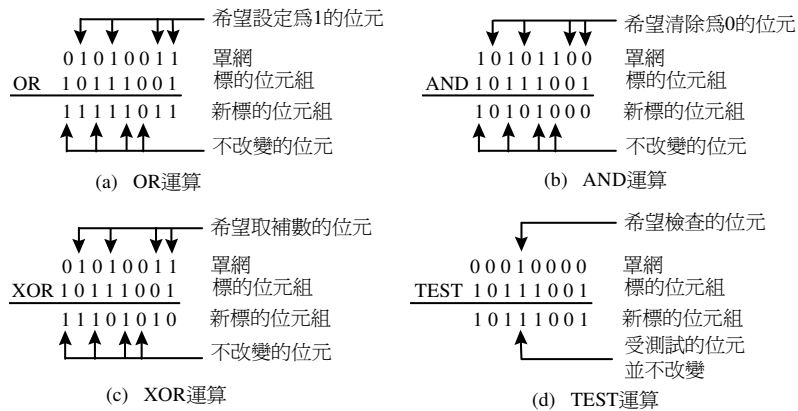


邏輯運算指令的動作



MCS-51邏輯運算指令

指令	動作	CY	AC	OV	P
ANL A,src-byte	$A \leftarrow A \wedge \text{src-byte}$ src-byte = Rn, direct, @Ri, #data8	-	-	-	*
ANL direct,A	$(\text{direct}) \leftarrow (\text{direct}) \wedge A$	-	-	-	-
ANL direct,#data8	$(\text{direct}) \leftarrow (\text{direct}) \wedge \text{\#data8}$	-	-	-	-
ORL A,src-byte	$A \leftarrow A \vee \text{src-byte}$ src-byte = Rn, direct, @Ri, #data8	-	-	-	*
ORL direct,A	$(\text{direct}) \leftarrow (\text{direct}) \vee A$	-	-	-	-
ORL direct,#data8	$(\text{direct}) \leftarrow (\text{direct}) \vee \text{\#data8}$	-	-	-	-
XRL A,src-byte	$A \leftarrow A \oplus \text{src-byte}$ src-byte = Rn, direct, @Ri, #data8	-	-	-	*
XRL direct,A	$(\text{direct}) \leftarrow (\text{direct}) \oplus A$	-	-	-	-
XRL direct,#data8	$(\text{direct}) \leftarrow (\text{direct}) \oplus \text{\#data8}$	-	-	-	-

計算布林表式

```

1 ;ex5.1-1.a51
----
2          DSEG AT 30H
0030      3  INPUT_A: DS 1 ;input a
0031      4  INPUT_B: DS 1 ;input b
0032      5  INPUT_C: DS 1 ;input c
0033      6  INPUT_D: DS 1 ;input d
0034      7  RESULT_F: DS 1 ;result f
8          ;program to evalute the boolean
9          ;
10         ;expression -->F = AB (C+D)
11         ;
----
12         CSEG AT 0000H
0000 E530 13  LGSM: MOV A,LOW INPUT_A;get INPUT_A
0002 F4    14  CPL A ;complement it
0003 5531 15  ANL A,LOW INPUT_B;AND INPUT_B
0005 F8    16  MOV R0,A ;save A temporarily
0006 E532 17  MOV A,LOW INPUT_C;get INPUT_C
0008 4533 18  ORL A,LOW INPUT_D;OR INPUT_D
000A 68    19  XRL A,R0 ;XOR A with R0
000B F534 20  MOV LOW RESULT_F,A ;save result
000D 22    21  RET
22          END

```

邏輯運算指令

```

1 ;ex5.1-2.a51
----
2          DSEG AT 30H
0030      3  BYTE1: DS 1 ;bit number
0031      4  BYTE2: DS 1 ;bit value
5          ;program to set the value of byte1
6          ;bit 3 with the value of byte2 bit 5
7          ;using logical instructions.
----
8          CSEG AT 0000H
0000 E531 9  SETBYTV: MOV A,LOW BYTE2 ;get byte2
0002 5420 10  ANL A,#20H ;extract bit 5
0004 B42006 11  CJNE A,#20H,CLRBIT ;test bit 5
0007 433008 12  SETBIT: ORL LOW BYTE1,#08H;set the
000A 020010 13  JMP RETURN ;bit 3 of byte1
000D 5330F7 14  CLRBIT: ANL LOW BYTE1,#0F7H;clear
0010 22    15  RETURN: RET ;the bit 3 of byte1
16          END

```

邏輯運算指令

```

1 ;ex5.1-3.a51
---- 2 DSEG AT 30H
0030 3 BITNO: DS 1 ;bit number
0031 4 VALUE: DS 1 ;bit value
0032 5 MEMORY: DS 1 ;memory location
6 ;program to set a given bit of MEMORY
7 ;byte with VALUE.
---- 8 CSEG AT 0000H
0000 E530 9 SETMBIT: MOV A,LOW BITNO;get bit number
0002 A831 10 MOV R0,LOW VALUE;get value
0004 900013 11 MOV DPTR,#BMASK;point to bit mask
0007 93 12 MOVC A,@A+DPTR ;get mask entry
0008 B80104 13 CJNE R0,#01H,CLRBIT;test value
000B 4232 14 SETBIT: ORL LOW MEMORY,A ;set the MEMORY
000D 8003 15 SJMP RETURN ;bit
000F F4 16 CLRBIT: CPL A ;clear the MEMORY
0010 5232 17 ANL LOW MEMORY,A ;bit
0012 22 18 RETURN: RET
0013 01020408 19 BMASK: DB 01H,02H,04H,08H
0017 10204080 20 DB 10H,20H,40H,80H
21 END

```

計數一個位元組中"1"的個數

```

1 ;ex5.1-4.a51
---- 2 DSEG AT 30H
0008 3 BCOUNT EQU 08H ;bit bumber
0030 4 TDATA: DS 1 ;test data
0031 5 COUNT: DS 1 ;result
6 ;count the number of 1-bit in a given byte
7 ;using MASK and AND instruction.
---- 8 CSEG AT 0000H
0000 7A08 9 B1CNTS: MOV R2,#BCOUNT ;set loop count
0002 75F000 10 MOV B,#00H ;zero index
0005 A9F0 11 MOV R1,B ;zero bit count
0007 90001E 12 MOV DPTR,#EMASK;point to bit mask
000A A830 13 MOV R0,LOW TDATA ;save test data

```

計數一個位元組中"1"的個數

```

000C 8830      14  BEGIN: MOV  LOW TDATA,R0 ;restore test data
000E E5F0      15      MOV  A,B      ;set emask index
0010 93         16      MOVC  A,@A+DPTR ;get test bit mask
0011 5230      17      ANL  LOW TDATA,A;test bit value
0013 B53001     18      CJNE  A,LOW TDATA,NEXT ;if not zero
0016 09         19      INC  R1      ;increase count
0017 05F0      20  NEXT: INC  B      ;increase index
0019 DAF1      21      DJNZ  R2,BEGIN ;repeat until
001B 8931      22      MOV  LOW COUNT,R1 ;store result
001D 22         23      RET
001E 01020408   24  EMASK: DB  01H,02H,04H,08H ;mask
0022 10204080   25      DB  10H,20H,40H,80H
                        26      END
    
```

MCS-51位元運算指令

指令	動作	CY	AC	OV	P
MOV C,bit	$C \leftarrow (\text{bit})$	*	-	-	-
MOV bit,C	$(\text{bit}) \leftarrow C$	-	-	-	-
ANL C,bit	$C \leftarrow (\text{bit}) \wedge C$	*	-	-	-
ANL C,/bit	$C \leftarrow (\overline{\text{bit}}) \wedge C$	*	-	-	-
ORL C,bit	$C \leftarrow (\text{bit}) \vee C$	*	-	-	-
ORL C,/bit	$C \leftarrow (\overline{\text{bit}}) \vee C$	*	-	-	-
CLR C	$C \leftarrow 0$	*	-	-	-
CLR bit	$(\text{bit}) \leftarrow 0$	-	-	-	-
SETB C	$C \leftarrow 1$	*	-	-	-
SETB bit	$(\text{bit}) \leftarrow 1$	-	-	-	-
CPL C	$C \leftarrow \overline{C}$	*	-	-	-
CPL bit	$(\text{bit}) \leftarrow \overline{(\text{bit})}$	-	-	-	-

MCS-51位元運算指令

```

1  ;ex5.2-1.a51
2  BSEG AT 20H
----
0020 3 BITA: DBIT 1 ;define input A
0021 4 BITB: DBIT 1 ;define input B
0022 5 BITC: DBIT 1 ;define input C
0023 6 BITD: DBIT 1 ;define input D
0024 7 BITF0: DBIT 1 ;define result F0
0025 8 BITF1: DBIT 1 ;define result F1
0026 9 BITF: DBIT 1 ;define result F
10 ;program to evaluate the boolean
11 ;
12 ;expression -->F = AB (C+D)
13 ;

```

MCS-51位元運算指令

```

---- 14 CSEG AT 0000H
0000 A221 15 LGSML: MOV C,LOW BITB;get INPUT B
0002 B020 16 ANL C,/LOW BITA;AND INPUT A
0004 9224 17 MOV LOW BITF0,C ;save first item
0006 A222 18 MOV C,LOW BITC;get INPUT C
0008 7223 19 ORL C,/LOW BITD;OR INPUT D
000A 9225 20 MOV LOW BITF1,C;save second item
000C B024 21 ANL C,/LOW BITF0 ;compute XOR operation
000E 9226 22 MOV LOW BITF,C ;with BITF1/BITF0
0010 A224 23 MOV C,LOW BITF0 ;+ BITF0/BITF1
0012 B025 24 ANL C,/LOW BITF1
0014 7226 25 ORL C,/LOW BITF
0016 9226 26 MOV LOW BITF,C ;save result
0018 22 27 RET
28 END

```

MCS-51位元測試指令

指令	動作	CY	AC	OV	P
JC disp8	PC ← PC + 2; If C = 1 then PC ← PC + disp8	-	-	-	-
JNC disp8	PC ← PC + 2; If C = 0 then PC ← PC + disp8	-	-	-	-
JB bit,disp8	PC ← PC + 3; If (bit) = 1 then PC ← PC + disp8	-	-	-	-
JBC bit,disp8	PC ← PC + 3; If (bit) = 1 then (bit) ← 0 and PC ← PC + disp8	-	-	-	-
JNB bit,disp8	PC ← PC + 3; If (bit) = 0 then PC ← PC + disp8	-	-	-	-

靜態位元測試指令

```

1 ;ex5.2-2.a51
---- 2 BSEG AT 20H
0020 3 BYTE1: DBIT 8 ;control word 1
0028 4 BYTE2: DBIT 8 ;control word 2
5 ;program to set the value of bit 3 of
6 ;byte1 with the value of bit 5 of byte2
7 ;using bit manipulation instructions
---- 8 CSEG AT 0000H
0000 304504 9 JNB LOW BYTE2.5,CLRBIT;test bit 5
0003 D203 10 SETB LOW BYTE1.3;set bit 3 of
0005 8002 11 SJMP RETURN ;byte1
0007 C203 12 CLRBIT: CLR LOW BYTE1.3;clear bit 3
0009 22 13 RETURN: RET ;of byte1
14 END

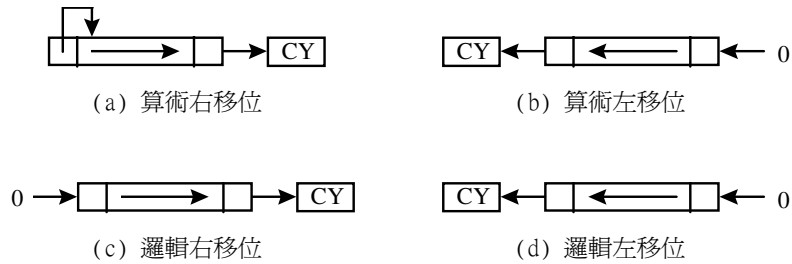
```

靜態位元測試指令

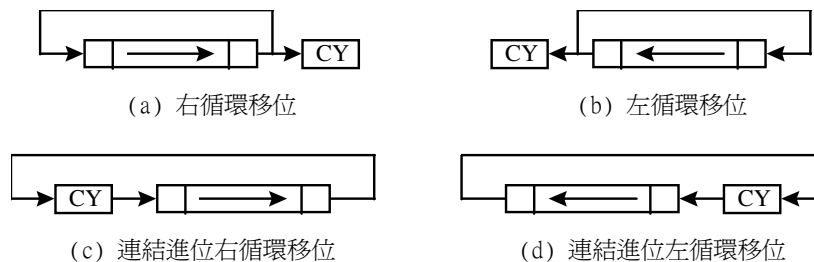
```

1 ;ex5.2-3.a51
---- 2 BSEG AT 20H
0020 3 BYTE1: DBIT 8 ;control word 1
0028 4 BYTE2: DBIT 8 ;control word 2
5 ;program to set the value of bit 3 of
6 ;byte1 with the value of bit 5 of byte2
7 ;using bit manipulation instructions
---- 8 CSEG AT 0000H
0000 D203 9 SETB LOW BYTE1.3;set bit 3 of byte1
0002 204502 10 JB LOW BYTE2.5,RETURN;test bit 5
0005 C203 11 CLRBIT: CLR LOW BYTE1.3;clear bit 3 of byte 1
0007 22 12 RETURN: RET
13 END
    
```

算術與邏輯移位



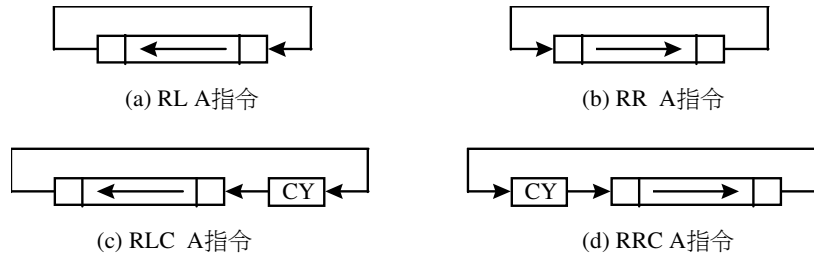
循環移位與連結進位循環移位



MCS-51移位與循環移位指令

指令	動作	CY	AC	OV	P
RL A	左循環移位累積器 A 的內容一個位元位置	-	-	-	-
RLC A	連結進位左循環移位累積器 A 的內容一個位元位置	*	-	-	*
RR A	右循環移位累積器 A 的內容一個位元位置	-	-	-	-
RRC A	連結進位右循環移位累積器 A 的內容一個位元位置	*	-	-	*

MCS-51的移位與循環移位指令



RL指令

```

1      ;ex5.3-1.a51
2      ;swap two nibbles in accumulator A
3      ;using RL A instruction
----
4      CSEG AT 0000H
0000 23      5      SWAP4B: RL A ;rotate accumulator
0001 23      6      RL A ;A left 4 bits
0002 23      7      RL A
0003 23      8      RL A
0004 22      9      RET
10     END
    
```

循環移位指令使用例

```

1 ;ex5.3-2.a51
----
2      DSEG AT 30H
0008 3      BCOUNT EQU 08H ;bit number
0030 4      TDATA: DS 1 ;test data
0031 5      COUNT: DS 1 ;result
6      ;count the number of 1-bit in a given byte
7      ;using rotation instruction.
----
8      CSEG AT 0000H
0000 7908 9      B1CNTS: MOV R1, #BCOUNT ;set loop counter
0002 75F000 10     MOV B, #00 ;zero 1's counter
0005 E530 11     MOV A, LOW TDATA ;get test data
0007 13 12     AGAIN: RRC A ;test bit value
0008 5002 13     JNC NEXT ;if not zero
000A 05F0 14     INC B ;increase 1's count
000C D9F9 15     NEXT: DJNZ R1, AGAIN ;loop BCOUNT times
000E 85F031 16     MOV LOW COUNT, B; store result
0011 22 17     RET
18     END
    
```

MCS-51 CPU控制指令

指令	動作	CY	AC	OV	P
NOP	沒有動作	-	-	-	-

軟體延遲程式

```

1 ;ex5.4-1.a51
00C8      2  COUNT EQU 200
----      3  CSEG AT 0000H
0000 7AC8      4  DELAY1MS: MOV R2,#COUNT ;1 cycle
0002 00      5  KTIME: NOP ;1 cycle
0003 00      6  NOP ;1 cycle
0004 00      7  NOP ;1 cycle
0005 00      8  NOP ;1 cycle
0006 00      9  NOP ;1 cycle
0007 DAF9     10  DJNZ R2,KTIME ;2 cycles
0009 22     11  RET ;2 cycles
          12  END
    
```

MCS-51 旗號位元運算指令

指令	動作	CY	EA	OV	P
CLR EA	抑制所有中斷要求輸入(清除 EA 為 0)。	-	0	-	-
SETB EA	致能所有中斷要求輸入(設定 EA 為 1)。	-	1	-	-
CLR P	清除同位旗號位元為 0。	-	-	-	0
SETB P	設定同位旗號位元為 1。	-	-	-	1
CLR C	清除進位旗號位元為 0。	0	-	-	-
CPL C	將進位旗號位元取補數。	$\bar{C}$	-	-	-
SETB C	設定進位旗號位元為 1。	1	-	-	-