

指令集说明

注：由于资源大小问题，其中以下几款 MCU 只有 62 条指令，其余均为 63 条指令。

HT48CA0/HT48RA0A 无 RETI 指令
HT48R05A-1 无 TABRDL 指令

指令寻址方式有下面 5 种：

➔ **立即寻址**

此一寻址法是将立即的常数值紧跟在运算码 (opcode) 后，例如：

```
MOV      A, 33H  
ADD      A, 33H
```

➔ **直接寻址**

直接寻址的情况只允许在存储器之间作数据传送，例如：

```
MOV      [33H], A  
CLR      [33H]
```

➔ **间接寻址**

在间接寻址方式中，必定要使用到间接寻址暂存器(R0, R1)，例如：

```
MOV      R0, A  
MOV      A, R0
```

➔ **特殊暂存器寻址**

此一寻址方式完全是针对某一暂存器作运算，例如：

```
CLR      WDT  
CLR      WDT1
```

➔ **指针寻址**

指针寻址只适用在配合做查表指令，例如：

```
MOV      A, 02H  
MOV      TBLP, A
```

指令集摘要

助记符	说明	指令周期	影响标志位
算术运算			
ADD A, [m]	累加器与数据存储器相加, 结果放入累加器	1	Z,C,AC,OV
ADDM A, [m]	累加器与数据存储器相加, 结果放入数据存储器	1 ⁽¹⁾	Z,C,AC,OV
ADD A, x	累加器与立即数相加, 结果放入累加器	1	Z,C,AC,OV
ADC A, [m]	累加器与数据存储器、进位标志相加, 结果放入累加器	1	Z,C,AC,OV
ADCM A, [m]	累加器与数据存储器、进位标志相加, 结果放入数据存储器	1 ⁽¹⁾	Z,C,AC,OV
SUB A, x	累加器与立即数相减, 结果放入累加器	1	Z,C,AC,OV
SUB A, [m]	累加器与数据存储器相减, 结果放入累加器	1	Z,C,AC,OV
SUBM A, [m]	累加器与数据存储器相减, 结果放入数据存储器	1 ⁽¹⁾	Z,C,AC,OV
SBC A, [m]	累加器与数据存储器、进位标志相减, 结果放入累加器	1	Z,C,AC,OV
SBCM A, [m]	累加器与数据存储器、进位标志相减, 结果放入数据存储器	1 ⁽¹⁾	Z,C,AC,OV
DAA [m]	将加法运算后放入累加器的值调整为十进制数, 并将结果放入数据存储器	1 ⁽¹⁾	C
逻辑运算			
AND A, [m]	累加器与数据存储器做“与”运算, 结果放入累加器	1	Z
OR A, [m]	累加器与数据存储器做“或”运算, 结果放入累加器	1	Z
XOR A, [m]	累加器与数据存储器做“异或”运算, 结果放入累加器	1	Z
ANDM A, [m]	累加器与数据存储器做“与”运算, 结果放入数据存储器	1 ⁽¹⁾	Z
ORM A, [m]	累加器与数据存储器做“或”运算, 结果放入数据存储器	1 ⁽¹⁾	Z
XORM A, [m]	累加器与数据存储器做“异或”运算, 结果放入数据存储器	1 ⁽¹⁾	Z
AND A, x	累加器与立即数做“与”运算, 结果放入累加器	1	Z
OR A, x	累加器与立即数做“或”运算, 结果放入累加器	1	Z
XOR A, x	累加器与立即数做“异或”运算, 结果放入累加器	1	Z
CPL [m]	对数据存储器取反, 结果放入数据存储器	1 ⁽¹⁾	Z
CPLA [m]	对数据存储器取反, 结果放入累加器	1	Z
递增和递减			
INCA [m]	数据存储器的内容加 1, 结果放入累加器	1	Z
INC [m]	数据存储器的内容加 1, 结果放入数据存储器	1 ⁽¹⁾	Z
DECA [m]	数据存储器的内容减 1, 结果放入累加器	1	Z
DEC [m]	数据存储器的内容减 1, 结果放入数据存储器	1 ⁽¹⁾	Z
移位			
RRA [m]	数据存储器右移一位, 结果放入累加器	1	无
RR [m]	数据存储器右移一位, 结果放入数据存储器	1 ⁽¹⁾	无
RRCA [m]	带进位将数据存储器右移一位, 结果放入累加器	1	C
RRC [m]	带进位将数据存储器右移一位, 结果放入数据存储器	1 ⁽¹⁾	C
RLA [m]	数据存储器左移一位, 结果放入累加器	1	无
RL [m]	数据存储器左移一位, 结果放入数据存储器	1 ⁽¹⁾	无
RLCA [m]	带进位将数据存储器左移一位, 结果放入累加器	1	C
RLC [m]	带进位将数据存储器左移一位, 结果放入数据存储器	1 ⁽¹⁾	C
数据传送			
MOV A, [m]	将数据存储器送至累加器	1	无
MOV [m], A	将累加器送至数据存储器	1 ⁽¹⁾	无
MOV A, x	将立即数送至累加器	1	无
位运算			
CLR [m]. i	将数据存储器的第 i 位清“0”	1 ⁽¹⁾	无
SET [m]. i	将数据存储器的第 i 位置“1”	1 ⁽¹⁾	无

助记符	说明	指令周期	影响标志位
转移			
JMP addr	无条件跳转	2	无
SZ [m]	如果数据存储器为“0”，则跳过下一条指令	1 ⁽²⁾	无
SZA [m]	数据存储器送至累加器，如果内容为“0”，则跳过下一条指令	1 ⁽²⁾	无
SZ [m].i	如果数据存储器的第 i 位为“0”，则跳过下一条指令	1 ⁽²⁾	无
SNZ [m].i	如果数据存储器的第 i 位不为“0”，则跳过下一条指令	1 ⁽²⁾	无
SIZ [m]	数据存储器加 1，如果结果为“0”，则跳过下一条指令	1 ⁽³⁾	无
SDZ [m]	数据存储器减 1，如果结果为“0”，则跳过下一条指令	1 ⁽³⁾	无
SIZA [m]	数据存储器加 1，将结果放入累加器，如果结果为“0”，则跳过下一条指令	1 ⁽²⁾	无
SDZA [m]	数据存储器减 1，将结果放入累加器，如果结果为“0”，则跳过下一条指令	1 ⁽²⁾	无
CALL addr	子程序调用	2	无
RET	从子程序返回	2	无
RET A, x	从子程序返回，并将立即数放入累加器	2	无
RETI	从中断返回	2	无
查表			
TABRDC [m]	读取当前页的 ROM 内容，并送至数据存储器 and TBLH	2 ⁽¹⁾	无
TABRDL [m]	读取最后页的 ROM 内容，并送至数据存储器 and TBLH	2 ⁽¹⁾	无
其它指令			
NOP	空指令	1	无
CLR [m]	清除数据存储器	1 ⁽¹⁾	无
SET [m]	置位数据存储器	1 ⁽¹⁾	无
CLR WDT	清除看门狗定时器	1	TO,PD
CLR WDT1	预清除看门狗定时器	1	TO ⁽⁴⁾ ,PD ⁽⁴⁾
CLR WDT2	预清除看门狗定时器	1	TO ⁽⁴⁾ ,PD ⁽⁴⁾
SWAP [m]	交换数据存储器的高低字节，结果放入数据存储器	1 ⁽¹⁾	无
SWAPA [m]	交换数据存储器的高低字节，结果放入累加器	1	无
HALT	进入暂停模式	1	TO,PD

注：x：立即数

m：数据存储器地址

A：累加器 ACC

i：第 0~7 位

addr：程序存储器地址

√：影响标志位

—：不影响标志位

⁽¹⁾：如果数据是加载到 PCL 寄存器，则指令执行周期会被延长一个指令周期（四个系统时钟）。

⁽²⁾：如果满足跳跃条件，则指令执行周期会被延长一个指令周期（四个系统时钟）；否则指令执行周期不会被延长。

⁽³⁾：⁽¹⁾ 和 ⁽²⁾

⁽⁴⁾：如果执行 CLR WDT1 或 CLR WDT2 指令后，看门狗定时器被清除，则会影响 TO 和 PD 标志位；否则不会影响 TO 和 PD 标志位。

ADC A, [m] 累加器与数据存储器、进位标志相加，结果放入累加器
 说明：本指令把累加器、数据存储器值以及进位标志相加，结果存放到累加器。
 运算过程： $ACC \leftarrow ACC + [m] + C$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

ADCM A, [m] 累加器与数据存储器、进位标志相加，结果放入数据存储器
 说明：本指令把累加器、数据存储器值以及进位标志相加，结果存放到存储器。
 运算过程： $[m] \leftarrow ACC + [m] + C$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

ADD A, [m] 累加器与数据存储器相加，结果放入累加器
 说明：本指令把累加器、数据存储器值相加，结果存放到累加器。
 运算过程： $ACC \leftarrow ACC + [m]$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

ADD A, x 累加器与立即数相加，结果放入累加器
 说明：本指令把累加器值和立即数相加，结果存放到累加器。
 运算过程： $ACC \leftarrow ACC + X$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

ADDM A, [m] 累加器与数据存储器相加，结果放入数据存储器
 说明：本指令把累加器、数据存储器值相加，结果放到数据存储器。
 运算过程： $[m] \leftarrow ACC + [m]$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

AND A, [m] 累加器与数据存储器做“与”运算，结果放入累加器
 说明：本指令把累加器值、数据存储器值做逻辑与，结果存放到累加器。
 运算过程： $ACC \leftarrow ACC \text{ “AND” } [m]$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

AND A, x 累加器与立即数做“与”运算，结果放入累加器
 说明：本指令把累加器值、立即数做逻辑与，结果存放到累加器。
 运算过程： $ACC \leftarrow ACC \text{ “AND” } X$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

ANDM A, [m] 累加器与数据存储器做“与”运算，结果放入数据存储器
 说明：本指令把累加器值、数据存储器值做逻辑与，结果放到数据存储器。
 运算过程： $ACC \leftarrow ACC \text{ “AND” } [m]$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

CALL addr 子程序调用
 说明：本指令直接调用地址所在处的子程序，此时程序计数器加一，将此程序计数器值存到堆栈寄存器中，再将子程序所在处的地址存放到程序计数器中。
 运算过程： $Stack \leftarrow PC+1$
 $PC \leftarrow addr$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

CLR [m] 清除数据存储器
 说明：本指令将数据存储器内的数值清零。
 运算过程： $[m] \leftarrow 00H$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

CLR [m].i 将数据存储器的第 i 位清“0”
 说明：本指令将数据存储器内第 i 位值清零。
 运算过程： $[m].i \leftarrow 0$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

CLR WDT 清除看门狗定时器
 说明：本指令清除 WDT 计数器（从 0 开始重新计数），暂停标志位（PD）和看门狗溢出标志位（TO）也被清零。
 运算过程： $WDT \leftarrow 00H$
 $PD \& TO \leftarrow 0$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	0	0	—	—	—	—

CLR WDT1 预清除看门狗定时器

说明：必须搭配 CLR WDT2 一起使用，才可清除 WDT 计时器（从 0 开始重新计数）。当程序只执行过该指令，没有执行 CLR WDT2 时，系统只会不会将暂停标志位（PD）和计数溢出位（TO）清零，PD 与 TO 保留原状态不变。

运算过程： $WDT \leftarrow 00H^*$

$PD \& TO \leftarrow 0^*$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	0*	0*	—	—	—	—

CLR WDT2 预清除看门狗定时器

说明：必须搭配 CLR WDT1 一起使用，才可清除 WDT 计时器（从 0 开始重新计数）。当程序只执行过该指令，没有执行 CLR WDT1 时，系统只会不会将暂停标志位（PD）和计数溢出位（TO）清零，PD 与 TO 保留原状态不变。

运算过程： $WDT \leftarrow 00H^*$

$PD \& TO \leftarrow 0^*$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	0*	0*	—	—	—	—

CPL [m] 对数据存储器取反，结果放入数据存储器

说明：本指令是将数据存储器内保存的数值取反。

运算过程： $[m] \leftarrow [\bar{m}]$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

CPLA [m] 对数据存储器取反，结果放入累加器

说明：本指令是将数据存储器内保存的值取反后，结果存放在累加器中。

运算过程： $ACC \leftarrow [\bar{m}]$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

DAA [m] 将加法运算后放入累加器的值调整为十进制数，并将结果放入数据存储器
 说明 本指令将累加器高低四位分别调整为 BCD 码。如果低四位的值大于“9”或 AC=1，那么 BCD 调整就执行对原值加“6”，并且内部进位标志 $AC1 = \overline{AC}$ ，即 AC 求反；否则原值保持不变。如果高四位的值大于“9”或 C=1，那么 BCD 调整就执行对原值加“6”再加 AC1，并把 C 置位；否则 BCD 调整就执行对原值加 AC1，C 的值保持不变。结果存放到数据存储器中，只有进位标志位（C）受影响。

操作 如果 $ACC.3 \sim ACC.0 > 9$ 或 $AC=1$
 那么 $[m].3 \sim [m].0 \leftarrow (ACC.3 \sim ACC.0) + 6, AC1 = \overline{AC}$
 否则 $[m].3 \sim [m].0 \leftarrow (ACC.3 \sim ACC.0), AC1 = 0$
 并且
 如果 $ACC.7 \sim ACC.4 + AC1 > 9$ 或 $C=1$
 那么 $[m].7 \sim [m].4 \leftarrow (ACC.7 \sim ACC.4) + 6 + AC1, C=1$
 否则 $[m].7 \sim [m].4 \leftarrow (ACC.7 \sim ACC.4) + AC1, C=C$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	√

DEC [m] 数据存储器内容减 1，结果放入数据存储器
 说明：本指令将数据存储器内的数值减一再放回数据存储器。
 运算过程： $[m] \leftarrow [m] - 1$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

DECA [m] 数据存储器内容减 1，结果放入累加器
 说明：本指令将存储器内的数值减一，再放到累加器。
 运算过程： $ACC \leftarrow [m] - 1$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

HALT 进入暂停模式
 说明：本指令终止程序执行并关掉系统时钟，RAM 和寄存器内的数值保持原状态，WDT 计数器清“0”，暂停标志位（PD）被设为 1，WDT 计数溢出位（TO）被清为 0。

运算过程：
 $PC \leftarrow PC + 1$
 $PD \leftarrow 1$
 $TO \leftarrow 0$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	0	1	—	—	—	—

INC [m] 数据存储器内容加 1，结果放入数据存储器
 说明：本指令将数据存储器内的数值加一，结果放回数据存储器。
 运算过程： $[m] \leftarrow [m]+1$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

INCA [m] 数据存储器内容加 1，结果放入数据存储器
 说明：本指令是将存储器内的数值加一，结果放到累加器。
 运算过程： $ACC \leftarrow [m]+1$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

JMP addr 无条件跳转
 说明：本指令是将要跳到的目的地直接放到程序计数器内。
 运算过程： $PC \leftarrow addr$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

MOV A, [m] 将数据存储器送至累加器
 说明：本指令是将数据存储器内的数值送到累加器内。
 运算过程： $ACC \leftarrow [m]$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

MOV A, x 将立即数送至累加器
 说明：本指令是将立即数送到累加器内。
 运算过程： $ACC \leftarrow X$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

MOV [m], A 将累加器送至数据存储器
 说明：本指令是将累加器值送到数据存储器内。
 运算过程： $[m] \leftarrow ACC$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

NOP

空指令

说明:

本指令不作任何运算, 而只将程序计数器加一。

运算过程:

 $PC \leftarrow PC+1$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

OR A, [m]

累加器与数据存储器做“或”运算, 结果放入累加器

说明:

本指令是把累加器、数据存储器值做逻辑或, 结果放到累加器。

运算过程:

 $ACC \leftarrow ACC \text{ “OR” } [m]$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

OR A, x

累加器与立即数做“或”运算, 结果放入累加器

说明:

本指令是把累加器值、立即数做逻辑或, 结果放到累加器。

运算过程:

 $ACC \leftarrow ACC \text{ “OR” } X$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

ORM A, [m]

累加器与数据存储器做“或”运算, 结果放入数据存储器

说明:

本指令是把累加器值、存储器值做逻辑或, 结果放到数据存储器。

运算过程:

 $ACC \leftarrow ACC \text{ “OR” } [m]$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

RET

从子程序返回

说明:

本指令是将堆栈寄存器中的程序计数器值送回程序计数器。

运算过程:

 $PC \leftarrow \text{Stack}$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RET A, x

从子程序返回, 并将立即数放入累加器

说明:

本指令是将堆栈寄存器中的程序计数器值送回程序计数器, 并将立即数送回累加器。

运算过程:

 $PC \leftarrow \text{Stack}$
 $ACC \leftarrow X$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RETI

从中断返回

说明:

本指令是将堆栈寄存器中的程序计数器值送回程序计数器，与 RET 不同的是它使用在中断程序结束返回时，它还会将中断控制寄存器 INTC 的 0 位 (EMI) 中断允许位置 1，允许中断服务。

运算过程:

 $PC \leftarrow Stack$
 $EMI \leftarrow 1$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RL [m]

数据存储器左移一位，结果放入数据存储器

说明:

本指令是将数据存储器内的数值左移一位，第 7 位移到第 0 位，结果送回数据存储器。

运算过程:

 $[m].0 \leftarrow [m].7, [m].(i+1) \leftarrow [m].i : i=0\sim 6$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RLA [m]

数据存储器左移一位，结果放入累加器

说明:

本指令是将存储器内的数值左移一位，第 7 位移到第 0 位，结果送到累加器，而数据存储器内的数值不变。

运算过程:

 $ACC.0 \leftarrow [m].7, ACC.(i+1) \leftarrow [m].i : i=0\sim 6$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RLC [m]

带进位将数据存储器左移一位，结果放入数据存储器

说明:

本指令是将存储器内的数值与进位位左移一位，第 7 位取代进位标志，进位标志移到第 0 位，结果送回数据存储器。

运算过程:

 $[m].(i+1) \leftarrow [m].i : i=0\sim 6$
 $[m].0 \leftarrow C$
 $C \leftarrow [m].7$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	√

RLCA [m]

带进位将数据存储器左移一位，结果放入累加器

说明:

本指令是将存储器内的数值与进位位左移一位，第七位取代进位标志，进位标志移到第 0 位，结果送回累加器。

运算过程:

 $ACC.(i+1) \leftarrow [m].i : i=0\sim 6$
 $ACC.0 \leftarrow C$
 $C \leftarrow [m].7$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	√

RR [m] 数据存储器右移一位，结果放入数据存储器

说明：本指令是将存储器内的数值循环右移，第 0 位移到第 7 位，结果送回数据存储器。

运算过程： $[m].7 \leftarrow [m].0, [m].i \leftarrow [m].(i+1) : i=0\sim6$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RRA [m] 数据存储器右移一位，结果放入累加器

说明：本指令是将数据存储器内的数值循环右移，第 0 位移到第 7 位，结果送回累加器，而数据存储器内的数值不变。

运算过程： $ACC.7 \leftarrow [m].0, ACC.i \leftarrow [m].(i+1) : i=0\sim6$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RRC [m] 带进位将数据存储器右移一位，结果放入数据存储器

说明：本指令是将存储器内的数值加进位位循环右移，第 0 位取代进位标志，进位标志移到第 7 位，结果送回存储器。

运算过程： $[m].i \leftarrow [m].(i+1) : i=0\sim6$

$[m].7 \leftarrow C$

$C \leftarrow [m].0$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	√

RRCA [m] 带进位将数据存储器右移一位，结果放入累加器

说明：本指令是将数据存储器内的数值加进位位循环右移，第 0 位取代进位标志，进位标志移到第 7 位，结果送回累加器，数据存储器内的数值不变。

运算过程： $ACC.i \leftarrow [m].(i+1) : i=0\sim6$

$ACC.7 \leftarrow C$

$C \leftarrow [m].0$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	√

SBC A,[m] 累加器与数据存储器、进位标志相减，结果放入累加器

说明：本指令是把累加器值减去数据存储器值以及进位标志的取反，结果放到累加器。

运算过程： $ACC \leftarrow ACC + [\overline{m}] + C$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

SBCM A,[m] 累加器与数据存储器、进位标志相减，结果放入数据存储器
 说明：本指令是把累加器值减去数据存储器值以及进位标志取反，结果放到数据存储器。
 运算过程： $[m] \leftarrow ACC + [\overline{m}] + C$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

SDZ [m] 数据存储器减 1，如果结果为“0”，则跳过下一条指令
 说明：本指令是把数据存储器内的数值减 1，判断是否为 0，若为 0 则跳过下一条指令，即如果结果为零，放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以取得正确的指令（二个指令周期）。否则执行下一条指令（一个指令周期）。
 运算过程：如果 $[m]-1=0$ ，跳过下一条指令执行再下一条。
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SDZA [m] 数据存储器减 1，将结果放入累加器，如果结果为“0”，则跳过下一条指令
 说明：本指令是把数据存储器内的数值减 1，判断是否为 0，为 0 则跳过下一行指令并将减完后数据存储器内的数值送到累加器，而数据存储器内的值不变，即若结果为 0，放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以取得正确的指令（二个指令周期）。否则执行下一条指令（一个指令周期）。
 运算过程：如果 $[m]-1=0$ ，跳过下一条指令执行再下一条。
 $ACC \leftarrow ([m]-1)$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SET [m] 置位数据存储器
 说明：本指令是把存储器内的数值每个位置为 1。
 运算过程： $[m] \leftarrow FFH$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SET [m].i 将数据存储器的第 i 位置“1”
 说明：本指令是把存储器内的数值的第 i 位置为 1。
 运算过程： $[m].i \leftarrow 1$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SIZ **[m]** 数据存储器加 1，如果结果为“0”，则跳过下一条指令
 说明： 本指令是把数据存储器内的数值加 1，判断是否为 0。若为 0，跳过下一条指令，即放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以取得正确的指令（二个指令周期）。否则执行下一条指令（一个指令周期）。
 运算过程： 如果 $([m]+1=0)$ ，跳过下一行指令； $[m] \leftarrow [m]+1$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SIZA 数据存储器加 1，将结果放入累加器，如果结果为“0”，则跳过下一条指令
 说明： 本指令是把数据存储器内的数值加 1，判断是否为 0，若为 0 跳过下一条指令，即放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以取得正确的指令（二个指令周期），并将加完后存储器内的数值送到累加器，而数据存储器的值保持不变。否则执行下一条指令（一个指令周期）。
 运算过程： 如果 $[m]+1=0$ ，跳过下一行指令； $ACC \leftarrow ([m]+1)$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SNZ **[m].i** 如果数据存储器的第 i 位不为“0”，则跳过下一条指令
 说明： 本指令是判断数据存储器内的数值的第 i 位，若不为 0，则程序计数器再加 1，跳过下一行指令，放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以取得正确的指令（二个指令周期）。否则执行下一条指令（一个指令周期）。
 运算过程： 如果 $[m].i \neq 0$ ，跳过下一行指令。
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SUB **A, [m]** 累加器与数据存储器相减，结果放入累加器
 说明： 本指令是把累加器值、数据存储器值相减，结果放到累加器。
 运算过程： $ACC \leftarrow ACC + [\bar{m}] + 1$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

SUB **A, x** 累加器与立即数相减，结果放入累加器
 说明： 本指令是把累加器值、立即数相减，结果放到累加器。
 运算过程： $ACC \leftarrow ACC + \bar{X} + 1$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

SUBM A, [m] 累加器与数据存储器相减，结果放入数据存储器

说明：本指令是把累加器值、存储器值相减，结果放到存储器。

运算过程： $[m] \leftarrow ACC + [\overline{m}] + 1$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

SWAP [m] 交换数据存储器的高低字节，结果放入数据存储器

说明：本指令是将数据存储器的低四位和高四位互换,再将结果送回数据存储器。

运算过程： $[m].7 \sim [m].4 \leftrightarrow [m].3 \sim [m].0$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SWAPA [m] 交换数据存储器的高低字节，结果放入累加器

说明：本指令是将数据存储器的低四位和高四位互换，再将结果送回累加器。

运算过程： $ACC.3 \sim ACC.0 \leftarrow [m].7 \sim [m].4$

$ACC.7 \sim ACC.4 \leftarrow [m].3 \sim [m].0$

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SZ [m] 如果数据存储器为“0”，则跳过下一条指令

说明：本指令是判断数据存储器内的数值是否为0，为0则跳过下一行指令，即放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以得正确的指令（二个指令周期）。否则执行下一条指令（一个指令周期）。

运算过程：如果 $[m] = 0$ ，跳过下一行指令。

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SZA [m] 数据存储器送至累加器，如果内容为“0”，则跳过下一条指令

说明：本指令是判断存储器内的数值是否为0，若为0则跳过下一行指令，即放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以得正确的指令（二个指令周期）。并把存储器内值送到累加器，而存储器的值保持不变。否则执行下一条指令（一个指令周期）。

运算过程：如果 $[m] = 0$ ，跳过下一行指令，并 $ACC \leftarrow [m]$ 。

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SZ **[m].i** 如果数据存储器的第 i 位为 “0”，则跳过下一条指令
 说明： 本指令是判断存储器内第 i 位值是否为 0，若为 0 则跳过下一行指令，即放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以得正确的指令（二个指令周期）。否则执行下一条指令（一个指令周期）。

运算过程： 如果 $[m].i = 0$ ，跳过下一行指令。
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

TABRDC [m] 读取 ROM 当前页的内容，并送至数据存储器 and TBLH
 说明： 本指令是将表格指针指向程序寄存器当前页，将低位送到存储器，高位直接送到 TBLH 寄存器内。

运算过程： $[m] \leftarrow$ 程序存储器低四位
 TBLH \leftarrow 程序存储器高四位

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

TABRDL [m] 读取 ROM 最后一页的内容，并送至数据存储器 and TBLH
 说明： 本指令是将 TABLE 指针指向程序寄存器最后页，将低位送到存储器，高位直接送到 TBLH 寄存器内。

运算过程： $[m] \leftarrow$ 程序存储器低四位
 TBLH \leftarrow 程序存储器高四位

影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

XOR A, [m] 累加器与立即数做 “异或” 运算，结果放入累加器
 说明： 本指令是把累加器值、数据存储器值做逻辑异或，结果放到累加器。

运算过程： $ACC \leftarrow ACC \text{ “XOR” } [m]$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

XORM A, [m] 累加器与数据存储器做 “异或” 运算，结果放入数据存储器
 说明： 本指令是把累加器值、数据存储器值做逻辑异或，结果放到数据存储器。

运算过程： $[m] \leftarrow ACC \text{ “XOR” } [m]$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

XOR A, x 累加器与数据存储器做 “异或” 运算，结果放入累加器
 说明： 本指令是把累加器值与立即数做逻辑异或，结果放到累加器。

运算过程： $ACC \leftarrow ACC \text{ “XOR” } X$
 影响标志位

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—