
BCC rel	进位位为 0 移	PC (PC)+2+rel, C=0	-	-	-	-	-	REL	24	rr	3
BCLRn opr	位清零	将 opr 中的 n 位清零	-	-	-	-	-	DIR(b0) DIR(b1) DIR(b2) DIR(b3) DIR(b4) DIR(b5) DIR(b6) DIR(b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS rel	进位为 1 移	PC ← (PC)+2+rel,C=1	-	-	-	-	-	REL	25	rr	3
BEQ rel	相等转移	PC ← (PC)+2+rel, Z=1	-	-	-	-	-	REL	27	rr	3
BHCC rel	半进位为 0 移	PC ← (PC)+2+rel,Z=0	-	-	-	-	-	REL	28	rr	3
BHCS rel	半进位为 1 移	PC ← (PC)+2+rel,H=1	-	-	-	-	-	REL	29	rr	3
BHI rel	大于转移	PC (PC)+2+rel, C=0	-	-	-	-	-	REL	22	rr	3
BCC rel	大于等于 转移	PC ← (PC)+2+rel,C=0	-	-	-	-	-	REL	24	rr	3
BIH rel	IRQ=1 移	PC ← (PC)+2+rel, IRQ=1	-	-	-	-	-	REL	2F	rr	3
BIL rel	IRQ=0 移	PC ← (PC)+2+rel, IRQ=0	-	-	-	-	-	REL	2E	rr	3
BIT #opr BIT opr BIT opr BIT opr,X BIT opr,X BIT,X	位测试累 加器 和存贮器 内容	(A) (M)	-	-	*	*	-	IMM DIR EXT IX2 IX1 IX	A5 B5 C5 D5 E5 F5	ii dd hh ll ee ff ff p	2 3 4 5 4 3
BCS rel	小于转移	PC ← (PC)+2+rel C=1,	-	-	-	-	-	REL	25	rr	3
BLS rel	小于或等 于转移	PC ← (PC)+2+rel, C =1	-	-	-	-	-	REL	23	rr	3
BMC rel	l=0 移	PC ← (PC)+2+rel, l=0	-	-	-	-	-	REL	2C	rr	3
BMI rel	N=1 转移	PC ← (PC)+2+rel, N=1	-	-	-	-	-	REL	2B	rr	3
BMS rel	l=1 移	PC ← (PC)+2+rel, l=1	-	-	-	-	-	REL	2D	rr	3
BNE rel	不等分支 转移	PC ← (PC)+2+rel, Z=0	-	-	-	-	-	REL	26	rr	3
BPL rel	大于等于 转移	PC ← (PC)+2+rel, N=0	-	-	-	-	-	REL	2A	rr	3
BRA rel	无条件转 移	PC ← (PC)+2+rel	-	-	-	-	-	REL	20	rr	3
BRN rel	2 字节的	PC ← (PC)+2	-	-	-	-	-		21	rr	3

	NOP											
BSETn,opr	设置 n 位	将 opr 中的 n 置 1	-	-	-	-	-	DIR(b0) DIR(b1) DIR(b2) DIR(b3) DIR(b4) DIR(b5) DIR(b6) DIR(b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5	
BSR rel	转到子程序指令	PC ← (PC); push (PCL) S ← (SP)-1; Push(PCH) SP ← (SP)-1 PC ← (PC)+2+rel	-	-	-	-	-	REL	AD	rr	6	
BRCLR n, opr, Rel	位测试"0"转移	n 位=0,则 PC ← (PC)+3+rel 否则继续	-	-	-	-	*	DIR(b0) DIR(b1) DIR(b2) DIR(b3) DIR(b4) DIR(b5) DIR(b6) DIR(b7)	01 03 05 07 09 0B 0D 0F	ddrr ddrr ddrr ddrr ddrr ddrr ddrr ddrr	5 5 5 5 5 5 5 5	
BRSETn, opr,Rel	位测试"1"转移	n 位=1,则 PC ← (PC)+3+rel 否则继续	-	-	-	-	*	DIR(b0) DIR(b1) DIR(b2) DIR(b3) DIR(b4) DIR(b5) DIR(b6) DIR(b7)	00 02 04 06 08 0A 0C 0E	ddrr ddrr ddrr ddrr ddrr ddrr ddrr ddrr	5 5 5 5 5 5 5 5	
CLC	清进位位	C ← 0	-	-	-	-	0	INH	98		2	
CLI	清中断屏蔽	I ← 0	-	0	-	-	-	INH	9A		2	
CLR opr CLRA CLR X CLR opr,X CLR,X	清寄存器	\$00 → opr A ← \$00 X ← \$00 \$00 → (x)+opr \$00 → (x)	-		-	1	-	DIR INH INH IX1 IX	3F 4F 5F 6F 7F	dd   ff	5 2 2 6 5	
CMP opr CMP opr CMP opr CMP opr,X CMP opr,X CMP,X	累加器内容同存贮器内容比较	(A) - (M)	-	-	*	*	*	IMM DIR EXT IX2 IX1 IX	A1 B1 C1 D1 E1 F1	ii dd hh ll ee ff ff	2 3 4 5 4 3	
COM opr COMA COMX COM opr,X	寄存器内容取反	$\overline{(\text{opr})} \rightarrow (\text{opr})$ A ← $\overline{A}$ X ← $\overline{X}$ $\overline{((x)+opr)} \rightarrow (x)+opr$ $\overline{((x))} \rightarrow (x)$	-	-	*	*	1	DIR INH INH IX1	33 43 53 63	dd  ff	5 2 2 6 5	

COM,X								IX	73		
Cpx opr CPX opr CPX opr CPX opr,X CPX opr,X CPX,X	变址寄存 器内 容同存贮 器内 容比较	(x) - (opr) (x) - (opr) (x) - (opr) (x) - ((x)+opr) (x) - ((x)+opr) (x) - (x)	-		_*	*	1	IMM DIR EXT IX2 IX1 IX	A3 B3 C3 D3 E3 F3	ii dd hh ll ee ff ff	2 3 4 5 4 3
DEC opr DECA DECX DEC opr,X DEC,X	寄存器内 容减 1	(opr) - 1 → opr (A) - 1 → A (X) - 1 → X ((X)+opr) - 1 → (x)+opr (x) - 1 → (x)	-		_*	*	-	DIR INH INH IX1 IX	3A 4A 5A 6A 7A	dd   ff	5 2 2 6 5
EOR opr EOR opr EOR opr EOR opr,X EOR opr,X EOR,X	累加器内 容同 存贮器内 容异 式	(A) ⊕ opr → A (A) ⊕ opr → A (A) ⊕ opr → A (A) ⊕ ((x)+opr) → A (A) ⊕ ((x)+opr) → A (A) ⊕ ((x)) → A	-		_*	*	-	IMM DIR EXT IX2 IX1 IX	A8 B8 C8 D8 E8 F8	ii dd hh ll ee ff ff	2 3 4 5 4 3
INC opr INCA INCX INC opr,X INC,X	存贮器或 寄存 器内 容加 1	(opr)+1 → opr (A)+1 → A (X)+1 → X ((x)+opr) → (x)+opr ((x)) +1 → (x)	-		_*	*	-	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd   ff	5 2 2 6 5
JMP opr JMP opr JMP opr,X JMP opr,X JMP ,X	无条件跳 转	PC 跳跃地址	-				--	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff	2 3 4 3 2
JSR opr JSR opr JSR opr,X JSR opr,X JSR ,X	跳转子程 序	PC ← (PC)+n(n=1,2,3) Push (PCL); SP ← (SP)-1 Push(PCH); SP ← (SP)-1 PC 转条件地址, 同上	-				--	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	5 6 6 5 5
LDA opr LDA opr8 LDA opr16 LDA opr,X LDA opr,X LDA,X	存贮器内 容装 载到累加 器	A ← opr A ← opr8 A ← opr16 A ← ((x)+opr8) A ← ((x)+opr16) A ← ((x))	-		_*	*	-	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff	2 3 4 5 4 3
LDX opr LDX opr8 LDX opr16 LDX opr,X LDX opr,X LDX,X	存贮器内 容装 载到变址 寄存 器	opr → x (opr8) → x (opr16) → x ((x)+opr8) → x ((x)+opr16) → x ((x)) → x	-		_*	*	-	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff	2 3 4 5 4 3

LSL opr LSLA LSLX LSL opr,X LSL ,X	逻辑左移		-		_*	*	*	DIR INH INH IX1 IX	38 48 58 68 78	dd  ff	5 3 3 6 5
LSR opr LSRA LSRX LSR opr,X LSR ,X	逻辑右移		-		_*	*	*	DIR INH INH IX1 IX	34 44 54 64 74	dd  ff	5 2 2 6 5
MUL	无符号乘法	$X:A \leftarrow (X)X(A)$	0	-	-	-	0	INH	42		10
NEG opr NEGA NEGX NEG opr,X NEG,X	存贮器或寄存 器内容取 补	$\$00-(M) \rightarrow opr$ $\$00-(A) \rightarrow A$ $\$00-(X) \rightarrow x$ $\$00-(M) \rightarrow M$ $\$00-(M) \rightarrow M$	-		_*	*	*	DIR INH INH IX1 IX	30 40 50 60 70	ii  ff	5 2 2 6 5
NOP	空操作		-	-	-	-	-	INH	9D		2
ORA opr ORA opr ORA opr ORA opr,X ORA opr,X ORA,X	累加器内 容同 存贮器内 容相 “或”送 入累 加器	$(A) \vee opr \rightarrow A$ $(A) \vee opr \rightarrow A$ $(A) \vee ((x) + opr) \rightarrow A$ $(A) \vee ((x) + opr) \rightarrow A$ $(A) \vee ((x)) \rightarrow A$	-		_*	*	-	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL opr ROLA ROLX ROL opr,X ROL ,X	带进位位 的循 环左移		-		_*	*	*	DIR INH INH IX1 IX	39 49 59 69 79	dd  ff	5 2 2 6 5
ROR opr RORA RORX ROR opr,X ROR ,X	带进位位 的循 环右移		-		_*	*	*	DIR INH INH IX1 IX	36 46 56 66 76	dd  ff	5 2 2 6 5
RSP	复位堆栈 指针	$SP \leftarrow \$00FF$	-	-	-	-	-	INH	9C		2
RTI	中断返回	$SP \leftarrow (SP)+1; Pull(CCR)$ $SP \leftarrow (SP)+1; Pull(A)$ $SP \leftarrow (SP)+1; Pull(X)$ $SP \leftarrow (SP)+1; Pull(PCH)$ $SP \leftarrow (SP)+1; Pull(PCL)$	*	*	*	*	*	INH	80		8
RTS	子程序返 回	$SP \leftarrow (SP)+1; Pull(PCH)$ $SP \leftarrow (SP)+1; Pull(PCL)$						INH	81		5
SBC #opr SBC opr SBC opr SBC opr,X SBC opr,X SBC,X	累加器内 容减 去存贮器 的内 容再减去	$A \leftarrow (A)-(M)-(C)$	-		_*	*	*	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff	2 3 4 5 4 3

	进位位											
SEC	设置进位位	$C \leftarrow 1$	-	-	-	-	1	INH	99			2
SEI	关中断	$I \leftarrow 1$	-	1	-	-	-	INH	9B			2
STA opr STA opr STA opr,X STA opr,X STA,X	将累加器内容保存到存储器	$M \leftarrow (A)$	-	-	*	*	-	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff		3 4 5 4 3
Stop	使能 IRQ 中断, 停止振荡器		-	*	-	-	-	INH	8E			2
STX opr STX opr STX opr,X STX opr,X STX,X	保存变址寄存器内容到存储器	$M \leftarrow (X)$	-	-	*	*	-	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff		3 4 5 4 3
SUC #opr SUB opr SUB opr SUB opr,X SUB opr,X SUB,X	累加器内容减存储器内容送入累加器	$A \leftarrow (A) - (M)$	-	-	*	*	*	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff		2 3 4 5 4 3
SWI	软件中断	$PC \leftarrow (PC) + 1$ ; Push(PCL) $SP \leftarrow (SP) - 1$ ; Push(PCH) $SP \leftarrow (SP) - 1$ ; Push(X) $SP \leftarrow (SP) - 1$ ; Push(CCR) $SP \leftarrow (SP) - 1$ ; I PCH ← 高位矢量中断 PCL ← 低位矢量中断	-	-	*	*	-	INH	83			10
TAX	传送累加器内容到变址寄存器	$X \leftarrow (A)$ (A)不变	-	-	-	-	-	INH	97			2
TST opr TSTA TSTX TST opr,X TST,X	测存储器内容是负或0	$(M) \text{ \$}00$	-	-	*	*	-	DIR INH INH IX1 IX	3D 4D 5D 6D 7D	dd ff		5 2 2 6 5
TXA	传送变址寄存器内容到累加器	$A \leftarrow (X)$ , (X) 不变	-	-	-	-	-	INH	9F			2
WAIT	CPU 使能中断	$I=0$	-	*	-	-	-	INH	8F			2

暂停 CPU									
--------	--	--	--	--	--	--	--	--	--

<b>A.</b> 累加器 <b>C.</b> 进位/借位标志 <b>CCR</b> 状态寄存器 <b>dd</b> 直接操作地址 <b>dd rr</b> 直接操作地址和相对偏移转移命令 <b>DIR</b> 直接地址方式 <b>ee ff</b> ' 址偏移量高、低位字节、 <b>16</b> 位偏移量变址 <b>EXT</b> 外部寻址方式 <b>ff</b> ' 址偏移量, 8 位偏移量变址 <b>H</b> %进位标志 <b>hh ll</b> 外部寻址的高、低位操作地 <b>L</b> 地址屏蔽 <b>ii</b> 立即操作数 <b>IM M</b> 立即寻址方式 <b>INH</b> 隐含寻址方式 <b>IX</b> 无偏移量变址寻址方 <b>IX 1</b> 8 位偏移量变址寻址方 <b>IX 2</b> 16 位偏移量变址寻址方 <b>M</b> 寄存器地址 <b>N</b> 负标志 <b>n</b> 任何位	<b>opr</b> 操作数(一或两字节) <b>PC</b> 程序计数器 <b>PCH</b> 程序计数器高位 <b>PCL</b> 程序计数器高位 <b>REL</b> 相对寻址方式 <b>rel</b> 相对程序计数器相对偏移字节 <b>rr</b> 相对程序计数器相对偏移字节 <b>sp</b> 堆栈指针 <b>z</b> 零标志 <b>#</b> 立即数值 <b>逻辑“与”</b> <b>∨</b> 逻辑“或” <b>⊕</b> 逻辑异或 <b>( )</b> 内容 <b>-( )</b> 负 <b>←</b> 装入到 <b>?</b> 如果 <b>:</b> 连接 <b>↔</b> 位置或清零 <b>-</b> 无影响 <b>*</b> 取决于指令执行结果 <b>x</b> 变址寄存器
--	--

## 第十一章 电气特性

### 11.1 极限参数值

注意:为避免损坏 MCU,不要超越表 10-1 列出的工作范围,V<sub>in</sub> 和 V<sub>out</sub> 必须维持在 V<sub>ss</sub>≤(V<sub>IN</sub> 或 V<sub>out</sub>)≤V<sub>DD</sub> 范围内。

**表 11-1 最大额定值:**

参数	符号	值	单位
电源	V <sub>dd</sub>	-0.3--7v	V
输入电压	V <sub>in</sub>	V <sub>ss</sub> -0.3 到 V <sub>dd</sub> +0.3	V
V <sub>pp</sub>	V <sub>in</sub>		V
输出驱动电流 (每 PIN)	I <sub>d</sub>	25	mA
工作温度 民用 工业用	T <sub>A</sub>	T <sub>l</sub> --T <sub>h</sub> 0° - 70°C -40° - +85°C	°C
储存温度范围	T <sub>stg</sub>	-65° - +150°C	°C

### 11.2 交流电特性及定义:

V<sub>dd</sub>=5.0v dc±10% V<sub>ss</sub>=0v T=0-70°C

参数	符号	最小	最大	单位
----	----	----	----	----

工作频率： RC 震荡选择 晶体震荡选择 外部时钟选择	Fosc	0.1 0.1 直流	4.0 4.0 4.0	Mhz Mhz Mhz
内部工作频率： RC 震荡 晶体震荡 外部时钟	Fop	- - -	2.0 2.0 2.0	Mhz Mhz Mhz
内部周期时间 (1/Fop)	Tcyc	500	-	ns
RC 震荡器稳定时间	Trcon	-	1	ms
晶体震荡器启震时间	Toxov	-	100	ms
Stop 唤醒时间 (晶体震荡器)	同 Toxov	-	100	ms
ResetB 脉冲最小宽度	Trl	1.5	-	Tcyc
Timer 分辨率	Tresl	4	-	Tcyc
中断脉冲"0"宽度(边缘触发)	Tilih	125	-	ns
中断脉冲周期	Tilil	19	-	Tcyc
PA0-7 中断脉冲"1"宽度(边缘触发)	Tihil	125	-	ns
PA0-7 中断脉冲周期	Tihih	19	-	Tcyc
OSC1 脉冲宽度	T	90	-	ns
RC 震荡器频率稳定性 Fosc=2Mhz Vdd=5v±10% Ta=-40°C 到+85°C Ta=0°C 到+40°C	ΔFosc	- -	±25 ±15	% %

### 11.3 直流电特性及定义：

Vdd=5.0Vdc±10% Vss=0Vdc T=0-70°C

特性	符号	最小	典型	最大	单位
输出电压 I load = -10μa I load = +10μa	Voh Vol	Vdd-0.1 -	- -	- 0.1	V V
输出高电平 (I load = -0.8ma) 全部 I/O 腿	Voh	Vdd-0.8	-	-	V
输出低电平 (I load = 1.6ma) 全部 I/O 腿	Vol	-	0.1	0.4	V
输出高电平电流 Voh=2.5v 全部 I/O 腿 Voh=3v (PB5-PB7 低 电流选择)	Ioh Ioh	10 2	- -	- -	ma ma
输出低电平电流 Vol=2.5v 全部 I/O 腿 Vol=3v (PB5-PB7 低 电流选择)	Iol Iol	10 2			ma ma
总的 I/O 驱动电流	Iport	-	100	-	ma

输入高电平 PA0-7,PB0-1,IrqB, ResetB,OSC1	Vih	0.7×Vdd	-	Vdd	V
输入低电平 PA0-7,PB0-1,IrqB, ResetB,OSC1	Vil	Vss	-	0.2×Vdd	V
功耗 Run(运行) Wait(等待) Stop(停机)	Idd	-	5.0 1.3	9.0 2.5 30	ma ma µa
I/O 高阻时漏电	Iil	-	-	±10	µa
输入电流 Timer, ResetB,IrqB,Osc1,	Iin	-	-	±1	µa
上拉电阻 PA0-7, PB0-7, PC0-7, PD0-7, ResetB,IqrB	Rpu	14 14 85	36 36 100	50 50 176	kΩ kΩ kΩ

备注:

(1) Idd (Run) 是指 Fosc=2.0MHz, 全部输入 PIN 为 0.2Vdc, 无直流负载时电流。

(2) Stop Idd: 全部 I/O 口为输入, Vil=0.2Vdc, Vih=Vdd-0.2Vddc, Osc1=Vss。