

ES7P系列单片机指令集

指令集简介

ES7P 系列单片机采用 79 条精简指令集系统。汇编指令为了方便程序设计者使用，指令名称大多是由指令功能的英文缩写所组成的。这些指令所组成的程序经过编译器的编译与链接后，会被转换为相对应的指令码。转换后的指令码可以分为操作码（OP Code）与操作数（Operand）两个部分。操作码部分对应到指令本身。

指令的字宽是 16 位，按照指令码的字数可将指令分为单字指令和双字指令。双字指令包括有 AJMP、LCALL，其他指令都为单字指令。

在 79 条指令集系统中，除了 NOP、NOP2 两条空操作指令不执行任何操作外，其余指令根据执行功能可分为三类：寄存器操作类指令、程序控制类指令、算术逻辑运算类指令。

寄存器操作类指令

指令		状态位	操作
1.	SECTIONI	-	$i<7:0> \rightarrow \text{BKSR}<7:0>$
2.	PAGE I	-	$i<4:0> \rightarrow \text{PCRH}<7:3>$
3.	ISTEP I	-	$\text{IAA} + i \rightarrow \text{IAA} (-128 \leq i \leq 127)$
4.	MOV R,F	Z,N	(R)→(目标)
5.	MOVA R	-	(A)→(R)
6.	MOVAR R	-	(A)→(R)
7.	MOVI I	-	$i \rightarrow (A)$
8.	MOVRA R	-	(R)→(A)

SECTION

- 指令功能：通用数据存储体选择指令。
 指令格式：SECTION i
 操作数：i 为 8 位立即数，即为所选存储体的编号。
 执行时间：1 个指令周期。
 执行过程： $i<7:0> \rightarrow \text{BKSR}<7:0>$

影响标志位: 无
指令说明: 该指令将立即数 i 置入 **BKSR** 寄存器, 若 i 的值大于单片机实际的最大存储体编号, 则默认选择 **SECTION0**。

PAGE

指令功能: 程序存储器选页指令。
指令格式: **PAGE** i
操作数: i 为 5 位立即数, 即为所选程序存储器页的编号。
执行时间: 1 个指令周期。
执行过程: $i<4:0>\rightarrow\text{PCRH}<7:3>$
影响标志位: 无
指令说明: 该指令将立即数 i 置入 **PCRH<7:3>**, 常用于 **GOTO**、**CALL** 指令之前, 进行程序存储器页的切换。

ISTEP

指令功能: 修改间接寻址地址寄存器 **IAA** 的指令。
指令格式: **ISTEP** i
操作数: i 为 8 位有符号立即数, 表示地址的偏移量。
执行时间: 1 个指令周期。
执行过程: $\text{IAA}+i\rightarrow\text{IAA}(-128\leq i\leq 127)$
影响标志位: 无
指令说明: 该指令将立即数 i 与 **IAA** 相加的结果置入 **IAA**, 其中 i 是有符号数。

MOV

指令功能: 把数据寄存器 **R** 的值传送到目的寄存器。
指令格式: **MOV** **R, F**
操作数: **R** 为数据寄存器, **F** 为运算结果方向位 (0~1)。关于 **F** 方向位的说明参见 **JINC** 指令。
执行时间: 1 个指令周期。
执行过程: **(R)**→(目标)
影响标志位: **Z, N**
指令说明: 如果传送的值为 0, 则置 **Z** 标志位。如果传送的值为负数, 则置 **N** 标志位。

MOVA

指令功能: 把 **A** 寄存器的值传送到数据寄存器 **R**。
指令格式: **MOVA** **R**

操作数:	R 为数据寄存器。
执行时间:	1 个指令周期。
执行过程:	(A)→(R)
影响标志位:	无
指令说明:	该指令常用于对数据寄存器进行赋值, 并且不影响任何标志位。

MOVI

指令功能:	把立即数 i 置入 A 寄存器。
指令格式:	MOVI i
操作数:	i 为 8 位立即数。
执行时间:	1 个指令周期。
执行过程:	i<7:0>→(A)
影响标志位:	无
指令说明:	该指令常用于对 A 寄存器进行赋值, 并且不影响任何标志位。

MOVAR

指令功能:	把 A 寄存器的值传送到通用数据寄存器 R。
指令格式:	MOVAR R
操作数:	R 为数据寄存器, 其最大寻址范围为 2K 字。
执行时间:	1 个指令周期。
执行过程:	(A)→(R)
影响标志位:	无
指令说明:	该指令常用于对数据寄存器进行赋值, 并且不影响任何标志位。与 MOV A 指令不同, MOVAR 指令的最大寻址范围为 2K 字, 可对通用数据存储器区进行全空间寻址。因此, MOVAR 指令并不依赖于 BKSR 寄存器的当前状态。但是, MOVAR 指令不能访问特殊寄存器区。

MOVRA

指令功能:	把通用数据寄存器 R 的值传送到 A 寄存器。
指令格式:	MOVRA R
操作数:	R 为数据寄存器, 其最大寻址范围为 2K 字。
执行时间:	1 个指令周期。
执行过程:	(R)→(A)
影响标志位:	无
指令说明:	该指令常用于获取数据寄存器的值, 并且不影响任何标志位。MOVRA 指令的最大寻址范围为 2K 字, 可对数据存储器进行全空间寻址。因此, MOVRA 指令并不依赖于 BKSR 寄存器的当前状态。但是, MOVRA 指

令不能访问特殊寄存器区。

程序控制类指令

指令			状态位	操作
1.	JUMP	I	-	PC+1+i→PC (-128≤i≤127)
2.	AJMP	I	-	I<15:0>→PC<15:0>, i<15:8>→PCRH<7:0>
3.	GOTO	I	-	i→PC<10:0>, PCRH<7:3>→PC<15:11>
4.	CALL	I	-	PC+1→TOS, i→PC<10:0>, PCRH<7:3>→PC<15:11>
5.	LCALL	I	-	PC+2→TOS, i→PC<15:0>, i<15:8>→PCRH<7:0>
6.	RCALL	R	-	PC+1→TOS, (R)→PC<7:0>, PCRH<7:0>→PC<15:8>
7.	RET		-	TOS→PC
8.	RETIA	I	-	i→(A), TOS→PC
9.	RETIE		-	TOS→PC, 1→GIE
10.	CWDT		N_TO,N_PD	00 _H →WDT, 0→WDT 预分频计数器, 1→N_TO, 1→N_PD
11.	RST		ALL	软件复位
12.	IDLE		N_TO,N_PD	00 _H →WDT, 0→WDT 预分频计数器 1→N_TO, 0→N_PD
13.	JBC	R,B	-	如果 R = 0 则跳过下一条指令
14.	JBS	R,B	-	如果 R = 1 则跳过下一条指令
15.	JINC	R,F	-	(R+1) →(目标), 如果(目标) = 0 则跳过下一条指令
16.	JDEC	R,F	-	(R-1) →(目标), 如果(目标) = 0 则跳过下一条指令
17.	JCAIE	I	-	如果(A) = I 则跳过下一条指令
18.	JCAIG	I	-	如果(A) > I 则跳过下一条指令
19.	JCAIL	I	-	如果(A) < I 则跳过下一条指令
20.	JCARE	R	-	如果(R) = (A)则跳过下一条指令
21.	JCARG	R	-	如果(R) > (A)则跳过下一条指令
22.	JCARL	R	-	如果(R) < (A)则跳过下一条指令

指令			状态位	操作
23.	JCCRE	B,R	-	如果 $C = R(B)$ 则跳过下一条指令
24.	JCCRG	B,R	-	如果 $C > R(B)$ 则跳过下一条指令
25.	JCCRL	B,R	-	如果 $C < R(B)$ 则跳过下一条指令
26.	POP		-	AS->A, BKSRS->BKSR PSWS->PSW, PCRHS->PCRH
27.	PUSH		-	A->AS, BKSR->BKSRS, PSW->PSWS, PCRH->PCRHS

JUMP

- 指令功能：程序相对短跳转。
- 指令格式：**JUMP** i
- 操作数：i 为 8 位有符号立即数，指定的跳转目标地址为 $PC + i$ 。
- 执行时间：2 个指令周期。
- 执行过程： $PC + 1 + i < 7:0 > \rightarrow PC (-128 \leq i \leq 127)$
- 影响标志位：无
- 指令说明：与 GOTO 指令的绝对跳转不同，JUMP 是相对跳转。如果 i 为负数，则表示向前跳转 i 个指令地址。如果 i 为正数，则表示向后跳转 i 个指令地址。

AJMP

- 指令功能：程序绝对长跳转。
- 指令格式：**AJMP** i
- 操作数：i 为 16 位立即数，指定的跳转目标地址。
- 执行时间：2 个指令周期。
- 执行过程： $i < 15:0 > \rightarrow PC < 15:0 >$, $i < 15:8 > \rightarrow PCRH < 7:0 >$
- 影响标志位：无
- 指令说明：程序无条件跳转到目标地址处继续执行。与 GOTO 指令不同，AJMP 支持程序存储器全空间跳转。

GOTO

- 指令功能：程序绝对跳转。
- 指令格式：**GOTO** i
- 操作数：i 为 11 位立即数，指定的跳转目标地址。
- 执行时间：2 个指令周期。
- 执行过程： $i < 10:0 > \rightarrow PC < 10:0 >$, $PCRH < 7:3 > \rightarrow PC < 15:11 >$

影响标志位： 无
指令说明： 程序无条件跳转到目的地址处继续执行。目标地址由本指令所含的 11 位地址码和 PCRH 特殊寄存器的位<5:3>一起构成完整的 14 位地址。

CALL

指令功能： 调用子程序。
指令格式： **CALL** i
操作数： i 为 11 位立即数，指定的子程序入口地址。
执行时间： 2 个指令周期。
执行过程： **PC+1**→TOS, i→PC<10:0>, PCRH<7:3>→PC<15:11>
影响标志位： 无
指令说明： 首先，将指向下一条指令的 PC 值压入硬件堆栈，然后，跳到目标子程序入口处继续执行。子程序的入口地址由本指令所含的 11 位地址码和 PCRH 特殊寄存器的位<5:3>一起构成完整的 14 位地址。

LCALL

指令功能： 调用子程序（即全空间范围寻址）。
指令格式： **LCALL** i
操作数： i 为 14 位立即数，指定的子程序入口地址。
执行时间： 2 个指令周期。
执行过程： **PC+1**→TOS, i→PC<13:0>, i<15:8>→PCRH<7:0>
影响标志位： 无
指令说明： 首先，将指向下一条指令的 PC 值压入硬件堆栈，然后，跳到目标子程序入口处继续执行。与 CALL 指令不同，LCALL 指令是全空间范围寻址的，因此，不依赖于 PCRH 寄存器的原始状态。

RCALL

指令功能： 间接调用子程序。
指令格式： **RCALL** R
操作数： R 为数据寄存器。
执行时间： 2 个指令周期。
执行过程： **PC+1**→TOS, (R)→PC<7:0>, PCRH<7:0>→PC<15:8>
影响标志位： 无
指令说明： 首先，将指向下一条指令的 PC 值压入硬件堆栈，然后，跳到子程序入口处继续执行。子程序的入口地址由本指令数据寄存器 R 的值和 PCRH 特殊寄存器的位<7:0>一起构成完整的 16 位地址。

RET

指令功能:	子程序结束返回。
指令格式:	RET
操作数:	无
执行时间:	2 个指令周期。
执行过程:	TOS→PC
影响标志位:	无
指令说明:	该指令把硬件堆栈栈顶的值弹出送到 PC 内。程序将回到子程序调用时的下一条指令处继续执行。注意，该指令并不改变特殊寄存器 PCRH 的内容。

RETIA

指令功能:	子程序结束返回，并将立即数 i 置入 A 寄存器。
指令格式:	RETIA i
操作数:	i 为 8 位立即数，返回时将 i 置入 A 寄存器。
执行时间:	2 个指令周期。
执行过程:	I→(A), TOS→PC
影响标志位:	无
指令说明:	该指令与 RET 指令类似，差异在于该指令中的立即数 i 将被自动置入 A 寄存器。

指令范例:

1	MOV	Index, A	
2	CALL	TBL_8LED	;调用查表子程序 TBL_8LED。
3	TBL_8LED:		
4	ADD	PCRL	;修改 PCRL 的值实现查表。
5	RETIA	0x00	;子程序返回，并将 0 置入 A 寄存器。
6	RETIA	0x01	;子程序返回，并将 1 置入 A 寄存器。

RETIE

指令功能:	中断服务程序结束返回。
指令格式:	RETIE
操作数:	无
执行时间:	2 个指令周期。
执行过程:	TOS→PC, 1→GIE
影响标志位:	无
指令说明:	该指令与 RET 指令类似，差异在于该指令会自动将中断控制寄存器

INTCO 中的第 7 位 GIE（全局中断使能位）置 1。

CWDT

指令功能：清除看门狗计数器。
指令格式：CWDT
操作数：无
执行时间：1 个指令周期。
执行过程：00_H→WDT，0→WDT 预分频计数器，1→N_TO，1→N_PD
影响标志位：N_TO，N_PD
指令说明：如果单片机在烧写配置字时，使能了 WDT 片上看门狗电路，那么，程序在运行时必须每隔一段时间执行一次该条清看门狗指令。如果时间没有执行这条指令，单片机将自动复位。

指令范例：

```
1 MainLoop:
2           CWDT           ;清看门狗。
3           CALL    KeyProcess ;按键处理。
4           CALL    DspProcess ;显示刷新。
5           GOTO    MainLoop   ;重复主循环。
```

RST

指令功能：软件复位。
指令格式：RST
操作数：无
执行时间：1 个指令周期。
执行过程：软件复位
影响标志位：全部状态位
指令说明：该指令使单片机软件复位。

IDLE

指令功能：进入低功耗休眠模式。
指令格式：IDLE
操作数：无
执行时间：1 个指令周期。
执行过程：00_H→WDT，0→WDT 预分频计数器，1→N_TO，0→N_PD
影响标志位：N_TO，N_PD
指令说明：该指令使单片机停止所有工作，并进入低功耗休眠模式。此时，芯片自

身的功耗降至最低。在休眠状态下，主时钟振荡器停振，所有内部寄存器值保持不变，直到满足唤醒条件时，单片机被唤醒。

指令范例:

1	CALL	PreIdle	;关闭不需要输出的负载，准备休眠。
2	IDLE		;进入休眠模式。
3	NOP		;唤醒后开始执行的第一条指令。

JBC

指令功能: 如果数据寄存器 R 中的第 B 位为 0，则跳过下一条指令。
 指令格式: JBC R, B
 操作数: R 为数据寄存器，B 为数据位编号 (0~7)。
 执行时间: 当满足条件跳转时，2 个指令周期，否则为 1 个指令周期。
 执行过程: 如果 R = 0，则跳过下一条指令。
 影响标志位: 无
 指令说明: 如果 R 寄存器的第 B 位为 0，则跳过下一条指令。常用于根据标志位实现程序分支跳转的控制。

指令范例:

1	JBC	PSW, Z	;判断状态寄存器的 z 标志位。
2	GOTO	Zero	;如果 Z=1，跳转到 Zero 标号继续执行。
3	NOP		;如果 z=0，继续执行此处指令。

JBS

指令功能: 如果数据寄存器 R 中的第 B 位为 1，则跳过下一条指令。
 指令格式: JBS R, B
 操作数: R 为数据寄存器，B 为数据位编号 (0~7)。
 执行时间: 当满足条件跳转时，2 个指令周期，否则为 1 个指令周期。
 执行过程: 如果 R = 1，则跳过下一条指令。
 影响标志位: 无
 指令说明: 如果 R 寄存器的第 B 位为 1，则跳过下一条指令。常用于根据标志位实现程序分支跳转的控制。

JINC

指令功能: 数据寄存器 R 加 1，如果运算结果为 0，则跳过下一条指令。
 指令格式: JINC R, F
 操作数: R 为数据寄存器，F 为运算结果方向位 (0~1)。如果 F=0，运算结果存

储到 A 寄存器中，R 寄存器的值不变；如果 F=1，运算结果存储到 R 寄存器内，A 寄存器的值不变。

执行时间：当满足条件跳转时，2 个指令周期，否则为 1 个指令周期。
执行过程： $(R) + 1 \rightarrow (\text{目标})$ ，如果 $(\text{目标}) = 0$ ，则跳过下一条指令。
影响标志位：无
指令说明：首先，该指令对数据寄存器 R 的值进行加 1 运算，运算结果存到 F 指定的目标寄存器。然后，根据状态寄存器的 Z 标志位进行程序分支跳转控制。该指令常用于计数次数或循环次数控制。

JDEC

指令功能：数据寄存器 R 减 1，如果运算结果为 0，则跳过下一条指令。
指令格式：JDEC R, F
操作数：R 为数据寄存器，F 为运算结果方向位 (0~1)。关于 F 方向位的说明参见 JINC 指令。
执行时间：当满足条件跳转时，2 个指令周期，否则为 1 个指令周期。
执行过程： $(R) - 1 \rightarrow (\text{目标})$ ，如果 $(\text{目标}) = 0$ ，则跳过下一条指令。
影响标志位：无
指令说明：首先，该指令对数据寄存器 R 的值进行减 1 运算，运算结果存到 F 指定的目标寄存器。然后，根据状态寄存器的 Z 标志位进行程序分支跳转控制。该指令常用于计数次数或循环次数控制。

JCAIE

指令功能：如果 A 寄存器的值与立即数 i 相等，则跳过下一条指令。
指令格式：JCAIE i
操作数：i 为 8 位立即数。
执行时间：当满足条件跳转时，2 个指令周期，否则为 1 个指令周期。
执行过程：如果 $(A) = i$ ，则跳过下一条指令。
影响标志位：无
指令说明：该指令常用于比较 A 寄存器的值与立即数是否相等。

JCAIG

指令功能：如果 A 寄存器的值大于立即数 i，则跳过下一条指令。
指令格式：JCAIG i
操作数：i 为 8 位立即数。
执行时间：当满足条件跳转时，2 个指令周期，否则为 1 个指令周期。
执行过程：如果 $(A) > i$ ，则跳过下一条指令。
影响标志位：无

指令说明: 该指令常用于比较 A 寄存器的值是否大于立即数。

JCAIL

指令功能: 如果 A 寄存器的值小于立即数 i, 则跳过下一条指令。
指令格式: JCAIL i
操作数: i 为 8 位立即数。
执行时间: 当满足条件跳转时, 2 个指令周期, 否则为 1 个指令周期。
执行过程: 如果(A)<i, 则跳过下一条指令。
影响标志位: 无
指令说明: 该指令常用于比较 A 寄存器的值是否小于立即数。

JCRAE

指令功能: 如果数据寄存器 R 的值与 A 寄存器的值相等, 则跳过下一条指令。
指令格式: JCRAE R
操作数: R 为数据寄存器。
执行时间: 当满足条件跳转时, 2 个指令周期, 否则为 1 个指令周期。
执行过程: 如果(R)=(A), 则跳过下一条指令。
影响标志位: 无
指令说明: 该指令常用于比较 A 寄存器的值与数据寄存器 R 的值是否相等。

JCRAG

指令功能: 如果数据寄存器 R 的值大于 A 寄存器的值, 则跳过下一条指令。
指令格式: JCRAG R
操作数: R 为数据寄存器。
执行时间: 当满足条件跳转时, 2 个指令周期, 否则为 1 个指令周期。
执行过程: 如果(R)>(A), 则跳过下一条指令。
影响标志位: 无
指令说明: 该指令常用于比较数据寄存器 R 的值是否大于 A 寄存器的值。

JCRAL

指令功能: 如果数据寄存器 R 的值小于 A 寄存器的值, 则跳过下一条指令。
指令格式: JCRAL R
操作数: R 为数据寄存器。
执行时间: 当满足条件跳转时, 2 个指令周期, 否则为 1 个指令周期。
执行过程: 如果(R)<(A), 则跳过下一条指令。
影响标志位: 无
指令说明: 该指令常用于比较数据寄存器 R 的值是否小于 A 寄存器的值。

JCCRE

指令功能:	如果 C 标志位的值与数据寄存器 R 的指定位的值相等, 则跳过下一条指令。
指令格式:	JCCRE R, B
操作数:	R 为数据寄存器。
执行时间:	当满足条件跳转时, 2 个指令周期, 否则为 1 个指令周期。
执行过程:	如果 $C=R$, 则跳过下一条指令。
影响标志位:	无
指令说明:	该指令常用于比较数据寄存器 R 的第 B 位的值是否与 C 标志位相等。

JCCRG

指令功能:	如果 C 标志位的值大于数据寄存器的指定位的值, 则跳过下一条指令。
指令格式:	JCCRG R, B
操作数:	R 为数据寄存器。
执行时间:	当满足条件跳转时, 2 个指令周期, 否则为 1 个指令周期。
执行过程:	如果 $C>R$, 则跳过下一条指令。
影响标志位:	无
指令说明:	该指令常用于比较 C 标志位是否大于数据寄存器 R 的第 B 位的值。

JCCRL

指令功能:	如果 C 标志位的值小于数据寄存器的指定位的值, 则跳过下一条指令。
指令格式:	JCCRL R, B
操作数:	R 为数据寄存器。
执行时间:	当满足条件跳转时, 2 个指令周期, 否则为 1 个指令周期。
执行过程:	如果 $C<R$, 则跳过下一条指令。
影响标志位:	无
指令说明:	该指令常用于比较 C 标志位是否小于数据寄存器 R 的第 B 位的值。

PUSH

指令功能:	保护 A、BKSR、PSW、PCRH 寄存器的值。
指令格式:	PUSH
操作数:	无
执行时间:	1 个指令周期。
执行过程:	A→AS, BKSR→BKSRs, PSW→PSWS, PCRH→PCRHS
影响标志位:	无
指令说明:	保护 A、BKSR、PSW、PCRH 寄存器的值。在中断入口处, 常用于保

护现场。

POP

指令功能:	恢复 A、BKSR、PSW、PCRH 寄存器的值。
指令格式:	POP
操作数:	无
执行时间:	1 个指令周期。
执行过程:	AS→A, BKSR→BKSR, PSWS→PSW, PCRHS→PCRH
影响标志位:	无
指令说明:	恢复 A、BKSR、PSW、PCRH 寄存器的值，在中断出口处，常用于恢复现场。

算术逻辑运算类指令

指令		状态位	操作
1.	ADD R,F	C,DC,Z,OV,N	(R)+(A) →(目标)
2.	ADDC R,F	C,DC,Z,OV,N	(R)+(A)+C →(目标)
3.	ADDI I	C,DC,Z,OV,N	i+(A)→(A)
4.	ADDCI I	C,DC,Z,OV,N	i+(A)+C→(A)
5.	SUB R,F	C,DC,Z,OV,N	(R)-(A) →(目标)
6.	SSUB R,F	C,DC,Z,OV,N	(A)-(R) →(目标)
7.	SUBC R,F	C,DC,Z,OV,N	(R)-(A)-(#C) →(目标)
8.	SSUBC R,F	C,DC,Z,OV,N	(A)-(R)-(#C) →(目标)
9.	SUBI I	C,DC,Z,OV,N	i-(A) →(A)
10.	SSUBI I	C,DC,Z,OV,N	(A)-i →(A)
11.	SUBCI I	C,DC,Z,OV,N	i-(A)-(#C) →(A)
12.	SSUBCI I	C,DC,Z,OV,N	(A)-i-(#C) →(A)
13.	AND R,F	Z,N	(A) AND (R) →(目标)
14.	ANDI I	Z,N	I AND (A) →(A)
15.	IOR R,F	Z,N	(A) OR (R) →(目标)
16.	IORI I	Z,N	i OR (A) →(A)
17.	XOR R,F	Z,N	(A) XOR (R) →(目标)
18.	XORI I	Z,N	i XOR (A) →(A)
19.	BSS R,B		1→R(B)
20.	BCC R,B		0→R(B)
21.	BTT R,B		(~R(B)) →R(B)

指令			状态位	操作
22.	CLR	R	Z	(R)=0
23.	SETR	R		(FFh) →(R)
24.	COM	R,F	Z,N	(~R) →(目标)
25.	NEG	R	C,DC,Z,OV,N	(~R)+1→(R)
26.	DAR	R,F	C	对(R)十进制调整 →(目标)
27.	DAA		C	对(A)十进制调整 →(目标)
28.	INC	R,F	C,DC,Z,OV,N	(R)+1 →(目标)
29.	DEC	R,F	C,DC,Z,OV,N	(R)-1 →(目标)
30.	RLB	R,F,B	C,Z,N	$C \ll R \langle 7:0 \rangle \ll C$
31.	RLBNC	R,F,B	Z,N	$R \langle 7:0 \rangle \ll R \langle 7 \rangle$
32.	RRB	R,F,B	C,Z,N	$C \gg R \langle 7:0 \rangle \gg C$
33.	RRBNC	R,F,B	Z,N	$R \langle 0 \rangle \gg R \langle 7:0 \rangle$
34.	SWAP	R,F		$R \langle 3:0 \rangle \rightarrow (\text{目标}) \langle 7:4 \rangle, R \langle 7:4 \rangle \rightarrow (\text{目标}) \langle 3:0 \rangle$
35.	TBR			Pmem(FRA) →ROMD
36.	TBR#1			Pmem(FRA) →ROMD,(FRA)+1→(FRA)
37.	TBR_1			Pmem(FRA) → ROMD,(FRA)-1→(FRA)
38.	TBR1#			(FRA) +1→(FRA), Pmem(FRA) →ROMD
39.	TBW			ROMD→编程缓冲器
40.	TBW#1			ROMD→编程缓冲器,(FRA)+1→(FRA)
41.	TBW_1			ROMD→编程缓冲器,(FRA)-1→(FRA)
42.	TBW1#			Pmem(FRA) →ROMD

ADD

指令功能： A 寄存器的值与数据寄存器 R 的值的 不带进位加法运算，结果传送到目标寄存器。

指令格式： ADD R, F

操作数： R 为数据寄存器，F 为运算结果方向位（0~1）。关于 F 方向位的说明参见 JINC 指令。

执行时间： 1 个指令周期。

执行过程: (R)+(A)→(目标)
 影响标志位: C, DC, Z, OV, N
 指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了进位, 则置 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。
 指令范例:

1	MOVI	0x55	; 将立即数 0x55 传送到 A 寄存器。
2	ADD	Sum	; 将 Sum 寄存器值与 A 寄存器的值相加。

ADDC

指令功能: A 寄存器的值与数据寄存器 R 的值的带进位加法运算, 结果传送到目标寄存器。
 指令格式: **ADDC R, F**
 操作数: R 为数据寄存器, F 为运算结果方向位 (0~1)。关于 F 方向位的说明参见 JINC 指令。
 执行时间: 1 个指令周期。
 执行过程: (R)+(A)+C→(目标)
 影响标志位: C, DC, Z, OV, N
 指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了进位, 则置 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。

ADDI

指令功能: A 寄存器的值与立即数 i 的不带进位加法运算, 结果传送到 A 寄存器。
 指令格式: **ADDI i**
 操作数: i 为 8 位立即数。
 执行时间: 1 个指令周期。
 执行过程: **i+(A)→(A)**
 影响标志位: C, DC, Z, OV, N
 指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了进位, 则置 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。

ADDCI

指令功能:	A 寄存器的值与立即数 i 的带进位加法运算, 结果传送到 A 寄存器。
指令格式:	ADDCI i
操作数:	i 为 8 位立即数。
执行时间:	1 个指令周期。
执行过程:	$i+(A)+C \rightarrow (A)$
影响标志位:	C, DC, Z, OV, N
指令说明:	如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了进位, 则置 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。

SUB

指令功能:	数据寄存器 R 的值与 A 寄存器的值的不带借位减法运算, 结果传送到目标寄存器。
指令格式:	SUB R, F
操作数:	R 为数据寄存器, F 为运算结果方向位 (0~1)。关于 F 方向位的说明参见 JINC 指令。
执行时间:	1 个指令周期。
执行过程:	$(R)-(A) \rightarrow (\text{目标})$
影响标志位:	C, DC, Z, OV, N
指令说明:	如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了借位, 则清 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。

指令范例:

1	MOVI 0x55 ; 将立即数 0x55 传送到 A 寄存器
2	SUB Sum ; (Sum) = (Sum) - (A)

SSUB

指令功能:	A 寄存器的值与数据寄存器 R 的值的不带借位减法运算, 结果传送到目标寄存器。
指令格式:	SSUB R, F
操作数:	R 为数据寄存器, F 为运算结果方向位 (0~1)。关于 F 方向位的说明参见 JINC 指令。

执行时间: 1 个指令周期。
执行过程: (A)-(R)→(目标)
影响标志位: C, DC, Z, OV, N
指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了借位, 则清 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。

SUBC

指令功能: 数据寄存器 R 的值与 A 寄存器的值的带借位减法运算, 结果传送到目标寄存器。
指令格式: SUBC R, F
操作数: R 为数据寄存器, F 为运算结果方向位 (0~1)。关于 F 方向位的说明参见 JINC 指令。
执行时间: 1 个指令周期。
执行过程: (R)-(A)-(~C)→(目标)
影响标志位: C, DC, Z, OV, N
指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了借位, 则清 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。

SSUBC

指令功能: A 寄存器的值与数据寄存器 R 的值的带借位减法运算, 结果传送到目标寄存器。
指令格式: SSUBC R, F
操作数: R 为数据寄存器, F 为运算结果方向位 (0~1)。关于 F 方向位的说明参见 JINC 指令。
执行时间: 1 个指令周期。
执行过程: (A)-(R)-(~C)→(目标)
影响标志位: C, DC, Z, OV, N
指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了借位, 则清 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。

SUBI

- 指令功能: 立即数 i 与 A 寄存器的值的不带借位减法运算, 结果传送到 A 寄存器。
- 指令格式: **SUBI** i
- 操作数: i 为 8 位立即数。
- 执行时间: 1 个指令周期。
- 执行过程: $i-(A)\rightarrow(A)$
- 影响标志位: **C, DC, Z, OV, N**
- 指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了借位, 则清 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。

SSUBI

- 指令功能: A 寄存器的值与立即数 i 的不带借位减法运算, 结果传送到 A 寄存器。
- 指令格式: **SSUBI** i
- 操作数: i 为 8 位立即数。
- 执行时间: 1 个指令周期。
- 执行过程: $(A)-i\rightarrow(A)$
- 影响标志位: **C, DC, Z, OV, N**
- 指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了借位, 则清 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。

SUBCI

- 指令功能: 立即数 i 与 A 寄存器的值的带借位减法运算, 结果传送到 A 寄存器。
- 指令格式: **SUBCI** i
- 操作数: i 为 8 位立即数。
- 执行时间: 1 个指令周期。
- 执行过程: $i-(A)-(\sim C)\rightarrow(A)$
- 影响标志位: **C, DC, Z, OV, N**
- 指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了借位, 则清 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。

SSUBCI

指令功能:	A 寄存器的值与立即数 i 的带借位减法运算, 结果传送到 A 寄存器。
指令格式:	SSUBCI i
操作数:	i 为 8 位立即数。
执行时间:	1 个指令周期。
执行过程:	(A)-i-(~C)→(A)
影响标志位:	C, DC, Z, OV, N
指令说明:	如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了借位, 则清 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。

AND

指令功能:	A 寄存器的值与数据寄存器 R 的值的位与运算, 结果传送到目标寄存器。
指令格式:	AND R, F
操作数:	R 为数据寄存器, F 为运算结果方向位 (0~1)。关于 F 方向位的说明参见 JINC 指令。
执行时间:	1 个指令周期。
执行过程:	(A) AND (R)→(目标)
影响标志位:	Z, N
指令说明:	如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。

ANDI

指令功能:	A 寄存器的值与立即数 i 的位与运算, 结果传送到 A 寄存器。
指令格式:	ANDI i
操作数:	i 为 8 位立即数。
执行时间:	1 个指令周期。
执行过程:	i AND (A)→(A)
影响标志位:	Z, N
指令说明:	如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。

IOR

指令功能:	A 寄存器的值与数据寄存器 R 的值的位或运算, 结果传送到目标寄存器。
指令格式:	IOR R, F
操作数:	R 为数据寄存器, F 为运算结果方向位 (0~1)。关于 F 方向位的说明参

见 JINC 指令。
执行时间: 1 个指令周期。
执行过程: (A) OR (R)→(目标)
影响标志位: Z, N
指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。

IORI

指令功能: A 寄存器的值与立即数 i 的位或运算, 结果传送到 A 寄存器。
指令格式: IORI i
操作数: i 为 8 位立即数。
执行时间: 1 个指令周期。
执行过程: i OR (A)→(A)
影响标志位: Z, N
指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。

XOR

指令功能: A 寄存器的值与数据寄存器 R 的值的位异或运算, 结果传送到目标寄存器。
指令格式: XOR R, F
操作数: R 为数据寄存器, F 为运算结果方向位 (0~1)。关于 F 方向位的说明参见 JINC 指令。
执行时间: 1 个指令周期。
执行过程: (A) XOR (R)→(目标)
影响标志位: Z, N
指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。

XORI

指令功能: A 寄存器的值与立即数 i 的位异或运算, 结果传送到 A 寄存器。
指令格式: XORI i
操作数: i 为 8 位立即数。
执行时间: 1 个指令周期。
执行过程: i XOR (A)→(A)
影响标志位: Z, N
指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。

位。

BSS

指令功能：将数据寄存器 R 的第 B 位置为 1。
指令格式：BSS R, B
操作数：R 为数据寄存器，B 为数据位编号（0~7）。
执行时间：1 个指令周期。
执行过程： $1 \rightarrow R\langle B \rangle$
影响标志位：无
指令说明：该指令对任何数据寄存器的某一位置 1，常用于标志位设置或将某输出引脚置高。

BCC

指令功能：将数据寄存器 R 的第 B 位置为 0。
指令格式：BCC R, B
操作数：R 为数据寄存器，B 为数据位编号（0~7）。
执行时间：1 个指令周期。
执行过程： $0 \rightarrow R\langle B \rangle$
影响标志位：无
指令说明：该指令对任何数据寄存器的某一位置 0，常用于标志位设置或将某输出引脚置低。

BTT

指令功能：将数据寄存器 R 的第 B 位置取反。
指令格式：BTT R, B
操作数：R 为数据寄存器，B 为数据位编号（0~7）。
执行时间：1 个指令周期。
执行过程： $(\sim R\langle B \rangle) \rightarrow R\langle B \rangle$
影响标志位：无
指令说明：该指令对任何数据寄存器中的某一位取反。

CLR

指令功能：将数据寄存器 R 清 0。
指令格式：CLR R
操作数：R 为数据寄存器。
执行时间：1 个指令周期。
执行过程： $0 \rightarrow (R)$

影响标志位: Z
指令说明: 该指令执行完后, 将置 Z 标志位。

SETR

指令功能: 将数据寄存器 R 置为 0xFF。
指令格式: SETR R
操作数: R 为数据寄存器。
执行时间: 1 个指令周期。
执行过程: 0xFF→(R)
影响标志位: 无
指令说明: 对数据寄存器 R 各数据位置 1。

COM

指令功能: 对数据寄存器 R 取反运算, 结果传送到目标寄存器。
指令格式: COM R, F
操作数: R 为数据寄存器, F 为运算结果方向位 (0~1)。关于 F 方向位的说明参见 JINC 指令。
执行时间: 1 个指令周期。
执行过程: (~R)→(目标)
影响标志位: Z, N
指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。

NEG

指令功能: 对数据寄存器 R 取补运算。
指令格式: NEG R
操作数: R 为数据寄存器。
执行时间: 1 个指令周期。
执行过程: ~(R)+1→(R)
影响标志位: C, DC, Z, OV, N
指令说明: 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了进位, 则置 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。

DAR

指令功能: 数据寄存器 R 的值的十进制调整运算, 结果传送到目标寄存器。

指令格式:	DAR R, F
操作数:	R 为数据寄存器, F 为运算结果方向位 (0~1)。关于 F 方向位的说明参见 JINC 指令。
执行时间:	1 个指令周期。
执行过程:	对(R)十进制调整→(目标)
影响标志位:	C
指令说明:	该指令的直接前趋指令必须是一条算术运算指令, 否则该指令无效。
指令范例:	

1	MOVI	0x0A	
2	MOVA	Tmp	; 将 0x0A 置入 Tmp。
3	MOVI	0x12	
4	ADD	Tmp	; 将计算结果 0x0C 传送到 Tmp。
5	DAR	Tmp	; 对 Tmp 进行十进制调整, 结果为 0x22。

DAA

指令功能:	A 寄存器的值的十进制调整运算, 结果传送 A 寄存器。
指令格式:	DAA
操作数:	无
执行时间:	1 个指令周期。
执行过程:	对(A)十进制调整→(A)
影响标志位:	C
指令说明:	该指令的直接前趋指令必须是一条算术运算指令, 否则该指令无效。

INC

指令功能:	数据寄存器 R 的值加 1, 结果传送到目标寄存器。
指令格式:	INC R, F
操作数:	R 为数据寄存器, F 为运算结果方向位 (0~1)。关于 F 方向位的说明参见 JINC 指令。
执行时间:	1 个指令周期。
执行过程:	(R)+1→(目标)
影响标志位:	C, DC, Z, OV, N
指令说明:	该指令对数据寄存器 R 的值进行加 1 运算, 结果存到 F 指定的目标寄存器。值得注意的是, JINC 指令不影响标志位, 而 INC 指令是影响标志位的。 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了进位, 则置 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志

位。

DEC

指令功能:	数据寄存器 R 的值减 1, 结果传送到目标寄存器。
指令格式:	DEC R, F
操作数:	R 为数据寄存器, F 为运算结果方向位 (0~1)。关于 F 方向位的说明参见 JINC 指令。
执行时间:	1 个指令周期。
执行过程:	(R)-1→(目标)
影响标志位:	C, DC, Z, OV, N
指令说明:	该指令对数据寄存器 R 的值进行减 1 运算, 结果存到 F 指定的目标寄存器。 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了进位, 则置 C 标志位。如果运算结果半字节产生了进位, 则置 DC 标志位。如果运算结果产生了溢出, 则置 OV 标志位。

RLB

指令功能:	数据寄存器 R 的值带 C 标志位的左移 B 位运算, 结果传送到目标寄存器。
指令格式:	RLB R, F, B
操作数:	R 为数据寄存器, F 为运算结果方向位 (0~1), B 为移位次数 (0~7)。关于 F 方向位的说明参见 JINC 指令。
执行时间:	1 个指令周期。
执行过程:	$C \ll R \langle 7:0 \rangle \ll C$
影响标志位:	C, Z, N
指令说明:	该指令对由数据寄存器 R 的值及 C 标志位组成的 9 位数据进行左移 B 位运算, 运算结果存到 F 指定的目标寄存器。 如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了进位, 则置 C 标志位。

RLBNC

指令功能:	数据寄存器 R 的值不带 C 标志位的左移 B 位运算, 结果传送到目标寄存器。
指令格式:	RLBNC R, F, B
操作数:	R 为数据寄存器, F 为运算结果方向位 (0~1), B 为移位次数 (0~7)。关于 F 方向位的说明参见 JINC 指令。

执行时间: 1 个指令周期。
执行过程: $R\langle 7:0 \rangle \ll R\langle 7 \rangle$
影响标志位: Z, N
指令说明: 该指令对数据寄存器 R 的值进行左移 B 位运算, 运算结果存到 F 指定的目标寄存器。
如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。

RRB

指令功能: 数据寄存器 R 的值带 C 标志位的右移 B 位运算, 结果传送到目标寄存器。
指令格式: $RRB \quad R, F, B$
操作数: R 为数据寄存器, F 为运算结果方向位 (0~1), B 为移位次数 (0~7)。
关于 F 方向位的说明参见 JINC 指令。
执行时间: 1 个指令周期。
执行过程: $C \gg R\langle 7:0 \rangle \gg C$
影响标志位: C, Z, N
指令说明: 该指令对由数据寄存器 R 的值及 C 标志位组成的 9 位数据进行右移 B 位运算, 运算结果存到 F 指定的目标寄存器。
如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。如果运算结果产生了进位, 则置 C 标志位。

RRBNC

指令功能: 数据寄存器 R 的值不带 C 标志位的右移 B 位运算, 结果传送到目标寄存器。
指令格式: $RRBNC \quad R, F, B$
操作数: R 为数据寄存器, F 为运算结果方向位 (0~1), B 为移位次数 (0~7)。
关于 F 方向位的说明参见 JINC 指令。
执行时间: 1 个指令周期。
执行过程: $R\langle 7:0 \rangle \gg R\langle 7 \rangle$
影响标志位: Z, N
指令说明: 该指令对数据寄存器 R 的值进行右移 B 位运算, 结果存到 F 指定的目标寄存器。
如果运算结果为 0, 则置 Z 标志位。如果运算结果为负数, 则置 N 标志位。

SWAP

- 指令功能: 数据寄存器 R 的值的高、低半字节交换, 结果传送到目标寄存器。
- 指令格式: SWAP R, F
- 操作数: R 为数据寄存器, F 为运算结果方向位 (0~1)。关于 F 方向位的说明参见 JINC 指令。
- 执行时间: 1 个指令周期。
- 执行过程: $R<3:0>\rightarrow(\text{目标})<7:4>$, $R<7:4>\rightarrow(\text{目标})<3:0>$
- 影响标志位: 无
- 指令说明: 该指令对数据寄存器 R 的值进行高、低半字节交换, 结果存到 F 指定的目标寄存器。

TBR

- 指令功能: 将 FRA 所指向的程序存储器的值传送到 ROMD 寄存器中。
- 指令格式: TBR
- 操作数: 无
- 执行时间: 1 个指令周期。
- 执行过程: $Pmem(FRA)\rightarrow ROMD$
- 影响标志位: 无
- 指令说明: 该指令将 FRA 查表指针寄存器组 (FRAL、FRAH 寄存器) 指向的程序存储器的值传送到 ROMDH 和 ROMDL 寄存器。

TBR#1

- 指令功能: 将 FRA 所指向的程序存储器的值传送到 ROMD 寄存器中, 再将 FRA 的值自增 1。
- 指令格式: TBR#1
- 操作数: 无
- 执行时间: 1 个指令周期。
- 执行过程: $Pmem(FRA)\rightarrow ROMD$, $(FRA)+1\rightarrow(FRA)$
- 影响标志位: 无
- 指令说明: 该指令将 FRA 查表指针寄存器组 (FRAL、FRAH 寄存器) 指向的程序存储器的值传送到 ROMDH 和 ROMDL 寄存器, 再将 FRA 的值自增 1。

TBR_1

- 指令功能: 将 FRA 所指向的程序存储器的值传送到 ROMD 寄存器中, 再将 FRA 的值自减 1。

指令格式:	TBR_1
操作数:	无
执行时间:	1 个指令周期。
执行过程:	Pmem(FRA)→ROMD, (FRA)-1→(FRA)
影响标志位:	无
指令说明:	该指令将 FRA 查表指针寄存器组 (FRAL、FRAH 寄存器) 指向的程序存储器的值传送到 ROMDH 和 ROMDL 寄存器中, 再将 FRA 的值自减 1。

TBR1#

指令功能:	FRA 的值自增 1, 再将 FRA 所指向的程序存储器的值传送到 ROMD 寄存器中。
指令格式:	TBR1#
操作数:	无
执行时间:	1 个指令周期。
执行过程:	(FRA)+1→(FRA) , Pmem(FRA)→ROMD
影响标志位:	无
指令说明:	FRA 的值自增 1, 再将 FRA 查表指针寄存器组 (FRAL、FRAH 寄存器) 指向的程序存储器的值传送到 ROMDH 和 ROMDL 寄存器中。

TBW

指令功能:	将 ROMD 寄存器中的值传送到 FRA 所指向的程序存储器中。
指令格式:	TBW
操作数:	无
执行时间:	1 个指令周期。
执行过程:	ROMD→Pmem(FRA)
影响标志位:	无
指令说明:	该指令 ROMDH 和 ROMDL 寄存器的值传送到 FRA 查表指针寄存器组 (FRAL、FRAH 寄存器) 指向的程序存储器中。

TBW#1

指令功能:	将 ROMD 寄存器中的值传送到 FRA 所指向的程序存储器中, 再将 FRA 的值自增 1。
指令格式:	TBW#1
操作数:	无

执行时间: 1 个指令周期。
执行过程: ROMD→Pmem(FRA), (FRA)+1→(FRA)
影响标志位: 无
指令说明: 该指令 ROMDH 和 ROMDL 寄存器的值传送到 FRA 查表指针寄存器组 (FRAL、FRAH 寄存器) 指向的程序存储器中, 再将 FRA 的值自增 1。

TBW_1

指令功能: 将 ROMD 寄存器中的值传送到 FRA 所指向的程序存储器中, 再将 FRA 的值自减 1。
指令格式: TBW_1
操作数: 无
执行时间: 1 个指令周期。
执行过程: ROMD→Pmem(FRA), (FRA)-1→(FRA)
影响标志位: 无
指令说明: 该指令 ROMDH 和 ROMDL 寄存器的值传送到 FRA 查表指针寄存器组 (FRAL、FRAH 寄存器) 指向的程序存储器中, 再将 FRA 的值自减 1。

TBW1#

指令功能: FRA 的值自增 1, 再将 ROMD 寄存器中的值传送到 FRA 所指向的程序存储器中。
指令格式: TBW1#
操作数: 无
执行时间: 1 个指令周期。
执行过程: (FRA)+1→(FRA), ROMD→Pmem(FRA)
影响标志位: 无
指令说明: FRA 的值自减 1, 再将 ROMDH 和 ROMDL 寄存器的值传送到 FRA 查表指针寄存器组 (FRAL、FRAH 寄存器) 指向的程序存储器中。