

指令集说明书

使用说明 (第二版)

上海芯旺微电子有限公司
2015. 10

目录

1	六种寻址方式.....	3
1.1	寄存器寻址.....	3
1.2	直接寻址.....	3
1.3	立即数寻址.....	3
1.4	寄存器间接寻址.....	4
1.5	寄存器直接寻址.....	4
1.6	位寻址.....	4
2	汇编指令集表.....	5
3	指令集具体说明.....	7

1 六种寻址方式

KF 系列单片机提供 6 种寻址方式，分别为：寄存器寻址、直接寻址、立即数寻址、寄存器间接寻址、寄存器直接寻址和位寻址。KF 系列单片机的指令可以没有操作数、一个操作数、两个操作数。

1.1 寄存器寻址

采用这种寻址方式的指令中的操作数为寄存器组 R0-R7 的一个。

例：

CLR R0 ; R0←0 将寄存器 R0 清 0
只有一个操作数(R0 的值)，寻址方式为寄存器寻址。

ADD R0, R1 ; R0←R0+R1 将寄存器 R0 与 R1 的和赋值给 R0
两个操作数 (R0 和 R1)，寻址方式为寄存器寻址。

1.2 直接寻址

在指令中的操作数为某个寄存器的直接地址，该地址指出其参与运算的数据所在的地址。直接寻址可以是：特殊功能寄存器、通用数据存储器。

例：

MOV R0, 0X81 ; R0←(81H) 将 81H 单元的数据送到 R0 中
指令中，源操作数寻址方式为直接寻址，目的操作数为寄存器寻址。

INC 0X3B ; 3BH←(3BH)+1 将地址 3BH 里的值加 1，3BH 即 BADDRL。
指令中含有一个操作数，寻址方式为直接寻址。

1.3 立即数寻址

在指令中的操作数为立即数。

例：

MOV R0, #0X20 ; R0←0X20 将立即数 0X20 送到寄存器 R0 中
ADD R0, #0X20 ; R0←(R0)+0X20 寄存器 R0 的值与 0X20 相加结果送到 R0
AND R0, #0X20 ; R0←(R0)&0X20 寄存器 R0 的值与 0X20 相与结果送到 R0
以上三条指令中源操作数都是#0X20，为立即数寻址，目的操作数为寄存器寻址。

1.4 寄存器间接寻址

这种寻址方式中，指令中的寄存器的内容指定操作数的地址，即寄存器中存放的是操作数的地址。间接寻址只有两条指令 LD 和 ST。

例：

LD R0, [R1] ; R0 ← ((R1)) 将 R1 的内容所指地址单元的数据送到 R0
指令中源操作数的寻址方式为寄存器间接寻址，目的操作数为寄存器寻址。

ST [R0], R1 ; (R0) ← (R1) 将 R1 的内容送到 R0 的内容所指向的地址单元
指令中目的操作数的寻址方式为寄存器间接寻址，源操作数为寄存器寻址。

1.5 寄存器直接寻址

这种寻址方式中，目标寄存器的内容可以直接送给某个寄存器组 Rn，以实现寄存器直接寻址。

例：

MOV R0, dir ; 将 dir 目标寄存器的值赋值给 R0

1.6 位寻址

指令中的操作数是寄存器的某位，这样针对位的寻址方式称为位寻址。

例：

CLR INTCTL, 1 ; 将 INTCTL 的第 1 位清 0
CLR 0X80, 1 ; 将 80H 的第 1 位清 0
JNB 0X80, 1 ; 如果 80H 的第 1 位为 0 则跳过下一条指令执行后面的程序

KF 系列单片机汇编指令共有 68 条，除子程序调用、子程序返回、中断返回、部分跳转指令为双周期指令外，其余指令均为单周期指令。所有指令都占两个字节。

按照指令的功能可将其分为：数据传送指令、算术运算指令、逻辑运算指令、位操作指令和转移指令和特殊指令。具体指令集请参考下表。

2 汇编指令集表

助记符、操作数	指令格式	指令说明	周期	影响标志
NOP	0000_0000_0000_0000	空操作指令	1	
NOPZ *	1111_1111_1111_1111	空操作指令	1	
CRET	0000_0000_0000_1000	子程序返回指令	2	
RRET Rn, #data	1011_0rrr_kkkk_kkkk	立即数送到 Rn 中返回	2	
IRET	0000_0000_0000_1001	中断返回指令	2	
CWDT	0000_0000_0110_0100	WDT 清 0	1	
IDLE	0000_0000_0110_0011	进入休眠模式	1	
数据传送指令				
MOV dir	0000_1111_ffff_ffff	$dir \leftarrow (dir)$	1	Z
MOV Rn, dir	0101_rrr0_ffff_ffff	$Rn \leftarrow (dir)$	1	
MOV dir, Rn	0101_rrr1_ffff_ffff	$dir \leftarrow (Rn)$	1	
MOV Rn, #data	1001_lrrr_kkkk_kkkk	$Rn \leftarrow data$	1	
MOV Rn, Rs	1111_1000_1lss_srrr	$Rn \leftarrow (Rs)$	1	
LD Rn, [Rs]	1111_0111_00ss_srrr	$Rn \leftarrow ((Rs))$	1	
ST [Rn], Rs	1111_0111_01ss_srrr	$(Rn) \leftarrow (Rs)$	1	
SWAPR Rn, dir	0100_rrr0_ffff_ffff	$Rn<7:4>=dir<3:0>$ $Rn<3:0>=dir<7:4>$	1	
SWAP dir	0100_rrr1_ffff_ffff	$dir<7:4>=dir<3:0>$ $dir<3:0>=dir<7:4>$	1	
MOVB #data *	1110_0001_kkkk_kkkk	$BANK \leftarrow data$	1	
MOVP #data *	1110_0000_kkkk_kkkk	$PCH \leftarrow data$	1	
算术运算指令				
ADD Rm, dir	0010_0rr0_ffff_ffff	$Rm \leftarrow (Rm) + (dir)$	1	CY、DC、Z
ADD dir, Rm	0010_0rr1_ffff_ffff	$dir \leftarrow (Rm) + (dir)$	1	CY、DC、Z
ADD Rn, #data	1000_0rrr_kkkk_kkkk	$Rn \leftarrow (Rn) + data$	1	CY、DC、Z
ADD Rn, Rs	1111_1000_00ss_srrr	$Rn \leftarrow (Rn) + (Rs)$	1	CY、DC、Z
SUB Rm, dir	0011_lrr0_ffff_ffff	$Rm \leftarrow (dir) - (Rm)$	1	CY、DC、Z
SUB dir, Rm	0011_lrr1_ffff_ffff	$dir \leftarrow (dir) - (Rm)$	1	CY、DC、Z
SUB Rn, #data	1010_0rrr_kkkk_kkkk	$Rn \leftarrow data - (Rn)$	1	CY、DC、Z
SUB Rn, Rs	1111_1000_01ss_srrr	$Rn \leftarrow (Rs) - (Rn)$	1	CY、DC、Z
CMP Rn, #data *	1111_0010_1kkk_krrr	-	1	CY、DC、Z
CMP Rn, Rs *	1111_0001_10ss_srrr	-	1	CY、DC、Z
INC dir	0000_1011_ffff_ffff	$dir \leftarrow (dir) + 1$	1	Z
INCR dir	0000_1010_ffff_ffff	$R0 \leftarrow (dir) + 1$	1	Z
INC Rn	1111_1111_0001_0rrr	$Rn \leftarrow (Rn) + 1$	1	Z
DEC dir	0000_0111_ffff_ffff	$dir \leftarrow (dir) - 1$	1	Z
DECR dir	0000_0110_ffff_ffff	$R0 \leftarrow (dir) - 1$	1	Z
DEC Rn	1111_1111_0000_lrrr	$Rn \leftarrow (Rn) - 1$	1	Z
逻辑运算指令				
AND Rm, dir	0010_lrr0_ffff_ffff	$Rm \leftarrow (Rm) \wedge (dir)$	1	Z
AND dir, Rm	0010_lrr1_ffff_ffff	$dir \leftarrow (dir) \wedge (Rm)$	1	Z
AND Rn, #data	1000_lrrr_kkkk_kkkk	$Rn \leftarrow (Rn) \wedge data$	1	Z
AND Rn, Rs	1111_1000_10ss_srrr	$Rn \leftarrow (Rn) \wedge (Rs)$	1	Z
ORL Rm, dir	0011_0rr0_ffff_ffff	$Rm \leftarrow (Rm) \vee (dir)$	1	Z

助记符、操作数	指令格式	指令说明	周期	影响标志
ORL dir, Rm	001l_0rrl_ffff_ffff	$dir \leftarrow (dir) \vee (Rm)$	1	Z
ORL Rn, #data	100l_0rrr_kkkk_kkkk	$Rn \leftarrow (Rn) \vee data$	1	Z
ORL Rn, Rs	111l_100l_00ss_srrr	$Rn \leftarrow (Rn) \vee (Rs)$	1	Z
XOR Rm, dir	000l_1rr0_ffff_ffff	$Rm \leftarrow (Rm) \oplus (dir)$	1	Z
XOR dir, Rm	000l_1rrl_ffff_ffff	$dir \leftarrow (dir) \oplus (Rm)$	1	Z
XOR Rn, #data	1010_1rrr_kkkk_kkkk	$Rn \leftarrow (Rn) \oplus data$	1	Z
XOR Rn, Rs	111l_100l_0lss_srrr	$Rn \leftarrow (Rn) \oplus (Rs)$	1	Z
CLR Rn	0000_0010_xxxx_1rrr	$Rn=0$	1	Z
CLR dir	0000_001l_ffff_ffff	$dir=0$	1	Z
CPLR dir	0000_0100_ffff_ffff	$R0 \leftarrow \neg (dir)$	1	Z
CPL dir	0000_010l_ffff_ffff	$dir \leftarrow \neg (dir)$	1	Z
CPL Rn	111l_111l_0000_0rrr	$Rn \leftarrow \neg (Rn)$	1	Z
RRCR dir	000l_0000_ffff_ffff	$R0 \leftarrow (dir)$ 带进位 C 循环右移 1 位	1	CY
RRC dir	000l_000l_ffff_ffff	$dir \leftarrow (dir)$ 带进位 C 循环右移 1 位	1	CY
RRC Rn	111l_111l_0010_0rrr	$Rn \leftarrow (Rn)$ 带进位 C 循环右移 1 位	1	CY
RLCR dir	000l_0010_ffff_ffff	$R0 \leftarrow (dir)$ 带进位 C 循环左移 1 位	1	CY
RLC dir	000l_001l_ffff_ffff	$dir \leftarrow (dir)$ 带进位 C 循环左移 1 位	1	CY
RLC Rn	111l_111l_000l_1rrr	$Rn \leftarrow (Rn)$ 带进位 C 循环左移 1 位	1	CY
位操作指令				
CLR dir, b	0110_0bbb_ffff_ffff	将 dir 的 b 位清 0	1	
SET dir, b	0110_1bbb_ffff_ffff	将 dir 的 b 位置 1	1	
CLR Rn, b	111l_1110_00bb_brrr	将 Rn 的 b 位清 0	1	
SET Rn, b	111l_1110_01bb_brrr	将 Rn 的 b 位置 1	1	
转移指令				
DECJZ dir	0000_1000_ffff_ffff	$R0 \leftarrow (dir) - 1$, 为 0 跳过下一条指令	1/2	
DECJZ dir	0000_100l_ffff_ffff	$dir \leftarrow (dir) - 1$, 为 0 跳过下一条指令	1/2	
DECJZ Rn	111l_111l_010l_1rrr	$Rn \leftarrow (Rn) - 1$, 为 0 跳过下一条指令	1/2	
INCRJZ dir	0000_1100_ffff_ffff	$R0 \leftarrow (dir) + 1$, 为 0 跳过下一条指令	1/2	
INCJZ dir	0000_110l_ffff_ffff	$dir \leftarrow (dir) + 1$, 为 0 跳过下一条指令	1/2	
INCJZ Rn	111l_111l_010l_0rrr	$Rn \leftarrow (Rn) + 1$, 为 0 跳过下一条指令	1/2	
JNB dir, b	011l_0bbb_ffff_ffff	dir 的 b 位为 0 跳过下一条指令	1/2	
JB dir, b	011l_1bbb_ffff_ffff	dir 的 b 位为 1 跳过下一条指令	1/2	
JNB Rn, b	111l_011l_10bb_brrr	Rn 的 b 位为 0 跳过下一条指令	1/2	
JB Rn, b	111l_011l_11bb_brrr	Rn 的 b 位为 1 跳过下一条指令	1/2	
JMP #data12	1100_kkkk_kkkk_kkkk	无条件转移指令	2	
CALL #data12	110l_kkkk_kkkk_kkkk	子程序调用指令	2	

dir 为通用寄存器或特殊功能寄存器；Rn、Rs 表示 R0~R7；Rm 表示 R0~R3；#data 表示 8 位立即数；#data12 表示 12 位立即数；b 表示寄存器的第 b 位；[Rn] 表示 Rn 中的数值指向的地址中数据；() 表示特殊功能寄存器、通用数据寄存器或寄存器组中的数据。

带*的仅部分型号具有，为增强型型号。

#data 的指令需要 0xXX 格式的十六进制，否则应为##Dec 的表达。

3 指令集具体说明

当指令使用#的立即数数，应该使用十六进制，如#0x33，如果使用使用 10 进制数，应该采用##51、##-1 的表达。

3.1 NOP NOPZ 空操作指令

语法：NOP
操作数：无
操作：无
受影响标志位：无
说明：两字节指令，用来产生一个机器周期的延时

3.2 CRET 子程序返回指令

语法：CRET
操作数：无
操作：栈顶内容装入 PC
受影响标志位：无
说明：从子程序返回，执行出栈操作，将栈顶单元内容返回装入 PC 程序计数器，双周期指令

3.3 RRET 立即数送到 Rd 中返回指令

语法：RRET Rd, #data
操作数： $0 \leq \#data \leq 255$
操作： $\#data \rightarrow Rd$
受影响标志位：无
说明：将 8 位的立即数 data 装入 Rd。将栈顶内容（返回地址）被装入程序计数器，这是一条两字节双周期指令。

3.4 IRET 中断返回指令

语法：IRET
操作时：无
操作：栈顶内容装入 PC
受影响标志位：无
说明：从中断程序返回，执行出栈操作，将栈顶单元内容返回装入 PC 程序计数器。通过置 1 全局中断允许位 AIE 允许中断，双周期两字节指令

3.5 CWDT 看门狗定时器 WDT 清 0

语法: CWDT
操作数: 无
操作: 00H \rightarrow WDT
0 \rightarrow WDT 预分频器
1 \rightarrow \overline{TO}
1 \rightarrow \overline{PD}

受影响标志位: \overline{TO} 和 \overline{PD}

说明: CWDT 指令用于复位看门狗定时器。他还将复位 WDT 预分频比, 分别将 \overline{TO} 和 \overline{PD} 位置 1。单周期两字节指令。

3.6 IDLE 进入休眠模式

语法: IDLE
操作数: 无
操作: 00H \rightarrow WDT
0 \rightarrow WDT 预分频器
1 \rightarrow \overline{TO}
1 \rightarrow \overline{PD}

受影响标志位: \overline{TO} 和 \overline{PD}

说明: 超时状态位 (\overline{TO}) 位置 1。看门狗定时器及其预分频器被清零。振荡器停止振动, 处理器进入休眠模式

3.7 MOV dir 数据传送指令

语法: MOV dir
操作数: dir, 00H~7FH
操作: $dir \leftarrow (dir)$
受影响标志位: Z

说明: 将通用寄存器或特殊寄存器 dir 的数据送给 dir 单元, 单周期指令

3.8 MOV Rn, dir 数据传送指令

语法: MOV Rn, dir
操作数: dir, Rn。
操作: $Rn \leftarrow (dir)$
受影响标志位: 无

说明: 将通用寄存器或特殊寄存器 dir 的数据送给 Rn 寄存器组单元, 单周期指令

3.9 MOV dir, Rn 数据传送指令

语法: MOV dir, Rn
操作数: Rn, dir。
操作: $dir \leftarrow (Rn)$
受影响标志位: 无
说明: 将寄存器组 Rn 的数据传送给通用寄存器或特殊功能寄存器 dir, 单周期指令

3.10 MOV Rn, #data 数据传送指令

语法: MOV Rn, #data
操作数: #data 八位立即数, 00H~FFH
操作: $Rn \leftarrow data$
受影响标志位: 无
说明: 将 8 位立即数送给寄存器组 Rn 单元, 单周期指令

3.11 MOV Rn, Rs 数据传送指令

语法: MOV Rn, Rs
操作数: Rn, Rs。
操作: $Rn \leftarrow (Rs)$
受影响标志位: 无
说明: 将寄存器组 Rs 单元里的数值传送给寄存器组 Rn 单元, 单周期指令

3.12 LD Rn, [Rs] 数据传送指令

语法: LD Rn, [Rs]
操作数: Rs 单元数据内容所指向的地址单元的数据,
操作: $Rn \leftarrow [Rs]$
受影响标志位: 无
说明: 将寄存器组 Rs 单元里面的内容所指的单元地址的内容传送给 Rn 单元, 单周期指令

3.13 ST [Rn], Rs 数据传送指令

语法: ST [Rn], Rs
操作数: 寄存器组 Rs 单元内容,
操作: $[Rn] \leftarrow Rs$

受影响标志位： 无

说明：将寄存器组 Rs 单元内容送到 Rn 的内容所指的地址单元，单周期指令

3.14 SWAPR Rn, dir 半字节交换指令

语法： SWAPR Rn, dir

操作数： Rn , dir

操作： Rn<7:4>=dir<3:0>

Rn<3:0>=dir<7:4>

受影响标志位： 无

说明：将寄存器组 Rn 的高低半字节分别于通用寄存器组或特殊功能寄存器 dir 的低高半字节进行相互交换，一般用来保护寄存器的值，在中断程序处理之前要进行保护特殊功能寄存器状态不受影响。单周期指令

示例： **SWAPR R1 , PSW**

MOV PSW_TEMP, R1

3.15 SWAP dir 半字节交换指令

语法： SWAP dir

操作数： dir

操作： dir<7:4>=dir<3:0>

dir<3:0>=dir<7:4>

受影响标志位： 无

说明：将通用寄存器或特殊功能寄存器的值进行高低半字节的交换，通常用来进行半字节交换，交换高低位状态。单周期指令

3.16 MOVP dir 给 PCH 赋值立即数

语法： MOVP dir

操作数： dir

操作： PCH=dir

受影响标志位： 无

说明：立即数赋值到 PCH 寄存器，用户实现 CALL 或 JMP 的高 bit 位地址修改。仅增强型型号具有该指令。

3.17 MOVB dir 给 BANK 赋值立即数

语法： MOVB dir

操作数： dir

操作： BANK=dir

受影响标志位： 无

说明：将区定位数据直接赋值到 BANK 寄存器。仅增强型型号具有该指令。

3.18 ADD Rm, dir 加法指令（不带进位）

语法： ADD Rm, dir

操作数： Rm , dir

操作： $Rm \leftarrow (Rm) + (dir)$

受影响标志位： CY、DC、Z

说明：将寄存器组 Rm 的内容与寄存器 dir 的内容相加，结果送给 Rm。当运算结果为 0，将影响零状态标志位 Z，使 Z 置 1。当执行结果的低 4 位向高 4 位有进位（加指令），影响辅助进位标志位 DC，使 DC 置 1。当执行结果（8 位）向高位有进位时（加指令），影响标志位 CY，使 CY 置 1。否则为 0。

3.19 ADD dir, Rn 加法指令（不带进位）

语法： ADD dir, Rn

操作数： dir , Rn

操作： $dir \leftarrow (Rn) + (dir)$

受影响标志位： CY、DC、Z

说明：将寄存器组 Rn 的内容与寄存器 dir 的内容相加，结果送给 dir。当运算结果为 0，将影响零状态标志位 Z，使 Z 置 1。当执行结果的低 4 位向高 4 位有进位（加指令），影响辅助进位标志位 DC，使 DC 置 1。当执行结果（8 位）向高位有进位时（加指令），影响标志位 CY，使 CY 置 1。否则为 0。

3.20 ADD Rn, #data 加法指令（不带进位）

语法： ADD Rn, #data

操作数： Rn , data

操作： $Rn \leftarrow (Rn) + data$

受影响标志位： CY、DC、Z

说明：将寄存器组 Rn 的内容与立即数 data 的内容相加，结果送给 Rn。当运算结果为 0，将影响零状态标志位 Z，使 Z 置 1。当执行结果的低 4 位向高 4 位有进位（加指令），影响辅助进位标志位 DC，使 DC 置 1。当执行结果（8 位）向高位有进位时（加指令），影响标志位 CY，使 CY 置 1。否则为 0。

3.21 ADD Rn, Rs 加法指令（不带进位）

语法： ADD Rn, Rs

操作数： Rn, Rs

操作: $Rn \leftarrow (Rn) + (Rs)$

受影响标志位: CY、DC、Z

说明: 将寄存器组 Rn 的内容与寄存器组 Rs 的内容相加, 结果送给 Rn。当运算结果为 0, 将影响零状态标志位 Z, 使 Z 置 1。当执行结果的低 4 位向高 4 位有进位(加指令), 影响辅助进位标志位 DC, 使 DC 置 1。当执行结果(8 位)向高位有进位时(加指令), 影响标志位 CY, 使 CY 置 1。否则为 0。

3.22 SUB Rm, dir 减法指令(不带进位)

语法: SUB Rm, dir

操作数: Rm, dir

操作: $Rm \leftarrow (dir) - (Rm)$

受影响标志位: CY、DC、Z

说明: 将通用寄存器或特殊功能寄存器 dir 的内容与寄存器组 Rm 的内容相减, 结果送给 Rm。当运算结果为 0, 将影响零状态标志位 Z, 使 Z 置 1。否则为 0。当执行结果的低 4 位向高 4 位有借位(减指令), 影响辅助进位标志位 DC, 使 DC 置 0。当执行结果(8 位)向高位有借位时(减指令), 影响标志位 CY, 使 CY 置 0。否则为 1。

3.23 SUB dir, Rm 减法指令(不带进位)

语法: SUB dir, Rm

操作数: dir, Rm

操作: $dir \leftarrow (dir) - (Rm)$

受影响标志位: CY、DC、Z

说明: 将通用寄存器或特殊功能寄存器 dir 的内容与寄存器组 Rm 的内容相减, 结果送给 dir。当运算结果为 0, 将影响零状态标志位 Z, 使 Z 置 1。否则为 0。当执行结果的低 4 位向高 4 位有借位(减指令), 影响辅助进位标志位 DC, 使 DC 置 0。当执行结果(8 位)向高位有借位时(减指令), 影响标志位 CY, 使 CY 置 0。否则为 1。

3.24 SUB Rn, #data 减法指令(不带进位)

语法: SUB Rn, #data

操作数: Rn, data

操作: $Rn \leftarrow data - (Rn)$

受影响标志位: CY、DC、Z

说明: 将立即数 data 的内容与寄存器组 Rn 的内容相减, 结果送给 Rn。当运算结果为 0, 将影响零状态标志位 Z, 使 Z 置 1。否则为 0。当执行结果的低 4 位向高 4 位有借位(减指令), 影响辅助进位标志位 DC, 使 DC 置 0。当执行结果(8 位)向高位有借位时(减指令), 影响标志位 CY, 使 CY 置 0。否则为 1。

3.25 SUB Rn, Rs 加法指令（不带进位）

语法: SUB Rn, Rs

操作数: Rn, Rs

操作: $Rn \leftarrow (Rs) - (Rn)$

受影响标志位: CY、DC、Z

说明: 将寄存器组 Rs 的内容与寄存器组 Rn 的内容相减, 结果送给 Rn。当运算结果为 0, 将影响零状态标志位 Z, 使 Z 置 1。否则为 0。当执行结果的低 4 位向高 4 位有借位(减指令), 影响辅助进位标志位 DC, 使 DC 置 0。当执行结果(8 位)向高位有借位时(减指令), 影响标志位 CY, 使 CY 置 0。否则为 1。

3.26 CMP Rn, #data 比较指令

语法: CMP Rn, #data

操作数: Rn, #data

操作: -

影响标志位: CY、DC、Z

说明: 将立即数与 Rn 内容进行比较, 类似 SUB Rn, #data, 但 Rn 内容不改变。

3.27 CMP Rn, Rs 比较指令

语法: CMP Rn, Rs

操作数: Rn, Rs

操作: -

影响标志位: CY、DC、Z

说明: 将 Rs 与 Rn 内容进行比较, 类似 SUB Rn, Rs, 但 Rn 内容不改变。

3.28 INC dir 加 1 指令

语法: INC dir

操作数: dir

操作: $dir \leftarrow (dir) + 1$

影响标志位: Z

说明: 将寄存器 dir 的内容加 1, 若原单元内容为 FFH, 加 1 后溢出为 00H 时只影响 PSW 标志位 Z, 但是不影响 PSW 其他标志位。

3.29 INCR dir 加 1 指令: 寄存器 dir 内容加 1, 结果送给寄存器组 R0

语法: INCR dir

操作数: dir

操作: $R0 \leftarrow (dir) + 1$

影响标志位: Z

说明: 将寄存器 dir 的内容加 1, 结果送给寄存器组 R0 先保存起来, 一般用来保护寄存器 dir 内容的值。若原单元内容为 FFH, 加 1 后溢出为 00H 时只影响 PSW 标志位 Z, 但是不影响 PSW 其他标志位。

3.30 INC Rn 加 1 指令

语法: INC Rn

操作数: Rn

操作: $Rn \leftarrow (Rn) + 1$

影响标志位: Z

说明: 将寄存器组 Rn 的内容加 1, 若原单元内容为 FFH, 加 1 后溢出为 00H 时只影响 PSW 标志位 Z, 但是不影响 PSW 其他标志位。

3.31 DEC dir 减 1 指令

语法: DEC dir

操作数: dir

操作: $dir \leftarrow (dir) - 1$

影响标志位: Z

说明: 将寄存器 dir 的内容减 1, 若原单元内容为 00H, 减 1 后为 FFH 时只影响 PSW 标志位 Z, 但是不影响 PSW 其他标志位。

3.32 DECR dir 减 1 指令, 寄存器 dir 内容减 1, 结果送给寄存器组 R0

语法: DECR dir

操作数: dir

操作: $R0 \leftarrow (dir) - 1$

影响标志位: Z

说明: 将寄存器 dir 的内容减 1, 结果送给寄存器组 R0 先保存起来, 一般用来保护寄存器 dir 内容的值。若原单元内容为 00H, 减 1 后为 FFH 时只影响 PSW 标志位 Z, 但是不影响 PSW 其他标志位。

3.33 DEC Rn 减 1 指令

语法: DEC Rn

操作数: Rn

操作: $Rn \leftarrow (Rn) - 1$

影响标志位： Z

说明：将寄存器组 Rn 的内容减 1，若原单元内容为 00H，加 1 后为 FFH 时只影响 PSW 标志位 Z，但是不影响 PSW 其他标志位。

3.34 AND Rm, dir : 逻辑与指令

语法： AND Rm, dir

操作数： Rm, dir

操作： $Rm \leftarrow (Rm) \wedge (dir)$

影响标志位： Z

说明：将寄存器组 Rm 的内容与寄存器 dir 的内容进行逐位相与，结果送回寄存器组 Rm。当结果为 0 时，PSW 标志位 Z 置 1，否则为 0

3.35 AND dir, Rm 逻辑与指令

语法： AND dir, Rm

操作数： dir, Rm

操作： $dir \leftarrow (dir) \wedge (Rm)$

影响标志位： Z

说明：将寄存器 dir 的内容与寄存器组 Rm 的内容进行逐位相与，结果送回寄存器组 dir。当结果为 0 时，PSW 标志位 Z 置 1，否则为 0

3.36 AND Rn, #data 逻辑与指令

语法： AND Rn, #data

操作数： Rn, data

操作： $Rn \leftarrow (Rn) \wedge (data)$

影响标志位： Z

说明：将寄存器组 Rn 的内容与立即数进行逐位相与，结果送回寄存器组 Rn。当结果为 0 时，PSW 标志位 Z 置 1，否则为 0

3.37 AND Rn, Rs 逻辑与指令

语法： AND Rn, Rs

操作数： Rn, Rs

操作： $Rn \leftarrow (Rn) \wedge (Rs)$

影响标志位： Z

说明：将寄存器组 Rn 的内容与寄存器组 Rs 进行逐位相与，结果送回寄存器

组 Rn。当结果为 0 时，PSW 标志位 Z 置 1，否则为 0

3.38 ORL Rm, dir 逻辑或指令

语法: ORL Rm, dir
操作数: Rm, dir
操作: $Rm \leftarrow (Rm) \vee (dir)$
影响标志位: Z

说明: 将寄存器组 Rm 的内容与寄存器 dir 的内容进行逐位相或，结果送回寄存器组 Rm。当结果为 0 时，PSW 标志位 Z 置 1，否则为 0

3.39 ORL dir, Rm 逻辑或指令

语法: ORL dir, Rm
操作数: dir, Rm
操作: $dir \leftarrow (dir) \vee (Rm)$
影响标志位: Z

说明: 将寄存器 dir 的内容与寄存器组 Rm 的内容进行逐位相或，结果送回寄存器组 dir。当结果为 0 时，PSW 标志位 Z 置 1，否则为 0

3.40 ORL Rn, #data 逻辑或指令

语法: ORL Rn, #data
操作数: Rn, data
操作: $Rn \leftarrow (Rn) \vee (data)$
影响标志位: Z

说明: 将寄存器组 Rn 的内容与立即数进行逐位相或，结果送回寄存器组 Rn，当结果为 0 时，PSW 标志位 Z 置 1，否则为 0

3.41 ORL Rn, Rs 逻辑或指令

语法: ORL Rn, Rs
操作数: Rn, Rs
操作: $Rn \leftarrow (Rn) \vee (Rs)$
影响标志位: Z

说明: 将寄存器组 Rn 的内容与寄存器组 Rs 进行逐位相或，结果送回寄存器组 Rn。当结果为 0 时，PSW 标志位 Z 置 1，否则为 0

3.42 XOR Rm, dir 逻辑异或指令

语法: XOR Rm, dir

操作数: Rm, dir
操作: $Rm \leftarrow (Rm) \oplus (dir)$
影响标志位: Z
说明: 将寄存器组 Rm 的内容与寄存器 dir 的内容进行逐位相异或, 结果送回寄存器组 Rm 。当两者相同时, 结果为 0 时, PSW 标志位 Z 置 1, 否则为 0

3.43 XOR dir, Rm 逻辑异或指令

语法: $XOR\ dir, Rm$
操作数: dir, Rm
操作: $dir \leftarrow (dir) \oplus (Rm)$
影响标志位: Z
说明: 将寄存器 dir 的内容与寄存器组 Rm 的内容进行逐位相异或, 结果送回寄存器 dir 。当两者相同时, 结果为 0 时, PSW 标志位 Z 置 1, 否则为 0

3.44 XOR $Rn, \#data$ 逻辑异或指令

语法: $XOR\ Rn, \#data$
操作数: $Rn, data$
操作: $Rn \leftarrow (Rn) \oplus (data)$
影响标志位: Z
说明: 将寄存器组 Rn 的内容与立即数进行逐位相异或, 结果送回寄存器组 Rn 。当两者相同时, 结果为 0 时, PSW 标志位 Z 置 1, 否则为 0

3.45 XOR Rn, Rs 逻辑异或指令

语法: $XOR\ Rn, Rs$
操作数: Rn, Rs
操作: $Rn \leftarrow (Rn) \oplus (Rs)$
影响标志位: Z
说明: 将寄存器组 Rn 的内容与寄存器组 Rs 进行逐位相异或, 结果送回寄存器组 Rn 。当两者相同时, 结果为 0 时, PSW 标志位 Z 置 1, 否则为 0

3.46 CLR Rn 寄存器组清 0 指令

语法: $CLR\ Rn$
操作数: Rn
操作: $Rn = 0$
影响标志位: Z
说明: 将寄存器组 Rn 的内容清 0, 只影响标志位 Z , 使 Z 置 1

3.47 CLR dir 寄存器 dir 清 0 指令

语法: CLR dir
操作数: dir
操作: dir=0
影响标志位: Z
说明: 将寄存器组 dir 的内容清 0, 只影响标志位 Z, 使 Z 置 1

3.48 CPLR dir 寄存器 dir 取反指令, 结果送给寄存器 Rn

语法: CPLR dir
操作数: dir
操作: $R0 \leftarrow (\overline{dir})$
影响标志位: Z
说明: 将寄存器 dir 的内容按位取反, 结果存放在寄存器组 Rn 中。影响标志位 Z, 结果为 0 时使 Z 置 1

3.49 CPL dir 寄存器 dir 取反指令: 结果返回给寄存器 dir

语法: CPL dir
操作数: dir
操作: $dir \leftarrow (\overline{dir})$
影响标志位: Z
说明: 将寄存器 dir 的内容按位取反, 结果存放在寄存器 dir 中。影响标志位 Z, 结果为 0 时使 Z 置 1

3.50 CPL Rn

语法: CPL Rn
操作数: Rn
操作: $Rn \leftarrow (\overline{Rn})$
影响标志位: Z
说明: 将寄存器 Rn 的内容按位取反, 结果存放在寄存器 Rn 中。影响标志位 Z, 结果为 0 时使 Z 置 1

3.51 RRCR dir 带进位循环右移 1 位指令，结果存放在寄存器 Rn 中

语法：RRCR dir
操作数：dir

操作：



影响标志位：CY

说明：将寄存器 dir 的内容带进位循环右移 1 位指令，结果存放在寄存器组 Rn 中。影响标志位 CY

3.52 RRC dir 带进位循环右移 1 位指令

语法：RRC dir
操作数：dir

操作：



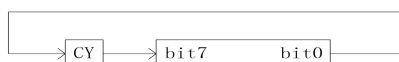
影响标志位：Z

说明：将寄存器 dir 的内容带进位循环右移 1 位指令，结果存放在寄存器 dir 中。影响标志位 CY。

3.53 RRC Rn 带进位循环右移 1 位指令

语法：RRC Rn
操作数：Rn

操作：

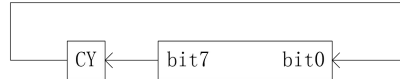


影响标志位：Z

说明：将寄存器 Rn 的内容带进位循环左移 1 位指令，结果存放在寄存器 Rn 中。影响标志位 CY

3.54 RLCR dir 带进位循环左移 1 位指令，结果存放在寄存器 Rn 中

语法：RLCR dir
操作数：dir



操作:

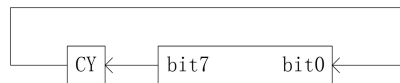
影响标志位: CY

说明: 将寄存器 dir 的内容带进位循环左移 1 位指令, 结果存放在寄存器组 Rn 中。影响标志位 CY

3.55 RLC dir 带进位循环左移 1 位指令

语法: RLC dir

操作数: dir



操作:

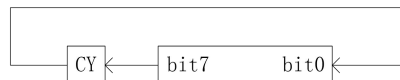
影响标志位: Z

说明: 将寄存器 dir 的内容带进位循环左移 1 位指令, 结果存放在寄存器 dir 中。影响标志位 CY

3.56 RLC Rn 带进位循环左移 1 位指令

语法: RLC Rn

操作数: Rn



操作:

影响标志位: Z

说明: 将寄存器 Rn 的内容带进位循环左移 1 位指令, 结果存放在寄存器 Rn 中。影响标志位 CY

3.57 CLR dir, b 清 0 位操作指令

语法: CLR dir, b

操作数: dir

b ($0 \leq b \leq 7$)

操作: 将 dir 的 b 位清 0

影响标志位: 无

说明: 此指令是将寄存器 dir 的 b 位 ($0 \leq b \leq 7$) 清 0, 不影响 PSW 标志位。单周期指令。

3.58 SET dir, b 置位操作指令

语法: SET dir , b
操作数: dir
b ($0 \leq b \leq 7$)
操作: 将 dir 的 b 位置 1
影响标志位: 无
说明: 此指令是将寄存器 dir 的 b 位 ($0 \leq b \leq 7$) 置 1, 不影响 PSW 标志位。
单周期指令。

3.59 CLR Rn, b 清 0 位操作指令

语法: CLR Rn, b
操作数: Rn
b ($0 \leq b \leq 7$)
操作: 将 Rn 的 b 位清 0
影响标志位: 无
说明: 此指令是将寄存器 Rn 的 b 位 ($0 \leq b \leq 7$) 清 0, 不影响 PSW 标志位。
单周期指令。

3.60 SET Rn, b 置位操作指令

语法: SET Rn , b
操作数: Rn
b ($0 \leq b \leq 7$)
操作: 将 Rn 的 b 位置 1
影响标志位: 无
说明: 此指令是将寄存器 Rn 的 b 位 ($0 \leq b \leq 7$) 置 1, 不影响 PSW 标志位。
单周期指令。

3.61 DECRJZ dir 减 1 跳转指令, 结果送给 Rn

语法: DECRJZ dir
操作数: dir
操作: $R0 \leftarrow (dir) - 1$, 为 0 跳过下一条指令
影响标志位: 无
说明: 此指令是将寄存器 dir 的内容减 1, 结果送给寄存器组 Rn。如果 Rn 内容为 0, 则跳过下一条指令, 如果不为 0, 则继续执行下一条指令

3.62 DECJZ dir 减 1 跳转指令

语法: DECJZ dir
操作数: dir
操作: $dir \leftarrow (dir) - 1$, 为 0 跳过下一条指令
影响标志位: 无
说明: 此指令是将寄存器 dir 的内容减 1, 结果送给寄存器 dir。如果 dir 内容为 0, 则跳过下一条指令, 如果不为 0, 则继续执行下一条指令

3.63 DECJZ Rn 减 1 跳转指令

语法: DECJZ Rn
操作数: Rn
操作: $Rn \leftarrow (Rn) - 1$, 为 0 跳过下一条指令
影响标志位: 无
说明: 此指令是将寄存器 Rn 的内容减 1, 结果送给寄存器 Rn。如果 Rn 内容为 0, 则跳过下一条指令, 如果不为 0, 则继续执行下一条指令。

3.64 INCRJZ dir 加 1 跳转指令, 结果送给 Rn

语法: INCRJZ dir
操作数: dir
操作: $R0 \leftarrow (dir) + 1$, 为 0 跳过下一条指令
影响标志位: 无
说明: 此指令是将寄存器 dir 的内容加 1, 结果送给寄存器组 Rn。如果 Rn 内容为 0, 则跳过下一条指令, 如果不为 0, 则继续执行下一条指令

3.65 INCJZ dir 加 1 跳转指令

语法: INCJZ dir
操作数: dir
操作: $dir \leftarrow (dir) + 1$, 为 0 跳过下一条指令
影响标志位: 无
说明: 此指令是将寄存器 dir 的内容加 1, 结果送给寄存器 dir。如果 dir 内容为 0, 则跳过下一条指令, 如果不为 0, 则继续执行下一条指令。

3.66 INCJZ Rn 加 1 跳转指令

语法: INCJZ Rn
操作数: Rn

操作: $Rn \leftarrow (Rn) + 1$, 为 0 跳过下一条指令
影响标志位: 无
说明: 此指令是将寄存器 Rn 的内容加 1, 结果送给寄存器 Rn 。如果 Rn 内容为 0, 则跳过下一条指令, 如果不为 0, 则继续执行下一条指令

3. 67 JNB dir, b 判位转移指令, 为 0 跳转

语法: JNB dir, b
操作数: dir
 b ($0 \leq b \leq 7$)
操作: dir 的 b 位为 0 跳过下一条指令
影响标志位: 无
说明: 此指令是判断转移指令, 判断 dir 的 b 位是否为 0, 如果为 0, 则跳过下一条指令, 如果不为 0, 则继续执行下一条指令。不影响 PSW 标志位

3. 68 JB dir, b 判位转移指令, 为 1 跳转

语法: JB dir, b
操作数: dir
 b ($0 \leq b \leq 7$)
操作: dir 的 b 位为 1 跳过下一条指令
影响标志位: 无
说明: 此指令是判断转移指令, 判断 dir 的 b 位是否为 1, 如果为 1, 则跳过下一条指令, 如果不为 1, 则继续执行下一条指令。不影响 PSW 标志位

3. 69 JNB Rn, b 判位转移指令, 为 0 跳转

语法: JNB Rn, b
操作数: Rn
 b ($0 \leq b \leq 7$)
操作: Rn 的 b 位为 0 跳过下一条指令
影响标志位: 无
说明: 此指令是判断转移指令, 判断 Rn 的 b 位是否为 0, 如果为 0, 则跳过下一条指令, 如果不为 0, 则继续执行下一条指令。不影响 PSW 标志位

3. 70 JB Rn, b 判位转移指令, 为 1 跳转

语法: JB Rn, b
操作数: Rn
 b ($0 \leq b \leq 7$)

操作：Rn 的 b 位为 1 跳过下一条指令
影响标志位：无
说明：此指令是判断转移指令，判断 Rn 的 b 位是否为 1，如果为 1，则跳过下一条指令，如果不为 1，则继续执行下一条指令。不影响 PSW 标志位

3.71 JMP #data12 无条件转移指令

语法：JMP #data12
操作数：data12，即 12 位立即数所指定的地址
操作：无
影响标志位：无
说明：此指令是无条件转移指令，提供 12 位的地址，可在 4K 字范围内无条件转移到我们 data12 所指的地址单元。双周期指令。如超过 4K 字空间型号，PC 内容需要组合 12bit 外 PCH 中的其余高位部分。

3.72 CALL #data12 子程序调用指令

语法：CALL #data12
操作数：data，即 12 位立即数所指定的地址
操作：无
影响标志位：无
说明：此程序是子程序调用指令。当 CPU 执行该句时，可直接寻址 4K 字范围内的子函数地址，CPU 自动把下一条指令第一字节的地址（PC 值，称为断点）压入堆栈，栈指针（SP）+2，并将子程序的起始地址送入 PC，于是 CPU 去执行子程序。当遇到 CRET 指令时，CPU 自动把断点的地址弹回到 PC 中，于是 CPU 又回到主程序继续往下执行。针对超过 4K 字空间型号，PC 内容需要组合 12bit 外 PCH 中的其余高位部分。