



# 8-Bit 通用 AD 型单片机

**20E24B**

版本：V00 日期：2013/08/22

[www.greenmcu.com](http://www.greenmcu.com)

# 目 录

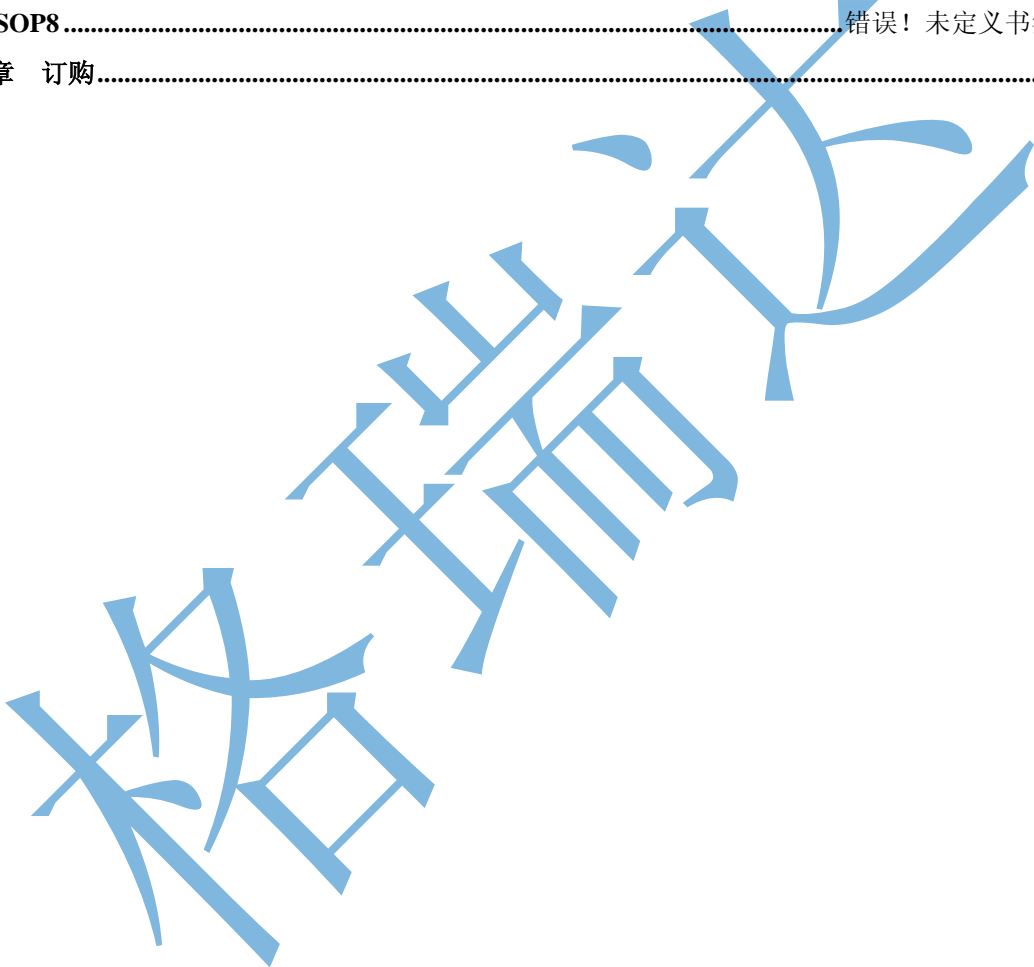
<b>第 1 章 简述</b> .....	<b>1</b>
<b>1.1 主要特点</b> .....	<b>1</b>
<b>1.2 管脚排列</b> .....	<b>1</b>
<b>1.3 电路特性参数</b> .....	<b>2</b>
1.3.1 极限参数.....	2
1.3.2 直流参数.....	3
1.3.3 LVR 电路参数.....	3
1.3.4 ADC 电路参数.....	3
1.3.5 管脚功能说明.....	4
<b>第 2 章 系统</b> .....	<b>5</b>
<b>2.1 系统框图</b> .....	<b>5</b>
<b>2.2 地址分配</b> .....	<b>5</b>
<b>2.3 控制寄存器说明</b> .....	<b>5</b>
2.3.1 TCNT (TIMER0 计数寄存器).....	6
2.3.2 TDATA (TIMER0 比较寄存器).....	6
2.3.3 TCON (TIMER0 控制寄存器).....	6
2.3.4 MCR (杂用控制寄存器).....	7
2.3.5 BTCON (BASIC TIMER 控制寄存器).....	7
2.3.6 BTCNT (BASIC TIMER 计数寄存器).....	7
2.3.7 P0 (P0 口数据寄存器).....	7
2.3.8 P1 (P1 口数据寄存器).....	7
2.3.9 P2 (P2 口数据寄存器).....	7
2.3.10 P0CONH (P0 口高位控制寄存器).....	7
2.3.11 P0CONL (P0 口低位控制寄存器).....	8
2.3.12 P0PND (外中断输入控制寄存器).....	8
2.3.13 P1CON (P1 口控制寄存器).....	9
2.3.14 P2CONH (P2 口高位控制寄存器).....	9
2.3.15 P2CONL (P2 口低位控制寄存器).....	9
2.3.16 PWMDATA (PWM 数据寄存器).....	10
2.3.17 PWMCON (PWM 控制寄存器).....	10
2.3.18 ADCON (AD 控制寄存器).....	10
2.3.19 ADDATAH (AD 数据寄存器 高 8 位).....	11
2.3.20 ADDATAL (AD 数据寄存器 低 2 位).....	11
<b>2.4 中断结构</b> .....	<b>11</b>
<b>2.5 系统时钟</b> .....	<b>11</b>
<b>2.6 系统复位</b> .....	<b>12</b>
<b>2.7 IO 口</b> .....	<b>12</b>
<b>2.8 BASIC TIMER (BT)</b> .....	<b>13</b>
<b>2.9 TIMER0 (T0)</b> .....	<b>14</b>

<b>2.10 PWM</b> .....	<b>15</b>
2.10.1 PWMCNT .....	15
2.10.2 PWMDATA.....	15
2.10.3 PWM 功能描述 .....	15
<b>2.11 ADC</b> .....	<b>16</b>
<b>2.12 OPBIT</b> .....	<b>16</b>
<b>第3章 H05 指令集</b> .....	<b>17</b>
<b>3.1 简介</b> .....	<b>17</b>
<b>3.2 CPU (中央处理器)</b> .....	<b>17</b>
3.2.1 ALU (算数逻辑单元) .....	18
3.2.2 CPU CONTROL (CPU 控制单元) .....	18
3.2.3 CPU 寄存器 .....	18
3.2.3.1 累加器.....	18
3.2.3.2 变址寄存器.....	18
3.2.3.3 状态寄存器.....	18
3.2.3.4 PC 指针.....	19
3.2.3.5 堆栈指针.....	19
<b>3.3 寻址方式</b> .....	<b>20</b>
3.3.1 隐含寻址.....	20
3.3.2 立即寻址.....	21
3.3.3 扩展寻址.....	21
3.3.4 直接寻址.....	22
3.3.5 变址寻址.....	23
3.3.6 无偏移量变址寻址.....	23
3.3.7 8 位偏移量变址寻址.....	24
3.3.8 16 位偏移量变址寻址.....	25
3.3.9 相对寻址.....	26
<b>3.4 按照功能分类的所有指令的汇总。</b> .....	<b>28</b>
<b>3.5 HC05 指令集汇总</b> .....	<b>31</b>
<b>3.6 HC05 指令集详述</b> .....	<b>40</b>
3.6.1 ADC 带进位的加法 .....	40
3.6.2 ADD 加法.....	40
3.6.3 AND 逻辑与.....	41
3.6.4 ASL 算术左移 (同 LSL) .....	42
3.6.5 ASR 算术右移.....	43
3.6.6 BCC 无进位则跳 (同 BHS).....	43
3.6.7 BCLR <sub>N</sub> 第 N 位清零.....	44
3.6.8 BCS 进位则跳转 (同 BLO) .....	44
3.6.9 BEQ 等于则跳转 .....	45
3.6.10 BHCC 无半进位则跳转.....	45
3.6.11 BHCS 半进位则跳转 .....	45
3.6.12 BHI 大于则跳转 .....	46
3.6.13 BHS 大于等于则跳转 (同 BCC) .....	46
3.6.14 BIH IRQ 为高则跳转①.....	46

3.6.15 BIL IRQ 为低则跳转① .....	47
3.6.16 BIT 位测试 .....	47
3.6.17 BLO 小于则跳转 (同 BCS) .....	48
3.6.18 BLS 小于等于则跳转 .....	48
3.6.19 BMC 中断未屏蔽则跳转 .....	49
3.6.20 BMI 值为负则跳转 .....	49
3.6.21 BMS 中断屏蔽则跳转 .....	49
3.6.22 BNE 不等于则跳转 .....	50
3.6.23 BPL 值为正则跳转 .....	50
3.6.24 BRA 无条件跳转 .....	51
3.6.25 BRCLR N 第 N 位为 0 则跳转 .....	51
3.6.26 BRN 永不跳转 .....	52
3.6.27 BRSET N 第 N 位为 1 则跳转 .....	52
3.6.28 BSET N 第 N 位置位 .....	53
3.6.29 BSR 跳转到子程序 .....	53
3.6.30 CLC C 标志清零 .....	54
3.6.31 CLI I 标志清零 .....	54
3.6.32 CLR 清零 .....	54
3.6.33 CMP A 寄存器比较 .....	55
3.6.34 COM 取反 .....	56
3.6.35 CPX X 寄存器比较 .....	56
3.6.36 DEC 减一 .....	57
3.6.37 EOR 逻辑异或 .....	57
3.6.38 INC 加一 .....	58
3.6.39 JMP 跳转 .....	59
3.6.40 JSR 调用子程序 .....	59
3.6.41 LDA A 寄存器存数 .....	60
3.6.42 LDX X 寄存器存数 .....	60
3.6.43 LSL 逻辑左移 (同 ASL) .....	61
3.6.44 LSR 逻辑右移 .....	62
3.6.45 MUL 乘法 (暂不支持) .....	62
3.6.46 NEG 取补 .....	63
3.6.47 NOP 空操作 .....	63
3.6.48 ORA 逻辑或 .....	63
3.6.49 ROL 循环左移 .....	64
3.6.50 RSR 循环右移 .....	65
3.6.51 RSP SP 复位 .....	65
3.6.52 RTI 中断返回 .....	66
3.6.53 RTS 子程序返回 .....	66
3.6.54 SBC 带借位的减法 .....	66
3.6.55 SEC C 标志置位 .....	67
3.6.56 SEI I 标志置位 .....	67
3.6.57 STA A 寄存器取数 .....	68
3.6.58 STOP 进 STOP 模式 .....	68
3.6.59 STX X 寄存器取数 .....	69
3.6.60 SUB 减法 .....	69
3.6.61 SWI 软中断 .....	70
3.6.62 TAX 将 A 的值传到 X .....	70
3.6.63 TST 零测试 .....	71

---

3.6.64 TXA 将 X 的值传到 A .....	71
3.6.65 WAIT 进 WAIT 模式 .....	72
<b>第 4 章 封装尺寸 .....</b>	<b>错误! 未定义书签。</b>
<b>4.1 DIP20 .....</b>	<b>73</b>
<b>4.2 SOP20 .....</b>	<b>错误! 未定义书签。</b>
<b>4.3 DIP16 .....</b>	<b>错误! 未定义书签。</b>
<b>4.4 SOP16 .....</b>	<b>错误! 未定义书签。</b>
<b>4.5 DIP8 .....</b>	<b>错误! 未定义书签。</b>
<b>4.6 SOP8 .....</b>	<b>错误! 未定义书签。</b>
<b>第 5 章 订购 .....</b>	<b>75</b>

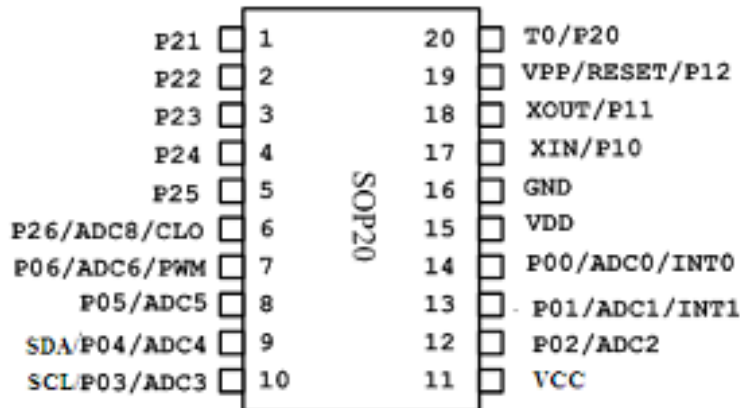
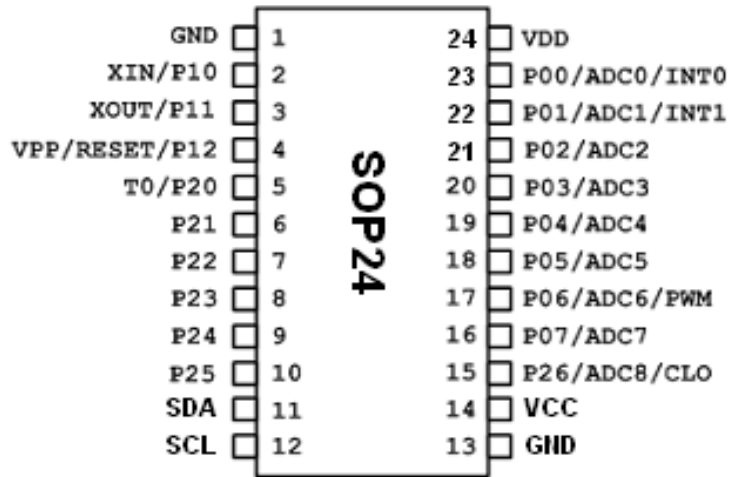


## 第1章 简述

### 1.1 主要特点

- ◇ 8 位 CISC 型内核 (HC05)
- ◇ 4K byte OTP ROM
- ◇ 2K bit 串行 EEPROM
- ◇ 208 byte RAM
- ◇ 3 组 IO 口 (最多可支持 17 个通用 IO 口和 1 个输入口)
- ◇ 1 个 PWM 输出
- ◇ 1 个 8 位基本定时器
- ◇ 1 个 8 位带比较输出的定时器
- ◇ 1 个 10 位 ADC (9 路输入)
- ◇ 2 个外中断、1 个定时器中断、1 个 PWM 中断
- ◇ 看门狗复位功能
- ◇ 2.1V/3.6V 低压复位
- ◇ 可选晶振/RC 振荡
  - 晶振 400K-8MHz
  - RC 振荡有 2MHz、4MHz、8MHz 3 种可选
- ◇ 工作电压
  - 2.2-5.5V (工作频率 400K-4MHz)
  - 2.7-5.5V (工作频率 4M-8MHz)
- ◇ 工作温度 -40-85°C
- ◇ 封装形式: SOP20/SOP24

## 1.2 管脚排列



## 1.3 电路特性参数

### 1.3.1 极限参数

( $T_A=25^{\circ}\text{C}$ )

Parameter	Symbol	Condition	Rating	Unit
Supply voltage	$V_{DD}$	-	-0.3 to +6.5	V
Input voltage	$V_I$	All ports	-0.3 to $V_{DD}+0.5$	V
Output voltage	$V_O$	All output ports	-0.3 to $V_{DD}+0.5$	V
Output current high	$I_{OH}$	One I/O pin active	-25	mA
		All I/O pin active	-80	mA
Output current low	$I_{OL}$	One I/O pin active	+30	mA
		All I/O pin active	+150	mA

Operating temperature	T <sub>A</sub>	-	-40 to +85	°C
Storage temperature	T <sub>STG</sub>	-	-65 to +150	°C

### 1.3.2 直流参数

(T<sub>A</sub>=25°C VDD=2.7-5.5V)

Parameter	Sym.	Condition	Min	Typ	Max	Unit
Input high voltage	V <sub>IH</sub>	P0,P1,P2 VDD=2.7-5.5V	0.8VDD	-	VDD	V
Input low voltage	V <sub>IL</sub>	P0,P1,P2 VDD=2.7-5.5V	0	-	0.2VDD	V
Output high voltage	V <sub>OH</sub>	I <sub>OH</sub> =-10mA P0,P1,P2 VDD=4.5-5.5V	VDD-1.5	VDD-0.4	-	V
Output low voltage	V <sub>OL</sub>	I <sub>OL</sub> =25mA P0,P1,P2 VDD=4.5-5.5V	-	0.5	2.0	V
Input high leakage current	I <sub>IH</sub>	All input V <sub>IN</sub> =VDD	-	-	1	uA
Input low leakage current	I <sub>IL</sub>	All input V <sub>IN</sub> =0	-	-	1	uA
Pull-up resistors	R <sub>PU1</sub>	P0,P1,P2 VDD=2.7-5.5V	-	20	-	KΩ
Pull-up resistors	R <sub>PU2</sub>	P12 VDD=2.7-5.5V	-	100	-	KΩ
Pull-down resistors	R <sub>PD</sub>	P0,P1,P2 VDD=2.7-5.5V	-	20	-	KΩ
Dynamic working current	I <sub>DD</sub>	4MHz clock VDD=4.5-5.5V	-	5	10	mA
Standby working current	I <sub>STD1</sub>	STOP mode LVR disable VDD=4.5-5.5V	-	0.1	1	uA
	I <sub>STD2</sub>	STOP mode LVR enable VDD=4.5-5.5V	-	10	20	

### 1.3.3 LVR 电路参数

(T<sub>A</sub>=25°C VDD=2.7-5.5V)

Parameter	Sym.	Condition	Min	Typ	Max	Unit
Low voltage reset	V <sub>LVR</sub>	-	-	2.1	-	V
Low voltage reset	V <sub>LVR</sub>	-	-	3.6	-	V

### 1.3.4 ADC 电路参数

(T<sub>A</sub>=25°C VDD=5.0V)

Parameter	Sym.	Condition	Min	Typ	Max	Unit
Total accuracy	-	VDD=5V fosc=4MHz	-	-	±3	LSB
Integral linearity error	ILE	VDD=5V fosc=4MHz	-	-	±2	LSB
Differential	DLE	VDD=5V	-	-	±1	LSB

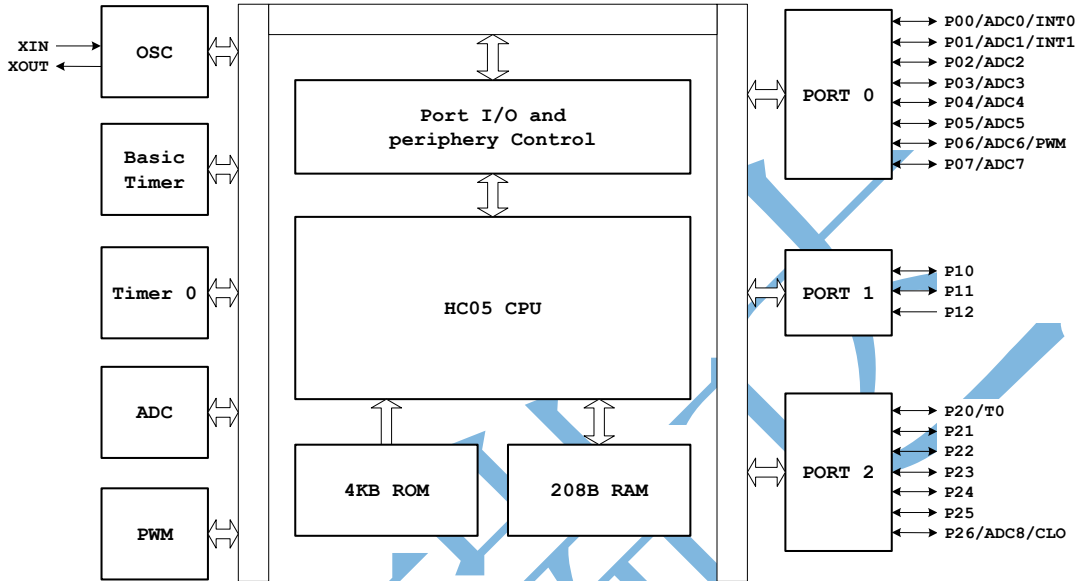
linearity error		fosc=4MHz				
Offset error of top	EOT	VDD=5V fosc=4MHz	-	±1	±3	LSB
Offset error of bottom	EOB	VDD=5V fosc=4MHz				
Conversion time	t <sub>CON</sub>	VDD=5V fosc=4MHz	-	25	-	us
Analog input voltage	V <sub>IAN</sub>	-	VSS	-	VDD	V
Analog input impedance	R <sub>AN</sub>	-	2	-	-	MΩ
Analog input current	I <sub>ADIN</sub>	VDD=5V	-	-	10	uA
Analog block current	I <sub>ADC</sub>	VDD=5V	-	1	3	mA
		VDD=5V power down mode	-	0.1	0.5	uA

### 1.3.5 管脚功能说明

Name	I/O	Alternative usage
GND	-	地。
P10	I/O	PORT1.0, 可复用为 XIN。
P11	I/O	PORT1.1, 可复用为 XOUT。
P12	INPUT	PORT1.2, 可复用为 RESET/VPP。
P20	I/O	PORT2.0, 可复用为 T0。
P21	I/O	PORT2.1。
P22	I/O	PORT2.2。
P23	I/O	PORT2.3。
P24	I/O	PORT2.4。
P25	I/O	PORT2.5。
SDA	I/O	EEPROM 数据脚
SCL	I	EEPROM 时钟脚
GND	-	EEPROM 地线
VCC	-	EEPROM 电源
P26	I/O	PORT2.6, 可复用为 ADC8/CLO。
P07	I/O	PORT0.7, 可复用为 ADC7。
P06	I/O	PORT0.6, 可复用为 ADC6/PWM。
P05	I/O	PORT0.5, 可复用为 ADC5。
P04	I/O	PORT0.4, 可复用为 ADC4/SDA。
P03	I/O	PORT0.3, 可复用为 ADC3/SCL。
P02	I/O	PORT0.2, 可复用为 ADC2。
P01	I/O	PORT0.1, 可复用为 ADC1/INT1。
P00	I/O	PORT0.0, 可复用为 ADC0/INT0。
VDD	-	电源。

## 第 2 章 系统

### 2.1 系统框图



系统框图

### 2.2 地址分配

\$0000-\$002F: Control registers  
 \$0030-\$00FF: RAM (208 bytes)  
 \$0100-\$0FFF: Reserved  
 \$1000-\$1FFF: OTP ROM (4096 bytes)

### 2.3 控制寄存器说明

MC20E24B 的全部控制寄存器见下表。

寄存器名	地址	R/W	缺省值
TOCNT	\$00	R	0000 0000
T0DATA	\$01	R/W	1111 1111
T0CON	\$02	R/W	00-- 0-00
MCR	\$03	R/W	---- --I
BTCN	\$0C	R/W	0000 1000
BTCNT	\$0D	R	0000 0000
P0	\$10	R/W	0000 0000
P1	\$11	R/W	---- -000
P2	\$12	R/W	-000 0000

P0CONH	\$16	R/W	0000 0000
P0CONL	\$17	R/W	0000 0000
P0PND	\$18	R/W	---- 0000
P1CON	\$19	R/W	00-- 0000
P2CONH	\$1A	R/W	-000 0000
P2CONL	\$1B	R/W	0000 0000
PWMDATA	\$22	R/W	0000 0000
PWMCON	\$23	R/W	00-0 0000
ADCON	\$27	R/W	0000 1000
ADDATAH	\$28	R	XXXX XXXX
ADDATAL	\$29	R	---- --XX

说明：

- 缺省值是指 MCU 复位时寄存器的初始值
- X 表示初始值不确定
- I 表示寄存器控制位的值由 OPBIT[2]决定

### 2.3.1 T0CNT (TIMER0 计数寄存器)

T0CNT 可用于查询 TIMER0 计数器 8 位计数值，该寄存器是只读的。

### 2.3.2 T0DATA (TIMER0 比较寄存器)

T0DATA 用于设置 TIMER0 的 MATCH 值，在计数过程中当 T0CNT=T0DATA 条件满足时，T0F 会置 1。

### 2.3.3 T0CON (TIMER0 控制寄存器)

T0CON 控制 TIMER0 的功能。

.7-.6 TOPS[1:0] TIMER0 时钟分频系数选择

00: Fsys/4096

01: Fsys/256

10: Fsys/8

11: Fsys

(注：Fsys 指的是 MCU 系统时钟频率，为 OSC 或 RC 振荡频率的 1/2)

.5-.4 保留

.3 T0CLR 计数器清 0 位，读出时总是为 0

写 0: 无效

写 1: 对 TIMER0 计数器清 0

.2 保留

.1 T0E TIMER0 中断使能位

0: T0 中断关闭

1: T0 中断打开

.0 T0F TIMER0 中断标志位

0: 无 T0 中断请求 (读操作)

0: 清 T0 中断标志 (写操作)

1: 有 T0 中断请求 (读操作)

1: 无效 (写操作)

(注：中断标志位 T0F 的状态与中断使能位 T0E 的设置无关，其他中断也类似)

### 2.3.4 MCR (杂用控制寄存器)

MCR 是仅有一个控制位 LVRE 的寄存器。LVRE 用以控制 LVR 的开关, 它的初始值由 OPBIT[2] 决定。

- .7-1 保留
- .0 LVRE LVR 功能使能位
  - 0: LVR 关闭
  - 1: LVR 开启

### 2.3.5 BTCON (Basic Timer 控制寄存器)

BTCON 用于选择输入时钟频率, 清 BT 计数器和分频器, 开关 WDT。

- .7-4 WDTE[3:0] WDT 功能使能位
  - 1010: 关闭 WDT
  - 其他: 打开 WDT

(注: WDT 复位后默认是打开的, 若要关闭 WDT, 必须设为 1010)
- .3-2 BTPS[1:0] BT 时钟分频系数选择
  - 00: F<sub>sys</sub>/4096
  - 01: F<sub>sys</sub>/1024
  - 10: F<sub>sys</sub>/256
  - 11: F<sub>sys</sub>/128
- .1 BTCLR BT 计数器清 0 位, 读出时总是为 0
  - 写 0: 无效
  - 写 1: 对 BT 计数器清 0
- .0 DVCLR BT 时钟分频器清 0 位, 读出时总是为 0
  - 写 0: 无效
  - 写 1: 对 BT 时钟分频器清 0

### 2.3.6 BTCNT (Basic Timer 计数寄存器)

BTCNT 可用于查询 BT 计数器 8 位计数值, 该寄存器是只读的。

### 2.3.7 P0 (P0 口数据寄存器)

当 P00-P07 作 IO 口时, 用于对其进行访问 (输入) 或设置 (输出)。

### 2.3.8 P1 (P1 口数据寄存器)

当 P10-P12 作 IO 口时, 用于对其进行访问 (输入) 或设置 (输出)。

### 2.3.9 P2 (P2 口数据寄存器)

当 P20-P26 作 IO 口时, 用于对其进行访问 (输入) 或设置 (输出)。

### 2.3.10 P0CONH (P0 口高位控制寄存器)

- .7-6 P07C[1:0] P07 控制位
  - 00: 带上拉电阻的输入
  - 01: 不带上拉电阻的输入
  - 10: 输出
  - 11: 作为 ADC7 输入
- .5-4 P06C[1:0] P06 控制位
  - 00: 带上拉电阻的输入
  - 01: PWM 输出
  - 10: 输出

- 11: 作为 ADC6 输入
- .3-.2 P05C[1:0] P05 控制位
  - 00: 带上拉电阻的输入
  - 01: 不带上拉电阻的输入
  - 10: 输出
  - 11: 作为 ADC5 输入
- .1-.0 P04C[1:0] P04 控制位
  - 00: 带上拉电阻的输入
  - 01: 不带上拉电阻的输入
  - 10: 输出
  - 11: 作为 ADC4 输入

### 2.3.11 P0CONL (P0 口低位控制寄存器)

- .7-.6 P03C[1:0] P03 控制位
  - 00: 不带上拉电阻的输入
  - 01: 带上拉电阻的输入
  - 10: 输出
  - 11: 作为 ADC3 输入
- .5-.4 P02C[1:0] P02 控制位
  - 00: 不带上拉电阻的输入
  - 01: 带上拉电阻的输入
  - 10: 输出
  - 11: 作为 ADC2 输入
- .3-.2 P01C[1:0] P01 控制位
  - 00: 不带上拉电阻的输入/INT1 输入
  - 01: 带上拉电阻的输入/INT1 输入
  - 10: 输出
  - 11: 作为 ADC1 输入
- .1-.0 P00C[1:0] P00 控制位
  - 00: 不带上拉电阻的输入/INT0 输入
  - 01: 带上拉电阻的输入/INT0 输入
  - 10: 输出
  - 11: 作为 ADC0 输入

### 2.3.12 P0PND (外中断输入控制寄存器)

用于控制 P00、P01 作外中断的功能。

- .7-.4 保留
- .3 INT1E 外中断 INT1 使能位
  - 0: INT1 关闭
  - 1: INT1 开启
- .2 INT1F 外中断 INT1 标志位
  - 0: 无 INT1 中断请求 (读操作)
  - 0: 清 INT1 中断标志 (写操作)
  - 1: 有 INT1 中断请求 (读操作)
  - 1: 无效 (写操作)
- .1 INT0E 外中断 INT0 使能位
  - 0: INT0 关闭
  - 1: INT0 开启
- .0 INT0F 外中断 INT0 标志位

- 0: 无 INTO 中断请求 (读操作)
- 0: 清 INTO 中断标志 (写操作)
- 1: 有 INTO 中断请求 (读操作)
- 1: 无效 (写操作)

### 2.3.13 P1CON (P1 口控制寄存器)

- .7 P11OD P11 口 open-drain 控制位
  - 0: 普通输出
  - 1: open-drain 输出
- .6 P10OD P10 口 open-drain 控制位
  - 0: 普通输出
  - 1: open-drain 输出
- .5-.4 保留
- .3-.2 P11C[1:0] P11 控制位
  - 00: 不带上下拉电阻的输入
  - 01: 带上拉电阻的输入
  - 10: 输出
  - 11: 带下拉电阻的输入
- .1-.0 P10C[1:0] P10 控制位
  - 00: 不带上下拉电阻的输入
  - 01: 带上拉电阻的输入
  - 10: 输出
  - 11: 带下拉电阻的输入

### 2.3.14 P2CONH (P2 口高位控制寄存器)

- .7 保留
- .6-.4 P26C[2:0] P26 控制位
  - 000: 带上拉电阻的输入
  - 001: 不带上拉电阻的输入
  - 01x: 作为 ADC8 输入
  - 100: 普通输出
  - 101: open-drain 输出, 带上拉电阻
  - 110: open-drain 输出, 不带上拉电阻
  - 111: 作为 CLO 输出 (CLO 频率为系统振荡频率的 1/8)
- .3-.2 P25C[1:0] P25 控制位
  - 00: 带上拉电阻的输入
  - 01: 不带上拉电阻的输入
  - 10: 普通输出
  - 11: open-drain 输出
- .1-.0 P24C[1:0] P24 控制位
  - 00: 带上拉电阻的输入
  - 01: 不带上拉电阻的输入
  - 10: 普通输出
  - 11: open-drain 输出

### 2.3.15 P2CONL (P2 口低位控制寄存器)

- .7-.6 P23C[2:0] P23 控制位
  - 00: 带上拉电阻的输入
  - 01: 不带上拉电阻的输入

- 10: 普通输出
- 11: open-drain 输出
- .5-4 P22C[2:0] P22 控制位
  - 00: 带上拉电阻的输入
  - 01: 不带上拉电阻的输入
  - 10: 普通输出
  - 11: open-drain 输出
- .3-2 P21C[1:0] P21 控制位
  - 00: 带上拉电阻的输入
  - 01: 不带上拉电阻的输入
  - 10: 普通输出
  - 11: open-drain 输出
- .1-0 P20C[1:0] P20 控制位
  - 00: 带上拉电阻的输入
  - 01: 不带上拉电阻的输入
  - 10: 普通输出
  - 11: 作为 T0 输出

### 2.3.16 PWMDATA (PWM 数据寄存器)

PWMDATA 用于设置 PWM 输出波形的宽度，具体设置方法参见 5.10。

### 2.3.17 PWMCON (PWM 控制寄存器)

- .7-6 PWMPS[1:0] PWM 时钟分频系数选择
  - 00: Fsys/64
  - 01: Fsys/8
  - 10: Fsys/2
  - 11: Fsys
- .5 保留
- .4 PWMDRS PWMDATA 重载模式选择
  - 0: 8bit 计数溢出时重载 PWMDATA
  - 1: 6bit 计数溢出时重载 PWMDATA
- .3 PWMCLR PWM 计数器清 0 位，读出时总是为 0
  - 写 0: 无效
  - 写 1: 对 PWM 计数器清 0
- .2 PWMCE PWM 计数器开关位
  - 0: PWM 计数器停止
  - 1: PWM 计数器开启
- .1 PWMIE PWM 中断使能位
  - 0: PWM 中断关闭
  - 1: PWM 中断开启
- .0 PWMIF PWM 中断标志位
  - 0: 无 PWM 中断请求 (读操作)
  - 0: 清 PWM 中断标志 (写操作)
  - 1: 有 PWM 中断请求 (读操作)
  - 1: 无效 (写操作)

### 2.3.18 ADCON (AD 控制寄存器)

- .7-4 ADCH[3:0] AD 通道选择
  - 0000: ADC0

- 0001: ADC1
- 0010: ADC2
- 0011: ADC3
- 0100: ADC4
- 0101: ADC5
- 0110: ADC6
- 0111: ADC7
- 1000: ADC8
- 1001: GND (供测试用)
- 1010: GND (供测试用)
- 1011: GND (供测试用)
- 1100: GND (供测试用)
- 1101: VDD (供测试用)
- 1110: VDD/4 (供测试用)
- 1111: VDD/2 (供测试用)
- .3 EOC AD 转换结束标志, 只读位
  - 0: A/D 转换进行中
  - 1: A/D 转换结束
- .2-.1 ADPS[1:0] AD 时钟分频系数选择
  - 00: Fsys/8
  - 01: Fsys/4
  - 10: Fsys/2
  - 11: Fsys
- .0 ADCE AD 转换启动位, 读出时总为 0
  - 写 0: 无效
  - 写 1: 启动 A/D 转换

### 2.3.19 ADDATAH (AD 数据寄存器 高 8 位)

### 2.3.20 ADDATAL (AD 数据寄存器 低 2 位)

ADDATAH、ADDATAL 用于存储 AD 转换的结果, 具体用法参见 2.11。

## 2.4 中断结构

20E24B 有 4 个中断源, 即外中断 0 (INT0)、外中断 1 (INT1)、TIMER0 中断 (T0INT)、PWM 中断 (PWMINT), 这 4 个中断分别有各自的使能位和标志位。要响应这些中断还必须使用 CLI 指令将 I 标志置 0; 反之, 若使用 SEI 指令将 I 标志置成 1, 则所有的中断源都不会触发中断。

另外指令系统中还有软中断 SWI, 系统复位也可视为一种特殊的中断, 它们不受 I 标志的影响。

以上 6 种中断均有独立的中断向量, 每个中断向量由 2 字节组成, 在中断向量中存放中断服务程序的入口地址。

1FF4:1FF5	INT1
1FF6:1FF7	T0INT
1FF8:1FF9	PWMINT
1FFA:1FFB	INT0
1FFC:1FFD	SWI
1FFE:1FFF	RESET

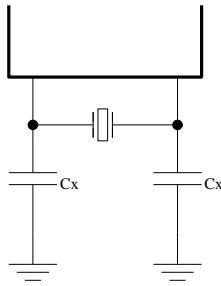
中断优先级从 RESET 到 INT1 依次降低。

## 2.5 系统时钟

20E24B 有 4 种时钟模式, 用户可通过 OPBIT 配置。

OSC 模式：当选用外部晶体振荡工作模式，连接方式见下图。晶体可选用 400KHz~8MHz，通常  $C_x$  是必须的（3.5M 以上的晶振可以不接电容  $C_x$ ）。在实际使用中，用户应使晶体离 XIN、XOUT 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。

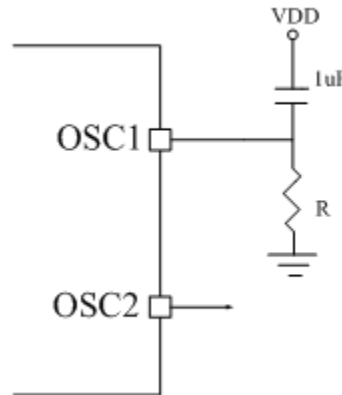


晶体频率	电容 $C_x$
8MHz	不接/15p
4MHz	不接/15p/30p
3.64MHz	不接/15p/30p
455KHz	100u-300u

### 晶体振荡

**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。

◇ 外部 RC 振荡模式 (RC1)：连接方式（两种）见下图。振荡频率曲线见附件。



◇ 内部 RC 振荡 (RC2、RC3)：其中 RC2 振荡频率约 5.2MHz，RC3 振荡频率约 2.6MHz。

这两种内部 RC 振荡均不需要外接任何元件，且 P10、P11 可作为普通 IO 口使用。

另外，系统支持 2 种省电工作模式。

当执行 WAIT 指令后，CPU 的时钟停止工作，任何一个中断源 (INT0、INT1、TOINT、PWMINT) 均可唤醒。

当执行 STOP 指令后，系统所有的时钟都停止工作，晶振（或 RC 振荡）也停振，此时整个芯片的功耗极低（见电性能参数  $I_{STD1}$  和  $I_{STD2}$ ）。外中断 (INT0、INT1) 可唤醒。

## 2.6 系统复位

20E24B 有 4 种方式可令系统复位。

- ◇ 上电复位
- ◇ 外部引脚 P12 低电平复位（需要通过 OPBIT 配置）
- ◇ Watchdog 复位
- ◇ LVR 低电压复位

## 2.7 IO 口

20E24B 最多可支持 18 个 IO 口，根据功能上的差异，有如下几种类型。

- A、带上拉、AD 输入的 IO 口 (P00-P07)
  - ◇ 输出有非交叠控制电路
  - ◇ 输入有施密特和滤波电路
  - ◇ 上拉电阻阻值 30Kohm (@5V TT)
- B、带上拉的 IO 口 (P20)
  - ◇ 输出有非交叠控制电路
  - ◇ 输入有施密特和滤波电路
  - ◇ 上拉电阻阻值 30Kohm (@5V TT)
- C、带上拉、开漏输出的 IO 口 (P21-P25)
  - ◇ 输出有非交叠控制电路
  - ◇ 输入有施密特和滤波电路
  - ◇ 上拉电阻阻值 30Kohm (@5V TT)
  - ◇ open drain 输出时, 输出驱动的 PMOS 管始终关闭
- D、带上拉、开漏输出、AD 输入的 IO 口 (P26)
  - ◇ 输出有非交叠控制电路
  - ◇ 输入有施密特和滤波电路
  - ◇ 上拉电阻阻值 30Kohm (@5V TT)
  - ◇ open drain 输出时, 输出驱动的 PMOS 管始终关闭
- E、VPP/RESET/输入复用引脚 (P12)
  - ◇ P12 输入的电压大于 8V 时, VPP\_FLAG 输出高电平, 芯片将进入烧写/校验模式
  - ◇ P12 在烧写 OTP 时需要 12V 电压
  - ◇ 在正常工作模式下, P12 可以做外部复位脚, 也可以做普通输入口, 通过 OPBIT[3] 进行选择; 无论是做外部复位脚还是输入口, P12 均内置有 50Kohm 左右的上拉电阻
  - ◇ 输入有施密特和滤波电路
- F、OSCIN/带上拉、下拉、开漏输出的 IO 口 (P10)
  - ◇ OPBIT[1:0]=00/01 时, P10 用于晶体振荡或外接 RC 振荡, 不能作为 IO 口
  - ◇ OPBIT[1:0]=10/11 时, 芯片工作于内部 RC 模式, P10 可作为 IO 口使用
  - ◇ 输出有非交叠控制电路
  - ◇ 输入有施密特和滤波电路
  - ◇ 上拉电阻阻值 30Kohm (@5V TT)
  - ◇ 下拉电阻阻值 30Kohm (@5V TT)
  - ◇ open drain 输出时, 输出驱动的 PMOS 管始终关闭
- G、OSCOUT/带上拉、下拉、开漏输出的 IO 口 (P11)
  - ◇ OPBIT[1:0]=00/01 时, P11 用于晶体振荡或外接 RC 振荡, 不能作为 IO 口
  - ◇ OPBIT[1:0]=10/11 时, 芯片工作于内部 RC 模式, P11 可作为 IO 口使用
  - ◇ 输出有非交叠控制电路
  - ◇ 输入有施密特和滤波电路
  - ◇ 上拉电阻阻值 30Kohm (@5V TT)
  - ◇ 下拉电阻阻值 30Kohm (@5V TT)
  - ◇ open drain 输出时, 输出驱动的 PMOS 管始终关闭

## 2.8 Basic Timer (BT)

BT 有如下功能

- ◇ 用于等待振荡稳定的 4096 脉冲计数
- ◇ 作为 WDT 的计数器
- ◇ 通过访问 BTCNT (只读) 来定时
- ◇ 4 种分频 (4096、1024、256、128)

与 BT 相关的寄存器有 BTCON、BTCNT。

BTCON 用于选择输入时钟频率，清 BT 计数器和分频器，开关 WDT。

复位时 BTCON 为 08H，此时打开 WDT，并将分频系数设为 256。若要关闭 WDT，需将 WDTE[3:0] 设为 1010B。

对 BTCLR 写“1”可将 BTCNT 的计数值清零；对 DVCLR 写“1”可对 BT 的分频器清零。

当 WDTE[3:0]不是 1010B 时，只要 BT 计数溢出就会引起系统复位。通常必须每隔一段时间对 BTCLR 写“1”，即对 BTCNT 清零，这样才能避免计数器溢出。如果系统因外部干扰或其他错误条件而出现异常，BTCNT 清零的动作不能按时完成，就会自动触发系统复位。

BT 还是保证振荡器起振稳定的计数器。不论是上电复位还是从 STOP 唤醒，BT 都会从 00H 计数到 0FH，然后系统完成复位或从 STOP 恢复。由于上电复位时 BTPS[1:0]=01（256 分频），所以等待时间是  $16 \times 256 = 4096$  时钟周期；而从 STOP 唤醒时 BTPS 的值由程序设定，等待时间由分频系数决定。

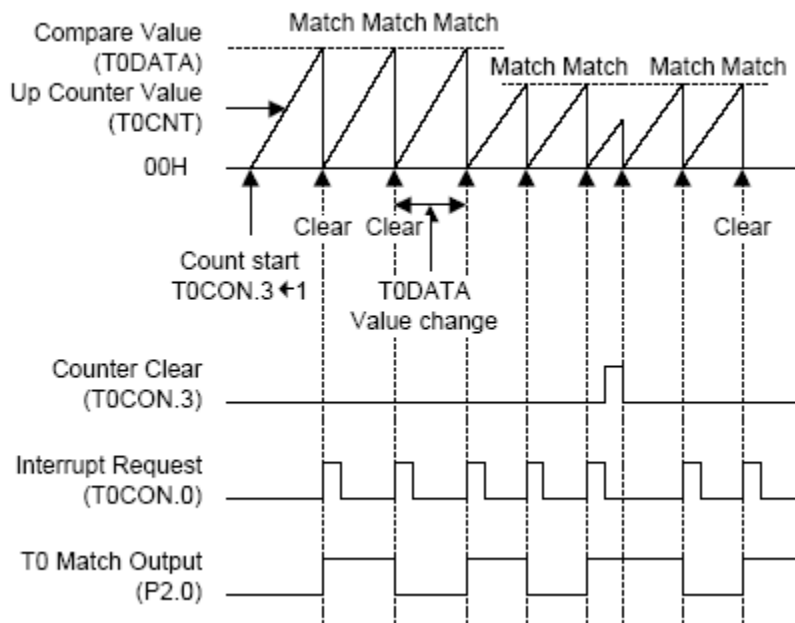
## 2.9 Timer0 (T0)

Timer0 有如下功能模块

- ◇ 分频器（分频系数 4096、256、8、1）
- ◇ 8 位计数器 T0CNT（只读）、8 位比较数据寄存器 T0DATA
- ◇ 控制寄存器 T0CON

T0CON 用于对 T0CNT 清零，开 T0 中断，并包括 T0 中断标志位。

T0CNT 在计数的过程中，如果  $T0CNT = T0DATA$  就会令 TOF 置 1（如果 TOE=1 就会触发 T0INT 中断），且 T0CNT 复位。例如，设置 T0DATA 为 10H，当 T0CNT 计数到 10H 时，触发 T0INT 中断，同时 T0CNT 清零重新计数。整个过程参见下图示意。



另外，如果设置 P20 口为 T0 输出，则 T0CNT match T0DATA 的同时 P20 会反一次相。可以计算出 P20 的周期为  $2 \times PS \times T0DATA \times T_{sys}$ （PS 是分频系数）。

当 T0DATA 设为 00H 时，P20 永远不反转，也不会有中断请求。

T0DATA 的值在更新的那一个周期即有效。

## 2.10 PWM

PWM 包含以下功能模块

- ◇ 分频器（分频系数 64、8、2、1）
- ◇ PWM 计数器 PWMCNT（不可访问）
- ◇ 6 位 PWM 参考数据寄存器 PWMDATA[7:2]
- ◇ 2 位 PWM 扩展数据寄存器 PWMDATA[1:0]
- ◇ PWM 控制寄存器 PWMCON

### 2.10.1 PWMCNT

PWM 的基本周期为  $64 * PS * T_{sys}$ 。

PWMCNT 在计数过程中与 PWMDATA[7:2] 比较, 当  $PWMCNT[5:0] < PWMDATA[7:2]$  时, PWM 输出 1, 其他时候输出 0。若要得到更高的分辨率, 可通过设置 PWMDATA[1:0] 来调整。

### 2.10.2 PWMDATA

PWMDATA 决定 PWM 输出的占空比。

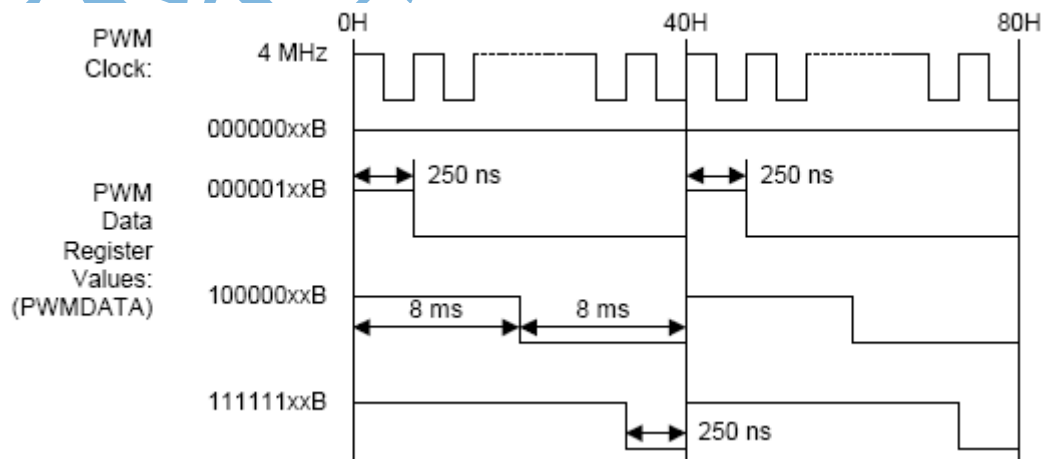
### 2.10.3 PWM 功能描述

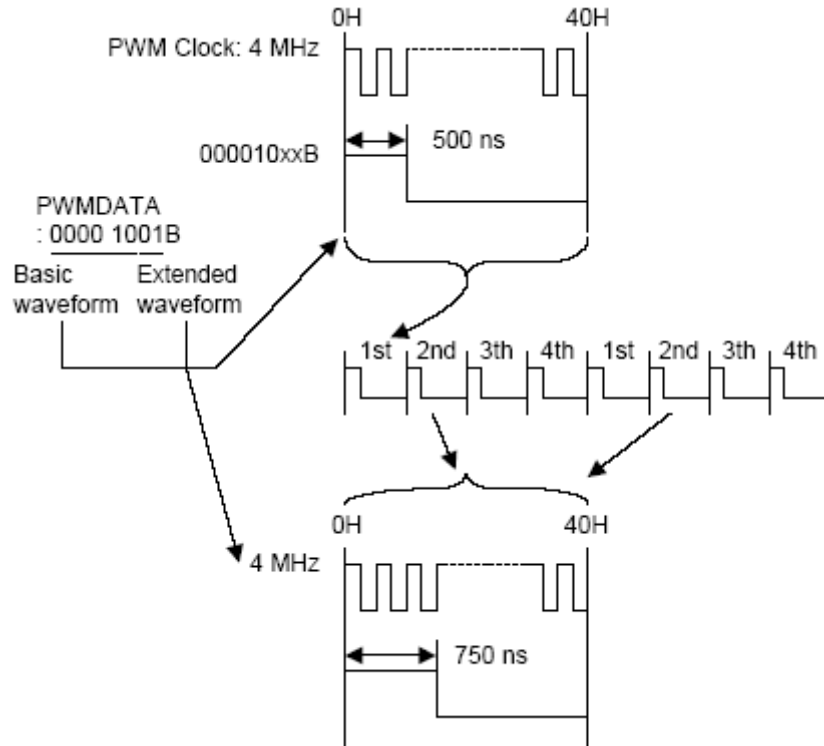
当  $PWMCNT[5:0] = PWMDATA[7:2]$  时 PWM 输出由高变低,  $PWMCNT[5:0]$  溢出时 PWM 输出由低变高, 这样就决定了 PWM 的基本占空比。对 PWMDATA 的设置并不是立刻生效, 而是等到 PWMCNT 计数器的 6bit 或 8bit 溢出时才生效。

PWMCNT 计数器的 8bit 溢出时会将 PWMIF 置 1, 如果此时  $PWMIE = 1$ , 则会触发 PWM 中断。

PWMCNT[7:6] 同 PWMDATA[1:0] 比较, 从而决定 PWM 输出是否需要延展一个时钟的宽度, 见下表

PWMDATA[1:0]	需要延展的周期 PWMCNT[7:6] 的值
00	-
01	01
10	00、10
11	00、01、10





## 2.11 ADC

20E24B 有一个 10 位 9 通道 ADC，主要特性如下：

- ◇ 逐次逼近的模数转换器
- ◇ ADC 状态控制寄存器 (ADCON) 地址: \$27H
- ◇ 10 位 AD 转换数据输出 (ADDATA) 地址: ADDATAH@\$28H, ADDATAL@\$29H
- ◇ 9 通道的模拟信号输入 (ADC0-ADC8)
- ◇ ADC 输入时钟最大频率  $f_{ADC(MAX)}=4MHz$
- ◇ A/D 转换一次周期为 51 个  $1/f_{ADC}$
- ◇ 10 位分辨率
- ◇ 8 位精度
- ◇ 静态功耗小于 0.1uA

由于 ADC 的输入通道与 IO 口复用，因此在使用 ADC 时，禁止将同个引脚既作为模拟输入又作为数字输入。

对 A/D 状态及控制寄存器 ADCON 中 ADCE 写入 1，就启动一个 A/D 转换过程；则所选择的输入通道的模拟信号的 A/D 转换 51 个周期( $1/f_{ADC}$ )执行一次，例如：当 ADC 输入时钟频率为  $F_{ADC}=4MHz$  时，即 250ns，A/D 转换一次周期为  $51*250ns=12.75us$ 。A/D 一次转换结束后，EOC 位置 1，并且停止 A/D 转换。

A/D 数据寄存器 ADDRH 是一个 8 位寄存器，它用于存放 A/D 转换的结果的高 8 位；ADDRL 是一个 2 位寄存器，它用于存放 A/D 转换结果的低 2 位。一旦当 A/D 状态及控制寄存器 ADCON 的转换结果标志 EOC 被置“1”时，则 ADDRH，ADDRL 的内容就会被更新，即由现行转换结果取代了原来的存放的转换结果。

## 2.12 OPBIT

OPBIT 是 OTP ROM 中的一个特殊字节，用于配置 20E24B 的某些特殊功能。OPBIT 在烧写程

序时通过烧写器一并写入。

- .7 ENCR
  - 0: 程序加密
  - 1: 不加密
- .6 保留
- .5 LVRS
  - 0: LVR 2.1V
  - 1: LVR 3.6V
- .4 RES
  - 0: P12 上拉电阻无效
  - 1: P12 上拉电阻有效
- .3 RST
  - 0: P12 作为 IO 口
  - 1: P12 作为外复位引脚
- .2 LVRE
  - 0: LVR 默认开启
  - 1: LVR 默认关闭
- .0-.1 OSC/RC
  - 00: 系统选用晶振/陶振
  - 01: 系统选用 2MHz RC 振荡 (RC1)
  - 10: 系统选用 4MHz RC 振荡 (RC2)
  - 11: 系统选用 8MHz RC 振荡 (RC3)

## 第 3 章 H05 指令集

### 3.1 简介

20E24B 采用 HC05 指令集。指令集详细资料见本公司手册《HC05 指令集》。

**注意：20E24B 不支持 MUL 指令**

### 3.2 CPU（中央处理器）

HC05 的 CPU 框图见图 1。

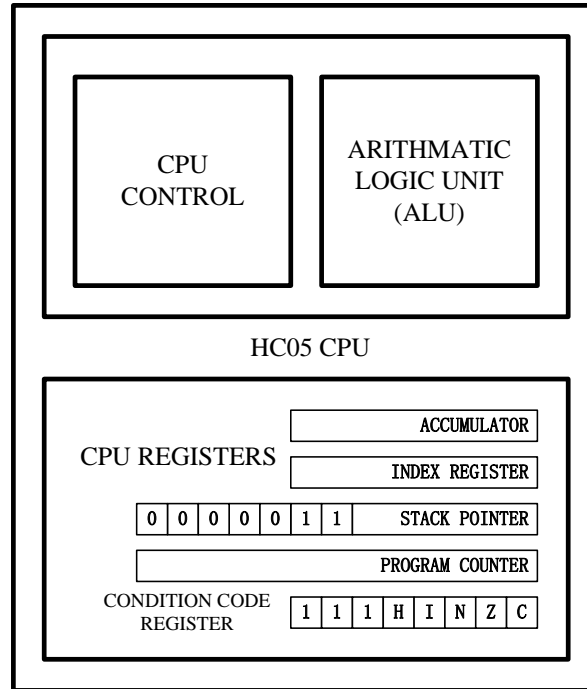


图1

### 3.2.1 ALU（算数逻辑单元）

ALU（算数逻辑单元）用于执行指令集中的算术和逻辑运算。指令译码器对指令进行译码，根据译码的结果，对应的算术运算电路便开始完成相应的运算。

大多数二进制算术运算是基于“加法”操作的。比如，“减法”就是带负号的“加法”。

“乘法”实际上是在 CPU 控制单元控制下的一系列加法和移位操作的组合，这一过程需要 11 个指令周期。

### 3.2.2 CPU CONTROL（CPU 控制单元）

CPU 控制单元对 ALU 的逻辑电路的时序进行控制。CPU 控制单元的“指令译码器”对指令进行译码，取得操作码（opcode），opcode 再经过译码以确定指令需要经过多少个操作，以及这些操作执行的顺序。一个指令执行完后，CPU 控制单元会读取下一条指令。

### 3.2.3 CPU 寄存器

CPU 寄存器在 CPU 内部，不是普通的存储单元（如 RAM、ROM）。HC05 的 CPU 寄存器包括累加器（ACCUMULATOR，简称 A）、变址寄存器（INDEX REGISTER，简称 X）、状态寄存器（CONDITION CODE REGISTER，简称 CCR）、PC 指针（PROGRAM COUNTER，简称 PC）、堆栈指针（STACK POINTER，简称 SP）。

#### 3.2.3.1 累加器

累加器是一个 8 位寄存器，用于存储操作数、算术运算的结果。

#### 3.2.3.2 变址寄存器

变址寄存器在变址寻址方式时使用，或作为运算的辅助寄存器使用。变址寄存器是一个 8 位寄存器。

在变址寻址指令中，变址寄存器提供一个地址偏移量，以计算有效地址。

#### 3.2.3.3 状态寄存器

状态寄存器包括一个中断屏蔽位（I）和 4 个状态位（H、N、Z、C），这 4 个状态位标志运算结

果及 CPU 的状态。

- ◇ 半进位 (H)
- ◇ 中断屏蔽 (I)
- ◇ 负标志 (N)
- ◇ 零标志 (Z)
- ◇ 进位/借位 (C)

#### 半进位 (H)

半进位 (H) 标志用于 BCD 运算, ADD 和 ADC 指令会影响 H 标志。当低 4 位向高 4 位发生进位时 H 标志被置 1。2 个二进制 BCD 数做加法后, 通过 H 标志可判断是否要对加法的结果进行 BCD 调整。

#### 中断屏蔽位 (I)

中断屏蔽位 (I) 功能是屏蔽或允许来自中断源的中断请求。当 I 为 1 时 CPU 不会相应中断, I 为 0 时 CPU 能响应中断。在 CPU 响应中断的过程中, 即在压栈完成之后, 取得中断向量之前, I 会被自动置 1。(禁止中断嵌套是 CPU 默认的方式, 若用户需要中断嵌套, 可在中断服务程序中用 CLI 指令将 I 清 0)

中断服务程序执行完后, 用 RTI 指令退出中断, A、X、CCR 通过弹栈恢复到响应中断前的值 (I 总是恢复为 0)。

系统复位后, I 被置 1, 若要相应中断, 必须在之前使用 CLI 指令将 I 清 0。

#### 负标志 (N)

当算术、逻辑、数据操作的结果为负时, 负标志位 (N) 会被置 1, 反之则清 0。一个 8 位二进制数的最高位为 1 则被认为是负值。

#### 零标志 (Z)

当算术、逻辑、数据操作的结果为零时, 零标志位 (Z) 会被置 1, 反之则清 0。

#### 进位/借位标志 (C)

当加法发生进位或减法发生借位时, 进位/借位标志位 (C) 会被置 1, 反之则清 0。移位和循环移位指令也会用到 C 标志。

### 3.2.3.4 PC 指针

PC 指针是一个 16 位的寄存器, 存储指令的操作码和操作数的地址。HC05 指令集最大的寻址空间为 64K (65536)。多数 HC05 的产品只用到这个寻址空间的一部分, 通常, 在这样的情况下, PC 未用的高位为 0。例如, BM22P64 的寻址空间为 0000H-1FFFH, 这时 PC 的 13、14、15 位总为 0。

### 3.2.3.5 堆栈指针

HC05 的堆栈指针 (SP) 的低 6 位是可变的, 也就是说 SP 的值总是在 00C0H-00FFH 范围内, 能够访问的地址空间为 64 字节, 这也是最大的可用堆栈的大小。

系统复位或执行 RSP 指令后, SP 被置成 00FFH, 压栈操作时 SP 减小, 弹栈时 SP 变大。有两类操作会使用堆栈, 子程序调用 (BSR、JSR) 使用 2 字节堆栈, 中断使用 5 字节堆栈。

在 HC05 系统中并不会明确区分用户 RAM 和堆栈空间, 如果用户程序嵌套情况比较少, 那么被定义为堆栈空间的地址 (00C0H-00FFH) 中没有使用的字节完全可以当成用户 RAM 使用。用户程序如果嵌套使用普遍, 则需留意堆栈空间的使用情况, 避免出现堆栈溢出的情况 (CPU 不会判断堆栈是否溢出, 当 SP 超出 00C0H 时, 会回到 00FFH)。在某些小容量 RAM 的产品中 (如 BM35P02, 用户 RAM 和堆栈空间共用地址空间 00E0H-00FFH), 整个 RAM 区甚至都不到 64 字节, 这时更需要注意程序嵌套的使用。

### 3.3 寻址方式

HC05 的寻址能力很强，通过不同的寻址方式，一条指令中存取数据的方式最多可达 6 种，正因如此，62 条基本指令可扩展为 210 条指令。

HC05 的 6 种寻址方式为：

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

隐含寻址的指令实际并不访问存储器，所以这类指令都是单字节指令。直接寻址指令能访问 0000H-00FFH 地址空间，是 2 字节指令。扩展寻址指令能访问所有 64K 地址空间，是 3 字节指令。变址寻址方式能访问存储空间中任何地方的数据表格、代码转换表格和顺序表格。无偏移量变址寻址指令是单字节指令，8 位偏移量变址寻址指令是 2 字节指令，16 位偏移量变址寻址指令是 3 字节指令。

后面的章节包括各种寻址方式的概述和示例。每条指令的详细描述见“指令集详解”。

在汇编程序源代码的示例中使用了几种符号前缀来标识各类数据，这些符号有

1. 无前缀的数表示 10 进制数，如 LDA 25
2. “\$”前缀的数表示 16 进制数，如 LDA \$19
3. “@”前缀的数表示 8 进制数，如 LDA @31
4. “%”前缀的数表示 2 进制数，如 LDA %00011001
5. “#”前缀表示立即数，可与以上 4 种前缀组合使用，如 LDA #\$19

#### 3.3.1 隐含寻址

对于隐含寻址指令，操作码中包含了所有 CPU 需要的信息，因而不需要额外的操作数。这类指令的操作对象通常都是 A 或 X，并且都是单字节指令。

##### 汇编代码的 Listing

```
0300 4c INCA ;Increment accumulator
```

注：Listing 是汇编源代码编译后产生的 LST 文件，0030 是指令的地址，4c 是指令的机器码，随后是源代码的内容。

##### 执行顺序

```
$0300 $4C [1], [2], [3]
```

##### 说明

[1] CPU 读取操作码\$4C

[2]和[3] CPU 取 A 寄存器的值，加 1，再将结果存回 A 寄存器，并根据情况设置 CCR 标志  
下面的表格列出了 HC05 中所有的隐含寻址指令。

指令	助记符
算术左移	ASLA ASLX
算术右移	ASRA ASRX
C 标志清零	CLC
I 标志清零	CLI
清零	CLRA CLRX
取反	COMA COMX
减一	DECA DECX
加一	INCA INCX
逻辑左移	LSLA LSLX

逻辑右移	LSRA LSRX
乘法（暂不支持）	MUL
取补	NEGA NEGX
空操作	NOP
循环左移	ROLA ROLX
循环右移	RORA RORX
SP 复位	RSP
中断返回	RTI
子程序返回	RTS
C 标志置位	SEC
I 标志置位	SEI
进 STOP 模式	STOP
软中断	SWI
将 A 的值传到 X	TAX
零测试	TSTA TSTX
将 X 的值传到 A	TXA
进 WAIT 模式	WAIT

### 3.3.2 立即寻址

在立即寻址指令中，操作数是紧跟在操作码后的那个字节。这类指令所访问的一个常数称为“立即数”。立即寻址指令是 2 字节指令，包括操作码和一个字节操作数。

#### 汇编代码的 Listing

```
0300 a6 03 LDA #03 ;Load accumulator with an immediate value
```

#### 执行顺序

```
$0300 $A6 [1]  
$0301 $03 [2]
```

#### 说明

[1] CPU 读取操作码\$A6

[2] CPU 从地址\$0301 读取立即数\$03，存入 A 寄存器  
下面的表格列出了 HC05 中所有的立即寻址指令。

指令	助记符
带进位的加法	ADC
加法	ADD
逻辑与	AND
位测试	BIT
A 寄存器比较	CMP
X 寄存器比较	CPX
逻辑异或	EOR
A 寄存器存数	LDA
X 寄存器存数	LDX
逻辑或	ORA
带借位的减法	SBC
减法	SUB

### 3.3.3 扩展寻址

在扩展寻址指令中，操作数是紧跟在操作码后的两个字节（绝对地址）。扩展寻址可访问 64K 寻址空间的任一地址。扩展寻址指令是 3 字节指令，包括操作码和 2 个字节操作数。

#### 汇编代码的 Listing

```
0300 c6 06 e5 LDA $06E5 ;Load accumulator from extended address
```

**执行顺序**

\$0300 \$C6 [1]  
 \$0301 \$06 [2]  
 \$0302 \$E5 [3] and [4]

**说明**

- [1] CPU 读取操作码\$C6
- [2] CPU 从地址\$0301 读取\$06, 作为要访问的地址高 8 位
- [3] CPU 从地址\$0302 读取\$E5, 作为要访问的地址低 8 位
- [4] CPU 访问地址\$06E5, 取数并存到 A 寄存器

下面的表格列出了 HC05 中所有的扩展寻址指令。

指令	助记符
带进位的加法	ADC
加法	ADD
逻辑与	AND
位测试	BIT
A 寄存器比较	CMP
X 寄存器比较	CPX
逻辑异或	EOR
跳转	JMP
调用子程序	JSR
A 寄存器存数	LDA
X 寄存器存数	LDX
逻辑或	ORA
带借位的减法	SBC
A 寄存器取数	STA
X 寄存器取数	STX
减法	SUB

### 3.3.4 直接寻址

直接寻址方式与扩展寻址方式很类似, 不同点在于高 8 位地址总是为\$00, 只有低 8 位地址作为操作数。直接寻址指令使用户能够方便地访问地址空间的低 256 字节, 这 256 字节的空间也被称为直接寻址空间 (通常 RAM 和控制寄存器都定义在这一空间内)。直接寻址方式既节省指令字长也节省指令周期。直接寻址指令是 2 字节指令, 包括操作码和 1 个字节操作数。

**汇编代码的 Listing**

```
0300 b6 50 LDA $50 ;Load accumulator from direct address
```

**执行顺序**

\$0300 \$B6 [1]  
 \$0301 \$50 [2] and [3]

**说明**

- [1] CPU 读取操作码\$B6
- [2] CPU 从地址\$0301 读取\$50, 作为要访问的地址低 8 位, 高 8 位地址为\$00
- [3] CPU 访问地址\$0050, 取数并存到 A 寄存器

下面的表格列出了 HC05 中所有的直接寻址指令。

指令	助记符
带进位的加法	ADC
加法	ADD
逻辑与	AND
算术左移	ASL

算术右移	ASR
第 n 位清零	BCLR
位测试	BIT
第 n 位为 0 则跳转	BRCLR
第 n 位为 1 则跳转	BRSET
第 n 位置位	BSET
清零	CLR
A 寄存器比较	CMP
取反	COM
X 寄存器比较	CPX
减一	DEC
逻辑异或	EOR
加一	INC
跳转	JMP
调用子程序	JSR
A 寄存器存数	LDA
X 寄存器存数	LDX
逻辑左移	LSL
逻辑右移	LSR
取补	NEG
逻辑或	ORA
循环左移	ROL
循环右移	ROR
带借位的减法	SBC
A 寄存器取数	STA
X 寄存器取数	STX
减法	SUB
零测试	TST

### 3.3.5 变址寻址

在变址寻址方式中，有效地址（简称 EA）由 2 个因素决定：

1. 当前变址寄存器 X 的值
2. 操作码的地址偏移量

变址寻址由分为 3 类，即

- ◇ 无偏移量变址寻址
- ◇ 8 位偏移量编制寻址
- ◇ 16 位偏移量编制寻址

### 3.3.6 无偏移量变址寻址

无偏移量变址寻址的有效地址就是 X 寄存器的值，可访问地址空间的低 256 位（直接寻址空间）。无偏移量变址寻址指令是单字节指令。

汇编代码的 Listing

```
0300 f6 LDX ,x ;Load accumulator from location
;pointed to by index reg (no offset)
```

执行顺序

```
$0300 $F6 [1], [2], [3]
```

说明

- [1] CPU 读取操作码 \$F6
- [2] CPU 计算有效地址（X 的值加 \$0000）

[3] CPU 访问有效地址，取数并存到 A 寄存器  
 下面的表格列出了 HC05 中所有的无偏移量变址寻址指令。

指令	助记符
带进位的加法	ADC
加法	ADD
逻辑与	AND
算术左移	ASL
算术右移	ASR
位测试	BIT
清零	CLR
A 寄存器比较	CMP
取反	COM
X 寄存器比较	CPX
减一	DEC
逻辑异或	EOR
加一	INC
跳转	JMP
调用子程序	JSR
A 寄存器存数	LDA
X 寄存器存数	LDX
逻辑左移	LSL
逻辑右移	LSR
取补	NEG
逻辑或	ORA
循环左移	ROL
循环右移	ROR
带借位的减法	SBC
A 寄存器取数	STA
X 寄存器取数	STX
减法	SUB
零测试	TST

### 3.3.7 8 位偏移量变址寻址

8 位偏移量变址寻址的有效地址就是 X 寄存器的值与操作数（8 位偏移量）之和（视为 2 个无符号数相加）。这种寻址方式能够方便地实现查表功能（如在有 N 个单元的表格中访问第 K 个单元）。要使用这种查表操作，数据表的起始地址必须在低 256 位地址空间（直接寻址空间）。这种寻址方式能访问到的最大地址是 \$01FE（\$00FF+\$00FF）。8 位偏移量变址寻址指令是 2 字节指令，包括操作码和 1 个字节操作数。

#### 汇编代码的 Listing

```
0300 e6 05 LDA $5,x ;Load accumulator from location
;pointed to by index reg (X) + $05
```

#### 执行顺序

```
$0300 $E6 [1]
$0301 $05 [2], [3], [4]
```

#### 说明

- [1] CPU 读取操作码 \$E6
- [2] CPU 从地址 \$0301 读取 \$05，作为基本地址的低 8 位，高 8 位为 \$00
- [3] CPU 计算有效地址（X 的值加基本地址 \$0005）
- [4] CPU 访问有效地址，取数并存到 A 寄存器

下面的表格列出了 HC05 中所有的 8 位偏移量变址寻址指令。

指令	助记符
带进位的加法	ADC
加法	ADD
逻辑与	AND
算术左移	ASL
算术右移	ASR
位测试	BIT
清零	CLR
A 寄存器比较	CMP
取反	COM
X 寄存器比较	CPX
减一	DEC
逻辑异或	EOR
加一	INC
跳转	JMP
调用子程序	JSR
A 寄存器存数	LDA
X 寄存器存数	LDX
逻辑左移	LSL
逻辑右移	LSR
取补	NEG
逻辑或	ORA
循环左移	ROL
循环右移	ROR
带借位的减法	SBC
A 寄存器取数	STA
X 寄存器取数	STX
减法	SUB
零测试	TST

### 3.3.8 16 位偏移量变址寻址

16 位偏移量变址寻址的有效地址就是 X 寄存器的值与 2 个操作数（16 位偏移量）之和（视为无符号数相加）。这种寻址方式能够方便地实现查表功能（如在有 N 个单元的表格中访问第 K 个单元），并能访问全部 64K 地址空间。16 位偏移量变址寻址指令是 3 字节指令，包括操作码和 2 个字节操作数。

#### 汇编代码的 Listing

```
0300 d6 07 00 LDA $0700,x ;Load accumulator from location
                                ;pointed to by index reg (X) + $0700
```

#### 执行顺序

```
$0300 $D6 [1]
$0301 $07 [2]
$0302 $00 [3], [4], [5]
```

#### 说明

- [1] CPU 读取操作码 \$D6
- [2] CPU 从地址 \$0301 读取 \$07，作为基本地址的高 8 位
- [3] CPU 从地址 \$0302 读取 \$00，作为基本地址的低 8 位
- [4] CPU 计算有效地址（X 的值加基本地址 \$0700）

[4] CPU 访问有效地址，取数并存到 A 寄存器

下面的表格列出了 HC05 中所有的 16 位偏移量变址寻址指令。

指令	助记符
带进位的加法	ADC
加法	ADD
逻辑与	AND
位测试	BIT
A 寄存器比较	CMP
X 寄存器比较	CPX
逻辑异或	EOR
跳转	JMP
调用子程序	JSR
A 寄存器存数	LDA
X 寄存器存数	LDX
逻辑或	ORA
带借位的减法	SBC
A 寄存器取数	STA
X 寄存器取数	STX
减法	SUB

### 3.3.9 相对寻址

相对寻址方式仅用于条件跳转指令。条件跳转指令是一个 2 字节指令，包括操作码和 1 个操作数，这个操作数就是跳转发生时的地址偏移量（视为有符号数），跳转可以向后也可以向前。如果条件判断为真，则跳转发生；否则，程序依次执行下一条指令。

用户在写汇编源代码时通常用 Label 标识跳转的目的，而由编译器计算偏移量。

#### 汇编代码的 Listing

```
0300 27 rr BEQ DEST ;Branch to DEST if Z = 1
; (branch if equal or zero)
```

#### 执行顺序

```
$0300 $27 [1]
$0301 $rr [2], [3]
```

#### 说明

[1] CPU 读取操作码\$27

[2] CPU 从地址\$0301 读取\$rr，作为跳转偏移量

[3] CPU 判断 Z 标志的状态，若 Z=1，则计算新的 PC 值发生跳转；否则 PC=PC+2，即执行下一条指令

下面的表格列出了 HC05 中所有的相对寻址指令。

指令	助记符
无进位则跳转	BCC
进位则跳转	BCS
等于则跳转	BEQ
无半进位则跳转	BHCC
半进位则跳转	BHCS
大于则跳转	BHI
大于等于则跳转	BHS
IRQ 为高则跳转	BIH
IRQ 为低则跳转	BIL
小于则跳转	BLO
小于等于则跳转	BLS

中断未屏蔽则跳转	BMC
值为负则跳转	BMI
中断屏蔽则跳转	BMS
不等于则跳转	BNE
值为正则跳转	BPL
无条件跳转	BRA
第 n 位为 0 则跳转	BRCLR
第 n 位为 1 则跳转	BRSET
永不跳转	BRN
跳转到子程序	BSR

**注 1:** BRCLR 和 BRSET 既可归入直接寻址指令，也可归入相对寻址指令，这是因为这两条指令的数据访问方式属于直接寻址，而 PC 指针的获得方式则属于相对寻址。

**注 2:** BIH 和 BIL 这在指令集中是存在的，但 IRQ 引脚在绝大多数产品中是没有的，所以这两条指令实际没有用。

### 3.4 按照功能分类的所有指令的汇总。

**表 1 寄存器/存储器指令**

功能	助记符	寻址方式																	
		立即寻址			直接寻址			扩展寻址			无偏移量变址寻址			8位偏移量变址寻址			16位偏移量变址寻址		
		操作码	字节数	指令周期	操作码	字节数	指令周期	操作码	字节数	指令周期	操作码	字节数	指令周期	操作码	字节数	指令周期	操作码	字节数	指令周期
A 寄存器存数	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
X 寄存器存数	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
A 寄存器取数	STA	-	-	-	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
X 寄存器取数	STX	-	-	-	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
加法	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
带进位的加法	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
减法	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
带借位的减法	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
逻辑与	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
逻辑或	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
逻辑异或	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
A 寄存器比较	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
X 寄存器比较	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
位测试	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
跳转	JMP	-	-	-	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
调用子程序	JSR	-	-	-	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

**表 2 读/写/修改指令**

功能	助记符	寻址方式														
		隐含寻址 (A)			隐含寻址 (X)			直接寻址			无偏移量变址寻址			8 位偏移量变址寻址		
		操作码	字节数	指令周期	操作码	字节数	指令周期	操作码	字节数	指令周期	操作码	字节数	指令周期	操作码	字节数	指令周期
加一	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
减一	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
清零	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
取反	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
取补	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
循环左移	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
循环右移	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
逻辑左移	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
逻辑右移	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
算术右移	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
零测试	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5
乘法 (暂不支持)	MUL	42	1	11	-	-	-	注 1	-	-	-	-	-	-	-	-
第 n 位清零	BCLR	-	-	-	-	-	-	注 2	2	5	-	-	-	-	-	-
第 n 位置位	BSET	-	-	-	-	-	-	注 2	2	5	-	-	-	-	-	-

注 1: MUL 暂不支持。

注 2: BCLR、BSET 按位的不同各自分为 8 个机器码

**表 3 条件跳转指令**

功能	助记符	相对寻址		
		操作码	字节数	指令周期
无条件跳转	BRA	20	2	3
永不跳转	BRN	21	2	3
大于则跳转	BHI	22	2	3
小于等于则跳转	BLS	23	2	3
无进位则跳转	BCC	24	2	3
大于等于则跳转	BHS	24	2	3
进位则跳转	BCS	25	2	3
小于则跳转	BLO	25	2	3
不等于则跳转	BNE	26	2	3
等于则跳转	BEQ	27	2	3
无半进位则跳转	BHCC	28	2	3
半进位则跳转	BHCS	29	2	3
值为正则跳转	BPL	2A	2	3
值为负则跳转	BMI	2B	2	3
中断未屏蔽则跳转	BMC	2C	2	3
中断屏蔽则跳转	BMS	2D	2	3
IRQ 为低则跳转	BIL	2E	2	3
IRQ 为高则跳转	BIH	2F	2	3
跳转到子程序	BSR	AD	2	6

注：BCC 和 BHS，BCS 和 BLO 实际上是相同的指令

**表 4 控制指令**

功能	助记符	隐含寻址		
		操作码	字节数	指令周期
将 A 的值传到 X	TAX	97	1	2
将 X 的值传到 A	TXA	9F	1	2
C 标志置位	SEC	99	1	2
C 标志清零	CLC	98	1	2
I 标志置位	SEI	9B	1	2
I 标志清零	CLI	9A	1	2
软中断	SWI	83	1	10
子程序返回	RTS	81	1	6
中断返回	RTI	80	1	9
SP 复位	RSP	9C	1	2
空操作	NOP	9D	1	2
进 STOP 模式	STOP	8E	1	2
进 WAIT 模式	WAIT	8F	1	2

### 3.5 HC05 指令集汇总

HC05 指令集有 62 个基本指令，每条基本指令根据寻址方式的不同需要不同的操作码。如果将每一个操作码视为一条单独的指令的话，HC05 实际有 210 条单独的指令。

下面的表格按字母次序列出全部指令。在这个表格中使用了以下符号。

#### 状态码

H	半进位位	◆	根据指令结果清零或置位
I	中断屏蔽位	—	不影响
N	负标志位	0	清零
Z	零标志位	1	置位
C	进位位		

#### 逻辑操作

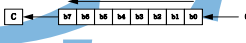
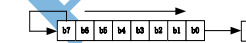
( )	寄存器的内容（例如	∨	逻辑或
(M)	表示地址 M 的内容	⊕	逻辑异或
←	取值	×	乘法
∧	逻辑与	-	取补（取负数）

#### CPU 寄存器

A	累加器	PC	PC 指针
CCR	状态寄存器	PCH	PC 指针高位
X	变址寄存器	PCL	PC 指针低位
M	存储器地址	SP	堆栈指针
		REL	相对地址（一字节）

寻址方式	简写	操作数
隐藏寻址	INH	无
立即寻址	IMM	ii
直接寻址	DIR	dd
（条件跳转）		dd rr
扩展寻址	EXT	hh ll
无偏移量变址寻址	IX	无
8 位偏移量变址寻址	IX1	ff
16 位偏移量变址寻址	IX2	ee ff
相对寻址	REL	rr

**表 5 指令集总表 (1/8)**

指令	功能	描述	对 CCR 的影响					寻址方式	操作码	操作数	指令周期		
			H	I	N	Z	C						
ADC #opr	带进位的加法	$A \leftarrow (A) + (M) + (C)$						IM	A9	ii	2		
ADC opr								M	B9	dd	3		
ADC opr					◆	—	◆	◆	DIR	C9	hh ll	4	
ADC opr,X									EXT	D9	ee ff	5	
ADC opr,X									IX2	E9	ff	4	
ADC ,X									IX1	F9		3	
ADD #opr	加法	$A \leftarrow (A) + (M)$						IM	AB	ii	2		
ADD opr								M	BB	dd	3		
ADD opr					◆	—	◆	◆	DIR	CB	hh ll	4	
ADD opr,X									EXT	DB	ee ff	5	
ADD opr,X									IX2	EB	ff	4	
ADD ,X									IX1	FB		3	
AND #opr	逻辑与	$A \leftarrow (A) \wedge (M)$						IM	A4	ii	2		
AND opr								M	44	dd	3		
AND opr					—	—	◆	◆	DIR	C4	hh ll	4	
AND opr,X									EXT	D4	ee ff	5	
AND opr,X									IX2	E4	ff	4	
AND ,X									IX1	F4		3	
ASL opr	算术左移 (同 LSL)							DIR	38	dd	5		
ASLA									INH	48		3	
ASLX					—	—	◆	◆	◆	INH	58		3
ASL opr,X										IX1	68	ff	6
ASL ,X										IX	78		5
ASR opr	算术右移								DIR	37	dd	5	
ASRA									INH	47		3	
ASRX					—	—	◆	◆	◆	INH	57		3
ASR opr,X										IX1	67	ff	6
ASR ,X										IX	77		5
BCC rel	无进位则跳转 (同 BHS)	$PC \leftarrow (PC) + 2 + rel ? C = 0$	—	—	—	—	—	REL	24	rr	3		

**表 5 指令集总表 (2/8)**

指令	功能	描述	对 CCR 的影响					寻址方式	操作码	操作数	指令周期
			H	I	N	Z	C				
BCLR n,opr	第 n 位清零	$M_n \leftarrow 0$	—	—	—	—	—	DIR(b0)	11	dd	5
								DIR(b1)	13	dd	5
								DIR(b2)	15	dd	5
								DIR(b3)	17	dd	5
								DIR(b4)	19	dd	5
								DIR(b5)	1B	dd	5
								DIR(b6)	1D	dd	5
								DIR(b7)	1F	dd	5
BCS rel	进位则跳转 (同 BLO)	$PC \leftarrow (PC) + 2 + rel ? C = 1$	—	—	—	—	—	REL	25	rr	3
BEQ rel	等于则跳转	$PC \leftarrow (PC) + 2 + rel ? Z = 1$	—	—	—	—	—	REL	27	rr	3
BHCC rel	无半进位则跳转	$PC \leftarrow (PC) + 2 + rel ? H = 0$	—	—	—	—	—	REL	28	rr	3
BHCS rel	半进位则跳转	$PC \leftarrow (PC) + 2 + rel ? H = 1$	—	—	—	—	—	REL	29	rr	3
BHI rel	大于则跳转	$PC \leftarrow (PC) + 2 + rel ? (C \vee Z) = 0$	—	—	—	—	—	REL	22	rr	3
BHS rel	大于等于则跳转 (同 BCC)	$PC \leftarrow (PC) + 2 + rel ? C = 0$	—	—	—	—	—	REL	24	rr	3
BIH rel	IRQ 为高则跳转	$PC \leftarrow (PC) + 2 + rel ? IRQ = 1$	—	—	—	—	—	REL	2F	rr	3
BIL rel	IRQ 为低则跳转	$PC \leftarrow (PC) + 2 + rel ? IRQ = 0$	—	—	—	—	—	REL	2E	rr	3
BIT #opr	位测试	$(A) \wedge (M)$	—	—	◆	◆	—	IMM	A5	ii	2
BIT opr								DIR	B5	dd	3
BIT opr								EXT	C5	hh ll	4
BIT opr,X								IX2	D5	ee ff	5
BIT opr,X								IX1	E5	ff	4
BIT ,X								IX	F5		3
BLO rel	小于则跳转 (同 BCS)	$PC \leftarrow (PC) + 2 + rel ? C = 1$	—	—	—	—	—	REL	25	rr	3
BLS rel	小于等于则跳转	$PC \leftarrow (PC) + 2 + rel ? (C \vee Z) = 1$	—	—	—	—	—	REL	23	rr	3
BMC rel	中断未屏蔽则跳转	$PC \leftarrow (PC) + 2 + rel ? I = 0$	—	—	—	—	—	REL	2C	rr	3
BMI rel	值为负则跳转	$PC \leftarrow (PC) + 2 + rel ? N = 1$	—	—	—	—	—	REL	2B	rr	3
BMS rel	中断屏蔽则跳转	$PC \leftarrow (PC) + 2 + rel ? I = 1$	—	—	—	—	—	REL	2D	rr	3
BNE rel	不等于则跳转	$PC \leftarrow (PC) + 2 + rel ? Z = 0$	—	—	—	—	—	REL	26	rr	3
BPL rel	值为正则跳转	$PC \leftarrow (PC) + 2 + rel ? N = 0$	—	—	—	—	—	REL	2A	rr	3
BRA rel	无条件跳转	$PC \leftarrow (PC) + 2 + rel$	—	—	—	—	—	REL	20	rr	3


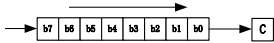
**表 5 指令集总表 (3/8)**

指令	功能	描述	对 CCR 的影响					寻址方式	操作码	操作数	指令周期
			H	I	N	Z	C				
BRCLR n opr rel	第 n 位为 0 则跳转	$PC \leftarrow (PC) + 3 + rel ? Mn=0$	—	—	—	—	◆	DIR(b0)	01	dd rr	5
								DIR(b1)	03	dd rr	5
								DIR(b2)	05	dd rr	5
								DIR(b3)	07	dd rr	5
								DIR(b4)	09	dd rr	5
								DIR(b5)	0B	dd rr	5
								DIR(b6)	0D	dd rr	5
								DIR(b7)	0F	dd rr	5
BRN rel	永不跳转	$PC \leftarrow (PC) + 2$	—	—	—	—	—	REL	21	rr	3
BRSET n,opr,rel	第 n 位为 1 则跳转	$PC \leftarrow (PC) + 3 + rel ? Mn=1$	—	—	—	—	◆	DIR(b0)	00	dd rr	5
								DIR(b1)	02	dd rr	5
								DIR(b2)	04	dd rr	5
								DIR(b3)	06	dd rr	5
								DIR(b4)	08	dd rr	5
								DIR(b5)	0A	dd rr	5
								DIR(b6)	0C	dd rr	5
								DIR(b7)	0E	dd rr	5
BSET n,opr	第 n 位置位	$Mn \leftarrow 1$	—	—	—	—	—	DIR(b0)	10	dd	5
								DIR(b1)	12	dd	5
								DIR(b2)	14	dd	5
								DIR(b3)	16	dd	5
								DIR(b4)	18	dd	5
								DIR(b5)	1A	dd	5
								DIR(b6)	1C	dd	5
								DIR(b7)	1E	dd	5
BSR rel	跳转到子程序	$PC \leftarrow (PC) + 2$ Push (PCL); $SP \leftarrow (SP) - 1$ Push (PCH); $SP \leftarrow (SP) - 1$ $PC \leftarrow (PC) + rel$	—	—	—	—	—	REL	A D	rr	6
CLC	C 标志清零	$C \leftarrow 0$	—	—	—	—	0	INH	98		2
CLI	I 标志清零	$I \leftarrow 0$	—	0	—	—	—	INH	9A		2
CLR opr CLRA CLR X CLR opr,X CLR ,X	清零	$M \leftarrow \$00$						DIR	3F	dd	5
		$A \leftarrow \$00$						INH	4F		3
		$X \leftarrow \$00$	—	—	0	1	—	INH	5F		3
		$M \leftarrow \$00$						IX1	6F	ff	6
		$M \leftarrow \$00$						IX	7F		5



**表 5 指令集总表 (4/8)**

指令	功能	描述	对 CCR 的影响					寻址方式	操作码	操作数	指令周期
			H	I	N	Z	C				
CMP #opr	A 寄存器比较	(A)-(M)	—	—	◆	◆	◆	IM	A1	dd	2
CMP opr								DIR	B1	dd	3
CMP opr								EXT	C1	hh ll	4
CMP opr,X								IX2	D1	ee ff	5
CMP opr,X								IX1	E1	ff	4
CMP ,X								IX	F1		3
COM opr	取反	$M \leftarrow \$FF-(M)$	—	—	◆	◆	1	DIR	33	dd	5
COMA		$A \leftarrow \$FF-(A)$						INH	43		3
COMX		$X \leftarrow \$FF-(X)$						INH	53		3
COM opr,X		$M \leftarrow \$FF-(M)$						IX1	63	ff	6
COM ,X		$M \leftarrow \$FF-(M)$						IX	73		5
CPX #opr	X 寄存器比较	(X)-(M)	—	—	◆	◆	◆	IM	A3	ii	2
CPX opr								DIR	B3	dd	3
CPX opr								EXT	C3	hh ll	4
CPX opr,X								IX2	D3	ee ff	5
CPX opr,X								IX1	E3	ff	4
CPX ,X								IX	F3		3
DEC opr	减一	$M \leftarrow (M)-1$	—	—	◆	◆	—	DIR	3A	dd	5
DECA		$A \leftarrow (A)-1$						INH	4A		3
DECX		$X \leftarrow (X)-1$						INH	5A		3
DEC opr,X		$M \leftarrow (M)-1$						IX1	6A	ff	6
DEC ,X		$M \leftarrow (M)-1$						IX	7A		5
EOR #opr	逻辑异或	$A \leftarrow (A) \square (M)$	—	—	◆	◆	—	IM	A8	ii	2
EOR opr								DIR	B8	dd	3
EOR opr								EXT	C8	hh ll	4
EOR opr,X								IX2	D8	ee ff	5
EOR opr,X								IX1	E8	ff	4
EOR ,X								IX	F8		3
INC opr	加一	$M \leftarrow (M)+1$	—	—	◆	◆	—	DIR	3C	dd	5
INCA		$A \leftarrow (A)+1$						INH	4C		3
INCX		$X \leftarrow (X)+1$						INH	5C		3
INC opr,X		$M \leftarrow (M)+1$						IX1	6C	ff	6
INC ,X		$M \leftarrow (M)+1$						IX	7C		5

表 5 指令集总表 (5/8)

指令	功能	描述	对 CCR 的影响					寻址方式	操作码	操作数	指令周期	
			H	I	N	Z	C					
JMP opr	跳转	PC ← Jump Address						DIR	BC	dd	2	
JMP opr								EXT	CC	hh ll	3	
JMP opr,X								IX2	DC	ee ff	4	
JMP opr,X								IX1	EC	ff	3	
JMP ,X								IX	FC		2	
JSR opr	调用子程序	PC ← (PC)+n (n=1,2,or 3)						DIR	BD	dd	5	
JSR opr		Push (PCL); SP ← (SP)-1						EXT	CD	hh ll	6	
JSR opr,X		Push (PCH); SP ← (SP)-1						IX2	DD	ee ff	7	
JSR opr,X		PC ← Effective Address						IX1	ED	ff	6	
JSR ,X								IX	FD		5	
LDA #opr	A 寄存器存数	A ← (M)						IM	A6	ii	2	
LDA opr								DIR	B6	dd	3	
LDA opr					◆	◆		EXT	C6	hh ll	4	
LDA opr,X								IX2	D6	ee ff	5	
LDA opr,X								IX1	E6	ff	4	
LDA ,X							IX	F6		3		
LDX #opr	X 寄存器存数	X ← (M)						IM	AE	ii	2	
LDX opr								DIR	BE	dd	3	
LDX opr					◆	◆		EXT	CE	hh ll	4	
LDX opr,X								IX2	DE	ee ff	5	
LDX opr,X								IX1	EE	ff	4	
LDX ,X							IX	FE		3		
LSL opr	逻辑左移 (同 ASL)							DIR	38	dd	5	
LSLA									INH	48		3
LSLX				◆	◆	◆			INH	58		3
LSL opr,X									IX1	68	ff	6
LSL ,X									IX	78		5
LSR opr	逻辑右移							DIR	34	dd	5	
LSRA									INH	44		3
LSRX				0	◆	◆			INH	54		3
LSR opr,X									IX1	64	ff	6
LSR ,X									IX	74		5
MUL	乘法 (暂不支持)	X:A ← (X) × (A)	0				0	INH	42		11	

**表 5 指令集总表 (6/8)**

指令	功能	描述	对 CCR 的影响					寻址方式	操作码	操作数	指令周期	
			H	I	N	Z	C					
NEG opr	取补	$M \leftarrow -(M) = \$00-(M)$						DIR	30	dd	5	
NEGA		$A \leftarrow -(A) = \$00-(M)$						INH	40		3	
NEGX		$X \leftarrow -(X) = \$00-(M)$	—	—	◆	◆	◆	INH	50		3	
NEG opr,X		$M \leftarrow -(M) = \$00-(M)$						IX1	60	ff	6	
NEG ,X		$M \leftarrow -(M) = \$00-(M)$						IX	70		5	
NOP	空操作		—	—	—	—	—	INH	9D		2	
ORA #opr	逻辑或	$A \leftarrow (A) \vee (M)$						IM	A	ii	2	
ORA opr								M	A			
ORA opr									DIR	BA	dd	3
ORA opr,X						◆	◆	—	EXT	CA	hh ll	4
ORA opr,X									IX2	D	ee ff	5
ORA ,X									IX1	EA		
ORA ,X							IX	FA	ff	4		
ROL opr	循环左移							DIR	39	dd	5	
ROLA									INH	49		3
ROLX						◆	◆	◆	INH	59		3
ROL opr,X									IX1	69	ff	6
ROL ,X									IX	79		5
ROR opr	循环右移							DIR	36	dd	5	
RORA									INH	46		3
RORX						◆	◆	◆	INH	56		3
ROR opr,X									IX1	66	ff	6
ROR ,X									IX	76		5
RSP	SP 复位	$SP \leftarrow \$00FF$	—	—	—	—	—	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		9	
RTS	子程序返回	$SP \leftarrow (SP)+1$ ; Pull (PCH) $SP \leftarrow (SP)+1$ ; Pull (PCL)	—	—	—	—	—	INH	81		6	

**表 5 指令集总表 (7/8)**

指令	功能	描述	对 CCR 的影响					寻址方式	操作码	操作数	指令周期
			H	I	N	Z	C				
SBC #opr	带借位的减法	$A \leftarrow (A) - (M) - (C)$						IM	A2	ii	2
SBC opr								M	B2	dd	3
SBC opr					◆	◆	◆	EXT	C2	hh ll	4
SBC opr,X								IX2	D2	ee ff	5
SBC opr,X								IX1	E2	ff	4
SBC ,X								IX	F2		3
SEC	C 标志置位	$C \leftarrow 1$	—	—	—	—	1	INH	99		2
SEI	I 标志置位	$I \leftarrow 1$	—	1	—	—	—	INH	9B		2
STA opr	A 寄存器取数	$M \leftarrow (A)$						DIR	B7	dd	4
STA opr								EXT	C7	hh ll	5
STA opr,X				◆	◆	—		IX2	D7	ee ff	6
STA opr,X								IX1	E7	ff	5
STA ,X								IX	F7		4
STOP	进 STOP 模式		—	0	—	—	—	INH	8E		2
STX opr	X 寄存器取数	$M \leftarrow (X)$						DIR	BF	dd	4
STX opr								EXT	CF	hh ll	5
STX opr,X				◆	◆	—		IX2	DF	ee ff	6
STX opr,X								IX1	EF	ff	5
STX ,X								IX	FF		4
SUB #opr	减法	$A \leftarrow (A) - (M)$						IM	A0	ii	2
SUB opr								M	B0	dd	3
SUB opr					◆	◆	◆	EXT	C0	hh ll	4
SUB opr,X								IX2	D0	ee ff	5
SUB opr,X								IX1	E0	ff	4
SUB ,X								IX	F0		3
SWI	软中断	$PC \leftarrow (PC) + 1$ Push (PCL); $SP \leftarrow (SP) - 1$ Push (PCH); $SP \leftarrow (SP) - 1$ Push (X); $SP \leftarrow (SP) - 1$ Push (A); $SP \leftarrow (SP) - 1$ Push (CCR); $SP \leftarrow (SP) - 1$ $I \leftarrow 1$ PCH ← 中断向量高位 PCL ← 中断向量低位	—	1	—	—	—	INH	83		10

**表 5 指令集总表 (8/8)**

指令	功能	描述	对 CCR 的影响					寻址方式	操作码	操作数	指令周期
			H	I	N	Z	C				
TAX	将 A 的值传到 X	$X \leftarrow (A)$	—	—	—	—	—	INH	97		2
TST opr	零测试	$(M) - \$00$						DIR	3D	dd	4
TSTA								INH	4D		3
TSTX			◆	◆	—			INH	5D		3
TST opr,X								IX1	6D	ff	5
TST ,X								IX	7D		4
TXA	将 X 的值传到 A	$A \leftarrow (X)$	—	—	—	—	—	INH	9F		2
WAIT	进 WAIT 模式		—	0	—	—	—	INH	8F		2

- |       |                  |     |                    |
|-------|------------------|-----|--------------------|
| A     | 累加器              | opr | 操作数 (一字节或两字节)      |
| C     | 进位位              | PC  | PC 指针              |
| CCR   | 状态寄存器            | PCH | PC 指针高位            |
| dd    | 直接寻址地址           | PCL | PC 指针低位            |
| dd rr | 直接寻址地址、条件跳转指令偏移量 | REL | 相对寻址               |
| DIR   | 直接寻址             | rel | PC 相对偏移量           |
| ee ff | 16 位偏移量地址        | rr  | PC 相对偏移量           |
| EXT   | 扩展寻址             | SP  | 堆栈指针               |
| ff    | 8 位偏移量地址         | X   | 变址寄存器              |
| H     | 半进位位             | Z   | 零标志位               |
| hh ll | 扩展寻址 16 位地址      | #   | 立即数                |
| I     | 中断屏蔽位            | ∧   | 逻辑与                |
| ii    | 立即数              | ∨   | 逻辑或                |
| IMM   | 立即寻址             | □   | 逻辑异或               |
| INH   | 隐含寻址             | ()  | 寄存器内容              |
| IX    | 无偏移量变址寻址         | -() | 取补 (取负数)           |
| IX1   | 8 位偏移量变址寻址       | ←   | 取值                 |
| IX2   | 16 位偏移量变址寻址      | ?   | 判断                 |
| M     | 存储器地址            | :   | 连接两个 8 位数据成 16 位数据 |
| N     | 负标志位             | ◆   | 清零或置位              |
| n     | 任意位 (n=0...7)    | —   | 不影响                |

### 3.6 HC05 指令集详述

下面将对 HC05 指令集的所有指令进行详细描述。这些指令按照助记符的字母顺序排列。

#### 3.6.1 ADC 带进位的加法

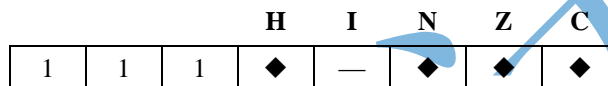
操作

$$A \leftarrow (A) + (M) + (C)$$

描述

将 A 寄存器和存储器 M 内的值以及 C 标志位相加，结果存在 A 寄存器中。

对 CCR 标志的影响



H  $A3 \cdot M3 + M3 \cdot \overline{R3} + \overline{R3} \cdot A3$  ①

若结果产生第 3 位向第 4 位的进位则置位，否则清零

N  $R7$

若结果最高位为 1 则置位，否则清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

若结果的所有位为 0 则置位，否则清零

C  $A7 \cdot M7 + M7 \cdot \overline{R7} + \overline{R7} \cdot A7$

若结果产生进位则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
ADC #opr	IMM	A9	ii	2
ADC opr	DIR	B9	dd	3
ADC opr	EXT	C9	hh ll	4
ADC opr,X	IX2	D9	ee ff	5
ADC opr,X	IX1	E9	ff	4
ADC ,X	IX	F9		3

注①：A3 表示 A 寄存器的第 3 位，M3 表示存储器的第 3 位，R3 表示运算结果的第 3 位

#### 3.6.2 ADD 加法

操作

$$A \leftarrow (A) + (M)$$

描述

将 A 寄存器和存储器 M 内的值相加，结果存在 A 寄存器中。

对 CCR 标志的影响

			H	I	N	Z	C
1	1	1	◆	—	◆	◆	◆

H  $A3 \cdot M3 + M3 \cdot \overline{R3} + \overline{R3} \cdot A3$

若结果产生第 3 位向第 4 位的进位则置位，否则清零

N  $R7$

若结果最高位为 1 则置位，否则清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

若结果的所有位为 0 则置位，否则清零

C  $A7 \cdot M7 + M7 \cdot \overline{R7} + \overline{R7} \cdot A7$

若结果产生进位则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
ADD #opr	IMM	AB	ii	2
ADD opr	DIR	BB	dd	3
ADD opr	EXT	CB	hh ll	4
ADD opr,X	IX2	DB	ee ff	5
ADD opr,X	IX1	EB	ff	4
ADD ,X	IX	FB		3

3.6.3 AND 逻辑与

操作

$A \leftarrow (A) \cdot (M)$

描述

将 A 寄存器和存储器 M 内的值做逻辑与，结果存在 A 寄存器中。

对 CCR 标志的影响

			H	I	N	Z	C
1	1	1	—	—	◆	◆	—

N  $R7$

若结果最高位为 1 则置位，否则清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

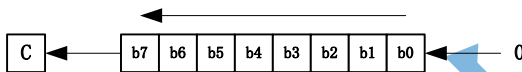
若结果的所有位为 0 则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
AND #opr	IMM	A4	ii	2
AND opr	DIR	44	dd	3
AND opr	EXT	C4	hh ll	4
AND opr,X	IX2	D4	ee ff	5
AND opr,X	IX1	E4	ff	4
AND ,X	IX	F4		3

### 3.6.4 ASL 算术左移 (同 LSL)

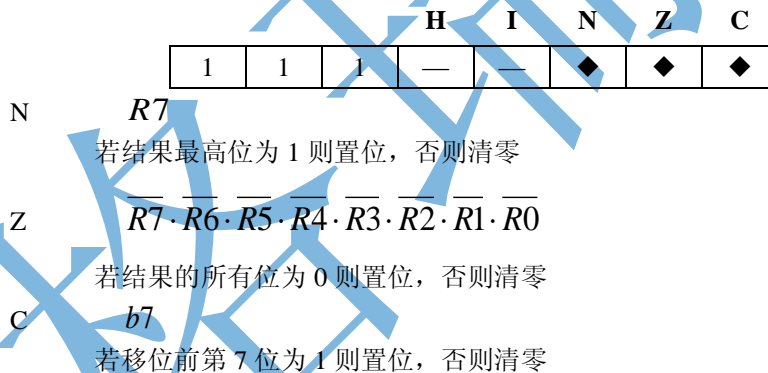
操作



描述

将 A 寄存器、X 寄存器或存储器 M 内的数左移一位。第 0 位总是移入 0，第 7 位移到 C 标志位中。

对 CCR 标志的影响

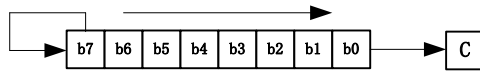


指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
ASL opr	DIR	38	dd	5
ASLA	INH	48		3
ASLX	INH	58		3
ASL opr,X	IX1	68	ff	6
ASL ,X	IX	78		5

### 3.6.5 ASR 算术右移

操作



描述

将 A 寄存器、X 寄存器或存储器 M 内的数右移一位。第 7 位保持原值，第 0 位移到 C 标志位中。这一操作等效于有符号整数除 2。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

N  $R7$   
若结果最高位为 1 则置位，否则清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$   
若结果的所有位为 0 则置位，否则清零

C  $b0$   
若移位前第 0 位为 1 则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
ASR opr	DIR	37	dd	5
ASRA	INH	47		3
ASRX	INH	57		3
ASR opr,X	IX1	67	ff	6
ASR ,X	IX	77		5

### 3.6.6 BCC 无进位则跳 (同 BHS)

操作

$$PC \leftarrow (PC) + 2 + \text{rel if } (C) = 0$$

描述

若 C 标志位为 0 则发生跳转；否则，执行下一条指令。

对 CCR 标志的影响

H	I	N	Z	C
1	1	1	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BCC rel	REL	24	rr	3

### 3.6.7 BCLR n 第 n 位清零

操作

$M_n \leftarrow 0$

描述

将存储器 M 的第 n 位 (n=7, 6, 5...0) 清零, 其他位保持不变。M 必须在地址空间 \$0000-\$00FF(即立即寻址空间)内。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

不影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BCLR 0,opr	DIR(b0)	11	dd	5
BCLR 1,opr	DIR(b1)	13	dd	5
BCLR 2,opr	DIR(b2)	15	dd	5
BCLR 3,opr	DIR(b3)	17	dd	5
BCLR 4,opr	DIR(b4)	19	dd	5
BCLR 5,opr	DIR(b5)	1B	dd	5
BCLR 6,opr	DIR(b6)	1D	dd	5
BCLR 7,opr	DIR(b7)	1F	dd	5

### 3.6.8 BCS 进位则跳转 (同 BLO)

操作

$PC \leftarrow (PC) + 2 + rel$  if (C)=1

描述

若 C 标志位为 1 则发生跳转; 否则, 执行下一条指令。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BCS rel	REL	25	rr	3

### 3.6.9 BEQ 等于则跳转

操作

$$PC \leftarrow (PC) + 2 + \text{rel if } (Z) = 1$$

描述

若 Z 标志位为 1 则发生跳转；否则，执行下一条指令。紧跟在 CMP、SUB 指令后执行 BEQ，则当(A)=(M)时发生跳转。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BEQ rel	REL	27	rr	3

### 3.6.10 BHCC 无半进位则跳转

操作

$$PC \leftarrow (PC) + 2 + \text{rel if } (H) = 0$$

描述

若 H 标志位为 0 则发生跳转；否则，执行下一条指令。这条指令用于判断是否要对运算结果进行 BCD 调整。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BHCC rel	REL	28	rr	3

### 3.6.11 BHCS 半进位则跳转

操作

$$PC \leftarrow (PC) + 2 + \text{rel if } (H) = 1$$

描述

若 H 标志位为 1 则发生跳转；否则，执行下一条指令。这条指令用于判断是否要对运算结果进行 BCD 调整。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BHCS rel	REL	29	rr	3

### 3.6.12 BHI 大于则跳转

操作

$PC \leftarrow (PC) + 2 + \text{rel}$  if  $[(C)+(Z)]=0$ , 等效于  $(A) > (M)$

描述

若 C 和 Z 同时为 0 则发生跳转；否则，执行下一条指令。紧跟在 CMP、SUB 指令后执行 BHI，则当  $(A) > (M)$  时发生跳转。

对 CCR 标志的影响

			H	I	N	Z	C
1	1	1	—	—	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BHI rel	REL	22	rr	3

### 3.6.13 BHS 大于等于则跳转（同 BCC）

操作

$PC \leftarrow (PC) + 2 + \text{rel}$  if  $(C)=0$ , 等效于  $(A) \geq (M)$

描述

若 C 标志位为 0 则发生跳转；否则，执行下一条指令。紧跟在 CMP、SUB 指令后执行 BHS，则当  $(A) \geq (M)$  时发生跳转。

对 CCR 标志的影响

			H	I	N	Z	C
1	1	1	—	—	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BHS rel	REL	24	rr	3

### 3.6.14 BIH IRQ 为高则跳转①

操作

$PC \leftarrow (PC) + 2 + \text{rel}$  if  $IRQ=1$

描述

若 IRQ 引脚为高则发生跳转；否则，执行下一条指令。

对 CCR 标志的影响

			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BIH rel	REL	2F	rr	3

注①：BIH 和 BIL 这在指令集中是存在的，但 IRQ 引脚在绝大多数产品中是没有的，所以这两条指令实际没有用。

### 3.6.15 BIL IRQ 为低则跳转①

操作

$PC \leftarrow (PC) + 2 + rel$  if IRQ=0

描述

若 IRQ 引脚为低则发生跳转；否则，执行下一条指令。

对 CCR 标志的影响

			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BIL rel	REL	2E	rr	3

注①：BIH 和 BIL 这在指令集中是存在的，但 IRQ 引脚在绝大多数产品中是没有的，所以这两条指令实际没有用。

### 3.6.16 BIT 位测试

操作

(A) (M)

描述

将 A 寄存器和存储器 M 内的值做逻辑与，但不影响 A 寄存器和 M 存储器的值（只影响 CCR 标志）。

对 CCR 标志的影响

			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	◆	◆	—

N R7

若结果最高位为 1 则置位，否则清零

$$Z \quad \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

若结果的所有位为 0 则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BIT #opr	IMM	A5	ii	2
BIT opr	DIR	B5	dd	3
BIT opr	EXT	C5	hh ll	4
BIT opr,X	IX2	D5	ee ff	5
BIT opr,X	IX1	E5	ff	4
BIT ,X	IX	F5		3

### 3.6.17 BLO 小于则跳转（同 BCS）

操作

$$PC \leftarrow (PC) + 2 + \text{rel if } (C) = 1, \text{ 等效于 } (A) < (M)$$

描述

若 C 标志位为 1 则发生跳转；否则，执行下一条指令。紧跟在 CMP、SUB 指令后执行 BLO，则当(A)<(M)时发生跳转。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BLO rel	REL	25	rr	3

### 3.6.18 BLS 小于等于则跳转

操作

$$PC \leftarrow (PC) + 2 + \text{rel if } [(C) + (Z)] = 1, \text{ 等效于 } (A) \leq (M)$$

描述

若 C 为 1 或 Z 为 1 则发生跳转；否则，执行下一条指令。紧跟在 CMP、SUB 指令后执行 BLS，则当(A)≤(M)时发生跳转。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BLS rel	REL	23	rr	3

### 3.6.19 BMC 中断未屏蔽则跳转

操作

$$PC \leftarrow (PC) + 2 + \text{rel if } (I) = 0$$

描述

若 I 标志位为 0（中断未屏蔽）则发生跳转；否则，执行下一条指令。

对 CCR 标志的影响

			H	I	N	Z	C
1	1	1	—	—	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BMC rel	REL	2C	rr	3

### 3.6.20 BMI 值为负则跳转

操作

$$PC \leftarrow (PC) + 2 + \text{rel if } (N) = 1$$

描述

若 N 标志位为 1 则发生跳转；否则，执行下一条指令。

对 CCR 标志的影响

			H	I	N	Z	C
1	1	1	—	—	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BMI rel	REL	2B	rr	3

### 3.6.21 BMS 中断屏蔽则跳转

操作

$$PC \leftarrow (PC) + 2 + \text{rel if } (I) = 1$$

描述

若 I 标志位为 1（中断屏蔽）则发生跳转；否则，执行下一条指令。

对 CCR 标志的影响

			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BMS rel	REL	2D	rr	3

### 3.6.22 BNE 不等于则跳转

操作

$$PC \leftarrow (PC) + 2 + \text{rel if } (Z) = 0$$

描述

若 Z 标志位为 0 则发生跳转；否则，执行下一条指令。紧跟在 CMP、SUB 指令后执行 BEQ，则当(A)≠(M)时发生跳转。

对 CCR 标志的影响

			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BNE rel	REL	26	rr	3

### 3.6.23 BPL 值为正则跳转

操作

$$PC \leftarrow (PC) + 2 + \text{rel if } (N) = 0$$

描述

若 N 标志位为 0 则发生跳转；否则，执行下一条指令。

对 CCR 标志的影响

			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BPL rel	REL	2A	rr	3

### 3.6.24 BRA 无条件跳转

操作

$$PC \leftarrow (PC) + 2 + rel$$

描述

无条件发生跳转。

对 CCR 标志的影响

	<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BRA rel	REL	20	rr	3

### 3.6.25 BRCLR n 第 n 位为 0 则跳转

操作

$$PC \leftarrow (PC) + 3 + rel \text{ if } M_n = 0$$

描述

若存储器 M 的第 n 位 (n=7, 6, 5...0) 为 0, 则发生跳转; 否则, 执行下一条指令。M 必须在地址空间 \$0000-\$00FF (即立即寻址空间) 内。C 标志位会根据判断的结果发生变化。与循环移位指令一起使用, 可以实现串并转换的功能。

对 CCR 标志的影响

	<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	◆

C Mn=1 则置位, 否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BRCLR 0,opr,rel	DIR(b0)	01	dd rr	5
BRCLR 1,opr,rel	DIR(b1)	03	dd rr	5
BRCLR 2,opr,rel	DIR(b2)	05	dd rr	5
BRCLR 3,opr,rel	DIR(b3)	07	dd rr	5
BRCLR 4,opr,rel	DIR(b4)	09	dd rr	5
BRCLR 5,opr,rel	DIR(b5)	0B	dd rr	5
BRCLR 6,opr,rel	DIR(b6)	0D	dd rr	5
BRCLR 7,opr,rel	DIR(b7)	0F	dd rr	5

### 3.6.26 BRN 永不跳转

操作

$$PC \leftarrow (PC) + 2$$

描述

永不发生跳转（执行下一条指令）。这条指令相当于执行 3 个周期的空操作。

对 CCR 标志的影响

H	I	N	Z	C
1	1	1	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BRA rel	REL	21	rr	3

### 3.6.27 BRSET n 第 n 位为 1 则跳转

操作

$$PC \leftarrow (PC) + 3 + \text{rel if } Mn=1$$

描述

若存储器 M 的第 n 位（n=7, 6, 5, ..., 0）为 1，则发生跳转；否则，执行下一条指令。M 必须在地址空间 \$0000-\$00FF（即立即寻址空间）内。C 标志位会根据判断的结果发生变化。与循环移位指令一起使用，可以实现串并转换的功能。

对 CCR 标志的影响

H	I	N	Z	C
1	1	1	—	◆

C Mn=1 则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BRSET 0,opr,rel	DIR(b0)	00	dd rr	5
BRSET 1,opr,rel	DIR(b1)	02	dd rr	5
BRSET 2,opr,rel	DIR(b2)	04	dd rr	5
BRSET 3,opr,rel	DIR(b3)	06	dd rr	5
BRSET 4,opr,rel	DIR(b4)	08	dd rr	5
BRSET 5,opr,rel	DIR(b5)	0A	dd rr	5
BRSET 6,opr,rel	DIR(b6)	0C	dd rr	5
BRSET 7,opr,rel	DIR(b7)	0E	dd rr	5

### 3.6.28 BSET n 第 n 位置位

操作

$M_n \leftarrow 1$

描述

将存储器 M 的第 n 位 (n=7, 6, 5...0) 置位, 其他位保持不变。M 必须在地址空间 \$0000-\$00FF(即立即寻址空间)内。

对 CCR 标志的影响

	<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BSET 0,opr	DIR(b0)	10	dd	5
BSET 1,opr	DIR(b1)	12	dd	5
BSET 2,opr	DIR(b2)	14	dd	5
BSET 3,opr	DIR(b3)	16	dd	5
BSET 4,opr	DIR(b4)	18	dd	5
BSET 5,opr	DIR(b5)	1A	dd	5
BSET 6,opr	DIR(b6)	1C	dd	5
BSET 7,opr	DIR(b7)	1E	dd	5

### 3.6.29 BSR 跳转到子程序

操作

$PC \leftarrow (PC)+2$   
 Push (PCL);  $SP \leftarrow (SP)-1$ ;  
 Push (PCH);  $SP \leftarrow (SP)-1$   
 $PC \leftarrow (PC)+rel$

描述

PC 指针首先加 2, 即指向下一条指令处, 然后将这个 PC 值压栈, 并跳转到子程序的地址 (PC 加偏移量 rel)。BSR 要与 RTS (子程序返回) 指令配合使用。

对 CCR 标志的影响

	<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
BSR rel	REL	AD	rr	6

### 3.6.30 CLC C 标志清零

操作

$C \leftarrow 0$

描述

对 C 标志位清零。

对 CCR 标志的影响

	<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	0

C 0  
清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
CLC	INH	98		2

### 3.6.31 CLI I 标志清零

操作

$I \leftarrow 0$

描述

对 I 标志位清零。I 标志清零后，系统才能响应中断。

对 CCR 标志的影响

	<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	0	—

I 0  
清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
CLI	INH	98		2

### 3.6.32 CLR 清零

操作

$A \leftarrow \$00$  或  $X \leftarrow \$00$  或  $M \leftarrow \$00$

描述

对 A 寄存器或 X 寄存器或存储器 M 的值清零。

对 CCR 标志的影响

	<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	0	1

N 0  
清零

Z 1  
置位

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
CLR opr	DIR	3F	dd	5
CLRA	INH	4F		3
CLR X	INH	5F		3
CLR opr,X	IX1	6F	ff	6
CLR ,X	IX	7F		5

### 3.6.33 CMP A 寄存器比较

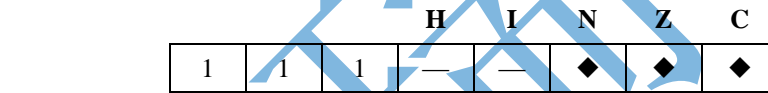
操作

(A)-(M)

描述

比较 A 寄存器和存储器 M 的值(实际就是将两者作减法,并根据计算结果更新 CCR 标志,由后续指令判断比较的结果)。A 寄存器和存储器 M 的值都不会改变。

对 CCR 标志的影响



N R7

若结果最高位为 1 则置位, 否则清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

若结果的所有位为 0 则置位, 否则清零

C  $A7 \cdot M7 + M7 \cdot \overline{R7} + \overline{R7} \cdot A7$

若结果产生进位则置位, 否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
CMP #opr	IMM	A1	dd	2
CMP opr	DIR	B1	dd	3
CMP opr	EXT	C1	hh ll	4
CMP opr,X	IX2	D1	ee ff	5
CMP opr,X	IX1	E1	ff	4
CMP ,X	IX	F1		3

### 3.6.34 COM 取反

操作

$A \leftarrow \$FF-(A)$  或  $X \leftarrow \$FF-(X)$  或  $M \leftarrow \$FF-(M)$

描述

对 A 寄存器或 X 寄存器或存储器 M 的值逻辑取反。

对 CCR 标志的影响

	H	I	N	Z	C			
	1	1	1	—	—	◆	◆	1

N  $R7$

若结果最高位为 1 则置位，否则清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

若结果的所有位为 0 则置位，否则清零

C 1  
置位

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
COM opr	DIR	33	dd	5
COMA	INH	43		3
COMX	INH	53		3
COM opr,X	IX1	63	ff	6
COM ,X	IX	73		5

### 3.6.35 CPX X 寄存器比较

操作

$(X)-(M)$

描述

比较 X 寄存器和存储器 M 的值(实际就是将两者作减法,并根据计算结果更新 CCR 标志,由后续指令判断比较的结果)。X 寄存器和存储器 M 的值都不会改变。

对 CCR 标志的影响

	H	I	N	Z	C			
	1	1	1	—	—	◆	◆	◆

N  $R7$

若结果最高位为 1 则置位，否则清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

若结果的所有位为 0 则置位，否则清零

$$C \quad X7 \cdot M7 + M7 \cdot \overline{R7} + \overline{R7} \cdot X7$$

若结果产生进位则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
CPX #opr	IMM	A3	ii	2
CPX opr	DIR	B3	dd	3
CPX opr	EXT	C3	hh ll	4
CPX opr,X	IX2	D3	ee ff	5
CPX opr,X	IX1	E3	ff	4
CPX ,X	IX	F3		3

### 3.6.36 DEC 减一

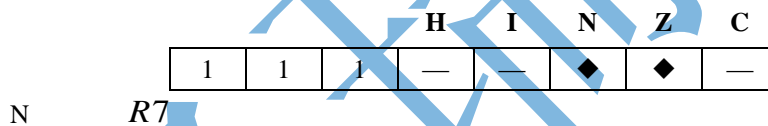
操作

$$A \leftarrow (A) - \$01 \quad \text{或} \quad X \leftarrow (X) - \$01 \quad \text{或} \quad M \leftarrow (M) - \$01$$

描述

对 A 寄存器或 X 寄存器或存储器 M 的值减一。CCR 的 N、Z 标志会根据减一的结果发生变化，但 C 标志不受影响。

对 CCR 标志的影响



N  $R7$   
若结果最高位为 1 则置位，否则清零

Z  $\overline{R7 \cdot R6 \cdot R5 \cdot R4 \cdot R3 \cdot R2 \cdot R1 \cdot R0}$   
若结果的所有位为 0 则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
DEC opr	DIR	3A	dd	5
DECA	INH	4A		3
DECX	INH	5A		3
DEC opr,X	IX1	6A	ff	6
DEC ,X	IX	7A		5

### 3.6.37 EOR 逻辑异或

操作

$$A \leftarrow (A) \oplus (M)$$

描述

将 A 寄存器和存储器 M 内的值做逻辑异或，结果存在 A 寄存器中。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

N  $R7$

若结果最高位为 1 则置位，否则清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

若结果的所有位为 0 则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
EOR #opr	IMM	A8	ii	2
EOR opr	DIR	B8	dd	3
EOR opr	EXT	C8	hh ll	4
EOR opr,X	IX2	D8	ee ff	5
EOR opr,X	IX1	E8	ff	4
EOR ,X	IX	F8		3

### 3.6.38 INC 加一

操作

$A \leftarrow (A) + \$01$  或  $X \leftarrow (X) + \$01$  或  $M \leftarrow (M) + \$01$

描述

对 A 寄存器或 X 寄存器或存储器 M 的值加一。CCR 的 N、Z 标志会根据加一的结果发生变化，但 C 标志不受影响。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

N  $R7$

若结果最高位为 1 则置位，否则清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

若结果的所有位为 0 则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
INC opr	DIR	3C	dd	5
INCA	INH	4C		3
INCX	INH	5C		3

INC opr,X	IX1	6C	ff	6
INC ,X	IX	7C		5

### 3.6.39 JMP 跳转

操作

PC ← Effective Address

描述

跳转到有效地址（直接寻址、扩展寻址、变址寻址）。

对 CCR 标志的影响

	<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>			
	1	1	1	—	—	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
JMP opr	DIR	BC	dd	2
JMP opr	EXT	CC	hh ll	3
JMP opr,X	IX2	DC	ee ff	4
JMP opr,X	IX1	EC	ff	3
JMP ,X	IX	FC		2

### 3.6.40 JSR 调用子程序

操作

PC ← (PC) + n (n=1,2,3,由寻址方式决定)

Push (PCL); SP ← (SP) - 1;

Push (PCH); SP ← (SP) - 1

PC ← Effective Address

描述

PC 指针首先加 n(n=1,2,3,由寻址方式决定),即指向下一条指令处,然后将这个 PC 值压栈,并跳转到子程序的有效地址(直接寻址、扩展寻址、变址寻址)。JSR 要与 RTS(子程序返回)指令配合使用。

对 CCR 标志的影响

	<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>			
	1	1	1	—	—	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
JSR opr	DIR	BD	dd	5
JSR opr	EXT	CD	hh ll	6

JSR opr,X	IX2	DD	ee ff	7
JSR opr,X	IX1	ED	ff	6
JSR ,X	IX	FD		5

### 3.6.41 LDA A 寄存器存数

操作

$A \leftarrow (M)$

描述

存储器 M 内的数存到 A 寄存器中。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

N  $R7$

若结果最高位为 1 则置位，否则清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

若结果的所有位为 0 则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
LDA #opr	IMM	A6	ii	2
LDA opr	DIR	B6	dd	3
LDA opr	EXT	C6	hh ll	4
LDA opr,X	IX2	D6	ee ff	5
LDA opr,X	IX1	E6	ff	4
LDA ,X	IX	F6		3

### 3.6.42 LDX X 寄存器存数

操作

$X \leftarrow (M)$

描述

存储器 M 内的数存到 X 寄存器中。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

N  $R7$

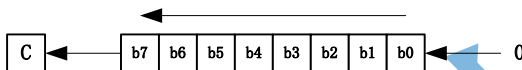
若结果最高位为 1 则置位，否则清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

若结果的所有位为 0 则置位，否则清零

**指令格式、寻址方式、机器码、指令周期**

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
LDX #opr	IMM	AE	ii	2
LDX opr	DIR	BE	dd	3
LDX opr	EXT	CE	hh ll	4
LDX opr,X	IX2	DE	ee ff	5
LDX opr,X	IX1	EE	ff	4
LDX ,X	IX	FE		3

**3.6.43 LSL 逻辑左移 (同 ASL)**
**操作**

**描述**

将 A 寄存器、X 寄存器或存储器 M 内的数左移一位。第 0 位总是移入 0，第 7 位移到 C 标志位中。

**对 CCR 标志的影响**


**N**  $R7$   
若结果最高位为 1 则置位，否则清零

**Z**  $R7 \cdot R6 \cdot R5 \cdot R4 \cdot R3 \cdot R2 \cdot R1 \cdot R0$   
若结果的所有位为 0 则置位，否则清零

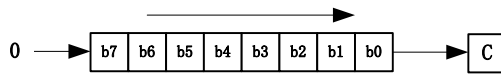
**C**  $b7$   
若移位前第 7 位为 1 则置位，否则清零

**指令格式、寻址方式、机器码、指令周期**

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
LSL opr	DIR	38	dd	5
LSLA	INH	48		3
LSLX	INH	58		3
LSL opr,X	IX1	68	ff	6
LSL ,X	IX	78		5

### 3.6.44 LSR 逻辑右移

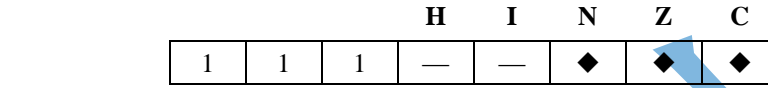
操作



描述

将 A 寄存器、X 寄存器或存储器 M 内的数右移一位。第 7 位总是移入 0，第 0 位移到 C 标志位中。

对 CCR 标志的影响



N 0  
清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$   
若结果的所有位为 0 则置位，否则清零

C b0  
若移位前第 0 位为 1 则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
LSR opr	DIR	34	dd	5
LSRA	INH	44		3
LSRX	INH	54		3
LSR opr,X	IX1	64	ff	6
LSR ,X	IX	74		5

### 3.6.45 MUL 乘法（暂不支持）

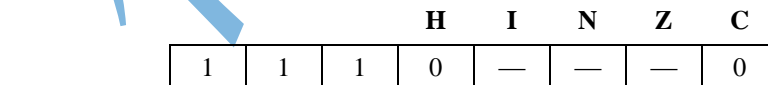
操作

$$X:A \leftarrow (X) \times (A)$$

描述

将 X 寄存器和 A 寄存器内的 2 个 8 位数据相乘（无符号数乘法），结果的低 8 位存到 A 寄存器中，高 8 位存到 X 寄存器中。

对 CCR 标志的影响



H 0  
清零

C 0  
清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
MUL	INH	42		11

注：请慎用，在 3V 以下工作电压时可能会存在该指令失效现象。

### 3.6.46 NEG 取补

**操作**

$A \leftarrow \neg(A)$  或  $X \leftarrow \neg(X)$  或  $M \leftarrow \neg(M)$

**描述**

对 A 寄存器或 X 寄存器或存储器 M 的值（有符号数）逻辑取补。

**对 CCR 标志的影响**

	<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
	1	1	1	—	—
	—	—	◆	◆	◆

**N**  $R7$

若结果最高位为 1 则置位，否则清零

**Z**  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

若结果的所有位为 0 则置位，否则清零

**C**  $R7 + R6 + R5 + R4 + R3 + R2 + R1 + R0$

用 \$00 减的结果有借位则置位，否则清零（等效条件：若原来的数不为 \$00 则置位，否则复位）

**指令格式、寻址方式、机器码、指令周期**

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
NEG opr	DIR	30	dd	5
NEGA	INH	40		3
NEGX	INH	50		3
NEG opr,X	IX1	60	ff	6
NEG ,X	IX	70		5

### 3.6.47 NOP 空操作

**操作**

无任何操作

**描述**

仅对 PC 加 1，并等待 2 周期，其他无任何影响。

**对 CCR 标志的影响**

	<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
	1	1	1	—	—
	—	—	—	—	—

无影响

**指令格式、寻址方式、机器码、指令周期**

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
NOP	INH	9D		2

### 3.6.48 ORA 逻辑或

**操作**

$A \leftarrow (A) + (M)$

**描述**

将 A 寄存器和存储器 M 内的值做逻辑或，结果存在 A 寄存器中。

**对 CCR 标志的影响**

	H	I	N	Z	C
	1	1	1	—	—

N  $R7$   
若结果最高位为 1 则置位, 否则清零

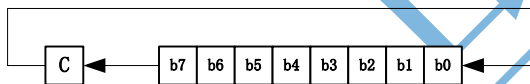
Z  $R7 \cdot R6 \cdot R5 \cdot R4 \cdot R3 \cdot R2 \cdot R1 \cdot R0$   
若结果的所有位为 0 则置位, 否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
ORA #opr	IMM	AA	ii	2
ORA opr	DIR	BA	dd	3
ORA opr	EXT	CA	hh ll	4
ORA opr,X	IX2	DA	ee ff	5
ORA opr,X	IX1	EA	ff	4
ORA ,X	IX	FA		3

### 3.6.49 ROL 循环左移

操作



描述

将 A 寄存器、X 寄存器或存储器 M 内的数左移一位。原来 C 标志位的值移到第 0 位中, 第 7 位移到 C 中。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

N  $R7$   
若结果最高位为 1 则置位, 否则清零

Z  $\overline{R7 \cdot R6 \cdot R5 \cdot R4 \cdot R3 \cdot R2 \cdot R1 \cdot R0}$   
若结果的所有位为 0 则置位, 否则清零

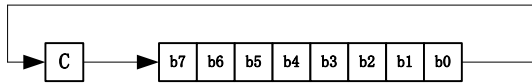
C  $b0$   
若移位前第 7 位为 1 则置位, 否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
ROL opr	DIR	39	dd	5
ROLA	INH	49		3
ROLX	INH	59		3
ROL opr,X	IX1	69	ff	6
ROL ,X	IX	79		5

### 3.6.50 RSR 循环右移

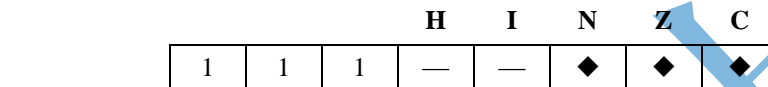
操作



描述

将 A 寄存器、X 寄存器或存储器 M 内的数右移一位。原来 C 标志位的值移到第 7 位中，第 0 位移到 C 中。

对 CCR 标志的影响



N 0  
清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$   
若结果的所有位为 0 则置位，否则清零

C  $b_0$   
若移位前第 0 位为 1 则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
ROR opr	DIR	36	dd	5
RORA	INH	46		3
RORX	INH	56		3
ROR opr,X	IX1	66	ff	6
ROR ,X	IX	76		5

### 3.6.51 RSP SP 复位

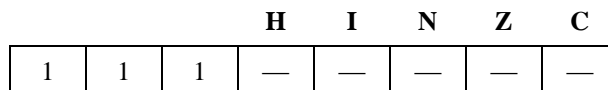
操作

SP ← \$00FF

描述

将堆栈指针复位成\$00FF。

对 CCR 标志的影响



无影响。

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
RSP	INH	9C		2

### 3.6.52 RTI 中断返回

操作

$SP \leftarrow (SP) + 1; \text{ Pull (CCR)}$   
 $SP \leftarrow (SP) + 1; \text{ Pull (A)}$   
 $SP \leftarrow (SP) + 1; \text{ Pull (X)}$   
 $SP \leftarrow (SP) + 1; \text{ Pull (PCH)}$   
 $SP \leftarrow (SP) + 1; \text{ Pull (PCL)}$

描述

CCR、A、X、PC 指针依次弹栈，回复中断前的状态。

对 CCR 标志的影响

			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	◆	◆	◆	◆	◆

CCR 恢复到中断前的状态

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
RTI	INH	80		9

### 3.6.53 RTS 子程序返回

操作

$SP \leftarrow (SP) + 1; \text{ Pull (PCH)}$   
 $SP \leftarrow (SP) + 1; \text{ Pull (PCL)}$

描述

PC 指针弹栈，程序从调用子程序的指令 JSR 或 BSR 的下一条开始执行。

对 CCR 标志的影响

			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	—	—	—

无影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
RTS	INH	81		6

### 3.6.54 SBC 带借位的减法

操作

$A \leftarrow (A) - (M) - (C)$

描述

将 A 寄存器的值减去存储器 M 内的值以及 C 标志位，结果存在 A 寄存器中。

对 CCR 标志的影响

			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	◆	◆	◆

N R7

若结果最高位为 1 则置位，否则清零

$$Z \quad \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

若结果的所有位为 0 则置位，否则清零

$$C \quad \overline{A7} \cdot M7 + M7 \cdot R7 + R7 \cdot \overline{A7}$$

若结果产生借位则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
SBC #opr	IMM	A2	ii	2
SBC opr	DIR	B2	dd	3
SBC opr	EXT	C2	hh ll	4
SBC opr,X	IX2	D2	ee ff	5
SBC opr,X	IX1	E2	ff	4
SBC ,X	IX	F2		3

### 3.6.55 SEC C 标志置位

操作

$$C \leftarrow 1$$

描述

对 C 标志位置位。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	1

C  
1  
置位

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
SEC	INH	99		2

### 3.6.56 SEI I 标志置位

操作

$$I \leftarrow 1$$

描述

对 I 标志位置位。I 标志置位后，系统不能响应中断。

对 CCR 标志的影响

	H	I	N	Z	C
	1	1	1	—	—

I  
1  
置位

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
SEI	INH	9B		2

### 3.6.57 STA A 寄存器取数

操作

$M \leftarrow (A)$

描述

取出 A 寄存器内的数存到存储器 M 中。

对 CCR 标志的影响

	<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
	1	1	1	—	—

N  $R7$

若结果最高位为 1 则置位，否则清零

Z  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

若结果的所有位为 0 则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
STA opr	DIR	B7	dd	4
STA opr	EXT	C7	hh ll	5
STA opr,X	IX2	D7	ee ff	6
STA opr,X	IX1	E7	ff	5
STA ,X	IX	F7		4
STA opr	DIR	B7	dd	4

### 3.6.58 STOP 进 STOP 模式

操作

进入 STOP 工作模式

描述

关闭主时钟，进入 STOP 工作模式，系统进入待机状态，可通过外中断等唤醒。I 标志自动清零，允许响应中断。

对 CCR 标志的影响

	<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
	1	1	1	—	0

I 0

清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
STOP	INH	8E		2

### 3.6.59 STX X 寄存器取数

操作

$$M \leftarrow (X)$$

描述

取出 X 寄存器内的数存到存储器 M 中。

对 CCR 标志的影响

	H	I	N	Z	C
1	1	1	—	—	◆

N  $R7$

若结果最高位为 1 则置位，否则清零

$$Z \quad \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

若结果的所有位为 0 则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
STX opr	DIR	BF	dd	4
STX opr	EXT	CF	hh ll	5
STX opr,X	IX2	DF	ee ff	6
STX opr,X	IX1	EF	ff	5
STX ,X	IX	FF		4
STX opr	DIR	BF	dd	4

### 3.6.60 SUB 减法

操作

$$A \leftarrow (A) - (M)$$

描述

将 A 寄存器的值减去存储器 M 内的值，结果存在 A 寄存器中。

对 CCR 标志的影响

	H	I	N	Z	C
1	1	1	—	—	◆

N  $R7$

若结果最高位为 1 则置位，否则清零

$$Z \quad \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

若结果的所有位为 0 则置位，否则清零

$$C \quad \overline{A7} \cdot M7 + M7 \cdot R7 + R7 \cdot \overline{A7}$$

若结果产生借位则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
SUB #opr	IMM	A0	ii	2
SUB opr	DIR	B0	dd	3

SUB opr	EXT	C0	hh ll	4
SUB opr,X	IX2	D0	ee ff	5
SUB opr,X	IX1	E0	ff	4
SUB ,X	IX	F0		3

### 3.6.61 SWI 软中断

操作

$PC \leftarrow (PC)+1$   
 Push (PCL);  $SP \leftarrow (SP)-1$   
 Push (PCH);  $SP \leftarrow (SP)-1$   
 Push (X);  $SP \leftarrow (SP)-1$   
 Push (A);  $SP \leftarrow (SP)-1$   
 Push (CCR);  $SP \leftarrow (SP)-1$   
 $I \leftarrow 1$   
 $PCH \leftarrow$  中断向量高位  
 $PCL \leftarrow$  中断向量低位

描述

软中断是由指令产生的，执行的过程与普通中断相同，即将 PC 指针、X 寄存器、A 寄存器、CCR 状态标志依次压栈，将 I 标志置位（禁止中断嵌套），然后从中断向量区读取中断服务程序入口地址。软中断不受中断屏蔽位 I 的影响。

对 CCR 标志的影响

			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	1	—	—	—

I 1 置位

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
SWI	INH	83		10

### 3.6.62 TAX 将 A 的值传到 X

操作

$X \leftarrow (A)$

描述

将 A 寄存器的值传到 X 寄存器中，A 寄存器的值保持不变。

对 CCR 标志的影响

			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	—	—	—

不影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
TAX	INH	97		10

### 3.6.63 TST 零测试

操作

(A)-\$00 或(X)-\$00 或(M)-\$00

描述

将 A 寄存器或 X 寄存器或存储器 M 的值与\$00 比较。

对 CCR 标志的影响

			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	◆	◆	—

N M7

最高位为 1 则置位，否则清零

Z  $\overline{M7} \cdot \overline{M6} \cdot \overline{M5} \cdot \overline{M4} \cdot \overline{M3} \cdot \overline{M2} \cdot \overline{M1} \cdot \overline{M0}$

若为 0 则置位，否则清零

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
TST opr	DIR	3D	dd	4
TSTA	INH	4D		3
TSTX	INH	5D		3
TST opr,X	IX1	6D	ff	5
TST ,X	IX	7D		4

### 3.6.64 TXA 将 X 的值传到 A

操作

$A \leftarrow (X)$

描述

将 X 寄存器的值传到 A 寄存器中，X 寄存器的值保持不变。

对 CCR 标志的影响

			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
1	1	1	—	—	—	—	—

不影响

指令格式、寻址方式、机器码、指令周期

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
TXA	INH	9F		10

**3.6.65 WAIT 进 WAIT 模式**
**操作**

进入 WAIT 工作模式

**描述**

关闭 CPU 时钟，进入 STOP 工作模式，可通过任意中断唤醒。I 标志自动清零，允许响应中断。

**对 CCR 标志的影响**

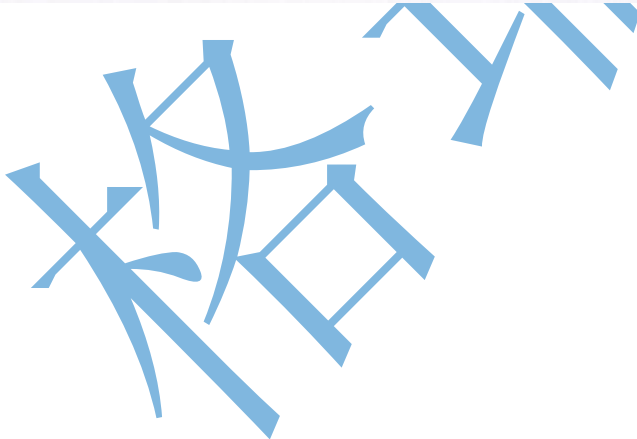
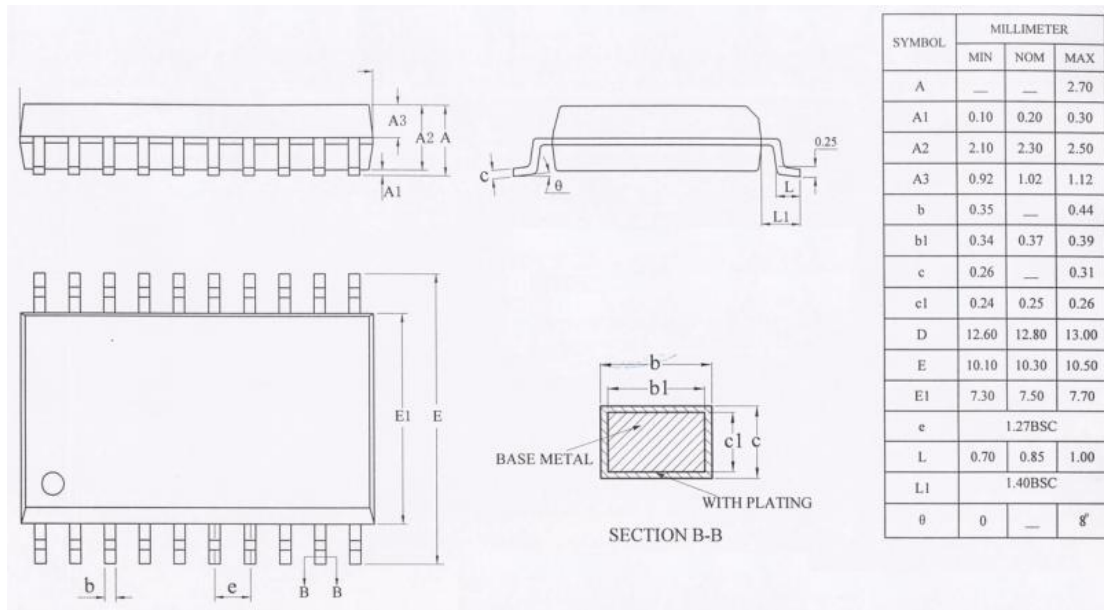
			<b>H</b>	<b>I</b>	<b>N</b>	<b>Z</b>	<b>C</b>
I	0	清零	1	1	1	—	0
			—	—	—	—	—

**指令格式、寻址方式、机器码、指令周期**

指令格式	寻址方式	机器码		指令周期
		操作码	操作数	
WAIT	INH	8F		2

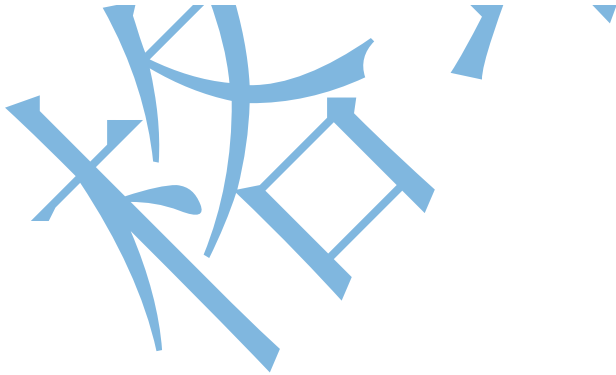
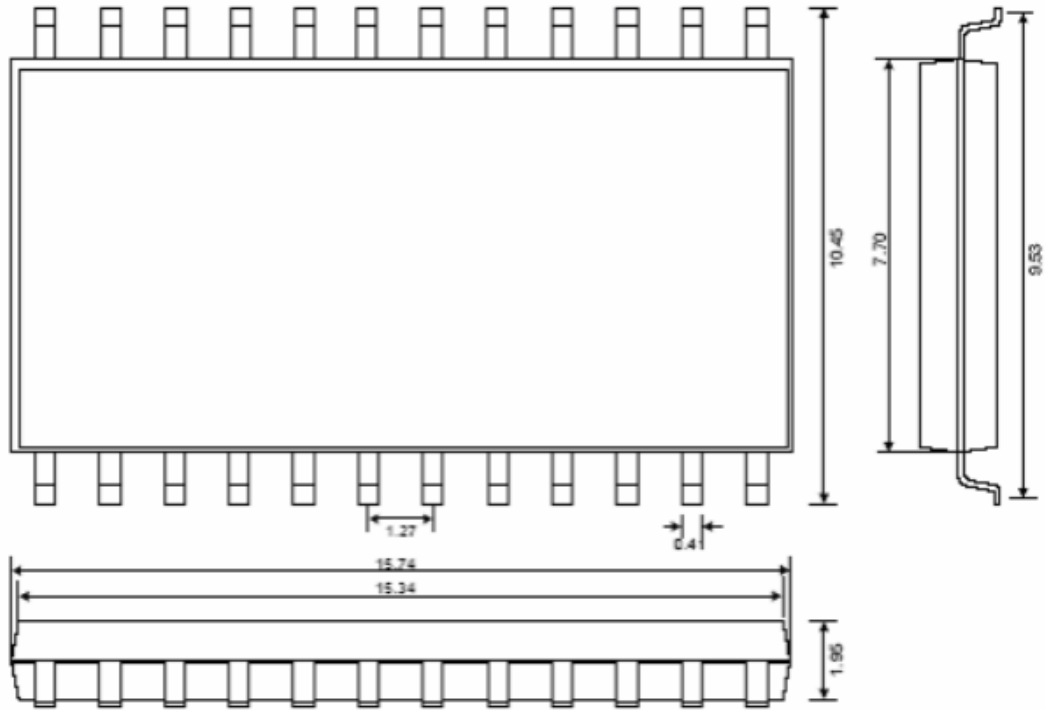
## 第 4 章 封装尺寸

### 4.1 SOP20



## 4.2 SOP24

SOP-24  
unit:mm



## 第 5 章 订购

下单规格	功能简述	芯片型号	封装
20E24BS24	8-Bit 通用 AD 型单片机	20E24B	SOP24
20E24BS20	8-Bit 通用 AD 型单片机	20E24B	SOP20

格瑞达

**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](http://www.greenmcu.com)