



PRODUCT NAME

TM58XX

TITLE

TM58XX 指令說明

APPLICATION NOTE

指令說明

1・清除位元操作

格式：BCM M, b0

操作：0→M(b0)

影響狀態位元：無

指令週期：1

例：BCM flag, b0

執行指令前 flag = 0F1H

執行指令後 flag = 0F0H

2・清除位元操作

格式：BCM M, b1

操作：0→M(b1)

影響狀態位元：無

指令週期：1

例：BCM flag, b1

執行指令前 flag = 0F2H

執行指令後 flag = 0F0H

3・清除位元操作

格式：BCM M, b2

操作：0→M(b2)

影響狀態位元：無

指令週期：1

例：BCM flag, b2

執行指令前 flag = 0F4H

執行指令後 flag = 0F0H

4・清除位元操作

格式：BCM M, b3

操作：0→M(b3)

影響狀態位元：無

指令週期：1

例：BCM flag, b3

執行指令前 flag = 0F8H

執行指令後 flag = 0F0H

5・清除位元操作

格式：BCM M, b4

操作：0→M(b4)

影響狀態位元：無

指令週期：1

例：BCM flag, b4

執行指令前 flag = 1FH

執行指令後 flag = 0FH

6・清除位元操作

格式：BCM M, b5

操作：0→M(b5)

影響狀態位元：無

指令週期：1

例：BCM flag, b5

執行指令前 flag = 2FH

執行指令後 flag = 0FH

7·清除位元操作

格式：BCM M, b6

操作：0→M(b6)

影響狀態位元：無

指令週期：1

例：BCM flag, b6

執行指令前 flag = 4FH

執行指令後 flag = 0FH

8·清除位元操作

格式：BCM M, b7

操作：0→M(b7)

影響狀態位元：無

指令週期：1

例：BCM flag, b7

執行指令前 flag = 8FH

執行指令後 flag = 0FH

9·置位操作

格式：BSM M, b0

操作：1→M(b0)

影響狀態位元：無

指令週期：1

例：BSM flag, b0

執行指令前 flag = 0F0H

執行指令後 flag = 0F1H

10·置位操作

格式：BSM M, b1

操作：1→M(b1)

影響狀態位元：無

指令週期：1

例：BSM flag, b1

執行指令前 flag = 0F0H

執行指令後 flag = 0F2H

11·置位操作

格式：BSM M, b2

操作：1→M(b2)

影響狀態位元：無

指令週期：1

例：BSM flag, b2

執行指令前 flag = 0F0H

執行指令後 flag = 0F4H

12 · 置位操作

格式：BSM M, b3

操作：1→M(b3)

影響狀態位元：無

指令週期：1

例：BSM flag, b3

執行指令前 flag = 0F0H

執行指令後 flag = 0F8H

13 · 置位操作

格式：BSM M, b4

操作：1→M(b4)

影響狀態位元：無

指令週期：1

例：BSM flag, b4

執行指令前 flag = 0FH

執行指令後 flag = 1FH

14 · 置位操作

格式：BSM M, b5

操作：1→M(b5)

影響狀態位元：無

指令週期：1

例：BSM flag, b5

執行指令前 flag = 0FH

執行指令後 flag = 2FH

15 · 置位操作

格式：BSM M, b6

操作：1→M(b6)

影響狀態位元：無

指令週期：1

例：BSM flag, b6

執行指令前 flag = 0FH

執行指令後 flag = 4FH

16 · 置位操作

格式：BSM M, b7

操作：1→M(b7)

影響狀態位元：無

指令週期：1

例：BSM flag, b7

執行指令前 flag = 0FH

執行指令後 flag = 8FH

17·清“0”位測試

格式：BTMSC M, b0

操作：若 $M(b0) = 0$,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

例：	NOW	BTMSC	flag, b0
	FALSE	LGOTO	PROCESS_PRO
	TRUE	
		

執行指令前

PC = address (NOW)

執行指令後

if	flag<b0>	=	0,
PC		=	address (TRUE);
if	flag<b0>	=	1,
PC		=	address (FALSE)

18·清“0”位測試

格式：BTMSC M, b1

操作：若 $M(b1) = 0$,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

例：	NOW	BTMSC	flag, b1
	FALSE	LGOTO	PROCESS_PRO
	TRUE	
		

執行指令前

PC = address (NOW)

執行指令後

if	flag<b1>	=	0,
PC		=	address (TRUE);
if	flag<b1>	=	1,
PC		=	address (FALSE)

19·清“0”位測試

格式：BTMSC M, b2

操作：若 $M(b2) = 0$,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

例：	NOW	BTMSC	flag,b2
	FALSE	LGOTO	PROCESS_PRO
	TRUE	
		

執行指令前

PC = address (NOW)

執行指令後

```

    if flag<b2>          = 0,
    PC                   = address (TRUE);
    if flag<b2>          = 1,
    PC                   = address (FALSE)

```

20 · 清 “0” 位測試

格式：BTMSC M, b3

操作：若 M(b3) = 0,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

```

例：      NOW      BTMSC      flag,b3
          FALSE    LGOTO      PROCESS_PRO
          TRUE     .....
          .....

```

執行指令前

PC = address (NOW)

執行指令後

```

    if flag<b3>          = 0,
    PC                   = address (TRUE);
    if flag<b3>          = 1,
    PC                   = address (FALSE)

```

21 · 清 “0” 位測試

格式：BTMSC M, b4

操作：若 M(b4) = 0,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

```

例：      NOW      BTMSC      flag,b4
          FALSE    LGOTO      PROCESS_PRO
          TRUE     .....
          .....

```

執行指令前

PC = address (NOW)

執行指令後

```

    if flag<b4>          = 0,
    PC                   = address (TRUE);
    if flag<b4>          = 1,
    PC                   = address (FALSE)

```

22 · 清 “0” 位測試

格式：BTMSC M, b5

操作：若 M(b5) = 0,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

```

例：      NOW      BTMSC      flag, b5
          FALSE    LGOTO      PROCESS_PRO
          TRUE     .....
          .....

```

執行指令前

PC = address (NOW)

執行指令後

if flag<b5> = 0,
PC = address (TRUE);
if flag<b5> = 1,
PC = address (FALSE)

23 · 清 “0” 位測試

格式：BTMSC M,b6

操作：若 M(b6) = 0,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

例： NOW BTMSC flag,b6
FALSE LGOTO PROCESS_PRO
TRUE
.....

執行指令前

PC = address (NOW)

執行指令後

if flag<b6> = 0,
PC = address (TRUE);
if flag<b6> = 1,
PC = address (FALSE)

24 · 清 “0” 位測試

格式：BTMSC M,b7

操作：若 M(b7) = 0,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

例： NOW BTMSC flag,b7
FALSE LGOTO PROCESS_PRO
TRUE
.....

執行指令前

PC = address (NOW)

執行指令後

if flag<b7> = 0,
PC = address (TRUE);
if flag<b7> = 1,
PC = address (FALSE)

25 · 置 “1” 位測試

格式：BTMSS M,b0

操作：若 M(b0) = 1,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

例： NOW BTMSS flag,b0

FALSE	LGOTO	PROCESS_PRO
TRUE	
	

執行指令前

PC = address (NOW)

執行指令後

if flag<b0>	=	1,
PC	=	address (TRUE);
if flag<b0>	=	0,
PC	=	address (FALSE)

26·置“1”位測試

格式：BTMSS M,b1

操作：若 M(b1) = 1,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

例：	NOW	BTMSS	flag,b1
	FALSE	LGOTO	PROCESS_PRO
	TRUE	
		

執行指令前

PC = address (NOW)

執行指令後

if flag<b1>	=	1,
PC	=	address (TRUE);
if flag<b1>	=	0,
PC	=	address (FALSE)

27·置“1”位測試

格式：BTMSS M,b2

操作：若 M(b2) = 1,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

例：	NOW	BTMSS	flag,b2
	FALSE	LGOTO	PROCESS_PRO
	TRUE	
		

執行指令前

PC = address (NOW)

執行指令後

if flag<b2>	=	1,
PC	=	address (TRUE);
if flag<b2>	=	0,
PC	=	address (FALSE)

28·置“1”位測試

格式：BTMSS M,b3

操作：若 M(b3) = 1,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

例： NOW BTMSS flag,b3
 FALSE LGOTO PROCESS_PRO
 TRUE

執行指令前

PC = address (NOW)

執行指令後

if flag<b3> = 1,
 PC = address (TRUE);
 if flag<b3> = 0,
 PC = address (FALSE)

29 · 置 “1” 位測試

格式：BTMSS M,b4

操作：若 M(b4) = 1,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

例： NOW BTMSS flag,b4
 FALSE LGOTO PROCESS_PRO
 TRUE

執行指令前

PC = address (NOW)

執行指令後

if flag<b4> = 1,
 PC = address (TRUE);
 if flag<b4> = 0,
 PC = address (FALSE)

30 · 置 “1” 位測試

格式：BTMSS M,b5

操作：若 M(b5) = 1,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

例： NOW BTMSS flag,b5
 FALSE LGOTO PROCESS_PRO
 TRUE

執行指令前

PC = address (NOW)

執行指令後

if flag<b5> = 1,
 PC = address (TRUE);
 if flag<b5> = 0,
 PC = address (FALSE)

31 · 置“1”位測試

格式：BTMSS M, b6

操作：若 M(b6) = 1,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

例：	NOW	BTMSS	flag, b6
	FALSE	LGOTO	PROCESS_PRO
	TRUE	
		

執行指令前

PC = address (NOW)

執行指令後

if	flag<b6>	=	1,
PC		=	address (TRUE);
if	flag<b6>	=	0,
PC		=	address (FALSE)

32 · 置“1”位測試

格式：BTMSS M, b7

操作：若 M(b7) = 1,則跳轉

影響狀態位元：無

指令週期：1 + (skip)

例：	NOW	BTMSS	flag, b7
	FALSE	LGOTO	PROCESS_PRO
	TRUE	
		

執行指令前

PC = address (NOW)

執行指令後

if	flag<b7>	=	1,
PC		=	address (TRUE);
if	flag<b7>	=	0,
PC		=	address (FALSE)

33 · 清除 WDT 計時器

格式：CLRWDT

操作：00h → WDT

影響狀態位元： \overline{TO} \overline{PD}

指令週期：1

例：CLRWDT

指令執行後：

WDT counter = 0

34 · 寫 SELECT 暫存器

格式：SELECT

操作：(A) → SELECT

影響狀態位元：無

指令週期：1

例： SELECT

指令執行前： (A) = 0FH

指令執行後： SELECT = 0FH

35 · SLEEP 操作

格式： SLEEP

操作： 00h → WDT

影響狀態位元： \overline{TO} \overline{PD}

指令週期：1

例： SLEEP

36 · I/O 做輸出入設定

格式： IODIR M

操作： A → I/O 控制暫存器

影響狀態位元：無

指令週期：1

例： IODIR PORTB

指令執行前： A = 0FFH

指令執行後： PORTB 控制暫存器 = 0FFH

37 · A 暫存器與立即數 AND 運算

格式： ANDLA I

操作： (A) AND I → A

影響狀態位元：Z

指令週期：1

例： ANDLA 01H

指令執行前： A = 0FFH

指令執行後： A = 01H

38 · A 暫存器與立即數 XOR 運算

格式： XORLA I

操作： (A) XOR I → A

影響狀態位元：Z

指令週期：1

例： XORLA 0FH

指令執行前： A = 0F0H

指令執行後： A = 0FFH

39 · 立即數送至 A 暫存器

格式： MOVL A I

操作： I → A

影響狀態位元：無

指令週期：1

例： MOVL A 0FH

指令執行前： A = 0FFH

指令執行後： A = 0FH

40 · A 暫存器與立即數 OR 運算

格式：IORLA I

操作：(A) IOR I → A

影響狀態位元：Z

指令週期：1

例：IORLA 11H

指令執行前： A = 0EEH

指令執行後： A = 0FFH

41 · 副程式帶值返回

格式：RETLA I

操作：I → A; TOS → PC

影響狀態位元：無

指令週期：2

例：CALL TAB

.....

.....

TAB

ADDAM PC,m

RETLA I0

RETLA I1

.....

RETLA IN

指令執行前： A = 01H

指令執行後： A = I1 的值

42 · 副程式呼叫

格式：LCALL I

操作：I → PC

可定址 2K (tm58p20)

可定址 1K (tm58p10)

影響狀態位元：無

指令週期：2

例：HERE LCALL THERE

指令執行前： PC= address (HERE)

指令執行後： PC= address (THERE)

TOS= address (HERE +1)

43 · 無條件跳躍

格式：LGOTO I

操作：I → PC; 可跳轉到任意地址

影響狀態位元：無

指令週期：2

例：LGOTO THERE

指令執行後： PC= address (THERE)

44 · 空操作

格式：NOP

操作：不操作

影響狀態位元：無

指令週期：1

例：NOP

45·暫存器傳送指令

格式：MOVAM M

操作：(A) → M

影響狀態位元：無

指令週期：1

例：MOVAM FLAG

指令執行前：FLAG = 0EEH

A = 0FFH

指令執行後：FLAG = 0FFH

A = 0FFH

46·暫存器取補數

格式： $\overline{\text{COMM}}$ M,m

操作： $\overline{(M)} \rightarrow M$

影響狀態位元：Z

指令週期：1

例：COMM TEMP,m

指令執行前：TEMP = 0FFH

指令執行後：TEMP = 00H

47·暫存器取補數

格式：COMM M,a

操作： $\overline{(M)} \rightarrow A$

影響狀態位元：Z

指令週期：1

例：COMM TEMP,a

指令執行前：TEMP = 0FFH

指令執行後：TEMP = 0FFH

A = 00H

48·暫存器轉移

格式：MOVMM M,m

操作：(M) → M

影響狀態位元：Z

指令週期：1

例：MOVMM TEMP,m

指令執行前：TEMP = 0FFH

指令執行後：TEMP = 0FFH

49·暫存器轉移

格式：MOVMM M,a

操作：(M) → A

影響狀態位元：Z

指令週期：1

例： **MOVM** **TEMP,a**

指令執行前： **TEMP** = 0FFH
 A = 00H

指令執行後： **TEMP** = 0FFH
 A = 0FFH

50·帶進位迴圈右移

格式：**RRM** **M,m**

操作：帶進位迴圈右移

影響狀態位元：C

指令週期：1

例： **RRM** **TEMP,m**

指令執行前： **TEMP** = 01H
 C = 0

指令執行後： **TEMP** = 00H
 C = 1

51·帶進位迴圈右移

格式：**RRM** **M,a**

操作：帶進位迴圈右移

影響狀態位元：C

指令週期：1

例： **RRM** **TEMP,a**

指令執行前： **TEMP** = 01H
 C = 0
 A = 00H

指令執行後： **TEMP** = 01H
 C = 1
 A = 00H

52·帶進位迴圈左移

格式：**RLM** **M,m**

操作：帶進位迴圈左移

影響狀態位元：C

指令週期：1

例： **RLM** **TEMP,m**

指令執行前： **TEMP** = 80H
 C = 0

指令執行後： **TEMP** = 00H
 C = 1

53·帶進位迴圈左移

格式：**RLM** **M,a**

操作：帶進位迴圈左移

影響狀態位元：C

指令週期：1

例： RLM TEMP,m
 指令執行前： TEMP = 80H
 C = 0
 A = 00H
 指令執行後： TEMP = 80H
 C = 1
 A = 00H

54 · 暫存器高低四位元交換指令

格式：SWAPM M,m

操作：(M<3:0>)→(M<7:4>)
 (M<7:4>)→(M<3:0>)

影響狀態位元：無

指令週期：1

例： SWAPM TEMP,m

指令執行前： TEMP = 0FH

指令執行後： TEMP = 0F0H

55 · 暫存器高低四位元交換指令

格式：SWAPM M,a

操作：(M<3:0>)→(A<7:4>)
 (M<7:4>)→(A<3:0>)

影響狀態位元：無

指令週期：1

例： SWAPM TEMP,a

指令執行前： TEMP = 0FH

A = 0FFH

指令執行後： TEMP = 0FH

A = 0F0H

56 · 清除 A 暫存器指令

格式：CLRA

操作：00H→(A)

影響狀態位元：Z

指令週期：1

例： CLRA

指令執行前： A = 33H

指令執行後： A = 00H

57 · 清除暫存器指令

格式：CLRM M

操作：00H→(M)

影響狀態位元：Z

指令週期：1

例： CLRM TEMP

指令執行前： TEMP = 33H

指令執行後： TEMP = 00H

58 · 暫存器加 1

格式：INCM M,m

操作： $(M) + 1 \rightarrow (M)$

影響狀態位元：Z

指令週期：1

例：INCM TEMP,m

指令執行前：TEMP = 00H

指令執行後：TEMP = 01H

59 · 暫存器加 1

格式：INCM M,a

操作： $(M) + 1 \rightarrow (A)$

影響狀態位元：Z

指令週期：1

例：INCM TEMP,a

指令執行前：TEMP = 00H

A = 00H

指令執行後：TEMP = 00H

A = 01H

60 · 加 1 爲零跳格

格式：INCMSZ M, m

操作： $(M) + 1 \rightarrow (M)$; skip if (M)=0

影響狀態位元：無

指令週期：1 + (skip)

例：HERE INCMSZ TEMP,m

LGOTO LOOP

CONT

.....

指令執行前：PC = address (HERE)

指令執行後：TEMP = TEMP + 1;

if TEMP = 0,

PC = address (CONT);

if TEMP \neq 0,

PC = address (HERE + 1);

61 · 加 1 爲零跳格

格式：INCMSZ M, a

操作： $(M) + 1 \rightarrow (M)$; skip if (A)=0

影響狀態位元：無

指令週期：1 + (skip)

例：HERE INCMSZ TEMP,a

LGOTO LOOP

CONT

.....

指令執行前：PC = address (HERE)

指令執行後：A = TEMP + 1;

if A = 0,

```

PC = address (CONT);
if A ≠ 0,
PC = address (HERE +1 );

```

62 · 暫存器減 1

格式：DECM M,m

操作：(M) - 1 →(M)

影響狀態位元：Z

指令週期：1

例：DECM TEMP,m

指令執行前：TEMP = 01H

指令執行後：TEMP = 00H

63 · 暫存器減 1

格式：DECM M,a

操作：(M) - 1 →(A)

影響狀態位元：Z

指令週期：1

例：DECM TEMP,a

指令執行前：TEMP = 01H

A = 0F0H

指令執行後：TEMP = 01H

A = 00H

64 · 減 1 為零跳格

格式：DECMSZ M,m

操作：(M) - 1 →(M);skip if (M)=0

影響狀態位元：無

指令週期：1 + (skip)

例：HERE DECMSZ TEMP,m

LGOTO LOOP

CONT

.....

指令執行前：PC = address (HERE)

指令執行後：TEMP = TEMP - 1;

if TEMP = 0,

PC = address (CONT);

if TEMP ≠ 0,

PC = address (HERE +1);

65 · 減 1 為零跳格

格式：DECMSZ M,a

操作：(M) - 1 →(A);skip if (M)=0

影響狀態位元：無

指令週期：1 + (skip)

例：HERE DECMSZ TEMP,a

LGOTO LOOP

CONT

.....

指令執行前： PC = address (HERE)
 指令執行後： A = TEMP - 1;
 if A = 0,
 PC = address (CONT);
 if A ≠ 0,
 PC = address (HERE +1);

66 · 暫存器相減

格式：SUBAM M,m

操作：(M) - (A) → M

影響狀態位元：C ; DC ; Z

指令週期：1

例： SUBAM TEMP,m

指令執行前： A = 0EH
 TEMP = 0FH
 指令執行後： A = 0EH
 TEMP = 01H

67 · 暫存器相減

格式：SUBAM M,a

操作：(M) - (A) → A

影響狀態位元：C ; DC ; Z

指令週期：1

例： SUBAM TEMP,a

指令執行前： A = 0EH
 TEMP = 0FH
 指令執行後： A = 01H
 TEMP = 0FH

68 · 暫存器 XOR 運算

格式：XORAM M,m

操作：(M) XOR (A) → M

影響狀態位元：Z

指令週期：1

例： XORAM TEMP,m

指令執行前： A = 0FH
 TEMP = 0FFH
 指令執行後： A = 0FH
 TEMP = 0F0H

69 · 暫存器 XOR 運算

格式：XORAM M,a

操作：(M) XOR (A) → A

影響狀態位元：Z

指令週期：1

例： XORAM TEMP,a

指令執行前： A = 0FH

TEMP = 0FFH
 指令執行後： A = 0F0H
 TEMP = 0FFH

70 · 暫存器 AND 運算

格式：ANDAM M,m

操作：(M) · (A) → M

影響狀態位元：Z

指令週期：1

例： ANDAM TEMP,m

指令執行前： A = 0FH
 TEMP = 11H
 指令執行後： A = 0FH
 TEMP = 01H

71 · 暫存器 AND 運算

格式：ANDAM M,a

操作：(M) · (A) → A

影響狀態位元：Z

指令週期：1

例： ANDAM TEMP,a

指令執行前： A = 0FH
 TEMP = 11H
 指令執行後： A = 01H
 TEMP = 11H

72 · 暫存器相加

格式：ADDAM M,m

操作：(M) + (A) → M

影響狀態位元：C ;DC ;Z

指令週期：1

例： ADDAM TEMP,m

指令執行前： TEMP = 0FH
 A = 0F0H
 指令執行後： TEMP = 0FFH
 A = 0F0H

73 · 暫存器相加

格式：ADDAM M,a

操作：(M) + (A) → A

影響狀態位元：C ;DC ;Z

指令週期：1

例： ADDAM TEMP,a

指令執行前： TEMP = 0FH
 A = 0F0H
 指令執行後： TEMP = 0FH
 A = 0FFH

74 · 暫存器 OR

格式：IORAM M,m

操作：(M) IOR (A) → M

影響狀態位元：Z

指令週期：1

例： IORAM TEMP,m

指令執行前：TEMP	=	01H
A	=	0F0H
指令執行後：TEMP	=	0F1H
A	=	0F0H

75 · 暫存器 OR

格式：IORAM M,a

操作：(M) IOR (A) → A

影響狀態位元：Z

指令週期：1

例： IORAM TEMP,a

指令執行前：TEMP	=	01H
A	=	0F0H
指令執行後：TEMP	=	01H
A	=	0F1H

76 · 副程式返回

格式：RET

操作：TOS → PC

影響狀態位元：無

指令週期：2

例：CONT CALL TAB

.....

.....

TAB

.....

HERE

.....

RET

指令執行前：PC = address (HERE);

指令執行後：PC = address (CONT + 1);

77 · 中斷返回指令

格式：RETI

操作：TOS → PC INTM=1

影響狀態位元：無

指令週期：2

例：CONT NOP

MOVLA	0FH
MOVAM	ACC
.....	
.....	
.....	

INTER

RETI

指令執行前： PC = address (CONT);

指令執行後： PC = address (CONT + 1);

78 · 副程式呼叫

格式：CALL I

操作：I→PC

影響狀態位元：無

指令週期：2

例： HERE CALL THERE

指令執行前： PC= address (HERE)

指令執行後： PC= address (THERE)

TOS= address (HERE + 1)

79 · 無條件跳轉指令

格式：GOTO I

操作：I→PC

影響狀態位元：無

指令週期：2

例：HERE GOTO THERE

.....

.....

.....

THERE

.....

.....

.....

指令執行前： PC= address (HERE)

指令執行後： PC= address (THERE)