



8-Bit 通用 I/O 型单片机

20P801

版本：V02 日期：2012/04/21

www.greenmcu.com

目录

第 1 章 简述	1
1.1 产品简介	1
1.1.1 产品特性.....	1
1.1.2 系统框图.....	2
1.1.3 引脚排列.....	3
1.1.4 引脚说明.....	3
1.1.5 引脚结构示意图.....	4
第 2 章 系统	6
2.1 中央处理器	6
2.1.1 指令集.....	6
2.1.2 地址空间分配.....	6
2.1.3 程序存储器 ROM.....	6
2.1.4 用户数据存储存储器 RAM.....	6
2.1.5 用户配置字 OPBIT.....	6
2.1.6 控制寄存器.....	7
2.2 系统时钟	8
2.2.1 内置高精度 RC 振荡.....	8
2.2.2 WDT 振荡器.....	8
2.3 复位	9
2.3.1 概述.....	9
2.3.2 上电复位.....	10
2.3.3 外部复位.....	10
2.3.4 LVR 复位.....	10
2.3.5 WDT 复位.....	10
2.4 I/O 口	12
2.4.1 IO 工作模式.....	12
2.4.2 上拉电阻.....	12
2.4.3 下拉电阻.....	12
2.4.4 开漏输出.....	13
2.5 定时器	14
2.5.1 定时器 T0.....	14
2.5.2 定时/计数器 T1.....	16
2.5.3 WDT 定时器.....	20
2.6 中断	21
2.6.1 概述.....	21
2.6.2 外中断.....	21

2.6.3 键盘中断	22
2.6.4 定时器中断	23
2.6.5 WDT 中断	23
2.7 系统工作模式	24
2.7.1 STOP 模式	24
2.7.2 WAIT 模式	24
2.8 电气参数	25
2.8.1 极限参数	25
2.8.2 直流电气参数	25
2.8.3 交流电气参数	26
2.9 封装外形尺寸	27
2.10 附录	29
第 3 章 H05 指令集	31
3.1 简介	31
3.2 CPU (中央处理器)	31
3.2.1 ALU (算数逻辑单元)	31
3.2.2 CPU CONTROL (CPU 控制单元)	31
3.2.2.1 CPU 寄存器	32
3.2.2.2 累加器	32
3.2.2.3 变址寄存器	32
3.2.2.4 状态寄存器	32
3.2.2.5 PC 指针	32
3.2.2.6 堆栈指针	32
3.3 寻址方式	34
3.3.1 隐含寻址	34
3.3.2 立即寻址	35
3.3.3 扩展寻址	35
3.3.4 直接寻址	36
3.3.5 变址寻址	37
3.3.5.1 无偏移量变址寻址	37
3.3.5.2 8 位偏移量变址寻址	38
3.3.5.3 16 位偏移量变址寻址	39
3.3.6 相对寻址	40
表 1 寄存器/存储器指令	41
表 2 读/写/修改指令	42
表 3 条件跳转指令	43
表 4 控制指令	43
3.4 HC05 指令集汇总	44
表 5 指令集总表 (1/8)	45

表 5 指令集总表 (2/8)	46
表 5 指令集总表 (3/8)	47
表 5 指令集总表 (4/8)	48
表 5 指令集总表 (5/8)	49
表 5 指令集总表 (6/8)	50
表 5 指令集总表 (7/8)	51
表 5 指令集总表 (8/8)	53
3.5 HC05 指令集详述.....	54
3.5.1 ADC 带进位的加法.....	54
3.5.2 ADD 加法	54
3.5.3 AND 逻辑与	55
3.5.4 ASL 算术左移 (同 LSL)	56
3.5.5 ASR 算术右移	56
3.5.6 BCC 无进位则跳 (同 BHS).....	57
3.5.7 BCLR _N 第 N 位清零	57
3.5.8 BCS 进位则跳转 (同 BLO)	57
3.5.9 BEQ 等于则跳转	58
3.5.10 BHCC 无半进位则跳转	58
3.5.11 BHCS 半进位则跳转	58
3.5.12 BHI 大于则跳转	59
3.5.13 BHS 大于等于则跳转 (同 BCC)	59
3.5.14 BIH IRQ 为高则跳转①	59
3.5.15 BIL IRQ 为低则跳转①	60
3.5.16 BIT 位测试.....	60
3.5.17 BLO 小于则跳转 (同 BCS)	61
3.5.18 BLS 小于等于则跳转	61
3.5.19 BMC 中断未屏蔽则跳转	61
3.5.20 BMI 值为负则跳转.....	62
3.5.21 BMS 中断屏蔽则跳转	62
3.5.22 BNE 不等于则跳转	62
3.5.23 BPL 值为正则跳转.....	63
3.5.24 BRA 无条件跳转	63
3.5.25 BRCLR _N 第 N 位为 0 则跳转	63
3.5.26 BRN 永不跳转	64
3.5.27 BRSET _N 第 N 位为 1 则跳转	64
3.5.28 BSET _N 第 N 位置位	64
3.5.29 BSR 跳转到子程序.....	65
3.5.30 CLCC 标志清零	65
3.5.31 CLIF 标志清零	65
3.5.32 CLR 清零.....	66
3.5.33 CMPA 寄存器比较	66
3.5.34 COM 取反.....	67
3.5.35 CPX _X 寄存器比较.....	67
3.5.36 DEC 减一	68

3.5.37 EOR 逻辑异或.....	68
3.5.38 INC 加一.....	69
3.5.39 JMP 跳转.....	69
3.5.40 JSR 调用子程序.....	70
3.5.41 LDA A 寄存器存数.....	70
3.5.42 LDX X 寄存器存数.....	71
3.5.43 LSL 逻辑左移 (同 ASL).....	71
3.5.44 LSR 逻辑右移.....	72
3.5.45 MUL 乘法 (暂不支持).....	72
3.5.46 NEG 取补.....	73
3.5.47 NOP 空操作.....	73
3.5.48 ORA 逻辑或.....	74
3.5.49 ROL 循环左移.....	74
3.5.50 RSR 循环右移.....	75
3.5.51 RSP SP 复位.....	75
3.5.52 RTI 中断返回.....	75
3.5.53 RTS 子程序返回.....	76
3.5.54 SBC 带借位的减法.....	76
3.5.55 SECC 标志置位.....	77
3.5.56 SEII 标志置位.....	77
3.5.57 STA A 寄存器取数.....	77
3.5.58 STOP 进 STOP 模.....	78
3.5.59 STXX 寄存器取数.....	78
3.5.60 SUB 减法.....	79
3.5.61 SWI 软中断.....	79
3.5.62 TAX 将 A 的值传到 X.....	80
3.5.63 TST 零测试.....	80
3.5.64 TXA 将 X 的值传到 A.....	81
3.5.65 WAIT 进 WAIT 模式.....	81
第 4 章 订购信息.....	82

第 1 章 简述

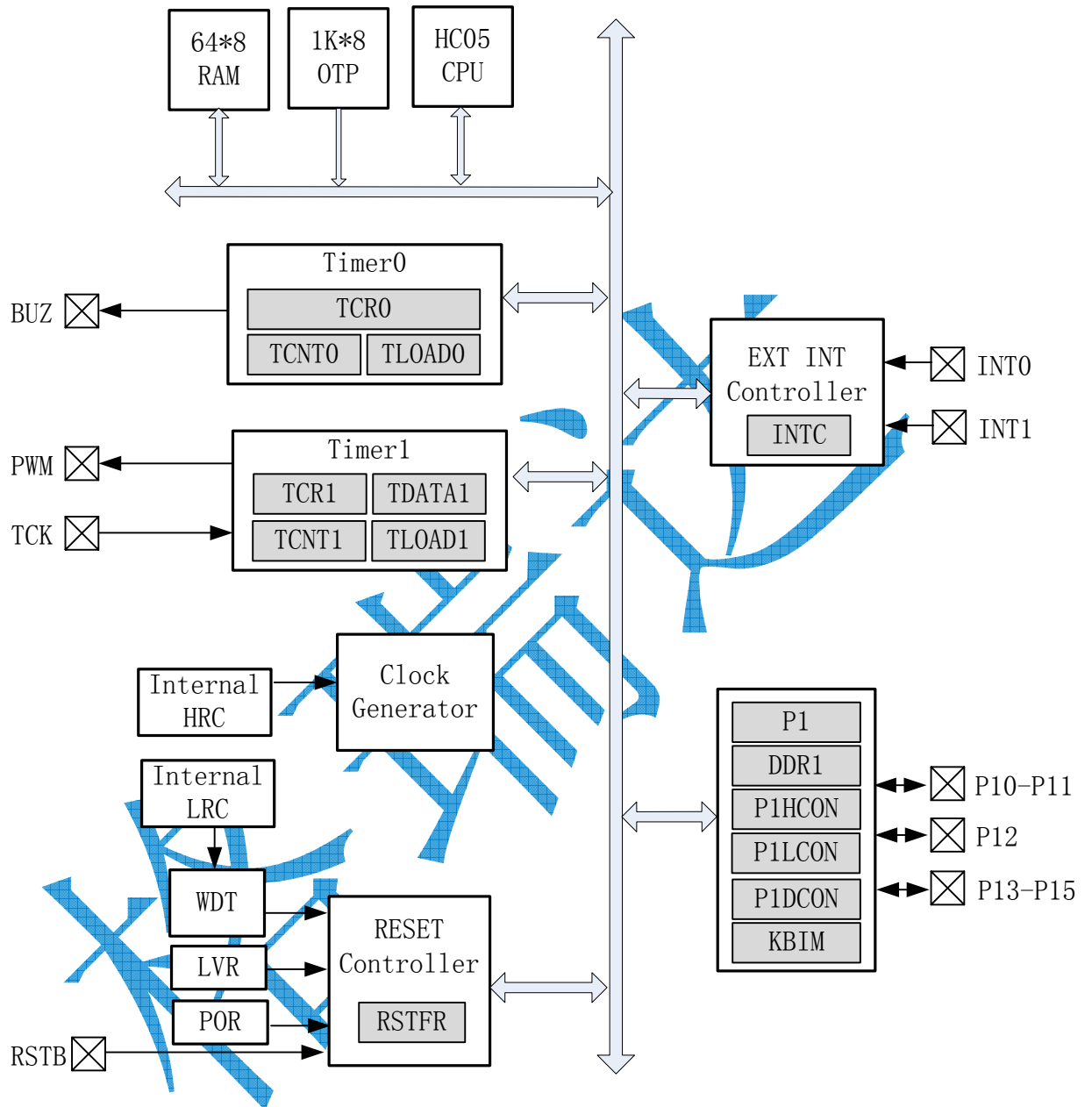
1.1 产品简介

20P801 是一款高性能 8 位 OTP 型 MCU，内置高精度 RC 振荡器。产品的高抗干扰性能能为小家电产品提供良好的解决方案。

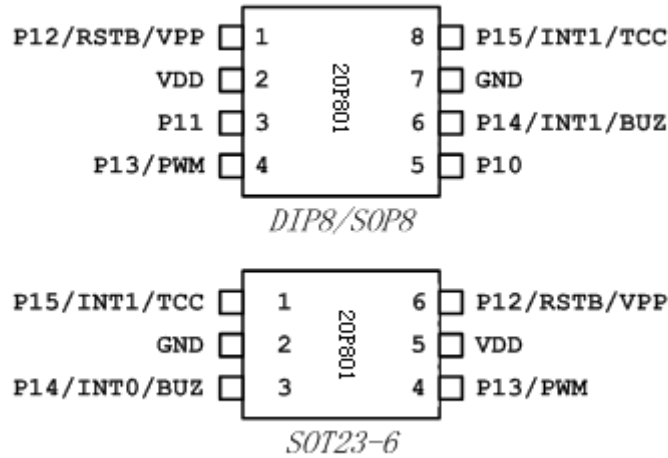
1.1.1 产品特性

- ◇ 8 位 CISC 结构 CPU (Motorola HC05 兼容)
- ◇ 1K*8 位 OTP ROM
- ◇ 64 字节 RAM
- ◇ 支持 6 个通用 I/O 口
- ◇ 1 个基本 8 位自装载定时器，可设置溢出中断，并可输出 BUZ
- ◇ 1 个基本 8 位自装载定时器/计数器，其信号源可由软件设定，可设置溢出中断，并可输出 PWM 信号
- ◇ 2 路外中断 (INT0/INT1)，可设置上升沿/下降沿/高电平/低电平触发方式
- ◇ 内部自振式看门狗计数器 (WDT)
- ◇ 低压复位 LVR (2.1V/3.6V 可选)
- ◇ 6 个中断源：INT0、INT1、TMI0、TMI1、KBI、WDT1
- ◇ 振荡模式
 - 内置 RC 振荡：1MHz、2MHz、4MHz、8MHz 可选 (偏差 $\lt\pm 1\%$ ，25°C，5V 工作电压)
- ◇ 低功耗设计 ($\lt 3\text{mA}@4\text{MHz}(5\text{V})$ ， $\lt 1\mu\text{A}@\text{STOP}$ 模式)
- ◇ 串行烧写接口电路
- ◇ 程序加密功能
- ◇ 工作电压
 - 2.0-5.5V@ (振荡频率 2MHz-4MHz)
 - 2.7-5.5V@ (振荡频率 2MHz-8MHz)
- ◇ 封装形式：DIP8、SOP8、SOT23-6

1.1.2 系统框图



1.1.3 引脚排列

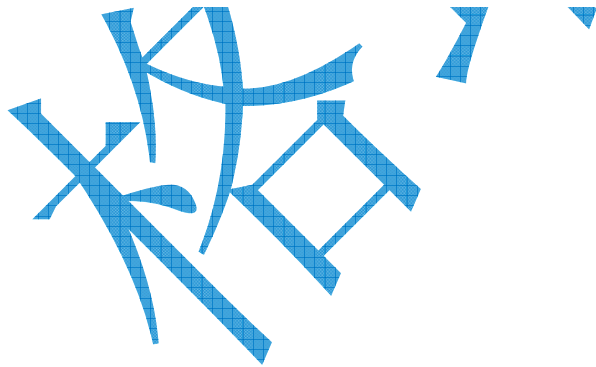
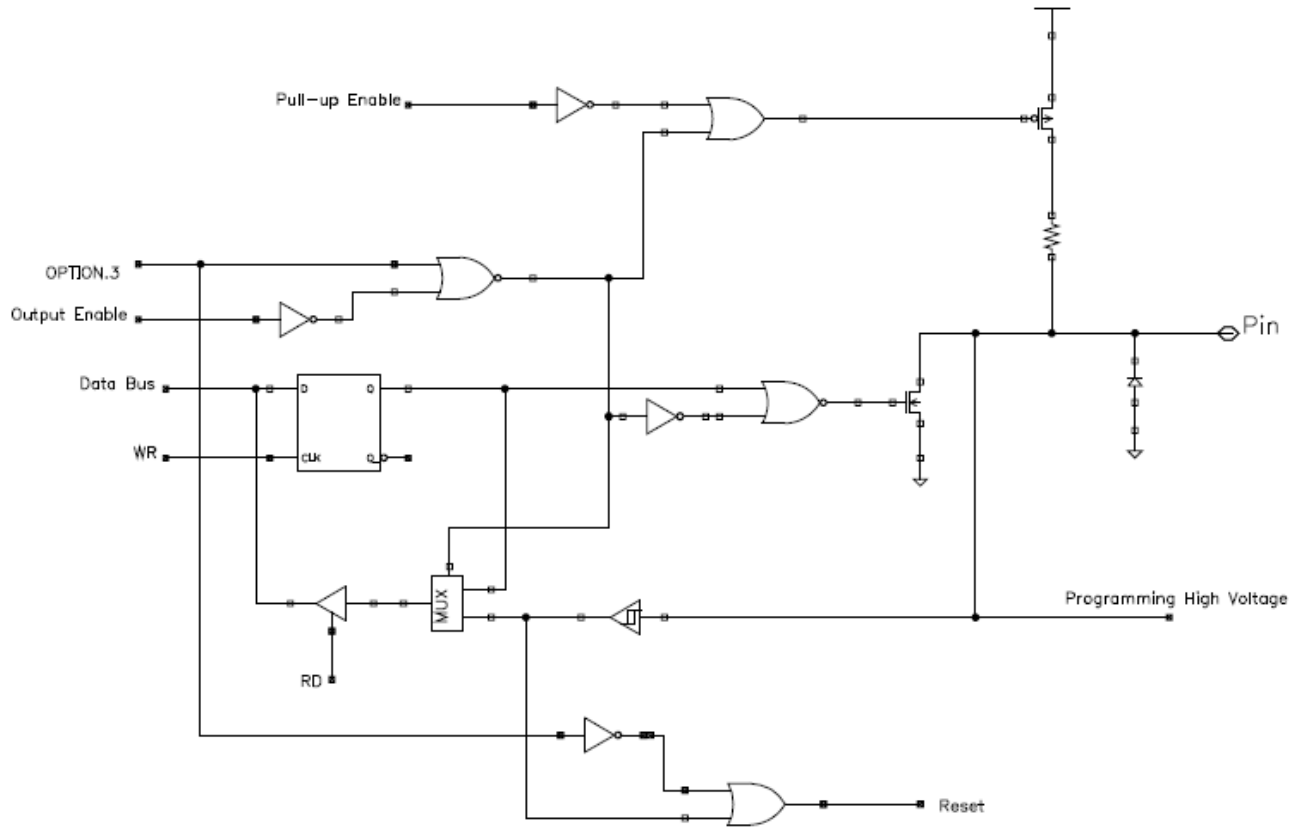


1.1.4 引脚说明

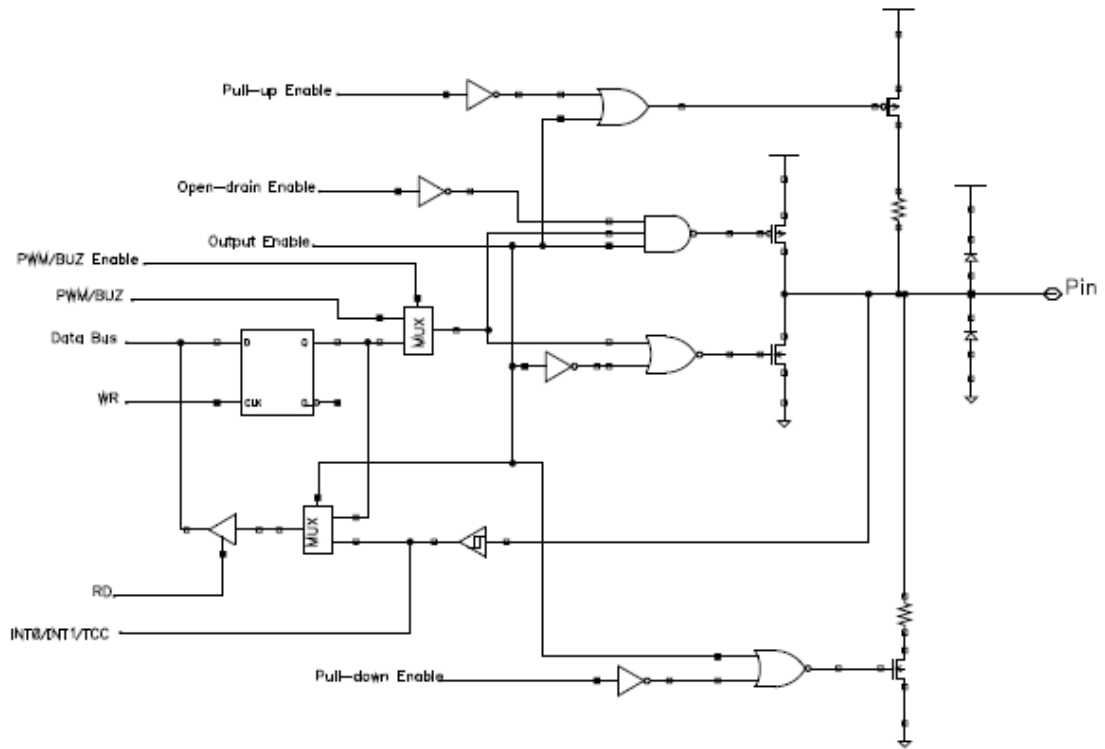
编号		引脚名	方向	类型	功能描述
8pin	6pin				
1	6	P12	I	PUO	输入、开漏输出口，上拉电阻可选，可触发键盘中断
		RSTB			外部复位输入
		VPP			编程高压输入
2	5	VDD	P		电源
3	-	P11	I/O	PUDO	输入输出口，上、下拉电阻可选，开漏输出可选，可触发键盘中断
4	4	P13	I/O	PUDO	输入输出口，上、下拉电阻可选，开漏输出可选，可触发键盘中断
		PWM			PWM 输出
5	-	P10	I/O	PUDO	输入输出口，上、下拉电阻可选，开漏输出可选，可触发键盘中断
6	3	P14	I/O	PUDO	输入输出口，上、下拉电阻可选，开漏输出可选，可触发键盘中断
		INT0			外中断 1 输入
		BUZ			BUZ 输出
7	2	GND	P		公共地
8	1	P15	I/O	PUDO	输入输出口，上、下拉电阻可选，开漏输出可选，可触发键盘中断
		INT1			外中断 0 输入
		TCC			TCC 输入

1.1.5 引脚结构示意图

PUO



PUDO



格瑞达

第 2 章 系统

2.1 中央处理器

2.1.1 指令集

20P801 采用 HC05 指令集。指令集详细资料见本公司手册《HC05 指令集》。

2.1.2 地址空间分配

\$0000-\$0011: 控制寄存器
 \$0012-\$00BF: 未定义
 \$00C0-\$00FF: RAM (含堆栈)
 \$0100-\$1BFF: 未定义
 \$1C00-\$1FFF: OTP ROM

2.1.3 程序存储器 ROM

20P801 的程序存储器是 1K 字节 (8 bits) 的 OTP ROM, 可用于存放用户程序。在程序存储区的最后, 即地址 \$1FE0~\$1FFF 这 32 个字节是复位和中断向量区 (见 §0)。

2.1.4 用户数据存储器 RAM

20P801 的用户数据存储器有 64 字节 (8 bits), 与堆栈复用。有关堆栈的说明见《HC05 指令集》。

2.1.5 用户配置字 OPBIT

用户配置字简称 OPBIT 是 OTP 中的一个特殊字节, 用于对系统功能进行配置。OPBIT 在烧写用户程序时通过专用烧写器来设置。20P801 的 OPBIT 定义如下。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPBIT	ENCR	-	LVRE	LVRS	RSTE	WDTE	OSCS1	OSCS0

BIT[7] ENCR - 程序区加密位
 0: 加密
 1: 不加密

BIT[5] LVRE - LVR 使能
 0: LVR 开启
 1: LVR 关闭

BIT[4] LVRS - LVR 电压选择
 0: LVR 电压 2.1V
 1: LVR 电压 3.6V

BIT[3] RSTE - P12/RSTB 功能选择
 0: P12/RSTB 引脚作为输入/开漏输出口
 1: P12/RSTB 引脚作为外部复位口

BIT[2] WDTE - WDT 振荡器使能
 0: WDT 振荡器关闭
 1: WDT 振荡器开启

BIT[1:0] OSCS[1:0] - 系统主时钟振荡方式选择

00: 内置 RC 振荡器 1MHz

01: 内置 RC 振荡器 2MHz

10: 内置 RC 振荡器 4MHz

11: 内置 RC 振荡器 8MHz

2.1.6 控制寄存器

20P801 的全部控制寄存器列在下表中，具体功能详见各功能模块的说明。

地址	助记符	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	初始值
\$03	P1	R/W	-	-	P15	P14	P13	P12	P11	P10	--00 0000
\$04	DDR1	R/W	-	-	DDR15	DDR14	DDR13	DDR12	DDR11	DDR10	--00 0000
\$05	P1HCON	R/W	-	-	PH15	PH14	PH13	PH12	PH11	PH10	--00 0000
\$06	P1LCON	R/W	-	-	PL15	PL14	PL13	-	PL11	PL10	--00 0-00
\$07	P1DCON	R/W	-	-	PD15	PD14	PD13	-	PD11	PD10	--00 0-00
\$08	KBIM	R/W	-	-	KBIM5	KBIM4	KBIM3	KBIM2	KBIM1	KBIM0	--00 0000
\$09	TCNT0	R	T0C7	T0C6	T0C5	T0C4	T0C3	T0C2	T0C1	T0C0	1111 1111
\$09	TLOAD0	W	T0L7	T0L6	T0L5	T0L4	T0L3	T0L2	T0L1	T0L0	1111 1111
\$0A	TCR0	R/W	T0IF	T0IM	T0PR2	T0PR1	T0PR0	T0PTA	T0PTS	BUZOE	0101 1000
\$0B	TDATA1	W	T1D7	T1D6	T1D5	T1D4	T1D3	T1D2	T1D1	T1D0	0000 0000
\$0C	TCNT1	R	T1C7	T1C6	T1C5	T1C4	T1C3	T1C2	T1C1	T1C0	0000 0000
\$0C	TLOAD1	W	T1L7	T1L6	T1L5	T1L4	T1L3	T1L2	T1L1	T1L0	0000 0000
\$0D	TCR1	R/W	T1IF	T1IM	T1PR2	T1PR1	T1PR0	T1EN	T1PTS	PWMOE	0100 0000
\$0E	INTC	R/W	INT0E	INT0M1	INT0M0	INT0F	INT1E	INT1M1	INT1M0	INT1F	0000 0000
\$0F	MCR	R/W	KBIE	KBIC	-	WDTC	WDTF	WDTM	-	-	000- 000-
\$10	RSTFR	R/W	-	-	-	-	WDTRF	LVRRF	RSTRF	PORRF	---- 0000

2.2 系统时钟

由内置高精度 RC 电路产生的时钟信号 F_{osc} 经 2 分频后产生系统主时钟 F_{sys} 。
另有一个低速 RC 振荡器专供 WDT（看门狗）电路使用。

2.2.1 内置高精度 RC 振荡

20P801 的内置高精度 RC 振荡器有 1MHz、2MHz、4MHz、8MHz 四种频率可选。

特别提示：为确保振荡的精度和稳定性，在实际应用时需要在芯片的 VDD 和 GND 之间加 10uF 以上的电解电容，且电容和芯片的距离尽可能靠近（建议控制在 5cm 以内）。

2.2.2 WDT 振荡器

20P801 内置一个低频的 RC 振荡器（频率典型值 10KHz），该振荡器仅供给 WDT 电路使用，而不能作为系统主时钟用。WDT 振荡器是否开启由 OPBIT 的 WDTE 配置。

2.3 复位

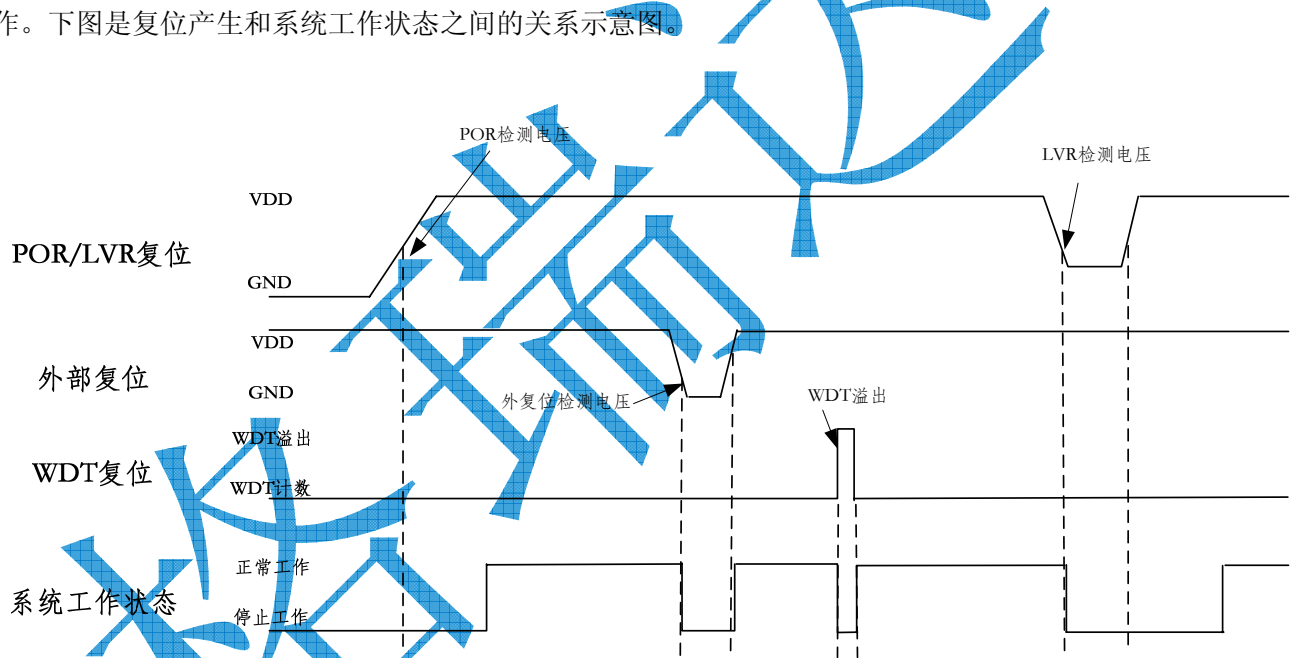
2.3.1 概述

20P801 有四种可能的复位方式：

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

任何一种复位发生时，系统将会从\$1FFE:\$1FFF 中取出复位向量地址，并从该地址处开始执行指令；另外系统还会将所有的控制寄存器重置为默认初始值。

上电复位和 LVR 复位会关闭系统主时钟的振荡器，复位解除后才重新打开振荡器，由于振荡器起振和稳定需要一定的时间，所以系统会在 256 个时钟周期后开始重新工作。外部复位和 WDT 复位不会关闭系统主时钟振荡器，所以复位解除后 2 个时钟周期后即开始工作。下图是复位产生和系统工作状态之间的关系示意图。



寄存器 RSTFR 会记录复位方式，当某类复位发生时，RSTFR 中相应标志位被系统置 1，要清除该标志，则必须对标志位写 0。

\$10	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSTFR	-	-	-	-	WDTRF	LVRRF	RSTRF	PORRF
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3] WDTRF - WDT 复位标志

0: 无 WDT 复位

1: 有 WDT 复位

写 0 清标志，写 1 无效。

BIT[2] LVRRF - LVR 复位标志

- 0: 无 LVR 复位
1: 有 LVR 复位
写 0 清标志, 写 1 无效。
- BIT[1] RSTRF** - 外部复位标志
0: 无外部复位
1: 有外部复位
写 0 清标志, 写 1 无效。
- BIT[0] PORRF** - 上电复位标志
0: 无上电复位
1: 有上电复位
写 0 清标志, 写 1 无效。

2.3.2 上电复位

20P801 的上电复位电路可以适应快速、慢速上电的情况, 并且当芯片上电过程中出现电源电压抖动时都能保证系统可靠的复位。

上电复位过程可以概括为以下几个步骤:

- (1) 检测系统工作电压, 等待电压高于 V_{POR} 并保持稳定;
- (2) 如果外部复位功能开启, 则需等待复位引脚电压高于 V_{IH} ;
- (3) 复位 PC 指针、初始化所有寄存器;
- (4) 开启主时钟振荡器, 并等待 256 个时钟周期;
- (5) 上电结束, 系统开始执行指令。

2.3.3 外部复位

外部复位功能是否开启可以通过 OPBIT 的 RSTE 配置, 同时 RESE 可以选择引脚的内部上拉电阻是否有效 (见 §0)。外部复位引脚是施密特结构的, 低电平有效。当外复位引脚为高电平时, 系统正常运行; 为低电平时, 系统产生复位。

2.3.4 LVR 复位

20P801 的 LVR 电压有两级 (2.1V 和 3.6V^①), 通过 OPBIT 的 LVRS 配置, LVR 功能的使用由 LVRE 配置 (见 §0)。LVR 电压检测电路有一定的回滞特性, 通常回滞电压为 0.05V^② 左右, 也就是说, 如果选择了 3.6V 的 LVR 电压, 则当电源电压下降到 3.6V 时 LVR 复位有效, 而电压需要上升到 3.65V 时 LVR 复位才会解除。

注①: LVR 电压 2.1V 和 3.6V 仅作为设计参考, 存在较明显的误差, 不能用作精确电压检测。

注②: 此处的 0.05V 为理论值。

2.3.5 WDT 复位

WDT 看门狗复位是一种对程序正常运行的保护机制。正常情况下, 用户软件会按时对 WDT 定时器进行清零操作, 定时器不会溢出。若出现异常状况, 程序未按预想执行, 出现程序跑飞的情况, 那么 WDT 定时器会出现溢出从而触发 WDT 复位, 系统重新初始化, 返回受控状态。

20P801 的 WDT 看门狗电路有独立的内置 RC 振荡器，不受系统主时钟的影响，即使主时钟振荡器出现异常停振，WDT 复位仍会产生。考虑到分频系数的不同，WDT 溢出复位时间范围约为 205ms~26.2s（典型值）。

有关 WDT 看门狗定时器，可参考 [§0](#)。

格瑞达

2.4 I/O 口

2.4.1 IO 工作模式

20P801 有 5 个通用双向 I/O 口 (P15-P13、P11-P10) 和一个输入/开漏输出口 (P12)。每一个双向 I/O 口都有相应的数据寄存器 (P1) 和方向寄存器 (DDR1) 控制, 功能如下表所示。

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

\$04	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DDR1	-	-	DDR15	DDR14	DDR13	DDR12	DDR11	DDR10
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] DDR1n - P1 口方向寄存器 (n=5-0)

0: 作为输入口

1: 作为输出口

2.4.2 上拉电阻

P1 口可通过 P1HCON 选择是否接上拉电阻 (约 25K Ω)。上拉电阻在端口置为输入状态时有效, 置为输出状态时无效。P1 口 (除 P12) 可以将上拉电阻和下拉电阻同时置为有效。

\$05	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1HCON	-	-	PH15	PH14	PH13	PH12	PH11	PH10
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] PH1n - P1 口上拉电阻选择 (n=5-0)

0: 上拉电阻无效

1: 上拉电阻有效

2.4.3 下拉电阻

P1 口 (除 P12 外) 可通过 P1LCON 选择是否接下拉电阻 (约 25K Ω)。下拉电阻在端口置为输入状态时有效, 置为输出状态时无效。P1 口 (除 P12) 可以将下拉电阻和上拉电阻同时置为有效。

\$06	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1LCON	-	-	PL15	PL14	PL13	-	PL11	PL10
R/W	-	-	R/W	R/W	R/W	-	R/W	R/W
初始值	-	-	0	0	0	-	0	0

BIT[5:3] **PL1n** - P1 口下拉电阻选择 (n=5-3)

0: 下拉电阻无效

1: 下拉电阻有效

BIT[1:0] **PL1n** - P1 口下拉电阻选择 (n=1-0)

0: 下拉电阻无效

1: 下拉电阻有效

2.4.4 开漏输出

P1 口 (除 P12 外) 可通过 P1DCON 选择是否为开漏输出。开漏输出在端口置为输出状态时有效, 置为输入状态时无效。P12 口的输出总是开漏的。

注: 开漏输出在外加上拉电阻应用时, 上拉电压不能超过 VDD 的电压。

\$07	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1DCON	-	-	PD15	PD14	PD13	-	PD11	PD10
R/W	-	-	R/W	R/W	R/W	-	R/W	R/W
初始值	-	-	0	0	0	-	0	0

BIT[5:3] **PD1n** - P1 口开漏输出选择 (n=5-3)

0: 开漏输出无效

1: 开漏输出有效

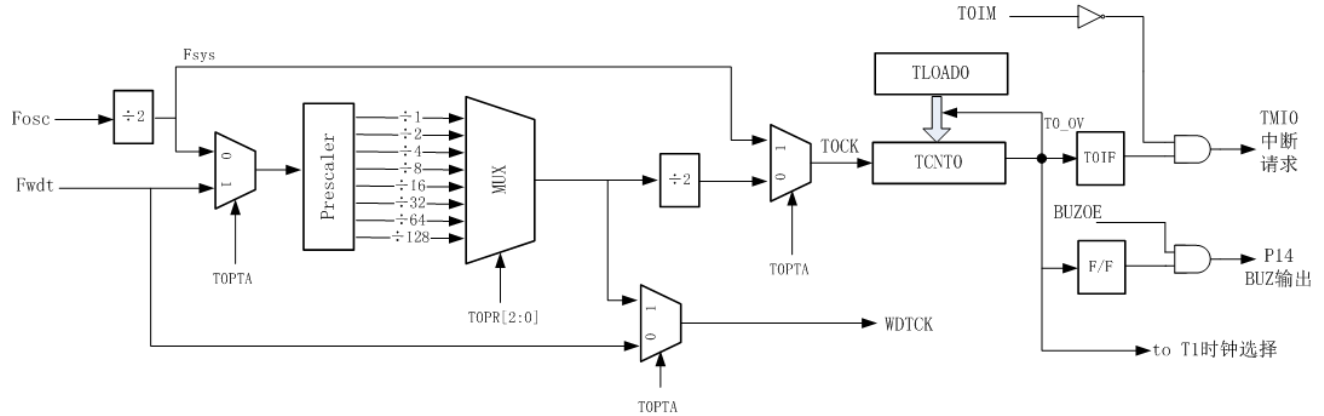
BIT[1:0] **PD1n** - P1 口开漏输出选择 (n=1-0)

0: 开漏输出无效

1: 开漏输出有效

2.5 定时器

2.5.1 定时器 T0



20P801 的定时器 T0 由 8 位计数器 TCNT0、8 位自加载寄存器 TLOAD0、可编程预分频器和控制寄存器 TCRO 组成。

TCNT0 的时钟 TOCK 来自系统主时钟 Fsys (Fosc/2) 或其分频信号。定时器 T0 的预分频器与 WDT 定时器共用，当 TOPTA=0 时，预分频器分配给 T0 使用；TOPTA=1 时，预分频器分配给 WDT 使用。分频系数由 TOPR[2:0] 决定。TOCK 和 WDTCK 的频率与 TOPTA、TOPR[2:0] 的关系如下：

TOPTA	TOPR[2:0]	TOCK (Fsys 的倍数)	WDTCK (Fwdt 的倍数)
0	n	1	2 ⁿ
1	n	2 ⁿ⁺¹	1

TCNT0 是一个递减计数器，它的值可以读出（不可写），当计数到零时，产生溢出信号 TO_OV，这个信号有 4 个功能：

- (1) 作为 TLOAD0 的自动加载信号，即溢出发生时，将 TLOAD0 的值载入 TCNT0；
- (2) 将 TOIF 标志位置 1，如果此时中断屏蔽位 TOIM=0 则产生 TMIO 中断请求；
- (3) 经一个触发器 (F/F) 产生 BUZ 信号，并在 BUZOE=1 时经 P14 输出；
- (4) 作为定时器 T1 的时钟信号 (TOPTS=1 时有效)。见 § 0。

TLOAD0 是一个只写寄存器，它与 TCNT0 共用同一个地址 \$09。如果预分频器分配给 T0 (即 TOPTA=0)，那么写 TLOAD0 会对预分频器清零。

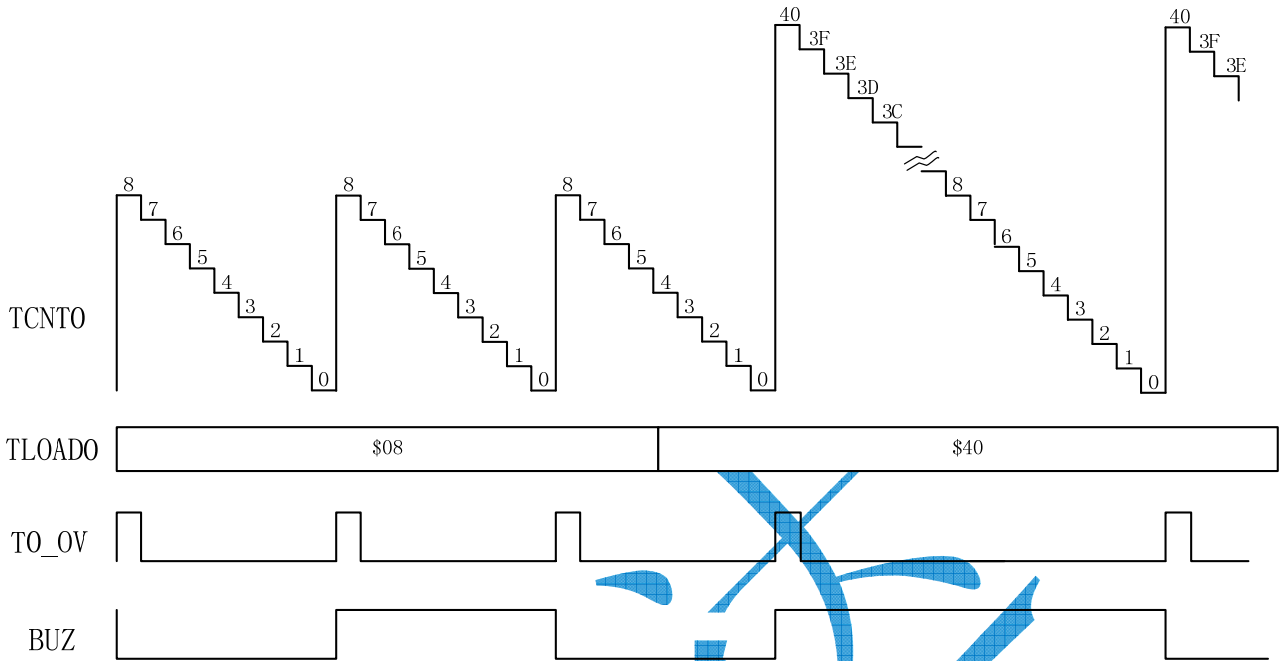
T0 定时周期的计算公式如下：

$$\text{Period}_{(T0)} = \frac{2}{F_{osc}} * (TLOAD0 + 1) * \text{分频倍数}$$

例如，选用 4MHz 振荡器，TLOAD0=\$63，TOPTA=0，TOPR=011，则 T0 的周期为 0.25us*2*(99+1)*16=800us。

当 BUZOE 置为 1 时，P14 输出 BUZ 信号，其通用 IO 口的功能被自动禁止。BUZ 是周期为 2*Period_(T0) 的方波。

定时器 T0 的计数过程、溢出信号和 BUZ 信号的波形示意图如下。



与定时器 T0 相关的寄存器说明如下。

\$07	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCNT0	T0C7	T0C6	T0C5	T0C4	T0C3	T0C2	T0C1	T0C0
R	R	R	R	R	R	R	R	R
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T0C[7:0] - TCNT0 的值，这是一个只读寄存器，用于访问 TCNT0 的当前值。

\$07	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TLOAD0	T0L7	T0L6	T0L5	T0L4	T0L3	T0L2	T0L1	T0L0
W	W	W	W	W	W	W	W	W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T0L[7:0] - TLOAD0 的值，这是一个只写寄存器，用于设置 TLOAD0 的值。

注：将 TLOAD0 设置成 \$00 会使 T0 停止计数。

\$08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCR0	T0IF	T0IM	T0PR2	T0PR1	T0PR0	T0PTA	T0PTS	BUZOE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	1	0	1	1	0	0	0

BIT[7] T0IF - T0 溢出标志位

0: T0 未溢出

1: T0 溢出

写 0 清标志，写 1 无效。

BIT[6] T0IM - T0 中断屏蔽位

0: TMIO 中断允许

1: TMIO 中断禁止

系统复位时，会把 TOIM 置 1，从而屏蔽 TMIO 中断，要允许 TMIO 中断，必须用软件把 TOIM 清 0。TOIM 只用于屏蔽中断请求，不影响 TOIF。

BIT[5:3] TOPR[2:0] - T0 预分频倍数选择

T0 和 WDT 共用预分频器的分频率的选择位，系统复位时被置为 011。TOPR[2:0] 的值和分频倍数的对应关系见下表。

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

BIT[2] TOPTA - 预分频器分配

0: 预分频器分配给 T0

1: 预分频器分配给 WDT

BIT[1] TOPTS - T1 时钟源选择

0: T1 时钟源为 T1PTS 的设置

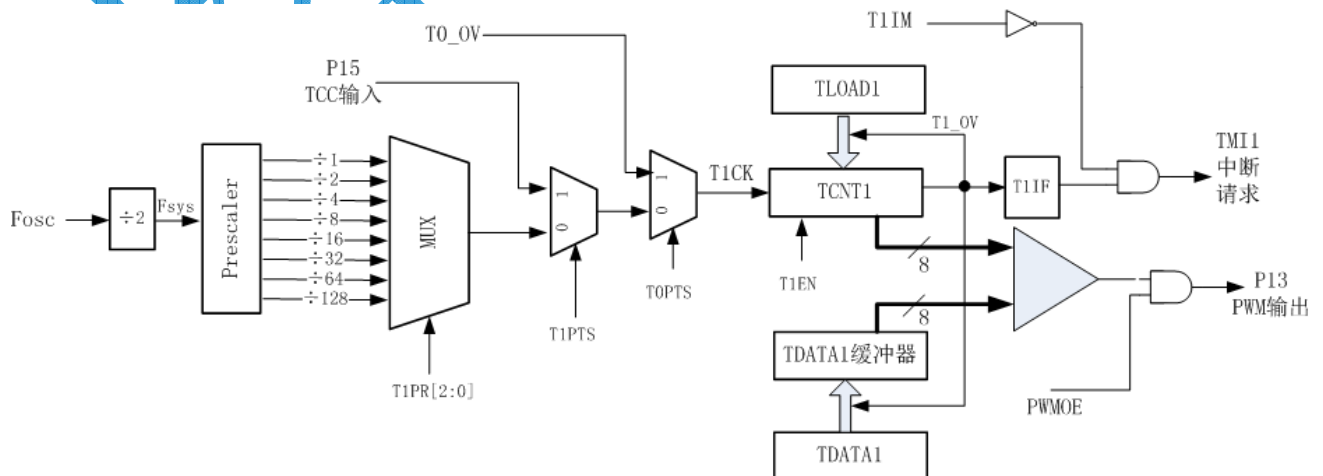
1: T1 时钟源为 T0 的溢出信号（此时 T1PR 和 T1PTS 无效）

BIT[0] BUZOE - BUZ 输出使能

0: 禁止 BUZ 输出，P14 作为 I/O 口

1: 允许 BUZ 输出，P14 输出 BUZ 信号

2.5.2 定时/计数器 T1



20P801 的定时/计数器 T1 由 8 位计数器 TCNT1、8 位自加载寄存器 TLOAD1、8 位比较寄存器 TDATA1（及其缓冲器）、可编程预分频器和控制寄存器 TCR1 组成。

TCNT1 的时钟 T1CK 有三种来源，即

- (1) 系统主时钟 Fsys (Fosc/2) 的分频信号；

- (2) TCC 输入 (P15 引脚)
- (3) T0_OV (定时器 T0 信号的溢出信号), 这种使用方式相当于 T0 和 T1 串联成一个 16 位定时器。

TCNT1 的这三种时钟源选择由 T1PTS 和 T0PTS (见 § 0) 决定, 如下表所示。

T0PTS	T1PTS	T1CK 的来源
0	0	Fsys 的分频信号
0	1	TCC 输入
1	X	T0 的溢出信号

TCNT1 是一个递减计数器, T1EN=1 时计数器使能, T1EN=0 时计数停止, 且 TCNT1 的值复位到 \$00。TCNT1 的值可以读出 (不可写)。当计数到零时, 产生溢出信号 T1_OV, 这个信号有 3 个功能:

- (1) 作为 TLOAD1 的自动加载信号, 即溢出发生时, 将 TLOAD1 的值载入 TCNT1;
- (2) 将 T1IF 标志位置 1, 如果此时中断屏蔽位 T1IM=0 则产生 TMI1 中断请求;
- (3) 作为 TDATA1 双缓冲的加载信号, 即溢出发生时, 将 TDATA1 的值载入 TDATA1 缓冲器。

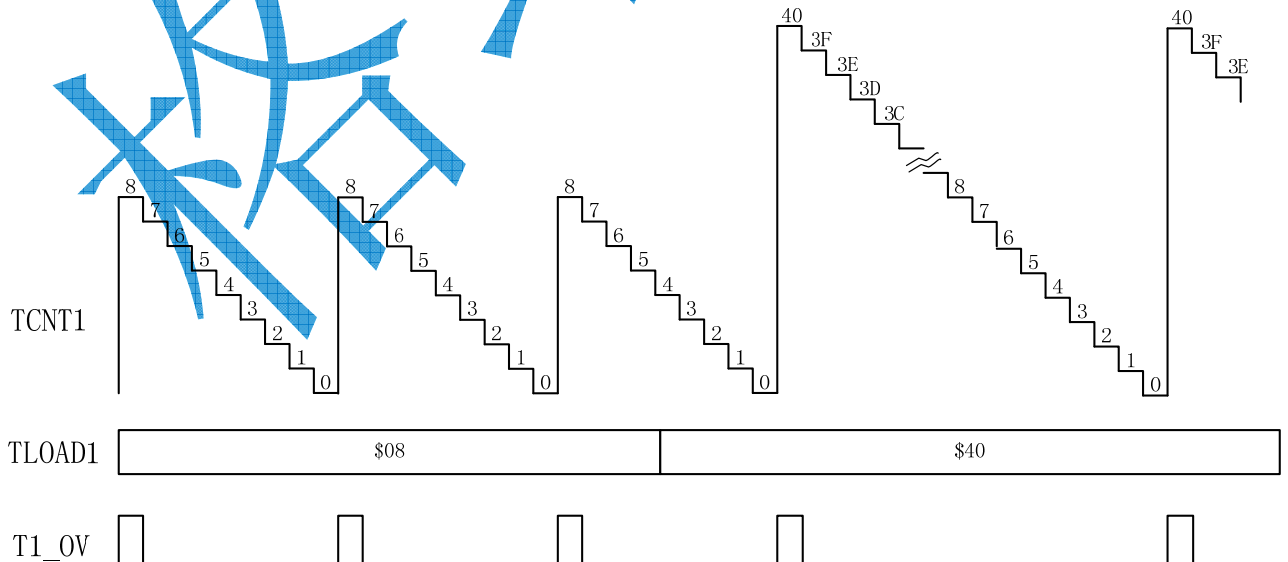
TLOAD1 是一个只写寄存器, 它与 TCNT1 共用同一个地址 \$0C。写 TLOAD1 会对 T1 预分频器清零。在 TCNT1 计数过程中对 T1EN 写 0, 则 TCNT1 停止计数, 并将 TCNT1 清零; 然后对 T1EN 写 1, TCNT1 将立刻加载 TLOAD1 并开始计数。

T1 定时周期的计算公式如下:

$$\text{Period}_{(T1)} = \frac{2}{\text{Fosc}} * (\text{TLOAD1} + 1) * \text{分频倍数}$$

例如, 选用 4MHz 振荡器, TLOAD1=\$C7, T1PTA=0, T1PR=100, 则 T1 的周期为 0.25us*2*(199+1)*16=1600us。

定时器 T1 的计数过程、溢出信号的波形示意图如下。



TDATA1 是用来设置 PWM 占空比的, 它具有双重缓存器结构。当 T1EN=1, TCNT1 在计数过程中, 此时对 TDATA1 写入一个新的值, 修改后的值并不会立刻刷新 TDATA1 缓冲器, 而是等到 TCNT1 计数溢出时才更新 TDATA1 缓冲器, 也就是说对 TDATA1 写的数要到计数的下

一个周期才开始生效。当 T1EN=0, 计数器停止计数, 这时对 TDATA1 写数会同时更新 TDATA1 缓冲器。

计数器 TCNT1 在减计数过程中不断与 TDATA1 缓冲器的值相比较, 当 TCNT1 ≥ TDATA1 时 PWM 输出低电平; 当 TCNT1 < TDATA1 时 PWM 输出高电平。

当 PWMOE 置为 1 时, P13 输出 PWM 信号, 其通用 IO 口的功能被自动禁止。

PWM 信号的周期就是 T1 定时周期。

PWM 信号高电平的时间计算公式:

$$T_{High} = \frac{2}{F_{osc}} * (TDATA1) * \text{分频倍数}$$

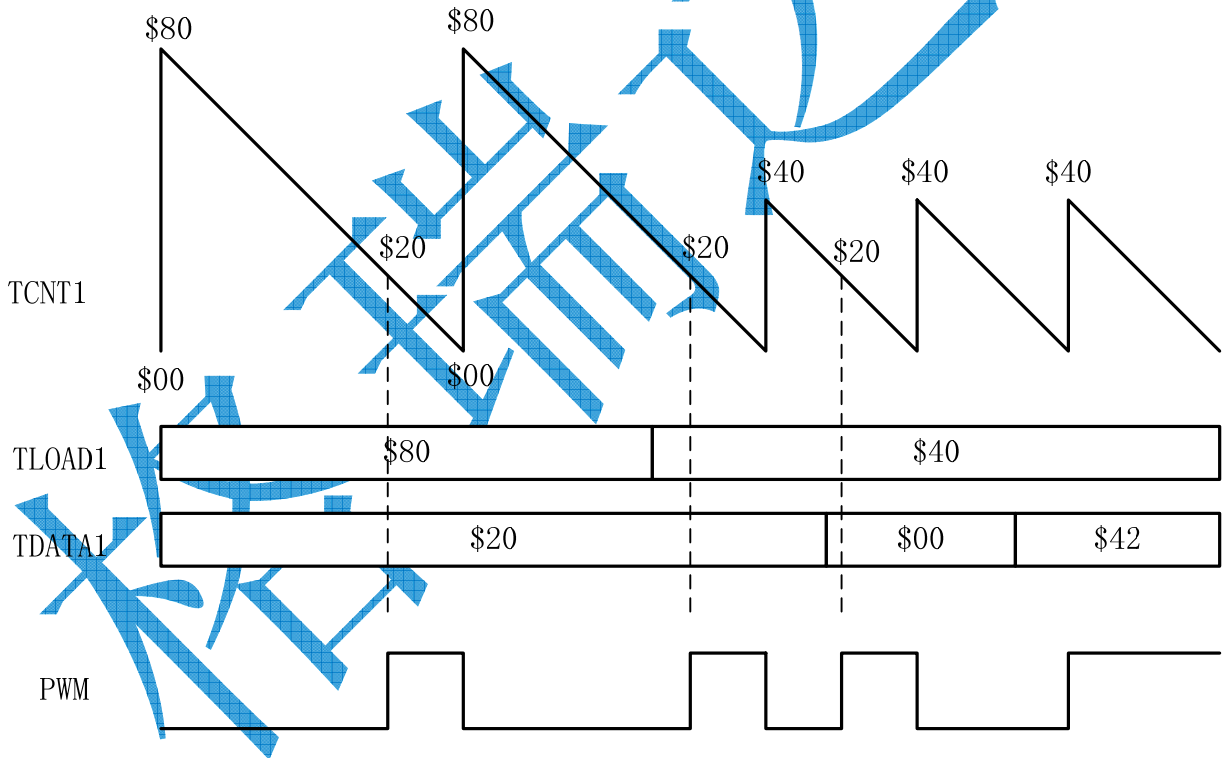
PWM 信号占空比为:

$$Duty = \frac{TDATA1}{TLOAD1 + 1}$$

注 1: 如将 TDATA1 设为 \$00, 则 PWM 输出低电平。

注 2: 如将 TDATA1 设为大于 TLOAD1, 则 PWM 输出高电平。

PWM 信号波形示意图如下。



与定时器 T1 相关的寄存器说明如下。

\$0A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCNT1	T1C7	T1C6	T1C5	T1C4	T1C3	T1C2	T1C1	T1C0
R	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

BIT[7:0] T1C[7:0] - TCNT1 的值, 这是一个只读寄存器, 用于访问 TCNT1 的当前值。

\$0A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
------	-------	-------	-------	-------	-------	-------	-------	-------

TLOAD1	T1L7	T1L6	T1L5	T1L4	T1L3	T1L2	T1L1	T1L0
W	W	W	W	W	W	W	W	W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] T1L[7:0] - TLOAD1 的值，这是一个只写寄存器，用于设置 TLOAD1 的值。

注：将 TLOAD1 设置成\$00 会使 T1 停止计数。

\$09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TDATA1	T1D7	T1D6	T1D5	T1D4	T1D3	T1D2	T1D1	T1D0
W	W	W	W	W	W	W	W	W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] T1D[7:0] - TDATA1 的值，这是一个只写寄存器，用于设置 TDATA1 的值。

\$0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCR1	T1IF	T1IM	T1PR2	T1PR1	T1PR0	T1EN	T1PTS	PWMOE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	1	0	0	0	0	0	0

BIT[7] T1IF - T1 溢出标志位

0: T1 未溢出
1: T1 溢出
写 0 清标志，写 1 无效。

BIT[6] T1IM - T1 中断屏蔽位

0: TMI1 中断允许
1: TMI1 中断禁止
系统复位时，会把 T1IM 置 1，从而屏蔽 TMI1 中断，要允许 TMI1 中断，必须用软件把 T1IM 清 0。T1IM 只用于屏蔽中断请求，不影响 T1IF。

BIT[5:3] T1PR[2:0] - T1 预分频倍数选择

T0 预分频器的分频率的选择位。T1PR[2:0] 的值和分频倍数的对应关系见下表。

T1PR[2]	T1PR [1]	T1PR [0]	T1
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

BIT[2] T1EN - T1 使能位

0: T1 停止计数
1: T1 允许计数

BIT[1] T1PTS - T1 时钟源选择

- 0: T1 时钟为系统时钟 Fsys 的分频
- 1: T1 时钟为 TCC 信号 (下降沿), 此时 P15 通用 I/O 口的功能被自动禁止

BIT[0] PWMOE - PWM 输出使能

- 0: 禁止 PWM 输出, P13 作为 I/O 口
- 1: 允许 PWM 输出, P13 输出 PWM 信号

2.5.3 WDT 定时器

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

当 MCR 的 WDTM=0 时, WDT 定时器作为 WDT 看门狗复位用; WDTM=1 时, WDT 定时器用来产生 WDTI 中断。

因为 WDT 定时器的时钟源与系统主时钟无关, 所以, 即使系统进入 STOP 状态, WDT 定时器仍会工作, WDT 复位或 WDTI 中断还是可以正常工作。

WDT 定时器是一个 11 级计数器, 当计数溢出时, 产生 WDT 复位或将 WDTF 标志位置 1 产生 WDTI 中断请求。WDTC 是 WDT 定时器的清零位, 对 WDTC 写 1 就会将 WDT 定时器的值清零。WDT 定时器的当前计数值是不可访问的。

考虑到与分频倍数, WDT 定时器周期的范围是 $2^{11}/F_{wdt} \sim 2^{18}/F_{wdt}$ 。由于 Fwdt 的典型值是 10KHz, WDT 定时器周期范围约为 205ms~26.2s。

与 WDT 定时器相关的寄存器说明如下。

\$0F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCR	KBIE	KBIC	-	WDTC	WDTF	WDTM	-	-
R/W	R/W	R/W	-	R/W	R/W	R/W	-	-
初始值	0	0	-	0	0	0	-	-

BIT[4] WDTC - WDT 定时器清零位
读 WDTC 始终为 0; 写 1 能清 WDT 定时器; 写 0 无效。

BIT[3] WDTF - WDT 定时器溢出标志
0: WDT 未溢出
1: WDT 溢出
写 0 清标志, 写 1 无效。

BIT[2] WDTM - WDT 定时器工作模式选择
0: WDT 溢出产生复位
1: WDT 溢出产生中断

2.6 中断

2.6.1 概述

20P801 的中断有外中断 (INT0、INT1)、键盘中断 (KBI)、定时器中断 (TMI0、TMI1)、WDT 溢出中断 (WDTI) 和软中断 (SWI)。外部中断、键盘中断、定时器中断和 WDT 溢出中断可被 CPU 状态寄存器 CCR 的 I 位屏蔽, 软中断不受屏蔽位 I 的影响。软中断 SWI 属于指令系统的一部分, 详细介绍见《HC05 指令集》。

中断响应过程如下:

- ◇ 当发生中断请求时, CPU 将相关状态寄存器的内容压栈保存 (共 5 个字节), 对中断屏蔽位 I 置 1, 禁止其他中断。与复位不同, 硬件中断不停止当前指令的执行, 而是暂时挂起中断直到当前指令执行完成。
- ◇ CPU 执行中断时, 首先到相应的中断向量中取出中断服务程序的入口地址, 然后跳转到中断服务程序中执行。
- ◇ 每个中断服务程序都应有 RTI 指令, 表示中断服务程序结束, 这时, 从堆栈取出状态寄存器的值, 然后从中断发生时的那条指令的后一条指令继续执行。

20P801 的中断向量地址见下表。中断优先级按表中次序由下到上依次降低。

向量地址	中断
\$1FE0~\$1FEF	保留
\$1FF0:\$1FF1	WDTI
\$1FF2:\$1FF3	KBI
\$1FF4:\$1FF5	TMI1
\$1FF6:\$1FF7	TMI0
\$1FF8:\$1FF9	INT1
\$1FFA:\$1FFB	INT0
\$1FFC:\$1FFD	SWI
\$1FFE:\$1FFF	RESET

2.6.2 外中断

20P801 的 P14 和 P15 可以作为外中断输入 INT0 和 INT1, 可以响应上升沿、下降沿、高电平、低电平 4 种方式的中断触发条件。

外部中断 INT0 控制位功能如下:

- (1) INT0E 为中断使能位, INT0E=0 时, 不允许外中断; 当 INT0E=1 时, 允许外中断。
- (2) INT0M[1:0] 为中断触发位, 有下降沿触发、上升沿触发、低电平触发和高电平触发四种方式。
- (3) INT0F 为中断标志位, INT0F 不会自动清零, 必须通过软件对其清零。当 INT0E=0 时, INT0F 不受端口变化的影响。

外中断 INT1 控制位功能同外中断 INT0 类似。

注: 要使用外中断 INT0, 还必须将 P14 口设置成输入状态, 即令 DDR14=0; 类似的, 要使用外中断 INT1, 还必须将 P15 口成输入状态, 即令 DDR15=0。

相关寄存器如下。

\$OE	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTC	INT0E	INT0M1	INT0M0	INT0F	INT1E	INT1M1	INT1M0	INT1F

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] INT0E - INT0 使能位

0: INT0 关闭

1: INT0 打开

BIT[6:5] INT0M[1:0] - INT0 沿口选择

00: 下降沿触发

01: 上升沿触发

10: 低电平触发

11: 高电平触发

BIT[4] INT0F - INT0 标志位

0: 无 INT0 中断请求

1: 有 INT0 中断请求

写 0 清标志, 写 1 无效。

BIT[3] INT1E - INT1 使能位

0: INT1 关闭

1: INT1 打开

BIT[2:1] INT1M[1:0] - INT1 沿口选择

00: 下降沿触发

01: 上升沿触发

10: 低电平触发

11: 高电平触发

BIT[0] INT1F - INT1 标志位

0: 无 INT1 中断请求

1: 有 INT1 中断请求

写 0 清标志, 写 1 无效。

2.6.3 键盘中断

20P801 的 P15-P10 可以作为键盘中断输入, 这些键盘中断请求信号共用一个中断请求端和一个中断向量, 因而在中断服务程序中通常还要读取 I/O 数据寄存器来判断究竟是一个键盘输入口有中断请求。

键盘中断请求与以下因素有关。

(1) KBIE 位, 这是 MCR 寄存器的一位。KBIE 是键盘中断允许位, 当 KBIE=1 时, 允许键盘中断, KBIE=0 时, 不允许键盘中断。

(2) KBIM[5:0] (对应 P1[5:0]), 当 KBIM_n=1 时, 表示 P1_n 的键盘中断功能打开, 同时将 P1_n 端口置为输入状态, 否则, 键盘中断功能关闭。

(3) P15-P10 的状态, 引脚输入电平状态与输出数据寄存器中的值进行比较, 如不同, 则触发键盘中断请求。

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

另外, MCR 还有一个控制位 KBIC 与键盘中断有关。当键盘中断请求产生并被响应后, 需要对 KBIC 位写 1, 否则键盘中断请求会被锁存, 也就是说, 如不对 KBIC 写 1, 则键盘中断将不停地被响应。

相关寄存器说明如下。

\$0F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCR	KBIE	KBIC	-	WDTC	WDTF	WDTM	-	-
R/W	R/W	R/W	-	R/W	R/W	R/W	-	-
初始值	0	0	-	0	0	0	-	-

BIT[7] KBIE - 键盘中断使能位

0: 禁止键盘中断

1: 允许键盘中断

BIT[6] KBIC - 键盘中断信号锁存

写 0: 无效

写 1: 清除键盘中断锁存信号

读 KBIC 的结果总为 0。

\$08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
KBIM	-	-	KBIM5	KBIM4	KBIM3	KBIM2	KBIM1	KBIM0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] KBIM0n - P1n 口键盘中断允许位 (n=5-0)

0: 不允许键盘中断

1: 允许键盘中断 (自动将 P1n 口设置成输入状态)

2.6.4 定时器中断

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

2.6.5 WDT 中断

当 WDTM=1 时, WDT 溢出会产生中断请求 WDTI。详见 §0。

2.7 系统工作模式

20P801 有两种低功耗工作方式：STOP 模式和 WAIT 模式。

2.7.1 STOP 模式

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

- ✧ 系统主时钟的振荡器停止振荡
- ✧ 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持原态不变
- ✧ 所有的内部操作全部停止

以下情况使 MCU 退出 STOP 方式：

- ✧ 有外中断 INTO、INT1 请求发生
- ✧ 有键盘中断 KBI 请求发生
- ✧ 有外部计数溢出中断 TMI1（对 TCC 计数时）请求发生
- ✧ 有 WDT 溢出 WDTI 中断请求发生
- ✧ 任何形式的系统复位发生

STOP 工作模式下，系统停止了几乎所有的操作，所以整体功耗水平非常低。

注：MCU 从 STOP 模式唤醒后会等待 N 个 Fosc 周期（振荡器重新起振并稳定），然后开始执行指令，N 的值由 T0 的分频系数决定，其计算方式如下

$$N = 2^{T0\text{分频系数}} \times 128$$

2.7.2 WAIT 模式

执行 WAIT 指令 MCU 使进入 WAIT 低功耗模式，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
- ✧ 定时器保持工作
- ✧ 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持原态不变
- ✧ 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：

- ✧ 任何形式的中断请求发生
- ✧ 任何形式的系统复位发生

WAIT 工作模式下，CPU 停止工作，但系统主时钟的振荡器仍维持振荡，整体功耗水平有所降低。

2.8 电气参数

2.8.1 极限参数

参数	符号	值	单位
工作电压	VDD	-0.3 ~ 6.5	V
输入电压	VIN	VSS-0.3 ~ VDD+0.3	V
工作温度	TA	-40 ~ 85	°C
储存温度	Tstg	-65 ~ 150	°C

2.8.2 直流电气参数

VDD=5V, T=25°C

特性	符号	引脚	条件	最小	典型	最大	单位
工作电压	VDD		2M-4M	2.0		5.5	V
			2M-16M	2.7		5.5	
输入漏电	V_{leak}	所有输入脚	VIN=VDD,0			±1	uA
输入高电平	V_{ih}	所有输入脚		0.6VDD		VDD	V
输入低电平	V_{il}	所有输入脚		0		0.3VDD	V
上拉电阻	R_{U1}	P10-P15		10	25	50	Kohm
下拉电阻	R_D	P10,P11,P13-P15		10	25	50	Kohm
输出高电平驱动电流	I_{oh}	所有输出脚	$V_{oh}=VDD-0.7V$	6			mA
输出低电平驱动电流	I_{ol}	所有输出脚	$V_{ol}=0.6V$	10			mA
静态功耗	I_{dds}	VDD	关 LVR,关 WDT			1	uA
			开 WDT			1	
			开 LVR			1	
动态功耗	I_{ddc}	VDD	VDD=5V Fosc=4MHz 无负载			3	mA
LVR	V_{lvr}		LVR=3.6V	3.2	3.6	4.0	V
			LVR=2.1V	2.0	2.1	2.2	

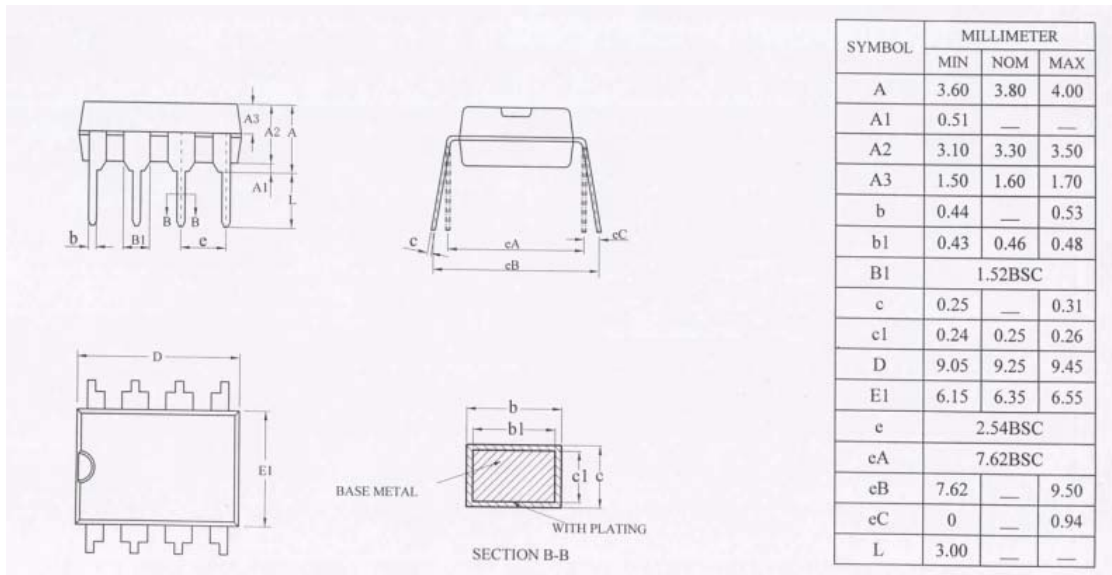
2.8.3 交流电气参数

VDD=5V, T=25°C

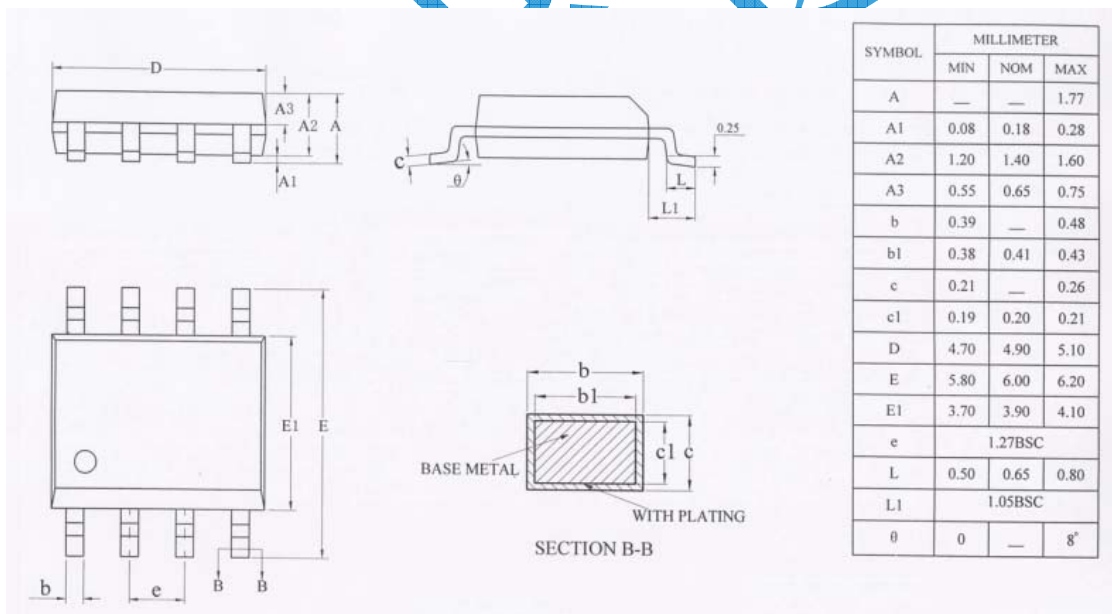
特性	符号	条件	最小	典型	最大	单位
内部高频 RC 振荡频率	F _{hrc1}	T=25°C VDD=5V	-1%	1	+1%	MHz
			-1%	2	+1%	
			-1%	4	+1%	
			-1%	8	+1%	
	F _{hrc2}	T=-40°C ~ 85°C VDD=5V	-2%	1	+2%	MHz
			-2%	2	+2%	
			-2%	4	+2%	
			-2%	8	+2%	
	F _{hrc3}	T=25°C VDD=2 ~ 5.5V	-1.5%	1	+1.5%	MHz
			-1.5%	2	+1.5%	
			-1.5%	4	+1.5%	
			-1.5%	8	+1.5%	
F _{hrc4}	T=-40°C ~ 85°C VDD=2~5.5V	-2.5%	1	+2.5%	MHz	
		-2.5%	2	+2.5%		
		-2.5%	4	+2.5%		
		-2.5%	8	+2.5%		
WDT 振荡器频率	F _{wdt}	T=25°C VDD=5V		10		KHz
振荡器起振时间	T _{oxov}				10	us

2.9 封装外形尺寸

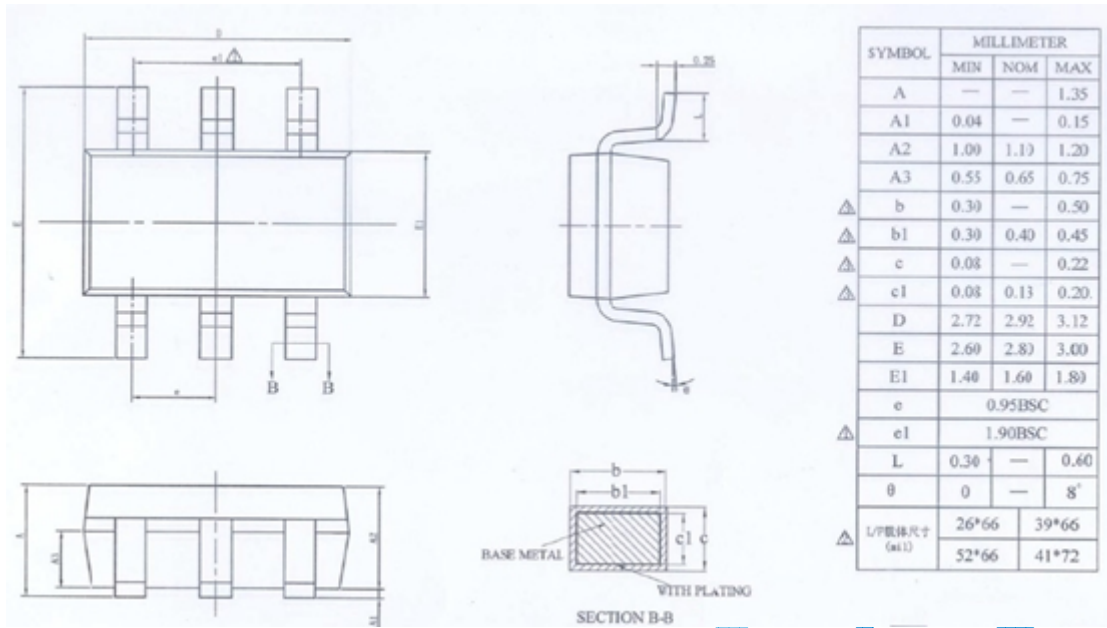
DIP8



SOP8



SOT23-6

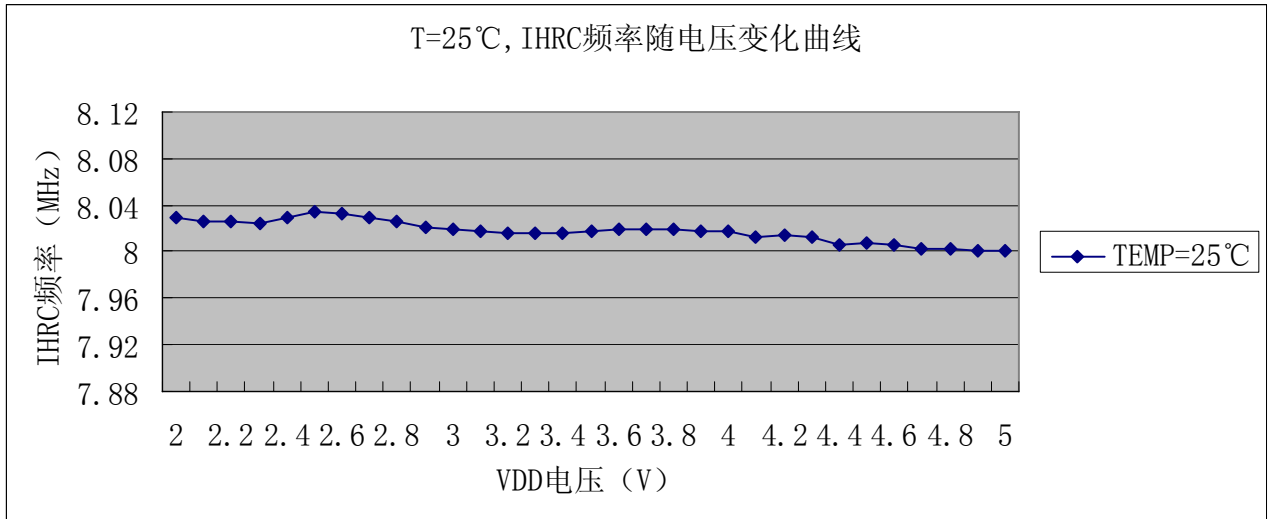


格瑞达

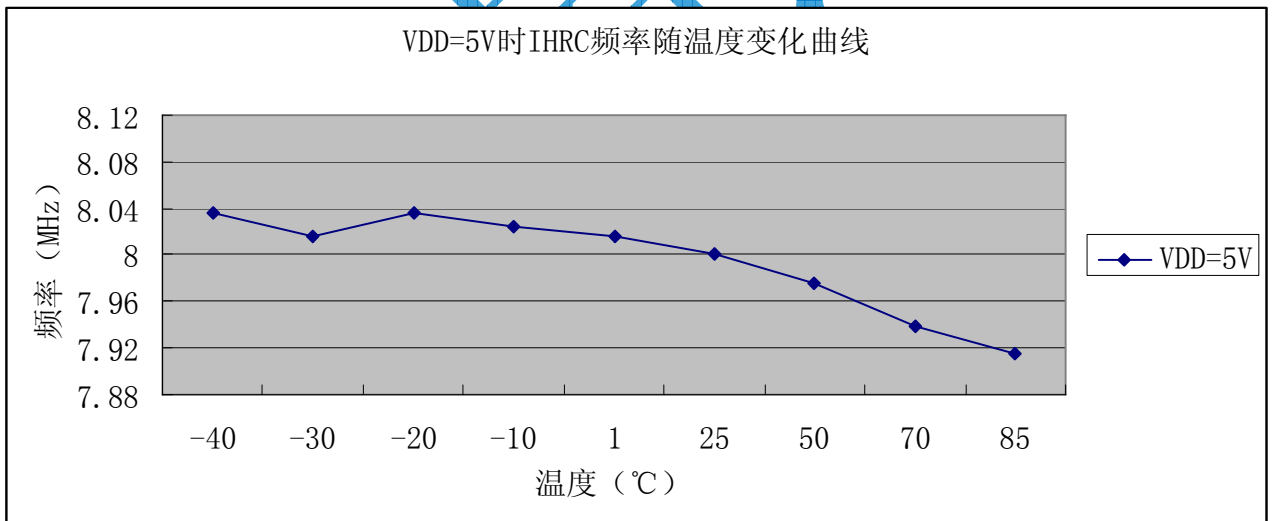
2.10 附录

注：附录的内容仅供参考。

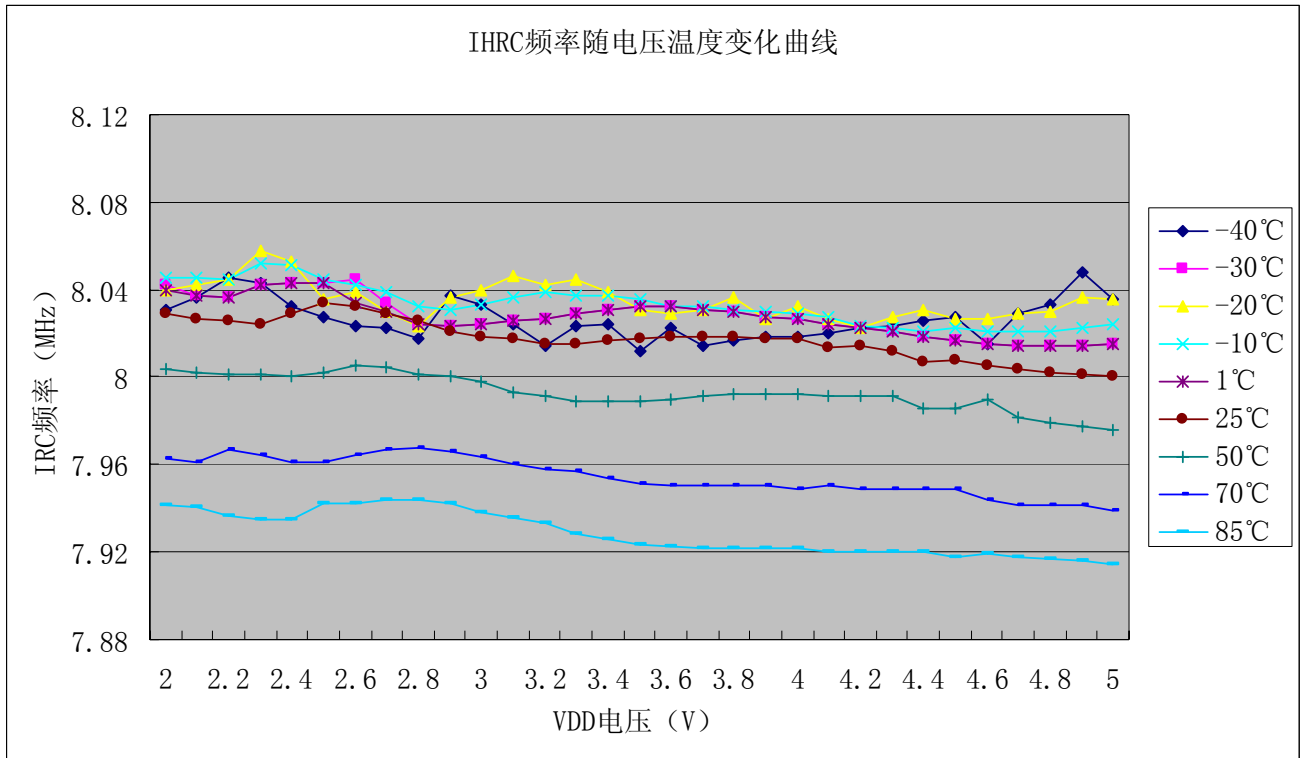
(1) 常温下内置 RC 振荡频率与工作电压的关系曲线



(2) 5V 工作电压下内置 RC 振荡频率与环境温度的关系曲线



(3) 内置 RC 振荡频率与工作电压和环境温度的关系曲线



第 3 章 H05 指令集

3.1 简介

HC05 指令是一个复杂指令集 (CISC)。本文将叙述 CPU 的结构、寻址方式、指令分类等，最后将逐一对每一条指令进行详细描述。

3.2 CPU (中央处理器)

HC05 的 CPU 框图见图 1。

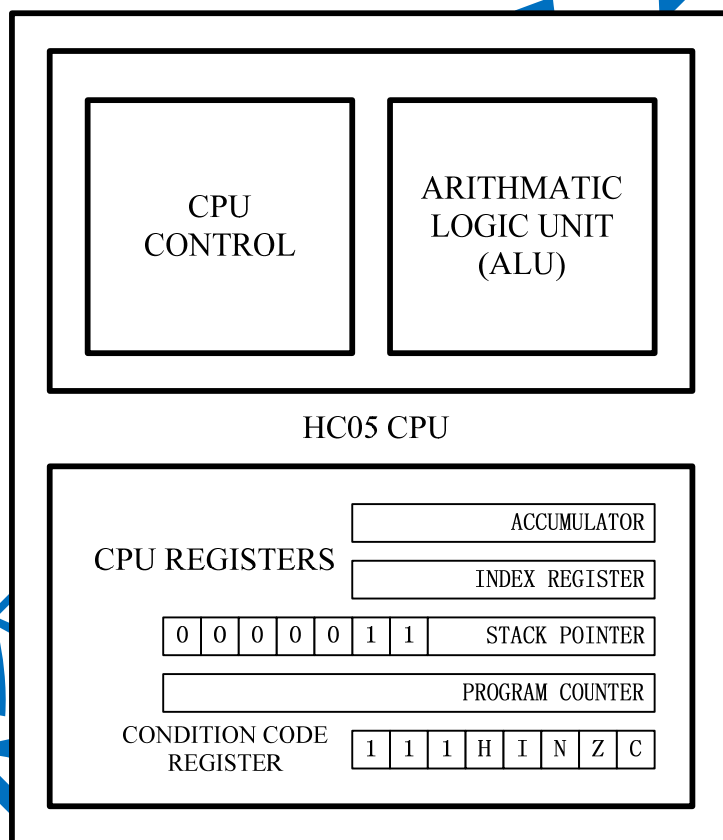


图1

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。

3.2.2.1 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）。

3.2.2.2 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。

3.2.2.3 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。

3.2.2.4 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。

3.2.2.5 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。

3.2.2.6 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。

格瑞达

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：

- ◇ 当前变址寄存器 X 的值
- ◇ 操作码的地址偏移量

变址寻址分为 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

3.3.5.1 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

3.3.5.2 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（\$00FE+\$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

3.3.5.3 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）
- [5] 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

3.3.6 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 Lable 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。

下面表 1 到表 4 是按照功能分类的所有指令的汇总。

表 1 寄存器/存储器指令

		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		
功能	助记符	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	MPJ	-	-	-	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
调用子程序	SRJ	-	-	-	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

表 2 读/写/修改指令

		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		
功能	助记符	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

3.4 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	∨ 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	$A \leftarrow (A) + (M) + (C)$	—	—	◆	◆	◆	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
ADD #opr	加法	$A \leftarrow (A) + (M)$	—	—	◆	◆	◆	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
AND #opr	逻辑与	$A \leftarrow (A) \wedge (M)$	—	—	◆	◆	—	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
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
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

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	AD	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	清零	$M \leftarrow \$00$	—	—	0	1	—	DIR	3F	dd	5
CLRA		$A \leftarrow \$00$						INH	4F		3

CLR _X		X←\$00						INH	5F		3
CLR opr,X		M←\$00						IX1	6F	ff	6
CLR ,X		M←\$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)	—	—	◆	◆	◆	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
COM opr	取反	M←\$FF-(M)	—	—	◆	◆	1	DIR	33	dd	5
COMA		INH						43		3	
COMX		INH						53		3	
COM opr,X		IX1						63	ff	6	
COM ,X		IX						73		5	
								M←\$FF-(M)			
CPX #opr	X 寄存器比较	(X)-(M)	—	—	◆	◆	◆	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
DEC opr	减一	M←(M)-1	—	—	◆	◆	—	DIR	3A	dd	5
DECA		INH						4A		3	
DECX		INH						5A		3	
DEC opr,X		IX1						6A	ff	6	
DEC ,X		IX						7A		5	
								M←(M)-1			
EOR #opr	逻辑异或	A←(A) (M)	—	—	◆	◆	—	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

INC opr	加一	M←(M)+1	-	-	◆	◆	-	DIR	3C	dd	5
INCA		A←(A)+1						INH	4C		3
INCX		X←(X)+1						INH	5C		3
INC opr,X		M←(M)+1						IX1	6C	ff	6
INC ,X		M←(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)	-	-	◆	◆	-	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
LDX #opr	X 寄存器存数	X←(M)	-	-	◆	◆	-	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

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		—	—	0	◆	◆	DIR	34	dd	5
LSRA			—	—	0	◆	◆	INH	44		3
LSRX			—	—	0	◆	◆	INH	54		3
LSR opr,X			—	—	0	◆	◆	IX1	64	ff	6
LSR ,X			—	—	0	◆	◆	IX	74		5
MUL	乘法 (暂不支持)	$X:A \leftarrow (X) \times (A)$	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	
ORA #opr	逻辑或	$A \leftarrow (A) \vee (M)$	—	—	—	—	—	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
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←\$00FF	—	—	—	—	—	INH	9C		2
RTI	中断返回	SP←(SP)+1; Pull (CCR)	◆	◆	◆	◆	◆	INH	80		9
		SP←(SP)+1; Pull (A)	◆	◆	◆	◆					
		SP←(SP)+1; Pull (X)	◆	◆	◆	◆					
		SP←(SP)+1; Pull (PCH)	◆	◆	◆	◆					
		SP←(SP)+1; Pull (PCL)	◆	◆	◆	◆					
RTS	子程序返回	SP←(SP)+1; Pull (PCH)	—	—	—	—	—	INH	81		6
		SP←(SP)+1; Pull (PCL)	—	—	—	—	—				

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	$A \leftarrow (A) - (M) - (C)$	—	—	◆	◆	◆	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
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	$A \leftarrow (A)-(M)$	—	—	◆	◆	◆	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
SWI	软中断	PC←(PC)+1	—	1	—	—	—	INH	83		10
		Push (PCL); SP←(SP)-1									
		Push (PCH); SP←(SP)-1									
		Push (X); SP←(SP)-1									
		Push (A); SP←(SP)-1									
		Push (CCR);SP←(SP)-1									
		I←1									
		PCH←中断向量高位									
		PCL←中断向量地位									

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	v	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

3.5 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。

3.5.1 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	I	N	Z	C		
	1	1	1	◆	—	◆	◆

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

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

3.5.2 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

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

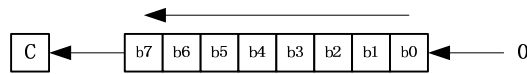
若结果的所有位为 0 则置位，否则清零

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

3.5.4 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

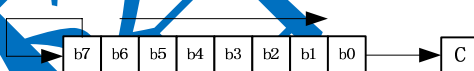
	H	I	N	Z	C
	1	1	1	—	—
N	R7				
	若结果最高位为 1 则置位，否则清零				
Z	$\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$				
	若结果的所有位为 0 则置位，否则清零				
C	b7				
	若移位前第 7 位为 1 则置位，否则清零				

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	
ASL opr	DJR	38	dd	5
ASLA	INH	48		3
ASLX	INH	58		3
ASL opr,X	IX1	68	ff	6
ASL ,X	IX	78		5

3.5.5 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

3.5.6 BCC 无进位则跳 (同 BHS)

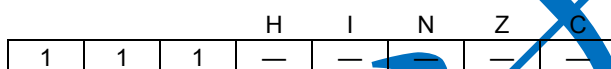
操作

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



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

3.5.7 BCLR n 第 n 位清零

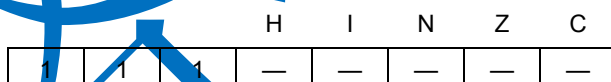
操作

$M_n \leftarrow 0$

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



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

3.5.8 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

3.5.9 BEQ 等于则跳转

操作

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

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

3.5.10 BHCC 无半进位则跳转

操作

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

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

3.5.11 BHCS 半进位则跳转

操作

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

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

3.5.12 BHI 大于则跳转

操作

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

3.5.13 BHS 大于等于则跳转（同 BCC）

操作

$PC \leftarrow (PC) + 2 + rel$ if $(C) \neq 0$, 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

3.5.14 BIH IRQ 为高则跳转④

操作

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

			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	
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。

3.5.15 BIL IRQ 为低则跳转①

操作

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

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

			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	
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。

3.5.16 BIT 位测试

操作

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

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

			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	
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

3.5.17 BLO 小于则跳转 (同 BCS)

操作

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

3.5.18 BLS 小于等于则跳转

操作

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

3.5.19 BMC 中断未屏蔽则跳转

操作

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

3.5.20 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

3.5.21 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

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	
BMS rel	REL	2D	rr	3

3.5.22 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

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	
BNE rel	REL	26	rr	3

3.5.23 BPL 值为正则跳转

操作

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

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

	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	
BPL rel	REL	2A	rr	3

3.5.24 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

	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	20	rr	3

3.5.25 BRCLR n 第 n 位为 0 则跳转

操作

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

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

	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	
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

3.5.26 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

3.5.27 BRSET n 第 n 位为 1 则跳转

操作

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

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

3.5.28 BSET n 第 n 位置位

操作

$$Mn \leftarrow 1$$

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

	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	
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

3.5.29 BSR 跳转到子程序

操作

PC ← (PC) + 2
 Push (PCL); SP ← (SP) - 1;
 Push (PCH); SP ← (SP) - 1
 PC ← (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



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

3.5.30 CLC C 标志清零

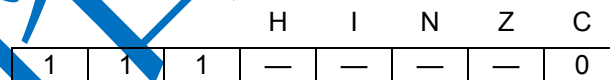
操作

C ← 0

描述

对 C 标志位清零。

对 CCR 标志的影响



C
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

3.5.31 CLI I 标志清零

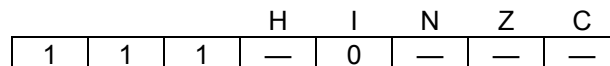
操作

I ← 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



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

3.5.32 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

	H	I	N	Z	C
	1	1	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

3.5.33 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

	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

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

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

3.5.34 COM 取反

操作

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

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

3.5.35 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

3.5.36 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

	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

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

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

3.5.37 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

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

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	
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

3.5.38 INC 加一

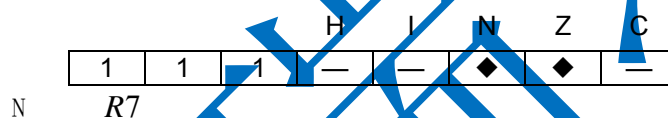
操作

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

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



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	
INC opr	DIR	3C	dd	5
INCA	INH	4C		3
INCA	INH	5C		3
INC opr,X	IX1	6C	ff	6
INC X	IX	7C		5

3.5.39 JMP 跳转

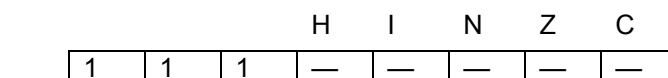
操作

$$PC \leftarrow \text{Effective Address}$$

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



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

3.5.40 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

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	
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

3.5.41 LDA A 寄存器存数

操作

A ← (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

3.5.42 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

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

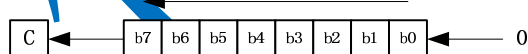
若结果的所有位为 0 则置位，否则清零

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

3.5.43 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

	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

C b7

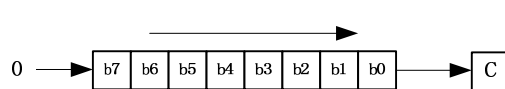
若移位前第 7 位为 1 则置位，否则清零

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

3.5.44 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

	H	I	N	Z	C
	1	1	1	—	—
N	0				
Z	清零				
C					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

3.5.45 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	I	N	Z	C
	1	1	1	0	—
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。

3.5.46 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

	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

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

3.5.47 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

	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	
NOP	INH	9D		2

3.5.48 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

$$R7 \cdot R6 \cdot R5 \cdot R4 \cdot R3 \cdot R2 \cdot R1 \cdot R0$$

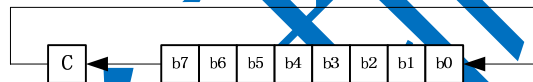
若结果的所有位为 0 则置位，否则清零

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

3.5.49 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

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

若结果的所有位为 0 则置位，否则清零

C

$$b0$$

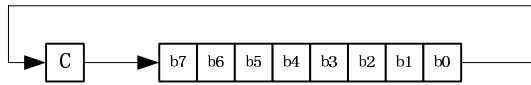
若移位前第 7 位为 1 则置位，否则清零

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

3.5.50 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

	H	I	N	Z	C
	1	1	1	—	—
N	0				
Z	清零 $\overline{R7 \cdot R6 \cdot R5 \cdot R4 \cdot R3 \cdot R2 \cdot R1 \cdot R0}$ 若结果的所有位为 0 则置位，否则清零				
C	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	#	6
ROR ,X	IX	76		5

3.5.51 RSP SP 复位

操作

$$SP \leftarrow \$00FF$$

描述

将堆栈指针复位成 \$00FF。

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	
RSP	INH	9C		2

3.5.52 RTI 中断返回

操作

- SP ← (SP)+1; Pull (CCR)
- SP ← (SP)+1; Pull (A)
- SP ← (SP)+1; Pull (X)
- SP ← (SP)+1; 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

			H	I	N	Z	C
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

3.5.53 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

			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	
RTS	INH	81		6

3.5.54 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

			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

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

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

3.5.55 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

3.5.56 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

3.5.57 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

	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	
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

3.5.58 STOP 进 STOP 模

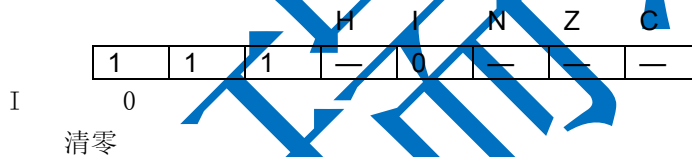
操作

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



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

3.5.59 STX X 寄存器取数

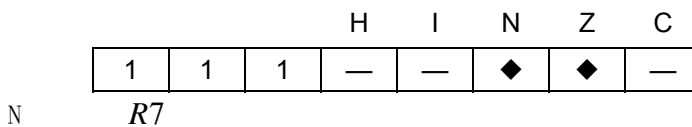
操作

$M \leftarrow (X)$

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



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

3.5.60 SUB 减法

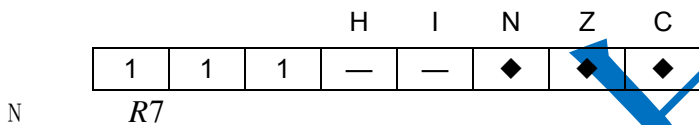
操作

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

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



若结果最高位为 1 则置位，否则清零

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

若结果的所有位为 0 则置位，否则清零

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

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

3.5.61 SWI 软中断

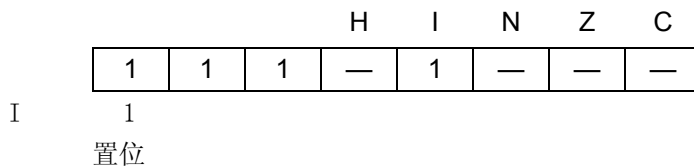
操作

- PC ← (PC) + 1
- Push (PCL); SP ← (SP) - 1
- Push (PCH); SP ← (SP) - 1
- Push (X); SP ← (SP) - 1
- Push (A); SP ← (SP) - 1
- Push (CCR); SP ← (SP) - 1
- I ← 1
- PCH ← 中断向量高位
- PCL ← 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



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

3.5.62 TAX 将 A 的值传到 X

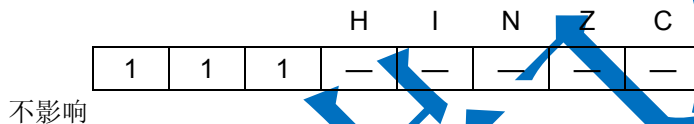
操作

$$X \leftarrow (A)$$

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



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

3.5.63 TST 零测试

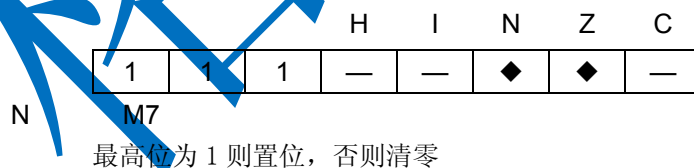
操作

$$(A) - \$00 \text{ 或 } (X) - \$00 \text{ 或 } (M) - \$00$$

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



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

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

3.5.64 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

				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	
TXA	INH	9F		10

3.5.65 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

				H	I	N	Z	C
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	
WAIT	INH	8F		2

第 4 章 订购信息

下单规格	功能简述	芯片型号	封装
20P801P08	8-Bit 通用 I/O 型单片机	20P801	DIP8
20P801S08	8-Bit 通用 I/O 型单片机	20P801	SOP8
20P801S06	8-Bit 通用 I/O 型单片机	20P801	SOT23-6

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