



## 指令集说明

### ● 简介

这里讲的是 MK7 系列的指令集。

### ● 指令集说明 1 (MK7A10P、MK7A20P、MK7A11P)

#### 1. 寄存器相加

格式: ADD M, m

操作: (M)+(acc) → (M)

影响状态位: C, DC, Z

指令周期: 1

例: ADD TEMP, m

指令执行前 TEMP=0FH

A=0F0H

指令执行后 TEMP=0FFH

A=0F0H

#### 2. 寄存器相加

格式: ADD M, a

操作: (M)+(acc) → (acc)

影响状态位: C, DC, Z

指令周期: 1

例: ADD TEMP, a

指令执行前 TEMP=0FH

A=0F0H

指令执行后 TEMP=0FH

A=0FFH

#### 3. 寄存器与

格式: AND M, m

操作: (M) · (acc) → (M)

影响状态位: Z

指令周期: 1

例: AND TEMP, m



指令执行前      TEMP=11H  
                          A=0FH  
指令执行后      TEMP=01H  
                          A=0FH

4. 寄存器与

格式: AND M, a

操作: (M) · (acc) → (acc)

影响状态位:Z

指令周期:1

例: AND TEMP, a

指令执行前              TEMP=11H  
                          A=0FH  
指令执行后      TEMP=11H  
                          A=01H

5. 立即数和寄存器与

格式: ANDLA I

操作: Literal · (acc) → (acc)

影响状态位:Z

指令周期:1

例: ANDLA    11H

指令执行前              A=0FH  
指令执行后              A=01H

6. 清位操作

格式: BC M, bn

操作: 清M的第n位

影响状态位:无

指令周期:1

例: BC    TEMP, b0

指令执行前              TEMP=11H  
指令执行后              TEMP=10H

7. 置位操作

格式: BS M, bn

操作: 置M的第n位

影响状态位: 无

指令周期:1

例: BS    TEMP, b0

指令执行前              TEMP=10H  
指令执行后              TEMP=11H



8. 清零测试

格式: BTSC M, bn

操作: 当M的第n位为0时, 跳过下一条指令

影响状态位: 无

指令周期: 1 + (skip)

例: NOW: BTSC flag, b0

FALSE: LGOTO PROCESS

TURE: .....

.....

指令执行前 PC=address(NOW)

指令执行后

if flag<b0>=0 PC= address(TURE)

if flag<b0>=1 PC= address(FALSE)

9. 置位测试

格式: BTSS M, bn

操作: 当M的第n位为1时, 跳过下一条指令

影响状态位: 无

指令周期: 1 + (skip)

例: NOW: BTSS flag, b0

FALSE: LGOTO PROCESS

TURE: .....

.....

指令执行前 PC=address(NOW)

指令执行后

if flag<b0>=1 PC= address(TURE)

if flag<b0>=0 PC= address(FALSE)

10. 清 A 寄存器

格式: CLRA

操作: 清A寄存器

影响状态位: Z

指令周期: 1

例: CLRA

指令执行前 A=68H

指令执行后

A=00H

11. 清寄存器

格式: CLR M

操作: 清M寄存器

影响状态位: Z

指令周期: 1



例: CLR TEMP

指令执行前

TEMP =68H

指令执行后

TEMP =00H

12. 清 WDT

格式: CLRWDT

操作: 00H→WDT

影响状态位: TO, PO

指令周期:1

例: CLRWDT

指令执行后:

WDT counter=0

13. 寄存器取反

格式: COM M, m

操作: ~(M) → (M)

影响状态位:Z

指令周期:1

例: COM TEMP, m

指令执行前

TEMP =FFH

指令执行后

TEMP =00H

14. 寄存器取反

格式: COM M, a

操作: ~(M) → (acc)

影响状态位:Z

指令周期:1

例: COM TEMP, a

指令执行前

TEMP =FFH

指令执行后

TEMP =FFH

A=00H

15. 寄存器减 1

格式: DEC M, m

操作: (M)--1 → (M)

影响状态位:Z

指令周期:1

例: DEC TEMP, m



指令执行前           TEMP =FFH  
指令执行后  
                          TEMP =FEH

16. 寄存器减 1

格式: DEC M, a

操作: (M) - 1 → (acc)

影响状态位:Z

指令周期:1

例: DEC TEMP,a

指令执行前           TEMP =FFH  
指令执行后  
                          TEMP =FFH  
                          A= FEH

17. 寄存器减 1 测试

格式: DECSZ M, m

操作: (M) - 1 → (M), 若(M) = 0跳转

影响状态位:无

指令周期: 1 + (skip)

例:

```
HERE:  DECSZ  TEMP,m
        LGOTO  LOOP
```

```
CONT:  .....
        .....
```

指令执行前 PC=address(HERE)

指令执行后

TEMP =TEMP-1

If TEMP=0

PC=address(CONT)

If TEMP!=0

PC=address(HERE+1)

18. 寄存器减 1 测试

格式: DECSZ M, a

操作: (M) - 1 → (acc), 若(M) = 0跳转

影响状态位: 无

指令周期: 1 + (skip)

例: HERE: DECSZ TEMP,a

```
        LGOTO  LOOP
```



CONT: .....

指令执行前 PC=address(HERE)

指令执行后

A =TEMP-1

If A=0

PC=address(CONT)

If A!=0

PC=address(HERE+1)

19. 寄存器加 1

格式: INC M, m

操作: (M) + 1 → (M)

影响状态位: Z

指令周期: 1

例: INC TEMP,m

指令执行前

TEMP =FEH

指令执行后

TEMP =FFH

20. 寄存器加 1

格式: INC M, a

操作: (M) + 1 → (acc)

影响状态位: Z

指令周期: 1

例: INC TEMP,a

指令执行前

TEMP =FEH

指令执行后

TEMP =FEH

A= FFH

21. 寄存器加 1 测试

格式: INCSZ M, m

操作: (M) + 1 → (M), 若(M) = 0跳转

影响状态位: 无

指令周期: 1 + (skip)

例:

HERE: INCSZ TEMP,m

LGOTO LOOP

CONT: .....



指令执行前 PC=address(HERE)  
指令执行后

TEMP =TEMP+1  
If TEMP=0  
PC=address(CONT)  
If TEMP!=0  
PC=address(HERE+1)

22. 寄存器加 1 测试

格式: INCSZ M, a

操作: (M) + 1 → (acc), 若(M) = 0跳转

影响状态位: 无

指令周期: 1 + (skip)

例: HERE: INCSZ TEMP,a  
LGOTO LOOP

CONT: .....

指令执行前 PC=address(HERE)  
指令执行后

A =TEMP+1  
If A=0  
PC=address(CONT)  
If A!=0  
PC=address(HERE+1)

23. 设置读写方式

格式: IODIR M

操作:A→IO控制寄存器

影响状态位: 无

指令周期: 1

例:IODIR PORTA

指令执行前

A=0FFH

指令执行后

PORTA 控制寄存器=0FFH

24. 寄存器或

格式: IOR M, m

操作: (M) ior (acc) → (M)



影响状态位: Z

指令周期: 1

例: IOR TEMP, m

指令执行前

TEMP=01H

A=0F0H

指令执行后

TEMP=0F1H

A=0F0H

## 25. 寄存器或

格式: IOR M, a

操作: (M) ior (acc) → (acc)

影响状态位: Z

指令周期: 1

例: IOR TEMP, a

指令执行前

TEMP=01H

A=0F0H

指令执行后 TEMP=01H

A=0F1H

## 26. 立即数与 A 寄存器或

格式: IORLA I

操作: Literal ior (acc) → (acc)

影响状态位: Z

指令周期: 1

例: IORLA 01H

指令执行前

A=0F0H

指令执行后

A=0F1H

## 27. 调用子程序

格式: LCALL I

操作: I → PC

影响状态位: 无

指令周期: 2

例: HERE: LCALL THERE

指令执行前

PC=address(HERE)

指令执行后

PC=address(THERE)





---

PC=address(HERE+1)

28. 跳转

格式: LGOTO I

操作: I→PC

影响状态位: 无

指令周期: 2

例: LGOTO THERE

指令执行后

PC=address(THERE)

29. 寄存器传送

格式: MOVAM m

操作: (acc) → (M)

影响状态位: 无

指令周期: 1

例: MOVAM TEMP

指令执行前

TEMP=0EEH

A=0FFH

指令执行后 TEMP=0FFH

A=0FFH

30. 立即数传送到 A 寄存器

格式: MOVLA I

操作: Literal→(acc)

影响状态位: 无

指令周期: 1

例: MOVLA 0EEH

指令执行后

A=0EEH

31. 空操作

格式: NOP

操作: 空操作

影响状态位: 无

指令周期: 1

例: NOP

32. 子程序返回

格式: RET

操作: TOS→PC



影响状态位: 无

指令周期: 2

例:CONT: CALL TAB

.....  
.....

TAB:

.....

HERE: .....

RET

指令执行前 PC= address (HERE)

指令执行后 PC= address (CONT+1)

33. 中断返回

格式: RETI

操作: TOS→PC , INTM=1

影响状态位: 无

指令周期: 2

例:CONT: NOP

MOVLA 08H

MOVAM TEMP

.....  
.....

INTER:

.....  
.....

RETI

指令执行前 PC= address (CONT)

指令执行后 PC= address (CONT+1)

34. 子程序返回并传送立即数到 A 寄存器

格式: RETLA I

操作: I → (acc) , TOS→PC

影响状态位: 无

指令周期: 2

例: MOVLA 01H

CALL TAB

.....  
.....

TAB:

.....



.....  
RETLA 0FEH

指令执行前 A= 01H

指令执行后 A= 0FEH

35. 寄存器带进位循环左移

格式: RL M, m

操作: 寄存器带进位循环左移

影响状态位: C

指令周期: 1

例: RL TEMP, m

指令执行前                   TEMP=80H

C=0

指令执行后                   TEMP=00H

C=1

36. 寄存器带进位循环左移

格式: RL M, a

操作: 寄存器带进位循环左移, 结果存入A

影响状态位: C

指令周期: 1

例:

RL TEMP, a

指令执行前                   TEMP=80H

C=0

A=00H

指令执行后                   TEMP=80H

C=1

A=00H

37. 寄存器带进位循环右移

格式: RR M, m

操作: 寄存器带进位循环右移

影响状态位: C

指令周期: 1

例: RR TEMP, m

指令执行前                   TEMP=01H

C=0

指令执行后                   TEMP=00H

C=1



38. 寄存器带进位循环右移

格式: RR M, a

操作: 寄存器带进位循环右移, 结果存入A

影响状态位: C

指令周期: 1

例: RR TEMP, a

指令执行前

TEMP=01H

C=0

A=00H

指令执行后

TEMP=01H

C=1

A=00H

39. 设置 SELECT 寄存器

格式: SELECT

操作: (acc) → SELECT

影响状态位: 无

指令周期: 1

例: SELECT

指令执行前 (A)=0FH

指令执行后 SELECT=0FH

40. 进入睡眠模式

格式: SLEEP

操作: 00H → WDT

影响状态位: TO, PO

指令周期: 1

例: SLEEP

41. 寄存器相减

格式: SUB M, m

操作: (M) - (acc) → (M)

影响状态位: C, DC, Z

指令周期: 1

例: SUB TEMP, m

指令执行前

TEMP=0FFH

A=0EEH

指令执行后

TEMP=11H

A=0EEH



42. 寄存器相减

格式: SUB M, a

操作: (M) -(acc) → (acc)

影响状态位: C, DC, Z

指令周期: 1

例: SUB TEMP, a

指令执行前           TEMP=0FFH

A=0EEH

指令执行后           TEMP=0FFH

A=11H

43. 交换寄存器高低四位

格式: SWAP M, m

操作: (M<3:0>)→(M<7:4>)

(M<7:4>)→(M<3:0>)

影响状态位: 无

指令周期: 1

例: SWAP TEMP, m

指令执行前           TEMP=0F1H

指令执行后           TEMP=1FH

44. 交换寄存器高低四位

格式: SWAP M, a

操作: (M<3:0>)→(M<7:4>)

(M<7:4>)→(M<3:0>)

影响状态位: 无

指令周期: 1

例: SWAP TEMP, a

指令执行前           TEMP=0F1H

指令执行后           TEMP=0F1H

A=1FH

45. 寄存器异或

格式: XOR M, m

操作: (M) xor (acc) → (M)

影响状态位: Z

指令周期: 1



例: XOR TEMP, m

指令执行前	A=0FH
	TEMP=0FFH
指令执行后	A=0FH
	TEMP=0F0H

#### 46. 寄存器异或

格式: XOR M, a

操作: (M) xor (acc) → (acc)

影响状态位: Z

指令周期: 1

例: XOR TEMP, a

指令执行前	A=0FH
	TEMP=0FFH
指令执行后	A=0F0H
	TEMP=0FFH

#### 47. 立即数与寄存器异或

格式: XORLA I

操作: Literal xor (acc) → (acc)

影响状态位: Z

指令周期: 1

例: XORLA 0FFH

指令执行前	A=0FH
指令执行后	A=0F0H

### ● 指令集说明 2 (MK7A21P)

#### 48. 寄存器相加

格式: ADD M, m

操作: (M)+(acc) → (M)

影响状态位: C, DC, Z

指令周期: 1

例: ADD TEMP, m

指令执行前	TEMP=0FH
	A=0F0H
指令执行后	TEMP=0FFH
	A=0F0H

#### 49. 寄存器相加



格式: ADDC M, m

操作: (M)+(acc)+(carry) → (M)

影响状态位: C, DC, Z

指令周期: 1

例: ADDC TEMP, m

指令执行前 TEMP=08H

A=0F0H

C=1

指令执行后 TEMP=0F9H

A=0F0H

#### 50. 寄存器相加

格式: ADD M, a

操作: (M)+(acc) → (acc)

影响状态位: C, DC, Z

指令周期: 1

例: ADD TEMP, a

指令执行前 TEMP=0FH

A=0F0H

指令执行后 TEMP=0FH

A=0FFH

#### 51. 寄存器相加

格式: ADDC M, a

操作: (M)+(acc) +(carry) → (acc)

影响状态位: C, DC, Z

指令周期: 1

例: ADDC TEMP, a

指令执行前 TEMP=08H

A=0F0H

C=1

指令执行后 TEMP=08H

A=0F9H

#### 52. 寄存器与

格式: AND M, m

操作: (M) · (acc) → (M)

影响状态位: Z

指令周期: 1



例: AND TEMP, m

指令执行前 TEMP=11H

A=0FH

指令执行后 TEMP=01H

A=0FH

### 53. 寄存器与

格式: AND M, a

操作: (M) · (acc) → (acc)

影响状态位:Z

指令周期:1

例: AND TEMP, a

指令执行前 TEMP=11H

A=0FH

指令执行后 TEMP=11H

A=01H

### 54. 立即数和寄存器与

格式: ANDLA I

操作: Literal · (acc) → (acc)

影响状态位:Z

指令周期:1

例: ANDLA 11H

指令执行前 A=0FH

指令执行后 A=01H

### 55. 清位操作

格式: BC M, bn

操作: 清M的第n位

影响状态位:无

指令周期:1

例: BC TEMP, b0

指令执行前 TEMP=11H

指令执行后 TEMP=10H

### 56. 置位操作

格式: BS M, bn

操作: 置M的第n位

影响状态位: 无

指令周期:1

例: BS TEMP, b0

指令执行前 TEMP=10H





指令执行后

TEMP=11H

57. 清零测试

格式: BTSC M, bn

操作: 当M的第n位为0时, 跳过下一条指令

影响状态位: 无

指令周期: 1 + (skip)

例: NOW: BTSC flag, b0

FALSE: LGOTO PROCESS

TURE: .....

.....

指令执行前 PC=address(NOW)

指令执行后

if flag<b0>=0 PC= address(TURE)

if flag<b0>=1 PC= address(FALSE)

58. 置位测试

格式: BTSS M, bn

操作: 当M的第n位为1时, 跳过下一条指令

影响状态位: 无

指令周期: 1 + (skip)

例: NOW: BTSS flag, b0

FALSE: LGOTO PROCESS

TURE: .....

.....

指令执行前 PC=address(NOW)

指令执行后

if flag<b0>=1 PC= address(TURE)

if flag<b0>=0 PC= address(FALSE)

59. 清 A 寄存器

格式: CLRA

操作: 清A寄存器

影响状态位: Z

指令周期: 1

例: CLRA

指令执行前 A=68H

指令执行后 A=00H

60. 清寄存器

格式: CLR M

操作: 清M寄存器

影响状态位: Z



指令周期:1

例: CLR TEMP

指令执行前

TEMP =68H

指令执行后

TEMP =00H

61. 清 WDT

格式: CLRWDT

操作: 00H→WDT

影响状态位: TO, PO

指令周期:1

例: CLRWDT

指令执行后:

WDT counter=0

62. 寄存器取反

格式: COM M, m

操作: ~(M) → (M)

影响状态位:Z

指令周期:1

例:COM TEMP,m

指令执行前

TEMP =FFH

指令执行后

TEMP =00H

63. 寄存器取反

格式: COM M, a

操作: ~(M) → (acc)

影响状态位:Z

指令周期:1

例: COM TEMP,a

指令执行前

TEMP =FFH

指令执行后

TEMP =FFH

A=00H

64. 十进制调整

格式: DA M, a

操作:

If ACC[3:0] > 9 or DC=1

Then

ACC[3:0]←ACC[3:0]+6,



```
DC1=~DC
else
ACC[3:0] ←ACC[3:0],
DC1=0
If ACC[7:4]+DC1 > 9 or C=1
Then
ACC[7:4] ←ACC[7:4]+6+DC1,
C=1
else
ACC[7:4] ←ACC[7:4]+DC1,
C=C
```

影响状态位: C

指令周期: 1

例:

```
movla 96H
movam 40h
movla 25H
add 40h,m
da 40h,a
指令执行前      C =0
指令执行后      A=21H
                  C=1
```

#### 65. 十进制调整

格式: DA M, m

操作:

```
If ACC[3:0] > 9 or DC=1
Then
M[3:0] ←ACC[3:0]+6,
DC1=DC
else
M[3:0] ←ACC[3:0], DC1=0
If ACC[7:4]+DC1 > 9 or C=1
Then
M[7:4] ←ACC[7:4]+6+DC1,
C=1
else M[7:4] ←ACC[7:4]+DC1,
C=C
```

影响状态位: C

指令周期: 1

例:



```

movla 96H
movam 40h
movla 25H
add 40h,m
da 40h,m
指令执行前      C=0
指令执行后      (40H) =21H
                  C=1

```

66. 寄存器减 1

格式: DEC M, m

操作: (M)--1 → (M)

影响状态位:Z

指令周期:1

例: DEC TEMP,m

指令执行前 TEMP =FFH

指令执行后 TEMP =FEH

67. 寄存器减 1

格式: DEC M, a

操作: (M) - 1 → (acc)

影响状态位:Z

指令周期:1

例: DEC TEMP,a

指令执行前 TEMP =FFH

指令执行后 TEMP =FFH  
 A= FEH

68. 寄存器减 1 测试

格式: DECSZ M, m

操作: (M) - 1 → (M), 若(M) = 0跳转

影响状态位:无

指令周期: 1 + (skip)

例:

```

HERE:  DECSZ  TEMP,m
       LGOTO  LOOP

```

```

CONT:  .....
       .....

```



指令执行前 PC=address(HERE)  
 指令执行后  
 TEMP =TEMP-1  
 If TEMP=0  
 PC=address(CONT)  
 If TEMP!=0  
 PC=address(HERE+1)

69. 寄存器减 1 测试

格式: DECSZ M, a

操作: (M) - 1 → (acc), 若(M) = 0跳转

影响状态位: 无

指令周期: 1 + (skip)

例: HERE: DECSZ TEMP,a  
 LGOTO LOOP

CONT: .....

指令执行前 PC=address(HERE)

指令执行后

A =TEMP-1  
 If A=0  
 PC=address(CONT)  
 If A!=0  
 PC=address(HERE+1)

70. 寄存器加 1

格式: INC M, m

操作: (M) + 1 → (M)

影响状态位: Z

指令周期: 1

例: INC TEMP,m

指令执行前

TEMP =FEH

指令执行后

TEMP =FFH

71. 寄存器加 1

格式: INC M, a

操作: (M) + 1 → (acc)

影响状态位: Z

指令周期: 1



例: INC TEMP,a

指令执行前

TEMP =FEH

指令执行后

TEMP =FEH

A= FFH

72. 寄存器加 1 测试

格式: INCSZ M, m

操作: (M) + 1 → (M), 若(M) = 0跳转

影响状态位: 无

指令周期: 1 + (skip)

例:

HERE: INCSZ TEMP,m  
LGOTO LOOP

CONT: .....

指令执行前

PC=address(HERE)

指令执行后

TEMP =TEMP+1

If TEMP=0

PC=address(CONT)

If TEMP!=0

PC=address(HERE+1)

73. 寄存器加 1 测试

格式: INCSZ M, a

操作: (M) + 1 → (acc), 若(M) = 0跳转

影响状态位: 无

指令周期: 1 + (skip)

例: HERE: INCSZ TEMP,a  
LGOTO LOOP

CONT: .....

指令执行前

PC=address(HERE)

指令执行后

A =TEMP+1

If A=0

PC=address(CONT)

If A!=0

PC=address(HERE+1)



74. 寄存器或

格式: IOR M, m

操作: (M) ior (acc) → (M)

影响状态位: Z

指令周期: 1

例: IOR TEMP, m

指令执行前

TEMP=01H

A=0F0H

指令执行后

TEMP=0F1H

A=0F0H

75. 寄存器或

格式: IOR M, a

操作: (M) ior (acc) → (acc)

影响状态位: Z

指令周期: 1

例: IOR TEMP, a

指令执行前

TEMP=01H

A=0F0H

指令执行后 TEMP=01H

A=0F1H

76. 立即数与 A 寄存器或

格式: IORLA I

操作: Literal ior (acc) → (acc)

影响状态位: Z

指令周期: 1

例: IORLA 01H

指令执行前

A=0F0H

指令执行后

A=0F1H

77. 测试跳转

格式: JC I

操作: 如 C=1 则跳转

影响状态位: NONE

指令周期: 2

例: MOVLA 0FFH

MOVAM 40H



ADD 40H,A  
JC NEXT

.....  
NEXT: .....

.....  
指令执行后 PC=address(NEXT)

78. 测试跳转

格式: JZ I

操作: 如 Z=1则跳转

影响状态位: NONE

指令周期: 2

例:           MOVLA 00H  
              MOVAM 40H  
              JZ NEXT

.....  
NEXT: .....

.....  
指令执行后 PC=address(NEXT)

79. 调用子程序

格式: LCALL I

操作: I→PC

影响状态位: 无

指令周期: 2

例: HERE: LCALL THERE

指令执行前  
          PC=address(HERE)  
指令执行后  
          PC=address(THERE)  
          PC=address(HERE+1)

80. 跳转

格式: LGOTO I

操作: I→PC

影响状态位: 无

指令周期: 2

例:       LGOTO THERE

指令执行后  
          PC=address(THERE)

81. 寄存器传送





格式: MOVAM m

操作: (acc) → (M)

影响状态位: 无

指令周期: 1

例: MOVAM TEMP

指令执行前	TEMP=0EEH
	A=0FFH

指令执行后	TEMP=0FFH
	A=0FFH

82. 立即数传送到 A 寄存器

格式: MOVLA I

操作: Literal → (acc)

影响状态位: 无

指令周期: 1

例: MOVLA 0EEH

指令执行后	A=0EEH
-------	--------

83. 空操作

格式: NOP

操作: 空操作

影响状态位: 无

指令周期: 1

例: NOP

84. 子程序返回

格式: RET

操作: TOS → PC

影响状态位: 无

指令周期: 2

例: CONT: CALL TAB

.....  
.....

TAB: .....  
.....

HERE: .....  
RET

指令执行前	PC=	address (HERE)
指令执行后	PC=	address (CONT+1)



85. 中断返回

格式: RETI

操作: TOS→PC , INTM=1

影响状态位: 无

指令周期: 2

例:CONT: NOP

MOVLA 08H

MOVAM TEMP

.....

.....

INTER:

.....

.....

RETI

指令执行前 PC= address (CONT)

指令执行后 PC= address (CONT+1)

86. 中断返回

格式: RET\_INT

操作: TOS→PC

影响状态位: 无

指令周期: 2

例:CONT: NOP

MOVLA 08H

MOVAM TEMP

.....

.....

INTER:

.....

.....

RET\_INT

指令执行前 PC= address (CONT)

指令执行后 PC= address (CONT+1)

87. 子程序返回并传送立即数到 A 寄存器

格式: RETLA I

操作: I → (acc) , TOS→PC

影响状态位: 无

指令周期: 2



例: MOVLA 01H  
CALL TAB

.....  
.....

TAB:

.....  
.....

RETLA 0FEH

指令执行前 A= 01H

指令执行后 A= 0FEH

88. 寄存器不带进位循环左移

格式: RL M, m

操作: 寄存器不带进位循环左移

影响状态位: 无

指令周期: 1

例: RL TEMP, m

指令执行前 TEMP=80H

指令执行后 TEMP=01H

89. 寄存器带进位循环左移

格式: RLC M, m

操作: 寄存器带进位循环左移

影响状态位: C

指令周期: 1

例: RLC TEMP, m

指令执行前 TEMP=80H

C=0

指令执行后 TEMP=00H

C=1

90. 寄存器不带进位循环左移

格式: RL M, a

操作: 寄存器不带进位循环左移, 结果存入A

影响状态位: 无

指令周期: 1

例: RL TEMP, a

指令执行前 TEMP=80H

A=00H

指令执行后 TEMP=80H

A=01H

91. 寄存器带进位循环左移

格式: RLC M, a



操作: 寄存器带进位循环左移, 结果存入A

影响状态位: C

指令周期: 1

例:

RLC TEMP, a

指令执行前           TEMP=80H

C=0

A=00H

指令执行后           TEMP=80H

C=1

A=00H

92. 寄存器不带进位循环右移

格式: RR M, m

操作: 寄存器不带进位循环右移

影响状态位: 无

指令周期: 1

例: RR TEMP, m

指令执行前           TEMP=01H

指令执行后           TEMP=80H

93. 寄存器带进位循环右移

格式: RRC M, m

操作: 寄存器带进位循环右移

影响状态位: C

指令周期: 1

例: RRC TEMP, m

指令执行前           TEMP=01H

C=0

指令执行后           TEMP=00H

C=1

94. 寄存器不带进位循环右移

格式: RR M, a

操作: 寄存器不带进位循环右移, 结果存入A

影响状态位: 无

指令周期: 1

例: RR TEMP, a

指令执行前           TEMP=01H

A=00H

指令执行后           TEMP=01H

A=80H

95. 寄存器带进位循环右移



格式: RRC M, a

操作: 寄存器带进位循环右移, 结果存入A

影响状态位: C

指令周期: 1

例: RRC TEMP, a

指令执行前

TEMP=01H

C=0

A=00H

指令执行后

TEMP=01H

C=1

A=00H

#### 96. 设置 SELECT 寄存器

格式: SELECT

操作: (acc) → SELECT

影响状态位: 无

指令周期: 1

例: SELECT

指令执行前 (A)=0FH

指令执行后 SELECT=0FH

#### 97. 进入睡眠模式

格式: SLEEP

操作: 00H → WDT

影响状态位: TO, PO

指令周期: 1

例: SLEEP

#### 98. 寄存器相减

格式: SUB M, m

操作: (M) - (acc) → (M)

影响状态位: C, DC, Z

指令周期: 1

例: SUB TEMP, m

指令执行前

TEMP=0FFH

A=0EEH

指令执行后

TEMP=11H

A=0EEH

#### 99. 寄存器相减

格式: SUBC M, m



操作: (M)-(acc)+(carry) → (M)

影响状态位: C, DC, Z

指令周期: 1

例: SUBC TEMP, m

指令执行前	TEMP=0FFH A=0EEH C=1
-------	----------------------------

指令执行后	TEMP=12H A=0EEH
-------	--------------------

100. 寄存器相减

格式: SUB M, a

操作: (M) -(acc) → (acc)

影响状态位: C, DC, Z

指令周期: 1

例: SUB TEMP, a

指令执行前	TEMP=0FFH A=0EEH
-------	---------------------

指令执行后	TEMP=0FFH A=11H
-------	--------------------

101. 寄存器相减

格式: SUBC M, a

操作: (M) -(acc) +(carry) → (acc)

影响状态位: C, DC, Z

指令周期: 1

例: SUBC TEMP, a

指令执行前	TEMP=0FFH A=0EEH C=1
-------	----------------------------

指令执行后	TEMP=0FFH A=12H
-------	--------------------

102. 交换寄存器高低四位

格式: SWAP M, m

操作: (M<3:0>)→(M<7:4>)

(M<7:4>)→(M<3:0>)

影响状态位: 无



指令周期: 1

例: SWAP TEMP, m

指令执行前 TEMP=0F1H

指令执行后 TEMP=1FH

103. 交换寄存器高低四位

格式: SWAP M, a

操作: (M<3:0>) $\rightarrow$ (M<7:4>)

(M<7:4>) $\rightarrow$ (M<3:0>)

影响状态位: 无

指令周期: 1

例: SWAP TEMP, a

指令执行前 TEMP=0F1H

指令执行后 TEMP=0F1H

A=1FH

104. 读表格高字节

格式: TABRDH M

操作: M所指向存储器的高字节 $\rightarrow$ (acc)

影响状态位: 无

指令周期: 2

例: #DEFINE TAB\_BNK 3EH

BUFA EQU 40H

ORG 0700H

MOVLA 00H

DW 1122H

DW 3344H

DW 6677H

DW 55AAH

MOVLA 03H ; 设置 BANK 内偏移地址

MOVAM BUFA

MOVLA 0FH

MOVAM TAB\_BNK ; 设置 BANK 位置

TABRDH BUFA

指令执行后 A=66H

105. 读表格低字节

格式: TABRDL M

操作: M所指向存储器的低字节 $\rightarrow$ (acc)



影响状态位: 无

指令周期: 2

例:

```
#DEFINE TAB_BNK 3EH
BUFA EQU 40H
ORG 0700H
MOVLA 00H
DW 1122H
    DW 3344H
    DW 6677H
    DW 55AAH
    MOVLA 03H           ; 设置 BANK 内偏移地址
    MOVAM BUFA
    MOVLA 0FH
    MOVAM TAB_BNK     ; 设置 BANK 位置
    TABRDL BUFA
    指令执行后 A=77H
```

106. 寄存器清零测试

格式: TMSC M

操作: 当M为0时, 跳过下一条指令

影响状态位: 无

指令周期: 1 + (skip)

例: NOW: TMSC M

FALSE: LGOTO PROCESS

TURE: .....

指令执行前 PC=address(NOW)

指令执行后

if M=0 PC= address(TURE)

if M! =0 PC= address(FALSE)

107. 寄存器清零测试

格式: TMSS A

操作: 当A为0时, 跳过下一条指令

影响状态位: 无

指令周期: 1 + (skip)

例: NOW: TMSS A

FALSE: LGOTO PROCESS

TURE: .....

指令执行前 PC=address(NOW)





指令执行后

if A=0 PC= address(TURE)

if A! =0 PC= address(FALSE)

108. 寄存器异或

格式: XOR M, m

操作: (M) xor (acc) → (M)

影响状态位: Z

指令周期: 1

例: XOR TEMP, m

指令执行前 A=0FH

TEMP=0FFH

指令执行后 A=0FH

TEMP=0F0H

109. 寄存器异或

格式: XOR M, a

操作: (M) xor (acc) → (acc)

影响状态位: Z

指令周期: 1

例: XOR TEMP, a

指令执行前 A=0FH

TEMP=0FFH

指令执行后 A=0F0H

TEMP=0FFH

110. 立即数与寄存器异或

格式: XORLA I

操作: Literal xor (acc) → (acc)

影响状态位: Z

指令周期: 1

例: XORLA 0FFH

指令执行前 A=0FH

指令执行后 A=0F0H