

指令集

本章以表格方式列出了 78K0/KC2 的指令集。如需了解每种指令的操作和操作代码，可参阅 **78K/0 系列指令用户手册（U12326E）**。

1 操作列表使用规则

1.1 操作数标识符和标识方法

根据规范确定的指令操作数标识方法（详情可参见汇编程序编程规范），在每种指令的“操作数”栏列出操作数。如果有两种或两种以上的标识方法，可选其中之一。大写字母和符号#、!、\$ 和 []是关键字，必须按其原样书写。每种符号的含义如下所示。

- #: 立即数标识
- !: 绝对地址标识
- \$: 相对地址标识
- [:间接地址标识

立即数用来描述一个数值型数据或标号。当使用标号时，注意必须加上符号#、!、\$、和 []。

对应操作数寄存器标识符 **r** 和 **rp**，功能名称（**X**、**A**、**C**，等）或绝对名称（下表括号中的名称：**R0**、**R1**、**R2** 等）都可用于标识。

表1-1. 操作数标识符和标识方法

标识符	标识方法
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) 特殊功能寄存器符号 [‡] 特殊功能寄存器符号（仅用于 16 位可操作寄存器偶地址） [‡]
saddr saddrp	FE20H ~ FF1FH 立即数或标号 FE20H ~ FF1FH 立即数或标号（仅用于偶地址）
addr16 addr11 addr5	0000H ~ FFFFH 立即数或标号 （仅用于 16 位数据传送指令的偶地址） 0800H ~ 0FFFH 立即数或标号 0040H ~ 007FH 立即数或标号（仅用于偶地址）
word byte bit	16 位立即数或标号 8 位立即数或标号 3 位立即数或标号
RBn	RB0 ~ RB3

注 不能使用这些操作数访问地址 FFD0H ~ FFD7H。

备注 特殊功能寄存器符号参见表 3-7 特殊功能寄存器列表。

1.2 操作栏描述

A:	A 寄存器; 8 位累加器
X:	X 寄存器
B:	B 寄存器
C:	C 寄存器
D:	D 寄存器
E:	E 寄存器
H:	H 寄存器
L:	L 寄存器
AX:	AX 寄存器对; 16 位累加器
BC:	BC 寄存器对
DE:	DE 寄存器对
HL:	HL 寄存器对
PC:	程序计数器
SP:	堆栈指针
PSW:	程序状态字
CY:	进位标志
AC:	辅助进位标志
Z:	零标志
RBS:	寄存器 bank 选择标志
IE:	中断请求允许标志
():	括号中的地址或寄存器所指的存储单元的内容
X _H , X _L :	16 位寄存器的高 8 位和低 8 位
^:	逻辑与 (AND)
∨:	逻辑或 (OR)
⊕:	逻辑异或 (exclusive OR)
—:	数据取反
addr16:	16 位立即数或标号
jdisp8:	带符号的 8 位数据 (偏移量)

1.3 标志操作栏的描述

(空):	不受影响
0:	清零
1:	设置为 1
×	根据结果进行设置/清零
R:	恢复先前保存的值

2 操作列表

指令组	助记符	操作数	字节	时钟		操作	标志
				注 1	注 2		Z AC CY
8 位数据传送	MOV	r, #byte	2	4	—	$r \leftarrow \text{byte}$	
		saddr, #byte	3	6	7	$(\text{saddr}) \leftarrow \text{byte}$	
		sfr, #byte	3	—	7	$\text{sfr} \leftarrow \text{byte}$	
		A, r ^{注 3}	1	2	—	$A \leftarrow r$	
		r, A ^{注 3}	1	2	—	$r \leftarrow A$	
		A, saddr	2	4	5	$A \leftarrow (\text{saddr})$	
		saddr, A	2	4	5	$(\text{saddr}) \leftarrow A$	
		A, sfr	2	—	5	$A \leftarrow \text{sfr}$	
		sfr, A	2	—	5	$\text{sfr} \leftarrow A$	
		A, !addr16	3	8	9	$A \leftarrow (\text{addr16})$	
		!addr16, A	3	8	9	$(\text{addr16}) \leftarrow A$	
		PSW, #byte	3	—	7	$\text{PSW} \leftarrow \text{byte}$	× × ×
		A, PSW	2	—	5	$A \leftarrow \text{PSW}$	
		PSW, A	2	—	5	$\text{PSW} \leftarrow A$	× × ×
		A, [DE]	1	4	5	$A \leftarrow (\text{DE})$	
		[DE], A	1	4	5	$(\text{DE}) \leftarrow A$	
		A, [HL]	1	4	5	$A \leftarrow (\text{HL})$	
		[HL], A	1	4	5	$(\text{HL}) \leftarrow A$	
		A, [HL + byte]	2	8	9	$A \leftarrow (\text{HL} + \text{byte})$	
		[HL + byte], A	2	8	9	$(\text{HL} + \text{byte}) \leftarrow A$	
		A, [HL + B]	1	6	7	$A \leftarrow (\text{HL} + \text{B})$	
		[HL + B], A	1	6	7	$(\text{HL} + \text{B}) \leftarrow A$	
		A, [HL + C]	1	6	7	$A \leftarrow (\text{HL} + \text{C})$	
		[HL + C], A	1	6	7	$(\text{HL} + \text{C}) \leftarrow A$	
	XCH	A, r ^{注 3}	1	2	—	$A \leftrightarrow r$	
		A, saddr	2	4	6	$A \leftrightarrow (\text{saddr})$	
		A, sfr	2	—	6	$A \leftrightarrow (\text{sfr})$	
		A, !addr16	3	8	10	$A \leftrightarrow (\text{addr16})$	
		A, [DE]	1	4	6	$A \leftrightarrow (\text{DE})$	
		A, [HL]	1	4	6	$A \leftrightarrow (\text{HL})$	
		A, [HL + byte]	2	8	10	$A \leftrightarrow (\text{HL} + \text{byte})$	
		A, [HL + B]	2	8	10	$A \leftrightarrow (\text{HL} + \text{B})$	
		A, [HL + C]	2	8	10	$A \leftrightarrow (\text{HL} + \text{C})$	

- 注
1. 当访问内部高速 RAM 时或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时
 3. “r = A” 除外

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（f_{cpu}）的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作数	字节	时钟		操作	标志
				注 1	注 2		Z AC CY
16 位数据 传送	MOVW	rp, #word	3	6	—	rp ← word	
		saddrp, #word	4	8	10	(saddrp) ← word	
		sfrp, #word	4	—	10	sfrp ← word	
		AX, saddrp	2	6	8	AX ← (saddrp)	
		saddrp, AX	2	6	8	(saddrp) ← AX	
		AX, sfrp	2	—	8	AX ← sfrp	
		sfrp, AX	2	—	8	sfrp ← AX	
		AX, rp 注 3	1	4	—	AX ← rp	
		rp, AX 注 3	1	4	—	rp ← AX	
		AX, !addr16	3	10	12	AX ← (addr16)	
		!addr16, AX	3	10	12	(addr16) ← AX	
	XCHW	AX, rp 注 3	1	4	—	AX ↔ rp	
8 位操作	ADD	A, #byte	2	4	—	A, CY ← A + byte	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	× × ×
		A, r 注 4	2	4	—	A, CY ← A + r	× × ×
		r, A	2	4	—	r, CY ← r + A	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr)	× × ×
		A, !addr16	3	8	9	A, CY ← A + (addr16)	× × ×
		A, [HL]	1	4	5	A, CY ← A + (HL)	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	× × ×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C)	× × ×
	ADDC	A, #byte	2	4	—	A, CY ← A + byte + CY	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	× × ×
		A, r 注 4	2	4	—	A, CY ← A + r + CY	× × ×
		r, A	2	4	—	r, CY ← r + A + CY	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	× × ×
		A, !addr16	3	8	9	A, CY ← A + (addr16) + C	× × ×
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	× × ×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	× × ×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C) + CY	× × ×

- 注
1. 当访问内部高速 RAM 时或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时
 3. 仅当 rp = BC, DE 或 HL
 4. “r = A” 除外

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{cpu}) 的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	SUB	A, #byte	2	4	–	$A, CY \leftarrow A - \text{byte}$	×	×	×
		saddr, #byte	3	6	8	$(saddr), CY \leftarrow (saddr) - \text{byte}$	×	×	×
		A, r ^{注 3}	2	4	–	$A, CY \leftarrow A - r$	×	×	×
		r, A	2	4	–	$r, CY \leftarrow r - A$	×	×	×
		A, saddr	2	4	5	$A, CY \leftarrow A - (saddr)$	×	×	×
		A, !addr16	3	8	9	$A, CY \leftarrow A - (\text{addr16})$	×	×	×
		A, [HL]	1	4	5	$A, CY \leftarrow A - (\text{HL})$	×	×	×
		A, [HL + byte]	2	8	9	$A, CY \leftarrow A - (\text{HL} + \text{byte})$	×	×	×
		A, [HL + B]	2	8	9	$A, CY \leftarrow A - (\text{HL} + B)$	×	×	×
		A, [HL + C]	2	8	9	$A, CY \leftarrow A - (\text{HL} + C)$	×	×	×
	SUBC	A, #byte	2	4	–	$A, CY \leftarrow A - \text{byte} - CY$	×	×	×
		saddr, #byte	3	6	8	$(saddr), CY \leftarrow (saddr) - \text{byte} - CY$	×	×	×
		A, r ^{注 3}	2	4	–	$A, CY \leftarrow A - r - CY$	×	×	×
		r, A	2	4	–	$r, CY \leftarrow r - A - CY$	×	×	×
		A, saddr	2	4	5	$A, CY \leftarrow A - (saddr) - CY$	×	×	×
		A, !addr16	3	8	9	$A, CY \leftarrow A - (\text{addr16}) - CY$	×	×	×
		A, [HL]	1	4	5	$A, CY \leftarrow A - (\text{HL}) - CY$	×	×	×
		A, [HL + byte]	2	8	9	$A, CY \leftarrow A - (\text{HL} + \text{byte}) - CY$	×	×	×
		A, [HL + B]	2	8	9	$A, CY \leftarrow A - (\text{HL} + B) - CY$	×	×	×
		A, [HL + C]	2	8	9	$A, CY \leftarrow A - (\text{HL} + C) - CY$	×	×	×
	AND	A, #byte	2	4	–	$A \leftarrow A \wedge \text{byte}$	×		
		saddr, #byte	3	6	8	$(saddr) \leftarrow (saddr) \wedge \text{byte}$	×		
		A, r ^{注 3}	2	4	–	$A \leftarrow A \wedge r$	×		
		r, A	2	4	–	$r \leftarrow r \wedge A$	×		
		A, saddr	2	4	5	$A \leftarrow A \wedge (saddr)$	×		
		A, !addr16	3	8	9	$A \leftarrow A \wedge (\text{addr16})$	×		
		A, [HL]	1	4	5	$A \leftarrow A \wedge (\text{HL})$	×		
		A, [HL + byte]	2	8	9	$A \leftarrow A \wedge (\text{HL} + \text{byte})$	×		
		A, [HL + B]	2	8	9	$A \leftarrow A \wedge (\text{HL} + B)$	×		
		A, [HL + C]	2	8	9	$A \leftarrow A \wedge (\text{HL} + C)$	×		

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时
 3. “r = A” 除外

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{cpu}) 的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作数	字节数	时钟		操作	标志
				注 1	注 2		Z AC CY
8 位操作	OR	A, #byte	2	4	—	$A \leftarrow A \vee \text{byte}$	×
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	×
		A, r ^{注 3}	2	4	—	$A \leftarrow A \vee r$	×
		r, A	2	4	—	$r \leftarrow r \vee A$	×
		A, saddr	2	4	5	$A \leftarrow A \vee (\text{saddr})$	×
		A, !addr16	3	8	9	$A \leftarrow A \vee (\text{addr16})$	×
		A, [HL]	1	4	5	$A \leftarrow A \vee (\text{HL})$	×
		A, [HL + byte]	2	8	9	$A \leftarrow A \vee (\text{HL} + \text{byte})$	×
		A, [HL + B]	2	8	9	$A \leftarrow A \vee (\text{HL} + B)$	×
		A, [HL + C]	2	8	9	$A \leftarrow A \vee (\text{HL} + C)$	×
	XOR	A, #byte	2	4	—	$A \leftarrow A \nabla \text{byte}$	×
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \nabla \text{byte}$	×
		A, r ^{注 3}	2	4	—	$A \leftarrow A \nabla r$	×
		r, A	2	4	—	$r \leftarrow r \nabla A$	×
		A, saddr	2	4	5	$A \leftarrow A \nabla (\text{saddr})$	×
		A, !addr16	3	8	9	$A \leftarrow A \nabla (\text{addr16})$	×
		A, [HL]	1	4	5	$A \leftarrow A \nabla (\text{HL})$	×
		A, [HL + byte]	2	8	9	$A \leftarrow A \nabla (\text{HL} + \text{byte})$	×
		A, [HL + B]	2	8	9	$A \leftarrow A \nabla (\text{HL} + B)$	×
		A, [HL + C]	2	8	9	$A \leftarrow A \nabla (\text{HL} + C)$	×
	CMP	A, #byte	2	4	—	$A - \text{byte}$	× × ×
		saddr, #byte	3	6	8	$(\text{saddr}) - \text{byte}$	× × ×
		A, r ^{注 3}	2	4	—	$A - r$	× × ×
		r, A	2	4	—	$r - A$	× × ×
		A, saddr	2	4	5	$A - (\text{saddr})$	× × ×
		A, !addr16	3	8	9	$A - (\text{addr16})$	× × ×
		A, [HL]	1	4	5	$A - (\text{HL})$	× × ×
		A, [HL + byte]	2	8	9	$A - (\text{HL} + \text{byte})$	× × ×
		A, [HL + B]	2	8	9	$A - (\text{HL} + B)$	× × ×
		A, [HL + C]	2	8	9	$A - (\text{HL} + C)$	× × ×

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时
 3. “r = A” 除外

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（f_{cpu}）的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位操作	ADDW	AX, #word	3	6	—	$AX, CY \leftarrow AX + word$	×	×	×
	SUBW	AX, #word	3	6	—	$AX, CY \leftarrow AX - word$	×	×	×
	CMPW	AX, #word	3	6	—	$AX - word$	×	×	×
乘法/ 除法	MULU	X	2	16	—	$AX \leftarrow A \times X$			
	DIVUW	C	2	25	—	$AX \text{ (商)}, C \text{ (余数)} \leftarrow AX \div C$			
递增/ 递减	INC	r	1	2	—	$r \leftarrow r + 1$	×	×	
		saddr	2	4	6	$(saddr) \leftarrow (saddr) + 1$	×	×	
	DEC	r	1	2	—	$r \leftarrow r - 1$	×	×	
		saddr	2	4	6	$(saddr) \leftarrow (saddr) - 1$	×	×	
	INCW	rp	1	4	—	$rp \leftarrow rp + 1$			
	DECW	rp	1	4	—	$rp \leftarrow rp - 1$			
循环	ROR	A, 1	1	2	—	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1 \text{ time}$			×
	ROL	A, 1	1	2	—	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1 \text{ time}$			×
	RORC	A, 1	1	2	—	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1 \text{ time}$			×
	ROLC	A, 1	1	2	—	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1 \text{ time}$			×
	ROR4	[HL]	2	10	12	$A_{3-0} \leftarrow (HL)_{3-0}, (HL)_{7-4} \leftarrow A_{3-0},$ $(HL)_{3-0} \leftarrow (HL)_{7-4}$			
	ROL4	[HL]	2	10	12	$A_{3-0} \leftarrow (HL)_{7-4}, (HL)_{3-0} \leftarrow A_{3-0},$ $(HL)_{7-4} \leftarrow (HL)_{3-0}$			
BCD 调整	ADJBA		2	4	—	加法后的十进制调整	×	×	×
	ADJBS		2	4	—	减法后的十进制调整	×	×	×
位操作	MOV1	CY, saddr.bit	3	6	7	$CY \leftarrow (saddr.bit)$			×
		CY, sfr.bit	3	—	7	$CY \leftarrow sfr.bit$			×
		CY, A.bit	2	4	—	$CY \leftarrow A.bit$			×
		CY, PSW.bit	3	—	7	$CY \leftarrow PSW.bit$			×
		CY, [HL].bit	2	6	7	$CY \leftarrow (HL).bit$			×
		saddr.bit, CY	3	6	8	$(saddr.bit) \leftarrow CY$			
		sfr.bit, CY	3	—	8	$sfr.bit \leftarrow CY$			
		A.bit, CY	2	4	—	$A.bit \leftarrow CY$			
		PSW.bit, CY	3	—	8	$PSW.bit \leftarrow CY$	×	×	
		[HL].bit, CY	2	6	8	$(HL).bit \leftarrow CY$			

- 注**
1. 当访问内部高速 RAM 或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时

- 备注**
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{cpu}) 的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作数	字节	时钟		操作	标志
				注 1	注 2		Z AC CY
位操作	AND1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \wedge (saddr.bit)$	×
		CY, sfr.bit	3	—	7	$CY \leftarrow CY \wedge sfr.bit$	×
		CY, A.bit	2	4	—	$CY \leftarrow CY \wedge A.bit$	×
		CY, PSW.bit	3	—	7	$CY \leftarrow CY \wedge PSW.bit$	×
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \wedge (HL).bit$	×
	OR1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \vee (saddr.bit)$	×
		CY, sfr.bit	3	—	7	$CY \leftarrow CY \vee sfr.bit$	×
		CY, A.bit	2	4	—	$CY \leftarrow CY \vee A.bit$	×
		CY, PSW.bit	3	—	7	$CY \leftarrow CY \vee PSW.bit$	×
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \vee (HL).bit$	×
	XOR1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \oplus (saddr.bit)$	×
		CY, sfr.bit	3	—	7	$CY \leftarrow CY \oplus sfr.bit$	×
		CY, A.bit	2	4	—	$CY \leftarrow CY \oplus A.bit$	×
		CY, PSW. bit	3	—	7	$CY \leftarrow CY \oplus PSW.bit$	×
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \oplus (HL).bit$	×
	SET1	saddr.bit	2	4	6	$(saddr.bit) \leftarrow 1$	
		sfr.bit	3	—	8	$sfr.bit \leftarrow 1$	
		A.bit	2	4	—	$A.bit \leftarrow 1$	
		PSW.bit	2	—	6	$PSW.bit \leftarrow 1$	× × ×
		[HL].bit	2	6	8	$(HL).bit \leftarrow 1$	
	CLR1	saddr.bit	2	4	6	$(saddr.bit) \leftarrow 0$	
		sfr.bit	3	—	8	$sfr.bit \leftarrow 0$	
		A.bit	2	4	—	$A.bit \leftarrow 0$	
		PSW.bit	2	—	6	$PSW.bit \leftarrow 0$	× × ×
		[HL].bit	2	6	8	$(HL).bit \leftarrow 0$	
	SET1	CY	1	2	—	$CY \leftarrow 1$	1
	CLR1	CY	1	2	—	$CY \leftarrow 0$	0
	NOT1	CY	1	2	—	$CY \leftarrow \overline{CY}$	×

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（f_{cpu}）的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作数	字节数	时钟		操作	标志
				注 1	注 2		Z AC CY
调用/返回	CALL	!addr16	3	7	—	$(SP - 1) \leftarrow (PC + 3)_H, (SP - 2) \leftarrow (PC + 3)_L,$ $PC \leftarrow \text{addr16}, SP \leftarrow SP - 2$	
	CALLF	!addr11	2	5	—	$(SP - 1) \leftarrow (PC + 2)_H, (SP - 2) \leftarrow (PC + 2)_L,$ $PC_{15-11} \leftarrow 00001, PC_{10-0} \leftarrow \text{addr11},$ $SP \leftarrow SP - 2$	
	CALLT	[addr5]	1	6	—	$(SP - 1) \leftarrow (PC + 1)_H, (SP - 2) \leftarrow (PC + 1)_L,$ $PC_H \leftarrow (00000000, \text{addr5} + 1),$ $PC_L \leftarrow (00000000, \text{addr5}),$ $SP \leftarrow SP - 2$	
	BRK		1	6	—	$(SP - 1) \leftarrow PSW, (SP - 2) \leftarrow (PC + 1)_H,$ $(SP - 3) \leftarrow (PC + 1)_L, PC_H \leftarrow (003FH),$ $PC_L \leftarrow (003EH), SP \leftarrow SP - 3, IE \leftarrow 0$	
	RET		1	6	—	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $SP \leftarrow SP + 2$	
	RETI		1	6	—	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $PSW \leftarrow (SP + 2), SP \leftarrow SP + 3$	R R R
	RETB		1	6	—	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $PSW \leftarrow (SP + 2), SP \leftarrow SP + 3$	R R R
堆栈操作	PUSH	PSW	1	2	—	$(SP - 1) \leftarrow PSW, SP \leftarrow SP - 1$	
		rp	1	4	—	$(SP - 1) \leftarrow rp_H, (SP - 2) \leftarrow rp_L,$ $SP \leftarrow SP - 2$	
	POP	PSW	1	2	—	$PSW \leftarrow (SP), SP \leftarrow SP + 1$	R R R
		rp	1	4	—	$rp_H \leftarrow (SP + 1), rp_L \leftarrow (SP),$ $SP \leftarrow SP + 2$	
	MOVW	SP, #word	4	—	10	$SP \leftarrow \text{word}$	
		SP, AX	2	—	8	$SP \leftarrow AX$	
		AX, SP	2	—	8	$AX \leftarrow SP$	
无条件转移	BR	!addr16	3	6	—	$PC \leftarrow \text{addr16}$	
		\$addr16	2	6	—	$PC \leftarrow PC + 2 + \text{jdisp8}$	
		AX	2	8	—	$PCH \leftarrow A, PCL \leftarrow X$	
条件转移	BC	\$addr16	2	6	—	如果 $CY = 1$, 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	
	BNC	\$addr16	2	6	—	如果 $CY = 0$, 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	
	BZ	\$addr16	2	6	—	如果 $Z = 1$, 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	
	BNZ	\$addr16	2	6	—	如果 $Z = 0$, 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	

- 注**
1. 当访问内部高速 RAM 或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时

- 备注**
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{cpu}) 的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作数	字节数	时钟		操作	标志
				注 1	注 2		Z AC CY
条件转移	BT	saddr.bit, \$addr16	3	8	9	如果(saddr.bit) = 1, 则 $PC \leftarrow PC + 3 + jdisp8$	
		sfr.bit, \$addr16	4	–	11	如果 sfr.bit = 1, 则 $PC \leftarrow PC + 4 + jdisp8$	
		A.bit, \$addr16	3	8	–	如果 A.bit = 1, 则 $PC \leftarrow PC + 3 + jdisp8$	
		PSW.bit, \$addr16	3	–	9	如果 PSW.bit = 1, 则 $PC \leftarrow PC + 3 + jdisp8$	
		[HL].bit, \$addr16	3	10	11	如果(HL).bit = 1, 则 $PC \leftarrow PC + 3 + jdisp8$	
	BF	saddr.bit, \$addr16	4	10	11	如果(saddr.bit) = 0, 则 $PC \leftarrow PC + 4 + jdisp8$	
		sfr.bit, \$addr16	4	–	11	如果 sfr.bit = 0, 则 $PC \leftarrow PC + 4 + jdisp8$	
		A.bit, \$addr16	3	8	–	如果 A.bit = 0, 则 $PC \leftarrow PC + 3 + jdisp8$	
		PSW.bit, \$addr16	4	–	11	如果 PSW.bit = 0, 则 $PC \leftarrow PC + 4 + jdisp8$	
		[HL].bit, \$addr16	3	10	11	如果 (HL).bit = 0, 则 $PC \leftarrow PC + 3 + jdisp8$	
	BTCLR	saddr.bit, \$addr16	4	10	12	如果(saddr.bit) = 1, 则 $PC \leftarrow PC + 4 + jdisp8$ 然后复位(saddr.bit)	
		sfr.bit, \$addr16	4	–	12	如果 sfr.bit = 1, 则 $PC \leftarrow PC + 4 + jdisp8$ 然后复位 sfr.bit	
		A.bit, \$addr16	3	8	–	如果 A.bit = 1, 则 $PC \leftarrow PC + 3 + jdisp8$ 然后复位 A.bit	
		PSW.bit, \$addr16	4	–	12	如果 PSW.bit = 1, 则 $PC \leftarrow PC + 4 + jdisp8$ 然后复位 PSW.bit	× × ×
		[HL].bit, \$addr16	3	10	12	如果(HL).bit = 1, 则 $PC \leftarrow PC + 3 + jdisp8$ 然后复位 (HL).bit	
	DBNZ	B, \$addr16	2	6	–	$B \leftarrow B - 1$, 然后 如果 $B \neq 0$, 则 $PC \leftarrow PC + 2 + jdisp8$	
		C, \$addr16	2	6	–	$C \leftarrow C - 1$, 然后 如果 $C \neq 0$, 则 $PC \leftarrow PC + 2 + jdisp8$	
		saddr, \$addr16	3	8	10	$(saddr) \leftarrow (saddr) - 1$, 然后 如果 $(saddr) \neq 0$, 则 $PC \leftarrow PC + 3 + jdisp8$	
CPU 控制	SEL	RBn	2	4	–	$RBS1, 0 \leftarrow n$	
	NOP		1	2	–	无操作	
	EI		2	–	6	$IE \leftarrow 1$ (允许中断)	
	DI		2	–	6	$IE \leftarrow 0$ (禁止中断)	
	HALT		2	6	–	设置 HALT 模式	
	STOP		2	6	–	设置 STOP 模式	

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

3 按寻址类型列出指令

(1) 8 位指令

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

第 2 操作数 第 1 操作数	#byte	A	r [#]	sfr	saddr	laddr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	\$addr16	1	无
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
laddr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

注 “r = A” 除外

(2) 16 位指令

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

第 2 操作数 第 1 操作数	#word	AX	rp ^註	sfrp	saddrp	!addr16	SP	无
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW ^註						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

注 仅当 rp = BC, DE, HL 时使用

(3) 位操作指令

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

第 2 操作数 第 1 操作数	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	\$addr16	无
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) 调用指令 / 转移指令

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

第 2 操作数 第 1 操作数	AX	!addr16	!addr11	[addr5]	\$addr16
基本指令	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
复合指令					BT BF BTCLR DBNZ

(5) 其它指令

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP